Implementation of multi-standard video decoding algorithms on a coarse-grained reconfigurable multimedia processor

Leibo Liu, Yingjie Chen, Shouyi Yin, Dong Wang*,
Xing Wang and Shaojun Wei
Institute of Microelectronics
Tsinghua University
Beijing, China
*Email: Doonny1212@gmail.com

Li Zhou
Institute of Microelectronics
Chinese Academy of Sciences
Beijing, China

Hao Lei
School of Electronic and Information Engineering
Xi'an Jiaotong University
Xi'an, China

Peng Cao
National ASIC System Engineering Research Center
Southeast University
Nanjing, China

Abstract—**This paper proposed a THPHP (Task-based Hybrid Parallels and Hybrid Pipelines) scheme to implement multi-standard video decoding algorithms, i.e. MPEG-2, H.264 and AVS (Audio Video coding Standard), on a heterogeneous coarse-grained reconfigurable multimedia processor called REMUS (REconfigurable MUltimedia System). Multiple level parallelism and multiple level pipeline techniques are proposed in this scheme. Simulation results show that the video decoder can support H.264 HP (High Profile) 1920x1080@30fps (frame per second) streams, AVS JP (Jizhun Profile) 1920x1080@39fps streams, and MPEG-2 MP (Main Profile) 1920x1080@41fps streams when exploiting a 200MHz working frequency.**

I. INTRODUCTION

With the multimedia market booming and fast upgrade of video coding standards, there is a potential market demand for the decoder supporting multiple video standards (e.g. MPEG-2[1], H.264[2], AVS[3]), which not only satisfies the requirements of current standards, but also the requirements of various unknown standards in the future. The ASIC (Application Specific Integrated Circuit) solution has very high energy efficiency (i.e. high performance and low power consumption) however with inferior flexibility. While the GPP (General Purpose Processor) solution and the programmable DSP solution have very high flexibility however suffered from relatively poor energy efficiency. It is very difficult to catch up with the rapid evolution of multimedia market by relying on the ASIC, GPP and programmable DSP solutions, or even the combination of them. Currently, reconfigurable system is becoming an attractive topic to fulfill the rigid requirements of the various evolving video standards.

So far, a lot of reconfigurable multimedia systems were proposed, and a lot of mapping and implementation approaches were raised too. [4] proposed a configurable architecture to explore the hardware and software co-design technique for H.264 decoder with task-based MB (Macro-Block) level pipeline and MB-based parallel technique. [4] implemented the reconfiguration by easily adding or deleting accelerator modules, which was more flexible than ASIC, and also more efficient than processor-based solution. However, its reconfiguration was still inferior. [5] mapped H.264 decoder onto ADRES [6], which was a flexible coarse-grained reconfigurable architecture. The overhead of the pipeline operations heavily hampered the performance of some mapped kernels in [5]. [7] implemented H.264 decoding algorithm onto XPP-III [8][9]. [7] could decode H.264 streams with high performance, i.e. 1920x1080@24fps streams at 450MHz working frequency. However, there is no detailed description about the task level scheduling and mapping methods for multi-standard video decoder based on reconfigurable architectures in the above literatures, which has a large impact on the performance of the video decoder.

In this paper, a scheme called THPHP is presented to implement multi-standard (MPEG-2, H.264 and AVS) video decoding on a heterogeneous coarse-grained reconfigurable multimedia processor REMUS [10]. In order to reduce synchronization overhead and improve the decoding performance, multiple level parallelism and multiple level pipeline techniques are proposed.

This work was supported by National 863 Program of China Grant No. 2009AA011702.

978-1-4673-5762-3/13/$31.00 ©2013 IEEE

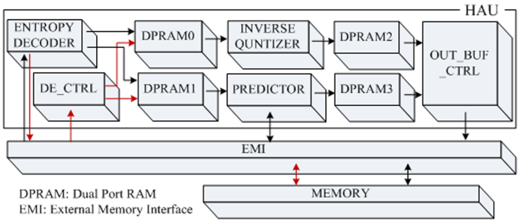

Figure 1. The architecture of HAU.

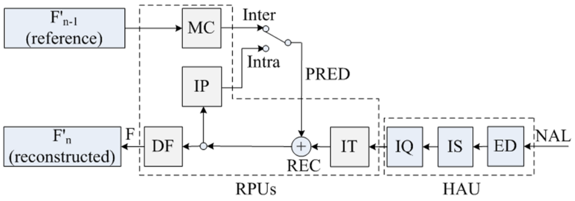

Figure 2. Block diagram of H.264 decoding process.

II. Review of the REMUS architecture

The architecture of REMUS was described in [10], which consists of a host processor (ARM11), two PEAs (Processing Element Arrays), an EnD (Entropy Decoder), and some assistant modules. Here, PEAs are called RPUs (Reconfigurable Processing Units), and EnD is called HAU (Hardware Acceleration Unit). The host processor is used for controlling application and scheduling reconfigurable hardware tasks. The RPU is a powerful dynamic reconfigurable processor, consisting of four basic units here called RCAs (Reconfigurable Computing Arrays). An RCA has three parts: DBI, FCI, and PE8x8, as described in [10]. The four RCAs can run independently to accelerate computing performance. HAU is a heterogeneous configurable module, which is mainly employed for control intensive tasks.

A. The architecture of HAU

Fig. 1 shows the architecture of HAU, including ENTROPY DECODER, INVERSE QUANTIZER, PREDICTOR, four DPRAMs with Ping-Pong mode, etc. ENTROPY DECODER can realize corresponding ED (Entropy Decoding) in H.264, AVS or MPEG-2. INVERSE QUANTIZER realizes IS (Inverse Scanning) and IQ (Inverse Quantization) in the three video standards. And PREDICTOR implements MVP (Motion Vector Prediction), IPMC (Intra Prediction Mode Calculation) and BSC (Boundary Strength Calculation).

III. The HW/SW partitioning strategies

The block diagram of H.264 decoding process is illustrated in Fig. 2, which mainly includes the following tasks: ED, IS, IQ, IT (Inverse Transform), MC (Motion Compensation) for inter MBs, IP (Intra Prediction) for intra MBs, REC (REConstruction), and DF (Deblocking Filter). MC, DF, IQ, IT and IP are computation-intensive tasks and account for approximately 77% of the execution time in H.264 HD decoding [11], which are partitioned to be performed by the two RPUs. The only exception is IQ. Since the required data-width of IQ exceeds the granularity of RPU, it is implemented by HAU to avoid the data overflow in RPU. While ED and IS are control-intensive tasks with a lot of judgment branches and irregular calculations and implemented by HAU. And the above MB level information, i.e. SPS (Sequence Parameter Set), PPS (Picture Parameter Set) and slice headers, is parsed by the host processor because their symbol rates are very low [12].

There are also the above tasks in the AVS video decoding. While the difference between MPEG-2 and H.264 is that there are no IP and DF tasks in MPEG-2. Hence, The HW/SW partitioning strategies for AVS and MPEG-2 are similar to that of H.264.

IV. The task level scheduling and mapping methods for H.264, AVS and MPEG-2 decoding algorithms

A. Task level scheduling on HAU

In the view of video decoding algorithms, MVP/IPMC plus BSC are independent of IS plus IQ. Hence, MB-based parallel technique can be adopted between INVERSE QUANTIZER and PREDICTOR. Slice level pipeline technique is used between ENTROPY DECODER and INVERSE QUANTIZER/PREDICTOR to reduce synchronization overhead and improve the decoding performance in H.264 or AVS video decoding process, while MB level pipeline technique in MPEG-2 video decoding process.

B. Task level scheduling and algorithm mapping on RPUs

To describe the mapping methods clearly, Y stands for the 16x16 luminance component, Y1-Y4 for luminance 8x8 blocks, U for the 8x8 chrominance component U, U1-U4 for the chrominance U 4x4 blocks, V for the 8x8 chrominance component V, and V1-V4 for the chrominance V 4x4 blocks. Fig. 3, Fig. 4 and Fig. 5 are the task level scheduling and mapping methods on RPUs for H.264, AVS and MPEG-2 decoding respectively. DF_H represents horizontal filter process for vertical edges, while DF_V means vertical filter process for horizontal edges in Fig. 3. The mapping methods in H.264, AVS and MPEG-2 are similar, i.e. a task is divided into several sub-tasks, which makes the utilization of RPUs maximum and improves the performance of the decoding by block-based parallel, sub-block-based parallel, and block level pipeline and sub-block level pipeline techniques. But the task level scheduling approaches are different.

1) H.264 decoding: RPU0 and RPU1 work with MB level pipeline technique, i.e. MC/IP, IDCT and REC are implemented on RPU0, while DF is executed on RPU1.

2) AVS decoding: RPU0 and RPU1 work with MB level parallel technique, i.e. MC/IP, IDCT and REC of the luminance component are implemented on RPU0, while those of the chrominance components are carried out on RPU1.

3) MPEG-2 decoding: there are no IP and DF processes, and it can meet the requirements of real-time decoding when MC, IDCT and REC are implemented in one RPU. So, RPU0 and RPU1 process different MBs, i.e. they work in parallel synchronous mode.

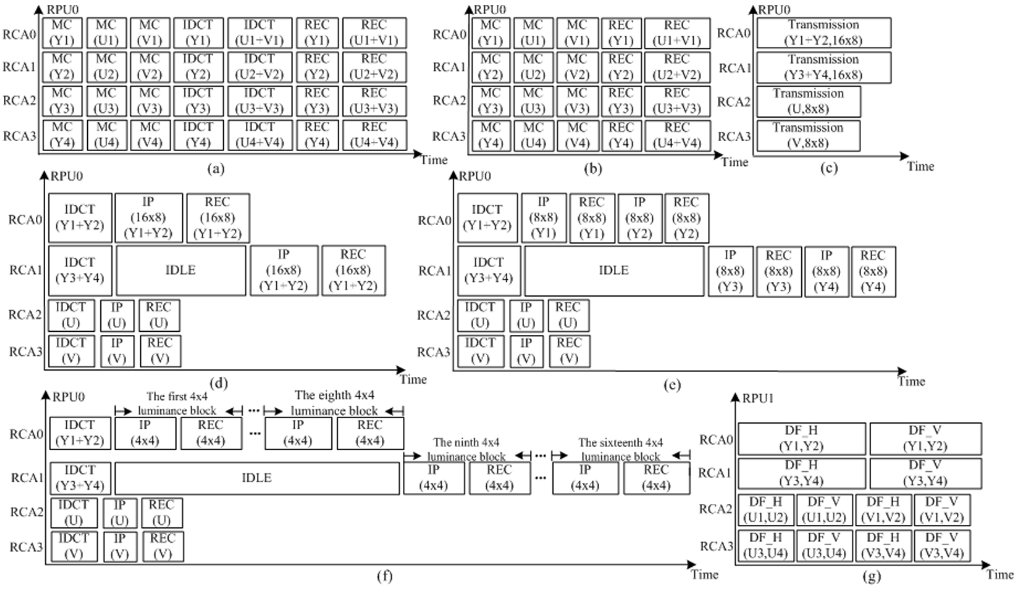
RPU0
RCA0 MC (Y1) MC (U1) MC (V1) IDCT (Y1) IDCT (U1+V1) REC (Y1) REC (U1+V1)
RCA1 MC (Y2) MC (U2) MC (V2) IDCT (Y2) IDCT (U2+V2) REC (Y2) REC (U2+V2)
RCA2 MC (Y3) MC (U3) MC (V3) IDCT (Y3) IDCT (U3+V3) REC (Y3) REC (U3+V3)
RCA3 MC (Y4) MC (U4) MC (V4) IDCT (Y4) IDCT (U4+V4) REC (Y4) REC (U4+V4)
Time
(a)
RPU0
RCA0 MC (Y1) MC (U1) MC (V1) REC (Y1) REC (U1+V1)
RCA1 MC (Y2) MC (U2) MC (V2) REC (Y2) REC (U2+V2)
RCA2 MC (Y3) MC (U3) MC (V3) REC (Y3) REC (U3+V3)
RCA3 MC (Y4) MC (U4) MC (V4) REC (Y4) REC (U4+V4)
Time
(b)
RPU0
RCA0 Transmission (Y1+Y2,16x8)
RCA1 Transmission (Y3+Y4,16x8)
RCA2 Transmission (U,8x8)
RCA3 Transmission (V,8x8)
Time
(c)
RPU0
RCA0 IDCT (Y1+Y2) IP (16x8) (Y1+Y2) REC (16x8) (Y1+Y2)
RCA1 IDCT (Y3+Y4) IDLE IP (16x8) (Y1+Y2) REC (16x8) (Y1+Y2)
RCA2 IDCT (U) IP (U) REC (U)
RCA3 IDCT (V) IP (V) REC (V)
Time
(d)
RPU0
RCA0 IDCT (Y1+Y2) IP (8x8) (Y1) REC (8x8) (Y1) IP (8x8) (Y2) REC (8x8) (Y2)
RCA1 IDCT (Y3+Y4) IDLE IP (8x8) (Y3) REC (8x8) (Y3) IP (8x8) (Y4) REC (8x8) (Y4)
RCA2 IDCT (U) IP (U) REC (U)
RCA3 IDCT (V) IP (V) REC (V)
Time
(e)
RPU0
The first 4x4 luminance block
The eighth 4x4 luminance block
RCA0 IDCT (Y1+Y2) IP (4x4) REC (4x4) ... IP (4x4) REC (4x4)
The ninth 4x4 luminance block
The sixteenth 4x4 luminance block
RCA1 IDCT (Y3+Y4) IDLE IP (4x4) REC (4x4) ... IP (4x4) REC (4x4)
RCA2 IDCT (U) IP (U) REC (U)
RCA3 IDCT (V) IP (V) REC (V)
Time
(f)
RPU1
RCA0 DF_H (Y1,Y2) DF_V (Y1,Y2)
RCA1 DF_H (Y3,Y4) DF_V (Y3,Y4)
RCA2 DF_H (U1,U2) DF_V (U1,U2) DF_H (V1,V2) DF_V (V1,V2)
RCA3 DF_H (U3,U4) DF_V (U3,U4) DF_H (V3,V4) DF_V (V3,V4)
Time
(g)

Figure 3. The task level scheduling and mapping method on RPUs for H.264, (a) non-skipped inter MBs, (b) skipped MBs, (c) I_PCM MBs (d) intra_16x16 MBs, (e) intra_8x8 MBs, (f) intra_4x4 MBs, and (g) DF on RPU1.

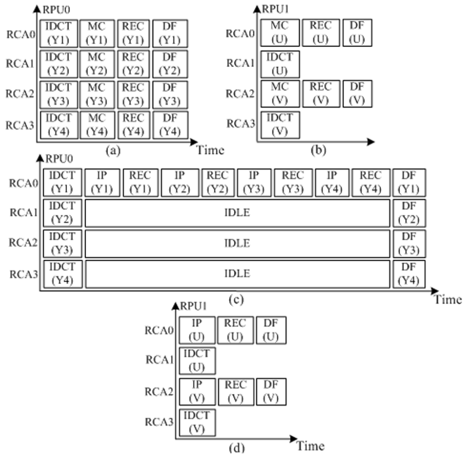
RPU0
RCA0 IDCT (Y1) MC (Y1) REC (Y1) DF (Y1)
RCA1 IDCT (Y2) MC (Y2) REC (Y2) DF (Y2)
RCA2 IDCT (Y3) MC (Y3) REC (Y3) DF (Y3)
RCA3 IDCT (Y4) MC (Y4) REC (Y4) DF (Y4)
Time
(a)
RPU1
RCA0 MC (U) REC (U) DF (U)
RCA1 IDCT (U)
RCA2 MC (V) REC (V) DF (V)
RCA3 IDCT (V)
(b)
RPU0
RCA0 IDCT (Y1) IP (Y1) REC (Y1) IP (Y2) REC (Y2) IP (Y3) REC (Y3) IP (Y4) REC (Y4) DF (Y1)
RCA1 IDCT (Y2) IDLE DF (Y2)
RCA2 IDCT (Y3) IDLE DF (Y3)
RCA3 IDCT (Y4) IDLE DF (Y4)
Time
(c)
RPU1
RCA0 IP (U) REC (U) DF (U)
RCA1 IDCT (U)
RCA2 IP (V) REC (V) DF (V)
RCA3 IDCT (V)
(d)
Time

Figure 4. The task level scheduling and mapping method on RPUs for AVS, (a) and (b) for inter MBs, (c) and (d) for intra MBs.

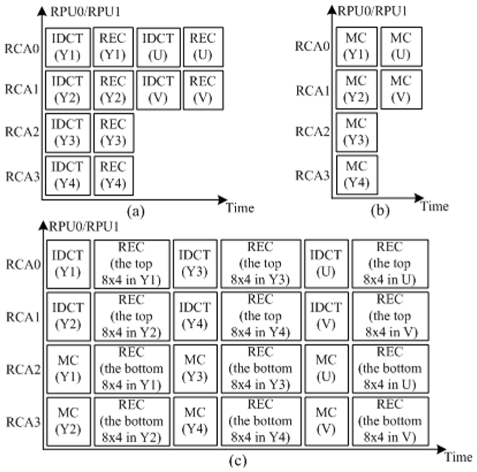
RPU0/RPU1
RCA0 IDCT (Y1) REC (Y1) IDCT (U) REC (U)
RCA1 IDCT (Y2) REC (Y2) IDCT (V) REC (V)
RCA2 IDCT (Y3) REC (Y3)
RCA3 IDCT (Y4) REC (Y4)
Time
(a)
RPU0/RPU1
RCA0 MC (Y1) MC (U)
RCA1 MC (Y2) MC (V)
RCA2 MC (Y3)
RCA3 MC (Y4)
Time
(b)
RPU0/RPU1
RCA0 IDCT (Y1) REC (the top 8x4 in Y1) IDCT (Y3) REC (the top 8x4 in Y3) IDCT (U) REC (the top 8x4 in U)
RCA1 IDCT (Y2) REC (the top 8x4 in Y2) IDCT (Y4) REC (the top 8x4 in Y4) IDCT (V) REC (the top 8x4 in V)
RCA2 MC (Y1) REC (the bottom 8x4 in Y1) MC (Y3) REC (the bottom 8x4 in Y3) MC (U) REC (the bottom 8x4 in U)
RCA3 MC (Y2) REC (the bottom 8x4 in Y2) MC (Y4) REC (the bottom 8x4 in Y4) MC (V) REC (the bottom 8x4 in V)
Time
(c)

Figure 5. The task level scheduling and mapping method on RPUs for MPEG-2, (a) intra MBs, (b) skipped MBs and (c) non-skipped inter MBs.

C. The System Scheduling Approach

The main process units for implementing tasks in REMUS include HAU, RPU0 and RPU1.

1) H.264 decoding: MB level pipeline technique is used among HAU, RPU0 and RPU1.

2) AVS decoding: MB level pipeline is used among HAU, and RPU0/RPU1. Also, MB-based parallel technique is used between RPU0 and RPU1.

3) MPEG-2 decoding: MB level pipeline technique is used among HAU and RPU0/RPU1; MB-based parallel technique is adopted between RPU0 and RPU1.

According to the above description, the performance of the multi-standard (i.e. H.264, AVS and MPEG-2) video decoding can be raised by using the proposed scheme which properly adopts some techniques, such as slice level pipeline, MB level pipeline, block level pipeline, sub-block level pipeline, MB-based parallel, block-based parallel and sub-block-based parallel techniques. All the techniques are based on tasks in video decoding algorithms, which is the reason that the proposed scheme is called THPHP.

V. IMPLEMENTATION AND COMPARISON

Fig. 6 shows the chip photograph of RHINOCEROS with TSMC 65nm technology, which is a reconfigurable high performance chip and can be applied to the set-top box, communication base station, etc. RHINOCEROS is mainly comprised of ARM11, PLL, REMUS, etc. REMUS supports H.264, AVS and MPEG-2 video decoding algorithms.

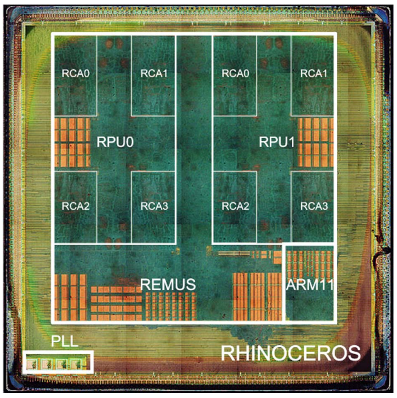

Figure 6. Chip photograph of RHINOCEROS.

TABLE I. SUB-ALGORITHMS IN H.264, AVS, MPEG-2 ON RPUS.

Sub-algorithms		H.264		AVS	MPEG-2
IDCT	Sub-Macro-block Size	4x4	8x8	8x8	8x8
	Cycles/MB	106	192	96	239
MC	Sub-Macro-block Size	4x4	8x8	8x8	8x8
	Cycles/MB	1059	323	204	191
DF	Cycles/MB	704		176	-

TABLE II. COMPARISON BETWEEN [7] AND REMUS.

Item	[7]	The proposed	Speed up
Platform	XPP-III	REMUS	N.A.
Technology	90nm	65nm	
Area (mm^2)	75	48.9	
clock	450MHz	200MHz	
Performance	1920x1080@24fps	1920x1080@30fps	181%
MBs/s/MHz	435	1224	

The decoding performances of H.264, AVS and MPEG-2 target at 1920x1080@30fps when exploiting a 200MHz working frequency, i.e. the number of clock cycles to process an MB should be limited to 816 cycles. The MB-level pipeline techniques are used in decoding process, so for HAU, RPU0 or RPU1, if the number of clock cycles to process an MB is smaller than 816, the video decoder can satisfy the real-time decoding requirement.

A. Performances of HAU and critical sub-algorithms implemented on RPUs

Simulation shows that decoding process of HAU consumes 512, 435 and 438 clock cycles per MB in H.264, AVS and MPEG-2 respectively. The performances of critical sub-algorithms implemented on RPUs in H.264, AVS and MPEG-2 video decoding are listed in TABLE I.

B. The performance comparison of H.264 decoding

TABLE II illustrates the comparison between [7] (the XPP-based reconfigurable system) and REMUS. The unit MBs/s/MHz represents the video decoding capability, meaning how many MBs can be processed per second at the same clock frequency. Compared with [7], H.264 decoding performance is improved by 181% as listed in TABLE II.

VI. CONCLUSION

This paper presents a scheme called THPHP to implement multi-standard (i.e. H.264, AVS and MPEG-2) video decoding on a coarse-grained dynamically reconfigurable multimedia processor called REMUS. In order to improve the decoding performance, several pipeline techniques (i.e. slice level pipeline, MB level pipeline, block level pipeline and sub-block level pipeline techniques) are introduced. Also, several parallel techniques are applied to computation-intensive tasks to improve the decoding performance, including MB-based parallel, block-based parallel and sub-based parallel techniques. Simulation shows that the proposed scheme supports H.264 HP 1920x1080@30fps streams, AVS JP 1920x1080@39fps streams, and MPEG-2 MP 1920x1080@41fps streams in 4:2:0 format when exploiting an operating frequency of 200MHz.

REFERENCES

[1] ISO/IEC 13818, “Generic coding of moving pictures and associated audio (MPEG-2)”, 1994.

[2] T. Wiegand, G. J. Sullivan, G. BjØntegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” Circuits and Systems for Video Technology, IEEE Transactions on, vol. 13, pp.560-576, 2003.

[3] AVS Workgroup, “Information technology - advanced coding of audio and video - part2: Video,” GB/T 200090.2-2006.

[4] G. A. Jian, J. C. Chu, T. Y. Huang, T. C. Chang, and J. I. Guo., “A system architecture exploration on the configurable HW/SW co-design for H.264 video decoder,” Proceedings - 2009 IEEE International Symposium on Circuits and Systems, ISCAS 2009, p 2237-2240.

[5] B. F. Mei, F. J. Veredas, and B. Masschelein, “Mapping an H.264/AVC decoder onto the ADRES reconfigurable architecture,” Proceedings - 2005 International Conference on Field Programmable Logic and Applications (IEEE Cat. No.05EX1155), p622-625, 2005

[6] B. F. Mei, S. Vernalde, D. Verkest, H. D. Man, and R. Lauwereins, “ADRES : an architecture with tightly coupled VLIW processor and coarse-grained reconfigurable matrix,” Field-Programmable Logic and Applications. 13th International Conference, FPL 2003. Proceedings (Lecture Notes in Comput. Sci. Vol.2778), p61-70, 2003.

[7] M. K. A. Ganesan, S. Singh, and F. May, “H. 264 decoder at HD resolution on a coarse grain dynamically reconfigurable architecture,” Proceedings - 2007 International Conference on Field Programmable Logic and Applications, FPL, p467-471, 2007.

[8] White Paper of Video Decoding on XPP-III, PACT, 2006.

[9] D. Rossi, F. Campi, S. Spolzino, S. Pucillo, and R. Guerrieri, “A heterogeneous digital signal processor for dynamically reconfigurable computing,” IEEE Journal of Solid-State Circuits, v 45, n 8, p 1615-1626, Aug. 2010.

[10] M. Zhu, L. Liu, S. Yin, C. Yin, and S. Wei, “A cycle-accurate simulator for a reconfigurable multi-media system,” IEICE Transactions on Information and Systems, v E93-D, n 12, p 3202-3210, December 2010.

[11] H. Baik, K. H. Sihn, Y. Kim, S. Bae, N. Han, and H. J. Song, “Analysis and parallelization of H.264 decoder on cell broadband engine architecture,” ISSPIT 2007 - 2007 IEEE International Symposium on Signal Processing and Information Technology, p791-795, 2007.

[12] T. W. Chen, Y. W. Huang, T. C. Chen, Y. H. Chen, C. Y. Tsai, and L. G. Chen, “Architecture design of H.264/AVC decoder with hybrid task pipelining for high definition videos,” IEEE International Symposium on Circuits and Systems (ISCAS) (IEEE Cat. No. 05CH37618), p2931-2934 Vol. 3, 2005.