

**CAD FOR VLSI : CS623**  
**Dept of Comp Sc and Engg, IIT Madras**  
**Mid Semester Examination**  
**Answer all Questions      Full Marks: 50**  
**Time 2.5 hours**

1. The MOS device is often abstracted as a switch in digital VLSI design. What is the operating mode of the device when abstracted as a switch?

(3 marks)

2. Given that the unit resistance of an NMOS channel is  $R$  and the capacitance is  $C$ , approximately compute the worst case gate propagation rise and fall delays of a 2-input CMOS OR gate having a fan out of  $h$  inverters. Note that the sizing of the gates is to be done so that the pull up and pull down resistances are equal.

Comment on the effect of fan out on the gate delays. Suggest a possible measure when the fan out is very high and the timing is critical (example a clock line).

(5 marks)

3. Consider the following algorithm to compute the gcd (greatest common divisor) of two integers:

gcd(in:  $u$ ,  $v$ , out:  $z$ )

```
while( $u \neq v$ ) do begin
  {
    if( $u$  and  $v$  are even)
       $z = 2 * \text{gcd}(u/2, v/2)$ ;
    else if( $u$  is odd and  $v$  is even)
       $z = \text{gcd}(u, v/2)$ ;
    else if( $u$  is even and  $v$  is odd)
       $z = \text{gcd}(u/2, v)$ ;
    else
      {
        if( $u \geq v$ )
           $z = \text{gcd}((u-v)/2, v)$ ;
        else
           $z = \text{gcd}(u, (v-u)/2)$ ;
      }
  }
}
```

Example:  $u=12, v=18$

$$\text{gcd}(12,18)=2*\text{gcd}(6,9)=2*\text{gcd}(3,9)=2*\text{gcd}(3,3)=6$$

Answer the following questions regarding to the above algorithm:

- a) What are the basic data-path elements you require to implement the above algorithm?
- b) Using the basic data-path elements write a HDL like code for the above algorithm.
- c) Write the state excitation table for the control unit of the above design. Use a hardwired approach. Is the state machine Mealy or Moore?
- d) Draw an architecture diagram for implementing the above algorithm. Note that no marks will be awarded if your architecture does not have a separate data and control paths.
- e) Write the verilog codes for the data-path and the control-path.  
(2+3+5+5+10=25 marks)

4. A designer implemented an 8-bit unsigned adder with an input from his boss that the sum is always less than 255. Hence he left out the hardware required to generate the carry out. But on the close to delivery date, his boss wanted the carry out signal. So, help the designer to design an efficient circuit to compute the final carry out. You may develop a sequential circuit, but what is important that the hardware required should be less.

(5 marks)

5. Consider the two always blocks:

(2+1+2=5 marks)

```
P1:
always@(a or b or c)
begin
  if(a==1'b1)
    d=2'b01;
  else if(b==1'b1)
    d=2'b10;
  else if(c==1'b1)
    d=2'b11;
  else
    d=2'b00;
end
```

```
P2:
always@(a or b or c)
begin
  d=2'b00;
  if(c==1'b1)
    d=2'b11;
  if(b==1'b1)
    d=2'b10;
  if(a==1'b1)
    d=2'b01;
end
```

- Are the two always blocks functionally equivalent? What do they get inferred in hardware? Just state the name, do not draw figures.
- Which of the two always blocks gets inferred to a latch?
- If the output is '10', what are the corresponding inputs in both the cases?

6. Write a synthesizable verilog code to realize a piece of hardware which shall receive 3 bit inputs and will indicate that the running sum is a multiple of 5. Ensure that the design has a synchronous reset and does not get into an undefined state. (5 marks)

7. What is the difference between the following verilog codes: (2 marks)

(a) (i) assign c = (en ?) a : b

(ii) always @(a or b) begin  
if(en) c=a;  
else c=b;  
end

(b) (i) always @(posedge clock)

```
begin
  a = #1 b;
  c = #2 a;
end
```

(ii) always @(posedge clock)

```
begin
  #1 a = b; #2 c = a;
end
```