A Low-Power Variable Length Decoder for MPEG-2 Based on Successive Decoding of Short Codewords

Sung-Won Lee, *Student Member, IEEE,* and In-Cheol Park, *Senior Member, IEEE*

***Abstract*—This paper presents a low-power variable length decoder exploiting the statistics of successive codewords. The decoder employs small look-up tables working as fixed caches to reduce the number of activations of a variable length code detector where considerable power is consumed. The power simulation results estimated using PowerMill show that 35% energy is reduced on the average compared to the previous low-power scheme.**

***Index Terms*—Huffman, low power, MPEG-2, variable length code, variable length decoder.**

I. Introduction

VARIABLE length coding that maps input source data onto codewords with variable length is an efficient method to minimize average code length [1]. 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 [2]–[6], 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 [5]. Although some improvements make it possible to decode more than one bit per cycle [6], 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, Lei and Sun proposed such a VLD that consists of two major blocks, a VLC detector and a look-up table (LUT) [3], [4].

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, thus require low-power functional units. 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. The first is to reduce the power of LUTs based on the fact reported in [4] that LUTs consume considerable power. A number of schemes such as prefix predecoding [7] and table partitioning [8] 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 [8] and barrel shifter optimizing [9]. 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 MPEG-2 source bit-streams and the parallel decoding architecture are presented in Section II. The proposed low-power VLD scheme is described in Section III. The implementation of the proposed low-power VLD is described in Section IV. Finally, conclusions are made in Section V.

II. Observations

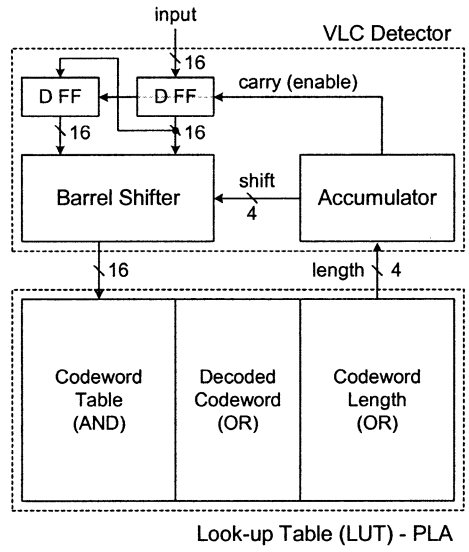
Fig. 1(a) shows the parallel decoding VLD architecture that is composed of a VLC detector and a LUT. The VLC detector is further decomposed into a barrel shifter and an accumulator. Since the longest codeword is assumed to be 16 bits long, the output size of the barrel shifter is set to 16 bits. To determine the shift amount of the barrel shifter, the accumulator adds the length of the previous codeword to indicate the location of the next codeword. If the shift amount exceeds 15 (the largest number with four binary digits), the two D flip-flops (FFs) update the input data under the control of the carry output of the accumulator, which is equivalent to a 16-bit shift. The LUT contains approximately 100 entries each of which represents a decoded codeword and a code length. Each entry is arranged by treating a codeword as an address. For this reason, the LUT is usually implemented by using PLA instead of ROM to achieve good area efficiency. As a result, the parallel decoding VLD can decode one codeword in a cycle.

In the parallel decoding VLD, the average energy consumed for decoding a codeword can be modeled by the following equ-

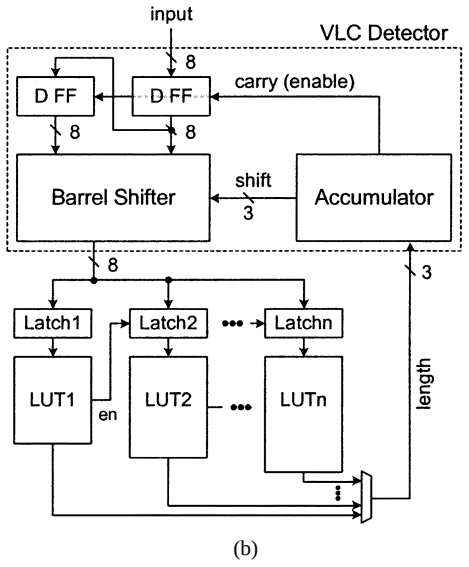
Manuscript received July 12, 2000; revised December 11, 2002. This work was supported in part by the Korea Science and Engineering Foundation through the Micro Information Communication and Remote Object-oriented Systems Research Center (MICROS) and in part by the Integrated Circuit (IC) Design Education Center (IDEC). This paper was recommended by Associate Editor Y. Wang.

The authors are with Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology (KAIST), Daejeon 305-701, Korea (e-mail: swlee@ieee.org, icpark@ee.kaist.ac.kr).

Digital Object Identifier 10.1109/TCSII.2003.808893



(a)



(b)

Fig. 1. VLD architecture: (a) parallel decoding architecture [3] and (b) M-way partitioned LUT architecture [8].

ation [8]:

$$\overline{E_{\rm VLD}} = \overline{E_{\rm LUT}} + \overline{E_{\rm VD}} = \sum_{i=1}^{n} P_{\rm cwi} \cdot E_{\rm cwi} + E(Y_b = 16) \quad (1)$$

where $\overline{x}$ denotes the average of x, $P_{\rm cwi}$ is the probability that codeword i occurs, $E_{\rm cwi}$ is the energy required to decode codeword i, n is the total number of codewords in the LUT, and $E(Y_b)$ is the energy consumption of the VLC detector whose size is Y_b. As there is one large LUT that is switched every cycle, $E_{\rm cwi}$ is little related to $P_{\rm cwi}$ ($E_{\rm cwi} \approx E_{\rm cw}$, $\forall\, i$, then $\sum P_{\rm cwi} E_{\rm cwi} = E_{\rm cw}$).

Although the large LUT is a key to achieving high throughput, it is not suitable for achieving low power. Due to this fact, the follow-up studies have focused on reducing LUT power consumption. For example, the latest low-power VLD scheme [8] uses M-way LUT partitioning to exploit the codeword occurring probability, as shown in Fig. 1(b). A single LUT is partitioned into a number of nonuniformly sized LUTs with considering the energy consumption and the occurring frequency. In this case, $E_{\rm cwi}$ depends on the size of the table containing codeword i. Low power is achieved by making $E_{\rm cwi}$ with high $P_{\rm cwi}$ small. The size of the VLC detector is changed from 16 to 8 bits, because the codewords in the most frequently accessed LUT are shorter than 8 bits. On the other hand, the throughput is lowered to 0.6 codewords per cycle. For this scheme, the average energy consumption per codeword is modeled below [8]

$$\begin{aligned}\overline{E_{\rm LUT}} =& Pr_1 E_{H1} + Pr_2 (E_{H2} + E_{M1}) + \cdots \\ &+ Pr_n \left(E_{Hn} + \sum_{i=1}^{n-1} E_{Mi} \right) + E_{\rm overhead}(n) \\ \overline{E_{\rm VD}} =& [P_{\rm cr}(0, Y_b = 8) + 2P_{\rm cr}(Y_b, 2Y_b) + \cdots \\ &+ mP_{\rm cr}((m-1)Y_b, mY_b)]\, E(Y_b) \end{aligned} \quad (2)$$

where Pr_i is the probability that LUT i is hit, n is the number of LUTs, E_{Hi} is the energy consumption of LUT i when there is a hit, E_{Mi} is the energy required for a miss, $E_{\rm overehead}(n)$ is the energy consumed by the circuit additionally required for n LUTs, $P_{\rm cr}(x, y)$ is the sum of the occurring probabilities of codewords whose lengths are greater than x and less than or equal to y, and m is the smallest integer not less than the maximum code length $L_{\max}$ divided by Y_b.

In (2), $\overline{E_{\rm VD}}$ is valid only if the VLC detector is independent of LUTs as in [10]. As the VLC detector is related to LUTs in [8], $\overline{E_{\rm VD}}$ in (2) underestimates the energy consumption. Let us assume a codeword whose length is less than Y_b. If the codeword is not in LUT1, it takes two cycles to decode. Although the accumulator does not have to operate during the first wait cycle, it consumes some amount of energy. The input change of the adder (3-bit adder for this case) leads to energy consumption. As the adder is as complex as LUT1, it consumes as much energy as LUT1. Note that the energy consumption during the wait cycle, $E_{\rm VD,wait}$ is smaller than $E(Y_b)$, because the accumulator output remains unchanged and there are no switching activities in the barrel shifter. Considering this claim, $\overline{E_{\rm VD}}$ is modified as follows:

$$\begin{aligned}\overline{E_{\rm VD}} =& \left[P_{\rm cr}(0, Y_b) \frac{Pr_1 + wPr_2}{Pr_1 + Pr_2} \right. \\ &\left. + P_{\rm cr}(Y_b, 2Y_b) \frac{(w+1)Pr_3 + 2wPr_4}{Pr_3 + Pr_4} + \cdots \right] E(Y_b) \end{aligned} \quad (3)$$

where $w = 1 + E_{\rm VD,wait}/E(Y_b) \approx 1 + E_{\rm adder}/E(Y_b)$. It is assumed that LUT1 and LUT2 have all codewords whose lengths are shorter than or equal to Y_b, i.e., $P_{\rm cr}(0, Y_b) = Pr_1 + Pr_2$, and LUT3 and LUT4 have the codewords whose lengths are longer than Y_b and shorter than or equal to $2Y_b$, i.e., $P_{\rm cr}(Y_b, 2Y_b) = Pr_3 + Pr_4$. For example, wPr_2 in the first term in (3) represents that it takes one $E_{\rm adder}$ (wait cycle) and one $E(Y_b)$ to decode a codeword in LUT2 and $(w+1)Pr_3$ in the second term in (3) represents that it takes one $E_{\rm adder}$ (wait cycle) and two $E(Y_b)$ to decode a codeword in LUT3. As $P_{\rm cr}(0, Y_b)$ is the largest of all, the first term in (3) greatly influences the overall VLC detector energy consumption. To achieve low energy consumption, therefore, we have to make $Pr_1 + wPr_2$ smaller and w equal to 1, which becomes more effective if $E(Y_b)$ is larger than E_{H1}.

TABLE I
LUT1 AND VLC DETECTOR POWER CONSUMPTION FOR VARIOUS CONFIGURATIONS

VLC detector size	LUT1 input bit-width	Number of entries in LUT1	LUT1 power (μW/MHz)	VLC detector power (μW/MHz)
8-bit	2-bit	2	4.6267	45.0204
	3-bit	3	6.0504	41.3111
	4-bit	5	9.7894	47.5852
	5-bit	8	13.8398	50.2851
	6-bit	12	17.4179	54.5104
	7-bit	16	21.1989	51.2981
12-bit	2-bit	2	4.7149	89.0507
	3-bit	3	5.9654	75.4895
	4-bit	5	10.1064	95.9392
	5-bit	8	14.5134	104.7494
	6-bit	12	17.9466	103.4029
	7-bit	16	23.2681	99.9795
	8-bit	24	31.7988	114.5757
	10-bit	32	38.1060	112.9218
16-bit	2-bit	2	4.7811	133.4916
	3-bit	3	5.8610	108.9769
	4-bit	5	10.2435	139.2018
	5-bit	8	14.3227	140.4556
	6-bit	12	18.1568	144.0675
	7-bit	16	23.6627	145.4009
	8-bit	24	32.1458	156.2268
	10-bit	32	38.9393	155.6910
	12-bit	48	57.5820	158.5675
	13-bit	64	67.6567	172.3900
	14-bit	80	68.4921	166.9540
	15-bit	96	73.4281	163.7231

To compare $E(Y_b)$ with E_{H1}, a number of power simulations are performed for various configurations. As $E(Y_b)$ and E_{H1} represent energy consumption during one cycle, they are closely related to the average power consumption ($E(Y_b) \cong P_{\mathrm{VD}} \times t_{\mathrm{cycle}}$, $E_{H1} \cong P_{\mathrm{LUT1}} \times t_{\mathrm{cycle}}$). Based on the M-way partitioned LUT architecture shown in Fig. 1(b), six different LUT1s whose input size ranges from 2 to 7 bits are incorporated with an 8-bit VLC detector, eight different LUT1s (2- to 10-bits long) are with a 12-bit VLC detector, and twelve different LUT1s (2 bits to 15-bits long) are with a 16-bit VLC detector. The VLC used in the experiment is the MPEG-2 DCT AC coefficients that occupy more than 80% of the whole VLC bit-stream. The VLC tables are constructed from Table-B14 in the MPEG-2 standard [11] while omitting some codewords of fixed size such as the DC coefficient and common escape codes. Note that these codewords are not considered in the previous work [8] either. In this experiment, the codewords that can be decoded in LUT1 are considered. The VLD is described in Verilog, synthesized using Synopsys Design Compiler with a 0.35-μm standard cell library [12], converted into spice netlists, and then simulated using PowerMill at the typical operating condition (3.3 V, 25°C). The power consumption results are collected using the hierarchical power simulation supported by PowerMill, as shown in Table I.

The results show that the average power consumption of LUT1, $\overline{P_{\mathrm{LUT1}}}$, increases as the input bit-width increases, while that of the VLC detector, $\overline{P_{\mathrm{VD}}}$, varies a little (within $\pm 10\%$ of the median) and is not monotonically proportional to the input size. For example, let us consider a VLD with a 16-bit VLC detector and a 15-bit LUT1. From the results in Table I, we expect that the power is reduced by approximately 60-μW/MHz if the M-way partitioned LUT scheme results in a 5-bit LUT1, and approximately 120-μW/MHz if the reduced VLC detector scheme [8] results in a 8-bit VLC detector.

As the VLC detector consumes more power than LUT1, power optimization focused on the VLC detector achieves good results. For a VLD that has an 8-bit VLC detector, further power reduction can be achieved by optimizing the VLC detector power consumption, not the LUT1 power consumption. Since LUT1 optimized by the low-power schemes [7], [8] has input bit-width of 4 or 5 bits, it consumes approximately one fifth of the VLC detector power. Consequently, a new low-power scheme must be developed to activate the VLC detector as minimal as possible.

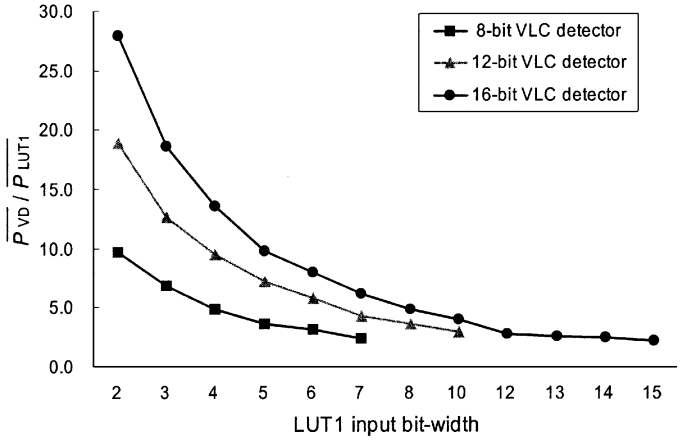


Fig. 2. Plot of $\overline{P_{\rm VD}}/\overline{P_{\rm LUT1}}$ for 8-bit, 12-bit, and 16-bit VLC detectors.

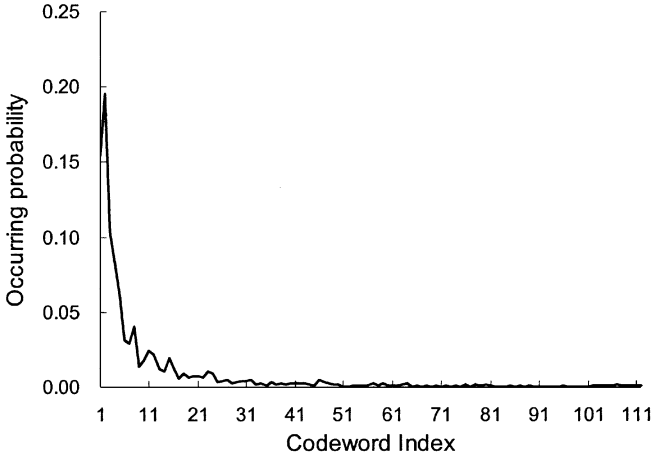


Fig. 3. Distribution of codeword occurring probability.

TABLE II
STATISTICS OF SHORT CODEWORDS

Codeword	Index	Length	Occurring probability
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

† s denotes a sign bit.

TABLE III
STATISTICS ON TWO SUCCESSIVE SHORT CODEWORDS FOR MPEG-2 DCT AC COEFFICIENTS

Short codeword (index)	Occurring probability of Successive codeword (by index)								
	1	2	3	4	5	6	7	8	Sum
10 (1)	0.13	0.08	0.11	0.11	0.07	0.05	0.04	0.05	0.65
11s (2)	0.12	0.32	0.13	0.07	0.08	0.03	0.03	0.05	0.81
011s (3)	0.16	0.29	0.13	0.05	0.08	0.02	0.04	0.05	0.81
0100s (4)	0.08	0.30	0.11	0.13	0.06	0.06	0.02	0.03	0.79
0101s (5)	0.20	0.26	0.12	0.04	0.08	0.01	0.04	0.05	0.81
00101s (6)	0.05	0.29	0.09	0.15	0.04	0.09	0.01	0.02	0.75
00110s (7)	0.27	0.22	0.11	0.03	0.07	0.01	0.04	0.05	0.80
00111s (8)	0.23	0.24	0.12	0.03	0.07	0.01	0.04	0.03	0.80

The ratio of $\overline{P_{\rm VD}}$ to $\overline{P_{\rm LUT1}}$ is plotted in Fig. 2 with varying the LUT1 input bit-width and the VLC detector size, in which $\overline{P_{\rm LUT1}}$ is always smaller than $\overline{P_{\rm VD}}$, $\overline{P_{\rm VD}}/\overline{P_{\rm LUT1}} > 1$. For a low-power VLD that has an 8-bit VLC detector and a 4-bit wide LUT1, the power consumption ratio is about five, which indicates that further optimization on the VLC detector can lead to considerable power reduction.

In order to develop a new low-power VLD scheme, we first examined the average occurring probability of each codeword contained in [11, Table-B14]. The reference implementation of the MPEG-2 standard [11], called MPEG-2 TM 5 was modified to store all the codewords encountered in running the program into a file. Using the MPEG-2 video conformance bit-streams, we generated 20 files each of which was approximately 100 kB. Based on these files, the average occurring probability was investigated, as shown in Fig. 3. As short codewords are closely related to the efficiency of low-power schemes, the probability has to be determined precisely. The average occurring probabilities of short codewords such as "10" and "11s" are as high as 0.15, which is similar to the previous results [8], [10].

The occurring probabilities of the short codewords are listed in Table II. Approximately 70% of the codewords have code lengths of shorter than 7 bits, and the most frequently occurring codewords are "10," "11s," and "011s." Regardless of the size of a VLC detector, these codewords activate the VLC detector and LUT1 as other codewords do. As the power dissipation of the VLC detector is larger than that of LUT1, considerable power saving can be achieved if two short codewords are decoded by activating the VLC detector only once. It is equivalent to lowering Pr_1 in the numerator of the first term in (3).

To validate this idea, the statistics on two successive short codewords was examined using the same files used in the previous experiment. Table III shows the result. The short codeword (index) indicates a preceding codeword and the occurring probabilities of successive short codewords are listed according to the codeword index. The result 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. The probability that two short codewords are located in one VLC detector window is high enough to validate the idea.

Since the basic principle of variable length coding is to assign short codewords to frequent input symbols, the observation result seems to be straightforward. The size of a VLC detector is enough to have two short codewords, even if an 8-bit VLC detector is considered. Therefore, it can be used to reduce the number of VLC detector activations by decoding two successive short codewords for a VLC detector activation. This is a major point different from the previous low-power schemes in which the VLC detector is always activated for each codeword.

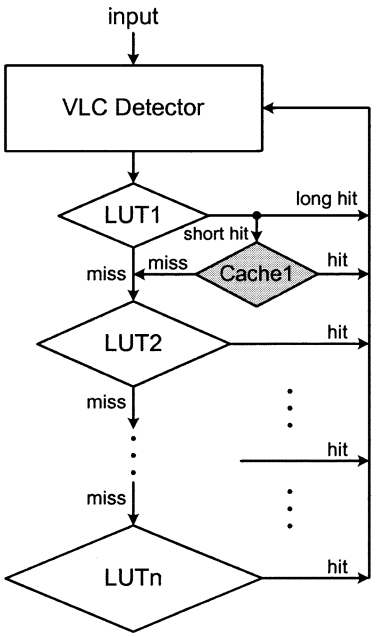


Fig. 4. Proposed low-power scheme.

III. Proposed Scheme

As described in Section II, a low-power VLD can be achieved by reducing the number of VLC detector activations based on the successive short codeword statistics. For this purpose, an additional LUT is introduced. The LUT called Cache1 is located between LUT1 and LUT2 as shown in Fig. 4, and accessed in the sequel of LUT1. During the next cycle, the output of the VLC detector is shifted by the length of the short codeword and then supplied to Cache1. This shift can be implemented by using a latch that points a different part of the VLC detector output. If a codeword is hit in Cache1, the power to be consumed in the VLC detector can be saved, because the VLC detector is not activated. Additional caches may be employed for the purpose of further power optimization.

The proposed scheme works as follows. The codeword aligned in the VLC detector is decoded in LUT1. The most frequent codewords such as "10," "11s," and "011s" are located in LUT1, where s$\in \{0,1\}$. Once a target symbol is found in LUT1, a new codeword is then searched in Cache1 without invoking the VLC detector to align the VLC stream. In the next cycle, the input latch of Cache1 is clocked to latch the output of the VLC detector. In the case that a codeword is hit in Cache1, the energy required to activate the VLC detector is saved. Otherwise, the energy and the cycle time needed to access Cache1 are wasteful. However, the power consumption of Cache1 is much less than that of the VLC detector and the probability that a codeword is hit in Cache1 is as high as 0.8 for even a small-sized cache containing only 8 short codewords. Therefore we can reduce the average power of the VLC detector. In addition, letting the next codeword go directly to LUT2, instead of LUT1, can compensate the cycle penalty caused by a cache miss. This can be achieved by making Cache1 contain all the short codewords of LUT1. If a codeword is not in Cache1 that satisfies the above property, it is guaranteed that the codeword is not in LUT1. As we can skip the access to LUT1, it is possible to save power without sacrificing performance. The proposed VLD architecture is briefly presented in Fig. 5 compared to the conventional one [8].

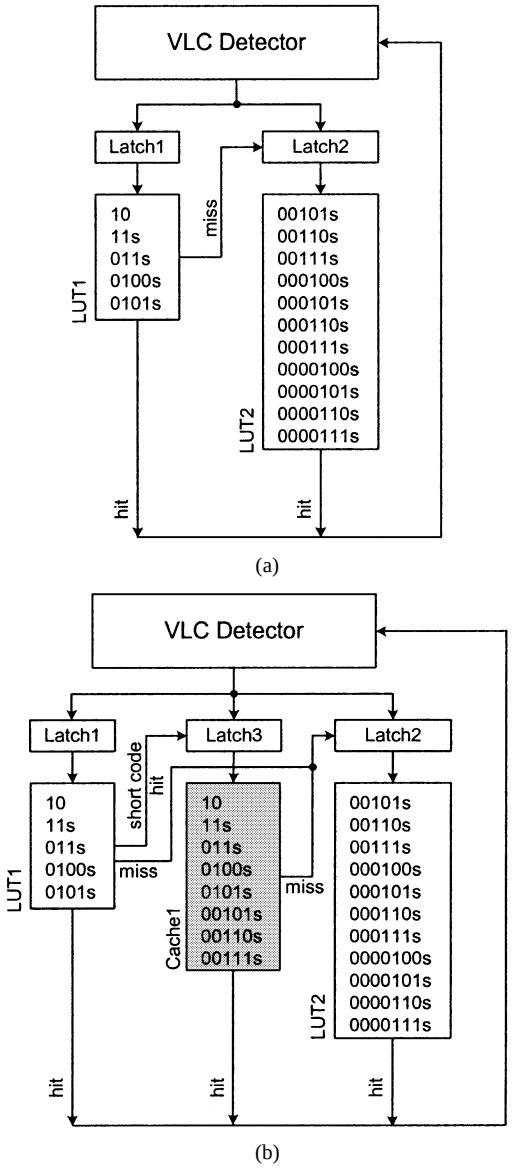


Fig. 5. VLD architecture: (a) previous low-power VLD [8] and (b) proposed low-power cache-equipped VLD.

In the proposed scheme, LUT1 and Cache1 are accessed sequentially. However, we can imagine other configurations, because the output size 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 larger LUT, and the latter activates Cache1 every cycle, resulting in more energy consumption than the proposed scheme in our power simulation. Another disadvantage of these configurations is that two symbols are decoded at a time. This makes the following hardware stages complex, as they have to process two symbols concurrently.

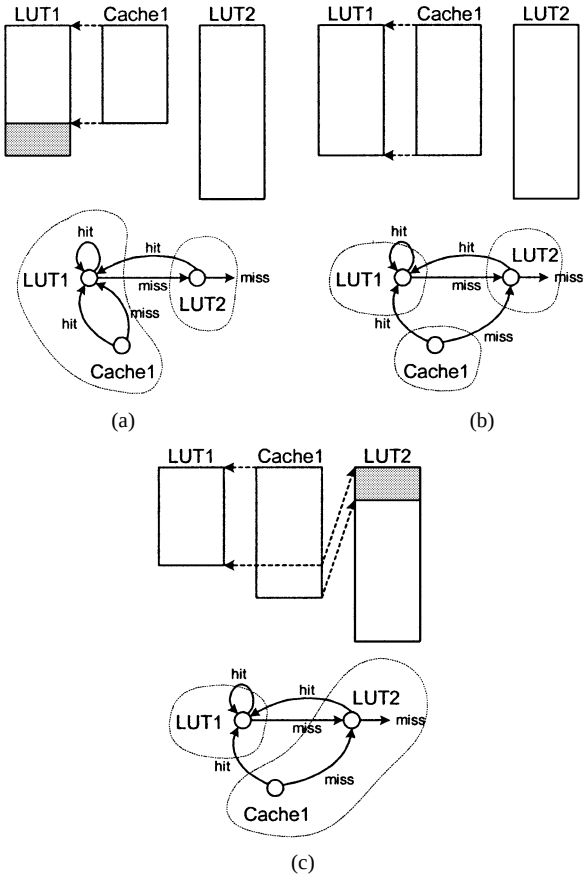


Fig. 6. Three cases to compute the average energy consumption per codeword for cache-equipped VLDs. (a) $Pr_C < Pr_1$, (b) $Pr_C = Pr_1$, (c) $Pr_C > Pr_1$.

The average energy consumption per codeword in a cache-equipped VLD is modeled by the following equation:

$$\overline{E_{c\mathrm{VLD}}} = \overline{E_{c\mathrm{LUT}}} + \overline{E_{c\mathrm{VD}}} \tag{4}$$

where c indicates that caches are employed. We assume that one cache (Cache1) is employed and $P_{\mathrm{cr}}(0, Y_b) = Pr_1 + Pr_2$ for the sake of simplicity. From this, three cases are considered separately.

Case 1) $Pr_C < Pr_1$. This is the case that all the entries in Cache1 are included in LUT1, but both are not the same as illustrated in Fig. 6(a). The average energy consumption to decode a codeword in LUT1 becomes $(1 - Ps)Pr_1E_{H1} + Ps(Pr_1 - Pr_C)E_{MC}$, where Ps denotes the probability that a short codeword enables the cache in the next cycle, Pr_C is the probability that Cache1 is hit, and E_{MC} is the energy consumption of Cache1 for a miss. Since Cache1 is smaller than LUT1, the term, $Ps(Pr_1 - Pr_C)E_{MC}$, is introduced to account for the case that some codewords in LUT1 lead to a miss in Cache1. Then, the energy consumption of the accompanying VLC detector is $((1 - Ps)Pr_1 + wPs(Pr_1 - Pr_C))E(Y_b)$. By the similar fashion, the average energy consumption in Cache1 and LUT2 are represented as $PsPr_CE_{HC}$ and $Pr_2(E_{H2} + E_{M1})$, and the corresponding VLC detector energy consumptions are $(w - 1)PsPr_CE(Y_b)$ and $wPr_2E(Y_b)$, respectively, where E_{HC} is the energy consumption of Cache1 for a hit. Then, $\overline{E_{c\mathrm{LUT}}}$ and $\overline{E_{c\mathrm{VD}}}$ are formulated as in (5) and (6), shown at the bottom of the page, where, we assume $E_{\mathrm{overhead}}(n + 1) \cong E_{\mathrm{overhead}}(n)$. Let Pr_C be xPr_1, $x < 1$, and we assume $E_{HC} = yE_{H1}$, $y < 1$ and $E_{MC} \cong E_{HC}$. Then, we have $(\overline{E_{c\mathrm{LUT}}} = \overline{E_{\mathrm{LUT}}} - (1 - y)PsPr_1E_{H1}) < \overline{E_{\mathrm{LUT}}}$, which indicates the average energy consumption of the LUTs is reduced in the cache-equipped VLD. Applying the same assumption to the numerator term in (6), we have a simplified numerator, $Pr_1 + (w - 1 - x)PsPr_1 + wPr_2$. If $x > (w - 1)$, the energy is reduced for the VLC detector as well.

Case 2) $Pr_C = Pr_1$. This is the case that Cache1 and LUT1 are the same. As shown in Fig. 6(b), the three LUTs can be examined separately. Then, $\overline{E_{c\mathrm{LUT}}}$ and $\overline{E_{c\mathrm{VD}}}$ are formulated as in (7) and (8), shown at the bottom of the next page. The average energy consumption of the LUTs remains unchanged after Cache1 is incorporated into the VLD, $\overline{E_{c\mathrm{VD}}}$ and can be reduced by $(w - 2)PsPr_1E(Y_b)$.

Case 3) $Pr_C > Pr_1$. This is the case that all the entries in LUT1 are included in Cache1, and some entries in LUT2 are also included in Cache1 as shown in Fig. 6(c). The average energy consumptions to decode a codeword in LUT1 and Cache1 become $(1 - Ps)Pr_1E_{H1}$ and $PsPr_1E_{HC} + Ps(Pr_C - Pr_1)E_{HC} = PsPr_CE_{HC}$, and the corresponding VLC detector energy consumptions are $(1 - Ps)Pr_1E(Y_b)$ and

$$\begin{aligned}\overline{E_{c\mathrm{LUT}}} &= (1 - Ps)\,Pr_1E_{H1} + PsPr_CE_{HC} + Ps\,(Pr_1 - Pr_C)\,E_{MC} + Pr_2\,(E_{H2} + E_{M1}) + \cdots \\ &\quad + Pr_n\left(E_{Hn} + \sum_{i=1}^{n-1} E_{Mi}\right) + E_{\mathrm{overhead}}\,(n+1) \\ &\cong \overline{E_{\mathrm{LUT}}} - PsPr_1E_{H1} + PsPr_CE_{HC} + Ps\,(Pr_1 - Pr_C)\,E_{MC}\end{aligned} \tag{5}$$

$$\overline{E_{c\mathrm{VD}}} = \left[P_{\mathrm{cr}}(0, Y_b)\frac{(1 - Ps)\,Pr_1 + (w-1)\,PsPr_C + wPs\,(Pr_1 - Pr_C) + wPr_2}{Pr_1 + Pr_2} + \cdots\right] E\,(Y_b)\,. \tag{6}$$

$(w-1)PsPr_CE(Y_b)$, respectively. The average energy consumption in LUT2 is represented as $(1-Ps)Pr_2(E_{H2}+E_{M1})+Ps(Pr_1+Pr_2-Pr_C)(E_{H2}+E_{MC})$, where the first term is for the case that LUT2 is hit after LUT1 is missed, and the second term is for the case that a codeword not in Cache1 is hit in LUT2. The corresponding VLC detector energy consumption is $(W(1-Ps)Pr_2+wPs(Pr_1+Pr_2-Pr_C))E(Y_b)$. Then we have a simplified form, $(wPr_2+wPs(Pr_1-Pr_C))E(Y_b)$. Finally, $\overline{E_{c\mathrm{LUT}}}$ and $\overline{E_{c\mathrm{VD}}}$ are formulated as in (9) and (10), shown at the bottom of the page. Based on the similar assumption to Case 1, i.e., $Pr_C=xPr_1$, $x>1$, $E_{HC}=yE_{H1}$, $y>1$, $E_{MC}\cong E_{HC}$, $E_{M1}\cong E_{H1}$, and $E_{H2}=zE_{H1}$, $z>1$, we have $\overline{E_{c\mathrm{LUT}}}\cong\overline{E_{\mathrm{LUT}}}+((y-1)-z(x-1)Pr_1)PsE_{H1}$ for (9), from which we can conclude that $\overline{E_{c\mathrm{LUT}}}$ is close to $\overline{E_{\mathrm{LUT}}}$. Applying the same assumption, the numerator in (10) can be rewritten as $Pr_1+(w-1-x)PsPr_1+wPr_2$. Since $w-1<1$ and $x>1$, $\overline{E_{c\mathrm{VD}}}$ is reduced by incorporating Cache1.

By the same manner, the throughput of a cache-equipped VLD, f_c, is modeled in (11), shown at the bottom of the page, where f is the throughput of the VLD in which no caches are adopted, and f_d is the operating frequency of the VLC detector. Observe that $f_c=f$ for $Pr_C=Pr_1$, $f_c<f$ for $Pr_C<Pr_1$, and $f_c>f$ for $Pr_C>Pr_1$.

As mentioned in Section II, another low-power approach is to make w equal to 1, which means the adder in the accumulator does not consume any energy for the wait cycle. As the adder and the VLC detector do not have to be activated to access Cache1 and LUT2, we can modify the accumulator as shown in Fig. 6 to prevent unintended adder activations. Note that the accumulator in Fig. 7(a) has a D FF at the output side of the adder, and, thus the adder is activated whenever the input changes. On the other hand, the modified accumulator in Fig. 7(b) employs two D FFs at the input side of the adder. Due to this, unintended adder activations are prevented. The modified accumulator leads to some changes on the remaining part of the VLD. As the carry of the adder is generated after the length and the shift are latched, it is no longer possible to update the D FFs in the accumulator and two D FFs in front of the barrel shifter at the same clock edge. For this reason, two length values are updated at different edges as shown in Fig. 7. If a new codeword shift is required, the enable signal is asserted using the current status of the controller and the table look-up results. Consequently, the average VLC detector energy consumption per codeword can be calculated by the following equation:

$$\overline{E_{c\mathrm{VD}}}=\left[P_{\mathrm{cr}}(0,Y_b)\frac{Pr_1-PsPr_C+Pr_2}{Pr_1+Pr_2}+\cdots\right]E(Y_b). \tag{12}$$

$$\begin{aligned}\overline{E_{c\mathrm{LUT}}}&=(1-Ps)\,Pr_1E_{H1}+PsPr_CE_{HC}+Pr_2\left(E_{H2}+(1-Ps)\,E_{M1}+PsE_{MC}\right)+\cdots\\&\quad+Pr_n\left(E_{Hn}+\sum_{i=2}^{n-1}E_{Mi}+(1-Ps)\,E_{M1}+PsE_{MC}\right)+E_{\mathrm{overhead}}\,(n+1)\\&\cong\overline{E_{\mathrm{LUT}}},\ \text{where } Pr_1=Pr_C \text{ implies } E_{H1}=E_{HC} \text{ and } E_{M1}=E_{MC}\end{aligned} \tag{7}$$

$$\overline{E_{c\mathrm{VD}}}=\left[P_{\mathrm{cr}}(0,Y_b)\frac{(1-Ps)\,Pr_1+(w-1)\,PsPr_C+wPr_2}{Pr_1+Pr_2}+\cdots\right]E(Y_b). \tag{8}$$

$$\begin{aligned}\overline{E_{c\mathrm{LUT}}}&=(1-Ps)\,Pr_1E_{H1}+PsPr_1E_{HC}+Ps\,(Pr_C-Pr_1)\,E_{HC}+(1-Ps)\,Pr_2\,(E_{H2}+E_{M1})\\&\quad+Ps\,(Pr_1+Pr_2-Pr_C)\,(E_{H2}+E_{MC})+Pr_3\,(E_{H3}+E_{M2}+(1-Ps)\,E_{M1}+P_SE_{MC})+\cdots\\&\quad+Pr_n\left(E_{Hn}+\sum_{i=2}^{n-1}E_{Mi}+(1-Ps)\,E_{M1}+PsE_{MC}\right)+E_{\mathrm{overhead}}\,(n+1)\\&\cong\overline{E_{\mathrm{LUT}}}+Ps\,(1-Pr_1)\,(E_{MC}-E_{M1})-PsPr_1E_{H1}+PsPr_CE_{HC}-Ps\,(Pr_C-Pr_1)\,(E_{H2}+E_{MC})\end{aligned} \tag{9}$$

$$\overline{E_{c\mathrm{VD}}}=\left[P_{\mathrm{cr}}(0,Y_b)\frac{(1-Ps)\,Pr_1+(w-1)\,PsPr_C+wPs\,(Pr_1-Pr_C)+wPr_2}{Pr_1+Pr_2}+\cdots\right]E(Y_b). \tag{10}$$

$$f_c=\begin{cases}\left[P_{\mathrm{cr}}(0,Y_b)\frac{(1-Ps)Pr_1+PsPr_C+2Ps(Pr_1-Pr_C)+2Pr_2}{Pr_1+Pr_2}+\cdots\right]^{-1}f_d, & \text{if } Pr_C\neq Pr_1\\ \left[P_{\mathrm{cr}}(0,Y_b)\frac{Pr_1+2Pr_2}{Pr_1+Pr_2}+\cdots\right]^{-1}f_d=f, & \text{if } Pr_C=Pr_1\end{cases} \tag{11}$$

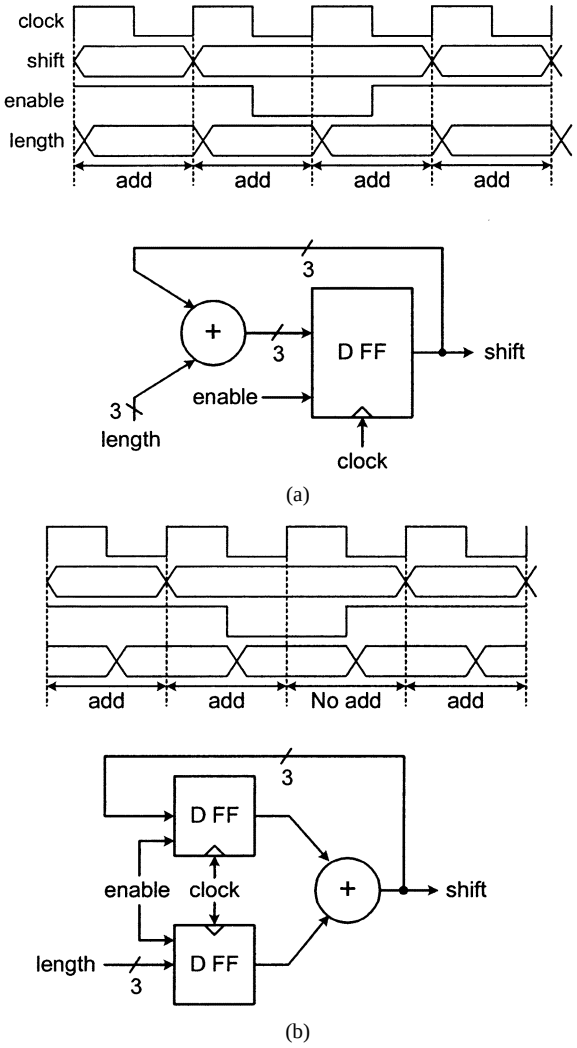


Fig. 7. Accumulator structure: (a) conventional structure that cannot avoid unintended adder activations [8] and (b) modified structure that prevents unintended adder activations.

IV. Experimental Results

To validate the proposed low-power scheme in practical designs, 24 VLDs were implemented for various configurations. The target VLC table, MPEG-2 DCT AC coefficient Table-B14, was partitioned into a number of LUTs and BLKs. The partitioning scheme employed in the implementation is different from the fine-grain partitioning scheme [8]. Since a VLC detector consumes much more power than a small LUT, a maximal number of codewords permitted by a given input bit-width are used to construct LUTs. Each VLD described in structure-level Verilog was synthesized with a 0.35-μm cell library [12] using Synopsys Design Compiler. The LUTs, Caches, and BLKs were implemented using random logic rather than PLA. After the synthesis, the gate-level netlist was converted into a transistor-level Spice file. Finally, power simulation was conducted using Epic PowerMill at the typical operating condition. The VLC streams obtained in Section II were also used as input stimuli for the power simulation.

The output size of the VLC detector greatly influences overall throughput and power consumption of the VLD. A small VLC detector is desirable for low power, and a large VLC detector is for high performance. Therefore we implemented 8-, 12-, and 16-bit VLC detectors. For each VLC detector, LUTs were optimized in terms of energy by repeating a number of power simulations.

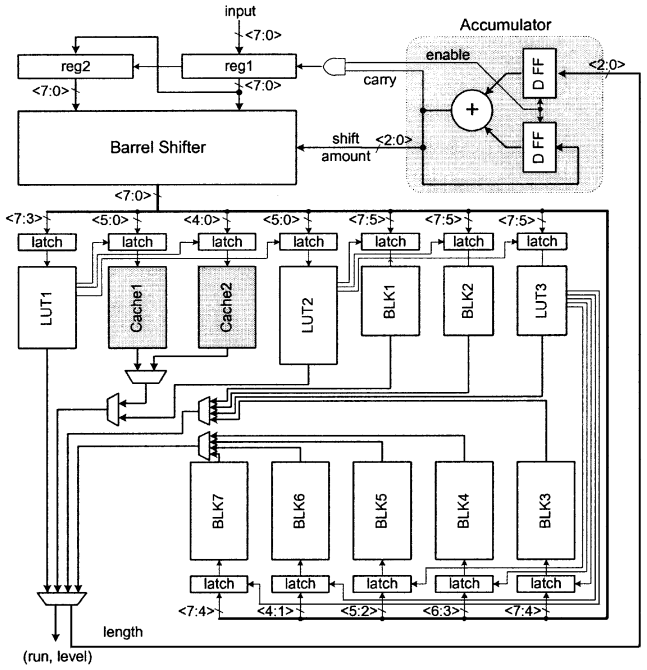


Fig. 8. Proposed low-power cache-equipped VLD architecture with an 8-bit VLC detector.

The VLD architecture based on the 8-bit VLC detector is shown in Fig. 8, which has two separated caches (Cache1 and Cache2) to further reduce power consumption. If the size of a short codeword found in LUT1 is 2 bits, Cache1 is accessed in the next cycle. 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 [8].

Fig. 9 shows the power consumption of the 8-bit VLD employing 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. The power dissipation of the VLC detector is also plotted in Fig. 9(b). The caches and the modified accumulator configuration are effective in reducing overall power. From the energy consumption result plotted in Fig. 9(d), LUT1 is determined to have eight entries.

Fig. 10 shows the low power VLD architecture for the 12-bit VLC detector. Like the case of the 8-bit VLC detector, table partitioning and cache insertion were applied together. As the codeword window is enlarged, the chance to insert caches is also increased. Therefore, four caches are employed in this architecture. This VLD operates in almost the same way as the one with the 8-bit VLC detector. The simulation results are presented in Fig. 11. The power reduction obtained by the caches is maximally 18.9%. The VLD with the 12-bit VLC detector consumes two times or more power than the one with the 8-bit VLC detector.

The low power VLD architecture based on the 16-bit VLC detector is shown in Fig. 12, where five caches are used. In this

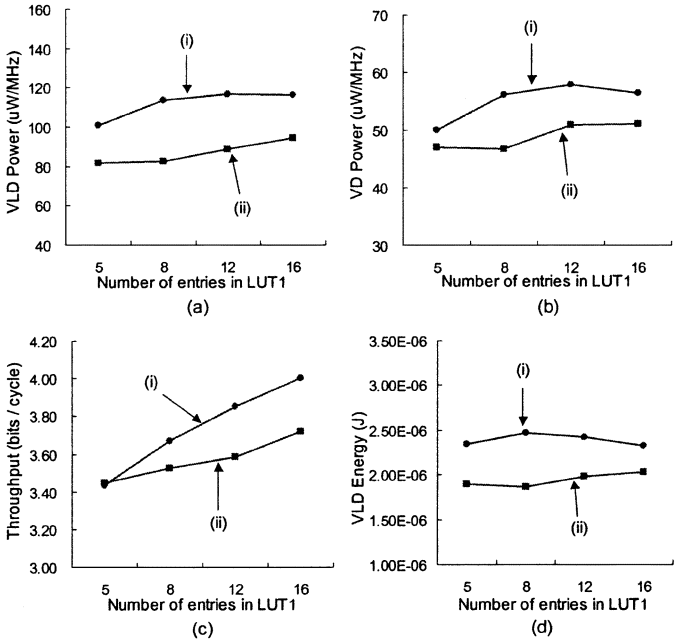


Fig. 9. Simulation results of the proposed VLD shown in Fig. 8. The proposed cache scheme is not applied in (i) and applied in (ii). (a) VLD power consumption, (b) VLC detector power consumption, (c) throughput, (d) VLD energy consumption for decoding a 10 kB VLC stream.

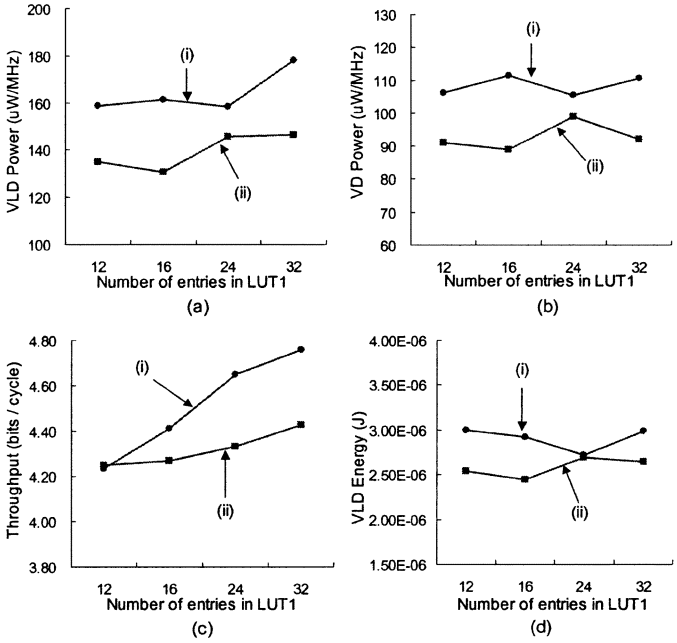


Fig. 11. Simulation results of the proposed VLD shown in Fig. 10. The proposed cache scheme is not applied in (i) and applied in (ii). (a) VLD power consumption, (b) VLC detector power consumption, (c) throughput, (d) VLD energy consumption for decoding a 10 kB VLC stream.

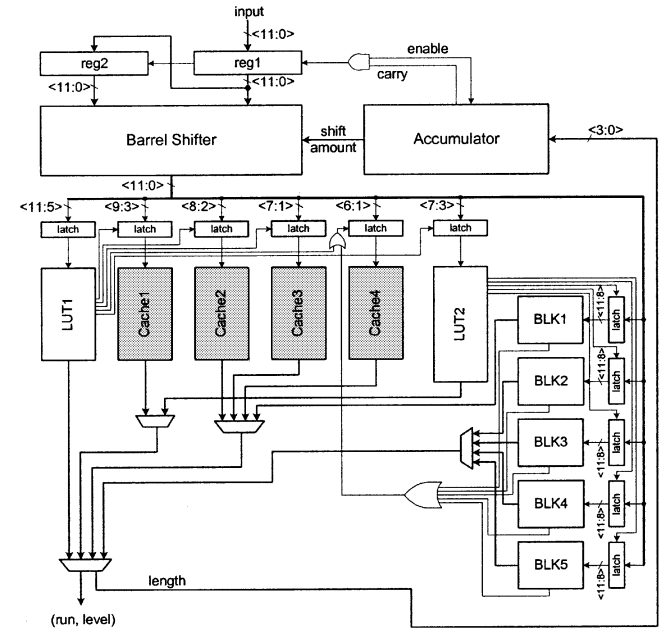


Fig. 10. Proposed low-power cache-equipped VLD architecture with a 12-bit VLC detector.

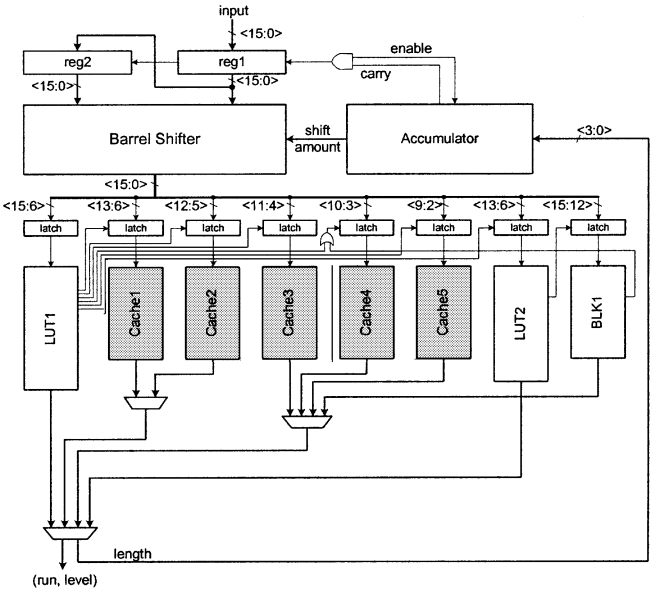


Fig. 12. Proposed low-power cache-equipped VLD architecture with a 16-bit VLC detector.

case, we can apply the proposed cache scheme to all codewords. Provided that long codewords are decomposed into prefixes and remaining codewords, the latter ones are also as short as those found in LUT1. The simulation result is presented in Fig. 13. Power reduction is plotted in Fig. 13(a) and (b). The power saving of 9.3% is obtained from the proposed cache insertion method. The power reduced in the VLC detector is as high as 22.8%. The power is mainly reduced by the fact that the 16-bit VLC detector is activated fewer times. The reduced number of VLC detector activations affects a great deal on the entire power dissipation.

The power consumption of the proposed VLD is compared to the previous VLD proposed in [8] that has been known as the best low-power architecture. To make the comparison fair, we redesigned the previous one with the same standard cell library [12] that was used for the design of the proposed VLD. The simulation results are compared in Table IV. Besides the higher throughput, approximately 30% of power reduction is achieved by employing the proposed cache scheme. Due to the two caches added, 5% area overhead is observed.

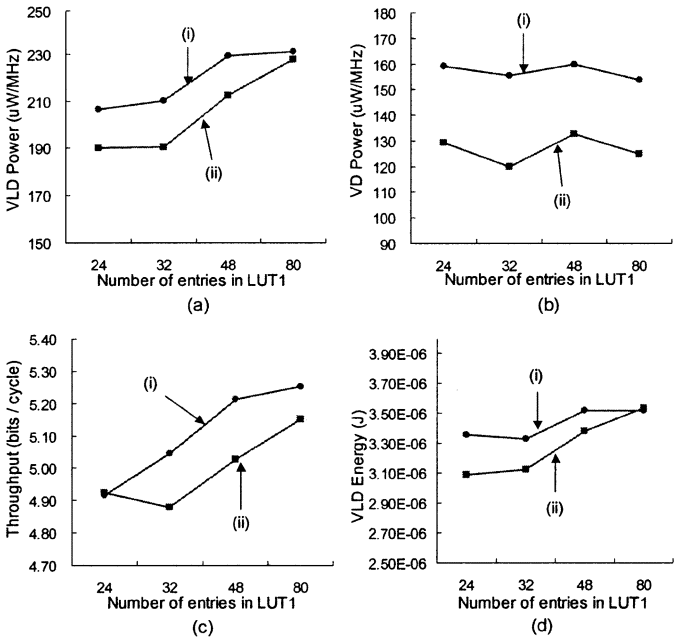


Fig. 13. Simulation results of the proposed VLD shown in Fig. 12. The proposed cache scheme is not applied in (i) and applied in (ii). (a) VLD power consumption, (b) VLC detector power consumption, (c) throughput, (d) VLD energy consumption for decoding a 10 kB VLC stream.

TABLE IV
PERFORMANCE COMPARISON

	Cho [8]	This work
Power	114.52 μW / MHz	82.66 μW / MHz
Throughput	3.15 bits/cycle	3.53 bits/cycle
	0.56 codewords/cycle	0.63 codewords/cycle
Energy	2.91 μJ/10KBytes	1.87 μJ/10KBytes
Normalized gate counts	1.00	1.05

V. CONCLUSION

In this paper, we have described a new low-power VLD scheme to reduce the power dissipation of a VLC detector where a lot of power is consumed. By exploiting the relation between two successive codewords, the number of VLD detector activations is reduced. This idea was implemented by employing small LUTs working as fixed caches. The proposed scheme was applied to three differently sized VLC detectors (8, 12, and 16 bits). For each VLC detector, the overall power consumption was significantly reduced at the expense of a little circuit overhead. Intensive simulation results show that the proposed cache-equipped VLD consumes 35% less energy on the average than the state-of-the-art low-power VLD [8] without sacrificing throughput.

REFERENCES

[1] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE*, vol. 40, pp. 1098–1101, Sept. 1952.
[2] S. F. Chang and D. G. Messerschmitt, "Designing high-throughput VLC decoder: Part I—Concurrent VLSI architectures," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 187–196, June 1992.
[3] M. T. Sun and S. M. Lei, "A parallel variable-length-code decoder for advanced television applications," in *Proc. 3rd Int. Workshop HDTV*, 1989.
[4] 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.
[5] 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.
[6] 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.
[7] 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.
[8] 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, June 1999.
[9] C. H. Lin and C. W. Jen, "Low power parallel Huffman decoding," *Electron. Lett.*, vol. 34, no. 3, pp. 240–241, Feb. 1998.
[10] S. H. Cho, T. Xanthopoulos, and A. P. Chandrakasan, "Design of low power variable length decoder using fine grain nonuniform table partitioning," in *Proc. IEEE Int. Symp. Circuits and Systems*, 1997, pp. 2156–2159.
[11] *Generic Coding of Moving Picture and Associated Audio, Draft International Standard*, Recommendation H.262, ISO/IEC 13 818, Nov. 1994.
[12] *0.35 μm 3.3 V CMOS Standard Cell Library Data Book*, Samsung Electronics Company, Ltd., Yongin, Kyunggi-Do, Korea, 1999 .

Sung-Won Lee (S'00) received the B.S. and M.S. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, in 1998 and 2000, respectively. Currently, he is working toward the Ph.D. degree in electrical engineering at KAIST.

His research interests include low-power system architecture and digital circuits design for a communications system.

In-Cheol Park (M'93-SM'02) received the B.S. degree in electronics engineering from Seoul National University, Seoul, Korea, in 1986, and the M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1988 and 1992, respectively.

From May 1995 to May 1996, he was a Researcher at the IBM T. J. Watson Research Center, Yorktown, NY where he worked on high-speed circuit design. In June 1996, he was an Assistant Professor and is currently an Associate Professor in the Department of Electrical Engineering and Computer Science, KAIST. His current research interests include CAD algorithms for high-level synthesis and VLSI architectures for general-purpose microprocessors.

Dr. Park received the best paper award at the ICCD in 1999, and the best design award at the ASP-DAC in 1997.