Optimization and Implementation of H.264 Encoder Based on DSP Platform

He Yikui*
Sichuan Aerospace Vocational College, Chengdu Sichuan, 610100, China
*Corresponding author: yangliu_astri@163.com

Abstract. Aiming at the problem of low precision and long running time of traditional H.264 encoder algorithm, this paper proposes two optimization methods. One is to propose a method suitable for DSP to carry the reference frame, which can reduce the waiting time of DSP operation unit. By using DMA to read the data stored in the external access device in advance, all the data involved in the operation are stored in the memory of DSP chip. This improves the operation efficiency of DSP. Second, the macroblock based loop filtering makes the access times of DSP external memory minimum, and enhances the DSP's ability of processing data. The test results show that the average decoding speed of the decoder is increased by 60% by the two optimization methods. The test system can decode CIF image at 526 MHz, and the speed can reach 30 frames per second.

Keywords: H.264, Data fusion, DSP, Macroblock based loop filtering

1. Introduction
H. 264 is the latest video coding standard in the world, which has high coding efficiency and can meet various application requirements [1]. There are many schemes for video decoding in embedded devices. The hardware decoder is fast and low cost, but it can't meet the improvement of coding standard by upgrading the hardware decoder. The H.264 decoder based on DSP (digital signal processor) can deal with the improvement and change of coding standard by firmware upgrade, which is a popular scheme of embedded device video decoder. However, the computation and storage capacity of H. 264 decoder are greatly improved, and the decoding complexity is about twice that of MPE-4 SP [2-3]. The H. 264 decoder based on DSP still has difficulties. Therefore, it is of great theoretical and practical significance to study the optimal design method of H. 264 decoder based on DSP [4-7]. In this paper, two structural optimization design methods are proposed to improve the decoding speed and complete the real-time decoding.

2. BF533 structural frame
Blackfin533 is a high-performance general fixed-point DSP developed by ADI company. It can be used in image, audio and video embedded devices [8].

Figure 1 shows the internal structure of Blackfin533. BF533 is a multimedia digital signal processor with improved Harvard architecture and reduced instruction set. BF533 core has dual Mac, single instruction multiple data (SMD) processing capability, and has DMA controller inside, so data transfer...
between peripheral devices and memory or between memory and memory can be realized without DSP kernel interference.

Figure 1 BF533 DSP structure frame diagram

3. H. 264 decoding optimization

3.1. Decoding process optimization
The flow chart of H. 264 decoder based on PC is shown in Fig. 2 (a). For each macroblock, the residual data is read for inverse DCT (IDCT), and then motion compensation is performed. Then the reconstruction data of the macroblock is obtained by adding the two. When the motion compensation is carried out, the reference frame data in the out of chip memory (SDRAM) needs to be read. At the same time, the motion prediction results are directly written to the current reconstruction frame in the off chip memory. Finally, the IDCT results are superimposed on the motion prediction data to complete the macroblock reconstruction. After the reconstruction of all macroblocks in a frame, the loop filtering is performed. As mentioned above, to complete the decoding of a macroblock, it needs to read and write the off chip memory many times. However, the access speed of SDRAM is far lower than that of CPU. The CPU operation unit is in the state of waiting for data for a long time, which greatly reduces the operation efficiency of the operation unit. Modern PC has a high speed, and there are a lot of multi-level cache in the chip. The running speed of H.264 decoder based on PC can meet the real-time requirements. However, due to the low frequency of DSP and the shortage of on-chip resources, real-time decoding can only be realized by optimizing the hardware characteristics of DSP.

Based on the above considerations, the design of H. 264 decoder suitable for DSP is as follows: allocate data space in SRAM and reduce the waiting time of reading data in DSP CORE. The residual data is transformed by IDCT, zigzag permutation and stored in IDCT BUF. REF BUF stores the reference frame data for motion compensation with the size of 32 × 32 bytes. Ping pong mechanism is used to complete the parallel operation, REC PNG and REC PONG are used to place the current macro block reconstruction data. The decoding process is shown in Fig. 2 (b). Firstly, all motion vectors of a macroblock are decoded, and then all the data positions required by the current macroblock for motion compensation can be calculated. The data can be transported to the processor's on-chip high-speed SRAM in advance, and the processor does IDCT during this process. In this way, in DSP system, DMA can be used to carry the data of reference frame. DMA and core work at the same time. When the processor completes IDCT, DMA also completes the data handling of reference frame, and then carries out motion compensation, the data will be in the on-chip high-speed SRAM. The processor will not be in a pause state because of waiting for the arrival of data, and the running efficiency of the processor will be greatly improved. At the same time, in order to minimize the access of core to SDRAM, the loop filter based on the whole frame image is improved to the filter based on macroblock. Finally, the filtered data is transferred to SDRAM through DMA.
3.2. Design method of reference window

H.264 There are seven modes of motion compensation in 264 standard. A macroblock can carry up to 16 motion vectors, that is, each 4 × 4 block can carry one motion vector. How to efficiently transport the necessary reference frame data to the on-chip memory becomes the key technology of decoding process design. It is mentioned in the literature that 4 × 4 is taken as the basic unit to carry data in motion compensation. This method is suitable for implementation on hardware platform and can give full play to the advantages of hardware parallel execution. However, this method is not suitable to be implemented on DSP platform. One reason is that this method will carry repeated data. For example, when the macroblock mode is 16 × 16, 441 bytes (21 × 21) are needed to be handled. However, this method can handle up to 1,296 bytes (16 × 9 × 9), which is nearly three times of the data needed in practice; The second reason is that this method has a lot of extra overhead. When DMA based on descriptor is used to realize linked list transmission, it takes at least one system cycle for DMA controller to acquire a 16 bit descriptor data. For the reference frame data of 4 × 4 sub macroblocks, it takes more time to obtain the descriptor than to transport the actual data. According to the specific block mode, each sub macroblock is used as the unit for handling, which can effectively reduce the handling of duplicate data. However, the main disadvantage of this method is that when a macroblock is transported in small blocks (for example, a macroblock is divided into 16 4×4 blocks), it will lead to frequent page shortage, which causes the SDRAM controller to close the current page and reopen a new page. This process usually takes 20 ~ 50 clock cycles. This method makes the DSP arithmetic unit wait for data for a long time, which greatly reduces the operation efficiency of the DSP core.

The handling method designed in this paper can give full play to the advantages of DMA and cache to maximize the handling efficiency. The specific design process of the reference window is described in detail below, as shown in Figure 3. For each block, four integer parameters (x_left, y_top, x_right, y_bottom) are used to describe the position of the rectangle window relative to the starting point of the current decoding macroblock. When all motion vectors of a macroblock are decoded, the minimum value XMN is found from the x_left of all blocks, the maximum value X_MAX is found from the x_right of all blocks, the minimum value Y_MN is found from the y_top of all blocks, and the maximum value Y_MAX is found from all y_bottom. The rectangle formed by (XMN, ymn, X_MAX, Y_MAX) is the reference window to be found. In other words, the reference window is the smallest rectangular window that can cover all the data needed for motion compensation of the current block. When the case shown in Fig. 4 occurs, the macroblock has 10 motion vectors, and the sub macroblocks are scattered. In this case, if all the data in the reference window are transported to the on-chip memory
L1, the traffic on the SDRAM bus will increase dramatically and the bus congestion may be caused. To solve this problem, we should set the area of cache block in L1. When the calculated area of reference window is larger than the area of cache block, we do not use DMA to transport the reference frame data, but use core to access SDRAM, that is to say, CACHE is involved. In this way, the advantages of CACHE and DMA are given full play, and the handling efficiency reaches the maximum.

3.3. Macroblock based loop filtering
In order to design macroblock based loop filtering, the reconstructed data structures REC PNG and REC PONG at L1 are shown in Fig. 5 (a). The top four rows and the left most four columns are designed for intra prediction and filtering, and the four columns on the right are designed for intra prediction. The Left Buffer stores the rightmost row of pixels before the current macroblock filtering, which is prepared for intra prediction of adjacent right macroblocks. The Top Buffer stores the bottom row of pixels in the current macroblock to prepare for intra prediction of the macroblock under the current macroblock. Lines Buf is to save the four lines of data that may change when filtering the bottom four lines of data of the current macroblock. The loop filtering process based on macroblock is as follows:

1) The rightmost column of pixels in the current reconstructed macroblock is stored in the Left Buffer for intra prediction of the possible next macroblock, as shown in Fig. 5 (b).

2) The bottom row of pixels in the current reconstructed macroblock is stored in the Top Buffer for intra prediction of the possible next macroblock, as shown in Fig. 5 (c).

3) Take out the four rightmost columns (20 pixels high) from the previous reconstructed
macroblock and place them on the far left of the current macroblock, as shown in Fig. 5 (d).

4) Take the bottom four rows of pixels from the top of the current macroblock from the line buf and store them in the corresponding position, as shown in Fig. 5 (e).

5) The current macroblock is loop filtered.

6) The M × N pixels that have been filtered are transported to the corresponding position of the current reconstructed frame through DMA. If the current macroblock is on the far right of the macroblock line, then M is 20 and other M is 16; if the current macroblock is the bottom macroblock line of the whole frame, N is 20 and other M is 16, as shown in Fig. 5 (f).

7) Store the bottom four rows of data (16 pixels wide) in the current REC PNG or REC PONG into the line buf, as shown in Figure 5 (g).

At the same time, there may be two DMA channels carrying data at the same time. At the same time, the DSP core is processing the data. Moreover, the macroblock based loop filtering reduces the CORE's behavior of accessing the off chip memory to the greatest extent, which can improve the efficiency of the core.
4. Experiment
After the above optimization, the speed of the decoder has been greatly improved. Table 1 shows the experimental data and analysis. The test code stream is generated by JM 9.5 encoding software. The basic file is used to encode 150 frames of CIF image. The I frame period is 25 and the code rate is 500 kbps. The DSP frequency is set to CCLK/SCL K = 526 MHz/131.5 MHz, and the platform is Blackfin533 EZ-K IT LITE REV 1.6.

|                        | akiyo | Vectra | stefan | mobile | tempete |
|------------------------|-------|--------|--------|--------|---------|
| Before optimization / (frame · s⁻¹) | 29.8  | 17.8   | 19.0   | 16.2   | 16.8    |
| Optimized / (frame · s⁻¹)    | 39.7  | 26.5   | 30.1   | 27.9   | 29.8    |
| Increase                 | 33.2% | 48.9%  | 58.4%  | 72.2%  | 77.4%   |

5. Conclusion
Two optimization techniques are used to improve the decoding speed of H. 264 decoder based on Blackfin533. One is to propose a method suitable for DSP to carry reference frame, which reduces the waiting time of DSP computing unit for data. By using DMA to read the data stored in the external access device in advance, all the data involved in the operation are stored in the DSP chip memory, which improves the operation efficiency of DSP. Second, the macroblock based loop filtering makes the access times of DSP external memory minimum, and enhances the DSP's ability of processing data. The test results show that the average speed is increased by 60% after optimization, and the test
sequence speed of decoding CIF format reaches 30 frames s (526 MHz), which fully meets the requirements of real-time decoding.

References
[1] Zhu, & Jia. . Midi and music teaching in colleges of multimedia system application. Key Engineering Materials, vol.1, no. 474-476, pp. 1926-1930, 2011.
[2] Kim, JiEon. "A Study on the Application of 3D Digital Technology to Producing Cyber Fashion Gallery." Molecular Neurobiology, vol.57, no.2, pp. 1-17, 2007.
[3] Yoshikawa, N, et al. "Design and component test of a tiny processor based on the SFQ technology." IEEE Transactions on Applied Superconductivity, vol.13, no.2, pp. 441-445, 2003.
[4] Tong, Rongbai, et al. "Recent research progress in the synthesis and properties of burning rate catalysts based on ferrocene-containing polymers and derivatives." Journal of Organometallic Chemistry, vol.755, no.2, pp. 16-32, 2014.
[5] Shimada, Izumi , and J. F. Merkel . "Copper-Alloy Metallurgy in Ancient Peru." entific American, vol.265, no.1, pp. 80-861991.
[6] Luger A.. "Advances in Blue OLEDs Increase Efficiency and Lifetime." Critical Care Medicine, vol.14, no.5, pp. 458-461, 1986.
[7] Segal, M. M. , et al. "Evidence-based decision support for neurological diagnosis reduces errors and unnecessary workup." Journal of Child Neurology, vol.29, no.4, pp. 487-92, 2014.
[8] Lindqvist, Olav, and C. Tishelman . "Room for Death – International museum-visitors' preferences regarding the end of their life." Social Science & Medicine, vol.139, no.2, pp. 1-8, 2015.