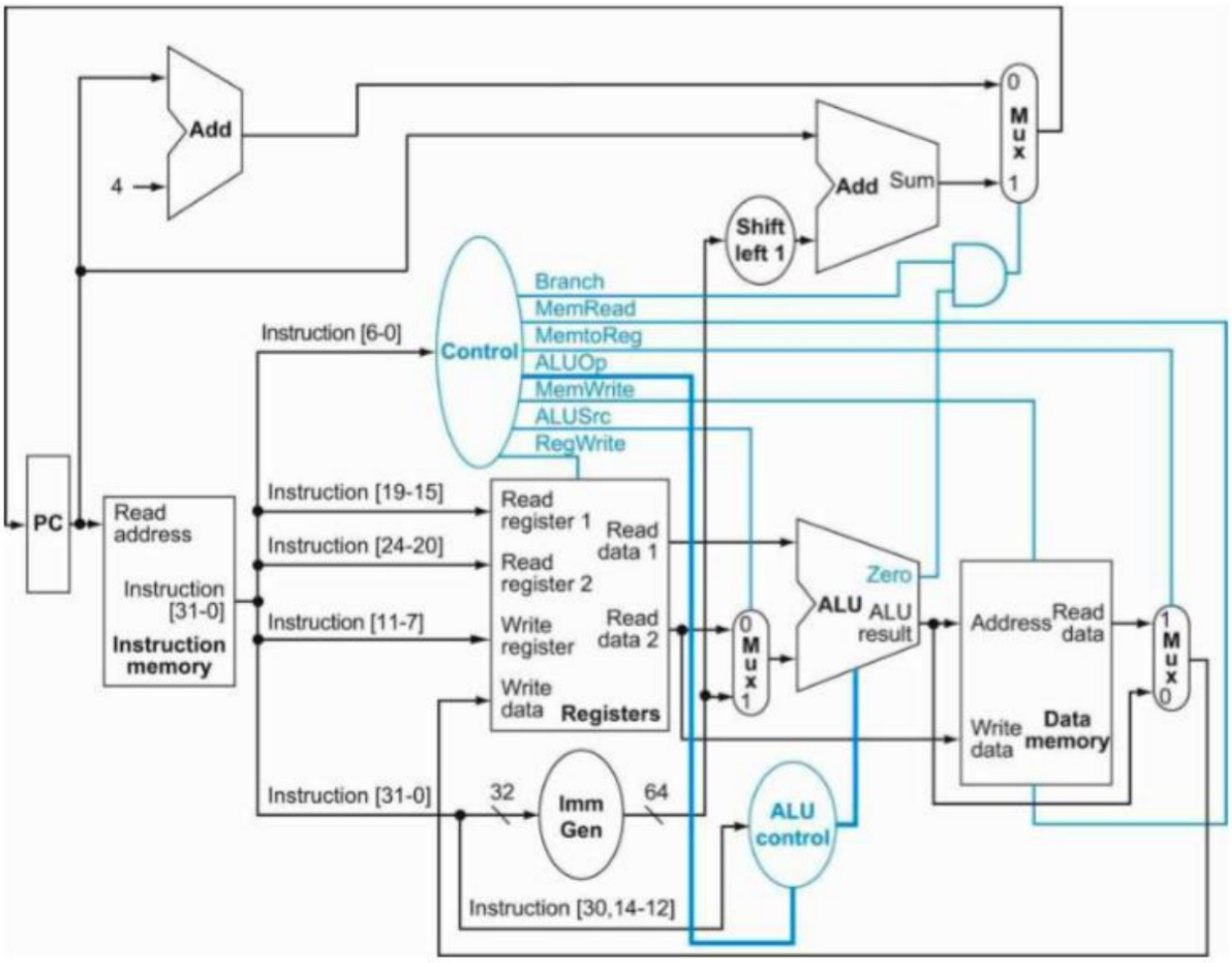
Week 5 discussion

Question 1

Describe the effect that a single stuck-at-1-fault(i.e., regardless of what it should be, the signal is always 1) would have the following signals shown below, in the single-cycle datapath.



Which instructions(R types, load, store, beq), if any, will not work correctly? Consider each of the following faults separately：

- RegWrite=1
- ALUop0=1
- ALUop1=1
- Branch=1
- MemRead=1
- MemWrite=1

The following is the truth-table of the main controller

Truth-Table for the Main Controller

Instruction	ALUSrc	Memto-Reg	Reg-Write	Mem-Read	Mem-Write	Branch	ALUOp1	ALUOp0
R-format	0	0	1	0	0	0	1	0
ld	1	1	1	1	0	0	0	0
sd	1	X	0	0	1	0	0	0
beq	0	X	0	0	0	1	0	1

Input or output	Signal name	R-format	ld	sd	beq
Inputs	I[6]	0	0	0	1
	I[5]	1	0	1	1
	I[4]	1	0	0	0
	I[3]	0	0	0	0
	I[2]	0	0	0	0
	I[1]	1	1	1	1
	I[0]	1	1	1	1
Outputs	ALUSrc	0	1	1	0
	MemtoReg	0	1	X	X
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

Question 2

Object file header			
	Name	Procedure A	
	Text Size	200_{hex}	
	Data Size	72_{hex}	
Text segment	Address	Instruction	
	0	jal $ra, 0400 0200_{hex}	
	…	…	
	48	lw $s0, -8000($gp)	
	…	…	
Data segment	0	(X)	
	…	…	
Relocation information	Address	Instruction type	Dependency
	0	jal	B
	48	lw	X
Symbol table	Label	Address	
	X	-	
	B	-	
Object file header			
	Name	Procedure B	
	Text Size	450_{hex}	
	Data Size	24_{hex}	
Text segment	Address	Instruction	
	0	lw $a0,-7928($gp)	
	4	lw $a1, -7916($gp)	
	8	jal $ra, 0400 0650_{hex}	
	…	…	
Data segment	0	(Y)	

Table 1

	12	(Z)	
	…	…	
Relocation information	Address	Instruction type	Dependency
	0	lw	Y
	4	lw	Z
	8	jal	C
Symbol table	Label	Address	
	Y	-	
	Z	-	
	C	-	
Object file header			
	Name	Procedure C	
	Text Size	380_{hex}	
	Data Size	12_{hex}	
Text segment	Address	Instruction	
	…	…	
	128	sw $a0, -7904($gp)	
	…	…	
Data segment	0	(W)	
Relocation information	Address	Instruction type	Dependency
	128	sw	W
Symbol table	Label	Address	
	W	-	

Table 2

Executable file header		
	Text size	… (a) …
	Data size	… (b) …
Text segment	Address	Instruction
	$0400\ 0000_{hex}$	jal $ra, $0400\ 0200_{hex}$
	…	…
	… (c) …	lw $S0 ,-0x8000 ($gp)
	…	…
	$0400\ 0200_{hex}$	lw $a0 ,-0x7F8E ($gp)
	$0400\ 0204_{hex}$	lw $a1 ,-0x7F7C ($gp)
	$0400\ 0208_{hex}$	jal $ra, $0400\ 0650_{hex}$
	…	…
	$0400\ 0650_{hex}$	…
	…	…
	$0400\ 0778_{hex}$	sw $a0, -7904($gp)
	…	…
Data segment	Address	
	$1000\ 0000_{hex}$	(X)
	…	…
	$1000\ 0072_{hex}$	(Y)
	…	…
	$1000\ 0084_{hex}$	(Z)
	…	…
	… (d) …	(W)
	…	…

Table 3

There are three object files, procedure A, procedure B, and procedure C. Some of their headers and contents are listed in table 1 and table 2.

After linking them into an executable file, we list the header and content of the executable file in `table 3`. Please answer the following questions.

1. What should be in (a) and (b)?
2. What should be in (c)?
3. What should be in (d)?

Question 3

Supposed We have a single-cycle machine with each functional units' execution time as below

Memory Access: 400ps ALU and adders: 150ps Register Access: 100ps

Now we have a program X with 2×10^9 instructions. And the Functional Units used by the instruction class are stated as below

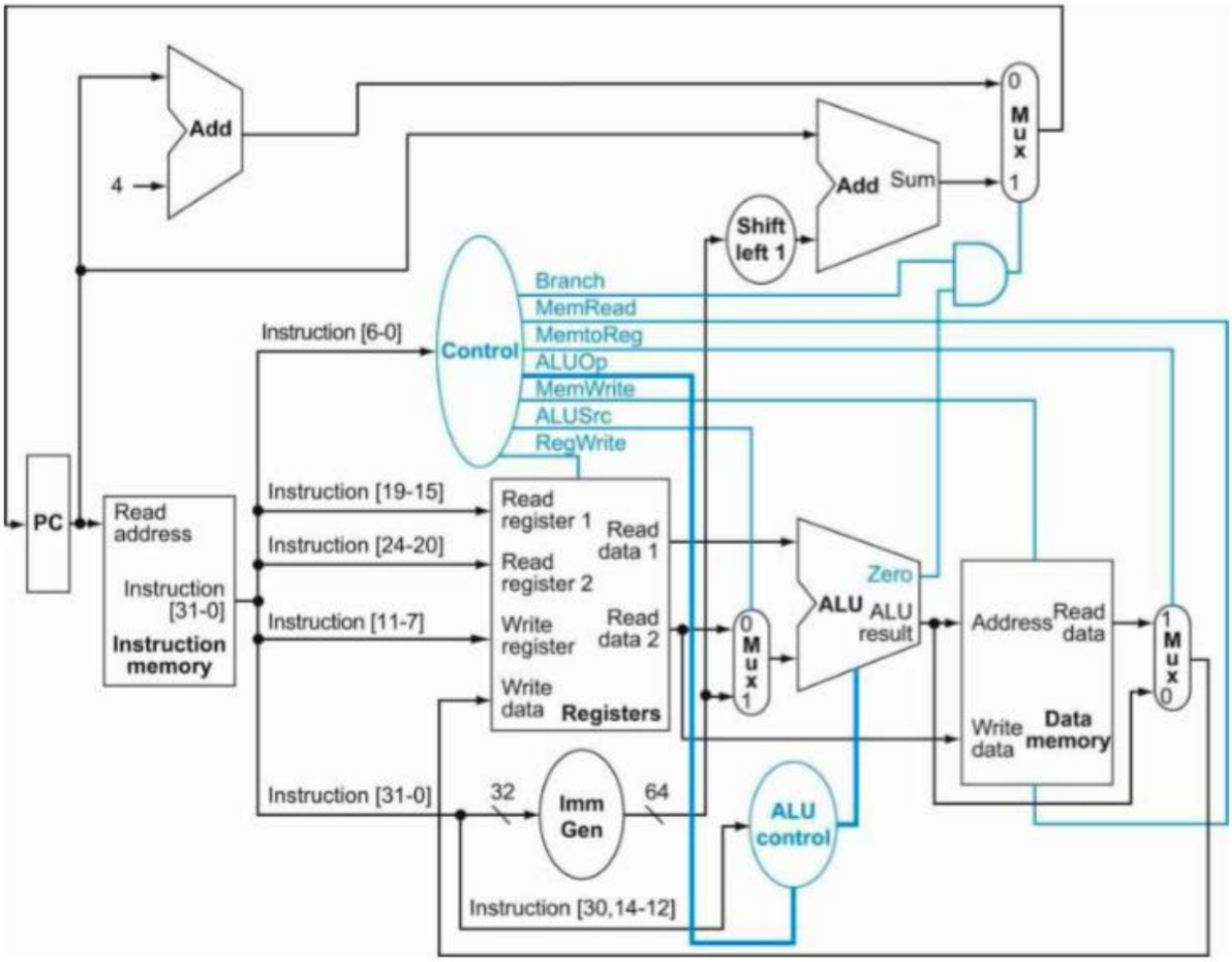
Instruction Class	Functional Units used by the instruction class
R-Type	Inst Fetch, Register Access, ALU, Register Access
Load	Inst Fetch, Register Access, ALU, Memory Access, Register Access
Store	Inst Fetch, Register Access, ALU, Memory Access
Branch	Inst Fetch, Register Access, ALU
Jump	Inst Fetch

Supposed now we have a new type of memory which can make the memory access time become only 100 ps, and the rest of the functional units' execution time remains the same.

1. What is the execution time of program X with the original functional units?
2. What is the execution time of program X with the upgraded memory version?

Question 4

The following diagram is a single-cycle implementation of a subset of RISC-V instructions:



We want to implement a new I-type RISC-V instruction `getpc $rt` which sets register `$rt` to the PC value of this instruction.

Question: Add changes to the given datapath to implement the getpc instruction. Indicate the value of all control signals, including any new control signals.