

## INTRODUCTION TO “MAXPLUS II”

MaxPlusII is produced by the US company Altera and is designed to be used with its logic programmable circuits. The program allows the designing of the digital circuit (schematic, or using a hardware description language VHDL, Verilog or AHDL), its compilation for certain programmable device, its simulation and programming the circuit onto the chip.

### **What are the programmable logic circuits?**

A logic programmable circuit (PLD – Programmable Logic Device) is an integrated circuit used for obtaining a reconfigurable circuit or application configurable. Unlike the classic logic circuits ( gates, counters, registers) that have a fixed function, the PLD has not a predefined function after the fabrication. Before being used, it must be programmed.

Before PLDs being invented, the memories were used to create combinational logic circuits. Considering a memory with  $m$  address inputs and  $n$  data outputs, we have  $2^m$  locations with  $n$  bits each. If the addresses are driven by  $m$  independent signals (the inputs of the logic function), and in the memory locations are stored the values of the  $n$  functions, this structure is equivalent with a programmable logic circuit with  $m$  inputs and  $n$  outputs. Because the memories have no input or output registers, they cannot be used for implementing sequential circuits.

In the past many types of PLDs were patented: PAL, GAL, CPLD, FPGA.

PAL – Programmable Array Logic are based on small PROM memories (read only programmable memory) plus additional output logic. They were used to implement digital custom functions into one integrated circuit.

GAL – Generic Array Logic is an evolved descendent of PALs that can include more functions than many PALs together and allows to be reprogrammed.

CPLD – Complex Programmable Logic Device is more complex than a GAL circuit. It is realized with cells (Macro Cell) that allow normal disjunctive implementations of logic functions. They allow the storage of the configuration after powering it off and include a large number of logic gates.

FPGA – Field Programmable Gate Array are the most evolved circuits from the flexibility and size point of view. The circuit is based on logic blocks (figure 1

LUT means Look Up Table – memory, FF is a circuit with 2 stable states (flip-flop) and MUX is a multiplexer. This kind of cells can implement either, combinational circuits using the LUT output, either sequential circuits using the FF output. The cells can be connected between them anyway, through a matrix of switches, besides that the cells are programmable. The input and outputs can be routed to any pin of the circuit.

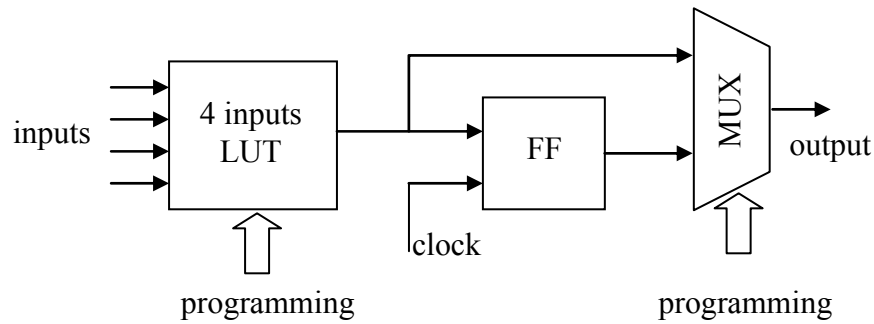


Figure 1. Logic block in FPGA

### MaxPlusII

MaxPlusII is a free software package that can be downloaded from Altera’s site este ([www.altera.com](http://www.altera.com)), the license is given for free after registration.

The program installs itself on C:\maxplus2\ and creates an additional folder C:\max2work\. When open the program it shows a window like in figure 2.

During the lab we will work with 2 types of files: gdf – graphic design file (file with schematic circuit) and scf – waveform editor file (file with simulation signals). For better understanding we will create a simple

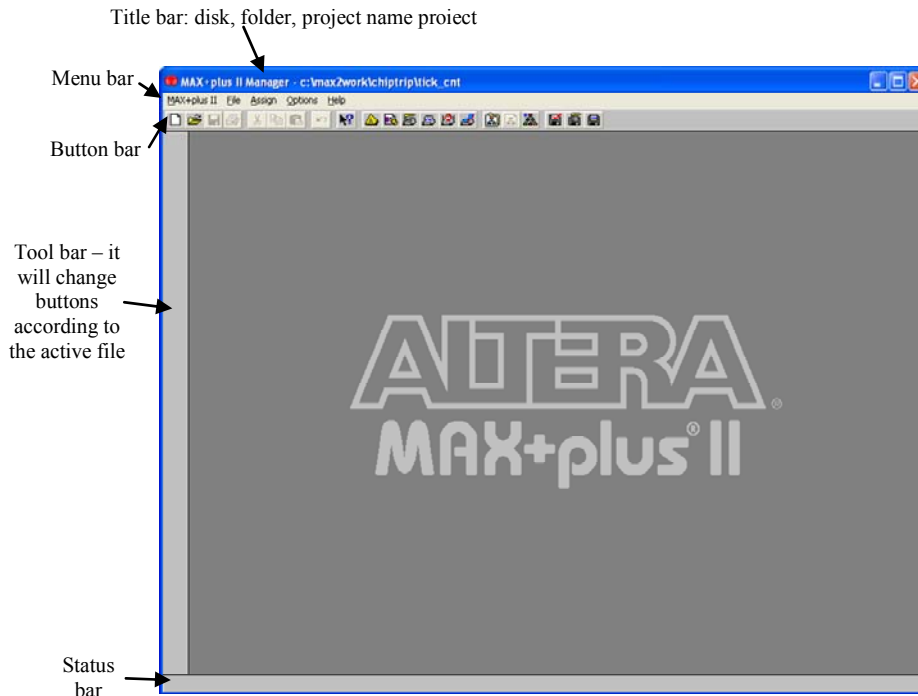

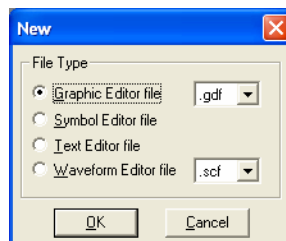


Figure 2. MaxPlusII main window

example.

From the *File* menu or using the button  on Button bar open new file. A dialog window opens:



Select the “Grafic editor file” and press OK. The main window populates with a blank sheet, and on the “Tool bar” and “Button bar” some buttons activate/change (figura 3). Moving the mouse arrow over them, their function will be displayed on the “Status bar”.

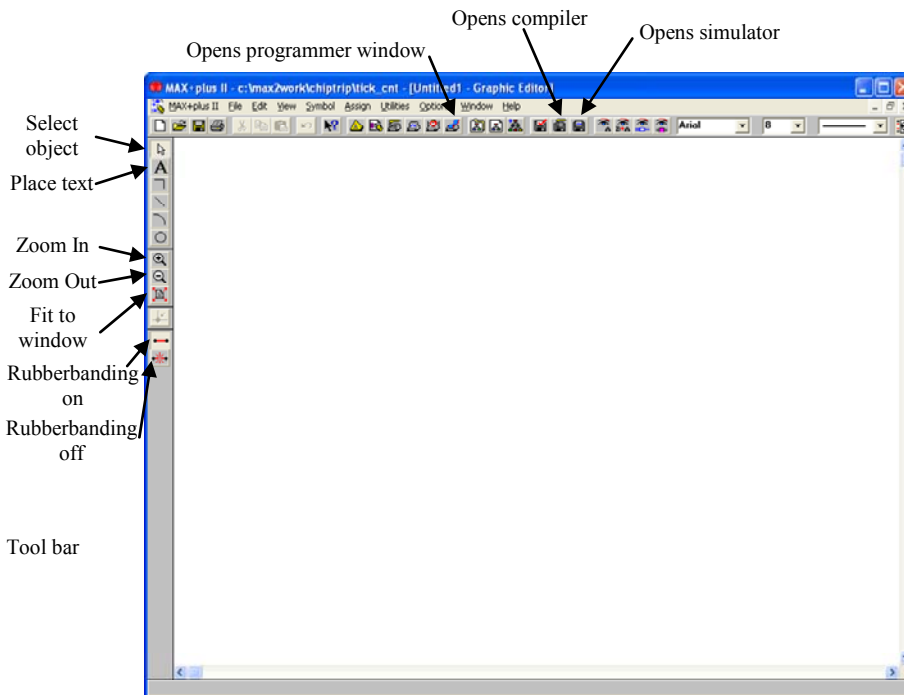
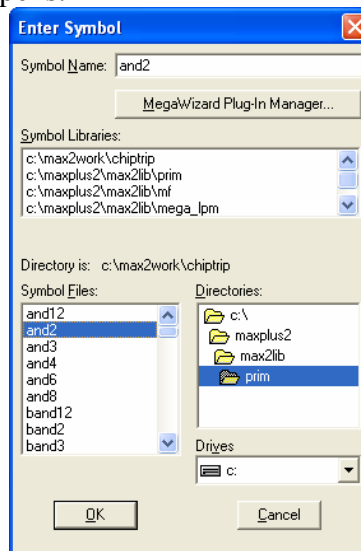


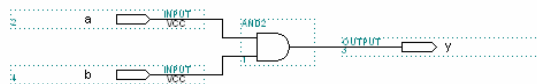
Figura 3. S-a deschis un nou fișier grafic


Double click on the white sheet (or right click and select *Enter Symbol*) and a new dialog window opens:




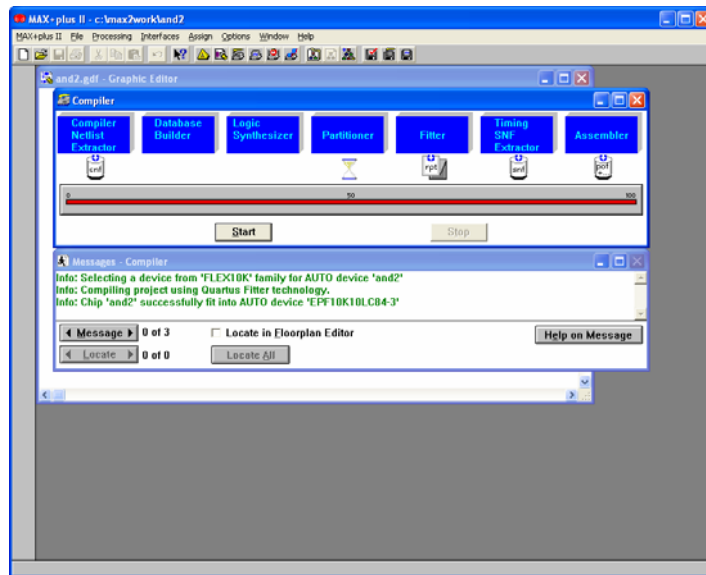
Here you can select the components, either by writing the name in the *Symbol name* dialog, either by selecting the component from a library

(double click on a library in the *Symbol libraries* dialog and then chose the symbol in the Symbol files dialog) and then press OK. Choose a 2 input AND gate (and2). After you press OK the gate will appear on the sheet. If you need more gates you proceed in the same way. The circuit needs inputs and outputs. These are components too and they are called *Input* and *Output*. You proceed like you did for the AND gate because they are components too. For this circuit you need 2 inputs and an output. After you place them on the sheet you must give them a name. You must double click on PIN\_NAME or right click and select “*Enter pin name*”. The area is getting black and you must write the name. The name must have only letters and numbers. The only special character is “\_”. In our case name the inputs as “a” and “b” and the output as “y”. The cursor is changing into a cross when is over a terminal of a component. You click the right button when it is a cross and keep it pressed while you move it toward another terminal. You must connect the inputs and the output to the gate like in the next picture.

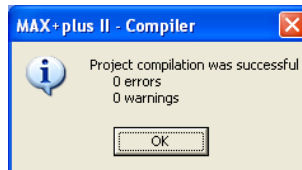


Now the circuit is ready. You can save the file now (File, Save or the button ). Automatically the gdf extension is attached. Then from *File, Project, Name* or “*Set Project to Current File*” the file is assigned to a project. **All the files of the project must have the same name and must be in the same folder.**

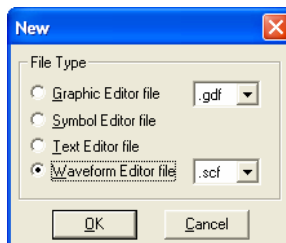
Now the project must be compiled by pressing :



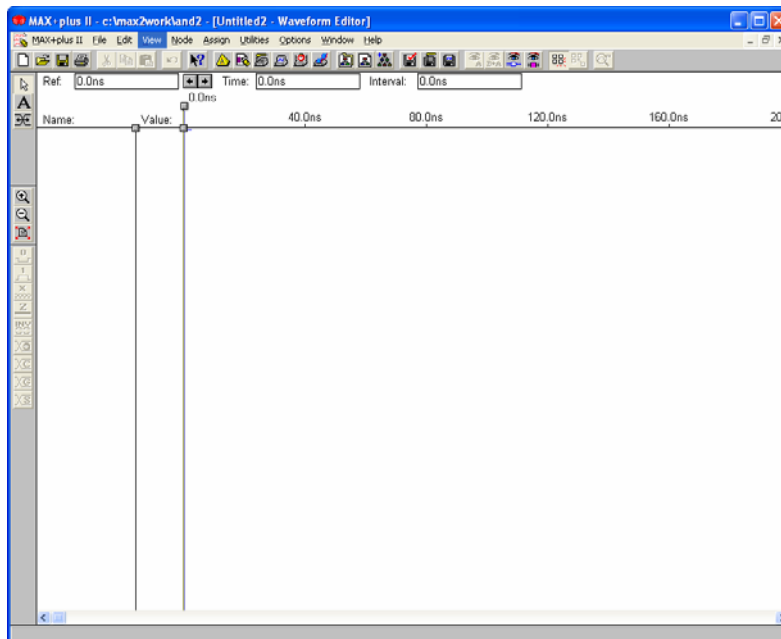
The compiler is opened and the compilation starts. In the message window you can get infos, errors and warnings. In order to solve the errors you double click on the error and the program will show the problem in the circuit by turning the parts red or you press “*Help on message*” and a help window will be opened. If the compilation is without errors you get the message:



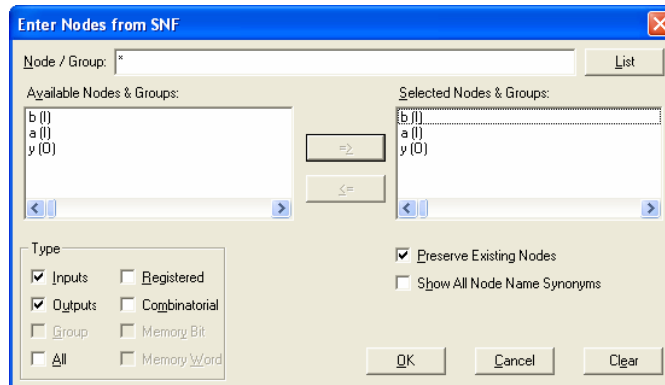
If the compilation has no errors you close the compiler and you can pass to the simulation stage. You open a new file, but you select a waveform editor file this time:

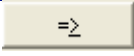


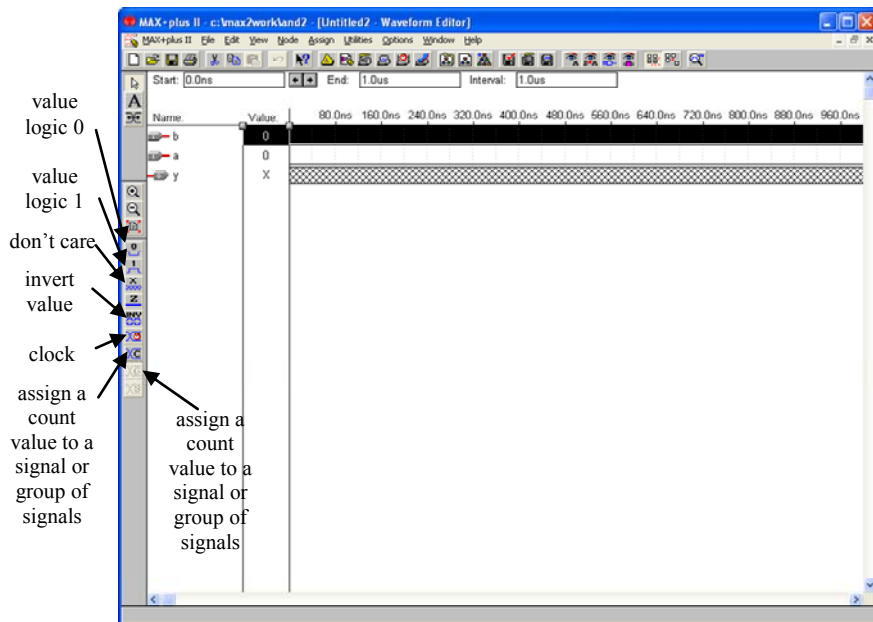
A new waveform white sheet is opened:



You can introduce the signals one by one (right click under “Name:” and press “Insert node”) or all at once (right click under “Name:” and press “Enter nodes from snf”). You choose this time the second option and a new dialog is opened:

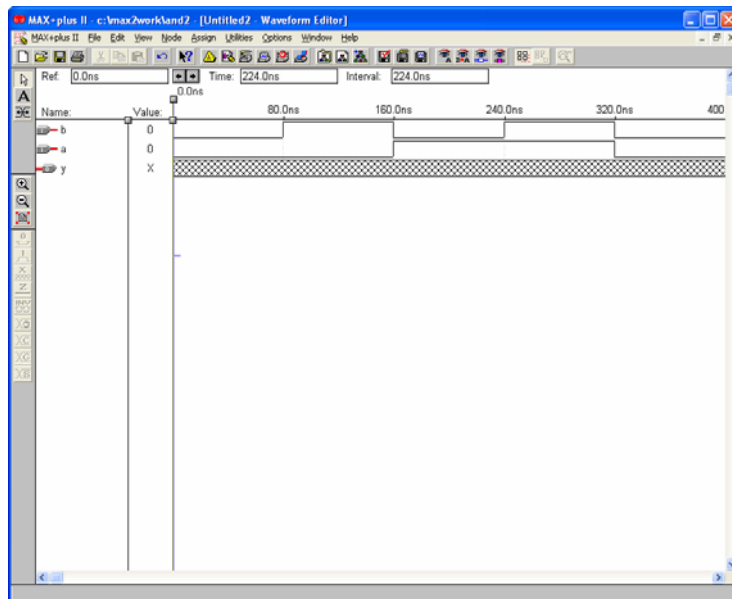



Press the button “List” and then  and *OK*. On the screen you have the three signals inserted. The output is hatched because the simulation was not performed yet. This means that the signal is unknown.

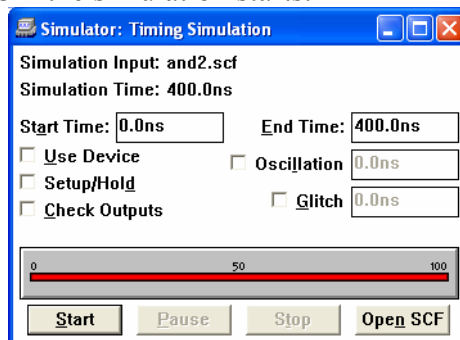


In order to perform a simulation you have to give values to the input signals. After the simulation the output signals will have the values resulted from simulation. If you click on each signal the Tool bar will activate and you can give values to them. For now you choose a clock type signal for both inputs: for input „a” you choose the multiplication factor 1 and for „b” the multiplication factor 2. Usually for n inputs there are  $2^n$  distinct states. In this case we have  $2^2=4$  states. For each input more the clock will be multiplied with 2 in respect with the previous. This way all possible states will be obtained. According to the grid size the state length is fixed: *Options, Grid size*. 100ns is the value you choose. The simulation end time is at least the number of states multiplied with the state length:  $4 \times 100\text{ns} = 400\text{ns}$ . From File menu you setup the End time to 500ns (a state more than necessary). This is because the circuit will introduce a delay due to the circuit finite switching time. You can see this way the delay for the final state.

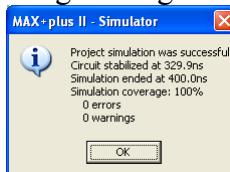




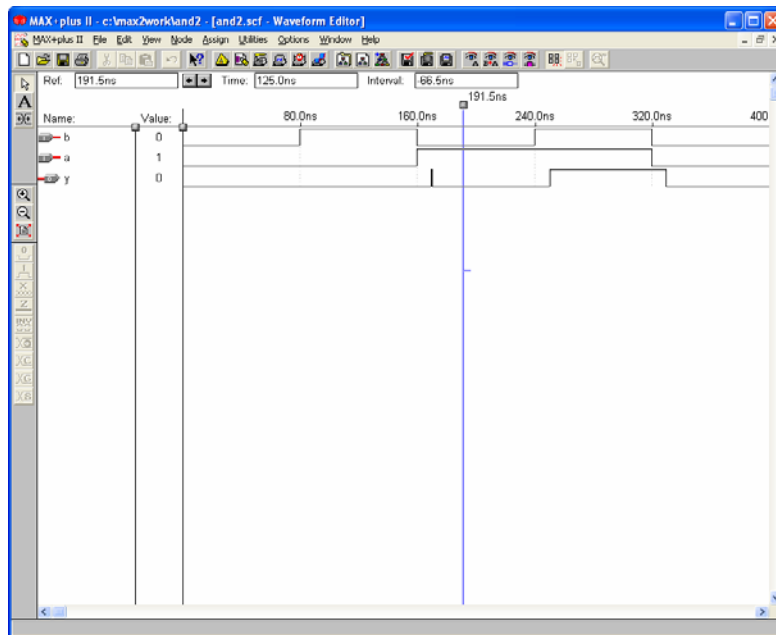
After setting up the input values the simulation can be started by pressing . The program will ask for saving the waveform file. In the dialog window it will have the same name with the gdf file, but with the extension scf. After pressing OK the simulation starts.



If there are no errors, the following message is displayed:

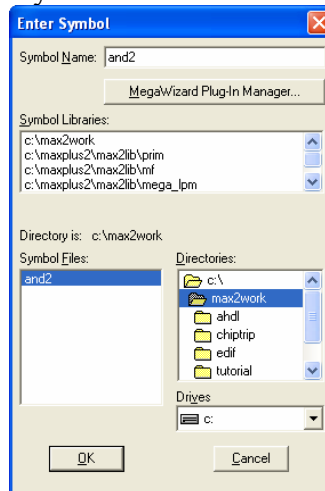


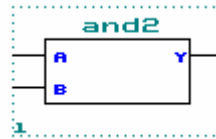
After pressing OK the simulator will be closed and in the waveform window the simulation results are available:



By clicking in the waveform area a blue cursor will be displayed. It can be moved along the waveform with the arrows are with the mouse from the small grey square on the top and in the „Value” column, the values at the cursor will be displayed. You have to notice the delay between the input and the output edges and the output values.

Once the circuit working you can attach a symbol to it. With the schematic gdf in front you use *File, Create Default Symbol*. A new symbol is created. You can use it in other projects. In order to see it, you open a new gdf sheet. You put in this file the new symbol:





In order to use it all the files must be in the same folder. If you double click on the symbol, the previous schematic will be opened. This allows to facilitate working with larger schematics.

You can also measure the delay between input and output, or the time interval between two time moments like it is shown below:

