

平成 29 年度 計算機科学実験及演習 3A  
(3 回生前期学生実験 HW 中間報告)

機能設計仕様書

提出期限：5 月 11 日  
提出日：5 月 11 日

第 22 班

1029272870 谷 勇輝

## 目次

1 コンポーネント分割と担当.....	2
1.1 コンポーネント分割.....	2
1.2 担当.....	2
2 外部仕様.....	3
2.1 EX.....	3
2.1.1 概要.....	3
2.1.2 構造.....	3
2.1.3 動作.....	4
2.2 MA.....	7
2.2.1 概要.....	7
2.2.2 構造.....	7
2.2.3 動作.....	8
2.3 WB.....	9
2.3.1 概要.....	9
2.3.2 構造.....	9
2.3.3 動作.....	10
2.4 TestEnvironment.....	11
2.4.1 概要.....	11
2.4.2 構造.....	11
2.4.3 動作.....	12
3 内部仕様.....	13
3.1 EX.....	13
3.1.1 構造.....	13
3.1.2 動作.....	13
3.2 MA.....	15
3.2.1 構造.....	15
3.2.2 動作.....	15
3.3 WB.....	16
3.3.1 構造.....	16
3.3.2 動作.....	16
3.4 Test Environment.....	17
3.4.1 構造.....	17
3.4.2 動作.....	18

# 1 コンポーネント分割と担当

## 1.1 コンポーネント分割

プロセッサは、最上位レベルの分割として、図 1 に示す 8 つのコンポーネントで構成される。各コンポーネント内部の設計（次レベルの分割等）はそれぞれの設計担当者が行う。

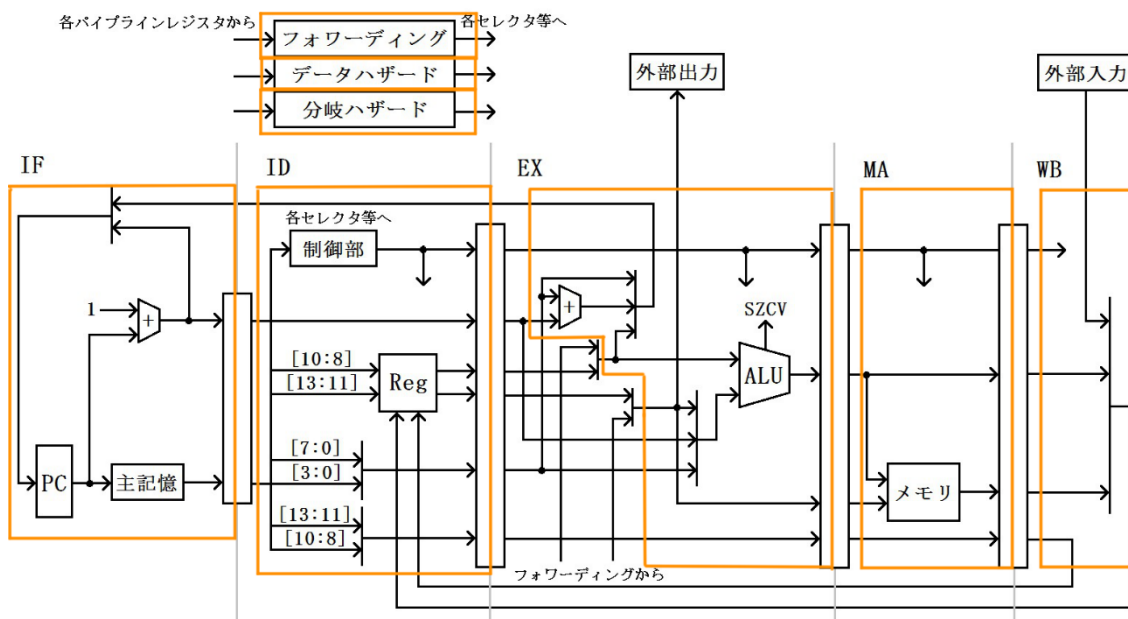


図 1 コンポーネント分割

## 1.2 担当

2017 年 5 月 11 日現在、私が設計・実装を担当したプロセッサのコンポーネントは以下のとおりである。

- EX
- MA
- WB

また、本体プロセッサとは別に、以下の設計・実装も担当した。

- FPGA ボード入出力用環境 TestEnvironment

## 2 外部仕様

### 2.1 EX

#### 2.1.1 概要

EX モジュールは、演算を担当する EX フェーズの一連の機能を提供する。

上流モジュール ID から制御に応じて同期的に動作し、内包する ALU、シフタを用いて下流モジュール MA へ演算結果を渡す。また、分岐の判断を行い IF に結果を伝搬する。

#### 2.1.2 構造

全体内の位置づけを図 2 に、入力構造を表 1 に、出力構造を表 2 に示す。

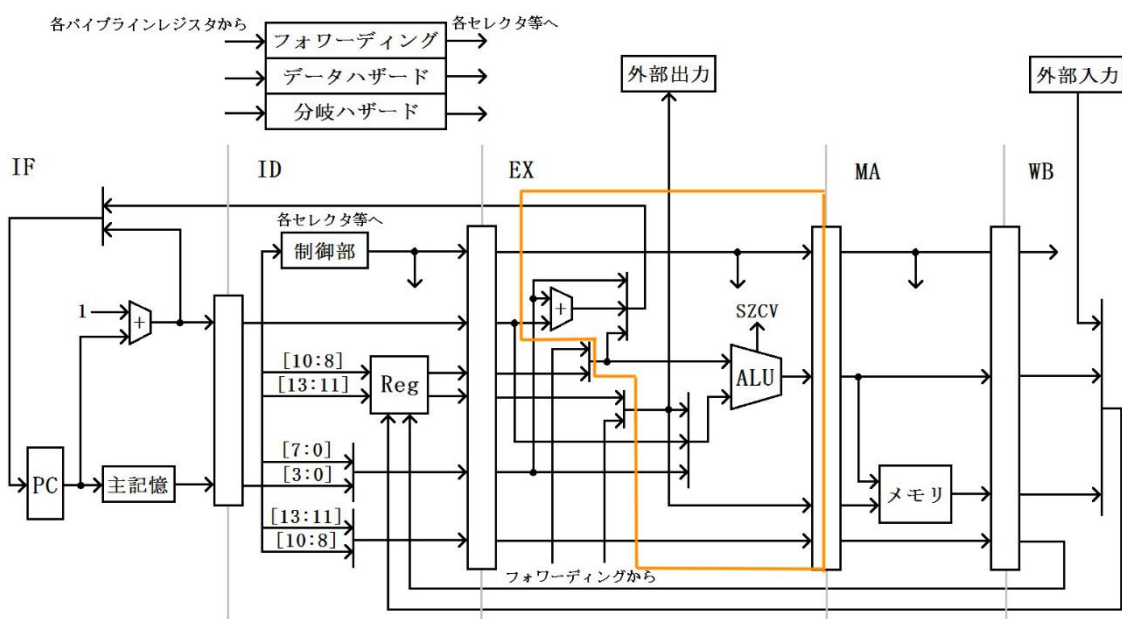


図 2 EX コンポーネントの位置づけ

表 1 入力構造 (EX)

入力信号名	bit 幅	接続	内容
clock	1		クロック信号
reset	1		リセット信号 (負論理)
PC	16	ID	プログラムカウンタ値+1
WBaddress	3	ID	レジスタ書き込みアドレス (or 分岐種類)

control	6	ID	制御信号
ALUcontrol	4	ID	演算器の機能制御コード
immediate	16	ID	即値 (or 演算器の第 2 入力)
Rs_Ra	16	ID	演算器の第 2 入力 (or メモリの指定アドレス)
Rd_Rb	16	ID	演算器の第 1 入力

表 2 出力構造 (EX)

出力信号名	bit 幅	接続	同期	内容
IMnextPC_	16	IF	×	分岐先プログラムカウンタ値
branch_	1	IF	×	分岐判定 (0:分岐無し 1:分岐有り)
Wbaddress_	3	MA	○	レジスタ書き込みアドレス
control_	4	MA	○	制御信号
result_	16	MA	○	演算結果
Ra_	16	MA	○	メモリの指定アドレス
ConditonCode_	4	MA	○	コンディションコード (S,Z,C,V)

### 2.1.3 動作

#### 分岐処理 (ID→EX→IF)

- PC (プログラムカウンタ+1 の値) + immediate (即値) を IMnextPC\_ (分岐先 PC アドレス) として出力
- control [4] (分岐制御信号 Branch) が 1 (分岐命令) であり、直前の演算のコンディションコードの状態が Wbaddress (分岐種類コード cond) の示す分岐条件 (表 3) に合致するならば、branch\_ (分岐判定) に 1 を出力

表 3 分岐コードと分岐条件の対応

分岐コード	分岐条件	主な使用法
000	Z が 1	直前の演算結果が 0
001	S XOR V が 1	直前の減算結果が理論的に負
010	Z が 1 または S XOR V が 1	直前の減算結果が理論的に 0 以下
011	Z が 0	直前の演算結果が 0 でない

### 演算処理 (ID→EX→MA)

- control[5] (第2入力制御信号 ALUSrc)が 0 ならば演算に使用する第2入力として Rs\_Ra を選択
- control[5] (第2入力制御信号 ALUSrc)が 1 ならば演算に使用する第2入力として immediate (即値)を選択
- 第1入力を Rd\_Rb、第2入力を選択した信号として、ALUControl (演算器機能制御コード) の値に従って演算を行い(表4)、clock と同期して演算結果を result\_に出力

表 4 演算器の機能制御コード

機能コード	動作	代用コード
0000	加算 (+)	
0001	減算 (-)	0101
0010	論理積 (AND)	
0011	論理和 (OR)	
0100	排他的論理和 (XOR)	
0110	第2入力を出力	1100 ,1101 ,1111
1000	左論理シフト	
1001	左循環シフト	
1010	右論理シフト	
1011	右算術シフト	

- 演算結果に従って4つのコンディションコード(表5)を設定し、clock と同期して ConditionCode\_に出力

表 5 コンディションコード

コード	内容	例外
S	負ならば 1	
Z	0 ならば 1	
C	桁上げがあれば 1	論理演算、第2入力出力、左循環シフト演算では0 その他のシフト演算では最後にシフトされた値で判定
V	オーバーフローで 1	シフト演算では 0

- WAddress (レジスタ書き込みアドレス)を clock と同期して WAddress\_にそのまま出力
- Rs\_Ra(メモリの書き込みアドレス)を clock と同期して Ra\_にそのまま出力
- control のうち、使用していない下 4bit を clock と同期して control\_に出力

#### リセット機構 (外部→EX→MA)

- clock 同期時 reset が 0 であれば全ての同期式出力を 0 にする

## 2.2 MA

### 2.2.1 概要

MA モジュールは、データメモリの書き込み、読み出しを管理する MA フェーズの一連の機能を提供する。

上流モジュール EX を伝播してきた制御信号に応じて同期的に動作し、内包するメモリを操作して下流モジュール WB に結果を伝搬する。

### 2.2.2 構造

全体内の位置づけを図 3 に、入力構造を表 6 に、出力構造を表 7 に示す。

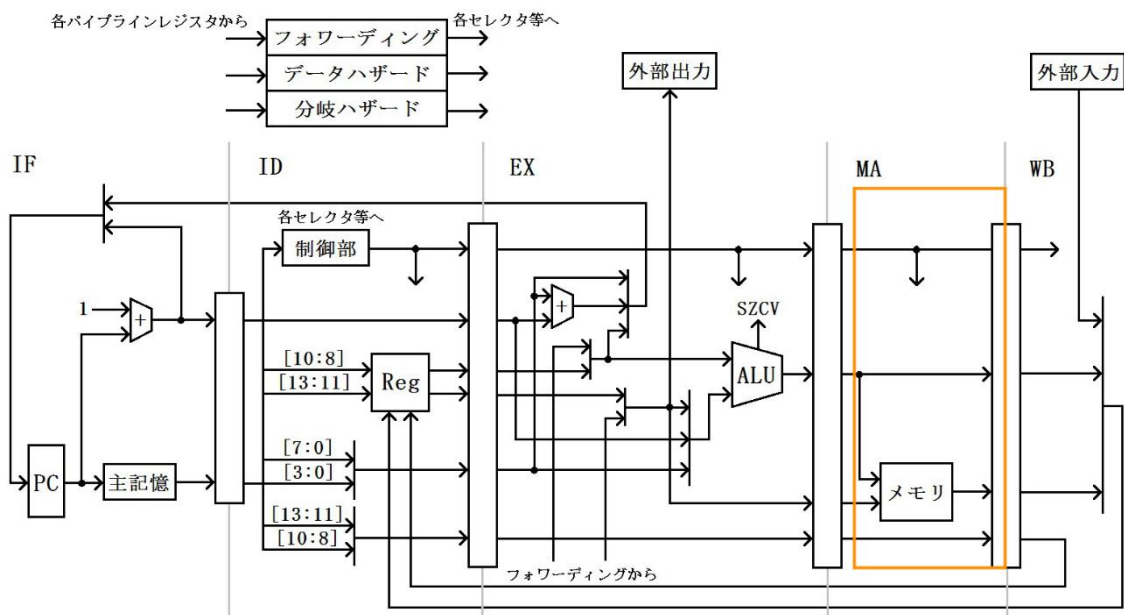


図 3 MA コンポーネントの位置づけ

表 6 入力構造 (MA)

入力信号名	bit 幅	接続	内容
clock	1		クロック信号
reset	1		リセット信号 (負論理)
Wbaddress	3	EX	レジスタ書き込みアドレス
control	4	EX	制御信号
ALUresult	16	EX	演算の結果
Ra	16	EX	メモリの指定アドレス



表 7 出力構造 (MA)

出力信号名	bit 幅	接続	同期	内容
Wbaddress_	3	WB	○	レジスタ書き込みアドレス
control_	2	WB	○	制御信号
ALUresult_	16	WB	○	演算結果
LDresult_	16	WB	○	メモリ読み出し結果

### 2.2.3 動作

#### メモリ処理 (EX→MA→WB)

- control[3](メモリ読み出し制御信号 MemRead) が 1 ならばメモリから Ra(メモリ指定アドレス)番地に格納されているデータを読み出し、clock と同期して LDresult\_に出力
- control[2](メモリ書き込み制御信号 MemWrite) が 1 ならばメモリの Ra(メモリ指定アドレス)番地に ALUresult を格納
- Wbaddress (レジスタ書き込みアドレス)を clock と同期して Wbaddress\_にそのまま出力
- ALUresult (演算結果)を clock と同期して ALUresult\_にそのまま出力
- control のうち、使用していない下 2bit を clock と同期して control\_に出力

#### リセット機構 (外部→MA→WB)

- clock 同期時 reset が 0 であれば全ての同期式出力を 0 にする

## 2.3 WB

### 2.3.1 概要

WB モジュールは、レジスタ書き込みを行う WB フェーズの機能の一部を提供する。

上流モジュール MA を伝播してきた制御信号に応じて動作し、ID モジュールにメモリ書き込みを依頼する。クロックと書き込みの管理は ID モジュールに委任する。

### 2.3.2 構造

全体内の位置づけを図 4 に、入力構造を表 8 に、出力構造を表 9 に示す。

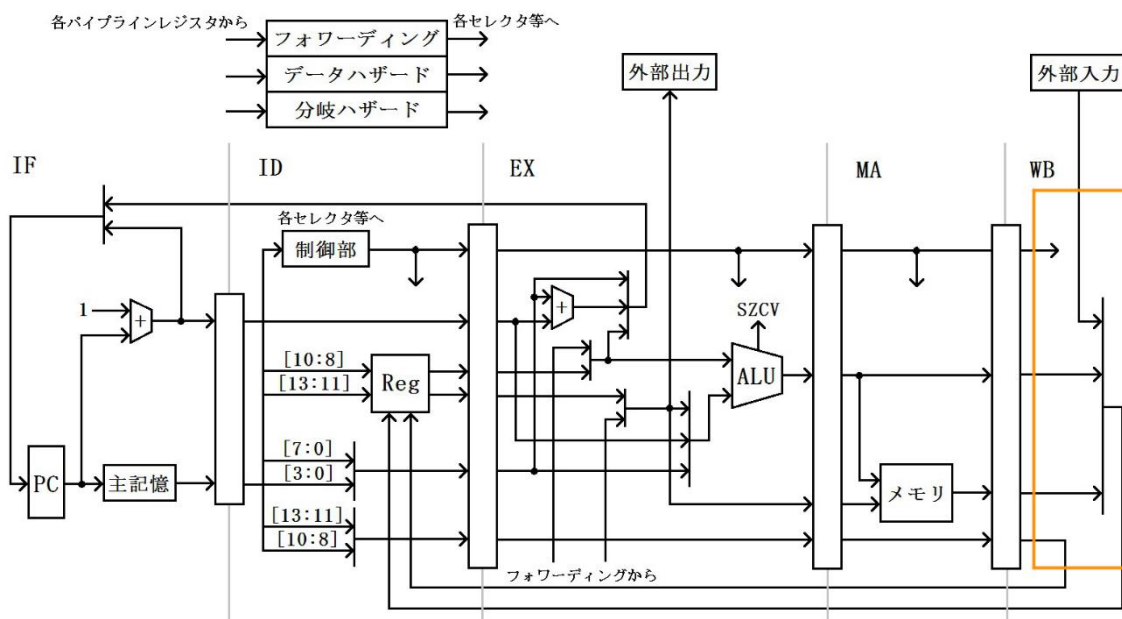


図 4 WB コンポーネントの位置づけ

表 8 入力構造 (WB)

入力信号名	bit 幅	接続	内容
clock	1		クロック信号 (拡張時用)
WBaddress	3	MA	レジスタ書き込みアドレス
control	2	MA	制御信号
ALUresult	16	MA	演算結果
LDresult	16	MA	メモリ読み出し結果

表 9 出力構造 (WB)

出力信号名	bit 幅	接続	同期	内容
RegWrite_	1	ID	×	レジスタ書き込み制御
WBaddress_	3	ID	×	レジスタ書き込みアドレス
WBdata_	16	ID	×	レジスタ書き込みデータ

### 2.3.3 動作

#### レジスタ書き込みデータの選択 (MA→WB→ID)

- control[0] (レジスタ書き込みデータ制御信号 MemtoReg)が 0 ならば、ALUresult(演算結果)を WBdata\_に出力
- control[0] (レジスタ書き込みデータ制御信号 MemtoReg)が 1 ならば、LDresult(メモリ読み出し結果)を WBdata\_に出力
- WBaddress (レジスタ書き込みアドレス)を WBaddress\_にそのまま出力
- control のうち、使用していない 1bit (第 1 番信号)を RegWrite\_に出力

## 2.4 TestEnvironment

### 2.4.1 概要

FPGA ボード (PowerMedusa MU500-RX/RK, 拡張 MU500-7SEG)のほぼ全ての入出力を簡易に利用できる環境を提供する。主な特徴は以下の3点である。

- 7segLED は、信号線を擬似出力に接続するだけで16進数を表示可能。
- 拡張ボードを含む全ての 7segLED について、任意のパターンを独立に表示可能。(72 個の 7segLED が全て使用可能となる)
- 全ての入力と拡張 LED ランプを除く全ての出力が利用可能

各種入力は FPGA ボードの入力に、各種出力は FPGA ボードの出力に対応している。接続対象のモジュールを TestEnvironment モジュール(以下、環境モジュールと示す)の中に配置し、接続対象モジュールの入出力を環境モジュールの入出力や擬似入出力に接続することで、あたかも外部の入出力に直接繋いだような状態とできる。

### 2.4.2 構造

入力構造を表 10 に、出力構造を表 11 に示す。内部の擬似入出力については内部仕様に記す。

表 10 入力構造 (TestEnvironment)

入力名	幅	内容	接続ピン
clock	1	20MHz 固定クロック	A12
clock2	1	可変クロック	B12
dipSW_A	8	ディップスイッチ群A	G11,G10,F10,E10,D10,C10,B10,A10
dipSW_B	8	ディップスイッチ群B	F13,E13,D13,C13,B13,A13,F11,E11
rotSW_A	4	ローテーションスイッチA	F14,E14,B14,A14
rotSW_B	4	ローテーションスイッチB	D15,C15,B15,A15
pushSW_A	5	プッシュスイッチ群A行	A16,H15,G15,F15,E15
pushSW_B	5	プッシュスイッチ群B行	A17,G16,F16,E16,B16
pushSW_C	5	プッシュスイッチ群C行	B18,A18,D17,C17,B17
pushSW_D	5	プッシュスイッチ群D行	A20,D19,C19,B19,A19

表 11 出力構造 (TestEnvironment)

出力名	幅	内容	接続ピン
LED7segA_D	8	UI ボード 7segLED,A~D	A3,B6,A6,A5,B4,B3,A4,B5
LED7segE_H	8	UI ボード 7segLED,E~H	C6,F7,E7,C7,B7,D6,A7,D7
slctA_D	4	UI 7seg セレクタ A~D	E6,E5,C4,C3
slctE_H	4	UI 7seg セレクタ E~H	G7,G8,G9,H10
LED	8	UI ボード LED ランプ群	F9,E9,B9,A9,F8,C8,B8,A8
BZ	1	UI ボード ブザー	B20
LED7ep0	8	拡張ボード 7segLED 0 列	Y6,AB5,W6,AB4,AA5,AA4,V5
LED7ep1	8	拡張ボード 7segLED 1 列	U8,AA7,T8,V7,Y7,U7,V6
LED7ep2	8	拡張ボード 7segLED 2 列	V9,AB8,U9,W8,AA8,V8,Y8,AB7
LED7ep3	8	拡張ボード 7segLED 3 列	AB10,W10,AA10,AB9,V10,AA9,U10,T9
LED7ep4	8	拡張ボード 7segLED 4 列	V13,V12,U13,V11,U12,U11,T12,Y10
LED7ep5	8	拡張ボード 7segLED 5 列	V14,R14,U14,Y13,AB13,W13,AA13,T13
LED7ep6	8	拡張ボード 7segLED 6 列	W15,T15,V15,AA14,R15,W14,AB14,T14
LED7ep7	8	拡張ボード 7segLED 7 列	AB16,U16,AA16,AB15,T16,AA15,R16,U15
slctEp	8	拡張ボード 7seg セレクタ	V16,AA17,W17,AB17,Y17,AB19,AA18,AA19

### 2.4.3 動作

#### FPGA ボードの使用 (ボード入力→TestEnvironment→ボード出力)

- ボードの入力を環境モジュール内に取り込み、作成した内部機構の動作に応じてボードの出力に表示する。

# 3 内部仕様

## 3.1 EX

### 3.1.1 構造

ブロック図を図 5 に示す。

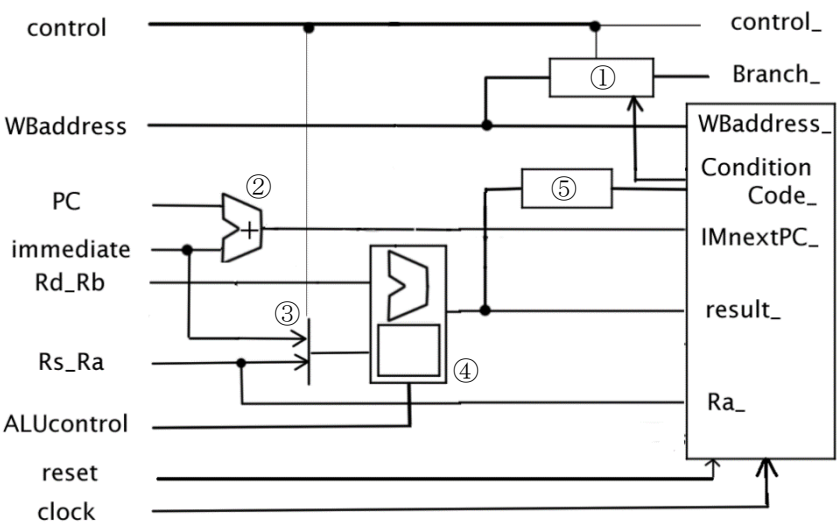


図 5 EX ブロック図

### 3.1.2 動作

#### ① 分岐判断 (control, WAddress → Branch\_)

$$\text{control}[4]==1 \wedge \text{judge}(\text{WAddress}, \text{ConditionCode}_) \rightarrow \text{Branch}_ = 1$$

bool 関数 judge を表 12 に示す。(ConditionCode\_は CC と略記)

表 12 関数 judge

WAddress(分岐コード)	関数値	内容
000	CC[2]	Z が 1 で true
001	CC[3] ^ CC[0]	S XOR V が 1 で true
010	CC[2]    (CC[3]^CC[0])	上 2 つのいずれかで true
011	!CC[2]	Z が 0 で true

② 分岐先アドレス計算 (PC, immediate → IMnextPC )

$$\text{IMnextPC} = \text{PC} + \text{immediate}$$

③ 第2入力選択 (immediate, Rs\_Ra, control → In2)

$\text{control} == 0 \rightarrow \text{In2} = \text{Rs\_Ra}$ $\text{control} == 1 \rightarrow \text{In2} = \text{immediate}$
---

④ 演算 (Rd\_Rb, In2, ALUcontrol → result\_)

外部仕様の表4 (演算器の機能制御コード) に従う演算@の元で、 $\text{result} = \text{Rd\_Rb} @ \text{In2}$
--

演算器は、加算、減算、各種論理演算を実装した **ALU** と表4内にある4種のシフト演算を提供するバレルシフタから成る。

⑤ コンディションコード設定 (result → ConditionCode\_)

外部仕様の表5 (コンディションコード) に従って設定する $\text{ConditionCode\_}[3] = \text{S}$ $\text{ConditionCode\_}[2] = \text{Z}$ $\text{ConditionCode\_}[1] = \text{C}$ $\text{ConditionCode\_}[0] = \text{V}$
---

設定の詳細は SIMPLE の仕様による。

その他

$\text{control\_} = \text{control} [3..0]$ $\text{WBaddress\_} = \text{WBaddress}$ $\text{Ra\_} = \text{Rs\_Ra}$
--

## 3.2 MA

### 3.2.1 構造

ブロック図を図 6 に示す。

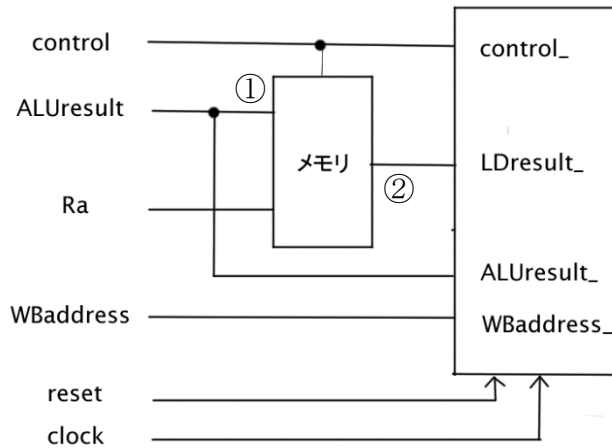


図 6 MA ブロック図

### 3.2.2 動作

①メモリ書き込み (control, Ra, ALUresult → Mem)

$\text{control}[3] == 1 \rightarrow (\text{ALUresult} \Rightarrow \text{Mem}(\text{Ra}))$
---

②メモリ読み出し (control, Ra → LDresult\_)

$\text{control}[2] == 1 \rightarrow \text{LDresult}_ = \text{Mem}(\text{Ra})$
---

その他

$\begin{aligned} \text{control}_ &= \text{control}[1..0] \\ \text{ALUresult}_ &= \text{ALUresult} \\ \text{WBaddress}_ &= \text{WBaddress} \end{aligned}$
---



### 3.3 WB

#### 3.3.1 構造

ブロック図を図 7 に示す。

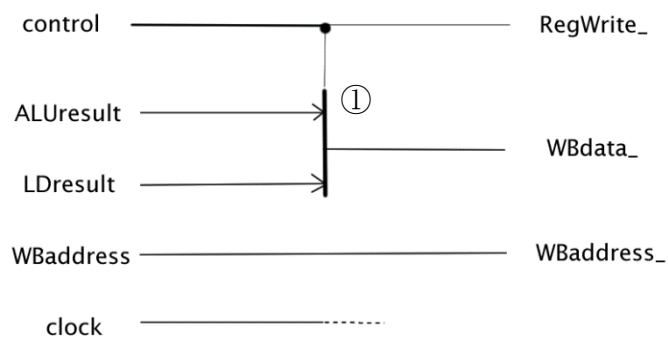


図 7 WB ブロック図

#### 3.3.2 動作

① データの選択 (ALUresult, LDresult, control → WBdata\_)

$\text{control}[0] == 0 \rightarrow \text{WBdata\_} = \text{ALUresult}$ $\text{control}[0] == 1 \rightarrow \text{WBdata\_} = \text{LDresult}$
---

その他

$\text{RegWrite\_} = \text{control}[1]$ $\text{WBaddress\_} = \text{WBaddress}$
--

## 3.4 Test Environment

### 3.4.1 構造

大きく分けて、①擬似入力生成部、②ユーザースペース、③擬似出力生成部の3つの領域からなる。環境モジュールの利用者は、②内部のみを編集し、実行を行う。(図9)

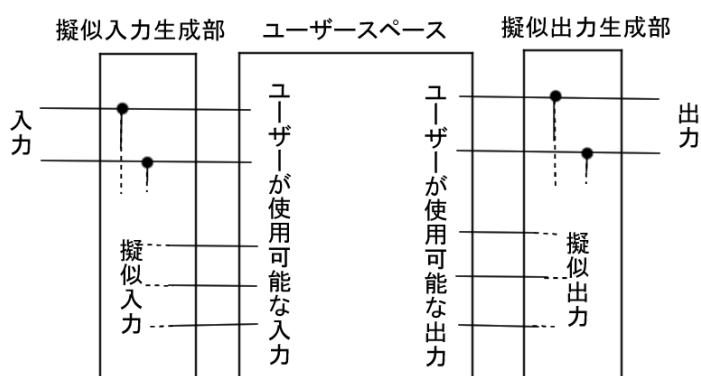


図9 TestEnvironmentの主な構成

### 擬似入力

環境モジュール内にて入力と同等の役割をもつ擬似入力を表13に示す。擬似入力のビット幅は全て1 bitである。

表13 擬似入力

擬似入力名	内容	実装
clock_x4	1/4 倍速クロック信号	4 進カウンタに clock を通す
clock_x8	1/8 倍速クロック信号	8 進カウンタに clock を通す
clock_x10	1/10 倍速クロック信号	10 進カウンタに clock を通す
clock_x100	1/100 倍速クロック信号	100 進カウンタに clock を通す
clock_x1000	1/1000 倍速クロック信号	1000 進カウンタに clock を通す
clock_x2p16	1/(2 の 16 乗)倍速クロック信号	2 <sup>16</sup> 進カウンタに clock を通す
vdd	電源	恒久的に値が 1 のレジスタ

### 擬似出力

環境モジュール内にて出力と同等の役割をもつ擬似出力を表14に示す。擬似出力のビット幅は全て8 bitである。

擬似出力名	内容	実装
LED_A	7segLED (UI) の独立表示	LED4set モジュールを使用して時間差表示
.....LED_H		
LED_exA[0..7]	7segLED (拡張) の独立表示	LED8set モジュールを使用して時間差表示
.....LED_exH[0..7]		

### 3.4.2 動作

#### LED4set モジュール (7segLED の独立表示)

7segLED は、4つ（拡張ボードは8つ）の表示盤が組になっており、組につき1つの表示パターンしか指定できない。そのため通常は同じ組の表示盤には同じパターンしか表示できない。しかし、短い時間間隔で表示盤を指定するセレクトと表示パターンを切り替えていくことにより、あたかも同じ組の表示盤に別のパターンが表示されているように見せることはできる。この機構を提供するのがLED4set モジュールである。

- LED4set は、同組の 7segLED に表示したい4つの表示パターンと clock を入力すると、1clock ごとに切り替わる1つの表示パターンと同じタイミングで切り替わるセレクト信号を出力する。
- TestEnvironment では入力クロックとして 5MHz クロックである擬似入力 clock\_x4 を使用する。