

**Câu 1:** Sinh viên chỉ được chọn làm 2 trong 5 câu sau đây: (3đ)

- a. Hãy mô tả mạch chuyển đổi từ số nhị phân 16 bits thành mã Gray 16 bits và ngược lại sử dụng VHDL. Mạch có ngõ vào A 16 bits, ngõ ra Q 16 bits, và 1 ngõ điều khiển M 1 bit. Khi M = '0' mạch chuyển từ mã nhị phân sang mã Gray. Khi M = '1' mạch chuyển từ mã Gray sang mã nhị phân. (1.5đ)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Question_1_a_BinaryToGray is
    Port ( I : in  STD_LOGIC_VECTOR (15 downto 0);
          Q : out STD_LOGIC_VECTOR (15 downto 0);
          M : in  STD_LOGIC);
end Question_1_a_BinaryToGray;

architecture Behavioral of Question_1_a_BinaryToGray is
    SIGNAL GRAY_OUT, BINARY_OUT : STD_LOGIC_VECTOR(15 DOWNT0 0) :=
        (OTHERS=>'0');
begin
    GRAY_OUT <= I XOR '0'&I(15 DOWNT0 1);
    BINARY_OUT <= I XOR '0'&BINARY_OUT(15 DOWNT0 1);
    Q <= GRAY_OUT WHEN M = '0' ELSE BINARY_OUT;
end Behavioral;
```

- b. Hãy thiết kế và mô tả bằng VHDL mạch nhân có 2 ngõ vào A, B 8 bits và ngõ ra Q 16 bits. (1.5đ)

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
entity Question_1_b_8_bits_multiplication is
    Port ( A : in  STD_LOGIC_VECTOR (7 downto 0);
          B : in  STD_LOGIC_VECTOR (7 downto 0);
          Q : out STD_LOGIC_VECTOR (15 downto 0));
end Question_1_b_8_bits_multiplication;

architecture Behavioral of Question_1_b_8_bits_multiplication
is
    SIGNAL B0, B1, B2, B3, B4, B5, B6, B7 : STD_LOGIC_VECTOR(7 DOWNT0
0) := (OTHERS=>'0');
    SIGNAL Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7 : STD_LOGIC_VECTOR(15
DOWNT0 0) := (OTHERS=>'0');
begin
    B0 <= (OTHERS=>B(0));
    B1 <= (OTHERS=>B(1));
    B2 <= (OTHERS=>B(2));
    B3 <= (OTHERS=>B(3));
    B4 <= (OTHERS=>B(4));
    B5 <= (OTHERS=>B(5));
```

```

B6 <= (OTHERS=>B(6));
B7 <= (OTHERS=>B(7));

Q0 <= "00000000"&(A AND B0);
Q1 <= "00000000"&(A AND B1)&"0";
Q2 <= "0000000"&(A AND B2)&"00";
Q3 <= "000000"&(A AND B3)&"000";
Q4 <= "00000"&(A AND B4)&"0000";
Q5 <= "000"&(A AND B5)&"00000";
Q6 <= "00"&(A AND B6)&"000000";
Q7 <= "0"&(A AND B7)&"0000000";

Q <= ((Q0 + Q1) + (Q2 + Q3)) + ((Q4 + Q5) + (Q6 + Q7));
end Behavioral;

```

- c. Hãy vẽ mạch đếm lên đồng bộ 2 bits sử dụng 2 flip-flop T và một số cổng logic cần thiết, sau đó mô tả mạch đếm này bằng VHDL. Mạch đếm này có các ngõ vào: CK (cạnh lên), Reset và Enable tích cực mức cao. (1.5đ)

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Question_1_c_SynchronousCounter_2_bits_T_FF is
    Port ( CK, E, R : in STD_LOGIC;
          Q : out STD_LOGIC_VECTOR (1 downto 0));
end Question_1_c_SynchronousCounter_2_bits_T_FF;

architecture Behavioral of
Question_1_c_SynchronousCounter_2_bits_T_FF is
SIGNAL CK0, CK1, T0, T1, Q0, Q1 : STD_LOGIC := '0';
begin
--Describe the T FlipFlop
PROCESS(CK0, E, R)
BEGIN
    IF R = '1' THEN
        Q0 <= '0';
    ELSE
        IF RISING_EDGE(CK0) THEN
            IF E = '1' THEN
                IF T0 = '0' THEN
                    Q0 <= Q0;
                ELSE
                    Q0 <= NOT Q0;
                END IF;
            END IF;
        END IF;
    END IF;
END PROCESS;

PROCESS(CK1, E, R)
BEGIN
    IF R = '1' THEN
        Q1 <= '0';
    ELSE
        IF RISING_EDGE(CK1) THEN
            IF E = '1' THEN

```

```

        IF T1 = '0' THEN
            Q1 <= Q1;
        ELSE
            Q1 <= NOT Q1;
        END IF;
    END IF;
END IF;
END IF;
END PROCESS;
CK0 <= CK;
CK1 <= CK;
T0 <= '1';
T1 <= Q0;
Q <= Q1&Q0;
end Behavioral;

```

- d. Hãy mô tả mạch tìm giá trị nhỏ nhất và lớn nhất trong 5 giá trị a, b, c, d, e bằng VHDL sử dụng hình thức mô tả PROCESS. Mạch có các ngõ vào 64 bits (a, b, c, d, e) và các ngõ ra 64 bits q\_min, q\_max để chứa giá trị nhỏ nhất và lớn nhất. (1.5đ)

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
entity Question_1_d_comparator_5_numbers_64_bits is
    Port ( A, B, C, D, E : in  STD_LOGIC_VECTOR (63 downto 0);
          Q_MIN, Q_MAX : out STD_LOGIC_VECTOR (63 downto 0);
          M : in  STD_LOGIC);
end Question_1_d_comparator_5_numbers_64_bits;

architecture Behavioral of
Question_1_d_comparator_5_numbers_64_bits is
begin
PROCESS(A, B, C, D, E)
BEGIN
    IF (A > B) AND (A > C) AND (A > D) AND (A > E) THEN
        Q_MAX <= A;
    ELSIF (B > C) AND (B > D) AND (B > E) THEN
        Q_MAX <= B;
    ELSIF (C > D) AND (C > E) THEN
        Q_MAX <= C;
    ELSIF (D > E) THEN
        Q_MAX <= D;
    ELSE
        Q_MAX <= E;
    END IF;
END PROCESS;

PROCESS(A, B, C, D, E)
BEGIN
    IF (A < B) AND (A < C) AND (A < D) AND (A < E) THEN
        Q_MIN <= A;
    ELSIF (B < C) AND (B < D) AND (B < E) THEN
        Q_MIN <= B;
    ELSIF (C < D) AND (C < E) THEN
        Q_MIN <= C;

```

```

        ELSIF (D < E) THEN
            Q_MIN <= D;
        ELSE
            Q_MIN <= E;
        END IF;
    END PROCESS;

```

```
end Behavioral;
```

- e. Hãy thiết kế và mô tả VHDL mạch đếm vòng 8 bits có chiều thay đổi được. Mạch có các ngõ vào CK tác động cạnh xuống, Reset và Enable tích cực mức thấp, và Direction. Khi Direction = '0' hoặc '1' thì dữ liệu 8 bits (gồm MỘT bit '1' và BẢY bit '0') tương ứng được xoay trái hoặc xoay phải. (1.5đ)

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity Question_1_e_shift_left_right is
    Port ( CK, E, R, Dir : in STD_LOGIC;
          Q : out STD_LOGIC_VECTOR (7 downto 0));
end Question_1_e_shift_left_right;

architecture Behavioral of Question_1_e_shift_left_right is
    SIGNAL Q_NEXT, Q_REG : STD_LOGIC_VECTOR(7 DOWNTO 0) :=
        "00000001";
begin
    PROCESS(CK, E, R, Q_NEXT)
    BEGIN
        IF R = '1' THEN
            Q_REG <= "00000001";
        ELSE
            IF RISING_EDGE(CK) THEN
                IF E = '1' THEN
                    Q_REG <= Q_NEXT;
                END IF;
            END IF;
        END IF;
    END PROCESS;
    Q_NEXT <= Q_REG(6 DOWNTO 0) & Q_REG(7) WHEN Dir = '0' ELSE
        Q_REG(0) & Q_REG(7 DOWNTO 1);
    Q <= Q_REG;
end Behavioral;

```

**\*\*Lưu ý:** Sinh viên phải viết đầy đủ những thành phần của một tập tin VHDL cho từng câu trả lời.

**Câu 2:** Cho một mạch ALU đơn giản có chức năng được mô tả như bảng trạng thái sau đây: **(3đ)**

CTRL(1:0)	Q(63:0)
00	$A + B$
01	$C + D$
10	$A - B$
11	$C - D$

Trong đó: CTRL(1:0) là ngõ vào điều khiển có 2 bits; Q(63:0) là ngõ ra; A, B, C, D là các ngõ vào số hạng của ALU đều có 64 bits.

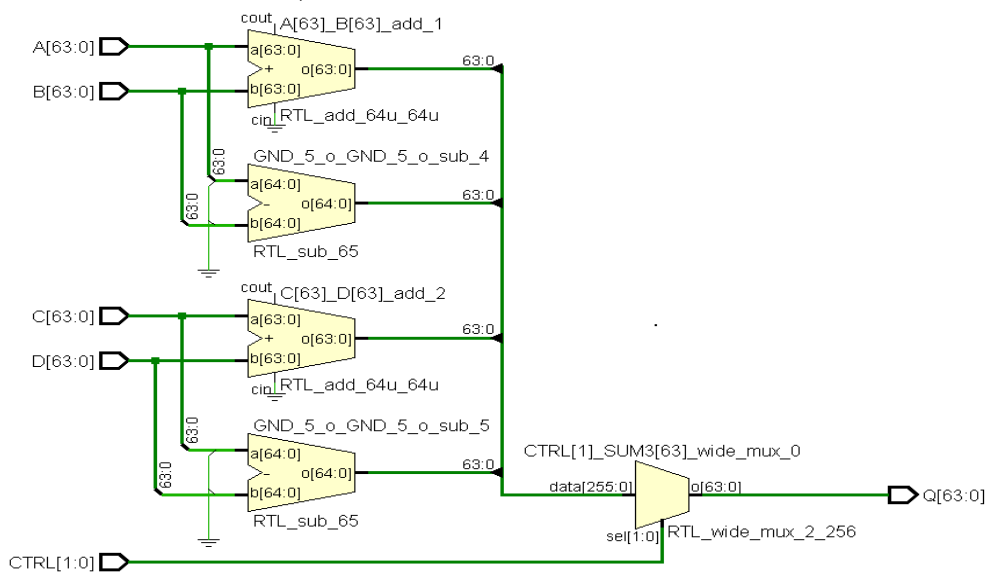
- a. Hãy mô tả mạch ALU này bằng VHDL sao cho chỉ dùng HAI mạch cộng, và HAI mạch trừ. Vẽ lại sơ đồ khái niệm đã mô tả. **(1đ)**

```

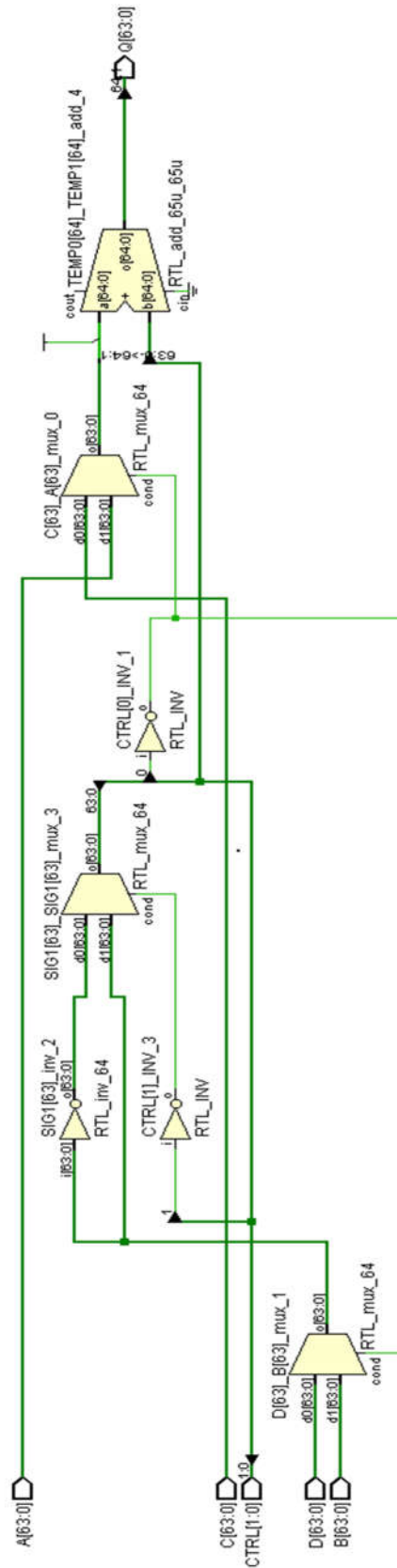
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
entity Question_2_a is
    Port ( A, B, C, D : in  STD_LOGIC_VECTOR (63 downto 0);
          Q : out  STD_LOGIC_VECTOR (63 downto 0);
          CTRL : in  STD_LOGIC_VECTOR (1 downto 0));
end Question_2_a;

architecture Behavioral of Question_2_a is
    SIGNAL SUM0, SUM1, SUM2, SUM3 : STD_LOGIC_VECTOR(63 DOWNTO 0) :=
        (OTHERS=>'0');
begin
    WITH CTRL SELECT
        Q <=
            SUM0 WHEN "00",
            SUM1 WHEN "01",
            SUM2 WHEN "10",
            SUM3 WHEN OTHERS;

    SUM0 <= A + B;
    SUM1 <= C + D;
    SUM2 <= A - B;
    SUM3 <= C - D;
end Behavioral;
    
```



- b. Hãy vẽ lại sơ đồ khái niệm ở câu a. sao cho chỉ dùng MỘT mạch cộng. Mô tả lại sơ đồ khái niệm này bằng VHDL. **(1đ)**



```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity Question_2_b is
    Port ( A, B, C, D : in  STD_LOGIC_VECTOR (63 downto 0);
          Q : out  STD_LOGIC_VECTOR (63 downto 0);

```

```

        CTRL : in  STD_LOGIC_VECTOR (1 downto 0));
end Question_2_b;

architecture Behavioral of Question_2_b is
SIGNAL TEMP0, TEMP1, SUM : STD_LOGIC_VECTOR(64 DOWNT0 0) :=
(OTHERS=>'0');
SIGNAL SIG0, SIG1, SIG2 : STD_LOGIC_VECTOR(63 DOWNT0 0) :=
(OTHERS=>'0');
begin
TEMP0 <= SIG0&'1';
TEMP1 <= SIG2&CTRL(1);
SIG0 <= A WHEN CTRL(0) = '0' ELSE C;
SIG1 <= B WHEN CTRL(0) = '0' ELSE D;
SIG2 <= SIG1 WHEN CTRL(1) = '0' ELSE NOT (SIG1);
SUM <= TEMP0 + TEMP1;
Q <= SUM(64 DOWNT0 1);
end Behavioral;

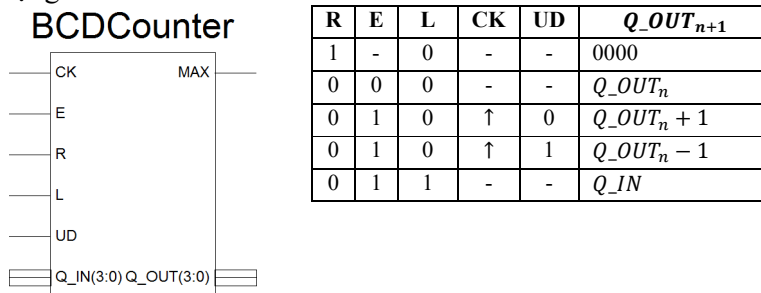
```

- c. Tính toán tài nguyên và thời gian trễ cho từng mạch được thiết kế ở câu a và câu b, sau đó rút ra kết luận về độ tối ưu của mạch ở câu b. (1đ)

Tính toán chỉ mang tính tương đối dựa vào sơ đồ khái niệm đã vẽ mà có những số liệu tính toán khác nhau. Quan trọng nhất sinh viên phải chứng minh được, dựa vào số liệu cụ thể, mạch tối ưu có tài nguyên sử dụng ít hơn so với tài nguyên được sử dụng trong mạch chưa tối ưu.

**Câu 3:** Sinh viên chỉ được chọn làm 1 trong 2 câu sau đây: (3đ)

- a. Hãy thiết kế mạch đếm BCD (**giá trị đếm chỉ bao gồm các giá trị từ 0 đến 9**) có sơ đồ khối và bảng trạng thái như hình vẽ bên dưới:



- Mô tả mạch đếm trên bằng VHDL sử dụng cấu trúc tuần tự 3 thành phần (mạch tổ hợp ngõ vào, mạch cập nhật trạng thái flip-flop D, và mạch tổ hợp ngõ ra); (2đ)

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
entity Question_3_a_BCD_Counter is
    Port ( CK : in  STD_LOGIC;
          R, E, L, UD : in  STD_LOGIC;
          Q_IN : in  STD_LOGIC_VECTOR (3 downto 0);
          Q_OUT : out  STD_LOGIC_VECTOR (3 downto 0);
          MAX : out  STD_LOGIC);
end Question_3_a_BCD_Counter;

architecture Behavioral of Question_3_a_BCD_Counter is
SIGNAL Q_NEXT, Q_REG : STD_LOGIC_VECTOR(3 DOWNT0 0) := "0000";
begin
PROCESS(CK, R, E, Q_NEXT)
BEGIN
    IF R = '1' THEN

```

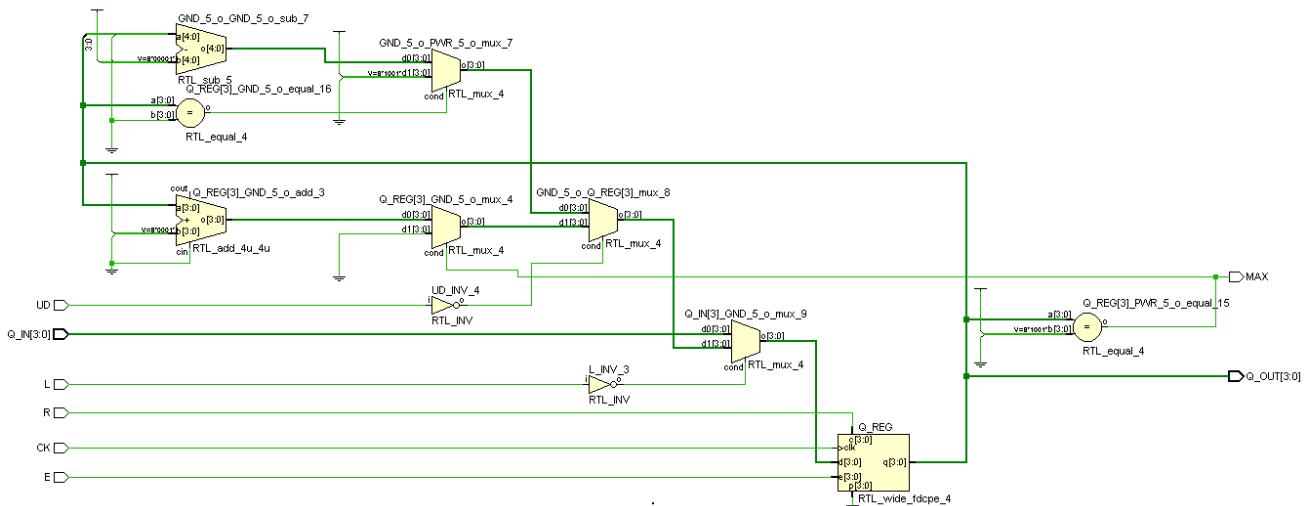
```

        Q_REG <= "0000";
    ELSE
        IF RISING_EDGE(CK) THEN
            IF E = '1' THEN
                Q_REG <= Q_NEXT;
            END IF;
        END IF;
    END IF;
END PROCESS;

PROCESS(Q_REG, L, UD)
BEGIN
    IF L = '0' THEN
        IF UD = '0' THEN
            IF Q_REG = 9 THEN
                Q_NEXT <= "0000";
            ELSE
                Q_NEXT <= Q_REG + 1;
            END IF;
        ELSE
            IF Q_REG = 0 THEN
                Q_NEXT <= "1001";
            ELSE
                Q_NEXT <= Q_REG - 1;
            END IF;
        END IF;
    ELSE
        Q_NEXT <= Q_IN;
    END IF;
END PROCESS;
Q_OUT <= Q_REG;
MAX <= '1' WHEN Q_REG = 9 ELSE '0';
end Behavioral;

```

- Vẽ sơ đồ khái niệm của chương trình VHDL vừa viết; (0.5đ)



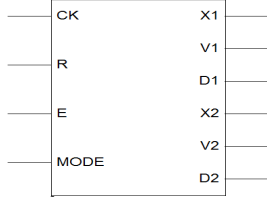
- Mở rộng bộ đếm có thêm chân ngõ ra MAX. MAX chỉ bằng '1' khi ngõ ra Q\_OUT đạt giá trị cực đại. (0.5đ) (Tương tự như chương trình được viết ở trên)

b. Hãy thiết kế mạch đèn đèn giao thông có 2 trụ, mỗi trụ có 3 đèn là Xanh, Vàng, và Đỏ; Có khả năng chọn chế độ hoạt động bằng công tắc. Chế độ hoạt động thứ nhất là hoạt động bình thường, chế độ thứ 2 là chỉ sáng đèn vàng ở cả 2 trụ. Khi hai trụ đèn hoạt động bình thường, thì thời gian sáng đèn Xanh



là 20s, thời gian sáng đèn Vàng là 5s, và thời gian sáng đèn Đỏ là 25s. Sơ đồ khối và bảng trạng thái được cho như bên dưới:

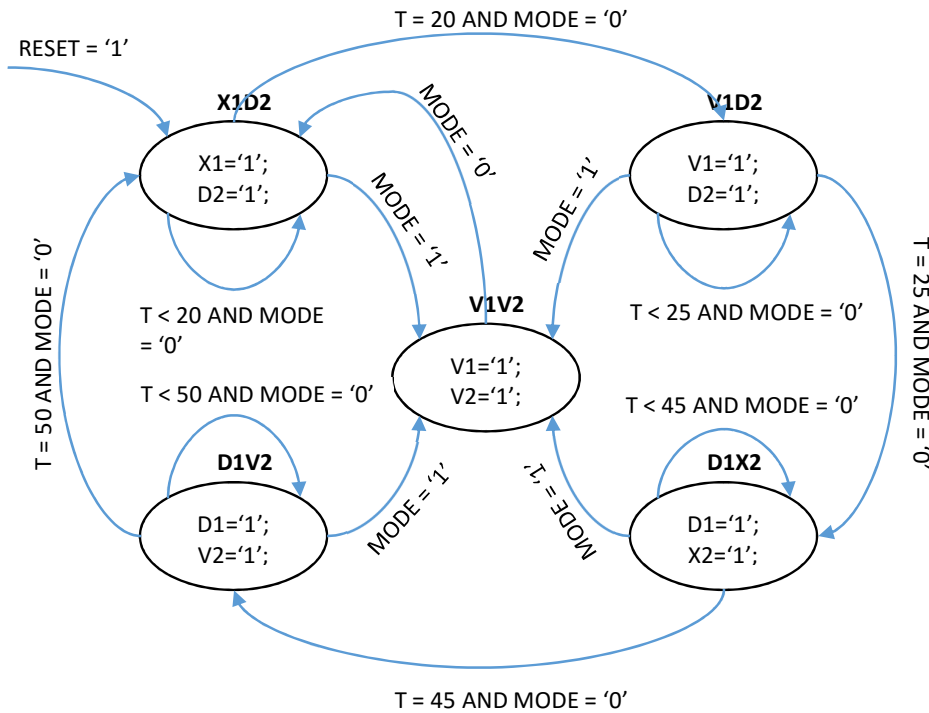
**TRAFFICLIGHTS**



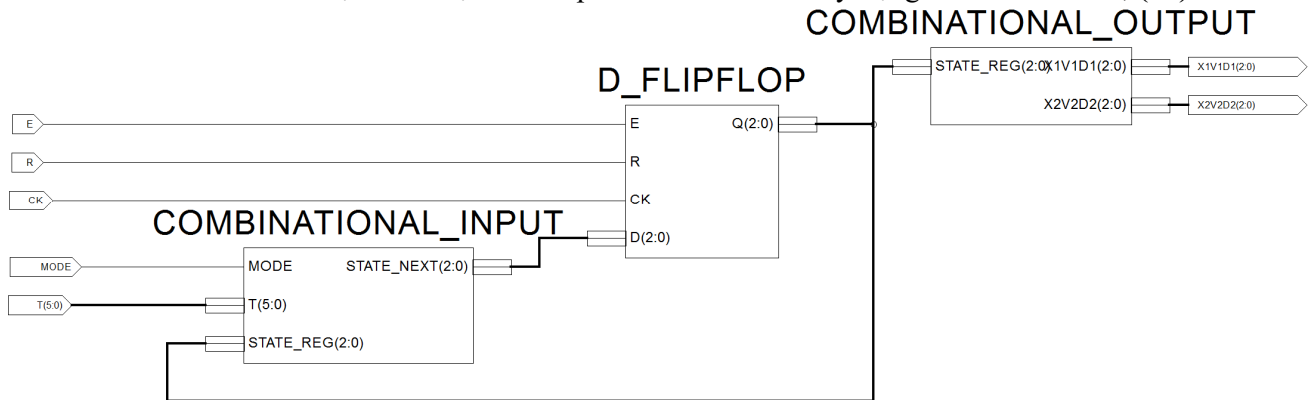
R	CK	E	MODE	Hoạt động
1	-	-	-	Xanh 1, đỏ 2
0	↑	0	-	Ngõ ra không thay đổi
0	↑	1	0	Đèn hoạt động bình thường
0	↑	1	1	Hai trụ chỉ sáng đèn vàng

Yêu cầu:

- Thiết kế mô hình máy trạng thái mô tả hoạt động hệ thống theo bảng trạng thái trên; **(1đ)**



- Vẽ mô hình mạch tuần tự 3 thành phần của mô hình máy trạng thái vừa thiết kế; **(1đ)**



- Mô tả lại mô hình mạch tuần tự vừa thiết kế bằng VHDL. **(1đ)**

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
entity Question_3_b_Traffic_light is
    Port ( CK, R, E, MODE : in STD_LOGIC;
          Q0, Q1 : out STD_LOGIC_VECTOR (2 downto 0));
end entity;

```

```

end Question_3_b_Traffic_light;

architecture Behavioral of Question_3_b_Traffic_light is
TYPE STATES IS (X1D2, V1D2, D1X2, D1V2, V1V2);
SIGNAL STATE_NEXT, STATE_REG : STATES := X1D2;
SIGNAL COUNTER_REG, COUNTER_NEXT : STD_LOGIC_VECTOR(5 DOWNTO 0) :=
"000000";
begin
--TIMING COUNTER
PROCESS(CK, R, E, COUNTER_NEXT, MODE)
BEGIN
    IF R = '1' THEN
        COUNTER_REG <= "000000";
    ELSE
        IF RISING_EDGE(CK) THEN
            IF E = '1' THEN
                COUNTER_REG <= COUNTER_NEXT;
            END IF;
        END IF;
    END IF;
END PROCESS;
COUNTER_NEXT <= "000000" WHEN COUNTER_REG = 50 OR MODE = '1' ELSE
    COUNTER_REG + 1;

--FOR TRAFFICLIGHT CONTROLLER
PROCESS(R, E, STATE_REG)
BEGIN
    IF R = '1' THEN
        STATE_REG <= X1D2;
    ELSE
        IF RISING_EDGE(CK) THEN
            IF E = '1' THEN
                STATE_REG <= STATE_NEXT;
            END IF;
        END IF;
    END IF;
END PROCESS;

PROCESS(STATE_REG, MODE)
BEGIN
    CASE STATE_REG IS
        WHEN X1D2 =>
            IF MODE = '0' THEN
                IF COUNTER_REG = 20 THEN
                    STATE_NEXT <= V1D2;
                ELSE
                    STATE_NEXT <= X1D2;
                END IF;
            ELSE
                STATE_NEXT <= V1V2;
            END IF;
        WHEN V1D2 =>
            IF MODE = '0' THEN
                IF COUNTER_REG = 25 THEN
                    STATE_NEXT <= D1X2;
                END IF;
            END IF;
    END CASE;
END PROCESS;

```

```

        ELSE
            STATE_NEXT <= V1D2;
        END IF;
    ELSE
        STATE_NEXT <= V1V2;
    END IF;
WHEN D1X2 =>
    IF MODE = '0' THEN
        IF COUNTER_REG = 45 THEN
            STATE_NEXT <= D1V2;
        ELSE
            STATE_NEXT <= D1X2;
        END IF;
    ELSE
        STATE_NEXT <= V1V2;
    END IF;
WHEN D1V2 =>
    IF MODE = '0' THEN
        IF COUNTER_REG = 50 THEN
            STATE_NEXT <= X1D2;
        ELSE
            STATE_NEXT <= D1V2;
        END IF;
    ELSE
        STATE_NEXT <= V1V2;
    END IF;
WHEN OTHERS =>
    IF MODE = '0' THEN
        STATE_NEXT <= X1D2;
    ELSE
        STATE_NEXT <= V1V2;
    END IF;
END CASE;
END PROCESS;

PROCESS (STATE_REG)
BEGIN
    CASE STATE_REG IS
        WHEN X1D2 => Q0 <= "100"; Q1 <= "001";
        WHEN V1D2 => Q0 <= "010"; Q1 <= "001";
        WHEN D1X2 => Q0 <= "001"; Q1 <= "100";
        WHEN D1V2 => Q0 <= "100"; Q1 <= "010";
        WHEN OTHERS => Q0 <= "010"; Q1 <= "010";
    END CASE;
END PROCESS;
end Behavioral;

```

**Câu 4:** Sinh viên chỉ được phép sử dụng tiếng Anh để trả lời câu hỏi sau đây. (1đ)

Write down the brief comparisons of Signals, Variables, and Constants in VHDL.

- Similarities:
  - o They are all the temporary wires supporting for designing digital circuit using VHDL;
  - o They do not generate more hardware consumptions after compiling to the lowest level of projects;
- Differences:

- Signals are considered as wires being able to save the logic states. They are declared after architecture keyword and before the begin keyword.
  - Variables are considered as the temporary variables that only have the actual meaning in the high level of describing logic circuits. They are used in process structure of VHDL;
  - Constants are considered as the unchanged values that help the describing tasks more flexible.
- \*\*Bảng dữ liệu về tài nguyên và thời gian trễ cho ở bảng bên dưới.**

Width	VHDL operator									
	nand	xor	> <sub>a</sub>	> <sub>d</sub>	=	+ <sub>1a</sub>	+ <sub>1d</sub>	+ <sub>a</sub>	+ <sub>d</sub>	mux
<b>Area (gate count)</b>										
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
<b>Delay (ns)</b>										
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.

<b>Chuẩn đầu ra của học phần (về kiến thức)</b>	<b>Nội dung kiểm tra</b>
[G 1.2]: Có khả năng tính toán các vấn đề về thời gian	Câu 2
[G 2.1]: Có khả năng phân tích chia sẻ toán tử	Câu 2, Câu 3
[G 2.2]: Có khả năng phân biệt mạch đồng bộ và không đồng bộ	Câu 1
[G 3.3]: Đọc hiểu và trình bày ý tưởng bằng tiếng Anh.	Câu 4
[G 4.1]: Có khả năng phân tích tính toán hiệu suất và cải tiến	Câu 2
[G 4.2]: Có khả năng thiết kế mạch tuần tự: mạch đếm	Câu 3

Tp. Hồ Chí Minh, ngày 15 tháng 12 năm 2015  
 Thông qua bộ môn