# A LOW-POWER VARIABLE LENGTH DECODER BASED ON SUCCESSIVE DECODING OF SHOFT CODEWORDS

*Sung-Won Lee and In-Cheol Park*
Department of Electrical Engineering and Computer Science
KAIST, 373-1 Gusung-dong Yusung-gu Taejon, Korea

## ABSTRACT

This paper presents a low-power variable length decoder exploiting the statistics of successive codewords. The decoder employs a small look-up table working as a fixed cache to reduce the number of activations of a variable length code detector where most power is consumed. The power simulation results estimated with Powermill show that 35% energy is reduced on the average compared with the previous low-power scheme.

## 1. INTRODUCTION

Variable length coding (VLC) that maps input source data into codewords of variable length is an efficient method of minimizing average code length. Compression is achieved by assigning short codewords into input symbols of high probability and long codewords to input symbols of low probability. VLC 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, VLC is used in MPEG-2 along with discrete cosine transform (DCT) and run-length coding in order to achieve very good compression efficiency.

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

Since earlier works have focused only on high throughput VLDs, low-power VLDs have not been received much attention until recently. This trend is rapidly changing as the target of the multimedia systems are moving toward portable applications. These systems highly demand low-power operations, and thus require low-power functional units including VLDs. Although the VLD proposed by Lei and Sun 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. First, many activities are focused on reducing the power of LUTs based on the fact reported in [2] that considerable power is consumed in LUTs. A number of schemes such as prefix predecoding [3] and table partitioning [4] have been presented and have reduced the power of LUTs significantly. Second, the other activities have tried to reduce the power of the VLC detector, and proposed several schemes such as VLC detector sizing [4] and barrel shifter optimizing [5]. All of these approaches assume that a codeword is independent of others, and do not consider the relation among codewords.

In this paper, we propose a new low-power VLD that considers the characteristics of successive codewords. The organization of this paper is as follows. Observations on the MPEG-2 source bit-streams and the parallel decoding architecture are presented in Section 2. The proposed low-power VLD architecture is described in Section 3. The implementation of the proposed low-power VLD is described in Section 4. Finally, conclusions are made in the last section.

## 2. OBSERVATIONS

The typical architecture of parallel decoding VLDs consists of a VLC detector and a set of LUTs. The VLC detector consists of an accumulator and a barrel shifter. The output of the barrel shifter is fed into the LUTs which convert a codeword to the corresponding symbol. In order to determine the position of the next codeword, the size of the decoded codeword is fed back to the accumulator which controls the barrel shifter. Usually, the barrel shifter has a 32-bit window and can align input data at a suitable point within one cycle time. The accumulator is consisted of a 4–bit adder and a 4-bit register. In contrast to the VLC detector, the LUTs contain hundreds of VLC information such as codewords and their code lengths and symbols. Therefore previous low power efforts are concentrated on the problem of reducing the LUT size and the average switching capacitance. Table partitioning is an example of these approaches. In [4], a low-power VLD was achieved using non-uniform fine grain table partitioning, which also reduced the size of LUTs that are most frequently accessed. As a result of table partitioning, the power dissipated in the VLC detector has become a significant portion of the total power.

To investigate the power consumption ratio of VLC detector to LUTs, the power dissipation is simulated for various conditions: The output size of the VLC detector is changed to one of 8, 12, and 16 bits, and the input size of LUTs varies from 2 to 15 bits. The target bit-stream considered in this paper is MPEG-2 DCT AC coefficient codes, which account for more than 80% of the whole variable length bit-stream. According to MPEG-2 standard , this DCT AC coefficients belong to Table B-14 and Table B-15 that have 114 entries and 113 entries, respectively.



0-7803-6685-9/01/$10.00©2001 IEEE

The VLC table is constructed from Table B-14 omitting some fixed length codes such as the DC coefficient and common escape codes. These fixed length code is not considered in the previous works [4] either. The parsed bit-streams are assumed to contain only variable length codes. The bit-streams are extracted from MPEG-2 video files using the modified MPEG-2 TM 5 code. The video sources used in experiment consist of MPEG-2 video conformance bit-streams and some MPEG-2 video clips.

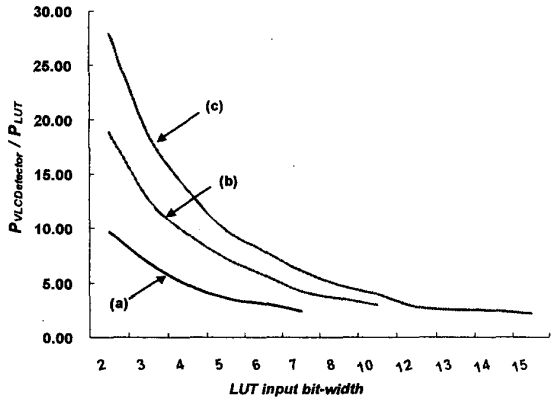


**Figure 1.** Plot of $P_{VLC\_detector}$ / $P_{LUT}$ for 8-bit, 12-bit, and 16-bit VLC detectors. (a) 8-bit. (b) 12-bit. (c) 16-bit.

Fig. 1 shows a plot of $P_{VLC\_detector}$ / $P_{LUT}$, where $P_{LUT}$ and $P_{VLC\_detector}$ represent the power dissipation of the LUT and the VLC detector for a single access, respectively. Power estimation is performed with PowerMill using a 0.35μm cell library. For every case, $P_{LUT}$ is much smaller than $P_{VLC\_detector}$.

It is clear that the power reduction using table partitioning is not expected to be significant, because the dominant power consumption block is not the LUT but the VLC detector. For example, the power consumption of the LUT that has 5 entries is at most one fifth with respect to that of the 8-bit VLC detector. Therefore, it is necessary to develop a new scheme that activates the VLC detector as minimal as possible

**Table 1.** Statistics of short codewords

| Codeword | Index | Length | Probability of occurrence |
|---|---|---|---|
| 10 | 1 | 2 | 0.1547 |
| 11s | 2 | 3 | 0.1954 |
| 011s | 3 | 4 | 0.1040 |
| 0100s | 4 | 5 | 0.0830 |
| 0101s | 5 | 5 | 0.0613 |
| 00101s | 6 | 6 | 0.0313 |
| 00110s | 7 | 6 | 0.0292 |
| 00111s | 8 | 6 | 0.0403 |

The probability of short codewords and prefixes are listed in Table 1. As can be seen from the table, about 78% of all codewords are limited to 7 bits, and the most frequently occurring codewords are "10", "11s", and "011s". In the case of typical VLC detectors whose output size is 8 bits or 16 bits, these codewords pass through the LUT as other codewords does, then activate the VLC detector. As mentioned above, the power dissipation of the VLC detector is 4 ~ 5 times as much as that of the LUT. Therefore, power saving can be achieved if two VLC codeword could be decoded by one activation of the VLC detector.

In order to confirm this idea, statistics of the two successive codewords is examined. In Table 2 showing the result, the codeword index indicates the preceding codewords and the successive codeword index represents the next codeword which follows the codeword indicated by codeword index. The result implies that almost short codewords are followed by less than 7-bit codewords in probability of 0.75. On the average, 90% of total successive codewords are within 8-bit long.

As the basic principle of variable length coding is to assign short codewords to frequent input symbols, the observation seems to be natural. Since the output size of a VLC detector is usually enough to cover two short codewords, this observation can be used to reduce the activation of a VLC detector by permitting one activation for two short codewords. This is very different from previous schemes in which a VLC detector is activated for every codeword.

**Table 2.** Statistics of two successive short codewords for MPEG-2 DCT AC coefficients

| Codeword index \ Successive codeword index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Sum |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.13 | 0.08 | 0.11 | 0.11 | 0.07 | 0.05 | 0.04 | 0.05 | 0.65 |
| 2 | 0.12 | 0.32 | 0.13 | 0.07 | 0.08 | 0.03 | 0.03 | 0.05 | 0.81 |
| 3 | 0.16 | 0.29 | 0.13 | 0.05 | 0.08 | 0.02 | 0.04 | 0.05 | 0.81 |
| 4 | 0.08 | 0.30 | 0.11 | 0.13 | 0.06 | 0.06 | 0.02 | 0.03 | 0.79 |
| 5 | 0.20 | 0.26 | 0.12 | 0.04 | 0.08 | 0.01 | 0.04 | 0.05 | 0.81 |
| 6 | 0.05 | 0.29 | 0.09 | 0.15 | 0.04 | 0.09 | 0.01 | 0.02 | 0.75 |
| 7 | 0.27 | 0.22 | 0.11 | 0.03 | 0.07 | 0.01 | 0.04 | 0.05 | 0.80 |
| 8 | 0.23 | 0.24 | 0.12 | 0.03 | 0.07 | 0.01 | 0.04 | 0.03 | 0.80 |

## 3. PROPOSED SCHEME

As shown in Fig. 2 and Fig. 3(b), the proposed scheme is to introduce a small-sized LUT called a cache next to the traditional LUT storing short codewords. The cache is accessed when a short codeword is detected in the LUT to eliminate the activation of a VLC detector that consumes much power. The proposed scheme works as follows. The codeword aligned in the VLC detector is decoded in the first LUT, LUT1. The most frequent codewords such as "10", "11s", and "011s" are located in LUT1. Once a symbol is found in LUT1, meaning that the codeword is short, a new codeword is searched in the next cycle at the cache, Cache1, without invoking the VLC detector to shift the bit stream. In the next cycle, the input latch of Cache1 is clocked to latch the output of the VLC detector. In the case that the VLC is hit in Cache1, the power required to activate the VLC detector is

saved. If not hit in Cache1, the power and the cycle needed to access the cache are wasteful. However, the cache power is much less than that of the VLC detector and the probability that the codeword is hit in the cache is as high as 0.8 for even a small-sized cache containing only 8 short codewords.

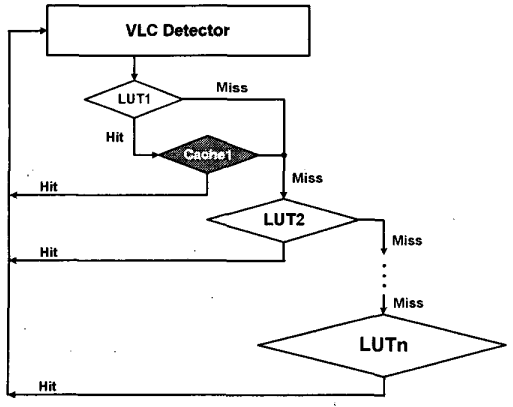


**Figure 2.** Proposed low-power scheme

Therefore we can reduce power on the average. In addition, letting the next codeword go directly to LUT2, instead of LUT1, can compensate the cycle penalty caused by the cache-miss. This can be achieved by making the cache contain all the short codewords of LUT1. We do not need to access LUT1, because if a codeword is not in the cache that satisfies the above property, it is guaranteed that the codeword is not in LUT1. Therefore, it is possible to save power without sacrificing performance. The proposed structure of VLD is briefly presented in Fig. 3 compared to the conventional VLD structure [4].

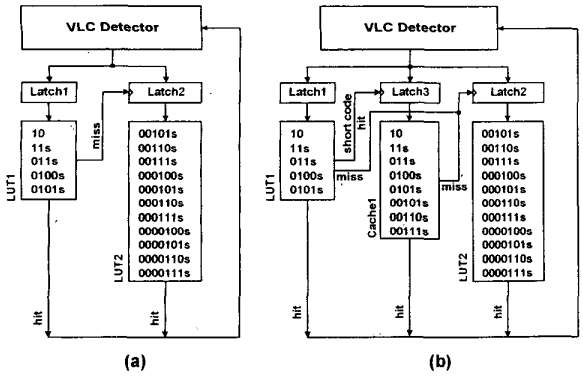


**Figure 3.** VLD architecture. (a) Conventional VLD [4]. (b) Proposed VLD.

In the proposed scheme, LUT1 and Cache1 are accessed sequentially. But we can imagine other configurations, because the output of the VLC detector is enough to cover two short codewords as mentioned before. One possible configuration is to enlarge LUT1 to generate two short codewords, and the other is to access LUT1 and Cache1 in parallel. Although these configurations are more useful in increasing throughput, the former results in a large-sized LUT and the latter activates the cache every cycle, resulting in more power consumption than the proposed scheme in our power simulation. Another disadvantage of the other configurations is that two symbols are decoded at a time, which makes the following hardware stages complex, as they have to process two symbols concurrently.

# 4. IMPLEMENTATION

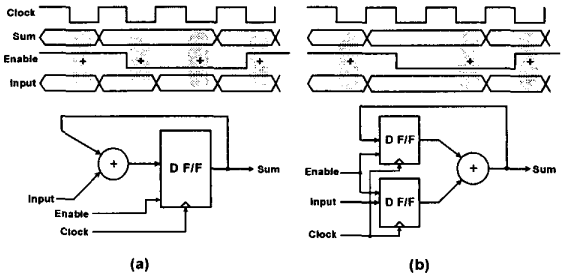


**Figure 4.** The structure of accumulator. (a) A typical structure which can not avoid an unintended add operation. (b) A modified architecture which has no an unintended add operation.

To validate the proposed cache scheme in practical designs, we implemented a low-power VLD that has a similar structure to that of [4], except that the proposed cache is introduced in the structure and the accumulator is modified to take into account the cache effect as shown in Fig. 4.

As the accumulator in [4] has a latch at the output side of the adder, the adder is activated whenever its inputs are changed. When we access the cache, activating the adder is not necessary and should be avoided to reduce power. For the purpose of preventing the unintended addition, two latches are placed at the input side of the adder.

The output size of VLC detectors determines throughput and power consumption of VLDs. VLC detectors that have small output size are desirable for low power, but the opposite cases that have large output size are preferred for high performance. Focusing on low-power, we apply the proposed scheme to the VLD architecture with an 8-bit VLC detector. In this case, LUT1 is resized to find an optimal solution.

The architecture of a VLD based on an 8-bit VLC detector is shown in Fig. 5, which has two caches separated to further reduce the cache power; Cache1 and Cache2. If the size of a short codeword found in LUT1 is 2 bits, Cache1 is accessed. Cache2 is accessed for a 3-bit codeword. The other blocks such as LUT2, BLK1, and BLK2 have the same function as those of the VLD structure presented in [4].

Fig. 6 shows the power consumption of the 8-bit VLD based on the modified table partitioning algorithm and caches. Compared to the VLD without caches, maximally 27.4% power is reduced at the cost of negligible area overhead. And the power dissipation of the VLC detector is also plotted in Fig. 6(b). The caches and modified accumulator configuration are effective in reducing overall power. From the energy consumption result plotted in Fig. 6(d), LUT1 is determined to have 8 entries.

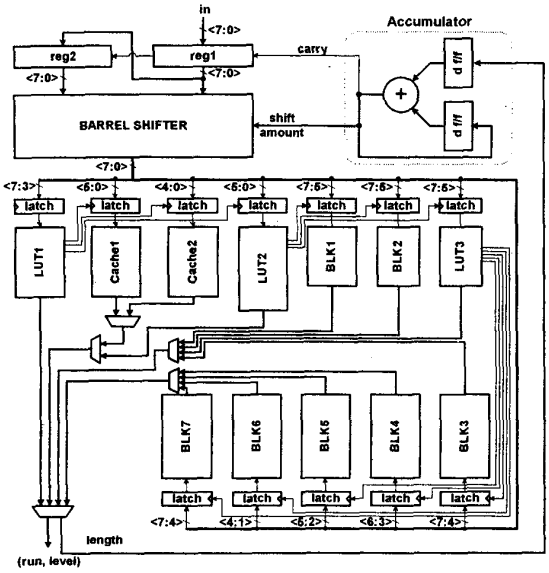


**Figure 5.** Low power VLD architecture with an 8-bit VLC detector

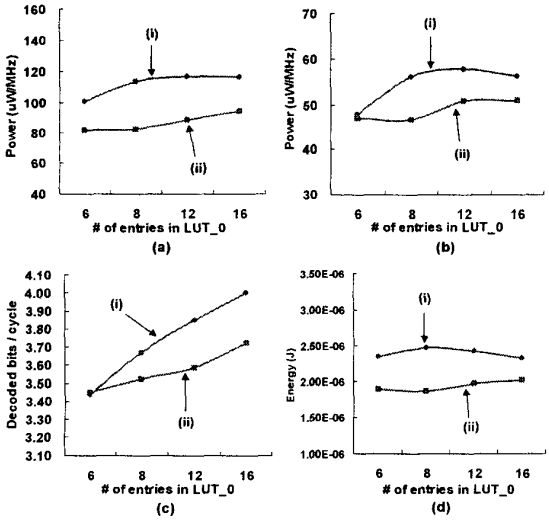


**Figure 6.** Simulation results for the proposed VLD having an 8-bit VLC detector; where only table partitioning is applied in (i), and the proposed cache is applied in (ii). (a) Power consumption of the VLD. (b) Power consumption of the 8-bit VLC detector only. (c) Throughput of the VLD. (d) Energy consumption (decoding 10KByte of encoded data).

The power consumption of the proposed VLD is compared to the previous VLD [4] that has been known as the best low-power structure. Fig. 7 shows the power consumption of each VLD with respect to 20 bit-streams. In every case, the proposed VLD is superior to the pervious one in the context of power consumption.

The power simulation results are summarized in Table 3. Besides the higher throughput, approximately 30 percent of power reduction is achieved by employing the proposed cache scheme.

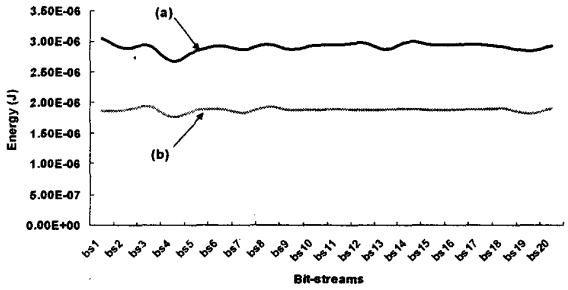


**Figure 7.** Energy consumption comparison for 20 bit streams. (a) Proposed VLD. (b) Previous VLD [4].

**Table 3.** Power comparison

| | Power consumption (μW / MHz) | Throughput (bits / cycle) | Energy for decoding 10KBytes (μJ) |
|---|---|---|---|
| Previous VLD [4] | 114.52 | 3.148 | 2.911 |
| Proposed VLD | 82.66 | 3.530 | 1.873 |

## 5. CONCLUSION

In this paper, a low-power VLD is presented based on the statistics of two successive codewords. It employs small sized LUTs working as fixed caches to reduce the number of activations of a VLC detector which consumes much power. The overall power is significantly reduced at the expense of a little circuit overhead. Intensive simulation results show that the proposed scheme can reduce energy by 35% on the average without sacrificing throughput.

## 6. REFERENCES


[1] S. F. Chang and D. G. Messerschmitt, "Designing high-throughput VLC decoder Part I – Concurrent VLSI architectures." IEEE Trans. Circuits and Systems for Video Technology, vol. 2, no. 2, pp. 187-196, June 1992.

[2] M. T. Sun and S. M. Lei, "A parallel variable-length-code decoder for advanced television applications," Proc. 3rd International Workshop on HDTV, Torino, Italy, Aug. 1989.

[3] S. B. Choi and M. H. Lee, "High speed pattern matching for a fast Huffman decoder," IEEE Trans. Consumer Electronics, vol. 41, pp. 97-103, Feb. 1995.

[4] S. H. Cho, T. Xanthopoulos, and A. P. Chandrakasan, "An ultra low power variable length decoder for MPEG-2 based on nonuniform fine-grain table partitioning," IEEE Trans. VLSI Systems, vol. 7, no. 2, pp. 249-257, June 1999.

[5] C. H. Lin and C. W. Jen, "Low power parallel Huffman decoding," IEE Electronics Letters, vol. 34, no. 3, pp. 240-241, Feb. 1998.