A LOW-POWER GROUP-BASED VLD DESIGN

Cheng-Hung Liu, Bai-Jue Shieh, and Chen-Yi Lee

Department of Electronics Engineering, National Chiao Tung University
1001, Ta Hsueh Road, Hsinchu, 300, Taiwan, R.O.C
Tel: +886-3-5712121 Ext.54238; Email: nina@si2lab.org

ABSTRACT

In this paper, a low power variable length code decoder with group-based scheme [10] is proposed. It can save at most 70% power dissipation while maintaining the same throughput rate as the conventional one. This proposal mainly consists of look-up table (LUT) partitioning and group-based memory storing, where the former is used to reduce the power wasted on looking-up unnecessary code-words due to a large LUT, especially those code-words with low probability; the latter is used to reduce the storage space of VLC tables and maintain the flexibility of our hardware decoder.

1. INTRODUCTION

Variable length coding that maps input source data onto codewords with variable length is an efficient method to minimize average code length. Compression is achieved by assigning short codewords to input symbols of high probability and long codewords to those of low probability. Variable length coding has been successfully used to relax the bit-rate requirements and storage spaces for many multimedia compression systems such as MPEG and H. 263. For example, a variable length code (VLC) is employed in MPEG-2 along with the discrete cosine transform (DCT), resulting in very good compression efficiency.

The most important objective in the early researches on variable length decoders (VLDs) is to achieve high throughput. There have been a lot of studies addressing high performance VLDs in [1-4], which can be classified into two groups: tree-based and parallel decoding approaches. The tree-based approach decodes input symbols bit-serially and is adopted by a preliminary VLD [3]. Although some improvements make it possible to decode more than one bit per cycle [4], the approach is not suitable for high performance applications such as MPEG-2 and HDTV, because high clock rate processing is inevitable. As opposed to the tree-based approach, the parallel decoding approach can decode one codeword per cycle regardless of its length. As an example, [1] and [2] proposed such a VLD that consists of two major blocks, a VLC detector and a look-up table (LUT).

Since early studies have focused only on high throughput VLDs, low-power VLDs have not been received much attention.

This trend is rapidly changing as the target of multimedia systems is moving toward portable applications. These systems highly demand low-power operations, and require low-power functional units. Although the VLD proposed in [2] and [10] is good for achieving high throughput, it is not optimized for low-power applications. Therefore, there have been considerable efforts to reduce power consumption, which can be classified into two categories. The first is to reduce the power of LUTs based on the fact reported in [2] that LUTs consume considerable power. A number of schemes such as prefix pre-decoding [5] and table partitioning [6] have been presented and have reduced the power of LUTs significantly. Second, the other activities have tried to reduce the power of a VLC detector, and proposed several schemes such as VLC detector sizing [6] and barrel shifter optimizing [7].

In this paper, we propose a low-power VLD that considers the probability distributions of codewords. Review of traditional group-based low-power algorithm is presented in Section 2. The proposed low-power VLD scheme is described in Section 3. The power modeling of the proposed low-power VLD is presented in Section 4. The simulation result is shown in Section 5. Finally, the conclusion is made in Section 6.

2. GROUP-BASED DECODING ALGORITHM

In [10], the conventional group-based VLC codec system is described. The codewords are partitioned into several groups according to the prefix of the codeword. Each group can be identified by a unique code named PCLC_mincode. PCLC_mincode is the codeword of a group with the smallest absolute value if we see the codeword as an unsigned binary number. However, the codec system is designed for simultaneously encoding and decoding VLCs. For decoding-only group-based VLC decoder, some memory content arrangement can be changed.

There is a change about how to store symbols in memory to save space. The rule for conventional VLC is that the members of the former group must be totally in front of those of the latter group, and the adapted rule for decoding only VLC decoder is that the first member (i.e. the member with the smallest PCLC_mincode) of the former group must be in front of that of the latter group. This appears to be interesting because some groups may be placed in the middle of another group just because of the adapted rule. An example is shown in Fig. 1.

The decoding algorithm is described as follows:

1. Do group searching.
2. Send group information.

*Work supported by the National Science Council of Taiwan, R.O.C., under Grant NSC 92-2220-E-009 -010-

3. Find the valid VLC_codeoffset, which is the codelength most significant bits of the result of subtracting the PCLC_mincode from the bitstream_num.
4. Extract the VLC_codeoffset operand, which has the same wordlength as the symbol address.
5. Calculate the decoded symbol address.
6. Fetch the decoded symbol.

Encoder + Decoder		Decoder only	
group	PCLC_codeword	group	PCLC_codeword
G0	00100100	G0	00100100
G0	00100101	G0	00100101
G0	00100110	G0	00100110
G0	00100111	G0	00100111
G1	00110000	G1	00110000
		G2	01100000
		G2	01110000
G1	00111100	G1	00111100
G2	01100000	G3	10000000
G2	01110000	G4	11000000
G3	10000000	G5	11100000
G4	11000000	G5	11101000
G5	11100000	G6	11110000
G5	11101000	G6	11110010
G6	11110000	G6	11110010
G6	11110010	G6	11110100
G6	11111000	G6	11111000

Fig. 1: Adapted Memory Arrangement of codeword

An example is shown as follows: (assume the input bitstream is 001111100110..., and table is shown as that in Fig. 2.)

group	symbol	PCLC_codeword	PCLC_code_num	Symbol address	PCLC_code_offset	Is PCLC_mincode
G0	S00	00100100	36	0	0	✓
G0	S01	00100101	37	1	1	
G0	S02	00100110	38	2	2	
G0	S03	00100111	39	3	3	
G1	S10	00110000	48	4	0	✓
G2	S20	01100000		5	0	✓
G2	S21	01110000		6	1	
G1	S11	00111100	56	7	3	
..	..	..	..	..	..	✓
..	..	..	..	..	..	..

Fig. 2: Example of Memory Arrangement of codeword

1. mincode1 (8'b00110000) < StreamNum (8'b00111110) < mincode2 (8'b01000000). The matching group: G1;
2. codelength = 6, mincode1 = 8'b00110000, base address = 5'b00100 = 4.
3. stream_num (8'b00111110) - mincode1 (8'b00110000) = 8'b00001110. The valid VLC_codeoffset = 6'b000011.
4. VLC_codeoffset = 5'b00011 = 3.
5. symbol_address = base_address (5'b00100) + VLC_codeoffset (5'b00011) = 5'b00111 = 7
5. symbol_memory[7] = S11;

3. LOW-POWER GROUP-BASED VLD DESIGN

Similar to the method of table partitioning in [5], the codewords with absolute values which are close to each other are clustered together and we don't have to store the repeated prefixes so as to reduce the memory requirement as well as power dissipation.

The postfix part is handled by VLC_codeoffset which is the difference between every codeword value in the group and the PCLC_mincode of the group. Due to systematic gathering of codewords, VLC_codeoffset is integrally continuous and can be regarded as memory address offset from the address storing the symbol of PCLC_mincode in the symbol memory. In another word, it is similar to the idea of the second cluster in [5].

Our proposed low-power scheme used in conventional group-based VLD is to separate codeword groups while taking probability distribution of every group, probability of successively appeared short VLC codewords, and memory partitioning of symbol LUT memory into consideration.

Fig. 3: Low-power VLD Design. (a) Low-power Adaptation of VLD Algorithm (b) Adapted Low-power Architecture

[9] indicates that 75% of short codewords are followed by another short codeword whose length is shorter than 7 bits. In addition, 90% of successive codewords are within 8 bits. It also tells us that it is worthwhile to add a cache, the power consumption of cache is less than that of the VLC detector, and the probability of cache hit is as high as 0.8 for even a small-sized cache containing only 8 entries. Not to overemphasize the effectiveness of cache, we only assume hit rate is 0.75, not 0.8 while cache is used.

Let us see the proposed algorithm shown in Fig. 3(a). As long as we want to decode a VLC, we first look it up in codeword cache. If it is found, then symbol value is retrieved. If it is not found, then we look it up in PCLC_mincode group 1 LUT. If it is found in group 1 LUT, then we retrieve symbol information from symbol table LUT 1. If it can't be found in group 1 LUT, then we look it up in PCLC_mincode group 2 LUT. Keep searching the rest of the LUTs sequentially until it is found in certain LUT afterward.

The conventional group-based VLD design is at the left-hand side of Fig. 3(b), and the proposed one is the right one. In Fig. 3(b), as mentioned before, a PLCP_mincode represents a group of symbols with similar codewords. We can see that adapted version VLD separates the conventional PLCP_mincode LUT into several parts, i.e. PLCP_mincode Group 1 LUT, PLCP_mincode Group 2 LUT, etc. PLCP_mincode Group 1 LUT contains the most frequent occurrences of PLCP_mincodes while PLCP_mincode Group 2 LUT consists of less frequently appeared PLCP_mincodes. As memory is the storage media of codewords and symbols, every PLCP_mincode in the same

PLCP_mincode group should be extended to the length of the longest PLCP_mincode to maintain regular memory access. Since maximum code length of Group 1 LUT is shorter than that of Group 2 LUT, we can shorten the word width of PLCP_mincode in Group 1 LUT as well as that of symbols in Symbol Table Memory 1. In the conventional design, there is only 1 piece of PLCP_mincode LUT memory as well as 1 block of Symbol table memory, more energy is consumed while decoding short VLC codeword due to altering the whole memory of the bit width of the longest PLCP_mincode of all PLCP_mincode.

4. POWER CONSUMPTION MODELING

Fig. 4: Power consumption of 0.18um process. (a) all used SRAM cells and (b) Register. (data from Synopsys Design Analyzer and Data Sheet)

Fig. 5: Codeword Distribution of MPEG2 Table 1 to Table 15

While cache is not taken into account (i.e. only consider the part of LUT), the energy dissipation can be modeled as follows:

$$E_{LUT, avg} = P_{r1}E_{H1} + (1 - P_{r1})E_{M1} + (1 - P_{r1})P_{r2}E_{H2} + (1 - P_{r1})(1 - P_{r2})E_{M2} + (1 - P_{r1})(1 - P_{r2})P_{r3}E_{H3} + (1 - P_{r1})(1 - P_{r2})(1 - P_{r3})E_{M3} + \dots + E_{overhead(n)} \quad (1)$$

where P_{ri} is the probability that table i is a hit (i.e., a match was found in table), n is the total number of decomposed tables, E_{Hi} is the energy consumption of table when there is a hit, E_{Mi} is the energy when there is a miss in LUT i , and

$E_{overhead(n)}$ is the energy consumed by the circuit overhead introduced by the table partitioning, such as conditional branches and clock generation. E_{Hi} is the energy used by PLCP_mincode group i LUT and symbol table LUT i ; E_{Mi} is only the energy used by PLCP_mincode group i LUT. The average energy consumption is the sum of the energy consumption weighed with the corresponding probability in the decomposed tables. To achieve minimum average energy, the first look-up table must consume the least energy while having the highest hit ratio. The hit ratio can be increased by allocating more codewords, but this will also increase the energy consumption of the first table.

The power consumption of 1-port SRAM cell with various configurations is shown in Fig. 4(a), and that of register cells is shown in Fig. 4(b). To reduce power-consumption, the original LUT has to be cut into several parts as sub LUTs, namely, LUT1, LUT2, LUT3, etc. The situation is like that shown in Fig. 5.

In Fig. 5, x-axis is codeword index. The index is obtained by sorting all the codewords appearing in MPEG2 VLC tables with the rule as follows: (assume we are sorting two different VLC codewords A and B)

1) If code length of A is smaller than that of B, A is put in the position with lower index.

2) If code lengths of A and B are the same, continue this step. A and B are now regarded as binary number. If value of A is smaller than B, A is located in the position with lower index.

As Fig. 5 shows, codewords with shorter length occurs much frequently than those with longer length do. As illustrated, we can easily see that probability distribution of tables is Laplacian-distributed; i.e. LUT1 has the highest hit rate while LUT1 also has the smallest size among all sub LUTs. Since number of LUT entries in group 1 LUT and symbol table LUT 1 is the smallest, as well as the word width of LUT 1 is the shortest, the energy dissipation of LUT 1 is smallest. This helps to design a low-power group-based VLD.

If cache is taken into account, the energy dissipation can be modeled as follows:

$$E_{LUT, avg} = P_{cache} E_{cache, hit} + (1 - P_{cache}) E_{cache, miss} + (1 - P_{cache}) [P_{r1}E_{H1} + (1 - P_{r1})E_{M1} + (1 - P_{r1})P_{r2}E_{H2} + (1 - P_{r1})(1 - P_{r2})E_{M2} + (1 - P_{r1})(1 - P_{r2})P_{r3}E_{H3} + (1 - P_{r1})(1 - P_{r2})(1 - P_{r3})E_{M3} + \dots] + E_{overhead(n)} \quad (2)$$

The definitions of P_{ri} , n , E_{Hi} , E_{Mi} , and $E_{overhead(n)}$ are the same as those in (1). $E_{cache, hit}$ is the energy consumed by cache when it is a hit in cache; $E_{cache, miss}$ is the energy dissipated by cache when it is a miss; $P_{r cache}$ is the hit ration of the cache. Now that $P_{r cache}$ is very high, as well as $E_{cache, miss}$ and $E_{cache, hit}$ are lower than E_{H1} , low power dissipation is achievable.

5. SIMULATION RESULT

According to the power model described, proper table partitioning is performed. The LUTs are partitioned into 6 parts at most because power improvement from 7-partition version to 6-partition version is about 0.1% of the worst power consumption, i.e. power consumption by un-partitioned LUT.

The comparison of n-part configuration is shown in Table 1 and Fig. 6. We take MPEG2 for instance. Fig.7 shows the VLC occurrence probability of every table in MPEG2 by summarizing decoded results of standard video bitstreams. We let the traditional design work under 4.78MHz clock to meet the decoding speed of MPEG2 MP@ML (30 frames/second, 720x576, 4:4:4 format). We operate our proposed design to work under 6.7MHz and the main purpose is to let the two designs have the same decoding throughput.

The calculated average power consumption is shown in Fig. 8, and we can see clearly the proposed method works well. It can be seen from Fig.8 that the proposed design can dissipate less energy (about 30.18% of the conventional design) while maintaining the same throughput rate as the conventional one.

n	1	2	3	4	5	6	7	8
%	100	34.80	32.87	32.30	31.90	31.61	31.49	31.39
n	9	10	11	12	13	14	15	16
%	31.32	31.27	31.23	31.20	31.14	31.13	31.123	31.121

Table 1: The power comparison of proposed Scheme with LUT partitioned into n parts

Fig. 6: The power comparison of proposed Scheme with LUT partitioned into n parts

Fig. 7: Probability Distribution of MPEG2 tables

Fig. 8: Power consumption of MPEG2 tables

6. CONCLUSION

Based on the proposed table portioning with consideration of probabilities of VLC groups, we are able to save at most 70% power dissipation of the conventional group-based VLD design

while maintaining the same throughput rate as the conventional one.

7. REFERENCES

- [1] M. T. Sun and S. M. Lei, "A parallel variable-length-code decoder for advanced television applications," *Proc. 3rd Int. Workshop HDTV*, 1989.
- [2] 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.
- [3] A. Mukherjee, N. Ranganathan, and M. Bassiouni, "Efficient VLSI designs for data transformations of tree-based codes," *IEEE Trans. Circuits Syst. II*, vol. 38, pp. 306-314, Mar. 1991.
- [4] Y. Ooi, A. Taniguchi, and S. Demura, "A 162 Mb/s variable length decoding circuit using an adaptive tree search technique," *Proc. IEEE Custom Integrated Circuits Conf.*, 1994, pp. 107-110.
- [5] S. B. Choi and M. H. Lee, "High speed pattern matching for a fast Huffman decoder," *IEEE Trans. Consumer Electron.*, vol. 41, pp. 97-103, Feb. 1995.
- [6] S. H. Cho, T. Xanthopoulos, and A. P. Chandrakasan, "A low power variable length decoder for MPEG-2 based on nonuniform fine-grain table partitioning," *IEEE Trans. VLSI Syst.*, vol. 7, pp. 249-257, Jun. 1999.
- [7] C. H. Lin and C.W. Jen, "Low power parallel Huffman decoding," *Electron. Lett.*, vol. 34, no. 3, pp. 240-241, Feb. 1998.
- [8] S. H. Cho, T. Xanthopoulos, and A. P. Chandrakasan, "An ultra low power variable length decoder for MPEG-2 exploiting codeword distribution," *IEEE Custom Integrated Circuits Conf.*, May 1998, pp.177-180.
- [9] Sung-Won Lee, and In-Cheol Park, "A Low-Power Variable Length Decoder for MPEG-2 Based on Successive Decoding of Short Codewords," *IEEE Trans. on Circuits Syst. Analog And Digital Signal Processing.*, vol.50, pp.73-82, Feb. 2003.
- [10] Bai-Jue Shie, Yew-San Lee, AND Chen-Yi Lee, "A new approach of group-gased VLC codec system with full table programmability," *IEEE Transactions on Circuits and Systems for Video Technology*, pp.210-221, vol. 10, issue 8, Dec. 2000