A Low-power Pipelined Implementation of 2D Discrete Wavelet Transform

Yong Liu[1], Edmund M-K. Lai[1], A.B. Premkumar[1] and Damu Radhakrishnan[2]

[1]School of Computer Engineering, Nanyang Technological University, Singapore 639798.
[2]Department of Electrical & Computer Engineering, State Univ. of New York, New Paltz, NY, USA.

ABSTRACT

Discrete wavelet transform has been incorporated as part of the JPEG2000 image compression standard and is being deployed in various portable consumer products. This raises the interest in low-power design of DWT processor. This paper presents a low-power implementation of a 2-D biorthogonal DWT processor that uses residue number arithmetic. By incorporating a 4-stage pipeline, the processor is able to sustain the same throughput with a lower supply voltage. Hardware complexity reduction and utilization improvement are achieved by resource sharing. Our implementation results show that the design is able to fit into a 1,000,000-gate FPGA device.

1. Introduction

The developments on wavelets and wavelet transforms in recent years have led to numerous applications in areas such as computer vision, image compression, denoising noisy data, and sound synthesis. Recently the image compression standard JPEG2000 has incorporated discrete wavelet transform (DWT) into its core specifications [1]. As a result of standardization, DWT is now becoming popular in various portable consumer-imaging products.

As discrete wavelet transform (DWT) requires intensive computation, for some real-time applications it is necessary to implement the wavelet transform algorithms in hardware so that timing constraints are met. DWT can be realized in hardware using FIR filters [3], which basically only involve additions and multiplications. How these arithmetic operations are implemented in hardware depends on the number system used to represent the data. Residue number systems (RNS) are suitable for implementations of high-speed digital signal processing devices because of their inherited parallelism, modularity, fault tolerance and localized carry propagation [4]. Arithmetic operations, such as addition and multiplication, can be carried out more efficiently in RNS than in conventional two's complement systems. Hence that makes RNS a good candidate for implementing DWT.

In order to prolong the battery life of the portable devices, low-power design is one of the key issues when designing the DWT processor for those applications. Voltage scaling together with pipelining has been proven to be an effective way to achieve low-power design [8,9]. In this paper, we present our design and implementation of a low-power pipelined RNS-based biorthogonal 2-D DWT processor using the Daubechies 9/7-tap filter banks. The requirements for filter coefficient word length are studied. The RNS-based FIR filter bank data path, controllers and memory interface are designed and simulated using a Field Programmable Gate Array (FPGA) development board. This design can be implemented on an ASIC with minimal modifications.

2. Dynamic Range Determination and Moduli Set Selection

Initial experiments were conducted to determine the word length required for RNS filtering computations. With the assumption that input image data are represented in 8 bits, experimental results show that the use of 24-bit dynamic range is able to obtain reconstructed images with PSNR values higher than 54dB. Therefore, we consider that a dynamic range of 24 bits as adequate.

The choice of a particular moduli set for RNS implementation of the filter modules was the next issue. This choice depends on various factors like, area, delay, power etc. We developed a software tool, called MODS, to automate the process of moduli selection based on the above factors. The power consumption of an RNS-based sub-filter tap, denoted as PPT is given by

$$\text{PPT} = \sum_{i=0}^{m-1} P(m_i) + \left(\sum_{i=0}^{m-1} FC\_P(m_i) + RC\_P\left(\{m_i \,|\, i \in [0, m-1]\}\right) \right)$$

where $P(x_i)$ is the power consumed by *a* modulo x_i sub-filter tap, $FC\_P(x_i)$ is the power consumed by *a* modulo x_i forward converter, and $RC\_P(\{x_1, x_2, ..., x_m\})$ is the power consumed by the reverse converter for a specific moduli set $\{x_1, x_2, ..., x_m\}$. Similarly, the critical path delay per tap, denoted as CPD, is given by

$$CPD = Max(D(m_i)) + Max(FC\_D(m_i)) + RC\_D\left(\{m_i \,|\, i \in [0, m-1]\}\right)$$

By making use of the above expressions, for a 24 bit dynamic range, MODS came up with a number of moduli sets as shown in Table 1 [7]. The data in this table is obtained by using Synopsis Design Analyzer and Avant! Passport 0.35-micron 3.3V optimum silicon Technology library. It shows that the triple moduli set {257,256,255}, and the 6 moduli set {33,32,31,17,7,5} provides the shortest delay and least power respectively, compared to other moduli sets. The triple moduli set is of the form $\{2^n\text{-}1, 2^n, 2^n\text{+}1\}$, which has the advantage of low cost forward conversion and modulo reduction. The reverse conversion architecture is also relatively simple. Furthermore, the triple moduli set will have uniform channel width for the filter modules compared to the nonuniform channel width imposed by the 6 moduli set. As a result, the use of the triple moduli set can significantly reduce hardware complexity and delay [5] and it is used in this implementation.

Table 1. Moduli Sets and Estimated Cost for RNS Based 3-level 2-D DWT

Dynamic Range		**24 bits**			
Filter Taps	**Filter Bank type and levels**		**Time critical**	**Area critical**	**Power critical**
LP = 9 HP = 7	**Type = 2** Level = 3	**Recommended moduli set**	**{257,256,255}**	**{32,31,17,15,11,7}**	**{33,32,31,17,7,5}**
		Delay (ns)	**43.77**	**62.0**	**59.5**
		Area/tap	**2270.5**	**1935**	**1942.0**
		Power/tap (mW)	**62.45**	**55.68**	**54.24**

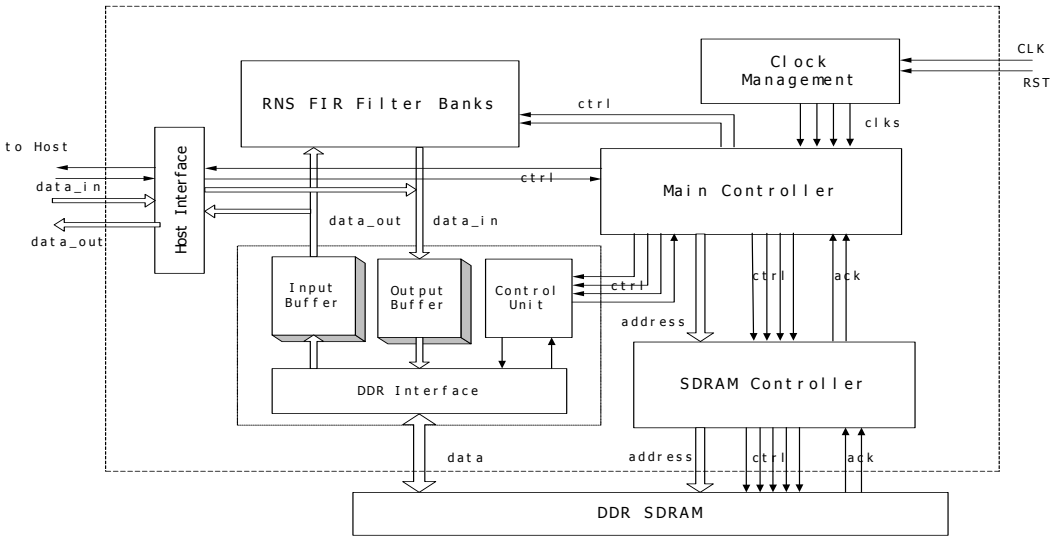

Figure 1. Organization of the 2-D DWT Processor

3. DWT Processor Design

The 2-D DWT processor consists of the following key components: RNS-based FIR filter bank data path, main controller, buffering and DDR interfacing unit, SDRAM controller, clock management circuitry and the host interface. A block diagram of its structure is shown in Figure 1.

3.1. The RNS-based FIR Filter Bank Unit

The FIR filter bank unit consists of two filter banks (filter bank 0 and filter bank 1). Each has a low pass filter module (LPF) and a high pass filter module (HPF). The choice of our triple moduli set {257,256,255} resulted in three separate channels for the implementation of each filter module. Each set is made up of three forward converters and three sub-filter banks corresponding to the three moduli in our chosen moduli set, as well as a reverse converter. A block diagram of the filter bank module is shown in Figure 2.

Forward converter architectures, which can handle signed numbers, for modulo-255 and modulo-257 channels are modified versions of the ones discussed in [6] and [2] respectively. Note that forward conversion for the modulo-256 channel is achieved simply by keeping the least significant 8 bits of the 2's complement data. The reverse converter structure used was based on the design presented in [5] with additional comparison and subtraction circuits for coping with signed numbers.

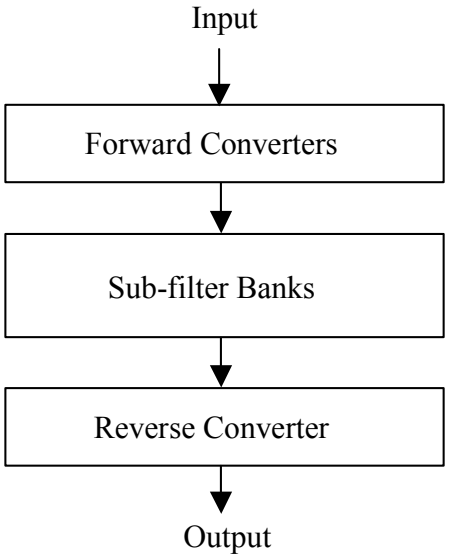

Figure 2. Block Diagram of the RNS- based FIR

The FIR sub-filters in this design are realized in transposed form. It involves modulo multipliers and adders. Since the filter coefficients are known a priori, it is possible to implement the modular

multiplier as look-up tables (LUT), which can reduce both the delay and hardware cost. In our design, 27 look-up tables (LUT), each with 8-bit width and 256 entries, are required to store all the results of modulo multiplications. Since very few results of the modulo-257 channel have a non-zero 9^{th} bit (having the output value of 256), storage for the 9^{th} bit is saved by substituting in some combinational logic to generate the 9^{th} bit of the output.

3.2. The Main Controller

The main controller, which is implemented as a finite state machine (FSM), is responsible to control the operations of the processor, such as DWT processing, communication with the host and data scheduling. The main controller enters initialization mode after reset. During initialization, the main controller sends out reset signals to all other units, initializes its internal counters and waits for the external SDRAM to be ready. After the processor is initialized, the main controller requests image data from the host and stores them into the external SDRAM. When all image data are ready, the main controller instructs the filter banks to perform 2-D DWT on the image data. Because the capacity of the internal RAM is not large enough to store a complete image in most cases, only part of an image will be read into the internal RAM for processing. After it is processed and stored back to the external SDRAM, other part of the same image will then be read into the internal RAM for processing. This process repeats until all parts of an image are processed and the processor will continue to process next image. After required processes are finished, the processed data are sent back to the host.

The data scheduling involves reading in correct data to the associated buffers of the row and column DWT filter banks from the external storage. Since the amount of data processed by each of the stage 2 filter banks will be half of that processed by the stage 1 filter bank, only one filter bank in the second stage is sufficient to process both the high pass and low pass outputs from the first stage without delaying the whole process. Therefore, we configured filter bank 0 to perform row processing and filter bank 1 to perform column processing.

Data scheduling for both the filter bank 0 and filter bank1 is illustrated in Fig. 3. In Cycle *X*, filter bank 0 performs row processing on Image *X* and filter bank 1 performs column processing on sub-images of Image *X-1*. In this manner, we can fully utilize the two filter banks and obtain the decomposed data at each DWT processing cycle (except the starting and the ending cycles).

3.3. Other Function Units

The buffering and DDR interfacing unit implements two functions. Firstly, it provides a dual clock rate (DDR) data interface between the DWT processor and the external DDR SDRAM. Secondly, it buffers the data read from the external SDRAM and the processed output data from the filter bank unit. Symmetric extension of the input data is also accomplished by properly presenting the buffered data to the filter banks.

The clock management unit, which is implemented by using the dedicated digital clock management block of the FPGA device, generates clock signals with the same frequency but different phases for the processor. The host interface allows the 2-D DWT processor to communicate with the host computer for downloading and uploading image data.

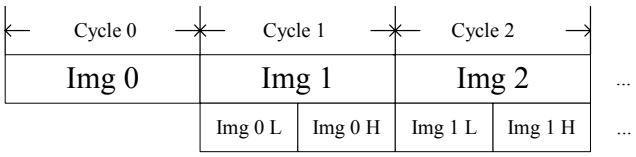

Figure 3. Data Scheduling

4. Low Power Processor Design

If the filter bank is implemented as shown in Figure 2, the maximum frequency of operation was found to be 15.15MHz. Assuming that the full image size is 512x512 pixels and a frame rate of 24 frames per second is required, the filter bank unit is able to satisfy the throughput requirements. However, the speed can be increased and the power consumption of the processor can be reduced by the use of pipeline operation.

4.1. Pipelining of the RNS-based Filter Banks

A four-stage pipeline, depicted in Figure 4, is introduced in the RNS-based filter bank design. The first stage consists of the forward converters, the second stage the LUT-based multipliers, the third stage the modular adders and the last stage the reverse converter. Since the data path only performs fixed filtering task, the introduction of the 4-stage pipeline only incurs small area penalty for the pipeline registers and presents almost no control complexity. Because the delay for each stage along the pipeline is different and the clocking frequency of the pipeline is limited by the slowest stage (reverse converter), the speedup was found to be 185%.

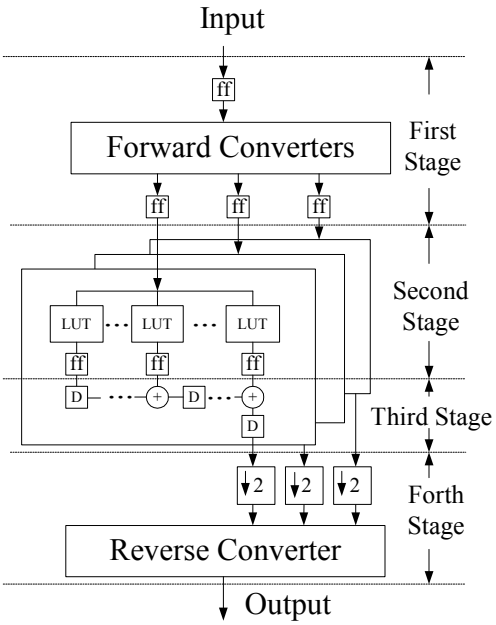

Figure 4. Pipelining of the RNS-based FIR Filter Bank

Reducing the supply voltage can slow the speed of the filter bank units down further. The reduction of supply voltage brings a quadratic reduction in power consumption. The above approach resulted in a voltage reduction from 5V to 2.7V in our design. Neglecting the power consumed by the extra registers used in pipeline operation, the power reduction achieved amounts to $(5^2-2.7^2)/5^2 \times 100 = 70\%$ assuming a non-pipelined voltage supply of 5V.

Compared to the reverse converter, the delay for accessing look-up tables is much shorter, therefore, area complexity can be reduced by sharing the same set of look-up tables between the two filter banks, as illustrated in Figure 5**.** In each cycle, the look-up tables are first accessed by inputs to filter bank 0 and the outputs are kept in a set of registers. Next, the look-up tables are accessed by inputs to filter bank 1. Even through the look-up tables are accessed sequentially, the delay is still shorter than that of the reverse converter. Hence, significant amount of area is reduced without incurring any extra delay for the processor.

The output of each filter is down-sampled by two to reduce the data rate after filtering. This down-sampling operation is performed before the reverse conversion as shown in Fig. 6. At the cycle where the output needs to be kept, RNS outputs from the low pass filter are fed into the reverse converter, while those from the high pass filter are stored in registers. At the cycle where the output needs to be discarded, reverse conversion is performed on the output from the high pass sub-filter of the previous cycle, which is stored in a register. Hence, the reverse converters are utilized fully and only one reverse converter is required for each filter bank.

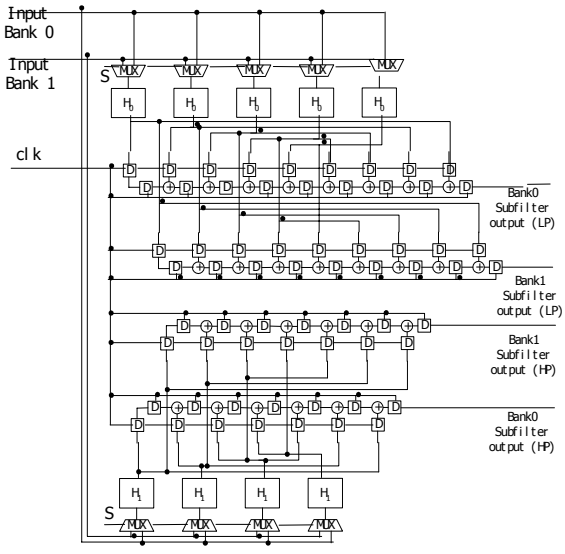

Figure 5. Two RNS-based FIR Sub-filter Banks with LUT Sharing

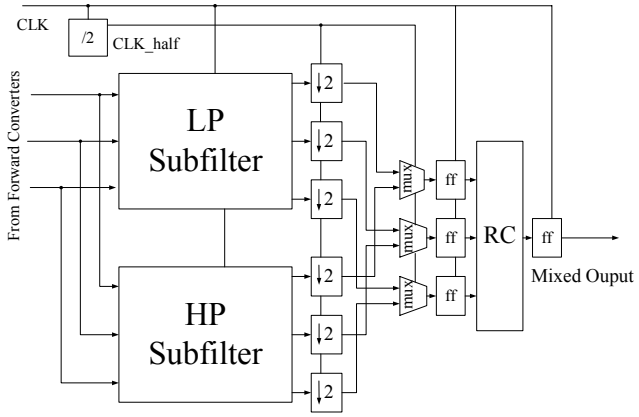

Figure 6. FIR Filter Bank with Reverse Converter Sharing Between LPF and HPF

5. Implementation Results

The DWT processor, which is able to process 32×32-pixel images, is coded in VHDL and synthesized into the target FPGA device and simulated. These VHDL models are parameterized so that the synthesized processor can process larger images. The complexity of various controllers is independent of the size of the image, but the size of input and output buffers need to increase to cope with the larger images. Table 2 shows the detailed timing for different parts of the RNS-based filter bank. Table 3 shows the FPGA resources consumed by various modules of the DWT processor. Altogether it occupies 93% of the slices and 20% of the block RAM available in the target FPGA device.

If all control units and the DDR interfacing and buffering unit is working under 100 MHz, then the RNS-based filter banks can be clocked by a 25 MHz clock signal derived from the main clock. From the simulation results, under these clock frequencies and a burst length of four for the external SDRAM, it takes 205 μs to finish a first level 2-D DWT decomposition of a 32×32 image, of which 72 μs is for accessing the external SDRAM and 133 μs is for FIR filtering.

Table 2 Timing Details for the RNS-based Filter Bank (ns)

Channels	Forward Converter	Sub-filters		Reverse Converter
		LUT	Mod Adder and Delay	
255	16.889	6.916	6.776	32.577
256	0	6.148	4.261	
257	18.837	6.933	7.571	

Table 3 FPGA Resource Consumption

	Slices	Flip-flop	4-input LUT	Block RAM	Max freq (MHz)
RNS Filter Banks	3335	1634	6070	0	28.043
Main Controller	453	172	860	0	100
DDR Interfacing and Buffering	865	505	1542	4K Bytes (8 blocks)	100
SDRAM Controller	140	73	263	0	122

6. Conclusions

This paper reports on the design and implementation of a 24-bit low power pipelined 2-D biorthogonal DWT processor based on RNS arithmetic. A 4-stage pipeline is incorporated in the design of the RNS-based DWT filter bank data path with minimal area overhead. By lowering the supply voltage, power consumption is reduced by 70% while sustaining the same data throughput for the data path. As pointed out in [10], it is possible to further reduce the power consumption of the processor because of the parallelism and switching activity reduction provided by RNS.

Our software tool (MODS) recommendation was a 6 moduli RNS implementation for achieving the least power. But a 6 moduli implementation would have resulted in non-uniform length channels for the filter modules. Hence, we used a used a triple moduli implementation for the RNS based filter bank unit. Further research could use the low power moduli set

(6 moduli set) implementation for the filter modules to reduce the power consumption of the processor.

Acknowledgements

The first two authors would like to acknowledge the technical support provided by the Digital Signal Processing Laboratory, the Centre for High Performance Embedded Systems and the Centre for Multimedia and Network Technologies, School of Computer Engineering, Nanyang Technological University.

References

[1] ISO/IEC FCD 15444-1 2000v1.0 JPEG 2000 Image Coding System.

[2] F. Pourbigharaz and H. M. Yassine. "Simple binary to residue transformation with respect to 2^m+1 moduli". *IEE Proceedings on Circuits Devices and Systems*, Vol. 141, No.6 pp. 522-526, Dec. 1994.

[3] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley: Wellesley-Cambridge Press, 1996.

[4] N. S. Szabo and R. I. Tanaka. *Residue Arithmetic and its Applications to Computer Technology*. New York: McGraw-Hill, 1967.

[5] Yuke Wang, Xiaoyu song, Mostapha Aboulhamid and Hong Shen. "Adder Based Residue to Binary Number Converters for $(2^n-1, 2^n, 2^n+1)$". *IEEE Trans. on Signal Processing*, Vol. 50, No. 7, pp. 1772-1779, Jul. 2002.

[6] B. Parhami. *Computer Arithmetic: Algorithms and Hardware Designs*, Oxford University Press, New York, 2000.

[7] Yong Liu and Edmund M-K Lai. "Moduli Set Selection and Cost Estimation for RNS-based FIR Filter and Filter Bank Design". Submitted to *Design Automation for Embedded Systems*, Kluwer Academic Publishers.

[8] S-M Kang and Y. Leblebici. *CMOS Digital Integrated Circuits-Analysis and Design*, McGraw-Hill, 2003.

[9] G. K. Yeap and F. N. Najm (Eds.). *Low Power VLSI Design and Technology*, World Scientific Singapore, 1996

[10] T. Stouraitis and V. Paliouras. "Considering the Alternatives in Low-Power Design". *IEEE Circuits and Devices Magazine*, Vol. 17, No. 4 pp. 22-29, Jul. 2001.