International Journal of Computer Applications (0975 – 8887)
Volume 45– No.11, May 2012

Platform Independent JPEG Encoder

Dnyaneshwar D. Ahire
PG Student
Department E & TC
JNEC, Aurangabad, India

Vijayshree A. More
Asst. Professor
Department E & TC
JNEC, Aurangabad, India

ABSTRACT

Image compression using hardware approach is emerging area for researchers. Paper focuses on a new technique using digital camera using the Evaluator-9t development board (ARM9TDMI) and the OVT9650 digital imaging sensor for compression of Bit Map Images. Uncompressed images taken by the image sensor and then compressed using JPEG compression using ARM processor. Compressed images then transferred to Desktop computer pertaining 32-bit Windows operating system. Compressed image follows all the steps involved in JPEG compression technique. We have used various techniques like Digital imaging, Image compression technologies, embedded application of JPEG compression, ARM microcontrollers Serial communication. Most important factor in image processing is to compress image which helps a lot in world of digital consumer application like handheld mobile, digital camera to save memory which is the most expensive thing in this era. So we need a new approach to do this and that is most recent and fastest way to compress the image using JPEG compression standard. This design can be used for digital photography applications to achieve low computation, low storage. It is a competitive solution for scanner to have compression, function embedded for scanner to solve the bandwidth problem between PC and scanner.

Keywords

JPEG, ARM, RISC, DCT, RLE

1. INTRODUCTION

Implementation of real time embedded system applications are really a challenge for us. Growing need ever increases complexity of embedded system It has many issues such as energy consumption by system, memory requirement, and most important thing for real time implementations is time necessitate executing. JPEG's proposed standard aims to be generic, to support a wide variety of applications for continuous tone images. To meet up the differing needs of many applications, the JPEG standard includes two basic compression methods, each with various modes of operation. A DCT based method is specified for "lossy'' compression, and a predictive method for "lossless'' compression. JPEG compression in software is widely used, but time require for conversion and power boundaries for real-time embedded hardware applications make it unfeasible to use a software solution in a general purpose processor. ARM based JPEG is useful for the applications such as scanners and still camera applications. A block diagram for the JPEG encoder is shown in Figure 1. The input for the application is a raw image from digicam. The implemented system included an ARM9TDMI processor core and an on-chip bus, as well as memory and other necessary peripherals. Although a hardware implementation of the JPEG Encoder was within the project scope, the cost and power constraints indicated running as much functionality as was realistic on the offered processor. ARM which is a commanding RISC processor [13]. AMBA specification defines two buses. One is a 32-bit Advanced High Performance Bus with the facet of high-speed and high bandwidth, the other is a 32-bit low power, low-speed, and un-clocked bus that is referred to as Advanced Peripheral Bus. In this proposed system still image from the source of input to ARM processor which is JPEG encoder gives the compressed image with good quality which requires less space to store compressed image into PC.

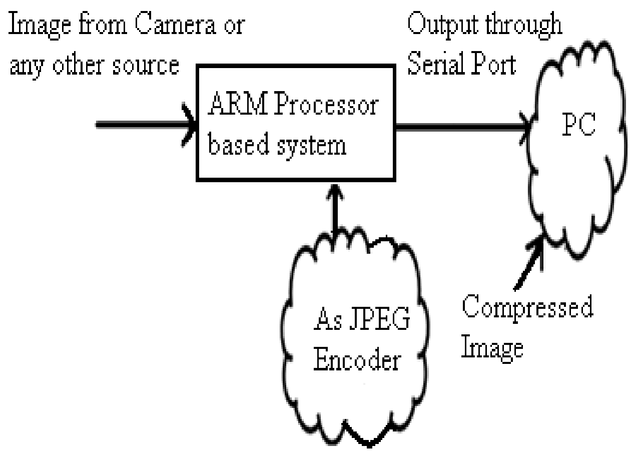
Image from Camera or any other source
Output through Serial Port
ARM Processor based system
PC
As JPEG Encoder
Compressed Image

Figure 1 System Block diagram

The rest of the paper is organized as follows. First it describes some related work, Then it gives details of hardware platform, Next to that describes the design steps needed to achieve a working prototype, Next section present the experimental evaluation of our design. And final section concludes the paper.

1.1 Related Work

Image compression is achieved by the JPEG encoder are based on the fact that some values of each individual blocks are lost because it uses lossy compression only few values can be used to describe each block of an image after DCT transform and quantization processes has been applied. After quantization these block values given to Huffman encoder which provide further compression to reduce the final number of bits that are finally stored. Research work has done on JPEG encoder by targeted different platforms: DSP processors TI 5410, dedicated FPGA implementations [4]. One was a VLIW processor created with Adelante Technologies designer and the other a dedicated hardware implementation created with Celoxica Conventional approaches to ASIP development can be classified into two kinds. First one is a "parameterized generic processor core" such as PEAS-I, satsuki. MetaCore, CASTLE, Xtensa. Most of the systems are dedicated to its job that means it can perform the specific task. The existing system for JPEG encoding was developed using ARM7 platform [8]. Which has limitations of handling image due to limited size of memory? The system which is present in the paper has been developed using ARM9 platform, overcomes the limitations of ARM7 platform mainly of memory.

2. ARM PROCESSOR HARDWARE PLATFORM OF EMBEDDED SYSTEM

Previous work done for the JPEG encoder only performs out-and-out task. The proposed system can be implemented for scanner & digital camera, in this kind of application other task can be achieved by utilizing other processors, Embedded ARM can use to steer clear of that idleness of the previous system. The proposed ARM-based platform not only speeds up system, but also provides the integration facility via AMBA bus for distinctive IP components. This system using ARM is appropriate for number of multimedia applications. Proposed hardware/software design results show that the system verification can be extensively give us a good quality with high compression results. The additional costs for a larger power supply for the high power system can be sizeable. Hardware for proposed system has been designed for ultra-low power applications digital cameras, smart cellular phones and industrial hand held information appliances. The proposed design is built around an ARM processor having 32-bit ARM instruction set for maximum performance and flexibility. It is software programmable but interacts with different pieces of hardware. ARM seems to be foremost the way in this field of processing because the architecture it has adopted.

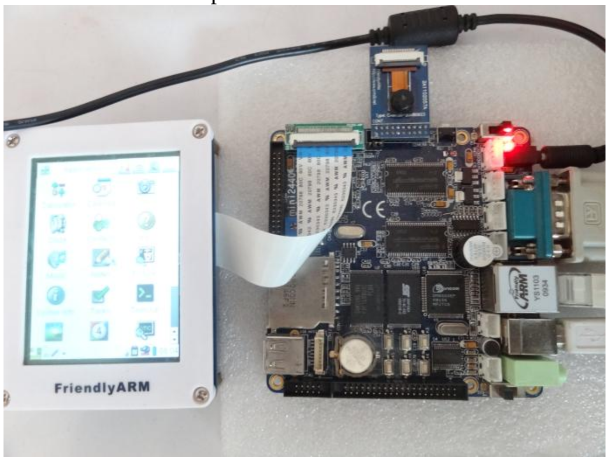

Figure 2 Hardware Interface

2.1 Selection criteria of processor

Following are the features of the ARM9 processor [1]. All in all, the S3C2416 presents a low-cost, highly embedded solution with upgraded features.

a) ARM926EJ CPU 400MHz with 16KB I-cache and 16KB D-cache
b) Dual port external memory controller: DRAM/ROM control and chip select logic.
c) 64KB internal general purpose SRAM
d) 32KB internal ROM for moviNAND booting.
e) LCD controller with DMA-dedicated: 24bpp, 2-PIP
f) 2D hi l t 2D graphics accelerator
g) 6-ch DMAs with external request pins.
h) 4-ch UART (3Mbps) with IrDA 1.0 (64B FIFO)
i) 1-ch HS-SPI (50Mbps)
j) 1 port USB Host v1.1 full speed
k) 1 port USB Device v2.0 high speed
l) 4-ch PWM timers & 1-ch internal timers
m) 2 PLLs with on-chip clock generator
n) Power modes: Normal, Idle, Stop & Deep Stop, Sleep and Power-off
o) 10-ch 12-bit ADC (Touch screen interface)
p) 65nm low-power technology and MtCMOS technology incorporated

3. SOFTWARE PLATFORM FOR JPEG

3.1 Linux on ARM

Complete scalable operating system providing a reliable multi-tasking environment based on an open source model (GPL) Leverage a wide range of UNIX and open source applications early availability on ARM processor-based platforms Used in many ARM technology-based designs including networking and wireless products Broad support through open discussion forums

3.2 U-Boot Module

U-boot(Universal Boot loader) is an open source, multi platform boot loader. U-boot supports a large variety of popular CPUs and CPU families used today. U-boot configures different hardware blocks on the board and brings them out of reset into a same state. It can load and start an OS automatically (auto-boot) or, alternatively, it allows the user to run commands to start the OS. Generally, u-boot resides in the beginning area of the flash. u-boot initialize the CPU and several peripherals located on board, create some critical data structures which will be used by kernel, and load itself into the top of the memory.

3.3 Kernel Module

Kernel is that process which is responsible for co-coordinating the various modules.
Embedded Linux takes the Linux kernel and extract the necessary modules as needed. Within the kernel layer, Linux is composed of five major subsystems:
The Process Scheduler (sched),
The Memory Manager (mm),
The Virtual File System (vfs),
The Network Interface (net),
The Inter-Process Communication (ipc).
Conceptually, the clustering of the components composes the Linux Kernel and each subsystem is an independent component of the Linux kernel.
Controlling the execution of processes
Scheduling processes fairly for execution on the CPU
Allocating main memory for an executing process
Kernel requires gcc (GNU C Compiler) which is readily available

3.4 Deployation of GCC- GNU's Compiler

GCC, the GNU Compiler Collection, is a tool used by nearly every embedded engineer, even those who don't target Linux. GCC is a juggernaut of software engineering that, because of its ubiquity and ease of use, doesn't get the admiration it deserves.

3.5 Integration of Minicom

A communication program which somewhat resembles the shareware program TELIX but is free with source code and runs under most unices. Features include dialing directory with auto-redial, support for UUCP-style lock files on serial devices, a separate script language interpreter, capture to file, multiple users with individual configurations, and more.

4. DESIGN STEPS FOR JPEG ENCODER

Using ARM, here five main units are needed to implement. As shown in Figure 2, DCT, Quantizer, Encoder, Huffman, Output. The zigzag unit found in typical architectures of JPEG encoders is removed The encoder compresses an image as a stream of 8×8 blocks with each element of the block applied and processed individually. For JPEG encoder, the input image is divided into non-overlapping blocks of 8×8 pixels, and the pixel values are converted from unsigned integer format to signed integer format, then 2-D DCT computation is performed on each block [4]. DCT transformation of a natural image from the spatial to the frequency domain results in most of the energy is packed into only a few top-left coefficients. In other words, only few of the original transform components were utilized for reconstruction. Next step for further compression is to apply quantization. Quantizer output puts into zigzag pattern, which uses extensively by RLE and Huffman encoding results more compression in the image.

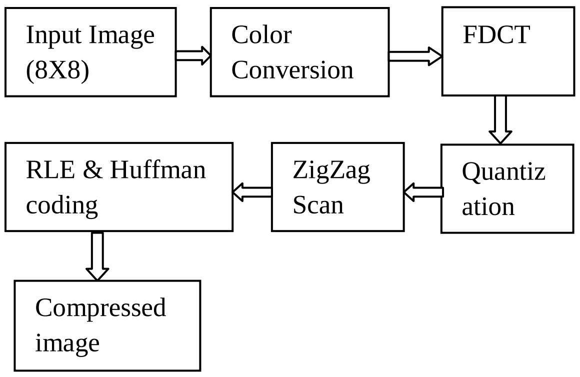

Figure 3 Steps for JPEG encoder

4.1 Succession of Discrete Cosine Transform (DCT)

Important step in JPEG encoder is DCT, DCT-based architecture has gained popularity in image and video processing. There are three basic types for 2-D DCT computation. The first implementation is with the aid of other transforms, such as the Discrete Hartley Transform (DHT) and the Discrete Fourier Transform (DFT). The second approach is a direct mapping of the algorithm by using the results of a polynomial transform. And the third style computes 2-D DCT by row-column decomposition [6]. In general, the direct method is considered to produce algorithms with lower number of multiplications than the row column approach. Reduces complexity in computation has gained popularity of the row column approach.

$$C(u,v) = \alpha(\mathrm{u})\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} \mathrm{f(x,y)}\cos\left[\frac{\pi(2\mathrm{x}+1)\mathrm{u}}{2\mathrm{N}}\right]\cos\left[\frac{\pi(2\mathrm{y}+1)\mathrm{v}}{2\mathrm{N}}\right]$$

For u=0,1,2,N-1.

As multiplier costs more hardware than adder, multiplication-free algorithms are very attractive. It performs a 2-dimensional DCT using clearly, 64 weights, w(y, x, j, i), are needed to process each input pixel(y, x).

4.2 Quantization

$$Q(u,v) \;=\; floor\left[\,\left(\frac{F(u,v)}{q(u,v)}\right) + \;0.5\right]$$

Where $Q(u,v)$ is the quantized value

$F(u,v)$ is the DCT coefficient

$q(u,v)$ is the corresponding quantization value.

To round to the nearest integer, 0.5 is added, and the floor of that value is taken.

The quantizer unit rejects information that is not visually significant. It is the principle source of information loss in DCT-based encoders. In the proposed implementation, after the last 8*8 block element finishes updating the first output DCT coefficient, the quantizer unit starts working thus saving few clock cycles by pipelining operation with the DCT unit. Quantization is defined as division of each DCT coefficient by the corresponding quantization value, followed by rounding to the nearest integer. Division operations are replaced with multiplication and shift operations. Quantization table is provided in the implementation of JPEG encoder.

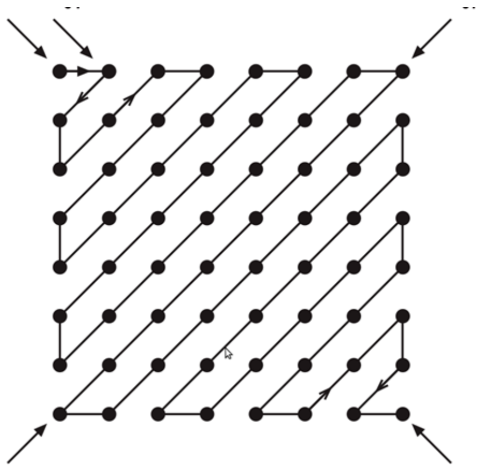

Figure 4 Zigzag order of 8 X 8 blocks

In a typical implementation of a JPEG encoder, the output DCT coefficients are quantized and reordered in the zigzag sequence of Figure 4. The main benefit of zigzag ordering is to group as many of the quantized zero-value coefficients so as to produce the longest runs of zero values.

4.3 Encoder and Huffman

These units achieve additional compression by encoding the quantized DCT coefficients more compactly based on their statistical characteristics [2]. Here, they are implemented as in typical JPEG encoders. The quantized DC coefficient is encoded based on its category as the difference from that of the previous block in the encoding order. The quantized AC coefficients are encoded as (run, cat), where run is the number of zeros preceding a nonzero value and cat is the category of this value. Three different encoding techniques are used in this implementation: run-length encoding, entropy encoding, and “end of block” tagging.

4.4 Results

This unit handles two constraints set by the JPEG standard. The first constraint is that the maximum run is 15. The second is that when there are zeros till the end of the block, a Huffman code corresponding to run=0 and cat=0 has to be sent. These constraints are handled by introducing four delay stages that force the output to be delayed so as to detect any sequence (run, cat) of (15, 0) and (0, 0). The delay stages are determined to be four because for an 8×8 block there may be a maximum of three (15, 0) which may be followed by (0, 0). A data ready signal is necessary for each valid output. Finally compressed image come out from all these process.

5. EXPERIMENTAL EVALUTION

Figure 5 Sample BMP Image no.1

Figure 6 Sample BMP Image No. 2.

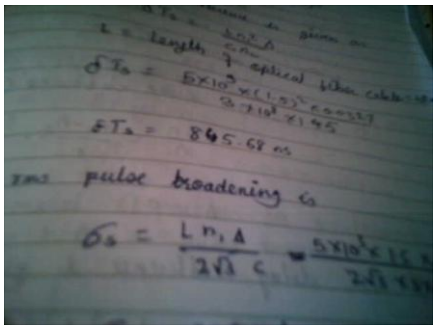

Figure 7 Sample BMP Image No. 3.

Figure 8 Sample BMP Image No. 4.

The sample BMP images, size of images are about 960KB. This image has some color coefficients, first converting this color coefficient into gray scale. Here image convert into gray on which basically system works, gray or color still images which either from camera or from scanner can apply to the system.

These coefficients are directly applied to ARM through serial bus. Taking image coefficients to process for DCT, This DCT code is written in 'Embedded C' code all the last values are get divided by 65535 because when we are converting sample image into gray we are using index 16. After encoding, encoded data is as like below Compressed image is about 26.5KBs and also mention in results about other sample images percentage compression using ARM and using Windows software.

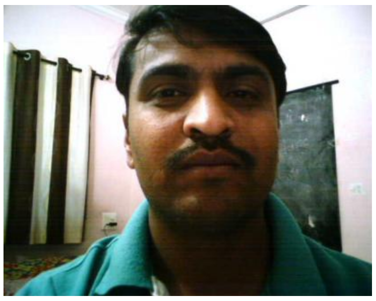

Figure 9 Compressed Image by ARM of sample image 1 about 26.5KB

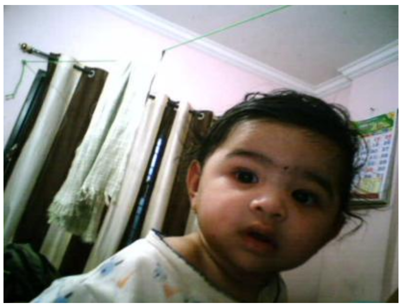

Figure 10 Compressed Image by ARM of sample image 2 about 26.4KB

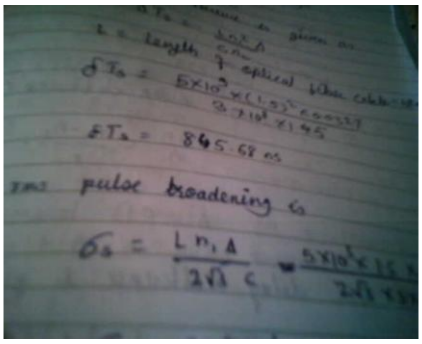

Figure 11 Compressed Image by ARM of sample image 3 about 15.3KB

Figure 12 Compressed Image by ARM of sample image 4 about 10.8KB

Figure 13 Compressed Image by Windows of sample image 1 about 32.6KB

Figure 14 Compressed Image by Windows of sample image 2 about 31.9KB

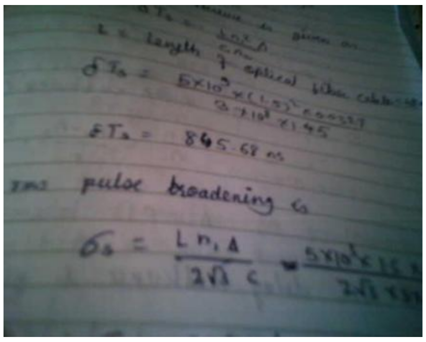

Figure 15 Compressed Image by Windows of sample image 3 about 18.3KB

Figure 16 Compressed Image by Windows of sample image 4 about 11.7KB

Table 1 Results of compressed image

Sample	**Size in BMP (KB)**	**Size IN JPEG by ARM (KB)**	**% Compression**	**Size in JPEG by (XP) (KB)**	**% Compression**
Sample No.1	960	26.5	97.24%	32.6	96.61%
Sample No.2	960	26.4	97.25%	31.9	96.68%
Sample No.3	960	15.3	98.41%	18.3	98.1%
Sample No.4	960	10.8	98.87%	11.7	98.79%

Results shows in table for four sample images are put side by side with existing technology for JPEG Encoder i.e. by using Windows software. It also gives niceties about entitlement compression. Graphical depiction shows JPEG compression using ARM9 and using Software for above shown images

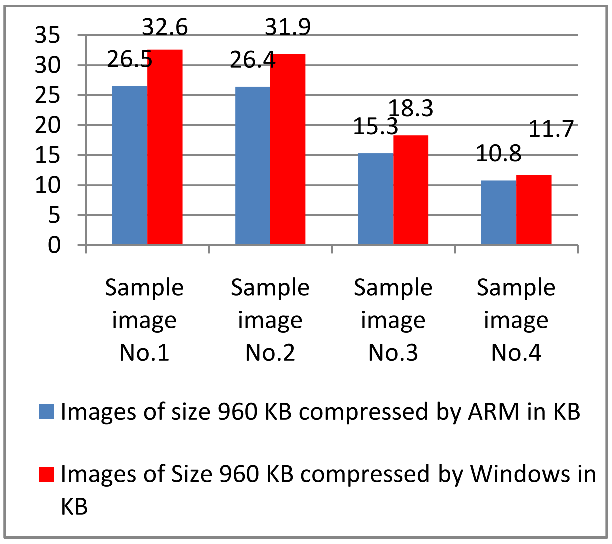

Figure No.17 JPEG Compression comparison

5. CONCLUSION

We draw a fourfold conclusion from this design experiment suitable for applications as scanners and still cameras. IP-based JPEG encoder integration with a memory efficient preprocessing architecture. These JPEG encoder IP features that the quantization, and the whole design is highly modularized, fully pipelined, and with friendly interface, which makes it easier for system integration. The proposed embedded JPEG encoder IP core with this memory efficient preprocessing circuit is a low cost and competitive solution for different multimedia application. The hardware provides 4 Synchronous and 2 Asynch Interfaces. More ever there are GPIO ports. External devices like camera, digital radios etc can be interfaces with the designed hardware through these interfaces. Biasing voltages of 3.3V, 50uA and 5V, 30uA have been ‘given to meet the power requirements of hardware. Because of low power consumption, the designed hardware ARM9 is well suited for applications. Implemented experimentation on JPEG lossy technique on ARM platform with 90 to 98% image compression shows the design success.

7. REFERANCES

[1] Chung-Jr Lian, Liang-Gee Chen, Hao-Chieh Chang, and Yung-Chi “Embedded JPEG Encoder IP Core and Memory Efficient Preprocessing Architecture for Scanner” Chang 0-7803-6253-5/00/$10.00 ©2000 IEEE.

[2] A. Staller, P. Dillinger, and R. Manner, “Implementation of the JPEG2000” Standard on a Virtex 1000 FPGA. Springer Berlin/Heidelberg Publishers, 2002.

[3] K. Sakiyama, P. R. Schaumont, and I. M. Verbauwhede “Finding the best system design flow for a high-speed JPEG encoder”, Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 577-578, 2003.

[4] Shinsuke Kobayushi, Kenturo Mita Graduate School of Engineering Science, “Rapid prototyping of jpeg encoder using the asip development systempeas-111” Osaka University 0-7803-7663- 3/03/$I7.00 ©2003 IEEE

[5] Bhatti, J.I.; Butt, U.M.; Hussain, K.; Raja, G. “ARM solutions for text, audio and image data processing for ultra low power applications”;Multitopic Conference, 2004. Proceedings of INMIC 2004. 8th International 24-26 Dec. 2004 Page(s):32 – 35”

[6] "SPIE Proceedings,E. Farzad, C. Matthieu, and W. Stefan, "JPEG versus JPEG 2000: anobjective comparison of image encoding quality” vol. 5558, pp. 300-308, 2004.

[7] T. Acharya and Ping-Sing Tsai,John Wiley & Sons press, “JPEG2000 Standard for Image Compression Concepts”, Algorithms and VLSI Architectures” 2005.

[8] Davare, A.; Qi Zhu; Moondanos, J.;Sangiovanni-Vincentelli, “A JPEG encoding on the Intel MXP5800: a platform based design case study Embedded Systems for Real-Time Multimedia”, 2005. 3rd workshop on 22-23 Sept. 2005

[9] Osman,;Mahjoup,W.;Nabih,A.;Aly,Digital ObjectIdentifies “*JPEG encoder for low cos tFPG As”* G.M.; Computer engineering & Systems, 2007. ICCES'07. International Conference on 27-29Nov.2007

[10] Tumeo A.; Monchiero M.; Palermo G.; Ferrandi F.;Sciuto D. “A Self-Reconfigurable Implementation of the JPEG Encoder” Application-specific systems, architectures and Processors, 2007. ASAP.IEEE International Conf.on 9-11 July 2007Page(s):24 - 29

[11] Yun-Lung Lee, Jun-Wei Yang, and Jer Min Jou Design of a Distributed “JPEG Encoder on a Scalable NoC Platform” Department of Electrical Engineering, National Cheng Kung University No.1,University Road, Tainan, Taiwan, R.O.C. 978-1-4244-1617-2/08/S25.00 © 2008 IEEE

[12] N.N. Ganvir, A.D. Jadhav SCOE, Pune “Explore the Performance of the ARM Processor Using JPEG” International Journal on Computer Science and Engineering, Vol. 2(1), 2010, 12-17