SPU81SystemDesign_V2　の内容

SPU81SystemDesign V1→　V2　への変更内容

SPU81（シーケンス・プロセッサ・ユニット）をVerilogHDLでFPGAに設計しました。

前に記録した数枚の仕様書とMyマイコンComet2eの設計資料を参考にして設計しました。このため、設計時間は短くなりましたが厳密な定義がおろそかになりました。全体的には、違和感が無く使い勝手の良い物となりました。しかし、一部の命令を削除したり使い勝手が悪いところもあります。

ここに、変更した内容を記録します。

①STM　a,M　命令の使用時の注意

STM　a,M　命令の直前の命令が、STM　a,M　命令で使うＡｃｃに書き込む場合、直前の命令がＡｃｃに書き込むデータは採用されません。データを書き込む前のデータが採用されます。

対策は、Ａｃｃに書き込む命令とSTM　a,M　命令の間にＮＯＰ命令を挿入します。

②BITS　b,r命令とBITR　b,r命令は、実装していません。使わないでください。

③Ａｃｃをレジスタ番号で指定できる様にしました。

例えば、ADD　a,r命令で、Ａｃｃ0とＡｃｃ0の加算ができます。

④STM　a,M　命令順序操作のシーケンスの一部変更

命令レジスタの内容をメモリアドレスにするルートを追加しました。

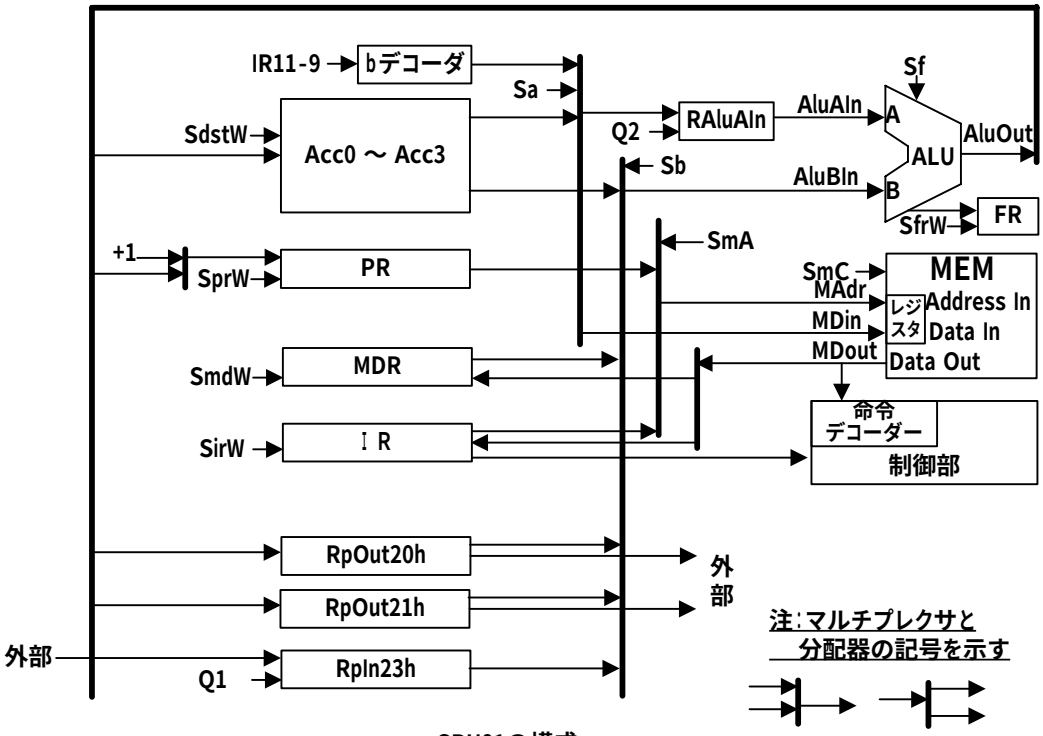
⑤アッセンブラで使用する「Asm命令処理コード」を追記しました。

SPU81命令一覧表

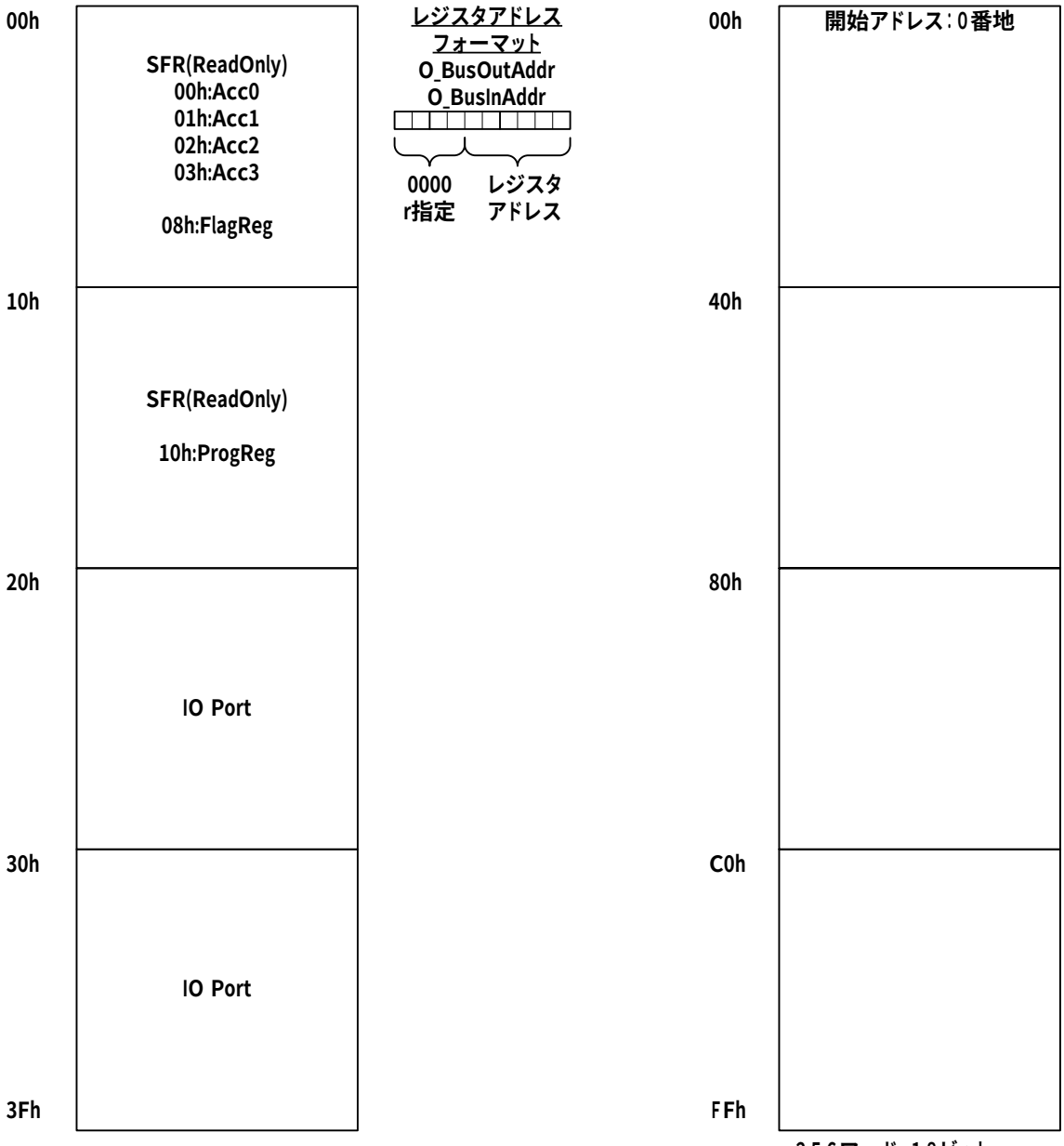
SPU81命令一覧表

番号	アッセンブル命令	機械語命令(ビット番号) 17 12	11 8	7 4	3 0	フラグ OF	SF	ZF	命令の説明
1	NOP	00	0	----	----	--	--	--	何もしない。
2	LDM a,M	01	-aa-	MMMMMMMM		0	○	○	ロード：a ←（メモリ）
3	LD a,r	02	-aa-	--rrrrrr		0	○	○	ロード：a ←（レジスタ）
4	LDI a,I	03	-aa-	IIIIIIII		0	○	○	ロード：a ←命令の下位８ビット
5	ADD a,r	04	-aa-	--rrrrrr		○	○	○	算術加算　：a ←（a）＋（レジスタ）
6	ADDI a,I	05	-aa-	IIIIIIII		○	○	○	算術加算　：a ←（a）＋ 命令の下位８ビット
7	SUB a,r	06	-aa-	--rrrrrr		○	○	○	算術減算　：a ←（a）－（レジスタ）
8	SUBI a,I	07	-aa-	IIIIIIII		○	○	○	算術減算　：a ←（a）－ 命令の下位８ビット
9	AND a,r	08	-aa-	--rrrrrr		0	○	○	論理積　：a ←（a）AND（レジスタ）
10	ANDI a,I	09	-aa-	IIIIIIII		0	○	○	論理積　：a ←（a）AND 命令の下位８ビット
11	OR a,r	0A	-aa-	--rrrrrr		0	○	○	論理和　：a ←（a）OR（レジスタ）
12	ORI a,I	0B	-aa-	IIIIIIII		0	○	○	論理和　：a ←（a）OR 命令の下位８ビット
13	XOR a,r	0C	-aa-	--rrrrrr		0	○	○	排他的論理和：a ←（a）XOR（レジスタ）
14	XORI a,I	0D	-aa-	IIIIIIII		0	○	○	排他的論理和：a ←（a）XOR 命令の下位８ビット
15	CP a,r	0E	-aa-	--rrrrrr		0	○	○	算術比較：（a）＞（r）なら SF=0，ZF=0
									（a）＝（r）なら SF=0，ZF=1
									（a）＜（r）なら SF=1，ZF=0
16	CPI a,I	0F	-aa-	IIIIIIII		0	○	○	算術比較：（a）＞ 命令の下位８ビット なら SF=0，ZF=0
									（a）＝ 命令の下位８ビット なら SF=0，ZF=1
									（a）＜ 命令の下位８ビット なら SF=1，ZF=0
17	ST a,r	10	-aa-	--rrrrrr		--	--	--	ストア：指定のレジスタ ←（a）
18	STM a,M	11	-aa-	MMMMMMMM		--	--	--	ストア：指定のメモリ　←（a） 直前の命令が同じＡｃｃに書き込む場合、その前のＡｃｃの内容が書き込まれる。対策は、ＮＯＰ命令を挿入する。
19	BITT b,r	14	bbb-	--rrrrrr		--	--	○	ビットテスト　：（r）の（b）が１なら ZF=1、０なら ZF=0
20	~~BITS b,r~~	~~16~~	~~bbb-~~	~~--rrrrrr~~		--	--	--	~~ビットセット　：（r）の（b）を１にする~~
21	~~BITR b,r~~	~~17~~	~~bbb-~~	~~--rrrrrr~~		--	--	--	~~ビットリセット：（r）の（b）を０にする~~
22	JUMP adr	30	----	address8		--	--	--	無条件に adr にジャンプする。
23	JNZ adr	32	----	address8		--	--	--	ZF=0 の場合 adr にジャンプ。以外は，次の命令。
24	JZE adr	33	----	address8		--	--	--	ZF=1 の場合 adr にジャンプ。以外は，次の命令。
25	JPL adr	34	----	address8		--	--	--	SF=0 で ZF=0 の場合 adr にジャンプ。以外は，次の命令。
26	JMI adr	35	----	address8		--	--	--	SF=1 の場合 adr にジャンプ。以外は，次の命令。
27	JOV adr	37	----	address8		--	--	--	OF=1 の場合 adr にジャンプ。以外は，次の命令。
28	ORG adr	--	----	address8		--	--	--	プログラムの先頭アドレスを設定する。擬似命令

注：機械語命令は、１６進表示。
-aa-：アキュームレータ を２ビットで指定する。
bbb-：アキュームレータ を３ビットで指定する。
--rrrrrr ：レジスタ を６ビットで指定する。
MMMMMMMM：メモリアドレス値を８ビットで指定する。
IIIIIIII：命令の下位８ビットの値を演算値とする。
address8：アドレス値を８ビットで指定する。



SPU81の構成



レジスタの割り付け

256ワードx18ビット
メモリの割り付け

命令とALU制御

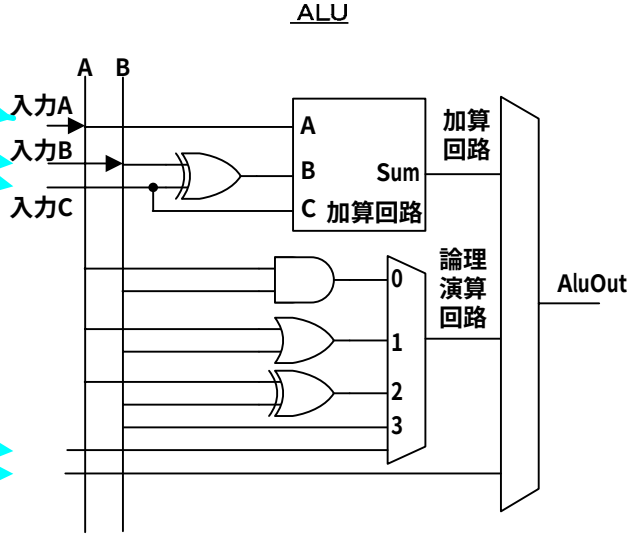
番号	ｱｯｾﾝﾌﾞﾙ命令	機械語命令(ビット番号) 17 12	11 8	7 4	3 0	Asm命令処理コード 1	2	3	4	5	6	グループ名	フラグ OF	SF	ZF	A Bus	B Bus	C Bus	演算内容
1	NOP	00 0000	0	----	----	○							--	--	--	-	-	-	-
2	LDM a,M	00 0001	-aa-	MMMMMMMM					○				0	○	○	-	Mdr	-	B
3	LD a,r	00 0010	-aa-	--rrrrrr			○						0	○	○	-	r	-	B
4	LDI a,I	00 0011	-aa-	IIIIIIII				○					0	○	○	-	I	-	B
5	ADD a,r	00 0100	-aa-	--rrrrrr			○						○	○	○	a	r	0	A+B+C
6	ADDI a,I	00 0101	-aa-	IIIIIIII				○					○	○	○	a	I	0	A+B+C
7	SUB a,r	00 0110	-aa-	--rrrrrr			○						○	○	○	a	r	1	A+/B+C
8	SUBI a,I	00 0111	-aa-	IIIIIIII				○					○	○	○	a	I	1	A+/B+C
9	AND a,r	00 1000	-aa-	--rrrrrr			○						0	○	○	a	r	-	A & B
10	ANDI a,I	00 1001	-aa-	IIIIIIII				○					0	○	○	a	I	-	A & B
11	OR a,r	00 1010	-aa-	--rrrrrr			○						0	○	○	a	r	-	A | B
12	ORI a,I	00 1011	-aa-	IIIIIIII				○					0	○	○	a	I	-	A | B
13	XOR a,r	00 1100	-aa-	--rrrrrr			○						0	○	○	a	r	-	A XOR B
14	XORI a,I	00 1101	-aa-	IIIIIIII				○					0	○	○	a	I	-	A XOR B
15	CP a,r	00 1110	-aa-	--rrrrrr			○						0	○	○	a	r	1	A+/B+C
16	CPI a,I	00 1111	-aa-	IIIIIIII				○					0	○	○	a	I	1	A+/B+C
17	ST a,r	01 0000	-aa-	--rrrrrr			○					Ist	--	--	--	a	0	-	A | B
18	STM a,M	01 0001	-aa-	MMMMMMMM					○				--	--	--	a	0	-	A | B
19	BITT b,r	01 0100	bbb-	--rrrrrr			○					Ild	--	--	○	b	r	-	A & B
20	~~BITS b,r~~	~~01 0110~~	~~bbb-~~	~~--rrrrrr~~			⊖						--	--	--	~~b~~	~~r~~	-	~~A | B~~
21	~~BITR b,r~~	~~01 0111~~	~~bbb-~~	~~--rrrrrr~~			⊖						--	--	--	~~/b~~	~~r~~	-	~~A & B~~
22	JUMP adr	11 0000	----	address8						○		Ijmp	--	--	--	-	adr	-	B
23	JNZ adr	11 0010	----	address8						○			--	--	--	-	adr	-	B
24	JZE adr	11 0011	----	address8						○			--	--	--	-	adr	-	B
25	JPL adr	11 0100	----	address8						○			--	--	--	-	adr	-	B
26	JMI adr	11 0101	----	address8						○			--	--	--	-	adr	-	B
27	JOV adr	11 0111	----	address8						○			--	--	--	-	adr	-	B

IRの命令をデコードしてALUを制御する

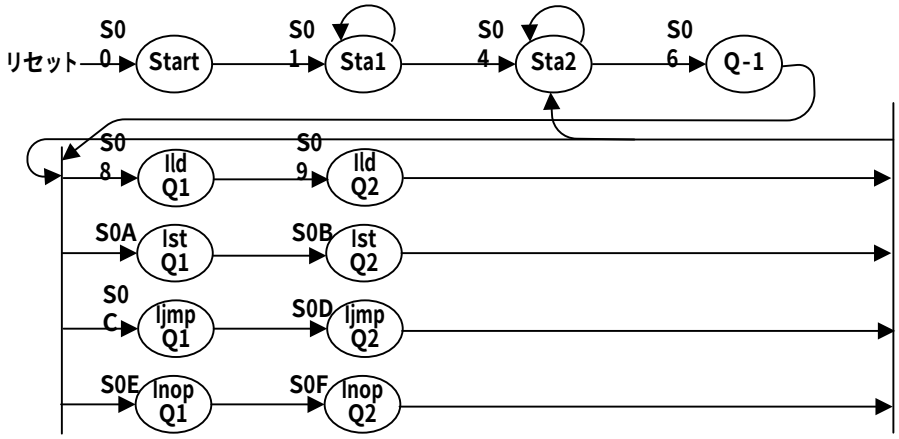
制御項目	条　件	制御信号
入力A制御	01 01xx:b選択	Sa:10
	01 0111:/b選択	Sa:11
	以外:a選択	Sa:00
入力B制御	00 0001:Mdr選択	Sb:010
	00 xxx1:I 選択	Sb:011
	除く:00 0001	
	01 000x:0 選択	Sb:100
	11 0xxx:adr 選択	Sb:101
	以外:r 選択	Sb:000
入力C制御	00 011x:1 選択	Sc:1
	00 111x:1 選択	Sc:1
	以外:0 選択	Sc:0
演算器選択制御	00 01xx:AddrU 選択	Salu:1
	00 111x:AddrU 選択	Salu:1
	以外:LogicU 選択	Salu:0
論理演算制御	00 100x:& 選択	Slu:00
	01 0100:& 選択	Slu:00
	01 0111:& 選択	Slu:00
	00 101x:| 選択	Slu:01
	01 000x:| 選択	Slu:01
	01 0110:| 選択	Slu:01
	00 1101:XOR 選択	Slu:10
	以外:B 選択	Slu:11
AluOut格納制御	00 xxxx:a 選択	Sdst:10
	除く:00 0000	
	00 111x	
	01 0000:r 選択	Sdst:11
	01 011x:r 選択	Sdst:11
	01 0001:メモリ 選択	Sdst:01
	以外:無し	Sdst:00

IRの命令をデコードしてフラグレジスタを制御する

制御項目	条　件	制御信号
フラグ格納制御	00 0001:0 ○○選択	Sf:0111
	00 001x:0 ○○選択	Sf:0111
	00 1xxx:0 ○○選択	Sf:0111
	00 01xx:○○○選択	Sf:1111
	01 0100:- - ○選択	Sf:0001
	以外:無し	Sf:0000



命令の状態遷移



ステート割付表

	0xh	1xh
0h	Start	
1h	Sta1	
2h		
3h		
4h	Sta2	
5h		
6h	Q-1	
7h		
8h	lld Q1	
9h	lld Q2	
Ah	lst Q1	
Bh	lst Q2	
Ch	ljmp Q1	
Dh	ljmp Q2	
Eh	lnop Q1	
Fh	lnop Q2	

Ildグループのシーケンス制御

グループ名	制御項目	Q-1		Q1		Q2	
		条　件	制御信号	条　件	制御信号	条　件	制御信号
Ild	入力A 制御	??	??	01 01xx：b選択 01 0111：/b選択 以外：a選択	Sa：10 Sa：11 Sa：00	a選択	Sa：00
	入力B 制御	??	??	同右	同右	00 0001：Mdr選択 00 xxx1：I 選択 除く：00 0001 以外：r 選択	Sb：010 Sb：011 Sb：000
	入力C 制御	??	??	同右	同右	00 011x：1 選択 00 111x：1 選択 以外：0 選択	Sc：1 Sc：1 Sc：0
	演算器選択 制御	??	??	00 01xx：AddrU 選択 00 111x：AddrU 選択 以外：LogicU 選択	Salu：1 Salu：1 Salu：0	00 01xx：AddrU 選択 00 111x：AddrU 選択 以外：LogicU 選択	Salu：1 Salu：1 Salu：0
	論理演算制御	??	??	同右	同右	00 100x：& 選択 01 0100：& 選択 01 0111：& 選択 00 101x：| 選択 01 000x：| 選択 01 0110：| 選択 00 1101：XOR 選択 以外：B 選択	Slu：00 Slu：00 Slu：00 Slu：01 Slu：01 Slu：01 Slu：10 Slu：11
	AluOut格納 制御	??	??	無し	Sdst：000	00 xxxx：a 選択 除く：00 0000 00 111x 01 011x：r 選択 以外：無し	Sdst：010 Sdst：011 Sdst：000
	フラグ格納制御	??	??	無し	Sf：0000	00 0001：0 ○○選択 00 001x：0 ○○選択 00 1xxx：0 ○○選択 00 01xx：○○○選択 01 0100：- - ○選択 以外：無し	Sf：0111 Sf：0111 Sf：0111 Sf：1111 Sf：0001 Sf：0000
	メモリ制御	次命令読み出し		読み出し中の次命令がLD a,M 命令なら、メモリ読み出し		次命令読み出し	

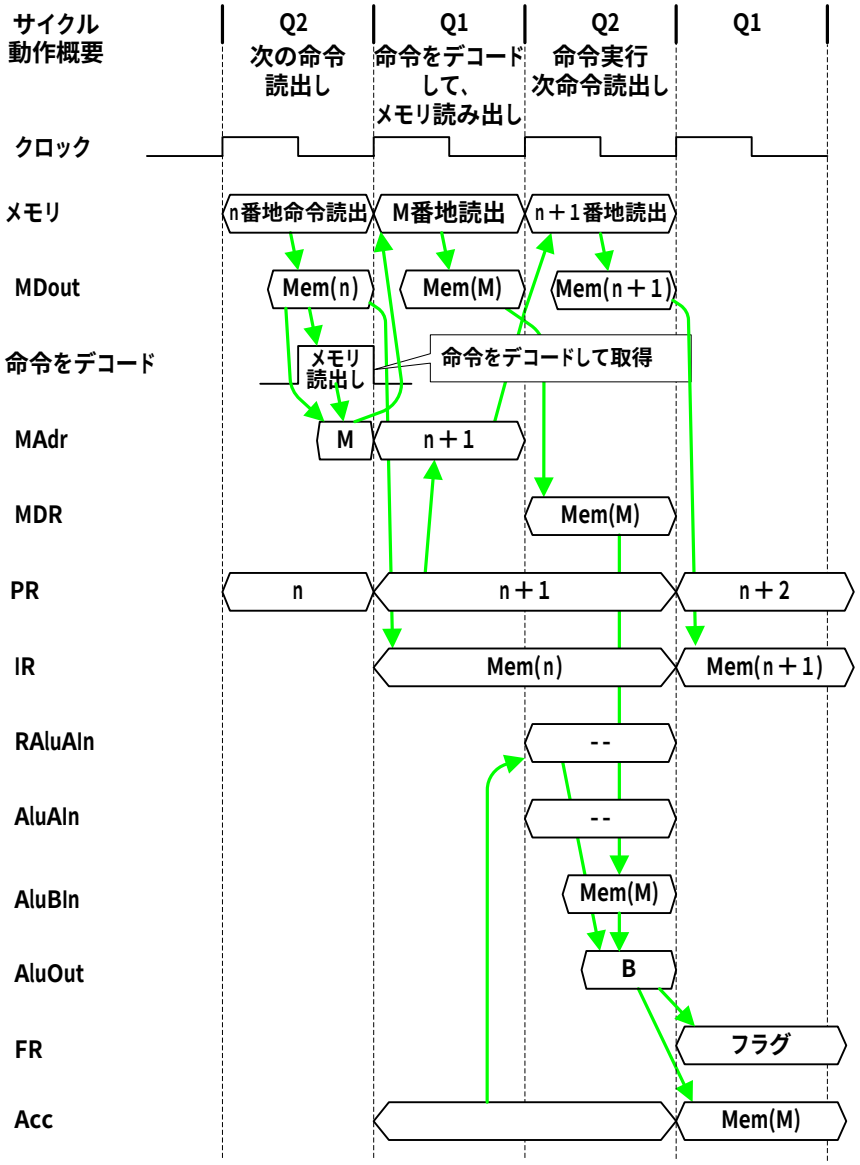
Istグループのシーケンス制御

グループ名	制御項目	Q-1		Q1		Q2	
		条　件	制御信号	条　件	制御信号	条　件	制御信号
Ist	入力A 制御	読み出し中の次命令がST a,M 命令なら、a選択	??	同右	同右	a選択	Sa：00
	入力B 制御	??	??	同右	同右	01 000x：0 選択	Sb：100
	入力C 制御	??	??	同右	同右	0 選択	Sc：0
	演算器選択制御	??	??	LogicU 選択	Salu：0	LogicU 選択	Salu：0
	論理演算制御	??	??	B 選択	Slu：11	B 選択	Slu：11
	AluOut格納制御	??	??	無し	Sdst：000	01 0000：r 選択 以外：無し	Sdst：011 Sdst：000
	フラグ格納制御	??	??	無し	Sf：0000	無し	Sf：0000
	メモリ制御	次命令読み出し		読み出し中の次命令がST a,M 命令なら、メモリ書き込み		次命令読み出し	

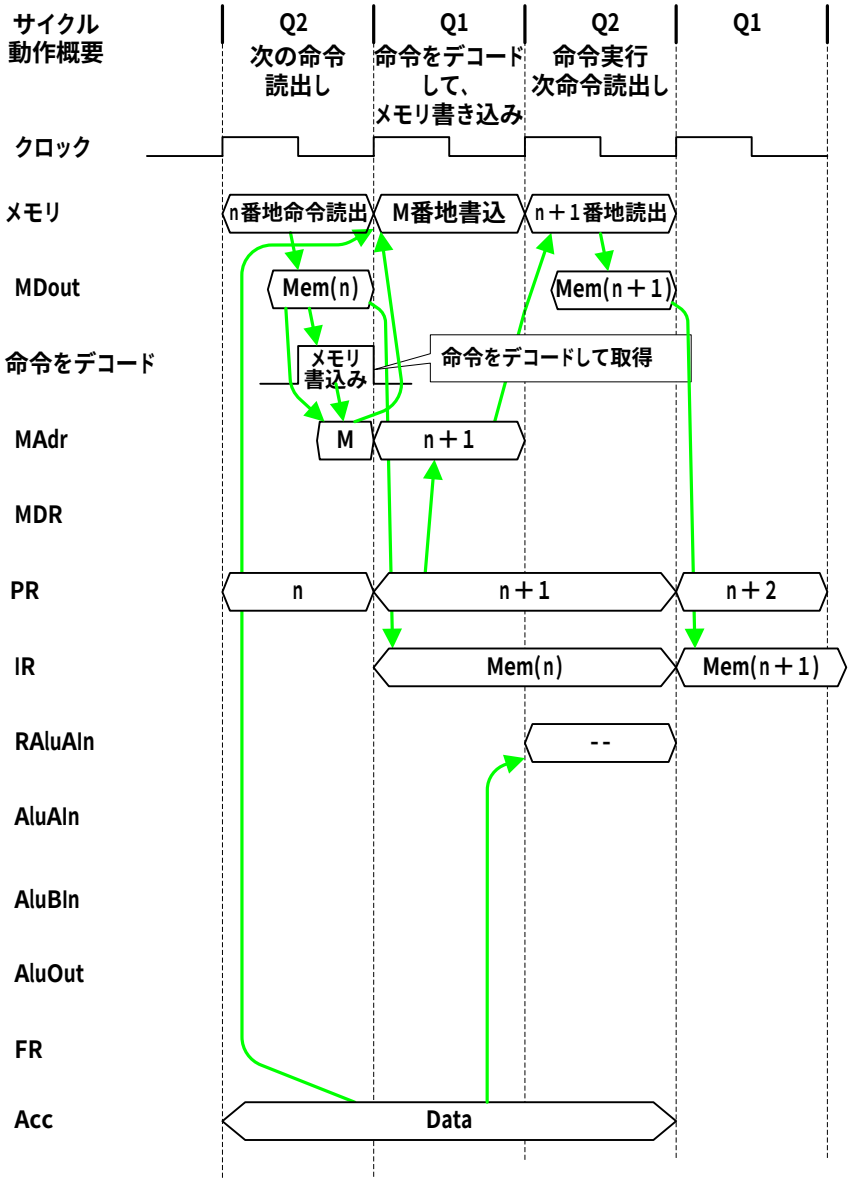
Ijmpグループのシーケンス制御

グループ名	制御項目	Q-1		Q1		Q2	
		条　件	制御信号	条　件	制御信号	条　件	制御信号
Ijmp	入力A 制御	??	??	a選択	Sa：00	同左	同左
	入力B 制御	??	??	IR 選択	Sb：101	同左	同左
	入力C 制御	??	??	0 選択	Sc：0	同左	同左
	演算器選択制御	??	??	LogicU 選択	Salu：0	同左	同左
	論理演算制御	??	??	B 選択	Slu：11	同左	同左
	AluOut格納制御	??	??	PR	Sdst：100	無し	Sdst：000
	フラグ格納制御	??	??	無し	Sf：0000	同左	同左
	メモ制御	次命令読み出し		LD a,M 命令なら、メモリ読み出し		次命令読み出し	

LDM a,M 命令順序操作のシーケンス



STM a,M 命令順序操作のシーケンス



JUMP　命令順序操作のシーケンス