

実行時再構成可能なアーキテクチャを用いた MPEG エンコーダの実現

大上 晃弘[†] 一色 剛[†] 國枝 博昭[†]

本稿は、実行時再構成可能なアーキテクチャがストリームデータの処理に適していることを示す。実行時再構成可能なアーキテクチャは、そのアーキテクチャ自体を再構成することにより高いパフォーマンスを達成できる可能性がある。その一方で再構成に時間がかかるという欠点を持つ為、現実的なアプリケーションに応用する事は困難であった。ところが、ストリームデータの処理においては適切なスケジューリングを行う事により、再構成の頻度を減らすことが出来る。この点に着目して本研究では、ストリームデータ処理である MPEG エンコーダをハードモデル上に実装することにより、その有効性を示す。

MPEG/video encoder based on run-time reconfigurable architecture

AKIHIRO OUE,[†] TSUYOSHI ISSHIKI[†] and HIROAKI KUNIEDA[†]

This paper shows that run time reconfigurable architecture is suitable for processing streaming data. Run time reconfigurable architecture has possibility to achieve higher performance by changing its architecture. On the other hand, since it takes many clocks for configuration, it is difficult to apply it directly to real time applications. However, appropriate scheduling reduces times for configuration in processing streaming data. In this research, by realizing MPEG encoder as processing streaming data on hardware model, we show the validity of run time reconfigurable architecture.

1. Introduction

Reconfigurable system is a system, whose configuration can be modified or programmed in the field before it invokes. FPGA is a typical example for this reconfigurable system. Since its configuration is performed in advance, it may be called as static or compile time reconfigurable system. On the other hand, if the system configuration or architecture can be programmed while invoking, it may be called as dynamic or run time reconfigurable system. Run time reconfigurable system is to achieve higher performance by changing its architecture drastically.

If run time reconfigurable system changes its configuration every clocks, the system may not be called run time reconfigurable system. It may rather be called programmable system, because the

provision of configuration data to the system in reconfigurable system is essentially equivalent to issue of control signal to data path in programmable system. Programmable system must use the instruction and control signals generated by control unit (CU) to control the data streams every clocks. On the other hand, in run time reconfigurable system, CU may also generate control signal or configuration signal to configure the data path in many clocks.

From this viewpoint, we can compare generation scheme of control signals in VLIW³⁾ and SIMD⁴⁾ as programmable system, with run time reconfigurable system.

In VLIW system, since PEs are controlled by corresponding CU, the system has an advantage that various algorithms can be realized. However, because circuit area of CUs are rather large, many PE cannot be implemented in area restricted cases. In SIMD system, since PEs are controlled by identical control signal generated by shared CU, circuit area of CU is small. However, because of identi-

[†] 東京工業大学 工学部 電気・電子工学科
Department of Electrical and Electronic Engineering,
Tokyo Institute of Technology

cal control signal, algorithms which have random dependency among operations of PEs cannot be effectively realized in SIMD system. In run time reconfigurable system, since each PE is controlled by each control signals generated by CU in many clocks, circuit area of CU is small and various algorithms can be realized.

One of objectives of this research is to investigate the validity of run time reconfigurable system for streaming data. For this purpose, we try to develop hardware model for reconfigurable system and to realize efficient system. We select MPEG2¹⁾ encoder as our target algorithm, because MPEG2 encoding includes various types of computations.

2. Run-time Reconfigurable System

We consider here the block diagram of run time reconfigurable system as shown in Fig 1. This system consists of functional blocks such as execute block, control block, internal ram, external ram and host processor. In the following, we explain outlines of these blocks.

Execute block This block executes algorithms which need relatively large calculation costs in MPEG encoder, such as gray parts in Fig 2. This block is structured PEs in rectangular array, and reconfigurable interconnections among them. The interconnections are implemented by separable buses. Also we assume here that array size is 32×24 .

Control block This block generates configuration signal for execute block, and address of both internal RAM and external RAM. Because one control block are shared by 384 PEs in execute block, control block takes 384 clocks for configuration of the array.

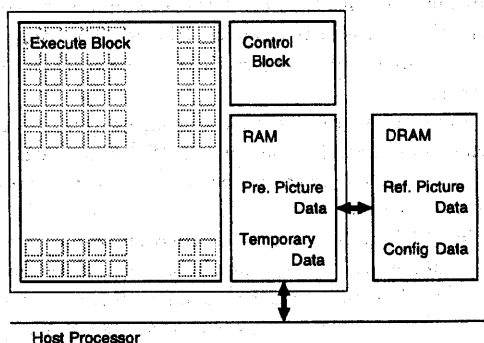


Fig 1. run time reconfigurable system

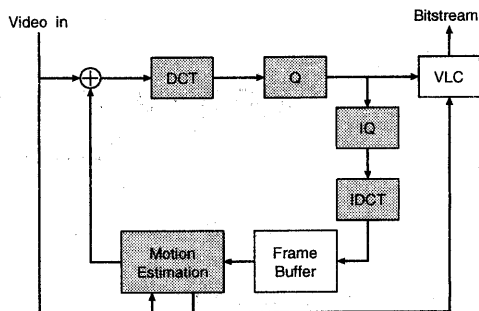


Fig 2. algorithms consist of MPEG encoder

Internal RAM This RAM stores prediction picture and temporary data. It needs relatively high band width for communication between execute block and internal RAM. We assume here internal RAM size is 16 k word. (1 word = 9 bit)

External RAM This storage is used for data of relatively low band width, such as reference picture and instruction data. We assume here band width between external ram and internal ram is 16 byte per clock.

Host processor This processor executes algorithms which is rather small calculation cost, such as VLC, rate control and so on. This part transfers prediction picture data and quantized data to outside.

This system has two modes, config-mode and execute-mode. When config-mode, control block issues instructions for all the processors in execute block. Each instruction is issued in every clocks for each processor. When execute-mode, execute block executes instruction issued by control block. To make this system in one of these two mode alternately, we can realize MPEG encoder.

We perform behavior simulator for small system consisting of several PEs. Although whole system in gate level was not realized, we could estimate that our system run at 50 MHz and configuration can be done within 384 clocks.

3. Execute Block

Execute block consists of PEs in rectangular array, and interconnection among them as shown in Fig 3. In the following section, We explain processor element and interconnection.

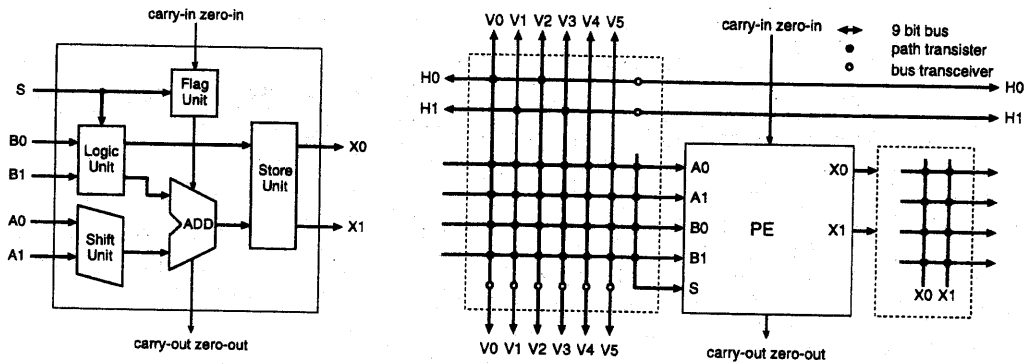


Fig 3. PE and interconnection

3.1 Processor Element

Each PE consists of flag unit, shift unit, logic unit, store unit and 9 bit adder. Control block can issue control signal for each unit each unit. In the following, we explain each unit in detail.

Shift unit This unit consisting of barrel shifter executes any shift operation. Data is input from bus A0 and A1, and output to adder.

Logic unit This unit consisting of look up table executes any logic operation. Data is input from bus B0, B1, and signal S, and output to adder and store unit.

Flag unit This unit controls carry-in and zero-in signal for adder. Carry-in signal can be 1, 0, S, \bar{S} , or carry-out signal from previous processor element. Signal zero-in can be 0 or signal zero-out like from previous processor element.

Store unit This unit consisting of two 9 bit registers, stores results of execution. One can store only adder's output. The other can store either one of bus B0, output of logic unit, or signal zero-out of adder.

3.2 Interconnection

Interconnection among PEs are implemented by global bus (H0, H1, V0, ...V5) and local bus (A0, A1, B0, B1, S). We assume here width of these buses is 9 bit. In the following, we explain buses around one processor element.

Global bus Global buses are separable, because of the usage of different bus transceivers. Bus H0 and H1 are available for communication for horizontal direction. Bus H0 can be connected to Bus V0 and V2. H1 can also be connected to V1 and V3. Bus V0, ...V5 are available for communication for vertical direction.

Local bus Local bus A0, A1, B0, B1 are con-

nected to the processor element as its input and can be connected to global bus V0, ...V5. While bus S is connected to the processor element as its input. It can be connected to bus A0, A1, B0, B1. Bus X0 and X1 which are output of the processor element, can also be connected to local bus A0, A1, B0, B1.

4. Algorithm

We can freely select algorithms and their parameters in MPEG encoder. However, goal of this research is to show that run time reconfigurable architecture is suited to MPEG encoder. Therefore we dare to select orthodox algorithms. In the following section, we explain main algorithms for MPEG encoder.

4.1 Motion Estimation

To search a motion vector for each macro blocks, we can compress image making use of timing redundancy. In this research, we adopt 2 level search as a motion estimation algorithm. This search algorithm is as following step.

Step 1 After applying down-sampling to both prediction picture and reference picture, we search the least point of Mean Absolute Difference (MAD) over -7 to +7 pixel range. (level 1 search)

Step 2 Applying up-sampling only for reference picture, we search the least point of MAD within -2 to +1 pixel shift range from the point found by Step 1. (half pel search)

4.2 DCT/IDCT

To apply two dimensional DCT for each block, we can compress image making use of special redundancy. In this research, since we realize two di-

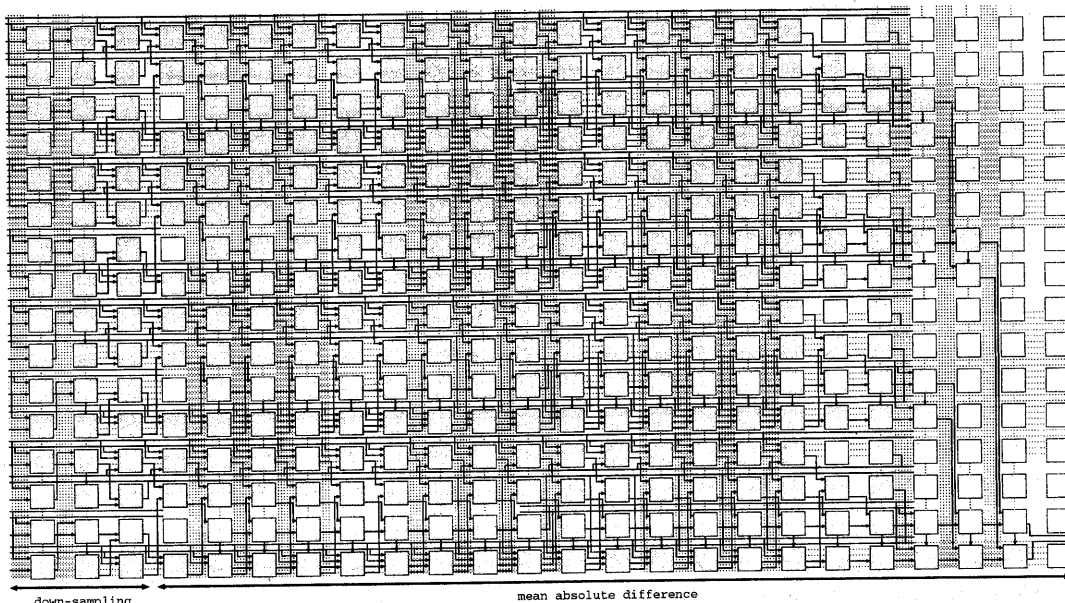


Fig 5. interconnections of ME1

mensional DCT by using fast algorithm⁵⁾ for one dimensional DCT and replacement, the system must be configured twice for two dimensional DCT. Two dimensional IDCT is realized two steps by using fast algorithm such as DCT, too.

5. Schedule

By making this system in one of config-mode and execute-mode alternately, we can realize MPEG encoder. In the following section, we explain about configuration, scheduling strategy and estimation of scheduling results.

5.1 Configuration

To realize MPEG encoder, this system must be configured several types as shown in Fig 4. Here two dimensional DCT is realized by a couple of configuration such as DCT1 and DCT2. DCT1 is one dimensional DCT and then some part of replacement. DCT2 is rest of replacement and then one dimensional DCT. Thus, two dimensional DCT is realized. IDCT is realized in similar manner.

For example, interconnection of ME1 is roughly shown in Fig 5. Both reference picture and prediction picture are input from the left edge of execute block. MAD as the result is output to the right edge.

type	function
ME1	Down-sampling for pre. picture and Motion Estimate (level 1 search)
ME2	Up-sampling for ref. picture and Motion Estimate (half pel search)
DCT1	1-D DCT then data replacement
DCT2	data replacement then 1-D DCT
IDCT1	1-D IDCT then data replacement
IDCT2	data replacement then 1-D IDCT
Q	Quantization
IQ	Inverse Quantization

Fig 4. configuration types for MPEG encoder

5.2 Strategy

In this system, control block takes 384 clocks to generate configuration signals. It is not desirable to configure the system for every macro block, because overhead of configuration is too large. This case is shown in Fig 6(a). Therefore we take a scheme that CU generates a configuration signal for several macro blocks. The cases of configuration for every 5 and 15 macro blocks are shown in Fig 6(b) (c), respectively. Using this scheme, we reduce overhead of configuration. However internal RAM size becomes larger, because large prediction picture data and temporary data size must be stored.

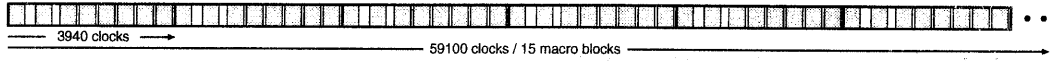


Fig 6(a). configure for every macro block

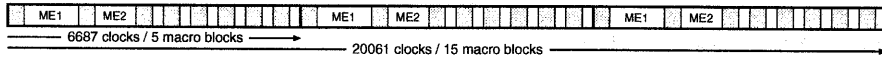


Fig 6(b). configure for every 5 macro blocks

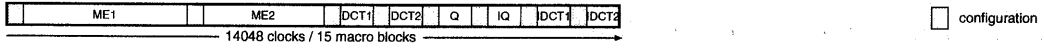


Fig 6(c). configure for every 15 macro blocks

5.3 Estimation

In the following, we explain the scheduling result for every picture types. Here, we assume system frequency of 50MHz, and reconfigure execute block in every 15 macro blocks.

I picture Intra-coded picture is encoded only by using own picture. For encoding I picture, encoder use DCT and Quantize. Moreover to use as reference picture, this picture is stored in external RAM after being applied Inverse Quantize and IDCT. Therefore the scheduling result for I picture is as shown in Fig 7.

P picture Predictive-coded picture is encoded by using both own picture and picture in the past. For encoding P picture, encoder use Motion Estimation, DCT and Quantize. Moreover to use as reference picture, this picture is stored in external RAM after being applied Inverse Quantize and IDCT. Therefore the scheduling result for P picture is as shown in Fig 8.

B picture Bidirectionally predictive-coded picture is encoded by using own picture and picture in both the past and the future. For encoding B picture, encoder use Motion Estimation, DCT and Quantize. Therefore the scheduling result for B picture is as shown in Fig 9.

All of these results show the processing time is taken in less than 33 m sec which is required processing time for real time encoding. Chip area is estimated as 100 mm² by 0.6 μm CMOS double metal process. This value may achieve area reduction from the conventional approach.

6. Summary

In this paper, we show hardware model for run time reconfigurable architecture and to realize MPEG encoder in our system. As the result

algorithm	execute	config	sum
DCT	2.68 ms	1.38 ms	4.06 ms
Q	1.34 ms	0.69 ms	2.03 ms
IQ	1.34 ms	0.69 ms	2.03 ms
IDCT	2.68 ms	1.38 ms	4.06 ms
Total	8.03 ms	4.15 ms	12.19 ms

Fig 7. scheduling result for I picture

algorithm	execute	config	sum
ME (level 1)	6.75 ms	0.69 ms	7.44 ms
(half pel)	4.97 ms	0.69 ms	5.66 ms
DCT	2.68 ms	1.38 ms	4.06 ms
Q	1.34 ms	0.69 ms	2.03 ms
IQ	1.34 ms	0.69 ms	2.03 ms
IDCT	2.68 ms	1.38 ms	4.06 ms
Total	19.76 ms	5.53 ms	25.29 ms

Fig 8. scheduling result for P picture

algorithm	execute	config	sum
ME (level 1)	13.50 ms	0.69 ms	14.19 ms
(half pel)	9.94 ms	0.69 ms	10.63 ms
DCT	2.68 ms	1.38 ms	4.06 ms
Q	1.34 ms	0.69 ms	2.03 ms
Total	27.46 ms	3.46 ms	30.91 ms

Fig 9. scheduling result for B picture

for scheduling, this system is able to encode in real time. Until now, no programmable system has been able to encode MPEG2 in real time. Proposed system is promising to realize programmable system for real time MPEG encoding.

Acknowledgment Authors would like to thank the members of CAD21 Research Body of Tokyo Institute of Technology and members of Kunieda Laboratory for their suggestion and cooperations.

References

- 1) "Information Technology - Generic Coding of Moving Pictures and Associated Audio," ISO/IEC 13818-2 International Standard (Video), Nov. 11, 1994
- 2) Yasushi Ooi "An MPEG2 Encoder Architecture Based on a Single-chip Dedicated LSI with a Control MPU" IEEE ICASSP, pp 599-602 April, 1997
- 3) Toru Shimizu., et al. "A Multimedia 32b RISC Microprocessor with 16Mb DRAM" IEEE ISSCC Digest of Technical Papers, pp 216-217 Feb., 1996
- 4) Masuyoshi Kurokawa., et al. "5.4GOPS Linear Array Architecture DSP for Video-Format Conversions" IEEE ISSCC Digest of Technical Papers, pp 254-255 Feb., 1996
- 5) C. Loeffler, A. Ligtenberg and G. S. Moschytz, "Practical Fast 1-D DCT Algorithms with 11 Multiplications," Proc ICASSP, pp 988-991 1989