CSEE 3827: Fundamentals of Computer Systems

Midterm Exam

March 8, 2010

Name: ______________________________ UNI: ____________________

Read all of the following information before starting the exam:

- Be sure to write your name on each page of the exam.
- Use the exam itself for your solutions (no blue books or spare sheets of paper). You may use the backside of pages if you need more space.
- Show your work in order to earn partial credit.
- You may use your textbook and class notes, but *absolutely no electronic devices (laptops, cell phones, etc.)*
- Good luck!

Problem	**Point Value**	**Points Earned**
1	14	
2	10	
3	10	
4(a)	10	
4(b)	10	
4(c)	10	
Total	64	

1. **(14 points)** Answer the following short answer questions:

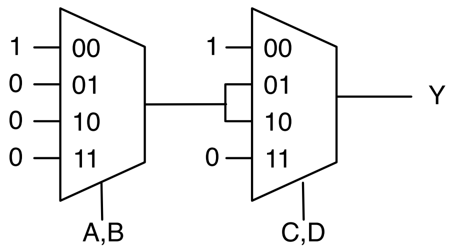
   (a) **(2 points)** The Babylonians developed the *sexagesimal* (base 60) number system about 4000 years ago. How many bits of information is conveyed with two sexagesimal digits? How do you write the number 4000_{10} in sexagesimal? (To write a number in sexagesimal, use a space to separate digits, e.g., 13 53 26)

   (b) **(2 points)** How many unique boolean functions of N variables are there?

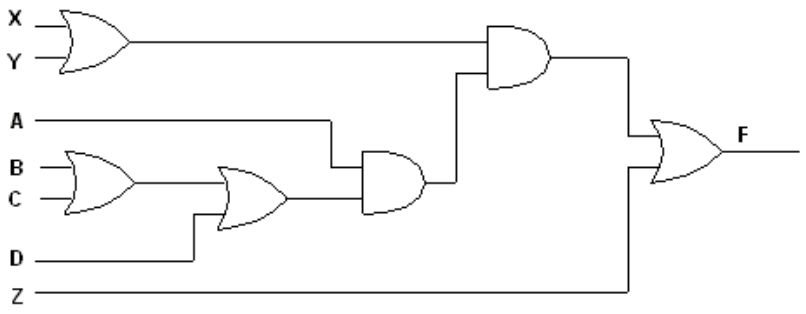
   (c) **(2 points)** A *sign extension unit* extends a signed number from M to N bits ($N > M$) by copying the most significant bit of the input into the upper bits of the output. It receives an M-bit input, A, and produces an N-bit output, Y. Draw a circuit for a sign extension unit with a 4-bit input and an 8-bit output.

   (d) **(2 points)** Explain why a designer might choose to use a ripple-carry adder instead of a carry-lookahead adder. A phrase or two is sufficient.

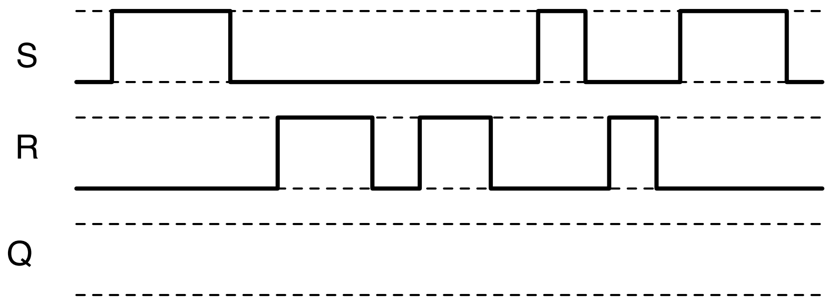
(e) **(2 points)** Give a boolean expression for the function performed by the following circuit:



(f) **(2 points)** What is the propagation delay (in unit gate delays) of this circuit?



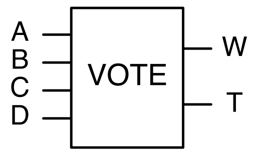
(g) **(2 points)** Given the input signals shown below, sketch the output, Q, of an SR latch.



2. **(10 points)** An n-input *vote* circuit takes n inputs and produces two outputs:

- **W**, the "winner" of the vote: 1 if the majority of the inputs are true, and 0 if the majority of the inputs are 0
- **T**, indicates that there was a "tie": 1 if the inputs are evenly split, in which case there was no winner

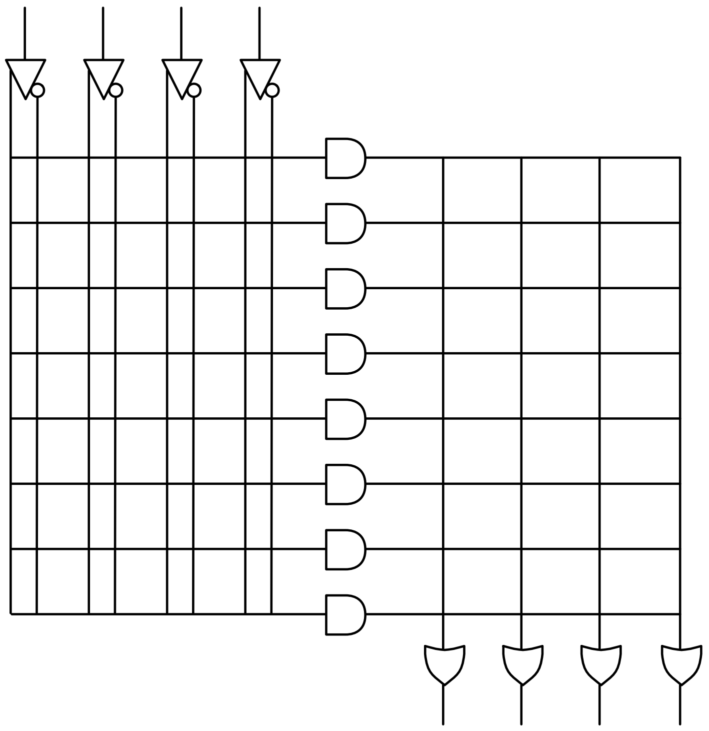
(a) **(4 points)** Complete a truth table for the four-input vote module shown below:



(b) **(3 points)** Give minimal SoP expressions for the VOTE module.

(Problem continues on next page.)

(c) **(3 points)** Show an implementation of the VOTE module using the PLA below:



Name: ______________________________

3. **(10 points)** Build a 32-bit *up/down counter*. The inputs are CLK, RST and UP. When RST=1, the outputs are all 0. Otherwise, when UP=1, the circuit counts up, and when UP=0 the circuit counts down.

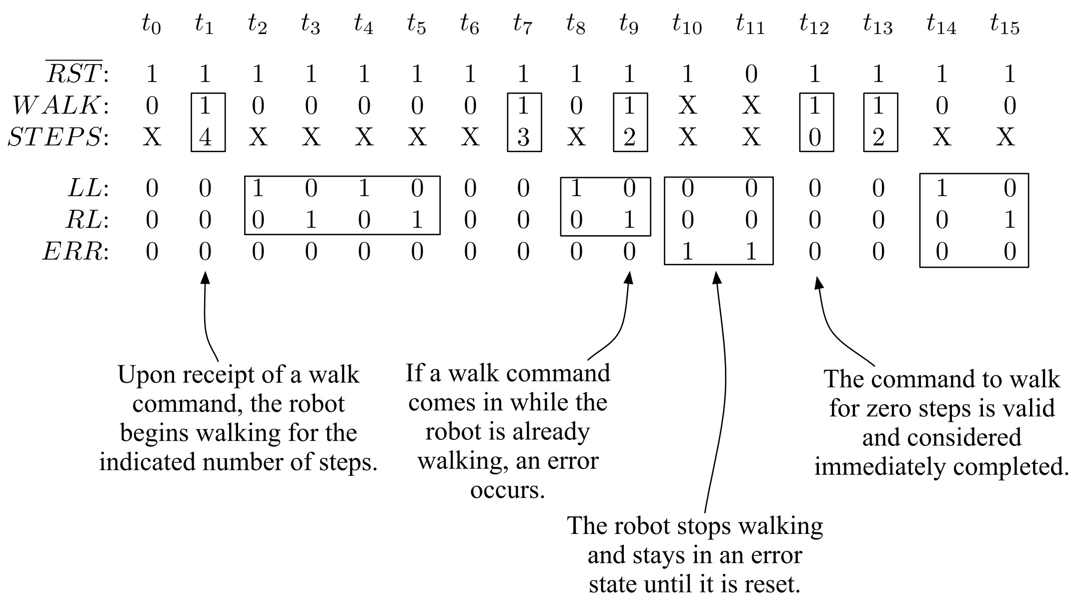
4. **(30 points)** In this exercise you will implement a walking robot module, ROBOT. The specification for this robot is as follows. Upon receipt of a walk command the ROBOT will begin walking (engaging its left leg, then its right leg, then its left and so on) for the number of steps specified at the time of the walk command. It will then stop and await the next walk command. The ROBOT takes the following inputs:

   - CLK
   - $\overline{RST}$, an active-low reset signal, which brings the robot to a standstill
   - $WALK$, a single-bit signal that, when high indicates that the robot should begin walking
   - $STEPS$, a four-bit number indicating the number of steps the robot should take

   The module produces the following signals

   - LL, "left leg" is 1 when the left leg is engaged
   - RL, "right leg" is 1 when the right leg is engaged
   - ERR, indicates that there has been an error

   Here is an example trace of the machine that highlights the desired behavior.

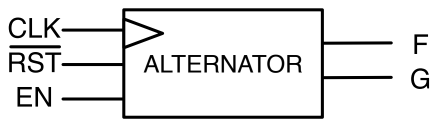


	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}	t_{15}
$\overline{RST}$:	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
$WALK$:	0	1	0	0	0	0	0	1	0	1	X	X	1	1	0	0
$STEPS$:	X	4	X	X	X	X	X	3	X	2	X	X	0	2	X	X
LL:	0	0	1	0	1	0	0	0	1	0	0	0	0	0	1	0
RL:	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	1
ERR:	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Upon receipt of a walk command, the robot begins walking for the indicated number of steps.
If a walk command comes in while the robot is already walking, an error occurs.
The robot stops walking and stays in an error state until it is reset.
The command to walk for zero steps is valid and considered immediately completed.

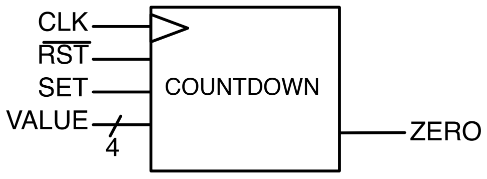
   We will implement this robot component by component. All components should be Moore machines. You may specify them using schematics or boolean expressions if you prefer. If you choose the latter be sure your signals are clearly labelled. Be sure to show your work for partial credit.

(a) **(10 points)** First, implement the ALTERNATOR module, as specified below.



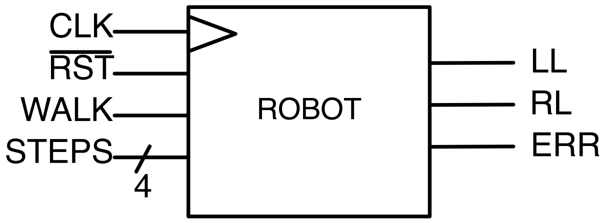
- When EN=1, the outputs F and G alternate between F=1,G=0 and F=0,G=1.
- When EN=0, F=0 and G=0.

(b) **(10 points)** Next, implement the COUNTDOWN module shown below. You may use basic components, such as registers and adders, if you like.



- When SET=1, the counter is initialized to VALUE.
- When SET=0 the counter counts down to 0.
- When the counter reaches zero, ZERO=1, and remains there until the counter is set to a new value.

(c) **(10 points)** Finally, assemble the COUNTDOWN and ALTERNATOR to implement the ROBOT module as specified at the start of the problem. For reference, the ROBOT interface is shown here.



Name: ______________________

(Scratch space.)