Ctrl Signals / Instruction	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite	Zero	ImmGen Output
add	0	0	0	10	0	0	1	0	X
beq	1	0	X	01	0	0	0	1	32'h 0...0100
lw	0	1	1	00	0	1	1	0	32'h 0

2. (10 points) Given following assembly instruction:
   `sw rs2, imm12(rs1)`

   (1) Which resources (blocks) perform a useful function for this instruction? (3 points)
   (2) Which resources (blocks) produce no output for this instruction? Which resources produce output that is not used? (7 points)

3. (10 points) Consider the following instruction mix:

R-type	I-type (non-lw)	Load	Store	Branch	Jump
20%	28%	25%	10%	15%	2%

   (1) What fraction of all instructions use data memory? (3 points)
   (2) What fraction of all instructions use instruction memory? (2 points)
   (3) What fraction of all instructions use the sign extend? (5 points)

4. (10 points) When silicon chips are fabricated, defects in materials (e.g., silicon) and manufacturing errors can result in defective circuits. A very common defect is for one signal wire to get "broken" and always read a logical 0. This is often called a "stuck-at-0" fault.
   (1) Which instructions fail to operate correctly if the MemToReg wire is stuck at 0? (5 points)
   (2) Which instructions fail to operate correctly if the ALUSrc wire is stuck at 0? (5 points)

5. (30 points) Problems in this exercise assume that the logic blocks used to implement a processor's datapath have the following latencies:

I-Mem / D-Mem	Register File	Mux	ALU	Adder	Single gate	Register Read	Register Setup	Sign extend	Control
250 ps	150 ps	25 ps	200 ps	150 ps	5 ps	30 ps	20 ps	30 ps	100 ps

   In above table, "Register Read" is the time needed after the rising clock edge for the new register value to appear on the output. This value applies to the PC only. "Register Setup" is the amount of time a register's data input must be stable before the rising edge of the clock. This value applies to both the PC and Register File.

1. above in table 1 姜子木 5203710037

2(1) PC, Instruction memory, Registers., ImmGen, ALU, Data memory, PC+4 adder, mux(branch), ALU control, Control, mux(ALU_in)

(2) no output: all blocks have some outputs

not used: Add-sum, zero of ALU, mux of MemToReg

3. (1) 35% (2) 100% (3) 80%

4 (1) Load instruction

(2) I-type(non-Load) Load, store, branch and jump

5. (1) L = PC + I-Mem + Register File + Mux + ALU + Mux + RS

= 30 + 250 + 150 + 25 + 200 + 25 + 20

= 700 ps

(2) L = PC + I-Mem + Register File + Mux + ALU + D-Mem + Mux + RS

= 30 + 250 + 150 + 25 + 200 + 250 + 25 + 20

= 950 ps

(3) L = PC + I-Mem + Register File + Mux + ALU + D-Mem

= 30 + 250 + 150 + 25 + 200 + 250

= 905 ps

(4) L = PC + I-Mem + Register File + Mux + ALU + Gate + Mux + RS

= 30 + 250 + 150 + 25 + 200 + 5 + 25 + 20

= 705 ps

(5) L = PC + I-Mem + Register File + Mux + ALU + Mux + RS

= 30 + 250 + 150 + 25 + 200 + 25 + 20

= 700 ps

(6) L = 950 ps

6.