

POLITECNICO DI MILANO
MICROLAB
HPPS PROJECT REPORT



Design Flow

A new work methodology definition - 3rd Phase Report

Instructor: Prof. Donatella Sciuto
Tutor: Marco Domenico Santambrogio

Project Author:
Antonio Piazzi
711430

A.A. 2006/07

Indice

1	Introduction	3
2	State of the art	3
3	A new work flow methodology	5
4	Results	6

Elenco delle figure

1	User attention spreading	4
2	User has to start again his previous work	5
3	Work flow methodology	5
4	Process' steps in details	7

Sommario

Our work answers to the need of the novice and expert users to make easy and faster their new feature's development. In this context we analyze two of the most famous work flow proposed by Xilinx to build a reconfigurable architecture. We notice that some operations needed by that flow design are very nasty and distract the users from their goals (build a new hardware software functionality).

In this report we describe the new work philosophy on which our frame work is based on and the new work flow (a mixture of a module based and early access work flow proposed by Xilinx) which help the user to focus his attention only on the develop of his functionality.

1 Introduction

Try to explain what are all the motivation which guide us to the developing of this supply chain is quite hard.

We notice a need of attention to the user, attention that the frame work proposed by Xilinx not always satisfy. The Xilinx's tools are vary usefull if an expert user try to do what they want he does, but when a user try to develop something different or try to implement a partiicular functionality (not yet standardizated) here he found not little problems and here he spend his time not for what he's pay to do but to solve heuristic problems surf upon the Internet praying to find a solution which is, sometimes, contained into a new service pack distributed by Xilinx twice a day.

We think that all this is less productive than pay a monkey to do our work, so we decide to think different and develop a supply chain which free the user from this nasty operation, flexible to support also future flow implementation and easy to understand.

In our mind the user has only to define what his work has to do, define some constraints (of time or of area or both) and nothing more.

We present here our work, in the first section we describe which are the operation in which our work in called to help the users a sort of the state of the art, in the second section we present which are the philosophy on which our frame work is based on, in the third section we describe the work step in which we divided our work methodology and how we implement all the process and, finally, in the forth section we present a sort of work in progress, a project naturally linked to the frame work which is the head of the strengthening of that frame work which is schedulated for that summer.

2 State of the art

In this section we try to explain which are and which are not permitted to do easy with the Xilinx's proposed tools.

First, all we have do until today is strictly linked with this tools because they are the only software we can use to develop somthing for the Xilinx's devices.

Xilinx develop some CAD software and that software are collected in two different tools (ISE and EDK) which, ideally, could help the user on the developing phase. We use “could” because, actually, that tools only automated some process otherwise executable only through command line. The thing is that when we try to compose a normal static architectur all works good, but when we work on border line develop somthing based on some reconfigurable work flow (ex. module based or early access) we have to fight against some stupid error throw some bad parameters default set by the tools or some operation not supported by the previous realise of the program (ex. ERP patch). That sort of thing force the user to spread his attention also on phase otherwise totaly automated or nearly.

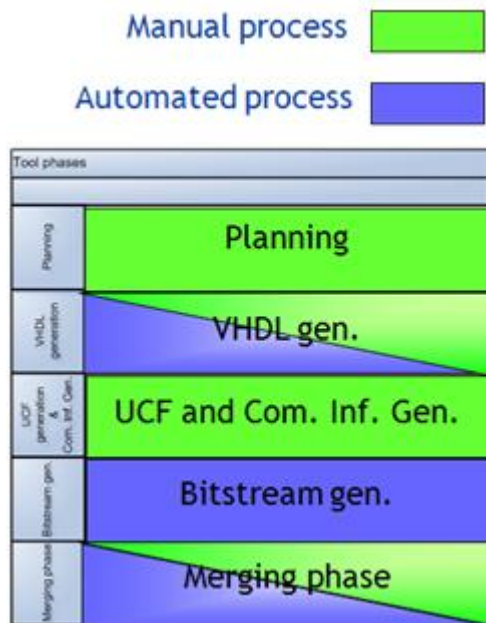


Figura 1: User attention spreading
The attention of the user is spread uselessly on thifferent phase

In this context we observe a fast loss of concentration from the user which has too much problems to solve and also, douring a formular phase a too long time to market, which is, in some critical sectors, too much expensive.



Figura 2: User has to start again his previous work

3 A new work flow methodology

We try to avoid to the user all the operation that may be automated. So we analyze both the Xilinx's flow to underline the main differences between them and collect all the common operations in a new basic work flow.

The result is a flexible flow which can switch on one of the two original flow based on the information collected in a natural way in the file generated by EDK which contains all the information about the architecture build.

That file (system.vhd) contains informations about the IBM core-connect architecture which is static and don't present any understanding problems by the user, but to work on reconfigurable set we need to modify that file using information collected inside it and other collected in other files not yet presents in the build enviroment.

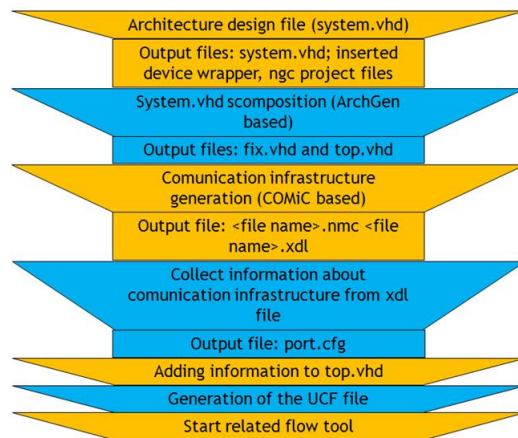


Figura 3: Work flow methodology

Each step is characterized by an input file (from the previous step) and a output file send to the next step

To switch from a static architecture to another one reconfigurable we use a previously tool developed in the Politecnico di Milano - MicroArchitecture laboratory, ArchGen. That tool generates two output files, one (fix.vhd) contains the information of the static architecture (as processors, bus, memory bank and that sort of thing) the other (top.vhd) contains an instantiation of the fix part but to work good need information also about the communication infrastructure.

To collect that information we use another tool proposed by Politecnico di Milano - MicroArchitecture laboratory which name is COMiC that generates two types of output files, an NCD file which contains the physical instantiations of the macro hardware (3-state or slice is not different) and one or xdl files which contains the same information in a text mode.

We need so of a parser to collect all this information and put into a new top.vhd file which has to contain the previous information and that news.

Generated that new file and take the user constraints file created by the user we are able to run all the flow compatibly with the used device.

4 Results

During development of this framework we want to maintain checked the scalability of our methodology, we know that using previous tools we import in our project all the bugs proper of that tools, so we define an increase plan in which we put at the first point the capabilities to work with any type of bus with the only restriction to use an xdl definition file. To be sure of that assertion we try to create an OPB replica bus based on 3-state buffer, the result was a transparent integration of that bus into the new top.vhd file, that shows us how use of the new device may increase our power with the utilization of the standard OPB transfer protocol as communication infrastructure between the fix part and the reconfigurable part.

Now that bus is hand made but, after all scheduled work, we presume to create a patch of COMiC to instantiate more types of bus.

This experiment grants us the retro-portability of our work and agrees to our idea to build an infrastructure around the user which has to adapt itself on the requirements of the user.

Our goal is help the user, not force the user to do what we want.

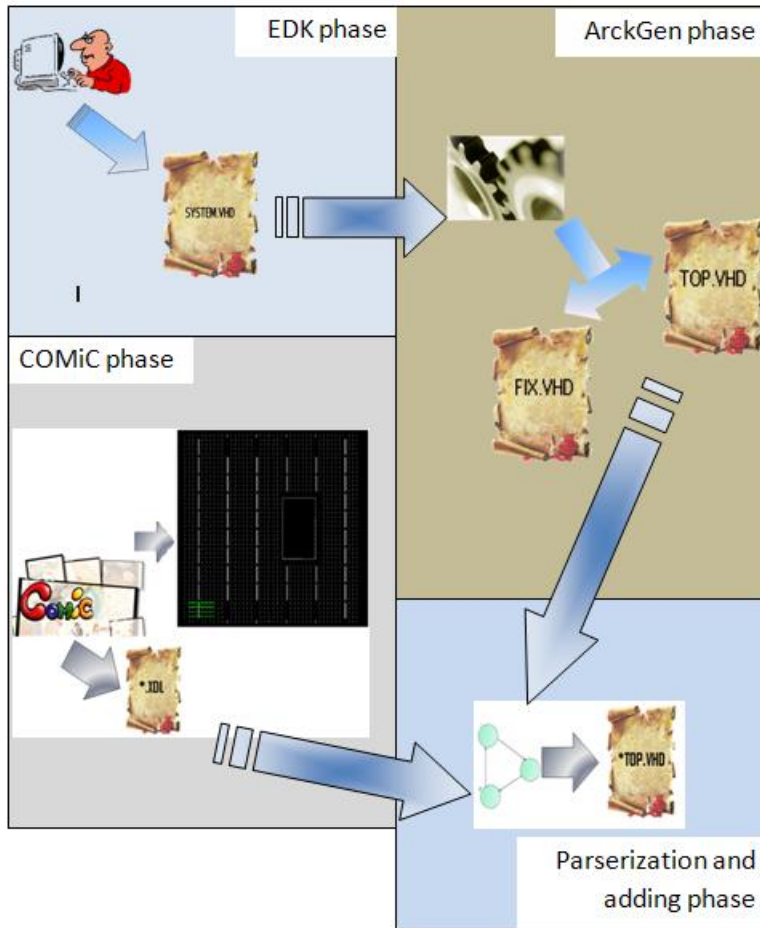


Figure 4: Process' steps in details