EFFICIENT VLSI IMPLEMENTATION OF A VLC DECODER FOR UNIVERSAL VARIABLE LENGTH CODE

Shang Xue and Bengt Oelmann

Department of Information Technology and Media, Mid Sweden University
SE-851 70 Sundsvall, Sweden
xue.shang@mh.se

ABSTRACT

Variable length code (VLC) is used in a large variety of lossless compression applications. A specially designed VLC, called "Universal Variable Length Code" (UVLC), is utilized in the latest video coding standard H.26L under development. In this work we develop an efficient decoder for UVLC by utilizing the special properties of UVLC which perform coding in an alternating way (ALT). We compare the ALT decoder with the decoder called "VLC decoder using plane separation" (PLS) which is claimed to be one of the most effective VLC decoders. Our results show that the ALT decoder is 1.34 times faster, 1.7 times smaller, and consumes 45% power in comparison to the PLS decoder.

1. INTRODUCTION

The video coding standard H.26L uses a unique variable length code pattern (VLC) which is called Universal Variable Length Code (UVLC) to perform entropy coding It was first proposed in [1, 2]. Although VLCs are efficient in compression, the variable code lengths of VLCs also limit the decoding throughput. VLC decoders are usually implemented with look-up tables and a shifting scheme [4,5]. Look-up tables and the shifting scheme occupy the largest portion of the area and are also the two performances limiting components of speed and power. With the development in mobile video communications, the construction of smaller, faster, and less power-consuming video CODECS becomes increasingly important. In this paper we present a new type of UVLC decoder. It takes advantage of the special properties of UVLC and performs coding in an "Alternating" way (ALT decoder). It does not contain look-up tables, and the sizes of barrel shifters as the shifting scheme are greatly reduced. Therefore it is faster, much smaller and less power-consuming. We compare the performances of the proposed UVLC decoder with a decoder developed by Jae Ho Jeon et al. [6], under the name of "Fast Variable-Length Decoder Using Plane Separation" (PLS decoder), which was claimed to be one of the most effective VLC decoders. Results show that the ALT decoder is 1.34 times faster, 1.7 times smaller, and consumes 45% power compare to the PLS decoder.

2.UVLC CODE PROPERTIES

Figure 1 illustrates the coding pattern of UVLC [3].

$$1$$
$$0\ x_0\ 1$$
$$0\ x_1\ 0\ x_0\ 1$$
$$0\ x_2\ 0\ x_1\ 0\ x_0\ 1$$
$$\cdots\cdots\cdots\cdots\cdots$$

Fig. 1: UVLC coding pattern

Each UVLC represents a code number n. The odd-indexed bits (OIB) of UVLC can be looked on as a unary expression. The even-indexed bits (EIB), as indicated by x_n in Fig 1, are simply some binary codes. In coding the unary OIBs, two different sets of alternative codes are applied, all-one and all-zero codes. They are used in an alternating way in the coding procedure so that the codeword boundaries can be easily determined by detecting the value changes in an OIB series. The EIBs are a set of binary codes whose length is determined by their corresponding odd-indexed parts. Therefore each code number can be represented as:

$$code\ number = 2^{length(OIB)-1} - 1 + decimal(EIB) \quad (1)$$

For example, if we have a UVLC series $001\ 00011\ 011\ 01011$, we will get one OIB series and one EIB series after coding in an alternating way. The OIB series will be $11\ 000\ 11\ 000$, and the EIB series will be $0\ 01\ 1\ 10$.

3.ALT DECODER

The ALT decoder proposed in this paper is based on the previous analysis of code properties of UVLC. The maximum codeword length here is set to be 31 bits in order to cover adequate number of codewords.

The ALT decoder consists of two decoders, one is the OIB decoder and the other is the EIB decoder. The architecture of the OIB decoder is described in Figure 2. The core of the decoder are one 16-to-4 priority encoder (PE_0), and one 4-to-16 decoder (DEC_0).

The OIB input of the decoder is put into the two buffers D_0 and D_1. The first two-byte OIB series is then fed to the "Boundary Detection Logic" (BDL), where two consecutive bits are xored with each other. At each OIB boundary a "1" will be generated. The output after the BDL is then fed into the priority encoder PE_0 in order to generate the

Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI'03)
0-7695-1904-0/03 $17.00 © 2003 IEEE

position of the first OIB boundary. The length of EIB is then calculated by the "EIB Length Calculation Logic" (ELCL). The DEC_0 generates the position of the first OIB boundary and disables the first "1" of the input of the priority encoder by using the or-gates and the "Codeword Disabling Logic" (CDL). In the next clock cycle, the next OIB boundary is encoded by PE_0. The same operations repeats. The "'load' Signal Generation Logic" (LGL) generates the load signal.

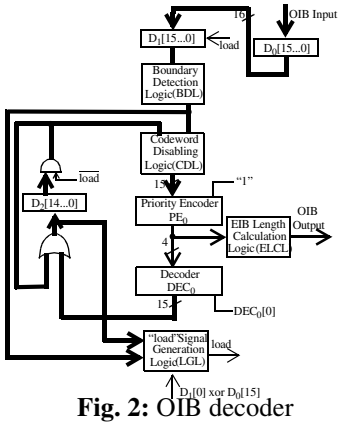

Fig. 2: OIB decoder

The EIB decoder is illustrated in Fig 3. The EIB series is first loaded in the lower half of the two barrel shifters, the first 15 bits in BS_0 and the following 15 bits in BS_1. The upper half of the barrel shifters are both loaded with 15 bits zeros. D_8 is originally loaded with "1111". BS_0 shifts the EIB series to the upper half of it according to the EIB length generated from the OIB decoder and gets EIB shifted out. SUB_2 outputs the length of the rest of the EIB series. In the next clock cycle, the lower half of BS_0 is loaded with the shifted EIB series and the upper half is cleared into all zeros. Therefore the decoding of the next EIB can be performed. The same operations are then repeated. Comparator $COMP_1$ is used to detect the end of the EIB series. The contents of BS_0 and BS_1 are both shifted according to the length of the next EIB. The two separated parts of last EIB in BS_0 can be connected by an or-gate and MUX_1 so that the complete EIB can be generated. MUX_2 is used to load new data into BS_0. Decoding can then be performed continuously.

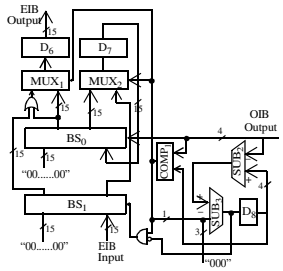

Fig. 3: EIB decoder

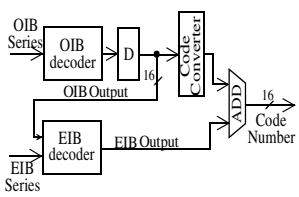

Fig. 4: ALT decoder

The complete architecture of the ALT decoder is shown in Fig 4. The function of this code converter is to change the length of the EIB (i.e. *length(OIB)-1*) into $2^{length(EIB)}-1$ so that the code number can be calculated as mentioned in relation (1). Decoding is then completed.

4.COMPARISON OF PERFORMANCE

The ALT decoder is compared with the PLS decoder developed by Jae Ho Jeon et al.[6].

We compare the delay, area and power consumption of the ALT decoder to those of the PLS decoder. Both of the decoder types have been implemented in VHDL and synthesized using *Design Compiler* from Synopsys. The delay has been obtained from static timing analysis and the figures for power consumption from Synopsys' *Power Compiler*. A standard cell library in a 0.5µm CMOS process has been used. The results are shown in Table 1.

Table 1. Comparison of performance

	ALT	**PLS**	**Ratio (ALT/PLS)**
Delay (ns)	8.96	12.0	75%
Area (gates)	1855	3146	59%
Power (mW)	6.74	15.0	45%

5.CONCLUSIONS

We propose the ALT decoder for decoding UVLC. This decoder takes advantage of the properties of UVLC which simplifies the decoding procedure. It can be seen that the ALT decoder is 1.34 times faster, 1.7 times smaller, and consumes 45% power compare to the PLS decoder.

6. REFERENCES

[1] Y. Itoh, Ngai-Man Cheung, "Universal variable length code for DCT coding," in *International Conference on Image Processing*, vol. 1, pp. 940-943, 2000.

[2] N.-M. Cheung, Itoh, Y., "Configurable variable length code for video coding," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 1805-1808, 2001.

[3] ITU-T, *H.26L TML8 Document from http://standard.pictel.com*, Sep., 2001.

[4] M. T. Lei, M. T. Sun, "An entropy coding system for digital HDTV applications," in *IEEE Trans. Circuits Syst. Video Technol.*, vol. 1, no. 1, pp. 147-155, March 1991.

[5] H. D. Lin, D. G. Messerchmitt, "Designing high-throughput VLC decoder Part II-Parallel decoding methods", in *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, pp. 197-206 June 1992.

[6] Jae Ho Jeon et al, "A fast variable-length decoder using plane separation," in *IEEE. Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 806-812, Aug. 2000.