A SOFTWARE/HARDWARE CO-DESIGN METHODOLOGY FOR EMBEDDED MICROPROCESSOR CORE DESIGN

Yong Zhang[1], IEEE Member, Kai Kuang Ma[1] IEEE Senior Member and Qindong Yao[2]
[1]School of Electronic &Electrical Eng., S2, Nanyang Tech. Univ., Singapore, 639798
[2]Inf. & Electronic Eng. Dept., Yuquan Campus, Zhejiang Univ., Hangzhou, Zhejian, P.R. China, 310027
Email: eyzhang@ntu.edu.sg

ABSTRACT: The increasing use of Embedded Microprocessor Core (EMC) has become a popular trend for the embedded systems. The EMC development is a complex interactive process with the algorithm, the software and the hardware. This paper presents a software/hardware co-design methodology to deal with the EMC design problems. The proposed stepwise method extracts the algorithm's features and maps them onto the EMC's architecture by performing the analysis on the system level. The optimization is taken in both software and hardware domains so that the designed EMC can be specific to the algorithm. An EMC design for the variable length coding (VLC) applications is exemplified in the paper to illustrate the design procedure.

Keywords: Software/Hardware Co-design, Embedded Microprocessor Core.

1. Introduction

The rapid evolution of current consumer electronic systems has been demanding the embedded products extremely short time-to-market simultaneously with very low costs. Recent introduction of embedded microprocessor core (EMC) technologies makes this requirement possible by integrating a programmable engine onto a single system chip. The benefits of using EMC are from three aspects: Firstly, short product design cycles can be achieved by implementing large parts of the system functionality on the integrated processor core. Secondly, the system cost can be reduced significantly thanks to the customization of the core's architecture for the specific application. In addition, large flexibility can be achieved because the embedded software running on the EMC can be revised quickly. Currently, the increasing use of EMC has become a clear trend for the embedded systems.

Similar with the generic microprocessor[1], an EMC is the integration of software and hardware components. The instruction set architecture (ISA) and the corresponding micro-organization should be developed coordinately during the system design. Furthermore, since various embedded systems are focused on their own narrow domains, the algorithms employed in different applications always have different behaviors. The custom EMC architecture is therefore required to adapt to the specific algorithm. Hence, the design of an EMC is an interactive process with the algorithm, the software and the hardware. However, although the boundaries of these three components are clear, their relationships are often unknown before the EMC designs begin. For this reason, today's EMC designs have to be heavily based on earlier experiences with similar products and on manuals or literatures [2]. Apparently, with the broadening application areas and the increasing complexity, such method is very insufficient because the previously prototypes are either absent or obsolete.

In this paper, we will present a software/hardware co-design methodology to tackle the EMC design. The proposed stepwise method extracts the algorithm's features and maps them onto the EMC's architecture in terms of the system level analysis. The optimization will be taken in both software and hardware domains so that the resulted EMC will be customized to the algorithm. Since no previous prototype is required, the methodology is suitable for any novel embedded application. In order to demonstrate its efficiency, this paper will also involve an EMC design example for the variable length coding (VLC) algorithm [3], which is commonly employed in the embedded video compression systems [4].

2. Software/Hardware Co-design Methodology

Software/hardware co-design originated from co-synthesis concept in the electronic computer-aided design (ECAD) [5]. In co-

Contributed Paper
Manuscript received September 16, 1999

synthesis scheme, a Control Data Flow Graph (CDFG) is first constructed from a given specification of the specific algorithm, then a suitable datapath is derived by scheduling and allocation. The datapath closely reflects the structure of the algorithm. Although this method has been successfully used in many applications which have a small number of operations (e.g., digital filter designs), it is not feasible for the EMC design where the operation amount is enormous and undetermined. This is because of not only the complex allocation procedure but also the extremely long simulation time [6]. Moreover, the co-synthesis method also provides poor flexibility for the design because the algorithm's features are mapped to the hardware directly. Therefore, a more abstract methodology should be developed for EMC design.

From the system-level view, although the micro organization of a typical EMC architecture provides the raw computing power, most of the features of the algorithm, are not directly implemented in the hardware, but instead designed into the application software. According to this essential feature of EMC, we explore a software/hardware co-design scheme as following four steps:

Step 1: Complexity analysis

The algorithm's requirement for the processing power is an important parameter for EMC design which determines the lower bound of the EMC's processing ability and limits the design space. Hence, the complexity of the embedded algorithm is computed in the first step of the whole design.

Since no complexity metric (e.g., million instructions per second - MIPS, or million operation per second - MOPS) is meaningful without presenting the processor's architecture, a uniform hardware model should be established before any complexity analysis starts. Meanwhile, the predefined model should be a generic one with little specific architecture in order to avoid biasing the statistics data. In this paper, a virtual RISC (Reduced Instruction Set Computing) core model (VRCM) is employed. The main features of the model can be expressed as follows:

- One instruction represents one basic operation, including computations, memory access and branch control for the program;
- Program flow is executed sequentially and one instruction is issued in one machine cycle;
- Having a primary datapath which can finish one ordinary ALU operation in one cycle.

In order to analyze its complexity, the algorithm is firstly translated to the VRCM's instruction stream, then the instruction number is recorded by dynamically executing the program on the VRCM. The complexity of the algorithm is consequently represented by the MIPS metric. This parameter can be used as a reference for future EMC architecture choice.

Step 2: Parallelism analysis

Although the complexity analysis provides us the primary trait of the algorithm, it is not sufficient for the settlement of the initial EMC structure because the complexity parameter doesn't unveil any relationship between the algorithm and the target software/hardware architectures. More important attribute for EMC design is the parallelism inside the algorithm. In the real-time embedded system, if the algorithm's parallelism is exploited and mapped to the architecture correctly, high throughput with low hardware cost can be achieved. For this reason, parallelism is considered as a key factor bridging the algorithm and software/hardware architectures in our method.

In this step, the VRCM's instructions are firstly parallelized (e.g. using VLIW - very long instruction words technology [7]) and executed (e.g. superscalar implementation [8]) in terms of different parallel architecture candidates. The performance of each architecture is quantitated with its speedup over the VRCM. A winner who gains the highest speedup will be chose as the initial architecture for further optimization.

Step 3: Operation pattern analysis

The occurrence probabilities of the operations in the algorithm are recorded by dynamically analyzing the executed instructions in this step. According to the resulted statistical data, the most frequently used instructions are selected to construct the EMC's ISA. Simultaneously, the hardware structure can also be ameliorated based on these data from the initial hardware framework established in step 2.

With the above approaches, the operations with different traits can be mapped to different functional parts. Furthermore, the hardware organization can be focused on the components corresponding to the most frequently used operations. In some sense, the target of this step

is similar with that of the allocation in co-synthesis method. However we don't map each operation to the hardware directly, but instead study the statistical characteristics of all the operations in the algorithm.

Step 4: Timing analysis

The last step is an optimization process for the EMC in term of the hardware profile. Since the hardware details have been built in the preceding steps, more accurate timing information can be extracted at this point. The EMC hardware components are then optimized according to these timing constraints, and correspondingly, fine ISA adjustment may be required in this step.

It should be noted that this stage also has a verification function for the EMC design. Since the hardware information is absent when the initial architecture is selected in step 2, the system architecture derived may fail to be implemented due to some physical constraints. For example, the architecture requiring the 32-port extern memory cannot be taken as a good design because such memory is either expensive or cumbersome. In this case, the design flow will be return the step 2 and find other initial architecture whose speedup is suboptimal.

Fig 1 shows the proposed software/hardware co-design flow chart.

3. An EMC Design for VLC Applications

VLC is an entropy coding algorithm widely used in video compression systems, such as digital TV, videophone and videoconfereeing. It is normally implemented as an independent part in the system because its characteristics are different from other components in the video compression scheme [9]. In this section, We will present an EMC design for the VLC algorithm. The applications we demonstrate here are H.263 [10] and MPEG-2 [11], which are two international standards for video compression.

Step 1: Complexity analysis

The VLC complexity is firstly computed with the VRCM presented in Section 2. As shown in Table 1, if the cycle per instruction (CPI) of EMC approaches one, VLC can be executed in real-time where the system clock frequency is around 20-40 MHz.

	H.263	MPEG-2
Complexity (MIPS)	18	35

Table 1. VLC Complexity

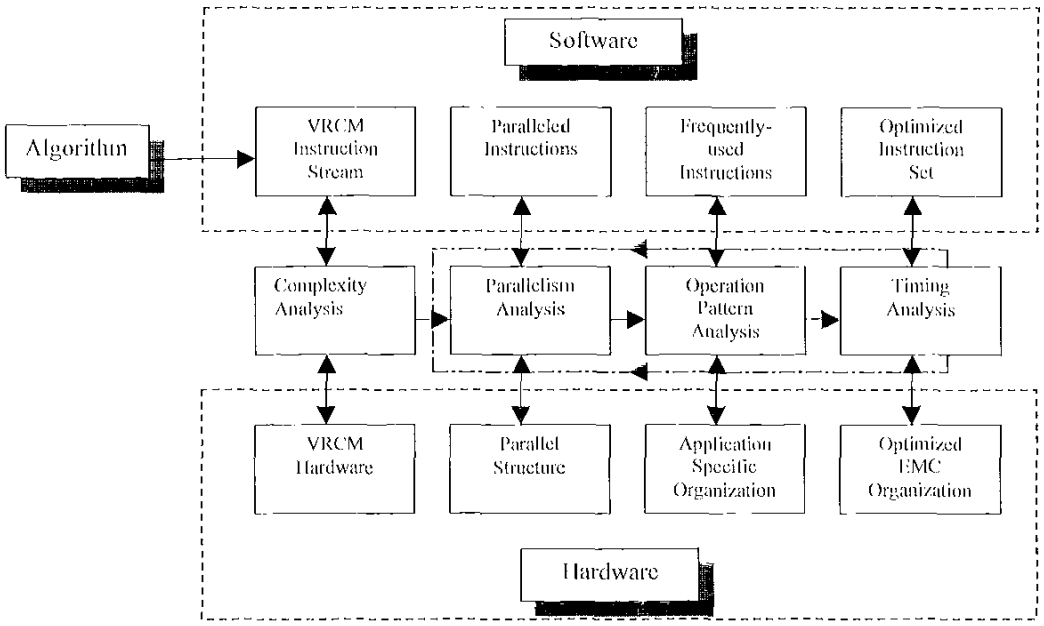

Fig. 1. Software/hardware co-design flow

Step 2: Parallelism analysis

Step 1 has shown that VLC is not a high power-consuming task, therefore, we choose the pipeline technology as a parallel implementation for EMC design because pipeline can gain moderate speedup whereas need less hardware overhead.

Normally, there are three types of pipeline architectures frequently used in embedded products,

- 3-stage pipeline, the pipeline segments are Instruction Fetch (IF), Instruction Decode (ID) and Execution (EX);
- 4-stage pipeline, the segments are IF, ID, EX and Results Written Back (WB);
- 5-stage pipeline, the segments include IF, ID, EX, Memory access (MEM) and WB.

The speedup obtained from pipeline is dependent on the feature of the target algorithm because both of the limited parallelism in the algorithm and the pipeline hazard happened during algorithm execution impact the speedup drastically. The speedup gained by the pipeline can be computed as follows [1]:

$$speedup = \frac{pipeline\ depth}{1 + pipeline\ stall\ cycles\ caused\ by\ hazard} \quad (1)$$

The results of each kind of pipeline speedup for VLC are shown in Table 2. It can be seen that the 5-stage pipeline gains the highest speedup. So the 5-stage pipeline is chose as the initial architecture of EMC.

	3-stage	4-stage	5-stage
Speedup for H.263	2.19	2.45	2.87
Speedup for MPEG-2	2.13	2.34	2.72

Table 2. Different pipeline speedups

Step 3: Operation pattern analysis

The operations in the VLC algorithms can be divided into three types, computation, branch control and memory access. Their occurrence frequencies are shown in Fig. 2. The EMC architecture will then be optimized in terms of the characters of each category.

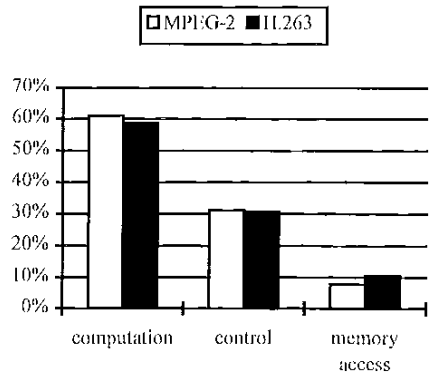

Fig. 2. Operation pattern in VLC algorithm

As shown in Fig. 3, there are three types of operations in the computation category. It can be seen that besides the ordinary add/sub and shift operations, compares also occupy large percentage in the category. Therefore, the specific comparison logic is added to the common ALU. Furthermore, since the comparison operations are often preceded by the add/sub/shift operations, the forwarding logic [1] is also supplemented to the datapath to minimize the data hazard.

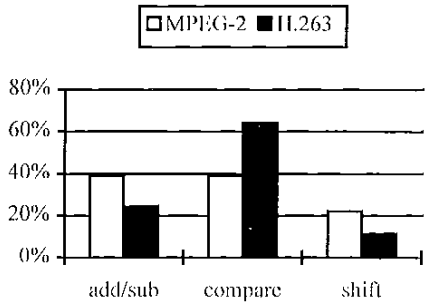

Fig. 3. Operation pattern in computation category

With these optimizations for the computation logic, the obtained average speedup is 1.12 in terms of the equation (1), Fig. 2 and Fig. 3.

Another class of VLC operations, branch control, can be subdivided into two types, i.e., loop backward branch and if-then-like forward branch (Fig. 4). Both of two branches may introduce control hazard to pipeline execution, which will cause a great performance loss for pipeline. In order to decrease the control hazard, the branch logic should be elaborated carefully

so that the branch decision can be made as early as possible. We employ condition code technology [1] to deal with the forward branch, which can assure EMC makes the branch decision just after the instruction is fetched, i.e. in ID stage. In addition, the loop counter and the return address register are introduced into the branch logic to ensure the decision stage of loop branch is the same with that in the forward branch [6].

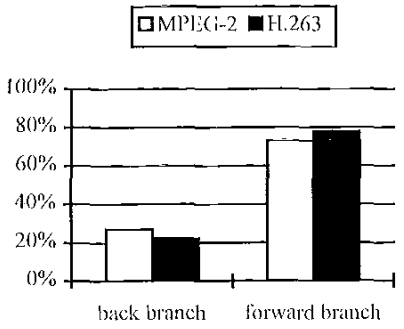

Fig. 4. Operation pattern in branch category

The speedup obtained after suppressing the control hazard is 1.36.

The main problem for memory access is that the read/write speed of the extern memory cannot keep up with the processing speed of the EMC datapath. In order to solve this problem, the data source/destination in memory access operation is analyzed. As shown in Fig 3, most memory accessing operations are reading/writing data which come from an 8*8 image block. Therefore, these data can be accessed continuously. We use a pipeline read/write method to speedup memory access operation [6]. The acquired speedup is 1.1.

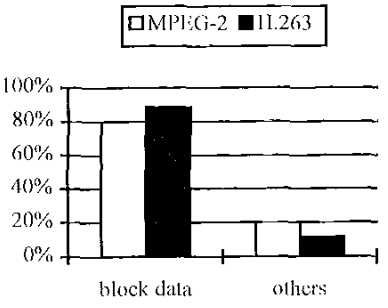

Fig. 5. Source/destination pattern in memory access category

Step 4: Timing analysis

Although the deep pipeline can acquire high speedup, it also spends many hardware resources and introduces serious clock skew problem. In the following, we will compress the EMC's pipeline stages by timing analysis without losing the earned speedup.

In an ideal pipeline, the achieved speedup is only decided by the pipeline clock cycle. This value, named T, is equal to the longest stage duration in the pipeline. Specifically, in the above 5-stage pipeline, we have

$$T_{pipeline} \geq MAX\{T_{IF}, T_{ID}, T_{EX}, T_{MEM}, T_{WB}\} \quad (2)$$

The operations in each pipeline stage are depicted as follows:

IF	ID	EX	MEM	WB
PC addressing	Instruction Decoding	ALU Operations	Memory Access	Writing Registers
Instruction Reading	Reading Register Index	Forwarding		
	Reading Operators			
	Branch Decision			

Fig. 6. Operations in each pipeline stage

Simulation shows the ID stage has the longest time duration among all five stages, i.e.:

$$T_{pipeline} = T_{ID} \quad (3)$$

We then compress the pipeline stage under the constraint of (3).

It can be seen from Fig 6 that the fourth stage of the pipeline doesn't do any operation for the computation instructions. In addition, due to the pipeline memory access mode is adopted, the EX stage is not required when reading/writing the block data. Therefore, the EX and MEM stages can be merged into one stage. The execution of the modified pipeline is shown in Fig. 7.

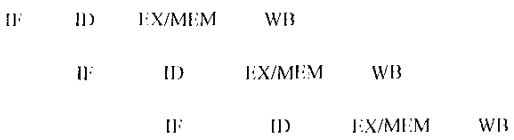

Fig. 7. Modified 4-stage pipeline

The simulation shows that 10% of the hardware resources are saved by this pipeline compaction.

Following the above 4-step design flow, the EMC has been prototyped by Field Programmable Gate Array (FPGA) and implemented successfully as a softcore for embedded VLC applications [6]. Some features of this EMC are shown in Table 3.

Number of Instructions	74
CPI	1.2
Number of Gates on FPGA	15,000
Number of Gates on softcore	9,000

Table 3. Some features of the proposed EMC

4. Conclusion

In this paper, we present a novel software/hardware co-design methodology for EMC design. Based on the generic model VRCM, an initial EMC architecture is firstly selected by analyzing the complexity and parallelism of the target algorithm. Then software and hardware components are optimized concurrently according to the algorithm characteristics. An EMC design example has been shown to demonstrate the design procedure. Although the proposed methodology is employed for the EMC applications, it can also be generalized to other kinds of embedded system designs [6].

REFERENCES

[1] D. A. Patterson, J. L. Hennessy, "Computer Achitecture a Quantitative Approach", Morgan Kaufmann Publishers Inc., 1996.
[2] S. Edward, *et al.*, "Design of Embedded Systems: Formal models, Validation and Synthesis", Proceeding of IEEE, vol. 85, no.3, pp. 366-390, 1997.
[3] A. M. Tekalp, "Digital Video Processing", Prentice Hall, 1995.
[4] G. Goossens, *et al.*, "Embedded Software in Real-Time Signal Processing Systems: Design Technologies", Proceeding of IEEE, vol. 85, no. 3, pp. 436-454, March, 1997.
[5] G. D. Micheli, *et al.*, "Hardware/Software Co-Design", Proceeding of IEEE, vol. 85, no. 3, pp. 349-365, 1997.
[6] Zhang Yong, "Research on Video Encoder Design", Ph. D. Dissertation, Zhejiang University, 1999.
[7] P. Clarke, "Compressed VLIW meets multimedia", Electronic Engineering, Dec. 1996.
[8] J. Smith and S. Weiss, "PowerPc601 and Alpha 21064: A Tale of Two RISCs", Computer, vol. 27, no. 6, pp. 46-58, Jun. 1994.
[9] Pirsch, N. Demassieux, and W. Gehrke, "VLSI Architectures for Video Compression -- A Survey", Proceeding of IEEE, vol. 83, no. 2, pp. 220-245, Feb. 1995.
[10] ITU-T Recommendation H.263, "Video Coding for Low Bit Rate Communication", 1996.
[11] ISO/IEC 13818, "Generic coding of Moving Pictures and Associated Audio", 1994.

BIOGRAPHIES

Yong Zhang, received the B.S. degree and M.S. degrees at Southeast University, China, in 1993 and 1996 respectively. In 1999, he received the Ph.D. degree from Zhejiang University, China. Currently he is a Research Fellow at Electronic and Electrical Engineering School of Nanyang Technological University, Singapore. His research interests include video communication, image coding and related VLSI implementation. Zhang Yong is an IEEE Member of both ComSoc and Signal Processing Society.

Kai Kuang Ma, is an Associate Professor of Electronic and Electrical Engineering School of Nanyang Technological University, Singapore. His research interests focus on various aspects of digital video processing, such as coding, indexing and wireless transmission. He is currently the Chairman and Head of Delegation of MPEG in Singapore.
Kai Kuang Ma is a Senior Member of IEEE.

Qingdong Yao, is a Professor in the Department of Information & Electronic Engineering at Zhejiang University, P. R. China. His major research interests include digital communication, parallel signal processing and HDTV.