Max Score = 15 points

CS 250 2018 Spring Homework 03 SOLUTION & GRADING GUIDE

This assignment is due at 11:59:00 pm Thursday, February 01, 2018.

Upload your typewritten answer document in either PDF or Word format to Blackboard. Download from Blackboard to be sure that your upload was successful. Your last upload that is not marked "LATE" by Blackboard is the upload that will be graded. There is no "grace" period for late uploads.

1. **[3 points; 1 point for the answer Reset only]** What sequence of commands to an S'R' latch will certainly produce a falling edge on the latch output Q?
   Answer: Give the latch a Set command followed by Reset command. All ways of communicating this sequence, such as SR or S'R' or Set followed by Reset, are acceptable answers.
   A falling edge is a transition from a logic 1 to a logic 0. So we must cause latch output at time x, Q(x)=1, to be followed by Q(x+1)=0. There are two cases to consider, each with its own way of producing the falling edge.
   Case 1: If it happens that Q(t)=1, then make R'=0 causing Q(t+1)=0 and generating the intended falling edge.
   Case 2: If it happens that Q(t)=0, then first make S'=0 causing Q(t+1)=1. Next, make S'=1 to make Q(t+2)=1, finally make R'=0 causing Q(t+3)=0 which generates the intended falling edge and does not present S'=R'=0 to the latch.

2. **[3 points; 1 point each fpr parts a, b, and c]** Regarding the decoder and ripple-carry adder.
   a. What is the number of gates required to construct a 5x32 decoder?
   [1 point] Answer: There is one NAND gate for each of the 32 outputs and there is one inverter for each of the 5 inputs, for a total of 37 gates.

   b. In units of gate delays, what is the propagation delay of the decoder?
   [1 point] Answer: The worst case propagation delay is for any decoder output that is a function of an inverted input. The path consists of one inverter and one NAND gate for a total propagation delay of 2 gate delays.

   c. How does this compare to the number of gates needed to build a 5-bit ripple-carry adder, an adder that accepts inputs represented using 5 bits, and to the propagation delay of the ripple-carry adder?
   [1 point] Answer: The ripple-carry adder requires 5 gates per operand bit, so the total for 5-bit operands is 25 gates. The propagation delay is 2n+1, where n is the number of bits in the operands, and thus is 11 gate delays.

   d. What is the ratio of the propagation delay of the decoder to that of the adder?
   Answer: The decoder propagation delay is 2, for the adder, 11. The ratio of delays decoder to adder is 1 : 5.5. Or viewed another way, pointing takes 1/5.5 = 18% as much time as adding, given 5-bit input size.

3. **[3 points; 1 point each for the three answers in bold in the table]** Fill in the following table to show how the given meaningless bit string is interpreted according to each of the six data representations: unsigned integer, sign magnitude, 1's complement, 2's complement, packed BCD, and ASCII character.
Express your answers using base 10 digits, hexadecimal notation, or ASCII character, whichever is most appropriate. Show a + sign or a – sign as appropriate for zero values when the representation permits both possibilities. If the bit string does not have a valid interpretation for a given representation, write "Error" in the space provided.

Bit string	Unsigned integer	Sign magnitude	1's comp-lement	2's comp-lement	Packed BCD	7-bit ASCII
00000000	0	+0	+0	0	00	nul
00000111	7	7	7	7	07	**bel**
10000000	128	**-0**	-127	**-128**	80	Error
11111110	254	-126	-1	-2	Error	Error

4. What is the 16-bit representation of the 2's complement number 11101010?
[3 points] Answer: 1111111111101010.

5. If propagation delay in a combinatorial circuit is measured in gate delays, how long before all outputs are valid for an 8-bit ripple carry adder circuit?
Answer: 17 gate delays, 1 gate delay plus 2 gate delays per bit to allow for the carry ripple to the next more significant bit position.

6. Complete the table to show how the bit string is written in octal or hexadecimal notation.
Answer:

Bit string	Written using octal notation	Written using hexadecimal (0x) notation
110011011111	6337	0xCDF
010111110001	2761	0x5F1

7. **[3 points; 1 point for each correctly sequenced byte up to a maximum of 3 points]** The 32-bit string 0x01A500FF is stored in little endian byte-addressed memory. Reading from memory starting at the lowest numbered address that contains a byte of this 32-bit string and incrementing the memory address three times will produce what sequence of bytes?
Answer: 0xFF then 0x00 then 0xA5 then 0x01.

8. An active high pushbutton switch input circuit has a poorly performing pushbutton characterized by a measured resistance of 100,000 ohms when open (not button is not pushed) and 1000 ohms when closed (the button is pushed). The fixed resistor in this circuit has a measured value of 10,000 ohms. If the reference logic level voltages are +5 volts and 0 volts, what are the voltages of $\text{Output}_{\text{button\_pushed}}$ and $\text{Output}_{\text{button\_not\_pushed}}$?
Answer:
An active high circuit, will have the pushbutton switch connected to +5 volts and the 10,000 ohm fixed resistor connected to ground. In our usual notation for a voltage divider, the

switch is R1 and the fixed resistor is R2. Pushing the button will generate a high output voltage. How high?
$\text{Output}_{\text{button\_pushed}} = V_{\text{high\_reference}}$ (R2 / (R1 + R2))
= 5 volts * (10,000 ohms / (1,000 ohms + 10,000 ohms)
= 5 volts * 10,000/11,000
= 4.55 volts.
$\text{Output}_{\text{button\_not\_pushed}}$ = 5 volts * (10,000 ohms / (100,000 ohms + 10,000 ohms)
= 5 volts * 10,000/110,000
= 0.45 volts.

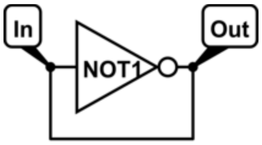
9. Plot the logic signal waveforms for signals In and Out for the following circuit using the supplied graph paper. You must plot the two waveforms for enough time to show how each waveform repeats. When creating your timing plot assume the following:

   a. At time = 0 ns (nanoseconds, 10^-9 second), the logic level at circuit node In = 1, and the logic level at Out = 1. The nodes In and Out are, respectively, the place of input to gate NOT1 and the place of output of NOT1. There is zero propagation delay between In and NOT1 input, nor any propagation delay between NOT1 and Out.

   b. The propagation delay of the gate NOT1 itself is *T* ns where *T* is either (a) the number of letters in the spelling of your preferred first name as shown in Blackboard or (b) the decimal number 9 if the number of letters from part (a) is 10 or more. [Across all students in the class, we will solve a set of related versions of this question.]

      Type the name you are using to determine the NOT1 propagation delay that you will use when answering this question here ___________________________________.
      Enter the number of letters in your name or 9, per instructions above, here ______ .

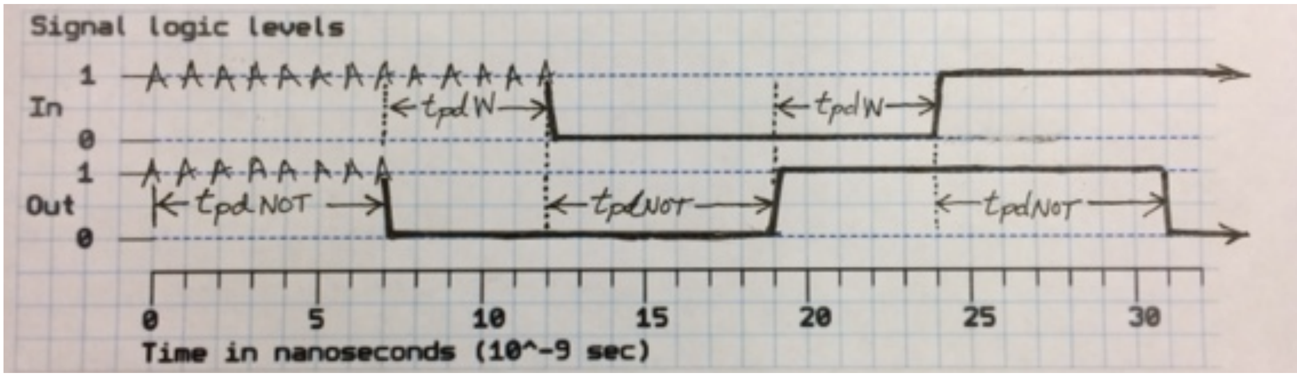
   c. The propagation delay through the wire from node Out to node In is *S* ns where *S* is defined as follows. If your lab session is on Tuesday, then *S* = 2. If your lab session is on Wednesday, then *S* = 3. If your lab session is on Thursday, then *S* = 4. If your lab session is on Friday, then *S* = 5.
      Enter your value for *S* here __________________ .



The generic graph here is shown in terms of t_{pdNOT}, the time of propagation delay through the NOT gate in units of nanoseconds (ns) and t_{pdW}, the time of propagation delay through the wire from Out to In. The A symbols show for how long the voltage values must be assumed for Out and In.

The range of values for t_{pdNOT} is 9 ns down to 3 ns, based on preferred first name, and 3 ns through 5 ns for the wire propagation delay. The drawing below is labeled using the variable names because all waveforms have the same shape. The differing propagation delay values only expand or contract the duration (width on the plot) of the corresponding interval. (In the drawing actual value for t_{pdNOT} is 7 ns and for t_{pdW} is 5 ns.) The pattern is that after an interval of t_{pdNOT} from the start time the NOT gate output, Out, changes. Then after a delay of t_{pdW}, the voltage at In changes to follow the voltage of Out. This starts a new oscillation, and t_{pdNOT} later the voltage at Out switches level. t_{pdW} later the voltage at In switches levels to follow Out, and the pattern continues.

I have drawn the rising and falling edges rather steeply because this makes it easier to see that the total time is just an accumulating sequence of t_{pdNOT} and t_{pdW} intervals and, besides, gates are designed to switch voltage levels fast.


Signal logic levels
In
Out
tpdW
tpdW
tpd NOT
tpdNOT
tpdNOT
Time in nanoseconds (10^-9 sec)