University *of* Alexandria
Faculty of Engineering
Division of Communications & Electronics

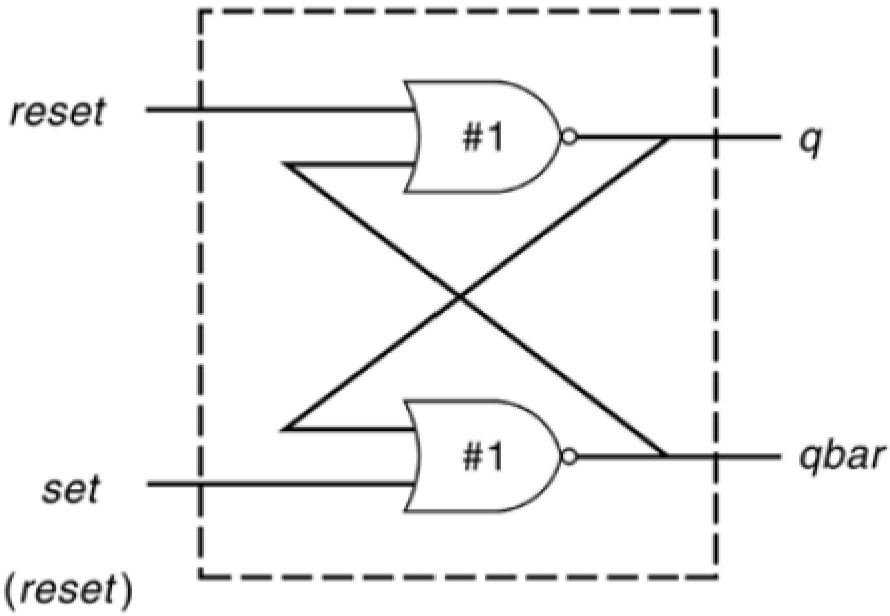
Subject Name: VLSI
Academic Year: 2012 – 2013

Lecturer: Dr. Mohammed Morsy
Assistants: Eng. Ahmed Bedewy

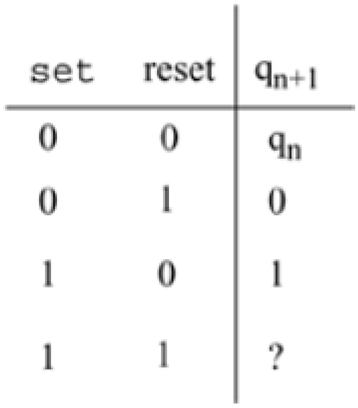
Fourth Year – Semester 2

SHEET 4

1-The logic diagram for an RS latch with delay is shown below.

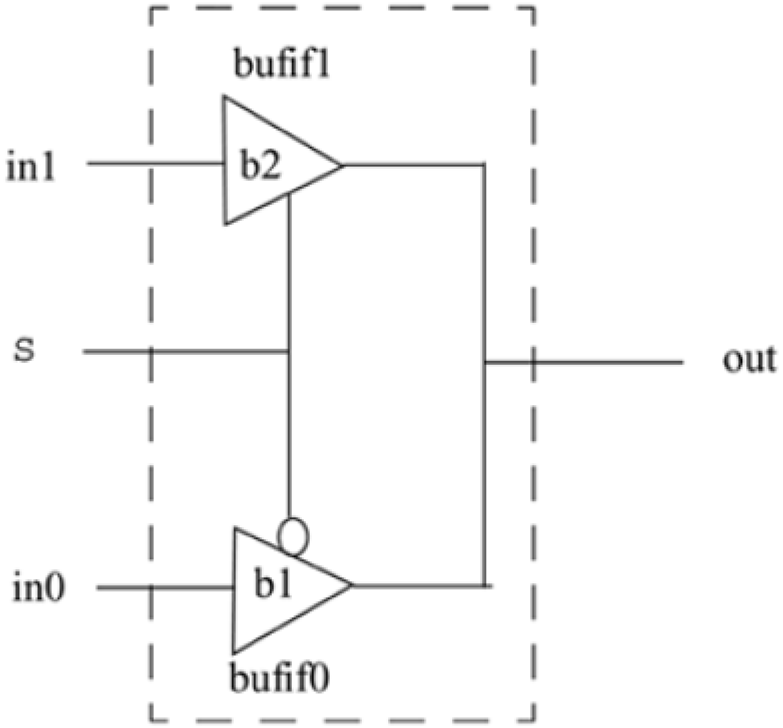


Write the Verilog description for the RS latch. Include delays of 1 unit when instantiating the nor gates. Write the stimulus module for the RS latch, using the following table, and verify the outputs.



set	reset	q_{n+1}
0	0	q_n
0	1	0
1	0	1
1	1	?

2- Design a 2-to-1 multiplexer using bufif0 and bufif1 gates as shown below.



The delay specification for gates b1 and b2 are as follows:

	Min	Typ	Max
Rise	1	2	3
Fall	3	4	5
Turnoff	5	6	7

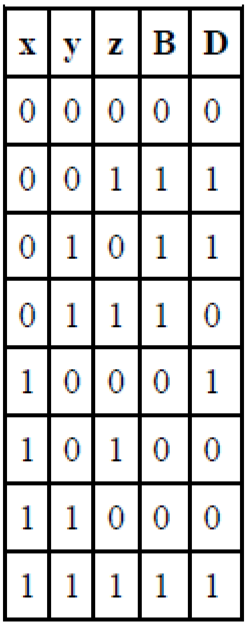
Apply stimulus and test the output values.

3-A full subtractor has three 1-bit inputs x, y, and z (previous borrow) and two 1-bit outputs D (difference) and B (borrow). The logic equations for D and B are as follows:

D = x'.y'.z + x'.y.z' + x.y'.z' + x.y.z

B = x'.y + x'.z +y.z

Write the full Verilog description for the full subtractor module, including I/O ports (Remember that + in logic equations corresponds to a logical or operator (||) in dataflow). Instantiate the subtractor inside a stimulus block and test all eight possible combinations of x, y, and z given in the following truth table.



x	y	z	B	D
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

4- A magnitude comparator checks if one number is greater than or equal to or less than another number. A 4-bit magnitude comparator takes two 4-bit numbers, A and B, as input. We write the bits in A and B as follows. The leftmost bit is the most significant bit.

A = A(3) A(2) A(1) A(0)

B = B(3) B(2) B(1) B(0)

The magnitude can be compared by comparing the numbers bit by bit, starting with the most significant bit. If any bit mismatches, the number with bit 0 is the lower number. To realize this functionality in logic equations, let us define an intermediate variable. Notice that the function below is an xnor function.

x(i) = A(i).B(i) + A(i)'.B(i)'

The three outputs of the magnitude comparator are A_gt_B, A_lt_B, A_eq_B. They are defined with the following logic equations:

A_gt_B = A(3).B(3)' + x(3).A(2).B(2)' + x(3).x(2).A(1).B(1)' + x(3).x(2).x(1).A(0).B(0)'

A_lt_B = A(3)'.B(3) + x(3).A(2)'.B(2) + x(3).x(2).A(1)'.B(1) + x(3).x(2).x(1).A(0)'.B(0)

A_eq_B = x(3).x(2).x(1).x(0)

Write the Verilog description of the module magnitude_comparator. Instantiate the magnitude comparator inside the stimulus module and try out a few combinations of A and B.

5-Design a clock with time period = 40 and a duty cycle of 25% by using the always and initial statements. The value of clock at time = 0 should be initialized to 0.

6-Given below is an initial block with blocking procedural assignments. At what simulation time is each statement executed? What are the intermediate and final values of a, b, c, d?

```
initial
begin
      a = 1'b0;
      b = #10 1'b1;
      c = #5 1'b0;
      d = #20 {a, b, c};
end
```

7- Repeat exercise 3 if nonblocking procedural assignments were used.

8-What is the order of execution of statements in the following Verilog code? Is there any ambiguity in the order of execution? What are the final values of a, b, c, d?

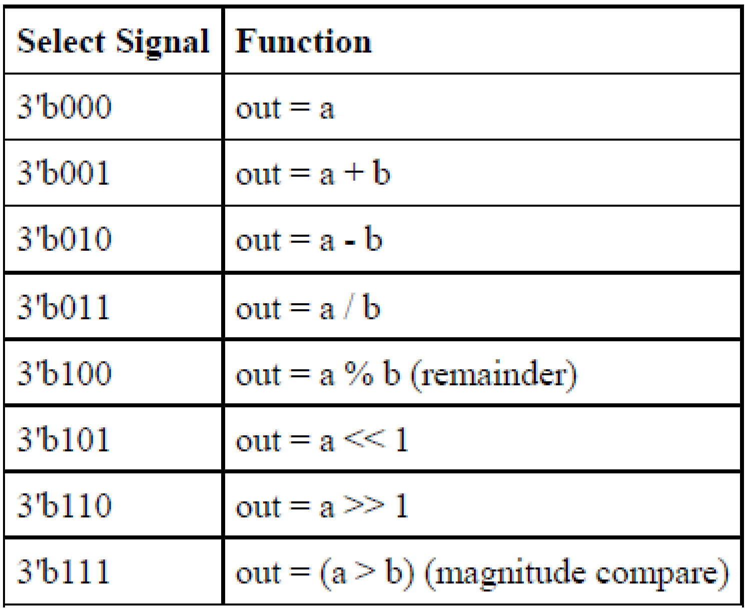
```
initial
begin
      a = 1'b0;
      #0 c = b;
end
initial
begin
      b = 1'b1;
      #0 d = a;
end
```

9-What is the final value of d in the following example? (Hint: See intraassignment delays.)

```
initial
begin
      b = 1'b1; c = 1'b0;
      #10 b = 1'b0;
initial
begin
      d = #25 (b | c);
end
```

9-Design a negative edge-triggered D-flipflop (D_FF) with synchronous clear, active high (D_FF clears only at a negative edge of clock when clear is high). Use behavioral statements only. (Hint: Output q of D_FF must be declared as reg). Design a clock with a period of 10 units and test the D_FF.

10-Using a case statement, design an 8-function ALU that takes 4-bit inputs a and b and a 3-bit input signal select, and gives a 5-bit output out. The ALU implements the following functions based on a 3-bit input signal select. Ignore any overflow or underflow bits.



Select Signal	Function
3'b000	out = a
3'b001	out = a + b
3'b010	out = a - b
3'b011	out = a / b
3'b100	out = a % b (remainder)
3'b101	out = a << 1
3'b110	out = a >> 1
3'b111	out = (a > b) (magnitude compare)

11- Design an 8-bit counter by using a forever loop, named block, and disabling of named block. The counter starts counting at count = 5 and finishes at count = 67. The count is incremented at positive edge of clock. The clock has a time period of 10. The counter counts through the loop only once and then is disabled. (Hint: Use the disable statement.)