

方式設計仕様書 (Group18)

1029-28-9483 勝田 峻太郎

1029-28-1547 住江 祐哉

2018 年 5 月 10 日

概要

現状で設計したプロセッサは,5 ステージのマルチサイクル方式のハーバード方式のプロセッサである.

構成要素

主記憶メモリ 1 語 16bit,4096 語のメモリ.

プログラムメモリ 1 語 16bit,4096 語のメモリ.

プログラムカウンタ 実行中の命令のアドレスを保持する 16bit レジスタ.

汎用レジスタ 16bit のレジスタを 8 つ.

条件コードレジスタ 演算命令結果を保持するレジスタ. 条件分岐命令のときに用いる.

命令セットアーキテクチャ

SIMPLE アーキテクチャと基本的には同じである. まず, 以下のような命令の形式を定める.

形式

R 形式

```
#-----|-----|-----|-----|---#
| op1 | rs | rd | op3 | d |
#15---|13--|10--|7----|3--#
```

I 形式

```
#-----|-----|-----|-----|---#
| op1 | ra | rb |    d    |
#15---|13--|10--|7-----#
```

演算命令と停止・入出力命令

演算命令と停止・入出力命令では,R 形式を用い,op1 の値は 11 である.

Table 1: 演算命令のときの各値の対応

name	value
op1	11
rs	レジスタ番号 1
rd	レジスタ番号 2
op3	演算コード

name	value
d	シフト演算のときのシフト数

ただし, 演算コードは以下のように定める.

Table 2: 演算・ 停止・ 入出力コードの対応

code	計算
0000	$\text{in1} + \text{in2}$
0001	$\text{in1} - \text{in2}$
1000	$\text{in1} \& \text{in2}$ (bitwise)
1001	$\text{in1} \text{in2}$ (bitwise)
1010	$\text{in1} \ll \text{i2}$
1011	$\text{in1} \gg \text{in2}$
1100	入力
1101	出力
1111	停止

算術演算

算術演算は, 演算コードが 0 で始まる. このときの動作は,

$$r[rd] = op3(r[rd], r[rs])$$

である.

ビット演算

ビット演算は演算コードが 1 ではじまる. このときの動作は,

$$r[rd] = op3(r[rs], d)$$

である.

停止/入出力命令

停止命令 動作を停止させることができる.

入力命令 ボード上のスイッチを押すことで動作が開始するようにできるようにする.

$$r[rd] = input$$

出力命令 ボード上の 7SEGLED に指定したデータを出力できる.

$$output = r[rs]$$

ロード, ストア命令

ロード, ストア命令はI形式を用いる.

ロード命令

ロード命令のとき, $op1 = 00$ である.

行う動作は,

$$r[ra] = *(r[rb] + \text{signext}(d))$$

ストア命令

ストア命令のとき, $op1 = 01$ である.

行う動作は,

$$*(r[rb] + \text{signext}(d)) = r[ra]$$

即値命令・分岐命令

即値命令, 分岐命令では,I形式を用いる. また, $op1 = 10$ である.

Table 3: 条件分岐・即値命令

ニモニツク	説明	動作	ra	rb
LI	即値ロード	$r[rb] = \text{signext}(d)$	001	rb
B	無条件分岐	$PC = PC + 1 + \text{signext}(d)$	100	dc
BE	条件分岐	$\text{if } (Z) \text{ } PC = PC + 1 + \text{sign ext}(d)$	111	000
BLT	条件分岐	$\text{if } (S \wedge V) \text{ } PC = PC + 1 + \text{sign ext}(d)$	111	001
BLE	条件分岐	$\text{if } (Z \vee (S \wedge V)) \text{ } PC = PC + 1 + \text{sign ext}(d)$	111	010
BNE	条件分岐	$\text{if } (!Z) \text{ } PC = PC + 1 + \text{sign ext}(d)$	111	011

制御回路

基盤上のプッシュスイッチをもちいて, 以下の操作を可能にする.

reset

基盤上のプッシュスイッチを1つ割り当て, スイッチが押されると1, 離されていると0となるように設定する. reset が1になると, プロセッサは初期化され, PC は0に戻る.

exec

基盤上のプッシュスイッチを1つ割り当て、スイッチが押されると1, 離されていると0となるように設定する.

- ・プロセッサが停止状態にあるときにスイッチが押されると, プロセッサは動作を再開する.
- ・プロセッサが動作状態にあるときにスイッチが押されると, プロセッサは実行中の命令を処理したあと, 停止する.

構造と動作

フェーズフローチャート

	p1	p2(IR)	p3	p4	p2(WB)
ADD/SUB	PC++ 命令メモリ (PC)->	r[rs]->alu1 r[rd]->alu2 (書き込み readOutData -> r[rd])	op(alu1, alu2)->aluOutput ex) alu1+alu2	aluOutput -> readOutData	r[writetarget] <- writeval
AND/OR		rd-> (書き込み readOutData -> r[rd])	op3(r[rs], d) -> aluOutput	aluOutput -> readOutData	
SLL/SRL		rs->regaddress	readEnable = 1	MEM[address] -> readOutData	
LD		r[rs]->storedata	writeEnable = 1	MEM[address] <- storeData	
ST		r[rs]<-signext(d)	なし		
LI		isbranch=1	pcsrc=1	なし	
B		isbranch->	pctarget=PC+signext(d)		
BREなど		r[rs]->alu1 (opcode=1101)->	showonLED(alu1)	なし	
OUT					
HLT			未実装		
共通		適切なopcode, 制御コードなど設定 PC->	制御コードの受け渡し		

Figure 1: フェーズフローチャート

フェーズフローチャートを,fig. 1 に示す.

データフローチャート

データフローチャートは,fig. 2, fig. 3,fig. 4,fig. 5, fig. 6 に示される通りである.

また, 図中の色は,

赤色 ほとんど全ての命令で使われるパス

青色 ストア命令で使われるパス

緑色 演算命令などで使われるパス

黄色 演算命令, ロード命令などで使われるパス

の対応をしている.

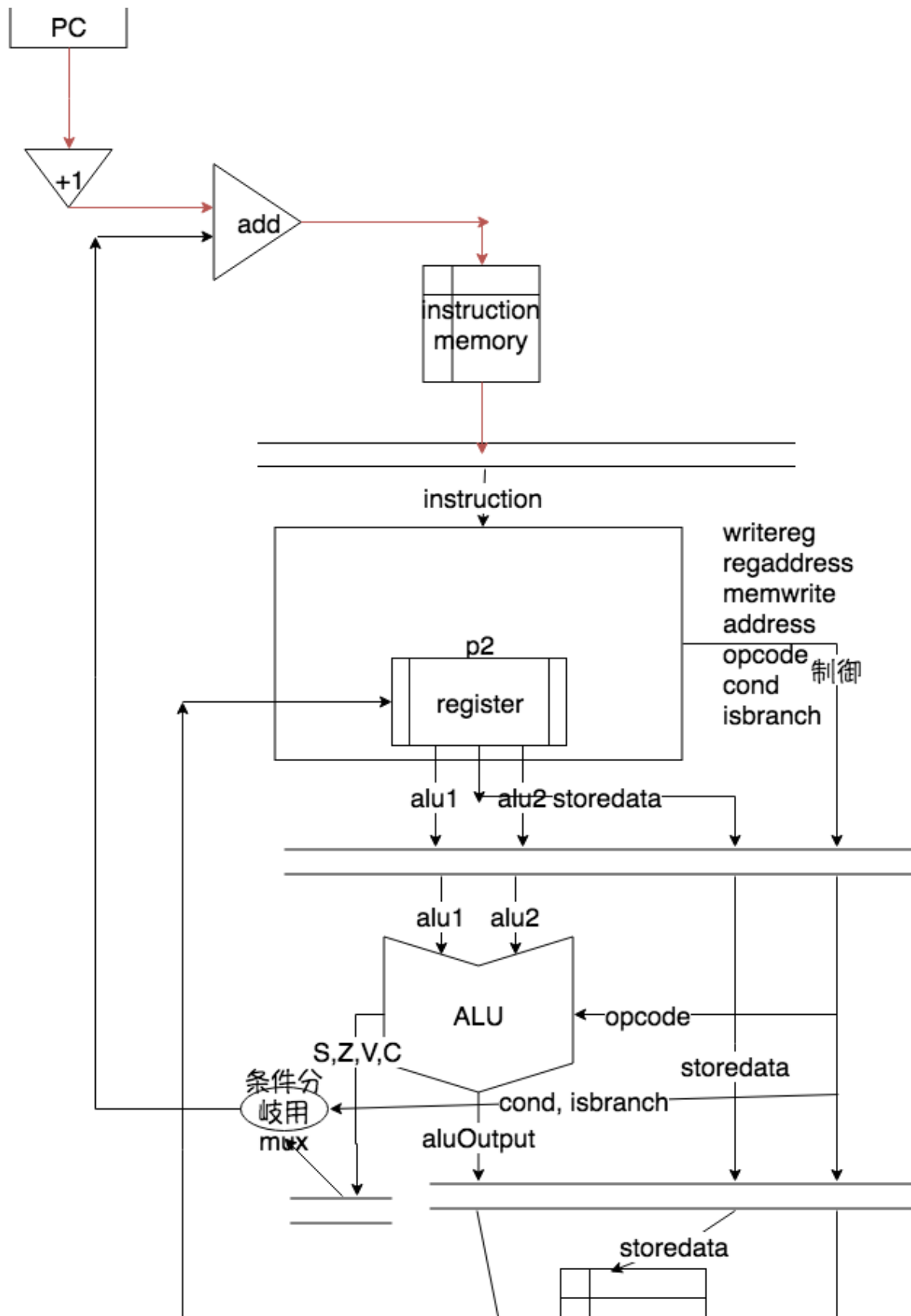


Figure 2: フェーズ 1

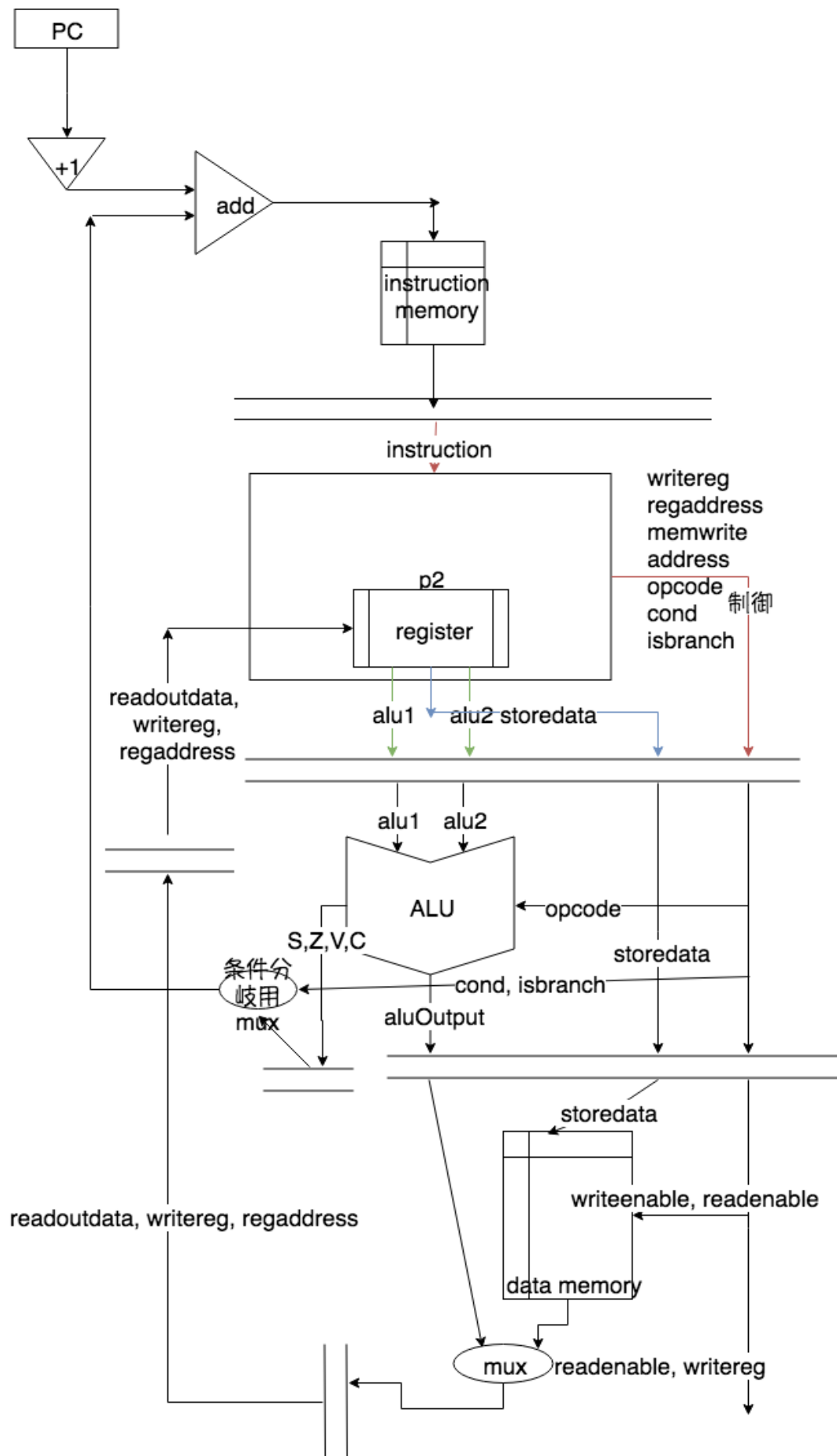


Figure 3: フェーズ 2
6

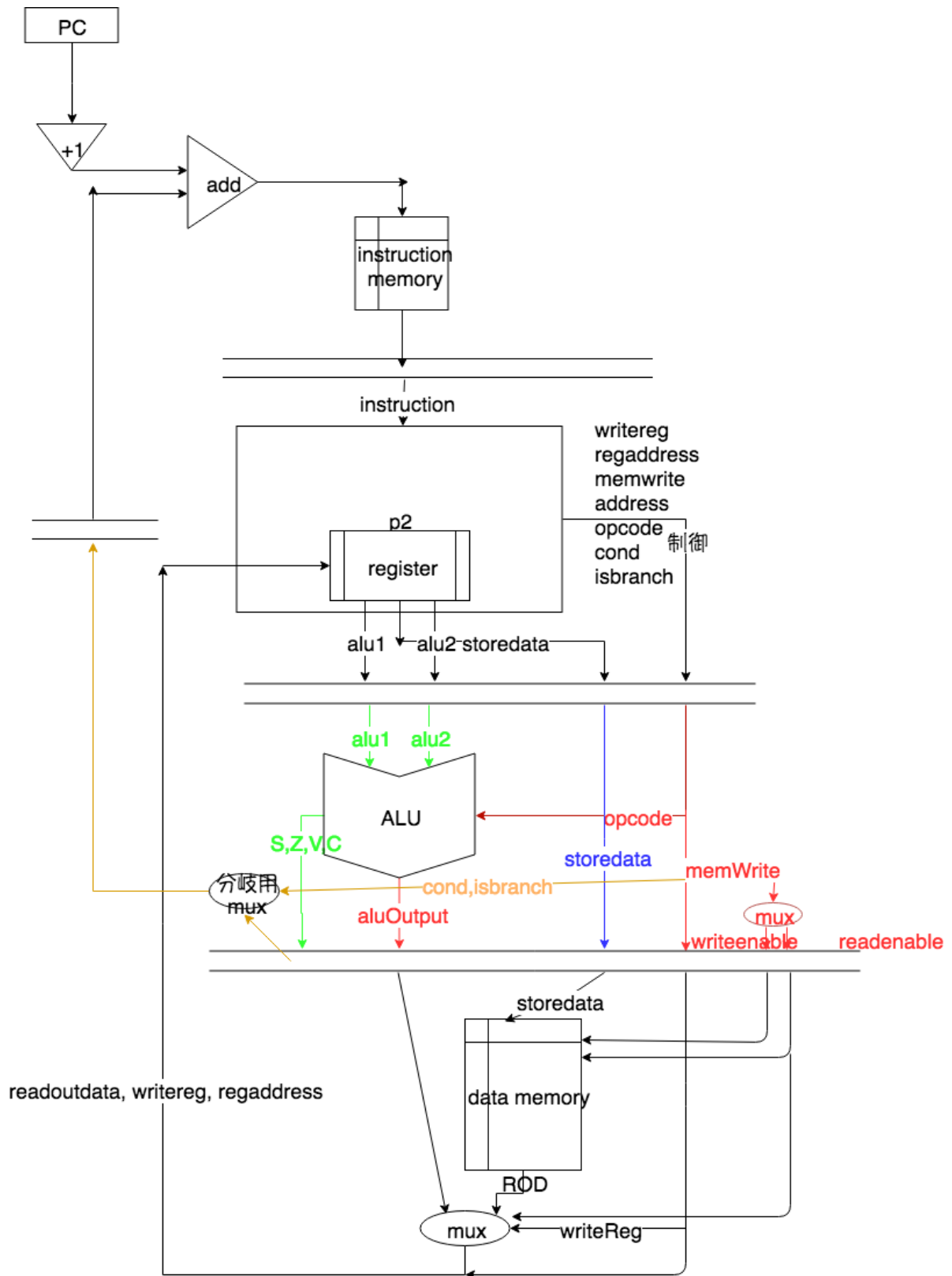


Figure 4: フェーズ 3

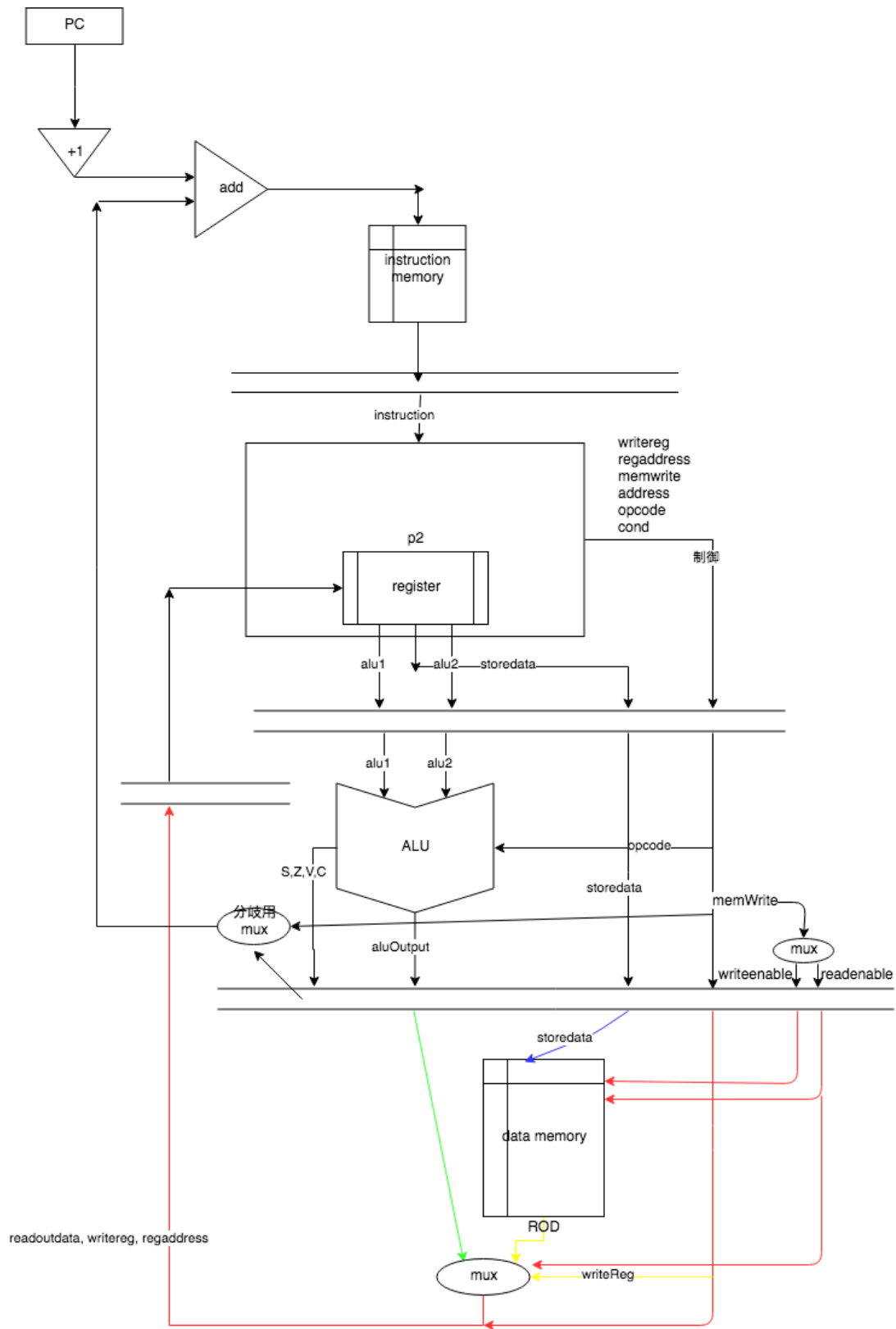


Figure 5: フェーズ 4

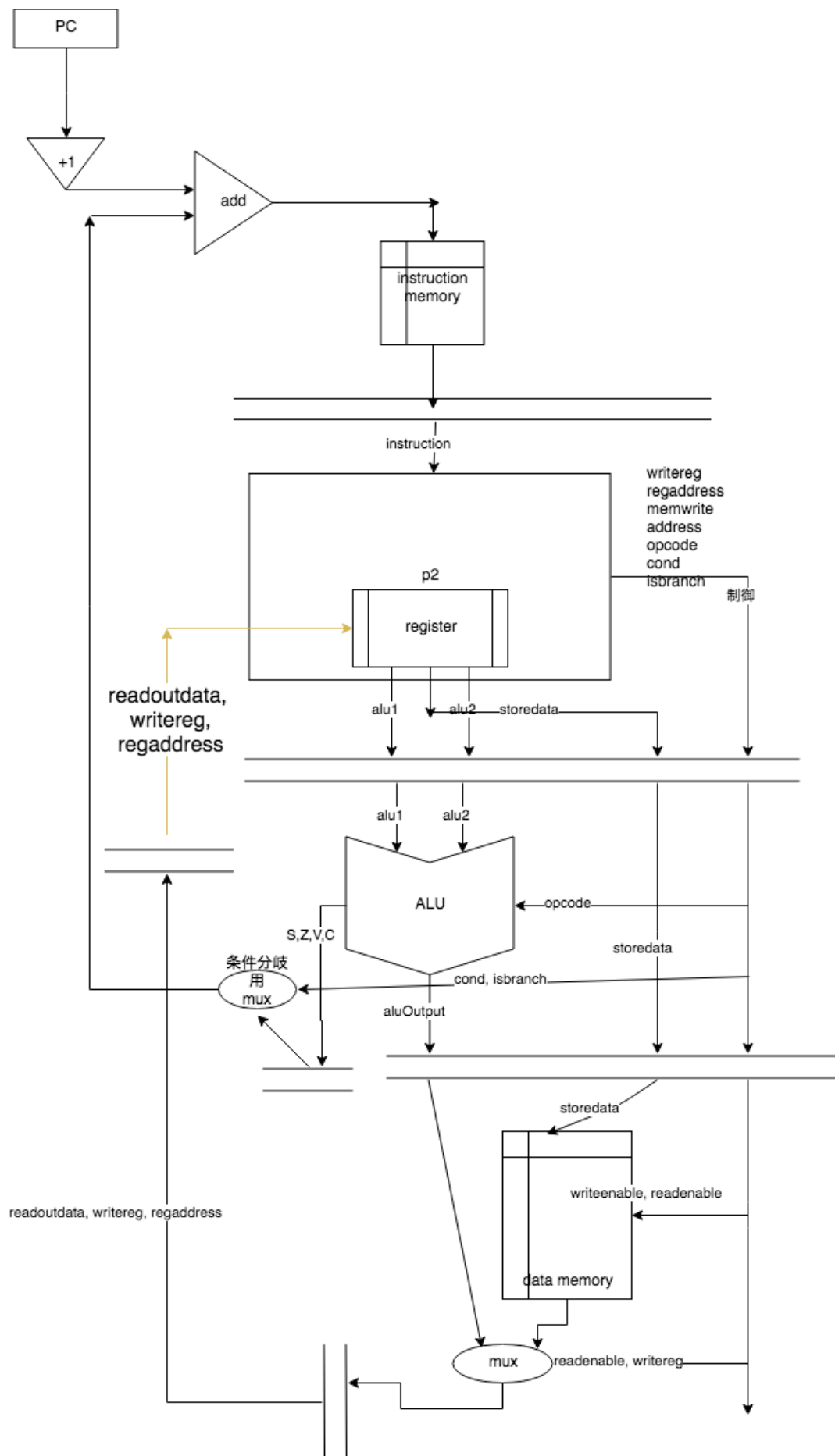


Figure 6: フェーズ 5
9