



KR0000250

KAERI/TR-1543/2000

# MELCOR 코드 구조 분석

## Analysis of MELCOR Code Structure

2000년 4월

한국 원자력 연구소

**KOREA ATOMIC ENERGY RESEARCH INSTITUTE**

**Please be aware that all of the Missing Pages in this document were originally blank pages**

## 제 출 문

한국원자력연구소장 귀하

본 보고서를 1999년도 "중대사고 위해도 완화전략 분석 종합전산코드 개발" 과제의 기술보고서로 제출합니다.

2000 년 4 월

보 고 서 명 : MELCOR 코드 구조 분석  
(Analysis of MELCOR Code Structure)

주 저 자 : 김동하  
공 저 자 : 박선희

## 요 약

MELCOR 프로그램은 크게 두 부분으로 실행되는데, 대부분의 입력이 처리되고 재계산 파일에 저장되는 MELGEN 부분과, MELGEN에 의해 생성된 파일을 이용하여 실제로 시간에 따른 중대사고 현상을 모사하는 MELCOR 부분이다. 특히 MELCOR 부분은 MELGEN으로부터의 재계산 파일을 읽고 계산을 준비하는 MEXSET와 사고 진행에 동반되는 물리 현상을 모사하는 MEXRUN으로 구분된다. 또한, MELCOR 프로그램은 실행 단계, 데이터 베이스 처리 단계, 현상 해석 (현상 팩키지) 단계, 그리고 다목적 이용 프로그램 등 4개의 수직 단계로 구성된 체계적인 모듈 구조를 가지고 있어 새로운 모델을 추가하거나 교체하기가 용이한 장점을 가지고 있다. 코드에 사용되는 부프로그램들은 MELCOR 프로그램을 구성하고 있는 24개의 팩키지에 포함되어 있고, 소속된 팩키지를 나타내기 위해 팩키지의 앞 두, 세자를 이용하여 명명하며, 관련 common block들도 동일한 방법으로 표시하고 있다. 현재 수행되고 있는 중대사고 해석 종합 전산 코드 개발에 필요한 모델 추가 및 개선에 활용하기 위해, MELGEN과 MELCOR에서의 전반적인 데이터 흐름과 관련된 세부 부프로그램들을 위의 4단계로 구분하여 MELGEN과 MELCOR의 기능별로 파악하여 정리하였다.

## SUMMARY

MELCOR executes in two parts. The first is a MELGEN program, in which most of the input is specified, processed, and checked. The second part of MELCOR is the MELCOR program itself, which advances the program through time based on the database generated by MELGEN and any additional MELCOR input. In particular, MELCOR execution involves two steps: (1) a setup mode in MEXSET, during which the database is read from the restart file and any additional input is processed, and (2) a run mode in MEXRUN, which advances the simulation through time, updating the time-dependent portion of the database each cycle. MELGEN and MELCOR share a structured and modular architecture that facilitates the incorporation of additional or alternative phenomenological modes. This structure consists of four primary levels: executive level, database manager routine level, package level, and utility level. MELCOR is composed of 24 different packages, each of which models a different portion of the accident phenomenology or program control. To identify the relation of the MELCOR subroutines with the packages, first two or three letters of the package's name are duplicated in the name of the subroutines. The same rule applies to the naming of the common block. Data flows and the specific subroutines in the MELGEN and MELCOR are analyzed by their functions according to the hierarchy of four levels for model improvement and replacement during the integral code development project.

## TABLE OF CONTENTS (1/2)

제 출 문 .....	I
요 약 .....	II
SUMMARY .....	III
TABLE OF CONTENTS .....	IV
표 목차 .....	VI
그림 목차 .....	VII
1. 서론 .....	1
2. MELCOR 프로그램 전반적 구조 분석 .....	5
3. MELGEN 구조 .....	8
3.1 데이터 초기화 및 파일 열기 .....	9
3.2 입력 선처리 (PRE-PROCESS) 및 실행 입력 수행 .....	10
3.3 민감도 계수 입력 처리 .....	11
3.4 데이터베이스 구축 .....	11
3.5 편집 파일 작성 .....	12
3.6 재계산 파일 작성 .....	12

## TABLE OF CONTENTS (2/2)

4.	MELCOR 구조 .....	14
4.1	MEXSET 구조 .....	15
4.1.1	데이터 초기화 및 파일 열기 .....	16
4.1.2	입력 선처리 (Pre-process) 및 실행 입력 수행 .....	16
4.1.3	재계산 파일 읽기 및 민감도 계수 수정 .....	17
4.1.4	팩키지 입력 검토 .....	17
4.1.5	데이터 베이스 검토 .....	17
4.1.6	입력 반복 및 편집 .....	18
4.2	MEXRUN 구조 .....	18
4.2.1	재계산 파일, 편집 및 그림 작성 .....	19
4.2.2	계산 시간 간격 결정과 물리적 현상 모사 .....	19
5.	노심 모듈 구조 .....	21
5.1	노심 팩키지 (COR PACKAGE) 구조 .....	21
5.2	다른 팩키지와의 연계 .....	23
5.3	노심 팩키지와 CVH 팩키지 관련 부프로그램 .....	24
6.	결론 .....	25
7.	참고문헌 .....	27

## 표 목차

Table 2-1	Files Used by MELGEN and MELCOR (1/2).....	28
Table 2-2	MELCOR Packages (1/3).....	30
Table 3-1	Data Flow of MELGEN Main Program.....	33
Table 3-2	Data Flow of MEGRUN in MELGEN.....	34
Table 3-3	List of Subroutines called from MELGEN (1/9).....	35
Table 4-1	Data Flow in MELCOR Main Program.....	44
Table 4-2	List of Subroutines called from MELCOR (1/7).....	45
Table 4.1-1	Data Flow of MEXSET in MELCOR.....	52
Table 4.2-1	Data Flow of MEXRUN in MELCOR.....	53
Table 4.2.2-1	Data Flow of MEXLOP in MEXRUN (1/2).....	54
Table 5.1-1	Data Flow of CORDBC in MEXLOP.....	56
Table 5.1-2	Data Flow of CORDBD in CORDBC.....	57
Table 5.1-3	Data Flow of CORR1 in CORDBD (1/3).....	58
Table 5.3-1	MELCOR COR Package Subroutines (1/10).....	61
Table 5.3-2	MELCOR CVH Package Subroutines (1/5).....	71



## 그림 목차

Figure 2-1	MELCOR Code and File Relations .....	76
Figure 2-2	Inreractions Between Physics Packages .....	77
Figure 3-1	Function of MELGEN and Its Related Subroutines .....	78
Figure 3.1-1	Data Initialization and File Opening in MELGEN .....	79
Figure 3.2-1	Input Processing in MELGEN .....	80
Figure 3.3-1	Sensitivity Coefficient Check in MELGEN.....	81
Figure 3.4-1	Database Generation in MELGEN .....	82
Figure 3.5-1	EDIT Generation in MELGEN .....	83
Figure 3.6-1	Restart File Generation in MELGEN.....	84
Figure 4-1	Two Functions of MELCOR .....	85
Figure 4.1-1	Functions of MELGEN and Its Related Subroutines.....	86
Figure 4.1.1-1	Data Initialization and File Opening in MEXSET .....	87
Figure 4.1.2-1	Input Processing in MEXSET .....	88
Figure 4.1.3-1	Restart File Reading in MEXSET.....	89
Figure 4.1.4-1	Package Input Reading in MEXSET .....	90
Figure 4.1.5-1	Database Check in MEXSET .....	91
Figure 4.1.6-1	Input Echo and EDIT Generation in MEXSET .....	92
Figure 4.2-1	Functions of MEXRUN in MEXSET .....	93
Figure 4.2.1-1	Restart, EDIT and Plots Generation in MEXRUN .....	94
Figure 4.2.2-1	Timestep Control and Physics Advancement in MEXRUN.....	95

## 1. 서론

1979년 TMI에서의 원자력 발전소 사고로, 상상으로도만 그쳤던 중대사고에 대한 인식이 세계적으로 높아짐에 따라 미국을 포함한 세계 여러 나라에서는 중대사고에 대한 실험 및 해석 코드 개발 프로그램에 착수하였고, 체르노빌 사고이후 더욱 더 원자력 발전소의 중대사고에 대한 안전성 확보에 관심을 갖게 되었다. 미국에서는 원자력 규제위원회(NRC)의 후원으로 국립연구소가 중심이 되어 상세 코드(SCDAP/RELAP5, CONTAIN 등)와 종합코드(MELCOR)를 개선하고 있으며, 산업계에서는 전력연구원(EPRI)을 중심으로 MAAP4 전산코드를 개발하고 있다. 가까운 일본의 경우 주로 미국에서 개발된 중대사고 해석용 전산코드 들의 검증/평가와 아울러 자체적으로 THALES/ART 등의 코드를 개발하여 중대사고 해석 및 원전 안전성 평가에 사용하고 있다. 프랑스에서도 ICARE 전산코드를 자체 개발하여 PHEBUS 실험결과와 비교하며 코드를 개선 중에 있고, 캐나다는 CANDU 발전소에서의 중대사고 해석을 위해 MAAP-CANDU를 비롯한 중대사고 분석용 종합코드를 개발하고 있다.

이와는 달리 국내에서의 전산 코드 개발은 주로 발전소 설계 국산화와 연관하여 설계 기준 사고 중심으로 진행되어 왔고, 중대사고에 대한 연구는 외국의 중대사고 해석 코드를 들여와 우리 시스템에 설치하고 우리 목적에 맞도록 사용하는데 치중하였다. 그러한 가운데서도 중대사고 개별 현상 모델 개발과 관련하여 증기폭발 모델, 화염 전파모델, MCCI 모델, 그리고 독립적인 수소연소분석 모델 및 원자로 하부 파손 기구 분석 모델들을 개선하였으며[1], 월성 2/3/4호기 원자력 발전소의 중대사고 해석을 위해 ISAAC 전산코드를 개발하였다 [2]. 그러나 중대사고와 관련한 자체적인 대형 코드 개발 능력은 아직 걸음마 단계에 머물고 있다.

MAP4 전산코드는 여러 가지 장점에도 불구하고, 1차 계통을 사용자가 재구성 할 수 없고, 만일 바뀔 필요가 있는 경우에는 프로그램 자체를 수정해야 하기때문에 코드의 사용권 문제를 차지하고라도 기존 코드의 어려움이 확인되었다. 한편, MELCOR는 계통 모사의 유연성 때문에 사용

후로 사용되고 있다. 현상 해석을 위한 CONTAIN들이 있으나, 위의 두 코드를 보완하는 범위에서 위해 노선 내에서의 현상 해석에 국한된 SCDAP/RELAP5와 격납건물에서의 응대사고 현상 해석에 이용되고 있다. 이 외에 좀더 상세한 현상 해석을 산업체에서 확률론적 안전성 해석과 자세대 원전의 설계 지원에 필요한 한국전력공사 전력연구원 (KEPRI)과 한국전력기술 주식회사 (KOPIC) 등의 (KAERI)와 한국원자력안전기술원 (KINS)에서 활용되고 있으며, MAP은 검토하였다 [3]. MELCOR는 주로 국립 연구 기관인 한국원자력연구소 코드를 선정하기 위해 국내에서 널리 사용되는 MAP4와 MELCOR를 그리고 최신 모델의 포함 여부 등이 있다. 이러한 요건에 적합한 기준 탄젠트 거동 모사, 사용자 편의성, 개선 및 새로운 모델 적용의 용이성, 종합적인 해석 코드, 1, 2차 계통과 격납건물 모사 능력의 유연성, 자세한 원자력 우리가 필요로 하는 응대사고 해석 코드가 갖추어야 할 요건으로는 인식되고 있다.

안전성에 대한 신뢰감을 증진시키기 위하여 더 늘릴 수 없는 필요한 파제로 선택할 수 있는 기반을 구축함으로써, 국민에게 원자력에 관한 자금성과 외국으로부터의 기술 의존에서 탈피하고, 한반도 더 나아가 우리가 주도적으로 국내 기술로써의 코드 자체의 필요성이 대두되고 있다. 코드 국산화 작업은 국가들이 외국에 코드를 더 이상 쉽게 사용할 수 없게 됨에 따라 이는 응대사고 해석 코드를 보유, 개선하고 있으며, 지적 소유권 문제로 모든 원자력 산업 분야 간에 현재 전세계적으로 원자력 선진 국가들은 자체적인

발전소 뿐만 아니라 앞으로 개발 계획 중인 차세대나 중소형 원자로까지도 확장이 가능하며, 상세한 열수력 기본 지배 방정식을 활용하고, 모델 분석 및 개선에 필요한 코드에의 자유로운 접근이 허용되며, 지속적인 코드 개선으로 최신 모델을 보유하고 있어, 사용자 편의성이 다소 떨어지는 약점에도 불구하고 기존 코드로의 요건을 만족하였다..

MELCOR 전산 코드는 비교적 계산 시간이 빠른 중대사고 해석용 종합 전산코드로서 제 1세대인 STCP (Source Term Code Package)와 구별되는 제 2세대 코드로, 위험도 평가를 위해 미국 NRC 후원으로 Sandia 국립연구소에서 STCP 코드를 대체하기 위해 개발되었다. MELCOR는 중대사고시 일차 계통의 열수력 현상, 노심 손상 과정, 핵분열 생성물의 거동, 격납건물 안에서의 열수력 및 에어로졸 거동, 수소 생성 및 연소 등 노심 손상으로부터 격납건물 파손까지의 전반적인 중대사고 현상을 종합적으로 해석한다. 1985년에 version 1.0이 발표된 이후 1986년 5월에 1.5, 1987년 6월에 1.7, 1989년 3월에 1.8이 발표되었고, 1994년 9월에 1.8.3이, 그리고 1997년에 1.8.4가 출시되었다.

이렇게 선정된 MELCOR는 이미 상당한 수준의 현상 모델을 보유하고 있지만, 최근들어 사고 관리 방안으로 연구되고 있는 노내 노심 용융물 억제 (In-Vessel Retention)와 외벽 냉각 (Ex-Vessel Cooling), 그리고 이와 연관된 원자로 용기 파손 모델을 개선하고 격납건물 안에서의 증기 폭발이나 수소제어 등의 주요 현상 모사 모델을 추가하며, 또한 국내에서 이루어지고 있는 SONATA 실험이나 증기 폭발 실험 결과들을 반영하여 효과적인 기술 자립을 이룰 수 있다. 이러한 기존 모델의 개선과 필요 모델을 추가하기 위해서는 MELCOR의 기본 구조와 현재 사용하고 있는 모델의 현황을 이해하는 것은 필수적이다. MELCOR 코드의 기존 사용 모델에 관하여는 여러 기술보고서를 통하여 이미 분석되었으므로 [4,5,6,7], 이 보고서에서는

코드에서 사용하고 있는 주요 모델에 관한 설명보다는 코드 구조에 대한 이해를 돕기 위해 MELCOR의 수직형 데이터 전달 구조 (즉, 실행 단계, 데이터베이스 처리 단계, 현상 해석 단계, 그리고 다목적 이용 프로그램 단계)에 근거하여 MELCOR 프로그램을 구성하고 있는 MELGEN과 MELCOR를 분석, 정리하였다. 또한, 이러한 작업은 보다 효율적인 데이터 관리와 사용자 편의성 확장 및 코드 이해도를 높이기 위한 데이터 전달 구조 개편에도 유용하게 활용될 수 있다 [8].

2장에는 MELCOR 코드의 전반적인 구조를 분석하여 데이터 처리의 4가지 단계와 프로그램에서 사용되는 파일의 이름, 부프로그램 명명 방식, 그리고 패키지에 대한 설명을 수록하였다. 3장과 4장에서는 MELCOR 프로그램을 구성하고 있는 MELGEN과 MELCOR 각각에 대한 구조 및 데이터 처리 절차를 프로그램 초기화, 사용자 입력 처리, 민감도 분석 계수 처리, 데이터베이스 구축, 편집 관련 내용 정리, 그리고 재시작 파일 작성 등의 기능에 따라 정리하였다. 마지막으로 5장에서는 특별히 노심 모듈에 대한 구조와 노심 패키지와 다른 패키지와의 상관 관계, 그리고, MELCOR 1.8.3과 1.8.4에서 사용하고 있는 부프로그램에 대한 설명 및 차이점 등이 기술되어 있다.

## 2. MELCOR 프로그램 전반적 구조 분석

MELCOR 프로그램은 크게 두 부분으로 실행되는데 첫 부분은 MELGEN이고, 두 번째 부분이 MELCOR이다. MELGEN은 대부분의 입력이 정의되고, 처리되며, 확인되는 부분으로, 모든 입력이 만족되면 초기 계산 조건을 재계산 파일 (restart file)에 저장한다. MELCOR에서는 MELGEN에 의해 생성된 파일과 MELCOR 자체의 입력을 함께 이용하여 사고를 시간에 따라 진행시킨다. 그림 2-1에는 MELCOR 프로그램에서 사용하고 있는 파일 (MELGEN과 MELCOR의 입출력, 진단 결과, 주요 사고 진행 상황, 그림, 그리고 재계산 파일 등) 간의 상호 연관 관계가 나타나 있고, 각 파일의 이름과 기능은 표 2-1에 자세히 설명되어 있다. 계산 후의 결과를 쉽게 그리기 위하여 HISPLT 프로그램을 연결하여 사용하며 MACCS는 격납건물 외기로 방출되는 방사능 물질에 의한 인체 피폭 선량 및 그로 인한 위험도 (risk)를 결정하는 프로그램이다. 여기서 설명된 것처럼, MELCOR라는 동일한 이름이 프로그램 전체를 의미하기도 하고, MELGEN과 비교되는 MELCOR로 쓰이기도 한다. 따라서, 본 보고서에서는 프로그램 전체를 의미할 경우에는 'MELCOR 프로그램'으로, 입력 처리 후의 시간에 따른 계산 부분을 담당하는 MELCOR 프로그램의 일부로 사용될 경우에는 'MELCOR'로 지정하여 사용하였다.

MELGEN과 MELCOR는 4개의 수직 단계로 구성된 체계적인 모듈 구조 (modular architecture)를 가지고 있어 추가하거나 대체할 수 있는 현상 모델의 연결이 용이하다. 제일 윗 단계인 실행 단계 (executive level)에서는 코드의 여러 위치에서 처리하여야 할 일들의 우선 순위와 전반적인 계산 순서를 조정한다. 두 번째 단계는 자료 관리 단계 (database manager level)로 바로 아래 단계인 현상 팩키지 단계 (package) 사이의 연계 관계와 동적 데이터 베이스를 이용하여 MELCOR의 자료를 다음 단계로 전달시켜 준다. 세

번째는 물리 현상을 모사하는 현상 팩키지 단계 (package level)로 각각의 팩키지에는 밀접하게 연관된 물리 현상에 대한 모델링과 잘 정의된 특별한 기능이 포함되어 있다. 마지막으로, 다목적 이용 프로그램 단계 (utility level)가 있는데 여기에는 행렬 (matrix) 변환, 상미분 방정식의 적분 혹은 팩키지들 간의 질량 및 에너지 이동 등을 다룰 수 있는 기능 위주의 부프로그램 (subroutine)들이 포함되어 있다.

MELCOR 프로그램은 팩키지 (package)라는 기본 단위로 개발되어 version 1.8.3 이후에는 모두 24개로 구성되어 있고, 표 2-2에는 각 팩키지의 특성이 요약되어 있다. 여기서, 기본 단위인 팩키지와 위에서 언급한 세 번째 단계인 팩키지 단계에서 같은 이름인 팩키지가 혼용되므로 서로 구분할 필요가 있다. 즉, 기본 단위인 팩키지는 비슷한 기능을 하는 부프로그램이 서로 묶여 있다는 의미이고, 단계에서 사용되는 팩키지는 주로 물리 현상을 분석하는 현상 위주의 부 프로그램을 포함하고 있다. 예로, 원자로 용기 하부구조에서의 열수력 현상을 분석할 수 있는 부 프로그램을 묶어놓은 bottom head (BH) package나 가스의 연소를 모사하는 BUR package는 현상 관련 팩키지로 데이터 처리와 관련하여 3단계인 현상 팩키지 단계에 속한다고 볼 수 있으며, MELGEN과 MELCOR의 실행에 관여하는 EXEC package나 PROG package는 1단계인 실행 단계에, 그리고 상태방정식에 따라 물의 특성을 계산할 수 있는 water properties (H2O) package는 마지막 단계인 다목적 이용 프로그램 단계에 속한다. 특히 물리적 현상과 관련된 3단계에 속해있는 팩키지들은 서로 입력과 출력의 형태로 연결되어 있으며, 이와 같은 현상 팩키지 사이의 상관 관계는 그림 2-2에 정리되어 있다.

MELCOR 프로그램의 팩키지들 중에는 디자인 특성에 따라 늘 사용되지 않는 부분이 있다. 예로, EXEC, CVH, CVT, 그리고 다목적 단계의 팩키지 (utility package)의 일부는 모든 계산에 사용되지만 나머지 대부분은 사용자의

요청에 따라 사용된다. 즉, 가스의 연소를 계산하는 BUR 팩키지는 평소에는 가동하지 않고 (inactive) 사용자가 입력으로 요청하면 (active) 관련 계산을 수행한다. 일반적으로, 발전소에서의 전반적인 사고 진행을 모사할 경우에는 모든 팩키지들이 다 가동되며, 가동 상황은 MELGEN의 출력 파일에 정리된다.

MELCOR에서 사용되는 부프로그램 (subroutine)의 이름은 소속된 팩키지를 나타내기 위하여 관련 팩키지의 앞 두자 혹은 세자를 이용하여 명명한다. 예로 COR로 시작하는 부프로그램들 (예로, CORRN1, COROXD등)은 COR package (노심 관련 현상 모델링)와, BUR로 시작되는 부프로그램들 (예로, BURDBC, BURPS1등)은 BUR package (연소관련 모델링) 와 관련된 프로그램임을 의미한다. 실행 단계와 관련된 프로그램은 MEG-, MEX-, 또는 MXX-로 시작하며, 각각은 MELGEN에서, MELCOR에서, 그리고 MELGEN과 MELCOR 모두에서 공통적으로 사용됨을 나타낸다. 부프로그램 이름 뿐 아니라 데이터를 전달하는 common block 이름도 이와 같은 방식으로 구분하여 관련 팩키지를 표시한다.



### 3. MELGEN 구조

MELGEN 의 임무는 크게 두 가지로 나눌 수 있다. 하나는, 코드에서 사용하고 있는 common block 변수를 코드에 내장되어 있는 값으로 초기화하는 작업으로 여기에는 데이터 베이스 포인터, 민감도 계수, 그리고 내부 데이터 베이스 변수들이 포함된다. 둘째는, 모든 MELGEN의 입출력 파일을 열고, 사용자 입력을 미리 처리, 정리하고 표준 양식으로 저장하여 나중에 여러 가지 팩키지들에 의해 사용될 때를 준비한다. 이때 실행 단계에서 필요한 입력은 바로 처리되며, 또한 민감도 계수도 사용자 입력에 따라 수정된다.

입력 처리 과정은 세 단계로 이루어진다. 첫 단계에서는 실행 팩키지에 의해 각 팩키지가 불러지며 이때 1) 각 팩키지에 필요한 사용자 지정 입력을 읽고, 2) 입력에 의해 정의된 문제의 크기에 맞도록 데이터 베이스 포인터를 재계산하며, 그리고 3) 각 팩키지 안의 데이터를 이용하여 가능한 초기화 및 오류를 확인한다. 두 번째 단계에서는 각 팩키지가 실행 팩키지에 의해 다시 불러지면서 첫 단계에서 정의된 다른 팩키지에서의 데이터를 이용하여 초기화를 마무리하며, 팩키지들 사이의 데이터의 일관성 여부를 확인한다. 마지막 세 번째 단계에서는 팩키지 데이터 중 control function을 초기화한다. 입력 처리는 각 단계별로 시행되며, 각 단계가 성공하지 않으면 다음 단계로 진행되지 않는다. 입력 처리가 끝나면, 실행 팩키지는 각 팩키지를 불러 출력을 작성하며 심각한 오류가 발생하지 않는 경우에는 초기 데이터를 재계산 파일 (restart file)에 기록한다.

MELGEN은 표 3-1에서 보듯이 바로 MEGRUN을 부르도록 설계되어 있고, MEGRUN에서의 데이터 흐름은 표 3-2에 MEGRUN에서 사용된 각 부프로그램의 기능과 함께 요약되어 있다. 그림 3-1에는 위에서 언급한

MELGEN의 기능을 6가지로 세분화하여 각 기능을 주관하는 실행 단계 (executive level)의 부프로그램 이름을 연결하였고, 각각의 부프로그램의 구조에 대해서는 다음 절에 언급하였다. MELGEN에서 각 프로그램별 호출 순서와 대략의 기능은 표 3-3에 정리되어 있다.

### 3.1 데이터 초기화 및 파일 열기

데이터 초기화 및 MELGEN에서 사용하는 입출력 파일 열기와 관련된 프로그램의 구조는 그림 3.1-1에 정리되어 있다. 실행 단계의 MXXPBD 부프로그램에서는 실행 팩키지에서 사용되는 control function 이름 ((MXXCFC(I), I=1,4), time, DT, cycle, CPU), 팩키지의 수와 이름 (NMSPAK), 그리고 팩키지에서 사용되는 민감도 계수의 범위 등이 정의된다. 또한, common block의 상수 (/const/)가 정의되고 각 팩키지의 앞자로 구성된 xyzPBD (여기서 xyz는 팩키지 표시로, COR, CVH, FL등을 의미함)를 마지막에 붙여 pseudo block 자료를 초기화한다. 예로, CORPBD에서는 노심 팩키지 (COR package)에 필요한 데이터 베이스 포인터와 변수 및 행렬들을 초기화한다.

또 다른 실행 단계의 부프로그램인 MEGOPN은 MELGEN에서 필요한 파일을 연다. 먼저 출력 기록을 위한 MEGOUT을 열고, 이곳에 MELGEN에서 처리된 데이터가 쓰여진다. 오류 진단 파일 (diagnostic file)인 MEGDIA에는 MELGEN에서 정의된 초기 조건들로부터 발생하는 오류나 경고가 기록된다. 계산 결과가 의심스러운 경우 사용자는 이 파일을 매 계산 후 꼼꼼하게 검토하여야 한다. 또한 이곳에 재계산에 필요한 파일, 출력과 오류 진단에 필요한 여분의 파일을 열수 있는 부프로그램인 MEGORS, MEGOOT, 그리고 MEGOUT 등이 ENTRY 형식으로 포함되어 있다.

### 3.2 입력 선처리 (PRE-PROCESS) 및 실행 입력 수행

MELGEN의 입력 처리와 관련된 프로그램의 구조는 그림 3.2-1에 정리되어 있다. MELGEN에서 불리는 MEGRUN은 프로그램이 시작되자마자 MEGEDH를 이용하여 MELGEN의 header를 정의한다. MELGEN의 모든 입력은 INPPRE 부프로그램에서 일단 먼저 처리된다. 이 곳에서는 사용자 입력을 읽고, record image를 출력 파일과 record image binary file에 복사한다. 또한, record ID 와 record number field에 따라 record image를 분리 및 대체한다. 이렇게 처리된 입력 파일을 부프로그램 CRACKR이 읽을 수 있도록 기록하며 record image를 해독하는데 필요한 변수들을 바꿔준다. 또한 UTLUND에서는 INPPRE에서 정의되지 않은 변수들에 대한 초기화를 수행한다.

또 다른 실행 단계의 부프로그램인 MEGEDG는 계산 제목 (title)과 다른 선택된 입력을 읽고 처리한다. 이 곳에서 MEGOOT, MEGODG, 그리고 MEGORS 부프로그램을 불러 교대 출력 파일 (alternate output file), 교대 진단 파일 (alternate diagnostic file), 그리고 교대 재계산 파일 (alternate restart file)을 연다. 또한 그림 파일 (plot file), 편지 파일 (mail file), 그리고 정지 파일 (stop file)을 정의한다. 사용자는 상황 파일 (또는 mail file로 코드에 내장된 이름은 MELMAIL)을 batch 계산 중에 생성할 수 있으며, 이 파일의 생성이 확인되면, 코드는 계산 상황을 간단히 요약하여 사용자에게 소식을 전하게 된다. 마찬가지로, 사용자는 정지 파일도 만들 수 있으며 (코드에 내장된 이름은 MELSTP), 이 파일이 확인되면, 코드는 계산을 정지하고 출력과 그림 파일을 생성하고, 재계산 파일에 마지막 단계에서의 결과를 수록한다. 이외에도, 입력 파일에 요구되는 계산 제목 (title), 작업 제목 (job id), 날짜, 그리고 시간 등의 정보를 처리하며, 계산 시간 간격 (time step)을 확인한다. 이와 같이 MEGEDG에서 모든 사용자 입력을 읽고 처리하는 작업이

이루어지며, 이 과정에서 다목적 단계 (utility level)에 속하는 CRACKR이라는 부프로그램을 사용한다.

### 3.3 민감도 계수 입력 처리

민감도 계수를 처리하는 실행 단계의 부프로그램은 MXXSCI로, 입력으로 제공된 민감도 계수 입력을 관련된 팩키지와 연결시키고, 사용자 입력을 CRACKR의 common block에 저장시킨다. 그림 3.3-1에는 민감도 계수 입력을 처리하는 MELGEN의 프로그램 구조가 도시되어 있다.

MXXSCI에서는 xyzSC로 시작하는 각 팩키지 (예로, MXXSC, NCGSC, CVHSC 등)를 두 번 부른다. 처음에는 팩키지 별로 가용한 범위를 결정하며 이 때 common block /CRACKN/의 INTEG 행렬의 처음 두 값에 민감도 계수의 범위 (1000부터 1999까지)를 정의한다. 그리고 나서 민감도 계수가 겹치는 부분이 있는지를 확인한다. 입력 자료에 오류가 없으면, 두 번째로 xyzSC 팩키지를 불러 민감도 계수 입력을 읽고 새로운 민감도 계수 값의 적절성을 확인한다. 바뀐 계수 값에 대해서는 xyzSCX와 xyzSCY 부프로그램을 이용하여 코드에 내장된 변수를 수정하고 이 프로그램에 연계된 SCEDT를 불러 민감도 계수의 처음 값과 수정된 값을 기록한다.

### 3.4 데이터베이스 구축

모든 입력 처리가 완료되면, MELGEN은 실행 단계의 프로그램인 MEGGDB와 MEGPS1, MEGPS2, 그리고 MEGPS3을 이용하여 데이터 베이스를 구축한다. 그림 3.4-1에는 데이터 베이스 구축과 관련된 MELGEN의 구조가 나타나 있다.

실행 단계의 부프로그램인 MEGPS1은 데이터 베이스 관리 단계의 xyzDB1을, xyzDB1은 xyzDBZ를 부르고, 부프로그램 xyzDBZ는 팩키지 단계의 부프로그램인 xyzPS1을 부르고, 마지막으로 xyzPS1은 다목적 단계의 CRACKR을 부르도록 구성되어 있다. 팩키지 입력을 저장하고, 데이터 베이스의 차원 (dimension)과 포인터의 위치 등이 MEGPS1에서 이루어진다.

MEGPS2에서도 MEGPS1에서의 순서처럼 xyzDB2, xyzDBY, 그리고 xyzPS2가 불리며, 이 곳에서는 데이터 베이스의 팩키지들 사이의 일관성을 확인하고 MEGPS1에서의 데이터를 이용한 초기화를 계속한다. MEGPS3에서도 xyzDB6, xyzDB7, 그리고 xyzPS3을 부르며 MEGPS2의 자료를 이용하여 데이터베이스 초기화를 수행한다.

### 3.5 편집 파일 작성

MEGEDT는 MELGEN에서 편집을 담당하는 실행 단계의 부프로그램으로, 문서 출력, 그림 출력, 그리고 재계산 출력을 처리한다. MEGEDT는 매 계산 과정의 처음 부분에서 한 번 불리도록 되어 있다. MEGEDT는 각 팩키지의 xyzDBE를, xyzDBE는 xyzDBF를, 그리고 xyzDBF는 xyzEDT를 불러 각 팩키지에 해당하는 문서 및 그림 출력을 처리한다. 관련 프로그램 구조는 그림 3.5-1에 정리되어 있다.

### 3.6 재계산 파일 작성

재계산을 위해 MELGEN에서 부르는 실행 단계의 파일은 MXXRSW와 MXXRS의 두 가지이다. MXXRSW는 단순히 사용자 입력을 읽어 재계산 파일로 지정된 곳에 복사한다. MXXRS에서는 먼저 재계산 파일을 읽고 쓰는 작업을 하고, 그리고 각 팩키지의 데이터 베이스를 쓰고 읽는 작업을

처리한다. 데이터 베이스를 처리하기 위해 MXXRS는 데이터 베이스 관리 단계의 부프로그램인 xyzPRS와 xyzRIO를 부른다. xyzPRS는 각 팩키지의 데이터 베이스 포인터를, xyzRIO는 각 팩키지의 데이터 베이스를 읽고 쓰는 기능을 담당한다. 재계산과 관련된 MELGEN의 구조는 그림 3.6-1에 나타나 있다.

#### 4. MELCOR 구조

MELCOR 실행은 두 과정을 거친다. 첫 과정은 준비 단계 (setup mode)로 MELGEN에 의해 생성된 재계산 파일 (restart file)로부터 데이터를 읽고 추가적인 입력을 처리한다. MELCOR의 계산 도중이라면 이전의 MELCOR 계산에 의해 생성된 재계산 파일을 읽는다. 두 번째 과정은 계산 단계 (run mode)로, 시간에 따라 사고를 모사하여 변하는 변수들의 값을 각 사이클마다 수정하고 주기적으로 재계산 파일에 새로운 데이터를 저장한다. 준비 단계에서는 MELGEN에서 수행되었던 많은 과정이 반복되는데 common block 초기화, 파일 열기, 민감도 계수 수정, 그리고 실행 팩키지 입력 처리 등이 이루어진다. 재계산 파일로부터 데이터를 읽고 나면, 각 팩키지의 추가적인 사용자 입력 여부를 확인하고 입력이 있으면 기존의 데이터와의 일관성을 확인한다. 위에서 언급한 준비 단계와 계산 단계는 MELCOR의 실행 단계의 프로그램인 MEXSET와 MEXRUN에서 처리되며, MELCOR의 데이터 전달 과정과 구조, 그리고 부프로그램 호출 순서 및 대략의 기능은 표 4-1과 그림 4-1, 그리고 표 4-2에 각각 정리되어 있다.

물론 MELCOR의 핵심은 시간의 경과에 따른 진행 모사 부분이다. 실행 팩키지는 제일 먼저 계산을 끝낼 시점인지, 아니면 다음 출력이나 그림 파일, 혹은 재계산 파일을 작성할 시점인지를 확인하고, 관련된 팩키지를 불러 필요한 작업을 수행하도록 명령한다. 곧 이어, 실행 팩키지는 모든 팩키지에 적용되는 새로운 시간 간격 (time step)을 계산하며 이때 고려하는 변수는 사용자가 입력으로 지정한 최대, 최소값과 전 시간의 계산 결과로부터 얻어진 각 팩키지에서 요구되는 계산 시간 간격이다.

실행 팩키지에서의 계산 시간 간격은 다음 세 단계를 거쳐 결정된다. 먼저, 시간 경과를 모의하기 위해 코드에 내장된 값이나 앞 계산에서 사용된

값으로 모든 팩키지에 필요한 계산 시간 간격을 초기화한다. 두 번째 단계에서는 각 팩키지에서 주어진 시간 만큼 진행시키면 일부 팩키지에서는 팩키지 안에서의 여러 조건을 맞추기 위해 주어진 시간 간격보다 작은 간격을 가지고 계산을 수행하게 되며, 이 경우에는 작은 시간 간격이 새로운 시간 간격으로 정의되어 실행 팩키지가 전 과정을 반복 계산한다. 마지막 단계는 두 번째 단계에서 진행된 시간 동안의 계산량을 검토하여 변화량이 너무 많으면 시간 간격을 줄여 (fallback) 재계산한다.

MELCOR의 특징 중의 하나는 유연한 재계산 능력이다. 즉, 재계산을 위한 데이터 처리 (restart dump)는 사용자에게 의해 정의된 시점이나, 사고 모의 시간 혹은 CPU 시간, 그리고 계산 종료 시점에서 이루어진다. 또한 원자로 용기 파손 이나 격납건물 파손과 같은 특별한 경우에도 재계산 파일이 생성된다. 민감도 계수나 MELCOR 입력 변수는 재계산 시점에서 수정이 가능하여 그 시점부터 다른 부분은 변화 없이 사용자 입력에 의해 바뀐 계통이나 모델에 의한 새로운 사고 경위를 분석할 수도 있다.

#### 4.1 MEXSET 구조

프로그램 MEXSET는 MELCOR 계산의 준비 단계 (setup mode)로, 재계산을 위한 준비 작업을 수행한다. 표 4.1-1에는 MEXSET에서의 데이터 흐름을 부프로그램을 중심으로 정리하였는데 코드의 구조는 MELGEN의 구조와 유사하고, 다만 MEG-로 시작하는 일부 부프로그램이 MEX-로 바뀔 수 있다 (표 3-2 참조). MEXSET는 block data를 만들고, 필요한 입출력 파일과 재계산 파일을 열고, 사용자 입력 파일을 처리하고, 재계산 및 편집을 위해 사용자 입력과 재계산 파일, 민감도 계수 입력을 읽고, 데이터 베이스 간의 일관성을 확인하고, 재계산 파일에 적고, 편집 파일을 작성한다. 그림 4.1-



1에는 MEXSET의 기능을 7가지로 세분하였고, 각 기능에 해당하는 실행 단계의 부프로그램을 연결하였다.

#### 4.1.1 데이터 초기화 및 파일 열기

데이터 베이스 포인터 및 common block 데이터의 초기화는 MELGEN에서 사용한 동일한 부프로그램인 MXXPBD를 이용하여 이루어진다 (3.1절 참조). MELCOR에서 필요한 입출력 파일은 MEXOPN을 이용하여 출력 기록을 위한 MELOUT과 오류 진단 기록을 위한 MELDIA를 연다. 또 MEXOPN에는 재계산 파일, 교대 출력 파일 (alternate output), 교대 오류 진단 파일 (alternate diagnostic), 그리고 특별한 알림 출력 (message output)을 열기 위한 MEXORS, MEXOOT, MEXODG, 그리고 MEXOMS 부프로그램이 entry 형식으로 포함되어 있다. 그림 4.1.1-1에는 MELCOR에서 사용하는 입출력 파일 열기와 관련된 실행단계의 프로그램이 정리되어 있다.

#### 4.1.2 입력 선처리 (Pre-process) 및 실행 입력 수행

입력의 선처리는 MELGEN에서와 마찬가지로 INPPRE 부프로그램에서 이루어진다 (3.2절 참조). INPPRE 부프로그램이 불리고 나면, 다목적 단계의 부프로그램인 UTLUND를 불러 정의되지 않은 변수에 대한 초기화를 수행한다. MEXEDI 부프로그램에서는 MEXOOT, MEXODG, MEXOMS, 그리고 MEXORS 부프로그램을 불러 관련 파일들을 연다. 그리고 나서, 계산 제목 (title), 작업 제목 (job id), 계산 종료 시간, 계산 시간 간격 (time step)의 최대, 최소값, CPU 지정 값 등을 확인하는 과정을 거쳐 입력을 읽고 처리하고, 이 과정에서 다목적 단계의 CRACKR을 이용한다. 마지막으로 관련 정보나 입력 오류를 출력한다. 그림 4.1.2-1에 MELCOR의 입력 처리와 관련된 프로그램의 구조가 정리되어 있다.

#### 4.1.3 재계산 파일 읽기 및 민감도 계수 수정

재계산 파일을 읽는 절차는 MELGEN에서와 마찬가지로 실행 단계의 프로그램인 MXXRS를 불러 수행하며 (3.6절 참조) 그림 4.1.3-1에 정리되어 있다. 민감도 계수 관련 입력도 MELCOR에서 처리되며 MELGEN에서처럼 동일한 MXXSCI를 이용한다 (3.3절 참조).

#### 4.1.4 팩키지 입력 검토

각 팩키지의 입력을 받고, 데이터 베이스를 확인 및 수정하는 작업은 실행 단계의 부프로그램인 MEXUIN에서 이루어진다. MEXUIN은 데이터베이스 관리 단계의 xyzDBI를, xyzDBI는 xyzDBJ를 부르고, xyzDBJ는 팩키지 단계의 부프로그램인 xyzUIN을 부르고, 마지막으로 xyzUIN은 다목적 단계의 CRACKR을 부르도록 구성되어 있다. 부프로그램의 앞에 사용된 xyz는 앞에서처럼 팩키지 이름의 앞 두 자나 세 자로 구성되어 각 팩키지별로 사용됨을 의미한다. 그림 4.1.4-1에는 MELCOR에서의 팩키지 입력 처리 과정이 도시되어 있다. 다만, 이러한 데이터 흐름의 구조는 일반적인 형태를 나타낸 것으로, 필요가 없는 부분은 누락되어 있거나 팩키지에 따라 약간씩 형식이 바뀔 수 있다.

#### 4.1.5 데이터 베이스 검토

MELCOR에서 사용하는 데이터 베이스 간의 일관성은 실행 단계의 부프로그램인 MEXCHK를 이용하여 확인한다. MEXCHK은 데이터베이스 관리 단계의 xyzDBG를, xyzDBG는 xyzDBH를 부르고, xyzDBH는 팩키지 단계의 부프로그램인 xyzCHK을 부른다. 이 단계에서 MELCOR 계산을

시작하기 전의 데이터 베이스를 마지막으로 확인한다. 위에서 언급한 데이터 베이스 확인 과정이 그림 4.1.5-1에 정리되어 있다.

#### 4.1.6 입력 반복 및 편집

MELCOR에서의 입력 반복 (echo) 및 편집에 관한 MEXSET의 구조는 그림 4.1.6-1에 나타나 있다. MXXRSW는 MELGEN에서처럼 입력을 읽고 재계산 파일에 반복 기록한다 (3.6절 참조). MELCOR에서의 편집을 관장하는 실행 단계의 프로그램은 MEXEDT로 이 곳에서 문서, 그림, 그리고 재계산 출력을 수행한다. MELGEN의 MEGEDT처럼 매 계산 과정의 처음 부분에서 한 번 불리도록 되어 있다. MEXEDT는 각 팩키지의 데이터 베이스 단계의 부프로그램인 xyzDBE를, xyzDBE는 xyzDBF를, xyzDBF는 팩키지 단계의 xyzEDT를, xyzEDT는 다목적 단계의 MXXPLK 를 부른다. 그림 3.5-1과 그림 4.1.6-1에서 보듯이 MEGEDT와 MEXEDT는 동일한 데이터 베이스 부프로그램과 팩키지 단계의 부프로그램을 사용함을 알 수 있다. MXXPLU에서는 plot key directory를 출력한다.

#### 4.2 MEXRUN 구조

MEXRUN은 MELCOR의 계산 부분을 실행하는 프로그램으로 그림 4.2-1에서처럼 크게 두 부분으로 구분된다. 하나는 재계산 파일을 작성하고, 편집하며, 그림을 준비하는 부분이고, 다른 하나는 사고 진행을 모사하기 위한 물리 현상 및 계산 시간 간격 (time step)을 결정하는 부분이다. MEXRUN에서의 전반적인 데이터 흐름은 표 4.2-1에 부프로그램 (subroutine) 중심으로 정리되어 있고, 중요한 실행 단계의 파일들은 MEXEND, MEXEDT, MEXDTC, 그리고 MEXLOP 등이다. 각각의 부프로그램에 대한 설명은 다음 절에 서술되어 있다.

#### 4.2.1 재계산 파일, 편집 및 그림 작성

MEXRUN에서는 그림 4.2.1-1에 단계 별로 정리되어 있는 것처럼 재계산 파일과 편집, 그리고 그림 파일을 작성한다. MEXEND에서는 계산 종료 시간을 확인하고, 확인되면 마지막 결과를 편집하고 재계산 파일을 작성한다. 또한 출력 파일 편집을 위하여 MEXSET에서 사용되었던 MEXEDT (그림 4.1.6-1 참조)를 부르고, 다만 MEXSET에서는 다목적 단계의 MXXPLK를 부르는 반면, MEXRUN에서는 MXXPLW를 부른다. MXXPLK와 MXXPLW는 다 같이 xyzEDT에서 불리며, 변수 MXXWTS에 의해 plot key record를 기록할 때에는 전자를, plot time record를 기록할 때에는 후자를 사용한다. MEXRUN에서는 이외에도 재계산 파일의 덤프를 확인하기 위해 MXXRS를 부르며, MXXRS의 구조는 그림 4.1.3-1과 같다.

#### 4.2.2 계산 시간 간격 결정과 물리적 현상 모사

MEXRUN에서의 계산 시간 간격 결정과, 물리적인 현상 모사와 관련된 부프로그램의 연결 구조는 그림 4.2.2-1에 정리되어 있다. 코드에서의 실제적인 계산을 수행하기 위해서는 사고 진행에 따라 변화하는 물리 변수들에 대하여 적분을 수행할 때 필요한 계산 시간 간격 (time step)을 결정해야 하는데, MEXRUN에서는 실행 단계의 부프로그램인 MEXDTC를 불러 시간 간격을 결정하며, 이 때 근거가 되는 자료는 사용자 입력과 여러 가지 현상 팩키지에서 제공되는 시간 간격 입력 들이다.

또 다른 중요한 MEXRUN의 기능은 시간 진행에 따른 사고 진행을 모의하는 것으로 원자력 발전소에서의 노심 손상 사고 후 발생 가능한 물리, 화학적 현상 분석을 포함한다. 이 작업은 실행 단계의 물리 현상 구동 프로그램인 MEXLOP을 이용하여 수행하며, MEXLOP에서의 데이터 흐름은 표 4.2.2-1에

부프로그램 위주로 정리하였다. MEXLOP은 데이터 교환을 위한 행렬을 준비하는 xyzDBM, 물리 현상을 모사하는 xyzDBC, 그리고 계산 후의 데이터 관리를 위한 xyzDBO로 구성된다. 데이터 베이스 관리 첫 번째 단계의 xyzDBM은 두 번째 단계의 xyzDBN을, xyzDBN은 팩키지 단계의 xyzMDS를 부른다. 데이터 베이스 관리 단계의 xyzDBO는 xyzDBP를, xyzDBP는 팩키지 단계의 xyzVTO를 부른다. 물리 현상을 담당하는 xyzDBC는 xyzDBD를 부르며, xyzDBD는 팩키지 단계의 xyzRUN을 불러 계산한다. 예로 가스 연소는 BUR 팩키지에서 분석되는데 MEXRUN에서의 데이터 흐름은 그림 4.2.2-1에서처럼 MEXLOP, BURDBC, BURDBD, 그리고 BURRUN으로 연결된다. 계산 시간 간격을 수정할 필요가 있을 때에는 MEXDTB를 불러 계산 중의 DT를 DTPKG로 교체한다.

## 5. 노심 모듈 구조

MELCOR의 노심 모듈은 그림 2-2에 정리되어 있는 MELCOR 프로그램의 중대사고 현상 팩키지 중 노심 팩키지 (COR package)에 포함되어 있다. 노심 팩키지는 MELCOR의 여러 팩키지 중 가장 중요한 부분으로, 노심 내부의 연료봉과 다른 구조물, 그리고 원자로 하부 용기 구조물의 거동을 모사하며, 이 때 연료봉 가열 및 산화, 연료봉 용융 및 녹아 내림 (캔들링 효과), 유로 차단, 노심 파쇄층 형성과 하부 용기로의 재배치, 원자로 용기 파손, 그리고 노심 물질의 격납건물로의 방출 현상 등이 포함된다. 이러한 현상들을 다루기 위해 노심 팩키지는 노심을 감싸고 있는 control volume 과의 연계를 위한 CVH와 FL 팩키지, 노심 주변의 열침원과의 열 교환을 위한 HS 팩키지, 그리고 원자로 용기 파손 이후의 노심 파쇄물의 이송을 다루는 FDI 팩키지와 밀접한 관계를 유지하고 있다. 이 절에서 다루고 있는 노심 구조는 MELCOR 1.8.3을 중심으로 작성하였고, 최근에 발표된 1.8.4에서 사용된 부프로그램들은 따로 5.3절에 언급하였다.

### 5.1 노심 팩키지 (COR PACKAGE) 구조

MELCOR에서의 전반적인 물리적 현상은 4.2.2절에 설명되어 있듯이 MEXLOP에서 다루어지며, 그림 4.2.2-1에는 MEXLOP과 연결되어 있는 데이터 베이스 단계의 부프로그램과 현상 해석 팩키지 단계의 부프로그램이 정리되어 있다. 이 중에서 노심에서의 중대사고 진행은 xyz 대신 COR로 표현되는 CORDBC와 CORDBD에서 데이터 베이스를 관리한다.

표 5.1-1에서 보듯이 부프로그램 CORDBC는 CORDBD를 부르되, 앞 뒤로 CPU 시간을 확인하기 위해 MXXCPU를 활용하고 있다. 표 5.1-2에는 CORDBD에서의 데이터 흐름이 정리되어 있다. 일반적으로 xyzDBD는

현상을 해석하기 위해 xyzRUN을 부르도록 설계 되었으나, 노심에서의 현상을 취급하는 COR 팩키지에서는 CORRUN 대신, 노심에서의 계산 순서에 순차적으로 번호를 붙여 부프로그램 이름을 구성하였다. 표 5.1-2에서 보는 것처럼 노심 팩키지를 시작하면서, CORRNb를 먼저 불러 하부 구조물에 재배치된 노심 파쇄층의 표면과 표면이 바라보는 노심 구조물과의 복사열 전달량을 계산하고, 계속하여 CORRN1, CORRN2, CORRN3, CORRN4, 그리고 CORRN6를 불러 노심 가열 및 용융, 열전달 비율, 팩키지 사이의 연계, 노심 손상 및 재배치, 그리고 전반적인 에너지 균형 관계를 각각 추적한다.

이 중에서 노심 재배치 이전의 가열 및 산화 현상은 CORRN1에서 다루어지며 데이터 흐름은 표 5.1-3에 부프로그램 별로 간단한 설명과 함께 요약되어 있다. 먼저 초기화 작업이 진행되고, RN1DHC에서 원자로 정지 이후의 붕괴열 (decay power)을 계산하여 사용자 입력인 첨두 인자 (peaking factor)를 이용하여 연료봉 각 노드의 위치에 따라 분포 시킨다. CORPOW에서는 사용자 입력 조건 (COR00004 카드의 ICFFIS)에 따라 핵분열 열 (fission power)을 계산한 후 붕괴열을 합하여 원자로에서의 출력을 얻는다 [9]. 그 다음에는 연료봉에서의 에너지 균형 방정식에 사용되는 여러 가지 열전달량 들을 계산하기 위해 관련 열전달 변수, 예로, 연료봉과 피복재 사이의 간극에서의 열전달 계수 (COR GAP), 노심과 하부 구조물 사이의 복사열량 (CORRAD), 구성 성분 사이의 열전도계수 (CORCTK), 연료봉 사이의 축 방향 열전도량 (CORCND), 노심에서의 대류 열전달량 (CORCNV) 등이 계산된다. 이러한 값들이 정해지면, 연료봉의 모든 노드에 대하여 앞 시간에 계산된 온도 차이에 의한 열 전달량과 새로운 중대사고 현상 (예로 산화 반응으로 인한 발열)에 의한 열원을 추가적으로 계산하여 에너지 균형 방정식으로부터 다음 시간에서의 구성 성분들의 온도를 계산한다.

## 5.2 다른 팩키지와의 연계

노심 팩키지는 노심 주변을 흐르는 유체 (냉각재나 기체)와 노심 구조물과의 열전달 현상을 모의한다. 이 때 유체의 조건은 CVH 팩키지와 FL 팩키지를 이용하여 묘사되며, 이 팩키지들이 노심 팩키지와 연결된다. 사용자는 반경 방향과 축 방향으로 구분되어 정의되는 연료봉의 각 노드로부터 노심 주위의 유체로 열전달 정보를 전달하기 위해 노심 입력 변수 중에서 CORijj01 카드 중 ICVHC를 이용하여 노심이 포함되어 있는 control volume을 정의한다. 주의 사항으로는, 이 곳에 정의되는 control volume은 CVH 팩키지 입력에서도 틀림없이 정의되어야 한다. Control volume 사이의 유체의 흐름은 FL 팩키지에서 정의된 유로를 이용하여 정의된다. 노심 팩키지는 입력 확인 단계에서 노심에서 언급된 control volume이 있는지 만을 확인한다. 따라서 사용자는 고려되고 있는 노심의 실제적인 구조에 맞도록 CVH와 FL 팩키지의 입력을 이용하여 적절하게 입력을 준비하여야 한다.

노심 안에서 축 방향으로의 온도 분포를 보완하기 위해  $dT/dz$  모델이 사용되는 경우에는, 노심으로의 입구 온도를 결정하기 위해 CORTINxx 입력을 사용하며 이 입력 역시 CVH 와 FL 팩키지와 간접적으로 연관되어 있다. CORTINxx의 xx에는 임의의 값이 사용될 수 있으며, IVOL과 ITIN의 두 값이 필요하다. IVOL에는 노심이 잠겨있는 control volume의 번호가, ITIN에는 각 control volume으로 공급되는 입력 온도를 계산하는 선택 입력 (0, 양수, 음수)이 요구된다. 사용자는 CORTINxx에서 언급되는 CVH나 CF 입력들이 각 팩키지에 포함되어 있는지 반드시 확인하여야 한다.

노심 주위의 구조물은 연료봉으로부터의 전달되는 복사열 등으로 가열되며, 따라서 노심에서의 열손실로 다루어진다. 따라서, 열침원 팩키지 (HS package)와 연결하여 주변 구조물의 가열 및 노심과의 열전달 현상을



모사한다. 사고가 진행되어 연료봉 용융 및 재배치로 원자로 용기 하부에 노심 파쇄물이 쌓이게 되면, 원자로 용기는 가열되어 손상될 수 있으며 이 경우 노심 용융물들이 격납건물로 이송된다. MELCOR는 FDI 팩키지를 이용하여 격납건물로의 이송, 열전달, 산화 및 고압분출 등의 현상을 다루며, 원자로 용기 하부에서의 자세한 열전달 거동을 분석하기 위해 BH 팩키지를 이용하여 원자로 용기가 완전히 고갈된 후의 노심 용융물의 거동 및 용기 벽의 가열 및 파손을 예측한다. KAERI/TR-1125/98 보고서의 부록 B에는 MELCOR의 BH 팩키지와 COR 팩키지를 포함한 다른 팩키지와 의 연관 관계가 간략하게 소개되어 있다 [4].

### 5.3 노심 팩키지와 CVH 팩키지 관련 부프로그램

MELCOR 프로그램에서 사용하고 있는 노심 관련 부프로그램의 이름과 기능을 표 5.3-1에 정리하였다. 또한 노심 팩키지와 가장 밀접하게 정보를 주고 받는 CVH 팩키지에서 사용하고 있는 부프로그램의 목록도 정리하여 표 5.3-2에 수록하였다. 이 표들은 노심 모듈의 여러 가지 물리 현상들이 구체적으로 어느 부프로그램에서 모델 되고 있는지를 알려주며, 현재 진행되고 있는 모델 개선 및 개발의 기본 자료로 활용될 수 있다. 앞에서 언급된 코드 구조는 주로 MELCOR version 1.8.3을 중심으로 작성되었으나, 최근 입수된 1.8.4에 대한 분석에 도움을 주기 위해 표 5.3-1와 표 5.3-2에는 1.8.3과 1.8.4를 구분하여 1.8.4에 새로이 추가된 부프로그램을 구분할 수 있도록 하였다.

## 6. 결론

중대사고 종합 전산 코드인 MELCOR 프로그램의 코드 구조를 분석하였다. MELCOR 프로그램은 대부분의 입력이 정의되고, 처리되며 확인되는 MELGEN과 MELGEN에서 처리된 입력과 자체의 입력을 이용하여 사고를 시간에 따라 진행시키는 MELCOR로 구성되어 있다. MELGEN과 MELCOR는 4개의 수직 단계로 구성된 체계적인 모듈 구조를 가지고 있어 개발자가 용이하게 새로운 모델을 추가하거나 대체할 수 있다.

MELGEN의 임무는 두 가지로, 하나는 코드에서 사용하고 있는 common block 변수를 코드에 내장되어 있는 값으로 초기화하는 작업으로 여기에는 데이터 베이스 포인터, 민감도 계수, 그리고 내부 데이터 베이스 변수 등이 포함된다. 또 하나는 모든 MELGEN의 입출력 파일을 열고 사용자 입력을 미리 처리, 정리하며, 표준 양식으로 저장하여 나중에 여러 팩키지들에 의해 사용될 때를 준비한다. MELCOR는 두 가지 과정을 거친다. 첫 과정은 준비단계로 MELGEN에 의해 생성된 재계산 파일로부터 데이터를 읽고 추가적인 입력을 처리하며, 두 번째 과정은 계산 단계로 시간에 따라 사고를 모사하여 변하는 변수들의 값을 각 사이클마다 수정하고 주기적으로 재계산 파일에 새로운 데이터를 저장한다. MELCOR의 핵심은 시간의 경과에 따른 진행 모사 부분으로, 계산의 계속, 편집, 그리고 종료 상황을 판단하여 관련 팩키지를 불러 필요한 작업을 수행하도록 명령한다.

특히 MELCOR 프로그램의 핵심 모듈인 노심 모듈 구조를 구별하여 분석, 정리하였다. 이 부분은 노심 내부의 연료봉과 다른 구조물, 그리고 원자로 하부 용기 구조물의 거동을 모사하며, 이때 연료봉의 가열 및 산화, 용융, 유로 차단, 노심 파쇄층 형성과 하부 용기로의 재배치, 원자로 용기 파손, 그리고 노심 물질의 격납건물로의 방출 현상 등이 포함된다. 따라서 이

팩키지는 제어 체적을 다루는 CVH와 유로를 모의하는 FL 팩키지, 노심 주변의 열침원을 모사하는 HS 팩키지, 그리고 노심 파쇄물의 이동을 다루는 FDI 팩키지와 밀접한 연계를 이루고 있다.

따라서, 여기서 분석되고 정리된 전반적인 코드 구조에 대한 기술은 코드 안에서의 데이터의 흐름에 대한 이해를 높이기 위한 필수적인 절차로, 이를 바탕으로 코드가 가지고 있는 기존의 모델을 개선하거나 노내 노심 용융물 억제나 외벽 냉각 등의 필요한 새로운 모델을 추가할 경우, 그리고 코드의 이해도를 높이기 위한 데이터 구조 개편이나 사용자 편의성을 향상시키기 위한 Graphical User Interface 등을 첨가할 경우 큰 도움이 될 것으로 판단된다.

## 7. 참고문헌

- [1] “중대사고 분석코드 개발,” 원자력 안전성 향상 연구, KAERI/RR-1748/96, 과학기술처, 1997.7
- [2] “가압 중수로형 원자력발전소의 2단계 PSA를 위한 전산코드 개발,” KAERI/RR-1573/95, 1995.12
- [3] 김동하 외, “국내 중대사고 해석 종합전산코드 개발 방향에 관한 연구,” 98춘계학술발표회, 한국원자력학회, 1998.5.30
- [4] 안광일 외, “기존 중대사고 해석코드 원자로 하반구 노심 용융물 냉각해석모델 비교분석,” KAERI/TR-1125/98, 1998.8
- [5] 박수용 외, ”MELCOR 및 MAAP 코드의 노심 용융물과 콘크리트 반응 해석모델 비교분석,” KAERI/TR-1255/99, 1999.3
- [6] 박수용 외, “중대사고시 캐비티내 노심 용융물 거동에 관한 연구,” KAERI/TR-1529/2000, 2000.3
- [7] 송용만 외, ” MELCOR 코드 내 동력학 모델의 추가 및 정지불능 예상 과도 상태 (ATWS)의 분석,” KAERI/TR-1273/99, 1999.3
- [8] 박선희 외, “중대사고 해석 코드 개발을 위한 MELCOR 개편시안,” KAERI/TR-1536/2000, 2000.3
- [9] R.O.Gauntt, et. al., “MELCOR Computer Code Manuals,” Primer and Users’ Guides, Version 1.8.4, July, 1977, NUREG/CR-6119, Vol. 1, Rev. 1, SAND97-2398.

Table 2-1 Files Used by MELGEN and MELCOR (1/2)

File Name/ Default File	Description
User Input (MELGIN, MELIN)	The MELGEN User Input File contains the majority of the user input defining the problem for MELCOR. MELGEN processes and checks this input and creates a Restart File for MELCOR.
Output (MEGOUT, MELOUT)	Both MELGEN and MELCOR generate printed output, all of which are written to their respective normal Output Files. Selected information is written also to the Diagnostic, Message, and Terminal Files for the convenience of the user.
Plot (PELPTF)	The values of all MELCOR plot variables are written to the Plot File at time intervals determined by the user. This file is read by the HISPLT graphic post-processing program, which generates a graphics metafile containing plots requested by the HISPLT User Input File.
Restart (MELRST)	The MELCOR database, containing all the necessary data to restart MELCOR, is written to the Restart File at time intervals determined by the user.
Message (MELMES)	Special messages are written to the Message File. This file is written only by MELCOR and contains the occurrence time of significant events such as vessel bottom head failure, melt ejection, hydrogen burns, etc.
Diagnostic (MEGDIA, MELDIA)	The Diagnostic File contains certain diagnostic messages generated by MELGEN and MELCOR, including error messages and warnings that are useful to the user.
Extended Diagnostic (extDIA)	The Extended Diagnostic File contains more complete diagnostic information than the Diagnostic File but retains only the latest messages. This file is most useful to the developers to trace code problems that the user can not control.

Table 2-1 Files Used by MELGEN and MELCOR (2/2)

File Name/ Default File	Description
Terminal	The Terminal File (or Batch Job "Log" File) contains direct terminal output from MELCOR giving a brief summary of the course of the calculation. In addition to special messages, the problem time, time step, and CPU time are written to this file as requested by the user.
Stop (MELSTP)	The user may create this file at any time during a batch MELCOR execution. If this file is present, the MELCOR calculation is terminated and data are written to the Output, Plot, and Restart Files for the last cycle.
Mail (MELMAIL)	The user may create this file at any time during a batch MELCOR execution. If this file is present, MELCOR will create a short summary of the state of the calculation and mail it to the user. The purpose of this feature is to give the user informed control over batch jobs.

Table 2-2 MELCOR Packages (1/3)

Package Name	Package Description
BH	Bottom Head. This model was developed by the Oak Ridge National Laboratory, and is an alternative to the lower plenum modeling in COR.
BUR	Burn (Combustion) of Gases. Compares conditions within control volumes against criteria for deflagrations and detonations. Initiates and propagates deflagrations involving hydrogen and carbon monoxide. Calculates burn completeness and flame speed.
CAV	Core-Concrete Interactions. Has CORCON-Mod3 features with enhanced sensitivity analysis and multi-cavity capabilities
CF	Control Functions. Evaluates user-specified "control functions" and applies them to define or control various aspects of the computation such as opening and closing of valves; controlling plot, edit, and restart frequencies; defining new plot variables, etc.
COR	Core Behavior. Evaluates the behavior of the fuel and other core and lower plenum structures including heatup, candling, flow blockages, debris formation and relocation, bottom head failure, and release of core material to containment.
CVH	Control Volume Hydrodynamics. In conjunction with the FL package, evaluates mass and energy flows between control volumes.
CVT	Control Volume Thermodynamics. Evaluates the thermodynamic state within each control volume for the CVH package. No User's Guide is written for this package since no user input is required. However, a Reference Manual is written.
DCH	DECAY HEAT. USED BY OTHER PACKAGES TO EVALUATE DECAY HEAT POWER ASSOCIATED WITH RADIONUCLIDE DECAY.

Table 2-2 MELCOR Packages (2/3)

Package Name	Package Description
EDF	External Data Files. Controls the reading and writing of large external data files, in close interface with the Control Function and Transfer Process packages.
EOS	Equation of State. The CVT, H2O, and NCG packages are stored as one block of code under this name.
ESF	Engineered Safety Features. Models the thermal-hydraulics of engineered safety features that cannot be effectively modeled by building appropriate components or systems using the CVH, FL, HS, and CF packages. Currently, only the fan cooler model is included in ESF; the containment sprays are modeled in the SPR package.
EXEC	EXECUTIVE PACKAGE. CONTROLS EXECUTION OF MELGEN AND MELCOR.
FDI	Fuel Dispersal Interactions. Models ex-vessel debris relocation, heat transfer, and oxidation due to fuel-coolant interactions and high pressure melt ejection.
FL	Flow Paths. Models, in conjunction with the CVH package, the flow rates of gases and liquid water through the flow paths that connect control volumes.
H2O	Water Properties. Evaluates the water properties based on the Keenan and Keyes equation of state extended to high temperatures using the JANAF data. This set of routines is in the "EOS" code package. No user input is required.
HS	Heat Structures. Models the thermal response of heat structures and mass and heat transfer between heat structures and control volume pools and atmospheres. Treats conduction, condensation, convection, and radiation, as well as degassing of unlined concrete.



Table 2-2 MELCOR Packages (3/3)

Package Name	Package Description
MP	Material Properties. Evaluates the physical properties of materials for other packages except for common steam and noncondensable gas properties (see H2O and NCG).
NCG	NonCondensable Gas Equation of State. Evaluates the properties of noncondensable gas mixtures using an equation of state based on the JANAF data. This set of routines is in the "EOS" code package.
PROG	Part of MELGEN/MELCOR executive package separated for computer library and link purposes.
RN	Radionuclide Behavior. Models radionuclide releases, aerosol and fission product vapor behavior, transport through flow paths, and removal due to ESFs. Allows for simplified chemistry.
SPR	Sprays. Models the mass and heat transfer rates between spray droplets and control volumes.
TF	Tabular Functions. Evaluates user-selected "tabular functions" to define or control various aspects of the computation such as mass and energy sources; integral decay heat; plot, edit, and restart frequencies, etc.
TP	Transfer Process. Controls the transfer of core debris between various packages and the associated transfer of radionuclides within the RN package. In order to transfer core material between packages, some TP input is required, and is described in the COR, FDI, and CAV package User's Guides.
UTIL	Utility Package. Contains various utilities employed by the rest of the code.

Table 3-1 Data Flow of MELGEN Main Program

Program MELGEN

Call MEGRUN

- EXEC package
- Top level controller for MELGEN

End

Table 3-2 Data Flow of MEGRUN in MELGEN

\*DECK MEGRUN

Call MEGEDH

- Define MELGEN header

Call MXXPBD

- Load constants and call other pseudo block data routines

Call MEGOPN

- Open MELGEN input-output files

Call INPPRE

- Process user input

1. Read the user input
2. Echo the record images to user output and record image binary file.
3. Sort and replace the record images according to record ID and record number field.
4. Write a processed input file to be read by subroutine CRACKR
5. Change the parameters governing the interpretation of the record images.

Call UTLUND

- Initialize undefined variables using XREALX

Call MEGEDG

- Read and process user information for title and other selected input.

Call MXXSCI

- Read the user input for the sensitivity coefficients
- Decide which package is associated with the sensitivity coefficient
- Call the correct package
- The data from the user input will be in the CRACKR common blocks

Call MEGGDB

- Read user input and generate the database

Call MEGEDT

- Generate edit files

Call MXXRSW

Call MXXRS

- Read and write restart files

Call CRAEND

- Check (locates and prints) any unprocessed record images

Return/End

Table 3-3 List of Subroutines called from MELGEN (1/9)

MELGEN						
-	MEGRUN : TOP LEVEL CONTROLLER FOR MELGEN					
-	MEGEDH : DEFINES THE MELGEN HEADER					
-	MXXPBD : THE MELCOR EXECUTIVE PSUEDO BLOCK DATA ROUTINE. IT HOLDS CONSTANTS AND CALLS OTHER PSUEDO BLOCK DATA ROUTINES					
-	SYSNUM					
-	NCGPBD					
-	CVHPBD	.....				
-	FLPBD	.....				
-	CVTPBD	.....				
-	CORPBD					
-	CORVUP					
-	MESERA	.....				
-	BHPBD	.....				
.....	.....					
-	MEGOPN : OPEN MELGEN INPUT - OUTPUT FILE					
-	MXXMES	.....				
-	MXXME1	.....				
-	MEXFOP	.....				
-	MXXQA	.....				
-	INPLNB	.....				
-	INPPRE 1. READS THE USER INPUT 2. ECHOS THE RECORD IMAGES TO USER OUTPUT AND RECORD IMAGE BINAR FILE. 3. SORTS AND REPLACES THE RECORD IMAGES ACCORDING TO RECORD ID AND RECORD NUMBER FIELD. 4. WRITES A PROCESSED INPUT FILE TO BE READ BY SUBROUTINE CRACKR 5. CHANGES THE PARAMETERS GOVERNING THE INTERPRETATION OF THE RECORD IMAGES.					
-	INPOPW	.....				
-	INPCW	.....				

Table 3-3 List of Subroutines called from MELGEN (2/9)

		-	INPSRT	.....			
		-	MXXNAN : Get any optional redefinition of UNDEF, and preset XREALX				
		-	CRACKR	.....			
		-	MESERA	.....			
		-	MESERA ..... : Prints standard format error messages 또는 UTLUNF ..... : INITIALIZES LOCATIONS N1 TO N2 OF ARRAY X TO THE VARIABLE UNDEF, WHICH IS INTENDED TO BE AN UNDEFINED OR ILLEGAL BIT PATTERN STORE IN COMMON BLOCK MXXUND				
		-	MEGEDG : Reads/processes user information				
		-	MXXSCI READS THE USER INPUT FOR THE SENSITIVITY COEFFICIENTS, DECIDES WHICH PACKAGE IS ASSOCIATED WITH THE SENSITIVITY COEFFICIENT AND CALLS THE CORRECT PACKAGE. THE DATA FROM THE USER INPUT WILL BE IN THE CRACKR COMMON BLOCKS.				
		-	MEGGDB : generate database				
		-	MEGPS1 : PASS 1 OF DATABASE GENERATOR PACKAGE				
		-	MEGZGE	.....			
		-	MEGDB1	.....			
		-	CVTDB1	.....			
		-	INPDB1	.....			
		-	CRADB1	.....			
		-	MESDB1	.....			
		-	SCDB1	.....			
		-	H2ODB1	.....			
		-	TFDB1	.....			
		-	CFDB1	.....			
		-	NCGDB1	.....			
		-	DCHDB1	.....			
		-	MPDB1	.....			
		-	CVHDB1	.....			
		-	FLDB1	.....			

Table 3-3 List of Subroutines called from MELGEN (3/9)

			-	CORDB1 : PART OF THE CORE SETUP PACKAGE. THIS SUBROUTINE INTERFACES THE DATABASE WITH THE REST OF THE CORE INITIALIZATION PACKAGE. ALL ARRAYS BELOW THIS LEVEL ARE IN CORE AND PASSED THROUGH ARGUMENT LISTS.		
			-	CORDBZ : THIS SUBROUTINE IS PART OF THE CORE PACKAGE IN MELGEN. IT DOES THE FOLLOWING TASKS: 3. READS THE USER INPUT 2. ALLOCATES DATA ARRAYS 3. STORES THE CORE DATA IN THE DATA ARRAYS *. CORPNT common block의 값, 즉, pointer value를 계산하며, 이후의 subroutine CORPRS에서 RESTART file에 WRITE함		
				-	CORP1A	.....
				-	CORP1Z	.....
				-	CORP1R	.....
				-	CORP1B	.....
				-	CORP1D	.....
				-	CORP1F	.....
				-	CORXRI	.....
				-	CORXRR	.....
				-	MESERR	.....
			-	BHDB1	.....	
			-	EDFDB1	.....	
			-	SPRDB1	.....	
			-	HSDB1	.....	
			-	BURDB1	.....	
			-	CAVDB1	.....	

Table 3-3 List of Subroutines called from MELGEN (4/9)

			-	FDIDB1	.....			
			-	RN1DB1	.....			
			-	RN2DB1	.....			
			-	TPDB1	.....			
			-	ESFDB1	.....			
			-	MEGPS2 : PASS 2 OF DATABASE GENERATOR PACKAGE				
			-	MEGDB2	.....			
			-	NCGDB2	.....			
			-	CVHDB2	.....			
			-	FLDB2	.....			
			-	CVTDB2	.....			
			-	CORDB2 : PART OF THE SECOND PASS GENERATING THE DATABASE FOR THE CORE PACKAGE. IT PASSES DOWN THE PACKAGE DATABASES REQUIRED FOR PASS 2 INPUT PROCESSING TO CORDBY.				
			-	CORDBY : THIS SUBROUTINE IS PART OF THE SECOND PASS IN GENERATING THE DATABASE FOR THE CORE PACKAGE IN MELGEN. IT DOES THE FOLLOWING TASKS: 2. CHECKS THE USER INPUT WITH INPUT FROM OTHER PACKAGES 2. CALCULATES ADDITIONAL QUANTITIES IN THE DATABASE THAT COULD NOT BE CALCULATED IN THE FIRST PASS.				
			-	CORVVE				
						-	MPMTCK.....	
						-	MPCONP.....	
						-	CVHVVE.....	
			-	CORIMT				
						-	MPMTCK.....	

Table 3-3 List of Subroutines called from MELGEN (5/9)

						-	MPCONP.....
						-	CORICG
						-	CORCDA
						-	CORICV
						-	CVHVVG.....
						-	CVHML.....
						-	CORNUM
						-	CORP2Y.....
						-	CORP2Z.....
						-	TPCIN.....
						-	CORPS2 .....
				-	INPDB2	.....	
				-	CRADB2	.....	
				-	CFDB2	.....	
				-	TFDB2	.....	
				-	SPRDB2	.....	
				-	MESDB2	.....	
				-	H2ODB2	.....	
				-	HSDB2	.....	
				-	SCDB2	.....	
				-	BURDB2	.....	
				-	DCHDB2	.....	
				-	CAVDB2	.....	
				-	MPDB2	.....	
				-	RN1DB2	.....	
				-	RN2DB2	.....	
				-	TPDB2	.....	
				-	FDIDB2	.....	
				-	ESFDB2	.....	
				-	BHDB2	.....	
				-	EDFDB2	.....	
				-	MEGPS3 : PASS 3 OF DATABASE GENERATOR PACKAGE		



Table 3-3 List of Subroutines called from MELGEN (6/9)

			-	MEGDB6	.....			
			-	NCGDB6	.....			
			-	CVHDB6	.....			
			-	FLDB6	.....			
			-	CVTDB6	.....			
			-	CORDB6	.....			
				: PASS 3 OF THE INPUT PROCESSING. Continue processing to include potential updating of the VOLU COR array with core boundary structure volume to include consideration of melting				
			-	CORDB7				
					-	CORPS3	.....	
			-	INPDB6	.....			
			-	CRADB6	.....			
			-	CFDB6	.....			
			-	TFDB6	.....			
			-	SPRDB6	.....			
			-	MESDB6	.....			
			-	H2ODB6	.....			
			-	HSDB6	.....			
			-	SCDB6	.....			
			-	BURDB6	.....			
			-	DCHDB6	.....			
			-	CAVDB6	.....			
			-	MPDB6	.....			
			-	RN1DB6	.....			
			-	RN2DB6	.....			
			-	TPDB6	.....			
			-	FD1DB6	.....			
			-	ESFDB6	.....			
			-	BHDB6	.....			

Table 3-3 List of Subroutines called from MELGEN (7/9)

			- EDFDB6	.....		
		-	MEGEDT			
		-	MXXRSW (write restart file)	: WRITES THE CONTENTS OF THE FILE IN UNIT NUINS ONTO THE FILE IN UNIT NREST. WRITES AN ENDFILE AND THEN BACKSPACES FILE NREST. THEN IT DELETES THE FILE IN UNIT NUINS		
		-	MXXRS (reads and writes restart files)			
		-	MXXPRS			
		-	MXXRIO			
		-	NCGPRS			
		-	NCGRIO			
		-	CVHPRS			
		-	CVHRIO			
		-	FLPRS			
		-	FLRIO			
		-	CVTPRS			
		-	CVTRIO			
		-	CORPRS	: WRITES/READS THE CORE ARRAY POINTERS AND DIMENSION TO/FROM THE RESTART FILE		
		-	CORRIO	: WRITES/READS THE SIZE-INDEPENDENT PORTION OF THE CORE PACKAGE DATABASE TO/FROM THE RESTART FILE.		
		-	INPPRS			
		-	INPRIO			
		-	CRAPRS			
		-	CRARIO			
		-	CFPRS			
		-	CFRIO			
		-	TFPRS			
		-	TFRIO			
		-	SPRPRS			

Table 3-3 List of Subroutines called from MELGEN (8/9)

		-	SPRIO						
		-	MESPRS						
		-	MESRIO						
		-	H2OPRS						
		-	H2ORIO						
		-	HSPRS						
		-	HSRIO						
		-	SCPRS						
		-	SCRIO						
		-	BURPRS						
		-	BURRIO						
		-	DCHPRS						
		-	DCHRIO						
		-	CAVPRS						
		-	CAVRIO						
		-	MPPRS						
		-	MPRIO						
		-	RN1PRS						
		-	RN1RIO						
		-	RN2PRS						
		-	RN2RIO						
		-	TPPRS						
		-	TPRIO						
		-	FDIPRS						
		-	FDIRIO						
		-	ESFPRS						
		-	ESFRIO						
		-	BHPRS						
		-	BHRIO						
		-	EDFPRS						
		-	EDFRIO						
		-	CRAEND (check unprocessed read images)						

Table 3-3 List of Subroutines called from MELGEN (9/9)

		-	CRARPG				
		-	CRAGCD	.....			

#### Table 4-1 Data Flow in MELCOR Main Program

Program MELCOR

Call MEXEDH

- Define the MELCOR header

Call MEXSET

- Setup (restart) MELCOR calculation

Call MEXRUN

- Execute MELCOR calculation

Stop

Table 4-2 List of Subroutines called from MELCOR (1/7)

MELCOR					
-	MEXEDI	: READS AND PROCESSES USER INFORMATION FOR TITLE, RESTART, END TIMES AND DTMAX, DTMIN, EDIT, PLOT AND RESTART FREQUENCY. DATA IN /MEXEI4/ IS USED IN SUBROUTINE MEXEDJ AND MEXDTC.			
-	MEXSET	: Setup (restart) MELCOR calculation			
-	MXXPBD	: THE MELCOR EXECUTIVE PSUEDO BLOCK DATA ROUTINE. IT HOLDS CONSTANTS AND CALLS OTHER PSUEDO BLOCK DATA ROUTINES			
	-	SYSNUM			
	-	NCGPBD			
	-	CVHPBD	.....		
	-	FLPBD	.....		
	-	CVTPBD	.....		
	-	CORPBD			
	-	CORVUP			
	-	MESERA	.....		
	-	BHPBD	.....		
	.....	.....			
-	MEXOPN	.....	: Open input, output and restart files		
-	INPPRE	: Process user input data file			
-	MESERA	.....			
-	MXXNAN	.....			
-	UTLUND				
-	MEXEDI	: Get user data for restart and edit			
	.....				
-	MEXRFM				
	-	MXXRS			
				-	MXXPRS
				-	MXXRIO
				.....	

Table 4-2 List of Subroutines called from MELCOR (2/7)

					- CORPRS : WRITES/READS THE CORE ARRAY POINTERS AND DIMENSION TO/FROM THE RESTART FILE
					- CORRIO : WRITES/READS THE SIZE-INDEPENDENT PORTION OF THE CORE PACKAGE DATABASE TO/FROM THE RESTART FILE.
					.....
					- EDFPRS
					- EDFRIO
					.....
					- MXXRS : Read restart file
					.....
					- CORPRS
					- CORRIO
					.....
					- MXXSCI : Read sensitivity coefficients
					.....
					- CORSC : PROCESSES THE CORE PACKAGE SENSITIVITY COEFFICIENT
					- CORSCY
					- CORSCX
					- MESERB
					.....
					- MEXUIN : Process rest of user input
					.....
					- CORDBI : PART OF THE CORE PACKAGE DATABASE MANAGER. IT INTERFACES THE DATABASE WITH THE DATA BLOCKS REQUIRED BY THE CORUIN SUBROUTINE.

Table 4-2 List of Subroutines called from MELCOR (3/7)

			- CORDBJ .....			
						: PART OF THE CORE PACKAGE DATABASE MANAGER. IT PULLS THE DATA ARRAYS FROM THE DATA BLOCKS REQUIRED BY THE CORUIN SUBROUTINE.
			.....			
		-	MEXCHK	:	Check for consistency in database	
			.....			
		-	CORDBG	.....		
			.....			
		-	CPRDBH	.....		
		-	MXRSW	:	Copy user input onto restart file	
		-	CRAEND	:	Release crackr storage files	
		-	MEXEDT			
			.....			
		-	CORDBE			
				:	PART OF THE CORE PACKAGE DATABASE MANAGER. IT PASSES THE CORE PACKAGE DATABASE TO THE EDIT ROUTINES.	
			.....			
		-	MXPLU	.....		
				:	GENERATE PRINTER LISTING OF KEY WORD DIRECTORY DELAYED UNTIL HERE SO THAT IN PLACE SORT CAN BE EMPLOYED MEXCFL NEEDS TO READ KEY IN STORED ORDER	
		-	MEXRUN	:	EXECUTE MELCOR CALCULATION	
		-	MEXSDO	.....		
		-	MESERS	.....		
		-	MEXUMS	.....		
		-	MXCPR	.....		
		-	MEXEND	.....		
		-	MEXEDT	:	Generate edit if correct time	
			.....			
		-	CORDBE			
				:	PART OF THE CORE PACKAGE DATABASE MANAGER. IT PASSES THE CORE PACKAGE DATABASE TO THE EDIT ROUTINES	



Table 4-2 List of Subroutines called from MELCOR (4/7)

			.....		
	-	MEXDTC .....			
		: Calculate new time step			
	-	MEXSD1 : Update statistics counters			
	-	MEXLOP : Advance physics package			
			.....		
		1. Communication array setup			
	-	MEXDBM			
	-	NCGDBM			
	-	CVHDBM	.....		
	-	FLDBM	.....		
	-	CVTDBM			
	-	CORDBM			
		: INITIALIZES ANY COMMUNICATION ARRAYS AT THE START OF A SYSTEM CYCLE. NO INITIALIZATION IS NECESSARY FOR THE CORE PACKAGE.			
	-	INPDBM			
	-	CRADB			
	-	CFDBM	.....		
	-	TFDBM			
	-	SPRDBM			
	-	MESDBM			
	-	H2ODBM			
	-	HSDBM	.....		
	-	SCDBM			
	-	BURDBM	.....		
	-	DCHDBM	.....		
	-	CAVDBM			
	-	MPDBM			
	-	RN1DBM	.....		
	-	RN2DBM	.....		
	-	TPDBM	.....		
	-	FDIDBM			

Table 4-2 List of Subroutines called from MELCOR (5/7)

		-	ESFDBM				
		-	BHDBM	.....			
		-	EDFDBM	.....			
			.....				
		2. PHYSICS PACKAGE					
		-	MEXDBC	.....			
		-	NCGDBC				
		-	CVHDBC	.....			
		-	FLDBC				
		-	CVTDBC				
		-	CORDBC	: PART OF THE CORE PACKAGE DATABASE MANAGER. IT PASSES THE CORE PACKAGE DATABASE TO THE PHYSICS RUN ROUTINES			
		-	MXXCPU				
		-	CORDBD	: PART OF THE CORE PACKAGE DATABASE MANAGER. IT PASSES THE CORE PACKAGE ARRAYS AND OTHER PACKAGES' ARRAYS TO THE RUN ROUTINES AND CONTROLS SUBCYCLING FOR THE CORE PACKAGE.			
				-	CORRNO	.....	
				-	MESSAG	.....	
				-	MEXDTB	.....	
				-	CORPSE	.....	
				-	CORRNB	.....	
				-	CORRN1	.....	
				-	CORRN2		
				-	CORRN3	.....	
				-	CORRN4	.....	
				-	BHORB4	.....	
				-	CORRN6		
				-	DBGSTP	.....	
		-	INPDBC				
		-	CRADBC				

Table 4-2 List of Subroutines called from MELCOR (6/7)

		-	CFDBC	.....			
		-	TFDBC				
		-	SPRDBC	.....			
		-	MESDBC				
		-	H2ODBC				
		-	HSDBC	.....			
		-	SCDBC				
		-	BURDBC	.....			
		-	DCHDBC	.....			
		-	CAVDBC	.....			
		-	MPDBC	.....			
		-	RN1DBC	.....			
		-	RN2DBC	.....			
		-	TPDBC	.....			
		-	FDIDBC	.....			
		-	ESFDBC	.....			
		-	BHDBC	.....			
		-	EDFDBC	.....			
		3. Last-chance fallback					
				.....			
		-	MEXDBO				
		-	NCGDBO				
		-	CVHDBO	.....			
		-	FLDBO				
		-	CVTDBO				
		-	CORDBO				
			: CORE PACKAGE				
			CHECK 'NEW' BEFORE ACCEPTANCE. CHANCE TO VETO				
			ADVANCEMENT. CORE PACKAGE DOES NOT HAVE ANY TESTS.				
		-	INPDBO				
		-	CRADBO				
		-	CFDBO				

Table 4-2 List of Subroutines called from MELCOR (7/7)

		-	TFDBO			
		-	SPRDBO			
		-	MESDBO			
		-	H2ODBO			
		-	HSDBO	.....		
		-	SCDBO			
		-	BURDBO	.....		
		-	DCHDBO			
		-	CAVDBO			
		-	MPDBO			
		-	RN1DBO			
		-	RN2DBO			
		-	TPDBO	.....		
		-	FD1DBO			
		-	ESFDBO			
		-	BHDBO			
		-	EDFDBO	.....		
			.....			
		.....				
		-	INPNBK			
		-	INPLNB			
		-	MESERA			
		-	MEXSTP : SET FLAG TO TERMINATE CALCULATION			
		-	MEX2PF .....			
			: check for executive package forced plot dump			
			.....			
		-	MEXSDO	.....		
			.....			

Table 4.1-1 Data Flow of MEXSET in MELCOR

\*DECK MEXSET

Call MXXPBD

- Generate pseudo block data

Call MEXOPN

- Open input, output and restart files

Call INPPRE

- Process user input data file

Call UTLUND

- Initialize undefined variables using xrealx

Call MEXEDI

- Get user data for restart and edit

Call MXXRS

- Read restart file

Call MXXSCI

- Read sensitivity coefficients

Call MEXUIN

- Process rest of user input

Call MEXCHK

- Check for consistency in database

Call MXXRSW

- Copy user input onto restart file

Call CRAEND

- Release CRACKR storage files

Call MEXEDT

- Create directory for control functions

Call MXXPLU

- Generate printer listing of key word directory

Return/End

Table 4.2-1 Data Flow of MEXRUN in MELCOR

\*DECK MEXRUN

Call MEXSDO:

- Saves various executive execution statistics

Call MESERS:

- Turn message level and deferred buffering process

10 CONTINUE

Call MEXUMS:

- Control for all interactive interrupt features

Call MXXCPR

- Checks remaining CPU time to see if there is sufficient time for another cycle.
- If there is not, set termination flags.

Call MEXEND

- Checks for end of problem

Call MEXEDT

- Part of MELCOR EXEC package to control editing.
- It does printing, plotting, and restarting output.

Call MEXDTC

- Calculates new time step for next cycle from data supplied by various packages and user input.

Call MEXSD1

- Updates statistics counters.

Call MEXLOP

- Driver for physics packages
- Advance physics package
- Advance time (TIME) and cycle (NCYCLE)

GO TO 10

Table 4.2.2-1 Data Flow of MEXLOP in MEXRUN (1/2)

\*DECK MEXLOP  
100 CONTINUE

\*\*\*\* START OF COMMUNICATION ARRAY SETUP LOOP \*\*\*

Call MEXDBM (and check NPKCUT to reduce time step)

- Defines the communication arrays for the MELCOR package.
- Zeros the global energy balance arrays.

Call NCGDBM (and check NPKCUT to reduce time step)

Call CVHDBM (and check NPKCUT to reduce time step)

Call CORDBM (and check NPKCUT to reduce time step)

- Initialize any communication arrays at the start of a system cycle

Call EDFDBM (and check NPKCUT to reduce time step)

- Defines the communication arrays for External Data File (EDF) Package

\*\*\* IF TIME STEP IS REQUIRED TO BE REDUCED AT ANY PACKAGES,  
THEN GO TO "100 CONTINUE" AND GO ON CALCULATION \*\*\*

\*\*\* END OF COMMUNICATION ARRAY SETUP LOOP \*\*\*

\*\*\* START OF PHYSICS PACKAGE LOOP \*\*\*

Call MEXDBC (and check NPKCUT to reduce time step)

- Performs a global energy balance calculation

Call NCGDBC (and check NPKCUT to reduce time step)

Call CVHDBC (and check NPKCUT to reduce time step)

Call CORDBC (and check NPKCUT to reduce time step)

- Passes the core package database to the physics run routines

Call EDFDBC (and check NPKCUT to reduce time step)

\*\*\* IF TIME STEP IS REQUIRED TO BE REDUCED AT ANY PACKAGES,  
THEN GO TO "100 CONTINUE" AND GO ON CALCULATION \*\*\*

\*\*\* END OF PHYSICS PACKAGE LOOP \*\*\*

Table 4.2.2-1 Data Flow of MEXLOP in MEXRUN (2/2)

Call MEXDBO

.

.

Call CORDBO

.

.

Call EDFDBO

\*\*\* CUT THE TIME STEP BY A SUGGESTED AMOUNT \*\*\*

Call MESDMP

- Print messages that are deferred until end of successful cycle completion.
- This routine processes file records for both MESSAG and MESERR packages.

Return/End



Table 5.1-1 Data Flow of CORDBC in MEXLOP

\*DECK CORDBC

Call MXXCPU

- find CPU time at beginning of core package execution

If IFIRST=0, calculate dimension on scratch arrays

Call CORDBD

- Passes the core package arrays and other packages' arrays to the run routines
- Controls subcycling for the core package
- 

Call MXXCPU

- Find CPU time at end of core package execution and  
calculate the total cumulative CPU time used by core Package

Return/End

Table 5.1-2 Data Flow of CORDBD in CORDBC

**\*DECK CORDBD**

Set pointers for scratch arrays if IFIRST=0

- Setup variable precision scratch arrays
- Scratch storage of core real database
- Scratch storage of control volume masses/energies/volumes

Set main database new/old time offsets

Reset short pointer names for use in CORRN\* call statements

\*\*\* START CORE PACKAGE CALCULATION \*\*\*

Call CORRNB

- Evaluate radiative heat transfer between the surfaces of the debris in the lower plenum and the surfaces above the debris

Call CORRN1

- Calculates core heat transfer and oxidation

Call CORRN2

- Controls the system and subcycle time step
- Defines the heat transfer rate to each control volume for the next system cycle

Call CORRN3

- Calls the CVH, HS, and RN1 interface routines, communicating energy and mass transfer information to the CVH package, energy transfer information to the HS package, and mass relocation information to the RN1 package

Call CORRN4

- Controls the modeling of the core degradation and relocation

\*\*\* SUCCESSFUL COMPLETION OF CORE PACKAGE SUBCYCLE \*\*\*

Call CORRN6

- Sets the global energy balance variables in common block CORGEB

Return/End

Table 5.1-3 Data Flow of CORRN1 in CORDBD (1/3)

\*DECK CORRN1

\*\*\* INITIALIZATION \*\*\*

\*\*\* CALCULATE DECAY HEAT AND FISSION POWER \*\*\*

Call RN1DHC

- Evaluate decay heat

Call CORDHD

- Redistribute decay heat by FRFIS and FXFIS

Call CORPOW

- Calculate the fission power for each cell and add it to the decay heat (DHEAT) array

\*\*\* CALCULATE HEAT TRANSFER TERMS \*\*\*

Call CORGAP (HFCA - total fuel-cladding HTC \* area (w/k))

- Calculate fuel-cladding gap conduction heat transfer coefficients

Call CORTSN

- calculate new-temperature distribution ( $dT/dz$ ) through each control volume in core/LP
- Estimate the new-time temperature of the cell components to use in a pseudo-implicit calculation of convective heat transfer later

Call CORTSV

- Calculate an approximate axial temperature distribution in each control volume in the core and lower plenum region

\* calculate radiation heat transfer for all cell components in exposed portion of core/LP

Call CORRDS

- Calculate the core emissivities and transmittances

Call CORRAD

- Calculate the radiation heat transfer terms in the core/lower plenum region (QRAD, QRLH, QRLP, QRADHS, QRADST)

\* calculate conductive heat transfer for all cell components at and above the liquid level in the core

Table 5.1-3 Data Flow of CORRN1 in CORDBD (2/3)

Call CORCTK

- Calculate an effective thermal conductivity of a component consisting of several layered materials

Call CORCND

- Calculate the cell to cell axial conduction heat transfer terms in the core region and above the liquid level (QCND((icl,ia,ir))

*\* calculate debris bed parameters for use in debris bed dryout correlation Eqs 6-13 in NUREG/CR-2646*

*\* calculate convective heat transfer for all cell component surfaces*

Call CORCNV

- Calculate convective heat transfer rates in the core for all cell components surfaces (QCNV(is,ia,ir), is=1,NSUR)

Call CORLHD

- calculate new temperatures of lower head and penetrations

Table 5.1-3 Data Flow of CORRN1 in CORDBD (3/3)

```

*** CALCULATE NEW ENERGY TRANSFER TO EACH COMPONENT, ITS NEW
    TEMPERATURE, MATERIAL ENERGIES, AND TOTAL HEAT CAPACITY
    USING THE TERMS PREVIOUSLY DETERMINED (Loop through Core/LP cells)
do IA (axial node)
do IR (radial node)

    QGAP(IA,IR) = HFCA(IA,IR) * (TMP(IFU,IA,IR) - TMPX(ICL,IA,IR))

* Calculate net energy source (NES) and update fuel state variables
  Call CORUV2

* Calculate NES and update cladding state variables

  Call BALOON (plan) determine whether clad balloons and failure
  Call CORRMX (plan) determine core flow redistribution after channel balloon
  Call COROXD
  - oxidize non-blocked portion of intact component surfaces (QOXIZ)
  - oxidize conglomerate debris zircaloy (QOXIZY)
  Evaluate net energy gain (loss) for cell components
    DENET(icl,ia,ir) = (DHEAT(icl,ia,ir) + QCND(icl,ia,ir) + QRAD(icl,ia,ir) +
      QGAP(icl,ia,ir) + QOXIZ + QOXIZY - QCNV(icl,ia,ir))* DTC
  Call COREU3
  - Determines new equilibrium temperature (TMP(icl,ia,ir)) and heat capacity using
    DENET.

* Calculate NES and update steel structure state variables
  Call COROXB
  Call COROXD
  - oxidize non-blocked portion of intact component surfaces
  - oxidize conglomerate debris zircaloy
  - oxidize conglomerate debris steel

* Calculate NES and update particulate debris state variables
  Call COROXD
  - oxidize non-blocked portion of particulate debris zircaloy
  - oxidize non-blocked portion of particulate debris steel
  - oxidize conglomerate debris zircaloy
  - oxidize conglomerate debris steel
  Call COROXB
  Call COREU3

* Call virtual volume interface routines
  Call CVHVVE

End do

```

Table 5.3-1 MELCOR COR Package Subroutines (1/10)

Subroutine Name	Version 1.8.3	Version 1.8.4	Function/Requirement
AACOR	o	o	Return COR package version number in text string text.
COR3PT	o	o	Part of package that dumps special records to plot file. It dumps COR related variables
CORABS	o	o	Determines the absorptivity and emissivity of steam
CORACX	o		Searches for an axial level and component to which small residual masses can be relocated
CORBAB		o	Adapted for MELCOR from CORA/BWR code development program
CORBAL	o	o	Calculates the global mass, energy, and volume balances for the core package.
CORBCP		o	Computes the integral of specific heat (CP) or CP/T
CORBCX	o	o	Evaluates reacting and remaining mass of B4C and H2O, and other gases (CO, CO2, CH4, H2, B2O3) generated from the oxidation reaction
CORBDV		o	Calculate gram-moles of each compound and heat of reaction in kJ
CORBGs		o	Adapted for MELCOR from CORA/BWR code development program
CORBHA		o	Adapted for MELCOR from CORA/BWR code development program
CORBHT		o	Computes the integral of specific heat (CP) or CP/T
CORBLK CORBLL CORBLM		o	Provides the runtime interface for flow blockage by the COR package. Consider core cells in levels IAMIN-IAMAX and rings IRMIN-IRMAX Flow direction is axial/radial if IZORR=+1/-1 Type of blockable path is flagged by IFBTYP Options are specified by IOPT; data are passed and returned in arrays IPARAM and RPARAM.
CORBXB		o	Adapted for MELCOR from CORA/BWR code development program
CORCAV	o	o	Converts TP packets retrieved by CAV from COR to CAV for LWR COR model. MELCOR > ... > CAVRUN > CAVDOP > CORCAV

Table 5.3-1 MELCOR COR Package Subroutines (2/10)

Subroutine Name	Version 1.8.3	Version 1.8.4	Function/Requirement
CORCDA	o	o	Calculates the surface area of conglomerate debris and the fraction of intact component area blocked by the conglomerate. The surface area of conglomerate debris is calculated in terms of a fraction of the intact component area.
CORCF1	o	o	Calculates the values for the COR-MMAT-CN.N and COR-MMAT-DB.N control function arguments
CORCF2	o	o	Calculates the argument value for the COR-AFLMIN.M.N control function.
CORCF3	o	o	Calculates the value for the COR-MLTFR.N.M.K control function argument
CORCFA	o	o	Part of the core package interface with the control function package. Returns the control function argument values to the control function package.
CORCFB	o	o	Calculates values for core control function arguments not directly accessible from the core database
CORCHK	o	o	Checks the database for internal consistency and consistency with other packages.
<b>CORCKB</b>		o	Checks the input for flow path blockage by the COR package Rings IRMIN-IRMAX and levels IAMIN-IAMAX have been specified as defining blockage for flow path IFLNUM, connecting control volumes ICVFM and ICVTO. If IZORR>0, it is an axial path, If IZORR<0 it is a radial path. Check that all core cells interface with 'from' or 'to' volume for IFLNUM and determine type of flow path. ENTRY CORBLK provides runtime interface. Options specified by IOPT; data passed and returned in arrays IPARAM and RPARAM.
CORCLL	o	o	Calculates convective heat transfer rates
CORCND	o	o	Calculates the cell to cell axial conduction heat transfer terms in the core region at and above the liquid LEVEL.C

Table 5.3-1 MELCOR COR Package Subroutines (3/10)

Subroutine name	Version 1.8.3	Version 1.8.4	Function/Requirement
CORCNV	o	o	Calculates the convective heat transfer rates in the core.
CORCOX	o	o	Calculates the rate constant for zircaloy or stainless steel oxidation.
CORCTK	o	o	Calculates an effective thermal conductivity of a component consisting of several layered materials
CORCVD	o	o	Evaluates heat or mass transfer coefficients for debris particles in atmosphere
CORDB1	o	o	Interfaces the database with the rest of the core initialization package. All arrays below this level are in core and passed through argument lists.
CORDB2	o	o	Part of the second pass generating the database for the core package. It passes down the package databases required for pass 2 input processing to CORDBY.
CORDB3	o	o	Passes the main database arrays down to subroutine CORDB5, which calculates the values of the pointers for the control function arguments.
CORDB4	o	o	Passes the main database arrays down to subroutine CORDB5, which calculates the values of the pointers for the control function arguments
CORDB5	o	o	Calculates the values of the pointers INARGA, INARGB, INARGC for the control function arguments
CORDB6	o	o	Pass 3 of the input processing. No additional processing is necessary for the core package.
CORDB7	o	o	Part of the 3rd pass of input processing for the COR package (part of BH until accepted into mainframe COR)
CORDBC	o	o	Passes the core package database to the physics run routines
CORDBD	o	o	Part of the core package database manager. It passes the core package arrays and other packages' arrays to the run routines and controls subcycling for the core package.
CORDBE	o	o	Part of the core package database manager. It passes the core package database to the edit routines
CORDBF	o	o	Part of the core package database manager. It passes the core package database down to the edit routine.
CORDBG	o	o	Part of the core package database manager. It checks the database for internal consistency and consistency with other packages.



Table 5.3-1 MELCOR COR Package Subroutines (4/10)

Subroutine name	Version 1.8.3	Version 1.8.4	Function/Requirement
CORDBH	o	o	Part of the core package database manager. It checks the database for internal consistency and consistency with packages.
CORDBI	o	o	Part of the core package database manager. It interfaces the database with the data blocks required by the CORUIN subroutine.
CORDBJ	o	o	Part of the core package database manager. It pulls the data arrays from the data blocks required by the CORUIN subroutine
CORDBM	o	o	Initializes any communication arrays at the start of a system cycle. No initialization is necessary for the core package.
CORDBO	o	o	Check 'new' before acceptance. Chance to veto advancement
CORDBT	o	o	Part of package that dumps special records to plot file. It dumps COR related variables and is part of COR.
CORDBU	o	o	Part of package that dumps special records to plot file. It dumps COR related variables and is part of COR
CORDBY	o	o	Part of the second pass in generating the database for the core package in MELGEN. It does the following tasks: 1. checks the user input with input from other packages 2. calculates additional quantities in the database that could not be calculated in the first pass
CORDBZ	o	o	It does the following tasks: 1. reads the user input 2. allocates data arrays 3. stores the core data in the data arrays
CORDDC	o	o	
CORDEF	o	o	Supply defaults and tests for core data input to DCH Contains data for decay heat which are core-model specific LWR COR model
CORDHC		o	Calculates the decay heat for each cell and adds it to the decay heat (DHEAT) array, and also includes gamma smearing
CORDHD	o		Is called when RNI is not active to redistribute decay heat through the core by the input power profile parameters FRFIS and FZFIS. Each cell receives decay heat by the formula $DHEAT(1,1,1)*XMFU(1,IA,IR)*FRFIS(IR)*FZFIS(IA)$ , but normalized to conserve energy. All debris receives an equal decay heat rate per unit mass of -DHEAT(1,1,1).

Table 5.3-1 MELCOR COR Package Subroutines (5/10)

Subroutine name	Version 1.8.3	Version 1.8.4	Function/Requirement
CORDRP	o	o	Calculates mass and energy relocation during candling (melt/flow/refreeze). The energy transferred from a molten film of material as it runs down a surface is based on the surface area, temperature difference between film and surface, and a refreezing heat transfer coefficient. Provision is made for flow blockage and formation of a molten pool in the core. A simple mechanism is also implemented to transport two additional materials by 'piggybacking' on the molten mass slug.
COREDTE	o	o	Print and plot editor for the core package
COREU0	o	o	Moves molten fabricated materials to the conglomerate debris array of the given component
COREU1	o	o	Calculates material dissolution in molten eutectic
COREU2	o	o	Calculates the relocation of molten eutectic mixtures during candling (flow/refreeze). The energy transferred from a molten film of material as it runs down a surface is based on the surface area, temperature difference between film and surface, and a refreezing heat transfer coefficient. Provision is made for flow blockage and formation of a molten pool in the core.
COREU3	o	o	Determines the new equilibrium temperature and the specific heat capacity for a core component consisting of fabricated materials XM FAB, conglomerate debris materials XMDC with internal energies ESI and incremental energy DEM
CORFMT		o	Print a single line of output to unit NOW with CHAR*NCHAR starting in column 2, followed by NCOL<9 columns of real data from XREL(*) in "preferred" format. Not suitable for negative numbers, because there may be no blank before the '-' sign. Column I of the output is to contain element XREL(IOUT(I)) if IOUT(I)>0 and blanks otherwise.
CORFRZ	o	o	Function (1., 1-exp(A)/A, 1./A)
CORGAP	o	o	Calculates the heat transfer rate from fuel to cladding across the gap. An effective total conductance is calculated which includes the thermal resistance of the fuel and the gas gap but neglects that of the cladding. The gas gap conductance includes a radiation contribution in parallel with the gas conduction, and allows the user to specify a control function that calculates a conductance in series with the gas conduction.
CORGEO	o	o	Redefines the cell component existence flags and calculates component surface areas

Table 5.3-1 MELCOR COR Package Subroutines (6/10)

Subroutine name	Version 1.8.3	Version 1.8.4	Function/Requirement
CORGPR	o	o	Evaluates gas properties at temperature TGAS: XMUG, REG, PRG, SCSG, SCOG, GRG Evaluates gas properties at temperature TFILM: TKONGF, DIFSB, DIFOB, XMUGF, REF, PRF, SCSF, SCOF, GRF (mixture viscosity and conductivity from MP package, 12/13/90)
CORGTM	o	o	
CORIAL	o	o	Returns the liquid level flag IALL from the cell atmosphere fractions XALL
CORICG		o	Part of pass 2 input processing for COR. It completes initialization of core geometry (A/V) and sets IEXIST. It also initializes the falling debris model. Constructed from parts of CORIVV
CORICM	o	o	Sets the flag ICMLT to 1 if TMP is within the melting range of some core material
CORICV		o	Part of pass 2 input processing for COR The interface to CVH is checked, and the initialization begun in CORVVE (called from CVHPS2 through CORVVD/VVE) is completed, including virtual volume in the CVH package. Constructed from parts of CORIVV/CORNUM and additional consistency
CORIMT	o	o	initializes the IMAT array, which points to the material properties package material identifiers for the core materials, and the constant properties arrays RHOM, TMLT, and ELHF
CORIVV	o		Checks the validity of the adjacent volume numbers input for the core package and initializes the virtual volumes for each volume. It also completes initialization of core surface areas and volumes
CORLHD	o	o	Calculates new temperatures of lower head and penetrations
CORLHF	o	o	Evaluates parameters regarding to lower head failure
CORLMP		o	Check mechanical failure of the lower head from creep rupture or through-wall material yielding
CORMCP	o	o	Calculates the total heat capacity of a structure
CORMEJ	o	o	Calculates the masses of core materials ejected from the vessel after failure of a lower head penetration
CORNPV	o	o	Calculates cell free volume for particulate debris relocation from current component and conglomerate volumes and effective 'packed' volumes occupied by components
CORNUM	o	o	Part of the pass 2 input processing for the core package. Numbering arrays and other interface arrays are defined and checked for validity.
COROX1		o	Is called before and after calls of COROXD and COROXB to check data and update oxidation variables
COROXB	o	o	Models the reaction of boron carbide (b4c) with steam. The reaction rate is calculated by the model developed by ORNL for MARCON (subroutine B4CRXK, renamed to CORBCX)

Table 5.3-1 MELCOR COR Package Subroutines (7/10)

Subroutine name	Version 1.8.3	Version 1.8.4	Function/Requirement
COROXD	o	o	Calculates the oxidation rate (and energy and hydrogen generation rates) for oxidation of zircaloy or stainless steel. The minimum of a solid-state diffusion rate and a gaseous diffusion rate is used.
CORP1A	o	o	It uses the CRACKR package to read the user input, checks for type consistency, prints any required diagnostic messages, sets flags to denote error conditions, and returns the input quantities to subroutine CORDBZ.
CORP1B	o	o	It uses the CRACKR package to read the user input for individual core cells, checks for type consistency, prints any required diagnostic messages and sets flags to denote error conditions.
CORP1D	o	o	It uses the CRACKR package to read the user input for lower head penetration and returns the input quantities to subroutine CORDBZ
CORP1F	o	o	It uses the CRACKR package to read the user input for lower head penetration and returns the input quantities to subroutine CORDBZ.
CORP1R	o	o	It uses the CRACKR package to read the user input for core radial rings, prints any required diagnostic messages and sets flags to denote error conditions.
CORP1Z	o	o	It uses the CRACKR package to read the user input for core axial levels, prints any required diagnostic messages and sets flags to denote error conditions.
CORP2D	o	o	It uses the CRACKR package to read the user input to define an inlet temperature control volume number or control function number for core control volumes and returns the input quantities to subroutine CORNUM.
CORP2Y		o	It uses the CRACKR package to read the user input to define the flow direction and ring area control functions for the CVH control volume IVOL
CORP2Z	o	o	It uses the CRACKR package to read the user input to define an inlet temperature control volume number or control function number for core control volumes and returns the input quantities to subroutine CORNUM.
CORPAE	o	o	Takes the sensible portion of the energy gain or loss and distributes it among various materials in a heat structure using an iterative procedure. This routine is called from subroutine CORPAR.
CORPAR	o	o	Takes the net energy gain or loss (DENET) for a heat structure, composed of several materials but represented by a single temperature, and distributes it among the various materials in the structure, allowing for phase changes and maintaining thermal equilibrium. An equilibrium temperature can also be calculated when forming debris from several structures having different temperatures.
CORPBD	o	o	Pseudo-block data subroutine for the core package. It initializes all database pointers and other core package variables and arrays
CORPOL	o	o	Polate is basically a linear tabular interpolation routine.
CORPOW	o	o	Calculates the fission power for each cell and adds it to the decay heat (DHEAT) array
CORPRS	o	o	Part of the core package database manager. It writes/reads the core array pointers and dimension to/from the restart file.
CORPS2	o	o	Part of the second pass used for generating the core package database

Table 5.3-1 MELCOR COR Package Subroutines (8/10)

Subroutine name	Version 1.8.3	Version 1.8.4	Function/Requirement
CORPS3	o	o	Purpose is to update the COR array element VOLU (LVTOT, IA, NRAD) with the volume occupied by the core shroud, to update the total COR package initial volume VOLTOI by the core shroud volume, and to increase the CVH total and virtual volume arrays (CVVOL and CVVVOL) by the shroud volume
CORPSE		o	The purpose of this routine is to adjust the pool elevation to take into account axial rings that may be blocked by core material
CORRAD	o	o	Calculates the radiation heat transfer terms in the core/lower plenum region
CORRBL		o	Calculates conversion of intact components in the core to particulate debris ("rubble") by various mechanisms including loss of strength and loss of support. It performs part of the calculations previously done in corslu, but leaves the resulting particulate in place. Extracted from CORSLU
CORRD1	o	o	Finds the next outer radial cell with an existing component in it and calculates its emissive power
CORRD2	o	o	Finds the next higher cell with an existing component and calculates its emissive power
CORRDG	o	o	Calculates net radiative heat flux from surfaces and heat flux from surface to medium
CORRDS	o	o	Calculates the core emissivities and transmittances
CORRIO	o	o	Part of the core package database manager. It writes/reads the size-independent portion of the core package database to/from the restart file
CORRN0	o	o	Calculates core lower head failure and debris ejection
CORRN1	o	o	Calculates core heat transfer and oxidation
CORRN2	o	o	Controls the system and subcycle time step and defines the heat transfer rate to each control volume for the next system cycle
CORRN3	o	o	Calls the CVH, HS, and RN1 interface routines, communicating energy and mass transfer information to the CVH package, energy transfer information to the HS package, and mass relocation information to the RN1 package
CORRN4	o	o	Controls the modeling of the core degradation and relocation.
CORRN6	o	o	Sets the global energy balance variables in common block CORGEB
CORRNB	o	o	Evaluates the BH debris to overlying surface radiation model
CORSC	o	o	Processes the core package sensitivity coefficient
CORSCX	o	o	Changes the value of sensitivity coefficient CXXXX(INT3) to the value real2 and edits the old and new values.
CORSHA	o	o	Calculates the sum (over components in a cell) of heat transfer coefficient (HTC) times surface area and HTC times surface area times temperature

Table 5.3-1 MELCOR COR Package Subroutines (9/10)

Subroutine name	Version 1.8.3	Version 1.8.4	Function/Requirement
CORSLU	o	o	Calculates slumping in the core by relocating mass to lower cells with free volume. This is a mechanical mechanism as opposed to the thermal mechanism in the candling routine CORDRP
CORSPD	o	o	Calculates radial particulate debris spreading in the core by relocating mass to lower cells with free volume. This is a mechanical mechanism as opposed to the thermal mechanism in the molten debris spreading routine CORSPR
CORSPR	o	o	Calculates radial relocation (spreading) of core debris to prevent unphysical 'stacking' in one ring
CORSRA	o	o	Sets elements in the NREL and FRACRL arrays for a single mass relocation.
CORTSN	o	o	Estimates the new-time temperature of the cell components to use in a pseudo-implicit calculation of convective heat transfer later
CORTSV	o	o	Calculates an approximate axial temperature distribution in each control volume in the core/lower plenum region.
CORUIN	o	o	Passes mass, energy, and temperature arrays to the thermal equilibration routine CORPAR. Entry points are provided for various numbers of equilibrating masses
CORUVN	o	o	Passes mass, energy, and temperature arrays to the thermal equilibration routine CORPAR. Entry points are provided for various numbers of equilibrating masses.
CORVVF		o	Part of the radiation model in the COR package. It calculates surface-to-surface view factors for components in a single core cell. Extracted from logic implicit in subroutine CORRA
CORVLS	o	o	Resets the core package volume arrays at the end of a subcycle based on end-of-subcycle masses. Volume error checking in subroutine CORBAL therefore will only detect volume errors occurring during that subcycle before the volumes are reset here.
CORVPR	o	o	Calculates the density and specific heat capacity of the atmosphere in a control volume
CORVUP	o	o	Sets latest applicable dates and version numbers for the core package coding, users' guide, and reference manual for printing to output
CORVVC	o	o	Part of a core package utility called from CVH that calculates changes in virtual volume. The utility is called before CVH initializes its thermodynamics state
CORVVD	o	o	Part of a core package utility called from CVH that calculates changes in virtual volume. The utility is called before CVH initializes its thermodynamics state. CORVVD passes parts of the CVH and COR databases to CORVVE

Table 5.3-1 MELCOR COR Package Subroutines (10/10)

Subroutine name	Version 1.8.3	Version 1.8.4	Function/Requirement
CORVVE	o	o	Part of a core package utility called from CVH that calculates changes in virtual volume. The utility is called before CVH initializes its thermodynamics state. CORVVE calculates the virtual volume changes and passes them to the CVH database through a call to CVHVVE
CORXBP		o	Models the reaction of boron carbide (B4C) with steam. The reaction rate is calculated by the model developed by ORNL for MARCON (subroutine B4CRXK, renamed to CORBCX)
CORXCH	o	o	Performs the mass, energy and volume bookkeeping when transferring conglomerate debris from the 'other structure' component to conglomerate on particulate debris
CORXRI	o	o	Copies a referenced integer value into a storage location making the reference and sends an error message if the referenced value is not valid.
CORXRN	o	o	Supply core-model dependent data, and data and pointers derived from core data input to RN1 LWR COR version
CORXRR	o	o	Copies a referenced real value into a storage location making the reference and sends an error message if the referenced value is not valid.
CORYCH	o	o	Performs the mass, energy and volume bookkeeping when transforming intact fuel to particulate debris
CORZCH	o	o	Performs the mass, energy and volume bookkeeping when moving a fixed fraction of the materials in a cell component to another cell or component
CORZSB		o	Initializes the chemical activity flags for Zirc, Steel, and B4C oxidation of the various surfaces of a COR cell IA,IR at the beginning of a subcycle and  initializes the cumulative steam consumption over the subcycle of each surface in that cell.  It also defines the total chemically active surface area of COR components in IA,IR exposed to the CVH materials of CVH volume ICV
DBGSTP	o	o	Stop calculation from debugger run
ELHEAT	o	o	Stop calculation due to no heating element power routine
EUPROP	o	o	Calculates material properties for eutectic mixtures
SOLPRO	o	o	Calculates thermophysical properties for eutectic mixtures

Table 5.3-2 MELCOR CVH Package Subroutines (1/5)

Subroutine Name	Version 1.8.3	Version 1.8.4	Function/Requirement
AACVH	o	o	Return CVH package version number in text string.
CVH3PT	o	o	Part of package that dumps special records to plot file. It dumps hydro related variables.
CVHADV	o	o	Updates the contents of control volumes to account for advection by adding contents of flow path volume moves
CVHALP	o	o	Evaluates average void fractions in flow paths
CVHBB1	o	o	Calculates the mass and energy moved by bubble separation and their derivatives with respect to pool void fraction. It also calculates the derivatives of pressure and of pool void fraction with respect to mass and energy, including the implicit effect of bubble separation.
CVHBB2	o	o	It calculates the linearized final pool void fraction and adds implicit corrections to bubble separation
CVHBFX	o	o	Post-advection 'fix-up' for excessive pool void fraction. If pool void exceeds limit (which should be consistent with limit in subroutine CVHBL), transfer excess bubbles to atmosphere.
CVHBL	o	o	Calculates the mass and energy transport between the pool and atmosphere in the non-equil thermo volumes
CVHBV1	o	o	Defines new conditions in boundary volumes
CVHBV2	o	o	Accounts for effective mass/energy sources in boundary volumes
CVHBV3	o	o	Defines inventories to calculated donor densities, differently for ordinary and boundary volumes
CVHBVL	o	o	Tests legality of specification for time-specified volume
CVHBVT	o	o	Defines new conditions in a boundary volume
CVHBVV	o	o	Retrieves value of variable for time-specified volume
CVHCAR	o	o	CVHCAR copies the old set of data into the new set. This is done in a subroutine to collapse the two or more subscripts into one. This makes much more efficient do loops
CVHCCN	o		Subroutine CVHCCN replaced by CVHADV under DIR 1164
CVHCFA CVHCFB CVHCFC	o	o	Returns the values of the arguments in the control volume hydro internal database
CVHCON	o	o	Convects material through the flow paths $x_{new} = x_{old} + \text{convected cold (quantity to be convected)}$
CVHDB1	o	o	Interfaces the database with the rest of the control volume initialization package. All arrays below this level are in core and passed through argument lists.
CVHDB2	o	o	Interfaces the main database blocks with the lower level packages
CVHDB3	o	o	Passes the main database arrays down to the subroutine CVHDB5 that calculates the values of the pointers for the control function arguments
CVHDB4	o	o	Passes the main database arrays down to the subroutine CVHDB5 that calculates the values of the pointers for the control function arguments
CVHDB5	o	o	Calculates the values of the pointers for the control function arguments
CVHDB6	o	o	Interfaces the main database blocks with the lower level packages
CVHDB7	o	o	Part of the database management package for the third pass for generating the database of the control volumes



Table 5.3-2 MELCOR CVH Package Subroutines (2/5)

Subroutine name	Version 1.8.3	Version 1.8.4	Function/Requirement
CVHDBC	o	o	Part of the control volume hydrodynamics database manager
CVHDBD	o	o	Part of the control volume hydrodynamics database manager
CVHDBE	o	o	Part of the control volume edit package database manager. This package pulls out the data blocks of the control volume
CVHDBF	o	o	Calculate of cumulative CPU for edit
CVHDBG	o	o	Part of the control volume hydrodynamics package. It checks the CVH package's database against the other packages database.
CVHDBI	o	o	Part of the control volume database management package. Interfaces the database with the data blocks required by the CVHUIN subroutine
CVHDBJ	o	o	Part of the control volume database management package. Pulls the data arrays from the data blocks required by the CVHUIN subroutine
CVHDBM	o	o	Part of the control volume hydrodynamics database manager
CVHDBN	o	o	Initialize the communication arrays
CVHDBO	o	o	Check 'new' before acceptance. Chance to veto advancement
CVHDBP	o	o	Part of the control volume hydrodynamics database manager
CVHDBT	o	o	Part of package which dumps special records to plot file. Dumps hydro related variables
CVHDBU	o	o	Part of package that dumps special records to plot file. Dumps hydro related variables
CVHDBY	o	o	Part of the control volume data base generate package. Decomposes the control volume database block into it elements
CVHDBZ	o	o	Part of the control volume setup package. 1. reads the user input 2. allocates data arrays for the control volumes. 3. stores the control volume data in the data arrays.
CVHDL	o		Subroutine CVHDL absorbed into CVHDVL under DIR 1164
CVHDNR	o	o	Evaluates the signed mass and energy donor densities for flow path volume moves
CVHDON	o	o	Defines modified volumes and enthalpies to reflect addition of sources at constant pressure and temperature May repartition water between fog and atmospheric water vapor
CVHDRV	o	o	Evaluates the derivatives of volume pressures with respect to flow path volume moves
CVHDTC	o	o	Calculates the new time step
CVHDVL	o	o	Evaluates the volume moves through flow paths (limited by availability), fractions of donor inventories moved, and material courant time step limit
CVHEDH	o	o	Defines the version number of the control volume hydrodynamics package
CVHEDT	o	o	Print and plot editor for CVHYDRO package of MELCOR
CVHFOG	o	o	Relocates fog (suspended liquid water droplets) in the atmosphere when necessary
CVHFRF	o	o	Calculates the friction factors for the flow paths
CVHJUN	o	o	Evaluates geometric void fractions in volumes at junctions and average elevations of phases within junction openings

Table 5.3-2 MELCOR CVH Package Subroutines (3/5)

Subroutine name	Version 1.8.3	Version 1.8.4	Function/Requirement
CVHMDR	o	o	Updates the mass sources and sinks, energy sources and sinks
CVHMDS	o	o	Initializes the communication arrays at the start of a cycle.
CVHMEB	o	o	Updates mass/energy accounting arrays
CVHMOM	o	o	Performs solution of linearized-implicit flow equations
CVHMT2	o	o	Modifies the implicit terms in the coefficient matrix to atmosphere velocity = pool velocity
CVHMTR	o	o	Evaluates the coefficient matrix for coupling of flows through pressure and head changes in connected volumes
CVHMTX	o		Subroutine CVHMTX replaced by CVHMTR under DIR 1164
CVHMLV	o	o	Calculates the volume of the control volume hydrodynamics material in control volume number NVOLS(n) between altitudes ZLWS(n) and ZUPS(n). The answer is returned in variable VOL(n).
CVHMXM	o		Subroutine CVHMXM eliminated under DIR 1164
CVHNUM		o	Given an internal number (index), it returns the user number for a control volume. It was written to allow the user number to be obtained for use in warning and error messages without adding the ICVNUM array to utility calls.
CVHPIT	o	o	Processes the thermo input
CVHP2A	o	o	Calculates the masses and energies of the atmosphere during pass 2 input processing, given the atmosphere pressure, temperature H <sub>2</sub> O partial pressure, relative humidity or dew point temperature, volume, fog volume fraction or mass, and NCG mole fractions.
CVHP2P	o	o	Calculates the mass, energy, and volume of the pool during pass 2 input processing, given the pool pressure, temperature, and either volume, liquid level, or mass.
CVHPAD	o	o	CVHPAD defines the PPRN array If component volume is zero (as from round-off), density will be zero even if mass (and all other properties) are non-zero
CVHPBD	o	o	This pseudo-block data routine is part of the control volume package. It is part of the code that interfaces the database management scheme with the control volume package.
CVHPRS	o	o	Writes the control volume hydrodynamics database manager information to the restart file
CVHPS1	o	o	Uses the CRACKT package to read the user input, checks for type consistency, prints any required diagnostic messages and sets flags to denote error conditions.
CVHPS2	o	o	Second pass of input processing that defines the thermodynamic data.
CVHPS3	o	o	Initialize values based on control functions
CVHPSA	o	o	Part of the pass 2 package Calculates the value of $da$ , $etotln(k)$ , $k=2,nummat$ .
CVHPSP	o	o	Part of the pass 2 package Calculates the value of $dp$ and $etotln(1)$ for liquid water in the pool

Table 5.3-2 MELCOR CVH Package Subroutines (4/5)

Subroutine name	Version 1.8.3	Version 1.8.4	Function/Requirement
CVHRCN	o	o	Uses the CRACKR package to read the user input, checks for type consistency, prints any required diagnostic messages and sets flags to denote error conditions. Reads the input for the external source data
CVHRIO	o	o	Reads and writes the database of the control volume hydrodynamics package
CVHRN1	o	o	Calculates the external sources of mass and energy for the control volume hydrodynamics package.
CVHRN2	o	o	Part of the flow path package It calculates the fraction open of the flow paths.
CVHRN3	o	o	Integrates the governing equations through system time step
CVHRN4	o	o	Calculate flow velocity in the control volume, liquid water level and then array PAPRN which contains pointers to the pool/atmosphere/control volume hydrodynamic properties
CVHRN5	o	o	Push integral flows through flow paths to EDF, if required
CVHSC	o	o	Processes sensitivity coefficients
CVHSP1	o	o	Modifies donor densities for acceptor volumes to account for bubble interactions (SPARC physics) within flow paths
CVHSPC	o		Subroutine CVHSPC replaced by CVHSPR under DIR 1164
CVHSPK	o	o	Modifies the signed mass and energy donor densities for flow path volume moves to account for SPARC bubble physics
CVHSPU	o	o	Calculates the bubble physics through a pool of water
CVHSTR	o	o	Calculates the linearized post-source state of the volumes
CVHTKT	o	o	Transfer the information from the PKCON array if mass or energy has changed then set ICVFLG(KCVACT,*) to 0 (activate cell)
CVHUIN	o	o	Read user input
CVHVEL	o	o	Evaluates volume-average mass flows and velocities and from/to mass flows Motivated by RELAP5-MOD1 model, but based on sum of absolute values of all flows rather than average of separate algebraic sums of 'from' and 'to'.
CVHVLP		o	Used by the BH package to change the virtual volume in the LP control volume. It does this by modifying the CVMVLN value in the CVMVLN-altitude table for a control volume This routine was created from CVHVVE
CVHVMT	o		Used by other packages to change the virtual volume in control volumes. It does this by modifying the CVMVLN values in the CVMVLN-altitude table for a control volume
CVHVVE		o	Calculates virtual volume
CVHVVG	o	o	Used by other packages to add virtual volume to the volume - altitude table for a control volume. User must supply arrays in CVH database and a virtual volume-altitude table

Table 5.3-2 MELCOR CVH Package Subroutines (5/5)

Subroutine name	Version 1.8.3	Version 1.8.4	Function/Requirement
<b>CVHWP</b>	o		Subroutine CVHWP eliminated under DIR 1164
<b>CVHWPM</b>	o		Subroutine CVHWPM eliminated under DIR 1164
<b>CVHZCP</b>	o	o	Calculates the collapsed liquid level
<b>CVHZPL</b>	o	o	Calculates the swollen liquid level

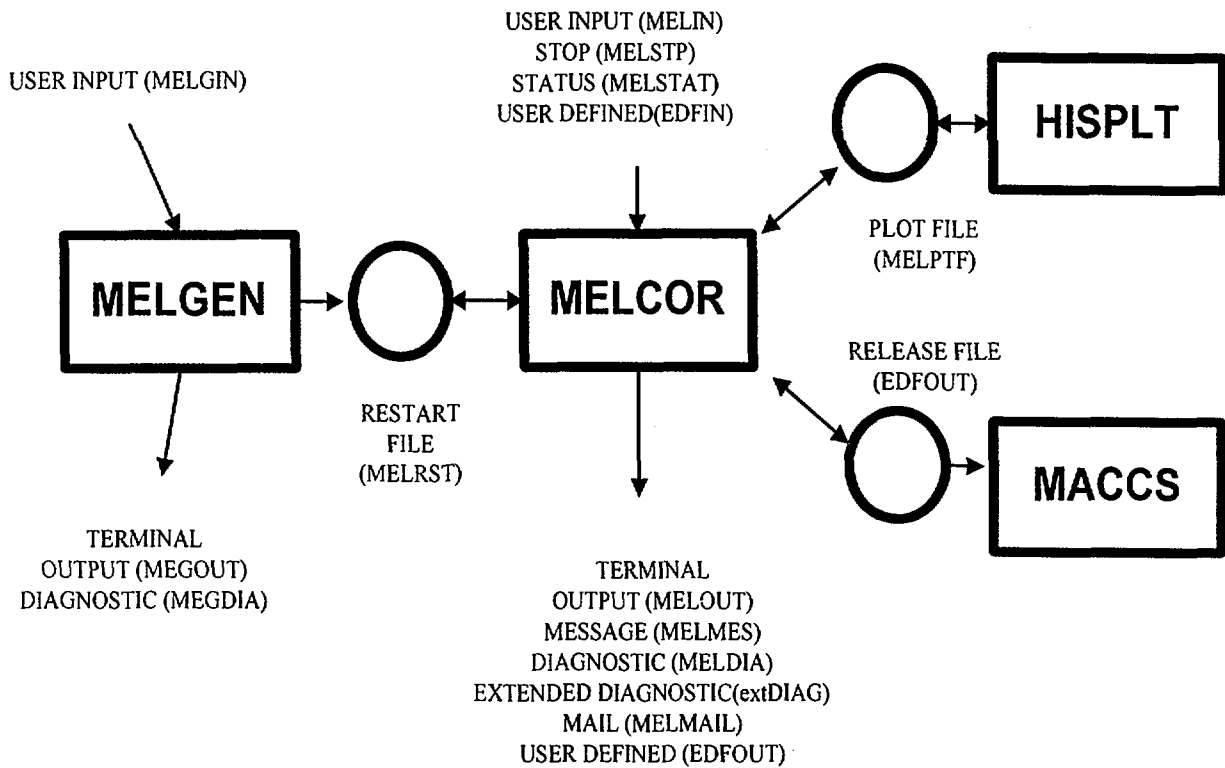


Figure 2-1 MELCOR Code and File Relations

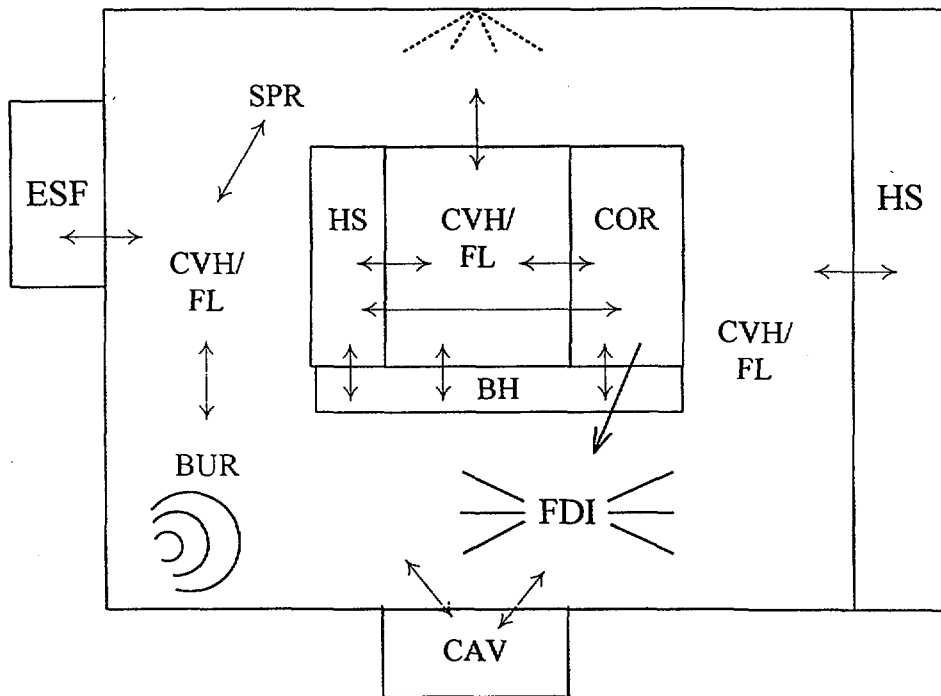


Figure 2-2 Interactions Between Physics Packages

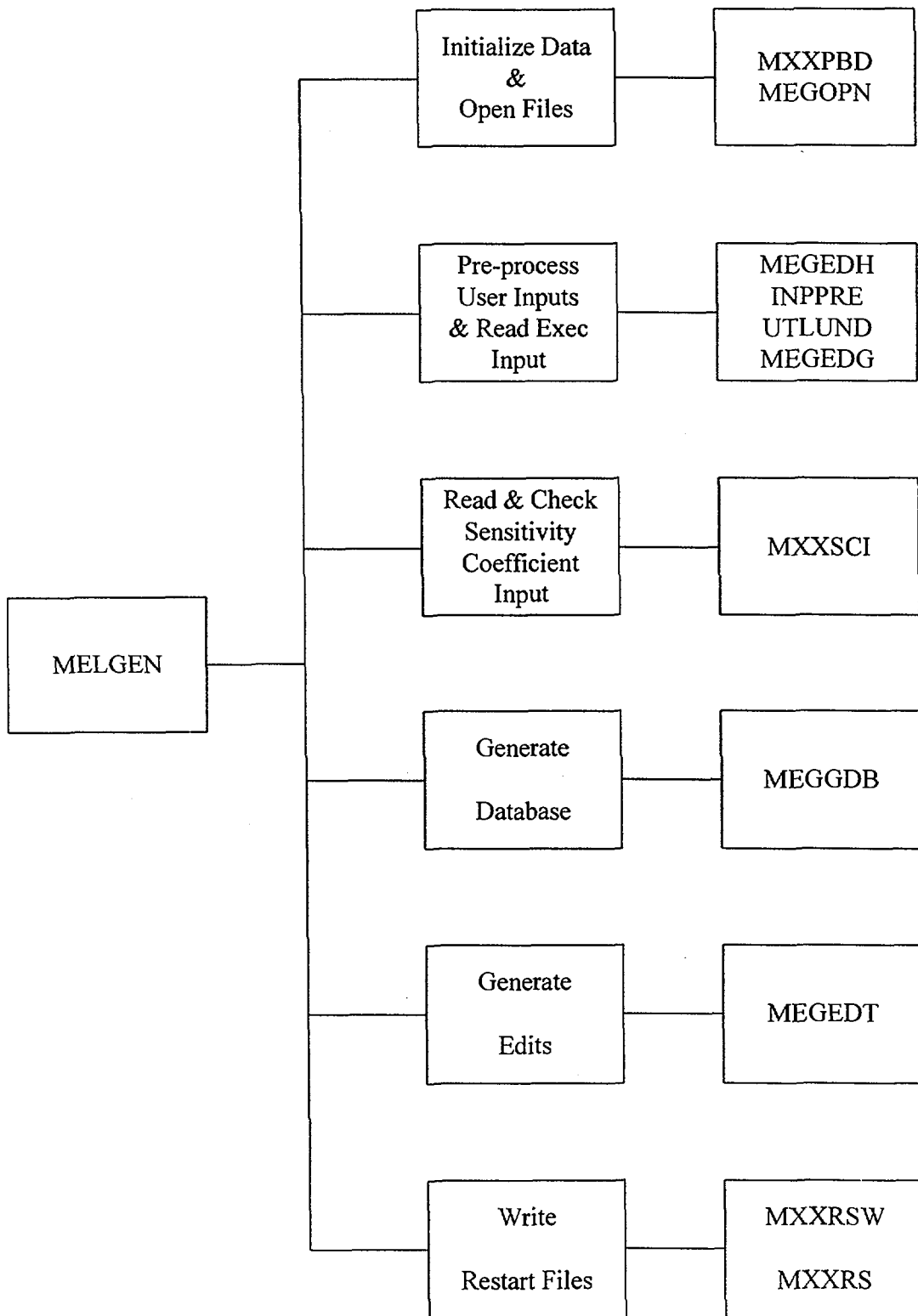


Figure 3-1 Function of MELGEN and Its Related Subroutines

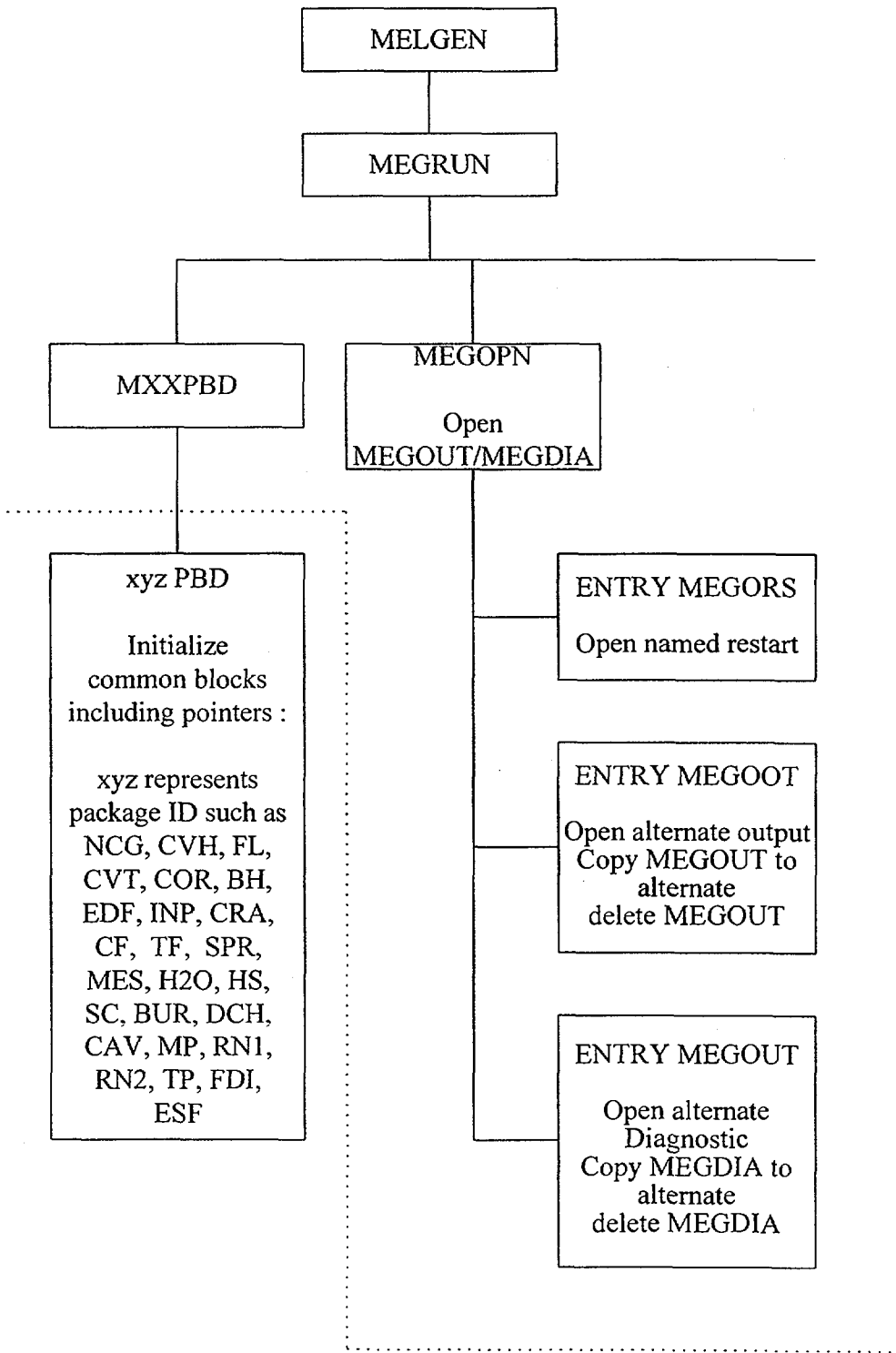


Figure 3.1-1 Data Initialization and File Opening in MELGEN



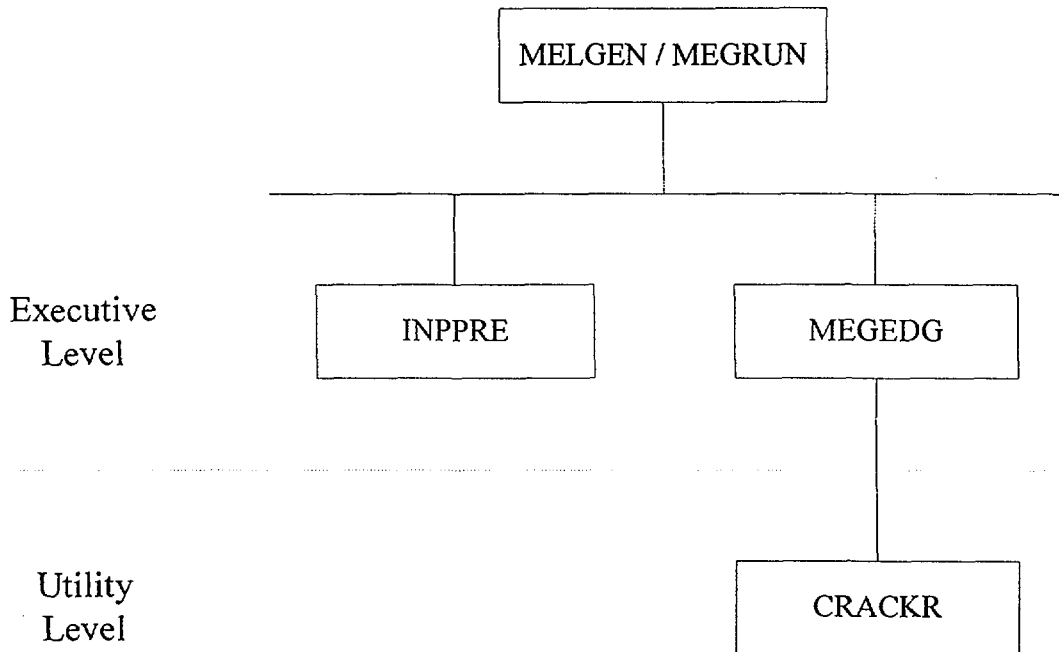


Figure 3.2-1 Input Processing in MELGEN

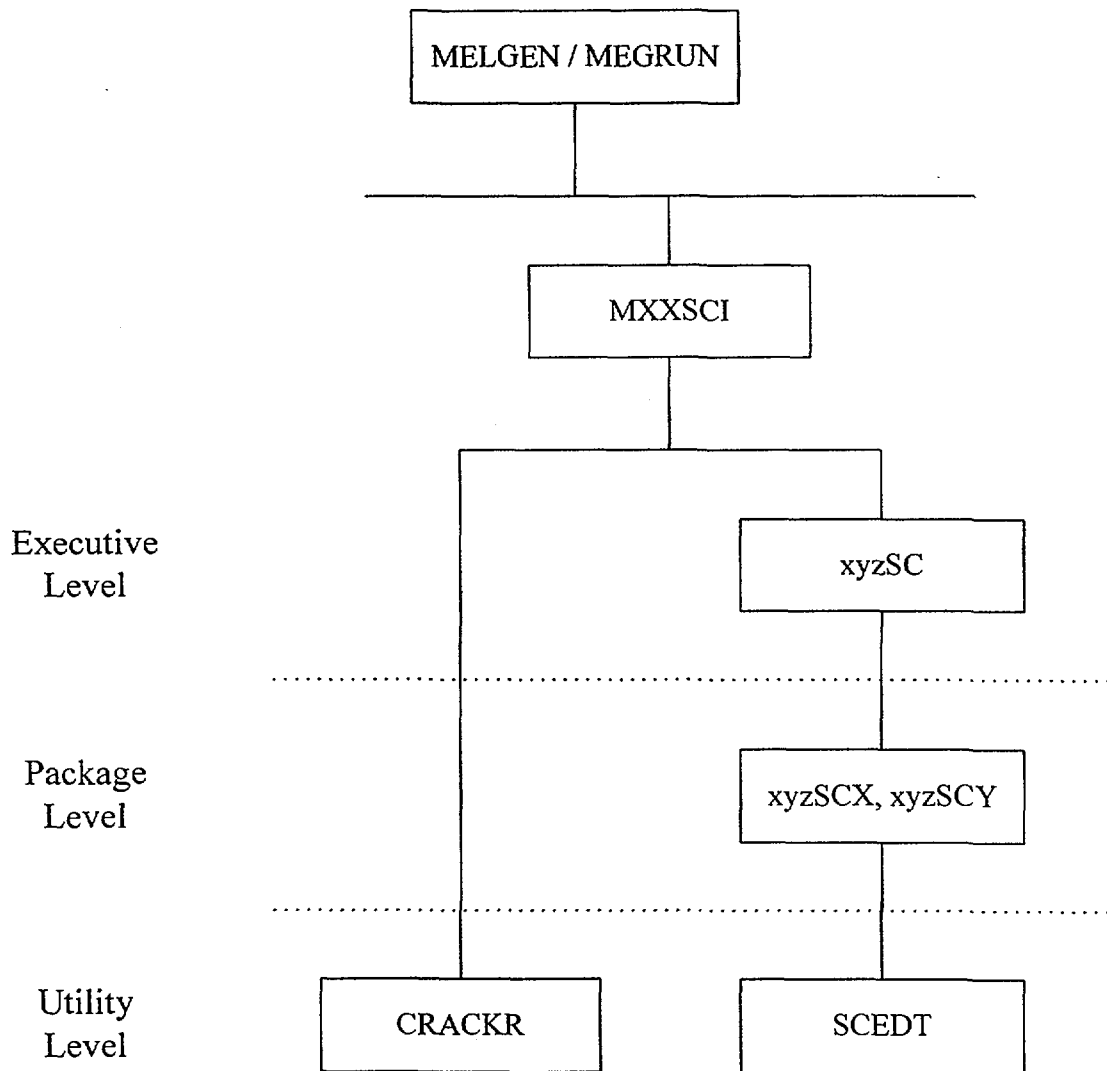


Figure 3.3-1 Sensitivity Coefficient Check in MELGEN

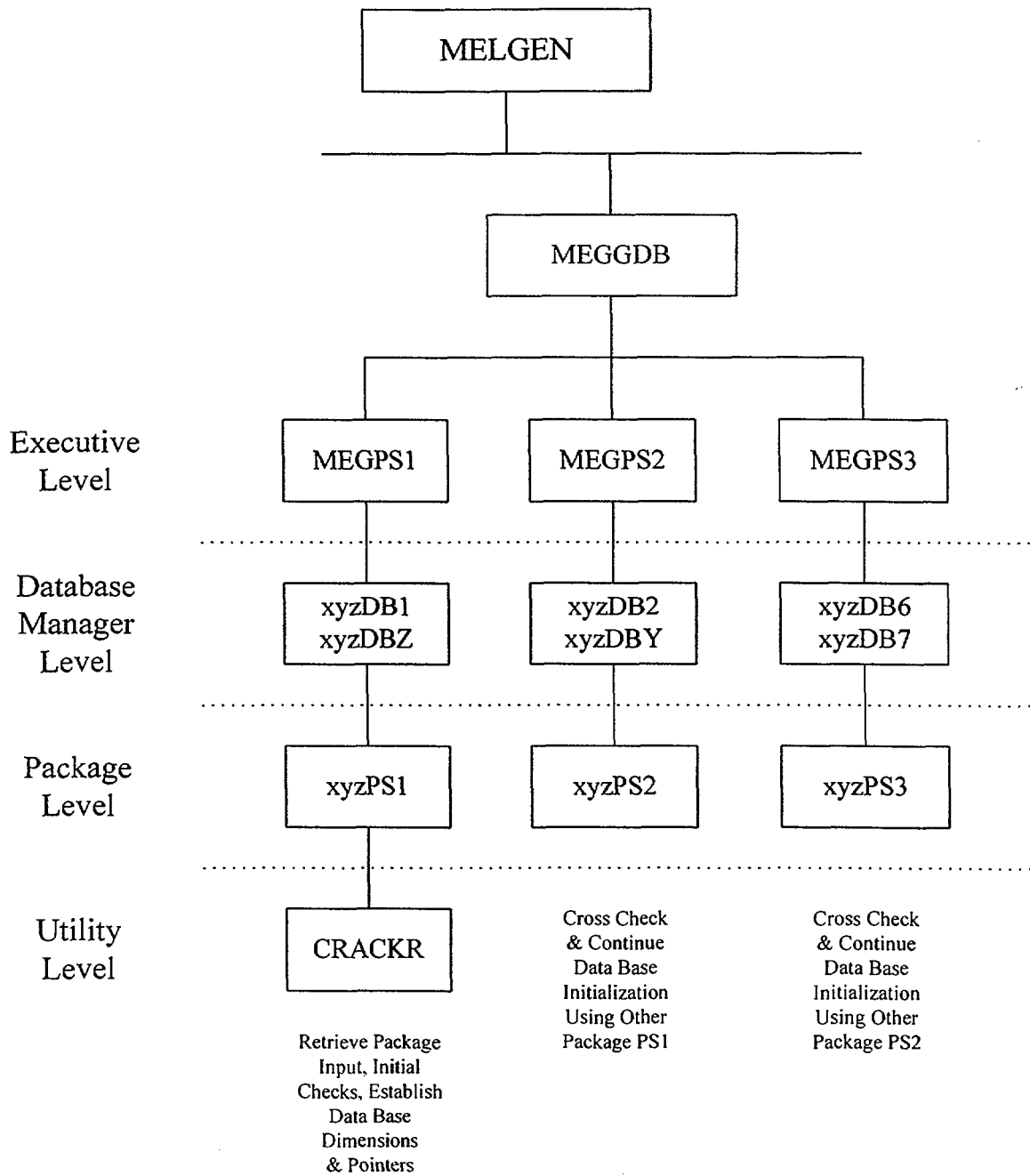


Figure 3.4-1 Database Generation in MELGEN

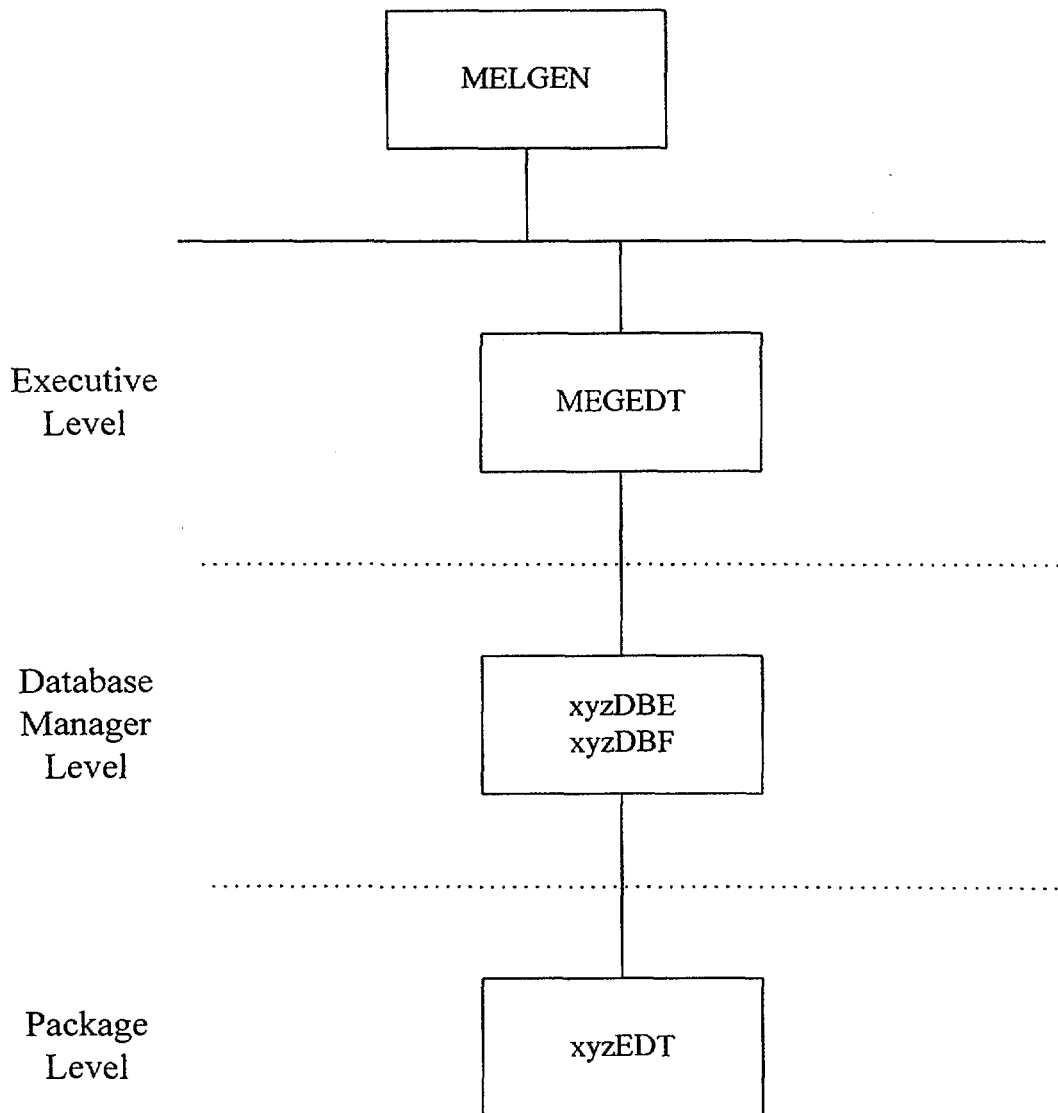


Figure 3.5-1 EDIT Generation in MELGEN

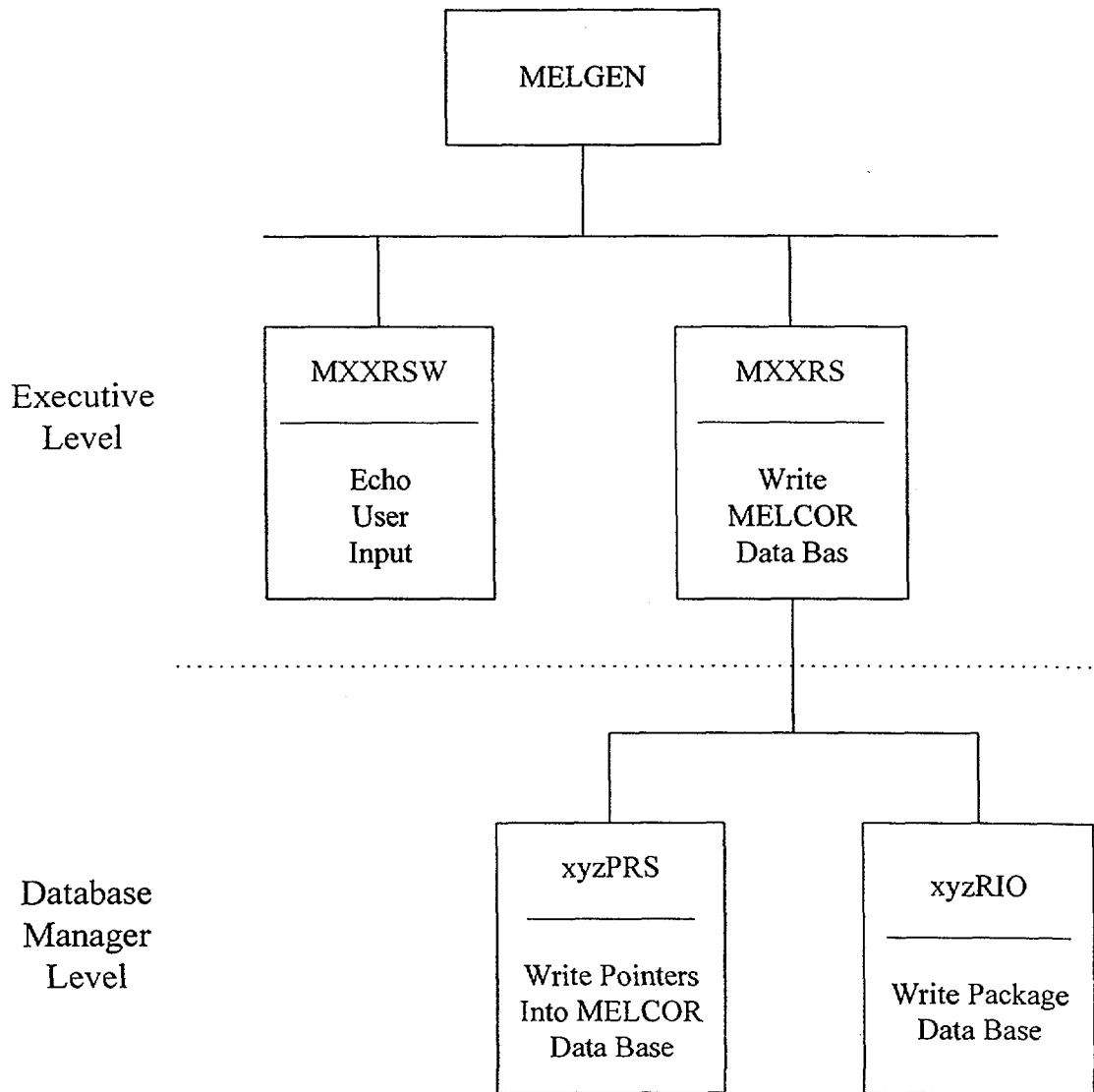


Figure 3.6-1 Restart File Generation in MELGEN

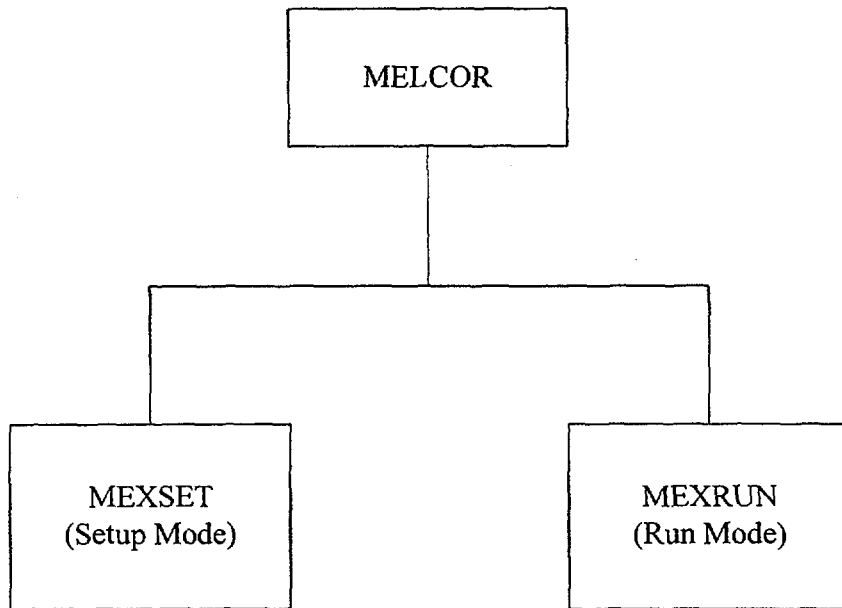


Figure 4-1 Two Functions of MELCOR

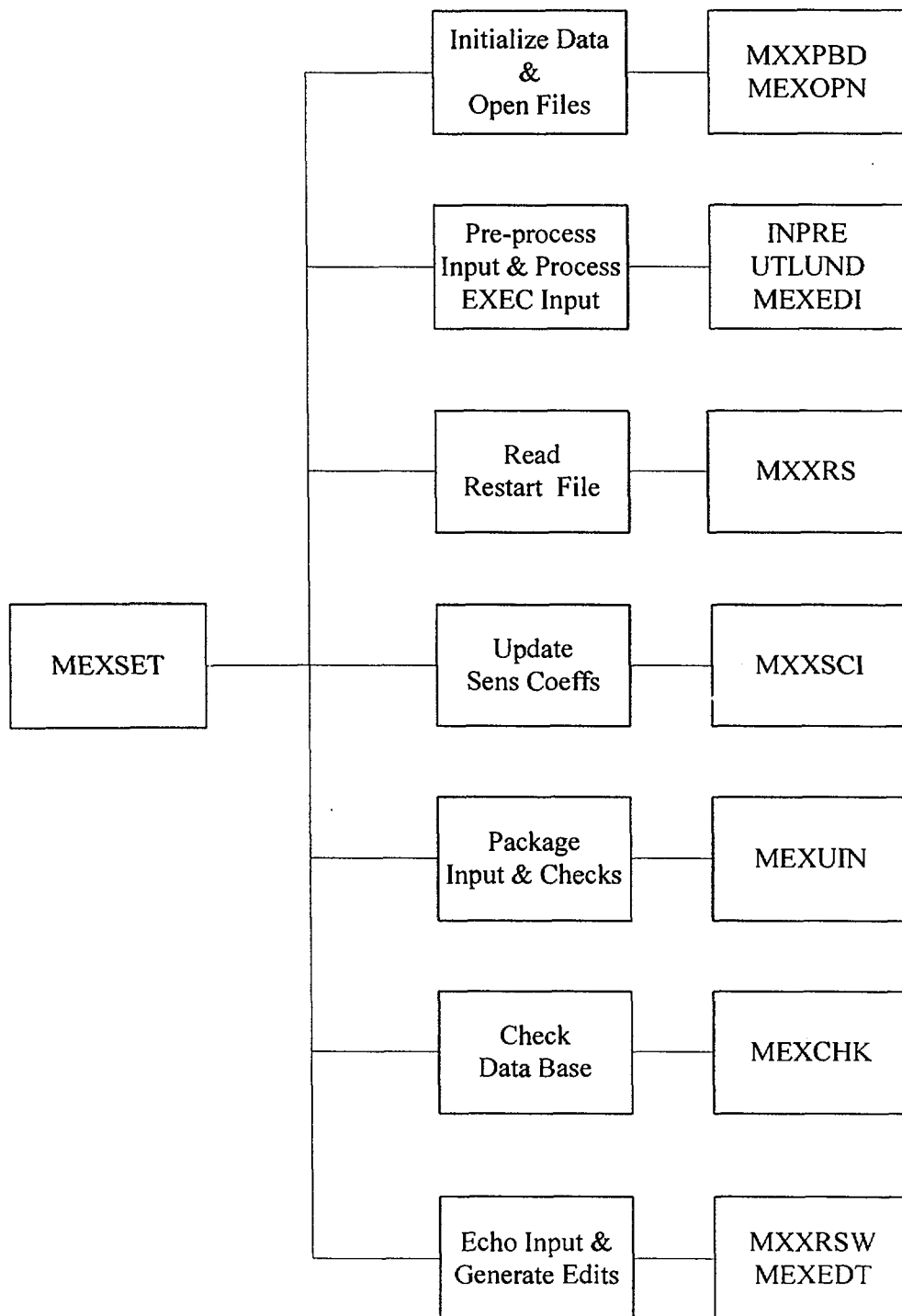


Figure 4.1-1. Functions of MELGEN and Its Related Subroutines

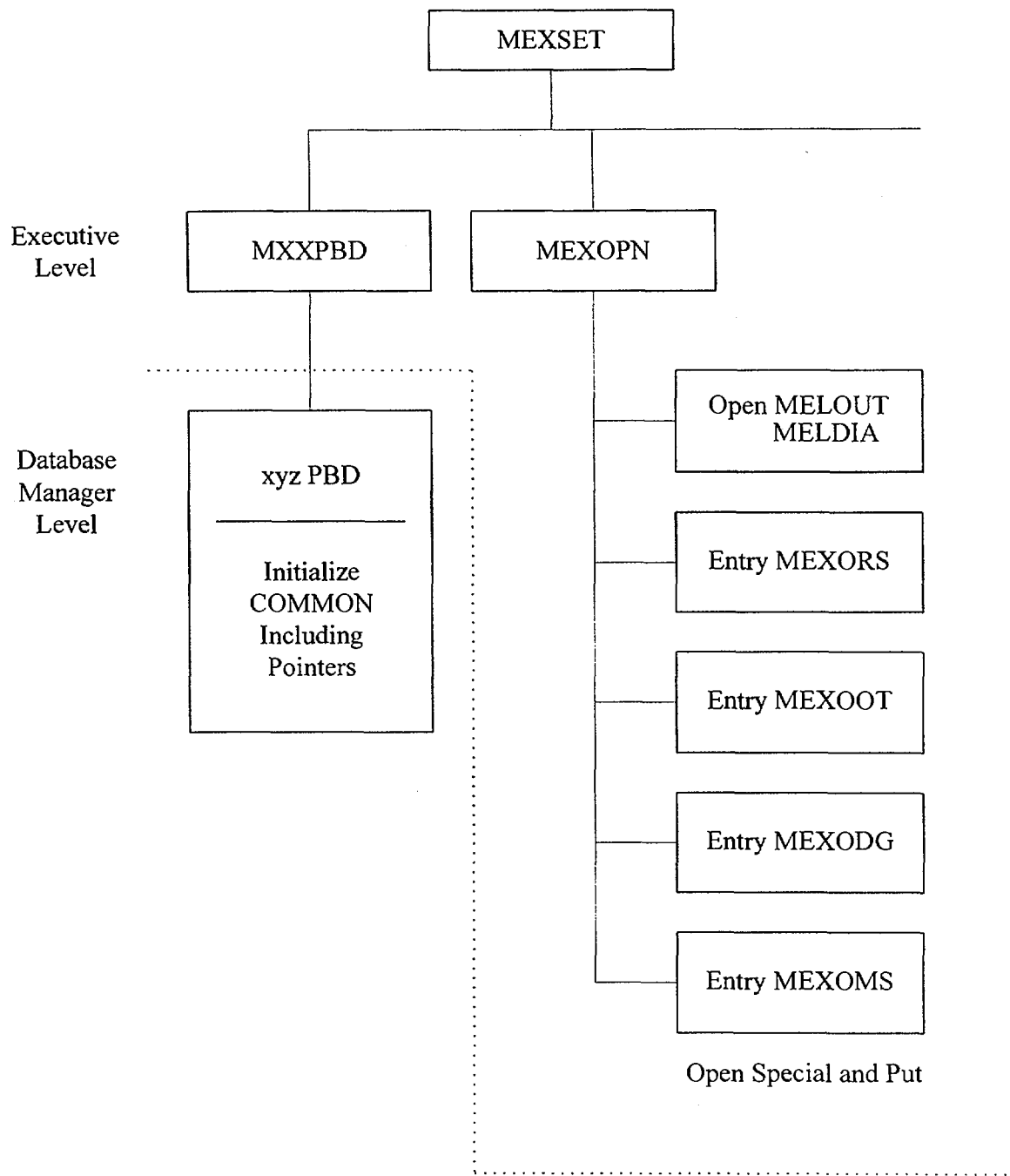


Figure 4.1.1-1 Data Initialization and File Opening in MEXSET



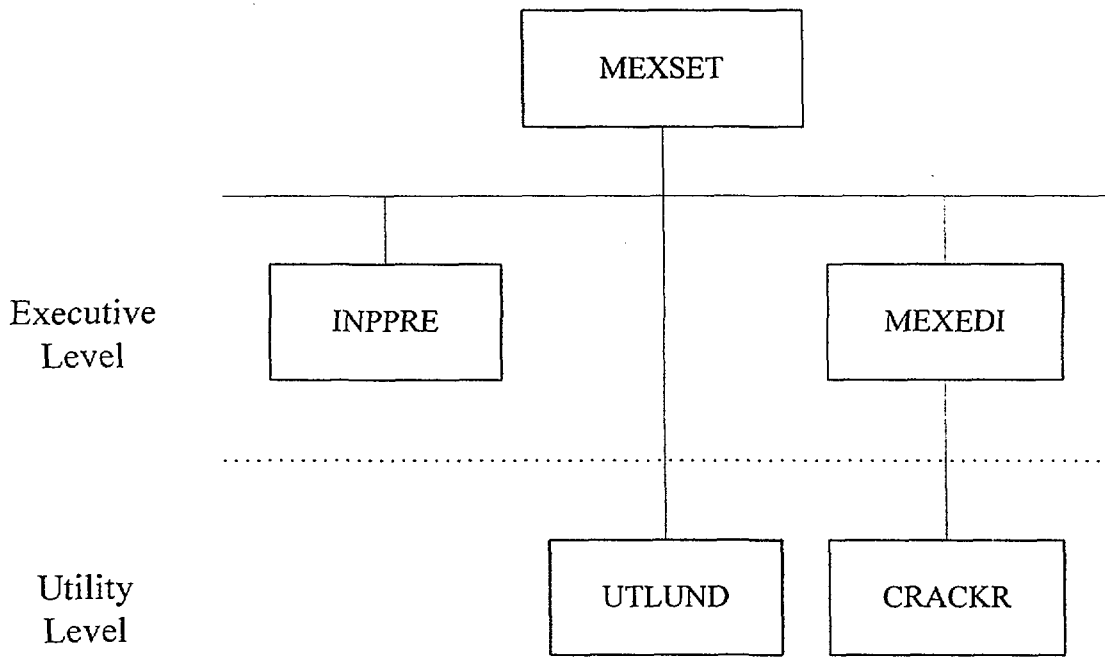


Figure 4.1.2-1 Input Processing in MEXSET

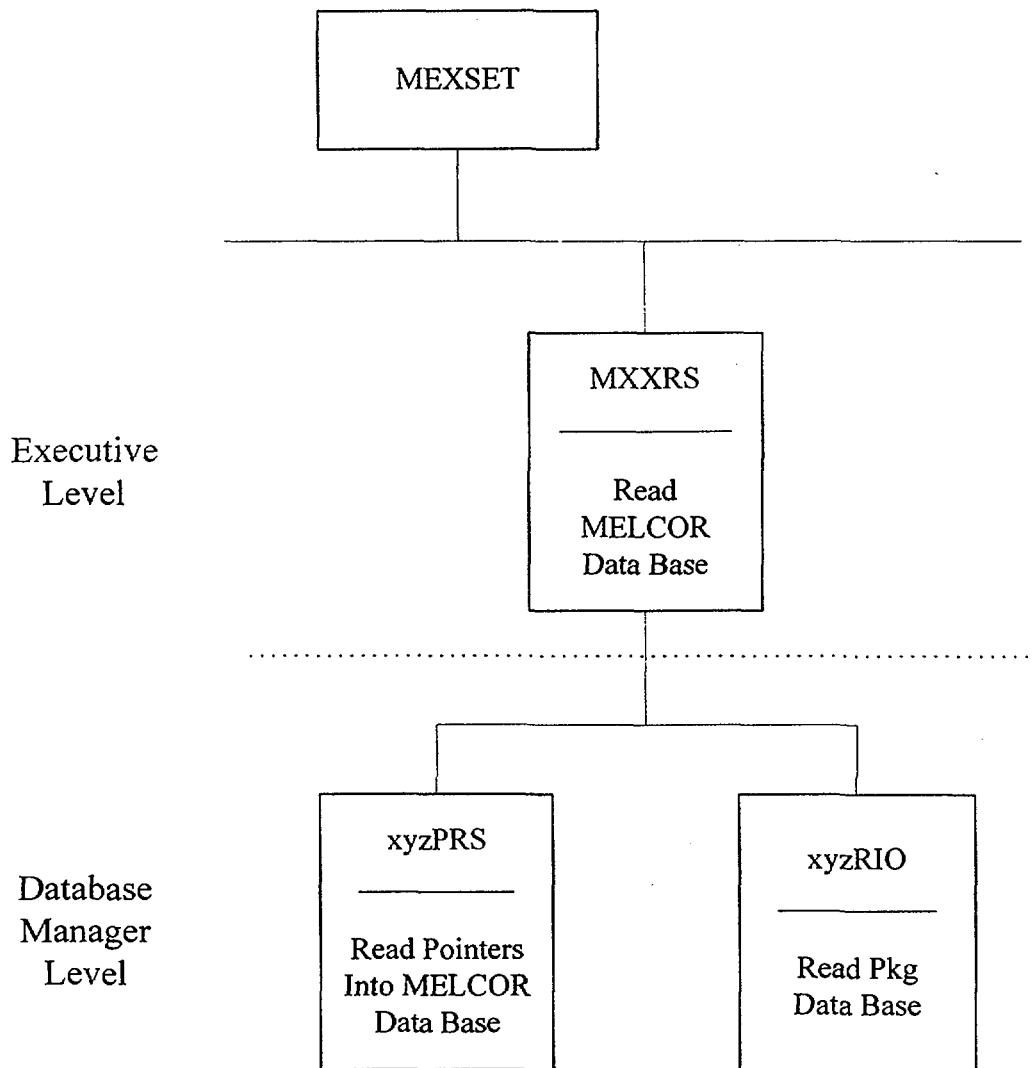


Figure 4.1.3-1 Restart File Reading in MEXSET

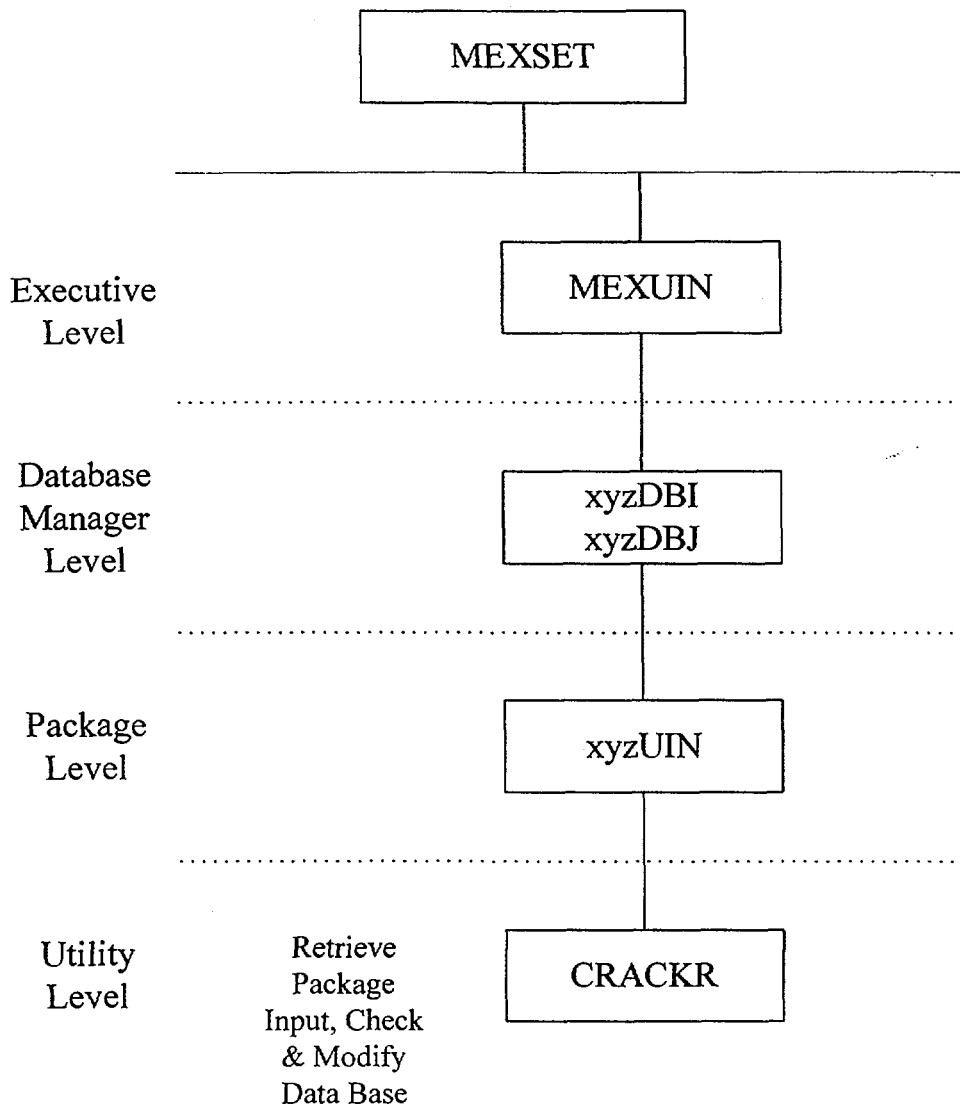


Figure 4.1.4-1 Package Input Reading in MEXSET

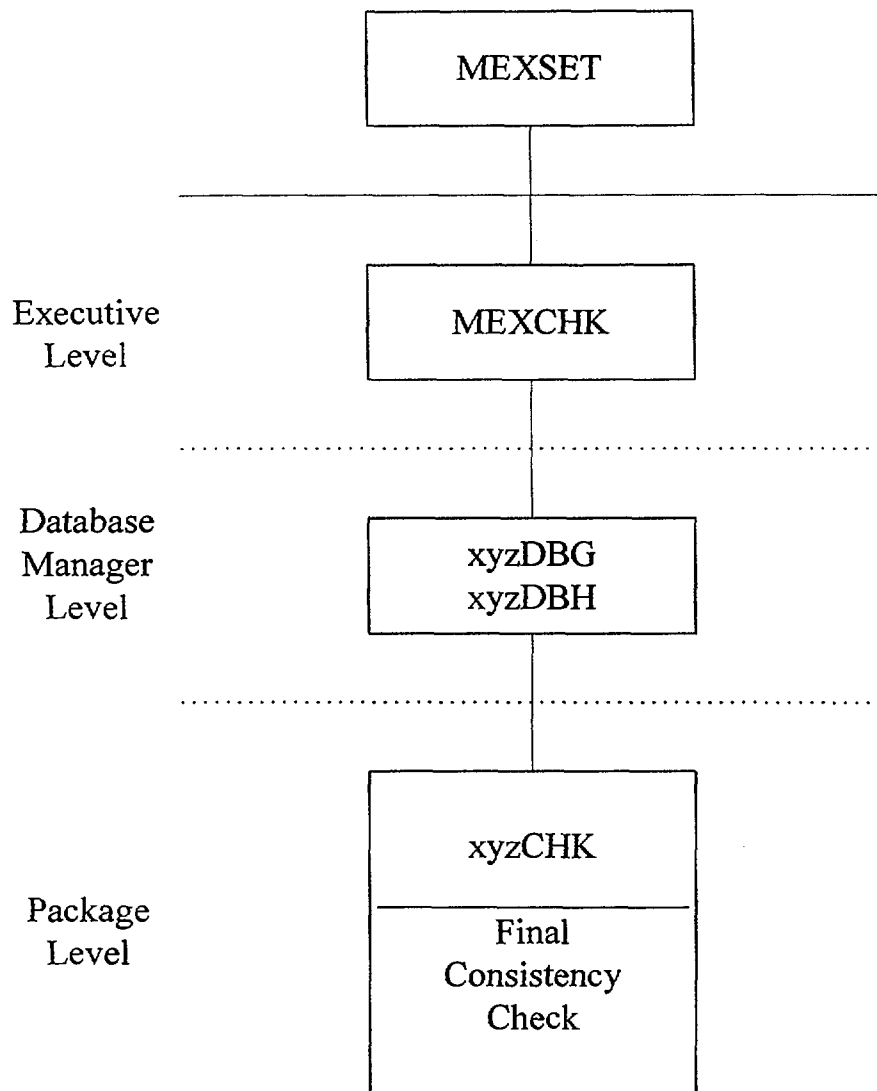


Figure 4.1.5-1 Database Check in MEXSET

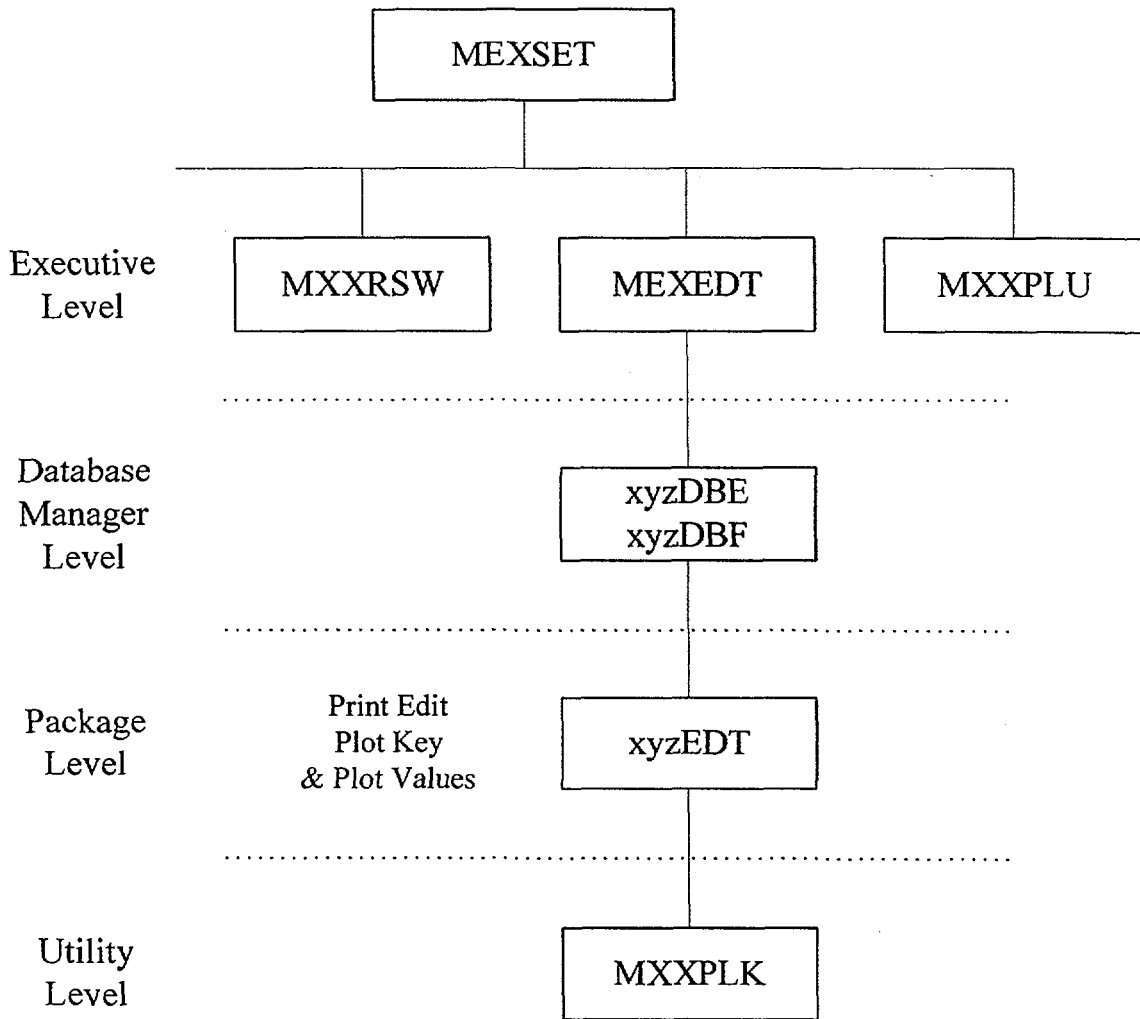


Figure 4.1.6-1 Input Echo and EDIT Generation in MEXSET

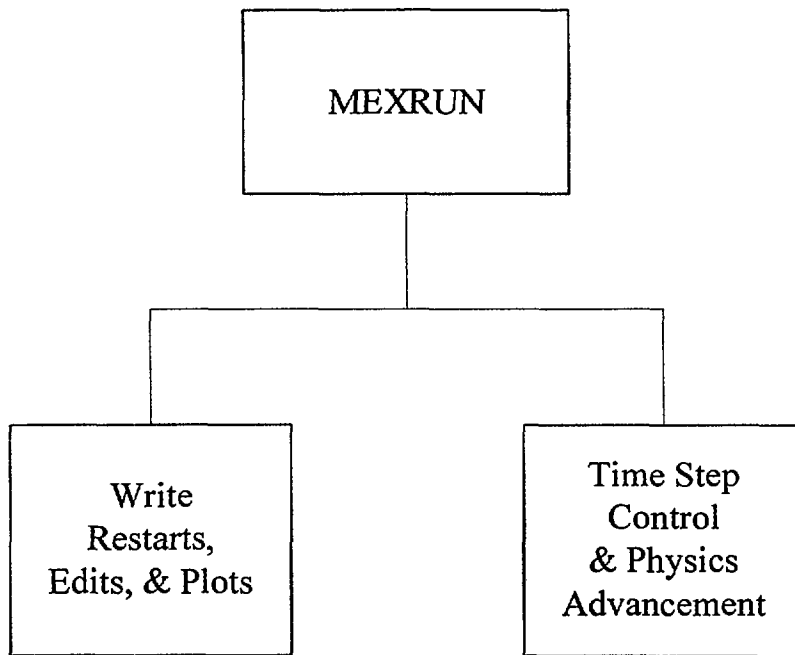


Figure 4.2-1 Functions of MEXRUN in MELCOR

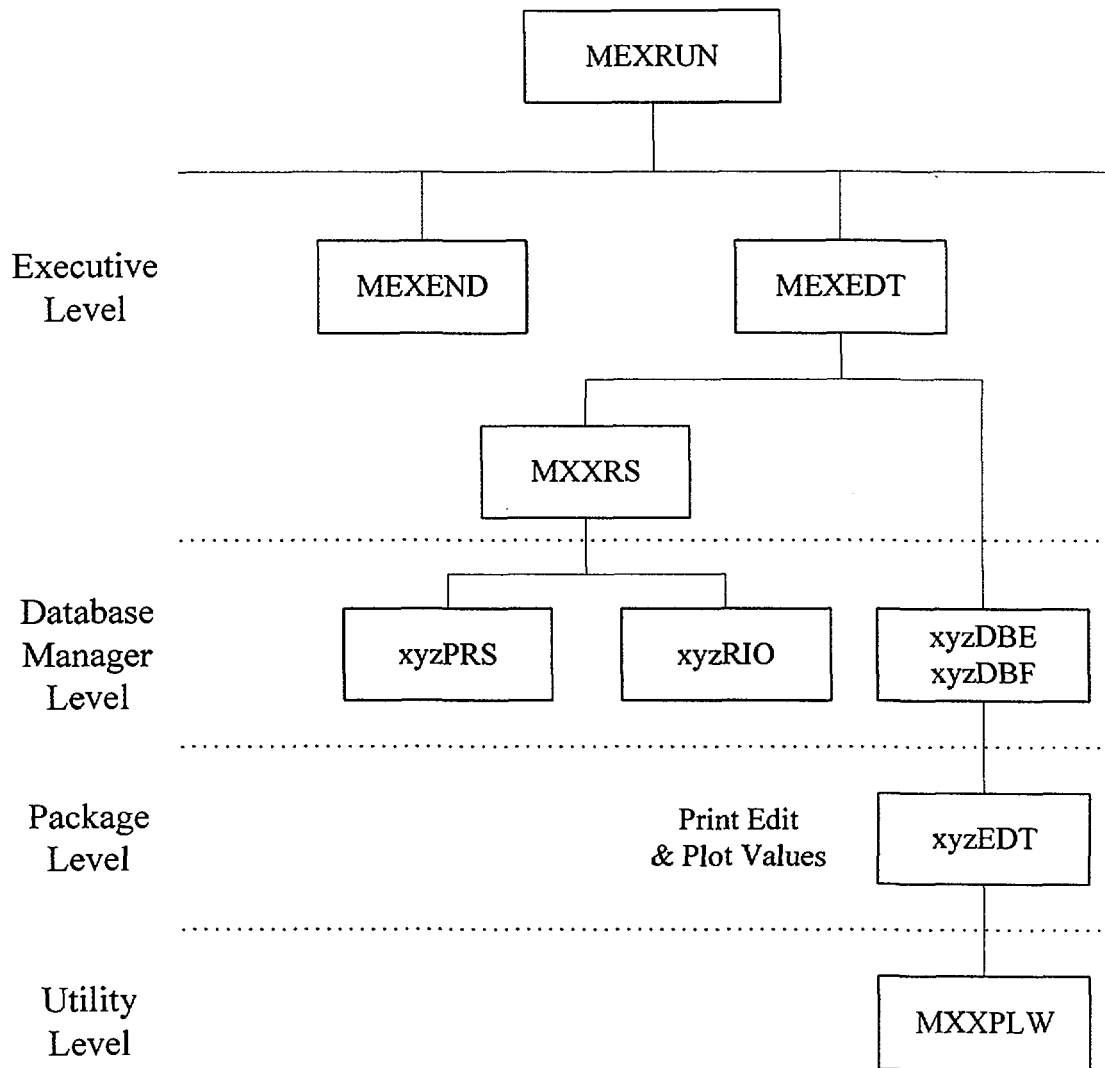


Figure 4.2.1-1 Restart, EDIT and Plots Generation in MEXRUN

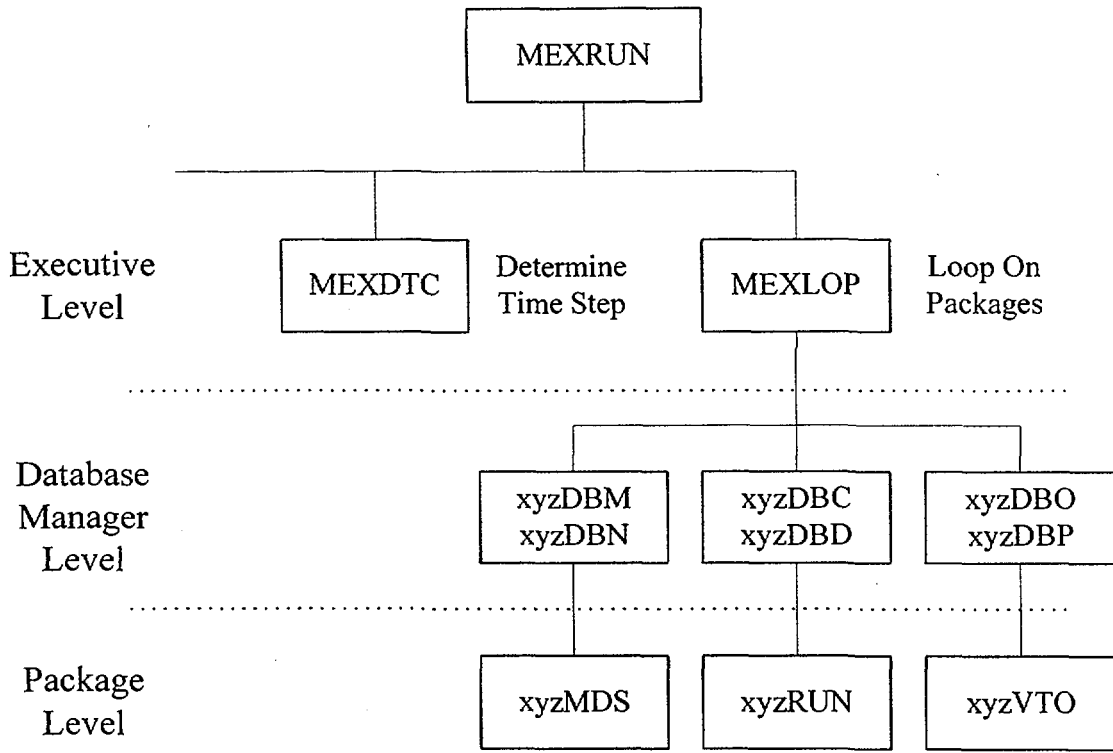


Figure 4.2.2-1 Time Step Control and Physics Advancement in MEXRUN



서 지 정 보 양 식

수행기관보고서번호	위탁연구기관보고서번호	표준보고서번호	INIS 주제코드
KAERI/TR-1543/2000			
제 목 / 부 제	MELCOR 코드 구조 분석		
주 저 자	김동하 (한국원자력연구소, 열수력안전연구팀)		
공 저 자	박선희 (한국원자력연구소, 열수력안전연구팀)		
발행지	대 전	발행 기관	한국원자력연구소
면 수	103p	도 표	유 ( O ), 무 ( )
발행일	2000년 4월		
크 기	A4		
참고사항	'99 중장기과제		
비밀여부	공개 ( O ), 대외비 ( ), ----비밀	보고서종류	기술보고서
위탁연구기관		계약번호	
초 록	<p>MELCOR 프로그램은 크게 두 부분으로 실행되는데, 대부분의 입력이 처리되고 재계산 파일에 저장되는 MELGEN 부분과, MELGEN에 의해 생성된 파일을 이용하여 실제로 시간에 따른 중대사고 현상을 모사하는 MELCOR 부분이다. 특히 MELCOR 부분은 MELGEN으로부터의 재계산 파일을 읽고 계산을 준비하는 MEXSET와 사고 진행에 동반되는 물리 현상을 모사하는 MEXRUN으로 구분된다. 또한, MELCOR 프로그램은 실행 단계, 데이터 베이스 처리 단계, 현상 해석 (현상 팩키지) 단계, 그리고 다목적 이용 프로그램 등 4개의 수직 단계로 구성된 체계적인 모듈 구조를 가지고 있어 새로운 모델을 추가하거나 교체하기가 용이한 장점을 가지고 있다. 코드에 사용되는 부 프로그램들은 MELCOR 프로그램을 구성하고 있는 24개의 팩키지에 포함되어 있고, 소속된 팩키지를 나타내기 위해 팩키지의 앞 두, 세자를 이용하여 명명하며, 관련 common block들도 동일한 방법으로 표시하고 있다. 현재 수행되고 있는 중대사고 해석 종합 전산 코드 개발에 필요한 모델 추가 및 개선에 활용하기 위해, MELGEN과 MELCOR에서의 전반적인 데이터 흐름과 관련된 세부 부프로그램들을 위의 4단계로 구분하여 MELGEN과 MELCOR의 기능별로 파악하여 정리하였다.</p>		
주제명 키워드	MELCOR, MELGEN, 중대사고, 팩키지, 모듈, 코드 구조		

BIBLIOGRAPHIC INFORMATION SHEET					
Performing Org. Report No.	Sponsoring Organization Report No.		Standard Report No.	INIS Subject Code	
KAERI/TR-1543/2000					
Title/Subtitle	Analysis of MELCOR Code Structure				
Main Author	Dong Ha Kim (KAERI, Thermal-hydraulic Safety Research)				
Author	Sunhee Park (KAERI, Thermal-hydraulic Safety Research)				
Pub. Place	Taejon	Pub. Org.	KAERI	Pub. Date	April 2000
Page	103p	Fig. & Tab.	Yes ( O ), No ( )	Size	A4
Note	'99 Mid-Long Term Project				
Classified	Open ( O ), Outside ( ), — Class		Report Type	Technical Report	
Sponsoring Org.			Contract No.		
Abstract	<p>MELCOR executes in two parts. The first is a MELGEN program, in which most of the input is specified, processed, and checked. The second part of MELCOR is the MELCOR program itself, which advances the program through time based on the database generated by MELGEN and any additional MELCOR input. In particular, MELCOR execution involves two steps: (1) a setup mode in MEXSET, during which the database is read from the restart file and any additional input is processed, and (2) a run mode in MEXRUN, which advances the simulation through time, updating the time-dependent portion of the database each cycle. MELGEN and MELCOR share a structured and modular architecture that facilitates the incorporation of additional or alternative phenomenological modes. This structure consists of four primary levels: executive level, database manager routine level, package level, and utility level. MELCOR is composed of 24 different packages, each of which models a different portion of the accident phenomenology or program control. To identify the relation of the MELCOR subroutines with the packages, first two or three letters of the package's name are duplicated in the name of the subroutines. The same rule applies to the naming of the common block. Data flows and the specific subroutines in the MELGEN and MELCOR are analyzed by their functions according to the hierarchy of four levels for model improvement and replacement during the integral code development project.</p>				
Subject Keywords	MELCOR, MELGEN, Severe Accident, Package, Module, Code Structure				