RC-008

Rapid Design of a Multiprocessor System for a JPEG Decoder on FPGA

Cao Dawei† Chen Keyan† Watanabe Takahiro†

Abstract

Multiple cores system at a lower frequency is efficient approach to increase performance where power consumption is limiting performance. Meanwhile FPGA is capable of providing designers with several benefits in system design, which decreasing the design cost and time for market. This paper demonstrates the design of an FPGA-based embedded multiprocessor system integrating up to three processor cores into a system for a JPEG Decoder. According to the process of JPEG decoding, task sharing is proposed to make the multi-core work in parallel without conflicts. The application program to decode a DCT-based JPEG compressed image is successfully implemented on the FPGA board with the 40MHz operating frequency. Based on the proposed hardware and software, the performance increasing of decoding is approximately close to 2.6 times in a large resolution image and up to 10% power saving without performance loss compared to the uni-core system.

1. Introduction

Recently, with the development of semiconductor processing technology, the increasing number of devices could be integrated into one chip. One chip with diverse function modules can be fulfilled as SoC (System on Chip).The complexity of designing and programming a SoC system is growing rapidly and high improvement of performance is expected intensively. But we're in a phase where power consumption is limiting performance. Power efficiency and performance/watt are now critical metrics along with absolute performance. One of approaches to increase performance efficiency is by adding multiple cores and running them at a lower frequency[1].

However, the traditional SoC developing method naturally takes long period and high cost. Continued increase in the nonrecurring expenses for the manufacturing and design of SoC, in the face of continued time-to-market pressures, is leading to the need for significant changes to the design and manufacture[2]. An alternative solution is FPGA-based system. In this way, we can obtain both high programmability and low risks. Soft-core processors implemented in FPGAs can be easily customized to the needs of a specific target application.

As mentioned above, combination of adding multiple cores and FPGA-based system is FPGA-based multiprocessor system. Our research is trying to figure out how to develop an MPSoC on FPGA efficiently, find the solution on developing an MPSoC embedded system, and optimize the system performance and power consumption. In this paper, JPEG decoder is designed as an example to realize a multi-core system and implemented on FPGA board with an optimal experiment result.

This paper is organized as follows. Section 2 describes the hardware design of our MPSoC platform for DCT-based JPEG decoder, and the software design is introduced in section 3. The experimented results of performance described by the execution time of decoding and the power dissipation are shown in section 4. Finally, conclusions and future work are presented.

2. Hardware design

We use Altera FPGA as a platform including tools, hardware, and software as well as IP cores for developing embedded multiprocessor systems since the hardware can be easily modified and tuned using the SOPC Builder tool to provide optimal system performance[3].

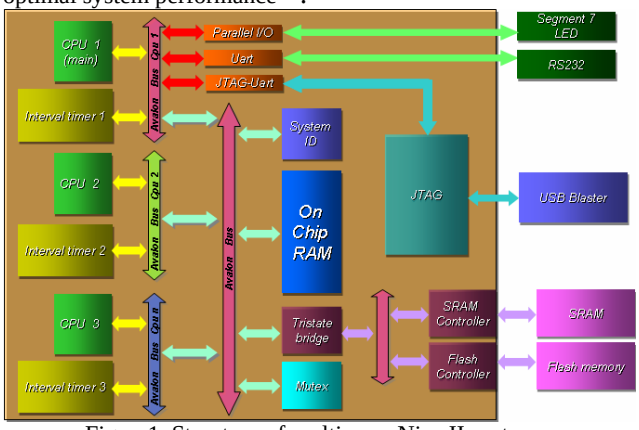

Figure1. Structure of multi-core Nios II system

The figure1 is our designed multi-core Nios II system. In this system, the Nios II soft CPU cores are used for processing the task separations. All Nios II cores share the same physical memory, which is composed of an on-chip RAM, an out-chip SRAM and a flash ROM, for data storage and instruction. But the program for each processor is located in its own region of memory. We use a region of the share-RAM to pass messages between CPU cores; it works for data communication of task sharing. The Mutex works as an arbitrator; it will set the priority under the situation when more than one master port in the system attempts to access the shared resource simultaneously[4]. This is an effective way to protect data from corruption. We use Avalon bus to connect the Nios II CPU cores to the peripherals, to provide the connection specification and communication timing relations between master and slave port. JTAG is an interface to download MPSoC design into the FPGA board, and also used to debug application software on hardware design. UART provides the process data transmission between FPGA and computer. The SRAM and flash memory supply complementary memory for on-chip RAM. However, out-chip memory does not compatible with the Avalon bus, the SRAM controller, flash memory interface and Tristate bridge are adopted to convert the data communication between Avalon bus and out-chip memory.

3. Software design

The design challenge in developing a multiprocessor system now lies in the software design for making the processors operate efficiently together and not conflict with one another. We developed JPEG decoder application programs for performance evaluation.

The JPEG decoder program in this paper is used to decode a JPEG File Interchange Format (JFIF) file into an uncompressed bitmap file. As shown in figure2, we separate the decode processing into following steps: first is to load the JFIF file and setup the quantization table and Huffman table which will be used in dequantization processing and entropy decoding respectively. According to the size of the image, the program will apply corresponding memory to store the decoding data. Then the key processing step starts, which is executed based on 8×8 blocks. This process includes: entropy decoding, inverse

†Graduation School of IPS, Waseda University

zig-zag, dequantization and IDCT[5]. After decoding all the blocks, the program converts the YUV (Y stands for the brightness and UV stands for chromatic aberration) into RGB (RGB stands for red, green and blue).

The significant work for software design is task sharing which make the multi-processors work in parallel without confliction to decrease the time consumption of decoding. According to the structure of JPEG decode, the whole task is separated as shown in figure3.

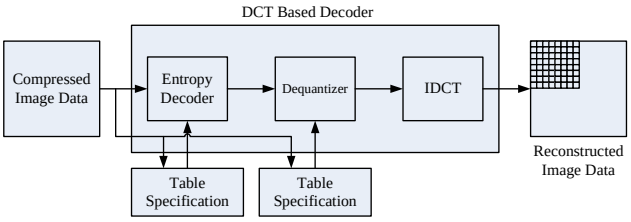

Figure2. Structure of JPEG decode

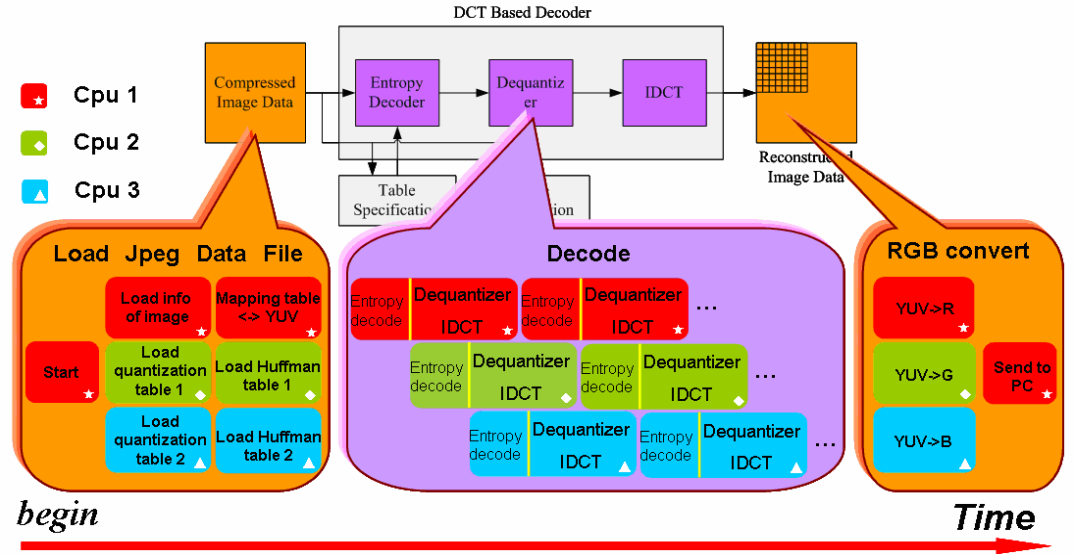

Figure3. Task sharing of ternate-core system

Red (pentacle), green (diamond) and blue (triangle) represents the CPU1, CPU2 and CPU3 working respectively in the time domain. CPU1 conducts all the processes of decoding. The beginning of task separation is separating the data file loading task. At this step, CPU1 loads the basic information of the image, allocates memory and maps the relationship between quantization table ID, Huffman table ID and YUV component. Generally speaking, the compressed image is colorful one which has two quantization tables and two Huffman tables for Y component and UV component respectively. The second step is Decode. This step is based on 8×8 blocks, each of which is decoded one by one. Because of the variation of entropy codes, we cannot predict the beginning of the second block before accomplishing the entropy decoding of the first block. So the CPU2 does not work until CPU1 finishes the first block. This problem also occurs on CPU3. Therefore, after decoding the first and second blocks, CPU3 begins to work. The third step is to convert YUV component into RGB to reconstruct the bmp file. In this step there are two proposals for dual-core and ternate-core system. In the dual-core system, the CPU1 will convert the YUV to RGB from the beginning of the image and the CPU2 will work from the end of the image, this method will make these two cups work in parallel. In the ternate-core system, these three cups convert the YUV into R, G and B component respectively. The final job is that CPU1 which has the communication port with PC sends the reconstructed data to the PC to validate the decoding.

4. Results

In this section, experimented results of performance described by the execution time of decoding and the power dissipation estimated by Altera PowerPlay Power Analyzer Tool are shown.

We implement the MPSoC system on a FPGA development board on which there is a FPGA EP1S25F1020C7[6] of Altera Stratix family. There are three types of Nios II CPU core: Nios II/F (Fast), Nios II/S (Standard), and Nios II/E (Economy). Standard and fast types are chosen to construct the CPU of the system. Table1 shows the FPGA utilization of multi-processor hardware systems. “type” means the feature of Nios II cores which compose the system.

Table1. FPGA utilization statistics

Type of system	LEs	RAM bits	DSP blocks
Uni-core Nios II (type S)	2571	1094912	8
Dual-core Nios II (type S, S)	4899	1140224	16
Ternate-core Nios II (type S, S, S)	7177	1185536	24
Uni-core Nios II (type F)	3294	1112896	8
Dual-core Nios II (type F, F)	6345	1176192	16
Ternate-core Nios II (type F, F, F)	9362	1239488	24

The application program that decodes a DCT-based compressed image is implemented on the FPGA board with the 40MHz operating frequency. Table2 shows the execution time through the numbers of clock cycle and figure4 shows the normalized execution time with uni-core system measured under different resolution image and different structure of the hardware system. Based on the proposed task separations, the

performance of decoding is approximately close to 2.6 times in a large resolution image. The performance increasing along with increment of image resolution is that the impact of the time interval between start of the first block decoding and start of the second or third block determined by dual-core or ternate-core is becoming less and less.

However, it is worth to mention that the performance of multiprocessor will not increase linearly with accordance of the number of processors, because collision exists in memory access between processors. Based on the proposed system architecture, overhead of shared memory access turns more with increasing number of processors.

Generally speaking, the size of image is larger than 256×192, so we can get at least 1.6 times performance increasing with dual-core system and 2.5 times with ternate-core system. Under this situation, if we want only the performance of uni-core system, we can decrease the operating frequency. The following table3 shows the estimation of power dissipation. Set the uni-core system running at 40 MHz as the benchmark, we can operate the dual-core system at 25MHz and the ternate-core system at 16 MHz without performance loss. Through the three processors working in parallel, we can obtain about 10% power saving compared with Uni-core system at 40MHz. Because the decoding task is shared evenly among the Nios II processors, the reduction of power dissipation takes effect for each processor correspondingly with same ratio.

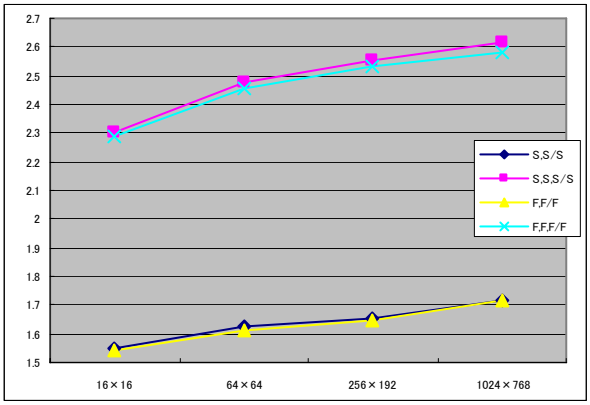

Figure4. Normalized execution time with uni-core system

Table2. Execution time of JPEG decoding under different resolution

Resolution of image	Uni-core Nios II (type S)	Dual-core Nios II (type S, S)	Ternate-core Nios II (type S, S, S)	Uni-core Nios II (type F)	Dual-core Nios II (type F, F)	Ternate-core Nios II (type F, F, F)
16×16	598,690	387,397	260,098	434,929	282,678	189,785
64×64	8,770,620	5,391,780	3,539,580	6,365,267	3,950,993	2,590,058
256×192	104,654,640	63,355,416	40,949,380	75,947,733	46,194,176	29,962,784
1024×768	1,673,664,135	976,857,031	639,953,796	1,214,572,821	708,906,816	469,905,376

Unit: clocks/40MHz

Table3. Power dissipation

Operating frequency	Uni-core Nios II (type S)	Dual-core Nios II (type S, S)	Ternate-core Nios II (type S, S, S)	Uni-core Nios II (type F)	Dual-core Nios II (type F, F)	Ternate-core Nios II (type F, F, F)
16 MHz			**585.15 mW**			**594.37 mW**
25 MHz		**617.23 mW**	660.91 mW		**627.44 mW**	675.01 mW
40 MHz	**646.82 mW**	717.17 mW	787.19 mW	**654.69 mW**	733.22 mW	809.35 mW

Estimated by Altera PowerPlay Power Analyzer Tool

5. Conclusions

An FPGA-based embedded multiprocessor system integrating up to three Nios II soft cores into a system for a JPEG Decoder is proposed and implemented. In order to make the multi-core work in parallel without conflicts, the task separations are proposed according to the process of JPEG decoding. Based on these methods, the performance increasing of decoding is approximately close to 2.6 times in a large resolution image and up to 10% power saving without performance loss compared to uni-core system.

Some further improvements will be made out in the future. For example, using a parallel decoder of entropy codes[7] to overcome the time interval between start of the first block decoding and start of the second or third block to achieve a triple performance increasing no matter it is a small size of image or large.

6. Acknowledgement

This work was partly supported by a grant of Knowledge Cluster Initiative 2nd stage implemented by Ministry of Education, Culture, Sports, Science and Technology (MEXT).

Reference

[1] Jerraya, A.A., Franza, O., Levy, M., Nakaya, M., Paulin, P., Ramacher, U., Talla, D., Wolf, W. Roundtable: Envisioning the Future for Multiprocessor SoC. Design & Test of Computers, IEEE. Volume 24, Issue 2, Page(s): 174 – 183, March-April 2007

[2] Paulin, P.G., Pilkington, C., Langevin, M., Bensoudane, E. Parallel programming models for a multiprocessor SoC platform applied to networking and multimedia. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. Volume 14, Issue 7, Page(s):667 – 680, July 2006

[3] http://www.altera.com

[4] Altera Inc. Creating Multiprocessor Nios II Systems Tutorial. v1.3. December 2007

[5] Wallace, G.K., The JPEG still picture compression standard, Consumer Electronics, IEEE Transactions on volume 38, Issue 1. Page(s): xviii – xxxiv, Feb. 1992

[6] Altera Inc. Stratix Device Handbook, Volume 1. January 2006

[7] Klein, S.T., Wiseman, Y. Parallel Huffman decoding. Data Compression Conference, 2000. Page(s): 383 – 392, March 2000