4. ORIGEN2 COMMANDS

The instructions defined in this section, called ORIGEN2 commands, enable the user to precisely define the order in which any or all of the ORIGEN2 program functions are executed. This procedure is analogous to writing a FORTRAN program in that the commands define a series of operations which will be performed sequentially, with the sequence being variable at the user's option. The use of the commands to define the ORIGEN2 problem flowsheet allows the use of a "DO loop" command, which executes a set of instructions within the range of the loop a prescribed number of times. Coupled with other options, this gives the user the capability for easily investigating fuel recycle (e.g., plutonium) and nuclear fuel cycle waste production rates as a function of time.

The general format of the ORIGEN2 commands is

COM PARM(1), PARM(2), . . . PARM(I) ,

where COM is a keyword defining the instruction type and the PARM(I) are parameters supplying various data necessary for the execution of the operational commands. Details on the data format are given in Sect. 2.2. A list of the ORIGEN2 commands and a brief description of their functions are given in Table 4.1.

Before attempting to use ORIGEN2, it should be noted that there are certain restrictions on the order in which the commands must occur. The primary restriction is that the LIB command (Sect. 4.18), which reads the decay and cross-section libraries, must precede most other commands since it defines the list of nuclides being considered. Other restrictions will be noted when the individual commands are discussed.

Each ORIGEN2 command can be present in a single input stream a maximum number of times; the limit depends on the specific command. This limit is given in the section (below) that describes each individual command. The limits can be changed by varying the dimensions of the appropriate array(s) within the ORIGEN2 source deck. The limit on the total number of ORIGEN2 commands that may be used is 300, a number which can also be changed by varying array dimensions within the source deck.

Table 4.1. List of ORIGEN2 commands

Command keyword	Description	Section	Page
ADD	Add two vectors	4.13	40
BAS	Case basis	4.3	28
BUP	Burnup calculation	4.14	42
CON	Continuation	4.28	60
CUT	Cutoff fractions for summary tables	4.9	34
DEC	Decay	4.23	54
DOL	DO loop	4.11	38
END	Terminate execution	4.30	61
FAC	Calculate a multiplication factor	4.4	28
HED	Vector headings	4.7	33
INP	Read input composition, continuous removal rate, and continuous feed rate	4.6	31
IRF	Flux irradiation	4.21	50
IRP	Specific power irradiation	4.22	52
KEQ	Match infinite multiplication factors	4.10	36
LIB	Library print control	4.18	45
LIP	Library print control	4.16	43
LPU	Data library replacement cards	4.20	49
MOV	Move nuclide composition from vector to vector	4.12	38
OPTA	Specify actinide nuclide output table options	4.26	58
OPTF	Specify fission product nuclide output table options	4.27	59
OPTL	Specify activation product output table options	4.25	56
OUT	Print calculated results	4.5	29
PCH	Punch an output vector	4.15	42
PHO	Read photon libraries	4.19	47
PRO	Reprocess fuel	4.24	55
RDA	Read comments regarding case being input	4.1	27
REC	Loop counter	4.8	34
TIT	Case title	4.2	27
WAC	Nuclide accumulation	4.17	44
GTO	GO TO	4.31	61a

4.1 RDA — Read Comments Regarding Case Being Input

A. Function: Prints alphanumeric comments among the listing of the operational commands being input.

B. Data sequence:

RDA COMMENT(S)

where

RDA = command keyword
COMMENT(S) = alphanumeric message

C. Allowable number of RDA commands: Maximum total number of commands.

D. Propagation: None.

E. Remarks: These comments are printed in the listing created when ORIGEN2 is interpreting the commands, which is separate from the card input echo described in Sect. 2.6.

4.2 TIT — Case Title

A. Function: Supplies case title printed in ORIGEN2 output.

B. Data sequence:

TIT A(9), . . . A(80)

where

TIT = command keyword
A(I) = alphanumeric characters in columns 9-80 only

C. Allowable number of TIT commands: 20

D. Propagation: Until changed.

E. Remarks: None.

4.3 BAS — Case Basis

A. Function: Supplies case basis printed in ORIGEN2 output.

B. Data sequence:

BAS A(9), . . . A(80)

where

BAS = command keyword
A(I) = alphanumeric characters in columns 9-80 only

C. Allowable number of BAS commands: 10

D. Propagation: Until changed.

E. Remarks: The BAS command only supplies an alphanumeric message. The user is responsible for the consistency of the basis, the input material masses, specific power, etc.

4.4 FAC — Calculate a Multiplication Factor Based on Total Vector Masses

A. Function: Calculates a multiplication factor, FACTOR[NFAC(1)], based on the total actinide plus fission product masses in one or two vectors for use in MOV (see Sect. 4.12) or ADD (see Sect. 4.13) commands.

B. Data sequence:

FAC NFAC(1), . . . NFAC(4), RFAC(1)

where

FAC = command keyword
NFAC(1) = number of factor calculated by this command (must be greater than zero and less than or equal to the maximum number of FAC commands)
NFAC(2) = vector number

NFAC(3) = vector number

NFAC(4) = method for calculating FACTOR[NFAC(1)]:

= 1 FACTOR[NFAC(1)] = T[NFAC(2)]+T[NFAC(3)]
= 2 FACTOR[NFAC(1)] = T[NFAC(2)]-T[NFAC(3)]
= 3 FACTOR[NFAC(1)] = T[NFAC(2)]*T[NFAC(3)]
= 4 FACTOR[NFAC(1)] = T[NFAC(2)]/T[NFAC(3)]
= 5 FACTOR[NFAC(1)] = T[NFAC(2)]
= 6 FACTOR[NFAC(1)] = T[NFAC(3)]
= 7 FACTOR[NFAC(1)] = 1.0/T[NFAC(2)]
= 8 FACTOR[NFAC(1)] = 1.0/T[NFAC(3)]

where the T[NFAC(I)] are the total fission product plus actinide masses for the indicated vectors, expressed in kilograms.

RFAC(1) = constant value to be used in place of the T[NFAC(I)]:

.GT.0 = substitute RFAC(1) for T[NFAC(2)] when calculating FACTOR[NFAC(1)]
.EQ.0 = use the T[NFAC(I)] as defined
.LT.0 = substitute [-RFAC(I)] for T[NFAC(3)] when calculating FACTOR[NFAC(1)]

The units of RFAC(1) are kilograms.

C. Allowed number of FAC commands: 20

D. Propagation: Until another FAC command with the same value of NFAC(1) is executed.

E. Remarks: Some characteristic results from this command are printed on unit 15.

4.5 OUT — Print Calculated Results

A. Function: Calls for the calculated results in some or all of the output vectors to be printed.

B. Data sequence:

OUT NOUT(1), . . . NOUT(4)

where

OUT = command keyword

NOUT(1) = number of vectors to be printed beginning with the first vector:

.GT.0 = output on units IOUT, JOUT, and KOUT (Unit 6)

.LT.0 = output on unit 11

NOUT(2) = frequency of print if instruction is in a loop (Sect. 4.11) [print occurs first time through loop and every NOUT(2)th recycle thereafter]

NOUT(3) = print number of present recycle:

.GT.0 = yes

.LE.0 = no

NOUT(4) = parameter controlling type of summary table printed:

.LT.0 = all vectors tested for inclusion in summary table except vector -NOUT(4)

.EQ.0 = all vectors tested for inclusion in summary table

.GT.0 = only vector NOUT(4) tested to see if a nuclide is included in the summary table

C. Allowable number of OUT commands: 20

D. Propagation: None.

E. Remarks:

1. If NOUT(2).NE.1, a REC command must be employed (Sect. 4.8).

4.6 INP — Read Input Composition, Continuous Removal Rate, and Continuous Feed Rate

A. Function: Calls for nuclide composition, continuous nuclide feed rate, or continuous elemental removal rate to be read.

B. Data sequence:

INP NINP(1), . . . NINP(6)

where

INP = command keyword

NINP(1) = number of vector in which initial compositions are to be stored

NINP(2) = read nuclide composition:

.EQ.0 = no

.EQ.1 = yes; units are g/basis unit (read on unit 5)

.EQ.2 = yes; units are g-atoms/basis unit (read on unit 5)

.EQ.-1 = yes; units are g/basis unit (read on unit 4)

.EQ.-2 = yes; units are g-atoms/basis unit (read on unit 4)

NINP(3) = read continuous nuclide feed rate:

.LE.0 = no

.EQ.1 = yes; units are g/(time)(basis unit)

.EQ.2 = yes; units are g-atoms/(time)(basis unit)

See NINP(5) for specification of time units.

NINP(4) = read element removal rate per unit time:

.LT.0 = no read; no propagation

.EQ.0 = no read, but propagate previously read values

.GT.0 = read NINP(4) data pairs (see Sect. 6.3)

See NINP(6) for specification of time units.

NINP(5) = time units of continuous nuclide feed rate data (see Table 4.2)

NINP(6) = time units of continuous elemental removal rate data (see Table 4.2)

Table 4.2. Time unit designation

1 = seconds

2 = minutes

3 = hours

4 = days

5 = years

6 = stable

7 = 10^3 years (kY)

8 = 10^6 years (MY)

9 = 10^9 years (GY)

C. Allowable number of INP commands: 15

D. Propagation: None.

E. Remarks: User is responsible for the consistency of the calculational basis with the input masses.

4.7 HED — Vector Headings

A. Function: Allows alphanumeric vector headings to be specified.

B. Data sequence:

HED NHED A(1), . . . A(10)

where

HED = command keyword
NHED = number of vector which is to be given heading
A(I) = ten-character alphanumeric heading anyplace on the card to the right of NHED

C. Allowable number of HED commands: 50

D. Propagation: Until the vector is overwritten.

E. Remarks:

1. The heading is moved with the vector when the MOV (Sect. 14.12) and ADD (Sect. 14.13) commands are used.
2. If a HED command is to be used to label either a vector of input concentrations [vector NINP(1), Sect. 4.6] or the vectors resulting from a PRO command [vectors NPRO(2) and NPRO(3), Sect. 4.24], the HED command must follow the INP or PRO command.
3. If A(1) is an apostrophe or asterisk (*), the ten characters immediately following A(1) are taken as the vector heading. This allows for the inclusion of leading blanks.

4.8 REC — Loop Counter

A. Function: Counts the number of times that a loop (DOL command, Sect. 14.11) has been executed.

B. Data sequence:

REC

where

REC = command keyword

C. Allowable number of REC commands: 1

D. Propagation: None.

E. Remarks:

1. This counter is output as the "Recycle #" in ORIGEN2 output.

4.9 CUT — Cutoff Fractions for Summary Tables

A. Function: Override default cutoff fractions for summary output tables.

B. Data sequence:

CUT[NCUT(1), RCUT(1)], . . . [NCUT(NT), RCUT(NT)], -1

where

CUT = operational command

NCUT(I) = number of the output table to which cutoff fraction RCUT(I) is to apply (see Table 4.3 for table numbers and descriptions)

RCUT(I) = new cutoff fraction for table number NCUT(I)

NT = total number of default cutoff values which are being overridden with this CUT command

C. Allowable number of CUT commands: 3

D. Propagation: Until changed.

Table 4.3. Description of ORIGEN2 output table

	Table number	Description of table	Units	
	1	Isotopic composition of each element	atom fraction	
	2	Isotopic composition of each element	weight fraction	
	3	Composition	gram-atoms	
	4	Composition	atom fraction	
	5	Composition	grams	
	6	Composition	weight fraction	
	7	Radioactivity (total)	Ci	
	8	Radioactivity (total)	fractional	
	9	Thermal power	watts	
	10	Thermal power	fractional	
	11	Radioactivity (total)	Bq	
add →	12	Radioactivity (total)	fractional	← add
	13	Radioactive inhalation hazard	m^3 air	
	14	Radioactive inhalation hazard	fractional	
	15	Radioactive ingestion hazard	m^3 water	
	16	Radioactive ingestion hazard	fractional	
	17	Chemical ingestion hazard	m^3 water	
	18	Chemical ingestion hazard	fractional	
	19	Neutron absorption rate	neutrons/sec	
	20	Neutron absorption rate	fractional	
	21	Neutron-induced fission rate	fissions/sec	
	22	Neutron-induced fission rate	fractional	
	23	Radioactivity (alpha)	Ci	
	24	Radioactivity (alpha)	fractional	
	25	(alpha,n) neutron production	neutrons/sec	
	26	Spontaneous fission neutron production	neutrons/sec	
	27	Photon emission rate	photons/sec	
	28	Set test parameter ERR	-	

E. Remarks:

1. If an output value for a particular nuclide is less than the cutoff fraction multiplied by the total table value for all vectors being tested (see Sect. 4.5 for additional details on which vectors are tested), then that particular nuclide is not printed.
2. Table number 28 can be used to override the default value for ERR, presently set at 1.0E-25. ERR is used in logical IF statements instead of 0.0.
3. An integer -1 must follow RCUT(NT) unless all 28 cutoff fractions are specified.
4. The default cutoff fractions for the first 26 tables (see Table 4.3) are 0.001; for Table 27 the cutoff is 0.01.
5. The [NCUT(I),RCUT(I)] may continue onto subsequent cards. No operational command is used on the additional cards.
6. The application of the cutoff value to photon tables is somewhat different; it is discussed in Sect. 8.2.2.

4.10 KEQ — Match Infinite Multiplication Factors

A. Function: Blend materials in two vectors so that the resulting infinite multiplication factor (IMF) matches that of another vector or an input value.

B. Data sequence:

```
KEQ    NKEQ(1), NKEQ(2), NKEQ(3), NKEQ(4), NKEQ(5), RKEQ(1)
```

where

KEQ = command keyword

NKEQ(1) = vector whose IMF is to be matched by vector NKEQ(4)

NKEQ(2) = vector whose material is to be wholly included in the final blended material in vector NKEQ(4)

NKEQ(3) = vector whose material is to be apportioned to obtain the proper IMF for vector NKEQ(4)

NKEQ(4) = vector containing all material in vector NKEQ(2) plus part of the material in NKEQ(3) and having the same IMF as either vector NKEQ(1) or RKEQ(1); that is,

$$NKEQ(4) = NKEQ(2) + f * NKEQ(3)$$

where f is the factor by which NKEQ(3) must be multiplied to obtain the correct IMF for NKEQ(4).

NKEQ(5) = vector containing the portion of NKEQ(3) not blended into NKEQ(4); that is,

$$NKEQ(5) = (1-f) * NKEQ(3)$$

If (1-f) is less than zero, then NKEQ(5) is set to zero.

RKEQ(1) = desired final IMF for vector NKEQ(4) if RKEQ(1).GT.0.0. If RKEQ(1).LT.0.0, the IMF of vector NKEQ(4) is matched to that of vector NKEQ(1). If RKEQ(1).EQ.0.0, the IMF is equal to RMULV(NREC,1). The RMULV values are specified in a data statement in MAIN (see Sect. 2.1); the NREC parameter is described in Sect. 4.8.

C. Allowable number of KEQ commands: 3

D. Propagation: None.

E. Remarks:

1. The equation used to calculate the parameter f, by which vector NKEQ(3) is multiplied before being combined with material in vector NKEQ(2) and being placed in vector NKEQ(4) is given by

$$f = (k_2 - k_1)*D_2/(k_1 - k_3)*D_3$$

where

k_1 = IMF to be matched from vector NKEQ(1) or RKEQ(1)

k_2 = IMF of material in vector NKEQ(2)

k_3 = IMF of material in vector NKEQ(3)

D_2 = neutron absorption rate of material in vector NKEQ(2), neutrons sec^{-1}

D_3 = neutron absorption rate of material in vector NKEQ(3), neutrons sec^{-1}

2. Some characteristic results from this command are printed on unit 15.

4.11 DOL — DO Loop

A. Function: A "DO loop" which executes the commands within its range a prescribed number of times.

B. Data sequence:

DOL NDOL(1), NDOL(2)

where

DOL = command keyword

NDOL(1) = number of the CON command (Sect. 4.28) which defines the range of this DOL. Each DOL must have a unique CON associated with it.

NDOL(2) = the total number of times the instructions within the loop are to be executed

C. Allowable number of DOL commands: 2

D. Propagation: None.

E. Remarks: None.

4.12 MOV — Move Nuclide Composition from Vector to Vector

A. Function: Moves (i.e., copies) the nuclide concentration data in one vector to another vector, nuclide by nuclide.

B. Data sequence:

MOV NMOV(1), NMOV(2), NMOV(3), RMOV(1)

where

MOV = command keyword

NMOV(1) = number of the vector where the concentrations to be moved are presently stored

NMOV(2) = number of the vector where the concentrations in vector NMOV(1) are to be moved. May be the same as NMOV(1).

NMOV(3) = source of additional multiplier

.GT.0 = number of variable multiplier vector that contains the additional factors by which vector NMOV(1) is to be multiplied before being moved to vector NMOV(2). The variable multipliers are in array RMULV and are initialized with a DATA statement in MAIN. The particular element of RMULV used is

RMULV[NREC, NMOV(3)]

where NREC is the recycle number (Sect. 4.8). The total multiplier, RMULT, is given by

RMULT = RMULV[NREC,NMOV(3)]*RMOV(1).

NREC <u>must</u> be defined to use the variable multiplier option.

.EQ.0 = no additional multiplier is used; that is,

RMULT = RMOV(1).

.LT.0 = The additional multiplier to be used was previously calculated by an FAC command (see Sect. 4.4) and designated as FACTOR[NFAC(1)] at that time. To use this factor, set NMOV(3) = -NFAC(1); the total multiplier is then given by

RMULT = FACTOR[-NMOV(3)]*RMOV(1).

RMOV(1) = factor by which vector NMOV(1) is to be multiplied before being stored in vector NMOV(2).

C. Allowable number of MOV commands: 99

D. Propagation: None.

E. Remarks:

1. Vector NMOV(2) can be zeroed by moving another vector to NMOV(2) with RMOV(1) = 0.0.
2. The information in vector NMOV(1) is not destroyed by the MOV command.
3. Vector NMOV(2) will have the same heading as vector NMOV(1) after the MOV command has been executed.

4.13 ADD — Add Two Vectors

A. Function: Adds the nuclide concentration data in one vector to that in another vector, nuclide by nuclide.

B. Data sequence:

ADD NADD(1), NADD(2), NADD(3), RADD(1)

where

ADD = operational command

NADD(1) = number of the vector where the concentrations to be added are presently stored

NADD(2) = number of the vector to which the concentrations in vector NADD(1) are to be added

NADD(3) = source of additional multiplier

.GT.0 = if NADD(3).GT.0, it is the number of the variable multiplier vector which contains the factors by which vector NADD(1) is tq be multiplied before being added to vector NADD(2). The variable multipliers are in array RMULV and are initialized with a DATA statement in MAIN. The particular element of RMULV used is

RMULV[NREC, NADD(3)]

where NREC is the recycle number (see Sect. 4.8). The total multiplier, RMULT, is given by

RMULT = RMULV[NREC,NADD(3)]*RADD(1)

NREC must be defined to use this option (see Sect. 4.8).

.EQ.0 = no additional multiplier used; that is,

RMULT = RADD(1).

.LT.0 = the additional multiplier to be used was previously calculated by a FAC command (see Sect. 4.4) and designated as FACTOR[NFAC(1)]. To use this factor, set NADD(3) = -NFAC(1); the total multiplier is then given by

RMULT = FACTOR[-NADD(3)]*RADD(1)

RADD(1) = factor by which vector NADD(1) is to be multiplied before being added to vector NADD(2) or as specified under NADD(3) above.

C. Allowable number of ADD commands: 30

D. Propagation: None.

E. Remarks:

1. Vector NADD(1) may be subtracted from vector NADD(2) by setting RADD(1) = -1.0. (CAUTION: Negative nuclide concentrations can result in fatal errors.)
2. The information in vector RADD(1) is not altered by the ADD command.
3. Vector NADD(2) will have the same headings as vector NADD(1) after the ADD command has been executed.

4.14 BUP — Burnup Calculation

A. Function: Defines the basis and calculates the average burnup, flux, and specific power for an irradiation.

B. Data sequence:

```
BUP
   Irradiation
BUP
```

where

BUP = command keyword

Irradiation = the operational commands (generally several IRPs or IRFs) that describe the fuel irradiation upon which the burnup calculation is to be based.

C. Allowable number of BUP commands: 20 (ten pair).

D. Propagation: Until superseded by other BUP commands.

E. Remarks:

1. A BUP command must appear both before and after the statements constituting the fuel irradiation upon which the burnup calculation is to be based. Other commands may be present between the BUP statements.

4.15 PCH — Punch an Output Vector

A. Function: Punch a designated output vector in ORIGEN2-readable format or write it to a disk file.

B. Data sequence:

```
PCH     NPCH(1), NPCH(2), NPCH(3)
```

where

PCH = command keyword

NPCH(1) = control character for light nuclide and structural material punch

NPCH(2) = control character for actinide nuclide punch

NPCH(3) = control character for fission product nuclide punch

If NPCH(I) .EQ.0 - no punch
.GT.0 - number of output vector to be punched
.LT.0 - number of storage vector to be punched

C. Allowable number of PCH commands: 54

D. Propagation: None.

E. Remarks:

1. Format of punched output is [2X,I2,4(1X,I6,2X,1PE10.4)]; see Sect. 6.1 for details.

2. Units of punched output are g-atoms.

3. The last record (card) written by each PCH command is

    0 BURNUP FLUX SPECIFIC POWER.

   The burnup, flux, and specific power are average values produced by the BUP command (Sect. 4.14) and <u>must</u> be present for a file read on unit 4 [NINP(2).LT.0; see Sect. 4.6]. These parameters are not necessary for input material compositions read with NINP(2).GT.0.

4.16 LIP _ Library Print Control

A. Function: Controls the printing of the input data libraries.

B. Data sequence:

LIP NLIP(1), NLIP(2), NLIP(3)

where

LIP = command keyword

NLIP(1) = control character for decay library print

NLIP(2) = control character for cross-section library print

NLIP(3) = control character for photon library print

If NLIP(I).EQ.0 - no print
.GT.0 - print library

C. Allowable number of LIP commands: 5

D. Propagation: Until superseded.

E. Remarks: None.

4.17 WAC — Nuclide Accumulation

A. Function: Multiplies a concentration vector by a fractional recovery vector and stores the result in vector B, which contains continuous feed rates.

B. Data sequence:

WAC NWAC(1), NWAC(2)

where

WAC = command keyword

NWAC(1) = number of fractional recovery vector (Sects. 3.4 and 3.5) which is to multiply concentration vector NWAC(2). Fractional recovery NWAC(1) should contain the removal rate of each element from the system in units of $\sec^{-1}$ (equivalent to the feed rate to the next system being analyzed).

NWAC(2) = number of concentration vector which is to be multiplied by fractional recovery vector NWAC(1)

C. Maximum allowable number of WAC commands: 2

D. Propagation: None.

E. Remarks:

1. This command will enable the continuous accumulation of waste from a reactor with continuous reprocessing (e.g., an MSBR) to be calculated. The steady-state fuel composition in vector NWAC(2) is multiplied by the appropriate continuous removal rates stored in fractional recovery vector NWAC(1); the result is subsequently stored in vector B. Then the waste is decayed,

with vector B representing the continuous feed of waste to the waste decay step from the continuously reprocessed steady-state reactor.

4.18 LIB — Read Decay and Cross-Section Libraries

A. Function: Read decay and cross-section libraries; substitute decay and cross-section cards and cards with non-standard reactions.

B. Data sequence:

```
LIB     NLIB(1), . . . NLIB(11)
```

where

LIB = command keyword

NLIB(1) = control character for printing matrix of non-zero reaction rates (array A) for the libraries read (see Sect. 8.2.1).
If NLIB(1).GT.0 - print
.LE.0 - no print

NLIB(2) = identification number of light nuclide decay library to be read; see Table 4.4

NLIB(3) = identification number of actinide nuclide decay library to be read; see Table 4.4

NLIB(4) = identification number of fission product nuclide decay library to be read; see Table 4.4

NLIB(5) = identification number of light nuclide cross-section library to be read; see Table 4.4

NLIB(6) = identification number of actinide nuclide cross-section library to be read; see Table 4.4

NLIB(7) = identification number of fission product nuclide yield and cross-section library to be read; see Table 4.4
If NLIB(2-7).EQ.0 - no read
.GT.0 - normal read on unit NLIB(8)

Table 4.4. Numbers of ORIGEN2 data libraries

Type of library	Category of isotope			
	Activation product [NLIB(2 or 5)][a]	Actinide [NLIB(3 or 6)][a]	Fission product [NLIB(4 or 7)][a]	NLIB(12)[a]
Decay	1	2	3	
Photon	101	102	103	
	Cross-section libraries			
PWR: ^{235}U-enriched UO_2: 33,000 MWd/metric ton	204	205	206	1
PWR: ^{235}U-enriched UO_2 in a self-generated Pu recycle reactor	207	208	209	2
PWR: Pu-enriched UO_2 in a self-generated Pu recycle reactor	210	211	212	3
BWR: ^{235}U-enriched UO_2	251	252	253	4
BWR: ^{235}U-enriched fuel in a self-generated Pu recycle reactor	254	255	256	5
BWR: Pu-enriched fuel in a self-generated Pu recycle reactor	257	258	259	6
PWR: ThO_2-enriched with denatured ^{233}U	213	214	215	7
PWR: Pu-enriched ThO_2	216	217	218	8
PWR: ^{235}U-enriched UO_2; 50,000 MWd/metric ton	219	220	221	9
PWR: ThO_2-enriched with makeup, denatured ^{235}U	222	223	224	10
PWR: ThO_2 enriched with recycled, denatured ^{233}U	225	226	227	11

Table 4.4 (continued)

Type of library	Category of isotope			
	Activation product [NLIB(2 or 5)][a]	Actinide [NLIB(3 or 6)][a]	Fission product [NLIB(4 or 7)][a]	NLIB(12)[a]
LMFBR: Early oxide, LWR-Pu/U/U/U				
Core	301	302	303	18
Axial blanket	304	305	316	19
Radial blanket	307	308	309	20
LMFBR: Advanced oxide, LWR-Pu/U/U/U				
Core	311	312	313	12
Axial blanket	314	315	316	13
Radial blanket	317	318	319	14
LMFBR: Advanced oxide, recycle-Pu/U/U/U				
Core	321	322	323	15
Axial blanket	324	325	326	16
Radial blanket	327	328	329	17
LMFBR: Advanced oxide, LWR-Pu/U/U/Th				
Core	331	332	333	32
Axial blanket	334	335	336	33
Radial blanket	337	338	339	34
LMFBR: Advanced oxide, LWR-Pu/Th/Th/Th				
Core	341	342	343	29
Axial blanket	344	345	346	30
Radial blanket	347	348	349	31
LMFBR: Advanced oxide, recycle ^{233}U/Th/Th/Th				
Core	351	352	353	35
Axial blanket	354	355	356	36
Radial blanket	357	358	359	37
LMFBR: Advanced oxide, 14% denatured ^{233}U/Th/Th/Th				
Core	361	362	363	23
Axial blanket	364	365	366	24
Radial blanket	367	368	369	25
LMFBR: Advanced oxide, 44% denatured ^{233}U/Th/Th/Th				
Core	371	372	373	26
Axial blanket	374	375	376	27
Radial blanket	377	378	379	28
LMFBR: FFTF Pu/U	381	382	383	0
Thermal: 0.0253-eV cross sections	201	202	203	0

[a] Refer to Sect. 4.18 for the use of these parameters.

.LT.0 - normal read on unit NLIB(8) and substitute card read on unit NLIB(9)

NLIB(8) = number of input unit for normal reading of the bulk of the libraries

NLIB(9) = number of input unit for reading substitute cards

NLIB(10) = number of non-standard reactions to be read

If NLIB(10).EQ.0 - no read

.GT.0 - non-standard reactions read on unit NLIB(8)

.LT.0 - non-standard reactions read on unit NLIB(9)

NLIB(11) = control character identifying the set of actinides with direct fission product yields; see Table 4.5

NLIB(12) = control character identifying the set of variable actinide cross sections to be used; see Table 4.4

C. Allowable number of LIB commands: 5

D. Propagation: Until another set of decay libraries is read.

E. Remarks:

1. If substitute cards are to be read, the LPU command(s) (Sect. 4.20) must precede the LIB command in which the cards are to be read.
2. See Sect. 5 for library format details.

4.19 PHO — Read Photon Libraries

A. Function: Read the photon production rate per disintegration in 18 energy groups.

Table 4.5. Actinide sets with direct fission product yields

NLIB(11)	Actinides with direct fission product yields
1	^{235,238}U, ^{239,241}Pu
2	^{232}Th, ^{233,235}U, ^{239}Pu
3	^{232}Th, ^{233,235,238}U, ^{239,241}Pu
4	^{232}Th, ^{233,234,238}U, ^{239,241}Pu, ^{245}Cm, ^{252}Cf

B. Data sequence:

```
PHO     NPHO(1), . . . NPHO(4)
```

where

PHO = command keyword

NPHO(1) = identification number of activation product photon library to be read; see Table 4.4

NPHO(2) = identification number of actinide nuclide photon library to be read; see Table 4.4

NPHO(3) = identification number of fission product nuclide photon library to be read; see Table 4.4
If NPHO(1-3).LE.0 - no read
                .GT.0 - read

NPHO(4) = number of input unit on which the photon libraries are to be read

C. Allowable number of PHO commands: 5

D. Propagation: Until another set of photon libraries is read.

E. Remarks: See Sect. 5.5 for library format details.

4.20 LPU — Data Library Replacement Cards

A. Function: Read nuclide identifiers for replacement decay and/or cross-section data cards to be read by LIB command (Sect. 4.18).

B. Data sequence:

```
LPU     NLPU(1), . . . NLPU(MAX), -1
```

where

LPU = command keyword

NLPU(1-MAX) = nuclide identifiers for replacement data cards in the order in which they occur in the original data library

MAX = number of nuclide identifiers to be read for a given LPU command; must be .LE.100

C. Allowable number of LPU cards: 9

D. Propagation: Until another LIB command is executed.

E. Remarks:

1. If less than 100 nuclide identifiers are specified, a -1 (integer) must appear after the last identifier.
2. As many cards may be used as are required.
3. The LPU command(s) must precede the LIB command in which the replacement data cards will be read.
4. The first LPU command is associated with the first negative control variable in the NLIB(2-7) set of control variables (Sect. 4.18). The second LPU command is associated with the second negative control variable in the NLIB(2-7) set of control variables, etc.
5. See Sects. 5.1 and 5.2 for library format details.

4.21 IRF — Flux Irradiation

A. Function: Irradiation for a single interval with the neutron flux specified.

B. Data sequence:

IRF RIRF(1), RIRF(2), NIRF(1). . . . NIRF(4)

where

IRF = command keyword

RIRF(1) = time at which this irradiation interval ends

RIRF(2) = if RIRF(2).GT.0.0, this is the neutron flux during this irradiation interval in neutrons cm^{-2} sec^{-1}. If RIFR(2).LT.0.0, the neutron flux is given by:

$$NEWFLUX = OLDFLUX*[-RIRF(2)]$$

where

NEWFLUX = flux to be used during this interval, neutrons cm^{-2} sec^{-1}

OLDFLUX = flux for the same time period from the previous irradiation, neutrons cm^{-2} sec^{-1}. See remark 2 below.

NIRF(1) = number of the vector where the material composition at the beginning of this irradiation interval is stored

NIRF(2) = number of the vector where the material composition at the end of this irradiation interval is to be stored

NIRF(3) = time units of RIRF(1); see Table 4.2

NIRF(4) = specification of time at which this irradiation interval begins:

0 = starting time is the end of the previous IRF, IRP, or DEC interval. All reactivity and burnup information is retained, and MIRR is not altered. Used for continuing irradiation/decay on the same output page.

1 = starting time is set to zero. All reactivity and burnup information is retained, and MIRR is set to zero. Used for beginning a new irradiation on the same output page.

2 = starting time is set to zero. All reactivity and burnup information and MIRR are set to zero. Used to begin a new irradiation/decay on a new output page.

3 = same as NIRF(4) = 0 except that the first seven lines of the irradiation information are set to zero. Used for continuing irradiation to a new output page.

4 = same as NIRF(4) = 1 except that the first seven lines of the reactivity and burnup information are set to zero. Used to begin the decay following irradiation on a new output page while retaining the average irradiation parameters.

C. Allowable number of IRF commands: See remark 1 below.

D. Propagation: None.

E. Remarks

1. The total number of IRF + IRP + DEC commands must be .LE.150.
2. For this option to be used, the time steps for the current irradiation and decay sequence must correspond exactly to those in the previous sequence. The fluxes from the previous irradiation are not altered if [-RIFF(2)] is less than zero.
3. The "reactivity and burnup information" referred to in NIRF(4) consists of seven lines of data characteristic of an individual vector (e.g., time, infinite multiplication factor, neutron flux) and three lines containing irradiation parameters (e.g., burnup) averaged over the range of the BUP commands (Sect. 4.14). Also, see Sect. 8.2.2.
4. Internal ORIGEN2 parameters related to the flux/specific power calculations are printed on unit 15 (see Sect. 8.2.1).

4.22 IRP _ Specific Power Irradiation

A. Function: Irradiation for a single interval with the specific power specified.

B. Data sequence:

IRP RIRP(1), RIRP(2), NIRP(1), . . . NIRP(4)

where

IRP = command keyword

RIRP(1) = time at which this irradiation interval ends

RIRP(2) = power level during this irradiation interval

.GT.0 = MW(t) per unit of fuel input

.LT.0 = the power is given by:

NEWPOWER = OLDPOWER*[-RIRP(2)]

where

NEWPOWER = power to be used during this interval, MW(+). See remark 2 below.

NIRP(1) = number of the vector where the material composition at the beginning of this irradiation interval is stored

NIRP(2) = number of the vector where the material composition at the end of this irradiation interval is to be stored

NIRP(3) = time units of RIRP(1); see Table 4.2

NIRP(4) = specification of the time at which this irradiation interval begins:

0 = starting time is the end of the previous IRF, IRP, or DEC interval. All reactivity and burnup information is retained, and MIRR is not altered. Used for continuing irradiation/decay on the same output page.

1 = starting time is set to zero. All reactivity and burnup information is retained, and MIRR is set to zero. Used for beginning a new irradiation on the same output page.

2 = starting time is set to zero. All reactivity and burnup information and MIRR are set to zero. Used to begin a new irradiation/decay on a new page.

3 = same as NIRP(4) = 0 except that the first seven lines of the irradiation information are set to zero. Used for continuing irradiation to a new output page.

4 = same as NIRP(4) = 1 except that the first seven lines of the reactivity and burnup information are set to zero. Used to begin the decay following irradiation on a new output page while retaining the average irradiation parameters.

C. Allowable number of IRP commands: See remark 1 below.

D. Propagation: None.

E. Remarks:

1. The total number of IRF + IRP + DEC commands must be .LE.150.

2. For this option to be used, the time steps for the current irradiation and decay sequence must correspond exactly to those in the previous sequence. The powers from the previous irradiation are not altered if [-RIRP(2)] is less than zero.

3. The "reactivity and burnup information" referred to in NIRP(4) consists of seven lines of data characteristic of an individual vector (e.g., time, infinite multiplication factor, neutron flux) and three lines containing irradiation parameters (e.g., burnup) averaged over the range of the BUP commands (Sect. 4.14).

4. Internal ORIGEN2 parameters related to the flux/specific power calculations are printed on unit 15 (see Sect. 8.2.1).

4.23 DEC — Decay

A. Function: Decay for a single interval.

B. Data sequence:

DEC   DEC(1), NDEC(1), . . . NDEC(4)

where

DEC = operational command

DEC(1) = time at which this decay interval ends

NDEC(1) = number of the vector where the material composition at the beginning of this decay interval is stored

NDEC(2) = number of the vector where the material composition at the end of this decay interval is stored

NDEC(3) = time units of DEC(1); see Table 4.2

NDEC(4) = specification of the time at which this decay interval begins:

0 = starting time is the end of the previous IRF, IRP, or DEC interval. All reactivity and burnup information is retained, and MIRR is not altered. Used for continuing irradiation/decay on the same output page.

1 = starting time is set to zero. All reactivity and burnup information is retained, and MIRR is set to zero. Used for beginning a new irradiation on the same output page.

2 = starting time is set to zero. All reactivity and burnup information and MIRR are set to zero. Used to begin a new irradiation/decay on a new output page.

3 = same as NDEC(4) = 0 except that the first seven lines of the reactivity and burnup information are set to zero. Used for continuing irradiation to a new output page.

4 = same as NDEC(4) = 1 except that the first seven lines of the reactivity and burnup information are set to zero. Used to begin the decay following irradiation on a new output page while retaining the average irradiation parameters.

C. Allowable number of DEC commands: See below.

D. Propagation: None.

E. Remarks:

1. The total number of IRF + IRP + DEC commands must be .LE.150.
2. The "reactivity and burnup information" referred to in NDEC(4) consists of seven lines of data characteristic of an individual vector (e.g. time, infinite multiplication factor, neutron flux) and three lines containing irradiation parameters (e.g., burnup) averaged over the range of the BUP commands (Sect. 4.14).

4.24 PRO — Reprocess Fuel

A. Function: Reprocess fuel into two product compositions.

B. Data sequence:

PRO NPRO(1), . . . NPRO(4)

where

NPRO(1) = number of the vector where the material composition that is to be reprocessed is stored

NPRO(2) = number of the vector where the material that is recovered is to be stored. The amount of an isotope of element NE recovered is given by:

[Mass of isotope NE][f(NPRO(4)].

The fraction f[NPRO(4)] is the fractional recovery of element NE specified by variable NRPO(4) below. See also Sects. 3.4 and 3.5.

NPRO(3) = number of the vector where the material not recovered is to be stored. The amount of an isotope of element NE not recovered is given by:

[Mass of isotope NE][1.0 - f(NPRO(4))].

NPRO(4) = number of the set of fractional recoveries which is to be used in this reprocessing operation. If NPRO(4) is greater than zero, individual fractional recoveries (Sect. 3.4) are to be used. If NPRO(4) is less than zero, group fractional recoveries are to be used (Sect. 3.5).

C. Allowable number of PRO commands: 20

D. Propagation: None.

E. Remarks: None.

4.25 OPTL — Specify Activation Product Output Options

A. Function: Specifies which output table types (nuclide, element, or summary) are to be printed for the activation products.

B. Data sequence:

OPTL NOPTL(1), . . . NOPTL(24)

where

OPTL = command keyword

NOPTL(I) = control character indicating which output table types are to be printed for the activation products; see Table 4.6

I = table number; see Table 4.3 for output table description

C. Allowable number of OPTL commands: 20

D. Propagation: Until changed.

Table 4.6. Specification of output table types to be printed

NOPTL(I)
NOPTA(I)
NOPTF(I)
	Table type printed		

	Nuclide	Element	Summary
1	Yes	Yes	Yes
2	Yes	Yes	No
3	Yes	No	Yes
4	No	Yes	Yes
5	Yes	No	No
6	No	Yes	No
7	No	No	Yes
8	No	No	No

E. Remarks:

1. The NOPTL(I) must all be on a single card.
2. If NOPTL(1) is less than 1, only a summary grams table is printed for <u>all</u> nuclides (including actinides and fission products) until new commands (after an STP, Sect. 4.29) are read.
3. Only the first 24 tables in Table 4.3 are controlled by the OPTL command.

4.26 OPTA — Specify Options for Actinide Nuclide Output Table

A. Function: Specifies which output table types (nuclide, element, or summary) are to be printed for the actinide nuclides.

B. Data sequence:

OPTA NOPTA(1), . . . NOPTA(24)

where

OPTA = command keyword

NOPTA(I) = control character indicating which output table types are to be printed for the actinide nuclides; see Table 4.6

I = table number; see Table 4.3 for output table description

C. Allowable number of OPTA commands: 20

D. Propagation: Until changed.

E. Remarks:

1. The NOPTA(I) must all be on a single card.
2. If NOPTA(1) is less than 1, only a summary grams table is printed for <u>all</u> nuclides (including activation and fission products) until new commands (after an STP, Sect. 4.29) are read.
3. Only the first 24 tables in Table 4.3 are controlled by the OPTA command.

4.27 OPTF — Specify Options for Fission Product Nuclide Output Table

A. Function: Specifies which types of output tables (nuclide, element, or summary) are to be printed for fission product nuclide

B. Data sequence:

```
OPTF     NOPTF(1), . . . NOPTF(24)
```

where

OPTF = command keyword

NOPTF(I) = control character indicating which output table types are to be printed for the fission product nuclides; see Table 4.6

I = table number; see Table 4.3 for output table description

C. Allowable number of OPTF commands: 20

D. Propagation: Until changed.

E. Remarks:

1. The NOPTF(I) must all appear on a single card.
2. If NOPTF(1) is less than 1, only a summary grams table is printed for all nuclides (including activation products and actinides) until new commands (after an STP, Sect. 4.29) are read.
3. Only the first 24 tables in Table 4.3 are controlled by the OPTF command.

4.28 CON — Continuation

A. Function: Defines the ranges of the DOL command (Sect. 4.11) or GTO command (Sect. 4.31).

B. Data sequence:

CON NCON

where

CON = command keyword

NCON = number of this CON command; must be equal to NDOL(1) for the DOL command which is to be associated with this CON command

C. Allowable number of CON commands: 20

D. Propagation: None.

E. Remarks:

1. There must be one, and only one, CON command for each DOL command.
2. If the DOL command is removed, the corresponding CON command must also be removed.

4.29 STP — Execute Previous Commands and Branch

A. Function: Execute the set of commands preceding the STP command. Then read and execute more commands.

B. Data sequence:

STP NSTP

where

STP = command keyword

NSTP = branching control character:

1 = read new miscellaneous initialization data (Sect. 3) and a new set of commands (Sect. 4), and execute them.

2 = read a new set of commands (Sect. 4) and execute them.

3 = execute the preceding set of commands again. Additional input data (libraries and initial nuclide concentrations) will be required.

4 = terminate execution (same as END).

C. Allowable number of STP commands: Unlimited.

D. Propagation: None.

E. Remarks: None.

4.30 END — Terminate Execution

A. Function: Terminate execution.

B. Data sequence:

END

where

END = command keyword

C. Allowable number of END commands: 1

D. Propagation: None.

E. Remarks: None.

4.31 GTO — Go To a Group of Instructions and Execute

A. Function: Indicates a range of instructions that should be executed and a flux/power multiplier for this range

B. Data Sequence:

GTO   NGTO(1)   NGTO(2)   RGTO

where

GTO = command keywork

NGTO(1) = number of CON command (Sect. 4.28) that immediately precedes the group of instructions to be executed (if GT.0) or that this command is the last to be executed (if LT.0)

NGTO(2) = number of CON command that immediately follows the group of instructions to be executed

RGTO = parameter by which any fluxes or powers in the group of instructions to be executed will be multiplied; RGTO does not alter the value of fluxes/powers stored for future use

C. Allowable number of GTO commands: 10

D. Propagation: None.

E. Remarks:

1. Following the execution of the group of instructions defined by the GTO instruction, control is returned to the instruction immediately following the GTO.