

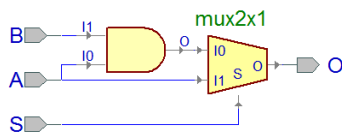
BỘ MÔN KTMT	MÔN : Thiết kế IC, IT4251 LỚP : ...62138..... HỌ VÀ TÊN :	LẦN: ...cuối kì.... STT:	ĐỀ
	Thời gian làm bài: 60 phút Ngày thi: 25/12/2013 Được phép sử dụng tài liệu Mỗi câu hỏi có duy nhất một đáp án đúng. Không trừ điểm khi trả lời sai. Đáp án phải được ghi vào phần trả lời mới hợp lệ, và nộp lại đề thi.		3163

PHẦN TRẢ LỜI TRẮC NGHIỆM

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

PHẦN CÂU HỎI TRẮC NGHIỆM (7.5 điểm)

Câu 1: Sơ đồ sau ứng với đoạn mã VHDL nào?



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

Câu 2: Nhận xét nào đúng về đoạn mã trong Process dưới?

```

if falling_edge(CLK) then
  if Clear = '1' then
    Q <= '0';
  else
    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 3: Giả sử rằng bảng LUT trong FPGA của Xilinx có 8 bit nhớ. Vậy bảng LUT này thực hiện được hàm logic có bao nhiêu biến đầu vào?

1. Chỉ thực hiện được hàm logic có đúng 3 biến
2. Thực hiện được hàm logic có <= 3 biến
3. Chỉ thực hiện được hàm logic có đúng 8 biến
4. Thực hiện được hàm logic có <= 8 biến

Câu 4: Nhận xét nào đúng?

1. Trong thiết kế IC, không có mã nguồn mở.
2. Ngôn ngữ HDL để tạo nên các phần mềm chạy trên bộ xử lý FPGA.

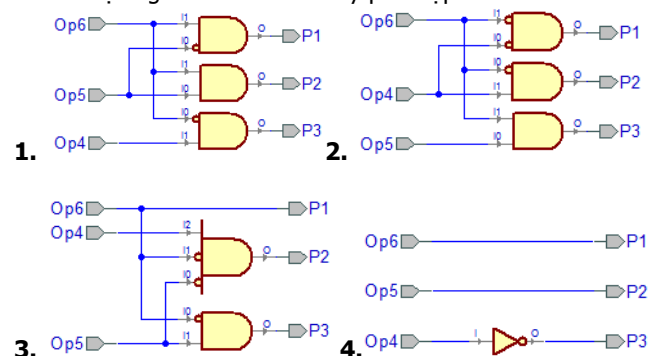
3. IC được thiết kế bởi ngôn ngữ HDL, có thể nạp lên chip FPGA để chạy thử.

4. FPGA là một loại vi điều khiển lập trình được bằng Pascal, C.

Câu 5: 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
X	1	X	P2
X	X	0	P3

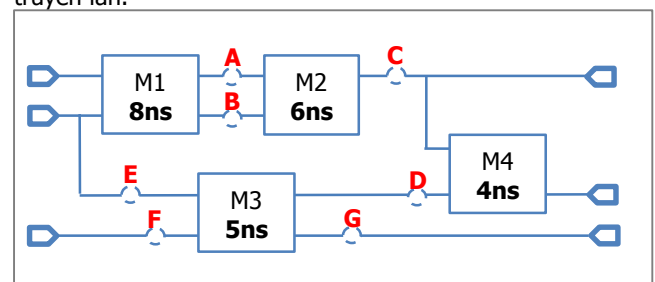
Sơ đồ mạch giải mã nào sau đây phù hợp?



Câu 6: Cần bao nhiêu D-FlipFlop để tạo thành thanh ghi vừa đủ để chứa số nguyên không dấu 2013?

1. 4
2. 11
3. 2013
4. Không cần

Câu 7,8: Cho sơ đồ và trễ trên từng module, bỏ qua trễ truyền lan.



Câu 7: Độ trễ của toàn mạch trên là bao nhiêu?

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

Câu 8: Với mạch trên, thực hiện bằng pipeline 2 công đoạn, người ta đặt thêm thanh ghi với một D-FF ở vị trí

điểm **B**. Hỏi thanh ghi đó cần có thêm D-FF ở các vị trí nào khác?

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

Câu 9: Trong FPGA, loại liên kết nối nào có trễ truyền lan thấp nhất và cũng có số lượng ít nhất?

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

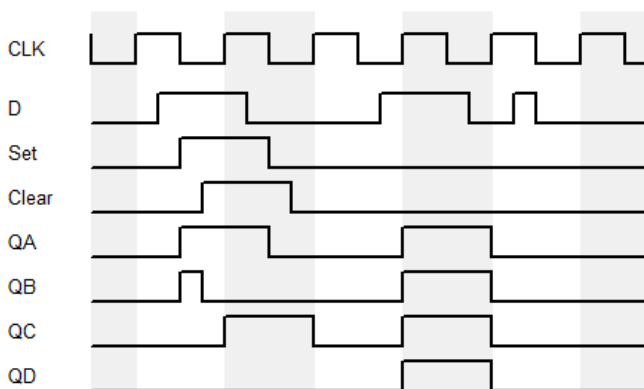
Câu 10: Lệnh nào sau đây đúng, để tạo đường đa tín hiệu 2 bit AB được ghép bởi 2 đường đơn tín hiệu A và B?

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

Câu 11: Cho D-FF được mô tả như sau:

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

Lời giải chi tiết:

Theo mã nguồn, ta thấy Set và Clear là 2 tín hiệu tích cực mức cao, mức 1.

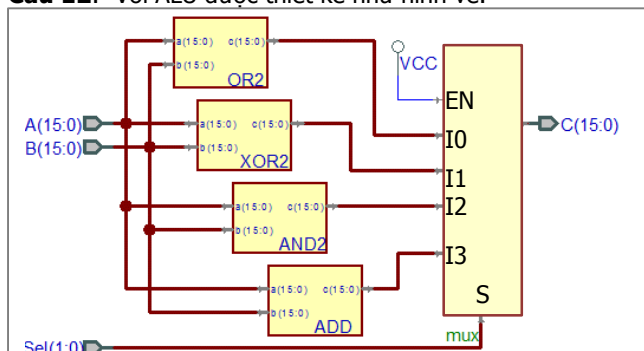
QA: Set. Clear không đóng bo. Set ưu tiên hơn. Không đồng bộ vì ta thấy khi Set=1, hoặc khi Set=0, Clear = 1 thì đầu ra Q thay đổi ngay lập tức, không bị phụ thuộc vào CLK. Và Set ưu tiên hơn vì ta thấy khi Set=1, thì dù Clear=1, QA vẫn bằng 1

QB: Set. Clear không đóng bo. Clear ưu tiên hơn.

QC: Set. Clear đóng bo. Set ưu tiên hơn.

QD: Set. Clear đóng bo. Clear ưu tiên hơn.

Câu 12: Với ALU được thiết kế như hình vẽ.

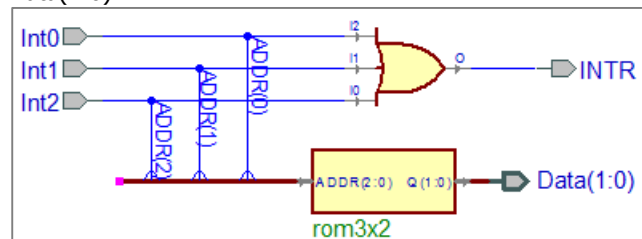


Thiết kế IC, IT4251

Cho biết giá trị các chân dữ liệu vào **A**=16_(h), **B**=3B_(h), **S**=1. Tín hiệu ra C bằng bao nhiêu?

1. 12 2. 3F 3. 51 4. 2D

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 "000" => Q <= "00";
    when "001" => Q <= "01";
    when "010" => Q <= "10";
    when "011" => Q <= "10";
    when "100" => Q <= "11";
    when "101" => Q <= "11";
    when "110" => Q <= "11";
    when "111" => Q <= "11";
    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

Lời giải chi tiết:

Kiểm tra xem bit xác định nguồn gây ngắt Q (hoặc Data) có giá trị nào xuất hiện nhiều nhất thì đó chính là địa chỉ của ngắt được ưu tiên nhất (khi ngắt đó xuất hiện thì mọi ngắt khác đều không bị bỏ qua). Ở đây Q <= "11" xuất hiện nhiều nhất, tương ứng khi ADDR(2)=1

Câu 14:

Cho đoạn mã VHDL thiết kế RAM

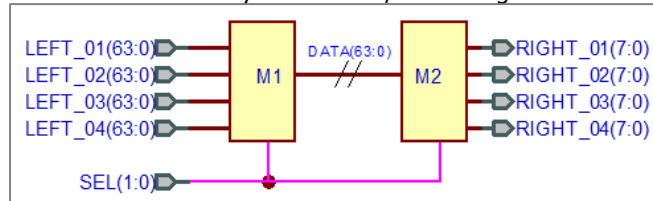
```
entity ram is
  port (
    WE : in std_logic;
    ADDR : in std_logic_vector(3 downto 0);
    DATA : in std_logic_vector(7 downto 0);
    Q : out std_logic_vector(7 downto 0)
  );
end entity;

architecture ram_arch of ram is
  type ram_mem_type is array (15 downto 0) of
    std_logic_vector(7 downto 0);
  signal ram_mem : ram_mem_type;
begin
  process (WE, ADDR, DATA)
  begin
    if (WE = '1') then
      ram_mem(CONV_INTEGER(ADDR)) <= DATA;
    end if;
  end process;
  Q <= ram_mem(CONV_INTEGER(ADDR));
end architecture;
```

Nhận xét nào sau đây sai?

1. Dữ liệu chỉ được ghi vào RAM khi tín hiệu WE bằng '1'
2. Khi thay đổi tín hiệu WE thì Q thay đổi theo
3. Khi thay đổi địa chỉ ADDR thì Q thay đổi theo
4. Sau khi ghi dữ liệu mới từ các chân DATA vào RAM, thì đầu ra Q cũng nhận ngay giá trị tương ứng từ DATA

Câu 15: Để truyền 4 đường tín hiệu 64 bit LEFT tới các đường tín hiệu RIGHT tương ứng, người ta sử dụng thêm 2 module M1 và M2. Hãy cho biết M1, và M2 là gì?



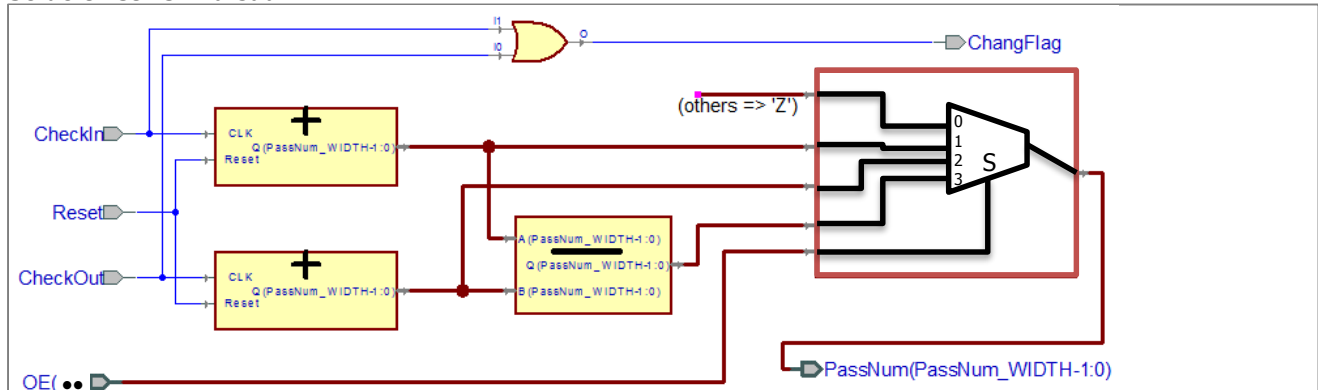
1. M1, M2 là bộ dồn kênh
2. M1 là dồn kênh, M2 là phân kênh
3. M1 là phân kênh, M2 là dồn kênh
4. M1 là phân kênh, M2 là phân kênh

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 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 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: process (Reset, CheckIn, CheckOut) -- Lời giải: tất cả tín hiệu vào có trong thân process
    begin
        if Reset = '1' then
            InCounter <= (others => '0');
            OutCounter <= (others => '0');
        else
            if falling_edge(CheckIn) then -- Lời giải: đề bài ghi rõ sườn âm
                InCounter <= InCounter + '1';
            end if;
            if falling_edge(CheckOut) then -- Lời giải: đề bài ghi rõ sườn âm
                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(OE)) is -- Lời giải: bộ dồn kênh ứng với khối trắng trên sơ đồ
            when 1 => PassNum <= InCounter;
            when 2 => PassNum <= OutCounter;
            when 3 => PassNum <= InCounter - OutCounter;
            when others => PassNum <= (others => 'Z');
        end case;
    end process;
end arch;
```