

UNCC – ECGR2181- Midterm Exam 2 – November 20, 2009

A

Name: Solution

Mosaic User ID _____

You are permitted 75 minutes to take this test, no more. You are allowed the following items for the test: single sheet of paper with notes, pencils and erasers. You are not permitted to have any of the following on your desk during the test: calculator, books, notes, homework, labs computer, cell phone, or other electronic assistance. Failure to abide by this policy will result in a zero for the test and a visit to the UNC Charlotte honor board. Put your answers on the scan sheet and the paper provided (pages 4 to 7 of the test), and turn in the scan sheet and the answer pages - use only that paper.

For these multiple choice, True/False, and ordering problems, fill in the letter bubble of the SINGLE best answer for each problem. Multiple choice are 5 points, true/false are worth 2 points, put in order are 2 points.

- 1) The ASCII string "90210" is represented by the values 0xF9F0F2F1F0
A) TRUE B) FALSE *would be 0x3930323130*
- 2) VHDL, Verilog, and Xilinx are all examples of software languages used to synthesize computer circuits.
A) TRUE B) FALSE
- 3) VHDL is case sensitive.
A) TRUE B) FALSE
- 4) A Half Adder can always be used in the place of Full Adder, provided you assign the correct value to C-in.
A) TRUE B) FALSE *other way around*
- 5) The duty cycle of a clock signal is the percentage of time it is logical 1.
 A) TRUE B) FALSE

- 6) Which of the following is not true for VHDL:
 - a. Case statements can be used inside or outside of process statement
 - b. Signals can be concatenated with other signals to treat them as a whole
 - c. & is the concatenation operator
 - d. '--' means that the rest of the line is a comment
 - e. All of these statements is true

All answers are accepted.

- 7) If you add the following six-bit 2's complement numbers, which one will result in an overflow?

a. 111111 + 000100	$(- \#) + (+ \#)$	<i>Never overflow</i>
b. 011111 + 101000	$(+ \#) + (- \#)$	<i>Never overflow</i>
c. 110000 + 111000	$(- \#) + (- \#)$	<i>still in range</i>
<input checked="" type="radio"/> d. 011111 + 000100	$(+ \#) + (+ \#)$	<i>very big</i>
e. None of the above		

- 8) In a half adder, what gates are used?
 - a. AND/OR
 - b. XOR/OR
 - c. NOT/AND
 - d. AND/XOR
 - e. XNOR

- 9) How many select bits are needed in a 64 bit multiplexer?
 - a. 6
 - b. 8
 - c. 2
 - d. 64
 - e. 32

$$2^n = 64$$

$$n = 6$$

- 10) The simplest device that stores either a logic 0 or 1 is:
 - a. Memory *big*
 - b. Flip-flop
 - c. A not gate *does not store*
 - d. RAM *big*
 - e. A 4-bit register *4-Flip flops*

UNCC – ECGR2181- Midterm Exam 2 – November 20, 2009

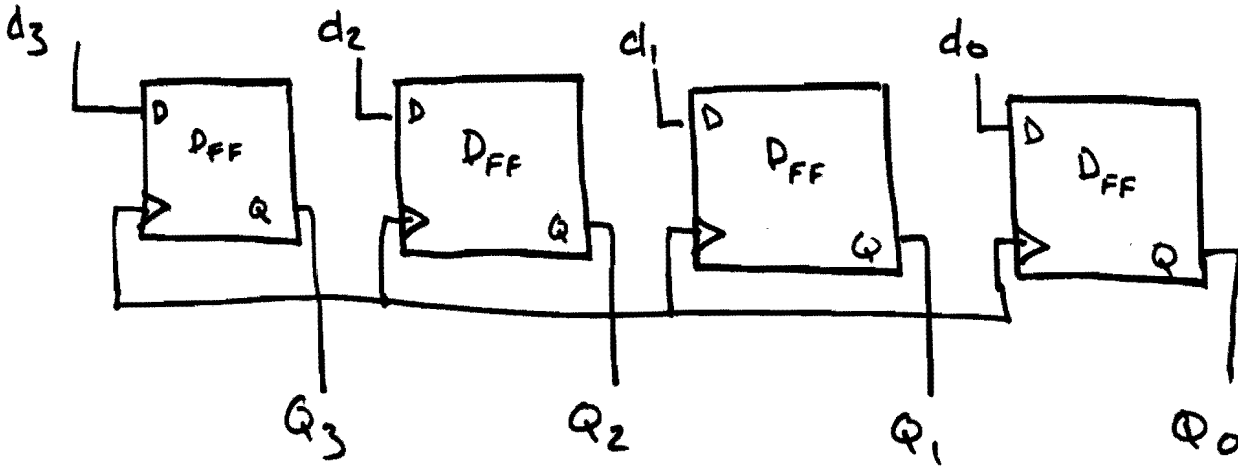
Name: SOLUTION Mosaic User ID _____

Question	1-21	22	23	24	25	26	27	28	Total
Score	/75	/5	/5	/10	/10	/10	/15	/20	/150

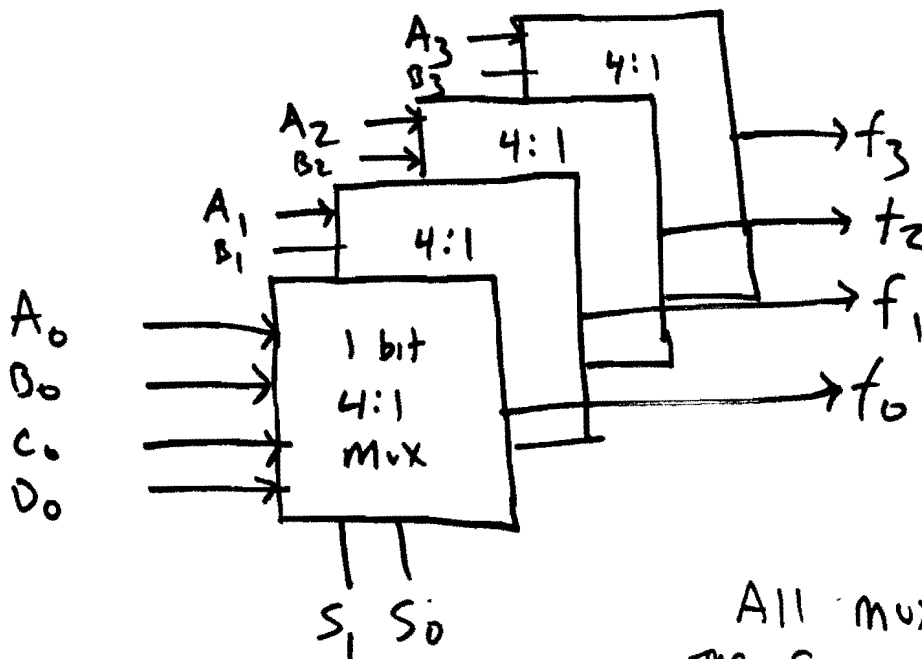
Please read and sign this statement: I have not received from anyone nor assisted others while taking this test. I have also notified the test proctor of any of these violations noted above.

Signature: _____

22. Build a 4-Bit Register from pre-existing D Flip-Flops. Label all Input and Outputs. (5 Points)



23. Build a 4-Bit 4x1 Multiplexer from pre-existing 1-Bit 4x1 Multiplexers. Label all Input and Outputs. (5 Points)



All muxes share the same select lines and have A-D respectively

24. Perform Binary Multiplication on the following two unsigned binary numbers. Show ALL work! Then list the necessary components (AND, NOT, XOR, Adder, D-Flip Flop, Multiplexer...) to build the Multiplier out of Gates. (10 points)

$$\begin{array}{r}
 1101 \\
 \times 1001 \\
 \hline
 1101 \\
 0000 \\
 0000 \\
 + 1101 \\
 \hline
 111001
 \end{array}$$

AND Gates
4 - Bit Adders

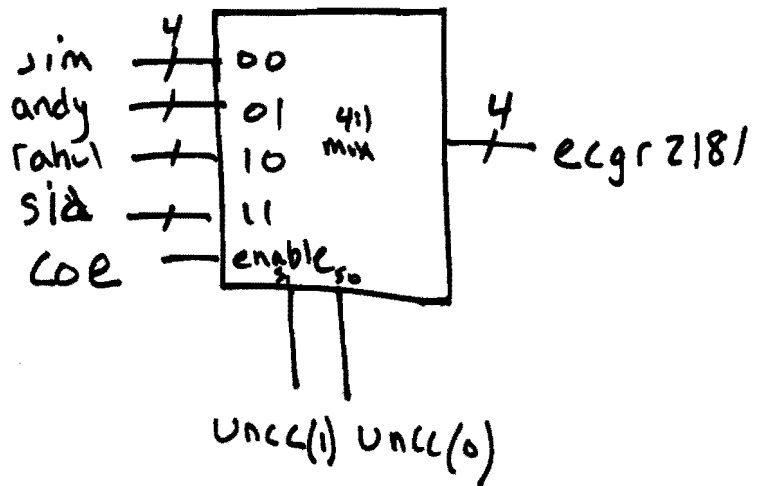
25. Given the following VHDL code identify and draw the component. Label all Inputs and Outputs. (10 Points)

```

entity my_comp is port (
  jim   : in std_logic_vector(3 downto 0);
  andy  : in std_logic_vector(3 downto 0);
  rahul : in std_logic_vector(3 downto 0);
  sid   : in std_logic_vector(3 downto 0);
  uncc  : in std_logic_vector(1 downto 0);
  coe   : in std_logic;
  ecgr2181 : out std_logic_vector(3 downto 0));
end my_comp ;

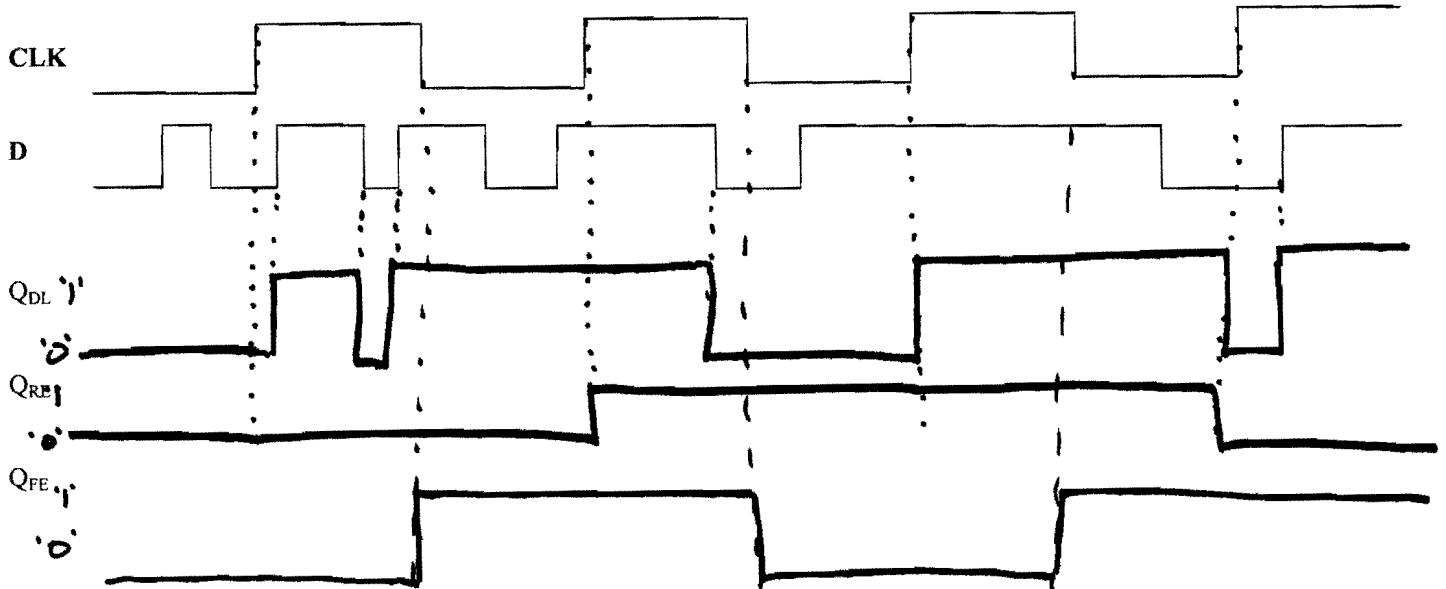
architecture beh of my_comp is
begin
  my_process : process (coe, jim, andy, rahul, sid, uncc) is
  begin
    if (coe = '1') then
      case (uncc) is
        when "00" => ecgr2181 <= jim;
        when "01" => ecgr2181 <= andy;
        when "10" => ecgr2181 <= rahul;
        when "11" => ecgr2181 <= sid;
        when others => ecgr2181 <= (others => '0');
      end case;
    else
      ecgr2181 <= (others => '0');
    end if;
  end process my_process;
end beh ;
  
```

4-bit 4:1 Multiplier
with an enable bit



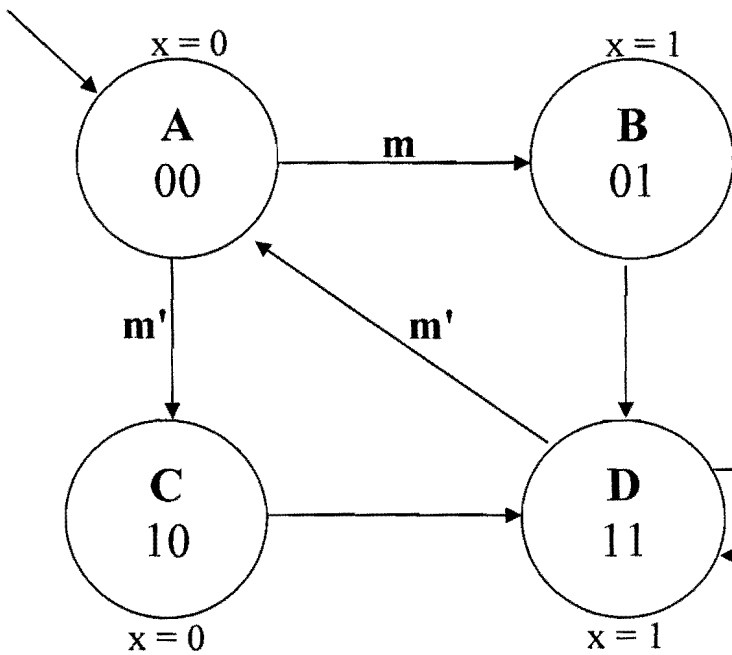
26. Given the following Clock (CLK) and Data Input (D) draw: (10 Points)

- a. Level Sensitive D Latch Waveform Outputs (Q_{DL})
- b. D Flip-Flop (Rising Edge) Waveform Output (Q_{RE})
- c. D Flip-Flop (Falling Edge) Waveform Output (Q_{FE})



27. Given the Finite State Machine below list the following: (15 Points)

- d. List all State Names: A, B, C, D
- e. List the Initial State: A
- f. Complete the State Table



	Inputs			Outputs		
	s1	s0	m	n1	n0	x
A	0	0	0	1	0	0
A	0	0	1	0	1	0
B	0	1	0	1	1	1
B	0	1	1	1	1	1
C	1	0	0	1	1	0
C	1	0	1	1	1	0
D	1	1	0	0	0	1
D	1	1	1	1	1	1

28. Design a circuit that for the function: more_ones(A, B) (20 Points)

Inputs: A and B are 3-bit unsigned binary numbers

Output: '1' - when A has more ones than B.

'0' - when B has more ones than A or B has the same number of ones as A.

For Example: Given A = 110 and B = 001 then the output is '1' because A has two 1's while B only has one.

Given A = 000 and B = 100 then the output is '0' because A has zero 1's while B has one.

Given A = 010 and B = 100 then the output is '0' because both A and B have one.

(Hint: What components would you need and how would they be connected? Think about past what you have done in the past to compare two numbers and what you would compare in this circuit.)

