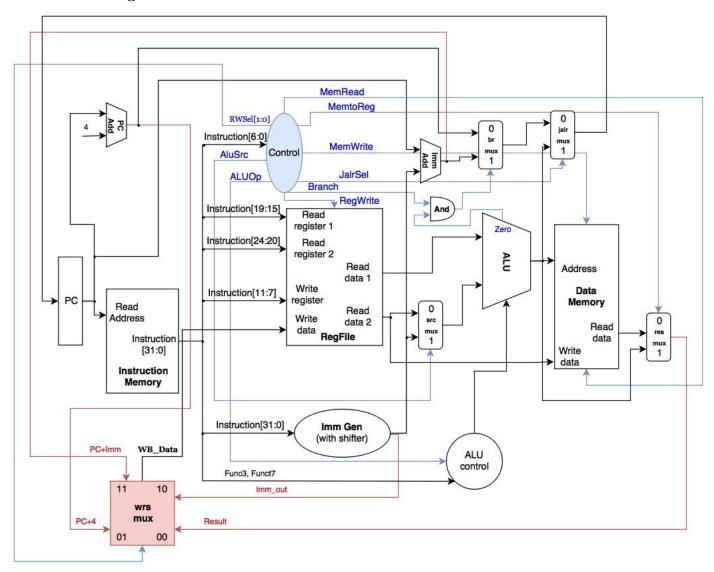
EECS 112L Lab2 Report

Yifan Xu March 24, 2018

1 Block Analysis and Diagram

1.1 Block Diagram



1.2 Imm Gen

Imm_Gen.sv is now a combination of Shifter and Imm Generator. The output is 32 bits. For different type of instruction, it will generate a after-shift final value for directly operation and computation.

1.3 Controller

Three new output signals are generated as select controllers:

Branch 1 bit signal, connected to an AND gate with Zero Signal from ALU, result is the select of Branch Mux(brsmux, 2to1)

0: branch is not taken

1: branch is taken

JalrSel 1 bit signal, generated as select of Jalr Mux(jalrmux, 2to1)

0: jalr is not taken

1: jalr is taken

RWSel 2 bits signal, generated as select of Write Register Mux(wrmux,4to1)

00: Write Back Data

01: PC+4 (JAL/JALR)

10: Imm-gen (LUI)

11: Pc+Imm-gen (AUIPC)

1.4 ALU

The ALU block support 14 operations:

Opcode	Operation
0000	AND
0001	OR
0010	ADD
0011	XOR
0100	<<
0101	>>
0110	SUB
0111	>>>
1000	=
1001	≠
1010	Return True
1011	
1100	<
1101	>
1110	unsigned <
1111	unsigned >

Note, contrary to some other processors, equal and not equal here are doing comparison directly instead of calling subtract. As a result, the signal <u>AluResult</u> of all conditional operations is 32'b000000001 if condition is true, or 32'b000000000 if condition is false.

New output signal Zero is generated as select control for B-type and Jal/Jalr instructions. Zero signal is assigned as the lowest bit of $\underline{\text{AluResult}}$. Therefore, whenever a B-type condition is true or Jal/Jalr is taken, $\underline{\text{Zero}}$ will be 1.

1.5 Branch Mux and Jalr Mux

Brach Mux is a 2 to 1 mux that in charge of selecting the next PC value. Two inputs are $\underline{PC+4}$ from PC Adder and $\underline{PC+Imm}$ from Imm Adder. If Branch is taken, mux select will be 1, $\underline{PC+Imm}$ will be the output. If Branch is not taken, $\underline{PC+4}$ will be the output.

Jalr Mux is a 2 to 1 mux works specifically when Jalr is taken. Under this circumstance, the next PC value should be R[rs1]+Imm, which is calculated and output by ALU in advance.

1.6 Data Memory

The new Data Memory can handle Store and Load Instruction of different size. A new input <u>Funct3</u> is input into Data Memory to help decide how many bits that Store or Load should operate on.

For Load, there are LB, which load the most significant 8 bits of a memory block into a register. While LH load the last 16 bits and LW load all 32 bits. Since register is 32 bits, there will be a sign extension made on the final value depending on the least significant bit of the right side.

There are also LBU and LHU, which will treat data as unsigned value. This means the data memory will only generate 0 to fill those empty digits on the right side.

For Store, there are SB, which store the most significant 8 bits from current register into the target memory block. While SH store the last 16 bits and SW store all 32 bits. **Note, Store operation will only modify specific bits while left others untouched. There is no bits filling operation or sign extension.** For example, SB will only changed [7:0] bits of the target memory block. If memory is never initialized, [32:8] bits will all be x(unknown status)

1.7 Write Register Mux

A new Write Register Mux(wrmux) will in charge of generating the Write-Back Data to a specific Register, one of the input signal of RegFile. In the block diagram before, this part is marked in red.

Normally, the write-back-data from Result Mux(resmux) will always be selected. However, when JAL/JALR is taken, signal $\underline{PC+4}$ will be selected and stored into corresponding register. When LUI is taken, $\underline{Imm-Gen}$ will be selected while $\underline{PC+Imm}$ will be stored only when AUIPC is taken.

2 Waveform

A the bouter of																
/tb_top/reset	'												-			
/tb_top/tb_clk	/ 221 22			221 22				221 22		7 221 22		V 221 22		221 224		
	32'h000					1						1	510293		_	
	32'h000	00000		32'h000	000001	32'h000	300002	32'h000	300004	32'h000	000005	, 32'h000	000007	32'h000	000008	
<pre>/tb_top/reset</pre>																
<pre>/tb_top/tb_clk</pre>																
													32'h000		32'hfc7	4fee3
→ /tb_top/tb_WB_Data → /tb_top/tb_WB_Data	32'h000	0000b	32'h000	000003	32'hfffff	ffe	32'h000	00000	32'h000	000005	32'h000	00001	32'h000	00001		
<pre>/tb_top/reset</pre>																
/tb_top/tb_clk																
													32'h004			
🛨 🔷 /tb_top/tb_WB_Data	32'h000	00002	32'h000	00001	32'h000	00003	32'h000	00001	32'h000	00004	32'h000	00001	32'h000	00005	32'h000	00001
/tb_top/reset																
/tb_top/tb_clk																
_ ⁴ /tb_top/instr_men	32'h008	3e6b3	32'h018	005ef	32'h007	/2c7b3	32'h002	35833	32'h403	34d8b3	32'h000	586e7	32'h02a	02823	32'h168	02023
🛨 🥠 /tb_top/tb_WB_Data	32'h000	0000f	32'h000	00064	32'h000	0000с	32'h000	00002	32'hfffff	fff	32'h000	00088	32'h000	00030	32'h000	00160
<pre>/tb_top/reset</pre>																
/tb_top/tb_clk																
	32'h030	02603	32'h003	11733	32'h00d	50a63	32'h016	14513	32'h02e	2e593	32'h06f	37613	32'h003	49693	32'h003	35713
→ /tb_top/tb_WB_Data	32'h000	00000	32'h000	00020	32'h000	00001	32'h000	00014	32'h000	0002f	32'h000	80000	32'hfffff	ff0	32'h000	00001
√/tb top/reset		[ĺ	ĺ	ĺ	ĺ	ĺ			ĺ	ĺ					
/tb top/tb clk																
	32'h402	6d793	32'h00a	8a833	32'h00a	8b833	32'h002	8a813	32'h002	8b813	32'hccc	cc837	32'hccc	cc817	32'h009	02a23
# 🍎 /tb_top/tb_WB_Data													32'hccc			
√ /tb top/reset			1)		ì	ì	1		ì	Ì					
/tb top/tb clk																
	32'h014	02103	32'h014	00183	32'h014	01203	32'h014	04283	32'h014	05303	32'h004	459693	32'h02d	00423	32'h028	02703
	32'hfffff												32'h000			
/tb top/reset																
/tb top/tb clk																
	32'h02c	01423	32'h029	302703												
						x0XxX										
- / /to_top/to_***B_Date	32 11000	700026	JZIIXXX	X0210	32 11888	XUXX										

3 Synthesis Report

3.1 Critical Path

Critical Path Length 5.31

Critical Path Slack -3.34

Levels of Logic: 37.00 Critical Path Length: 5.31 Critical Path Slack: -3.34 Critical Path Clk Period: 4.00 Total Negative Slack: -53820.01 No. of Violating Paths: 17417.00 Worst Hold Violation: 0.00 Total Hold Violation: 0.00 No. of Hold Violations: 0.00

3.2 Area

Combinational Area: 93501.357157 Noncombinational Area: 115258.882369

Buf/Inv Area: 6168.074989 Total Buffer Area: 3058.11 Total Inverter Area: 3109.96 Macro/Black Box Area: 0.000000

Net Area: 155176.720782

Cell Area: 208760.239526 Design Area: 363936.960308

3.3 Power

	- Internal Power — —— 27.879mW ——	O			
	Switch Power	Int Power	Leak Power	Total Power	%
riscv	137.750	2.79e+04	1.74e+11	2.02e+05	100.0
dp (Datapath)	136.488	2.79e+04	1.74e+11	2.02e+05	100.0
data_mem (datamemory)	68.062	2.61e+04	1.60e+11	1.86e+05	92.0
rf (RegFile)	30.839	1.65e+03	1.10e+10	1.27e+04	6.3
instr_mem (instructionmemo	ry) 5.631	7.042	2.59e+08	272.088	0.1
_seo_mux_C8 (_seo_mux_C8	5.471	6.987	2.55e+08	267.232	0.1