A LOW POWER PROTOTYPE FOR A 3-D DISCRETE WAVELET TRANSFORM PROCESSOR

Guoqing Zhang, Mike Talley, Wael Badawy, Micheal Weeks and Magdy Bayoumi

The Center for Advanced Computer Studies
University of Southwestern Louisiana
Louisiana, LA 70504
{gxz0233, mxt1206, wmb, mxw, mab}@cacs.usl.edu

Abstract

This paper presents a low power prototype of a 3-D discrete wavelet transform processor. It illustrates the design flow used in developing the prototype, and the mapping and verification starting from the high level modeling of the chip design and concluding with the low level implementation details at the transistor level. The main application of this architecture is toward the 3-D MRI data, but other applications will benefit from the 3-D DWT as well, such as digital broadcast. The processor has been prototyped using 0.6 μm CMOS (three metal) technology, the prototype processor is modular. It has been simulated at the functional, circuit, and physical levels. The performance measures of the prototype, area, time delay, power and utilization has been evaluated. The prototype operates at an estimated frequency of 272Mhz and dissipating 0.5W of power. The prototype processor uses 16-bit adder, 16-bit Booth Multiplier and 1 KB cache with a maximum of 64-bit data bandwidth. Lower power has been achieved by using low power building blocks and the minimal number of computational units with high throughput

INTRODUCTION

With the recent explosion of the Internet, technologies such as Tele-medicine, video conferencing, and wireless data communication have become a reality. However, these technologies require an immense data bandwidth, which is a always limited. One of the proposed solutions to this problem is to compress the data before transmission. Data compression is an important factor in many applications, e.g. storage of large data sets, which include Magnetic Resonance Imaging (MRI), Digital Television, and Seismic Data Collection. DWT is one of the most efficient compression algorithms because of its perfect reconstruction property and lack of blocking artifacts [1]. Different real time wavelet transform systems are used for signal processing applications such as video compression [2], Internet communications [3] and object recognition [4]. But, applications such as digital video will benefit from 3D DWT [5]. The 3-D wavelet transform offers a better compression ratio than conventional compression methods, such as the Discrete Cosine Transform (DCT). The DCT considers the correlation of the images it is compressing. The blocking effect inherent to the DCT is also negligible in the 3-D DWT. In comparison to the 2-D DWT, the 3-D DWT outperforms the 2-D DWT by 40-90% [6].

Different DWT architectures have been proposed in the last decade. All of them vary in terms of scalability (e.g. number of filter coefficients), area, control complexity, latency, and memory. These architectures depend on the degree of the DWT i.e. 1-D, 2-D, or 3-D. Different design approaches have been reported such as systolic [7], semi-systolic [8, 9, 10], folded [11], digit-serial [11], etc. Some architectures use centralized control structure [12, 13], while others use distributed control structure [14]. The centralized control signal is easier to implement, but it suffers from the limitation of the scalability [15]. Organization of the memory architecture varies from architecture to another on the following basis MUX-based, systolic, RAM, reduced storage, and distributed [16].

While several 1-D and 2-D architectures have been introduced and evaluated, very few 3-D architectures have been reported. One main architecture has been introduced in [12], in which it operates on blocks of data rather than rows which reduces the complexity and size of the implementation. The 3-D wavelet transform computation is decomposed into three steps, each one is a 1-D transform in the x, y, and z directions. It functions as a completely integrated unit performing all of the required operations necessary to compute the wavelet transform. The Wavelet Processor (WP) requires a data block of size L1*L2*L3.

In this paper, an integrated design method for developing a 3D-processor prototype is presented based on the 3D architecture in [12]. Several design issues and parameters are addressed, analyzed and evaluated. The prototype processor consists of a single filter pair performing the calculation for one dimension. The output of each filter is stored in on-chip memory, it is considered as input for the next step. The filters are efficiently folded with an adequate amount of memory between the filters. The proposed architecture is area efficient, provides parallel execution for the filters and consumes less power

The organization of this paper is as follow: Section 2 introduces the 3D wavelet transform. The details of the prototype architecture is presented in Section 3. Section 4 decribes the implementation and simulation results and conclusions are summerized in Section 5.

0-7803-5471-0/99/$10.00©1999 IEEE

3D WAVELET TRANSFORM

The Discrete Wavelet Transform (DWT) has discrete values for its time and scale. In the DWT, a sub-band filter transforms a discrete-time signal into an average signal and a detailed signal to produce one level of resolution, called an Octave [17]. The average and detailed signals are produced using low and high pass filters, respectively. An iterative process of feeding the average signal into successive filters is performed, discarding the redundant higher octave average signals. The following two equations give the low pass and high pass outputs for a 1-D DWT.

The low-pass output is:

$$W(n,j) = \sum_{m=0}^{2n} W(m, j-1) * \text{lowfilter}(2n - m)$$

The high-pass output is

$$W(n,j) = \sum_{m=0}^{2n} W(m, j-1) * \text{highfilter}(2n - m)$$

where

j is the current octave.
n is the current input.
k is the current wavelet coefficient.
W(n,j) represents the DWT, except:
W(n,0) which is the input signal.

There are various architectures that compute the 1-D wavelet efficiently. One such architecture is based on the fast pyramid algorithm, proposed by Mallat and Meyer [18]. The number of wavelet coefficients plays a major role in the time taken to compute the wavelet transform, but since the number of coefficients are proportional to its accuracy, a trade-off must be made between the number of coefficients and the desired accuracy.

THE PROTOTYPE PROCESSOR

The prototype processor consists of one processing element, which contains both the high pass and low pass filters and 1 K on-chip cache. The wavelet coefficients are loaded into the 16-bit coefficient registers. Figure 1 shows a generic block diagram of the 3-DWT processor. The typical sequence of operations involves loading data blocks into the cache from an off-chip memory, while the transform coefficients are loaded into the coefficient register. Then the 3-D wavelet transform is scheduled using the coefficients and data block as 3 consequent 1-D wavelet transform. Figure 3 shows the complete system including all modules and their connectivity. In our proposed prototype, low power presents a major performance factor. Low power has been considered at several levels. It is achieved by applying the following strategies in the design flow:

1. Usage of specially designed low power building block cells such as adder, multiplier, etc.
2. Minimizing the number of processing elements by using the Central Control design introducing less circuit complexity in the architecture.
3. Maximizing the usage of sub-chip components by eliminating any redundant modules.
4. Making compromises concerning the tradeoffs of power, speed, and circuit complexity.

The High Pass and the Low Pass Filter

Both the High Pass and the Low Pass filters are identical with respect to their internal organization, Figure 1, although their respective coefficients vary. The filters include 16-bit Carry Look Ahead (CLA) adders and 16 bit Booth multipliers, to perform the arithmetic computations. After considering the attribute of many adders, the CLA was found to have the best possible trade-off in terms of power, speed, size and complexity. Figure 1 shows the filter computational tree where the * is the 16-bit Booth Multiplier and the '+' is the 16 bit-Carry Look Ahead adder, which uses a basic low power 1-bit adder cell [19]. The cell inherits a computational parallelism by using two independent data paths for calculating the Sum and the Carry. The output from the Booth Multiplier is 32 bits but the most significant 16 bits are discarded as described in [17]. After the layouts are generated, we use compactor to compact layout to a more regular shape. This compaction uses MOSIS-based rules, less compact each dimension independently.

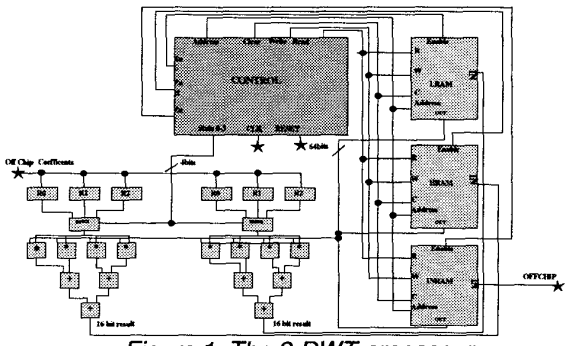

Figure 1: The 3-DWT processor

The Coefficient Module

The Coefficient Module, Figure 1, where DWTC is the discrete wavelet transform coefficient 64-bit register. The R0, R1 and R3 are three 64 bit registers that contain the discrete wavelet transform coefficient. The MUX is a 3-1 multiplexor, it selects which register is passed on to the control unit. Each 64-bit register is simultaneously loaded from an off-chip memory with four different 16-bit wavelet coefficients.

The Memory Module Unit

The prototype processor uses 1K on-chip cache memory. The cache unit is required for temporary storage of intermediate wavelet values. The unit consists of three independent cache modules, the INRAM, LRAM and HRAM. The INRAM module stores a whole block from an off-chip memory while the LRAM and HRAM modules contain the computed results from the filter modules. The INRAM has a 64-bits input/output data width, while both the LRAM and the HRAM has a 16-bit input data width, and 64-bit output data width. This is to accomplish the filters input/output width.

The cache modules are implemented using static memory cells, as opposed to dynamic memory cells, due to the reduction of the design complexity.

The Control Unit

The control unit coordinates the wavelet processor operations. It is basically a state machine, which consists of 25 internal cycles. Nine states are used for loading off-chip data, while the remaining states are used for the wavelet computation. Figure 2 illustrates the states of the controller. This sates show the flow of the required steps to compute the wavelet transform. The processor loads the eight data blocks from off-chip memory in states 1-8. The controller sends the necessary signals required for this operation. After the data is loaded from off-chip, intermediate results are calculated.

Both low pass and high pass filters generate the output using the off-chip data during states 9-16. The final values of the wavelet calculations are performed in states 17-25, using the intermediate data calculated in clocks 0-7 during clocks 9-16.

IMPLEMENTATION AND SIMULATION RESULTS

The chip has been implemented using 0.6 μm CMOS technology with three layers of metal. Synopsys™ tools are used for the "front-end" steps, which capture and simulate the desired behavior of the architecture, while Cadence ™ tools are used for the "back-end "steps.

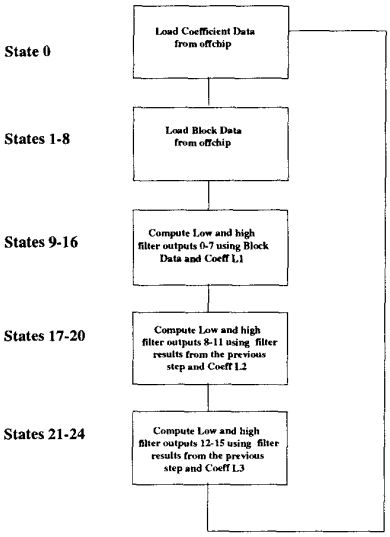

Figure 2: States of Controller

The behavior of the architecture is modeled using Verilog™, and it is synthesized, simulated and analyzed using Synopsys™. Then, Cadence ™ is used to do schematic capture and modification, using gate-level Verilog code to generate final layout.

The technology libraries and symbol libraries that were used in both Synopsys™ and Cadence™ for synthesis and simulation were generated based on the standard-cell library. This guarantees that the synthesized design has the nearest simulation results as the final layout.

Transferring the design from Synopsys™ to Cadence™ is accomplished by generating a Verilog™ netlist, which creates symbol and schematic views for all levels of the design hierarchy. Power and ground pads are then added to the top-level schematic.

Figure 3 shows the processor structure, where the different blocks are as follow: Block 1 is the INRAM. Block 2 is the LRAM. Block 3 is the HRAM. Block 4 is the High pass filter. Block 5 is the Low Pass filter. Block 6 is the get coefficient block for the high pass filter. Block 7 is the get coefficient block for the Low pass filter, and Block 8 is the controller.

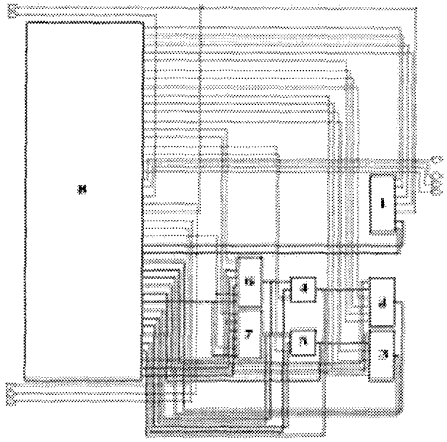

Figure 3: The Processor Structure

Two groups of simulation are used, the first one, proving the correctness of the design at different design steps, while the second one measuring different performance parameters for the proposed architecture.

Many verification simulation steps are involved in the design flow. The first one, uses Verilog simulation tools to verify the correctness of the architecture's behavior. The second is applied to confirm the functionality of the extracted gate level circuit. The third ensures that the gate-level schematic, which has been imported and edited in Cadence™, is equivalent to the proposed architecture. These are followed by the DRC (Design Rule Checking) and the LVS (Layout Versus Schematic), which are done with Cadence Diva. Once LVS is done, it is assumed that the layout is functionally correct. Figure 4 show the processor layout, which has a total area of 9935.50 x 6236.40 μm^2. The evaluation of the implemented architecture considers different parameters as follows:

Area

The prototype of the architecture uses 168k transistors, the areas of the main building blocks are as follows: (Table 1)

The block	*The Area in (μm^2)*
16-bit adder	*368.80 X 99.70*
16-bit booth multiplier	*954.50 X 476.60*
RAM	*1883.50 X 1753.60*

Table 1: Area of different building block

Power Consumption

According to the Synopsys™ simulation, the architecture is expected to consumes power 0.5 W. Table 2 shows the power evaluation of the main building blocks.

The block	*The Power (mW)*

16-bit adder	*5.5828*
16-bit booth multiplier	*13.0729*
The processor	*500*

Table 2: The Power of different building block

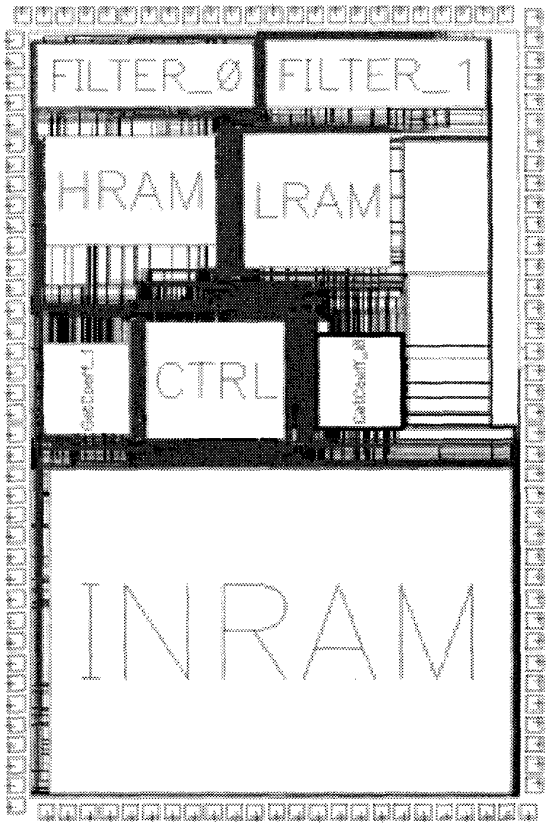

Figure 4: The Processor Layout

Latency

The latency measures the speed of the operation. In this evaluation, we express the latency as a function of the input rate. Table 3 shows the time delay for the building blocks. The latency for the processor is to perform the 25 states, described in Section 3.

The block	*The delay (ns)*
16-bit adder	*4.7*
16-bit booth multiplier	*8.93*
The processor	*91.65*

Table 3: The delay of different building block

CONCLUSIONS

This paper presents a low power prototype for a 3-D DWT processor based on centralized control unit architecture. The simulation results show the efficiency of the wavelet processor. The prototype processor consumes 0.5 watt with total delay of 91.65 ns. The processor operates at max frequency of 272 Mhz.

ACKNOWLEDGEMENT

The authors acknowledge the support of U.S. Department of Energy (DoE), EETAPP program (1997-2000) and the support of the LEQSF (1996-1999) RD-B-13, State of Louisiana.

REFERENCES

[1] Keshab K. Parhi, Takao Nishitani, "VLSI Architectures for Discrete Wavelet Transforms," IEEE Trans. on Very Large Scale Intergration (VLSI) Systems, *Vol. 1, No 2, Jun 1993, pp 191-202.*

[2] Jan Ozer, "New Compression Codec Promises Rates Close to MPEG," CD-ROM Professional, *Sept. 1995, pp 24.*

[3] Angela Hickman, John Morris, Carol Levin, Sebastian Rupley, and David Willmott, "Web Acceleration," PC Magazine, *June 10, 1997, pp 10.*

[4] W. W. Boles and Quang Minh Tieng, "Recognition of 2-D Objects from the Wavelet Transform Zero-crossing Representation," Proceedings SPIE, *Vol. 2034, Mathematical Imaging, San Diego, July 11-16, 1993, pp 104-114.*

[5] W. Wayt Gibbs, "Making Wavelets," Scientific American, *June 1996, pp 137-138*

[6] Neil Weste and Kamran Eshraghian, Principals of CMOS VLSI Design, *Addison-Wesley, Reading Massachusetts, 1993.*

[7] M. Vishwanath, R. M. Owens, and M. J. Irwin, "Discrete Wavelet Transforms in VLSI," Proc. of the Int. Conf. on Application Specific Array Processors, *Berkeley, August 1-2, 1992, pp 218-229.*

[8] Jimmy Limqueco and Magdy Bayoumi, "A 2-D DWT Architecture," Proc. of the 39th Midwest Symp. on Circuits and Systems, *Iowa State University, Ames, Iowa, August 18-21, 1996, pp 1239-1242.*

[9] Jimmy Limqueco and Magdy Bayoumi, "A Scalable Architecture for 2-D Discrete Wavelet Transform," VLSI Signal Processing IX, *San Francisco, Oct. 30-Nov. 1, 1996, pp 369-377.*

[10] Jimmy Limqueco and Magdy Bayoumi, "A VLSI Architecture for Separable 2-D Discrete Wavelet Transform," Journal of VLSI Signal Processing, *Vol 18, 1998, pp 125-140*

[11] Keshab K. Parhi and Takao Nishitani, "VLSI Architectures for Discrete Wavelet Transforms," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, *Vol. 1, No. 2, 1993, pp 191-202.*

[12] Michael Weeks, Magdy Bayoumi, "3-D Discrete Wavelet Transform Architectures," IEEE Int. Symposium on Circuits and Systems (ISCAS '98), *Monterey, California, May 31 - June 3, 1998.*

[13] G. Knowles, "VLSI Architecture for the Discrete Wavelet Transform," Electronics Letters, *Vol 26, No. 15, 90, pp. 1184-1185.*

[14] Jose Fridman and Elias S. Manolakos, "Distributed Memory and Control VLSI Architectures for the 1-D Discrete Wavelet Transform," IEEE Proc. VLSI Signal Processing VII, *La Jolla, California, October 26-28, 1994, pp. 388-397.*

[15] Jose Fridman and Elias S. Manolakos, "Discrete Wavelet Transform: Data Dependence Analysis and Synthesis of Distributed Memory and Control Array Architectures," IEEE Transactions on Signal Processing, *Vol. 45, No. 5, May 1997, pp 1291-1308.*

[16] Chaitali Chakrabarti, Mohan Vishwanath, and Robert Owens, "Architectures for Wavelet Transforms: A Survey," Journal of VLSI Signal Processing, *Vol. 14, No. 1, 1996, pages 171-192*

[17] Michael Weeks, "3-D Discrete Wavelet Transform Architectures ", Ph.D Dissertation. University of Southwestern LA, 1998

[18] S. Mallat "A Theory for Multiresolution Signal Decompisition: The Wavelet Representation," IEEE Pattern Analysis and Machine Intelligence, Vol.11, No. 2, June 1993, pp 647-693

[19] W. M. Badawy, A. M. Shams and M. A. Bayoumi "An Enhanced Low-Power Pipelined Multiplier-Accumulator for DSP Applications", The 7Th NASA Symp. on VLSI, *Albuquerque, NM Oct.1-2,98.*