

剰余演算μコード設計仕様

1.仕様

- ・すべての命令コードは 1cyc で実行
- ・1cyc は、8MC
- ・1cyc の遅延分岐
- ・サブルーチンコールは、1 段のみ可能
- ・容量 512ワード(1ワード 32bit)

2.剰余演算器レジスタ及び、フラグ

	レジスタ名	bit 幅	用途
1	A	1024	演算レジスタ、R0-R15 にストア可能な唯一のレジスタ
2	B	1024	演算レジスタ
3	C	1024	演算レジスタ
4	D	1024	演算レジスタ
5	M0A	1024	モンゴリ乗算専用レジスタ
6	M0B	1024	モンゴリ乗算専用レジスタ
7	M1A	1024	モンゴリ乗算専用レジスタ
8	M1B	1024	モンゴリ乗算専用レジスタ
9	DP	10	モンゴリ乗算専用レジスタ、Dレジスタから 1ビットを選択するためのポインタ
10	CF	1	キャリーフラグ、(暫定仕様)
11	ZF	1	ゼロフラグ、Dレジスタが 0 のとき 1、TESTD/SUB が選択されたとき更新(暫定仕様)
12	I	10	ループ制御カウンタ
13	J	10	ループ制御カウンタ
14	R0-R15	1024	データレジスタ
15	PC	9	プログラムカウンタ
16	RETADR	9	サブルーチンコールのためのリターンアドレス格納

2-1.演算レジスタとデータレジスタ間のデータ転送サイクル

- ロード レジスタ ← データレジスタ 1cyc(但し、1cyc 前にデータレジスタのレジスタ番号を指定)
 ストア データレジスタ ← Aレジスタ 1cyc

2-2. キャリーフラグとゼロフラグ

CF(キャリーフラグ)は、μコードのビット15を1としたこき加算器のキャリー信号値を取り込む。主に BCF 命令とともに、値の比較を行なうために利用される。

ZF(ゼロフラグ)は、SUB/TESTD が選択されたとき Dレジスタが 0 のとき 1 の値をセットし、それ以外の場合 0 の値がセットされる。

2-3. レジスタI,J

レジスタI,J はループ制御専用レジスタであり、ロードとデクリメントがある。I,J に違いはないが、ロードできる値が異なる。Jレジスタで乗算、除算、モンゴリ乗算のループ制御を行ない、Iレジスタでべき乗の指数のループ制御を行なうことを前提として I,J にロードされる固定値が決められている。

2-4. プログラムカウンタ

μコード領域 512ワードからμコードを実行するアドレスを指す 9bit のレジスタ。分岐命令を除き、いずれの命令を実行してもインクリメントされる。分岐命令(Bを除く)の場合は、μコードの bit24-31 の 8bit が 9bit に符号拡張されプログラムカウンタに加算される。小さいアドレスへの分岐は、2 の補数を利用する。μコードの bit 24-31 はデータレジスタ番号の指定と兼用であるため分岐命令と同じサイクルではデータレジスタ番号を指す命令は使用できない。B 命令ではμコードの bit23-31 がプログラムカウンタにセットされる。このため B 命令をモンゴリ乗算を行なうところで使用する場合には注意する。

3. μ コード命令コードフォーマット

bit0	bit1	bit2	bit3	bit4	bit5	bit6	bit7
制御命令コード				加算器左側セクタ		加算器右側セクタ	

bit8	bit9	bit10	bit11	bit12	bit13	bit14	bit15
加減算指定	INV	加算器出力	比較減算	乗算用加算	ストア	ALレジスタセクタ	

bit16	bit17	bit18	bit19	bit20	bit21	bit22	bit23
Bレジスタセクタ		Cレジスタセクタ		Dレジスタセクタ		MMUL	MRES

制御コードが非分岐命令ケース

bit24	bit25	bit26	bit27	bit28	bit29	bit30	bit31
ロードするデータレジスタ番号				ストアするデータレジスタ番号			

制御コードが分岐命令ケース

bit24	bit25	bit26	bit27	bit28	bit29	bit30	bit31
相対アドレス							

注) B 命令時は、23-31ビットが絶対アドレスを示す。

4. μ コードニモニック

項#	ニモニック	bit	値	意味
1	NoP	0-3	0	NoP
2	I=160		1	I=160
3	I=1024		2	I=1024
4	I <= n		3	ILレジスタに n(10bit 符号無)の値を代入
5	J=256		4	J=256
6	J=512		5	J=512
7	J=1024		6	J=1024
8	B(ラベル)		7	bit23-31 で示されるアドレスに分岐
9	BZF(ラベル)		8	ZF=1 の場合、分岐
10	BC(ラベル)		9	Cレジスタの bit1023 が 0 なら分岐
11	BI(ラベル)		10	I=I-1, I≠0 なら分岐
12	BJ(ラベル)		11	J=J-1, J≠0 なら分岐
13	BCF(ラベル)		12	CF=1 の場合、分岐
14	CALL(ラベル)		13	PC を RETADR に退避し、分岐
15	RET		14	RETAADR+1 のアドレスにリターン
16	END		15	μ コード終了
17	ADDL=[0,A,DIVA]	4-5	0:0 1:A 2:割り算用 A	
18	ADDR=[0,B,C,D]	6-7	0:0, 1:B, 2:C, 3:D	
19	SUB / TESTD	8	加算器のキャリーイン ; Dレジスタの 0テスト	
20	INV	9	ADDR 出力を反転	
21	SEL (X)	10	データレジスタからの出力切替え	
22	CMP	11	比較減算指定、DPカウントアップ信号	
23	MUL	12	乗算用加算指定、モンゴリ乗算器面数指定	
24	Rn=A	13	ALレジスタの内容をデータレジスタn 番にストア	
25	A=[SEL0,SEL1,SEL2,SEL3]	14-15	ALレジスタ入力セクタ	
26	B=[SEL0,SEL1,SEL2,SEL3]	16-17	BLレジスタ入力セクタ	
27	C=[SEL0,SEL1,SEL2,SEL3]	18-19	CLレジスタ入力セクタ	
28	D=[SEL0,SEL1,SEL2]	20-21	DLレジスタ入力セクタ	
29	MMUL	22	モンゴリ乗算	
30	MRES	23	モンゴリ乗算演算レジスタクリア	
31	MMUL;MRES	22-23	モンゴリ乗算結果セレクト	
32	X=n	24-27	ロードするデータレジスタの番号指定	

5.μ コーディング

5-1. 乗算

式

$$BC = A \times C + B$$

モニツク

J=512

MUL0:

ADDL=A;ADDR=B;MUL;B=SEL1;C=SEL1;BJ(MUL0)

ADDL=A;ADDR=B;MUL;B=SEL1;C=SEL1

5-2. 除算

式

$$BC = BC \div D$$

$$A = BC \bmod D$$

モニツク

J=1024;A=SEL1

DIV0:

ADDL=DIVA;ADDR=D;CMP;SUB;INV;A=SEL3;B=SEL3;C=SEL3;BJ(DIV0)

ADDL=DIVA;ADDR=D;CMP;SUB;INV;A=SEL3;B=SEL3;C=SEL3

5-3. 剰余演算(R2 偶数ケース)

式

$$R3 = R0^{R1} \bmod R2$$

モニツク

I=1024;X=0

SEL(X);D=SEL2

CALL(POWERSUB)

NOP

END(0)

POWERSUB:

ADDL=0;ADDR=0;SUB;A=SEL1

R0=A;X=2

ZLBL0:

SEL(X);C=SEL2;

ADDL=0;ADDR=C;B=SEL1

BC(ZLBL1);ADDL=0;ADDR=B;A=SEL1

R2=A

J=512;ADDL=0;ADDR=0;B=SEL2;X=0

SEL(X);A=SEL1;X=1

SEL(X);C=SEL2

ZMUL0:

ADDL=A;ADDR=B;MUL;B=SEL1;C=SEL1;BJ(ZMUL0)

ADDL=A;ADDR=B;MUL;B=SEL1;C=SEL1

J=1024;ADDL=0;ADDR=0;A=SEL1

ZDIV0:

ADDL=DIVA;ADDR=D;CMP;SUB;INV;A=SEL3;B=SEL3;C=SEL3;BJ(ZDIV0)

ADDL=DIVA;ADDR=D;CMP;SUB;INV;A=SEL3;B=SEL3;C=SEL3

R0=A

ZLBL1:

J=512;ADDL=0;ADDR=0;B=SEL2;X=1

SEL(X);A=SEL1;C=SEL2

ZMUL1:

ADDL=A;ADDR=B;MUL;B=SEL1;C=SEL1;BJ(ZMUL1)

ADDL=A;ADDR=B;MUL;B=SEL1;C=SEL1

J=1024;ADDL=0;ADDR=0;A=SEL1

ZDIV1:

ADDL=DIVA;ADDR=D;CMP;SUB;INV;A=SEL3;B=SEL3;C=SEL3;BJ(ZDIV1);

ADDL=DIVA;ADDR=D;CMP;SUB;INV;A=SEL3;B=SEL3;C=SEL3

R1=A;ADDL=0;ADDR=0;B=SEL2;

BI(ZLBL0)

X=2;

RET

NOP

5-4. 剰余演算(R2 奇数ケース)

モンゴリ乗算を用いた剰余演算法

式

$$R3 = R0^{R1} \bmod R2$$

モニツク

```
A=SEL1;C=SEL2;X=0
SEL (X);D=SEL2;X=1
SEL (X);B=SEL2;J=1024
ADIV0:
ADDL=DIVA;ADDR=D;CMP;SUB;INV;A=SEL3;B=SEL3;C=SEL3;BJ(ADIV0)
ADDL=DIVA;ADDR=D;CMP;SUB;INV;A=SEL3;B=SEL3;C=SEL3
R1=A;SUB;B=SEL2
A=SEL1;C=SEL2;J=1024
ADIV1:
ADDL=DIVA;ADDR=D;CMP;SUB;INV;A=SEL3;B=SEL3;C=SEL3;BJ(ADIV1)
ADDL=DIVA;ADDR=D;CMP;SUB;INV;A=SEL3;B=SEL3;C=SEL3;X=0
SEL (X);B=SEL2;X=2
SEL (X);C=SEL2;I=1024;X=1
SEL (X);D=SEL2;J=256;XCOUNT;MRES
### main loop ###
ALBL1:
XCOUNT;MMUL;BJ(ALBL1)
XCOUNT;MMUL
MMUL;MRES;MUL;D=SEL1;A=SEL1
CMP;SUB;INV;ADDL=A;ADDR=B;A=SEL1;BC(ALBL2)
D=SEL1;ADDL=0;ADDR=D;A=SEL1
MMUL;MRES;A=SEL1
CMP;SUB;INV;ADDL=A;ADDR=B;A=SEL1
R0=A
ALBL2:
BI(ALBL1);MRES
C=SEL1;J=256;XCOUNT
NOP
J=256;XCOUNT;SUB;D=SEL2;MRES
ALBL3:
XCOUNT;MMUL;BJ(ALBL3)
XCOUNT;MMUL
MMUL;MRES;A=SEL1;
CMP;SUB;INV;ADDL=A;ADDR=B;A=SEL1
R0=A
END (0)
```