



# LUND UNIVERSITY

## Controller Synthesis for Hardware Accelerator Design

Jiang, Hongtu; Öwall, Viktor

2002

[Link to publication](#)

*Citation for published version (APA):*

Jiang, H., & Öwall, V. (2002). *Controller Synthesis for Hardware Accelerator Design*. Paper presented at Swedish System-on-Chip Conference 2002 (SSoCC 02), Falkenberg, Sweden.

*Total number of authors:*

2

### General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117  
221 00 Lund  
+46 46-222 00 00

# Controller Synthesis for Hardware Accelerator Design

Hongtu Jiang and Viktor Öwall

Digital ASIC Group, Department of Electrosience, Lund University

## Abstract

Efficient CAD tools are desired to reduce the increasing design efforts when algorithms implemented on ASICs are getting more complicated. For microprogrammed accelerator design, a control unit synthesizer is of great importance since the manual design of a controller for a complicated task requires substantial effort. While a hardware specific implementation results in higher performance or lower power consumption than a programmable solution, flexibility might be crucial. Therefore, in the future it is desirable to have on chip control units to be more or less programmable. The goal of the project is to develop design methodologies for such a design environment. Those methodologies should be implemented into a tool to reduce design time and allow flexibility of the design process.

## 1 Introduction

For the last two decades, the tremendous growth in the area of microelectronics technology has enabled more complicated circuit design in both analog and digital domains. Building a whole system on a single chip seems to be right at hand. On the other hand, increased complexity along with the desire for high performance, low cost and increased pressure on design time has constantly called for more powerful CAD tools. Currently design tools at physical and logical level are widely available and extensively used in the industry. More and more research work is conducted on the behavioral and architectural levels[1].

When a system is to be implemented on hardware today, it is usually composed of several components: general purpose processors, memories, hardware accelerators, etc. as shown in figure 1. Together with software they will comprise an embedded system[2, 3].

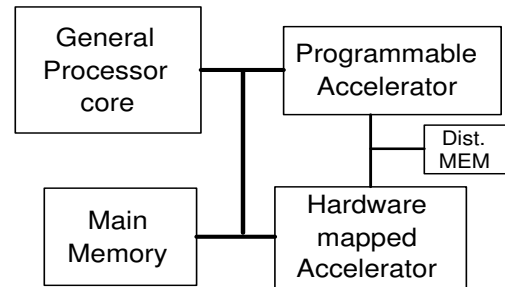


Figure 1: embedded system

In such architectures, a general-purpose processor core can always be used to implement control-intensive functions and system I/O, while the computation-intensive tasks are left to hardware accelerators for improvement of the calculation capacity. Controller structures are used as well within controller-datapath type of accelerator for the control of datapath modules and communication with "outside" components like processors, memories, and other accelerators, etc. Manual design of such controllers can be achieved by writing corresponding VHDL specifications which is synthesized by CAD tools as long as the design task remains relatively simple. However, with the increased complexity of integration, where more functions are expected to be added, such design could encounter great difficulty. To bridge this gap, some kind of design automation is needed to reduce the design effort and shorten the time to market.

## 2 COMA

A previous synthesis tool called COMA[4] was developed to attack those problems. In order to automatically synthesize a controller, COMA requires two specifications: the behavioral description of the

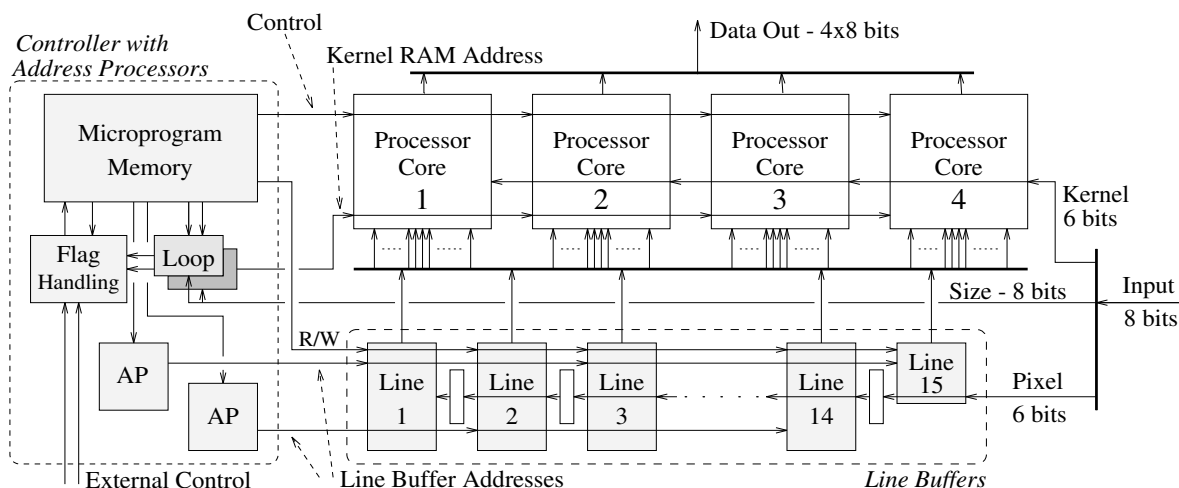


Figure 2: Block diagram of the processor

processor architecture defining the available set of micro operations and the microprogram containing the algorithm with additional declarations such as memories. The parser generator YACC has been used to construct the parsers of the behavioral description and the program. A C-like input syntax is in use to provide easy programming and the high readability. A range of controller architectures are available for the designers to choose for the implementation of the controllers. The output is given in the form of a complete controller with module descriptions and interconnection specifications. Additionally, command files to run logic generators, memory generators, and datapath compiler are created.

However there is still a lot of work to do regarding controller synthesis and ways of improving the performance and increasing the possibilities provided by the tools. Examples are: improving the state coding, implementation of other algorithms for encoding of control signals, further improvement of input and output format, etc.

### 3 Current work

Over the years, design environment has been subject to many changes and hardware description language, like VHDL, has become an actual standard that is widely accepted among the circuit design community. Hardware synthesis designed using such languages are now supported by most CAD tool providers. Since COMA was developed for use in an old design environment, the output format as well as some other features were found to be outdated. Therefore, many

modifications are under development in order to make it fully integrated into a present design environment. Improvements, for instance, adding VHDL support and FPGAs as prototypes are being made.

#### 3.1 Controller synthesis for image processing application

One of the applications being developed is the controller synthesis for an image convolution processor[5]. In this application, a customized processor for real time image processing is designed to increase the performance of an instrument for automated cereal grain quality assessment. The performed image processing is a two-dimensional convolution[6, 7] of the image with a 15\*15 kernel function in order to detect certain features of the image, such as outline, color, lines, etc. Since image convolution requires an extensive amount of calculation capacity and a corresponding amount of data transfers, a tailored architecture with a streamlined dataflow was developed, to achieve the desired filtering which are hard to implement with standard processors in real time. The block diagram of the processor architecture is shown in figure 2. The image is scanned from the upper left corner of the image, first horizontally and then vertically, and one convolution is completed when the kernel has reached the lower right corner of the image. Since each pixel except the extreme corner pixels is used in several calculations, the pipelined memory bank is implemented to store successive pixel values allowing each value to be read only once, thus reducing the input datarate. The designed circuit has four processor cores containing the

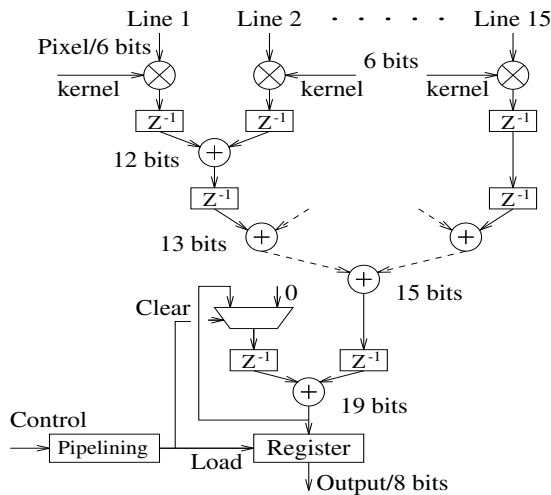


Figure 3: schematic diagram of the processor core.

kernel functions and performs one of the four convolutions in parallel. A schematic diagram of the processor core is given in figure 3. Each processor core contains 15 multipliers with adjoining RAM for the kernel function meaning that one column of the kernel, 15 pixels, can be calculated in parallel. The calculated values are added in a tree structure of adders and pipeline registers and stored in accumulator. In the tree structure, the number of bits increases to avoid overflow in the adders.

The processor was designed for a clock frequency of 20 MHz resulting in  $>2G$  arithmetic operations/ $s$ . For the time being, VHDL specification for the processor core has been developed and the synthesized netlists with Synopsis are completed for later use of constructing the whole design.

### 3.2 Controller design

The processor cores require a very simple controller with only a single control signal while the line buffers and the kernel RAMs require extensive address calculations and loop control. Therefore, a controller synthesizer was used to synthesize a complete controller dedicated to this algorithm from a microprogram.

Currently, a controller architecture with incremental circuitry, decision handling part is under development, as shown in Figure 4. In this architecture, the branch address calculation within the same block of codes is performed by the hardware incrementer while at the end of a block a non-incremental branch address is calculated by the control logic and a select branch signal, also referred to as end of block signal, is set. An address processing unit is also used together

with controllers to handle the memories for storing both the coefficients and temporarily computed values.

Synthesizable VHDL specifications of the whole controllers are expected within a short time. Combined with VHDL code for processor cores and memory banks, targeting FPGAs prototypes will be available. However the synthesis work of controllers will not be restricted to this application. In the future more architectures are within interests and supposed to be available in VHDL as well.

## 4 Future work

In modern circuit design, design cost and design risk has been a major concern among the companies making ASICs. In previous design fashions, if new functions are needed, the hardware has to change accordingly, which would take considerable time for the design, verification and fabrication. But if a design is made a programmable platform, which means providing user programmability, the users can often provide functionality equal to that of a hardware by just writing new lines of code for a programmable processor without any change of the circuits. In recent years, this has already become a hot topic among networking and wireless communication system designs [8]. Today, however, there is still in short of EDA tools that support for designing these programmable platforms.

On the other hand, the level of programmability should be taken as a trade off between flexibility and performance-the most flexible processors like Pentium suffers from relatively lower performance and higher power consumption when dealing with real

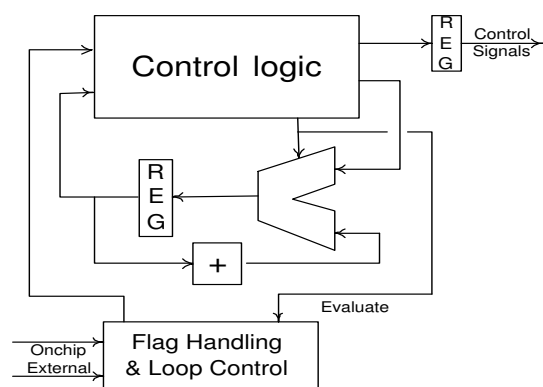


Figure 4: Controller Architecture with incremental circuitry

time signal processing while on the other hand hardware implementation of algorithms like accelerators has no flexibility after fabrication. The possible solutions seem to be somewhere in between and supposed to be different for variant applications. So in the future, how trade offs should be made for various systems is probably one of the major concerns in the later design of controller synthesis tools.

Other issues under consideration includes the functionality to be added to the controllers for supporting communications with "outside" components, this will comprise the consideration of both a generalized description of such communications and how such specification is transformed into hardware and software modules.

## 5 Conclusion

Controller synthesis tools are powerful for the design of controller/datapath systems. It can reduce design effort and design time substantially. In future designs, programmability is expected to be one of the major concerns since a sustainable platform is desired nowadays to ameliorate design risk and design cost. At the same time, the level of programmability should be taken as a trade off between performance and flexibility.

## References

- [1] P. Eles, K. Kuchcinski, and Z. Peng, *System Synthesis with VHDL*, Kluwer Academic Publishers, 1998.
- [2] S. Edwards, L. Lavagno, E. A. Lee and A. Sangiovanni-Vincentelli, "Design of embedded systems: formal models, validation, and synthesis," *Proceedings of IEEE*, vol. 85, no. 3, March 1997
- [3] M. Chiodo, P. Giusto, A. Jurecska, H. C. Hsieh, A. Sangiovanni-Vincentelli, L. Lavagno, "Hardware-software codesign of embedded systems," *IEEE Micro*, Volume: 14 Issue: 4, Aug. 1994.
- [4] V. Öwall, *Synthesis of Controllers from a Range of Controller Architectures*, Ph.D. Thesis, Department of Applied Electronics, Lund University, Dec 1994
- [5] V. Öwall, M. Torkelson, and Egelberg, "A Custom Image Convolution DSP with a Sustained Calculation Capacity of  $> 1$  GMAC/s and Low I/O Bandwidth", *The Journal of VLSI Signal Processing*, 23(2): 335-350; Nov 1999.
- [6] J. S. Lim, *Two Dimensional Signal and Image Processing*. Prentice Hall, 1990
- [7] W. K. Pratt, *Digital Image Processing.*, John Wiley & Son, Inc., second edition, 1991.
- [8] K. Keutzer, "Bright Future for Programmable Processors", *IEEE Design & Test of Computers*, vol. 18, issue. 6, PP. 7 -8, Nov.-Dec. 2001.