

FPGA Dynamic Reconfiguration using the RVC Technology: Inverse Quantization Case Study

Manel Hentati, Yassine Aoudni, Jean François Nezan, Mohamed Abid, Olivier Déforges

▶ **To cite this version:**

Manel Hentati, Yassine Aoudni, Jean François Nezan, Mohamed Abid, Olivier Déforges. FPGA Dynamic Reconfiguration using the RVC Technology: Inverse Quantization Case Study. Conference on Design and Architectures for Signal and Image Processing (DASIP), Nov 2011, Tampere, Finland. pp.CD. hal-00661337

HAL Id: hal-00661337
https://hal.science/hal-00661337

Submitted on 19 Jan 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FPGA DYNAMIC RECONFIGURATION USING THE RVC TECHNOLOGY: INVERSE QUANTIZATION CASE STUDY

Manel Hentati[1,2] *, Yassine Aoudni* [2] *, Jean-François Nezan*[1]*, Mohamed Abid* [2] *and Olivier Deforges*[1]

[1] INSA/IETR, 20, av des Buttes de coesmes CS 14315 F-35043 RENNES, France
[2] ENIS/CES, Route Soukra B.P. 1173, 3038 Sfax, Tunisie

ABSTRACT

With the rapid evolution of technology, the latest FPGA architectures such as Virtex series of Xilinx introduced a new feature called Dynamic Partial Reconfiguration (DPR). This technique allows designer to configure a portion of the FPGA while other parts continue to run on the same FPGA. The design of an embedded system based on the DPR functionality is still complex and tedious. The MPEG consortium proposes the Reconfigurable Video Coding (RVC) technology. RVC provides a high level description of video decoders described as a set of interconnected Functional Units . This paper studies the use of the RVC technology for the specification of an application and the design of a system based on the DPR functionality. In this paper, we study the Inverse Quantization (IQ) algorithm of an MPEG-4 decoder and how to switch between the MPEG-2 and the H263 IQ algorithms using RVC and DPR. This simple and concrete case study highlights the DPR restrictions to take into account in MPEG RVC description in order to use the DPR.

Index Terms— MPEG-RVC, Dynamic Partial Reconfiguration, Inverse Quantization, RVC-CAL language, FPGA, RVC framwork.

I. INTRODUCTION

Multimedia processing becomes more and more important with wide variety of applications. So the video standards Such as ITU-T H.261, H.263, ISO/IEC MPEG-1, MPEG-2 and MPEG-4 can't satisfy the need of the embedded systems'designers. Indeed, these standards suffer from lack of re-usability and generality of the code provided by the standard (usually a C/C++ monolithic specification). Moving Picture Expert Group (MPEG) proposes a new standard which is RVC [1]. This standard aims at providing a unified high-level specification of current and future MPEG video coding technologies by using dataflow models. This standard offers the means to overcome the lack of interpretability between the many video codecs deployed in the market. RVC is based on a dataflow-oriented language called RVC-CAL [12] which is a subset of the original CAL "Caltrop Actor Language". This language is a textual and domain specific language for writing dataflow models. RVC-CAL is supported by several tools including OpenDF [20] for the simulation, CAL2HDL [17][18] and ORCC [19] for the automatic code generation (C, HDL, LLVM ...).

In order to implement an application from a dataflow description, we use FPGA technology. This later offers a great balance between performance, parallelism, and cost. Besides, the last generation of FPGA from Xilinx provides DPR. This technique has prominent advantages such as the ability to change hardware,less reconfiguration time and hardware sharing. DPR of FPGA seem to be a promising approach in the future, that is why we study this technology.

In this paper, we investigate the use of DPR in the RVC context. As far as we know, there are not many papers published concerning the use of this technique in the RVC technology.

In our work, DPR is applied on the IQ of a MPEG-4 decoder. In fact, this decoder provided by the RVC group only supports H263 IQ. We first add the MPEG-2 IQ algorithm in RVC-CAL and we use the CAL2HDL code generator to transform IQ algorithm from RVC-CAL to hardware description. After that we implement these two modules in reconfigurable platform from Xilinx using DRP.

In the remainder of this paper, Section 2 presents the context, Section 3 describes these two inverse quantization modules, Section 4 exposes the implementation of our application in FPGA using DPR in the RVC framework and the experimental results, the Section 5 presents the discussion of our work. Section 6 concludes the paper.

II. CONTEXT

This section presents a study on the MPEG-RVC and DPR technique.

II-A. MPEG RVC

MPEG RVC aims to provide a framework to define a multitude different codecs by combining together blocks, named functional unit (FU) from a standard video tool library (VTL) [13]. This standard has been standardized by MPEG as the part of MPEG-B and MPEG-C standards. But it is still evolving. The MPEG-C standard[10] presents the library of video coding tools employed in existing MPEG standards. And the MPEG-B standard [14] presents

the overall framework and the standard languages used to describe the different components of the framework.

In the RVC framework, a decoder is formed by a set of FUs and a decoder description. The Figure 1 illustrates the structure of a RVC decoder

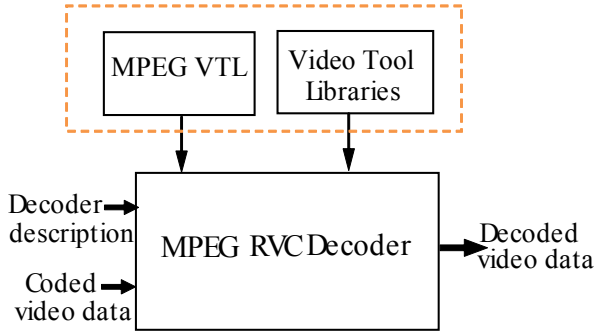


Fig. 1. Structure of MPEG_RVC decoder

FUs define a processing entity of a decoder. It may be from MPEG VTL or/and other video tools libraries. It is described in RVC-CAL language. A FU contains some processing units, control units and input and output ports

in [12] authors define a reconfigurable video with two types of data. The first one is the bit stream syntax description (BSD), which describes the structure of the bitstream. The BSD is written in RVC-BSDL. The second one is the FU network description (FND), which describes the connections between the FUs. The FND is written in FNL. The RVC framework is supported by several tools to secure efficient development, reconfiguration, and implementation processes. These tools are capable to directly synthesize the RVC decoder into both Hardware Description Languages (HDL) and/or software description(e.g., C, C++).

Several works discussed the implementation in RVC framework using reconfigurable platform.

In [16] the authors present a method to quickly prototype and generate hardware implementation of a baseline part of the LAR coder, from an RVC-CAL description.

Co-design approach is used in paper [5] to implement MPEG-4 decoder SP that is described in RVC-CAL.

In this paper [15] the authors propose a model of implementation a motion estimation module in the RVC context.

In our work, we propose an optimal implementation of H263 and MPEG-2 IQ algorithms using DPR.

II-B. FPGA Partial Reconfiguration

The last generation of FPGA from Xilinx adds a new technique called Dynamic Partial Reconfiguration (DPR). DPR provides the modification of a portion of the device while the rest remains unchanged and active. The Figure 2 shows a reconfigurable FPGA structure.

This technique can provide some benefits.

- Increase in functionalities of a single FPGA.

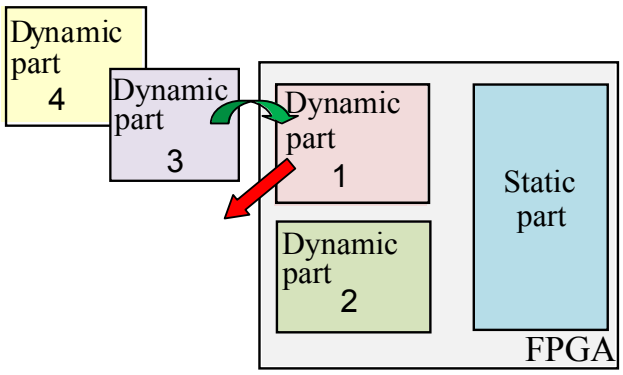


Fig. 2. A partially-reconfigurable FPGA

- Improve FPGA area efficiency
- Augment architectures flexibility [3]
- Reduce power consumption [4]

However, it has two drawbacks. The first one is the reconfiguration time(latency) and the second one is the lack of design tools and documentations. The most used partially reconfigurable FPGAs are the Xilinx Virtex series devices such as Virtex-II, Virtex-II Pro, Virtex-4, Virtex-5 and Virtex-6. A reconfigurable system typically comprises an area for static system components and one or more partial reconfigurable region for dynamic reconfigurable modules.

The dynamic partial reconfiguration is used in the different domains such as security, networks, and image processing.

In [6] authors presents hardware/software implementation of AES (Advanced Encryption Standard) cryptography algorithm using partial dynamic reconfiguration technique. The proposed architecture is based on a Microblaze processor that manages the reconfigurable module. It allows to modify or/and change the size of the key without stopping the normal operation of the system. The experimental results show performance of the AES algorithm in terms of security and safety.

The paper [7] proposes a methodology and modular architecture to implement situation-based reconfiguration in Wireless Network using partial reconfiguration technique. This main advantage of this work appears in the reduction of power consumption and saving cost.

In the reference [8], the authors present the implementation for a scalable H.264/AVC deblocking filter using partial reconfiguration System. The adopted architecture is based on Xilinx Virtex-4 ML410 FPGA board. The results show performance in terms of maximum frequency and throughput.

We notice that DPR is more and more used, so this technique is a promise solution to design a preferment embedded system. But, there is not much research on how efficiently use DPR to implement an algorithm which is described in a high-level specification language such as RVC-CAL.

III. INVERSE QUANTIZATION

Inverse quantization is an important step in the video decoding process. Essentially, the inverse quantization algorithm reproduces the DCT coefficients computed by the encoder. There are two inverse quantization methods specified for MPEG4 video. The first one uses matrix called MPEG-2 inverse quantization and the second one is based on scalar coefficient named H263 inverse quantization.

MPEG-2 inverse quantization method [9] (shown in Figure 3) introduces matrix as a weighting factor in the process. In fact, different matrix for intra and inter blocks are used. The purpose of this method is to exploit properties of the human visual system. When human eyes are less sensitive to low frequencies, which can be quantized with a coarser step-size compared to important frequency. The result of quantization method gives a coded bit-stream more compact with minimal distortion to the picture.

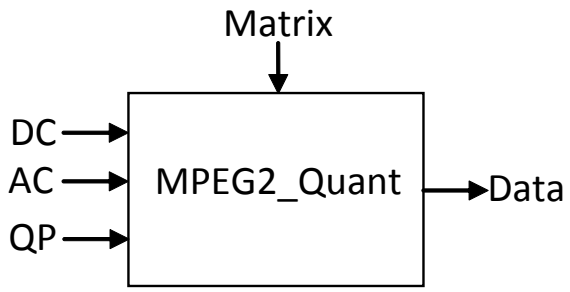


Fig. 3. Inverse quantization MPEG-2 diagram

H263 inverse quantization method [9] is based on the calculation of scalar. Figure 4 shows the H263 inverse quantization diagram. The scalar is delivered by the standard H263. This method is easy and simple.

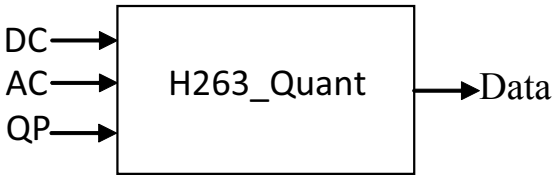


Fig. 4. Inverse quantization H263 diagram

We note that these methods of inverse quantization have the same structure except the first one that has an input "Matrix" in extra.

In our work, we adopted the default quantization matrix defined in [2].

IV. IMPLEMENTATION & RESULTS

In this section, we present a design method for implementation of two IQ algorithms using DPR.

To implement two reconfigurable modules on the same place, they must have the same inputs / outputs. According to the study on these IQ algorithms, MPEG-2 IQ and H263 IQ haven't the same number of inputs. Therefore, we must add an interface and encapsulate the IQ algorithm in this interface. the number of interface inputs should be equal to the maximum of MPEG-2 IQ inputs number and H263 IQ inputs number. The general structure of the inverse quantization is shown in Figure 5.

8	17	18	21	21	23	25	27
17	18	19	21	23	25	27	28
20	21	22	23	24	26	28	30
21	22	23	24	26	28	30	32
22	23	24	26	28	30	32	35
23	24	26	28	30	32	35	38
25	26	28	30	32	35	38	41
27	28	30	32	35	38	41	45

16	17	18	19	20	21	22	23
17	18	19	20	21	22	23	24
18	19	20	21	22	23	24	25
19	20	21	22	23	24	26	27
20	21	22	23	25	26	27	28
21	22	23	24	26	27	28	30
22	23	24	26	27	28	30	31
23	24	25	27	28	30	31	33

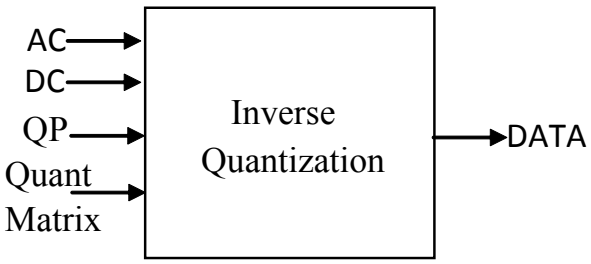


Fig. 5. The general structure of the inverse quantization

In our design method, we first add the MPEG-2 IQ algorithm to MPEG-4 decoder. Than we modify the H263 IQ algorithm. Finally we implement these two modules using DPR.

In order to add the inverse quantization MPEG2 to MPEG-4 decoder we follow three steps:

1) The description of MPEG-2 IQ as a dataflow program with the RVC-CAL language.
2) The generation the code in C language with an open source tool named ORCC "Open RVC-CAL Compiler".
3) The test C code with Microsoft visual C++ using a test video sequence.

After testing the decoder and verifying that is running correctly for both inverse quantization methods, we can automatically transform the code in hardware description using CAL2HDL. The code generated is formed by VERILOG files that present the actors and a VHDL file for top. The top file defines the highest hierarchical representation of the design connections. The connection between the FUs is insured by synchronous or asynchronous FIFO buffers.

IV-A. Target platform

Our application has been tested and implemented in the architecture which is defined in [21]. Figure 6 illustrates our target architecture

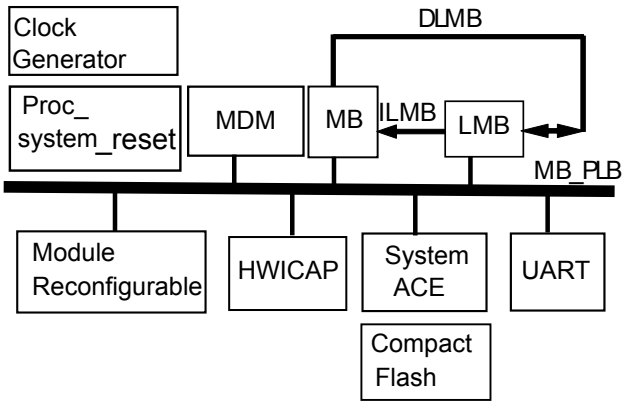


Fig. 6. Target architecture

The main parts of this architecture are:

- The processor Microblaze (MB) manages the implementation of reconfigurable modules.
- The Flash memory stores bitstreams partially reconfigurable.
- HWICAP (Hardware Internal Configuration Access Port) loads partial bitstream from flash memory . this component is provided by Xilinx. It has 32-bits data port and operating at 100 MHz.
- The bus PLB (Processor Local Bus) transfers data between peripherals.
- UART provides serial communication between PC and FPGA.
- The LMB is a fast local bus for connecting MicroBlaze instruction and data ports to high-speed peripherals. It is the standard communication bus for Xilinx systems.
- The reconfigurable module is inverse quantization.

IV-B. The implementation of two inverse quantization modules

Applying the flow of partial reconfiguration to Xilinx FPGAs requires design tools and methodologies that exploit the partial reconfiguration capabilities. Xilinx proposes this set of tools: synthesis tool like Xilinx ISE (Integrated Synthesis Environment), PlanAhead provides friendly graphical interface for placement/routing jobs, EDK (embedded development kit) helps to build an on-chip system.

We verify the functionality of our design using the Xilinx Virtex-5 development board. This board has several peripheral devices and connectors such as FLASH, DDRAM, UART, JTAG, VGA and other peripherals which allow implementation of complex FPGA applications. The core of the board is Xilinx virtex-5 lx110tff1136 FPGA in which the Microblaze soft-core processor is implemented. The Microblaze is a 32-bit RISC embedded [22] processor. In this approach, to commute from one reconfigurable module to another, we need a unit of control to manage efficiently the partial reconfiguration. And to define the reconfigurable partition region in the FPGA, we should know the occupied area by each IQ module. To validate the efficiency of partial reconfiguration technique in the RVC technology, we compare the area occupied, maximun frequency, power consumption and design time of an architecture with DPR and anothor one without DPR . In the first time, we have implemented the inverse quantization algorithm in the virtex-5 (XC5LX110T) board and virtex-4 (XC4VFX12) board of Xilinx The synthesis results are summarized in the Table I. this results, we allow to know the value of the area required for each module reconfigurable. The IQ H263 module and QI-MPEG-2 the module correspond at inverse quantization H263 and MPEG-2 algorithms, respectively. IQ MPEG2-H263 module contains two types of inverse quantization H263 and MPEG-2. To switch between these two modules we add a loop "if" in the CAL file

Table I. On-chip area results

	FPGA Resource	XC5LX110T	XC4VFX12
IQ H263	Slices	83\69120	83\69120
	Flip_Flop	19\140	83\69120
	DSP	1\64	1\32
	LUTs	76\69120	96\69120
IQ MPEG2	Slices	122\69120	99\69120
	Flip_Flop	57\140	123\69120
	DSP	2\64	2\32
	LUTs	133\69120	123\69120
IQ MPEG2-H263	Slices	152\69120	123\69120
	Flip_Flop	87\140	143\69120
	DSP	2\64	2\32
	LUTs	182\69120	202\69120

We note that MPEG-2 IQ uses more area than H263 IQ. Because the computation of MPEG-2 IQ is more complex.

According to these results we can define the reconfigurable partition region. This region should be at least equal to the greater value of the area required for each module reconfigurable.

The performance implementation of IQ modules is grouped in the Table II

Table III gives the experimental results in term of power consumption. The results are obtained via XPower which is the power estimation tool of Xilinx.

We find that the saving area on chip indicates directly the optimized power consumption.

After checking of different hardware implementation of

Table II. Performance implementation for IQ

	FPGA Resource	XC5LX110T	XC4VFX12
IQ H263	Minimum Period (ns)	9.85	7.869
	Maximum Frequency(MHZ)	110.072	127.08
IQ MPEG-2	Minimum Period (ns)	10.882	10.258
	Maximum Frequency(MHZ)	91.455	97.481
IQ MPEG-2-H263	Minimum Period (ns)	11.919	11.120
	Maximum Frequency(MHZ)	83.895	90.770

Table III. Power consumption estimation of IQ modules

	Total power (mw)
IQ MPEG-2	859
IQ H263	845
IQ MPEG-2-H263	872

IQ algorithms, we continue with the implementation of the application using DPR.

In our method design, first we have created a processor system using EDK. Than, we have added a user peripheral which included a place holder for the reconfigurable partition, and we have generated netlist files. After that, we have created an software application (code C) using SDK. This code is executed by the Microblaze processor, to switch a configuration to another. Finally, full bitstreams as well as partial reconfiguration bitstreams have been generated using the PlanAhead software. Also, we have generated an ACE file for Compact Flash memory. With ACE file we can automatically program the FPGA because this file contain the bitstreams and other information for programming FPGA.

In our experimentation, we use virtex-5 board and the HWICAP from Xilinx which has 32-bits data port and operating at 100 MHz.

Table IV shows the synthesis results of the static and reconfigurable regions, and their device utilization.

Table IV. Performance implementation for IQ

	Register	LUTs	slices	DSP48E
IQ MPEG-2	6537	6743	3769	5
IQ H263	6498	6686	3708	4

Table V presents the partial bitstream size and the configuration time. The configuration time is obtained by a hardware timer implemented in FPGA. This time is dependent on the size of the configuration file and on the way to download it into FPGA.

Table V. bitstream information

bitstreams	Size per bitstream (Bytes)	Configuration time per bitstream (us)
IQ MPEG-2	61440	157,6
IQ H263	61440	155,8

Table IV illustrates the design time for an architecture with DPR and another one without this technique. This time presents the time required to develop an application on FPGA.

Table VI. Design time

	Design time
architecture with DPR	2 mouths
architecture without DPR	2 weeks

The experimental results show the efficiency of the RDP approach in reducing the occupied area and the consumption power. However, the design time is still high. This is mainly due to the lack of a well-adapted design process methodology and tools supporting DPR. The results presented in this paper where our first experience using the DPR technology. The time spent for the proposed experiments can thus be decreased but the process still long, tedious and error-prone.

V. DISCUSSION

Based on the research presented on this paper, it is clear that the MPEG RVC standard should evolve in order to really complete the functionality defined in its requirements.

To dynamically replace any algorithm part (FU) with another one in a FPGA, you should know from the synthesis phase the Inputs /Outputs (I/O) and area required of each of the reconfigurable part .

The reconfiguration times for the inverse quantization are very short (about 157 us) compared to the time of an image decoding (typically 40ms). This shows that, the DPR may be used between two images of a video sequence to reconfigure the system. We can consider a reconfiguration depending on the type of decoded images (I for Intra, P for Predictive or B for bi-Directional)

These results were obtained with a Virtex-5 ICAP. The later operating at 100 MHZ and has 32-bits data port. But, if we use a Virtex-II ICAP, (operating at 50 MHz and has 8-bits data port), we will not achieve sufficient performance

(reconfiguration time is about 1228.2 us). So the PRD technique is not interesting in this case (reconfiguration between two images of a video sequence).

In the RVC framework, the modified FU should be a part of the VTL. Indeed, if two FUs of the VTL are defined for the same algorithm (MPEG-2 and H263 FUs for IQ is an example) these two FUs must have the same I/O to apply the DPR, even if these I/O are not used during the execution of this algorithm. Moreover, it is necessary that the area allocated for these FUs is at least equal to the greater of two area needed for each of two FUs.

Actually, the main objective of MPEG RVC standard is to describe the algorithms in the VTL as they exist in previous standards. This is against the use of DPR for existing algorithms and it must be taken into account to future algorithms in which it is difficult to know a priori the I/O. It is even more difficult to predict the surface required of algorithms without knowing the details of their programming. So a new research is needed to define a "generic codec" in which these parameters (I/O and area) are defined. These parameters should adopt the worst case. The final implementation will not be fully optimized in terms of area and latency.

A second solution is to consider all of the decoder as a single zone for the DPR. This technique is certainly less interesting in the context of the DPR (reconfiguration time is longer). But the final solution will be optimized in terms of area and latency. This solution can use the standard MPEG RVC without modification . However the generation of bitstreams must be done on the FPGA when a new RVC codec must be instantiated (Online Generation). This approach is already used in some works such as the generation of LLVM code from RVC-CAL [11]. DPR seems to be able to give good results. Therefore, in our future work, we will adopt this approach. As we notice, partial reconfiguration is one of the useful solutions to reduce on chip area an increase performance, we use this technique to design an optimal implementation of RVC application.

VI. CONCLUSION

This paper introduces DPR technique and proposes to use it in a RVC technology. Since, the RVC framework is still under development at MPEG. There is not much research on how efficiently use DPR in the RVC framework. Based on our study, DPR approach should satisfy these three constraints:

- Reconfigurable modules implemented on the same place, must have the same inputs / outputs.
- A unit of control is required to manage efficiently the partial reconfiguration. This unit control is a code written in C and executed by a Microblaze processor.
- The area of the reconfigurable partition region is at least equal to the greater of two area needed for each of two reconfigurable modules.

The proposed architecture is feasible for implementation an IQ algorithm of MPEG-4 decoder using DPR and the experimental results show that dynamic reconfiguration of FPGAs is a promising approach for saving resources and increasing performance.

As perspectives, we propose to continue the use DPR in the RVC technology by applying this technique between two different RVC decoders. These two decoders will be placed in same reconfigurable partition regions in FPGA. We plan also to improve the tools which automatically implement the decoder by integrating DPR functionality.

VII. REFERENCES

[1] Matthieu Wipliez, Ghislain Roquier, Jean-Franois Nezan Software code generation for the RVC-CAL language , Springer link ,Journal of signal processing systems June 2009

[2] ISO/IEC 14496-2: 2004, Information technology - Coding of audio-visual objects - Part 2: Visual, 2004.

[3] Cindy Kao *Benefits of Partial Reconfiguration Take advantage of even more capabilities in your FPGA*, Xcell Journal Xilinx,, vol. I, pp. 65-67, 2005.

[4] Michael G. Lorenz, Luis Mengibar, Mario G. Valderas, and Luis Entrena Power Consumption Reduction Through Dynamic Reconfiguration, Springer-Verlag Berlin Heidelberg, pp. 751-760,2004 .

[5] Nicolas Siret, Ismail Sabry, Jean Franois Nezan and Mickael Raulet *A codesign synthesis from an MPEG-4 decoder dataflow description*,IEEE international Symposium Circuit and Systems ISCAS,2010 .

[6] Zine El Abidine ALAOUI ISMAILI and Ahmed MOUSSA*Self-Partial and Dynamic Reconfiguration Implementation for AES using FPGA*, IJCSI International Journal of Computer Science Issues, Vol. 2, 2009.

[7] Rafael Garcia, Ann Gordon-Ross, and Alan D. George *Exploiting Partially Reconfigurable FPGAs for Situation-Based Reconfiguration in Wireless Sensor Networks*, FPCCM ,IEEE International Symposium Field-Programmable Custom Computing Machines, April 2009.

[8] Rakan Khraisha and Jooheung Lee *A scalable H.264/AVC deblocking filter architecture using dynamic Partial reconfiguration*, ICASSP International Conference on Acoustics, Speech, and Signal Processing , pp. 1566-1569, Mars 2010 .

[9] M. Closson, B. Blodget, J. Mason, B. Bridgford, and J. Young *MPEG-4 Natural Video Codings*,book ,April 2002

[10] MPEG Video TechnologiesPart 4: Video Tool Library, ISO/IEC FDIS 23002-4, 2009

[11] Jerome Gorin, Matthieu Wipliez Franoise Prteux , Mickael Raulet *LLVM-based and scalable MPEG-RVC decoder*,Journal of Real Time Image Processing ,2010.

[12] Mattavelli, M., Amer, I., Raulet, M *The reconfigurable*

video coding, Signal Processing Magazine, IEEE 27(3), pp159 -167, 2010.

[13] Ihab Amer, Christophe Lucarz, Ghislain Roquier,Marco Mattavelli, Mickal Raulet, Jean-Franois Nezan, and Olivier Dforges *Reconfigurable Video Coding on Multicore*, IEEE Signal Processing magazine, 2009.

[14] MPEG Systems TechnologiesPart 4: Codec Configuration Representation, ISO/IEC FDIS 23001-4, 2009

[15] Julien Dubois, Richard Thavot ,Romuald Mosqueron ,Johel Miteran and Christophe Lucarz *Motion Estimation Accelerator with User Search Strategy in an RVC Context* ,IEEE ICIP Journal of Real Time Image Processing ,2009.

[16] Khaled Jerbi, Matthieu Wipliez, Mickael Raulet, Olivier Dforges, Marie Babel and Mohamed Abid *Automatic Method For Efficient Hardware Implementation From RVC-CAL Dataflow: A LAR Coder baseline Case Study*,Journal of Convergence , 2010.

[17] R. Gu, J. W. Janneck, S. S. Bhattacharyya, M. Raulet, M. Wipliez, and W. Plishker, *Exploring the concurrency of an MPEG RVC decoder based on dataflow program analysis*,IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, no. 11, pp. 16461657, 2009.

[18] *Cal2HDL-openforge source* Available from: http://openforge.sourceforge.net. [Accessed: December 2010]

[19] J. W. Janneck, M. Mattavelli, M. Raulet, and M. Wipliez, *Reconfigurable video coding: a stream programming approach to the specification of new video coding standards*,in MMSys 10: Proceedings of the first annual ACM SIGMM conference on Multimedia systems. New York, USA: ACM, pp. 223234, 2010.

[20] S. Bhattacharyya, G. Brebner, J. Eker, J. Janneck, M. Mattavelli, C. von Platen, and M. Raulet, *OpenDF - A Dataflow Toolset for Reconfigurable Hardware and Multicore Systems*,First Swedish Workshop on Multi-Core Computing, MCC , Ronneby, Sweden, November 27-28, 2008.

[21] PlanAhead Software Tutorial Partial Reconfiguration of a Processor Peripheral,UG744, 21 September 2010.

[22] MicroBlaze Processor Reference Guide Embedded Development Kit EDK 10.1i, UG081 (v9.0), 2008.