



The Master of IEEE Projects

LeMenizInfotech

36, 100 Feet Road, Natesan Nagar, Near Indira Gandhi Statue, Pondicherry-605 005.

Call: 0413-4205444, +91 9566355386, 99625 88976.

Web : [www.lemenizinfotech.com/](http://www.lemenizinfotech.com/) [www.ieeemaster.com](http://www.ieeemaster.com)

Mail : [projects@lemenizinfotech.com](mailto:projects@lemenizinfotech.com)

## Code Compression for Embedded Systems Using Separated Dictionaries

### Abstract:

Engineers must consider performance, power consumption, and cost when designing embedded digital systems; furthermore, memory is a key factor in such systems. Code compression is a technique used in embedded systems to reduce the memory usage. BitMask-based code compression is a modified version of dictionary-based code compression. The basic purpose of BitMask is to record mismatched values and their positions to compress a greater number of instructions; it can be used exclusively or incorporated with the reference instructions to decode the codewords. In this paper, we applied a small separated dictionary, and variable mask numbers were used with the BitMask algorithm to reduce the codeword length of high frequency instructions. In addition, a novel dictionary selection algorithm was proposed to increase the instruction match rates. The fully separated dictionary method was used to improve the performance of the decompression engine without affecting the compression ratio (CR) (the compressed code size divided by original code size). Based on the experimental results, the proposed method can achieve a 7.5% improvement in the CR with nearly no hardware overhead. The proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.2.

### Enhancement of the project:

Reduce the power and area too more than the existing system.

### Existing System:

Dictionary-based code compression (DCC) is commonly used in embedded systems, because it can achieve an efficient CR, possess a relatively simple decoding hardware, and provide a higher decompression bandwidth than the code compression by applying lossless data compression methods. Thus, it is suitable for architectures with high-bandwidth instruction-fetch requirements, such as the very long instruction word (VLIW) processors. Although several existing code compression algorithms have exhibited favorable compression performance, no single compression algorithm has efficiently worked for all kinds of benchmarks. In this paper,





## The Master of IEEE Projects

### LeMenizInfotech

36, 100 Feet Road, Natesan Nagar, Near Indira Gandhi Statue, Pondicherry-605 005.

Call: 0413-4205444, +91 9566355386, 99625 88976.

Web : [www.lemenizinfotech.com/](http://www.lemenizinfotech.com/) [www.ieeemaster.com](http://www.ieeemaster.com)

Mail : [projects@lemenizinfotech.com](mailto:projects@lemenizinfotech.com)

various steps in the code compression process were combined into a new algorithm to improve the compression performance (including the CR) with a smaller hardware overhead. Based on the BitMask code compression (BCC) algorithm, a small separated dictionary is proposed to restrict the codeword length of high-frequency instructions, and a novel dictionary selection algorithm is proposed to achieve more satisfactory instruction selection, which in turn may reduce the average CR. Furthermore, the fully separated dictionary architecture is proposed to improve the performance of the dictionary-based decompression engine. This architecture has a better chance to parallel decompress instructions than existing single dictionary decoders.

### Disadvantages:

- Low decompression bandwidth
- Compression ratio is small

### Proposed System:

Proposed algorithms are described. A separate dictionary was used to reduce the codeword length of high-frequency instructions. Variable mask numbers were used to eliminate the encoding redundancy. The combination of these methods is called as the CLCBCC. A modified version of a MBSDS algorithm from was used to select an improved instruction combination for the dictionary. Compared a fully separated dictionary architecture is proposed to reduce the access latency of the dictionary.

### Separated Dictionaries

In certain cases, such as in low code density architecture, which contains a high number of unique instructions or because of algorithmic characteristics, a large LUT is required to compress the programs. A large LUT has several disadvantages: it requires a large chip area, additional power consumption, a long LUT latency, and a long codeword length. Thus, it is desirable to minimize the dictionary size.

### Variable Mask Numbers

Seong and Mishra surveyed the size and combination of the masks and they concluded that a 4-bit fixed (4f) and a 1-bit sliding (1s) mask achieves an optimal CR. However, Wang and Lin determined that using a 4-bit fixed and a 2-bit fixed masks in addition to a single 4-bit fixed





## The Master of IEEE Projects

### LeMenizInfotech

36, 100 Feet Road, Natesan Nagar, Near Indira Gandhi Statue, Pondicherry-605 005.

Call: 0413-4205444, +91 9566355386, 99625 88976.

Web : [www.lemenizinfotech.com/](http://www.lemenizinfotech.com/) [www.ieeemaster.com](http://www.ieeemaster.com)

Mail : [projects@lemenizinfotech.com](mailto:projects@lemenizinfotech.com)

mask achieves better results for the benchmarks. Although the maximum mask overhead was 13 bits (4 bits for 4-bit mask, 3 bits to record the position of the 4-bit fixed mask, 2 bits for 2-bit mask, and 4 bits to record the position of the 2-bit fixed mask), it was determined that ~50% of the instructions were compressed using only the 4-bit fixed mask in the benchmarks.

### Mixed Bit Saving Algorithm

FDS cannot achieve an optimal CR in BCC, because it cannot guarantee that the matched rate of high-frequency instructions is maximized. The proposed dictionary selection algorithm is based on the graph representation. The instructions are transformed into nodes, and an edge between two nodes indicates that these two instructions have been matched to each other using the BitMask approach.



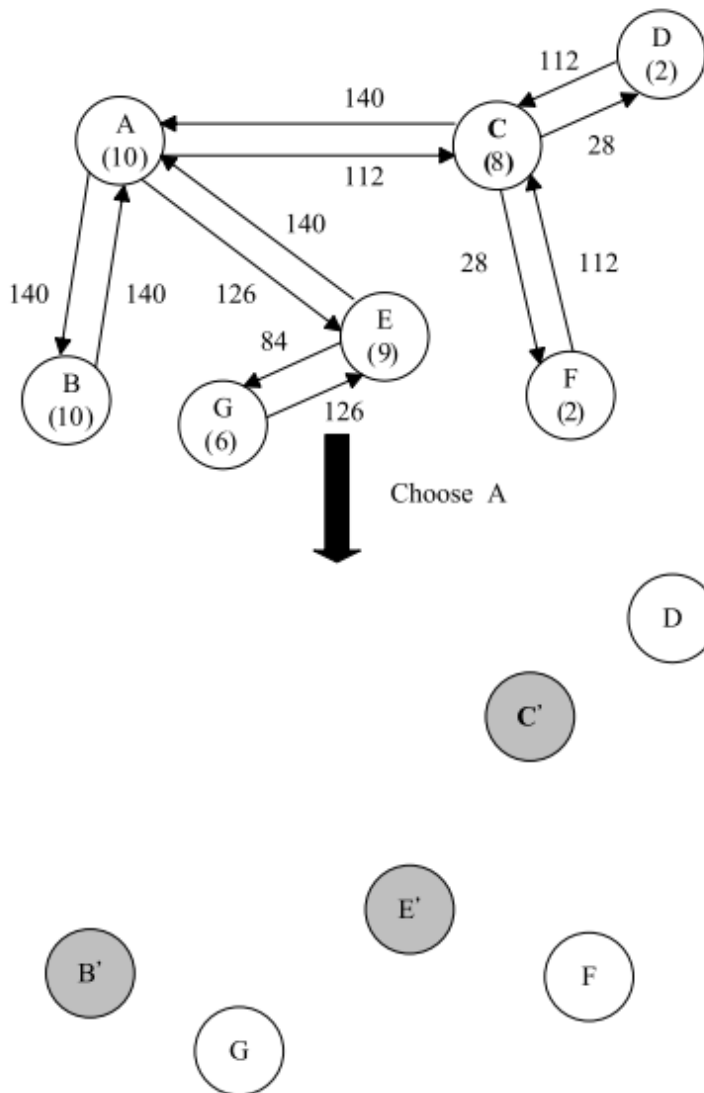


Fig. 1. MBSDS.

Fig. 1 shows an example of selection using MBSDS. All symbols in this example are 32-bit wide, the dictionary contained 1024 entries, only one 2-bit mask was used, and the overhead of the identification tag is 2-bit.

Decompression Engine



## The Master of IEEE Projects

The proposed decompression engine was implemented using the Verilog hardware description language and synthesized using a Synopsys' Design Compiler and a TSMC 0.13- $\mu\text{m}$  cell library. The decompression engine, the logic diagram of which is shown in Fig. 2 consisted of a control unit, a demultiplexer, shift buffers, LUTs, and the BitMask unit.

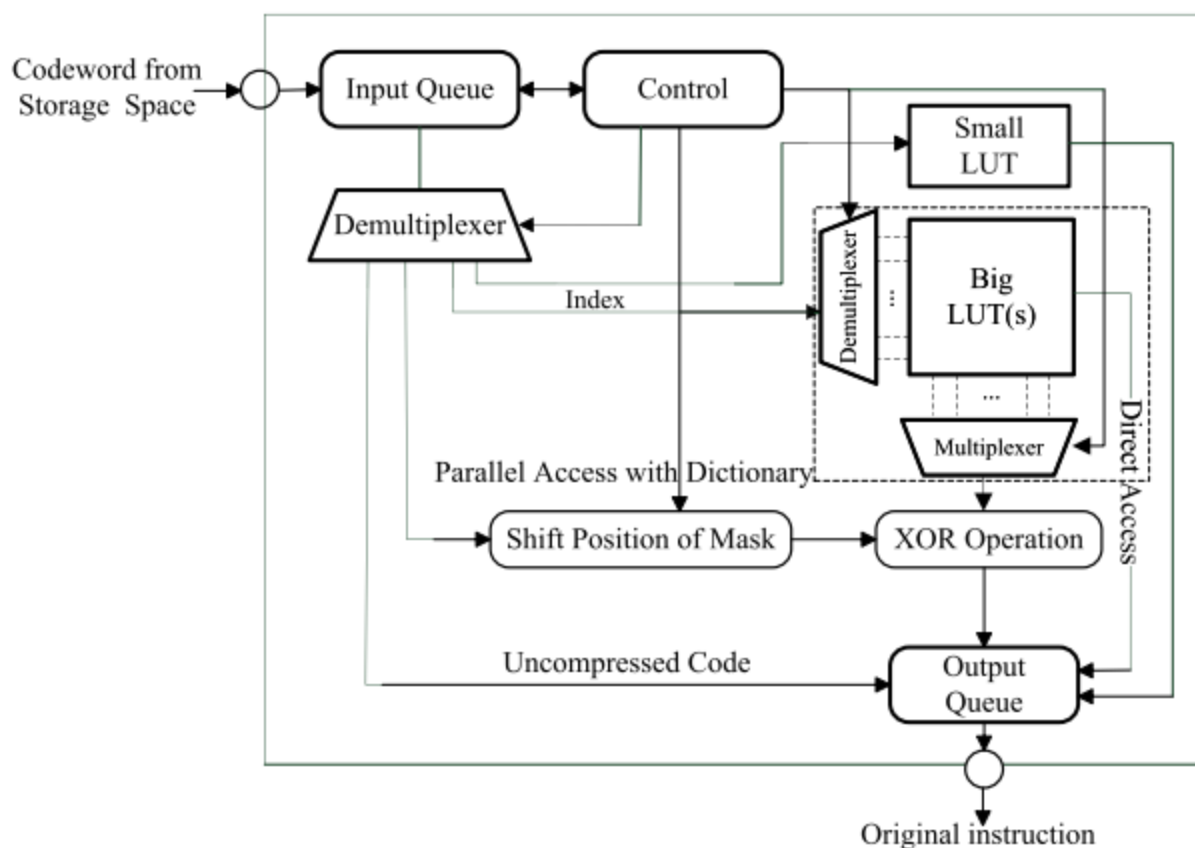


Fig. 2. Logic diagram of decompression engine.

The fully separated dictionary architecture is proposed as a solution to this problem. As shown in Fig. 3, an  $n$ -entry LUT is separated into  $i$   $n/i$ -entry LUTs. Another selector is used to decide which LUT should be accessed. The router is used to route the exact LUT entry to the output buffer.



## The Master of IEEE Projects

The original compressed codeword requires  $\log_2(n)$  + identify codeword type tag(s) bits in the DCC. The modified LUT architecture requires  $\log_2(i)$  bits to inform the selector and router as to which LUT must be accessed to receive the instructions.

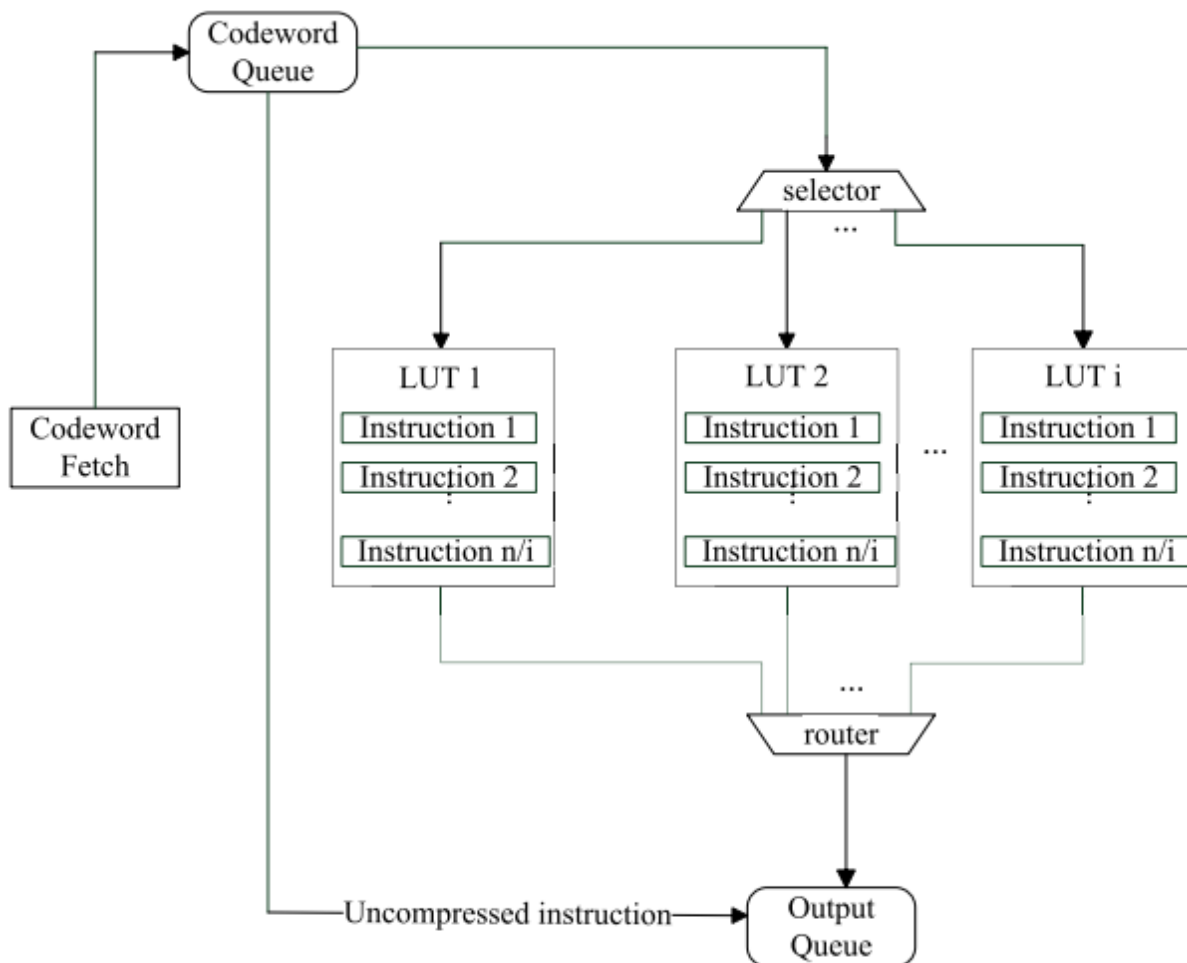


Fig. 3. Fully separated dictionary architecture.

### Advantages:

- increase the decompression bandwidth





### **The Master of IEEE Projects**

- improve the performance of the decoder
- improve the Compression ratio

### **Software implementation:**

- Modelsim
- Xilinx ISE

### **LeMenizInfotech**

**36, 100 Feet Road, Natesan Nagar, Near Indira Gandhi Statue, Pondicherry-605 005.**

**Call: 0413-4205444, +91 9566355386, 99625 88976.**

**Web : [www.lemenizinfotech.com/](http://www.lemenizinfotech.com/) [www.ieeemaster.com](http://www.ieeemaster.com)**

**Mail : [projects@lemenizinfotech.com](mailto:projects@lemenizinfotech.com)**

