CentraleSupélec

[illegible]

ICS M2 Degree - 2021-2022 - Advanced Digital Electronics

Practical Session N°2
Getting started with the Altera / Quartus tools
Realization of arithmetic operators

We want to create a 16-bit binary / decimal converter, i.e. a combinatorial function which receives as input a number val coded on 16 bits (therefore ranging from 0 to 65535) and which provides 5 digits (seg0 to seg4) which each contain, coded on 4 bits, the value of the different digits of the decimal writing of the number.

The conversion algorithm used is as follows
Seg0 = val mod 10
Val1 = val / 10

Seg1 = val1 mod 10
Val2 = Val1 / 10...

And so on up to val4 and seg4.

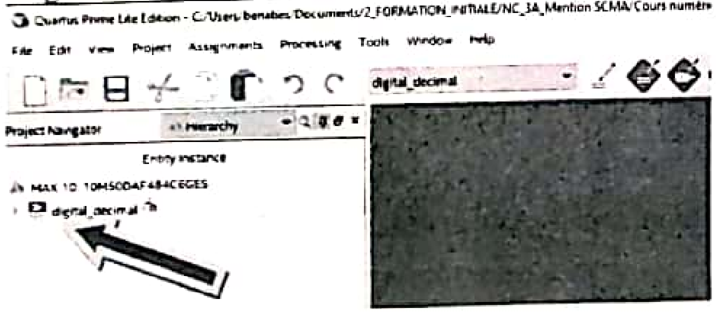
This function will be integrated into a project running in a quartus environment, and can be tested immediately with the de10-lite cards supplied.

1°) Project opening

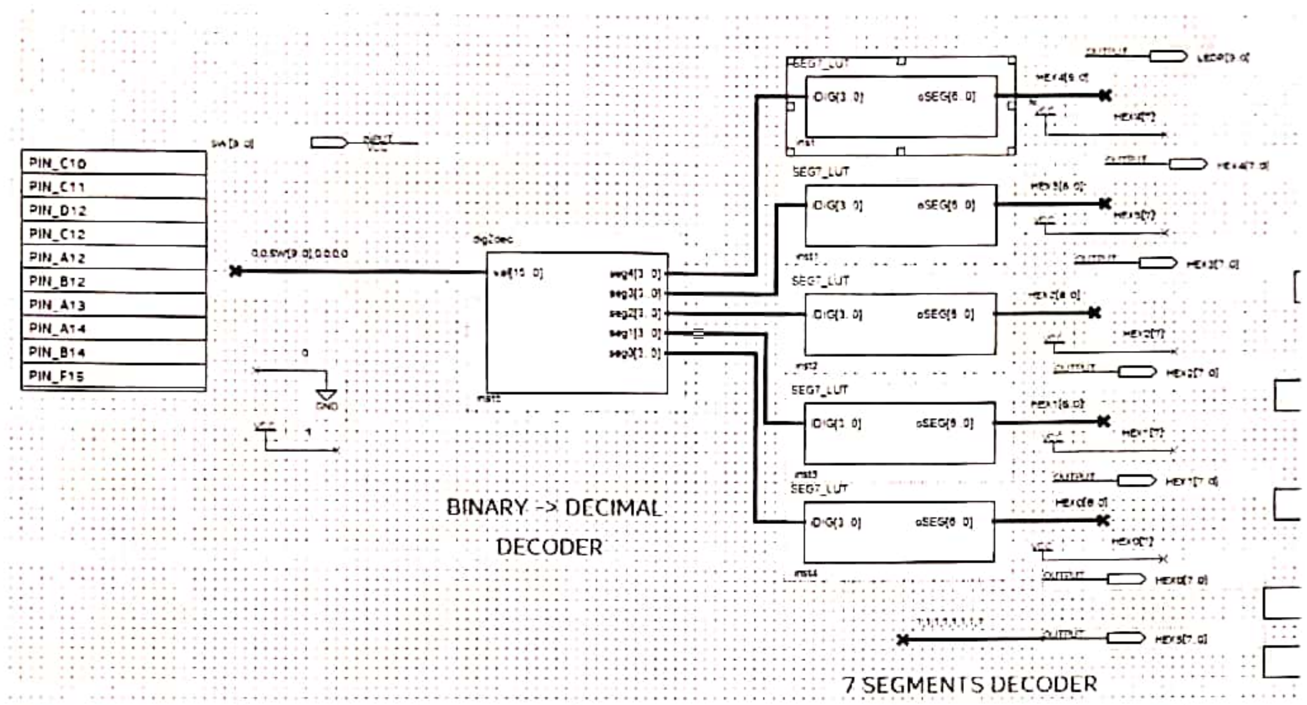
The supervisor has placed on your account a directory named BE_ALTERA in which you will be working. You will work in this directory.
You go to the TD2 subdirectory and then run the ../QUARTUS command, which launches the Altera development tool.

You then go to the menu file -> Open project
Then you open the digital_decimal project

The project appears on the left in the window. You double click on the icon to the left of the diagram


Quartus Prime Lite Edition - C:/Users/benabes/Documents/2_FORMATION_INITIALE/NC_3A_Mention SCMA/Cours numéri
File Edit View Project Assignments Processing Tools Window Help
digital_decimal
Project Navigator
Hierarchy
Entity instance
MAX 10: 10M50DAF484C6GES
digital_decimal

This brings up the diagram of the project in the center. It contains an input, the binary -> decimal decoder and 5 7-segment decoders which will control the displays.



You have already been given the code for the 7-segment decoders. You can view it by double clicking on one of the 5 boxes on the right.

We will mainly work on the dig2dec module which is the binary -> decimal decoder
You double click on the box, which contains an empty code that you will have to fill in.

2 °) Writing and compilation of the code

The entity is already provided to you. It contains the input: val and the 5 output vectors seg0 to seg4. They are of type std_logic_vector
You must now write the architecture code.

Starting from the algorithm provided, calculate on how many bits are encoded the vectors val1 to val4.
You will define the 4 signals of the right size using an unsigned type.
Remember to check if all the correct libraries have been opened.

Then translate the algorithm by making concurrent assignments of the different signals and outputs and using the integer division / and the modulo: mod function.

You have to think about the size of the result and make good use of the resize function as well as the unsigned () or std_logic_vector () cast functions.

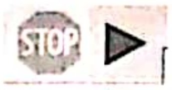
Once the code is written, you pre-compile it and you check it using the menu
Processing -> analyse current file

You will have to start over until there are no more syntax or other errors.

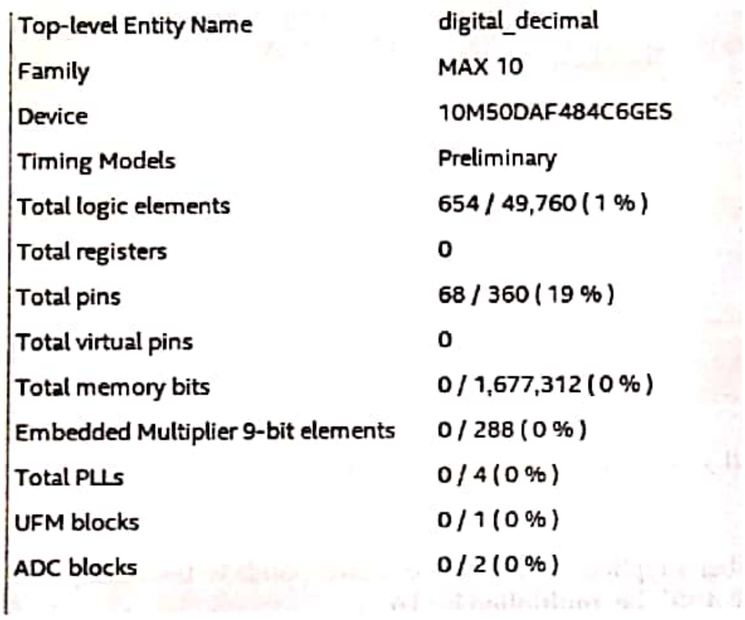
3 °) Elaboration. placement, routing and test on board.

All the first part is done by clicking on the blue arrow next to the stop sign or by ctrl l on the keyboard

257 $dec_0 = 0$ $dec_1 =$
$k_0 = 0$ $\rightarrow t = 2$
$Y_0 =$ ~~257~~ 0,7

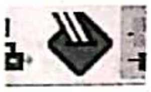


This initiates the compilation, elaboration, placement and routing.
If there are no errors all that remains to do is to download to the board.
Before that, we will note the consumed resources provided here in the flow summary. We can see that we use in this first example 346 logic blocks and 0 registers.



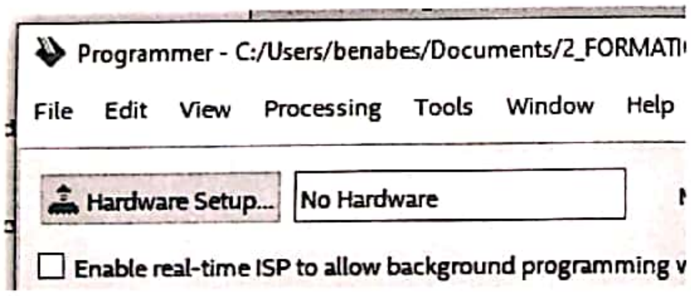
Top-level Entity Name	digital_decimal
Family	MAX 10
Device	10M50DAF484C6GES
Timing Models	Preliminary
Total logic elements	654 / 49,760 (1 %)
Total registers	0
Total pins	68 / 360 (19 %)
Total virtual pins	0
Total memory bits	0 / 1,677,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 288 (0 %)
Total PLLs	0 / 4 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 2 (0 %)

The card download is done via the programmer which is launched via the following icon



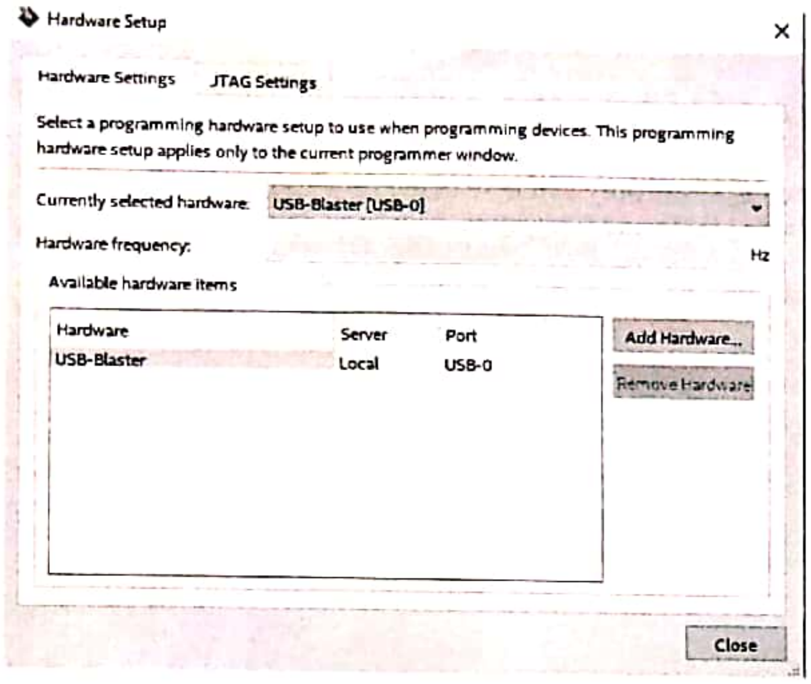
Of course, you have to plug the board into a USB port on the station before.

If the card is not recognized immediately (no hardware), click on "hardware setup"



Programmer - C:/Users/benabes/Documents/2_FORMATI

And select USB blaster


Hardware Setup

Hardware Settings | JTAG Settings

Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

Currently selected hardware: USB-Blaster [USB-0]

Hardware frequency: Hz

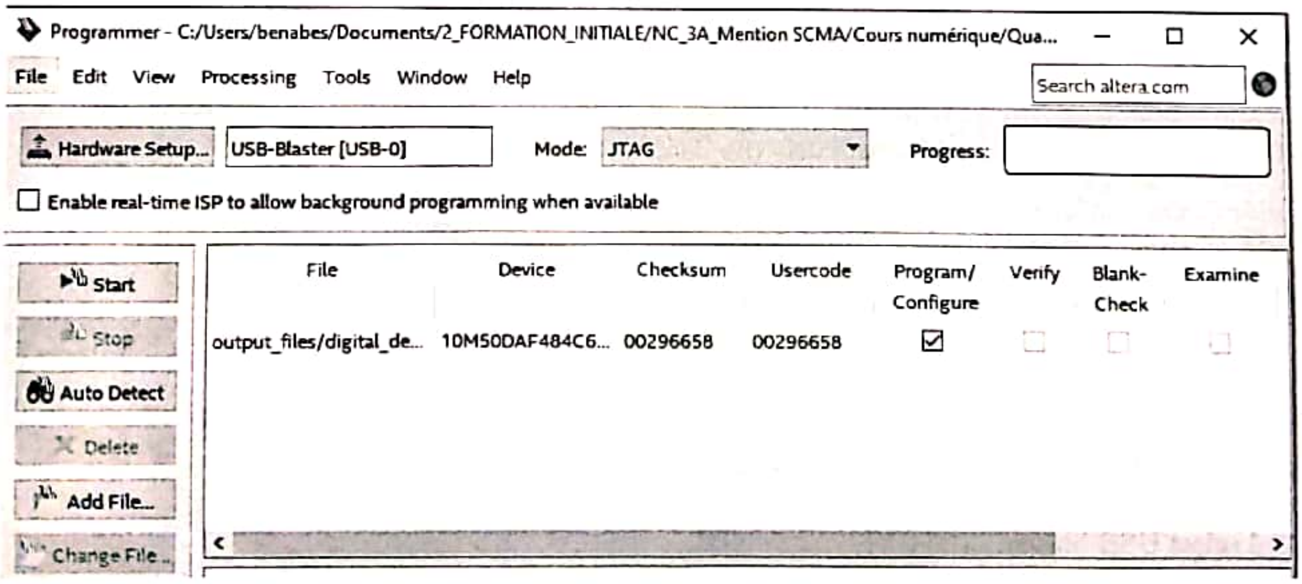
Available hardware items

Hardware	Server	Port
USB-Blaster	Local	USB-0

Now that the USB blaster is recognized, all you have to do is click Start to download the code to the card and test.

If we look closely at the diagram, the number supplied to the decoder corresponds to the binary number coded by the switches followed by 4 '0', i.e. multiplied by 16.

We will play a little on all the switches in order to test that each of the digits is working correctly.


Programmer - C:/Users/benabes/Documents/2_FORMATION_INITIALE/NC_3A_Mention SCMA/Cours numérique/Qua...

USB-Blaster [USB-0] Mode: JTAG Progress:

☐ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine
output_files/digital_de...	10M50DAF484C6...	00296658	00296658	☑	☐	☐	☐

4°) Code improvement

The most expensive functions in arithmetic are divisions and modulo. In our case, we cannot do without divisions, but we can avoid modulos

Rewrite the operations without using modulo and recode the operator
We will go through all the previous steps until the download and the card test.
Note the number of logical elements used.

Did we win in terms of logic used?

5°) Using a process

We now want to code the whole function in a single process
Define the order of operations
What should we now encode val1 to val4 with ?

Adapt the architecture accordingly and re-test the whole circuit
Note the amount of resources used.
Did the use of one process change the result?