UNIVERSIDAD CARLOS III DE MADRID

MODULE 1 EXAM

Integrated Circuits and Microelectronics

Authors: Mario García, Marta Portela, Enrique San Millán, Almudena Lindoso, Celia López, Luis Entrena

Question 1 (30 min, 3 pts)

Given the following VHDL description:

```
1   entity question is
2     port (Clk:          in std_logic;
3           Reset:        in std_logic;
4           Enable:       in std_logic;
5           Up_down:      in std_logic;
6           Count:     out std_logic_vector(3 downto 0));
7   end question;
8
9   architecture a of cuestion is
10    signal sCount: std_logic_vector(3 downto 0));
11    signal a,b: std_logic;
12  begin
13    process(…)
14    begin
15      if Reset = ‘1’ then
16          sCount <= “0000”;
17          a <= ‘0’;
18      elsif clk'event and clk = ‘1’ then
19          if Enable = ‘1’ then
20            if Up_down = ‘1’ then
21              if sCount = “1001” then
22                sCount <= “0000”;
23                a <=’1’;
24              else
25                sCount <= sCount + '1';
26                a <=’0’;
27              end if;
28            else
29              if sCount = “0000” then
30                sCount = “1001”;
31                a <=’1’;
32              else
33                sCount = sCount – ‘1’;
34                a <=’0’;
35            end if;
36          end if;
37      end if;
38    end process;
39
40    Count <= sCount;
41
42    process(…)
43    begin
44      if sCount = “0111” then
45        b <=’1’;
46      else
47        b <=’0’;
48      end if;
49    end process;
50
51  end a;
```

a) Fill in the sensitivity lists of the processes with the correct signals.
b) Draw the circuit that is going to be synthesized from this VHDL code.
c) Modify the code to obtain the same circuit with generic size, with the following specifications:
    1. Count is a n-bit signal
    2. The last possible count value of the counter is “100…0001” (starts and ends with ‘1’, all zeros in the middle)
    3. b is activated when the count value is “0000….0011” (ends with two ones and the rest are zeros)

Note: For c) it is not necessary to rewrite all the code. Write only those lines with some changes, indicating the line number.

Problem 1 (1h 15 min, 7 pts)

We want to design an infrared receiver according to the RC5 protocol from Philips. RC5 code consists in sending 14-bits words, see Figure 1. The first two bits are start bits and they must be '1'. Next bit corresponds to a control bit and it is inverted each time that a new code is transmitted. This allows the receiver to distinguish if a button has been pressed in a remote control once or twice. The following 5 bits in the frame are used to identify the destination system (TV, video, etc.). The last 6 bits identify the command to be run.

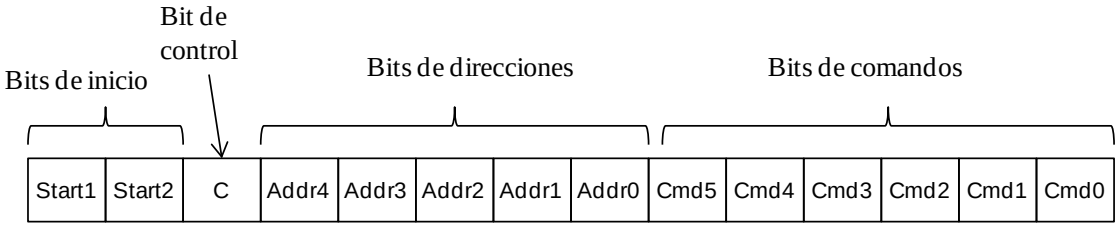

Figure 1. Format for an RC5 frame

Bits are transmitted by using a Manchester code, that is, every logical value is indicated with an edge, as Figure 2 shows.

Figure 2. Manchester code

Every bit lasts 1.778 ms and therefore a frame lasts 24.892 ms. The system clock works at 50MHz. Design a circuit capable of decoding a RC5 frame. The circuit interface is described in the following paragraph:

- Inputs:
  - Clk: clock signal
  - Reset: asynchronous initialization signal, active at high level.
  - Sin: 1bit signal generated by an infrared detector from the signal transmitted by the remote control.
- Output:
  - TV: active at high level only when the received command is addressed to a system with addr = 0.
  - Cmd: 6 bits. This output must show the last pressed command after it has been decoded.
  - Error: active at high level when there is a fault in the transmission when start bits have not been received correctly or when communication has been interrupted (signal edges are not received any more during a frame transmission).

A counter module is available and it can be used as a component in the system design. Its entity is described with the following code:

```
Entity counter is
Generic( n: integer );
Port(
  Clk     : in std_logic;--clock signal
  Reset  : in std_logic;-- asynchronous inizialization signal, active at high level
  Ena     : in std_logic;--enable
  Clr      : in std_logic; --synchronous inizialization signal, active at high level
  Count: out integer range 0 to 2**n-1);
End counter;
```

Answer the following questions:

a) In order to sample each bit, an edge detector module is needed. It should be capable of detecting edges that occur in the middle of a bit transmission and avoiding those that may appear on the boundary between two bits transmissions. Design the hardware needed to detect edges in the input Sin, only when an enable signal is active (*EnaEdge* = 1). This hardware must generate a one cycle pulse (*Edge*) whenever an edge is detected (either rising or falling, if *EnaEdge* = 1), and a signal for indicating the bit value (*ReadBit*) that depends on the type of edge (‘1’ for rising and ‘0’ for falling, as shown in Figure 2).

b) Since the bits are transmitted by serial, a shift register is required to store the frame bits. Design a shift register with an input *ReadBit* and an enable signal (*EnaReg*) to control when you load a new bit.

c) Specify how many counters you consider necessary and write down its/their instance/s. Explain its/their function.

d) Design a finite state machine that controls the different elements of the circuit and generate the necessary outputs.

e) Draw a diagram block at RT (Register Transfer) level of the circuit indicating the necessary interconnections.

f) Describe the complete circuit with VHDL. You can indicate with comments where is included the code developed in the previous steps.

g) Describe a VHDL test bench to verify the proper operation of the design. Perform self-checks for at least the following cases:

   a. When an error is produced due to a fault in the start bits. Simulation should not be stopped in this case.

   b. When an error is detected by an interruption in the transmission. In this cases, simulation must be halted.

Clear and optimal VHDL descriptions will be good considered.