行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※※
※　　　　　　　　　　　　　　　　　　　　　　　　※
※ **具有信號處理能力的嵌入式微處理機的實現及應用－** ※
※ **子計畫一：嵌入式微處理機微架構設計與驗證及其模** ※
※ **擬軟體** ※
※　　　　　　　　　　　　　　　　　　　　　　　　※
※※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：□個別型計畫　　■整合型計畫

計畫編號：NSC 89-2218-E-009-057-

執行期間：89 年 8 月 1 日至 90 年 7 月 31 日

計畫主持人：　吳全臨　教授
計畫參與人員：王仕杰　國立交通大學
林銘豐　國立交通大學
胡文祥　國立交通大學
明　權　國立交通大學
練彩茹　國立交通大學
蘇耀群　國立交通大學
闕國勳　國立交通大學

執行單位：國立交通大學資訊工程學系

中　華　民　國　90　年　7 月　31 日

具有信號處理能力的嵌入式微處理機的實現及應用－

子計畫一：嵌入式微處理機微架構設計與驗證及其模擬軟體

計畫編號：NSC 89-2218-E-009-057-

執行期間：89 年 8 月 1 日至 90 年 7 月 31 日

計畫主持人：　吳全臨 國立交通大學資訊工程系教授

計畫參與人員：王仕杰 國立交通大學資訊工程系

林銘豐 國立交通大學資訊工程系

胡文祥 國立交通大學資訊工程系

明　國立交通大學資訊工程系

練彩茹 國立交通大學資訊工程系

蘇耀群 國立交通大學資訊工程系

闕國勳 國立交通大學資訊工程系

Abstract

IAM 2000S (Information Appliance Microprocessor with Digital Signal Processing Capability) is a 32-bit hybrid RISC(Reduced Instruction Set Computer) / DSP(Digital Signal Processing) microprocessor which is targeted for IA applications. It integrates 32-bit RISC processor architecture and DSP strongly capable instruction set. The processor core is designed and implemented in register transfer language.

The core is verified and works well in simulation environment. Major performance parameters include the followings.

- Power consumption (at 2.5V / 0.25 cell library)：301mW ;
- Clock rate：180MHz ;
- Gate count：29860 gates ;
- Number of registers：44 general purpose registers , 3 status registers ; and
- Number of modes：5 (USR , USR1 , USR2 , PRI , SYS).

It is expected that the clock rate can reach 300MHz in the next version.

In this article, we will discuss micro-architectare design and RTL design of IAM 2000S processor core. An efficient verification system is also created to help verify our design.

1.Introduction

In post PC era, information appliance(IA) market creates new application demands which enlarge the application domain of signal processors. Information appliance applications: cellular phones, voice messaging, set-top box, all require control and signal processing capability [1,2]. In the past, two separate chips Reduced Instruction set Computer (RISC) microprocessor chip and Digital Signal Processing (DSP) microprocessor chip are used in a single system. But under the new market demands for smaller, quicker, and more modern chips, many core technology suppliers are diligently working on an integrated processor chip. This integrated processor employs single chip socket, single operating system, and single development system. This simplified architecture will reduce manufacturing cost significantly and also meet modern demands for smaller and quicker chips.

Toward this trend, we design IAM 2000S (Information Appliance Microprocessor with DSP Capability), a 32-bit hybrid RISC/DSP microprocessor , which is targeted for IA applications. It integrates 32-bit RISC processor architecture and DSP strongly capable instruction set. In this article, we will discuss micro-architecture design and Register Transfer Language (RTL) implementation of IAM 2000S processor core. An efficient verification system is also created to help verify our RTL design.

The rest of this article is organized as follows.

Section 2：Micro-architecture ;

Section 3：Data on IAM 2000S ;

Section 4：Instruction set ;

Section 5：Decoder ;

Section 6：Controller ;

Section 7：Data path ;

Section 8：Embedded ICE ; and

Section 9：Auto-comparison verification system .

2. Micro-architecture

The IAM 2000S is a 32-bit hybrid RISC/DSP microprocessor core which targets on information appliance applications. The IAM 2000S integrates DSP capability into a 32-bit RISC microprocessor to increase its strength in DSP applications. Below we will highlight the micro-architecture of IAM 2000S.

Instruction Set

The design of the IAM 2000S' instruction set is based on load-store architecture. All data processing instructions use internal registers as the source and destination operands. Data to be processed are first loaded from memory into registers by load instructions. To support DSP applications, the IAM 2000S has some special instructions to enhance its capability in the DSP application domain. The LDM/STM (load/store multiple registers) instruction can load/store multiple registers from/to memory by only executing a single instruction. IAM 2000S also has several types of multiplication instructions for different types of DSP applications. The instruction set can be further extended by using a coprocessor . Fifty eight instructions are implemented in the first successful RTL core.

Pipeline Execution

The IAM 2000S uses a pipeline to increase the execution rate. The instruction pipeline is divided into five stages, consisting of fetch, decode, execution, memory access, and write back. This makes it possible for IAM 2000S to have a clock speed at 180 MHz using a 0.25 μm cell library. With a further architectural optimization , the clock speed can possibly reach to 300 MHz. The IAM 2000S also has data forwarding paths between the pipeline stages to further enhance its performance.

Processor Mode

The IAM 2000S has five processor modes: User Mode, User1 Mode, User2 Mode, Privileged Mode, and System Mode. Each processor mode has its special function domain. The User Mode is used for executing normal programs. The User1 Mode and User2 Mode are used for executing specific user programs. The Privileged Mode is used for exception handling. The System Mode is used for executing system programs. Task switching time between two modes of processors is reduced to a minimum with a register window overlap. Table 1 summarizes tasks performed under different processor modes.

Table 1 IAM 2000S Processor Mode

Processor mode	Description
User	Normal program execution mode
User1	Specific user program execution mode1
User2	Specific user program execution mode2
Privileged	Exception mode, including: 1.FIQ 2.IRQ 3.Supervisor 4.Abort 5.SWI 6.Undefined
System	Run operating system tasks

Registers

The IAM 2000S has 47 registers in total. 44 of them are the general-purpose registers (GPR) and the other three registers are the status registers. The 44 GPRs are used as source operands or destination operands of all instructions while the status registers holds the information about the processor state and the processor configuration. The register organization is shown in

Table 2. IAM 2000S operates on different sets of registers while it is in different processor modes. For example, IAM 2000S is allowed to access R0~R7,R8_pri~R14_pri, R15, CSW, and SSW_pri when it is in the Privileged mode and access to other registers is not allowed.

Table 2 IAM 2000S Register Organization

User	User1	User2	Privileged	System
R0~R7	R0~R7	R0~R7	R0~R7	R0~R7
R8~R14	R8_usr1~ R14_usr1	R8_usr2~ R14_usr2	R8_pri~ R14_pri	R8_sys~ R14_sys
R15(PC)	R15(PC)	R15(PC)	R15(PC)	R15(PC)
CSW	CSW	CSW	CSW	CSW
			SSW_pri	SSW_sys

Memory System

The IAM 2000S adopts the Harvard memory architecture. It has the separate instruction memory bus and data memory bus. Thus the IAM 2000S can fetch an instruction and access a data memory location at the same time. The memory control signals are pipelined to let the memory system have a full clock cycle to perform a memory access.

3.Data on IAM 2000S

There are some differences between IAM 2000S and ARM9E-S. Three major issues: technical issue, performance and register model are used in comparison as shown in Table 3.

Table 3 A comparison Between IAM 2000S and ARM 9E-S.

	IAM 2000S	ARM9E-S
Power consumption (at 2.5V/ .25 cell library)	301mW	288mW
Clock rate (at .25 cell library)	180MHz	160MHz
Gate count	29680 gates	NA
Number of registers	44 general-purpose registers 3 status registers Total : 47	31 general-purpose registers 6 status registers Total : 37
Number of modes	5(USR, USR1, USR2, PRI, SYS)	6 1st: USR 2nd~6th: exception modes

The registers of IAM 2000S are more than ARM9E-S. The additional 2 modes are USR1, USR2. The main advantage is handling context switch more efficiently. There are additional 7 registers for USR1 or USR2 mode. Instead, there is only one exception mode called PRI in IAM 2000S. The advantage is handling exception more uniformly.

Technical issue:

A main difference between IAM 2000S and ARM9E-S is “instruction cycle time”. For a multiply instruction, we increase the ability of ALU so that a multiply instruction can be executed in one cycle and a multiply-long instruction can be executed in two cycles. Flags generated by multiply instruction are forwarded if necessary. Instead, for ARM9E-S, forwarding is not used when flags generated by the multiply instruction are necessary to be forwarded ; and a multiply instruction are executed in two cycle and a multiply-long instruction is executed in three cycle.

The design of the register file module is also different. For IAM 2000S, there are 3 read ports and for ARM9E-S, there are only 2 read ports. Therefore some data-processing instructions that are executed in two cycles in ARM9E-S and all data processing instructions including Q-type instructions are executed in one cycle in IAM 2000S. IAM 2000S has a more powerful ALU.

The instruction cycle of LDM/STM for IAM

2000S is just a little bigger because LDM/STM instruction is fully executed by controller (not including ALU). In this design, the clock rate can be increased even if the instruction cycle is just bigger. When exceptions or debug trigger happened, the instruction cycle time is not defined in the specification of ARM9E-S. However, when exceptions happened, the fetcher works in the next cycle in IAM 2000S, so the handling time is very short. And so is the debug trigger.

For other cases, the differences also exist. From the whole aspect, the performance of IAM 2000S is just a little better than ARM9E-S.

4.Instruction Set

Since ARM has a popular market in IA applications , we seek no confrontation with ARM's patent rights. There are 2 patents concerning ARM instruction set in America. One is related to the transformation between thumb and arm instructions; the other one is instructions with conditional execution claimed in the multiply-accumulation process. Since IAM 2000S excludes thumb instructions, the latter is what we don't take into consideration.

First, we do not use all condition codes from each instruction except branch instructions to avoid claims in the patents. Leaving branch instructions with a condition code makes transformation between the ARM9E-S and the IAM 2000S instruction sets easier. In fact, an evaluation proves that the instruction set without condition codes perform better in DSP purpose applications, because the large amounts of computation requests are issued without conditional execution here.

Second, reducing decoding levels from four at the ARM9E-S instruction set to one at the IAM 2000S instruction set. Instead of repeatedly decoding bits 7~4 in ARM9E-S decoding stage, IAM 2000S replaces these important decoding messages to bits 28~25. Then moving original bits located at position 27~25 to 31~29, we obtain the most significant 7 bits in the head of each instruction. Therefore the IAM 2000S instructions not only occupy the position of condition codes in ARM9E-S instructions but also demand only one decoding level to pass internal signals out at very beginning. Because of this simplicity, lots of instruction encoding spaces are left in each category for future extension where the ARM9E-S microprocessor tries its best to fill unpredictable encoding position with different instruction categories.

We summarize what was mentioned above into three contributions from the IAM 2000S instruction set. Take off condition codes for higher performance under DSP purpose applications and avoiding the claims of the patent. Reduce decoding level from 4 to 1. Separate each instruction category clearly with loose encoding format embedded.

5.Decoder

Based on the IAM 2000S instruction set encoding, it could easily pass related control signals to Controller and DataPath under one decoding level. The decoding tree presents the simple decoding level while identifying each instruction category.

Synthesizing decoder core by using the SYNOPSYS design compiler under CIC 0.35ì m and WSMC 0.25ì m cell library, we know the performance of the decoder in area、timing、critical path and power (See Table 4 and Table 5). This performance information reports that the decoder unit of the IAM 2000S microprocessor uses about 5200 and 4000 transistors, corresponding to 3.5 times the reguired area under CIC 0.35ì m and WSMC 0.25ì m technology respectively. Since the decoder phase occupies half of a clock cycle time while the other half is reserved for operand preparation, which is reading operation source from the register bank, we should estimate maximum clock frequency as 2 times the timing showed below. Therefore

the clock frequency can be 400MHz or 700MHz depending on CIC 0.35ì m or WSMC 0.25ì m technology is used.

Table 4 Decoder Performance under CIC 0.35ì m technology

CIC 0.35ì m	Area (gates)	Timing (ns)	Critical path (gate numbers)	Power (mW)
No constrained	435	5.5	11	40
Min timing	1483	1.2	8	125

Table 5 Decoder Performance under WSMC 0.25ì m technology

WSMC 0.25ì m	Area (gates)	Timing (ns)	Critical path (gate numbers)	Power (mW)
No constrained	715	2.2	10	32
Min timing	1152	0.7	6	53

6.Controller

The controller is the most critical part among all modules of IAM 2000S. The controller carries out some major jobs: (1) fetch the instruction from the instruction memory; (2) ensure the accuracy of pipeline flow when stall or flush occurs;(3) handle the exception and debug mode; (4) communicate with memory and coprocessor.; and (5) control the five pipeline stages at anytime. The five pipeline stage, are fetch, decode, execute, memory-access, and write-back. The controller takes control over all other modules in each time, so other modules are all combinational logic. Operation of other modules are monitered by the controller (Some are all D-flip-flops). In the IAM 2000S design, the modules are: RF (register file), RD (register address decoder), Decoder, and ALU. In the fetch stage, the controller decides the appropriate PC program counter in the previous cycle and latches the PC to feed to instruction memory. In the decode stage, it latches the instruction data placed in IR (instruction register), and Decoder module works to decode basic control signals. The ALU inputs from RF are prepared and the register data are decided by RD module, In this stage, the forwarding logic works in this cycle, too. In addition, if previous instruction needs to be executed for multi-cycles, the next instruction which is in the decode stage is not to be decoded and appropriate signals are cleared by controller. In the execute stage, the ALU calculates its result and flag which are propagated through two D-flip-flips for write-back. In this stage, flush and stall control works for the current instruction and data dependency is checked in the cycle, too. In the memory-access stage, the controller decides whether reading data from memory, or writing data to memory, just sending outputs of the decoder. If this is a LOAD or STORE instruction, a memory request is necessary in this cycle. For the coprocessor instruction, either the IAM 2000S accesses coprocessor in this stage, or the coprocessor accesses memory in this stage. It is in the write-back stage, the data is written in the RF module on the falling clock edge. So, an instruction is retired in this cycle.

For the exception and debug control, the exception handler and debug handler works in each stages. If more than an exception happened, the concurrency and priority arechecked and the controller decides which one exception is taken. When an exception really happened and is taken, the controller flushes 4 cycles and PC exception vector is decided immediately. For the pipeline control, the debug handler works the same way as the exception module does. However, in the different part, the exception handler will do the following tasks when an exception occurs : save CPSR to SPSR, save PC+4 to register14 and assign PC to exception vector address. The fetcher doesn’t work in the debug mode, and all instructions are fed from eICE. In this mode, the pipeline stages is not used. So when an instruction is retired, it will flush 4 cycles.

7.Datapath

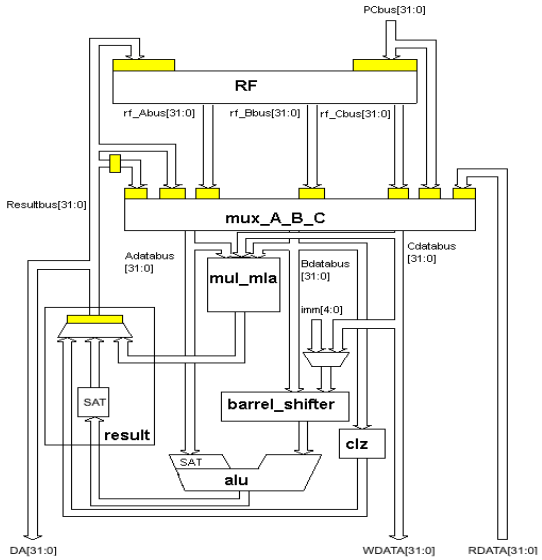

Figure 1 IAM2000 Datapath Architeture Diagram

We probes into how to establish no-multiple-cycle instruction execution and designed the corresponding architecture. Fig 1 presents the datapath diagram. It includes RF module, a multiplexor module , multiply module , SAT module , barrel shifter , CLZ modules and ALU.

We analyzed the differences among various implementation methods according to different performance requirements and identified most-fit-architecture method. The result shows that the best single execution cycle implementation can obtain 182MHz clock rate. On the hand , if the critical path is divided into two cycles , we can obtain a 339MHz implementation. This single-execution-cycle architecture is about 1.67 times better than the performance of the ARM9 Processor Families and the double-execution-cycle architecture is about 3.11 times better than that due to shorter program-executing time.

8.Embedded ICE

The IAM 2000 includes hardware features that support in-circuit emulation. Figure 2 shows the architecture of these facilities in IAM 2000S.

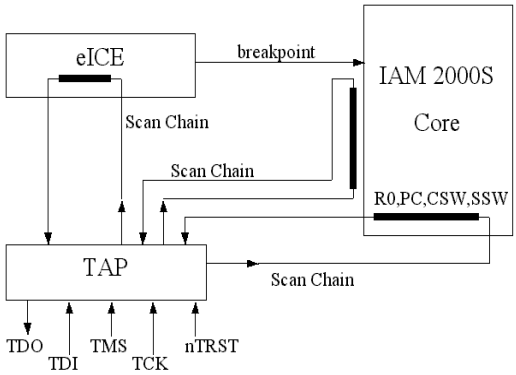

Figure 2 IAM 2000S Embedded ICA Architecture

The eICE module is used to generate breakpoints and to control the core execution. The TAP module is compliant with the JTAG standard and is responsible for scanning data into or out of the core. Breakpoints are generated on a given instruction fetch or a data access. After a breakpoint is recognized, the core will stop execution and enter the debug state in which the debug tasks are performed. In the debug state, the core states (register contents) can be observed or modified. Single step execution is also supported in this architecture.

9.Auto-comparison Verification System

An efficient verification method is required to verify IAM 2000S RTL design and reduce the effort wasted during the verification period. The conventional simulation-based approach has some drawbacks. For example, it is not an automatic system such that additional time is required for verification. Moreover ATPG cannot generate the test vector based on the coverage analysis. Hence, a new integrated verification system was developed, which includes the following criteria: automatic, quality, reusability, complexity and applicability. Our verification system was constructed following the previous criteria to verify the entire IAM 2000S RTL code. The verification system contains some components. Figure 3 shows the architecture of the integrated verification system.

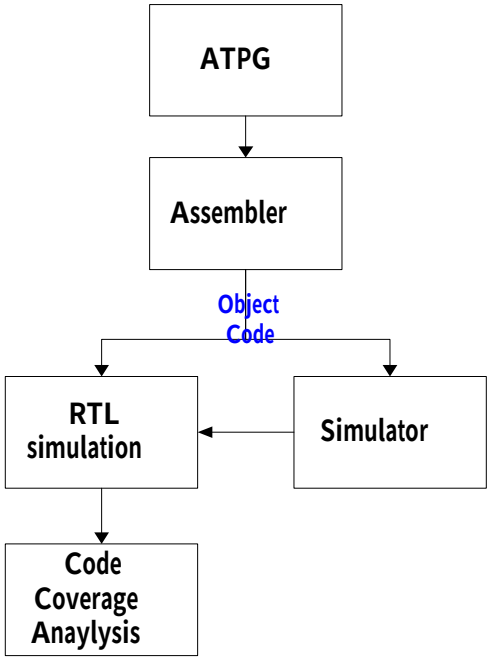

Figure 3 The architecture of Integrated Verification System

The integrated verification system contains 5 components, and each function is described as follows:

1. ATPG generates the test programs randomly to address high coverage measurement based on the previous coverage analysis ;
2. The assembler translates the IAM2000s assembly code generated by ATPG to IAM2000s object code ;
3. The functional behavioral model generates the expected results (golden pattern) of each instruction ;
4. One comparison behavioral module compares the results generated between the by RTL code simulation and by the simulator; and
5. Using coverage analysis measures the RTL verification quality.

Each component is independent as well as the entire system is automatic without the engineer intervention. Over one billion IAM 2000S instructions were executed in the verification process , Details of the verification system is presented in another article [5].

10.Summary and Future Work

In this article, we introduce an innovative microprocessor-IAM 2000S and the design features are also discussed. Since the RTL design of IAM 2000S has been finished and verified, we could move toward the back-end design stage. Actually, the physical implementation is in progress currently. Other function blocks ,memory management unit (MMU), and cache, are under development now. After the processor is fabricated, we will use it to build a VOIP system.

Reference

[1] *IAM 2000S Micro-architecture Specification*.

[2] *IAM 2000S Instruction Set Manual*.

[3] Steve Furber, *ARM: System-On-Chip Architecture*, 2000.

[4] ARM, *ARM9E-S Technical Reference Manual*, 1999.

[5] M.C.Huang , Y.C. Su , and C.L Wu “Mixed Mode Verification for 32-bit Microprocessor Core”, Submitted for publication, 2001.