CE1 情報電子工学総合実験　　第５回　配布資料

ステートマシン設計

1. データパス設計の確認（複数人で以下をチェックしあう）
   - 設計レビュー後の変更に問題はないか？
   - 詳細化は十分か？（制御信号に抜けはないか？）
2. 自分が設計したデータパスで各命令を実行するマイクロ操作を決定

   (1) EXCEL で以下のような表を作る

ADD 命令	
ステート	マイクロ操作
0	PC→MAR, PC++
1	MEM→IR, PC→MAR
2	MEM→DR, PC++
3	DR→MAR(ダイレクト) DR+GR[Rx]→MAR(インデクスド) やることなし(イミディエイト)
4	MEM→DR(ダイレクト、インデクスド) やることなし(イミディエイト)
5	DR+GR[Rd]→GR[Rd]

フェッチ前半は、全命令共通にす

マイクロ操作は、次のステートに進む時一斉に実行される。

   (2) 表を簡単化する

   多少の例外を無視すると同じ制御手順になる命令は一つの表に統合する。例外の内容は表中に書いておく。((1)のアドレッシングモードのように)

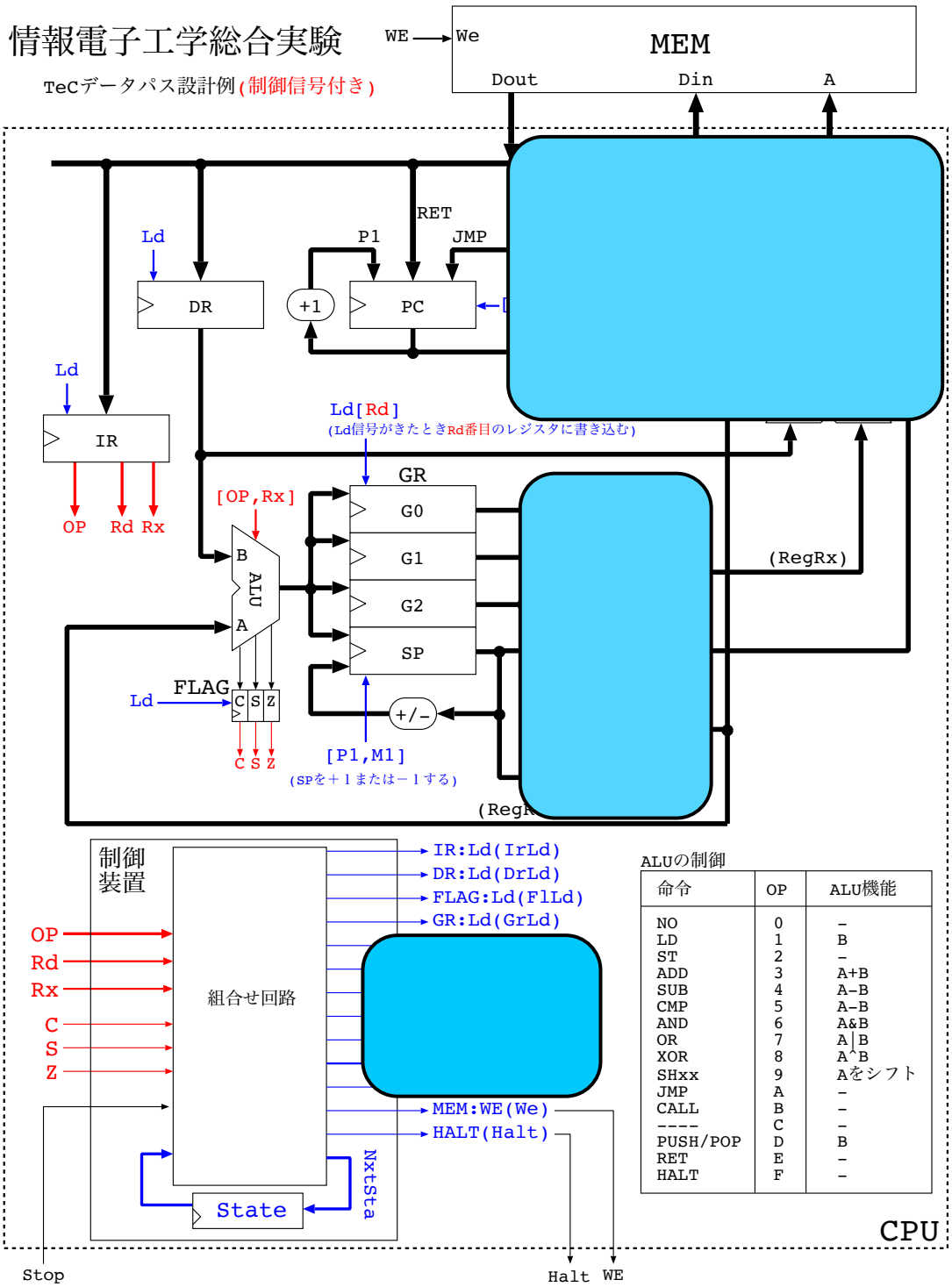
   (3) 表に制御線を追加する

ADD、SUB 命令						
ステート	マイクロ操作	Mar:Ld	Pc:++	Pc:Ld	Ir:Ld	...
0	PC→MAR, PC++	1	1			...
1	MEM→IR,PC→MAR	1			1	...

   データパスの設計が良ければ（？）、例外が制御線に影響しない。

   (4) 表の完成度が高くなったら**ステートマシン図**を作成する。その後は、表とステートマシン図の内容が乖離しないように気をつけながら完成度を増す。

参考：完成したデータパスの例



ALUの制御

命令	OP	ALU機能
NO	0	-
LD	1	B
ST	2	-
ADD	3	A+B
SUB	4	A-B
CMP	5	A-B
AND	6	A&B
OR	7	A|B
XOR	8	A^B
SHxx	9	Aをシフト
JMP	A	-
CALL	B	-
----	C	-
PUSH/POP	D	B
RET	E	-
HALT	F	-

赤色の信号は、制御装置以外から出力されるが制御に関係する。

青色の信号は、制御装置から出力される。

参考：制御手順の例

各命令の制御手順

命令フェッチ・デコード（各命令共通）															
ステート	マイクロ操作	IR:Ld	DR:Ld	FLG:Ld	GR:Ld	SP:P1	SP:M1	PC:P1	PC:Jmp	PC:Ret	MA	MD	MEM:WE	Halt	NxtSta
0	MEM[PC]→IR, PC++	1						1(*0)			PC				1(*1)
1	MEM[PC]→DR		1								PC				(*2)

LD,ADD,SUB,CMP,AND,OR,XOR命令															
ステート	マイクロ操作	IR:Ld	DR:Ld	FLG:Ld	GR:Ld	SP:P1	SP:M1	PC:P1	PC:Jmp	PC:Ret	MA	MD	MEM:WE	Halt	NxtSta
2	MEM[EA]→DR, PC++		1(*3)					1			EA				3

SHLA,SHLL,SHRA,SHRL命令															
ステート	マイクロ操作	IR:Ld	DR:Ld	FLG:Ld	GR:Ld	SP:P1	SP:M1	PC:P1	PC:Jmp	PC:Ret	MA	MD	MEM:WE	Halt	NxtSta
3	ALU→GR[Rd], Flag変化			1(*4)	1(*5)										0

ST命令															
ステート	マイクロ操作	IR:Ld	DR:Ld	FLG:Ld	GR:Ld	SP:P1	SP:M1	PC:P1	PC:Jmp	PC:Ret	MA	MD	MEM:WE	Halt	NxtSta
4	GR[Rd]→MEM[EA], PC++							1			EA	GR	1		0

JMP,JZ,JC,JM命令															
ステート	マイクロ操作	IR:Ld	DR:Ld	FLG:Ld	GR:Ld	SP:P1	SP:M1	PC:P1	PC:Jmp	PC:Ret	MA	MD	MEM:WE	Halt	NxtSta
5	PC++ or EA→PC							1(*6)	1(*7)						0

CALL命令															
ステート	マイクロ操作	IR:Ld	DR:Ld	FLG:Ld	GR:Ld	SP:P1	SP:M1	PC:P1	PC:Jmp	PC:Ret	MA	MD	MEM:WE	Halt	NxtSta
6	SP−−, PC++						1	1							7
7	EA→PC, PC→MEM[SP]								1		SP	PC	1		0

PUSH命令															
ステート	マイクロ操作	IR:Ld	DR:Ld	FLG:Ld	GR:Ld	SP:P1	SP:M1	PC:P1	PC:Jmp	PC:Ret	MA	MD	MEM:WE	Halt	NxtSta
8	SP−−						1								9
9	GR[Rd]→MEM[SP]										SP	GR	1		0

PUSH SPの動作が本物と異なる

POP命令															
ステート	マイクロ操作	IR:Ld	DR:Ld	FLG:Ld	GR:Ld	SP:P1	SP:M1	PC:P1	PC:Jmp	PC:Ret	MA	MD	MEM:WE	Halt	NxtSta
10	MEM[SP]→DR, SP++		1			1					SP				11
11	ALU→GR[Rd]				1										0

RET命令															
ステート	マイクロ操作	IR:Ld	DR:Ld	FLG:Ld	GR:Ld	SP:P1	SP:M1	PC:P1	PC:Jmp	PC:Ret	MA	MD	MEM:WE	Halt	NxtSta
12	MEM[SP]→PC, SP++					1				1	SP				0

HALT命令															
ステート	マイクロ操作	IR:Ld	DR:Ld	FLG:Ld	GR:Ld	SP:P1	SP:M1	PC:P1	PC:Jmp	PC:Ret	MA	MD	MEM:WE	Halt	NxtSta
13	HALT信号出力													1	0

*0: Stop=1の場合を除く
*1: Stop=1の場合は0
*2: 命令の種類による（命令デコード, 左表の通り）
*3: イミディエイトモードを除く
*4: LD命令を除く
*5: CMP命令を除く
*6: ジャンプ条件が成立しない場合のみ
*7: ジャンプ条件が成立した場合のみ

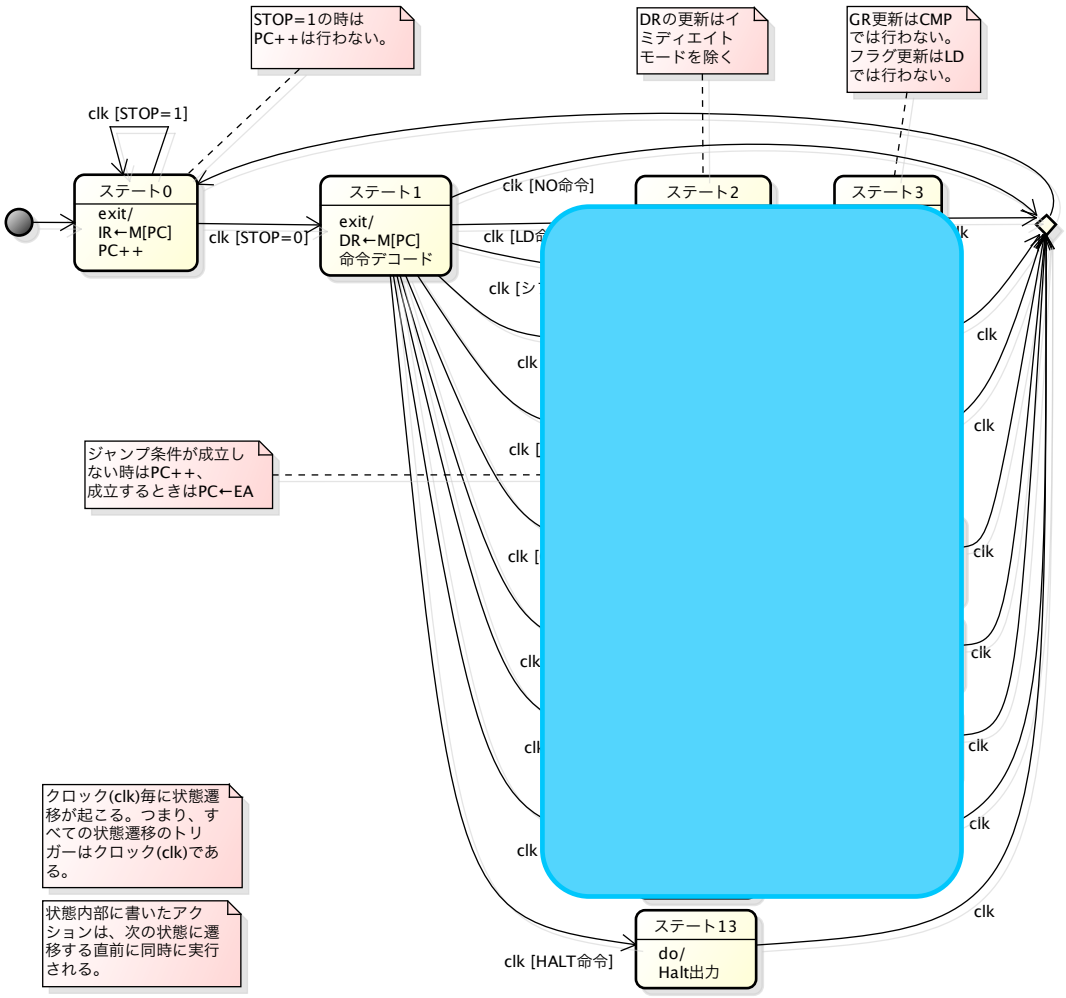
*2：命令の種類と次のステート	
命令	NxtSta
NO	0
LD,ADD,SUB,CMP,AND,OR,XOR	2
ST	4
SHRA,SHRL,SHLA,SHLL	3
JMP,JZ,JC,JM	5
CALL	6
PUSH	8
POP	10
RET	12
HALT	13

MA	
PC	00
EA	01
SP	10
禁止	11

MD	
PC	0
GR	1

例外は、＊印で示してある。

参考：ステートマシン図（レビュー会は，主にこの図を用いて進める）



遷移の条件は次の例のように読む。

clk	クロックがきたら無条件に遷移する。
clk[STOP=1]	クロックがきて、かつ、STOP が 1 なら遷移する。

上記の状態遷移図は astah*を使用して描いた。

draw.io を使用する場合は次の URL を参考にすると良い。

https://drawio-app.com/uml-state-diagrams-with-draw-io/

第２回レビュー会は、

(1) データパス（完成版）

(2) 各命令の制御手順表

(3) ステートマシン図

を対象に行う。