CRWMS/M&O	**Calculation Cover Sheet** *Complete only applicable items.*	1. QA: L Page: 1 Of: 65

2. Calculation Title CRC Depletion Calculations for Quad Cities Unit 2			MOL.19990929.0121
3. Document Identifier (including Revision Number) B00000000-01717-0210-00009 REV 01			4. Total Pages 65
5. Total Attachments 8	6. Attachment Numbers - Number of pages in each I-151, II-82, III-29, IV-163, V-146, VI-256, VII-37, VIII-8		
	Print Name	Signature	Date
7. Originator	David P. Henderson	D A Thomas FOR DPH	09/29/1999
8. Checker	Peter M. Noel	Peter M Noel	9/29/1999
9. Lead	Daniel A. Thomas	D A Thomas	09/29/1999

10. Remarks

Attachments III, IV, V, and VIII are contained on attachment compact disks for this calculation file, Reference 7.10. The numbers shown in Box 6 for Attachments III, IV, V, and VIII refer to the number of pages in the hard-copy listing of each file's attributes on the compact disks.

TBV-1349 has been assigned to Reference 7.3 as a result of the DOE letter titled "Accepted Data Call" from R.E. Spence to J.L. Younker, issued on 07/27/1999 (MOL: 19990811.0170).

TBV-1349 has been cleared with the issuance of the DOE letter titled "Office of Project Execution (OPE) Approval of LV.WP.TWD.08/99-131 as Accepted Data" from Suzanne P. Mellington to T.W. Doering (Reference 7.11) on 09/20/1999 (MOL.19990927.0472)

Revision History

11. Revision No.	12. Description of Revision
00	Initial Issuance
01	Complete revision to incorporate revised input data and control blade insertion data.

CONTENTS

1. PURPOSE

The purpose of this calculation is to document the Quad Cities Unit 2 boiling water reactor (BWR) fuel depletion calculations performed as part of the commercial reactor critical (CRC) evaluation program. The CRC evaluations constitute benchmark calculations that support the development and validation of the neutronics models used for criticality analyses involving commercial spent nuclear fuel in a geologic repository. The revision of this calculation incorporates control blade effects and minor variations in the SAS2H assembly modeling.

2. METHOD

The calculational method used to perform the Quad Cities Unit 2 fuel depletion calculations consisted of using the SAS2H control sequence of the SCALE, Version 4.3, code system (Reference 7.1) to deplete the necessary fuel assemblies. The various fuel assemblies were depleted through their unique operating histories such that their modified fuel isotopic compositions would be available at specific exposure times corresponding to the times (statepoints) at which detailed core reactivity calculations would be performed. The fuel assembly depletion calculations were based on core follow information for each assembly.

3. ASSUMPTIONS

3.1 The assumption is that the approximation of uniformly distributed non-fuel lattice cells in the Path B models of the SAS2H calculations, as described in Section 5.4, is considered acceptable within the fidelity of these calculations. The basis for this assumption is provided in Section S2.2.3.1 of Volume 1, Rev. 5 in Reference 7.1. This assumption was used throughout all of the depletion calculations documented in Section 5.

3.2 The assumption is that the use of the 44-group ENDF/B-V cross section library, which was originally collapsed from the 238-group ENDF/B-V cross section library using a fuel cell spectrum described by a pressurized water reactor assembly, is considered acceptable for BWR depletions. The applicability of this cross section collapsing method for BWR assemblies has not been determined. A known bias caused by inadequate plutonium cross section representation has been identified (Volume 0, Reference 7.1). The basis for this assumption is provided in Section M4.2.9 of Volume 3, Rev. 5 in Reference 7.1. Additional confirmatory sensitivity calculations can be performed to evaluate the impact of this assumption. This assumption was used throughout all of the depletion calculations documented in Section 5.

3.3 The assumption is that the approximations made in modeling gadolinia-bearing fuel rods (GDRs), water rods, axial and radial enrichment distributions, and two-phase moderator density profiles due to the limitations of SAS2H are considered acceptable for BWR depletion calculations. The complexity of BWR fuel designs requires unique modeling methods be made in the development of the Path B model for the SAS2H calculations described in Section 5.4. The basis for this assumption is engineering judgement. The impact of this assumption on the accuracy of the results contained in this calculation has not been quantified. Additional confirmatory sensitivity calculations can be performed to evaluate the impact of this modeling assumption. This assumption was used throughout all of the depletion calculations documented in Section 5.

3.4 The assumption is that the collapsed 10 node assembly model (Reference 7.3, p. 33) used for the fuel assembly depletion calculations documented in this calculation is considered acceptable. The basis for this assumption is engineering judgement. Nodal data for Quad Cities Unit 2 BWR assemblies is available in 24-node format. The averaging and combining of various node data to collapse to a 10 node format was a necessary approximation in this calculation because of time constraints and the large number of calculations required to process assemblies. The impact of this assumption on the accuracy of the results contained herein has not been quantified. Additional confirmatory sensitivity calculations can be performed to evaluate the impact of this modeling assumption. This assumption was used throughout all of the depletion calculations documented in Section 5.

3.5 The assumption is that the approximations required for the modeling of control blades, modeling of control blade exposure time (Reference 7.3, p. 546), and nodal average power for controlled nodes during the life of an assembly are considered acceptable. These approximations, described in Section 5, are required due to the limitations of both SAS2H and the software routine automation required to operate SAS2H. This assumption incorporates the fact that controlled nodes were depleted at full power in the SAS2H model without adjustment for local power decreases in the vicinity of a control blade. The basis for this assumption is engineering judgement. The impact of this assumption on the accuracy of the results contained in this calculation has not been quantified. Additional confirmatory sensitivity calculations can be performed to evaluate the impact of this modeling assumption. This assumption was used throughout all of the depletion calculations documented in Section 5.

3.6 The assumption is that small variations in nodal depleted fuel isotopic concentrations from one depletion calculation to the subsequent depletion calculation are inconsequential. Due to limitations in the SCALE4.3 version of SAS2H, software routine automation is required to operate SAS2H for CRC depletion calculations. The software routine automation is described in Attachments I and II. The basis for this assumption is engineering judgement. The impact of this assumption on the accuracy of the results contained in this calculation has not been quantified. Additional confirmatory sensitivity calculations can be performed to evaluate the impact of this modeling assumption. This assumption was used throughout all of the depletion calculations documented in Section 5.

3.7 The assumption is that the use of nodal average powers in SAS2H depletion calculations that reflect full power reactor operation without adjustment for plant capacity factor does not adversely affect the calculated isotopic concentrations. The basis for this assumption is engineering judgement. The impact of this assumption on the accuracy of the results contained in this calculation has not been quantified. Additional confirmatory sensitivity calculations can be performed to evaluate the impact of this modeling assumption. This assumption was used throughout all of the depletion calculations documented in Section 5.

4. USE OF COMPUTER SOFTWARE

4.1. SOFTWARE APPROVED FOR QA WORK

4.1.1. SAS2H

The SAS2H control module of the SCALE, Version 4.3, modular code system (Reference 7.1) was used to perform the fuel assembly depletion calculations required for the Quad Cities Unit 2 CRC evaluations. The software specifications are as follows:

- Program Name: SAS2H of the SCALE Modular Code System
- Version/Revision Number: Version 4.3
- Computer Software Configuration Item (CSCI) Number: 30011 V4.3
- Computer Type: Hewlett Packard (HP) 9000 Series Workstations
- Computer Unit : CRWMS M&O 700887 (BLOOM)

The input and output files for the various SAS2H calculations were documented in the attachments and Reference 7.10 as described in Section 5, such that an independent repetition of the software use could be performed. The SAS2H software used was: (a) appropriate for the application of commercial fuel assembly depletion, (b) used only within the range of validation as documented in References 7.1 and 7.2, (c) obtained from the Software Configuration Manager in accordance with appropriate procedures.

4.2. SOFTWARE ROUTINES

The description documentation for each of the software routines identified in this section, other than the acquired software routine Excel described in Section 4.2.1, contains the following information:

- Descriptions and equations of mathematical algorithms
- Description of software routine including execution environment
- Description of test cases
- Description of test results
- Range of input parameter values for which results were verified
- Identification of any limitations on software routine applications or validity
- Reference list of all documentation relevant to the qualification
- Directory listing of executable and data files
- Computer listing of source code

4.2.1. EXCEL

- Title: Excel
- Version/Revision Number: Microsoft® Excel 97

The Excel spreadsheet program was used for simple numeric calculations as documented in Section 5 of this calculation. The user-defined formulas, inputs, and results were documented in sufficient detail in Section 5 to allow an independent repetition of the various computations.

4.2.2. SPACE V01

- Title: System Processor and CRAFT V4C Executor (SPACE)
- Version/Revision Number: Version 01

The SPACE software routine produced the required CRAFT input files for each assembly and directed the execution of CRAFT. The assembly input file formats for SPACE and CRAFT input files were documented in Attachment III such that an independent repetition of the software routine use could be performed. The description of the SPACE, Version 01, software routine was provided in Attachment II of this calculation.

4.2.3. CRAFT V4C

- Title: Commercial Reactor Assembly Follow Taskmaster (CRAFT)
- Version/Revision Number: Version 4C

The CRAFT software routine produced the input and directed the execution for the various SAS2H calculations required to deplete a commercial reactor fuel assembly to support a CRC evaluation. The output from SPACE-CRAFT-SAS2H depletion calculations was documented in Attachments IV and V and Reference 7.10, such that an independent repetition of the software routine use could be performed. The description of the CRAFT, Version 4C, software routine was provided in Attachment I of this calculation.

4.3. MODELS

None.

5. CALCULATION

5.1. QUAD CITIES UNIT 2 CRC EVALUATION DESCRIPTION

The Quad Cities Unit 2 CRC evaluations were performed at six statepoints: Cycle 13 [0.0, 10.1, 112.94, 324.73 Effective Full-Power Days (EFPD)], Cycle 14 [0.0 and 211.09 EFPD]. Each statepoint represented a specific time when the reactor was brought to a zero power, critical condition where the effective critical multiplication factor (k_{eff}) for the core is determined by experiment to equal 1. The CRC evaluations of each of these critical statepoints involved the use of SAS2H to deplete the various fuel assemblies and MCNP (Reference 7.4) to model the reactor core such that the k_{eff} value at each of the critical statepoints could be predicted to demonstrate the ability of the dual code system. Hence, the objective of each CRC statepoint evaluation was to predict the reactor core k_{eff} as close to measurement as possible (the measurement is always $k_{eff} = 1$). The objective of the SAS2H depletion calculations documented in this calculation was to provide the depleted fuel and burnable poison isotopic compositions to be used in the corresponding CRC reactivity calculations.

Fuel isotopic compositions were calculated with SAS2H for each depleted fuel assembly in each of the critical statepoint configurations to facilitate MCNP modeling. The Quad Cities Unit 2 statepoint calculations required the depletion of fuel assemblies from twelve fuel batches. Fuel assembly design characteristics may vary between each fuel batch. Section 5.2 presents the input parameters required to perform the various fuel assembly depletion calculations. Sections 5.3 through 5.6 describe how the parameters listed in Section 5.2 were utilized to perform the SAS2H depletion calculations relevant to the CRC statepoint evaluations. The SPACE-CRAFT description and user information provided in Attachments I and II are essential for understanding the SAS2H modeling techniques employed in the calculations. The information provided in Attachments I and II, the input parameters provided in Section 5.2, and the SPACE-CRAFT input files contained in Attachment III and Reference 7.10, work together to provide a complete description of how all of SAS2H depletion calculations were performed.

The number of digits in the values cited herein may be the result of a calculation or may reflect the input from another source; consequently, the number of digits should not be interpreted as an indication of accuracy.

5.2. INPUT SPECIFICATIONS FOR DEPLETION CALCULATIONS

The information documented in this section describes the design specifications and irradiation histories for the fuel assemblies required for the Quad Cities Unit 2 CRC evaluations. Most of the input specifications presented in this section were obtained from Reference 7.3. Initially, these inputs were not classified as "accepted data" per the retroactive procedural requirement of AP-SIII.2Q initiated by the July 27, 1999 issuance of the DOE Letter, "Accepted Data Call", from R. E. Spence to J. L. Younker (Reference 7.9). As a result, TBV-1349 was assigned to Reference 7.3. Subsequently, DOE issued a letter (Reference 7.11), approving the request to identify the inputs from Reference 7.3 as accepted data. Consequently, these inputs are hereupon considered accepted data. Inputs obtained from Reference 7.5 , Reference 7.7, and Reference 7.8 are considered accepted data. Input specifications from other sources are as indicated. The Quad Cities Unit 2 CRC evaluations included fuel assemblies from twelve fuel types (batches) with the following fuel type identifications: 6, 7, 9, 10, 12, 14, 15, 16, 17, 18, 1, and 2.

During depletion calculations for Quad Cities Unit 2, input for assembly fuel type 16 (designated as J), was incorrectly specified. The "J" assembly input for these depletions did not contain the correct mass of gadolinia in the gadolinia-bearing fuel rods. The depletion calculations for the "J" assembly were rerun correctly and documented in Attachment VIII and Reference 7.10.

5.2.1. FUEL ASSEMBLY DESCRIPTIONS

Table 5-1 contains a description of the fuel assemblies for fuel types 6, 7, 9, 10, 12, 14, 15, 16, 17, 18, 1, and 2 of Quad Cities Unit 2. All fuel assemblies within a given fuel type have the same characteristics as presented in Table 5-1. The assemblies are modeled and depleted in a 10 node collapsed scheme as provided in Reference 7.3, p. 290. The node mass and gadolinia-bearing fuel rod smeared pellet density in Table 5-1 are calculated values and are determined by Equations 5-1 and 5-2.

Equation 5-1. Mass of Uranium in Node

$$\text{Nodemass} = \text{RLM} \cdot (\#\,\text{fuelrods}) \cdot (\text{nodeheight})$$

where: *RLM* is the rod linear mass in g/cm (Reference 7.3, p. 545); *#fuelrods* is the number of fuel rods in the node; *nodeheight* is the height of the node in centimeters.

Equation 5-2. Gadolinia-Bearing Fuel Rod Smeared Pellet Density

$$\text{Density}_{\text{smeared}} = \text{RLM} \cdot \frac{1}{\pi \cdot \left(\frac{\text{cladid}}{2}\right)^2} \cdot \frac{1}{\text{wfrac}_{\text{oxygen}}}$$

where: *RLM* is the rod linear mass in g/cm; *cladid* is the inner dimension of the fuel rod clad *wfrac* is the weight fraction of oxygen in the gadolinia-fuel mixture (0.8815).

Table 5-1. Fuel Assembly Descriptions for the Quad Cities Unit 2 CRC Evaluations (Reference 7.3, pp. 5-15)

	Fuel Type Identifier					
Parameter	**6 (A)**	**7 (B)**	**9 (C)**	**10 (D)**	**12 (F)**	**14 (E)**
Assembly Average Enrich. U-235	2.99	2.99	3.00	3.16	2.99	3.10
Node 1						
Average Enrich. U-235	0.71	0.71	0.71	0.71	0.71	0.71
# GDRs[1]	0	0	0	0	0	0
GDR Enrichment	0.0	0.0	0.0	0.0	0.0	0.0
Node Mass (kg)	7.2856	7.2759	7.2025	7.1910	7.2334	7.2315
Node Height (cm)	15.24	15.24	15.24	15.24	15.24	15.24
Node 2						
Average Enrich. U-235	3.19	3.19	3.20	3.37	3.22	3.34
# GDRs	7	7	9	7	9	7
GDR Enrichment	3.0	4.0	3.0	4.0	3.0	3.0
Node Mass (kg)	14.5712	14.5517	14.4049	14.3820	14.4668	14.4630
Node Height (cm)	30.48	30.48	30.48	30.48	30.48	30.48
Node 3						
Average Enrich. U-235	3.19	3.19	3.20	3.37	3.22	3.34
# GDRs	7	7	9	7	9	7
GDR Enrichment	3.0	4.0	3.0	4.0	3.0	3.0
Node Mass (kg)	14.5712	14.5517	14.4049	14.3820	14.4668	14.4630
Node Height (cm)	30.48	30.48	30.48	30.48	30.48	30.48
Node 4						
Average Enrich. U-235	3.19	3.19	3.20	3.37	3.22	3.34
# GDRs	7	7	9	7	9	7
GDR Enrichment	3.0	4.0	3.0	4.0	3.0	3.0
Node Mass (kg)	21.8568	21.8276	21.6074	21.5730	21.7001	21.6944
Node Height (cm)	45.72	45.72	45.72	45.72	45.72	45.72
Node 5						
Average Enrich. U-235	3.19	3.19	3.20	3.37	3.37	3.50
# GDRs	7	7	9	7	9	7
GDR Enrichment	3.0	4.0	3.0	4.0	3.0	3.0
Node Mass (kg)	14.5712	14.5517	14.4049	14.3820	14.4668	14.4630
Node Height (cm)	30.48	30.48	30.48	30.48	30.48	30.48
Node 6						
Average Enrich. U-235	3.19	3.19	3.20	3.37	3.37	3.50
# GDRs	7	7	9	7	9	7
GDR Enrichment	3.0	4.0	3.0	4.0	3.0	3.0
Node Mass (kg)	21.8568	21.8276	21.6074	21.5730	21.7001	21.6944
Node Height (cm)	45.72	45.72	45.72	45.72	45.72	45.72
Node 7						
Average Enrich. U-235	3.19	3.19	3.20	3.37	3.37	3.50
# GDRs	7	7	9	7	9	7
GDR Enrichment	3.0	4.0	3.0	4.0	3.0	3.0
Node Mass (kg)	21.8568	21.8276	21.6074	21.5730	21.7001	21.6944
Node Height (cm)	45.72	45.72	45.72	45.72	45.72	45.72
Node 8						
Average Enrich. U-235	3.19	3.19	3.20	3.37	3.37	3.50
# GDRs	7	7	9	7	2/9	2/7
GDR Enrichment	3.0	4.0	3.0	4.0	4.0/3.0	4.0/3.0
Node Mass (kg)	21.8568	21.8276	21.6074	21.5730	21.7001	21.6944
Node Height (cm)	45.72	45.72	45.72	45.72	45.72	45.72

	Fuel Type Identifier					
Parameter	**6 (A)**	**7 (B)**	**9 (C)**	**10 (D)**	**12 (F)**	**14 (E)**
Node 9 Average Enrich. U-235 # GDRs GDR Enrichment Node Mass (kg) Node Height (cm)	 3.19 7 3.0 30.1424 60.96	 3.19 7 4.0 29.1034 60.96	 3.20 9 3.0 28.8098 60.96	 3.37 7 4.0 28.7640 60.96	 3.22 9 3.0 21.7001 45.72	 3.34 7 3.0 21.6944 45.72
Node 10 Average Enrich. U-235 # GDRs GDR Enrichment Node Mass (kg) Node Height (cm)	 0.71 0 0.0 7.2856 15.24	 0.71 0 0.0 7.2759 15.24	 0.71 0 0.0 6.1221 15.24	 0.71 0 0.0 6.3520 15.24	 0.71 0 0.0 13.1407 30.48	 0.71 0 0.0 13.3783 30.48
Active Fuel Height (cm)	365.76	365.76	365.76	365.76	365.76	365.76
Fuel Pellet OD[2] (cm)	1.0414	1.0414	1.0414	1.0414	1.0414	1.0414
Fuel Rod Clad OD (cm)	1.2268	1.2268	1.2268	1.2268	1.2268	1.2268
Fuel Rod Clad ID[3] (cm)	1.0643	1.0643	1.0643	1.0643	1.0643	1.0643
Channel Thickness (cm)	0.2032	0.2032	0.2032	0.2032	0.2032	0.2032
Channel Inside Width (cm)	13.4061	13.4061	13.4061	13.4061	13.4061	13.4061
Water Rod Material	Zirc4[4]	Zirc4	Zirc4	Zirc4	Zirc4	Zirc4
Water Rod OD (cm)	2.4536	2.4536	2.6187	2.6187	2.6187	2.6187
Water Rod ID (cm)	2.1285	2.1285	2.4561	2.4561	2.4561	2.4561
Array Size	8 x 8	8 x 8	8 x 8	8 x 8	8 x 8	8 x 8
Number of Fuel Rods[5]	62	62	60	60	60	60
Pin Pitch (cm)	1.6256	1.6256	1.6256	1.6256	1.6256	1.6256
Assembly Pitch (cm)	15.24	15.24	15.24	15.24	15.24	15.24

Table 5-1. (cont.) Fuel Assembly Descriptions for the Quad Cities Unit 2 CRC Evaluations

	Fuel Type Identifier					
Parameter	**15 (K)**	**16 (J)**	**17 (G)**	**18 (H)**	**1 (L)**	**2 (M)**
Assembly Average Enrich. U-235	3.10	3.08	2.86	2.86	3.12	3.16
Node 1 Average Enrich. U-235 # GDRs GDR Enrichment Node Mass (kg) Node Height (cm)	 0.71 0 0.0 7.2230 15.24	 0.71 0 0.0 7.2148 15.24	 0.71 0 0.0 7.2316 15.24	 0.71 0 0.0 7.2231 15.24	 0.71 0 0.0 7.1856 15.24	 0.71 0 0.0 7.1765 15.24
Node 2 Average Enrich. U-235 # GDRs GDR Enrichment Node Mass (kg) Node Height (cm)	 3.34 7 3.0 14.4459 30.48	 3.32 8/2 4.0/3.0 14.4296 30.48	 3.06 7 3.0 14.4632 30.48	 3.06 7 3.0 14.4462 30.48	 3.36 3/4 4.0/3.0 14.3712 30.48	 3.41 6/2 4.0/3.0 14.3530 30.48

	Fuel Type Identifier					
Parameter	**15 (K)**	**16 (J)**	**17 (G)**	**18 (H)**	**1 (L)**	**2 (M)**
Node 3						
Average Enrich. U-235	3.34	3.32	3.06	3.06	3.36	3.41
# GDRs	7	8/2	7	7	3/4	6/2
GDR Enrichment	3.0	4.0/3.0	3.0	3.0	4.0/3.0	4.0/3.0
Node Mass (kg)	14.4459	14.4296	14.4632	14.4462	14.3712	14.3530
Node Height (cm)	30.48	30.48	30.48	30.48	30.48	30.48
Node 4						
Average Enrich. U-235	3.34	3.32	3.06	3.06	3.36	3.41
# GDRs	7	8/2	7	7	3/4	6/2
GDR Enrichment	3.0	4.0/3.0	3.0	3.0	4.0/3.0	4.0/3.0
Node Mass (kg)	21.6689	21.6444	21.6948	21.6693	21.5568	21.5295
Node Height (cm)	45.72	45.72	45.72	45.72	45.72	45.72
Node 5						
Average Enrich. U-235	3.50	3.48	3.24	3.24	3.54	3.57
# GDRs	7	8/2	7	7	1/6	6/2
GDR Enrichment	3.0	4.0/3.0	3.0	3.0	4.0/3.0	4.0/3.0
Node Mass (kg)	14.4459	14.4296	14.4632	14.4462	14.3712	14.3530
Node Height (cm)	30.48	30.48	30.48	30.48	30.48	30.48
Node 6						
Average Enrich. U-235	3.50	3.48	3.24	3.24	3.54	3.57
# GDRs	7	8/2	7	7	1/6	6/2
GDR Enrichment	3.0	4.0/3.0	3.0	3.0	4.0/3.0	4.0/3.0
Node Mass (kg)	21.6689	21.6444	21.6948	21.6693	21.5568	21.5295
Node Height (cm)	45.72	45.72	45.72	45.72	45.72	45.72
Node 7						
Average Enrich. U-235	3.50	3.48	3.24	3.24	3.54	3.57
# GDRs	7	8/2	7	7	1/6	6/2
GDR Enrichment	3.0	4.0/3.0	3.0	3.0	4.0/3.0	4.0/3.0
Node Mass (kg)	21.6689	21.6444	21.6948	21.6693	21.5568	21.5295
Node Height (cm)	45.72	45.72	45.72	45.72	45.72	45.72
Node 8						
Average Enrich. U-235	3.50	3.48	3.24	3.24	3.54	3.57
# GDRs	7	8/2	2/7	7	1/6	6/2
GDR Enrichment	3.0	4.0/3.0	4.0/3.0	3.0	4.0/3.0	4.0/3.0
Node Mass (kg)	21.6689	21.6444	21.6948	21.6693	21.5568	21.5295
Node Height (cm)	45.72	45.72	45.72	45.72	45.72	45.72
Node 9						
Average Enrich. U-235	3.34	3.32	3.06	3.06	3.36	3.41
# GDRs	7	8/2	7	7	7	8
GDR Enrichment	3.0	4.0/3.0	3.0	3.0	3.0	3.0
Node Mass (kg)	21.6689	21.6444	21.6948	21.6693	21.5568	21.5295
Node Height (cm)	45.72	45.72	45.72	45.72	45.72	45.72
Node 10						
Average Enrich. U-235	0.71	0.71	0.71	0.71	0.71	0.71
# GDRs	0	0	0	0	0	0
GDR Enrichment	0.0	0.0	0.0	0.0	0.0	0.0
Node Mass (kg)	13.6033	13.2271	13.3785	13.6035	13.5329	13.3961
Node Height (cm)	30.48	30.48	30.48	30.48	30.48	30.48
Active Fuel Height (cm)	365.76	365.76	365.76	365.76	365.76	365.76
Fuel Pellet OD (cm)	1.0414	1.0414	1.0414	1.0414	1.0414	1.0414
Fuel Rod Clad OD (cm)	1.2268	1.2268	1.2268	1.2268	1.2268	1.2268
Fuel Rod Clad ID (cm)	1.0643	1.0643	1.0643	1.0643	1.0643	1.0643

	Fuel Type Identifier					
Parameter	**15 (K)**	**16 (J)**	**17 (G)**	**18 (H)**	**1 (L)**	**2 (M)**
Channel Thickness (cm)	0.2032	0.2032	0.2032	0.2032	0.254[6]	0.254
Channel Inside Width (cm)	13.4061	13.4061	13.4061	13.4061	13.4061	13.4061
Water Rod Material	Zirc4	Zirc4	Zirc4	Zirc4	Zirc4	Zirc4
Water Rod OD (cm)	2.6187	2.6187	2.6187	2.6187	2.6187	2.6187
Water Rod ID (cm)	2.4561	2.4561	2.4561	2.4561	2.4561	2.4561
Array Size	8 x 8	8 x 8	8 x 8	8 x 8	8 x 8	8 x 8
Number of Fuel Rods	60	60	60	60	60	60
Pin Pitch (cm)	1.6256	1.6256	1.6256	1.6256	1.6256	1.6256
Assembly Pitch (cm)	15.24	15.24	15.24	15.24	15.24	15.24

[1] GDR = Gadolinia-bearing fuel rod; GDR enrichment is the wt% of Gd_2O_3 in Gd_2O_3/UO_2.
[2] OD = Outer diameter.
[3] ID = Inner diameter.
[4] Zirc4 = Material Zircaloy-4.
[5] The number of fuel rods in an assembly is based on the actual number of UO_2 or UO_2/Gd_2O_3 rods that are present in each node of the active fuel. In fuel assemblies E, F, G, H, L, and M the second axial node from the top of the assembly (node 23) which contains natural uranium (axial blanket) can have a different number of uranium rods. For modeling purposes, the number of uranium rods in this unique node is assumed to be the same as the number of active fuel rods present throughout the assembly. Because this approximation occurs in a node of the axial blanket it is assumed that there will be a negligible effect on the depletion calculation for the assembly.
[6] For assemblies L and M, 0.254 cm is used for the entire channel thickness. This is the corner thickness of the assembly channel (Reference 7.3, p. 5). The wall thickness is 0.1651 cm. This small difference in mass of Zirc4 is expected to have a negligible effect on the neutron spectrum in the depletion calculation.

In some cases, nodes have more than one gadolinia enrichment for GDRs and the actual number of rods for each type of GDR may vary. In these cases, the weighted average enrichment in the node for all GDRs was calculated. For example, if there are 2 GDRs at 4.0 weight percent (wt%) and 9 GDRs at 3.0 wt% then the number of GDR rods and weighted average enrichment would be 11 and 3.18 wt%, respectively. Typically, these different enrichment GDRs all have the same UO_2 enrichment within a node. Table 5-1 provides the specific inputs of GDR numbers and enrichments for each node that are to be used in this averaging method.

GDR material wt% are required for each GDR specification in a node. These percentages are listed in the SPACE input files and are used to define the initial GDR material in the SAS2H depletion calculation. Multiple depletion calculations are performed for each node using the SPACE-CRAFT-SAS2H combination. As the depletion calculation progresses and the gadolinia is depleted, the GDR material isotopic wt% are recalculated by the CRAFT software routine and entered into the next SAS2H calculation. This process is discussed in more detail in Section 5.4.

Equation 5-3 is used to calculate the uranium isotopic wt% that will be used in the initial GDR material specification (Reference 7.6, p. 20). The calculation of initial GDR material isotopic wt% uses Equations 5-4 through 5-10. The inputs required to calculate these percentages include the UO_2 enrichment, the average gadolinia enrichment, the number of GDRs, the atomic weight of each isotope, and the weight percent of the gadolinia isotopes. All of the inputs necessary for calculating the initial GDR material specification are included in Tables 5-1 and 5-2. In order to calculate the initial GDR material specifications, the molecular weights of Gd_2O_3 and UO_2 are first determined. Next, the weight fractions for uranium and gadolinium are calculated. From this point the isotopic wt% for the required isotopes of uranium and gadolinium are calculated. The wt% of the total oxygen in the GDR material can also be calculated.

Equation 5-3. U-234, U-236, U-238 Isotopic wt% For Initial GDR Material Specification

$$\mathrm{wt\%U236} = 0.0046 \cdot (\mathrm{wt\%U235})$$

$$\mathrm{wt\%U234} = 0.007731 \cdot (\mathrm{wt\%U235})^{1.0837}$$

$$\mathrm{wt\%U238} = 100 - (\mathrm{wt\%U234} + \mathrm{wt\%U235} + \mathrm{wt\%U236})$$

Equation 5-4. Molecular Weight of Uranium For Initial GDR Material Specification

$$(1/\mathrm{mw})^{\mathrm{uranium}} = 0.01 \cdot \sum_i \left(\frac{\mathrm{wt\%}}{\mathrm{atwt}}\right)_i$$

where: the term on the left-hand side of the equation is the reciprocal of the molecular weight for uranium; *i* is the subscript indicating each isotope of the element; *wt%* and *atwt* are weight percentage and atomic weight of the i^{th} uranium isotopes, respectively.

Equation 5-5. Molecular Weight of Gadolinium For Initial GDR Material Specification

$$(1/\mathrm{mw})^{\mathrm{gadolinium}} = 0.01 \cdot \sum_i \left(\frac{\mathrm{wt\%}}{\mathrm{atwt}}\right)_i$$

where: the term on the left-hand side of the equation is the reciprocal of the molecular weight for gadolinium; *i* is the subscript indicating each isotope of the element; *wt%* and *atwt* are weight percent and atomic weight of the i^{th} gadolinium isotopes, respectively.

Equation 5-6. Molecular Weight of UO_2 or GD_2O_3 for Initial GDR Material Specification

$$(mw)_X = N \cdot (mw)_Y + M \cdot atwt_{oxygen}$$

where: the term on the left-hand side of the equation is the molecular weight for UO_2 or GD_2O_3; *X* subscript denotes UO_2 or GD_2O_3 term and *Y* denotes uranium or gadolinium term (calculated with Equations 5-3 and 5-4); *N*=1 for 1 atom of uranium in UO_2 and *N*=2 for 2 atoms of gadolinia in Gd_2O_3; *M*=2 for 2 atoms of oxygen in UO_2 and *M*=3 for 3 atoms of oxygen in Gd_2O_3.

Equation 5-7. Weight Fraction of Uranium, Gadolinium, Or Oxygen in UO_2 or Gd_2O_3, For Initial GDR Material Specification

$$(wfrac)_J = N \cdot (\frac{mw_J}{mw_K})$$

$$(wfrac)_{oxygen} = 1 - (wfrac_J)$$

where: the term on the left-hand side of the equation is weight fraction for oxygen, uranium, or gadolinium; *J* subscript denotes uranium or gadolinium term, *K* denotes UO_2 or Gd_2O_3 term; *N*=1 for 1 atom of uranium per UO_2 molecule and *N*=2 for 2 atoms of gadolinia per Gd_2O_3 molecule.

Equation 5-8. Gadolinium Isotopic wt% Used In Initial GDR Material Specification

$$(wt\%Gd)_i^{GDR} = enrich^{GDR} \cdot (wt\%Gd)_i \cdot (wfrac)_{gadolinium}$$

where: the term on the left-hand side of the equation is *wt%* for the i^{th} gadolinium isotope to be used for the initial material specification of the GDR; the *enrich* term is the fractional enrichment of gadolinia in the GDR; *i* subscript is for the i^{th} gadolinium isotope; *wt%Gd* is the weight percentage for the i^{th} gadolinium isotopes; *wfrac* term is the calculated gadolinium weight fraction from Equation 5-7.

Equation 5-9. Uranium Isotopic wt% Used In Initial GDR Material Specification

$$(wt\%U)_i^{GDR} = (1 - enrich^{GDR}) \cdot (wt\%U)_i \cdot (wfrac)_{uranium}$$

where: the term on the left-hand side of the equation is *wt%* for the i^{th} uranium isotope to be used for the initial material specification for the GDR; the *enrich* term is the fractional enrichment of gadolinia in the GDR material; *i* subscript is for the i^{th} uranium isotope; *wt%U* is the wt% for the i^{th} uranium isotope calculated with Equation 5-3; *wfrac* term is the calculated uranium weight fraction from Equation 5-7.

Equation 5-10. Oxygen wt% Used In Initial GDR Material Specification

$$(\text{wt\%O})^{\text{GDR}} = 100 - \sum_{\text{i}} \left[(\text{wt\%U})_{\text{i}} + (\text{wt\%Gd})_{\text{i}} \right]$$

where: the term on the left-hand side of the equation is wt% for the oxygen to be used for the initial material specification for the GDR; *i* subscript is for the i^{th} uranium or gadolinium isotope; *wt%U* is the wt% for the i^{th} uranium isotope and ; *wt%Gd* is the wt% for the i^{th} gadolinium isotope calculated with Equations 5-8 and 5-9.

Table 5-2. Atomic Weights and Weight Percent for Selected Isotopes Used in GDR Material Specification (Reference 7.7, pp. 20, 36, 46)

Element/Isotope	Atomic Weight	Weight Percent (%)
Oxygen	15.9994	--
U-234	234.0409	--
U-235	235.0439	--
U-236	236.0456	--
U-238	238.0508	--
Gd-152	151.9198	0.19
Gd-154	153.9209	2.13
Gd-155	154.9226	14.6
Gd-156	155.9221	20.30
Gd-157	156.9240	15.62
Gd-158	157.9241	24.95
Gd-160	159.9270	22.23

5.2.2. CONTROL BLADES

In the Quad Cities Unit 2 depletion calculations, BWR cruciform control blade histories are tracked and included in the modeling of bladed assemblies. The blade insertion history information is obtained from Reference 7.3, p. 583 and is provided in Attachment VII. The control blade dimensions and composition are obtained from Reference 7.5, pp. A-9 and C-15. The BWR cruciform control blade is illustrated in Figure 5-1. The portions of the control blade that are included in the model for each assembly are shown in Figure 5-2. In the SAS2H path B model (described in Section 5.4) the outer ring of the assembly cell model contains bypass water (density = 0.7396 g/cm^3) for non-bladed nodes and a homogenization of bypass water, stainless steel 304, and B_4C (density = 1.6704 g/cm^3) for bladed nodes. Inputs required for determination of the homogenized control blade composition are included in Table 5-3 and 5-4.

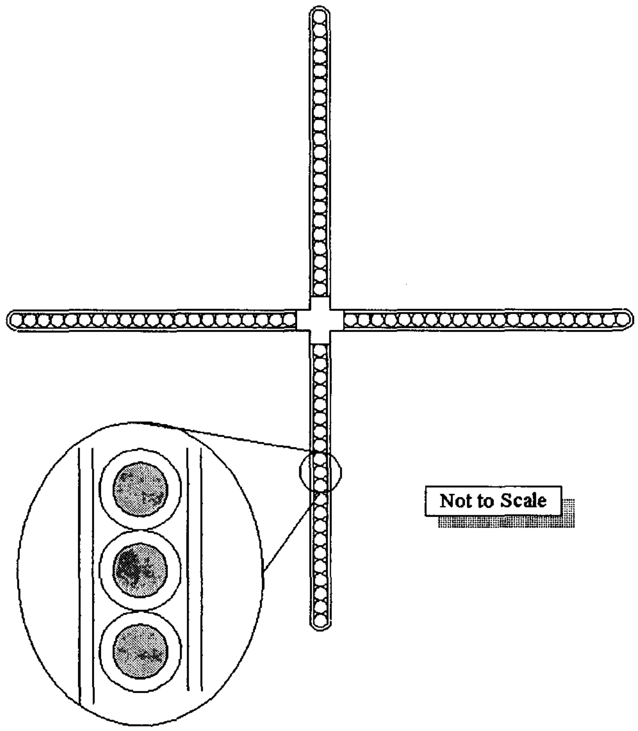

Figure 5-1. Quad Cities Unit 2 Cruciform Control Blade

Table 5-3. Inputs for Quad Cities Unit 2 Control Blade Homogenization

	SS304	Water	B_4C
Volume Fraction	0.1222	0.8289	0.0489
Density (g/cm^3)	7.94	0.7396	1.78 (70.6% theoretical)
Weight Fraction	0.5809	0.3670	0.0521
Homogenized Density	1.6704 (g/cm^3)		

Table 5-4. Composition Weight Percentages for Quad Cities Unit 2 Control Blade Homogenization (Reference 7.8, p. 20, p.85)

Element	SS304 (wt%)	Water (wt%)	B_4C (wt%)	Homogenized
Carbon	0.080		21.739	1.1791
Nitrogen	0.100			0.0581
Silicon	0.750			0.4357
Phosphorus	0.045			0.0261
Sulfur	0.030			0.0174
Chromium	19.00			11.037
Manganese	2.00			1.1618
Iron	68.745			39.935
Nickel	9.250			5.3734

Hydrogen		11.19		4.1070
Oxygen		88.81		32.595
Boron-10			14.424	0.7510
Boron-11			63.837	3.3259

When a node is required to be bladed during Quad Cities Unit 2 depletion calculations, the mixture description for the node contains the homogenized composition as described in Table 5-4. During the depletion calculation when there is no control blade present, the mixture description is set for bypass water. The control blade insertion history required for each bladed assembly is provided in terms of relative cycle, statepoint, and step numbers and is listed in Attachment VII.

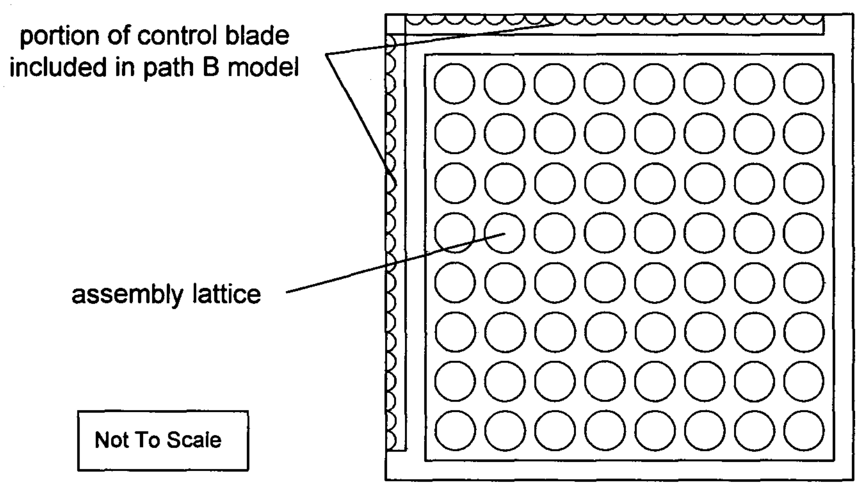

Figure 5-2. Assembly Cross Section with Control Blade Portions

Nodes with control blades present are modeled in SAS2H at full power for the depletion calculation. In an actual BWR reactor core the power in the vicinity of a control blade is significantly decreased. Thus, the neutron spectrum used in the modeled "bladed" nodes will be incorrect (see Section 3.5). This inaccuracy will result in a different distribution of depleted fuel isotopic concentrations. The effect of the modeling approximation has not been determined. Additionally, the effect of the control blade model (dimensions and composition) on the SAS2H calculation of depleted isotopics has not been determined. The appropriateness of a one dimensional "super-cell" model for BWR fuel assemblies with strong absorbers present (gadolinia-bearing rods, control blades) in a process for flux-weighting and zone-weighting cross sections to be used in fuel depletion calculations should be examined in more detail.

5.2.3. MODERATOR DENSITY PROFILES

Moderator densities are reported for each node of each assembly in Reference 7.3, p. 290. These density profiles are used in the SAS2H depletion calculations to help provide the correct neutron spectrum at different axial positions along the length of the assembly. A listing of the nodal moderator densities for each assembly is included in Attachment VI. Details of the utilization of this information in the Quad Cities Unit 2 depletion calculation are included in Section 5.4.

The reference water density used in this calculation is 0.7396 g/cm^3 (Reference 7.3, p. 5). This is the saturated water density for 1020 psia.

5.2.4. REACTOR OPERATING PRESSURE

Quad Cities Unit 2 is a General Electric, Type 3, BWR that operates at a pressure of 1020 psia (Reference 7.3, p. 5).

5.2.5. FUEL ASSEMBLY INSERTION HISTORY

The actual irradiation histories for the fuel assemblies from Quad Cities Unit 2, Cycles 9-14 were used to perform the SAS2H depletion calculations relevant to the CRC evaluations (Reference 7.3, pp. 16-32). Table 5-5 identifies the following information:

- the cycles in which the various fuel assemblies were inserted
- the locations of the various fuel assemblies by core map row and column in each cycle as shown in Figure 5-3
- the assembly fuel type

Figures 5-4 through 5-9 indicate schematically the location of the tracked assemblies in Cycles 9-14 of Quad Cities Unit 2 (Reference 7.3, pp. 20-25).

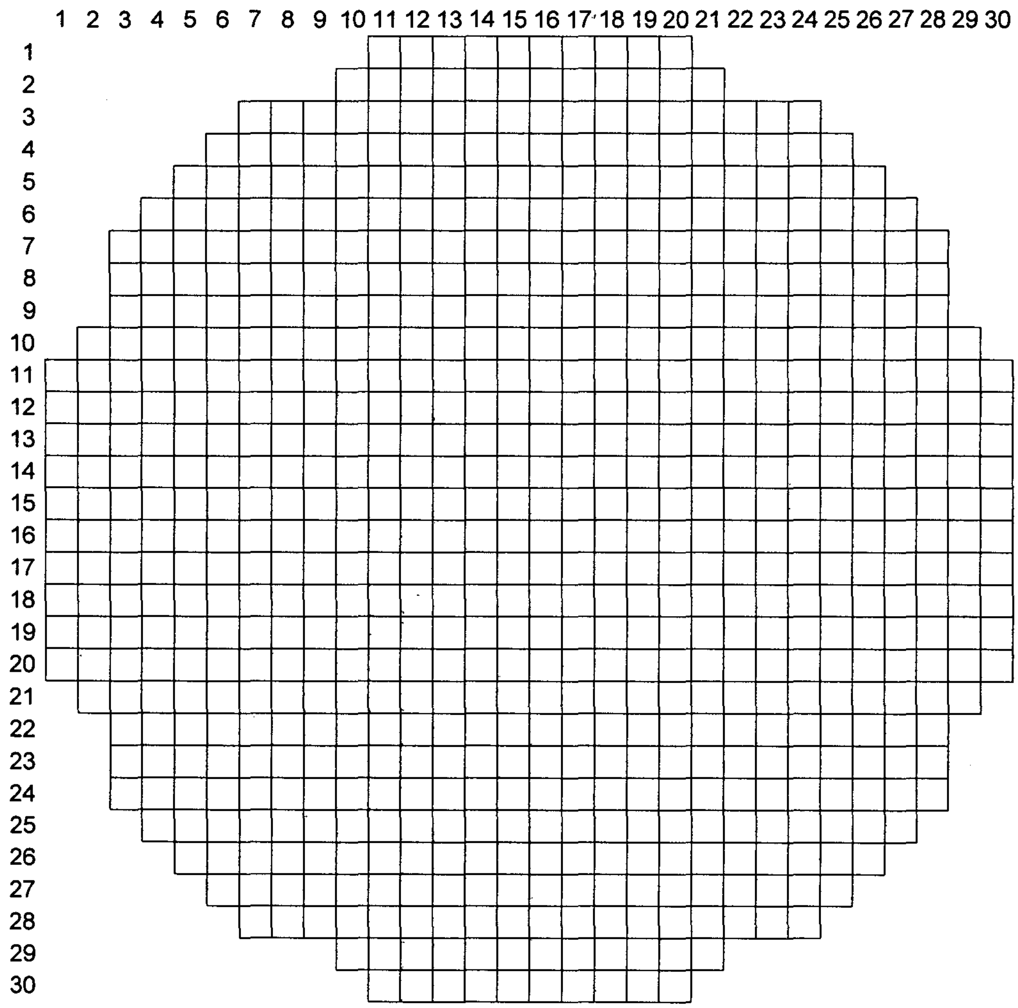

Figure 5-3. Quad Cities Unit 2 Core Map Assembly Locations by Row and Column

Table 5-5. Fuel Assembly Insertion Cycles for the Quad Cities Unit 2 Depletion Calculations (Reference 7.3, pp. 26-28)

Assembly Identifier / Fuel Type	Fuel Assembly Location by Core Map Row and Column					
	Cycle 9	Cycle 10	Cycle 11	Cycle 12	Cycle 13	Cycle 14
A1 / 6	19, 18	17, 27	20, 24	16, 29	25, 27	
A2 / 6	21, 25	16, 28	17, 17	19, 28	24, 28	
A3 / 6	24, 17	26, 22	16, 19	24, 26	19, 30	
A4 / 6	25, 18	25, 24	16, 24	18, 28	16, 30	
A5 / 6	26, 19	23, 25	19, 25	20,28	21, 29	
A6/ 6	17, 26	21, 18	18, 28	16, 28	22, 28	
A7 / 6	25, 21	28, 16	25, 25	16,16	25,25	
A8 / 6	25, 22	27, 18	19, 27	24, 25	21, 28	
B1 / 7	20, 17	20, 22	23, 19	29, 19	26,26	
B2 / 7	17, 22	16, 22	16,27	17, 29	17,30	
B3 / 7	21, 24	20, 24	22, 26	22, 27	18,30	
B4 / 7	22, 19	19, 27	17, 25	18, 29	20, 30	
B5/ 7	20, 21	26, 16	16, 23	24, 27	23, 28	
C1/ 9		22, 22	22, 22	23, 23	22, 22	
C2/ 9		20, 21	22, 20	24, 23	17, 23	
C3 / 9		19, 26	19, 21	16, 19	17, 21	
C4 / 9		17, 26	23, 24	16, 21	16, 16	28,24
C5 / 9		23, 21	18, 24	19, 23	20, 29	24, 24
C6 / 9		17, 22	17, 27	16, 23	17, 19	24, 28
C7 / 9		25, 22	22, 24	16, 20	19, 21	
C8 / 9		17, 24	27, 21	19, 19	17, 27	25, 27
C9 / 9		21, 25	26, 20	26, 17	19, 19	20, 30
C10 / 9		26, 17	24, 23	21, 16	18, 18	
C11 / 9		22, 23	24, 25	20, 24	22, 27	25, 26
C12 / 9		18, 23	16, 22	19, 26	24, 27	22, 28
C13 / 9		17, 20	17, 19	21, 22	25, 26	21, 29
C14 / 9		25, 21	20, 26	17, 26	20, 20	30, 20
D1 / 10		16, 25	16, 18	17, 22	24, 26	
D2 / 10		21, 24	18, 20	19, 22	17, 29	25, 25
D3 / 10		19, 22	17, 18	20, 23	19, 29	17, 30
D4 / 10		25, 20	23, 26	16, 25	18, 29	20, 29
D5 / 10		16, 17	21, 26	17, 24	23, 26	16, 30

Assembly Identifier / Fuel Type	Fuel Assembly Location by Core Map Row and Column					
	Cycle 9	Cycle 10	Cycle 11	Cycle 12	Cycle 13	Cycle 14
D6 / 10		25, 18	18, 27	16, 24	19, 28	19, 30
D7 / 10		25, 16	18, 16	22, 17	26, 24	26, 26
D8 / 10		17, 18	19, 24	18, 19	16, 29	18, 30
D9 / 10		19, 18	18, 26	19, 24	23, 27	21, 28
D10 / 10		16, 21	16, 26	19, 20	20, 28	23, 28
E1 / 14			18, 25	18, 25	20, 26	17, 29
E2 / 14			17,26	18,23	20, 23	16, 29
E3 / 14			16, 25	18, 27	18, 26	21, 27
E4 / 14			20, 25	17, 17	17, 25	24, 26
E5 / 14			19, 26	16, 17	16, 27	23, 26
E6 / 14			18, 19	16, 27	16, 26	22, 27
E7 / 14			22, 25	17, 27	23, 23	18, 18
E8 / 14			23, 25	21, 21	17, 17	18, 20
E9 / 14			21, 24	17, 28	22, 25	17, 23
E10 / 14			18, 23	20, 26	19, 26	19, 28
E11 / 14			25, 22	27, 17	24, 24	21, 21
E12 / 14			22, 23	22, 24	19, 24	18, 28
E13 / 14			25, 23	18, 18	21, 21	20, 18
E14 /14			19, 22	19, 27	22, 26	20, 23
F1 / 12			17, 20	23, 26	23, 24	17, 21
F2 / 12			16, 17	21, 25	19, 23	23, 27
F3 / 12			17, 24	20, 21	16, 18	18, 29
F4 / 12			25, 21	27, 20	18, 28	16, 16
F5 / 12			17, 22	18, 21	16, 24	19, 29
F6 / 12			18, 21	22, 23	16, 22	20, 28
F7 / 12			21, 25	20, 27	28, 18	17, 17
F8 / 12			20, 21	22, 26	16, 20	18, 22
F9 / 12			21, 22	17, 20	21, 23	24, 27
F10 / 12			16, 21	23, 25	19, 20	22, 23
G1 / 17				26, 21	20, 27	17, 26
G2 / 17				22, 20	17, 26	20, 20
G3/ 17				22, 25	17, 28	20, 24
G4 / 17				21, 23	21, 26	17, 19
G5/ 17				21, 26	27, 20	26, 17

Assembly Identifier / Fuel Type	Fuel Assembly Location by Core Map Row and Column					
	Cycle 9	Cycle 10	Cycle 11	Cycle 12	Cycle 13	Cycle 14
G6 / 17				20, 22	26, 17	19,19
G7 / 17				20, 25	23, 25	19, 21
G8 / 17				17, 21	18, 20	18, 24
G9 / 17				17, 23	16, 19	16, 18
G10 / 17				17, 25	18, 27	16, 19
G11 / 17				16, 18	24, 25	16, 27
H1 / 18				26, 16	27, 19	19, 24
H2 / 18				21, 24	22, 24	16, 24
H3 /18				19, 21	21, 25	16, 20
H4 / 18				19, 25	20, 24	22, 25
H5 / 18				18, 20	18, 24	19, 26
H6 / 18				18, 22	18, 22	20, 22
H7 / 18				18, 24	16, 23	16, 26
H8 / 18				18, 26	16, 28	16, 22
H9 / 18				17, 19	21, 27	19, 23
H10 / 18				16, 22	20, 22	17, 25
H11 / 18				16, 26	19, 27	24, 19
J1 / 16					20, 25	20, 26
J2 / 16					19, 25	23, 24
J3 / 16					18, 25	19, 20
J4 / 16					22, 23	17, 28
J5 / 16					18, 23	16, 23
J6 / 16					21, 22	18, 26
J7 / 16					17, 22	24, 25
J8 / 16					20, 21	16, 28
J9 / 16					21, 18	16, 21
J10 / 16					18, 21	21, 16
J11 / 16					16, 17	18, 27
J12 / 16					22, 19	22, 22
J13 / 16					19, 22	23, 23
J14 / 16					16, 25	21, 26
J15 / 16					21, 24	20, 27
J16 / 16					17, 18	23, 25
K1 / 15					17, 24	17, 27

Assembly Identifier / Fuel Type	Fuel Assembly Location by Core Map Row and Column					
	Cycle 9	Cycle 10	Cycle 11	Cycle 12	Cycle 13	Cycle 14
K2 /15					18, 19	22, 26
K3 / 15					16, 21	22, 24
K4 / 15					17, 20	19, 27
L1 / 1						19, 22
L2 / 1						21, 25
M1 / 2						16, 17
M2 / 2						16, 25
M3 / 2						17, 18
M4 / 2						17, 20
M5 / 2						17, 22
M6 / 2						17, 24
M7 / 2						18, 19
M8 / 2						18, 21
M9 / 2						18, 23
M10 / 2						18, 25
M11 / 2						19, 25
M12 / 2						20, 21
M13 / 2						20, 25
M14 / 2						21, 22
M15 / 2						21, 23
M16 / 2						21, 24

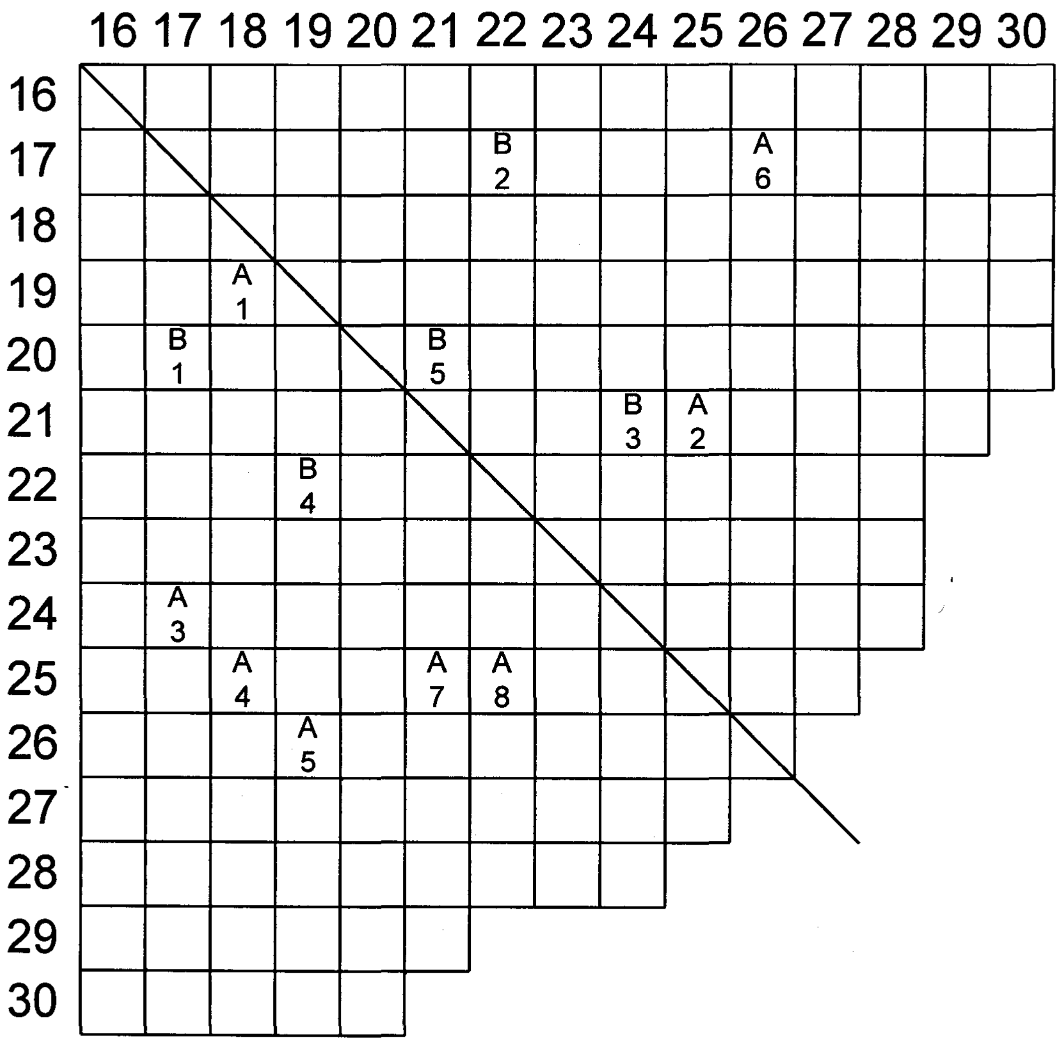

Figure 5-4. Fuel Assembly Locations Quad Cities Unit 2, Cycle 9

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
16		D 5				D 10	B 2			D 1			A 2		
17			D 8		C 13		C 6		C 8		C 4	A 1			
18								C 12							
19			D 9				D 3				C 3	B 4			
20						C 2	B 1		B 3						
21			A 6						D 2	C 9					
22							C 1	C 11							
23						C 5				A 5					
24															
25	D 7		D 6		D 4	C 14	C 7		A 4						
26	B 5	C 10					A 3								
27			A 8												
28	A 7														
29															
30															

Figure 5-5. Fuel Assembly Locations Quad Cities Unit 2, Cycle 10

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
16		F 2	D 1	A 3		F 10	C 12	B 5	A 4	E 3	D 10	B 2			
17		A 2	D 3	C 13	F 1		F 5		F 3	B 4	E 2	C 6			
18	D 7			E 6	D 2	F 6		E 10	C 5	E 1	D 9	D 6	A 6		
19						C 3	E 14		D 8	A 5	E 5	A 8			
20						F 8			A 1	E 4	C 14				
21							F 9		E 9	F 7	D 5				
22					C 2		C 1	E 12	C 7	E 7	B 3				
23				B 1					C 4	E 8	D 4				
24								C 10		C 11					
25						F 4	E 11	E 13		A 7					
26					C 9										
27						C 8									
28															
29															
30															

Figure 5-6. Fuel Assembly Locations Quad Cities Unit 2, Cycle 11

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
16	A 7	E 5	G 11	C 3	C 7	C 4	H 10	C 6	D 6	D 4	H 11	E 6	A 6	A 1	
17		E 4		H 9	F 9	G 8	D 1	G 9	D 5	G 10	C 14	E 7	E 9	B 2	
18			E 13	D 8	H 5	F 5	H 6	E 2	H 7	E 1	H 8	E 3	A 4	B 4	
19				C 8	D 10	H 3	D 2	C 5	D 9	H 4	C 12	E 14	A 2		
20						F 3	G 6	D 3	C 11	G 7	E 10	F 7	A 5		
21	C 10					E 8	C 13	G 4	H 2	F 2	G 5				
22		D 7			G 2			F 6	E 12	G 3	F 8	B 3			
23								C 1		F 10	F 1				
24								C 2		A 8	A 3	B 5			
25															
26	H 1	C 9				G 1									
27		E 11			F 4										
28															
29				B 1											
30															

Figure 5-7. Fuel Assembly Locations Quad Cities Unit 2, Cycle 12

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
16	C 4	J 11	F 3	G 9	F 8	K 3	F 6	H 7	F 5	J 14	E 6	E 5	H 8	D 8	A 4
17		E 8	J 16	C 6	K 4	C 3	J 7	C 2	K 1	E 4	G 2	C 8	G 3	D 2	B 2
18			C 10	K 2	G 8	J 10	H 6	J 5	H 5	J 3	E 3	G 10	F 4	D 4	B 3
19				C 9	F 10	C 7	J 13	F 2	E 12	J 2	E 10	H 11	D 6	D 3	A 3
20					C 14	J 8	H 10	E 2	H 4	J 1	E 1	G 1	D 10	C 5	B 4
21			J 9			E 13	J 6	F 9	J 15	H 3	G 4	H 9	A 8	A 5	
22				J 12			C 1	J 4	H 2	E 9	E 14	C 11	A 6		
23								E 7	F 1	G 7	D 5	D 9	B 5		
24									E 11	G 11	D 1	C 12	A 2		
25										A 7	C 13	A 1			
26		G 6									B 1				
27				H 1	G 5										
28			F 7												
29															
30															

Figure 5-8. Fuel Assembly Locations Quad Cities Unit 2, Cycle 13

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
16	F4	M1	G9	G10	H3	J9	H8	J5	H2	M2	H7	G11	J8	E2	D5
17		F7	M3	G4	M4	F1	M5	E9	M6	H10	G1	K1	J4	E1	D3
18			E7	M7	E8	M8	F8	M9	G8	M10	J6	J11	E12	F3	D8
19				G6	J3	G7	L1	H9	H1	M11	H5	K4	E10	F5	D6
20			E13		G2	M12	H6	E14	G3	M13	J1	J15	F6	D4	C9
21	J10					E11	M14	M15	M16	L2	J14	E3	D9	C13	
22							J12	F10	K3	H4	K2	E6	C12		
23								J13	J2	J16	E5	F2	D10		
24				H11					C5	J7	E4	F9	C6		
25										D2	C11	C8			
26		G5									D7				
27															
28									C4						
29															
30					C14										

Figure 5-9. Fuel Assembly Locations Quad Cities Unit 2, Cycle 14

5.2.6. REACTOR CYCLE OPERATION HISTORY

This section contains the Quad Cities Unit 2 reactor cycle summary information relevant to the CRC evaluations documented in this calculation. The calendar day duration between reactor startup and shutdown along with end-of-cycle downtimes are included in Table 5-6. Statepoint and datapoint summary information is included in Table 5-7. Table 5-8 provides the EFPD burnup steps that are used in the SAS2H depletion calculation for each cycle. The statepoints refer to times when the reactor was previously operated over a time duration that is suitable for a reactivity calculation. MCNP reactivity calculations for the CRC evaluations will be performed using the reactor startup conditions and appropriate depleted and decayed isotopics after each statepoint shutdown. The datapoints refer to times when the depletion calculations were halted to adjust various input parameters such as average fuel temperatures and in-channel moderator densities. The depletion calculations were continued after each datapoint halt without modeling any reactor downtime.

Table 5-6. Cycle Summary Information for Quad Cities Unit 2 Depletion Calculations (Reference 7.3, p. 16)

Cycle	Startup Date	Shutdown Date	Cycle Length (EFPD)	Downtime at EOC[1] (days)
9	2/1/87	4/10/88	348.44	75
10	6/25/88	2/3/90	467.48	95
11	5/8/90	12/20/91	484.2	138
12	5/8/92	3/6/93	263.7	85
13	6/1/93	3/4/95	387.07	136
14	7/20/95	Not Avail.	211.09[2]	Not Avail.

[1] End of Cycle
[2] This EFPD value is not for EOC but occurs during the middle of Cycle 14.

Table 5-7. Statepoint and Datapoint Summary Information for Quad Cities Unit 2 Depletion Calculations (Reference 7.3, p. 18)

Cycle	EFPD	Statepoint or Datapoint Identifier	Statepoint Downtime (days)
9	0.0	DP1[1]	--
9	167.47	DP2	--
9	348.44		--
10	0.0	DP3	--
10	222.47	DP4	--
10	467.48		--
11	0.0	DP5	--
11	180.3	DP6	--
11	484.2		--
12	0.0	DP7	--
12	142.2	DP8	--
12	263.7		--
13	0.0	SP9[2]	--
13	10.1	SP10	16.7
13	112.94	SP11	16.7
13	224.4	DP12[3]	13.1
13	324.73	SP13	16.7
13	387.07		--

Cycle	EFPD	Statepoint or Datapoint Identifier	Statepoint Downtime (days)
14	0.0	SP14	--
14	93.56	DP15[4]	Not Avail.
14	211.09	SP16	16.7

[1] The letters "DP" refer to a CRC datapoint. The number immediately following the "DP" refers to the relative datapoint for the Quad Cities Unit 2 CRC evaluations.
[2] The letters "SP" refer to a CRC statepoint. The number immediately following the "SP" refers to the relative statepoint for the Quad Cities Unit 2 CRC evaluations in the BWR CRC evaluation program.
[3] The critical experiment occurred at a different exposure than what is represented at this statepoint. Isotopic compositions calculated at this depletion point will not be used in a reactivity calculation.
[4] Critical experiment data is not available at this statepoint. Isotopic compositions calculated at this depletion point will not be used in a reactivity calculation.

Table 5-8. EFPD Burnup Steps for Quad Cities Unit 2 Depletion Calculations

Cycle	Total EFPD for Cycle	EFPD Burnup Steps[1]
9A[2]	--	55.83
9A	--	55.83
9A	--	55.83
9B	--	60.3
9B	--	60.3
9B	348.44	60.3
10A	--	55.63
10A	--	55.63
10A	--	55.63
10A	--	55.63
10B	--	61.25
10B	--	61.25
10B	--	61.25
10B	467.48	61.25
11A	--	60.1
11A	--	60.1
11A	--	60.1
11B	--	60.78
11B	--	60.78
11B	--	60.78
11B	--	60.78
11B	484.2	60.78
12A	--	47.4
12A	--	47.4
12A	--	47.4
12B	--	60.75
12B	263.7	60.75
13A	--	10.1
13B	--	51.42
13B	--	51.42
13C	--	55.73
13C	--	55.73
13D	--	50.17

Cycle	Total EFPD for Cycle	EFPD Burnup Steps[1]
13D	387.07	50.17
13E	--	62.34
14A	--	43.28
14A	--	43.28
14B	--	58.77
14B	211.09	58.77

[1] The total EFPD for each cycle is divided into equal EFPD steps less than 70 EFPD. The 70 EFPD limit for each burn step is required for the SAS2H depletion calculation.
[2] The letters with each cycle number designate the individual SAS2H depletion steps that are required in an individual calculation for part of a cycle, i.e., 12A repeated three times signifies one SAS2H calculation with three 47.4 EFPD burnup steps for the first half of the Cycle 12 depletion.

5.2.7. BURNUP, FUEL TEMPERATURE, AND MODERATOR DENSITY

Burnup, fuel temperature, and moderator density data were required for each node of each assembly in each SAS2H depletion calculation. A set of nodal burnup data at the beginning and end of each SAS2H depletion calculation was required. A set of nodal fuel temperature and moderator density data representative of full-power operation during each depletion calculation of interest (between statepoints and/or datapoints) was required. The tabulation of this data for Quad Cities Unit 2 is included in Attachment VI. Attachment VI contains the burnup, fuel temperature, and moderator density data required to perform all depletion calculations for each of the fuel assemblies present in the Quad Cities Unit 2 CRC evaluations. The bottom of node 1 begins at the bottom of the active fuel region in each assembly. The burnup data is presented in units of gigawatt-days per metric ton of uranium (GWd/MTU). The fuel temperature data is presented in units of degrees Kelvin. The moderator density data is presented in units of grams per cubic centimeter. Each set of fuel temperature and moderator density data listed in Attachment VI was applicable to the depletion calculation performed between the statepoints and/or datapoints identified above the particular data.

5.3. ASSEMBLY DEPLETION CALCULATION PROCEDURE

The procedure for performing the fuel assembly SAS2H depletion calculations documented in this analysis was based on the utilization of the SPACE, Version 01 and CRAFT, Version 4C software routines. The SPACE-CRAFT software routines used for SAS2H automation are described generally in Section 5.5. The complete detailed description of the SPACE-CRAFT software routines is provided in Attachment I and Attachment II. An overview of the operation of the software routines and script files is shown in Figure 5-10. The procedure for performing a fuel assembly depletion calculation with SPACE-CRAFT, consisted of the following steps:

- Create a SPACE input deck for the assembly depletion calculation.
- Assure that the executable files for SPACE, CRAFT, LIFTOFF, BATCH43 and SEDEXECUTE, and the SPACE input deck entitled "spacein", are in the same directory. The LIFTOFF executable file is a script that is executed by SPACE, which directs CRAFT through all of the SAS2H nodal depletion calculations for each designated assembly. The BATCH43 executable

file is a script that directs the execution of SAS2H for each depletion calculation. The SEDEXECUTE executable file is a script file, which is used in conjunction with the CRAFT code to create the consolidated output files described in Section 5.5.

- Execute SPACE. SPACE generates the required CRAFT input decks for the assembly depletion calculation and then calls for the LIFTOFF script to execute CRAFT in the proper sequence. CRAFT then creates and executes the required SAS2H input files for the assembly depletion calculation.
- Check and analyze the CRAFT generated SAS2H input decks and the SAS2H isotopic results.

The various CRAFT generated and consolidated SAS2H output files contain unique filenames, which specify the following information:

- reactor identifier
- assembly insertion number in current reactor cycle
- axial node number
- reactor cycle number in which the SAS2H calculation begins
- EFPD statepoint at which the SAS2H calculation begins
- reactor cycle number in which the SAS2H calculation ends
- EFPD statepoint at which the SAS2H calculation ends.

A complete detailed description of the CRAFT filename content and format is provided in Attachment I.

Transfer of depleted fuel and depleted gadolinia isotopic concentrations between subsequent nodal SAS2H calculations is processed by the CRAFT Version 4C software routine automation. It is suspected that minor variations in isotopic concentrations occur during this process. The magnitude and impact of these variations has not been determined.

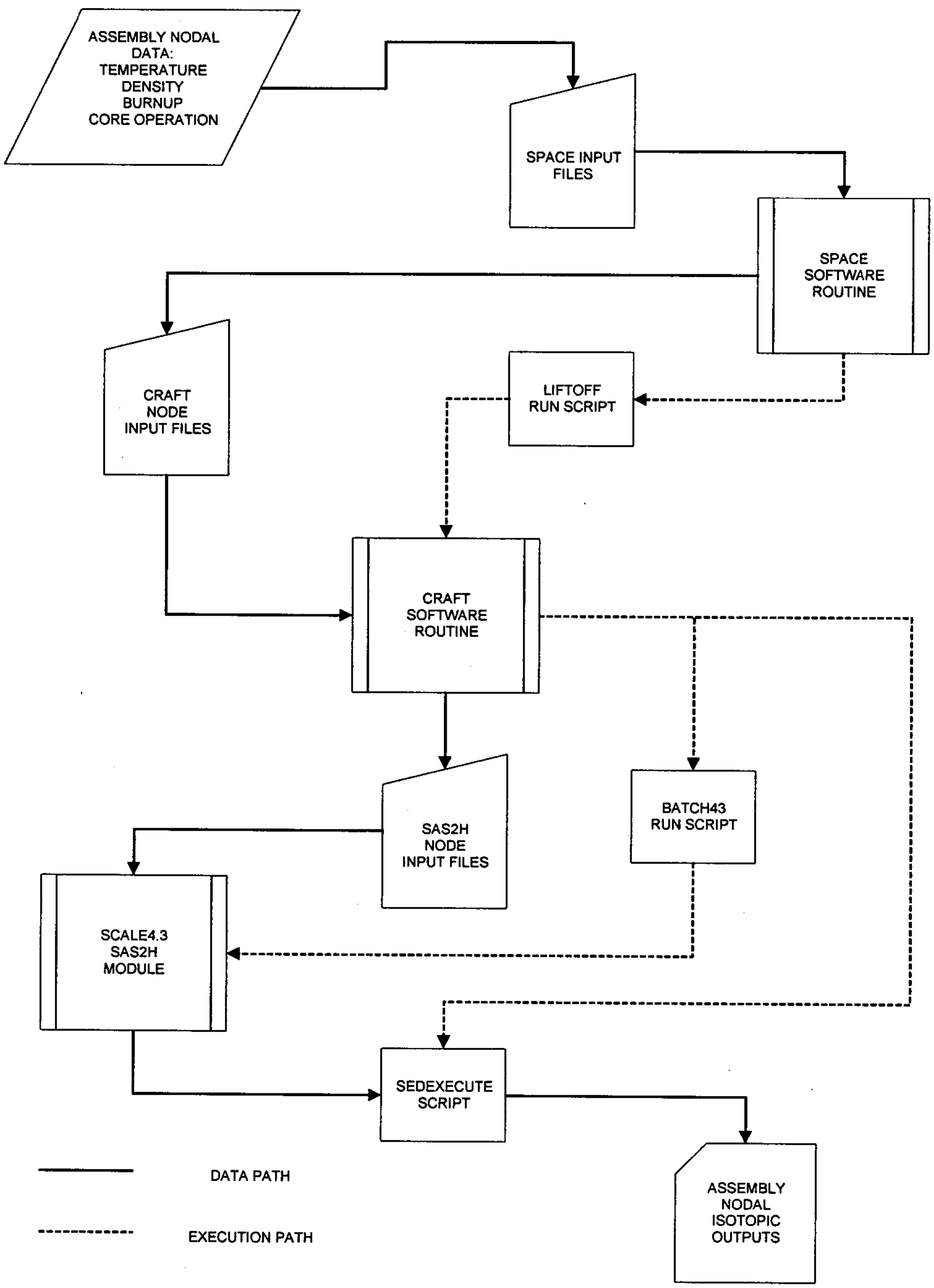

Figure 5-10. Software Routine and Script Flow Chart

5.4. PATH B MODEL DEVELOPMENT

The SAS2H control module used ORIGEN-S to perform a point depletion calculation for the fuel assembly or section of the fuel assembly described in the SAS2H input deck. The ORIGEN-S calculational module used cell-weighted cross sections based on one-dimensional (1-D) transport calculations performed by XSDRNPM. One-dimensional transport calculations were performed on two models, Path A and Path B, to calculate energy-dependent spatial neutron flux distributions necessary to perform cross section cell-weighting calculations.

The Path A model was simply a unit cell of the fuel assembly lattice containing a fuel rod. In the Path A model, the fuel pellet and clad were modeled explicitly. The only modification required to develop the Path A model was the conversion of the fuel assembly's square lattice unit cell perimeter to a radial perimeter conserving moderator volume within the unit cell (exterior to the fuel rod cladding). This modification was performed automatically by the SAS2H control module. A 1-D transport calculation was performed on the Path A model for each energy group, and the spatial flux distributions for each energy group were used to calculate cell-weighted cross sections for the fuel, clad, and moderator.

The Path B model was a larger representation of the assembly than the Path A model. The Path B model approximated spectral effects due to heterogeneity within the fuel assembly such as water gaps, GDRs, or control blades (CRBs) if CRBs were to be tracked in the calculation.

The basic structure of the Path B model for the fuel assembly depletion calculations performed in this analysis included an inner region composed of a representation of the non-fuel (GDR or water rod) assembly lattice cell. A region containing the homogenization of the Path A model surrounded the inner region in the Path B model. The next region, representing the Zirc4 assembly channel, surrounded the homogenized region in the Path B model. The final region of the Path B model represents the bypass moderator and control blade homogenization found in the channel-to-channel spacing. The size of each radial region in the Path B model was determined by conservation of the fuel-to-moderator mass in the assembly (Reference 7.1, Section S2.2.3.1 of Volume 1, Rev. 5). The cell-weighted cross sections from the Path A model were applied to the homogenized region during the Path B model transport calculations. New cell-weighted cross sections for each energy group were then developed using the unit cell spatial flux distribution results from the Path B model transport calculations. These cell-weighted cross sections were ultimately used in point depletion calculations performed by ORIGEN-S to calculate both the depleted fuel and the depleted integral burnable poison (if present) isotopic compositions in the corresponding section of the fuel assembly. A detailed description of the calculations used to produce time-dependent cross sections by SAS2H is documented in Section S2.2.4 of Volume 1, Rev. 5 in Reference 7.1.

The following list of approximations used in the depletion and decay neutronics model provides some additional details regarding assumptions previously stated in Section 3:

1. Moderator density obtained from moderator density profiles is assumed to be uniform in all fuel rod or GDR unit cells in the node.

2. Fuel pellet or GDR pellet density is assumed to be uniform in all rods of the assembly. Fuel pellet and GDR pellet diameters and densities are assumed to be the same.
3. The mass of the integral burnable absorber Gd_2O_3 in the node (approximately 40 grams) is not accounted for in the nodal fuel mass specified for the SAS2H calculation. It is assumed that such a small mass of additional fuel (UO_2) would have little effect when compared to the typical 7.0 kilograms of fuel mass in a node.
4. The 10 node collapsed format was used in the Quad Cities Unit 2 depletion calculations. The effects of averaging nodal burnup, temperature, density, and enrichment parameters have not been examined in detail. Further investigation and confirmatory calculations may be performed to quantify these effects on isotopic depletions.
5. To protect the proprietary nature of the radial enrichment distributions in nodes of assemblies used in Cycles 9-14 of Quad Cities Unit 2, average nodal enrichments were used in all depletion calculations. The effects of averaging the radial enrichment distribution in a node have not been examined in detail. Further investigation and confirmatory calculations may be performed to quantify these effects on isotopic depletions.
6. The mass of uranium in UO_2 was calculated using a mass fraction without dependence on U-235 enrichment. It was also assumed that all fuel pellets in each assembly have the same diameter. Approximations of this type for fuel mass are not expected to adversely affect this calculation.

Additional discussion concerning approximations used in this calculation are listed later in this section. The above approximations and limitations inherent to the Path B model used in the SAS2H calculation can be confirmed by additional sensitivity analyses.

The Path B model development calculations for the Quad Cities Unit 2 depletion calculations are presented in Tables 5-9 through 5-18 and contain the following information:

- the fuel assembly section characteristics for which the Path B model is developed
- the required Path B model development input parameters
- matrix indicating Path B models used in specific assemblies
- Path B model dimensions for GDR models

Enrichments and the number of GDRs vary axially in BWR assemblies. The number of GDRs in an axial node determines the type Path B model that should used. Figures 5-11 through 5-22 show the 10 node axial format of each type fuel assembly used in the depletion calculation. The figures indicate which of the actual 24 nodes are collapsed into the 10 node format. Additionally, a description of the U-235 wt% enrichment in UO_2, the number of GDRs, and the GDR enrichment in Gd_2O_3 is presented for each node. The node heights used in the SAS2H calculation are also provided. Nodes that are represented by a Path B model are indicated in Table 5-19.

COLLAPSED NODES USED IN SAS2H	24 ACTUAL NODES	NODE DESCRIPTION	NODE HEIGHT
AXIAL NODE 10	24	URANIUM	15.24CM
AXIAL NODE 9	23, 22, 21, 20	3.19 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	60.96CM
AXIAL NODE 8	19, 18, 17	3.19 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 7	16, 15, 14	3.19 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 6	13, 12, 11	3.19 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 5	10, 9	3.19 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 4	8, 7, 6	3.19 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 3	5, 4	3.19 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 2	3, 2	3.19 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 1	1	URANIUM	15.24CM

Figure 5-11. Quad Cities Unit 2 Axial Loading - Assembly A

COLLAPSED NODES USED IN SAS2H	24 ACTUAL NODES	NODE DESCRIPTION	NODE HEIGHT
AXIAL NODE 10	24	URANIUM	15.24CM
AXIAL NODE 9	23, 22, 21, 20	3.19 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	60.96CM
AXIAL NODE 8	19, 18, 17	3.19 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	45.72CM
AXIAL NODE 7	16, 15, 14	3.19 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	45.72CM
AXIAL NODE 6	13, 12, 11	3.19 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	45.72CM
AXIAL NODE 5	10, 9	3.19 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	30.48CM
AXIAL NODE 4	8, 7, 6	3.19 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	45.72CM
AXIAL NODE 3	5, 4	3.19 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	30.48CM
AXIAL NODE 2	3, 2	3.19 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	30.48CM
AXIAL NODE 1	1	URANIUM	15.24CM

Figure 5-12. Quad Cities Unit 2 Axial Loading - Assembly B

COLLAPSED NODES USED IN SAS2H	24 ACTUAL NODES	NODE DESCRIPTION	NODE HEIGHT
AXIAL NODE 10	24	URANIUM	15.24CM
AXIAL NODE 9	23, 22, 21, 20	3.20 WT% UO_2 / 9 - 3.0 WT% Gd_2O_3	60.96CM
AXIAL NODE 8	19, 18, 17	3.20 WT% UO_2 / 9 - 3.0 WT% Gd_2O_3	45.72CM
AXIAL NODE 7	16, 15, 14	3.20 WT% UO_2 / 9 - 3.0 WT% Gd_2O_3	45.72CM
AXIAL NODE 6	13, 12, 11	3.20 WT% UO_2 / 9 - 3.0 WT% Gd_2O_3	45.72CM
AXIAL NODE 5	10, 9	3.20 WT% UO_2 / 9 - 3.0 WT% Gd_2O_3	30.48CM
AXIAL NODE 4	8, 7, 6	3.20 WT% UO_2 / 9 - 3.0 WT% Gd_2O_3	45.72CM
AXIAL NODE 3	5, 4	3.20 WT% UO_2 / 9 - 3.0 WT% Gd_2O_3	30.48CM
AXIAL NODE 2	3, 2	3.20 WT% UO_2 / 9 - 3.0 WT% Gd_2O_3	30.48CM
AXIAL NODE 1	1	URANIUM	15.24CM

Figure 5-13. Quad Cities Unit 2 Axial Loading - Assembly C

COLLAPSED NODES USED IN SAS2H	24 ACTUAL NODES	NODE DESCRIPTION	NODE HEIGHT
AXIAL NODE 10	24	URANIUM	15.24CM
AXIAL NODE 9	23, 22, 21, 20	3.37 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	60.96CM
AXIAL NODE 8	19, 18, 17	3.37 WT% UO_2 / 7 - 40 WT% GD_2O_3	45.72CM
AXIAL NODE 7	16, 15, 14	3.37 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	45.72CM
AXIAL NODE 6	13, 12, 11	3.37 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	45.72CM
AXIAL NODE 5	10, 9	3.37 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	30.48CM
AXIAL NODE 4	8, 7, 6	3.37 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	45.72CM
AXIAL NODE 3	5, 4	3.37 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	30.48CM
AXIAL NODE 2	3, 2	3.37 WT% UO_2 / 7 - 4.0 WT% GD_2O_3	30.48CM
AXIAL NODE 1	1	URANIUM	15.24CM

Figure 5-14. Quad Cities Unit 2 Axial Loading - Assembly D

COLLAPSED NODES USED IN SAS2H	24 ACTUAL NODES	NODE DESCRIPTION	NODE HEIGHT
AXIAL NODE 10	24, 23	URANIUM	30.48CM
AXIAL NODE 9	22, 21, 20	3.34 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 8	19, 18, 17	3.50 WT% UO_2 / 9 - 3.22 WT% GD_2O_3	45.72CM
AXIAL NODE 7	16, 15, 14	3.50 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 6	13, 12, 11	3.50 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 5	10, 9	3.50 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 4	8, 7, 6	3.34 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 3	5, 4	3.34 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 2	3, 2	3.34 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 1	1	URANIUM	15.24CM

Figure 5-15. Quad Cities Unit 2 Axial Loading - Assembly E

COLLAPSED NODES USED IN SAS2H	24 ACTUAL NODES	NODE DESCRIPTION	NODE HEIGHT
AXIAL NODE 10	24, 23	URANIUM	30.48CM
AXIAL NODE 9	22, 21, 20	3.22 WT% UO_2 / 9 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 8	19, 18, 17	3.37 WT% UO_2 / 11 - 3.18 WT% GD_2O_3	45.72CM
AXIAL NODE 7	16, 15, 14	3.37 WT% UO_2 / 9 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 6	13, 12, 11	3.37 WT% UO_2 / 9 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 5	10, 9	3.37 WT% UO_2 / 9 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 4	8, 7, 6	3.22 WT% UO_2 / 9 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 3	5, 4	3.22 WT% UO_2 / 9 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 2	3, 2	3.22 WT% UO_2 / 9 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 1	1	URANIUM	15.24CM

Figure 5-16. Quad Cities Unit 2 Axial Loading - Assembly F

COLLAPSED NODES USED IN SAS2H	24 ACTUAL NODES	NODE DESCRIPTION	NODE HEIGHT
AXIAL NODE 10	24, 23	URANIUM	30.48CM
AXIAL NODE 9	22, 21, 20	3.06 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 8	19, 18, 17	3.24 WT% UO_2 / 9 - 3.22 WT% GD_2O_3	45.72CM
AXIAL NODE 7	16, 15, 14	3.24 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 6	13, 12, 11	3.24 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 5	10, 9	3.24 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 4	8, 7, 6	3.06 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 3	5, 4	3.06 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 2	3, 2	3.06 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 1	1	URANIUM	15.24CM

Figure 5-17. Quad Cities Unit 2 Axial Loading - Assembly G

COLLAPSED NODES USED IN SAS2H	24 ACTUAL NODES	NODE DESCRIPTION	NODE HEIGHT
AXIAL NODE 10	24, 23	URANIUM	30.48CM
AXIAL NODE 9	22, 21, 20	3.06 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 8	19, 18, 17	3.24 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 7	16, 15, 14	3.24 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 6	13, 12, 11	3.24 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 5	10, 9	3.24 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 4	8, 7, 6	3.06 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 3	5, 4	3.06 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 2	3, 2	3.06 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 1	1	URANIUM	15.24CM

Figure 5-18. Quad Cities Unit 2 Axial Loading - Assembly H

COLLAPSED NODES USED IN SAS2H	24 ACTUAL NODES	NODE DESCRIPTION	NODE HEIGHT
AXIAL NODE 10	24, 23	URANIUM	30.48CM
AXIAL NODE 9	22, 21, 20	3.32 WT% UO_2 / 10- 3.8 WT% GD_2O_3	45.72CM
AXIAL NODE 8	19, 18, 17	3.48 WT% UO_2 / 10 - 3.8WT% GD_2O_3	45.72CM
AXIAL NODE 7	16, 15, 14	3.48WT% UO_2 / 10 - 3.8 WT% GD_2O_3	45.72CM
AXIAL NODE 6	13, 12, 11	3.48 WT% UO_2 / 10 - 3.8 WT% GD_2O_3	45.72CM
AXIAL NODE 5	10, 9	3.48 WT% UO_2 / 10 - 3.8 WT% GD_2O_3	30.48CM
AXIAL NODE 4	8, 7, 6	3.32 WT% UO_2 / 10 - 3.8 WT% GD_2O_3	45.72CM
AXIAL NODE 3	5, 4	3.32 WT% UO_2 / 10 - 3.8 WT% GD_2O_3	30.48CM
AXIAL NODE 2	3, 2	3.32 WT% UO_2 / 10 - 3.8 WT% GD_2O_3	30.48CM
AXIAL NODE 1	1	URANIUM	15.24CM

Figure 5-19. Quad Cities Unit 2 Axial Loading - Assembly J

COLLAPSED NODES USED IN SAS2H	24 ACTUAL NODES	NODE DESCRIPTION	NODE HEIGHT
AXIAL NODE 10	24 23	URANIUM	30.48CM
AXIAL NODE 9	22 21 20	3.34 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 8	19 18 17	3.50 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 7	16 15 14	3.50WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 6	13 12 11	3.50 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 5	10 9	3.50 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 4	8 7 6	3.34 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 3	5 4	3.34 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 2	3 2	3.34 WT% UO_2 / 7 - 3.0 WT% GD_2O_3	30.48CM
AXIAL NODE 1	1	URANIUM	15.24CM

Figure 5-20. Quad Cities Unit 2 Axial Loading - Assembly K

COLLAPSED NODES USED IN SAS2H	24 ACTUAL NODES	NODE DESCRIPTION	NODE HEIGHT
AXIAL NODE 10	24, 23	URANIUM	30.48CM
AXIAL NODE 9	22, 21, 20	3.36 WT% UO_2 / 7- 3.0 WT% GD_2O_3	45.72CM
AXIAL NODE 8	19, 18, 17	3.54 WT% UO_2 / 7 - 3.14 WT% GD_2O_3	45.72CM
AXIAL NODE 7	16, 15, 14	3.54 WT% UO_2 / 7 - 3.14 WT% GD_2O_3	45.72CM
AXIAL NODE 6	13, 12, 11	3.54 WT% UO_2 / 7 - 3.14 WT% GD_2O_3	45.72CM
AXIAL NODE 5	10, 9	3.54 WT% UO_2 / 7 - 3.14 WT% GD_2O_3	30.48CM
AXIAL NODE 4	8, 7, 6	3.36 WT% UO_2 / 7 - 3.43 WT% GD_2O_3	45.72CM
AXIAL NODE 3	5, 4	3.36 WT% UO_2 / 7 - 3.43 WT% GD_2O_3	30.48CM
AXIAL NODE 2	3, 2	3.36 WT% UO_2 / 7 - 3.43 WT% GD_2O_3	30.48CM
AXIAL NODE 1	1	URANIUM	15.24CM

Figure 5-21. Quad Cities Unit 2 Axial Loading - Assembly L

COLLAPSED NODES USED IN SAS2H	24 ACTUAL NODES	NODE DESCRIPTION	NODE HEIGHT
AXIAL NODE 10	24, 23	URANIUM	30.48CM
AXIAL NODE 9	22, 21, 20	3.41 WT% UO_2 / 8 - 3.0 WT% Gd_2O_3	45.72CM
AXIAL NODE 8	19, 18, 17	3.57 WT% UO_2 / 8 - 3.75 WT% Gd_2O_3	45.72CM
AXIAL NODE 7	16, 15, 14	3.57 WT% UO_2 / 8 - 3.75 WT% Gd_2O_3	45.72CM
AXIAL NODE 6	13, 12, 11	3.57 WT% UO_2 / 8 - 3.75 WT% Gd_2O_3	45.72CM
AXIAL NODE 5	10, 9	3.57 WT% UO_2 / 8 - 3.75 WT% Gd_2O_3	30.48CM
AXIAL NODE 4	8, 7, 6	3.41 WT% UO_2 / 8 - 3.75WT% Gd_2O_3	45.72CM
AXIAL NODE 3	5, 4	3.41 WT% UO_2 / 8 - 3.75 WT% Gd_2O_3	30.48CM
AXIAL NODE 2	3, 2	3.41 WT% UO_2 / 8 - 3.75 WT% Gd_2O_3	30.48CM
AXIAL NODE 1	1	URANIUM	15.24CM

Figure 5-22. Quad Cities Unit 2 Axial Loading - Assembly M

Figures 5-23 and 5-24 show general diagrams (without radial dimensions) of the Path B models required to describe the different types of nodes. In the figures, R_n refers to the radial dimensions. The general model described in Figure 5-23 is used for all nodes that have no GDRs present. Figure 5-24 shows the general Path B model for all nodes containing GDRs. A listing of the equations referenced and utilized in the development of the radial dimensions in each of the Path B models is provided in Equations 5-11 through 5-16. Fuel pellet dimensions are equivalent for fuel rod and GDR calculations. For SAS2H modeling requirements unique to BWRs, fuel rod and GDR pellet dimensions were increased to the dimensions of the inner clad diameter and the material density was adjusted (smeared) accordingly. The bypass moderator region (R_5-R_6) contains bypass moderator at a density of 0.7396 g/cm^3 in nonbladed nodes and a homogenized control blade mixture for bladed nodes. The homogenized control blade mixture is explained in Section 5.2.2.

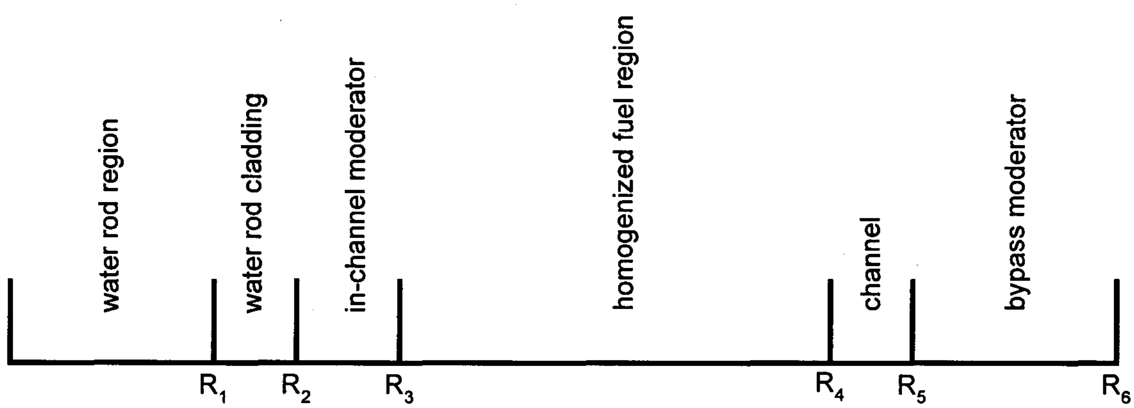

Figure 5-23. Quad Cities Unit 2 SAS2H General Path B Model: For Nodes That Do Not Contain GDRs

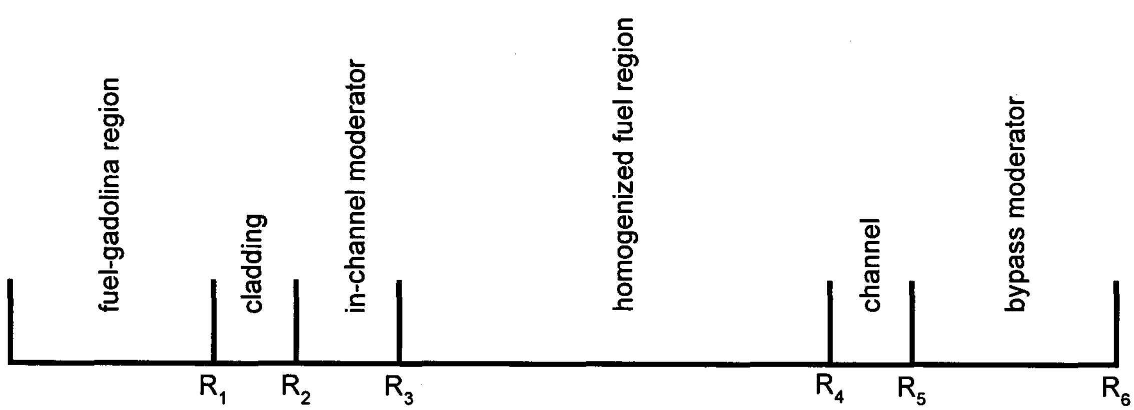

Figure 5-24. Quad Cities Unit 2 SAS2H General Path B Model: For Nodes That Contain GDRs

Table 5-9. Quad Cities Unit 2 SAS2H Path B Model Dimension Calculations: Path B Model Description 1

Input Parameters

Number of unit cells in assembly: 64
Number of fuel rods in assembly: 62
Rod pitch in assembly (cm): 1.6256
Fuel pellet diameter (cm): 1.0414
Fuel cladding inner diameter (cm): 1.0643
Fuel cladding outer diameter (cm): 1.2268
Assembly pitch (cm): 15.24
Water rod inside diameter (cm): 2.1285
Water rod outside diameter (cm): 2.4536
Number of GDRs: 0
Assembly channel inside width (cm): 13.4061
Assembly channel outside width (cm): 13.8125

Path B Model Dimensions

	Radius #	Outer Radius (cm)	Region Description	Equation #
Inner	1	1.0643	Water Rod	5-11
	2	1.2268	Water Rod Cladding	5-12
	3	1.2970	In-Channel Moderator	5-13
	4	7.3372	Homogenized Fuel Region	5-14
	5	7.5733	Channel	5-15
Outer	6	8.5982	Bypass Moderator	5-16

Table 5-10. Quad Cities Unit 2 SAS2H Path B Model Dimension Calculations: Path B Model Description 2

Input Parameters

Number of unit cells in assembly: 64
Number of fuel rods in assembly: 60
Rod pitch in assembly (cm): 1.6256
Fuel pellet diameter (cm): 1.0414
Fuel cladding inner diameter (cm): 1.0643
Fuel cladding outer diameter (cm): 1.2268
Assembly pitch (cm): 15.24
Water rod inside diameter (cm): 2.4561
Water rod outside diameter (cm): 2.6187
Number of GDRs: 0
Assembly channel inside width (cm): 13.4061
Assembly channel outside width (cm): 13.8125

Path B Model Dimensions

	Radius #	Outer Radius (cm)	Region Description	Equation #
Inner	1	1.2281	Water Rod	5-11
	2	1.3094	Water Rod Cladding	5-12
	3	1.8343	In-Channel Moderator	5-13
	4	7.3372	Homogenized Fuel Region	5-14
	5	7.5733	Channel	5-15
Outer	6	8.5982	Bypass Moderator	5-16

Table 5-11. Quad Cities Unit 2 SAS2H Path B Model Dimension Calculations: Path B Model Description 3

Input Parameters

Number of unit cells in assembly: 64
Number of fuel rods in assembly: 60
Rod pitch in assembly (cm): 1.6256
Fuel pellet diameter (cm): 1.0414
Fuel cladding inner diameter (cm): 1.0643
Fuel cladding outer diameter (cm): 1.2268
Assembly pitch (cm): 15.24
Water rod inside diameter (cm): 2.4561
Water rod outside diameter (cm): 2.6187
Number of GDRs: 0
Assembly channel inside width (cm): 13.4061
Assembly channel outside width (cm): 13.9141

Path B Model Dimensions

	Radius #	Outer Radius (cm)	Region Description	Equation #
Inner	1	1.2281	Water Rod	5-11
	2	1.3094	Water Rod Cladding	5-12
	3	1.8343	In-Channel Moderator	5-13
	4	7.3372	Homogenized Fuel Region	5-14
	5	7.6323	Channel	5-15
Outer	6	8.5982	Bypass Moderator	5-16

Table 5-12. Quad Cities Unit 2 SAS2H Path B Model Dimension Calculations: Path B Model Description 4

Input Parameters

Number of unit cells in assembly: 64
Number of fuel rods in assembly: 62
Rod pitch in assembly (cm): 1.6256
Fuel pellet diameter (cm): 1.0414
Fuel cladding inner diameter (cm): 1.0643
Fuel cladding outer diameter (cm): 1.2268
Assembly pitch (cm): 15.24
Number of GDRs: 7
Assembly channel inside width (cm): 13.4061
Assembly channel outside width (cm): 13.8125

Path B Model Dimensions

	Radius #	Outer Radius (cm)	Region Description	Equation #
Inner	1	0.5321	GDR	5-11
	2	0.6134	GDR Cladding	5-12
	3	0.9171	In-Channel Moderator	5-13
	4	2.7295	Homogenized Fuel Region	5-14
	5	2.8201	Channel	5-15
Outer	6	3.2498	Bypass Moderator	5-16

Table 5-13. Quad Cities Unit 2 SAS2H Path B Model Dimension Calculations: Path B Model Description 5

Input Parameters

Number of unit cells in assembly: 64
Number of fuel rods in assembly: 60
Rod pitch in assembly (cm): 1.6256
Fuel pellet diameter (cm): 1.0414
Fuel cladding inner diameter (cm): 1.0643
Fuel cladding outer diameter (cm): 1.2268
Assembly pitch (cm): 15.24
Number of GDRs: 9
Assembly channel inside width (cm): 13.4061
Assembly channel outside width (cm): 13.8125

Path B Model Dimensions

	Radius #	Outer Radius (cm)	Region Description	Equation #
Inner	1	0.5321	GDR	5-11
	2	0.6134	GDR Cladding	5-12
	3	0.9171	In-Channel Moderator	5-13
	4	2.3681	Homogenized Fuel Region	5-14
	5	2.4493	Channel	5-15
Outer	6	2.8661	Bypass Moderator	5-16

Table 5-14. Quad Cities Unit 2 SAS2H Path B Model Dimension Calculations: Path B Model Description 6

Input Parameters

Number of unit cells in assembly: 64
Number of fuel rods in assembly: 60
Rod pitch in assembly (cm): 1.6256
Fuel pellet diameter (cm): 1.0414
Fuel cladding inner diameter (cm): 1.0643
Fuel cladding outer diameter (cm): 1.2268
Assembly pitch (cm): 15.24
Number of GDRs: 7
Assembly channel inside width (cm): 13.4061
Assembly channel outside width (cm): 13.8125

Path B Model Dimensions

	Radius #	Outer Radius (cm)	Region Description	Equation #
Inner	1	0.5321	GDR	5-11
	2	0.6134	GDR Cladding	5-12
	3	0.9171	In-Channel Moderator	5-13
	4	2.6851	Homogenized Fuel Region	5-14
	5	2.7772	Channel	5-15
Outer	6	3.2498	Bypass Moderator	5-16

Table 5-15. Quad Cities Unit 2 SAS2H Path B Model Dimension Calculations: Path B Model Description 7

Input Parameters

Number of unit cells in assembly: 64
Number of fuel rods in assembly: 60
Rod pitch in assembly (cm): 1.6256
Fuel pellet diameter (cm): 1.0414
Fuel cladding inner diameter (cm): 1.0643
Fuel cladding outer diameter (cm): 1.2268
Assembly pitch (cm): 15.24
Number of GDRs: 11
Assembly channel inside width (cm): 13.4061
Assembly channel outside width (cm): 13.8125

Path B Model Dimensions

	Radius #	Outer Radius (cm)	Region Description	Equation #
Inner	1	0.5321	GDR	5-11
	2	0.6134	GDR Cladding	5-12
	3	0.9171	In-Channel Moderator	5-13
	4	2.1420	Homogenized Fuel Region	5-14
	5	2.2154	Channel	5-15
Outer	6	2.5925	Bypass Moderator	5-16

Table 5-16. Quad Cities Unit 2 SAS2H Path B Model Dimension Calculations: Path B Model Description 8

Input Parameters

Number of unit cells in assembly: 64
Number of fuel rods in assembly: 60
Rod pitch in assembly (cm): 1.6256
Fuel pellet diameter (cm): 1.0414
Fuel cladding inner diameter (cm): 1.0643
Fuel cladding outer diameter (cm): 1.2268
Assembly pitch (cm): 15.24
Number of GDRs: 10
Assembly channel inside width (cm): 13.4061
Assembly channel outside width (cm): 13.8125

Path B Model Dimensions

	Radius #	Outer Radius (cm)	Region Description	Equation #
Inner	1	0.5321	GDR	5-11
	2	0.6134	GDR Cladding	5-12
	3	0.9171	In-Channel Moderator	5-13
	4	2.2465	Homogenized Fuel Region	5-14
	5	2.3236	Channel	5-15
Outer	6	2.7190	Bypass Moderator	5-16

Table 5-17. Quad Cities Unit 2 SAS2H Path B Model Dimension Calculations: Path B Model Description 9

Input Parameters

Number of unit cells in assembly: 64
Number of fuel rods in assembly: 60
Rod pitch in assembly (cm): 1.6256
Fuel pellet diameter (cm): 1.0414
Fuel cladding inner diameter (cm): 1.0643
Fuel cladding outer diameter (cm): 1.2268
Assembly pitch (cm): 15.24
Number of GDRs: 7
Assembly channel inside width (cm): 13.4061
Assembly channel outside width (cm): 13.9141

Path B Model Dimensions

	Radius #	Outer Radius (cm)	Region Description	Equation #
Inner	1	0.5321	GDR	5-11
	2	0.6134	GDR Cladding	5-12
	3	0.9171	In-Channel Moderator	5-13
	4	2.6851	Homogenized Fuel Region	5-14
	5	2.8002	Channel	5-15
Outer	6	3.2498	Bypass Moderator	5-16

Table 5-18. Quad Cities Unit 2 SAS2H Path B Model Dimension Calculations: Path B Model Description 10

Input Parameters

Number of unit cells in assembly: 64
Number of fuel rods in assembly: 60
Rod pitch in assembly (cm): 1.6256
Fuel pellet diameter (cm): 1.0414
Fuel cladding inner diameter (cm): 1.0643

Fuel cladding outer diameter (cm): 1.2268
Assembly pitch (cm): 15.24
Number of GDRs: 8
Assembly channel inside width (cm): 13.4061
Assembly channel outside width (cm): 13.9141

Path B Model Dimensions

	Radius #	Outer Radius (cm)	Region Description	Equation #
Inner	1	0.5321	GDR	5-11
	2	0.6134	GDR Cladding	5-12
	3	0.9171	In-Channel Moderator	5-13
	4	2.5117	Homogenized Fuel Region	5-14
	5	2.6193	Channel	5-15
Outer	6	3.0399	Bypass Moderator	5-16

Table 5-19. Quad Cities Unit 2 SAS2H Path B Model Dimension Calculations: Matrix For Assembly Nodes Represented By Path B Model Type

	Path B	Nodes Represented By Path B Model For Assembly Type											
Model	**Model Description[1]**	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**J**	**K**	**L**	**M**
1	0GD/2WR/80MIL	1,10	1,10										
2	0GD/4WR/80MIL			1,10	1,10	1,10	1,10	1,10	1,10	1,10	1,10		
3	0GD/4WR/100MIL											1,10	1,10
4	7GD/2WR/80MIL	2-9	2-9										
5	9GD/4WR/80MIL			2-9		8	2-7, 9	8					
6	7GD/4WR/80MIL				2-9	2-7, 9		2-7, 9	2-9		2-9		
7	11GD/4WR/80MIL						8						
8	10GD/4WR/80MIL									2-9			
9	7GD/4WR/100MIL											2-9	
10	8GD/4WR/100MIL												2-9

[1]0GD = number of GDRs in model
2WR = 1 smaller water rod (2 rod area)
4WR = 1 large water rod (4 rod area)
80MIL = channel material thickness 80 mil (0.2032 cm)
100MIL = channel material thickness 100 mil (0.254 cm)

The equations listed below were derived. All distance dimensions are in centimeters. All area dimensions are in square centimeters. All other parameters are dimensionless.

Equation 5-11. Water Rod or GDR Radius in Center Zone of Path B Model

$$R_1 = ID_{clad} / 2$$

where: R_1 is the radius of the center or first Path B zone; ID_{clad} is the inner diameter of the cladding of either the GDR or the water rod.

Equation 5-12. Water Rod or GDR Cladding Radius in Second Zone of Path B Model

$$R_2 = OD_{clad} / 2$$

where: R_2 is the radius of the second Path B zone; OD_{clad} is the outer diameter of the cladding for either the water rod or GDR.

Equation 5-13. Unit Cell Moderator Radius in Third Zone of Path B Model

$$R_3 = \sqrt{\frac{\#\text{waterrods} \cdot \text{rodpitch}^2}{p}}$$

where: R_3 is the radius of the third Path B zone; *#waterrods* is 2 for 0GDR/2WR/80-100MIL Path B models, 4 for 0GDR/4WR/80-100MIL Path B models, and 1 for the #GDR/4WR/80-100MIL; *rodpitch* is the pin pitch of the assembly.

Equation 5-14. Homogenized Fuel Radius in Fourth Zone of Path B Model

$$R_4 = \sqrt{R_3^2 + \frac{\#\text{fuelrods} - \#\text{gdrods}}{\#\text{gdrods}} \cdot \frac{\text{rodpitch}^2}{p}}$$

where: R_3 is the radius of the third Path B zone; R_4 is the radius of the fourth Path B zone; *#fuelrods* is the total number of fuel and GDRs in the assembly; *#gdrods* is the number of GDRs in the assembly (note: in nodes that contain no GDRs the *#gdrods* term is completely removed from the above equation leaving only the *#fuelrods* term); *rodpitch* is the pin pitch of the assembly.

Equation 5-15. Assembly Channel Radius in Fifth Zone of Path B Model

$$R_5 = \sqrt{R_4^2 + \frac{1}{\#gdrods} \cdot \frac{outwidth^2 - inwidth^2}{p}}$$

where: R_4 is the radius of the fourth Path B zone; R_5 is the radius of the fifth Path B zone; *#gdrods* is the number of GDRs in the assembly (note: in nodes that contain no GDRs the #gdrods term is completely removed from the above equation); *outwidth* is the outer width of the assembly channel; *inwidth* is the inner width of the assembly channel.

Equation 5-16. Bypass Moderator Radius in Sixth Zone of Path B Model

$$R_6 = \sqrt{\frac{1}{\#gdrods} \cdot \frac{assempitch^2}{p}}$$

where: R_6 is the radius of the sixth Path B zone; *#gdrods* is the number of GDRs in the assembly (note: in nodes that contain no GDRs the *#gdrods* term is completely removed from the above equation); *assempitch* is the assembly pitch of the Quad Cities 2 reactor.

5.5. SPACE-CRAFT Software Routine and Usage

The SPACE software is a driver routine for CRAFT. For the Quad Cities 2 BWR assemblies, the CRAFT routine is operated in the one node format. This means that each time CRAFT is executed, the SAS2H files and runs are only for one node of the designated assembly. The SPACE routine operates CRAFT in a loop, which processes the CRAFT and SAS2H cases for the entire assembly. The SPACE routine allows the user to input all of the required information describing an assembly into one SPACE input file "spacein".

The CRAFT software routine directed the performance of the assembly depletion and decay calculations relevant to CRC evaluations. The CRAFT software routine generated input files for the SAS2H control module of the SCALE modular code system based on user-defined nodal input, which described the fuel assembly's nodal specifications and irradiation history. Appropriate isotopic compositions relevant to the CRC evaluations containing the fuel assembly and subsequent depletion and decay calculations of the fuel assembly were extracted and stored by CRAFT as it generated and executed the SAS2H cases for the complete fuel assembly irradiation history.

The CRAFT software routine was developed with a high degree of flexibility to provide for the depletion and decay of fuel assemblies that have widely varying features under flexible core operating conditions. The following listing describes some of the capabilities and usage of CRAFT.

- The CRAFT software routine generates and executes appropriate SAS2H cases required to perform a prescribed depletion and decay sequence for a fuel node. The depletion and decay sequence is orchestrated from the beginning of cycle (BOC) statepoint calculation of the initial prescribed insertion cycle through the final statepoint calculation of the last prescribed insertion cycle. The CRAFT software routine extracts and saves fuel and GDR isotopics at each statepoint, including BOC statepoints, during the fuel node's depletion and decay sequence. A certain number of the generated isotopics in the depleted fuel composition obtained from a SAS2H calculation are not used in the initial charge composition to the next SAS2H calculation due to a lack of cross section data in the specified SAS2H master cross section library. The CRAFT software routine provides a listing of the fuel isotopics from the output of a SAS2H calculation which are not used in the initial charge to the next SAS2H calculation. The isotopics left out of the initial charge are fission products whose reactivity worth is small relative to the isotopics retained in the initial charge composition. The listing of excluded initial charge isotopics (*.notes files) allows for a determination of the impact upon the reactivity worth of the initial fuel composition in the subsequent depletion calculation.

- The fuel cladding, GDR cladding, or CRB cladding in the CRAFT calculation may be designated as any of the following materials: ZIRC-4, INCONEL, SS316, SS316S, SS304, SS304S.
- A fuel assembly may be inserted in a maximum of 10 reactor cycles during a CRAFT calculation.
- A maximum of 20 statepoints or datapoints (BOC is always considered a statepoint) may be specified in any given reactor cycle in a CRAFT calculation.
- A maximum of 23 irradiation steps of variable duration may be specified in any given SAS2H depletion calculation that is generated by CRAFT.
- A maximum of 50 axial assembly nodes may be specified for use in a CRAFT calculation. Each axial node may have a unique height.
- The CRAFT software routine utilizes a user-defined input format for fuel temperature, moderator density, and burnup data. The input data must be specified for each axial node in a user-defined nodal format of up to 50 nodes of arbitrary height. The total assembly active fuel height for the input data descriptions may be different than that specified for use in the CRAFT generated SAS2H depletion calculations. Nominal fuel temperature input data representing full-power reactor operation must be provided in units of degrees Kelvin for each node in each CRC statepoint depletion calculation that will be generated by CRAFT. Nominal moderator density input data representing full-power reactor operation must be provided in units of grams per cubic centimeter for each node in each statepoint calculation that will be generated by CRAFT. The nodal average burnup input data must be provided in units of GWd/MTU for each node at each statepoint or datapoint including all BOC statepoints. All burnup input data that is specified must be cumulative from the initial insertion of the fuel assembly in the reactor.

Additional information on the CRAFT software routine is provided in the CRAFT user information in Attachment I. Instructions on how to develop CRAFT input files and execute CRAFT calculations are also provided in Attachment I. This attachment also discusses specific modeling procedures and details relevant to the SAS2H fuel assembly depletion calculations, which were generated by CRAFT.

5.6. Filename Descriptions for SPACE-CRAFT and SAS2H

The SPACE software routine generates CRAFT calculation input files for various types of CRAFT calculations. The CRAFT input files generated by SPACE are named with a two digit integer identical to the node number that is being processed in the CRAFT-SAS2H calculation. The filenames for these input files would be as follows:

GDR and Uranium Blanket CRAFT input files:
01, 02, 03, 04, 05, 06, 07, 08, 09, 10

When a CRAFT execution for a node is complete the CRAFT input file for the node is renamed with the "*.B" extension.

The CRAFT code generated five types of files identified as either "*.input", "*.output", "*.cut", "*.msgs", or "*.notes", where the "*" is the base file set identifier for the statepoint depletion calculation of interest. The "*.cut" and "*.notes" files were the only files that had to be retained for CRC reactivity evaluations and documentation purposes. All files were generated in the working directory in which the CRAFT calculation was performed.

All CRAFT generated filenames utilized the following format: "{Base File Set Identifier}.{suffix}", where the suffix corresponded to one of the five file types previously mentioned, and the base file set identifier was a 25 character name containing essential information necessary to uniquely identify each CRAFT generated SAS2H depletion calculation.

The base file set identifier for each statepoint depletion calculation contained the following information:

1. reactor identifier (three character)
2. one-eighth core symmetry assembly number in current reactor cycle (two digit)
3. axial node number (node 1 is always the bottom node) (two digit)
4. reactor cycle number in which the SAS2H calculation starts (two character)
5. EFPD statepoint at which the SAS2H calculation starts (three digit)
6. reactor cycle number in which the SAS2H calculation ends (two character)
7. EFPD statepoint at which the SAS2H calculation ends (three digit).

The format of the base file set identifier was as follows where the numbers identified as #{number} correspond to one of the seven items previously listed-- #1 A #2 N #3 DC #4 T #5 AC #6 T #7. The letters contained in the base file set identifier were presented explicitly as shown in the previous format. The base file set identifier did not contain any spaces.

The "*.input" files each contained a CRAFT generated SAS2H input deck. The "*.output" files each contained a complete SAS2H depletion calculation output file. The "*.cut" files each contained the corresponding SAS2H input deck followed by an output extraction from the final ORIGEN-S pass of the SAS2H depletion calculation, which contained data relevant to subsequent CRC reactivity calculations. The "*.msgs" files each contained the standard run-time messages associated with the SAS2H calculations. The "*.notes" files each contained a listing of the isotopes and associated concentrations which were left behind in generating the initial charge fuel composition for the next continuation SAS2H calculation. The "*.notes" files were only created for CRAFT generated SAS2H calculations which were continuation depletion calculations. The "*.cut" and "*.notes" files contained all of the information required to perform CRC reactivity evaluations or repeat calculations as necessary for quality assurance purposes. The remainder of the CRAFT generated files were discarded once the "*.cut" and "*.notes" files were generated and retained.

In Attachment III, all SPACE and CRAFT input files are renamed and stored to a writable compact disk (WCD) (Reference 7.10). The naming method for all SPACE and CRAFT input files that are stored on the WCD is demonstrated below.

CRAFT Input Files: Naming Format: "MMANN" Example: "05J12"

where:

MM - node number
A - letter designation of the assembly type
NN - assembly number

SPACE Input Files: Naming Format: "SPANN" Example: "spJ12"

where:

SP - designation for SPACE input file
A - letter designation of the assembly type
NN - the assembly number

6. RESULTS

This calculation contains assumptions as listed in Section 3 that must be confirmed prior to the use of the results of this calculation to support construction, fabrication, or procurement activities.

Depletion calculations for 121 fuel assemblies from Quad Cities Unit 2 were documented in this analysis. The depleted fuel and depleted burnable poison isotopics for these fuel assemblies had to be calculated at a number of statepoints in Cycles 13 and 14 for use in subsequent CRC reactivity calculations. Table 5-7 identifies the CRC statepoint EFPD values in each of these cycles for which isotopic compositions were required. Table 5-7 also identifies a number of datapoints at which the depletion calculations were interrupted to update input parameters. Even though the depleted isotopics available at each of the datapoints were not required for subsequent reactivity calculations, they were retained in this calculation for completeness.

The SPACE input files for each assembly depletion were developed in accordance with the instructions presented in Sections 5 and 7 of Attachment II. The SPACE modeling features incorporated in the depletion calculations are described in Attachment II. The SPACE input files for the assembly depletions documented in this calculation are provided in Attachment III and Reference 7.10, as documented in Section 8.

The CRAFT input files for each nodal depletion were developed in accordance with the instructions presented in Sections 5 and 7 of Attachment I. The SAS2H modeling features incorporated in the depletion calculations are described in Attachment I. The CRAFT input files for the nodal depletions documented in this calculation are provided in Attachment III and Reference 7.10, as documented in Section 8.

Attachment IV and Reference 7.10, contain the CRAFT generated consolidated SAS2H output files for the depletion calculations documented in this analysis as identified in the attachment listing of Section 8. The consolidated output files contain the following information:

- time/date stamp for when the SAS2H depletion calculation was performed
- echo of the SAS2H input deck generated by CRAFT
- the output extraction of information pertinent to CRC evaluations from the final ORIGEN-S calculation of the SAS2H depletion calculation.

Isotopic results for the set of 29 principal isotopes identified in Table 6-1 were processed for each assembly node at each CRC statepoint other than beginning of life (BOC of first reactor cycle in which0 the assembly is inserted) statepoint.

Table 6-1. The Set of 29 Principal Isotopes

Mo-95	Tc-99	Ru-101	Rh-103	Ag-109
Nd-143	Nd-145	Sm-147	Sm-149	Sm-150
Sm-151	Sm-152	Eu-151	Eu-153	Gd-155
U-233	U-234	U-235	U-236	U-238
Np-237	Pu-238	Pu-239	Pu-240	Pu-241
Pu-242	Am-241	Am-242m	Am-243	---

7. REFERENCES

In addition to the references listed below, Attachments I and II include a reference section which lists Reference 7.1 separately.

7.1 Oak Ridge National Laboratory 1995. *SCALE, Version 4.3: Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation.* User's Manual Volumes 0 through 3. CCC-545. Oak Ridge, Tennessee: Distributed by the Radiation Shielding Information Center. TIC: 235920.

7.2 CRWMS M&O (Civilian Radioactive Waste Management System Management and Operating Contractor) 1997. *Software Qualification Report for the SCALE Modular Code System Version 4.3.* CSCI: 30011 V4.3. DI: 30011-2002 REV 01. Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19970731.0884.

7.3 CRWMS M&O 1999. *Summary Report of Commercial Reactor Criticality Data for Quad Cities Unit 2.* B00000000-01717-5705-00096 REV 01. Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19990917.0184.

7.4 Briesmeister, Judith F., ed. 1997. *MCNP, Version 4B: Monte Carlo N-Particle Transport Code System.* User Manual. LA-12625-M, Version 4B. Los Alamos, New Mexico: Los Alamos National Laboratory. TIC: 241044.

7.5 General Electric Company 1976. *Core Design and Operating Data for Cycles 1 and 2 of Quad Cities 1.* NP-240, Project 497-1. San Jose, California: Electric Power Research Institute. TIC: 237267.

7.6 Bowman, S.M.; Hermann, O.W.; and Brady, M.C. 1995. *Scale-4 Analysis of Pressurized Water Reactor Critical Configurations: Volume 2 - Sequoyah Unit 2 Cycle 3.* ORNL/TM-12294/V2. Oak Ridge, Tennessee: Oak Ridge National Laboratory. TIC: 244397.

7.7 General Electric Company 1989. *Nuclides and Isotopes*, Fourteenth Edition. San Jose, California: General Electric Company. TIC: 201637

7.8 CRWMS M&O 1999. *Waste Package Material Properties*. BBA000000-01717-0210-00017 REV 00. Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19990407.0172.

7.9 Spence, R.E. 1999. "Accepted Data Call." Letter from R.E. Spence (Acting Assistant Manager, Office of Project Execution, DOE/YMSCO) to J.L. Younker (Deputy Assistant General Manager, Technical, TRW Environmental Safety Systems, Inc.), July 27, 1999. ACC: MOL.19990811.0170.

7.10 CRWMS M&O 1999. *Six (6) Compact Discs for CRC Depletion Calculations for Quad Cities Unit 2, Attachments III, IV, V, and VIII.* B00000000-01717-0210-00009 REV 01. Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19990923.0237.

7.11 Mellington, S.P. 1999. "Office of Project Execution (OPE) Approval of LV.WP.TWD.08/99-131 as Accepted Data." Letter from S.P. Mellington (Assistant Manager, Office of Project Execution, DOE/YMSCO) to T.W. Doering (Framatome Cogema Fuels, CRWMS M&O), September 20, 1999. ACC: MOL.19990927.0472.

8. ATTACHMENTS

The attachments referenced throughout this calculation are listed in Table 8-1. Attachment I is the CRAFT software routine documentation. Attachment II is the SPACE software routine documentation. Attachment III contains the SPACE-CRAFT input files for the assembly depletion calculations. Attachment IV contains the "*.cut" files for the assembly depletion calculations. Attachment V contains the "*.notes" files for the assembly depletion calculations. Attachment VI lists the burnup, fuel temperature, and moderator density information for Quad Cities 2. Attachment VII contains the assembly control blade insertion history statements. Attachments III, IV, V, and VIII were written in ASCII format to a writable compact disk (WCD) (Reference 7.10). Detailed listings of the content of Attachments III, IV, V, and VIII on the WCD are provided in their corresponding hard-copy attachment locations in this calculation. The listing of the content of Attachments III, IV, V, and VIII contain the following information, as appropriate, for each of the files that were written to the WCD:

- the directory and filename as taken from the HP workstation
- the corresponding filename on the attachment WCD
- the date that the file was created on the HP workstation or personal computer
- the size of the file on the HP workstation or personal computer in bytes.

The WCDs containing Attachments III, IV, V, and VIII, (Reference 7.10), were written using the HP Compact Disk Writer Plus 7200e External CD-ReWritable Drive for personal computers.

Table 8-1. Attachment Listing

Attachment #	# of Pages	Creation Date	Description
I	151	05/28/99	CRAFT, Version 4C, User Information
II	82	05/28/99	SPACE, Version 01, User Information
III	29 (Hard-Copy Listing of WCD Content)	6/03/99 (Reference 7.10)	SPACE-CRAFT Input Files for the Quad Cities Unit 2 Depletion Calculations
IV	163 (Hard-Copy Listing of WCD Content)	6/10/99 (Reference 7.10)	"*.cut" Consolidated Output Files for the Quad Cities Unit 2 Depletion Calculations
V	146 (Hard-Copy Listing of WCD Content)	6/08/99 (Reference 7.10)	"*.notes" Files for the Quad Cities Unit 2 Depletion Calculations
VI	256	5/28/99	Burnup, Fuel Temperature, and Moderator Density Information for Quad Cities 2
VII	37	5/28/99	Assembly Control Blade Insertion History Statements
VIII	8 (Hard-Copy Listing of WCD Content)	9/8/99 (Reference 7.10)	Corrected Depletion Calculations for Assembly Type J (Fuel Type 16)

CRAFT, Version 4C
Documentation of Modifications to CRAFT, Version 4

Commercial Reactor Assembly Follow Taskmaster
CRAFT, Version 4 developed by Kenneth D. Wright
CRAFT, Version 4C developed by David P. Henderson

Framatome Cogema Fuels
High-Level Waste Division

under contract with the

Management and Operating Contractor for the
Yucca Mountain High-Level Radioactive Waste Repository Project

Table of Contents

Item **Page**

1. Introduction

The Commercial Reactor Assembly Follow Taskmaster (CRAFT, Version 4C) software routine directs the performance of assembly nodal depletion and decay calculations relevant to Commercial Reactor Critical (CRC) evaluations for Boiling Water Reactors (BWRs). The CRAFT software routine has been modified as described in Section 3 to perform depletion and decay calculations on a nodal rather than assembly basis. The capability of CRAFT to perform depletion and decay calculations for Pressurized Water Reactor (PWR) assemblies has been disabled in Version 4C. Details of the general operation and subroutine structure of the CRAFT software routine are located in Reference 1. Details of the modifications made to CRAFT for the processing of BWR assemblies are included in Section 3 of this attachment. Additionally, information describing input file requirements for CRAFT Version 4C are provided in Section 4 of this attachment.

The CRAFT software routine generates input files for the SAS2H control module of the SCALE modular code system (Reference 2) based on user-defined input which describes the BWR fuel node irradiation history. Isotopic concentrations relevant to both the CRC evaluations containing the fuel node and the subsequent depletion and decay calculations for the fuel node are extracted and stored by CRAFT as it generates and executes SAS2H cases.

2. CRAFT, Version 4C

The CRAFT, Version 4C software routine directs the performance of depletion and decay calculations required to simulate the complete irradiation history of specific nodes defining a fuel assembly. During the CRAFT orchestration of the fuel assembly nodal depletion and decay calculations, fuel and burnable poison isotopic concentrations are retained at user-defined statepoints. The fuel and burnable poison isotopic concentrations may be used for input to subsequent CRC statepoint reactivity calculations or in other analyses concerning spent nuclear fuel from commercial power reactors.

The CRAFT, Version 4C software routine was developed specifically for use with BWR assembly nodes. A degree of flexibility remains in CRAFT that allows for the depletion and decay of nodes that vary with axial location in an assembly. Features available for BWR assemblies available in CRAFT are listed as follows:

- CRAFT operates in a nodal format for isotopic depletion rather than orchestrating the depletion calculations for all of the nodes in one assembly. A separate driver software routine entitled System Processor and CRAFT Executor (SPACE), Version 01, creates the CRAFT input files and then executes CRAFT for each individual nodal calculation required for a specific BWR assembly design. The SPACE, Version 01 software routine is described in Attachment II of this document. One calculation using CRAFT can generate all of the SAS2H input files to process and deplete one specific node of an assembly through its lifetime in a BWR core. The nodal processing capability of CRAFT combined with SPACE, enable SAS2H input files to be generated and executed for an entire

BWR assembly. This feature of CRAFT is useful in extending modeling capability to the multiple axial enrichments and varied gadolinia-bearing fuel rod (GDR) configurations common to BWR assemblies.

- In CRAFT, depletion and decay calculations for GDR nodes track the primary gadolinia isotopes through each statepoint. The depleted gadolinia isotopics are passed from the end of one statepoint calculation to the beginning of the next for each GDR node in a BWR assembly.
- BWR assembly in-channel moderator densities and fuel average temperatures are tracked independently for each node of an assembly and are updated in each new SAS2H statepoint depletion calculation. Additionally, in-channel moderator densities are input into the SAS2H calculations as a function of irradiation time step durations in the calculation.

3. Modifications Made Between CRAFT, Version 4 and Version 4C

Modifications between the CRAFT, Version 4, and Version 4C, software routines are described in this section. CRAFT, Version 4, was created strictly to incorporate features for BWR fuel assembly depletion calculations. CRAFT, Version 4, provides no features beyond those present in CRAFT, Version 3 that relate to PWR fuel assembly depletion calculations. For completeness, the initial features available in CRAFT, Version 4, for BWR assembly depletions are briefly outlined below (Reference 1). Although Reference 1 identifies Version 5 of the CRAFT routine, this document also contains the information specific to Version 4.

- The capability to follow moderator density changes as a function of each irradiation time step in each CRAFT-generated SAS2H calculation is available in CRAFT, Version 4.
- The capability to provide different densities for each of the control blade absorber material specifications is available in CRAFT, Version 4.
- The elimination of certain input requirements that are only necessary for PWR assembly depletion calculations is available in CRAFT, Version 4.

Modifications made to CRAFT, Version 4 to create Version 4C are listed below. These modifications were made to improve the capability of CRAFT in processing BWR assembly depletion and decay calculations.

Modification 1

CRAFT, Version 4 was modified to process the depletion of gadolinium isotopes in one SAS2H calculation and then pass the depleted isotopes into the material specification of the subsequent SAS2H calculation. This enables CRAFT and SAS2H to model the depletion of the GDRs as accurately as possible within the inherent limitations of the SAS2H program.

Modification 2

A time/date/version stamp routine was developed in CRAFT, Version 4C. This time/date/version stamp appears on every SAS2H input file that is created by the software routine.

Modification 3

CRAFT, Version 4 was modified to use the fuel rod inner cladding radius as the fuel pellet dimension for the nodal SAS2H depletion calculation. This modification was required because SAS2H does not model the helium gap of a GDR properly. The fuel density is corrected to compensate for the smearing effect of the larger simulated fuel pellet dimension.

Modification 4

CRAFT, Version 4 was modified to enable multiple control blade materials to be specified for all nodal depletion calculations. The control blades (CRBs) are required to be inserted at all times in the CRAFT nodal calculation model whether there is an actual CRB located in the node or not. This modification allows the user to specify the GDR materials for the SAS2H Path B model and also specify either the homogenized borated water to model the presence of a CRB or non-borated water to simulate a non-bladed node.

Modification 5

In CRAFT, Version 4, the boron stepdown function was modified to track the changes of in-channel moderator densities with SAS2H irradiation step length. CRAFT calculates fractions of a user-defined reference density which are then used to determine the SAS2H "H2OFRAC" parameter. This enables in-channel moderator densities to be different for each node and to change during the nodal calculation depending upon statepoint and irradiation step length.

The source code changes that were made between CRAFT, Versions 4 and 4C to incorporate the modifications listed in this section are presented in Table 3-1. Table 3-1 shows the lines of code that would need to be altered to make CRAFT Versions 4 and 4C identical. The information in Table 3-1 was obtained using the "diff" command that is available on the Hewlett Packard 9000 series workstations. The lines with "<" in column one represent CRAFT Version 4 source code. The lines with ">" represent CRAFT Version 4C source code. The corresponding source code line numbers are provided above each set of lines initiated by either "<" or ">".

Table 3-1. Source Code Differences Between CRAFT Versions 4 and 4C

```
1a2,52
> **************************************************************************
> *   Commercial Reactor Assembly Follow Taskmaster
> *   Boiling Water Reactor (BWR) Version
> *
> *
> *
> *   craftv4c.f  version 4c  12/28/98  dph
> **************************************************************************
> *   This software routine writes the SAS2H input decks necessary to
> *   perform depletion and decay calculations on an assembly
> *   required in subsequent Commercial Reactor Critical
> *   evaluations. The software routine controls the SAS2H input deck
> *   creation such that a new SAS2H input deck is developed
> *   to perform depletion and decay calculations between CRC
> *   statepoints in a given sequence. The depletion and
> *   decay of the fuel assembly through all CRC statepoints
> *   is simulated as a continuous process by using feed fuel
> *   isotopics from the previous calculation in the sequence.
> **************************************************************************
> *   This software routine is only for use with BWR CRC depletion
> *   calculations.
> **************************************************************************
> *
> *   1/8/98  : dph blocked mod to detect fuel in center zone of pathB
> *             and adjust SAS2H input
> *
> *   1/15/98 : dph turned off fuel temp conversion for BWRs
> *
> *   1/16/98 : dph inserted STOP feature for RESTARTS
> *
> *   1/19/98 : dph replaced boron depletion secton for BPRA to
> *             Gd depletion and calculation of depleted
> *             Gd and UO2 mixture for continuation case
> *
> *
> *   3/10/98 : dph date/time stamp feature and clean up
> *             version 4b
> *
> *   12/28/98: dph version 4c
> *
> *   12/29/98: dph made corrections for first moderator density
> *             calculation in SAS2H
> *
> *
> *    1/23/99: dph modified calculation of gadolinia isotopics for
> *             continuation case
> *
> *    2/4/99: dph cleanup and corrected bug in retreiver sub to prevent
> *            misread of gadolinia isotope masses
> *
```

```
> *
3,15d53
< *   Commercial Reactor Assembly Follow Taskmaster
< ****************************************************************
< *   This code writes the SAS2H input decks necessary to
< *   perform depletion and decay calculations on an assembly
< *   required in subsequent Commercial Reactor Critical
< *   evaluations.  The code controls the SAS2H input deck
< *   creation such that a new SAS2H input deck is developed
< *   to perform depletion and decay calculations between CRC
< *   statepoints in a given sequence.  The depletion and
< *   decay of the fuel assembly through all CRC statepoints
< *   is simulated as a continuous process by using feed fuel
< *   isotopics from the previous calculation in the sequence.
< ****************************************************************
24c62
<      c CRISOID(25,10), AXBLANK(50), AXBLANKNODNUM,
---
>      c CRISOID(25,20), AXBLANK(50), AXBLANKNODNUM,
35c73,74
<      c APSRFM(15,10), APSRFOLLOWMIX(10,20,23,50)
---
>      c APSRFM(15,10), APSRFOLLOWMIX(10,20,23,50),CYCSTOP,
>      C STPTSTOP,GDROD
37c76
<       REAL CLTEMP, PRESS, BPDEN(10), BPRA(15,10), CRISOWTPCT(25,10),
---
>       REAL CLTEMP, PRESS, BPDEN(10), BPRA(15,10), CRISOWTPCT(25,20),
50c89
<      c CRMIXDEN(25)
---
>      c CRMIXDEN(25),FIRSTMODEN(20,20)
52c91
<       CHARACTER REACT*21, PREFIX*3, AXBLANKET*1, BPRFLAG*1,
---
>       CHARACTER REACT*23, PREFIX*3, AXBLANKET*1, BPRFLAG*1,
54,55c93,95
<      c APSRSTAT*6, LIB*15, NM*31, CLADDESNAME(10)*7,
<      c SPACERMAT*7, STEPCONTROL*1, ABOVEBP(10)*5, RTYPE*3
---
>      c APSRSTAT*6, LIB*15, NM*31, CLADDESNAME(10)*7,STOPFLAG*1,
>      c SPACERMAT*7, STEPCONTROL*1, ABOVEBP(10)*5, RTYPE*3,
>      C PICKUPFLAG*1
96a137
> C       OPEN (UNIT=99, FILE='debug', STATUS='unknown')
125c166,170
<      c RTYPE, MODREFDEN, MODREFTEMP, CRMIXDEN)
---
>      c RTYPE, MODREFDEN, MODREFTEMP, CRMIXDEN,STOPFLAG,CYCSTOP,
>      C STPTSTOP,PICKUPFLAG,GDROD,FIRSTMODEN)
>
>
>
133,135c178,180
<       write (*,*) 'calling fueltemp_format'
<       CALL FUELTEMP_FORMAT (STPTSUM, AXNUM, FTNUM,
<      c NODES, FTNDES, FTDAT, FTIN)
```

```
---
>       write (*,*) 'calling fueltemp_format'
>       CALL FUELTEMP_FORMAT (STPTSUM, AXNUM, FTNUM,
>    c  NODES, FTNDES, FTDAT, FTIN)
137,141c182,186
<     IF (RTYPE.EQ.'PWR') THEN
<       write (*,*) 'calling modspecvol_format'
<       CALL MODSPECVOL_FORMAT (STPTSUM, AXNUM, MONUM,
<    c  NODES, MONDES, MODAT, MOIN)
<     ENDIF
---
> C     IF (RTYPE.EQ.'PWR') THE
> C       write (*,*) 'calling modspecvol_format'
> C       CALL MODSPECVOL_FORMAT (STPTSUM, AXNUM, MONUM,
> C    c  NODES, MONDES, MODAT, MOIN)
> C     ENDIF
156c201
<    c DENDAT, RTYPE, MODREFTEMP, RTYPE)
---
>    c DENDAT, RTYPE, MODREFTEMP)
188c233,234
<    c MODREFTEMP, CRMIXDEN)
---
>    c MODREFTEMP, CRMIXDEN,STOPFLAG,CYCSTOP,STPTSTOP,
>    C PICKUPFLAG,FTNDES,GDROD,FIRSTMODEN)
190c236,239
<     END
---
>
> C     close(99)
>
>       END
223c272,273
<    c RTYPE, MODREFDEN, MODREFTEMP, CRMIXDEN)
---
>    c RTYPE, MODREFDEN, MODREFTEMP, CRMIXDEN,STOPFLAG,CYCSTOP,
>    C STPTSTOP,PICKUPFLAG,GDROD,FIRSTMODEN)
232c282
<    c CRISOID(25,10), AXBLANK(50), AXBLANKNODNUM, AXBLANKTEMP,
---
>    c CRISOID(25,20), AXBLANK(50), AXBLANKNODNUM, AXBLANKTEMP,
240c290
<    c APSRCLAD(10), BPMIXNUM, BPMIX(10), BPMIXID(10),
---
>    c APSRCLAD(10), BPMIXNUM, BPMIX(10), BPMIXID(10),GDROD,
244c294
<    c NUMOFSECTIONS, SECT
---
>    c NUMOFSECTIONS, SECT,CYCSTOP,STPTSTOP
246c296
<     REAL CLTEMP, PRESS, BPDEN(10), BPRA(15,10), CRISOWTPCT(25,10),
---
>     REAL CLTEMP, PRESS, BPDEN(10), BPRA(15,10), CRISOWTPCT(25,20),
255c305,306
<    c APSRFR(15,10), MODREFDEN, MODREFTEMP, CRMIXDEN(25)
---
>    c APSRFR(15,10), MODREFDEN, MODREFTEMP, CRMIXDEN(25),
```

```
>     c FIRSTMODEN(20,20)
257,258c308,309
<      CHARACTER REACT*21, PREFIX*3, AXBLANKET*1, BPRFLAG*1,
<     c FUELCLAD*10, FLAG2*7, CYCLEID(10)*2, CRSTAT*6,
---
>      CHARACTER REACT*23, PREFIX*3, AXBLANKET*1, BPRFLAG*1,
>     c FUELCLAD*10, FLAG2*7, CYCLEID(10)*2, CRSTAT*6,STOPFLAG*1,
278a330,345
>
>      READ (10,*)
>      READ (10,*)
>      READ (10,*)
>      READ (10,*)
>
>      READ (10,3) STOPFLAG
>      IF(STOPFLAG.EQ.'Y')THEN
>        READ (10,4) CYCSTOP
>        READ (10,4) STPTSTOP
>      ELSE
>        CYCSTOP=0
>        STPTSTOP=0
>      ENDIF
>    3 FORMAT(A1)
>    4 FORMAT(I3)
286,287c353,354
<        READ(10,*) CT1START
<        READ(10,*) CT2START
---
>        READ(10,*) CT1START ! cycle start
>        READ(10,*) CT2START ! statepoint start
291a359,366
>
> C     IF (PICKUPFLAG.EQ.'N'.AND.STOPFLAG.EQ.'Y') THEN
> C       READ(10,*) GDROD
> C     ENDIF
> C always read GDROD in
>
>        READ(10,*) GDROD
>
294c369
< *                   (up to 21 characters).
---
> *                   (up to 23 characters).
299a375,393
>
>     IF (RTYPE.EQ.'PWR') THEN
>       write (*,*) ''
>       write (*,*) ''
>       write (*,*) '****************************************',
>    &           '****************************************',
>    &           '* WARNING: This version of CRAFT is  *',
>    &           '* only for BWR reactors and can not  *',
>    &           '* be used for PWR depletion          *',
>    &           '* calculations. Further execution of *',
>    &           '* this software routine will be      *',
>    &           '* cancelled.                         *',
>    &           '****************************************',
```

```
>     &          '****************************************'
>       write (*,*) ' '
>       write (*,*) ' '
>       GOTO 999
>     ENDIF
>
303c397
<   10 FORMAT (A21)
---
>   10 FORMAT (A23)
485a580,587
>                 IF(CT3.EQ.1)THEN
>                   FIRSTMODEN(CT1,CT2)=
>     C               VARBLETDOWN(CYCHOLDER,STPTHOLDER,CT3,2)
> c                 ELSE
> c                   FIRSTMODEN(CT1,CT2)=0.0
>                 ENDIF
> c             WRITE(99,*)'CT1,CT2,CT3 ',CT1,CT2,CT3
> c             WRITE(99,*)'FIRSTMODEN(CT1,CT2) ',FIRSTMODEN(CT1,CT2)
635a738
>  999 STOP
638c741
<
---
>
1010c1113
<     c MODREFTEMP, RTYPE)
---
>     c MODREFTEMP)
1021a1125
>       IF(RTYPE.EQ.'PWR')THEN
1023a1128,1130
>       ELSE
>         FTFINAL(CT2,CT1)=FTIN(CT2,CT1)
>       ENDIF
1099c1206,1207
<    c       MODREFTEMP, CRMIXDEN)
---
>    c       MODREFTEMP, CRMIXDEN,STOPFLAG,CYCSTOP,STPTSTOP,
>    C       PICKUPFLAG,FTNDES,GDROD,FIRSTMODEN)
1104c1212
<    c       CRMIXNUM, CRMIXID(25), CRNUMISOS(25), CRISOID(25,10),
---
>    c       CRMIXNUM, CRMIXID(25), CRNUMISOS(25), CRISOID(25,20),
1117,1118c1225,1227
<    c       BPFISOID(25,10), ABOVEBPNUM(10), APSRFM(15,10),
<    c       APSRFOLLOWMIX(10,20,23,50), APSRINSOLD(10,20,23,50)
---
>    c       BPFISOID(25,10), ABOVEBPNUM(10), APSRFM(15,10),GDROD,
>    c       APSRFOLLOWMIX(10,20,23,50), APSRINSOLD(10,20,23,50),
>    C       CYCSTOP,STPTSTOP
1124c1233
<    c       CRISOWTPCT(25,10), APSRDEN(10), APSRISOWTPCT(25,10),
---
>    c       CRISOWTPCT(25,20), APSRDEN(10), APSRISOWTPCT(25,10),
1129c1238
<    c       BPRAISOVALUE(2), LEFTVAL(1000), CLTEMP,
```

```
---
>   c     BPRAISOVALUE(10), LEFTVAL(1000), CLTEMP,
1133c1242,1243
<   c     MODREFDEN, MODREFTEMP, CRMIXDEN(25)
---
>   c     MODREFDEN, MODREFTEMP, CRMIXDEN(25),FTNDES(50,2,20),
>   C      FIRSTMODEN(20,20)
1135c1245
<     CHARACTER PREFIX*3, NM*31, CYCLEID(10)*2, REACT*21, LIB*15,
---
>     CHARACTER PREFIX*3, NM*31, CYCLEID(10)*2, REACT*23, LIB*15,
1139c1249
<   c     BPRAISONAME(2)*6, LEFTLIST(1000)*6,
---
>   c     BPRAISONAME(12)*6, LEFTLIST(1000)*6,STOPFLAG*1,
1141c1251
<   c     ABOVEBP(10)*5, RTYPE*3
---
>   c     ABOVEBP(10)*5, RTYPE*3,PICKUPFLAG*1
1167c1277,1282
<         IF ((CT1.EQ.1).AND.(CT2.EQ.1)) THEN
---
> c       this condition stops craft if a RSTART case will be needed
>         IF ((STOPFLAG.EQ.'Y').AND.(CT1.EQ.CYCSTOP).AND.
>   C      (CT2.EQ.STPTSTOP))THEN
>           GOTO 999
>         ENDIF
>         IF ((CT1.EQ.1).AND.(CT2.EQ.1)) THEN
1197c1312,1313
<   c     CT2GOVALUE, APSRINSOLD, RTYPE, MODREFDEN, CRMIXDEN)
---
>   c     CT2GOVALUE, APSRINSOLD, RTYPE, MODREFDEN, CRMIXDEN,
>   C      FIRSTMODEN)
1236c1352,1354
<   c     CT2GOVALUE, APSRINSOLD, RTYPE, MODREFDEN, CRMIXDEN)
---
>   c     CT2GOVALUE, APSRINSOLD, RTYPE, MODREFDEN, CRMIXDEN,
>   C      PICKUPFLAG,FTNDES,GDROD,STOPFLAG,FIRSTMODEN)
>
1250c1368
<     RETURN
---
>  999 RETURN
1284c1402
<   c APSRINSOLD, RTYPE, MODREFDEN, CRMIXDEN)
---
>   c APSRINSOLD, RTYPE, MODREFDEN, CRMIXDEN,FIRSTMODEN)
1291c1409
<   c CRISOID(25,10), APSRINS(10,20,23,50), APSR_MIXTURE_ID,
---
>   c CRISOID(25,20), APSRINS(10,20,23,50), APSR_MIXTURE_ID,
1312c1430
<   c CRISOWTPCT(25,10), APSRDEN(10), APSRISOWTPCT(25,10),
---
>   c CRISOWTPCT(25,20), APSRDEN(10), APSRISOWTPCT(25,10),
1320c1438
<   c CRMIXDEN(25)
```

```
---
>     c CRMIXDEN(25),FIRSTMODEN(20,20)
1323c1441
<     c CHSTPT3*1, NM*31, CYCLEID(10)*2, REACT*21, LIB*15,
---
>     c CHSTPT3*1, NM*31, CYCLEID(10)*2, REACT*23, LIB*15,
1328c1446,1447
<     c SPACERMAT*7, STEPCONTROL*1, ABOVEBP(10)*5, RTYPE*3
---
>     c SPACERMAT*7, STEPCONTROL*1, ABOVEBP(10)*5, RTYPE*3, CTIME*8,
>     c CDATE*9, VERS*9
1331c1450,1451
<     c APSRCOMPFLAG, BPRA_FOLLOW, APSRBOTFLAG, FOLLOWIN
---
>     c APSRCOMPFLAG, BPRA_FOLLOW, APSRBOTFLAG, FOLLOWIN,
>     c UO2FLAG(25)
1394c1514
<  20      FORMAT (A21,1X,'Assy-',A2,
---
>  20      FORMAT (A23,1X,'Assy-',A2,
1402c1522
<  25      FORMAT (A21,1X,'Assy-',A2,
---
>  25      FORMAT (A23,1X,'Assy-',A2,
1411,1413c1531,1543
<        WRITE (100,50)
<  50   FORMAT ('''   fuel density based on mass of uranium per',
<     c  ' assembly',T56,'& total pellet stack')
---
>        CALL DATE(CDATE)
>        CALL TIME(CTIME)
>        VERS='CRAFT V4C'
>        WRITE (100,48)
>  48   FORMAT ('''',T5,'*******************************')
>        WRITE (100,49)
>  49   FORMAT ('''',T5,'this file created by:')
>        WRITE (100,50)VERS,CDATE,CTIME
>  50   FORMAT ('''',T5,A9,2X,A9,2X,A8)
>        WRITE (100,51)
>  51   FORMAT ('''',T5,'*******************************')
>        WRITE (100,52)
>  52   FORMAT ('''')
1415,1416c1545,1546
<  60   FORMAT ('''   volume to account for fuel volume loss to',
<     c  ' pellet c',T55,'hamfers')
---
>  60   FORMAT ('''   fuel density based on mass of uranium per',
>     c  ' node')
1446c1576,1579
<           FVOL=(PI/4)*(FOD**2)*(NODES(CT3,2))*(RODS)
---
> c modification here to calculate proper density using CLADID rather than
> c FOD because gap is removed from pathA model
> c         FVOL=(PI/4)*(FOD**2)*(NODES(CT3,2))*(RODS)
>           FVOL=(PI/4)*(CID**2)*(NODES(CT3,2))*(RODS)
1457,1459c1590,1592
<  100     FORMAT ('uo2 1 den=',F5.3,1X,'1',1X,F6.1,1X,'92234',1X,F5.3,
```

```
<    c    1X,'92235',1X,F5.3,1X,'92236',1X,F5.3,1X,'92238',1X,F6.3,1X,
<    c    'end')
---
>  100     FORMAT ('uo2 1 den=',F6.3,1X,'1',1X,F6.1,1X,'92234',
>    c    1X,F5.4,1X,'92235',1X,F6.4,1X,'92236',1X,F5.4,1X,
>    c    '92238',1X,F6.3,1X,'end')
1463,1465c1596,1598
<  110     FORMAT ('uo2 1 den=',F6.3,1X,'1',1X,F6.1,1X,'92234',1X,F5.3,
<    c    1X,'92235',1X,F5.3,1X,'92236',1X,F5.3,1X,'92238',1X,F6.3,1X,
<    c    'end')
---
>  110     FORMAT ('uo2 1 den=',F6.3,1X,'1',1X,F6.1,1X,'92234',
>    c    1X,F5.4,1X,'92235',1X,F6.4,1X,'92236',1X,F5.4,1X,
>    c    '92238',1X,F6.3,1X,'end')
1612c1745,1749
<          UCMODREGIONDEN=(MODREFDEN*
---
> C         UCMODREGIONDEN=(MODREFDEN*
> C    c    BORATEDMODVF)+(6.56*UCSPACERFRAC)
> c         write(99,*)'*****CT1,CT2 ',CT1,CT2
> c         write(99,*)'**FIRSTMODEN(CT1,CT2) ',FIRSTMODEN(CT1,CT2)
>          UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
1643c1780,1781
<  569     FORMAT ('''    with smeared inconel spacer grids')
---
> c 569     FORMAT ('''    with smeared inconel spacer grids')
>  569     FORMAT ('''    without smeared inconel spacer grids')
1648c1786,1790
<          UCMODREGIONDEN=(MODREFDEN*
---
> C         UCMODREGIONDEN=(MODREFDEN*
> C    c    BORATEDMODVF)+(8.3*UCSPACERFRAC)
> c         write(99,*)'*****CT1,CT2 ',CT1,CT2
> c         write(99,*)'**FIRSTMODEN(CT1,CT2) ',FIRSTMODEN(CT1,CT2)
>          UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
1654c1796
<  570      FORMAT ('h2o   3   den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
---
>  570      FORMAT ('h2o   3   den=',F5.4,3X,G14.6,3X,F7.1,3X,'end')
1658c1800
<  571      FORMAT ('h2o   3   den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
---
>  571      FORMAT ('h2o   3   den=',F6.4,3X,G14.6,3X,F7.1,3X,'end')
1671c1813
<  574     FORMAT (T17'26000 7.0 28000 73.0 3',1X,F6.5,1X,
---
>  574     FORMAT (T17'26000 7.0 28000 73.0 3',1X,G14.6,1X,
1684c1826,1828
<          UCMODREGIONDEN=(MODREFDEN*
---
> C         UCMODREGIONDEN=(MODREFDEN*
> C    c    BORATEDMODVF)+(7.75*UCSPACERFRAC)
>          UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
1720c1864,1866
<          UCMODREGIONDEN=(MODREFDEN*
---
> C         UCMODREGIONDEN=(MODREFDEN*
```

```
> C    c      BORATEDMODVF)+(7.75*UCSPACERFRAC)
>            UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
1756c1902,1904
<            UCMODREGIONDEN=(MODREFDEN*
---
> C           UCMODREGIONDEN=(MODREFDEN*
> C    c      BORATEDMODVF)+(7.92*UCSPACERFRAC)
>            UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
1791c1939,1941
<            UCMODREGIONDEN=(MODREFDEN*
---
> C           UCMODREGIONDEN=(MODREFDEN*
> C    c      BORATEDMODVF)+(7.92*UCSPACERFRAC)
>            UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
1825c1975,1977
<            UCMODREGIONDEN=(MODREFDEN*
---
> C           UCMODREGIONDEN=(MODREFDEN*
> C    c      BORATEDMODVF)
>            UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
2152a2305
> * ***********************************************************************************
2154c2307,2310
<          CRCOMPFLAG=.FALSE.
---
> c         CRCOMPFLAG=.FALSE.
>          CRCOMPFLAG=.TRUE.
>          CR_INSERTED=.TRUE.
>          CR_DESCRIPTION=1
2159c2315,2316
<             CR_DESCRIPTION=CRDES(CT1,CT2,CT4,CT3)
---
> C            CR_DESCRIPTION=CRDES(CT1,CT2,CT4,CT3)
>             CR_DESCRIPTION=1
2167c2324,2325
< 1520       FORMAT ("",T5,' control blade material specifications')
---
> 1520       FORMAT ("",T5,' GDR/CRB/MOD material',
>    &           ' specifications')
2214a2373,2383
>          DO 1672 RELATIVE_CR_MIX_ID=1,CRMIXNUM
>            DO 1671 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
>              UO2FLAG(RELATIVE_CR_MIX_ID)=.FALSE.
> c             IF (CRISOID(RELATIVE_CR_MIX_ID,CT4)
> c    c          .EQ.'92235') THEN
> c               UO2FLAG(RELATIVE_CR_MIX_ID)=.TRUE.
> c             ELSE
> c               UO2FLAG(RELATIVE_CR_MIX_ID)=.FALSE.
> c             ENDIF
> 1671       CONTINUE
> 1672     CONTINUE
2216,2237c2385,2461
<          IF (RELATIVE_CR_MIX_ID.LT.(10)) THEN
<            WRITE (100,*) 'arbm-cr',I1,3X,
<    c       CRMIXDEN(RELATIVE_CR_MIX_ID),
<    c       ' ', CRNUMISOS(RELATIVE_CR_MIX_ID),'  0  0  0'
<            DO 1690 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
```

```
<           WRITE (100,1680) CRISOID(RELATIVE_CR_MIX_ID,CT4),
<    c       CRISOWTPCT(RELATIVE_CR_MIX_ID, CT4)
< 1680         FORMAT (10X,I5,3X,F10.5)
< 1690       CONTINUE
<         WRITE (100,*) '        ', CRMIXID(RELATIVE_CR_MIX),
<    c     ' 1.0 ', MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '  end'
<        ELSEIF (RELATIVE_CR_MIX_ID.EQ.(10)) THEN
<         WRITE (100,*) 'arbm-cr',I2,3X,
<    c     CRMIXDEN(RELATIVE_CR_MIX_ID),
<    c     '  ', CRNUMISOS(RELATIVE_CR_MIX_ID), '  0  0  0'
<         DO 1710 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
<           WRITE (100,1700) CRISOID(RELATIVE_CR_MIX_ID,CT4),
<    c       CRISOWTPCT(RELATIVE_CR_MIX_ID, CT4)
< 1700         FORMAT (10X,I5,3X,F10.5)
< 1710       CONTINUE
<         WRITE (100,*) '        ', CRMIXID(RELATIVE_CR_MIX),
<    c     ' 1.0 ', MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '  end'
---
>        IF (UO2FLAG(RELATIVE_CR_MIX_ID).EQ..TRUE.) THEN
>          IF (FDEN.LT.(10.0)) THEN
>            WRITE (100,1673) RELATIVE_CR_MIX_ID,
>    c        CRMIXDEN(RELATIVE_CR_MIX_ID),
>    c        FTFINAL(CT3,RELATIVE_STPT_NUM), WT234,
>    c        WT235, WT236, WT238
> 1673          FORMAT ('uo2',1X,I2,1X,'den=',F6.3,1X,'1',1X,
>    c        F6.1,1X,'92234',
>    c        1X,F5.4,1X,'92235',1X,F6.4,1X,'92236',
>    c        1X,F5.4,1X,
>    c        '92238',1X,F6.3,1X,'end')
>          ELSE
>            WRITE (100,1674) RELATIVE_CR_MIX_ID,
>    c        CRMIXDEN(RELATIVE_CR_MIX_ID),
>    c        FTFINAL(CT3,RELATIVE_STPT_NUM), WT234,
>    c        WT235, WT236, WT238
> 1674          FORMAT ('uo2',1X,I2,1X,'den=',F6.3,1X,'1',1X,
>    c        F6.1,1X,'92234',
>    c        1X,F5.4,1X,'92235',1X,F6.4,1X,'92236',
>    c        1X,F5.4,1X,
>    c        '92238',1X,F6.3,1X,'end')
>          ENDIF
>          IF (RELATIVE_CR_MIX_ID.LT.10) THEN
>            WRITE (100,1675) RELATIVE_CR_MIX_ID,
>    c        CRMIXDEN(RELATIVE_CR_MIX_ID),
>    c        CRNUMISOS(RELATIVE_CR_MIX_ID)
> c 1675          FORMAT(T1,'arbm-cr',I1,3X,
> 1675          FORMAT(T1,'arbm-gcm',I1,3X,
>    c        G14.8,3X,I2,'  0  0  0')
>          ELSEIF (RELATIVE_CR_MIX_ID.EQ.10) THEN
>            WRITE (100,1676) RELATIVE_CR_MIX_ID,
>    c        CRMIXDEN(RELATIVE_CR_MIX_ID),
>    c        CRNUMISOS(RELATIVE_CR_MIX_ID)
> c 1676          FORMAT(T1,'arbm-cr',I2,3X,
> 1676          FORMAT(T1,'arbm-gcm',I2,3X,
>    c        G14.8,3X,I2,'  0  0  0')
>          ENDIF
>          DO 1678 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
> c             IF ((CRISOID(RELATIVE_CR_MIX_ID,CT4).NE.'92234').AND.
```

```
> c   c        (CRISOID(RELATIVE_CR_MIX_ID,CT4).NE.'92235').AND.
> c   c        (CRISOID(RELATIVE_CR_MIX_ID,CT4).NE.'92236').AND.
> c   c        (CRISOID(RELATIVE_CR_MIX_ID,CT4).NE.'92238')) THEN
>              WRITE (100,1677) CRISOID(RELATIVE_CR_MIX_ID,CT4),
>     c         CRISOWTPCT(RELATIVE_CR_MIX_ID, CT4)
> 1677            FORMAT (10X,I5,3X,F10.5)
> c            ENDIF
> 1678        CONTINUE
>           WRITE (100,*) '       ', CRMIXID(RELATIVE_CR_MIX_ID),
>     c      ' 0.03 ', MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '   end'
>         ELSE
>           IF (RELATIVE_CR_MIX_ID.LT.10) THEN
>             WRITE (100,1679) RELATIVE_CR_MIX_ID,
>     c        CRMIXDEN(RELATIVE_CR_MIX_ID),
>     c        CRNUMISOS(RELATIVE_CR_MIX_ID)
> c 1679          FORMAT(T1,'arbm-cr',I1,3X,
> 1679          FORMAT(T1,'arbm-gcm',I1,3X,
>     c        G14.8,3X,I2,'  0  0  0')
>           ELSEIF (RELATIVE_CR_MIX_ID.EQ.10) THEN
>             WRITE (100,1680) RELATIVE_CR_MIX_ID,
>     c        CRMIXDEN(RELATIVE_CR_MIX_ID),
>     c        CRNUMISOS(RELATIVE_CR_MIX_ID)
> c 1680          FORMAT(T1,'arbm-cr',I2,3X,
> 1680          FORMAT(T1,'arbm-gcm',I2,3X,
>     c        G14.8,3X,I2,'  0  0  0')
>           ENDIF
>           DO 1682 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
>             WRITE (100,1681) CRISOID(RELATIVE_CR_MIX_ID,CT4),
>     c        CRISOWTPCT(RELATIVE_CR_MIX_ID, CT4)
> 1681          FORMAT (10X,I5,3X,F10.5)
> 1682        CONTINUE
>          IF(RELATIVE_CR_MIX_ID.EQ.1)THEN
>           WRITE (100,*) '       ',CRMIXID(RELATIVE_CR_MIX_ID),
>     c      ' 1.0 ', FTFINAL(CT3,RELATIVE_STPT_NUM), '   end'
>          ELSE
>           WRITE (100,*) '       ',CRMIXID(RELATIVE_CR_MIX_ID),
>     c      ' 1.0 ', MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '   end'
>          ENDIF
2241a2466
>       ENDIF
2424,2425c2649,2652
<       WRITE (100,920)
<  920   FORMAT ('he    5   end')
---
>       IF(RTYPE.EQ.'PWR')THEN
>        WRITE (100,920)
>  920   FORMAT ('he    5   end')
>       ENDIF
2435c2662
<       WRITE (100,970) PITCH, FOD, COD, CID
---
>       WRITE (100,970) PITCH, FOD, COD
2437c2664,2667
<     c  3X,'2',3X,F6.4,3X,'0   end')
---
>     c  3X,'2',' end')
> c      WRITE (100,970) PITCH, FOD, COD, CID
```

```
> c 970   FORMAT ('squarepitch',3X,F7.5,3X,F6.4,3X,'1  3',3X,F6.4,
> c   c  3X,'2',3X,F6.4,3X,'0  end')
2786c3016
< 1350   FORMAT ('''',T5,'Cycle-',A2,', one-eighth core',
---
> 1350   FORMAT ('''',T5,'Cycle-',A2,', ',
2841c3071
<    c       G10.5,1X,'bfrac=',G9.4,1X,'end')
---
>    c       G10.5,1X,'h2ofrac=',G9.4,1X,'end')
2856a3087,3088
> C          BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
> C    c     MODREFDEN)
2858c3090
<    c       MODREFDEN)
---
>    c       FIRSTMODEN(CT1,CT2))
2862c3094
<    c       G10.5,1X,'h2ofrac=',G9.4,1X,'end')
---
>    c       G10.5,1X,'h2ofrac=',G10.5,1X,'end')
2874a3107,3108
> C          BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
> C    c     MODREFDEN)
2876c3110
<    c       MODREFDEN)
---
>    c       FIRSTMODEN(CT1,CT2))
2880c3114
<    c       G10.5,1X,'h2ofrac=',G9.4,1X,'end')
---
>    c       G10.5,1X,'h2ofrac=',G10.5,1X,'end')
2892a3127,3128
> C          BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
> C    c     MODREFDEN)
2894,2895c3130,3131
<    c       MODREFDEN)
<           WRITE (100,1500) VARPOWER(CT1,CT2,CT4,CT3),
---
>    c       FIRSTMODEN(CT1,CT2))
>           WRITE (100,1800) VARPOWER(CT1,CT2,CT4,CT3),
2897,2898c3133,3134
< 1500         FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
<    c       G10.5,1X,'h2ofrac=',G9.4,1X,'end')
---
> 1800         FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
>    c       G10.5,1X,'h2ofrac=',G10.5,1X,'end')
2954c3190,3191
<    c APSRINSOLD, RTYPE, MODREFDEN, CRMIXDEN)
---
>    c APSRINSOLD, RTYPE, MODREFDEN, CRMIXDEN,PICKUPFLAG,FTNDES,
>    C GDROD,STOPFLAG,FIRSTMODEN)
2961c3198
<    c CRISOID(25,10), APSRINS(10,20,23,50), APSR_MIXTURE_ID,
---
>    c CRISOID(25,20), APSRINS(10,20,23,50), APSR_MIXTURE_ID,
2974c3211
```

```
<     c BPFMNUMISOS(25), BPFISOID(25,10), ABOVEBPNUM(10),
---
>     c BPFMNUMISOS(25), BPFISOID(25,10), ABOVEBPNUM(10),GDROD,
2984c3221
<     c CRISOWTPCT(25,10), APSRDEN(10), APSRISOWTPCT(25,10),
---
>     c CRISOWTPCT(25,20), APSRDEN(10), APSRISOWTPCT(25,10),
2988c3225
<     c MASSTOTAL, FUELISOWTPCT(1000), BPRAISOVALUE(2), BPXSECT(10),
---
>     c MASSTOTAL, FUELISOWTPCT(1000), BPRAISOVALUE(10), BPXSECT(10),
2995c3232,3234
<     c CRMIXDEN(25)
---
>     c CRMIXDEN(25),FTNDES(50,2,20),PRODMASS,
>     C NEWVALWTPCT(15),NEWDEN,FIRSTMODEN(20,20),
>     C FMASS1,FMASS2,FMASS3,FMASS4,FUELMASS,GDMASS,NONZERO,GDROMASS
2998c3237
<     c CHSTPT3*1, NM*31, CYCLEID(10)*2, REACT*21, LIB*15,
---
>     c CHSTPT3*1, NM*31, CYCLEID(10)*2, REACT*23, LIB*15,
3001c3240
<     c APSRZONECH*2, LUZONECH*2, FUELISONAME(1000)*5, BPRAISONAME(2)*6,
---
>     c APSRZONECH*2, LUZONECH*2, FUELISONAME(1000)*5, BPRAISONAME(12)*6,
3005c3244,3245
<     c SPACERMAT*7, STEPCONTROL*1, ABOVEBP(10)*5, RTYPE*3
---
>     c SPACERMAT*7, STEPCONTROL*1, ABOVEBP(10)*5, RTYPE*3,PICKUPFLAG*1,
>     C STOPFLAG*1,CTIME*8,CDATE*9,VERS*9
3128c3368
< 20      FORMAT (A21,1X,'Assy-',A2,
---
> 20      FORMAT (A23,1X,'Assy-',A2,
3136c3376
< 25      FORMAT (A21,1X,'Assy-',A2,
---
> 25      FORMAT (A23,1X,'Assy-',A2,
3145,3147c3385,3397
<       WRITE (100,50)
< 50    FORMAT ('"   fuel density based on mass of uranium per',
<     c   ' assembly',T56,'& total pellet stack')
---
>       CALL DATE(CDATE)
>       CALL TIME(CTIME)
>       VERS='CRAFT V4C'
>       WRITE (100,48)
> 48    FORMAT ('"',T5,'*******************************')
>       WRITE (100,49)
> 49    FORMAT ('"',T5,'this file created by:')
>       WRITE (100,50)VERS,CDATE,CTIME
> 50    FORMAT ('"',T5,A9,2X,A9,2X,A8)
>       WRITE (100,51)
> 51    FORMAT ('"',T5,'*******************************')
>       WRITE (100,52)
> 52    FORMAT ('"')
3149,3150c3399,3400
```

```
<  60   FORMAT ('"  volume to account for fuel volume loss to',
<    c  ' pellet c',T55,'hamfers')
---
>  60   FORMAT ('"  fuel density based on mass of uranium per',
>    c  ' node')
3175a3426,3429
>       IF((RTYPE.EQ.'BWR').AND.(GDROD.NE.0))THEN
>        BPRA_INSERTED=.TRUE.
>       ENDIF
> C  always looks for BPRA-gadolinia isostopics-no restart
3181c3435,3438
<       FVOL=(PI/4)*(FOD**2)*(NODES(CT3,2))*(RODS)
---
> c  modification here to calculate proper density using CLADID rather than
> c  FOD because gap is removed from pathA model
> c        FVOL=(PI/4)*(FOD**2)*(NODES(CT3,2))*(RODS)
>       FVOL=(PI/4)*(CID**2)*(NODES(CT3,2))*(RODS)
3266c3523,3525
<            UCMODREGIONDEN=(MODREFDEN*
---
> C             UCMODREGIONDEN=(MODREFDEN*
> C     c       BORATEDMODVF)+(6.56*UCSPACERFRAC)
>            UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
3297c3556,3557
<  569      FORMAT ('"   with smeared inconel spacer grids')
---
> c  569     FORMAT ('"   with smeared inconel spacer grids')
>  569      FORMAT ('"   without smeared inconel spacer grids')
3302c3562,3564
<            UCMODREGIONDEN=(MODREFDEN*
---
> C             UCMODREGIONDEN=(MODREFDEN*
> C     c       BORATEDMODVF)+(8.3*UCSPACERFRAC)
>            UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
3308c3570
<  570        FORMAT ('h2o  3  den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
---
>  570        FORMAT ('h2o  3  den=',F5.4,3X,G14.8,3X,F7.1,3X,'end')
3312c3574
<  571        FORMAT ('h2o  3  den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
---
>  571        FORMAT ('h2o  3  den=',F6.4,3X,G14.8,3X,F7.1,3X,'end')
3325c3587
<  574      FORMAT (T17'26000 7.0 28000 73.0 3',1X,F6.5,1X,
---
>  574      FORMAT (T17'26000 7.0 28000 73.0 3',1X,G14.6,1X,
3338c3600,3602
<            UCMODREGIONDEN=(MODREFDEN*
---
> C             UCMODREGIONDEN=(MODREFDEN*
> C     c       BORATEDMODVF)+(7.75*UCSPACERFRAC)
>            UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
3374c3638,3640
<            UCMODREGIONDEN=(MODREFDEN*
---
> C             UCMODREGIONDEN=(MODREFDEN*
> C     c       BORATEDMODVF)+(7.75*UCSPACERFRAC)
```

```
>          UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
3410c3676,3678
<          UCMODREGIONDEN=(MODREFDEN*
---
> C         UCMODREGIONDEN=(MODREFDEN*
> C   c     BORATEDMODVF)+(7.92*UCSPACERFRAC)
>          UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
3445c3713,3715
<          UCMODREGIONDEN=(MODREFDEN*
---
> C         UCMODREGIONDEN=(MODREFDEN*
> C   c     BORATEDMODVF)+(7.92*UCSPACERFRAC)
>          UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
3479c3749,3751
<          UCMODREGIONDEN=(MODREFDEN*
---
> C         UCMODREGIONDEN=(MODREFDEN*
> C   c     BORATEDMODVF)
>          UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
3815a4088
>
3820,3821c4093,4096
<        CRCOMPFLAG=.FALSE.
<        DO 760 CT4=1,23
---
> c       CRCOMPFLAG=.FALSE.
>        CRCOMPFLAG=.TRUE.
>        CR_INSERTED=.TRUE.
>       DO 760 CT4=1,23
3896a4172,4173
> ****************************************************************************************
> * this is where arbitrary material specifications are written
3898c4175,4178
<        CRCOMPFLAG=.FALSE.
---
> C       CRCOMPFLAG=.FALSE.
>        CRCOMPFLAG=.TRUE.
>        CR_INSERTED=.TRUE.
>        CR_DESCRIPTION=1
3903c4183,4184
<           CR_DESCRIPTION=CRDES(CT1,CT2,CT4,CT3)
---
> C          CR_DESCRIPTION=CRDES(CT1,CT2,CT4,CT3)
>           CR_DESCRIPTION=1
3911c4192,4193
< 1520      FORMAT ("",T5,' control blade material specifications')
---
> 1520      FORMAT ("",T5,' GDR/CRB/MOD material',
>   &            ' specifications')
3959,3961c4241,4345
<       DO 1720 RELATIVE_CR_MIX_ID=1,CRMIXNUM
<        IF (RELATIVE_CR_MIX_ID.LT.(10)) THEN
<         WRITE (100,*) 'arbm-cr',I1,3X,
---
> * ***********************************************************************
> * this is where the CB specification must change for the pickup case
> * in the continuation writer
```

```
> * this calculation of depleted Gd isotopes is VERY specific to CRMIX #1
> * 12 isotope entries
>
>        IF(GDROD.NE.0)THEN
> C this is case where gadolinina is burning out-the last time this
> C section runs the fuel is considered fresh-gd has burned away
>
>        FUELMASS=MASSTOTAL/RODS*GDROD
>        GDMASS=BPRAISOVALUE(1)+BPRAISOVALUE(3)+
>    &    BPRAISOVALUE(4)+BPRAISOVALUE(5)+BPRAISOVALUE(6)+
>    &    BPRAISOVALUE(7)+BPRAISOVALUE(9)
>        GDROMASS=(OXYGMS/RODS)*GDROD
>        DO 3130 CT4=1,CARRYCOUNTER
>         IF(FUELISONAME(CT4).EQ.'92234')THEN
>          FMASS1=FUELMASS*FUELISOWTPCT(CT4)*0.01
>         ELSEIF(FUELISONAME(CT4).EQ.'92235')THEN
>          FMASS2=FUELMASS*FUELISOWTPCT(CT4)*0.01
>         ELSEIF(FUELISONAME(CT4).EQ.'92236')THEN
>          FMASS3=FUELMASS*FUELISOWTPCT(CT4)*0.01
>         ELSEIF(FUELISONAME(CT4).EQ.'92238')THEN
>          FMASS4=FUELMASS*FUELISOWTPCT(CT4)*0.01
>         ENDIF
> 3130     CONTINUE
>        GDROMASS=(OXYGMS/RODS)*GDROD
>        PRODMASS=FMASS1+FMASS2+FMASS3+FMASS4+GDMASS+GDROMASS
>        NEWVALWTPCT(1)=FMASS1/PRODMASS*100
>        NEWVALWTPCT(2)=FMASS2/PRODMASS*100
>        NEWVALWTPCT(3)=FMASS3/PRODMASS*100
>        NEWVALWTPCT(4)=FMASS4/PRODMASS*100
>        NEWVALWTPCT(5)=BPRAISOVALUE(1)/PRODMASS*100
>        NEWVALWTPCT(6)=BPRAISOVALUE(3)/PRODMASS*100
>        NEWVALWTPCT(7)=BPRAISOVALUE(4)/PRODMASS*100
>        NEWVALWTPCT(8)=BPRAISOVALUE(5)/PRODMASS*100
>        NEWVALWTPCT(9)=BPRAISOVALUE(6)/PRODMASS*100
>        NEWVALWTPCT(10)=BPRAISOVALUE(7)/PRODMASS*100
>        NEWVALWTPCT(11)=BPRAISOVALUE(9)/PRODMASS*100
>        NEWVALWTPCT(12)=GDROMASS/PRODMASS*100
>
>        NONZERO=0.00001
>        NEWDEN=PRODMASS/(GDROD*3.141593*((CID/2)**2)
>    &    *NODES(AXNUM,2))
>        IF(NEWDEN.LT.CRMIXDEN(1))THEN
>         CRMIXDEN(1)=NEWDEN
>        ENDIF
>        RELATIVE_CR_MIX_ID=1
>        WRITE (100,2672) RELATIVE_CR_MIX_ID,
>    c   CRMIXDEN(RELATIVE_CR_MIX_ID),
>    c   CRNUMISOS(RELATIVE_CR_MIX_ID)
> c 2672     FORMAT(T1,'arbm-cr',I1,3X,
> 2672     FORMAT(T1,'arbm-gcm',I1,3X,
>    c   G14.8,3X,I2,'  0  0  0')
>        DO 2690 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
>        IF(NEWVALWTPCT(CT4).GE.0.00001)THEN
>         WRITE (100,2680) CRISOID(RELATIVE_CR_MIX_ID,CT4),
>    c    NEWVALWTPCT(CT4)
> 2680      FORMAT (10X,I5,3X,F10.5)
>        ELSE
```

```
>           WRITE (100,2680) CRISOID(RELATIVE_CR_MIX_ID,CT4),
>     c      NONZERO
>          ENDIF
> 2690      CONTINUE
>         IF(RELATIVE_CR_MIX_ID.EQ.1)THEN
>          WRITE (100,*) '        ',CRMIXID(RELATIVE_CR_MIX_ID),
>     c     ' 1.0 ', FTFINAL(CT3,RELATIVE_STPT_NUM), '  end'
>         ELSE
>          WRITE (100,*) '        ',CRMIXID(RELATIVE_CR_MIX_ID),
>     c     ' 1.0 ', MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '  end'
>         ENDIF
>        ENDIF
>
>
> C          NEWDEN=NEWDMASS/GDROD
> C          IF(NEWDEN.LT.CRMIXDEN(1))THEN
> C           CRMIXDEN(1)=NEWDMASS/GDROD
> C          ENDIF
> c          write(99,*)'CRMIXDEN',CRMIXDEN(1)
> C          RELATIVE_CR_MIX_ID=1
> C          WRITE (100,3672) RELATIVE_CR_MIX_ID,
> C    c     CRMIXDEN(RELATIVE_CR_MIX_ID),
> C    c     CRNUMISOS(RELATIVE_CR_MIX_ID)
> C 3672      FORMAT(T1,'arbm-cr',I1,3X,
> C    c     G14.8,3X,I2,'  0  0  0')
> C          DO 3690 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
> C            WRITE (100,3680) CRISOID(RELATIVE_CR_MIX_ID,CT4),
> C    c       NEWVALWTPCT(CT4)
> C 3680        FORMAT (10X,I5,3X,F10.5)
> C 3690      CONTINUE
> C          IF(RELATIVE_CR_MIX_ID.EQ.1)THEN
> C           WRITE (100,*) '        ',CRMIXID(RELATIVE_CR_MIX_ID),
> C    c      ' 1.0 ', FTFINAL(CT3,RELATIVE_STPT_NUM), '  end'
> C          ELSE
> C           WRITE (100,*) '        ',CRMIXID(RELATIVE_CR_MIX_ID),
> C    c      ' 1.0 ', MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '  end'
> C          ENDIF
> C         ENDIF
> C        ENDIF
>
> C this is the case without gd
>
>        DO 1720 RELATIVE_CR_MIX_ID=2,CRMIXNUM
>         IF (RELATIVE_CR_MIX_ID.LT.10) THEN
>          WRITE (100,1672) RELATIVE_CR_MIX_ID,
3963c4347,4358
<    c     '  ', CRNUMISOS(RELATIVE_CR_MIX_ID), '  0  0  0'
---
>    c     CRNUMISOS(RELATIVE_CR_MIX_ID)
> c 1672      FORMAT(T1,'arbm-cr',I1,3X,
> 1672      FORMAT(T1,'arbm-gcm',I1,3X,
>    c     G14.8,3X,I2,'  0  0  0')
>        ELSEIF (RELATIVE_CR_MIX_ID.EQ.10) THEN
>         WRITE (100,1674) RELATIVE_CR_MIX_ID,
>    c     CRMIXDEN(RELATIVE_CR_MIX_ID),
>    c     CRNUMISOS(RELATIVE_CR_MIX_ID)
> c 1674      FORMAT(T1,'arbm-cr',I2,3X,
```

```
> 1674      FORMAT(T1,'arbm-gcm',I2,3X,
>    c      G14.8,3X,I2,'  0  0  0')
>         ENDIF
3969c4364,4368
<          WRITE (100,*) '        ', CRMIXID(RELATIVE_CR_MIX),
---
> C         IF(RELATIVE_CR_MIX_ID.EQ.1)THEN
> C          WRITE (100,*) '        ',CRMIXID(RELATIVE_CR_MIX_ID),
> C    c      ' 1.0 ', FTFINAL(CT3,RELATIVE_STPT_NUM), '   end'
> C         ELSE
>          WRITE (100,*) '        ',CRMIXID(RELATIVE_CR_MIX_ID),
3971,3982c4370
<         ELSEIF (RELATIVE_CR_MIX_ID.EQ.(10)) THEN
<          WRITE (100,*) 'arbm-cr',I2,3X,
<    c      CRMIXDEN(RELATIVE_CR_MIX_ID),
<    c      '  ', CRNUMISOS(RELATIVE_CR_MIX_ID), '  0  0  0'
<          DO 1710 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
<            WRITE (100,1700) CRISOID(RELATIVE_CR_MIX_ID,CT4),
<    c        CRISOWTPCT(RELATIVE_CR_MIX_ID, CT4)
< 1700         FORMAT (10X,I5,3X,F10.5)
< 1710       CONTINUE
<          WRITE (100,*) '        ', CRMIXID(RELATIVE_CR_MIX),
<    c      ' 1.0 ', MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '   end'
<         ENDIF
---
> C         ENDIF
3985a4374
>       ENDIF
4162c4551
<    c      MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '   end'
---
>    c      FTFINAL(CT3,RELATIVE_STPT_NUM), '   end'
4168,4169c4557,4560
<       WRITE (100,920)
< 920   FORMAT ('he    5   end')
---
>       IF(RTYPE.EQ.'PWR')THEN
>        WRITE (100,920)
> 920    FORMAT ('he    5   end')
>       ENDIF
4179c4570,4573
<       WRITE (100,970) PITCH, FOD, COD, CID
---
> c      WRITE (100,970) PITCH, FOD, COD, CID
> c 970   FORMAT ('squarepitch',3X,F7.5,3X,F6.4,3X,'1   3',3X,F6.4,
> c    c  3X,'2',3X,F6.4,3X,'0   end')
>       WRITE (100,970) PITCH, FOD, COD
4181c4575
<    c  3X,'2',3X,F6.4,3X,'0   end')
---
>    c  3X,'2',' end')
4530c4924
< 1350   FORMAT ("",T5,'Cycle-',A2,', one-eighth core',
---
> 1350   FORMAT ("",T5,'Cycle-',A2,', ',
4584a4979,4980
> C            BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
```

```
> C    c        MODREFDEN)
4586c4982
<    c        MODREFDEN)
---
>    c        FIRSTMODEN(CT1,CT2))
4591c4987
<    c      G10.5,1X,'bfrac=',G9.4,1X,'end')
---
>    c      G10.5,1X,'h2ofrac=',G10.5,1X,'end')
4598a4995,4996
> C             BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
> C    c        MODREFDEN)
4600c4998
<    c        MODREFDEN)
---
>    c        FIRSTMODEN(CT1,CT2))
4605c5003
<    c      G10.5,1X,'bfrac=',G9.4,1X,'end')
---
>    c      G10.5,1X,'h2ofrac=',G10.5,1X,'end')
4612a5011,5012
> C             BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
> C    c        MODREFDEN)
4614c5014
<    c        MODREFDEN)
---
>    c        FIRSTMODEN(CT1,CT2))
4619c5019
<    c      G10.5,1X,'bfrac=',G9.4,1X,'end')
---
>    c      G10.5,1X,'h2ofrac=',G10.5,1X,'end')
4886c5286
<     c LEFTCOUNTER, CARRYCOUNTER, CT2, ISOFLAG(1000), Z
---
>     c LEFTCOUNTER,CARRYCOUNTER,CT2,ISOFLAG(1000),Z,CC
4888c5288
<      REAL ISOVALUE(1000), BPRAISOVALUE(2), MASSTOTAL,
---
>      REAL ISOVALUE(1000), BPRAISOVALUE(10), MASSTOTAL,
4895c5295
<     c FISSPRODLABEL*16, BPRAISONAME(2)*6, ORIGNAME(297)*6,
---
>     c FISSPRODLABEL*16, BPRAISONAME(12)*6, ORIGNAME(297)*6,
4898c5298
<     c NOTESFILE*31
---
>     c NOTESFILE*31, GDNAME(12)*6
5076c5476,5489
< *  Get B-10 and B-11 composition data for BPRA
---
> *  Get Gd composition data for Gd-RODS
>      GDNAME(1)=' gd152'
>      GDNAME(2)=' gd153'
>      GDNAME(3)=' gd154'
>      GDNAME(4)=' gd155'
>      GDNAME(5)=' gd156'
>      GDNAME(6)=' gd157'
```

```
>     GDNAME(7)=' gd158'
>     GDNAME(8)=' gd159'
>     GDNAME(9)=' gd160'
>     DO 500 CT1=1,10
>      BPRAISOVALUE(CT1)=0.0
>      BPRAISONAME(CT1)='    '
> 500  CONTINUE
5078,5081c5491
<       BPRAISOVALUE(1)=0.0
<       BPRAISOVALUE(2)=0.0
<       BPRAISONAME(1)='    '
<       BPRAISONAME(2)='    '
---
>      DO 700 CC=1,9
5096,5099c5506,5531
<       DO 24 CT1=1,25
<         READ (300,22) BPRAISONAME(1)
<  22     FORMAT(T6,A6)
<         IF (BPRAISONAME(1).EQ.' b 10 ') THEN
---
> C       DO 24 CT1=1,25
> C         READ (300,22) BPRAISONAME(1)
> C  22     FORMAT(T6,A6)
> C        IF (BPRAISONAME(1).EQ.' b 10 ') THEN
> C           BACKSPACE(300)
> C           EXIT
> C         ENDIF
> C  24   CONTINUE
> C       READ (300,26) BPRAISONAME(1), BPRAISOVALUE(1)
> C  26     FORMAT(T6,A6,T<COLUMNSTART>,G10.2)
> C       READ (300,29) BPRAISONAME(2), BPRAISOVALUE(2)
> C  29     FORMAT(T6,A6,T<COLUMNSTART>,G10.2)
> C       IF (BPRAISONAME(1).EQ.' b 10 ') THEN
> C         BPRAISONAME(1)=' 5010'
> C       ENDIF
> C       IF (BPRAISONAME(2).EQ.' b 11 ') THEN
> C         BPRAISONAME(2)=' 5011'
> C       ENDIF
> C     ENDIF
> * this section retrieves Gd concentrations and calculates
> * new Gd2O3-UO2 concentrations to be in subseqent SAS2H calculation
>  22     FORMAT(T5,A6)
>  26     FORMAT(T5,A6,T<COLUMNSTART>,G10.2)
>       DO 324 CT1=1,65
>         READ (300,22) BPRAISONAME(CC)
>         IF (BPRAISONAME(CC).EQ.GDNAME(CC))THEN
5101c5533,5534
<           EXIT
---
>           READ (300,26) BPRAISONAME(CC), BPRAISOVALUE(CC)
>           GOTO 700
5103,5109c5536,5559
<  24   CONTINUE
<       READ (300,26) BPRAISONAME(1), BPRAISOVALUE(1)
<  26     FORMAT(T6,A6,T<COLUMNSTART>,G10.2)
<       READ (300,29) BPRAISONAME(2), BPRAISOVALUE(2)
<  29     FORMAT(T6,A6,T<COLUMNSTART>,G10.2)
```

```
<       IF (BPRAISONAME(1).EQ.' b 10 ') THEN
<         BPRAISONAME(1)='  5010'
---
> 324   CONTINUE
>       BPRAISONAME(CC)=GDNAME(CC)
>       BPRAISOVALUE(CC)=0.0
> 700   CONTINUE
>
> C       READ (300,26) BPRAISONAME(CC), BPRAISOVALUE(CC)
> C       DO 325 CT1=1,10
> C         READ (300,22) BPRAISONAME(2)
> C         IF (BPRAISONAME(2).EQ.' gd153') THEN
> C           BACKSPACE(300)
> C           EXIT
> C         ENDIF
> C 325   CONTINUE
> C       DO 326 CT1=1,10
> C         READ (300,22) BPRAISONAME(3)
> C         IF (BPRAISONAME(3).EQ.' gd154') THEN
> C           BACKSPACE(300)
> C           EXIT
> C         ENDIF
> C 326   CONTINUE
> C       READ (300,26) BPRAISONAME(3), BPRAISOVALUE(3)
>
>       IF (BPRAISONAME(1).EQ.' gd152') THEN
>         BPRAISONAME(1)='  64152'
5111,5112c5561,5562
<       IF (BPRAISONAME(2).EQ.' b 11 ') THEN
<         BPRAISONAME(2)='  5011'
---
>       IF (BPRAISONAME(2).EQ.' gd153') THEN
>         BPRAISONAME(2)='  64153'
5113a5564,5584
>       IF (BPRAISONAME(3).EQ.' gd154') THEN
>         BPRAISONAME(3)='  64154'
>       ENDIF
>       IF (BPRAISONAME(4).EQ.' gd155') THEN
>         BPRAISONAME(4)='  64155'
>       ENDIF
>       IF (BPRAISONAME(5).EQ.' gd156') THEN
>         BPRAISONAME(5)='  64156'
>       ENDIF
>       IF (BPRAISONAME(6).EQ.' gd157') THEN
>         BPRAISONAME(6)='  64157'
>       ENDIF
>       IF (BPRAISONAME(7).EQ.' gd158') THEN
>         BPRAISONAME(7)='  64158'
>       ENDIF
>       IF (BPRAISONAME(8).EQ.' gd159') THEN
>         BPRAISONAME(8)='  64159'
>       ENDIF
>       IF (BPRAISONAME(9).EQ.' gd160') THEN
>         BPRAISONAME(9)='  64160'
>       ENDIF
5187,5188c5658,5659
<       WRITE (500,*) 'B-10 AND B-11 IN BPRA'
```

```
<       DO 150 CT1=1,2
---
>       WRITE (500,*) 'Gd IN Gd-ROD'
>       DO 150 CT1=1,9
```

The objective of the CRAFT Version 4C methodology was to develop a mechanism by which BWR fuel nodal depletion and decay calculations required to support CRC evaluations could be performed most efficiently with minimal required user interface. The result was the CRAFT software routine which automates the process of performing numerous complex SAS2H depletion and decay calculations while extracting and archiving results pertinent to CRC analyses.

4. Input Summary for CRAFT, Version 4C

The input file for a BWR nodal depletion calculation using CRAFT, Version 4C is described below. These input files are created by the CRAFT driver software routine SPACE. The actual execution of CRAFT is accomplished by the use of a separate script file entitled LIFTOFF that is called by the SPACE software routine. The CRAFT input filename must be "datain". Table 4-1 provides a summary of the CRAFT input file. Table 4-2 shows an example CRAFT input file for an uranium axial blanket node (node 1). Tables 4-3 and 4-4 show CRAFT input files for a non-bladed GDR Path B model (node 5) and a bladed GDR Path B model (node 3). The "Group Number" (GN) heading for column one refers to a related set of input lines and may not correspond to the input line number.

Table 4-1. CRAFT Input Summary

Group Number	Special Notes	Line Format	Line Description
1-4		4 lines, 31 characters/line	Time/date stamp for creation of input file. This information is placed in the input file by SPACE
5		1 Character	Should always be "N"; obsolete RESTART function not used
6		1 Character	Should always be "N"; obsolete PICKUP function not used
7		Integer	Number of gadolinia-bearing fuel rods in node
8		21 Characters, integer	Assembly identifier (i.e., "D9 - no CRB tracking"), node identifier (i.e., 4)
9		3 Characters	Problem prefix to be used as an identifier in all filenames
10		3 Characters	Reactor Type – these characters should always be "BWR"
11		15 Characters	SCALE cross-section library to be utilized by SAS2H
12		Real	wt% U-235 enrichment in UO_2
13		Real	Mass of U per node (g)

Group Number	Special Notes	Line Format	Line Description
14		Real	Number of fuel rods in node
15		Real	Rod pitch in node (cm)
16		Real	Fuel pellet diameter (cm), (for BWR calculations, this value should be fuel rod cladding inner diameter)
17		Real	Fuel rod cladding inner diameter (cm)
18		Real	Fuel rod cladding outer diameter (cm)
19		Real	Node Height (cm)
20		1 Character	Axial blanket fuel, (this character should always be "N")
21		7 Characters	Spacer grid material identification (this character word should always be "INCONEL")
22		Real	Volume fraction of spacer grids in the moderator of the fuel assembly, (this value should always be 0.000001)
23		10 Characters	Fuel cladding material identification (this character word should always be "ZIRC-4")
24		Real	Average fuel cladding temperature (K)
25		1 Character	Indicates if a cladding specification other than Zirc-4 is required, (this character should always be "N")
26		Real	Reference moderator density (g/cm^3)
27		Real	In-channel/bypass moderator temperature (K)
28		1 Character	Activate BPRA Tracking, (this character should always be "N")
29		Integer	Number of radial zones in standard Path B model, (this character should always be 1)
30		Integer, Real	Material mixture number for zone of standard Path B model, Outer radii (cm) for zone of standard Path B model
31		Integer	Number of cross-section libraries to be created per irradiation step
32		Integer	SAS2H print level
33		Real	Zone mesh factor for use by XSDRNPM
34		7 Character	[SPECIAL] to indicate the input of 7 XSDRNPM calculational control parameters to follow, any other character string indicates no XSDRNPM calculational control parameter input
34A	☆	Real	XSDRNPM calculational control parameter: Spatial Mesh Factor (SZF < 1 for finer, SZF > 1 for coarser), Default = 1 {If GN 34 = "SPECIAL"}

Group Number	Special Notes	Line Format	Line Description
34B	✯	Integer	XSDRNPM calculational control parameter: Order of Angular Quadrature, Default = 8 {If GN 34 = "SPECIAL"}
34C	✯	Integer	XSDRNPM calculational control parameter: Maximum Number of Inner Iterations, Default = 20 {If GN 34 = "SPECIAL"}
34D	✯	Integer	XSDRNPM calculational control parameter: Maximum Number of Outer Iterations, Default = 25 {If GN 34 = "SPECIAL"}
34E	✯	Real	XSDRNPM calculational control parameter: Overall Convergence Criteria, Default = 0.0001 {If GN 34 = "SPECIAL"}
34F	✯	Real	XSDRNPM calculational control parameter: Scalar Flux Point Convergence, Default = 0.0001 {If GN 34 = "SPECIAL"}
34G	✯	Integer	XSDRNPM calculational control parameter: IUS = 1 for upscatter scaling to speed convergence, IUS = 0 for no scaling, Default = 0 {If GN 34 = "SPECIAL"}
35		Integer	Number of reactor cycles in which the node is inserted
36	ℜ	2 Characters	Reactor cycle identifier in which node is inserted
37	ℜ	Integer	Number of CRC statepoints in reactor cycle in which the node is inserted (BOC is always considered statepoint 1 in a cycle)
38	ℜ	Real	Statepoint EFPD
39	ℜ	Real	Length to statepoint in calendar days
40	ℜ	Real	Downtime at statepoint
41	ℜ	Real	Days of downtime at EOC
42	ℜ	Real	Total cycle length in EFPD
43	ℜ	Real	Total cycle length in calendar days
44	ℜ	Integer	Two digit identifier indicating the current number of cycles that the node has been exposed
45		1 Character	Flag to signal if constant or variable irradiation step histories will be specified (this character should always be "Y")
46	✯, ℜ	Integer	Relative cycle number to which moderator density data applies
47	✯, ℜ	Integer	Relative statepoint number in the relative cycle to which moderator density data applies (BOC statepoint equals 1)
48	✯, ℜ	Real	Number of irradiation steps to next statepoint

Group Number	Special Notes	Line Format	Line Description
49	★, ℜ	Real, Real	Irradiation step length in EFPD, in-channel moderator density (g/cm^3) for irradiation step {If GN 45 = "Y"}
50		Integer	Number of axial nodes for CRC calculation (this value should always be 1)
51	ℜ	Real, Real	Node number, Node height (cm)
52		6 Characters	CRB data for node (this character word should always be "RODDED" whether there are control blades present or not)
53	★	Integer	Number of previously defined irradiation steps in which the node contains a CRB
53A	★, ℜ	Integer	Number of axial assembly sections containing the CRB during the irradiation step of interest (this value should always be 1)
53B	★, ℜ	Integer, Integer, Integer, Integer, Integer, Integer, Integer	Relative cycle number containing the CRB, Relative statepoint in cycle (BOC=stpt 1), Relative irradiation step number, Bottom node number containing CRB (always 1), Top node number containing CRB (always 1), CRB absorber material mixture, CRB design description number
53C	★	Integer	Number of different gadolinia-bearing fuel rod (GDR) and CRB absorber material mixtures that will be specified for use in this fuel node: mixture 1 = GDR, mixture 2 = borated bypass moderator (if bladed nodes are being processed), mixture 3 = bypass moderator
53D	★, ℜ	Integer, Real	SAS2H material mixture identifier GDR or CRB absorber material mixture, Density for GDR or CRB material mixture
53E	★, ℜ	Integer	Number of isotopes in GDR or CRB absorber material mixture
53F	★, ℜ	Integer, Real	SCALE nuclide identifier GDR or CRB absorber material mixture, wt% for nuclide in mixture
53G	★	Integer	Number of different CRB or GDR designs that will be specified for use with this fuel node (this value should always be 1)
53H	★, ℜ	Real, Integer	CRB or GDR absorber material density for design, SAS2H material mixture number for CRB or GDR cladding in CRB design (the value for CRB or GDR cladding mixture should always be 0)
53I	★, ℜ	Integer	Number of radial zones and dimensions for the Path B unit cell model for the node containing CRB or GDR
53J	★, ℜ	Integer, Real	Zone mixture identifier for use in CRB or GDR design Path B unit cell model, Corresponding zone outer radii (cm)
53K	★, ℜ	Integer, Real	Zone mixture identifier for use in Path B unit cell model after the CRB or GDR is removed, Corresponding zone outer radii (cm)

Group Number	Special Notes	Line Format	Line Description
54		6 Characters	Axial power shaping rod assembly data, (this character word should always be "NO APSRA INSERTION HISTORY")
55	ℜ	Integer	Number of axial nodes for fuel temperature input, (this value should always be 1)
56	ℜ	Real, Real	Axial node number for fuel temperature input, Corresponding axial node height (cm), (the axial node number should always be 1)
57	ℜ	Real	Axial node fuel temperature input data (K)
58	ℜ	Integer	Number of axial nodes for burnup input data (this value should always be 1)
59	ℜ	Real, Real	Axial node number for burnup input, Corresponding axial node height (cm), (the axial node number should always be 1)
60	ℜ	Real	Axial node burnup input data (GWd/MTU)

★: The existence of these input lines is dependent on certain previous input line values. The detailed descriptions for these input lines in Section 5 explain the various dependencies.

ℜ: These are recursive input lines that must be entered multiple times in a specific grouping format. The detailed descriptions for the recursive input lines in Section 5 explain the specific grouping formats and number of required input iterations.

: The continuous shaded boxes in the special notes column indicate groupings of recursive input lines. The format and content of these recursive groupings are explained in the detailed input descriptions in Section 5.

The SPACE software routine generates CRAFT calculation input files for various types of CRAFT calculations. The CRAFT input files generated by SPACE are named with a two digit integer identical to the node number that is being processed in the CRAFT-SAS2H calculation. The filenames for these file types would be as follows:

GDR and Uranium Blanket CRAFT input files:
01, 02, 03, 04, 05, 06, 07, 08, 09, 10

When a CRAFT execution for a node is complete the CRAFT input file for the node is renamed with the "*.B" extension.

In Attachment III, all SPACE and CRAFT input files are renamed and stored to a writable compact disk (WCD). The naming method for all SPACE and CRAFT input files that are stored on the WCD is demonstrated below.

CRAFT Input Files: Naming Format: "MMANN" Example: "05J12"

where:

MM - node number
A - letter designation of the assembly type
NN - assembly number

SPACE Input Files: Naming Format: "SPANN" Example: "spJ12"

where:

SP - designation for SPACE input file
A - letter designation of the assembly type
NN - the assembly number

Table 4-2. Example CRAFT Input File for Uranium Node (Node 1)

```
******************************
 THIS FILE CREATED BY:
 SPACEV01    7-Apr-99  09:54:19
 ******************************
N                         : This case NO RESTART=N
N                         : This is not a pickup case
 0                        : Gd rods/assembly node
a5                       1: Reactor Identifier, NODE #
```

```
QC2                          : Prefix Identifier for reactor
BWR                          : Reactor Type
44group                      : Scale cross section library
 .710                        : U-235 wt% enrichment in U of UO2
 7285.6                      : Grams of U per node
62                           : # rods per node
1.62560                      : Pitch
1.06430                      : Fuel Pellet diameter
1.06430                      : Fuel rod clad ID
1.22680                      : Fuel rod clad OD
15.240                       : Node height
N                            : No axial blanket fuel
INCONEL                      : Spacer grid material
0.000001                     : Mod volfrac displaced by grids
ZIRC-4                       : Fuel rod cladding material
620.00                       : Avg nodal Fuel rod clad temp (K)
N                            : Cladding materials other than ZIRC-4
 .7396                       : Reference moderator density (g/cc)
559.10                       : Reference mof/bypass temperature (K)
N                            : Activate BPRA tracking
1                            : # radial zones in pathb model
1    0.50                    : Standard PathB model
1                            : # cross section libraries per irrad step
5                            : SAS2h output print level
0.5                          : Zone mesh factor for XSDRNPM
NO SPECIAL                   : No special XSDRNPM control params
 5                           : # of reactor insertion cycles
09                           : Insertion reactor cycle indentifier
 2                           : # statepoints in cycle
   .0                        : Statepoint EFPD
   .0                        : Length to statepoint on calendar days
   .0                        : Downtime at statepoint
167.5                        : Statepoint EFPD
167.5                        : Length to statepoint on calendar days
   .0                        : Downtime at statepoint
 75.0                        : Days of downtime at end of cycle
348.4                        : Total cycle EFPD
348.4                        : Total cycle length in calendar days
 1                           : Integer position of assembly in cycle
10                           : Insertion reactor cycle indentifier
 2                           : # statepoints in cycle
   .0                        : Statepoint EFPD
   .0                        : Length to statepoint on calendar days
   .0                        : Downtime at statepoint
222.5                        : Statepoint EFPD
222.5                        : Length to statepoint on calendar days
   .0                        : Downtime at statepoint
 95.0                        : Days of downtime at end of cycle
467.5                        : Total cycle EFPD
467.5                        : Total cycle length in calendar days
 2                           : Integer position of assembly in cycle
11                           : Insertion reactor cycle indentifier
 2                           : # statepoints in cycle
   .0                        : Statepoint EFPD
   .0                        : Length to statepoint on calendar days
   .0                        : Downtime at statepoint
180.3                        : Statepoint EFPD
180.3                        : Length to statepoint on calendar days
   .0                        : Downtime at statepoint
138.0                        : Days of downtime at end of cycle
484.2                        : Total cycle EFPD
484.2                        : Total cycle length in calendar days
 3                           : Integer position of assembly in cycle
12                           : Insertion reactor cycle indentifier
```

```
 2                              : # statepoints in cycle
    .0                          : Statepoint EFPD
    .0                          : Length to statepoint on calendar days
    .0                          : Downtime at statepoint
142.2                           : Statepoint EFPD
142.2                           : Length to statepoint on calendar days
    .0                          : Downtime at statepoint
 85.0                           : Days of downtime at end of cycle
263.7                           : Total cycle EFPD
263.7                           : Total cycle length in calendar days
 4                              : Integer position of assembly in cycle
13                              : Insertion reactor cycle indentifier
 6                              : # statepoints in cycle
    .0                          : Statepoint EFPD
    .0                          : Length to statepoint on calendar days
    .0                          : Downtime at statepoint
 10.1                           : Statepoint EFPD
 10.1                           : Length to statepoint on calendar days
 16.7                           : Downtime at statepoint
112.9                           : Statepoint EFPD
112.9                           : Length to statepoint on calendar days
 16.7                           : Downtime at statepoint
224.4                           : Statepoint EFPD
224.4                           : Length to statepoint on calendar days
 13.1                           : Downtime at statepoint
324.7                           : Statepoint EFPD
324.7                           : Length to statepoint on calendar days
 16.7                           : Downtime at statepoint
387.1                           : Statepoint EFPD
387.1                           : Length to statepoint on calendar days
    .0                          : Downtime at statepoint
136.0                           : Days of downtime at end of cycle
387.1                           : Total cycle EFPD
387.1                           : Total cycle length in calendar days
 5                              : Integer position of assembly in cycle
Y                               : Flag for variable irradiation steps
 1                              : Relative insertion cycle
 1                              : Relative statepoint in insertion cycle
 3                              : # steps in statepoint calculation
 55.8          .73960           : Step length, mod density
 55.8          .73960           : Step length, mod density
 55.8          .73960           : Step length, mod density
 2                              : Relative statepoint in insertion cycle
 3                              : # steps in statepoint calculation
 60.3          .73960           : Step length, mod density
 60.3          .73960           : Step length, mod density
 60.3          .73960           : Step length, mod density
 2                              : Relative insertion cycle
 1                              : Relative statepoint in insertion cycle
 4                              : # steps in statepoint calculation
 55.6          .73960           : Step length, mod density
 55.6          .73960           : Step length, mod density
 55.6          .73960           : Step length, mod density
 55.6          .73960           : Step length, mod density
 2                              : Relative statepoint in insertion cycle
 4                              : # steps in statepoint calculation
 61.3          .73960           : Step length, mod density
 61.3          .73960           : Step length, mod density
 61.3          .73960           : Step length, mod density
 61.3          .73960           : Step length, mod density
 3                              : Relative insertion cycle
 1                              : Relative statepoint in insertion cycle
 3                              : # steps in statepoint calculation
 60.1          .73960           : Step length, mod density
```

```
 60.1       .73960         : Step length, mod density
 60.1       .73960         : Step length, mod density
 2                         : Relative statepoint in insertion cycle
 5                         : # steps in statepoint calculation
 60.8       .73960         : Step length, mod density
 60.8       .73960         : Step length, mod density
 60.8       .73960         : Step length, mod density
 60.8       .73960         : Step length, mod density
 60.8       .73960         : Step length, mod density
 4                         : Relative insertion cycle
 1                         : Relative statepoint in insertion cycle
 3                         : # steps in statepoint calculation
 47.4       .73960         : Step length, mod density
 47.4       .73960         : Step length, mod density
 47.4       .73960         : Step length, mod density
 2                         : Relative statepoint in insertion cycle
 2                         : # steps in statepoint calculation
 60.8       .73960         : Step length, mod density
 60.8       .73960         : Step length, mod density
 5                         : Relative insertion cycle
 1                         : Relative statepoint in insertion cycle
 1                         : # steps in statepoint calculation
 10.1       .73960         : Step length, mod density
 2                         : Relative statepoint in insertion cycle
 2                         : # steps in statepoint calculation
 51.4       .73960         : Step length, mod density
 51.4       .73960         : Step length, mod density
 3                         : Relative statepoint in insertion cycle
 2                         : # steps in statepoint calculation
 55.7       .73960         : Step length, mod density
 55.7       .73960         : Step length, mod density
 4                         : Relative statepoint in insertion cycle
 2                         : # steps in statepoint calculation
 50.2       .73960         : Step length, mod density
 50.2       .73960         : Step length, mod density
 5                         : Relative statepoint in insertion cycle
 1                         : # steps in statepoint calculation
 62.3       .73960         : Step length, mod density
 6                         : Relative statepoint in insertion cycle
 1                         : # steps in statepoint calculation
 62.3       .73960         : Step length, mod density
1                          : # of axial nodes in CRC format
1     15.2400              : Node #, node height
RODDED
 1                         : # of irraditation steps with CRB inserted
1                          : # axial sections with CRB inserted
 4  1  1  1  1  9  1       : Input card 47B
3                          : # of different CRA mixtures
 6      9.8300             : SAS2H mix # and density
12                         : # of isotopes in mixture
92234     .0232            : SCALE ID, wt%
92235    2.7275            : SCALE ID, wt%
92236     .0125            : SCALE ID, wt%
92238   82.7390            : SCALE ID, wt%
64152     .0050            : SCALE ID, wt%
64154     .0555            : SCALE ID, wt%
64155     .3795            : SCALE ID, wt%
64156     .5283            : SCALE ID, wt%
64157     .4065            : SCALE ID, wt%
64158     .6493            : SCALE ID, wt%
64160     .5786            : SCALE ID, wt%
 8016   11.8950            : SCALE ID, wt%
11       .7396             : SAS2H mix # and density
 2                         : # of isotopes in mixture
```

```
 1000    11.1900          : SCALE ID, wt%
 8016    88.8100          : SCALE ID, wt%
12         .7396          : SAS2H mix # and density
 2                        : # of isotopes in mixture
 1000    11.1900          : SCALE ID, wt%
 8016    88.8100          : SCALE ID, wt%
1                         : # of CRB designs
10.17    0                : CRB dens, clad mix #
 6                        : # of radial zones CRB PathB
  3   1.0643              : PathB radial zones with CRB
  2   1.2268              : PathB radial zones with CRB
  3   1.2970              : PathB radial zones with CRB
500   7.3372              : PathB radial zones with CRB
  2   7.5733              : PathB radial zones with CRB
 11   8.5982              : PathB radial zones with CRB
  3   1.0643              : PathB radial zones
  2   1.2268              : PathB radial zones
  3   1.2970              : PathB radial zones
500   7.3372              : PathB radial zones
  2   7.5733              : PathB radial zones
 12   8.5982              : PathB radial zones
NO APSRA INSERTION HISTORY
1                         : # of fuel temp axial nodes
1   15.2400               : Node #, node height (cm)
 635.5
1                         : # of fuel temp axial nodes
1   15.2400               : Node #, node height (cm)
 632.2
1                         : # of fuel temp axial nodes
1   15.2400               : Node #, node height (cm)
 603.3
1                         : # of fuel temp axial nodes
1   15.2400               : Node #, node height (cm)
 617.1
1                         : # of fuel temp axial nodes
1   15.2400               : Node #, node height (cm)
 623.3
1                         : # of fuel temp axial nodes
1   15.2400               : Node #, node height (cm)
 631.9
1                         : # of fuel temp axial nodes
1   15.2400               : Node #, node height (cm)
 590.0
1                         : # of fuel temp axial nodes
1   15.2400               : Node #, node height (cm)
 586.8
1                         : # of fuel temp axial nodes
1   15.2400               : Node #, node height (cm)
 571.7
1                         : # of fuel temp axial nodes
1   15.2400               : Node #, node height (cm)
 576.8
1                         : # of fuel temp axial nodes
1   15.2400               : Node #, node height (cm)
 573.3
1                         : # of fuel temp axial nodes
1   15.2400               : Node #, node height (cm)
 570.2
1                         : # of fuel temp axial nodes
1   15.2400               : Node #, node height (cm)
 567.4
1                         : # of burnup axial nodes
1   15.2400               : Node #, node height (cm)
 .0
```

```
1                          : # of burnup axial nodes
1   15.2400                : Node #, node height (cm)
 1.062
1                          : # of burnup axial nodes
1   15.2400                : Node #, node height (cm)
 2.162
1                          : # of burnup axial nodes
1   15.2400                : Node #, node height (cm)
 2.994
1                          : # of burnup axial nodes
1   15.2400                : Node #, node height (cm)
 4.186
1                          : # of burnup axial nodes
1   15.2400                : Node #, node height (cm)
 5.154
1                          : # of burnup axial nodes
1   15.2400                : Node #, node height (cm)
 6.994
1                          : # of burnup axial nodes
1   15.2400                : Node #, node height (cm)
 7.369
1                          : # of burnup axial nodes
1   15.2400                : Node #, node height (cm)
 7.657
1                          : # of burnup axial nodes
1   15.2400                : Node #, node height (cm)
 7.668
1                          : # of burnup axial nodes
1   15.2400                : Node #, node height (cm)
 7.825
1                          : # of burnup axial nodes
1   15.2400                : Node #, node height (cm)
 7.962
1                          : # of burnup axial nodes
1   15.2400                : Node #, node height (cm)
 8.058
1                          : # of burnup axial nodes
1   15.2400                : Node #, node height (cm)
 8.103
```

Table 4-3. Example CRAFT Input File for Node 5 (Non-bladed GDR Model)

```
******************************
 THIS FILE CREATED BY:
 SPACEV01    7-Apr-99  09:54:19
 ******************************
N                          : This case NO RESTART=N
N                          : This is not a pickup case
 7                         : Gd rods/assembly node
a5                        5: Reactor Identifier, NODE #
QC2                        : Prefix Identifier for reactor
BWR                        : Reactor Type
44group                    : Scale cross section library
3.190                      : U-235 wt% enrichment in U of UO2
14571.2                    : Grams of U per node
62                         : # rods per node
1.62560                    : Pitch
1.06430                    : Fuel Pellet diameter
1.06430                    : Fuel rod clad ID
1.22680                    : Fuel rod clad OD
```

```
30.480                  : Node height
N                       : No axial blanket fuel
INCONEL                 : Spacer grid material
0.000001                : Mod volfrac displaced by grids
ZIRC-4                  : Fuel rod cladding material
620.00                  : Avg nodal Fuel rod clad temp (K)
N                       : Cladding materials other than ZIRC-4
 .7396                  : Reference moderator density (g/cc)
559.10                  : Reference mof/bypass temperature (K)
N                       : Activate BPRA tracking
1                       : # radial zones in pathb model
1    0.50               : Standard PathB model
1                       : # cross section libraries per irrad step
5                       : SAS2h output print level
0.5                     : Zone mesh factor for XSDRNPM
NO SPECIAL              : No special XSDRNPM control params
 5                      : # of reactor insertion cycles
09                      : Insertion reactor cycle indentifier
 2                      : # statepoints in cycle
   .0                   : Statepoint EFPD
   .0                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
167.5                   : Statepoint EFPD
167.5                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
 75.0                   : Days of downtime at end of cycle
348.4                   : Total cycle EFPD
348.4                   : Total cycle length in calendar days
 1                      : Integer position of assembly in cycle
10                      : Insertion reactor cycle indentifier
 2                      : # statepoints in cycle
   .0                   : Statepoint EFPD
   .0                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
222.5                   : Statepoint EFPD
222.5                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
 95.0                   : Days of downtime at end of cycle
467.5                   : Total cycle EFPD
467.5                   : Total cycle length in calendar days
 2                      : Integer position of assembly in cycle
11                      : Insertion reactor cycle indentifier
 2                      : # statepoints in cycle
   .0                   : Statepoint EFPD
   .0                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
180.3                   : Statepoint EFPD
180.3                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
138.0                   : Days of downtime at end of cycle
484.2                   : Total cycle EFPD
484.2                   : Total cycle length in calendar days
 3                      : Integer position of assembly in cycle
12                      : Insertion reactor cycle indentifier
 2                      : # statepoints in cycle
   .0                   : Statepoint EFPD
   .0                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
142.2                   : Statepoint EFPD
142.2                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
 85.0                   : Days of downtime at end of cycle
263.7                   : Total cycle EFPD
263.7                   : Total cycle length in calendar days
```

```
 4                         : Integer position of assembly in cycle
13                         : Insertion reactor cycle indentifier
 6                         : # statepoints in cycle
   .0                      : Statepoint EFPD
   .0                      : Length to statepoint on calendar days
   .0                      : Downtime at statepoint
 10.1                      : Statepoint EFPD
 10.1                      : Length to statepoint on calendar days
 16.7                      : Downtime at statepoint
112.9                      : Statepoint EFPD
112.9                      : Length to statepoint on calendar days
 16.7                      : Downtime at statepoint
224.4                      : Statepoint EFPD
224.4                      : Length to statepoint on calendar days
 13.1                      : Downtime at statepoint
324.7                      : Statepoint EFPD
324.7                      : Length to statepoint on calendar days
 16.7                      : Downtime at statepoint
387.1                      : Statepoint EFPD
387.1                      : Length to statepoint on calendar days
   .0                      : Downtime at statepoint
136.0                      : Days of downtime at end of cycle
387.1                      : Total cycle EFPD
387.1                      : Total cycle length in calendar days
 5                         : Integer position of assembly in cycle
Y                          : Flag for variable irradiation steps
 1                         : Relative insertion cycle
 1                         : Relative statepoint in insertion cycle
 3                         : # steps in statepoint calculation
 55.8      .43403          : Step length, mod density
 55.8      .43610          : Step length, mod density
 55.8      .43817          : Step length, mod density
 2                         : Relative statepoint in insertion cycle
 3                         : # steps in statepoint calculation
 60.3      .44156          : Step length, mod density
 60.3      .44630          : Step length, mod density
 60.3      .45103          : Step length, mod density
 2                         : Relative insertion cycle
 1                         : Relative statepoint in insertion cycle
 4                         : # steps in statepoint calculation
 55.6      .45876          : Step length, mod density
 55.6      .46949          : Step length, mod density
 55.6      .48021          : Step length, mod density
 55.6      .49093          : Step length, mod density
 2                         : Relative statepoint in insertion cycle
 4                         : # steps in statepoint calculation
 61.3      .49797          : Step length, mod density
 61.3      .50132          : Step length, mod density
 61.3      .50467          : Step length, mod density
 61.3      .50802          : Step length, mod density
 3                         : Relative insertion cycle
 1                         : Relative statepoint in insertion cycle
 3                         : # steps in statepoint calculation
 60.1      .50877          : Step length, mod density
 60.1      .50690          : Step length, mod density
 60.1      .50503          : Step length, mod density
 2                         : Relative statepoint in insertion cycle
 5                         : # steps in statepoint calculation
 60.8      .50472          : Step length, mod density
 60.8      .50596          : Step length, mod density
 60.8      .50720          : Step length, mod density
 60.8      .50844          : Step length, mod density
 60.8      .50968          : Step length, mod density
 4                         : Relative insertion cycle
```

```
 1                          : Relative statepoint in insertion cycle
 3                          : # steps in statepoint calculation
 47.4      .51198           : Step length, mod density
 47.4      .51535           : Step length, mod density
 47.4      .51872           : Step length, mod density
 2                          : Relative statepoint in insertion cycle
 2                          : # steps in statepoint calculation
 60.8      .52223           : Step length, mod density
 60.8      .52587           : Step length, mod density
 5                          : Relative insertion cycle
 1                          : Relative statepoint in insertion cycle
 1                          : # steps in statepoint calculation
 10.1      .52790           : Step length, mod density
 2                          : Relative statepoint in insertion cycle
 2                          : # steps in statepoint calculation
 51.4      .52895           : Step length, mod density
 51.4      .53065           : Step length, mod density
 3                          : Relative statepoint in insertion cycle
 2                          : # steps in statepoint calculation
 55.7      .53230           : Step length, mod density
 55.7      .53390           : Step length, mod density
 4                          : Relative statepoint in insertion cycle
 2                          : # steps in statepoint calculation
 50.2      .53525           : Step length, mod density
 50.2      .53635           : Step length, mod density
 5                          : Relative statepoint in insertion cycle
 1                          : # steps in statepoint calculation
 62.3      .53745           : Step length, mod density
 6                          : Relative statepoint in insertion cycle
 1                          : # steps in statepoint calculation
 62.3      .53745           : Step length, mod density
1                           : # of axial nodes in CRC format
1     30.4800               : Node #, node height
RODDED
 1                          : # of irraditation steps with CRB inserted
1                           : # axial sections with CRB inserted
 4  1  1  1  1  9  1        : Input card 47B
3                           : # of different CRA mixtures
 7      9.8300              : SAS2H mix # and density
12                          : # of isotopes in mixture
92234     .0232             : SCALE ID, wt%
92235    2.7275             : SCALE ID, wt%
92236     .0125             : SCALE ID, wt%
92238   82.7390             : SCALE ID, wt%
64152     .0050             : SCALE ID, wt%
64154     .0555             : SCALE ID, wt%
64155     .3795             : SCALE ID, wt%
64156     .5283             : SCALE ID, wt%
64157     .4065             : SCALE ID, wt%
64158     .6493             : SCALE ID, wt%
64160     .5786             : SCALE ID, wt%
 8016   11.8950             : SCALE ID, wt%
11       .7396              : SAS2H mix # and density
 2                          : # of isotopes in mixture
 1000   11.1900             : SCALE ID, wt%
 8016   88.8100             : SCALE ID, wt%
12       .7396              : SAS2H mix # and density
 2                          : # of isotopes in mixture
 1000   11.1900             : SCALE ID, wt%
 8016   88.8100             : SCALE ID, wt%
1                           : # of CRB designs
10.17   0                   : CRB dens, clad mix #
 6                          : # of radial zones CRB PathB
  7    .5321                : PathB radial zones with CRB
```

```
  2    .6134          : PathB radial zones with CRB
  3    .9171          : PathB radial zones with CRB
500   2.7295          : PathB radial zones with CRB
  2   2.8201          : PathB radial zones with CRB
 11   3.2498          : PathB radial zones with CRB
  7    .5321          : PathB radial zones
  2    .6134          : PathB radial zones
  3    .9171          : PathB radial zones
500   2.7295          : PathB radial zones
  2   2.8201          : PathB radial zones
 12   3.2498          : PathB radial zones
NO APSRA INSERTION HISTORY
1                     : # of fuel temp axial nodes
1   30.4800           : Node #, node height (cm)
 999.7
1                     : # of fuel temp axial nodes
1   30.4800           : Node #, node height (cm)
 983.7
1                     : # of fuel temp axial nodes
1   30.4800           : Node #, node height (cm)
 906.0
1                     : # of fuel temp axial nodes
1   30.4800           : Node #, node height (cm)
 824.2
1                     : # of fuel temp axial nodes
1   30.4800           : Node #, node height (cm)
 978.8
1                     : # of fuel temp axial nodes
1   30.4800           : Node #, node height (cm)
 874.9
1                     : # of fuel temp axial nodes
1   30.4800           : Node #, node height (cm)
 694.7
1                     : # of fuel temp axial nodes
1   30.4800           : Node #, node height (cm)
 676.5
1                     : # of fuel temp axial nodes
1   30.4800           : Node #, node height (cm)
 641.8
1                     : # of fuel temp axial nodes
1   30.4800           : Node #, node height (cm)
 633.3
1                     : # of fuel temp axial nodes
1   30.4800           : Node #, node height (cm)
 619.2
1                     : # of fuel temp axial nodes
1   30.4800           : Node #, node height (cm)
 606.0
1                     : # of fuel temp axial nodes
1   30.4800           : Node #, node height (cm)
 596.5
1                     : # of burnup axial nodes
1   30.4800           : Node #, node height (cm)
 .0
1                     : # of burnup axial nodes
1   30.4800           : Node #, node height (cm)
 5.743
1                     : # of burnup axial nodes
1   30.4800           : Node #, node height (cm)
 11.776
1                     : # of burnup axial nodes
1   30.4800           : Node #, node height (cm)
 18.104
1                     : # of burnup axial nodes
```

```
1   30.4800              : Node #, node height (cm)
 23.684
1                        : # of burnup axial nodes
1   30.4800              : Node #, node height (cm)
 29.642
1                        : # of burnup axial nodes
1   30.4800              : Node #, node height (cm)
 37.65
1                        : # of burnup axial nodes
1   30.4800              : Node #, node height (cm)
 39.429
1                        : # of burnup axial nodes
1   30.4800              : Node #, node height (cm)
 40.755
1                        : # of burnup axial nodes
1   30.4800              : Node #, node height (cm)
 40.833
1                        : # of burnup axial nodes
1   30.4800              : Node #, node height (cm)
 41.551
1                        : # of burnup axial nodes
1   30.4800              : Node #, node height (cm)
 42.18
1                        : # of burnup axial nodes
1   30.4800              : Node #, node height (cm)
 42.618
1                        : # of burnup axial nodes
1   30.4800              : Node #, node height (cm)
 42.832
```

Table 4-4. Example CRAFT Input File for Node 3 (Bladed GDR Model)

```
******************************
 THIS FILE CREATED BY:
 SPACEV01     9-Apr-99  14:17:07
 ******************************
N                           : This case NO RESTART=N
N                           : This is not a pickup case
 7                          : Gd rods/assembly node
a1 - b                     3: Reactor Identifier, NODE #
QC2                         : Prefix Identifier for reactor
BWR                         : Reactor Type
44group                     : Scale cross section library
3.190                       : U-235 wt% enrichment in U of UO2
14571.2                     : Grams of U per node
62                          : # rods per node
1.62560                     : Pitch
1.06430                     : Fuel Pellet diameter
1.06430                     : Fuel rod clad ID
1.22680                     : Fuel rod clad OD
30.480                      : Node height
N                           : No axial blanket fuel
INCONEL                     : Spacer grid material
0.000001                    : Mod volfrac displaced by grids
ZIRC-4                      : Fuel rod cladding material
620.00                      : Avg nodal Fuel rod clad temp (K)
N                           : Cladding materials other than ZIRC-4
 .7396                      : Reference moderator density (g/cc)
559.10                      : Reference mof/bypass temperature (K)
N                           : Activate BPRA tracking
1                           : # radial zones in pathb model
1    0.50                   : Standard PathB model
1                           : # cross section libraries per irrad step
5                           : SAS2h output print level
0.5                         : Zone mesh factor for XSDRNPM
NO SPECIAL                  : No special XSDRNPM control params
 5                          : # of reactor insertion cycles
09                          : Insertion reactor cycle indentifier
 2                          : # statepoints in cycle
   .0                       : Statepoint EFPD
   .0                       : Length to statepoint on calendar days
   .0                       : Downtime at statepoint
167.5                       : Statepoint EFPD
167.5                       : Length to statepoint on calendar days
   .0                       : Downtime at statepoint
 75.0                       : Days of downtime at end of cycle
348.4                       : Total cycle EFPD
348.4                       : Total cycle length in calendar days
 1                          : Integer position of assembly in cycle
10                          : Insertion reactor cycle indentifier
 2                          : # statepoints in cycle
   .0                       : Statepoint EFPD
   .0                       : Length to statepoint on calendar days
   .0                       : Downtime at statepoint
222.5                       : Statepoint EFPD
222.5                       : Length to statepoint on calendar days
   .0                       : Downtime at statepoint
 95.0                       : Days of downtime at end of cycle
467.5                       : Total cycle EFPD
467.5                       : Total cycle length in calendar days
 2                          : Integer position of assembly in cycle
11                          : Insertion reactor cycle indentifier
 2                          : # statepoints in cycle
```

```
   .0                   : Statepoint EFPD
   .0                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
180.3                   : Statepoint EFPD
180.3                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
138.0                   : Days of downtime at end of cycle
484.2                   : Total cycle EFPD
484.2                   : Total cycle length in calendar days
 3                      : Integer position of assembly in cycle
12                      : Insertion reactor cycle indentifier
 2                      : # statepoints in cycle
   .0                   : Statepoint EFPD
   .0                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
142.2                   : Statepoint EFPD
142.2                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
 85.0                   : Days of downtime at end of cycle
263.7                   : Total cycle EFPD
263.7                   : Total cycle length in calendar days
 4                      : Integer position of assembly in cycle
13                      : Insertion reactor cycle indentifier
 6                      : # statepoints in cycle
   .0                   : Statepoint EFPD
   .0                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
 10.1                   : Statepoint EFPD
 10.1                   : Length to statepoint on calendar days
 16.7                   : Downtime at statepoint
112.9                   : Statepoint EFPD
112.9                   : Length to statepoint on calendar days
 16.7                   : Downtime at statepoint
224.4                   : Statepoint EFPD
224.4                   : Length to statepoint on calendar days
 13.1                   : Downtime at statepoint
324.7                   : Statepoint EFPD
324.7                   : Length to statepoint on calendar days
 16.7                   : Downtime at statepoint
387.1                   : Statepoint EFPD
387.1                   : Length to statepoint on calendar days
   .0                   : Downtime at statepoint
136.0                   : Days of downtime at end of cycle
387.1                   : Total cycle EFPD
387.1                   : Total cycle length in calendar days
 5                      : Integer position of assembly in cycle
Y                       : Flag for variable irradiation steps
 1                      : Relative insertion cycle
 1                      : Relative statepoint in insertion cycle
 3                      : # steps in statepoint calculation
 55.8       .67767      : Step length, mod density
 55.8       .68680      : Step length, mod density
 55.8       .69593      : Step length, mod density
 2                      : Relative statepoint in insertion cycle
 3                      : # steps in statepoint calculation
 60.3       .70002      : Step length, mod density
 60.3       .69905      : Step length, mod density
 60.3       .69808      : Step length, mod density
 2                      : Relative insertion cycle
 1                      : Relative statepoint in insertion cycle
 4                      : # steps in statepoint calculation
 55.6       .69804      : Step length, mod density
 55.6       .69891      : Step length, mod density
 55.6       .69979      : Step length, mod density
```

```
 55.6       .70066      : Step length, mod density
 2                      : Relative statepoint in insertion cycle
 4                      : # steps in statepoint calculation
 61.3       .70214      : Step length, mod density
 61.3       .70421      : Step length, mod density
 61.3       .70629      : Step length, mod density
 61.3       .70836      : Step length, mod density
 3                      : Relative insertion cycle
 1                      : Relative statepoint in insertion cycle
 3                      : # steps in statepoint calculation
 60.1       .71010      : Step length, mod density
 60.1       .71150      : Step length, mod density
 60.1       .71290      : Step length, mod density
 2                      : Relative statepoint in insertion cycle
 5                      : # steps in statepoint calculation
 60.8       .71374      : Step length, mod density
 60.8       .71402      : Step length, mod density
 60.8       .71430      : Step length, mod density
 60.8       .71458      : Step length, mod density
 60.8       .71486      : Step length, mod density
 4                      : Relative insertion cycle
 1                      : Relative statepoint in insertion cycle
 3                      : # steps in statepoint calculation
 47.4       .71522      : Step length, mod density
 47.4       .71565      : Step length, mod density
 47.4       .71608      : Step length, mod density
 2                      : Relative statepoint in insertion cycle
 2                      : # steps in statepoint calculation
 60.8       .71653      : Step length, mod density
 60.8       .71697      : Step length, mod density
 5                      : Relative insertion cycle
 1                      : Relative statepoint in insertion cycle
 1                      : # steps in statepoint calculation
 10.1       .71725      : Step length, mod density
 2                      : Relative statepoint in insertion cycle
 2                      : # steps in statepoint calculation
 51.4       .71740      : Step length, mod density
 51.4       .71760      : Step length, mod density
 3                      : Relative statepoint in insertion cycle
 2                      : # steps in statepoint calculation
 55.7       .71780      : Step length, mod density
 55.7       .71800      : Step length, mod density
 4                      : Relative statepoint in insertion cycle
 2                      : # steps in statepoint calculation
 50.2       .71815      : Step length, mod density
 50.2       .71825      : Step length, mod density
 5                      : Relative statepoint in insertion cycle
 1                      : # steps in statepoint calculation
 62.3       .71840      : Step length, mod density
 6                      : Relative statepoint in insertion cycle
 1                      : # steps in statepoint calculation
 62.3       .71840      : Step length, mod density
1                       : # of axial nodes in CRC format
1    30.4800            : Node #, node height
RODDED
 5                      : # of irraditation steps with CRB inserted
1                       : # axial sections with CRB inserted
 3  1  1  1  1 11  1    : Input card 47B
1                       : # axial sections with CRB inserted
 3  1  2  1  1 11  1    : Input card 47B
1                       : # axial sections with CRB inserted
 3  1  3  1  1 11  1    : Input card 47B
1                       : # axial sections with CRB inserted
 3  2  1  1  1 11  1    : Input card 47B
```

```
1                          : # axial sections with CRB inserted
 3  2  2  1  1 11  1       : Input card 47B
3                          : # of different CRA mixtures
 6      9.8300             : SAS2H mix # and density
12                         : # of isotopes in mixture
92234     .0232            : SCALE ID, wt%
92235    2.7275            : SCALE ID, wt%
92236     .0125            : SCALE ID, wt%
92238   82.7390            : SCALE ID, wt%
64152     .0050            : SCALE ID, wt%
64154     .0555            : SCALE ID, wt%
64155     .3795            : SCALE ID, wt%
64156     .5283            : SCALE ID, wt%
64157     .4065            : SCALE ID, wt%
64158     .6493            : SCALE ID, wt%
64160     .5786            : SCALE ID, wt%
 8016   11.8950            : SCALE ID, wt%
11      1.6704             : SAS2H mix # and density
13                         : # of isotopes in mixture
 6012    1.1784            : SCALE ID, wt%
 7014     .0581            : SCALE ID, wt%
14000     .4357            : SCALE ID, wt%
15000     .0261            : SCALE ID, wt%
16000     .0174            : SCALE ID, wt%
24000   11.0370            : SCALE ID, wt%
25000    1.1618            : SCALE ID, wt%
26000   39.9350            : SCALE ID, wt%
28000    5.3734            : SCALE ID, wt%
 1000    4.1070            : SCALE ID, wt%
 5010     .7510            : SCALE ID, wt%
 5011    3.3239            : SCALE ID, wt%
 8016   32.5950            : SCALE ID, wt%
12       .7396             : SAS2H mix # and density
 2                         : # of isotopes in mixture
 1000   11.1900            : SCALE ID, wt%
 8016   88.8100            : SCALE ID, wt%
1                          : # of CRB designs
10.17   0                  : CRB dens, clad mix #
 6                         : # of radial zones CRB PathB
  6    .5321               : PathB radial zones with CRB
  2    .6134               : PathB radial zones with CRB
  3    .9171               : PathB radial zones with CRB
500   2.7295               : PathB radial zones with CRB
  2   2.8201               : PathB radial zones with CRB
 11   3.2498               : PathB radial zones with CRB
  6    .5321               : PathB radial zones
  2    .6134               : PathB radial zones
  3    .9171               : PathB radial zones
500   2.7295               : PathB radial zones
  2   2.8201               : PathB radial zones
 12   3.2498               : PathB radial zones
NO APSRA INSERTION HISTORY
1                          : # of fuel temp axial nodes
1   30.4800                : Node #, node height (cm)
 992.3
1                          : # of fuel temp axial nodes
1   30.4800                : Node #, node height (cm)
 992.6
1                          : # of fuel temp axial nodes
1   30.4800                : Node #, node height (cm)
 958.5
1                          : # of fuel temp axial nodes
1   30.4800                : Node #, node height (cm)
 797.6
```

```
1                         : # of fuel temp axial nodes
1   30.4800               : Node #, node height (cm)
 746.1
1                         : # of fuel temp axial nodes
1   30.4800               : Node #, node height (cm)
 809.7
1                         : # of fuel temp axial nodes
1   30.4800               : Node #, node height (cm)
 648.9
1                         : # of fuel temp axial nodes
1   30.4800               : Node #, node height (cm)
 636.1
1                         : # of fuel temp axial nodes
1   30.4800               : Node #, node height (cm)
 597.3
1                         : # of fuel temp axial nodes
1   30.4800               : Node #, node height (cm)
 603.8
1                         : # of fuel temp axial nodes
1   30.4800               : Node #, node height (cm)
 590.5
1                         : # of fuel temp axial nodes
1   30.4800               : Node #, node height (cm)
 586.0
1                         : # of fuel temp axial nodes
1   30.4800               : Node #, node height (cm)
 580.9
1                         : # of burnup axial nodes
1   30.4800               : Node #, node height (cm)
 .0
1                         : # of burnup axial nodes
1   30.4800               : Node #, node height (cm)
 5.669
1                         : # of burnup axial nodes
1   30.4800               : Node #, node height (cm)
 11.8
1                         : # of burnup axial nodes
1   30.4800               : Node #, node height (cm)
 18.874
1                         : # of burnup axial nodes
1   30.4800               : Node #, node height (cm)
 23.972
1                         : # of burnup axial nodes
1   30.4800               : Node #, node height (cm)
 26.998
1                         : # of burnup axial nodes
1   30.4800               : Node #, node height (cm)
 33.597
1                         : # of burnup axial nodes
1   30.4800               : Node #, node height (cm)
 34.796
1                         : # of burnup axial nodes
1   30.4800               : Node #, node height (cm)
 35.676
1                         : # of burnup axial nodes
1   30.4800               : Node #, node height (cm)
 35.712
1                         : # of burnup axial nodes
1   30.4800               : Node #, node height (cm)
 36.139
1                         : # of burnup axial nodes
1   30.4800               : Node #, node height (cm)
 36.457
1                         : # of burnup axial nodes
```

```
1   30.4800            : Node #, node height (cm)
 36.698
1                      : # of burnup axial nodes
1   30.4800            : Node #, node height (cm)
 36.816
```

5. Detailed Description of CRAFT, Version 4C Input

A detailed description of the CRAFT input file format is provided in this section.

Input Group Number		Detailed Description
1-4	:	The input file for the CRAFT software routine must contain a time/date/version stamp for the SPACE, Version 01 software routine. If the input file is made without using SPACE, the first 4 lines of the input file must be blank. A similar time stamp is placed at the beginning of each SAS2H input file generated by CRAFT.
5	:	The first column in this line should always be "N". This indicates the RESTART function is not used. The RESTART function in CRAFT is obsolete and is not used.
6	:	The first column in this line should always be "N". This indicates the PICKUP CASE function is not used. The PICKUP CASE function in CRAFT is obsolete and is not used.
7	:	The first column in this line should contain the number of gadolinia-bearing fuel rods present in the node. If the node contains no gadolinia-bearing fuel rods the value should be "0".
8	:	This line should contain a 21 character problem identifier which will be placed on all SAS2H input files and echoed throughout the SAS2H output. The problem identifier must be placed in columns 1 through 21 of this line. The next 3 columns on line 7 should contain an integer node identifier . An example of a problem identifier would be "D9-No CRB tracking 3".

Input Group Number		**Detailed Description**
9	:	This line should contain a 3 character prefix which will be used as the initial 3 characters of each file generated in the CRAFT calculation. The prefix must be placed in columns 1 through 3 of this line. An example of a prefix meaningful for use with the problem identifier example previously provided would be "QC2".
10	:	This line should contain 3 characters starting in column 1. These characters should always be "BWR".
11	:	This line should contain the identifier for the SCALE cross section library which is to be used in all of the SAS2H calculations generated by the CRAFT calculation. Available SCALE cross section libraries include the following: 1) 44GROUPNDF5 or 44group; 2) 27BURNUPLIB; 3) 27GROUPNDF4; 4) 238GROUPNDF5; 5) HANSEN-ROACH. The 44group cross-section library is used in all CRAFT calculations relevant to Commercial Reactor Critical evaluations.
12	:	This line should contain the weight percent of U-235 in the UO_2 fuel of the node. This value should not be adjusted to compensate for axial blanket fuel.
13	:	This line should contain the total mass of uranium metal in the fuel node in units of grams per node.
14	:	This line should contain the number of fuel rods in the fuel node.
15	:	This line should contain the rod pitch in the node in units of cm.
16	:	For BWR depletion calculations, the nominal fuel rod cladding inner diameter in units of cm should be used here.
17	:	This line should again contain the nominal fuel rod cladding inner diameter in units of cm.

Input Group Number		Detailed Description
18	:	This line should contain the nominal fuel rod cladding outer diameter in units of cm.
19	:	This line should contain the nominal fuel node height in units of cm.
20	:	The character in the first column of this line should be "N". This character requests that the CRAFT software routine make no model calculations based on assembly axial blanket designs.
21	:	This line should contain a 7 character name, beginning in column 1, which specifies the spacer grid material, this character name should always be "INCONEL". The INCONEL material is only represented in trace amounts as indicated by input group 22.
22	:	This line should contain a value representing the volume fraction of the moderated region of the fuel node, which is displaced by spacer grid material. For the BWR depletion calculations, this value should be set to 0.000001. The sum of the moderator volume fraction and the spacer grid volume fraction should equal one. This value defines the model of the moderated region of the node to be pure water without a spacer grid present.
23	:	This line should contain the identification of the fuel cladding material. This character name should always be "ZIRC-4".
24	:	This line should contain an average fuel rod cladding temperature value in units of degrees Kelvin that will be used consistently throughout the CRAFT generated SAS2H calculations.
25	:	This character should be "N". ZIRC-4 is the only cladding specification available in CRAFT, Version 4C.
26	:	Reference moderator density is entered on this line in the units of grams per cubic centimeter. This reference moderator density will be used to determine other relative density fractions (axial in-channel moderator profiles will reference this value). This reference moderator density will be applied in the material specification of the moderator in all SAS2H calculations

Input Group Number		**Detailed Description**
		generated by CRAFT.
27	:	This line should contain the in-channel/bypass moderator temperature in units of degrees K.
28	:	This character should be "N". There is no BPRA tracking used in BWR depletion calculations.
29	:	This line should contain an integer value representing the number of radial zones in the SAS2H Path B model for the fuel node. For BWR depletion calculations this value should be 6. This is called the standard Path B model.
30	:	This line contains the description of a single radial zone in the standard Path B model for the fuel node. This line should contain two values delimited by spaces. The first should be an integer value representing the SAS2H material mixture number for the Path B model radial zone which this line represents. The second value should be the outer radius (cm) of the Path B model radial zone which this line represents.
31	:	This line should contain an integer value representing the number of cross-section libraries that are to be produced for each irradiation step in the SAS2H calculations generated by CRAFT. The number of cross-section libraries per irradiation step for CRC evaluations should be set to 1.
32	:	This line should contain an integer value representing the SAS2H print level desired for the output of SAS2H calculations generated by CRAFT. The minimum print level allowed for CRC evaluations is 5. A complete listing of the available print levels is provided on page S2.5.18 of Reference 2.
33	:	This line should contain the zone mesh factor that should be utilized by XSDRNPM in the SAS2H calculations generated by CRAFT. A description of the zone mesh factor is provided on page S2.5.5 of Reference 2.
34	:	The CRAFT calculation allows the specification of special XSDRNPM control parameters that will be utilized in SAS2H calculations generated by CRAFT. If any of the special control parameters described in input groups 34A through 34G are to be

Input Group Number		Detailed Description
		specified, the character string "SPECIAL" must be provided in columns 1 through 7 of this line. Any other character string specification indicates that the default XSDRNPM control parameters are to be utilized.
34A	:	This line should only be specified if the value of input group number 34 is "SPECIAL". This line contains the XSDRNPM calculation control parameter SZF. The size of the largest spatial mesh interval can be adjusted by entering a value for SZF. SZF less than 1 indicates finer mesh spacing. SZF greater than one indicates a coarser mesh spacing. SZF equal to 1 is the default.
34B	:	This line should only be specified if the value of input group number 34 is "SPECIAL". This line contains the XSDRNPM calculation control parameter ISN. The ISN value specifies the order of angular quadrature for XSDRNPM. Quadrature sets are geometry-dependent quantities that are defaulted to a value of 8.
34C	:	This line should only be specified if the value of input group number 34 is "SPECIAL". This line contains the XSDRNPM calculation control parameter IIM. The IIM value specifies the maximum number of inner iterations to be used by XSDRNPM. The default value is 20.
34D	:	This line should only be specified if the value of input group number 34 is "SPECIAL". This line contains the XSDRNPM calculation control parameter ICM. The ICM value specifies the maximum number of outer iterations to be used by XSDRNPM. The default value is 25.
34E	:	This line should only be specified if the value of input group number 34 is "SPECIAL". This line contains the XSDRNPM calculation control parameter EPS. The EPS value specifies the overall convergence criteria. This value is used by XSDRNPM after each outer iteration to determine if the problem has converged. The default value of EPS is 0.0001. A smaller value tightens the convergence criteria, and a larger value loosens the convergence criteria.
34F	:	This line should only be specified if the value of input group

Input Group Number		Detailed Description
		number 34 is "SPECIAL". This line contains the XSDRNPM calculation control parameter PTC. The PTC value specifies the point flux convergence criteria used by XSDRNPM to determine if convergence has been achieved after an inner iteration. The default value of PTC is 0.0001. A smaller value tightens the convergence criteria, and a larger value loosens the convergence criteria.
34G	:	This line should only be specified if the value of input group number 34 is "SPECIAL". This line contains the XSDRNPM calculation control parameter IUS. The IUS value is a flag to direct XSDRNPM to use an upscatter scaling technique to accelerate the solution or force convergence. The default value is 0, which indicates that upscatter scaling is not used. An IUS value of 1 directs XSDRNPM to use the upscatter scaling technique. The default value is 0.
35	:	This line should specify an integer number of reactor cycles in which the fuel node is inserted in the CRAFT calculation.

Input groups 36 through 44 represent an input grouping that must be specified recursively for each reactor cycle in which the fuel node is inserted in the CRAFT calculation as denoted in input group 35. This means that input groups 36 through 44 would be input for the first reactor cycle, and then input again for the second reactor cycle, etc., until all of the number of reactor cycles specified in input group 35 have been described.

36	:	This line should contain a 2-character reactor cycle identifier that will be used to identify the cycle on appropriate SAS2H input files generated by the CRAFT calculation. For example, if the first reactor cycle were identified as "Cycle-1A", the value of this input line should be "1A". If a reactor cycle were identified as "Cycle-1", the value of this input line should be "01", etc..
37	:	This line should contain an integer value specifying the number of CRC statepoints in the reactor cycle specified in input group number 35. The BOC is always considered statepoint 1 in a CRC evaluation. For example, if the reactor cycle specified in input line 35 contained one mid-cycle CRC statepoint, the value specified on this line would be 2.

Input Group Number		Detailed Description

Input groups 38 through 40 represent an input grouping that must be specified recursively for each CRC statepoint in the reactor cycle as denoted in input line 37. This means that input groups 38 through 40 would be input for the first statepoint (BOC), and then input again for the second statepoint, etc., until all of the number of CRC statepoints in the reactor cycle as specified in input group 37 have been described.

38	:	This line should contain a value specifying the EFPD for the statepoint. If the first statepoint in a reactor cycle (BOC) is being described, the value of this line should be 0.
39	:	This line should contain a value specifying the length in calendar days from the BOC to the CRC statepoint. If the first statepoint in a reactor cycle (BOC) is being described, the value of this line should be 0.
40	:	This line should contain a value specifying the downtime in calendar days for the reactor shutdown at the CRC statepoint. If the first statepoint in a reactor cycle (BOC) is being described, the value of this line should be 0.
41	:	This line should contain a value specifying the downtime in calendar days at the EOC reactor shutdown.
42	:	This line should contain a value specifying the total EFPD for the reactor cycle from the BOC startup to the EOC shutdown.
43	:	This line should contain a value specifying the total cycle length in calendar days from the BOC startup to the EOC shutdown.
44	:	This line should contain a two-digit integer value that indicates the current number of cycles that the node has been exposed in the reactor core. For example, 01 would be the integer position of the node for the first exposure of the assembly in the core; 02 would be the second cycle of exposure, etc.
45	:	This line should contain a single character to signal the selection of a variable or constant irradiation step to CRAFT. For the BWR depletion calculations this character should always be "Y". This variable step description allows the specification of unique irradiation step duration for each irradiation step in a statepoint calculation.

Input Group Number		Detailed Description

Input groups 46 through 49 represent an input grouping that must be specified recursively for each reactor cycle in which the fuel node is inserted in the CRAFT calculation as denoted in input group 35. This means that input lines 46 through 49 would be input for the first reactor cycle, and then input again for the second reactor cycle, etc., until all of the number of reactor cycles specified in input group 35 have been described.

46	:	This line should contain an integer value specifying the relative cycle number to which the input data provided in the current grouping of input groups 46 through 49 apply. For example, if the CRAFT calculation involved two reactor cycles labeled Cycle-1 and Cycle-5, the relative cycle number corresponding to Cycle-5 would be specified as 2.

Input groups 47 through 49 represent an input grouping that must be specified recursively for the SAS2H calculations commencing from each statepoint in the relative reactor cycle specified in input group 35. This means that input groups 47 through 49 would be input for the first statepoint calculation (BOC to statepoint 2) in the reactor cycle, and then input again for the second statepoint calculation (perhaps statepoint 2 to statepoint 3) in the reactor cycle, etc., until all of the statepoint calculations in the reactor cycle, as specified in input group 37 corresponding to the appropriate reactor cycle, have been described. The last iteration of input groups 47 through 49 for a given reactor cycle should correspond to the last mid-cycle statepoint to EOC SAS2H calculation.

47	:	This line should contain an integer value corresponding to the relative statepoint calculation number in the reactor cycle for which input data is being provided. The BOC to mid-cycle statepoint 2 calculation is always considered relative statepoint calculation 1. The last mid-cycle statepoint to EOC calculation is always considered the last relative statepoint calculation in a given reactor cycle.
48	:	This line should contain an integer value specifying the number of irradiation steps to be utilized in the CRAFT generated SAS2H calculation corresponding to the statepoint calculation for which input data is being provided.
49	:	This line should contain two real values delimited by spaces. The first value on this line should specify the irradiation step length in EFPD for the SAS2H statepoint calculation for which input data is being provided. For BWR assemblies, this line

Input Group Number		**Detailed Description**
		should contain the moderator density in units of g/cm^3 at the mid-point of a given irradiation step in the current statepoint calculation for which input data is being provided. This input line must be repeated a number of times equal to that specified in input group 48. The order of repetition of this input group should be such that the initial moderator density corresponds to the first irradiation step, and the final moderator density corresponds to the last irradiation step in the statepoint calculation of interest.
50	:	This line should contain an integer value corresponding to the number of axial nodes utilized in the CRC evaluation. For the BWR calculations, this value should always be 1.
51	:	This line contains two integer values delimited by spaces. The first value specifies the axial node and should always be 1. The second value specifies the corresponding node height in units of cm.
52	:	In the BWR calculations, the CRAFT software routine models nodes containing CRBs always, regardless of whether or not a CRB is actually present in the modeled node. This function enables CRAFT to model CRB and GDR material mixtures in the Path B model of SAS2H. The character string "RODDED" should be placed in columns 1 through 6 of this line for all BWR depletion calculations.
53	:	This line should contain an integer value specifying the number of previously defined irradiation steps in the CRAFT calculation in which the fuel node contains a CRB.
53A	:	This line should contain an integer value specifying the number of axial sections of the fuel assembly, which contain a CRB during the irradiation step for which data is being provided. For the BWR calculations, this value should always be 1. This line should be repeated the number of times specified in input group number 53.
53B	:	At least 1 CRB insertion format should be indicated on this line whether or not a CRB is actually present in the node. For a bladed or non-bladed node the bypass water mixture should be

Input Group Number		**Detailed Description**
		either borated or left pure to indicate the presence of a CRB. In other words, a CRB format definition must be made in CRAFT for a BWR calculation, but the actual insertion of a CRB in a node is determined by the bypass moderator material specification. This line must be repeated a number of times equal to that specified in input group 53. This line should contain 7 integer values delimited by spaces. The first integer value specifies the relative cycle number in the CRAFT calculation in which a CRB is inserted. The second integer value specifies the relative statepoint calculation number in which a CRB is inserted in the cycle identified by the first value of this line. The BOC to statepoint 1 is always considered statepoint calculation 1. The third value specifies the relative irradiation step number in the statepoint calculation identified by the second value of this line in which the CRB is inserted. The fourth and fifth values in this line are always 1. The sixth value specifies the CRB absorber material mixture number for SAS2H corresponding to the CRB described on this line. The CRB and GDR absorber material specifications and mixture numbers are specified in input groups 53C through 53F. The seventh value is always 1. The first material specification made at input group 52D is always reserved for GDR materials.
53C	:	This line should contain an integer value specifying the number of different CRB or GDR absorber material mixtures which must be specified for use in the various CRB and GDR designs which are inserted in the fuel node during its irradiation history relevant to the CRAFT calculation. For the BWR calculations, this number should always be 3.

Input groups 53D through 53F represent an input grouping that must be specified recursively for each CRB absorber material mixture used in the CRAFT calculation as denoted in input group 53C. This means that input groups 53D through 53F would be input for the first CRB absorber material mixture, and then input again for the second CRB absorber material mixture, etc., until all of the CRB absorber material mixtures specified in input group 53C have been described.

53D	:	For BWR assemblies, this line should contain the SAS2H material mixture number for the CRB or GDR followed by a real value for the density of the corresponding absorber material

Input Group Number		Detailed Description
		mixture.
53E	:	This line should contain an integer value specifying the number of isotopes in the CRB or GDR absorber material mixture specified in input group 53D.
53F	:	This line should contain two values delimited by spaces. The first value should be the SCALE nuclide identifier corresponding to a constituent of the CRB or GDR absorber material mixture specified in input group 53D. The second value should be the weight percent of the nuclide, identified by the first value, in the CRB or GDR absorber material mixture specified in input group 53D. This input line must be repeated a number of times equal to that specified in input group 53E such that data for all nuclides in the CRB absorber material mixture are provided, and the sum of the weight percents of the nuclides in the mixture equals 100.
53G	:	This line should contain an integer value specifying the number of different CRB design descriptions that will be specified for use in the CRAFT calculation. This value should always be 1 for BWR depletion calculations.

Input groups 53H through 53K represent an input grouping that must be specified recursively for each CRB design used in the CRAFT calculation as denoted in input group 53G. This means that input groups 53H through 53K would be input for the first CRB design description, and then input again for the second CRB design description, etc., until all of the CRB design descriptions specified in input group 53G have been described. The order in which the CRB design descriptions are provided determines the relative CRB design number that corresponds to the description.

53H	:	This line contains two values delimited by spaces. The first value should specify the absorber material density in units of g/cm^3 for the CRB design for which input is being provided. The second value should be 0, which specifies the SAS2H material mixture number for the CRB cladding.
53I	:	This line should contain an integer value specifying the number of radial zones utilized in the SAS2H Path B model for the fuel node containing the CRB design for which input is being provided.

Input Group Number		Detailed Description

Input groups 53J and 53K represent an input grouping that must be specified recursively for a Path B unit cell with and without a CRB. This means that input groups 53J and 53K would be input for the node model containing a CRB first, and then contain input for the node model without a CRB present. Note that input groups 53J and 53K require repetitive input themselves.

53J	:	This line contains the description of a single radial zone in the SAS2H Path B model for the fuel node containing the CRB design for which input is being provided. This line should contain two values delimited by spaces. The first of which should be an integer value representing the SAS2H material mixture number for the Path B model radial zone which this line represents. The second value should be the outer radius of the Path B model radial zone which this line represents. This input line must be repeated a number of times equal to that specified in input group 53I.
53K	:	If a node contains a CRB in one cycle but not in another, an alternative SAS2H Path B model must be provided that describes the node after removal of the CRB. This alternative Path B model must contain the same number of radial zones as the Path B model with the CRB inserted. This line contains the description of a single radial zone in the SAS2H Path B model for the node after the removal of the CRB design for which input is being provided. This line should contain two values delimited by spaces. The first of which should be an integer value representing the SAS2H material mixture number for the Path B model radial zone which this line represents. The second value should be the outer radius of the Path B model radial zone which this line represents. This input line must be repeated a number of times equal to that specified in input group 53I.
54	:	For the BWR calculations, the character words "NO APSR INSERTION HISTORY" should be entered on this line.

Input groups 55 through 57 represent input grouping that must be specified recursively for each statepoint calculation to be generated by the CRAFT calculation. This means that input groups 55 through 57 would be input for the first statepoint calculation (BOC to statepoint 2 of relative cycle number 1), and then input again for the second statepoint calculation, etc., until all of the statepoint calculations to be generated by CRAFT have

Input Group Number		**Detailed Description**

been addressed (the final statepoint calculation would be that ending at the final statepoint in the last relative cycle).

55	:	For BWR calculations the integer value for this line should be 1. This line should contain an integer value specifying the number of axial nodes in the axial format in which the current fuel temperature input data is being provided.
56	:	This line should contain two values delimited by spaces. The first value should always be 1. The second value should be the node height corresponding to the axial node number identified by the first value.
57	:	This line should contain an exposure weighted average fuel temperature value in units of degrees Kelvin for the appropriate node in the fuel temperature input axial format corresponding to the statepoint calculation for which input data is being provided.

Input groups 58 through 60 represent input groupings that must be specified recursively for each statepoint calculation to be generated by the CRAFT calculation. This means that input groups 58 through 60 would be input for the first statepoint calculation (BOC to statepoint 2 of relative cycle number 1), and then input again for the second statepoint calculation, etc., until all of the statepoint calculations to be generated by CRAFT have been addressed (the final statepoint calculation would be that ending at the final statepoint in the last relative cycle).

58	:	This line should contain an integer value specifying the number of axial nodes in the axial format in which the current burnup input data is being provided. For BWR calculations, this value should always be 1.
59	:	This line should contain two values delimited by spaces. The first value should always be 1. The second value should be the node height corresponding to the axial node number identified by the first value.
60	:	This line should contain an exposure weighted average burnup value in units of GWd/MTU corresponding to the total burnup of the node at the beginning of the statepoint calculation for which input data is being provided.

6. CRAFT Output Description

The CRAFT software routine generates five types of files identified as either "*.input", "*.output", "*.cut", "*.msgs", or "*.notes", where the "*" is the base file set identifier for the statepoint calculation of interest. The "*.cut" and "*.notes" files are the only files that must be retained for CRC evaluation and documentation purposes. All files are generated in the working directory in which the CRAFT calculation is performed.

All CRAFT, Version 4C generated filenames utilize the following format: "{Base File Set Identifier}.{suffix}". Where the suffix corresponds to one of the five file types previously mentioned, and the base file set identifier is a 25 character name containing essential information necessary to delineate one CRAFT generated SAS2H calculation from another for a particular assembly.

The base file set identifier for a statepoint calculation contains the following information:

1) reactor identifier (three character);
2) sequential node insertion number (two digit);
3) axial node number (node 1 is always the top node) (two digit);
4) reactor cycle number in which the SAS2H calculation starts (two character);
5) EFPD statepoint at which the SAS2H calculation starts (truncated to three digits);
6) reactor cycle number in which the SAS2H calculation ends (two character);
7) EFPD statepoint at which the SAS2H calculation ends (truncated to three digits).

The format of the base file set identifier is as follows where the numbers identified as #{number} correspond to one of the seven items previously listed: #1 A #2 N #3 DC #4 T #5 AC #6 T #7. The base file set identifier does not contain any spaces.

The "*.input" files contain a CRAFT generated SAS2H input file. The "*.output" files contain a complete SAS2H calculation output file. The "*.cut" files contain the corresponding SAS2H input file followed by an output extraction, from the final ORIGEN pass of the SAS2H calculation, which contains data relevant to CRC evaluations. The "*.msgs" files contain the standard run-time messages associated with the SAS2H calculation. The "*.notes" files contain a listing of the isotopes and their concentrations which were left behind in generating the initial charge fuel composition for a continuation SAS2H calculation. The "*.notes" files are only generated for CRAFT generated SAS2H calculations which are continuing depletion and decay calculations. The "*.cut" and "*.notes" files contain all of the information which is required to perform CRC evaluations or repeat calculations as necessary for quality assurance purposes. The remainder of the CRAFT generated files may be discarded once the "*.cut" and "*.notes" files have been produced correctly.

7. Testing of CRAFT, Version 4C

The CRAFT, Version 4C input files for the Quad Cities Unit 2 CRC evaluations presented in this depletion calculation file serve as the input test cases for CRAFT, Version 4C. These input files and their corresponding output files were thoroughly reviewed to verify that CRAFT was operating correctly.

8. References

1) CRWMS M&O (Civilian Radioactive Waste Management System Management and Operation Contractor) 1998. *CRC Depletion Calculations for McQuire Unit 1*. Attachment I, CRAFT, Version 5. B00000000-01717-0210-00003 REV 00. Las Vegas, Nevada: CRWMS M&O. ACC: MOL.19980626.0579.

2) Oak Ridge National Laboratory 1995. *SCALE, Version 4.3: Modular Code System for Performing Standardized Computer Analyses for Licensing Evaluation*. User's Manual Volumes 0 through 3. CCC-545. Oak Ridge, Tennessee: Distributed by the Radiation Shielding Information Center. TIC: 235920.

9. CRAFT, Version 4C Fortran Source Code Listing

```
 PROGRAM CRAFT
************************************************************************
*     Commercial Reactor Assembly Follow Taskmaster                   *
*     Boiling Water Reactor (BWR) Version                             *
*                                                                     *
*                                                                     *
*                                                                     *
*     craftv4c.f   version 4c  12/28/98  dph                          *
************************************************************************
*     This software routine writes the SAS2H input decks necessary to *
*     perform depletion and decay calculations on an assembly         *
*     required in subsequent Commercial Reactor Critical              *
*     evaluations.  The software routine controls the SAS2H input deck *
*     creation such that a new SAS2H input deck is developed          *
*     to perform depletion and decay calculations between CRC         *
*     statepoints in a given sequence.  The depletion and             *
*     decay of the fuel assembly through all CRC statepoints          *
*     is simulated as a continuous process by using feed fuel         *
*     isotopics from the previous calculation in the sequence.        *
************************************************************************
*     This software routine is only for use with BWR CRC depletion    *
*     calculations.                                                   *
************************************************************************
*
*     1/8/98  : dph blocked mod to detect fuel in center zone of pathB
*                   and adjust SAS2H input
*
*     1/15/98 : dph turned off fuel temp conversion for BWRs
*
*     1/16/98 : dph inserted STOP feature for RESTARTS
*
*     1/19/98 : dph replaced boron depletion secton for BPRA to
*                   Gd depletion and calculation of depleted
*                   Gd and UO2 mixture for continuation case
*
*
```

```
*     3/10/98 : dph date/time stamp feature and clean up
*                   version 4b
*
*     12/28/98: dph version 4c
*
*     12/29/98: dph made corrections for first moderator density
*                   calculation in SAS2H
*
*
*      1/23/99: dph modified calculation of gadolinia isotopics for
*                   continuation case
*
*      2/4/99: dph cleanup and corrected bug in retreiver sub to prevent
*                  misread of gadolinia isotope masses
*
*
****************************************************************
*
      INTEGER*4 BPZONE(10), BPMA(15,10), LMA(15,10), LUZONE,
     c LMB(15), NLIB, PLEVEL, ISN, IIM, ICM, IUS, NBR, AXNUM,
     c FTNUM(20), MONUM(20), BUNUM(20), CT1, CT2,
     c APSRINS(10,20,23,50), APSRSTEPNUM,
     c APSRMIXNUM, APSRMIXID(25),
     c CRINS(10,20,23,50), CRSTEPNUM,
     c CRMIXNUM, CRMIXID(25), CRNUMISOS(25),
     c CRISOID(25,20), AXBLANK(50), AXBLANKNODNUM,
     c STPTS(10), CYCPOS(10), APSRNUMISOS(25), APSRISOID(25,10),
     c STPTSUM, BPRADESNUM, CRDESNUM, CRZONE(10), CRMA(15,10),
     c LMC(15,10), APSRDESNUM, APSRZONE(10), APSRMA(15,10),
     c LMD(15,10), BPCYCID, BPTN(10), BPBN(10), DES, BPCYCNUM,
     c BPDESID(10), CRDES(10,20,23,50), APSRDES(10,20,23,50),
     c RELATIVE_STPT_NUM, RELATIVE_APSR_MIX_ID,
     c STPTTALLY(20), CT1START, CT2START, CLADTOT, CLADDESNUM(10),
     c BPRCLAD(10), CRCLAD(10), APSRCLAD(10), BPMIXNUM, BPMIX(10),
     c BPMIXID(10), BPNUMISOS(20), BPISOID(10,20), VARSTEPNUM(10,20),
     c BPRFM(15,10), BPFMNUMISOS(25), BPFISOID(25,10), ABOVEBPNUM(10),
     c APSRFM(15,10), APSRFOLLOWMIX(10,20,23,50),CYCSTOP,
     C STPTSTOP,GDROD
*
      REAL CLTEMP, PRESS, BPDEN(10), BPRA(15,10), CRISOWTPCT(25,20),
     c LRA(15,10), LRB(15), MESH, SZF, EPS, PTC, APSRISOWTPCT(25,10),
     c NODES(50,2), BLETDOWN(10,20,25), AXBLANKRICH, STPTDAT(10,20,3),
     c FTNDES(50,2,20), FTDAT(50,20), MONDES(50,2,20), MODAT(50,20),
     c BUNDES(50,2,20), BUDAT(50,20), RICH, FMASS, RODS, CYCLEN(10,2),
     c PITCH, FOD, CID, COD, LENGTH, CYCDOWN(10), CRDEN(10),
     c CRRA(15,10), LRC(15,10), APSRDEN(10), APSRRA(15,10), LRD(15,10),
     c BPWTPCT(10), HTOT, FDHT(20), MDHT(20), BDHT(20), FTIN(50,20),
     c MOIN(50,20), BUIN(50,20), GRAMS(50), POWER(50,20),
     c FTFINAL(50,20), MODDENFINAL(50,20), MODTEMPFINAL(50,20),
     c DENDAT(29,10), BPISOWTPCT(10,20), BPXSECT(10), UCSPACERFRAC,
     c VARBLETDOWN(10,20,25,25), VARPOWER(10,20,25,50), BPRFR(15,10),
     c BPFISOWTPCT(25,10), APSRFR(15,10), MODREFDEN, MODREFTEMP,
     c CRMIXDEN(25),FIRSTMODEN(20,20)
*
      CHARACTER REACT*23, PREFIX*3, AXBLANKET*1, BPRFLAG*1,
     c FUELCLAD*10, FLAG2*7, CYCLEID(10)*2, CRSTAT*6,
     c APSRSTAT*6, LIB*15, NM*31, CLADDESNAME(10)*7,STOPFLAG*1,
     c SPACERMAT*7, STEPCONTROL*1, ABOVEBP(10)*5, RTYPE*3,
     C PICKUPFLAG*1
*
*    Data input for table of subcooled water density (g/cc) at
*    various temperatures (F) and pressures (psia).
*    (REFERENCE: Radiation Shielding Information Center Number
*    CCC-545, "SCALE 4.2, Modular Code System for Performing
```

```
*     Standardized Computer Analyses for Licensing Evaluation,
*     Volume 1, Page S2.5.14, Table S2.5.2.)
*
       DATA ((DENDAT(E,Q),Q=1,10),E=1,29) /0.0,3000.0,2500.0,
      c 2000.0,1500.0,1000.0,
      c 800.0,600.0,400.0,200.0,50.0,1.0084,1.0069,1.0055,1.0040,
      c 1.0025,1.0019,
      c 1.0013,1.0007,1.000,100,1.0018,1.0004,0.9989,0.9975,0.9960,
      c 0.9954,0.9948,0.9942,0.9936,150.0,0.9893,0.9878,0.9864,0.9849,
      c 0.9834,0.9828,0.9822,0.9815,0.9809,200,0.9725,0.9709,0.9694,
      c 0.9679,0.9663,0.9656,0.9650,0.9644,0.9637,250.0,0.9522,0.9505,
      c 0.9489,0.9472,0.9455,0.9449,0.9442,0.9435,0.9428,300,0.9289,
      c 0.9271,0.9252,0.9234,0.9215,0.9208,0.9200,0.9192,0.9185,350.0,
      c 0.9026,0.9006,0.8985,0.8964,0.8943,0.8934,0.8925,0.8916,0,
      c 400.0,0.8733,0.8709,0.8685,0.8660,0.8634,0.8624,0.8613,0.8603,0,
      c 450.0,0.8405,0.8375,0.8345,0.8314,0.8281,0.8268,0.8255,0,0,
      c 500.0,0.8029,0.7992,0.7952,0.7911,0.7869,0.7851,0,0,0,
      c 510.0,0.7947,0.7907,0.7866,0.7822,0.7776,0,0,0,0,
      c 520.0,0.7862,0.7820,0.7776,0.7729,0.7680,0,0,0,0,
      c 530.0,0.7775,0.7729,0.7682,0.7632,0.7579,0,0,0,0,
      c 540.0,0.7683,0.7635,0.7584,0.7530,0.7472,0,0,0,0,
      c 550.0,0.7589,0.7537,0.7482,0.7423,0,0,0,0,0,
      c 560.0,0.7490,0.7434,0.7374,0.7310,0,0,0,0,0,
      c 570.0,0.7386,0.7326,0.7261,0.7190,0,0,0,0,0,
      c 580.0,0.7278,0.7212,0.7141,0.7062,0,0,0,0,0,
      c 590.0,0.7164,0.7092,0.7012,0.6923,0,0,0,0,0,
      c 600.0,0.7043,0.6963,0.6874,0,0,0,0,0,0,
      c 610.0,0.6915,0.6825,0.6724,0,0,0,0,0,0,
      c 620.0,0.6777,0.6676,0.6558,0,0,0,0,0,0,
      c 630.0,0.6629,0.6512,0.6370,0,0,0,0,0,0,
      c 640.0,0.6467,0.6329,0,0,0,0,0,0,0,
      c 650.0,0.6288,0.6119,0,0,0,0,0,0,0,
      c 660.0,0.6086,0.5866,0,0,0,0,0,0,0,
      c 670.0,0.5850,0,0,0,0,0,0,0,0,
      c 680.0,0.5559,0,0,0,0,0,0,0,0/
*
C        OPEN (UNIT=99, FILE='debug', STATUS='unknown')
       write (*,*) 'calling data_aquisition'
       CALL DATA_AQUISITION (BPZONE, BPMA,
      c LMB, NLIB, PLEVEL, ISN, IIM, ICM, IUS, NBR, AXNUM,
      c FTNUM, MONUM, BUNUM, APSRINS,
      c APSRSTEPNUM, APSRMIXNUM, APSRMIXID, CRINS,
      c CRSTEPNUM, CRMIXNUM, CRMIXID, CRNUMISOS,
      c CRISOID, AXBLANK, AXBLANKNODNUM, STPTS,
      c CYCPOS, APSRNUMISOS, APSRISOID, STPTSUM,
      c BPRADESNUM, CRDESNUM, CRZONE, CRMA, LMC,
      c APSRDESNUM, APSRZONE, APSRMA, LMD,
      c BPCYCID, BPTN, BPBN, DES, BPCYCNUM, BPDESID,
      c CRDES, APSRDES, LMA, LUZONE,
      c CLTEMP, PRESS, BPDEN, BPRA, CRISOWTPCT,
      c LRA, LRB, MESH, SZF, EPS, PTC, APSRISOWTPCT,
      c NODES, BLETDOWN, AXBLANKRICH, STPTDAT,
      c FTNDES, FTDAT, MONDES, MODAT,
      c BUNDES, BUDAT, RICH, FMASS, RODS, CYCLEN,
      c PITCH, FOD, CID, COD, LENGTH, CYCDOWN, CRDEN,
      c CRRA, LRC, APSRDEN, APSRRA, LRD,
      c BPWTPCT, REACT, PREFIX, AXBLANKET, BPRFLAG,
      c FUELCLAD, FLAG2, CYCLEID, CRSTAT,
      c APSRSTAT, LIB, BPXSECT, BPRODS, CT1START,
      c CT2START, CLADTOT, CLADDESNUM, CLADDESNAME,
      c BPRCLAD, CRCLAD, APSRCLAD, BPMIXNUM, BPMIX, BPMIXID,
      c BPNUMISOS, BPISOID, BPISOWTPCT, UCSPACERFRAC,
      c SPACERMAT, STEPCONTROL, VARBLETDOWN, VARSTEPNUM,
      c BPRFM, BPFMNUMISOS, BPFISOID, ABOVEBPNUM, APSRFM,
```

```
     c BPRFR, BPFISOWTPCT, APSRFR, ABOVEBP, APSRFOLLOWMIX,
     c RTYPE, MODREFDEN, MODREFTEMP, CRMIXDEN,STOPFLAG,CYCSTOP,
     C STPTSTOP,PICKUPFLAG,GDROD,FIRSTMODEN)


*
      write (*,*) 'calling std_height'
      CALL STD_HEIGHT (AXNUM, FTNUM,
     c MONUM, BUNUM, HTOT, NODES, STPTSUM,
     c FDHT, FTNDES, MDHT, MONDES,
     c BDHT, BUNDES)
*
        write (*,*) 'calling fueltemp_format'
        CALL FUELTEMP_FORMAT (STPTSUM, AXNUM, FTNUM,
     c  NODES, FTNDES, FTDAT, FTIN)
*
C       IF (RTYPE.EQ.'PWR') THE
C          write (*,*) 'calling modspecvol_format'
C          CALL MODSPECVOL_FORMAT (STPTSUM, AXNUM, MONUM,
C     c    NODES, MONDES, MODAT, MOIN)
C       ENDIF
*
      write (*,*) 'calling burnup_format'
      CALL BURNUP_FORMAT (STPTSUM, AXNUM, BUNUM,
     c NODES, BUNDES, BUDAT, BUIN)
*
      write (*,*) 'calling power_calcs'
      CALL POWER_CALCS (NBR, AXNUM, STPTSUM, STPTTALLY,
     c STPTS, GRAMS, FMASS, NODES, HTOT, BUIN,
     c STPTDAT, POWER, CYCLEN, STEPCONTROL, VARBLETDOWN,
     c VARSTEPNUM, VARPOWER)
*
      write (*,*) 'calling units_conversion'
      CALL UNITS_CONVERSION (STPTSUM, AXNUM, FTFINAL,
     c FTIN, MODDENFINAL, MOIN, PRESS, MODTEMPFINAL,
     c DENDAT, RTYPE, MODREFTEMP)
*
      write (*,*) 'calling execution_control'
      CALL EXECUTION_CONTROL (NBR, RELATIVE_STPT_NUM,
     c CT1, CT2, CT3, AXNUM, CYCPOS, AXBLANK,
     c BPDESID, CRINS, CRDES,
     c CRMIXNUM, CRMIXID, CRNUMISOS, CRISOID,
     c APSRINS, APSRMIXNUM, APSRMIXID,
     c RELATIVE_APSR_MIX_ID, APSRNUMISOS,
     c APSRISOID, ISN, IIM, ICM, IUS, PLEVEL,
     c BPZONE, BPMA, CRZONE, CRMA,
     c LMC, APSRZONE, APSRMA, LMD,
     c BPTN, BPBN, STPTS, APSRDES,
     c STPTDAT, AXBLANKRICH, GRAMS,
     c NODES, RODS, RICH, FTFINAL,
     c MODDENFINAL, MODTEMPFINAL,
     c BLETDOWN, BPWTPCT, BPDEN, CRDEN,
     c CRISOWTPCT, APSRDEN, APSRISOWTPCT,
     c PITCH, FOD, COD, CID, SZF, EPS, PTC, MESH,
     c BPRA, CRRA, LRC, APSRRA,
     c LRD, POWER, CYCDOWN, PREFIX,
     c NM, CYCLEID, REACT, LIB, AXBLANKET,
     c FUELCLAD, BPRFLAG, CRSTAT, APSRSTAT, FLAG2,
     c LUZONE, LMB, LRB, BPXSECT, BPRODS,
     c CT1START, CT2START, STPTTALLY, CLADTOT,
     c CLADDESNUM, CLADDESNAME, BPRCLAD, CRCLAD,
     c APSRCLAD, CLTEMP, BPMIXNUM, BPMIX, BPMIXID,
     c BPNUMISOS, BPISOID, BPISOWTPCT, UCSPACERFRAC,
```

```
     c SPACERMAT, STEPCONTROL, VARBLETDOWN, VARSTEPNUM,
     c VARPOWER, BPRFM, BPFMNUMISOS, BPFISOID,
     c ABOVEBPNUM, APSRFM, BPRFR, BPFISOWTPCT,
     c APSRFR, ABOVEBP, APSRFOLLOWMIX, RTYPE, MODREFDEN,
     c MODREFTEMP, CRMIXDEN,STOPFLAG,CYCSTOP,STPTSTOP,
     C PICKUPFLAG,FTNDES,GDROD,FIRSTMODEN)
*

C      close(99)

       END

****************************************************************
*     Reactor and Problem Data Acquisition Subroutine          *
****************************************************************
      SUBROUTINE DATA_AQUISITION (BPZONE, BPMA,
     c LMB, NLIB, PLEVEL, ISN, IIM, ICM, IUS, NBR, AXNUM,
     c FTNUM, MONUM, BUNUM, APSRINS,
     c APSRSTEPNUM, APSRMIXNUM, APSRMIXID, CRINS,
     c CRSTEPNUM, CRMIXNUM, CRMIXID, CRNUMISOS,
     c CRISOID, AXBLANK, AXBLANKNODNUM, STPTS,
     c CYCPOS, APSRNUMISOS, APSRISOID, STPTSUM,
     c BPRADESNUM, CRDESNUM, CRZONE, CRMA, LMC,
     c APSRDESNUM, APSRZONE, APSRMA, LMD,
     c BPCYCID, BPTN, BPBN, DES, BPCYCNUM, BPDESID,
     c CRDES, APSRDES, LMA, LUZONE,
     c CLTEMP, PRESS, BPDEN, BPRA, CRISOWTPCT,
     c LRA, LRB, MESH, SZF, EPS, PTC, APSRISOWTPCT,
     c NODES, BLETDOWN, AXBLANKRICH, STPTDAT,
     c FTNDES, FTDAT, MONDES, MODAT,
     c BUNDES, BUDAT, RICH, FMASS, RODS, CYCLEN,
     c PITCH, FOD, CID, COD, LENGTH, CYCDOWN, CRDEN,
     c CRRA, LRC, APSRDEN, APSRRA, LRD,
     c BPWTPCT, REACT, PREFIX, AXBLANKET, BPRFLAG,
     c FUELCLAD, FLAG2, CYCLEID, CRSTAT,
     c APSRSTAT, LIB, BPXSECT, BPRODS, CT1START,
     c CT2START, CLADTOT, CLADDESNUM, CLADDESNAME,
     c BPRCLAD, CRCLAD, APSRCLAD, BPMIXNUM, BPMIX, BPMIXID,
     c BPNUMISOS, BPISOID, BPISOWTPCT, UCSPACERFRAC,
     c SPACERMAT, STEPCONTROL, VARBLETDOWN, VARSTEPNUM,
     c BPRFM, BPFMNUMISOS, BPFISOID, ABOVEBPNUM, APSRFM,
     c BPRFR, BPFISOWTPCT, APSRFR, ABOVEBP, APSRFOLLOWMIX,
     c RTYPE, MODREFDEN, MODREFTEMP, CRMIXDEN,STOPFLAG,CYCSTOP,
     C STPTSTOP,PICKUPFLAG,GDROD,FIRSTMODEN)
*
      INTEGER*4 BPZONE(10), BPMA(15,10), LMA(15,10), LUZONE,
     c LMB(15), NLIB, PLEVEL, ISN, IIM, ICM, IUS, NBR, AXNUM,
     c FTNUM(20), MONUM(20), BUNUM(20), CT1, CT2, CT3,
     c APSRINS(10,20,23,50), APSRSTEPNUM, APSRCYC, APSRSTEP,
     c TOPN, BOTN, APSRMIX, APSRMIXNUM, APSRMIXID(25),
     c CRINS(10,20,23,50), CRSTEPNUM, CRCYC, CRSTEP, CYCHOLDER,
     c CRMIX, STPTHOLDER, CRMIXNUM, CRMIXID(25), CRNUMISOS(25),
     c CRISOID(25,20), AXBLANK(50), AXBLANKNODNUM, AXBLANKTEMP,
     c STPTS(10), CYCPOS(10), APSRNUMISOS(25), APSRISOID(25,10),
     c STPTSUM, BPRADESNUM, CRDESNUM, CRZONE(10), CRMA(15,10),
     c LMC(15,10), APSRDESNUM, APSRZONE(10), APSRMA(15,10),
     c LMD(15,10), BPCYCID, BPTN(10), BPBN(10), DES, BPCYCNUM,
     c BPDESID(10), CRDES(10,20,23,50), APSRDES(10,20,23,50),
     c BPRODS(10), CT1START, CT2START, APSRSTPT, CRSTPT,
     c CLADTOT, CLADDESNUM(10), BPRCLAD(10), CRCLAD(10),
     c APSRCLAD(10), BPMIXNUM, BPMIX(10), BPMIXID(10),GDROD,
     c BPNUMISOS(20), BPISOID(10,20), VARSTEPNUM(10,20),
     c BPRFM(15,10), BPFMNUMISOS(25), BPFISOID(25,10),
```

```
     c ABOVEBPNUM(10), APSRFM(15,10), FMIX, APSRFOLLOWMIX(10,20,23,50),
     c NUMOFSECTIONS, SECT,CYCSTOP,STPTSTOP
*
      REAL CLTEMP, PRESS, BPDEN(10), BPRA(15,10), CRISOWTPCT(25,20),
     c LRA(15,10), LRB(15), MESH, SZF, EPS, PTC, APSRISOWTPCT(25,10),
     c NODES(50,2), BLETDOWN(10,20,25), AXBLANKRICH, STPTDAT(10,20,3),
     c FTNDES(50,2,20), FTDAT(50,20), MONDES(50,2,20), MODAT(50,20),
     c BUNDES(50,2,20), BUDAT(50,20), RICH, FMASS, RODS, CYCLEN(10,2),
     c PITCH, FOD, CID, COD, LENGTH, CYCDOWN(10), CRDEN(10),
     c CRRA(15,10), LRC(15,10), APSRDEN(10), APSRRA(15,10), LRD(15,10),
     c BPWTPCT(10), BPXSECT(10), BPISOWTPCT(10,20), UCSPACERFRAC,
     c VARBLETDOWN(10,20,25,25), BPRFR(15,10), BPFISOWTPCT(25,10),
     c APSRFR(15,10), MODREFDEN, MODREFTEMP, CRMIXDEN(25),
     c FIRSTMODEN(20,20)
*
      CHARACTER REACT*23, PREFIX*3, AXBLANKET*1, BPRFLAG*1,
     c FUELCLAD*10, FLAG2*7, CYCLEID(10)*2, CRSTAT*6,STOPFLAG*1,
     c APSRSTAT*6, LIB*15, PICKUPFLAG*1, CLADFLAG*1, CLADDESNAME(10)*7,
     c SPACERMAT*7, STEPCONTROL*1, ABOVEBP(10)*5, RTYPE*3
*
*     Hardwired ASSYFOLLOW limitations:
********************************************************************************

*     Maximum number of irradiation steps in a given SAS2H input deck = 23.     *
*     Maximum number of isotopes in a CR or APSR material composition = 10.     *
*     Maximum number of concentric zones in a Path B Model = 15.                *
*     Maximum number of axial nodes in any axial format = 50.                   *
*     Maximum number of reactor cycles in which an assembly may be inserted = 10. *
*     Maximum number of CRC statepoints allowed in a given cycle = 20.          *
*     Maximum number of BPRA designs = 10.                                      *
*     Maximum number of CR absorber material mixtures = 25.                     *
*     Maximum number of CRA designs = 10.                                       *
*     Maximum number of APSR absorber material mixtures = 25.                   *
*     Maximum number of APSR assembly designs = 10.                             *
********************************************************************************
*
      OPEN (UNIT=10, FILE='datain', STATUS='OLD')
      REWIND (UNIT=10)

      READ (10,*)
      READ (10,*)
      READ (10,*)
      READ (10,*)

      READ (10,3) STOPFLAG
      IF(STOPFLAG.EQ.'Y')THEN
        READ (10,4) CYCSTOP
        READ (10,4) STPTSTOP
      ELSE
        CYCSTOP=0
        STPTSTOP=0
      ENDIF
    3 FORMAT(A1)
    4 FORMAT(I3)
      READ (10,2) PICKUPFLAG ! PICKUPFLAG is a signal to begin the assembly
*                              depletion and decay calculation at a point
*                              other than the beginning of the assembly's
*                              irradiation history as specified in the input
*                              deck
    2 FORMAT(A1)
      IF (PICKUPFLAG.EQ.'Y') THEN
         READ(10,*) CT1START ! cycle start
         READ(10,*) CT2START ! statepoint start
      ELSE
```

```
          CT1START=1
          CT2START=1
       ENDIF

C       IF (PICKUPFLAG.EQ.'N'.AND.STOPFLAG.EQ.'Y') THEN
C          READ(10,*) GDROD
C       ENDIF
C always read GDROD in

          READ(10,*) GDROD

*
       READ (10,10) REACT    ! REACT is the problem identification
*                              (up to 23 characters).
       READ (10,20) PREFIX   ! PREFIX is a 3 character prefix to be
*                              placed at the beginning of all SAS2H
*                              input decks produced.
       READ (10,35) RTYPE    ! RTYPE is a 3 character acronym to indentify
*                               the type of reactor (i.e. PWR, BWR)

       IF (RTYPE.EQ.'PWR') THEN
         write (*,*) ' '
         write (*,*) ' '
         write (*,*) '**************************************',
     &               '**************************************',
     &               '* WARNING: This version of CRAFT is  *',
     &               '* only for BWR reactors and can not  *',
     &               '* be used for PWR depletion          *',
     &               '* calculations. Further execution of *',
     &               '* this software routine will be      *',
     &               '* cancelled.                         *',
     &               '**************************************',
     &               '**************************************'
         write (*,*) ' '
         write (*,*) ' '
         GOTO 999
       ENDIF

       READ (10,40) LIB      ! LIB is a 15 character identification
*                              of the cross-section library requested
*                              for use in the SCALE code system.
   10 FORMAT (A23)
   20 FORMAT (A3)
   30 FORMAT (A2)
   35 FORMAT (A3)
   40 FORMAT (A15)
*
*      Fuel Batch Data Acquisition
*
       READ (10,*) RICH    ! RICH is the fuel assy wt% U-235 in UO2
*                            enrichment.
       READ (10,*) FMASS   ! FMASS is the fuel assy loading of
*                            uranium in g/assy.
       READ (10,*) RODS    ! RODS is the number of fuel rods in the assy.
       READ (10,*) PITCH   ! PITCH is the fuel rod pitch in the assy.
       READ (10,*) FOD     ! FOD is the fuel rod outer diameter in cm.
       READ (10,*) CID     ! CID is the clad inner diameter in cm.
       READ (10,*) COD     ! COD is the clad outer diameter in cm.
       READ (10,*) LENGTH ! LENGTH is the active fuel length in cm.
       READ (10,70) AXBLANKET ! Flag for axial blanket modelling.
   70 FORMAT(A1)
       IF (AXBLANKET.EQ.'Y') THEN
          READ (10,*) AXBLANKRICH ! Axial blanket fuel U-235 enrichment.
*         Initialize AXBLANK array
```

```
         DO 80 CT1=1,50
            AXBLANK(CT1)=0
   80    CONTINUE
*        Gather data for AXBLANK array
         READ (10,*) AXBLANKNODNUM ! Number of nodes with axial
*                                    blanket fuel.
         DO 90 CT1=1,AXBLANKNODNUM
            READ (10,*) AXBLANKTEMP ! Node containing axial
*                                     blanket fuel.
            AXBLANK(AXBLANKTEMP)=1 ! Identify axial blanket fuel
*                                    node location in AXBLANK.
   90    CONTINUE
      ENDIF
*     Spacer data acquisition
      READ (10,92) SPACERMAT
   92 FORMAT(A7)
      READ (10,*) UCSPACERFRAC
*     Cladding data acquisition
      READ (10,100) FUELCLAD
  100 FORMAT (A10)
      READ (10,*) CLTEMP
      READ (10,101) CLADFLAG
  101 FORMAT (A1)
      IF (CLADFLAG.EQ.'Y') THEN
         READ(10,*) CLADTOT
         DO 108 CT1=1,CLADTOT
            READ(10,*) CLADDESNUM(CT1)
            READ(10,105) CLADDESNAME(CT1)
  105       FORMAT (A7)
  108    CONTINUE
      ENDIF
*
*     System Pressure
      IF (RTYPE.EQ.'PWR') THEN
         READ (10,*) PRESS
      ELSEIF (RTYPE.EQ.'BWR') THEN
         READ (10,*) MODREFDEN
         READ (10,*) MODREFTEMP
      ENDIF
      READ (10,110) BPRFLAG
  110 FORMAT (A1)
      IF (BPRFLAG.EQ.'Y') THEN
         READ(10,*) BPCYCNUM  ! Number of cycles with BPRA
         READ(10,*) BPRADESNUM, BPMIXNUM
         DO 145 CT2=1,BPRADESNUM
*           Get BP density, B4C wt% in Al2O3-B4C,
*           BP x-sectional area, # BP rods, and BPR clad mix num
            READ (10,*) BPDEN(CT2), BPWTPCT(CT2), BPXSECT(CT2),
     c      BPRODS(CT2), BPRCLAD(CT2), BPMIX(CT2)
*           Larger BPRA unit cell data acquisition
            READ (10,*) BPZONE(CT2)
            DO 112 CT1=1,BPZONE(CT2)
               READ (10,*) BPMA(CT1,CT2), BPRA(CT1,CT2)
  112       CONTINUE
*           Larger standard unit cell for use with BPRAs
            DO 114 CT1=1,BPZONE(CT2)
               READ (10,*) LMA(CT1,CT2), LRA(CT1,CT2)
  114       CONTINUE
            DO 116 CT1=1,BPZONE(CT2)
               READ(10,*) BPRFM(CT1,CT2), BPRFR(CT1,CT2)
  116       CONTINUE
            READ(10,118) ABOVEBP(CT2), ABOVEBPNUM(CT2)
  118       FORMAT (A5,1X,I3)
            IF (ABOVEBP(CT2).NE.'AL2O3') THEN
```

```
            READ (10,*) BPFMNUMISOS(CT2)
            DO 120 CT1=1,BPFMNUMISOS(CT2)
               READ (10,*) BPFISOID(CT2,CT1),
     c         BPFISOWTPCT(CT2,CT1)
 120        CONTINUE
         ENDIF
 145    CONTINUE
        DO 147 CT1=1,10
           DO 146 CT2=1,20
              BPISOID(CT1,CT2)=0
              BPISOWTPCT(CT1,CT2)=0.0
 146       CONTINUE
 147    CONTINUE
        IF (BPMIXNUM.NE.0) THEN
           DO 150 CT1=1,BPMIXNUM
              READ (10,*) BPMIXID(CT1) ! SAS2H Mixture ID for CR
              READ (10,*) BPNUMISOS(CT1)
              DO 149 CT2=1,BPNUMISOS(CT1)
                 READ (10,*) BPISOID(CT1,CT2), BPISOWTPCT(CT1,CT2)
 149          CONTINUE
 150       CONTINUE
        ENDIF
        DO 156 CT1=1,10
           BPDESID(CT1)=0
 156    CONTINUE
        DO 157 CT1=1,BPCYCNUM
           READ(10,*) BPCYCID, BPDESID(BPCYCID), BPTN(BPCYCID),
     c     BPBN(BPCYCID)
 157    CONTINUE
      ENDIF
*     Larger standard unit cell
      READ (10,*) LUZONE
      DO 170 CT1=1,LUZONE
         READ (10,*) LMB(CT1), LRB(CT1)
 170  CONTINUE
*     Contol parameter data acquisition
      READ (10,*) NLIB
      READ (10,*) PLEVEL
      READ (10,*) MESH
      READ (10,180) FLAG2
 180  FORMAT (A7)
      IF (FLAG2.EQ.'SPECIAL') THEN
         READ (10,*) SZF
         READ (10,*) ISN
         READ (10,*) IIM
         READ (10,*) ICM
         READ (10,*) EPS
         READ (10,*) PTC
         READ (10,*) IUS
      ENDIF
*     Reactor history data acquisition
      READ (10,*) NBR
      DO 210 CT1=1,NBR
         READ (10,190) CYCLEID(CT1)
 190     FORMAT (A2)
         READ (10,*) STPTS(CT1)
         DO 200 CT2=1,STPTS(CT1)
            READ (10,*) STPTDAT(CT1,CT2,1)
            READ (10,*) STPTDAT(CT1,CT2,2)
            READ (10,*) STPTDAT(CT1,CT2,3)
 200     CONTINUE
         READ (10,*) CYCDOWN(CT1)
         READ (10,*) CYCLEN(CT1,1)
         READ (10,*) CYCLEN(CT1,2)
```

```
          READ (10,*) CYCPOS(CT1)
  210 CONTINUE
      STEPCONTROL='N'
      READ (10,212) STEPCONTROL
  212 FORMAT(A1)
*
*     Note that the BLETDOWN and VARBLETDOWN variables will carry
*     boron letdown data for CRAFT calculations performed on PWR
*     reactors, but will carry moderator density information for
*     calculations performed on BWRs.
*
      IF (STEPCONTROL.EQ.'N') THEN
         DO 220 CT1=1,NBR
            READ (10,*) CYCHOLDER
            DO 217 CT2=1,STPTS(CYCHOLDER)
               READ (10,*) STPTHOLDER
               READ (10,*) BLETDOWN(CYCHOLDER,STPTHOLDER,1)
               READ (10,*) BLETDOWN(CYCHOLDER,STPTHOLDER,2)
               DO 213 CT3=3,(INT(BLETDOWN(CYCHOLDER,STPTHOLDER,2))+2)
                  READ (10,*) BLETDOWN(CYCHOLDER,STPTHOLDER,CT3)
  213          CONTINUE
  217       CONTINUE
  220    CONTINUE
      ELSEIF (STEPCONTROL.EQ.'Y') THEN
         DO 240 CT1=1,NBR
            READ (10,*) CYCHOLDER
            DO 235 CT2=1,STPTS(CYCHOLDER)
               READ (10,*) STPTHOLDER
               READ (10,*) VARSTEPNUM(CYCHOLDER,STPTHOLDER)
               DO 230 CT3=1,VARSTEPNUM(CYCHOLDER,STPTHOLDER)
                  READ (10,*) VARBLETDOWN(CYCHOLDER,STPTHOLDER,CT3,1),
     c            VARBLETDOWN(CYCHOLDER,STPTHOLDER,CT3,2)
                  IF(CT3.EQ.1)THEN
                    FIRSTMODEN(CT1,CT2)=
     C              VARBLETDOWN(CYCHOLDER,STPTHOLDER,CT3,2)
c                  ELSE
c                    FIRSTMODEN(CT1,CT2)=0.0
                  ENDIF
c                WRITE(99,*)'CT1,CT2,CT3 ',CT1,CT2,CT3
c                WRITE(99,*)'FIRSTMODEN(CT1,CT2) ',FIRSTMODEN(CT1,CT2)
  230          CONTINUE
  235       CONTINUE
  240    CONTINUE
      ENDIF
      READ (10,*) AXNUM
      DO 260 CT1=1,AXNUM
         READ (10,250) NODES(CT1,1), NODES(CT1,2)
  250    FORMAT (F3.0,1X,F10.7)
  260 CONTINUE
*     Control Rod Data Aquisition
      READ (10,270) CRSTAT
  270 FORMAT (A6)
      IF (CRSTAT.EQ.'RODDED') THEN
         DO 300 CT1=1,10
            DO 295 CT2=1,20
               DO 290 CT3=1,23
                  DO 280 CT4=1,50
                     CRINS(CT1,CT2,CT3,CT4)=0
  280             CONTINUE
  290          CONTINUE
  295       CONTINUE
  300    CONTINUE
         READ (10,*) CRSTEPNUM ! Number of pre-defined irradiation steps
*                                in which the assembly contains a control
```

```
*                                  rod assembly.
        DO 320 CT1=1,CRSTEPNUM
           READ (10,*) NUMOFSECTIONS ! Number of axial sections of the fuel
*                                      assembly which have a rod assembly inserted.
           DO 315 SECT=1,NUMOFSECTIONS
              READ (10,*) CRCYC, CRSTPT, CRSTEP, TOPN,
     c                    BOTN, CRMIX, DES
              DO 310 CT2=TOPN,BOTN
                 CRINS(CRCYC,CRSTPT,CRSTEP,CT2)=CRMIX
                 CRDES(CRCYC,CRSTPT,CRSTEP,CT2)=DES
  310         CONTINUE
  315      CONTINUE
  320   CONTINUE
      READ (10,*) CRMIXNUM
      DO 340 CT1=1,CRMIXNUM
       IF (RTYPE.EQ.'PWR') THEN
         READ (10,*) CRMIXID(CT1) ! SAS2H Mixture ID for CR
       ELSEIF (RTYPE.EQ.'BWR') THEN
         READ (10,*) CRMIXID(CT1), CRMIXDEN(CT1) ! SAS2H Mixture ID for CR
       ENDIF
         READ (10,*) CRNUMISOS(CT1)
         DO 330 CT2=1,CRNUMISOS(CT1)
            READ(10,*) CRISOID(CT1,CT2), CRISOWTPCT(CT1,CT2)
  330    CONTINUE
  340 CONTINUE
      READ(10,*) CRDESNUM
      DO 349 CT2=1,CRDESNUM
         READ(10,*) CRDEN(CT2), CRCLAD(CT2)
         READ(10,*) CRZONE(CT2)
         DO 344 CT1=1,CRZONE(CT2)
            READ(10,*) CRMA(CT1,CT2), CRRA(CT1,CT2)
  344    CONTINUE
         DO 347 CT1=1,CRZONE(CT2)
            READ(10,*) LMC(CT1,CT2), LRC(CT1,CT2)
  347    CONTINUE
  349 CONTINUE
      ENDIF
*     Axial Power Shaping Rod Data Aquisition
      READ (10,350) APSRSTAT
  350 FORMAT (A6)
      IF (APSRSTAT.EQ.'RODDED') THEN
         DO 380 CT1=1,10
            DO 375 CT2=1,20
               DO 370 CT3=1,23
                  DO 360 CT4=1,50
                     APSRINS(CT1,CT2,CT3,CT4)=0
  360             CONTINUE
  370          CONTINUE
  375       CONTINUE
  380    CONTINUE
         READ (10,*) APSRSTEPNUM ! Number of pre-defined irradiation steps
*                                  in which the assembly contains an axial
*                                  power shaping rod assembly.
         DO 400 CT1=1,APSRSTEPNUM
            READ (10,*) APSRCYC, APSRSTPT, APSRSTEP, TOPN, BOTN,
     c      APSRMIX, DES, FMIX
            DO 390 CT2=TOPN,BOTN
               APSRINS(APSRCYC,APSRSTPT,APSRSTEP,CT2)=APSRMIX
               APSRDES(APSRCYC,APSRSTPT,APSRSTEP,CT2)=DES
               APSRFOLLOWMIX(APSRCYC,APSRSTPT,APSRSTEP,CT2)=FMIX
  390       CONTINUE
  400    CONTINUE
      READ (10,*) APSRMIXNUM
      DO 410 CT1=1,APSRMIXNUM
```

```
          READ (10,*) APSRMIXID(CT1) ! SAS2H Mixture ID for APSR's
          READ (10,*) APSRNUMISOS(CT1)
          DO 405 CT2=1,APSRNUMISOS(CT1)
             READ(10,*) APSRISOID(CT1,CT2), APSRISOWTPCT(CT1,CT2)
 405      CONTINUE
 410   CONTINUE
       READ(10,*) APSRDESNUM
       DO 429 CT2=1,APSRDESNUM
          READ(10,*) APSRDEN(CT2), APSRCLAD(CT2)
          READ(10,*) APSRZONE(CT2)
          DO 412 CT1=1,APSRZONE(CT2)
             READ(10,*) APSRMA(CT1,CT2), APSRRA(CT1,CT2)
 412      CONTINUE
          DO 414 CT1=1,APSRZONE(CT2)
             READ(10,*) LMD(CT1,CT2), LRD(CT1,CT2)
 414      CONTINUE
          DO 416 CT1=1,APSRZONE(CT2)
             READ(10,*) APSRFM(CT1,CT2), APSRFR(CT1,CT2)
 416      CONTINUE
 429   CONTINUE
       ENDIF
       STPTSUM=0
       DO 430 CT1=1,10
          STPTSUM=STPTSUM+STPTS(CT1)
 430   CONTINUE
*      Acquisition of fuel temperature data for each node
          DO 470 CT1=1,(STPTSUM-1)
             READ (10,*) FTNUM(CT1)
             DO 450 CT2=1,FTNUM(CT1)
               READ (10,440) FTNDES(CT2,1,CT1), FTNDES(CT2,2,CT1)
 440           FORMAT (F3.0,1X,F10.7)
 450         CONTINUE
             DO 460 CT2=1,FTNUM(CT1)
                READ (10,*) FTDAT(CT2,CT1)
 460         CONTINUE
 470      CONTINUE
       IF (RTYPE.EQ.'PWR') THEN
*      Acquisition of moderator specific volume data for each node
          DO 510 CT1=1,(STPTSUM-1)
             READ (10,*) MONUM(CT1)
             DO 490 CT2=1,MONUM(CT1)
                READ (10,480) MONDES(CT2,1,CT1), MONDES(CT2,2,CT1)
 480            FORMAT (F3.0,1X,F10.7)
 490         CONTINUE
             DO 500 CT2=1,MONUM(CT1)
                READ (10,*) MODAT(CT2,CT1)
 500         CONTINUE
 510      CONTINUE
       ENDIF
*      Acquisition of nodal burnup data for each statepoint in each cycle
          DO 550 CT1=1,STPTSUM
             READ (10,*) BUNUM(CT1)
             DO 530 CT2=1,BUNUM(CT1)
                READ (10,520) BUNDES(CT2,1,CT1), BUNDES(CT2,2,CT1)
 520            FORMAT (F3.0,1X,F10.7)
 530         CONTINUE
             DO 540 CT2=1,BUNUM(CT1)
                READ (10,*) BUDAT(CT2,CT1)
 540         CONTINUE
 550      CONTINUE

       RETURN
 999   STOP
       END
```

```
****************************************************************
*   Subroutine to standardize the assembly height to          *
*   the desired CRC assembly height.                          *
****************************************************************
      SUBROUTINE STD_HEIGHT (AXNUM, FTNUM,
     c MONUM, BUNUM, HTOT, NODES, STPTSUM,
     c FDHT, FTNDES, MDHT, MONDES,
     c BDHT, BUNDES)
*
      INTEGER*4 AXNUM, CT1, CT2, FTNUM(20), MONUM(20), BUNUM(20),
     c STPTSUM
*
      REAL HTOT, NODES(50,2), FDHT(20), FTNDES(50,2,20),
     c MDHT(20), MONDES(50,2,20), BDHT(20), BUNDES(50,2,20)
*
*
      HTOT=0
      DO 10 CT1=1,AXNUM
         HTOT=HTOT+NODES(CT1,2)
   10 CONTINUE
      DO 30 CT1=1,STPTSUM
         FDHT(CT1)=0
         DO 20 CT2=1,FTNUM(CT1)
            FDHT(CT1)=FDHT(CT1)+FTNDES(CT2,2,CT1)
   20    CONTINUE
   30 CONTINUE
      DO 50 CT1=1,STPTSUM
         MDHT(CT1)=0
         DO 40 CT2=1,MONUM(CT1)
            MDHT(CT1)=MDHT(CT1)+MONDES(CT2,2,CT1)
   40    CONTINUE
   50 CONTINUE
      DO 70 CT1=1,STPTSUM
         BDHT(CT1)=0
         DO 60 CT2=1,BUNUM(CT1)
            BDHT(CT1)=BDHT(CT1)+BUNDES(CT2,2,CT1)
   60    CONTINUE
   70 CONTINUE
      DO 90 CT1=1,STPTSUM
         DO 80 CT2=1,FTNUM(CT1)
            FTNDES(CT2,2,CT1)=FTNDES(CT2,2,CT1)*(HTOT/FDHT(CT1))
   80    CONTINUE
   90 CONTINUE
      DO 110 CT1=1,STPTSUM
         DO 100 CT2=1,MONUM(CT1)
            MONDES(CT2,2,CT1)=MONDES(CT2,2,CT1)*(HTOT/MDHT(CT1))
  100    CONTINUE
  110 CONTINUE
      DO 130 CT1=1,STPTSUM
         DO 120 CT2=1,BUNUM(CT1)
            BUNDES(CT2,2,CT1)=BUNDES(CT2,2,CT1)*(HTOT/BDHT(CT1))
  120    CONTINUE
  130 CONTINUE

      RETURN
      END

********************************************************************
*   Subroutine to convert fuel temperature input nodal formats    *
*   into the requested CRC nodal format                           *
********************************************************************
```

```
      SUBROUTINE FUELTEMP_FORMAT (STPTSUM, AXNUM, FTNUM,
     c NODES, FTNDES, FTDAT, FTIN)
*
      INTEGER*4 CT1, CT2, CT3, STPTSUM, AXNUM, FTNUM(20)
*
      REAL HCTOLD, HCT, SUM, NODES(50,2), FTHOLD, FTHCT,
     c FTNDES(50,2,20), FTDAT(50,20), FTIN(50,20)
*
      DO 30 CT1=1,STPTSUM
         HCTOLD=0
         HCT=0
         DO 20 CT2=1,AXNUM
            SUM=0
            HCTOLD=HCT
            HCT=HCT+NODES(CT2,2)
            FTHOLD=0
            FTHCT=0
            DO 10 CT3=1,FTNUM(CT1)
               FTHOLD=FTHCT
               FTHCT=FTHCT+FTNDES(CT3,2,CT1)
               IF ((FTHOLD.LT.HCTOLD).AND.(FTHCT.GT.HCTOLD).AND.
     c            (FTHCT.LT.HCT)) THEN
                  SUM=SUM+(((FTHCT-HCTOLD)/NODES(CT2,2))
     c               *FTDAT(CT3,CT1))
               ENDIF
               IF ((FTHOLD.EQ.HCTOLD).AND.(FTHCT.EQ.HCT)) THEN
                  SUM=SUM+FTDAT(CT3,CT1)
               ENDIF
               IF ((FTHOLD.GT.HCTOLD).AND.(FTHOLD.LT.HCT).AND.
     c            (FTHCT.GT.HCT)) THEN
                  SUM=SUM+(((HCT-FTHOLD)/NODES(CT2,2))
     c               *FTDAT(CT3,CT1))
               ENDIF
               IF ((FTHOLD.EQ.HCTOLD).AND.(FTHCT.GT.HCTOLD).AND.
     c            (FTHCT.LT.HCT)) THEN
                  SUM=SUM+(((FTHCT-FTHOLD)/NODES(CT2,2))
     c               *FTDAT(CT3,CT1))
               ENDIF
               IF ((FTHOLD.GT.HCTOLD).AND.(FTHCT.LT.HCT)) THEN
                  SUM=SUM+(((FTHCT-FTHOLD)/NODES(CT2,2))
     c               *FTDAT(CT3,CT1))
               ENDIF
               IF ((FTHOLD.GT.HCTOLD).AND.(FTHOLD.LT.HCT).AND.
     c            (FTHCT.EQ.HCT)) THEN
                  SUM=SUM+(((FTHCT-FTHOLD)/NODES(CT2,2))
     c               *FTDAT(CT3,CT1))
               ENDIF
               IF ((FTHOLD.LT.HCTOLD).AND.(FTHCT.EQ.HCT)) THEN
                  SUM=SUM+FTDAT(CT3,CT1)
               ENDIF
               IF ((FTHOLD.LT.HCTOLD).AND.(FTHCT.GT.HCT)) THEN
                  SUM=SUM+FTDAT(CT3,CT1)
               ENDIF
               IF ((FTHOLD.EQ.HCTOLD).AND.(FTHCT.GT.HCT)) THEN
                  SUM=SUM+FTDAT(CT3,CT1)
               ENDIF
   10       CONTINUE
            FTIN(CT2,CT1)=SUM
   20    CONTINUE
   30 CONTINUE

      RETURN
      END
```

```
*****************************************************************
*   Subroutine to convert moderator specific volume input nodal   *
*   formats into the requested CRC nodal format                   *
*****************************************************************
      SUBROUTINE MODSPECVOL_FORMAT (STPTSUM, AXNUM, MONUM,
     c NODES, MONDES, MODAT, MOIN)
*
      INTEGER*4 CT1, CT2, CT3, STPTSUM, AXNUM, MONUM(20)
*
      REAL HCTOLD, HCT, SUM, NODES(50,2), MOHOLD, MOHCT,
     c MONDES(50,2,20), MODAT(50,20), MOIN(50,20)
*
      DO 30 CT1=1,STPTSUM
         HCTOLD=0
         HCT=0
         DO 20 CT2=1,AXNUM
            SUM=0
            HCTOLD=HCT
            HCT=HCT+NODES(CT2,2)
            MOHOLD=0
            MOHCT=0
            DO 10 CT3=1,MONUM(CT1)
               MOHOLD=MOHCT
               MOHCT=MOHCT+MONDES(CT3,2,CT1)
               IF ((MOHOLD.LT.HCTOLD).AND.(MOHCT.GT.HCTOLD).AND.
     c            (MOHCT.LT.HCT)) THEN
                  SUM=SUM+(((MOHCT-HCTOLD)/NODES(CT2,2))
     c               *MODAT(CT3,CT1))
               ENDIF
               IF ((MOHOLD.EQ.HCTOLD).AND.(MOHCT.EQ.HCT)) THEN
                  SUM=SUM+MODAT(CT3,CT1)
               ENDIF
               IF ((MOHOLD.GT.HCTOLD).AND.(MOHOLD.LT.HCT).AND.
     c            (MOHCT.GT.HCT)) THEN
                  SUM=SUM+(((HCT-MOHOLD)/NODES(CT2,2))
     c               *MODAT(CT3,CT1))
               ENDIF
               IF ((MOHOLD.EQ.HCTOLD).AND.(MOHCT.GT.HCTOLD).AND.
     c            (MOHCT.LT.HCT)) THEN
                  SUM=SUM+(((MOHCT-MOHOLD)/NODES(CT2,2))
     c               *MODAT(CT3,CT1))
               ENDIF
               IF ((MOHOLD.GT.HCTOLD).AND.(MOHCT.LT.HCT)) THEN
                  SUM=SUM+(((MOHCT-MOHOLD)/NODES(CT2,2))
     c               *MODAT(CT3,CT1))
               ENDIF
               IF ((MOHOLD.GT.HCTOLD).AND.(MOHOLD.LT.HCT).AND.
     c            (MOHCT.EQ.HCT)) THEN
                  SUM=SUM+(((MOHCT-MOHOLD)/NODES(CT2,2))
     c               *MODAT(CT3,CT1))
               ENDIF
               IF ((MOHOLD.LT.HCTOLD).AND.(MOHCT.EQ.HCT)) THEN
                  SUM=SUM+MODAT(CT3,CT1)
               ENDIF
               IF ((MOHOLD.LT.HCTOLD).AND.(MOHCT.GT.HCT)) THEN
                  SUM=SUM+MODAT(CT3,CT1)
               ENDIF
               IF ((MOHOLD.EQ.HCTOLD).AND.(MOHCT.GT.HCT)) THEN
                  SUM=SUM+MODAT(CT3,CT1)
               ENDIF
 10         CONTINUE
            MOIN(CT2,CT1)=SUM
 20      CONTINUE
```

```
   30 CONTINUE

      RETURN
      END

*****************************************************************
*   Subroutine to convert burnup input nodal formats into the   *
*   requested CRC nodal format                                  *
*****************************************************************
      SUBROUTINE BURNUP_FORMAT (STPTSUM, AXNUM, BUNUM,
     c NODES, BUNDES, BUDAT, BUIN)
*
      INTEGER*4 CT1, CT2, CT3, STPTSUM, AXNUM, BUNUM(20)
*
      REAL HCTOLD, HCT, SUM, NODES(50,2), BUHOLD, BUHCT,
     c BUNDES(50,2,20), BUDAT(50,20), BUIN(50,20)
*
      DO 30 CT1=1,STPTSUM
         HCTOLD=0
         HCT=0
         DO 20 CT2=1,AXNUM
            SUM=0
            HCTOLD=HCT
            HCT=HCT+NODES(CT2,2)
            BUHOLD=0
            BUHCT=0
            DO 10 CT3=1,BUNUM(CT1)
               BUHOLD=BUHCT
               BUHCT=BUHCT+BUNDES(CT3,2,CT1)
               IF ((BUHOLD.LT.HCTOLD).AND.(BUHCT.GT.HCTOLD).AND.
     c            (BUHCT.LT.HCT)) THEN
                  SUM=SUM+(((BUHCT-HCTOLD)/NODES(CT2,2))
     c               *BUDAT(CT3,CT1))
               ENDIF
               IF ((BUHOLD.EQ.HCTOLD).AND.(BUHCT.EQ.HCT)) THEN
                  SUM=SUM+BUDAT(CT3,CT1)
               ENDIF
               IF ((BUHOLD.GT.HCTOLD).AND.(BUHOLD.LT.HCT).AND.
     c            (BUHCT.GT.HCT)) THEN
                  SUM=SUM+(((HCT-BUHOLD)/NODES(CT2,2))
     c               *BUDAT(CT3,CT1))
               ENDIF
               IF ((BUHOLD.EQ.HCTOLD).AND.(BUHCT.GT.HCTOLD).AND.
     c            (BUHCT.LT.HCT)) THEN
                  SUM=SUM+(((BUHCT-BUHOLD)/NODES(CT2,2))
     c               *BUDAT(CT3,CT1))
               ENDIF
               IF ((BUHOLD.GT.HCTOLD).AND.(BUHCT.LT.HCT)) THEN
                  SUM=SUM+(((BUHCT-BUHOLD)/NODES(CT2,2))
     c               *BUDAT(CT3,CT1))
               ENDIF
               IF ((BUHOLD.GT.HCTOLD).AND.(BUHOLD.LT.HCT).AND.
     c            (BUHCT.EQ.HCT)) THEN
                  SUM=SUM+(((BUHCT-BUHOLD)/NODES(CT2,2))
     c               *BUDAT(CT3,CT1))
               ENDIF
               IF ((BUHOLD.LT.HCTOLD).AND.(BUHCT.EQ.HCT)) THEN
                  SUM=SUM+BUDAT(CT3,CT1)
               ENDIF
               IF ((BUHOLD.LT.HCTOLD).AND.(BUHCT.GT.HCT)) THEN
                  SUM=SUM+BUDAT(CT3,CT1)
               ENDIF
               IF ((BUHOLD.EQ.HCTOLD).AND.(BUHCT.GT.HCT)) THEN
```

```
              SUM=SUM+BUDAT(CT3,CT1)
           ENDIF
10       CONTINUE
         BUIN(CT2,CT1)=SUM
20     CONTINUE
30  CONTINUE


    RETURN
    END

************************************************************************
*     Subroutine to calculate nodal powers for each reactor cycle       *
************************************************************************
      SUBROUTINE POWER_CALCS (NBR, AXNUM, STPTSUM, STPTTALLY,
     c STPTS, GRAMS, FMASS, NODES, HTOT, BUIN,
     c STPTDAT, POWER, CYCLEN, STEPCONTROL, VARBLETDOWN,
     c VARSTEPNUM, VARPOWER)
*
      INTEGER*4 CT1, NBR, AXNUM, CT2, CT3, CYCLENUMBER, STPTNUMBER,
     c STPTSUM, STPTTALLY(20), STPTS(10), VARSTEPNUM(10,20), CT4
*
      REAL GRAMS(50), FMASS, NODES(50,2), HTOT, BURN, BUIN(50,20),
     c DAYS, STPTDAT(10,20,3), POWER(50,20), CYCLEN(10,2),
     c VARPOWER(10,20,25,50), VARBLETDOWN(10,20,25,25),
     c TOTALBURNDAYS
*
      CHARACTER STEPCONTROL*1
*
      DO 10 CT1=1,10
         STPTTALLY(CT1)=0
   10 CONTINUE
      STPTTALLY(1)=STPTS(1)
      IF (NBR.GE.2) THEN
         DO 20 CT1=2,NBR
            STPTTALLY(CT1)=STPTTALLY(CT1-1)+STPTS(CT1)
   20    CONTINUE
      ENDIF
      IF (STEPCONTROL.EQ.'N') THEN
       DO 50 CT1=1,AXNUM
         GRAMS(CT1)=FMASS*(NODES(CT1,2)/HTOT)
         DO 40 CT2=1,(STPTSUM-1)
            BURN=BUIN(CT1,(CT2+1))-BUIN(CT1,CT2)
            IF (NBR.GE.2) THEN
               DO 30 CT3=2,NBR
                  IF((CT2.LE.STPTTALLY(CT3)).AND.
     c                (CT2.GT.STPTTALLY(CT3-1))) THEN
                     CYCLENUMBER=CT3
                  ELSEIF (CT2.LE.STPTTALLY(1)) THEN
                     CYCLENUMBER=1
                  ENDIF
   30          CONTINUE
            ELSEIF (NBR.EQ.1) THEN
               CYCLENUMBER=1
            ENDIF
            IF (CYCLENUMBER.EQ.1) THEN
               STPTNUMBER=CT2
            ELSEIF (CYCLENUMBER.GT.1) THEN
               STPTNUMBER=CT2-STPTTALLY(CYCLENUMBER-1)
            ENDIF
            IF (STPTNUMBER.EQ.STPTS(CYCLENUMBER)) THEN
               DAYS=CYCLEN(CYCLENUMBER,1)-
     c         STPTDAT(CYCLENUMBER,STPTNUMBER,1)
```

```
            ELSE
               DAYS=STPTDAT(CYCLENUMBER,(STPTNUMBER+1),1)-
     c         STPTDAT(CYCLENUMBER,STPTNUMBER,1)
            ENDIF
            POWER(CT1,CT2)=BURN*GRAMS(CT1)*(1.0/1000.0)*(1/DAYS)
   40    CONTINUE
   50  CONTINUE
      ELSEIF (STEPCONTROL.EQ.'Y') THEN
       DO 100 CT1=1,AXNUM
         GRAMS(CT1)=FMASS*(NODES(CT1,2)/HTOT)
         DO 90 CT2=1,(STPTSUM-1)
            IF (NBR.GE.2) THEN
               DO 70 CT3=2,NBR
                  IF((CT2.LE.STPTTALLY(CT3)).AND.
     c               (CT2.GT.STPTTALLY(CT3-1))) THEN
                     CYCLENUMBER=CT3
                  ELSEIF (CT2.LE.STPTTALLY(1)) THEN
                     CYCLENUMBER=1
                  ENDIF
   70          CONTINUE
            ELSEIF (NBR.EQ.1) THEN
               CYCLENUMBER=1
            ENDIF
            IF (CYCLENUMBER.EQ.1) THEN
               STPTNUMBER=CT2
            ELSEIF (CYCLENUMBER.GT.1) THEN
               STPTNUMBER=CT2-STPTTALLY(CYCLENUMBER-1)
            ENDIF
            TOTALBURNDAYS=0.0
            DO 75 CT4=1,VARSTEPNUM(CYCLENUMBER,STPTNUMBER)
               TOTALBURNDAYS=TOTALBURNDAYS+
     c         VARBLETDOWN(CYCLENUMBER,STPTNUMBER,CT4,1)
   75       CONTINUE
            DO 80 CT4=1,VARSTEPNUM(CYCLENUMBER,STPTNUMBER)
               DAYS=VARBLETDOWN(CYCLENUMBER,STPTNUMBER,CT4,1)
               BURN=(BUIN(CT1,(CT2+1))-BUIN(CT1,CT2))*
     c         (DAYS/TOTALBURNDAYS)
               VARPOWER(CYCLENUMBER,STPTNUMBER,CT4,CT1)=BURN*
     c         GRAMS(CT1)*(1.0/1000.0)*(1/DAYS)
   80       CONTINUE
   90    CONTINUE
  100  CONTINUE
      ENDIF

      RETURN
      END


***********************************************************************
*     Subroutine to convert fuel temperature units and calculate      *
*     moderator specific volumes and densities with the correct units *
***********************************************************************
      SUBROUTINE UNITS_CONVERSION (STPTSUM, AXNUM, FTFINAL,
     c FTIN, MODDENFINAL, MOIN, PRESS, MODTEMPFINAL, DENDAT, RTYPE,
     c MODREFTEMP)
*
      INTEGER*4 CT1, CT2, CT3, STPTSUM, AXNUM, COL1, COL2, ROW1, ROW2
*
      REAL FTFINAL(50,20), FTIN(50,20), MODDENFINAL(50,20), MOIN(50,20),
     c PRESS, DENDAT(29,10), P1, P2, DENCOL(29), T, MODTEMPFINAL(50,20),
     c MODREFTEMP
*
      CHARACTER RTYPE*3
*
```

```
      DO 50 CT1=1,STPTSUM
         DO 40 CT2=1,AXNUM
         IF(RTYPE.EQ.'PWR')THEN
            FTFINAL(CT2,CT1)=((FTIN(CT2,CT1)-32.0)*(5.0/9.0))
     c                       +273.15
         ELSE
            FTFINAL(CT2,CT1)=FTIN(CT2,CT1)
         ENDIF
         IF (RTYPE.EQ.'PWR') THEN
            MODDENFINAL(CT2,CT1)=(1/(MOIN(CT2,CT1)*62.42691))
            DO 10 CT3=2,10
               IF ((PRESS.LT.DENDAT(1,CT3)).AND.
     c         (PRESS.GT.DENDAT(1,(CT3+1)))) THEN
                  P1=DENDAT(1,CT3)
                  P2=DENDAT(1,(CT3+1))
                  COL1=CT3
                  COL2=(CT3+1)
               ELSEIF (PRESS.EQ.DENDAT(1,CT3)) THEN
                  P1=PRESS
                  P2=DENDAT(1,(CT3+1))
                  COL1=CT3
                  COL2=(CT3+1)
               ENDIF
 10         CONTINUE
            DO 20 CT3=2,29
               DENCOL(CT3)=((PRESS-P2)*((DENDAT(CT3,COL1)
     c         -DENDAT(CT3,COL2))/(P1-P2)))+DENDAT(CT3,COL2)
 20         CONTINUE
            DO 30 CT3=2,29
               IF ((MODDENFINAL(CT2,CT1).LT.DENCOL(CT3)).AND.
     c            (MODDENFINAL(CT2,CT1).GT.DENCOL(CT3+1))) THEN
                  ROW1=CT3
                  ROW2=CT3+1
                  T=(((MODDENFINAL(CT2,CT1)-DENCOL(ROW2))*
     c               (DENDAT(ROW1,1)-DENDAT(ROW2,1)))/(DENCOL(ROW1)
     c               -DENCOL(ROW2)))+DENDAT(ROW2,1)
               ELSEIF ((MODDENFINAL(CT2,CT1)).EQ.DENCOL(CT3)) THEN
                  T=DENDAT(CT3,1)
               ENDIF
 30         CONTINUE
            MODTEMPFINAL(CT2,CT1)=((T-32.0)*(5.0/9.0))+273.15
         ELSEIF (RTYPE.EQ.'BWR') THEN
            MODTEMPFINAL(CT2,CT1)=MODREFTEMP
         ENDIF
 40      CONTINUE
 50   CONTINUE

      RETURN
      END

*******************************************************************
*     SAS2H Input Deck Creation and Execution Control Subroutine  *
*******************************************************************
      SUBROUTINE EXECUTION_CONTROL (NBR, RELATIVE_STPT_NUM,
     c           CT1, CT2, CT3, AXNUM, CYCPOS, AXBLANK,
     c           BPDESID, CRINS, CRDES,
     c           CRMIXNUM, CRMIXID, CRNUMISOS, CRISOID,
     c           APSRINS, APSRMIXNUM, APSRMIXID,
     c           RELATIVE_APSR_MIX_ID, APSRNUMISOS,
     c           APSRISOID, ISN, IIM, ICM, IUS, PLEVEL,
     c           BPZONE, BPMA, CRZONE, CRMA,
     c           LMC, APSRZONE, APSRMA, LMD,
     c           BPTN, BPBN, STPTS, APSRDES,
```

```
     c          STPTDAT, AXBLANKRICH, GRAMS,
     c          NODES, RODS, RICH, FTFINAL,
     c          MODDENFINAL, MODTEMPFINAL,
     c          BLETDOWN, BPWTPCT, BPDEN, CRDEN,
     c          CRISOWTPCT, APSRDEN, APSRISOWTPCT,
     c          PITCH, FOD, COD, CID, SZF, EPS, PTC, MESH,
     c          BPRA, CRRA, LRC, APSRRA,
     c          LRD, POWER, CYCDOWN, PREFIX,
     c          NM, CYCLEID, REACT, LIB, AXBLANKET,
     c          FUELCLAD, BPRFLAG, CRSTAT, APSRSTAT, FLAG2,
     c          LUZONE, LMB, LRB, BPXSECT, BPRODS, CT1START,
     c          CT2START, STPTTALLY, CLADTOT, CLADDESNUM,
     c          CLADDESNAME, BPRCLAD, CRCLAD, APSRCLAD,
     c          CLTEMP, BPMIXNUM, BPMIX, BPMIXID,
     c          BPNUMISOS, BPISOID, BPISOWTPCT, UCSPACERFRAC,
     c          SPACERMAT, STEPCONTROL, VARBLETDOWN, VARSTEPNUM,
     c          VARPOWER, BPRFM, BPFMNUMISOS, BPFISOID,
     c          ABOVEBPNUM, APSRFM, BPRFR, BPFISOWTPCT,
     c          APSRFR, ABOVEBP, APSRFOLLOWMIX, RTYPE, MODREFDEN,
     c          MODREFTEMP, CRMIXDEN,STOPFLAG,CYCSTOP,STPTSTOP,
     C          PICKUPFLAG,FTNDES,GDROD,FIRSTMODEN)
*
      INTEGER*4 CT1, CT2, CT3, NBR, RELATIVE_STPT_NUM,
     c          AXNUM, CYCPOS(10), AXBLANK(50),
     c          BPDESID(10), CRINS(10,20,23,50), CRDES(10,20,23,50),
     c          CRMIXNUM, CRMIXID(25), CRNUMISOS(25), CRISOID(25,20),
     c          APSRINS(10,20,23,50), APSRMIXNUM, APSRMIXID(25),
     c          RELATIVE_APSR_MIX_ID, APSRNUMISOS(25),
     c          APSRISOID(25,10), ISN, IIM, ICM, IUS, PLEVEL,
     c          BPZONE(10), BPMA(15,10), CRZONE(10), CRMA(15,10),
     c          LMC(15,10), APSRZONE(10), APSRMA(15,10), LMD(15,10),
     c          BPTN(10), BPBN(10), STPTS(10), APSRDES(10,20,23,50),
     c          BPRODS(10), SYSTEM, SASEXERESULT,
     c          CARRYCOUNTER, CT1START, CT2START, CT2GOVALUE,
     c          STPTTALLY(20), CT2ENDVALUE, CLADTOT, CLADDESNUM(10),
     c          BPRCLAD(10), CRCLAD(10), APSRCLAD(10), BPMIXNUM,
     c          BPMIX(10), BPMIXID(10), BPNUMISOS(20), BPISOID(10,20),
     c          VARSTEPNUM(10,20), BPRFM(15,10), BPFMNUMISOS(25),
     c          BPFISOID(25,10), ABOVEBPNUM(10), APSRFM(15,10),GDROD,
     c          APSRFOLLOWMIX(10,20,23,50), APSRINSOLD(10,20,23,50),
     C          CYCSTOP,STPTSTOP
*
      REAL      STPTDAT(10,20,3), AXBLANKRICH, GRAMS(50),
     c          NODES(50,2), RODS, RICH, FTFINAL(50,20),
     c          MODDENFINAL(50,20), MODTEMPFINAL(50,20),
     c          BLETDOWN(10,20,25), BPWTPCT(10), BPDEN(10), CRDEN(10),
     c          CRISOWTPCT(25,20), APSRDEN(10), APSRISOWTPCT(25,10),
     c          PITCH, FOD, COD, CID, SZF, EPS, PTC, MESH,
     c          BPRA(15,10), CRRA(15,10), LRC(15,10), APSRRA(15,10),
     c          LRD(15,10), POWER(50,20), CYCDOWN(10), BPXSECT(10),
     c          FINALDOWNTIME, MASSTOTAL, FUELISOWTPCT(1000),
     c          BPRAISOVALUE(10), LEFTVAL(1000), CLTEMP,
     c          BPISOWTPCT(10,20), UCSPACERFRAC,
     c          VARBLETDOWN(10,20,25,25), VARPOWER(10,20,25,50),
     c          BPRFR(15,10), BPFISOWTPCT(25,10), APSRFR(15,10),
     c          MODREFDEN, MODREFTEMP, CRMIXDEN(25),FTNDES(50,2,20),
     C          FIRSTMODEN(20,20)
*
      CHARACTER PREFIX*3, NM*31, CYCLEID(10)*2, REACT*23, LIB*15,
     c          AXBLANKET*1, FUELCLAD*10, BPRFLAG*1, CRSTAT*6,
     c          APSRSTAT*6, FLAG2*7, SASEXECOMMAND*33,
     c          PREVIOUSNAME*25, FUELISONAME(1000)*5,
     c          BPRAISONAME(12)*6, LEFTLIST(1000)*6,STOPFLAG*1,
     c          CLADDESNAME(10)*7, SPACERMAT*7, STEPCONTROL*1,
```

```
     c         ABOVEBP(10)*5, RTYPE*3,PICKUPFLAG*1
*
      LOGICAL  BPRA_INSERTED
*
      RELATIVE_STPT_NUM=0
      DO 30 CT3=1,AXNUM
       IF (CT1START.EQ.1) THEN
          RELATIVE_STPT_NUM=CT2START-1
       ELSE
          RELATIVE_STPT_NUM=STPTTALLY(CT1START-1)+CT2START-1
       ENDIF
       DO 20 CT1=CT1START,NBR
*  CT1 is the insertion cycle incrementer
         IF (CT1.EQ.CT1START) THEN
            CT2GOVALUE=CT2START
         ELSE
            CT2GOVALUE=1
         ENDIF
         IF (CT1.EQ.NBR) THEN
            CT2ENDVALUE=STPTS(CT1)-1
         ELSE
            CT2ENDVALUE=STPTS(CT1)
         ENDIF
*  CT2 is the statepoint incrementer within cycle CT1
         DO 10 CT2=CT2GOVALUE,CT2ENDVALUE
               RELATIVE_STPT_NUM=RELATIVE_STPT_NUM+1
c           this condition stops craft if a RSTART case will be needed
            IF ((STOPFLAG.EQ.'Y').AND.(CT1.EQ.CYCSTOP).AND.
     C      (CT2.EQ.STPTSTOP))THEN
               GOTO 999
            ENDIF
            IF ((CT1.EQ.1).AND.(CT2.EQ.1)) THEN
               CALL STANDARD_WRITER (RELATIVE_STPT_NUM, CT1,
     c         CT2, CT3, AXNUM, CYCPOS, AXBLANK,
     c         BPDESID, CRINS, CRDES,
     c         CRMIXNUM, CRMIXID, CRNUMISOS, CRISOID,
     c         APSRINS, APSRMIXNUM, APSRMIXID,
     c         RELATIVE_APSR_MIX_ID, APSRNUMISOS,
     c         APSRISOID, ISN, IIM, ICM, IUS, PLEVEL,
     c         BPZONE, BPMA, CRZONE, CRMA,
     c         LMC, APSRZONE, APSRMA, LMD,
     c         BPTN, BPBN, STPTS, APSRDES,
     c         STPTDAT, AXBLANKRICH, GRAMS,
     c         NODES, RODS, RICH, FTFINAL,
     c         MODDENFINAL, MODTEMPFINAL,
     c         BLETDOWN, BPWTPCT, BPDEN, CRDEN,
     c         CRISOWTPCT, APSRDEN, APSRISOWTPCT,
     c         PITCH, FOD, COD, CID, SZF, EPS, PTC, MESH,
     c         BPRA, CRRA, LRC, APSRRA,
     c         LRD, POWER, CYCDOWN, PREFIX,
     c         NM, CYCLEID, REACT, LIB, AXBLANKET,
     c         FUELCLAD, BPRFLAG, CRSTAT, APSRSTAT, FLAG2,
     c         LUZONE, LMB, LRB, PREVIOUSNAME, FINALDOWNTIME,
     c         BPRA_INSERTED, CLADTOT, CLADDESNUM,
     c         CLADDESNAME, BPRCLAD, CRCLAD, APSRCLAD,
     c         CLTEMP, BPMIXNUM, BPMIX, BPMIXID,
     c         BPNUMISOS, BPISOID, BPISOWTPCT, UCSPACERFRAC,
     c         SPACERMAT, STEPCONTROL, VARBLETDOWN, VARSTEPNUM,
     c         VARPOWER, BPRFM, BPFMNUMISOS, BPFISOID,
     c         ABOVEBPNUM, APSRFM, BPRFR, BPFISOWTPCT,
     c         APSRFR, ABOVEBP, APSRFOLLOWMIX, CT1START,
     c         CT2GOVALUE, APSRINSOLD, RTYPE, MODREFDEN, CRMIXDEN,
     C         FIRSTMODEN)
               SASEXECOMMAND(1:8)='batch43 '
```

```
            SASEXECOMMAND(9:33)=NM(1:25)
            SASEXERESULT=SYSTEM(SASEXECOMMAND)
            IF (SASEXERESULT.LT.0) THEN
               WRITE (*,*) 'AN ERROR OCCURRED DURING SAS2H',
     c         'EXECUTION OF ', NM(1:25)
            ENDIF
            CALL CUTTER (NM)
         ELSE
            CALL CONTINUATION_WRITER (RELATIVE_STPT_NUM,
     c      CT1, CT2, CT3, AXNUM, CYCPOS, AXBLANK, BPDESID,
     c      CRINS, CRDES, CRMIXNUM, CRMIXID,
     c      CRNUMISOS, CRISOID, APSRINS,
     c      APSRMIXNUM, APSRMIXID, RELATIVE_APSR_MIX_ID,
     c      APSRNUMISOS, APSRISOID, ISN, IIM, ICM, IUS,
     c      PLEVEL, BPZONE, BPMA, CRZONE, CRMA,
     c      LMC, APSRZONE, APSRMA, LMD,
     c      BPTN, BPBN, STPTS, APSRDES,
     c      STPTDAT, AXBLANKRICH, GRAMS,
     c      NODES, RODS, RICH, FTFINAL, MODDENFINAL,
     c      MODTEMPFINAL, BLETDOWN, BPWTPCT,
     c      BPDEN, CRDEN, CRISOWTPCT, APSRDEN,
     c      APSRISOWTPCT, PITCH, FOD, COD, CID, SZF,
     c      EPS, PTC, MESH, BPRA, CRRA, LRC, APSRRA,
     c      LRD, POWER, CYCDOWN, PREFIX, NM,
     c      CYCLEID, REACT, LIB, AXBLANKET, FUELCLAD,
     c      BPRFLAG, CRSTAT, APSRSTAT, FLAG2, LUZONE,
     c      LMB, LRB, MASSTOTAL, FUELISONAME, FUELISOWTPCT,
     c      BPRAISONAME, BPRAISOVALUE, LEFTLIST, CARRYCOUNTER,
     c      BPXSECT, BPRODS, PREVIOUSNAME, FINALDOWNTIME,
     c      LEFTVAL, BPRA_INSERTED, CLADTOT, CLADDESNUM,
     c      CLADDESNAME, BPRCLAD, CRCLAD, APSRCLAD,
     c      CLTEMP, BPMIXNUM, BPMIX, BPMIXID,
     c      BPNUMISOS, BPISOID, BPISOWTPCT, UCSPACERFRAC,
     c      SPACERMAT, STEPCONTROL, VARBLETDOWN, VARSTEPNUM,
     c      VARPOWER, BPRFM, BPFMNUMISOS, BPFISOID,
     c      ABOVEBPNUM, APSRFM, BPRFR, BPFISOWTPCT,
     c      APSRFR, ABOVEBP, APSRFOLLOWMIX, CT1START,
     c      CT2GOVALUE, APSRINSOLD, RTYPE, MODREFDEN, CRMIXDEN,
     C      PICKUPFLAG,FTNDES,GDROD,STOPFLAG,FIRSTMODEN)

            SASEXECOMMAND(1:8)='batch43 '
            SASEXECOMMAND(9:33)=NM(1:25)
            SASEXERESULT=SYSTEM(SASEXECOMMAND)
            IF (SASEXERESULT.LT.0) THEN
               WRITE (*,*) 'AN ERROR OCCURRED DURING SAS2H',
     c         'EXECUTION OF ', NM(1:25)
            ENDIF
            CALL CUTTER (NM)
         ENDIF
   10    CONTINUE
   20  CONTINUE
   30 CONTINUE

  999 RETURN
      END

*****************************************************************
*     Subroutine to write standard beginning of assembly life   *
*     SAS2H input decks                                          *
*****************************************************************
      SUBROUTINE STANDARD_WRITER (RELATIVE_STPT_NUM, CT1, CT2, CT3,
     c AXNUM, CYCPOS, AXBLANK, BPDESID,
     c CRINS, CRDES, CRMIXNUM, CRMIXID,
```

```
     c CRNUMISOS, CRISOID, APSRINS,
     c APSRMIXNUM, APSRMIXID, RELATIVE_APSR_MIX_ID,
     c APSRNUMISOS, APSRISOID, ISN, IIM, ICM, IUS,
     c PLEVEL, BPZONE, BPMA, CRZONE, CRMA,
     c LMC, APSRZONE, APSRMA, LMD,
     c BPTN, BPBN, STPTS, APSRDES,
     c STPTDAT, AXBLANKRICH, GRAMS,
     c NODES, RODS, RICH, FTFINAL, MODDENFINAL,
     c MODTEMPFINAL, BLETDOWN, BPWTPCT,
     c BPDEN, CRDEN, CRISOWTPCT, APSRDEN,
     c APSRISOWTPCT, PITCH, FOD, COD, CID, SZF, EPS, PTC,
     c MESH, BPRA, CRRA, LRC, APSRRA,
     c LRD, POWER, CYCDOWN, PREFIX, NM,
     c CYCLEID, REACT, LIB, AXBLANKET, FUELCLAD,
     c BPRFLAG, CRSTAT, APSRSTAT, FLAG2, LUZONE, LMB, LRB,
     c PREVIOUSNAME, FINALDOWNTIME, BPRA_INSERTED, CLADTOT,
     c CLADDESNUM, CLADDESNAME, BPRCLAD, CRCLAD, APSRCLAD,
     c CLTEMP, BPMIXNUM, BPMIX, BPMIXID,
     c BPNUMISOS, BPISOID, BPISOWTPCT, UCSPACERFRAC,
     c SPACERMAT, STEPCONTROL, VARBLETDOWN, VARSTEPNUM,
     c VARPOWER, BPRFM, BPFMNUMISOS, BPFISOID,
     c ABOVEBPNUM, APSRFM, BPRFR, BPFISOWTPCT,
     c APSRFR, ABOVEBP, APSRFOLLOWMIX, CT1START, CT2GOVALUE,
     c APSRINSOLD, RTYPE, MODREFDEN, CRMIXDEN,FIRSTMODEN)
*
      INTEGER*4 RELATIVE_STPT_NUM, CT1, CT2, CT3, AXNUM,
     c NUMSTPT1, NUMSTPT2, NUMSTPT3, CYCPOS(10), AXBLANK(50),
     c BPDESID(10), BPRA_DESCRIPTION_ID, CT4, CT5, CRINS(10,20,23,50),
     c CR_MIXTURE_ID, CR_DESCRIPTION, CRDES(10,20,23,50), CRMIXNUM,
     c CRMIXID(25), RELATIVE_CR_MIX_ID, CRNUMISOS(25),
     c CRISOID(25,20), APSRINS(10,20,23,50), APSR_MIXTURE_ID,
     c APSR_DESCRIPTION, APSRMIXNUM, APSRMIXID(25),
     c RELATIVE_APSR_MIX_ID, APSRNUMISOS(25), APSRISOID(25,10),
     c ISN, IIM, ICM, IUS, PLEVEL, BPZONE(10), BPMA(15,10),
     c CRZONE(10), CRMA(15,10), LMC(15,10), APSRZONE(10),
     c APSRMA(15,10), LMD(15,10), BPTN(10), BPBN(10), STPTS(10),
     c APSRDES(10,20,23,50), LUZONE, LMB(15), NUMSTPT4, NUMSTPT5,
     c NUMSTPT6, CLADTOT, CLADDESNUM(10), BPRCLAD(10), CRCLAD(10),
     c APSRCLAD(10), APSRCLNUM, CRCLNUM, BPRCLNUM, BPMIXNUM,
     c BPMIX(10), BPMIXID(10), BPNUMISOS(20), BPISOID(10,20),
     c VARSTEPNUM(10,20), BPRFM(15,10), BPFMNUMISOS(25),
     c BPFISOID(25,10), ABOVEBPNUM(10), APSRFM(15,10),
     c APSRFOLLOWMIX(10,20,23,50), FOLNODKEEP,
     c FOLSTEPKEEP, APSRFOLNUM, APSRFOLLOWDATA(10,20,23,50),
     c CT1START, CT2GOVALUE, APSRINSOLD(10,20,23,50)
*
      REAL STPTDAT(10,20,3), ENR, AXBLANKRICH, OXYGMS, GRAMS(50),
     c UO2GMS, FVOL, PI, NODES(50,2), RODS, FDEN, WT234,
     c WT235, WT236, WT238, RICH, FTFINAL(50,20),
     c MODDENFINAL(50,20), MODTEMPFINAL(50,20), BLETDOWN(10,20,25),
     c BPWTPCT(10), BPDEN(10), ALFRAC, OFRAC, CRDEN(10),
     c CRISOWTPCT(25,20), APSRDEN(10), APSRISOWTPCT(25,10),
     c PITCH, FOD, COD, CID, SZF, EPS, PTC, MESH, BPRA(15,10),
     c CRRA(15,10), LRC(15,10), APSRRA(15,10), LRD(15,10),
     c DOWNTIME, BORON_FRACTION, POWER(50,20), CYCDOWN(10), LRB(15),
     c FINALDOWNTIME, CLTEMP, BPISOWTPCT(10,20), UCSPACERFRAC,
     c BORATEDMODVF, BORONVF, UCMODREGIONDEN,
     c VARBLETDOWN(10,20,25,25), VARPOWER(10,20,25,50),
     c BPRFR(15,10), BPFISOWTPCT(25,10), APSRFR(15,10), MODREFDEN,
     c CRMIXDEN(25),FIRSTMODEN(20,20)
*
      CHARACTER CHNODE*2, CHID*2, PREFIX*3, CHSTPT1*1, CHSTPT2*1,
     c CHSTPT3*1, NM*31, CYCLEID(10)*2, REACT*23, LIB*15,
     c AXBLANKET*1, FUELCLAD*10, BPRFLAG*1, CRSTAT*6, APSRSTAT*6,
```

```
     c FLAG2*7, IRRAD_STEPS*2, PLEVELCH*2, BPZONECH*2, CRZONECH*2,
     c APSRZONECH*2, LUZONECH*2, PREVIOUSNAME*25, ASSYPOSITION*2,
     c CHSTPT4*1, CHSTPT5*1, CHSTPT6*1, CLADDESNAME(10)*7,
     c SPACERMAT*7, STEPCONTROL*1, ABOVEBP(10)*5, RTYPE*3, CTIME*8,
     c CDATE*9, VERS*9
*
      LOGICAL BPRA_INSERTED, CR_INSERTED, CRCOMPFLAG, APSR_INSERTED,
     c APSRCOMPFLAG, BPRA_FOLLOW, APSRBOTFLAG, FOLLOWIN,
     c UO2FLAG(25)
*
      PI=3.14159265359
*  Determination of the input deck filename
         CALL ZEROS(CT3,CHNODE)
         CALL ZEROS(CYCPOS(CT1),CHID)
         NUMSTPT1=INT(STPTDAT(CT1,CT2,1)/100.0)
         CHSTPT1=CHAR(NUMSTPT1+48)
         NUMSTPT2=INT((STPTDAT(CT1,CT2,1)-(NUMSTPT1*100))/10.0)
         CHSTPT2=CHAR(NUMSTPT2+48)
         NUMSTPT3=INT((STPTDAT(CT1,CT2,1)-(NUMSTPT1*100)-
     c    (NUMSTPT2*10)))
         CHSTPT3=CHAR(NUMSTPT3+48)
         IF (CT2.LT.STPTS(CT1)) THEN
            NUMSTPT4=INT(STPTDAT(CT1,(CT2+1),1)/100.0)
            CHSTPT4=CHAR(NUMSTPT4+48)
            NUMSTPT5=INT((STPTDAT(CT1,(CT2+1),1)-(NUMSTPT4*100))/10.0)
            CHSTPT5=CHAR(NUMSTPT5+48)
            NUMSTPT6=INT((STPTDAT(CT1,(CT2+1),1)-(NUMSTPT4*100)-
     c       (NUMSTPT5*10)))
            CHSTPT6=CHAR(NUMSTPT6+48)
         ELSEIF (CT2.EQ.STPTS(CT1)) THEN
            NUMSTPT4=INT(STPTDAT((CT1+1),1,1)/100.0)
            CHSTPT4=CHAR(NUMSTPT4+48)
            NUMSTPT5=INT((STPTDAT((CT1+1),1,1)-(NUMSTPT4*100))/10.0)
            CHSTPT5=CHAR(NUMSTPT5+48)
            NUMSTPT6=INT((STPTDAT((CT1+1),1,1)-(NUMSTPT4*100)-
     c       (NUMSTPT5*10)))
            CHSTPT6=CHAR(NUMSTPT6+48)
         ENDIF
         NM(1:3)=PREFIX
         NM(4:4)='A'
         NM(5:6)=CHID
         NM(7:7)='N'
         NM(8:9)=CHNODE
         NM(10:11)='DC'
         NM(12:13)=CYCLEID(CT1)
         NM(14:14)='T'
         NM(15:15)=CHSTPT1
         NM(16:16)=CHSTPT2
         NM(17:17)=CHSTPT3
         NM(18:19)='AC'
         IF (CT2.EQ.STPTS(CT1)) THEN
            NM(20:21)=CYCLEID(CT1+1)
         ELSE
            NM(20:21)=CYCLEID(CT1)
         ENDIF
         NM(22:22)='T'
         NM(23:23)=CHSTPT4
         NM(24:24)=CHSTPT5
         NM(25:25)=CHSTPT6
         NM(26:31)='.input'
         PREVIOUSNAME=NM(1:25)
*  Open and rewind the input deck file
         OPEN(UNIT=100, FILE=NM, STATUS='UNKNOWN')
         REWIND(UNIT=100)
```

```
*   Write first section of input deck
          WRITE (100,10)
   10     FORMAT ('=sas2h',T11,'parm=skipshipdata')
          IF (CT2.LT.STPTS(CT1)) THEN
             WRITE (100,20) REACT, CHID, CHNODE,
     c       NM(12:13), STPTDAT(CT1,CT2,1), NM(20:21),
     c       STPTDAT(CT1,(CT2+1),1)
   20        FORMAT (A23,1X,'Assy-',A2,
     c       ', Node-',A2,1X,
     c       '{Cyc-',A2,', 'F5.1,' to Cyc-',
     c       A2,', ',F5.1,' EFPD}')
          ELSEIF (CT2.EQ.STPTS(CT1)) THEN
             WRITE (100,25) REACT, CHID, CHNODE,
     c       NM(12:13), STPTDAT(CT1,CT2,1), NM(20:21),
     c       STPTDAT((CT1+1),1,1)
   25        FORMAT (A23,1X,'Assy-',A2,
     c       ', Node-',A2,1X,
     c       '{Cyc-',A2,', 'F5.1,' to Cyc-',
     c       A2,', ',F5.1,' EFPD}')
          ENDIF
          WRITE (100,30) LIB
   30     FORMAT (A15,1X,'latticecell')
          WRITE (100,40)
   40     FORMAT ('''')
          CALL DATE(CDATE)
          CALL TIME(CTIME)
          VERS='CRAFT V4C'
          WRITE (100,48)
   48     FORMAT ('''',T5,'******************************')
          WRITE (100,49)
   49     FORMAT ('''',T5,'this file created by:')
          WRITE (100,50)VERS,CDATE,CTIME
   50     FORMAT ('''',T5,A9,2X,A9,2X,A8)
          WRITE (100,51)
   51     FORMAT ('''',T5,'******************************')
          WRITE (100,52)
   52     FORMAT ('''')
          WRITE (100,60)
   60     FORMAT ('''    fuel density based on mass of uranium per',
     c    ' node')
          WRITE (100,70)
   70     FORMAT ('''')
*   Write second section of input deck (material specifications)
          WRITE (100,80)
   80     FORMAT ('''',5X,'material specification input')
          WRITE (100,90)
   90     FORMAT ('''')
*   Calculate initital fuel parameters depending upon whether or not the
*   node represents axial blanket fuel
          IF ((AXBLANKET.EQ.'Y').AND.(AXBLANK(CT3).EQ.1)) THEN
             ENR=AXBLANKRICH
             OXYGMS=(GRAMS(CT3)*2*15.994915)/(((ENR/100)*235.043915)+
     c       (((0.007731*((ENR)**1.0837))/100)*234.040904)+
     c       (((0.0046*ENR)/100)*236.045637)+(((100-(0.007731*
     c       (ENR**1.0837))-(ENR)-(0.0046*ENR))/100)*238.05077))
             UO2GMS=GRAMS(CT3)+OXYGMS
             FVOL=(PI/4)*(FOD**2)*(NODES(CT3,2))*(RODS)
             FDEN=UO2GMS/FVOL
             WT234=0.007731*(ENR**1.0837)
             WT235=ENR
             WT236=0.0046*ENR
             WT238=100.0-WT234-ENR-WT236
          ELSE
             ENR=RICH
```

```
          OXYGMS=(GRAMS(CT3)*2*15.994915)/(((ENR/100)*235.043915)+
    c     (((0.007731*((ENR)**1.0837))/100)*234.040904)+
    c     (((0.0046*ENR)/100)*236.045637)+(((100-(0.007731*
    c     (ENR**1.0837))-(ENR)-(0.0046*ENR))/100)*238.05077))
          UO2GMS=GRAMS(CT3)+OXYGMS
c modification here to calculate proper density using CLADID rather than
c FOD because gap is removed from pathA model
c          FVOL=(PI/4)*(FOD**2)*(NODES(CT3,2))*(RODS)
          FVOL=(PI/4)*(CID**2)*(NODES(CT3,2))*(RODS)
          FDEN=UO2GMS/FVOL
          WT234=0.007731*(ENR**1.0837)
          WT235=ENR
          WT236=0.0046*ENR
          WT238=100.0-WT234-ENR-WT236
       ENDIF
*  Write fuel composition input description
       IF (FDEN.LT.(10.0)) THEN
          WRITE (100,100) FDEN, FTFINAL(CT3,RELATIVE_STPT_NUM), WT234,
     c    WT235, WT236, WT238
  100     FORMAT ('uo2 1 den=',F6.3,1X,'1',1X,F6.1,1X,'92234',
     c    1X,F5.4,1X,'92235',1X,F6.4,1X,'92236',1X,F5.4,1X,
     c    '92238',1X,F6.3,1X,'end')
       ELSE
          WRITE (100,110) FDEN, FTFINAL(CT3,RELATIVE_STPT_NUM), WT234,
     c    WT235, WT236, WT238
  110     FORMAT ('uo2 1 den=',F6.3,1X,'1',1X,F6.1,1X,'92234',
     c    1X,F5.4,1X,'92235',1X,F6.4,1X,'92236',1X,F5.4,1X,
     c    '92238',1X,F6.3,1X,'end')
       ENDIF
       WRITE (100,120) FTFINAL(CT3,RELATIVE_STPT_NUM)
  120  FORMAT ('kr-83          1   0   1-21   ',F6.1,'   end')
       WRITE (100,130) FTFINAL(CT3,RELATIVE_STPT_NUM)
  130  FORMAT ('kr-85          1   0   1-21   ',F6.1,'   end')
       WRITE (100,140) FTFINAL(CT3,RELATIVE_STPT_NUM)
  140  FORMAT ('sr-90          1   0   1-21   ',F6.1,'   end')
       WRITE (100,150) FTFINAL(CT3,RELATIVE_STPT_NUM)
  150  FORMAT ('y-89           1   0   1-21   ',F6.1,'   end')
       WRITE (100,160) FTFINAL(CT3,RELATIVE_STPT_NUM)
  160  FORMAT ('mo-95          1   0   1-21   ',F6.1,'   end')
       WRITE (100,170) FTFINAL(CT3,RELATIVE_STPT_NUM)
  170  FORMAT ('zr-93          1   0   1-21   ',F6.1,'   end')
       WRITE (100,180) FTFINAL(CT3,RELATIVE_STPT_NUM)
  180  FORMAT ('zr-94          1   0   1-21   ',F6.1,'   end')
       WRITE (100,190) FTFINAL(CT3,RELATIVE_STPT_NUM)
  190  FORMAT ('zr-95          1   0   1-21   ',F6.1,'   end')
       WRITE (100,200) FTFINAL(CT3,RELATIVE_STPT_NUM)
  200  FORMAT ('nb-94          1   0   1-21   ',F6.1,'   end')
       WRITE (100,210) FTFINAL(CT3,RELATIVE_STPT_NUM)
  210  FORMAT ('tc-99          1   0   1-21   ',F6.1,'   end')
       WRITE (100,220) FTFINAL(CT3,RELATIVE_STPT_NUM)
  220  FORMAT ('rh-103         1   0   1-21   ',F6.1,'   end')
       WRITE (100,230) FTFINAL(CT3,RELATIVE_STPT_NUM)
  230  FORMAT ('rh-105         1   0   1-21   ',F6.1,'   end')
       WRITE (100,240) FTFINAL(CT3,RELATIVE_STPT_NUM)
  240  FORMAT ('ru-101         1   0   1-21   ',F6.1,'   end')
       WRITE (100,250) FTFINAL(CT3,RELATIVE_STPT_NUM)
  250  FORMAT ('ru-106         1   0   1-21   ',F6.1,'   end')
       WRITE (100,260) FTFINAL(CT3,RELATIVE_STPT_NUM)
  260  FORMAT ('pd-105         1   0   1-21   ',F6.1,'   end')
       WRITE (100,270) FTFINAL(CT3,RELATIVE_STPT_NUM)
  270  FORMAT ('pd-108         1   0   1-21   ',F6.1,'   end')
       WRITE (100,280) FTFINAL(CT3,RELATIVE_STPT_NUM)
  280  FORMAT ('ag-109         1   0   1-21   ',F6.1,'   end')
       WRITE (100,290) FTFINAL(CT3,RELATIVE_STPT_NUM)
```

```
  290     FORMAT ('sb-124          1   0   1-21    ',F6.1,'   end')
          WRITE (100,300) FTFINAL(CT3,RELATIVE_STPT_NUM)
  300     FORMAT ('xe-131          1   0   1-21    ',F6.1,'   end')
          WRITE (100,310) FTFINAL(CT3,RELATIVE_STPT_NUM)
  310     FORMAT ('xe-132          1   0   1-21    ',F6.1,'   end')
          WRITE (100,320) FTFINAL(CT3,RELATIVE_STPT_NUM)
  320     FORMAT ('xe-135          1   0   1-21    ',F6.1,'   end')
          WRITE (100,330) FTFINAL(CT3,RELATIVE_STPT_NUM)
  330     FORMAT ('xe-136          1   0   1-21    ',F6.1,'   end')
          WRITE (100,340) FTFINAL(CT3,RELATIVE_STPT_NUM)
  340     FORMAT ('cs-134          1   0   1-21    ',F6.1,'   end')
          WRITE (100,350) FTFINAL(CT3,RELATIVE_STPT_NUM)
  350     FORMAT ('cs-135          1   0   1-21    ',F6.1,'   end')
          WRITE (100,360) FTFINAL(CT3,RELATIVE_STPT_NUM)
  360     FORMAT ('cs-137          1   0   1-21    ',F6.1,'   end')
          WRITE (100,370) FTFINAL(CT3,RELATIVE_STPT_NUM)
  370     FORMAT ('ba-136          1   0   1-21    ',F6.1,'   end')
          WRITE (100,380) FTFINAL(CT3,RELATIVE_STPT_NUM)
  380     FORMAT ('la-139          1   0   1-21    ',F6.1,'   end')
          WRITE (100,390) FTFINAL(CT3,RELATIVE_STPT_NUM)
  390     FORMAT ('ce-144          1   0   1-21    ',F6.1,'   end')
          WRITE (100,400) FTFINAL(CT3,RELATIVE_STPT_NUM)
  400     FORMAT ('nd-143          1   0   1-21    ',F6.1,'   end')
          WRITE (100,410) FTFINAL(CT3,RELATIVE_STPT_NUM)
  410     FORMAT ('nd-145          1   0   1-21    ',F6.1,'   end')
          WRITE (100,420) FTFINAL(CT3,RELATIVE_STPT_NUM)
  420     FORMAT ('pm-147          1   0   1-21    ',F6.1,'   end')
          WRITE (100,430) FTFINAL(CT3,RELATIVE_STPT_NUM)
  430     FORMAT ('pm-148          1   0   1-21    ',F6.1,'   end')
          WRITE (100,440) FTFINAL(CT3,RELATIVE_STPT_NUM)
  440     FORMAT ('nd-147          1   0   1-21    ',F6.1,'   end')
          WRITE (100,450) FTFINAL(CT3,RELATIVE_STPT_NUM)
  450     FORMAT ('sm-147          1   0   1-21    ',F6.1,'   end')
          WRITE (100,460) FTFINAL(CT3,RELATIVE_STPT_NUM)
  460     FORMAT ('sm-149          1   0   1-21    ',F6.1,'   end')
          WRITE (100,470) FTFINAL(CT3,RELATIVE_STPT_NUM)
  470     FORMAT ('sm-150          1   0   1-21    ',F6.1,'   end')
          WRITE (100,480) FTFINAL(CT3,RELATIVE_STPT_NUM)
  480     FORMAT ('sm-151          1   0   1-21    ',F6.1,'   end')
          WRITE (100,490) FTFINAL(CT3,RELATIVE_STPT_NUM)
  490     FORMAT ('sm-152          1   0   1-21    ',F6.1,'   end')
          WRITE (100,500) FTFINAL(CT3,RELATIVE_STPT_NUM)
  500     FORMAT ('gd-155          1   0   1-21    ',F6.1,'   end')
          WRITE (100,510) FTFINAL(CT3,RELATIVE_STPT_NUM)
  510     FORMAT ('eu-153          1   0   1-21    ',F6.1,'   end')
          WRITE (100,520) FTFINAL(CT3,RELATIVE_STPT_NUM)
  520     FORMAT ('eu-154          1   0   1-21    ',F6.1,'   end')
          WRITE (100,530) FTFINAL(CT3,RELATIVE_STPT_NUM)
  530     FORMAT ('eu-155          1   0   1-21    ',F6.1,'   end')
*  Write cladding material specifications
*  Additional cladding material specifications may be added to the
*  following IF statement as required
          IF ((FUELCLAD.EQ.'ZIRC-4     ').OR.
     c    (FUELCLAD.EQ.'ZIRCALLOY4')) THEN
            WRITE (100,532)
  532       FORMAT ('arbm-zirc4 6.56 5 0 0 0 8016 0.12 24000',
     c      ' 0.10 26000 0.20 50000 1.40')
            WRITE (100,535) CLTEMP
  535       FORMAT (T12,'40000 98.18 2 1.0 ',F5.1,' end')
          ELSEIF (FUELCLAD.EQ.'SS304      ') THEN
            WRITE (100,537)
  537       FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055',
     c      ' 2.0 26304 69.5 28304 9.5')
            WRITE (100,540) CLTEMP
```

```
  540         FORMAT (T12,'2 1.0 ',F5.1,' end')
           ELSEIF (FUELCLAD.EQ.'SS304S    ') THEN
              WRITE (100,542)
  542         FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055',
     c        ' 2.0 26000 69.5 28000 9.5')
              WRITE (100,545) CLTEMP
  545         FORMAT (T13,'2 1.0 ',F5.1,' end')
           ELSEIF (FUELCLAD.EQ.'SS316     ') THEN
              WRITE (100,547)
  547         FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000',
     c        ' 1.0 24304 17.0 25055 2.0')
              WRITE (100,550)
  550         FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
              WRITE (100,552) CLTEMP
  552         FORMAT (T12,'2 1.0 ',F5.1,' end')
           ELSEIF (FUELCLAD.EQ.'SS316S    ') THEN
              WRITE (100,555)
  555         FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000',
     c        ' 1.0 24000 17.0 25055 2.0')
              WRITE (100,557)
  557         FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')
              WRITE (100,559) CLTEMP
  559         FORMAT (T13,'2 1.0 ',F5.1,' end')
           ENDIF
*  Write moderator material specifications
           BORATEDMODVF=1.0-UCSPACERFRAC
           IF (RTYPE.EQ.'PWR') THEN
              IF (STEPCONTROL.EQ.'N') THEN
                 BORONVF=BLETDOWN(CT1,CT2,3)*(1E-6)*BORATEDMODVF
              ELSEIF (STEPCONTROL.EQ.'Y') THEN
                 BORONVF=VARBLETDOWN(CT1,CT2,1,2)*(1E-6)*BORATEDMODVF
              ENDIF
           ENDIF
           WRITE (100,560)
  560      FORMAT ('''')
           IF ((SPACERMAT.EQ.'ZIRC-4 ').AND.
     c     (UCSPACERFRAC.GT.(0.0))) THEN
              WRITE (100,561)
  561         FORMAT ('''      material composition of moderator',
     c        ' within unit cell')
              WRITE (100,562)
  562         FORMAT ('''      with smeared zirc-4 spacer grids')
              IF (RTYPE.EQ.'PWR') THEN
                 UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
     c           BORATEDMODVF)+(6.56*UCSPACERFRAC)
              ELSEIF (RTYPE.EQ.'BWR') THEN
C                  UCMODREGIONDEN=(MODREFDEN*
C     c            BORATEDMODVF)+(6.56*UCSPACERFRAC)
c                    write(99,*)'*****CT1,CT2 ',CT1,CT2
c                    write(99,*)'**FIRSTMODEN(CT1,CT2) ',FIRSTMODEN(CT1,CT2)
                 UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
     c           BORATEDMODVF)+(6.56*UCSPACERFRAC)
              ENDIF
              IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
                 WRITE (100,563) UCMODREGIONDEN, BORATEDMODVF,
     c           MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  563            FORMAT ('h2o    3    den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
              ELSE
                 WRITE (100,564) UCMODREGIONDEN, BORATEDMODVF,
     c           MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  564            FORMAT ('h2o    3    den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
              ENDIF
              IF (RTYPE.EQ.'PWR') THEN
                 WRITE (100,565) UCMODREGIONDEN, BORONVF,
```

```
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  565          FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
     c         1X,F6.5,1X,F7.1,1X,'end')
            ENDIF
            WRITE (100,566) UCMODREGIONDEN
  566       FORMAT ('arbm-spacer',3X,F6.4,1X,'5 0 0 0 8016 0.12',
     c      ' 24000 0.10 26000 0.25')
            WRITE (100,567) UCSPACERFRAC,
     c      MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  567       FORMAT (T17'50000 1.40 40000 98.18 3',1X,F6.5,1X,
     c      F7.1,1X,'end')
         ELSEIF ((SPACERMAT.EQ.'INCONEL').AND.
     c   (UCSPACERFRAC.GT.(0.0))) THEN
            WRITE (100,568)
  568       FORMAT ('''     material composition of moderator',
     c      ' within unit cell')
            WRITE (100,569)
c 569        FORMAT ('''     with smeared inconel spacer grids')
  569       FORMAT ('''     without smeared inconel spacer grids')
            IF (RTYPE.EQ.'PWR') THEN
               UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
     c         BORATEDMODVF)+(8.3*UCSPACERFRAC)
            ELSEIF (RTYPE.EQ.'BWR') THEN
C               UCMODREGIONDEN=(MODREFDEN*
C     c         BORATEDMODVF)+(8.3*UCSPACERFRAC)
c                write(99,*)'*****CT1,CT2 ',CT1,CT2
c                write(99,*)'**FIRSTMODEN(CT1,CT2) ',FIRSTMODEN(CT1,CT2)
               UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
     c         BORATEDMODVF)+(8.3*UCSPACERFRAC)
            ENDIF
            IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
               WRITE (100,570) UCMODREGIONDEN, BORATEDMODVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  570          FORMAT ('h2o    3    den=',F5.4,3X,G14.6,3X,F7.1,3X,'end')
            ELSE
               WRITE (100,571) UCMODREGIONDEN, BORATEDMODVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  571          FORMAT ('h2o    3    den=',F6.4,3X,G14.6,3X,F7.1,3X,'end')
            ENDIF
            IF (RTYPE.EQ.'PWR') THEN
               WRITE (100,572) UCMODREGIONDEN, BORONVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  572          FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
     c         1X,F6.5,1X,F7.1,1X,'end')
            ENDIF
            WRITE (100,573) UCMODREGIONDEN
  573       FORMAT ('arbm-spacer',3X,F6.4,1X,'5 0 0 0 14000 2.5',
     c      ' 22000 2.5 24000 15.0')
            WRITE (100,574) UCSPACERFRAC,
     c      MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  574       FORMAT (T17'26000 7.0 28000 73.0 3',1X,G14.6,1X,
     c      F7.1,1X,'end')
         ELSEIF ((SPACERMAT.EQ.'SS316  ').AND.
     c   (UCSPACERFRAC.GT.(0.0))) THEN
            WRITE (100,575)
  575       FORMAT ('''     material composition of moderator',
     c      ' within unit cell')
            WRITE (100,576)
  576       FORMAT ('''     with smeared ss316 spacer grids')
            IF (RTYPE.EQ.'PWR') THEN
               UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
     c         BORATEDMODVF)+(7.75*UCSPACERFRAC)
            ELSEIF (RTYPE.EQ.'BWR') THEN
C               UCMODREGIONDEN=(MODREFDEN*
```

```
C     c          BORATEDMODVF)+(7.75*UCSPACERFRAC)
                UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
    c           BORATEDMODVF)+(7.75*UCSPACERFRAC)
             ENDIF
             IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
                WRITE (100,577) UCMODREGIONDEN, BORATEDMODVF,
    c           MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  577           FORMAT ('h2o    3    den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
             ELSE
                WRITE (100,578) UCMODREGIONDEN, BORATEDMODVF,
    c           MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  578           FORMAT ('h2o    3    den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
             ENDIF
             IF (RTYPE.EQ.'PWR') THEN
                WRITE (100,579) UCMODREGIONDEN, BORONVF,
    c           MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  579           FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
    c           1X,F6.5,1X,F7.1,1X,'end')
             ENDIF
             WRITE (100,580) UCMODREGIONDEN
  580        FORMAT ('arbm-spacer',3X,F6.4,1X,'7 0 0 0 6012 0.08',
    c        ' 14000 1.0 24304 17.0 25055 2.0')
             WRITE (100,581) UCSPACERFRAC,
    c        MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  581        FORMAT (T5'26304 65.42 28304 12.0 42000 2.5 3',1X,F6.5,1X,
    c        F7.1,1X,'end')
          ELSEIF ((SPACERMAT.EQ.'SS316S ').AND.
    c     (UCSPACERFRAC.GT.(0.0))) THEN
             WRITE (100,582)
  582        FORMAT ('''     material composition of moderator',
    c        ' within unit cell')
             WRITE (100,583)
  583        FORMAT ('''     with smeared ss316s spacer grids')
             IF (RTYPE.EQ.'PWR') THEN
                UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
    c           BORATEDMODVF)+(7.75*UCSPACERFRAC)
             ELSEIF (RTYPE.EQ.'BWR') THEN
C                UCMODREGIONDEN=(MODREFDEN*
C     c          BORATEDMODVF)+(7.75*UCSPACERFRAC)
                UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
    c           BORATEDMODVF)+(7.75*UCSPACERFRAC)
             ENDIF
             IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
                WRITE (100,584) UCMODREGIONDEN, BORATEDMODVF,
    c           MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  584           FORMAT ('h2o    3    den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
             ELSE
                WRITE (100,585) UCMODREGIONDEN, BORATEDMODVF,
    c           MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  585           FORMAT ('h2o    3    den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
             ENDIF
             IF (RTYPE.EQ.'PWR') THEN
                WRITE (100,586) UCMODREGIONDEN, BORONVF,
    c           MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  586           FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
    c           1X,F6.5,1X,F7.1,1X,'end')
             ENDIF
             WRITE (100,587) UCMODREGIONDEN
  587        FORMAT ('arbm-spacer',3X,F6.4,1X,'7 0 0 0 6012 0.08',
    c        ' 14000 1.0 24000 17.0 25055 2.0')
             WRITE (100,588) UCSPACERFRAC,
    c        MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  588        FORMAT (T5'26000 65.42 28000 12.0 42000 2.5 3',1X,F6.5,1X,
    c        F7.1,1X,'end')
```

```
        ELSEIF ((SPACERMAT.EQ.'SS304  ').AND.
     c  (UCSPACERFRAC.GT.(0.0))) THEN
           WRITE (100,589)
  589      FORMAT ('''      material composition of moderator',
     c     ' within unit cell')
           WRITE (100,590)
  590      FORMAT ('''      with smeared ss304 spacer grids')
           IF (RTYPE.EQ.'PWR') THEN
              UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
     c        BORATEDMODVF)+(7.92*UCSPACERFRAC)
           ELSEIF (RTYPE.EQ.'BWR') THEN
C               UCMODREGIONDEN=(MODREFDEN*
C     c         BORATEDMODVF)+(7.92*UCSPACERFRAC)
              UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
     c        BORATEDMODVF)+(7.92*UCSPACERFRAC)
           ENDIF
           IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
              WRITE (100,591) UCMODREGIONDEN, BORATEDMODVF,
     c        MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  591         FORMAT ('h2o    3    den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
           ELSE
              WRITE (100,592) UCMODREGIONDEN, BORATEDMODVF,
     c        MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  592         FORMAT ('h2o    3    den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
           ENDIF
           IF (RTYPE.EQ.'PWR') THEN
              WRITE (100,593) UCMODREGIONDEN, BORONVF,
     c        MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  593         FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
     c        1X,F6.5,1X,F7.1,1X,'end')
           ENDIF
           WRITE (100,594) UCMODREGIONDEN
  594      FORMAT ('arbm-spacer',3X,F6.4,1X,'4 0 0 0 24304 19.0',
     c     ' 25055 2.0 26304 69.5 28304 9.5')
           WRITE (100,595) UCSPACERFRAC,
     c     MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  595      FORMAT (T15'3',1X,F6.5,1X,F7.1,1X,'end')
        ELSEIF ((SPACERMAT.EQ.'SS304S ').AND.
     c  (UCSPACERFRAC.GT.(0.0))) THEN
           WRITE (100,596)
  596      FORMAT ('''      material composition of moderator',
     c     ' within unit cell')
           WRITE (100,597)
  597      FORMAT ('''      with smeared ss304s spacer grids')
           IF (RTYPE.EQ.'PWR') THEN
              UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
     c        BORATEDMODVF)+(7.92*UCSPACERFRAC)
           ELSEIF (RTYPE.EQ.'BWR') THEN
C               UCMODREGIONDEN=(MODREFDEN*
C     c         BORATEDMODVF)+(7.92*UCSPACERFRAC)
              UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
     c        BORATEDMODVF)+(7.92*UCSPACERFRAC)
           ENDIF
           IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
              WRITE (100,598) UCMODREGIONDEN, BORATEDMODVF,
     c        MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  598         FORMAT ('h2o    3    den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
           ELSE
              WRITE (100,599) UCMODREGIONDEN, BORATEDMODVF,
     c        MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  599         FORMAT ('h2o    3    den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
           ENDIF
           IF (RTYPE.EQ.'PWR') THEN
              WRITE (100,600) UCMODREGIONDEN, BORONVF,
```

```
     c          MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  600           FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
     c          1X,F6.5,1X,F7.1,1X,'end')
             ENDIF
             WRITE (100,601) UCMODREGIONDEN
  601        FORMAT ('arbm-spacer',3X,F6.4,1X,'4 0 0 0 24000 19.0',
     c       ' 25055 2.0 26000 69.5 28000 9.5')
             WRITE (100,602) UCSPACERFRAC,
     c       MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  602        FORMAT (T15'3',1X,F6.5,1X,F7.1,1X,'end')
          ELSEIF (UCSPACERFRAC.EQ.(0.0)) THEN
             WRITE (100,603)
  603        FORMAT ('''     material composition of moderator',
     c       ' within unit cell')
             WRITE (100,604)
  604        FORMAT ('''     with no smeared spacer grids')
             IF (RTYPE.EQ.'PWR') THEN
                UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
     c          BORATEDMODVF)
             ELSEIF (RTYPE.EQ.'BWR') THEN
C                UCMODREGIONDEN=(MODREFDEN*
C     c          BORATEDMODVF)
                UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
     c          BORATEDMODVF)
             ENDIF
             IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
                WRITE (100,605) UCMODREGIONDEN, BORATEDMODVF,
     c          MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  605           FORMAT ('h2o    3    den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
             ELSE
                WRITE (100,606) UCMODREGIONDEN, BORATEDMODVF,
     c          MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  606           FORMAT ('h2o    3    den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
             ENDIF
             IF (RTYPE.EQ.'PWR') THEN
                WRITE (100,607) UCMODREGIONDEN, BORONVF,
     c          MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  607           FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
     c          1X,F6.5,1X,F7.1,1X,'end')
             ENDIF
          ENDIF
          WRITE (100,608)
  608     FORMAT ('''')
*  Write BPRA material specifications
*  BPR follow specifications
          BPRA_FOLLOW=.FALSE.
          IF ((BPRFLAG.EQ.'Y').AND.(BPDESID(CT1).NE.0).AND.
     c       (CT3.LT.BPTN(CT1))) THEN
             BPRA_FOLLOW=.TRUE.
             BPRA_DESCRIPTION_ID=BPDESID(CT1)
             WRITE(100,610)
  610        FORMAT('''')
             WRITE(100,612)
  612        FORMAT('''',5X,'BPR above the BP absorber region')
             WRITE(100,614)
  614        FORMAT('''')
             IF ((BPRCLAD(BPDESID(CT1)).NE.0).AND.
     c       (BPRCLAD(BPDESID(CT1)).NE.2)) THEN
               DO 616 CT5=1,10
                IF (BPRCLAD(BPDESID(CT1)).EQ.CLADDESNUM(CT5)) THEN
                 BPRCLNUM=CT5
                 EXIT
                ENDIF
  616          CONTINUE
```

```
            IF (CLADDESNAME(BPRCLNUM).EQ.'SS304   ') THEN
             WRITE (100,618)
 618         FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055',
   c         ' 2.0 26304 69.5 28304 9.5')
             WRITE (100,620) CLADDESNUM(BPRCLNUM), CLTEMP
 620         FORMAT (T12,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'SS304S ') THEN
             WRITE (100,622)
 622         FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055',
   c         ' 2.0 26000 69.5 28000 9.5')
             WRITE (100,624) CLADDESNUM(BPRCLNUM), CLTEMP
 624         FORMAT (T13,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'SS316   ') THEN
             WRITE (100,626)
 626         FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000',
   c         ' 1.0 24304 17.0 25055 2.0')
             WRITE (100,628)
 628         FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
             WRITE (100,630) CLADDESNUM(BPRCLNUM), CLTEMP
 630         FORMAT (T12,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'SS316S ') THEN
             WRITE (100,632)
 632         FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000',
   c         ' 1.0 24000 17.0 25055 2.0')
             WRITE (100,633)
 633         FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')
             WRITE (100,634) CLADDESNUM(BPRCLNUM), CLTEMP
 634         FORMAT (T13,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'INCONEL') THEN
             WRITE (100,635)
 635         FORMAT ('arbm-inconel 8.3 5 0 0 0 14000 2.5',
   c         ' 22000 2.5 24000 15.0')
             WRITE (100,636)
 636         FORMAT (T13,'26000 7.0 28000 73.0')
             WRITE (100,637) CLADDESNUM(BPRCLNUM), CLTEMP
 637         FORMAT (T13,I2,' 1.0 ',F5.1,' end')
            ENDIF
          ENDIF
          IF (ABOVEBP(BPDESID(CT1)).EQ.'AL2O3') THEN
             ALFRAC=(((BPDEN(BPDESID(CT1)))*2.0*26.981539)/
   c         (101.9631))/BPDEN(BPDESID(CT1))
             OFRAC=1.0-ALFRAC
             IF (BPDEN(BPDESID(CT1)).LT.(1.0)) THEN
                WRITE (100,638) ABOVEBPNUM(BPDESID(CT1)),
   c            BPDEN(BPDESID(CT1)), ALFRAC,
   c            MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
 638            FORMAT ('al',3X,I3,3X,'den=',F4.3,1X,F7.5,
   c            1X,F7.1,1X,'end')
                WRITE (100,640) ABOVEBPNUM(BPDESID(CT1)),
   c            BPDEN(BPDESID(CT1)), OFRAC,
   c            MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
 640            FORMAT ('o',3X,I3,3X,'den=',F4.3,1X,F7.5,
   c            1X,F7.1,1X,'end')
             ELSE
                WRITE (100,642) ABOVEBPNUM(BPDESID(CT1)),
   c            BPDEN(BPDESID(CT1)), ALFRAC,
   c            MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
 642            FORMAT ('al',3X,I3,3X,'den=',F5.3,1X,F7.5,
   c            1X,F7.1,1X,'end')
                WRITE (100,644) ABOVEBPNUM(BPDESID(CT1)),
   c            BPDEN(BPDESID(CT1)), OFRAC,
   c            MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
 644            FORMAT ('o',3X,I3,3X,'den=',F5.3,1X,F7.5,
   c            1X,F7.1,1X,'end')
```

```
              ENDIF
           ELSE
              WRITE (100,*) 'arbm-bp    ',
     c        BPDEN(BPRA_DESCRIPTION_ID),
     c        '    ', BPFMNUMISOS(BPRA_DESCRIPTION_ID),
     c        '  0   0   0'
              DO 650 CT4=1,BPFMNUMISOS(BPRA_DESCRIPTION_ID)
                 WRITE (100,648)
     c           BPFISOID(BPRA_DESCRIPTION_ID,CT4),
     c           BPFISOWTPCT(BPRA_DESCRIPTION_ID,CT4)
  648            FORMAT (10X,I6,3X,F10.5)
  650         CONTINUE
              WRITE (100,*) '            ',
     c        ABOVEBPNUM(BPRA_DESCRIPTION_ID),
     c        ' 1.0 ',MODTEMPFINAL(CT3,RELATIVE_STPT_NUM),
     c        '   end'
           ENDIF
        ENDIF
*  Actual BPRA specifications
        BPRA_INSERTED=.FALSE.
        IF ((BPRFLAG.EQ.'Y').AND.(BPDESID(CT1).NE.0).AND.
     c     (CT3.GE.BPTN(CT1)).AND.(CT3.LE.BPBN(CT1))) THEN
           BPRA_INSERTED=.TRUE.
           BPRA_DESCRIPTION_ID=BPDESID(CT1)
           WRITE (100,685)
  685      FORMAT ('''')
           IF (BPMIX(BPRA_DESCRIPTION_ID).EQ.0) THEN
              WRITE (100,690) BPWTPCT(BPDESID(CT1))
  690         FORMAT ('''',5X,'Al2O3-B4C burnable absorber pellet',1X,
     c        'specification (',F4.2,1X,'wt% b4c)')
           ELSE
              WRITE (100,695)
  695         FORMAT ('''',5X,'burnable absorber pellet ',
     c        'specification')
           ENDIF
           WRITE (100,700)
  700      FORMAT ('''')
*  Write B4C material specification
           IF ((BPRCLAD(BPDESID(CT1)).NE.0).AND.
     c     (BPRCLAD(BPDESID(CT1)).NE.2)) THEN
             DO 701 CT5=1,10
              IF (BPRCLAD(BPDESID(CT1)).EQ.CLADDESNUM(CT5)) THEN
               BPRCLNUM=CT5
               EXIT
              ENDIF
  701        CONTINUE
             IF (CLADDESNAME(BPRCLNUM).EQ.'SS304   ') THEN
              WRITE (100,702)
  702         FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055',
     c        ' 2.0 26304 69.5 28304 9.5')
              WRITE (100,703) CLADDESNUM(BPRCLNUM), CLTEMP
  703         FORMAT (T12,I2,' 1.0 ',F5.1,' end')
             ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'SS304S ') THEN
              WRITE (100,704)
  704         FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055',
     c        ' 2.0 26000 69.5 28000 9.5')
              WRITE (100,705) CLADDESNUM(BPRCLNUM), CLTEMP
  705         FORMAT (T13,I2,' 1.0 ',F5.1,' end')
             ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'SS316   ') THEN
              WRITE (100,706)
  706         FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000',
     c        ' 1.0 24304 17.0 25055 2.0')
              WRITE (100,707)
  707         FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
```

```
              WRITE (100,708) CLADDESNUM(BPRCLNUM), CLTEMP
  708         FORMAT (T12,I2,' 1.0 ',F5.1,' end')
             ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'SS316S ') THEN
              WRITE (100,709)
  709         FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000',
     c        ' 1.0 24000 17.0 25055 2.0')
              WRITE (100,710)
  710         FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')
              WRITE (100,711) CLADDESNUM(BPRCLNUM), CLTEMP
  711         FORMAT (T13,I2,' 1.0 ',F5.1,' end')
             ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'INCONEL') THEN
              WRITE (100,712)
  712         FORMAT ('arbm-inconel 8.3 5 0 0 0 14000 2.5',
     c        ' 22000 2.5 24000 15.0')
              WRITE (100,713)
  713         FORMAT (T13,'26000 7.0 28000 73.0')
              WRITE (100,714) CLADDESNUM(BPRCLNUM), CLTEMP
  714         FORMAT (T13,I2,' 1.0 ',F5.1,' end')
             ENDIF
           ENDIF
*  Material specification for AL2O3-B4C
           IF ((BPMIX(BPRA_DESCRIPTION_ID).EQ.0).OR.
     c     (BPMIX(BPRA_DESCRIPTION_ID).EQ.4)) THEN
              IF (BPWTPCT(BPDESID(CT1)).NE.(0.0)) THEN
                 IF (BPDEN(BPDESID(CT1)).LT.(1.0)) THEN
                    WRITE (100,718) BPDEN(BPDESID(CT1)),
     c              (BPWTPCT(BPDESID(CT1))/100.0),
     c              MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  718               FORMAT ('b4c    4    den=',F4.3,1X,F7.5,1X,F7.1,1X,
     c              'end')
                 ELSE
                    WRITE (100,720) BPDEN(BPDESID(CT1)),
     c              (BPWTPCT(BPDESID(CT1))/100.0),
     c              MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  720               FORMAT ('b4c    4    den=',F5.3,1X,F7.5,1X,F7.1,1X,
     c              'end')
                 ENDIF
              ENDIF
*  Calculate aluminum and oxygen material specifications
              ALFRAC=(((((100.0-BPWTPCT(BPDESID(CT1)))/100.0)*
     c        BPDEN(BPDESID(CT1)))*2.0*26.981539)/(101.9631))/
     c        BPDEN(BPDESID(CT1))
              OFRAC=1.0-(BPWTPCT(BPDESID(CT1))/100.0)-ALFRAC
              IF (BPDEN(BPDESID(CT1)).LT.(1.0)) THEN
                 WRITE (100,734) BPDEN(BPDESID(CT1)), ALFRAC,
     c           MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  734         FORMAT ('al     4    den=',F4.3,1X,F7.5,1X,F7.1,1X,'end')
                 WRITE (100,736) BPDEN(BPDESID(CT1)), OFRAC,
     c           MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  736         FORMAT ('o      4    den=',F4.3,1X,F7.5,1X,F7.1,1X,'end')
              ELSE
                 WRITE (100,738) BPDEN(BPDESID(CT1)), ALFRAC,
     c           MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  738         FORMAT ('al     4    den=',F5.3,1X,F7.5,1X,F7.1,1X,'end')
                 WRITE (100,740) BPDEN(BPDESID(CT1)), OFRAC,
     c           MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  740         FORMAT ('o      4    den=',F5.3,1X,F7.5,1X,F7.1,1X,'end')
              ENDIF
           ELSE
*  Material specification for BP other than Al2O3-B4C
              DO 742 CT4=1,BPMIXNUM
                 IF (BPMIXID(CT4).EQ.BPMIX(BPRA_DESCRIPTION_ID)) THEN
                    RELATIVE_BP_MIX_ID=CT4
                 ENDIF
```

```
 742        CONTINUE
            WRITE (100,*) 'arbm-bp    ', BPDEN(BPRA_DESCRIPTION_ID),
     c      '    ', BPNUMISOS(RELATIVE_BP_MIX_ID),
     c      '  0   0   0'
            DO 750 CT4=1,BPNUMISOS(RELATIVE_BP_MIX_ID)
               WRITE (100,745) BPISOID(RELATIVE_BP_MIX_ID,CT4),
     c         BPISOWTPCT(RELATIVE_BP_MIX_ID,CT4)
 745           FORMAT (10X,I6,3X,F10.5)
 750        CONTINUE
            WRITE (100,*) '             ', BPMIX(BPRA_DESCRIPTION_ID),
     c      ' 1.0 ',MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '    end'
         ENDIF
      ENDIF
*  Write control rod material specification
      CR_INSERTED=.FALSE.
      IF (CRSTAT.EQ.'RODDED') THEN
      IF (RTYPE.EQ.'PWR') THEN
         CRCOMPFLAG=.FALSE.
         DO 760 CT4=1,23
            IF (CRINS(CT1,CT2,CT4,CT3).NE.0) THEN
               CRCOMPFLAG=.TRUE.
               CR_INSERTED=.TRUE.
               CR_MIXTURE_ID=CRINS(CT1,CT2,CT4,CT3)
               CR_DESCRIPTION=CRDES(CT1,CT2,CT4,CT3)
               EXIT
            ENDIF
 760     CONTINUE
         IF (CRCOMPFLAG.EQ..TRUE.) THEN
            DO 770 CT4=1,CRMIXNUM
               IF (CRMIXID(CT4).EQ.CR_MIXTURE_ID) THEN
                  RELATIVE_CR_MIX_ID=CT4
               ENDIF
 770        CONTINUE
            WRITE (100,780)
 780        FORMAT ('''')
            WRITE (100,790)
 790        FORMAT ('''',T5,' control rod material specification')
            WRITE (100,800)
 800        FORMAT ('''')
         IF (CRCLAD(CR_DESCRIPTION).NE.0) THEN
           DO 801 CT5=1,10
            IF (CRCLAD(CR_DESCRIPTION).EQ.CLADDESNUM(CT5)) THEN
             CRCLNUM=CT5
             EXIT
            ENDIF
 801       CONTINUE
           IF (CLADDESNAME(CRCLNUM).EQ.'SS304   ') THEN
            WRITE (100,802)
 802        FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055 ',
     c      '2.0 26304 69.5 28304 9.5')
            WRITE (100,803) CLADDESNUM(CRCLNUM), CLTEMP
 803        FORMAT (T12,I2,' 1.0 ',F5.1,' end')
           ELSEIF (CLADDESNAME(CRCLNUM).EQ.'SS304S ') THEN
            WRITE (100,804)
 804        FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055 ',
     c      '2.0 26000 69.5 28000 9.5')
            WRITE (100,805) CLADDESNUM(CRCLNUM), CLTEMP
 805        FORMAT (T13,I2,' 1.0 ',F5.1,' end')
           ELSEIF (CLADDESNAME(CRCLNUM).EQ.'SS316   ') THEN
            WRITE (100,806)
 806        FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000 ',
     c      '1.0 24304 17.0 25055 2.0')
            WRITE (100,807)
 807        FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
```

```
                WRITE (100,808) CLADDESNUM(CRCLNUM), CLTEMP
 808            FORMAT (T12,I2,' 1.0 ',F5.1,' end')
               ELSEIF (CLADDESNAME(CRCLNUM).EQ.'SS316S ') THEN
                WRITE (100,809)
 809            FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000 ',
    c           '1.0 24000 17.0 25055 2.0')
                WRITE (100,810)
 810            FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')
                WRITE (100,811) CLADDESNUM(CRCLNUM), CLTEMP
 811            FORMAT (T13,I2,' 1.0 ',F5.1,' end')
               ELSEIF (CLADDESNAME(CRCLNUM).EQ.'INCONEL') THEN
                WRITE (100,812)
 812            FORMAT ('arbm-inconel 8.3 5 0 0 0 14000 2.5',
    c           ' 22000 2.5 24000 15.0')
                WRITE (100,813)
 813            FORMAT (T13,'26000 7.0 28000 73.0')
                WRITE (100,814) CLADDESNUM(CRCLNUM), CLTEMP
 814            FORMAT (T13,I2,' 1.0 ',F5.1,' end')
               ENDIF
             ENDIF
                WRITE (100,*) 'arbm-cr   ', CRDEN(CR_DESCRIPTION),
    c           '    ', CRNUMISOS(RELATIVE_CR_MIX_ID), '   0   0   0'
                DO 820 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
                   WRITE (100,815) CRISOID(RELATIVE_CR_MIX_ID,CT4),
    c              CRISOWTPCT(RELATIVE_CR_MIX_ID, CT4)
 815               FORMAT (10X,I5,3X,F10.5)
 820            CONTINUE
                WRITE (100,*) '           ', CR_MIXTURE_ID, ' 1.0 ',
    c           MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '    end'
             ENDIF
* ***************************************************************************************
          ELSEIF (RTYPE.EQ.'BWR') THEN
c              CRCOMPFLAG=.FALSE.
             CRCOMPFLAG=.TRUE.
             CR_INSERTED=.TRUE.
             CR_DESCRIPTION=1
             DO 1500 CT4=1,23
                IF (CRINS(CT1,CT2,CT4,CT3).NE.0) THEN
                   CRCOMPFLAG=.TRUE.
                   CR_INSERTED=.TRUE.
C                   CR_DESCRIPTION=CRDES(CT1,CT2,CT4,CT3)
                   CR_DESCRIPTION=1
                   EXIT
                ENDIF
 1500        CONTINUE
             IF (CRCOMPFLAG.EQ..TRUE.) THEN
                WRITE (100,1510)
 1510           FORMAT ('''')
                WRITE (100,1520)
 1520           FORMAT ('''',T5,' GDR/CRB/MOD material',
    &                   ' specifications')
                WRITE (100,1530)
 1530           FORMAT ('''')
             IF (CRCLAD(CR_DESCRIPTION).NE.0) THEN
               DO 1540 CT5=1,10
                IF (CRCLAD(CR_DESCRIPTION).EQ.CLADDESNUM(CT5)) THEN
                 CRCLNUM=CT5
                 EXIT
                ENDIF
 1540          CONTINUE
               IF (CLADDESNAME(CRCLNUM).EQ.'SS304  ') THEN
                WRITE (100,1550)
 1550           FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055 ',
    c           '2.0 26304 69.5 28304 9.5')
```

```
               WRITE (100,1560) CLADDESNUM(CRCLNUM), CLTEMP
 1560          FORMAT (T12,I2,' 1.0 ',F5.1,' end')
              ELSEIF (CLADDESNAME(CRCLNUM).EQ.'SS304S ') THEN
               WRITE (100,1570)
 1570          FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055 ',
     c         '2.0 26000 69.5 28000 9.5')
               WRITE (100,1580) CLADDESNUM(CRCLNUM), CLTEMP
 1580          FORMAT (T13,I2,' 1.0 ',F5.1,' end')
              ELSEIF (CLADDESNAME(CRCLNUM).EQ.'SS316  ') THEN
               WRITE (100,1590)
 1590          FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000 ',
     c         '1.0 24304 17.0 25055 2.0')
               WRITE (100,1600)
 1600          FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
               WRITE (100,1610) CLADDESNUM(CRCLNUM), CLTEMP
 1610          FORMAT (T12,I2,' 1.0 ',F5.1,' end')
              ELSEIF (CLADDESNAME(CRCLNUM).EQ.'SS316S ') THEN
               WRITE (100,1620)
 1620          FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000 ',
     c         '1.0 24000 17.0 25055 2.0')
               WRITE (100,1630)
 1630          FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')
               WRITE (100,1640) CLADDESNUM(CRCLNUM), CLTEMP
 1640          FORMAT (T13,I2,' 1.0 ',F5.1,' end')
              ELSEIF (CLADDESNAME(CRCLNUM).EQ.'INCONEL') THEN
               WRITE (100,1650)
 1650          FORMAT ('arbm-inconel 8.3 5 0 0 0 14000 2.5',
     c         ' 22000 2.5 24000 15.0')
               WRITE (100,1660)
 1660          FORMAT (T13,'26000 7.0 28000 73.0')
               WRITE (100,1670) CLADDESNUM(CRCLNUM), CLTEMP
 1670          FORMAT (T13,I2,' 1.0 ',F5.1,' end')
              ENDIF
            ENDIF
            DO 1672 RELATIVE_CR_MIX_ID=1,CRMIXNUM
               DO 1671 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
                  UO2FLAG(RELATIVE_CR_MIX_ID)=.FALSE.
c                  IF (CRISOID(RELATIVE_CR_MIX_ID,CT4)
c    c                .EQ.'92235') THEN
c                     UO2FLAG(RELATIVE_CR_MIX_ID)=.TRUE.
c                  ELSE
c                     UO2FLAG(RELATIVE_CR_MIX_ID)=.FALSE.
c                  ENDIF
 1671          CONTINUE
 1672       CONTINUE
            DO 1720 RELATIVE_CR_MIX_ID=1,CRMIXNUM
             IF (UO2FLAG(RELATIVE_CR_MIX_ID).EQ..TRUE.) THEN
                IF (FDEN.LT.(10.0)) THEN
                   WRITE (100,1673) RELATIVE_CR_MIX_ID,
     c             CRMIXDEN(RELATIVE_CR_MIX_ID),
     c             FTFINAL(CT3,RELATIVE_STPT_NUM), WT234,
     c             WT235, WT236, WT238
 1673              FORMAT ('uo2',1X,I2,1X,'den=',F6.3,1X,'1',1X,
     c             F6.1,1X,'92234',
     c             1X,F5.4,1X,'92235',1X,F6.4,1X,'92236',
     c             1X,F5.4,1X,
     c             '92238',1X,F6.3,1X,'end')
                ELSE
                   WRITE (100,1674) RELATIVE_CR_MIX_ID,
     c             CRMIXDEN(RELATIVE_CR_MIX_ID),
     c             FTFINAL(CT3,RELATIVE_STPT_NUM), WT234,
     c             WT235, WT236, WT238
 1674              FORMAT ('uo2',1X,I2,1X,'den=',F6.3,1X,'1',1X,
     c             F6.1,1X,'92234',
```

```
     c            1X,F5.4,1X,'92235',1X,F6.4,1X,'92236',
     c            1X,F5.4,1X,
     c            '92238',1X,F6.3,1X,'end')
               ENDIF
               IF (RELATIVE_CR_MIX_ID.LT.10) THEN
                  WRITE (100,1675) RELATIVE_CR_MIX_ID,
     c            CRMIXDEN(RELATIVE_CR_MIX_ID),
     c            CRNUMISOS(RELATIVE_CR_MIX_ID)
c 1675             FORMAT(T1,'arbm-cr',I1,3X,
 1675             FORMAT(T1,'arbm-gcm',I1,3X,
     c            G14.8,3X,I2,'   0    0    0')
               ELSEIF (RELATIVE_CR_MIX_ID.EQ.10) THEN
                  WRITE (100,1676) RELATIVE_CR_MIX_ID,
     c            CRMIXDEN(RELATIVE_CR_MIX_ID),
     c            CRNUMISOS(RELATIVE_CR_MIX_ID)
c 1676             FORMAT(T1,'arbm-cr',I2,3X,
 1676             FORMAT(T1,'arbm-gcm',I2,3X,
     c            G14.8,3X,I2,'   0    0    0')
               ENDIF
               DO 1678 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
c                    IF ((CRISOID(RELATIVE_CR_MIX_ID,CT4).NE.'92234').AND.
c     c              (CRISOID(RELATIVE_CR_MIX_ID,CT4).NE.'92235').AND.
c     c              (CRISOID(RELATIVE_CR_MIX_ID,CT4).NE.'92236').AND.
c     c              (CRISOID(RELATIVE_CR_MIX_ID,CT4).NE.'92238')) THEN
                       WRITE (100,1677) CRISOID(RELATIVE_CR_MIX_ID,CT4),
     c                 CRISOWTPCT(RELATIVE_CR_MIX_ID, CT4)
 1677                  FORMAT (10X,I5,3X,F10.5)
c                    ENDIF
 1678          CONTINUE
               WRITE (100,*) '             ', CRMIXID(RELATIVE_CR_MIX_ID),
     c         ' 0.03 ', MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '    end'
            ELSE
               IF (RELATIVE_CR_MIX_ID.LT.10) THEN
                  WRITE (100,1679) RELATIVE_CR_MIX_ID,
     c            CRMIXDEN(RELATIVE_CR_MIX_ID),
     c            CRNUMISOS(RELATIVE_CR_MIX_ID)
c 1679             FORMAT(T1,'arbm-cr',I1,3X,
 1679             FORMAT(T1,'arbm-gcm',I1,3X,
     c            G14.8,3X,I2,'   0    0    0')
               ELSEIF (RELATIVE_CR_MIX_ID.EQ.10) THEN
                  WRITE (100,1680) RELATIVE_CR_MIX_ID,
     c            CRMIXDEN(RELATIVE_CR_MIX_ID),
     c            CRNUMISOS(RELATIVE_CR_MIX_ID)
c 1680             FORMAT(T1,'arbm-cr',I2,3X,
 1680             FORMAT(T1,'arbm-gcm',I2,3X,
     c            G14.8,3X,I2,'   0    0    0')
               ENDIF
               DO 1682 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
                  WRITE (100,1681) CRISOID(RELATIVE_CR_MIX_ID,CT4),
     c            CRISOWTPCT(RELATIVE_CR_MIX_ID, CT4)
 1681             FORMAT (10X,I5,3X,F10.5)
 1682          CONTINUE
              IF(RELATIVE_CR_MIX_ID.EQ.1)THEN
               WRITE (100,*) '             ',CRMIXID(RELATIVE_CR_MIX_ID),
     c         ' 1.0 ', FTFINAL(CT3,RELATIVE_STPT_NUM), '    end'
              ELSE
               WRITE (100,*) '             ',CRMIXID(RELATIVE_CR_MIX_ID),
     c         ' 1.0 ', MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '    end'
              ENDIF
            ENDIF
 1720     CONTINUE
          ENDIF
       ENDIF
       ENDIF
```

```
*  Write APSR material specification
         IF ((CT1.EQ.CT1START).AND.(CT2.EQ.CT2GOVALUE).AND.
     c   (CT3.EQ.1)) THEN
         DO 824 CT4=1,10
            DO 823 CT5=1,20
               DO 822 CT6=1,23
                  DO 821 CT7=1,50
                     APSRINSOLD(CT4,CT5,CT6,CT7)=
     c               APSRINS(CT4,CT5,CT6,CT7)
 821              CONTINUE
 822           CONTINUE
 823        CONTINUE
 824     CONTINUE
         ENDIF
         APSR_INSERTED=.FALSE.
         IF (APSRSTAT.EQ.'RODDED') THEN
            DO 830 CT4=1,23
               APSRBOTFLAG=.FALSE.
               DO 825 CT5=50,1,-1
                  IF ((APSRINSOLD(CT1,CT2,CT4,CT5).NE.0).AND.
     c            (APSRBOTFLAG.EQ..FALSE.)) THEN
                     APSR_DESCRIPTION=APSRDES(CT1,CT2,CT4,CT5)
                     APSRBOTFLAG=.TRUE.
                     FOLNODKEEP=CT5
                     FOLSTEPKEEP=CT4
                  ENDIF
                  IF ((APSRINSOLD(CT1,CT2,CT4,CT5).EQ.0).AND.
     c            (APSRBOTFLAG.EQ..TRUE.)) THEN
                     APSRINS(CT1,CT2,CT4,CT5)=
     c               APSRFOLLOWMIX(CT1,CT2,FOLSTEPKEEP,FOLNODKEEP)
                     APSRFOLLOWDATA(CT1,CT2,CT4,CT5)=3
                  ENDIF
 825           CONTINUE
 830        CONTINUE
            FOLLOWIN=.FALSE.
            DO 831 CT4=1,23
               IF (APSRFOLLOWDATA(CT1,CT2,CT4,CT3).EQ.3) THEN
                  FOLLOWIN=.TRUE.
                  EXIT
               ENDIF
 831        CONTINUE
            IF (FOLLOWIN.EQ..TRUE.) THEN
               WRITE (100,832)
 832           FORMAT ('''')
               WRITE (100,834)
 834           FORMAT ('''',T5,' APSR follow rod material',
     c         ' specification')
               WRITE (100,836)
 836           FORMAT ('''')
            IF ((APSRFOLLOWMIX(CT1,CT2,FOLSTEPKEEP,FOLNODKEEP).NE.0)
     c      .AND.
     c      (APSRFOLLOWMIX(CT1,CT2,FOLSTEPKEEP,FOLNODKEEP).NE.2)) THEN
              DO 838 CT5=1,10
               IF (APSRFOLLOWMIX(CT1,CT2,FOLSTEPKEEP,FOLNODKEEP)
     c         .EQ.CLADDESNUM(CT5)) THEN
                APSRFOLNUM=CT5
                EXIT
               ENDIF
 838          CONTINUE
              IF (CLADDESNAME(APSRFOLNUM).EQ.'SS304   ') THEN
               WRITE (100,840)
 840           FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055 ',
     c         '2.0 26304 69.5 28304 9.5')
               WRITE (100,842) CLADDESNUM(APSRFOLNUM), CLTEMP
```

```
842          FORMAT (T12,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(APSRFOLNUM).EQ.'SS304S ') THEN
             WRITE (100,844)
844          FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055 ',
   c         '2.0 26000 69.5 28000 9.5')
             WRITE (100,846) CLADDESNUM(APSRFOLNUM), CLTEMP
846          FORMAT (T13,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(APSRFOLNUM).EQ.'SS316  ') THEN
             WRITE (100,848)
848          FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000 ',
   c         '1.0 24304 17.0 25055 2.0')
             WRITE (100,850)
850          FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
             WRITE (100,852) CLADDESNUM(APSRFOLNUM), CLTEMP
852          FORMAT (T12,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(APSRFOLNUM).EQ.'SS316S ') THEN
             WRITE (100,854)
854          FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000 ',
   c         '1.0 24000 17.0 25055 2.0')
             WRITE (100,856)
856          FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')
             WRITE (100,858) CLADDESNUM(APSRFOLNUM), CLTEMP
858          FORMAT (T13,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(APSRFOLNUM).EQ.'INCONEL') THEN
             WRITE (100,860)
860          FORMAT ('arbm-inconel 8.3 5 0 0 0 14000 2.5',
   c         ' 22000 2.5 24000 15.0')
             WRITE (100,862)
862          FORMAT (T13,'26000 7.0 28000 73.0')
             WRITE (100,864) CLADDESNUM(APSRFOLNUM), CLTEMP
864          FORMAT (T13,I2,' 1.0 ',F5.1,' end')
            ENDIF
          ENDIF
          ENDIF
          APSRCOMPFLAG=.FALSE.
          DO 865 CT4=1,23
             IF ((APSRINS(CT1,CT2,CT4,CT3).NE.0).AND.
   c         (APSRINS(CT1,CT2,CT4,CT3).NE.
   c         APSRFOLLOWMIX(CT1,CT2,FOLSTEPKEEP,FOLNODKEEP))) THEN
                APSRCOMPFLAG=.TRUE.
                APSR_INSERTED=.TRUE.
                APSR_MIXTURE_ID=APSRINS(CT1,CT2,CT4,CT3)
                APSR_DESCRIPTION=APSRDES(CT1,CT2,CT4,CT3)
                EXIT
             ENDIF
865       CONTINUE
          IF (APSRCOMPFLAG.EQ..TRUE.) THEN
             DO 866 CT4=1,APSRMIXNUM
                IF (APSRMIXID(CT4).EQ.APSR_MIXTURE_ID) THEN
                   RELATIVE_APSR_MIX_ID=CT4
                ENDIF
866          CONTINUE
             WRITE (100,868)
868          FORMAT ('''')
             WRITE (100,870)
870          FORMAT ('''',T5,' axial power shaping rod material',
   c         ' specification')
             WRITE (100,880)
880          FORMAT ('''')
          IF (APSRCLAD(APSR_DESCRIPTION).NE.0) THEN
            DO 881 CT5=1,10
             IF (APSRCLAD(APSR_DESCRIPTION).EQ.CLADDESNUM(CT5)) THEN
              APSRCLNUM=CT5
              EXIT
```

```
             ENDIF
 881        CONTINUE
            IF (CLADDESNAME(APSRCLNUM).EQ.'SS304   ') THEN
             WRITE (100,882)
 882         FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055 ',
    c        '2.0 26304 69.5 28304 9.5')
             WRITE (100,883) CLADDESNUM(APSRCLNUM), CLTEMP
 883         FORMAT (T12,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(APSRCLNUM).EQ.'SS304S ') THEN
             WRITE (100,884)
 884         FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055 ',
    c        '2.0 26000 69.5 28000 9.5')
             WRITE (100,885) CLADDESNUM(APSRCLNUM), CLTEMP
 885         FORMAT (T13,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(APSRCLNUM).EQ.'SS316   ') THEN
             WRITE (100,886)
 886         FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000 ',
    c        '1.0 24304 17.0 25055 2.0')
             WRITE (100,887)
 887         FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
             WRITE (100,888) CLADDESNUM(APSRCLNUM), CLTEMP
 888         FORMAT (T12,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(APSRCLNUM).EQ.'SS316S ') THEN
             WRITE (100,889)
 889         FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000 ',
    c        '1.0 24000 17.0 25055 2.0')
             WRITE (100,890)
 890         FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')
             WRITE (100,891) CLADDESNUM(APSRCLNUM), CLTEMP
 891         FORMAT (T13,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(APSRCLNUM).EQ.'INCONEL') THEN
             WRITE (100,892)
 892         FORMAT ('arbm-inconel 8.3 5 0 0 0 14000 2.5',
    c        ' 22000 2.5 24000 15.0')
             WRITE (100,893)
 893         FORMAT (T13,'26000 7.0 28000 73.0')
             WRITE (100,894) CLADDESNUM(APSRCLNUM), CLTEMP
 894         FORMAT (T13,I2,' 1.0 ',F5.1,' end')
            ENDIF
          ENDIF
             WRITE (100,*) 'arbm-apsr    ', APSRDEN(APSR_DESCRIPTION),
    c        '    ', APSRNUMISOS(RELATIVE_APSR_MIX_ID), '    0    0    0'
             DO 900 CT4=1,APSRNUMISOS(RELATIVE_APSR_MIX_ID)
                WRITE (100,895) APSRISOID(RELATIVE_APSR_MIX_ID,CT4),
    c           APSRISOWTPCT(RELATIVE_APSR_MIX_ID, CT4)
 895            FORMAT (10X,I5,3X,F10.5)
 900         CONTINUE
             WRITE (100,*) '           ', APSR_MIXTURE_ID, ' 1.0 ',
    c        MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '   end'
          ENDIF
       ENDIF
*  Write fuel rod fill gas material specification
       WRITE (100,910)
 910   FORMAT ('''')
       IF(RTYPE.EQ.'PWR')THEN
        WRITE (100,920)
 920    FORMAT ('he     5   end')
       ENDIF
       WRITE (100,930)
 930   FORMAT ('end comp')
*  Write base reactor lattice specifications
       WRITE (100,940)
 940   FORMAT ('''')
       WRITE (100,950)
```

```
  950     FORMAT ('''      base reactor lattice specification')
          WRITE (100,960)
  960     FORMAT ('''')
          WRITE (100,970) PITCH, FOD, COD
  970     FORMAT ('squarepitch',3X,F7.5,3X,F6.4,3X,'1   3',3X,F6.4,
     c    3X,'2',' end')
c          WRITE (100,970) PITCH, FOD, COD, CID
c  970     FORMAT ('squarepitch',3X,F7.5,3X,F6.4,3X,'1   3',3X,F6.4,
c     c    3X,'2',3X,F6.4,3X,'0   end')
* The following writing routine for 'SPECIAL' input data
* has not been formatted to compensate for FORTRAN's ingenious
* incapability to print leading zeros in numeric fields.
* Errors will occur in the FIDO input if null space exists
* between an equal sign and the appropriate value.  Therefore,
* the IIM and ICM factors must always be at least 10.
          IF (FLAG2.EQ.'SPECIAL') THEN
             IF (SZF.LT.1) THEN
                WRITE (100,980) SZF, ISN, IIM, ICM, EPS, PTC, IUS
  980           FORMAT ('more data',1X,'szf=0',F3.2,1X,'isn=',I1,1X,
     c          'iim=',I2,1X,'icm=',I2,1X,'eps=0',G7.2,1X,'ptc=0',G7.2,
     c          1X,'ius=',I1,3X,'end')
             ELSE
                WRITE (100,990) SZF, ISN, IIM, ICM, EPS, PTC, IUS
  990           FORMAT ('more data',1X,'szf=',F4.2,1X,'isn=',I1,1X,
     c          'iim=',I2,1X,'icm=',I2,1X,'eps=0',G7.2,1X,'ptc=0',G7.2,
     c          1X,'ius=',I1,3X,'end')
             ENDIF
          ELSEIF (FLAG2.NE.'SPECIAL') THEN
             IF (MESH.LT.1) THEN
                WRITE (100,1000) MESH
 1000           FORMAT ('more data',1X,'szf=0',F3.2,1X,'end')
             ELSE
                WRITE (100,1010) MESH
 1010           FORMAT ('more data',1X,'szf=',F4.2,1X,'end')
             ENDIF
          ENDIF
*  Write assembly specifications
          WRITE (100,1020)
 1020     FORMAT ('''')
          WRITE (100,1030)
 1030     FORMAT ('''      assembly specification')
          WRITE (100,1040)
 1040     FORMAT ('''')
          IF (STEPCONTROL.EQ.'Y') THEN
             CALL ZEROS(VARSTEPNUM(CT1,CT2),IRRAD_STEPS)
          ELSEIF (STEPCONTROL.EQ.'N') THEN
             CALL ZEROS(INT(BLETDOWN(CT1,CT2,2)),IRRAD_STEPS)
          ENDIF
*  Assembly specification if no BPRA, no CR, and no APSR is inserted
          IF ((BPRA_INSERTED.EQ..FALSE.).AND.(CR_INSERTED.EQ..FALSE.)
     c    .AND.(APSR_INSERTED.EQ..FALSE.)
     c    .AND.(BPRA_FOLLOW.EQ..FALSE.)
     c    .AND.(FOLLOWIN.EQ..FALSE.)) THEN
             IF (NODES(CT3,2).GE.(100.0)) THEN
                WRITE (100,1041) RODS, NODES(CT3,2), IRRAD_STEPS
 1041           FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F7.3,1X,
     c          'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
             ELSEIF ((NODES(CT3,2).LT.(100.0)).AND.
     c       (NODES(CT3,2).GE.(10.0))) THEN
                WRITE (100,1042) RODS, NODES(CT3,2), IRRAD_STEPS
 1042           FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F6.3,1X,
     c          'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
             ELSEIF (NODES(CT3,2).LT.(10.0)) THEN
                WRITE (100,1043) RODS, NODES(CT3,2), IRRAD_STEPS
```

```
 1043          FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F5.3,1X,
     c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
            ENDIF
            CALL ZEROS(PLEVEL,PLEVELCH)
            CALL ZEROS(LUZONE,LUZONECH)
            IF (MESH.LT.(1.0)) THEN
               WRITE (100,1044) PLEVELCH, LUZONECH, MESH
 1044          FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
     c         'numztotal=',A2,1X,'mxrepeats=1',1X,
     c         'mixmod=3 facmesh=',F3.2,1X,'end')
            ELSE
               WRITE (100,1045) PLEVELCH, LUZONECH, MESH
 1045          FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
     c         'numztotal=',A2,1X,'mxrepeats=1',1X,
     c         'mixmod=3 facmesh=',F4.2,1X,'end')
            ENDIF
            DO 1047 CT4=1,LUZONE
               IF (MOD(CT4,6).EQ.0) THEN
                  WRITE (100,*)
               ENDIF
               WRITE (100,1046) LMB(CT4), LRB(CT4)
 1046          FORMAT (I3,1X,F7.5,1X,$)
 1047       CONTINUE
            WRITE (100,*)
         ENDIF
*  Assembly specification if BPRA is inserted
         IF (BPRA_FOLLOW.EQ..TRUE.) THEN
            IF (NODES(CT3,2).GE.(100.0)) THEN
               WRITE (100,1050) RODS, NODES(CT3,2), IRRAD_STEPS
 1050          FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F7.3,1X,
     c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
            ELSEIF ((NODES(CT3,2).LT.(100.0)).AND.
     c      (NODES(CT3,2).GE.(10.0))) THEN
               WRITE (100,1052) RODS, NODES(CT3,2), IRRAD_STEPS
 1052          FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F6.3,1X,
     c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
            ELSEIF (NODES(CT3,2).LT.(10.0)) THEN
               WRITE (100,1054) RODS, NODES(CT3,2), IRRAD_STEPS
 1054          FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F5.3,1X,
     c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
            ENDIF
            CALL ZEROS(PLEVEL,PLEVELCH)
            CALL ZEROS(BPZONE(BPRA_DESCRIPTION_ID),BPZONECH)
            IF (MESH.LT.(1.0)) THEN
               WRITE (100,1056) PLEVELCH, BPZONECH, MESH
 1056          FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
     c         'numztotal=',A2,1X,'mxrepeats=1',1X,
     c         'mixmod=3 facmesh=',F3.2,1X,'end')
            ELSE
               WRITE (100,1058) PLEVELCH, BPZONECH, MESH
 1058          FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
     c         'numztotal=',A2,1X,'mxrepeats=1',1X,
     c         'mixmod=3 facmesh=',F4.2,1X,'end')
            ENDIF
            DO 1062 CT4=1,BPZONE(BPRA_DESCRIPTION_ID)
               IF (MOD(CT4,6).EQ.0) THEN
                  WRITE (100,*)
               ENDIF
               WRITE (100,1060) BPRFM(CT4,BPRA_DESCRIPTION_ID),
     c         BPRFR(CT4,BPRA_DESCRIPTION_ID)
 1060          FORMAT (I3,1X,F7.5,1X,$)
 1062       CONTINUE
            WRITE (100,*)
         ENDIF
```

```
      IF (BPRA_INSERTED.EQ..TRUE.) THEN
         IF (NODES(CT3,2).GE.(100.0)) THEN
            WRITE (100,1098) RODS, NODES(CT3,2), IRRAD_STEPS
1098        FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F7.3,1X,
   c        'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
         ELSEIF ((NODES(CT3,2).LT.(100.0)).AND.
   c     (NODES(CT3,2).GE.(10.0))) THEN
            WRITE (100,1100) RODS, NODES(CT3,2), IRRAD_STEPS
1100        FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F6.3,1X,
   c        'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
         ELSEIF (NODES(CT3,2).LT.(10.0)) THEN
            WRITE (100,1103) RODS, NODES(CT3,2), IRRAD_STEPS
1103        FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F5.3,1X,
   c        'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
         ENDIF
         CALL ZEROS(PLEVEL,PLEVELCH)
         CALL ZEROS(BPZONE(BPRA_DESCRIPTION_ID),BPZONECH)
         IF (MESH.LT.(1.0)) THEN
            WRITE (100,1104) PLEVELCH, BPZONECH, MESH
1104        FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
   c        'numztotal=',A2,1X,'mxrepeats=1',1X,
   c        'mixmod=3 facmesh=',F3.2,1X,'end')
         ELSE
            WRITE (100,1106) PLEVELCH, BPZONECH, MESH
1106        FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
   c        'numztotal=',A2,1X,'mxrepeats=1',1X,
   c        'mixmod=3 facmesh=',F4.2,1X,'end')
         ENDIF
         DO 1110 CT4=1,BPZONE(BPRA_DESCRIPTION_ID)
            IF (MOD(CT4,6).EQ.0) THEN
               WRITE (100,*)
            ENDIF
            WRITE (100,1108) BPMA(CT4,BPRA_DESCRIPTION_ID),
   c        BPRA(CT4,BPRA_DESCRIPTION_ID)
1108        FORMAT (I3,1X,F7.5,1X,$)
1110     CONTINUE
         WRITE (100,*)
      ENDIF
*  Assembly specification if CR is inserted
      IF (CR_INSERTED.EQ..TRUE.) THEN
         IF (NODES(CT3,2).GE.(100.0)) THEN
            WRITE (100,1120) RODS, NODES(CT3,2), IRRAD_STEPS
1120        FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F7.3,1X,
   c        'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
         ELSEIF ((NODES(CT3,2).LT.(100.0)).AND.
   c     (NODES(CT3,2).GE.(10.0))) THEN
            WRITE (100,1130) RODS, NODES(CT3,2), IRRAD_STEPS
1130        FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F6.3,1X,
   c        'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
         ELSEIF (NODES(CT3,2).LT.(10.0)) THEN
            WRITE (100,1140) RODS, NODES(CT3,2), IRRAD_STEPS
1140        FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F5.3,1X,
   c        'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
         ENDIF
         CALL ZEROS(PLEVEL,PLEVELCH)
         CALL ZEROS(CRZONE(CR_DESCRIPTION),CRZONECH)
         IF (MESH.LT.(1.0)) THEN
            WRITE (100,1150) PLEVELCH, CRZONECH, MESH
1150        FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
   c        'numztotal=',A2,1X,'mxrepeats=0',1X,
   c        'mixmod=3 facmesh=',F3.2,1X,'end')
         ELSE
            WRITE (100,1160) PLEVELCH, CRZONECH, MESH
1160        FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
```

```
     c          'numztotal=',A2,1X,'mxrepeats=0',1X,
     c          'mixmod=3 facmesh=',F4.2,1X,'end')
             ENDIF
             IF (STEPCONTROL.EQ.'N') THEN
              DO 1169 CT4=1,INT(BLETDOWN(CT1,CT2,2))
                IF (CRINS(CT1,CT2,CT4,CT3).NE.0) THEN
                   DO 1164 CT5=1,CRZONE(CR_DESCRIPTION)
                      IF (MOD(CT5,6).EQ.0) THEN
                         WRITE (100,*)
                      ENDIF
                      WRITE (100,1162) CRMA(CT5,CR_DESCRIPTION),
     c                CRRA(CT5,CR_DESCRIPTION)
 1162                 FORMAT (I3,1X,F7.5,1X,$)
 1164              CONTINUE
                   WRITE (100,*)
                ELSEIF (CRINS(CT1,CT2,CT4,CT3).EQ.0) THEN
                   DO 1168 CT5=1,CRZONE(CR_DESCRIPTION)
                      IF (MOD(CT5,6).EQ.0) THEN
                         WRITE (100,*)
                      ENDIF
                      WRITE (100,1166) LMC(CT5,CR_DESCRIPTION),
     c                LRC(CT5,CR_DESCRIPTION)
 1166                 FORMAT (I3,1X,F7.5,1X,$)
 1168              CONTINUE
                   WRITE (100,*)
                ENDIF
 1169         CONTINUE
             ELSEIF (STEPCONTROL.EQ.'Y') THEN
              DO 1210 CT4=1,VARSTEPNUM(CT1,CT2)
                IF (CRINS(CT1,CT2,CT4,CT3).NE.0) THEN
                   DO 1180 CT5=1,CRZONE(CR_DESCRIPTION)
                      IF (MOD(CT5,6).EQ.0) THEN
                         WRITE (100,*)
                      ENDIF
                      WRITE (100,1170) CRMA(CT5,CR_DESCRIPTION),
     c                CRRA(CT5,CR_DESCRIPTION)
 1170                 FORMAT (I3,1X,F7.5,1X,$)
 1180              CONTINUE
                   WRITE (100,*)
                ELSEIF (CRINS(CT1,CT2,CT4,CT3).EQ.0) THEN
                   DO 1200 CT5=1,CRZONE(CR_DESCRIPTION)
                      IF (MOD(CT5,6).EQ.0) THEN
                         WRITE (100,*)
                      ENDIF
                      WRITE (100,1190) LMC(CT5,CR_DESCRIPTION),
     c                LRC(CT5,CR_DESCRIPTION)
 1190                 FORMAT (I3,1X,F7.5,1X,$)
 1200              CONTINUE
                   WRITE (100,*)
                ENDIF
 1210         CONTINUE
             ENDIF
          ENDIF
*  Assembly specification if APSR is inserted
          IF ((APSR_INSERTED.EQ..TRUE.).OR.(FOLLOWIN.EQ..TRUE.)) THEN
             IF (NODES(CT3,2).GE.(100.0)) THEN
                WRITE (100,1220) RODS, NODES(CT3,2), IRRAD_STEPS
 1220           FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F7.3,1X,
     c          'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
             ELSEIF ((NODES(CT3,2).LT.(100.0)).AND.
     c       (NODES(CT3,2).GE.(10.0))) THEN
                WRITE (100,1230) RODS, NODES(CT3,2), IRRAD_STEPS
 1230           FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F6.3,1X,
     c          'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
```

```
          ELSEIF (NODES(CT3,2).LT.(10.0)) THEN
             WRITE (100,1240) RODS, NODES(CT3,2), IRRAD_STEPS
 1240        FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F5.3,1X,
    c        'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
          ENDIF
          CALL ZEROS(PLEVEL,PLEVELCH)
          CALL ZEROS(APSRZONE(APSR_DESCRIPTION),APSRZONECH)
          IF (MESH.LT.(1.0)) THEN
             WRITE (100,1250) PLEVELCH, APSRZONECH, MESH
 1250        FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
    c        'numztotal=',A2,1X,'mxrepeats=0',1X,
    c        'mixmod=3 facmesh=',F3.2,1X,'end')
          ELSE
             WRITE (100,1252) PLEVELCH, APSRZONECH, MESH
 1252        FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
    c        'numztotal=',A2,1X,'mxrepeats=0',1X,
    c        'mixmod=3 facmesh=',F4.2,1X,'end')
          ENDIF
          IF (STEPCONTROL.EQ.'N') THEN
           DO 1268 CT4=1,INT(BLETDOWN(CT1,CT2,2))
             IF ((APSRINS(CT1,CT2,CT4,CT3).NE.0).AND.
    c        (APSRFOLLOWDATA(CT1,CT2,CT4,CT3).NE.3)) THEN
                DO 1258 CT5=1,APSRZONE(APSR_DESCRIPTION)
                   IF (MOD(CT5,6).EQ.0) THEN
                      WRITE (100,*)
                   ENDIF
                   WRITE (100,1256) APSRMA(CT5,APSR_DESCRIPTION),
    c              APSRRA(CT5,APSR_DESCRIPTION)
 1256              FORMAT (I3,1X,F7.5,1X,$)
 1258           CONTINUE
                WRITE (100,*)
             ELSEIF ((APSRINS(CT1,CT2,CT4,CT3).NE.0).AND.
    c        (APSRFOLLOWDATA(CT1,CT2,CT4,CT3).EQ.3)) THEN
                DO 1262 CT5=1,APSRZONE(APSR_DESCRIPTION)
                   IF (MOD(CT5,6).EQ.0) THEN
                      WRITE (100,*)
                   ENDIF
                   WRITE (100,1260) APSRFM(CT5,APSR_DESCRIPTION),
    c              APSRFR(CT5,APSR_DESCRIPTION)
 1260              FORMAT (I3,1X,F7.5,1X,$)
 1262           CONTINUE
                WRITE (100,*)
             ELSEIF (APSRINS(CT1,CT2,CT4,CT3).EQ.0) THEN
                DO 1266 CT5=1,APSRZONE(APSR_DESCRIPTION)
                   IF (MOD(CT5,6).EQ.0) THEN
                      WRITE (100,*)
                   ENDIF
                   WRITE (100,1264) LMD(CT5,APSR_DESCRIPTION),
    c              LRD(CT5,APSR_DESCRIPTION)
 1264              FORMAT (I3,1X,F7.5,1X,$)
 1266           CONTINUE
                WRITE (100,*)
             ENDIF
 1268      CONTINUE
          ELSEIF (STEPCONTROL.EQ.'Y') THEN
           DO 1310 CT4=1,VARSTEPNUM(CT1,CT2)
             IF ((APSRINS(CT1,CT2,CT4,CT3).NE.0).AND.
    c        (APSRFOLLOWDATA(CT1,CT2,CT4,CT3).NE.3)) THEN
                DO 1280 CT5=1,APSRZONE(APSR_DESCRIPTION)
                   IF (MOD(CT5,6).EQ.0) THEN
                      WRITE (100,*)
                   ENDIF
                   WRITE (100,1270) APSRMA(CT5,APSR_DESCRIPTION),
    c              APSRRA(CT5,APSR_DESCRIPTION)
```

```
1270                FORMAT (I3,1X,F7.5,1X,$)
1280             CONTINUE
                 WRITE (100,*)
              ELSEIF ((APSRINS(CT1,CT2,CT4,CT3).NE.0).AND.
    c         (APSRFOLLOWDATA(CT1,CT2,CT4,CT3).EQ.3)) THEN
                 DO 1290 CT5=1,APSRZONE(APSR_DESCRIPTION)
                    IF (MOD(CT5,6).EQ.0) THEN
                       WRITE (100,*)
                    ENDIF
                    WRITE (100,1285) APSRFM(CT5,APSR_DESCRIPTION),
    c               APSRFR(CT5,APSR_DESCRIPTION)
1285                FORMAT (I3,1X,F7.5,1X,$)
1290             CONTINUE
                 WRITE (100,*)
              ELSEIF (APSRINS(CT1,CT2,CT4,CT3).EQ.0) THEN
                 DO 1300 CT5=1,APSRZONE(APSR_DESCRIPTION)
                    IF (MOD(CT5,6).EQ.0) THEN
                       WRITE (100,*)
                    ENDIF
                    WRITE (100,1295) LMD(CT5,APSR_DESCRIPTION),
    c               LRD(CT5,APSR_DESCRIPTION)
1295                FORMAT (I3,1X,F7.5,1X,$)
1300             CONTINUE
                 WRITE (100,*)
              ENDIF
1310        CONTINUE
           ENDIF
        ENDIF
*  Write assembly depletion/decay parameters
        WRITE (100,1320)
1320    FORMAT ('''')
        WRITE (100,1330)
1330    FORMAT ('''     assembly depletion/decay parameters')
        WRITE (100,1340)
1340    FORMAT ('''')
        CALL ZEROS(CYCPOS(CT1),ASSYPOSITION)
        WRITE (100,1350) CYCLEID(CT1), ASSYPOSITION
1350    FORMAT ('''',T5,'Cycle-',A2,',  ',
    c   ' assembly number ',A2)
        IF (STEPCONTROL.EQ.'N') THEN
         DO 1380 CT4=3,(INT(BLETDOWN(CT1,CT2,2))+2)
           IF (CT4.LT.(BLETDOWN(CT1,CT2,2)+2)) THEN
              DOWNTIME=0.0
              IF (RTYPE.EQ.'PWR') THEN
                 BORON_FRACTION=(BLETDOWN(CT1,CT2,CT4)/
    c            BLETDOWN(CT1,CT2,3))
                 WRITE (100,1360) POWER(CT3,RELATIVE_STPT_NUM),
    c            BLETDOWN(CT1,CT2,1), DOWNTIME, BORON_FRACTION
1360             FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
    c            G10.5,1X,'bfrac=',G9.4,1X,'end')
              ELSEIF (RTYPE.EQ.'BWR') THEN
                 BORON_FRACTION=(BLETDOWN(CT1,CT2,CT4)/
    c            MODREFDEN)
                 WRITE (100,1450) POWER(CT3,RELATIVE_STPT_NUM),
    c            BLETDOWN(CT1,CT2,1), DOWNTIME, BORON_FRACTION
1450             FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
    c            G10.5,1X,'h2ofrac=',G9.4,1X,'end')
              ENDIF
           ELSEIF ((CT4.EQ.(INT(BLETDOWN(CT1,CT2,2))+2)).AND.
    c      (CT2.LT.STPTS(CT1))) THEN
              DOWNTIME=STPTDAT(CT1,(CT2+1),3)
              IF (RTYPE.EQ.'PWR') THEN
                 BORON_FRACTION=(BLETDOWN(CT1,CT2,CT4)/
    c            BLETDOWN(CT1,CT2,3))
```

```
                 WRITE (100,1365) POWER(CT3,RELATIVE_STPT_NUM),
     c           BLETDOWN(CT1,CT2,1), DOWNTIME, BORON_FRACTION
 1365            FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c           G10.5,1X,'bfrac=',G9.4,1X,'end')
              ELSEIF (RTYPE.EQ.'BWR') THEN
                 BORON_FRACTION=(BLETDOWN(CT1,CT2,CT4)/
     c           MODREFDEN)
                 WRITE (100,1460) POWER(CT3,RELATIVE_STPT_NUM),
     c           BLETDOWN(CT1,CT2,1), DOWNTIME, BORON_FRACTION
 1460            FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c           G10.5,1X,'h2ofrac=',G9.4,1X,'end')
              ENDIF
           ELSEIF ((CT4.EQ.(INT(BLETDOWN(CT1,CT2,2))+2)).AND.
     c     (CT2.EQ.STPTS(CT1))) THEN
              DOWNTIME=CYCDOWN(CT1)
              IF (RTYPE.EQ.'PWR') THEN
                 BORON_FRACTION=(BLETDOWN(CT1,CT2,CT4)/
     c           BLETDOWN(CT1,CT2,3))
                 WRITE (100,1370) POWER(CT3,RELATIVE_STPT_NUM),
     c           BLETDOWN(CT1,CT2,1), DOWNTIME, BORON_FRACTION
 1370            FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c           G10.5,1X,'bfrac=',G9.4,1X,'end')
              ELSEIF (RTYPE.EQ.'BWR') THEN
                 BORON_FRACTION=(BLETDOWN(CT1,CT2,CT4)/
     c           MODREFDEN)
                 WRITE (100,1470) POWER(CT3,RELATIVE_STPT_NUM),
     c           BLETDOWN(CT1,CT2,1), DOWNTIME, BORON_FRACTION
 1470            FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c           G10.5,1X,'h2ofrac=',G9.4,1X,'end')
              ENDIF
           ENDIF
 1380    CONTINUE
        ELSEIF (STEPCONTROL.EQ.'Y') THEN
         DO 1388 CT4=1,VARSTEPNUM(CT1,CT2)
           IF (CT4.LT.VARSTEPNUM(CT1,CT2)) THEN
              DOWNTIME=0.0
              IF (RTYPE.EQ.'PWR') THEN
                 BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
     c           VARBLETDOWN(CT1,CT2,1,2))
                 WRITE (100,1382) VARPOWER(CT1,CT2,CT4,CT3),
     c           VARBLETDOWN(CT1,CT2,CT4,1), DOWNTIME, BORON_FRACTION
 1382            FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c           G10.5,1X,'bfrac=',G9.4,1X,'end')
              ELSEIF (RTYPE.EQ.'BWR') THEN
C                 BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
C     c           MODREFDEN)
                 BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
     c           FIRSTMODEN(CT1,CT2))
                 WRITE (100,1480) VARPOWER(CT1,CT2,CT4,CT3),
     c           VARBLETDOWN(CT1,CT2,CT4,1), DOWNTIME, BORON_FRACTION
 1480            FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c           G10.5,1X,'h2ofrac=',G10.5,1X,'end')
              ENDIF
           ELSEIF ((CT4.EQ.VARSTEPNUM(CT1,CT2)).AND.
     c     (CT2.LT.STPTS(CT1))) THEN
              DOWNTIME=STPTDAT(CT1,(CT2+1),3)
              IF (RTYPE.EQ.'PWR') THEN
                 BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
     c           VARBLETDOWN(CT1,CT2,1,2))
                 WRITE (100,1384) VARPOWER(CT1,CT2,CT4,CT3),
     c           VARBLETDOWN(CT1,CT2,CT4,1), DOWNTIME, BORON_FRACTION
 1384            FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c           G10.5,1X,'bfrac=',G9.4,1X,'end')
              ELSEIF (RTYPE.EQ.'BWR') THEN
```

```
C                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
C     c            MODREFDEN)
                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
     c            FIRSTMODEN(CT1,CT2))
                  WRITE (100,1490) VARPOWER(CT1,CT2,CT4,CT3),
     c            VARBLETDOWN(CT1,CT2,CT4,1), DOWNTIME, BORON_FRACTION
 1490             FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c            G10.5,1X,'h2ofrac=',G10.5,1X,'end')
               ENDIF
            ELSEIF ((CT4.EQ.VARSTEPNUM(CT1,CT2)).AND.
     c      (CT2.EQ.STPTS(CT1))) THEN
               DOWNTIME=CYCDOWN(CT1)
               IF (RTYPE.EQ.'PWR') THEN
                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
     c            VARBLETDOWN(CT1,CT2,1,2))
                  WRITE (100,1386) VARPOWER(CT1,CT2,CT4,CT3),
     c            VARBLETDOWN(CT1,CT2,CT4,1), DOWNTIME, BORON_FRACTION
 1386             FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c            G10.5,1X,'bfrac=',G9.4,1X,'end')
               ELSEIF (RTYPE.EQ.'BWR') THEN
C                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
C     c            MODREFDEN)
                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
     c            FIRSTMODEN(CT1,CT2))
                  WRITE (100,1800) VARPOWER(CT1,CT2,CT4,CT3),
     c            VARBLETDOWN(CT1,CT2,CT4,1), DOWNTIME, BORON_FRACTION
 1800             FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c            G10.5,1X,'h2ofrac=',G10.5,1X,'end')
               ENDIF
            ENDIF
 1388     CONTINUE
         ENDIF
*  Store final downtime for use in extraction script 'sedexecute.exe'
         FINALDOWNTIME=DOWNTIME
*  Write input deck closing statement
         WRITE (100,1390)
 1390    FORMAT ('''')
         WRITE (100,1400)
 1400    FORMAT ('''     end of input')
         WRITE (100,1410)
 1410    FORMAT ('''')
         WRITE (100,1420)
 1420    FORMAT ('end')
         CLOSE(UNIT=100)

      RETURN
      END


******************************************************************
*     Subroutine to write continuation depletion/decay SAS2H     *
*     input decks utilizing fuel and burnable poison compositions *
*     from the assembly's previous depletion/decay calculation   *
*     en-route to the final CRC depletion/decay calculation      *
******************************************************************
      SUBROUTINE CONTINUATION_WRITER (RELATIVE_STPT_NUM, CT1, CT2,
     c CT3, AXNUM, CYCPOS, AXBLANK, BPDESID,
     c CRINS, CRDES, CRMIXNUM, CRMIXID,
     c CRNUMISOS, CRISOID, APSRINS,
     c APSRMIXNUM, APSRMIXID, RELATIVE_APSR_MIX_ID,
     c APSRNUMISOS, APSRISOID, ISN, IIM, ICM, IUS,
     c PLEVEL, BPZONE, BPMA, CRZONE, CRMA,
     c LMC, APSRZONE, APSRMA, LMD,
     c BPTN, BPBN, STPTS, APSRDES,
```

```
     c STPTDAT, AXBLANKRICH, GRAMS,
     c NODES, RODS, RICH, FTFINAL, MODDENFINAL,
     c MODTEMPFINAL, BLETDOWN, BPWTPCT,
     c BPDEN, CRDEN, CRISOWTPCT, APSRDEN,
     c APSRISOWTPCT, PITCH, FOD, COD, CID, SZF, EPS, PTC,
     c MESH, BPRA, CRRA, LRC, APSRRA,
     c LRD, POWER, CYCDOWN, PREFIX, NM,
     c CYCLEID, REACT, LIB, AXBLANKET, FUELCLAD,
     c BPRFLAG, CRSTAT, APSRSTAT, FLAG2, LUZONE, LMB, LRB,
     c MASSTOTAL, FUELISONAME, FUELISOWTPCT, BPRAISONAME,
     c BPRAISOVALUE, LEFTLIST, CARRYCOUNTER, BPXSECT, BPRODS,
     c PREVIOUSNAME, FINALDOWNTIME, LEFTVAL, BPRA_INSERTED, CLADTOT,
     c CLADDESNUM, CLADDESNAME, BPRCLAD, CRCLAD, APSRCLAD,
     c CLTEMP, BPMIXNUM, BPMIX, BPMIXID,
     c BPNUMISOS, BPISOID, BPISOWTPCT, UCSPACERFRAC,
     c SPACERMAT, STEPCONTROL, VARBLETDOWN, VARSTEPNUM,
     c VARPOWER, BPRFM, BPFMNUMISOS, BPFISOID,
     c ABOVEBPNUM, APSRFM, BPRFR, BPFISOWTPCT,
     c APSRFR, ABOVEBP, APSRFOLLOWMIX, CT1START, CT2GOVALUE,
     c APSRINSOLD, RTYPE, MODREFDEN, CRMIXDEN,PICKUPFLAG,FTNDES,
     C GDROD,STOPFLAG,FIRSTMODEN)
*
      INTEGER*4 RELATIVE_STPT_NUM, CT1, CT2, CT3, AXNUM,
     c NUMSTPT1, NUMSTPT2, NUMSTPT3, CYCPOS(10), AXBLANK(50),
     c BPDESID(10), BPRA_DESCRIPTION_ID, CT4, CT5, CRINS(10,20,23,50),
     c CR_MIXTURE_ID, CR_DESCRIPTION, CRDES(10,20,23,50), CRMIXNUM,
     c CRMIXID(25), RELATIVE_CR_MIX_ID, CRNUMISOS(25),
     c CRISOID(25,20), APSRINS(10,20,23,50), APSR_MIXTURE_ID,
     c APSR_DESCRIPTION, APSRMIXNUM, APSRMIXID(25),
     c RELATIVE_APSR_MIX_ID, APSRNUMISOS(25), APSRISOID(25,10),
     c ISN, IIM, ICM, IUS, PLEVEL, BPZONE(10), BPMA(15,10),
     c CRZONE(10), CRMA(15,10), LMC(15,10), APSRZONE(10),
     c APSRMA(15,10), LMD(15,10), BPTN(10), BPBN(10), STPTS(10),
     c APSRDES(10,20,23,50), LUZONE, LMB(15), CARRYCOUNTER,
     c FUELISOTOPENUMBER, BPRODS(10), PNMCT1, PNMCT2, PNUMSTPT1,
     c PNUMSTPT2, PNUMSTPT3, NUMSTPT4, NUMSTPT5, NUMSTPT6,
     c PNUMSTPT4, PNUMSTPT5, PNUMSTPT6, CLADTOT, CLADDESNUM(10),
     c BPRCLAD(10), CRCLAD(10), APSRCLAD(10), BPRCLNUM, CRCLNUM,
     c APSRCLNUM, BPMIXNUM, BPMIX(10), BPMIXID(10), BPNUMISOS(10),
     c BPISOID(10,20), VARSTEPNUM(10,20), BPRFM(15,10),
     c BPFMNUMISOS(25), BPFISOID(25,10), ABOVEBPNUM(10),GDROD,
     c APSRFM(15,10), APSRFOLLOWMIX(10,20,23,50),
     c FOLNODKEEP, FOLSTEPKEEP, APSRFOLNUM, APSRINSOLD(10,20,23,50),
     c APSRFOLLOWDATA(10,20,23,50), CT1START, CT2GOVALUE
*
      REAL STPTDAT(10,20,3), ENR, AXBLANKRICH, OXYGMS, GRAMS(50),
     c FVOL, PI, NODES(50,2), RODS, FDEN,
     c RICH, FTFINAL(50,20),
     c MODDENFINAL(50,20), MODTEMPFINAL(50,20), BLETDOWN(10,20,25),
     c BPWTPCT(10), BPDEN(10), ALFRAC, OFRAC, CRDEN(10),
     c CRISOWTPCT(25,20), APSRDEN(10), APSRISOWTPCT(25,10),
     c PITCH, FOD, COD, CID, SZF, EPS, PTC, MESH, BPRA(15,10),
     c CRRA(15,10), LRC(15,10), APSRRA(15,10), LRD(15,10),
     c DOWNTIME, BORON_FRACTION, POWER(50,20), CYCDOWN(10), LRB(15),
     c MASSTOTAL, FUELISOWTPCT(1000), BPRAISOVALUE(10), BPXSECT(10),
     c BPVOL, FINALDOWNTIME, LEFTVAL(1000), CLTEMP,
     c BPISOWTPCT(10,20), UCSPACERFRAC, BORATEDMODVF,
     c BORONVF, UCMODREGIONDEN, B4CMASS, ALMASS, OMASS, CMASS,
     c NEWBPMASSTOTAL, NEWBPDEN, ALWTPCT, OWTPCT, CWTPCT, B10WTPCT,
     c B11WTPCT, VARBLETDOWN(10,20,25,25), VARPOWER(10,20,25,50),
     c BPRFR(15,10), BPFISOWTPCT(25,10), APSRFR(15,10), MODREFDEN,
     c CRMIXDEN(25),FTNDES(50,2,20),PRODMASS,
     C NEWVALWTPCT(15),NEWDEN,FIRSTMODEN(20,20),
     C FMASS1,FMASS2,FMASS3,FMASS4,FUELMASS,GDMASS,NONZERO,GDROMASS
```

```
*
      CHARACTER CHNODE*2, CHID*2, PREFIX*3, CHSTPT1*1, CHSTPT2*1,
     c CHSTPT3*1, NM*31, CYCLEID(10)*2, REACT*23, LIB*15,
     c AXBLANKET*1, FUELCLAD*10, BPRFLAG*1, CRSTAT*6, APSRSTAT*6,
     c FLAG2*7, IRRAD_STEPS*2, PLEVELCH*2, BPZONECH*2, CRZONECH*2,
     c APSRZONECH*2, LUZONECH*2, FUELISONAME(1000)*5, BPRAISONAME(12)*6,
     c LEFTLIST(1000)*6, PREVIOUSNAME*25, PCHSTPT1*1, PCHSTPT2*1,
     c PCHSTPT3*1, ASSYPOSITION*2, CHSTPT4*1, CHSTPT5*1, CHSTPT6*1,
     c PCHSTPT4*1, PCHSTPT5*1, PCHSTPT6*1, PCHID*2, CLADDESNAME(10)*7,
     c SPACERMAT*7, STEPCONTROL*1, ABOVEBP(10)*5, RTYPE*3,PICKUPFLAG*1,
     C STOPFLAG*1,CTIME*8,CDATE*9,VERS*9
*
      LOGICAL BPRA_INSERTED, CR_INSERTED, CRCOMPFLAG, APSR_INSERTED,
     c APSRCOMPFLAG, BPRA_FOLLOW, APSRBOTFLAG, FOLLOWIN
*
      PI=3.14159265359
*  Determination of the input deck filename
         CALL ZEROS(CT3,CHNODE)
         CALL ZEROS(CYCPOS(CT1),CHID)
         IF ((CT2-1).EQ.0) THEN
            PNMCT1=CT1-1
            PNMCT2=STPTS(PNMCT1)
         ELSE
            PNMCT1=CT1
            PNMCT2=CT2-1
         ENDIF
         CALL ZEROS(CYCPOS(PNMCT1),PCHID)
*  Determine new filename
         NUMSTPT1=INT(STPTDAT(CT1,CT2,1)/100.0)
         CHSTPT1=CHAR(NUMSTPT1+48)
         NUMSTPT2=INT((STPTDAT(CT1,CT2,1)-(NUMSTPT1*100))/10.0)
         CHSTPT2=CHAR(NUMSTPT2+48)
         NUMSTPT3=INT((STPTDAT(CT1,CT2,1)-(NUMSTPT1*100)-
     c   (NUMSTPT2*10)))
         CHSTPT3=CHAR(NUMSTPT3+48)
         IF (CT2.LT.STPTS(CT1)) THEN
            NUMSTPT4=INT(STPTDAT(CT1,(CT2+1),1)/100.0)
            CHSTPT4=CHAR(NUMSTPT4+48)
            NUMSTPT5=INT((STPTDAT(CT1,(CT2+1),1)-(NUMSTPT4*100))/10.0)
            CHSTPT5=CHAR(NUMSTPT5+48)
            NUMSTPT6=INT((STPTDAT(CT1,(CT2+1),1)-(NUMSTPT4*100)-
     c      (NUMSTPT5*10)))
            CHSTPT6=CHAR(NUMSTPT6+48)
         ELSEIF (CT2.EQ.STPTS(CT1)) THEN
            NUMSTPT4=INT(STPTDAT((CT1+1),1,1)/100.0)
            CHSTPT4=CHAR(NUMSTPT4+48)
            NUMSTPT5=INT((STPTDAT((CT1+1),1,1)-(NUMSTPT4*100))/10.0)
            CHSTPT5=CHAR(NUMSTPT5+48)
            NUMSTPT6=INT((STPTDAT((CT1+1),1,1)-(NUMSTPT4*100)-
     c      (NUMSTPT5*10)))
            CHSTPT6=CHAR(NUMSTPT6+48)
         ENDIF
         NM(1:3)=PREFIX
         NM(4:4)='A'
         NM(5:6)=CHID
         NM(7:7)='N'
         NM(8:9)=CHNODE
         NM(10:11)='DC'
         NM(12:13)=CYCLEID(CT1)
         NM(14:14)='T'
         NM(15:15)=CHSTPT1
         NM(16:16)=CHSTPT2
         NM(17:17)=CHSTPT3
         NM(18:19)='AC'
```

```
      IF (CT2.EQ.STPTS(CT1)) THEN
         NM(20:21)=CYCLEID(CT1+1)
      ELSE
         NM(20:21)=CYCLEID(CT1)
      ENDIF
      NM(22:22)='T'
      NM(23:23)=CHSTPT4
      NM(24:24)=CHSTPT5
      NM(25:25)=CHSTPT6
      NM(26:31)='.input'
*  Determine previous filename
      PNUMSTPT1=INT(STPTDAT(PNMCT1,PNMCT2,1)/100.0)
      PCHSTPT1=CHAR(PNUMSTPT1+48)
      PNUMSTPT2=INT((STPTDAT(PNMCT1,PNMCT2,1)-
   c  (PNUMSTPT1*100))/10.0)
      PCHSTPT2=CHAR(PNUMSTPT2+48)
      PNUMSTPT3=INT((STPTDAT(PNMCT1,PNMCT2,1)-(PNUMSTPT1*100)-
   c  (PNUMSTPT2*10)))
      PCHSTPT3=CHAR(PNUMSTPT3+48)
      IF (PNMCT2.LT.STPTS(PNMCT1)) THEN
         PNUMSTPT4=INT(STPTDAT(PNMCT1,(PNMCT2+1),1)/100.0)
         PCHSTPT4=CHAR(PNUMSTPT4+48)
         PNUMSTPT5=INT((STPTDAT(PNMCT1,(PNMCT2+1),1)-
   c     (PNUMSTPT4*100))/10.0)
         PCHSTPT5=CHAR(PNUMSTPT5+48)
         PNUMSTPT6=INT((STPTDAT(PNMCT1,(PNMCT2+1),1)-
   c     (PNUMSTPT4*100)-(PNUMSTPT5*10)))
         PCHSTPT6=CHAR(PNUMSTPT6+48)
      ELSEIF (PNMCT2.EQ.STPTS(PNMCT1)) THEN
         PNUMSTPT4=INT(STPTDAT((PNMCT1+1),1,1)/100.0)
         PCHSTPT4=CHAR(PNUMSTPT4+48)
         PNUMSTPT5=INT((STPTDAT((PNMCT1+1),1,1)-
   c     (PNUMSTPT4*100))/10.0)
         PCHSTPT5=CHAR(PNUMSTPT5+48)
         PNUMSTPT6=INT((STPTDAT((PNMCT1+1),1,1)-(PNUMSTPT4*100)-
   c     (PNUMSTPT5*10)))
         PCHSTPT6=CHAR(PNUMSTPT6+48)
      ENDIF
      PREVIOUSNAME(1:3)=PREFIX
      PREVIOUSNAME(4:4)='A'
      PREVIOUSNAME(5:6)=PCHID
      PREVIOUSNAME(7:7)='N'
      PREVIOUSNAME(8:9)=CHNODE
      PREVIOUSNAME(10:11)='DC'
      IF (CT2.EQ.1) THEN
         PREVIOUSNAME(12:13)=CYCLEID(CT1-1)
      ELSE
         PREVIOUSNAME(12:13)=CYCLEID(CT1)
      ENDIF
      PREVIOUSNAME(14:14)='T'
      PREVIOUSNAME(15:15)=PCHSTPT1
      PREVIOUSNAME(16:16)=PCHSTPT2
      PREVIOUSNAME(17:17)=PCHSTPT3
      PREVIOUSNAME(18:19)='AC'
      PREVIOUSNAME(20:21)=CYCLEID(CT1)
      PREVIOUSNAME(22:22)='T'
      PREVIOUSNAME(23:23)=PCHSTPT4
      PREVIOUSNAME(24:24)=PCHSTPT5
      PREVIOUSNAME(25:25)=PCHSTPT6
*  Open and rewind the input deck file
      OPEN(UNIT=100, FILE=NM, STATUS='UNKNOWN')
      REWIND(UNIT=100)
*  Write first section of input deck
      WRITE (100,10)
```

```
   10     FORMAT ('=sas2h',T11,'parm=skipshipdata')
          IF (CT2.LT.STPTS(CT1)) THEN
             WRITE (100,20) REACT, CHID, CHNODE,
     c       NM(12:13), STPTDAT(CT1,CT2,1), NM(20:21),
     c       STPTDAT(CT1,(CT2+1),1)
   20        FORMAT (A23,1X,'Assy-',A2,
     c       ', Node-',A2,1X,
     c       '{Cyc-',A2,', 'F5.1,' to Cyc-',
     c       A2,', ',F5.1,' EFPD}')
          ELSEIF (CT2.EQ.STPTS(CT1)) THEN
             WRITE (100,25) REACT, CHID, CHNODE,
     c       NM(12:13), STPTDAT(CT1,CT2,1), NM(20:21),
     c       STPTDAT((CT1+1),1,1)
   25        FORMAT (A23,1X,'Assy-',A2,
     c       ', Node-',A2,1X,
     c       '{Cyc-',A2,', 'F5.1,' to Cyc-',
     c       A2,', ',F5.1,' EFPD}')
          ENDIF
          WRITE (100,30) LIB
   30     FORMAT (A15,1X,'latticecell')
          WRITE (100,40)
   40     FORMAT ('''')
          CALL DATE(CDATE)
          CALL TIME(CTIME)
          VERS='CRAFT V4C'
          WRITE (100,48)
   48     FORMAT ('''',T5,'******************************')
          WRITE (100,49)
   49     FORMAT ('''',T5,'this file created by:')
          WRITE (100,50)VERS,CDATE,CTIME
   50     FORMAT ('''',T5,A9,2X,A9,2X,A8)
          WRITE (100,51)
   51     FORMAT ('''',T5,'******************************')
          WRITE (100,52)
   52     FORMAT ('''')
          WRITE (100,60)
   60     FORMAT ('''   fuel density based on mass of uranium per',
     c    ' node')
          WRITE (100,70)
   70     FORMAT ('''')
*  Write second section of input deck (material specifications)
          WRITE (100,80)
   80     FORMAT ('''',5X,'material specification input')
          WRITE (100,90)
   90     FORMAT ('''')
*  Calculate initital fuel parameters depending upon whether or not the
*  node represents axial blanket fuel
          IF ((AXBLANKET.EQ.'Y').AND.(AXBLANK(CT3).EQ.1)) THEN
             ENR=AXBLANKRICH
          ELSE
             ENR=RICH
          ENDIF
          OXYGMS=(GRAMS(CT3)*2*15.994915)/(((ENR/100)*235.043915)+
     c    (((0.007731*((ENR)**1.0837))/100)*234.040904)+
     c    (((0.0046*ENR)/100)*236.045637)+(((100-(0.007731*
     c    (ENR**1.0837))-(ENR)-(0.0046*ENR))/100)*238.05077))
*  Determine if the burnable poison charge isotopics should be retrieved
          BPRA_INSERTED=.FALSE.
          IF ((BPRFLAG.EQ.'Y').AND.(BPDESID(CT1).NE.0).AND.
     c       (CT3.GE.BPTN(CT1)).AND.(CT3.LE.BPBN(CT1))) THEN
            BPRA_INSERTED=.TRUE.
          ENDIF
*  Call subroutine to retrieve charge for fuel and bp isotopics
          IF((RTYPE.EQ.'BWR').AND.(GDROD.NE.0))THEN
```

```
          BPRA_INSERTED=.TRUE.
        ENDIF
C   always looks for BPRA-gadolinia isostopics-no restart
        CALL RETRIEVER (OXYGMS, MASSTOTAL,
     c  FUELISONAME, FUELISOWTPCT, BPRAISONAME,
     c  BPRAISOVALUE, LEFTLIST, CARRYCOUNTER,
     c  PREVIOUSNAME, LEFTVAL, NM, BPRA_INSERTED)
*   Calculate the nodal fuel volume, fuel density, and oxygen wt%
c   modification here to calculate proper density using CLADID rather than
c   FOD because gap is removed from pathA model
c           FVOL=(PI/4)*(FOD**2)*(NODES(CT3,2))*(RODS)
        FVOL=(PI/4)*(CID**2)*(NODES(CT3,2))*(RODS)
        FDEN=MASSTOTAL/FVOL
        OXYWTPCT=(OXYGMS/MASSTOTAL)*100.0
        FUELISOTOPENUMBER=CARRYCOUNTER+1
*   Write fuel composition input description
        IF (FDEN.LT.(10.0)) THEN
          WRITE (100,100) FDEN, FUELISOTOPENUMBER, OXYWTPCT
  100     FORMAT ('arbm-fuel',1X,G10.3,1X,I3,1X,'0 0 0',1X,
     c    '8016',1X,G10.3)
        ELSE
          WRITE (100,110) FDEN, FUELISOTOPENUMBER, OXYWTPCT
  110     FORMAT ('arbm-fuel',1X,G10.3,1X,I3,1X,'0 0 0',1X,
     c    '8016',1X,G10.3)
        ENDIF
        DO 130 CT4=1,CARRYCOUNTER
          IF (MOD(CT4,3).EQ.0) THEN
            WRITE (100,*)
          ENDIF
          WRITE (100,120) FUELISONAME(CT4), FUELISOWTPCT(CT4)
  120     FORMAT (5X,A5,1X,G10.3,1X,$)
  130   CONTINUE
        WRITE (100,*)
        WRITE (100,140) FTFINAL(CT3,RELATIVE_STPT_NUM)
  140   FORMAT (5X,'1',3X,'1.0',3X,F6.1,'   end')
*   Write cladding material specifications
*   Additional cladding material specifications may be added to the
*   following IF statement as required
        IF ((FUELCLAD.EQ.'ZIRC-4     ').OR.
     c  (FUELCLAD.EQ.'ZIRCALLOY4')) THEN
          WRITE (100,532)
  532     FORMAT ('arbm-zirc4 6.56 5 0 0 0 8016 0.12 24000',
     c    ' 0.10 26000 0.20 50000 1.40')
          WRITE (100,535) CLTEMP
  535     FORMAT (T12,'40000 98.18 2 1.0 ',F5.1,' end')
        ELSEIF (FUELCLAD.EQ.'SS304      ') THEN
          WRITE (100,537)
  537     FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055',
     c    ' 2.0 26304 69.5 28304 9.5')
          WRITE (100,540) CLTEMP
  540     FORMAT (T12,'2 1.0 ',F5.1,' end')
        ELSEIF (FUELCLAD.EQ.'SS304S     ') THEN
          WRITE (100,542)
  542     FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055',
     c    ' 2.0 26000 69.5 28000 9.5')
          WRITE (100,545) CLTEMP
  545     FORMAT (T13,'2 1.0 ',F5.1,' end')
        ELSEIF (FUELCLAD.EQ.'SS316      ') THEN
          WRITE (100,547)
  547     FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000',
     c    ' 1.0 24304 17.0 25055 2.0')
          WRITE (100,550)
  550     FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
          WRITE (100,552) CLTEMP
```

```
  552       FORMAT (T12,'2 1.0 ',F5.1,' end')
         ELSEIF (FUELCLAD.EQ.'SS316S    ') THEN
            WRITE (100,555)
  555       FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000',
     c      ' 1.0 24000 17.0 25055 2.0')
            WRITE (100,557)
  557       FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')
            WRITE (100,559) CLTEMP
  559       FORMAT (T13,'2 1.0 ',F5.1,' end')
         ENDIF
*  Write moderator material specifications
         BORATEDMODVF=1.0-UCSPACERFRAC
         IF (RTYPE.EQ.'PWR') THEN
            IF (STEPCONTROL.EQ.'N') THEN
               BORONVF=BLETDOWN(CT1,CT2,3)*(1E-6)*BORATEDMODVF
            ELSEIF (STEPCONTROL.EQ.'Y') THEN
               BORONVF=VARBLETDOWN(CT1,CT2,1,2)*(1E-6)*BORATEDMODVF
            ENDIF
         ENDIF
         WRITE (100,560)
  560    FORMAT ('''')
         IF ((SPACERMAT.EQ.'ZIRC-4 ').AND.
     c   (UCSPACERFRAC.GT.(0.0))) THEN
            WRITE (100,561)
  561       FORMAT ('''     material composition of moderator',
     c      ' within unit cell')
            WRITE (100,562)
  562       FORMAT ('''     with smeared zirc-4 spacer grids')
            IF (RTYPE.EQ.'PWR') THEN
               UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
     c         BORATEDMODVF)+(6.56*UCSPACERFRAC)
            ELSEIF (RTYPE.EQ.'BWR') THEN
C                UCMODREGIONDEN=(MODREFDEN*
C     c          BORATEDMODVF)+(6.56*UCSPACERFRAC)
               UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
     c         BORATEDMODVF)+(6.56*UCSPACERFRAC)
            ENDIF
            IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
               WRITE (100,563) UCMODREGIONDEN, BORATEDMODVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  563          FORMAT ('h2o    3    den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
            ELSE
               WRITE (100,564) UCMODREGIONDEN, BORATEDMODVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  564          FORMAT ('h2o    3    den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
            ENDIF
            IF (RTYPE.EQ.'PWR') THEN
               WRITE (100,565) UCMODREGIONDEN, BORONVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  565          FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
     c         1X,F6.5,1X,F7.1,1X,'end')
            ENDIF
            WRITE (100,566) UCMODREGIONDEN
  566       FORMAT ('arbm-spacer',3X,F6.4,1X,'5 0 0 0 8016 0.12',
     c      ' 24000 0.10 26000 0.25')
            WRITE (100,567) UCSPACERFRAC,
     c      MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  567       FORMAT (T17'50000 1.40 40000 98.18 3',1X,F6.5,1X,
     c      F7.1,1X,'end')
         ELSEIF ((SPACERMAT.EQ.'INCONEL').AND.
     c   (UCSPACERFRAC.GT.(0.0))) THEN
            WRITE (100,568)
  568       FORMAT ('''     material composition of moderator',
     c      ' within unit cell')
```

```
            WRITE (100,569)
c  569       FORMAT ('''     with smeared inconel spacer grids')
  569       FORMAT ('''     without smeared inconel spacer grids')
            IF (RTYPE.EQ.'PWR') THEN
               UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
     c         BORATEDMODVF)+(8.3*UCSPACERFRAC)
            ELSEIF (RTYPE.EQ.'BWR') THEN
C               UCMODREGIONDEN=(MODREFDEN*
C    c          BORATEDMODVF)+(8.3*UCSPACERFRAC)
               UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
     c         BORATEDMODVF)+(8.3*UCSPACERFRAC)
            ENDIF
            IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
               WRITE (100,570) UCMODREGIONDEN, BORATEDMODVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  570          FORMAT ('h2o    3    den=',F5.4,3X,G14.8,3X,F7.1,3X,'end')
            ELSE
               WRITE (100,571) UCMODREGIONDEN, BORATEDMODVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  571          FORMAT ('h2o    3    den=',F6.4,3X,G14.8,3X,F7.1,3X,'end')
            ENDIF
            IF (RTYPE.EQ.'PWR') THEN
               WRITE (100,572) UCMODREGIONDEN, BORONVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  572          FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
     c         1X,F6.5,1X,F7.1,1X,'end')
            ENDIF
            WRITE (100,573) UCMODREGIONDEN
  573       FORMAT ('arbm-spacer',3X,F6.4,1X,'5 0 0 0 14000 2.5',
     c      ' 22000 2.5 24000 15.0')
            WRITE (100,574) UCSPACERFRAC,
     c      MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  574       FORMAT (T17'26000 7.0 28000 73.0 3',1X,G14.6,1X,
     c      F7.1,1X,'end')
         ELSEIF ((SPACERMAT.EQ.'SS316  ').AND.
     c   (UCSPACERFRAC.GT.(0.0))) THEN
            WRITE (100,575)
  575       FORMAT ('''     material composition of moderator',
     c      ' within unit cell')
            WRITE (100,576)
  576       FORMAT ('''     with smeared ss316 spacer grids')
            IF (RTYPE.EQ.'PWR') THEN
               UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
     c         BORATEDMODVF)+(7.75*UCSPACERFRAC)
            ELSEIF (RTYPE.EQ.'BWR') THEN
C               UCMODREGIONDEN=(MODREFDEN*
C    c          BORATEDMODVF)+(7.75*UCSPACERFRAC)
               UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
     c         BORATEDMODVF)+(7.75*UCSPACERFRAC)
            ENDIF
            IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
               WRITE (100,577) UCMODREGIONDEN, BORATEDMODVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  577          FORMAT ('h2o    3    den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
            ELSE
               WRITE (100,578) UCMODREGIONDEN, BORATEDMODVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  578          FORMAT ('h2o    3    den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
            ENDIF
            IF (RTYPE.EQ.'PWR') THEN
               WRITE (100,579) UCMODREGIONDEN, BORONVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  579          FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
     c         1X,F6.5,1X,F7.1,1X,'end')
```

```
            ENDIF
            WRITE (100,580) UCMODREGIONDEN
  580       FORMAT ('arbm-spacer',3X,F6.4,1X,'7 0 0 0 6012 0.08',
     c      ' 14000 1.0 24304 17.0 25055 2.0')
            WRITE (100,581) UCSPACERFRAC,
     c      MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  581       FORMAT (T5'26304 65.42 28304 12.0 42000 2.5 3',1X,F6.5,1X,
     c      F7.1,1X,'end')
         ELSEIF ((SPACERMAT.EQ.'SS316S ').AND.
     c   (UCSPACERFRAC.GT.(0.0))) THEN
            WRITE (100,582)
  582       FORMAT ('''     material composition of moderator',
     c      ' within unit cell')
            WRITE (100,583)
  583       FORMAT ('''     with smeared ss316s spacer grids')
            IF (RTYPE.EQ.'PWR') THEN
               UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
     c         BORATEDMODVF)+(7.75*UCSPACERFRAC)
            ELSEIF (RTYPE.EQ.'BWR') THEN
C                UCMODREGIONDEN=(MODREFDEN*
C     c          BORATEDMODVF)+(7.75*UCSPACERFRAC)
               UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
     c         BORATEDMODVF)+(7.75*UCSPACERFRAC)
            ENDIF
            IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
               WRITE (100,584) UCMODREGIONDEN, BORATEDMODVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  584          FORMAT ('h2o    3    den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
            ELSE
               WRITE (100,585) UCMODREGIONDEN, BORATEDMODVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  585          FORMAT ('h2o    3    den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
            ENDIF
            IF (RTYPE.EQ.'PWR') THEN
               WRITE (100,586) UCMODREGIONDEN, BORONVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  586          FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
     c         1X,F6.5,1X,F7.1,1X,'end')
            ENDIF
            WRITE (100,587) UCMODREGIONDEN
  587       FORMAT ('arbm-spacer',3X,F6.4,1X,'7 0 0 0 6012 0.08',
     c      ' 14000 1.0 24000 17.0 25055 2.0')
            WRITE (100,588) UCSPACERFRAC,
     c      MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  588       FORMAT (T5'26000 65.42 28000 12.0 42000 2.5 3',1X,F6.5,1X,
     c      F7.1,1X,'end')
         ELSEIF ((SPACERMAT.EQ.'SS304  ').AND.
     c   (UCSPACERFRAC.GT.(0.0))) THEN
            WRITE (100,589)
  589       FORMAT ('''     material composition of moderator',
     c      ' within unit cell')
            WRITE (100,590)
  590       FORMAT ('''     with smeared ss304 spacer grids')
            IF (RTYPE.EQ.'PWR') THEN
               UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
     c         BORATEDMODVF)+(7.92*UCSPACERFRAC)
            ELSEIF (RTYPE.EQ.'BWR') THEN
C                UCMODREGIONDEN=(MODREFDEN*
C     c          BORATEDMODVF)+(7.92*UCSPACERFRAC)
               UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
     c         BORATEDMODVF)+(7.92*UCSPACERFRAC)
            ENDIF
            IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
               WRITE (100,591) UCMODREGIONDEN, BORATEDMODVF,
```

```
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  591          FORMAT ('h2o    3    den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
            ELSE
               WRITE (100,592) UCMODREGIONDEN, BORATEDMODVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  592          FORMAT ('h2o    3    den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
            ENDIF
            IF (RTYPE.EQ.'PWR') THEN
               WRITE (100,593) UCMODREGIONDEN, BORONVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  593          FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
     c         1X,F6.5,1X,F7.1,1X,'end')
            ENDIF
            WRITE (100,594) UCMODREGIONDEN
  594       FORMAT ('arbm-spacer',3X,F6.4,1X,'4 0 0 0 24304 19.0',
     c      ' 25055 2.0 26304 69.5 28304 9.5')
            WRITE (100,595) UCSPACERFRAC,
     c      MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  595       FORMAT (T15'3',1X,F6.5,1X,F7.1,1X,'end')
         ELSEIF ((SPACERMAT.EQ.'SS304S ').AND.
     c   (UCSPACERFRAC.GT.(0.0))) THEN
            WRITE (100,596)
  596       FORMAT ('''      material composition of moderator',
     c      ' within unit cell')
            WRITE (100,597)
  597       FORMAT ('''      with smeared ss304s spacer grids')
            IF (RTYPE.EQ.'PWR') THEN
               UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
     c         BORATEDMODVF)+(7.92*UCSPACERFRAC)
            ELSEIF (RTYPE.EQ.'BWR') THEN
C                UCMODREGIONDEN=(MODREFDEN*
C     c          BORATEDMODVF)+(7.92*UCSPACERFRAC)
               UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
     c         BORATEDMODVF)+(7.92*UCSPACERFRAC)
            ENDIF
            IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
               WRITE (100,598) UCMODREGIONDEN, BORATEDMODVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  598          FORMAT ('h2o    3    den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
            ELSE
               WRITE (100,599) UCMODREGIONDEN, BORATEDMODVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  599          FORMAT ('h2o    3    den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
            ENDIF
            IF (RTYPE.EQ.'PWR') THEN
               WRITE (100,600) UCMODREGIONDEN, BORONVF,
     c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  600          FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
     c         1X,F6.5,1X,F7.1,1X,'end')
            ENDIF
            WRITE (100,601) UCMODREGIONDEN
  601       FORMAT ('arbm-spacer',3X,F6.4,1X,'4 0 0 0 24000 19.0',
     c      ' 25055 2.0 26000 69.5 28000 9.5')
            WRITE (100,602) UCSPACERFRAC,
     c      MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  602       FORMAT (T15'3',1X,F6.5,1X,F7.1,1X,'end')
         ELSEIF (UCSPACERFRAC.EQ.(0.0)) THEN
            WRITE (100,603)
  603       FORMAT ('''      material composition of moderator',
     c      ' within unit cell')
            WRITE (100,604)
  604       FORMAT ('''      with no smeared spacer grids')
            IF (RTYPE.EQ.'PWR') THEN
               UCMODREGIONDEN=(MODDENFINAL(CT3,RELATIVE_STPT_NUM)*
```

```
     c        BORATEDMODVF)
           ELSEIF (RTYPE.EQ.'BWR') THEN
C               UCMODREGIONDEN=(MODREFDEN*
C     c         BORATEDMODVF)
              UCMODREGIONDEN=(FIRSTMODEN(CT1,CT2)*
     c        BORATEDMODVF)
           ENDIF
           IF (MODDENFINAL(CT3,RELATIVE_STPT_NUM).LT.(1.0)) THEN
              WRITE (100,605) UCMODREGIONDEN, BORATEDMODVF,
     c        MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  605         FORMAT ('h2o    3    den=',F5.4,3X,F6.5,3X,F7.1,3X,'end')
           ELSE
              WRITE (100,606) UCMODREGIONDEN, BORATEDMODVF,
     c        MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  606         FORMAT ('h2o    3    den=',F6.4,3X,F6.5,3X,F7.1,3X,'end')
           ENDIF
           IF (RTYPE.EQ.'PWR') THEN
              WRITE (100,607) UCMODREGIONDEN, BORONVF,
     c        MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  607         FORMAT ('arbm-bormod',3X,F6.4,1X,'1 0 0 0 5000 100 3',
     c        1X,F6.5,1X,F7.1,1X,'end')
           ENDIF
        ENDIF
        WRITE (100,608)
  608   FORMAT ('''')
*  Write BPRA material specifications
*  BPR follow specifications
        BPRA_FOLLOW=.FALSE.
        IF ((BPRFLAG.EQ.'Y').AND.(BPDESID(CT1).NE.0).AND.
     c     (CT3.LT.BPTN(CT1))) THEN
           BPRA_FOLLOW=.TRUE.
           BPRA_DESCRIPTION_ID=BPDESID(CT1)
           WRITE(100,610)
  610      FORMAT('''')
           WRITE(100,612)
  612      FORMAT('''',5X,'BPR above the BP absorber region')
           WRITE(100,614)
  614      FORMAT('''')
           IF ((BPRCLAD(BPDESID(CT1)).NE.0).AND.
     c     (BPRCLAD(BPDESID(CT1)).NE.2)) THEN
             DO 616 CT5=1,10
              IF (BPRCLAD(BPDESID(CT1)).EQ.CLADDESNUM(CT5)) THEN
               BPRCLNUM=CT5
               EXIT
              ENDIF
  616        CONTINUE
             IF (CLADDESNAME(BPRCLNUM).EQ.'SS304  ') THEN
              WRITE (100,618)
  618         FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055',
     c        ' 2.0 26304 69.5 28304 9.5')
              WRITE (100,620) CLADDESNUM(BPRCLNUM), CLTEMP
  620         FORMAT (T12,I2,' 1.0 ',F5.1,' end')
             ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'SS304S ') THEN
              WRITE (100,622)
  622         FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055',
     c        ' 2.0 26000 69.5 28000 9.5')
              WRITE (100,624) CLADDESNUM(BPRCLNUM), CLTEMP
  624         FORMAT (T13,I2,' 1.0 ',F5.1,' end')
             ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'SS316  ') THEN
              WRITE (100,626)
  626         FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000',
     c        ' 1.0 24304 17.0 25055 2.0')
              WRITE (100,628)
  628         FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
```

```
                  WRITE (100,630) CLADDESNUM(BPRCLNUM), CLTEMP
 630              FORMAT (T12,I2,' 1.0 ',F5.1,' end')
                ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'SS316S ') THEN
                  WRITE (100,632)
 632              FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000',
     c            ' 1.0 24000 17.0 25055 2.0')
                  WRITE (100,633)
 633              FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')
                  WRITE (100,634) CLADDESNUM(BPRCLNUM), CLTEMP
 634              FORMAT (T13,I2,' 1.0 ',F5.1,' end')
                ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'INCONEL') THEN
                  WRITE (100,635)
 635              FORMAT ('arbm-inconel 8.3 5 0 0 0 14000 2.5',
     c            ' 22000 2.5 24000 15.0')
                  WRITE (100,636)
 636              FORMAT (T13,'26000 7.0 28000 73.0')
                  WRITE (100,637) CLADDESNUM(BPRCLNUM), CLTEMP
 637              FORMAT (T13,I2,' 1.0 ',F5.1,' end')
                ENDIF
              ENDIF
              IF (ABOVEBP(BPDESID(CT1)).EQ.'AL2O3') THEN
                  ALFRAC=(((BPDEN(BPDESID(CT1)))*2.0*26.981539)/
     c            (101.9631))/BPDEN(BPDESID(CT1))
                  OFRAC=1.0-ALFRAC
                  IF (BPDEN(BPDESID(CT1)).LT.(1.0)) THEN
                     WRITE (100,638) ABOVEBPNUM(BPDESID(CT1)),
     c               BPDEN(BPDESID(CT1)), ALFRAC,
     c               MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
 638                 FORMAT ('al',3X,I3,3X,'den=',F4.3,1X,F7.5,
     c               1X,F7.1,1X,'end')
                     WRITE (100,640) ABOVEBPNUM(BPDESID(CT1)),
     c               BPDEN(BPDESID(CT1)), OFRAC,
     c               MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
 640                 FORMAT ('o',3X,I3,3X,'den=',F4.3,1X,F7.5,
     c               1X,F7.1,1X,'end')
                  ELSE
                     WRITE (100,642) ABOVEBPNUM(BPDESID(CT1)),
     c               BPDEN(BPDESID(CT1)), ALFRAC,
     c               MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
 642                 FORMAT ('al',3X,I3,3X,'den=',F5.3,1X,F7.5,
     c               1X,F7.1,1X,'end')
                     WRITE (100,644) ABOVEBPNUM(BPDESID(CT1)),
     c               BPDEN(BPDESID(CT1)), OFRAC,
     c               MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
 644                 FORMAT ('o',3X,I3,3X,'den=',F5.3,1X,F7.5,
     c               1X,F7.1,1X,'end')
                  ENDIF
              ELSE
                  WRITE (100,*) 'arbm-bp    ',
     c            BPDEN(BPRA_DESCRIPTION_ID),
     c            '   ', BPFMNUMISOS(BPRA_DESCRIPTION_ID),
     c            '   0    0    0'
                  DO 650 CT4=1,BPFMNUMISOS(BPRA_DESCRIPTION_ID)
                     WRITE (100,648)
     c               BPFISOID(BPRA_DESCRIPTION_ID,CT4),
     c               BPFISOWTPCT(BPRA_DESCRIPTION_ID,CT4)
 648                 FORMAT (10X,I6,3X,F10.5)
 650              CONTINUE
                  WRITE (100,*) '             ',
     c            ABOVEBPNUM(BPRA_DESCRIPTION_ID),
     c            ' 1.0 ',MODTEMPFINAL(CT3,RELATIVE_STPT_NUM),
     c            '    end'
              ENDIF
           ENDIF
```

```
*  Actual BPRA specifications
         BPRA_INSERTED=.FALSE.
         IF ((BPRFLAG.EQ.'Y').AND.(BPDESID(CT1).NE.0).AND.
    c       (CT3.GE.BPTN(CT1)).AND.(CT3.LE.BPBN(CT1))) THEN
            BPRA_INSERTED=.TRUE.
            BPRA_DESCRIPTION_ID=BPDESID(CT1)
            WRITE (100,685)
  685       FORMAT ('''')
            IF ((BPMIX(BPRA_DESCRIPTION_ID).EQ.0).OR.
    c       (BPMIX(BPRA_DESCRIPTION_ID).EQ.4)) THEN
               WRITE (100,690) BPWTPCT(BPDESID(CT1))
  690          FORMAT ('''',5X,'Al2O3-B4C burnable absorber pellet',1X,
    c          'specification (',F4.2,1X,'wt% b4c)')
            ELSE
               WRITE (100,695)
  695          FORMAT ('''',5X,'burnable absorber pellet ',
    c          'specification')
            ENDIF
            WRITE (100,700)
  700       FORMAT ('''')
*  Write B4C material specification
            IF ((BPRCLAD(BPDESID(CT1)).NE.0).AND.
    c       (BPRCLAD(BPDESID(CT1)).NE.2)) THEN
              DO 701 CT5=1,10
               IF (BPRCLAD(BPDESID(CT1)).EQ.CLADDESNUM(CT5)) THEN
                BPRCLNUM=CT5
                EXIT
               ENDIF
  701         CONTINUE
              IF (CLADDESNAME(BPRCLNUM).EQ.'SS304  ') THEN
               WRITE (100,702)
  702          FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055',
    c          ' 2.0 26304 69.5 28304 9.5')
               WRITE (100,703) CLADDESNUM(BPRCLNUM), CLTEMP
  703          FORMAT (T12,I2,' 1.0 ',F5.1,' end')
              ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'SS304S ') THEN
               WRITE (100,704)
  704          FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055',
    c          ' 2.0 26000 69.5 28000 9.5')
               WRITE (100,705) CLADDESNUM(BPRCLNUM), CLTEMP
  705          FORMAT (T13,I2,' 1.0 ',F5.1,' end')
              ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'SS316  ') THEN
               WRITE (100,706)
  706          FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000',
    c          ' 1.0 24304 17.0 25055 2.0')
               WRITE (100,707)
  707          FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
               WRITE (100,708) CLADDESNUM(BPRCLNUM), CLTEMP
  708          FORMAT (T12,I2,' 1.0 ',F5.1,' end')
              ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'SS316S ') THEN
               WRITE (100,709)
  709          FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000',
    c          ' 1.0 24000 17.0 25055 2.0')
               WRITE (100,710)
  710          FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')

               WRITE (100,711) CLADDESNUM(BPRCLNUM), CLTEMP
  711          FORMAT (T13,I2,' 1.0 ',F5.1,' end')
              ELSEIF (CLADDESNAME(BPRCLNUM).EQ.'INCONEL') THEN
               WRITE (100,712)
  712          FORMAT ('arbm-inconel 8.3 5 0 0 0 14000 2.5',
    c          ' 22000 2.5 24000 15.0')
               WRITE (100,713)
  713          FORMAT (T13,'26000 7.0 28000 73.0')
```

```
            WRITE (100,714) CLADDESNUM(BPRCLNUM), CLTEMP
 714        FORMAT (T13,I2,' 1.0 ',F5.1,' end')
          ENDIF
        ENDIF
*  Material specification if it is a BOC to statepoint 1 calculation
        IF (CT2.EQ.1) THEN
*  Material specification for AL2O3-B4C
          IF ((BPMIX(BPRA_DESCRIPTION_ID).EQ.0).OR.
     c    (BPMIX(BPRA_DESCRIPTION_ID).EQ.4)) THEN
            IF (BPWTPCT(BPDESID(CT1)).NE.(0.0)) THEN
              IF (BPDEN(BPDESID(CT1)).LT.(1.0)) THEN
                WRITE (100,715) BPDEN(BPDESID(CT1)),
     c          (BPWTPCT(BPDESID(CT1))/100.0),
     c          MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
 715            FORMAT ('b4c    4    den=',F4.3,1X,F7.5,1X,F7.1,1X,
     c          'end')
              ELSE
                WRITE (100,716) BPDEN(BPDESID(CT1)),
     c          (BPWTPCT(BPDESID(CT1))/100.0),
     c          MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
 716            FORMAT ('b4c    4    den=',F5.3,1X,F7.5,1X,F7.1,1X,
     c          'end')
              ENDIF
            ENDIF
*  Calculate aluminum and oxygen material specifications
            ALFRAC=(((((100.0-BPWTPCT(BPDESID(CT1)))/100)*
     c      BPDEN(BPDESID(CT1)))*2*26.981539)/(101.9631))/
     c      BPDEN(BPDESID(CT1))
            OFRAC=1-(BPWTPCT(BPDESID(CT1))/100.0)-ALFRAC
            WRITE (100,718) BPDEN(BPDESID(CT1)), ALFRAC,
     c      MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
 718        FORMAT ('al    4    den=',F5.3,1X,F7.5,1X,F7.1,1X,'end')
            WRITE (100,720) BPDEN(BPDESID(CT1)), OFRAC,
     c      MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
 720        FORMAT ('o     4    den=',F5.3,1X,F7.5,1X,F7.1,1X,'end')
          ELSE
*  Material specification for BP other than Al2O3-B4C
            DO 722 CT4=1,BPMIXNUM
              IF (BPMIXID(CT4).EQ.BPMIX(BPRA_DESCRIPTION_ID)) THEN
                RELATIVE_BP_MIX_ID=CT4
              ENDIF
 722        CONTINUE
            WRITE (100,*) 'arbm-bp    ', BPDEN(BPRA_DESCRIPTION_ID),
     c      '   ', BPNUMISOS(RELATIVE_BP_MIX_ID),
     c      '  0   0   0'
            DO 726 CT4=1,BPNUMISOS(RELATIVE_BP_MIX_ID)
              WRITE (100,724) BPISOID(RELATIVE_BP_MIX_ID,CT4),
     c        BPISOWTPCT(RELATIVE_BP_MIX_ID,CT4)
 724          FORMAT (10X,I6,3X,F10.5)
 726        CONTINUE
            WRITE (100,*) '            ', BPMIX(BPRA_DESCRIPTION_ID),
     c      ' 1.0 ',MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '   end'
          ENDIF
*  Material specification if it is not a BOC to statepoint 1 calculation
        ELSEIF(CT2.NE.1) THEN
          BPVOL=BPXSECT(BPRA_DESCRIPTION_ID)*
     c    BPRODS(BPRA_DESCRIPTION_ID)*NODES(CT3,2)
*  Material specification for AL2O3-B4C
          IF ((BPMIX(BPRA_DESCRIPTION_ID).EQ.0).OR.
     c    (BPMIX(BPRA_DESCRIPTION_ID).EQ.4)) THEN
            B4CMASS=(BPWTPCT(BPDESID(CT1))/100.0)*
     c      BPDEN(BPDESID(CT1))*BPVOL
            ALMASS=((((100-BPWTPCT(BPDESID(CT1)))/100.0)*
     c      BPDEN(BPDESID(CT1)))*BPVOL)*((2*26.981539)/101.961278)
```

```
          OMASS=((((100-BPWTPCT(BPDESID(CT1)))/100.0)*
    c     BPDEN(BPDESID(CT1)))*BPVOL)-ALMASS
          CMASS=B4CMASS*0.217374
          NEWBPMASSTOTAL=ALMASS+OMASS+CMASS+BPRAISOVALUE(1)+
    c     BPRAISOVALUE(2)
          NEWBPDEN=NEWBPMASSTOTAL/BPVOL
          ALWTPCT=(ALMASS/NEWBPMASSTOTAL)*100.0
          OWTPCT=(OMASS/NEWBPMASSTOTAL)*100.0
          CWTPCT=(CMASS/NEWBPMASSTOTAL)*100.0
          B10WTPCT=(BPRAISOVALUE(1)/NEWBPMASSTOTAL)*100.0
          B11WTPCT=(BPRAISOVALUE(2)/NEWBPMASSTOTAL)*100.0
          IF (BPWTPCT(BPDESID(CT1)).NE.(0.0)) THEN
             WRITE (100,728) NEWBPDEN
  728        FORMAT ('arbm-bp',1X,F7.3,1X,'5 0 0 0')
             IF (BPRAISOVALUE(1).NE.0) THEN
                WRITE (100,730) BPRAISONAME(1),
    c           B10WTPCT
  730           FORMAT(5X,A6,1X,G10.3)
             ENDIF
             IF (BPRAISOVALUE(2).NE.0) THEN
                WRITE (100,732) BPRAISONAME(2),
    c           B11WTPCT
  732           FORMAT(5X,A6,1X,G10.3)
             ENDIF
             WRITE (100,734) CWTPCT
  734        FORMAT (5X,' 6012',1X,G10.3)
          ELSE
             WRITE (100,736) NEWBPDEN
  736        FORMAT ('arbm-bp',1X,F7.3,1X,'2 0 0 0')
          ENDIF
*  Calculate aluminum and oxygen material specifications
          WRITE (100,738) ALWTPCT, OWTPCT
  738     FORMAT (5X,'13027',1X,F7.3,1X,' 8016',1X,F7.3)
          WRITE (100,740) MODTEMPFINAL(CT3,RELATIVE_STPT_NUM)
  740     FORMAT (5X,'4',1X,'1.0',1X,F6.1,1X,'end')
        ELSE
*  Material specification for BP other than Al2O3-B4C
          DO 742 CT4=1,BPMIXNUM
             IF (BPMIXID(CT4).EQ.BPMIX(BPRA_DESCRIPTION_ID)) THEN
                RELATIVE_BP_MIX_ID=CT4
             ENDIF
  742     CONTINUE
          NEWBPMASSTOTAL=0.0
          DO 743 CT4=1,BPNUMISOS(RELATIVE_BP_MIX_ID)
           IF (BPISOID(RELATIVE_BP_MIX_ID,CT4).EQ.5010) THEN
              NEWBPMASSTOTAL=NEWBPMASSTOTAL+BPRAISOVALUE(1)
           ELSEIF (BPISOID(RELATIVE_BP_MIX_ID,CT4).EQ.5011)
    c      THEN
              NEWBPMASSTOTAL=NEWBPMASSTOTAL+BPRAISOVALUE(2)
           ELSE
              NEWBPMASSTOTAL=NEWBPMASSTOTAL+
    c         ((BPISOWTPCT(RELATIVE_BP_MIX_ID,CT4)/100.0)*
    c         BPDEN(BPRA_DESCRIPTION_ID)*BPVOL)
           ENDIF
  743     CONTINUE
          NEWBPDEN=NEWBPMASSTOTAL/BPVOL
          WRITE (100,*) 'arbm-bp   ', NEWBPDEN,
    c     '    ', BPNUMISOS(RELATIVE_BP_MIX_ID),
    c     '  0   0   0'
          DO 750 CT4=1,BPNUMISOS(RELATIVE_BP_MIX_ID)
           IF (BPISOID(RELATIVE_BP_MIX_ID,CT4).EQ.5010) THEN
              IF (BPRAISOVALUE(1).NE.0) THEN
                 WRITE (100,744) BPRAISONAME(1),
    c            ((BPRAISOVALUE(1)/NEWBPMASSTOTAL)*100.0)
```

```
 744                  FORMAT(5X,A6,1X,G10.3)
                   ENDIF
                ELSEIF (BPISOID(RELATIVE_BP_MIX_ID,CT4).EQ.5011)
    c           THEN
                   IF (BPRAISOVALUE(2).NE.0) THEN
                      WRITE (100,746) BPRAISONAME(2),
    c                 ((BPRAISOVALUE(2)/NEWBPMASSTOTAL)*100.0)
 746                  FORMAT(5X,A6,1X,G10.3)
                   ENDIF
                ELSE
                   WRITE (100,748) BPISOID(RELATIVE_BP_MIX_ID,CT4),
    c              (((BPISOWTPCT(RELATIVE_BP_MIX_ID,CT4)/100.0)*
    c              BPDEN(BPRA_DESCRIPTION_ID)*BPVOL)/
    c              NEWBPMASSTOTAL)*100.0
 748               FORMAT (10X,I6,3X,F10.5)
                ENDIF
 750           CONTINUE
               WRITE (100,*) '           ', BPMIX(BPRA_DESCRIPTION_ID),
    c          ' 1.0 ',MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '    end'
             ENDIF
           ENDIF
         ENDIF

*  Write control rod material specification
         CR_INSERTED=.FALSE.
         IF (CRSTAT.EQ.'RODDED') THEN
         IF (RTYPE.EQ.'PWR') THEN
c            CRCOMPFLAG=.FALSE.
           CRCOMPFLAG=.TRUE.
           CR_INSERTED=.TRUE.
           DO 760 CT4=1,23
               IF (CRINS(CT1,CT2,CT4,CT3).NE.0) THEN
                  CRCOMPFLAG=.TRUE.
                  CR_INSERTED=.TRUE.
                  CR_MIXTURE_ID=CRINS(CT1,CT2,CT4,CT3)
                  CR_DESCRIPTION=CRDES(CT1,CT2,CT4,CT3)
                  EXIT
               ENDIF
 760       CONTINUE
           IF (CRCOMPFLAG.EQ..TRUE.) THEN
               DO 770 CT4=1,CRMIXNUM
                  IF (CRMIXID(CT4).EQ.CR_MIXTURE_ID) THEN
                     RELATIVE_CR_MIX_ID=CT4
                  ENDIF
 770           CONTINUE
               WRITE (100,780)
 780           FORMAT ('''')
               WRITE (100,790)
 790           FORMAT ('''',T5,' control rod material specification')
               WRITE (100,800)
 800           FORMAT ('''')
           IF (CRCLAD(CR_DESCRIPTION).NE.0) THEN
             DO 801 CT5=1,10
               IF (CRCLAD(CR_DESCRIPTION).EQ.CLADDESNUM(CT5)) THEN
                CRCLNUM=CT5
                EXIT
               ENDIF
 801          CONTINUE
              IF (CLADDESNAME(CRCLNUM).EQ.'SS304   ') THEN
               WRITE (100,802)
 802           FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055 ',
    c          '2.0 26304 69.5 28304 9.5')
               WRITE (100,803) CLADDESNUM(CRCLNUM), CLTEMP
 803           FORMAT (T12,I2,' 1.0 ',F5.1,' end')
```

```
            ELSEIF (CLADDESNAME(CRCLNUM).EQ.'SS304S ') THEN
             WRITE (100,804)
 804         FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055 ',
   c         '2.0 26000 69.5 28000 9.5')
             WRITE (100,805) CLADDESNUM(CRCLNUM), CLTEMP
 805         FORMAT (T13,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(CRCLNUM).EQ.'SS316  ') THEN
             WRITE (100,806)
 806         FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000 ',
   c         '1.0 24304 17.0 25055 2.0')
             WRITE (100,807)
 807         FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
             WRITE (100,808) CLADDESNUM(CRCLNUM), CLTEMP
 808         FORMAT (T12,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(CRCLNUM).EQ.'SS316S ') THEN
             WRITE (100,809)
 809         FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000 ',
   c         '1.0 24000 17.0 25055 2.0')
             WRITE (100,810)
 810         FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')
             WRITE (100,811) CLADDESNUM(CRCLNUM), CLTEMP
 811         FORMAT (T13,I2,' 1.0 ',F5.1,' end')
            ELSEIF (CLADDESNAME(CRCLNUM).EQ.'INCONEL') THEN
             WRITE (100,812)
 812         FORMAT ('arbm-inconel 8.3 5 0 0 0 14000 2.5',
   c         ' 22000 2.5 24000 15.0')
             WRITE (100,813)
 813         FORMAT (T13,'26000 7.0 28000 73.0')
             WRITE (100,814) CLADDESNUM(CRCLNUM), CLTEMP
 814         FORMAT (T13,I2,' 1.0 ',F5.1,' end')
            ENDIF
          ENDIF
             WRITE (100,*) 'arbm-cr   ', CRDEN(CR_DESCRIPTION),
   c         '    ', CRNUMISOS(RELATIVE_CR_MIX_ID), '   0   0   0'
             DO 820 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
                WRITE (100,815) CRISOID(RELATIVE_CR_MIX_ID,CT4),
   c            CRISOWTPCT(RELATIVE_CR_MIX_ID, CT4)
 815            FORMAT (10X,I5,3X,F10.5)
 820         CONTINUE
             WRITE (100,*) '          ', CR_MIXTURE_ID, ' 1.0 ',
   c         MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '   end'
          ENDIF
*********************************************************************************
* this is where arbitrary material specifications are written
        ELSEIF (RTYPE.EQ.'BWR') THEN
C           CRCOMPFLAG=.FALSE.
          CRCOMPFLAG=.TRUE.
          CR_INSERTED=.TRUE.
          CR_DESCRIPTION=1
          DO 1500 CT4=1,23
             IF (CRINS(CT1,CT2,CT4,CT3).NE.0) THEN
                CRCOMPFLAG=.TRUE.
                CR_INSERTED=.TRUE.
C                CR_DESCRIPTION=CRDES(CT1,CT2,CT4,CT3)
                CR_DESCRIPTION=1
                EXIT
             ENDIF
 1500     CONTINUE
          IF (CRCOMPFLAG.EQ..TRUE.) THEN
             WRITE (100,1510)
 1510        FORMAT ('''')
             WRITE (100,1520)
 1520        FORMAT ('''',T5,' GDR/CRB/MOD material',
    &                ' specifications')
```

```
                 WRITE (100,1530)
 1530            FORMAT ('''')
              IF (CRCLAD(CR_DESCRIPTION).NE.0) THEN
                DO 1540 CT5=1,10
                 IF (CRCLAD(CR_DESCRIPTION).EQ.CLADDESNUM(CT5)) THEN
                  CRCLNUM=CT5
                  EXIT
                 ENDIF
 1540           CONTINUE
                IF (CLADDESNAME(CRCLNUM).EQ.'SS304  ') THEN
                 WRITE (100,1550)
 1550            FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055 ',
     c           '2.0 26304 69.5 28304 9.5')
                 WRITE (100,1560) CLADDESNUM(CRCLNUM), CLTEMP
 1560            FORMAT (T12,I2,' 1.0 ',F5.1,' end')
                ELSEIF (CLADDESNAME(CRCLNUM).EQ.'SS304S ') THEN
                 WRITE (100,1570)
 1570            FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055 ',
     c           '2.0 26000 69.5 28000 9.5')
                 WRITE (100,1580) CLADDESNUM(CRCLNUM), CLTEMP
 1580            FORMAT (T13,I2,' 1.0 ',F5.1,' end')
                ELSEIF (CLADDESNAME(CRCLNUM).EQ.'SS316  ') THEN
                 WRITE (100,1590)
 1590            FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000 ',
     c           '1.0 24304 17.0 25055 2.0')
                 WRITE (100,1600)
 1600            FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
                 WRITE (100,1610) CLADDESNUM(CRCLNUM), CLTEMP
 1610            FORMAT (T12,I2,' 1.0 ',F5.1,' end')
                ELSEIF (CLADDESNAME(CRCLNUM).EQ.'SS316S ') THEN
                 WRITE (100,1620)
 1620            FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000 ',
     c           '1.0 24000 17.0 25055 2.0')
                 WRITE (100,1630)
 1630            FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')
                 WRITE (100,1640) CLADDESNUM(CRCLNUM), CLTEMP
 1640            FORMAT (T13,I2,' 1.0 ',F5.1,' end')
                ELSEIF (CLADDESNAME(CRCLNUM).EQ.'INCONEL') THEN
                 WRITE (100,1650)
 1650            FORMAT ('arbm-inconel 8.3 5 0 0 0 14000 2.5',
     c           ' 22000 2.5 24000 15.0')
                 WRITE (100,1660)
 1660            FORMAT (T13,'26000 7.0 28000 73.0')
                 WRITE (100,1670) CLADDESNUM(CRCLNUM), CLTEMP
 1670            FORMAT (T13,I2,' 1.0 ',F5.1,' end')
                ENDIF
              ENDIF
* ***********************************************************************
* this is where the CB specification must change for the pickup case
* in the continuation writer
* this calculation of depleted Gd isotopes is VERY specific to CRMIX #1
* 12 isotope entries

             IF(GDROD.NE.0)THEN
C this is case where gadolinina is burning out-the last time this
C section runs the fuel is considered fresh-gd has burned away

               FUELMASS=MASSTOTAL/RODS*GDROD
               GDMASS=BPRAISOVALUE(1)+BPRAISOVALUE(3)+
     &         BPRAISOVALUE(4)+BPRAISOVALUE(5)+BPRAISOVALUE(6)+
     &         BPRAISOVALUE(7)+BPRAISOVALUE(9)
               GDROMASS=(OXYGMS/RODS)*GDROD
               DO 3130 CT4=1,CARRYCOUNTER
                IF(FUELISONAME(CT4).EQ.'92234')THEN
```

```
            FMASS1=FUELMASS*FUELISOWTPCT(CT4)*0.01
           ELSEIF(FUELISONAME(CT4).EQ.'92235')THEN
            FMASS2=FUELMASS*FUELISOWTPCT(CT4)*0.01
           ELSEIF(FUELISONAME(CT4).EQ.'92236')THEN
            FMASS3=FUELMASS*FUELISOWTPCT(CT4)*0.01
           ELSEIF(FUELISONAME(CT4).EQ.'92238')THEN
            FMASS4=FUELMASS*FUELISOWTPCT(CT4)*0.01
           ENDIF
 3130     CONTINUE
          GDROMASS=(OXYGMS/RODS)*GDROD
          PRODMASS=FMASS1+FMASS2+FMASS3+FMASS4+GDMASS+GDROMASS
          NEWVALWTPCT(1)=FMASS1/PRODMASS*100
          NEWVALWTPCT(2)=FMASS2/PRODMASS*100
          NEWVALWTPCT(3)=FMASS3/PRODMASS*100
          NEWVALWTPCT(4)=FMASS4/PRODMASS*100
          NEWVALWTPCT(5)=BPRAISOVALUE(1)/PRODMASS*100
          NEWVALWTPCT(6)=BPRAISOVALUE(3)/PRODMASS*100
          NEWVALWTPCT(7)=BPRAISOVALUE(4)/PRODMASS*100
          NEWVALWTPCT(8)=BPRAISOVALUE(5)/PRODMASS*100
          NEWVALWTPCT(9)=BPRAISOVALUE(6)/PRODMASS*100
          NEWVALWTPCT(10)=BPRAISOVALUE(7)/PRODMASS*100
          NEWVALWTPCT(11)=BPRAISOVALUE(9)/PRODMASS*100
          NEWVALWTPCT(12)=GDROMASS/PRODMASS*100

          NONZERO=0.00001
          NEWDEN=PRODMASS/(GDROD*3.141593*((CID/2)**2)
     &    *NODES(AXNUM,2))
          IF(NEWDEN.LT.CRMIXDEN(1))THEN
           CRMIXDEN(1)=NEWDEN
          ENDIF
          RELATIVE_CR_MIX_ID=1
          WRITE (100,2672) RELATIVE_CR_MIX_ID,
     c    CRMIXDEN(RELATIVE_CR_MIX_ID),
     c    CRNUMISOS(RELATIVE_CR_MIX_ID)
c 2672     FORMAT(T1,'arbm-cr',I1,3X,
 2672     FORMAT(T1,'arbm-gcm',I1,3X,
     c    G14.8,3X,I2,'   0   0   0')
          DO 2690 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
           IF(NEWVALWTPCT(CT4).GE.0.00001)THEN
             WRITE (100,2680) CRISOID(RELATIVE_CR_MIX_ID,CT4),
     c       NEWVALWTPCT(CT4)
 2680        FORMAT (10X,I5,3X,F10.5)
           ELSE
             WRITE (100,2680) CRISOID(RELATIVE_CR_MIX_ID,CT4),
     c       NONZERO
           ENDIF
 2690     CONTINUE
          IF(RELATIVE_CR_MIX_ID.EQ.1)THEN
            WRITE (100,*) '           ',CRMIXID(RELATIVE_CR_MIX_ID),
     c      ' 1.0 ', FTFINAL(CT3,RELATIVE_STPT_NUM), '    end'
          ELSE
            WRITE (100,*) '           ',CRMIXID(RELATIVE_CR_MIX_ID),
     c      ' 1.0 ', MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '    end'
          ENDIF
         ENDIF


C          NEWDEN=NEWDMASS/GDROD
C          IF(NEWDEN.LT.CRMIXDEN(1))THEN
C           CRMIXDEN(1)=NEWDMASS/GDROD
C          ENDIF
c          write(99,*)'CRMIXDEN',CRMIXDEN(1)
C          RELATIVE_CR_MIX_ID=1
C          WRITE (100,3672) RELATIVE_CR_MIX_ID,
```

```
C     c        CRMIXDEN(RELATIVE_CR_MIX_ID),
C      c       CRNUMISOS(RELATIVE_CR_MIX_ID)
C 3672         FORMAT(T1,'arbm-cr',I1,3X,
C     c        G14.8,3X,I2,'    0    0    0')
C              DO 3690 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
C                 WRITE (100,3680) CRISOID(RELATIVE_CR_MIX_ID,CT4),
C     c           NEWVALWTPCT(CT4)
C 3680            FORMAT (10X,I5,3X,F10.5)
C 3690         CONTINUE
C              IF(RELATIVE_CR_MIX_ID.EQ.1)THEN
C                WRITE (100,*) '           ',CRMIXID(RELATIVE_CR_MIX_ID),
C     c          ' 1.0 ', FTFINAL(CT3,RELATIVE_STPT_NUM), '    end'
C              ELSE
C                WRITE (100,*) '           ',CRMIXID(RELATIVE_CR_MIX_ID),
C     c          ' 1.0 ', MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '    end'
C              ENDIF
C             ENDIF
C            ENDIF

C this is the case without gd

            DO 1720 RELATIVE_CR_MIX_ID=2,CRMIXNUM
             IF (RELATIVE_CR_MIX_ID.LT.10) THEN
               WRITE (100,1672) RELATIVE_CR_MIX_ID,
     c         CRMIXDEN(RELATIVE_CR_MIX_ID),
     c         CRNUMISOS(RELATIVE_CR_MIX_ID)
c 1672          FORMAT(T1,'arbm-cr',I1,3X,
 1672          FORMAT(T1,'arbm-gcm',I1,3X,
     c         G14.8,3X,I2,'    0    0    0')
             ELSEIF (RELATIVE_CR_MIX_ID.EQ.10) THEN
               WRITE (100,1674) RELATIVE_CR_MIX_ID,
     c         CRMIXDEN(RELATIVE_CR_MIX_ID),
     c         CRNUMISOS(RELATIVE_CR_MIX_ID)
c 1674          FORMAT(T1,'arbm-cr',I2,3X,
 1674          FORMAT(T1,'arbm-gcm',I2,3X,
     c         G14.8,3X,I2,'    0    0    0')
             ENDIF
               DO 1690 CT4=1,CRNUMISOS(RELATIVE_CR_MIX_ID)
                  WRITE (100,1680) CRISOID(RELATIVE_CR_MIX_ID,CT4),
     c            CRISOWTPCT(RELATIVE_CR_MIX_ID, CT4)
 1680             FORMAT (10X,I5,3X,F10.5)
 1690          CONTINUE
C              IF(RELATIVE_CR_MIX_ID.EQ.1)THEN
C               WRITE (100,*) '           ',CRMIXID(RELATIVE_CR_MIX_ID),
C     c         ' 1.0 ', FTFINAL(CT3,RELATIVE_STPT_NUM), '    end'
C              ELSE
               WRITE (100,*) '           ',CRMIXID(RELATIVE_CR_MIX_ID),
     c         ' 1.0 ', MODTEMPFINAL(CT3,RELATIVE_STPT_NUM), '    end'
C              ENDIF
 1720       CONTINUE
            ENDIF
         ENDIF
         ENDIF
*  Write APSR material specification
         IF ((CT1.EQ.CT1START).AND.(CT2.EQ.CT2GOVALUE).AND.
     c   (CT3.EQ.1)) THEN
         DO 824 CT4=1,10
            DO 823 CT5=1,20
               DO 822 CT6=1,23
                  DO 821 CT7=1,50
                     APSRINSOLD(CT4,CT5,CT6,CT7)=
     c               APSRINS(CT4,CT5,CT6,CT7)
  821             CONTINUE
  822          CONTINUE
```

```
 823        CONTINUE
 824     CONTINUE
         ENDIF
         APSR_INSERTED=.FALSE.
         IF (APSRSTAT.EQ.'RODDED') THEN
            DO 830 CT4=1,23
               APSRBOTFLAG=.FALSE.
               DO 825 CT5=50,1,-1
                  IF ((APSRINSOLD(CT1,CT2,CT4,CT5).NE.0).AND.
     c            (APSRBOTFLAG.EQ..FALSE.)) THEN
                     APSR_DESCRIPTION=APSRDES(CT1,CT2,CT4,CT5)
                     APSRBOTFLAG=.TRUE.
                     FOLNODKEEP=CT5
                     FOLSTEPKEEP=CT4
                  ENDIF
                  IF ((APSRINSOLD(CT1,CT2,CT4,CT5).EQ.0).AND.
     c            (APSRBOTFLAG.EQ..TRUE.)) THEN
                     APSRINS(CT1,CT2,CT4,CT5)=
     c               APSRFOLLOWMIX(CT1,CT2,FOLSTEPKEEP,FOLNODKEEP)
                     APSRFOLLOWDATA(CT1,CT2,CT4,CT5)=3
                  ENDIF
 825           CONTINUE
 830        CONTINUE
            FOLLOWIN=.FALSE.
            DO 831 CT4=1,23
               IF (APSRFOLLOWDATA(CT1,CT2,CT4,CT3).EQ.3) THEN
                  FOLLOWIN=.TRUE.
                  EXIT
               ENDIF
 831        CONTINUE
            IF (FOLLOWIN.EQ..TRUE.) THEN
               WRITE (100,832)
 832           FORMAT ('''')
               WRITE (100,834)
 834           FORMAT ('''',T5,' APSR follow rod material',
     c         ' specification')
               WRITE (100,836)
 836           FORMAT ('''')
            IF ((APSRFOLLOWMIX(CT1,CT2,FOLSTEPKEEP,FOLNODKEEP).NE.0)
     c      .AND.
     c      (APSRFOLLOWMIX(CT1,CT2,FOLSTEPKEEP,FOLNODKEEP).NE.2)) THEN
              DO 838 CT5=1,10
               IF (APSRFOLLOWMIX(CT1,CT2,FOLSTEPKEEP,FOLNODKEEP)
     c         .EQ.CLADDESNUM(CT5)) THEN
                APSRFOLNUM=CT5
                EXIT
               ENDIF
 838          CONTINUE
              IF (CLADDESNAME(APSRFOLNUM).EQ.'SS304   ') THEN
               WRITE (100,840)
 840           FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055 ',
     c         '2.0 26304 69.5 28304 9.5')
               WRITE (100,842) CLADDESNUM(APSRFOLNUM), CLTEMP
 842           FORMAT (T12,I2,' 1.0 ',F5.1,' end')
              ELSEIF (CLADDESNAME(APSRFOLNUM).EQ.'SS304S ') THEN
               WRITE (100,844)
 844           FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055 ',
     c         '2.0 26000 69.5 28000 9.5')
               WRITE (100,846) CLADDESNUM(APSRFOLNUM), CLTEMP
 846           FORMAT (T13,I2,' 1.0 ',F5.1,' end')
              ELSEIF (CLADDESNAME(APSRFOLNUM).EQ.'SS316   ') THEN
               WRITE (100,848)
 848           FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000 ',
     c         '1.0 24304 17.0 25055 2.0')
```

```
              WRITE (100,850)
 850          FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
              WRITE (100,852) CLADDESNUM(APSRFOLNUM), CLTEMP
 852          FORMAT (T12,I2,' 1.0 ',F5.1,' end')
             ELSEIF (CLADDESNAME(APSRFOLNUM).EQ.'SS316S ') THEN
              WRITE (100,854)
 854          FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000 ',
     c        '1.0 24000 17.0 25055 2.0')
              WRITE (100,856)
 856          FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')
              WRITE (100,858) CLADDESNUM(APSRFOLNUM), CLTEMP
 858          FORMAT (T13,I2,' 1.0 ',F5.1,' end')
             ELSEIF (CLADDESNAME(APSRFOLNUM).EQ.'INCONEL') THEN
              WRITE (100,860)
 860          FORMAT ('arbm-inconel 8.3 5 0 0 0 14000 2.5',
     c        ' 22000 2.5 24000 15.0')
              WRITE (100,862)
 862          FORMAT (T13,'26000 7.0 28000 73.0')
              WRITE (100,864) CLADDESNUM(APSRFOLNUM), CLTEMP
 864          FORMAT (T13,I2,' 1.0 ',F5.1,' end')
             ENDIF
           ENDIF
           ENDIF
           APSRCOMPFLAG=.FALSE.
           DO 865 CT4=1,23
              IF ((APSRINS(CT1,CT2,CT4,CT3).NE.0).AND.
     c        (APSRINS(CT1,CT2,CT4,CT3).NE.
     c        APSRFOLLOWMIX(CT1,CT2,FOLSTEPKEEP,FOLNODKEEP))) THEN
                 APSRCOMPFLAG=.TRUE.
                 APSR_INSERTED=.TRUE.
                 APSR_MIXTURE_ID=APSRINS(CT1,CT2,CT4,CT3)
                 APSR_DESCRIPTION=APSRDES(CT1,CT2,CT4,CT3)
                 EXIT
              ENDIF
 865       CONTINUE
           IF (APSRCOMPFLAG.EQ..TRUE.) THEN
              DO 866 CT4=1,APSRMIXNUM
                 IF (APSRMIXID(CT4).EQ.APSR_MIXTURE_ID) THEN
                    RELATIVE_APSR_MIX_ID=CT4
                 ENDIF
 866          CONTINUE
              WRITE (100,868)
 868          FORMAT ('''')
              WRITE (100,870)
 870          FORMAT ('''',T5,' axial power shaping rod material',
     c        ' specification')
              WRITE (100,880)
 880          FORMAT ('''')
           IF (APSRCLAD(APSR_DESCRIPTION).NE.0) THEN
             DO 881 CT5=1,10
              IF (APSRCLAD(APSR_DESCRIPTION).EQ.CLADDESNUM(CT5)) THEN
               APSRCLNUM=CT5
               EXIT
              ENDIF
 881         CONTINUE
             IF (CLADDESNAME(APSRCLNUM).EQ.'SS304  ') THEN
              WRITE (100,882)
 882          FORMAT ('arbm-ss304 7.92 4 0 0 0 24304 19.0 25055 ',
     c        '2.0 26304 69.5 28304 9.5')
              WRITE (100,883) CLADDESNUM(APSRCLNUM), CLTEMP
 883          FORMAT (T12,I2,' 1.0 ',F5.1,' end')
             ELSEIF (CLADDESNAME(APSRCLNUM).EQ.'SS304S ') THEN
              WRITE (100,884)
 884          FORMAT ('arbm-ss304s 7.92 4 0 0 0 24000 19.0 25055 ',
```

```
     c         '2.0 26000 69.5 28000 9.5')
               WRITE (100,885) CLADDESNUM(APSRCLNUM), CLTEMP
  885          FORMAT (T13,I2,' 1.0 ',F5.1,' end')
              ELSEIF (CLADDESNAME(APSRCLNUM).EQ.'SS316  ') THEN
               WRITE (100,886)
  886          FORMAT ('arbm-ss316 7.75 7 0 0 0 6012 0.08 14000 ',
     c         '1.0 24304 17.0 25055 2.0')
               WRITE (100,887)
  887          FORMAT (T12,'26304 65.42 28304 12.0 42000 2.5')
               WRITE (100,888) CLADDESNUM(APSRCLNUM), CLTEMP
  888          FORMAT (T12,I2,' 1.0 ',F5.1,' end')
              ELSEIF (CLADDESNAME(APSRCLNUM).EQ.'SS316S ') THEN
               WRITE (100,889)
  889          FORMAT ('arbm-ss316s 7.75 7 0 0 0 6012 0.08 14000 ',
     c         '1.0 24000 17.0 25055 2.0')
               WRITE (100,890)
  890          FORMAT (T13,'26000 65.42 28000 12.0 42000 2.5')
               WRITE (100,891) CLADDESNUM(APSRCLNUM), CLTEMP
  891          FORMAT (T13,I2,' 1.0 ',F5.1,' end')
              ELSEIF (CLADDESNAME(APSRCLNUM).EQ.'INCONEL') THEN
               WRITE (100,892)
  892          FORMAT ('arbm-inconel 8.3 5 0 0 0 14000 2.5',
     c         ' 22000 2.5 24000 15.0')
               WRITE (100,893)
  893          FORMAT (T13,'26000 7.0 28000 73.0')
               WRITE (100,894) CLADDESNUM(APSRCLNUM), CLTEMP
  894          FORMAT (T13,I2,' 1.0 ',F5.1,' end')
              ENDIF
            ENDIF
               WRITE (100,*) 'arbm-apsr   ', APSRDEN(APSR_DESCRIPTION),
     c         '   ', APSRNUMISOS(RELATIVE_APSR_MIX_ID), '   0   0   0'
               DO 900 CT4=1,APSRNUMISOS(RELATIVE_APSR_MIX_ID)
                  WRITE (100,895) APSRISOID(RELATIVE_APSR_MIX_ID,CT4),
     c            APSRISOWTPCT(RELATIVE_APSR_MIX_ID, CT4)
  895             FORMAT (10X,I5,3X,F10.5)
  900          CONTINUE
               WRITE (100,*) '           ', APSR_MIXTURE_ID, ' 1.0 ',
     c         FTFINAL(CT3,RELATIVE_STPT_NUM), '   end'
            ENDIF
         ENDIF
*  Write fuel rod fill gas material specification
         WRITE (100,910)
  910    FORMAT ('''')
         IF(RTYPE.EQ.'PWR')THEN
           WRITE (100,920)
  920      FORMAT ('he    5   end')
         ENDIF
         WRITE (100,930)
  930    FORMAT ('end comp')
*  Write base reactor lattice specifications
         WRITE (100,940)
  940    FORMAT ('''')
         WRITE (100,950)
  950    FORMAT ('''     base reactor lattice specification')
         WRITE (100,960)
  960    FORMAT ('''')
c         WRITE (100,970) PITCH, FOD, COD, CID
c  970    FORMAT ('squarepitch',3X,F7.5,3X,F6.4,3X,'1   3',3X,F6.4,
c     c   3X,'2',3X,F6.4,3X,'0   end')
         WRITE (100,970) PITCH, FOD, COD
  970    FORMAT ('squarepitch',3X,F7.5,3X,F6.4,3X,'1   3',3X,F6.4,
     c   3X,'2',' end')
* The following writing routine for 'SPECIAL' input data
* has not been formatted to compensate for FORTRAN's ingenious
```

```
* incapability to print leading zeros in numeric fields.
* Errors will occur in the FIDO input if null space exists
* between an equal sign and the appropriate value.  Therefore,
* the IIM and ICM factors must always be at least 10.
         IF (FLAG2.EQ.'SPECIAL') THEN
            IF (SZF.LT.1) THEN
               WRITE (100,980) SZF, ISN, IIM, ICM, EPS, PTC, IUS
 980           FORMAT ('more data',1X,'szf=0',F3.2,1X,'isn=',I1,1X,
     c         'iim=',I2,1X,'icm=',I2,1X,'eps=0',G7.2,1X,'ptc=0',G7.2,
     c         1X,'ius=',I1,3X,'end')
            ELSE
               WRITE (100,990) SZF, ISN, IIM, ICM, EPS, PTC, IUS
 990           FORMAT ('more data',1X,'szf=',F4.2,1X,'isn=',I1,1X,
     c         'iim=',I2,1X,'icm=',I2,1X,'eps=0',G7.2,1X,'ptc=0',G7.2,
     c         1X,'ius=',I1,3X,'end')
            ENDIF
         ELSEIF (FLAG2.NE.'SPECIAL') THEN
            IF (MESH.LT.1) THEN
               WRITE (100,1000) MESH
 1000          FORMAT ('more data',1X,'szf=0',F3.2,1X,'end')
            ELSE
               WRITE (100,1010) MESH
 1010          FORMAT ('more data',1X,'szf=',F4.2,1X,'end')
            ENDIF
         ENDIF
*  Write assembly specifications
         WRITE (100,1020)
 1020    FORMAT ('''')
         WRITE (100,1030)
 1030    FORMAT ('''     assembly specification')
         WRITE (100,1040)
 1040    FORMAT ('''')
         IF (STEPCONTROL.EQ.'Y') THEN
            CALL ZEROS(VARSTEPNUM(CT1,CT2),IRRAD_STEPS)
         ELSEIF (STEPCONTROL.EQ.'N') THEN
            CALL ZEROS(INT(BLETDOWN(CT1,CT2,2)),IRRAD_STEPS)
         ENDIF
*  Assembly specification if no BPRA, no CR, and no APSR is inserted
         IF ((BPRA_INSERTED.EQ..FALSE.).AND.(CR_INSERTED.EQ..FALSE.)
     c   .AND.(APSR_INSERTED.EQ..FALSE.)
     c   .AND.(BPRA_FOLLOW.EQ..FALSE.)
     c   .AND.(FOLLOWIN.EQ..FALSE.)) THEN
            IF (NODES(CT3,2).GE.(100.0)) THEN
               WRITE (100,1041) RODS, NODES(CT3,2), IRRAD_STEPS
 1041          FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F7.3,1X,
     c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
            ELSEIF ((NODES(CT3,2).LT.(100.0)).AND.
     c      (NODES(CT3,2).GE.(10.0))) THEN
               WRITE (100,1042) RODS, NODES(CT3,2), IRRAD_STEPS
 1042          FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F6.3,1X,
     c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
            ELSEIF (NODES(CT3,2).LT.(10.0)) THEN
               WRITE (100,1043) RODS, NODES(CT3,2), IRRAD_STEPS
 1043          FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F5.3,1X,
     c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
            ENDIF
            CALL ZEROS(PLEVEL,PLEVELCH)
            CALL ZEROS(LUZONE,LUZONECH)
            IF (MESH.LT.(1.0)) THEN
               WRITE (100,1044) PLEVELCH, LUZONECH, MESH
 1044          FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
     c         'numztotal=',A2,1X,'mxrepeats=1',1X,
     c         'mixmod=3 facmesh=',F3.2,1X,'end')
            ELSE
```

```
              WRITE (100,1045) PLEVELCH, LUZONECH, MESH
 1045         FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
    c         'numztotal=',A2,1X,'mxrepeats=1',1X,
    c         'mixmod=3 facmesh=',F4.2,1X,'end')
           ENDIF
           DO 1047 CT4=1,LUZONE
              IF (MOD(CT4,6).EQ.0) THEN
                 WRITE (100,*)
              ENDIF
              WRITE (100,1046) LMB(CT4), LRB(CT4)
 1046         FORMAT (I3,1X,F7.5,1X,$)
 1047      CONTINUE
           WRITE (100,*)
        ENDIF
*  Assembly specification if BPRA is inserted
        IF (BPRA_FOLLOW.EQ..TRUE.) THEN
           IF (NODES(CT3,2).GE.(100.0)) THEN
              WRITE (100,1050) RODS, NODES(CT3,2), IRRAD_STEPS
 1050         FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F7.3,1X,
    c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
           ELSEIF ((NODES(CT3,2).LT.(100.0)).AND.
    c      (NODES(CT3,2).GE.(10.0))) THEN
              WRITE (100,1052) RODS, NODES(CT3,2), IRRAD_STEPS
 1052         FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F6.3,1X,
    c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
           ELSEIF (NODES(CT3,2).LT.(10.0)) THEN
              WRITE (100,1054) RODS, NODES(CT3,2), IRRAD_STEPS
 1054         FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F5.3,1X,
    c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
           ENDIF
           CALL ZEROS(PLEVEL,PLEVELCH)
           CALL ZEROS(BPZONE(BPRA_DESCRIPTION_ID),BPZONECH)
           IF (MESH.LT.(1.0)) THEN
              WRITE (100,1056) PLEVELCH, BPZONECH, MESH
 1056         FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
    c         'numztotal=',A2,1X,'mxrepeats=1',1X,
    c         'mixmod=3 facmesh=',F3.2,1X,'end')
           ELSE
              WRITE (100,1058) PLEVELCH, BPZONECH, MESH
 1058         FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
    c         'numztotal=',A2,1X,'mxrepeats=1',1X,
    c         'mixmod=3 facmesh=',F4.2,1X,'end')
           ENDIF
           DO 1062 CT4=1,BPZONE(BPRA_DESCRIPTION_ID)
              IF (MOD(CT4,6).EQ.0) THEN
                 WRITE (100,*)
              ENDIF
              WRITE (100,1060) BPRFM(CT4,BPRA_DESCRIPTION_ID),
    c         BPRFR(CT4,BPRA_DESCRIPTION_ID)
 1060         FORMAT (I3,1X,F7.5,1X,$)
 1062      CONTINUE
           WRITE (100,*)
        ENDIF
        IF (BPRA_INSERTED.EQ..TRUE.) THEN
           IF (NODES(CT3,2).GE.(100.0)) THEN
              WRITE (100,1098) RODS, NODES(CT3,2), IRRAD_STEPS
 1098         FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F7.3,1X,
    c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
           ELSEIF ((NODES(CT3,2).LT.(100.0)).AND.
    c      (NODES(CT3,2).GE.(10.0))) THEN
              WRITE (100,1100) RODS, NODES(CT3,2), IRRAD_STEPS
 1100         FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F6.3,1X,
    c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
           ELSEIF (NODES(CT3,2).LT.(10.0)) THEN
```

```
              WRITE (100,1102) RODS, NODES(CT3,2), IRRAD_STEPS
 1102         FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F5.3,1X,
    c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
           ENDIF
           CALL ZEROS(PLEVEL,PLEVELCH)
           CALL ZEROS(BPZONE(BPRA_DESCRIPTION_ID),BPZONECH)
           IF (MESH.LT.(1.0)) THEN
              WRITE (100,1104) PLEVELCH, BPZONECH, MESH
 1104         FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
    c         'numztotal=',A2,1X,'mxrepeats=1',1X,
    c         'mixmod=3 facmesh=',F3.2,1X,'end')
           ELSE
              WRITE (100,1106) PLEVELCH, BPZONECH, MESH
 1106         FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
    c         'numztotal=',A2,1X,'mxrepeats=1',1X,
    c         'mixmod=3 facmesh=',F4.2,1X,'end')
           ENDIF
           DO 1110 CT4=1,BPZONE(BPRA_DESCRIPTION_ID)
              IF (MOD(CT4,6).EQ.0) THEN
                 WRITE (100,*)
              ENDIF
              WRITE (100,1108) BPMA(CT4,BPRA_DESCRIPTION_ID),
    c         BPRA(CT4,BPRA_DESCRIPTION_ID)
 1108         FORMAT (I3,1X,F7.5,1X,$)
 1110      CONTINUE
           WRITE (100,*)
        ENDIF
*  Assembly specification if CR is inserted
        IF (CR_INSERTED.EQ..TRUE.) THEN
           IF (NODES(CT3,2).GE.(100.0)) THEN
              WRITE (100,1120) RODS, NODES(CT3,2), IRRAD_STEPS
 1120         FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F7.3,1X,
    c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
           ELSEIF ((NODES(CT3,2).LT.(100.0)).AND.
    c      (NODES(CT3,2).GE.(10.0))) THEN
              WRITE (100,1130) RODS, NODES(CT3,2), IRRAD_STEPS
 1130         FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F6.3,1X,
    c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
           ELSEIF (NODES(CT3,2).LT.(10.0)) THEN
              WRITE (100,1140) RODS, NODES(CT3,2), IRRAD_STEPS
 1140         FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F5.3,1X,
    c         'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
           ENDIF
           CALL ZEROS(PLEVEL,PLEVELCH)
           CALL ZEROS(CRZONE(CR_DESCRIPTION),CRZONECH)
           IF (MESH.LT.(1.0)) THEN
              WRITE (100,1150) PLEVELCH, CRZONECH, MESH
 1150         FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
    c         'numztotal=',A2,1X,'mxrepeats=0',1X,
    c         'mixmod=3 facmesh=',F3.2,1X,'end')
           ELSE
              WRITE (100,1160) PLEVELCH, CRZONECH, MESH
 1160         FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
    c         'numztotal=',A2,1X,'mxrepeats=0',1X,
    c         'mixmod=3 facmesh=',F4.2,1X,'end')
           ENDIF
           IF (STEPCONTROL.EQ.'N') THEN
            DO 1168 CT4=1,INT(BLETDOWN(CT1,CT2,2))
              IF (CRINS(CT1,CT2,CT4,CT3).NE.0) THEN
                 DO 1162 CT5=1,CRZONE(CR_DESCRIPTION)
                    IF (MOD(CT5,6).EQ.0) THEN
                       WRITE (100,*)
                    ENDIF
                    WRITE (100,1161) CRMA(CT5,CR_DESCRIPTION),
```

```
     c              CRRA(CT5,CR_DESCRIPTION)
 1161               FORMAT (I3,1X,F7.5,1X,$)
 1162            CONTINUE
                 WRITE (100,*)
              ELSEIF (CRINS(CT1,CT2,CT4,CT3).EQ.0) THEN
                 DO 1166 CT5=1,CRZONE(CR_DESCRIPTION)
                    IF (MOD(CT5,6).EQ.0) THEN
                       WRITE (100,*)
                    ENDIF
                    WRITE (100,1164) LMC(CT5,CR_DESCRIPTION),
     c              LRC(CT5,CR_DESCRIPTION)
 1164               FORMAT (I3,1X,F7.5,1X,$)
 1166            CONTINUE
                 WRITE (100,*)
              ENDIF
 1168       CONTINUE
           ELSEIF (STEPCONTROL.EQ.'Y') THEN
            DO 1210 CT4=1,VARSTEPNUM(CT1,CT2)
              IF (CRINS(CT1,CT2,CT4,CT3).NE.0) THEN
                 DO 1180 CT5=1,CRZONE(CR_DESCRIPTION)
                    IF (MOD(CT5,6).EQ.0) THEN
                       WRITE (100,*)
                    ENDIF
                    WRITE (100,1170) CRMA(CT5,CR_DESCRIPTION),
     c              CRRA(CT5,CR_DESCRIPTION)
 1170               FORMAT (I3,1X,F7.5,1X,$)
 1180            CONTINUE
                 WRITE (100,*)
              ELSEIF (CRINS(CT1,CT2,CT4,CT3).EQ.0) THEN
                 DO 1200 CT5=1,CRZONE(CR_DESCRIPTION)
                    IF (MOD(CT5,6).EQ.0) THEN
                       WRITE (100,*)
                    ENDIF
                    WRITE (100,1190) LMC(CT5,CR_DESCRIPTION),
     c              LRC(CT5,CR_DESCRIPTION)
 1190               FORMAT (I3,1X,F7.5,1X,$)
 1200            CONTINUE
                 WRITE (100,*)
              ENDIF
 1210       CONTINUE
           ENDIF
        ENDIF
*  Assembly specification if APSR is inserted
        IF ((APSR_INSERTED.EQ..TRUE.).OR.(FOLLOWIN.EQ..TRUE.)) THEN
           IF (NODES(CT3,2).GE.(100.0)) THEN
              WRITE (100,1220) RODS, NODES(CT3,2), IRRAD_STEPS
 1220         FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F7.3,1X,
     c        'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
           ELSEIF ((NODES(CT3,2).LT.(100.0)).AND.
     c     (NODES(CT3,2).GE.(10.0))) THEN
              WRITE (100,1230) RODS, NODES(CT3,2), IRRAD_STEPS
 1230         FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F6.3,1X,
     c        'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
           ELSEIF (NODES(CT3,2).LT.(10.0)) THEN
              WRITE (100,1240) RODS, NODES(CT3,2), IRRAD_STEPS
 1240         FORMAT ('npin/assembly=',I3,1X,'fuelngth=',F5.3,1X,
     c        'ncycles=',A2,1X,'nlib/cyc=1 lightel=0')
           ENDIF
           CALL ZEROS(PLEVEL,PLEVELCH)
           CALL ZEROS(APSRZONE(APSR_DESCRIPTION),APSRZONECH)
           IF (MESH.LT.(1.0)) THEN
              WRITE (100,1250) PLEVELCH, APSRZONECH, MESH
 1250         FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
     c        'numztotal=',A2,1X,'mxrepeats=0',1X,
```

```
     c        'mixmod=3 facmesh=',F3.2,1X,'end')
            ELSE
              WRITE (100,1252) PLEVELCH, APSRZONECH, MESH
 1252         FORMAT ('printlevel=',A2,1X,'inplevel=2',1X,
     c        'numztotal=',A2,1X,'mxrepeats=0',1X,
     c        'mixmod=3 facmesh=',F4.2,1X,'end')
            ENDIF
            IF (STEPCONTROL.EQ.'N') THEN
             DO 1268 CT4=1,INT(BLETDOWN(CT1,CT2,2))
               IF ((APSRINS(CT1,CT2,CT4,CT3).NE.0).AND.
     c         (APSRFOLLOWDATA(CT1,CT2,CT4,CT3).NE.3)) THEN
                 DO 1258 CT5=1,APSRZONE(APSR_DESCRIPTION)
                   IF (MOD(CT5,6).EQ.0) THEN
                     WRITE (100,*)
                   ENDIF
                   WRITE (100,1256) APSRMA(CT5,APSR_DESCRIPTION),
     c             APSRRA(CT5,APSR_DESCRIPTION)
 1256              FORMAT (I3,1X,F7.5,1X,$)
 1258            CONTINUE
                 WRITE (100,*)
               ELSEIF ((APSRINS(CT1,CT2,CT4,CT3).NE.0).AND.
     c         (APSRFOLLOWDATA(CT1,CT2,CT4,CT3).EQ.3)) THEN
                 DO 1262 CT5=1,APSRZONE(APSR_DESCRIPTION)
                   IF (MOD(CT5,6).EQ.0) THEN
                     WRITE (100,*)
                   ENDIF
                   WRITE (100,1260) APSRFM(CT5,APSR_DESCRIPTION),
     c             APSRFR(CT5,APSR_DESCRIPTION)
 1260              FORMAT (I3,1X,F7.5,1X,$)
 1262            CONTINUE
                 WRITE (100,*)
               ELSEIF (APSRINS(CT1,CT2,CT4,CT3).EQ.0) THEN
                 DO 1266 CT5=1,APSRZONE(APSR_DESCRIPTION)
                   IF (MOD(CT5,6).EQ.0) THEN
                     WRITE (100,*)
                   ENDIF
                   WRITE (100,1264) LMD(CT5,APSR_DESCRIPTION),
     c             LRD(CT5,APSR_DESCRIPTION)
 1264              FORMAT (I3,1X,F7.5,1X,$)
 1266            CONTINUE
                 WRITE (100,*)
               ENDIF
 1268        CONTINUE
            ELSEIF (STEPCONTROL.EQ.'Y') THEN
             DO 1310 CT4=1,VARSTEPNUM(CT1,CT2)
               IF ((APSRINS(CT1,CT2,CT4,CT3).NE.0).AND.
     c         (APSRFOLLOWDATA(CT1,CT2,CT4,CT3).NE.3)) THEN
                 DO 1280 CT5=1,APSRZONE(APSR_DESCRIPTION)
                   IF (MOD(CT5,6).EQ.0) THEN
                     WRITE (100,*)
                   ENDIF
                   WRITE (100,1270) APSRMA(CT5,APSR_DESCRIPTION),
     c             APSRRA(CT5,APSR_DESCRIPTION)
 1270              FORMAT (I3,1X,F7.5,1X,$)
 1280            CONTINUE
                 WRITE (100,*)
               ELSEIF ((APSRINS(CT1,CT2,CT4,CT3).NE.0).AND.
     c         (APSRFOLLOWDATA(CT1,CT2,CT4,CT3).EQ.3)) THEN
                 DO 1290 CT5=1,APSRZONE(APSR_DESCRIPTION)
                   IF (MOD(CT5,6).EQ.0) THEN
                     WRITE (100,*)
                   ENDIF
                   WRITE (100,1285) APSRFM(CT5,APSR_DESCRIPTION),
     c             APSRFR(CT5,APSR_DESCRIPTION)
```

```
 1285             FORMAT (I3,1X,F7.5,1X,$)
 1290          CONTINUE
               WRITE (100,*)
            ELSEIF (APSRINS(CT1,CT2,CT4,CT3).EQ.0) THEN
               DO 1300 CT5=1,APSRZONE(APSR_DESCRIPTION)
                  IF (MOD(CT5,6).EQ.0) THEN
                     WRITE (100,*)
                  ENDIF
                  WRITE (100,1295) LMD(CT5,APSR_DESCRIPTION),
     c            LRD(CT5,APSR_DESCRIPTION)
 1295             FORMAT (I3,1X,F7.5,1X,$)
 1300          CONTINUE
               WRITE (100,*)
            ENDIF
 1310     CONTINUE
         ENDIF
      ENDIF
*  Write assembly depletion/decay parameters
      WRITE (100,1320)
 1320 FORMAT ('''')
      WRITE (100,1330)
 1330 FORMAT ('''     assembly depletion/decay parameters')
      WRITE (100,1340)
 1340 FORMAT ('''')
      CALL ZEROS(CYCPOS(CT1),ASSYPOSITION)
      WRITE (100,1350) CYCLEID(CT1), ASSYPOSITION
 1350 FORMAT ('''',T5,'Cycle-',A2,', ',
     c ' assembly number ',A2)
      IF (STEPCONTROL.EQ.'N') THEN
       DO 1380 CT4=3,(INT(BLETDOWN(CT1,CT2,2))+2)
         IF (CT4.LT.(BLETDOWN(CT1,CT2,2)+2)) THEN
            DOWNTIME=0.0
            IF (RTYPE.EQ.'PWR') THEN
               BORON_FRACTION=(BLETDOWN(CT1,CT2,CT4)/
     c         BLETDOWN(CT1,CT2,3))
            ELSEIF (RTYPE.EQ.'BWR') THEN
               BORON_FRACTION=(BLETDOWN(CT1,CT2,CT4)/
     c         MODREFDEN)
            ENDIF
            WRITE (100,1360) POWER(CT3,RELATIVE_STPT_NUM),
     c      BLETDOWN(CT1,CT2,1), DOWNTIME, BORON_FRACTION
 1360       FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c      G10.5,1X,'bfrac=',G9.4,1X,'end')
         ELSEIF ((CT4.EQ.(INT(BLETDOWN(CT1,CT2,2))+2)).AND.
     c   (CT2.LT.STPTS(CT1))) THEN
            DOWNTIME=STPTDAT(CT1,(CT2+1),3)
            IF (RTYPE.EQ.'PWR') THEN
               BORON_FRACTION=(BLETDOWN(CT1,CT2,CT4)/
     c         BLETDOWN(CT1,CT2,3))
            ELSEIF (RTYPE.EQ.'BWR') THEN
               BORON_FRACTION=(BLETDOWN(CT1,CT2,CT4)/
     c         MODREFDEN)
            ENDIF
            WRITE (100,1365) POWER(CT3,RELATIVE_STPT_NUM),
     c      BLETDOWN(CT1,CT2,1), DOWNTIME, BORON_FRACTION
 1365       FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c      G10.5,1X,'bfrac=',G9.4,1X,'end')
         ELSEIF ((CT4.EQ.(INT(BLETDOWN(CT1,CT2,2))+2)).AND.
     c   (CT2.EQ.STPTS(CT1))) THEN
            DOWNTIME=CYCDOWN(CT1)
            IF (RTYPE.EQ.'PWR') THEN
               BORON_FRACTION=(BLETDOWN(CT1,CT2,CT4)/
     c         BLETDOWN(CT1,CT2,3))
            ELSEIF (RTYPE.EQ.'BWR') THEN
```

```
                  BORON_FRACTION=(BLETDOWN(CT1,CT2,CT4)/
     c            MODREFDEN)
               ENDIF
               WRITE (100,1370) POWER(CT3,RELATIVE_STPT_NUM),
     c         BLETDOWN(CT1,CT2,1), DOWNTIME, BORON_FRACTION
 1370          FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c         G10.5,1X,'bfrac=',G9.4,1X,'end')
            ENDIF
 1380     CONTINUE
         ELSEIF (STEPCONTROL.EQ.'Y') THEN
          DO 1388 CT4=1,VARSTEPNUM(CT1,CT2)
            IF (CT4.LT.VARSTEPNUM(CT1,CT2)) THEN
               DOWNTIME=0.0
               IF (RTYPE.EQ.'PWR') THEN
                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
     c            VARBLETDOWN(CT1,CT2,1,2))
               ELSEIF (RTYPE.EQ.'BWR') THEN
C                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
C     c            MODREFDEN)
                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
     c            FIRSTMODEN(CT1,CT2))
               ENDIF
               WRITE (100,1382) VARPOWER(CT1,CT2,CT4,CT3),
     c         VARBLETDOWN(CT1,CT2,CT4,1), DOWNTIME, BORON_FRACTION
 1382          FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c         G10.5,1X,'h2ofrac=',G10.5,1X,'end')
            ELSEIF ((CT4.EQ.VARSTEPNUM(CT1,CT2)).AND.
     c      (CT2.LT.STPTS(CT1))) THEN
               DOWNTIME=STPTDAT(CT1,(CT2+1),3)
               IF (RTYPE.EQ.'PWR') THEN
                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
     c            VARBLETDOWN(CT1,CT2,1,2))
               ELSEIF (RTYPE.EQ.'BWR') THEN
C                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
C     c            MODREFDEN)
                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
     c            FIRSTMODEN(CT1,CT2))
               ENDIF
               WRITE (100,1384) VARPOWER(CT1,CT2,CT4,CT3),
     c         VARBLETDOWN(CT1,CT2,CT4,1), DOWNTIME, BORON_FRACTION
 1384          FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c         G10.5,1X,'h2ofrac=',G10.5,1X,'end')
            ELSEIF ((CT4.EQ.VARSTEPNUM(CT1,CT2)).AND.
     c      (CT2.EQ.STPTS(CT1))) THEN
               DOWNTIME=CYCDOWN(CT1)
               IF (RTYPE.EQ.'PWR') THEN
                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
     c            VARBLETDOWN(CT1,CT2,1,2))
               ELSEIF (RTYPE.EQ.'BWR') THEN
C                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
C     c            MODREFDEN)
                  BORON_FRACTION=(VARBLETDOWN(CT1,CT2,CT4,2)/
     c            FIRSTMODEN(CT1,CT2))
               ENDIF
               WRITE (100,1386) VARPOWER(CT1,CT2,CT4,CT3),
     c         VARBLETDOWN(CT1,CT2,CT4,1), DOWNTIME, BORON_FRACTION
 1386          FORMAT ('power=',G10.5,1X,'burn=',G9.4,1X,'down=',
     c         G10.5,1X,'h2ofrac=',G10.5,1X,'end')
            ENDIF
 1388     CONTINUE
         ENDIF
*  Store final downtime for use in extraction script 'sedexecute.exe'
         FINALDOWNTIME=DOWNTIME
*  Write input deck closing statement
```

```
        WRITE (100,1390)
 1390   FORMAT ('''')
        WRITE (100,1400)
 1400   FORMAT ('''     end of input')
        WRITE (100,1410)
 1410   FORMAT ('''')
        WRITE (100,1420)
 1420   FORMAT ('end')
        CLOSE(UNIT=100)

      RETURN
      END

********************************************************
*     This subroutine cuts the final ORIGEN output in  *
*     the SAS2H output file down to the essential      *
*     data needed for the CRC calculations.            *
********************************************************
      SUBROUTINE CUTTER (NM)
*
      INTEGER*4 LINECOUNTER, CUTLINE, NUM1, NUM2,
     c NUM3, NUM4, NUM5, NUM6, NUM7, SEDEXERESULT,
     c VERIFCOUNTER, VERIFCUTLINE, VERIFENDCUTLINE,
     c OUTPUTREMOVALRESULT
*
      CHARACTER NM*31, OUTPUTFILE*32, BPLABEL*14,
     c LINVAL*7, SEDEXECOMMAND*60, FORMATLABEL*29,
     c VERIFLABEL*14, VERIFLINVAL*7, VERIFENDLINVAL*7,
     c OUTPUTREMOVAL*35
*
      LOGICAL BPFIND, NUMZEROFLAG, VERIFFIND
*
      OUTPUTFILE(1:25)=NM(1:25)
      OUTPUTFILE(26:32)='.output'
      OPEN (UNIT=700, FILE=OUTPUTFILE, STATUS='OLD')
      REWIND(700)
      LINECOUNTER=0
      BPFIND=.FALSE.
      DO 14 WHILE (BPFIND.EQ..FALSE.)
         LINECOUNTER=LINECOUNTER+1
         READ(700,12) BPLABEL
   12    FORMAT (T98,A14)
         IF (BPLABEL.EQ.'light elements') THEN
            READ(700,*)
            READ(700,13) FORMATLABEL
   13       FORMAT (T46,A29)
            IF (FORMATLABEL.EQ.'nuclide concentrations, grams') THEN
               BPFIND=.TRUE.
            ELSE
               BACKSPACE(700)
               BACKSPACE(700)
            ENDIF
         ENDIF
   14 CONTINUE
      NUMZEROFLAG=.FALSE.
      CUTLINE=LINECOUNTER-2
               NUM1=INT(CUTLINE/1000000.0)
               IF ((NUM1.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
                  LINVAL(1:1)=' '
               ELSE
                  LINVAL(1:1)=CHAR(NUM1+48)
                  NUMZEROFLAG=.TRUE.
               ENDIF
               NUM2=INT((CUTLINE-(NUM1*1000000))/100000.0)
```

```
              IF ((NUM2.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
                 LINVAL(2:2)=' '
              ELSE
                 LINVAL(2:2)=CHAR(NUM2+48)
                 NUMZEROFLAG=.TRUE.
              ENDIF
              NUM3=INT((CUTLINE-(NUM2*100000)-
     c        (NUM1*1000000))/10000.0)
              IF ((NUM3.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
                 LINVAL(3:3)=' '
              ELSE
                 LINVAL(3:3)=CHAR(NUM3+48)
                 NUMZEROFLAG=.TRUE.
              ENDIF
              NUM4=INT((CUTLINE-(NUM3*10000)-
     c        (NUM2*100000)-(NUM1*1000000))/1000.0)
              IF ((NUM4.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
                 LINVAL(4:4)=' '
              ELSE
                 LINVAL(4:4)=CHAR(NUM4+48)
                 NUMZEROFLAG=.TRUE.
              ENDIF
              NUM5=INT((CUTLINE-(NUM4*1000)-(NUM3*10000)-
     c        (NUM2*100000)-(NUM1*1000000))/100.0)
              IF ((NUM5.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
                 LINVAL(5:5)=' '
              ELSE
                 LINVAL(5:5)=CHAR(NUM5+48)
                 NUMZEROFLAG=.TRUE.
              ENDIF
              NUM6=INT((CUTLINE-(NUM5*100)-(NUM4*1000)-
     c        (NUM3*10000)-(NUM2*100000)-
     c        (NUM1*1000000))/10.0)
              IF ((NUM6.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
                 LINVAL(6:6)=' '
              ELSE
                 LINVAL(6:6)=CHAR(NUM6+48)
                 NUMZEROFLAG=.TRUE.
              ENDIF
              NUM7=INT(CUTLINE-(NUM6*10)-(NUM5*100)-
     c        (NUM4*1000)-(NUM3*10000)-(NUM2*100000)-
     c        (NUM1*1000000))
              LINVAL(7:7)=CHAR(NUM7+48)
      REWIND(700)
      VERIFCOUNTER=0
      VERIFFIND=.FALSE.
      DO 30 WHILE (VERIFFIND.EQ..FALSE.)
         VERIFCOUNTER=VERIFCOUNTER+1
         READ(700,20) VERIFLABEL
   20    FORMAT (T45,A14)
         IF (VERIFLABEL.EQ.'program:  sas2') THEN
            VERIFFIND=.TRUE.
         ENDIF
   30 CONTINUE
      NUMZEROFLAG=.FALSE.
      VERIFCUTLINE=VERIFCOUNTER-12
      VERIFENDCUTLINE=VERIFCOUNTER+18
              NUM1=INT(VERIFCUTLINE/1000000.0)
              IF ((NUM1.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
                 VERIFLINVAL(1:1)=' '
              ELSE
                 VERIFLINVAL(1:1)=CHAR(NUM1+48)
                 NUMZEROFLAG=.TRUE.
              ENDIF
```

```
      NUM2=INT((VERIFCUTLINE-(NUM1*1000000))/100000.0)
      IF ((NUM2.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
         VERIFLINVAL(2:2)=' '
      ELSE
         VERIFLINVAL(2:2)=CHAR(NUM2+48)
         NUMZEROFLAG=.TRUE.
      ENDIF
      NUM3=INT((VERIFCUTLINE-(NUM2*100000)-
c     (NUM1*1000000))/10000.0)
      IF ((NUM3.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
         VERIFLINVAL(3:3)=' '
      ELSE
         VERIFLINVAL(3:3)=CHAR(NUM3+48)
         NUMZEROFLAG=.TRUE.
      ENDIF
      NUM4=INT((VERIFCUTLINE-(NUM3*10000)-
c     (NUM2*100000)-(NUM1*1000000))/1000.0)
      IF ((NUM4.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
         VERIFLINVAL(4:4)=' '
      ELSE
         VERIFLINVAL(4:4)=CHAR(NUM4+48)
         NUMZEROFLAG=.TRUE.
      ENDIF
      NUM5=INT((VERIFCUTLINE-(NUM4*1000)-(NUM3*10000)-
c     (NUM2*100000)-(NUM1*1000000))/100.0)
      IF ((NUM5.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
         VERIFLINVAL(5:5)=' '
      ELSE
         VERIFLINVAL(5:5)=CHAR(NUM5+48)
         NUMZEROFLAG=.TRUE.
      ENDIF
      NUM6=INT((VERIFCUTLINE-(NUM5*100)-(NUM4*1000)-
c     (NUM3*10000)-(NUM2*100000)-
c     (NUM1*1000000))/10.0)
      IF ((NUM6.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
         VERIFLINVAL(6:6)=' '
      ELSE
         VERIFLINVAL(6:6)=CHAR(NUM6+48)
         NUMZEROFLAG=.TRUE.
      ENDIF
      NUM7=INT(VERIFCUTLINE-(NUM6*10)-(NUM5*100)-
c     (NUM4*1000)-(NUM3*10000)-(NUM2*100000)-
c     (NUM1*1000000))
      VERIFLINVAL(7:7)=CHAR(NUM7+48)
      NUM1=INT(VERIFENDCUTLINE/1000000.0)
      IF ((NUM1.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
         VERIFENDLINVAL(1:1)=' '
      ELSE
         VERIFENDLINVAL(1:1)=CHAR(NUM1+48)
         NUMZEROFLAG=.TRUE.
      ENDIF
      NUM2=INT((VERIFENDCUTLINE-(NUM1*1000000))/100000.0)
      IF ((NUM2.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
         VERIFENDLINVAL(2:2)=' '
      ELSE
         VERIFENDLINVAL(2:2)=CHAR(NUM2+48)
         NUMZEROFLAG=.TRUE.
      ENDIF
      NUM3=INT((VERIFENDCUTLINE-(NUM2*100000)-
c     (NUM1*1000000))/10000.0)
      IF ((NUM3.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
         VERIFENDLINVAL(3:3)=' '
      ELSE
         VERIFENDLINVAL(3:3)=CHAR(NUM3+48)
```

```
               NUMZEROFLAG=.TRUE.
            ENDIF
            NUM4=INT((VERIFENDCUTLINE-(NUM3*10000)-
     c      (NUM2*100000)-(NUM1*1000000))/1000.0)
            IF ((NUM4.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
               VERIFENDLINVAL(4:4)=' '
            ELSE
               VERIFENDLINVAL(4:4)=CHAR(NUM4+48)
               NUMZEROFLAG=.TRUE.
            ENDIF
            NUM5=INT((VERIFENDCUTLINE-(NUM4*1000)-(NUM3*10000)-
     c      (NUM2*100000)-(NUM1*1000000))/100.0)
            IF ((NUM5.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
               VERIFENDLINVAL(5:5)=' '
            ELSE
               VERIFENDLINVAL(5:5)=CHAR(NUM5+48)
               NUMZEROFLAG=.TRUE.
            ENDIF
            NUM6=INT((VERIFENDCUTLINE-(NUM5*100)-(NUM4*1000)-
     c      (NUM3*10000)-(NUM2*100000)-
     c      (NUM1*1000000))/10.0)
            IF ((NUM6.EQ.0).AND.(NUMZEROFLAG.EQ..FALSE.)) THEN
               VERIFENDLINVAL(6:6)=' '
            ELSE
               VERIFENDLINVAL(6:6)=CHAR(NUM6+48)
               NUMZEROFLAG=.TRUE.
            ENDIF
            NUM7=INT(VERIFENDCUTLINE-(NUM6*10)-(NUM5*100)-
     c      (NUM4*1000)-(NUM3*10000)-(NUM2*100000)-
     c      (NUM1*1000000))
            VERIFENDLINVAL(7:7)=CHAR(NUM7+48)
      SEDEXECOMMAND(1:11)='sedexecute '
      SEDEXECOMMAND(12:36)=NM(1:25)
      SEDEXECOMMAND(37:37)=' '
      SEDEXECOMMAND(38:44)=LINVAL
      SEDEXECOMMAND(45:45)=' '
      SEDEXECOMMAND(46:52)=VERIFLINVAL
      SEDEXECOMMAND(53:53)=' '
      SEDEXECOMMAND(54:60)=VERIFENDLINVAL
      SEDEXERESULT=SYSTEM(SEDEXECOMMAND)
      IF (SEDEXERESULT.LT.0) THEN
         WRITE (*,*) 'AN ERROR OCCURRED DURING OUTPUT',
     c   'EXTRACTION FROM ', NM(1:25), '.output'
      ENDIF
      OUTPUTREMOVAL(1:3)='rm '
      OUTPUTREMOVAL(4:28)=NM(1:25)
      OUTPUTREMOVAL(29:35)='.output'
      OUTPUTREMOVALRESULT=SYSTEM(OUTPUTREMOVAL)
      IF (OUTPUTREMOVALRESULT.LT.0) THEN
         WRITE (*,*) 'AN ERROR OCCURRED DURING ',
     c   'DELETION OF ', NM(1:25), '.output'
      ENDIF

      RETURN
      END

*******************************************************
*     This subroutine retrieves the fuel and burnable  *
*     poison composition information from the previous *
*     depletion and decay calculation for the assembly.*
*******************************************************
      SUBROUTINE RETRIEVER (OXYGMS, MASSTOTAL,
     c FUELISONAME, FUELISOWTPCT, BPRAISONAME,
```

```
     c BPRAISOVALUE, LEFTLIST, CARRYCOUNTER,
     c PREVIOUSNAME, LEFTVAL, NM, BPRA_INSERTED)
*
      INTEGER*4 COLUMNSTART, COLUMNEND, ISONUMBER, CT1,
     c LEFTCOUNTER,CARRYCOUNTER,CT2,ISOFLAG(1000),Z,CC
*
      REAL ISOVALUE(1000), BPRAISOVALUE(10), MASSTOTAL,
     c FUELISOVALUE(1000), FUELISOWTPCT(1000), OXYGMS,
     c LEFTVAL(1000)
*
      CHARACTER ROWFLAG*7, COL1*8, COL2*8, COL3*8, COL4*8,
     c COL5*8, COL6*8, COL7*8, COL8*8, ACTINIDELABEL*9,
     c FORMATLABEL*29, ISOLABEL*6, ISONAME(1000)*6,
     c FISSPRODLABEL*16, BPRAISONAME(12)*6, ORIGNAME(297)*6,
     c LIBRARYID(297)*5, FUELISONAME(1000)*5, LEFTLIST(1000)*6,
     c PREVIOUSNAME*25, RETRIEVALFILE*29, BPLABEL*14, NM*31,
     c NOTESFILE*31, GDNAME(12)*6
*
      LOGICAL ROWFLAGLOG, ACTINIDEFIND, FISSPRODFIND, BPFIND,
     c BPRA_INSERTED
*
      DATA (LIBRARYID(Z),Z=1,297) /' 1001',
     c ' 1002',' 1003',' 2003',' 2004',' 3006',
     c ' 3007',' 4009',' 5010',' 5011',' 6012',' 7014',
     c ' 7015',' 8016',' 8017',' 9019','11023','12000',
     c '13027','14000','15031','16000','16032','17000',
     c '19000','20000','22000','23000','24000','25055',
     c '26000','27059','28000','29000','31000','32072',
     c '32073','32074','32076','33075','34074','34076',
     c '34077','34078','34080','34082','35079','35081',
     c '36078','36080','36082','36083','36084','36085',
     c '36086','37085','37086','37087','38084','38086',
     c '38087','38088','38089','38090','39089','39090',
     c '39091','40000','40090','40091','40092','40093',
     c '40094','40095','40096','41093','41094','41095',
     c '42000','42092','42094','42095','42096','42097',
     c '42098','42099','42100','43099','44096','44098',
     c '44099','44100','44101','44102','44103','44104',
     c '44105','44106','45103','45105','46102','46104',
     c '46105','46106','46107','46108','46110','47107',
     c '47109','47111','48000','48106','48108','48110',
     c '48111','48112','48113','48114','48116','48601',
     c '49113','49115','50112','50114','50115','50116',
     c '50117','50118','50119','50120','50122','50123',
     c '50124','50125','50126','51121','51123','51124',
     c '51125','51126','52120','52122','52123','52124',
     c '52125','52126','52128','52130','52132','52601',
     c '52611','53127','53129','53130','53131','53135',
     c '54124','54126','54128','54129','54130','54131',
     c '54132','54133','54134','54135','54136','55133',
     c '55134','55135','55136','55137','56134','56135',
     c '56136','56137','56138','56140','57139','57140',
     c '58140','58141','58142','58143','58144','59141',
     c '59142','59143','60142','60143','60144','60145',
     c '60146','60147','60148','60150','61147','61148',
     c '61149','61151','61601','62144','62147','62148',
     c '62149','62150','62151','62152','62153','62154',
     c '63000','63151','63152','63153','63154','63155',
     c '63156','63157','64152','64154','64155','64156',
     c '64157','64158','64160','65159','65160','66160',
     c '66161','66162','66163','66164','67165','68166',
     c '68167','71175','71176','72000','72174','72176',
     c '72177','72178','72179','72180','73181','73182',
     c '74000','74182','74183','74184','74186','75185',
```

```
     c '75187','79197','82000','83209','90230','90232',
     c '91231','91233','92232','92233','92234','92235',
     c '92236','92237','92238','93237','93238','94236',
     c '94237','94238','94239','94240','94241','94242',
     c '94243','94244','95241','95242','95243','95601',
     c '96241','96242','96243','96244','96245','96246',
     c '96247','96248','97249','98249','98250','98251',
     c '98252','98253','99253'/
      DATA (ORIGNAME(Z),Z=1,297) /' h  1 ',
     c ' h  2 ',' h  3 ','he  3 ','he  4 ','li  6 ',
     c 'li  7 ','be  9 ',' b 10 ',' b 11 ',' c 12 ',
     c ' n 14 ',' n 15 ',' o 16 ',' o 17 ',' f 19 ',
     c 'na 23 ',' mg   ','al 27 ',' si   ',' p 31 ',
     c '  s   ',' s 32 ',' cl   ','  k   ',' ca   ',
     c ' ti   ','  v   ',' cr   ','mn 55 ',' fe   ',
     c 'co 59 ',' ni   ',' cu   ',' ga   ','ge 72 ',
     c 'ge 73 ','ge 74 ','ge 76 ','as 75 ','se 74 ',
     c 'se 76 ','se 77 ','se 78 ','se 80 ','se 82 ',
     c 'br 79 ','br 81 ','kr 78 ','kr 80 ','kr 82 ',
     c 'kr 83 ','kr 84 ','kr 85 ','kr 86 ','rb 85 ',
     c 'rb 86 ','rb 87 ','sr 84 ','sr 86 ','sr 87 ',
     c 'sr 88 ','sr 89 ','sr 90 ',' y 89 ',' y 90 ',
     c ' y 91 ',' zr   ','zr 90 ','zr 91 ','zr 92 ',
     c 'zr 93 ','zr 94 ','zr 95 ','zr 96 ','nb 93 ',
     c 'nb 94 ','nb 95 ',' mo   ','mo 92 ','mo 94 ',
     c 'mo 95 ','mo 96 ','mo 97 ','mo 98 ','mo 99 ',
     c 'mo100 ','tc 99 ','ru 96 ','ru 98 ','ru 99 ',
     c 'ru100 ','ru101 ','ru102 ','ru103 ','ru104 ',
     c 'ru105 ','ru106 ','rh103 ','rh105 ','pd102 ',
     c 'pd104 ','pd105 ','pd106 ','pd107 ','pd108 ',
     c 'pd110 ','ag107 ','ag109 ','ag111 ',' cd   ',
     c 'cd106 ','cd108 ','cd110 ','cd111 ','cd112 ',
     c 'cd113 ','cd114 ','cd116 ','cd115m','in113 ',
     c 'in115 ','sn112 ','sn114 ','sn115 ','sn116 ',
     c 'sn117 ','sn118 ','sn119 ','sn120 ','sn122 ',
     c 'sn123 ','sn124 ','sn125 ','sn126 ','sb121 ',
     c 'sb123 ','sb124 ','sb125 ','sb126 ','te120 ',
     c 'te122 ','te123 ','te124 ','te125 ','te126 ',
     c 'te128 ','te130 ','te132 ','te127m','te129m',
     c ' i127 ',' i129 ',' i130 ',' i131 ',' i135 ',
     c 'xe124 ','xe126 ','xe128 ','xe129 ','xe130 ',
     c 'xe131 ','xe132 ','xe133 ','xe134 ','xe135 ',
     c 'xe136 ','cs133 ','cs134 ','cs135 ','cs136 ',
     c 'cs137 ','ba134 ','ba135 ','ba136 ','ba137 ',
     c 'ba138 ','ba140 ','la139 ','la140 ','ce140 ',
     c 'ce141 ','ce142 ','ce143 ','ce144 ','pr141 ',
     c 'pr142 ','pr143 ','nd142 ','nd143 ','nd144 ',
     c 'nd145 ','nd146 ','nd147 ','nd148 ','nd150 ',
     c 'pm147 ','pm148 ','pm149 ','pm151 ','pm148m',
     c 'sm144 ','sm147 ','sm148 ','sm149 ','sm150 ',
     c 'sm151 ','sm152 ','sm153 ','sm154 ',' eu   ',
     c 'eu151 ','eu152 ','eu153 ','eu154 ','eu155 ',
     c 'eu156 ','eu157 ','gd152 ','gd154 ','gd155 ',
     c 'gd156 ','gd157 ','gd158 ','gd160 ','tb159 ',
     c 'tb160 ','dy160 ','dy161 ','dy162 ','dy163 ',
     c 'dy164 ','ho165 ','er166 ','er167 ','lu175 ',
     c 'lu176 ',' hf   ','hf174 ','hf176 ','hf177 ',
     c 'hf178 ','hf179 ','hf180 ','ta181 ','ta182 ',
     c '  w   ',' w182 ',' w183 ',' w184 ',' w186 ',
     c 're185 ','re187 ','au197 ',' pb   ','bi209 ',
     c 'th230 ','th232 ','pa231 ','pa233 ',' u232 ',
     c ' u233 ',' u234 ',' u235 ',' u236 ',' u237 ',
     c ' u238 ','np237 ','np238 ','pu236 ','pu237 ',
     c 'pu238 ','pu239 ','pu240 ','pu241 ','pu242 ',
```

```
     c 'pu243 ','pu244 ','am241 ','am242 ','am243 ',
     c 'am242m','cm241 ','cm242 ','cm243 ','cm244 ',
     c 'cm245 ','cm246 ','cm247 ','cm248 ','bk249 ',
     c 'cf249 ','cf250 ','cf251 ','cf252 ','cf253 ',
     c 'es253 '/
*
      RETRIEVALFILE(1:25)=PREVIOUSNAME
      RETRIEVALFILE(26:29)='.cut'
      NOTESFILE(1:25)=NM(1:25)
      NOTESFILE(26:31)='.notes'
      OPEN(UNIT=300, FILE=RETRIEVALFILE, STATUS='OLD')
      OPEN(UNIT=500, FILE=NOTESFILE, STATUS='UNKNOWN')
      REWIND(300)
      REWIND(500)
      DO 5 CT1=1,1000
         ISOVALUE(CT1)=0.0
         FUELISOVALUE(CT1)=0.0
         FUELISOWTPCT(CT1)=0.0
         LEFTVAL=0.0
         ISONAME='      '
         FUELISONAME='      '
         LEFTLIST='      '
         ISOFLAG=0
    5 CONTINUE
      ROWFLAGLOG=.FALSE.
      DO 11 WHILE (ROWFLAGLOG.EQ..FALSE.)
         READ (300,10) ROWFLAG, COL1, COL2, COL3,
     c   COL4, COL5, COL6, COL7, COL8
   10    FORMAT (T15,A7,T24,A8,T34,A8,T44,A8,T54,A8,
     c   T64,A8,T74,A8,T84,A8,T94,A8)
         IF (ROWFLAG.EQ.'initial') THEN
            ROWFLAGLOG=.TRUE.
         ENDIF
   11 CONTINUE
      IF (COL1.NE.'        ') THEN
         COLUMNSTART=23
         COLUMNEND=32
      ENDIF
      IF (COL2.NE.'        ') THEN
         COLUMNSTART=33
         COLUMNEND=42
      ENDIF
      IF (COL3.NE.'        ') THEN
         COLUMNSTART=43
         COLUMNEND=52
      ENDIF
      IF (COL4.NE.'        ') THEN
         COLUMNSTART=53
         COLUMNEND=62
      ENDIF
      IF (COL5.NE.'        ') THEN
         COLUMNSTART=63
         COLUMNEND=72
      ENDIF
      IF (COL6.NE.'        ') THEN
         COLUMNSTART=73
         COLUMNEND=82
      ENDIF
      IF (COL7.NE.'        ') THEN
         COLUMNSTART=83
         COLUMNEND=92
      ENDIF
      IF (COL8.NE.'        ') THEN
         COLUMNSTART=93
```

```
          COLUMNEND=102
        ENDIF
*   Get Gd composition data for Gd-RODS
        GDNAME(1)=' gd152'
        GDNAME(2)=' gd153'
        GDNAME(3)=' gd154'
        GDNAME(4)=' gd155'
        GDNAME(5)=' gd156'
        GDNAME(6)=' gd157'
        GDNAME(7)=' gd158'
        GDNAME(8)=' gd159'
        GDNAME(9)=' gd160'
        DO 500 CT1=1,10
         BPRAISOVALUE(CT1)=0.0
         BPRAISONAME(CT1)='       '
 500    CONTINUE
        IF (BPRA_INSERTED.EQ..TRUE.) THEN
         DO 700 CC=1,9
           REWIND(300)
           BPFIND=.FALSE.
           DO 14 WHILE (BPFIND.EQ..FALSE.)
              READ(300,12) BPLABEL
  12          FORMAT (T98,A14)
              IF (BPLABEL.EQ.'light elements') THEN
                 READ(300,*)
                 READ(300,13) FORMATLABEL
  13             FORMAT (T46,A29)
                 IF (FORMATLABEL.EQ.'nuclide concentrations, grams') THEN
                    BPFIND=.TRUE.
                 ENDIF
              ENDIF
  14       CONTINUE
C           DO 24 CT1=1,25
C              READ (300,22) BPRAISONAME(1)
C   22         FORMAT(T6,A6)
C             IF (BPRAISONAME(1).EQ.' b 10 ') THEN
C                 BACKSPACE(300)
C                 EXIT
C              ENDIF
C   24      CONTINUE
C           READ (300,26) BPRAISONAME(1), BPRAISOVALUE(1)
C   26         FORMAT(T6,A6,T<COLUMNSTART>,G10.2)
C           READ (300,29) BPRAISONAME(2), BPRAISOVALUE(2)
C   29         FORMAT(T6,A6,T<COLUMNSTART>,G10.2)
C           IF (BPRAISONAME(1).EQ.' b 10 ') THEN
C              BPRAISONAME(1)='  5010'
C           ENDIF
C           IF (BPRAISONAME(2).EQ.' b 11 ') THEN
C              BPRAISONAME(2)='  5011'
C           ENDIF
C        ENDIF
* this section retrieves Gd concentrations and calculates
* new Gd2O3-UO2 concentrations to be in subseqent SAS2H calculation
  22          FORMAT(T5,A6)
  26          FORMAT(T5,A6,T<COLUMNSTART>,G10.2)
           DO 324 CT1=1,65
              READ (300,22) BPRAISONAME(CC)
              IF (BPRAISONAME(CC).EQ.GDNAME(CC))THEN
                 BACKSPACE(300)
                 READ (300,26) BPRAISONAME(CC), BPRAISOVALUE(CC)
                 GOTO 700
              ENDIF
 324       CONTINUE
           BPRAISONAME(CC)=GDNAME(CC)
```

```
        BPRAISOVALUE(CC)=0.0
 700   CONTINUE

C         READ (300,26) BPRAISONAME(CC), BPRAISOVALUE(CC)
C         DO 325 CT1=1,10
C            READ (300,22) BPRAISONAME(2)
C            IF (BPRAISONAME(2).EQ.' gd153') THEN
C               BACKSPACE(300)
C               EXIT
C            ENDIF
C  325    CONTINUE
C         DO 326 CT1=1,10
C            READ (300,22) BPRAISONAME(3)
C            IF (BPRAISONAME(3).EQ.' gd154') THEN
C               BACKSPACE(300)
C               EXIT
C            ENDIF
C  326    CONTINUE
C         READ (300,26) BPRAISONAME(3), BPRAISOVALUE(3)

        IF (BPRAISONAME(1).EQ.' gd152') THEN
           BPRAISONAME(1)='  64152'
        ENDIF
        IF (BPRAISONAME(2).EQ.' gd153') THEN
           BPRAISONAME(2)='  64153'
        ENDIF
        IF (BPRAISONAME(3).EQ.' gd154') THEN
           BPRAISONAME(3)='  64154'
        ENDIF
        IF (BPRAISONAME(4).EQ.' gd155') THEN
           BPRAISONAME(4)='  64155'
        ENDIF
        IF (BPRAISONAME(5).EQ.' gd156') THEN
           BPRAISONAME(5)='  64156'
        ENDIF
        IF (BPRAISONAME(6).EQ.' gd157') THEN
           BPRAISONAME(6)='  64157'
        ENDIF
        IF (BPRAISONAME(7).EQ.' gd158') THEN
           BPRAISONAME(7)='  64158'
        ENDIF
        IF (BPRAISONAME(8).EQ.' gd159') THEN
           BPRAISONAME(8)='  64159'
        ENDIF
        IF (BPRAISONAME(9).EQ.' gd160') THEN
           BPRAISONAME(9)='  64160'
        ENDIF
     ENDIF
*  Get fuel composition data
     REWIND(300)
     ACTINIDEFIND=.FALSE.
     DO 50 WHILE (ACTINIDEFIND.EQ..FALSE.)
        READ(300,30) ACTINIDELABEL
 30     FORMAT (T103,A9)
        IF (ACTINIDELABEL.EQ.'actinides') THEN
           READ(300,*)
           READ(300,40) FORMATLABEL
 40        FORMAT (T46,A29)
           IF (FORMATLABEL.EQ.'nuclide concentrations, grams') THEN
              ACTINIDEFIND=.TRUE.
           ENDIF
        ENDIF
 50  CONTINUE
     READ(300,*)
```

```
      READ(300,*)
      ISOLABEL='      '
      ISONUMBER=0
      DO 70 WHILE (ISOLABEL.NE.'tal   ')
         ISONUMBER=ISONUMBER+1
         READ(300,60) ISONAME(ISONUMBER), ISOVALUE(ISONUMBER)
 60      FORMAT(T6,A6,T<COLUMNSTART>,G10.2)
         ISOLABEL=ISONAME(ISONUMBER)
         IF (ISOLABEL.EQ.'tal   ') THEN
            ISONAME(ISONUMBER)='      '
            ISOVALUE(ISONUMBER)=0
         ENDIF
 70   CONTINUE
      ISONUMBER=ISONUMBER-1
      REWIND(300)
      FISSPRODFIND=.FALSE.
      DO 110 WHILE (FISSPRODFIND.EQ..FALSE.)
         READ(300,90) FISSPRODLABEL
 90      FORMAT (T96,A16)
         IF (FISSPRODLABEL.EQ.'fission products') THEN
            READ(300,*)
            READ(300,100) FORMATLABEL
100         FORMAT (T46,A29)
            IF (FORMATLABEL.EQ.'nuclide concentrations, grams') THEN
               FISSPRODFIND=.TRUE.
            ENDIF
         ENDIF
110   CONTINUE
      READ(300,*)
      READ(300,*)
      ISOLABEL='      '
      DO 130 WHILE (ISOLABEL.NE.'tal   ')
         ISONUMBER=ISONUMBER+1
         READ(300,120) ISONAME(ISONUMBER), ISOVALUE(ISONUMBER)
120      FORMAT(T6,A6,T<COLUMNSTART>,G10.2)
         ISOLABEL=ISONAME(ISONUMBER)
         IF (ISOLABEL.EQ.'      ') THEN
            ISONUMBER=ISONUMBER-1
            READ(300,*)
            READ(300,*)
            READ(300,*)
            READ(300,*)
            READ(300,*)
         ENDIF
         IF (ISOLABEL.EQ.'tal   ') THEN
            ISONAME(ISONUMBER)='      '
            ISOVALUE(ISONUMBER)=0
         ENDIF
130   CONTINUE
      ISONUMBER=ISONUMBER-1
      WRITE (500,*) 'FUEL COMPOSITION'
      DO 140 CT1=1,ISONUMBER
         WRITE (500,*) ISONAME(CT1), '      ', ISOVALUE(CT1)
140   CONTINUE
      WRITE (500,*)
      IF (BPRA_INSERTED.EQ..TRUE.) THEN
         WRITE (500,*) 'Gd IN Gd-ROD'
         DO 150 CT1=1,9
            WRITE (500,*) BPRAISONAME(CT1), '      ',
     c      BPRAISOVALUE(CT1)
150      CONTINUE
      ENDIF
      MASSTOTAL=OXYGMS
      LEFTCOUNTER=0
```

```
      CARRYCOUNTER=0
      DO 190 CT1=1,ISONUMBER
         DO 180 CT2=1,297
            IF (ISONAME(CT1).EQ.ORIGNAME(CT2)) THEN
               CARRYCOUNTER=CARRYCOUNTER+1
               ISOFLAG(CT1)=1
               FUELISONAME(CARRYCOUNTER)=LIBRARYID(CT2)
               FUELISOVALUE(CARRYCOUNTER)=ISOVALUE(CT1)
            ENDIF
            IF ((CT2.EQ.297).AND.(ISOFLAG(CT1).NE.1)) THEN
               LEFTCOUNTER=LEFTCOUNTER+1
               LEFTLIST(LEFTCOUNTER)=ISONAME(CT1)
               LEFTVAL(LEFTCOUNTER)=ISOVALUE(CT1)
            ENDIF
 180     CONTINUE
 190  CONTINUE
      DO 195 CT1=1,CARRYCOUNTER
         MASSTOTAL=MASSTOTAL+FUELISOVALUE(CT1)
 195  CONTINUE
      DO 200 CT1=1,CARRYCOUNTER
         FUELISOWTPCT(CT1)=(FUELISOVALUE(CT1)/MASSTOTAL)*100.0
 200  CONTINUE
      WRITE (500,*) 'SAS2H FUEL COMPOSITION INPUT FROM ORIGIN OUTPUT'
      DO 230 CT1=1,CARRYCOUNTER
         WRITE (500,*) FUELISONAME(CT1), '      ', FUELISOVALUE(CT1)
 230  CONTINUE
      WRITE (500,*) 'ISOTOPES IN ORIGIN OUTPUT LEFT OUT OF SAS2H INPUT'
      DO 240 CT1=1,LEFTCOUNTER
         WRITE (500,*) LEFTLIST(CT1), '      ', LEFTVAL(CT1)
 240  CONTINUE

      RETURN
      END

*****************************************************************************
*       Two digit integer conversion subroutine which adds leading zeros      *
*****************************************************************************
      SUBROUTINE ZEROS(IN,CHOUT)

      INTEGER*4 IN
      CHARACTER CHOUT*2, CH1, CH2

      CH1=CHAR((IN/10)+48)
      CH2=CHAR((IN-(INT(IN/10)*10))+48)
      CHOUT(1:1)=CH1
      CHOUT(2:2)=CH2

      RETURN
      END
```

SPACE, Version 01
System Processor and CRAFT Executor

Developed by David P. Henderson
Framatome Cogema Fuels
High-Level Waste Division

under contract with the

Management and Operating Contractor for the
Yucca Mountain High-Level Radioactive Waste Repository Project

Table of Contents

Item **Page**

1. Introduction

The System Processor and CRAFT Executor (SPACE), Version 01, software routine directs the performance of assembly depletion and decay calculations relevant to Boiling Water Reactor (BWR) Commercial Reactor Critical (CRC) evaluations. The SPACE software routine is a driver routine for CRAFT, Version 4C (Attachment I). The SPACE software routine generates node input files for CRAFT and then executes CRAFT for all of the nodal depletions required for a user-specified assembly. CRAFT creates node input files for the SAS2H control module of the SCALE 4.3 modular code system (Reference 1). Appropriate isotopic concentrations relevant to the CRC evaluations that include the subsequent depletion and decay calculations for the fuel assembly are extracted and stored by CRAFT as it generates and executes SAS2H cases.

2. SPACE Applications

The SPACE software routine directs the performance of the CRAFT software routine for depletion and decay calculations required to simulate the complete irradiation history of a fuel assembly. Thus, all of the assembly nodal information for a specified assembly is required to be entered in the SPACE input file ("spacein"). When SPACE is executed, it creates all of the required CRAFT input files and then initiates the execution of CRAFT. After CRAFT has begun processing SAS2H node input calculations, the SPACE software routine pauses until all of the SAS2H depletion and decay calculations are complete. At this point the SPACE software routine finishes and writes out completed job messages.

3. SPACE Methodology

The purpose of the SPACE software routine is to create the input files and execute CRAFT Version 4C for an entire BWR assembly. SPACE generates the input files and then calls a script file named LIFTOFF that executes CRAFT in a nodal depletion and decay calculation. SPACE is made up of 7 subroutines that read and process assembly nodal data and then write out the appropriate CRAFT input files for nodal calculations. Figure 3-1 shows a diagram of the operation of the SPACE software routine.

Data describing the assembly to be depleted is compiled in the SPACE input file called "spacein". The assembly is described in terms of axial nodes with each node having unique data. Descriptions of this input file are given in Sections 5 and 7 of this attachment. The DATA_ACQUISITION subroutine reads in the nodal information from the input file and loads all of this information into variables and variable arrays. After the assembly data is read, the SPACE software routine proceeds to the REWRITER subroutine where all of the input data is written back out into a text file called REWRITE. This file enables the user to check the data that has been written into the variable locations in the SPACE software routine. During the operation of the SPACE software routine a text log file called SPACELOG is created and updated with progress statements and error messages when they are required.

After the REWRITE text file is completed the SPACE software routine proceeds to the NODEWRITER subroutines. The NODEWRITER subroutine writes the CRAFT input files for each node of the assembly. These files include those cases that require a depletion with a gadolinia-bearing fuel rod (GDR) modeled in the SAS2H "supercell " or "Path B" model as well as uranium blanket nodes that contain no GDRs. Processing and CRAFT input file creation performed by the NODEWRITER subroutine will be described in more detail in Sections 4.4 and 4.5.

The NODEWRITER subroutine combined with the MODTERP subroutine interpolate user-defined moderator density values and then begin writing out the specific CRAFT input files for nodal calculations. These CRAFT input files are numbered according to their unique node number. NODEWRITER uses the ZEROS subroutine to develop these nodal input file names. After all of the CRAFT input files are written for each node, the LAUNCH subroutine is called. The LAUNCH subroutine executes a script file called LIFTOFF that runs CRAFT for each nodal depletion calculation required for a complete BWR assembly. After completion of the CRAFT orchestrated nodal depletion and decay calculations using SAS2H, system control is returned to the SPACE software routine, which ends after the appropriate completion messages are written to SPACELOG.

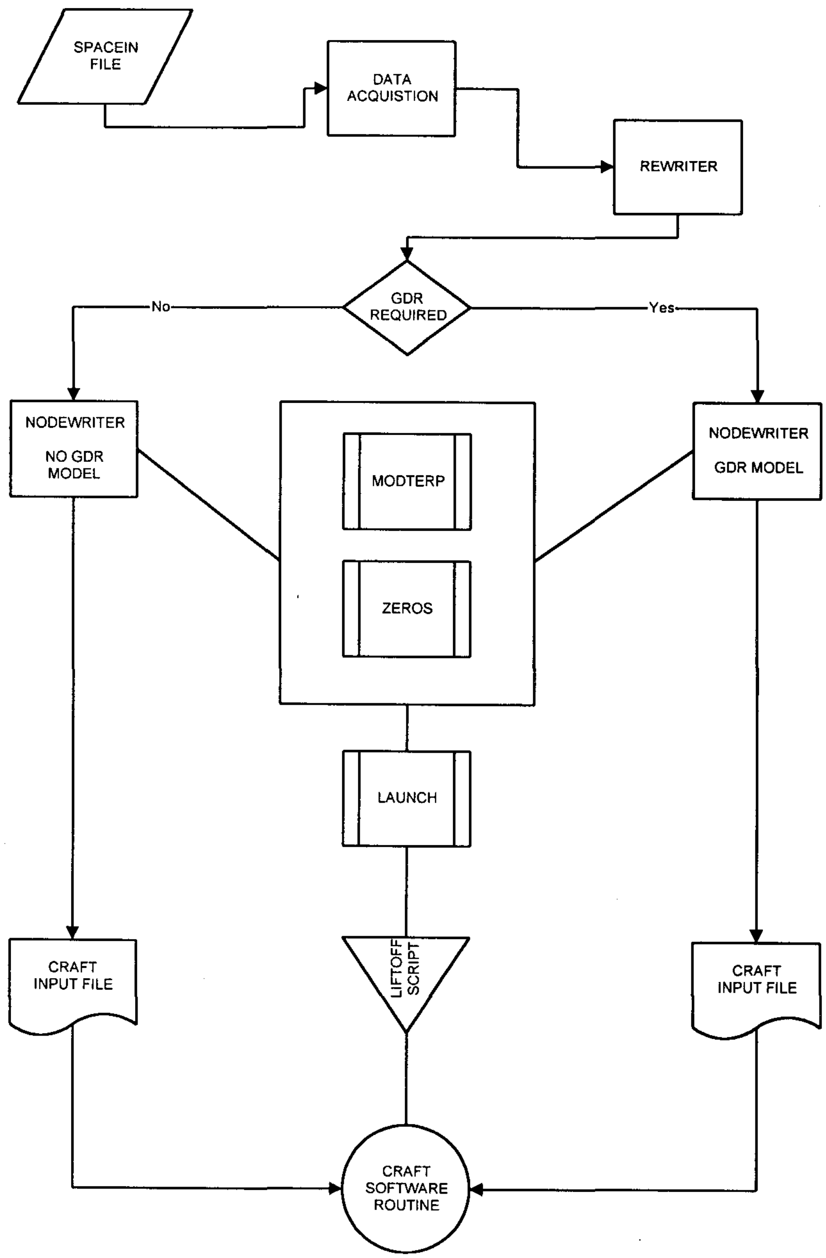

Figure 3-1. Operation Flow Diagram for SPACE Software Routine

4. SPACE Subroutine Descriptions

The SPACE software routine is organized into 7 subroutines. Subroutine number 5 (NODEWRITERII) is obsolete in SPACE, Version 01 and is not used. Each of the subroutines has a specific responsibility during the SPACE software routine operation. The following sections provide descriptions of the structure and task of each subroutine. The main program block and subroutines comprising the SPACE software routine include the following:

1) Main program block:
   PROGRAM SPACE
2) Reactor and problem data acquisition subroutine:
   DATA_ACQUISITION
3) Input data REWRITE file generator:
   REWRITER
4) CRAFT case input file writer:
   NODEWRITER
5) Obsolete: RESTART CRAFT case input file writer.
   Obsolete: NODEWRITERII.
6) CRAFT input filename character generator:
   ZEROS
7) Moderator density interpolation subroutine:
   MODTERP
8) LIFTOFF script executor for CRAFT runs:
   LAUNCH

4.1. Program SPACE

The main program block orchestrates the SPACE software routine operation. The purpose of the main program block is to define fixed data sets and initiate the sequential execution of appropriate subroutines required during SPACE operation. The subroutines initiated by the main program block of the SPACE software routine include the following, in order of initiation: DATA_ACQUISITION, REWRITER, NODEWRITER, and LAUNCH. The subroutines MODTERP and ZEROS are called by the NODEWRITER subroutine.

4.2. DATA_ACQUISITION Subroutine

The DATA_ACQUISITION subroutine opens and reads in all of the assembly nodal data necessary for the CRAFT-SAS2H depletion and decay calculations. All of the input parameters are read into variables or variable arrays for storage until later use by other subroutines. This subroutine also writes out input error messages to the SPACELOG text file if they occur. A detailed description of the SPACE input file format is provided in Sections 5 and 7.

4.3. REWRITER Subroutine

This subroutine creates a new text file called REWRITE that contains an echo of the SPACE input data. This file serves as a check for the proper input of assembly and node data into variables and variable arrays contained in the SPACE software routine.

4.4. NODEWRITER Subroutine

The NODEWRITER subroutine first determines the effective full power days (EFPD) midpoints for a depletion calculation and then calls the MODTERP subroutine which determines the midstep moderator densities. The MODTERP subroutine is described in detail in Section 4.5. Equation 4-1 is used for calculating midstep EFPD. For each nodal burnup step that is required in a specific depletion calculation, the EFPD midpoints for each burnup step in the calculation are determined and then the moderator density is interpolated as a function of EFPD across these midpoints and is written into the CRAFT input file for that node. The NODEWRITER subroutine writes all of the necessary data to the CRAFT input, and a time/date/version stamp is also written to each CRAFT input file. The identification for the reactor, assembly, and node is also written into each CRAFT input case. This subroutine is used to write out the CRAFT input file for a nodal depletion calculation that models either a GDR Path B model or a water rod Path B model (uranium blanket Path B model).

Equation 4-1. Midstep EFPD Calculation

$$(\mathrm{EFPD})^{i}_{\mathrm{midstep}} = \sum (\mathrm{EFPD})^{i}{}_{\mathrm{step}} + 1/2 \cdot (\mathrm{EFPD})^{i+1}{}_{\mathrm{step}}$$

where: *EFPD* is effective full power days and this equation is calculated for each step through a statepoint calculation; *i* superscript is for the i^{th} step.

4.5. MODTERP Subroutine

The MODTERP subroutine interpolates user-specified nodal in-channel moderator densities as a function of midstep EFPD for a single CRAFT-SAS2H depletion calculation. Midstep EFPDs are determined as described in Section 4.4 and are passed to the MODTERP subroutine. The equation used in this subroutine is given in Equation 4-2.

Equation 4-2. Midstep Moderator Density Calculation

$$\mathrm{density}^{i}{}_{\mathrm{midstep}} = \frac{((\mathrm{CEFPD})^{i}{}_{\mathrm{step}} \cdot (\mathrm{MDEN2} - \mathrm{MDEN1}))}{(\mathrm{EFPD2} - \mathrm{EFPD1})} + \mathrm{MDEN1}$$

where: density is the midstep moderator density for a specific EFPD point in the CRAFT-SAS2H depletion calculation; *CEFPD* is the cumulative effective full power days at the specific EFPD point;

MDEN1 and MDEN2 are the beginning and ending moderator density, respectively, for the statepoint calculation; EFPD1 and EFPD2 are the beginning and ending EFPD values, respectively, for the statepoint calculation.

4.6. ZEROS Subroutine

The ZEROS subroutine converts the integer counter identification for a specific node calculation to a character variable that can be used to name the CRAFT input files for each specific node. This subroutine is called by the NODEWRITER subroutines when each nodal CRAFT input file is created.

4.7. LAUNCH Subroutine

The LAUNCH subroutine is called after all of the CRAFT input files have been made. This subroutine executes the LIFTOFF script file that executes CRAFT through all of the required node calculations. If the LAUNCH is not successful, the SPACE software routine aborts and error messages are written to the SPACELOG file.

5. SPACE Input Summary

The following table summarizes the input line formats and parameters required to perform a depletion calculation using SPACE. The SPACE input file filename must be "spacein". Table 5-1 categorizes the input into groups of lines that may contain one or more lines depending upon the specific requirement of the data being listed. For example, one input group can be one line with one character of data or one group can be made up of 10 individual lines of input in the input file. Table 5-2 provides an example of a SPACE input file.

Table 5-1. SPACE Input Summary

Input Group Number	Special Notes	Format	Description
1		1 Character	Flag to execute CRAFT, Version 4C; "N" = do not execute CRAFT; "Y" = execute CRAFT
2		21 Characters	Assembly identifier (i.e., "D9 - no CRB tracking")
3		3 Characters	Problem prefix to be used as an identifier in all filenames
4		15 Characters	SCALE cross-section library to be utilized by SAS2H
5		Integer	Number of nodal fuel enrichments required to describe assembly
5A	☆	Real	Nodal fuel enrichment (U-235 wt%) for each required node description in assembly
6		Integer	Number of different GDRs per node required to describe assembly
6A	☆	Real	Number of GDRs per node for each required node description in assembly
7		Integer	Number of additional material specifications required to describe control blade (CRB) and GDR components in assembly
8		Integer	Number of Path B models required to describe assembly
8A	☆	Real	For each Path B model required in input group 8, this line must contain the mass of uranium in the node in units of grams for the specified Path B model
8B	☆	Integer	For each Path B model required in input group 8, this line must contain the total number of fuel rods in the node for the specified Path B model
9		Real	Rod pitch in node (cm)
10		Real	Smeared fuel pellet diameter (cm); (this value should be fuel rod cladding inner diameter)
11		Real	Fuel rod cladding inner diameter (cm)
12		Real	Fuel rod cladding outer diameter (cm)
13		Real	Reference moderator density (g/cm^3)
14	ℜ	Integer	For each Path B model required in input group 8, this line should contain the number of radial zones required in the Path B model

Input Group Number	Special Notes	Format	Description
14A & 14B	ℜ	Integer, Real	Mixture number followed by Path B radius for each radial zone specified in input group 14; this configuration of mixture number and radial zone must be repeated for each Path B model to specify the correct model when a CRB would be present, i.e., borated moderator placed in the bypass moderator radial zone instead of pure moderator
15		Integer	Number of reactor cycles in which the assembly is inserted
16	ℜ	2 Characters	Reactor cycle identifier in which assembly is inserted
17	ℜ	Integer	Number of CRC statepoints in reactor cycle in which the node is inserted (BOC is always considered statepoint 1 in a cycle)
18	ℜ	Real	Statepoint EFPD
19	ℜ	Real	Length to statepoint in calendar days
20	ℜ	Real	Downtime at statepoint
21	ℜ	Real	Days of downtime at EOC
22	ℜ	Real	Total cycle length in EFPD
23	ℜ	Real	Total cycle length in calendar days
24	ℜ	Integer	Integer position of statepoint calculation
25	☆,ℜ	Integer	Relative cycle number to which moderator density data applies
26	☆,ℜ	Integer	Relative statepoint number in the relative cycle to which moderator density data applies (BOC statepoint equals 1)
27	☆,ℜ	Real	Number of irradiation steps to next statepoint
28	☆,ℜ	Real, Real	Irradiation step length in EFPD, in-channel moderator density (g/cm^3) for irradiation step
29		Integer	Number of axial nodes for CRC calculation, (for all calculations this value should always be 1)
30	ℜ	Real, Real	Node number, node height (cm)
31	ℜ	1 Character	This character should be "N". Obsolete RESTART function is not used

Input Group Number	Special Notes	Format	Description
32	☆,ℜ	Integer, Integer, Integer, Integer, Integer, Integer, Integer	For each specified node in the assembly, seven integer values selecting the Path B model configuration are required on this line; the integer values determine: enrichment # for specified node uranium mass # for specified node Path B model # for specified node gadolinia mixture # for specified node GDRs required # in specified node borated moderator mixture # for specified node moderator mixture # for specified node
33		Integer	An arbitrary 2-digit integer should be specified on this line. The obsolete RESTART function is not used
34	☆	Integer	Number of previously defined irradiation steps in which the node contains a CRB
34A	☆,ℜ	Integer	Number of delimited axial assembly sections containing the CRB during the specific irradiation step
34B	☆,ℜ	Integer, Integer, Integer, Integer, Integer, Integer, Integer	Relative cycle number containing the CRB, Relative statepoint in cycle (BOC=stpt 1), Relative irradiation step number, Bottom node number containing CRB, Top node number containing CRB, CRB absorber material mixture
35	☆,ℜ	Integer, Integer, Real	Depending upon number of additional mixtures as specified in input group 7, values for the number of isotopes in the GDR or CRB absorber material mixture, SAS2H material mixture identifier for the GDR or CRB absorber material mixture, and density for GDR or CRB material mixture should be specified on this line; (note: the first three materials specified in this format should always be GDR absorber materials and the remaining material specifications should describe the model without and with CRBs)
35A	☆,ℜ	Integer, Real	SCALE nuclide identifier for the GDR or CRB absorber material mixture, wt% for nuclide in mixture; the ordering of GDR isotopes is important and should follow the listed format: 92234 0.0000 92235 0.0000 92236 0.0000 92238 0.0000 64152 0.0000 64154 0.0000 64155 0.0000 64156 0.0000 64157 0.0000 64158 0.0000 64160 0.0000 8016 0.0000

Input Group Number	Special Notes	Format	Description
36	ℜ	Integer	Number of axial nodes for clad temperature input
37	ℜ	Real, Real	Axial node number for clad temperature input Corresponding axial node height (cm)
38	ℜ	Real	Axial node clad temperature input data (K)
39	ℜ	Integer	Number of axial nodes for moderator temperature input
40	ℜ	Real, Real	Axial node number for moderator temperature input Corresponding axial node height (cm)
41	ℜ	Real	Axial node moderator temperature input data (K)
42	ℜ	Integer	Number of axial nodes for fuel temperature input
43	ℜ	Real, Real	Axial node number for fuel temperature input Corresponding axial node height (cm)
44	ℜ	Real	Axial node fuel temperature input data (K)
45	ℜ	Integer	Number of axial nodes for moderator density input
46	ℜ	Real, Real	Axial node number for moderator density input Corresponding axial node height (cm)
47	ℜ	Real	Axial node moderator density input data (g/cm^3)
48	ℜ	Integer	Number of axial nodes for burnup input
49	ℜ	Real, Real	Axial node number for burnup input, Corresponding axial node height (cm)
50	ℜ	Real	Axial node burnup input data (GWd/MTU)

✭: The existence of these input lines is dependent on certain previous input line values. The detailed descriptions for these input lines in Section 7 explain the various dependencies.

ℜ: These are recursive input lines that must be entered multiple times in a specific grouping format. The detailed descriptions for the recursive input lines in Section 7 explain the specific grouping formats and number of required input iterations.

█ : The continuous shaded boxes in the special notes column indicate groupings of recursive input lines. The format and content of these recursive groupings are explained in the detailed input descriptions in Section 7.

Table 5-2. SPACE Input File

```
This is a SPACE INPUT File
Y                              : Flag to execute CRAFT_V4C
b4 - b - no blade              : Reactor Identifier
QC2                            : Prefix Identifier for reactor
44group                        : Scale cross-section library
3                              : Number of fuel enrichments
0.71                           : U-235 wt% enrichment in U of UO2
3.19                           : U-235 wt% enrichment in U of UO2
3.19                           : U-235 wt% enrichment in U of UO2
2                              : Different Gd rods in node type
0                              : Gd rods in node type 1
7                              : Gd rods in node type 2
5                              : Number of additional mixtures
8                              : Number of pathb models
7275.9                         : Grams of U/node 1,10
14551.7                        : Grams of U/node 2,3,5
21827.6                        : Grams of U/node 4,6,7,8
29103.4                        : Grams of U/node 9
1.0                            : Grams of U/node set=1
1.0                            : Grams of U/node set=1
1.0                            : Grams of U/node set=1
1.0                            : Grams of U/node set=1
62                             : Number of fuel rods in node for 1st pathb
62                             : Number of fuel rods in node for 2nd pathb
62                             : Number of fuel rods in node for 3rd pathb
62                             : Number of fuel rods in node for 4th pathb
62                             : Number of fuel rods in node for 5th pathb
62                             : Number of fuel rods in node for 6th pathb
62                             : Number of fuel rods in node for 7th pathb
62                             : Number of fuel rods in node for 8th pathb
1.6256                         : Pin-pitch in assembly (cm)
1.0643                         : Fuel pellet diameter (cm),CLADID
1.0643                         : Fuel rod cladding ID (cm)
1.2268                         : Fuel rod cladding OD (cm)
0.7396                         : Reference moderator density (g/cc)
6                              : # of radial zones in the 1st PathB model(0.71)
3       1.0643                 : mixture number, radius
2       1.2268                 : mixture number, radius
3       1.2970                 : mixture number, radius
500     7.3372                 : mixture number, radius
2       7.5733                 : mixture number, radius
12      8.5982                 : mixture number, radius
6                              : # of radial zones in the 2nd PathB model(0.71+CB)
3       1.0643                 : mixture number, radius
2       1.2268                 : mixture number, radius
3       1.2970                 : mixture number, radius
500     7.3372                 : mixture number, radius
2       7.5733                 : mixture number, radius
11      8.5982                 : mixture number, radius
6                              : # of radial zones in the 3rd PathB model(3.19/7Gd4.0)
6       0.5321                 : mixture number, radius
2       0.6134                 : mixture number, radius
3       0.9171                 : mixture number, radius
500     2.7295                 : mixture number, radius
2       2.8201                 : mixture number, radius
12      3.2498                 : mixture number, radius
6                              : # of radial zones in the 4th PathB model(3.19/7Gd4.0+CB)
6       0.5321                 : mixture number, radius
2       0.6134                 : mixture number, radius
3       0.9171                 : mixture number, radius
```

```
500     2.7295          : mixture number, radius
2       2.8201          : mixture number, radius
11      3.2498          : mixture number, radius
6                       : # of radial zones in the 5th PathB model(3.19/7Gd4)
7       0.5321          : mixture number, radius
2       0.6134          : mixture number, radius
3       0.9171          : mixture number, radius
500     2.7295          : mixture number, radius
2       2.8201          : mixture number, radius
12      3.2498          : mixture number, radius
6                       : # of radial zones in the 6th PathB model(3.19/7Gd4+CB)
7       0.5321          : mixture number, radius
2       0.6134          : mixture number, radius
3       0.9171          : mixture number, radius
500     2.7295          : mixture number, radius
2       2.8201          : mixture number, radius
11      3.2498          : mixture number, radius
6                       : # of radial zones in the 7th PathB model(3.19/7Gd4.0)
8       0.5321          : mixture number, radius
2       0.6134          : mixture number, radius
3       0.9171          : mixture number, radius
500     2.7295          : mixture number, radius
2       2.8201          : mixture number, radius
12      3.2498          : mixture number, radius
6                       : # of radial zones in the 8th PathB model(3.19/7Gd4.0+CB)
8       0.5321          : mixture number, radius
2       0.6134          : mixture number, radius
3       0.9171          : mixture number, radius
500     2.7295          : mixture number, radius
2       2.8201          : mixture number, radius
11      3.2498          : mixture number, radius
5                       : # of insertion reactor cycles
09                      : Insertion reactor CYCLE identifier
2                       : # of stpts in cycle
0.0                     : Stpt EFPD
0.0                     : Length to stpt in calendar days
0.0                     : Downtime at stpt
167.5                   : Stpt EFPD
167.5                   : Length to stpt in calendar days
0.0                     : Downtime at stpt
75.0                    : Days of downtime at EOC
348.4                   : Total cycle EFPD
348.4                   : Total cycle length in calendar days
01                      : Integer position of assembly in cycle
10                      : Insertion reactor CYCLE identifier
2                       : # of stpts in cycle
0.0                     : Stpt EFPD
0.0                     : Length to stpt in calendar days
0.0                     : Downtime at stpt
222.5                   : Stpt EFPD
222.5                   : Length to stpt in calendar days
0.0                     : Downtime at stpt
95.0                    : Days of downtime at EOC
467.5                   : Total cycle EFPD
467.5                   : Total cycle length in calendar days
02                      : Integer position of assembly in cycle
11                      : Insertion reactor CYCLE identifier
2                       : # of stpts in cycle
0.0                     : Stpt EFPD
0.0                     : Length to stpt in calendar days
0.0                     : Downtime at stpt
180.3                   : Stpt EFPD
180.3                   : Length to stpt in calendar days
0.0                     : Downtime at stpt
```

```
138.0        : Days of downtime at EOC
484.2        : Total cycle EFPD
484.2        : Total cycle length in calendar days
03           : Integer position of assembly in cycle
12           : Insertion reactor CYCLE identifier
2            : # of stpts in cycle
0.0          : Stpt EFPD
0.0          : Length to stpt in calendar days
0.0          : Downtime at stpt
142.2        : Stpt EFPD
142.2        : Length to stpt in calendar days
0.0          : Downtime at stpt
85.0         : Days of downtime at EOC
263.7        : Total cycle EFPD
263.7        : Total cycle length in calendar days
04           : Integer position of assembly in cycle
13           : Insertion reactor CYCLE identifier
6            : # of stpts in cycle
0.0          : Stpt EFPD
0.0          : Length to stpt in calendar days
0.0          : Downtime at stpt
10.1         : Stpt EFPD
10.1         : Length to stpt in calendar days
16.7         : Downtime at stpt
112.9        : Stpt EFPD
112.9        : Length to stpt in calendar days
16.7         : Downtime at stpt
224.4        : Stpt EFPD
224.4        : Length to stpt in calendar days
13.1         : Downtime at stpt
324.7        : Stpt EFPD
324.7        : Length to stpt in calendar days
16.7         : Downtime at stpt
387.1        : Stpt EFPD
387.1        : Length to stpt in calendar days
0.0          : Downtime at stpt
136.0        : Days of downtime at EOC
387.1        : Total cycle EFPD
387.1        : Total cycle length in calendar days
05           : Integer position of assembly in cycle
1            : Relative insertion cycle
1            : Relative statepoint in insertion cycle
3            : Number of steps in statepoint calculation
55.82        : Step length (EFPD)
55.82        : Step length (EFPD)
55.82        : Step length (EFPD)
2            : Relative statepoint in insertion cycle
3            : Number of steps in statepoint calculation
60.32        : Step length (EFPD)
60.32        : Step length (EFPD)
60.32        : Step length (EFPD)
2            : Relative insertion cycle
1            : Relative statepoint in insertion cycle
4            : Number of steps in statepoint calculation
55.62        : Step length (EFPD)
55.62        : Step length (EFPD)
55.62        : Step length (EFPD)
55.62        : Step length (EFPD)
2            : Relative statepoint in insertion cycle
4            : Number of steps in statepoint calculation
61.25        : Step length (EFPD)
61.25        : Step length (EFPD)
61.25        : Step length (EFPD)
61.25        : Step length (EFPD)
```

```
3                              : Relative insertion cycle
1                              : Relative statepoint in insertion cycle
3                              : Number of steps in statepoint calculation
60.1                           : Step length (EFPD)
60.1                           : Step length (EFPD)
60.1                           : Step length (EFPD)
2                              : Relative statepoint in insertion cycle
5                              : Number of steps in statepoint calculation
60.78                          : Step length (EFPD)
60.78                          : Step length (EFPD)
60.78                          : Step length (EFPD)
60.78                          : Step length (EFPD)
60.78                          : Step length (EFPD)
4                              : Relative insertion cycle
1                              : Relative statepoint in insertion cycle
3                              : Number of steps in statepoint calculation
47.4                           : Step length (EFPD)
47.4                           : Step length (EFPD)
47.4                           : Step length (EFPD)
2                              : Relative statepoint in insertion cycle
2                              : Number of steps in statepoint calculation
60.75                          : Step length (EFPD)
60.75                          : Step length (EFPD)
5                              : Relative insertion cycle
1                              : Relative statepoint in insertion cycle
1                              : Number of steps in statepoint calculation
10.1                           : Step length (EFPD)
2                              : Relative statepoint in insertion cycle
2                              : Number of steps in statepoint calculation
51.42                          : Step length (EFPD)
51.42                          : Step length (EFPD)
3                              : Relative statepoint in insertion cycle
2                              : Number of steps in statepoint calculation
55.73                          : Step length (EFPD)
55.73                          : Step length (EFPD)
4                              : Relative statepoint in insertion cycle
2                              : Number of steps in statepoint calculation
50.17                          : Step length (EFPD)
50.17                          : Step length (EFPD)
5                              : Relative statepoint in insertion cycle
1                              : Number of steps in statepoint calculation
62.34                          : Step length (EFPD)
6                              : Relative statepoint in insertion cycle
1                              : Number of steps in statepoint calculation
62.34                          : Step length (EFPD)
10                             : # of axial nodes in CRC format
1     15.2400                  : Node #, node height (cm)
2     30.4800                  : Node #, node height (cm)
3     30.4800                  : Node #, node height (cm)
4     45.7200                  : Node #, node height (cm)
5     30.4800                  : Node #, node height (cm)
6     45.7200                  : Node #, node height (cm)
7     45.7200                  : Node #, node height (cm)
8     45.7200                  : Node #, node height (cm)
9     60.9600                  : Node #, node height (cm)
10    15.2400                  : Node #, node height (cm)
N                              : CRAFT restart required=Y; node1
N                              : CRAFT restart required=Y; node2
N                              : CRAFT restart required=Y; node3
N                              : CRAFT restart required=Y; node4
N                              : CRAFT restart required=Y; node5
N                              : CRAFT restart required=Y; node6
N                              : CRAFT restart required=Y; node7
N                              : CRAFT restart required=Y; node8
```

```
N                              : CRAFT restart required=Y; node9
N                              : CRAFT restart required=Y; node10
 1  1  1  1  1  5  5           : Enrich, Mass, PathB, Gdmix, Gdrod, Bmodmix, Modmix: Node 1
 2  2  3  1  2  5  5           : Enrich, Mass, PathB, Gdmix, Gdrod, Bmodmix, Modmix: Node 2
 2  2  3  1  2  5  5           : Enrich, Mass, PathB, Gdmix, Gdrod, Bmodmix, Modmix: Node 3
 2  3  3  1  2  5  5           : Enrich, Mass, PathB, Gdmix, Gdrod, Bmodmix, Modmix: Node 4
 2  2  5  2  2  5  5           : Enrich, Mass, PathB, Gdmix, Gdrod, Bmodmix, Modmix: Node 5
 2  3  5  2  2  5  5           : Enrich, Mass, PathB, Gdmix, Gdrod, Bmodmix, Modmix: Node 6
 2  3  5  2  2  5  5           : Enrich, Mass, PathB, Gdmix, Gdrod, Bmodmix, Modmix: Node 7
 2  3  7  3  2  5  5           : Enrich, Mass, PathB, Gdmix, Gdrod, Bmodmix, Modmix: Node 8
 2  4  3  2  2  5  5           : Enrich, Mass, PathB, Gdmix, Gdrod, Bmodmix, Modmix: Node 9
 1  1  1  1  1  5  5           : Enrich, Mass, PathB, Gdmix, Gdrod, Bmodmix, Modmix: Node 10
10                             : RSTART PathB model #
1                              : Number of irradiation steps with CRA inserted
1                              : Number of axial sections with CRA inserted in step
4 1 1 1 10 9                   : Input card 47B
12     6     9.82              : # elements in 3.19/7Gd4.0 absorber, mixture #, density
92234   0.0230                 : SCALE isotope ID, Isotope wt%
92235   2.6994                 : SCALE isotope ID, Isotope wt%
92236   0.0124                 : SCALE isotope ID, Isotope wt%
92238   81.886                 : SCALE isotope ID, Isotope wt%
64152   0.0067                 : SCALE isotope ID, Isotope wt%
64154   0.0741                 : SCALE isotope ID, Isotope wt%
64155   0.5060                 : SCALE isotope ID, Isotope wt%
64156   0.7044                 : SCALE isotope ID, Isotope wt%
64157   0.5420                 : SCALE isotope ID, Isotope wt%
64158   0.8657                 : SCALE isotope ID, Isotope wt%
64160   0.7715                 : SCALE isotope ID, Isotope wt%
8016    11.909                 : SCALE isotope ID, Isotope wt%
12     7     9.82              : # elements in 3.19/7Gd4.0 absorber, mixture #, density
92234   0.0230                 : SCALE isotope ID, Isotope wt%
92235   2.6994                 : SCALE isotope ID, Isotope wt%
92236   0.0124                 : SCALE isotope ID, Isotope wt%
92238   81.886                 : SCALE isotope ID, Isotope wt%
64152   0.0067                 : SCALE isotope ID, Isotope wt%
64154   0.0741                 : SCALE isotope ID, Isotope wt%
64155   0.5060                 : SCALE isotope ID, Isotope wt%
64156   0.7044                 : SCALE isotope ID, Isotope wt%
64157   0.5420                 : SCALE isotope ID, Isotope wt%
64158   0.8657                 : SCALE isotope ID, Isotope wt%
64160   0.7715                 : SCALE isotope ID, Isotope wt%
8016    11.909                 : SCALE isotope ID, Isotope wt%
12     8     9.82              : # elements in 3.19/7Gd4.0 absorber, mixture #, density
92234   0.0230                 : SCALE isotope ID, Isotope wt%
92235   2.6994                 : SCALE isotope ID, Isotope wt%
92236   0.0124                 : SCALE isotope ID, Isotope wt%
92238   81.886                 : SCALE isotope ID, Isotope wt%
64152   0.0067                 : SCALE isotope ID, Isotope wt%
64154   0.0741                 : SCALE isotope ID, Isotope wt%
64155   0.5060                 : SCALE isotope ID, Isotope wt%
64156   0.7044                 : SCALE isotope ID, Isotope wt%
64157   0.5420                 : SCALE isotope ID, Isotope wt%
64158   0.8657                 : SCALE isotope ID, Isotope wt%
64160   0.7715                 : SCALE isotope ID, Isotope wt%
8016    11.909                 : SCALE isotope ID, Isotope wt%
13     11    1.6704            : # elements bypass moderator + CB,  mixture #, density
6012    1.1784                 : SCALE isotope ID, Isotope wt%
7014    0.0581                 : SCALE isotope ID, Isotope wt%
14000   0.4357                 : SCALE isotope ID, Isotope wt%
15000   0.0261                 : SCALE isotope ID, Isotope wt%
16000   0.0174                 : SCALE isotope ID, Isotope wt%
24000   11.037                 : SCALE isotope ID, Isotope wt%
25000   1.1618                 : SCALE isotope ID, Isotope wt%
26000   39.935                 : SCALE isotope ID, Isotope wt%
```

```
28000   5.3734          : SCALE isotope ID, Isotope wt%
1000    4.1070          : SCALE isotope ID, Isotope wt%
5010    0.7510          : SCALE isotope ID, Isotope wt%
5011    3.3239          : SCALE isotope ID, Isotope wt%
8016    32.595          : SCALE isotope ID, Isotope wt%
2   12    0.7396        : # elements in bypass moderator, mixture #, density
1000    11.19           : SCALE isotope ID, Isotope wt%
8016    88.81           : SCALE isotope ID, Isotope wt%
10                      :# of clad temp axial nodes; stcalc =  1
 1    15.2400           :Node #, node height (cm)
 2    30.4800           :Node #, node height (cm)
 3    30.4800           :Node #, node height (cm)
 4    45.7200           :Node #, node height (cm)
 5    30.4800           :Node #, node height (cm)
 6    45.7200           :Node #, node height (cm)
 7    45.7200           :Node #, node height (cm)
 8    45.7200           :Node #, node height (cm)
 9    60.9600           :Node #, node height (cm)
10    15.2400           :Node #, node height (cm)
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
10                      :# of clad temp axial nodes; stcalc =  2
 1    15.2400           :Node #, node height (cm)
 2    30.4800           :Node #, node height (cm)
 3    30.4800           :Node #, node height (cm)
 4    45.7200           :Node #, node height (cm)
 5    30.4800           :Node #, node height (cm)
 6    45.7200           :Node #, node height (cm)
 7    45.7200           :Node #, node height (cm)
 8    45.7200           :Node #, node height (cm)
 9    60.9600           :Node #, node height (cm)
10    15.2400           :Node #, node height (cm)
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
10                      :# of clad temp axial nodes; stcalc =  3
 1    15.2400           :Node #, node height (cm)
 2    30.4800           :Node #, node height (cm)
 3    30.4800           :Node #, node height (cm)
 4    45.7200           :Node #, node height (cm)
 5    30.4800           :Node #, node height (cm)
 6    45.7200           :Node #, node height (cm)
 7    45.7200           :Node #, node height (cm)
 8    45.7200           :Node #, node height (cm)
 9    60.9600           :Node #, node height (cm)
10    15.2400           :Node #, node height (cm)
620.0
620.0
620.0
```

```
620.0
620.0
620.0
620.0
620.0
620.0
620.0
10                          :# of clad temp axial nodes; stcalc =  4
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
10                          :# of clad temp axial nodes; stcalc =  5
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
10                          :# of clad temp axial nodes; stcalc =  6
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
620.0
620.0
620.0
620.0
```

```
620.0
620.0
620.0
620.0
620.0
620.0
10                      :# of clad temp axial nodes; stcalc =  7
 1     15.2400          :Node #, node height (cm)
 2     30.4800          :Node #, node height (cm)
 3     30.4800          :Node #, node height (cm)
 4     45.7200          :Node #, node height (cm)
 5     30.4800          :Node #, node height (cm)
 6     45.7200          :Node #, node height (cm)
 7     45.7200          :Node #, node height (cm)
 8     45.7200          :Node #, node height (cm)
 9     60.9600          :Node #, node height (cm)
10     15.2400          :Node #, node height (cm)
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
10                      :# of clad temp axial nodes; stcalc =  8
 1     15.2400          :Node #, node height (cm)
 2     30.4800          :Node #, node height (cm)
 3     30.4800          :Node #, node height (cm)
 4     45.7200          :Node #, node height (cm)
 5     30.4800          :Node #, node height (cm)
 6     45.7200          :Node #, node height (cm)
 7     45.7200          :Node #, node height (cm)
 8     45.7200          :Node #, node height (cm)
 9     60.9600          :Node #, node height (cm)
10     15.2400          :Node #, node height (cm)
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
10                      :# of clad temp axial nodes; stcalc =  9
 1     15.2400          :Node #, node height (cm)
 2     30.4800          :Node #, node height (cm)
 3     30.4800          :Node #, node height (cm)
 4     45.7200          :Node #, node height (cm)
 5     30.4800          :Node #, node height (cm)
 6     45.7200          :Node #, node height (cm)
 7     45.7200          :Node #, node height (cm)
 8     45.7200          :Node #, node height (cm)
 9     60.9600          :Node #, node height (cm)
10     15.2400          :Node #, node height (cm)
620.0
620.0
620.0
620.0
620.0
```

```
620.0
620.0
620.0
620.0
620.0
10                          :# of clad temp axial nodes; stcalc = 10
 1    15.2400               :Node #, node height (cm)
 2    30.4800               :Node #, node height (cm)
 3    30.4800               :Node #, node height (cm)
 4    45.7200               :Node #, node height (cm)
 5    30.4800               :Node #, node height (cm)
 6    45.7200               :Node #, node height (cm)
 7    45.7200               :Node #, node height (cm)
 8    45.7200               :Node #, node height (cm)
 9    60.9600               :Node #, node height (cm)
10    15.2400               :Node #, node height (cm)
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
10                          :# of clad temp axial nodes; stcalc = 11
 1    15.2400               :Node #, node height (cm)
 2    30.4800               :Node #, node height (cm)
 3    30.4800               :Node #, node height (cm)
 4    45.7200               :Node #, node height (cm)
 5    30.4800               :Node #, node height (cm)
 6    45.7200               :Node #, node height (cm)
 7    45.7200               :Node #, node height (cm)
 8    45.7200               :Node #, node height (cm)
 9    60.9600               :Node #, node height (cm)
10    15.2400               :Node #, node height (cm)
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
10                          :# of clad temp axial nodes; stcalc = 12
 1    15.2400               :Node #, node height (cm)
 2    30.4800               :Node #, node height (cm)
 3    30.4800               :Node #, node height (cm)
 4    45.7200               :Node #, node height (cm)
 5    30.4800               :Node #, node height (cm)
 6    45.7200               :Node #, node height (cm)
 7    45.7200               :Node #, node height (cm)
 8    45.7200               :Node #, node height (cm)
 9    60.9600               :Node #, node height (cm)
10    15.2400               :Node #, node height (cm)
620.0
620.0
620.0
620.0
620.0
620.0
```

```
620.0
620.0
620.0
620.0
10                              :# of clad temp axial nodes; stcalc = 13
 1     15.2400                  :Node #, node height (cm)
 2     30.4800                  :Node #, node height (cm)
 3     30.4800                  :Node #, node height (cm)
 4     45.7200                  :Node #, node height (cm)
 5     30.4800                  :Node #, node height (cm)
 6     45.7200                  :Node #, node height (cm)
 7     45.7200                  :Node #, node height (cm)
 8     45.7200                  :Node #, node height (cm)
 9     60.9600                  :Node #, node height (cm)
10     15.2400                  :Node #, node height (cm)
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
620.0
10                              :# of modd temp axial nodes; stcalc =  1
 1     15.2400                  :Node #, node height (cm)
 2     30.4800                  :Node #, node height (cm)
 3     30.4800                  :Node #, node height (cm)
 4     45.7200                  :Node #, node height (cm)
 5     30.4800                  :Node #, node height (cm)
 6     45.7200                  :Node #, node height (cm)
 7     45.7200                  :Node #, node height (cm)
 8     45.7200                  :Node #, node height (cm)
 9     60.9600                  :Node #, node height (cm)
10     15.2400                  :Node #, node height (cm)
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
10                              :# of modd temp axial nodes; stcalc =  2
 1     15.2400                  :Node #, node height (cm)
 2     30.4800                  :Node #, node height (cm)
 3     30.4800                  :Node #, node height (cm)
 4     45.7200                  :Node #, node height (cm)
 5     30.4800                  :Node #, node height (cm)
 6     45.7200                  :Node #, node height (cm)
 7     45.7200                  :Node #, node height (cm)
 8     45.7200                  :Node #, node height (cm)
 9     60.9600                  :Node #, node height (cm)
10     15.2400                  :Node #, node height (cm)
559.1
559.1
559.1
559.1
559.1
559.1
559.1
```

```
559.1
559.1
559.1
10                          :# of modd temp axial nodes; stcalc =  3
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
10                          :# of modd temp axial nodes; stcalc =  4
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
10                          :# of modd temp axial nodes; stcalc =  5
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
```

```
559.1
559.1
10                      :# of modd temp axial nodes; stcalc =  6
 1    15.2400           :Node #, node height (cm)
 2    30.4800           :Node #, node height (cm)
 3    30.4800           :Node #, node height (cm)
 4    45.7200           :Node #, node height (cm)
 5    30.4800           :Node #, node height (cm)
 6    45.7200           :Node #, node height (cm)
 7    45.7200           :Node #, node height (cm)
 8    45.7200           :Node #, node height (cm)
 9    60.9600           :Node #, node height (cm)
10    15.2400           :Node #, node height (cm)
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
10                      :# of modd temp axial nodes; stcalc =  7
 1    15.2400           :Node #, node height (cm)
 2    30.4800           :Node #, node height (cm)
 3    30.4800           :Node #, node height (cm)
 4    45.7200           :Node #, node height (cm)
 5    30.4800           :Node #, node height (cm)
 6    45.7200           :Node #, node height (cm)
 7    45.7200           :Node #, node height (cm)
 8    45.7200           :Node #, node height (cm)
 9    60.9600           :Node #, node height (cm)
10    15.2400           :Node #, node height (cm)
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
10                      :# of modd temp axial nodes; stcalc =  8
 1    15.2400           :Node #, node height (cm)
 2    30.4800           :Node #, node height (cm)
 3    30.4800           :Node #, node height (cm)
 4    45.7200           :Node #, node height (cm)
 5    30.4800           :Node #, node height (cm)
 6    45.7200           :Node #, node height (cm)
 7    45.7200           :Node #, node height (cm)
 8    45.7200           :Node #, node height (cm)
 9    60.9600           :Node #, node height (cm)
10    15.2400           :Node #, node height (cm)
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
```

```
559.1
10                      :# of modd temp axial nodes; stcalc =  9
 1     15.2400          :Node #, node height (cm)
 2     30.4800          :Node #, node height (cm)
 3     30.4800          :Node #, node height (cm)
 4     45.7200          :Node #, node height (cm)
 5     30.4800          :Node #, node height (cm)
 6     45.7200          :Node #, node height (cm)
 7     45.7200          :Node #, node height (cm)
 8     45.7200          :Node #, node height (cm)
 9     60.9600          :Node #, node height (cm)
10     15.2400          :Node #, node height (cm)
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
10                      :# of modd temp axial nodes; stcalc = 10
 1     15.2400          :Node #, node height (cm)
 2     30.4800          :Node #, node height (cm)
 3     30.4800          :Node #, node height (cm)
 4     45.7200          :Node #, node height (cm)
 5     30.4800          :Node #, node height (cm)
 6     45.7200          :Node #, node height (cm)
 7     45.7200          :Node #, node height (cm)
 8     45.7200          :Node #, node height (cm)
 9     60.9600          :Node #, node height (cm)
10     15.2400          :Node #, node height (cm)
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
10                      :# of modd temp axial nodes; stcalc = 11
 1     15.2400          :Node #, node height (cm)
 2     30.4800          :Node #, node height (cm)
 3     30.4800          :Node #, node height (cm)
 4     45.7200          :Node #, node height (cm)
 5     30.4800          :Node #, node height (cm)
 6     45.7200          :Node #, node height (cm)
 7     45.7200          :Node #, node height (cm)
 8     45.7200          :Node #, node height (cm)
 9     60.9600          :Node #, node height (cm)
10     15.2400          :Node #, node height (cm)
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
```

```
10                          :# of modd temp axial nodes; stcalc = 12
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
10                          :# of modd temp axial nodes; stcalc = 13
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
559.1
10                          :# of fuel temp axial nodes; stcalc =  1
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
 627.2
 899.6
 998.2
1051.5
1002.2
 961.3
 923.9
 882.1
 777.2
 612.6
10                          :# of fuel temp axial nodes; stcalc =  2
```

```
 1     15.2400            :Node #, node height (cm)
 2     30.4800            :Node #, node height (cm)
 3     30.4800            :Node #, node height (cm)
 4     45.7200            :Node #, node height (cm)
 5     30.4800            :Node #, node height (cm)
 6     45.7200            :Node #, node height (cm)
 7     45.7200            :Node #, node height (cm)
 8     45.7200            :Node #, node height (cm)
 9     60.9600            :Node #, node height (cm)
10     15.2400            :Node #, node height (cm)
 634.7
 935.4
 996.1
 986.1
 996.9
 995.3
 982.6
 956.8
 842.8
 628.3
10                        :# of fuel temp axial nodes; stcalc =  3
 1     15.2400            :Node #, node height (cm)
 2     30.4800            :Node #, node height (cm)
 3     30.4800            :Node #, node height (cm)
 4     45.7200            :Node #, node height (cm)
 5     30.4800            :Node #, node height (cm)
 6     45.7200            :Node #, node height (cm)
 7     45.7200            :Node #, node height (cm)
 8     45.7200            :Node #, node height (cm)
 9     60.9600            :Node #, node height (cm)
10     15.2400            :Node #, node height (cm)
 623.0
 868.6
 914.4
 931.5
 918.2
 891.1
 861.3
 833.6
 752.8
 607.5
10                        :# of fuel temp axial nodes; stcalc =  4
 1     15.2400            :Node #, node height (cm)
 2     30.4800            :Node #, node height (cm)
 3     30.4800            :Node #, node height (cm)
 4     45.7200            :Node #, node height (cm)
 5     30.4800            :Node #, node height (cm)
 6     45.7200            :Node #, node height (cm)
 7     45.7200            :Node #, node height (cm)
 8     45.7200            :Node #, node height (cm)
 9     60.9600            :Node #, node height (cm)
10     15.2400            :Node #, node height (cm)
 614.6
 789.1
 791.4
 804.4
 822.9
 840.6
 852.5
 850.9
 789.5
 619.8
10                        :# of fuel temp axial nodes; stcalc =  5
 1     15.2400            :Node #, node height (cm)
```

```
2     30.4800          :Node #, node height (cm)
3     30.4800          :Node #, node height (cm)
4     45.7200          :Node #, node height (cm)
5     30.4800          :Node #, node height (cm)
6     45.7200          :Node #, node height (cm)
7     45.7200          :Node #, node height (cm)
8     45.7200          :Node #, node height (cm)
9     60.9600          :Node #, node height (cm)
10    15.2400          :Node #, node height (cm)
 630.1
 873.7
 936.4
 987.0
 986.1
 946.2
 886.5
 825.8
 746.1
 608.7
10                     :# of fuel temp axial nodes; stcalc =   6
 1    15.2400          :Node #, node height (cm)
 2    30.4800          :Node #, node height (cm)
 3    30.4800          :Node #, node height (cm)
 4    45.7200          :Node #, node height (cm)
 5    30.4800          :Node #, node height (cm)
 6    45.7200          :Node #, node height (cm)
 7    45.7200          :Node #, node height (cm)
 8    45.7200          :Node #, node height (cm)
 9    60.9600          :Node #, node height (cm)
10    15.2400          :Node #, node height (cm)
 635.0
 849.8
 858.2
 871.0
 892.1
 911.7
 920.0
 897.4
 800.0
 620.9
10                     :# of fuel temp axial nodes; stcalc =   7
 1    15.2400          :Node #, node height (cm)
 2    30.4800          :Node #, node height (cm)
 3    30.4800          :Node #, node height (cm)
 4    45.7200          :Node #, node height (cm)
 5    30.4800          :Node #, node height (cm)
 6    45.7200          :Node #, node height (cm)
 7    45.7200          :Node #, node height (cm)
 8    45.7200          :Node #, node height (cm)
 9    60.9600          :Node #, node height (cm)
10    15.2400          :Node #, node height (cm)
 583.3
 632.6
 633.9
 645.3
 662.3
 680.3
 695.6
 698.8
 668.7
 588.4
10                     :# of fuel temp axial nodes; stcalc =   8
 1    15.2400          :Node #, node height (cm)
 2    30.4800          :Node #, node height (cm)
```

```
 3    30.4800             :Node #, node height (cm)
 4    45.7200             :Node #, node height (cm)
 5    30.4800             :Node #, node height (cm)
 6    45.7200             :Node #, node height (cm)
 7    45.7200             :Node #, node height (cm)
 8    45.7200             :Node #, node height (cm)
 9    60.9600             :Node #, node height (cm)
10    15.2400             :Node #, node height (cm)
 580.4
 622.2
 623.4
 632.4
 645.5
 660.7
 677.3
 685.2
 663.0
 587.7
10                        :# of fuel temp axial nodes; stcalc =  9
 1    15.2400             :Node #, node height (cm)
 2    30.4800             :Node #, node height (cm)
 3    30.4800             :Node #, node height (cm)
 4    45.7200             :Node #, node height (cm)
 5    30.4800             :Node #, node height (cm)
 6    45.7200             :Node #, node height (cm)
 7    45.7200             :Node #, node height (cm)
 8    45.7200             :Node #, node height (cm)
 9    60.9600             :Node #, node height (cm)
10    15.2400             :Node #, node height (cm)
 568.2
 586.8
 594.2
 603.1
 610.4
 617.8
 622.3
 621.6
 608.4
 573.6
10                        :# of fuel temp axial nodes; stcalc = 10
 1    15.2400             :Node #, node height (cm)
 2    30.4800             :Node #, node height (cm)
 3    30.4800             :Node #, node height (cm)
 4    45.7200             :Node #, node height (cm)
 5    30.4800             :Node #, node height (cm)
 6    45.7200             :Node #, node height (cm)
 7    45.7200             :Node #, node height (cm)
 8    45.7200             :Node #, node height (cm)
 9    60.9600             :Node #, node height (cm)
10    15.2400             :Node #, node height (cm)
 570.0
 591.8
 594.3
 599.0
 605.6
 612.8
 619.0
 621.0
 609.7
 574.0
10                        :# of fuel temp axial nodes; stcalc = 11
 1    15.2400             :Node #, node height (cm)
 2    30.4800             :Node #, node height (cm)
 3    30.4800             :Node #, node height (cm)
```

```
 4    45.7200          :Node #, node height (cm)
 5    30.4800          :Node #, node height (cm)
 6    45.7200          :Node #, node height (cm)
 7    45.7200          :Node #, node height (cm)
 8    45.7200          :Node #, node height (cm)
 9    60.9600          :Node #, node height (cm)
10    15.2400          :Node #, node height (cm)
 567.8
 584.1
 586.3
 590.4
 596.5
 603.2
 610.7
 616.1
 609.2
 573.9
10                     :# of fuel temp axial nodes; stcalc = 12
 1    15.2400          :Node #, node height (cm)
 2    30.4800          :Node #, node height (cm)
 3    30.4800          :Node #, node height (cm)
 4    45.7200          :Node #, node height (cm)
 5    30.4800          :Node #, node height (cm)
 6    45.7200          :Node #, node height (cm)
 7    45.7200          :Node #, node height (cm)
 8    45.7200          :Node #, node height (cm)
 9    60.9600          :Node #, node height (cm)
10    15.2400          :Node #, node height (cm)
 565.8
 577.9
 580.0
 583.1
 588.2
 594.4
 602.5
 610.4
 607.9
 574.0
10                     :# of fuel temp axial nodes; stcalc = 13
 1    15.2400          :Node #, node height (cm)
 2    30.4800          :Node #, node height (cm)
 3    30.4800          :Node #, node height (cm)
 4    45.7200          :Node #, node height (cm)
 5    30.4800          :Node #, node height (cm)
 6    45.7200          :Node #, node height (cm)
 7    45.7200          :Node #, node height (cm)
 8    45.7200          :Node #, node height (cm)
 9    60.9600          :Node #, node height (cm)
10    15.2400          :Node #, node height (cm)
 564.1
 573.3
 575.5
 577.9
 582.6
 588.8
 597.7
 607.6
 608.7
 574.6
10                     :# of modd denn axial nodes; stpoint = 1
 1    15.2400          :Node #, node height (cm)
 2    30.4800          :Node #, node height (cm)
 3    30.4800          :Node #, node height (cm)
 4    45.7200          :Node #, node height (cm)
```

```
 5     30.4800             :Node #, node height (cm)
 6     45.7200             :Node #, node height (cm)
 7     45.7200             :Node #, node height (cm)
 8     45.7200             :Node #, node height (cm)
 9     60.9600             :Node #, node height (cm)
10     15.2400             :Node #, node height (cm)
0.7396
0.7396
0.6906
0.5451
0.4391
0.3740
0.3215
0.2868
0.2622
0.2529
10                         :# of modd denn axial nodes; stpoint = 2
 1     15.2400             :Node #, node height (cm)
 2     30.4800             :Node #, node height (cm)
 3     30.4800             :Node #, node height (cm)
 4     45.7200             :Node #, node height (cm)
 5     30.4800             :Node #, node height (cm)
 6     45.7200             :Node #, node height (cm)
 7     45.7200             :Node #, node height (cm)
 8     45.7200             :Node #, node height (cm)
 9     60.9600             :Node #, node height (cm)
10     15.2400             :Node #, node height (cm)
0.7396
0.7396
0.7052
0.5667
0.4540
0.3824
0.3245
0.2853
0.2560
0.2445
10                         :# of modd denn axial nodes; stpoint = 3
 1     15.2400             :Node #, node height (cm)
 2     30.4800             :Node #, node height (cm)
 3     30.4800             :Node #, node height (cm)
 4     45.7200             :Node #, node height (cm)
 5     30.4800             :Node #, node height (cm)
 6     45.7200             :Node #, node height (cm)
 7     45.7200             :Node #, node height (cm)
 8     45.7200             :Node #, node height (cm)
 9     60.9600             :Node #, node height (cm)
10     15.2400             :Node #, node height (cm)
0.7396
0.7396
0.7005
0.5667
0.4571
0.3836
0.3230
0.2814
0.2495
0.2365
10                         :# of modd denn axial nodes; stpoint = 4
 1     15.2400             :Node #, node height (cm)
 2     30.4800             :Node #, node height (cm)
 3     30.4800             :Node #, node height (cm)
 4     45.7200             :Node #, node height (cm)
 5     30.4800             :Node #, node height (cm)
```

```
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
0.7396
0.7396
0.7090
0.5845
0.4776
0.4035
0.3417
0.2994
0.2667
0.2531
10                          :# of modd denn axial nodes; stpoint = 5
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
0.7396
0.7396
0.7159
0.6006
0.4983
0.4230
0.3561
0.3080
0.2709
0.2555
10                          :# of modd denn axial nodes; stpoint = 6
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
0.7396
0.7396
0.7113
0.5955
0.4917
0.4167
0.3515
0.3053
0.2694
0.2545
10                          :# of modd denn axial nodes; stpoint = 7
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
```

```
 7     45.7200                 :Node #, node height (cm)
 8     45.7200                 :Node #, node height (cm)
 9     60.9600                 :Node #, node height (cm)
10     15.2400                 :Node #, node height (cm)
0.7396
0.7396
0.7079
0.5972
0.4982
0.4252
0.3598
0.3124
0.2749
0.2594
10                             :# of modd denn axial nodes; stpoint = 8
 1     15.2400                 :Node #, node height (cm)
 2     30.4800                 :Node #, node height (cm)
 3     30.4800                 :Node #, node height (cm)
 4     45.7200                 :Node #, node height (cm)
 5     30.4800                 :Node #, node height (cm)
 6     45.7200                 :Node #, node height (cm)
 7     45.7200                 :Node #, node height (cm)
 8     45.7200                 :Node #, node height (cm)
 9     60.9600                 :Node #, node height (cm)
10     15.2400                 :Node #, node height (cm)
0.7396
0.7396
0.7091
0.6018
0.5069
0.4377
0.3752
0.3285
0.2904
0.2735
10                             :# of modd denn axial nodes; stpoint = 9
 1     15.2400                 :Node #, node height (cm)
 2     30.4800                 :Node #, node height (cm)
 3     30.4800                 :Node #, node height (cm)
 4     45.7200                 :Node #, node height (cm)
 5     30.4800                 :Node #, node height (cm)
 6     45.7200                 :Node #, node height (cm)
 7     45.7200                 :Node #, node height (cm)
 8     45.7200                 :Node #, node height (cm)
 9     60.9600                 :Node #, node height (cm)
10     15.2400                 :Node #, node height (cm)
0.7396
0.7396
0.7100
0.6050
0.5127
0.4465
0.3868
0.3415
0.3036
0.2858
10                             :# of modd denn axial nodes; stpoint =10
 1     15.2400                 :Node #, node height (cm)
 2     30.4800                 :Node #, node height (cm)
 3     30.4800                 :Node #, node height (cm)
 4     45.7200                 :Node #, node height (cm)
 5     30.4800                 :Node #, node height (cm)
 6     45.7200                 :Node #, node height (cm)
 7     45.7200                 :Node #, node height (cm)
```

```
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
0.7396
0.7396
0.7100
0.6051
0.5130
0.4469
0.3873
0.3419
0.3041
0.2862
10                          :# of modd denn axial nodes; stpoint =11
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
0.7396
0.7396
0.7104
0.6065
0.5156
0.4506
0.3919
0.3468
0.3088
0.2905
10                          :# of modd denn axial nodes; stpoint =12
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
 9     60.9600              :Node #, node height (cm)
10     15.2400              :Node #, node height (cm)
0.7396
0.7396
0.7108
0.6077
0.5178
0.4539
0.3964
0.3522
0.3146
0.2959
10                          :# of modd denn axial nodes; stpoint =13
 1     15.2400              :Node #, node height (cm)
 2     30.4800              :Node #, node height (cm)
 3     30.4800              :Node #, node height (cm)
 4     45.7200              :Node #, node height (cm)
 5     30.4800              :Node #, node height (cm)
 6     45.7200              :Node #, node height (cm)
 7     45.7200              :Node #, node height (cm)
 8     45.7200              :Node #, node height (cm)
```