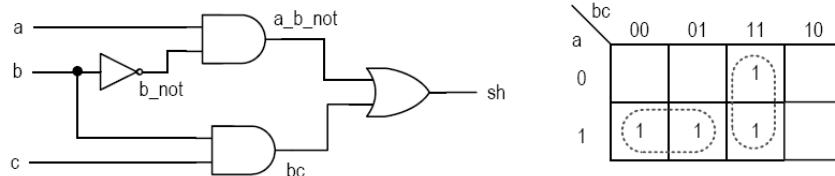
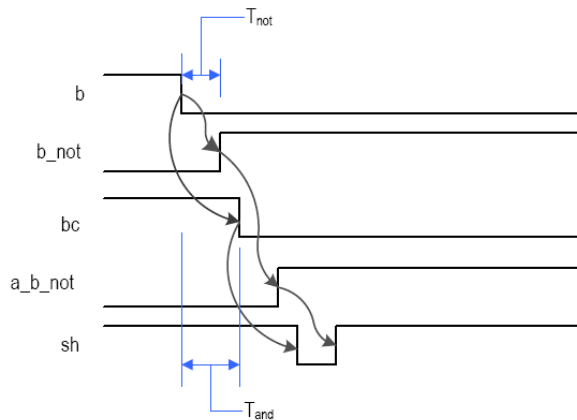


Câu 1: Các bạn được chọn làm 2 trong các câu sau: (2đ)

a. Hãy vẽ một mạch điện và giải thích để minh họa cho khái niệm thời gian hazard tĩnh. (1đ)



(a) Karnaugh map and schematic



(b) Timing diagram

Hiện tượng Hazard tĩnh là hiện tượng có xung bất ổn định xảy ra ở ngõ ra của mạch mà đáng lẽ ở trong trường hợp lý tưởng (không có thời gian trễ của các cổng logic) thì ngõ ra đó không thay đổi. Thời gian Hazard tĩnh là khoảng thời gian xảy ra sự bất ổn định đó.

Hình (a) mô tả mạch minh họa cho việc xảy ra hiện tượng Harzard tĩnh. Công thức logic ngõ ra của mạch:

$$sh = a \text{ AND } (\text{not}(b)) + c \text{ AND } b$$

Giả sử ban đầu các tín hiệu ngõ vào a, b, c đều có mức logic 1, đột nhiên tín hiệu b xuống mức logic 0;

- Trong trường hợp lý tưởng, không có thời gian trễ giữa các cổng logic, thì ngõ ra sh luôn có giá trị logic 1 bất chấp giá trị logic của b.
- Trong trường hợp xét đến thời gian trễ của các cổng logic, do tín hiệu b ở cổng AND thứ nhất được qua cổng NOT trước khi vào cổng AND. Nhưng tín hiệu vào cổng AND thứ hai của b thì vào trực tiếp do đó xuất hiện sự trễ tín hiệu và xuất hiện thời gian mà cả hai a_b_not và bc đều bằng 0. Điều này làm cho ngõ ra sh xuất hiện giá trị logic 0 trước khi lên mức 1.

b. Hãy liệt kê các hàm Big_O và cho biết đặc tính của chúng. (1đ)

Các hàm Big_O bao gồm: $1, n, \log_2 n, n^2, n^3, 2^n$

Các hàm Big_O cho biết mối liên hệ giữa số bit ngõ vào và độ phức tạp của thuật toán xử lý (được đại diện bằng đại lượng thời gian xử lý thuật toán).

Hàm Big_O(1) thuật toán có thời gian xử lý không thay đổi khi thay đổi số bit ngõ vào;

Hàm Big_O(n) thuật toán có thời gian xử lý thay đổi tuyến tính với sự thay đổi của số bit ngõ vào;

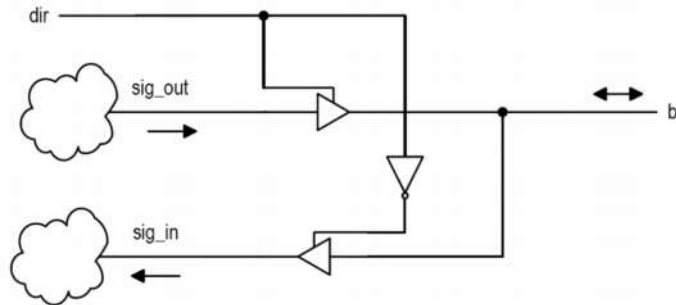
Hàm $Big_O(\log_2 n)$ thuật toán có thời gian xử lý thay đổi chậm hơn so với sự thay đổi của số bit ngõ vào;

Hàm $Big_O(n^2)$ thuật toán có thời gian xử lý thay đổi nhanh bình phương lần so với sự thay đổi số bit ngõ vào;

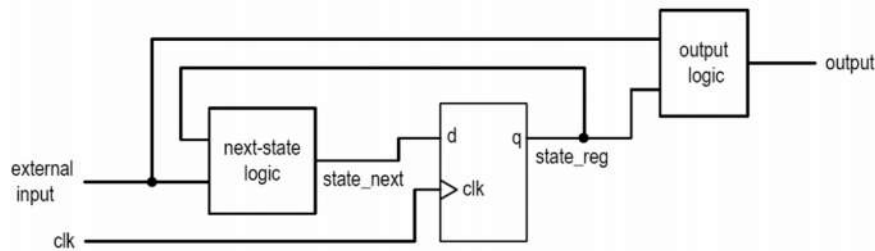
Hàm $Big_O(n^3)$ thuật toán có thời gian xử lý thay đổi mũ 3 lần so với sự thay đổi của số bit ngõ vào;

Hàm $Big_O(2^n)$ thuật toán có thời gian xử lý thay đổi rất nhanh so với sự thay đổi của số bit ngõ vào;

- c. Hãy vẽ mạch cho port IO hai chiều dung 2 cổng 3 trạng thái và giải thích hoạt động của mạch (1đ)



- d. Hãy vẽ sơ đồ khối cơ bản của mạch đồng bộ và giải thích hoạt động của mạch theo xung. (1đ)



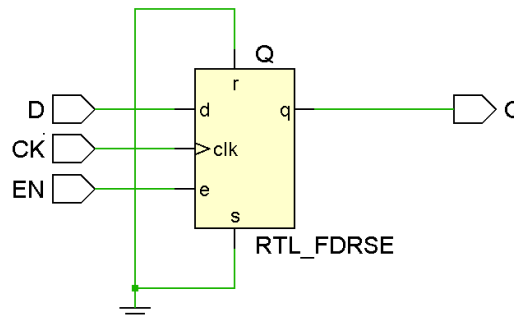
Mạch có thành phần đồng bộ là flip-flop D, trạng thái hiện tại tại `state_reg` được cập nhật giá trị của trạng thái mới `state_next` khi có cạnh lên của xung `clk`;

Mạch tổ hợp ngõ vào lấy giá trị ngõ vào là trạng thái hiện tại tại `state_reg`, kết hợp với giá trị `external input` được cung cấp từ bên ngoài để tạo ra trạng thái mới;

Mạch tổ hợp ngõ ra xem trạng thái hiện tại là gì, tương ứng với các ngõ vào điều khiển gì để cho ra ngõ ra tương ứng với bảng sự thật;

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

- a. Hãy thiết kế mạch FF D có tín hiệu cho phép en, viết bảng trạng thái, viết code VHDL, vẽ sơ đồ khái niệm. (1.5đ)



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity D_FF is
    Port ( D : in  STD_LOGIC;
          CK : in  STD_LOGIC;
          EN : in  STD_LOGIC;
```

```

        Q : out  STD_LOGIC);
end D_FF;

architecture Behavioral of D_FF is
begin
process(CK, D, EN)
begin
    if rising_edge(CK) then
        if EN = '1' then
            Q <= D;
        end if;
    end if;
end process;

end Behavioral;

```

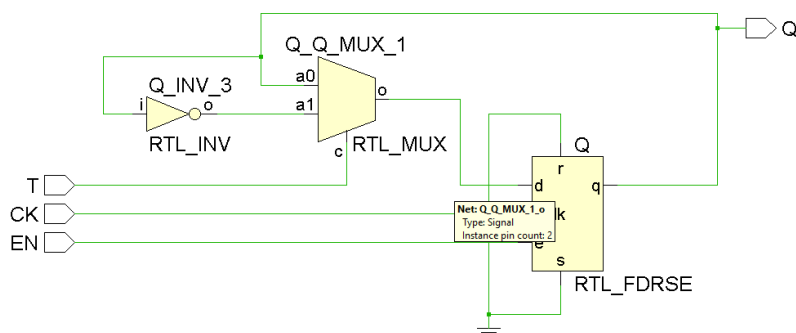
- b. Hãy thiết kế mạch FF T có tín hiệu cho phép en, viết bảng trạng thái, viết code VHDL, vẽ sơ đồ khái niệm. (1.5đ)

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity T_FF is
    Port ( T : in  STD_LOGIC;
          EN, CK : in  STD_LOGIC;
          Q : out  STD_LOGIC);
end T_FF;

architecture Behavioral of T_FF is
signal QT : std_logic := '0';
begin
process(CK, EN, T)
begin
    if rising_edge(CK) then
        if EN = '1' then
            if T = '1' then
                Q <= not(Q);
            end if;
        end if;
    end if;
end process;
end Behavioral;

```



- c. Hãy thiết kế mạch thanh ghi dịch phải chạy tự do 8 bits: Viết code VHDL, vẽ sơ đồ khái niệm. (1.5đ)

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity auto_shift_register is
    Port ( CK : in  STD_LOGIC;

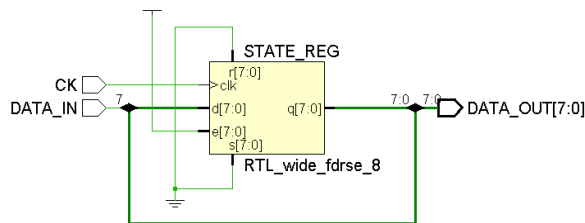
```

```

        DATA_IN : in  STD_LOGIC;
        DATA_OUT : out STD_LOGIC_VECTOR (7 downto 0));
end auto_shift_register;

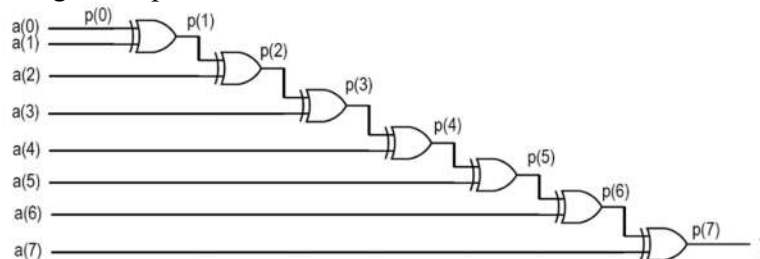
architecture Behavioral of auto_shift_register is
SIGNAL STATE_NEXT, STATE_REG : STD_LOGIC_VECTOR(7 DOWNTO 0) :=
"00000000";
begin
PROCESS(CK, STATE_NEXT)
BEGIN
    IF RISING_EDGE(CK) THEN
        STATE_REG <= STATE_NEXT;
    END IF;
    STATE_NEXT <= DATA_IN & STATE_REG(7 DOWNTO 1);
    DATA_OUT <= STATE_REG;
END PROCESS;
end Behavioral;

```



d. Hãy vẽ sơ đồ của mạch XOR giảm có 8 ngõ vào dạng nối tiếp và dạng cây để minh họa cho cách lập trình có liên quan đến layout và hãy tính thời gian trễ của 2 mạch nếu cho thời gian trễ của 1 cổng XOR là 1 ns. (1.5đ)

- Dạng nối tiếp:



Câu lệnh VHDL để mô tả cấu trúc này:

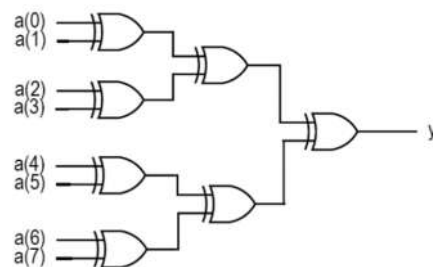
```

Y <= A(0) XOR A(1) XOR A(2) XOR A(3) XOR A(4) XOR A(5) XOR
A(6) XOR A(7);

```

Thời gian trễ là 7ns;

- Dạng cây:



Câu lệnh VHDL để mô tả cấu trúc này:

```

Y <= ((A(0) XOR A(1)) XOR (A(2) XOR A(3))) XOR ((A(4) XOR
A(5)) XOR (A(6) XOR A(7)));

```

Thời gian trễ của cấu trúc này là: 3ns

Câu 3: (2.5đ)

Hãy thiết kế một thanh ghi dịch với tín hiệu vào là a (12 bits) và tín hiệu ra là y (12 bits) có thể xoay trái (rotate left) hoặc dịch trái logic (logic shift left) được chọn bởi tín hiệu vào là S (1 bit) và tín hiệu chọn số bit cần dịch num (3 bits) để chọn từ 0 đến 7 bits dịch.

- a. Hãy viết chương trình dùng ngôn ngữ VHDL dùng lệnh gán tín hiệu có lựa chọn, chưa cải tiến.

```
entity CLC_QUESTION_3 is
    Port ( A : in  STD_LOGIC_VECTOR (11 downto 0);
          Y : out STD_LOGIC_VECTOR (11 downto 0);
          S, CK : in  STD_LOGIC;
          NUM : in  STD_LOGIC_VECTOR (2 downto 0));
end CLC_QUESTION_3;

architecture Behavioral of CLC_QUESTION_3 is
    SIGNAL STATE_NEXT, STATE_REG, SHIFT_OUT, ROTATE_OUT :
    STD_LOGIC_VECTOR(11 DOWNTO 0) := (OTHERS=>'0');
begin
    PROCESS(CK, A, NUM)
    BEGIN
        IF RISING_EDGE(CK) THEN
            STATE_REG <= STATE_NEXT;
        END IF;
    END PROCESS;

    STATE_NEXT <= ROTATE_OUT WHEN S = '0' ELSE SHIFT_OUT;
    WITH NUM SELECT
        SHIFT_OUT <= A WHEN "000",
            SHIFT_OUT(10 DOWNTO 0)&'0' WHEN "001",
            SHIFT_OUT(9 DOWNTO 0)&"00" WHEN "010",
            SHIFT_OUT(8 DOWNTO 0)&"000" WHEN "011",
            SHIFT_OUT(7 DOWNTO 0)&"0000" WHEN "100",
            SHIFT_OUT(6 DOWNTO 0)&"00000" WHEN "101",
            SHIFT_OUT(5 DOWNTO 0)&"000000" WHEN "110",
            SHIFT_OUT(4 DOWNTO 0)&"0000000" WHEN OTHERS;
    WITH NUM SELECT
    ROTATE_OUT <= A WHEN "000",
        ROTATE_OUT(10 DOWNTO 0)&ROTATE_OUT(11) WHEN "001",
        ROTATE_OUT(9 DOWNTO 0)&ROTATE_OUT(11 DOWNTO 8) WHEN "010",
        ROTATE_OUT(8 DOWNTO 0)&ROTATE_OUT(11 DOWNTO 7) WHEN "011",
        ROTATE_OUT(7 DOWNTO 0)&ROTATE_OUT(11 DOWNTO 6) WHEN "100",
        ROTATE_OUT(6 DOWNTO 0)&ROTATE_OUT(11 DOWNTO 5) WHEN "101",
        ROTATE_OUT(5 DOWNTO 0)&ROTATE_OUT(11 DOWNTO 4) WHEN "110",
        ROTATE_OUT(4 DOWNTO 0)&ROTATE_OUT(11 DOWNTO 3) WHEN OTHERS;
    Y <= STATE_REG;
end Behavioral;
```

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

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity CLC_QUESTION_3_B is
    Port ( A : in  STD_LOGIC_VECTOR (11 downto 0);
          Y : out STD_LOGIC_VECTOR (11 downto 0);
          S, CK : in  STD_LOGIC;
          NUM : in  STD_LOGIC_VECTOR (2 downto 0));
end CLC_QUESTION_3_B;

architecture Behavioral of CLC_QUESTION_3_B is
```

```

SIGNAL STATE_NEXT, STATE_REG : STD_LOGIC_VECTOR(11 DOWNTO 0) :=
(Others=>'0');
SIGNAL OUT1, OUT2, OUT4 : STD_LOGIC_VECTOR(11 DOWNTO 0) :=
(Others=>'0');
SIGNAL DATA1 : STD_LOGIC := '0';
SIGNAL DATA2 : STD_LOGIC_VECTOR(1 DOWNTO 0) := "00";
SIGNAL DATA4 : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
begin
PROCESS(CK, DATA_IN, NUM)
BEGIN
IF RISING_EDGE(CK) THEN
STATE_REG <= STATE_NEXT;
END IF;
END PROCESS;

STATE_NEXT <= OUT4;

OUT1 <= A WHEN NUM(0) = '0' ELSE A(10 DOWNTO 0)&DATA1;
OUT2 <= OUT1 WHEN NUM(1) = '0' ELSE OUT1(9 DOWNTO 0)&DATA2;
OUT4 <= OUT2 WHEN NUM(2) = '0' ELSE OUT2(7 DOWNTO 0)&DATA4;

DATA1 <= '0' WHEN S = '1' ELSE A(11);
DATA2 <= "00" WHEN S = '1' ELSE OUT1(11 DOWNTO 10);
DATA4 <= "0000" WHEN S = '1' ELSE OUT2(11 DOWNTO 8);

Y <= STATE_REG;
end Behavioral;

```

Câu 4: (2.5 đ)

Một mạch đếm 8 bits có bảng trạng thái như sau:

Inputs						Outputs	Operation
Ck	Reset	Load	En	Ud	D	Q*	
↓	1	-	-	-	-	00000000	Clear
↓	0	1	-	-	D	D	Parallel
↓	0	0	0	-	-	Q	Pause
↓	0	0	1	1	-	Q + 1	Count up
↓	0	0	1	0	-	Q - 1	Count down

a. Hãy viết chương trình dùng ngôn ngữ VHDL, xung clk cạnh xuống.

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

entity CLC_QUESTION_4 is
Port ( CK, RST, LOAD, EN, UD : in STD_LOGIC;
      D : in STD_LOGIC_VECTOR (7 downto 0);
      Q : out STD_LOGIC_VECTOR (7 downto 0));
end CLC_QUESTION_4;

architecture Behavioral of CLC_QUESTION_4 is
SIGNAL Q_REG, Q_NEXT : STD_LOGIC_VECTOR(7 DOWNTO 0) := "00000000";
begin
PROCESS(CK, Q_NEXT, EN, LOAD, RST)
BEGIN
IF RST = '1' THEN
Q_REG <= "00000000";
ELSE

```

```

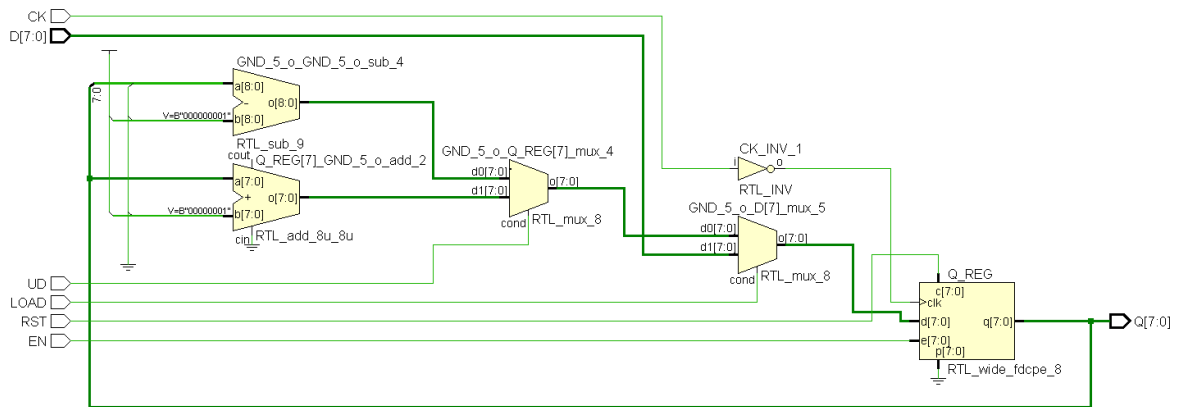
        IF FALLING_EDGE(CK) THEN
            IF EN = '1' THEN
                Q_REG <= Q_NEXT;
            ELSE
                Q_REG <= Q_REG;
            END IF;
        END IF;
    END IF;
END IF;
END PROCESS;

PROCESS (UD, Q_REG, LOAD)
BEGIN
    IF LOAD = '1' THEN
        Q_NEXT <= D;
    ELSE
        IF UD = '1' THEN
            Q_NEXT <= Q_REG + 1;
        ELSE
            Q_NEXT <= Q_REG - 1;
        END IF;
    END IF;
END IF;
END PROCESS;

Q <= Q_REG;
end Behavioral;

```

b. Hãy vẽ sơ đồ khái niệm.



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

Tp. Hồ Chí Minh, ngày 23 tháng 12 năm 2015
Thông qua bộ môn