ModEasy: technical meeting on FPGA

Luc Charest and Sébastien Le Beux

26 May 2005

Participation

- **UK**
  - Klaus
- **IEMN**:
  - Smail
  - Jamel
  - Yassin
- **LIFL**
  - Eric
  - Samy
  - Ouassila
  - Luc
  - Sébastien

Objective

This first meeting targets an information exchange on several FPGA aspects.

Chapter 1

Gaspard flow presentation

Sébastien begins the meeting with presentation of the Gaspard flow, which is a co-design approach. Co-design means that application and architecture development are made independently. A third concept (association) places and routes application on architecture. Until now, it was possible to place application on processor or DMA. Within ModEasy project, it will be necessary to use FPGA.

1.1 Using specific FPGA

A link with an university tool will permit to target a reconfigurable architecture, described in this annexe tool. The problem is that each time a new architecture is targeted, its description has to be provided in annexe tool. Furthermore, to be able to really configure a particular FPGA, knowledge about its bitstream structure of is needed. Unfortunately, those kind of information are unknown.

To be able to successfully complete ModEasy project and the real implementation of task on reconfigurable architecture, we will have to use a commercial tool.

1.2 Using commercial FPGA

To configure a commercial FPGA with our algorithm, it requires the use of a commercial tool dedicated to the target architecture. For its new algorithm concerning obstacle detection, University of Valanciennes bought the Altera commercial tool and one of their chip. The main reason of this change of commercial tool (the one used previously was Xilinx) is the maximum frequency supported by the FPGA selected. In their case, the FPGA is provided on a board which contains analog to digital converter. Thus, the interface with radar (which provides analog signal) is available. University of Kent especially works with Xilinx and ST tool.

In order to target several FPGA from several constructors, it seems to be possible to use Mentor tool. But, is Mentor able to really configure hardware?

1.3 Several step to chose the best architecture

In case that reconfigurable hardware hasn't been selected, it should be necessary to enable an selection of the best architecture for one application. This selection shall be done using the characteristics of reconfigurable hardware. It will permit to select on architecture contain in library.

Selection of the best architecture can be done in three steps:

- The **Resources needed** to fully implement application on hardware. At this step, all reconfigurable hardware that are not able to support full application are deselected.
- The **Context** shall permit to remove other possible solutions. Context may be other hardware component, bus... existing on chip.
- The **Commercial tool**, which permits, fine estimation of power consumption, maximum frequency... will permit to take final decision. As simulation time is quite important at this step, the number of possible solution existing before this step shall be reduced as much as possible.

Step to select the best architecture for an application shall permits, in future work, integration of architectural exploration.

Chapter 2

Avoidance radar from Valanciennes

2.1 Correlator algorithm

Yassin presents the avoidance radar application using the correlator algorithm. Several version of this algorithm has been implemented on four different FPGAs. The differences between the FPGAs are their maximum frequency, number of logic cells they contain...

A common entry point of all these versions is the selection of a run time frequency. This frequency fix the application resolution of the circuit, and, thus, permits an estimation of logic elements used by the application.

Information about frequency and logic elements to use are not fixed. For example, it is possible to reduce by half circuit frequency with an increase by twice of logical elements used. Thus, there is a compromise between frequency and logic elements used.

2.2 HOS algorithm

Jamel has presented a new algorithm to implement application on hardware, the Higher Order Statistics algorithm. The same approach can be used for this kind of algorithm as the first one.

Chapter 3

Discussion / proposition

3.1 Characterisation of reconfigurable architecture

In order to choose the best architecture for one application, it is necessary to increase a certain number of characteristics contained in the architecture model of the Gaspard flow. Those characteristics can be specific to reconfigurable architecture.

The maximum frequency supported by the reconfigurable architecture is a parameter quite important which can be crucial (if architecture can support or not implementation of application).

Basic elements contained in the reconfigurable architecture is an important parameter too. But this element is different from a constructor to other one. For Altera, the basic element is call “Logic Element” (LE), whereas “Logic Cell” for Xilinx.

In the avoidance radar application case, input and output pins dedicated to the user is not a restrictive parameter. Indeed, current FPGAs offer, at least, 100 I/O pins, whereas avoidance radar application only needs inputs and outputs to consume signal provided by the analog to digital converter (about 12 bits) and to produce the computed value. But, if additional parameters have to be introduced in reconfigurable hardware (to create other function for example), more inputs will be necessary. That is why this characteristic should be introduced in the architecture model.

Data storage capability is another main FPGA characteristic to include in architecture model.

Several FPGA contains DSP blocks, which can contain several multipliers. For future work, they could be important elements, but currently, they involve some problems. One of them is that their instantiation requires the use of specific VHDL code. Furthemore, there are not enough multipliers in FPGA (even in recent ones) to support full the avoidance radar application.

The last point implies that only generic VHDL should be use to describe application.

3.2 Benchmark application

Basic elements of reconfigurable hardware are not the same from a constructor to other one. Thus, to be able to compare power computation of several FPGAs from different constructors, it is necessary to create a library. This library should contain implementation characteristics of test bench application on architecture. Thus, it would be possible to get information about basic elements used for an application implementation.

Because University of Valanciennes has used four different FPGAs from two constructors, we propose to begin by characterization of a test bench application on these four architectures.

Nature of test bench application as to determined.

Chapter 4

Next meeting

- **Date:** Monday 13 June
- **Work to do:** Model correlator algorithm in Gaspard, using ArrayOl.