A 124.8Msps, 15.6mW Field-Programmable Variable-Length Codec for Multimedia Applications

*Chingwei Yeh**
Nat'l Chung-Cheng University
ieecwy@ccu.edu.tw

Chao-Ching Wang
Nat'l Chung-Cheng University

Lin-Chi Lee
Nat'l Chung-Cheng University

Jinn-Shyan Wang
Nat'l Chung-Cheng University
ieegsw@ccu.edu.tw

Abstract

Variable-length coding is one of the key compression methods for multimedia bitstreams. To accommodate new or user-defined variable-length codes (VLC) for maximal compressions in various applications, we propose a variable-length codec that supports field programmability along with very competitive performance indices. The design has 33% less transistors than its field-programmable predecessor. Moreover, measurement on the real chip demonstrates that the design is capable of processing 124.8 mega-symbols (Msym) per second for MPEG4, while consuming only 15.6mW at 1.4V. When measured by μW/Msym, the realized variable-length codec is even 5% better than the state-of-the-art non-programmable MPEG2 variable-length decoder that hardwires the entire design into random logic.

1. Introduction

Variable-length coding is a data compression scheme originated from [1]. The main idea is to minimize the average codeword length by exploiting the statistics of the data. Shorter codewords are assigned to more frequent data while longer codewords are assigned to less frequent data. Therefore, minimum average code length can be achieved. Due to its high compression efficiency and simple operations, the variable-length code (VLC) has been adopted as a part of many image and video coding standards, such as JPEG, MPEG2, MPEG4, etc. Traditional designs of VLC encoder/decoders followed [1] by traversing the code tree [2][3][4], meaning that encoding/decoding time is proportional to the length of the codeword. Hence, the scheme is not quite suitable for high-performance applications, particularly when there are many long codewords. Also, direct storage of the code table results in very inefficient memory usage. These drawbacks soon attracted several research attentions. The major novelties lay in flattening the sequential, code-dependent search into parallel pattern-matching [5][6][7], and removing redundancies in the code space so as to reduce the memory requirement [8][9]. In these works, hardwired kernels, e.g., PLA, ROM, random logic, were often the choice for high speed pattern matching.

Nevertheless, it is aware that new or user-defined VLC's may be necessary to maximize compression for applications that may experience changes or upgrades during product life cycle. In this case, hardwired tables become awkward and expensive as they require new mask preparation and lead-time for each new design. Also, the hardwired approach lacks the ability to update the device on-line, which is now an important feature in a ubiquitous networking environment.

The above problems call for field programmability that was beyond the capability of hardwired kernels. One might then think of content-addressable memory (CAM) to the rescue [10][11]. However, the area and cost penalty is often too large to be practical. It was not until [12] that parallel, arithmetic computations on grouped codewords and their indices were employed to eliminate content-based searching and to reduce memory size. The scheme starts by dividing VLC's into groups according to code length and code prefix. With such a grouping, the VLC's within the same group display a nice property: their equivalent numerical values are continuous. In other words, the first VLC in a group and the offset are sufficient to index all other VLC's in the same group. Thus, it is no longer necessary to store all information pertaining to each VLC, and a smaller RAM instead of CAM can be used to attain field programmability.

It can be seen that the critical operation in such a scheme, called *group indexing*, is identifying the group to which an input bit string belongs. In [12], this is accomplished via as many parallel subtractors as the designated parallelism. Such use of arithmetic computations results in serious degrades of the design quality. To solve the problem, we propose a new type of memory called the Ternary Data-Indexed Memory (**TDIM**). The TDIM is used for group indexing as follows. Each bit of a VLC codeword is treated as a Boolean bit and so each VLC codeword as a product term (PT). A group of VLC codewords can then be

3-9810801-0-6/DATE06 © 2006 EDAA

represented as a *Boolean function* whose ON-set comprises of the PTs. By storing all PTs in TDIM, group indexing is achieved via asserting whether a given VLC codeword matches with (in a ternary sense) the designated content in TDIM. As will be elaborated in subsequent sections, the use of a well-designed TDIM substantiates the power and speed advantage of our variable-length codec (VLC/D).

2. System architecture

Fig. 1 shows the VLC/D core architecture proposed by [12], with the arithmetic computations replaced with TDIM's. Basically, for encoding, the bit string is checked for escape conditions and adapted to fetch the right symbol address. The symbol address is used to index the TDIM for group indentification. The output of TDIM is then used to compile the final VLC. Along the way, the amount of shifts to align the input for the next symbol processing is computed (shown with multiple additions in the figure). The decoding process is similar to the encoding and hence is omitted. The readers are referred to [12] for detail description of each component.

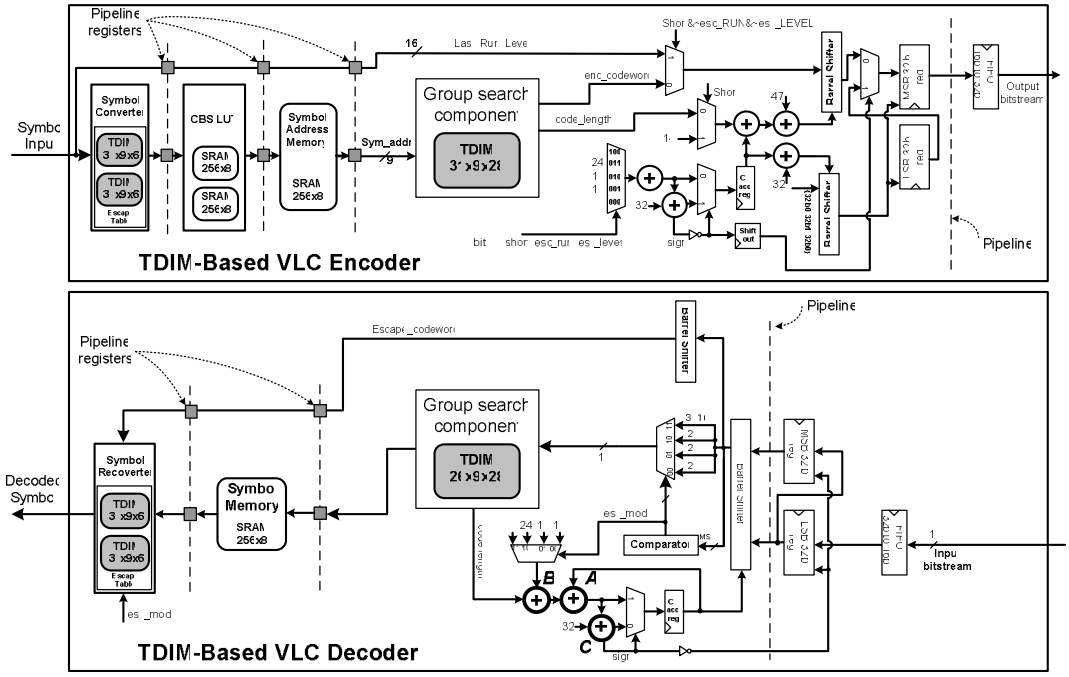


Fig. 1 VLC/D Architecture [12] adapted with TDIM's (initial pipeline).

3. The TDIM design

The functionality of TDIM is realized in three parts (Fig. 2): a parallel ternary comparator array (PTCA) similar to the matching circuitry of a ternary CAM, a pair-wise priority encoder (PWPE), and a binary cell array (BCA) that stores the information for subsequent processing when the right group has been identified for an input bit string.

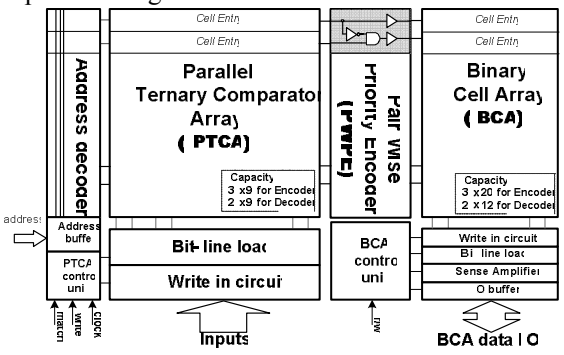


Fig. 2 Block diagram of TDIM

The PTCA is the critical part of TDIM as it consumes the most time and power. In contrast to recent works on CAM that focused on reducing the power of NOR-type matching circuits, the PTCA employ cascaded domino AND gates to quickly compute the bit-wise comparison results from the 9-transistor comparator cell. The design not only takes advantage of the inherent low-switching-activity in the AND circuitry as in [13], but also achieves much less comparison time via pseudo-footless clock-and-data precharged dynamic (PF-CDPD) circuitry [13].

To see how the PF-CDPD scheme contributes to search speed, use Fig. 3 as the example. The slowest case happens when the input data matches with the stored data. In this case, all NMOS transistors in the pull-down networks (PDNs) receive logic 1 during pre-charge and their drain nodes are being pulled to the ground level. This results in pseudo ground during evaluation, and the PF-CDPD behaves much like a series of inverters. Therefore, propagation time of the AND gates is greatly reduced. A 2-stage PF-CDPD circuit, with a fan-in of

four for the first stage and five for the second stage, has been empirically determined to offer the best operating speed.

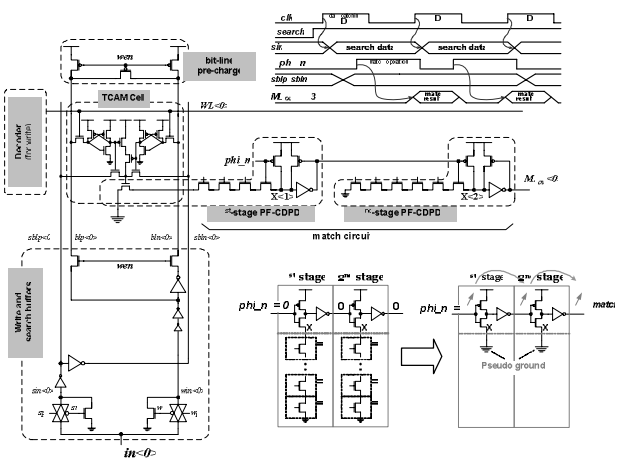


Fig. 3 PF-CDPD Circuits for the PTCA in TDIM

In addition to high speed, the PF-CDPD match circuit also results in low power for the following reasons. Firstly, the match circuit has inherently a low switching activity due to AND configuration. Secondly, due to the CDPD scheme, the evaluation of the second stage depends on the result of the first stage. In other words, partial mismatch in any bit of a VLC codeword immediately turns off the matching for subsequent bits. Since the VLC codewords are distinguishable in big-endian order, such a conditional matching offers dramatic power saving. Lastly, the pesudo-footless scheme reduces the charging/discharging capacitance and again contributes to low power.

Since the number of PTs is directly related to the size of TDIM, typical logic minimization serves as the first, necessary procedure. Still, further minimization is possible by adding proper dummy PTs into the Boolean function of a group. This is where the PWPE at the PTCA outputs come into play—solving the aliasing problems between adjacent groups. Taking the MPEG4 AC tables as an example, logic minimization with the dummy PTs successfully reduces the number of PTs from 103 to 21.

4. Retiming for speed and power tradeoff

The architectural simplicity of using TDIM in VLC/D also motivated us to think of reducing cycle time via retiming. Referring to Fig. 1, it can be seen that cycle time can be reduced only by moving adders A, B, C, and their multiplexer outside the current TDIM cycle. However, the data dependency problem, shown in the feedback path within the TDIM cycle, precludes straightforward retiming. We therefore resort to algorithmic features and add one register for moving adder A, one Barrel shifter for moving adder B. Once adders A, B have been moved, the remaining components can be moved, too. Fig. 4 shows the resultant, complete VLC/D for MPEG4 (the VLC/D for MPEG2 is the same in architecture but different in memory configuration and escape handling).

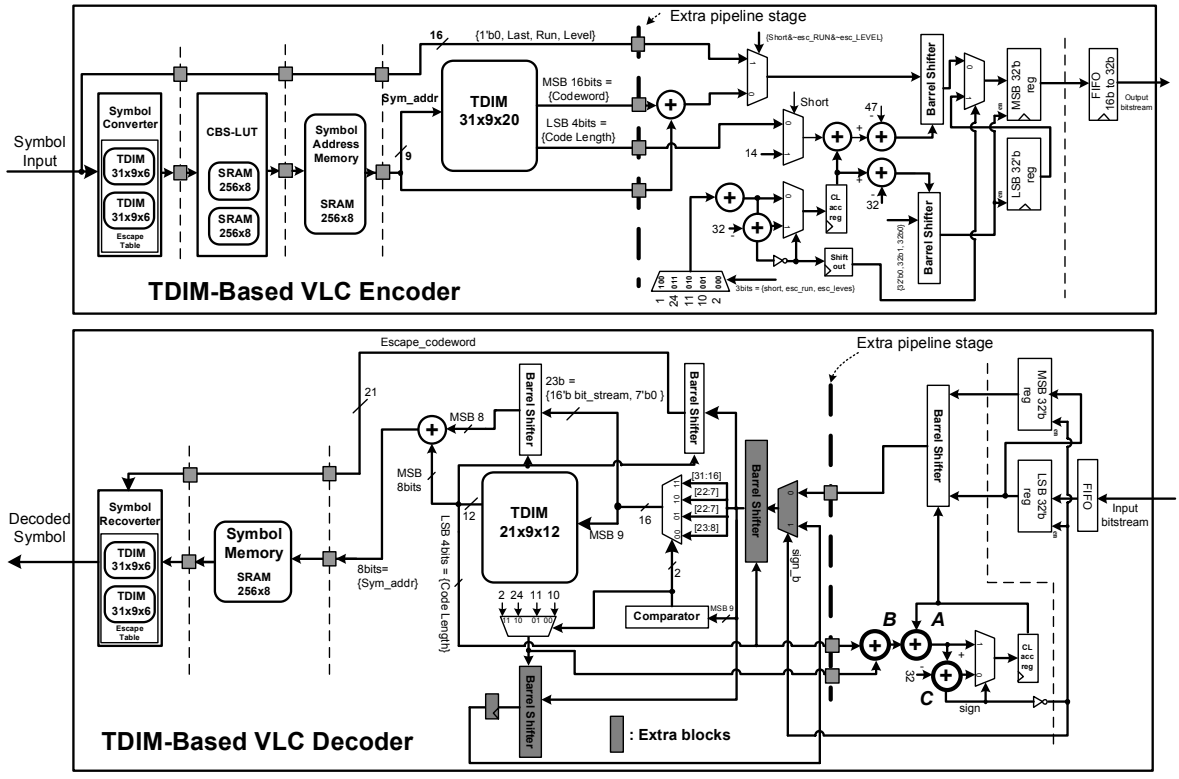
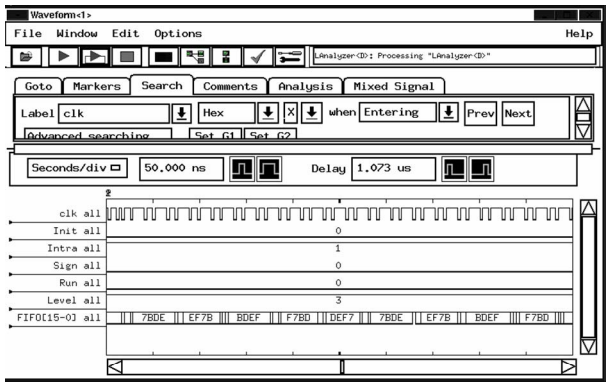


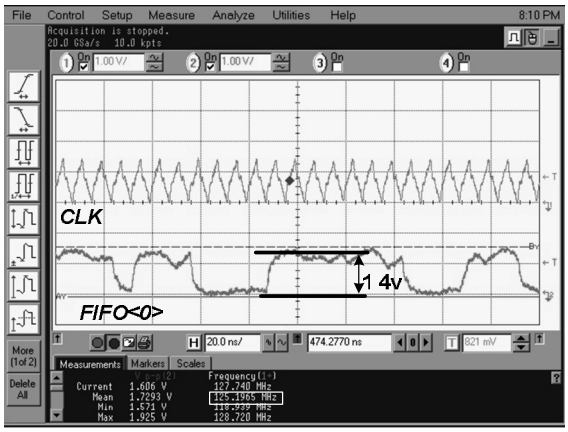
Fig. 4 Retimed VLC/D of [12] with TDIM's

5. Results and conclusions

The proposed VLC/D for MPEG4 is designed and fabricated in CMOS 0.18μm technology. For comparison, we also built a pre-layout version for MPEG2 based on the same TDIM architecture. The results are shown in Table 1. Compared to previously reported field-programmable work for MPEG2 [12], our design offers 33% reduction in transistor count and has much less layout complexity due to the use of TDIM. Thus, it is conceivably that our design would have very favorable post-layout result compared to [12]. Moreover, when measured by power consumption (μW) per mega symbols, our MPEG4 VLC/D is even 5% better (124.80μW/Msym versus 131.21μW/Msym) than the state-of-the-art low-power MPEG2 VLD (decoder only) [9] which hardwires the entire design into random logic. The measurement plots are shown in Fig. 5. The chip micrograph is shown in Fig. 6 with a VCO embedded for generating the on-chip clock.



(a)



(b)

Fig. 5 Measurement results, (a) logic analyzer, (b) scope

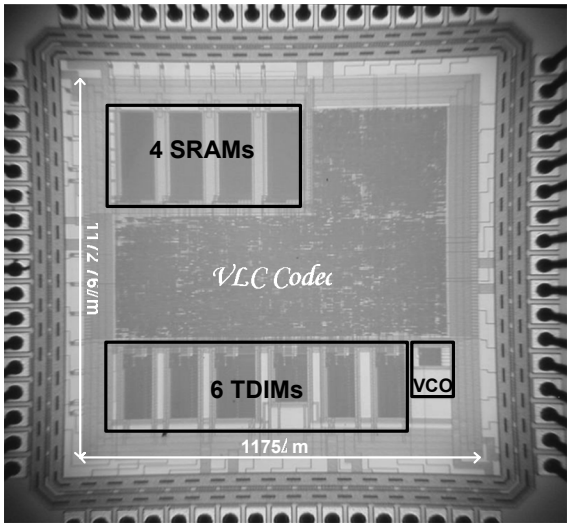


Fig. 6 Die Photo

Table 1 Comparison of Different VLC/Ds and VLDs

Programmability	None (Hardwired)		Field-Programmable VLC/D		
Function	Decoder Only		Encoder & Decoder		
References	[8] (Redesign by [9])	[9]	[12]	This Work (Pre-layout)	This Work **(Measured)**
Standard	MPEG2		MPEG2		MPEG**4**
Technology	0.35μm		0.6μm	0.18μm	0.18μm
Transistor Count	-	-	***110K*** (with mem)	***74K*** (with mem.)	200.6K (with mem.)
Area (mm^2)	-	-	22.5	-	1.38
VDD	-	-	5.0V	-	1.4V
Frequency	-	-	100MHz	-	125MHz
Throughput (sym/cycle)	0.56 (avg)	0.63 (avg)	1	-	1
Power metric (μW/Msym)	***204.48***	***131.21***	***N.A.***	-	***124.80***

References

[1] A. Huffman, “A method for the construction of minimum–redundancy codes,” Proc. IRE, vol. 40, pp. 1098–1101, Sept. 1952.

[2] Mukherjee, N. Ranganathan, J. W. Flieder, and T. Acharya, “MARVLE: A VLSI chip for data compression using tree-based codes,” IEEE Trans. VLSI Syst., vol. 1, pp. 203–213, June 1993.

[3] Y. Ooi, A. Taniguchi, and S. Demura, “A 162 Mbits/s variable length decoding circuit using an adaptive tree search technique,” in Proc. IEEE Custom Integrated Circuits Conf., 1994, pp. 107–110.

[4] R. Hashemian, “Design and hardware implementation of a memory efficient Huffman decoding,” IEEE Trans. Consumer Electron., vol. 40, pp. 345–352, Aug. 1994.

[5] S.M. Lei and M.T. Sun, “An entropy coding system for digital HDTV applications,” IEEE Trans. Circuits Syst. Video Technol., vol. 1, pp. 147–155, Mar. 1991.

[6] S.F. Chang and D.G. Messerschmitt, “Designing a high-throughput VLC decoder, Part I-Concurrent VLSI architectures,“ IEEE Trans. Circuits Syst. for Video Tech., vol. 2, pp.187-196, June 1992; Also in: H.D. Lin and D.G. Messerschmitt, “Designing a high-throughput VLC decoder Part II—Parallel decoding methods,” IEEE Trans. Circuits Syst. Video Tech., vol. 2, pp. 197–206, June 1992.

[7] B.W.Y. Wei and T.H. Meng, “A parallel decoder of programmable Huffman codes,” IEEE Trans. Circuits Syst. Video Techn., vol. 5, pp.175–178, Apr. 1995.

[8] S.H. Cho, et. al., "A low power variable length decoder for MPEG-2 based on nonuniform fine-grain table partitioning," IEEE Trans. VLSI Syst., 7(2), pp.249-257, June 1999.

[9] S.W. Lee and I.C. Park, "A Low-Power Variable Length Decoder for MPEG-2 Based on Successive Decoding of Short Codewords," IEEE Trans. CAS-II: Analog and Digital Signal Processing, 50(2), Feb. 2003, pp. 73-82.

[10] L.Y. Liu, J.F. Wang, and J.Y. Lee, “CAM-based VLSI architecture for dynamic Huffman coding,” IEEE Trans. Consumer Electron., vol. 40, no. 3, pp. 282–289, Aug. 1994.

[11] C.T. Hsieh and S. P. Kim, “A Concurrent Memory-Efficient VLC Decoder for MPEG Applications,” IEEE Trans. Consumer Electron., vol.42, pp. 439–446, Aug. 1996.

[12] B.J. Shieh, et. al., "A New Approach of Group-Based VLC Codec System with Full Table Programmability," IEEE Trans. CAS for Video Tech., 11(2), Feb. 2001, pp. 210-221.

[13] J.-S. Wang, et al., “An AND type match-line scheme for energy efficient content addressable memories” IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers, Feb. 2005, pp. 464–465.