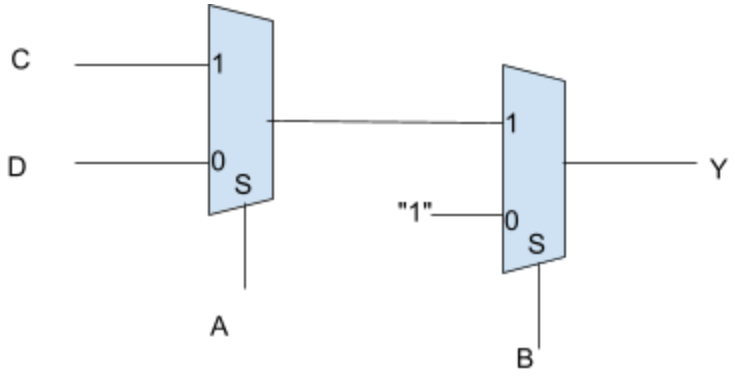
HW10: Digital Logic & Sequential Logic

Not graded

Question 1

Complete the truth table for the following circuit. (that is what is the value of Y for a given set of inputs).

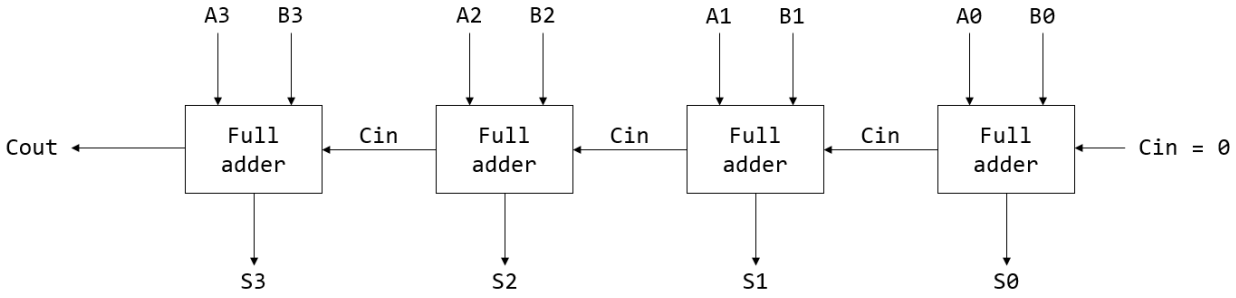


A	B	C	D	Y
0	0	0	0	1
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	

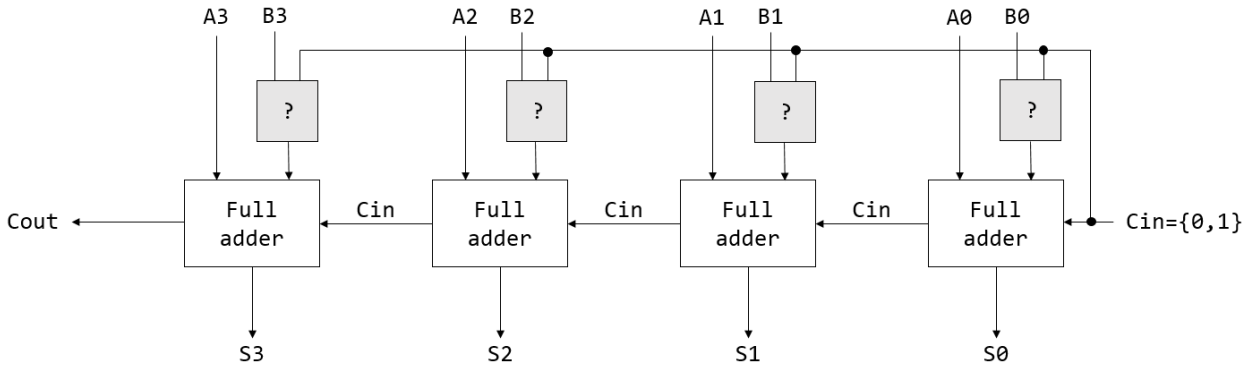
A	B	C	D	Y
1	0	0	0	1
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

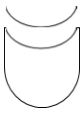
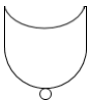
Question 2

Below we have a diagram of a 4-bit ripple carry adder.



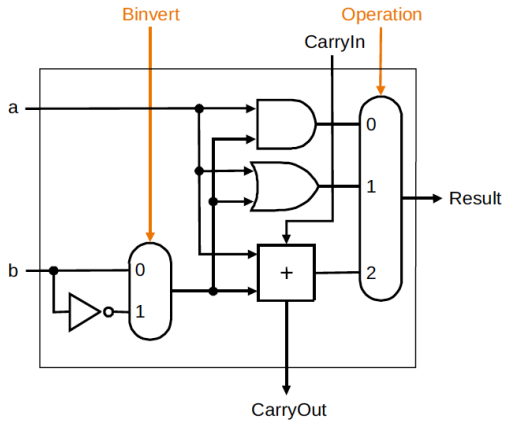
What if we wanted to modify this to also handle subtraction? Instead of the initial carry in (C_{in}) value for the first bit adder being `0`, we can imagine it being either a `1` or a `0`. If $C_{in} = 0$, then our adder will perform addition, but if $C_{in} = 1$, then our adder will perform subtraction. What is the appropriate gate that belongs in the "?" boxes below to achieve this functionality?



A. 	B. 	C. 
D. 	E. 	

Question 7 [3 points]: ALU (graded for correctness)

Given the following diagram of a 1-bit ALU, fill out the table below for what signals will configure the ALU for the given instruction. If a signal doesn't matter, indicate with an "X".



	Binvert	CarryIn	Operation(2-bits)
Add			
Subtract			
And			
Or			
BIC (Bit Clear)			

Question 8 [2 pts]: Machine Code

In our subset of the ARM instruction set, how many bits are used to specify the destination register of an add instruction

Answer	

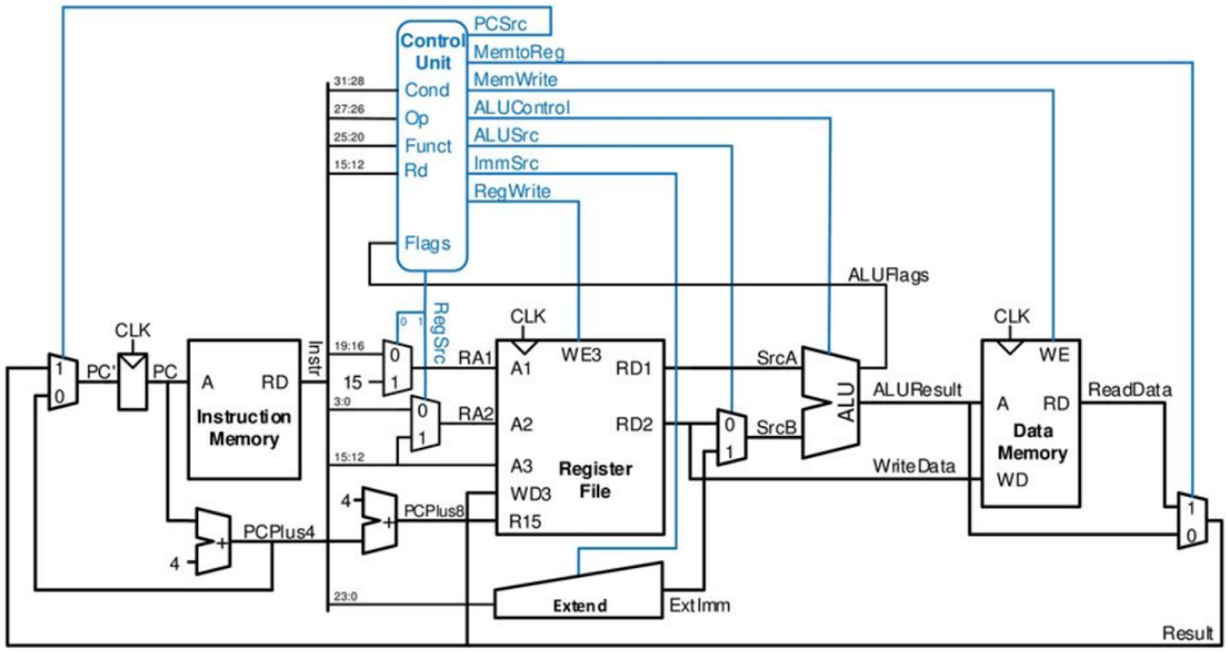
For the ARM instruction data processing instructions, the machine code specifies fields Rd, Rn and Operand2. What value (in binary) would you have in field Rn. for the instruction add r2, r4, r6.

Answer	

a. ldr r9 [r7, r3]

Question 9 [1 pts]: Datapath (graded for completion)

Given this single cycle datapath from lecture, answer the following questions below: (put X if the signal's value doesn't matter)



A. What are the values on RegSrc, MemtoReg, MemWrite, ALUSrc, and RegWrite wire for the following instructions?

Rn Rd R2

a. add r3, r1, r2

RegSrc[1:0]	RA1[3:0]	RA2[3:0]	MemtoReg	MemWrite	ALUSrc	RegWrite

b. str r2, [r3, #100] writing from register to memory (Not writing into any register)

RegSrc[1:0]	RA1[3:0]	RA2[3:0]	MemtoReg	MemWrite	ALUSrc	RegWrite

c. ldr r3, [r4, #7]

RegSrc[1:0]	RA1[3:0]	RA2[3:0]	MemtoReg	MemWrite	ALUSrc	RegWrite

B. So far we have discussed add, sub, ldr, str, b instructions. Answer the following questions:

a. If the RegWrite wire was stuck on 0, which of the above instructions would **NOT** break?

Answer	

b. If the RegSrc[0] wire were stuck at 1, which of the above instructions would break?

Answer	

c. If the ALUSrc wire was stuck on 1, which of the instructions from part A would work correctly?

Answer	