A Proposed AVS Decoder Configuration in the Reconfigurable Video Coding Framework

Dandan Ding[1], Honggang Qi[2], Lu Yu[1], Tiejun Huang[3] and Wen Gao[3]
[1]Institute of Information and Communication Engineering, Zhejiang University, Hangzhou, China
[2]School of Information Science and Engineering, Graduate University of Chinese Academy of Sciences, Beijing, China
[3]School of Electronics Engineering and Computer Science, Peiking University, Beijing, China

***Abstract*—This demonstration shows an AVS intra decoder configuration in the RVC framework. It explains how to use the dataflow mechanism offered by the RVC framework to support AVS decoder configuration. It also shows the flexibility and convenience to reconfigure decoders in the RVC framework. In this work, the AVS VTL is established containing FUs from AVS. The proposed AVS decoder configuration is implemented by connecting some FUs from AVS VTL and reusing some FUs from MPEG VTL. The demonstration shows that the decoder can decode AVS conformance bitstreams correctly in RVC simulator.**

I. INTRODUCTION

This work belongs to ISCAS track: Multimedia System and Application.

Setup of the demonstration only requires one laptop computer which will be brought to ISCAS by the author performing the demonstration.

II. DEMOSTRATION EXPERIENCING

This work demonstrates the AVS decoder configuration in the Reconfigurable Video Coding (RVC) framework. The AVS Video Tool Library (VTL) is established to collect Functional Units (FUs) from AVS. What should be mentioned is that, FUs in AVS VTL is partitioned at the similar granularity as those in MPEG VTL with the purpose of reusing or exchanging FUs flexibly.

The demonstration takes AVS decoder configuration for instance to show how the RVC framework supports various decoder configurations flexibly and dynamically. It shows the flexibility of the dataflow mechanism offered by the RVC framework in connecting FUs to reconfigure decoders. The proposed AVS decoder configuration is implemented by connecting some FUs from AVS VTL and reusing some FUs from MPEG VTL. It reflects the key idea behind RVC which is to reuse as most as possible the algorithms or architectures which are common to several different standards and to reconfigure video decoders in a flexible way at the coding tool level.

The demonstration includes two parts: one is parallel implementation and the other is serial implementation. In the parallel version, Y, U and V component are processed in parallel and there are three networks for Y, U and V component separately. In the serial version, Y, U and V are processed serially. This demonstration takes 420 format for instance to explain the procedure.

For the detail of the proposed AVS decoder configuration, such as the complete architecture, FU partition principle, FU reusability and exchangeability, as well as BSDL Schema for AVS, **Error! Reference source not found.Error! Reference source not found.** can be referred.

Bitstream input to the AVS decoder configuration in this paper is generated by AVS reference software "RM" which conforms to AVS standard [3]. As the complete AVS decoder configuration is proceeding, this demonstration only shows the intra decoder configuration at this time.

The demonstration runs in RVC simulator OpenDF which is a plugin of Eclipse. The simulator is accessed through the "Run" dialog which provides access to various simulator options, specification of input and output files, and control over simulation logging. More information can be found in [4][5].

REFERENCES

[1] Dandan Ding, Honggang Qi, Lu Yu, Tiejun Huang and Wen Gao, "A Decoder Reconfiguration for AVS in the MPEG Reconfigurable Video Coding Framework," Circuits and Systems for Video Technology (submitted).

[2] D.Ding,etal., Reconfigurable video coding framework and decoder reconfiguration instantiation of AVS, Signal Process. Image Commun. (2009), doi:10.1016/j.image.2008.12.002.

[3] China Audio and Video Standard (AVS), Information technology advanced coding of audio and video part2: Video, GB/T 20090.2/-2006.

[4] http://opendf.sourceforge.net/eclipse.

[5] http://opendf.wiki.sour ceforge.net.

This work was supported by the National Science Foundation of China under Grants 90207005 and the Major State Basic Research Development Program of China (973 Program) under Grants 2009CB320900.

Authorized licensed use limited to: INSTITUTE OF COMPUTING TECHNOLOGY CAS. Downloaded on July 2, 2009 at 04:25 from IEEE Xplore. Restrictions apply.