Fast transient recorder and data acquisition based on FPGA

Gottardo M.[1]

[1] Consorzio RFX, EURATOM-ENEA, Corso Stati Uniti 4, I-35127 Padova, Italy

marco.gottardo@igi.cnr.it

research goal:
Fast development of FPGA solutions for time critical tasks

Introduction

In fusion research it is sometimes necessary to handle datat a very high sampling rate. The normal current microprocessor systems are inadequate because of their limitations in speed. A solution is to realize the data acquisition using FPGA (Field Programmable Gate Array) technology, which allows parallel calculations in real time. FPGA programming is however not an easy task because of the required skills in HDL. **The main topic of my research program consists in developing a LABVIEW framework to manage the FPGA hardware, which represents an alternative approach to FPGA application development**. In particular the developments of this tool will start from the managements of the asynchronous events in a new configuration for fast ADC . Possible applications:Fast transient,ADC converter, Saddle coils fast control.

Hypothesis: Each hardware logic function can be emulated software and vice versa.

Take in example the Truth table:

In 3	In 2	In 1	Out 2	Out 1
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	0
1	0	1	0	1
1	1	0	0	1
1	1	1	0	1

The boolean function is:

(In3 * in 2) + (in 3 * in 2) = Out 1

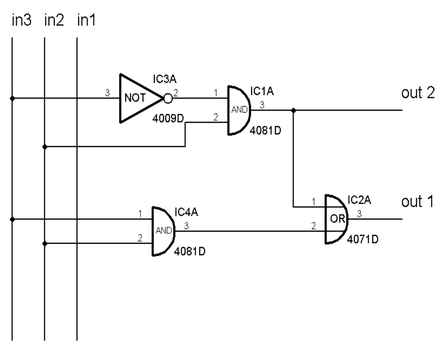
(In3^{not} * in2) = Out 2

It can be emulated by the software:

```
/*********************************************
*Program virtual LUT for PIC16f876A  *
* ing. Marco Gottardo                *
* Phd. Event Lisbon 2014             *
*********************************************
char in1,in2,in3,out1,out2;
void main(){
    if (!in3 && in2){
      out2=1;
    }
    else{
      out2=0;
    }
   If ((!in3 && in2) || (in3 && in2)){
      out1=1;
    }
    else {
       out1 = 1;
    }
}
```

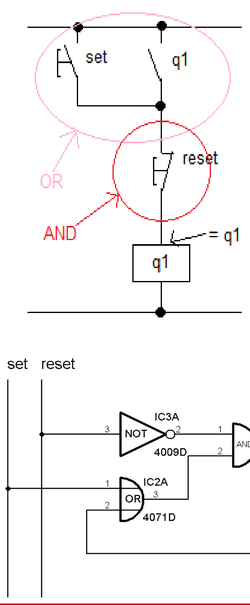
Typical execution time microsecond

The truth table rappresent the LUT element inside the FPGA unit



Typical execution time nanosecond

in the same way we can emulate sequential networks such as the flip-flop which is the fundamental cell of the FPGA.



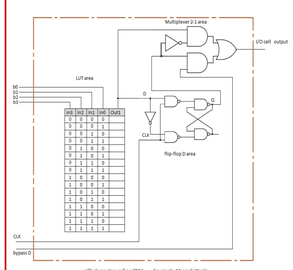

```
/*********************************************
*Program virtual flip-flop for PIC16f876A *
* ing. Marco Gottardo                *
* Phd. Event Lisbon 2014             *
*********************************************
char set,reset,in3,q1;
void main(){
    if (set || q1) & (!reset){ //if the sensor on
machine is NC avoid the not operator "!"
      q1=1;
    }
    else{
      q1=0;
    }
}
```

The flip-flop unit rappresent a fundamental element inside the FPGA

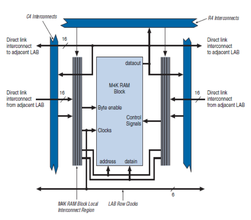
The **Field Programmable Gate Array** are composed of three elements.

- **logic cells**
- **I/O cell**
- **Interconnection cells.**

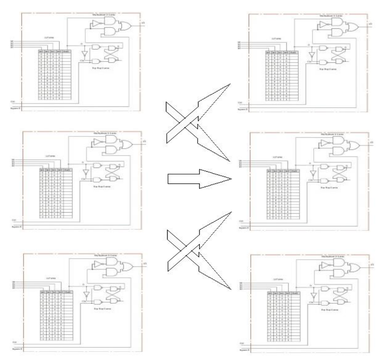
There are in addition groups of internal RAM.



Logic cell whit 4 input Look up table

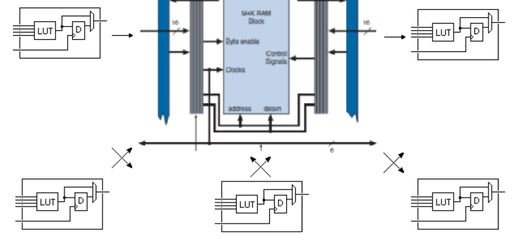


internal RAM blocks.

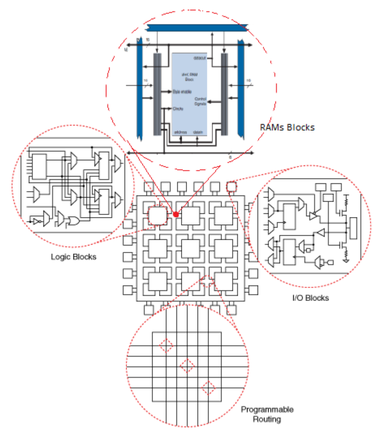


Programming a FPGA means to connect together many the I/O cells

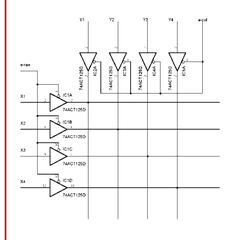
This is an example of interconnection.



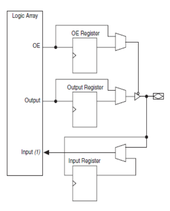
Between the logic blocks can be connected to the RAM areas



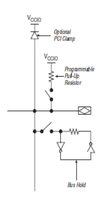
Typical complete architecture of an FPGA.



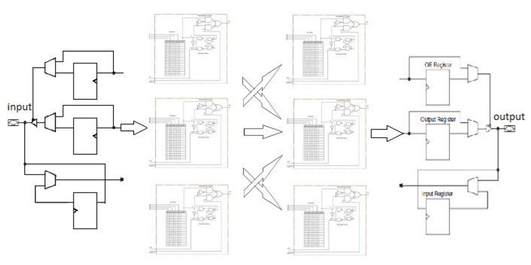
Interconnection cell



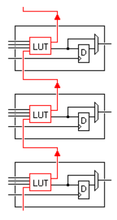
I/O interna configuration



Digital pinout configuration

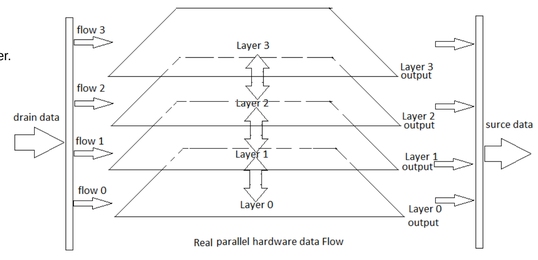


interconnection of the internal architecture with the cells of I/O



interconnection by layers parallel

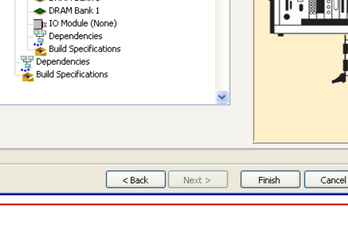
We can configure the internal architecture to get more parallel layers each composed of portions of the inner matrix shown in the upper right corner.



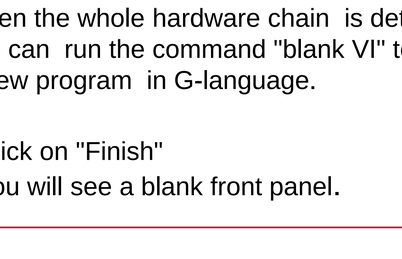
Real parallel hardware data Flow

FPGA can be configured on Labview environment using G language

The system must be composed of a host that is running Labview and a target containing an FPGA configured from the host, but it can work independently in a real time.
Host can be a normal PC because during the data acquisition there aren't any relation between PC OS and the rack where the FPGA is running.
In Labview must be present the FPGA module.



Software installed in host PC



Target device is the Flex RIO PXI Rack

Target device is a PXI 7953R, ad it communicates with the host in the same way of any PCI device

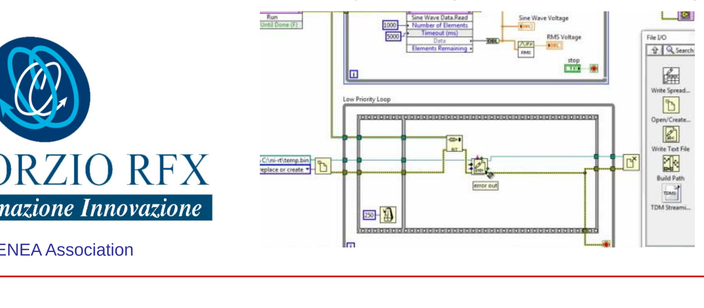
The FPGA board is inserted into a slot, but you need to equip it with an extender.
The one shown here is the expansion is the I/O module NI6581 that provides 48 channels.

Digital signals are input or output depending on how you configure the FPGA.
You can configure the voltage levels of 1.8, 2.5, 3.3 volts, it is recommended to use the largest value in noisy environment.

Host must recognize the target during the project creation like shown in the image.

When the whole hardware chain is detected you can run the command "blank VI" to start a new program in G-language.

Click on "Finish"
You will see a blank front panel.



Blank front panel. Right click to have a tool box

Is possible create configurations FPGA for digital or analog.

Drag and drop analog tool

Fast analog data aquisition, FPGA example

Drag and drop digital tool

Example of acquisition of digital I/O with FPGA.

Sigma delta fast ADC architeture