

BỘ MÔN KTMT	MÔN : Thiết kế IC, IT4251	LẦN:....cuối kì....	ĐỀ
	LỚP :...62138.....	STT:.....	
	HỌ VÀ TÊN :.....	STT:.....	
	Thời gian làm bài: <b>60 phút</b> Ngày thi: 25/12/2013 Được phép sử dụng tài liệu Mỗi câu hỏi có duy nhất một đáp án đúng. Không trừ điểm khi trả lời sai. Đáp án phải được ghi vào phần trả lời mới hợp lệ, và nộp lại đề thi.		

## PHẦN TRẢ LỜI TRẮC NGHIỆM

Câu hỏi	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Trả lời															

## PHẦN CÂU HỎI TRẮC NGHIỆM (7.5 điểm)

**Câu 1:** Nhận xét nào đúng về đoạn mã trong Process dưới?

```

if Clear = '0' then
  Q <= '0';
else
  if falling_edge(CLK) then
    Q <= D;
  end if;
end if;

```

1. CLK tích cực sườn lên
2. Clear không đồng bộ theo CLK
3. Clear tích cực mức cao
4. Tất cả nhận xét trên đều sai

**Câu 2:** Giả sử rằng bảng LUT trong FPGA của Xilinx có 16 bit nhớ. Vậy bảng LUT này thực hiện được hàm logic có bao nhiêu biến đầu vào?

1. Thực hiện được hàm logic có <= 4 biến
2. Chỉ thực hiện được hàm logic có đúng 4 biến
3. Thực hiện được hàm logic có <= 16 biến
4. Chỉ thực hiện được hàm logic có đúng 16 biến

**Câu 3:**

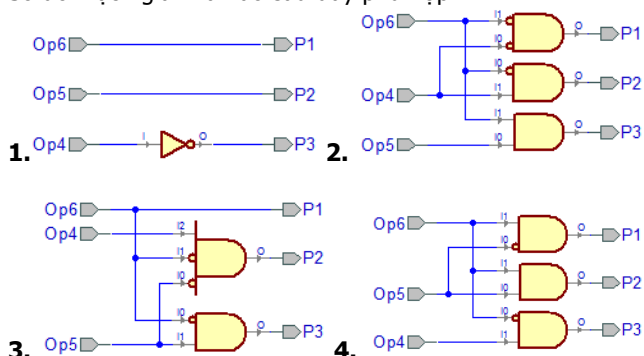
Các module mạch logic cho trước, được cung cấp sẵn, cho phép tích hợp trong các thiết kế bằng VHDL. Các module này có thể sử dụng, có thể cấu hình một số đặc tính, nhưng không có mã nguồn. Tên gọi chung của dạng module là gì?

1. IPCore
2. HardCores
3. FirmCores
4. SoftCores

**Câu 4:** Một bộ xử lý chỉ có 3 bit Opcode Op6, 5, 4 để giải mã thành 3 phép xử lý là P1, P2, P3. Với qui định Opcode như sau (Kí hiệu X là tùy ý):

Op6	Op5	Op4	Phép xử lý
1	X	X	P1
0	0	1	P2
0	1	X	P3

Sơ đồ mạch giải mã nào sau đây phù hợp?



**Câu 5:** Lệnh nào trong số các lệnh VHDL sau không thể được tổng hợp thành netlist để nạp lên FPGA được

1. A <= B & C;
2. A <= B + C;
3. A <= B after 8ns;
4. A <= B when (C='1');

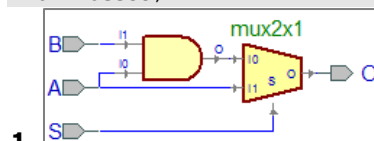
**Câu 6:** Đoạn mã sau ứng với sơ đồ nào bên dưới

Process (A, B, S)

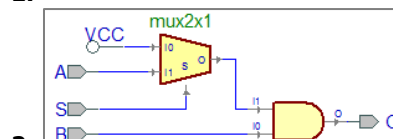
```

Begin
  If (S = '0') Then
    O <= A and B;
  Else
    O <= B;
  End If;
End Process;

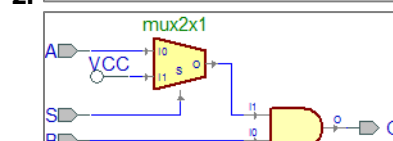
```



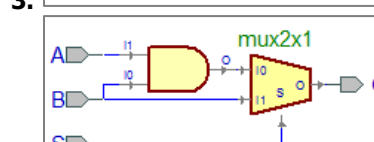
1.



2.

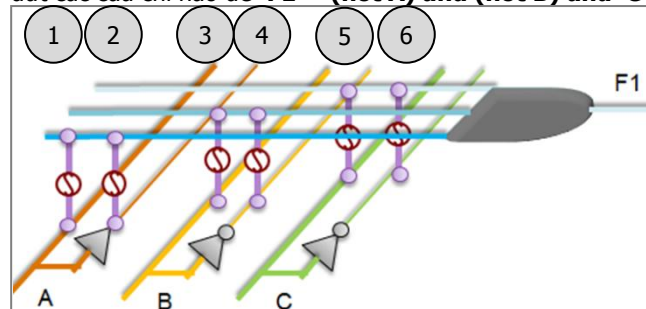


3.



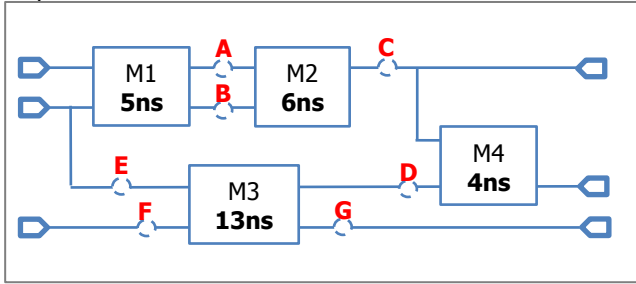
4.

**Câu 7:** Cho một phần mạch PAL như hình vẽ. Cần phải làm đứt các cầu chì nào để **F1 = (not A) and (not B) and C**



1. số 2, 4, 5
2. số 2, 4, 6
3. số 1, 3, 5
4. số 1, 3, 6

**Câu 8,9:** Cho sơ đồ và trễ trên từng module, bỏ qua trễ truyền lan.



**Câu 8:** Độ trễ của toàn mạch trên là bao nhiêu?

1. 15ns      2. 17ns      3. 18ns      4. 23ns

**Câu 9:** Với mạch trên, để thực hiện bằng pipeline 2 công đoạn, người ta đặt thêm thanh ghi với một D-FF ở vị trí điểm A. Hỏi thanh ghi đó cần có thêm D-FF ở các vị trí nào khác?

1. B, E, F      2. B, E, G      3. D, G      4. C, F

**Câu 10:** Trong FPGA, loại liên kết nối nào sau đây có số lượng ít nhất, cho phép truyền tín hiệu với tốc độ nhanh nhất?

1. Trực tiếp      2. Đa năng      3. Kết nối dài      4. Song song

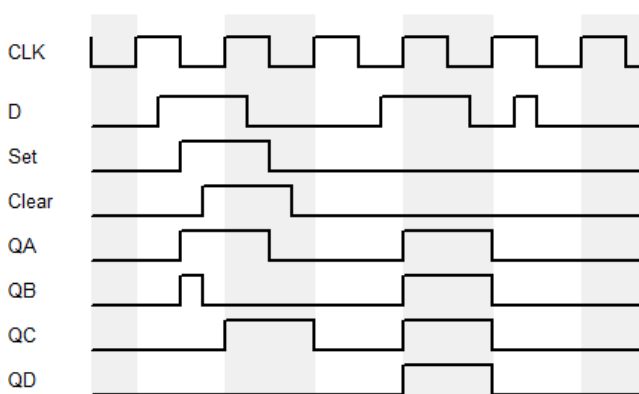
**Câu 11:** Lệnh nào sau đây đúng, để tạo đường đa tín hiệu 2 bit AB được ghép bởi 2 đường đơn tín hiệu A và B?

1. AB <= A : B      2. AB <= A & B  
3. AB <= A and B      4. AB <= A, B

**Câu 12:** Cho D-FF được mô tả như sau:

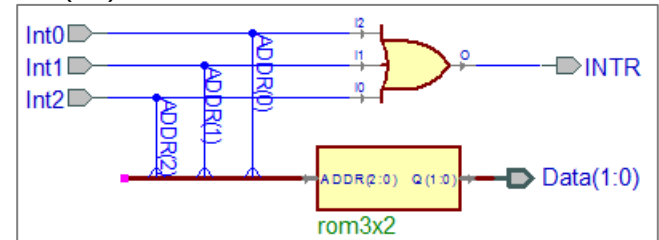
```
architecture DFF4 of DFF is
begin
  process (CLK, Set, Clear)
  begin
    if (Set = '1') then
      Q <= '1';
    elsif (Clear = '1') then
      Q <= '0';
    elsif (rising_edge(CLK) then
      Q <= D;
    end if;
  end process
end DFF4;
```

Tín hiệu ra Q phù hợp với tín hiệu nào trong giản đồ sau?



1. QA      2. QB      3. QC      4. QD

**Câu 13:** Thiết kế IC sau để mở rộng số lượng ngắt, cho phép khi 1 trong ngắt đầu vào Int0..Int2 tích cực mức cao, thì IC sẽ tích cực mức cao tín hiệu INTR, gửi ra bên ngoài. Và, để mạch bên ngoài biết được nguyên nhân ngắt là do ngắt nào trong số các Int0..Int2 thì đã có chân dữ liệu ra Data(1:0).



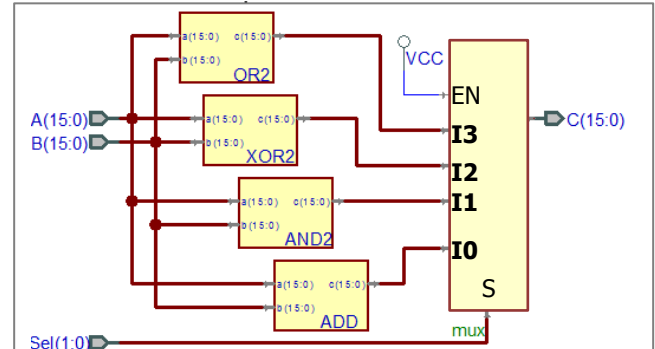
Biết rằng kiến trúc bên trong của rom2x4 là

```
process (ADDR)
begin
  case (ADDR) is
    when "111" => Q <= "00";
    when "110" => Q <= "00";
    when "101" => Q <= "00";
    when "100" => Q <= "11";
    when "011" => Q <= "00";
    when "010" => Q <= "10";
    when "001" => Q <= "01";
    when "000" => Q <= "00";
    when others => Q <= "ZZ";
  end case;
end process;
```

Hỏi ngắt nào là ngắt được ưu tiên nhất?

1. Int2      2. Int1      3. Int0      4. Bình đẳng

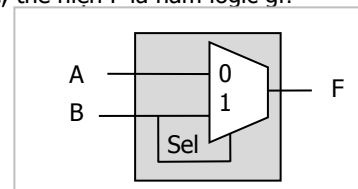
**Câu 14:** Với ALU được thiết kế như hình vẽ.



Cho biết giá trị các chân dữ liệu vào  $A=38_{(h)}$ ,  $B=15_{(h)}$ ,  $S=0$ . Tín hiệu ra C bằng bao nhiêu?

1. 10      2. 2D      3. 3D      4. 4D

**Câu 15:** Cho bộ dồn kênh ( $F = S.B + (\text{not } S).A$ ) được mắc như hình vẽ, thể hiện F là hàm logic gì?



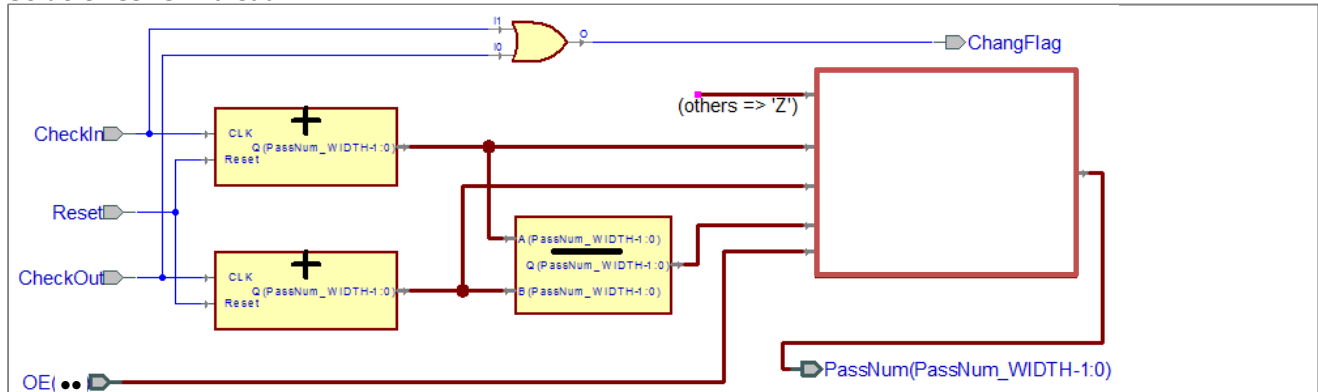
1. OR      2. AND      3. =0      4. XOR

## PHẦN CÂU HỎI TỰ LUẬN (2.5 điểm)

Thiết kế một IC có tên **PassengerCouter** để đếm số hành khách lên xuống xe buýt. Nguyên tắc hoạt động như sau:

- **Để phát hiện khách lên xe:** dùng một tia laser chiếu xuyên qua lối cửa lên của xe, và chạm vào cảm biến, làm cảm biến phát ra logic '0'. Khi có khách lên xe, khách sẽ chặn luồng sáng từ laser làm cảm biến sẽ phát ra logic '1'.
- **Để phát hiện khách xuống xe:** Áp dụng cơ chế toàn toàn tương tự như với cửa lên.
- Các tín hiệu từ cảm biến sẽ được gửi tới IC **PassengerCouter**. IC này có 2 bộ đếm, dùng để đếm số khách lên, và số khách xuống theo **sườn âm** của tín hiệu từ cảm biến.
- IC có tín hiệu ra là kết quả đếm số khách đã lên xe, số khách đã xuống xe, số khách còn lại trên xe, và **được dồn kênh** vào cùng một cổng ra.
- IC cũng tạo ra cờ ngắt báo hiệu đã có sự thay đổi khách trên xe, tích cực mức cao.

Sơ đồ thiết kế như sau:



Hãy điền **các lệnh còn thiếu** vào đoạn chương trình, và cho biết **module thiếu trên sơ đồ là gì**, các chân vào ra tương ứng ra sao? (Kí hiệu **••** là bỏ qua, không cần trả lời)

```
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.all;
entity PassengerCouter is
    generic (
        PassNum_WIDTH : integer := 11
    );
    port (
        Reset      : in STD_LOGIC; -- Khởi động lại quá trình đếm
        CheckIn    : in STD_LOGIC; -- Tín hiệu từ cảm biến của lên . =1 là bị che, có khách đang lên xe
        CheckOut   : in STD_LOGIC; -- Tín hiệu từ cảm biến của xuống. =1 là bị che, có khách đang xuống xe
        OE         : in STD_LOGIC_VECTOR(••); -- Tín hiệu chọn kênh dữ liệu ra.
        PassNum    : out STD_LOGIC_VECTOR(•••••); -- Số lượng hành khách
        ChangeFlag : out STD_LOGIC -- Cờ ngắt, báo hiệu có sự thay đổi về số lượng khách trên xe.
    );
end PassengerCouter;

architecture arch of PassengerCouter is
    signal InCounter, OutCounter : STD_LOGIC_VECTOR(•••••); -- lưu trữ kết quả đếm
begin
    -- Xử lý đếm số hành khách lên xe, và xuống xe
    COUNTER: (•••••)
    begin
        if Reset = '1' then
            InCounter <= (others => '0');
            OutCounter <= (others => '0');
        else
            if [ ] then
                InCounter <= InCounter + '1';
            end if;
            if [ ] then
                OutCounter <= OutCounter + '1';
            end if;
        end if;
    end process;

    -- Tạo cờ ngắt báo hiệu có sự thay đổi về số lượng hành khách trên xe
    ChangeFlag <= CheckIn or CheckOut;

    -- Dồn kênh xuất dữ liệu ra
    OUTPUT: process(InCounter, OutCounter, OE)
    begin
        case (CONV_INTEGER([ ])) is
            when 3 => PassNum <= OutCounter;
            when 2 => PassNum <= InCounter;
            when 1 => PassNum <= InCounter - OutCounter;
            when others => PassNum <= (others => 'Z');
        end case;
    end process;
end arch;
```