2022 年度 3 回生前期学生実験 HW

team02 方式設計仕様書

方式設計仕様書作成者: 植田健斗
グループメンバー：
伊藤舜一郎 (学籍番号:1029-32-7548)
植田健斗　(学籍番号:1029-32-6498)

提出期限：5 月 12 日 18 時　提出日: 2022 年 5 月 7 日

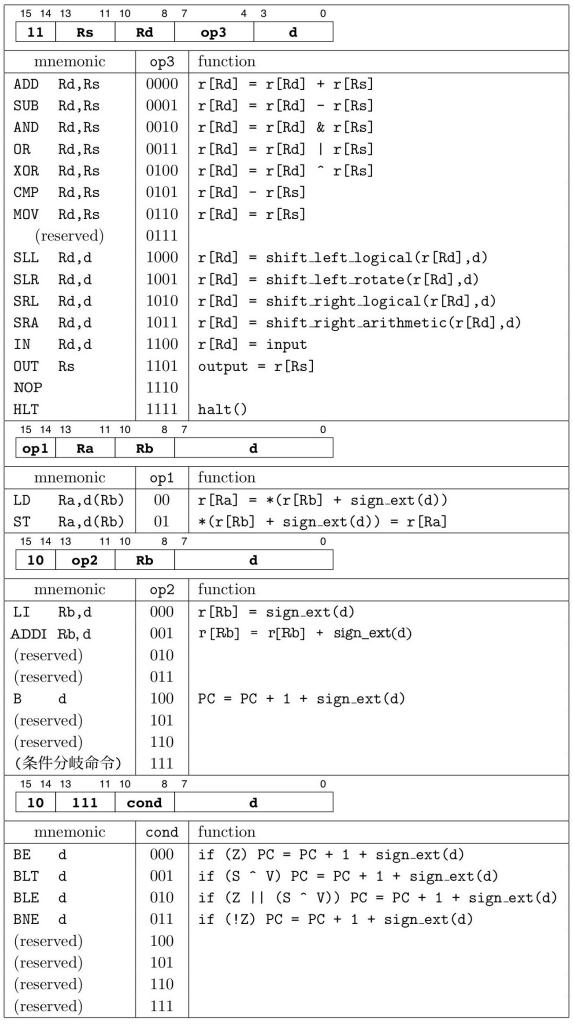
1　概要

授業資料の仕様を満たすプロセッサを作った。このプロセッサをもとにして、今後拡張機能の追加を目標とする。授業資料の仕様に加えて、以下の処理を実現するプロセッサの作成を目標とする。

- 5 段のパイプライン処理の設計をする。
- 即値命令 ADDI を実装する。

2　命令セット・アーキテクチャ

命令セットアーキテクチャは以下の図 1 のようになる。



15 14 13	11 10	8 7	4 3	0
11	Rs	Rd	op3	d

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

15 14 13	11 10	8 7	0
op1	Ra	Rb	d

mnemonic	op1	function
LD Ra,d(Rb)	00	r[Ra] = *(r[Rb] + sign_ext(d))
ST Ra,d(Rb)	01	*(r[Rb] + sign_ext(d)) = r[Ra]

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	
B d	100	PC = PC + 1 + sign_ext(d)
(reserved)	101	
(reserved)	110	
(条件分岐命令)	111	

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)
(reserved)	100	
(reserved)	101	
(reserved)	110	
(reserved)	111	

図 1: 命令セット・アーキテクチャ

2.1 IN 命令

IN 命令が呼ばれると実行を一時的に中断し、外部入力で入力された値をうけとる。中間発表の段階の外部入力では、実行の中断中に 16 個のディップスイッチを変更し、16bit の値を設定し、exec ボタンを押すことで exec ボタンが押された時のディップスイッチを用いて表現された値を入力として受け取り、実行を再開するようにした。

2.2 OUT 命令

OUT 命令が呼ばれると、フェーズが p3 の時に外部出力に Rs フィールドで指定したレジスタの中身が渡される。中間発表の段階の外部出力では 7SEG-LED を 4 つ用いて 16 進数表示で 16bit のデータを表示する。また、過去 16 回の OUT 命令で出力されたデータを保持し、7SEG-LED に表示する。

2.3 HLT 命令

命令の実行を中断するための命令である。内部的には exec ボタンが押されたのと同じ動作をする。HLT 命令が呼ばれた後に、exec ボタンを押すとメモリ上で HLT 命令の次の命令から命令の実行を再開する。

2.4 NOP 命令 (=non operation)

レジスタ・メモリ書き込み、外部入出力などの動作を、何も行わない命令である。機器の初期化の際に、reset ボタンが押されると Instruction Register の値は 2'b 1100000011100000 となり、NOP 命令の値に設定される。

2.5 ADDI 命令

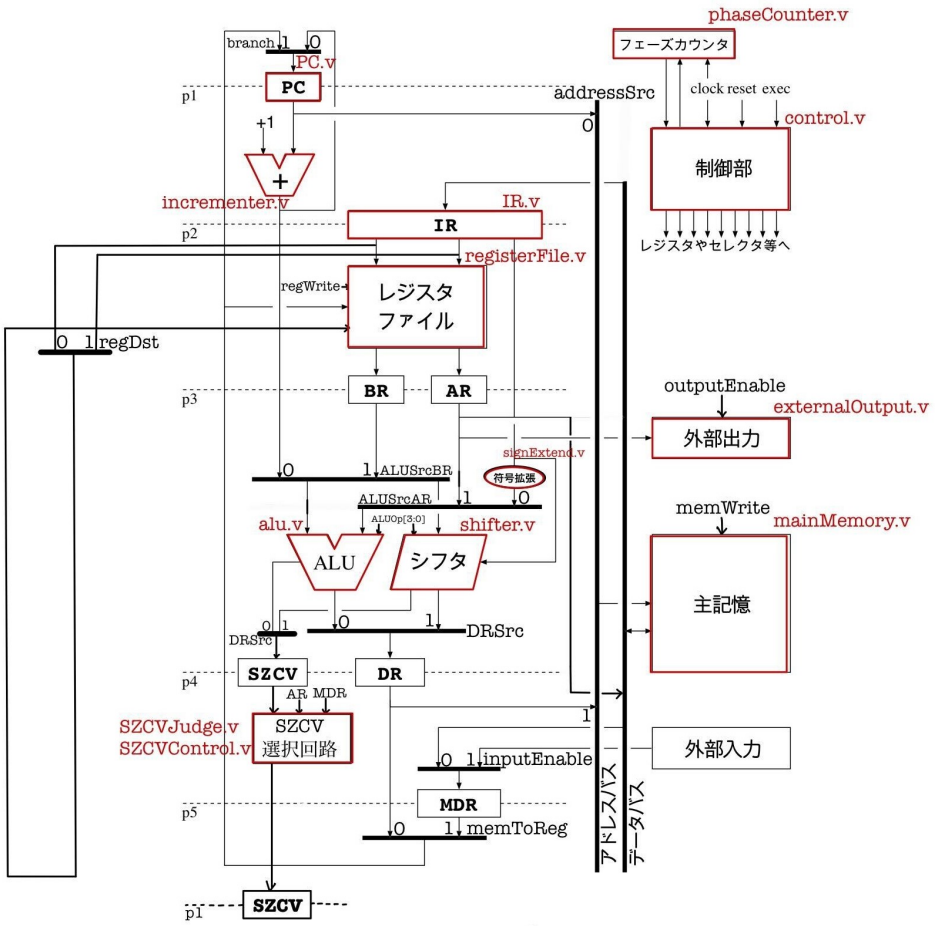
Rb フィールドで指定したレジスタの中身を読み出し、命令の下 8bit で指定した符号付きの値 d を足し合わせる。足し合わせた値を Rb フィールドで指定したレジスタに格納する。中間発表の段階では未実装の機能である。

2.6 その他の命令

ADD,SUB,AND,OR,XOR,CMP,MOV,SLL,SLR,SRL,SRA,LD,ST,LI,B,BE,BLT,BLE,BNE については授業資料の命令セットの仕様通りに設計を行ったため、授業資料の命令セットの説明と被る箇所は割愛する (授業資料のコピーを載せることになるので割愛)。ただし、LD,ST,LI,IN,OUT 命令での SZCV のフラグについて、それぞれの命令で移動するデータの値 (LD ならばメモリからロードするデータ、ST ならばメモリに格納するデータ、LI ならばレジスタに格納する即値の値、IN ならば入力として受け取るデータ、OUT ならば出力するデータ) が、正のとき SZCV=0000、負のとき SZCV=1000、0 のとき SZCV=0100 となるように SZCV フラグを設定した。

3　構造と動作

現状のプロセッサの構造は以下の図 2 のようになる。



各レジスタ:register16.v,register4.v,register3.v
各マルチプレクサ:multiplexer16.v,multiplexer4.v,multiplexer3.v
全体:processor.bdf

図 2: プロセッサの構造 (現状)

このプロセッサは 5 フェーズに処理をわけ、フェーズカウンタによりクロックサイクルごとに p1,p2,p3,p4,p5 というフェーズ信号が順に立ち上がる。それぞれのフェーズ信号は図 2 の破線上のレジスタのイネーブル信号として入力される。レジスタのイネーブル信号とは、クロックの立ち上がりでかつ、レジスタのイネーブル信号が 1 であるときのみ、レジスタの中身が書き変わるような信号である。これにより、各フェーズの書き換えをしたいレジスタのみが順に書き変わっていき、処理が実現する。

このプロセッサは制御部内に SystemRunning という 1bit のレジスタをもっており、そのレジスタの値が 0 のときに SystemRunning レジスタ以外のすべてのレジスタのイネーブル信号が 0 になるようになっている。exec ボタンを押されると、この SystemRunning の反転が行われる。これにより exec ボタンを押すたびに実行停止・再開ができる。reset ボタンを押すと、以下のことがすべて行われる。

- IR に NOP 命令を表す命令コード 2'b 1100000011100000 が格納される。
- PC に 0 が格納される。
- 外部出力の過去の出力履歴がクリアされる。
- レジスタファイルの内部のレジスタに 0 が格納される。
- フェーズカウンタのフェーズ信号を p1 に戻す。
- SystemRunning に 0 が格納される。

これにより reset ボタンを押すことで機器を初期状態にし、実行待機状態にすることができる。この状態で exec ボタンを押すとメモリの 0 番地から順に命令が実行される。FPGA にプログラムをダウンロードした後に exec ボタンを押さなければ、命令が実行されないことに注意が必要である。

以下はまだ実現されていない機能であるが、このプロセッサにパイプライン処理を追加する。p1 から p5 の 5 フェーズを並列に実行する。パイプライン化により、プロセッサの動作は変えずに、処理の高速化を行う。構造については、以下の変更をする。

- メモリを命令メモリとデータメモリに分け、構造ハザードを防ぐ。
- ALU とは別にアドレス計算用の加算器を用意し、アドレス計算と演算の際の ALU での構造ハザードを防ぐ。
- 各フェーズの間にレジスタを用意し、フェーズ間での制御信号やデータの受け渡しを行う。
- データのフォーワーディングをおこなうフォーワーディングユニットを追加する。
- パイプラインストールを防止するハザード検出ユニットを追加する。
- 分岐ハザードに対処する回路を追加する。
- 必要な制御信号の追加や、制御信号の変更を行う。
- フェーズカウンタを削除する。

これらの変更を行い実装をする。パイプライン化の構造の具体的な設計はまだできていないので、最終課題で報告する。