- DRAFT -

Co-Design at Chess-iT

by

G. Bosman

Supervisors:

Dr. Ing. R. Lämmel (VU)

Ing. P. Eussen (Chess-iT)

Drs. B. Bos (Chess-iT)

Contents

Chapter 1

Introduction

Here be words.

Chapter 2

Assignment

2.1 Development methods

In my assignment I will select 3 development methods. This selection will be based on:

- Complete coverage of the process from design to implementation,
- Having a representative candidate of each class of developments methods,
- Expectations on which language is the most likely candidate to become an industry standard in the field.

These methods will be chosen in close coordination with my internship supervisors. A refinement will have to be made for some languages. For example UML is such a broad language that a description of the parts of UML used has to be given.

I will investigate the relevance and relationships between these languages in the track from specification to implementation. I'll research to which degree existing development and implementation methods fit into a single paradigm.

2.2 Scenarios

I'll carry out two small experiments using these methods, to get a good understanding of the problem of distribution. These experiments will be based on existing problems Chess faces and should have different 'natural' distributions of hardware and software. The following two projects have been given by Chess:

Table 2.1: 3 methods, 2 scenarios.

	Method 1	Method 2	Method 3
DSP chip	A1	A2	A3
Fuel processing system controller	B1	B3	B3

- Design and implement a DSP chip that compresses or encrypts information (streaming) (A)
- Design and implement a controller that is used in a fuel processing system to send information back to the gas-company (B)

2.3 Methods vs. scenarios

The methods and scenarios can be seen in Table 2.1.

For each combination I will try to do the whole process of specification to an (abstract) implementation in hardware and software. The hardware will be made using a FPGA.

Chapter 3

System Level Modelling

Mooney et al.[2]: Approaches to hardware/software co-design of embedded systems can be differentiated in several ways. One way is to consider the system-level specification, which is either homogeneous (i.e., in a single specification language) or heterogeneous (i.,e. involving multiple modelling paradigms). Another way is to distinguish how the CAD tool partitions the system specification: fine-grained or coarse-grained.

3.1 Homogenous modelling

Homogeneous modelling implies the use of single specification language for the modelling of the overall system. Co-design starts with a global specification given in a single language. This specification may be independent of the future implementation and the partitioning of the system into hardware and software parts. In this case co-design includes a partitioning step aimed to split this initial model into hardware and software. The outcome is an architecture made of hardware processors and software processors. The is generally called a virtual prototype and may be given in a single language or different languages. [1]

3.2 Heterogeneous modelling

Heterogeneous modelling allows the use of specific language s for the hardware and software parts. The co-design starts with a virtual protoype where the hardware/software partitioning is already made. Here the emphasis is on the integral designing of the parts to make sure

the overall system has the required properties. The key issues are validation and interfacing. [1]

3.3 Previous work

In this thesis we will deal with homogeneous modelling. Several models are available.

- Cosyma
- VULCAN (Hardware C)
- VHDL
-
- Polis (Esterel)
- Spec-syn (Spec-Charts)
- SDL (COSMOS)

Bibliography

[1] A. Jerraya, M. Romdhani, P. Le Marrec, F. Hessel, P. Coste, C. Valderrama, G. Marchioro, J. Daveau, and N. Zergainoh. Multilanguage specification for system design and codesign.

[2] III Vincent J. Mooney and Giovanni De Micheli. Real time analysis and priority scheduler generation for hardware-software systems with a synthesized run-time system. In Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design, pages 605–612. IEEE Computer Society, 1997.

[3] Ti-Yen Yen and Wayne Wolf. Hardware-Software Co-Synthesis of Distributed Embedded Systems. Kluwer Academic Publishers, 1996.