計算機論理設計

5. 計算機ハードウェアアーキテクチャと命令実行サイクル

一色 剛

工学院情報通信系

isshiki@ict.e.titech.ac.jp

16ビットマイクロプロセッサ命令セット構成

■ メモリ参照命令 :

- 命令中に12ビットアドレス値を含む
- アドレスモード : 直接アドレス、間接アドレス → 1ビットで表現
- 命令種別 : 残り3ビット(16 – 12 – 1 = 3)で表現
- メモリ参照命令（7個）: AND, ADD, LDA, STA, BUN, BSA, ISZ

■ 非メモリ参照命令:

- 命令中にアドレス値を含まず、他の付加情報も無い
- 命令種別 : 16ビットで表現可能 → ただし、「非メモリ参照命令」であることを表現する必要あり
- レジスタ参照命令（12個）: CLA, CLE, CMA, CME, CIR, CIL, INC, SPA, SNA, SZE, HLT
- 入出力命令（7個 → 詳細は後の講義で説明）: INP, OUT, SKI, SKO, ION, IOF, IMK

第1命令デコード回路

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I	OP1			ADI（アドレス値）											

メモリ参照命令
(OP1 ≠ 111)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I	1	1	1	OP2（オプコード2）											

非メモリ参照命令
(OP1 = 111)

OP1(2:0)	命令	デコード論理関数
000	AND	$O_{AND} = \overline{OP1(2)} \cdot \overline{OP1(1)} \cdot \overline{OP1(0)}$
001	ADD	$O_{ADD} = \overline{OP1(2)} \cdot \overline{OP1(1)} \cdot OP1(0)$
010	LDA	$O_{LDA} = \overline{OP1(2)} \cdot OP1(1) \cdot \overline{OP1(0)}$
011	STA	$O_{STA} = \overline{OP1(2)} \cdot OP1(1) \cdot OP1(0)$
100	BUN	$O_{BUN} = OP1(2) \cdot \overline{OP1(1)} \cdot \overline{OP1(0)}$
101	BSA	$O_{BSA} = OP1(2) \cdot \overline{OP1(1)} \cdot OP1(0)$
110	ISZ	$O_{ISZ} = OP1(2) \cdot OP1(1) \cdot \overline{OP1(0)}$
111	非メモリ参照命令	$O_{NM} = OP1(2) \cdot OP1(1) \cdot OP1(0)$

第2命令デコード回路

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I	1	1	1	OP2（オプコード2）											

非メモリ参照命令
(OP1 = 111)

レジスタ参照命令(I = 0)

OP2	命令	デコード論理関数
1000 0000 0000	CLA	$O_{CLA} = O_{NM} \cdot \bar{I} \cdot OP2(11)$
0100 0000 0000	CLE	$O_{CLE} = O_{NM} \cdot \bar{I} \cdot OP2(10)$
0010 0000 0000	CMA	$O_{CMA} = O_{NM} \cdot \bar{I} \cdot OP2(9)$
0001 0000 0000	CME	$O_{CME} = O_{NM} \cdot \bar{I} \cdot OP2(8)$
0000 1000 0000	CIR	$O_{CIR} = O_{NM} \cdot \bar{I} \cdot OP2(7)$
0000 0100 0000	CIL	$O_{CIL} = O_{NM} \cdot \bar{I} \cdot OP2(6)$
0000 0010 0000	INC	$O_{INC} = O_{NM} \cdot \bar{I} \cdot OP2(5)$
0000 0001 0000	SPA	$O_{SPA} = O_{NM} \cdot \bar{I} \cdot OP2(4)$
0000 0000 1000	SNA	$O_{SNA} = O_{NM} \cdot \bar{I} \cdot OP2(3)$
0000 0000 0100	SZA	$O_{SZA} = O_{NM} \cdot \bar{I} \cdot OP2(2)$
0000 0000 0010	SZE	$O_{SZE} = O_{NM} \cdot \bar{I} \cdot OP2(1)$
0000 0000 0001	HLT	$O_{HLT} = O_{NM} \cdot \bar{I} \cdot OP2(0)$

第2命令デコード回路

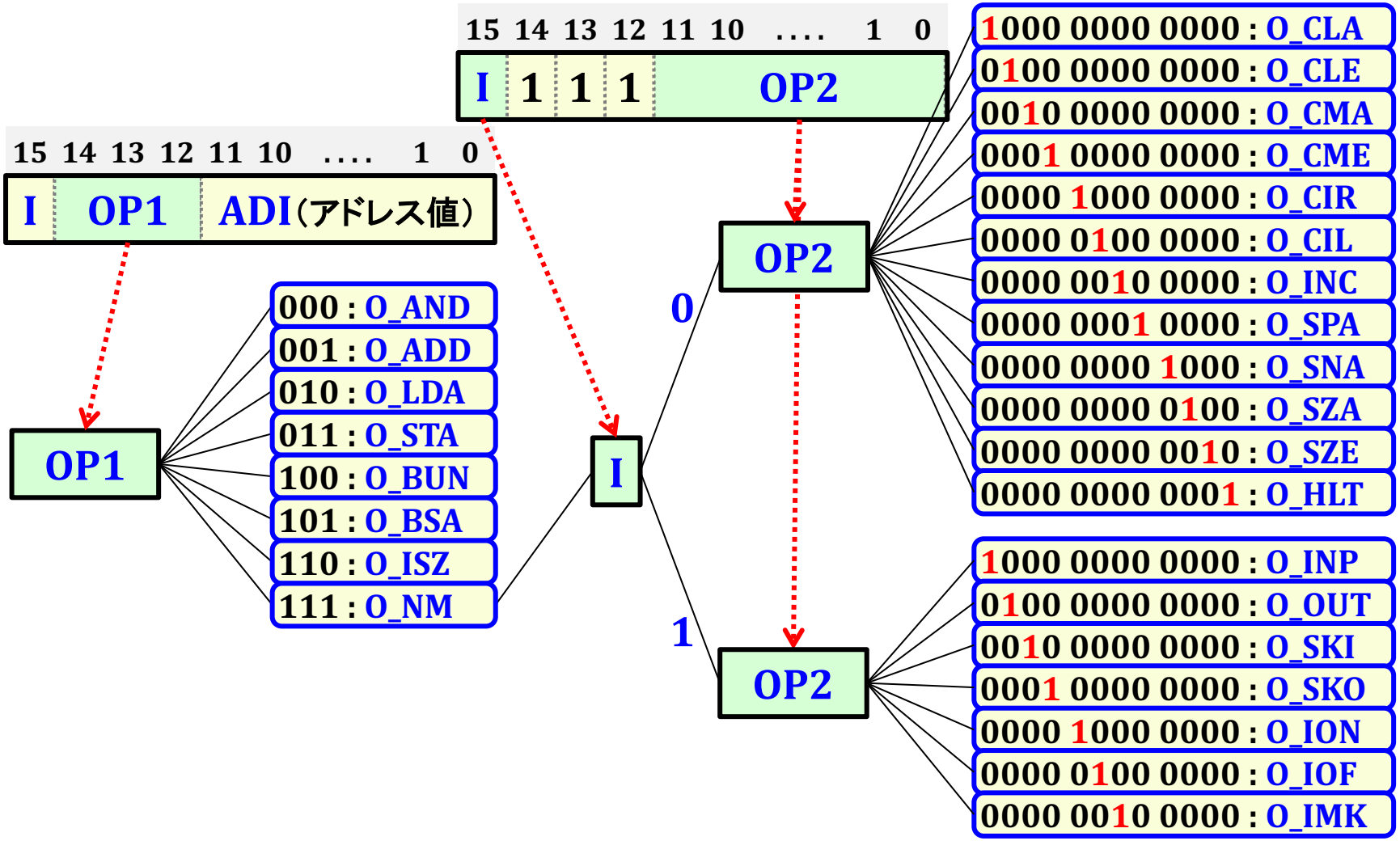
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I	1	1	1	OP2（オプコード2）											

非メモリ参照命令
(OP1 = 111)

入出力命令(I = 1)

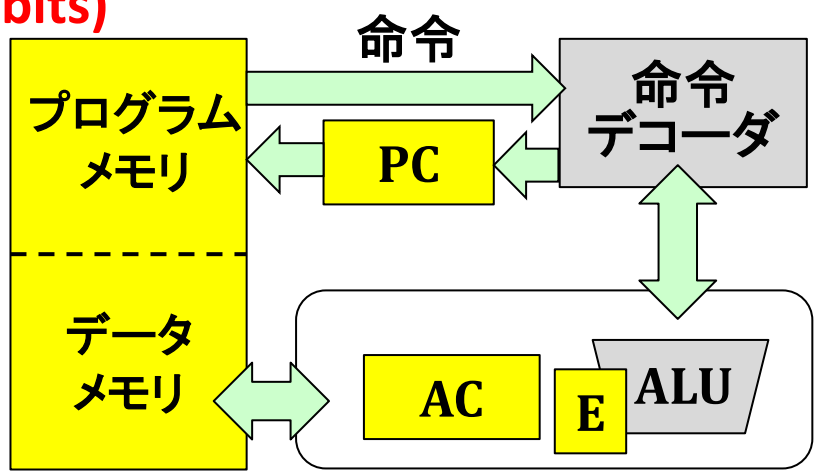
OP2	命令	デコード論理関数
1000 0000 0000	INP	$O_{INP} = O_{NM} \cdot I \cdot OP2(11)$
0100 0000 0000	OUT	$O_{CLE} = O_{NM} \cdot I \cdot OP2(10)$
0010 0000 0000	SKI	$O_{CMA} = O_{NM} \cdot I \cdot OP2(9)$
0001 0000 0000	SKO	$O_{CME} = O_{NM} \cdot I \cdot OP2(8)$
0000 1000 0000	ION	$O_{CIR} = O_{NM} \cdot I \cdot OP2(7)$
0000 0100 0000	IOF	$O_{CIL} = O_{NM} \cdot I \cdot OP2(6)$
0000 0010 0000	IMK	$O_{INC} = O_{NM} \cdot I \cdot OP2(5)$

命令デコード木


15 14 13 12 11 10 .... 1 0
I 1 1 1 OP2
15 14 13 12 11 10 .... 1 0
I OP1 ADI（アドレス値）
OP1
000 : O_AND
001 : O_ADD
010 : O_LDA
011 : O_STA
100 : O_BUN
101 : O_BSA
110 : O_ISZ
111 : O_NM
I
0
1
OP2
OP2
1000 0000 0000 : O_CLA
0100 0000 0000 : O_CLE
0010 0000 0000 : O_CMA
0001 0000 0000 : O_CME
0000 1000 0000 : O_CIR
0000 0100 0000 : O_CIL
0000 0010 0000 : O_INC
0000 0001 0000 : O_SPA
0000 0000 1000 : O_SNA
0000 0000 0100 : O_SZA
0000 0000 0010 : O_SZE
0000 0000 0001 : O_HLT
1000 0000 0000 : O_INP
0100 0000 0000 : O_OUT
0010 0000 0000 : O_SKI
0001 0000 0000 : O_SKO
0000 1000 0000 : O_ION
0000 0100 0000 : O_IOF
0000 0010 0000 : O_IMK

命令セット定義における計算機構成図

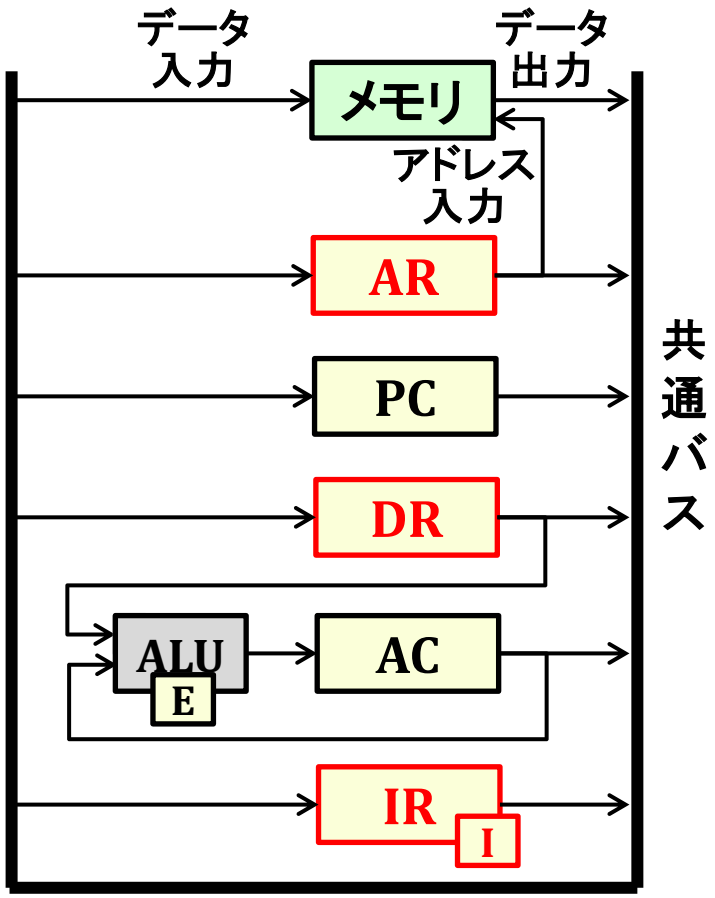
- 命令オペランド : 命令が参照・代入するレジスタやメモリ
  - メモリ : アドレス(12 bits)、データ(16 bits)
  - アキュムレータ (AC: 16 bits)
  - プログラムカウンタ (PC : 12 bits)
  - 状態レジスタ (E)
- 命令形式: 16ビット命令
  - メモリ参照命令
    - 直接アドレス
    - 間接アドレス
  - レジスタ参照命令



命令動作解析上必要な構成要素
→ 論理回路として命令動作を実装するためには更に詳細な計算機構成図が必要

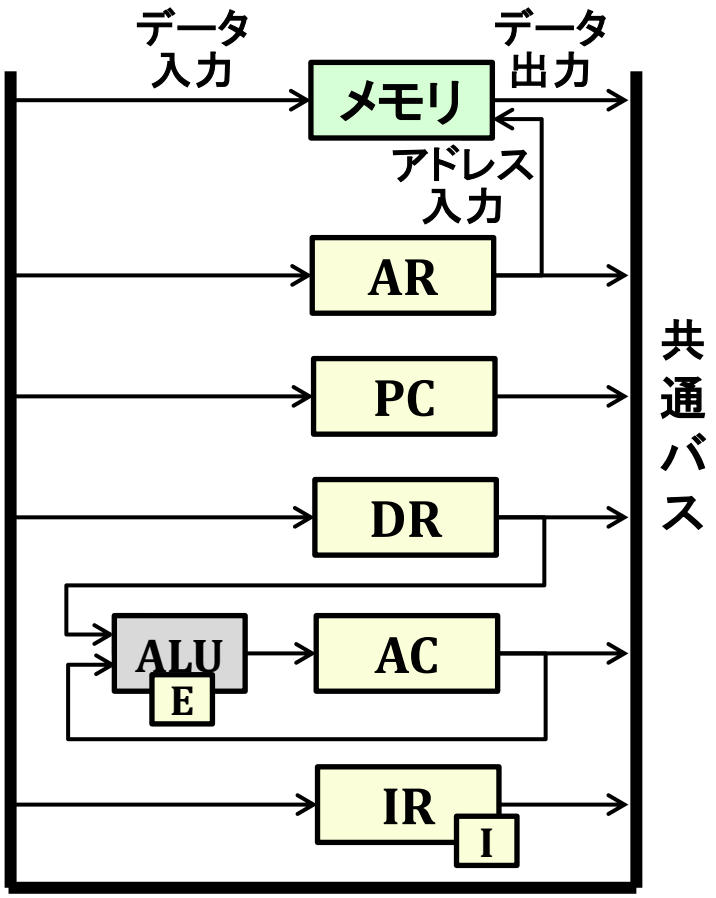
命令実行制御のための計算機詳細構成図（計算機ハードウェアアーキテクチャ）

- メモリ
  - アドレス : 12 bits
  - データ : 16 bits
- アドレスレジスタ (AR : 12 bits)
- プログラムカウンタ (PC : 12 bits)
- データレジスタ (DR : 16 bits)
- アキュムレータ (AC : 16 bits)
- 算術論理演算器 (ALU : 組合せ回路)
- 命令レジスタ (IR : 16 bits)
- 状態レジスタ (E : 1 bit)
- 命令フラグレジスタ (I : 1 bit)
- 共通バス (組合せ回路)



命令実行サイクル

- ■ 計算機構成要素（メモリ、レジスタ、ALU、共通バス）間のデータ転送を順序回路で制御し、以下の動作を実現
- 命令フェッチ：命令読出し
- 命令デコード：計算機の状態遷移の決定
- 命令実行：メモリアクセス、演算、レジスタ更新
- PC更新：次命令のアドレス計算
- ■ データ転送の制約
- メモリアドレス：AR出力が直接接続
- 共通バス出力：共通バスに出力するメモリ（データ出力）, AR, PC, DR, AC, IRから一つ選択し、その信号をバスに転送する
- 共通バス入力：共通バスから入力するメモリ（データ入力）, AR, PC, DR, IRはバス信号を常に受け取ることができる
- AC入力：ACは共通バスからの直接の入力はなく、ALUの出力信号が入力に接続される



命令フェッチサイクル（①～③サイクル）

■ 命令フェッチサイクル：各命令実行の最初の状態遷移

① $AR \leftarrow PC$

- 次命令アドレス値をPCからARに転送

② $IR \leftarrow M[AR], PC \leftarrow PC + 1$

- 次命令をメモリから読出し、IR（命令レジスタ）に転送
- PCをインクリメント

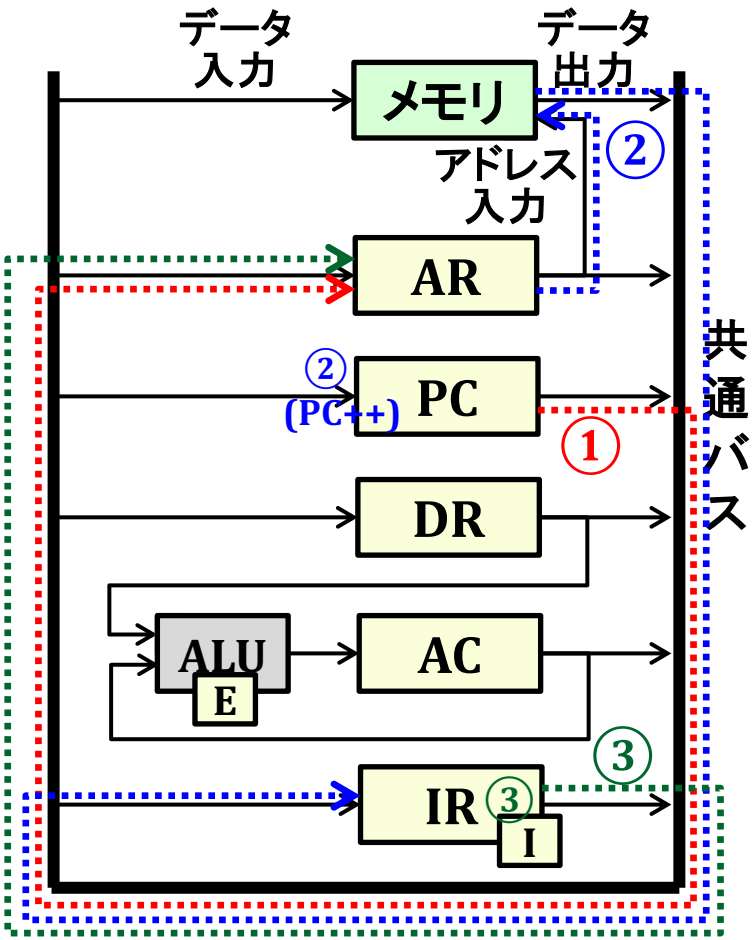
③ $I \leftarrow IR(15), AR \leftarrow IR(11{:}0)$

- 命令レジスタから対応するビットをＩとARに転送（間接アドレスメモリ参照命令の準備）

■ 第1命令デコード　OP1 = IR(14:12)

- OP1 = 111 : 非メモリ参照命令サイクル
- OP1 ≠ 111 : メモリ参照命令サイクル

15	14	13	12	11	10	....	1	0
I	OP1			ADI（アドレス値）				



レジスタ参照命令実行サイクル（④サイクル）

I = IR(15), OP1 = IR(14:12), OP2 = IR(11:0)

15	14	13	12	11 10 .... 1 0
I	1	1	1	OP2

■ レジスタ参照命令判定 : (OP1 = 111) & (I = 0)

OP2	命令	動作（④サイクル）
1000 0000 0000	CLA	$AC \leftarrow 0$
0100 0000 0000	CLE	$E \leftarrow 0$
0010 0000 0000	CMA	$AC \leftarrow \overline{AC}$
0001 0000 0000	CME	$E \leftarrow \overline{E}$
0000 1000 0000	CIR	$AC(14:0) \leftarrow AC(15:1), AC(15) \leftarrow E, E \leftarrow AC(0)$
0000 0100 0000	CIL	$AC(15:1) \leftarrow AC(14:0), AC(0) \leftarrow E, E \leftarrow AC(15)$
0000 0010 0000	INC	$AC \leftarrow AC + 1$
0000 0001 0000	SPA	$IF\ (AC(15) = 0)\ THEN\ PC \leftarrow PC + 1$
0000 0000 1000	SNA	$IF\ (AC(15) = 1)\ THEN\ PC \leftarrow PC + 1$
0000 0000 0100	SZA	$IF\ (AC = 0)\ THEN\ PC \leftarrow PC + 1$
0000 0000 0010	SZE	$IF\ (E = 0)\ THEN\ PC \leftarrow PC + 1$
0000 0000 0001	HLT	$HALT$

メモリ参照命令実行サイクル（④サイクル）

I = IR(15), OP1 = IR(14:12)

■ 命令フェッチの最終サイクル

③ $I \leftarrow IR(15), AR \leftarrow IR(11:0)$

■ 間接アドレス判定・実行サイクル

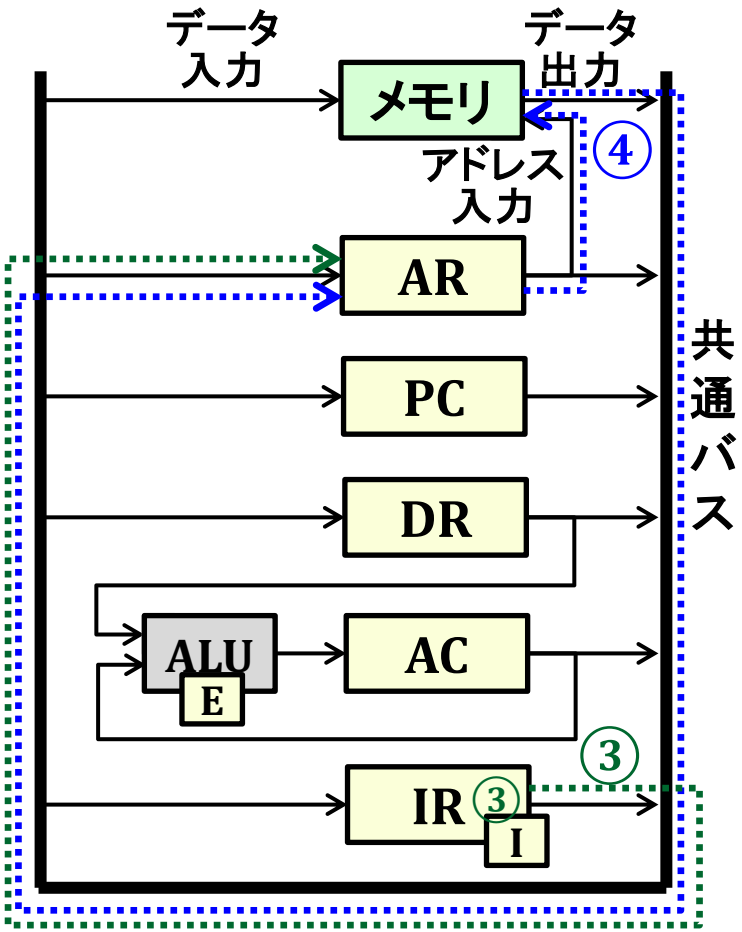
④ （間接アドレス：$I = 1$）$AR \leftarrow M[AR]$
（直接アドレス：$I = 0$）無操作

→ この段階で間接アドレスと直接アドレスの違いが吸収される

⑤～⑦：各メモリ参照命令実行サイクル

ADD A　：$AC \leftarrow AC + M[A]$（直接アドレス）

ADD B I：$AC \leftarrow AC + M[M[B]]$（間接アドレス）



メモリ参照命令実行サイクル（⑤～⑦サイクル）

I = IR(15), OP1 = IR(14:12)

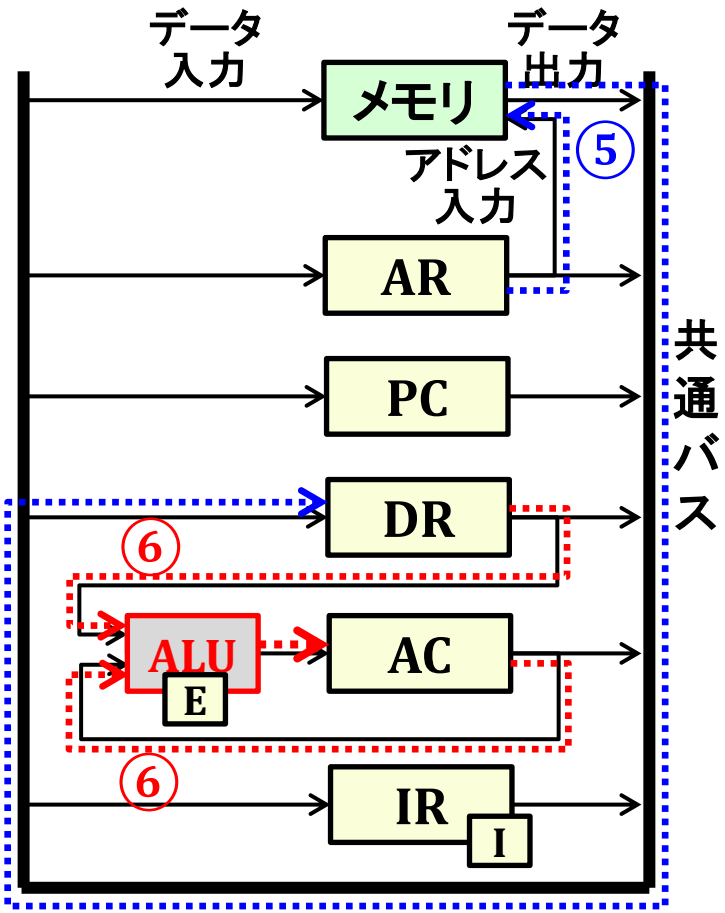
OP1	命令	動作（⑤～⑦サイクル）
000	AND	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow AC \,\&\, DR$
001	ADD	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow AC + DR, E \leftarrow C_{out}(16)$
010	LDA	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow DR$
011	STA	⑤ $M[AR] \leftarrow AC$
100	BUN	⑤ $PC \leftarrow AR$
101	BSA	⑤ $M[AR] \leftarrow PC, AR \leftarrow AR + 1$ ⑥ $PC \leftarrow AR$
110	ISZ	⑤ $DR \leftarrow M[AR]$ ⑥ $DR \leftarrow DR + 1$ ⑦ $M[AR] \leftarrow DR$, $IF(DR = 0)\ THEN\ PC \leftarrow PC + 1$

命令	（命令セット動作定義） レジスタ転送記述
AND	$AC \leftarrow AC \,\&\, M[ad]$
ADD	$AC \leftarrow AC + M[ad]$, $E \leftarrow C_{out}(16)$
LDA	$AC \leftarrow M[ad]$
STA	$M[ad] \leftarrow AC$
BUN	$PC \leftarrow ad$
BSA	$M[ad] \leftarrow PC, PC \leftarrow ad + 1$
ISZ	$M[ad] \leftarrow M[ad] + 1$, $IF(M[ad] + 1 = 0)\ THEN$ $PC \leftarrow PC + 1$

メモリ参照命令実行サイクル（⑤～⑦サイクル）

I = IR(15), OP1 = IR(14:12)

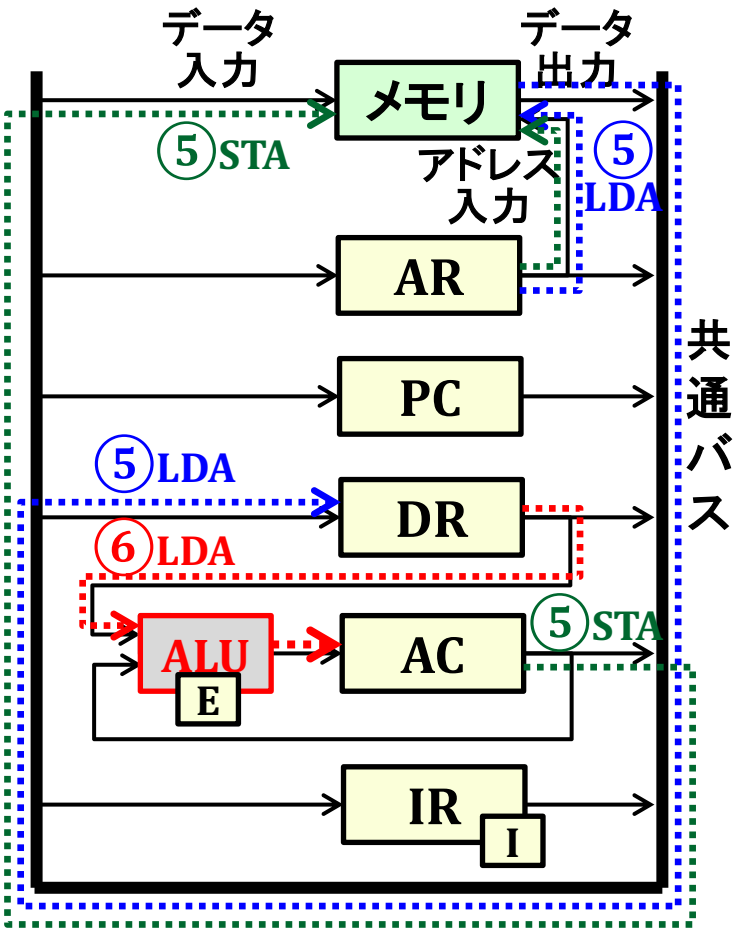
OP1	命令	動作（⑤～⑦サイクル）
000	AND	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow AC \,\&\, DR$
001	ADD	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow AC + DR, E \leftarrow C_{out}(16)$
010	LDA	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow DR$
011	STA	⑤ $M[AR] \leftarrow AC$
100	BUN	⑤ $PC \leftarrow AR$
101	BSA	⑤ $M[AR] \leftarrow PC, AR \leftarrow AR + 1$ ⑥ $PC \leftarrow AR$
110	ISZ	⑤ $DR \leftarrow M[AR]$ ⑥ $DR \leftarrow DR + 1$ ⑦ $M[AR] \leftarrow DR$, $IF(DR = 0)\ THEN\ PC \leftarrow PC + 1$



メモリ参照命令実行サイクル（⑤～⑦サイクル）

I = IR(15), OP1 = IR(14:12)

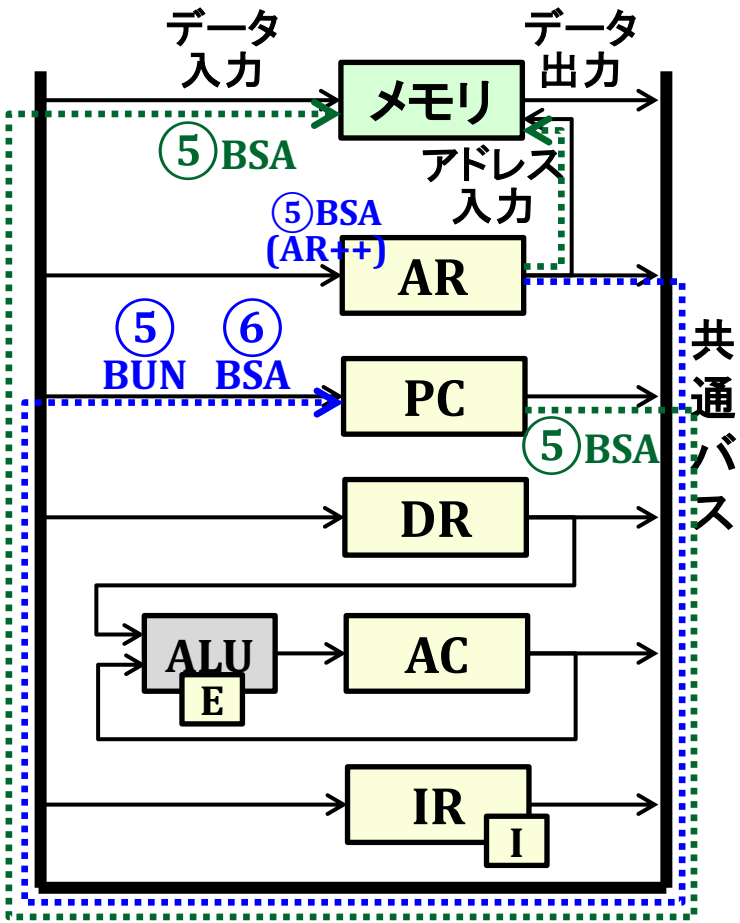
OP1	命令	動作（⑤～⑦サイクル）
000	AND	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow AC \,\&\, DR$
001	ADD	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow AC + DR, E \leftarrow C_{out}(16)$
010	LDA	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow DR$
011	STA	⑤ $M[AR] \leftarrow AC$
100	BUN	⑤ $PC \leftarrow AR$
101	BSA	⑤ $M[AR] \leftarrow PC, AR \leftarrow AR + 1$ ⑥ $PC \leftarrow AR$
110	ISZ	⑤ $DR \leftarrow M[AR]$ ⑥ $DR \leftarrow DR + 1$ ⑦ $M[AR] \leftarrow DR$, $IF(DR = 0)\ THEN\ PC \leftarrow PC + 1$



メモリ参照命令実行サイクル（⑤～⑦サイクル）

I = IR(15), OP1 = IR(14:12)

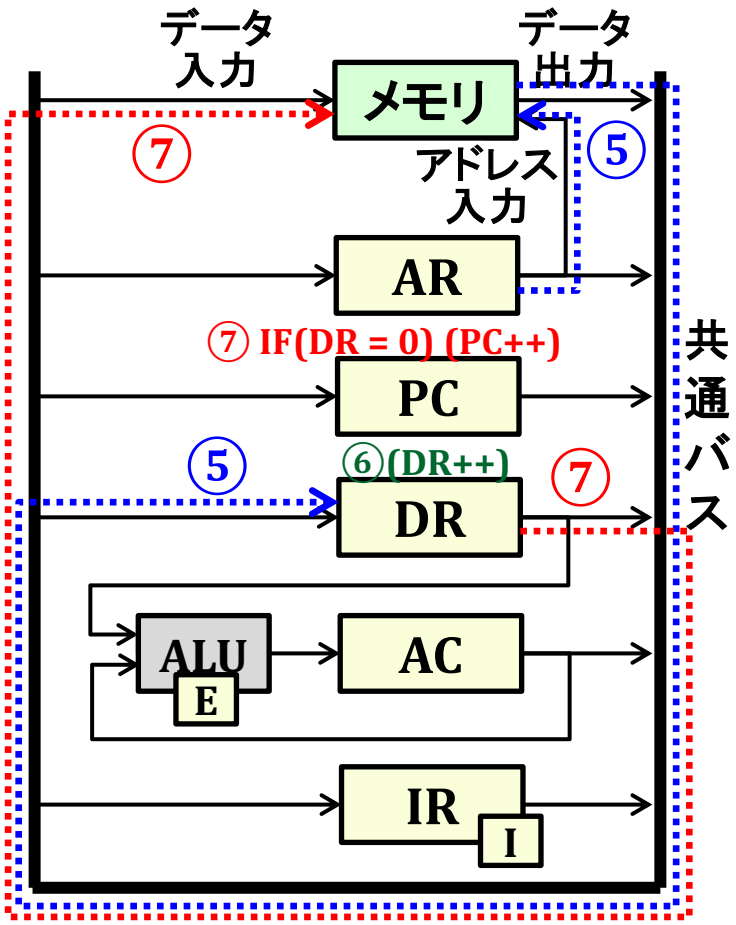
OP1	命令	動作（⑤～⑦サイクル）
000	AND	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow AC \,\&\, DR$
001	ADD	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow AC + DR, E \leftarrow C_{out}(16)$
010	LDA	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow DR$
011	STA	⑤ $M[AR] \leftarrow AC$
100	BUN	⑤ $PC \leftarrow AR$
101	BSA	⑤ $M[AR] \leftarrow PC, AR \leftarrow AR + 1$ ⑥ $PC \leftarrow AR$
110	ISZ	⑤ $DR \leftarrow M[AR]$ ⑥ $DR \leftarrow DR + 1$ ⑦ $M[AR] \leftarrow DR$, $IF(DR = 0)\ THEN\ PC \leftarrow PC + 1$



メモリ参照命令実行サイクル（⑤～⑦サイクル）

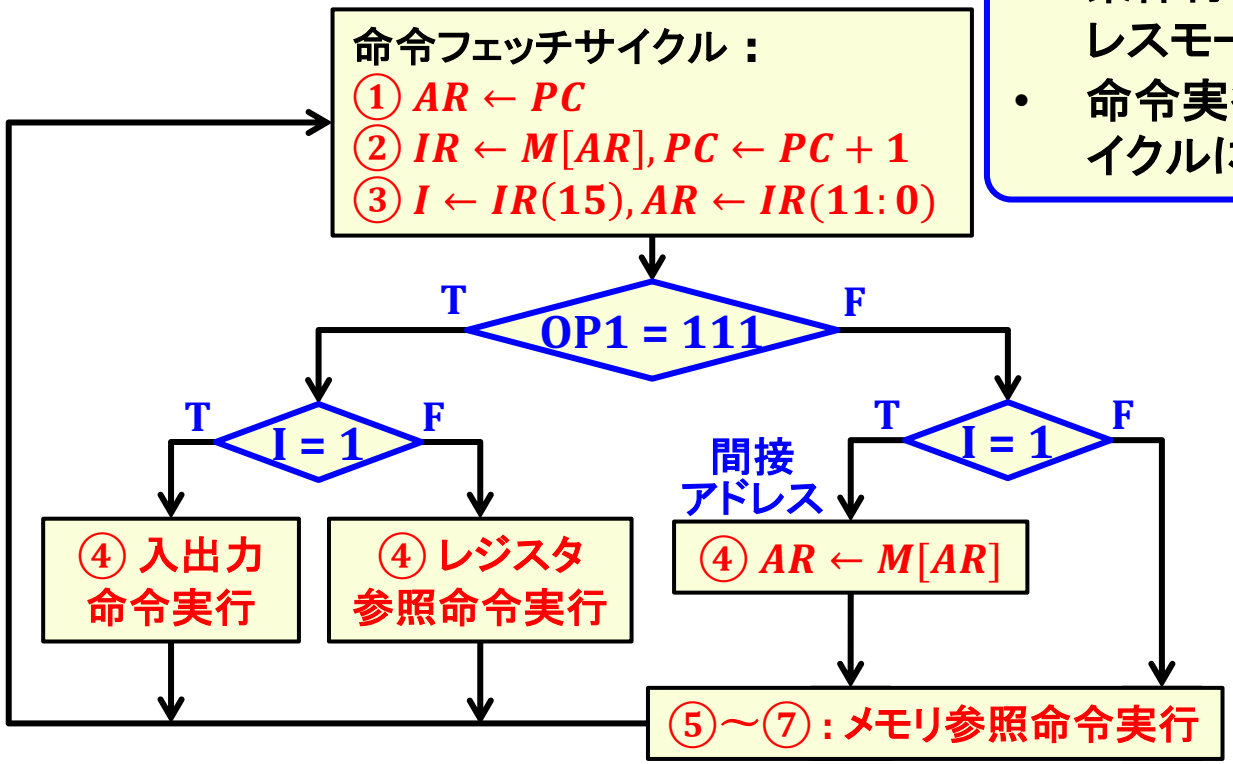
I = IR(15), OP1 = IR(14:12)

OP1	命令	動作（⑤～⑦サイクル）
000	AND	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow AC \,\&\, DR$
001	ADD	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow AC + DR, E \leftarrow C_{out}(16)$
010	LDA	⑤ $DR \leftarrow M[AR]$ ⑥ $AC \leftarrow DR$
011	STA	⑤ $M[AR] \leftarrow AC$
100	BUN	⑤ $PC \leftarrow AR$
101	BSA	⑤ $M[AR] \leftarrow PC, AR \leftarrow AR + 1$ ⑥ $PC \leftarrow AR$
110	ISZ	⑤ $DR \leftarrow M[AR]$ ⑥ $DR \leftarrow DR + 1$ ⑦ $M[AR] \leftarrow DR$, $IF(DR = 0)\ THEN\ PC \leftarrow PC + 1$



命令実行状態遷移

I = IR(15), OP1 = IR(14:12), OP2 = IR(11:0)



命令実行状態遷移の特徴

- 条件付き状態遷移：命令種、アドレスモード等 → 制御フローの分岐
- 命令実行終了後、命令フェッチサイクルに戻る → 制御フローの合流

命令実行状態表現

命令実行状態遷移の特徴

- 条件付き状態遷移：命令種、アドレスモード等 → 制御フローの分岐
- 命令実行終了後、命令フェッチサイクルに戻る → 制御フローの合流

一般的な順序回路の状態遷移よりも簡単な遷移構造のため、順序回路で一般に使われる「状態変数」ではなく、「タイミング情報」と「命令情報」を表す論理変数の「組」として命令実行状態を表現し、論理回路設計を単純化することを考える

■ **タイミング情報**：命令フェッチサイクル開始時からのクロック数

- 命令フェッチサイクル：①～③サイクル
- 命令実行サイクル：④～⑦サイクル

→ タイミング生成回路（3ビットカウンタ）

■ **命令情報**：命令レジスタの各ビット

- I ← IR(15)：アドレスモード（メモリ参照）、レジスタ参照・入出力命令判別
- OP1 = IR(14:12)：メモリ参照命令・非メモリ参照命令判別
- OP2 = IR(11:0)：非メモリ参照命令種判別

→ デコード回路

タイミング生成回路 : 3ビットカウンタ

- 入力 : clk（クロック）, clr（クリア）
- 出力 : q （3ビットカウンタ値）

clr	$q^{(1)}$ (qの次状態)
1	0
0	q + 1

インクリメント（+1）回路 : 半加算器 (Half-Adder)

- 入力 : a_i（データ入力）, c_i（キャリー入力）
- 出力 : s_i（データ出力）, c_{i+1}（キャリー出力）
- 入出力関係 : $a_i + c_i = 2 * c_{i+1} + s_i$

（+, * : 算術加算、算術乗算 → 入力の「1」の個数を2進数で表す）

講義資料1 : P10 (Full-Adder)参照

入力		出力	
a_i	c_i	c_{i+1}	s_i
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

- $c_0 = 1$
- $s_i = a_i \oplus c_i$
- $c_{i+1} = a_i \cdot c_i$

```
   01110  (carry)
   00111  (7)
+  00001  (1)
---------------
   01000  (8)
```

```
   01111  (1+carry)
+  00111  (7)
---------------
   01000  (8)
```

N ビット加算器 → インクリメント(+1)に限定

講義資料1：P11 参照

Nビット加算器：N個の1ビット加算器（Full-Adder）を従属接続

carry-bit

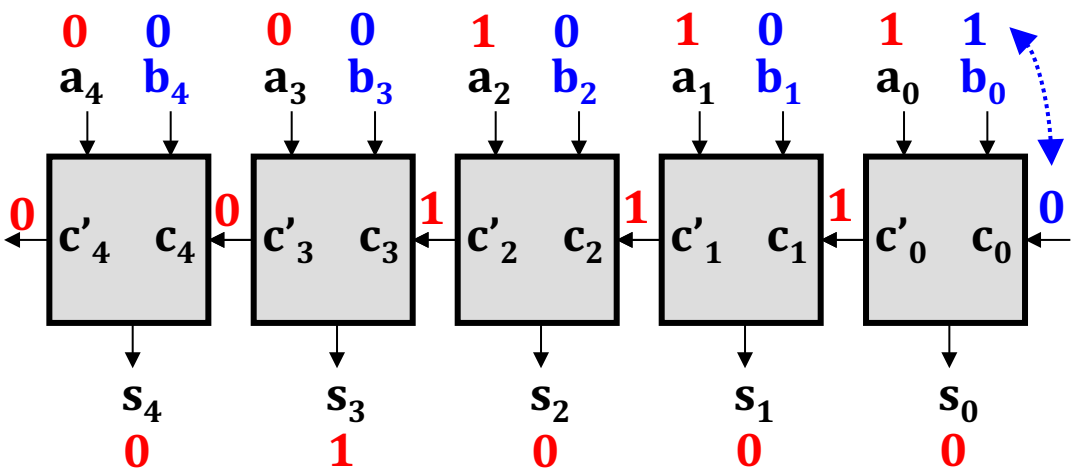
```
   0        1        1        1        0
 00111    00111    00111    00111    00111
+ 00001  + 00001  + 00001  + 00001  + 00001
 -----    -----    -----    -----    -----
   10       10       10       01       00
 -----    -----    -----    -----    -----
    0       00       000     1000    01000
```

sum-bit　　result

```
  01110  (carry)
  00111  (7)
+ 00001  (1)
 -------
  01000  (8)
```

```
  01111  (1+carry)
+ 00111  (7)
 -------
  01000  (8)
```

+1加算を0ビット目のキャリー入力に入れれば、b入力は全て0になる



Full-Adder

a	b	c	c'	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Half-Adder

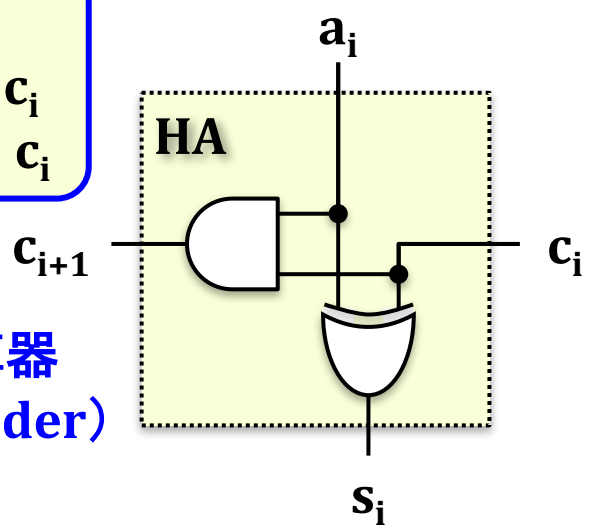
a_i	c_i	c_{i+1}	s_i
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

タイミング生成回路：3ビットカウンタ

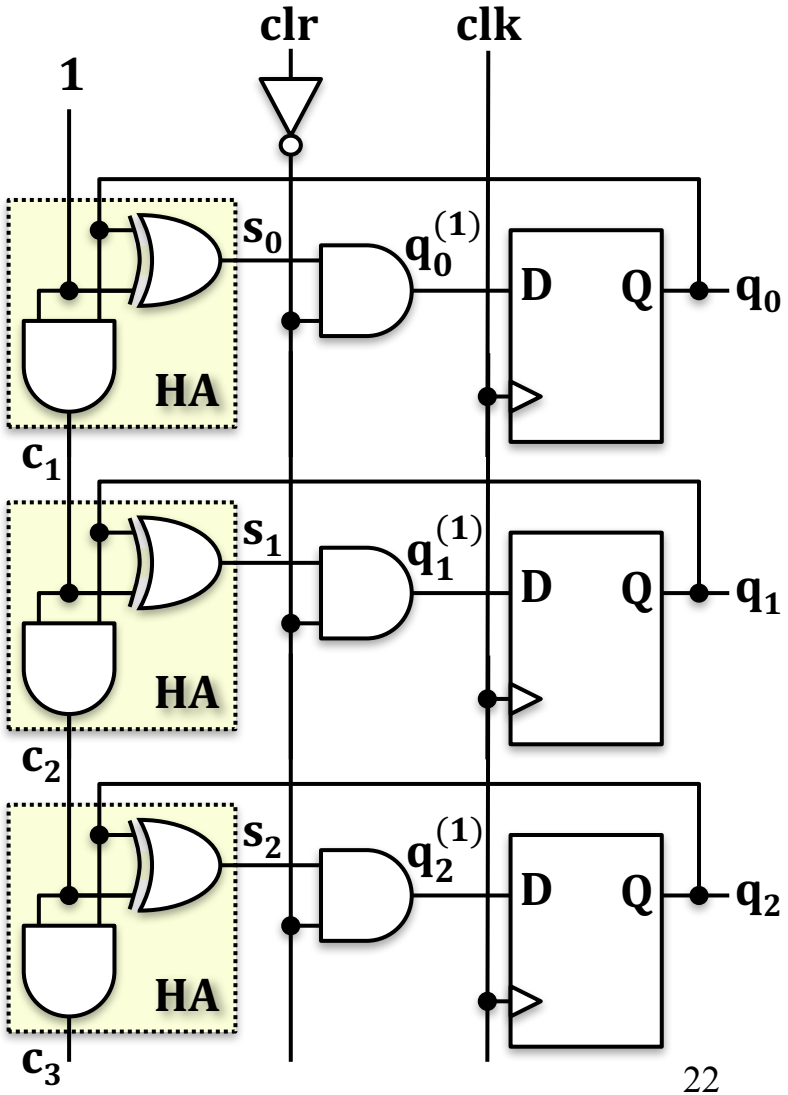
- 入力：clk（クロック），clr（クリア）
- 出力：q（3ビットカウンタ値）

clr	$q^{(1)}$ (qの次状態)
1	0
0	q + 1

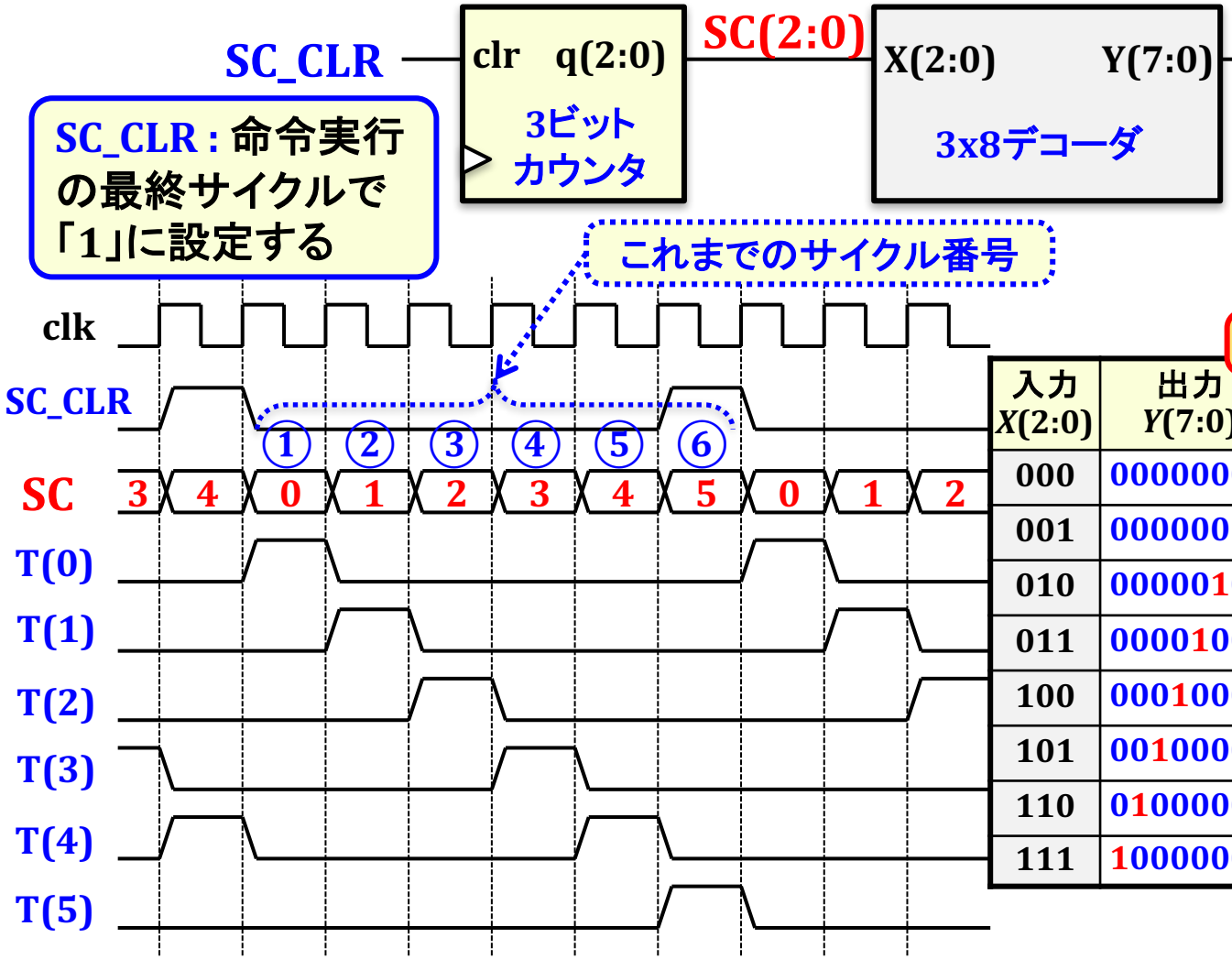
- $c_0 = 1$
- $s_i = a_i \oplus c_i$
- $c_{i+1} = a_i \cdot c_i$



半加算器
（Half-Adder）



タイミング情報 : 3ビットカウンタ → 3x8デコーダ



カウンタ値 SC(2:0) を符号化し8つの論理変数 T(7:0) に変換

講義資料4 : P5 参照

入力 $X(2:0)$	出力 $Y(7:0)$	論理関数
000	00000001	$Y(0) = \overline{X(2)} \cdot \overline{X(1)} \cdot \overline{X(0)}$
001	00000010	$Y(1) = \overline{X(2)} \cdot \overline{X(1)} \cdot X(0)$
010	00000100	$Y(2) = \overline{X(2)} \cdot X(1) \cdot \overline{X(0)}$
011	00001000	$Y(3) = \overline{X(2)} \cdot X(1) \cdot X(0)$
100	00010000	$Y(4) = X(2) \cdot \overline{X(1)} \cdot \overline{X(0)}$
101	00100000	$Y(5) = X(2) \cdot \overline{X(1)} \cdot X(0)$
110	01000000	$Y(6) = X(2) \cdot X(1) \cdot \overline{X(0)}$
111	10000000	$Y(7) = X(2) \cdot X(1) \cdot X(0)$

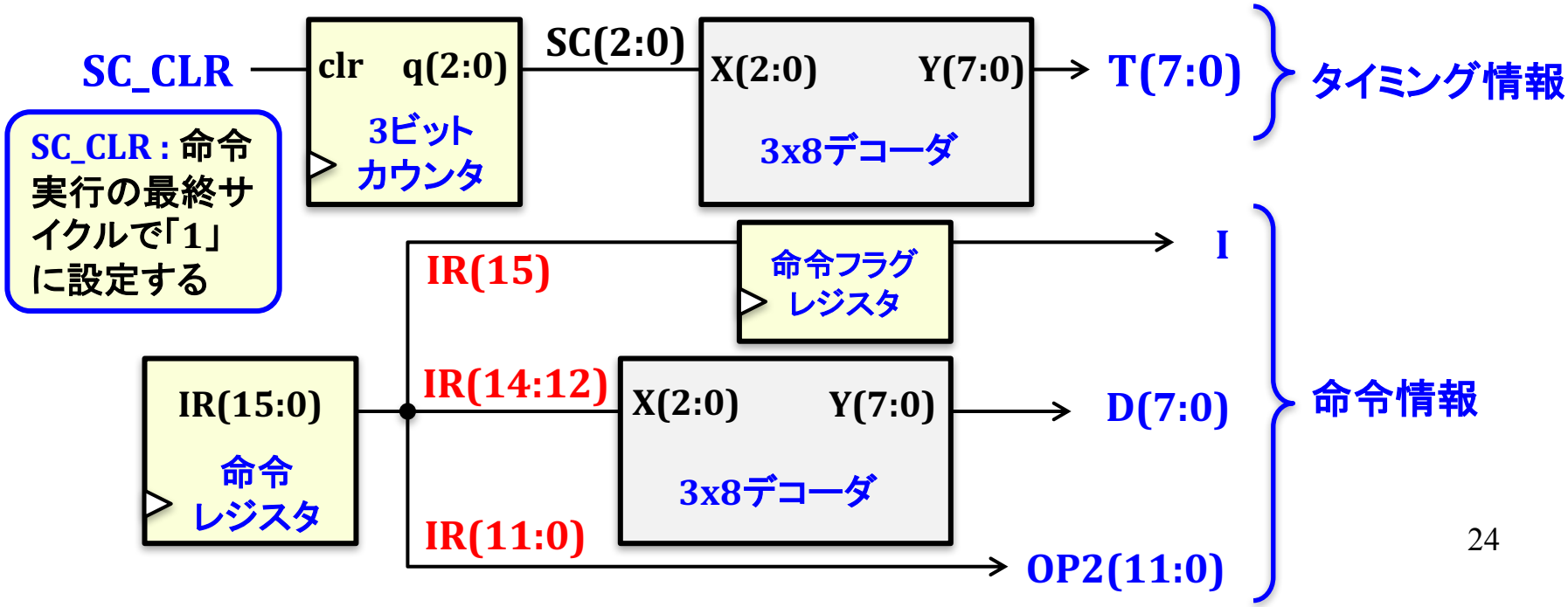
命令実行状態表現

- **タイミング情報 :** 命令フェッチサイクル開始時からのクロック数
  - 命令フェッチサイクル : ①～③サイクル
  - 命令実行サイクル : ④～⑦サイクル

→ タイミング生成回路（3ビットカウンタ）

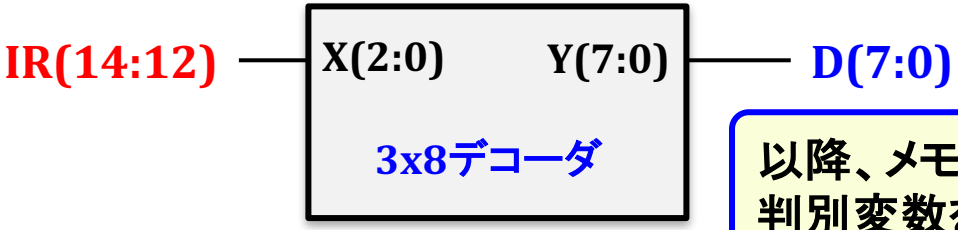
- **命令情報 :** 命令レジスタの各ビット
  - I ← IR(15) : アドレスモード（メモリ参照）、レジスタ参照・入出力命令判別
  - OP1 = IR(14:12) : メモリ参照命令・非メモリ参照命令判別
  - OP2 = IR(11:0) : 非メモリ参照命令種判別

→ デコード回路



命令実行状態表現

OP1(2:0)	命令	デコード論理関数
000	AND	$D(0) = O_{AND} = \overline{OP1(2)} \cdot \overline{OP1(1)} \cdot \overline{OP1(0)}$
001	ADD	$D(1) = O_{ADD} = \overline{OP1(2)} \cdot \overline{OP1(1)} \cdot OP1(0)$
010	LDA	$D(2) = O_{LDA} = \overline{OP1(2)} \cdot OP1(1) \cdot \overline{OP1(0)}$
011	STA	$D(3) = O_{STA} = \overline{OP1(2)} \cdot OP1(1) \cdot OP1(0)$
100	BUN	$D(4) = O_{BUN} = OP1(2) \cdot \overline{OP1(1)} \cdot \overline{OP1(0)}$
101	BSA	$D(5) = O_{BSA} = OP1(2) \cdot \overline{OP1(1)} \cdot OP1(0)$
110	ISZ	$D(6) = O_{ISZ} = OP1(2) \cdot OP1(1) \cdot \overline{OP1(0)}$
111	非メモリ参照命令	$D(7) = O_{NM} = OP1(2) \cdot OP1(1) \cdot OP1(0)$



以降、メモリ参照命令の判別変数をD(7:0)とする

レジスタ転送記述による命令実行動作記述

レジスタ転送記述形式

＜条件式＞：＜レジスタ転送式＞

- 条件式：＜レジスタ転送式＞を実行するための条件を表した論理式
- レジスタ転送式：レジスタ・メモリへの代入式（コンマ「，」で複数の式を連結）

命令フェッチサイクル：①～③

$T(0): AR \leftarrow PC$

$T(1): IR \leftarrow M[AR], PC \leftarrow PC + 1$

$T(2): I \leftarrow IR(15), AR \leftarrow IR(11:0)$

- 3ビットカウンタ(SC)へのレジスタ転送式は、リセット動作のみを明記
- 明記されていない時の動作は

  SC ← SC + 1

レジスタ参照命令実行サイクル：④

$r = D(7) \cdot \bar{I} \cdot T(3)$ とする

$r: SC \leftarrow 0$ (3ビットカウンタのリセット)

$r \cdot OP2(11): AC \leftarrow 0$ (CLA)

$r \cdot OP2(10): E \leftarrow 0$ (CLE)

$r \cdot OP2(9): AC \leftarrow \overline{AC}$ (CMA)

$r \cdot OP2(8): E \leftarrow \bar{E}$ (CME)

. . . . . .

レジスタ転送記述による命令実行動作記述

メモリ参照命令実行サイクル：④～⑦

$\overline{D(7)} \cdot I \cdot T(3) : AR \leftarrow M[AR]$ （④：間接アドレスサイクル）

$D(0) \cdot T(4) : DR \leftarrow M[AR]$ （⑤：AND）

$D(0) \cdot T(5) : AC \leftarrow AC \,\&\, DR, \boxed{SC \leftarrow 0}$ （⑥：AND）

$D(1) \cdot T(4) : DR \leftarrow M[AR]$ （⑤：ADD）

$D(1) \cdot T(5) : AC \leftarrow AC + DR, E \leftarrow C_{out}(16), \boxed{SC \leftarrow 0}$ （⑥：ADD）

$D(2) \cdot T(4) : DR \leftarrow M[AR]$ （⑤：LDA）

$D(2) \cdot T(5) : AC \leftarrow DR, \boxed{SC \leftarrow 0}$ （⑥：LDA）

$D(3) \cdot T(4) : M[AR] \leftarrow AC, \boxed{SC \leftarrow 0}$ （⑤：STA）

$D(4) \cdot T(4) : PC \leftarrow AR, \boxed{SC \leftarrow 0}$ （⑤：BUN）

. . . . . .