AVS 3D Real-time Decoder Design and Implementation Based on FPGA/SoC Platform

REN Peng-fei[1]，YU Hong-yang[1]

[1]Research Institute of Electronic Science and Technology, University of Electronic Science and Technology of China, Chengdu Sichuan 611731, China

Keywords: 3D video, stereo-packing algorithm, decoder design, FPGA/SoC Co-platform

Abstract. AVS(audio video coding standard)Group formulates stereo-packing scheme aimed at 3D video. In this paper, based on stereo-packing algorithm, using FPGA hardware accelerate module to parse the stereo-packing ES stream syntax element and cooperating with the Xilinx ZYNQ 7020 SoC development board ,we complete the AVS 3D decoder on FPGA/SoC Co-platform. Using HDMI port to export the decoded data to the 3D display device, we get the 3D video with depth information and verify the validity of AVS 3D real-time decoder.

Introduction

AVS[1] is the abbreviation of audio and video coding standard, it is the second generation codec standard in China. In the end of 2008, AVS Group began to formulate stereoscopic video codec standard which was on the basis of stereo-packing algorithm[2]. The stereoscopic video is captured by two cameras whose base-lines are parallel. Based on the correlation between the left view and the right view and each view’s time and space correlation, we jointly encode the left and right view video into one bit-stream; at the decoder side, we decode one stereo-packing bit-stream to reconstruct two channels of left and right view video data. This scheme has more compression efficiency compared with the traditional simulcast scheme, and has some advantages such as lower encoding complexity and better compatibility compared with other stereoscopic video codec standards.

But now, there is no AVS 3D real-time decoder design and implementation base on FPGA/SoC platform. In this paper, we use Xilinx ZYNQ 7020 development board as the SoC platform. ZYNQ 7020 is a SoC system which has two internal M9 processing system (PS) hardcore chips. It has some advantages such as high level of integration, strong control ability and good software commonality. We use one master PS as top-level control system to complete the external interface communication with the 3D ES stream and the display of the decoded images; the other slave PS and some hardware acceleration modules (including ES stream syntax element parsing module, CABAC and CAVLC) work together to complete the stereo-packing decoding algorithm; the two PSs cooperate to implement the design of AVS 3D real-time decoder based on FPGA/SoC platform. At last, by means of HDMI port, we export the decoded stereoscopic video including left and right view data to the 3D display device. Through the viewpoints interlacement, we can get the 3D video with disparity information and verify the validity of the design.

AVS 3D decoder algorithm flow

AVS 3D decoder is based on stereo-packing algorithm. AVS 3D ES stream includes two channels of left view and right view data at the same time. We can use the disparity of left and right view to obtain depth information and use the viewpoints interlacement to obtain the final 3D video. 3D decoder’s algorithm flow is shown in Figure 1. Firstly, the 3D decoder reads 3D ES stream and decodes the ES stream frame by frame, the principle is similar to the traditional 2D decoder. Secondly, we get one frame of stereo-packing image which includes both left view and right view data. The base-layer images of left and right view can be obtained by viewpoints separation processing. Thirdly, we interpolate the base layer images by inter layer up-sample filter to get left and right view’s enhancement-layer images for 3D display. Finally, when the stop code is obtained, we know that the decoding process is over and input the decoded image data to the subsequent display modules.

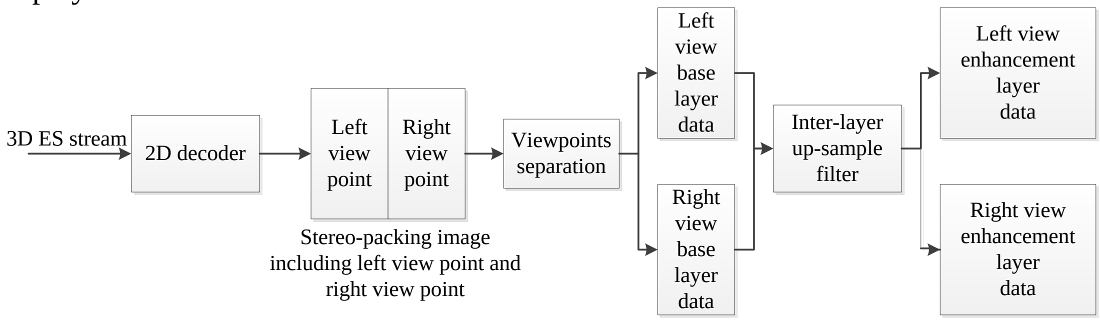
3D ES stream
2D decoder
Left view point
Right view point
Stereo-packing image including left view point and right view point
Viewpoints separation
Left view base layer data
Right view base layer data
Inter-layer up-sample filter
Left view enhancement layer data
Right view enhancement layer data

Figure 1 The stereo-packing algorithm flow of AVS 3D decoder

Through the processing of viewpoints separation, the left and right base-layer images' horizontal resolution are reduced by half. Because of the strong spatial correlation between base-layer image and enhancement-layer image, the inter layer motion vector and texture information have proportional relationship. For that reason, we can interpolate the base-layer image in horizontal dimension to get the enhancement-layer images with horizontal doubled resolution. For the sake of guaranteeing the enhancement-layer images' quality, we need to set proper kernel vector of the up-sample filter[3]. The principle is shown as in equation (1):

$$h_k = W_k b_k \tag{1}$$

In this equation, $\mathrm{h_k}$ represents for enhancement layer image, $\mathrm{b_k}$ represents for base layer image, and $\mathrm{W_k}$ represents for the optimal kernel vector of the up-sample filter. Due to the inverse quantization, motion compensation and other irreversible process during decoding, the base-layer image $\mathrm{b_k}$ contains useful signal $\mathrm{s_k}$ and the noise $\mathrm{e_k}$, shown as in equation (2):

$$b_k = s_k + e_{\mathrm{k}} \tag{2}$$

In the up-sample filter, the cost function is the MSE signal $\mathrm{E}\{\|\mathrm{b_k} - \mathrm{h_k}\|^2\}$, where $\mathrm{E}\{\cdot\}$ represents for mathematical expectation operator. According to the principle of least squares estimation method, the kernel vector of the optimal up-sample filter $\widehat{\mathrm{W}}_{\mathrm{k}}$ should guarantee the minimum cost function, so that we can get equation (3):

$$\widehat{W}_k = arg\ min_{W_k} E\{\|b_k - h_k\|^2\} \tag{3}$$

By taking a derivation of equation (3), we can deduce the optimal kernel vector of up-sample filter should satisfy equation (4):

$$\widehat{W}_k = \ (HR_b)(HR_bH^T + R_{e_k})^{-1} \tag{4}$$

Where H is the down-sample filter when AVS 3D encoder use it to get the 3D ES stream, $\mathrm{R_b}$ is the autocorrelation matrix of base-layer useful signal, $\mathrm{R_{e_k}}$ is the autocorrelation matrix of base-layer noise. At the encoder side, we already know that H = {2,0,-4,-3,5,19,26,19,5,-3,-4,0,2}. According to recursive least-squares algorithm, we can calculate that the optimal up-sample filter needs 6 taps, and optimal kernel vector $\widehat{\mathrm{W}}_{\mathrm{k}}$ should be {1,-5,20,20,-5,1}.

The base-layer images with a resolution of $\mathrm{M}/2 \times \mathrm{N}$ need to be interpolated in horizontal dimension. We should insert one half-pixel $\mathrm{v_i}$ between two horizontal adjacent base- layer pixels $\mathrm{x_{i,j}}$ and $\mathrm{x_{i,j+1}}$. The inter layer interpolation is shown in equation (5):

$$v_i = \sum_{i=1}^{i+k}\left(x_{i-\mathrm{k}/2,\mathrm{j}} \times w_k\right),\ \mathrm{i} = 1,2,\cdots\cdots,\mathrm{M}/2\ ,\ \mathrm{j} = 1,2,\cdots\cdots,\mathrm{N} \tag{5}$$

Using the horizontal interpolation, we can obtain the enhancement-layer images with a resolution of $\mathrm{M} \times \mathrm{N}$ from the base-layer images with a resolution of $\mathrm{M}/2 \times \mathrm{N}$. Where k represents for the number of up-sample filter taps, $\mathrm{w_k}$ represents for the value of each tap.

Since there are two left and right viewpoints during the process of AVS 3D decoding, AVS standard introduces inter-view prediction to get higher compression efficiency. Inter-view prediction uses the left view base-layer image to predict the right view base-layer image, the principle is shown in Figure 2. The left view base-layer image should be decoded independently; in the right view, the first frame is inter-predicted form the reconstructed I frame in the left view; other P frames in the right view can reference either the previous P frames in the same view or the corresponding simultaneously reconstructed P frames in the left view. Inter-view prediction for the B frame does not affect coding performance; therefore, references for the B frame can only be reconstructed from forward and backward directions in the same view.

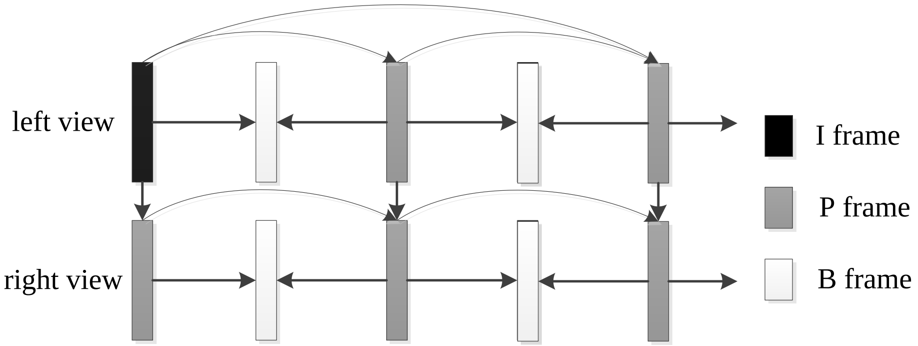

Figure 2 Inter-view prediction structure in AVS

In the display side, we use viewpoints interlacement[4] to get AVS 3D video. The principle is shown in Figure 3, we jointly use spatial multiplexing and time multiplexing to get 3D video. In spatial multiplexing process, the left and right view appear in edge-to-edge format; in time multiplexing process, the left and right view are interlaced like a frame or field simultaneously. During the process of capturing the 3D video, point A is in a distance of Da to the camera lens, and respectively maps to positions a1 and a2 in the left and right view image; point B is in a distance of Db to the camera lens, and respectively maps to the positions b1 and b2 in the left and right view image. The distance between a1 and a2 and the distance between b1 and b2 is the disparity information. Through the viewpoints interlacement, the viewer’s left and right eyes respectively watch the position $a1'$ in the left view image and the positon $a2'$ in the right view image, so that we can get the image A' which is in a distance of Da' to the viewer; in the same way we can get the image B' which is in a distance of Db' to the viewer. Owing to the difference between Da' and Db', the depth information of left and right view images can be obtained.

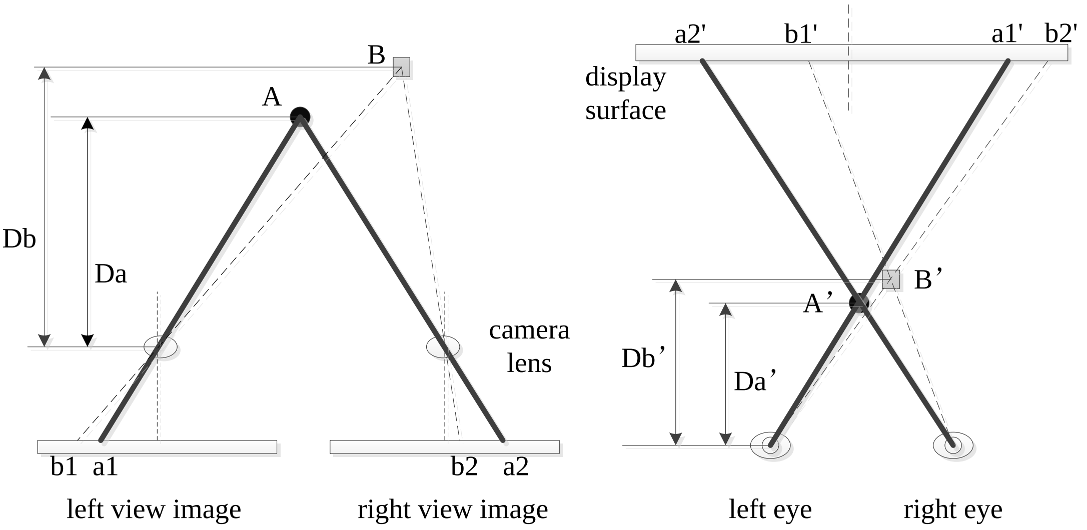

Figure 3 The display principle of AVS 3D video

CABAC and CAVLC syntax element parsing hardware acceleration module design

In this paper, we use hardware acceleration module to parse the syntax element. There are CABAC and CAVLC two ways to do so. CABAC[5] is the abbreviation of context-based adaptive binary arithmetic coding, and CAVLC[6] is the abbreviation of context-based adaptive variable length coding. The design of syntax element parsing hardware acceleration module is shown in Figure 4.

The task of input stream management module is to read the original ES stream. $\mathrm{indata}[7:0]$

stores 8 bits of the original ES stream; avail_n detects the number of available bits of the input stream; strobe checks the validity of the whole input stream; when the ES stream is parsed, req requests the input stream management module to read the next ES stream.

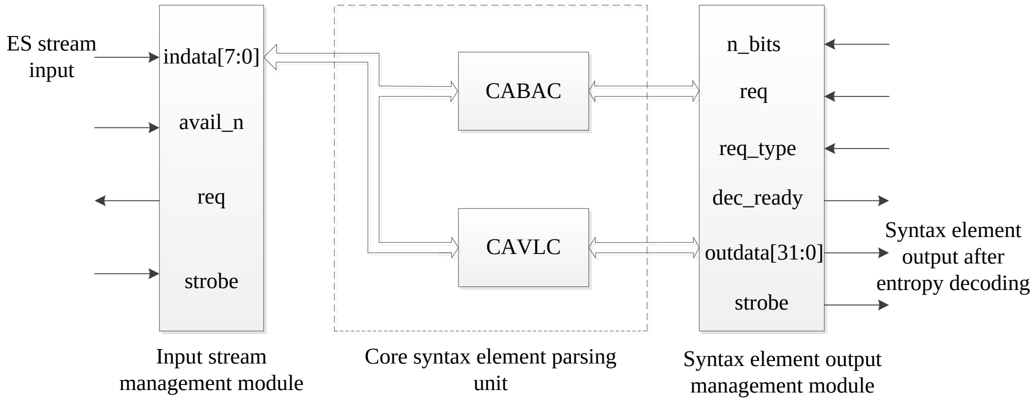
ES stream input
indata[7:0]
avail_n
req
strobe
CABAC
CAVLC
n_bits
req
req_type
dec_ready
outdata[31:0]
strobe
Syntax element output after entropy decoding
Input stream management module
Core syntax element parsing unit
Syntax element output management module

Figure 4 The structure of AVS syntax element parsing module

The task of syntax element output management module is to control the output of the syntax element when the parsing is done. n_bits represents for the requested bits number to parse the current syntax element; req request the module to read one syntax element; req_type represents for the entropy decoding ways of the current syntax element, we should use CABAC or CALVC to parse the current syntax element depending on the value of req_type; outdata[31:0] stores the entropy decoded data; dec_ready indicates whether the syntax element module is ready to accept external requests; strobe checks the validity of the current output syntax element data.

Core syntax element parsing unit runs the CABAC or CAVLC algorithm. The parsing process includes the generation of syntax model index, model adaptive update, anti-binarization and so on. In AVS, CABAC is mainly used to parse the syntax elements such as macroblock type, luma and chroma prediction mode, transform coefficients and so on; CALVC is mainly used to parse luam and chorma residual data.

The hierarchical architecture of AVS stream includes sequence , image, slice, macroblock and block. During the process of decoding, we need to parse every syntax element from top to bottom and assign each syntax element value to corresponding variables. In the end, we run the AVS 3D decoding algorithm on SoC platform and cooperate with the hardware acceleration module to reconstruct the left and right view images.

The design of AVS 3D decoder on SoC platform

In this paper, we adopt ZYNQ 7020 of Xilinx which integrates a dual-core design in a single device. It has two chips of Cortex-M9, one is used as the main processing system; the other is used to run the core algorithm of AVS 3D decoder. The two chips of M9 share the memory and external peripherals. Aimed at the implementation of AVS 3D decoder, the Master-Slave design pattern should be adopted. In this pattern, Master M9 is used as top-level control unit to complete the external interface communication with the 3D ES stream and the display of the decoded 3D images; Slave M9 and hardware acceleration module work together to implement the AVS 3D decoding algorithm. Master M9 and Slave M9 cooperate to achieve the whole AVS 3D decoder design on SoC platform[7].

The boot of AVS 3D deocoder on ZYNQ 7020

When booting the AVS 3D decoder on SoC platform, we should configure the clock of ZYNQ 7020. Then we should execute the BootROM code on MASTER M9. BootROM is the first procedure to execute on SoC platform. When SLAVE M9 is waiting for the decoding start command, BootROM is already executed on MASTER M9. The boot flow is shown in Figure 5.

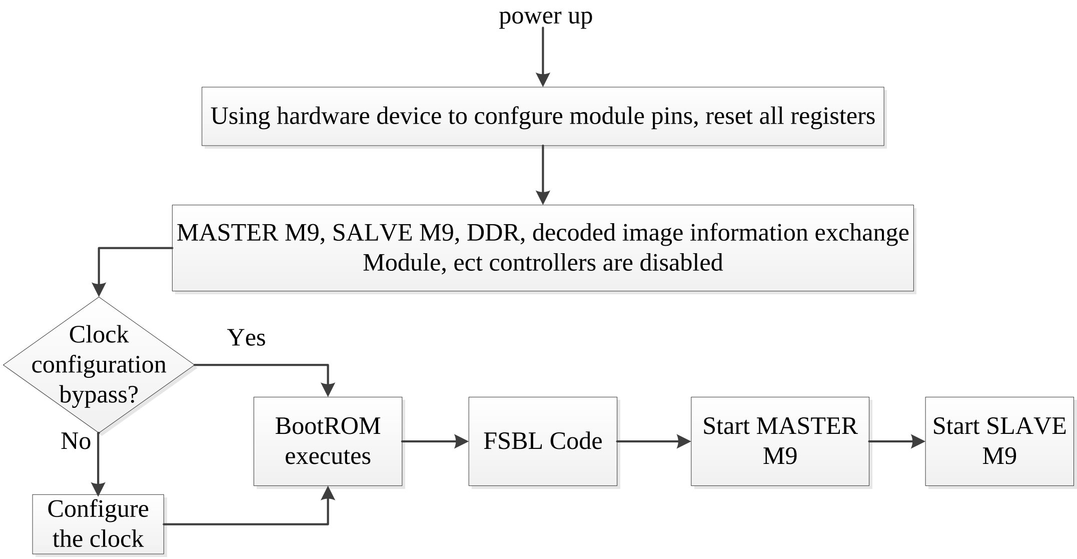
power up
Using hardware device to confgure module pins, reset all registers
MASTER M9, SALVE M9, DDR, decoded image information exchange Module, ect controllers are disabled
Clock configuration bypass?
Yes
No
BootROM executes
FSBL Code
Start MASTER M9
Start SLAVE M9
Configure the clock

Figure 5 The boot process of AVS 3D decoder on SoC system

The main task of BootROM is to configure the UART information of the whole AVS 3D decoder and copy the FBSL (first stage boot loader) from the boot device to the on chip memory of MASTER M9. The FBSL will initialize the Xilinx hardware configuration information of SLAVE M9, and read BootHeader file to decide whether MASTER M9 should be booted or not. When the FBSL is excuted, BootROM will set the SLAVE M9 to event waiting mode. When MASTER M9 was booted, MASTER M9 will inform whether SLAVE M9 should execute the AVS 3D decode procedure or not.

The design of AVS 3D decoder on ZYNQ 7020

AVS 3D decoder of SoC platform is shown in Figure 6. During the interaction of each module, the smaller account of data is transmitted by AXI LITE; the larger account of data such as decoded image is transmitted by AXI VDMA; the data interacted frequently is transmitted by AXI CON.

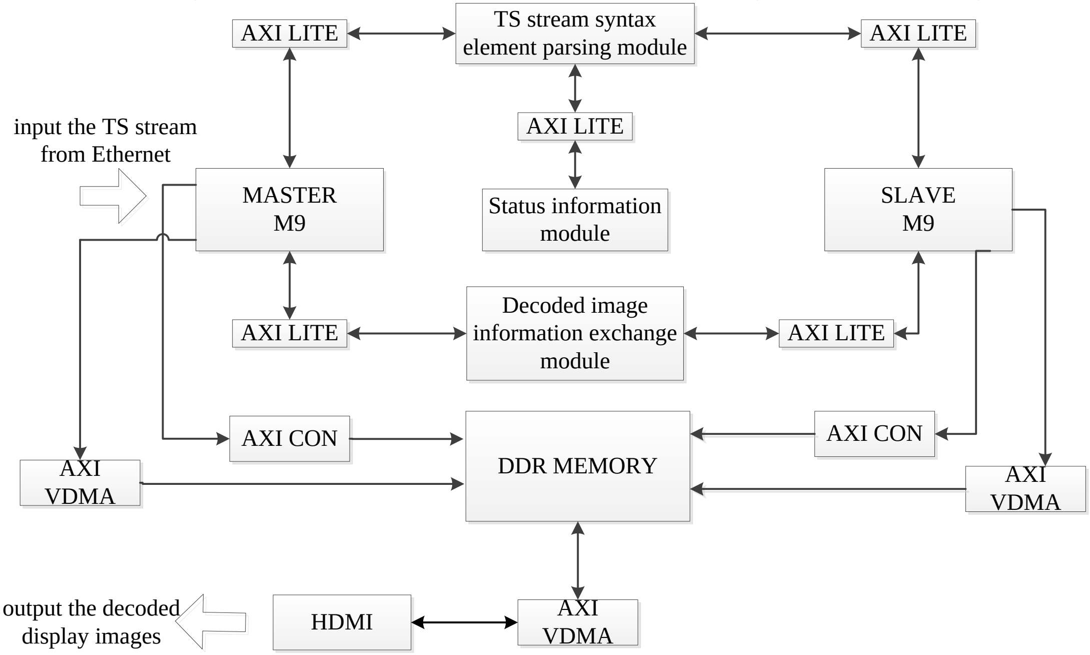
AXI LITE
TS stream syntax element parsing module
AXI LITE
AXI LITE
input the TS stream from Ethernet
MASTER M9
Status information module
SLAVE M9
AXI LITE
Decoded image information exchange module
AXI LITE
AXI CON
AXI CON
AXI VDMA
DDR MEMORY
AXI VDMA
output the decoded display images
HDMI
AXI VDMA

Figure 6 The design of AVS 3D decoder on SoC system

MASTER M9 is the top-level control unit of the entire AVS 3D decoder on SoC platform. The main function of MASTER M9 includes: 1) Getting the 3D TS stream from Ethernet; 2) Exchanging information with TS stream syntax element parsing module and informing SALVE M9 to start AVS 3D decoding procedure. According to the different value of parsing flag, SALVE M9 will call hardware acceleration module to adopt CABAC or CAVLC entropy decoding ways to parse different syntax elements and interact with status information module to return the entropy coding flag, buffer size, parse finish flag and so on to MASTER M9; 3) Managing the address pointers of decoded images and reference frames as well as sending corresponding data to different storage addresses of DDR MEMORY.

SLAVE M9 runs the core algorithm of AVS 3D decoder. The decoding algorithm includes stereo-packing algorithm and traditional 2D decoding algorithm. AVS stereo-packing algorithm is elaborated in Part 2. Traditional 2D decoding algorithm includes: start code checking, sequence and image header reading, entropy decoding, macroblock data obtaining, inverse transform, inverse quantization, intra or inter prediction, 1/4 sub-pixel interpolation, residual reconstruction, loop filtering and so on. We should transplant the C programming language to Xilinx SDK platform and input the parsed syntax element value to every level's decoding functions, so that SLAVE M9 can achieve AVS 3D decoding algorithm. Finally, the left and right view images will be written to DDR MEMORY's different storage address.

Decoded image information exchange module is the information intermediary of MASTER M9 and SLAVE M9. When SLAVE M9 executes the 3D decoding procedure, it will produce three types of image pointers: 1) Reference frame pointer (particularly for I, P frames), it points to the address of prediction frames which will not be displayed immediately; 2) Display frame pointer (particularly for B frames), it points to the address of frames which will be put into the display queue immediately; 3) Writing address pointer. When SLAVE M9 decodes one frame, the decoded information will be written into the pointed address of DDR MEMORY. Using this module, SLAVE M9 transmits the type and value of different image pointers to MASTER M9. MASTER M9 interacts with DDR MEMORY to ensure the correct decoding and display order of images.

DDR MEMORY uses Ping-Pong storage mode. It has two memory blocks, each memory block respectively allocates 5 frames of storage space for stereo-packing images, left view images and right view images. When memory block 1 sends message to HDMI port, memory block 2 receives message form SLAVE M9; when memory block 2 receives message form SLAVE M9, memory block 1 sends message to HDMI port; cycle operation like this, the work efficiency can be improved. MASTER M9 determines the read and write address of each memory block to ensure that DDR MEMORY can send the image data to HDMI port in a correct order.

Experiment result and analysis

In this paper, the AVS 3D decoder design on SoC system is achieved on Xilinx ZYNQ 7020 development board. By adding the viewpoints separation module，inter layer up-sample filter and inter-view prediction to traditional AVS 2D decoder RM52k, AVS 3D decoding algorithm can be achieved. The corresponding code should be transplanted to Xilinx SDK 2014.2 embedded software development platform. The parameter configuration of AVS 3D decoder is shown in Table 1:

Table 1 Parameter configuration of AVS 3D decoder

frame number 100	adaptive QP 1	the proportion of enhancement layer enhancement 2
frame rate 30	decoded frame structure 0	the number of enhancement layer 1
entropy decoding 2	image format 1920×1080	the number of up-sample filter taps 6
QP value 30	stereo packing mode 2	kernel vector of up-sample filter $\{1,-5,20,20,-5,1\}$

AVS 3D ES stream should be packaged into TS stream for network transmission. Based on Xilinx LWIP criterion, the TS stream will be input into MASTER M9 from Ethernet through the modulation of IP QAM. Using hardware acceleration module to parse the syntax element and coordination with SoC system, the image information will be export to 3D television through HDMI port. The stereo-packing images can be obtained in 3D television. By viewpoints separation and inter- layer interpolation, the enhancement layer images of left and right view can be reconstructed. By viewpoints interlacement, the 3D video with depth information can be displayed in 3D television. The 3D video got by AVS 3D decoder is shown in Figure 7. It can prove the validity of the AVS 3D decoder design on FPGA/SoC platform.

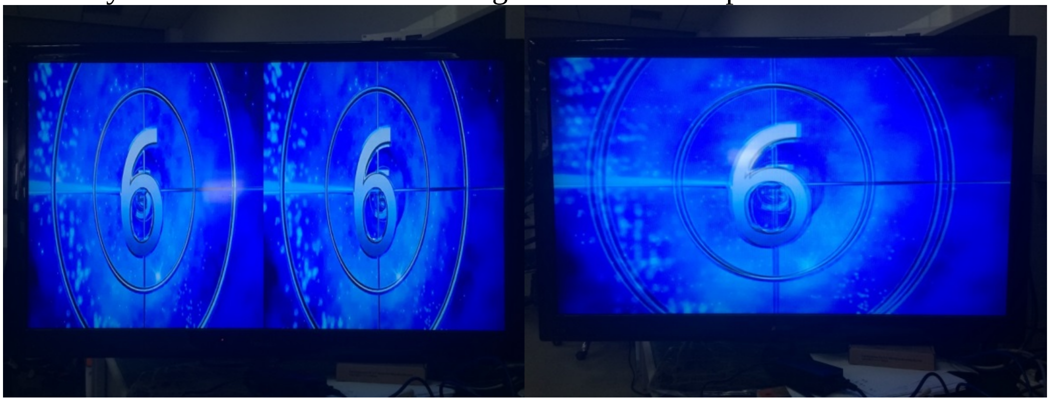

Figure 7 The display effect of AVS 3D real- time decoder

Conclusions

A 3D decoder based on AVS standard is achieved by adding new function modules to the traditional 2D decoder. AVS 3D decoder is innovatively implemented on FPGA/SoC Co-platform with the coordination of hardware acceleration module. The decoded images should be input into 3D display device. Through viewpoints interlacement, we can observe the disparity and depth information of the 3D video which can verify the validity of the AVS 3D real-time decoder design on FPGA/SoC Co-platform.

References

[1] HOU Jin-ting, MA Si-wei, Gao Wen . Overview of AVS Standard[J]. Computer Engineering, 2009,
08:247-249+252.

[2] MA Qian, LI Dong, WANG Qi-fei, ZHANG Yong-bing, JI Xiang-yang, DAI Qiong-hai. Stereoscopic Video
Coding Standard in AVS[J]. Journal of Shanghai University(Natural Science),2013,03:225-228.

[3] WANG Zhang, LIU Jian, YAN Cuo-ping. Inter layer up-sampling filter scheme applied in SVC[J]. Journal of
Communication,2008,04:8-12.

[4] ZHAO Yin. Research on 3D Video Visual Quality and Enhancement Processing[D].Zhe Jiang University,2013.

[5] D. Marpe, H. Schwarz, and T. Wiegand, “Context-based adaptive binary arithmetic coding in the H.264/AVC
video compression standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 620-636,
July 2003.

[6] MythriAlle,JBiswas,S.K.Nandy .“High Performance VLSIArchitecture Design for H.264 CAVLC Decoder” .
ASAP 2006 IEEE.

[7] WANG Zhong-ping, Design and Research of Unified SoC Architecture of H.264 and AVS Dual-standard
Decoder[D].Shanghai Jiao Tong University,2008.