DESIGN OF LOW POWER VARIABLE LENGTH DECODER USING FINE GRAIN NON-UNIFORM TABLE PARTITIONING

SeongHwan Cho Thucydides Xanthopoulos Anantha P. Chandrakasan

Department of EECS
Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT

Variable length coding is a widely used technique in digital video compression systems. Previous work related to variable length decoders are primarily aimed at high throughput applications, but the increased demand for portable multimedia systems has made power a very important factor. In this paper a data driven variable length decoding algorithm is presented, which exploits the signal statistics of variable length codes to reduce power. It uses fine grain lookup table partitioning based on codeword frequency. An order magnitude of power reduction is possible compared to conventional parallel decoding scheme with a single lookup table.

1. Introduction

Variable length decoding (e.g., Huffman decoding) is a widely used technique in video compression modules. 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 [1]. Various approaches have been presented to achieve high throughput variable length decoding [2], such as parallel decoding [3],[4] and adaptive tree decoding [5]. This paper presents an approach to reduce the power dissipation of variable length decoders by exploiting signal statistics.

There are two main approaches for implementing a variable length decoder. One is a binary tree search method and the other is the parallel decoding method [3]. The binary tree search technique can be implemented based on a Huffman tree, which uses the principle of a token propagation in a reverse binary tree constructed from the original codes [6]. However, for high performance systems like MPEG2 or HDTV, this approach is not suitable since it can decode only one bit per cycle and for large lookup tables, area and power consumption increase dramatically. On the other hand, parallel decoding approach has the advantage of decoding more than one bit per cycle.

A parallel variable length decoder can be decomposed into two parts, the variable length code detector and the lookup table. The variable length code (VLC) detector receives the input VLCs and generates an address for the lookup table (LUT). To reduce additional circuit overhead, address generation is implemented by aligning the variable length codes in such a way so that the lookup table uses the VLC itself as the address. The throughput and power of the variable length decoder depend on how the VLC detector is implemented and how big the size of the VLC table is.

2. Low Power Variable Length Decoder

2.1 VLC Table Partitioning

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

$$Energy = \sum P_{cwi} \cdot E_i \qquad (1)$$

In conventional approaches [2]-[4], the energy required to decode a VLC does not vary much over the codeword probability (i.e., $E_i \approx E_{constant}$ for all i). Therefore, the average energy in equation (1) is dominated by codewords which have high probability of occurrence.

Figure 1 shows the proposed algorithm for low power variable length decoding. The idea is to partition the VLC table into several variable size blocks with respect to their energy dissipation 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 equation (1) is made small, which is E_i with high P_{cwi}.

Given the codeword lookup table split into several blocks as Figure 1, the variable length code from the VLC detector goes through a series of blocks looking for a match. If there's a match, the output word will be produced and the next variable length code will be processed, going through the same process. 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, equation (1) is modified as

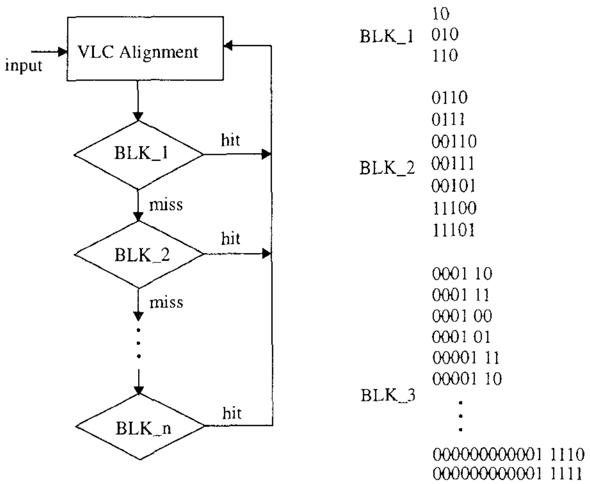


Figure 1. Low Power VLD Algorithm.

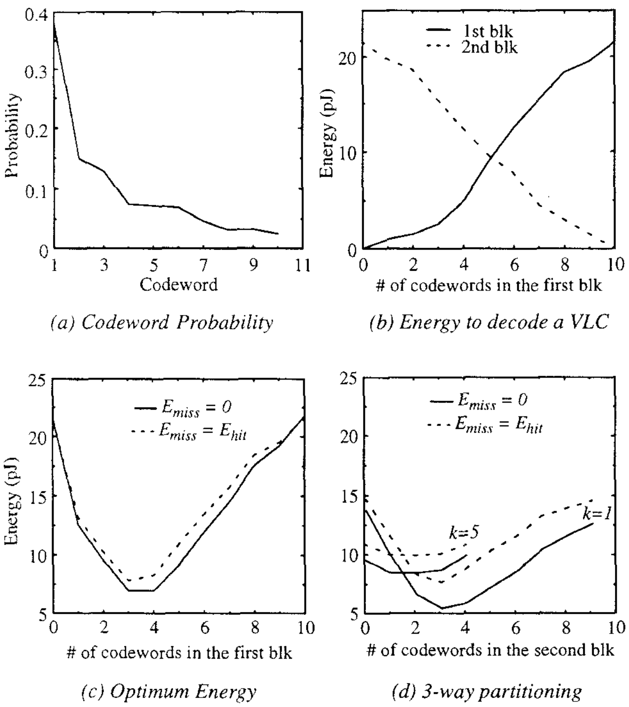
$$Energy = Pr_1 E_{H1} + Pr_2 (E_{H2} + E_{M1}) + Pr_3 (E_{H3} + E_{M1} + E_{M2}) + \ldots + Pr_n (E_{Hn} + \sum_{i=1}^{n-1} E_{Mi}) \qquad (2)$$

where Pr_i is the probability that block i is a hit (i.e. a match was found in block i), 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.

To come up with an optimum solution on how to partition the codeword table, first consider splitting the codeword table into two blocks (2-way partition). The question now comes down to how many codewords should be assigned to the first block to achieve minimum energy. With the proposed algorithm,

$$Energy = Pr_1 E_{H1} + (1 - Pr_1)(E_{H2} + E_{M1}) \qquad (3)$$

Consider a VLC table composed of the first 10 codewords in Figure 1. The probability of occurrence for each codeword is shown in Figure 2(a). Assuming that the first n codewords are in the first block, the energy required to decode a codeword in the first and the second blocks is shown in Figure 2(b). In general, determining a specific value for E_{hit} and E_{miss} is not an easy task. These energies will vary depending on how the block is implemented. For blocks implemented in PLA or ROM, $E_{miss} \approx E_{hit}$. In general, it is reasonable to assume that E_{miss} will be same as or less than E_{hit}, since the switched capacitance will typically be larger in case of a hit than a miss. Hence, considering this variation of E_{miss}, let's assume $0 < E_{miss} \leq E_{hit}$ and solve equation (3) for two extreme cases: $E_{miss}=0$ and $E_{miss} = E_{hit}$. Each will represent a lower and an upper bound for equation (3). Using the data from Figure 2(b), we can numerically solve equation (3) and find the optimum number of codewords n, that should be allocated to the first block for minimum energy. The results are shown in Figure 2(c). The graph shows total energy vs. n, the number of codewords in the first block. The energy is calculated for two cases: $E_{miss}=0$ and $E_{miss} = E_{hit}$. The actual energy plot will lie somewhere between the two curves. As can be seen in the graph, the energy does not depend much on E_{miss}. For an optimum partitioned table, this is rather an obvious result since the block size and its energy will be increasing as the codeword goes through the series of blocks and thus, E_{miss} of previous block will be negligible compared to E_{hit} or E_{miss} of the next block.



(a) Codeword Probability

(b) Energy to decode a VLC

(c) Optimum Energy

(d) 3-way partitioning

Figure 2. Finding the Optimum LUT Partitioning.

In this example, the optimum number of codewords that we get is around $n = 3$ *or* 4. This 2-way table partitioning can be used again to further partition the LUT. An M-way partition is also possible. Figure 2(d) shows the energy plot for 3-way partitioning, where k represents number of codewords in the first block.

There are some circuit overheads introduced by partitioning the LUT. If the LUT is split into M blocks, there need to be M registers for the input of each block and an M to 1 multiplexer for the output (Figures 4,5). With this overhead taken into account, another term $E_{overhead}$ must be added to equations (2) and (3), where $E_{overhead}$ increases as the number of partitioning, M. The partitioning should be done to a limit where this overhead doesn't effect the power of the overall system. For the example shown in Figure 2(d), 3-way partitioning does not have much power savings over 2-way partitioning if this overhead is considered.

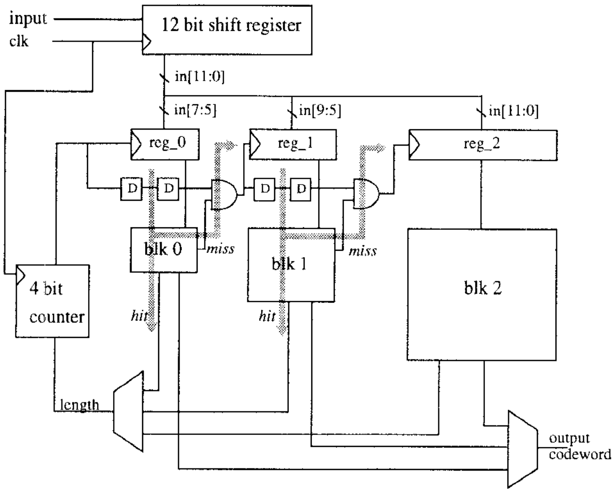


Figure 3. Serial Approach.

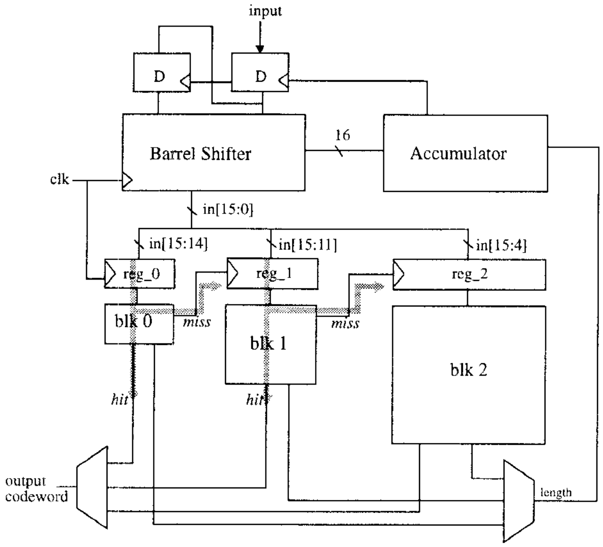


Figure 4. Parallel Approach.

2.2 Variable Length Code Detector (VLC Alignment)

Figures 3 and 4 show two implementations of the proposed algorithm. Both architectures use the lookup table which is partitioned into variable size blocks. The serial approach in Figure 3 uses a shift register and a counter for the variable length code detector. The shift register receives a serial input bitstream and generates address for the partitioned VLC tables. The address is generated by properly aligning the VLCs so that the VLC itself is used as the address of each block. The counter counts the length of the decoded VLC and latches the register when the new VLC is aligned to be processed. To allow extended computation time in the critical path, the VLC is aligned at the intermediate bits of the shift register.

As a result of the previous section, the VLC table is partitioned into blocks in such a way that the block size, hence the energy consumption per block increases as its probability of hit decreases. In Figure 3, block 0 which has the lowest energy will be accessed most frequently in the decoding process. Therefore a very low power operation can be achieved.

Figure 4 shows another implementation of the proposed algorithm. The lookup tables are partitioned in the same manner as in the serial approach. For the VLC detector, a barrel shifter and an accumulator is used instead of the shift register and the counter [2]. Barrel shifter allows us to shift many bits per cycle whereas in using shift registers, the system has to wait several cycles until the VLC is aligned. Therefore many bits can be decoded in one clock cycle by using the barrel shifter. The drawback in this case is that the circuit complexity and power increases with the barrel shifter and the accumulator.

The power consumption of the VLC detector will depend on the clock frequency and its switching capacitance. Unlike the VLC lookup tables, VLC detector will always be operating. The clock frequency at which the VLC detector is running is set by the output codeword rate. For the parallel approach, the output rate is same as the barrel shifter clock frequency and for the serial approach, the output rate, $f_{out} = f_{SR} / L_{avg}$, where f_{SR} is the clock frequency of the shift register and L_{avg} is the average length of the variable length codes. For the two implementations to run at the same given output rate, the shift register in the serial approach has to run L_{avg} times faster than the barrel shifter. Although the shift register has a higher operating frequency, simulation results show that the power consumption in the VLC detector is less for the serial approach than the parallel approach. This is because the barrel shifter and the accumulator has a higher switched capacitance than the shift register and the counter. At the same average output rate, a factor of 2 power reduction can be achieved in the VLC detector by using the serial method. However, for applications which require high throughput, the clock frequency of the serial approach has to be up to several hundred MHz, which may be too much an overhead to generate for the MPEG2 decoder system. Also, the rate at which the output codewords are produced will vary considerably in the serial approach. Additional circuits must be added to the output if a constant output rate is to be achieved [7].

2.3 LUT Power Reduction

Reducing power for the LUT at a high level has been proposed in the previous section. For low power operation, each partitioned block in LUT has to be optimized at a low level. The VLCs in the largest block of the partitioned VLC table usually have some prefixes that are common to several variable length codes. By clustering the VLCs by their prefixes, power and area is further reduced [8],[9].

At a circuit level, there are different types of circuit style that can be used for the lookup table implementation. In the conventional method, PLA is preferred since PLA does not waste memory space compared to a ROM. For example, building a VLC table which has VLC length up to 16 bits would require 2^{16} size ROM. There are ways to efficiently use the ROM without wasting memory cells, but additional lookup table is needed to convert variable length codes to fixed length addresses [10]. In our proposed approach, static CMOS is used for small blocks. Simulation results show that a factor of two power reduction can be achieved by using a CMOS compared to a PLA. It is important to reduce the energy consumption of the first block as much as possible since it will be accessed most frequently.

3. Experimental Results

Figure 5 shows the energy dissipation of the variable length codes in the proposed parallel architecture. MPEG2 DCT AC VLC table is used and it is partitioned into three blocks. The graph shows the energy for the first 40 codewords.

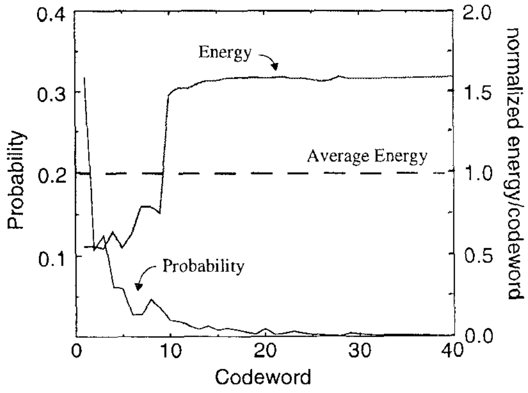


Figure 5. Energy dissipation of the proposed VLD.

It is interesting to note that the energy consumption of the third block, which has about 100 codewords, is only three times that of the first block which has only 3 codewords. This is mainly due to two reasons. First, the third block is further optimized for power. The energy consumption in the third block is reduced by decomposing the VLCs into their common prefix. Second, the power dissipated by the VLC detector has now become a significant portion of the overall power due to the reduced power in the lookup table. The VLC detector, which is always operating gives an offset to energy dissipation for decoding the codeword in every block. This causes the average energy to go up higher than expected.

Compared to the conventional approach where the VLC lookup table is implemented in one single PLA, more than an order magnitude of power reduction is achieved by using the fine grain partitioned lookup table.

4. Conclusion

We have developed a low power variable length decoder by exploiting the signal statistics of the compressed video bitstream. An algorithm for low power is presented and it is implemented in two ways. The parallel approach has a high throughput and the serial approach has small power and area overhead for the VLC detector. The power of the overall system is significantly reduced by optimizing the lookup table, which dissipates the dominant power of the decoder system. The table is partitioned into variable size blocks based on the probability and energy consumption of the variable length codes. An order magnitude of power reduction was possible by optimally partitioning the VLC table.

5. Acknowledgment

This research is sponsored by SHARP Corporation. Thucydides Xanthopoulos is supported by a National Semiconductor graduate fellowship. The authors would like to thank Yoshifumi Yaoi for his valuable contributions.

6. References

[1] D.A. Huffman, "A method for the construction of minimum redundancy codes," *Proceedings of IRE*, vol. 40, no. 10, pp. 1098-1101, Sept. 1952.

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

[3] S. M. Lei and M. T. Sun, "An entropy coding system for digital HDTV applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 1, pp.147-155, Mar. 1991.

[4] Y. J. Jan, "A high speed variable length decoder for digital HDTV systems," *Signal Processing of HDTV IV*, pp. 333-340, 1992.

[5] Yasushi Ooi, Atsushi Taniguchi, Shigeki Demura, "A 162 Mbit/s Variable Length Decoding Circuit using an Adaptive Tree Search Technique," *IEEE Custom Integrated Circuits Conference*, pp.107-110,1994.

[6] A.Mukherjee, N. Ranganathan, and M. Bassiouni, "Efficient VLSI designs for data transformations of tree-based codes," *IEEE Transactions on Circuits & Systems*, pp. 306-314, 1991.

[7] Mikael Rudberg, Lars Wanhammar, "New Approaches to High Speed Huffman Decoding," *IEEE International Symposium on Circuits and Systems*, pp.149-152, 1996.

[8] S.B. Choi and M.H. Lee, "High Speed Pattern Matching for a Fast Huffman Decoder," *IEEE Transactions on Consumer Electronics*, vol. 41, no. 1, pp.97-103, Feb. 1995.

[9] R. Hhashemian, "Design and hardware construction of a high speed and memory efficient Huffman decoding," *IEEE International Conference on Consumer Electronics*, pp.74-75,1994.

[10] E. Komoto and M. Seguchi, "A 110Mhz MPEG2 variable length decoder LSI," *Proceedings of Symposium on VLSI circuits*, pp. 71-72, 1994.