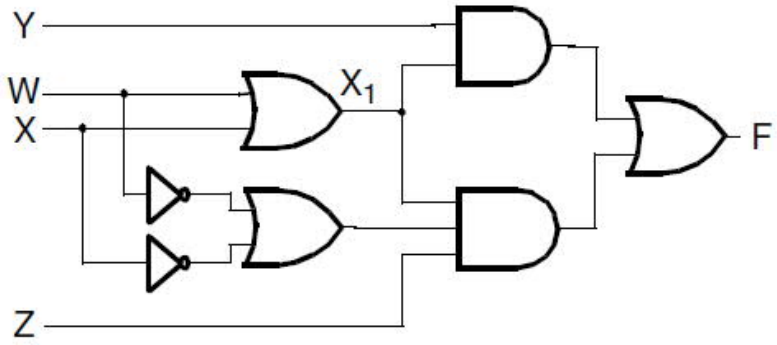
Đáp án của Bài tập ôn kiểm tra giữa học kỳ

1. Với mạch tổ hợp sau:



Hãy viết mã Verilog với các cách sau:

1) Dùng mô hình cấu trúc.
2) Dùng mô hình luồng dữ liệu.
3) Dùng mô hình hành vi.

BG.

1) Mô hình cấu trúc:

Đặt tên các cổng từ trên xuống dưới và từ trái qua phải.

```
module BT1_Q2s (Y, W, X, Z, F);
input   Y, W, X, Z;
output  F;

wire W_n, X_n;
wire T1, T2, T3, T4;

not u1(W_n, W);
not u2(X_n, X);
or u3(T1, W, X);
or u4(T2, W_n, X_n);
and u5(T3, Y, T1);
and u6(T4, T1, T2, Z);
or u7(F, T3, T4);

endmodule
```

2) Mô hình luồng dữ liệu

```
module BT1_Q2d (Y, W, X, Z, F);
input   Y, W, X, Z;
output  F;

wire W_n, X_n;
wire T1, T2, T3, T4;

assign T1 = W | X;
assign T2 = ~W | ~X;
assign T3 = Y & T1;
assign T4 = T1 & T2 & Z;
assign F = T3 | T4;

endmodule
```

3) Mô hình hành vi

F = Y (W + X) + (W+X)(W'+X').Z = Y(W + X) + Z(W ⊕ X) =

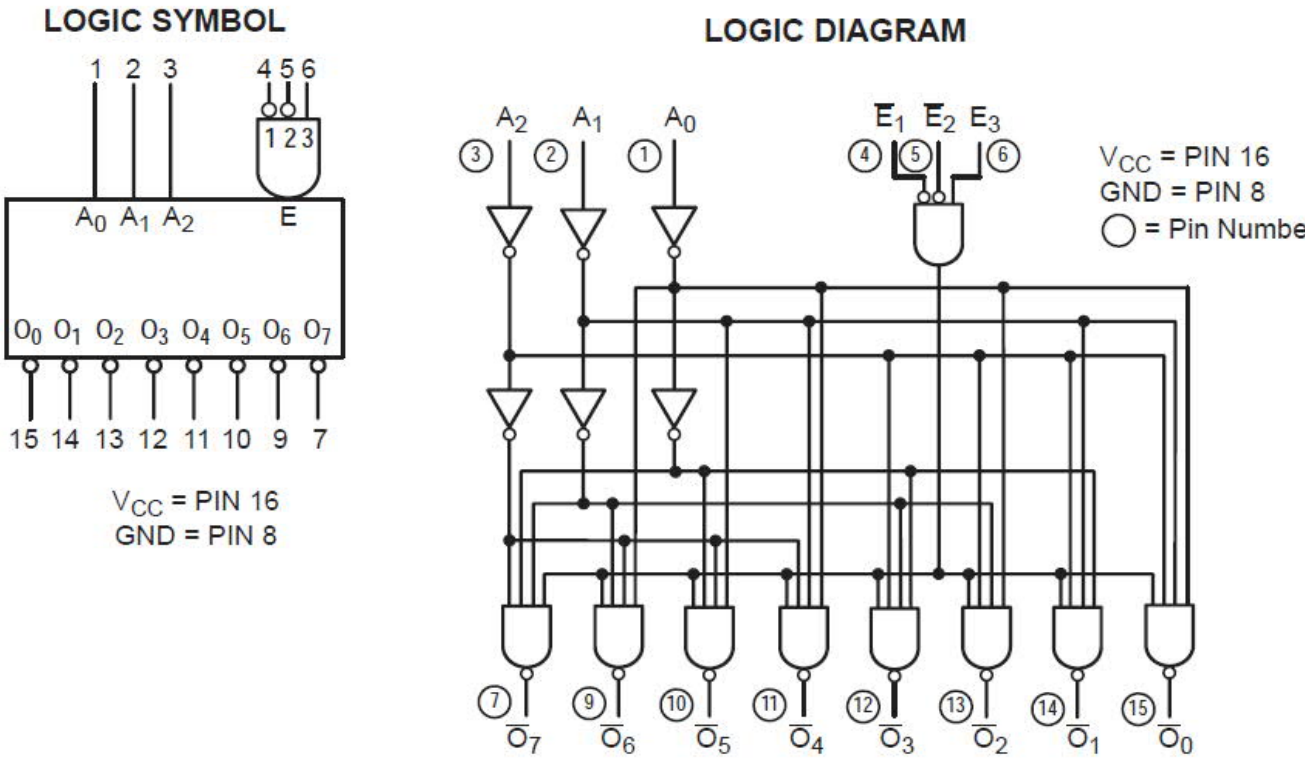
W = 0 ⇒ F = YX + ZX = XYZ' + XYZ + XY'Z + XYZ = Σm(5, 6, 7)

W = 1 ⇒ F = Y + ZX' = X'YZ' + X'YZ + XYZ' + XYZ + X'Y'Z + X'YZ = Σm(1, 2, 3, 6, 7)

⇒ F (W, X, Y, Z) = Σm(3, 5, 7, 9, 10, 11. 14, 15)

```
module BT1_Q2h (Y, W, X, Z, F);
input   Y, W, X, Z;
output  F;
reg  F;

wire [3:0] data;

assign data = {W, X, Y, Z};

always @ (data)
        case (data)
                5,6,7,9,10,11,14,15: F =1;
                default: F = 0;
        endcase
endmodule
```

2. Thiết kế mạch giải mã 74138 (đặt tên biến có bù bằng cáh thêm "_n" hay "_bar" phía sau, TD: ngõ ra $\overline{O7}$ có thể dùng danh hiệu **Q7_n** hay **Q7_bar**, dùng Q vì O dễ nhầm số 0).



Hãy viết mã Verilog với các cách sau:

1) Dùng mô hình cấu trúc.
2) Dùng mô hình luồng dữ liệu.
3) Dùng mô hình hành vi.

BG.

1) Mô hình cấu trúc

```
module dec138s (
                A0, A1, A2, E1_n, E2_n, E3, // inputs
                Q0_n, Q1_n, Q2_n, Q3_n, Q4_n, Q5_n, Q6_n, Q7_n // outputs
);
```

```
input  A0, A1, A2, E1_n, E2_n, E3;
output Q0_n, Q1_n, Q2_n, Q3_n, Q4_n, Q5_n, Q6_n, Q7_n;

wire A0_n, A1_n, A2_n;
wire EN, E1, E2;

not(A0_n, A0);
not(A1_n, A1);
not(A2_n, A2);
not(E1, E1_n);
not(E2, E2_n);

and(EN, E1, E2, E3);
nand(Q0_n, EN, A2_n,  A1_n, A0_n);
nand(Q1_n, EN, A2_n,  A1_n, A0);
nand(Q2_n, EN, A2_n,  A1, A0_n);
nand(Q3_n, EN, A2_n,  A1, A0);
nand(Q4_n, EN, A2,  A1_n, A0_n);
nand(Q5_n, EN, A2,  A1_n, A0);
nand(Q6_n, EN, A2_n,  A1, A0);
nand(Q7_n, EN, A2,  A1, A0);

endmodule
```

2) Mô hình luồng dữ liệu

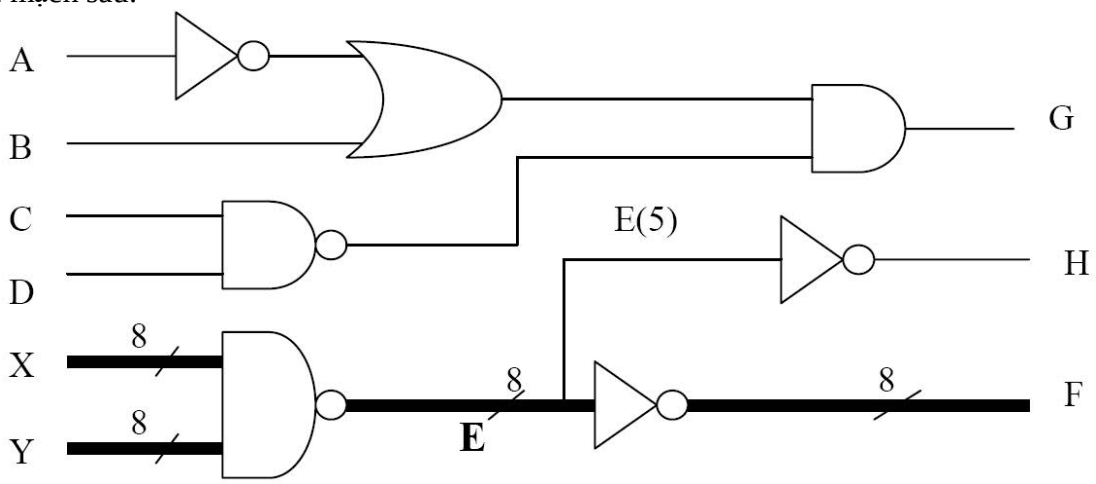
```
module dec138d (
                        A0, A1, A2, E1_n, E2_n, E3, // inputs
                        Q0_n, Q1_n, Q2_n, Q3_n, Q4_n, Q5_n, Q6_n, Q7_n // outputs
);
input  A0, A1, A2, E1_n, E2_n, E3;
output Q0_n, Q1_n, Q2_n, Q3_n, Q4_n, Q5_n, Q6_n, Q7_n;

wire EN_n;

assign EN_n = E1_n | E2_n | ~E3;

assign Q0_n = EN_n |  A2 |  A1  |  A0;
assign Q1_n = EN_n |  A2 |  A1  | ~A0;
assign Q2_n = EN_n |  A2 | ~A1  |  A0;
assign Q3_n = EN_n |  A2 | ~A1  |  ~A0;
assign Q4_n = EN_n | ~A2 |  A1  |  A0;
assign Q5_n = EN_n | ~A2 |  A1  |  ~A0;
assign Q6_n = EN_n | ~A2 | ~A1  |  A0;
assign Q7_n = EN_n | ~A2 | ~A1  | ~A0;

endmodule
```

3) ***Mô hình hành vi***

```
module dec138h (
                        A0, A1, A2, E1_n, E2_n, E3, // inputs
                        Q0_n, Q1_n, Q2_n, Q3_n, Q4_n, Q5_n, Q6_n, Q7_n // outputs
);
input  A0, A1, A2, E1_n, E2_n, E3;
output Q0_n, Q1_n, Q2_n, Q3_n, Q4_n, Q5_n, Q6_n, Q7_n;

reg [0:7] out;
```

```
wire [2:0] A, E;
integer k;

assign {Q0_n, Q1_n, Q2_n, Q3_n, Q4_n, Q5_n, Q6_n, Q7_n} = out;
assign A = {A2, A1, A0};
assign E = {E3, E2_n, E1_n};

always @ (A or E)
        begin
                out = 8'b1111_1111;
                if (E == 3'b100)
                        for ( k = 0;  k < 8; k = k +1)
                                if (k == A)
                                        out[k] = 0;
                        end
endmodule
```

3. Cho trước mạch sau:



Hãy viết mã Verilog mô tả mạch này với các cách sau:

1) Dùng mô hình cấu trúc.
2) Dùng mô hình luồng dữ liệu.
3) Dùng mô hình hành vi.

BG.

1) Mô hình cấu trúc

```
// Đánh số cổng từ trên xuống dưới và từ trái qua phải
module BT2_Q1s (A, B, C, D, X, Y, G, H, F);
input A, B, C, D; input [7:0] X, Y;
output G, H; output [7:0] F;

wire An, T1, T2;
wire [7:0] E;
integer k;

not (An, A);
or (T1, An, B);
nand (T2, C, D);
and (G, T1, T2);
nand (E[0], X[0], Y[0]);
nand (E[1], X[1], Y[1]);
nand (E[2], X[2], Y[2]);
nand (E[3], X[3], Y[3]);
nand (E[4], X[4], Y[4]);
nand (E[5], X[5], Y[5]);
```

```
nand (E[6], X[6], Y[6]);
nand (E[7], X[7], Y[7]);
not(H, E[5]);
not(F[0], E[0]);
not(F[1], E[1]);
not(F[2], E[2]);
not(F[3], E[3]);
not(F[4], E[4]);
not(F[5], E[5]);
not(F[6], E[6]);
not(F[7], E[7]);

endmodule
```

Chú ý: Có thể viết gọn hơn bằng cách tự tạo các cổng mong muốn:

```
// Main program
module BT2_Q1s2 (A, B, C, D, X, Y, G, H, F);
input A, B, C, D; input [7:0] X, Y;
output G, H; output [7:0] F;

wire An, T1, T2;
wire [7:0] E;

not u1(An, A);
or u2(T1, An, B);
nand u3(T2, C, D);
and u4(G, T1, T2);
not  u5(H, E[5]);
mynot u6(.out(F), .in(E));
mynand u7(.out(E), .in1(X), .in2(Y));

endmodule

// NAND 8-bit data
module mynand(out,in1, in2);
input [7:0] in1, in2;
output [7:0] out;

assign out = ~(in1 & in2);

endmodule

// NOT 8-bit data
module mynot(out,in);
input [7:0] in;
output [7:0] out;

assign out = ~in;
endmodule
```

2) Mô hình luồng dữ liệu

```
module BT2_Q1d (A, B, C, D, X, Y, G, H, F);
input A, B, C, D; input [7:0] X, Y;
output G, H; output [7:0] F;
wire [7:0] E;
```

```
assign   G = (~A | B) & ~ (C & D);
assign   E = ~(X & Y);
assign   H = ~E[5];
assign   F = ~E;
endmodule
```

3) Mô hình hành vi

```
module BT2_Q1h (A, B, C, D, X, Y, G, H, F);
input A, B, C, D; input [7:0] X, Y;
output G, H; output [7:0] F;
reg G, H; reg [7:0] F;
reg [7:0] E;
wire [19:0] data;

assign data = {A, B, C, D, X, Y};

always@(data)
begin
        if (!(C && D))
                G = ~A | B;
        else
                G = 0;
        E = ~(X & Y);
        H = ~E[5];
        F = ~E;
end
endmodule
```

4. Cho trước mạch tổ hợp có hàm Boole sau:

$$F(A, B, C, D) = \bar{C}\bar{D} + \bar{A}CD + BD$$

Viết mã Verilog mô tả mạch dùng

a) mô hình luồng dữ liệu.
b) mô hình hành vi với phát biểu **if . . . else . . . .**
c) mô hình hành vi với phát biểu **case**

Chú ý: Không cần viết lại các khai báo cổng cho các mô hình.

BG.

```
module cau_1(A, B, C, D, F);
input A, B, C, D;    output F;
```

a) Mô hình luồng dữ liệu

```
        assign    F = (~C & ~D) | (~A & C & D) | (B & D);
```

b) Mô hình hành vi với if … else …
Ta có: F = D'C' + D(A'C + B)

```
        reg F;

        always@(A or B or C or D)

                if  (D == 1)  // hoặc dùng:  if   (D)

                        F = (~A & C ) | B;

                else

                        F = ~C;
```

c) Mô hình hành vi với case

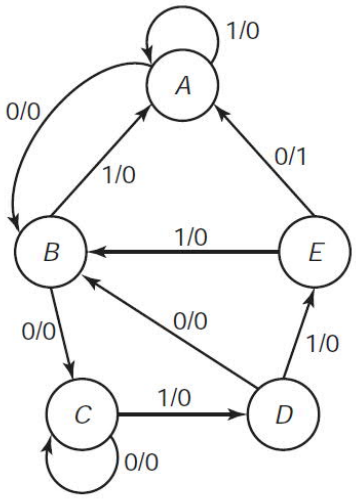
F(A,B, C,D) = C'D' + A'CD + BD = Σm(0,3,4,5,7,8,12,13,15) = ΠM(1,2,6,9,10,11,14)

```
        reg F;

        always@(A or B or C or D)

        case({A, B, C, D})

                1,2,6,9,10,11,14: F = 0;

                default: F = 1;

        endcase;

// hoặc theo A, B
```

AB \ CD	00	01	11	10
00	1		1	
01	1	1	1	
11	1	1	1	
10	1			

```
endmodule
```

5. Dùng Verilog mô tả FSM có giản đồ trạng thái hình bên.Giả sử hệ có xung nhịp **clk** kích cạnh lên, ngõ vào **X**, ngõ ra **Y**, và ngõ điều khiển đồng bộ **Reset_n** tích cực thấp (Khi hệ bị Reset sẽ về trạng thái **A**).



BG.

```
module cau_2(clk, Reset_n, X, Y);
input clk, Reset, X;
output Y;  reg Y;
reg [1:0] state, nstate; // nstate = next state
parameter  A = 3'b000, B = 3'b001, C = 3'b010;
parameter D = 3'b011, E =3'b100;

// State register
always@(posedge clk)
        if (!Reset_n)
                state <= A;
        else
                state <= nstate;

// Next state Logic
always@(state or x)
        case(state)
                 A: nstate = X? A : B;
                 B: nstate = X? A : C;
                 C: nstate = X? D : C;
                 D: nstate = X? E : B;
                 E: nstate = X? B : A;
                 default: nstate = A;
        endcase

// Output Logic
     assign Y = (state == E) && (X == 0) ;
endmodule
```

6. Thiết kế mạch đếm lên có modulo chỉnh được (từ 1 đến 15), ngõ vào N dùng để chỉnh modulo.

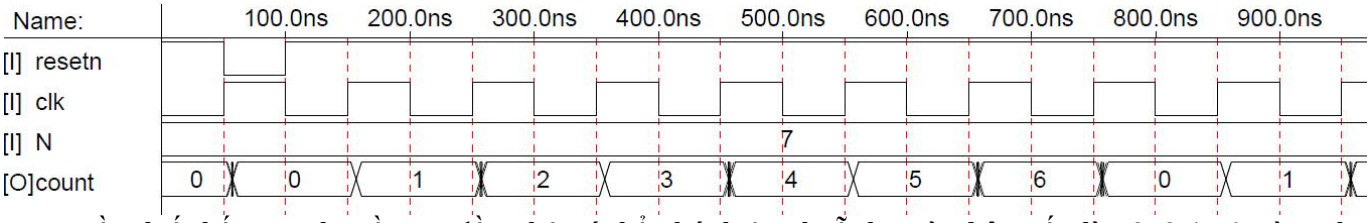
Thí dụ: N=5 thì ngõ ra Q có chuỗi đếm 0000, 0001, 0010, 0011, 0100, 0000.

Bộ đếm này dùng xung nhịp CLK tác động cạnh lên và có ngõ reset bất đồng bộ tích cực thấp.

BG.

```
module BT2_Q2(resetn, clk, N, count);
input resetn, clk; input [7:0] N;
output [7:0] count;
reg [7:0] count;

always@(posedge clk or negedge resetn)
begin
        if (!resetn)
                count <= 0;
        else
                if (count ==  N - 1)
                        count <= 0;
                else
                        count <= count + 1;
end
endmodule
```

>> Dạng sóng mô phỏng với N = 7:



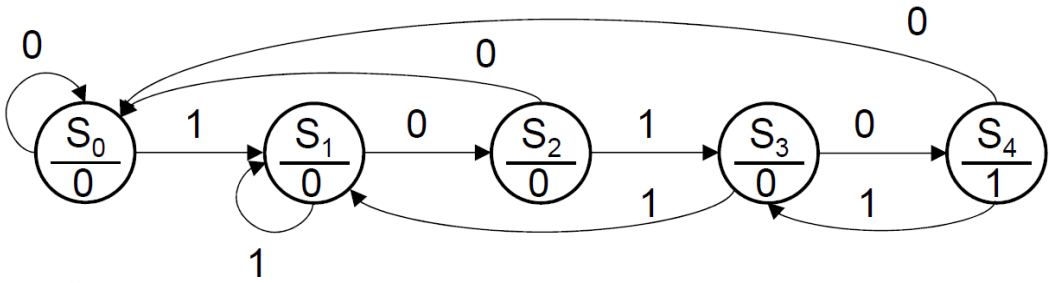
7. Ta cần thiết kế 1 mạch tuần tự đồng bộ có thể phát hiện chuỗi bit vào liên tiếp là 1010 (ngõ vào 1 bit nối tiếp) và khi có phát hiện thì ngõ ra sẽ là 1. Hãy viết mã Verilog với

a) Máy trạng thái Moore.
b) Máy trạng thái Mealy.

BG.

a) Máy trạng thái Moore

Giản đồ trạng thái

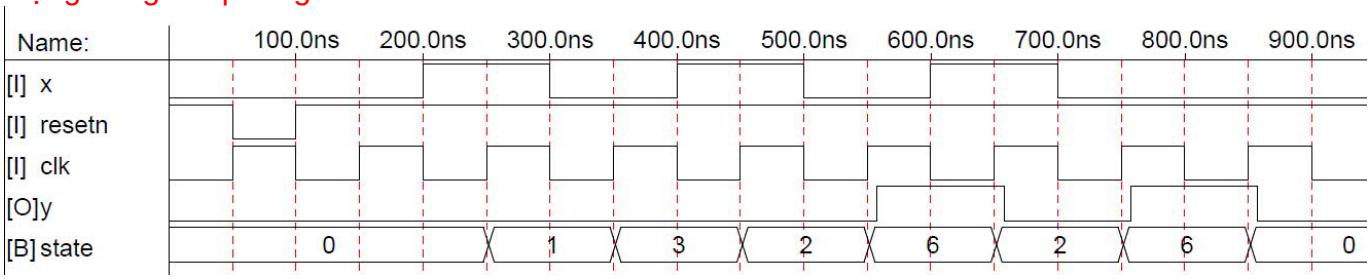


Chương trình Verilog:

```
module BT2_Q3a (x, clk, resetn, y);
input x, clk, resetn;
output y;
reg [2:0] state;
parameter S0='b000, S1='b001, S2='b011, S3='b010, S4 = 'b110; // 3-bit Gray code

// Define the sequential block
always @ (posedge clk or negedge resetn)
if (!resetn)
        state <= S0;
else
        case (state)
                S0:  state <= x ? S1 : S0;
                S1:  state <= x ? S1 : S2;
                S2:  state <= x ? S3 : S0;
                S3:  state <= x ? S1 : S4;
```

```
            S4:  state <= x ? S3 : S0;
        endcase

// Define output during S4
assign   y = (state == S4);
endmodule
```

Dạng sóng mô phỏng:

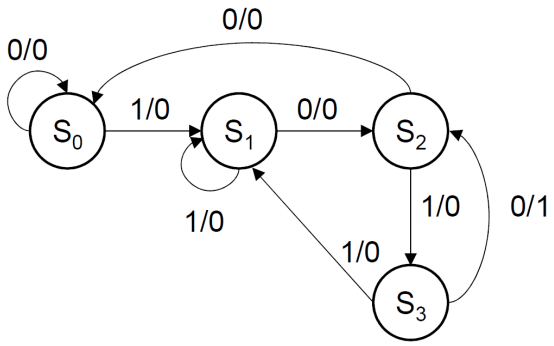


Chú ý:

Cách 2: dùng thanh ghi dịch chuyển 4 bit nối tiếp thành song song và so sánh với chuỗi “1010”.

b) Máy trạng thái Mealy

Giản đồ trạng thái

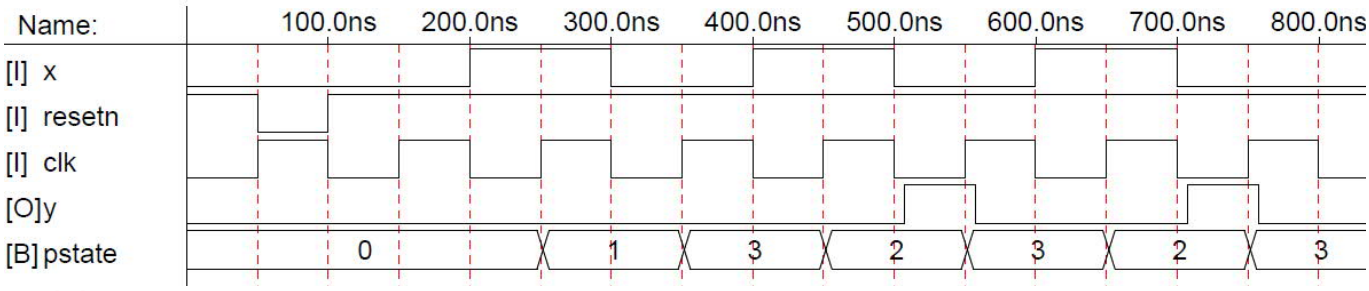


Chương trình:

```
module BT2_Q3b (x,clk, resetn, y);
// Mealy machine for a "1010" sequence detection
input x, clk, resetn;
output y;
reg y;
reg [1:0] pstate, nstate; //present and next states
parameter S0=2'b00, S1=2'b01, S2=2'b11, S3=2'b10;

// Next state and output combinational logic
// Use blocking assignments "="
always @(x or pstate)
case (pstate)
        S0: if (x) begin nstate = S1; y = 0; end
                else begin nstate = S0; y = 0; end
        S1: if (x) begin nstate = S1; y = 0; end
                else begin nstate = S2; y = 0; end
        S2: if (x) begin nstate = S3; y = 0; end
                else begin nstate = S0; y = 0; end
        S3: if (x) begin nstate = S1; y = 0; end
                else begin nstate = S2; y = 1; end
endcase

// Sequential logic, use nonblocking assignments "<="
always @(posedge clk or negedge resetn)
        if (!resetn)
```

```
            pstate <= S0;
        else
            pstate <= nstate;
endmodule
```

Dạng sóng mô phỏng:

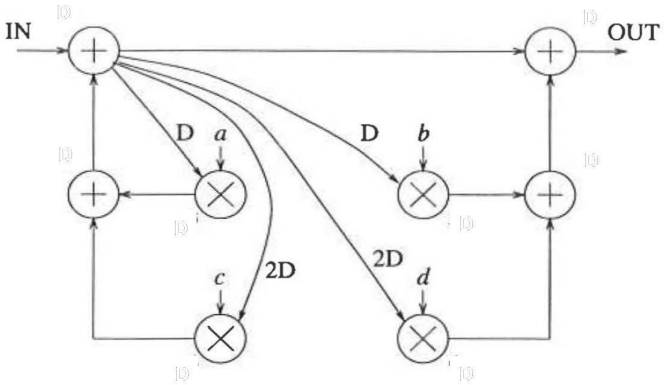


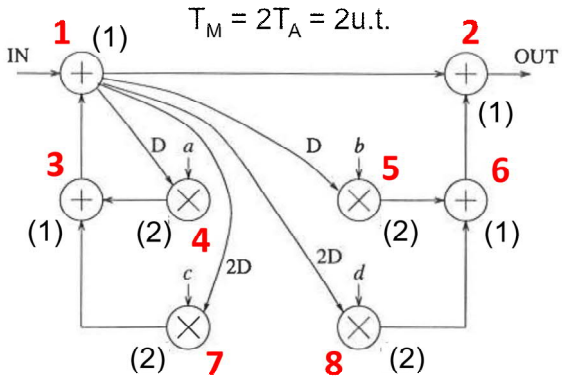
Chú ý:

Có thể viết gọn hơn chỗ lệnh always tính trạng thái kế như sau:

```
always @ (x or pstate)
begin
    if ((pstate == S3) && (x == 0))  y = 1;  else  y = 0;
    case (pstate)
        S0: nstate = x ? S1 : S0;
        S1: nstate = x ? S1 : S2;
        S2: nstate = x ? S3 : S0;
        S3: nstate = x ? S1 : S2;
    endcase
end
```

8. Hãy tìm đường tới hạn $T_{critical}$ cho hệ sau, biết T_M =2 và T_A = 1 u.t.



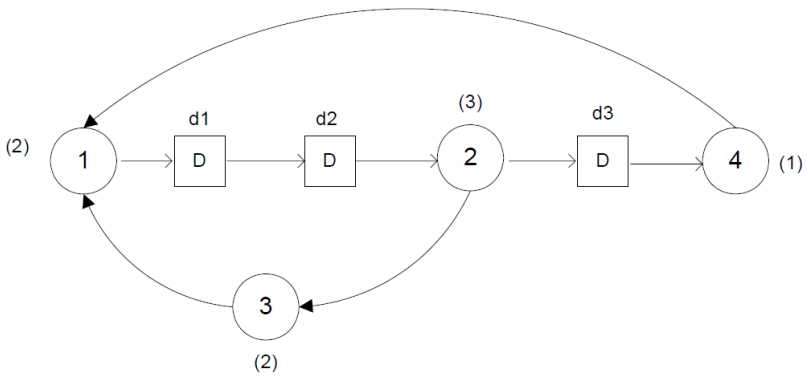


ĐS.

Các nút được đánh số màu đỏ, các đường tới hạn qua các nút 4-3-1-2 hoặc 7-3-1-2 hoặc , do đó ta tìm được:

$$T_{critical} = T_M + T_A + T_A + T_A = T_M + 3T_A = 2 + 3 \times 1 = 5 \text{ u.t.}$$

9. (Ch2. Prob 1) Với DFG trong hình 2.12, thời gian tính toán của nút được cho trong dấu ngoặc. Tính giới hạn lặp của DFG này bằng quan sát và giải thuật LPM.



Hình 2.12

BG.

a) Phương pháp quan sát:

L1: 1-2-3 có giới hạn vòng là (2+3+2)/2 = 3.5 u.t.

L2: 1-2-4 có giới hạn vòng là (2+3+1)/2 = 3 u.t.

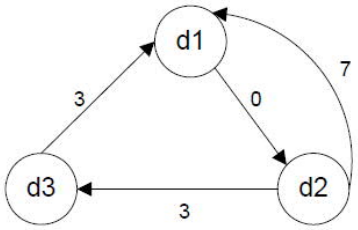
Như vậy giới hạn lặp T_∞ = max(gh vòng) = 3.5 u.t.

b) Giải thuật LPM

Theo các bước sau:

***S1.* Xây dựng giản đồ delay G_d từ DFG**

$$d1 \to d2 : 0$$
$$d2 \to d1 : d2 \to 2 \to 3 \to 1 \to d1 = 7$$
$$d2 \to d3 : d2 \to d3 = 3$$
$$d3 \to d1 : d3 \to 4 \to 1 \to d1 = 3$$



G_d

***S2.* Xây dựng ma trận L(1)**

$$L^{(1)} = \begin{bmatrix} -1 & 0 & -1 \\ 7 & -1 & 3 \\ 3 & -1 & -1 \end{bmatrix}$$

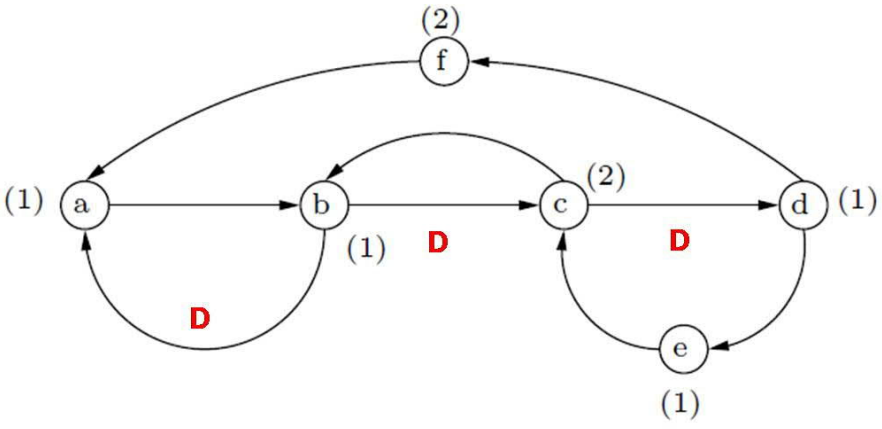
***S3.* Xây dựng các ma trận tiếp theo L(2) và L(3)**

$$L^{(2)} = \begin{bmatrix} 7 & -1 & 3 \\ 6 & 7 & -1 \\ -1 & 3 & -1 \end{bmatrix} \qquad L^{(3)} = \begin{bmatrix} 6 & 7 & -1 \\ 14 & 6 & 10 \\ 10 & -1 & 6 \end{bmatrix}$$

***S4.* Tính giới hạn lặp T_∞:**

$$T_\infty = \max\left(\frac{7}{2}, \frac{7}{2}, \frac{6}{3}, \frac{6}{3}, \frac{6}{3}\right)$$
$$= 3.5$$

10. Cho trước DFG sau và thời gian tính toán của nút được ghi trong dấu ngoặc kế bên nút đó (đơn vị u.t.)



a) Bằng quan sát tính các giới hạn vòng và giới hạn lặp.

b) Tính lại giới hạn vòng bằng giải thuật LPM.

BG.

a) Bằng quan sát tính các giới hạn vòng và giới hạn lặp.
Bảng các giới hạn vòng trong DFG

Vòng thứ	Các nút trong vòng	Giới hạn vòng (u.t.)
1	a–b	(1 + 1)/1 = 2
2	b–c	(1 + 2)/1 = 3
3	c–d–e	(2 + 1 + 1)/1 = 4
4	a–b–c–d–f	(1+1+2+1+2)/2 = 5/2 = 2.5

Suy ra giới hạn lặp T_∞ = max{các giới hạn vòng} = max{2, 3, 4, 2.5} = 4 u.t

b) Tính lại giới hạn vòng bằng giải thuật LPM.
Đánh số các phần tử trễ (D) theo thứ tự từ trái qua phải là d1, d2, và d3.
Ta có ma trân ban đầu L(1):

$$L^{(1)} = \begin{bmatrix} 2 & 2 & -1 \\ 3 & 3 & 2 \\ 5 & 5 & 4 \end{bmatrix}$$

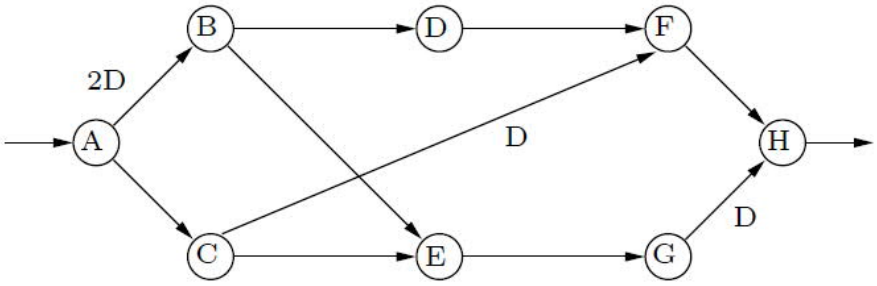
Các ma trận $L^{(2)}$ và $L^{(3)}$

$$L^{(2)} = \begin{bmatrix} 5 & 5 & 4 \\ 7 & 7 & 6 \\ 9 & 9 & 8 \end{bmatrix}$$

$$L^{(3)} = \begin{bmatrix} 9 & 9 & 8 \\ 11 & 11 & 10 \\ 13 & 13 & 12 \end{bmatrix}$$

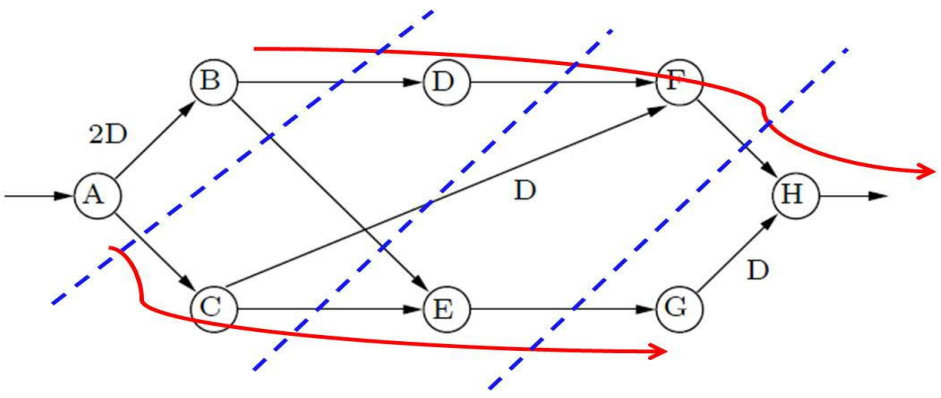
Suy ra giới hạn lặp T_∞ = max{2, 3, 4, 5/2, 7/2, 8/2, 9/3, 11/3, 12/3} = 4 u.t.

11. Xét DFG sau, giả sử thời gian tính toán tại mỗi nút là T.



a) Vẽ và tính đường tới hạn của hệ này.
b) Hãy tìm các nhát cắt thuận thích hợp để tạo đường ống làm cho hệ có đường tới hạn là T? Khi đó cần thêm bao nhiêu Delay (D) để tạo đường ống?

BG.



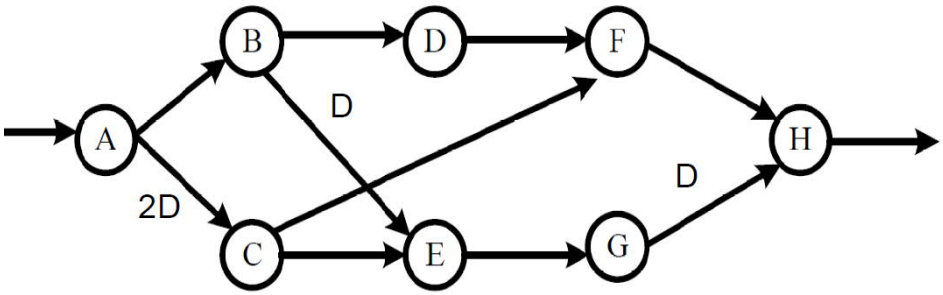
a) Các đường màu đỏ là các đường tới hạn và đường tới hạn đi qua 4 nút có thời gian tính giống nhau nên $T_{critical}$ = 4T.

b) Các đường đứt nét xanh dương là các nhát cắt thuận để tạo đường ống. Khi đó tất cả các nhánh nối gữa 2 nút đều có D ⇒ Tcritical = thời gian tính toán tại mỗi nút = T.
Theo thứ tự các nhát cắt thuận từ trái qua phải, ta có:

- **Nhát cắt thuận 1 cắt 3 nhánh ⇒ cần thêm 3 phần tử Delay(D).**
- **Nhát cắt thuận 2 cắt 4 nhánh ⇒ cần thêm 4 phần tử Delay(D).**
- **Nhát cắt thuận 3 cắt 2 nhánh ⇒ cần thêm 2 phần tử Delay(D).**

Như vậy để tạo đường ống cho DFG này ta cần thêm (3+4+2 =) 9 phần tử Delay (D).

12. Xét DFG trong hình sau với thời gian cần cho mỗi phép toán tại mỗi nút là T:

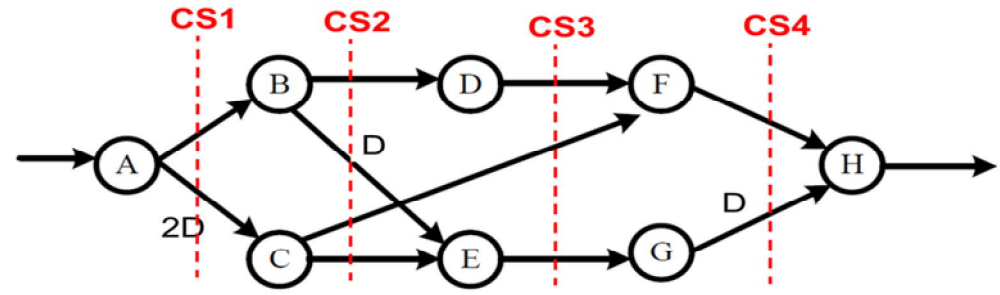


a) Tốc độ mẫu cực đại có thể đạt được trong hệ thống này là bao nhiêu? (chú ý: tốc độ mẫu = 1/ $T_{critical}$)
b) Đặt các thanh ghi tạo đường ống tại các tập cắt thuận thích hợp (feed-forward cutset) để cho đường tới hạn $T_{critical}$ nhỏ nhất có thể được. Khi đó cần thêm bao nhiêu phần tử D nữa?
c) Nếu chỉ có thể thêm 4 phần tử D cho pipeline thì tốc độ mẫu là bao nhiêu?

BG.

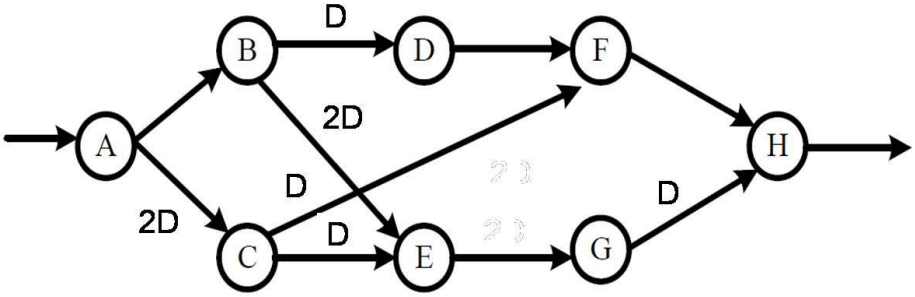
a) Đường tới hạn là 5T (qua các nút $A \rightarrow B \rightarrow D \rightarrow F \rightarrow H$)
Do đó tốc độ mẫu cực đại có thể đạt được trong hệ thống này là T_{sample} = 1/5T

b) Ta nhận thấy để cho Tcritical nhỏ nhất thì phải cho tất cả các nhánh có D ⇒Thực hiện tạo đường ống với các tập cắt thuận theo đường đứt nét trong hình sau:



Với hình mới ta có Tcritical = T, suy ra đó tốc độ mẫu trong hệ thống này là 1/T.
Khi đó cần thêm 11 thanh ghi nữa (vì mỗi cạnh trong tập cắt thêm 1 và tổng cộng tập cắt đi qua 11 cạnh).

c) Vì chỉ có 4D, do đó ta chỉ dùng tập cắt CS2 đi qua 4 cạnh và có hình sau:



Và Tcritical = 3T (đi qua D, F và H) ⇒ Tốc độ mẫu = 1/3T.

13. Một bộ lọc số được mô tả bởi phương trình sau:

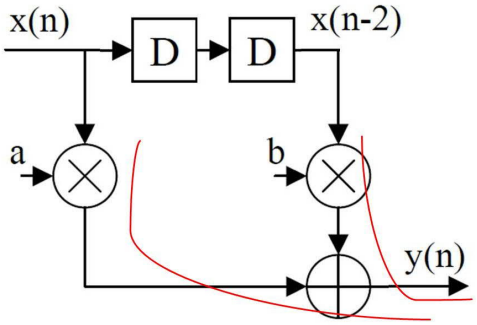
$$y(n) = ax(n) + bx(n–2)$$

a) Vẽ sơ đồ khối của mạch để cho chu kỳ xung nhịp $T_{CLK} = T_M + T_A$ (T_M là thời gian tính toán của bộ nhân và T_A là thời gian tính toán của bộ cộng):

b) Thực hiện song song hệ này với kích thước khối là 2: Viết phương trình và vẽ hình.

BG.

a) Đây là bộ lọc FIR 2 rẽ nhánh (2-tap FIR filter), ta có sơ đồ khối của mạch:



Với các đường đỏ là đường tới hạn ⇒ $T_{critical} = T_M + T_A$ ⇒ $T_{CLK} \geq T_{ciritcal}$ ⇒ Chọn $T_{CLK} = T_M + T_A$

b) Phương trình thực hiện song song với kích thước khối là 2:

$y(2k) = ax(2k) + bx(2k – 2)$

$y(2k+1) = ax(2k+1) + bx(2k – 1)$

SV tự vẽ sơ khối mạch thực hiện các phương trình trên

14. (Ch3. Prob 7) Xét bộ lọc FIR bậc 6:

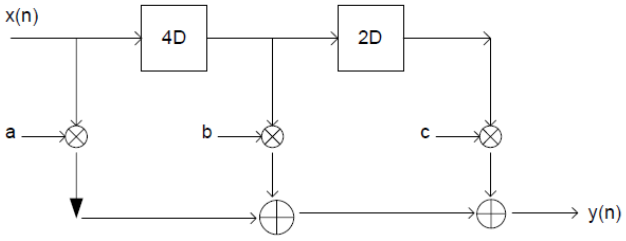
$$y(n) = ax(n) + bx(n-4) + cx(n-6)$$

a) Vẽ sơ đồ khối của mạch để cho chu kỳ xung nhịp $T_{CLK} = T_M + T_A$ (T_M là thời gian tính toán của bộ nhân và T_A là thời gian tính toán của bộ cộng)

b) Vẽ kiến trúc khối của cấu trúc a) với kích thước khối là 3. Sắp xếp lại kiến trúc này sao cho chu kỳ xung nhịp $T_{CLK} = (T_M + T_A)/4$. Giả sử rằng $T_M = 3T_A$.

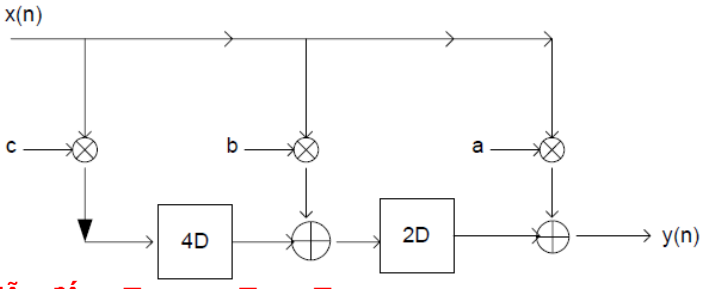
BG.

a) Sơ đồ khối của mạch



Sơ đồ này dẫn đến $T_{sample} = T_M + 2T_A$

Ta vẽ sơ đồ khối chuyển vị của nó



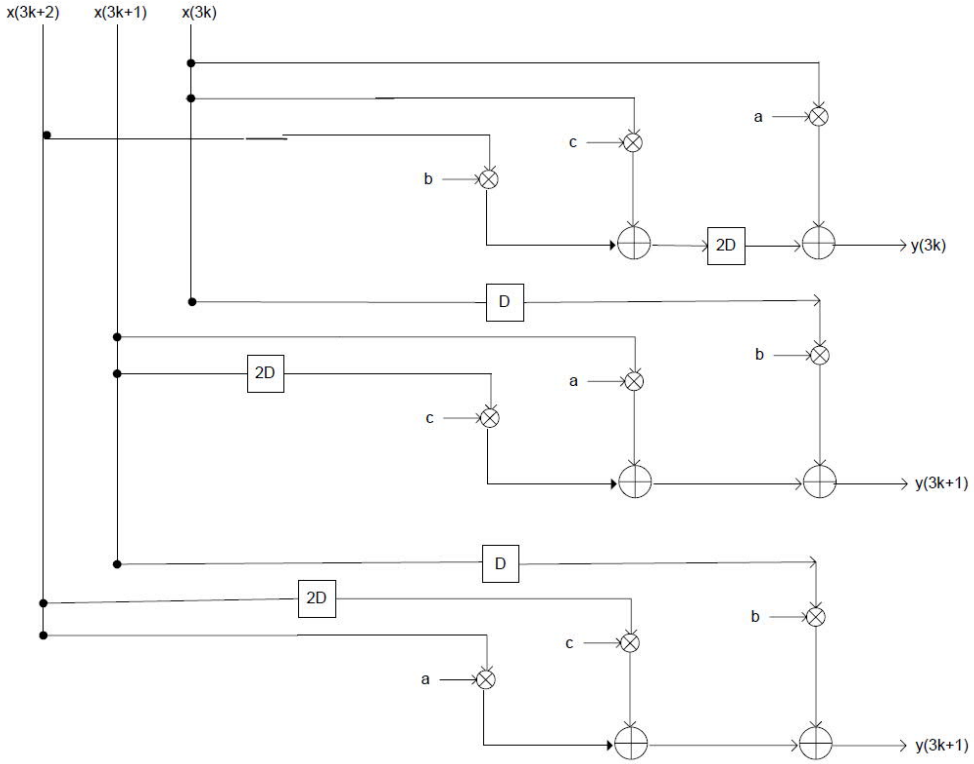
Sơ đồ này dẫn đến $T_{sample} = T_M + T_A$

b) Với kích thước khối là 3 ta có:

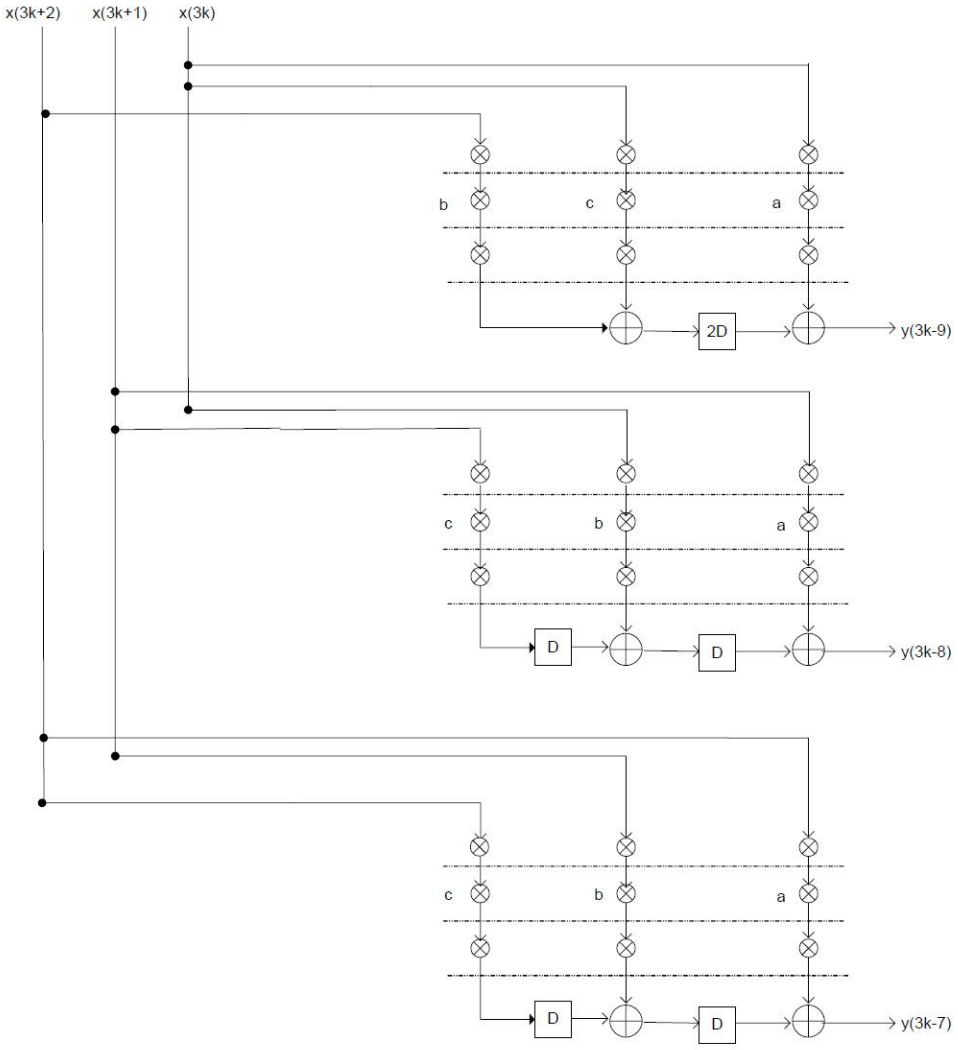
$y(3k) = ax(3k) + bx(3k-4) + cx(3k-6)$

$y(3k+1) = ax(3k+1) + bx(3k-3) + cx(3k-5)$

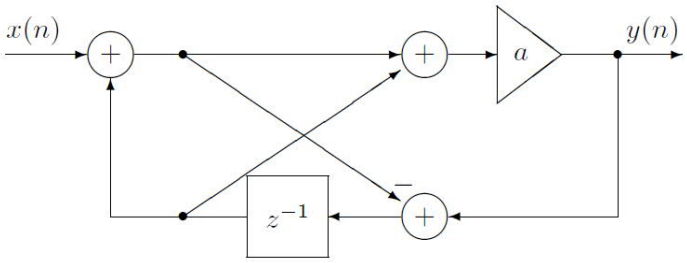
$y(3k+2) = ax(3k+2) + bx(3k-2) + cx(3k-4)$



Sắp xếp lại sơ đồ trên để cho $T_{CLK} = (T_M + T_A)/4$ với $T_M = 3T_A$ bằng cách chia bộ nhân làm 3 phần với các đường đứt nét chia ra pipeline:

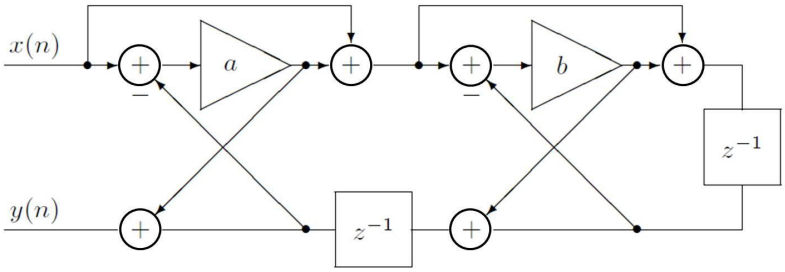


15. Xét mạch lọc sau:



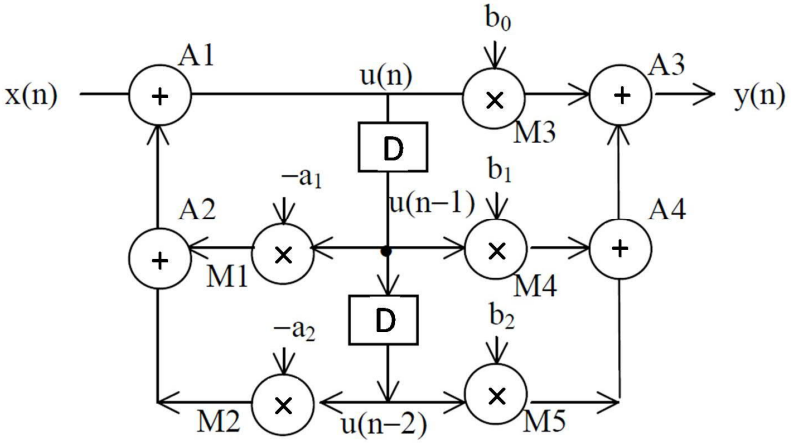
a) Tìm biểu thức của y(n). Từ đó suy ra hàm truyền của mạch lọc này.
b) Vẽ DFG của mạch lọc. Giả sử thời gian tính toán của bộ cộng là T_A = 1 u.t. và bộ nhân là T_M = 2u.t.
c) Hãy tìm đường tới hạn $\boldsymbol{T_{critical}}$ và giới hạn lặp $\boldsymbol{T_\infty}$ cho DFG ở b).
d) Tính lại $\boldsymbol{T_\infty}$ dùng giải thuật LPM.

16. Xét mạch lọc sau:



a) Vẽ DFG của mạch lọc. Giả sử thời gian tính toán của bộ cộng là T_A = 1 u.t. và bộ nhân là T_M = 2u.t.
b) Hãy tìm đường tới hạn $\boldsymbol{T_{critical}}$ và giới hạn lặp $\boldsymbol{T_\infty}$ cho DFG ở b).
c) Tính lại $\boldsymbol{T_\infty}$ dùng giải thuật LPM.

17. Xét bộ lọc số IIR sau:



Trong bộ lọc trên: a1, a2, b0, b1, và b2 là các hệ số bộ lọc; *A1—A4, M1—M5* là nhãn của các bộ cộng (Adder) và các bộ nhân (Multiplier).

a) Hãy vẽ DFG của sơ đồ khối này. Đánh nhãn các nút bằng A1, M1, …
b) Hãy vẽ (các) đường tới hạn trên DFG và tính Tcritical theo T_A (thời gian tính toán của bộ cộng) và T_M.(thời gian tính toán của bộ nhân).
c) Giả sử T_A = 1 u.t. và T_M = 2 u.t., hãy tìm chu kỳ mẫu tối thiểu và giới hạn lặp $\boldsymbol{T_\infty}$.
d) Tái định thì DFG để cho chu kỳ mẫu = $\boldsymbol{T_\infty}$ trong khi vẫn giữ số thanh ghi nhỏ nhất có thể được.

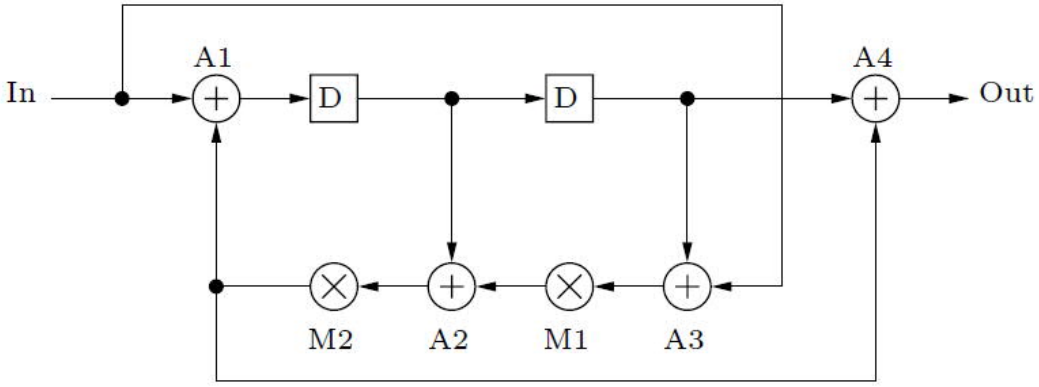
18. Một bộ lọc FIR có cài đặt dạng trực tiếp sau:

$$y(n) = ax(n) + bx(n-2) + cx(n-3)$$

Giả sử thời gian tính toán cho phép cộng-nhân là T.

a) Tạo đường ống cho bộ lọc này để chu kỳ xung nhịp xấp xỉ là T.
b) Thực hiện song song với kích thước khối là 3. Tạo đường ống cho bộ lọc này để chu kỳ xung nhịp là T. Tốc độ mẫu của hệ này là bao nhiêu?
c) Tạo đường ống cho bộ lọc có được từ b) sao cho chu kỳ xung nhịp là T/2. Tốc độ mẫu bây giờ là bao nhiêu?

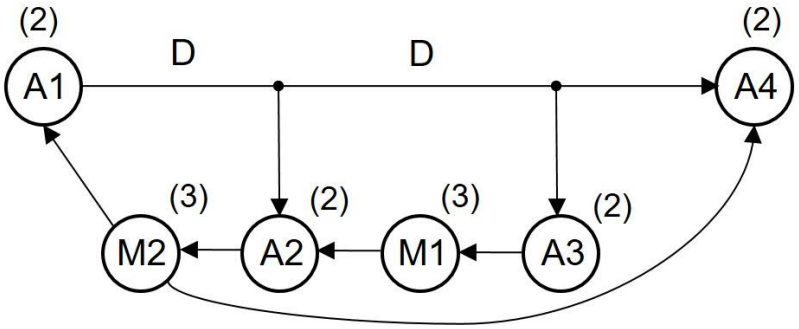
19. Cho trước mạch lọc số sau:



a) Vẽ DFG của mạch này.
b) Hãy tìm đường tới hạn $T_{critical}$ và giới hạn lặp T_∞ bằng quan sát cho DFG ở a). Giả sử thời gian tính toán của bộ cộng là T_A=2 u.t. và bộ nhân là T_M=3 u.t.
c) Tính lại T_∞ dùng giải thuật LPM (đánh số thứ tự các phần tử D theo thứ tự từ trái sang phải).

BG.

a) DFG của mạch trên:



b) Đường [dẫn] tới hạn $T_{critical}$ và giới hạn lặp bằng quan sát:
Đường [dẫn] tới hạn đi qua A3–M1–A2–M2–A1 (hoặc nút cuối là A4)

$$T_{critical} = T_A + T_M + T_A + T_M + T_A = 3T_A + 2T_M = 3 \times 2 + 2 \times 3 = \underline{12\ ns}$$

Tìm giới hạn lặp bằng quan sát:
Tính các giới hạn vòng trong DFG

Vòng thứ	Các nút trong vòng	Giới hạn vòng (u.t.)
1	A1–A2–M2	(2 + 2 + 3)/1 = 7 ns
2	A1–A3–M1–A2–M2	(2 + 2 + 3 + 2 + 3)/2 = 6 ns

Suy ra giới hạn lặp T_∞ = max{các giới hạn vòng} = max{6, 7} = $\underline{\ 7\ ns}$

c) Tìm giới hạn lặp dùng giải thuật LPM:
Đánh số các phần tử trễ (D) theo thứ tự từ trái qua phải là d1, và d2
Ta có ma trận ban đầu $L^{(1)}$ và ma trận cuối $L^{(2)}$:

$$L^{(1)} = \begin{bmatrix} 7 & 0 \\ 12 & -1 \end{bmatrix} \qquad L^{(2)} = \begin{bmatrix} 14 & 7 \\ 19 & 12 \end{bmatrix}$$

Suy ra giới hạn lặp T_∞ = max{7/1, 14/2, 12/2} = $\underline{\ 7\ ns}$