計算機論理設計

2. 命令セットアーキテクチャと数値演算プログラミング(1)

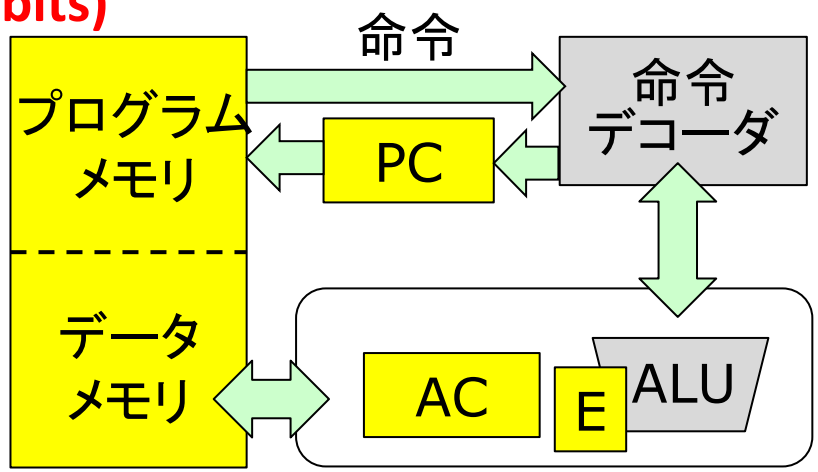
一色 剛

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

- 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



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）

ラベル名の後に"I"

メモリ参照命令

命令	レジスタ転送記述	説明
AND adi	$AC \leftarrow AC\&M[ad]$	ACとメモリオペランドのAND演算
ADD adi	$AC \leftarrow AC + M[ad]$, $E \leftarrow C_{out}(16)$	ACとメモリオペランドの加算 （桁上げ$C_{out}(16)$をEに保存）
LDA adi	$AC \leftarrow M[ad]$	ACにメモリオペランドをロード
STA adi	$M[ad] \leftarrow AC$	ACをメモリにストア
BUN adi	$PC \leftarrow ad$	無条件にadに分岐
BSA adi	$M[ad] \leftarrow PC, PC \leftarrow ad + 1$	復帰アドレスをメモリにストア（退避）し、ad+1に分岐
ISZ adi	$M[ad] \leftarrow M[ad] + 1$, $IF(M[ad] + 1 = 0)\ THEN$ $PC \leftarrow PC + 1$	メモリオペランドをインクリメントし、その結果が0ならば次命令をスキップ

adi : 命令中で指定されるアドレス値
ad : メモリオペランドのアドレス値

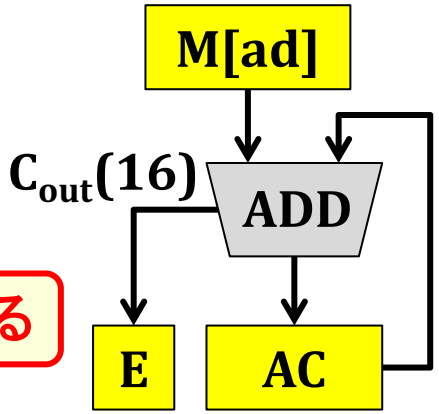
直接アドレス : ad = adi
間接アドレス : ad = M[adi]

1. コンマで区切られた記述は「同時」に実行されるとする
2. 各命令の実行時は、すでに $PC \leftarrow PC+1$ が実行されているとする

メモリ参照命令（補足1）

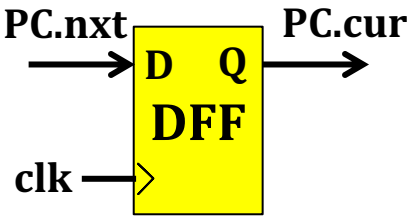
ADD	$AC \leftarrow AC + M[ad], E \leftarrow C_{out}(16)$

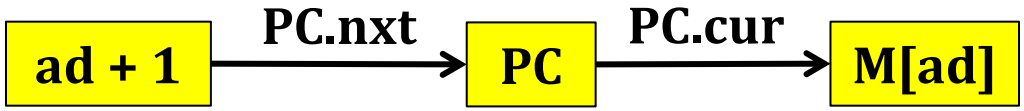
$C_{out}(16)$: 16ビット加算桁上げビット



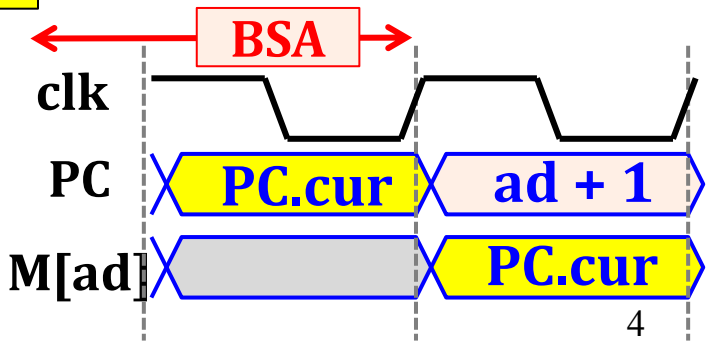
1. コンマで区切られた記述は「同時」に実行されるとする

BSA	$M[ad] \leftarrow PC, PC \leftarrow ad + 1$
復帰アドレスをメモリにストア（退避）し、ad+1に分岐	





PC.cur : 現状態のPC（ M[ad]へ書込み）
PC.nxt = ad + 1 : 次状態のPC → clkの次の立ち上がりで現状態のPCに反映される



メモリ参照命令（補足2）

ISZ	$M[ad] \leftarrow M[ad] + 1,$ $IF(M[ad] + 1 = 0)\ THEN\ PC \leftarrow PC + 1$
メモリオペランドをインクリメントし、その結果が0ならば次命令をスキップ	

1. コンマで区切られた記述は「同時」に実行されるとする

IF (M[ad]+1=0) の判定は、M[ad]が更新される前に行われる

2. 各命令の実行時は、すでに *PC* ← *PC* + 1 が実行されているとする

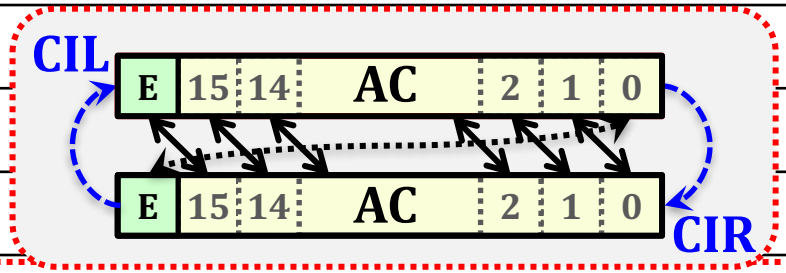
IF (条件式) *THEN PC* ← *PC* + 1 : 条件式がTRUEの場合、さらにPC ← PC + 1を実行する（次命令をスキップする）

IF (条件式) *THEN PC* ← *PC* + 1 → *SKIP-NEXT IF*(条件式)

（以後、上記の省略記述を使用する場合あり）

レジスタ参照命令

命令	レジスタ転送記述	説明
CLA	$AC \leftarrow 0$	AC をクリア
CLE	$E \leftarrow 0$	E をクリア
CMA	$AC \leftarrow \overline{AC}$	AC をビット反転
CME	$E \leftarrow \overline{E}$	E をビット反転
CIR	$AC(14{:}0) \leftarrow AC(15{:}1), AC(15) \leftarrow E, E \leftarrow AC(0)$	AC と Eを右回転シフト
CIL	$AC(15{:}1) \leftarrow AC(14{:}0), AC(0) \leftarrow E, E \leftarrow AC(15)$	AC と Eを左回転シフト
INC	$AC \leftarrow AC + 1$	AC をインクリメント
SPA	$IF(AC(15) = 0)\ THEN\ PC \leftarrow PC + 1$	*SKIP-NEXT IF*(AC ≥ 0)
SNA	$IF(AC(15) = 1)\ THEN\ PC \leftarrow PC + 1$	*SKIP-NEXT IF*(AC < 0)
SZA	$IF(AC = 0)\ THEN\ PC \leftarrow PC + 1$	*SKIP-NEXT IF*(AC = 0)
SZE	$IF(E = 0)\ THEN\ PC \leftarrow PC + 1$	*SKIP-NEXT IF*(E = 0)
HLT	HALT	計算停止



本命令セットの特徴 (1)

■ メモリ参照命令 :

- 命令中にアドレス値(12ビット)を含む（間接アドレス・直接アドレス）→ 命令コード長 =16 – 12 = 4 bits
- メモリ参照演算・転送命令 : AND, ADD, LDA, SRA, ISZ
- プログラム実行制御命令 : BUN（無条件分岐）, BSA （サブルーチン呼出し）, ISZ （条件付き次命令スキップ）

■ レジスタ参照命令（非メモリ参照命令）:

- 命令中にアドレス値を含まない → 命令コード長 = 16 bits
- 算術論理演算命令 : CLA, CLE, CMA, CME, INC
- シフト演算命令 : CIR, CIL
- プログラム実行制御命令 : SPA, SNA, SZA, SZE, HLT

本命令セットの特徴 (2)

AC を更新する命令群

AND	$AC \leftarrow AC\&M[ad]$	LDA	$AC \leftarrow M[ad]$
ADD	$AC \leftarrow AC + M[ad]$	CLA	$AC \leftarrow 0$
CMA	$AC \leftarrow \overline{AC}$	INC	$AC \leftarrow AC + 1$
CIR	$AC(14:0) \leftarrow AC(15:1), AC(15) \leftarrow E$		
CIL	$AC(15:1) \leftarrow AC(14:0), AC(0) \leftarrow E$		

E を更新する命令群

ADD	$E \leftarrow C_{out}(16)$
CLE	$E \leftarrow 0$
CME	$E \leftarrow \overline{E}$
CIR	$E \leftarrow AC(0)$
CIL	$E \leftarrow AC(15)$

PCを更新する命令群

BUN	$PC \leftarrow ad$	BSA	$PC \leftarrow ad + 1$
ISZ	$IF(M[ad] + 1 = 0)\ THEN\ PC \leftarrow PC + 1$		
SPA	$IF(AC(15) = 0)\ THEN\ PC \leftarrow PC + 1$		
SNA	$IF(AC(15) = 1)\ THEN\ PC \leftarrow PC + 1$		
SZA	$IF(AC = 0)\ THEN\ PC \leftarrow PC + 1$		
SZE	$IF(E = 0)\ THEN\ PC \leftarrow PC + 1$		

メモリを更新する命令群

STA	$M[ad] \leftarrow AC$
BSA	$M[ad] \leftarrow PC$
ISZ	$M[ad] \leftarrow M[ad] + 1$

本命令セットのアセンブリ言語仕様

命令：
<命令名>　　（非メモリ参照命令）
<命令名> <label>　（直接アドレスメモリ参照命令）
<命令名> <label> I（間接アドレスメモリ参照命令）

ラベル：
<label> , <命令>
<label> , <データ初期値>

データ宣言：プログラム中で参照するデータラベルを宣言
<label> , <データ初期値>

データ初期値：
HEX <16進定数>
DEC <10進定数>

アドレス指定：ORGに続くワードのアドレスを<16進定数>で指定
ORG <16進定数>
<命令ワード> or <データワード>
（アドレス指定のないワードは0番地から順次配置される）

アセンブリプログラム終端：
END

絶対値計算

C = |A|; ⇒ C = (A < 0) ? – A : A ;

```
        LDA  A      : AC ← M[A]
        SNA         : SKIP-NEXT IF(AC(15) = 1)
        BUN  L0     : PC ← L0
        CMA         : AC ← ‾AC
        INC         : AC ← AC + 1
L0,     STA  C      : M[C] ← AC
        HLT         : 計算停止
A,      DEC  –100   : データ A 宣言(M[A] = −100)
C,      DEC  0      : データ C 宣言(M[C] = 0)
```

AC < 0 ならば
次命令スキップ

AC >= 0 ならば
L0へ分岐

AC ← – AC

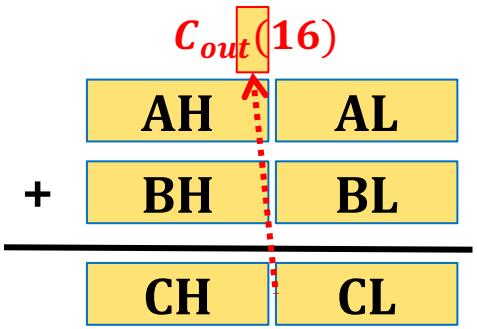
命令
ラベル

データ
ラベル

データ宣言 : プログラムが到達可能なアドレスには配置しない
初期化が必要でないデータは、任意の初期値でよい

倍精度加算

{ CH, CL } = { AH, AL } + { BH, BL };



16ビットデータを2ワードで
32ビットデータを表現

LDA	AL	: $AC \leftarrow M[AL]$
ADD	BL	: $AC \leftarrow AC + M[BL], E \leftarrow C_{out}(16)$
STA	CL	: $M[CL] \leftarrow AC$
CLA		: $AC \leftarrow 0$
CIL		: $AC(0) \leftarrow E$（ACの上位15ビットは0）
ADD	AH	: $AC \leftarrow AC + M[AH]$
ADD	BH	: $AC \leftarrow AC + M[BH]$
STA	CH	: $M[CH] \leftarrow AC$

下位16ビット加算の
桁上げを反映

HLT命令やデータ宣言は省略

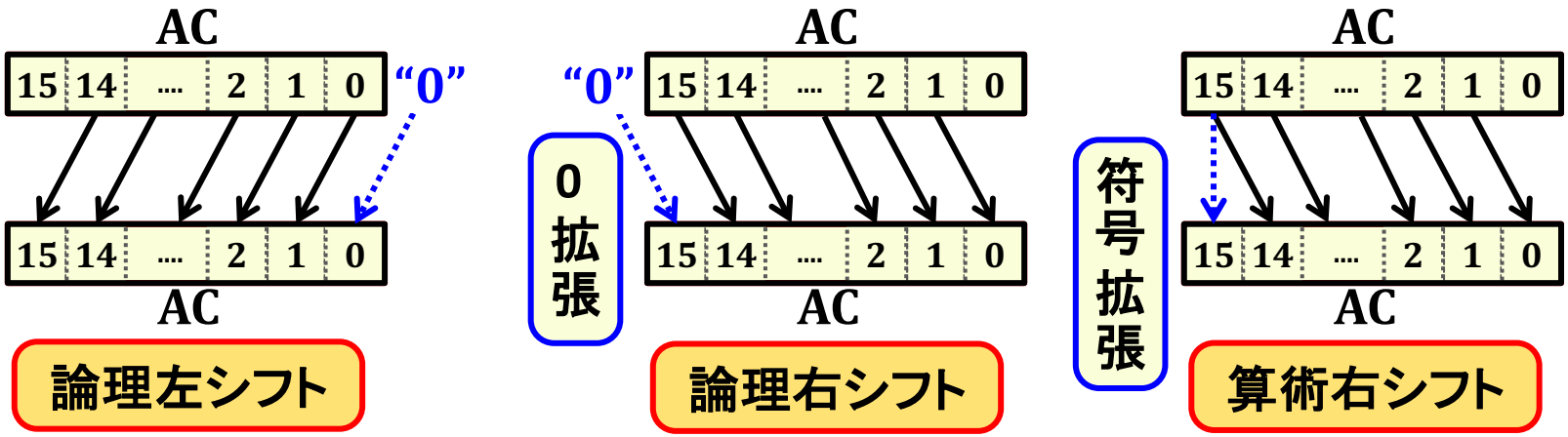
シフト演算

■ 論理シフト：データを「符号なし」と見なす

- 論理左シフト（×2）：$AC(15:1) \leftarrow AC(14:0), AC(0) \leftarrow 0$
- 論理右シフト（÷2）：$AC(14:0) \leftarrow AC(15:1), AC(15) \leftarrow 0$

■ 算術シフト：データを「2の補数」と見なす（AC(15):符号）

- 算術左シフト（×2）：$AC(15:1) \leftarrow AC(14:0), AC(0) \leftarrow 0$
  （論理左シフトと同じ）
- 算術右シフト（÷2）：$AC(14:0) \leftarrow AC(15:1), AC(15) \leftarrow AC(15)$



シフト演算

■ 論理シフト：データを「符号なし」と見なす

- 論理左シフト（×2）：$AC(15:1) \leftarrow AC(14:0), AC(0) \leftarrow 0$
- 論理右シフト（÷2）：$AC(14:0) \leftarrow AC(15:1), AC(15) \leftarrow 0$

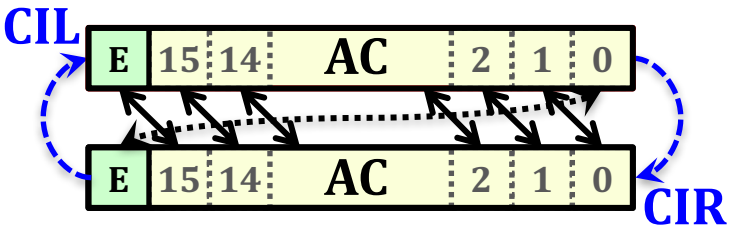
■ 算術シフト：データを「2の補数」と見なす（AC(15):符号）

- 算術左シフト（×2）：$AC(15:1) \leftarrow AC(14:0), AC(0) \leftarrow 0$
  （論理左シフトと同じ）
- 算術右シフト（÷2）：$AC(14:0) \leftarrow AC(15:1), AC(15) \leftarrow AC(15)$

入力値	論理右シフト	算術右シフト	論理左シフト	算術左シフト
1111 符号なし：15 2の補数：-1	0111 $\lfloor 15/2 \rfloor = 7$	1111 $\lfloor -1/2 \rfloor = -1$	1110 $15 \cdot 2 = 30$ （オーバーフロー）	1110 $(-1) \cdot 2 = -2$
0111 符号なし：7 2の補数：7	0011 $\lfloor 7/2 \rfloor = 3$	0011 $\lfloor 7/2 \rfloor = 3$	1110 $7 \cdot 2 = 14$	1110 $7 \cdot 2 = 14$ （オーバーフロー）

シフト演算プログラム

CIR	$AC(14:0) \leftarrow AC(15:1), AC(15) \leftarrow E, E \leftarrow AC(0)$
CIL	$AC(15:1) \leftarrow AC(14:0), AC(0) \leftarrow E, E \leftarrow AC(15)$



- 論理左シフト（÷2）: $AC(15:1) \leftarrow AC(14:0), AC(0) \leftarrow 0$

```
CLE   : E ← 0
CIL   : AC(15:1) ← AC(14:0), AC(0) ← E
```

- 論理右シフト（×2）: $AC(14:0) \leftarrow AC(15:1), AC(15) \leftarrow 0$

```
CLE   : E ← 0
CIR   : AC(14:0) ← AC(15:1), AC(15) ← E
```

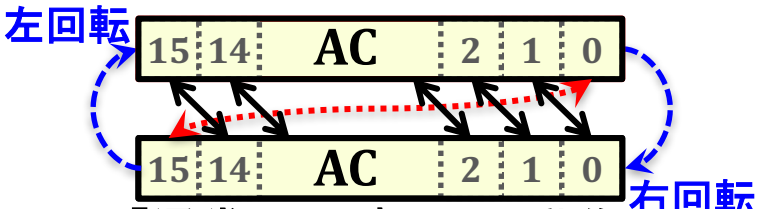
- 算術右シフト（×2）: $AC(14:0) \leftarrow AC(15:1), AC(15) \leftarrow AC(15)$

```
CLE   : E ← 0
SPA   : SKIP-NEXT (AC(15) = 0)
CME   : E ← E̅  (E = 1)
CIR   : AC(14:0) ← AC(15:1), AC(15) ← E
```

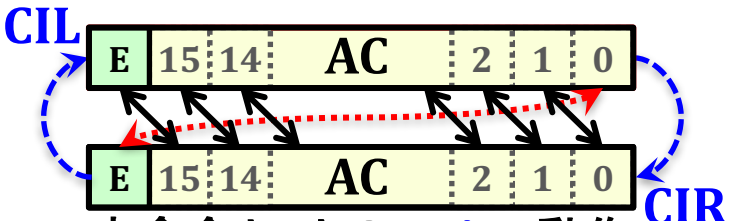
回転シフト演算

■ 回転シフト：データのビット列をシフト方向に回転する

- 左回転シフト：$AC(15:1) \leftarrow AC(14:0), AC(0) \leftarrow AC(15)$
- 右回転シフト：$AC(14:0) \leftarrow AC(15:1), AC(15) \leftarrow AC(0)$



「通常」の回転シフト動作



本命令セットのCIL/CIR動作

左回転シフトプログラム

CLE	: $E \leftarrow 0$
CIL	: $AC(15:1) \leftarrow AC(14:0), AC(0) \leftarrow E, E \leftarrow AC(15)$（AC(0) = 0）
SZE	: $SKIP\text{-}NEXT\ IF(E = 0)$（元のAC(15)が 0であればこのまま）
INC	: $AC \leftarrow AC + 1$ （AC(0)を 1 にするため）

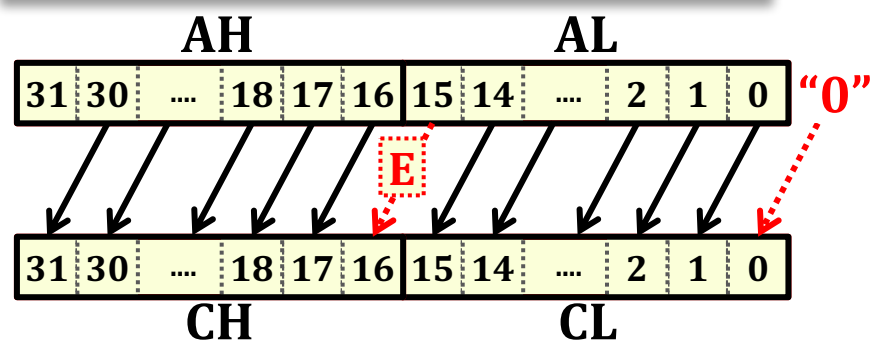
右回転シフトプログラム

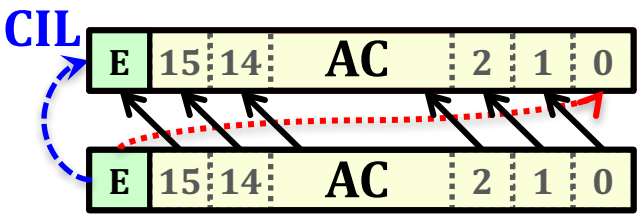
CLE	: $E \leftarrow 0$
CIR	: $AC(14:0) \leftarrow AC(15:1), AC(15) \leftarrow E, E \leftarrow AC(0)$（AC(15) = 0）
SZE	: $SKIP\text{-}NEXT\ IF(E = 0)$（元のAC(0)が 0であればこのまま）
ADD T	: $AC \leftarrow AC + M[T]$ （M[T]=0x8000とする → AC(15)を 1 にするため）

倍精度論理左シフト演算

{ CH, CL } = { AH, AL } << 1;

16ビットデータを2ワードで32ビットデータを表現



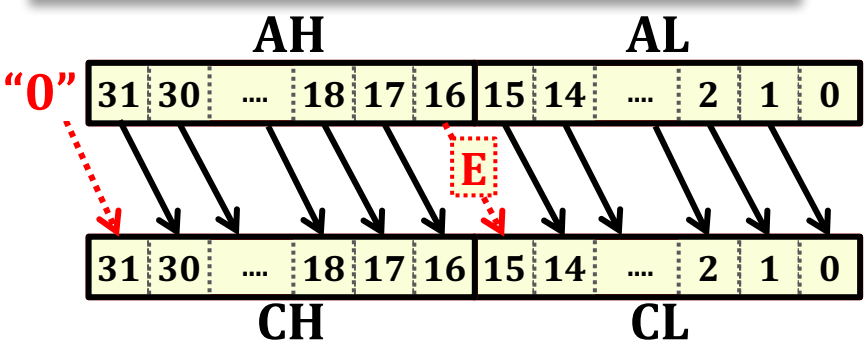


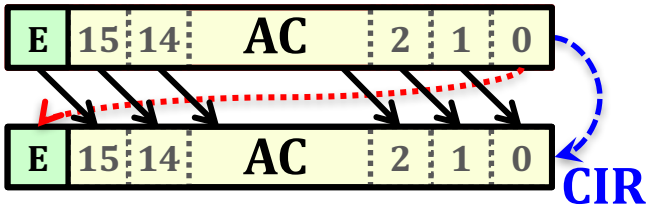
CLE	: $E \leftarrow 0$
LDA AL	: $AC \leftarrow M[AL]$
CIL	: $AC(15:1) \leftarrow AC(14:0), AC(0) \leftarrow E, E \leftarrow AC(15)$
STA CL	: $M[CL] \leftarrow AC$
LDA AH	: $AC \leftarrow M[AH]$
CIL	: $AC(15:1) \leftarrow AC(14:0), AC(0) \leftarrow E, E \leftarrow AC(15)$
STA CH	: $M[CH] \leftarrow AC$

倍精度論理右シフト演算

{ CH, CL } = { AH, AL } >> 1;

16ビットデータを2ワードで32ビットデータを表現



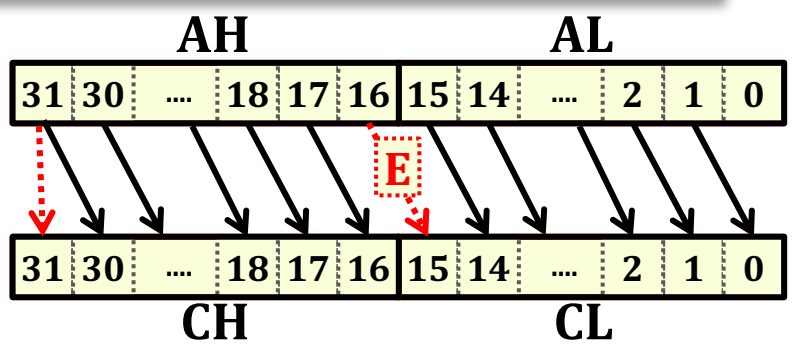


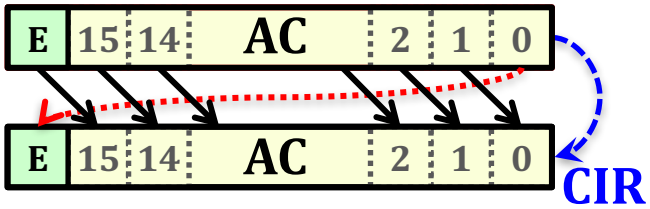
CLE	: $E \leftarrow 0$
LDA AH	: $AC \leftarrow M[AH]$
CIR	: $AC(14:0) \leftarrow AC(15:1), AC(15) \leftarrow E, E \leftarrow AC(0)$
STA CH	: $M[CH] \leftarrow AC$
LDA AL	: $AC \leftarrow M[AL]$
CIR	: $AC(14:0) \leftarrow AC(15:1), AC(15) \leftarrow E, E \leftarrow AC(0)$
STA CL	: $M[CL] \leftarrow AC$

倍精度算術右シフト演算

{ CH, CL } = { AH, AL } >> 1;

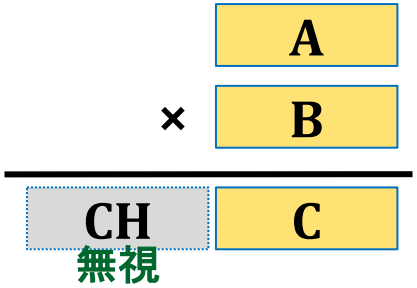
16ビットデータを2ワードで
32ビットデータを表現





CLE	: $E \leftarrow 0$
LDA AH	: $AC \leftarrow M[AH]$
SPA	: $SKIP\text{-}NEXT\ IF(AC(15) = 0)$
CME	: $E \leftarrow \overline{E}\ \ (E = 1)$
CIR	: $AC(14{:}0) \leftarrow AC(15{:}1), AC(15) \leftarrow E, E \leftarrow AC(0)$
STA CH	: $M[CH] \leftarrow AC$
LDA AL	: $AC \leftarrow M[AL]$
CIR	: $AC(14{:}0) \leftarrow AC(15{:}1), AC(15) \leftarrow E, E \leftarrow AC(0)$
STA CL	: $M[CL] \leftarrow AC$

単精度乗算



Nビット×Nビット = 2Nビット
→単精度乗算は積の
下位Nビットだけを出力
→ 積がNビットに収まるように
にプログラムで調整が必要

```
int a, b, c;
c = a * b;
```

32ビット×32ビット →
32ビット積 （C言語）

■ Nビット単精度乗算:「符号なし」データとして考える

$$A = \sum_{i=0}^{N-1} A_i \cdot 2^i$$

$$B = \sum_{i=0}^{N-1} B_i \cdot 2^i$$

$$A \cdot B = A \cdot \sum_{i=0}^{N-1} B_i \cdot 2^i = \sum_{i=0}^{N-1} B_i \cdot (A \cdot 2^i)$$

(N=8)	00001001	(A) 9
	00001101	(B) 13
	00001001	27
0	00000000	9
00	00100100	(C) 117
000	01001000	
0000	00000000	
00000	00000000	
000000	00000000	
0000000	00000000	
00000000	01110101	

単精度乗算プログラム

$$A \cdot B = \sum_{i=0}^{N-1} B_i \cdot (A \cdot 2^i)$$

$A \cdot 2^i$: A の論理左シフト
B_i : B の i ビット目
$B_i = 1$ の場合 $A \cdot 2^i$を C に加算

00001001	A
00001101	B
00001001	$B_0 \cdot A \cdot 2^0$
00000000	$B_1 \cdot A \cdot 2^1$
00100100	$B_2 \cdot A \cdot 2^2$
01001000	$B_3 \cdot A \cdot 2^3$
00000000	$B_4 \cdot A \cdot 2^4$
00000000	$B_5 \cdot A \cdot 2^5$
00000000	$B_6 \cdot A \cdot 2^6$
00000000	$B_7 \cdot A \cdot 2^7$
01110101	C

```
c = 0;
for (k = 0; k < N; k++) {
    bi = b & 1;
    b = b >> 1;
    if (bi)
        c = c + a;
    a = a << 1;
}
```

a = A, b = B とする

bi : B の i ビット目

$A \cdot 2^i$を C に加算

a = $A \cdot 2^i$

単精度乗算プログラム

```
c = 0; k = –N;
do {
   bi = b & 1;
   b = b >> 1;
   if (bi)
      c = c + a;
   a = a << 1;
   k = k + 1;
} while (k != 0);
```

```
L0,  LDA  B
     CIR
     STA  B
     SZE
     BUN  L1
     BUN  L2
L1,  LDA  C
     ADD  A
     STA  C
     CLE
L2,  LDA  A
     CIL
     STA  A
     ISZ  K
     BUN  L0
     HLT
<データ宣言部>
```

$M[B] \leftarrow M[B] \gg 1$

$E \leftarrow B_i$

SKIP-NEXT IF $(E = 0)$

$M[C] \leftarrow M[C] + M[A]$

$M[A] \leftarrow M[A] \ll 1$

$M[K] \leftarrow M[K] + 1,$

SKIP-NEXT IF $(M[K] + 1 = 0)$

```
<データ宣言部>
A,    DEC  9
B,    DEC  13
C,    DEC  0
K,    DEC  –16
```

単精度乗算プログラム

00001001	A
00001101	B
00001001	$B_0 \cdot A \cdot 2^0$
00000000	$B_1 \cdot A \cdot 2^1$
00100100	$B_2 \cdot A \cdot 2^2$
01001000	$B_3 \cdot A \cdot 2^3$
00000000	$B_4 \cdot A \cdot 2^4$
00000000	$B_5 \cdot A \cdot 2^5$
00000000	$B_6 \cdot A \cdot 2^6$
00000000	$B_7 \cdot A \cdot 2^7$
01110101	C

```
L0,  LDA  B
     CIR
     STA  B
     SZE
     BUN  L1
     BUN  L2
L1,  LDA  C
     ADD  A
     STA  C
     CLE
L2,  LDA  A
     CIL
     STA  A
     ISZ  K
     BUN  L0
     HLT
<データ宣言部>
```

$M[B] \leftarrow M[B] \gg 1$
$E \leftarrow B_i$

$SKIP\text{-}NEXT\ IF\ (E = 0)$

$M[C] \leftarrow M[C] + M[A]$

$M[A] \leftarrow M[A] \ll 1$

$M[K] \leftarrow M[K] + 1,$
$SKIP\text{-}NEXT\ IF\ (M[K] + 1 = 0)$

```
<データ宣言部>
A,    DEC  9
B,    DEC  13
C,    DEC  0
K,    DEC  -8
```

次ページからは、N=8ビットとしてプログラム動作を解析する

単精度乗算プログラム

M[B]の最上位ビットにE（不定値）が代入される

00001001	A
00001101	B
00001001	$B_0 \cdot A \cdot 2^0$
00000000	$B_1 \cdot A \cdot 2^1$
00100100	$B_2 \cdot A \cdot 2^2$
01001000	$B_3 \cdot A \cdot 2^3$
00000000	$B_4 \cdot A \cdot 2^4$
00000000	$B_5 \cdot A \cdot 2^5$
00000000	$B_6 \cdot A \cdot 2^6$
00000000	$B_7 \cdot A \cdot 2^7$
01110101	C

```
L0,  LDA  B
     CIR
     STA  B
     SZE
     BUN  L1
     BUN  L2
L1,  LDA  C
     ADD  A
     STA  C
     CLE
L2,  LDA  A
     CIL
     STA  A
     ISZ  K
     BUN  L0
     HLT
<データ宣言部>
```

$M[B] = 00001101$
$E \leftarrow 1$
$M[B] = x0000110$

$SKIP\text{-}NEXT\ IF\ (E = 0)$

$M[C] = 00000000$
$M[A] = 00001001$
$M[C] = 00001001$

$M[A] = 00010010$

$M[K] = -7$
$SKIP\text{-}NEXT\ IF\ (M[K] + 1 = 0)$

単精度乗算プログラム

M[B]の最上位ビットにE（不定値）が代入される

00001001	A
00001101	B
00001001	$B_0 \cdot A \cdot 2^0$
00000000	$B_1 \cdot A \cdot 2^1$
00100100	$B_2 \cdot A \cdot 2^2$
01001000	$B_3 \cdot A \cdot 2^3$
00000000	$B_4 \cdot A \cdot 2^4$
00000000	$B_5 \cdot A \cdot 2^5$
00000000	$B_6 \cdot A \cdot 2^6$
00000000	$B_7 \cdot A \cdot 2^7$
01110101	C

```
L0,  LDA  B
     CIR
     STA  B
     SZE
     BUN  L1
     BUN  L2
L1,  LDA  C
     ADD  A
     STA  C
     CLE
L2,  LDA  A
     CIL
     STA  A
     ISZ  K
     BUN  L0
     HLT
<データ宣言部>
```

$M[B] = x00001110$

$E \leftarrow 0$

$M[B] = 0x000011$

SKIP-NEXT IF $(E = 0)$

$M[A] = 00100100$

$M[K] = -6$

SKIP-NEXT IF $(M[K] + 1 = 0)$

単精度乗算プログラム

M[B]の最上位ビットにE（不定値）が代入される

00001001	A
00001101	B
00001001	$B_0 \cdot A \cdot 2^0$
00000000	$B_1 \cdot A \cdot 2^1$
00100100	$B_2 \cdot A \cdot 2^2$
01001000	$B_3 \cdot A \cdot 2^3$
00000000	$B_4 \cdot A \cdot 2^4$
00000000	$B_5 \cdot A \cdot 2^5$
00000000	$B_6 \cdot A \cdot 2^6$
00000000	$B_7 \cdot A \cdot 2^7$
01110101	C

```
L0,  LDA  B
     CIR
     STA  B
     SZE
     BUN  L1
     BUN  L2
L1,  LDA  C
     ADD  A
     STA  C
     CLE
L2,  LDA  A
     CIL
     STA  A
     ISZ  K
     BUN  L0
     HLT
<データ宣言部>
```

$M[B] = 0x000011$
$E \leftarrow 1$
$M[B] = 00x00001$

$SKIP\text{-}NEXT\ IF\ (E = 0)$

$M[C] = 00001001$
$M[A] = 00100100$
$M[C] = 00101101$

$M[A] = 01001000$

$M[K] = -5$
$SKIP\text{-}NEXT\ IF\ (M[K] + 1 = 0)$

単精度乗算プログラム

M[B]の最上位ビットにE（不定値）が代入される

00001001	A
00001101	B
00001001	$B_0 \cdot A \cdot 2^0$
00000000	$B_1 \cdot A \cdot 2^1$
00100100	$B_2 \cdot A \cdot 2^2$
01001000	$B_3 \cdot A \cdot 2^3$
00000000	$B_4 \cdot A \cdot 2^4$
00000000	$B_5 \cdot A \cdot 2^5$
00000000	$B_6 \cdot A \cdot 2^6$
00000000	$B_7 \cdot A \cdot 2^7$
01110101	C

```
L0,  LDA  B
     CIR
     STA  B
     SZE
     BUN  L1
     BUN  L2
L1,  LDA  C
     ADD  A
     STA  C
     CLE
L2,  LDA  A
     CIL
     STA  A
     ISZ  K
     BUN  L0
     HLT
<データ宣言部>
```

$M[B] = 00x00001$
$E \leftarrow 1$
$M[B] = 000x0000$

$SKIP\text{-}NEXT\ IF\ (E = 0)$

$M[C] = 00101101$
$M[A] = 01001000$
$M[C] = 01110101$

$M[A] = 10010000$

$M[K] = -4$
$SKIP\text{-}NEXT\ IF\ (M[K] + 1 = 0)$

単精度乗算プログラム

M[B]の最上位ビットにE（不定値）が代入される

00001001	A
00001101	B
00001001	$B_0 \cdot A \cdot 2^0$
00000000	$B_1 \cdot A \cdot 2^1$
00100100	$B_2 \cdot A \cdot 2^2$
01001000	$B_3 \cdot A \cdot 2^3$
00000000	$B_4 \cdot A \cdot 2^4$
00000000	$B_5 \cdot A \cdot 2^5$
00000000	$B_6 \cdot A \cdot 2^6$
00000000	$B_7 \cdot A \cdot 2^7$
01110101	C

```
L0,  LDA  B
     CIR
     STA  B
     SZE
     BUN  L1
     BUN  L2
L1,  LDA  C
     ADD  A
     STA  C
     CLE
L2,  LDA  A
     CIL
     STA  A
     ISZ  K
     BUN  L0
     HLT
<データ宣言部>
```

$M[B] = 000x0000$
$E \leftarrow 0$
$M[B] = 0000x000$

SKIP-NEXT IF $(E = 0)$

$M[A] = 00100000$

$M[K] = -3$
SKIP-NEXT IF $(M[K] + 1 = 0)$

単精度乗算プログラム

M[B]の最上位ビットにE（不定値）が代入される

00001001	A
00001101	B
00001001	$B_0 \cdot A \cdot 2^0$
00000000	$B_1 \cdot A \cdot 2^1$
00100100	$B_2 \cdot A \cdot 2^2$
01001000	$B_3 \cdot A \cdot 2^3$
00000000	$B_4 \cdot A \cdot 2^4$
00000000	$B_5 \cdot A \cdot 2^5$
00000000	$B_6 \cdot A \cdot 2^6$
00000000	$B_7 \cdot A \cdot 2^7$
01110101	C

```
L0,  LDA  B
     CIR
     STA  B
     SZE
     BUN  L1
     BUN  L2
L1,  LDA  C
     ADD  A
     STA  C
     CLE
L2,  LDA  A
     CIL
     STA  A
     ISZ  K
     BUN  L0
     HLT
<データ宣言部>
```

$M[B] = 0000x000$

$E \leftarrow 0$

$M[B] = 10000x00$

SKIP-NEXT IF $(E = 0)$

$M[A] = 01000000$

$M[K] = -2$

SKIP-NEXT IF $(M[K] + 1 = 0)$

単精度乗算プログラム

M[B]の最上位ビットにE（不定値）が代入される

00001001	A
00001101	B
00001001	$B_0 \cdot A \cdot 2^0$
00000000	$B_1 \cdot A \cdot 2^1$
00100100	$B_2 \cdot A \cdot 2^2$
01001000	$B_3 \cdot A \cdot 2^3$
00000000	$B_4 \cdot A \cdot 2^4$
00000000	$B_5 \cdot A \cdot 2^5$
00000000	$B_6 \cdot A \cdot 2^6$
00000000	$B_7 \cdot A \cdot 2^7$
01110101	C

```
L0,  LDA  B
     CIR
     STA  B
     SZE
     BUN  L1
     BUN  L2
L1,  LDA  C
     ADD  A
     STA  C
     CLE
L2,  LDA  A
     CIL
     STA  A
     ISZ  K
     BUN  L0
     HLT
<データ宣言部>
```

$M[B] = 10000x00$

$E \leftarrow 0$

$M[B] = 010000x0$

SKIP-NEXT IF $(E = 0)$

$M[A] = 10000000$

$M[K] = -1$

SKIP-NEXT IF $(M[K] + 1 = 0)$

単精度乗算プログラム

M[B]の最上位ビットにE（不定値）が代入される

00001001	A
00001101	B
00001001	$B_0 \cdot A \cdot 2^0$
00000000	$B_1 \cdot A \cdot 2^1$
00100100	$B_2 \cdot A \cdot 2^2$
01001000	$B_3 \cdot A \cdot 2^3$
00000000	$B_4 \cdot A \cdot 2^4$
00000000	$B_5 \cdot A \cdot 2^5$
00000000	$B_6 \cdot A \cdot 2^6$
00000000	$B_7 \cdot A \cdot 2^7$
01110101	C

```
L0,  LDA  B
     CIR
     STA  B
     SZE
     BUN  L1
     BUN  L2
L1,  LDA  C
     ADD  A
     STA  C
     CLE
L2,  LDA  A
     CIL
     STA  A
     ISZ  K
     BUN  L0
     HLT
<データ宣言部>
```

$M[B] = 010000x0$
$E \leftarrow 0$
$M[B] = 0010000x$

$SKIP\text{-}NEXT\ IF\ (E = 0)$

$M[A] = 00000000$

$M[K] = 0$
$SKIP\text{-}NEXT\ IF\ (M[K] + 1 = 0)$