A Single Scalar DSP based Programmable H.264 Decoder

Di Wu, Tiejun Hu
Dept. of Electrical Engineering
Linköping University
Linköping, Sweden, 581 83
Email: (diwu004,tiehu911)@student.liu.se

Dake Liu
Dept. of Electrical Engineering
Linköping University
Linköping, Sweden, 581 83
Email: dake@isy.liu.se
Telephone: +46(13)281256

Abstract— **This paper presents a feasibility study of applying a legacy reconfigurable single scalar DSP processor to media processing. The latest video compression standard - H.264 was adopted as target application. First, a pure software H.264 decoder was implemented based on the legacy DSP processor. Although real-time performance was not achieved, it exposed essential computational as well memory access costs. The second design explored the possibility of achieving both programmability and real-time performance with affordable hardware acceleration. At the same time, features of H.264 processing were analyzed for more suitable accelerator design. In order to avoid conflicts in memory access, tasks scheduling was carefully considered. In the end, performance of proposed solution and the bottlenecks which still limit the performance are presented as conclusion.**

I. Introduction

Single scalar DSP processors have been widely used in mobile handsets for voice codec and other applications. Because of the emerging era of streaming media, more and more mobile handsets need to support streaming media such as MPEG-4 etc, which requires much higher processing performance than earlier applications. One way to solve this problem is to use new powerful media processors which are mainly SIMD and VLIW. However this solution requires redesign of the whole hardware platform which costs a lot and is not affordable to some HW vendors. Thus for them, the only way is to explore the possibility that single scalar processor could be used in media processing with affordable extension. If it were practical, not only time to market could be saved but also investment in legacy IP design could be protected. Since few single scalar DSP processors were originally designed for media processing, investigation needs to be carried out in order to explore the potential of applying single scalar DSP processors to media processing.

H.264 (MPEG-4 part 10) is the latest video compression standard especially suitable for wireless network. In wireless communications, bandwidth is too expensive to waste. For most of the mobile users, bandwidth for steaming video will not go beyond 128kbps. Among current video compression standards, only H.264 can provide high quality at such low bit rate, because compared with former video coding standards such as MPEG-4 part 2, it saves up to 40% in bit rate and provides error resilience features. Thus we adopted H.264 decoding as our target application. However, the improvement in performance also introduces increase in computational complexity, which requires more powerful hardware. At the same time, there are several image and video coding standards currently used such as JPEG and MPEG-4. Although ASIC design can meet real-time performance, it is not programmable which lacks flexibility for heterogeneous media standards. Hence programmable DSP processor is believed by many people to be the trend since it provides both real-time performance and flexibility.

In this paper, a method to promote the performance of single scalar DSP with programmable accelerator is proposed. And the bottleneck for performance promotion is investigated and the upper limit of acceleration of a certain single scalar DSP for H.264 decoding is presented.

II. Software Implementation and Analysis

A. New Features of H.264

1) Entropy Decoding: CAVLC is context adaptive variable length coding based on 4×4 blocks. It is combination of both entropy coding and run length coding.

2) Inverse Integer Transform: In H.264, IDCT has been replaced by inverse integer transform (IIT) which only involves shift and addition without multiplication. Thus this part is not as computational intensive as in former standards while it provides more accuracy without drift caused by approximation in transform.

3) Quarter Pixel Interpolation and Variable Block Size Motion Compensation: Most of the existing standards support the motion estimation accuracy up to 1/2 pixel. In H.264, the maximum accuracy is 1/4 pixel. In former standards, motion estimation is based on 16x16 macroblock for luma component as well as 8x8 block for chroma component. While in H.264, variable block-size motion compensation with smaller block sizes is supported, thus picture can be restored with higher accuracy. However, this largely increases the computational complexity.

4) De-blocking Filter: De-blocking filter is a new added part for H.264 in order to minimize the artifacts introduced mainly by present compression technique such as intra prediction. Although it does not promote the PSNR greatly, it can largely improve the subjective appearance.

B. Computational Complexity Analysis

Decoding Steps	MIPS Cost (QCIF 30fps)
Entropy Decoding (CAVLC)	66
Inverse Quantization	21
Inverse Transform	40
Intra Prediction	30
Inter Prediction (Interpolation)	256
De-blocking Filter	467
Other	30

TABLE I

COMPLEXITY

As show above, three tasks appear to be most computational complex in H.264 decoding which are CAVLC decoding, interpolation and de-blocking filter. To be noticed is that owing to limited number of registers (16) in this single scalar DSP, the performance is extremely low.

III. Processor Model

This single scalar DSP processor consists of a DSP core, DSP Debug Unit and a set of Application Specific Accelerators. The DSP core is originally designed for voice processing and supports certain parallel operations. Also reconfigurable memories are used and can be shared with accelerators. An IDE was developed by the DSP vendor to perform simulation, with which new accelerator instructions can be easily defined and simulated. As is shown in Fig. 1, three accelerators are attached to the DSP core in our design. All accelerators receive instructions from the DSP core. The Address Generator can access both Mem0 and Mem1 simultaneously. Pixel Processor and VLC Processor can access data both via DSP core and via Address Generator. Data can be swapped via DMA between off-chip memory and on-chip memory (Mem0 and Mem1).

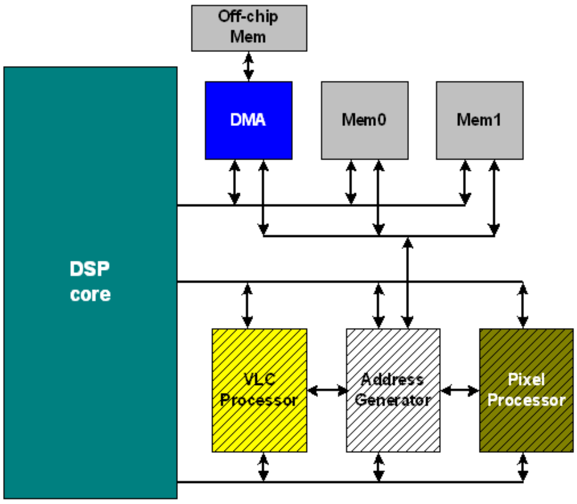

Fig. 1. Processor Model

IV. Memory Issues

According to pixel based operations, number of pixels that can be accessed in one cycle determines the computing parallelism, which means multiple memory banks are required to provide simultaneous memory access. As different from mainstream media processors, the single scalar DSP only has two memory banks. Thus up to two data can be accessed in one clock cycle. In order to achieve access parallelism in pixel manipulation, the following memory partitioning scheme has been proposed.

Based on the memory architecture and instruction set architecture (ISA), a method has been proposed for parallel memory access which improves the performance in pixel level operations. Pixel data in the frame are stored in both Mem0 and Mem1 in an interleaved way, which means adjacent pixel data are stored in different memories as shown in Fig. 2:

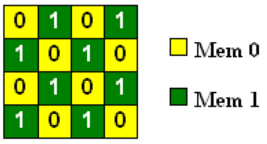

Fig. 2. Memory Architecture

Two pixel data can be accessed (read or write) simultaneously either vertically or horizontally. For example, the pixel data of one 4×4 sub-block can be read out in 8 clock cycles.

V. Addressing Generation Processor

For applications such as video and image processing, 2D addressing is always needed to provide immediate memory access of certain pixel arrays in a certain frame. However this DSP processor originally was not designed for video processing and only supports 1D addressing. Thus in order to perform efficient pixel based memory access, certain 2D to 1D addressing hardware translation is needed. An address generator was designed to achieve addressing together with data load and store.

VI. VLC Processor

Since single scalar DSP is inefficient in entropy decoding which is mainly bit manipulation. A CAVLC accelerator is designed to perform entropy decoding in fewer clock cycles. According to H.264 baseline profile, CAVLC decoding is mapped to this accelerator. VLC processor provides instructions for bit manipulation and lookup table search, as is shown in TABLE II.

The block diagram of VLC Processor is shown in Fig. 3.

VII. Pixel Processor

A. Design Consideration

According to the computational analysis of our software implementation, there are following bottlenecks to be relieved:

1) Inverse Integer Transform: In H.264, inverse transform only involves shift and addition. However, the single scalar DSP can not exploit the parallelism in 4x4 block based IIT operation. Thus simple parallel data path can be applied to achieve high throughput of this part.

Instructions	Functionality
bitreadreset	Initialize the VLC accelerator
showbits Rs, Rt	Load the certain number of bits(number stored in Rs) from bitstream to general register (Rt), without changing the current bit position.
showbits imm, Rt	Same as above, using immediate value (imm)
getbits Rs, Rt	Load the certain number of bits (number stored in Rs) from bitstream to general register (Rt), and change the current bit position.
getbits imm, Rt	Same as above, using immediate value (imm)
flushbits Rs	Change the current bit position
showzeros Rs, imm	Count the number of leading zeros before first 1 in the bitstream, no larger than the immediate value (imm)

TABLE II
VLC INSTRUCTION SET

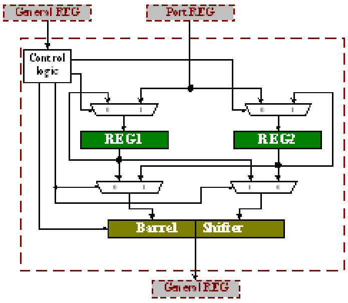

Fig. 3. VLC Processor Architecture

2) Interpolation: Only interpolation of QCIF resolution at 30 fps costs around 250 MIPS which is inapplicable with current single scalar DSP. However, there is large parallelism in interpolation that could be utilized for acceleration. Basically interpolation in H.264 consists of 6-tap half-sample and bilinear quarter-sample operations, which are mainly filtering operations. Thus a multi-tap filter like architecture can be applied to this task.

3) De-blocking: De-blocking has been proved to be the most computational complex part in H.264 decoding. The basic operation of de-blocking is filtering operation on the edges of 4x4 sub-blocks, which is pixel level manipulation. Parallelism of de-blocking operation is related to both data path width and memory access capacity. For de-blocking procedure, there are two major tasks both with parallelism to be exploited:

Getstrength: This part calculates the filtering strength of each edge based on the context information which involves large amount of conditional branches. For example, as is shown in Fig. 4, the filtering strength of sub-block b0 depends on only not the feature of Macroblock A and B but also that of sub-block a0 and b0. And the filtering strength of b0b3 can be calculated along multiple data path in parallel because they are independent to each other. However, because of conditional branches, the parallel data path needs to support simple conditional execution in parallel which requires a small program memory for the accelerator.

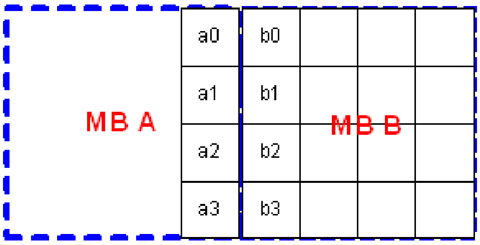

Fig. 4. Getstrength Context

Loopfilter: Loopfilter is mainly multi-tap filters applied to the edge pixels in the decoded frame. Thus similar architecture used by interpolation can also be applied to this part. Still conditional branches can be found in this part, thus conditional execution is also needed for efficient processing.

B. HW Implementation

All tasks list above are pixel based manipulations, from which we can find a lot of similarities in their hardware acceleration. In this paper, a configurable Pixel Processor has been proposed to perform these tasks. Fig. 5 shows the architecture of Pixel Processor, which is divided into two pipelined units: Configurable Filter and Configurable ALU.

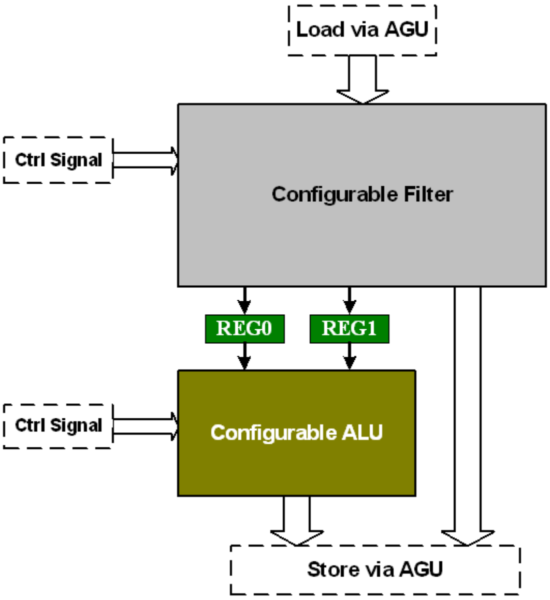

Fig. 5. Pixel Processor Architecture

The Configurable Filter supports the following operations:

- 2-tap to 6-tap filter without multiplication (only constant coefficients are supported)
- 4 way parallel datapath with conditional execution
- 4 way IT/IIT per cycle
- 4/8 way SAD (Sum of Absolute Difference)

The Configurable ALU supports each of the following operations and the combination of them:

- Bi-linear operation
- Round operation
- MAX/MIN operation
- Bubble sort operation

C. *Top-flow*

The tasks are distributed among DSP core, VLC Processor and Pixel Processor as is described in Fig. 6.

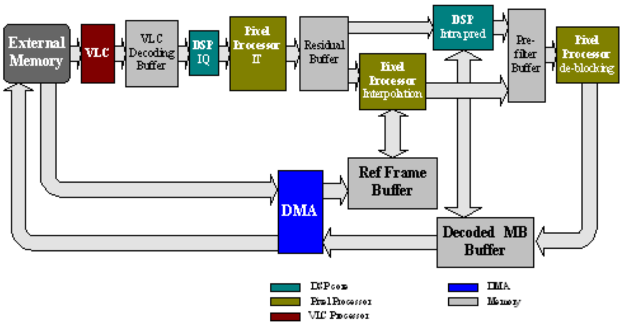

Fig. 6. Top Flow

VIII. Scheduling

Task level parallelism is limited by memory access conflicts, which means, in the same clock cycle, different accelerators may need to access (read or write) one single port memory simultaneously. Because only two memory banks with single port are available in the current processor, memory access is the bottleneck for parallelization. This is the reason why only one Pixel Processor is used in our solution. Since the number of cycles for memory access is very close to that of parallel computing, even if tasks are distributed to more accelerators, they can not be processes in parallel for conflicts in memory access. In our solution, most of the tasks are executed sequentially. However, this solution is not suitable for higher end media processing which requires even higher parallelism.

IX. Performance

In order to perform benchmarking of the newly designed ASIP, accelerators with cycle costs modeling and finite length data types are implemented in C++. Instruction set extension for accelerators was added to the assembly instruction set simulation platform. Based on the simulation platform, the improvement of cycle costs by hardware acceleration was evaluated and compared. The following MIPS cost is estimated based on the new architecture and test sequence such as Forman etc.

A. *VLC Processor*

Tasks	MIPS (QCIF 30fps)	MIPS (CIF 30fps)
CAVLC	6	24

TABLE III

VLC Performance

B. *Pixel Processor*

Because of parallel data path (though without multipliers), the Pixel Processor very well exploits the parallelism in pixel based operations. Thus it provides much higher performance than single scalar processor while compared to traditional SIMD and VLIW processor, it has advantages in lower cost and smaller silicon area.

Tasks	MIPS (QCIF 30fps)	MIPS (CIF 30fps)
IT/IIT	2.5	10
Interpolation	7	28
Getstrength	3.5	15
Loopfilter	8	32
Total	21	85

TABLE IV

Pixel Processor Performance

As show in TABLE III and IV, with configurable HW acceleration, real-time decoding performance of CIF (352×288) resolution or even higher at 30 fps is achieved with acceptable increase of silicon area using clock frequency at 150 MHz.

X. Conclusion

In this paper, a programmable solution has been proposed for media processing. Programmable single scalar DSP processor works as the controller and performs less intensive tasks. Low-cost and configurable accelerators perform computational intensive tasks with high parallelism. Real-time decoding performance can be achieved at resolution which is high enough for all mobile terminals. It has been proved that the solution of single scalar DSP processor plus low-cost accelerators in media processing is not only practical but also has its advantages in time to market and silicon cost. HW reusability also has been explored and has proved that configurable accelerator can accommodate different computing tasks which have similarities in computing and data access.

XI. Future Work

To be noted is that Pixel Processor also supports parallel SAD operations, which means that H.264 encoder can also be implemented together with decoder at lower resolution (e.g. QCIF) which enables video conference. This part will be included in our future work.

References

[1] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050, *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 ISO/IEC 14496-10 AVC)*, March 2003.

[2] Ostermann, J.; Bormans, J.; List, P.; Marpe, D.; Narroschke, M.; Pereira, F.; Stockhammer, T.; Wedi, T., *Video coding with H.264/AVC: tools, performance, and complexity*, Circuits and Systems Magazine, IEEE, Volume: 4, Issue: 1, First Quarter 2004

[3] Malvar, H.S.; Hallapuro, A.; Karczewicz, M.; Kerofsky, L., *Low-complexity transform and quantization in H.264/AVC*, Circuits and Systems for Video Technology, IEEE Transactions on, Volume: 13, Issue: 7, July 2003

[4] Horowitz, M.; Joch, A.; Kossentini, F.; Hallapuro, A., *H.264/AVC baseline profile decoder complexity analysis*, Circuits and Systems for Video Technology, IEEE Transactions on, Volume: 13, Issue: 7, July 2003