

平成 29 年度 計算機科学実験及演習 3 HW

中間報告

方式設計仕様書

提出日
5 月 11 日

グループ 22
1029277526 白石竜也
1029272870 谷勇輝

1 概要

SIMPLE/B のハードウェア設計を 5 段パイプライン化、さらに即値加算命令と関数呼び出しの機能を追加したプロセッサを設計する。

2 命令セットアーキテクチャ

SIMPLE の命令セット・アーキテクチャからの変更点を説明する。演算/入出力命令形式、ロード/ストア命令形式の命令について変更はない。即値ロード/無条件分岐命令形式、条件分岐命令形式の命令については 3 点変更がある。変更したものを図 2.1、図 2.2 に示す。

まず、無条件分岐命令 B を条件分岐命令形式の中に移動させている。この理由は ID ステージの動作のところで説明する。次に、即値加算命令 ADDI を即値ロード/無条件分岐命令形式に加えている。ADDI ではレジスタ Rb の値と即値 d の加算の結果をレジスタ Rb に格納する。また、ADD 命令と同様に条件コードの設定を行う。

さらに、関数呼び出しのための JAL(Jump And Link) 命令と JR(Jump Register) 命令を条件分岐命令形式に加えている。JAL 命令では PC + 1 の値をレジスタ 0 に退避するとともに、d で指定したアドレスにジャンプする。JR 命令ではレジスタ 0 に退避されているアドレスにジャンプする。PC + 1 の退避先はレジスタ 0 に固定され、レジスタ 0 はこの 2 つの命令でのみ書きかわる。

15 14 13	11 10	8 7	0
10	op2	Rb	d
mnemonic	op2	function	
LI Rb, d	000	r[Rb] = sign_ext(d)	
ADDI Rb, d	001	r[Rb] = r[Rb] + sign_ext(d)	
(reserved)	010		
(reserved)	011		
(reserved)	100		
(reserved)	101		
(reserved)	110		
条件分岐命令	111	表 2.2 参照	

図 2.1 即値ロード/無条件分岐命令

15	14	13	11	10	8	7	0
10	111	cond	d				
mnemonic			cond	function			
BE d			000	if (Z) PC = PC + 1 + sign_ext(d)			
BLT d			001	if (S^V) PC = PC + 1 + sign_ext(d)			
BLE d			010	if (Z S^V) PC = PC + 1 + sign_ext(d)			
BNE d			011	if (!Z) PC = PC + 1 + sign_ext(d)			
B d			100	PC = PC + 1 + sign_ext(d)			
JAL d			101	r[0] = PC + 1, PC = zero_ext(d)			
JR d			110	PC = r[0]			
(reserved)			111				

図 2.2 条件分岐命令形式

3 構造と動作

3.1 構造

ハードウェアのブロック図を図 3.1 に示す。SIMPLE/B の p1~p5 にあたるフェーズをそれぞれ IF、ID、EX、MA、WB とし、ALU はシフト演算も兼ねるものとしてまとめて描いてある。このハードウェアは 5 段パイプライン化されているので、5 つの命令が各フェーズで同時並行的に実行される。

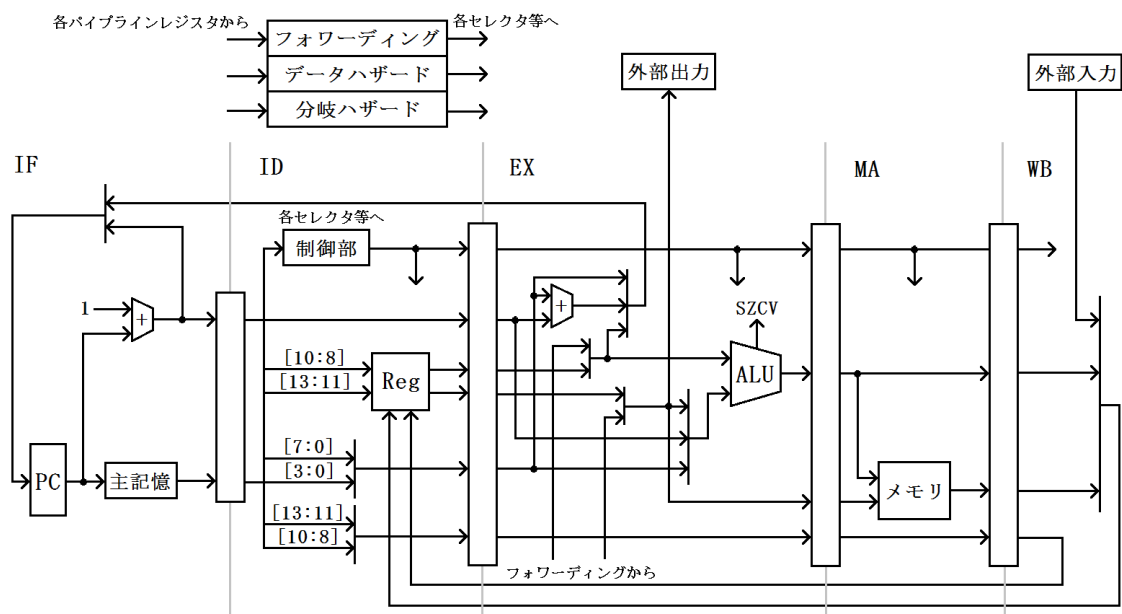


図 3.1 ブロック図

3.2 動作

IF ステージ

IF ステージのブロック図を図 3.2 に示す。IF では、PC の値をアドレスとして命令を取り出す。また、PC+1 の値も同様にパイプラインレジスタに入れる。これは EX で分岐先アドレスの計算に使用される。そして最後に、PC の値を更新する。

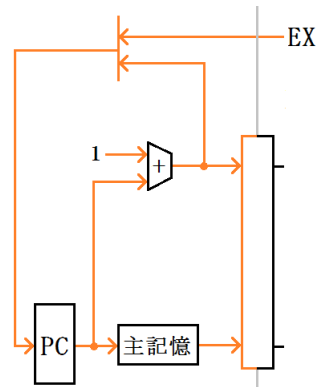


図 3.2 IF ステージ

ID ステージ

ID ステージのブロック図を図 3.3 に示す。ID ではまず、必要な場合にレジスタへの書き込みを行う。その後、レジスタ読み込み、即値のビット拡張をして EX での演算に必要なデータを揃える。また、回路全体の制御信号もこのステージで生成される。さらに、書き込みアドレスをパイプラインレジスタに入れる。これは WB で使用される。ただし、条件分岐命令の場合はここに cond の値を入れて渡す。これは EX でどの条件を計算するか判断するのに使用される。B 命令を条件分岐命令形式に移した理由は、この cond を渡すだけで、EX でどの分岐命令か判断できるようにするためである。IF から渡される PC+1 は使用しないのでそのまま EX へ渡す。

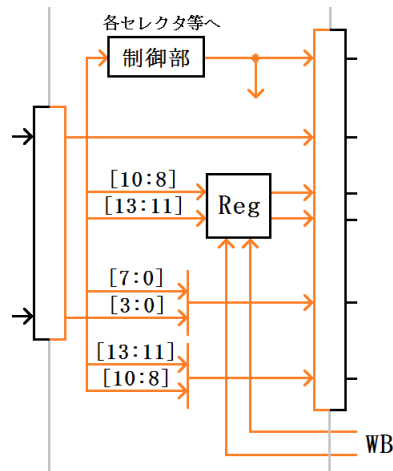


図 3.3 ID ステージ

EX ステージ

EX ステージのブロック図を図 3.4 に示す。EX では演算と条件コードの設定を行い、演算結果をパイプラインレジスタに入れる。さらに、レジスタの値の片方をパイプラインレジスタに入れる。これは MA でメモリへの書き込みアドレスとして使用される。また、分岐条件、分岐先アドレスを計算して IF に送る。OUT 命令による外部出力もこのステージで行う。ID から渡される書き込みアドレスは使用しないので、そのままパイプラインレジスタに入れる。セレクトが多いが、各命令でどの値が選択されるかは後述するフェーズ・フロー・チャートを参考。

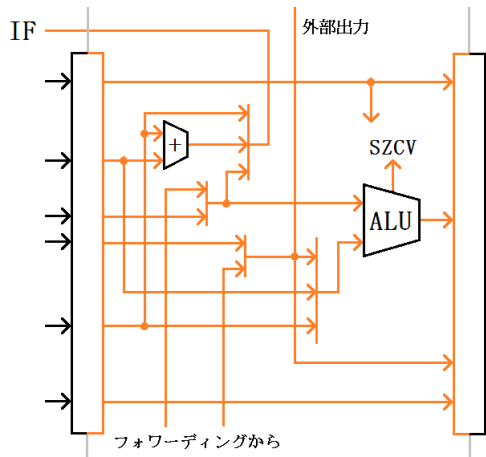


図 3.4 EX ステージ

MA ステージ

MA ステージのブロック図を図 3.5 に示す。MA では EX での演算結果と、メモリから読みだした値をパイプラインレジスタに入れる。書き込みアドレスはここでも使用しないので、WB にそのまま送る。

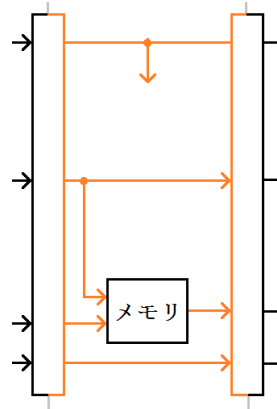


図 3.5 MA ステージ

WB ステージ

WB ステージのブロック図を図 3.6 に示す。WB ではレジスタ書き込みを行うが、これは実際には ID で行われるので、ここでは書き込みアドレスと書き込むデータを ID へ送るだけである。また、外部入力の値もここで書き込むデータとして選択され ID へ送られる。

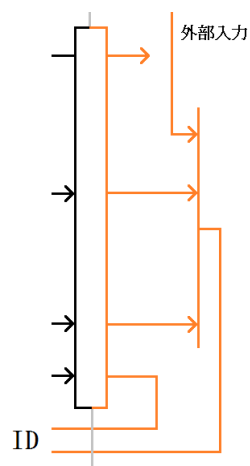


図 3.6 WB ステージ

3.3 フェーズ・フロー・チャート

このプロセッサで実装するすべての命令のフェーズ・フロー・チャートを図 3.7～3.12 に示す。新たに追加する ADDI、JAL、JR 命令の動作について説明する。ADDI は命令フェッチ後、ID でレジスタ Rb の値と符号拡張した即値 d を EX に渡す。EX ではこれらを加算する。MA では何も行わず、WB でレジスタ Rb に演算結果を格納する。

JAL は命令フェッチ後、ID で 0 拡張した即値 d を EX に渡す。EX では分岐先アドレスを d として、IF に分岐するように信号を送る。また、PC+1 を演算結果として後続のステージに渡す。MA では何も行わず、WB でレジスタ 0 に PC+1 を格納する。

JR は命令フェッチ後、ID でレジスタ 0 の値を EX に渡す。EX では分岐先アドレスをこのレジスタ 0 の値として、IF に分岐するように信号を送る。また、0 を演算結果として後続のステージに渡す。MA では何も行わず、WB でレジスタ 0 に 0 を格納する。

	IF	ID
ADD	・メモリ → 命令 ・PC++	・r[Rs] → EX ・r[Rd] → EX ・Rd → 書き込みアドレス
SUB		
AND		
OR		
XOR		
CMP		・r[Rs] → EX ・r[Rd] → EX
MOV		・r[Rs] → EX ・Rd → 書き込みアドレス
SLL		・r[Rd] → EX ・0 拡張した d → EX ・Rd → 書き込みアドレス
SLR		
SRL		
SRA		・Rd → 書き込みアドレス
IN		
OUT		・r[Rs] → EX
HLT		halt

図 3.7 演算/入出力命令 IF～ID

	EX	MA	WB
ADD	<ul style="list-style-type: none"> • $r[Rd] + r[Rs] \rightarrow$ 演算結果 • 条件コード設定 		<ul style="list-style-type: none"> • 演算結果 \rightarrow ID • 書き込みアドレス \rightarrow ID
SUB	<ul style="list-style-type: none"> • $r[Rd] - r[Rs] \rightarrow$ 演算結果 • 条件コード設定 		
AND	<ul style="list-style-type: none"> • $r[Rd] \& r[Rs] \rightarrow$ 演算結果 • 条件コード設定 		
OR	<ul style="list-style-type: none"> • $r[Rd] r[Rs] \rightarrow$ 演算結果 • 条件コード設定 		
XOR	<ul style="list-style-type: none"> • $r[Rd] \wedge r[Rs] \rightarrow$ 演算結果 • 条件コード設定 		
CMP	<ul style="list-style-type: none"> • $r[Rd] - r[Rs]$を計算 • 条件コード設定 		
MOV	<ul style="list-style-type: none"> • 条件コード設定 		<ul style="list-style-type: none"> • $r[Rs] \rightarrow$ ID • 書き込みアドレス \rightarrow ID
SLL	<ul style="list-style-type: none"> • $\text{shift_left_logical}(r[Rd], d)$ \rightarrow 演算結果 • 条件コード設定 		<ul style="list-style-type: none"> • 演算結果 \rightarrow ID • 書き込みアドレス \rightarrow ID
SLR	<ul style="list-style-type: none"> • $\text{shift_left_rotate}(r[Rd], d)$ \rightarrow 演算結果 • 条件コード設定 		
SRL	<ul style="list-style-type: none"> • $\text{shift_right_logical}(r[Rd], d)$ \rightarrow 演算結果 • 条件コード設定 		
SRA	<ul style="list-style-type: none"> • $\text{shift_right_arithmetic}(r[Rd], d)$ \rightarrow 演算結果 • 条件コード設定 		
IN			<ul style="list-style-type: none"> • input \rightarrow ID • 書き込みアドレス \rightarrow ID
OUT	<ul style="list-style-type: none"> • $r[Rs] \rightarrow$ output 		
HLT			

図 3.8 演算/入出力命令 EX～WB

	IF	ID	EX
LD	<ul style="list-style-type: none"> ・メモリ→命令 ・PC++ 	<ul style="list-style-type: none"> ・$r[Rb] \rightarrow EX$ ・符号拡張した $d \rightarrow EX$ ・$Ra \rightarrow$書き込みアドレス 	<ul style="list-style-type: none"> ・$r[Rb] + d \rightarrow$演算結果
ST		<ul style="list-style-type: none"> ・$r[Ra] \rightarrow EX$ ・$r[Rb] \rightarrow EX$ ・符号拡張した $d \rightarrow EX$ 	
LI		<ul style="list-style-type: none"> ・符号拡張した $d \rightarrow EX$ ・$Rb \rightarrow$書き込みアドレス 	
ADDI		<ul style="list-style-type: none"> ・$r[Rb] \rightarrow EX$ ・符号拡張した $d \rightarrow EX$ ・$Rb \rightarrow$書き込みアドレス 	<ul style="list-style-type: none"> ・$r[Rb] + d \rightarrow$演算結果

図 3.9 ロードストア・即値ロード/無条件分岐命令 IF~EX

	MA	WB
LD	<ul style="list-style-type: none"> ・メモリ読み込み 	<ul style="list-style-type: none"> ・読み込みデータ $\rightarrow ID$ ・書き込みアドレス $\rightarrow ID$
ST	<ul style="list-style-type: none"> ・メモリ書き込み 	
LI		<ul style="list-style-type: none"> ・$d \rightarrow ID$ ・書き込みアドレス $\rightarrow ID$
ADDI		<ul style="list-style-type: none"> ・演算結果 $\rightarrow ID$ ・書き込みアドレス $\rightarrow ID$

図 3.10 ロードストア・即値ロード/無条件分岐命令 MA~WB

	IF	ID
BE	<ul style="list-style-type: none"> ・メモリ → 命令 ・PC++ 	<ul style="list-style-type: none"> ・符号拡張した d → EX ・cond → 書き込みアドレス
BLT		
BLE		
BNE		
B		
JAL		<ul style="list-style-type: none"> ・0 拡張した d → EX ・0 → 書き込みアドレス
JR		<ul style="list-style-type: none"> ・r[0] → EX ・0 → 書き込みアドレス

図 3.11 条件分岐命令 IF～ID

	EX	MA	WB
BE	if (Z) $PC + 1 + d \rightarrow \text{分岐先アドレス}$		
BLT	if ($S \wedge V$) $PC + 1 + d \rightarrow \text{分岐先アドレス}$		
BLE	if ($Z \vee S \wedge V$) $PC + 1 + d \rightarrow \text{分岐先アドレス}$		
BNE	if ($\neg Z$) $PC + 1 + d \rightarrow \text{分岐先アドレス}$		
B	$PC + 1 + d \rightarrow \text{分岐先アドレス}$		
JAL	$d \rightarrow \text{分岐先アドレス}$		<ul style="list-style-type: none"> ・PC + 1 → ID ・書き込みアドレス → ID
JR	$r[0] \rightarrow \text{分岐先アドレス}$		<ul style="list-style-type: none"> ・0 → ID ・書き込みアドレス → ID

図 3.12 条件分岐命令 EX～WB

参考文献

- 「計算機科学実験及演習 3 SIMPLE 設計資料」
www.lab3.kuis.kyoto-u.ac.jp/~ktakagi/le3a/simple.pdf