

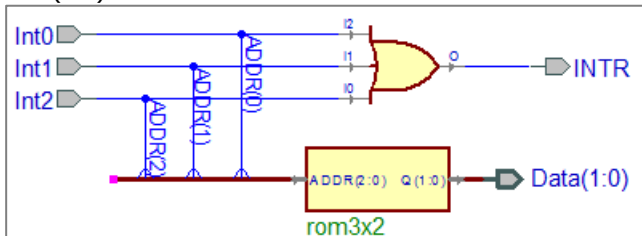
BỘ MÔN KTMT	MÔN : Thiết kế IC, IT4251	LẦN:....cuối kì....	ĐỀ
	LỚP :...62138.....	STT:.....	
HỌ VÀ TÊN :.....		Ngày thi: 25/12/2013	
Thời gian làm bài: <b>60 phút</b>		Được phép sử dụng tài liệu	
		Mỗi câu hỏi có duy nhất một đáp án đúng. Không trừ điểm khi trả lời sai.	
		Đáp án phải được ghi vào phần 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:** 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 "000" => Q <= "00";
    when "001" => Q <= "01";
    when "010" => Q <= "10";
    when "011" => Q <= "00";
    when "100" => Q <= "11";
    when "101" => Q <= "00";
    when "110" => Q <= "00";
    when "111" => 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 2:** Các module mạch logic cho trước, được cung cấp sẵn, cho phép tích hợp trong các thiết kế bằng VHDL. Các module này chỉ cho phép dùng luôn, không thể thay đổi được các đặc tính thiết kế. Tên gọi chung của dạng module là gì?

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

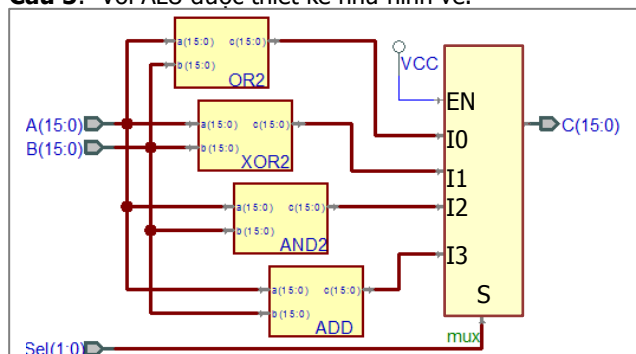
**Câu 3:** Trong FPGA, loại liên kết nối có số lượng nhiều nhất và sử dụng các ma trận chuyển mạch?

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

**Câu 4:** Có thể thiết kế bộ chia tần số div 17 bằng cách nào trong số các cách sau đây?

1. Dựa trên bộ đếm 4 bit  
2. Dựa trên thanh ghi dịch xoay vòng 5 bit  
3. Dựa trên bộ đếm 20 bit  
4. Dựa trên thanh ghi dịch xoay vòng 20 bit

**Câu 5:** Với ALU được thiết kế như hình vẽ.



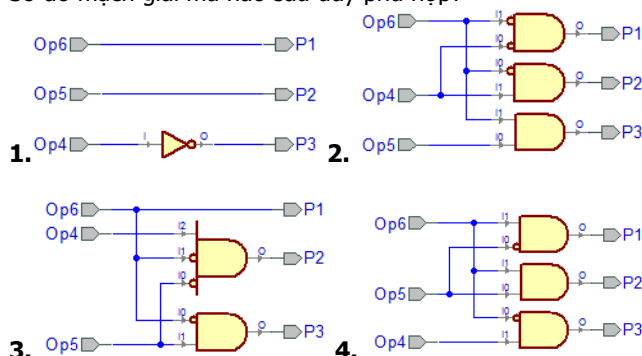
Cho biết giá trị các chân dữ liệu vào  $A=15_{(h)}$ ,  $B=38_{(h)}$ ,  $S=2$  Tín hiệu ra C bằng bao nhiêu?

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

**Câu 6:** 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
0	X	0	P1
0	X	1	P2
1	1	X	P3

Sơ đồ mạch giải mã nào sau đây phù hợp?



**Câu 7:** Lệnh nào trong số các lệnh VHDL sau không thể được tổng hợp thành netlist để nạp lên FPGA được

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

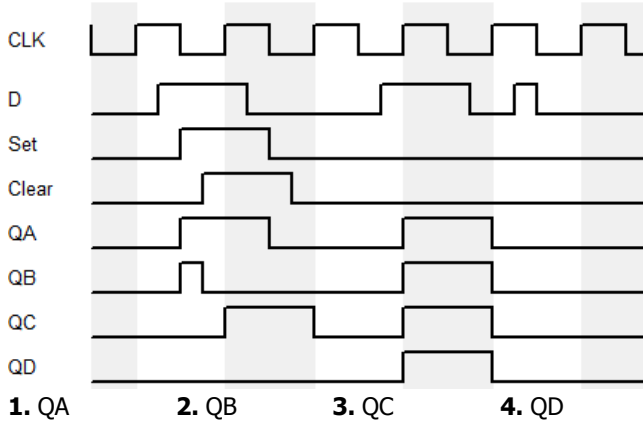
**Câu 8:** Lệnh nào sau đây đúng, để tạo đường đa tín hiệu 2 bit AB được ghép bởi 2 đường đơn tín hiệu A và B?

1.  $AB <= A : B$       2.  $AB <= A \& B$   
3.  $AB <= A + B$       4.  $AB <= A, B$

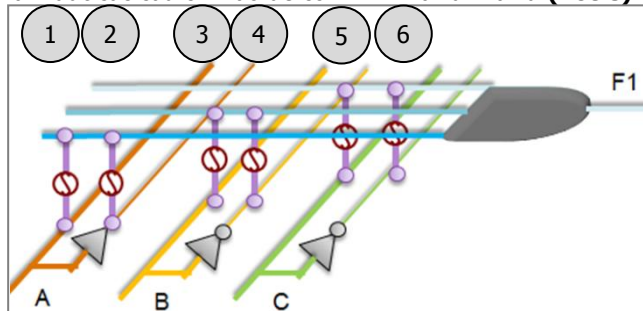
**Câu 9:** Cho D-FF được mô tả như sau:

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



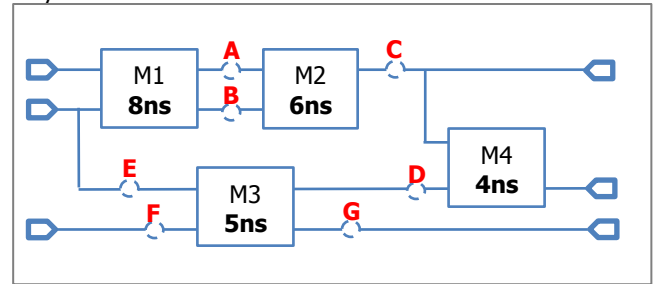
**Câu 10:** Cho một phần mạch PAL như hình vẽ. Cần phải làm đứt các cầu chì nào để có **F1 = A and B and (not C)**



**Câu 11:** Khai báo component trong VHDL có ý nghĩa gì

1. Tạo ra thể hiện một kiến trúc architecture
2. Định nghĩa các biến trong một kiến trúc architecture
3. Định nghĩa các cổng vào ra của thực thể
4. Tạo ra thể hiện một thực thể entity

**Câu 12,13:** Cho sơ đồ và trễ trên từng module, bỏ qua trễ truyền lan.



**Câu 12:** Độ trễ của toàn mạch trên là bao nhiêu?

1. 8ns      2. 17ns      3. 18ns      4. 23ns

**Câu 13:** Với mạch trên, thực hiện bằng pipeline 2 công đoạn, người ta đặt thêm thanh ghi với một D-FF ở vị trí điểm E. Hỏi thanh ghi đó cần có thêm D-FF ở các vị trí nào khác?

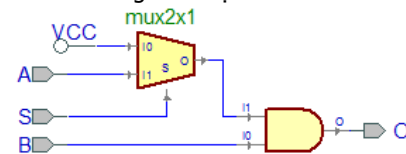
1. A, F      2. C, G      3. A, B      4. A, B, F

**Câu 14:** Nhận xét nào đúng về đoạn mã 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 đều đúng

**Câu 15:** Sơ đồ sau ứng với đoạn mã VHDL nào?



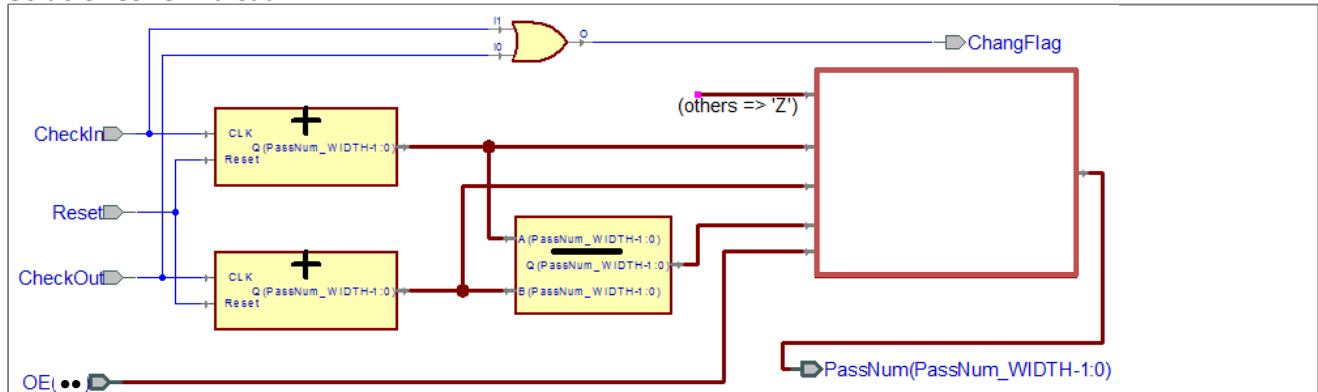
<b>1</b> Process (A,B,S) Begin If (S = '1') Then O <= A; Else O <= A and B; End If; End Process;	<b>2</b> Process (A,B,S) Begin If (S = '0') Then O <= A and B; Else O <= B; End If; End Process;
<b>3</b> Process (A,B,S) Begin If (S = '1') Then net <= '1'; Else net <= A; End If; End Process; O <= B and net;	<b>4</b> Process (A,B,S) Begin If (S = '0') Then net <= '1'; Else net <= A; End If; End Process; O <= B and net;

## PHẦN CÂU HỎI TỰ LUẬN (2.5 điểm)

Thiết kế một IC có tên **PassengerCounter** 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 **PassengerCounter**. IC này có 2 bộ đếm, dùng để đếm số khách lên, và số khách xuống theo sườn dương 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 PassengerCounter 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 cửa lên . =1 là bị che, có khách đang lên xe
        CheckOut   : in STD_LOGIC; -- Tín hiệu từ cảm biến cửa 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 PassengerCounter;

architecture arch of PassengerCounter 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 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 <= InCounter;
                when 2 => PassNum <= OutCounter;
                when 1 => PassNum <= InCounter - OutCounter;
                when others => PassNum <= (others => 'Z');
            end case;
        end process;
    end arch;
```