TIGHTLY-COUPLED MPEG-4 VIDEO ENCODER FRAMEWORK ON ASSYMETRIC DUAL-CORE PLATFORMS

Cheng-Nan Chiu, Chien-Tang Tseng, and Chun-Jen Tsai

Dept. of Computer Science and Information Engineering
National Chiao Tung University

Abstract—**This paper performs detail analysis of a tightly-coupled scheme for multimedia applications on asymmetric multi-processor dual-core platforms. Existing partitioning methodologies for asymmetric dual-core systems typically perform software partitioning offline (during compile time), and assign tasks to either the RISC core or the DSP core. In this paper, we study a tightly-coupled approach where the same computation task will be assigned simultaneously in a tight cooperation manner. An MPEG-4 Simple Profile video encoder has been implemented on a TI-OMAP 1510 platform for the analysis. Initial experiments show that using tightly-coupled approach could increase the overall performance of a complicated multimedia system. The paper also illustrates the weakness of current dual-core architecture design for tightly-coupled approach and provides information for future improvement.**

I. INTRODUCTION

Due to heavy computational requirements of multimedia communication tasks, many handsets start to incorporate an application processor into digital baseband architecture [1]. Quite often, these application processors apply an asymmetric dual-core (a RISC core plus a DSP core) design to increase the performance of multimedia processing. So far, task partitions between the RISC and the DSP have been done in a mutual exclusive manner. That is, a task is either assigned to the RISC or the DSP, but never both. This methodology makes sense in the past since the function of a RISC core is very different from that of a DSP. However, new generations of RISC processors are usually powerful enough to take over some of the computationally expensive jobs.

In this paper, we use MPEG-4 Simple Profile encoder [2] as an example to investigate tightly-coupled software partitioning on a dual-core platform. By tightly-coupled partitioning, we mean that a task, e.g., motion estimation, is assigned simultaneously to both the MCU and the DSP. It is shown in this paper that this methodology achieves higher performance than assigning the whole motion estimation task to the DSP alone. In the implemented system, dynamic partitioning of the ratio of the task between the RISC core and the DSP core is done via an interrupt driven control module, which can be integrated into a process scheduler of an embedded OS kernel for tightly-coupled dual-core scheduling.

Even though the communication overhead between the two cores is very high under existing dual-core architecture designs, one can still infer from the experimental results that with the proposed approach the performance of a multimedia processing task on a dual-core platform will be much higher than that on a loosely-coupled dual-core platform. As embedded multimedia systems become more and more sophisticated, this approach can be very promising.

The paper is organized as follows. The scheme of tightly-coupled dual-core system is presented in section II. An implementation of MPEG-4 Simple Profile video encoder is described in section III. Experimental results and analysis are conducted in section IV. Finally, some discussions are given in section V.

II. DYNAMIC KERNEL SCHEDULER MODEL

A typical loosely-coupled dual-core framework is shown in Fig. 1. Quite often, when the DSP core is processing data, the RISC core is in idle status. As the performance and capability of the embedded RISC core become more powerful and the applications become more sophisticated, it is reasonable to investigate the possibility of a tightly-coupled system as shown in Fig. 2.

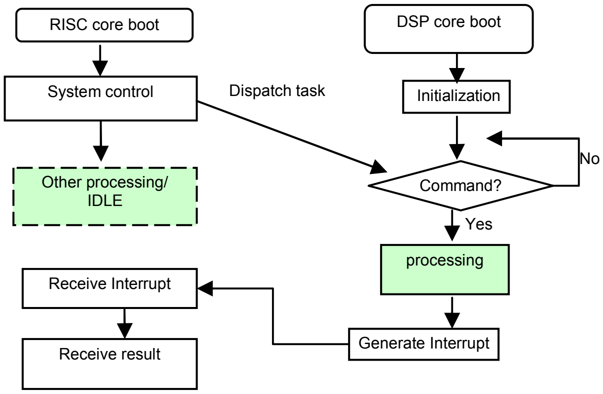

Fig. 1. A loosely-coupled dual-core

0-7803-8834-8/05/$20.00 ©2005 IEEE.

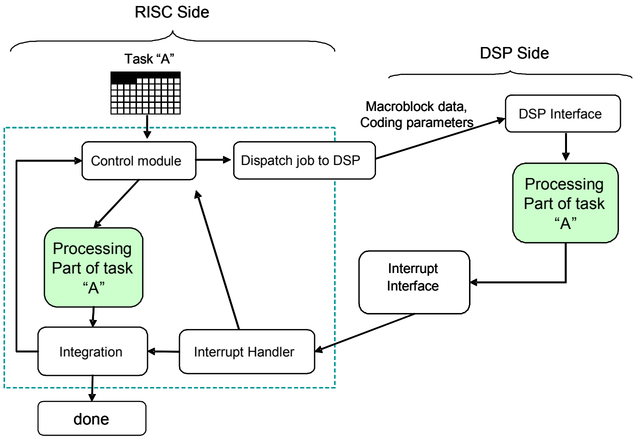

Fig. 2. A tightly-coupled dual-core system

In Fig. 2, portions of the same task "A" will be executed by both the RISC core and the DSP core simultaneously. Even though the computational performance will certainly be increase, the communication overhead could be too high that override the reduction in computation time. However, under deeper investigation, this is not always true. First, in a sophisticated embedded multimedia systems, the data streams usually arrives from the RISC side (running OS), if we only feed portion of the incoming data to the DSP core for processing, the inter-communication cost is lower. Secondly, based on our experiments, a lot of communication overhead actually comes from data format conversion, not from bus bandwidth limitation. That is, the processing module on the RISC side requires different data arrangement than that on the DSP side. With smart DMA design (which does not exist on our experimental system), this overhead can be greatly reduced. Third, a multi-interconnect distributed memory subsystem can greatly improve the communication performance as well.

However, one key issue here is that most development system only generates the target application tasks in either the RISC instruction binary format or the DSP instruction binary format, but never both. This makes the design of a dynamic scheduling tightly-coupled system very difficult. There is a recent trend in the industry that an application does not issue system calls to reach directly the hardware service. Instead, a middleware is inserted between the application layer, and the OS layer to provide services for various tasks. This new model of system design facilitates porting of applications to different hardware platform. It is suggested that this model can be applied for embedded OS kernel to perform dynamic tightly-coupled scheduling on asymmetric multiple-core platforms. If a particular computation function units has both MCU and DSP versions of the service registered (e.g. binary codes are available for both cores), the kernel will dynamically schedule the tasks to either the RISC or the DSP, depending on the runtime computational load of both cores.

III. TIGHTLY-COUPLED DUAL-CORE MPEG-4 VIDEO ENCODER

We have implemented an MPEG-4 SP video encoder based on the framework described in section II. This section gives a brief description of the system. The target platform is an TI OMAP 1510. The RISC and DSP interface is shown in Fig. 3. Commands are passed between RISC and DSP using mailbox devices while multimedia data are passed between two cores using memory devices. Both SRAM and SDRAM are tested, but the SRAM results are illustrated in this section.

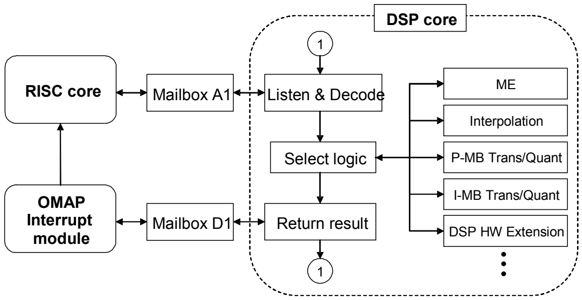

Fig. 3. A tightly-coupled dual-core system

A. Intra frame encoding

In order to reduce memory bandwidth overhead per macroblock for encoding I-frames, the four coding processes, namely FDCT, quantization, de-quantization and IDCT, for a macroblock is combined as a subtask. Therefore, a single transfer of input data to the DSP will produces the coding results, including quantized coefficients and reconstructed block Fig. 4. The architecture and computation time percentage among modules are illustrated in Fig. 5. The computation time is measured by a hardware timer on the target platform.

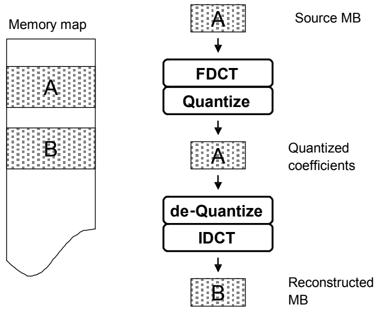

Fig. 4. Memory usage for I-MB encoding

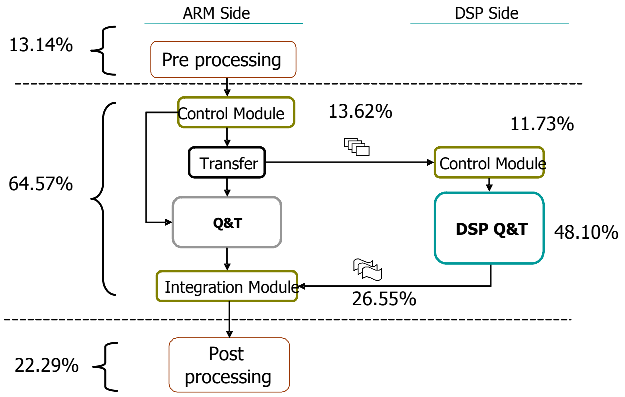

Fig. 5. Intra computation time distribution

B. Inter frame encoding

The most computationally expensive tasks of inter frame encoding are interpolation and motion estimation. In this section, we give a brief discussion on how these modules are done in the proposed tightly-coupled dual-core system.

B.1 Interpolation for MC and ME

In MPEG-4 Simple Profile, motion compensation (MC) and motion estimation (ME) are done with half-pixel accuracy. Therefore, sub-pixel interpolation of the original reference frame pixels for both MC and ME is necessary. The architecture and computation time percentage among modules are illustrated in Fig. 6.

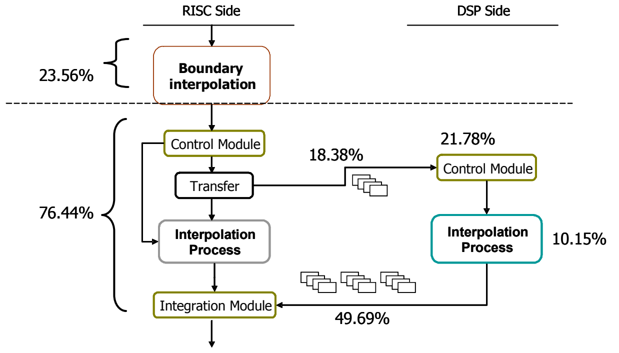

Fig. 6. Interpolation computation time distribution

B.2 Motion Estimation

In the proposed framework, ME is divided into two levels. The first one is ME search in macroblock level, and the second one is search in block level. Since the Sum-of-Absolute-Difference (SAD) output of the ME process is typically used for MB encoding mode decision, a mode decision module is included in the ME control flow. Fig. 7 shows the framework and computation time distribution for ME.

The ME algorithm used in current implementation is the four-step hierarchical search algorithm. The search range is selected to be from -16 to 15 so that the search window size is 48x48 pixels. The choice of this algorithm is because that this is the main algorithm supported by the TI-C55x image/video hardware accelerator library [3]. Both MB-level ME search and block-level ME search are achieved using the same search algorithm.

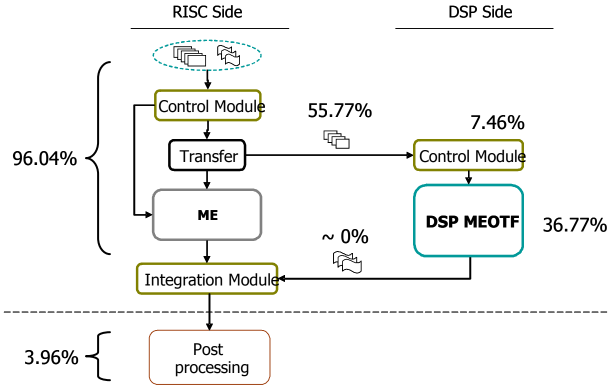

Fig. 7. ME computation time distribution

If the ME task of a macroblock is dispatched to the DSP, the DSP needs the interpolated reference pixels as well as the original reference pixels in order to perform half-pixel motion vector search. There are two different ways to do this. The first one is to save the interpolated pixels from previous steps (on the ARM side) and transfer the data to the DSP side for sub-pixel ME. Another method is to transfer only the original reference frame pixels to the DSP and let it compute the interpolated pixels on the fly. These two methods are illustrated in Fig. 8.

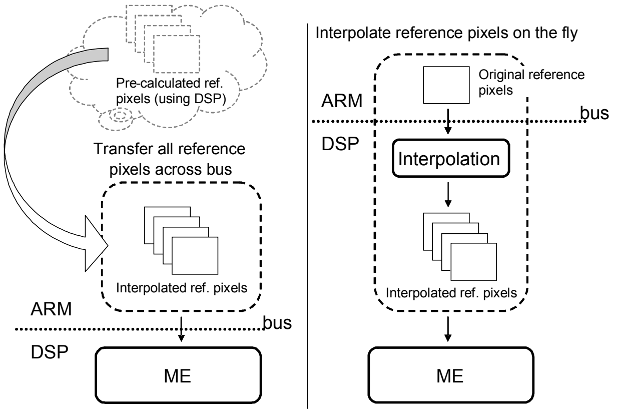
Pre-calculated ref. pixels (using DSP)
Transfer all reference pixels across bus
Interpolated ref. pixels
ARM
bus
DSP
ME
Interpolate reference pixels on the fly
Original reference pixels
ARM
bus
DSP
Interpolation
Interpolated ref. pixels
ME

Fig. 8. On-the-fly interpolation for ME

The first approach consumes more memory bandwidth as a tradeoff for less computation while the second approach requires much less memory bandwidth at the expense of higher computational cost. It can be seen from the experiments that the second approach achieves better performance. This is quite common for multimedia applications that bus bandwidth is often the bottleneck of the system performance.

B.3 Usage of DMA for ARM/DSP Data Transfer

Since the data transfer between ARM and DSP is a very demanding job for multimedia applications such as the MPEG-4 video codec implemented in this paper, it is important to let the DMA logic to handle this task. In the proposed framework, exchange of large amount of data between the two cores is done via DMA.

IV. EXPERIMENTS

In this section, some experiments are conducted to demonstrate the performance of the co-design architecture. The QCIF version of the STEFAN sequence is used for the experiments. Only the first 150 frames are used in the tests, and the target bitrate is 100 kbps. Table 1 shows the test environment. The main program for ARM is stored in SDRAM. On the DSP side, main program are stored in SARAM, and local data are put on the DARAM. Finally, the MPUI mode is set as shared mode for ARM core to access DSP core's memory. The reference frame can be stored in SDRAM or on-chip SRAM. Only the results for SRAM are presented here.

Setup of the experiments	
ARM core	150 MHz
DSP core	150 MHz
Traffic controller	75MHz
System DMA	No burst, 16-bit width

Table 1. Setup of the experiment

In particular, the ratio of the task portion that are dynamically assigned to the RISC core and the DSP core during runtime for Interpolation, ME, MC, Intra T/Q, and Inter T/Q are listed in column two of the table. This ratio roughly matched to the relative speed ratio for each task between the RISC core and the DSP core. The ratio takes into account the factor of data access overhead. Therefore, for some computational tasks, the RISC core runs faster than one might expect (comparing to the DSP core) because the RISC core can access the data faster.

It must be pointed out that currently the performance of the implemented tightly-coupled system is much slower than the published numbers by TI [6]. There are several reasons. First, the data structure used in the implementation is a close derivation from the reference software and is not really suitable for such tightly-coupled system. Secondly, the architecture is not designed for tightly-coupled implementation of multimedia algorithm as discussed in section II.

	Execution time	Task Portion RSIC:DSP	Run time percentage
Initialization	203 ms		0.838 %
Conversion (I/O)	1179 ms		4.857 %
Set edge	274 ms		1.129 %
Interpolation	2616 ms	1 : 1.48	10.779 %
ME	9270 ms	1 : 6.36	38.198 %
MC	2893 ms		11.922 %
Intra T/Q	49 ms	1 : 3.5	0.200 %
Inter T/Q	3771 ms	1 : 3.62	15.537 %
Rate control	1520 ms		6.265 %
AC/DC Prediction	79 ms		0.325 %
VLC	737 ms		3.035 %
Others	1679 ms		6.918 %
Total	24270 ms		100 %
Coding speed (fps)	6.2 fps		
Average Bitrate	100 kbps		
Average QP	29.23		

Table 2. Encoding performance

V. CONCLUSIONS

This paper proposes a tightly-coupled dual-core application framework and demonstrated an initial implementation of MPEG-4 video encoder running on OMAP platform. It has been shown that although the computation time is reduced in a tightly-coupled manner, the overall performance is hindered by the architecture design of existing inter-processor communication link.

VI. ACKNOWLEDGEMENT

This research is partly funded by National Science Council, Taiwan, R.O.C., under grant number NSC 93-2220-E-009-008.

REFERENCES

[1] M. L. McMahan, *Evolving Cellular Handset Architectures but a Continuing, Insatiable Desire for DSP MIPS, TI App. Report SPRA650*, Mar. 2000.

[2] *ISO/IEC 14496-2 Info. Tech. -Coding of audio-visual objects- Part 2: Visual,* 3nd edition, April, 2003.

[3] Texas Instruments, "TMS320C55x Hardware Extensions for Image/Video Applications Programmer's Reference," *TI Technical Document SPRU098*, Feb. 2002.

[4] S. De-Gregorio, M. Budagavi, and C. Chaoui, *Bringing Streaming Video to Wireless Handheld Devices, Texas Instrument Technical White Paper SWPY005*, May 2002.

[5] J. Chaoui et al., *OMAP: Enabling Multimedia Applications in Third Generation (3G) Wireless Terminals*, *Texas Instrument Technical White Paper SWPA001*, Dec. 2000.

[6] Thanh Tran, *OMAP 5910 Video Encoding and Decoding*, *Texas Instrument Application Report SPRA985*, Dec. 2003.