TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT THÀNH PHỐ HỒ CHÍ MINH

KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN TỬ TRUYỀN THÔNG

ĐÁP ÁN ĐỀ THI CUỐI KỲ HỌC KỲ I NĂM HỌC 2016-2017

Môn: Thiết kế vi mạch số dùng HDL
Mã môn học: DSIC330563
Đề số/Mã đề: 01. Đề thi có 2 trang.
Thời gian: 90 phút.
Không được phép sử dụng tài liệu.

Câu 1: (2 điểm)

Sinh viên chọn 2 trong 5 câu sau đây:

a. Định nghĩa Synthesis trong thiết kế vi mạch. Quá trình Synthesis bao gồm những bước nào và cho ví dụ minh họa từng bước bằng thiết kế mạch Mux từ 2 sang 1 sử dụng ASIC Gate-Array NAND.

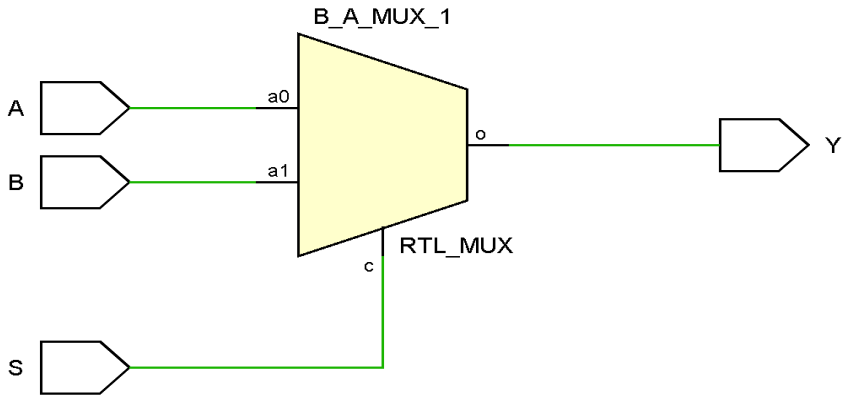
Quá trình Synthesis bao gồm những bước sau đây:

- RTL level synthesis: Là bước chuyển đổi từ ngôn ngữ HDL sang sơ đồ khối Logic, hay còn gọi là sơ đồ khái niệm.

Ví dụ mạch mux từ 2 sang 1 có chương trình mô tả dùng HDL như sau:

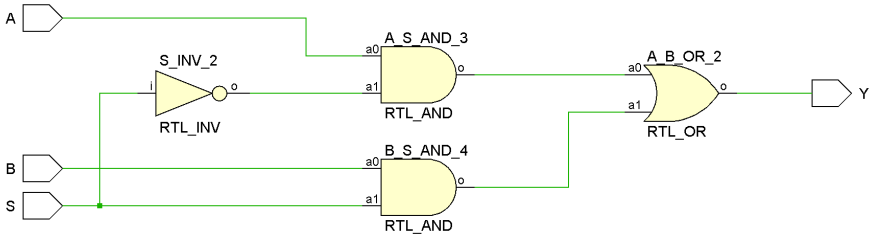
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity MUX4To1 is
    Port ( A, B : in  STD_LOGIC;
           Y : out  STD_LOGIC;
           S : in  STD_LOGIC);
end MUX4To1;
architecture Behavioral of MUX4To1 is
begin
    Y <= A WHEN S = '0' ELSE B;
end Behavioral;
```

Sơ đồ khái niệm (RTL Diagram) được vẽ như hình bên dưới:



Screen clipping taken: 25/12/2016 09:28

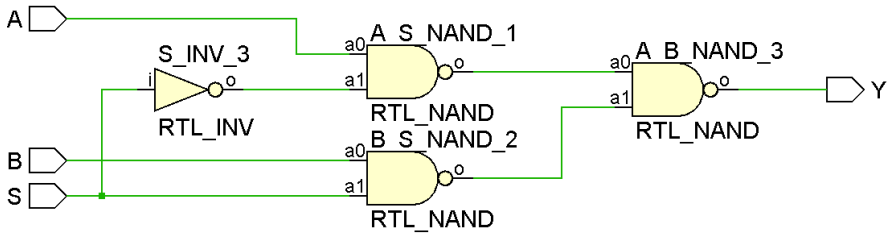
- Logic Synthesis: Là bước chuyển đổi từ sơ đồ khối logic (sơ đồ khái niệm) thành sơ đồ logic (logic diagrams). Mạch Mux từ 2 sang 1 được chuyển sang sơ đồ Logic như sau:



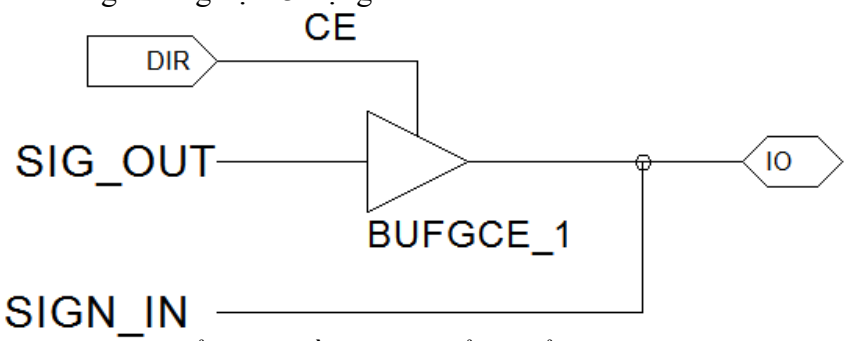
Screen clipping taken: 25/12/2016 09:40

- Techonlogy mapping: Là bước chuyển đổi từ sơ đồ logic sang sơ đồ kết nối của những linh kiện tương ứng được hỗ trợ bởi những công nghệ thiết bị khác nhau. Trong trường hợp này, sơ đồ logic của mạch Mux từ 2 sang 1 được chuyển sang sơ đồ mạch chỉ dùng cổng NAND như sau:

  $Y = A\bar{B} + BS = \overline{A\bar{S}}\ \overline{BS}$. Như vậy sơ đồ logic được vẽ như sau:



b. Vẽ cấu trúc cổng 2 chiều dùng 1 cổng đệm 3 trạng thái. Viết VHDL mô tả cấu trúc 2 chiều này.



Chương trình VHDL mô tả cổng 2 chiều dùng 1 cổng điểm 3 trạng thái:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity BIDIRECTION_1_TRISTATE is
    Port ( IO : inout  STD_LOGIC;
DIR : IN STD_LOGIC);
end BIDIRECTION_1_TRISTATE;

architecture Behavioral of BIDIRECTION_1_TRISTATE is
SIGNAL SIG_IN, SIG_OUT : STD_LOGIC;
begin
     IO <= SIG_OUT WHEN DIR = '1' ELSE
     'Z';
     SIG_IN <= IO;
end Behavioral;
```

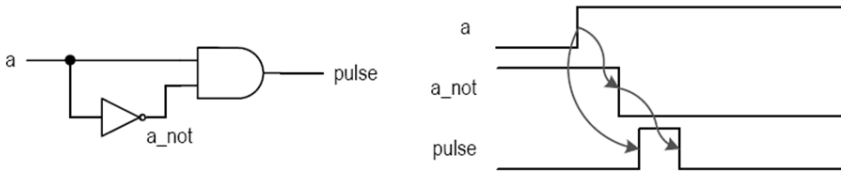
c. Định nghĩa Hazard, Hazard tĩnh, Hazard động. Cho một ví dụ về ứng dụng của hiện tượng Hazard tĩnh. Có thể thực hiện mô tả mạch ứng dụng này bằng VHDL được hay không, vì sao?

Hazard là hiện tượng xảy ra trong khoảng thời gian trễ của hệ thống được định nghĩa là sự mất ổn định ở ngõ ra của mạch trước khi ổn định ở một trạng thái mà nó phải có.

Hazard tĩnh là sự mất ổn định xảy ra ở ngõ ra về mặt lý thuyết là ổn định.

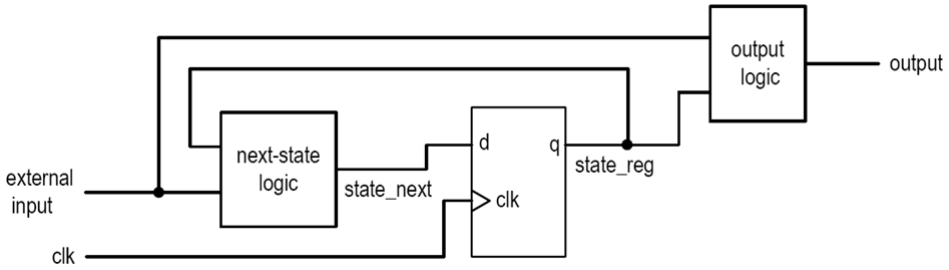
Hazard động là sự bất ổn định xảy ra ở ngõ ra trong quá trình chuyển trạng thái. Về mặt lý thuyết, ngõ ra chuyển trạng thái từ 0 đến 1 hay từ 1 về 0 không có trạng thái trung gian.

Hiện tượng Hazard tĩnh được ứng dụng để làm mạch tạo xung cạnh lên:



Không thể mô tả mạch phát hiển xung cạnh lên trực tiếp bằng VHDL vì VHDL chỉ mô tả mạch từ lớp Logic Level trở lên, không mô tả được lớp Physical Level. Hiện tượng Hazard chỉ được quan tâm ở lớp Physical Level.

d. Vẽ cấu trúc mạch đồng bộ 3 thành phần, nêu nguyên lý hoạt động và chức năng của từng khối.



Giải thích chức năng từng khối:

- Khối Nex-State logic: Có chức năng tạo trạng thái mới dựa vào các tín hiệu ngõ vào và trạng thái hiện tại;
- Khối output logic: Có chức năng tạo ra tín hiệu ở ngõ ra dựa vào trạng thái hiện tại và các tín hiệu ngõ vào điều khiển;
- Mạch FlipFlop D: Có chức năng cập nhật trạng thái hiện tại bằng trạng thái mới;

Giải thích nguyên lý hoạt động: Ban đầu mạch đang ở trạng thái hiện tại, ngõ ra được gán một giá trị xác định bởi mạch tổ hợp ngõ ra dựa vào trạng thái hiện tại state_reg và ngõ vào điều khiển. Mạch tổ hợp ngõ vào tạo trạng thái mới state_next ở ngõ vào d của Flip Flop. Khi có cạnh lên clk tác động vào chân clk của Flip Flop, trạng thái hiện tại state_reg được cập nhật bằng trạng thái mới state_next. Quá trình tiếp tục khi có xung cạnh lên clk tiếp theo.

e. Hãy nêu những nhược điểm của phần mềm EDA và cho ví dụ.

Những nhược điểm của phần mêm EDA:

- Phần mêm tổng hợp mạch HDL không biết được ý định của người thiết kế. Đặc biệt, những phép toán cấp thấp logic phần mềm EDA có thể hiểu được và tối ưu tốt hơn, những phép toán cấp cao khác như +, -, *, … phần mêm EDA không thể hiểu được và không thể tối ưu hiệu quả;
- Phần mềm tổng hợp EDA không thể đạt được sự tối ưu toàn cục, chỉ đạt được sự tối ưu cục bộ;
- Để đạt được sự tối ưu toàn cục, người thiết kế phải tối ưu cấu trúc mạch sao cho sử dụng ít những phép toán cấp cao, ưu tiên sử dụng những phép toán logic để phần mềm có thể tối ưu cục bộ hiệu quả hơn.

Câu 2: (2 điểm)

Trong thiết kế vi mạch số có 3 công nghệ thường được sử dụng, FPGA, Gate-Array và standard cells. Chi phí tài nguyên để sản xuất 1 IC khi sử dụng FPGA, Gate Array và standard cell tương ứng là 20$, 5$, và 2$. Chi phí để sản xuất Mask ban đầu khi sử dụng Gate array và standard cell tương ứng là 20.000$ và 100.000$.

a. Giả sử rằng số lượng IC được sản xuất ra là N. Viết biểu thức tính chi phí cho mỗi IC tương ứng cho 3 công nghệ trên.

Biểu thức tính chi phí sản xuất theo số lượng N cho mỗi sản phẩm được sản xuất ra theo mỗi công nghệ như sau:

Biểu thức chi phí khi dùng công nghệ FPGA:

$$C_{FPGA} = 20\ (USD)$$

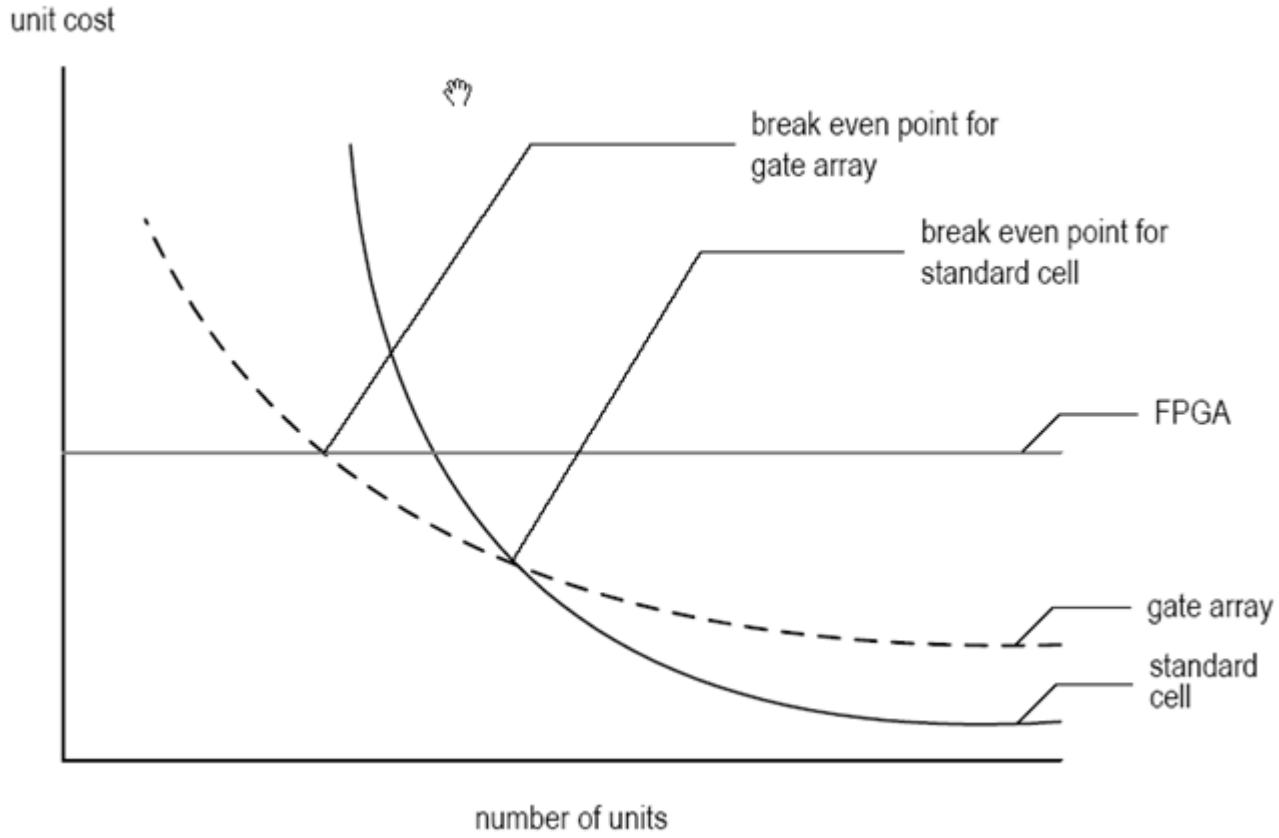
Biểu thức chi phí khi dùng công nghệ Gate_Array:

$$C_{Gatearray} = 5 + \frac{20000}{N} \text{ (USD)}$$

Biểu thức chi phí khi dùng công nghệ Standard Cell:

$$C_{StandardCell} = 2 + \frac{100000}{N} \text{ (USD)}$$

b. Vẽ biểu đồ chi phí của mỗi IC theo số lượng sản phẩm bán ra tương ứng cho mỗi công nghệ được sử dụng.



a. Xác định số lượng sản phẩm tối đa để sử dụng công nghệ FPGA sao cho chi phí là thấp nhất.

$$1 \leq N \leq 1.334 \text{ (Sản phẩm)}$$

b. Xác định số lượng sản phẩm tối đa để sử dụng công nghệ gate array sao cho chi phí là thấp nhất.

$$1.335 \leq N \leq 26.667 \text{ (Sản phẩm)}$$

c. Xác định số lượng sản phẩm tối đa để sử dụng công nghệ standard cell sao cho chi phí là thấp nhất.

$$N \geq 2668 \text{ (Sản phẩm)}$$

Câu 3: (3 điểm)

Thiết kế mạch ALU theo bảng sự thật sau:

CTRL[7:0]	R[7:0]
0	A + B
63	C - D
127	Unsigned(A) > Unsigned(B)

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

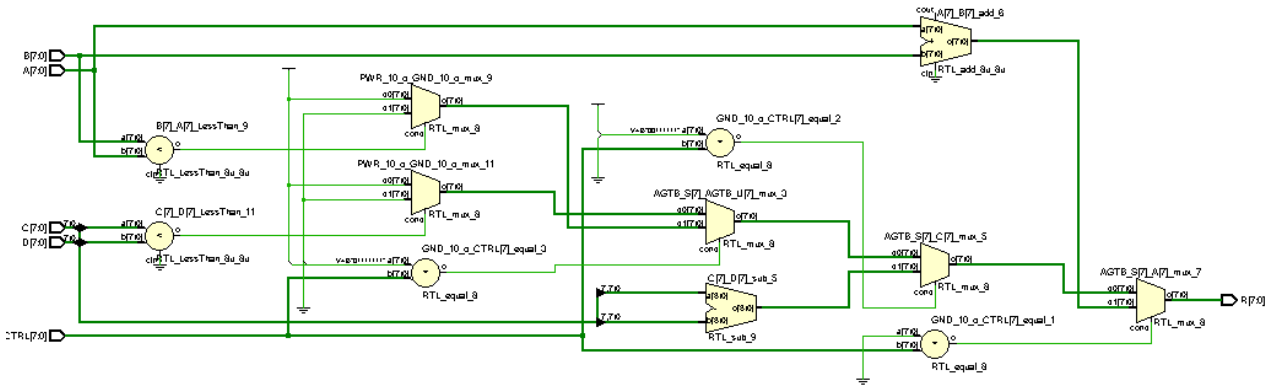
255	Signed(C) > Signed(D)

**Lưu ý: Khi ngõ ra R[7:0] là biểu thức Boolean, nếu biểu thức Boolean đúng thì ngõ ra bằng "00000000", ngược lại thì bằng "11111111".

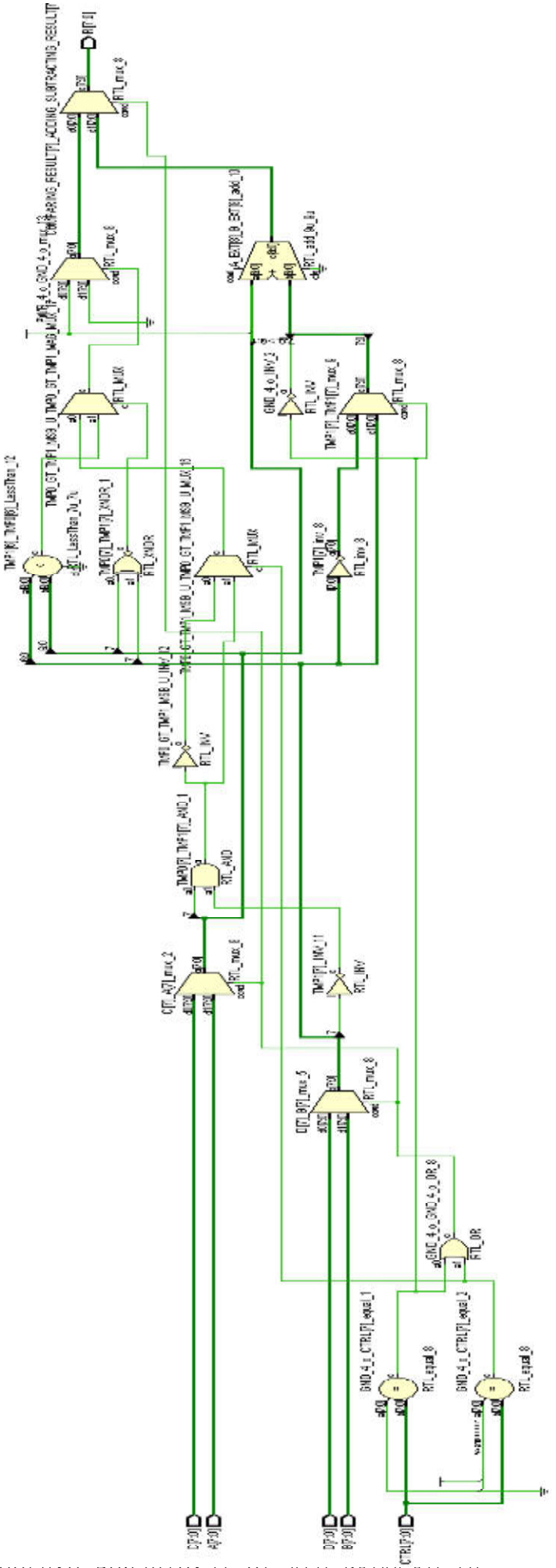
a. Viết VHDL mô tả mạch theo yêu cầu trên sao cho sử dụng 1 mạch "+", 1 mạch "-", 1 mạch so sánh ">" theo Unsigned, 1 mạch so sánh ">" theo Signed.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
entity ALU is
    Port ( A, B, C, D : in  STD_LOGIC_VECTOR (7 downto 0);
           CTRL : in  STD_LOGIC_VECTOR (7 downto 0);
           R : out  STD_LOGIC_VECTOR (7 downto 0));
end ALU;
architecture Behavioral of ALU is
SIGNAL AGTB_U, AGTB_S : STD_LOGIC_VECTOR(7 DOWNTO 0);
begin
R <= STD_LOGIC_VECTOR(SIGNED(A) + SIGNED(B)) WHEN CTRL = "00000000" ELSE
     STD_LOGIC_VECTOR(SIGNED(C) - SIGNED(D)) WHEN CTRL = "00111111" ELSE
     AGTB_U WHEN CTRL = "01111111" ELSE AGTB_S;
AGTB_U <= "00000000" WHEN UNSIGNED(A) > UNSIGNED(B) ELSE
     "11111111";
AGTB_S <= "00000000" WHEN SIGNED(C) > SIGNED(D) ELSE "11111111";
end Behavioral;
```

b. Vẽ sơ đồ khái niệm câu a.



c. Tối ưu sơ đồ khái niệm câu b sao cho chỉ sử dụng 1 mạch "+" và 1 mạch so sánh ">" theo unsigned.



d.

```
li
us
us
en
                                                    downto 0);
                                                     0);
                                                    ));
en
architecture Behavioral of ALU_REDUCED is
```

```
--SIGNAL FOR ADDING AND SUBTRACTING
SIGNAL A_EXT, B_EXT, SUM_EXT : STD_LOGIC_VECTOR(8 DOWNTO 0);
SIGNAL TMP0, TMP1, TMP1T : STD_LOGIC_VECTOR(7 DOWNTO 0);
SIGNAL T : STD_LOGIC;
SIGNAL ADDING_SUBTRACTING_RESULT : STD_LOGIC_VECTOR(7 DOWNTO 0);

--SIGNAL FOR COMPARING THE SIGNED AND UNSIGNED NUMBERS
SIGNAL TMP0_GT_TMP1_MSB_U, TMP0_GT_TMP1_MAG : STD_LOGIC;
SIGNAL TMP0_GT_TMP1 : STD_LOGIC;
SIGNAL COMPARING_RESULT : STD_LOGIC_VECTOR(7 DOWNTO 0);
begin
-- FOR MUXING THE INPUT VALUES
TMP0 <= A WHEN CTRL = "00000000" OR CTRL = "01111111" ELSE
C;
TMP1 <= B WHEN CTRL = "00000000" OR CTRL = "01111111" ELSE
D;

-- FOR REDUCING THE ADDING AND SUBTRACTING
T <= '0' WHEN CTRL = "00000000" ELSE '1';
TMP1T <= TMP1 WHEN CTRL = "00000000" ELSE NOT (TMP1);
A_EXT <= TMP0 & '1';
B_EXT <= TMP1T & T;
SUM_EXT <= STD_LOGIC_VECTOR(SIGNED(A_EXT) + SIGNED(B_EXT));
ADDING_SUBTRACTING_RESULT <= SUM_EXT(8 DOWNTO 1);

-- FOR COMPARING SIGNED AND UNSIGNED NUMBER
TMP0_GT_TMP1_MSB_U <= '1' WHEN TMP0(7) = '1' AND TMP1(7) = '0' ELSE
'0';
TMP0_GT_TMP1_MAG <= '1' WHEN TMP0(6 DOWNTO 0) > TMP1(6 DOWNTO 0) ELSE
'0';
TMP0_GT_TMP1 <= TMP0_GT_TMP1_MAG WHEN TMP0(7) = TMP1(7) ELSE
TMP0_GT_TMP1_MSB_U WHEN CTRL = "01111111" ELSE
NOT TMP0_GT_TMP1_MSB_U;
COMPARING_RESULT <= "00000000" WHEN TMP0_GT_TMP1 = '1' ELSE
"11111111";

-- FOR GETTING THE RESULT
R <= ADDING_SUBTRACTING_RESULT WHEN CTRL = "00000000" OR CTRL = "01111111"
ELSE
COMPARING_RESULT;
end Behavioral;
```

Câu 4: (3 điểm)

Thiết kế mạch đếm đồng bộ dùng cấu trúc 3 thành phần theo yêu cầu sau:

CTRL[1:0]	Chức năng
00	Mạch dịch phải logic 1 bit
01	Mạch dịch trái logic 1 bit
10	Mạch đếm Mode 5
11	Mạch đếm Mode 10

Mạch	Thời gian trễ
Mux 2-1 (4 bits)	2 ns
Mux 4-1 (4 bits)	2.5 ns
Adder (4 bits) "+"	4.5 ns
Subtractor (4 bits) "-"	5 ns
D Flip Flop	0.5 ns
Comparator (4 bits) "="	1 ns

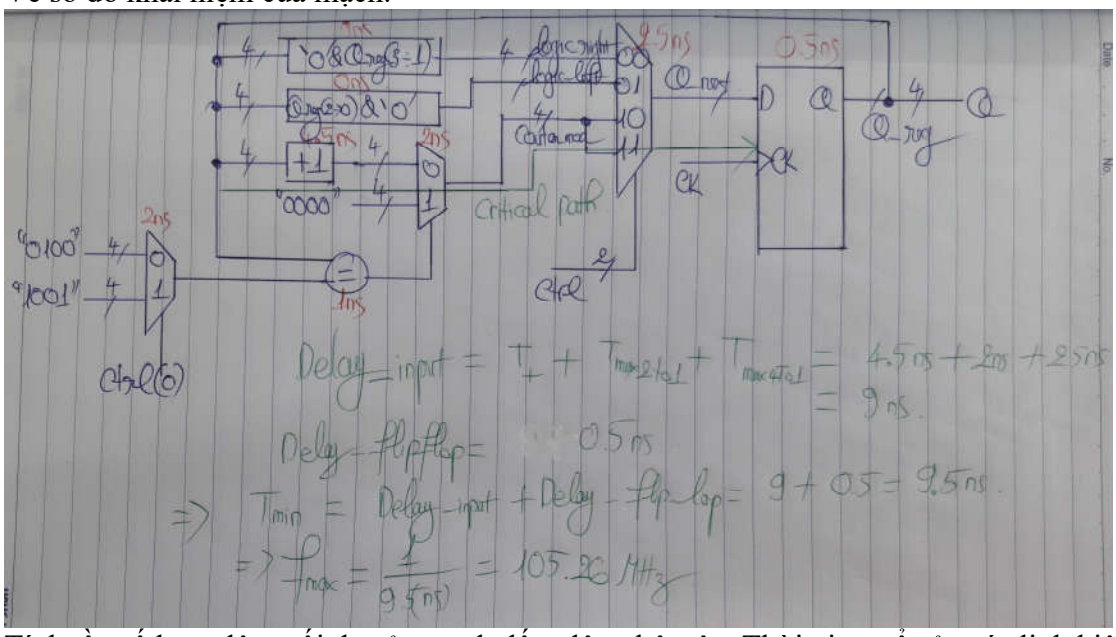
a. Viết VHDL mô tả mạch theo chức năng được mô tả.

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

entity COUNTERSHIFTER is
```

```
    Port ( CLK : in  STD_LOGIC;
           CTRL : in  STD_LOGIC_VECTOR (1 downto 0);
           Q : out  STD_LOGIC_VECTOR (3 downto 0));
end COUNTERSHIFTER;
architecture Behavioral of COUNTERSHIFTER is
SIGNAL Q_NEXT, Q_REG : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
SIGNAL LOGIC_LEFT, LOGIC_RIGHT : STD_LOGIC_VECTOR(3 DOWNTO 0);
SIGNAL MODE, COUNTER_MOD : STD_LOGIC_VECTOR(3 DOWNTO 0);
begin
-- D FLIP FLOP
PROCESS(Q_NEXT, CLK)
BEGIN
     IF RISING_EDGE(CLK) THEN
          Q_REG <= Q_NEXT;
     END IF;
END PROCESS;
-- INPUT LOGIC
PROCESS(Q_REG, MODE, CTRL)
BEGIN
     CASE CTRL IS
          WHEN "00" =>
               Q_NEXT <= '0'&Q_REG(3 DOWNTO 1);
          WHEN "01" =>
               Q_NEXT <= Q_REG(2 DOWNTO 0)&'0';
          WHEN OTHERS =>
               IF Q_REG = MODE THEN
                    Q_NEXT <= "0000";
               ELSE
                    Q_NEXT <= Q_REG + 1;
               END IF;
     END CASE;
END PROCESS;
MODE <= "0100" WHEN CTRL(0) = '0' ELSE "1001";
-- OUTPUT LOGIC
Q <= Q_REG;
end Behavioral;
```

b. Vẽ sơ đồ khái niệm của mạch.



c. Tính tần số hoạt động tối đa của mạch đếm động bộ trên. Thời gian trễ của các linh kiện được cho trong bảng trên đây.

$$T_{InputDelay} = T_{adder} + T_{mux2to1} + T_{mux4to1} = 4.5 + 2 + 2.5 = 9ns$$

$$T_{FlipFlop} = 0.5ns$$

$$T_{Min} = T_{InputDelay} + T_{FlipFlop} = 9 + 0.5 = 9.5ns$$

$$F_{Max} = \frac{1}{T_{Min}} = \frac{1}{9.5ns} = 105.26MHz$$

Lưu ý: Phần tính toán tần số cực đại của hệ thống thay đổi tùy theo cấu trúc thiết kế ban đầu của mạch. Câu này được tính là đúng khi thực hiện tính toán hợp lý theo đúng nguyên tắc tính tần số cực đại đã học.

Ghi chú: Cán bộ coi thi không được giải thích đề thi.

Chuẩn đầu ra của học phần (về kiến thức)	**Nội dung kiểm tra**
[CĐR 1.1]: Trình bày được các công nghệ thiết bị, quy trình thiết kế. Mô tả tổng quát, mô tả cấu trúc, mô tả hành vi, mô phỏng mạch. Trình bày tổng quan về ngôn ngữ mô tả phần cứng VHDL. Trình bày được cấu trúc ngôn ngữ cơ bản của VHDL.	Câu 1a, 1b, 1c, 1d, 1e Câu 2
[CĐR 2.1]: Hàm Big-O dùng để tính toán tài nguyên và hiệu suất của mạch theo ngõ vào. Phân tích, giải thích mạch điện của các toán tử sử dụng trong ngôn ngữ lập trình VHDL, tìm hàm Big-O. Phân tích thời gian trễ hệ thống, thời gian nguy hiểm, cách khắc phục.	Câu 1c Câu 3
[CĐR 2.2]: Phân tích chia sẻ toán tử, phân tích chia sẻ chức năng, phân tích bố trí linh kiện trong thiết kế để tìm cách tối ưu về tài nguyên, tối ưu về thời gian trễ. Phân tích và so sánh tài nguyên, thời gian trễ của mạch trước và sau khi tối ưu.	Câu 3
[CĐR 2.3]: Phương pháp thiết kế mạch tuần tự đồng bộ, thiết kế các mạch tuần tự cơ bản. Tính toán đáp ứng tần số cực đại của các mạch tuần tự.	Câu 4

Ngày 03 tháng 12 năm 2016
Thông qua Trưởng ngành

Ths. Nguyễn Ngô Lâm