EDC: exam

Renaud PACALET

18 June 2013

You can use any document you need. On each page of your paper please indicate your name. Put your answers in any order but indicate first the corresponding question number. Example:

```
Question 2.1: I think this is a very good question and...
```

Advice: read every word very carefully and answer first the easy parts. You can write your answers in English or in French, as you like. Do not forget to attach your labs paperwork if you did not send it to me already or send them to me today, sharp deadline.

The five questions of the first part are worth 2 points each. If you consider a question as ambiguous explain why, then make the assumptions needed to solve the ambiguities and answer the modified question. If you consider a question as absurd explain why. If you find an error in a question explain why you think it is a error, propose a correction if it is reasonable and answer the modified question.

The second part is a small problem about median filters. Median filters are used in image processing as a cleaning tool to remove noise. It is very simple in principle but its implementation in a dedicated piece of hardware poses some interesting challenges. This part is worth 10 points.

1 Questions

1.1. What do you think of the following VHDL process? Is it synchronous or combinatorial? Identify the errors if any and, for each of them, explain why it is an error, what undesirable effect it has and finally, write down a new VHDL code with all the errors fixed.

```
process(clk_en, di, dsi, rstn)
begin
  if rstn = '0' then
    do <= (others => '0');
  elsif clk_en = '1' then
    if dsi = '1' then
      do <= di;
    end if;
  end if;
end process;
```

1.2. What are the VHDL resolved types, what are they used for, where should they be used, where should they not be used and why?

1.3. When designing a finite state machine in VHDL is it preferable to hard-code the different states with bit-strings or to use an enumerated type? Why?

1.4. What is the minimum number of processes required to model a Moore finite state machine[1]? Why? Same questions with a Mealy finite state machine?

1.5. Translate the following property from natural language to CTL: *If, while the END signal is de-asserted, the START input is asserted, it remains high until the END input is also asserted.*

2 Design of a median filter

The median filter is used to remove defects and noise from pictures. The algorithm is very simple : every pixel of the picture to be filtered is replaced by the median value of the neighbouring pixels. The picture is thus transformed by the median filter into another picture that has exactly the same size. For every pixel P of the input picture we first create a list of the 9 (3×3) pixels surrounding P[2]. The 9 pixels are then sorted. The median value is the value located in the middle of the sorted list. The pixel P in the filtered picture takes this median value. In our example the pictures are grey scale pictures, 8 bits per pixel. The pixel values are between 0 (black) and 255 (white).

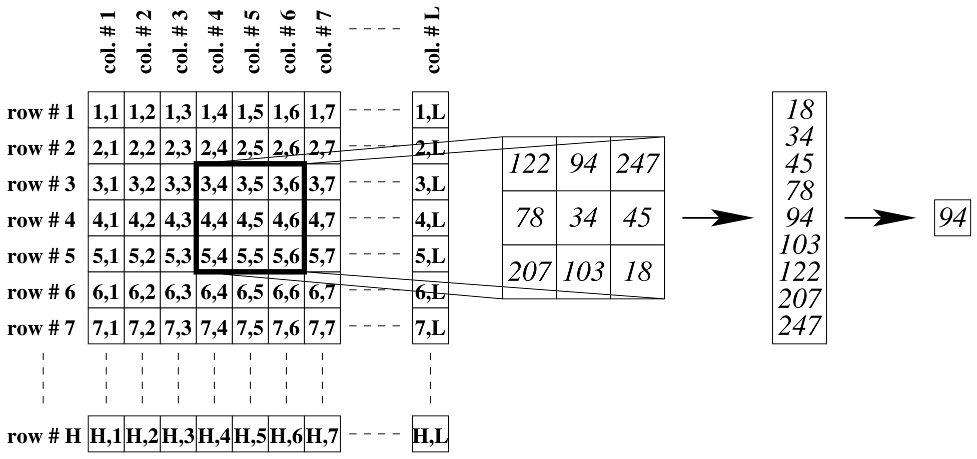

col. # 1
col. # 2
col. # 3
col. # 4
col. # 5
col. # 6
col. # 7
col. # L
row # 1
row # 2
row # 3
row # 4
row # 5
row # 6
row # 7
row # H
1,1 1,2 1,3 1,4 1,5 1,6 1,7 1,L
2,1 2,2 2,3 2,4 2,5 2,6 2,7 2,L
3,1 3,2 3,3 3,4 3,5 3,6 3,7 3,L
4,1 4,2 4,3 4,4 4,5 4,6 4,7 4,L
5,1 5,2 5,3 5,4 5,5 5,6 5,7 5,L
6,1 6,2 6,3 6,4 6,5 6,6 6,7 6,L
7,1 7,2 7,3 7,4 7,5 7,6 7,7 7,L
H,1 H,2 H,3 H,4 H,5 H,6 H,7 H,L
122 94 247
78 34 45
207 103 18
18 34 45 78 94 103 122 207 247
94

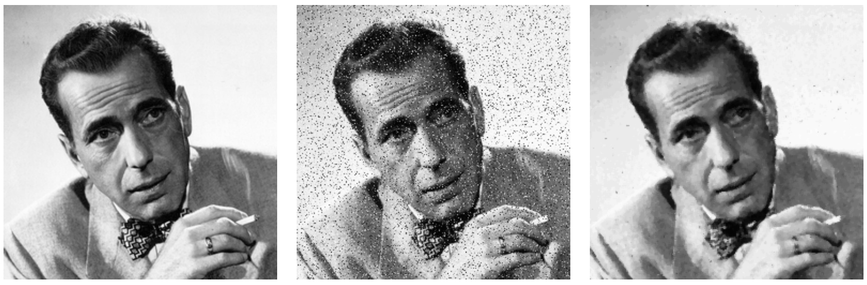
Figure 1: The principle of the median filter

Consider the situation depicted in figure 1. Let $[X, Y]$ be the pixel located at row X, column Y of a picture. The value of pixel $[4, 5]$ in the filtered picture is computed from the 3×3 neighborhood of the pixel $[4, 5]$ of the source picture, that is, the list of pixels $[3, 4]$, $[3, 5]$, $[3, 6]$, $[4, 4]$, $[4, 5]$, $[4, 6]$, $[5, 4]$, $[5, 5]$, $[5, 6]$. The list of the pixel values in our example is $122, 94, 247, 78, 34, 45, 207, 103, 18$. The sorted list is $18, 34, 45, 78, 94, 103, 122, 207, 247$. The median value is 94, so the value of the pixel

[1]The outputs of a Moore finite state machine depend on the current state only while the outputs of a Mealy finite state machine may depend on the current state and also on the inputs

[2]In most cases the pixels on the edges of the picture are simply repeated to replace the missing pixels in a 3×3 neighborhood

in the filtered picture will be 94. Figure 2 shows the effect of a 3×3 median filter on a picture of Humphrey Bogart.



(a) Original (b) Noisy (c) Filtered

Figure 2: Example of effect of a 3×3 median filter

The goal of this exercise is to design a module extracting the median value of a list of 9 pixels. This module could then be used as a coprocessor for a general purpose CPU to speed up a software implementation. It could also be integrated in a larger design that would implement the filter completely in hardware.

TODO (3 points): Design, in plain synthesizable VHDL (entity and architecture), a simple compare and exchange module (CE). The interface of the CE module is specified in figure 3 and table 1.

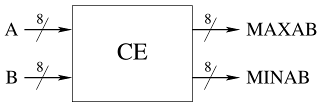


Figure 3: Interface specification of the CE module

Name	Size	Direction	Description
A	8	Input	Pixel value
B	8	Input	Pixel value
MAXAB	8	Output	The maximum of the two pixel values
MINAB	8	Output	The minimum of the two pixel values

Table 1: Interface specification of the CE module

The CE module is purely combinatorial. It compares the two input pixels and puts the largest of its two inputs on the MAXAB output and the other on its MINAB output.

To extract the median value of 9 pixels we will now design a median sorter module, MED. It is a sequential operator using one (and only one) CE module. Its interface is specified in figure 4 and table 2. The communications between MED and the enclosing system are summarized on figure 5.

The system inputs 9 pixels (P0, P1, ..., P8), one per clock period. During the 9 periods DSI is active. There is no interruption during this input phase: the 9 periods are consecutive. After the last pixel is input MED starts extracting the median value

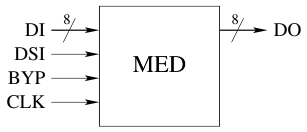


Figure 4: Interface specification of the MED module

Name	Size	Direction	Description
DI	8	Input	Data input bus
DSI	1	Input	Active high data strobe in. Indicate, when active, that DI carries a valid input pixel
BYP	1	Input	A command signal that changes the behavior of MED (see below)
CLK	1	Input	Clock. MED is synchronous on the rising edge of CLK
DO	8	Output	Data output bus. When the processing is over this output bus holds the resulting median value

Table 2: Interface specification of the MED module

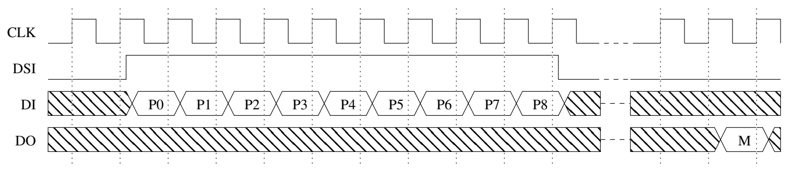


Figure 5: Waveforms of the MED I/O communications

and, when over, puts it on the DO bus. This computation takes several clock periods. After the processing is over and the result is output a new set of 9 pixels may be presented. We assume in the following that the surrounding system always uses this simple protocol. It never tries to input a new set before the previous computation is over. The architecture of MED is depicted on figure 6. MUX8 objects are 2 to 1 multiplexers of 8 bits words. R0, R1, ..., R8 are 8 bits registers.

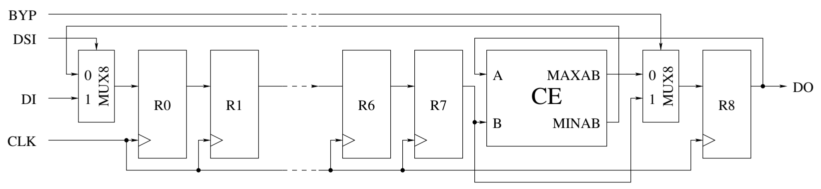


Figure 6: Internal architecture of the MED module

TODO (5 points): As you did for the CE module, design the MED module in plain synthesizable VHDL (entity and architecture).

TODO (2 points): Bonus question: study the internal architecture of MED and try to imagine how the external system extracts the median value by driving the DI, DSI and BYP inputs. Describe the sequence of operations, from the input of the 9 pixels to the output of the result. Assuming this module runs at a 500 MHz clock frequency, what is the maximum throughput (in pixels per second)? Would it be sufficient to filter in real time a video stream with 25 pictures per second, 720 $\times$ 576 pixels each?