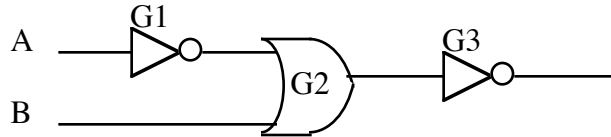


Homework 2

Due: Beginning of class Thursday, February 28th

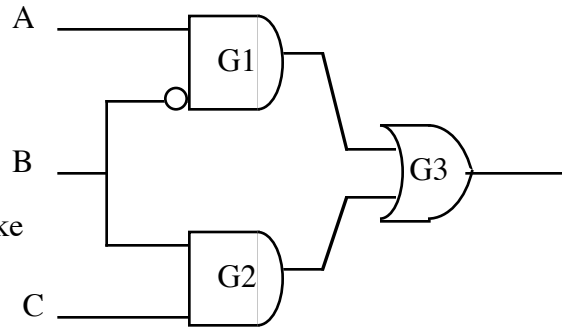
Turnin: Three solutions: 5, 11 and 16. To receive credit for your solutions, you must show your work where applicable. Solutions to the remaining problems are attached.

1. — For the logic circuit shown to the right, complete the truth table. For the labeled gates, G1, G2, and G3, state the output for the given input values of A, B, and C. Note that the output for G3 is also the output of the circuit.



A	B	G1	G2	G3
0	0			
0	1			
1	0			
1	1			

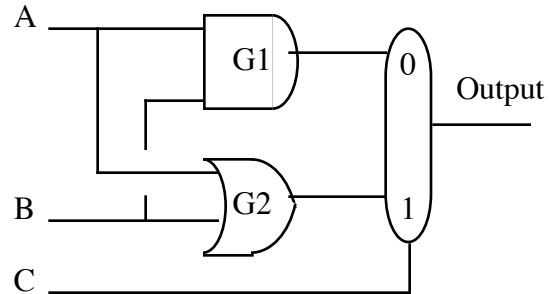
2. — For the logic circuit shown to the right, complete the truth table. For the labeled gates, G1, G2, and G3, state the output for the given input values of A, B, and C. Recall that the small circle, used for one input for G1, represents a NOT gate. You do not have to explicitly show the output of the NOT gate in the truth table, but you do need to take it into account in determining the output of G1. Note that the output for G3 is also the output of the circuit.



A	B	C	G1	G2	G3
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			

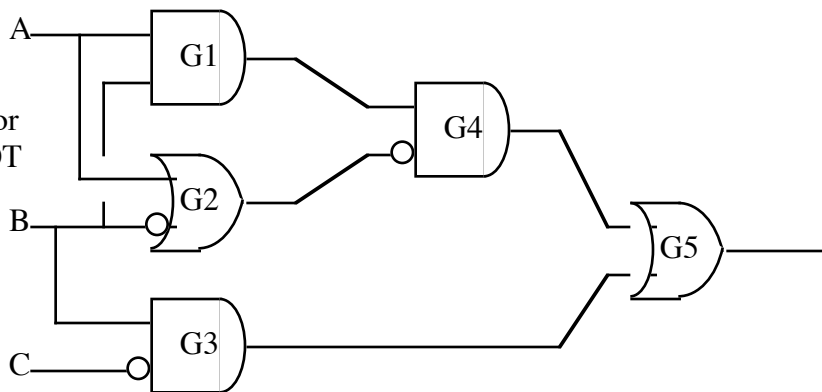
1	1	0			
1	1	1			

3. — For the logic circuit shown to the right, complete the truth table. For the labeled gates, G1, G2, and Output, state the output for the given input values of A, B, and C. Note that Output is from a Multiplexor. The input C is the control line for the multiplexor.



A	B	C	G1	G2	Output
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

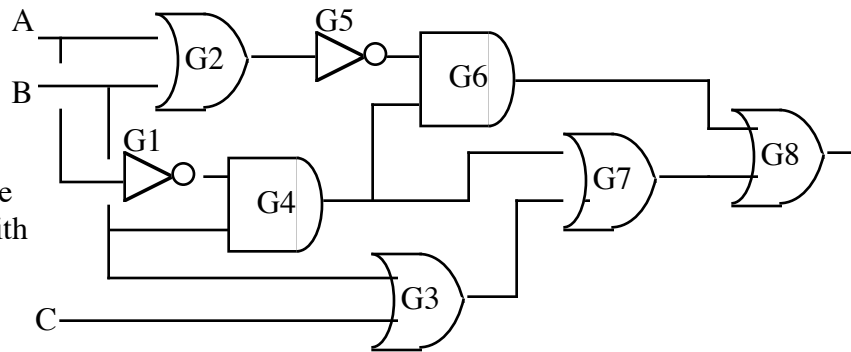
4. — For the logic circuit shown to the right, complete the truth table. Recall that the small circle, used for one input for G2, G3, and G4, represents a NOT gate. You do not have to explicitly show the output of the NOT gates in the truth table, but you do need to take them into account in determining the outputs of G2, G3, and G4. Note that the output for G5 is also the output of the circuit.



A	B	C	G1	G2	G3	G4	G5
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

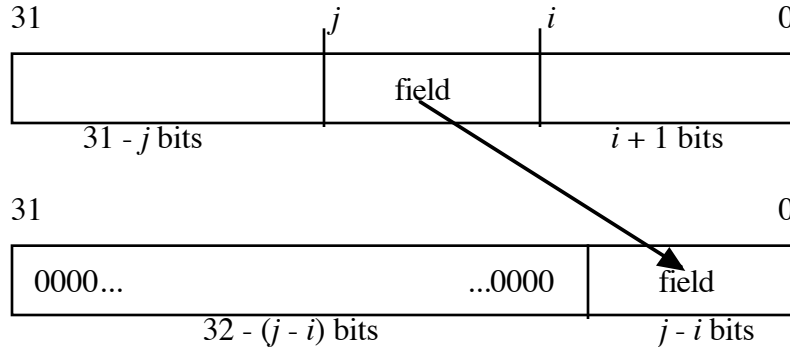
5. — turnin this one:

For the logic circuit shown below, complete the truth table. Note that the output for G8 is also the output of the circuit. (You can fill in the table below and include this page with your solutions.)



A	B	C	G1	G2	G3	G4	G5	G6	G7	G8
0	0	0								
0	0	1								
0	1	0								
0	1	1								
1	0	0								
1	0	1								
1	1	0								
1	1	1								

6. — (Problem 2.6, page 148) [15] <§2.5> Some computers have explicit instructions to extract an arbitrary field from a 32-bit register and to place it in the least significant bits of a register. The figure below shows the desired operation:



Find the shortest sequence of MIPS instructions that extracts a field for the constant values $i = 5$ and $j = 22$ from register $\$t3$ and places it in register $\$t0$. (Hint: It can be done in two instructions.)

7. — Same as the previous problem. This time use $i = 12$ and $j = 22$. Assume the initial value is in register $\$s0$ and the final value will go in register $\$s1$.

8. — This question explores the format of MIPS instructions and points out the inherent problem with just looking at the contents of bytes and trying to figure out what they represent.

Given the 32-bit pattern: **1000 1111 1110 1111 1100 0000 0000 0000**

- What does it represent (in decimal) assuming it is a 32-bit two's complement integer?
- What does it represent assuming it is a MIPS instruction?

9. — Same as the previous exercise, but this time use the 32-bit pattern:

0000 0000 0000 0000 0000 0000 0000 0000

10. — Same as the previous exercise, but this time use the 32-bit pattern:

0000 0001 0001 1001 1000 0000 0010 1011

11. — turnin this one: Same as the previous exercise, but this time use the 32-bit pattern:

1010 0110 0100 1100 1111 1111 0000 0100

12. — problem 3.14, [5] <§3.6>

Instead of using a special hardware multiplier, it is possible to multiply using shift and add instructions. This is particularly attractive when multiplying by small constants. Suppose we want to put nine times the value of $\$s0$ into $\$s1$, ignoring any overflow that may occur. Show a minimal sequence of MIPS instructions for doing this without using a multiply instruction.

13. —

- Same as the previous problem. This time, we want to multiply by 15.
- There is a potential problem with the minimum two-instruction solution from part a that arises when we are not ignoring overflow. What is it?

14. — Masking is a technique that allows certain bits within a word to remain while all the other bits are set to zero. The idea is to create a mask that has **1**'s in the positions you wish to

retain and 0's elsewhere. For example, if we want to keep bits 24 to 31 within a word, but set all the other bits to zero, we can use:

```
1111 1111 0000 0000 0000 0000 0000
```

or, written as a hexadecimal constant:

```
mask: .word 0xFF000000
```

To use the mask, use the **and** operation:

```
la    $t0, mask
lw    $t1, 0($t0)
and   $s1, $s0, $t1
```

Now, **\$s1** contains only the values from **\$s0** that were in the most significant byte with zeroes in the other three bytes.

- a.) Write two shift operations to move bits 8-15 into position 0-7, and leave zeroes in bits 31-8.
- b.) Repeat part a.), but this time use a mask in the manner shown above in the example. You will need one shift instruction as well.
- c.) Repeat part b.), but this time use exactly two MIPS instructions: one instruction to do the mask, plus one instruction to do the shift.

15. —

Same as the previous problem. This time:

- a.) Write three shift operations to move bits 6-13 into position 2-9, and leave zeroes in bits 0-1 and bits 10-31.
- b.) Repeat part a.), but this time use a mask in the manner shown in the example in the previous problem. You will need one shift instruction as well.
- c.) Repeat part b.), but this time use exactly two MIPS instructions: one instruction to do the mask, plus one instruction to do the shift.

16. — turnin this one:

Same as the previous problem. This time:

- a.) Write three shift operations to move bits 5-12 into position 23-30, and leave zeroes in bits 0-22 and bit 31.
- b.) Repeat part a.), but this time use a mask in the manner shown in the example in the previous problem. You will need one shift instruction as well.
- c.) Repeat part b.), but this time use exactly two MIPS instructions: one instruction to do the mask, plus one instruction to do the shift.