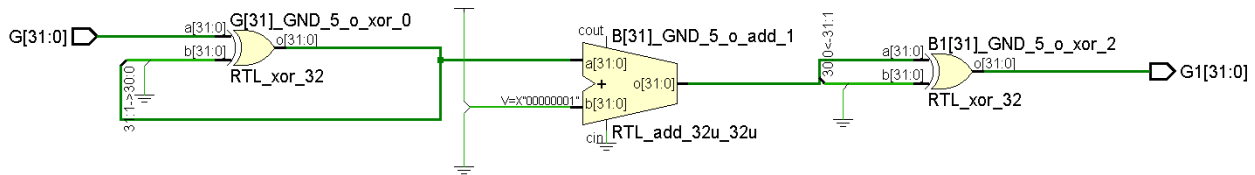


Câu 1: Sinh viên chỉ được chọn làm 2 trong 5 câu sau đây: (Mỗi câu 1.5đ, 2 câu 3đ)

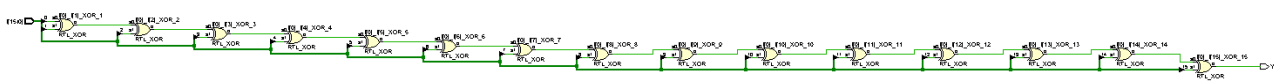
- a. Hãy thiết kế và mô tả mạch tăng mã Gray 32 bit dùng VHDL. Cụ thể ngõ vào là G[31:0] thì ngõ ra là G1[31:0]

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
entity Question_1_a is
    Port ( G : in  STD_LOGIC_VECTOR (31 downto 0);
          G1 : out STD_LOGIC_VECTOR (31 downto 0));
end Question_1_a;
architecture Behavioral of Question_1_a is
    SIGNAL B, B1 : STD_LOGIC_VECTOR(31 DOWNTO 0) := (OTHERS=>'0');
begin
    B <= G XOR '0'&B(31 DOWNTO 1);
    B1 <= B + 1;
    G1 <= B1 XOR '0'&B1(31 DOWNTO 1);
end Behavioral;
```



- b. Hãy thiết kế mạch XOR giảm 16 bits dùng VHDL dạng nối tiếp và vẽ sơ đồ khái niệm. Hãy cải tiến lại chương trình để được cấu trúc dạng cây (Tree Design) và vẽ sơ đồ khái niệm. So sánh hiệu suất và tài nguyên của mạch.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity QUESTION_1_B is
    Port ( I : in  STD_LOGIC_VECTOR (15 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) XOR
        I(8) XOR I(9) XOR I(10) XOR I(11) XOR
        I(12) XOR I(13) XOR I(14) XOR I(15);
end Behavioral;
```

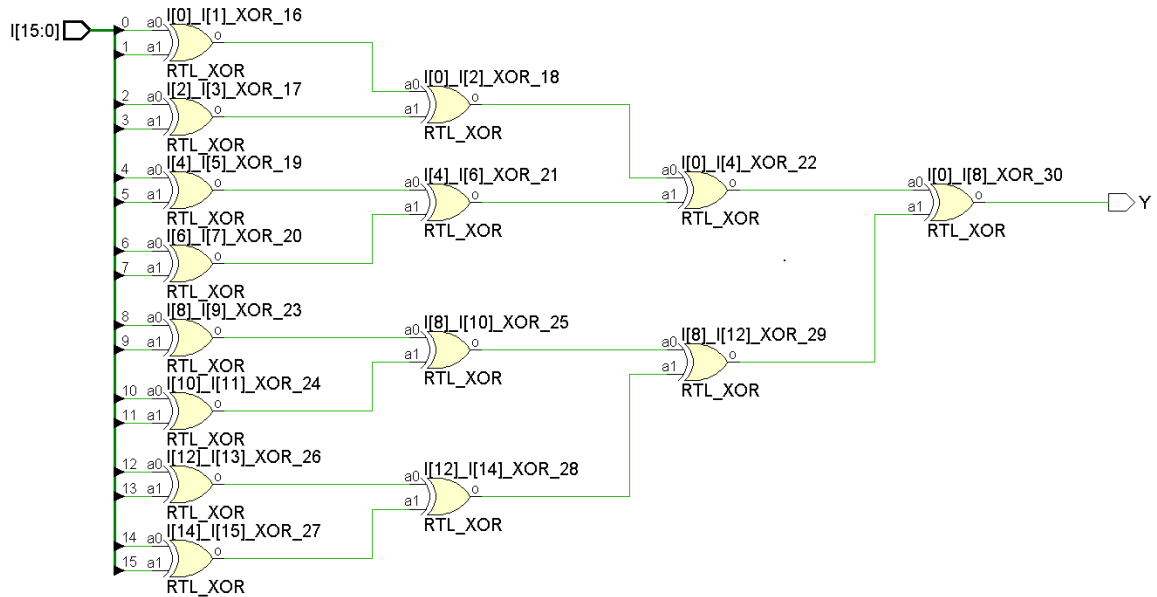


```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity QUESTION_1_B is
    Port ( I : in  STD_LOGIC_VECTOR (15 downto 0);
          Y : out STD_LOGIC);
end QUESTION_1_B;
```

```

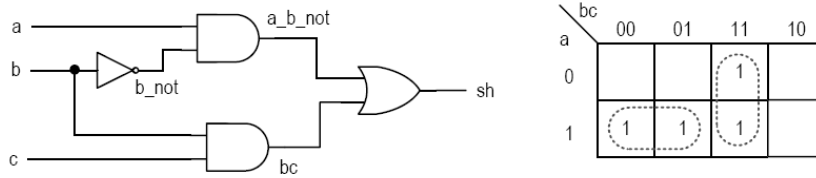
end QUESTION_1_B;
architecture Behavioral of QUESTION_1_B is
begin
Y <= (((I(0) XOR I(1)) XOR (I(2) XOR I(3))) XOR
      ((I(4) XOR I(5)) XOR (I(6) XOR I(7)))) XOR
      (((I(8) XOR I(9)) XOR (I(10) XOR I(11))) XOR
      ((I(12) XOR I(13)) XOR (I(14) XOR I(15))));
end Behavioral;

```

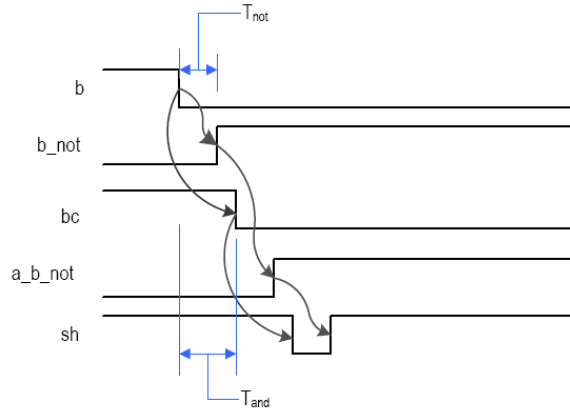


Tính toán tài nguyên và kết luận:

- Thời gian trễ:
 - o Dạng nối tiếp: $Delay = 15 \times 0.4 = 6ns$
 - o Dạng song song: $Delay = 4 \times 0.4 = 1.6ns$
 - Tài nguyên:
 - o Dạng nối tiếp: $Gate\ count = 15 \times 44 = 660gates$
 - o Dạng song song: $Gate\ count = 15 \times 44 = 660gates$
 - 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 3.75 lần.
- c. Hãy phát biểu hiện tượng Hazard tĩnh, Hazard động và cho một ví dụ minh họa của mạch điện gây ra hiện tượng Hazard tĩnh và cách loại bỏ Hazard.
- 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:

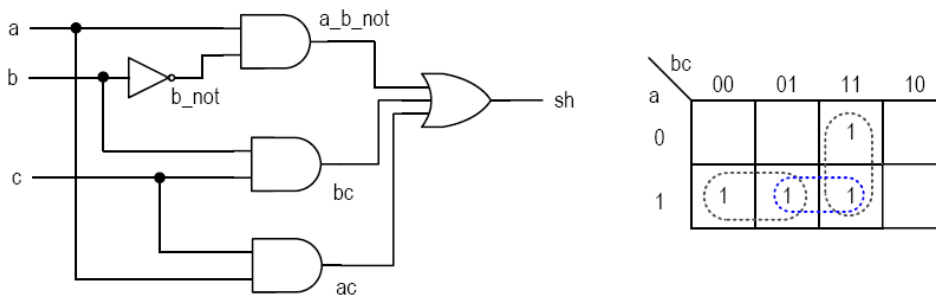


(a) Karnaugh map and schematic



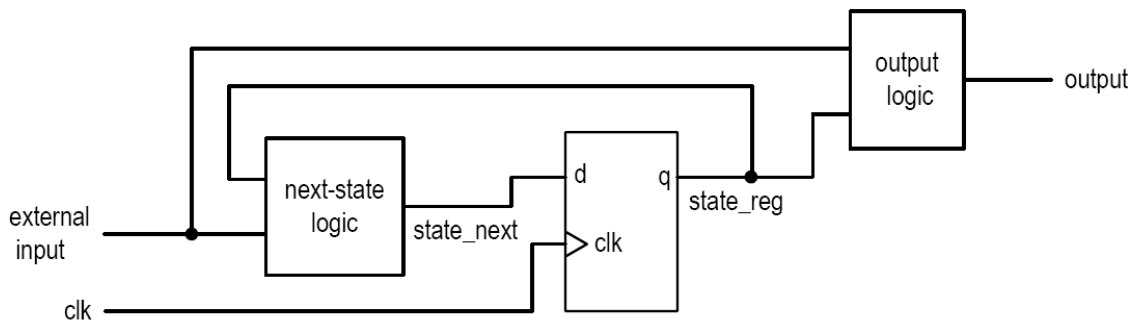
(b) Timing diagram

- Hazard động là một sự cố xảy ra trong thực tế ở tín hiệu ngõ ra khi chuyển từ mức cao sang mức thấp hoặc chuyển từ mức thấp sang mức cao có xét đến thời gian trễ, nhưng về mặt lý thuyết không xét đến thời gian trễ thì tín hiệu ngõ ra không có sự cố khi chuyển từ mức 0 sang mức 1 hoặc ngược lại.
- Cách hạn chế Hazard tĩnh:



(c) Revised Karnaugh map and schematic to eliminate hazards

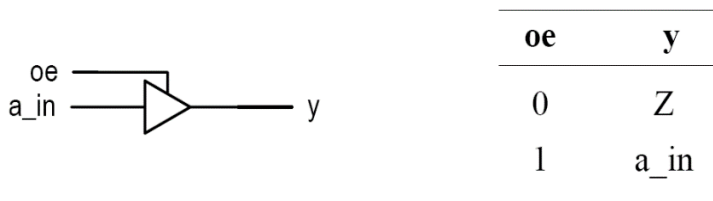
d. Hãy vẽ sơ đồ khối của mạch tuần tự đồng bộ và giải thích nguyên lý hoạt động của mạch tuần tự đồng bộ.



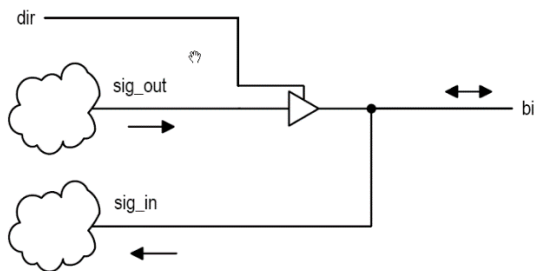
- Hệ thống đồng bộ bao gồm 3 khối chính:
 - o Mạch tổ hợp ngõ vào;
 - o Mạch tổ hợp ngõ ra;
 - o Mạch ghi nhận trạng thái;
- Nguyên lý hoạt động:
 - o Khi xuất hiện cạnh lên xung clk, tín hiệu state_reg được cập nhật giá trị của state_next và trở thành trạng thái mới;

- 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`;

e. Hãy vẽ ký hiệu cổng 3 trạng thái và cổng 3 trạng thái.

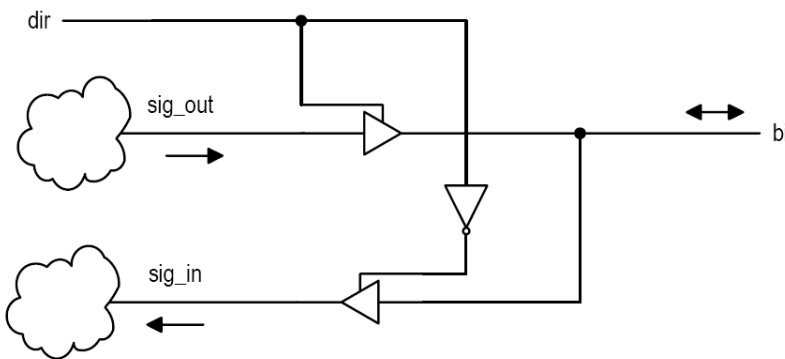


Hãy vẽ cho port 2 chiều dùng 1 cổng 3 trạng thái và viết Code VHDL tương ứng;



```
entity bi_demo is
    port (bi: inout std_logic;
          . . .
    );
begin
    sig_out <= output_expression;
    . . . <= expression_with_sig_in;
    . . .
    bi <= sig_out when dir='1' else 'Z';
    sig_in <= bi;
    . . .
end;
```

Hãy vẽ mạch cho port 2 chiều dùng 2 cổng 3 trạng thái và viết Code VHDL tương ứng.



```
entity bi_demo is
    port (bi: inout std_logic;
          . . .
    );
begin
    sig_out <= output_expression;
    . . . <= expression_with_sig_in;
    . . .
    bi <= sig_out when dir='1' else 'Z';
    sig_in <= bi when dir='0' else 'Z';
end;
```

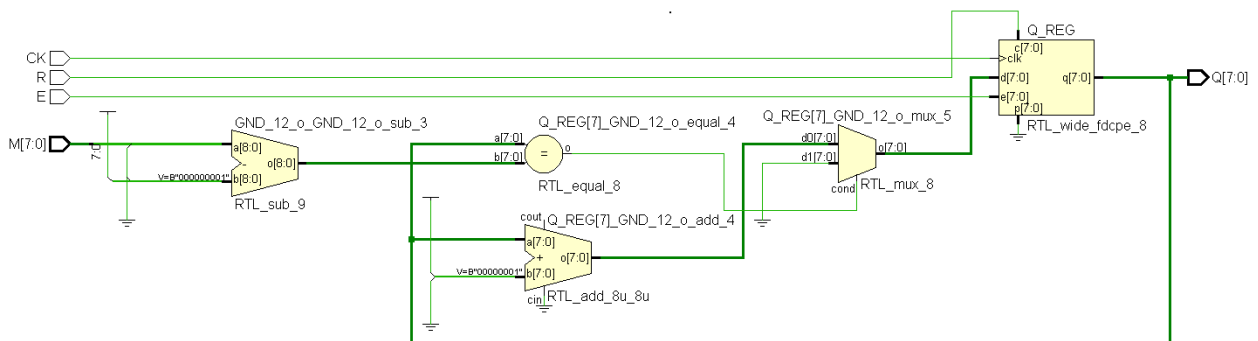
Câu 2: Hãy thiết kế mạch đếm lên 8 bit Mode M có thể lập trình: Các bước thực hiện: Vẽ sơ đồ khối, viết Code VHDL và vẽ sơ đồ khái niệm. (2.5đ)

library IEEE;

```

use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
entity Question_2 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;
architecture Behavioral of Question_2 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_REG = M - 1 ELSE
              Q_REG + 1;
    Q <= Q_REG;
end Behavioral;

```



Câu 3: Hãy viết chương trình dịch trái logic (logic shift) hoặc xoay trái logic (left rotation) được lựa chọn bởi tín hiệu Ctrl 1 bit (Ctrl = '0' thì thực hiện Logic Shift, bằng 1 thì thực hiện Left rotation), dữ liệu ngõ vào để dịch là Y[11:0], tín hiệu AMT 3 bits sẽ lựa chọn số bit cần dịch từ 0 đến 7

a. Sử dụng 3 lệnh gán có lựa chọn để thực hiện theo giải pháp chưa cải tiến (1.5đ)

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity QUESTION_3_A is
    Port ( Y : in  STD_LOGIC_VECTOR (11 downto 0);
          AMT : in  STD_LOGIC_VECTOR (2 downto 0);
          CTRL : in  STD_LOGIC;
          Q : out  STD_LOGIC_VECTOR (11 downto 0));
end QUESTION_3_A;
architecture Behavioral of QUESTION_3_A is
    SIGNAL ROTATE, SHIFT : STD_LOGIC_VECTOR(11 DOWNTO 0) :=
(Others=>'0');
begin
    ROTATE <= Y WHEN AMT = "000" ELSE
    Y(10 DOWNTO 0)&Y(11) WHEN AMT = "001" ELSE

```

```

        Y(9 DOWNT0 0)&Y(11 DOWNT0 10) WHEN AMT = "010" ELSE
        Y(8 DOWNT0 0)&Y(11 DOWNT0 9) WHEN AMT = "011" ELSE
        Y(7 DOWNT0 0)&Y(11 DOWNT0 8) WHEN AMT = "100" ELSE
        Y(6 DOWNT0 0)&Y(11 DOWNT0 7) WHEN AMT = "101" ELSE
        Y(5 DOWNT0 0)&Y(11 DOWNT0 6) WHEN AMT = "110" ELSE
        Y(4 DOWNT0 0)&Y(11 DOWNT0 5);
    SHIFT <= Y WHEN AMT = "000" ELSE
        Y(10 DOWNT0 0)&'0' WHEN AMT = "001" ELSE
        Y(9 DOWNT0 0)&"00" WHEN AMT = "010" ELSE
        Y(8 DOWNT0 0)&"000" WHEN AMT = "011" ELSE
        Y(7 DOWNT0 0)&"0000" WHEN AMT = "100" ELSE
        Y(6 DOWNT0 0)&"00000" WHEN AMT = "101" ELSE
        Y(5 DOWNT0 0)&"000000" WHEN AMT = "110" ELSE
        Y(4 DOWNT0 0)&"0000000";
    Q <= ROTATE WHEN CTRL = '1' ELSE
        SHIFT;
end Behavioral;

```

b. Sử dụng 2 lệnh gán có lựa chọn để thực hiện theo giải pháp cải tiến (1.5đ)

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity QUESTION_3_B is
    Port ( Y : in  STD_LOGIC_VECTOR (11 downto 0);
          AMT : in  STD_LOGIC_VECTOR (2 downto 0);
          CTRL : in  STD_LOGIC;
          Q : out  STD_LOGIC_VECTOR (11 downto 0));
end QUESTION_3_B;
architecture Behavioral of QUESTION_3_B is
    SIGNAL ROTATE, SHIFT : STD_LOGIC_VECTOR(11 DOWNT0 0) :=
(Others=>'0');
    SIGNAL DATA : STD_LOGIC_VECTOR(11 DOWNT0 0) := (Others=>'0');
begin
    DATA <= Y WHEN CTRL = '1' ELSE
        (Others=>'0');
    Q <= Y WHEN AMT = "000" ELSE
        Y(10 DOWNT0 0)&DATA(11) WHEN AMT = "001" ELSE
        Y(9 DOWNT0 0)&DATA(11 DOWNT0 10) WHEN AMT = "010" ELSE
        Y(8 DOWNT0 0)&DATA(11 DOWNT0 9) WHEN AMT = "011" ELSE
        Y(7 DOWNT0 0)&DATA(11 DOWNT0 8) WHEN AMT = "100" ELSE
        Y(6 DOWNT0 0)&DATA(11 DOWNT0 7) WHEN AMT = "101" ELSE
        Y(5 DOWNT0 0)&DATA(11 DOWNT0 6) WHEN AMT = "110" ELSE
        Y(4 DOWNT0 0)&DATA(11 DOWNT0 5);
end Behavioral;

```

Câu 4: Hãy thiết kế và mô tả dùng VHDL mạch giải mã từ 3 sang 8 có ngõ vào I[2:0], chân ngõ vào cho phép E tích cực mức cao, ngõ ra Y[7:0], chân ngõ vào S chọn mức tích cực ngõ ra: Khi S = '0' ngõ ra tích cực mức cao, S = '1' ngõ ra tích cực mức thấp. (1.5đ)

```

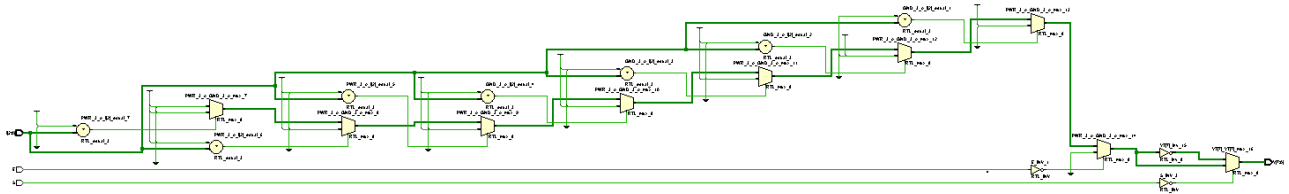
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity QUESTION_4 is
    Port ( E, S : in  STD_LOGIC;
          I : in  STD_LOGIC_VECTOR (2 downto 0);
          Y : out  STD_LOGIC_VECTOR (7 downto 0));
end QUESTION_4;
architecture Behavioral of QUESTION_4 is
    SIGNAL YT : STD_LOGIC_VECTOR(7 DOWNT0 0) := (Others=>'0');
begin

```

```

YT <= "00000000" WHEN E = '0' ELSE
      "00000001" WHEN I = "000" ELSE
      "00000010" WHEN I = "001" ELSE
      "00000100" WHEN I = "010" ELSE
      "00001000" WHEN I = "011" ELSE
      "00010000" WHEN I = "100" ELSE
      "00100000" WHEN I = "101" ELSE
      "01000000" WHEN I = "110" ELSE
      "10000000";
Y <= YT WHEN S = '0' ELSE
     NOT (YT);
end Behavioral;

```



Width	VHDL operator									
	nand	xor	> _a	> _d	=	+1 _a	+1 _d	+ _a	+ _d	mux
Area (gate count)										
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
Delay (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 1, 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 2

Tp. Hồ Chí Minh, ngày 30 tháng 05 năm 2016
 Thông qua bộ môn