DESIGN OF THE HADAMARD COPROCESSOR WITH THE ALLIANCE CAD SYSTEM CARRIED BY POST-GRADUATING STUDENTS

A. ZERROUKI, J. DUNOYER, F. WAJSBÜRT, A. DERIEUX
University P.et M. Curie Laboratoire LIP6, ASIM Team
4 Place Jussieu F75252 Paris Cedex 05 France
Tel : +33 1 44 27 54 15   Fax : +33 1 44 27 72 80
{Amal.Zerrouki, Julien.Dunoyer, Franck.Wajsburt, Anne.Derieux}@lip6.fr

1. Introduction

University Pierre and Marie Curie in Paris, offers a one-year course on Advanced CAD and VLSI design for post graduate students (DEA ASIME, DESS CISAN). This course starts with a general introduction to circuit and system architecture and a presentation of ALLIANCE[1], the educational CAD system developed by the ASIM team of the university. This first part lasts 4 weeks and aims to settle the basic knowledge necessary to follow a more specific course focusing on either CAD tools design, Circuit design , or System design. In between, a 3-weeks project is proposed to allow the students practicing the concepts presented earlier. Depending on the students option, the project consists in implementing either a MIPS R3000[2] or the Hadamard coprocessor[3].
This paper presents the Hadamard project with a special focus on the methodology followed by the students from specification to layout.

2. Hadamard Implementation

The Hadamard coprocessor designed is actually the CMOS implementation of the Hadamard transform used for image compression. It computes the H*P*H matrix, where H is the Hadamard matrix, and P is the image matrix to be compressed. H is a square matrix and contains 8x8 bits. P is also a square matrix and contains 8x8 bytes. Computing the matrix H*P*H consists in 2 steps : computing first the intermediate matrix H*P, then the final matrix (H*P)*H.
Given the possible values of H, the product of a matrix with H is nothing more than a series of additions and subtractions. Basically, 8 operations are performed to obtain each element of a product matrix. Each element of H*P is stored in the corresponding row of a register file until a complete row of the intermediate matrix is obtained. Then, a row of the final matrix H*P*H can be calculated by multiplying the row in the register file by the corresponding columns of the H matrix. These steps are performed 8 times to complete the calculation of the 64 elements on the H*P*H product.
The circuit architecture is presented in figure 1. A *RAM* is used to fetch the matrix P corresponding to an image. *COMPUTE* is provided to compute the elements of both the intermediate and the final matrices. A *register file* is also included to store the

intermediate elements. *COUNTERS* includes 3 counters C1, C2 and C3 used to handle the indexes of the different matrices in use. The H matrix values are delivered by the *MAT* bloc. A sequencer, *SEQ*, is provided to control the blocs operation.

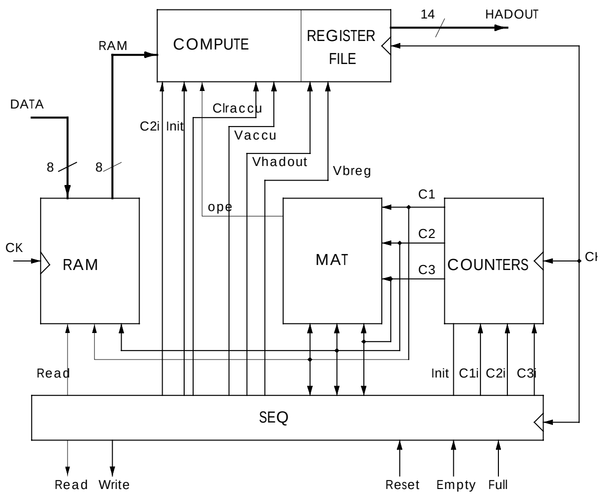

Figure 1 : HADAMARD Architecture

3. The Choice of the Hadamard Coprocessor

Although it is rarely used for image processing given its poor efficiency, this algorithm is still a very interesting example for education for various reasons. First, the algorithm is simple enough to be understood and implemented in 3 weeks. Then, the resulting architecture is complete enough to exercise the major architectural concepts (automata, register files, rams, fifo interfaces, synchronous communication, ...). Last, the VLSI implementation is simple enough to let the students get familiar with the methodology behind ALLIANCE.

This project encompasses the various aspects of VLSI design and offers the opportunity to students to measure the effects of their various implementations.

4. Design Methodology

4.1. DESIGN VIEWS

The methodology behind ALLIANCE is said *zero-default* and *top-down.* Basically, three different views are used : *the behavioral view*, *the structural view* and *the layout*.

The *behavioral view* is generally the starting point. It describes the functional specifications in VHDL (dataflow and finite-state-machines). The *structural view* corresponds to a netlist connecting standard cells or hierarchical blocs. It is either elaborated by the designer or obtained by synthesis including eventually an optimization loop. The *layout* of a chip is generated by Place & Route tools.

4.2. A HIERARCHICAL DESCRIPTION

Figure 2 presents the functional structure of Hadamard. The highest level is *Chip*, a structural view interconnecting the core to its pads. *Core* is itself a structural view. It connects the different blocs building the Hadamard function. *Seq* is a finite-state machine which controls the operation of the operators. *Ram* is a hand-written gate netlist. *Counters* is a behavioral view describing *C1*, *C2* and *C3* counters. *Mat* is the behavioral description of the Hadamard matrix. *Compute* is a behavioral description of the arithmetic unit performing calculations on matrix elements.

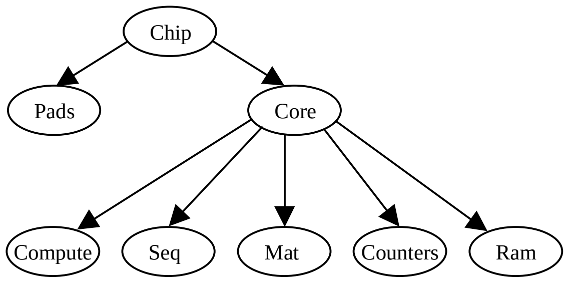

Figure 2 : Functional hierarchy

4.3. BLOC SYNTHESIS

Given the variety of input descriptions, different paths are considered to generate a gate-netlist. A finite-state-machine is first processed by SYF to generate a behavioral description. This last is then handled by the BOP optimizer to reduce its boolean expressions. The resulting behavior can finally be mapped by the tool SCMAP in standard cells using the portable cell library of ALLIANCE. This generates a gate-netlist necessary to move to the Place&Route step.
A behavioral description such as *Mat* or *Counters* is directly processed by the boolean optimizer. A new behavioral description is generated and mapped by SCMAP. The input description of *Ram* is a gate-netlist. There is no need to synthesis.

4.4. SCAN PATH INSERTION AND FANOUT OPTIMIZATION

In 3 weeks, a full study of chip testabilty is impossible. A scan path is therefore preferred. All registers are scanned, except those of the *Ram*. GENSCAN generates a new gate-netlist in which registers are chained. To ease the use of the scan path, students can specify the order in which the registers should be scanned.
The last step before Place&Route consists in running on each netlist a fanout optimizer to make sure that no fanout rule is violated. Such a step is not compulsary at this point of the design. If one needs to reduce the overall chip area, timing analysis will help identifying critical paths and thus those for which fanout optimization is necessary.

4.5. CHIP PLACE&ROUTE AND DESIGN RULE CHECKING

At this stage, all blocs building the chip are available under the form of gate-netlists. *Mat*, *Counters*, *Seq* and *Compute* are joined in an intermediate netlist named *Control*. SCR, the standard cell router is used to generate the layout of both *Ram* and *Control*. The Bi-Block Router BBR generates the *Core* layout, and the whole chip layout is generated by RING, the pad router. This last places and routes the pads in a ring around the *Core* and connects each pad to *Core*. The layout of the core and the chip are finally verified using the design rule checker DRUC to make sure that no symbolic rule has been violated.

5. Timing Analysis

The transistor-level netlist of the chip is extracted from layout using LYNX, the ALLIANCE extractor. The resulting netlist includes the capacitors attached to each physical segment. Using such a netlist, static timing analysis is performed with the tool TAS to identify critical paths. Depending on their analysis, students may have to return to synthesis and optimization.

6. Validation Environment

In ALLIANCE, each step of the design should be followed by a validation step.This is often achieved through logical simulation. In this project, system-level simulation is considered. A test bench is defined including the Hadamard core and a Stimulator bloc. This last exchanges data with Hadamard using a fifo interface. One advantage of such an approach resides in the fact that the circuit is used in its environment. Logical simulations are performed using ASIMUT, the ALLIANCE simulator. This tool accepts both behavioral and structural views. As a consequence, this unique test bench is complete enough to validate the initial behavioral view written by the students as well as all other views obtained throughout the design.

7. Results

The resulting Hadamard chip contains 36000 transistors for an area of around $6530*6400\lambda^2$. In a 0.35μm, this leads to 5.2 mm^2 area. The operating frequency determined by timing analysis varies from 45MHz to 60MHz.
The circuit has been successfully implemented in 3 weeks by groups of 3 students each. Although a real and complete circuit cannot be implemented in such a short time, this project is essential to students since it gives them the opportunity to practice circuit design from specification to layout using various CAD tools.

8. References

1. Greiner A., F. Pêcheux, “ALLIANCE . A Complete Set of CAD Tools for Teaching VLSI Design”, *In The 3rd Eurochip Workshop on VLSI Design Training*, pp. 230-37, Grenoble, France, 1992.
2. Bazargan Sabet P., Dunoyer J., Greiner A., Rosset-Louërat M.M, A tutorial for Advanced VLSI Course : Designing the 32 bit DLX with the ALLIANCE CAD System, In *The 5th Eurochip Workshop on VLSI Design Training*, p 120-27, Dresden, Germany 1994.
3. Gonzalez R. C., Woods R.E.(1992) *Digital Image Processing*, Addision Wesley.