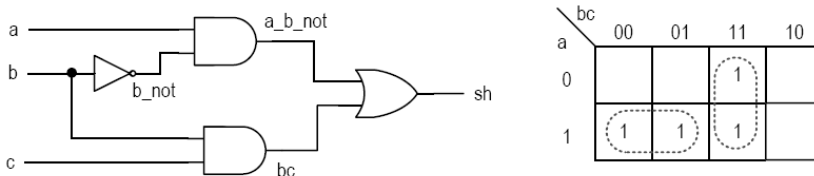
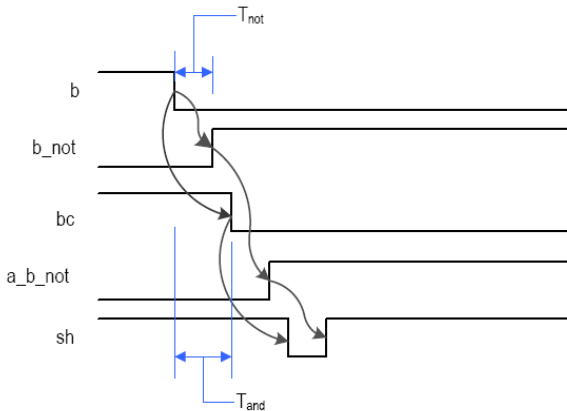


Câu 1: Các bạn được chọn làm 2 trong các câu sau: (3đ)

- a. Hãy vẽ một mạch điện và giải thích để minh họa cho khái niệm thời gian hazard tĩnh. Hazard tĩnh là một sự cố xảy ra trong thực tế ở tín hiệu ngõ ra khi xét đến thời gian trễ, nhưng về mặt lý thuyết thì tín hiệu này không thay đổi giá trị khi không xét đến thời gian trễ; Ví dụ và dạng sóng của hiện tượng Hazard tĩnh:

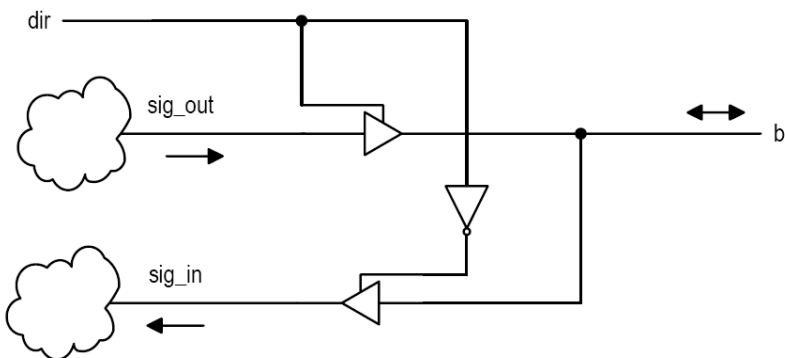


(a) Karnaugh map and schematic

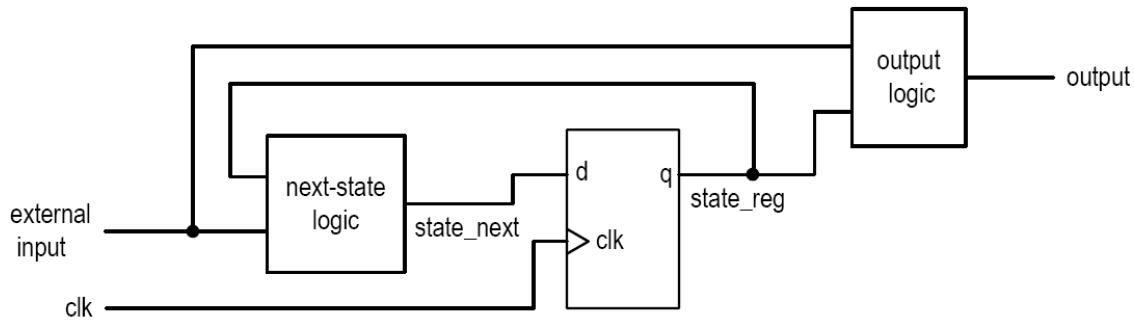


(b) Timing diagram

- b. Hãy vẽ mạch cho port I/O hai chiều dùng 2 cổng 3 trạng thái và giải thích hoạt động của mạch.



- Tín hiệu bi có khả năng đọc lưu vào tín hiệu sig_in, xuất giá trị của tín hiệu sig_out ra ngoài;
 - Tín hiệu sig_out được xuất ra ngoài khi dir có giá trị logic 1;
 - Tín hiệu sig_in nhận giá trị của bi khi dir có giá trị logic 0;
 - Tại một thời điểm, chân bi chỉ có thể đọc vào hoặc xuất giá trị logic ra ngoài;
- c. Hãy vẽ sơ đồ khối cơ bản của mạch đồng bộ và giải thích hoạt động của mạch theo xung clk.



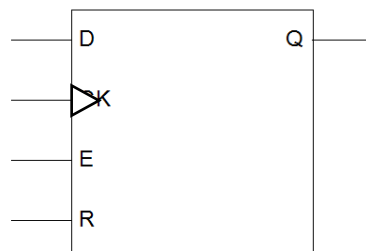
- Hệ thống đồng bộ bao gồm 3 khối chính:
 - o Mạch tổ hợp ngõ vào;
 - o Mạch tổ hợp ngõ ra;
 - o Mạch ghi nhận trạng thái;
 - Nguyên lý hoạt động:
 - o Khi xuất hiện cạnh lên xung clk, tín hiệu state_reg được cập nhật giá trị của state_next và trở thành trạng thái mới;
 - o Giá trị của tín hiệu state_next được tạo ra từ mạch tổ hợp ngõ vào next-state logic; Mạch tổ hợp ngõ ra output logic tạo giá trị ngõ ra căn cứ vào trạng thái hiện tại state_reg và giá trị ngõ vào external input;
- d. Hãy viết bảng trạng thái, viết code VHDL và vẽ sơ đồ khái niệm cho Flip Flop D có tín hiệu cho phép. Bảng trạng thái của FlipFlop D có những ngõ vào: E, R, CK tác động cạnh lên được trình bày ở Bảng 1:

Bảng 1. Bảng trạng thái của FlipFlop D

R	E	CK	Q
1	-	-	0
0	0	-	Q
0	1	↑	D

Sơ đồ giao diện IO được mô tả như Hình 1:

Question_1_d_clc



Hình 1. Sơ đồ giao diện IO của FlipFlop D

Chương trình mô tả VHDL:

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Question_1_d_clc is
    Port ( D, CK, E, R : in  STD_LOGIC;
          Q : out  STD_LOGIC);
end Question_1_d_clc;
architecture Behavioral of Question_1_d_clc is
begin
    PROCESS (CK, E, R)
    BEGIN

```

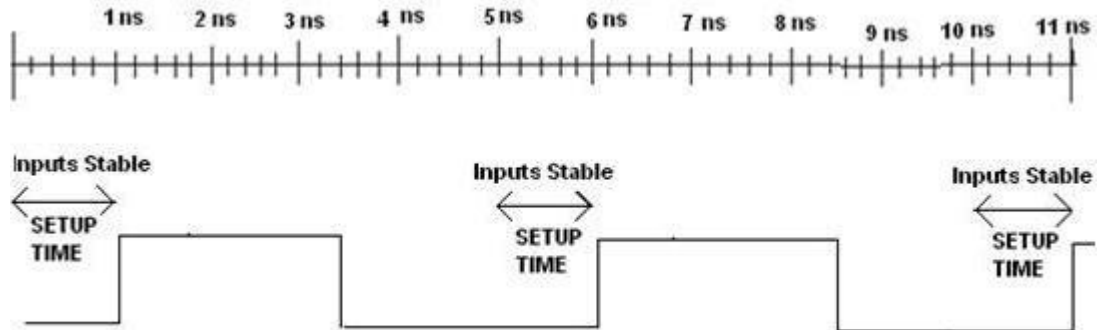
```

IF R = '1' THEN
    Q <= '0';
ELSE
    IF E = '1' THEN
        IF RISING_EDGE(CK) THEN
            Q <= D;
        END IF;
    END IF;
END IF;
END PROCESS;
end Behavioral;

```

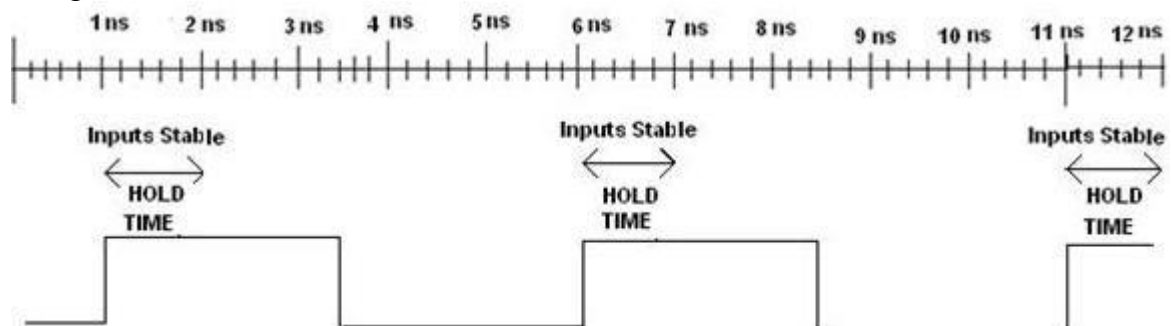
e. Hãy nêu các thông số thời gian cơ bản của Flip Flop D, vẽ dạng sóng minh họa.

- Những thông số thời gian cơ bản của FlipFlop D là: Setup Time (T_{setup}), Hold Time (T_{hold}), Propagation Time ($T_{propagation}$)
- Setup time (T_{setup}) là khoảng thời gian nhỏ nhất ngõ vào phải ổn định ngay trước khi xảy ra sự thay đổi của xung CK. Ví dụ, giả sử có một FlipFlop D ngõ vào CK tác động cạnh lên và có $T_{setup} = 1ns$. Nếu xung CK có chu kỳ 5ns có cạnh lên đầu tiên xảy ra tại thời điểm 1ns thì ngõ vào D của flipflop phải ổn định kể từ thời điểm từ 0ns đến 1ns. Tương tự, ngõ vào D phải ổn định trong những khoảng thời gian từ 5ns đến 6ns, từ 10ns đến 11ns đối với cạnh lên CK thứ hai và thứ ba. Dạng sóng được mô tả như Hình 2;



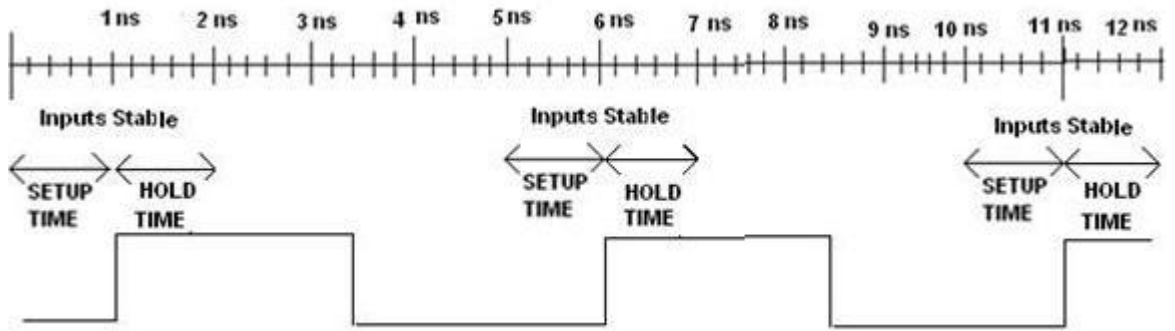
Hình 2. Dạng sóng minh họa cho thông số Setup Time của Flip Flop D

- Hold Time (T_{hold}) là khoảng thời gian nhỏ nhất ngõ vào phải ổn định ngay sau khi xảy ra sự thay đổi của xung CK. Ví dụ, FlipFlop D tác động tích cực cạnh lên có $T_{hold} = 1ns$. Nếu xung CK có chu kỳ 1ns tác động cạnh lên đầu tiên tại thời điểm 1ns thì ngõ vào D phải ổn định trong khoảng từ 1ns đến 2n. Tương tự, ngõ vào D phải ổn định trong các khoảng từ 6ns đến 7ns và 11ns đến 12 ns cho các cạnh lên thứ hai và thứ ba. Sơ đồ dạng sóng minh họa cho thông số Hold Time được mô tả ở **Hình 3**.



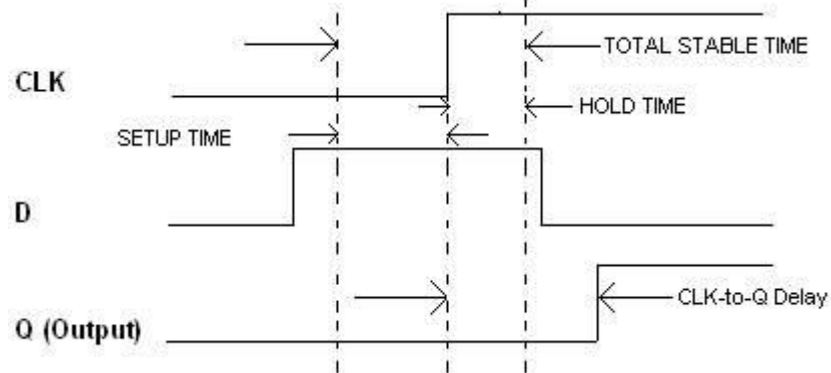
Hình 3. Dạng sóng minh họa cho thông số Hold Time của Flip Flop D

- ⇒ Như vậy kết hợp 2 điều kiện Setup time và Hold time, tín hiệu ngõ vào D phải ổn định 1ns trước khi xuất hiện cạnh lên CK, và 1ns sau khi xuất hiện cạnh lên CK. Tổng cộng thời gian ổn định tối thiểu phải là 2ns, như mô tả ở Hình 4.



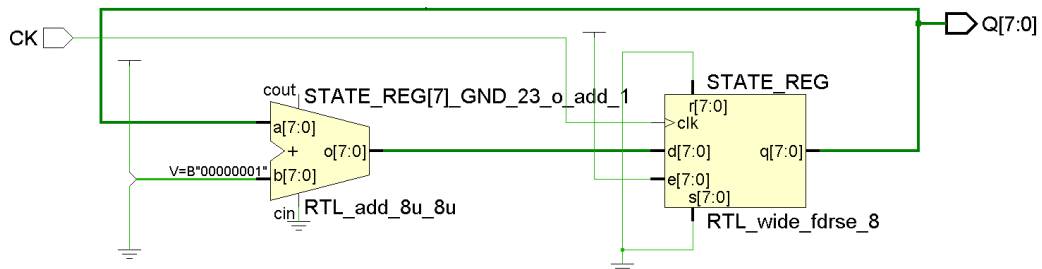
Hình 4. Dạng sóng mô tả hai trường hợp Setup Time và Hold Time

- Propagation Time ($T_{propagation}$) là khoảng thời gian để Flip Flop xuất hiện ngõ ra Q kể từ khi xuất hiện cạnh lên của xung CK. Propagation Delay còn được gọi với một tên khác là: CLOCK to Q Delay ($T_{clocktoQ}, T_{cq}$). Thời gian Propagation Time được minh họa như Hình 5.

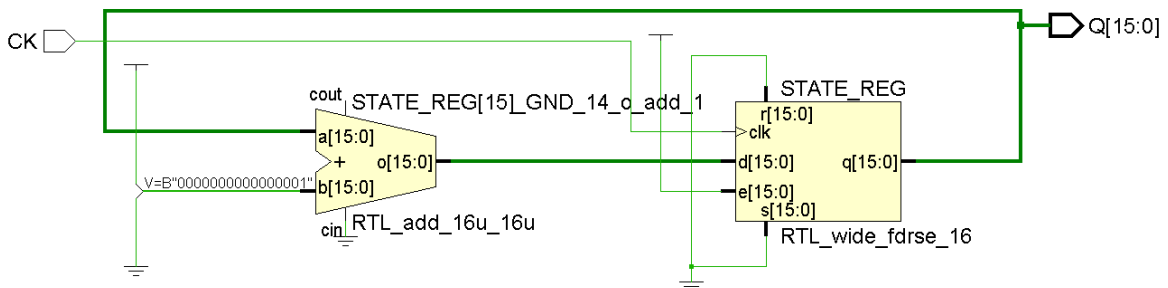


Hình 5. Dạng sóng minh họa thời gian propagation delay

- f. Hãy tính tần số xung ck lớn nhất của mạch đếm nhị phân chạy tự do 8 bit, 16 bit theo 2 chế độ tối ưu về tài nguyên và thời gian. Cho và T_{cq} và T_{setup} của Flip Flop D là 1 và 0,5 ns tương ứng. Sơ đồ khái niệm của mạch đếm nhị phân chạy tự do 8 bits và 16 bits dùng Flip Flop D được mô tả như Hình 6 và Hình 7.



Hình 6. Sơ đồ khái niệm mạch đếm nhị phân tự do 8 bits



Hình 7. Sơ đồ khái niệm mạch đếm nhị phân tự do 16 bits;

Thông qua 2 trường hợp, chúng ta nhận thấy rằng dù là mạch đếm 8 bits hay 16 bits thời gian trễ của tầng FlipFlop đều bằng nhau, vì các flipflop được mắc song song để tạo thành nhiều bits hơn. Thời gian trễ của khối FlipFlop D trong 2 trường hợp được tính như sau:

$$T_{Delay-of-D} = T_{cq} + T_{setup} = 1 + 0.5 = 1.5ns$$

- Đối với mạch đếm nhị phân tự do 8 bits:

o Tần số xung CK tối đa theo hướng tối ưu về tài nguyên:

$$F_{Max} = \frac{1}{T_{+1a8bits} + T_{Delay-of-D}} = \frac{1}{2.4ns + 1.5ns} = 256.41MHz$$

o Tần số xung CK tối đa theo hướng tối ưu về thời gian trễ:

$$F_{Max} = \frac{1}{T_{+1d8bits} + T_{Delay-of-D}} = \frac{1}{1.5ns + 1.5ns} = 333.33MHz$$

- Đối với mạch đếm nhị phân tự do 16 bits:

o Tần số xung CK tối đa theo hướng tối ưu về tài nguyên:

$$F_{Max} = \frac{1}{T_{+1a16bits} + T_{Delay-of-D}} = \frac{1}{5.5ns + 1.5ns} = 142.86MHz$$

o Tần số xung CK tối đa theo hướng tối ưu về thời gian trễ:

$$F_{Max} = \frac{1}{T_{+1d16bits} + T_{Delay-of-D}} = \frac{1}{3.3ns + 1.5ns} = 208.33MHz$$

Câu 2: Các bạn được chọn làm 2 trong các câu sau: (4đ)

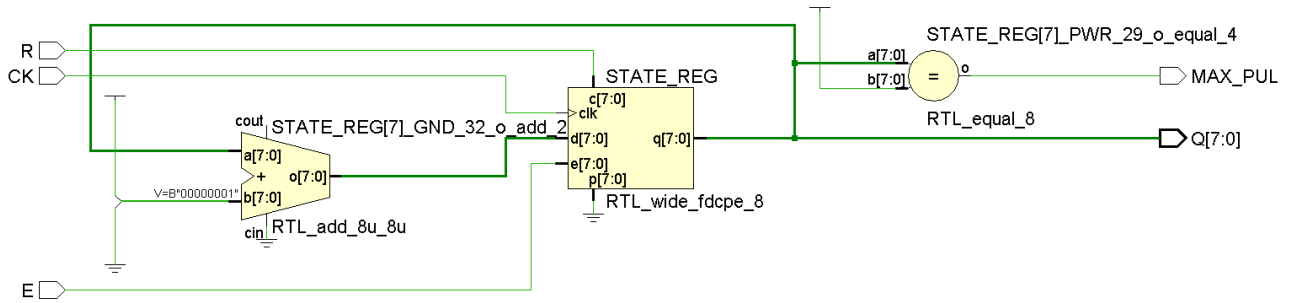
a. Hãy viết code VHDL và vẽ sơ đồ khái niệm cho mạch đếm chạy tự do 8 bit đếm lên gồm các tín hiệu vào là *clk*, *reset* và *en*, tín hiệu ra gồm *q* 8 bit và tín hiệu *max_pul*. Tín hiệu *en* bằng 0 thì mạch ngừng, bằng 1 thì cho phép mạch đếm.

Chương trình mô tả VHDL:

```
library IEEE;
    use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
entity Counter8Bits is
    Port ( R,E,CK : in  STD_LOGIC;
          Q : out  STD_LOGIC_VECTOR (7 downto 0);
          MAX_PUL : OUT STD_LOGIC);
end Counter8Bits;
architecture Behavioral of Counter8Bits is
SIGNAL STATE_NEXT, STATE_REG : STD_LOGIC_VECTOR(7 DOWNT0 0) :=
"00000000";
begin
    PROCESS(CK)
    BEGIN
    IF R = '1' THEN
        STATE_REG <= "00000000";
    ELSE
        IF E = '1' THEN
            IF RISING_EDGE(CK) THEN
                STATE_REG <= STATE_NEXT;
            END IF;
        END IF;
    END IF;
    END PROCESS;
    STATE_NEXT <= STATE_REG + 1;
    Q <= STATE_REG;
    MAX_PUL <= '1' WHEN STATE_REG = "11111111" ELSE
        '0';
```

end Behavioral;

Sơ đồ khái niệm như mô tả ở Hình 8.



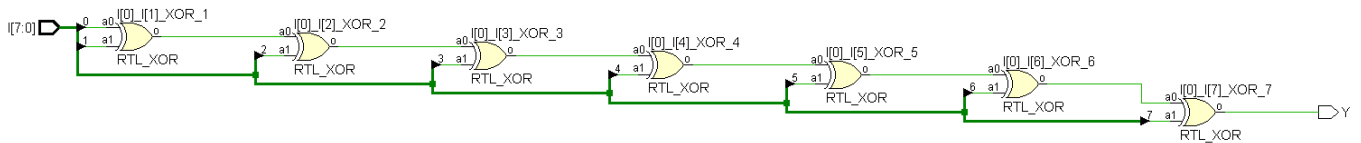
Hình 8. Sơ đồ khái niệm mạch đếm 8 bits

- b. Hãy viết chương trình và vẽ sơ đồ của mạch xor giảm có 8 ngõ vào dạng nối tiếp và dạng cây và nhận xét về thời gian trễ của 2 mạch. Hãy cho biết 2 mạch này minh chứng cho điều gì?

Chương trình mô tả VHDL mạch XOR 8 bits ngõ vào dạng nối tiếp:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity QUESTION_1_B is
    Port ( I : in  STD_LOGIC_VECTOR (7 downto 0);
          Y : out STD_LOGIC);
end QUESTION_1_B;
architecture Behavioral of QUESTION_1_B is
begin
    Y <= I(0) XOR I(1) XOR I(2) XOR I(3) XOR
        I(4) XOR I(5) XOR I(6) XOR I(7);
end Behavioral;
```

Sơ đồ khái niệm mạch XOR 8 bits theo dạng nối tiếp được mô tả ở Hình 9:

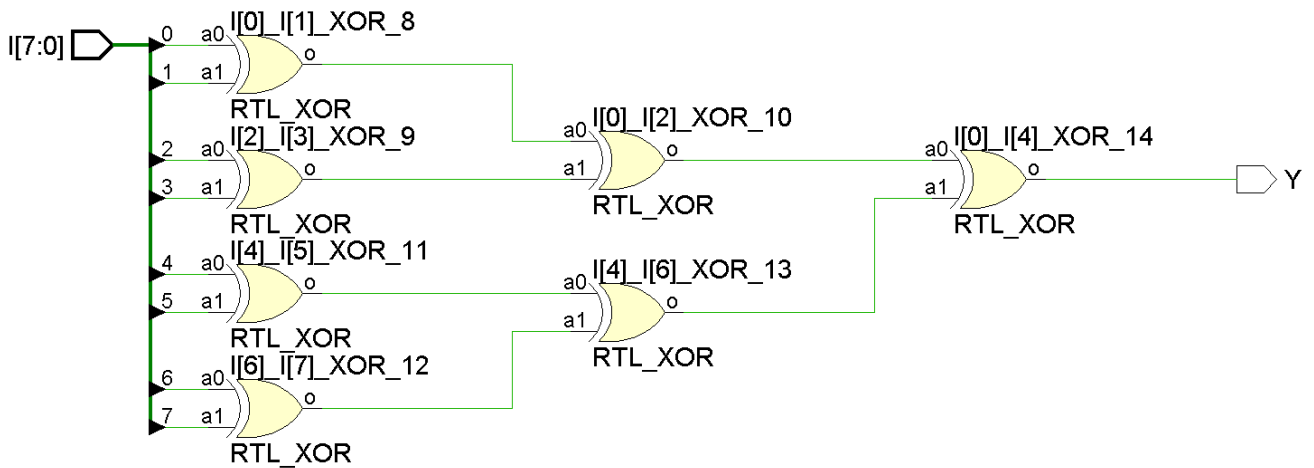


Hình 9. Sơ đồ khái niệm XOR 8 bits theo dạng nối tiếp

Chương trình mô tả VHDL mạch XOR 8 bits dạng hình cây:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity QUESTION_1_B is
    Port ( I : in  STD_LOGIC_VECTOR (7 downto 0);
          Y : out STD_LOGIC);
end QUESTION_1_B;
architecture Behavioral of QUESTION_1_B is
begin
    Y <= ((I(0) XOR I(1)) XOR (I(2) XOR I(3))) XOR
        ((I(4) XOR I(5)) XOR (I(6) XOR I(7)));
end Behavioral;
```

Sơ đồ khái niệm mạch XOR 8 bits theo dạng hình cây được mô tả như Hình 10.



Hình 10. Sơ đồ khái niệm XOR 8 bits theo dạng hình cây

Tính toán tài nguyên và kết luận:

- Thời gian trễ:
 - o Dạng nối tiếp: $Delay = 8 \times 0.4 = 3.2ns$
 - o Dạng song song: $Delay = 3 \times 0.4 = 1.2ns$
- Tài nguyên:
 - o Dạng nối tiếp: $Gate\ count = 7 \times 44 = 308gates$
 - o Dạng song song: $Gate\ count = 7 \times 44 = 308gates$

Kết luận: Mặc dù có cùng tài nguyên được sử dụng, nhưng thời gian trễ của của cấu trúc dạng song song nhỏ hơn thời gian trễ của cấu trúc nối tiếp gấp 2.66 lần.

c. Hãy viết code VHDL và vẽ sơ đồ khái niệm cho mạch đếm mod 100.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
entity Question_2_c is
    Port ( CK, E, R : in  STD_LOGIC;
          Q : out  STD_LOGIC_VECTOR (6 downto 0));
end Question_2_c;
architecture Behavioral of Question_2_c is
    SIGNAL Q_NEXT, Q_REG : STD_LOGIC_VECTOR(6 DOWNTO 0) := (
        OTHERS=>'0');
begin
    -- FOR D FLIP FLOP
    PROCESS(CK, E, R)
    BEGIN
        IF R = '1' THEN
            Q_REG <= (OTHERS=>'0');
        ELSE
            IF E = '1' THEN
                IF RISING_EDGE(CK) THEN
                    Q_REG <= Q_NEXT;
                END IF;
            END IF;
        END IF;
    END PROCESS;

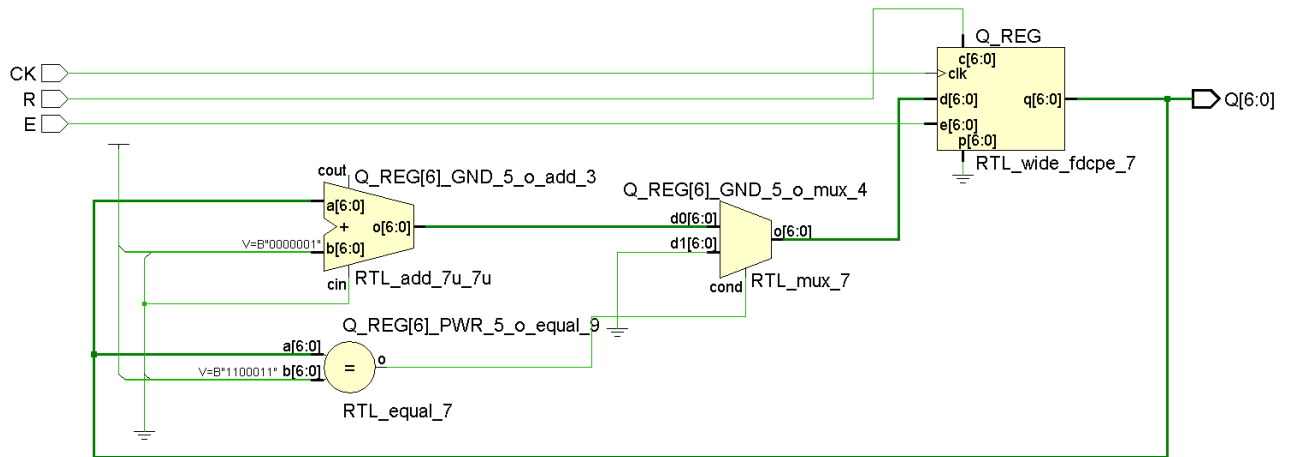
    Q_NEXT <= (OTHERS => '0') WHEN Q_REG = 99 ELSE
        Q_REG + 1;

    Q <= Q_REG;

```

end Behavioral;

Sơ đồ khái niệm được mô tả như Hình 11.



Hình 11. Sơ đồ khái niệm mạch đếm Mod 100

- d. Hãy viết code VHDL và vẽ sơ đồ khái niệm cho mạch đếm mod m có thể lập trình 8 bit, theo cách tối ưu.

```

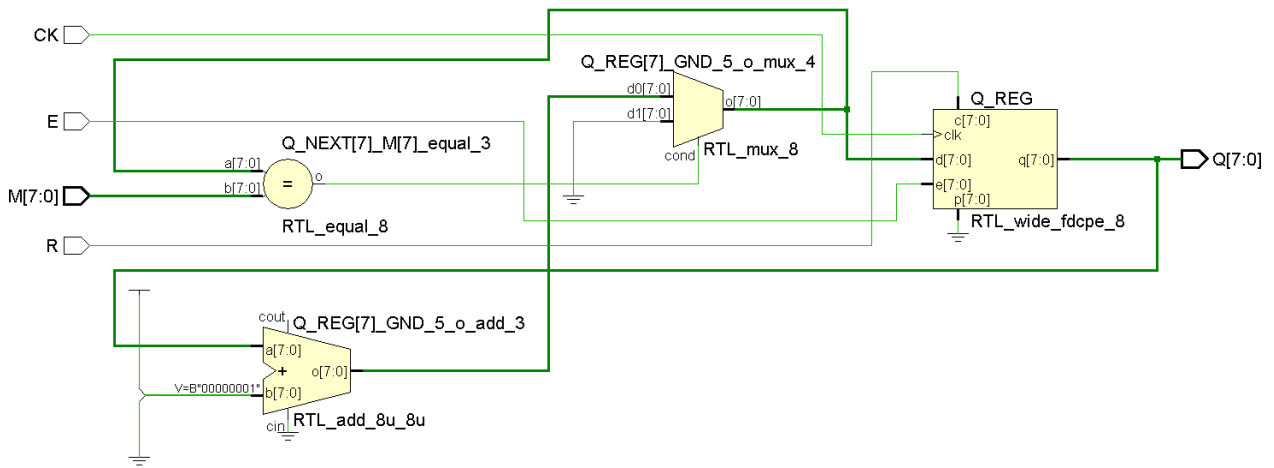
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
entity Question_2_D is
    Port ( CK, E, R : in  STD_LOGIC;
          M : in  STD_LOGIC_VECTOR (7 downto 0);
          Q : out  STD_LOGIC_VECTOR (7 downto 0));
end Question_2_D;
architecture Behavioral of Question_2_D is
    SIGNAL Q_NEXT, Q_REG : STD_LOGIC_VECTOR(7 DOWNTO 0) :=
        (OTHERS=>'0');
begin
    -- FOR D FLIP FLOP
    PROCESS(CK, E, R)
    BEGIN
        IF R = '1' THEN
            Q_REG <= (OTHERS=>'0');
        ELSE
            IF E = '1' THEN
                IF RISING_EDGE(CK) THEN
                    Q_REG <= Q_NEXT;
                END IF;
            END IF;
        END IF;
    END PROCESS;

    Q_NEXT <= (OTHERS => '0') WHEN Q_NEXT = M ELSE
        Q_REG + 1;

    Q <= Q_REG;
end Behavioral;

```

Sơ đồ khái niệm được mô tả như Hình 12.



Hình 12. Sơ đồ khái niệm mạch đếm Mode M theo dạng tổ ưu

Câu 3: (3đ)

Hãy thiết kế một thanh ghi dịch có chức năng xoay phải (rotate right), dịch phải logic (logic shift right) và dịch phải số học (Arithmetic shift right), có tín hiệu dữ liệu vào để dịch là *a* 10 bit, tín hiệu *ctrl* 2 bit để chọn 1 trong 3 chế độ dịch và tín hiệu *amt* 2 bit có 4 trạng thái để chọn dịch 1, 2, 3, 4 bit, tín hiệu ra là *Q*.

a. Hãy viết chương trình dùng ngôn ngữ VHDL thực hiện chức năng trên bằng cách dùng lệnh gán tín hiệu có lựa chọn.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Question_3_a_clc is
    Port ( A : in STD_LOGIC_VECTOR (9 downto 0);
          CTRL : in STD_LOGIC_VECTOR (1 downto 0);
          AMT : in STD_LOGIC_VECTOR (1 downto 0);
          Q : out STD_LOGIC_VECTOR (9 downto 0));
end Question_3_a_clc;
architecture Behavioral of Question_3_a_clc is
    SIGNAL ROTATE, LOGICRIGHT, ARITHRIGH : STD_LOGIC_VECTOR(9
        DOWNT0 0) := (OTHERS=>'0');
begin
    WITH AMT SELECT
        ROTATE <= A(0)&A(9 DOWNT0 1) WHEN "00",
                A(1 DOWNT0 0)&A(9 DOWNT0 2) WHEN "01",
                A(2 DOWNT0 0)&A(9 DOWNT0 3) WHEN "10",
                A(3 DOWNT0 0)&A(9 DOWNT0 4) WHEN OTHERS;

    WITH AMT SELECT
        LOGICRIGHT <= '0'&A(9 DOWNT0 1) WHEN "00",
                    "00"&A(9 DOWNT0 2) WHEN "01",
                    "000"&A(9 DOWNT0 3) WHEN "10",
                    "0000"&A(9 DOWNT0 4) WHEN OTHERS;

    WITH AMT SELECT
        ARITHRIGH <= A(9)&A(9 DOWNT0 1) WHEN "00",
                    A(9)&A(9)&A(9 DOWNT0 2) WHEN "01",
                    A(9)&A(9)&A(9)&A(9 DOWNT0 3) WHEN "10",
                    A(9)&A(9)&A(9)&A(9)&A(9 DOWNT0 4) WHEN OTHERS;

    WITH CTRL SELECT
        Q <= ROTATE WHEN "00",
            LOGICRIGHT WHEN "01",

```

```

ARITHRIGHT WHEN "10",
A WHEN OTHERS;

```

```
end Behavioral;
```

b. Hãy cải tiến lại chương trình cho tối ưu.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Question_3_a_clc is
    Port ( A : in  STD_LOGIC_VECTOR (9 downto 0);
          CTRL : in  STD_LOGIC_VECTOR (1 downto 0);
          AMT : in  STD_LOGIC_VECTOR (1 downto 0);
          Q : out STD_LOGIC_VECTOR (9 downto 0));
end Question_3_a_clc;

architecture Behavioral of Question_3_a_clc is
    SIGNAL TEMP : STD_LOGIC_VECTOR(9 DOWNTO 0) := (OTHERS=>'0');
begin
    WITH CTRL SELECT
        TEMP <= A WHEN "00",
              (OTHERS => '0') WHEN "01",
              (OTHERS => A(9)) WHEN "10",
              A WHEN OTHERS;

    with AMT SELECT
        Q <= TEMP(0)&TEMP(9 DOWNTO 1) WHEN "00",
          TEMP(1 DOWNTO 0)&TEMP(9 DOWNTO 2) WHEN "01",
          TEMP(2 DOWNTO 0)&TEMP(9 DOWNTO 3) WHEN "10",
          TEMP(3 DOWNTO 0)&TEMP(9 DOWNTO 4) WHEN OTHERS;
end Behavioral;

```

Bảng tài nguyên và thời gian trễ của 1 số toán tử.

Kích thước ngõ vào n	Toán tử VHDL									
	nand	xor	>_a	>_d	=	+1_a	+1_d	+_a	+_d	mux
TÀI NGUYÊN										
8	8	22	25	68	26	27	33	51	118	21
16	16	44	52	102	51	55	73	101	265	42
32	32	85	105	211	102	113	153	203	437	85
64	64	171	212	398	204	227	313	405	755	171
THỜI GIAN TRỄ (ns)										
8	0.1	0.4	4.0	1.9	1.0	2.4	1.5	4.2	3.2	0.3
16	0.1	0.4	8.6	3.7	1.7	5.5	3.3	8.2	5.5	0.3
32	0.1	0.4	17.6	6.7	1.8	11.6	7.5	16.2	11.1	0.3
64	0.1	0.4	35.7	14.3	2.2	24.0	15.7	32.2	22.9	0.3

Cán bộ coi thi không được giải thích đề thi, cho đề thi vào túi bài thi để chấm bài.

Chuẩn đầu ra của học phần (về kiến thức)	Nội dung kiểm tra
[G 1.2]: Có khả năng tính toán các vấn đề về thời gian	Câu 1

[G 2.1]: Có khả năng phân tích chia sẻ toán tử	Câu 2
[G 2.2]: Có khả năng phân biệt mạch đồng bộ và không đồng bộ	Câu 1
[G 4.1]: Có khả năng phân tích tính toán hiệu suất và cải tiến	Câu 3
[G 4.2]: Có khả năng thiết kế mạch tuần tự: mạch đếm	Câu 4

Ngày 13 tháng 06 năm 2016
Thông qua Trưởng ngành