3.4-4

Software-only Multiple Variable Length Decoding for Real-Time Video on MDSP*

Ganesh Yadav[1], R.K. Singh[2], and Vipin Chaudhary[1]
[1]Dept. of Computer Science, Wayne State University and [2]Cradle Technologies Inc.

***Abstract*—We present the multiple variable length decode algorithm implemented in most video applications on MDSP. In our implementations we were able to decode multiple symbols per cycle. The implementation is efficient and is targeted towards memory constrained embedded systems. We have confirmed this algorithm in our implementations of H261/3, MPEG2/4 and achieved multifold speedup improvements against algorithms, which can decode at the symbol rate only. This limits the decoding throughput capability of these algorithms. Most parallel decoding approaches use the length of the first codeword to detect the second codeword in parallel. By a single table lookup operation we detect multiple codewords without any detection mechanism.**

I. INTRODUCTION

Variable Length Decoding (VLD) is the most important part of the video standards like MPEG2/4 and H261/3. VLD is the first stage that feeds the rest of the processing in the pipe like IDCT, motion compensation, etc. So VLD computation throughput drives the throughput of the whole decode pipe. Rest of the processing in the decode pipe can be parallelized based on VLD throughput. Variable Length Coding (VLC), also known as Huffman coding, is a mapping process between source symbols and variable length code words. The variable length coder assigns shorter code words to frequently occurring source symbols, and vice versa, so that the average bit rate is reduced. In order to achieve maximum compression, the coded data is sent through a continuous stream of bits with no specific guard bit assigned to separate between two consecutive symbols. As a result, decoding procedure must recognize the code length as well as the symbol itself. VLD is carried out using tree-based [7,13] search and table lookup [1-2, 5,14,16] techniques. Table lookup based approaches decode at the symbol rate whereas tree based search algorithms decode at bit rate.

II. RELATED WORK

Various approaches have been presented to achieve high throughput VLD, such as parallel decoding [10,11] and adaptive tree search decoding [7]. Fig. 1 shows the traditional lookup based approach. Most decoding implementations use detection mechanisms [1,3] for decoding multiple symbols in parallel. The detection mechanism detects the code boundary of the first codeword and feeds the length to another parallel symbol decoder. This kind of mechanism is used in parallel/serial approach in [3,10] for decoding VLC in parallel. Although it is true that the bit stream needs to be decoded serially, the belief [2] that every symbol needs to be decoded before the following symbol can be decoded is misplaced. All these approaches [1-5,10,12,16-17] take a hardwired approach to speeding up VLD. Iwata & Deeley [2,4] implement a lookup table based approach on the FPGA Trimedia/CPU64 reconfigurable core and decode a single VLC per operation. Yang & Ishii [6,15] describe a parallel VLD approach on general purpose processors with SIMD instructions, however, it becomes infeasible for memory constrained embedded systems. Deeley & Yagi [4,12] present VLD processors which implement generic functions like get_bits(), put_bits(), etc. for VLD, whereas, Bublil [1] presents a VLD processor for MPEG1/2 video. Iwata & Deeley [2,14] have increased VLD speed by partitioning VLD tables into statistically optimized major/minor tables. Penna [5] presents a processor, which performs integer to float conversion on the bit stream. It then uses the mantissa and exponent so obtained to decode code words.

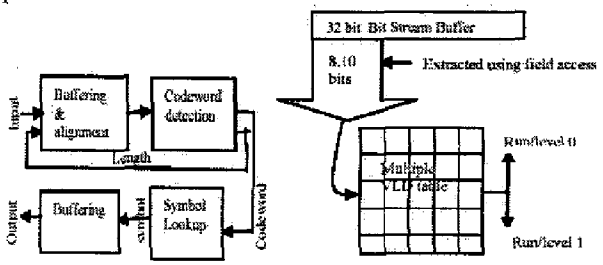

Fig. 1 Traditional VLD approach versus MDSP VLD approach

III. MULTIPLE VLD IMPLEMENTATION

Our approach is mainly targeted towards memory-constrained programmable embedded processors. We have implemented multiple-VLD [8,9] on MDSP, an array of RISC and Digital Signal Engines (DSE). Our approach does not detect the length of the first code word. It decodes two or more code words in a single access and uses the cumulative length to detect the next set of code words. In VLC, shorter code words are assigned to frequently occurring data while longer code words are assigned to infrequently occurring data. This idea is in turn exploited to create the packed Multiple-VLD tables. Each table contains multiple code words and signed level values in each entry. Each entry in the table is a 32-bit value. An example of part of MPEG-4 intra coefficients table is shown in Table 1. Another example of part of MPEG2 Table B-15 is also provided. Multiple such tables are required for implementation of each video standard. All the tables are indexed with less than or equal to 8-10 bits.

The software-only Multiple-VLD decoding is made easier due to a field-access unit available on each DSE. This helps to extract multiple bit fields of any length from the given 32 bits and facilitates sign extension. Using an index to the Multiple-

*We would like to thank Cradle Technologies Inc. for all the hardware support and Cradle Technologies India Development Center for all the software support.

0-7803-8838-0/05/$20.00 ©2005 IEEE.

VLD table, normally 8-10 bit value, we fetch the entry containing the multiple run-levels and the cumulative length as shown in fig. 1. This cumulative length is then used to align the 32-bit buffer to decode the next set of entries. The sign extension capability of the field access unit is exploited for sign extending the level values. No special instructions are provided by the architecture for VLC decoding.

The field access control register (see Fig. 2) allows signed extraction of 4 different bit lengths from the 32-bit packed entry/value as shown in Table 1.

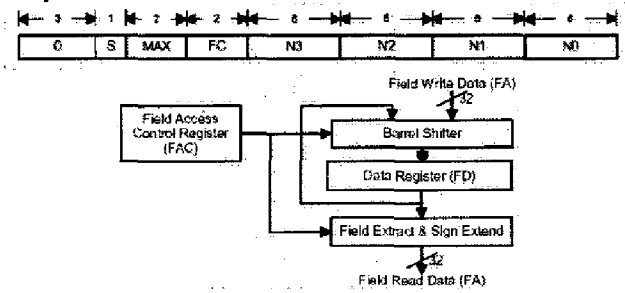

Fig. 2. Field Access Register Format and field access unit present on each DSE of the MDSP

In case of code words greater than 8 bits, they are indexed to proper Multiple-VLD table using branching in software. All tables are stored in local data memory and the implementation is thus not subject to data cache behavior.

Table 1 MPEG-4 Intra TCOEFF Table

Index	Level0 (6bits)	Level1 (6bits)	Run0 (6bits)	Run1 (6bits)	Length (6bits)	Last (2bits)
00100000	1	0	4	0	8	1
00100001	-1	0	4	0	8	1
...	...	...	...	...	...	...
10001110	1	1	0	0	8	1
10001111	1	-1	0	0	8	1
...	...	...	...	...	...	...
10010000	1	1	0	0	6	0
...	...	...	...	...	...	...
10010011	1	1	0	0	6	0
...	...	...	...	...	...	...
11110100	3	1	0	0	8	0
11110101	3	1	0	0	8	0
...	...	...	...	...	...	...

Table 2 MPEG2 MVLD Table (Table B-15 from MPEG-2 video standard)

Index	Run-0 (6bits)	Level-0 (6bits)	Run-1 (6bits)	Level-1 (6bits)	Valid Bit Count (8bit)
...	...	...	...	...	...
01000100	1	1	1	1	8
01000101	1	1	1	-1	8
...	...	...	...	...	...
01010100	1	-1	1	1	8
01010101	1	-1	1	-1	8
...	...	...	...	...	...
01001000	1	1	0	1	7
...	...	...	...	...	...
01001011	1	1	0	-1	7
...	...	...	...	...	...
01011000	1	-1	0	1	7
...	...	...	...	...	...
01011011	1	-1	0	-1	7
...	...	...	...	...	...
01110100	0	3	0	1	8

IV. Conclusion

We have demonstrated the implementation of the software only multi-symbol variable length decoder. We have implemented this method in H261/3, MPEG2/4 real-time video decoders on the MDSP CRA2003[3] & CRA3001[3] evaluation boards. Our approach decodes multiple symbols whenever allowed by the bit stream without detecting the length of the previous symbol. This approach is generic and can be used in high-throughput video applications. We achieve a high-throughput, multiple symbols per cycle decoding in software-only for video applications. We found that the general implementation is flexible and easily adaptable across standards in terms of code reuse. VLD accelerator processing cores need to be tightly coupled to the instruction pipeline and are part of the core of the main processor. Our solution is a loosely coupled data parallel approach to decoding compared to instruction parallel approach of VLIW.

References

[1] Moshe Bublil, Subroto Bose et. al., "Variable Length Decoder for Digitally Encoded Video Signals", *United States Patent 6704361*, 2004.

[2] Haruya Iwata,"Variable Length Decoder and Video Decoding Apparatus Using the Same", *United States Patent 6738525*, 2004.

[3] Eric Yang, "Multi –Symbol Variable Length Code Decoder", *United States Patent 0210163*, 2003.

[4] Reechard Deeley, Yatin Mundkur, Woobin Lee.,"Processing Circuit and Method For Variable Length Coding and Decoding", *United States Patent 6507293*, 2003.

[5] David Penna,"Variable Length Decoder", *United States Patent 0118114*, 2003.

[6] Jia Quan Yang et. al.,"Method of MPEG-2 Video Variable Length Decoding in Software", *United States Patent 0118115*, 2003.

[7] Vivian Hsiun,"Programmable Variable Length Decoder", *United States Patent 0184457*, 2003.

[8] Ganesh Yadav, R.K. Singh, Vipin Chaudhary, "MAVD: Mpeg-2 Audio Video Decode Sytem on MDSP™", *Proc. of the IEEE Intl. Symposium on Consumer Electronics*, pg. 19-24, 2004.

[9] Ganesh Yadav, R.K. Singh, Vipin Chaudhary, "On Implementation of MPEG-2 like Real-Time Parallel Media Applications on MDSP™ SoC Cradle Architecture",*Proc. of Intl. Conference on Embedded and Ubiquitous Computing*, pg. 281-290, 2004, Springer-Verlag.

[10] Jari Nikara, Stamatis Vasiliadis, Jarmo Takala et.. al.,"Parallel Multiple-Symbol Variable-Length Decoding", *Proc. of IEEE Intl. Conf. on Computer Design: VLSI in Computers and Processors*, 2002.

[11] Mihai Sima et. al.,"MPEG-Compliant Entropy Decoding on FPGA-augmented Trimedia/CPU64", *Proc. of IEEE Symposium on Field-programmable Custom Computing Machines*, 2002.

[12] Osamu Yagi,"High Speed Variable Length Decoding Processor", *United States Patent 6459391*, 2002.

[13] Bai-Jue Shieh e. al., "A High Throughput Variable Length Decoder with Modified Memory Based Architecture", *Proc. of IEEE ISCAS* 1998.

[14] Chungrong Zhu,"Method and Apparatus of Decoding Variable Length Codes", *United States Patent 5821887*, 1998.

[15] Daiji Ishii et. al.,"Parallel Variable Length Decoding With Inverse Quantization for Software MPEG2 Decoders", *IEEE Workshop on Signal Processing and Systems*, pg. 500-509, 1997.

[16] Yasuhsi Ooi et. al.,"A 162 MBits/s Variable Length decoding Circuit using Adaptive Tree Search Technique", *Proc. of IEEE Custom Integrated Circuits Conference*, pg. 107-110, 1994.

[3] Cradle Technologies Inc. sells a more powerful CT3400 chip now.