An Ultra Low Power Variable Length Decoder for MPEG-2 Exploiting Codeword Distribution

SeongHwan Cho, Thucydides Xanthopoulos, and Anantha P. Chandrakasan

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge, MA

Abstract

A data driven variable length decoder chip is presented which exploits the signal statistics of variable length codes to reduce power. The approach uses fine grain lookup table partitioning to reduce switched capacitance based on codeword frequency. The variable length decoder for MPEG-2 has been fabricated and consumes 530μW at 1.35V for a video rate of 48M DCT samples/sec using a 0.6μm CMOS technology. More than an order magnitude of power reduction is demonstrated without performance loss compared to a conventional parallel decoding scheme with a single lookup table.

I. INTRODUCTION

Variable length coding is a widely used technique in video compression systems. It is often applied together with other lossy image compression techniques to further compress the data. The main idea of variable length coding is to minimize the average codeword length by exploiting the statistics of the data. Shorter codewords are assigned to frequently occurring data while longer codewords are assigned to infrequently occurring data. Therefore, minimum average codeword length and bit rate reduction can be achieved.

Various approaches have been presented to implement high throughput variable length decoders. They can be largely divided into the binary tree search method (decoding one bit per cycle [1]) and the parallel method (decoding multiple bits per cycle [2]). Further improvements based on these methods have also been proposed [3] [4]. However, these variable length decoders were mainly aimed at high throughput and power dissipation was not the main focus. In this paper, we discuss a method for designing low power variable length decoders for portable video systems. The concepts are applied to a variable length decoder which decodes all the variable length codes in the MPEG-2 standard.

II. LOW POWER ARCHITECTURE

The proposed variable length decoder (VLD) is based on the parallel method [2]. Since the parallel method processes multiple bits per cycle, the VLD system clock frequency can be reduced while achieving high throughput. Hence, voltage scaling can be applied more effectively with the parallel structure.

At a high level, the parallel VLD can be decomposed into two components, the variable length code (VLC) detector and the lookup table. The VLC detector receives the input variable length codes and generates an address for the lookup table (LUT). To reduce additional circuit overhead, address generation is implemented by aligning the VLCs at a fixed position so that the lookup table uses the VLC itself as the address. The lookup table receives the address from the VLC detector and produces the corresponding output codeword and length. Both components of the variable length decoder must be optimized to achieve low power operation.

A. Prefix Pre-decoding

In most cases when the number of codewords in the table is large, there are some prefixes that are common to the long variable length codes. By exploiting these common prefixes, the size of the lookup table can be reduced. Several approaches have exploited this prefix pre-decoding method to efficiently decode the variable length codes [5] [6]. The basic idea of prefix pre-decoding is to group the variable length codes by their common prefixes. With this prefix pre-decoding method, the size of the lookup table is reduced, because the prefixes are no longer redundant in the lookup table. In addition, power dissipation can be lowered since the switched capacitance is also reduced.

B. Non Uniform Table Partitioning

The VLC lookup tables are often the most area and power intensive blocks of a variable length decoder. The average energy consumption per codeword in the lookup table can be modeled by the following equation, where P_{cwi} is the probability that codeword i will occur, E_i is the energy required to decode the codeword i, and n is the total number of codewords in the table.

$$Energy = \sum_{i=1}^{n} P_{cwi} E_i \qquad (1)$$

In conventional approaches the energy required to decode a variable length code does not vary much over the codeword probability (i.e., $E_i \approx Constant$, $\forall i$). This is because the variable length code table is implemented in a single lookup table and the whole table has to be charged and discharged every cycle. The single lookup table method does not exploit the fact that the short codewords are frequently occurring. Therefore, the average energy in Eq. 1 is dominated by codewords which have high probability of occurrence.

0-7803-4292-5/97/$10.00 © 1998 IEEE

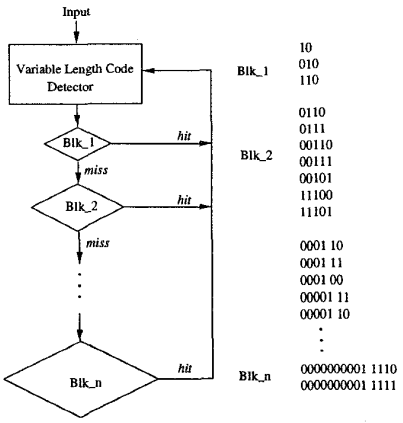

Fig. 1. Low Power Algorithm based on LUT partition

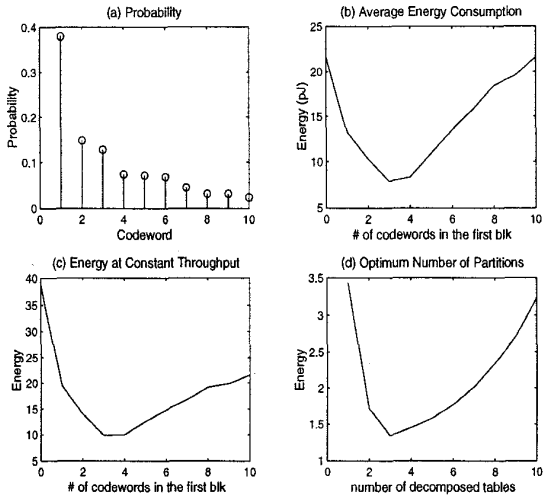

Fig. 2. Example of Table Partitioning

We have previously proposed a low power algorithm that exploits the variable length codeword statistics, making the energy to decode a codeword depend on the codeword probability [7]. Fig. 1 shows the basic algorithm for low power variable length decoding. The main idea is to partition the single lookup table into several variable size blocks with respect to their energy consumption and frequency of occurrence. By having variable size blocks, the energy required to decode a codeword will vary depending on the size of the partitioned block. Low power can be achieved if the dominant term in Eq. 1 is made small (i.e., E_i with high P_{cwi}). In our proposed approach, frequently occurring codewords use small tables with less switched capacitance and infrequent codewords use larger tables.

The flowchart of Fig. 1 works as follows: Given the codeword lookup table decomposed into several blocks, the VLC from the VLC detector goes through a series of blocks looking for a match. If there is a match, the output word will be produced and the next VLC will be processed, going through the same procedure. In case of a miss, the matching process will continue until the codeword is fully decoded. To achieve the maximum hit ratio, blocks are ordered in order of descending probability. With this data driven decoding process, Eq. 1 is modified as the follows.

$$\begin{aligned} Energy &= Pr_1 E_{H1} + Pr_2(E_{H2} + E_{M1}) \\ &+ Pr_3(E_{H3} + E_{M1} + E_{M2}) + \ldots \\ &+ Pr_n(E_{Hn} + \sum_{i=1}^{n-1} E_{Mi}) + E_{overhead}(n) \quad (2) \end{aligned}$$

where Pr_i is the probability that block i is a hit (i.e., a match was found in block i), n is the total number of decomposed blocks, E_{Hi} is the energy dissipated by block i when there is a hit and E_{Mi} is the energy when there is a miss. $E_{overhead}(n)$ is the energy consumed by the circuit overhead introduced by the table partitioning, such as conditional branches and clock generation which increases with n. The average energy consumption is no longer a constant value, but the sum of the energy consumption in the decomposed blocks weighed with the corresponding probability. To achieve minimum average energy, the first block must consume the least energy while having a high hit ratio. The hit ratio can be improved by allocating more codewords, but this will also increase the power dissipation. Hence, an optimum solution must be found.

Fig. 2 illustrates an example of table partitioning for a 10 codeword table. The probability distribution of each codeword occurrence is shown in Fig. 2(a). The average energy when the table is decomposed into two blocks (i.e. 2-way partition) is shown in (b). As can be seen, there is an optimum number of codewords that should be assigned to each block to achieve minimum energy. Another issue that must be considered in the table partitioning is the throughput. Each conditional branch that the VLC goes through in the flow chart of Fig. 1 is in the critical path. With this scheme, it would take n cycles to decode a codeword in the nth block. In the two way partitioning scheme shown in (b), although the energy consumption at $n = 0$ and $n = 10$ is about the same, the throughput differs by a factor of two. A fair comparison should account for variation in throughput. Fig. 2(c) shows the energy for a fixed throughput obtained by varying the supply voltage. In Fig. 2(d), the graph shows the optimum number of partitions that should be used for minimum energy. It shows the minimum energy for each partition with the overhead energy considered. In this example, the optimum number of partitioning is 3. The average energy consumption increases when the number of partitioning exceeds 4, when the $E_{overhead}$ starts to become dominant.

C. Variable Length Code Detector Based Optimization

Unlike the decomposed lookup table which is accessed according to their probability, the variable length code detector is operating on every cycle. From the result of prefix pre-decoding and fine grain non-uniform table partitioning, the lookup table is optimized to dissipate minimum power. On the other hand, the VLC detector is still designed for

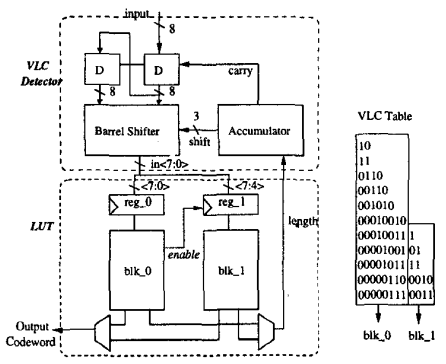

Fig. 3. VLC Detector Optimization

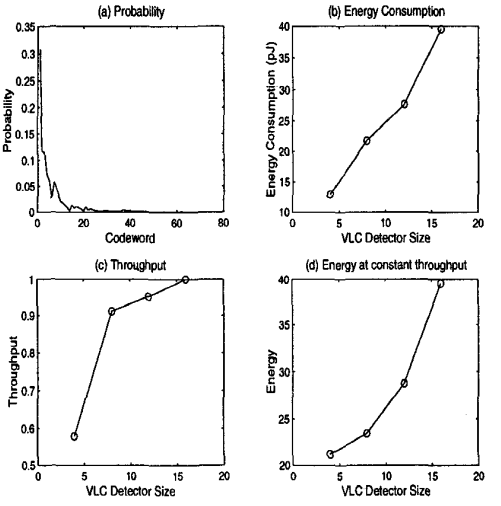

Fig. 4. Results of VLC Detector Based Table Partitioning

the worst case input, which is for the case when the longest VLC comes in. Since the variable length code or a prefix has to be fully decoded in one lookup table operation in a hit cycle, the output bit width of the barrel shifter must be at least the size of the maximum length VLC or prefix. This is because the output of the barrel shifter is the aligned VLC which acts as the address to the lookup table. Hence, a large capacitance is switched in the VLC detector even when short variable length codes are input. To minimize the power consumption in the VLC detector, we propose a method which exploits the signal statistics of the input variable length codes. By breaking up the variable length codeword into two or more parts, we can reduce the size of the VLC detector and the power consumption.

Fig 3 illustrates an example of the reduced VLC detector architecture. Since the output bit width of the VLC detector is only 8 while the maximum length VLC is 12, VLCs of length greater than 8 is decoded in two cycles, in blk_0 and blk_1. For example, if a VLC of 000001110011 is input, the first 8 bits (00000111) are decoded in blk_0 and then the rest (0011) is decoded in the next cycle in blk_1. Using this scheme we have reduced the energy consumption in the VLC detector but decreased the throughput.[1] An optimum size of the VLC detector can be determined by the simulation results shown in Fig. 4. It shows the plot of the energy consumption in the VLC detector as we vary its size.

The codeword probability of the first 80 MPEG-2 DCT variable length codes is plotted in Fig. 4(a). Graph in Fig. 4(b) shows the energy consumption of the VLC detector as we increase its size (i.e., output bit width of the VLC detector). Throughput is shown in the next graph, where it can be seen that although the energy decreases with smaller size of the VLC detector, the throughput also decreases. To achieve the optimum performance, we look at the energy consumption at same throughput in Fig. 4(d). Minimum energy is achieved when the size of VLC detector is $4 \sim 8$. Although $size = 4$ shows a slight improvement, we choose $size = 8$ since a system with $size = 4$ has to run almost at twice the frequency of a system with $size = 8$ to achieve the same throughput.

[1]The average power consumption in the LUT will decrease since the long VLCs are split in two. However, additional circuit overhead is introduced by decomposing the table. The LUT power reduction is dominated by table partitioning and it is neglected in this section.

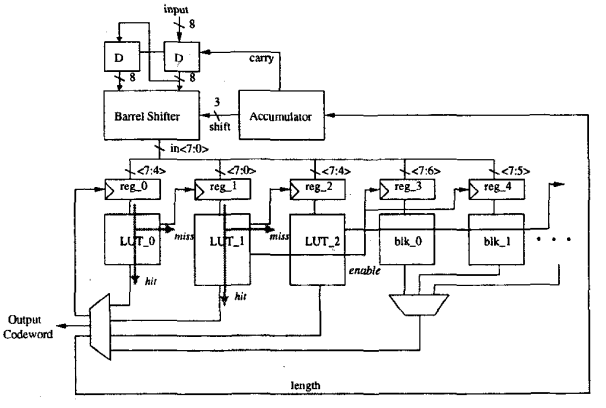

Fig. 5. Low Power Variable Length Decoder Architecture

D. Low Power Variable Length Decoder Architecture

The proposed low power VLD architecture is shown in Fig. 5. The table partitioning and the VLC detector optimization is applied together with the prefix based decoding. The first three blocks, LUT_0, LUT_1, and LUT_2 decode the short codewords and prefixes. Blocks blk_0, blk_1 have codewords without the prefixes. When a variable length comes in, reg_0 is latched and it is fed to LUT_0. If it is a hit, then the corresponding output codeword is produced. If it is a miss, the next block (i.e., LUT_1) is enabled. Most of the time, only the first block will be enabled, resulting in low energy consumption. The power is further reduced by choosing static CMOS in implementing the lookup table rather than a PLA. Although the area is larger than the PLA, static CMOS has the advantage that its performance

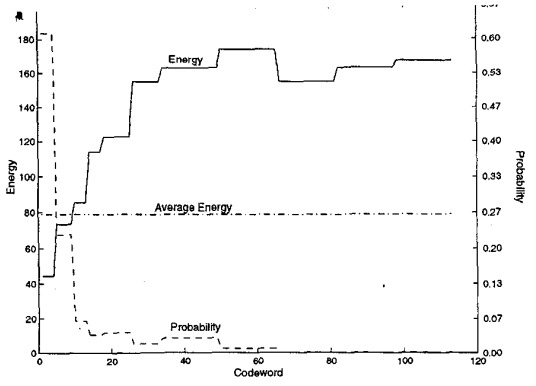

Fig. 6. Energy Consumption

and power consumption are better at a low supply voltage.

III. Experimental Results

A chip has been fabricated and tested and the results are plotted in Fig. 6. The graph shows the probability and the energy consumption of the decomposed tables based on the MPEG-2 DCT table. Each codeword is numbered and represented on the horizontal axis. The solid line represents the energy consumption of each block. It was obtained by feeding in codewords of that particular block only. The energy correlation that may occur when the input bit stream is composed of different blocks is neglected. As one can see, the energy consumption increases as the probability of a hit decreases. There are three big jumps in the energy plot at $n = 4, 13, 27$, each indicating a miss. The energy consumption of the block does not increase monotonically due to the varying switching activities of each block. The dotted line shows the average energy consumption of the actual MPEG-2 DCT sequence.

Table I summarizes the test results of the low power variable length decoder chip. The chip is implemented using $0.6\mu m$ CMOS process as shown in Fig. 7. The boxes with 'B-' indicate the VLC table as denoted in ISO/SEC 13818-2. The largest tables are B-14 and B-15 which are the DCT AC coefficients. The measured power consumption for the DCT sequence was about $0.530\ mW$ at $1.35V$. More than two orders of magnitude power reduction was achieved compared to the result shown in [3].

TABLE I
Chip Test Results

Power	$0.530mW$ @ 1.35V
Area	$5.24mm^2$(core), $16.54mm^2$(full)
Technology	$0.6\mu m$ CMOS (V_{TN}=0.67V. V_{TP}=-0.93V)
Frequency	$12.5MHz$
Video rate	$48.75Msamples/s$ (4:2:0)
Input rate	$69.38\ Mbits/s$

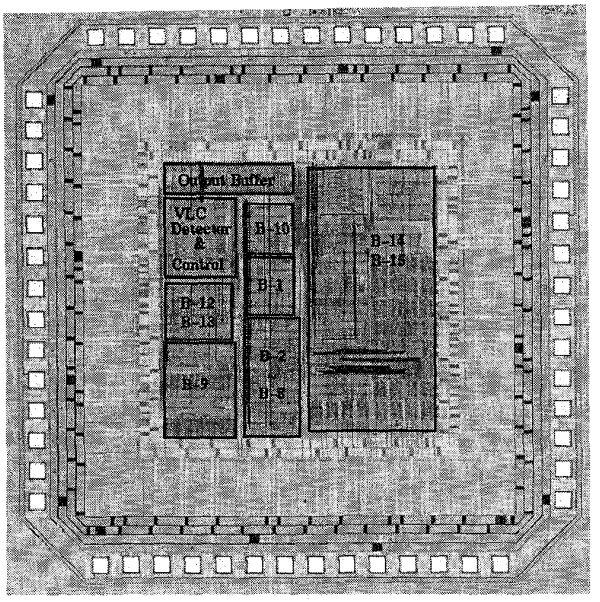

Fig. 7. Die Photo of VLD for MPEG-2 Decoder

IV. Conclusion

A low power variable length decoder has been presented. Power reduction was achieved mainly by architecture level optimizations, where we reduced the energy consumption of the lookup table and the VLC detector. The lookup table was decomposed into several variable size blocks by exploiting the statistics of the variable length code. Optimization of the VLC detector significantly reduced the average power of the VLD by lowering the offset power of the overall system.

V. Acknowledgment

This research is sponsored by SHARP Corporation. The authors would like to thank Yoshifumi Yaoi for his valuable contributions.

References

[1] A. Mukherjee, N. Ranganathan, and M. Bassiouni, "Efficient VLSI designs for data transformations of tree-based codes," *IEEE Trans. Circ. & Syst.*, pp. 306-314, March 1991.

[2] S.M. Lei and M.T. Sun, "An entropy coding system for digital HDTV applications," *IEEE Trans. Circ. & Syst. Video Tech.*, vol. 1, pp. 147-155, Mar. 1991.

[3] E. Komoto and M. Seguchi, "A 110Mhz MPEG-2 variable length decoder LSI," *Proc. of Symp. on VLSI Circuits*, pp. 71-72, 1994.

[4] Y. Ooi, A. Taniguchi, and S. Demura, "A 162 Mbits/s variable length decoding circuit using an adaptive tree search technique" *IEEE CICC*, pp. 107-110, 1994.

[5] R. Hashemian, "Design and hardware construction of a high speed and memory efficient Huffman decoding," *IEEE Int. Conf. on Consumer Electronics*, pp.74-75, 1994.

[6] S.F. Chang and D.G. Messerschmitt, "Designing high-throughput VLC decoder Part I - Concurrent VLSI architectures," *IEEE Trans. Circ. & Syst. Video Tech.*, vol. 2, no. 2, pp. 187-196, June 1992.

[7] S.H. Cho, T. Xanthopoulos, and A.P. Chandrakasan, "Design of Low Power Variable Length Decoder using Fine Grain Non-Uniform Table Partitioning," *IEEE ISCAS*, pp. 2156-2159, June 1997.