A Scalable Embedded JPEG2000 Architecture

Chunhui Zhang, Yun Long, and Fadi Kurdahi

Department of EECS, University of California,
ET508, zotcode 2625, UCI, Irvine, CA, 92697, USA
{chunhuiz, longy, kurdahi}@ece.uci.edu

Abstract. It takes more than a good tool to shorten the time-to-market window: the scalability of a design also plays an important role in rapid prototyping if it needs to satisfy various demands. The design of JPEG2000 belongs to such cases. As the latest compression standard for still images, JPEG2000 is well tuned for diverse applications, raising different throughput requirements on its composed blocks. In this paper, a scalable embedded JPEG2000 encoder architecture is presented and prototyped onto Xilinx FPGA. The system level design presents dynamic profiling outcomes, proving the necessity of the design for scalability.

1 Introduction

JPEG2000 is the latest compression standard for still images [1]. Due to the adaptations of the discrete wavelet transform (DWT) and the adaptive binary arithmetic coder (Tier-1), JPEG2000 provides a rich set of features not available in its predecessor JPEG. In particular, the core algorithm of the standard addresses some of the shortcomings of baseline JPEG by supporting features like superior low bit-rate performance, lossy to lossless compression, multiple resolution representation, embedded bit-stream and so forth.

Several JPEG2000 encoder architectures have been implemented [2][3][4]. They employed a fixed number of dedicated hardware accelerators for the two compute-intensive blocks, DWT and Tier-1. However, JPEG2000 aims at a broad application scope, thus the relative processing demand on DWT and Tier-1 varies tremendously as the situations alter (e.g. image type as compression ratio). Therefore, such rigid architectures, which restrict the relative throughput between DWT and Tier-1 only optimal at specific conditions, will easily lose the balancing points and even be severely unbalanced under a changed circumstance.

In this paper, a scalable JPEG2000 encoder is presented and prototyped onto Xilinx Virtex-II Pro FPGA. It takes the advantages of Virtex-II Pro's embedded PowerPC RISC core and the flexible on-chip bus structure. The SW/HW (Software/Hardware) partition scheme used is based on our profiling experiments and the hardware accelerators are carefully designed for scalability concern. Beside the performance advantages afforded by the customized hardware cores, the available scalability facilitates the throughput exploration for a given specification. The paper is organized as follows: In Section 2, the proposed JPEG2000 system design is discussed. The two main hardware accelerators for DWT and Tier-1 are presented in Section 3. Section 4 gives the prototyping results together with the scalable performance. The paper is then concluded in Section 5.

T.D. Hämäläinen et al. (Eds.): SAMOS 2005, LNCS 3553, pp. 334–343, 2005.

2 JPEG2000 System Design

Before prototyping the encoder onto FPGA, intensive simulations have been done to verify the design. The specification is coded in floating-point C at the beginning and then translated into fixed-point for hardware verification.

2.1 JPEG2000 Overview

A typical JPEG2000 encoder is composed of the fundamental building blocks shown in Fig. 1. The decoding process is symmetric to the encoding but in the reverse direction. This subsection gives a brief overview of the encoding steps and the details are referred to the standard [1].

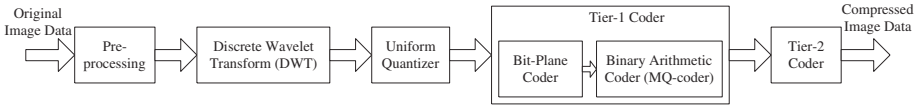

Fig. 1. JPEG2000 fundamental building blocks

During encoding, the image is partitioned into rectangular and non-overlapping tiles which are processed separately. Next, unsigned sample values are level shifted to make them symmetric around zero. Counting the optional inter-component transform (ICT), those procedures are summarized as pre-processing. The dyadic DWT is applied on the tile repeatedly to de-correlate it into different decomposition levels (resolutions). For lossy compression, the wavelet coefficients are fed to a uniform quantizer with central deadzone.

Each subband is further divided into rectangular blocks namely, code blocks, which are entropy-coded independently. In JPEG2000, entropy coding is two-tiered. Tier-1 is a context-based adaptive arithmetic coder composed of Bit-Plane Coder (BPC) and Binary Arithmetic Coder (MQ-coder). It accepts the quantized wavelet coefficients along with their corresponding probability estimates generated by the BPC, and produces highly compressed codestream. This codestream is carefully organized by Tier-2 coder, constructing a flexible formatted file. New terms and techniques like precinct, packet, tag-tree, and rate control enable features like random access, region of interest coding and scalability.

2.2 Profiling and SW/HW Partitioning

Although C open sources for JPEG 2000 are available, e.g. Jasper [1], almost all of them are software-oriented. Their merits of full specification capture and extra facilities bring complicated design and inferior performance adversely. Therefore, we program our own hardware-oriented JPEG 2000 encoder.

We profile the C implementation of the encoding algorithm on a standard PC (Pentium IV 2.4 GHz, 512M RAM) for rough software estimation and subsequent SW/HW

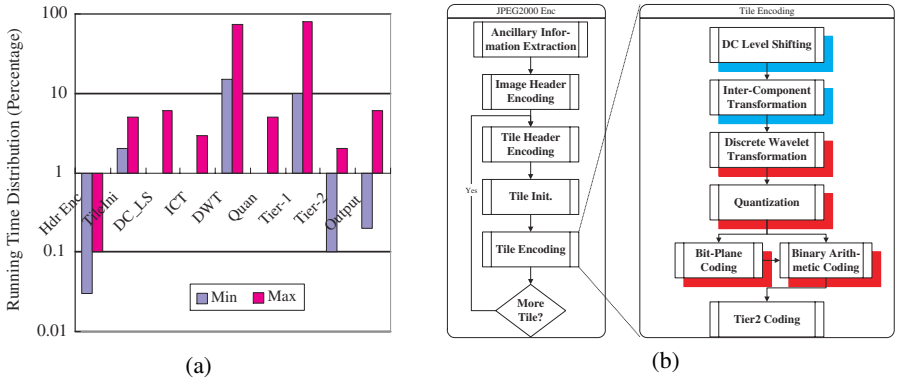

Fig. 2. (a)The profiling results; (b) Proposed partitioning schemes

partitioning. Based on the profiling results shown in Fig. 2 (a), two candidate partitions are given in Fig. 2 (b). DWT and Tier-1 coder are computationally, as well as memory, intensive. Therefore, partition 1 allocates DWT, Quantizer and Tier-1 coder into hardware part while the rests run in the host processor in software manner. Partition 2 further assigns DC level shifting and ICT to hardware. By contrast, Tier-2 encoder is computationally inexpensive and showing inferior data locality. Therefore, Tier-2 is appointed to software although it "appears" complicated. Besides performance improvement through hardware customization, the partitioning schemes also decrease communication budgets by confining Tier-2 and the initialization parts in the host processor.

An important phenomena can be noticed via the profiling — the distribution of run time over the blocks varies with large dynamic ranges, prominent for DWT and Tier-1. Actually, Fig. 2 gives two run times, *Min* and *Max*, for each block as the bounds (those optional blocks have zero *Min*). The relative throughput between DWT and Tier-1 alters over ten times in common configuration ranges (e.g. compression ratio from about 2 to 50). When the compression ratio is low (as lossless as the extreme in medical image processing), all information are preserved for Tier-1 processing, whereas most of which could have been eliminated away by the quantizer when the compression ratio becomes high.

2.3 Proposed JPEG2000 Architecture

We propose a system level architecture for JPEG2000 encoding core algorithm targeting toward Xilinx Virtex-II Pro. Fig. 3 shows the diagram based on XC2VP7, a cheap device in Virtex-II Pro family. While the Virtex-II Pro FPGA can be embedded with up to 4 PowerPC cores and 556 multipliers, it is ideal for single-chip embedded system design with scaling up potential. The corresponding decoder can be realized in almost the identical structure by inverting the data path.

Both of the two partition schemes presented in section 2.2 are considered in the architecture. Tier-2 is completed in PowerPC, while DWT and Tier-1 are implemented using hardware-specific design with FPGA logic. The remainder optional blocks (DC_LS,

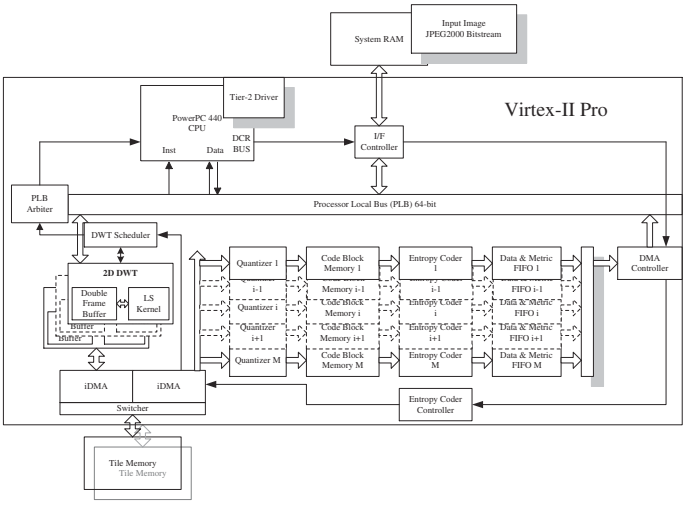

Fig. 3. Proposed JPEG2000 encoder based on Virtex-II Pro XC2VP7

ICT and Quantization) can be either inserted into the data path or run on the PowerPC because of their fine granularity (full parallelism and scalability) and lesser computational requirements. The 64-bit width PLB bus sustains most data communication loads and the DCR bus is utilized for control and parameter-tracing purposes.

The salient feature in the proposed architecture is its scalability. The major blocks, DWT and Tier-1, are configurable with parameters N and M which represent the numbers of DWT and Tier-1 hardware modules respectively. N and M are constrained by hardware resources. The corresponding peripherals can be also scaled to sustain the data consuming and delivering. The dedicated hardware accelerators with their scalability will be detailed in Section 3.

2.4 Scheduling Scheme

JPEG2000 fundamental blocks are divided into two parts, DWT part (including inter-component transform, DC level shifting and DWT) and Tier-1 part (Quantization, Tier-1 and Tier-2). Table 1 lists the comparison of four practical scheduling granularity (S1-S4): tile, resolution, subband and precinct. Synchronization here is used as between the two parts. S1 has simple synchronization scheme in contrary with S3 and S4. Furthermore, S3 complicates Tier-2 and S4 conflicts with DWT characteristics. Between the two preferable candidates, S1 shows advantages in most aspects over S2, such as synchronization, memory size and Tier-2 coding. Throughput of S1 and S2 are the same for continuous tiles processing. Smaller latency maybe the only merit introduced by S2. Therefore, S1 is the most appreciated choice which uses tile as the grain size.

The original images, logically divided into tiles, are stored in off-chip memory. Every time, one new tile is transferred to the DWT accelerator through the PLB bus. The decomposed data are stored inside Tile Memory which can contain two tiles and switch in a “ping-pong” way. Concurrently, the previous tile which has just finished DWT pro-

Table 1. Comparison of different scheduling granularity

Scheduling Granularity	Tile (S1)	Resolution (S2)	Subband (S3)	Precinct (S4)
Synchronization	Simple	Moderate	Hard	Hardest
Memory size	2T+1P[1]	2T + 1 compressed T	Almost no saving	Almost no saving
Throughput	$1/max\{T_{DWT}, T_{T1}\}$[2]			
Latency	$T_{DWT}+T_{T1}$ $+T_{PT2}$ [3]; or $2*T_{T1}+T_{PT2}$	Potential Improvement	Smaller than S2	Smallest
T2 coding	Forward	Reverse	Complex	Reverse

cessing starts the procedure of Tier-1 encoding. Tier-1 can be processed in the order of the final bitstream. Thus Tier-2 is encoded in a straightforward way without large amount of buffering.

3 Main Hardware Accelerators

Customized designs for DWT and Tier-1 are primary for our JPEG2000 architecture on Xilinx platform. The C codes of the two blocks are replaced by their VHDL implementations at RTL level. We designed all four DWT engines: default lossless and lossy — LeGal (5, 3) and Daubechies (9, 7); forward and inverse transformations, respectively. Due to the space concern, only forward Daubechies (9, 7) DWT is discussed in this paper.

3.1 Scalable DWT Architecture

Lifting Scheme. The basic principle of lifting scheme is to factorize the polyphase matrix of a wavelet filter into a sequence of alternating upper and lower triangular matrices and a diagonal matrix [5]. Following is one factorization form,

$$A(z) = \begin{bmatrix} H_e(z) & H_o(z) \\ G_e(z) & G_o(z) \end{bmatrix} \tag{1}$$

$$A(z) = \left(\prod_{i=1}^{m} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \right) \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \tag{2}$$

where $s_i(z)$ and $t_i(z)$ are Laurent polynomials, which are called prediction and updating operations respectively. Lifting scheme has several advantages, such as the reduction in the number of multiplications (odd-tap filters only) and additions, "in-place" computation, symmetric forward and inverse transform, etc. Although problems of border mirroring, synchronization and programmability caused by lifting scheme have restrained some designers back to the classical convolution way [6], lifting scheme is

[1] T stands for "Tile", and P stands for "Precinct"

[2] T_{DWT} and T_{T1} mean the processing times of DWT and Tier-1 for one tile

[3] T1 stands for "Tier-1" and T2, "Tier-2"; T_{PT2} is the time for one precinct Tier-2 coding.

widespread [7][8]. We find that the main contribution of "in-place" computation is not to save storage, but to simplify the control. Therefore, lifting scheme is selected in our design for the sake of computation reduction while the "in-place" feature is replaced by "switching frame buffers" for an even simpler control. Although frame buffers are doubled, the whole storage increases less than 1 percent for a 512 by 512 image.

Fixed-Point C Implementation and Precision Analysis. The floating-point C implementation is translated into fixed-point for the precision analysis. We carried out experiments on the peak signal to noise ratio (PSNR) under different filter coefficient word length and concluded that the PSNR saturates at 10-bit. For an 8-bit decompressed image, PSNR is defined as $10log_{10}\frac{255^2}{MSE}$, where MSE refers to the mean squared error between the original image and the reconstructed image. The input signals are shifted left to decrease the normalization errors. EB, Extra Bits, is introduced as the number of shifting bits. Extensive simulations proved that 16-bit signal width with $EB = 5$ satisfy both the required accuracy and the dynamic range.

"Software Pipelined" Lifting Scheme Kernel (LS Kernel). We proposed a "software pipeline" method to implement lifting scheme, referred to [9] for details. Illustrated in Fig. 4 (a), the conventional step by step "zigzag" data flow way is replaced by a single-step scan. Each shadowed diamond is a lifting scheme element (LS element), containing 2 additions and 1 multiplication. A LS Kernel is composed of four LS elements and 2 extra multiplications, delimited by two neighboring dash lines. In order to eliminate data dependencies, software pipeline technique is applied, extracting parallelism among LS kernels thoroughly. Our approach largely improves the data reuse, reducing memory access count about 5 times for Daubechies (9, 7) transform.

Hierarchical Pipelining. The overall DWT architecture is composed of five blocks: iDMA, Frame Buffer, LS Kernel, Scheduler and Tile Memory, shown in Fig. 4 (b). Contrived for image processing, iDMA, or image DMA, is an enhanced DMA engine providing two dimensional addressability. The image is loaded into frame buffer

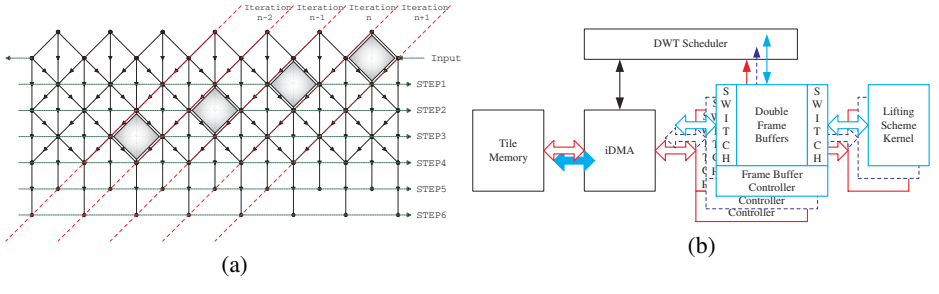

Fig. 4. (a)"Software pipelining" illustration for lifting-based Daubechies (9, 7) Filtering; (b) Proposed DWT block diagram

through iDMA and processed line by line. Above the pipelined LS Kernel (computation pipeline), Tile Memory, Frame Buffer and LS Kernel form a higher level of pipeline — task pipeline. Computation pipeline and task pipeline are well balanced. The overall performance is limited by both of them.

Scalability. Due to the symmetry and independence of line processing during one dimensional DWT decomposition at one level, the DWT architecture can be easily scaled up, referred to Fig. 4 (b). For the LS Kernel, duplication can multiple the throughput directly and almost linearly. For the peripherals, throughput can be flexibly controlled by adjusting the data bus width. Such configurability only introduces a little more complexity to the Scheduler.

3.2 Scalable Tier-1 Architecture

Overview. Various architectures of Tier-1 encoder have been proposed. Many of them [4][10] adopt the default mode (or called sequential mode when juxtaposed with parallel mode) of JPEG2000, processing on the code blocks bit-plane by bit-plane. Although default mode has the advantage on the compression rate, it is weak on error resilience and lacks intra-code-block parallelism. To overcome those drawbacks, we use a combination of parallel mode and stripe-causal mode [1] for the purpose of hardware acceleration. As shown in Fig. 5, Tier-1 coder reads the DWT coefficients from Tile Memory and writes the Tier-1 coding result to internal buffer, ready to be fetched by Tier-2 coder. The Tier-1 coder is proposed with full scalability (we use 5 BPCs, each including an MQ-coder, for better illustration). Data Collector is used to collect and organize the output of BPCs for Tier-2 coder. The details of the architecture is referred to [11].

Data Dispatcher. Data Dispatcher is the most control intensive part in the whole design. It is used to load none-all-zero bit-planes to the BPCs. It has an interval state machine working in 3 phases: *initialization*, *programming*, and *program execution*. The state machine is reset at the beginning of each precinct. In the *initialization* phase counters are initialized and starting/ending addresses of a precinct, which are pre-stored by PowerPC, and then loaded. In the *programming* phase, the state machine checks the code blocks in the order that will be used in Tier-2 coding within a precinct, and assigns the bit-planes in those code blocks to the 5 BPCs. During each *program execution*, all 5 BPCs are loaded with a column of data. A program takes 2-5 cycles to execute in different situations.

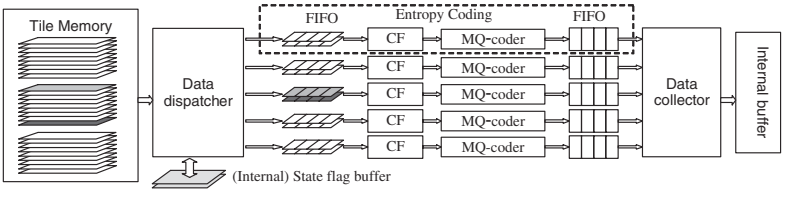

Fig. 5. Tier-1 coder architecture

Entropy Coding. The core Tier-1 coding task performs entropy coding, which includes Bit-Plane Coding and Binary Arithmetic Coding, by Context Formatter (CF) and MQ-coder individually, illustrated in Fig. 5. Each BPC acquires the data of one bit-plane from Data Dispatcher column by column. The difficulty of bit-plane coder design is that it scans data in 3 passes, where the latter two passes are dependent on the state bits updated in the earlier passes. Processing data pass by pass requires buffering and reusing the data pairs and those state flags of the whole bit-plane, which demands large internal memory space. Fortunately some research work has been conducted on this problem [10][12]. Our BPC design is inspired by these papers, so that the 3 passes can be combined into one pass, thereby saving memory cost.

4 FPGA Prototyping and Performance

The architecture has been prototyped on Xilinx Virtex-II XC2VP7-7 FPGA. As mentioned before, it can be parameterized with N DWT modules and M Tier-1 coders with area and performance concerns. The hardware costs are detailed in Table 2. The post-Place&Route shows that the entire system can operate at 130 MHz. XC2VP7 contains 4,928 slices, 44 multipliers and 44 BRAMs, thus it can roughly support up to 2 DWT modules with 6 Tier-1 modules.

Table 2. The main prototyping costs (N DWT and M Tier-1 modules)

Building blocks	LS Kernel	Frame Buffer	iDMA	Scheduler	Data Dispatcher	BPC+ MQ-Coder	Data Collector	Double buffer
Slice	$167 \times N$	$165 \times N$	$312+0.2 \times N$	342	370	$486 \times M$	244	-
Memory and other costs	6 Multipliers	8 KB $\times N$	-	-	-	2.8 KB $\times M$	-	1.2 KB $\times M$

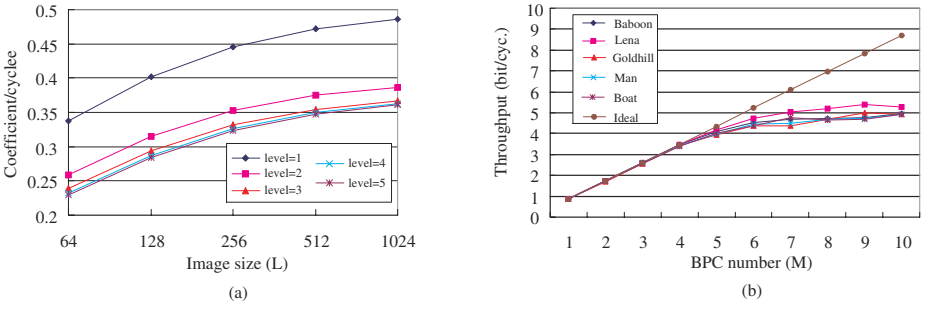

Fig. 6. (a) Throughput of DWT accelerator (N=1; test images are square with size L by L; scalability over N is almost linear); (b) Performance scalability of Tier-1 Coder

Fig. 6 (a) demonstrates the experimental evaluation of our DWT architecture in terms of throughput versus decomposition level and image size. As the decomposition level increases, the throughput decreases due to more computational budgets. The

throughput is also affected by image size because of the varying relative cost between overhead and computation. For a given circumstance, we can easily determine the number of DWT modules N based on this figure. The measured PSNR is in the range of 48.6-56.52 dB.

The throughput scalability of the proposed Tier-1 architecture is conducted in the sense of BPC numbers M, shown in Fig. 6 (b). The curves go off the ideal course because the Data Dispatcher (section 3.2) is not able to feed that many BPCs. However, such scalability is good enough since 6 BPCs or less satisfy most real-time coding applications. Furthermore, the linearity can be extended by extracting inter-code-block parallelism and Data Dispatcher duplication.

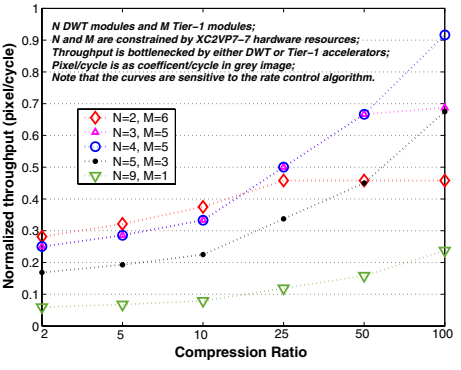

Fig. 7. Throughput exploration by N and M tradeoff (grey image, $level = 5$, $L = 64$)

One of the particular characteristics of FPGA is the in-field reconfigurability. While the number of DWT and Tier-1 modules, (N, M), is constrained by hardware resources, the field-programmability of FPGAs allows us to trade off N versus M within the hardware constraint. Fig. 7 illustrates the throughput exploration of the overall JPEG 2000 encoder by adjusting different (N, M) pair under various compression ratios. Note that the curves are also sensitive to the rate control algorithm and the image itself. In this example, the selection of (N, M) versus the compression ratio CR for optimal throughput are: (2, 6) when $CR \leq 10$; (3, 5), rather than (4, 5) for more efficient power and area despite the same throughput, when $25 \leq CR \leq 50$; and (4, 5) when CR is around 100. In case the throughput constraint is also known, the (N, M) pair which satisfies the real-time processing demand and costs least hardware resource is the optimal tradeoff. For instance, when the throughput constraint is 0.3 pixel/cycle and $CR \geq 25$ in Fig. 7, $(N, M) = (5, 3)$ is the best choice due to its area efficiency.

A comparison of JPEG 2000 encoders on similar FPGA platforms is given in Table 3. It clearly shows the superiority of our proposal with respect to commercially available IP solutions [2][3]. Actually, the routing costs over half of the critical path delay in our implementation, and the un-optimized FPGA logic also constrains the potential throughput. Generally speaking, the custom IC fabrication can further improve the throughput about two times.

Table 3. Comparison of JPEG 2000 encoders

Architecture	Family	Device	Slices	CLK (MHz)	P(Msample/s)	Image size
Amphion CS6510X2[3]	Virtex-II	-	14,034	40.4	16	1024×1024
CAST JPEG2K_E[2]	Virtex-II Pro	2VP30-7	12,885	107	34.7	256×256
Our implementation[1]	Virtex-II Pro	2VP7-7	4,671	130	60.1	1024×1024

[1] $N = 1$, $M = 5$; 512 by 512 image; 5 level decomposition; 2VP7-7 has the same speed grad of 2VP30-7;

5 Conclusion

A system level JPEG2000 Part I encoder architecture has been proposed in this paper. It is prototyped on Xilinx technology, showing high performance compared with other designs on similar platforms. The hardware accelerators are carefully designed for scalability concern with adaptive performance.

References

1. JPEG2000 image coding system, ISO/IEC International Standard 15444-1. ITU Recommendation T.800 (2000)
2. JPEG2000 Encoder Core. Cast Inc. (2002)
3. CS6510 JPEG2000 Encoder. Amphion (http://www.amphion.com/cs6510.html)
4. Andra, K., Chakrabarti, C., Acharya, T.: A high-performance JPEG2000 architecture. IEEE CSVT **13** (2003) 209–218
5. Daubechies, I., Sweldens, W.: Factoring wavelet transforms into lifting schemes. Journal of Fourier Anal. Appl. **41** (1998) 247–269
6. Ravasi, M., Tenze, L., Mattavelli, M.: A scalable and programmable architecture for 2-D DWT decoding. IEEE Trans. on Video Tech. **12** (2002) 671–677
7. Andra, K., et al.: A VLSI architecture for lifting-based forward and inverse wavelet transform. IEEE Transactions on Signal Processing **50** (2002) 966–977
8. Chen, C.Y., et al.: A programmable parallel VLSI architecture for 2-D discrete wavelet transform. Journal of VLSI Signal Processing **28** (2001) 151–163
9. Zhang, C., et al.: 'software-pipelined' 2-D discrete wavelet transform with VLSI hierarchical implementation. In: Proc. of RISSP '03. (2003) 148–153
10. Chen, K., Lian, C., Chen, H., Chen, L.: Analysis and architecture design of ebcot for jpeg-2000. In: IEEE ISCAS. (2001) 765–768
11. Long, Y., Zhang, C., Kurdahi, F.: A high-performance parallel mode EBCOT architecture design for JPEG2000. In: Proc. IEEE SOCC. (2004) 213–216
12. Chiang, J., Lin, Y., Hsieh, C.: Efficient pass-parallel architecture for EBCOT in JPEG2000. In: IEEE ISCAS. (2002) 773–776