DigitalSystems: exam

Renaud Pacalet

17 June 2016

You can use any document you need. Please number the different pages of your work and indicate on each page your first and last names. Write your answers in French or in English, as you wish, but avoid mixing the languages. If some extra information or hypotheses are missing to answer a question or solve a problem, decide by yourself and write down the added hypotheses or information. If you consider a question as absurd and thus decide not to answer, explain why. If you don't have time to answer a question or solve a problem but know how to, briefly explain your ideas.

The first part is a set of five questions (2 points each). The second and third parts are two small problems (5 points each).

Important advice #1: quickly go through the document and answer first the easy parts.

Important advice #2: copying verbatim the slides of the lectures or any other provided material is not considered a valid answer.

1 Questions

1.1. What is a logic optimizer? What kind of data does it use and what kind of data does it produce?

1.2. What happens when synthesizing a VHDL model where a combinational process does not assign every output in every situation?

1.3. `rd`, `req` and `ack` being bits (with values 0 or 1), translate the following property in CTL:

```
If the client requests the shared resource (req=1)
it cannot modify the signal rd before receiving the
acknowledge (ack=1).
```

1.4. The VHDL model shown on listing 1 is synthesizable. `clk`, `x`, `y` and `di` are signals of type `bit`. Draw a logic schematic of the circuit that a logic synthesizer would produce with such a model. If you do not know how to draw a logic gate, use a rectangular symbol and name it properly (`and3`, `dff...`).

```
process(clk)
    variable q0, q1, q2: bit;
begin
    if rising_edge(clk) then
        q2 := q1;
        q0 := di;
        q1 := q0;
        x <= q2;
    end if;
end process;

process(clk, di)
begin
    if clk = '1' then
        y <= di;
    end if;
end process;
```

Listing 1: Synthesizable model

1.5. We consider `INC`, a small digital circuit, represented on figure 1. `A` is a 16 bits primary input vector. `SEL` is a single bit primary input. `DO` is a 16 bits primary output vector. The circuit contains a 16 bits multiplexer, a 16 bits register (that samples its input on the rising edge of clock `CLK`) and a 16 bits incrementer (a simple combinatorial element that adds 1 to its 16 bits input and outputs the 16 least significant bits of the result). What is the minimum number of processes required to model this circuit in synthesizable VHDL? Explain your answer.

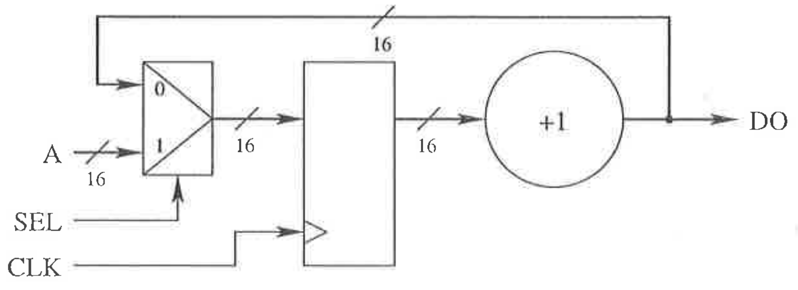

Figure 1: The `INC` digital circuit

2 John Cooley's design contest

This exercsie is directly derived from John Cooley's design contest at SNUG'95 (Synopsys Users Group meeting). The contest was intended to oppose VHDL and Verilog designers on the same design problem. What John had in mind was probably to determine what language was the most efficient. The results were that 8 out of the 9 Verilog designers managed to complete the design contest yet none of the 5 VHDL designers could. Hopefully you will do a much better job.

2.1. Design in plain synthesizable VHDL (entity and architecture) a synchronous up-by-3, down-by-5, loadable, modulus 512 counter, with carry output, borrow output and parity output. The interface specification of the counter is given in table 1. The block diagram of the counter is represented on figure 2. The counter is a 9 bits unsigned counter so it ranges between 0 and 511. When counting up beyond its maximum value or when counting down below its minimum value it wraps around according table 2.

Name	Size	Direction	Description
CLOCK	1	Input	Master clock; the counter is synchronized on the rising edge of CLOCK
DI	9	Input	Data input bus; the counter is loaded with DI when UP and DOWN are both low
UP	1	Input	Up-by-3 count command; when UP is high and DOWN is low the counter increments by 3, wrapping around its maximum value (511)
DOWN	1	Input	Down-by-5 count command; when DOWN is high and UP is low the counter decrements by 5, wrapping around its minimum value (0)
CO	1	Output	Carry out signal; high only when counting up beyond the maximum value (511) and thus wrapping around
BO	1	Output	Borrow out signal; high only when counting down below the minimum value (0) and thus wrapping around
DO	9	Output	Output bus; the current value of the counter; when UP and DOWN are both high the counter retains its value
PO	1	Output	Parity out signal; high when the current value of the counter contains an even number of 1's

Table 1: Interface specification of the counter

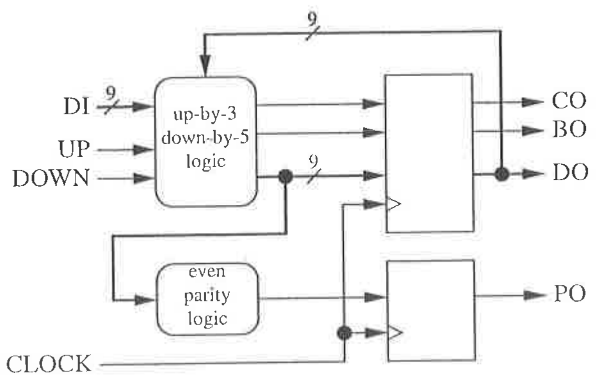

Figure 2: The block diagram of the counter

2.2. When synthesizing your design with a logic synthesizer how many memory elements will be inferred?

3 Mealy and Moore state machines

Consider the device represented on figure 3.

Counter current value	UP DOWN	Counter next value	next CO	next BO	next PO
x	00	DI	0	0	$parity(DI)$
x	11	x	0	0	$parity(x)$
$0 \leq x \leq 508$	10	$x+3$	0	0	$parity(x+3)$
509	10	0	1	0	1
510	10	1	1	0	0
511	10	2	1	0	0
$5 \leq x \leq 511$	01	$x-5$	0	0	$parity(x-5)$
4	01	511	0	1	0
3	01	510	0	1	1
2	01	509	0	1	1
1	01	508	0	1	0
0	01	507	0	1	1

Table 2: Wrapping mechanism of the counter

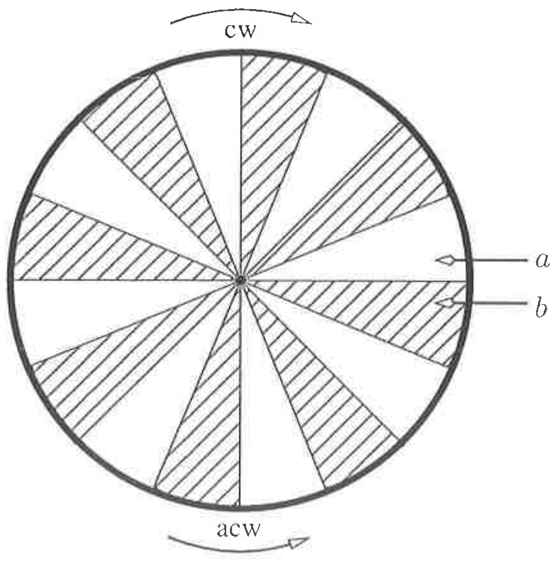

Figure 3: The rotating disk

a and b are two stationary copper brushes, touching the surface of the disk at the same distance of its center. The white segments of the disk are conducting segments while the striped segments are not. The disk rotates either clockwise (cw) or anti-clockwise (acw). During rotation the two copper wires a and b pass alternately over conducting and non-conducting segments. They are close enough to each other to fit in the same segment. By convention $a = 1$ when a is over a conducting segment and $a = 0$ when a is over a non-conducting segment. Same for b. The situation depicted on figure 3 is thus characterized by $a = 1$ and $b = 0$ but the 3 other combinations of a and b are also encountered during the rotation. Starting from the situation depicted on figure 3, if the disk rotates clockwise, the sequence of values taken by AB is $10, 11, 01, 00, 10\ldots$ While, if the disk rotates anti-clockwise, the sequence is $10, 00, 01, 11, 10\ldots$

The clock frequency of the state machines you will design is so high, compared to the rotating speed of the disk, that a and b cannot both change during the same clock period. Design the state diagram of a synchronous Mealy state machine with clock `clk`, asynchronous, active low reset `rstn`, two data inputs `a` and `b` connected to the copper brushes and one output `s`. Its functional specification is that `s='0'` when the reset is active (`rstn='0'`), else `s='1'` if the disk rotates clockwise and `s='0'` if the disk rotates anti-clockwise.

Write down the VHDL model of the architecture of this state machine. The entity is shown in listing 2.

```vhdl
library ieee;
use ieee.std_logic_1164.all;

entity dsm is
    port(clk, rstn, a, b: in  std_ulogic;
         s:               out std_ulogic);
end entity dsm;
```

Listing 2: Entity

Same questions but this time with a Moore state machine.