Name: KEY **Id#**

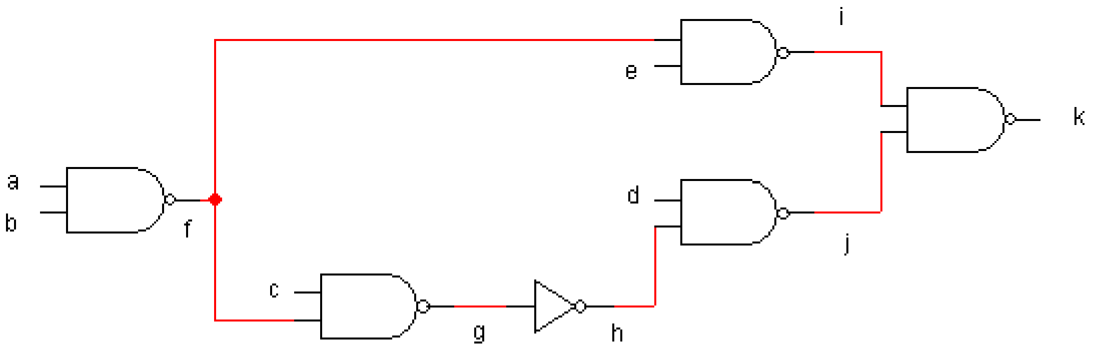
COE 405, Term 131

Design & Modeling of Digital Systems

Quiz# 4

Date: Sunday, Dec. 15, 2013

Q.1. Consider the logic network below with inputs {*a*, *b*, *c*, *d*, *e*} and output {*k*}:



Assume that the delay of the inverter gate is 1 and that the delay of the 2-input NAND gate is 2. Also, assume that the input data-ready times are zero except for input *a*, which is equal to 2.

(i) Compute the data ready times and slacks for all vertices in the network.

(i)

Data ready time	Data required time	Slack
$t_a = 2$	$\bar{t}_a = 4-2=2$	$s_a = 2-2=0$
$t_b = 0$	$\bar{t}_b = 4-2=2$	$s_b = 2-0=2$
$t_c = 0$	$\bar{t}_c = 6-2=4$	$s_c = 4-0=4$
$t_d = 0$	$\bar{t}_d = 9-2=7$	$s_d = 7-0=7$
$t_e = 0$	$\bar{t}_e = 9-2=7$	$s_e = 7-0=7$
$t_f = 2+2=4$	$\bar{t}_f = 6-2=4$	$s_f = 4-4=0$
$t_g = 4+2=6$	$\bar{t}_g = 7-1=6$	$s_g = 6-6=0$
$t_h = 6+1=7$	$\bar{t}_h = 9-2=7$	$s_h = 7-7=0$
$t_i = 4+2=6$	$\bar{t}_i = 11-2=9$	$s_i = 9-6=3$
$t_j = 7+2=9$	$\bar{t}_j = 11-2=9$	$s_j = 9-9=0$
$t_k = 9+2=11$	$\bar{t}_k = 11$	$s_k = 11-11=0$

(ii) Determine the topological critical path.

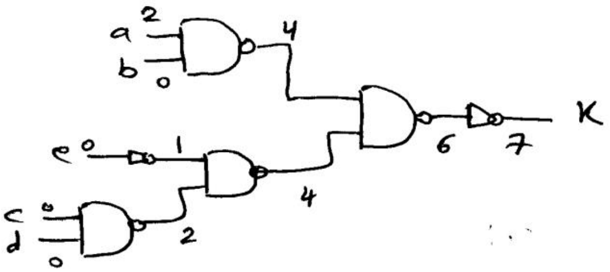
(iii) Suggest an implementation of the function *k* using only inverters and 2-input NAND gates to reduce the delay of the circuit to the minimum possible and determine the maximum propagation delay in the optimized circuit. Has the area been affected?

(ii) The topological critical path in the input/output path with zero slack which is $\{a, f, g, h, j, K\}$.

(iii) $K = fe + dh = fe + fcd = f\,(e+cd)$

$= \overline{(a\,b)}\;(e+cd)$

This leads to the following implementation:



The maximum propagation delay in the optimized circuit is 7.

The number of literals in the original circuit is 11.

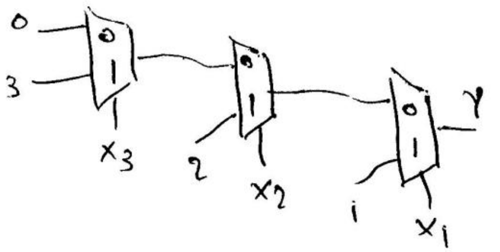
The number of literals in the optimized circuit is 6.

Thus, the area has also been reduced.

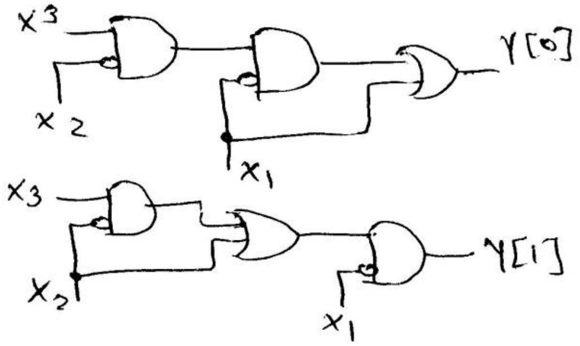
Q.2. Determine possible circuits that will be synthesized by each of the following modules:

(i)

```
module m1  (output [1:0] y, input x1, x2, x3);
        assign y = x1?1:x2?2:x3?3:0;
endmodule
```

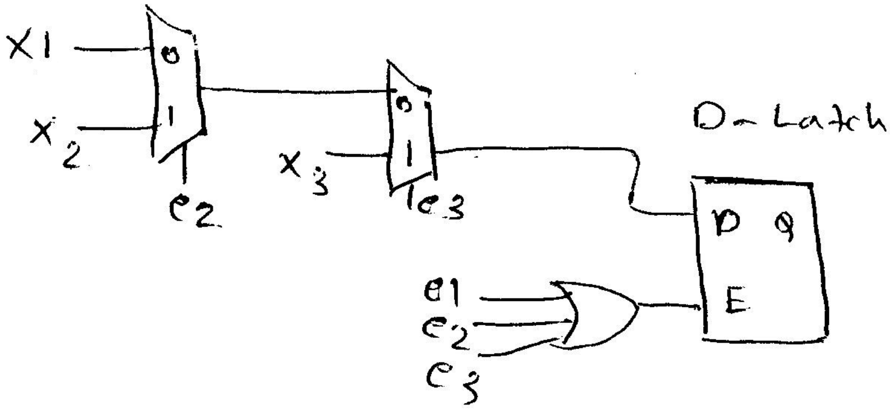


This will simplify to :



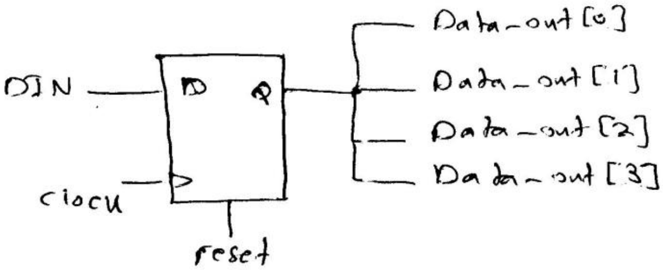
(ii)

```
module m2 (output  reg y, input e1, e2, e3, x1, x2, x3 );
always @ (e1, e2, e3, x1, x2, x3) begin
        if (e1) y=x1;
        if (e2) y=x2;
        if (e3) y=x3;
end
endmodule
```



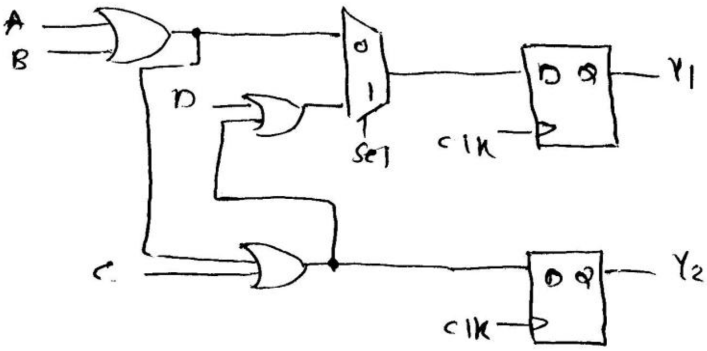
(iii)

```
module m3 (output reg [3:0] Data_out, input DIN, clock, reset);
always @ (posedge clock,  posedge reset) begin
  if (reset) Data_out = 0;
  else begin
              Data_out[0]=DIN;
              Data_out[1]=Data_out[0];
              Data_out[2]=Data_out[1];
              Data_out[3]=Data_out[2];
  end
end
endmodule
```



(iv)

```
module  m4 (output reg Y1, Y2, input  A, B, C, D, sel, clk);
   always @ (posedge clk) begin
         Y1= A | B; Y2 = Y1 | C;
         if (sel) Y1 = Y2 | D;
   end
 endmodule
```


A
B
D
0
1
Sel
D Q
Y1
clk
C
D Q
Y2
clk