

POCO の命令セット、ハードウェア構成は 2 ページ目以降に示しますが、問題中に不明な点がある場合や手元に資料がない場合は、各自判断して答え、その旨を記して下さい。

1-1. 以下の 16 進数を 2 進数で示せ。また符号付き数と考えて 16bit に符号拡張せよ。

(1) 32 (2) A82 (3) EF

(1) 00110010 (2) 101010000010 (3) 11101111

(1) 0000000000110010 (2) 1111101010000010 (3) 1111111111101111

1-2. 2 の補数を求めてこれを加算することにより、以下の減算を実行する様子をせよ。

(1) 1010 - 0110 (2) 0111 - 1001

(1) 0110 の 2 の補数は、 $1001+1=1010$   $1010+1010 = 10100$  で最上位を無視して 0100

(2) 1001 の 2 の補数は、 $0110+1=0111$   $0111+0111 = 1110$  これはマイナス 2 に当たる

2. 16bit RISC POCO で下の命令を順に実行した。

```
LDI r0, #0x02
MV r1,r0
LD r2,(r0)
LD r3,(r2)
LD r1,(r1)
ADD r1, r3
ST r3,(r1)
```

ここで、データメモリ中の値は以下のように設定した。

0 番地 0005, 1 番地 0003, 2 番地 0001, 3 番地 0004, 4 番地 0002

a) 上記の機械語命令を示せ。またそれぞれの opcode フィールド、function(funcnt) フィールド (あれば) はどうなるかを示せ

```
LDI r0, #0x02 01000 000 00000010 op: 01000
MV r1,r0      00000 001 000 00001 op: 00000 funct:00001
LD r2,(r0)    00000 010 000 01001 op: 00000 funct:01001
LD r3,(r2)    00000 011 010 01001 op: 00000 funct:01001
LD r1,(r1)    00000 001 001 01001 op: 00000 funct:01001
ADD r1, r3    01100 001 011 00110 op: 00000 funct:00110
ST r3,(r1)    00000 011 001 01000 op: 00000 funct:01000
```

b) r0, r1, r2, r3 の値はどのように変化するかを示せ。また、最終的に、どの番地にどのような値が書き込まれるかを示せ。

```
LDI r0, #0x02 r0:2
MV r1,r0      r1:2
LD r2,(r0)    r2:1
LD r3,(r2)    r3:3
LD r1,(r1)    r1:1
ADD r1, r3    r1:4
ST r3,(r1)    4 番地に 3
```

3 16bit RISC POCO で a)SUB 命令、b)LDIU 命令、c)LD 命令を実行する際に、各制御信号線をどのように設定すれば良いか? 表に付け加えよ。

4-1. メモリの特定の番地に入っている数の 2 乗を計算する POCO のプログラム sqmem を書け。ただし、対象となる数の番地はレジスタ r0 中に入っており、2 乗した答えは r1 に入れるものとする。

表 1: 各命令の制御信号

	comsel	alu_bsel	rf_csel	rwe	we
ADDI	01	01	0	1	0
SUB	00	00	0	1	0
LDIU	00	00	0	1	0
LD	00	-	1	1	0

```
sqmem:LD r2,(r0)
      MV r3,r2
      LDI r1,#0
loop: ADD r1,r2
      ADDI r3,#-1
      BNZ r3,loop
      JR r7
```

4-2. sqmem のプログラムをサブルーチンとして利用し、0番地から7番地にしまわれている8つの数の内積(2乗して総和を取った値)を計算するプログラムを書け。答えはどのレジスタに入れてもかまわない。ただし、8としたのは、話をわかりやすくするためなので、対象データ数が増えてもちゃんと計算できるプログラムとすること。8回 sqmem を呼ぶ答えは0点とする。

```
      LDI r0,#0
      LDI r4,#8
      LDI r2,#0
loop: JAL sqmem
      ADD r2,r1
      ADDI r0,#1
      ADDI r4,#-1
      BNZ r4,loop
```

答えは r2 に入る。

5-1.4Kword の主記憶に対して 128word のキャッシュを設けた。ブロックサイズは8ワードとした時、以下の構成のキャッシュの index と tag(key) の大きさを求めよ。(a) ダイレクトマップキャッシュ (b) 2-way セットアソシアティブキャッシュ (c) 4-way セットアソシアティブキャッシュ

- (a) キャッシュブロック数は 16、index 4bit, tag: 5bit
- (b) index 3bit, tag: 6bit
- (c) index 2bit, tag: 7bit

5-2. way 数を大きくすることの利点と欠点を簡単に示せ。

利点：衝突ミスが減る 欠点：ハードウェア量が増え、ヒット時の遅延が増える。

### A) POCO の命令コード

NOP		00000 --- --- 00000
MV rd,rs	rd <- rs	00000 ddd sss 00001
AND rd,rs	rd <- rd AND rs	00000 ddd sss 00010
OR rd,rs	rd <- rd OR rs	00000 ddd sss 00011
SL rd	rd <- rd<<1	00000 ddd --- 00100
SR rd	rd <- rd>>1	00000 ddd --- 00101
ADD rd,rs	rd <- rd + rs	00000 ddd sss 00110
SUB rd,rs	rd <- rd - rs	00000 ddd sss 00111
ST rs, (ra)	rs -> (ra)	00000 sss aaa 01000
LD rd, (ra)	rd <- (ra)	00000 ddd aaa 01001
LDI rd,#X	rd <- X (符号拡張)	01000 ddd XXXXXXXX
LDIU rd,#X	rd <- X (符号拡張なし)	01001 ddd XXXXXXXX
ADDI rd,#X	rd <- rd + X (符号拡張)	01100 ddd XXXXXXXX
ADDIU rd,#X	rd <- rd + X (符号拡張なし)	01101 ddd XXXXXXXX
LDHI rd,#X	rd <- X 0	01010 ddd XXXXXXXX
BEZ rd, X	if (rd==0) pc <- pc + X	10000 ddd XXXXXXXX
BNZ rd, X	if (rd!=0) pc <- pc + X	10001 ddd XXXXXXXX
BPL rd, X	if (rd>=0) pc <- pc + X	10010 ddd XXXXXXXX
BMI rd, X	if (rd<0) pc <- pc + X	10011 ddd XXXXXXXX
JMP X	pc <- pc + X	10100 XXXXXXXXXXXX
JAL X	r7 <- pc, pc <- pc + X	10101 XXXXXXXXXXXX
JR rd	pc <- rd	00000 ddd --- 01010

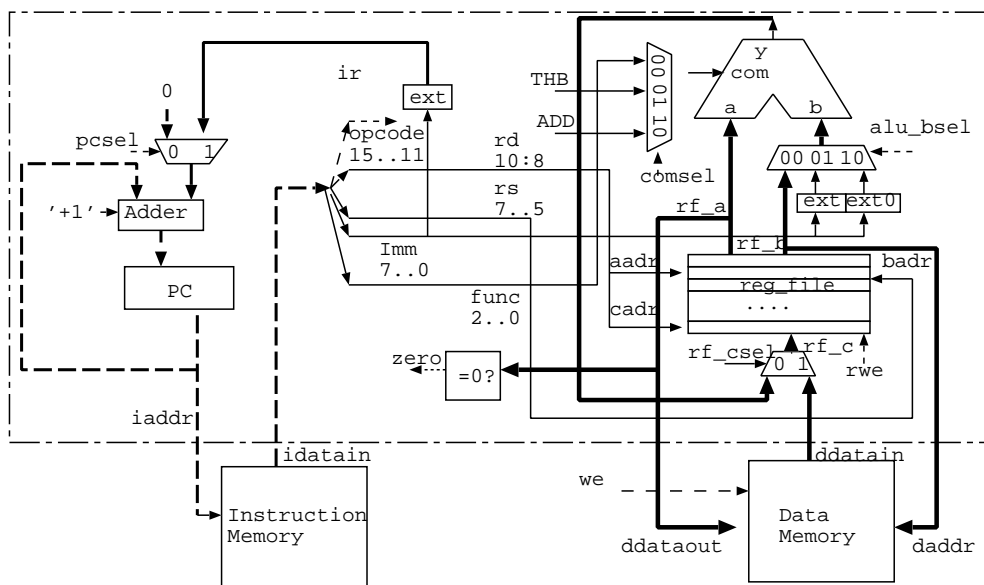


図 1: POCO のデータパス