

## 機能設計仕様書 (Group18)

1029-28-9483 勝田 峻太郎

2018 年 5 月 10 日

## 設計したモジュール

設計したモジュールは以下

- p2
- Controller
- Alu(p3 で使われている)
- main(top level)

## p2

### コンポーネントの外部仕様

#### 概要

このコンポーネントはレジスタを含む. このコンポーネントの基本的な動作としては, 命令を解釈すると同時に, 後続の p3 モジュールが必要とする値をレジスタから読み出す.

#### 入力

このコンポーネントは, 入力として, 以下を受け取る.

**clockp2** ID ステージ (命令デコードとレジスタフェッチ) を行うときのクロック.

**clockp5** WB ステージ (レジスタ書き込み) を行う時のクロック.

**clockp2** 立ち上がり時に読む入力

**command(16bit)** p1 によって読み出された命令を入力とする.

**clockp5** 立ち上がり時に読む入力

**writeflag** レジスタに値を書き込むときは 1, 書き込まないときは 0 を入力.

**writetarget(3bit)** メモリから読んだ値を格納するレジスタの番号.(writeflag==1 のときのみ)

**writeval(16bit)** レジスタに書き込む値.(writeflag==1 のときのみ)

#### 出力

**alu1, alu2(16bit)** 演算命令の場合,ALU が使用する値 2 つをレジスタや即値から取得する.

**opcode(4bit)** ALU で処理をする場合に, 行うべき演算を示している.

Table 1: 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

**writereg(1bit)** 演算または, ロード命令の場合, 結果をレジスタに書き込む必要がある. 書き込む場合は 1, 書き込まない場合は 0 である.

**regaddress(3bit)** writereg が 1 の場合, 書き込む対象となるレジスタの番号を示す.

**memwrite(2bit)** メモリに行う操作をコードで表す. 何もしない場合は 00, 読み込み時は 01, 書き込み時は 10 を示す.

**address(16bit)** メモリに操作をを行う場合, どの番地に行うかを示している.

**storedata(16bit)** メモリに書き込みを行う場合, 書き込む内容を示す.

## 内部仕様

入力として命令とクロックを受け取り, 出力レジスタに対して, クロックの立ち上がりとともに, 対応するデータを書き込む. 書き込む値は, 全て場合分け関数を持ちいて出力される.

## クロック p2 立ち上がり時

- alu に入力するべき 2 つの値 (使用しない場合は未定義) をレジスタまたは即値から取得する.
- alu が必要とする opcode を供給する.
- 最後に結果をレジスタに書き込むかのフラグ, および書き込み先を指定する.
- メモリに対する動作 (書き込み, 読み出し, 何もしない) と書き込む場合は書き込む値を供給する.

## clockp5 立ち上がり時

- 必要に応じてレジスタに書き込みを行う.

## タイミング制約

このモジュールの最大動作可能周波数は, 299.85(Mhz) である. 250(MHz) 動作させたときの最悪 **setup slack** は, 0.665(ns) で, **hold slack** は 0.996(ns) であった.

# Controller.v

## 外部仕様

### 概要

今後パイプラインプロセッサに進化させる可能性のあるもの, 現状では, マルチサイクル方式のプロセッサであるため, 各モジュール (p1~p4) に対して適切なクロックを流す必要がある.

このために, このモジュールはクロックを受け取り, 各モジュールが適切なタイミングで処理をするように, generated clock を出力する.

また, 基板上のボタンによる処理の開始, 停止, 及びリセットを実現する.

### 入力

**clock** 供給されるクロック

**execbutton** テンキーボタンからの入力

### 出力

**clock0** p1 に供給するクロック (IF)

**clock1** p2 に供給するクロック (ID)

**clock2** p3 に供給するクロック (EX)

**clock3** p4 に供給するクロック (MEM)

**clock4** p2 に供給するクロック (WB)

**statusled(8bit)** 7seg led 上に現在のプロセッサの状態 (実行中は'E', 停止中は'S') を出力する.

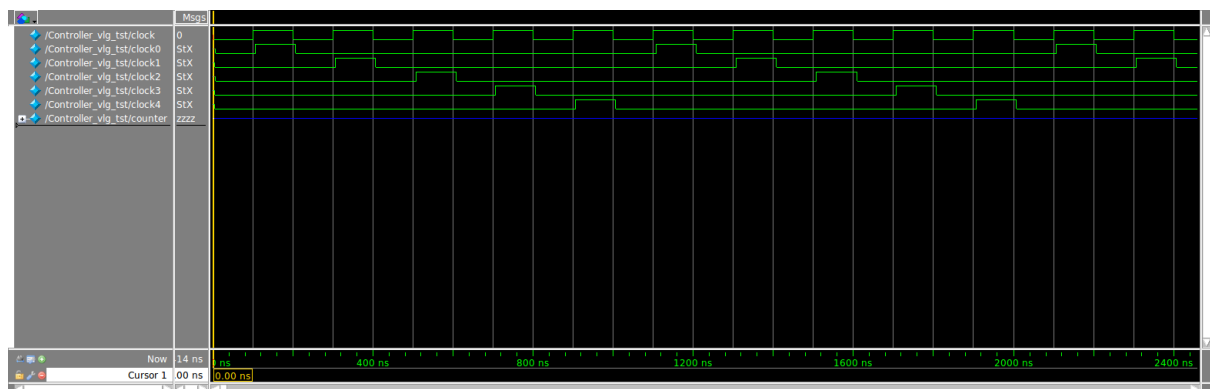


Figure 1: シミュレーション画像

コントローラーの入力と出力は,fig. 1 のようになる.

## Alu

### 外部仕様

#### 入力

**in1, in2(16bit)** 計算する 2 つの数  
**opcode(4bit)** 行う演算を表す数 (tbl. 1 参照)

#### 出力

**result(16bit)** 演算結果  
**v, z, c, s(1bit)** 演算の条件コード (オーバーフロー, ゼロ, キャリー, サイン)

### 内部仕様

opcode による条件分岐により, 異なった計算を行い, 出力する.

## main

### 外部仕様

クロックを入力とするプロセッサ.

### 内部仕様

main モジュールは, ブロック図で記述する.

構成は, 各 p1~p4 とコントローラーから出力されるクロックを適切に配線する.

fig. 2 中に見られる出力の一部は, シミュレーションによる動作確認のためのものである.

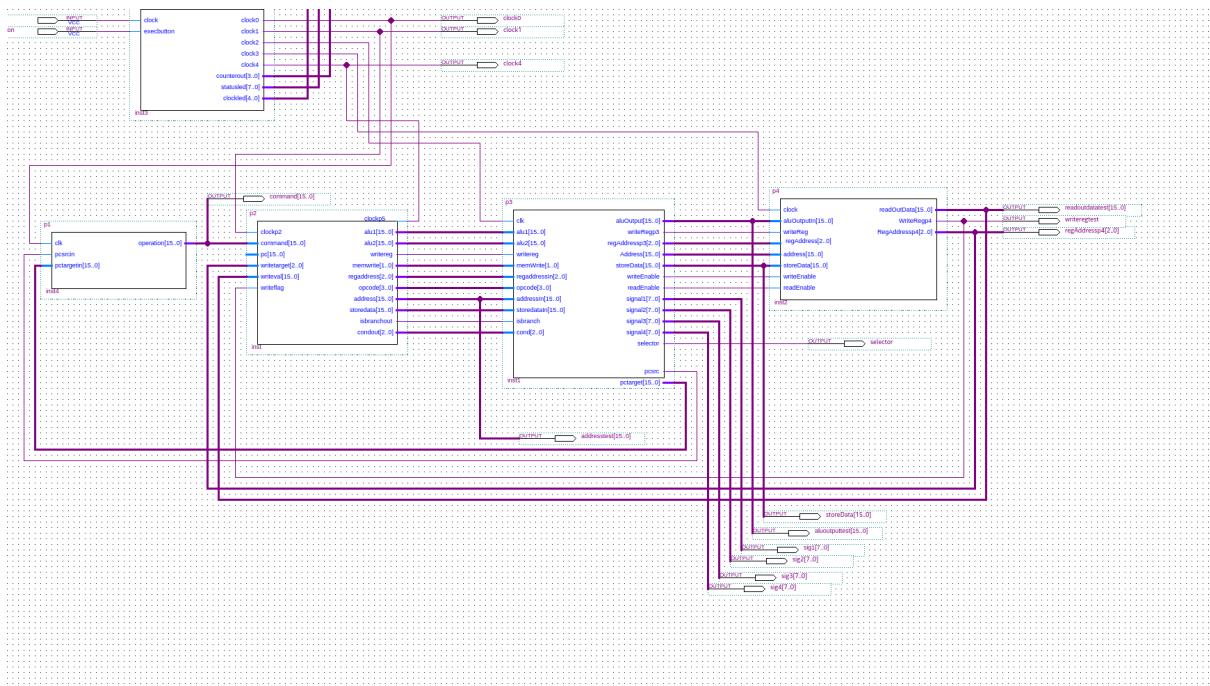


Figure 2: main モジュール