EVALUATION OF DLX MICROPROCESSOR INSTRUCTIONS EFFICIENCY FOR IMAGE COMPRESSION

Nimas Fatihah[1], Nyoman Karna[2], Raditiana Patmasari[3]

[1,2,3]Telecommunication Engineering Department, School of Electrical Engineering, Telkom University

[1]**nimsekar@student.telkomuniversity.ac.id**, [2]**aditya@telkomuniversity.ac.id**, [3]**raditiana@telkomuniversity.ac.id.**

Abstract

Internet of things (IoT) nowadays uses a generic microprocessor, which is applicable for general purpose and produces many machine instructions. Likewise, IoT can also be integrated on ASIC (Application-specific integrated circuit) which is customized for partial use. ASIC is hardcoded, meaning that the program cannot be modified, therefore it tends to consume less power compared to generic microprocessor. This thesis considers a compression for an image of CCTV, which is using a microprocessor that is designed for a specific and general purposes as the compression. Compressing image is required to reduce the size of the original image. This thesis uses the Deluxe (DLX) microprocessor with a high performance to design an image compressor, and the machine instructions were determined with a specific algorithm. The compression uses Joint Photographic Experts Group (JPEG) format lossy compression, which is the most commonly used to compress multimedia data. The proposed compression method is Huffman Coding, coded in the assembly DLX programming language. DCT and Quantization are needed to be simulated in image processing tool to do the Huffman coding process. Then, the result data can be processed into Huffman. The result of this stage is by using Huffman Coding in the DLX microprocessor, it requires total of 11657 cycles executed by 8622 instructions. Thus, with such specific machine instructions, the performance of DLX microprocessor to execute Huffman Coding can be efficient.

Keywords: IoT, DLX microprocessor, Huffman Coding, image compression, JPEG.

1. Background

IoT is a physical network that interconnected devices by using Wireless Sensor Network (WSN). Tools that are used for IoT are electronic stuff, sensors, actuators, etc which is embedded in a platform software to connect and enable data exchange. With the IoT, our daily activities can be done easily without having to worry about the situation and condition, because it can be controlled from a certain distance in a real-time interface or virtual interface.

People's daily activities are getting hectic, thus the growth of IoT demand is rising rapidly. As shown in Figure 1.1, IoT can be specified into some usability, such as security and surveillance, supply chain management, inventory and warehouse, customer order, facility management, industrial asset management, smart products, energy management, and fleet management. Each demand can be quantified on the given parameters such as cool, warm, hot, hotter, the hottest.

Security and surveillance IoT implementation is the most indispensable one for daily necessity, preeminently for crime prevention with the major utility are on Government, education, and social services [1]. The most common surveillance tool to implement is CCTV. The principal work of CCTV is the use of video cameras to transmit signal and data into specific place on a limited set of monitors [2].

The use of CCTV is to report the situation by sending the video which requires a large amount of memory, high performances, and trace many machine instructions. Thus, the aim of this thesis is making a program for CCTV with specific machine instructions, to only report the situation by sending a compressed picture which is simulated in the DLX Microprocessor that has Reduced Instruction Set Computer (RISC) [3]. With the IoT which uses wireless sensor network and a microprocessor, it can be effectively implemented for CCTV to make the performance more efficient.

2. Basic Concept

2.1 Internet of Things

IoT is the network of physical device where it's a concept when an object has the capability of transmitting data through the WSN and providing open access, it doesn't require the interaction of human to human or human to device, thus people or device can control the IoT device directly even in a distance [4].

The recent communication of IoT envisions a near future equipped with microcontrollers, transceivers for digital communication, and suitable protocol stacks [5]. The term of IoT is closely related with sensor technologies, wireless technologies, as the method of communication and it's resulting in improved efficiency, and accuracy. The efficiency means that when people using IoT device it doesn't need to be wired to be connected to power supply because the power can be changed with battery to avoid profuse use of power, it also doesn't need to be wired to another device because the connected devices are being embedded with wireless sensor network which can directly recording and monitoring physical condition and collecting its data such as temperature, humidity, sound, and another physical changing.

2.2 Microprocessor

Microprocessor is a very small processor, it is a Central Processing Unit (CPU) that made from integrated circuit. Microprocessor is also called as a brain of the computer because it must calculate and control other devices. Microprocessor is basically made up from several parts to do its main functions there are Arithmetic Logic Unit (ALU), Control Unit (CU), and Register Unit:

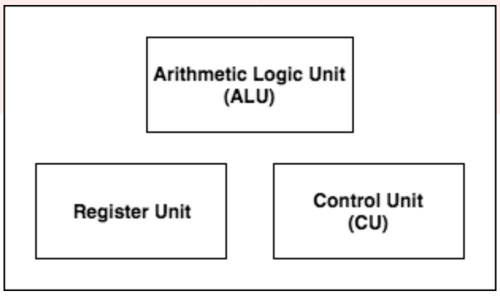

Figure 2. 1 Inside of Microprocessor

ALU works to do mathematical and logical operations, for mathematical operations include addition, subtraction, multiplication, and division, while the logic operations are AND, OR, XOR, NOT, and others. CU handles to take instructions from memory then executes the instructions and manages all the process in microprocessor. Register Unit is a location to accommodate the temporary data from memory before it is processed by the ALU.

2.2.1 DLX Microprocessor

The DLX (pronounced "Deluxe") is a Reduced Instruction Set Computer (RISC) processor architecture designed by John L. Hennessy and David A. Patterson, the principal designers of Stanford MIPS and the Berkeley RISC design. The character of DLX architecture is storage efficient, with the use of a simple load instruction set, and targeting effective and efficient compiler results. Furthermore, the goals of DLX microprocessor design must be less power consumption and has a higher capability performance [6].

The DLX instructions are divided into 3 types, there are R-type, I-type and J-type. R-type instructions are pure register instructions consist of 2 register references that has 32-bit word. I-type instructions define 2 registers and use 16 bits to hold immediate value. J-type is a jump instruction containing 24-bit address.

1. **Register-Register (R-type) Instruction**

6 bits	5 bits	5 bits	5 bits	11 bits
Opcode	rs1	rs2	rd	func

0 5 6 10 11 15 16 20 31

Register-register ALU operations: rd ← rs1 func rs2. Function encodes the data path operation: Add, Sub.

Read/write special registers and moves.

2. **Register-Immediate (I-type) Instruction**

6 bits 5 bits 5 bits 16 bits

Opcode	rs1	rs2	Immediate

0 5 6 10 11 15 16 31

Encodes: Loads and stores of bytes, words, half words. All immediates (rd ← rs1 op immediate)
Conditional branch instructions (rs1 is register, rd unused)

Jump register, jump and link register (rd = 0, rs = destination, immediate = 0)

3. **Jump / Call (J-type) Instruction)**

6 bits	26 bits
Opcode	Offset added to PC

0 5 6 31

Jump and jump and link. Trap and return from exception

Opcode is the stage of converting instruction set architecture into machine language for microprocessor to know the meaning of instructions to be performed. Opcodes are 6 bits long with total 64 possible basic instructions. If user wants to select one of 32 registers it needed 5 bits. The DLX supports more than 64 instructions under the condition that the instructions work purely on registers.

Because of DLX capability high performance, there were many previous research using its architecture. The Intel Microprocessor Laboratory and The Microelectronics and Microprocessor Laboratory have been organizing research of Microprocessor. Previous DLX project was resulting optimization of power, speed, and area with using each different technique. The progress of these project has lead to the invention of Base DLX design that can be adjusted for each need.

2.3 Image Compression

Image compression is the process of data compression on the digital images in order to reduce the size of the original data. The algorithm affects visual perception to provide results compared with generic compression method.

The method of image compression is divided into lossy and lossless. The difference between those two are lossy compression is specialized for compressing photograph images to achieve a substantial reduction in bit rate. On the other hand, lossless compression is for archival purpose such as technical drawings, medical imaging, and clip art.

2.4 Joint Photographic Experts Group (JPEG)

JPEG is one of picture format that its compression method is using lossy compression for photography digital images. It reduces the size from the original image into smaller size and the original image can't be stored and it will be affecting image duality, but the degree of the compression can be adjusted based on the image quality that will be chosen.

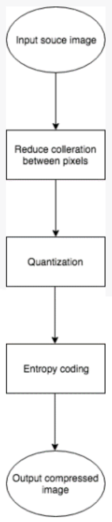

Figure 2. 2 Encoding steps of image compression

Main components form a typical lossy image compression system [7] and the steps of image compression are:

a. Source Encoder
Decorrelating the input image by transforming into set of sparse data value as smaller number of coefficients. It can be done by Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Discrete Fourier Transform (DFT).

b. Quantizer
It's for reducing the number of bits that needed to store transformed coefficients by decreasing the precision of the value.

c. Entropy Coding
Entropy Coding removes repeated bit patterns in the output of quantizer. The most common entropy coders are Huffman Coding, Arithmetic Coding, Lempel-Ziv (LZ), and Run Length Encoding (RLE) algorithm.

The algorithm for JPEG compression is specialized on digital photograph with soft variations of color and tone, there are also sharp contrast on each pixel. The compression is based on the DCT, it converts each frame of the 2D video into a specific transform domain which is quantization or reducing information such as a large number of scale into a smaller one.

2.4.1 DCT

Discrete Cosine Transform (DCT) is transformation model on discrete function that only take a part of cosines on exponential complex. This method is the international standard lossy image compression for JPEG. JPEG compression using DCT is for processing every 8x8 pixel data and differing frequencies. The feature of DCT is that significant changes in image information are concentrated only on a few DCT coefficients.

The original image with obtained RGB components are divided into 8x8 blocks. Each of 8 bits has range from 0 which is pure black to 255 as white. Data range is rounded off to zero by shifted [0, 255] to [-128,127]. By DCT, parts inside the image are separated into different frequencies. The quantization step removes less important frequencies and the decompression step uses the important frequencies to retrieve the image. The formula for DCT as follows:

$$[F]u,v = ([DC]u,x\,.\,[f]x,y).\,[DC]y,v \tag{1}$$

where,

$$DC(u,x) = \sqrt{\frac{2}{N}}\, Cu \cos\left[\frac{(2x+1)u\pi}{2N}\right] \tag{2}$$

$$DC(y,v) = \sqrt{\frac{2}{N}}\, Cv \cos\left[\frac{(2x+1)v\pi}{2N}\right] \tag{3}$$

(1) the forward 2D_DCT transformation
(2) calculation of DCT
(3) transpose calculation from *DC(u,x)*

2.4.2 Huffman Coding

Huffman Coding is one of lossless compression technique, it is an entropy coding part for compressing JPEG [8]. As a kind of lossless compression coding, Huffman Coding has higher compression efficiency than any other methods. This method based on the probability of occurrence of symbols, then make it as characters to construct a different prefix code word with the shortest average length [9]. It begins with comparing weights to build a Huffman tree, and then encode and decode it.

The steps to do the Huffman coding are:

1. Obtain quantized frequency on each pixel that has been done by DCT.
2. Select the lowest to highest frequency on the pixel.
3. Construct a Huffman tree from the lowest frequency until all frequency implied from bottom up.
4. Make a binary code by putting 0 on the left branch of the tree and 1 on the other branch on each pixel frequency.
5. Each pixel frequency now has each binary code.
6. Gather and sum up all the binary codes to get the final lower bits than the original one.

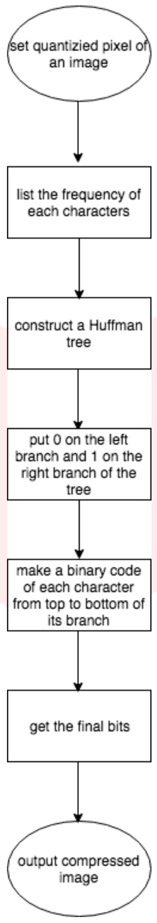
set quantizied pixel of an image
list the frequency of each characters
construct a Huffman tree
put 0 on the left branch and 1 on the right branch of the tree
make a binary code of each character from top to bottom of its branch
get the final bits
output compressed image

Figure 2. 3 Steps of Huffman Coding.

3. System Model and Proposed Technique

3.1. System Model

3.1.1 Block Diagram

The block diagram below shows the complete system of how JPEG is being compressed with Huffman coding and using DLX emulator. The first step is set the high quality of an image and transforming the image into set of sparse data value as smaller number of coefficients with DCT. After DCT, Quantization decides the quality level of the image. The last is doing the Huffman coding with assembly DLX machine instruction and get the output of binary code.

After the Huffman coding, the next step is doing a simulation in DLX emulator, then evaluate the instructions.

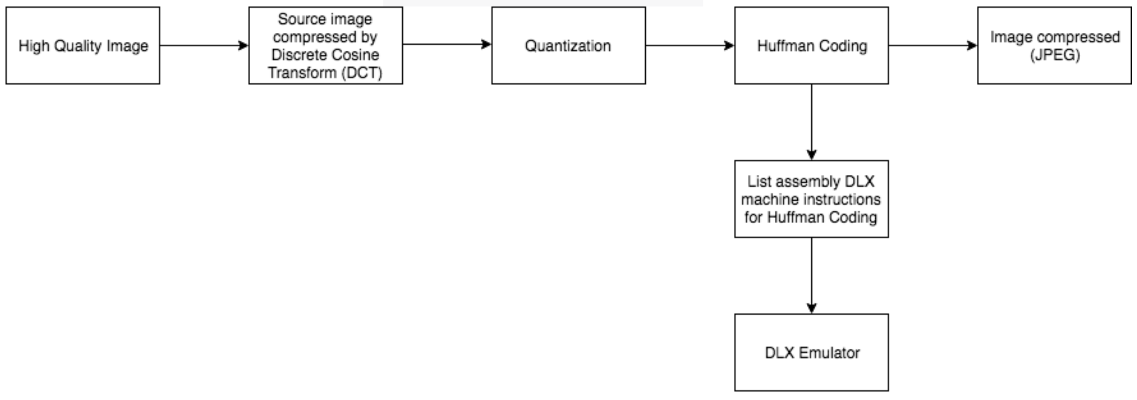
High Quality Image
Source image compressed by Discrete Cosine Transform (DCT)
Quantization
Huffman Coding
Image compressed (JPEG)
List assembly DLX machine instructions for Huffman Coding
DLX Emulator

Figure 3. 1 Block Diagram of JPEG compression using Huffman Coding and DLX emulator.

To create the program for JPEG compression, author decides which of the JPEG compression steps will be executed, then in this thesis Huffman Coding is chosen. Furthermore, after getting binary code from the result of Huffman coding, the next step is selecting the necessity machine instruction to be represented whether it is optimal or not. If it is not, then design the machine instruction part should be redone until the most optimal instruction is obtained.

4. Conclusion

The DLX microprocessor has 92 instruction in 6 classes, but this experiment did not require all instructions to be used. Some chosen instructions are only add, addi, subi, lw, sw, beqz, bnez, seq, sgt, and trap. On the sub process of input the value of quantization program used instructions 18 addi(s), 1 jal, 2 beqz(s), 2 bnez(s), 2 sw(s), and 1 trap. For sub process counting frequency program used instructions 3 lw(s), 7 sw(s), 5 beqz(s), 4 bnez(s), 1 seq, 9 addi(s), 2 add(s), 3 subi(s), and 1 trap. The last sub process is sorting frequency program which used instructions 9 lw(s), 31 addi(s), 11 add(s), 6 beqz(s), 8 bnez(s), 1 seq, 3 sgt(s), and. 1 trap.

The evaluation of the DLX microprocessor performance is based on statistic result on each sub-process of Huffman Coding specific instructions. Thus, the fewer the instructions, the more efficient the performance can be achieved.

Reference:

[1] H. Borrion, H. Deghanniri, Y. Li, “Comparative Analysis of Crime Scripts: One CCTV Footage–Twenty-One Scripts”, European Intelligenve and Security Informatics Conference, Athens, Greece, September 2017.

[2] S. Arora, K. Bhatia, Amit V, “Storage Optimization of Video Surveillance from CCTV Camera”, International Conference on Next Generation Computing Technologies (NGCT-2016), Dehradun, India, October 2016.

[3] V. Valero, F. Cuartero, A. Garrido, F. Quiles, Conference of IEEE, Washington, DC, USA, USA, “A Simulation Tool of Parallel Architectures for Digital Image Processing Applications Based on DLX Processors”, October 1995.

[4] H. Aksu, L. Babun, M. Conti, G. Tolomei, A. Selcuk Uluagac, “Advertising in the IoT Era: Vision and Challenges”, IEEE Communications Magazine (Early Access), April 2018.

[5] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi, “Internet of Things for Smart Cities”, IEEE Internet of Things Journal, Vol. 1, No.1, February 2014.

[6] Dean Michael B. Ancajas, Anastacia P. Ballesil, John Richard E. Hizon, Eugene A. Opelinia, Joy Alindia P. Reyes, Allan Gordon L. Sepillo, Winston A. Sumalia, Wilson M. Tan, Conference of IEEE, Taipe, Taiwan, “Dual Core Capability of a 32-bit DLX Microprocessor”, January 2008.

[7] A.M Raid, W.M. Khedr, M.A. El-dosuky, Wesam Ahmed, “JPEG Image Compression Using Discrete Cosine”, International Journal of Computer Science and Engineering Survey (IJCSES) Vol.5, No.2, April 2014.

[8] S.T. Klein, Y. Wiseman, “Paralell Huffman Decoding with Applications to JPEG Files”, The Computer Journal, Vol. 46, Issue. 5, February 2003.

[9] Yi Chen, Guo Chun Wan, Ling Yi Tang, and Mei Song Tong, “Huffman Coding Method Based on Parallel Implementation of FPGA”, IEEE Conference, Singapore, Singapore, November 2017.