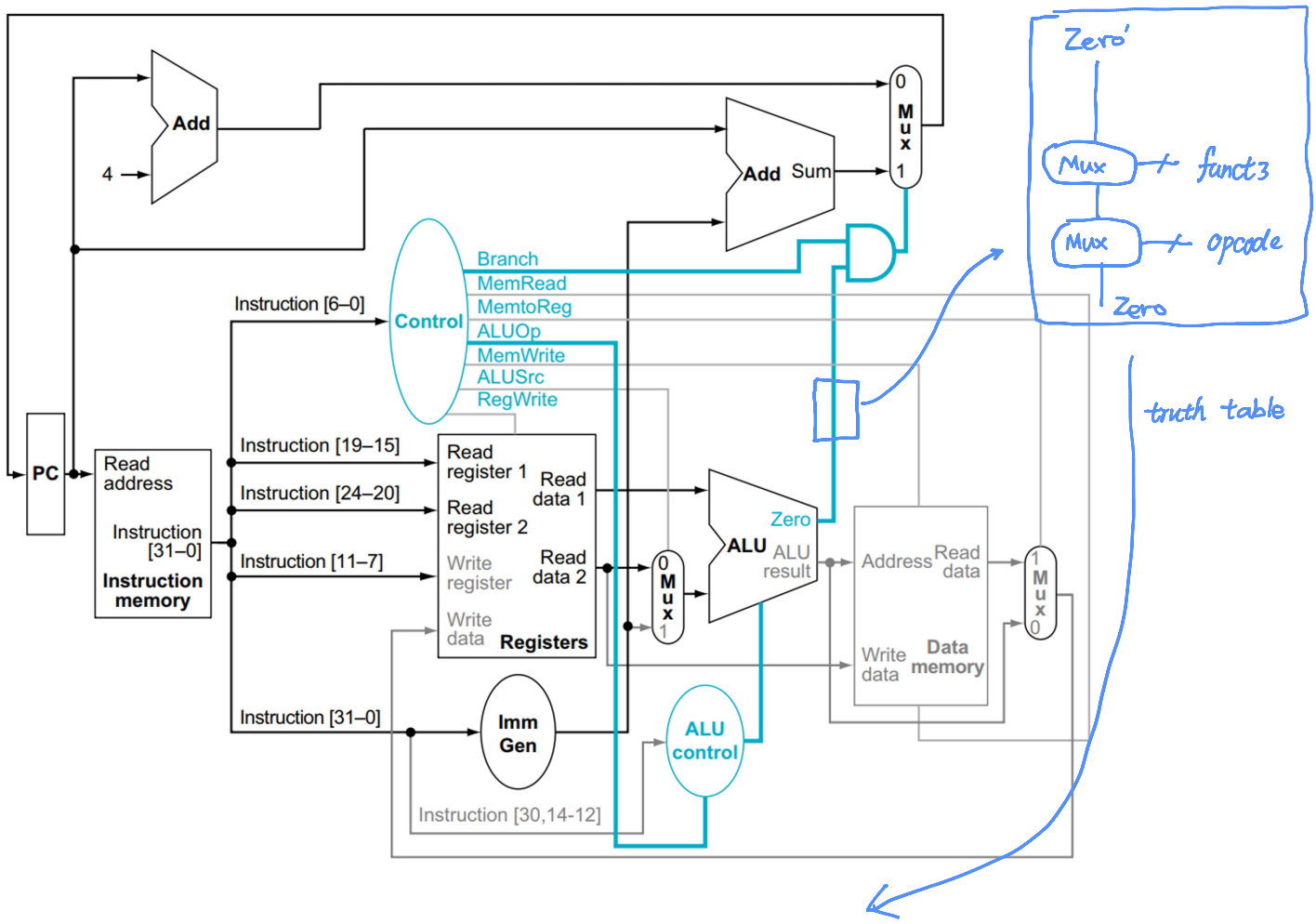
Project phase 1

2024年3月26日 19:14



Schematics

The schematics of my single stage simulator generally follows the design of the image above (taken from Computer Organization and Design RISC-V Edition 2nd ED 2021) with one exception. The simulator changes Zero value depends on the instruction type (BEQ, BNE, and JAL). Here are the truth tables are shown below:

- Zero value truth table

Opcode	funct3	Initial Zero	Final Zero	Remark
1100011	000	0	0	BEQ, Zero unchanged
1100011	000	1	1	BEQ, Zero unchanged
1100011	001	0	1	BNE, Zero flipped
1100011	001	1	0	BNE, Zero flipped
1101111	N/A	0	1	JAL, Zero set to 1
1101111	N/A	1	1	JAL, Zero remains 1

- Control truth table

Opcode	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch	ALUOp1	ALUOp0	Instruction Type
0110011	0	0	1	0	0	0	1	0	R-format (add, sub, xor, or, and)
0010011	1	0	1	0	0	0	1	0	I-type (addi, xori, ori, andi)
0000011	1	1	1	1	0	0	0	0	Load (lw)
0100011	1	X	0	0	1	0	0	0	Store (sw)
1100011	0	X	0	0	0	1	0	1	B-type (beq, bne)
1101111	X	0	1	0	0	0	X	X	J-type (jal)
1111111	X	X	X	X	X	X	X	X	Halt

- ALU control truth table
  - ALUOp1 and ALUOp2 are not used in this simulator because there are 5 possible operations: ADD SUB XOR OR and undefined. 2 bit cannot represent 5 possible operations.

Opcode	funct3	funct7	ALU Operation	ALU Control Signal
1100011	N/A	N/A	SUB (B-type for BEQ, BNE)	0110
0100011	N/A	N/A	ADD (S-type for SW)	0010
0110011	000	0100000	SUB (R-type)	0110
0110011	000	0000000	ADD (R-type)	0010
0110011	100	N/A	XOR (R-type)	0111
0110011	110	N/A	OR (R-type)	0001
0110011	111	N/A	AND (R-type)	0000
0010011	000	N/A	ADD (I-type)	0010
0010011	100	N/A	XOR (I-type)	0111
0010011	110	N/A	OR (I-type)	0001
0010011	111	N/A	AND (I-type)	0000
0000011	000	N/A	ADD (for LW)	0010
_ (default)	_	_	Undefined	1111

- ALU truth table

aluControl	Operation	Result	Zero Flag
0010	ADD	readData1 + readData2	"1" if result is 0, otherwise "0"
0110	SUB	readData1 - readData2	"1" if result is 0, otherwise "0"
0000	AND	readData1 & readData2	"1" if result is 0, otherwise "0"
0001	OR	readData1 | readData2	"1" if result is 0, otherwise "0"
0111	XOR	readData1 ^ readData2	"1" if result is 0, otherwise "0"

Possible Improvement

- The most effective improvement is to change the single cycle behavior to a pipeline one. This improvement can utilize all non-active components, but should be aware of data hazard.