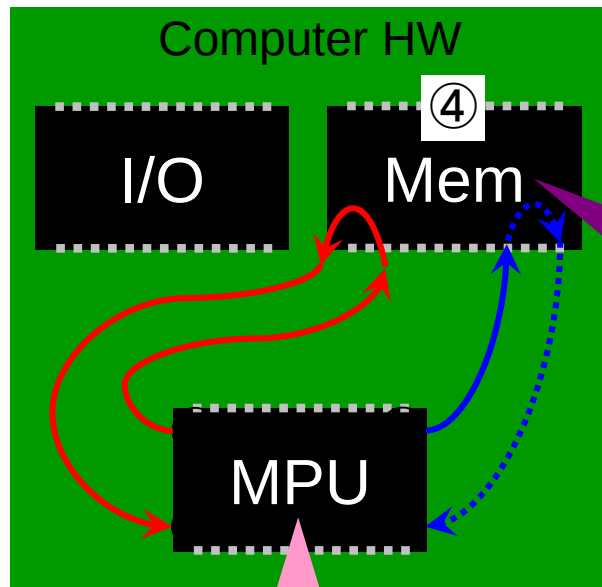


命令セットアーキテクチャ (ISA: Instruction Set Architecture)



(機械語) プログラム = 命令の列とデータの列

ストアードプログラム方式



②

〇〇操作をしないで、×××について

OPcode
(オペコード)
オペレーションコード

operand
(オペランド)

メモリから命令を取ってくる (命令フェッチ: IF)

命令を解釈する (命令デコード: ID) OPcode...何をすべきか
operand...何についてか

命令を実行する (命令実行: EX)

① 作業のための記憶装置

③ アドレッシングモード
「何」のありかの示し方
《EXで使うデータの場所》

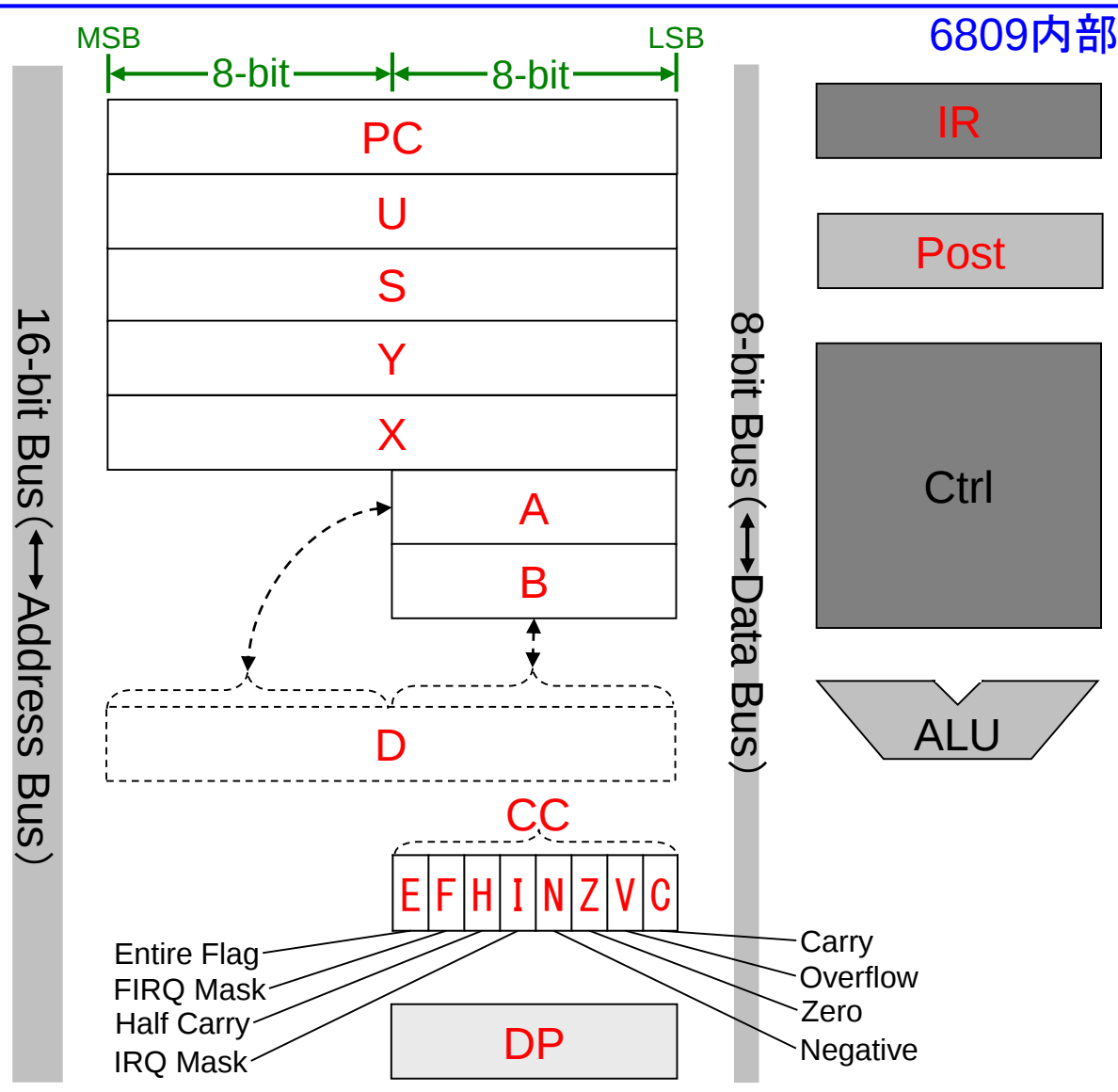
- operandそのものが「何」である。
- operandは「何」のありかを示すものである。

① プロセッサ内部の記憶装置、② 命令 (種類、形式)、③ アドレッシングモード、④ メモリ
これらは、プログラマがコンピュータハードウェアについて知っているべき事項

ISA (命令セットアーキテクチャ): ハードウェアとソフトウェアのインターフェース

6809のISA : 内部記憶装置(1/3)

ISA: **プロセッサ内部の記憶装置**、命令(種類、形式)、アドレッシングモード、メモリ



U: User Stack Register
S: Hardware Stack Register
X, Y: Index Register
A, B, (D): Accumulator
CC: Condition Code Register

PC: Program Counter
DP: Direct Page Register

Post: Post Byte Register
ALU: Arithmetic Logic Unit
IR: Instruction Register
Ctrl: Control Unit

赤文字の装置: 記憶装置

装置の地の色: 白 ↔ 濃い灰
重要度: 高 ↔ 低

CCの各ビットの意味については
man6809.pdf p.1-4 を参照。

6809のISA：内部記憶装置(2/3)

ISA: **プロセッサ内部の記憶装置**、命令(種類、形式)、アドレッシングモード、メモリ

装置名		記憶装置	重要度	分類用別名1	分類用別名2	備考
U	User Stack Register	○	高	Reg	Ptr	—
S	Hardware Stack Register	○	高	Reg	Ptr	ユーザは直接利用しない
X,Y	Index Register	○	高	Reg	Ptr	—
A,B, (D)	Accumulator	○	高	Reg	Acc	被演算データ及び 演算命令実行結果の記憶
CC	Condition Code Register	○	高	Reg	—	条件(注:後述)の記憶
PC	Program Counter	○	高	—	—	次にIFするメモリアドレスを記憶
DP	Direct Page Register	○	低	—	—	—
Post	Post Byte Register	○	中	—	—	—
ALU	Arithmetic Logic Unit	×	中	—	—	—
IR	Instruction Register	○	低	—	—	—
Ctrl	Control Unit	×	低	—	—	—

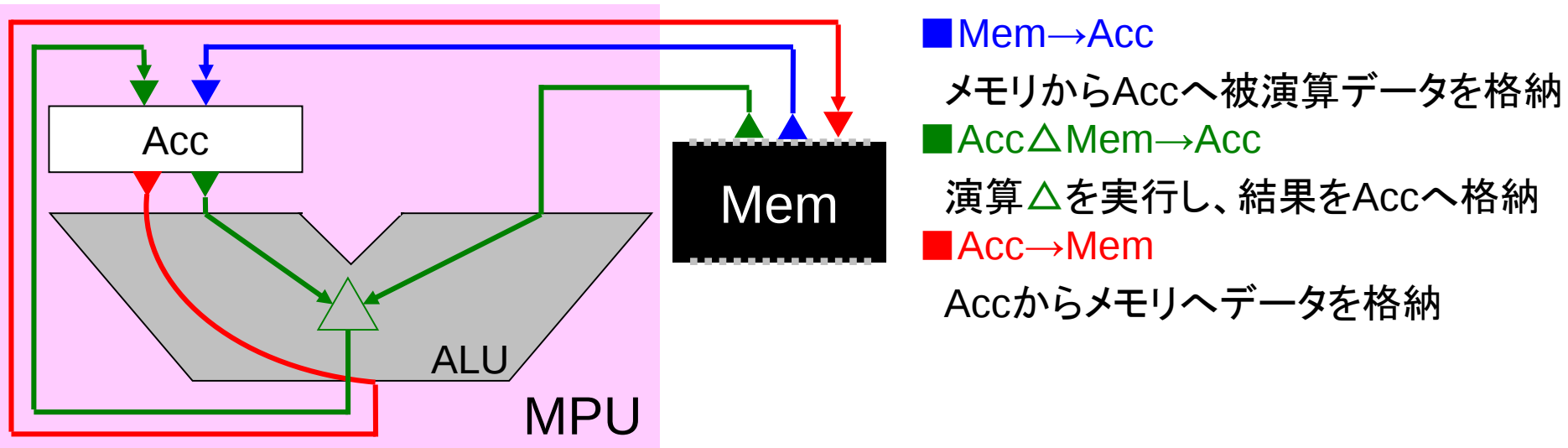
【注】装置の重要度の高低は、本科目の修得においての意味で付したもの

6809のISA：内部記憶装置(3/3)

ISA: **プロセッサ内部の記憶装置**、命令(種類、形式)、アドレッシングモード、メモリ

装置名	記憶装置	重要度	分類用別名1	分類用別名2	備考
A,B, (D) Accumulator	○	高	Reg	Acc	被演算データ及び 演算命令実行結果の記憶

Accを持つ一般的コンピュータシステムにおける、Accとデータの流れ



【注】三色の線は全て、装置間の物理的な配線ではなく、論理的なデータの流れを表す。

- ・この配線を知らなくて良い ⇔ ISAの階層
- ・この配線を知らなくてはならない ⇔ ISAの下階層

6809のISA : 命令(1/3)

ISA: プロセッサ内部の記憶装置、**命令**(種類、形式)、アドレッシングモード、メモリ

基本命令			基本命令			基本命令			基本命令		
機能	名		機能	名		機能	名		機能	名	
BとXの加算	ABX	算術演算(12)	Reg間データ交換	EXG	実行順序変更(5)	ジャンプ	JMP	条件分岐(19)	キャリービットがセットされていなければ分岐	BCC	
桁上げ付き加算	ADC		ロード	LD		サブルーチンへジャンプ	JSR		キャリービットがセットされていれば分岐	BCS	
加算	ADD		プッシュ	PSH		割込みルーチンからの復帰	RTI		等しいならば分岐	BEQ	
0代入	CLR		プル(ポップ)	PUL		サブルーチンからの復帰	RTS		より以上ならば分岐(符号付きデータ比較)	BGE	
10進補正	DAA		ストア	ST		SW割込み	SWI		より大ならば分岐(符号付きデータ比較)	BGT	
1減	DEC		Reg間データ転送	TFR	その他(4)	Regを退避し、HW割込み待ち	CWAI		より大ならば分岐(符号無しデータ比較)	BHI	
1増	INC		算術左シフト	ASL		実効アドレスのロード	LEA		より以上ならば分岐(符号無しデータ比較)	BHS	
乗算	MUL		算術右シフト	ASR		無操作	NOP		より以下ならば分岐(符号付きデータ比較)	BLE	
符号反転	NEG		論理左シフト	LSL		HW割込み待ち	SYNC		より小ならば分岐(符号無しデータ比較)	BLO	
桁上げ付き減算	SBC		論理右シフト	LSR	シフト(6)				より以下ならば分岐(符号無しデータ比較)	BLS	
符号付ビット拡張	SEX		左循環シフト	ROL					より小ならば分岐(符号付きデータ比較)	BLT	
減算	SUB		右循環シフト	ROR					負数ならば分岐	BMI	
論理積	AND	比較、テスト(3)	論理比較	BIT					等しくないならば分岐	BNE	
ビット反転	COM		算術比較	CMP					正数ならば分岐	BPL	
排他的論理和	EOR		テスト(0減)	TST					無条件に分岐	BRA	
論理和	OR								分岐しない分岐	BRN	
		データ転送(6)							サブルーチンへ分岐	BSR	
									オーバーフロービットがセットされていなければ分岐	BVC	
									オーバーフロービットがセットされていれば分岐	BVS	

59種類の基本命令(機能、名前)とその機能的分類

6809のISA : 命令(2/3)

ISA: プロセッサ内部の記憶装置、命令(種類、形式)、アドレッシングモード、メモリ

基本命令名	OPcode		基本命令名	OPcode		基本命令名	OPcode		基本命令名	OPcode
ABX	1種類【3A】	データ転送(6)	EXG	1種類【1E】	実行順序変更(5)	JMP	3種類【0E, 7E, 6E】	条件分岐(19)	BCC	2種類【24, 1024】
ADC	8種類		LD	28種類		JSR	3種類【9D, BD, AD】		BCS	2種類【25, 1025】
ADD	12種類		PSH	2種類【34, 36】		RTI	1種類【3B】		BEQ	2種類【27, 1027】
CLR	5種類		PUL	2種類【35, 37】		RTS	1種類【39】		BGE	2種類【2C, 102C】
DAA	1種類【19】		ST	21種類		SWI	3種類【3F, 103F, 113F】		BGT	2種類【2E, 102E】
DEC	5種類		TFR	1種類【1F】	その他(4)	CWAI	1種類【3C】		BHI	2種類【22, 1022】
INC	5種類		ASL	5種類		LEA	4種類【32, 33, 30, 31】		BHS	2種類【24, 1024】
MUL	1種類【3D】		ASR	5種類		NOP	1種類【12】		BLE	2種類【2F, 102F】
NEG	5種類		LSL	5種類		SYNC	1種類【13】		BLO	2種類【25, 1025】
SBC	8種類		LSR	5種類					BLS	2種類【23, 1023】
SEX	1種類【1D】	シフト(6)	ROL	5種類					BLT	2種類【2D, 102D】
SUB	12種類		ROR	5種類					BMI	2種類【2B, 102B】
AND	9種類		BIT	8種類					BNE	2種類【26, 1026】
COM	5種類		CMP	28種類					BPL	2種類【2A, 102A】
EOR	8種類	比較、テスト(3)	TST	5種類					BRA	2種類【20, 16】
OR	9種類								BRN	2種類【21, 1021】
									BSR	2種類【8D, 17】
									BVC	2種類【28, 1028】
									BVS	2種類【29, 1029】

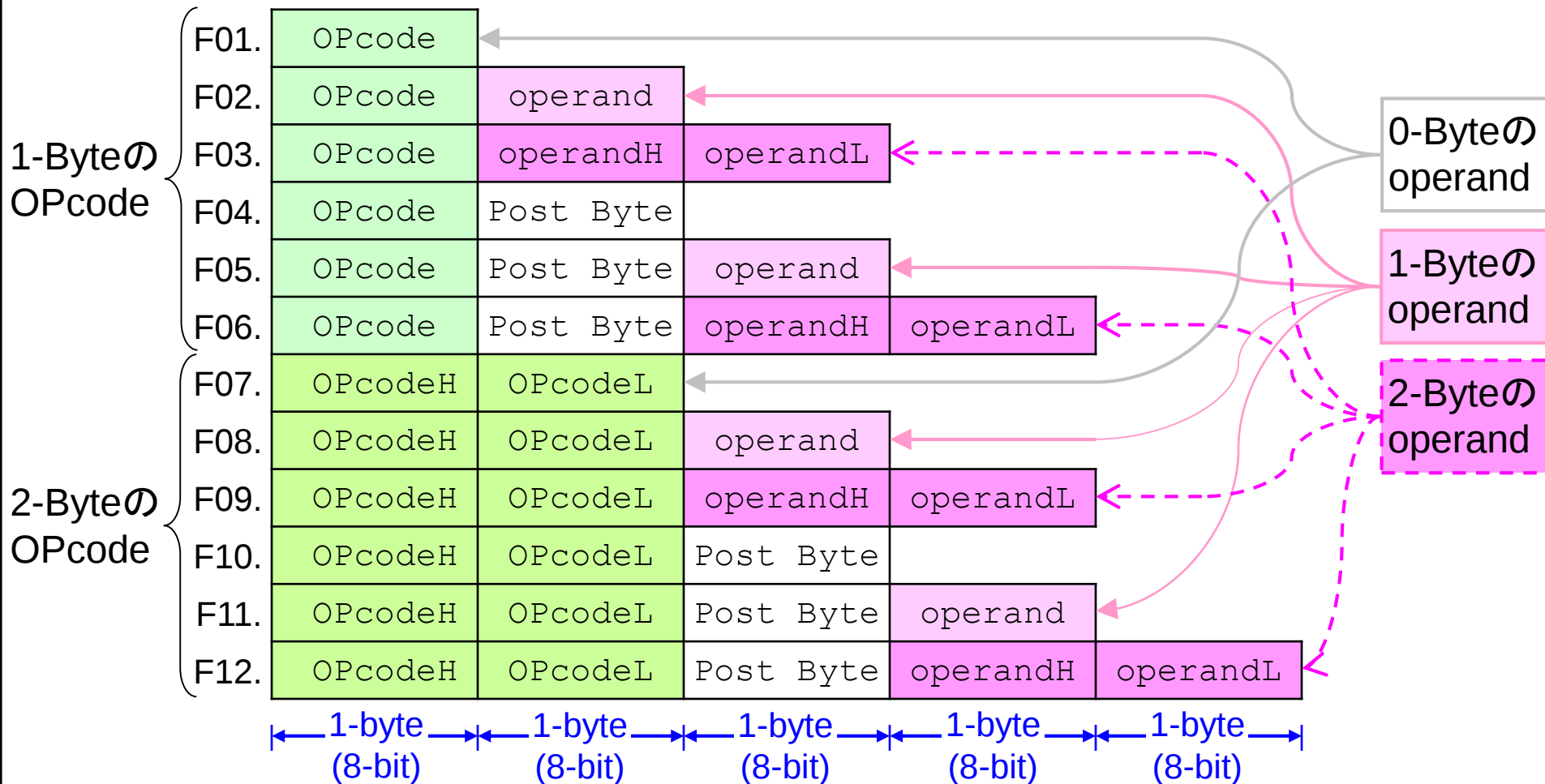
59種類の基本命令それぞれに対応する、OPcode(全268種類)がある。

- ・1つの基本命令に1つのOPcodeが対応: 11基本命令
- ・1つの基本命令に複数のOPcodeが対応: 48基本命令

操作対象のバリエーション
(記憶装置、アドレッシングモード)

6809のISA : 命令(3/3)

ISA: プロセッサ内部の記憶装置、**命令**(種類、形式)、アドレッシングモード、メモリ



・H: 上位バイト、L: 下位バイト

・OPcodeH : Pre Byte(プリ バイト)とも呼ぶ。コードは「10」または「11」

・Post Byte(ポストバイト) : OPcodeに対する付加的情報 操作対象のバリエーション

6809のISA : アドレッシングモード(1/2)

ISA: プロセッサ内部の記憶装置、命令(種類、形式)、**アドレッシングモード**、メモリ

Addressing Mode名			EXで使うデータの場合	operand	備考
Immediate			operandがデータそのもの	hh	
				hhhh	
Inherent			参照Regにデータがある		
Direct			DP:上位8-bit & operand:下位8-bitが示すアドレスを直接参照	hh	
Extended			operandが示すアドレスを直接参照	hhhh	A直
Indexed	Constant Offset	0-bit	Ptrが示すアドレスを直接参照		B直
		5-bit	Ptr+変位が示すアドレスを直接参照	変位はPostByteに。operand無し	C直
		8-bit		hh	D直
		16-bit		hhhh	E直
	Acc Offset		Ptr+Accが示すアドレスを直接参照		F直
	Auto Increment		Ptrが示すアドレスを直接参照⇒Ptr=Ptr+1		G直
			Ptrが示すアドレスを直接参照⇒Ptr=Ptr+2		H直
	Auto Decrement		Ptr=Ptr-1⇒Ptrが示すアドレスを直接参照		I直
			Ptr=Ptr-2⇒Ptrが示すアドレスを直接参照		J直
	PC Relative		PC+変位が示すアドレスを直接参照	hh / hhhh	K直
	Extended Indirection		operandが示すアドレスを間接参照	hhhh	A間
	Constant Offset	0-bit Indirection	Ptrが示すアドレスを間接参照		B間
		8-bit Indirection	Ptr+変位が示すアドレスを間接参照	hh	D間
		16-bit Indirection		hhhh	E間
	Acc Offset Indirection		Ptr+Accが示すアドレスを間接参照		F間
	Auto Increment Indirection		Ptrが示すアドレスを間接参照⇒Ptr=Ptr+2		H間
	Auto Decrement Indirection		Ptr=Ptr-2⇒Ptrが示すアドレスを間接参照		J間
	PC Relative Indirection		PC+変位が示すアドレスを間接参照	hh / hhhh	K間
Branch Relative			PC+変位が示すアドレスを直接参照	hh / hhhh	

6809のISA : アドレッシングモード(2/2)

ISA: プロセッサ内部の記憶装置、命令(種類、形式)、**アドレッシングモード**、メモリ

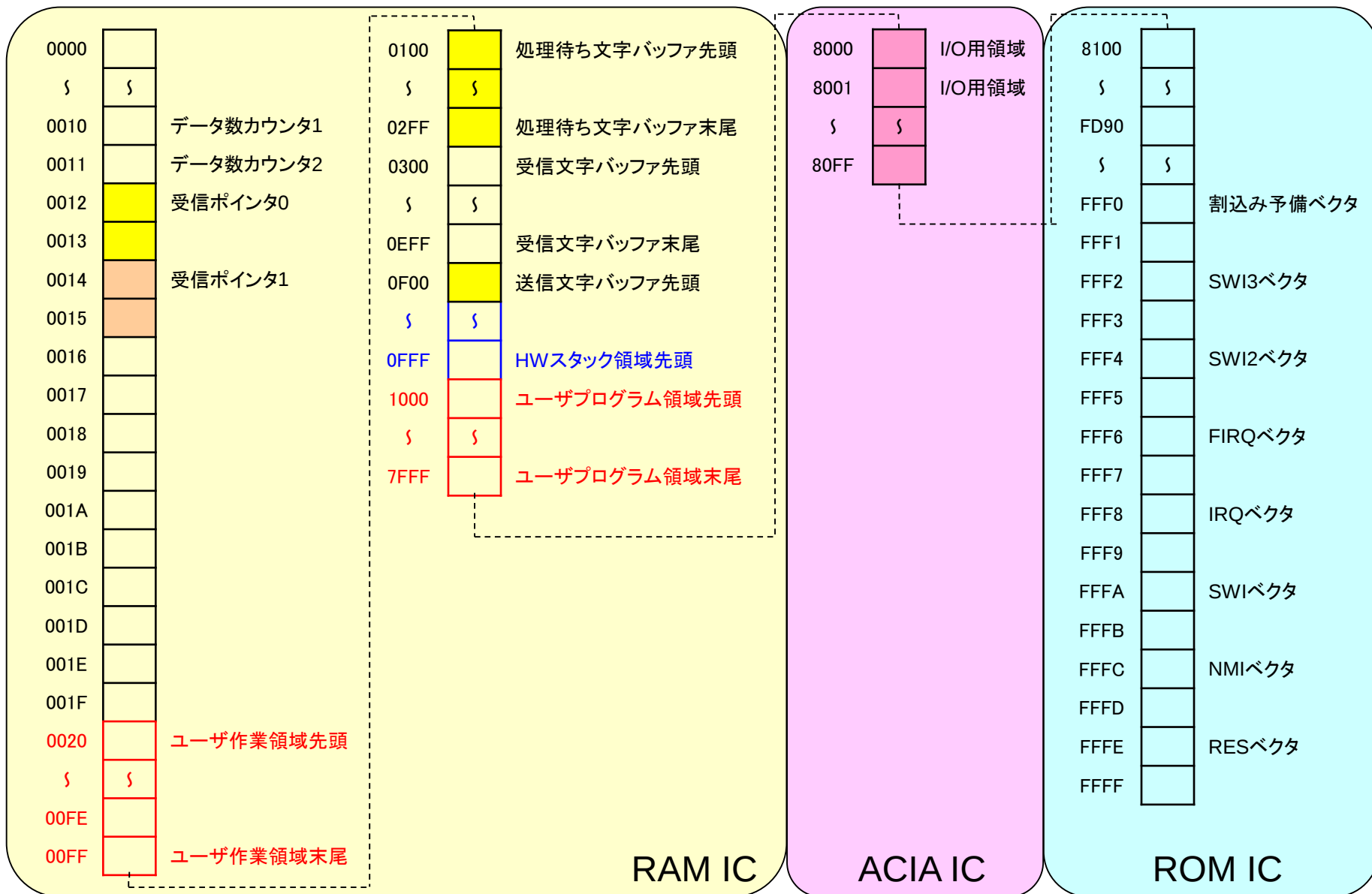
Addressing Mode名			EXで使うデータの場合	operand	備考	
Immediate			operandがデータそのもの	hh		
				hhhh		
Inherent			参照Regにデータがある			
Direct			DP:上位8-bit & operand:下位8-bitが示すアドレスを直接参照	hh		
Extended			operandが示すアドレスを直接参照	hhhh	A直	
Indexed	Constant Offset	0-bit	Ptrが示すアドレスを直接参照		B直	
		5-bit	Ptr+変位が示すアドレスを直接参照	変位はPostByteに。operand無し	C直	
		8-bit		hh	D直	
		16-bit		hhhh	E直	
	Acc Offset		Ptr+Accが示すアドレスを直接参照		F直	
	Auto Increment		Ptrが示すアドレスを直接参照⇒Ptr=Ptr+1		G直	
			Ptrが示すアドレスを直接参照⇒Ptr=Ptr+2		H直	
	Auto Decrement		Ptr=Ptr-1⇒Ptrが示すアドレスを直接参照		I直	
			Ptr=Ptr-2⇒Ptrが示すアドレスを直接参照		J直	
	PC Relative		PC+変位が示すアドレスを直接参照	hh / hhhh	K直	
	Extended Indirection			operandが示すアドレスを間接参照	hhhh	A間
	Constant Offset	0-bit Indirection	Ptrが示すアドレスを間接参照		B間	
		8-bit Indirection	Ptr+変位が示すアドレスを間接参照	hh	D間	
		16-bit Indirection			E間	
Acc Offset Indirection			Ptr+Accが示すアドレスを間接参照		F間	
Auto Increment Indirection			Ptrが示すアドレスを間接参照⇒Ptr=Ptr+2		H間	
Auto Decrement Indirection			Ptr=Ptr-2⇒Ptrが示すアドレスを間接参照		J間	
PC Relative Indirection			PC+変位が示すアドレスを間接参照	hh / hhhh	K間	
Branch Relative			PC+変位が示すアドレスを直接参照	hh / hhhh		

初心者のうちは、扱わなくても良いアドレッシングモード

初心者のうちは、扱わなくても良いアドレッシングモード

6809のISA : 09brdメモリ(1/2)

ISA: プロセッサ内部の記憶装置、命令(種類、形式)、アドレッシングモード、**メモリ**



6809のISA : 09brdメモリ(2/2)

ISA: プロセッサ内部の記憶装置、命令(種類、形式)、アドレッシングモード、**メモリ**

スタック の使い方

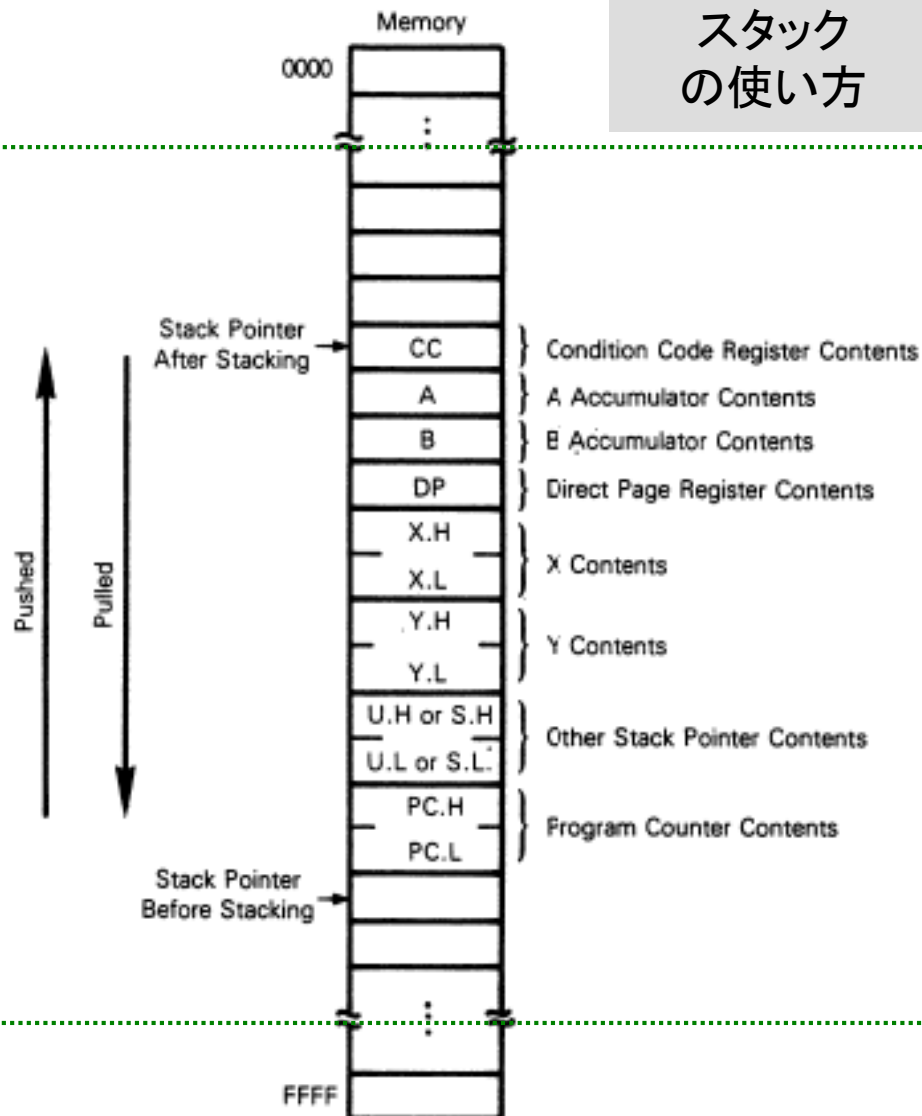
0020		ユーザ作業領域先頭
...	...	
00FF		ユーザ作業領域末尾
0100		処理待ち文字バッファ先頭
...	...	
02FF		処理待ち文字バッファ末尾
0300		受信文字バッファ先頭
...	...	
0EFF		受信文字バッファ末尾
0F00		送信文字バッファ先頭
...	...	
0FFF		HWスタック領域先頭
1000		ユーザプログラム領域先頭
...	...	
7FFF		ユーザプログラム領域末尾

RAM IC

Stack Pointerが
Uレジスタの場合

Stack Pointerが
Sレジスタの場合

Stack Pointerが
Uレジスタの場合



【注】
.H: 上位バイト
.L: 下位バイト

6809のISAの追加資料

ISA: プロセッサ内部の記憶装置、命令(種類、形式)、アドレッシングモード、メモリ

▪ [man6809.pdf](#)
([p.1-3](#) ~ [p.1-4](#))

▪ [man6809.pdf](#)
([p.A-1](#) ~ [p.A-72](#)、[p.C-2](#) ~ [p.C-4](#)、[p.D-1](#) ~ [p.D-4](#))

▪ [6809_Insn_MainList-ML_oa.pdf](#) [6809_Insn_MainList-FULL.pdf](#)

▪ [6809_Insn_MainList-ML_on.pdf](#)

▪ [6809_Insn_Branch-ML.pdf](#) [6809_Insn_Branch-FULL.pdf](#)

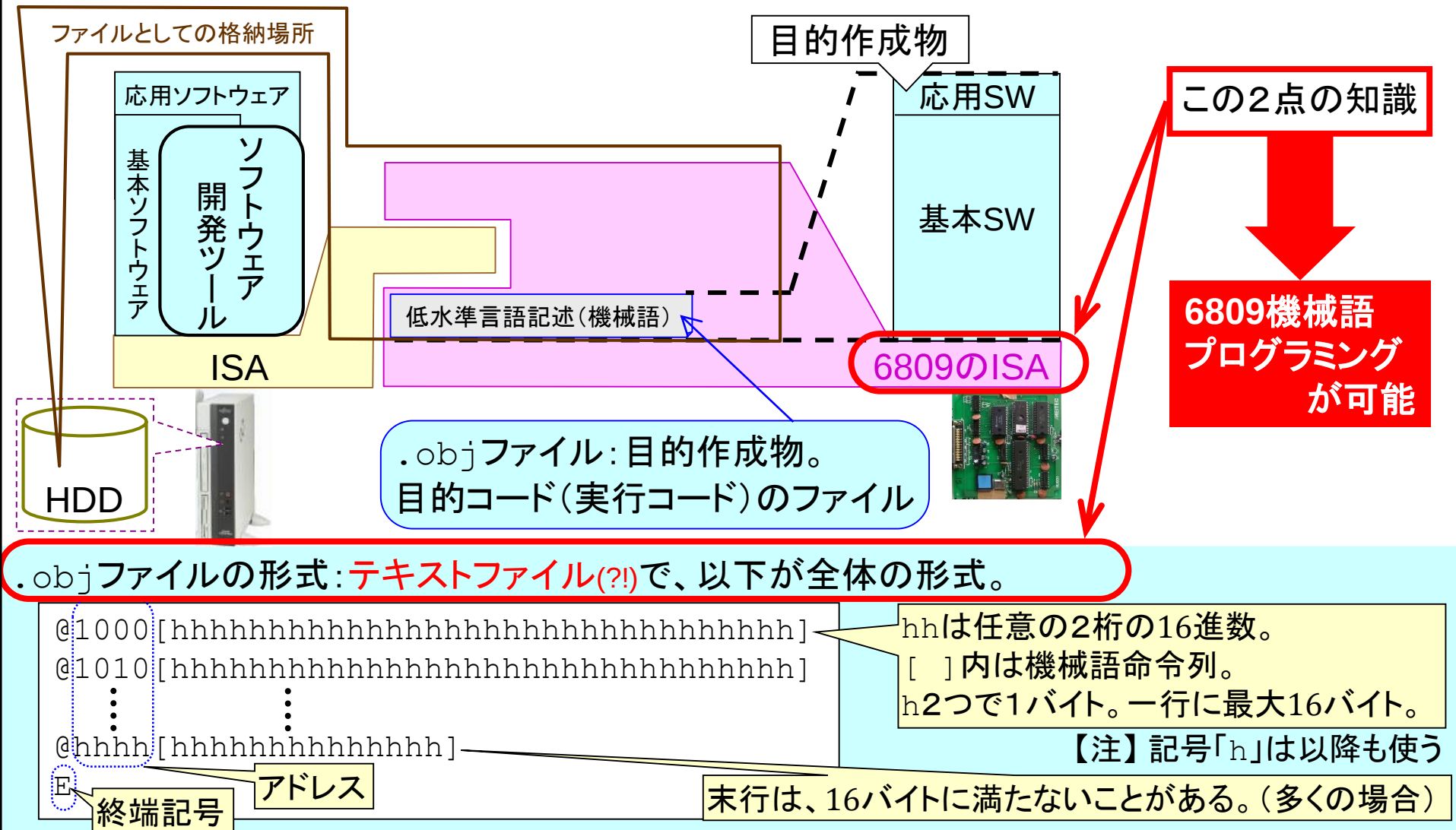
▪ [6809_Insn_PshPul-ML.pdf](#) [6809_Insn_PshPul-FULL.pdf](#)

▪ [man6809.pdf](#)
([p.2-1](#) ~ [p.2-6](#)、[p.D-1](#) ~ [p.D-4](#)、[p.F-3/F-4](#))

▪ [CAE_AddressingModes_V2.pdf](#)

資料は全て、
[講義用Webページ](#)からも開ける。

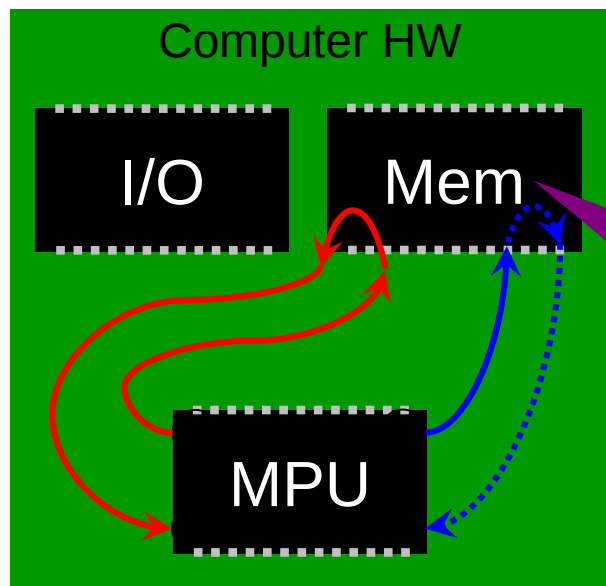
6809機械語プログラミング



【注】

- ・テキストエディタによる機械語プログラミングが可能・・・本科目独特の特徴
- ・機械語プログラムの、最後に実行される命令は必ず「39」とする・・・モニタプログラムとの関係による。

機械語プログラムの使用メモリサイズ、実行サイクル数



(機械語)プログラム = 命令の列とデータの列

ストアードプログラム方式

命令1
命令2
命令3
命令N

#₁(命令1の#)、
#₂(命令2の#)、
#₃(命令3の#)、
#_N(命令Nの#)

~₁(命令1の~)、
~₂(命令2の~)、
~₃(命令3の~)、
~_N(命令Nの~)

- [man6809.pdf](#)
([p.C-2](#) ~ [p.C-4](#)、[p.D-1](#) ~ [p.D-4](#)、[p.F-3/F-4](#))
- [6809_Insn_MainList-ML_oa.pdf](#)
- [6809_Insn_MainList-ML_on.pdf](#)
- [6809_Insn_Branch-ML.pdf](#)
- [6809_Insn_PshPul-ML.pdf](#)

: 命令長(バイト数)

~ : 実行サイクル数

プログラムの
使用メモリサイズ(Bytes) $\equiv \sum_{i=1}^N \#_i + \text{プログラム実行時に静的に使用するメモリのバイト数}$
(プログラムが使用するデータの格納領域)

プログラムの
実行サイクル数(回) $\equiv \sum_{i=1}^N \sim_i$

練習

```
@1000[8668B711004CB71101BE1100BF110239]  
E
```

上のような内容の.objファイルがあるとき、この機械語プログラムの動作を説明せよ。
また、使用メモリサイズ、実行サイクル数を求めよ。

ヒント

(a)	調べたいOPcodeについて、 その意味、アドレッシングモード、命令長(#)、実行サイクル数(~)を知る。 注a-1: 機械語プログラム(機械語命令列)の先頭は、必ずOPcodeである。 注a-2: OPcodeには、1-byteのものと2-byteのものがある。 注a-3: 2-byte長のOPcodeは、必ず「10」または「11」で始まる。 注a-3: PostByteを伴う命令もあることに注意する。
(b)	命令長(#)からoperand長を求める。(operand長 = 命令長 - OPcode長)
(c)	(b) で知ったoperand長から、operandの値と、次に調べるOPcodeの位置を知る。
(d)	・ (a) で知った、OPcodeの意味、アドレッシングモード、実行サイクル数(~) ・ (c) で知った、operandの値 を総合して、1つの命令の動作と実行サイクル数を知る。
(e)	(c) で知った、次に調べるOPcodeの位置をみて、 ・ OPcodeが「39」ならば、その実行サイクル数を記録し、(f) へ。 ・ そうでなければ、(a) へ。
(f)	全ての命令の動作(39を除く)と実行サイクル数(39を含む)を総合すると、 プログラムの動作が説明でき、実行サイクル数が分かる。

練習

3に8を加算し(3+8を行い)、結果をメモリに格納する機械語プログラムを作成せよ。
また、使用メモリサイズ、実行サイクル数を求めよ。

ヒント

		(α)	(β)
(a)	プログラムを、メモリのどこに格納するか。 注a-1: ユーザプログラム領域内でなければならない。 注a-2: 結果を格納する位置と重ならないようにした方がよい。	1000番地から格納。	
(b1)	被演算データ3と8を、メモリにどのように格納し、どのように取得するか。	Immediate(即値)アドレッシングモードを使用する。	最後に実行される命令「39」のすぐ後ろの位置に格納し、Extendedアドレッシングモードを使用する。
(b2)	結果を、メモリのどこに格納するか。 注b2-1: ユーザプログラム領域内、またはユーザ作業領域内でなければならない。 注b2-2: プログラムと重ならないようにした方がよい。	最後に実行される命令「39」のすぐ後ろの位置。	(b1- β)の格納位置のすぐ後ろの位置。
(c)	どのAccを使用するか。 注c-1: 8-bitのものが2種類(A,B)、16-bitのものが1種類(D)、ある。 注c-2: 3も8も想定される結果も、8-bitで収まる。	Acc A を使用。	Acc B を使用。
(d)	(b1) と (c) を踏まえ、被演算データ3を扱うOPcodeをどれにするか。 またそのとき、PostByteとoperandはどうなるか。	Accについて復習 (1) (2) (3)	
(e)	(b1) と (c) を踏まえ、被演算データ8を扱うOPcodeをどれにするか。 またそのとき、PostByteとoperandはどうなるか		
(f)	(b2) と (c) を踏まえ、結果を扱うOPcodeをどれにするか。 またそのとき、PostByteとoperandはどうなるか。		
(g)	最後に実行される命令「39」を忘れない。		

右の2通りの手順で
作成してみよ。

- ① (a- α) $\rightarrow$ (b1- α) $\rightarrow$ (b2- α) $\rightarrow$ (c- α) $\rightarrow$ (d) $\rightarrow$ (e) $\rightarrow$ (f) $\rightarrow$ (g)
 ② (a- α) $\rightarrow$ (b1- β) $\rightarrow$ (b2- β) $\rightarrow$ (c- β) $\rightarrow$ (d) $\rightarrow$ (e) $\rightarrow$ (f) $\rightarrow$ (g)

練習

```
@1000[7E100039]
```

```
E
```

上のような内容の.objファイルがあるとき、この機械語プログラムの動作を説明せよ。
また、使用メモリサイズ、実行サイクル数を求めよ。

練習

```
@1000 [ 6E8CFD39 ]
```

```
E
```

上のような内容の.objファイルがあるとき、この機械語プログラムの動作を説明せよ。
また、使用メモリサイズ、実行サイクル数を求めよ。

練習

@1000 [6E8CFD39]

E

上の機械語プログラムと同じ動作をするプログラムを、
OPcode「20」を用いて作成せよ。
また、使用メモリサイズ、実行サイクル数を求めよ。

ヒント

OPcode「20」は、条件分岐 (Branch) に分類されるOpcodeである。

- [man6809.pdf](#)
([p.C-2](#)～[p.C-4](#)、[p.D-1](#)～[p.D-2](#))
- [6809_Insn_Branch-ML.pdf](#) [6809_Insn_Branch-FULL.pdf](#)