

提出日：2023 年 5 月 12 日

2023 年度 3 回生前期学生実験 HW

方式設計仕様書

23 班

構成員：加藤利梓・神事倫紀・PYII PHYO MAUNG

1. 概要

方式設計仕様書では、アーキテクチャ検討報告書で示したプロセッサを具体的にどのように実現するのかを示す。

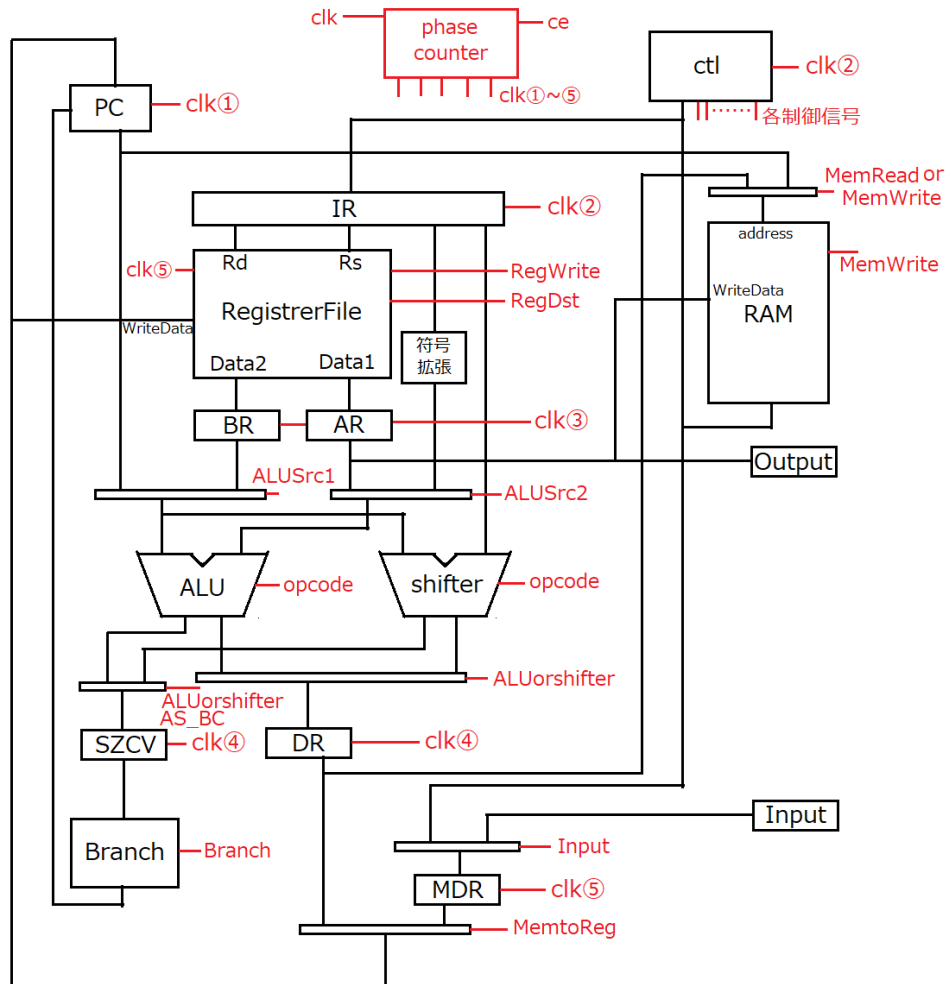


図1 設計するアーキテクチャのブロック図

図1に設計するアーキテクチャのブロック図を示す。図に示すようにプロセッサを複数のコンポーネントに分け、班員が役割分担をして各コンポーネントを設計し、次にそれらを組み合わせて全体のプロセッサを構成するボトムアップ方式で実装する。

さらに以下を行う。

- RAM を命令保持用と主記憶用に分ける。(ハーバードアーキテクチャ)
- パイプライン化のためのフォワーディングユニットとハザード検出ユニットを追加。
- 不成立分岐予測ユニットを追加。

2. 命令セット・アーキテクチャ

今回設計するプロセッサの命令はすべて 16bit の固定長とし、以下の四種類の命令形式がある。基本的には SIMPLE アーキテクチャと同じだが、即値加算命令を追加する。

(1) 演算/入出力命令形式

演算/入出力命令の命令セットを表 1 に示す。

- $I_{15:14}(\text{op1})$ …操作コード(11)
- $I_{13:11}(\text{Rs})$ …ソースレジスタ番号
- $I_{10:8}(\text{Rd})$ …デスティネーションレジスタ番号
- $I_{7:4}(\text{op3})$ …操作コード(0000~1111)
- $I_{3:0}(\text{d})$ …シフト桁数

mnemonic	op3	function
ADD	0000	$r[\text{Rd}] = r[\text{Rd}] + r[\text{Rs}]$
SUB	0001	$r[\text{Rd}] = r[\text{Rd}] - r[\text{Rs}]$
AND	0010	$r[\text{Rd}] = r[\text{Rd}] \& r[\text{Rs}]$
OR	0011	$r[\text{Rd}] = r[\text{Rd}] r[\text{Rs}]$
XOR	0100	$r[\text{Rd}] = r[\text{Rd}] \wedge r[\text{Rs}]$
CMP	0101	$r[\text{Rd}] - r[\text{Rs}]$
MOV	0110	$r[\text{Rd}] = r[\text{Rs}]$
reserved	0111	
SLL	1000	$r[\text{Rd}] = \text{shift_left_logical}(r[\text{Rd}], d)$
SLR	1001	$r[\text{Rd}] = \text{shift_left_rotate}(r[\text{Rd}], d)$
SRL	1010	$r[\text{Rd}] = \text{shift_right_logical}(r[\text{Rd}], d)$
SRA	1011	$r[\text{Rd}] = \text{shift_left_arithmetic}(r[\text{Rd}], d)$
IN	1100	$r[\text{Rd}] = \text{input}$
OUT	1101	$\text{output} = r[\text{Rs}]$
reserved	1110	
HLT	1111	停止

表 1 演算/入出力命令

(2) ロード/ストア命令形式

ロード/ストア命令の命令セットを表 2 に示す。

- $I_{15:14}(\text{op1})$ …操作コード(00/01)
- $I_{13:11}(\text{Ra})$ …ソース/デスティネーションレジスタ番号
- $I_{10:8}(\text{Rb})$ …ベースレジスタ番号
- $I_{7:0}(\text{d})$ …変位

mnemonic	Op1	function
LD	00	$r[\text{Ra}] = *(r[\text{Rb}] + \text{sign_ext}(\text{d}))$
ST	01	$*(r[\text{Rb}] + \text{sign_ext}(\text{d})) = r[\text{Ra}]$

表 2 ロード/ストア命令

(3) 即値ロード/即値加算/無条件分岐命令形式

即値ロード/即値加算/無条件分岐命令の命令セットを表 3 に示す。

- $I_{15:14}(\text{op1})$ …操作コード(10)
- $I_{13:11}(\text{op2})$ …操作コード(000~110)
- $I_{10:8}(\text{Rb})$ …ソース/デスティネーション/ベースレジスタ番号
- $I_{7:0}(\text{d})$ …即値または変位

mnemonic	op2	function
LI	000	$r[\text{Rd}] = \text{sign_ext}(\text{d})$
ADDI	001	$r[\text{Rd}] = r[\text{Rd}] + \text{sign_ext}(\text{d})$
reserved	010	
reserved	011	
B	100	$\text{PC} = \text{PC} + 1 + \text{sign_ext}(\text{d})$
reserved	101	
reserved	110	
条件分岐命令	111	

表 3 即値ロード/即値加算/無条件分岐命令

(4) 条件分岐命令形式

条件分岐命令の命令セットを表 4 に示す。

- $I_{15:14}(\text{op1})$ …操作コード(10)
- $I_{13:11}(\text{op2})$ …操作コード(111)
- $I_{10:8}(\text{cond})$ …分岐条件
- $I_{7:0}(\text{d})$ …変位

mnemonic	op3	function
BE	0000	if (Z) $PC = PC + 1 + \text{sign_ext}(d)$
BLT	0001	if ($S \wedge V$) $PC = PC + 1 + \text{sign_ext}(d)$
BLE	0010	if ($Z \vee (S \wedge V)$) $PC = PC + 1 + \text{sign_ext}(d)$
BNE	0011	if ($\neg Z$) $PC = PC + 1 + \text{sign_ext}(d)$
reserved	0100	
reserved	0101	
reserved	0110	
reserved	0111	

表 4 条件分岐命令

3. 構造

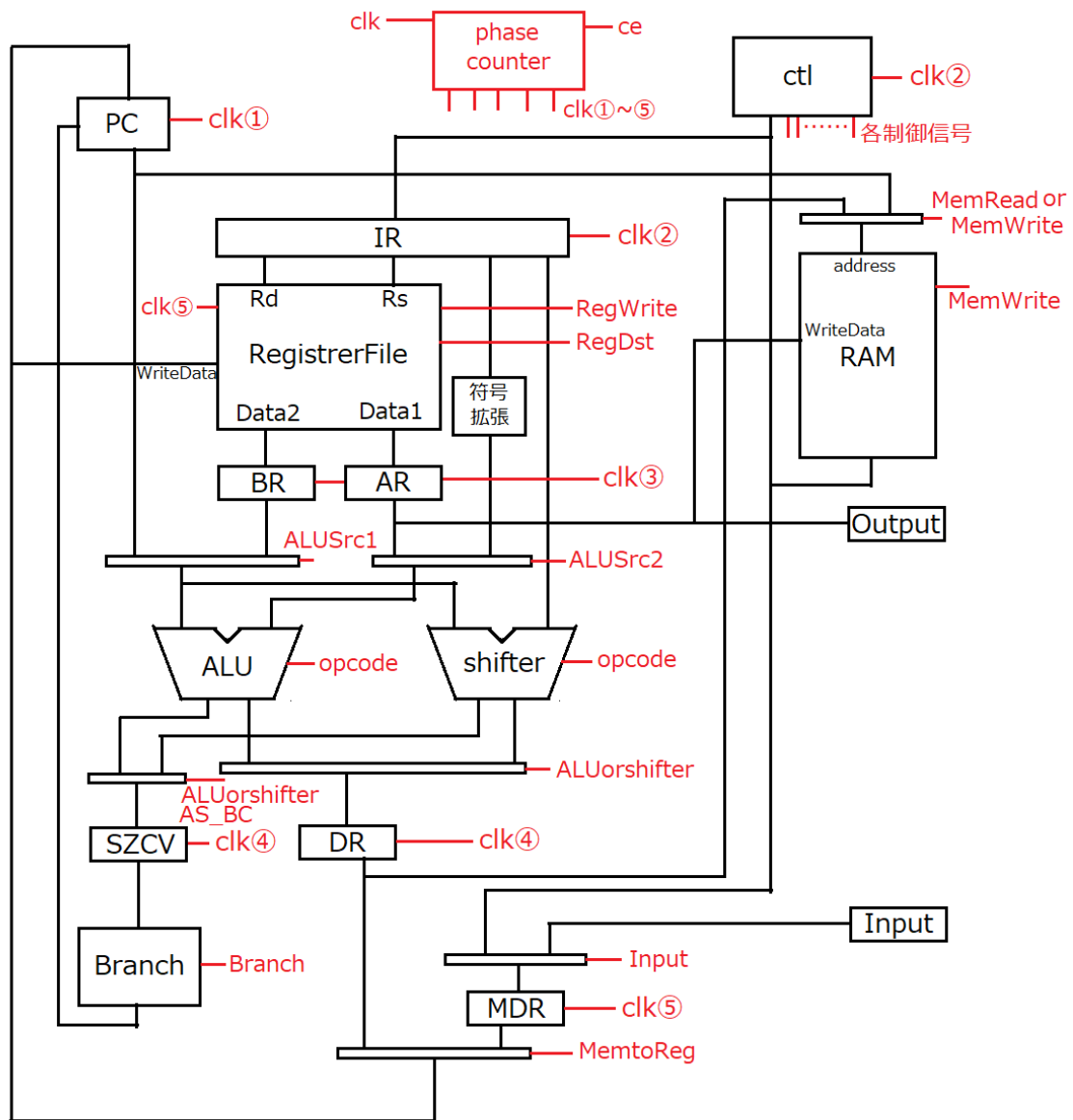


図1 設計するアーキテクチャのブロック図（再掲）

各コンポーネントを wire で接続し、マルチプレクサの部分はマルチプレクサ構文を使って記述する。現時点ではフェイズカウンタから出たフェーズ信号をそのままクロックとしてレジスタに入力しているが、のちにクロックとフェーズ信号を入力とし、フェーズ信号が1の時だけ値を書き換えるように修正する。

各コンポーネントの説明を表5に示す。

コンポーネント	説明
PC	PC の値を計算し、メモリーのアドレス入力にを入力する。内部で+1 した値をあらかじめ計算しておき、分岐命令が来たときは分岐先のアドレスと命令と分岐するかどうかの信号を受けとり、次のアドレスを出力する。
phasecounter	クロックごとにフェーズ信号を出力する。
ctl	命令を受け取り、各制御信号を出力する。
IR,BR,AR,SZCV,DR,MDR	フェーズごとにデータを受け取り、保持し、出力する。
RegisterFile	16bit 8 本のレジスタから構成されている。レジスタの読み出しと書き込みをする。
ALU	四則演算、論理演算、移動演算、比較演算を計算する。
Shifter	シフト演算を計算する。
Branch	条件コードと branch 命令の種類を示す信号受け取り、分岐するかどうかの信号を出力する。
RAM	主記憶メモリー。命令とデータを書き込み、ここから読み出す。

表5 各コンポーネントの説明

4. 動作

4.1. 制御回路

以下の信号が外部から供給される。

1. clock…マスタークロック、このクロックをもとにフェーズ信号を順次活性化する。
2. reset…リセット信号、この信号が1になると内部コンポーネントをリセットする。
3. exec…起動/停止信号、停止状態の時に1になると命令の実行を開始し、実行状態の時に0になるとその時点で実行中の命令を完了してから停止する。

4.2. フェーズ

一つの命令をフェーズ①~⑤を順に活性化することで実行する

フェーズ①：命令のフェッチをし、IR と ctl に格納する。

フェーズ②：ctl から各制御信号を出力する。IR が保持する命令の Ra/Rs フィールドと Rb/Rd フィールドで指定される汎用レジスタの値を読み出し、AR と BR に保持する。

フェーズ③：ALU またはシフターで命令が定める演算を行い、その結果を DR に保持する。また、条件コードを SZCV に保持する。

フェーズ④：ロードストア命令において、DR の値をアドレスとして主記憶にアクセスする。ロード命令の時は値を読み出し MDR に格納する。ストア命令の時は AR が保持する内容を書き込む。また、入出力命令では、入力スイッチの値の MDR への格納や、AR の値をセグメントに出力するなどを行う。

フェーズ⑤：汎用レジスタへの書き込みを伴う命令で、DR または MDR の値を命令の Ra,Rb または Rs フィールドで指定される汎用レジスタに書きこむ。また、分岐命令で DR の値を分岐先アドレスとして PC に書き込む。

4.3. 主記憶と入出力機器

主記憶の容量は SIMPLE のアドレス空間と同じ 64KW とする。この主記憶は FPGA 上に搭載されている RAM にて構成する。

ハーバードアーキテクチャにするために命令用のメモリーとデータ用のメモリーを分ける。

In 命令と Out 命令に対応する入出力インターフェース

In 命令：MU500-RK 上の 8 ビット DIP スイッチ

Out 命令：MU500-RK 上の 7SEG LED の右半分

4.4. パイプライン化

マルチサイクルアーキテクチャを設計した後、五段パイプラインを設計する。データハザードが起これないようにフォワーディングユニット、ハザード検出ユニットを実装する。

4.5. 不成立分岐予測

分岐予測には、不成立分岐予測を実装する。不成立を予測しておき、成立した時はそれまでにフェッチした命令を破棄し、分岐先の命令に飛ぶように回路を実装する。