2022 年度 3 回生前期学生実験 HW

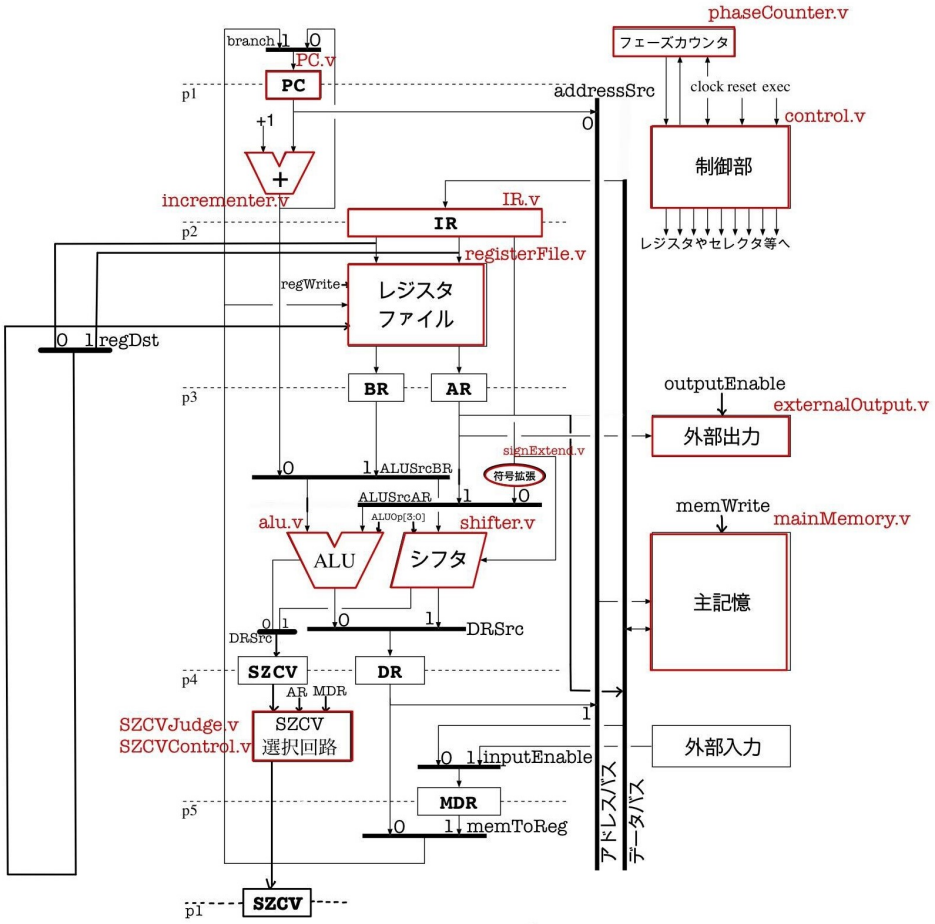
team02 機能設計仕様書 植田健斗担当分

機能設計仕様書作成者: 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 12 日

1　全体のコンポーネント分割方法

それぞれの機能を持った回路ごとにモジュール分割した。各回路につけたファイル名は図 1 に示した通りである。（次の章の図 2 でまとめてある。）



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 1: コンポーネント分割の方法の図

役割分担の表を以下の図 2 に示す。

モジュール名	対応ファイル名	担当
プログラムカウンタ	PC.v	植田
インストラクションレジスタ	IR.v	
レジスタファイル	resisterFile.v	
フェーズカウンタ	phaseCounter.v	
制御部	control.v	
PCをインクリメントする組み合わせ回路	incrementer.v	
主記憶	mainMemory.v	
SZCV選択回路	SZCVJudge16.v,SZCVControl.v	
符号拡張	signExtend.v	
各マルチプレクサー	multiplexer16.v,multiplexer4.v,multiplexer3.v	
各レジスタ	register16.v,register4.v,register3.v	
全体	processser.v	
ALU	alu.v	伊藤
シフタ	shifter.v	
外部出力	externalOutput.v	

図 2: 役割分担（4 月 22 日時点）

2　設計を担当したコンポーネント

設計を担当したコンポーネントは以下である。

- プログラムカウンタ (PC.v)
- インストラクションレジスタ (IR.v)
- レジスタファイル (registerFile.v)
- フェーズカウンタ (phaseCounter.v)
- 制御部 (control.v)
- PC をインクリメントする組み合わせ回路 (incrementer.v)
- 主記憶 (mainMemory.v)
- SZCV 選択回路 (SZCVJudge16.v,SZCVControl.v)
- 符号拡張 (signExtend.v)
- 各マルチプレクサー (multiplexer16.v,multiplexer4.v,multiplexer3.v)
- イネーブル・レジスタ (register16.v,register4.v,register3.v)
- 全体 (processor.bdf)

各コンポーネントごとに外部仕様と内部仕様の説明を行う。

2.1 プログラムカウンタ (PC.v)

2.1.1 外部仕様　入力

この回路の入力は以下の表 1 のようになる。

表 1: プログラムカウンタ (PC.v) の入力 (=input)

input	input の説明
d[15:0]	クロックの立ち上がりにレジスタ書き込み可能ならば、内部のレジスタに書きこむ値
clock	クロック信号。clock の立ち上がりで、レジスタ書きこみが可能な場合、書きこみが行われる。
changeEnable	内部のレジスタが書き換え可能かを表す信号 (イネーブル信号)。この信号が 1 のときに clock が立ち上がると内部のレジスタの値が更新される。
reset	初期化信号。reset が 1 のときにクロックが立ち上がると、レジスタの中身を 0 で初期化する。

2.1.2 外部仕様　出力

この回路の出力は以下の表 2 のようになる。

表 2: プログラムカウンタ (PC.v) の出力 (=output)

output	output の説明
q[15:0]	プログラムカウンタに格納されている値で、命令を読みだすアドレスである。

2.1.3 内部仕様

モジュール register16.v とおなじ仕様である。reset のときに格納する値を変えやすくして保守性を高めるために、register16.v と分けてモジュール化した。

2.2 インストラクションレジスタ (IR.v)

2.2.1 外部仕様　入力

この回路の入力は以下の表 3 のようになる。

表 3: インストラクションレジスタ (IR.v) の入力 (=input)

input	input の説明
d[15:0]	クロックの立ち上がりにレジスタ書き込み可能ならば、内部のレジスタに書きこむ値
clock	クロック信号。clock の立ち上がりで、レジスタ書きこみが可能な場合、書きこみが行われる。
changeEnable	内部のレジスタが書き換え可能かを表す信号 (イネーブル信号)。この信号が 1 のときに clock が立ち上がると内部のレジスタの値が更新される。
reset	初期化信号。reset が 1 のときにクロックが立ち上がると、レジスタの中身を 16'b 1100000011100000 で初期化する。

2.2.2 外部仕様　出力

この回路の出力は以下の表 4 のようになる。

表 4: インストラクションレジスタ (IR.v) の出力 (=output)

output	output の説明
q[15:0]	IR に格納されている値で、主記憶から読みだされた命令である。

2.2.3 内部仕様

モジュール register16.v とほとんどおなじ仕様である。reset のときに格納する値を 16'b 1100000011100000(NOP 命令を表す命令列) にしている点が異なる。

2.3 レジスタファイル (registerFile.v)

2.3.1 外部仕様　入力

この回路の入力は以下の表 5 のようになる。

表 5: レジスタファイル (registerFile.v) の入力 (=input)

input	input の説明
Rs[2:0]	命令中の Rs・Ra フィールドを受け取る。
Rd[2:0]	命令中の Rd・Rb フィールドを受け取る。
regWrite	内部のレジスタが書き換え可能かを表す信号 regWrite と changeEnable がともに 1 のときのみ書き換え可能
writeData[15:0]	レジスタに書き込むデータ
writeRegister[2:0]	writeData[15:0] を書きこむレジスタの番号
clock	クロック信号。clock の立ち上がりで、レジスタ書きこみが可能な場合、書きこみが行われる。
reset	初期化信号。reset が 1 のときにクロックが立ち上がると、レジスタの中身を 0 で初期化する。
changeEnable	内部のレジスタが書き換え可能かを表す信号。regWrite と changeEnable がともに 1 のときのみ書き換え可能

2.3.2 外部仕様　出力

この回路の出力は以下の表 6 のようになる。

表 6: レジスタファイル (registerFile.v) の出力 (=output)

output	output の説明
AR[15:0]	Rs(Ra) フィールドで指定したレジスタに格納されているデータを読み込む。
BR[15:0]	Rd(Rb) フィールドで指定したレジスタに格納されているデータを読み込む。

2.3.3 内部仕様

内部に 8 個の 16bit レジスタ reg [15:0] r [7:0] を持ち、入力 Rs[2:0],Rd[2:0] の値に応じたレジスタに格納されたデータをそれぞれ AR[15:0],BR[15:0] として出力する。また、reset が 0 で changeEnable と regWrite がともに 1 のときのみ書き換え可能状態になり、この状態で clock が立ち上がると、入力 writeData[15:0] で受け取ったデータを writeRegister[2:0] に対応する番号の内部レジスタ r にデータを書きこむ。

reset が 1 のときに clock が立ち上がると、すべての内部レジスタ r の値を 0 で初期化する。

このレジスタファイルはパイプライン化をするにあたって改良の余地がある。あるレジスタへの書き込みとそのレジスタからのデータの読み出しを同じクロック周期の中で行ったときに、中間報告時点で作成したレジスタファイルでは、書き込みを行う前のデータが読みだされてしまう。最終課題までには、あるレジスタへの書き込みとそのレジスタからのデータの読み出しを同じクロック周期の中で行ったときに、同じクロックの中で書き込みするデータを読みだせるように改良をする必要がある。

2.4 フェーズカウンタ (phaseCounter.v)

2.4.1 外部仕様　入力

この回路の入力は以下の表 7 のようになる。

表 7: フェーズカウンタ (phaseCounter.v) の入力 (=input)

input	input の説明
clock	クロック信号。clock 立ち上がりで、レジスタ書きこみが可能な場合、書きこみが行われる。
reset	初期化信号。reset が 1 のときにクロックが立ち上がると、フェーズを p1 に戻す。
changeEnable	内部のレジスタが書き換え可能かを表す信号 (イネーブル信号)。この信号が 1 のときに clock が立ち上がると内部のレジスタの値が更新される。

2.4.2 外部仕様　出力

この回路の出力は以下の表 8 のようになる。

表 8: フェーズカウンタ (phaseCounter.v) の出力 (=output)

output	output の説明
p1	p1 フェーズを表す信号。IR のイネーブル信号に使われる。
p2	p2 フェーズを表す信号。AR,BR のイネーブル信号に使われる。
p3	p3 フェーズを表す信号。外部出力、SZCV,DR のイネーブル信号に使われる。
p3to4	p3 と p4 の間で立ち上がる信号。メモリのイネーブル信号に使われる。
p4	p4 フェーズを表す信号。MDR のイネーブル信号に使われる。
p5	p5 フェーズを表す信号。SZCV 選択回路を通過した後の SZCV, レジスタファイル,PC のイネーブル信号に使われる。

2.4.3 内部仕様

クロックの立ち上がりごとに p1→p2→p3→p4→p5 の順にフェーズ信号が立ち上がる。マスタースレーブ方式の dff を用いてクロックの立ち上がりから半周期ずらして p1 から p5 のフェーズ信号が立ち上がるようにしている。また p3 と p4 の間に p3to4 という信号が立ち上がる。各信号の時間変化は以下の図 3 のようになる。クロックの立下りのタイミングで p1 から p5 の各フェーズ信号が立ち上がっていて、p3top4 だけ clock の立ち上がりのタイミングにに立ち上がっている。各フェーズ信号は 5 クロックごとに立ち上がっている。

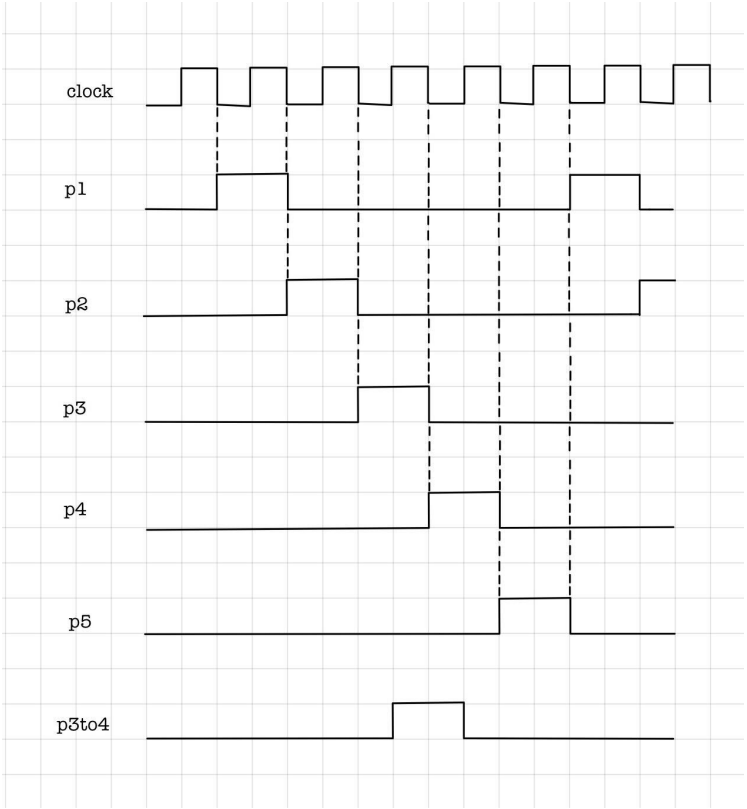


図 3: フェーズカウンタの各信号の時間変化

2.5 制御部 (control.v)

2.5.1 外部仕様　入力

この回路の入力は以下の表 9 のようになる。

表 9: 制御部 (control.v) の入力 (=input)

input	input の説明
clock	クロック信号。clock 立ち上がりで、レジスタ書きこみが可能な場合、書きこみが行われる。
instruction[15:0]	命令の格納された IR の出力を受け取る。
reset	初期化信号。reset が 1 のときにクロックが立ち上がると、出力 systemRunning 信号を 0 にする。
exec	全体の実行・停止信号 exec を受け取る。
p1	フェーズカウンタの出力 p1 を受け取る。
p2	フェーズカウンタの出力 p2 を受け取る。
p3	フェーズカウンタの出力 p3 を受け取る。
p3to4	フェーズカウンタの出力 p3to4 を受け取る。
p4	フェーズカウンタの出力 p4 を受け取る。
p5	フェーズカウンタの出力 p5 を受け取る。
SZCV[3:0]	前回の命令での SZCV フラグを受け取る。

2.5.2 外部仕様　出力

この回路の出力は以下の表 10 のようになる。

表 10: 制御部 (control.v) の出力 (=output)

output	output の説明
addressSrc	メモリに入力するアドレスが、IR の値 (=0) か DR の値 (=1) かを決める。
regDst	書き込みをするレジスタを、入力 Instruction[10:8](Rd,Rb フィールド) で指定する (=0) か、Instruction[13:11](Ra,Rs フィールド) で指定する (=1) かを決める。
ALUSrcAR	ALU の入力 inB[15:0] の受け取る値が、命令中の即値 (=0) か、AR の値 (=1) かを決める。
ALUSrcBR	ALU の入力 inA[15:0] の受け取る値が、((プログラムカウンタの値)+1) にする (=0) か BR の値にするか (=1) かを決める。
ALUOp[3:0]	ALU の入力 op[3:0] に入力する値で、ALU で行う演算を決める。
DRSrc	DR に入力する値を、ALU の出力 out[15:0] にする (=0) か、シフタの出力 out[15:0] にする (=1) かを決める。また、SZCV に入力する値を、ALU の出力 SZCV[3:0] にする (=0) か、シフタの出力 SZCV[3:0] にする (=1) かを決める。
outputEnable	外部出力に値を受け渡す (=1) か受け渡さない (=0) かを決める。正確には OUT 命令のときのみ 1 となる。
inputEnable	外部入力から値を受け取る (=1) か受け取らない (=0) かを決める。正確には IN 命令のときのみ 1 となる。
memWrite	メモリに書きこむ (=1) か書きこまない (=0) かを決める。正確には ST 命令のときのみ 1 となる。
branch	入力の SZCV[3:0] と instruction[15:0] から、分岐をする (=1) か、分岐をしない (=0) かを決める。
regWrite	レジスタファイルに書き込みを行う (=1) か、レジスタファイルに書き込みを行わない (=0) かを決める。
memToReg	プログラムカウンタや、レジスタファイルに渡すデータを、DR の値にする (=0) か MDR の値にする (=1) かを決める。
systemRunning	すべてのレジスタの書き換えを不可能にする (=0) か可能にする (=1) かを決める。つまり、機械が実行待機状態 (=0) か動作状態 (=1) かを決める。

2.5.3 内部仕様

systemRunning という 1bit の内部レジスタを持ち、その値を変えることで機械を実行待機状態 (systemRunning=0) か実行状態 (systemRunning=1) かを切り替えている。exec 入力については 1 クロック前の exec 入力を保持する 1bit レジスタ exec_pre とカウンタを表す 16bit の内部レジスタ counter を持つ。これを用いて以下の表 11 のように exec と exec_pre の値に応じて、counter と systemRunning を変更する。これにより exec ボタンのチャタリングを除去しながら、exec ボタンが押されるたびに機器が実行状態かどうかを決める systemRunning を反転させることができる。

表 11: exec と exec_pre と cnt(=counter) の値に応じて、更新する cnt(=counter) と systemRunning の値

exec_pre exec	00	01	10	11
cnt=0	cnt<=0	cnt<=1	cnt<=1	cnt<=0
0<cnt<15	cnt<=cnt+1	cnt<=0	cnt<=0	cnt<=cnt+1
cnt=15	cnt<=0	cnt<=0	cnt<=0	cnt<=0, sysstemRunning<=∼sysstemRunning

reset が呼ばれると、

```
counter <= 16'h 0000;
systemRunning <= 1'b 0;
exec_pre <= exec;
```

により初期化を行う。reset が押されると systemRunning は 0 になるので、実行が停止し、実行待機状態になる。その後 exec ボタンが押されることで systemRunning が 1 になり実行が行われるようになる。また、インプット命令の p3 と halt 命令の p5 のフェーズで systemRunning を 0 にし実行待機状態になるようにしている。この場合も exec ボタンを押すことで実行を再開できる。

各制御信号は組み合わせ回路を用いて実現している。以下の表 12 でそれぞれの制御信号の出力の仕様をまとめた。

表 12: 制御信号の仕様

制御信号名	制御信号の仕様
addressSrc	=p3to4
regDst	LD 命令の時のみ 1
ALUSrcAR	演算, シフト,IN,OUT,NOP,HALT 命令の時に 1 になる。(MOV の下の reserved の時も 1)
ALUSrcBR	演算、シフト、IN,OUT,NOP,HALT,LD,ST 命令の時に 1 になる。(MOV の下の reserved の時も 1)
ALUOp[3:0]	演算命令 (MOV の下の reserved も含む) のときは instruction[7:4](命令で指定した演算) になる。LI 命令の時は 4'b 0110(MOV 命令) になる。その他のときは 4'b 0000(アドレス計算のための足し算) になる。
DRSrc	シフト,IN,OUT,NOP,HALT 命令の時のみ 1
outputEnable	OUT 命令の時のみ 1
inputEnable	IN 命令の時のみ 1
memWrite	ST 命令の時のみ 1
branch	B 命令または (BE 命令かつ Z=1) または (BLT 命令かつ S^V=1) または (BLE 命令かつ Z|(S^V)=1) または (BNE かつ ~Z=1) のときのみ 1
regWrite	ADD,SUB,AND,OR,XOR,MOV, シフト,IN,LD,LI 命令の時のみ 1
memToReg	LD,IN 命令の時のみ 1

2.6 PC をインクリメントする組み合わせ回路 (incrementer.v)

2.6.1 外部仕様　入力

この回路の入力は以下の表 13 のようになる。

表 13: PC をインクリメントする組み合わせ回路 (incrementer.v) の入力 (=input)

input	input の説明
pc_pre[15:0]	プログラムカウンタの値を受け取る。

2.6.2 外部仕様　出力

この回路の出力は以下の表 14 のようになる。

表 14: PC をインクリメントする組み合わせ回路 (incrementer.v) の出力 (=output)

output	output の説明
pc_post[15:0]	pc_post = pc_pre + 1

2.6.3 内部仕様

プログラムカウンタの値を受け取り加算器を用いてインクリメントした後出力する。pc_post = pc_pre + 1 を実行する。

2.7 主記憶 (mainMemory.v)

2.7.1 外部仕様　入力

この回路の入力は以下の表 15 のようになる。

表 15: 主記憶 (mainMemory.v) の入力 (=input)

input	input の説明
address[11:0]	メモリの読み書きを行う番地のアドレス。
clock	クロック信号を反転させて入力する。clock の立ち上がり (=クロック信号の立ち下り) で、メモリ書きこみが可能な場合書きこみが行われ、また、address[11:0] により指定した番地のデータの読み出しが行われる。
data[15:0]	メモリに書きこむためのデータ
wren	書き込み可能信号。この信号が 1 でかつ clock が立ち上がった時にメモリに書き込みが行われる。

2.7.2 外部仕様　出力

この回路の出力は以下の表 16 のようになる。

表 16: 主記憶 (mainMemory.v) の出力 (=output)

output	output の説明
q[15:0]	メモリから読み出しされたデータ

2.7.3 内部仕様

RAM を用いて実装した。address[11:0] で指定した番地に対して、clock が立ち上がる度にデータの読み出し・書き込みを行う。書き込みは wren 信号が 1 のときにしか行われない。

2.8 SZCV 選択回路 (SZCVJudge16.v)

2.8.1 外部仕様　入力

この回路の入力は以下の表 17 のようになる。

表 17: SZCV 選択回路 (SZCVJudge16.v) の入力 (=input)

input	input の説明
data[15:0]	SZCV フラグを求めたいデータを受け取る。

2.8.2 外部仕様　出力

この回路の出力は以下の表 18 のようになる。

表 18: SZCV 選択回路 (SZCVJudge16.v) の出力 (=output)

output	output の説明
SZCV[3:0]	入力 data[15:0] に応じた SZCV フラグを出力する。

2.8.3 内部仕様

data[15:0] を符号付き 16 進数としたときに data[15:0] が正なら SZCV[3:0]=4'b 0000, data[15:0] が負なら SZCV[3:0]=4'b 1000,data[15:0] が 0 なら SZCV[3:0]=4'b 0100 を出力する。つまり、SZCV[3]=data[15],SZCV[2]=(data[15:0]==16'h 0000),SZCV[1]=0,SZCV[0]=0 となっている。

2.9 SZCV 選択回路 (SZCVControl.v)

2.9.1 外部仕様 入力

この回路の入力は以下の表 19 のようになる。

表 19: SZCV 選択回路 (SZCVControl.v) の入力 (=input)

input	input の説明
instruction[15:0]	命令の格納された IR の出力を受け取る。

2.9.2 外部仕様 出力

この回路の出力は以下の表 20 のようになる。

表 20: SZCV 選択回路 (SZCVControl.v) の出力 (=output)

output	output の説明
SZCVSrc[1:0]	どの SZCV フラグを使うかを決める。AR の SZCV フラグを使う (=2'b 11)、ALU・シフタの SZCV フラグを使う (=2'b 00,2'b 10)、MDR の SZCV フラグを使う (=2'b 01)

2.9.3 内部仕様

SZCVSrc[0] は IN,OUT,ST,LD 命令の時に 1 になり、SZCVSrc[1] は OUT,ST のときに 1 になる組み合わせ回路である。これにより OUT,ST 命令のときは AR の SZCV フラグを、IN,LD 命令の時には MDR の SZCV フラグを、その他の命令のときには ALU・シフタで計算された SZCV フラグを、それぞれ選択できるようになっている。SZCV 選択回路自体は SZCVJudge.v と SZCVControl.v と multiplexter4.v を用いて、全体 (processor.bdf) の中で構成している。(つまり、SZCV 選択回路自体の.v ファイルは存在しない。)

(なぜこのようになっているかと言うと ST 命令のときの SZCV フラグの立て方を授業資料で指定されていなかったので未定義にして設計をしたところ、フィボナッチ数列を求めるテストケースでは ST の SZCV フラグは格納するデータの SZCV フラグでなければいけないことになっていた。これにより急遽追加したのが SZCV 選択回路である。この回路はパイプライン処理で再利用ができないので不要になる。そのため、.v ファイルを作ることをしなかった。つまり、SZCV 選択回路が当初の計画に組み込まれていなかったため、この回路を全体 (processor.bdf) の中で構成をすることになり、全体の回路 (processor.bdf) が見にくくなってしまったが、ご容赦ください。)

2.10　符号拡張 (signExtend.v)

2.10.1　外部仕様　入力

この回路の入力は以下の表 21 のようになる。

表 21: 符号拡張 (signExtend.v) の入力 (=input)

input	input の説明
in[7:0]	符号拡張する 8bit の数字を受け取る。

2.10.2　外部仕様　出力

この回路の出力は以下の表 22 のようになる。

表 22: 符号拡張 (signExtend.v) の出力 (=output)

output	output の説明
out[15:0]	符号拡張後の 16bit の数字を出力

2.10.3　内部仕様

`out[15:0] = {{8{in[7]}},in[7:0]}`により、符号拡張を行う。

2.11 各マルチプレクサー (multiplexer16.v,multiplexer4.v,multiplexer3.v)

2.11.1 外部仕様 入力

この回路の入力は以下の表 23 のようになる。

表 23: 各マルチプレクサー (multiplexer16.v,multiplexer4.v,multiplexer3.v) の入力 (=input)

input	input の説明
in0[15:0 or 3:0 or 2:0]	セレクタ信号が 0 の時に出力されるデータ
in1[15:0 or 3:0 or 2:0]	セレクタ信号が 1 の時に出力されるデータ
selector	1bit のセレクタ信号

2.11.2 外部仕様 出力

この回路の出力は以下の表 24 のようになる。

表 24: 各マルチプレクサー (multiplexer16.v,multiplexer4.v,multiplexer3.v) の出力 (=output)

output	output の説明
out[15:0]	in0 または in1 のうち、セレクタ信号により選ばれた方のデータの値になる。

2.11.3 内部仕様

`out[15:0]= in0 & {16{~selector}} | in1 & {16{selector}}`により実現している。

2.12 各レジスタ (register16.v,register4.v,register3.v)

2.12.1 外部仕様 入力

この回路の入力は以下の表 25 のようになる。

表 25: 各レジスタ (register16.v,register4.v,register3.v) の入力 (=input)

input	input の説明
d[15:0 or 3:0 or 2:0]	クロックの立ち上がりにレジスタ書き込み可能ならば、内部のレジスタに書きこむ値
changeEnable	内部のレジスタが書き換え可能かを表す信号 (イネーブル信号)。この信号が 1 のときに clock が立ち上がると内部のレジスタの値が更新される。
reset	初期化信号。reset が 1 のときにクロックが立ち上がると、レジスタの中身を 0 で初期化する。
clock	クロック信号。clock 立ち上がりで、レジスタ書きこみが可能な場合、書きこみが行われる。

2.12.2 外部仕様 出力

この回路の出力は以下の表 26 のようになる。

表 26: 各レジスタ (register16.v,register4.v,register3.v) の出力 (=output)

output	output の説明
q[15:0]	このレジスタに格納されている値である。

2.12.3 内部仕様

レジスタを表している。ファイル名 register*.v の*の所に入る数字 (16,4,3 のいずれか) が、そのレジスタが何 bit のレジスタであるかを表している。clock の立ち上がりごとに以下のことを行う。reset 信号が 1 のときレジスタを 0 で初期化する。reset 信号が 0 のとき、changeEnable が 1 なら

```
q[15:0]<=d[15:0]
```

によりレジスタの値の更新を行う。それ以外ならば値の変更は何もしない。

2.13 全体 (processor.bdf)

2.13.1 外部仕様　入力

この回路の入力は以下の表 27 のようになる。

表 27: 全体 (processor.bdf) の入力 (=input)

input	input の説明
clock	クロック信号の入力を受け取る。
reset	リセットボタンの入力を受け取り、機械を初期状態にし、実行待機状態にする。
exec	exec ボタンの入力を受け取る。機械を実行停止・再開する。reset ボタンを押した後 exec ボタンを押すことで実行が開始される。
externalInput[15:0]	IN 命令のときに入力されるデータ

2.13.2 外部仕様　出力

この回路の出力は以下の表 28 のようになる。

表 28: 全体 (processor.bdf) の出力 (=output)

output	output の説明
SEG_A[7:0]	7SEG-LED に表示する数字の一つ
SEG_B[7:0]	7SEG-LED に表示する数字の一つ
SEG_C[7:0]	7SEG-LED に表示する数字の一つ
SEG_D[7:0]	7SEG-LED に表示する数字の一つ
SEG_E[7:0]	7SEG-LED に表示する数字の一つ
SEG_F[7:0]	7SEG-LED に表示する数字の一つ
SEG_G[7:0]	7SEG-LED に表示する数字の一つ
SEG_H[7:0]	7SEG-LED に表示する数字の一つ
select[7:0]	7SEG-LED の表示を行うためのセレクタ信号

2.13.3 内部仕様

全体の配線は図 1 のように行った。コントロールからの制御信号のうち、addressSrc,regDst,ALUSrcAR,ALUSrcBR, DRSrc,inputEnable,branch,memToReg については図 1 のマルチプレクサ (図では太線がマルチプレクサを表す) の制御信号として入力されている。

ALUOp[3:0] は ALU とシフタのオペコードを受け取る入力 op[3:0] につないだ。

outputEnable,memWrite,regWrite,systemRunning についてはそれらを用いてそれぞれのレジスタやメモリのイネーブル信号を構成し入力した。IR の入力 changeEnable には p1&systemRunning を、AR,BR の入力 changeEnable には p2&systemRunning を、ALU とシフタから出力された SZCV を格納するレジスタと DR と externalOutput の入力 changeEnable には p3&systemRunning を、MDR の入力 changeEnable には、p4&systemRunning を、PC とレジスタファイルと選択後の SZCV を格納するレジスタの入力 changeEnable には p5&systemRunning を、主記憶の入力 wren には memWrite&p3to4&systemRunning を、フェーズカウンタには systemRunning を、それぞれ入力した。

入力 clock はすべてのモジュールの clock 入力につなぎ、さらに clock を反転させた信号をメモリの入力 clock につないだ。

入力 reset はメタステーブルを防止するため dff を 2 つ直列につないだ後、負論理を正論理にするため反転させ、PC, レジスタファイル,IR, フェーズカウンタ, 制御部,externalOutput の入力 reset につないだ。

入力 exec はメタステーブルを防止するため dff を 2 つ直列につないだ後、負論理を正論理にするため反転させ、制御部につないだ。

3 実験環境

実験で使用したボードや CAD ツールを以下に記す。

- ボード
  Rapid Prototyping Kit PowerMedusa
  MU500-RXSET01(MU500-RX, MU500-RK, MU500-7SEG)
- CAD ツール
  Quartus Prime Version 17.1.0 Build 590 10/25/2017 SJ Lite Edition