

A Vector-Pipeline DSP for Low-Rate Videophones

K. KOBAYASHI, M. EGUCHI, T. IWAHASHI, T. SHIBAYAMA, X. Li, K. TAKAI and H. ONODERA

Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan

Abstract

We propose a vector-pipeline processor VP-DSP for low-rate videophones, which can encode and decode 10 frames/sec. of QCIF through a 29.2kbps low-rate line. We have already fabricated a VP-DSP LSI by a $0.35\mu\text{m}$ CMOS process. The area of the VP-DSP core is 4.2mm^2 . It works properly at 25MHz/1.6V with the power dissipation of 49mW. Its peak performance is up to 400MOPS, 8.2GOPS/W.

1 Introduction

Standard video codecs, such as MPEG and H.263 are based on discrete cosine transform (DCT). They require a large amount of computation for both encoding and decoding. On the other hand, Vector Quantization (VQ) is a powerful technique for low-rate image coding[1]. Compared with DCT-based techniques, a video sequence compressed by VQ can be easily decompressed.

We have proposed a fixed-rate multi-stage hierarchical vector quantization algorithm[2] abbreviated to FRMSHVQ for real-time low-rate video compression. It has the capability to send 10 QCIF(176×144) frames per second through a 29.2kbps transmission line. We have already developed a PC-based real-time compression system, where a functional memory called FMPP-VQ performs vector quantization and a PC performs the rest of the operations. FMPP-VQ performs VQ at low power, but the rest of the operations are performed on the PC and consume much power. The FRMSHVQ algorithm contains two time-consuming operations, VQ and ME (Motion Estimation). They are similar operations in that a vector nearest to an input one is sought from a set of reference vectors. A lot of distances between the input and reference vectors must be computed. Since the distance computation accumulates the distances of all elements, an SIMD parallel processor adapted to VQ and ME can process the FRMSHVQ algorithm efficiently.

2 Vector-Pipeline DSP for Real-Time Image Compression

The FRMSHVQ algorithm contains ME and VQ, which search the vector or the pixel block nearest to an input one among a prepared set of reference vectors or blocks. They are suitable to SIMD processing since the large amount of SADs(Sum of absolute distances) must be computed.

In the FRMSHVQ algorithm, VQ is applied hierarchically to multiple stages, where the dimension of vectors is always fixed to 16. A codebook has 64 code vectors, each of which is expanded to 16 to rearrange its elements. The nearest vector is chosen from $64 \times 16 (=1024)$ code vectors. Therefore, 16 discrete SADs are computed for an input vector and a code vector. If a processor has a 16-parallel SIMD operation, these 16 discrete SADs can be computed in parallel.

To make the FRMSHVQ algorithm process efficiently, we propose a vector-pipeline DSP abbreviated to VP-DSP. VP-DSP has short-bit scalar registers and long-bit vector registers in a register file. The ALU performs short-bit scalar operations in serial and vector operations, where each element is short-bit, in an SIMD manner. Unlike the subword extensions in general-purpose or multimedia processors, vector registers are not used for long-bit scalar operations, but only for short-bit SIMD operations. The VP-DSP architecture realizes a relatively simple structure since the ALU handles short-bit operations and only a limited number of registers have long bit-widths.

Fig. 1 shows a block diagram of VP-DSP. It is based on a conventional DLX processor, but four registers (V1-V4) out of 15 are replaced by 160-bit vector registers for SIMD parallel processing. The other 11 registers (S1-S11) are 20-bit for scalar processing. A vector register stores 16 10-bit elements for SIMD operations. VP-DSP performs vector/scalar instructions between vector/scalar registers according to 22-bit instruction codes. Vector instructions perform 16-parallel SIMD operations, during which the same operation is applied to 16 elements stored in vector registers. To resolve pipeline hazards, the pipeline registers in the MEM and WB stages are forwarded to the EX stage.

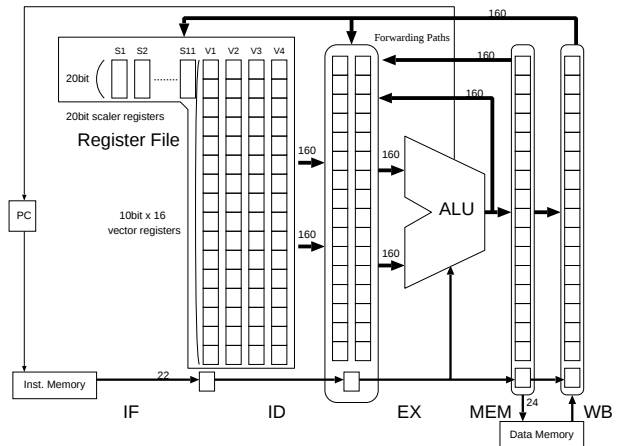


Figure 1: Block diagram of VP-DSP.

To accelerate VQ and ME, VP-DSP has vector instructions as in Table 1. To compute the 16 absolute distances for an input vector $\vec{x}$ and a code vector $\vec{y}$, the suba and accd instructions are iterated 16 times.

Table 1: Vector instructions.

Assembly	Semantics
suba rdv,rsv1,rsv2	$rdv[i] \leftarrow rsv1[i] - rsv2[i] $
adds rdv,rsv1,rsv2	$rdv \leftarrow rsv1 + rsv2$ ($0 \leq rdv \leq 63$)
thin rdv,rsv1,rsv2,d	$rdv[d \times 4] \leftarrow rsv1[0,4,8,12]$ $rdv[others] \leftarrow rsv2$
accu rdv,rsv1,rsv2	$rdv[i] \leftarrow rsv1[i+1] + rsv2[i]$
accd rdv,rsv1,rsv2	$rdv[i] \leftarrow rsv1[i-1] + rsv2[i]$

rsv: src vector register rss: src scalar register
rdv: dst vector register rds: dst scalar register

3 Performance

We compare VP-DSP with a virtual scalar DSP that has the same instruction set as VP-DSP. Table 2 shows the number of clock cycles for VQ and ME and the areas of logic gates. VP-DSP is 28 times faster than the virtual scalar DSP for VQ and 5 times faster for ME. In ME, out of 1,294 operations, 790 are load instructions. The load instruction on VP-DSP loads four elements simultaneously in a vector register. This is why VP-DSP is only 5 times faster than the virtual scalar DSP. In VQ, VP-DSP computes 16 absolute distances without any load instruction. VP-DSP loads 65 vectors (one input vector and 64 code vectors) during the VQ operations, which takes $520 (= 2 \times 4 \times 65)$ clock cycles. Note that four iterations of addi and lu operations are required to load 16 elements to a vector register. On the other hand, the virtual scalar DSP has to load code vectors from data memory for each element. This is why VP-DSP is 28 times faster, even though the number of parallel operations is 16. As for the area, VP-DSP is only four times larger than the scalar DSP. Therefore, VP-DSP can compute the FRMSHVQ algorithm very efficiently.

Table 2: Number of clock cycles of VQ and ME and areas.

		VP-DSP	virtual scalar DSP
# of clock cycles	VQ	4,872	138,243
	ME	1,294	6,221
Area(mm ²)		2.26	0.57

4 Implementation

We have fabricated a VP-DSP LSI using a 0.35 μ m process on a 24.0mm² die. The VP-DSP core is synthesized from a Verilog-HDL description and automatically placed-and-routed.

Figure 2 shows the chip micrograph of the VP-DSP LSI. Two DSP cores are implemented on the same die in order to evaluate two LSI libraries, EXD Lib.(#1) and On-Demand Lib(#2). The detail of the libraries is explained in [4]. The 512-word 24-bit SRAM works as a part of data memory. Table 3 describes the specification of the VP-DSP LSI

We have measured the VP-DSP LSI by an LSI tester. Fig. 3 shows a shmoo plot of the VP-DSP core #2 to sweep the supply voltage and clock cycle. Note that the constraint at design time is 50MHz and 3.3V. Fig. 3 contains several measured power

Table 3: Specifications of the VP-DSP LSI.

Area	24.0mm ² (Die), 4.68mm ² (#1), 4.26mm ² (#2)
# of gates	28,956(#1), 31,393(#2)
Power	50mW(#1), 49mW(#1) (@25MHz/1.6V)

dissipation values, at which the test vector is the VQ of 64 code vectors. To complete the FRMSHVQ algorithm in real time, the power dissipation is 49mW at 25MHz/1.6V. The peak performance is 400MOPS (=25MHz $\times$ 16 parallel operations) and 8.2GOPS/W.

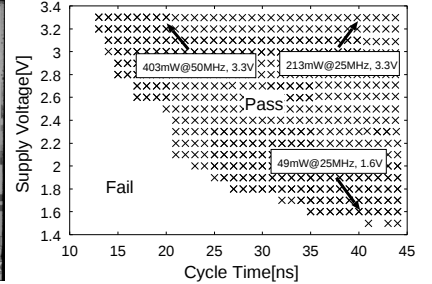
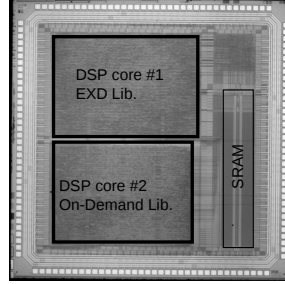


Figure 2: Chip micrograph Figure 3: Shmoo plot of the core #2 of the VP-DSP LSI.

5 Conclusion

VP-DSP is a vector-pipeline processor for real-time image compression using the FRMSHVQ algorithm. It has four 160-bit vector registers and 11 20-bit scalar registers. Each vector register contains 16 10-bit elements that enable 16-parallel SIMD vector operations. The FRMSHVQ algorithm requires no multiplication and handles an image by a 4 $\times$ 4 pixel block. Thus, VP-DSP contains no multiplier, which minimizes the required area and decreases power consumption.

We have fabricated a VP-DSP LSI using a 0.35 μ m CMOS process on a 24.0mm² die. Two DSP cores are implemented by two independent libraries. One DSP core occupies 4.26mm² and consumes 49mW at 25MHz/1.6V. The peak performance of VP-DSP is 400MOPS and 8.2GOPS/W. VP-DSP compresses 10 QCIF frames to 29.2kbps in real time under the low clock frequency of 25MHz.

Acknowledgment

The VLSI chip in this study has been fabricated in the chip fabrication program of VLSI Design and Education Center(VDEC), the University of Tokyo with the collaboration by Rohm Corporation and Toppan Printing Corporation.

References

- [1] Y. Linde et. al., IEEE Trans. Commun. (USA), vol.COM-28, no.1, pp. 84–95, 1980.
- [2] K. Kobayashi et. al., IEICE Trans. on Fundamentals, vol.E82-A(2), pp. 215–222, 1999.
- [3] J. Henessy and D. Patterson, “Computer Architecture, A Quantitative Approach”, Morgan Kaufmann.
- [4] T. Hashimoto and H. Onodera, Proc. of SASIMI2000, pp. 122–129, 2000.