March - April 2020
ISSN: 0193-4120 Page No. 4096 - 4101

The Parallel Decoding Architecture of Multi-Format VLD

Ki-Bum Suh
Professor, Dept. of Rail Electrical System Engineering, Woosong University, 171 Dong-Dae-jun-Ro Dong-ku, Daejeon, 34606 Korea
kbsuh@wsu.ac.kr

Article Info
Volume 83
Page Number: 4096 - 4101
Publication Issue:
March - April 2020

Article History
Article Received*: 24 July 2019*
Revised*: 12 September 2019*
Accepted*: 15 February 2020*
Publication*: 26 March 2020*

Abstract
Establishment and focus: In this paper, we propose an efficient architecture of MF-VLD (Multi-Format Variable Length Decoder), which is capable of variable length decoding and inverse quantization of various codec standards. The proposed MF-VLD is designed to be suitable for MPSOC (Multiprocessor System on Chip), and the bandwidth of AHB bus is reduced by applying bit-plane algorithm to inverse quantized data. It supports H.264, MPEG-2, MPEG-4, AVS, and VC-1 codec standards.

System: Parallel Decoding method using multi-processor is verified using INEXT board, which is emulation board with Xilinx Vertex5 XC4VL330 FPGA. The designed MF-VLD operates at 200 MHz in a 0.18 μm process and is approximately 620 K gates in size. The memory used is about 27 K bytes. In order to reduce the chip size, each variable-length decoding modules are arranged independently without combining, so that it is easy to add and remove according to the market situation later, and the design and verification shorten the time.

Keywords: *Multi-processor, Multi-Format VLD, Parallel Decoding, Video Codec, MPSOC.*

1. Introduction

As H.261 became the first international standard for video compression in 1990, video codecs have been constantly evolving. Currently, MPEG-2 is widely used for television broadcasting, and H.264 / AVC is also used for domestic terrestrial digital multimedia broadcasting (DMB), European satellite broadcasting, and digital video broadcasting-handheld (DVB-H). For UHD broadcasting, H.265 / HEVC is used. DivX and XviD, MPEG-4 series, are also widely used for web and mobile, and AVS is developed and used as a national standard in China, which has a huge multimedia market. In addition, the VC-1, developed by Microsoft, is standard on high-definition DVD (HD DVD), Blu-ray, and Xbox 360[1]. Therefore, there is a need for production of products that support high-definition video and process various video codecs.

There has been much research on Multi-Format Video Decoder (MFD). Among the previously published studies, the high-definition multi-format video decoder [2] with separate structure supports the decoding of high-definition video and is designed with separate structure for MPEG-2, MPEG-4, H.264, and VC-1 codec. Although similar in this regard, this paper shows a big difference in that it is designed for support of AVS codec and multiprocessor. In addition, there are good studies on MFD such as Chien [3], Cheng [4], and Jo [5], but they show a big difference from this paper in terms of supporting codec and design architecture. Efficient video decoding of H.264 and VC-1 using parallel cores [6] can be

March - April 2020
ISSN: 0193-4120 Page No. 4096 - 4101

used as a multi-decoder combined with MF_VLD of this paper by suggesting a parallel processing method using multi-core macroblock lines. For the HEVC decoder, the HEVC video decoder Chip [7-10]can be combined with this proposed design.

2. The architecture of MF-VLD

The architecture of the proposed MF-VLD is shown in Figure 1. It is designed to be combined with multiprocessor using AHB 32bit bus. This module consists of AHB REG_slave, AHB READ_Master, Stream Buffr, B-Direct SRAM, AHB WRITE_Master, BitPlane processing unit and VLD of each image standard.

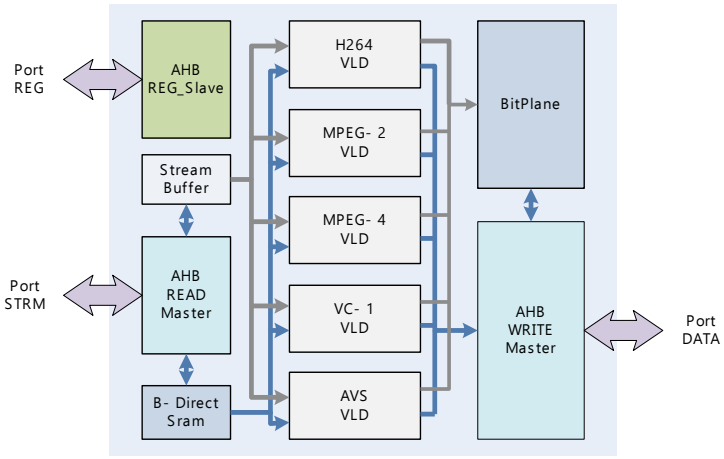

Figure 1. The architecture of the proposed MF-VLD.

AHB REG_Slave receives register data from external controller and transfers decryption status of MF-VLD to external controller. AHB READ_Master reads video stream and reads data from external memory (B-Direct Sram). WRITE_Master transmits the decoded data to Port DATA, Stream Buffer stores the video stream, and B-Direct Sram stores the motion vector of the reference picture to support the B-Direct prediction mode for B pictures. The remaining modules consist of a module that processes bit-plane information on inverse quantized data and a variable length decoding module of H.264, MPEG-2, MPEG-4, AVS, and VC-1. The operation of MF-VLD is as follows. When the input signal is received from the external controller and the start signal is received, the MF-VLD reads the external video stream through READ_Master and stores it in the internal memory. According to the type of the stored video stream, the variable length decoding module reads the video stream and starts decoding. After decoding the macroblock, it is stored in external memory through WRITE_Master. Inverse quantization is saved in external memory after bit-plane processing. Finally, the decoded data is stored in an external memory area in units of macroblock lines.

2.1 Bit-plane application of inverse quant-ization data

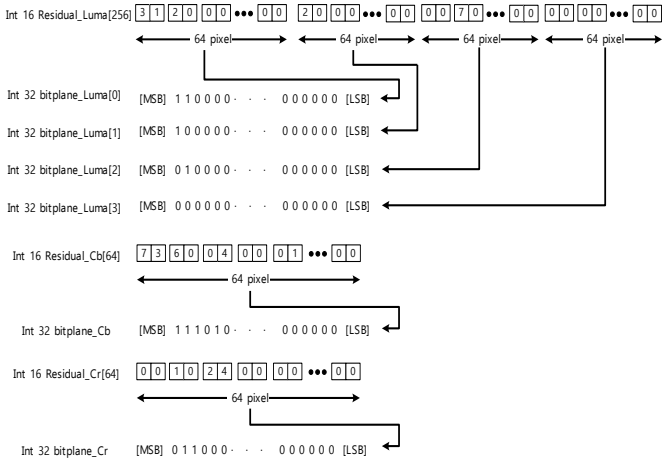

Figure 2. Bit-plane configuration

Most inverse quantization data has a value of zero. By using the characteristics of the inverse quantization data, zero-value data is not stored in the external memory, but only a valid value can be stored to reduce the bandwidth of the bus. Figure 2 shows how the bit-plane is constructed for inverse quantization data. It collects two consecutive coefficients and determines whether all of them are 0 or not, and if they are all 0, the corresponding bit-plane value is written as 0, and if any pixel is not 0, 1 is written. First, since 256 pixels of luminance are composed of four blocks of 64 pixels, four registers of bitplane_luma represent them, and two registers of bitplane_Cb and bitplane_Cr are represented for the color

March - April 2020
ISSN: 0193-4120 Page No. 4096 - 4101

difference (Cb, Cr). In other words, if you have six 32-bit registers for bit-plane values to store zero positions for luminance and chrominance, only valid data needs to be stored in external memory. If the bit-plane is not applied, it should be stored in the external memory using 192 transfers when using 32bit bus for 256 luminance and 128 chroma pixels.

Table 1. Average number of inverse quantized data transmissions per macroblock

	H.264	MPEG-2	MPEG-4	AVS	VC-1
Video Sequence	Blue sky	BQ terrace	BQ terrace	BQ terrace	tractor
Trans. No	21	18	46	56	19
Video Sequence	Sun flower	Sun flower	cactus	Basket ball	Sun flower
Trans. No	19	13	31	22	14

Table 1 shows the average number of times that a bit-plane value and valid inverse quantization data should be transmitted by 32 bits when bit-plane is applied to two video sequences for each codec. Although the number of transmissions varies depending on the codec and the type of video, the number of data transmissions can be reduced by about 90 to 50%.

2.2 Minimizing Internal Memory

Since the MF-VLD can decode the motion vector for the time-directed prediction mode in the case of a B picture, all motion vectors of the reference picture must be stored in the memory. The size of the memory used depends on the size of the image to be supported, but the MF-VLD of this paper requires a large memory size because it supports up to FHD (1920x1088) image size.

Table 2. Memory size of motion vector of reference picture

	H.264	MPEG-4	VC-1
Size of Ram	27bit x 32640	24bit x 2640	24bit x 8160

There are 8160 (120 x 68) macroblocks in the FHD image, H.264 and MPEG-4 store four motion vector values per macroblock, and VC-1 stores one motion vector value per macroblock. Saving. Even if you share the three-codec memories shown in Table 2, you must have about 110K bytes of memory inside. Having such a large memory inside can have a huge impact on chip size. In order to reduce the usage of internal memory, MF-VLD stores the decoded motion vector in external memory in the case of I or P pictures, and then reads them in macro block line units from the external memory in internal memory when decoding B pictures. In order to reduce the reading time, two memory (27bit x 480) can be stored to store one line of macro block, and the motion vectors of the next decoding line are read in advance to prevent the decrease of decoding speed.

2.3 The Architecture for Processing Multi-processors

Since MF-VLD performs only variable length decoding and inverse quantization, it is designed to function as a perfect decoder in combination with a multiprocessor. Figure 3 shows how each processor takes one macroblock line and decodes it in parallel using eight processors per frame.

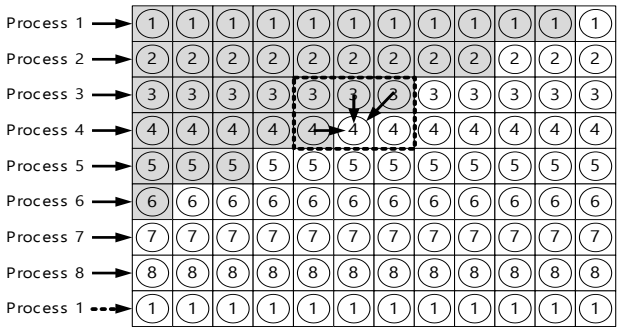

Figure 3. Parallel Decoding of Multiprocessors

Figure 4 shows the operation flow between the processor and MF-VLD. When the processor starts MF-VLD, the MF-VLD checks the COREXSTART register, which holds status information about the memory area to store the decoded data. It stops until the memory area becomes empty and then operates again. After decoding one macro block line, MF-VLD marks the memory area where the decoded data is stored

March - April 2020
ISSN: 0193-4120 Page No. 4096 - 4101

in the COREXDONE register so that the processor can read it.

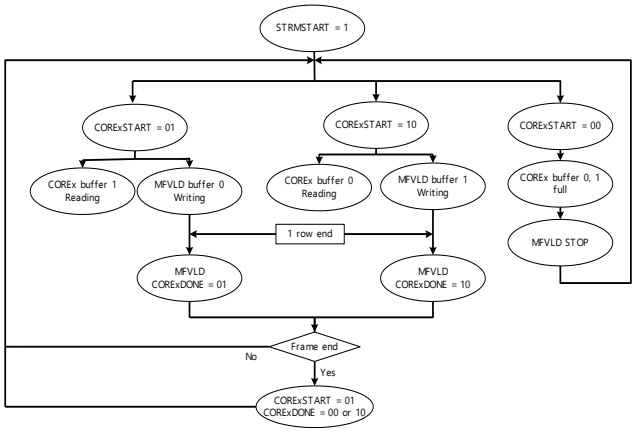

Figure 4. Parallel Decoding of Multiprocessors

3. Results and Discussion

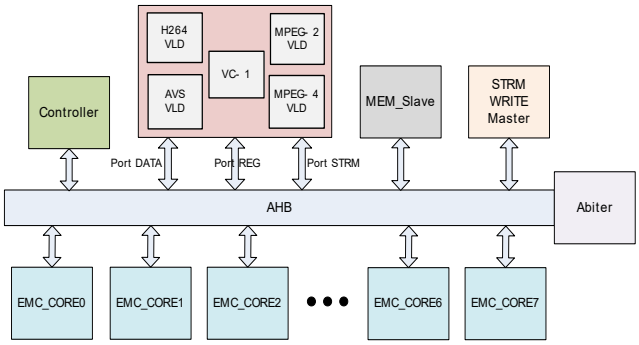

Figure 5. Simulation Environment of MF-VLD

The simulation environment of the proposed MF-VLD is simulated by adding 8 virtual processor modules(EMC_COREx) as shown in Figure 5. The results were confirmed for 20 images of each codec.

Table 3 shows the average cycles for decoding one macroblock for each codec. The above cycle is the time taken to store the decoded data to external memory via the bus. It also includes bandwidth to read external memory from eight virtual processors, which is expected to be similar to actual performance.

Table 3. The average decoding cycles per 1 MB for each codec

	I picture	P picture	B picture
H.264 CAVLD	436 cycle	381cycle	362cycle
H.264 CABAD	1243 cycle	966 cycle	577cycle
MPEG-2	540 cycle	547cycle	557cycle
MPEG-4	860 cycle	550cycle	580cycle
AVS	840 cycle	707cycle	820cycle
VC-1	680 cycle	620cycle	604cycle

The designed MF-VLD was FPGA verified with an INEXT board using Vertex5 XC5VLX330 chip. The inext board is designed for the purpose of the simulation accelerator. The INEXT board connects the PC to the board and outputs the results to the Modelsim simulator for viewing as waveforms. Figure 6 shows the image of downloading the bit-file generated after the synthesis and P & R process for FPGA verification to the board through the simulator. In the FPGA, all 20 images of each codec were tested and the results were confirmed.

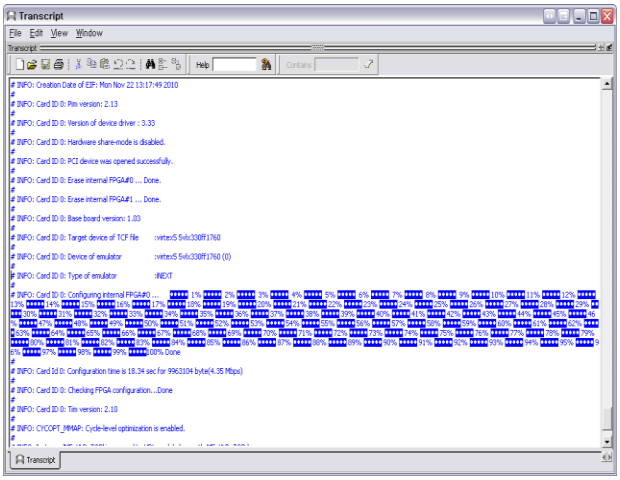

Figure 6. Bit file download Image

Figure 7 shows the result of P & R of MF-VLD on Vertex5 chip. The MF-VLD uses a 55,668 LUT, which is about 60% of the chip, with an operating speed of 50 MHz in an FPGA. This MF-VLD is designed to operate at the speed of 200 MHz in the 0.18 μm process. The MF-VLD is designed to operate at a speed of 200 MHz in a 0.18 μm process, and Table 4 shows the gate sizes for each codec. Except for the variable-length decoding module, the control and common modules are about 41 K gates and the overall size is about 707

K gates. The internal memory used uses about 23 K bytes.

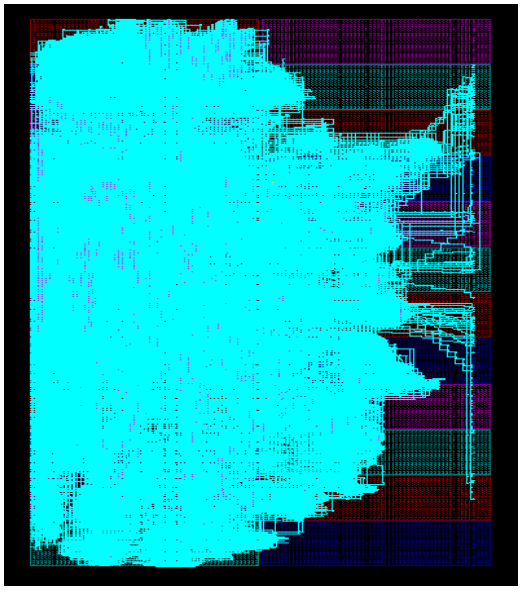

Figure 7.P & R of MF-VLD in XC5VLX330 FPGA

Table 4: Gate size for each codec

	H.264	MPEG-2	MPEG-4	AVS	VC-1
gates	398 K	34 K	72 K	85 K	77 K

4. Conclusion

In this paper, we propose an efficient structure of MF-VLD including variable length decoders of H.264, MPEG-2, MPEG-4, AVS, and VC-1 codecs that can handle high performance and higher quality. The proposed architecture minimizes the performance reduction due to bus bandwidth by proposing a bit-plane for inverse quantization data. In addition, the use of the internal memory is greatly reduced by storing the motion vector of the reference picture in the external memory. In order to reduce the chip size, each variable-length decoding module is arranged independently without combining, so that it is easy to add and remove according to the market situation later, and design and verification shorten time. If the proposed MF-VLD is manufactured as a multi-format video decoder in combination with a multiprocessor capable of parallel processing, it can be used for high-definition display devices and various mobile devices as MPSOC capable of decoding the most kinds of video compression standards.

Acknowledgment

This research is based on the support of 2019 Woosong University Academic Research Funding.

References

[1] Yo-Han Lim, Jung-Sun Kang, An efficient architecture of bitplane coding with high frame rate for VC-1. Signal Processing: Image Communication. 2008 Oct:62(1): 692~8. DOI : 10.1016/j.image.2008.08.001

[2] M. Hase, et al., Development of Low-power and Real-time VC-1/H.264/MPEG-4 Video Processing Hardware, Design Automation conference, 2007

[3] Chih-Da Chien, et al., A 252kgates/71mW Multi-Standard Multi-Channal Video Decoder for High Definition Video Applications, ISSCC Dig. Tech. Papers, 2007

[4] Chi-Cheng,Yung-Chang Chang, et al., A full-HD 60fps AVS/H.264/VC-1/MPEG-2 video decoder for digital home applications, Proceedings of 2011 International Symposium on VLSI Design, Automation and Test, April 2011

[5] Jo, H.-H., Ahn, Y.-J., Kang, D.-B., Ji, B., Sim, D.-G.: Flexible multi-core platform for a multiple-format video decoder. J. Signal Process. Syst. Signal Image Video Technol. 80(2), 163–179 (2013)

[6] J.-Y. Lee, J.-J. Lee, S.M. Park, Multi-core platform for an efficient H.264 and VC-1 video decoding based on macroblock row-level parallelism, Published in IET Circuits, Devices & Systems, February 2009

[7] Yang A, Troup M, Ho JWK. Scalability and validation of big data bioinformatics software. Comput Struct Biotechnol J 2017:8. Article in press.

[8] Hyunmi Kim, Seunghyun Cho, Kyungjin Byun and Nak-Woong Eum, Multi-core based HEVC hardware decoding system. 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), Chengdu, 2014, pp. 1-2. **DOI:** 10.1109/ICMEW.2014.6890626

March - April 2020
ISSN: 0193-4120 Page No. 4096 - 4101

[9] Chi, Chi Ching & Mesa, Mauricio & Lucas, Jan & Juurlink, Ben & Schierl, Thomas. (2012). Parallel HEVC Decoding on Multi- and Many-core Architectures. Journal of Signal Processing Systems. 71. 1-14. **DOI:**10.1007/s11265-012-0714-2.

[10] Ahn, Y., Yoo, J., Jo, H. et al. Software pipelining with CGA and proposed intrinsics on a reconfigurable processor for HEVC decoders. J Real-Time Image Proc 16, 2173–2187 (2019). DOI:10.1007/s11554-017-0729-9