COE 405, Term 162

Design & Modeling of Digital Systems

Assignment# 5 Solution

Due date: Saturday, May 13

Q.1. Consider the following function:

X=AC+BC+AD+BD+A'B'C'+A'B'D'+AEF+BEF+AE'F'+BE'F'+A'B'EF'+ A'B'E'F

(i) Compute all double-cube divisors of *X* along with their bases and their weights. Show only double-cube divisors that have non-empty bases.

(i) Double Cube Divisors

Double-Cube divisor	Base	weight
$A+B$	$C, D, EF, \bar{E}\bar{F}$	$4\times2-4-2+6+4=12$
$C+D$	A, B	$2\times2-2-2+2+0=2$
$C+EF$	A, B	$2\times3-2-3+2=3$
$C+\bar{E}\bar{F}$	A, B	$2\times3-2-3+2=3$
$D+EF$	A, B	$2\times3-2-3+2=3$
$D+\bar{E}\bar{F}$	A, B	$2\times3-2-3+2=3$
$\bar{C}+\bar{D}$	$\bar{A}\bar{B}$	$1\times2-1-2+2+0=1$
$\bar{C}+E\bar{F}$	$\bar{A}\bar{B}$	$1\times3-1-3+2=1$
$\bar{C}+\bar{E}F$	$\bar{A}\bar{B}$	$1\times3-1-3+2=1$
$\bar{D}+E\bar{F}$	$\bar{A}\bar{B}$	$1\times3-1-3+2=1$
$\bar{D}+\bar{E}F$	$\bar{A}\bar{B}$	$1\times3-1-3+2=1$
$EF+\bar{E}\bar{F}$	A, B	$3\times4-3-4+4=9$
$AF+\bar{A}\bar{B}\bar{F}$	E	$1\times5-1-5+1=0$
$AE+\bar{A}\bar{B}\bar{E}$	F	$1\times5-1-5+1=0$
$BF+\bar{A}\bar{B}\bar{F}$	E	$1\times5-1-5+1=0$
$BE+\bar{A}\bar{B}\bar{E}$	F	$1\times5-1-5+1=0$
$A\bar{E}+\bar{A}\bar{B}E$	$\bar{F}$	$1\times5-1-5+1=0$
$A\bar{F}+\bar{A}\bar{B}F$	$\bar{E}$	$1\times5-1-5+1=0$
$B\bar{E}+\bar{A}\bar{B}E$	$\bar{F}$	$1\times5-1-5+1=0$
$B\bar{F}+\bar{A}\bar{B}F$	$\bar{E}$	$1\times5-1-5+1=0$
$E\bar{F}+\bar{E}F$	$\bar{A}\bar{B}$	$3\times4-3-4+4=9$

(ii) Apply the fast extraction algorithm based on extracting double-cube divisors along with complements or single-cube divisors with two-literals. Show all steps of the algorithm. Determine the number of literals saved. Compare your solution with the result obtained by running the sis commands ***fx***.

(ii) Fast Extraction

From part (i), we can see that $W_{dmax} = 12$ and the double cube divisor $A+B$ has the highest weight.

The highest weight single cube divisor is $\bar{A}\bar{B}$ with a weight of 2.

Thus, the double cube divisor is selected and the resulting network is:

$$[1] = A + B$$

$$X = C[1] + D[1] + \bar{C}\overline{[1]} + \bar{D}\overline{[1]} + EF[1] + \bar{E}\bar{F}[1] + E\bar{F}\overline{[1]} + \bar{E}F\overline{[1]}$$

22 literals

Double cube dividers are updated in the same way and the double cube divider $EF + \bar{E}\bar{F}$ is extracted as it has the highest weight of $2\times4-2-4+2=4$.

Thus, the resulting network will be:

$$[1] = A + B$$

$$[2] = EF + \bar{E}\bar{F}$$

$$X = C[1] + D[1] + \bar{C}\overline{[1]} + \bar{D}\overline{[1]} + [2][1] + \overline{[2]}\overline{[1]}$$

18 literals

Since none of the remaining double cube divisors or single cube divisors has a positive weight, none of them will be extracted.

SIS has resulted in the same extracted network as shown below:

```
sis> read_eqn hw5q1.eqn
sis> print
    {X} = A C + A D + A E F + A E' F' + A' B' C' + A' B' D' + A' B' E F' + A' B
' E' F + B C + B D + B E F + B E' F'
sis> print_stats
hw5q1.eqn  pi= 6  po= 1  node=  1  latch= 0  lits(sop)=  34  lits(ff)=  16
sis> fx
sis> print
    {X} = C [1] + C' [1]' + D [1] + D' [1]' + [1] [2] + [1]' [2]'
    [1] = A + B
    [2] = E F + E' F'
sis> print_stats
hw5q1.eqn  pi= 6  po= 1  node=  3  latch= 0  lits(sop)=  18  lits(ff)=  14
sis>
```

Q.2. Consider the logic network defined by the following expressions:

e=a b d

f =c d

g = e + f

h = a d

i =a’ b’ d’

j = h + i

k= b d

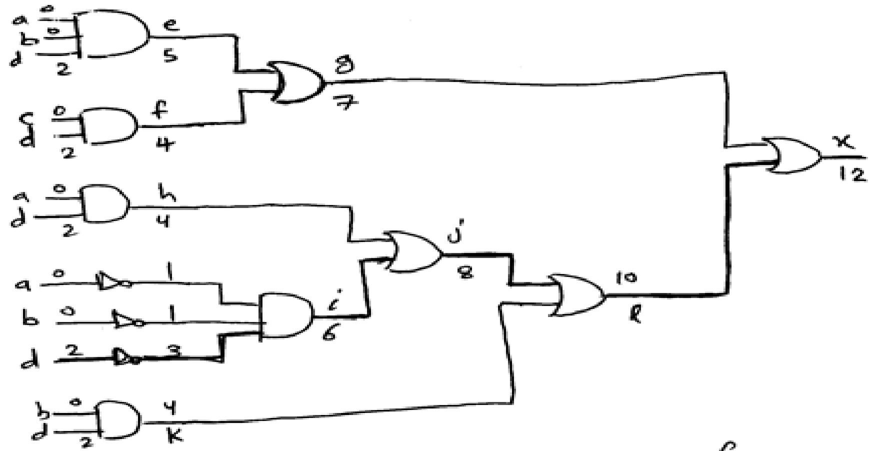
l =j + k

x =g + l

Inputs are {a, b, c, d} and output is {x}. Assume that the delay of a gate is related to the number of its inputs. Also, assume that the input data-ready times are zero except for input d, which is equal to 2.

(i) Draw the logic network graph and compute the data ready times and slacks for all vertices in the network.

(i)



The data ready times are shown on the figure. The maximum propagation delay is 12. To compute the slack for each node, the required time for x is set to 12.

$\bar{t}_x = 12$ $S_x = 12-12=0$

$\bar{t}_g = 12-2 = 10$ $S_g = 10-7=3$

$\bar{t}_l = 12-2=10$ $S_l = 10-10=0$

$\bar{t}_e = 10-2=8$ $S_e = 8-5=3$

$\bar{t}_f = 10-2=8$ $S_f = 8-4=4$

$\bar{t}_j = 10-2=8$ $S_j = 8-8=0$

$\bar{t}_k = 10-2=8$ $S_k = 8-4=4$

$\bar{t}_h = 8-2=6$ $S_h = 6-4=2$

$\bar{t}_i = 8-2=6$ $S_i = 6-6=0$

$\bar{t}_a = \min\{8-3, 6-2, 6-3-1\} = 2$ $S_a = 2-0=2$

$\bar{t}_b = \min\{8-3, 6-3-1, 8-2\} = 2$ $S_b = 2-0=2$

$\bar{t}_c = 8-2=6$ $S_c = 6-0=6$

$\bar{t}_d = \min\{8-3, 8-2, 6-2, 6-3-1, 8-2\} = 2$ $S_d = 2-2=0$

(ii) Determine the **maximum propagation delay** and the **topological critical path**.

(ii) The maximum propagation delay is 12 and the topological critical path is : $\{d, i, j, l, x\}$.

(iii) Suggest an implementation of the function ***x*** to reduce the delay of the circuit. What is the **maximum propagation delay** after the modified implementation?

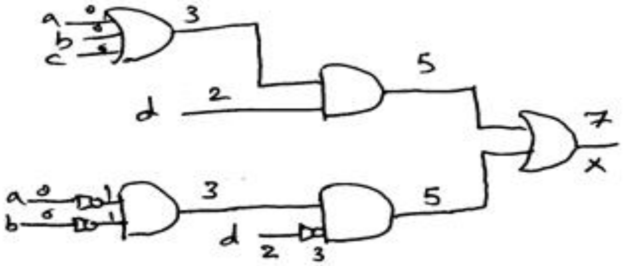
(iii) $x = g + l = e + f + j + k$

$= abd + cd + ad + \bar{a}\bar{b}\bar{d} + bd$

$= d\,[ab + c + a + b] + \bar{a}\bar{b}\bar{d}$

$= d\,[a + b + c] + \bar{a}\bar{b}\bar{d}$

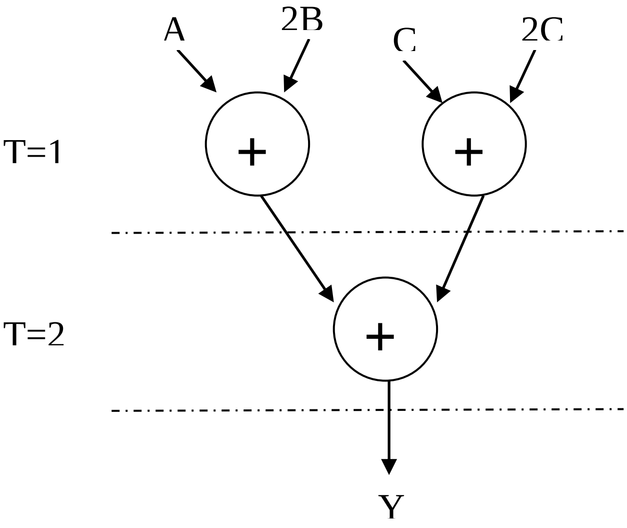
Thus, to improve the delay of x, we can implement it as follows:



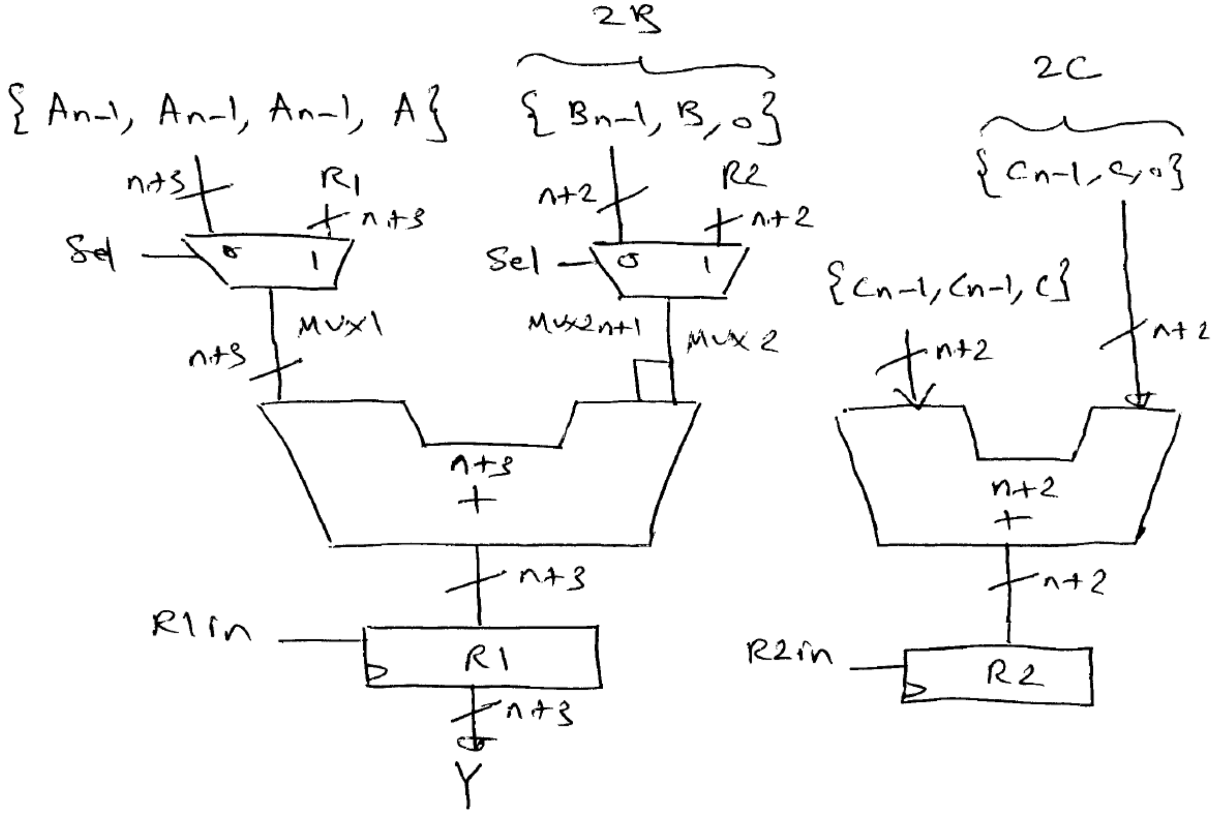
Thus, the delay of x is reduced from 12 to 7. At the same time, the area of the circuit was improved.

Q.3. It is required to design a circuit to compute the equation Y=A+2*B+3*C, where A, B, and C are N-bit inputs representing signed numbers in 2's complement representation. Assume that inputs are available only during the first cycle when a **START** input is asserted. Assume that a **DONE** signal will be set when the result is ready and the result will remain held until the next Start operation.

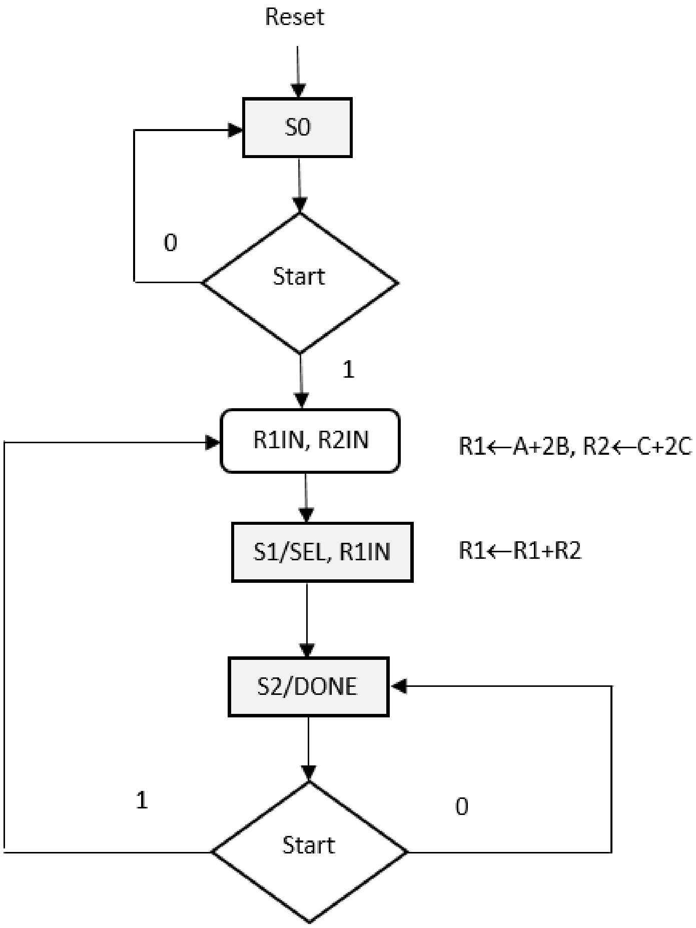
(i) Show a schedule of the operations with minimum latency (i.e., clock cycles) assuming that the clock cycle is limited by the time for performing one addition operation. Store the output Y in a register.



(ii) Show the Data Path design of your circuit indicating all the control signals and the used adder sizes.



(iii) Show the ASMD diagram of your control unit.

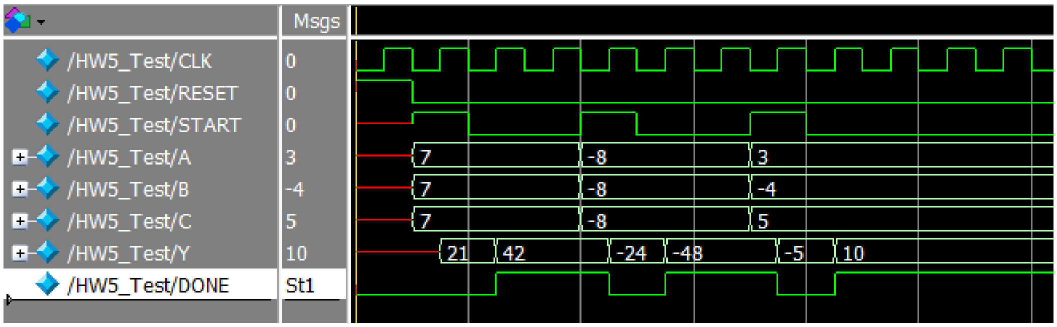


(iv) Write the necessary Verilog modules to module the data path unit, control unit and the overall circuit.

```
module HW5 #(parameter N=4) (output [N+2:0] Y, output DONE,
input [N-1:0] A, B, C, input START, RESET, CLK);


HW5_DPU #(N) M1 ( Y, A, B, C, SEL, R1IN, R2IN, CLK);

HW5_CU M2 (SEL, R1IN, R2IN, DONE, START, RESET, CLK);


endmodule


module HW5_DPU #(parameter N=4) (output [N+2:0] Y, input [N-
1:0] A, B, C, input SEL, R1IN, R2IN,  CLK);


reg [N+2:0] R1;
reg [N+1:0] R2;
wire [N+2:0] MUX1;
wire [N+1:0] MUX2;

assign MUX1 = SEL? R1 : {A[N-1],A[N-1],A[N-1],A};
assign MUX2 = SEL? R2 : {B[N-1],B,1'b0};

assign Y = R1;


always @(posedge CLK) begin
if (R1IN)
  R1 = MUX1 + {MUX2[N+1],MUX2} ;

if (R2IN)
  R2 = {C[N-1],C[N-1],C} + {C[N-1],C,1'b0};

end


endmodule;


module HW5_CU (output reg SEL, R1IN, R2IN, DONE, input START,
RESET, CLK);

parameter S0 = 2'b00, S1=2'b01, S2=2'b10;

reg [1:0] state, next_state;

always @(posedge CLK, posedge RESET)
   if (RESET) state <= S0;
   else state <= next_state;

always @(state, START) begin
   SEL=0; R1IN=0; R2IN=0; DONE=0;
   case (state)
   S0:
        if (START)  begin
           next_state=S1; R1IN=1; R2IN=1; end
        else next_state=S0;
   S1: begin SEL=1; R1IN=1; next_state=S2; end
   S2: begin DONE=1;
        if (START)  begin
           next_state=S1; R1IN=1; R2IN=1; end
        else next_state=S2;
        end
       default: begin
        next_state=1'bx;
        SEL=1'bx; R1IN=1'bx; R2IN=1'bx; DONE=1'bx;
       end


   endcase
end

endmodule
```

(v) Write a test bench to test the correct operation of your circuit. Include simulation snapshots.

```
module HW5_Test();



wire [6:0] Y;
wire DONE;
reg  [3:0] A, B, C;
reg  START, RESET, CLK;


HW5 #(4) M1 ( Y,  DONE, A, B, C, START, RESET, CLK);

initial begin
CLK = 0;
forever
#50 CLK = ~ CLK;
end

initial begin
RESET=1;
#100 RESET=0; START=1; A=7; B=7; C=7;
#100 START=0;
#100 ;
#100 START=1; A=-8; B=-8; C=-8;
#100 START=0;
#100 ;
#100 START=1; A=3; B=-4; C=5;
#100 START=0;

end


endmodule
```



(vi) Implement your circuit on FPGA assuming N=2 bits. Include a link for a video snapshot to demonstrate correct functionality of your circuit.