

JAERI-M
84-173

計測・制御系モデル化プログラム・ライブラリ：
LOGIC 1 の使用手引

1984年9月

渡辺 憲夫・千葉 猛美[★]・及川 哲邦・阿部 清治

JAERI-M レポートは、日本原子力研究所が不定期に公刊している研究報告書です。
入手の問合わせは、日本原子力研究所技術情報部情報資料課（〒319-11 茨城県那珂郡東海村）
あて、お申しこしてください。なお、このほかに財団法人原子力弘済会資料センター（〒319-11 茨城
県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Section, Division
of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun,
Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1984

編集兼発行 日本原子力研究所
印刷 山田軽印刷所

計測・制御系モデル化プログラム・ライブラリ：
LOGIC 1の使用手引

日本原子力研究所東海研究所安全解析部
渡辺憲夫・千葉猛美*・及川哲邦・阿部清治

(1984年8月29日受理)

原子力プラントの安全性を評価するための熱水力計算コードが数多く開発されている。それらの計算コードでは、工学的安全施設などの動的機器の作動/不作動を考慮する必要がある。本報で紹介するLOGIC 1は、動的機器の動作を模擬するプログラム・ライブラリであり、計算コードの標準化の一貫として開発したものである。本ライブラリは、計測・制御系に関する情報を記憶する機能と、記憶した情報に基づいて論理演算式を実行し、各時刻の動的機器の作動/不作動を定める機能を有している。

本ライブラリを各種熱水力計算コードに適用すれば、動的機器の制御論理だけでなく、運転員の介入や機器・格納容器の破損などを熱水力計算とは独立に模擬することができ、各種計算コードの作成を容易に行える。また、フォールト・ツリーやイベント・ツリーの作成により得られる機器故障の因果関係や事故シーケンスを直接熱水力挙動解析に反映させることが可能である。

特に、炉心溶融事故解析のように、考慮すべき動的機器の数や種類が多い場合には、本ライブラリの適用が有効であり、現在、開発中である炉心溶融事故時の熱水力挙動解析コードシステム THALES へ適用する作業が進行中である。

* 日本ソフトウェア開発 ㈱

Users' Manual of the LOGIC1 Program Library
for Modeling of Instrument and Control System

Norio WATANABE, Takemi CHIBA^{*}, Tetsukuni OIKAWA and Kiyoharu ABE

Department of Nuclear Safety Evaluation,
Tokai Research Establishment, JAERI

(Received August 29, 1984)

Accident analysis codes of nuclear power plants usually need models for instrument and control systems to simulate the actuation of active components such as Engineered Safety Features. As a standardized tool for the modeling of those instrument and control systems, a computer program library LOGIC1 has been developed.

This library has two functions. One is to memorize the information of the modeled instrument and control systems, and the other is to execute logical operation and determine success/failure of the active components at each step of the transient analyzed.

Using LOGIC1 program library, not only automatic control logics but also operator intervention and component degradation can be modeled. LOGIC1 is separate from the thermal-hydraulic analysis codes, which can provide the information on the instrument and control system function. The causal relation among component failures and accident sequences obtained from fault tree and event tree analysis respectively can be directly reflected on the transient analysis.

Such a modeling is especially important in the analysis of the core meltdown accident sequence where a lot of operations of safety related systems should be considered. LOGIC1 is being applied to the THALES code system which is under development at JAERI for the analysis of core meltdown accidents.

Keywords : LOGIC1, Instrument, Control System, THALES,
Program Library, Thermal-hydraulic Analysis
User's Manual

* NSK

目 次

1. はじめに	1
2. LOGIC 1の概要	3
2.1 LOGIC 1の機能	3
2.2 LOGIC 1の構成	6
3. LOGIC 1のサブプログラムの説明	9
3.1 LOGIC 1の初期設定	9
3.2 計測系のモデル化	9
3.2.1 設定値の登録	10
3.2.2 計測値の更新	10
3.2.3 計測値名称の登録	11
3.2.4 設定値, 計測値の引用	11
3.2.5 計測値名称及び計測値の出力	11
3.3 信号系のモデル化	12
3.3.1 信号値の登録	12
3.3.2 信号名称の登録	13
3.3.3 信号値の引用	13
3.3.4 信号名称及び信号値の出力	14
3.4 論理回路のモデル化	14
3.4.1 論理回路の登録	15
3.4.2 制御論理の変更	22
3.4.3 論理回路の出力	25
3.5 論理演算の実行	26
3.6 実行結果の出力指定	27
4. LOGIC 1の使用例	28
4.1 サンプル・プログラムの作成	28
4.1.1 主プログラムの作成	30
4.1.2 自動起動論理プログラムの作成	33
4.1.3 運転員操作による停止論理プログラムの作成	34
4.2 実行結果の出力	36
5. おわりに	40
参考文献	40

CONTENTS

1. Introduction	1
2. General Description of LOGIC1	3
2.1 Functions of LOGIC1	3
2.2 Composition of LOGIC1	6
3. Description of sub-programs in LOGIC1	9
3.1 Initialization of LOGIC1	9
3.2 Modeling of the Instrument System	9
3.2.1 Registry of Setpoint	10
3.2.2 Registry of Measurement Value	10
3.2.3 Registry of Instrumental Name	11
3.2.4 Reference of Measurement Value	11
3.2.5 Output of Registered Instrument Information	11
3.3 Modeling of the Signal System	12
3.3.1 Registry of Signal Value	12
3.3.2 Registry of Signal Name	13
3.3.3 Reference of Signal Value	13
3.3.4 Output of Registered Signal Information	14
3.4 Modeling of Logic Circuit	14
3.4.1 Registry of Logic Circuit	15
3.4.2 Update of Control Logic	22
3.4.3 Output of Registered Logical Information	25
3.5 Execution of Logical Expression	26
3.6 Output of Results	27
4. Sample Problem	28
4.1 Preparation of Sample Program	28
4.1.1 Main Program	30
4.1.2 Sub-program of Automatic Control Logic	33
4.1.3 Sub-program of Manual Operation	34
4.2 Output of Sample Program	36
5. Acknowledgement	40
References	40

1. はじめに

原子力の安全性研究の分野では、原子力プラント事故時の熱水力挙動を解析するための計算コードが数多く開発されている。近年の計算機の発達に伴い、大型の計算コードも増えているが、このような計算コード開発では、

- (1) 数多くの専門分野の知識に基づいた計算モデルの開発を必要とする、
- (2) 多大な労力を要する上、開発されたコードの修正が難しい、
- (3) 類似のプログラムを計算コードごとに重複して開発する傾向がある、

等の問題がある。このような問題を解決するためには、計算プログラムを標準化、汎用化し、それら計算プログラムの使用頻度を高めて、その信頼性や精度の向上を図ることが必要である。

リスク評価解析室では、炉心溶融事故時の熱水力挙動解析コードシステム THALES^[1]の開発にあたり、計算コードの標準化、汎用化を考慮して、その開発を進めている。Fig. 1.1 に THALES コードシステムの構成を示す。図示するように、これまでに既に、単位換算プログ

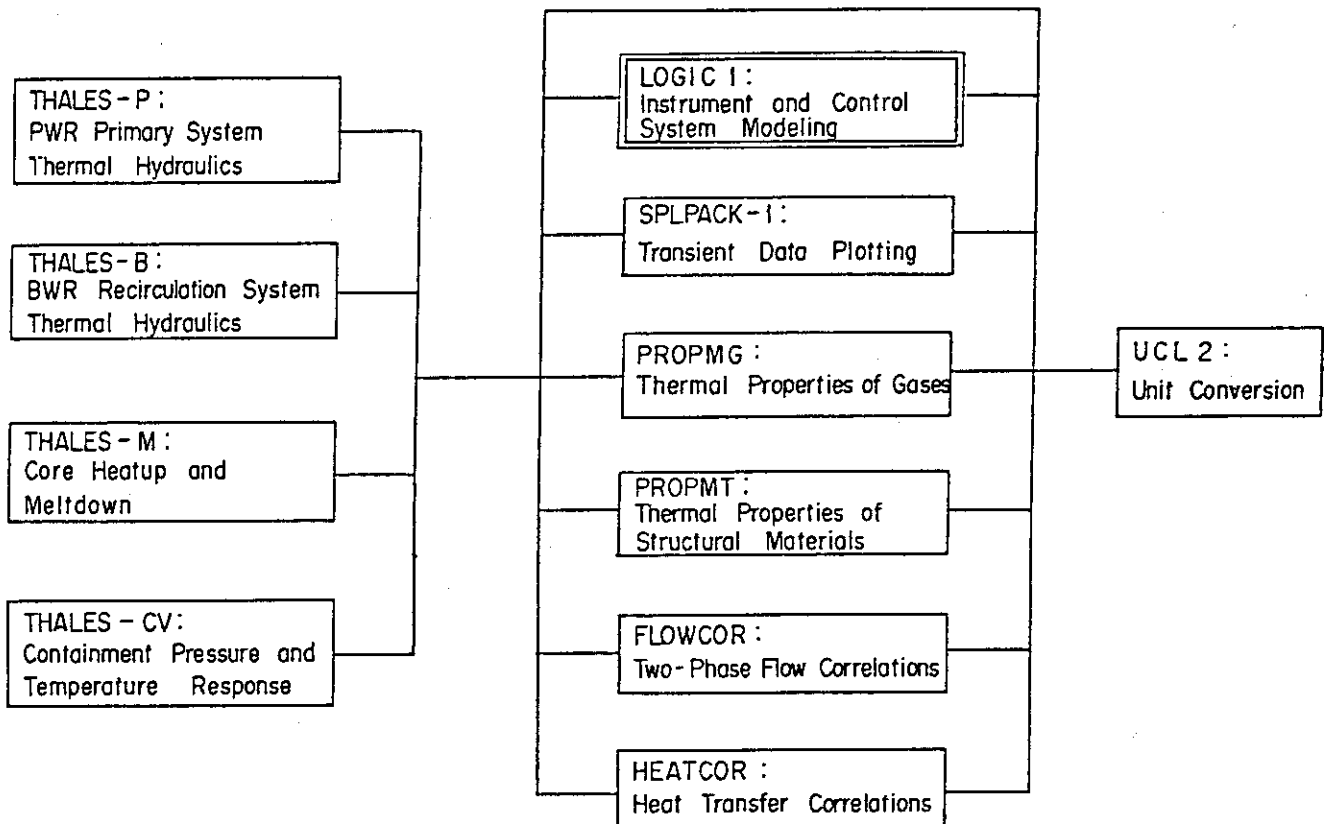


Fig. 1.1 THALES コードシステムの構成

ラム・ライブラリUCL2^[2]やプロッター・プログラム・ライブラリSPLPACK-1^[3]等、汎用プログラム・ライブラリを開発、使用している。

本報告書で紹介するLOGIC1も、このような標準化、汎用化を目指して開発したプログラム・ライブラリである。本ライブラリは、計測・制御系の機能を首尾一貫した方法でモデル化することができ、非常用炉心冷却系などの各種動的機器の作動論理を正確に表現できる。特に、炉心溶融事故解析のように、考慮すべき動的機器の種類や数が多い場合には、それら全てを熱水力計算コード内部で扱うことにより、プログラムが煩雑になり、その品質保証や修正が困難となる。LOGIC1を適用すれば、各種動的機器の自動作動だけでなく、運転員の介入による作動・停止や誤作動・誤停止を、熱水力計算コードとは独立した方法で表現することができ、熱水力計算コードの縮小が可能となる。また、LOGIC1は、熱水力計算コードと独立のプログラムなので、フォールト・ツリーやイベント・ツリーの作成により判明する機器故障の因果関係や事故シーケンスの推移をそのまま熱水力挙動解析に反映させることができよう。

本報告書では、LOGIC1の計算モデルを説明すると共に、その使用例を紹介する。

2. LOGIC 1 の概要

LOGIC 1 は、計測・制御系をモデル化するプログラム・ライブラリである。本ライブラリの目的は、原子炉の事故解析、トランジェント解析で考慮する各種動的機器、例えば工学的安全施設や加圧器逃し弁などの作動・不作動を、熱水力挙動解析コードと独立に、汎用プログラムの形で模擬することである。

本章では、LOGIC 1 の主要機能と基本的な構成について記述する。

2.1 LOGIC 1 の機能

LOGIC 1 の主要機能は、計測・制御系に関する情報を論理式の形で記憶する（本ライブラリでは、これを情報の「登録」と呼ぶ）機能と、登録された情報を用いて論理演算を実行し、各種信号の状態（即ち、信号の発生／消滅）を定める機能とに大別される。LOGIC 1 では、計測・制御系を、計測系、信号系及び論理回路に分類して扱う。LOGIC 1 の機能は、次に示す 4 種に分けられる。

- (1) 計測系情報の登録
- (2) 信号系情報の登録
- (3) 論理回路情報の登録
- (4) 論理演算の実行

ここで、(1)は、運転員による各種動的機器作動設定点の設定、トランジェント各時刻における計測値の測定、及びそれを読みとる計測機器の識別名称を登録するものであり、LOGIC 1 プログラム内部に、計測値名称-計測値あるいは、設定値名称-設定値の組で登録する。ここで、計測値は、熱水力計算コードによる各種パラメータの計算値である。(2)は、各種動的機器の作動信号の初期状態を設定するものであり、信号名称-信号値の組で登録する。信号値は、信号の発生あるいは消滅のいずれかの状態を表わすものである。(3)は、計測値や信号値を入力として新たな信号を発生させる制御系の論理回路をモデル化するものであり、そのモデルを論理式の形で登録する。(4)は、トランジェント各時刻において、(1)で与えられた設定値や計測値、及び、(2)で初期設定された信号値を基に、(3)で登録された論理回路に従って論理演算を実行するものであり、その時刻での各信号の状態（即ち、信号値）を更新する。このようにして定められた信号値により、各種動的機器の作動／不作動を制御できる。

なお、このライブラリは、動的機器の自動作動だけではなく、ある条件が満足された時の運転員の介入による機器作動や停止、動的機器の誤作動や誤停止、機器の破損や機器間の相互依存性も一種の制御系と考えて模擬することができる。

LOGIC 1 の使用方法を説明するために、Fig. 2. 1 に計算手順の概要を示す。図中④、⑤は、熱水力計算の部分である。LOGIC 1 では、①～③で計測系、信号系、論理回路を登録し、熱水力挙動解析コードにより計算された値を計測値として⑥で登録する。これらの情報を基に⑦

で論理演算を実行し、各信号の状態を更新する。⑧では、更新された各信号の状態を引用し、熱水力挙動解析コードにおけるトランジェント計算に必要な動的機器作動・不作動が定められる。

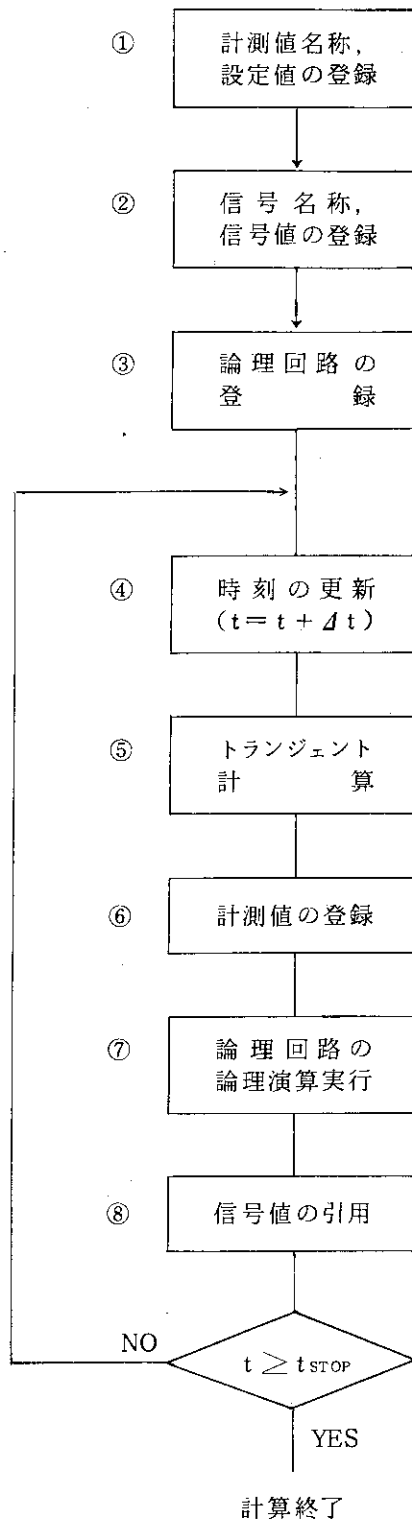


Fig. 2.1 LOGIC 1 の計算手順

2.2 LOGIC 1の構成

LOGIC 1は内部共通領域の初期設定に必要なサブプログラムの他、Fig. 2.2に示すように、計測系、信号系、論理回路に関する情報をLOGIC 1内部共通領域に登録する3種のサブプログラム群（登録内容を引用、出力するサブプログラムを含む）と、登録内容を基に論理演算を実行するサブプログラムから構成される。各サブプログラムの名称と各々の機能概略をTable 2.1に一覧する。また、各サブプログラムの使用方法は次章で説明する。

登録情報は、以下の3つの共通領域に記憶される。

- (1) 計測系モデルの内容…………… INSCOM
- (2) 信号系モデルの内容…………… SIGCOM
- (3) 論理回路モデルの内容…………… LOGCOM

LOGIC 1の最終結果として、各種熱水力挙動解析コードで考慮する動的機器の作動／不作物を制御するのに必要な情報は、上記(2)の共通領域SIGCOMに記憶された内容から引用する。

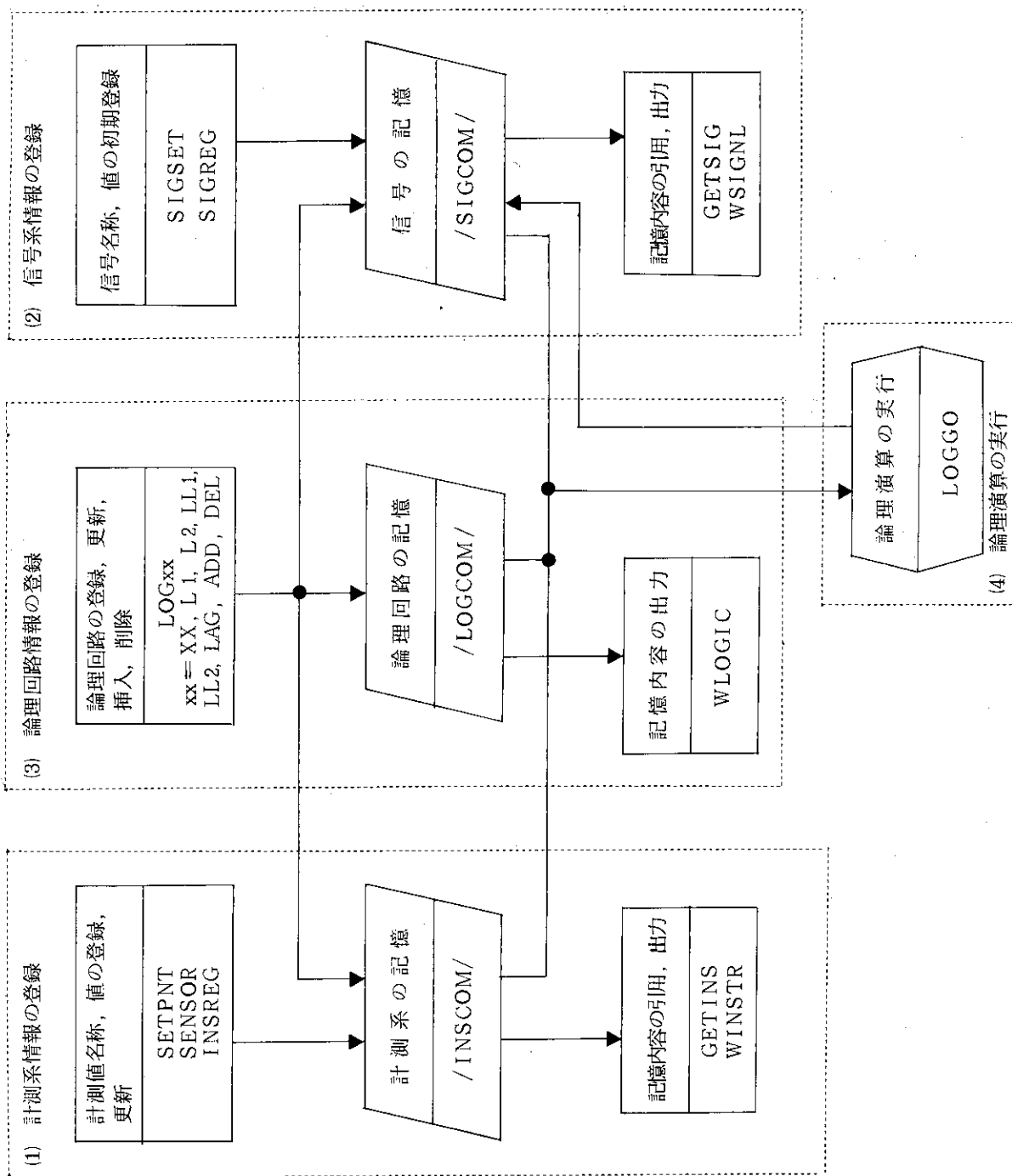


Fig. 2.2 LOGIC 1 の基本構成

Table 2.1 サブプログラム一覧

用 途	サブプログラム名	機 能 概 略
初 期 設 定	LOGINI	LOGIC 1 の初期設定
計測系のモデル化	SETPNT SENSOR INSREG GETINS WINSTR	設定値の登録・更新 計測値の登録・更新 計測値名称の登録 計測値, 設定値の引用 計測値名称及び計測値 (設定値) の出力
信号系のモデル化	SIGSET SIGREG GETINS WSIGNL	信号値の初期登録 信号名称の登録 信号値の引用 信号名称及び信号値の出力
論 理 回 路 の モ デ ル 化	LOGXX LOGL 1 LOGL 2 LOGLL 1 LOGLL 2 LOGLAG LOGADD LOGDEL LOGMOV WLOGIC	<p>論理回路の形により使用する サブプログラムが異なる。 (詳細は 3 章参照)</p> <p>時間遅れ論理回路の登録・更新 論理回路の挿入 論理回路の削除 論理回路の移動 論理回路の登録内容の出力</p>
実 行	LOGGO	論理演算の実行
出 力	SIGVRF	実行結果の出力指定

3. LOGIC 1 のサブプログラムの説明

本章では、LOGIC 1を構成するサブプログラムの使用方法を各サブプログラムごとに、参照形式、引数、機能、使用例に分けて説明する。

ここで、引数の項目で括弧内の記述を予め定義する。括弧内の前者は、当該サブプログラムに対する入出力の別（入：入力、出：出力）を表わし、後者は、本サブプログラムで必要とする変数の形式を示す。変数の形式とその意味は、以下に示す通りである。

変数の形式	意味
文……………	文字型（4文字列）
実……………	実数型（4バイトの実数）
整……………	整数型（4バイトの整数）
論……………	論理型（.TRUE.あるいは.FALSE.）

3.1 LOGIC 1の初期設定（LOGINI）

LOGIC 1を使用するにあたっては、共通領域の初期設定や出力機番の指定が必要である。本節で説明するサブプログラムLOGINIを呼び出すことにより、LOGIC 1の初期設定を行うことができる。

(a) 参照形式

```
CALL LOGINI (NOUT, NERR)
```

(b) 引数

NOUT（入、整）……………LOGIC 1の出力機番

NERR（入、整）……………LOGIC 1のエラーメッセージの出力機番

(c) 機能

LOGIC 1の計算結果の出力機番及びエラーメッセージの出力機番を決定し、LOGIC 1の有する各共通領域の初期設定を行う。本サブプログラムは、LOGIC 1を使用する際、最初に呼びださなければならない。

(d) 使用例

```
CALL LOGINI (6, 9)
```

LOGIC 1の計算結果の出力を6番に、エラーメッセージの出力を9番と指定し、LOGIC 1内部の共通領域の初期設定を行う。

3.2 計測系のモデル化

計測系をモデル化し、登録するサブプログラムとして、SETPNT, SENSOR, INSREGがある。これらのサブプログラムは、設定値の名称及びその値、あるいは計測値の名称及びその

値を LOGIC 1 内部の共通領域 INSCOM に登録する。登録できる最大数は、SETPNT, SENSOR, INSREG 合わせて 100 個である。また、登録内容の引用、出力は、それぞれ、サブプログラム GETINS, WINSTR を用いて行う。

3.2.1 設定値の登録 (SETPNT)

(a) 参照形式

```
CALL SETPNT (NAMINS, VALINS, MINS)
```

(b) 引数

NAMINS (入, 文) ……設定値の名称

VALINS (入, 実) ……設定値

MINS (出, 整) ……登録番号 (最大数 100)

(c) 機能

機器の作動・不作動 (作動信号の発生や消滅) を規定するための設定点を、設定値名称 - 設定値の組で登録する。また、既に設定値名称が登録されている場合には、設定値のみを更新する。

(d) 使用例

安全弁が開くための設定圧 PSVO として、1200 (psi) なる値を登録する。

```
PSVO = 1200.0 * UCONB (▽PSI▽, 3, IDQ) ……単位換算
```

```
CALL SETPNT (▽PSVO▽, PSVO, M)
```

この例では、単位換算プログラム・ライブラリ UCL 2 を用いて、設定値の登録前に単位の変換をした後、安全弁開の設定圧に ∇ PSVO ∇ という名称を付けて、その値 PSVO と共に登録している。

3.2.2 計測値の更新 (SENSOR)

(a) 参照形式

```
CALL SENSOR (NAMINS, VALINS, MINS)
```

(b) 引数

NAMINS (入, 文) ……計測値の名称

VALINS (入, 実) ……計測値

MINS (出, 整) ……登録番号 (最大数 100)

(c) 機能

計測値を更新する。計測値名称が未登録の場合には、計測値名称 - 計測値の組で登録する。通常は、計測値として熱水力挙動解析コードにより得られる計算値を登録、更新する。

(d) 使用例

熱水力計算コードによって計算される一次系圧力 PRPV を計測値名称 ∇ PRPV ∇ と共に、計測値として登録する。

```
CALL SENSOR (▽PRPV▽, PRPV, M)
```

一次系圧力 PRPV は、トランジェント計算の各時刻において更新される。

3.2.3 計測値名称の登録 (INSREG)

(a) 参照形式

CALL INSREG (NAMINS, MINS)

(b) 引数

NAMINS (入, 文) ……計測値の名称

MINS (出, 整) ……登録番号 (最大数 100)

(c) 機能

計測値の名称のみを登録する。既に同一の計測値名称が登録されている場合には、その登録番号が返される。本サブプログラムは計測値が未登録の段階で、論理回路の登録を行う際、計測値名称だけ登録するためのものである。本サブプログラムで計測値名称を登録した場合、計測値にはデフォルト値 (-7.23701×10^{75}) が設定される。

(d) 使用例

一次系圧力を表わす計測値の名称を ∇ PRPV ∇ として登録する。

CALL INSREG (∇ PRPV ∇ , M)

こうして登録した計測値名称 ∇ PRPV ∇ は後述の論理回路登録に用いられる。

3.2.4 設定値, 計測値の引用 (GETINS)

(a) 参照形式

CALL GETINS (NAMINS, VALINS, MINS)

(b) 引数

NAMINS (入, 文) ……設定値名称または計測値名称

VALINS (出, 実) ……設定値または計測値

MINS (出, 整) ……登録番号

(c) 機能

指定された設定値名称 (または計測値名称) に対応する設定値 (または計測値) を、登録情報の中から検索する。指定された設定値名称 (または計測値名称) が未登録の場合には、エラーとなり登録番号として 0 を設定する。

(d) 使用例

計測系に関して、登録されている情報の中から、計測値名称が ∇ PRPV ∇ である一次系圧力 PRPV (通常、熱水力計算コードによる計算値) を引き出す。

CALL GETINS (∇ PRPV ∇ , PRPV, M)

3.2.5 計測値名称及び計測値の出力 (WINSTR)

(a) 参照形式

CALL WINSTR (NPAGE)

(b) 引数

NPAGE (入, 整) ……改ページ指示子

(c) 機能

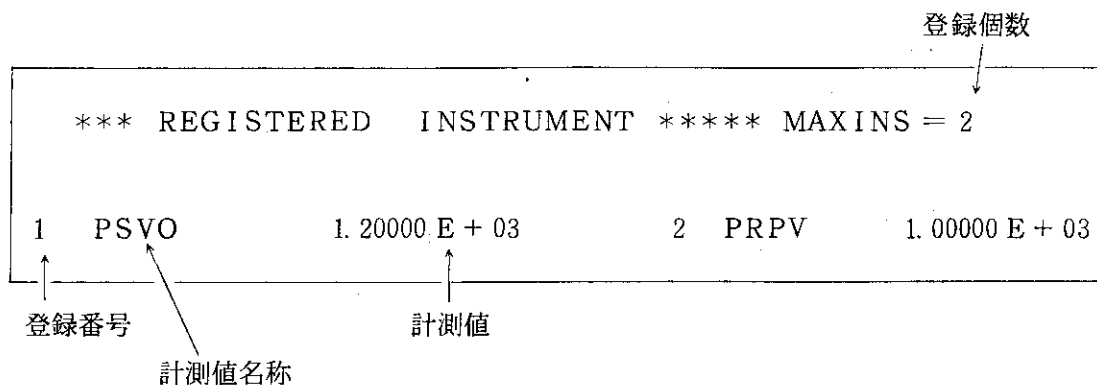
内部共通領域に登録されている計測系の内容を計測値名称-計測値の組で出力する。引数 NPAGE が、" 1 " の場合に限り、改ページが行われ、" 1 " 以外の場合には改ページされない。

(d) 使用例

登録されている計測値の名称及びその値を出力する。

CALL WINSTR (1)

この時の出力は、以下のような形式となる。



3.3 信号系のモデル化

信号系をモデル化し、登録するサブプログラムとして、SIGSET、SIGREG がある。これらのサブプログラムでは、信号の名称及びその値を LOGIC 1 内部の共通領域 SIGCOM に登録する。登録できる最大数は、SIGSET、SIGREG 合わせて 100 個である。また、登録内容の引用にはサブプログラム GETSIG、出力にはサブプログラム WSIGNL を用いて行う。

3.3.1 信号値の登録 (SIGSET)

(a) 参照形式

CALL SIGSET (NAMSIG, LLSIG, MSIG)

(b) 引数

NAMSIG (入, 文)信号名称

LLSIG (入, 論あるいは整)信号値

MSIG (出, 整)登録番号 (最大数 100)

(c) 機能

信号の状態を信号名称-信号値の組で登録する。既に同一の信号名称が登録されている場合には、信号値を更新する。信号値は、信号の発生/消滅を表わすもので、 ∇ . FALSE. ∇ (または ∇ 0 ∇) あるいは、 ∇ . TRUE. ∇ (または ∇ 1 ∇) で表現する。

(d) 使用例

安全弁を開けるための信号の信号値を次のように設定する。

「安全弁開信号 ∇ SVSO ∇ が発生していない (.FALSE.)」

これは、

```
CALL SIGSET ( $\nabla$ SVSO  $\nabla$ , .FALSE., M)
```

あるいは

```
CALL SIGSET ( $\nabla$ SVSO  $\nabla$ , 0, M)
```

で表現される。

3.3.2 信号名称の登録 (SIGREG)

(a) 参照形式

```
CALL SIGREG (NAMSIG, MSIG)
```

(b) 引数

NAMSIG (入, 文) ……信号名称

MSIG (出, 整) ……登録番号 (最大数 100)

(c) 機能

信号名称のみを登録する。既に同一の信号名称が登録されている場合には、その登録番号を設定する。本サブプログラムは、信号値の初期設定が行われていない段階で、論理回路の登録を行う場合に、信号名称だけ登録するためのものである。本サブプログラムで信号名称を登録した場合、信号値にはデフォルト値として ∇ .FALSE. ∇ が設定される。

(d) 使用例

安全弁を開けるための信号の名称を ∇ SVSO ∇ として登録する。

```
CALL SIGREG ( $\nabla$ SVSO  $\nabla$ , M)
```

3.3.3 信号値の引用 (GETSIG)

(a) 参照形式

```
CALL GETSIG (NAMSIG, LLSIG, MSIG)
```

(b) 引数

NAMSIG (入, 文) ……信号名称

LLSIG (出, 論あるいは整) ……信号値

MSIG (出, 整) ……登録番号 (最大数 100)

(c) 機能

指定された信号名称に対応する信号値を、登録されている情報の中から検索する。指定した信号名称が未登録の場合は、エラーとなり登録番号として 0 を設定する。

(d) 使用例

信号系に関して登録されている情報の中から、安全弁を開けるための信号 ∇ SVSO ∇ に対応する信号値 SVSO を引き出す。

```
CALL GETSIG ( $\nabla$ SVSO  $\nabla$ , SVSO, M)
```

3.3.4 信号名称及び信号値の出力 (WSIGNL)

(a) 参照形式

CALL WSIGNL (NPAGE)

(b) 引数

NPAGE (入, 整) ……改ページ指示子

(c) 機能

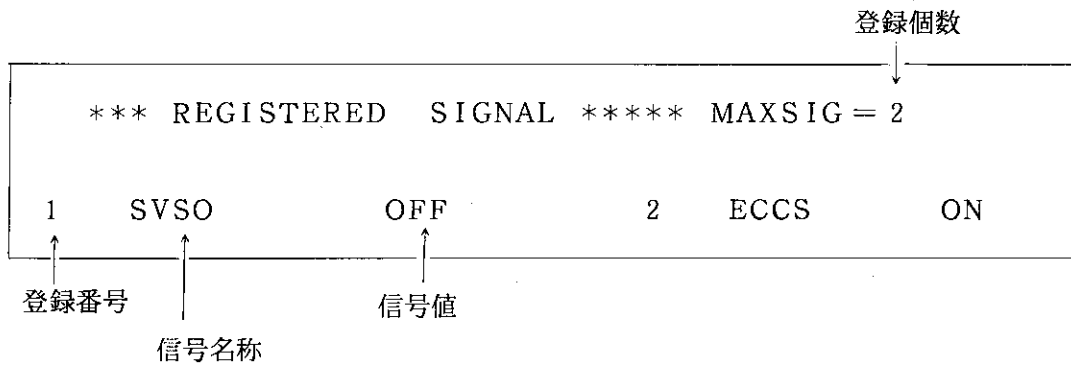
内部共通領域に登録されている信号系の内容を信号名称-信号値の組で出力する。引数 NPAGE が " 1 " の場合に限り, 改ページが行われ, " 1 " 以外の場合には改ページされない。

(d) 使用例

登録されている信号名称及び信号値を出力する。

CALL WSIGNL (0)

この時の出力は, 以下に示すような形式になる。



ここで, ∇ ON ∇ は ∇ .TRUE. ∇ , ∇ OFF ∇ は ∇ .FALSE. ∇ を表わす。

3.4 論理回路のモデル化

動的機器の作動信号を発生あるいは消滅させるための制御論理は, 一般には幾つかの論理回路の組合せから成る。その基本となる論理回路は, " 1 out of 2 " 回路や " 2 out of 2 " 回路などに代表される論理和, 論理積の回路である。これら論理回路の他に, 工学的安全施設作動設備に信号 (例えば, 原子炉圧力低信号など) を供給する安全保護系のプロセス計装のように, 2 値間の大小関係により出力信号を規定するものがあるが, LOGIC 1 ではこれも一種の論理回路と考える。LOGIC 1 は, 各々の論理回路に適合する数種のサブプログラムを用意している。それらのサブプログラムを用いて, 一連の制御論理を論理式の形でモデル化し, 内部共通領域, LOGCOM に登録する。モデル化された制御論理は, 論理回路の登録順序によって一意に定まる。制御論理の変更は, 論理回路の挿入, 削除, 変更及び移動の 4 種の方法で行うことができる。本節では, 論理回路を登録するサブプログラムと, 制御論理の変更に用いるサブプログラムについて説明する。

3.4.1 論理回路の登録

前述したように、一連の制御論理を構成するものには、論理和回路や論理積回路の他、2値間の大小判定を行う比較回路などがあるが、LOGIC 1では、これらを総称して「論理回路」と呼ぶ。本ライブラリには使用性を踏えて各々の論理回路に適合したサブプログラムを用意しているため、それらのサブプログラムを用いて容易に制御論理をモデル化することができる。Table 3.1に、本ライブラリで考慮する5種の論理回路について、その型式、具体例及びそれに対応するサブプログラムを一覧する。この他に、時間遅れを取り扱えるよう、タイマー付論理回路をモデル化するサブプログラム LOGLAG がある。

(1) 型式(1)の論理回路の登録 (LOGXX)

(a) 参照形式

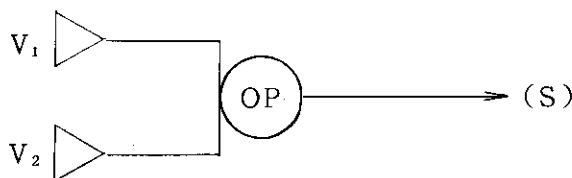
CALL LOGXX (NAMLOG, NINS 1, NOPER, NINS 2, NSIG, LSIG)

(b) 引数

- NAMLOG (入, 文) 論理回路名称
- NINS 1 (入, 文) 計測値名称-1
- NOPER (入, 文) 算術演算子
- NINS 2 (入, 文) 計測値名称-2
- NSIG (入, 文) 信号名称
- LSIG (入, 論あるいは整) 信号値

(c) 機能

下図に示すような論理回路の登録を行う。この論理回路は、LOGIC 1では型式(1)と呼称するものであり、設定値-計測値間、あるいは計測値-計測値間の大小関係によって信号状態を定める論理回路を表わすものである。



ここで、 V_1 , V_2 は設定値もしくは計測値、OP は大小判定をする算術演算子、S は出力信号を表わす。算術演算子として使用できるものは以下の通りである。

GT (>), GE (\geq), EQ (=), NE (\neq), LE (\leq), LT (<)

(d) 使用例

一次系圧力 $P_{(RPV)}$ が安全弁開設定圧 $P_{(SVO)}$ を超えると、安全弁開信号 $S_{(SVO)}$ が発生する。これで論理式で書くと

IF ($P_{(RPV)}.GT. P_{(SVO)}$) $S_{(SVO)} = .TRUE.$

となり、これは、論理回路の型式(1)に相当する。これをLOGIC 1では、次のように表現して登録する。

Table 3.1 論理回路の形式と専用サブプログラム

論理回路の型式	論理式の具体例	サブプログラム
<p>(1) IF (V₁.OP.V₂) S₁=L</p> <p>V₁, V₂: 計測値あるいは設定値 OP: 大小判定をする算術演算子 (GT, GE, EQ, NE, LE, LT) S₁: 出力信号 L: 信号状態</p> <p>2つの入力値の大小関係より出力信号の状態を定める。</p>	<p>IF (V₁.GT.V₂) S₁=.TRUE. ・V₁>V₂ならば, S₁が発生する。</p> <p>IF (V₁.LE.V₂) S₁=.FALSE. ・V₁≤V₂ならば, S₁が消滅する。</p> <p>IF (V₁.NE.V₂) S₁=.FALSE. ・V₁≠V₂ならば, S₁が消滅する。</p>	LOGXX
<p>(2) IF (S₁) S₂=L</p> <p>S₁, S₂: 信号 L: 信号状態</p> <p>ある信号が発生している時, 他の信号の状態を定める。</p>	<p>IF (S₁) S₂=.TRUE. ・S₁が発生すれば, S₂が発生する。</p> <p>IF (S₁) S₂=.FALSE. ・S₁が発生すれば, S₂が消滅する。</p>	LOGL 1
<p>(3) IF (S₁.EQ.L₁) S₂=L₂</p> <p>S₁, S₂: 信号 L₁, L₂: 信号状態</p> <p>ある信号が指定した状態である時に, 他の信号の状態を定める。</p>	<p>IF (S₁.EQ.FALSE.) } S₂=.TRUE. IF (.NOT.S₁) } ・S₁が消滅すれば, S₂が発生する</p> <p>IF (S₁.EQ.TRUE.) S₂=.FALSE. ・S₁が発生すれば, S₂が消滅する。</p>	LOGL 2
<p>(4) IF (S₁.OP.S₂) S₃=L₃</p> <p>S₁, S₂: 入力信号, S₃: 出力信号 L₃: 信号状態 OP: 演算子 (AND, OR, EQ, NE)</p> <p>2つの入力信号の組合せにより, 出力信号の状態を定める。</p>	<p>IF (S₁.OR.S₂) S₃=.TRUE. ・S₁, S₂の一方が発生すれば, S₃が発生する。</p> <p>IF (S₁.AND.NOT.S₂) S₃=.FALSE. ・S₁が発生し, S₂が消滅すれば, S₃が消滅する。</p> <p>IF (S₁.EQ.S₂) S₃=.TRUE. ・S₁, S₂が共に発生, あるいは共に消滅すれば, S₃が発生する。</p>	LOGLL 1
<p>(5) IF (S₁.EQ.L₁.OP.S₂.EQ.L₂) S₃=L₃</p> <p>S₁, S₂: 入力信号 S₃: 出力信号 L₁, L₂, L₃: 信号状態 OP: 論理演算子 (AND, OR)</p> <p>2つの入力信号の状態を論理結合することにより, 出力信号の状態を定める。</p>	<p>IF (S₁.EQ.TRUE.AND.S₂.EQ.FALSE.) S₃=.TRUE. ・S₁が発生し, S₂が消滅すれば, S₃が発生する。これは以下のようにも表現できる。</p> <p>IF (S₁.AND.NOT.S₂) S₃=.TRUE.</p>	LOGLL 2

CALL LOGXX (▽SVOL▽, ▽PRPV▽, ▽GT▽, ▽PSVO▽, ▽SSVO▽, .TRUE.)

ここで、▽SVOL▽はこの論理回路の名称、▽PRPV▽, ▽PSVO▽, ▽SSVO▽は、それぞれ計測値 $P_{(RPV)}$, 設定値 $P_{(SVO)}$, 信号 $S_{(SVO)}$ の名称である。

(2) 型式(2)の論理回路の登録 (LOGL1)

(a) 参照形式

CALL LOGL1 (NAMLOG, NSIG1, NSIG2, LSIG)

(b) 引数

NAMLOG (入, 文) ……論理回路名称

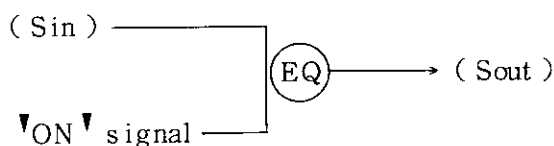
NSIG1 (入, 文) ……信号名称-1 (入力信号)

NSIG2 (入, 文) ……信号名称-2 (出力信号)

LSIG (入, 論あるいは整) ……信号値

(c) 機能

下図に示すような論理回路の登録を行う。この論理回路は LOGIC1 では、型式(2)と呼称するものであり、ある入力信号が発生した時のみ、出力信号の状態を定める論理回路を表わすものである。



ここで、Sin は入力信号、Sout は出力信号、EQ は等号を表わす。

(d) 使用例

ECCS 作動信号 $S_{(ECC)}$ が発生すると、格納容器隔離信号 $S_{(ISL)}$ が発生すると想定した場合、これを論理式で書くと、

IF ($S_{(ECC)}$) $S_{(ISL)} = .TRUE.$

あるいは

IF ($S_{(ECC)} .EQ. .TRUE.$) $S_{(ISL)} = .TRUE.$

となり、これは型式(2)の論理回路に相当する。これを LOGIC1 では次のように表現して登録する。

CALL LOGL1 (▽ISOL▽, ▽SECC▽, ▽SISL▽, .TRUE.)

ここで、▽ISOL▽はこの論理回路の名称、▽SECC▽, ▽SISL▽はそれぞれ信号 $S_{(ECC)}$, $S_{(ISL)}$ の名称である。

また、本サブプログラムは、次のような場合にも使用することができる。

「安全弁が開くと、安全弁からの蒸気流出が始まる」

この場合、「安全弁の開閉状態を表わす信号 $S_{(SVL)}$ が発生する (安全弁が開く) と、安全弁からの蒸気流出有無を表わす信号 $S_{(WSV)}$ が発生する (蒸気流出が始まる)」と考えることができ、これを論理式で書くと

IF (S_(svL)) S_(wsv) = .TRUE.

となり、型式(2)の論理回路に相当する。

(3) 型式(3)の論理回路の登録 (LOGL 2)

(a) 参照形式

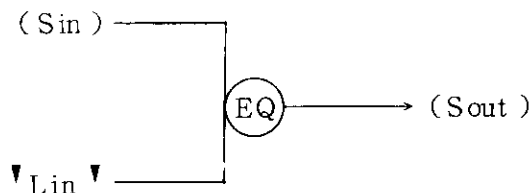
CALL LOGL 2 (NAMLOG, NSIG1, LSIG1, NSIG2, LSIG2)

(b) 引 数

- NAMLOG (入, 文) ……論理回路名称
- NSIG 1 (入, 文) ……信号名称-1 (入力信号)
- LSIG 1 (入, 論あるいは整) ……信号値-1
- NSIG 2 (入, 文) ……信号名称-2 (出力信号)
- LSIG 2 (入, 論あるいは整) ……信号値-2

(c) 機 能

下図に示すような論理回路の登録を行う。この論理回路は、LOGIC 1では型式(3)と呼称するものであり、入力信号が予め指定した状態になった場合に、出力信号の状態を定める論理回路を表わすものである。



ここで、Sin は入力信号、Sout は出力信号、Lin は指定した信号状態、EQ は等号を表わす。

(d) 使用例

タービンがトリップすると、原子炉がトリップする。これを「タービンの作動状態を表わす信号S_(TBN)が消滅する(タービンがトリップする)と、原子炉の運転状態を表わす信号S_(RTP)が消滅する(原子炉がトリップする)」と考えて、次のような論理式で表わすことができる。

IF (.NOT. S_(TBN)) S_(RTP) = .FALSE.

あるいは

IF (S_(TBN).EQ..FALSE.) S_(RTP) = .FALSE.

上式は、型式(3)の論理回路に相当する。これを、LOGIC 1では以下のように表現して登録する。

CALL LOGL 2 (▽RTPL▽,▽STBN▽,.FALSE.,▽SRTP▽,.FALSE.)

ここで、▽RTPL▽はこの論理回路の名称、▽STBN▽、▽SRTP▽はそれぞれ信号S_(TBN)、S_(RTP)の名称である。

また、本サブプログラムは、次のような場合にも使用することができる。

「安全弁が閉じると、安全弁からの蒸気流出がなくなる」

この場合、「安全弁の開閉状態を表わす信号S_(svL)が消滅する(安全弁が閉じる)と、安全

弁からの蒸気流出有無を表わす信号 $S_{(WSV)}$ が消滅する（蒸気流出がなくなる）」と考えることができ、これを論理式で書くと、

$$\text{IF} (.NOT. S_{(SVL)}) \quad S_{(WSV)} = .FALSE.$$

となり、型式(3)の論理回路に相当する。

(4) 型式(4)の論理回路の登録 (LOGLL1)

(a) 参照形式

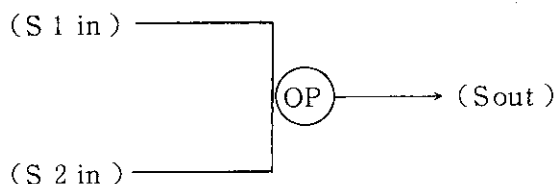
CALL LOGLL 1 (NAMLOG, NSIG1, NOPER, NSIG2, NSIG3, LSIG)

(b) 引数

NAMLOG (入, 文) ……論理回路名称
 NSIG1 (入, 文) ……信号名称-1 (入力信号)
 NOPER (入, 文) ……論理演算子
 NSIG2 (入, 文) ……信号名称-2 (入力信号)
 NSIG3 (入, 文) ……信号名称-3 (出力信号)
 LSIG (入, 論, あるいは整) ……信号値

(c) 機能

下図に示すような論理回路の登録を行う。この論理回路は、LOGIC 1 では型式(4) と呼称するものであり、2つの入力信号の論理結合によって出力信号の状態を定める論理回路を表わすものである。



ここで、 S_{1in} 、 S_{2in} は異なる2つの入力信号、 OP は論理演算子、 S_{out} は出力信号を表わす。論理演算子としては、AND (論理積)、OR (論理和)、EQ (=)、NE (\neq) を使用できる。

(d) 使用例

ポンプの起動信号 $S_{(PPS)}$ が発生し、駆動電源 $S_{(PWR)}$ が供給されると、ポンプが起動する (ポンプ作動/不作動を表わす信号 $S_{(PMP)}$ が発生する)。これを論理式で書くと

$$\text{IF} (S_{(PPS)} .AND. S_{(PWR)}) \quad S_{(PMP)} = .TRUE.$$

となり、これは、型式(4)の論理回路に相当する。

これを LOGIC 1 では次のように表現し登録する。

CALL LOGLL 1 (PMPL, SPPS, AND, SPWR, SPMP, .TRUE.)

ここで、 $\nabla PMPL \nabla$ はこの論理回路の名称、 $\nabla SPPS \nabla$ 、 $\nabla SPWR \nabla$ 、 $\nabla SPMP \nabla$ は信号 $S_{(PPS)}$ 、 $S_{(PWR)}$ 、 $S_{(PMP)}$ の名称である。

この例は、2つの入力信号両方が発生した時に出力信号の状態を定める "2 out of 2"

の論理回路である。

なお、その他の論理演算子を適用した場合は、以下のような論理式に展開される。

$$\textcircled{1} \text{ LOGLL1}(\nabla \text{PMPL} \nabla, \nabla \text{SPPS} \nabla, \nabla \text{OR} \nabla, \nabla \text{SPWR} \nabla, \nabla \text{SPMP} \nabla, \text{LSIG}) \\ \text{IF} (S_{(\text{PPS})} \cdot \text{OR} \cdot S_{(\text{PWR})}) \quad S_{(\text{PMP})} = \text{LSIG}$$

これは、"1 out of 2"の論理回路であり、2つの入力信号のうち、いずれか一方が発生した時に出力信号の状態を定める。

$$\textcircled{2} \text{ LOGLL1}(\nabla \text{PMPL} \nabla, \nabla \text{SPPS} \nabla, \nabla \text{EQ} \nabla, \nabla \text{SPWR} \nabla, \nabla \text{SPMP} \nabla, \text{LSIG}) \\ \text{IF} (S_{(\text{PPS})} \cdot \text{AND} \cdot S_{(\text{PWR})}) \quad S_{(\text{PMP})} = \text{LSIG}$$

$$\text{あるいは、IF} (\cdot \text{NOT} \cdot S_{(\text{PPS})} \cdot \text{AND} \cdot \cdot \text{NOT} \cdot S_{(\text{PWR})}) \quad S_{(\text{PMP})} = \text{LSIG}$$

これは、2つの入力信号の信号値が一致した時のみ、出力信号の状態を規定するものである。

$$\textcircled{3} \text{ LOGLL1}(\nabla \text{PMPL} \nabla, \nabla \text{SPPS} \nabla, \nabla \text{NE} \nabla, \nabla \text{SPWR} \nabla, \nabla \text{SPMP} \nabla, \text{LSIG}) \\ \text{IF} (S_{(\text{PPS})} \cdot \text{AND} \cdot \cdot \text{NOT} \cdot S_{(\text{PWR})}) \quad S_{(\text{PMP})} = \text{LSIG}$$

$$\text{あるいは、IF} (\cdot \text{NOT} \cdot S_{(\text{PPS})} \cdot \text{AND} \cdot S_{(\text{PWR})}) \quad S_{(\text{PMP})} = \text{LSIG}$$

これは、2つの入力信号の信号値が不一致の時のみ、出力信号の状態を規定するものである。

ここで、LSIGは".TRUE."か".FALSE."のいずれかである。

(5) 型式(5)の論理回路の登録 (LOGLL 2)

(a) 参照形式

```
CALL LOGLL 2 (NAMLOG, NSIG 1, LSIG 1, NOPER, NSIG 2, LSIG 2,
              NSIG 3, LSIG 3)
```

(b) 引数

NAMLOG (入, 文) ……論理回路名称

NSIG 1 (入, 文) ……信号名称-1 (入力信号)

LSIG 1 (入, 論あるいは整) ……信号値-1

NOPER (入, 文) ……論理演算子

NSIG 2 (入, 文) ……信号名称-2 (入力信号)

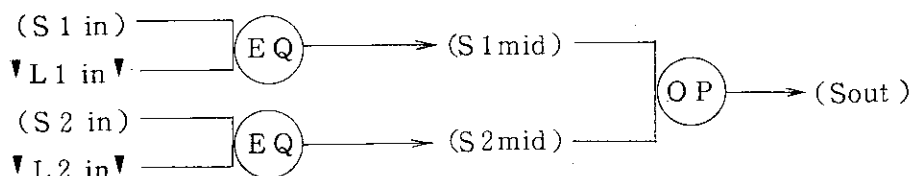
LSIG 2 (入, 論あるいは整) ……信号値-2

NSIG 3 (入, 文) ……信号名称-3 (出力信号)

LSIG 3 (入, 論あるいは整) ……信号値-3

(c) 機能

下図に示すような論理回路の登録を行う。この論理回路は、LOGIC1では型式(5)と呼称するものであり、前述の型式(3)の論理回路を2つ論理結合したものである。即ち、2つの型式(3)の論理回路の出力信号を入力信号とし、それらを論理和や論理積の関係によって出力信号の状態を定める論理回路を表わすものである。



ここで、S1 in, S2 inは入力信号, L1 in, L2 inは指定した信号状態, EQは等号, OPは論理演算子, Soutは出力信号を表わす。また, S1 mid, S2 midは, 中間信号と定義する。論理演算子としては, AND (論理積), OR (論理和)を使用できる。

(d) 使用例

ECCS注入モードで, ポンプが停止 (ポンプの作動 / 不作動を表わす信号 $S_{(PMP)}$ が消滅)するか, あるいは注入弁が閉じる (弁開閉状態を示す信号 $S_{(VLV)}$ が消滅する) と, ECCSの注入が停止 (ECC流量の有無を表わす信号 $S_{(ECC)}$ が消滅) する。これを論理式で書くと,

$$IF (.NOT. S_{(PMP)}, OR. NOT. S_{(VLV)}) \quad S_{(ECC)} = .FALSE.$$

あるいは, $IF ((S_{(PMP)}. EQ. FALSE.) OR. (S_{(VLV)}. EQ. FALSE.))$

$$S_{(ECC)} = .FALSE.$$

となり, これは, 型式(5)の論理回路に相当する。これをLOGIC1では次のように表現し登録する。

```
CALL LOGLL 2 (▽PMPL▽, ▽SPMP▽, .FALSE., ▽OR▽, ▽SVLV▽, .FALSE.,
              ▽SECC▽, .FALSE.)
```

ここで, ▽PMPL▽ は, この論理回路の名称, ▽SPMP▽, ▽SVLV▽, ▽SECC▽はそれぞれ信号 $S_{(PMP)}$, $S_{(VLV)}$, $S_{(ECC)}$ の名称である。

(6) タイマー付論理回路 (形式(6)の論理回路) の登録 (LOGLAG)

(a) 参照形式

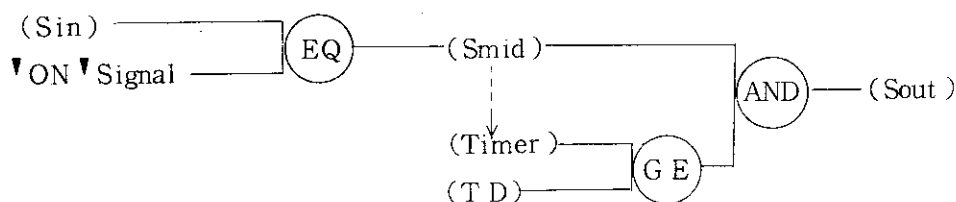
```
CALL LOGLAG (NAMLOG, ISIG, NINS1, NINS2, OSIG, LSIG)
```

(b) 引数

- NAMLOG (入, 文) …… 論理回路名称
- ISIG (入, 文) …… 信号名称 (入力信号)
- NINS1 (入, 文) …… 計測値名称 - 1 (設定値)
- NINS2 (入, 文) …… 計測値名称 - 2 (計測値)
- OSIG (入, 文) …… 信号名称 (出力信号)
- LSIG (入, 論あるいは整) …… 信号値

(c) 機能

時間遅れを模擬するために, 下図に示すようなタイマー付論理回路の登録を行う。この論理回路は, 入力信号の発生以後, ある特定の時間を経過した後に出力信号の状態を規定するものである。



ここで、Sin, Smid, Sout は、信号を表わし、EQ, GE, AND は演算子、TD は時間遅れ設定値を示す。

(d) 使用例

ECCS 起動信号 $S_{(ECC)}$ が発生した後、時間 TIME が、設定した時間遅れ分 $T_{(LAG)}$ を経過すると、ポンプが起動（ポンプ作動 / 不作動を表わす信号 $S_{(PMP)}$ が発生）する。これを LOGIC1 で表現すると、

```
CALL SETPNT (▽TLAG▽, TLAG, M) ..... ①
CALL LOGLL2 (▽PMPL▽, ▽SPMP▽, FALSE., ▽OR▽, ▽SVLV▽, FALSE.,
             .TRUE.) ..... ②
```

となる。この例では、①で時間遅れ分を設定し、②でタイマー付論理回路を登録する。ここで、▽TLAG▽ は時間遅れ設定値 $T_{(LAG)}$ の設定値名称、▽TIME▽ は計測値 TIME の名称を示す。また、▽TDEC▽ は、論理回路の名称、▽SECC▽, ▽SPMP▽ は、それぞれ信号 $S_{(ECC)}$, $S_{(PMP)}$ の名称である。

各サブプログラムとも、通常は論理回路に名称をつけて論理式の形で登録を行うが、既に同一の名称が登録されている場合には、その論理式を更新する。また、名称を付けずに登録することもできる。論理回路の登録時に、必要となる計測値名称や信号名称が未登録であってもエラーとはならず、各サブプログラムが前述の INSREG あるいは SIGREG を用いて自動登録する。

3.4.2 制御論理の変更

3.4.1 節の各サブプログラムを用いて模擬した一連の制御論理は、論理回路の登録順に実行される。一度登録した制御論理は、次のような方法で変更することができる。

(i) 論理回路の変更

3.4.1 節で説明した各サブプログラムを用いて、既に登録されている論理回路名称と同一名称を入力指定することで、その論理式を更新する。

(ii) 論理回路の挿入

後述するサブプログラム LOGADD を用いて、新しい論理回路名称を一連の制御論理の適当な位置に挿入し、その後、(i) で述べた方法でその論理式を更新する。

(iii) 論理回路の削除

後述するサブプログラム LOGDEL を用いて、削除対象とする論理回路名称を入力指定することにより、その論理回路の削除を行う。

(iv) 論理回路の順序変更

後述するサブプログラム LOGMOV を用いて、入力指定する論理回路の位置を移動することにより、論理回路の登録順序を変更する。

各サブプログラムの使用方法は次のとおりである。

(1) 論理回路名称の挿入 (LOGADD)

(a) 参照形式

CALL LOGADD (NAMLG1, NAMLG2)

(b) 引数

NAMLG1 (入, 文) ……挿入先の論理回路名称

NAMLG2 (入, 文) ……挿入する論理回路名称

(c) 機能

3.4.1節で説明した各サブプログラムにより登録されているいくつかの論理回路のうち ∇ NAMLG1 ∇ という名称の論理回路の直後に, ∇ NAMLG2 ∇ という名称の論理回路を挿入する。この場合, 挿入した論理回路 ∇ NAMLG2 ∇ は論理回路 ∇ NAMLG1 ∇ と同じになる。挿入した論理回路を変更したい場合には, 3.4.1節で説明した各サブプログラムを用いて登録・更新する。

なお, 次の場合には論理回路の挿入は行わずエラーとなる。

①挿入したい論理回路名称 ∇ NAMLG2 ∇ が既登録である場合

②挿入したい箇所の直前の論理回路名称 ∇ NAMLG1 ∇ が未登録である場合 (∇ NAMLG1 ∇ を入力指定しない場合も含む)

(d) 使用例

論理回路 ∇ PMPL ∇ の直後に, 次の論理式で示される論理回路 (型式(1)) ∇ STOP ∇ を挿入する。

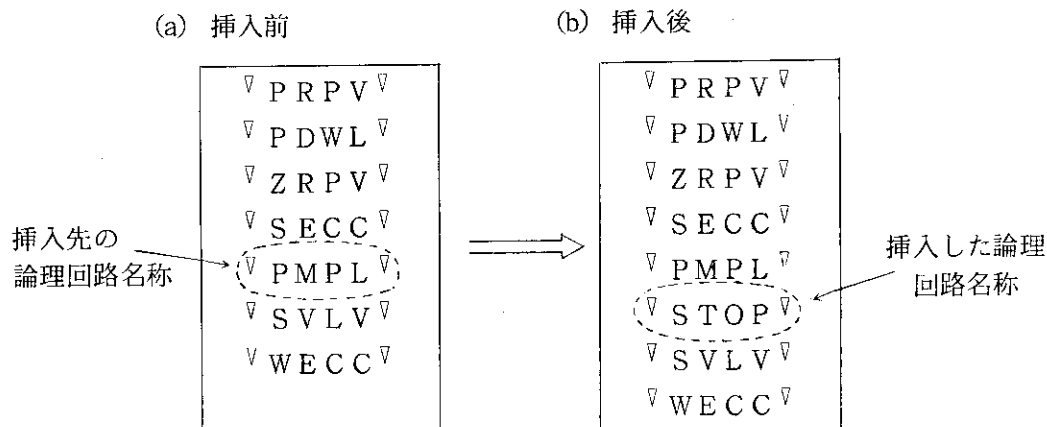
IF (S_(STP)) S_(PMP) = .FALSE.

この場合, LOGIC1では以下のように表現すれば, 既に登録されている情報に追加される。

CALL LOGADD (∇ PMPL ∇ , ∇ STOP ∇)

CALL LOGL1 (∇ STOP ∇ , ∇ SSTP ∇ , ∇ SPMP ∇ , .FALSE.)

LOGADDで挿入箇所を確保し, そこにLOGL1で論理式を登録する。具体例を以下に示す。



(2) 論理回路の削除 (LOGDEL)

(a) 参照形式

CALL LOGDEL (NAMLOG)

(b) 引数

NAMLOG (入, 文) ……削除する論理回路の名称

(c) 機能

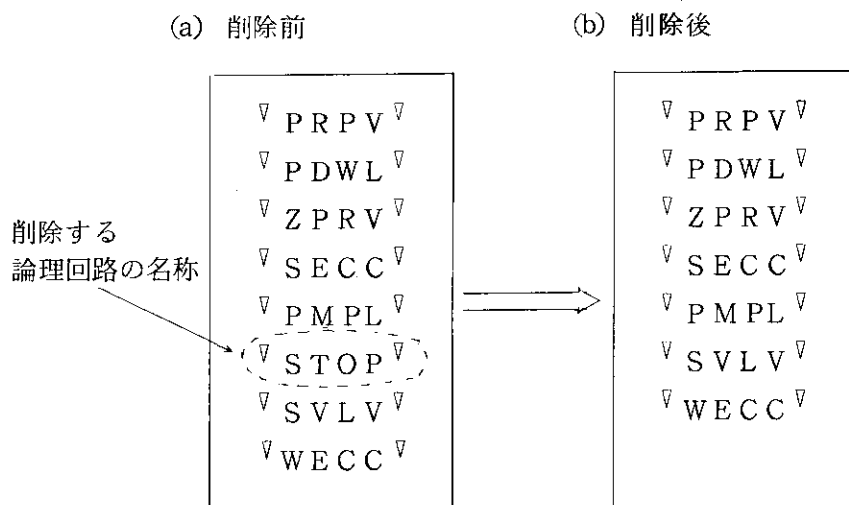
入力指定した論理回路名称に該当するものを、登録されている論理回路の中から削除する。しかし、登録されている論理回路であっても、その名称の付いていないものは削除対象外となる。

(d) 使用例

登録されている論理回路の中から、論理回路名称 ∇ STOP ∇ に対応する論理回路の内容を全て削除する。この場合、LOGIC1では次のように表現する。

CALL LOGDEL (∇ STOP ∇)

具体例を以下に示す。



(3) 論理回路の移動 (LOGMOV)

(a) 参照形式

CALL LOGMOV (NAMLG1, NAMLG2)

(b) 引数

NAMLG1 (入, 文) ……移動先の論理回路名称

NAMLG2 (入, 文) ……移動する論理回路名称

(c) 機能

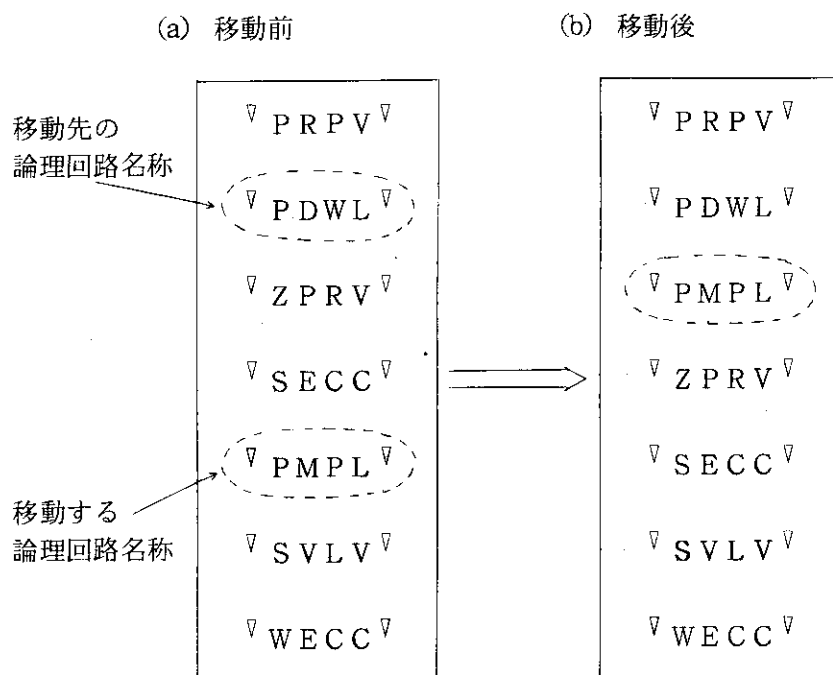
既に登録してある論理回路の中で、 ∇ NAMLG1 ∇ という名称の論理回路の直後に ∇ NAMLG2 ∇ という名称の論理回路を移動する。ただし、論理回路 ∇ NAMLG1 ∇ あるいは、 ∇ NAMLG2 ∇ が未登録であるか、入力指定がない場合にはエラーとなり、移動を行わない。

(d) 使用例

下記(a)に示すような順序で登録されている論理回路の中で、 ∇ PDWL ∇ という名称の論理回路の直後に、 ∇ PMPL ∇ という名称の論理回路を移動させる場合、LOGIC1では次のように表現する。

CALL LOGMOV (∇ PDWL ∇ , ∇ PMPL ∇)

具体例を以下に示す。



3.4.3 論理回路の出力 (WLOGIC)

(a) 参照形式

CALL WLOGIC (NPAGE)

(b) 引数

NPAGE (入, 整) ……改ページ指示子

(c) 機能

内部共通領域 LOGCOM に登録されている論理回路に関する情報を出力する。引数 NPAGE が "1" のときは改ページ, それ以外のときは改ページせずに出力する。

(d) 使用例

CALL WLOGIC (0)

上記の表現を用いて, 論理回路の登録情報を出力させた例を以下に示す。

*** LOGICAL FUNCTION *****		MAXLOG = 7					
NO	NLOG	JTYP	MM1	NOP	MM2	MM3	LL3
1	SVOL	1	1 (INS)	GT	2 (INS)	1 (SIG)	T
2	BVOL	1	3 (INS)	GT	4 (INS)	2 (SIG)	F
3	WVOL	4	1 (SIG)	AND	2 (SIG)	3 (SIG)	T
4	FVOL	3	-3 (SIG)	—	—	4 (SIG)	T
5	TDEL	6	4 (SIG)	5 (INS)	6 (INS)	5 (SIG)	T
6	ISSL	2	2 (SIG)	—	—	6 (SIG)	F
7	PPOL	5	-5 (SIG)	OR	-6 (SIG)	7 (SIG)	T

ここで

NLOG : 論理回路名称

JTYP : 論理回路の型式 (論理回路型式(1)~(6)に対応)

MM1~3: (INS) 計測値登録番号

(SIG) 信号値登録番号

NOP : 論理演算子あるいは算術演算子

LL3 : 出力信号の信号値 ("T" は TRUE, "F" は FALSE を表わす)

である。また、JTYPが3または5のときMM1あるいはMM2の欄に負符号が付いているが、これは、その信号値がFALSEであることを示す。また、JTYPが6のときは、NOPは計測値登録番号を示す。

3.5 論理演算の実行 (LOGGO)

本節では、前節で説明したサブプログラムによりモデル化された一連の制御論理を実行するサブプログラムLOGGOについて説明する。

(a) 参照形式

CALL LOGGO

(b) 引数

なし

(c) 機能

計測系、信号系及び論理回路について、前述の各サブプログラムによりモデル化し登録された情報を基に、論理回路の登録順にその論理演算を実行する。これにより内部共通領域 SIGCOM内に登録されている各信号の状態を更新する。LOGGO実行後、必要な信号値をこのSIGCOMから前述のサブプログラムGETSIGを用いて引き出し、熱水力計算コードで引用する。

(d) 使用例

```
CALL LOGGO
```

```
CALL GETSIG (▽ECCS▽, LECC, M)
```

論理演算を実行した後、▽ECCS▽なる名称の信号が発生 (LECC=, TRUE.) しているかどうかを調べる。

3.6 実行結果の出力指定 (SIGVRF)

前述したサブプログラムLOGGOにより制御論理を実行した結果、ある信号の状態が変化した場合に、主要な各種計測値を出力することができる。本サブプログラムSIGVRFは、出力すべき計測値を指定するものである。

(a) 参照形式

```
CALL SIGVRF (NSIG, NIST, NINST)
```

(b) 引数

NSIG (入, 文) ……出力対象とする信号名称

NIST (入, 整) ……出力する計測値の数 (最大10個)

NINST (入, 文) ……出力する計測値名称

ここで、NINSTで指定する計測値名称は、NISTで指定する個数と同じ数だけ入力する必要がある。また、指定する計測値名称は、内部共通領域 INSCOMに登録されているものを用いる。

(c) 機能

トランジェント計算の各時刻において、出力対象とする信号の信号値が変化した場合に、その時点での計測値に関する情報を出力するよう指定する。計測値に関する情報は、計測値名称-計測値の組で出力される。計測値は、熱水力計算コードによる各種パラメータの計算値である。

(d) 使用例

注入弁開信号 S_(IJV)が発生した時の時刻 TIME 及びその時点での一次系圧力 P_(RPV) (熱水力計算コードによる計算値) を出力する場合、LOGIC1 では次のように表現する。

```
CALL SIGVRF (▽SIJV▽, 2, ▽TIMEPRPV▽)
```

ここで、▽SIJV▽は信号 S_(IJV)の名称、▽TIMEPRPV▽は、計測値 TIME と P_(RPV) の名称である。この場合の出力例は以下の通りである。

```
◀SIJV SWITCHED ON ▶
```

```
TIME 2.1500E+02 PRPV 1.1688E+05
```

4. LOGIC 1 の使用例

本章では、LOGIC1を用いて現実のシステム作動論理をモデル化した例を示す。対象とするシステムとしてBWRの非常用炉心冷却系の1つである、炉心スプレー系を選択した。ここに示すのは、Browns Ferry 1号炉のプラントデータを基にしており、その作動論理はFig. 4.1(a)に示す通りである。

図示するように、ドライウエル内の圧力が設定値(16.7 psia)を越えると、ドライウエル圧力高信号が発生する。また、原子炉水位が設定値(380 inch)より低くなると、原子炉水位低信号が発生する。これらの信号のうち、どちらか一方の信号を受けて炉心スプレー系のポンプ起動信号が発生する。この起動信号発生から7秒後に炉心スプレー系のポンプが自動起動する(時間遅れ)。一方、一次系圧力が設定値(31.7 kgf/cm²)より低くなると、原子炉圧力低信号が発生する。炉心スプレーポンプ起動と原子炉圧力低信号の一致で炉心スプレー系の注入弁が開き、炉内にスプレー水が注入される。

このような自動作動論理の他に、ここではFig. 4.1(b)に示すように事故開始後40分経過時に運転員が介入して、スプレーポンプを停止させるものとする。図中、注入弁作動信号は、運転員による注入弁開閉操作と考え、この信号が発生した時は「開操作」消滅した時は「閉操作」を意味する。

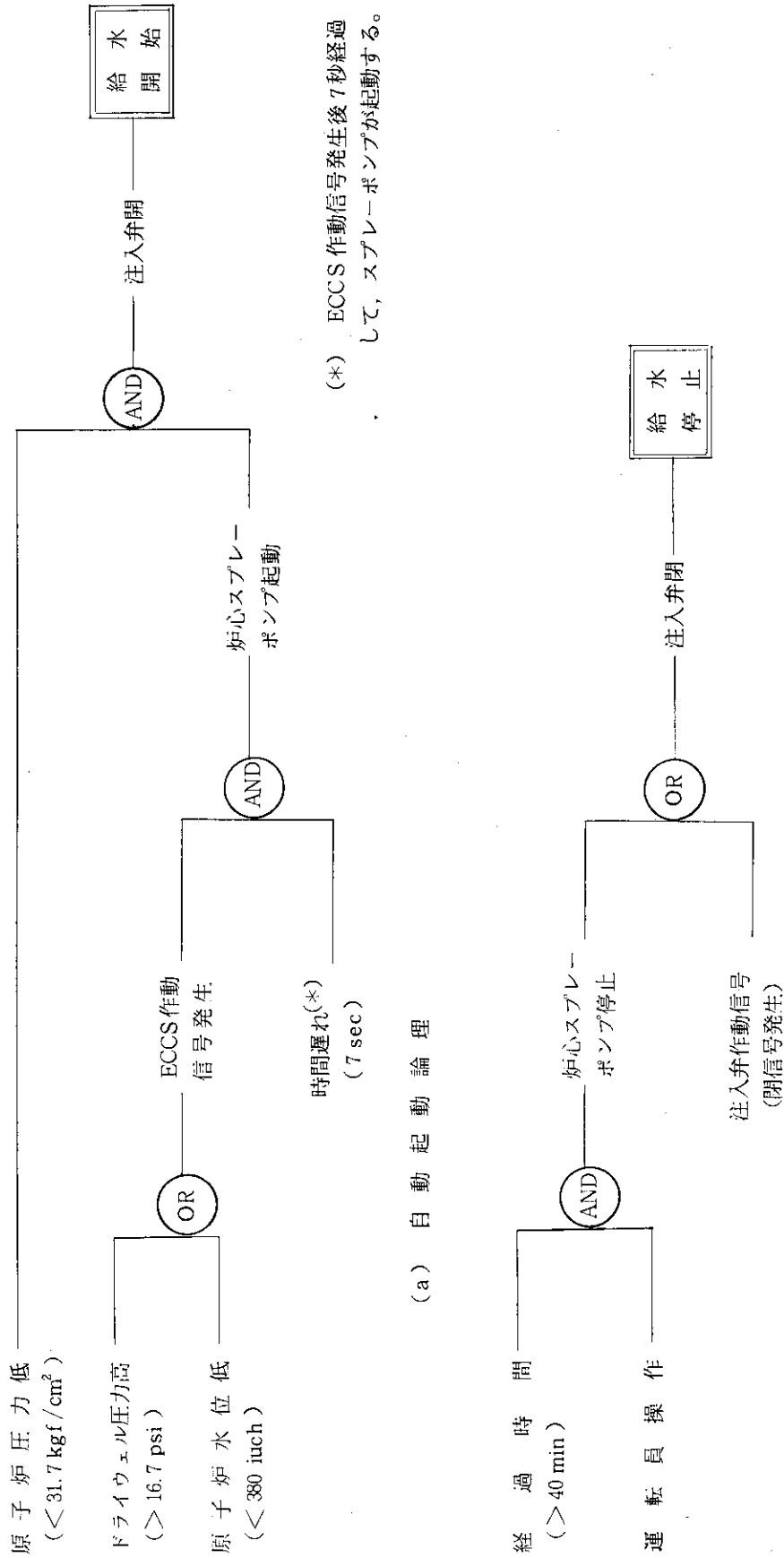
4.1 サンプル・プログラムの作成

Fig. 4.1に示す炉心スプレー系作動論理を計測値名称、信号名称及び論理演算子を用いて正確、詳細に表現するとFig. 4.2に示すようになる。この図に従って以下に作動論理を説明する。

ドライウエル圧力 $P_{(PDW)}$ (熱水力コード計算値)が、設定値 $P_{(PDWS)}$ を超えるとドライウエル圧力高信号 $S_{(PDW)}$ が発生する(信号値が.TRUE.になる)また、原子炉水位 $Z_{(ZRV)}$ (計算値)が設定値 $Z_{(ZRVs)}$ より低くなると、原子炉水位低信号 $S_{(ZRV)}$ が発生する。

これらの信号のうち、どちらか一方の信号を受けてECCS作動信号 $S_{(TEP)}$ が発生する。この信号発生後、時刻TIMEが時間遅れ設定値TEPL(7秒)を経過すると、スプレーポンプが起動する(スプレーポンプの起動/停止を表わす信号 $S_{(SEP)}$ が発生する)。一方、熱水力パラメータ、一次系圧力 $P_{(PRV)}$ (計算値)が設定値 $P_{(PRVS)}$ より低くなると、原子炉圧力低信号 $S_{(PRV)}$ が発生する。この信号とスプレーポンプ起動によって、注入弁が開となり(注入弁の開閉状態を表わす信号 $S_{(IIV)}$ が発生する)、炉心へのスプレー注入流量が確保される(スプレー注入流量の有無を表わす信号 $S_{(WEP)}$ が発生する)。即ち、給水開始となる。

給水停止は、事故開始から時刻TIMEが設定時間TEPS(40分)を経過した後、運転員操作によって行われる。この操作により、まず、スプレーポンプが停止し(スプレーポンプの起動/停止を表わす信号 $S_{(SEP)}$ が消滅する。即ち、.FALSE.になる)。さらに、注入弁が閉じる(注入弁の開閉状態を表わす信号 $S_{(IIV)}$ が消滅する)。また、注入弁作動信号 $S_{(IIV)}$ が閉信号になる



(*) ECCS 作動信号発生後 7 秒経過して、スプレ-ポンプが起動する。

(a) 自動起動論理

(b) 運転員による停止論理

Fig. 4.1 炉心スプレ-系の作動論理例

($S_{(STV)}$ が消滅する)と、注入弁が閉じる。注入弁閉により、炉内へのスプレー注入流量は失くなり(スプレー流量の有無を表わす信号 $S_{(WEF)}$ が消滅する)、給水停止となる。

上述した作動論理を考慮したサンプル・プログラム作成を、次の3段階に分けて行った。

- (1) 主プログラム(熱水力計算部分に対応するもの)
- (2) 炉心スプレー系の自動起動論理
- (3) 運転員の操作による作動停止論理

4.1.1 主プログラムの作成

本節では、炉心スプレー系の自動起動論理、運転員の操作による炉心スプレー系の作動停止を考慮した熱水力計算プログラムの作成例を示す。熱水力計算部分では、各種の熱水力パラメータ(圧力、水位など)の増分値を一定とした単純なモデルを想定した。Fig.4.3に作成したプログラムを示す。各部分の内容は次の通りである。

- ① ループの回数(MAXLOP)、計算結果の出力間隔(NP)の指定、及びスプレー注入流量の有無を表わす信号の信号値LWEFを論理型変数と宣言する。
- ② 単位換算プログラム・ライブラリUCL2の自動単位登録機能呼び出し、使用単位系をMKSCとする。
- ③ LOGIC1初期設定サブプログラムの呼び出し(LOGINI)、及び4.1.2節の炉心スプレー自動起動論理の登録を行う(AUTORG)。このとき、登録情報を出力する。
- ④ 4.1.3節の運転員操作による停止論理を登録(MANURG)し、登録情報を出力する。
- ⑤ 変化信号の出力指示。スプレー注入流量の有無を表わす信号 ∇WEF が変化したら、計測値名称 $\nabla TIME$ 、 ∇PDW 、 ∇ZRV 、 ∇PRV の計測値とともに、メッセージを出力することを指示する。
- ⑥ UCL2の単位換算関数(UCONB)を用いて各種熱水力パラメータの初期状態を以下のように設定する。

TIME	: 経過時間	(0.0 sec)
PDW	: ドライウェル内圧力	(14.7 psia)
ZRV	: 原子炉水位	(500.0 inch)
PRV	: 一次系圧力	(1050.0 psia)
DTM	: タイムステップ	(1.0 min)
DPD	: ドライウェル内圧力増分	(8.0 psia/hr)
DZR	: 原子炉水位増分	(-10.0 inch/min)
DPR	: 一次系圧力増分	(-20.0 psia/min)

- ⑦ 経過時間の更新を行う。
- ⑧ スプレー注入流量の有無を表わす信号(∇WEF)を引用する。
- ⑨ 各状態量を計算する。

炉心スプレー系作動時は

$$X = X - dx \cdot dt / 2.0$$

とし

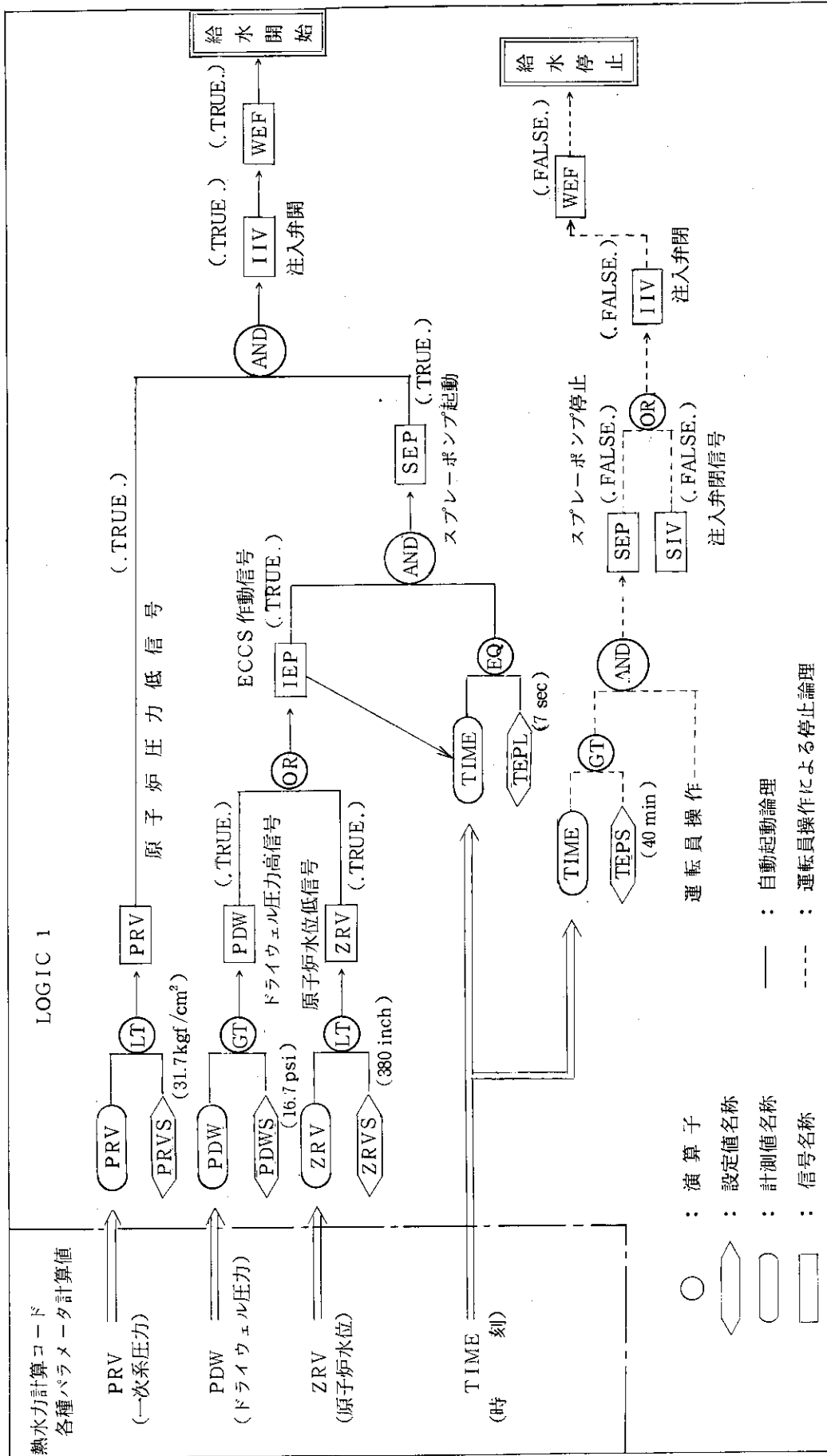


Fig. 4.2 炉心スプレー系の作動論理モデル

```

C*****
C** SAMPLE PROBLEM OF CORE SPRAY ACTION ***
C*****

①  PARAMETER (MAXLOP = 50)
    PARAMETER (NP      = 10)
    LOGICAL*4  LWEF

C** UCL2 *****
②  CALL AUTREG (1, 1, 1, 0, 0)
    CALL AUTSYS (1, 'MKSC')

C** LOGIC1 INITIAL AND AUTOMATIC REGISTRY *****
③  WRITE(6,('1 == AUTOMATIC REGISTRY OF CORE SPRAY =='))
    CALL LOGINI (6, 6)
    CALL AUTORG
    CALL WINSTR ( 0 )
    CALL WSIGNL ( 0 )
    CALL WLOGIC ( 0 )

C** LOGIC1 MANUAL OPERATION REGISTRY *****
④  WRITE(6,('1 == MANUAL REGISTRY OF CORE SPRAY =='))
    CALL MANURG
    CALL WINSTR ( 0 )
    CALL WSIGNL ( 0 )
    CALL WLOGIC ( 0 )

C** LOGIC1 SIGNAL VERIFY REGISTRY *****
⑤  CALL SIGVRF ('WEF ', 4, 'TIMEPDW ZRV PRV')

C** INITIAL CONDITIONS *****
⑥  PRV = 1050.0*UCONB('PSI', 3, IDQ)
    PDW = 14.7*UCONB('PSI', 3, IDQ)
    ZRV = 500.0*UCONB('INCH', 4, IDQ)
    TIME = 0.0

    DPR = -20.0*UCONB('PSI/MIN', 7, IDQ)
    DPD = 8.0*UCONB('PSI/HOUR', 8, IDQ)
    DZR = -10.0*UCONB('INCH/MIN', 8, IDQ)
    DTM = 1.0*UCONB('MIN', 3, IDQ)

C** TRANSIENT CALCULATION *****
    DO 4000 LOOP = 1, MAXLOP

⑦  TIME = DTM*LOOP

⑧  CALL GETSIG ('WEF ', LWEF, M)
    IF (LWEF) THEN
⑨  PRV = PRV - DPR*DTM/2.0
    PDW = PDW - DPD*DTM/2.0
    ZRV = ZRV - DZR*DTM/2.0
    ELSE
    PRV = PRV + DPR*DTM
    PDW = PDW + DPD*DTM
    ZRV = ZRV + DZR*DTM
    END IF

⑩  CALL SENSOR ('PRV ', PRV, M)
    CALL SENSOR ('PDW ', PDW, M)
    CALL SENSOR ('ZRV ', ZRV, M)
    CALL SENSOR ('TIME', TIME, M)

⑪  CALL LOGGO

⑫  IF (MOD(LLOOP - 1, NP) .EQ. 0) THEN
    WRITE(6,('0==',I4,' =====')) LOOP
    CALL WINSTR ( 0 )
    CALL WSIGNL ( 0 )
    END IF

⑬  CALL SPLEDZ ( TIME, PDW, ZRV, PRV )

4000 CONTINUE

STOP
END

```

Fig. 4.3 主プログラム例

不作動時は

$$X = X + dx \cdot dt$$

とする。ここでXは各状態量、dxはその増分、dtはタイムステップを示す。

- ⑩ 計測値の更新を行う。
- ⑪ 論理演算の実行を行う。
- ⑫ 計算結果を出力する。本例では、LOGIC1の出力機能を用いる。
- ⑬ SPL-EDIT (作図プログラム)の呼び出しを行う。

4.1.2 自動起動論理プログラムの作成

本節では、Fig. 4.2に示した炉心スプレー系の作動論理のうち、自動起動論理部分について、LOGIC1によるプログラム作成例を示す。作成したプログラムをFig. 4.4に示し、この図に従って、プログラム各部の内容を説明する。

```

C*****
C* AUTOMATIC REGISTRY OF CORE SPRAY
C*****
      SUBROUTINE AUTORG

C*****
C** SET INSTR. VALUE ***
① C*****
C-- UNIT CONVERSION ----
      PRVS = 31.7*UCONB('KGF/CM2', 7, IDQ)
      PDWS = 16.7*UCONB('PSI', 3, IDQ)
      ZRVS = 380.0*UCONB('INCH', 4, IDQ)
      TEPL = 7.0*UCONB('SEC', 3, IDQ)
C-- POINT SETTING ----
      CALL SETPNT('PRVS', PRVS, M)
      CALL SETPNT('PDWS', PDWS, M)
      CALL SETPNT('ZRVS', ZRVS, M)
      CALL SETPNT('TEPL', TEPL, M)

C*****
② C** SET SIGNAL STATE ***
C*****
      CALL SIGSET('PDW ', .FALSE., M)
      CALL SIGSET('ZRV ', .FALSE., M)
      CALL SIGSET('PRV ', .FALSE., M)
      CALL SIGSET('IEP ', .FALSE., M)
      CALL SIGSET('SEP ', .FALSE., M)
      CALL SIGSET('IIV ', .FALSE., M)
      CALL SIGSET('WEF ', .FALSE., M)

C*****
③ C** SET LOGICAL FUNCTION ***
C*****
      CALL LOGXX ('PRVO','PRV ','LT ','PRVS','PRV ', .TRUE. )
      CALL LOGXX ('PDWO','PDW ','GT ','PDWS','PDW ', .TRUE. )
      CALL LOGXX ('ZRVO','ZRV ','LT ','ZRVS','ZRV ', .TRUE. )

      CALL LOGLL1('IEPO','PDW ','OR ','ZRV ','IEP ', .TRUE. )
      CALL LOGLAG('SEPO','IEP ','TEPL','TIME','SEP ', .TRUE. )

      CALL LOGLL1('IIVO','PRV ','AND ','SEP ','IIV ', .TRUE. )
      CALL LOGL1 ('WEFO','IIV ', 'WEF ', .TRUE. )

      RETURN
      END

```

Fig. 4.4 炉心スプレー系自動起動論理のプログラム例

- ① UCL2の単位換算関数(UCONB)を用いて、次のように設定値を登録する。
 PDWS : ドライウェル圧力高信号設定圧 (16.7 psia)
 ZRVS : 原子炉水位低信号設定水位 (380 inch)

PRVS : 原子炉圧力低信号設定圧 (31.7 kg f/cm²)

TEPL : スプレーポンプ作動の時間遅れ (7.0 sec)

② 以下に示す各種信号の初期状態を設定し登録する。本例では、全て“.FALSE.”と初期状態を登録している。

PDW : ドライウエル圧力高信号

ZRV : 原子炉水位低信号

PRV : 原子炉圧力低信号

IEP : ECCS作動信号

SEP : スプレーポンプ起動/停止を表わす信号

IIV : 注入弁開閉状態を表わす信号

WEF : スプレー注入流量の有無を表わす信号

主プログラムでは、炉心スプレー系の作動/不作動を信号 ∇ WEF ∇ の信号値で判断する。即ち、信号 ∇ WEF ∇ が“.TRUE.”ならば炉心スプレー系作動となり、信号 ∇ WEF ∇ が“.FALSE.”ならば炉心スプレー系不作動となる。

③ Fig. 4.2 に示す炉心スプレー系の作動論理のうち、自動起動論理を、LOGIC1 の論理回路登録機能を用いて表現する。各論理回路の内容を以下に説明する。

- (a) 論理回路 PRVO : 一次系圧力 $P_{(PRV)}$ が、設定値 $P_{(PRVS)}$ より低くなると、原子炉圧力低信号 $S_{(PRV)}$ が発生する(.TRUE.になる)。
- (b) 論理回路 PDWO : ドライウエル圧力 $P_{(PDW)}$ が設定値 $P_{(PDWS)}$ より高くなると、ドライウエル圧力高信号 $S_{(PDW)}$ が発生する。
- (c) 論理回路 ZRVO : 原子炉水位 $Z_{(ZRV)}$ が設定値 $Z_{(ZRVs)}$ より低くなると、原子炉水位低信号 $S_{(ZRV)}$ が発生する。
- (d) 論理回路 IEPO : ドライウエル圧力高信号 $S_{(PDW)}$ あるいは原子炉水位低信号 $S_{(ZRV)}$ のいずれかが発生すると、ECCS作動信号 $S_{(IEP)}$ が発生する。
- (e) 論理回路 SEPO : ECCS作動信号 $S_{(IEP)}$ が発生してから、時間TIMEが時間遅れ設定値TEPLを経過すると、ポンプ起動/停止を表わす信号 $S_{(SEP)}$ が発生する(即ち、ポンプが起動する)。
- (f) 論理回路 IIVO : ポンプが起動(信号 $S_{(SEP)}$ が発生)し、原子炉圧力低信号 $S_{(PRV)}$ が発生すると、注入弁開閉状態を表わす信号 $S_{(IIV)}$ が発生する(即ち、注入弁が開く)。
- (g) 論理回路 WEFO : 注入弁が開く(信号 $S_{(IIV)}$ が発生)と、スプレー注入流量の有無を表わす信号 $S_{(WEF)}$ が発生する。即ち、炉心スプレー系による注入が開始する。

4.1.3 運転員操作による停止論理プログラムの作成

本例では、運転員が事故開始からの経過時間を判断し、ある時刻(40分)になった時、スプレーポンプを停止するものと仮定する。本節では、このような運転員の操作による炉心スプレー系の停止論理について、LOGIC1によるプログラム作成例を示す。作成したプログラムを、Fig. 4.5 に示し、この図に従って、プログラム各部の内容を説明する。


```

C*****
C** MANUAL STOP OF CORE SPRAY BY TIME
C*****
      SUBROUTINE MANURG

① C** LOGICAL TO STOP ECC FLOW BY TIME ***
    (a) CALL LOGADD ('SEPO','SEPX')
        CALL LOGXX ('SEPX','TIME', 'GT ', 'TEPS', 'SEP ', 0)

    (b) CALL LOGADD ('IIVO','IIVX')
        CALL LOGLL2 ('IIVX','SEP ', 0, 'OR ', 'SIV ', 0, 'IIV ', 0)

    (c) CALL LOGADD ('WEFO','WEFX')
        CALL LOGL2 ('WEFX','IIV ', 0, 'WEF ', 0)

② C** TIME SET POINT ***
      TEPS = 40.0*UCONB('MIN', 3, IDQ)
      CALL SETPNT ('TEPS', TEPS, M)

③ C** INITIAL SIGNAL STATE ***
      CALL SIGSET ('SIV ', 1, M)

      RETURN
      END

```

Fig. 4.5 運転員の介入による炉心スプレー系停止論理のプログラム例

- ① 4.1.2節に示した自動起動論理に、運転員の介入による停止論理を追加登録する。
LOGADDで挿入位置と挿入する論理回路名称を登録し、その後に論理回路の内容を登録する。
- (a) 論理回路 ∇ SEPO ∇ の直後に、次のような論理回路を挿入する。
論理回路SEPX：時刻TIMEが、設定時間TEPSを経過すると、ポンプ起動/停止を表わす信号 $S_{(SEP)}$ が消滅(ポンプが停止)する。
- (b) 論理回路 ∇ IIVO ∇ の直後に、次のような論理回路を挿入する。
論理回路IIVX：信号 $S_{(SEP)}$ が消滅(ポンプが停止)するか、もしくは、注入弁作動信号 $S_{(SIV)}$ が消滅(注入弁閉信号が発生)すると、注入弁開閉状態を表わす信号 $S_{(IIV)}$ が消滅する(注入弁が閉じる)。
- (c) 論理回路 ∇ WEFO ∇ の直後に、次のような論理回路を挿入する。
論理回路WEFX：信号 $S_{(IIV)}$ が消滅する(注入弁が閉じる)と、スプレー注入流量の有無を表わす信号 $S_{(WEF)}$ が消滅する。即ち、給水停止となる。
- ② 事故開始から運転員が介入して炉心スプレー系を停止させるまでの経過時間を40分とし、単位換算プログラム・ライブラリUCL2の単位換算関数UCONBを用いて TEPS という設定値名称で登録する。
- ③ 注入弁作動信号 $S_{(SIV)}$ の初期状態を、TRUE(即ち、開信号が発生している)と設定する。この信号が、FALSEの時、閉信号が発生するものとする。

4.2 実行結果の出力

サンプルプログラムの出力結果を Fig. 4.6～4.9に示す。Fig. 4.6は、炉心スプレー系の自動起動論理の登録内容を示す。Fig. 4.7は、自動起動論理に、運転員によるスプレーポンプ停止論理を追加登録した後の登録内容を表わす。これらの図の説明は以下に示す通りである。

- ① 計測値及び設定値に関する初期状態の登録内容を表わす。ここで、 $(-7.23701 E+75)$ という値はデフォルト値である。
- ② 各種信号の初期状態登録内容を表わす。ここで、 $\nabla OFF \nabla$ は、 $.FALSE$ を、 $\nabla ON \nabla$ は $.TRUE$ を示す。
- ③ 制御論理の登録内容を表わす。制御論理は、論理回路の登録順序で定まるため、Fig. 4.6とFig. 4.7を比べると、運転員の介入による制御論理の変更が判る (SEPX, IIVX, WEFXが追加登録されている)。

また、Fig. 4.8は、トランジェント計算結果の一部を出力したものである。図の説明は以下の通りである。

- ① トランジェント各時刻において、計測値 (熱水力コード計算値) 及び、設定値を出力表示する。
- ② トランジェント各時刻において、各種信号の状態を出力表示する。
- ③ トランジェント各時刻において信号 $S_{(WEF)}$ が発生した時点で、その旨出力する。また、その時点における各状態量 (熱水力コード計算値) を出力表示する。この例では、 $TIME = 1800$ (sec) で炉心スプレー系が起動したことを示し、その時点のドライウェル圧力 (PDW)、原子炉水位 (ZRV)、一次系圧力 (PRV) を出力する。

Fig. 4.9は、各状態量 (熱水力コード計算値) を SPL-PLOT (SPLPACK-1のサブプログラムの1つ) を用いて図化出力したものである。時刻1800秒で炉心スプレー系が起動し、注水が開始した。そして2460秒で運転員が介入してスプレーポンプを停止させたため、炉心スプレー系による注水が停止した。

```

=== AUTOMATIC REGISTRY OF CORE SPRAY ===
① *** REGISTERED INSTRUMENT ***** MAXINS = 8
1 PRVS 3.17000E+05 2 PDWS 1.17413E+04 3 ZRVS 9.65199E+00 4 TEPL 7.00000E+00 5 PRV -7.23701E+75
6 PDW -7.23701E+75 7 ZRV -7.23701E+75 8 TIME -7.23701E+75

*** REGISTERED SIGNAL ***** MAXSIG = 7
1 PDW OFF 2 ZRV OFF 3 PRV OFF 4 IEP OFF 5 SEP OFF
6 IIV OFF 7 WEF OFF

*** LOGICAL FUNCTION ***** MAXLOG = 7
③ NO. NLOG JTYPE MM1 NOP MM2 MM3 LL3
1 PRVO 1 5<INS> LT NOP 1<INS> 3<SIG> T
2 PDWO 1 6<INS> GT 2<INS> 1<SIG> T
3 ZRVO 1 7<INS> LT 3<INS> 2<SIG> T
4 IEPV 4 1<SIG> OR 4<INS> 4<SIG> T
5 SEPO 6 4<SIG> AND 8<INS> 5<SIG> T
6 IIVV 4 3<SIG> AND 6<SIG> 6<SIG> T
7 WEFO 2 6<SIG> 0 7<SIG> T
    
```

Fig. 4.6 炉心スプレー系自動起動論理の登録内容

```

=== MANUAL REGISTRY OF CORE SPRAY ===
① *** REGISTERED INSTRUMENT ***** MAXINS = 9
1 PRVS 3.17000E+05 2 PDWS 1.17413E+04 3 ZRVS 9.65199E+00 4 TEPL 7.00000E+00 5 PRV -7.23701E+75
6 PDW -7.23701E+75 7 ZRV -7.23701E+75 8 TIME -7.23701E+75 9 TEPS 2.40000E+03

*** REGISTERED SIGNAL ***** MAXSIG = 8
② 1 PDW OFF 2 ZRV OFF 3 PRV OFF 4 IEP OFF 5 SEP OFF
6 IIV OFF 7 WEF OFF 8 SIV ON

*** LOGICAL FUNCTION ***** MAXLOG = 10
③ NO. NLOG JTYPE MM1 NOP MM2 MM3 LL3
1 PRVO 1 5<INS> LT NOP 1<INS> 3<SIG> T
2 PDWO 1 6<INS> GT 2<INS> 1<SIG> T
3 ZRVO 1 7<INS> LT 3<INS> 2<SIG> T
4 IEPV 4 1<SIG> OR 4<INS> 4<SIG> T
5 SEPO 6 4<SIG> AND 8<INS> 5<SIG> T
6 SEPX 1 8<INS> GT 9<INS> 5<SIG> F
7 IIVV 4 3<SIG> AND 6<SIG> 6<SIG> T
8 IIVX 5 -5<SIG> OR -8<SIG> 6<SIG> F
9 WEFO 2 6<SIG> 0 7<SIG> T
10 WEPX 3 -6<SIG> 0 7<SIG> F
    
```

Fig. 4.7 運転員の操作によるスプレーポンプ停止論理を追加した後の登録内容

```

== 21 =====
① *** REGISTERED INSTRUMENT ***** MAXINS = 9
1 PRVS 3.17000E+05 2 PDWS 1.17413E+04 3 ZRVS 9.65199E+00 4 TEPL 7.00000E+00 5 PRV 4.42934E+05
6 PDW 1.23037E+04 7 ZRV 7.36599E+00 8 TIME 1.26000E+03 9 TEPS 2.40000E+03

② *** REGISTERED SIGNAL ***** MAXSIG = 8
1 PDW ON 2 ZRV ON 3 PRV OFF 4 IEP ON
6 IIV OFF 7 WEF OFF 8 SIV ON

③ << WEF SWITCHED ON >>
TIME 1.8000E+03 PDW 1.3147E+04 ZRV 5.0800E+00 PRV 3.1638E+05

```

Fig. 4.8 トランジェント計算結果の出力例

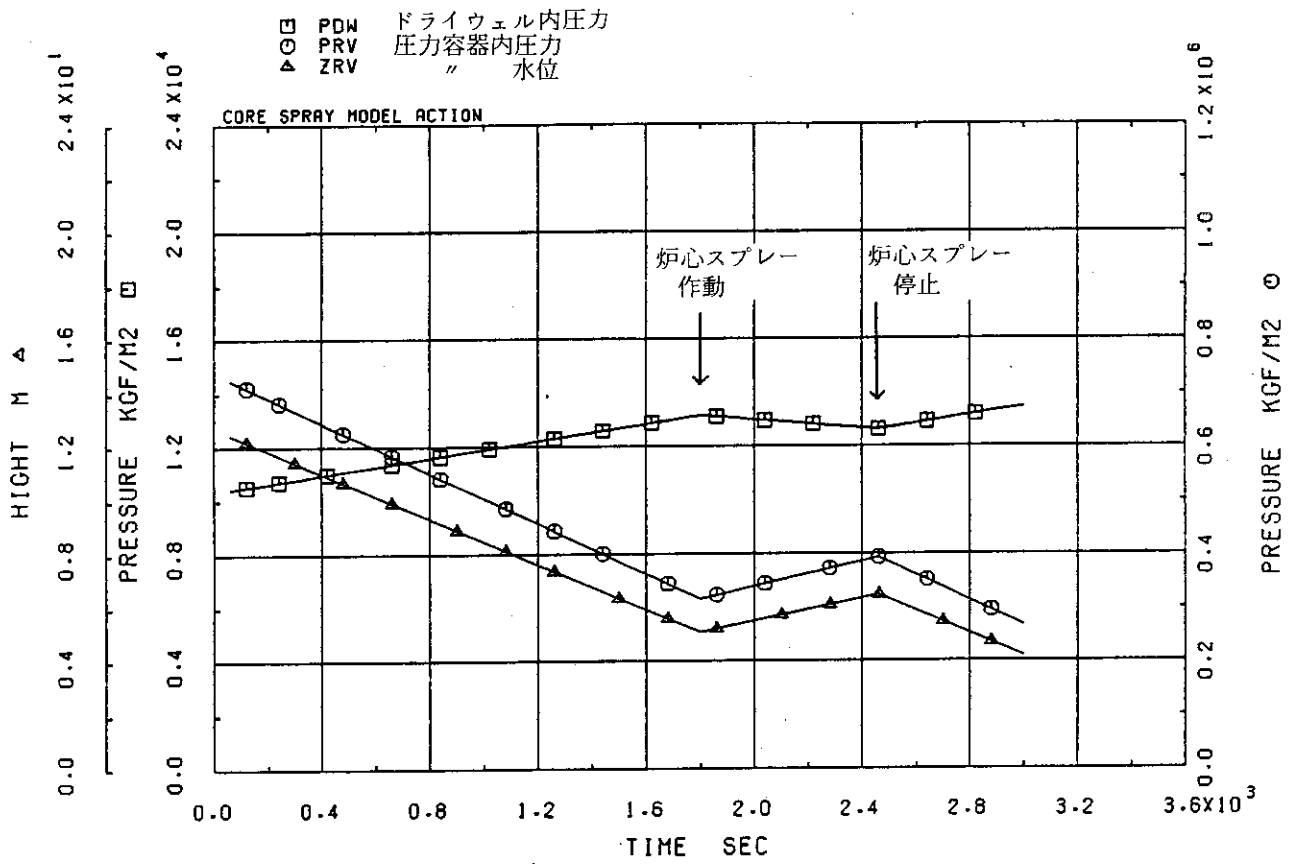


Fig. 4.9 計算結果のプロット例

5. お わ り に

本報告書で紹介したプログラム・ライブラリ LOGIC1 により、熱水力挙動解析コードで扱う各種動的機器の作動論理を標準的方法で模擬することが可能となった。原子力プラントの各種動的機器の作動条件等には、各プラントごとに多少の論理の違いや設定点の違いなどがあるが、共通のものとして考えられるところも多い。したがって、今後、非常用炉心冷却系の作動論理や、圧力逃し弁及び安全弁の開閉論理等を調査し、標準的なものをサブプログラムとして LOGIC1 内部に所有する予定である。なお、本報告書作成にあたって、リスク評価解析室長飛岡利明氏から多大なる御指導をいただいた。心から謝意を表したい。

参 考 文 献

- (1) 阿部清治他, " 炉心溶融事故時原子炉格納容器温度・圧力計算コード THALES-CV1 説明書", JAERI-M 83-037, Mar. 1983.
- (2) 阿部清治, " 汎用単位換算プログラム・ライブラリ: UCL2 の使用手引", JAERI-M 9592, July 1981.
- (3) 村松健他, " 過渡現象の実験結果及び計算結果の編集・作図用標準プログラムパッケージ SPLPACK-1 の使用手引", JAERI-M 83-166, Nov. 1983.

5. お わ り に

本報告書で紹介したプログラム・ライブラリ LOGIC1 により、熱水力挙動解析コードで扱う各種動的機器の作動論理を標準的方法で模擬することが可能となった。原子力プラントの各種動的機器の作動条件等には、各プラントごとに多少の論理の違いや設定点の違いなどがあるが、共通のものとして考えられるところも多い。したがって、今後、非常用炉心冷却系の作動論理や、圧力逃し弁及び安全弁の開閉論理等を調査し、標準的なものをサブプログラムとして LOGIC1 内部に所有する予定である。なお、本報告書作成にあたって、リスク評価解析室長飛岡利明氏から多大なる御指導をいただいた。心から謝意を表したい。

参 考 文 献

- (1) 阿部清治他, " 炉心溶融事故時原子炉格納容器温度・圧力計算コード THALES-CV1 説明書", JAERI-M 83-037, Mar. 1983.
- (2) 阿部清治, " 汎用単位換算プログラム・ライブラリ: UCL2 の使用手引", JAERI-M 9592, July 1981.
- (3) 村松健他, " 過渡現象の実験結果及び計算結果の編集・作図用標準プログラムパッケージ SPLPACK-1 の使用手引", JAERI-M 83-166, Nov. 1983.