



DSP Implementation and Optimization of Pseudo Analog Video Transmission Algorithm

Chengcheng Wang, Pengfei Xia^(✉), Haoqi Ren, Jun Wu,
and Zhifeng Zhang

College of Electronics and Information Engineering, Tongji University,
Shanghai, China

1631731@tongji.edu.cn, pengfei.xia@gmail.com

Abstract. With the development of wireless video technology and embedded technology, a dedicated digital signal processor (DSP) can achieve the video transmission stably and flexibly. Some existing wireless video transmission algorithms do not perform well in response to complex channel environments. A pseudo-analog video algorithm that can be run in a dedicated instruction set was proposed. At the transmitter, the image data which are removed spatially redundant are divided into L-shaped blocks for power allocation, and the digital signal are sent to CRC and Turbo coding. Finally, the modulated digital signal and the pseudo-analog data after power allocation are sent to framing. The receiver includes channel estimation and de-framing, recovers digital signal and pseudo-analog signal through error detection and decoding. We have optimized the algorithm at the assembly level, so that the entire system is more flexible. The entire transfer system will run on the FPGA and hardware DSP boards for debugging.

Keywords: DSP · DMA · Dedicated instruction set · Power allocation
Turbo encoding · Framing

1 Introduction

With the rapid development of wireless communication technologies, various smart devices have rapidly become popular. Emerging application platforms such as drones and smart wearable devices have emerged, which have increased the demand for various applications of wireless video transmission. The current video transmission scheme can not match the video quality with the channel quality [1], thus it is not optimal from the perspective of network information theory [2]. Someone designed a set of pseudo-analog video transmission system, which is based on SoftCast [3, 4]. SoftCast transmits video in a way of linear transformation, so that guarantees the linear relationship between video signal and image pixels. This relationship will not be changed by the noise or interference [2]. SoftCast is a video multicast method based on real-number transmission, which greatly reduces the redundancy of video frames. Pseudo-analog system has a low delay and eliminates the cliff effect in a certain extent [5], and provides different video quality for different users in different channel environments.

1.1 Structure of Transmitter and a Receiver

The entire implementation of Softcast includes a transmitter and a receiver. As shown in Fig. 1, the transmitter’s data are sent to the encoder through the HDMI interface. Encoder executes discrete cosine transform (DCT) on the data firstly. The DCT transformation can remove the intra correlations of pixel values [6]. Then, the power allocation module calculates the power allocation factor. Power allocation also sends the average power of each transform domain as a digital signal to encoding and modulation. In the digital signal part, CRC and Turbo coding are involved to improve the system’s error correction capability. In framing module, in order to combat frequency-selective fading in digital part code blocks, the modulated digital signal and the pseudo-analog signal after power allocation will be interleaved. The synchronization data and pilot data are inserted into the OFDM symbols in the radio frame.

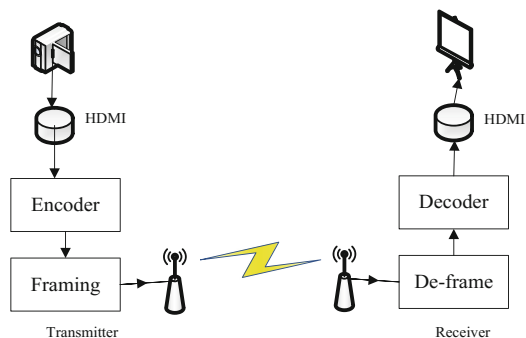


Fig. 1. The pseudo analog video transmission system

1.2 Advantages of DSP Implementation

If each module is implemented in hardware, the entire system would have a much lower flexibility. Although the DSP algorithm is more complex and less efficient, it is more flexible to implement the scheme by DSP programming. An assembly-based power allocation, CRC and Turbo algorithm are designed. Due to the uniqueness of the frame format, a framing algorithm which is multiple look-up tables with linear time complexity is designed. DSP will process the channel estimation section, the de-framing section, and the linear least-squares estimation (LLSE) in the receiver. Digital signals and pseudo-analog signals can be separated by de-framing algorithms. The digital signal can be demodulated based on the result of the channel estimation.

1.3 Dedicated DSP Features

This pseudo-analog video algorithm was ported to a dedicated digital signal processor. A digital signal processor (DSP) is a specialized microprocessor, which is Harvard structure [7]. A specific instruction set is designed for this DSP processor. To improve

its performance, single instruction multiple data (SIMD) [8] and very long instruction word (VLIW) [9, 10] are widely used in DSP design. DSP has huge capacity in data processing, as well as powerful memory access(DMA) channels. By configuring the corresponding source address register, target address register, and transmission counter to access data. CPU core and the DMA may operate in parallel. Therefore, the efficiency of the entire computing system is greatly improved.

The rest of this paper is organized as follows: Sect. 2, introduce the structure of the pseudo-analog transmission system. Section 3, discuss the DSP implementation of pseudo-analog algorithm. Section 4, analyze the DSP performance. Section 5, draw the conclusion.

2 Architecture Design

As shown in Fig. 2, the encoder includes DCT, power allocation, CRC, Turbo encoding and modulation. They are implemented by DSP except DCT.

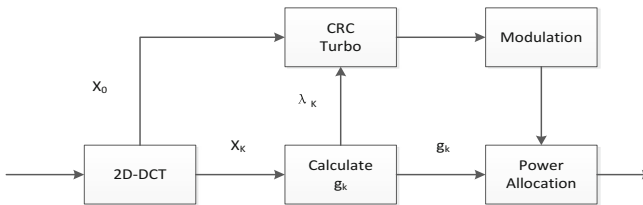


Fig. 2. Encoder

2.1 Power Allocation Algorithm

Firstly, the image data will be converted by 2D-DCT. The DCT transform can eliminate the spatial redundancy of the image data, and can transform the image from the spatial domain to the frequency domain without damaging the image information. The DCT result is a number of matrices ($32*32$). According to the characteristics of the DCT operation, the matrix stores the DC component and the low frequency coefficient which record most of the information of the image in the top left corner. So the first data stored in each row of each matrix is sent to the Turbo encoding. Turbo encoding has strong anti-interference and anti-fading capabilities. CRC is added before Turbo encoding to ensure error checking at the receiver.

As shown in Fig. 3, during the process of power allocation, the data matrix is divided into L-shaped blocks. There are 15 L-shaped blocks of each matrix. Each L-shaped block corresponding to the corresponding average power λ_k ($k = 1, 2, \dots, 15$), λ_k is the weighted sum of the squares of the corresponding block data. 30 matrices are divided into groups and obtain the average power $\bar{\lambda}_k$. This can ensure that the average power obtained for different coefficients has universality and stability.

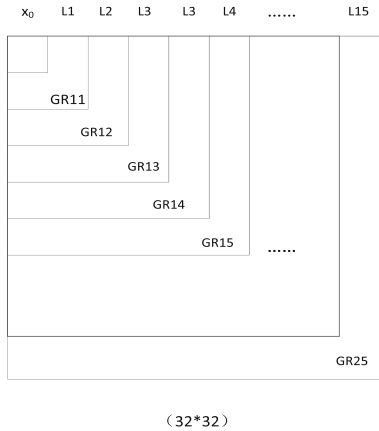


Fig. 3. L-shaped block

The formula for λ_k is as follows:

$$\lambda_k = \frac{\sum_{x \in L_k} x^2}{N} \tag{1}$$

L_k is the data set of the L-shaped blocks, N is the number of the data set L_k . After calculating the corresponding average power for each matrix, the average power $\bar{\lambda}_i$ of the corresponding blocks of the 30 matrices should be calculated, According to the average power $\bar{\lambda}_i$, the power allocation factor g_k can be calculated by the following formula:

$$g_k = \bar{\lambda}_k^{-1/4} \sqrt{\frac{P}{\sum_{k=1}^{15} \sqrt{\bar{\lambda}_k}}} \tag{2}$$

P is the total power [3]. Then we can get the pseudo-analog data Y_K to be transmitted, $Y_K = X_k * g_k$.

2.2 Turbo Encoding and Framing

As already mentioned above, the first data of each matrix is sent to the CRC and Turbo encoding as a digital signal, actually average power is added to it too. Then, the coded digital part code block is subjected to 16QAM constellation point modulation according to the normalized energy. The modulated digital data and the pseudo-analog data are framing together. This algorithm proposes a set of schemes that can be compatible with the traditional physical layer. The real-number signal generated by the encoder on the transmitter does not perform traditional error correction, but directly maps the data to OFDM symbols. A radio frame contains 32 OFDM symbols with 2048 subcarriers per OFDM symbol, each subcarrier can carry 4 bytes of data.

Some OFDM symbols contain pilot data and synchronization data. We use synchronization data to perform frame synchronization on the received frame data. The pilot data are used for channel estimation at the receiver to help restore the digital signal data.

As shown in Fig. 4, the decoder includes channel estimation, de-framing, LLSE, demodulation, Turbo decoding, IDCT. Considering the complexity of related modules, DSP only processes channel estimation, de-framing, LLSE.

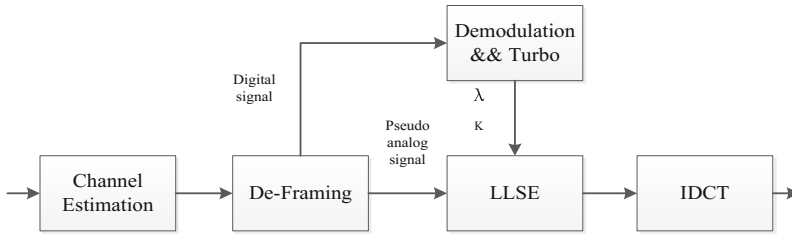


Fig. 4. Decoder

2.3 Channel Estimation

The receiver sends the received frame data to the channel estimation module, and the channel estimation is divided into two parts: one is the least square (LS) channel estimation which estimates the pilot positions only. The advantage of the LS is that the calculation is small and the structure is simple. However, when the channel noise is large, its accuracy will be reduced. Another is minimum mean square error (MMSE) which estimates the position of other subcarriers outside the pilot.

2.4 LLSE Algorithm

The digital signal is used for soft demodulation and Turbo decoding, both are achieved by hardware. Linear least squares estimation (LLSE) is executed on the pseudo-analog data. The average power is got based on the result of the decoding of the digital part. According to the power allocation algorithm, the data received can be described like this: $Y = g_k * X + \text{sigma}$. The LLSE calculation formula as follows:

$$X_{llse} = \frac{\bar{\lambda}_k * g_k}{\bar{\lambda}_k * g_k * \bar{\lambda}_k + \text{sigma}^2} * Y \tag{3}$$

Y is the data received by the receiver; g_k is the corresponding the power allocation factor; $\bar{\lambda}_k$ is the average power; sigma is the Gaussian white noise signal matrix.

3 DSP Implementation and Optimization

3.1 Swift DSP

A digital signal processor named Swift [11] is designed for the wireless communications. It has a high-performance 32-bit fixed-point arithmetic unit, a high-performance 160-bit SIMD unit, and supports four 40-bit, eight 20-bit, or 16 10-bit vector operations. The instruction set includes 11 control instructions, 19 data transfer instructions, 54 load/store instructions, 49 scalar operation instructions, and 60 vector operation instructions. At the same time, there is a powerful DMA module to transfer data from external storage to RAM.

3.2 Power Allocation and LLSE Implementation

Firstly, the DMA transfers the result of the DCT operation to the RAM and hands it to the DSP for power allocation. The problem that power allocation ported on the DSP platform needs to solve is its unique L-shaped data block. Since the result of the DCT output is a 32*32 matrix data block stored in a row, the DSP needs to read the data which are not sequential storage when calculating the average power. We have designed an algorithm that uses scalar instructions to configure the scalar load's addressing address in advance. This algorithm needs to configure the memory address for each instruction, resulting in a huge amount of code, and a lot of instruction cycles. The number of cycles is roughly 4 times that of the algorithm implemented by the later optimized algorithm. So an algorithm that can use the vector instruction sequential addressing dislocation addition to obtain the power sum is proposed. As shown in Fig. 3, general registers are allocated for each L data block to hold the current power sum. The vector instructions are used to read data by row. The vector instructions need to be completed in multiple cycles. The data processing of the upper and lower rows are made into the instruction pipeline and are relatively parallel.

LLSE is similar to the power allocation. The power allocation factor g_k should be calculated firstly. Because of the result of decoder, the average power $\bar{\lambda}_i$ can be obtained directly. According to the formula (2), then the power allocation factor g_k can be calculated easily.

3.3 Turbo Encoding and Channel Estimation

Turbo encoding in high-level language can be achieved through the finite state machine. But it is very complicated to carry on the conditional judgment in DSP. So as shown in Table 1, a Turbo encoding algorithm that sets general register as flag register to realize convolution is designed. The running time of the algorithm has been tested far less than the running time of hardware acceleration.

The LS channel estimation is mainly the multiplication of the second-order complex matrix. Each second-order complex matrix was respectively sent to the specified memory area through DSP. As shown in Fig. 5, After the hardware module calculation is completed, the result is stored into the sequential address space. The basic operation of MMSE channel estimation is similar to LS.

Table 1. Turbo

Algorithm Turbo Convolutional coding 32-bit register

```

1: Load 32-bit data to GR1
2: State{GR2, GR3, GR4};
3: For i in range(31)
4:   GR5 = GR1 && 0x01 , then GR1 >>1
5:   GR6 = GR5^GR2, GR7 = GR5^GR3
6:   GR6 = GR6^GR3, GR7 = GR7^GR4
7:   Store GR6 as the result of the convolution
8:   GR4 = GR3, GR3 = GR2, GR2 = GR7
9: end For

```

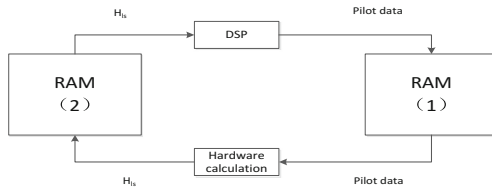


Fig. 5. Channel estimation

3.4 Framing and De-Framing

A single physical frame contains 32 OFDM symbols. One OFDM symbol contains 2048 subcarriers, 1296 effective subcarriers. Every 4 OFDM symbols have 2 OFDM symbols storing pilots. The second and third OFDM symbols store synchronization data. Because the location of the pilot data and synchronization data are fixed, these data are stored into RAM before the project is run. Framing and De-framing need to store or read valid data into the specified RAM space. Firstly, all symbols are divided into two kinds of symbols according to the synchronization pilot data OFDM and normal OFDM symbols. The former is considered to be a scalar symbol and the latter is a vector symbol. The difference between the two symbols is that the framing target address of the scalar symbol is not in order. After data are stored into the target address, scalar symbol needs look up the table to calculate the next target address. The target address of the vector symbol is in order. After the target data stored, the target address can add a fixed number to get the next storage target address.

Since the framing and de-framing algorithms are similar, the framing algorithm will be introduced. In the beginning, a large number of judgment statements are added to distinguish different symbols and memory addresses. The result of this is that the

execution cycle is too long. Therefore, an algorithm using multiple lookup tables to obtain the target address of the framing is designed. The precondition of the algorithm is that the source data address is in order. After each fetch, the address can be obtained by adding a fixed number. As shown in Fig. 6, the target address was divided into 32 memory regions corresponding to 32 OFDM symbols. The rules of address changes of these 32 memory regions are set into 32 tables that are stored sequentially. The flag in Fig. 6A is set to let the program distinguish between two symbols. The framing algorithm execution parameters for different symbols are different.

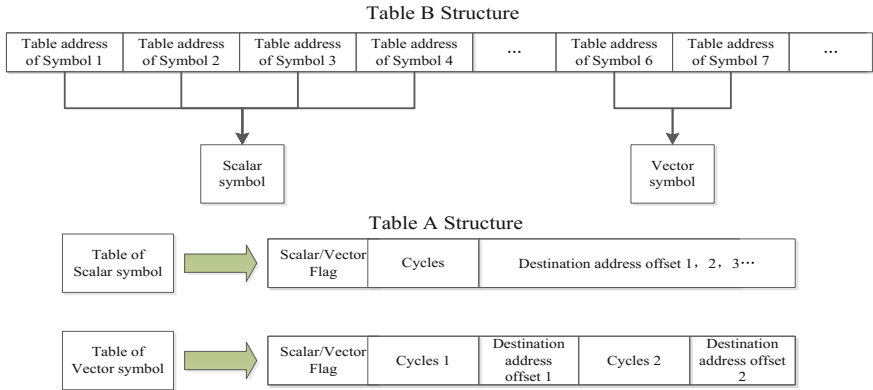


Fig. 6. Table structure

The Table A can configure the framing algorithm for the number of cycles and the target address offset. As shown in Fig. 6B, Table B is the address index of Table A. As flow chart is shown in Fig. 7, obtain the address of the current framing symbol parameter table through Table B. After obtaining the address of Table A, reading the contents of Table A to configure the framing parameters. Transfer data and complete the data loading of the symbol. After 32 cycles, the framing of one radio frame is completed.

4 Performance Analysis

The implementation of the pseudo-analog video transmission algorithm is based on the DSP and the corresponding hardware accelerators. There is a hardware system that contains a DSP. The DSP performance parameters are shown in Table 2. The real video data are used to test the entire system. The resolution of the video is 960*640, Y : U : V is 4:2:0. The video data are sent to the FPGA(Xilinx’s Virtex – 7V x 475T) board through the HDMI interface. The DSP is connected to the FPGA board through the FMC interface. Furthermore, there is a hardware monitoring module which is used for calculating the operating cycles. Besides this, there is a fully hardware designed Softcast system based on the same FPGA for comparison.

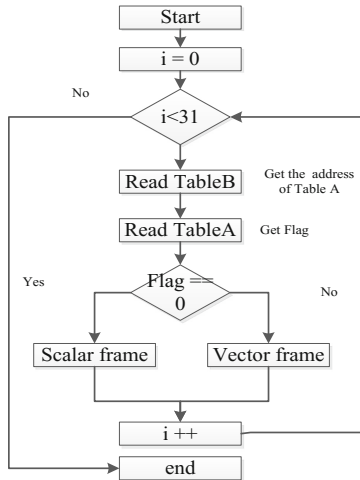


Fig. 7. Framing algorithm

Table 2. DSP performance parameters

Frequency	250 MHz
Power Consumption	150 mW
Instruction memory	256 KB
Data Memory	1 MB

As shown in Table 3, the number of receiver LLSE cycles is less than the number of transmitter PA cycles, because the average power at the receiver can be obtained according to the digital data. Framing and de-framing take more cycles because DSP involves a lot of data transferring. In fact, DMA is not completely parallel to DSP, DMA also consumes cycles.

Table 3. The average number of cycles for each module

Transmitter		Receiver	
Modules	Cycles	Modules	Cycles
PA	1,584,000	LLSE	1,068,000
Turbo	716,000	Channel estimation	721,000
Framing	2,542,000	De-framing	2,244,000
Modulation	520,000	–	–
DMA	64,1000	DMA	64,2000
Total	6,003,000	Total	4,675,000

According to the above table data, when the DSP clock frequency is 200 MHz, the video can reach 30FPS, while the fully hardware designed system performs 30 FPS at 150 MHz. However, the DSP based design has more flexibility, that the algorithm of encoder and decoder can be easily modified as applications demand, and the frame structure can be easily changed as well. It is maintainable and can be optimized. The DSP can also run at higher frequencies with better silicon process, to further improves the processing video resolution.

5 Conclusion

This paper focuses on the design and the DSP implementation of pseudo-analog video transmission algorithm. The video transmission efficiency can be improved through optimizing the assembly algorithm later. Increasing the frequency of the DSP can run the higher resolution video. In the future research, we will continue to improve the system efficiency and optimize DSP instruction set, making the system more stable and efficient.

Acknowledgment. This work was supported in part by the National Natural Science Foundation of China (Nos.61631017 and 61502341); National Science and Technology Support Plan (Grant no. 2012BAH15F03).

References

1. Fan, X., Wu, F., Zhao, D.: D-cast: Dsc based soft mobile video broadcast. In: ACM International Conference on Mobile and Ubiquitous Multimedia (2011)
2. Ding, Z., Wu, J., Yu, W., Han, Y.Q., Chen, X.H.: Pseudo analog video transmission based on LTE physical layer. In: IEEE/CIC International Conference on Communications in China (ICCC) (2016)
3. Jakubczak, S., Katabi, D.: Softcast:one-size-fits-all wireless video. In: ACM SIGCOMM 2010 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (2010)
4. Zheng, S., Antonini, M., Cagnazzo, M., Guerrieri, L.: Softcast with per-carrier powerconstrained channels. In: IEEE International Conference on Image Processing (2016)
5. Jiang, J., Xia, P.F., Wu, J., Chen, S., Zhang, B.Y.: Pseudo-analog wireless stereo video transmission in hardware acceleration. In: 2017 9th International Conference on Wireless Communications and Signal Processing (2017)
6. Borkar, S., Chien, A.: The future of microprocessors: communications of the ACM, **54**(67–77) (2011)
7. Zhao, C.X., Wu, J., Chen, X.: Design and implementation of a memory architecture in DSP for wireless communication. In: 2015 10th International Conference on Communications and Networking in China (2015)
8. Derby, J.H., Moreno, J.: A high-performance embedded DSP core with novel SIMD features. In: Acoustics, Speech, and Signal Processing, pp. 301–304 (2003)

9. Anderson, T., Bui, D., et al.: A 1.5 GHz VLIW DSP CPU with integrated floating point and fixed point instructions in 40 nm CMOS. In: 2011 20th IEEE Symposium on Computer Arithmetic (ARITH), pp. 82–86 (2011)
10. Fridman, J., Greenfield, Z.: The TigerSHARC DSP Architecture. *IEEE Micro* **20**, 66–76 (2000)
11. Ren, H.Q., Zhang, Z.F., Wu, J.: SWIFT: A computationally-intensive DSP architecture for communication applications. *Mob. Netw. Appl.* **21**(6), 974–982 (2016)