低功率數位訊號處理器智產核心
Low Power DSP Processor Core

計畫編號：NSC90-2218-E-009-037
執行期限：90 年 8 月 1 日至 91 年 7 月 31 日
主持人：任建葳 國立交通大學電子工程學系
參與人員：林泰吉、張金祺、張育銘、林建宏

一、中文摘要

我們的計畫目標為設計一個應用在無線通訊的可程式化數位訊號處理器(programmable DSP, or DSP processor)，它擁有以下特性：(1)高效能(>2,000MOPS)、(2)高程式密度(good code density)、(3) 低功率(<1mW/MOP)及(4)可重新組態。本年度計畫完成的項目包括了新一代 DSP 處理器的相關資料搜集與整理、提出了一個可變長度的超長指令集 DSP 處理器架構，同時也提供單一指令多重資料(SIMD)處理的能力，我們也完成了其指令集模擬器及其應用於數個重要的數位訊號處理應用的效能評估(包含 DLMS、motion estimation、Viterbi decoding)。
關鍵詞：數位訊號處理器、矽智產、低功率、指令集模擬器

Abstract

This project is to develop a programmable digital signal processor (programmable DSP or DSP processor) for wireless communications, which features: (1) high performance (> 2,000MOPS), (2) good code density, (3) low power (<1mW/MOP), and (4) configurability. Various architectures of the state-of-the-art DSP processors are surveyed and we have proposed a new variable-length VLIW DSP with SIMD capability. We have also constructed its instruction set simulator (ISS) and evaluated the performance to execute several DSP kernels, including DLMS, motion estimation, and Viterbi decoding. The results are very promising.
Keywords: digital signal processor (DSP), silicon IP, low power, instruction set simulator (ISS)

二、計畫緣由與目的

隨著 IC 製程技術不斷創新，單晶片系(SoC)已是現代電子系統必要的關鍵性組件。用於下一代無線通訊系統的系統晶片，在數位基頻部份將包括一個 RISC 控制器、DSP 處理器核心，特定功能單元、記憶體單元、視訊顯示與網路通訊規約處理單元等。此系統晶片或核心模組的主要設計目標是**低功率**、**高性能**和**低成本**。由於單晶片系統的高複雜度(十~百百萬閘)以及開發時間縮短等因素關係，可再用之矽智產(silicon intellectual property)核心設計技術變成單晶片系統之重要設計考量。下一代無線通訊系統雖然尚在發展中，但基本需求大致上已可以看出：(1) high data rate，(2) sophisticated algorithms，(3) configurable for divergent markets，(4) low power。也就是需要一個高性能的 DSP 處理器來從事通訊、視訊方面所需之各種運算，此 DSP 處理器將以矽智產(IP)的方式與 RISC 控制器和其他模組等整合成一個系統晶片。

DSP 處理器 IP 是 3C 整合產品的重要核心零組件已是眾所周知之事。追求高性能與低功率 DSP 處理器(它們的本質是相互抵觸的)與其新架構提出，仍是許多學術界、產業界努力的研究課題，也是國科會工程處近年來推動的重要研究主題之一。雖然 DSP 處理器和其 Core 已有許多 vendors 存在市場，例如 Texas Instruments (TI)、Analog Devices Inc. (ADI)、Motorola、Agere (Lucent)、DSP Group…等等(詳見 Berkeley Design Technology Inc.; BDTI http://www.bdti.com)。國內產業界已經或正在研發的有華邦、旺宏、創意、智原…等。大學方面也有清華、台大、成大、中正、中山等校投入研究。但本計畫重點在於新架構與新指令集的提出，目標是高性能低功率與可重新組態的特性，因此具有極高之研究挑戰性。

我們的 DSP 處理器核心主要是能支援 DAB 及 DVB-T 基頻運算處理的要求，其重要的特色有以下幾個：(1)高速度：高於 2,000 MOPS 的運算能力(16 位元資料在 200MHz 的工作頻率下)，(2)低功率，低於 1mW/MOP，(3)具可再組、可延展能力(包含了 customizable 的指令集

設計及 configurable 硬體加速器模組)。高速度、低功率是無線通訊基本要求。可重新組態之能力將提供此系可以(1)支援多標準、多工作模態，(2)具有架構台的差異性，(3)實體操作環境的適應性(例如高雜訊環境)。此處理器其他重要性能規格包括：32 位元定點資料，具 SIMD 與次字元平行度的能力, 不同長度指令集，高程式碼密度，並採用.18um CMOS 製程，提供高度的架構延展性等。我們所開發的 DSP 智產核心將是用於下一代無線通訊 SoC 的關鍵模組。

本篇報告將針對今年度計畫工作項目分項敘述與討論。包含：(1)目前新一代 DSP 處理器架構之相關資料的搜集與整理、(2)可支援單一指令多資料(SIMD)之可變長度超長指令字元(variable-length VLIW)的 DSP 處理器架構設計、及(3)此架構之指令集模擬及其應用於數個數位訊號處理核心的實例。

三、研究方法及成果

(1) 數位訊號處理器之設計趨勢 (Trends in DSP Processor Design)

我們搜集了新一代高效能低功率的 DSP 處理器之架構資料，包含了

- C64, C55 series DSP & OMAP from Texas Instruments (TI)
- Blackfin from Intel and Analog Devices Inc. (ADI)
- StarCores from Motorola and Agere (formerly known as Lucent)
- Carmel from Infineon, etc

同時也整理了一些可組態架構及並探討重新組態的機置，

- Improv Jazz platform
- Tensilica Xtensa
- ARC cores
- Triscend E5 & A7, etc

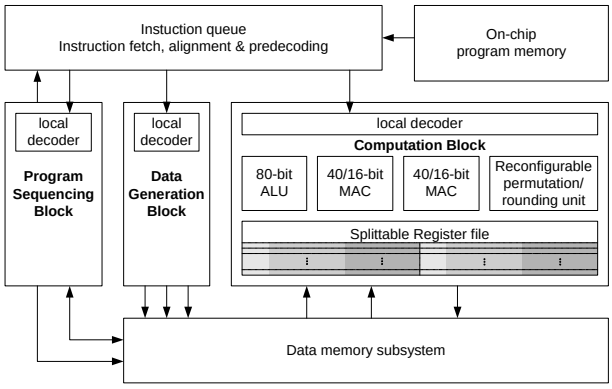
這些設計的技巧及創新的架構將會視情況整合入我們將提出的 DSP 處理器中，並會進行詳細的分析與比較。

(2) 含 SIMD 之可變長度超長指令 DSP 處理器架構 (Variable-length VLIW DSP Processor with SIMD Capability)

我們所提出的 DSP 處理器包括了以下的特性

- 可變長度之超長指令架構(variable-length VLIW;使用多個基本的 16-bit 指令組成)
- 提供 optional 及 user-defined (customized) 的指令空間
- 可分解的(splittable)功能模組， 用來執行 SIMD 之動作，80-bit 的暫存器組可以分割為兩個 40-bit 的累加器(accumulators) 或四個 16-bit 的通用暫存器 (general-purpose registers; GPR)
- 可重新組態並輕易延展
- 提供 power-aware 的指令

其架構圖如下



圖一 DSP 處理器架構圖

(3) 指令集模擬器及效能評估 (Instruction Set Simulator & Performance Evaluation)

我們已經完成指令集的定義(表一)並完成了一指令集模擬器(instruction-set simulator; ISS)。經由此指令集模擬器我們可以輕易地完成此 DSP 處理器的效能評估，以下是我們幾個初步的結果

a. DLMS

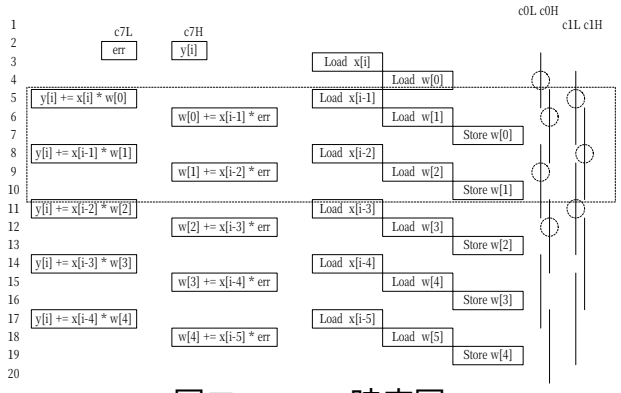
DLMS 演算法中最基本的運算如下

$$y_I = \sum_{j=0}^{T-1} w_j \times x_{I-j}$$

$$e_i = d_i - y_i$$

$$w_j = w_j + 2\mu \times e_i \times x_{i-j},\ j = 0,1,\ldots,T-1$$

圖二是使用我們所提供的 VLIW 指令集所完成的運算時序，每個 tap 平均需要三個指令週期來完成。



圖二 DLMS 時序圖

b. Viterbi 解碼

Viterbi 解碼器中主要的運算即是 add, compare, and select (ACS)，而這個運算恰好可以與我們的 SIMD 資料路徑相戶搭配，達到每個指令週期皆可完成一個 ACS 動作的處理能力，圖三就是我們 DSP 處理器所對應的組合語言。

```
     Scalar unit       AGU              ALU
1    sh r11 r3 8                        addv a0 a1 a2
2    sh r11 r4 10      addia a3 a3 4    minv a2 a3
3    sh r11 r4 12      addia a0 a0 8    subv a0 a1 a2
4    sh r11 r3 14                       minv a2 a5
5    addi r10 r10 8 addia a5 a5 4       addv a0 a1 a2
6    lw r10 r1 0                        minv a2 a3
7    sub r0 r1 r2      addia a0 a0 8    subv a0 a1 a2
8                      addia a3 a3 4    minv a2 a5
9    lw r10 r3 4       addia a5 a5 4    addv a0 a4 a2
10   sub r0 r3 r4      addia a3 a3 4    minv a2 a3
11   sh r11 r1 0       addia a0 a0 8    subv a0 a4 a2
12   sh r11 r2 2                        minv a2 a5
13   sh r11 r2 4       addia a5 a5 4    addv a0 a4 a2
14   sh r11 r1 6       mova a3 13100    minv a2 a3
15                     mova a0 13000    subv a0 a4 a2
16                     mova a5 13016    minv a2 a5
```

圖三 Viterbi 解碼器之核心組語

c. Motion estimation

Mean absolute error (MAE)是最常被用來判斷兩個區塊相似度的量，幾乎所有的 motion estimation 演算法及硬體架構都採用此種運算。它可以表示如下

$$\text{MAE} = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left|a_{m,n} - b_{m,n}\right|$$

式中的 $a_{m,n}$ 與 $b_{m,n}$ 分別表示目前的影像區塊與前一個參考區塊的 pixel。

```
Scalar Unit          AGU              ALU(SIMD)
sub r1 r1 r1     mova a0 10000
                 mova a1 10128
                 mova a2 20000
                 mova a3 30000
addi r1 r1 1                          subv a0 a1 a2
addi r1 r2 -16 addia a0 a0 8          absv a2 a2
bne r2 r0 MAE  addia a1 a1 8          addv a2 a3 a3
addi r1 r1 1                          subv a0 a1 a2
addi r1 r2 -16 addia a0 a0 8          absv a2 a2
bne r2 r0 MAE  addia a1 a1 8          addv a2 a3 a3
```

圖四 MAE 之組語碼

圖四是 motion estimation 的組語碼片段，也就是其核心 MAE 的平行動作，我們的處理器平均使用 0.75 個指令週期就可以完成一個 pixel 所需的 MAE 動作。

四、結論與討論

本計畫已順利完成各項預期工作項目。研究成果正陸續整理投稿於國際會議和期刊中。

五、參考文獻

1. E. A. Lee, “Programmable DSP Architectures, Part I,” *IEEE Acoustics, Speech and Signal Processing Magazine*, October 1988
2. E. A. Lee, “Programmable DSP Architectures, Part II,” *IEEE Acoustics, Speech and Signal Processing Magazine*, January 1989
3. V. K. Madisetti, *VLSI Digital Signal Processors – An Introduction to Rapid Prototyping and Design Synthesis*, IEEE Press, 1995
4. P. Lapsley, et al, *DSP Processor Fundamentals – Architectures and Features*, IEEE Press, 1996
5. J. L. Hennessy, D. A. Patterson, *Computer Architecture – A Quantitative Approach*, 2nd Edition Morgan Kaufmann, 1996
6. I. Kuroda, T. Nishitani, “Multimedia Processors,” *Proceedings of the IEEE*, June 1998
7. Hoon Choi, et al, “Synthesis of Application Specific Instructions for Embedded DSP Software,” *IEEE Transactions on Computers*, June 1999
8. Steve Furber, *ARM System-on-Chip Architecture*, Addison Wesley, 2000
9. Jennifer Eyre, Jeff Bier, “The Evolution of DSP Processors – From Early Architectures to the Latest Developments,” *IEEE Signal Processing Magazine*, March 2000
10. Margarida F. Jacome, Gustavo de Veciana, “Design Challenges for New Application Specific Processors,” *IEEE Design & Test of Computers*, April-June 2000
11. Alex Peleg, Uri Weiser, “MMX Technology Extension to the Intel Architecture,” *IEEE Micro*, August 1996
12. R. B. Lee, “Subword Parallelism with MAX-2,” *IEEE Micro*, August 1996
13. Andre DeHon, “Reconfigurable Architectures for General Purpose Computing,” *MIT A. I. Technical Report No. 1586*, October 1996
14. *Application Specific Microprocessor Solutions – Datasheet for Xtensa V1*, Tensilica, 1998
15. *Configurable Processors – An Emerging Solution for Embedded System Design*, Triscend, 1998
16. T. Fujii, et al, “A Dynamically Reconfigurable Logic Engine with a Multi-Context / Multi-Mode Unified-Cell Architecture,” *International Solid State Circuits Conference (ISSCC‘99)*, 1999
17. J. T. J. van Eijndhoven, et al, “TriMedia CPU64 Architecture,” *International Conference on Computer Design (ICCD‘99)*, 1999

表一 指令集

Program Sequencing and Scalar Operations	
No operation	`nop`
Jump	`jump label`
Jump and link	`jal label`
Jump register	`jr r0`
Branch equal	`beq r0 r1 label`
Branch not equal	`bne r0 r1 label`
Load byte or half word or word	`lb(lh、lw) r0 r1 offset`
Store byte or half word or word	`sb(sh、sw) r0 r1 offset`
Add or subtract two operands	`add(sub) r0 r1 r2`
Add immediate value	`addi r0 r1 imm`
Multiply two operations	`mult r0 r1`
Move multiplication result high part to register	`mfhi r1`
Move multiplication result low part to register	`mflo r1`
Various logical operations	`and(or、xor) r0 r1 r2`
Logical shift right or left	`sll(srl) r0 r1 shamt`
Arithmetic shift right	`sra r0 r1`
Set less than	`slt r0 r1 r2`
Set less than immediate	`slti r0 r1 r2`
End of program	`end`
ALU Operations	
Absolute value with SIMD	`absv a0 a1`
Add or subtract two memory data with SIMD	`addv(subv) a0 a1 a2`
Maximum or minimum with SIMD	`maxv(minv) a0 a1 a2`
Variable logical operations with SIMD	`andv(orv、xorv) a0 a1 a2`
Load 32-bit value from memory	`l32v a0 c0H`
Load 16-bit value from memory	`l16v a0 c00`
Store 40-bit reg to 32 or 16-bit mem with rounding	`sr32v(sr16v) a0 c0H`
MAC Operations	
Multiply-accumulate	`macv c0H c1H c2H`
Multiply-accumulate unsigned	`macuv c0H c1H c2H`
Permutation Operations	
Permute four memory value in any order	`Perm a0 1230`
AGU Operations	
Add two address register	`adda a0 a1 a2`
Add constant value to address register	`addia a0 a1 offset`
Move constant value to address register	`mova a0 offset`

18. R. Tessier, W. Burleson, “Reconfigurable Computing for Digital Signal Processing – A Survey,” *Journal of VLSI Signal Processing*, 2000
19. Hui Zhang, et al, “A 1V Heterogeneous Reconfigurable Processor IC for Baseband Wireless Applications,” *International Symposium on Solid-State Circuits (ISSCC‘00)*, 2000
20. T. J. Callahan, et al, “The Garp Architecture and C Compiler,” *IEEE Computers*, April 2000
21. H. Singh, et al, “MorphoSys: an integrated reconfigurable system for data-parallel and computation-intensive applications,” *IEEE Transactions on Computers*, May 2000
22. R. B. Lee, “Subword Permutation Instructions for Two-dimensional Multimedia Processing in MicroSIMD Architectures,” *International Conference on Application-Specific Systems, Architectures and Processors*, 2000
23. Yoshihisa Kondo, et al, “A 4GOPS 3Way-VLIW Image Recognition Processor based on a Configurable Media Processor,” *International Symposium on Solid-State Circuits (ISSCC’01)*, 2001
24. T. J. Lin and C. W. Jen, "An Efficient 2-D DWT Architecture via Resource Cycling," *IEEE International Symposium on Circuits and Systems (ISCAS’01)*, May 2001
25. T. J. Lin and C. W. Jen, “CASCADE – Configurable and Scalable DSP Environment,” *International Conference on Circuits and Systems (ISCAS’02)*, May 2002
26. Jan M. Rabaey, *Low Power Design Methodologies*, Academic Publishers, 1996
27. Thomas D. Burd, Robert W. Brodersen, “Processor Design for Portable Systems,” *Journal of VLSI Signal Processing Systems*, 1996
28. Krste Asanovic, “Energy-Exposed Instruction Set Architectures,” *International Symposium on High Performance Computer Architecture (HPCA’00)*, 2000
29. Thomas D. Burd, et al, “A Dynamic Voltage Scaled Microprocessor System,” *International Symposium on Solid-State Circuits (ISSCC’00)*, 2000
30. M. Sami, et al, “Instruction-Level Power Estimation for Embedded VLIW Cores,” *International Workshop on Hardware/Software Codesign (CODES’00)*, 2000
31. L. Benini, G. Micheli, “System-Level Power Optimization: Techniques and Tools,” *ACM Transactions on Design Automation of Electronic Systems*, April 2000
32. M. Lewis, L. Brackenbury, “An Instruction Buffer for a Low Power DSP,” *International Symposium on Advanced Research in Asynchronous Circuits and Systems (ASYNC’00)*, 2000
33. W. Burleson, et al, “Dynamically Parameterized Architectures for Power-Aware Video Coding – Motion Estimation and DCT,” *International Workshop on Digital and Computational Video (DCV’01)*, 2001