

3章 CPU 制御セクション(CPU 制御部)

o 始めに

両 CPU もまったく同じで、独立した制御部はレジスタと命令発行と制御のための命令バッファを含んでいる。制御部は、命令の実行をプログラムからプログラムへと切り替えるために、交換メカニズムを使用する。これらのレジスタとバッファと交換メカニズム(機構)はこの章で説明される。メモリ・フィールド保護、プログラマブル・クロック、deadstart シーケンスも説明される。

o 命令発行と制御

命令発行と制御に関するレジスタと命令バッファを、これからの段落で説明する。図 3-1 は、レジスタとバッファを通した命令パーセルの全体の流れを描いている。

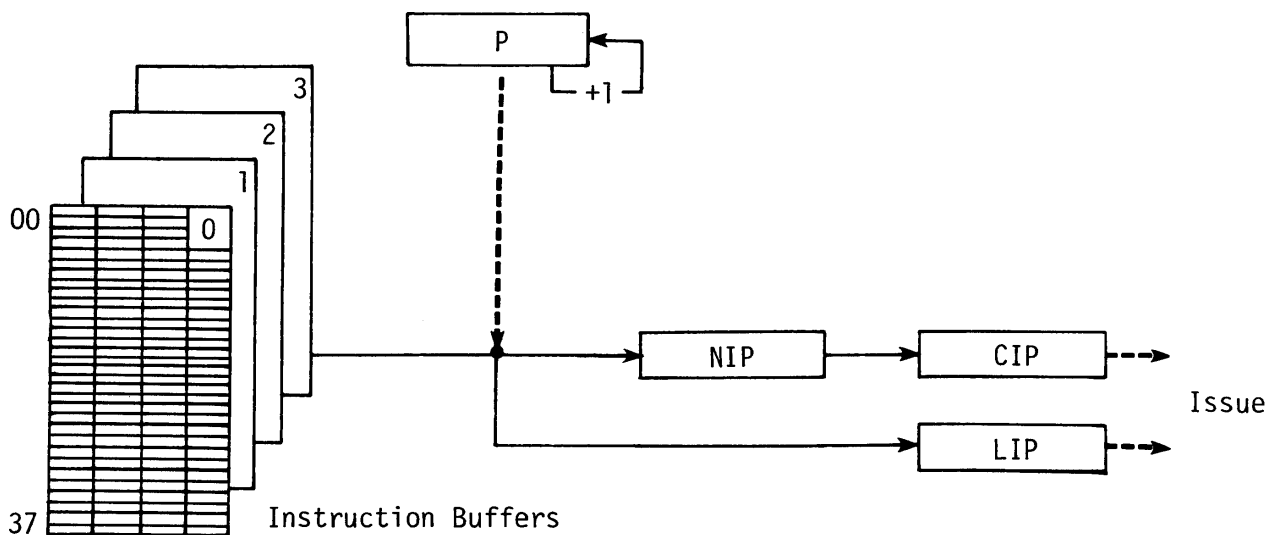


図 3-1. 命令発行と制御要素
(pp.3-1)

--

- プログラム・アドレスとレジスタ

24bit プログラム・アドレス(P)レジスタは、次命令パーセル(NIP; Next Instruction Parcel)レジスタに入るべき、プログラム・コードの次のパーセルを示す。Pレジスタの上位 22bit は、メモリ中のプログラム・ワードのワード・アドレスを示す。下位 2bit はワード中のパーセルを示す。分岐が取られた時の分岐命令と交換時を除いて、命令パーセルが NIP レジスタに入ったときに、P レジスタの内容は 1 進められる。

命令分岐か交換シーケンスの時、新しいデータが P レジスタに入る。(交換シーケンスはこの章の後方「交換機構」で説明される) P の内容は、次の分岐か交換シーケンスまで、順番に進められていく。P レジスタの値は、交換シーケンスの間、直接に終端交換パッケージ(terminating Exchange Package)に格納される。

P レジスタはマスタ・クリアされない。deadstart シーケンス中は、P に格納されている値は、正しくないだろう。

- 次命令パーセルレジスタ

16bit 次命令パーセル(NIP)レジスタは、現在命令パーセル(CIP; Current Instruction Parcel)レジスタに入る前の、プログラム・コードのパーセルを保持する。

NIP レジスタはマスタ・クリアされない。割り込み状態が NIP レジスタへのデータの登録をブロックする前に、未確定の命令は、マスタ・クリアの合間の間に、発行できる。(※訳注: 発行された命令が確定された命令だということだろう。「未確定の命令(undetermined instruction)」)

- 現在命令パーセルレジスタ(Current Instruction Parcel Register)

16bit の現在命令パーセル(CIP; Current Instruction Parcel)レジスタは発行を待っている命令を保持する。「発行(issue)」という言葉は CIP 中の命令が実行フェーズへ遷移することを意味する。

もし命令が 2 パーセル命令であれば、CIP レジスタは命令の最初のパーセルを保持し、下位命令パーセル(LIP: Lower Instruction Parcel)レジスタが 2 番目のパーセルを保持する。CIP 中の命令の発行は、競合中の操作が完了させられるまで、遅れさせられる。データは NIP から CIP に到着する。命令を作り上げる指示子(インジケータ)は、命令が発行した時、モード選択要求を持つすべてのモジュールに供給される。

CIP レジスタと結びついている制御フラグはマスタ・クリアされる、CIP レジスタそれ自体はされないが、未確定の命令はマスタ・クリア・シーケンス中も発行可能である。

- 下位命令パーセル(LIP: Lower Instruction Parcel)レジスタ

下位命令パーセル(LIP: Lower Instruction Parcel)レジスタは、2 パーセル命令の 2 番目のパーセルを保持する、2 パーセル命令の最初のパーセルが CIP レジスタに入る時に。

- 命令バッファ

CPUは4つの命令バッファを持つ。それぞれ128の連続した16bit命令パーセルを保持できる(図3-2)。命令パーセルは、NIPやLIPレジスタに届けられる前に、バッファ中に保持される。

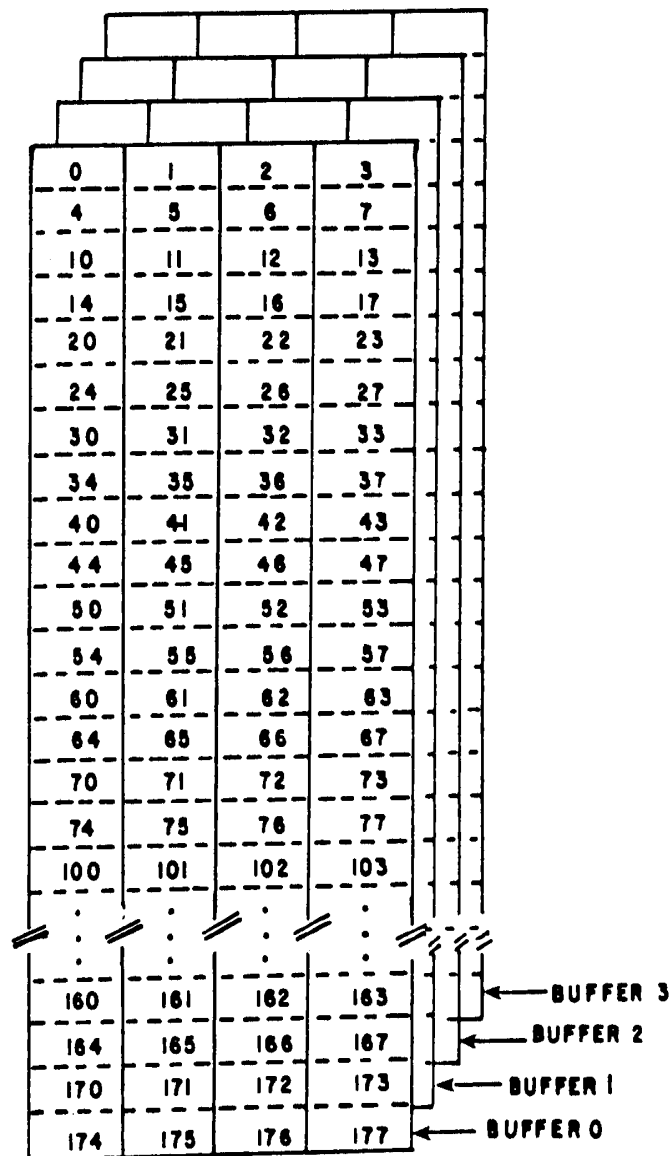


図3-2. 命令バッファ
(pp.3-3)

バッファ中での、命令パーセルの始まりは常にワード・アドレスを持つ。それは40(八進)の倍数であり(パーセル・アドレスは200(八進)の倍数である)、バッファ中の命令のためのアドレスの全体の範囲が、パーセル・アドレスの上位17bitで決定できるためにそのようにしてある。各バッファはこの値を収める17bitの開始アドレス(beginning address)・レジスタを持っている。

開始アドレス(beginning address)・レジスタは、各CPごとにスキャンされる。もしPレジスタの上位17bitが開

始アドレスのひとつと合致したら、**in-buffer**(バッファにあり)状態が存在し、適切な命令パーセルが命令バッファから選ばれる。実行されるべき命令パーセルは通常、**NIP** に送られる。しかし、**2** パーセル命令の **2** 番目のパーセルは、**NIP** レジスタへ入ることをブロックされ、その代わりに、**LIP** レジスタへ送られる。**2** パーセル命令の **2** 番目のパーセルは **1** 番目のパーセルが **CIP** レジスタから発行した時に、利用可能になる。その時同時に、全部 **0** のパーセルが **NIP** レジスタに入れられる。

in-buffer 状態で、もし、命令が、直前の命令と異なるバッファにあった場合、バッファの変更が発生し、命令が **NIP** レジスタに到達するまで **2CP** の遅延が要求される。

もし **P** レジスタの上位 **17bit** がどのバッファの開始アドレスとも合致しなければ、**out-of-buffer**(バッファの外)状態が存在する。この状態が発生した時、実行を続ける前に、命令は、メモリから命令バッファのひとつに読み込まなければならない。**2bit** のカウンタが、命令を受け取る命令バッファを決定する。各 **out-of-buffer** 状態は、バッファがローテーションで選ばれるように、カウンタの **1** 増加を起こす。

バッファはメモリから、完全にメモリを占有して **CP** あたり **8** ワードのレートで、読み込まれる。バッファへ届けられる **32** パーセルの最初のグループは、常に、実行に必要な次の命令を含んでいる。その理由は、分岐の **out-of-buffer** 時間はメモリがビジーでないとして、**32** バンクメモリで **16CP**、**16** バンクメモリで **18CP** である。(もし、メモリがビジーなら分岐は、ビジーが解消されるまで遅らされる)一度、フェッチが進行しだすと、残りのグループは、**CP** あたり **32** パーセルのレートで到着し、バッファは循環的に充填される。

命令バッファは、命令の **1** ワードを **32** の各メモリ・バンクから、または、**2** ワードを **16** の各バンクから、読み込む。命令バッファにある、最初の **4** 命令パーセルは、常にバンク **0** からである。交換シーケンスは命令バッファを無効にし、**P** レジスタと合致しないようにして、必要に応じて、バッファのロードが起きるようにする。

前方への分岐と後方への分岐はバッファの中で可能である。もし、分岐しようとしている命令のアドレスがバッファの一つの中にあった場合、分岐は命令バッファの再ロードを起こさない。命令バッファの中での、命令パーセルの多重コピーは起こらない。命令は発行前も後も、命令バッファの中に(そのバッファがリロードされるまで)保持されるため、自己書き換えコードは使用してはならない。また、データの独立性と命令メモリの保護のために、自己書き換えコードは不可能だろう。変更されていない命令のアドレスが、命令バッファ中にある間は、メモリ中で変更された命令は命令バッファへロードされない。

CPU をプログラムする時、命令バッファのためのコード・セグメント長の最適化が一番重要なのではないのだが、バッファの数とサイズと。前方と後方への分岐の能力は、良い有利さを出すために使われるだろう。**512** までの連続した命令パーセルを含む大きなループは、**4** つのバッファで維持できる。または、**1** つか **2** つのバッファの中のメイン・プログラム・シーケンスに、別なバッファ中で維持される短いサブルーチンを繰り返し呼び出させるというもある。**out-of-bffer** 状況にならない間か、交換がバッファの再ロードを引き起こさない限り、プログラムとサブルーチンは平穏にバッファ中にとどまる。

o 交換機構

CPU は命令実行をプログラムからプログラムへと切り替えるのに、交換機構を使用する。この交換機構は、交換パッケージとして知られるプログラム・パラメータのブロックの使用と、交換シーケンスとして参照される CPU 操作に、関係する交換パッケージの議論をする時には、Cray アセンブリ言語(CAL)プログラマの利便性のために、別なビット位置の表現が使用される。ビットは、左から右へ、0 から 2^{63} までビット位置に番号が付いている。

- 交換パッケージ

交換パッケージ(図.3-3)は、ある計算機プログラムに結びついた、16 ワードのメモリ中のデータのブロックである。交換パッケージは、プログラムの一つの実行期間から次への継続性を与えるのに、必要な基本パラメータを含んでいる。

交換パッケージの内容は、16 ワード・ブロックで並べられている。交換シーケンスは、データを、メモリから操作レジスタへ、またレジスタからメモリへ入れ替える。このシーケンスは、操作レジスタ上のアクティブな交換パッケージ内のデータを、メモリ上の非アクティブな交換パッケージと交換する。アクティブな交換パッケージの交換アドレス(XA: Exchange Address) レジスタは、入れ替えのために使用されるべきメモリのアドレスを指定する。データが交換され、新しいプログラム実行期間の開始は、交換パッケージによってなされる。

B,T,V,VM,SB,ST,SM レジスタの内容は交換シーケンスで入れ替えされない。これらのレジスタのデータは、目的プログラムの実行を管理するプログラムの特殊なコードか、このデータを必要とするいくつかのプログラムによって、必要に応じて格納と置き換えをされなければならない。(操作レジスタと VL レジスタの説明は 4 章を参照)

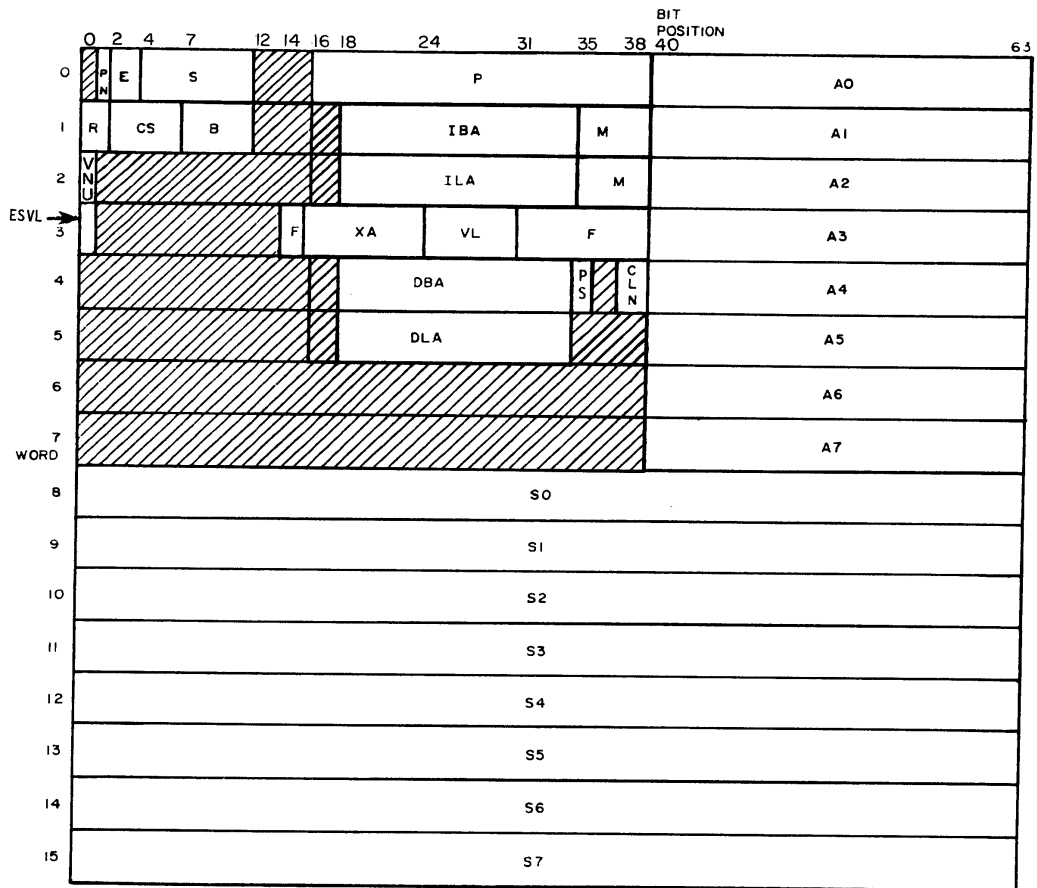


図 3-3. デュアル・プロセッサのための交換パッケージ (pp.3-6)

表 3-1. 交換パッケージの割り当て

Field	Word	Bits
Processor number (PN)	0	1
Error type (E)	0	2-3
Syndrome bits (S)	0	4-11
Program Address register (P)	0	16-39
Read mode (R)	1	0-1
Read address (CSB)	1	2-6 (CS) ;
	1	7-11 (B)
Instruction Base Address (IBA)	1	18-34
Instruction Limit Address (ILA)	2	18-34
Mode register (M)	1-2	35-39
Vector not used (VNU)	2	0
Enable Second Vector Logical (ESVL) [†]	3	0
Flag register (F)	3	14-15 ;
	3	31-39
Exchange Address register (XA)	3	16-23
Vector Length register (VL)	3	24-30
Data Base Address (DBA)	4	18-34
Program State (PS)	4	35
Cluster Number (CLN)	4	38-39
Data Limit Address (DLA)	5	18-34
Eight A register contents	0-7	40-63
Eight S register contents	8-15	0-63

(pp.3-7)

--

†: すべてのデュアル・プロセッサ・システムに存在しない

o プロセッサ番号

交換パッケージ内に位置する、プロセッサ番号(PN)の内容は、交換パッケージを実行した CPU を示す。この値は CPU に読み込まれず、格納されようとするパッケージのなかに挿入される定数である。

o ベクトル未使用(VNU: Vector not used)

交換パッケージ内に位置する、ベクトル未使用(VNU)は、実行期間の間に、命令 076, 077, 140 から 177 が発行されたか否かを示す。もし、命令が一つも発行していなければ、ビットはセットされる。もし、一つ以上の命令が発行していたら、ビットはセットされない。

o 第2ベクトル論理ユニット・イネーブル(ESVL: Enable Second Vector Logical) (+)

交換パッケージ内に位置する、第2ベクトル論理ユニット・イネーブル(ESVL)の内容は、第2ベクトル論理ユニットが使用可能かを示す。もしセットされていれば、命令140から145が第2ベクトル論理ユニットを選択する。もしクリアされていれば、第2ベクトル論理ユニットは使用できない;フル・ベクトル論理ユニットだけが使用できる。

(+: すべてのデュアル・プロセッサ・システムに存在しない)

o メモリ・エラー・データ

M(モード)レジスタ中の、bit36 (訂正可能メモリ・エラー・ビットでの割り込み) と、bit38 (訂正不可能メモリ・エラー・ビットでの割り込み) はメモリ・エラー・データが交換パッケージに含まれるかどうかを決定できる。エラー・データは、情報の4つのフィールドからなり、交換パッケージの中に、bit36がセットされていたら訂正可能メモリ・エラーに出会った、か、bit38がセットされていたら訂正不可能メモリ・エラーを検出した、として現れる。(++)
(++:複数ビットのメモリ・エラーの時、ハードウェアは常に、割り込み交換パッケージ中の訂正可能メモリ・エラー・フラグをセットする。)

メモリ・エラー・データ・フィールドを以下に説明する。

E(エラー型)

出会ったメモリ・エラーの型、訂正不可能か訂正可能は、交換パッケージのワード0のbit2とbit3で示される。bit2のセットは訂正不可能メモリ・エラー、bit3のセットは訂正可能メモリ・エラーである。

S(シンδροーム)

8つのシンδροーム・ビットはメモリ・データ・エラーを検出するのに使用され、交換パッケージのワード0のbit4から11を返される。詳しくは2章を参照。

R(リード・モード)

このフィールドは、メモリ・データ・エラーが起きたとき、進行中であったのが、リード・モードであることを示し、交換パッケージのワード1のbit0とbit1に位置する。これらのビットは、以下の値を表す:

00 I/O

01 スカラ(AかSのメモリ参照)

10 ベクトル,B,T

11 命令フェッチか交換

CSB(リード・アドレス)

10bitのCSBフィールドは、メモリ・データ・エラーが起きたアドレスを持つ。交換パッケージのワード1のbit7から11(B)までが、バンク・アドレスと考えられるアドレスの 2^4 から 2^0 を持つ。交換パッケージのワード1のbit2から6までが、アドレスの 2^{21} から 2^{17} を持つ。12カラムのメインフレームでは、これらのビットはアドレスのチップ・セレクト(CS)を表す;6カラムのメインフレームでは、上位3bitだけがチップ・セレクト(CS)と考えられる。

- 交換レジスタ

交換機構中に、3つの特殊レジスタが装備されている: 交換アドレス(XA)レジスタ、モード(M)レジスタ、フラグ(F)レジスタ。これら3つのレジスタは以下で説明する。

o 交換アドレス・レジスタ

8bitの交換アドレス(XA)レジスタは、交換操作によってロードされる、16ワードの交換パッケージの先頭ワードのアドレスを指定する。そのレジスタは、アドレスを指定する12bitフィールドの上位8bitを持つ。フィールドの下位ビットは常に0である; 交換パッケージは16ワード境界で始まらなければならない。12bitの限界はその絶対アドレスが、メモリの下位4096(10,000(八進))ワード内にあることを要求する。

実行期間が終了する時、交換シーケンスは、レジスタの内容と、開始アドレス(XA)レジスタで指されるメモリ中の交換パッケージの内容とを、交換する。

o モード・レジスタ

10bitモード(M)レジスタは、現在アクティブなプログラムのための交換パッケージの一部を持つ。Mレジスタ・ビットは、交換パッケージのワード1と2に、次のように、割り当てられている:

ワード1

Bit 記述

35 セマフォ待ち(WS: Waiting Semaphore)フラグ;

もしセットされていれば、テスト・アンド・セット命令がCIPレジスタ内でホールドしている時に、CPUが交換された。

36 浮動小数点エラー・ステータス (FPS: Floating-point Error Status)フラグ;

もしセットされていれば、浮動小数点エラー・モード・フラグの状態に関わらず、浮動小数点エラーが発生した。

37 双方向メモリ・モード(BDM: Bidirectional Memory Mode)フラグ;

もしセットされていれば、ブロック・リードとライト操作が並行してできる。

38 外部割込み選択(SEI: Selected for External Interrupts)フラグ;

もしセットされていれば、I/O割込みにCPUは供出された。

39 割込みモニタ・モード(IMM: Interrupt Monitor Mode)フラグ;

もしセットされていれば、モニタ・モードで、PC, MCU, I/O, 通常脱出以外のすべての割込みを許可する。

ワード2

35 オペランド範囲エラー・モード(IOR: Operand Range Error Mode)フラグ;
もしセットされていれば、オペランド範囲エラーでの割り込みを許可。

36 訂正可能メモリ・エラー・モード(ICM: Correctable Memory Error Mode)フラグ;
もしセットされていれば、「訂正可能メモリ・データ・エラー」での割り込みを許可。

37 浮動小数点エラー・モード(IFP: Floating-point Error Mode)フラグ;
もしセットされていれば、浮動小数点エラーでの割り込みを許可。

38 訂正不可能メモリ・エラー・モード(IUM: Uncorrectable Memory Error Mode)フラグ;
もしセットされていれば、訂正不可能メモリ・データ・エラーでの割り込みを許可。

39 モニタ・モード(MM)フラグ;
もしセットされていれば、メモリ・エラー以外のすべての割り込みを禁止。

交換シーケンス中に、10 個の bit が選択的にセットされる。

ワード 1, bit37 双方向メモリ・モード(BDM: Bidirectional Memory Mode)フラグ
は、命令 002600(双方向メモリ転送許可)と 002500(双方向メモリ転送禁止)を使うことによって、セットとクリアが可能。

ワード 2, bit35 オペランド範囲エラー・モード(IOR: Operand Range Error Mode)フラグ
は、プログラムの実行期間中に、命令 002300(オペランド範囲エラー割り込み許可)と、命令 002400(オペランド範囲エラー割り込み禁止)を使うことで、セットとクリアが可能。

ワード 2, bit 37 浮動小数点エラー・モード(IFP: Floating-point Error Mode)フラグ
は、プログラムの実行期間中に、命令 002100(浮動小数点エラー割り込み許可)と、命令 002200(浮動小数点エラー割り込み禁止)を使うことで、セットとクリアが可能。

ワード 1 の bit36,bit37 とワード 2 の bit35,bit37 は、命令 073i01 で読める。

ワード 1 の bit35,bit36 は実行時点の CPU の状態を示す。残りのビットは、交換パッケージのために実行期間の間には変更されない、そして、交換パッケージが記憶中で非アクティブの時にだけ変更することができる。

o フラグ・レジスタ

11bit のフラグ(F)レジスタは、現在、アクティブなプログラムのための交換パッケージの一部を持つ。このレジスタは交換パッケージ中のワード 3 にあり、各々識別される 11 個のフラグを持つ。これらのどのフラグへのセットもプログラムの実行に割り込みを起こさせる。一つ以上のフラグがセットされる時、割り込み要求(Request Interrupt) 信号が、交換シーケンスを開始するために送られる。Fレジスタの内容は、交換パッケージの残りと共に格納されている。モニタ・プログラムは、割り込みの要因のために、11 個のフラグを解析できる。モニタ・プログラムが、そのパッケージに交換し戻す前に、パッケージの Fレジスタの領域内のフラグをクリアしなければならない。もし、どれかのビットがセットされたままであると、別な交換が即座に発生する。

Fレジスタのビットは交換パッケージのワード 3 内に、次のように割り当てられている:

ワード 3

Bit 説明

14 内部 CPU からの割り込み(ICP: Interrupt From Internal CPU)フラグ;
他の CPU が命令 001401 を発行した時、セットされる。

15 デッドロック(DL)フラグ;
クラスタ中のすべての CPU がテスト・アンド・セット命令でホールド中の時、セットされる。

31 プログラム可能クロック割り込み(PCI: Programmable Clock Interrupt)フラグ;
プログラム可能クロック中の、「割り込みカウントダウン・カウンタ」が 0 になった時、セットされる。プログラム可能クロックはこの章で後述される。

32 MCU 割り込み(MCU)フラグ;
MIOP がこの信号を送った時、セットされる。

33 浮動小数点エラー(FPE)フラグ;
浮動小数点範囲エラーが、浮動小数点機能ユニットのどれかで発生し、かつ、「浮動小数点割り込み許可フラグ」がセットされている時、セットされる。浮動小数点機能ユニットは、4 章,計算で詳述される。

34 オペランド範囲エラー(ORE)フラグ;
「データ・ベース・アドレス(DBA: Data Base Address)」と「データ限界アドレス(DLA: Data Limit Address)」レジスタの境界の外部に、データ参照が成され、「オペランド範囲エラー割り込み許可フラグ」がセットされていた時、セットされる。オペランド範囲エラーはこの章で後述される。

35 プログラム範囲エラー(PRE)フラグ;
命令フェッチが「命令ベース・アドレス(IBA: Instruction Base Address)」と「命令限界アドレス(ILA: Instruction Limit Address)」レジスタの境界の外部に成された時、セットされる。プログラム範囲エラーはこの章で後述される。

36 メモリ・エラー(ME)フラグ;

訂正可能か訂正不可能メモリ・エラーが発生し、Mレジスタ内の、対応するメモリ・エラー・モード許可ビットがセットされている時、セットされる。

37 I/O 割り込み(IOI)フラグ;

6MByte/秒か 1250MByte/秒チャンネルの転送が完了した時にセットされる。

38 エラー終了(EEX: Error Exit)フラグ;

エラー終了命令(000)によってセットされる。

39 通常終了(NEX: Normal Exit)フラグ;

通常終了命令(004)によってセットされる。

Fレジスタ中のいずれかのフラグ(メモリ・エラー・フラグ以外)は、アクティブな交換パッケージがモニタ・モードに*無い*時だけ、セットできる。これらのフラグは、Mレジスタのワード2のbit39が0である時のみ、セットできる。メモリ・エラー・フラグを除いて、もしプログラムがモニタ・モードで、かつ、Fレジスタをセットする条件が存在しても、フラグはクリアされたままで、交換シーケンスは開始されない。

o クラスタ番号レジスタ

クラスタ番号(CLN)レジスタはCPUのクラスタを決める。CLNレジスタの内容は、SB,ST,SMレジスタのどの組にCPUがアクセスできるかを、決めるのに使用される。もし、CLNレジスタが0だと、CPUはどのSB,ST,SMレジスタにもアクセスできない。また、2つのCPUのCLNレジスタの内容は、デッドロック割り込みの必要な状態かを判断するのに、使用される。

o プログラム状態(Program State)レジスタ

1bitのプログラム状態(PS)レジスタの内容は、CPUが現在、処理中の一つのプログラムのプログラム状態が異なることを表現するために、オペレーティング・システムによって、操作される。

o Aレジスタ

全Aレジスタの現在の内容は、交換中に、ワード0から7のbit40から63に格納される。

o Sレジスタ

全Sレジスタの現在の内容は、交換中に、ワード8から15のbit0から63に格納される。

o プログラム・アドレス・レジスタ

プログラム・アドレス(P)レジスタ(まだ発行していないプログラム命令の最初のアドレス)の内容は、ワード0のbit16から39に格納される。この位置の命令は、このプログラムが再び始まる時に発行されるべき最初の命令である。

o メモリ・フィールド・レジスタ

各目的プログラムは、目的プログラムがロードされ、開始された時に、モニタ・プログラムによって指定された、命令とデータのためのメモリの割り当てられた領域(field)を持つ。目的プログラム・コード中に含まれている全メモリアドレスは、割り当てられたフィールドの始まり、と、限界を大きさで指定する2つのベース・アドレスの1つと関係している。各目的プログラムのメモリの参照は限界とベース・アドレスに対して、割り当てられた境界の中にアドレスがあるかどうかを判別することで、チェックされる。

これらのフィールドの限界は、交換パッケージにセーブされた4つのレジスタに含まれている。4つのレジスタは: 命令ベース・アドレス(IBA)レジスタ、命令限界アドレス(ILA)レジスタ、データ・ベース・アドレス(DBA)レジスタ、データ限界アドレス(DLA)レジスタである。このレジスタ群についての説明は、この章で後に出てくる「メモリ・フィールド保護」の節を参照。

- アクティブ交換パッケージ

アクティブ交換パッケージは、操作レジスタの中に居る。交換パッケージとそれに結びついているプログラムがアクティブである時の時間の区間が、実行期間(execution interval)と呼ばれる。実行期間は、目的とする交換パッケージをメモリから操作レジスタに移動するという交換シーケンスと共に始まる。実行期間は、引き続き交換シーケンス中で、交換パッケージがメモリに書き戻されると共に終了する。

- 交換シーケンス

交換シーケンスは、非アクティブな交換パッケージを、メモリから操作レジスタに運ぶための輸送手段である。同時に、交換シーケンスは、現在アクティブな交換パッケージを、操作レジスタからメモリへ運ぶ。この入れ替え操作は、固定されたシーケンスで、現在アクティブな交換パッケージと関連した、すべての計算アクティビティが停止した時に、終了する。メモリの16ワードの同じブロックが、非アクティブな交換パッケージの元としても、現在アクティブな交換パッケージの先としても、使われる。このブロックの位置は、XAレジスタの内容で指定され、それは現在アクティブな交換パッケージの一部である。交換シーケンスはdeadstartシーケンス、割り込みフラグのセット、プログラム終了によって開始される。

o deadstart シーケンスによる交換の開始

deadstart シーケンスは強制的に、両 CPU の XA レジスタの内容を 0 にし、1 つの CPU で割り込みを強制的に起こす。これらの 2 つの動作は、交換パッケージの位置をメモリの 0 番地として使用する交換を引き起こす。0 番地にある非アクティブな交換パッケージは、操作レジスタに移され、このパラメータを使用するプログラムが開始する。0 番地に入れ替え出された交換パッケージは、deadstart 操作のせいで、非常に不確かである。これらの格納アドレスに新しいデータが登録され、そして、CPU が引き続きプロセッサ間通信割り込みを開始するための、準備中に、古い交換パッケージを捨てる。

命令 001401(IP)が最初の CPU で発行した時、2 番目の CPU はメモリ中の番地 0 へ交換を行う。
(メインフレームの制御パネルのスイッチは、どちらの CPU が最初に deadstart されるかを選ぶ)

o 割り込みフラグセットによる交換の開始

F レジスタ中の割り込みフラグのどれか一つをセットすることで、交換シーケンスを開始させることができる。1 つ以上のフラグのセットは交換シーケンスを開始するための、割り込み要求信号を発生する。

o プログラム終了による交換の開始

2 つのプログラム終了命令が交換シーケンスを開始する。どちらの時でも、命令実行のタイミングは同じである；違いは、F レジスタ中の 2 つのフラグのどちらがセットされているかで、判断する。2 つの命令は：

000 ERR エラー終了

004 EX 通常終了

2 つの終了は、プログラムが、それ自身の終了の要求をできるようにしている。モニタでない(目的)プログラムは普段、モニタ・プログラムと交換するために、通常終了命令を使用する。エラー終了は目的プログラムの異常終了ができるようにする。選ばれた交換アドレスは、通常終了と同じである。

各命令は、F レジスタ中のフラグを持つ。現在アクティブな交換パッケージがモニタ・モードでなければ、適切なフラグがセットされる。このケースで、呼ばれる非アクティブな交換パッケージは、通常、モニタ・モードで実行しているものである。プログラムの終了が起こしたものを評価するために、フラグがチェックされる。

モニタ・プログラムは非アクティブな交換パッケージをアクティブにするために、非アクティブ交換パッケージのアドレスを、XA レジスタに設定し、そして、通常終了命令を実行することによって選ぶ。

o 交換シーケンス発行状態

以下は、ホールド発行状態、実行時間、そして、交換シーケンスの特別な場合である。

ホールド状態:

- NIPレジスタが正しい命令を持っている
- S,V,Aレジスタがビジー

実行時間:

- 32バンクでは、40CP; 交換シーケンス(24CP)とフェッチ操作(16CP)から成る
- 16バンクでは、42CP; 交換シーケンス(24CP)とフェッチ操作(18CP)から成る

特殊な場合:

もしテスト・アンド・セット命令がCIPレジスタ中でホールドされていれば、CIPとNIPの両レジスタがクリアされて、そして、WS(セマフォ待ち)フラグがセットされて、Pレジスタがテスト・アンド・セット命令を指して、交換が起きる。

- 交換パッケージ管理

各16ワードの交換パッケージは、システムdeadstart中に決めた領域の中にある。決められた領域は、メモリの下位の4096(10,000(八進))ワード内になければならない。0番地のパッケージはdeadstartモニタ・プログラムの交換パッケージである。他のパッケージは、目的プログラムとモニタ・タスクのために備える。モニタでないパッケージは、プログラムのための領域(フィールド)の長さの外にある。プログラムのフィールドの長さは、プログラムのベースと限界のアドレスで決定されているように表現している。モニタ・プログラムだけが、交換パッケージのエリアを含んだ、メモリのすべてをアクセスできるフィールドの定義を持っている。決められるフィールドは、モニタ・プログラムによって、定義されるか、現在アクティブな交換パッケージを除いた、すべての交換パッケージが変更される。あるCPU内の交換シーケンスと、別なCPU中のメモリ転送との間には、インターロックが存在しないので、ソフトウェアによって制御されている状況以外では、他のCPUに使用される交換パッケージの変更はやってはならない。

交換パッケージの適切な管理は、モニタでないプログラムが、交換してもらった、元のモニタ・プログラムに、常に、交換して戻ることを、命じる。交換は、プログラム情報が常に、その固有の交換パッケージに交換されることを保証する。例えば、pp.3-15 モニタ・プログラム(A)はdeadstartに引き続く実行期間を開始する。どの割り込みも(メモリを除いて)、モニタ・モードの間は、それを打ち切ることはできない。プログラムAは、通常終了命令(004)を発行して、任意に終了する。しかし、それを行う前に、プログラムAがXAレジスタの内容を、ユーザ・プログラム(B)の交換パッケージを指すようにセットしたら、プログラムBが次に実行されるプログラムである。プログラムAは、プログラムBの交換パッケージ内の交換アドレスを、プログラムAを指すようにセットする。

プログラムBへの交換シーケンスは、プログラムBの交換パッケージからの交換アドレスを、XAレジスタに入れる。同時に、XAレジスタ中の交換アドレスは、プログラムAのための他のプログラム・パラメータと一緒に、プログラムBの交換パッケージ領域に行く。交換が完了したとき、プログラムBは実行期間を開始する。

交換シーケンスを詳説するには、プログラムBが実行している間の、割り込みフラグが交換シーケンスの開始をセットする、と想定する。

プログラムBはXAレジスタを変更できないため、終了は、プログラムAに戻る。プログラムBのパラメータはそ

の交換パッケージ領域に戻れされる;実行期間の間プログラム B のパッケージ領域に保持されていたプログラム A のパラメータは、操作レジスタに交換して戻される。

プログラム A は、実行再開する時、適切な割り込み処理プロセッサを呼び出し実行させるべきかどうかのために、割り込みが、交換と XA レジスタのセットを起こしたかを判断する。これを行うために、プログラム A は、XA が割り込みプログラム(C)のための交換パッケージを指すようにセットする。プログラム A は割り込みをクリアして、通常終了命令(004)を実行することによって、プログラム C の実行を開始する。実行する仕事によって、プログラム C はモニタ・モードやユーザ・モードで実行することができる。

交換パッケージ管理についてのより詳しい情報は、「COS EXEC/STP/CSP 内部リファレンス・マニュアル」,SM-0040 に含まれている。

o メモリ領域保護

実行時に目的プログラムは、命令とデータのために、メモリの指定された領域(field)を持つ。領域の限界は、プログラムがロードされ、開始される時に、モニタ・プログラムによって指定される。領域は、32(40(八進))の倍数のいかなるワード・アドレスからでも始められ、1 以上の 32(40(八進))の倍数の連続した別なアドレスまでである。

目的プログラム・コード中の全てのメモリ・アドレスは、領域を指定する 2 つのベース・アドレスのうちの一つと相対的である。目的プログラムは、ベース・アドレスより低い絶対アドレスのメモリ番地を読んだり変更することはできない。各目的プログラムのメモリ参照は、アドレスが設定された境界の中か否か、ベース・アドレスと限界に対してチェックされる。設定された領域の限界を超えたメモリ読み出し参照は、発行され、完了する、が、メモリからは 0 が転送される。設定された領域の限界を超えたメモリ書き込み参照は発行は許される、が、書き込みは起こらない。

領域の限界は 4 つのレジスタに含まれている: 命令ベース・アドレス(IBA)レジスタ、命令限界アドレス(ILA)レジスタ、データ・ベース・アドレス(DBA)レジスタ、データ限界アドレス(DLA)レジスタである。これら領域の限界と結び付けられた 4 つのレジスタとフラグは以降の段落で説明される。

- 命令ベース・アドレス(IBA)レジスタ

命令ベース・アドレス(IBA)レジスタは、ユーザ命令領域のベース・アドレスを保持する。命令が置かれた位置の絶対アドレスが、プログラム実行中の現在の交換パッケージの IBA レジスタの内容以上であるときだけ、命令は、CPU によって実行される。この判別は、CPU による命令バッファへのフェッチ時に行われる。

IBA レジスタの内容は、22bit メモリ・アドレスの上位 17bit が解釈される。アドレスの下位 5bit は、32 個のバンクのバンク番号なので、0 として解釈される。命令フェッチのための絶対メモリ・アドレスは、IBA レジスタを P レジスタ(上位 22bit)に加えて、2 の 22 乗の剰余をとり、形作られる。

IBA で定義されたアドレスより小さい絶対アドレスへの参照は、機械のメモリ容量を超えたアドレスへのジャンプか分岐命令を通してしかありえない。

- 命令限界アドレス(ILA)レジスタ

命令限界アドレス(ILA)レジスタはユーザ領域の限界アドレスを保持する。命令が置かれた位置の絶対アドレ

すが、現在の交換パッケージのプログラム実行の ILA レジスタの内容より小さいときだけ、命令は、CPU によって実行される。この判別は、CPU による命令バッファへのフェッチ時に行われる。

ILA レジスタの内容は、22bit メモリ・アドレスの上位 17bit が解釈される。アドレスの下位 5bit は、32 個のバンクのバンク番号なので、0 として解釈される。プログラムによって実行可能な最大の絶対アドレスは、 $[(ILA) \times 2^5] - 1$ として定義される。

もし、CPU によって計算された命令バッファ・フェッチの最終の絶対アドレスが、現在の実行中の交換パッケージの IBA と ILA レジスタに入っているアドレスの範囲に落ちなかった時、CPU はプログラム範囲エラー割り込みを生成する。

- データ・ベース・アドレス(DBA)レジスタ

データ・ベース・アドレス(DBA)レジスタは、ユーザのデータ領域のベース・アドレスを保持する。オペランドが置かれた位置の絶対アドレスが、プログラム実行中の現在の交換パッケージの DBA レジスタの内容以上であるときだけ、オペランドは、CPU によってフェッチか格納される。この判別は、CPU によるフェッチか格納時に、毎回、行われる。

DBA レジスタの内容は、22bit メモリ・アドレスの上位 17bit が解釈される。DBA レジスタの下位 5bit は 0 として解釈される。命令フェッチのための絶対メモリ・アドレスは、変更されるオペランドのアドレスに DBA レジスタを加えて、2 の 22 乗の剰余をとって形作られる。

- データ限界アドレス(DLA)レジスタ

データ限界アドレス(DLA)レジスタは、ユーザのデータ領域の上限のアドレスを保持する。命令が置かれた位置の絶対アドレスが、実行中の現在の交換パッケージの DLA レジスタの内容より小さいときだけ、オペランドは、CPU によってフェッチか格納される。この判別は、CPU によるフェッチか格納時に、毎回、行われる。

DLA レジスタの内容は、22bit メモリ・アドレスの上位 17bit が解釈される。DLA レジスタの下位 5bit は、0 として解釈される。プログラムによって可能なデータの参照の最大の絶対アドレスは、 $[(DLA) \times 2^5] - 1$ として定義される。

もし、CPU によって計算されたオペランドの最終の絶対アドレスが、現在の実行中の交換パッケージの DBA と DLA レジスタに入っているアドレスの範囲に落ちなかった時、CPU はオペランド(アドレス)範囲エラー割り込みを生成する。

- プログラム範囲エラー

プログラム範囲エラー・フラグは、命令フェッチのためのメモリ参照が、IBA と ILA レジスタの境界の外側であった時に、セットされる。範囲外(out-of-range)メモリ参照は、非モニタ・モード・プログラムでの、限界の下か上のプログラム・アドレスへの呼び出しを行う、分岐かジャンプ命令で、発生する。プログラム範囲エラー・フラグは、プログラム実行を終わらせるエラー状態を引き起こす。モニタ・プログラムは、プログラム範囲エラー・フラグの状態をチェックし、適切なアクションを行う、たぶん、ユーザ・プログラムを中止する。

- オペランド範囲エラー

オペランド範囲エラー・フラグは、オペランド範囲エラー・モード(**Operand Range Error Mode**)フラグがセットされており、かつ、**A,B,S,T,V** レジスタのためのオペランドの読み書きのメモリ参照が、**DBA** と **DLA** レジスタの境界の外側であった時、かつ、オペランド範囲割り込みエラー(**Operand Range Interrupt Error**)フラグがセットされている時、セットされる。オペランド範囲エラー・フラグは、ユーザ・プログラム実行を終わらせるエラー状態を引き起こす。モニタ・プログラムは、オペランド範囲エラー・フラグの状態をチェックし、適切なアクションを行う、たぶん、ユーザ・プログラムを中止する。

o プログラム可能クロック

プログラム可能クロックは、期間の時間の正確な計測に使用することができる。モニタ・プログラム制御の元で選ばれた期間は、周期的な割り込みを生成する。クロック周波数は **105MHz** である。期間は、**9.5** ナノ秒から、おおよそ **40.8** 秒まで可能である。**100** マイクロ秒より短い期間は、割り込み処理に関するモニタのオーバヘッドから、滋養的ではない。割り込み間隔(**II: Interrupt Interval**)レジスタ、割り込みカウントダウン(**ICD**)カウンタ、**4** つのモニタ・モード命令が、プログラム可能クロックをサポートする。

- 命令

4 つのモニタ・モード命令が、プログラム可能クロックをサポートする:

0014j4 PCI Sj

(Sj)を割り込み間隔(**II**)レジスタに入れる

001405 CCI

プログラム可能クロック割り込み要求をクリアする

001406 ECI

プログラム可能クロック割り込み要求を許可する

001407 DCI

プログラム可能クロック割り込み要求を禁止する

- 割り込み間隔(**II**)レジスタ

32bit の割り込み間隔(**II**)レジスタは、**CP** の数と同じ二進数の値をロードでき、その **CP** だけ経過したら、プログラム可能クロック割り込み要求になる。割り込み間隔は、命令 **0014j4** が実行された時に、**Sj** レジスタの下位 **32bit** から、**II** レジスタと **ICD** カウンタに転送される。この値は、**II** レジスタに保持され、そして、**ICD** カウンタが **0** になる毎に、カウンタへ転送され、割り込み要求が生成される。**II** レジスタの内容は、命令 **0014j4** でしか変更できない。

- 割り込みカウントダウン・カウンタ

32bit の割り込みカウントダウン(**ICD**)カウンタは、命令 **0014j4** が実行された時に、**II** レジスタの内容にプリセット

トされる。このカウンタは、連続的にカウントダウンを行う。そのカウントダウンは、カウンタのカウントが **0** になるまで、各 **CP** 毎に **1** のデクリメントを行う。**ICD** はプログラム可能クロック割り込み要求をセットし、**II** レジスタに保持されている間隔値を取り込む。**ICD** はカウントダウンが **0** になるサイクルを繰り返す。規則的な間隔のプログラム可能クロック割り込み要求の設定は、間隔値(inteval value)によって決めまる。

プログラム可能クロック割り込み要求がセットされた時、プログラム可能クロック割り込み要求のクリアが実行されるまで、割り込み要求は記憶される。プログラム可能クロック割り込み要求は、プログラム可能クロック割り込み要求許可が実行された後だけ、セット可能である。プログラム可能クロック割り込み要求は、モニタ・モードで無い時だけ割り込みを起こす。モニタ・モードでの要求はシステムがユーザ・モードに切り替わるまで保持される。

- プログラム可能クロック割り込み要求クリア

プログラム割り込み間隔に続いて、アクティブなプログラム可能クロック割り込みの状態は、命令 **001405** か **001407** の発行によって、クリアされる。

o 性能モニタ

システムは、相対的な性能を示すために使用できる、ハードウェアに関するイベントを追跡ための、**8** つの性能カウンタの組の一つ持っている。追跡できるイベントは、命令 **0015j0** を通して選択された、いくつかの指定した命令の発行、発行ホールド状態、フェッチの数、参照の数などである。

o deadstart シーケンス

運用の **deadstart** シーケンスは、電源が切断され、再び投入された後か、オペレーティング・システムがメインフレームの再初期化を行った時に、メインフレーム内でプログラムの走行を開始する。機械内の全レジスタ、全部の制御ラッチ、とすべてのメモリ中の全ワードは、電源投入後は、無効(**invalid**)と考えるべきである。プログラムを開始する操作の引き続くシーケンスは、**I/O** サブシステムによって初期化される。

1. マスタ・クリア信号を **ON** にする。
2. **I/O** クリア信号を **ON** にする。
3. **I/O** クリア信号を **OFF** にする。
4. **I/O** サブシステムを通してメモリをロードする。
5. マスタ・クリア信号を **OFF** にする。

マスタ・クリア信号は、内部のすべての計算を止め、重要な制御ラッチをあらかじめ定義された状態にする。**I/O** クリア信号は、**MCU** チャンネルの入力チャンネル・アドレス・レジスタをクリアし、**MCU** 入力チャンネルのをアクティベートする。他の全入力チャンネルは非アクティブにされる。**I/O** サブシステムは、初期の交換パッケージとモニタ・プログラムをロードする。交換パッケージはメモリの **0** 番地に位置しなければならない。マスタ・クリア信号を **OFF** にすることは、この交換パッケージを読み、**CPU0(PN=0)** でモニタ・プログラムの実行を開始するという、交換シーケンスを開始する。

CPU1(PN=1)は、命令 001401(IP)が CPU0 で発行されるまで、マスタ・クリア状態をつづける。

CPU0 の交換が非アクティブな 0 番地の交換パッケージの内容を上書きするので、CPU0 は、他の CPU をスタートを許可する前に、0 番地の交換パッケージを再初期化しなければならない。(メインフレームの制御パネルのスイッチでどちらの CPU でも最初にスタートさせることができる) 引き続き動作は、オペレーティング・システムの設計によって支配される。

-- EOF