

IBM S/360-20 におけるマイクロプログラミング技術の実際

竹内 廣 文†

1. S/360-20 概説

S/360-20 で使われるデータ、命令の形式を Tab. 1 に示す。CPU は、最大 16k バイトの Main Storage の他に、8つの半語長 (2バイト) の General Register を持っている。この指定は命令 (インストラクション) 中の R および B フィールドの 8~15 によりなされる。Register 間、Register-Storage 間の動作 (オペレーション) は半語単位で行なわれる。命令の長さを示すビットは op コードのビット 0, 1 で、次のように解釈される。

ビット 0,1	命令の形式	命令の長さ
00	RR	2 バイト
01	RX	4 バイト
10	SI	4 バイト
11	SS	6 バイト

入出力動作については、入出力機器とのデータの転送は SS 形式の XIO により、各入力機器の制御、条件 (コンディション) のチェックなどはそれぞれ CIO,

TIOB の命令により行なわれる。命令中の DA, FS はそれぞれ Device Address, Function Specification で、入出力機器の指定、行なわれる動作のタイプを示す。

2. データフロー

Fig. 1 に示すように、CPU は Main/Auxiliary Storage およびそのアドレス回路 (STAR, STAR DECODER, AUX, STR, CTL), ROS およびその周辺回路 (ROSDR, ROAR, DECODER), A および DR (E, S, T, R, U, L, P, I), および Modifier より構成されている。CPU は ALU を持たず、算術、論理演算はマイクロプログラムで処理される。

2.1 Main/Auxiliary Storage

Main/Auxiliary Storage のアドレスは STAR の内容をデコードすることにより決まる。Main/Auxiliary の区別は、マイクロ命令による AUX, STR, CTL を通して行なわれる。Storage の各ポジションは 1 バイトの情報を貯えておくことができる。Auxiliary

FORMAT and APPLICATION	Bin																																														
	Byte 0								Byte 1								Byte 2								Byte 3								Byte 2n								Byte 2n-1						
	Half-word																Half-word																														
DATA FORMATS																																															
Binary Halfword	S Binary Number																																														
Packed Decimal Number (maximum 16 bytes)	0 1 Digit Digit Digit Digit Digit Digit Digit Digit Digit Digit Digit Digit Digit Digit Digit																																														
Zoned Decimal Number (unpacked) (maximum 16 bytes)	0 Zone Digit Zone Digit Zone Digit Zone Digit Zone Digit Zone Digit Zone Digit Zone Digit																																														
Fixed Length Logical Information (1 byte) Exception: SP5W contains a fixed operand length of 4 bytes	0 Single Character																																														
Variable Length Logical Information (maximum 256 bytes)	0 Character Character Character Character Character Character																																														
INSTRUCTION FORMATS																																															
RR - Format (Register - Register) AR, SR, BASR, BCR	0 OP Code R1/M1 R2																																														
RX - Format (Register - Storage) STH, LH, CH, AH, SH, BAS, BC	0 OP Code R1/M1 0 0 0 0 B2 D2																																														
SI - Format (Storage - Immediate Data) MVI, NI, CLI, OI, TM, SP5W, HPR (DI, BI) may be used as Stop - Number	0 OP Code I2/M2/Zero R1 D1																																														
TIOB, CIO (DI, BI) is used as Detailed Specification	0 OP Code DA FS R1 D1																																														
SS - Format (Storage - Storage) MVN, MVC, MVZ, CLC, ED, TR MVO, PACK, UNPK, ZAP, CP, AP, SP, MP, OP	0 OP Code Length - Code R1 D1 R2 D2																																														
XIO (DI, BI) = Storage Address, D2(BI) = Length of Area	0 OP Code DA FS R1 D1 R2 D2																																														


Tab. 1 Data & Instruction Format Summary

† 日本アイ・ビー・エム (株) 藤沢工場 計画ニューテクノロジー

liary Storage は 256 バイトのポジションを持っており、そのうち 8×2 バイト分は、General Register としてマイクロ命令で直接使用される (Fig. 4 参照)。他のポジションはマイクロプログラムの一時記憶として、種々の目的のために使われる。Storage への情報の出し入れは DR (Data Register)-U, L を通して行なわれる。

2.2 ROS および周辺回路

ROS Word は 60 ビットより成り、3 つのマイクロ命令 A (16 ビット), B, C (22 ビット) が貯えられる。ROAR (Read Only Address Register) で指定されるアドレスに従って ROS Word が選ばれ、次に A, B, C のうちの 1 つのマイクロ命令が選ばれ ROSDR (ROS Data Register) へセットされる。ROSDR にセットされたマイクロ命令のインストラクション・ポーションであるビット 1~9 がデコードされ、CPU データフローを制御し、そのマイクロ命令が実行される。この間マイクロ命令の NSI (Next Sequential Instruction) アドレス・ポーション、つまりビット 10~21 に従い、次のマイクロ命令の入っている ROS のアドレスを指定する数が ROAR にセットされる。ROSDR 中のマイクロ命令 (実行中のもの) が後述する USE または INCREMENT/DECREMENT のときは、USE オペレーションの結果、INC/DECR のときはキャリーが発生するか否かにより、ROSDR の 10~21 ビットを ROAR に移すときに、18~21 ビットまたは 21 ビットの値が変わり、その結果が ROAR にセットされる。

Fig. 1 において、各マイクロ命令がどのようにデータフローに関係するかを  で示した。

2.3 Data Register

E, S, T, R, U, L, P, I の 8 つのハーフバイト (4 ビット+P ビット) の DR が用意されている。Tab. 2 のマイクロ命令中の 3 ビットの X, X* ポーションによりこの 8 つの DR がアドレスできるように、各 DR に 0~7 のアドレスが付けられている。

これらの他に、Main Storage のアドレス生成のために、Extension-Bit として 2 ビットの Register A がある。また、この A と DR-S, T, R には Sub-Register があり、マイクロ命令 CTL-15 により A, S, T, R とこの Sub-Register の内容を交換することができる。

DR はマイクロプログラム中で種々の目的のために使われる。さらに、DR-U, L は Storage と直結して

Tab. 2 Micro Instructions Summary

C A S Repr	ROSDR Bits																					
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
X → X	P	0	1	X*	X*	X*	0	X	X	X	← NSI Address →											
N → X	P	1	1	X*	X*	X*	N	N	N	N	← NSI Address →											
UOX	P	0	0	0	1	1	1	X	X	X	← NSI Address → Mask											
UAX	P	0	0	0	1	0	1	X	X	X	← NSI Address → Mask											
UXX	P	0	0	0	0	1	1	X	X	X	← NSI Address → Mask											
X + 1	P	0	0	0	1	0	0	X	X	X	← NSI Address → Carry											
X - 1	P	0	0	0	0	1	0	X	X	X	← NSI Address → Carry											
FN	P	1	0	0	1	Y	N	N	N	N	← NSI Address →											
FPN	P	1	0	1	1	Y	N	N	N	N	← NSI Address →											
FSTR	P	0	0	1	1	Y	0	0	1	1	← NSI Address →											
SN	P	1	0	0	0	Y	N	N	N	N	← NSI Address →											
SPN	P	1	0	1	0	Y	N	N	N	N	← NSI Address →											
SSTR	P	0	0	1	0	Y	0	0	1	1	← NSI Address →											
SNS n	P	0	1	n	n	n	1	n	n	n	← NSI Address →											
CTL n	P	0	0	1	n	n	1	n	n	n	← NSI Address →											
CTL n	P	0	0	0	0	0	1	n	n	n	← NSI Address →											
CTL n	P	0	0	0	1	1	0	n	n	n	← NSI Address →											
X 0-7				X*	X*	X*		X	X	X												
N 0-15								N	N	N	N	N	N	N	N	N	N	N	N	N	N	N
n 0-63					n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n	n
P 0-1								Y	0 = Long Cycle 1 = Short cycle													

おり、Main/Auxiliary Storage への読み書きの際に使用され、DR-S, T, R は A とともに Storage のアドレス生成のために使用される。DR-P は Auxiliary Storage のアドレス生成に使われる。通常、DR-E, I はオペレータへのインディケータとして使用される。

3. マイクロ命令の説明

Tab. 2 はマイクロ命令の一覧表である。マイクロ命令は 22 ビットより成り (16 ビットのものここではふれない)、ビット 0 は P チェックであり、ビット 1~9 でインストラクションを決める。ビット 10~21 は NSI アドレスとして使用される。Fig. 1 のデータフローを参照しながら各マイクロ命令を解説する。

3.1 X→X Move (MX=X)

ROSDR のビット 7~9 の 3 ビット X で指定される DR (E, S, T, R, U, L, P, I) の内容を、ビット 3~5 の 3 ビット X* で指定される DR へ移すマイクロ命令である。X で指定された DR の内容は Bus に入り、Modifier (ここでは何の変化もうけない) を通

って X* で指定された DR に入る。データの変化はない。

例 X=100, X*=100 の場合, DR-U の内容を DR-E に移す。[U~E] で表わす。ROSDR のビット 10~21 で指定されたものが, NSI アドレスとして ROAR にセットされる。

3.2 N→X Move (MX=N)

マイクロ命令中のビット 6~9 の 4 ビットの N を X で指定された DR (4 ビット) へ挿入する。N は 0~15 の任意の数である。ROSDR のビット 6~9 は Bus を通り, X* で指定された DR に入る。

例 N=1001, X*=011 の場合, DR-R の内容は 1001 になる。[9→R] で表わす。NSI アドレスはビット 10~21 で指定されたものが使われる。

3.3 UOX, UAX, UXX, USE

USE オペレーションと呼ばれ, マイクロプログラム中の種々の意志決定に使われるマイクロ命令である。DR の内容に従ってマイクロプログラムの分岐先を決めるもので, DR の任意のビットポジションの判定ができる。マイクロ命令中のマスクビットであるビット 18~21 と X で指定される DR の内容との論理オペレーションの結果が NSI アドレスの最後の 4 ビット (18~21) を形成し ROAR にセットされることにより, 分岐が行なわれる。論理オペレーションは UOX, UAX, UXX に対し, それぞれ OR, AND, EXOR が使われる。

例 DR-L の L2, L8 を判定する場合

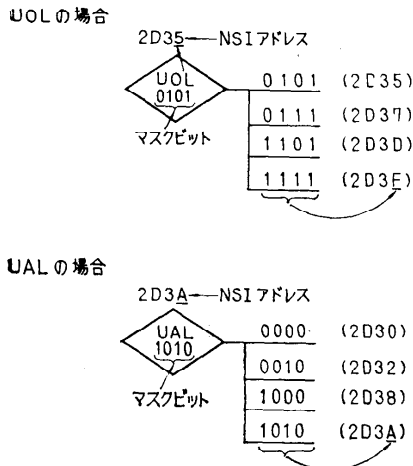


Fig. 2 Use of UOX and UAX

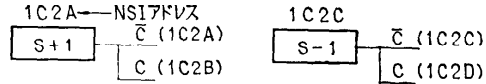


Fig. 3 Use of X+1 and X-1

UOL と UAL を用いたときを Fig. 3 に示す。NSI アドレスに関しては, 最後の 16 進数 (0~F) についてのみ注目すること。

3.4 X+1, X-1 INCREMENT/DECREMENT (INCRX, DECRX)

X で指定された DR の内容を +1, または -1 するマイクロ命令である。X で指定される DR の内容は Bus を通り Modifier を通過するときに +1, または -1 されてもとの DR にもどる。DR の内容がすでに 15 のとき (DECREMENT のときは 0) はキャリーが出て NSI アドレス最後のビット 21 を 1 にして ROAR にセットする。このマイクロ命令のビット 21 は 0 でなければならない。X がアドレス 1 のときの例を Fig. 3 に示す。

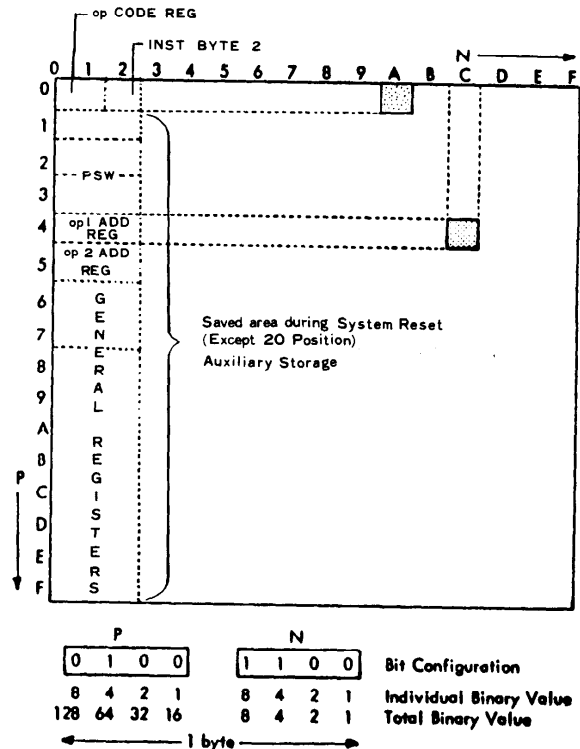


Fig. 4 Auxiliary Storage

3.5 FN/SN, FPN/SPN Fetch/Store Auxiliary Storage

256 バイトの Auxiliary Storage のアドレッシングは、DR-P およびマイクロ命令中のビット6~9のNのマトリックスにより Fig. 4 のようにして行なわれる。最上段の0~Fのアドレス領域はマイクロプログラムにより頻繁に使用されるもので、DR-P の値に関係なくアドレスできるように FN/SN マイクロ命令が用意されている。通常は、あらかじめ DR-P に必要な数値をセットしておいて、FPN/SPN マイクロ命令を使用する。ビット5のYについてはここではふれない。Fetch/Store の情報は DR-U, L を通して行なわれる。

3.6 F-STR/S-STR Fetch/Store Main Storage

予め Main Storage のアドレス生成のために A, S, T, R を続いた Register と考え、必要なアドレスをセットしておく。F-STR/S-STR のマイクロ命令により A, S, T, R の内容が STAR にセットされ、指定されたロケーションの Storage がアクセスされる。2ビットの Extension-Bit A のコントロールには、DR-U の1, 2ビット U1, U2 が使用される。

$L \rightarrow S$ を行なうと、DR-U ビット1, 2がそれぞれ4096, 8192を表わすAにセットされる。また $S \rightarrow L$ を行なうとAの2ビットがDR-Uのビット1, 2にセットされる。後で述べる Control マイクロ命令 (CTL-5, Set Auxiliary Storage) と F-STR/S-STR により Auxiliary Storage もアクセスが可能である。このとき、Main Storage のアドレス指定にもどるためには、CTL-6(Reset Auxiliary Storage) のオペレーションが必要である。Fetch/Store の情報はDR-U, Lを通る。

3.7 SNS-n Sense-n

マイクロプログラムでは、CPU 内部の種々の条件、コンソル上のスイッチの位置、入出力機器から読まれてくるデータなどを知る必要がある。この目的のためのマイクロ命令が SNS-n である。1つの SNS-n マイクロ命令によって4つのビットが読み込まれ、DR-L または U にセットされる。n は0~63の64通りである。Tab. 3 に DR-L にセットされるいくつかの Sense テーブルを示す。SNS-0 ではコンソル上の Mode Switch が読み取られ Switch の状態に従ったビットが DR-L にセットされる。SNS-2, 3, 4, 5, 6, 7 では、Storage Alter のようなマニュアル操作で、ストアすべきアドレス、データを指定するために

Tab. 3 Sense Table

DR - L			
L 8	L 4	L 2	L 1
MODE SWITCH			
0	STOR SCAN	DISPL	ALTER REG
ADDR/REG DATA SWITCH 1 (LEFT)			
2	BIT 8	BIT 4	BIT 2 BIT 1
ADDR/REG DATA SWITCH 2			
3	BIT 8	BIT 4	BIT 2 BIT 1
ADDR/REG DATA SWITCH 3			
4	BIT 8	BIT 4	BIT 2 BIT 1
ADDR/REG DATA SWITCH 4 (RIGHT)			
5	BIT 8	BIT 4	BIT 2 BIT 1
DATA SWITCH 1 (LEFT)			
6	BIT 8	BIT 4	BIT 2 BIT 1
DATA SWITCH 2 (RIGHT)			
7	BIT 8	BIT 4	BIT 2 BIT 1
2520/2560 CD DATA			
12	4	5	6 7
2520/2560 CD DATA			
13	12	11	0 9
2520/2560 CD DATA			
14	8	1	2 3
MODE SWITCH			
15	LOAD OR NOT IS OR STOP	STOP COND	SINGLE INST ADDR STOP
2560 MFCM			
29	CAKD IN PRE READ	HOPPER CHCK	CD IN PRT CARD IN PFE PCH

使用されるコンソル上のロータリスイッチの位置を(0~F)読み取りDR-Lの対応したビットにセットする。SNS-12, 13, 14 はカード入出力機器から読まれてくるカラムのデータを示す。SNS-29の結果L1が1のとき、カードがカード入出力機器の中でパンチされるために準備された位置についていることを示し、マイクロプログラムでその状態を知ることができる。ほとんどの SNS は入出力オペレーションで使用されるので、他の説明は省略する。

3.8 CTL-n Control-n

CPU 内のクロックの停止、インディケータのセット、リセット、入出力オペレーションの開始などの種類の制御に用いられる。通常は、DR-L(U, S, T, I)の各ビットとこの CTL-n の組合せによりオペレーションが決まる。DR の内容に関係なくコントロールするものもいくつかあり、代表的なものは CTL-0 で、Process Latch をリセットし CPU を停止させる。F-STR/S-STR で述べた CTL-5, 6 も DR の内容に左右されない。また、CTL-15 は Address Exchange

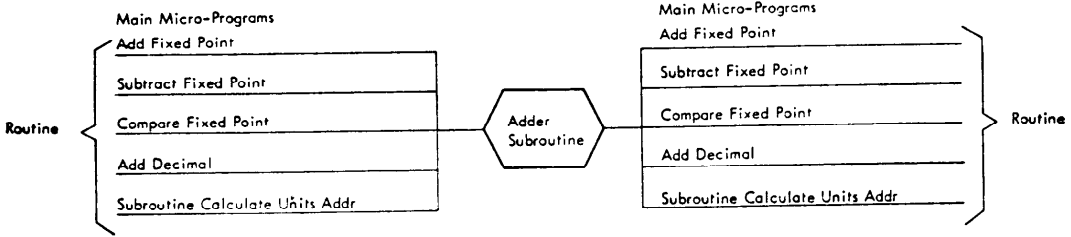


Fig. 5 Principles of Using a Subroutine

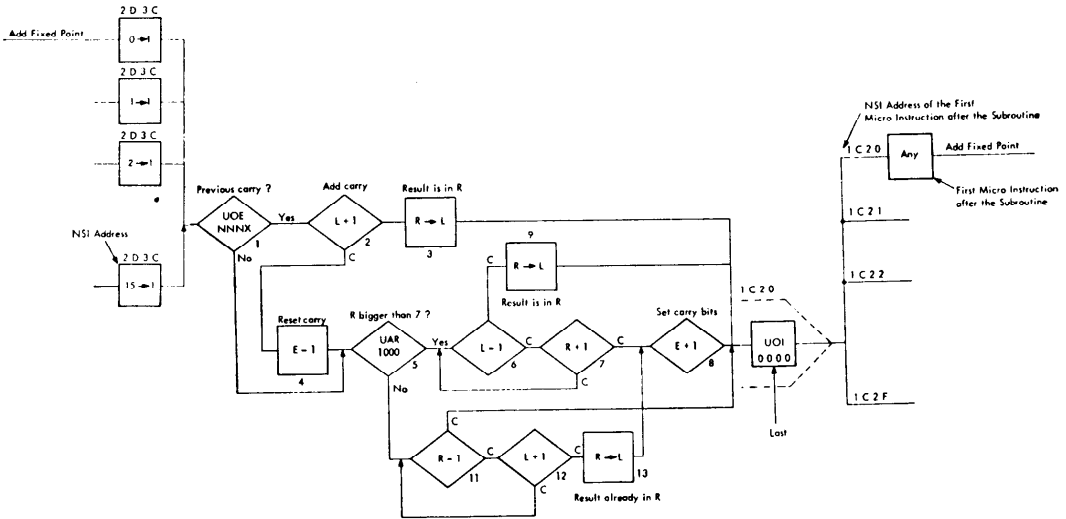


Fig. 6 Adder Subroutine

オペレーションで、A, S, T, R とその Sub-Register の入れ換えを行なうことができ、この CTL-15 を使えばマイクロプログラム中で2組の Main Storage のアドレスを保持することができる。他の場合は、例えば DR-L のビット1、すなわち L1 が1で CTL-1 を使うと、Set Stop Condition が行なわれ、この条件 (コンディション) は後でマイクロプログラム中の SNS-15 で L4 に読み込まれる。

ほとんどの CTL-n は入出力オペレーションに使用されるので、あとは省略する。

4. マイクロプログラム・サブルーチン

マイクロプログラムでは共通のルーチンがよく使われる。この種のサブルーチンを例にとって解説する。

4.1 Adder

S/360-20 は ALU を持たないので Add もマイクロプログラムで実行される。Adder のサブルーチンは、Add Fixed Point, Subtract Fixed Point, Compare Fixed Point, Add Decimal などのマイクロ

ログラム中で Fig. 5 のように使われる。Fig. 6 に Adder のマイクロプログラム・サブルーチンを示す。この Adder は DR-R と DR-L との Add を行なうもので、サブルーチンに入る前にデータを DR-R, L に移しておかなければならない。結果は DR-L に残り、DR-R の内容は破壊される。Add の結果、もしキャリーが出ると DR-E の1ビット、つまり E1 にストアされる。このサブルーチンに入る前にサブルーチンから自分に帰るために DR-I に定数を入れ、かつその $\overline{N} \rightarrow \overline{I}$ の NSI アドレスとしてサブルーチンの最初のアドレス (2D3C) を指定する。サブルーチンの最後のマイクロ命令は $\overline{UOI} \begin{matrix} 0 & 0 & 0 & 0 \end{matrix}$ で、DR-I の内容に従って16通りの分岐アドレスの1つを決定する。すなわち、サブルーチンは最大16のマイクロプログラムで共有され得る。Add Fixed Point の場合は $\overline{0} \rightarrow \overline{I}$ としてあるので、サブルーチンから出るアドレスは IC 20 となる。

Fig. 7 の例に従って Fig. 6 のサブルーチンを1ス

Example Carry + 7 + 6 = 14

Step	Micro Instruction	DR-E	DR-R	DR-L
Entry		1	7	6
1	Test previous carry	Yes		
2	L + 1 (Add carry)			7
4	E - 1 (Reset carry)	0		
5	Test DR-R for an 8 bit		No	
11	R - 1		6	
12	L + 1			8
11	R - 1		5	
12	L + 1			9
11	R - 1		4	
12	L + 1			10
11	R - 1		3	
12	L + 1			11
11	R - 1		2	
12	L + 1			12
11	R - 1		1	
12	L + 1			13
11	R - 1		0	
12	L + 1			14
11	R - 1		15 carry	
10	Exit	0 No carry to next position		14 = Sum

Example Carry + 7 + 9 = 1 and Carry

Step	Micro Instruction	DR-E	DR-R	DR-L
Entry		1	7	9
1	Test previous carry	Yes		
2	L + 1			10
4	E - 1	0		
5	Test DR-R for an 8 bit		No	
11	R - 1		6	
12	L + 1			11
11	R - 1		5	
12	L + 1			12
11	R - 1		4	
12	L + 1			13
11	R - 1		3	
12	L + 1			14
11	R - 1		2	
12	L + 1			15
11	R - 1		1	
12	L + 1			0 carry
13	R to L			
8	E + 1	1		
10	Exit	1 carry to next position		1 = Sum

Example 9 + 8 = 1 and Carry

Step	Micro Instruction	DR-E	DR-R	DR-L
Entry		0	9	8
1	Test previous carry	No		
5	Test DR-R for an 8 bit		Yes	
6	L - 1			7
7	R + 1		10	
6	L - 1			6
7	R + 1		11	
6	L - 1			5
7	R + 1		12	
6	L - 1			4
7	R + 1		13	
6	L - 1			3
7	R + 1		14	
6	L - 1			2
7	R + 1		15	
6	L - 1			1
7	R + 1		0 carry	
8	E + 1	1		
10	Exit	1 carry to next position		1 = Sum

Fig. 7 Examples of Add Operation

テップずつ追えば動作原理は自明であろう。

4.2 STR+1

マイクロプログラム中で Main Storage を次々にアドレスするためにアドレスに +1 して行くときに使われる。Fig. 8 に STR+1 のマイクロプログラム・サブルーチンを示す。このサブルーチンは A, S, T, R を連結した DR として +1 を行なう。サブルーチンからの出口は、オーバーフローしたときとそうでないときの 2 つがある。ステップ中、 $\boxed{S \rightarrow L}$ は Extension-Bit A の内容を DR-U の 1, 2 ビットに移すため、また $\boxed{L \rightarrow S}$ は DR-U の 1, 2 ビットに従って Extension-Bit を新しくセットするためのものである。このサブルーチンに入る前に出口を決めるための数値を DR-I にセットする。

4.3 Compare Table with Three Exits

DR-L の内容と DR-X の内容を比較するときに使われる。X は D, S, T, R, U のどれでもよい。DR-L = DR-X, DR-L < DR-X, DR-L > DR-X の 3 通りの判断ができる。Fig. 9 にこのマイクロプログラム・サブルーチンを示す。USE マイクロ命令を有効に使った例として興味深い。

4.4 Validity and Zero Test Table

Fig. 10 のマイクロプログラム・サブルーチンは、DR-X (U, L, E, P) の内容が 0 か、1 ~ 9 か、10 ~ 15 かの判断をするときに用いられる。Decimal オペレーション中の Validity テストなどに使用できる。

5. マイクロプログラム・フロー

S/360-20 のオペレーションを行なうためのマイクロプログラム・フローは大きく分けると Fig. 11 のようになる。Start Key を押すことによりすべてのオペレーションが開始される。Manual Routine では、CPU オペレーションのモード、すなわちプロセス、インストラクション・ステップ、またはマニュアル・オペレーションなどを判断する。Storage Display などのマニュアル・オペレーションはこの Routine でなされる。また、Stop Key が押されると、停止条件がこの Routine で検出され CPU は停止させられる。

I-Phase では、これから実行される命令の op コードが Main Storage から取られ、どの命令形式か (R, R, RX, SI, SS) を決め、命令を解析して E-Phase に必要な情報を DR, Auxiliary Storage の特定のアドレスにストアする。また、これから命令の長さを知

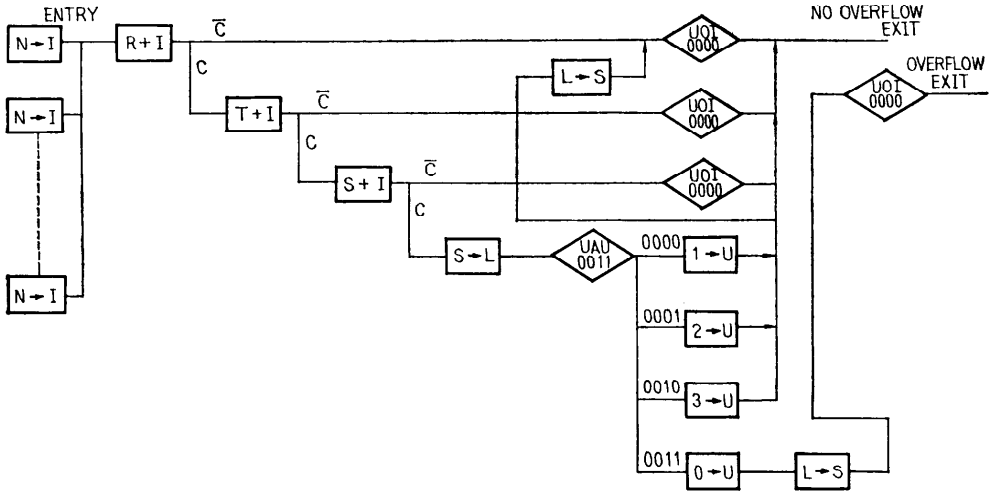


Fig. 8 STR+1 Subroutine, 16 k Byte Memory

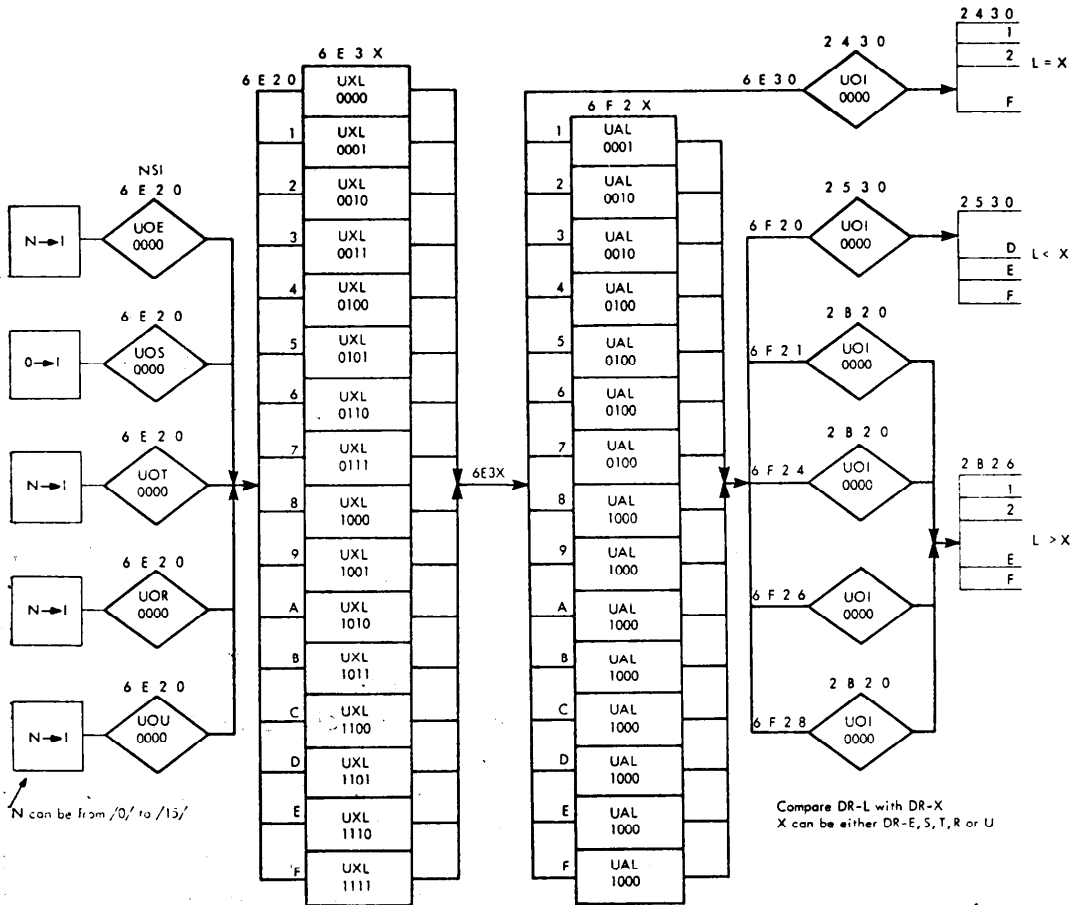


Fig. 9 Compare Table with Three Exits Subroutine

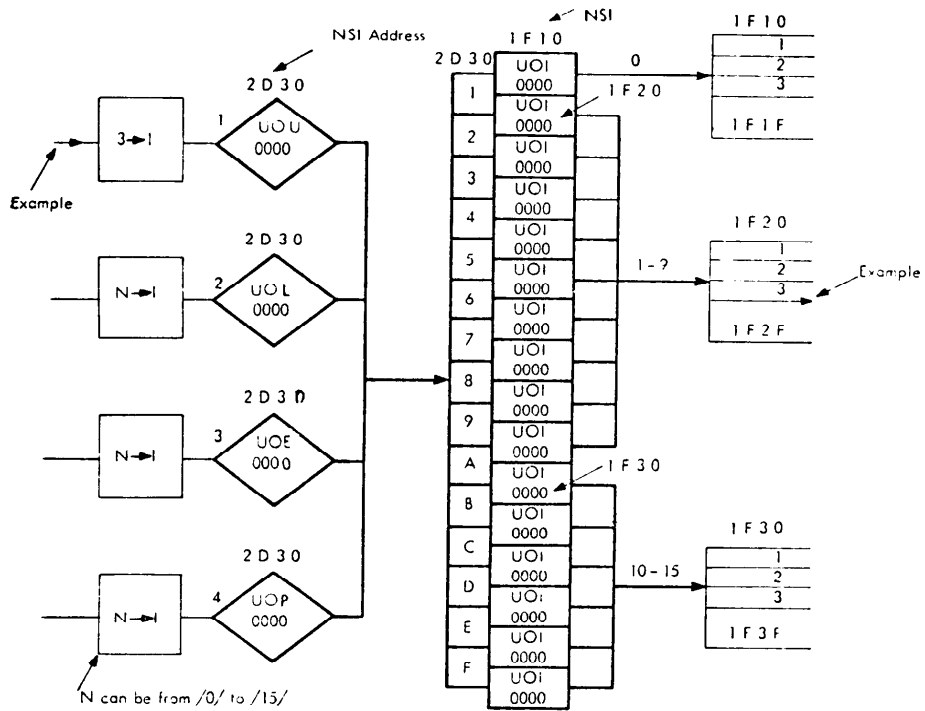


Fig. 10 Validity and Zero Test Table Subroutine

Common E-Phase を取り、各入出力機器固有の E-Phase を取ってオペレーションを完了する。

6. マニュアルオペレーション・マイクロプログラム

5 で述べたマイクロプログラム・フロー中の Manual Routine に相当する部分について説明する。実装されているマイクロプログラムは種々の入出力オペレーション、CE オペレーションを含んでおり複雑であるが、ここでは System Reset と基本的なマニュアルオペレーションに関係したものだけを拾うことにした。Fig. 13, Fig. 14 のマイクロプログラム、および Fig. 12 のコンソルパネルを参照しながら説明する。

6.1 System Reset

Power On したとき、または System Reset Key を押すと、ROAR に特定アドレスが強制的にストアされ、System Reset のマイクロプログラムが走り出す。このオペレーションは CPU をプロセス可能にするもので、エラー条件などをリセットするため、データフロー中の DR, Auxiliary Storage をクリアする。General Register, PSW の一部、INST RECALL

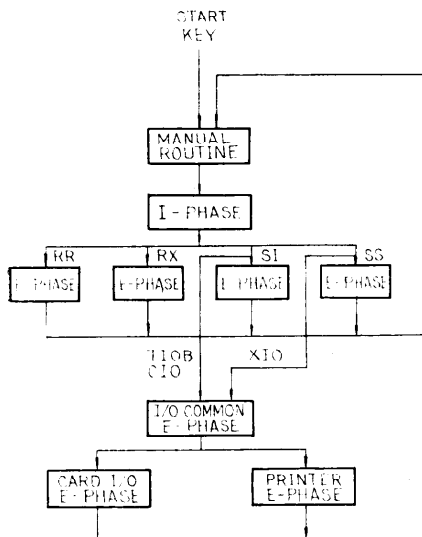
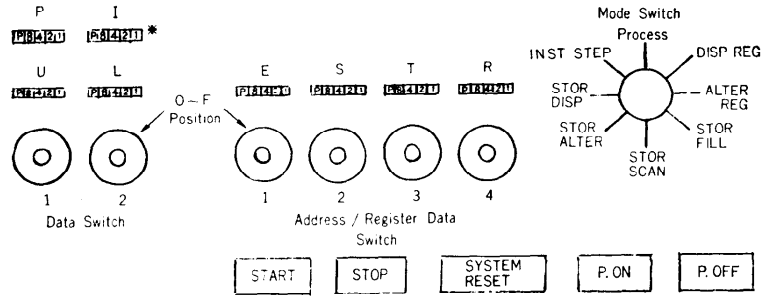


Fig. 11 System Operation Microprogram Flow

り次の命令のアドレスを生成して Auxiliary Storage の決められたアドレスにストアし、次の I-Phase の準備をする。各命令はそれぞれの E-Phase に従って実行される。入出力命令の TIOB/CIO, XIO では、



* P|I|U|L は P|8|4|2|1 である。

Fig. 12 Console

ADD REG など、次の命令を実行するのに必要な情報は Auxiliary Storage 中でもクリアされない (Fig. 2 参照)。Auxiliary Storage が F·STR/S·STR によってアクセスできるように CTL-5 が使用される。System Reset のマイクロプログラムは CTL-0 により CPU を停止させる。このあと Start Key を押すと CTL-0 の NSI アドレスで指定されるマイクロプログラムから始まる。CPU の停止はすべてこの CTL-0 によって行なわれる。すべてのオペレーションは Start Key を押すことによって CTL-0 で指定された NSI アドレスのマイクロ命令から始まる。

6.2 Storage Display

Address/Register Data Switch 1, 2, 3, 4 で指定されたアドレスの Main Storage の内容を DR-U, L に、指定したアドレスは DR-E, S, T, R に表示される。Fig. 12 の Mode Switch を STOR DISP の位置にし、Start Key を押すとこのオペレーションが行なわれる。この位置のとき SNS-0 の結果 DR-L=4 となる。DR-E は直接には Storage のアドレス生成につながらないが、A がコンソル上に表示されないで E を表示用使用する。

6.3 Storage Alter

Address/Register Data Switch 1, 2, 3, 4 で指定されたアドレスの Main Storage に Data Switch 1, 2 の内容を書き込む。書き込まれたデータは DR-U, L へ、指定されたアドレスは DR-E, S, T, R へ表示される。SNS-0 の結果 DR-L=2 となる。

6.4 Storage Scan

Storage にストアされている情報を次から次に読み出し正しい P ビットを持っているか否かをチェックす

る。エラーがあると CPU は停止し、エラーローションアドレスが DR-E, S, T, R に表示される。アドレスを自動的に +1 するために Fig. 8 の STR+1 のサブルーチンが使用されている。アドレスは最大アドレスを越えると 0000 にもどる。Stop Key を押すと停止条件がセットされ、SNS-15 により L4 に Sense され、オペレーションは CTL-0 により停止する。SNS-0 の結果 DR-L=8 となる。

6.5 Storage Fill

Data Switch 1, 2 の内容を Storage に次から次へとストアする。SNS-0 の結果は DR-L=10 となる。オペレーションは Stop Key を押すまで続けられる。

6.6 Register Display

General Register (半語長) などを DR-E, S, T, R へ表示する。Data Switch 1 のポジション (0~F) に従い、Fig. 2 の左から 2 バイト分が上から下に表示される。General Register の指定は Data Switch 1 の 8~F によって行なわれる。DR-P は Data Switch の位置を示す。DR-U, L はクリアされている。SNS-0 の結果 DR-L=5 となる。

6.7 Register Alter

Data Switch 1 により指定される。Fig. 2 の左から 2 バイト分に、Address/Register Data Switch 1, 2, 3, 4 で指定されたデータをストアする。General Register を指定する場合は Data Switch 1 が 8~F でなければならない。DR-P は Data Switch 1 の位置を、DR-E, S, T, R はストアされたデータを表示する。SNS-0 の結果は、DR-L=3 となる。

(昭和 48 年 3 月 1 日受付)