I.J. Image, Graphics and Signal Processing, **2012, 8, 57-65**
Published Online August 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2012.08.08

Hybrid Decoder Reconfiguration of AVS-P7 and MPEG-4 /AVC in the Reconfigurable Video Coding Framework

Yang Gaobo[1,a], Xiang Renhua[1,a], Zhang Zhaoyang[2,b]
[1] College of Information Science and Engineering, Hunan University, Changsha, China
[2] College of Communication and Information Engineering, Shanghai University, Shanghai, China
[a]yanggaobo@hnu.edu.cn, [b]zhyzhang@mail.shu.edu.cn

Abstract— With the rapid development of video coding technology, all kinds of video coding standards have been advanced in recent years with a variety of different and complex algorithms. They share common and/or similar coding tools, yet there is currently no explicit way to exploit such commonalities at the level of specifications or implementations. Reconfigurable video coding (RVC) is to develop a video coding standard that overcomes many shortcomings of current standardization and specification process by updating and progressively incrementing a modular library of components. In this paper, a hybrid decoder reconfiguration is instantiated in the RVC framework by grouping the coding tools from AVS-P7 and MPEG-4/AVC. Experimental results show that compared with MPEG-4/AVC baseline profile, the reconfigurable coding system reduces the computational complexity and guarantees the coding performance at low bit rate. Moreover, it enriches the RVC video tool library (VTL) by introducing the coding tools of AVS-P7, and also verifies the flexibility and re-configurability of RVC framework to meet the needs of different applications.

Index Terms— Reconfigurable video coding (RVC), Syntax parser, Decoder description, Hybrid decoder reconfiguration, Coding tool replacement

I. INTRODUCTION

In the past decades, there exist various video coding standards such as MPEG-x, H.26x, VC-1 (Video Codec 1) and AVS (Audio Video coding Standard of China) [1]. To meet the different requirements of various video applications, these video coding standards define different profiles and levels, respectively. A profile defines a set of coding tools or algorithms that can be used in generating a compliant bit-stream, whereas a level places constraints on certain key parameters of the bit-stream [2]. Although many of these video coding standards share common and/or similar coding tools, up to now there is still no explicit way to exploit such commonalities at the specifications or implementation level. The continuous improvements of video coding technology can not benefit us unless an old standard is replaced with a new one [3]. This usually results in the replacement of the existing multimedia devices with new ones supporting the new deployed standards. It hinders timely product improvements that technology innovations could potentially provide. Moreover, current codec level definition of video coding standards restricts the implementations to some profiles of a specific standard, which lacks flexibility and does not provide interoperability between different codecs.

To address the existing shortcomings of current standardization and specification process, reconfigurable video coding (RVC) standard intends to support the incremental deployment of new technologies. The basic idea of RVC framework is designing a dynamic data flow mechanism and constructing new video codecs by a collection of video coding tools from video tool libraries. With this objective, RVC framework is not restricted to specific coding standard, but defined at coding tools' level with interoperability to achieve high flexibility and reusability. Three elements are normative in RVC framework: decoder description (DD), video tool library (VTL) and abstract decoder model (ADM). With these elements, a standard or new decoder is able to be reconfigured in RVC framework. The most attractive features of the RVC standard are flexibility and reconfigurability. By updating and progressively incrementing a modular library of components, the RVC framework offers a great flexibility in selecting coding tools for decoder reconfigurations.

There are some existing works in the field of RVC. A conceptual view of RVC and its expected impact on the time reduction for deployment of new video coding solutions is presented in [3]. Christophe et al. discusses the automatic synthesis of parsers and validation of bitstreams within the MPEG RVC Framework [4]. It describes the methodologies and the tools for the validation of bitstream syntaxes descriptions as well as a systematic procedure for automatically synthesizing parsers from the bitstream descriptions. Ding et al. propose a decoder reconfiguration procedure in RVC framework by showing a decoder configuration architecture which successfully combines coding tools from AVS and MPEG VTL to form different decoding solutions [5]. Bitstream syntax schema of AVS intra decoder configuration is detailed to explain how to define the BS schema for a reconfigured bitstream. Ding et al. also reconfigure a new codec solution which replaced the inverse quantization block and inverse transform block of MPEG-4 Simple Profile with those

of AVS Part 2 [6]. In the example, coding tools from AVS Part 2 are used in the reconfigured decoder which yields both complexity reduction and improvement of performance in specific bitrate ranges.

AVS-P7 is also called AVS-M. It is fully defined in the seventh part of AVS [7]. It mainly aims at providing video decoding specification and standard for mobility systems and devices with limited computation capability and power consumption. AVS-P7 covers a wide range of applications such as mobile multimedia broadcasting, video conferencing, IP multimedia subsystem (IMS), video phone and video surveillance [8]. The extension of AVS-P7 into RVC VTL is worthy of investigation, simply because of its relatively easy configuration of corresponding codec to meet the specific application requirements. In this paper, a hybrid decoder reconfiguration is instantiated in the RVC framework by utilizing the inverse quantization (IQ) and inverse DCT (IDCT) modules to replace those of MPEG-4 AVC baseline profile. The involvement of AVS enriches current VTL of RVC and promotes the development of video coding technology.

II. About RVC

The emerging MPEG RVC standard presents an alternative paradigm for codec standardization and deployment aiming at providing a unified, dynamic, and incremental development, implementation, and adoption of standardized video coding solutions. MPEG RVC enables, in principle, the selection and usage of any arbitrary combinations of standardized basic coding algorithms, but obviously not all combinations are meaningful and interesting for a particular application. Such flexibility is achieved by adopting the concept of standardizing a unified library of video coding algorithms (at the moment taken from the existing standards, but incrementally upgraded with new successive algorithms) instead of adding more and more monolithic versions of new standard "profiles."

RVC offers a flexible mechanism of combing coding tools to reconfigure decoding solutions. The RVC framework adopts dataflow process formalism to modular designs, which is different from traditional design methods. Two standards are defined within the context of the MPEG RVC: Reconfigurable code framework (MPEG-B) [9], which defines the overall framework as well as the standard languages that are used to describe the different components of the framework, and the Video Tool Library (MPEG-C)[10], which defines the library of video coding tools (VTL) employed in existing MPEG standards. As shown in the RVC conceptual diagram (Fig.1) [11], the three types of decoders within the RVC framework all constructed using the MPEG-B standardized languages. Hence, they all conform to the MPEG-B standard. A Type-1 decoder is constructed using the FUs within the MPEG VTL only. Hence, this type of decoder conforms to both the MPEG-B and MPEG-C standards. A Type-2 decoder is constructed using FUs from the MPEG VTL as well as one or more proprietary libraries (VTL 1-n). It conforms to the MPEG-B standard only. Finally, a Type-3 decoder is constructed using some proprietary VTLs (VTL 1-n), without using the MPEG VTL. It also conforms to the MPEG-B standard only. An RVC decoder (i.e., conformant to MPEG-B) is reconfigured with coding tools described in VTLs according to the decoder description.

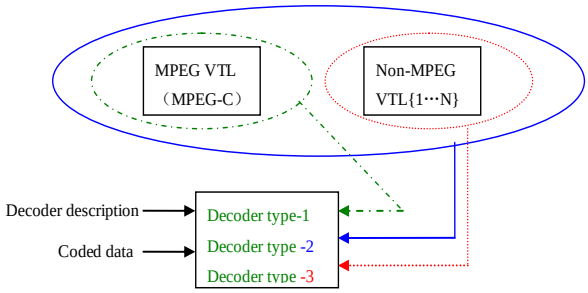


Fig.1 The conceptual view of RVC

A. Video tool library

Each coding tool within video tool library has been defined an unique serial number, by which RVC framework can call the related video tool to complete the current configuration, as shown in the process diagram (Fig.2). Each coding tool (FU) is a well self-contained modular element with the specification of its I/O interfaces. And RVC-CAL is a subset of the CAL Actor Language (CAL) normalized as a part of the RVC standard to describe the algorithm and behavior inside FU. It sets some restrictions to the data types, operators, and features when describing an FU. Compared with the traditional C/C++ programming language, CAL is clearly portable, less code and can be used on wide variety of different platforms [12]. Functionality of the coding tools and their potential concurrency are explicitly exposed to implementers by the specification formalism chosen [13]. This solution is by far a better starting point for any design and implementation methodology and process. As a textual language, Each Functional Unit (FU) is modularized to an actor in CAL. A CAL actor is a computational entity with interfaces (input and output ports), internal state and parameters. An actor is strongly encapsulated; it can neither access nor modify the state of any other actor. An actor may only interact with others by sending data (called tokens) along channels. During an execution (called firing), it maps input tokens onto output tokens and changes its internal state. As shown in a simple actor (Fig.3). CAL has another obviously advantage that independent of any platform. They can be transformed to each other for different platform, such as CAL2C, CAL2VHDL, CAL2JAVA, those transform tools were provided by Open RVC-CAL Compiler [14]. The Open RVC-CAL Compiler (orcc) can generate code for any platform, including hardware (VHDL), software (C/C++, Java…), and heterogeneous platforms (mixed hardware/software) from a platform-agnostic, high-level design.

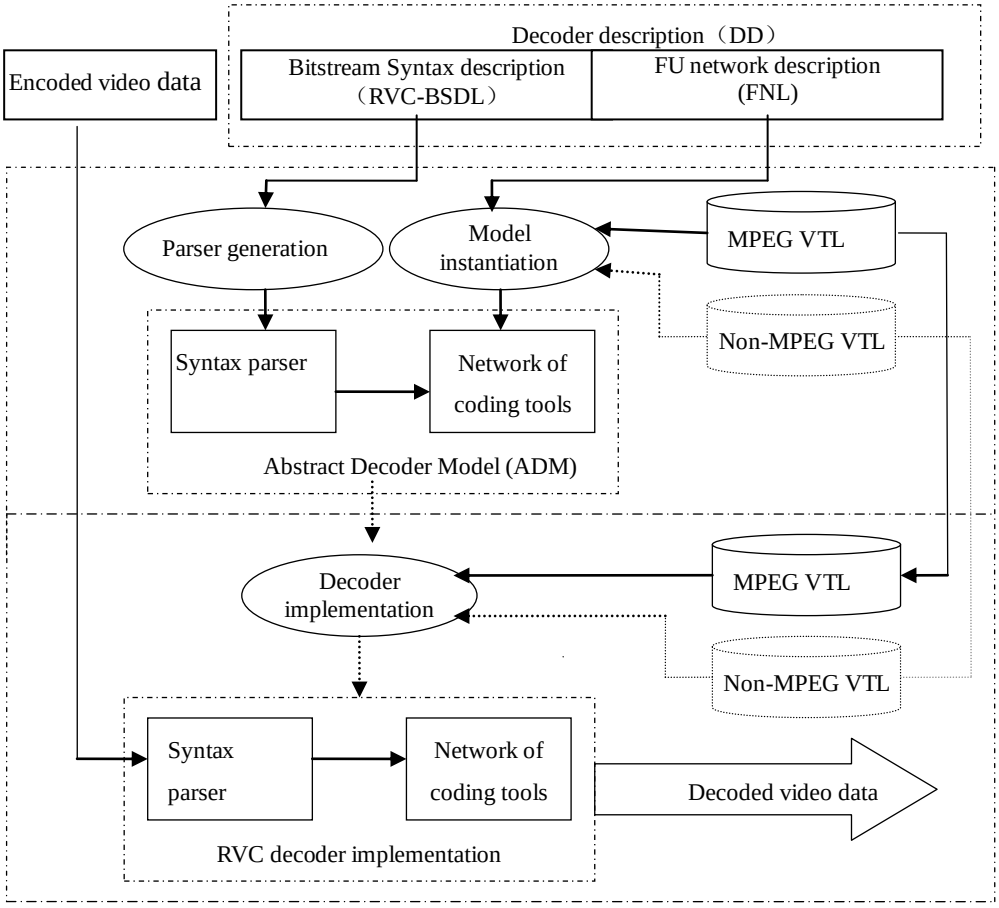


Fig.2 the RVC framework

```
actor Algo_IT4x4_1d () int(size=16) X ==> int(size=16) Y :
action X:[x] repeat 4 ==> Y:[ y[0], y[1], y[2], y[3]]
var
List(type:int(size=16), size=4 ) buf,
List(type:int(size=16), size=4 ) y
do
buf := [ x[0] + x[2],  x[0] - x[2], rshift(x[1],1) - x[3], x[1] + rshift(x[3],1) ];
y :=[ buf[0] + buf[3], buf[1] + buf[2], buf[1] - buf[2], buf[0] - buf[3] ];
end
```

Fig.3 Basic structure of a CAL actor

B. RVC Framework

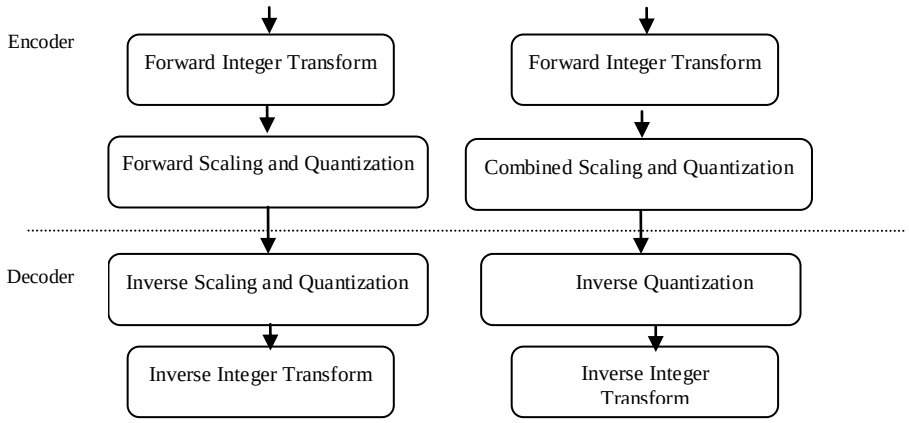
As shown in the components and instantiation process for ADM and the platform-dependent decoder implementation in RVC framework (Fig.2). The essential elements of RVC framework are:

Decoder Description creatively introduced in RVC framework provides reconfiguring decoder with necessary information. It consists of FU Network Description (FND) and Bitstream Syntax Description (BSD). Decoder Description and encoded video data comprise the reconfigurable bitstream, which is transferred to decoder by system layer. FND is written by the high-level XML-based Network Description Language, namely, FU Network Language (FNL). It describes all related FUs and interconnection between them. In RVC framework, when decoder received the reconfigurable bitstream, a DD decoder process is called immediately to initialize the interconnection of assigned FUs. A MPEG-21 Bitstream Syntax Description Language (BSDL) schema describes the syntax of the bitstream. In [4, 15], tools and methodologies for validation of BSDL syntaxes are described in full details as well as some examples of systematic procedures for the direct synthesis of syntax parsers in the CAL dataflow specification formalism.

Abstract Decoder Model (ADM), a behavioral model of the decoder composed of the syntax parser generated

from BSDL schema, FUs from the VTL and their connections. The ADM will be used to implement the decoder using proprietary tools and mechanisms.

Decoder Implementation, with those elements mentioned above, the final decoder implementation is platform-dependent and regarded as non-normative part of RVC framework. It is either generated by instantiating any implementation of proprietary VTLs which own identical I/O behavior of the FUs in standard VTLs, or obtained directly from the ADM by generating SW or HW implementations by means of appropriate synthesis tools [16, 17].



(a)Block diagram of traditional coding (b)Block diagram of PIT coding

Fig.4 The comparison diagram between traditional and PIT coding

```
<NALUnit>
 <startCode>00000001</startCode>
 <forbidden0bit>0</forbidden0bit>
 <nalReference>3</nalReference>
 <nalUnitType>20</nalUnitType>
 <payload>5 100</payload>
</NALUnit>
```

(a) the BSD fragement of AVS bitstream

```
<element name="NALUnit" bs2:ifNext="00000001">
<xsd:sequence>
 <xsd:element name="startCode" type="avc:hex4" fixed="00000001"/>
 <xsd:element name="nalUnit" type="avc:NALUnitType"/>
 <xdd:element ref="payload"/>
</xsd:sequence>
<!--Type of NALUint Type--!>
<!--and so on--!>
```

(b) a fragment from AVS bitstream schema

Fig.5 bitstream syntax description of MPEG-4 AVC

III. The Decoder Reconfiguration of MPEG-4 AVC Baseline Profile

A. A brief introduction of Audio Video Coding Standard of China（AVS）

The AVS video coding standard has attracted more and more attentions both from the industries and research institutes. Recently it has been accepted as an option by ITU for IPTV applications. As MPEG standards, the AVS standard is composed of several parts, such as system, video, audio, conformance testing, and reference software etc. AVS Part 7, also called AVS-P7, is for low resolution, low bit rate video applications, such as streaming, wireless multimedia communication etc. AVS-P7 shares the similar coding framework with AVS Part 2. However, there are difference in complexity and coding efficiency for them.

AVS-P7 follows the methodology of intra prediction in MPEG-4/AVC [18], which exploits the statistically spatial dependencies between pixels. There are 9 prediction modes for luminance blocks and 3 modes for chrominance blocks. For lower-bitrate mobile video applications, smaller block size will lead to better coding efficiency. Compared with MPEG-4/AVC, AVS-P7 takes the enhanced best mode to increase matching effciency between the best mode and the most probable

mode, reduce the number of coded bits and guarantee the image quality. All intra predictions based on 4×4 block instead of 16 × 16 block. Moreover, the number of adjacent pixels used for prediction is decreased from 17 to 9, in order to reduce the computational complexity. The transform algorithm of AVS-P7 [7] is also different from MPEG-4 AVC. It introduces the pre-scaled integer transform (PIT) technique which combines scaling, quantization and inverse scaling together at the encoder, thus only inverse quantization is needed at the decoder to reduce complexity of decoder, as shown in the comparison diagram (Fig.4). AVS-P7 takes 64-level step-length quantizer to quantify transformed residual coefficients instead of 52 in MPEG-4 AVC. The scope extension of quantization step can flexibly and accurately compromise between bit rate and image quality. A simple and effective inter prediction is adopted in AVS-P7. It does not include bi-directional prediction and weighted prediction, and limits the number of the reference frame less than two. Therefore, AVS-7 is more easily to be implemented on mobile devices with low complexity and low requirement of storage capacity.

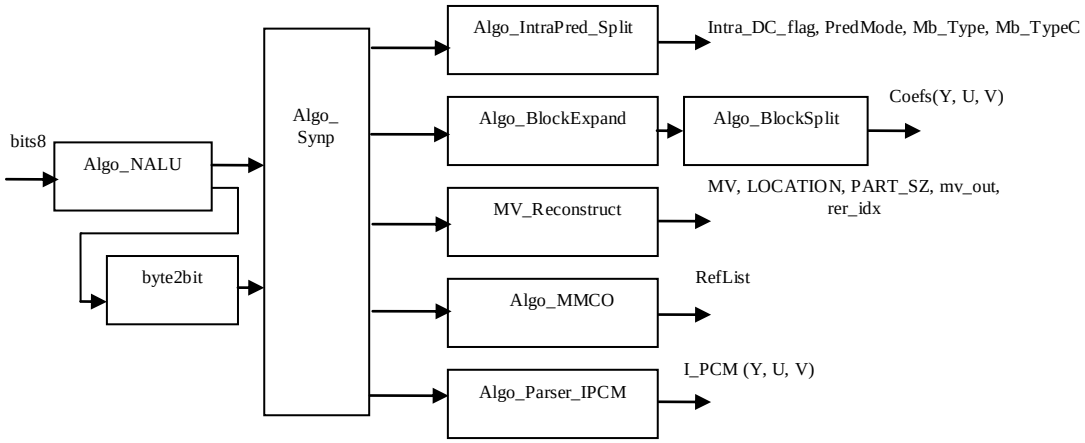


Fig.6 Architecture of Parser network in the reconfigured hybrid decoder

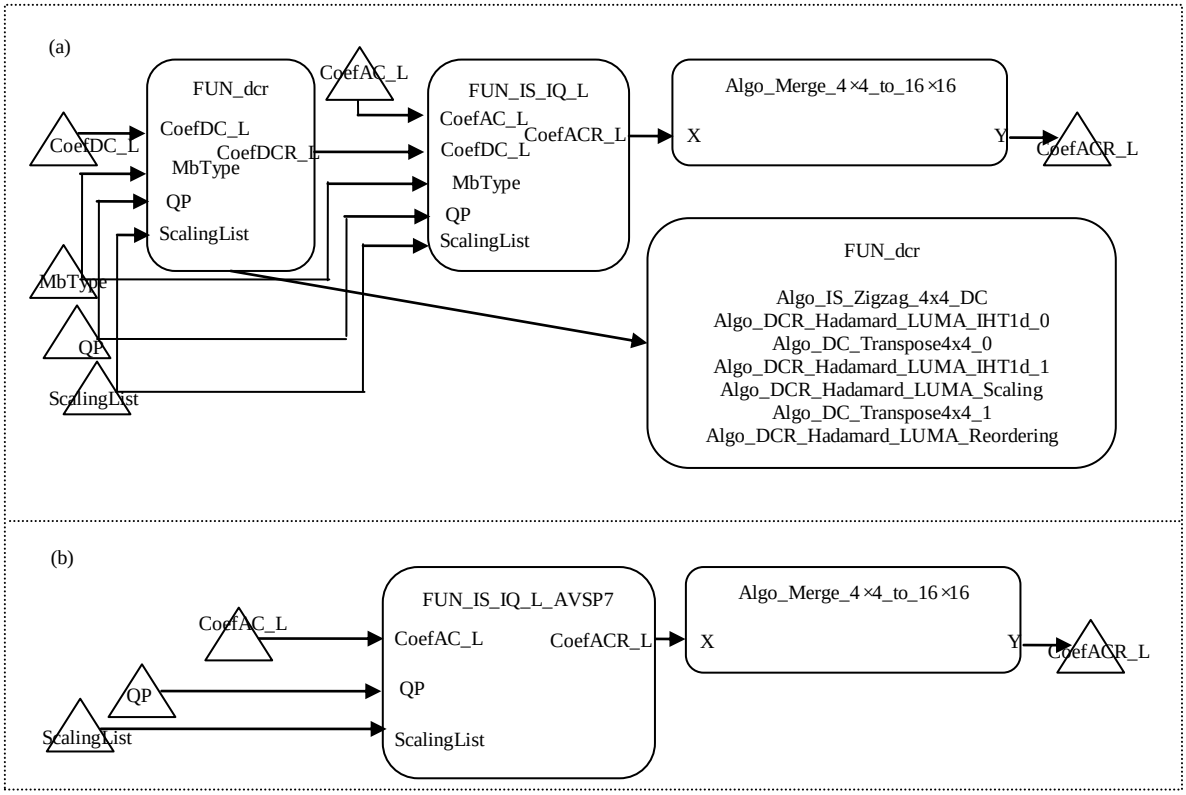


Fig.7 The implementation architecture of "Residuals Decoding" for the MPEG-4 AVC decoder (a) and the proposed hybrid decoder (b)

As mentioned above, the RVC framework provides a dynamic dataflow mechanism for constructing new video codecs by a collection of video coding tools from different video code standard, and AVS-P7 possesses several advantages in some certain situation. Although the Inverse Quantization (IQ) and Inverse Transform (IT) algorithms are completely different between AVS P7and MPEG-4/AVC, they both base on 4×4 blocks and have the same input and output token requirements. Thanks to the flexibility of the RVC framework, original MPEG-4 AVC baseline profile FUs (IT, IQ) can be easily replaced by the new FUs (IT, IQ) coming from AVS P7, the same principles to the functional block of Intra Prediction [19]. A new decoder solution is reconfigured in this paper. It is a hybrid decoder of AVS P7 and MPEG-4 AVC. The RVC framework provides a very interesting framework in which dynamic reconfigurations of decoders is straightforward. In the following, we describe the process of reconstructing the hybrid decoder within the RVC framework.

B. The generation of syntax parser

For the reconfiguration instantiation in this paper, encoder produces corresponding reconfigured bitstream and transmits it to decoder. Then decoder firstly parses the Decoder Description (DD). The Bitstream Syntax Description included in the DD explains how to fully describe the syntax structure of input bitstream and define corresponding Bitstream Syntax schema (BS schema). As shown in Bitstream Syntax Description fragment of MPEG-4 AVC (Fig. 5(a)) and BS schema fragment of MPEG-4 AVC codec (Fig. 5(b)). Before generating syntax parser from bitstream schema, one must guarantee that the schemas are correct, i.e. the schema reflects perfectly the structure of the encoded video data. So there must be a validation procedure, which guarantees that the bitstream is structured exactly as the schema describes it, then the generated parser can correctly parse the input bitstream [4]. Obviously, the Decoder Description guarantees the communication between encoder and decoder, and presented the flexibility and dynamic of reconfiguring decoder.

As stated above, the bitstream structure corresponding to the proposed hybrid decoder is described in BSDL, namely, a XML dialect. And FUs described in CAL can be also represented in a XML format: CALML. XSL Transformations are perfectly adapted to convert a bitstream schema written in BSDL into a parser in CALML [15]. As shown in the architecture of parser in the proposed hybrid decoder (Fig.6). “Algo_Synp” is generated by XSLT mentioned above. “Algo_IntraPred _Split” is employed to get the intra information and separate it into Y, U and V. “Algo_BlockExpand” gets and stuffs residual coefficients according “run, last, level” acquired from the “Algo_Synp”. And “Algo_ MMCO” is employed to update the “RefList”. “Algo_Parser_IPCM” derectly gets the pixel value of the current block.

(a) MPEG-4 AVC BP decoder:	(b) Reconfigured hybrid decoder:
network IS_IQ_IT_L	network IS_IQ_IT_L_AVSP7
entities	entities
Algo_IS_Zigzag_4×4_AC	Algo_IS_Zigzag_4×4_coef_AVSP7
Mgnt_IQ_INTRA16×16	Algo_IQ _4×4_coef_AVSP7 //inverse quantization
Algo_IQ_QSAndSLAndIDCTScaler_4×4	IT4×4_AVSP7 //inverse transform
IT4×4	
netwok IT4×4	netwok IT4×4_AVSP7
entities	entities
Algo_IT4×4_1d_0	Algo_IT4×4_1d_AVSP7_0
Algo_Transpose4×4_0	Algo_Transpose4×4 _0
Algo_IT4×4_1d_1	Algo_IT4×4_1d_AVSP7_1
Algo_Transpose4×4_1	Algo_Transpose4×4 _1
Algo_ IT4×4_Addshift	Algo_ IT4×4_Addshift_AVSP7

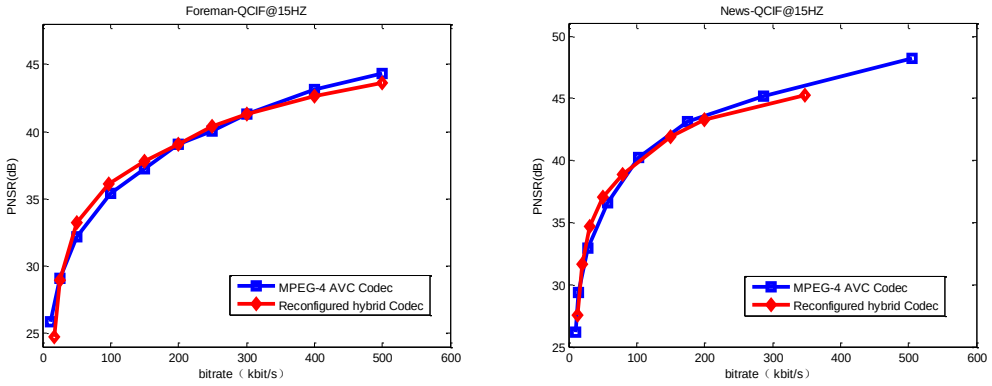
Fig.8 the structure of inverse scan, IQ and IT of AVC BP decoder and reconfigured hybrid decoder

C. Instantiation of reconfigured decoder

With those above-mentioned elements, the reconfigured decoder gets the attributes of each syntax element and the structure of current input bitstream through the Syntax Parser generated from BS schema, and gets required FUs and their connections for reconfiguring new decoder by parsing the FUs Network Description. Then Syntax Parser and those FUs are connected into a network called Abstract Decoder Model by parameter assignment and instantiation process, thus the decoder is configured successfully.

The structure of proposed hybrid decoder instantiation is similar to the MPEG-4 AVC Baseline Profile decoder, except that the IQ, IT block are replaced by those of AVS P7, and the reconfigured decoder discards a set of modules for intra 16×16 prediction mode (such as Inverse DC Hadamard transform, residual coefficient reordering). Figure 7 (a) shows the Y component implementtation architecture of “Residuals Decoding” within the MPEG-4 AVC decoder, and Fig.7(b) shows the corresponding functional modules within the proposed hybrid decoder. For U and V component, they are processed in the similar way as Y. These functional units

are themselves hierarchical compositions of actor networks. The "FUN_dcr" module of MPEG-4 AVC BP decoder is used to decode the DC coefficients for intra 16×16 prediction mode. It includes "Algo_IS_Zigzag_4x4_DC", "Algo_DCR_Hadamard_LUMA_IHT1d", "Algo_DC_Transpose4x4", "Algo_DCR_Hadamard_LUMA_Scaling" and "Algo_DCR_Hadamard_LUMA_Reordering". The "Algo_Merge_4x4_to_16×16" functional unit puts together 4x4 block in raster scan order to generate a 16×16 macroblock. Fig.8(a) illustrated the network of "FUN_IS_IQ_IT_L" within MPEG-4 AVC BP decoder. It completes the process of Inverse Scan, Inverse Quantization and Inverse Transform for residuals coefficients. The same function network of "FUN_IS_IQ_IT_L_AVSP7" within the proposed hybrid decoder is illustrated in Fig.8(b), of which the "Algo_IT4×4_1d_0" and the "Algo_IT4×4_1d_1" reused the same actor "Algo_IT4×4_1d", and the "Algo_Transpose4×4_0" and "Algo_Transpose4×4 _0" reused the same actor "Algo_Transpose4×4". The "Algo_ IT4×4_Addshift" of MPEG-4 AVC decoder can be replaced by "Algo_IT4×4_Addshift_AVSP7", which has the same input and output tokens as it. It is obvious that the reusability and exchangeability between different standard have been exploited in this instantiation, and the implementation architecture of reconfigured hybrid decoder is less complexity than the MPEG-4 AVC BP decoder.



(a) Foreman (b) News

Fig.9 the R-D performance of AVC BP and the reconfigured hybrid decoder

IV. Performance Comparisons of Reconfigurable Video Codec

Besides the predominant reconfigurable characters of RVC standard, it is necessary to make further performance comparison between the MPEG-4 AVC BP codec and the reconfigured hybrid codec. Since AVS-P7 is mainly stipulated for mobile video standard, the number of reference frame is generally less than two instead of sixteen in MPEG-4 AVC BP. Experimental results are shown for two video sequences in QCIF format. Both Foreman and News sequences are 200 frames, and they have different motion complexity. The overall experiment environments are set as follows: the simulation was implemented on a PC with Pentium(R) 4 CPU 3.00 GHZ 2.99 GHZ and 512MB of RAM, GOP=IPPP…., and frame rate=15 frame/second. A reconfigured bitstream is produced from MPEG-4 AVC reference software and provided as an input to the reconfigured decoder.

Fig.9 shows the Rate-Distortion (R-D) performance of MPEG-4 AVC BP and the reconfigured hybrid codec. Obviously, the two coding solutions can yield pretty nearly performance at low bitrates (<400 kbps), although the reconfigured codec gradually underperforms the MPEG-4 AVC BP as the bitrate increases.

A synthesis tool called Cal2C validates another implementation methodology of the MPEG-4 AVC BP dataflow program provided by "Open RVC-CAL Compiler (ORCC)". It translates RVC-CAL actors to C files. The "Decoding_Y" component of MPEG-4 AVC dataflow program is composed of 34 actor instantiations in the flattened dataflow program. Each actor instantiation becomes a C file containing all its action /processing with its overall action scheduling/control. In the same way, the "Decoding_Y" component of the proposed hybrid decoder includes 26 actor instantiations. A comparison of "Decoding_Y" component between the MPEG-4 AVC BP and the reconfigured decoder is shown in Table1. It is evident that the processing resources required for the proposed hybrid decoder are less than the MPEG-4 AVC BP decoder by a reduction factor between 1 and 2.

As analyzed above, compared with the MPEG-4 AVC BP decoder, the reconfigured decoder realizes complexity reduction and guarantees performance in specific bitrate ranges. Therefore, for mobile video applications, such as mobile communication and mobile television, which have constrains of low bitrate and low computational complexity, the reconfigured decoder has significant advantages.

TABLE1 A COMPARISON OF MPEG-4 AVC BP AND RECONFIGURED DECODER FOR "DECODING_Y" COMPONENT

Type of decoder	CAL Number of files	C actors Number of files
MPEG-4 AVC BP decoder	30	34
The proposed hybrid decoder	20	26
reduction factor	1.5	1.31

V. IMPLEMENTATION AND SIMULATION FOR THE PROPOSED DECODER CONFIGURATION

The reconfigured hybrid decoder in this paper consists of FUs written in RVC-CAL from the video tool library. Fig.10 illustrates the top-level view of the reconfigured decoder in this paper. The main functional blocks include the Serialize, Bitstream Syntax Parser, the reconstruction block, the inverse integer cosine transform(AVS P7), the inverse quantization (AVS P7), the frame buffer, the motion compensation module [20]. These functional units are hierarchical compositions of actor networks.

In the reconfiguration decoder, the input bitstream generated by MPEG-4 AVC reference software is serialized bit by bit through FU "Serialize" ("byte2bit" in Fig.6) before decoding process. The Syntax Parser network decodes the coming bitstream and produces context-control signal and a sequence of coefficients for each 4×4 block which will be used in the following networks. Then the following networks are to decode residuals (IQ and IICT) and implement prediction and motion compensation. The reconfigured hybrid decoder is compiled with RVC simulation tool (Open RVC-CAL Compiler) and the result of simulation shows that it can decode MPEG-4 AVC conformant bitstream correctly and elegantly.

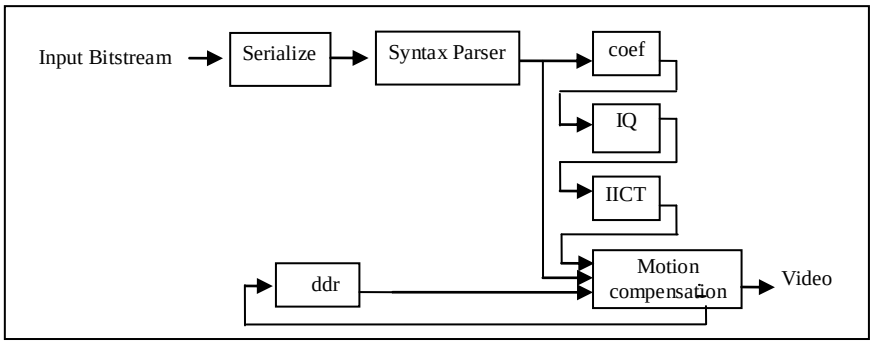


Fig.10 Top-level dataflow graph of the Hybrid decoder reconfiguration in RVC framework

VI. CONCLUSIONS

This paper describes the essential components of the MPEG Reconfigurable Video Coding framework based on the dataflow concept and has presented the new specification formalism called RVC-CAL. Compared with those traditional video coding standards, RVC framework provides a flexible mechanism to reconfigure video codec from the perspective of Functional Units, rather than the standard concept. The designer can devise proprietary codec system to meet different application requirements by avoiding unnecessary syntax elements and coding tools. This paper introduced the FUs of AVS P7 into Video Tool Library. Participation of AVS P7 in RVC enriches the significance of RVC framework and brings affirmative impulse for video coding field. In this paper, a hybrid decoder with FUs from MPEG-4/AVC and AVS P7 has been reconfigured. The instantiation shows great reconfiguration potential of decoder within RVC framework by replacing seemingly the existing FUs by new ones which have the same I/O tokens but different internal algorithm. This paper presents the decoder reconfiguration procedure in RVC framework, the definition of BS schema for reconfigured bitstream, and the generation of Syntax Parser. Experimental results show that compared with MPEG-4/AVC baseline profile, the reconfigured codec system reduces the computational complexity and guarantees the coding performance at low bit rate. Obviously, the decoder configuration architecture presented in this instantiation keeps the dataflow and modular mechanism of RVC. It can be further extended to support other new configurations, which will be our future investigation.

ACKNOWLEDGMENT

This work is supported in part by National Natural Science Foundation of China (No: 60702065), Special Pro-phase Project on National Basic Research Program of China (No: 2010CB334706) and Program for New Century Excellent Talents in University (NCET-11-0134).

REFERENCES

[1] Houjie Bi, Jian Wang et al. H.264/AVC and MPEG-4 Video Compression Standard for new generation Multimedia [M]. By Andrew Koenig Barbara Moo, 2009.

[2] HSIAO J M, TSAI C J. Analysis of an SoC architecture for MPEG reconfigurable video coding framework [C]. *Proc. ISCAS 2007*. New Orleans: IEEE Press:761-764, 2007.

[3] Mattavelli, M., I. Amer and M. Raulet. The reconfigurable video coding standard. *IEEE Signal*

Processing Magazine, 27: 159-167, 2010.

[4] Lucarz, C., J. Piat, M. Mattavelli. Automatic synthesis of parsers and validation of bitstreams within the MPEG reconfigurable video coding framework. *Journal of Signal Processing S*ystem, 63(2): 215-225, 2011.

[5] Ding, D., H. Qi, L. Yu, T. Huang, W. Gao. Reconfigurable video coding framework and decoder reconfiguration instantiation of AVS. *Signal Processing: Image Commun-ication*, 24: 287-299, 2009.

[6] Dandan, D., Lu Yu, C. Lucarz and M. Mattavelli. Video decoder reconfiguration and AVS extension in the new MPEG reconfigurable video coding framework. *Proceedings of the IEEE Workshop on the Signal Processing Systems*, Oct. 8-10, Washington, DC., pp: 164-169, 2008.

[7] AVS Video Expert Group, Information Technology – Advanced Audio Video Coding Standard Part 7: Mobility Video, in Audio Video Coding Standard Group of China (AVS), Doc. AVS – N1151, Dec.2004.

[8] Yu L., S. Chen and J. Wang, 2009. Overview of AVS-video coding standards. *Signal Processing: Image Communication*, 24: 247–262.

[9] ISO/IEC 23001-4, 2009. Information technology-MPEG systems technologies-Part 4: Codec configuration represen-tation.

[10] ISO/IEC 23001-4, 2010. Information technology-MPEG video technologies-Part 4: Video tool library.

[11] Amer I., C. Lucarz, G. Roquier, M. Mattavelli, M. Raulet, J.F. Nezan and O. Deforges, 2009. Reconfigurable video coding on multicore. *IEEE Signal Processing Magazine*, 26: 113-123.

[12] Eker J. and J. Janneck, 2003. CAL Language Report: Specification of the Cal Actor Language. University of Califonia, Berkeley CA.

[13] Gu R., J.W. Janneck, S.S. Bhattacharyya, M. Raulet, M. Wipliez and W. Plishker. Exploring the concurrency of an MPEG RVC decoder bases on dataflow program analysis. *IEEE Trans. Cirruits and System for Video Technology*. 19: 1646-1657, 2009.

[14] Open RVC-CAL Compiler [Online]. Available: http:// sourceforge.net/projects/orcc/.

[15] M. Bystrom, I. Richardson, S. Kannangara, M. de-Frutos-Lopez. Dynamic replacement of video coding elements. *Signal Processing: Image Communication*, 25(4):303-313, 2010

[16] Wipliez M., G. Roquier, M. Raulet, J.F. Nezan and O. Deforges, 2008. Code generation for the MPEG reconfigurable video coding framework: From CAL actions to C functions. *Proceedings of IEEE International Conference on Multimedia and Expo*, Hannover, pp: 1049-1052.

[17] Janneck J.W., I.D. Miller, D.B. Parlour, M. Mattavelli and C. Lucarz et al. 2008. Translating dataflow programs to efficient hardware: an MPEG-4 simple profile decoder case study. Design, Automation and Test in Europe (DATE), Munich.

[18] Wiegand T., G.J. Sullivan, G. Bjontegaard and A. Luthra, 2003. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Systems Video Technolo.*, 13: 560-576.

[19] Yanfei Shen, Dongmin Zhang, Lejun Yu, Chao Huang and Shouxun Lin, A simplified intra prediction method, Doc. AVS – M1419, 2004.

[20] Shuvra S. Bhattacharyya, Johan Eker, Jörn W. Janneck, etc. Overview of the MPEG reconfigurable video coding framework. *Journal of Signal Processing System*, 63(2): 251–263, 2011.

Yang Gaobo received his PhD degree from Shanghai University, China in March 2004. From that time, he has been working in Hunan University, China. In 2008, he was promoted as a professor because of his distinguished academic achievements. Currently, his research interests are image and video signal processing, digital media forensics.

Xiang Renhua received her master degree from Hunan University in 2010. She is working in ZTE Corporation, Shenzhen, China as an electronic engineer. Her research interests are multimedia communication.

Zhang Zhaoyang received his Bachelor degree from Xi'an Jiaotong University, China. He is now one of the most distinguished professors in Shanghai University. Currently, his research interests are image and video signal processing, multimedia communication.