

輔仁學誌—理工類

中華民國九十五年十二月

第四十期

目 錄

頁次

可排程 DMA 控制器之設計與實作	林寬仁 黃莊翔	1
以 64 位元北歐武夫電腦叢集進行第一原理計算	林銘杰 黃國華	17
拿破崙紙積木之研究	呂克明 曹子殷 陳美汎	39
使用四端主動電流傳輸器合成主動電感模擬電路	鄧永昌	71
α 混合條件下之核密度估計量的均勻一致性	洪萬吉	81
小波方法應用於神經元動作電位波形分類之正確度探討	李 元 蔡孟利 單維彰 嚴健彰	95
94 學年度理工學院專任教師對外發表之論文摘要		109

FU JEN STUDIES

SCIENCE AND ENGINEERING

NO. 40, dec. 2006

CONTENTS

	Page
Design and Implementation of a Schedulable DMA Controller by Kuan Jen Lin* and Chuang Hsiang Huang	1
First-Principles Computation Based on a 64-bit Beowulf PC Cluster..... by Ming-Chieh Lin* and Kuo-Hua Huang	17
A Study of Napoleon Paper Building Blocks by Keh-Ming Lu and Alan S. Tsaur and Mei-Fen Chen	39
Active simulation of grounded inductor using CFCCIIs by Yung-Chang Yin and Yung-Chwan Yin	71
Uniform Consistency of Kernel Density Estimator under α -mixing by Wann-Jyi Horng	81
Accuracy of Wavelet-based Spike Classifier..... by Yuan Li and Meng-Li Tsai and Wei-Chang Shann and Chien-Chang Yen	95
Abstracts of Papers by Faculty Members of the College of Science and Engineering that Appeared in the 2005~2006 Academic Year	109

Design and Implementation of a Schedulable DMA Controller

Kuan Jen Lin* and Chuang Hsiang Huang

Department of Electronic Engineering, Fu Jen Catholic University, Taiwan

ABSTRACT

A traditional DMA scheme hardly guarantees I/O throughput for real time multimedia applications. In this paper, we propose an SDMA (Schedulable DMA) scheme that guarantees real time I/O throughput and interrupts the CPU as few as possible. Furthermore, it can accept unscheduled requests and provide facility to treat unexpected delay to access I/O or memory. The controller handling the SDMA was designed and successfully implemented on an AMBA-based SOPC platform. The experimental results show the efficiency and effectiveness of the proposed SDMA design.

Key Words: DMA, Real time system, FPGA, AMBA, I/O Devices

1. INTRODUCTION

Most real time scheduling algorithms demand that the worst-case execution time of each task is known in advance. This is hardly satisfied if a task uses a DMA (Direct Memory Access) I/O method to transfer data between I/O devices and memories [11, 4]. Figure 1 shows a typical real time multimedia system that has four I/O channels each providing video or audio stream at a specified rate. To ensure real time application like playback, the computing system requires

*Corresponding author. Tel: +886-2-29032159

E-mail: kjlin@mails.fju.edu.tw

guaranteed I/O throughput. However, a traditional DMA mechanism neither friendly nor efficiently supports such real time tasks. The reasons are described as follows [6]:

- (1) The bus bandwidth allocated to DMAC is not guaranteed: Traditional cycle-stealing DMA I/O tasks proceed by stealing bus cycles from the CPU. How many the bus bandwidth a DMAC can get depends on the program instructions. It is not easy to precisely estimate.
- (2) The DMA service time allocated to an I/O device is not guaranteed: A traditional DMA controller, like IC 8237, serves I/O requests with first-in-first-serving strategy. When more than one I/O device share the DMA service, if the DAM action is activated only by I/O requests, the controller does not ensure how long the service time is allocated to an I/O device.
- (3) Frequent interrupts decrease the CPU's efficiency: If DMA tasks switch frequently, the overhead of setting up DMA transfers and executing interrupt services can heavily disturb the normal CPU tasks. This is not wanted in real time systems.

Several DMA controllers were proposed to support a real time system. Back to 1988, Sprunt et al. proposed a Preemptable I/O Controller (PIOC) to avoid priority inversion [12]. The commercial product TMS320C621 DSP contains an Enhanced DMA (EDMA) controller [16], which prioritizes transfer requests and prefers serving higher-priority requests. Furthermore, the EDMA uses RAM to store transfer parameters and allows a new channel parameters immediately loaded via a linking mechanism. This saves the time to set up transfer parameters. Srinivasam [13] proposed a PDMA (Pre-programmed DMA mechanism) that allows a DMA action to continue moving data even that the source or destination addresses are not consecutive. Although the PDMA can be used to execute a task chain according to a predetermined schedule, it cannot accept any unscheduled request and does not provide facility to circumvent unexpected delays to access I/O or memory. The work in [4] presented algorithms to bound DMA I/O tasks to meet real time requirements. However, it did not address the design of DMA controller. Previous works addressed the hardware support for real time OS [1, 5, 8, 9], but did not consider the use of DMA to guarantee I/O throughput.

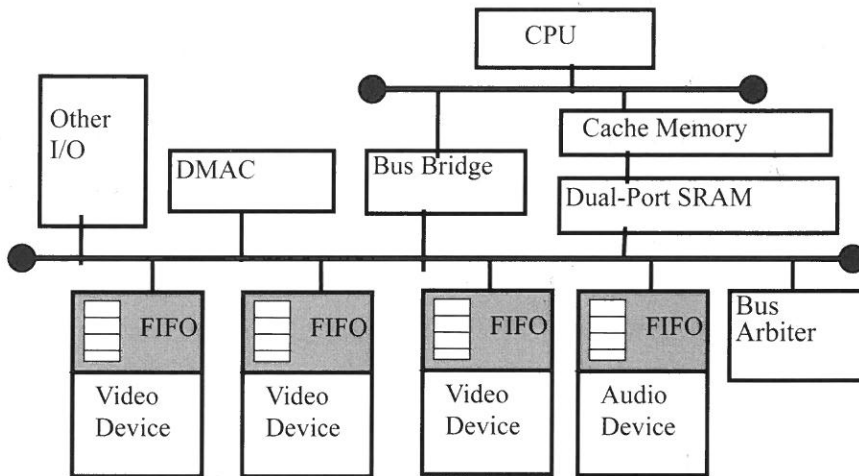


Figure1: A Typical multimedia embedded system using DMA.

To remedy the shortcoming of the traditional DMA for real time system, we propose an SDMA (Schedulable DMA) mechanism that guarantees real time I/O throughput and interrupts the CPU as few as possible. Furthermore, it can accept unscheduled requests and provide facility to treat unexpected delay to access I/O and memory. The controller handling the SDMA was designed and successfully implemented on an AMBA-based SOPC platform. The experimental results show the efficiency and effectiveness of the proposed SDMA design.

The next section formulates the I/O task model and investigates the schedulability for a set of tasks. Section 3 describes the design of the SDMA controller. Implementation of the controller on an ARM9-based SOPC platform is shown in Section 4. The final section draws conclusions.

2. I/O TASK MODEL AND SCHEDULE

The I/O device under our consideration produces data stream at a specified rate. We assume that each device uses a FIFO as a buffer to store the data from peripheral devices, as shown in Fig. 1. In order not to miss any data, the DMAC must transfer an appropriate amount of

the data from the FIFO to memory within a period. Assume that for the device D_i , the requested rate is R_i and the size of FIFO is B_i . Thus, we can consider such a rate-based IO data-transfer task as a periodic job, τ_i , denoted as (P_i, N_i) , where $P_i = B_i / (2 \times R_i)$ is the period of τ_i and N_i is the number of data word to be transferred within the period. The length of P_i is equal to the time required to put half of B_i words into the FIFO by the I/O device. To avoid overflows or underflows in the initial stage, we assume that (1) when the SDMAC starts running I/O task schedule, the FIFO has stored exactly $B_i/2$ words of data (null tokens are inserted if data are not enough); (2) N_i is set to be $B_i/2$.

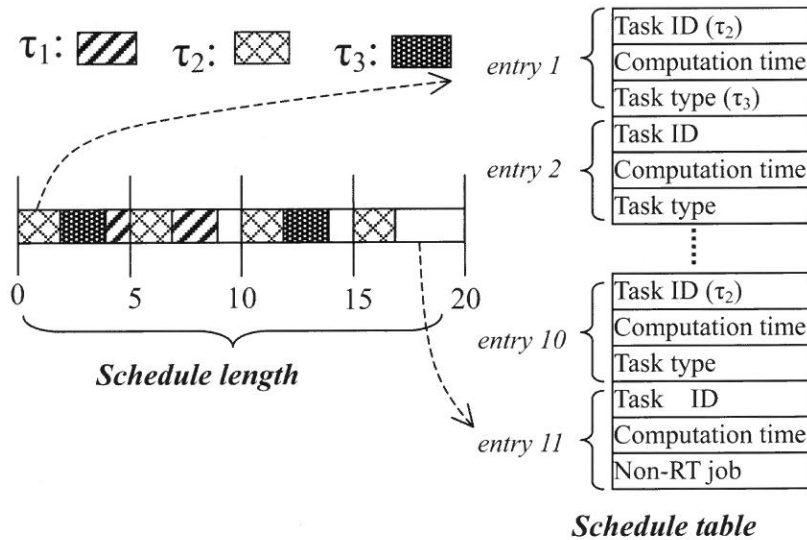
We assume all the I/O devices have the same transfer time T for moving a data word from FIFO to memory. Then the real time rate-based I/O task τ_i can be denoted as (C_i, P_i) , where $C_i = T \times N_i$ is the total transfer time required for task τ_i . The C_i is analogous to the computation time considered in real time scheduling for CPUs. Therefore, well-known real time scheduling algorithms such as RMS and EDF [2, 7] can be directly applied to schedule I/O tasks to run on a DMAC. The proposed work currently adopts RM algorithm that assigns tasks with shorter periods to have higher priorities. Figure 2 shows the pseudo code that performs the scheduling and constructs the table. Figure 3 shows an example of a RM schedule on a set of three rate-based I/O tasks for which the deadline is equal to the period: $\tau_1(3, 20)$, $\tau_2(2, 5)$ and $\tau_3(2, 10)$. It also shows the schedule table for a schedule length, which is the least common multiple (LCM) of all P_i , i.e. 20 in this example. The duration not assigned for any job is reserved for non-RT jobs from other DMA requests. These requests are considered as aperiodic jobs and the reservation for them can be considered as running a background scheduler [3]. The whole schedule is represented by a schedule table, in which each entry represents a RT job or a non-RT reservation. In this example, entries 6, 9 and 11 represent the non-RT jobs. The information in an entry includes the task ID the job belongs to, computation time (corresponding to the number of data-transfer) and the task type (RT or non-RT). The information will be converted to parameters to set up DMA data transfer.

```

1.   $n$ : the number of I/O devices
2.  for  $i \leftarrow 1, 2, 3, \dots, n$  do
3.       $P_i \leftarrow B_i / (2 \times R_i)$ 
4.       $N_i \leftarrow B_i / 2$ 
5.       $C_i \leftarrow T \times N_i$ 
6.  end for
7.   $\text{Schedule length} \leftarrow \text{L.C.M.}(P_1, P_2, P_3, \dots, P_n)$ 
8.  do RMS ( $P_1, P_2, P_3, \dots, P_n, C_1, C_2, C_3, \dots, C_n$ )
9.  until schedule length is reached
10.  $m$ : the number of RT jobs and non-RT reservations
11. for  $j \leftarrow 1, 2, 3, \dots, m$  do
12.     construct an entry in the schedule table
13. end for

```

Figure2: The Algorithm constructing schedule tables.

Figure 3: A feasible RM schedule for a task set: $\tau_1(3, 20)$, $\tau_2(2, 5)$ and $\tau_3(2, 10)$, and its corresponding schedule table

Schedulability

A schedule is feasible if every job in the schedule completes by its deadline. A sufficient condition to ensure the RM schedulability for n tasks has been derived [7] as follows:

$$U = \sum_{i=1}^n \frac{C_i}{P_i} \leq n(2^{1/n} - 1)$$

In this work, U is bus bandwidth required by the DMAC (analogous to the CPU utilization). If the maximal bus bandwidth allocated to the DMAC is fixed, whether a feasible schedule exists can be determined by the summation of all R_i . This is derived as follows:

$$\sum_{i=1}^n \frac{C_i}{P_i} = \sum_{i=1}^n \frac{TN_i}{\left(\frac{B_i}{2R_i}\right)} = \sum_{i=1}^n \frac{TB_i R_i}{B_i} = T \sum_{i=1}^n R_i$$

Let L be the given upper bound of bus bandwidth allocated to the DMAC, and a sufficient condition to ensure a feasible schedule is

$$\sum_{i=1}^n R_i \leq (L/T) \quad \text{and} \quad L \leq n(2^{1/n} - 1)$$

The size of FIFO also affects the schedulability. Let s be the smallest time unit of the schedule length. So, P_i and C_i must be a multiple of s . We have known $P_i = B_i / 2 R_i$ and $C_i = N_i \times T = (B_i / 2) \times T$. Let

$$(1) P_i = k_1 \times s \text{ and}$$

$$(2) C_i = k_2 \times s, \text{ where } k_1 \text{ and } k_2 \text{ are nonzero integers.}$$

$$\text{From (1), we have } B_i = 2 R_i \times s \times k_1 \dots (3)$$

$$\text{From (2), we have } (B_i / 2) = (s/T) \times k_2 \dots (4)$$

Then, from (3) and (4) we have $(R_i T) = (k_2 / k_1)$. Note that k_1 and k_2 must be integers. The minimal pair (k_1, k_2) decides the minimal B_i to ensure the schedulability.

3. SDMA CONTROLLER

The proposed SDMA scheme is coordinated by an SDMA controller that can automatically repeatedly performs DMA tasks for a real time schedule as well as it can serve other non RT DMA request when it does not run a RT job. Figure 4 shows its block diagram. The design is a modified version on an open core from [10], which is a traditional AMBA-compatible DMAC. A typical DMA action consists of two steps: (1) the DMAC loads a datum word from the source address into the buffer inside the DMAC and then (2) write the datum from the buffer into the destination. In the original design, the two interfaces both are AMBA-compatible. In our design,

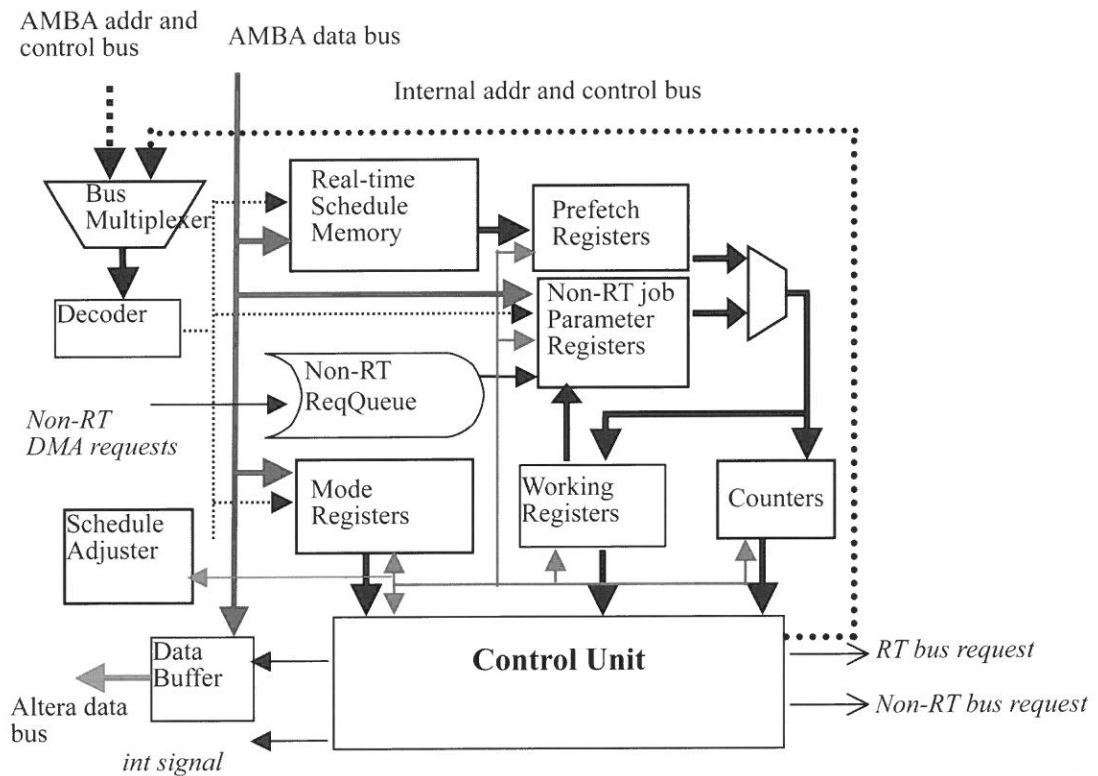


Figure 4: Block diagram of the SMDA controller.

the connection to dual-port memories, used as the destination, is modified to Altera-compatible memory interface. The AMBA-compatible interface is used to connect the FIFO, used as the source in our design. Furthermore, it also connects the DMAC to the system bus, through that the CPU can program the DMAC. Other new design features proposed in this paper are described as follows:

RT schedule table

The memory contains the schedule table. Each entry contains the transfer parameters such as the source address, destination address, data-transfer count and task type. The table is designed to contain up to 64 entries.

Non-RT DMAREQ

Besides performing the jobs in the RT schedule table, the DMAC can also serve other DMA requests. These requests are performed only during the time reserved for non-RT jobs. Hence, if they occur when a RT job is running, they will be kept in the waiting queue.

Prioritized bus request

The DMAC provides two kinds of bus requests to the bus arbiter: RT and non-RT bus-requests. The RT bus request is issued when it run a RT job. The bus arbiter grants the bus to the DMAC at once. For other DMA services, the DMAC issues non-RT requests and the arbiter will serve them according to FIFO strategy.

Zero Switch Time

Each time the DMAC starts a job, the control unit needs to load transfer parameters from the schedule table or non-RT DMA transfer parameter registers. In order to have zero switching time, the DMAC prefetches transfer parameters of the next RT job into the prefetch registers. When the current job is completed, it loads the transfer parameters directly from the prefetch registers into the working registers.

Schedule Adjustment

The schedule is derived based on pre-determined timing model. Namely, the transfer time is assumed to be fixed and known in advance. However, unexpected delays may occur when

accessing I/O devices or dual-port memory. The adjustment timer records the delays and automatically adjusts the time length reserved for non-RT jobs. To see the case in Fig. 5, the schedule in Fig. 5(a) is originally derived. In Fig. 5(b), unexpected delays are inserted into the schedule. If the job time is not adjusted, the schedule must be delayed. This will cause some RT-job to miss deadline. The proposed adjustment scheme decreases the time left for non-RT DMA services and retain the schedule for RT jobs, as shown in Fig. 5(c). If the delays are too large such that even when no time left for non-RT jobs, some RT job still misses the deadline, the DMAC will issue an interrupt to the CPU.

Control Unit

Before accepting any DMA request, the schedule table should be already prepared and loaded into the controller. Once the SDMAC is enabled, the first entry information is loaded into the *working registers* and the corresponding job is executed. At the same time, the next entry information is prefetched into the *prefetch registers*. When a RT-job is completed, the SDMA loads the parameters in the prefetch registers into the working registers. Again, it also loads the information of the next job into the prefetch registers. If the job is a non-RT-job, the SDMA will check if any non-RT request is pending. If some non-RT request is pending, it loads the job's information into the working registers from the *Non-RT parameter registers*. Otherwise, it suspends itself until the time reserved to non-RT jobs is over. During the SDMAC executing a non-RT DMA job, the reserved time may be over. In this case, the SDMAC suspends the non-RT job and loads its information back to the parameter registers. When the next duration reserved for non-RT jobs begins, it is resumed immediately. The job order is cyclic, so the first job will follow the last one.

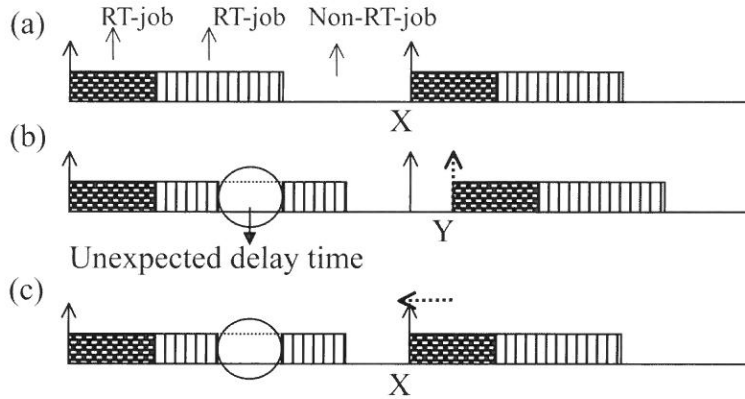


Figure 5: Schedule adjustment.

4. IMPLEMENTATION

We implemented the design prototype of the SDMA scheme on the Altera EPXA10 DDR development board [14], as shown in Fig. 6. The board features an Excalibur EPXA10 SOPC (System On Programmable Chip) device that contains an ARM922T 32-bit RISC processor core, 256K single-port SRAM, 128K dual-port SRAM, and a variety of peripherals such as interrupt controller and UART. Figure 7 shows its system architecture. Our own designs including the DMAC and bus arbiter are placed in the PLD part.

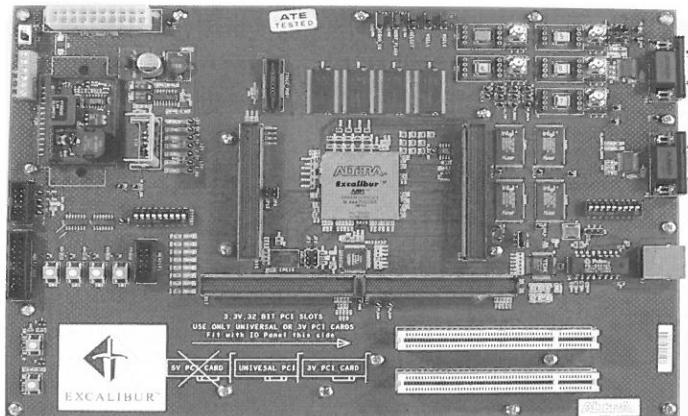


Figure 6: The Altera EPXA10 DDR EVB.

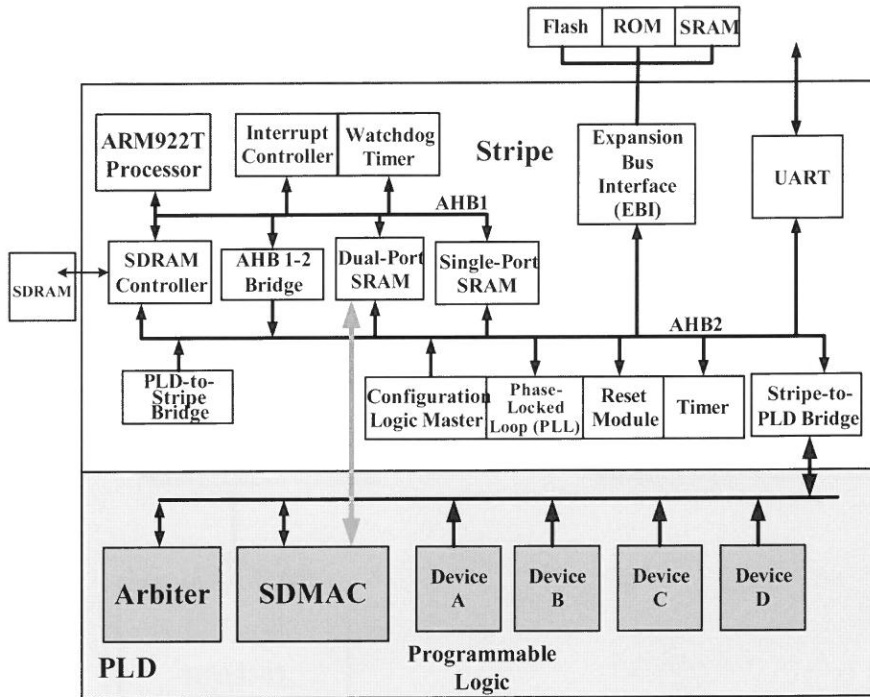


Figure 7: Excalibur device system architecture.

It is not trivial to verify a design that has an AMBA bus interface [15]. Altera corporation provides an AMBA bus functional model for the EPXA10 device and models for PLD-to-stripe and stripe-to-PLD bridges. The design overall flow used in this project is shown in Fig. 8. The *MegaWizard* Plug-in Manager is a graphical user interface utility, which provides the designers the facility to set the operational parameters for the embedded stripe on Excalibur device. After setting the operational parameters, the configured stripe instance is generated. The top level design combines the configured stripe instance and Verilog codes for the SDMAC. Then, with the bus model for AMBA, we can use *Modelsim* to simulate the SDMAC. After the verification by simulation, the design is synthesized and mapped to Excalibur device.

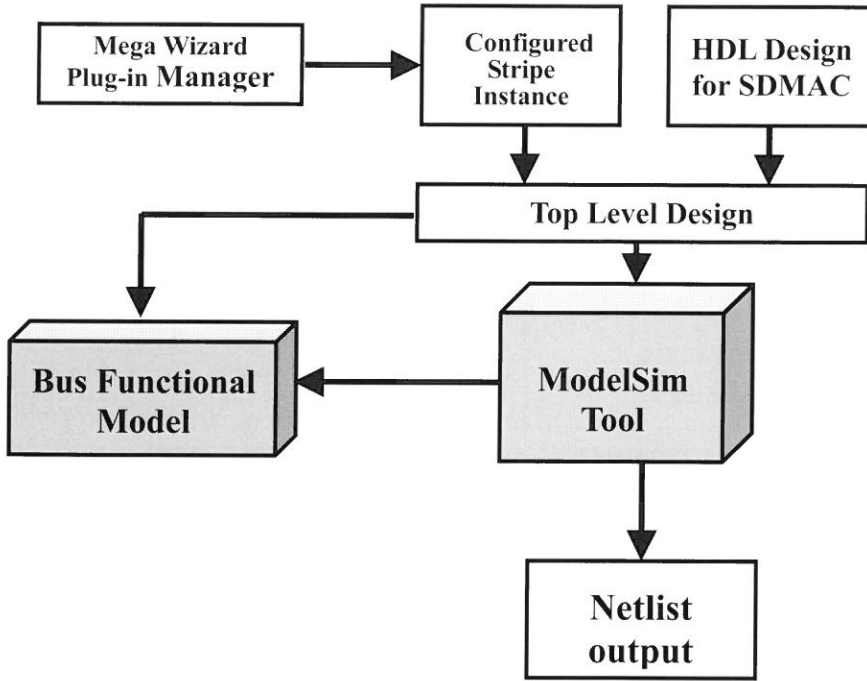


Figure 8: Design flow.

The SDMAC was designed in Verilog HDL and successfully synthesized into the gate-level circuit. The control unit of the SDMAC occupies 386 logic elements. Currently, it runs at 25 MHz and takes 9 clocks to complete a data transfer (from IO to memory). The performance of the SDMA is compared to that of the traditional DMA run by software-control. Assume that there are four video capture devices that are required to capture 2, 4, 8 and 12 frames per second respectively. All devices have the same video format RGB32 and resolution 320x240 pixels per frame. According to the frame rate, their corresponding R_i 's are 153600, 307200, 614400 and 921600 words, respectively. Their FIFO sizes are also assumed to be the same. This means that all devices have the same C_i . We derive the task schedule according to the RM algorithm. In the traditional DMA scheme, we use the software-enabled mode to execute the task schedule and use interrupts to switch tasks. Each interrupt has an overhead penalty. The overhead corre-

sponds to the summation of the interrupt latency (IL) and the execution time of ISR (T_{ISR}). For the EPXA10, the IL takes 20 clocks. The CPU runs at 100 MHz in the experiment. We assume that the ISR only executes instructions to set DMA's parameters. It needs 11 PLD clocks running at 25MHz. Various FIFO sizes were evaluated. As shown in Figure 9, the case with 30-word FIFO size has the best improvement, in that the execution time falls by 15%.

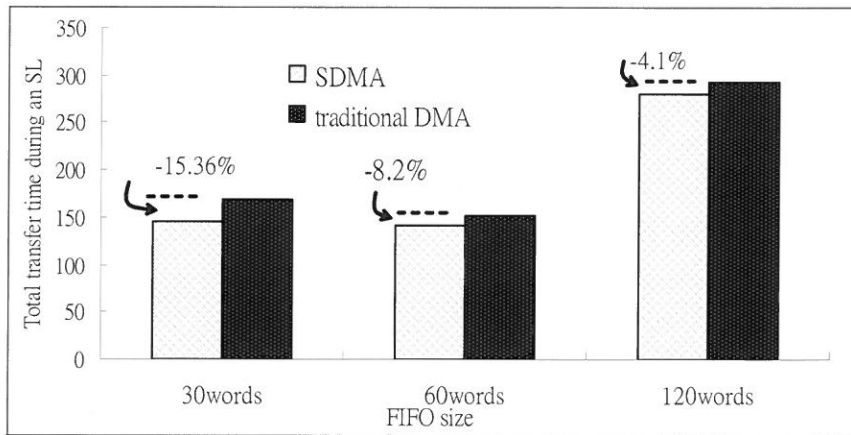


Figure 9: Performance comparison under various FIFO sizes.

5. CONCLUSION

The traditional DMA mechanism hardly satisfies I/O requests with time constraints. In this paper, we have proposed an SDMA (Schedulable DMA) mechanism that guarantees real time I/O throughput and interrupts the CPU as few as possible. Furthermore, it can accept unscheduled requests and provide facility to treat unexpected delay to access I/O or memory. The controller handling the SDMA was designed and successfully implemented on an AMBA-based SOPC platform. The experimental result has shown the efficiency and effectiveness of the proposed SDMA design.

6. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their comments to improve the quality of this paper.

7. REFERENCES

- [1] B. S. Akgul, V. J. Mooney, H. Thane, and P. Kuacharoen, "Hardware Support for Priority Inheritance," *Proceedings of the 24th IEEE International Real-Time Systems Symposium*, pp. 246-255, 2003.
- [2] F. Balarin, L. Lavagno, P. Murthy, and A. Sangiovanni-Vincentelli, "Scheduling for Embedded Real-Time Systems," *IEEE Design & Test of Computer*, pp. 71-82, January-March 1998.
- [3] F. Cottet, J. Delacroix, and Z. Mammeri, "Scheduling in Real-Time Systems," JOHN WILEY & SONS, LTD, 2002.
- [4] T. Y. Huang, J. W.-S. Liu, and D. Hull, "A Method for Bounding the Effect of DMA I/O Interference on Program Execution Time," *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, pp. 275-285, December 1996.
- [5] P. Kohout, B. Ganesh, and B. Jacob, "Hardware Support for Real-time Operating Systems," *IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, pp. 45-51, Oct. 2003.
- [6] K. J. Lin, C. H. Huang, and C. C. Lo, "Design and Implementation of a Schedulable DMAC on an AMBA-Based SOPC Platform," *IEEE APCCAS*, pp. 279-282, 2006.
- [7] C. L. Liu and J. W. Layland, "Scheduling Algorithms for Multi-Programming in a Hard-Real-Time Environment," *Journal of the Association for Computing Machinery*, vol. 20, no. 1, pp. 46-61, 1973.
- [8] R. Lysecky and F. vahid, "Prefetching for Improved Bus Wrapper Performance in Cores," *ACM Transactions on Design of Electronic Systems*, vol. 7, no. 1, pp. 58-90, January 2002.
- [9] V. J. Mooney and D. M. Blough, "A Hardware-Software Real-Time Operating System Framework for SOCs," *IEEE Design and Test of Computers*, pp. 44-51, November-December

2002.

- [10]G. Morris, "DMA Controller for AMBA Bus IP Core," Final Report, Version 1.00, <http://www.preromanbritain.com/gwem/project/>.
- [11]A. L. N. Reddy and J. C. Wyllie, "I/O Issues in a Multimedia System," *IEEE Computer*, vol. 27, no. 3, pp. 69-74, March 1994.
- [12]B. Sprunt, D. Kirk, and L. Sha, "Priority-Driven, Preemptive I/O Controllers for Real-Time Systems," *Proceedings of the 15th Annual International Symposium on Computer architecture*, pp. 152-159, 1988.
- [13]S. Srinivasan and D. B. Stewart, "High Speed Hardware-Assisted Real-Time Interprocess Communication for Embedded Microcontrollers," *Proceedings of the 21st Real-Time Systems Symposium*, pp. 269-279, 2000.
- [14]ALTERA, "EPXA10 DDR Development Board Hardware Reference Manual," 2003.
- [15]ARM Ltd. "AMBA Bus Specification, Revision 2.0".
- [16]Texas Instrument, "TMS320C6000 Peripherals Reference Guide".

received October 27,2006

revised November3 25,2006

accepted December 1, 2006

可排程 DMA 控制器之設計與實作

林寬仁* 黃莊翔

輔仁大學電子工程系

*E-mail: kjlin@mails.fju.edu.tw

*Tel: 886-2-29032159

中文摘要

確保 I/O 資料傳輸的速率，係即時多媒體應用成功與否的關鍵議題。使用傳統 DMA(記憶體直接存取)機制在 I/O 與記憶體間搬動資料，很難保證符合即時系統之要求。本論文提出一種可排程之 DMA 機制 (SDMA)，其可確保 I/O 資料傳輸的速率，並盡量減少中斷 CPU 之次數。此外它還能接受非即時之工作與處理預期外之工作延遲。我們研討了 I/O 裝置之工作模型，並推導其可排程之條件。我們設計此一 SDMA 機制之控制器，並在一個 ARM-based 的晶片系統發展平台上，成功實作此一設計。實驗結果彰顯示我們所提之 SDMA 機制之效能與極高之應用性。

關鍵詞：記憶體直接存取、即時系統、輸出入裝置、AMBA、FPGA

First-Principles Computation Based on a 64-bit Beowulf PC Cluster

Ming-Chieh Lin* and Kuo-Hua Huang

NanoScience Simulation Laboratory

Department of Physics

Fu Jen Catholic University

Taipei County 24205, Taiwan, R.O.C.

Abstract

In recent years, numerical simulations play more and more an important role in nano-science research. For the first-principles computations, we usually spend a lot of time on the computer calculation, maybe several hours, a few days, or even several weeks. How can we reduce the calculation time? Low-cost and high-performance Linux clusters are the best solutions for increasing computation performance. Parallel computing is highly required in performing these huge calculations.

Due to the well development of computer industry, the rapid improvement of network technology, and lower costs, PC clusters are widely used in scientific computing. This paper mainly introduces the homemade 64-bit Beowulf PC cluster built in the NanoScience Simulation Laboratory (NSSL), at Physics Department of Fu Jen Catholic University. Not only the experience in construction and performance testing of the PC cluster, but also the installation of first-principles computation packages and the corresponding performance benchmark are introduced.

Key Words: PC cluster; Beowulf; AMD; 64-bit; first-principles

* Corresponding author. Tel: +886-2-29052585; Fax: +886-2-29021038
E-mail: mclin@mail.fju.edu.tw

1. Introduction

In recent years, nanoscience and nanotechnology almost become the term of high-technology. The word "nanotechnology" originated from the unit "nanometer". It's the unit of length equal to 10^{-9} meter, thinner than 1/100,000 of a human's hair. Microorganisms such as virus, single molecules, and microchips are all operated on the scale of a few nanometers. There are wide applications of nanotechnology, from fundamentals to applied sciences, including physics, chemistry, material, photoelectronics, biotechnology, and medicine, etc. Because nano-materials have micro-structures, the classical theory can not apply. The quantum effects cannot be ignored, the proportion accounted of surface area is heightened, and the material will appear back to different physics, chemistry and biological nature to huge view yardstick. To find out the characteristics of nano-materials, such as electronic structures, optical properties, temperature effects, magnetic characteristics, and mechanical properties, the discovery of quantum mechanics provides us an opportunity to probe into the questions mentioned above in terms of microscopic behaviors, and study their properties using the first-principles computation.

In nanoscience research, due to the well development of the computer science and technology, the researchers gradually rely on the numerical simulations. As the simulation models become huge, we may meet some problems such as memory insufficient, computing speed too slow, and too much time consuming in the simulations. To solve these predicaments, first, we can buy a high-performance supercomputer, but it does need massive funds! Or one can apply for an account from the National Center for High-performance Computing (NCHC) [1]. The PC clusters can be another solution instead of supercomputers. The PC clusters may satisfy the requirement of high-performance computation and save the massive funds. In addition, we don't have to wait for the queue.

Generally, Beowulf clustered systems may include many PCs or workstations connected by network for parallel computing. The "Beowulf" means that the systems generally use existing commercialized hardware components on the market, including relevant products, such as personal computers and network devices [2,3]. All components, including software and hardware, are not designed specially or only produced by some specific manufacturers or some specific products. All of them can be easy to obtain in the general computer sales fields. More accurate speaking, a Beowulf is a cluster of mass-market commodity off the shelf PCs intercon-

nected by low cost local area network (LAN) technology running an open source code Unix-like operating system and executing parallel applications programmed with an industry standard message passing model and library. Beowulf-class computers are simply the best price/performance systems available today for many high-end applications [4].

The cluster we studied is a Beowulf. The strict definition is the first kind Beowulf. The hardware is cheap and easy to obtain. The development of PC cluster also admires the selfless contributions of freeware groups. All software such as operating systems, each kind of servers, compilers, and message passing tools all may be free or cost low price to obtain from the network. The parallel computing systems consisting of a small number of PCs are more popular at present. In addition to its low cost, it is easier to maintain and upgrade. Although the computation ability can not be competitive with large-scale clusters, but this kind of small-scale PC cluster suits the general small or middle scale laboratory to carry on some medium-scale computations, such as first-principles computation, plasma simulation, and fluid dynamics, etc.

2. System Description

For the cluster developed in the NanoScience Simulation Laboratory (NSSL), at Physics Department of Fu Jen Catholic University, we had chosen PCs as the basic platform. Why did we choose PCs? Part of the popularity comes from the fact that the systems can be self-made with reasonable effort, most often using standard desktop PCs and Ethernet networks. The self-made systems are often tailored for the specific use and constructed with a small budget. The self-made systems also have several advantages: lower maintenance cost, best performance to price ratio, quickly updated, and easier to maintain and upgrade.

In order to performing the first-principles computation and plasma simulation, we made a plan to build a PC cluster at the end of year 2003, and purchased the relevant hardware components in 2004. We finished the installation and started testing in the summer of 2004. We also launched the relevant standard benchmark to compare the performance of our cluster with that of other platforms. Figure 1 shows the network structure of NSSL PC cluster. Figure 2 is the photography of the cluster we have built. The system is truly self-made, as each individual PC was constructed from respective PC components.

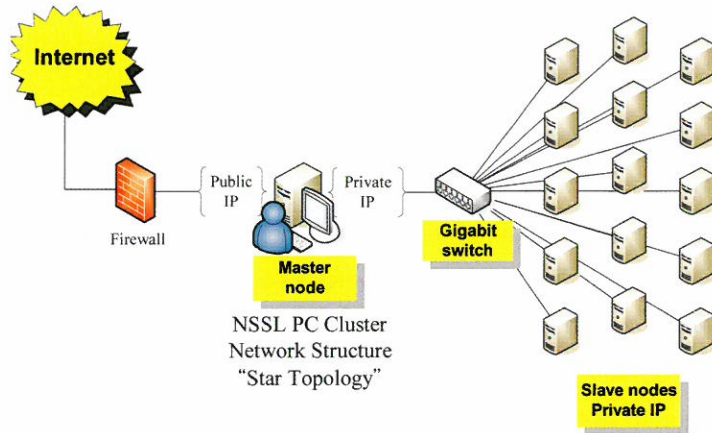


Fig. 1. Generic diagram of the network structure of NSSL PC cluster.



Fig. 2. Photograph of NSSL PC cluster.

The NSSL PC cluster consists of 16 nodes. Each computing node was equipped with one AMD® Athlon 64® 2900+ processor running at 1.8 GHz with 1024 KB L2 cache, 1.5 GB of PC3200 DDR400 SDRAM, and onboard 3COM 3C940 Gigabit Ethernet interconnects. The hardware components are listed as below:

- Processor (CPU): AMD Athlon 64 2900+
- Motherboard: Asus K8V Deluxe with VIA K8T800 chipset
- Memory: Transcend PC3200 (DDR 400) 512MBx3
- Storage device: Hitach 80GB 7200RPM (ATA100)
- Network Interface Card (NIC): Onboard 3COM 3C940 (Gigabit)
- Graphic card: Asus VT7100 (GeForce2 MX440)
- Others: chase, power supply, etc.

In the following subsections, we briefly introduce the choices of the hardware:

2.1 Processor

The processor can be regarded as the most important component affecting the system performance. There were many kinds of x86 series processor available on the market at that time. The Intel's products include Pentium 4, Xeon, and Itanium. AMD's products include Athlon series (formerly knows as K7). At the moment, the AMD64 [5] (a new computing platform that extends the ubiquitous x86 architecture to accommodate 64-bit processing, called x86-64 or x64 [6]) series processors just went on the market. We planned to build a 64-bit system. AMD64 series CPUs include the AMD Athlon 64 and Opteron. Because the funds were limited, we chose Athlon 64 as the basic platform. As shown in Figure 3, we can use the software WCPUID [7] to measure the detailed specification of the processors.

With the help of WCPUID, we can realize the internal clock is 1802.29 MHz, the system clock is 200.25 MHz, and that of the system bus is also 200.25 MHz. The L2 cache size is 1024 KB, and the full speed is the same with the internal clock.

Because the technology continues to advance, following "Moore's Law" [8], a wide range of prices, performance ranges, and capabilities are now available in individual computer systems. Improvements in IC technology continue to put more and more computing power into smaller and smaller packages. Systems and their processors are now edging into the 64-bit world, allowing access to larger and larger of RAM. The trends are continuing to push the cost of computing resources lower and lower per unit of computing power. Alongside the advances in processor and system technology, came advances in networking. The 64-bit technology will play an important role in high performance computing. Why 64-bit? What's the meaning of "

bit"? The word "bit" means that the width of "Register" in a processor. The "64-bit" CPU generally has integer registers that are 64 bits wide and thus directly supports dealing both internally and externally with 64-bit "chunks" of data. The advantages of 64-bit are:

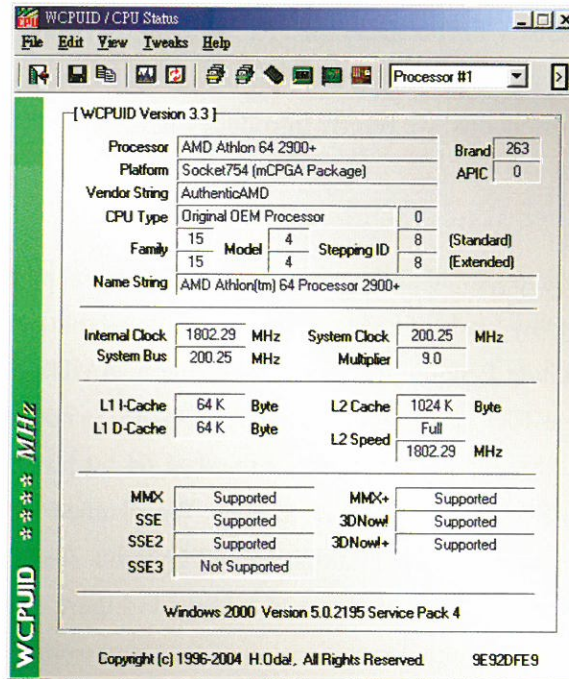


Fig. 3. Observing the characteristic of processor by WCPUID.

1. Memory addressing supports up to 64-bit that is 16 EB [9]. (EB: exabyte, 2^{60} bytes, that is 1,152,921,504,606,846,976 bytes.)
2. It can handle one data length larger than 32-bit or double precision flop points with one instruction per clock cycle (32-bit data length, for integers, i.e., from the smallest -2,147,483,648 to the largest 2,147,483,647.)

In the 32-bit system, the memory addressing supports only up to 4 GB. But for the restriction of operating systems, the application can access only up to 2 GB, the excess memory will be reserved for the system. Memory insufficient will be the "Bottleneck" of system performance. When the system has insufficient physical memory, the applications will use the virtual

memory. The usage of the virtual memory usually causes a swap of the hard disk space in which the data transmission rate is much slower than that of physical memory.

In the 64-bit system, it can address the memory up to 16 EB theoretically. Because of the physical restriction of the AMD Athlon 64 processor, for example, it allows up to 40-bit of physical, and 48-bit virtual memory address, which is 1 TB (terabyte) and 256 TB, respectively. Surely, it's also a big memory size. Common users might not need 64-bit technology and they don't need the huge memory. However, 64-bit technology will benefit scientific data analysis and related applications.

AMD64 also provides excellent compatibility and flexibility by supporting both 64-bit and 32-bit architectures. AMD's 64-bit allows the latest in processor innovation to be brought to the existing installed base of 32-bit applications and operating systems, while establishing an installed base of systems that are 64-bit capable.

AMD64 is not the first 64-bit processor. Before AMD64, a lot of manufacturers introduce 64-bit processor early. However, these 64-bit processors are all locked on the markets, such as the servers or the high-class workstations. For examples, DEC, IBM, Sun, and Hewlett-Packard introduced Alpha, Power, SPARC, and PA-RISE are all 64-bit processors. These processors are mostly used in UNIX servers. On the market of the desktop computers, the first step of marching toward 64-bit is Power Mac G5 belonging to the Apple computers, but the price is comparatively expensive and is incompatible with the general PC x86 platform. Therefore, AMD64 series processors are the first 64-bit processors widely welcomed by the market.

2.2 Motherboard

The motherboard chosen with the compatible CPU AMD Athlon 64 is the Socket 754. The chipset we chose is VIA K8T800 (north bridge).

2.3 Memory

The bandwidth of main memory has a great influence on the system performance. It's better to choose the memory with highest bandwidth supported by the motherboards. We specified the DDR DIMM module and PC3200 (DDR400) RAM.

2.4 Storage device

A hard disk is the basic storage device on PC. The interfaces of the hard disk have three kinds: IDE, SATA, and SCSI. The SCSI has higher data transfer rate, lower CPU usage, and relatively a lot of advantages, but the price is comparatively expensive. The SATA had just gone on the market at that time, though the price was cheaper than that of SCSI. Limited by the budget, we still chose the IDE interface hard disks that support Ultra ATA 100. The motor with faster rotational speed and bigger size of cache is preferred. The hard disks we chose are 7200 RPM and with a 2 MB buffer.

2.5 Network device

The network interconnects we used is 3COM 3C940 Gigabit controller chip on board. We used the Fast Ethernet switching hub at first, but the benchmark results were not so good. So we purchased the Gigabit switch later. In the next section, we can realize these two kinds of network interfaces have great influence on system performance through the benchmark results.

2.6 Display device

The PC cluster do not need high performance display card except the console (or master node), so the cheapest AGP display card is enough. In addition, only one set of keyboard, monitor, and mouse is needed. We can use the KVM (Keyboard, Video, Mouse switch) to control each node.

3 Benchmarking NSSL PC Cluster

The factors have influence on the system performance [10]:

- CPU performance (hardware architecture).
- Bandwidth of main memory (size).
- Message passing rate.
- Speed of PCI (Peripheral Component Interconnect) bus.
- Choice of solving dense linear system and BLAS [11] Libraries.

In order to realize the performance of the cluster, we evaluated the performance of the

cluster using some benchmark programs. According to the benchmark results, we can grade the performance objectively, although the benchmark results cannot represent all real efficiency of application programs. In this section, we present some benchmark results for the High-Performance Linpack and NAS Parallel benchmarks.

Before benchmarking, we compare some rates of CPU's internal clock first. As shown in Figure 4, the internal clock of Xeon 2.8 GHz is the highest, and that of IBM SP2 is the lowest. AMD 2900+ we used is ranked as the second of all, and its real internal clock is 1.8 GHz. In fact, the clock of processor can not really indicate the real performance. Different architectures of CPUs running different applications will exhibit different results.

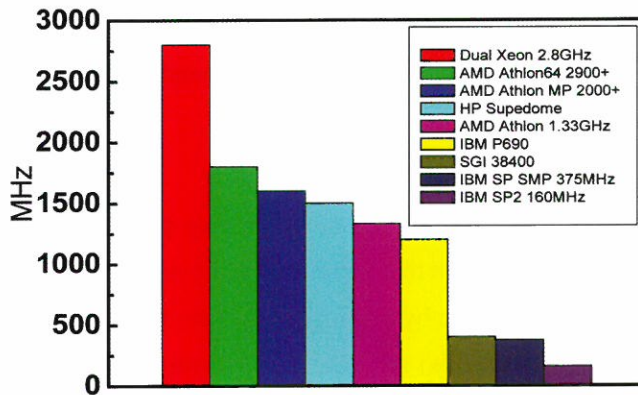


Fig. 4. Comparison of internal clock of some processors.

As shown in Figure 5, we compare some network protocols commonly used at present, such as Fast Ethernet, Gigabit Ethernet, Myrinet, Quadrics, and Infiniband. The bandwidth of the Fast Ethernet is the slowest, only 100 Mbps, and that of Infiniband achieves 7000 Mbps, the fastest one. Quadric and Myrinet are in between. Although Infiniband, Quadric, and Myrinet are all ultra-fast network devices, their prices are extremely expensive. The Gigabit devices are gradually matured at present, and the prices are much cheaper day after day, so it is increasingly popular.

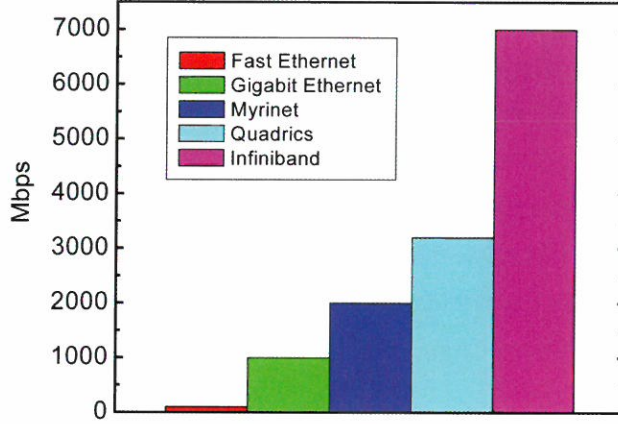


Fig. 5. Comparison of network bandwidth.

3.1 High-performance Linpack benchmark [12]

The Linpack is a measurement of a computer's floating-point rate of execution. It is determined by running a computer program that solves a dense system of linear equations. The HPL benchmark measures the performance by solving a (random) large-scale dense linear system of equations in double precision (64 bits) arithmetic on distributed-memory computers. HPL is characterized by favorable memory reference patterns and highly optimized software for memory hierarchies. However, HPL benchmark also requires a substantial amount of inter-processor communication and is especially sensitive to high communication latency. This version of the benchmark, also used by TOP500 list [13], allows the user to scale the size of the problem and to optimize the software in order to achieving the best performance for a given machine.

As shown in Table 1, we compared the benchmark results of the NSSL cluster with that of the high-performance computers in the National Center for High-Performance Computing (NCHC):

Table 1. The HPL benchmark results compared with that of NCHC.

HPL Benchmark	NSSL Cluster	Formosa Cluster	HP superdome	IBM P690
Measured performance R_{\max} (Gflops)	33.01	1166	642.9	736.6
Theoretical performance R_{peak} (Gflops)	57.6	1920	768	1331.2
Efficiency (%)	57	61	84	55
Number of CPUs	16	340	128	256
Performance/CPU (GFlops)	2.06	3.43	5	2.88
Choice of solving dense linear system and BLAS libs	HPL & ATLAS	HPL & GOTO lib	CXML	ESSL lib

R_{\max} shows the performance in Gflops for the largest problem running on a machine. R_{peak} is the theoretical peak performance Gflops for the machine. $R_{\text{peak}} = \text{Instruction per clock (IPC)} \times \text{Frequency} \times \text{Number of processors}$. The NSSL cluster, for example, .

$$R_{\text{peak}} = \frac{2}{\text{Hz}} \times 1.8 \text{ GHz} \times 16 = 57.6 \text{ Gflops}.$$

The benchmark result of HPL for the NSSL pc cluster achieves 33.01 Gflops, and the corresponding efficiency is 57%. We can compare the benchmark results with that of Formosa cluster (32-bit). The parallel efficiency is good with the aid of the Gigabit network. The Formosa cluster use GOTO [14] library, but we use ATLAS [15]. Generally, GOTO library shows better performance than ATLAS one. It's a pity that GOTO only provides the binary code only to some specific machines, such as Intel Pentium III, 4, Itanium, Itanium 2, IBM Power 3, Power 4, and AMD Opteron. It does not support the Athlon 64 processor. So we only use AT-

LAS in cooperating with HPL to do the benchmark.

3.2 NAS parallel benchmark [16]

The NAS parallel benchmark (NPB) suit was developed by NASA. The NPB has been widely used to objectively measure and compare the performance of highly parallel computers. The standard MPI programming model of NPB version 2.3 makes it possible to conduct comparative analysis of a commodity cluster with different interconnects. The NPB suite consists of a set of eight programs, which are derived from computational fluid dynamics (CFD) code. These eight programs are IS, FT, EP, MG, CG, LU, SP, and BT. All NPB benchmark programs were compiled with pgcc and pgf77 of PGI compilers.

Figure 6 shows the NAS-BT Class A benchmark results of the system with the Fast Ethernet and Gigabit and Figure 7 is the NAS-LU Class A benchmark results with the Fast Ethernet and Gigabit. As one can see in the figures, the bandwidth of network has great influence on the performance when a large number of processors are used. We also benchmark the performance of the system performed on 32-bit Red Hat Linux 9 (kernel 2.4.8-20) and 64-bit SuSE Linux 9 (kernel 2.4.21-102default), respectively, as shown in Figure 8. Basically, the results are similar.

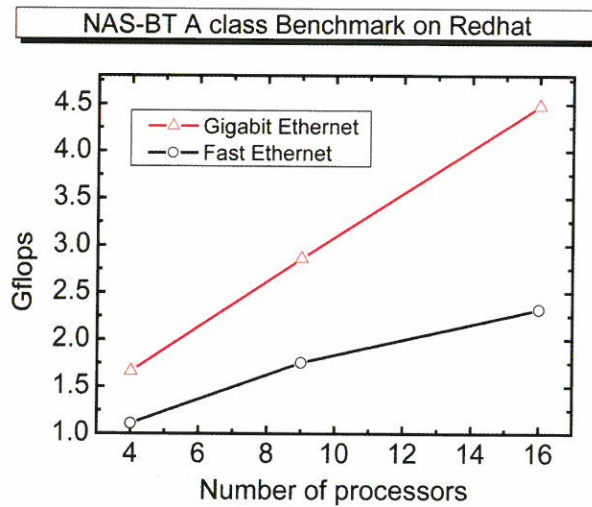


Fig. 6. NAS-BT Class A benchmark results of the system with the Fast Ethernet and Gigabit, respectively.

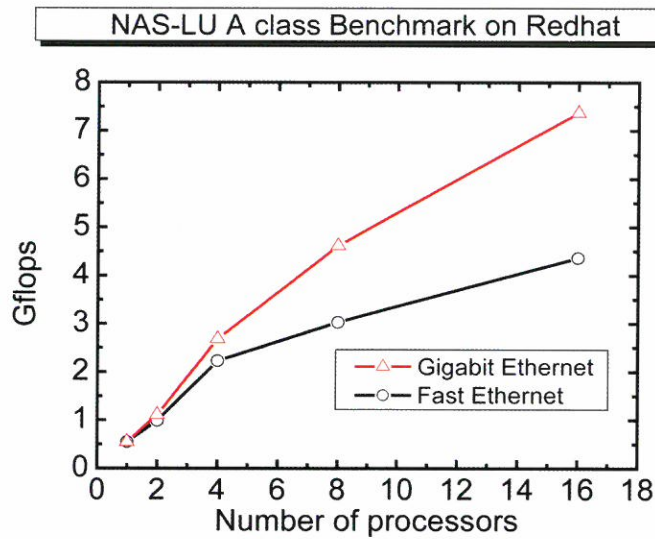


Fig. 7. NAS-LU Class A benchmark results of the system with the Fast Ethernet and Gigabit, respectively.

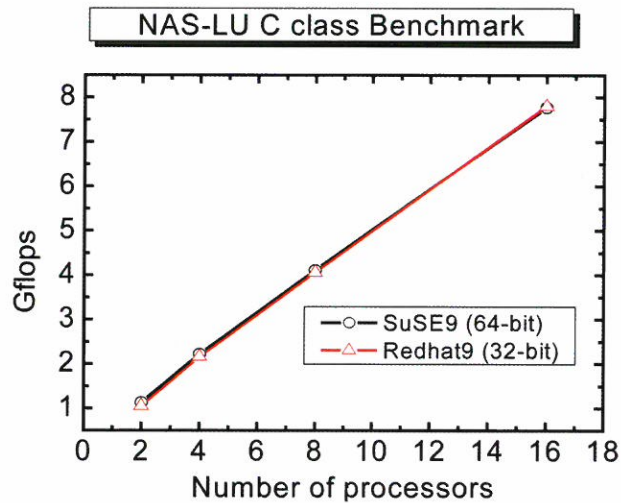


Fig. 8. NAS-LU Class C benchmark results of the system performed on the 32-bit and 64-bit platforms, respectively.

Table 2 shows the NAS-LU benchmark results with 3 levels. They are class A, B, and C. The biggest problem is the class C, the class B takes the second, and the class A is the smallest.

Table. 2, The results of NAS 2.3 LU benchmark.

Class A benchmark results			
NP	Time	Mflop/s	Mflop/s/proc
1	217.24	549.14	549.14
2	107.64	1108.32	554.16
4	44.48	2682.01	670.5
8	25.85	4614.37	576.8
16	16.2	7365.16	460.32
Class B benchmark results			
NP	Time	Mflop/s	Mflop/s/proc
1	1548.23	322.19	322.19
2	762.92	653.83	326.92
4	389.28	1281.39	320.35
8	173.43	2876.28	359.53
16	65.64	7599.96	475
Class C benchmark results			
NP	Time	Mflop/s	Mflop/s/proc
2	1941.03	1132.1	566.05
4	942.78	2218.04	554.51
8	502.21	4108.23	638.53
16	261.43	7771.31	485.71

Each level of benchmark programs has different settlement, and the detailed information can be found in the manual of NPB. We represent Table 2 with diagrams, as shown in Figure

9 and Figure 10. If we compare the Class C results in Figure 10 with that of Formosa cluster in NCHC, then we can find that our efficiency is similar to it, and the performance achieves about 4 Gflops and 7.8 Gflops with 8 processors and 16 processors, respectively.

If we compare the performance relative to a single processor, then the parallel efficiency is obvious. We found that the parallel efficiency is good, and the performance grows up linearly with the number of PC nodes.

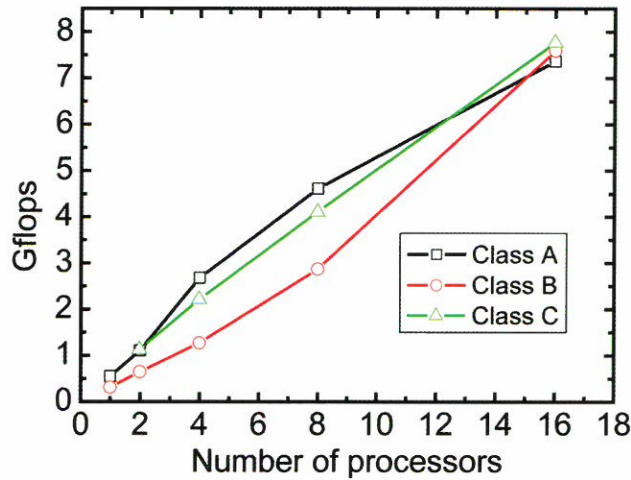


Fig. 9. Benchmark of NAS-LU. The performance achieves 7.8Gflops.

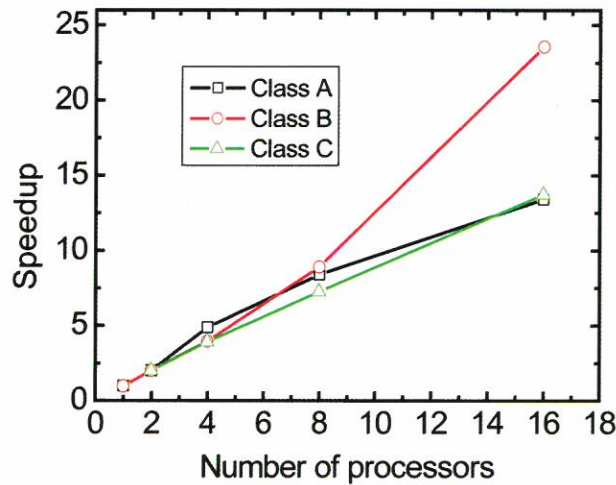


Fig. 10. Benchmark of NAS-LU. The parallel efficiency grows up linearly with the number of processors. (The curve of Class C is based on 2 processors.)

4 Testing For First-principles Packages

The first-principles computation packages installed on our cluster system are ABINIT, VASP, CPMD, WIEN2k, and SIESTA. Except for SIESTA, we can proceed the parallel computing of each.

First of all, we measure the time elapsed for computing with different number of processors. As shown in Figure 11, we solve an example of 32 water molecules with CPMD. The time elapsed of the simulation calculated with 16 processor is 1/13 of that of calculation with only one processor. Similarly, in order to realize the performance of VASP, we calculate a silver atom in a cell with 25 angstroms width. We compared the time elapsed for different number of processors. The parallel efficiency of VASP seems poorer than that of CPMD.

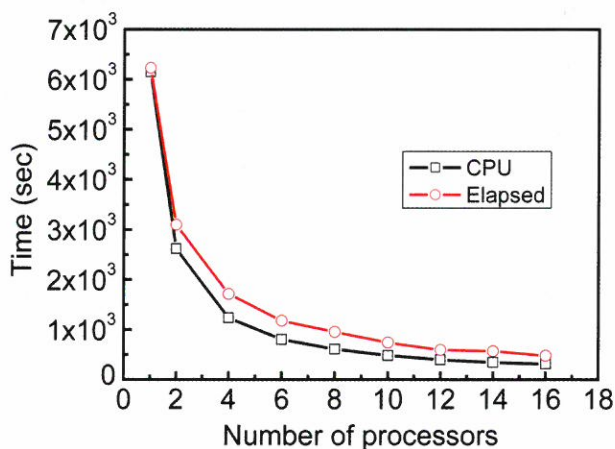


Fig. 11. Time elapsed for solving 32 water molecule with CPMD.

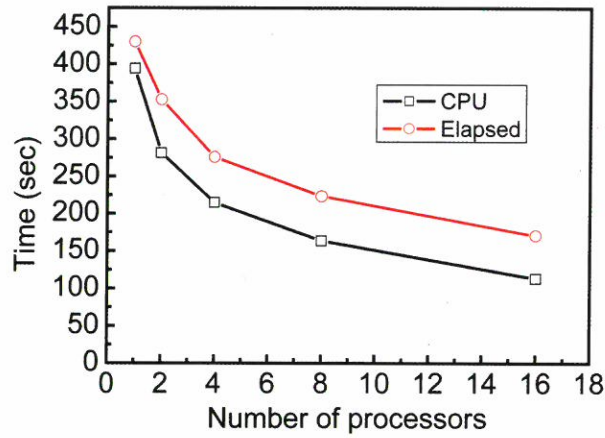


Fig. 12. Time elapsed for calculating a silver atom in a 25 angstroms wide cells with VASP.

Then we calculate the properties of bulk crystal structure, including the band structure, density of states, and optical properties. Figure 13 shows the band structure of silicon calculated using ABINIT. We compare this result with that of Chelikowsky [17]. The results are almost the same.

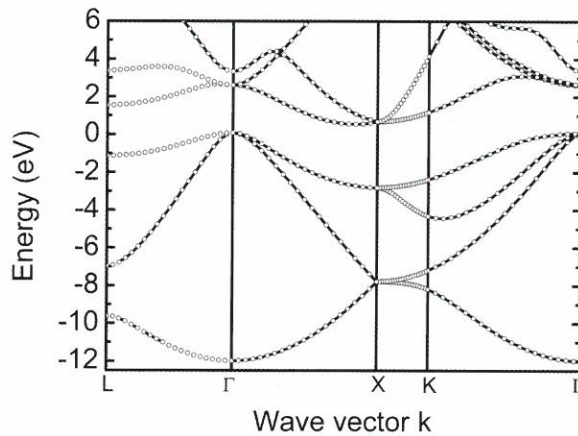


Fig. 13. Band structure of silicon calculated using ABINIT.

In addition to semiconductors, we calculate metals. Figure 14 shows the band structure of silver calculated using VASP. We compare the results with that of C. Y. Fong [18]. The results are quite identical.

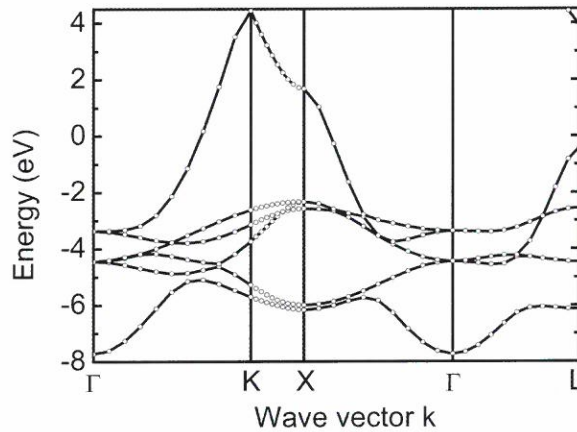


Fig. 14. Band structure of silver calculated using VASP.

The density of states of the silicon crystal is also calculated using VASP as shown in Figure 15. We compare this result with that of D. J. Stukel and R. N. Euwema [19], as shown in Figure 16, giving good agreement. Finally, we calculate the optical properties of silicon, as shown in Figure 17.

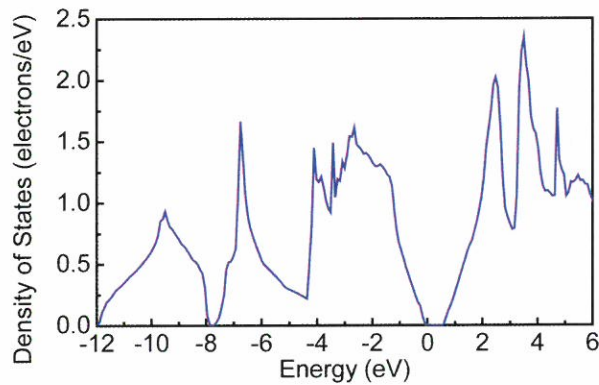


Fig. 15. Density of states of silicon calculated using VASP. We compare this result with that of D. J. Stukel and R. N. Euwema, giving good agreement.

of PC clusters and the corresponding low costs have attracted many researchers involved in the evolution of this kind of parallel computation. PC clusters will be highly required in the future high-performance computing.

In NSSL, at Physics Department of Fu Jen Catholic University, we have successfully built a 64-bit Beowulf PC cluster. From now on, the access memory of each PC node is not limited to 2 GB anymore. In addition, several first-principles simulation packages have been installed and tested. The preliminary work in this paper will be very helpful for our future research on first-principles computations.

Acknowledgment

The authors would like to thank the National Science Council under Grant No. NSC94-2112-M-030-005, National Center for High-performance Computing (NCHC), AMD, and Fu Jen Catholic University for the support of this research project.

References

- (1) National Center for High-performance Computing (NCHC), <http://www.nchc.org.tw> (2005).
- (2) 曾耀寰，企鵝雄兵—以 Linux 進行電腦叢集計算 (2001 年 11 月)。
- (3) Beowulf.org <http://www.beowulf.org> (2004).
- (4) T. L. Sterling et al., How to Build a Beowulf: a guide to the implementation and application of PC clusters (Scientific and Engineering Computation series, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1999).
- (5) AMD, Inc. <http://www.amd.com> (2004).
- (6) x86-64. <http://en.wikipedia.org/wiki/X86-64> (2005).
- (7) WCPUID version 3.3. <http://www.h-oda.com> (2004).
- (8) Gordon E. Moore, "Cramming more components onto integrated circuits," Electronics Magazine 19 (April, 1965).
- (9) EB, Exabyte. <http://en.wikipedia.org/wiki/Exabyte> (2005).

- (10) 周朝宜, Benchmark Results, CLUSTER/GRID 基礎研習營, 國家高速網路與計算中心, 民國 94 年三月。
- (11) BLAS, Basic Linear Algebra Subprograms. <http://www.netlib.org/blas> (2004).
- (12) High-Performance Linpack (HPL) Benchmark. <http://www.netlib.org/benchmark/hpl> (2004).
- (13) TOP500 list for November 2003. <http://www.top500.org/list/2003/11/> (2003).
- (14) GOTO, High-Performance BLAS by Kazushige Goto. <http://www.cs.utexas.edu/users/flame/goto> (2004).
- (15) ATLAS, Automatically Tuned Linear Algebra. <http://math-atlas.sourceforge.net> (2004).
- (16) NAS Parallel Benchmark. <http://www.nas.nasa.gov/Software/NPB> (2004).
- (17) J. R. Chelikowsky and M. L. Cohen, Nonlocal pseudopotential calculations for the electronic structure of eleven diamond and zinc-blende semiconductors, Phys. Rev. B 14, 1976, pp. 556-582.
- (18) C. Y. Fong, J. P. Walter, and Marvin L. Cohen, Comparison of band structures and charge distributions of copper and silver, Phys. Rev. B 11, 1975, pp. 2760.
- (19) D. J. Stukel and R. N. Euwema, Self-consistent orthogonalized- plane-wave energy-band study of silicon, Phys. Rev. B 1, 1970, pp. 1635-1643.

received October 27, 2006

accepted December 2, 2006

以 64 位元北歐武夫電腦叢集進行第一原理計算

林銘杰 黃國華

輔仁大學物理系

中文摘要

近年來，在奈米科學研究上，數值模擬愈來愈受人重視。在第一原理模擬的研究領域，我們常常需要花大量時間在電腦模擬上，也許是數個小時、數天，甚至幾個星期。我們要如何能夠減少花費在電腦模擬計算的時間呢？低成本、高效能的電腦叢集是增進計算效能的最佳解決方案。目前龐大的科學計算極需要平行計算。

一方面由於個人電腦運算速度效能快速提升，而價格相對愈來愈低，再加上網路技術快速發展，個人電腦叢集在科學計算的應用也愈來愈廣泛。本文主要介紹輔仁大學物理系奈米科學模擬研究室(NSSL)自行組裝的 64 位元北歐武夫(Beowulf)電腦叢集計算平台，包含建置經驗、心得和效能測試等，以及常用的第一原理計算程式之安裝測試與運算效能。

關鍵詞：電腦叢集、北歐武夫(Beowulf)、超微(AMD)、64 位元、第一原理

拿破崙紙積木之研究

呂克明

亞洲大學資訊工程學系教授

曹子殷

亞洲大學資訊與設計學系兼任講師

陳美汎

亞洲大學資訊工程學系碩士研究生

摘 要

一般積木立體觀念，都屬於立體幾何圖形，有各種不同的立體外型。單一積木的外型於成型時，就被限制住。使用者需事先構思好圖案，再藉由各種不同造型的積木而堆疊出所需表達的立體形狀。故，如何將一個菱形平面密鋪圖案，轉變成立體密鋪模型，實為紙積木研究之重點。紙積木，有拆開重覆使用與可著色的特點。當孩子不喜歡的造型，可以拆開重來，甚至基本造型也可以，這是木質積木或是塑膠積木所不能取代的。這個創造力就是一種生產力，也是推陳出新的力量。

本研究從拿破崙平面密鋪的理論開始，藉 32 個正三角形組成的大菱形和四種方法建構，領結、翅膀、眼淚、巧克力條、船型、星型、山形袖章與鯉魚等 8 種拿破崙紙積木模型。為了增加紙積木模塊的數目，我們開發兩個步驟，有系統地創造出包括基本型、從屬型、與衍生型等 54 種模型。我們連接這些紙積木，創作包括有把手

*Corresponding author. Tel:+886-4-23323456 ext.6277

E-mail: klu@asia.edu.tw

的結合體。藉由模型皆有角、面、邊三個要件的特點，我們可以計算出多面體的歐拉特性數，來驗證所創造出虧格數為 1 或是大於 1 的結合體。

關鍵詞：拿破崙密鋪圖案、歐拉-龐加萊公式、歐拉特性數、虧格數

一、前言

積木玩法變化多端，可讓孩子發揮無窮想像力，自由砌出自己心愛的模型，是經濟又好玩的玩具。

一般積木立體觀念，都屬於立體幾何圖形，例如四方體、柱體、椎體、球體等各種不同的立體外型。單一積木的外型於成型時，就被限制住。使用者需事先構思好圖案，再藉由各種不同造型的積木而堆疊出所需表達的立體形狀。故，如何將一個菱形平面密鋪圖案，轉變成立體密鋪模型，實為摺紙積木研究之重點。紙積木，有拆開重覆使用與可著色的特點。當孩子不喜歡的造型，可以拆開重來，甚至基本造型也可以，這是木質積木或是塑膠積木所不能取代的。這個創造力就是一種生產力，也是推陳出新的力量。凱茲（Katz, 1978）曾指出，要有創造力，除須有認知外，尚須能把此認知整合與精進，而這也就是邏輯思考的能力，也是一種擴散性思考能力¹。

1.1 積木

積木的英文，是「toy blocks」、「building blocks」或簡稱是「blocks」。積木，是正方形、圓柱形、拱形、三角形等形狀與顏色的木質或塑膠質模塊²。威多德·黎辛斯基³發現最早提到兒童積木是以「合理的玩具⁴」一詞，出現在馬利亞和她父親⁵的共同

¹ divergent thinking

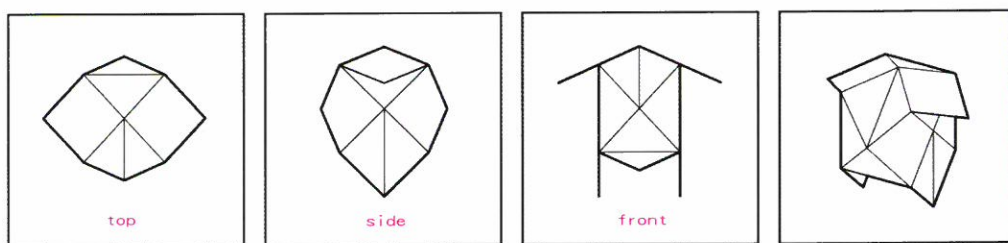
² <http://encyclopedia.thefreedictionary.com/Toy+block>

³ 黎辛斯基(Witold Rybczynski)為波蘭裔，1943 年出生於愛丁堡，在英國與加拿大的耶穌會學校進學。他獲有蒙特婁麥基爾(McGill)大學的建築學士(1960)與碩士(1972)學位，著有五十多項關於房屋、建築、與科技主題的著述，包括《馴虎》、《紙英雄》、《世上最美麗的房屋》、《等候週末》、《漫遊建築世界》、《城市生活》(City Life)，以及使他榮獲克里斯多福獎(Christopher Award)和魯卡斯獎(J. Anthony Lukas Prize)的《遠方林中的空地》，最新作品為《螺絲、起子演化史》。他定期為《大西洋月刊》、《紐約客》雜誌、《紐約時報雜誌》及《紐約書評》撰稿，目前則任教於賓州大學。

著作《實驗教育學》一書。這些方塊目的是在教導兒童有關地引力物理以及空間的關係，使得他們了解許多不同的部件如何組合成整體（Rybczynski 1992）。在十九世紀中葉，亨利·高爾爵士⁶以筆名費立克斯·桑莫利⁷，撰寫了一套兒童書籍。他的《家庭寶藏的故事》套書中，描述一箱陶土製成的積木。同時附帶有一本題名為《舊式建築》的小冊子內含有這些如何建造的藍圖。

1.2 紙積木的好處

紙積木，是紙質積木，或泛指可摺疊平面材料做成的積木。平面密鋪圖案，有些可以做成立體密鋪，例如饕餮立體密鋪⁸與拿破崙立體密鋪。但，不是所有的立體密鋪均可以堆疊紙積木，饕餮立體密鋪其中的一個立體造型就不可以，例如六面饕餮，請參見圖一。拿破崙紙積木，是應用拿破崙平面密鋪的原理，摺紙做成，並且可以堆疊的立體模型。它們與其他積木一樣，有四個好處⁹：



圖一 六面饕餮

1. 體能—積木可以增強兒童手指與手掌，以及增進眼睛的協調。
2. 社會—玩積木在於鼓勵兒童結交朋友與合作，往往就是小孩初次與其他小孩在一起玩的經驗。

⁴ rational toys

⁵ Richard Lovell Edgeworth, 1798

⁶ Sir Henry Cole, 1808-1882

⁷ Felix Summerly

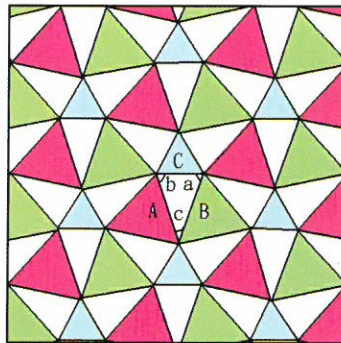
⁸ 呂克明、曹子殷、白嘉祥在東吳大學 CGW2006 (Computer Graphics Workshop), Soochow University 發表《饕餮密鋪 (Taotie Tessellation)》(Lu 2006)

⁹ <http://encyclopedia.thefreedictionary.com/Toy+block>

3. 智慧—兒童可以藉由學習描述大小、形狀、位置之時，增加字彙；藉由群組、加法與減法的流程來發展數學的技巧；藉由從積木學習地心引力、平衡與幾何的經驗來激發智慧。
4. 創造—兒童藉由親手設計，獲得創造的激發。

二、拿破崙平面密鋪

吉姆·羅依¹⁰（2003）為拿破崙定理下了定義¹¹。根據此定理，一個任意邊長的三角形的三個邊，分別做出三個正三角形，三個任意三角形與三個正三角形，即構成拿破崙平面密鋪。請參照圖二。此任意三角形的邊分別為 A 、 B 、 C 與對角 a 、 b 、 c 。拿破崙平面密鋪，就是由這六個三角形所組成的，假若每個三角形塗上不同的顏色，就需要六個顏色。必須要將六個三角形組，簡單化與美化。



圖二 拿破崙平面密鋪¹²

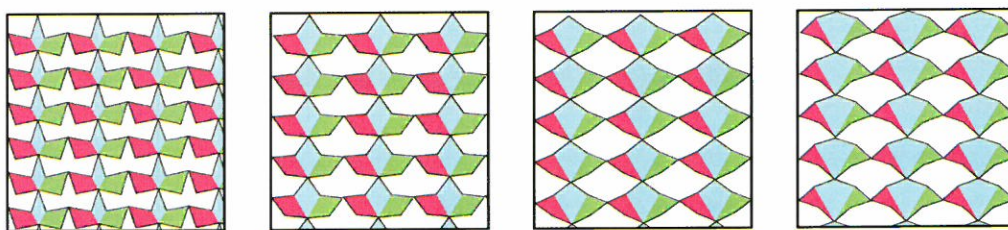
¹⁰ Jim Loy' s website- <http://www.jimloy.com/geometry/napoleon.htm>

¹¹ The earliest definite appearance of this theorem is an 1825 article by Dr. W. Rutherford in "The Ladies Diary". Although Rutherford was probably not the first discoverer, there seems to be no direct evidence supporting any connection with Napoleon Bonaparte, although we know that he did well in mathematics as a school boy. (<http://www.mathpages.com/home/index.htm>)

¹² Excerpt from Author' s article, A Study of Innovative Plane Tessellation Patterns Systems (Lu 2005)

2.1 緣起

呂克明（2005）採用創新的簡化與美化的流程，建立了六邊鼓形、八邊三葉形與八邊雁形等三個拿破崙平面圖案組。其中僅有八邊三葉形是等邊對稱的圖案。請參照圖三有關 36° 、 72° 、 108° 與 144° 拿破崙平面密鋪。同時注意 36° 與 72° 八邊三葉形與彭羅斯的 36° 與 72° 黃金分割比率¹³有關（Penrose 1989）。



圖三 36° 、 72° 、 108° 與 144° 拿破崙平面密鋪

2.2 α -拿破崙圖案與其特性

圖三是由八邊三葉形的等腰三角形的夾角 α 等於 36° 、 72° 、 108° 與 144° 的拿破崙平面密鋪。在偶然的機會，我們發現當夾角 α 從 0° 變化到 180° 的 α -拿破崙平面密鋪圖案，有幾個如下有趣的現象：

特性 1：等腰三角形的底邊長 $\lambda=2\sin(\alpha/2)$

假設等腰三角形的頂夾角 α ，我們可以算出兩個底角 $\beta=(180^\circ-\alpha)/2$ 。根據正弦定律¹⁴，我們可以推導出底邊長 $\lambda=2\sin(\alpha/2)$ 。

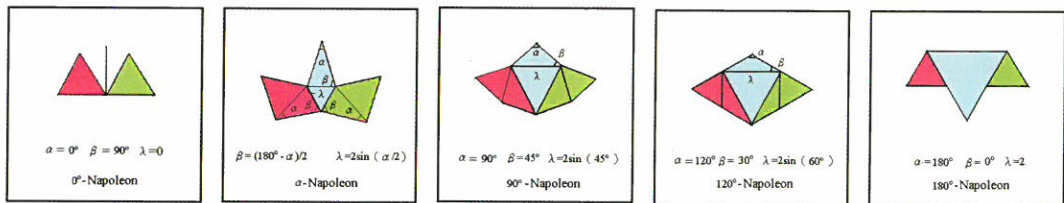
特性 2：頂夾角 α 的等腰三角形面積為 $\sin(\alpha)/2$

根據面積公式，我們可以計算出等腰三角形面積為 $1 \times 1 \times \sin(\alpha)/2 = \sin(\alpha)/2$ 。

¹³ 黃金分割（Golden Section）的比率，是方程式 $X^2-X-1=0$ 的正值根，即 $(1+\sqrt{5})/2=1.61803\ 39887\ 49894\ 84820$ （又稱 Phi Number）。

¹⁴ The law of sine states: $\sin(A)/a = \sin(B)/b = \sin(C)/c$

特性 3： α -拿破崙圖案最大與最小面積，分別當 $\alpha=120^\circ$ 與 $\alpha=0^\circ$ 時，為 8 個與 2 個單位正三角形。

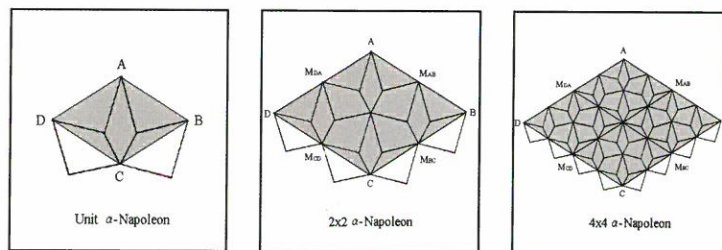


圖四 從 0° — 180° 變化之 α 角的拿破崙圖案

α -拿破崙圖案是由 6 個三角形組成，包括 3 個頂夾角 α 的等腰三角形、2 個單位長度為邊長的正三角形、1 個以底邊長 λ 為邊長的正三角形。將 6 個三角形的面積對頂夾角 α 取微分，我們可以得到當 $\alpha=0^\circ$ 時，最小面積為 2 個單位正三角形，而當 $\alpha=120^\circ$ 時，最大面積為 8 個單位正三角形。

特性 4： α -拿破崙圖案能夠製作平面密鋪。

根據兩平移準則之一¹⁵，我們可以證明拿破崙圖案可以創造出夾角 α -拿破崙平面鋪。設 α -拿破崙圖案的四個角點分別為 A、B、C 與 D， α -拿破崙圖案的 A 到 B 邊的部份可以平移 D 到 C 邊，並且 B 到 C 邊的部份可以平移 A 到 D 邊。請參照圖五兩平移準則之一，可資證明 α -拿破崙圖案能夠密鋪。



圖五 一個單位、 2×2 、與 4×4 α -拿破崙圖案

¹⁵ Refer to Translation criterion from Chapter 20 Tiling, p.767. (COMAP, 2005)

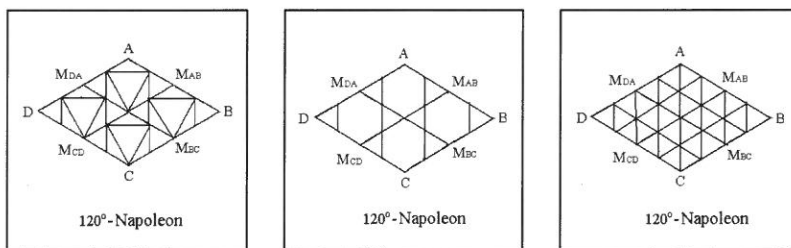
特性 5： $2n \times 2n$ α -拿破崙圖案能夠創造出一個立體的表面被一個週期平面密鋪包裹。設 n 屬正整數 \mathbb{Z}^+ 。

我們重複 $2n$ 次一個單位的 α -拿破崙圖案製成 $2n \times 2n$ α -拿破崙圖案。請參照圖五的 2×2 α -拿破崙圖案與 4×4 α -拿破崙圖案。設 $2n \times 2n$ α -拿破崙圖案的四個角點分別為 A、B、C 與 D， α -拿破崙圖案的 A 到 B 邊的部份可以平移 D 到 C 邊，並且 B 到 C 邊的部份可以平移 A 到 D 邊。

我們以 AB 邊與 DC 邊中點 M_{AB} 與 M_{CD} 連線對摺，然後邊到邊用膠帶分別連結¹⁶AB 邊與 DC 邊，兩個角 A 與 B 以及兩個角 C 與 D 重疊。兩中點 M_{AB} 與 M_{CD} 之間會有 $2n$ 單位的線段。其次，我們在兩中點 M_{DA} 與 M_{BC} 連線對摺，做邊到邊用膠帶之連結 AD 邊與 BC 邊。最後 A、B、C 與 D 四點重疊在一起，於是證明一個立體的表面被一個週期平面密鋪包裹。

三、 2×2 拿破崙立體密鋪

根據特性 5 並假設 n 等於 1，我們當下研究 2×2 120° 拿破崙圖案，如圖六的菱形 ABCD。



圖六 轉換成簡化 2×2 120° 拿破崙圖案

¹⁶ In an edge-to-edge gluing each entire edge of the polygon is glued to precisely one other edge of the polygon. (Demaine and O'Rourke, 2005)

我們以 AB 邊與 DC 邊中點 M_{AB} 與 M_{CD} 連線對摺，然後邊到邊用膠帶分別連結 AB 邊與 DC 邊，兩個角 A 與 B 以及兩個角 C 與 D 重疊。因為 120° 拿破崙圖案是一個 60° 的菱形，兩中點 M_{AB} 與 M_{CD} 之間不是 $2n$ 單位的線段，而是直線。其次，我們在兩中點 M_{DA} 與 M_{BC} 連線對摺，做邊到邊用膠帶之連結 AD 邊與 BC 邊。最後 A、B、C 與 D 四點重疊在一起，於是證明一個立體的表面被一個週期平面密鋪包裹。

我們在特性 3 得到 120° 拿破崙圖案是 α -拿破崙圖案中最大面積為 8 個單位的結論。同時，它的兩中點 M_{AB} 與 M_{CD} 之間以及兩中點 M_{BC} 與 M_{DA} 是直線，我們有足夠的理由，進一步簡化 120° 拿破崙圖案成為一個菱形的 8 正三角形塊組¹⁷而不改 120° 拿破崙圖案的形狀。簡化後的 2×2 120° 拿破崙圖案是 4×8 或 32 正三角形，或 32 正三角形塊組¹⁸。

四、 2×2 拿破崙立體模型

一個 2×2 120° 拿破崙圖案，具有 25 個角、32 個面以及 56 個邊。我們將其中 16 個邊分成 8 個連續邊與 8 個連續邊連結。當完成時，我們將會減少 7 個角與 8 個邊。換句話說，拿破崙立體模型將有 $18(=25-7)$ 個角、32 個面以及 $48(=56-8)$ 個邊。根據歐拉-龐加萊¹⁹ 公式 (Bondy and Murty 1976)：歐拉特性數 = $VFE(18, 32, 48) = V + F - E = 18 + 32 - 48 = 2$ 。

為了建構拿破崙立體模型，我們必須先訂立建構原則與建構步驟。立體密鋪模型建構之三原則：1. 兩正三角形不可密貼；2. 模型必須至少有一方向是對稱；3. 翻轉祇能算一個。立體密鋪創新建構四步驟：1. G1: 兩組八邊開口或簡稱 2×8 ；2. G2: 一組八邊開口與兩組四邊粘合簡稱 1×8 與 2×4 ；3. G3: 兩組六邊開口與兩組二邊粘合簡稱 2×6 與 2×2 ；4. G4: 兩組四邊開口與一組八邊粘合簡稱 2×4 與 1×8 。我們將討論四組立體製模的流程。

¹⁷ A polyiamond (also polyamond or simply iamond) is a polyform in which the base form is an equilateral triangle.
<http://en.wikipedia.org/wiki/Polyiamond>, 8-polyiamond

¹⁸ 32-polyiamond

¹⁹ Jules-Henri Poincare, 1854-1912, 歐拉特性數= $VFE(V, F, E) = V + F - E = 2 - 2 * g$, g 為虧格數, $g=0$.

4.1 G1 群組立體製模

G1 群組立體製模，含有連接角點 A 與 C 的次群組 SG1 和連接角點 B 與 D 的次群組 SG2。因為角點 A 與 C 均屬於 120° ，在次群組 SG1 的流程中，8 個邊的開口僅存留 60° ，因此我們僅能夠創造出領結模型與翅膀模型。相反地，因為角點 B 與 D 均屬於 60° ，在次群組 SG2 的流程中，8 個邊的開口存留 120° ，我們可以創造出領結模型與眼淚模型。

4.1.1 領結模型

領結模型，具有俯視、側視與前視中心線對稱的特性。它的體積 2.857 立方單位²⁰，為 8 個基本型中最大者。我們可以使用 G1、G2 與 G4 三個群組當中的 10 種方法建立領結模型。創造領結模型，比較經常使用的方法是 G4 群組立體製模。

4.1.2 翅膀模型

翅膀模型的體積 1.886 立方單位，為 8 個基本型中最小者。含有連接角點 A 與 C 的次群組 SG1 流程是唯一，也是比較容易製作翅膀模型的方法。

4.1.3 眼淚模型

在次群組 SG1 流程當中，有兩種方法製作眼淚模型。要訣是將獨立四面體先連接，然後才是主體。

4.2 G2 群組立體製模

在 G2 群組立體製模流程當中，首先連接兩端各 4 個邊，然後使用次群組 SG1 與次群組 SG2 將 8 個邊的開口連接，製作巧克力條、領結與星型 3 個模型。

4.2.1 星型模型

星型模型以 4 個角，為 8 大基本型中，角最多而著名。我們可以將 4 個角截去而製作成鼓型，它與領結模型同時被發現的。

²⁰ 將正三角形的邊長設為一個單位。

4.2.2 巧克力條模型

巧克力條模型，具有俯視、側視與前視中心線對稱的特性。它是 8 大基本型中，有 7 個相鄰的正三角形在一個平面上的模型。往後，可以製作虧格數 n 的結合體的衍生型之一。

4.3 G3 群組立體製模

連接 2 組 2 邊，留下 2 個 6 邊的開口，是 G3 群組立體製模獨特，也是唯一可以製作船型所使用的流程。

4.3.1 船型模型

與巧克力條模型是孿生模型的船型模型，具有俯視、側視與前視中心線對稱的特性。它也是 8 大基本型中，有 7 個相鄰的正三角形在一個平面上的模型。

4.3.2 山形袖章模型

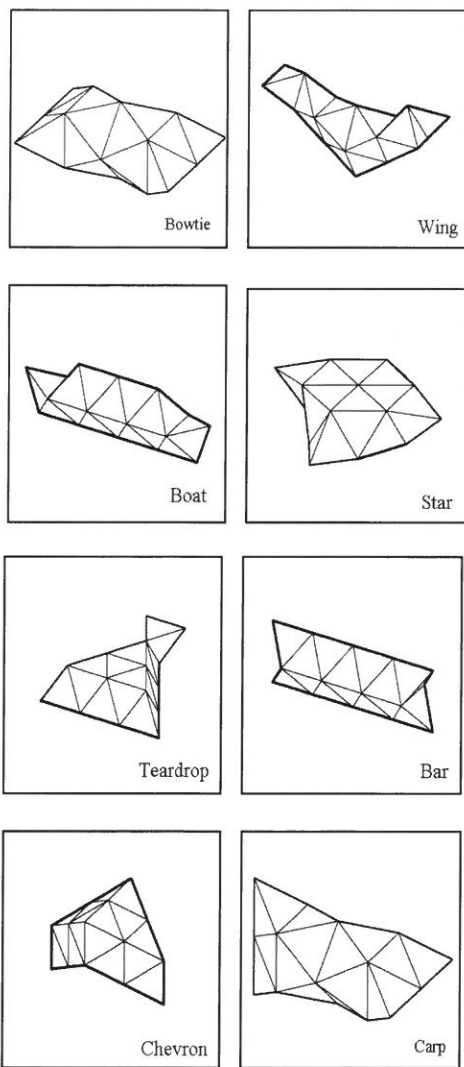
倒立 V 字形的山形袖章，具有俯視、側視與前視中心線對稱的特性。它的兩片四個正三角形的同一平面構面，是獨一無二的模型。

4.4 G4 群組立體製模

G4 群組立體製模，是鯉魚模型專屬製作流程。將 2 組 4 個邊連接，是 4 個群組立體製模當中，困難度比較高的一種。

4.4.1 鯉魚模型

中國字鯉魚的「鯉」與「利」同音。因此，鯉魚象徵事業成功、發大利市。還有鯉魚跳龍門的傳說故事，一般人對鯉魚都有好



圖七 拿破崙紙積木 8 個基本型

感²¹。與領結模型是孿生模型的鯉魚模型，也具有俯視、側視與前視中心線對稱的特性。

4.5 基本型製模總結

應用立體密鋪建構四步驟，我們創造出 8 個基本型。請參照圖七拿破崙紙積木 8 個基本型與表一拿破崙立體模型與建構步驟。

表一 拿破崙立體模型與建構步驟

組別	第一部結合	第二部結合	模型
G1 2×8	SG1 A-C	A-M _{DA} /D-M _{DA} , D-M _{CD} /C-M _{CD} , C-M _{BC} /B-M _{BC} , and B-M _{AB} /A-M _{AB}	領結
		A-M _{AB} /C-M _{BC} and A-M _{DA} /C-M _{CD}	翅膀
	SG2 B-D	A-M _{AB} /M _{DA} -D and C-M _{BC} /M _{CD} -D	眼淚
		A-M _{DA} /M _{AB} -B and C-M _{CD} /M _{BC} -B	眼淚
		B-M _{AB} /A-M _{AB} , A-M _{DA} /D-M _{DA} , D-M _{CD} /C-M _{CD} , and C-M _{BC} /B-M _{BC}	領結
G2 1×8, 2×4	SG1 A-M _{AB} /B-M _{AB} and C-M _{CD} /D-M _{CD}	A-M _{DA} -D and B-M _{BC} -C	巧克力條
		A-M _{DA} /B-M _{BC} and C-M _{BC} /D-M _{DA}	領結
		A-M _{DA} /D-M _{DA} and B-M _{BC} /C-M _{BC}	領結
		B-M _{BC} and M _{DA} -D	星型
	SG2 A-M _{DA} /D-M _{DA} and B-M _{BC} /C-M _{BC}	A-M _{AB} -B and D-M _{CD} -C	巧克力條
		A-M _{AB} /B-M _{AB} and C-M _{CD} /D-M _{CD}	領結
		A-M _{AB} /D-M _{CD} and B-M _{AB} /C-M _{CD}	領結
G3 2×6, 2×2	SG1 A-M _{AB} and M _{CD} -C	B-M _{AB} and M _{CD} -D	星型
		A-M _{AB} /M _{DA} -D and B-M _{CD} /M _{BC} -C	山形袖章
	SG2 A-M _{DA} and M _{BC} -C	A-B-M _{DA} and C-D-M _{BC}	星型
		A-M _{DA} /M _{AB} -B and C-M _{BC} /M _{CD} -D	山形袖章
	SG3 B-M _{AB} and M _{CD} -D	A-M _{AD} /D-M _{AD} and B-M _{BC} /C-M _{BC}	領結
		A-M _{DA} /D and B-M _{BC} /C	領結
	SG4 B-M _{BC} and M _{DA} -D	A-M _{AD} /D-M _{AD} and B-M _{BC} /C	鯉魚
A-M _{DA} /D and B-M _{BC} /C-M _{BC}		鯉魚	
G4 2×4, 1×8	SG1 A-M _{AB} -B and D-M _{CD} -C	A-M _{AB} /B-M _{AB} and C-M _{CD} /D-M _{CD}	領結
		A-M _{AB} /B and C-M _{CD} /D	領結
		A-M _{AB} /B-M _{AB} and C-M _{CD} /D	鯉魚
		A-M _{AB} /B and C-M _{CD} /D-M _{CD}	鯉魚
	SG2 A-M _{DA} -D and B-M _{BC} -C	A-M _{AB} /B and C-M _{CD} /D-M _{CD}	鯉魚
		A-M _{AB} /B and C-M _{CD} /D-M _{CD}	鯉魚

²¹ The text of the article was excerpted from the website of Feng Shui at Dragon Gate.
<http://www.dragon-gate.com/>.

布施知子（Fuse 1990）建議使用角、面與邊的數目來代表多面體。請參照表二，了解拿破崙紙積木相關之拓樸、幾何與所屬特性。我們同時發現每一基本型的同形圖案數²²。柏化沙那菲（Parthasarathy 1994）指出在平面上使用不同的方法畫出相同圖案並不是很重要，代表特定圖案的數目才是重要。

表二 拿破崙紙積木之拓樸、幾何與所屬特性

模型	角	面	邊	表面積	體積	同形圖案數	六面體	四面體	金字塔型體
領結	18	32	48	13.856	2.857	10	4	4	0
翅膀					1.886	1	0	0	1
眼淚					2.357	2	0	2	0
巧克力條					2.357	4	0	4	0
船型					2.357	2	0	2	0
星型					2.809	4	0	4	0
山形袖章					2.828	2	1 隱藏	2, 2 隱藏	2
鯉魚					2.857	4	2	4	0

五、拿破崙紙積木造型

爲了增加紙積木模塊的數目，我們開發兩個步驟，創造拿破崙紙積木 8 個基本型外的 46 個衍生造型。首先，我們找出 8 個基本型，可能含有六面體的屬性。藉由將此六面體向內壓擠，或是向外壓出來發展出「從屬型」。從屬型的型號，將是在基本型型號 00，更換第一個 0，而成 A0、B0、C0。在這個步驟當中，基本型所衍生出來的歐拉特性數將不改變。換句話說，仍然是 VFE (18, 32, 48)。

²² Definition: Two graphs are isomorphic if there is a one-to-one correspondence between their vertices and there is an edge between two vertices of one graph if and only if there is an edge between the two corresponding vertices in the other graph. <http://www.nist.gov/dads/HTML/isomorphic.html>

其次，我們將截去所有可能的四面體或是金字塔體，發展出「衍生型」。這個步驟，有兩個限制：a) 相鄰的四面體或是金字塔體不可以截去；b) 維持對稱性。我們將依照基本型與從屬型的特性分別討論。衍生型的型號，將是在基本型型號 00 或是從屬型型號 A0、B0、C0，分別更換第二個 0，而成 01、02、03 或是 A1、A2、A3、B1、B2、B3、C1、C2、C3。

因為歐拉特性數是結合多個拿破崙紙積木，計算虧格數的重要數字。以下是截去四面體或是金字塔體計算歐拉特性數的四個公式：

(一) 截去 1 個四面體的歐拉特性數

截去 1 個四面體的積木，我們將會扣除四面體具有的 1 個角點、3 個面與 3 個邊，然後補上 1 個面，即扣除 1 個角點、2 個面與 3 個邊。故，歐拉特性數將會從 VFE (18, 32, 48) 減去 VFE (1, 2, 3)。其歐拉特性數計算公式如下：

$$\text{VFE (18, 32, 48)} - \text{VFE (1, 2, 3)} = \text{VFE (17, 30, 45)} = 17+30-45=2.$$

(二) 截去 2 個四面體的歐拉特性數

截去 2 個四面體的積木，我們將會扣除二個四面體具有的 2 個角點、6 個面與 6 個邊，然後補上 2 個面，即扣除 2 個角點、4 個面與 6 個邊。故，歐拉特性數將會從 VFE (18, 32, 48) 減去 2*VFE (1, 2, 3)。其歐拉特性數計算公式如下：

$$\text{VFE (18, 32, 48)} - 2*\text{VFE (1, 2, 3)} = \text{VFE (16, 28, 42)} = 16+28-42=2.$$

(三) 截去 1 個金字塔體的歐拉特性數

截去 1 個金字塔體的積木，我們將會扣除金字塔體具有的 1 個角點、4 個面與 4 個邊，然後補上 1 個面，即扣除 1 個角點、3 個面與 4 個邊。故，歐拉特性數將會從 VFE (18, 32, 48) 減去 VFE (1, 3, 4)。其歐拉特性數計算公式如下：

$$\text{VFE (18, 32, 48)} - \text{VFE (1, 3, 4)} = \text{VFE (17, 29, 44)} = 17+29-44=2.$$

(四) 截去 2 個金字塔體的歐拉特性數

截去 2 個金字塔體的積木，我們將會扣除 2 個金字塔體具有的 2 個角點、8 個面與 8 個邊，然後補上 2 個面，即扣除 2 個角點、6 個面與 8 個邊。故，歐拉特性數將會從 VFE (18, 32, 48) 減去 2*VFE (1, 3, 4)。其歐拉特性數計算公式如下：

$$\text{VFE (18, 32, 48)} - 2*\text{VFE (1, 3, 4)} = \text{VFE (16, 26, 40)} = 16+26-40=2.$$

5.1 領結模型

領結模型，具有俯視、側視與前視中心線對稱的特性，並且有兩對六面體。故，我們會有 1 個基本型領結 00 與 2 個從屬型領結 A0 與領結 B0。領結 A0 與領結 B0，分別是壓擠一對六面體後創造出的從屬型。我們就這 1 個基本型領結 00 與 2 個從屬型領結 A0 與領結 B0，分別討論其衍生型。

5.1.1 領結 00

領結 00 代表一組 4 個沒有壓擠六面體的領結基本型：領結 00、領結 01、領結 02、領結 03。截去 1 個四面體的衍生型領結 01 的歐拉特性數，計算如下：

$$VFE(18, 32, 48) - VFE(1, 2, 3) = VFE(17, 30, 45) = 17+30-45=2$$

同樣地，因為相鄰的 2 個四面體不能被截去，我們將僅有 2 個截去 2 個四面體的衍生型：領結 02 與領結 03 的歐拉特性數，計算如下：

$$VFE(18, 32, 48) - 2*VFE(1, 2, 3) = VFE(16, 28, 42) = 16+28-42=2.$$

5.1.2 領結 A0

領結 A0 代表一組 4 個沒有壓擠六面體的領結基本型：領結 A0、領結 A1、領結 A2、領結 A3。截去 1 個四面體的衍生型領結 A1 的歐拉特性數，計算如下：

$$VFE(18, 32, 48) - VFE(1, 2, 3) = VFE(17, 30, 45) = 17+30-45=2$$

同樣地，因為相鄰的 2 個四面體不能被截去，我們將僅有 2 個截去 2 個四面體的衍生型：領結 A2 與領結 A3 的歐拉特性數，計算如下：

$$VFE(18, 32, 48) - 2*VFE(1, 2, 3) = VFE(16, 28, 42) = 16+28-42=2.$$

5.1.3 領結 B0

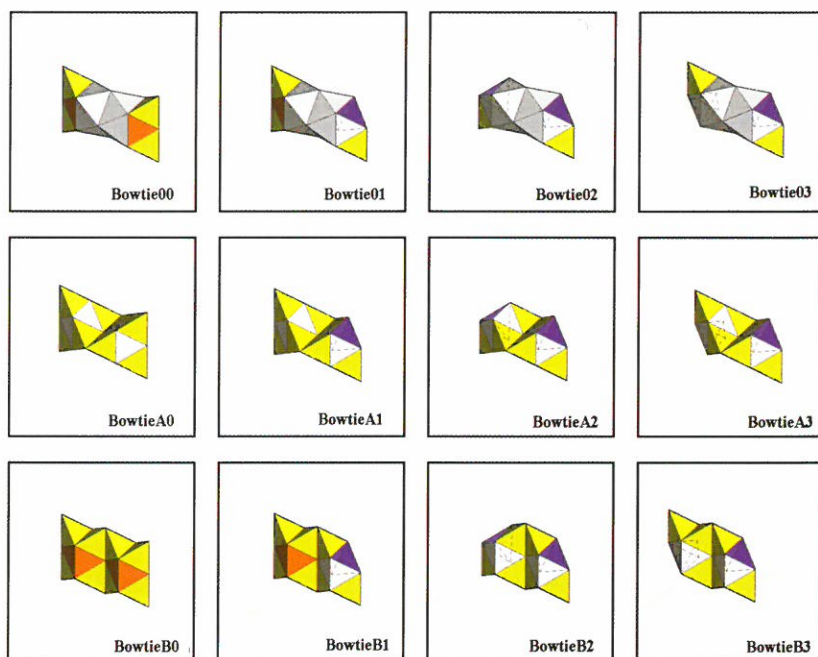
領結 B0 代表一組 4 個沒有壓擠六面體的領結基本型：領結 B0、領結 B1、領結 B2、領結 B3。截去 1 個四面體的衍生型領結 B1 的歐拉特性數，計算如下：

$$VFE(18, 32, 48) - VFE(1, 2, 3) = VFE(17, 30, 45) = 17+30-45=2$$

同樣地，因為相鄰的 2 個四面體不能被截去，我們將僅有 2 個截去 2 個四面體的衍

生型：領結 B2 與領結 B3 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - 2 * \text{VFE}(1, 2, 3) = \text{VFE}(16, 28, 42) = 16 + 28 - 42 = 2.$$



圖八 領結 00、領結 A0 與領結 B0 的衍生型

5.2 眼淚模型與翅膀模型

因為眼淚模型與翅膀模型均沒有六面體的部分，所以僅有眼淚 00 與翅膀 00 兩種基本型，並沒有它們的從屬型。

5.2.1 眼淚 00 衍生型

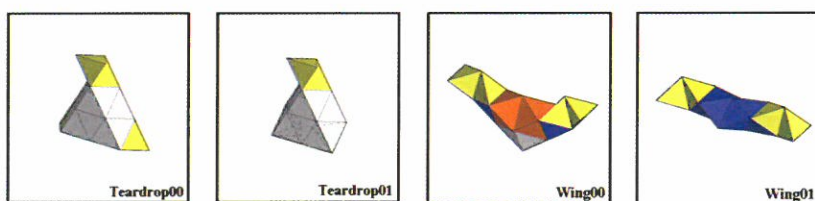
因為眼淚 00 基本型僅有正面正中的對稱線，並且僅有一對四面體，我們僅能截去 2 個四面體。故，僅有一個眼淚 01 衍生型。眼淚 01 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - 2 * \text{VFE}(1, 2, 3) = \text{VFE}(16, 28, 42) = 16 + 28 - 42 = 2$$

5.2.2 翅膀 00 衍生型

因為翅膀 00 的底端有個金字塔體，我們會有一個翅膀 01 衍生型。翅膀 01 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - \text{VFE}(1, 3, 4) = \text{VFE}(17, 29, 44) = 17 + 29 - 44 = 2$$



圖九 眼淚 00 與翅膀 00 的衍生型

5.3 巧克力條模型

巧克力條 00 基本型，具有俯視、側視與前視中心線對稱的特性，並沒有部分是六面體。但是卻有 2 個四面體可以壓擠進去，而形成從屬型巧克力條 A0。

5.3.1 巧克力條 00

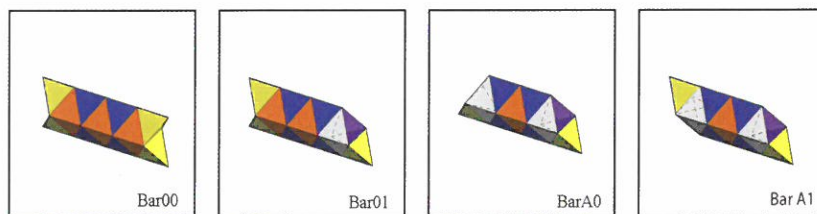
巧克力條 00 具有四個可能壓擠進去的四面體，但在壓擠進去一個四面體的四種相同的情況下，僅能算一種。可稱為巧克力條 01。巧克力條 01 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - \text{VFE}(1, 2, 3) = \text{VFE}(17, 30, 45) = 17 + 30 - 45 = 2.$$

5.3.2 巧克力條 A0

巧克力條 A0 具有四個可能壓擠進去的四面體，但在同時壓擠進去 2 個四面體的四種相同的情況下，僅能算 2 種：巧克力條 A0 與巧克力條 A1。巧克力條 A0 與巧克力條 A1 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - 2 * \text{VFE}(1, 2, 3) = \text{VFE}(16, 28, 42) = 16 + 28 - 42 = 2.$$



圖十 巧克力條 00 與巧克力條 A0 的衍生型

5.4 船型模型

船型 00 基本型，具有俯視、側視與前視中心線對稱的特性，並沒有部分是六面體。但是卻有 2 個四面體可以壓擠進去，而形成從屬型船型 A0。

5.4.1 船型 00

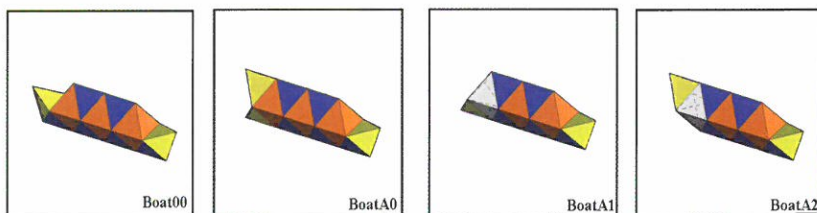
船型 00 具有四個可能壓擠進去的四面體，但在壓擠進去一個四面體的四種相同的情況下，僅能算一種。可稱為船型 01。船型 01 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - \text{VFE}(1, 2, 3) = \text{VFE}(17, 30, 45) = 17 + 30 - 45 = 2.$$

5.4.2 船型 A0

船型 A0 具有四個可能壓擠進去的四面體，但在兩端可以分別壓擠進去 1 個四面體的四種相同的情況下，僅能算 2 種：船型 A1 與船型 A2。船型 A1 與船型 A2 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - \text{VFE}(1, 2, 3) = \text{VFE}(17, 30, 45) = 17 + 30 - 45 = 2$$



圖十一 船型 00 與船型 A0 的衍生型

5.5 星型模型

因為星型基本型星型 00 有四個四面體，我們會有 4 種壓擠四面體的方法。

首先，是壓擠 1 個四面體，但，壓擠進去 1 個四面體的四種相同的情況下，僅能算一種。我們稱為星型 01。星型 01 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - \text{VFE}(1, 2, 3) = \text{VFE}(17, 30, 45) = 17 + 30 - 45 = 2.$$

其次，是壓擠 2 個四面體，但，壓擠進去 2 個四面體的四種相同的情況下，僅能算一種。我們稱為星型 02。星型 02 的歐拉特性數，計算如下：

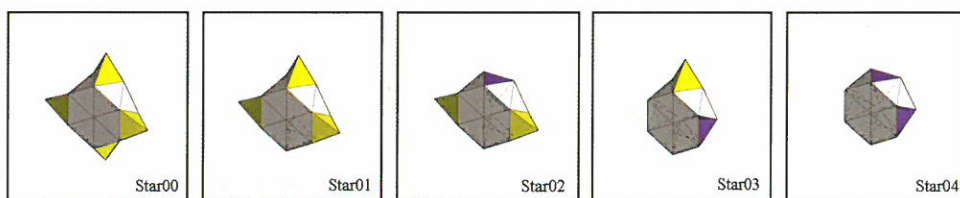
$$\text{VFE}(18, 32, 48) - 2 * \text{VFE}(1, 2, 3) = \text{VFE}(16, 28, 42) = 16 + 28 - 42 = 2.$$

其三，是壓擠 3 個四面體，但，壓擠進去 3 個四面體的四種相同的情況下，僅能算一種。我們稱為星型 03。星型 03 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - 3 * \text{VFE}(1, 2, 3) = \text{VFE}(15, 26, 39) = 15 + 26 - 39 = 2.$$

其四，是壓擠 4 個四面體，僅有唯一的一種。我們稱為星型 04。星型 04 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - 4 * \text{VFE}(1, 2, 3) = \text{VFE}(14, 24, 36) = 14 + 24 - 36 = 2.$$



圖十二 星型 00 的衍生型

5.6 鯉魚模型

鯉魚模型，具有俯視、側視與前視中心線對稱的特性，並且有兩對六面體。故，依照對稱的原則，我們會有 1 個基本型鯉魚 00 與 1 個從屬型鯉魚 A0。鯉魚 A0，是壓擠

一對六面體後創造出的從屬型。我們分別就這 1 個基本型鯉魚 00 與 1 個從屬型鯉魚 A0，討論其衍生型。

5.6.1 鯉魚 00

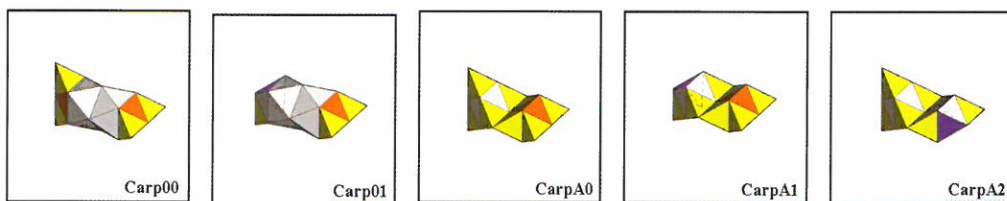
鯉魚 00 的 2 對四面體之中，每一對僅能壓擠 1 個四面體。這會有 2 種可能，但是僅能算一個，即鯉魚 01。鯉魚 01 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - \text{VFE}(1, 2, 3) = \text{VFE}(17, 30, 45) = 17 + 30 - 45 = 2.$$

5.6.2 鯉魚 A0

鯉魚 A0 的 2 對四面體之中，每一對僅能壓擠 1 個四面體。這會有 2 種可能，即鯉魚 A1 與鯉魚 A2。鯉魚 A1 與鯉魚 A2 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - \text{VFE}(1, 2, 3) = \text{VFE}(17, 30, 45) = 17 + 30 - 45 = 2.$$



圖十三 鯉魚 00 與鯉魚 A0 的衍生型

5.7 山形袖章模型

山形袖章模型，具有俯視、側視與前視中心線對稱的特性，並且有一對四面體與一對隱藏的金字塔體。因此我們可以將分為 1 個基本型山形袖章 00 與 3 個從屬型：山形袖章 A0、山形袖章 B0、山形袖章 C0。我們就這 1 個基本型山形袖章 00 與 3 個從屬型：山形袖章 A0、山形袖章 B0、山形袖章 C0，分別討論其衍生型。

5.7.1 山形袖章 00

因為山形袖章 00 具有一對四面體與一對金字塔體，我們可以結合分別壓擠 2 個四

面體與 2 個金字塔體等不同的組合，創造出 7 種衍生型：

首先，壓擠 1 個金字塔體，創造出山形袖章 01。山形袖章 01 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - \text{VFE}(1, 3, 4) = \text{VFE}(17, 29, 44) = 17+29-44=2.$$

其次，壓擠 2 個金字塔體，創造出山形袖章 02。山形袖章 02 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - 2*\text{VFE}(1, 3, 4) = \text{VFE}(16, 26, 40) = 16+26-40=2.$$

其三，壓擠 1 個四面體，創造出山形袖章 03。山形袖章 03 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - \text{VFE}(1, 2, 3) = \text{VFE}(17, 30, 45) = 17+30-45=2.$$

其四，壓擠 2 個金字塔體與 1 個四面體，創造出山形袖章 04。山形袖章 04 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - 2*\text{VFE}(1, 3, 4) - \text{VFE}(1, 2, 3) = \text{VFE}(15, 24, 37) = 2.$$

其五，壓擠 2 個四面體，創造出山形袖章 05。山形袖章 05 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - 2*\text{VFE}(1, 2, 3) = \text{VFE}(16, 28, 42) = 2.$$

其六，壓擠 1 個金字塔體與 2 個四面體，創造出山形袖章 06。山形袖章 06 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - \text{VFE}(1, 3, 4) - 2*\text{VFE}(1, 2, 3) = \text{VFE}(15, 25, 38) = 2.$$

其七，壓擠 2 個金字塔體與 2 個四面體，創造出山形袖章 07。山形袖章 07 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - 2*\text{VFE}(1, 3, 4) - 2*\text{VFE}(1, 2, 3) = \text{VFE}(14, 22, 34) = 2$$

5.7.2 山形袖章 A0

山形袖章 A0 代表具有被擠出 1 個隱藏六面體的山形袖章從屬型組。我們可以結合壓擠 2 個四面體，從而衍生出山形袖章 A1 與山形袖章 A2。我們分別介紹如下：

首先，壓擠 1 個四面體，創造出山形袖章 A1。山形袖章 A1 的歐拉特性數，計算如下：

$$VFE(18, 32, 48) - VFE(1, 2, 3) = VFE(17, 30, 45) = 17+30-45=2$$

其次，壓擠 2 個四面體，創造出山形袖章 A2。山形袖章 A2 的歐拉特性數，計算如下：

$$VFE(18, 32, 48) - 2*VFE(1, 2, 3) = VFE(16, 28, 42) = 16+28-42=2$$

5.7.3 山形袖章 B0

山形袖章 B0 代表具有被擠出 1 個隱藏六面體與 2 個隱藏四面體的山形袖章從屬型組。我們可以結合壓擠 2 個四面體，從而衍生出山形袖章 B1、山形袖章 B2、山形袖章 B3、山形袖章 B4、山形袖章 B5。我們分別介紹如下：

首先，壓擠 1 個四面體，創造出山形袖章 B1。山形袖章 B1 的歐拉特性數，計算如下：

$$VFE(18, 32, 48) - VFE(1, 2, 3) = VFE(17, 30, 45) = 17+30-45=2$$

其次，壓擠 2 個四面體，創造出山形袖章 B2。山形袖章 B2 的歐拉特性數，計算如下：

$$VFE(18, 32, 48) - 2*VFE(1, 2, 3) = VFE(16, 28, 42) = 16+28-42=2$$

其三，壓擠 1 個四面體，創造出山形袖章 B3。山形袖章 B3 的歐拉特性數，計算如下：

$$VFE(18, 32, 48) - VFE(1, 2, 3) = VFE(17, 30, 45) = 17+30-45=2$$

其四，壓擠 2 個四面體，創造出山形袖章 B4。山形袖章 B4 的歐拉特性數，計算如下：

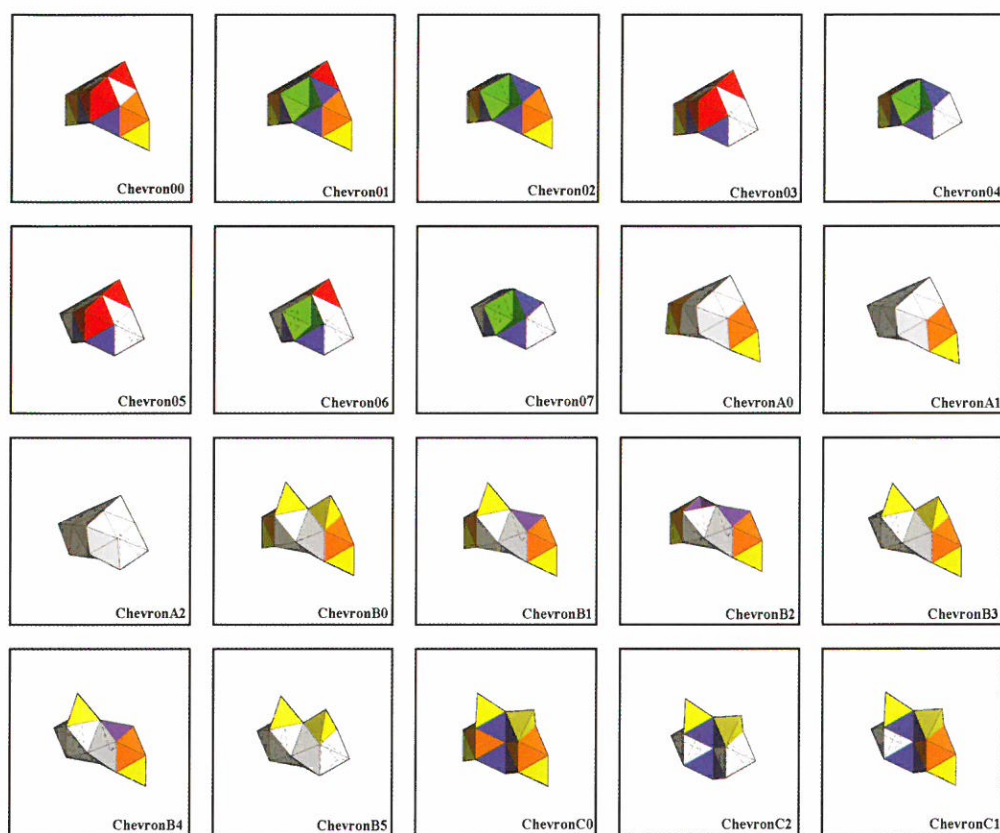
$$VFE(18, 32, 48) - 2*VFE(1, 2, 3) = VFE(16, 28, 42) = 16+28-42=2$$

其五，壓擠 2 個四面體，創造出山形袖章 B5。山形袖章 B5 的歐拉特性數，計算如下：

$$VFE(18, 32, 48) - 2*VFE(1, 2, 3) = VFE(16, 28, 42) = 16+28-42=2$$

5.7.4 山形袖章 C0

山形袖章 C0 代表具有被擠出 1 個隱藏六面體與 2 個隱藏四面體，以及壓擠 2 個六面體的山形袖章從屬型組。我們可以結合壓擠 2 個四面體，從而衍生出山形袖章 C1、山形袖章 C2。我們分別介紹如下：



圖十四 山形袖章 00、山形袖章 A0、山形袖章 B0 與山形袖章 C0 的衍生型

首先，壓擠 1 個四面體，創造出山形袖章 C1。山形袖章 C1 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - \text{VFE}(1, 2, 3) = \text{VFE}(17, 30, 45) = 17 + 30 - 45 = 2$$

其次，壓擠 2 個四面體，創造出山形袖章 C2。山形袖章 C2 的歐拉特性數，計算如下：

$$\text{VFE}(18, 32, 48) - 2 * \text{VFE}(1, 2, 3) = \text{VFE}(16, 28, 42) = 16 + 28 - 42 = 2$$

5.8 拿破崙紙積木製模總結

爲了增加紙積木模塊的數目，我們開發兩個步驟，創造拿破崙紙積木 54 個模型。首先，我們找出 8 個「基本型」，可能含有六面體的屬性，這樣我們就可以創造出可能的「從屬型」。其次，我們將截去所有可能的四面體或是金字塔體，發展出「衍生型」。有關 54 個模型的屬性，請參照表三。

表三 拿破崙紙積木模型基本型、從屬型與其衍生型

模型	基本型/從屬型	屬性	序號	特性	角	面	邊
領結	領結 00	壓擠 4 個四面體	領結 00	0T	18	32	48
			領結 01	1T	17	30	45
			領結 02	2T	16	28	42
			領結 03	2T	16	28	42
	領結 A0	壓擠 2 個六面體與 4 個四面體	領結 A0	0T	18	32	48
			領結 A1	1T	17	30	45
			領結 A2	2T	16	28	42
			領結 A3	2T	16	28	42
	領結 B0	壓擠 2 個六面體與 4 個四面體	領結 B0	0T	18	32	48
			領結 B1	1T	17	30	45
			領結 B2	2T	16	28	42
			領結 B3	2T	16	28	42
翅膀	翅膀 00	壓擠 1 個金字塔體	翅膀 00	0P	18	32	48
			翅膀 01	1P	17	29	44
眼淚	眼淚 00	壓擠 2 個四面體	眼淚 00	0T	18	32	48
			眼淚 01	2T	16	28	42
巧克力條	巧克力條 00	壓擠 4 個四面體	巧克力條 00	0T	18	32	48
			巧克力條 01	1T	17	30	45
	巧克力條 A0	壓擠 4 個四面體	巧克力條 A0	2T	16	28	42
			巧克力條 A1	2T	16	28	42

表三 拿破崙紙積木模型基本型、從屬型與其衍生型（續）

模型	基本型/從屬	屬性	序號	特性	角	面	邊
船型	船型 00	壓擠 2 個四面體	船型 00	0T	18	32	48
	船型 A0	壓擠 1 個四面體	船型 A0	0T	18	32	48
			船型 A1	1T	17	30	45
			船型 A2	1T	17	30	45
星型	星型 00	壓擠 4 個四面體	星型 00	0T	18	32	48
			星型 01	1T	17	30	45
			星型 02	2T	16	28	42
			星型 03	3T	15	26	39
			星型 04	4T	14	24	36
山形袖章	山形袖章 00	壓擠 2 個六面體 與 2 個四面體	山形袖章 00	0P,0T	18	32	48
			山形袖章 01	1P,0T	17	29	44
			山形袖章 02	2P,0T	16	26	40
			山形袖章 03	0P,1T	17	30	45
			山形袖章 04	2P,1T	15	24	37
			山形袖章 05	0P,2T	16	28	42
			山形袖章 06	1P,2T	15	25	38
			山形袖章 07	2P,2T	14	22	34
	山形袖章 A0	壓出六面體 與 2 個四面體	山形袖章 A0	0T	18	32	48
			山形袖章 A1	1T	17	30	45
			山形袖章 A2	2T	16	28	42
	山形袖章 B0	壓出六面體 與 2 個四面體壓擠 2 個四面體	山形袖章 B0	0T	18	32	48
			山形袖章 B1	1T	17	30	45
			山形袖章 B2	2T	16	28	42
			山形袖章 B3	1T	17	30	45
			山形袖章 B4	2T	16	28	42
			山形袖章 B5	2T	16	28	42
	山形袖章 C0	壓出六面體 與 2 個四面體 以及壓擠 2 個六面體 與 2 個四面體	山形袖章 C0	0T	18	32	48
			山形袖章 C1	1T	17	30	45
			山形袖章 C2	2T	16	28	42
鯉魚	鯉魚 00	壓擠 4 個四面體	鯉魚 00	0T	18	32	48
			鯉魚 01	1T	17	30	45
	鯉魚 A0	壓擠 2 個六面體 與 4 個四面體	鯉魚 A0	0T	18	32	48
			鯉魚 A1	1T	17	30	45
			鯉魚 A2	1T	17	30	45

六、虧格數 1^{23} 之模型

在創造出虧格數 0 的 54 個基本型、從屬型與其衍生型之後，我們從而研究連結這些模型後，其虧格數大於或等於 1 的結合體。歐拉特性數為 0 或負數時，虧格數大於或等於 1。但是任何多面體的歐拉特性數都是 2，當 2 個多面體做一個連結面的連結，結合體的歐拉特性數會從 2 個多面體分別的歐拉特性數 2 加 2 之和，減去 2，結果還是 2。但，將 2 個多面體做 2 個連結面的連結，結果會從歐拉特性數 2 加 2 之和，減去 4。經過 2 個連結面的連結的步驟，這個結合體的歐拉特性數將是 0，也就是說它的虧格數是 1。

連結這 2 個連結面的動作，在拓樸圖論來說，就是加入一個把手²⁴，使得虧格數增加 1。這是增加虧格數的一種方法。因此問題就變成如何將 2 個多面體做 2 個連結面的連結？其實這個道理很簡單，我們想像將 1 個甜甜圈²⁵擘成兩半，每一半甜甜圈的歐拉特性數均為 2。但，當我們將半個甜甜圈(把手)與另外半個甜甜圈，經過 2 個連結面的連結，就成了虧格數是 1 的整個甜甜圈。同樣推理，我們假如經過 3 個連結面的連結，就可以做出虧格數是 2，也就是 2 個把手的甜甜圈，有人稱呼為椒鹽卷餅²⁶。如此可推，我們可以製造出虧格數 n 的甜甜圈，到那時候，不再稱呼為甜甜圈，可能是其他的名詞。我們就下列，虧格數 1 領結 02 甜甜圈與虧格數 5 領結 02 甜甜圈，以及虧格數 1 與 4 的巧克力條 A1 菱形，分別驗證 1、5、1 與 4 個把手的 4 個例子說明。

6.1 虧格數 1 領結 02 甜甜圈

我們需要 4 個領結 02，經由兩大步驟，做成甜甜圈。首先，尋找領結 02 的連接面。取 4 個領結 02 的其中 1 個，找到 2 個被截去四面體後，補上的紫色正三角形。然後，順著紫色正三角形的 1 個邊，找到 1 個黃色與 1 個白色正三角形組成的「菱形連結面」。我們將這個領結 02 的「菱形連結面」和另外一個領結 02 的「菱形連結面」重疊，使用 2 小段膠帶將這 2 個領結 02 連結在一起。我們可以看到結合體的兩端，各有

²³ Genus-1

²⁴ handle

²⁵ donut, 甜甜圈是曲面體，輪胎面 (torus) 則是甜甜圈的表面。

²⁶ twist

一個「菱形連結面」。同樣地，將另外一對領結 02 連結在一起，其結合體的兩端，也各有一個「菱形連結面」。

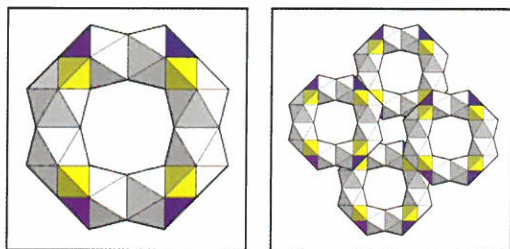
因為 1 個「菱形連結面」的歐拉特性數是 $VFE(4, 2, 5)$ ，當 2 個「菱形連結面」連結在一起時候，將會損失 $VFE(4, 4, 6) = 2$ 。因為領結 02 的歐拉特性數是 $VFE(16, 28, 42) = 2$ ，以及 2 個「菱形連結面」的歐拉特性數會損失 $VFE(4, 4, 6) = 2$ 。因此當我們完成這個步驟的時候，這 2 個結合體的歐拉特性數，分別是 $2 * VFE(16, 28, 42) - VFE(4, 4, 6) = VFE(32, 56, 84) - VFE(4, 4, 6) = VFE(28, 52, 78) = 28 + 52 - 78 = 2$ ，仍然是 2。

其次，連結 2 個結合體，2 對「菱形連結面」。我們使用 4 小段膠帶，將這 2 個結合體兩端的 2 個「菱形連結面」分別重疊並且連接在一起。因為 2 個結合體的歐拉特性數分別是 $VFE(28, 52, 78) = 2$ ，以及 2 個「菱形連結面」的歐拉特性數分別會損失 $VFE(4, 4, 6) = 2$ ，因此當我們完成這個步驟的時候，這 2 個結合體的歐拉特性數，分別是 $2 * VFE(28, 52, 78) - 2 * VFE(4, 4, 6) = VFE(56, 104, 156) - VFE(8, 8, 12) = VFE(48, 96, 144) = 48 + 96 - 144 = 0$ ；根據歐拉-龐加萊公式：歐拉特性數 $= 2 - 2g = 0$ ，即虧格數 g 等於 1。請參照圖十五的左圖虧格數 1 領結 02 甜甜圈。

6.1.1 虧格數 5 領結 02 甜甜圈

為了做虧格數 5 甜甜圈，我們需要四個上述虧格數 1 甜甜圈。首先，我們必須要找到虧格數 1 甜甜圈與另一個虧格數 1 甜甜圈對應的「正三角形連結面」。因為一個「正三角形連結面」的歐拉特性數是 $VFE(3, 1, 3)$ ，當兩個「正三角形連結面」連結在一起時候，將會損失 $VFE(3, 2, 3) = 2$ 。因為虧格數 1 甜甜圈的歐拉特性數是 $VFE(48, 96, 144) = 0$ ，以及 2 個「正三角形連結面」歐拉特性數會損失 $VFE(3, 2, 3) = 2$ ，因此當我們完成這個步驟的時候，這個結合體的歐拉特性數，是 $2 * VFE(48, 96, 144) - VFE(3, 2, 3) = VFE(93, 190, 285) = -2$ ；根據歐拉-龐加萊公式：歐拉特性數 $= 2 - 2g = -2$ ，即虧格數 g 等於 2，稱呼為虧格數 2 結合體。

其次，連結 2 個虧格數 2 結合體的 2 個「正三角形連結面」。因為 2 個「正三角形連結面」連結在一起時候，將會損失 $VFE(3, 2, 3) = 2$ 。因為虧格數 2 結合體的歐拉特性數是 $VFE(93, 190, 285) = -2$ ，以及 2 對中每 2 個「正三角形連結面」歐拉特性數會損失 $VFE(3, 2, 3) = 2$ ，因此當我們完成這個步驟的時候，這個結合體的歐拉特性數，是 $2 * VFE(93, 190, 285) - 2 * VFE(3, 2, 3) = VFE(180, 376, 564) = -8$ ；根據歐拉-龐加萊公式：歐拉特性數 $= 2 - 2g = -8$ ，即虧格數 g 等於 5，簡稱為虧格數 5，即 5 個把手的甜甜圈，全稱為虧格數 5 領結 02 甜甜圈，請參照圖十五的右圖。



圖十五 虧格數 1 領結 02 甜甜圈與虧格數 5 領結 02 甜甜圈

6.2 虧格數 1 巧克力條 A1 菱形

我們需要 4 個巧克力條 A1，經由兩大步驟，做成虧格數 1 的菱形。首先，尋找巧克力條 A1 的連接面。取 4 個巧克力條 A1 的其中 1 個，找到 2 個其中的 1 個被截去四面體後，補上的紫色正三角形。然後，順著紫色正三角形的 1 個邊，找到 1 個黃色與 1 個白色正三角形組成的「菱形連結面」。我們將這個巧克力條 A1 的「菱形連結面」和另外一個巧克力條 A1 的「菱形連結面」重疊，使用 2 小段膠帶將這 2 個巧克力條 A1 連結在一起。我們可以看到結合體的兩端，各有 1 個「菱形連結面」。同樣地，將另外一對巧克力條 A1 連結在一起，其結合體的兩端，也各有 1 個「菱形連結面」。

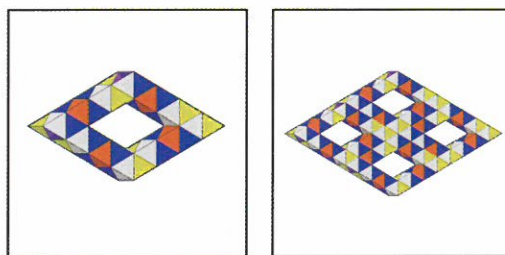
因為 1 個「菱形連結面」的歐拉特性數是 $VFE(4, 2, 5)$ ，當兩個「菱形連結面」連結在一起時候，將會損失 $VFE(4, 4, 6) = 2$ 。因為巧克力條 A1 的歐拉特性數是 $VFE(16, 28, 42) = 2$ ，以及 2 個「連結面菱形」的歐拉特性數會損失 $VFE(4, 4, 6) = 2$ ，因此當我們完成這個步驟的時候，這 2 個結合體的歐拉特性數，分別是 $2 * VFE(16, 28, 42) - VFE(4, 4, 6) = VFE(32, 56, 84) - VFE(4, 4, 6) = VFE(28, 52, 78) = 28 + 52 - 78 = 2$ ，仍然是 2。

其次，連結 2 個結合體兩端，各有一個「連結面菱形」。我們使用 4 小段膠帶，將這 2 個結合體兩端的 2 個「連結面菱形」分別重疊並且連接在一起。因為 2 個結合體的歐拉特性數分別是 $VFE(28, 52, 78) = 2$ ，以及 2 個「連結面菱形」的歐拉特性數分別會損失 $VFE(4, 4, 6) = 2$ ，因此當我們完成這個步驟的時候，這 2 個結合體的歐拉特性數，分別是 $2 * VFE(28, 52, 78) - 2 * VFE(4, 4, 6) = VFE(56, 104, 156) - VFE(8, 8, 12) = VFE(48, 96, 144) = 48 + 96 - 144 = 0$ ；根據歐拉-龐加萊公式：歐拉特性數 $= 2 - 2g = 0$ ，即虧格數 g 等於 1，從而完成虧格數 1 菱形，請參照圖十六的左圖。

6.2.1 虧格數 4 巧克力條 A1 菱形

爲了做虧格數 4 菱形，我們需要四個上述虧格數 1 菱形。首先，我們必須要找到虧格數 1 菱形與另一個虧格數 1 菱形對應的 8 個正三角形組²⁷成的「長條連結面」。因爲一個「長條連結面」的歐拉特性數是 $VFE(10, 8, 17)$ ，當兩個「長條連結面」連結在一起時候，將會損失 $VFE(10, 16, 24) = 2$ 。因爲虧格數 1 菱形的歐拉特性數是 $VFE(48, 96, 144) = 0$ ，以及 2 個「長條連結面」歐拉特性數會損失 $VFE(10, 16, 24) = 2$ ，因此當我們完成這個步驟的時候，這個結合體的歐拉特性數，是 $2 * VFE(48, 96, 144) - VFE(10, 16, 24) = VFE(86, 176, 264) = -2$ ；根據歐拉-龐加萊公式：歐拉特性數 $= 2 - 2g = -2$ ，即虧格數 g 等於 2，稱呼爲虧格數 2 結合體。

其次，連結 2 個虧格數 2 結合體的長邊。這個長邊，是由 16 個正三角形組²⁸成的「2 倍長條連結面」。因爲 1 個「2 倍長條連結面」是 2 個「長條連結面」扣除損失的 $VFE(2, 0, 1)$ 所組成，即 $2 * VFE(10, 8, 17) - VFE(2, 0, 1) = VFE(18, 16, 33)$ 。當 2 個「2 倍長條連結面」連結在一起時候，將會損失 $VFE(18, 32, 48) = 2$ 。因爲虧格數 2 結合體的歐拉特性數是 $VFE(86, 176, 264) = -2$ ，以及 2 個「2 倍長條連結面」歐拉特性數會損失 $VFE(18, 32, 48) = 2$ ，因此當我們完成這個步驟的時候，這個結合體的歐拉特性數，是 $2 * VFE(86, 176, 264) - VFE(18, 32, 48) = VFE(154, 320, 480) = -6$ ；根據歐拉-龐加萊公式：歐拉特性數 $= 2 - 2g = -6$ ，即虧格數 g 等於 4，簡稱爲虧格數 4 菱形，全稱爲虧格數 4 巧克力條 A1 菱形，請參照圖十六的右圖。



圖十六 虧格數 1 與 4 巧克力條 A1 菱形

²⁷ 8-iamond

²⁸ 16-iamond

七、結 論

拿破崙平面密鋪，是由包含六個三角形的平面密鋪組，被簡化與美化，最終創造出六邊鼓形、八邊三葉形與八邊雁形等三個拿破崙平面密鋪。因為八邊三葉形為等邊對稱的附帶 α 角的八邊形圖案，我們稱呼它為 α -拿破崙圖案。我們討論許多有關 α -拿破崙圖案之有趣特性時候，發現它能夠創造出一個立體的表面被一個週期平面密鋪包裹，亦即拿破崙立體密鋪。我們發現一個更有趣的事實，就是 120° - 拿破崙圖案，不僅擁有所有 α 角範圍內最大的面積，並且可以簡化成為 32 個正三角形組成的大菱形。這就是拿破崙紙積木的開端。我們發現有四種方法建構領結、翅膀、眼淚、巧克力條、船型、星型、山形袖章與鯉魚等 8 種拿破崙立體模型。

為了增加紙積木模塊的數目，我們開發兩個步驟，有系統地創造出 54 種包括基本型、從屬型、與衍生型等模型。不但，方塊的數目增加了，我們能夠結合這些紙積木，創作不同的結合體。這些結合體包括有把手的立體模型。由於，這些模型均有角、面、邊三個要件的特點，我們可以計算出多面體的歐拉特性數。藉由歐拉特性數之計算，我們能夠驗證所創造出 1 個把手或多於 1 個把手的結合體，亦即虧格數為 1 或是大於 1 的結合體。紙積木，還有拆開重覆使用與可著色的特點。當孩子不喜歡的造型，可以拆開重來，甚至基本造型也可以，這是木質積木或是塑膠積木所不能取代的。最後，值得一提的是，不喜歡的模型可以重做，沒有儲存的問題。

八、參考文獻

- [1] Bondy, J. A. and Murty, U. S. R. (1976), "Graph Theory with Applications", Elsevier
- [2] COMAP, Inc., (2005), For All Practical Purposes Mathematical Literacy in Today's World, Seventh Edition, COMAP, Inc. (The Consortium for Mathematics and its Applications)
- [3] Demaine, Erik D. and O'Rourke Joseph, (2005), A Survey of Folding and Unfolding in Computational Geometry, in Combinatorial and Computational Geometry, edited by Jacob E. Goodman, Janos Pach, and Emo Welzl, Mathematical Sciences Research Institute Publications, volume 52, August 2005, pages 167-211, Cambridge University Press.
- [4] Fuse, Tomoko, (1990), Unit Origami- Multidimensional Transformations, Tokyo and New York: Japan Publication, Inc. p.228.

- [5] Katz, Lilian, (1978), Principles of staff development in programs for young children, Children in Contemporary Society, 12(2), 1-4 ED160229
- [6] Lu, Keh-Ming, (2005), A Study of Innovative Plane Tessellation Patterns Systems, Fu Ren Studies (Science and Engineering), Taipei: Fu Ren Catholic University, College of Science and Engineering, Vol. 39, pp. 111-140.
- [7] Lu, Keh-Ming, Tsaur, Alan S, Bai, Jia-Xiang (2006) "Taotie Tessellation," CGW2006 (Computer Graphics Workshop), Soochow University
- [8] Parthasarathy, K.R., (1994), Basic Graph Theory, India: Tata McGraw-Hill Publishing Company Limited, pp. 3-5
- [9] Penrose, Roger, (1989), The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics, Oxford University Press.
- [10] Rybczynski, Witold, (1992), Looking Around: A Journey Through Architecture (Paperback) Penguin Books USA, Inc.

received October 30, 2006

revised November 29, 2006

accepted December 9, 2006

A Study of Napoleon Paper Building Blocks

Keh-Ming Lu

Department of Computer Science and Information Engineering, Asia University

Alan S. Tsaur

Department of Information and Design, Asia University

Mei-Fen Chen

Department of Computer Science and Information Engineering, Asia University

Abstract

A general concept of building blocks is of 3D geometric graphics presented by their different appearance. The shape of each individual building block is initially confined. Constructors must conceive the design pattern and then select different building blocks to construct the patterns. Therefore, converting a rhombic tessellation motif into a 3D tessellation model is an essential goal of paper building blocks. Reusability and capability of coloring are two specific features of paper building blocks. Combinations of building blocks including the prime models are reusable. These important features are irreplaceable by wooden and plastic building blocks. The creativity is productivity and an innovative force as well.

This study starts with description of the basics of Napoleon plane tessellation and a development of 8 basic building blocks such as Bowtie, Wing, Teardrop, Bar, Boat, Star, Chevron, and Carp that are made of a rhombic 32-diamond through four construction processes. In order to increase the number of types of building blocks, we developed two steps to create 54 types of primary, secondary, and derivative models. We also create innovative models including handles by combining these building blocks.

Based on a polyhedron's numbers of vertices, faces, and edges, we can calculate its Euler characteristic. Furthermore we construct and verify the genus- n combinatorial models.

Key Words: Napoleon motif, Euler-Poincare formula, Euler characteristic, genus

Active simulation of grounded inductor using CFCCIIs

Yung-Chang Yin

*Department of Electronic Engineering, Fu Jen Catholic University,
Taipei 242, Taiwan, R.O.C.*

Yung-Chwan Yin

*Department of Animation Design and Game Programming, TOKO University,
Chia-Yi 613, Taiwan, R.O.C*

Abstract

The simulation of grounded inductor using two four-terminal active current conveyors (CFCCIIs) and three passive elements is presented. Proper selection of the values of the passive elements can yield very large as well as very small of simulated inductances. The grounded inductor simulated can be used in the LCR passive filters. Thus, these filters are suitable for monolithic implementation. Meanwhile, their advantages include low component sensitivities and the ability to utilize the extensive knowledge of LCR filter design. Finally, the experimental result of a RLC passive bandpass filter confirming the theory is included.

Key words: four-terminal active current conveyor

I. INTRODUCTION

In recent years, four-terminal active current conveyor (CFCCII) has become very popular because of its high signal bandwidth, great linearity and large dynamic range [1][2]. Many applications in the realization of various filter functions using some CFCCIIs have been published

*Corresponding author.

E-mail:yin@ee.fju.edu.tw

[3]-[14]. However, the inductance-simulation circuit has never been presented. In this paper, two CFCCII and three passive elements are used to design simulated grounded inductance.

In the area of active filter design, inductor simulation has attracted considerable interest, because the advantage of designing active filters by simulating the inductor of a passive LCR realization of the filter include low component sensitivities and the ability to utilize the extensive knowledge of LCR filter design. In addition to the above advantages, the CFCCII can also circumvent the finite gain-bandwidth limitation of the conventional operational amplifier. Meanwhile, it can also offer both a constant bandwidth and a high slew rate (i.e. 2000 V/us) [3] [4]. On the other hand, the grounded inductor is also frequently required for the design of analogue IC building blocks. It is, therefore, attractive to use CFCCII-based circuits for simulating ground inductors. In this paper, a grounded inductance simulator using two CFCCII and three passive elements was constructed. This inductance simulator was used to RLC passive filters to enjoy the advantages of the RLC filters. Finally, the RLC passive bandpass filter using the proposed simulated inductor is experimentally verified.

II. CIRCUIT DESCRIPTION

The circuit symbol for a CFCCII is shown in Fig.1. The port relations of a CFCCII can be characterized as $i_z = \pm i_x$, $v_z = v_x$, $i_y = 0$ and $i_0 = i_x$. The '+' and '-' signs of the current i_z denote the non-inverting and inverting, respectively. The proposed simulation of grounded inductor circuit is shown in Fig. 2. Using the standard notations of ideal CFCCII, it is easy to show that the input impedance of the circuit of Fig. 2 can be expressed as

$$Z_{in} = \frac{Z_1 Z_2}{Z_3} \quad (1)$$

where $Z_1 - Z_3$ are the impedances. If the impedances are chosen as $Z_1 = R_1$, $Z_2 = R_2$, and $Z_3 = 1/SC$ shown in Fig.3. This is equivalent to an inductor with inductance L_{eq} given by

$$L_{eq} = R_1 R_2 C \quad (2)$$

From the above equation, it can be seen that by properly selecting values of the resistors R_1 , R_2 and the capacitor C , very large as well as very small values of inductance can be easily

obtained. Clearly, the grounded inductance simulator is constructed.

III. EXPERIMENTAL RESULTS

The AD844 can be used to construct as a CFCCII. The proposed grounded inductor simulated was experimentally tested using $R_1 = R_2 = 1K\Omega$, $C = 1\mu F$ and AD844s, so this is equivalent to an inductor with $L_{eq} = 1H$. Then, this grounded inductor simulated was used in the realization of the RLC bandpass filter shown in Fig.4 (a), and the transfer function has a biquadratic bandpass characteristic with

$$\frac{v_{out}}{v_{in}} = \frac{(\frac{1}{CR})^s}{s^2 + s(\frac{1}{CR}) + \frac{1}{L_{eq}C}} \quad (5)$$

the central frequency: $\omega_o = (\frac{1}{L_{eq}C})^{1/2}$

the quality factor: $\vartheta = R(\frac{C}{L_{eq}})^{1/2}$

From Fig.4 (b), a biquadratic bandpass filter was constructed with $R = 10^3\Omega$, $C = 1\mu F$ and $L_{eq}=1H$. The resonance frequency was monitored and measured and the corresponding inductance value was calculated and compared with the theoretical value calculated using equation (2). The experimental results for the gain and phase responses shown in Fig.5 (a) (b) show a good agreement between theoretical calculations and practical measurements. The theoretical analysis correlated with the measured results with few errors which due to the errors of the use of passive elements.

IV. CONCLUSION

The realization of grounded inductor using two four-terminal active current CFCCII and three passive elements has been proposed. A proper selection of the values of the resistors and

capacitor, of Fig. 2, can yield very large as well as very small of simulated inductances. The advantages of designing active filters by simulating the inductors of a passive LCR realization of the filter include low component sensitivities and the ability to utilize the extensive knowledge of LCR filter design. Meanwhile, these filters are suitable for monolithic implementation. Finally, the experimental grounded inductors results of a LCR bandpass passive filter confirmed the theoretical analysis.

REFERENCES

- [1] B.Wilson, "Recent Developments in Current Conveyors and Current-Mode Circuits", *IEE Proc-G*, vol. 137, no. 2, pp.63-77, 1990.
- [2] G.W.Rober and A.S.Sedra, "All Current-Mode Frequency Selective Circuits", *Electron. Lett.*, Vol.25, pp.759-761, 1989.
- [3] C.Toumazou and E.J.Lidgey, "Universal active filter using current conveyors", *Electron. Lett.*, vol. 22, pp.662-664, 1986.
- [4] Y.Sun and J.K.Fidler, "Versatile active biquad based on second-generation current conveyors", *Int. J Electron.*, vol. 76, pp91-98, 1994.
- [5] V.K.Singh and R.Senani, "New multifunction active configuration employing current conveyors", *Electron. Lett.*, vol. 26, pp.1814-1816, 1990.
- [6] G.W.Robert and A.S.Sedra, "A general class of current amplifier-based biquadratic filter circuits", *IEEE Transactions on Circuits and Systems.*, vol. 39, pp.257-263, 1992.
- [7] C.M.Chang, "Universal active current filters using single second-generation current conveyors", *Electron. Lett.*, vol. 27, no.18, pp.1614-1617, 1991
- [8] C.M.Chang and P.C.Chen, "Universal active current filter with three inputs and one output using current conveyors", *Int. J Electron.*, vol. 71, no.5, pp.817-819, 1991.
- [9] R. Senani, "New current-mode biquad filter", *Int. J Electron.*, vol. 73, no. 4, pp.735-742, 1992.
- [10] Y.C. Yin, "Current-Mode Biquad Using Two CFCCIIps", *Fu Jen Studies.*, vol. 3, no. 4, pp.367-370, 2003.
- [11] Y.C.Yin, Y.C. Liou "Realization of Current-Mode Highpass Lowpass and Bandpass Biquad Filters using Single CFCCIIp", *Fu Jen Studies*, No.38, 2004, pp.89-99

- [12] Y.C. Yin, "Realization of Current-Mode Notch and Allpass Filters using Single CFCCII", *Fu Jen Studies*, No.39, 2005, pp.11-22.
- [13] E.O.Gues and F. Anday, "Realization of current-mode universal filter using CFCCIIps ", *Electron. Lett.*, vol. 32, pp.1081-1082 , 1996.
- [14] C.M. Chang and S.H. Tu, "Universal current-mode filters employing CFCCIIps", *Int. J Electron.*, vol. 85, no.6, pp.749-754 , 1998.



Fig.1 A CFCCII symbol

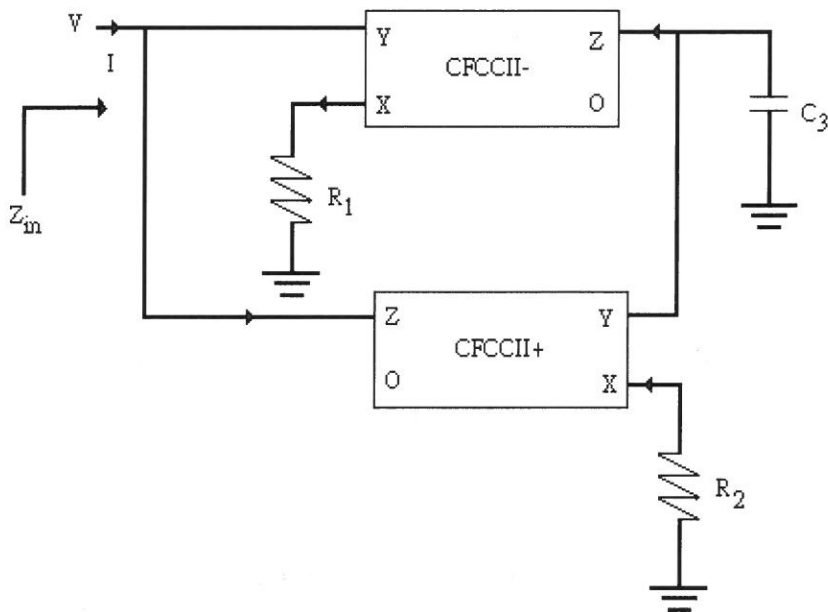


Fig.2 The proposed ground impedance simulator.

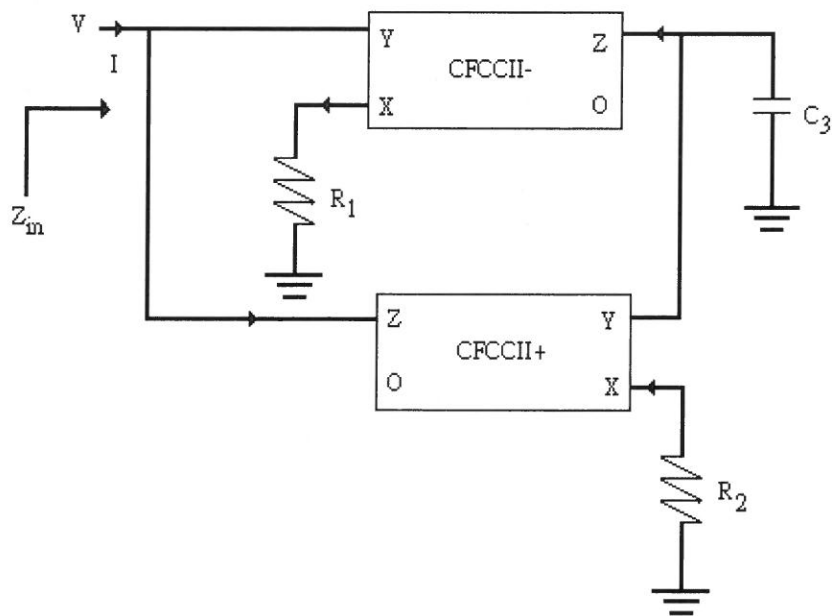
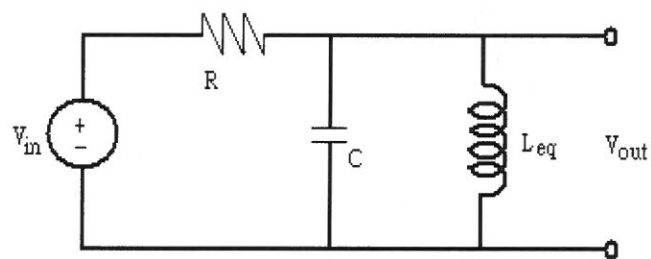
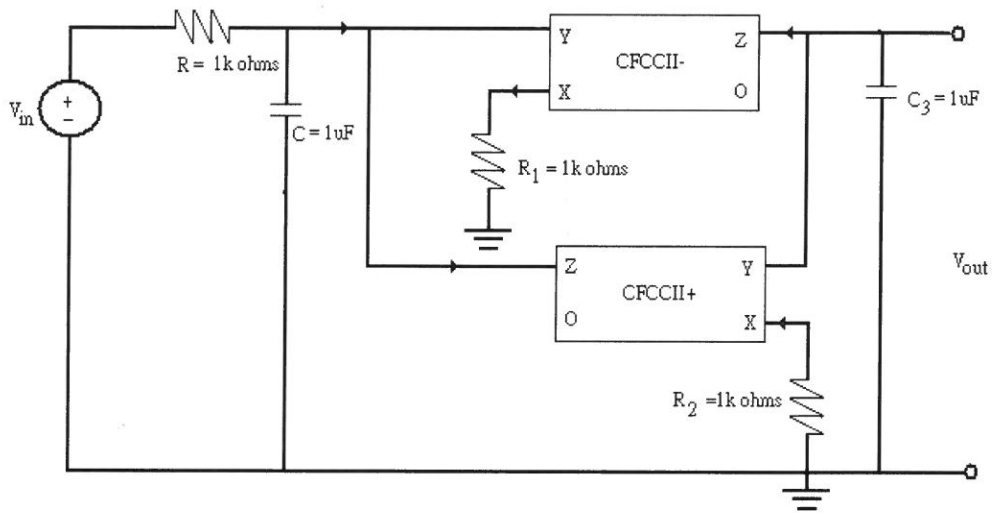


Fig.3. Grounded inductance simulator



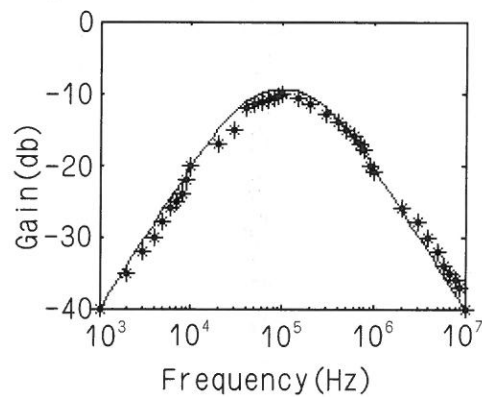
(a) The prototype passive RLC bandpass filter



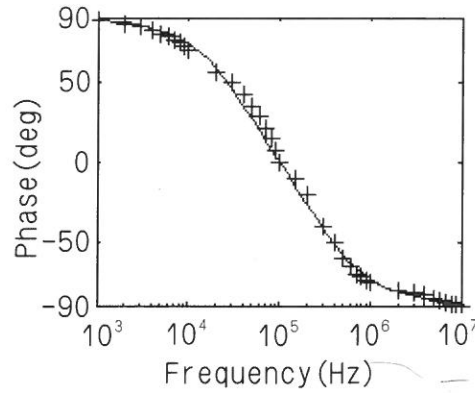
(b) Bandpass filter circuit used to test the inductor realized using circuit of Fig. 3

Fig.4 (a) : The prototype passive bandpass filter

(b) : Bandpass filter circuit used to test the inductor realized using circuit of Fig. 3



(a) The gain response curve of the bandpass filter using simulated inductor.



(b) The phase response curve of the bandpass filter using simulated inductor.

Fig.5 (a) : The gain response curve of the bandpass filter using simulated inductor.

(b) : The phase response curve of the bandpass filter using simulated inductor.

*: Experimental result for bandpass gain

+: Experimental result for bandpass phase

—: Ideal curve

received October 15,2006

revised December 2,2006

accepted December 14, 2006

使用四端主動電流傳輸器合成主動電感模擬電路

鄧永昌

輔仁大學大學電子工程學系

鄧永傳

稻江科技暨管理學院動畫與遊戲軟體設計學系

中文摘要

本文提出使用兩個四端主動電流傳輸器和三個被動元件合成接地電感模擬電路。此合成接地電感模擬電路，只要選擇適當的電阻值與電容值，就可以得到非常大或非常小的模擬接地電感值。電感模擬電路的好處是，可以用來取代現有被動濾波器的電感值，使其可以被製作成積體電路，同時被動濾波器既有的好處，在被轉換成主動濾波器後，這些好處仍可繼續存在。最後，以既有的二階被動帶通濾波器，應用電感模擬電路取代其電感值，以驗證本文之理論預測。

關鍵詞：四端主動電流傳輸器

Uniform Consistency of Kernel Density Estimator under α -mixing

Wann-Jyi Horng

*Department of Finance, Ling-Tung University,
No. 1 Ling-Tung Road, Nantun, Taichung, Taiwan.*

Abstract

A limiting behavior and convergence rate of a kernel density estimator under α -mixing condition are studied. As the domain of density function is compactly supported, we know that the traditional density estimator will encounter the problem of the boundary effects. In order to improve the limiting behavior and convergence rate, one follows the idea of linear-fit method to construct a new weighted kernel density estimator. In this paper, the limiting behavior, the convergence rate and some properties of the proposed estimator are given. The proposed estimator does not need to improve on the boundary regions under the condition of α -mixing, and its convergence rate is achieved, $O(n^{-\frac{p+1}{2p+4}}(\log \log n)^{\frac{p+1}{2p+4}})$ for all $p \geq 1$.

Keywords and Phrases: kernel density estimator, boundary effects, bandwidth, uniform consistency, convergence rates, bias reduction, α -mixing.

1. Introduction

Let X_1, \dots, X_n be α -mixing sequence of random variables. X_1 has a distribution function $F(x)$ and probability density function $f(x)$ and mixing coefficient $\alpha(n)$, $\alpha(n) \rightarrow 0$ as $n \rightarrow \infty$. The α -

*Corresponding author. Tel: +886-2-29052451

E-mail: hwj@mail.ltu.edu.tw

mixing coefficients of the sequence $\{X_k\}$ of random variables is defined by

$$\alpha(n) = \sup_{k=1,2,\dots} \sup \left\{ |P(A \cap B) - P(A)P(B)| : A \in \mathfrak{F}_1^k, B \in \mathfrak{F}_{n+k}^\infty \right\},$$

where $\mathfrak{F}_m^n = \sigma(X_t : m \leq t < n)$, $-\infty \leq m < n \leq \infty$, and σ is the σ algebra.

In the literature, various estimators for $f(x)$, based on a random sample X_1, \dots, X_n , have been proposed and their properties are studied, for examples, the monographs by Wand and Jones (1995), Simonoff (1996), Silverman (1986) and Prakasa Rao (1983). An ordinary kernel estimate for f is defined as follows:

$$\hat{f}_{rp}(x; h) = \frac{1}{nh} \sum_{j=1}^n K \left\{ \frac{x - X_j}{h} \right\}, \quad (1)$$

where K is a (symmetric) kernel function and $h = h(n)$ is called the bandwidth.

This estimator (1) is called the kernel density estimator (Parzen, 1962). The studies of \hat{f}_{rp} can refer to, for examples, Kim and Lee (2004), Cai and Roussas (1992), Kim and Cox (1996), Schuster (1969) and Stone (1982). When point x is near the boundary of their support, the kernel density estimator (KDE) (1) has suffered a serious problem of boundary effects. Some boundary modification methods have been proposed by, for examples, Jones and Foster (1996), Muller (1993) and Jones (1993).

In this paper, we will relax the constraint that the density estimator has to be a bona fide density, and propose a new weighted KDE which does not encounter boundary effects. The proposed method uses the idea of linear fit. The method of linear fit is assumed as follows:

$$\text{Min } \varepsilon' \varepsilon = \text{Min}_{\bar{\beta}} (\bar{t} - A_{p+1} \bar{\beta})' (\bar{t} - A_{p+1} \bar{\beta}),$$

where $\bar{t}' = (t_0, t_1, \dots, t_p)$, $\bar{\beta}' = (\beta_0, \beta_1, \dots, \beta_p)$, $\varepsilon = (\bar{t} - A_{p+1} \bar{\beta})$,

$$t_k = 1/nh \sum_{i=1}^n K \{ (x - X_i)/h \} ((x - X_i)/h)^k,$$

β_0, \dots, β_p are the unknown parameters and A_{p+1} is given as below, for all

$k=0,1,2,\dots,p$ and $p \geq 1$.

According to linear fit method, we minimize $\varepsilon'\varepsilon$ with respect to $\vec{\beta}$, denoting the solution of $\vec{\beta}$ by $\hat{\vec{\beta}}$. Further to calculate and use the Cramer's rule for $\hat{\beta}_0$. Denoting $\hat{\beta}_0 = \hat{f}_{p+1}$, we can get a new weighted kernel density estimator which is given by

$$\begin{aligned}\hat{f}_{p+1}(x;h) &= \frac{1}{nh} \sum_{j=1}^n \frac{\det(U_{p+1}(\frac{x-X_j}{h}))}{\det(A_{p+1})} K\left\{\frac{x-X_j}{h}\right\} \\ &= \frac{1}{nh} \sum_{j=1}^n W\left\{\frac{x-X_j}{h}\right\},\end{aligned}\quad (2)$$

where, $W(u) = \frac{\det(U_{p+1}(u))}{\det(A_{p+1})} K(u)$, A_{p+1} and $U_{p+1}(\cdot)$ are $(p+1) \times (p+1)$ matrices, $\det(\cdot)$ denotes the determinant. Here A_{p+1} and $U_{p+1}(\cdot)$ are defined as below:

$$A_{p+1} = \begin{bmatrix} S_0 & S_1 & S_2 & \cdots & S_p \\ S_1 & S_2 & S_3 & \cdots & S_{p+1} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ S_p & S_{p+1} & S_{p+2} & \cdots & S_{2p} \end{bmatrix},$$

and

$$U_{p+1}\left(\frac{x-X_j}{h}\right) = \begin{bmatrix} 1 & S_1 & \cdots & S_p \\ (x-X_j)/h & S_2 & \cdots & S_{p+1} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ ((x-X_j)/h)^p & S_{p+1} & \cdots & S_{2p} \end{bmatrix},$$

$$S_j = \frac{1}{h} \int K \left\{ \frac{x-y}{h} \right\} \left(\frac{x-y}{h} \right)^j dy, \quad (3)$$

for all $j=0, 1, \dots, 2p$ and $p \geq 1$.

Here we use the estimator of \hat{f}_{p+1} to estimate $f(x)$, for all $p \geq 1$. We will study the limiting behavior and convergence rates of estimator (2) which does have the uniformly consistency property.

From (2), we know that it does have a nice property which is given by

$$\frac{1}{h} \int K \left\{ \frac{x-u}{h} \right\} \det(U_{p+1}(\frac{x-u}{h})) (\frac{x-u}{h})^m du = 0, \quad (4)$$

for $m=1, 2, \dots, p \geq 1$. We know that the above result by the linear algebra theory.

This paper is organized as follows: in section 2, we state the main results of estimator (2) and give the explicit formula for convergence rates. In section 3, we give the proofs of the main results.

2. Main results

In this section, one states the main results of estimator (2). Before one states the main results one will give the following assumptions:

(A1) X_1, \dots, X_n is α -mixing sequence with the probability density function $f(x)$, $f(x)$ is bounded on its domain, and continuous at the point x and $f^{(m)}(x)$ exists, for $m \geq p+1$ and $p \geq 1$. Suppose that f satisfies the Lipschitz condition

$$|f(x) - f(y)| \leq C|x - y|, \text{ } C \text{ is a constant.}$$

(A2) $W(u) = \frac{\det(U_{p+1}(u))}{\det(A_{p+1})} K(u)$ is bounded variation. $K(\cdot)$ is the kernel function

and satisfies

$$\limsup_{|u| \rightarrow \infty} \left| u^{k+p+4} K(u) \right| < \infty, \text{ for each } k \geq 0 \text{ and } p \leq 1.$$

(A3) $\sum_{n=1}^{\infty} \frac{\log n}{n} (\log \log n)^{1+r} \alpha(n)$ converges for some $r > 0$, the condition of Cai and Roussas (1992).

Let us now give the main results of the estimator (2). In the following Theorem 2.1, we give the strong convergence of the estimator (2) at the point $x \in (-\infty, \infty)$.

Theorem 2.1. Assume that conditions (A1)–(A3) is satisfied. And suppose that $h \rightarrow 0$ and $nh \rightarrow \infty$, as $n \rightarrow \infty$, then we have

$$\sup_{x \in (-\infty, \infty)} \left| \hat{f}_{p+1}(x; h) - f(x) \right| \rightarrow 0, \quad (5)$$

almost surely, as $n \rightarrow \infty$.

Let $[a, b]$ be the compact support of $f(x)$. In the following Theorem 2.2, one give the strong convergence of the estimator (2) at the point $x \in [a, b]$, for example, $a=0$ and $b=1$.

Theorem 2.2. Assume that conditions (A1)–(A3) are satisfied. And suppose that $h \rightarrow 0$ and $nh \rightarrow \infty$, as $n \rightarrow \infty$, then we have

$$\sup_{x \in [a, b]} \left| \hat{f}_{p+1}(x; h) - f(x) \right| \rightarrow 0, \quad (6)$$

almost surely, as $n \rightarrow \infty$.

Remark 1: The estimator (2) has the strong convergence property as that of the traditional kernel density estimator as $p=1$. As the domain of $f(x)$ is compactly supported, the strong convergence of $\hat{f}_{p+1}(x; h)$ is also held. The proposed estimator does not have the problem of boundary effects, for details, refer to Fan and Gijbels paper (1992).

If $f^{(p+1)}(x)$ is uniformly continuous function on $(-\infty, \infty)$, then, we have the following theorem about the convergence rate of the estimator $\hat{f}_{p+1}(x; h)$.

Theorem 2.3. Under the assumptions of Theorem 2.1 and assume that $f^{(p+1)}(x)$ is uniformly continuous, then we have

$$\sup_{x \in (-\infty, \infty)} \left| \hat{f}_{p+1}(x; h) - f(x) \right| = O\left(h^{p+1} + \frac{\sqrt{\log \log n}}{\sqrt{nh}}\right), \quad (7)$$

for $p \geq 1$. As $h = O\left(\left(\frac{n}{\log \log n}\right)^{-\frac{1}{2p+4}}\right)$, we also have

$$\sup_{x \in (-\infty, \infty)} \left| \hat{f}_{p+1}(x; h) - f(x) \right| = O\left(n^{-\frac{p+1}{2p+4}} (\log \log n)^{\frac{p+1}{2p+4}}\right). \quad (8)$$

If $f^{(p+1)}(x)$ is an uniformly continuous function on $[a, b]$, then we have the following theorem about the convergence rate of the estimator $\hat{f}_{p+1}(x; h)$.

Theorem 2.4. Under the assumptions of Theorem 2.2 and assume that $f^{(p+1)}(x)$ is uniformly continuous, then we have

$$\sup_{x \in [a, b]} \left| \hat{f}_{p+1}(x; h) - f(x) \right| = O\left(h^{p+1} + \frac{\sqrt{\log \log n}}{\sqrt{nh}}\right), \quad (9)$$

for $p \geq 1$. As $h = O\left(\left(\frac{n}{\log \log n}\right)^{-\frac{1}{2p+4}}\right)$, we also have

$$\sup_{x \in [a, b]} \left| \hat{f}_{p+1}(x; h) - f(x) \right| = O\left(n^{-\frac{p+1}{2p+4}} (\log \log n)^{\frac{p+1}{2p+4}}\right). \quad (10)$$

Remark 2: From Theorem 2.3, the convergence rate is constructed for the estimator (2). From the Theorem 2.4, the estimator (2) is better than the traditional KDE, because it does not have the boundary effects. For examples of why the smoother parameter h is chosen, see Silverman (1986) and Hardle (1991). For more details on the problem of bias reduction, see Gasser and

Muller (1979).

Remark 3: From Theorem 2.3, when the kernel function $K(\cdot)$ is symmetric function, we have

$$\sup_{x \in (-\infty, \infty)} \left| \hat{f}_{p+1}(x; h) - f(x) \right| = O\left(h^{2p} + \frac{\sqrt{\log \log n}}{\sqrt{n} h}\right), \quad (11)$$

for each $p \geq 1$.

3. Proofs:

In this section our main purpose is to prove Theorem 2.1-4. First, we introduce three auxiliary lemmas as follows:

Lemma 3.1. Let X_1, \dots, X_n be a stationary α -mixing sequence of real-valued random variables with distribution function $F(x)$ and mixing coefficient $\alpha(n)$ satisfying (A3), and let $F_n(x)$ be the empirical distribution function based on the segment X_1, \dots, X_n . Then

$$\sup_x |F_n(x) - F(x)| \rightarrow 0 \quad \text{almost surely, as } n \rightarrow \infty. \quad (12)$$

Proof: The proof follows from the Theorem 1 in Tucker (1967) and the Corollary 2.1 in Cai and Roussas (1992).

Lemma 3.2. Let X_1, \dots, X_n be a stationary α -mixing sequence of real-valued random variables with distribution function $F(x)$ and mixing coefficient $\alpha(n) = O(n^{-\delta})$, for some $\delta > 3$, and let $F_n(x)$ be the empirical distribution function based on the segment X_1, \dots, X_n . Let $F(x)$ satisfy the Lipschitz condition $|F(x) - F(y)| \leq C_1|x - y|$, for some constant C_1 . Then

$$P\left\{\limsup_x \left[\left(\frac{n}{2 \log \log n}\right)^{1/2} |F_n(x) - F(x)|\right] = \frac{1}{2}\right\} = 1, \quad (13)$$

almost surely, as $n \rightarrow \infty$.

Proof: The proof follows from the Theorem 3.2 in Cai and Roussas (1992).

For conveniences, let

$$D_{p+1} = \begin{bmatrix} \mu_0 & \mu_1 & \mu_2 & \cdots & \mu_p \\ \mu_1 & \mu_2 & \mu_3 & \cdots & \mu_{p+1} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \mu_p & \mu_{p+1} & \mu_{p+2} & \cdots & \mu_{2p} \end{bmatrix},$$

and

$$E_{p+1} = \begin{bmatrix} \mu_{p+1} & \mu_1 & \mu_2 & \cdots & \mu_p \\ \mu_{p+2} & \mu_2 & \mu_3 & \cdots & \mu_{p+1} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \mu_{2p+1} & \mu_{p+1} & \mu_{p+2} & \cdots & \mu_{2p} \end{bmatrix},$$

where $\mu_j = \int u^j K\{u\} du$, for $j=0,1,2,\dots, 2p+1$ and $p \geq 1$.

Lemma 3.3. Assume that the conditions (A1)–(A2) are satisfied. Then we have

$$E \hat{f}_{p+1}(x; h) - f(x) = O(h^{p+1}), \quad (13)$$

for $p \geq 1$ and $x \in (-\infty, \infty)$ or $x \in [a, b]$.

Proof: By the calculation of expectation, as $x \in (-\infty, \infty)$, we have

$$\begin{aligned} & E \hat{f}_{p+1}(x; h) \\ &= \int \frac{\det(U_{p+1}(\frac{x-y}{h}))}{\det(A_{p+1})} \frac{1}{h} K\left\{\frac{x-y}{h}\right\} f(y) dy \\ &= \int \frac{\det(U_{p+1}(\frac{x-y}{h}))}{\det(A_{p+1})} \frac{1}{h} K\left\{\frac{x-y}{h}\right\} \sum_{j=0}^p \left(\frac{x-y}{h}\right)^j \frac{(-1)^j h^j}{j!} f^{(j)}(x) dy \end{aligned}$$

$$\begin{aligned}
 & + \int \frac{\det(U_{p+1}(\frac{x-y}{h}))}{\det(A_{p+1})} \frac{1}{h} K\left\{\frac{x-y}{h}\right\} \left[f(y) - \sum_{j=0}^p (x-y)^j \frac{(-1)^j}{j!} f^{(j)}(x) \right] dy \\
 & = f(x) \frac{\det(A_{p+1})}{\det(A_{p+1})} + \int \frac{\det(U_{p+1}(\frac{x-y}{h}))}{\det(A_{p+1})} \frac{1}{h} K\left\{\frac{x-y}{h}\right\} \bullet \\
 & \quad [f(y) - \sum_{j=1}^p \frac{(x-y)^j (-1)^j}{j!} f^{(j)}(x)] dy \\
 & = f(x) + \frac{\det(E_{p+1})}{\det(D_{p+1})} \frac{h^{p+1}}{(p+1)!} (-1)^{p+1} f^{(p+1)}(x) + o(h^{p+1}) \\
 & = f(x) + O(h^{p+1}),
 \end{aligned} \tag{14}$$

from the property of (4) and the techniques of calculus underlying the conditions (A1)-(A2), for $p \geq 1$. Similarly, for $x \in [a, b]$ case, for a and b are finite, or by the paper of Fan and Gijbels (1992), we can also prove that

$$E \hat{f}_{p+1}(x; h) - f(x) = O(h^{p+1}). \quad (15)$$

Therefore, the proof of Lemma 3.3 is proved.

Proof of Theorem 2.1. We know that

$$\begin{aligned}
 & \sup_x \left| \hat{f}_{p+1}(x; h) - E \hat{f}_{p+1}(x; h) \right| \\
 & = \sup_x \left| \frac{1}{h} \int_{-\infty}^{\infty} W\left\{\frac{x-y}{h}\right\} dF_n(y) - \frac{1}{h} \int_{-\infty}^{\infty} W\left\{\frac{x-y}{h}\right\} dF(y) \right| \\
 & \leq \sup_x \frac{1}{h} \int_{-\infty}^{\infty} |F_n(y) - F(y)| \left| dW\left(\frac{x-y}{h}\right) \right| \\
 & \leq \frac{v}{h} \sup_x |F_n(x) - F(x)|, \quad (16)
 \end{aligned}$$

by the techniques of calculus and the conditions (A1) and (A2), where v is the variation of

$$W(\cdot) = \frac{\det(U_{p+1}(\cdot))}{\det(A_{p+1})} K(\cdot).$$

From (16) and the Lemma 3.1, we have

$$\sup_x |\hat{f}_{p+1}(x; h) - E\hat{f}_{p+1}(x; h)| \rightarrow 0, \quad (17)$$

almost surely, as $n \rightarrow \infty$.

Now, we consider the expectation of $\hat{f}_{p+1}(x; h)$. By the Lemma 3.3, we have $E\hat{f}_{p+1}(x; h) = O(h^{p+1})$. Hence one can prove that

$$\sup_x |E\hat{f}_{p+1}(x; h) - f(x)| \rightarrow 0, \quad (18)$$

almost surely, as $h \rightarrow 0$ and $n \rightarrow \infty$. From (17) and (18), the proof of Theorem 2.1 is completed.

Proof of Theorem 2.3. Let $D_n = \sup_x |F_n(x) - F(x)|$. By the Lemma 3.2, we have

$$P \left\{ \limsup_n \left(\frac{n}{2 \log \log n} \right)^{1/2} D_n = \frac{1}{2} \right\} = 1, \quad (19)$$

$F(x)$ is continuous.

By (19) as above, we have

$$\sup_x |\hat{f}_{p+1}(x; h) - E\hat{f}_{p+1}(x; h)| = O(n^{-1/2} (\log \log n)^{1/2} h^{-1}). \quad (20)$$

From (20) and Lemma 3.3, the proof of Theorem 2.3 is completed.

Proof of Theorem 2.2. The proof of Theorem 2.2 is similar to that of Theorem 2.1, the details are omitted.

Proof of Theorem 2.4. The proof of Theorem 2.4 is also similar to that of Theorem 2.3, the details are omitted.

References:

- Cai, Z. and Roussas, G.G. (1992). Uniform strong estimation under α -mixing, with rates. *Stat. & Prob. Lett.* 15, 47-55.
- Fan, J. and Gijbels, I. (1992). Variable bandwidth and local linear regression smoothers. *Ann. Statist.* 20, 2008-2036.
- Gasser, T. and Muller, H.G. (1979). Kernel estimation of regression functions. In *Smoothing Techniques for Curve Estimation* (eds. T. Gasser and M. Rosenblatt). Springer Verlag, Heidelberg, 23-68.
- Hardle, W. (1991). *Smoothing Techniques with Implementation in S*. Springer-Verlag, New York.
- Jones, M.C. and Foster, P.J. (1996). A simple nonnegative boundary correction method for kernel density estimation. *Statist. Sinica*, 6, 1005-1013.
- Jones, M.C. (1993). Simple boundary correction for kernel density estimation. *Statist. Computing* 3, 135-146.
- Kim, T.Y. and Cox, D.D. (1996). Uniform strong consistency of kernel density estimators under dependence. *Stat. & Prob. Lett.* 26, 179-185.
- Kim, T.Y. and Lee, S. (2004). Kernel density estimator for strong mixing processes. *Journal of Statistical Planning and Inference* (accepted).
- Muller, H.G. (1993). On the boundary kernel method for nonparametric curve estimation near endpoints. *Scandinavian Journal of Statistics*, 313-328.
- Parzen, E. (1962). On estimation of a probability density function and mode. *Ann. Math. Statist.* 33, 1065-1076.
- Prakasa Rao, B.L.S. (1983). *Nonparametric Functional Estimation*. Academic Press, New York.
- Rosenblatt, M. (1956). Remarks on some non-parametric estimates of a density function. *Ann. Math. Statist.* 27, 832-837.
- Simonoff, J.S. (1996). *Smoothing Methods in Statistics*. Springer-Verlag, New York.
- Schuster, E. F. (1969). Estimation of a probability density function and its derivatives. *Ann. Math. Statist.* 40, 1187-1195.

- Stone, C.J. (1982). Optimal global rates of convergence of nonparametric regression. *Ann. Statist.* 10, 1040-1053.
- Silverman, B.W. (1986). *Density Estimation for Statistics and Data analysis*. Chapman and Hall, London.
- Tucker, H.G. (1967). *A Graduate Course in Probability*. Academic Press, New York.
- Wand, M.P. and Jones, M.C. (1995). *Kernel Smoothing*. Chapman and Hall, London.

received October 18, 2006

revised December 2, 2006

accepted December 16, 2006

α 混合條件下之核密度估計量的均勻一致性

洪萬吉

嶺東科技大學財務金融系
台中市南屯區 40852 嶺東路 1 號

中文摘要

在混合條件下，研究核密度估計量的極限行為與收斂速率。當密度函數之定義域為緊緻時，傳統的密度函數估計量將遭遇到邊界效果的問題。為了改善其極限行為與收斂速率，本文依線性擬合法之觀點來建立一個新加權核密度估計量。此估計量之極限行為與收斂速率將被提供，且在 α 混合條件下，所提之估計量是不需要去改善其邊界效果，及它的收斂速率是達 $O(n^{-\frac{p+1}{2p+4}}(\log \log n)^{\frac{p+1}{2p+4}})$ ，對所有的 $p \geq 1$ 。

關鍵詞：核密度估計量，邊界效果，帶寬，均勻一致性，收斂速率，偏差減小， α 混合。

小波方法應用於神經元動作電位波形分類之正確度探討

李 元

輔仁大學數學系

蔡孟利

國立宜蘭大學生物機電工程學系

單維彰

國立中央大學數學系

嚴健彰

輔仁大學數學系

摘 要

Wavelet-based Spike Classifier (WSC)的方法是一種小波方法應用在神經元動作電位波形分類，本論文的研究目的是針對其精確度做探討。我們的研究方法是利用人爲的資料，建立精確度與訊號比的表格，結論是成正比關係。我們也將此結果應用於真實實驗中的資料，進一步去估計分出來的神經元動作電位波形類別的可信度。

關鍵詞：小波方法、神經元動作電位波形、精確度

一、簡 介

神經系統是動物體內主要的生理協調管制系統，控制動物體的感覺及運動等功能。神經元(neurons)是神經系統的基本構造及功能單位，而其傳遞訊息的主要機制即是透過

*Corresponding author. Tel:+886-2-29053547

E-mail:yen@math.fju.edu.tw

動作電位的產生及傳導，運載感覺訊息的傳入及運動指令的輸出[8]。雖然現代生物醫學影像方法突飛猛進，已經可記錄各種行為及心智活動時，實驗動物或人的腦中各部位的血流量、代謝速率、特定基因表現量及傳遞物質作用的能力，但電生理方法仍是目前唯一可直接觀察記錄單一神經元訊號的方法。自八十年代起，一些研究者便藉由微電子及個人電腦的發展，發明了將多根微電極植入動物腦中，然後在麻醉消失、傷口復原之後，在具有清醒行為的動物腦中，同時記錄許多個單一神經元的方法，此即為多通道記錄法(multichannel recording)。此法因可以同時在動物腦中多個部位同時記錄數十到數百個單一神經元的活性，是晚近神經電生理一個重要研究方法上的突破，又因其能在清醒行為的動物腦中不斷的記錄觀察，所以能用來解決許多行為學及認知科學上的重要問題[5]。因為通常微電極所記錄到的不只是多顆神經元的動作電位訊號，還有許多未知來源的雜訊干擾，所以多通道記錄法在數據分析上有一關鍵性的步驟，亦即對每一根記錄電極所感測到的電位訊號作適當的處理及分類，以確定每一顆神經元的動作電位在時間軸上的正確序列。在神經元訊號的分類上，通常是在以下的假設前提下進行工作，亦即每根電極內的每個被記錄到的神經元動作電位之波形均是唯一的，且不隨時間的進程而改變。因此動作電位的分類工作，常被化約成於一段固定時間的記錄中，分辨出擁有的波形種類。主成分分析(principal component analysis)為經常被使用的分析方式，小波分析(wavelet analysis)近年來也被引薦入此一問題之解決[4]。然而，如何判斷分類結果的正確度或估計其可能的錯誤率，目前並無較好之工具可資使用，因此本文即就如何鑑別波形分類結果的正確度加以討論。

本論文以下幾節為第二節的『小波理論簡介』，介紹小波及其相關的基本理論介紹。第三節的『小波轉換』，用以介紹 WSC 方法。第四節的『正確度分析』，介紹本論文如何探討正確度，與應用於真實的實驗數據。最後，第五節的『結論與討論』，做一個總結及討論日後可能改進的方向。

二、小波理論簡介

在 1980 年代，小波理論開始被人們注意，大約在 1987 年由 Mallat 和 Meyer 發展出來多層解析空間的觀念，之後就產生了一套製造各種不同小波函數的系統化方法。從這套系統化的方法產生了很多族類的小波函數，例如有 Daubechies 的正交有限涵蓋的小波函數[3]，Chui-Wang 的半正交(semi-orthogonal)小波函數[1]，Cohen-Daubechies-

Feauveau 的雙正交 (biorthogonal) 小波函數[2]。本節簡介小波理論，從多層解析空間開始，引進小波函數，對於小波函數的分類，我們採用連續與離散型及正交性做介紹。

設 $\{V_j : j \in \mathbb{Z}\}$ 是 $L^2(\mathbb{R})$ 的子空間且滿足以下條件

$$1. V_j \subset V_{j+1}$$

$$2. \bigcup_{j=-\infty}^{\infty} V_j \text{ 在 } L^2(\mathbb{R}) \text{ 中稠密, 亦即 } \overline{\bigcup_{j=-\infty}^{\infty} V_j} = L^2(\mathbb{R})$$

$$3. \bigcap_{j=-\infty}^{\infty} V_j = \{0\}$$

$$4. f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$$

$$5. f(x) \in V_0 \Leftrightarrow f(x-k) \in V_0, \text{ 其中 } k \text{ 是一整數}$$

$$6. \{\phi(x-k) : k \in \mathbb{Z}\} \text{ 是 } V_0 \text{ 的一組正則基底}$$

則稱 $\{V_j\}$ 組成一個多層解析空間。根據多層解析空間的定義，可以得到 $\{\phi_{j,k}(x) \equiv 2^{j/2} \phi(2^j x - k), k \in \mathbb{Z}\}$ 是 $\{V_j\}$ 的一組正則基底，記作 $V_j = \text{span}\{\phi_{j,k} : k \in \mathbb{Z}\}$ 。如果 $\phi(x) \in V_0$ ，從條件 1， $\phi(x)$ 也會落在 V_1 ，而 $\{\phi(2x-k) : k \in \mathbb{Z}\}$ 是 V_1 的基底，所以 $\phi(x)$ 可以表示成

$$\phi(x) = \sum_k h_k \phi(2x - k) \quad (1)$$

我們稱此等式(1)為自格等式(scaling equation)，其中 $\{h_k\}$ 是一組自格係數(scaling coefficients)，而 $\phi(x)$ 為一個自格函數(scaling function)。由於 $V_j \subset V_{j+1}$ ，定義 W_j 是 V_j 的垂直補空間，

$$\begin{aligned} V_{j+1} &= V_j \oplus W_j \\ &= V_{j-1} \oplus W_{j-1} \oplus W_j \\ &= \dots \\ &= V_0 \oplus W_0 \oplus W_1 \oplus \dots \oplus W_j \end{aligned}$$

存在一組函數 $\{\psi_{j,k}(x) \equiv 2^{j/2} \psi(2^j x - k), k \in \mathbb{Z}\}$ 是 $\{W_j\}$ 的一組正則基底。小波函數 $\Psi(x)$ 有兩個必要條件：

$$1. \int \psi(x) dx = 0 \text{ 且 } \int |\psi(x)| dx \neq 0$$

$$2. |\text{supp } \Psi(x)| < \infty \text{ 其中 } \text{supp } \Psi(x) \equiv \overline{\{x : \Psi(x) \neq 0\}}$$

因為第一個條件所以它的圖形一定上下起伏如波狀，且 $\Psi(x) \in L^1(R)$ ，所以 $|\Psi(x)|$ 底下所圍的面積不會很大，因此 Ψ 的圖形相較於整個實數軸，像是小波，這也是小波名稱的由來。以下是對小波做分類。

小波轉換， $\Psi^{a,b} \equiv |a|^{-1/2} \Psi(ax-b)$ ，依型態來分大致分成兩類，一類為連續型，另一類為離散型。 $\Psi^{a,b}$ 當為連續型；當 $a = \frac{1}{2^j}$ ， $b = \frac{k}{2^j}$ ， $j, k \in Z$ 為離散型。如果我們討論的空間是 $L^2(R)$ 空間，會有更好的性質。在連續型中，一個函數可以被它的小波變換式重建 [5]，也就是說

$$f(x) = C_\psi^{-1} \iint \langle f, \psi^{a,b} \rangle \psi^{a,b} \frac{da db}{a^2}$$

式中 \langle, \rangle 表示 L^2 的內積，而 $C_\psi = 2\pi \int \frac{|\hat{\psi}(w)|}{|w|} dw < \infty$ ，其中 $\hat{f}(w)$ 是傅立葉變換 (Four-

ier transform) 定義為

$$\hat{f}(w) = \int f(x) e^{-iwx} dx$$

在時間及頻率的分析時，如果僅對 $a > 0$ 有興趣。則它的重建公式寫成 [5]：

$$f(x) = C_\psi^{-1} \int_0^\infty \left[\int \langle f, \psi^{a,b} \rangle \psi^{a,b} db \right] \frac{da}{a^2}$$

離散型則為

$$f(x) = \sum_{j,k=-\infty}^{\infty} \langle f, \psi_{j,k} \rangle \tilde{\psi}_{j,k}(x) = \sum_{j,k=-\infty}^{\infty} \langle f, \tilde{\psi}_{j,k} \rangle \psi_{j,k}(x)$$

其中 $\{\psi_{j,k}\}$ 及 $\{\tilde{\psi}_{j,k}\}$ 都是 Riesz 基 [5] 且

$$\langle \psi_{j,k}, \tilde{\psi}_{\ell,m} \rangle = \delta_{j\ell} \delta_{km}, \quad j, k, \ell, m \in Z$$

稱 $\{\psi_{j,k}\}$ 及 $\{\tilde{\psi}_{j,k}\}$ 互為對偶基底 (dual base)。若 $\{\psi_{j,k}\}$ 是正規基底，則 $\{\tilde{\psi}_{j,k}\} = \{\psi_{j,k}\}$ 。

一般著名的小波，我們依正交特性將它們區分為：

1. 非正交(non-orthogonal)小波函數：任兩個小波函數間 $\{\Psi_{j,k}\}$ 和 $\{\tilde{\Psi}_{j,k}\}$ 彼此不正交。
2. 正交(orthogonal)小波函數： $\{\Psi_{j,k}\}=\{\tilde{\Psi}_{j,k}\}$ 且任兩個小波函數

$$\langle \Psi_{j,k}, \Psi_{\ell,m} \rangle = \delta_{j\ell} \delta_{km}, \quad j, k, \ell, m \in Z$$

彼此正交。例如：Daubechie's 的正交有限涵蓋的小波函數。

3. 半正交(semi-orthogonal)小波函數：任兩個小波函數

$$\langle \Psi_{j,k}, \Psi_{\ell,m} \rangle = \delta_{j\ell}, \quad j, k, \ell, m \in Z$$

間彼此正交。但不同層未必正交。例如：Chui-Wang 的半正交(semi-orthogonal)小波函數。

4. 雙正交(bi-orthogonal)小波函數：小波函數和其對偶(dual)小波函數間彼此正交，亦即

$$\langle \Psi_{j,k}, \tilde{\Psi}_{\ell,m} \rangle = \delta_{j\ell} \delta_{km}, \quad j, k, \ell, m \in Z$$

例如 Cohen-Daubechies-Feauveau (CDF)的雙正交(biorthogonal)小波函數。

三、小波轉換(Discrete Wavelet Transform, DWT)

我們以 Daubechie 的正交小波函數為例，來介紹小波函數的分解與合成。首先，定義 $H(\xi) = \frac{1}{2} \sum h_k e^{-ik\xi}$ 為 ϕ 的特徵函數 (characteristic function)，其中 $\{h_k\}$ 是自格係數。特徵函數 $H(\xi)$ 必須滿足以下等式， $|H(\xi)|^2 + |H(\xi + \pi)|^2 = 1$ [6]。接下來我們介紹如何求得 Daubechies 小波自格函數。令 $h_k = 0, k \notin \{0, \dots, 2p-1\}$ 且滿足以下條件

1. $\sum_k h_k = 2$
2. $\sum_m h_{2k+m} h_m = 2\delta_{0,k}$
3. $\sum_k (-1)^k k^m \cdot h_k = 0, \text{ for } 0 \leq m \leq p-1$

當 $p=1$ 時，係數為

$$h_0 = 1, h_1 = 1。$$

當 $p=2$ 時，係數為

$$h_0 = \frac{3+\sqrt{3}}{4}, h_1 = \frac{1+\sqrt{3}}{4}, h_2 = \frac{1-\sqrt{3}}{4}, h_3 = \frac{3-\sqrt{3}}{4}$$

當 $p=3$ 時，係數為

$$h_0 = \frac{1+\sqrt{10}+k}{16}, h_1 = \frac{5+\sqrt{10}+3k}{16}, h_2 = \frac{5-\sqrt{10}+k}{8},$$

$$h_3 = \frac{5-\sqrt{10}-k}{8}, h_4 = \frac{5+\sqrt{10}-3k}{16}, h_5 = \frac{1+\sqrt{10}-k}{16},$$

其中 $k = \sqrt{5+2\sqrt{10}}$ 。 $p=2$ 及 $p=3$ 的自格函數及小波函數圖形，請參考圖一。令 $\phi(x) = \sum h_k \phi(2x-k)$ 和 $\psi(x) = \sum g_k \phi(2x-k)$ ，其中 $g_k = (-1)^k h_{1-k}$ 。則我們有 p 階的 Daubechies 小波，它的涵蓋為 $\text{supp } \phi = [0, 2p-1]$ 。

將多層解析空間 V_j 中的函數從 $\{\phi_{j,k}(x)\}$ 的基底變換到 $V_{j-1} \oplus W_{j-1}$ 的基底 $\{\phi_{j-1,k}(x)\} \cup \{\psi_{j-1,k}(x)\}$ 的過程就是小波轉換中的分解(decomposition)，而從 $\{\phi_{j-1,k}(x)\} \cup \{\psi_{j-1,k}(x)\}$ 轉換到 $\{\phi_{j,k}(x)\}$ 的過程為合成(reconstruction)。針對離散化的函數數值或數位訊號 $v = (v_0, v_1, \dots, v_{N-1}) \in R^N$ ，為了方便起見，假設 $N=2^j$ ，則小波轉換就是在下式中的轉換：

$$\sum_{n=0}^{N-1} v_n \phi(2^j x - n) = \sum_{n=0}^{\frac{N}{2}-1} u_n \phi(2^{j-1} x - n) + \sum_{n=0}^{\frac{N}{2}-1} w_n \psi(2^{j-1} x - n)$$

將係數 $\{v_n\}$ 轉換成 $\{u_n, w_n\}$ 為分解，把係數 $\{u_n, w_n\}$ 轉換成 $\{v_n\}$ 為合成。

我們說明離散小波轉換 (DWT) 是如何作用的，假設輸入一個資料 v

$$v = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_{N-1} \end{pmatrix}$$

對它做 DWT，把 $v \in V_j$ 分解成

$$u = \begin{pmatrix} u_0 \\ u_1 \\ \vdots \\ u_{\frac{N}{2}-1} \end{pmatrix} \in V_{j-1} \quad \text{和} \quad w = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_{\frac{N}{2}-1} \end{pmatrix} \in W_{j-1}$$

而分解的步驟相當於兩個矩陣的乘法 $u=Hv$, $w=Gv$, 其中 H 是由 $\{h_k\}$ 所生成, 我們叫它為低頻濾波器, G 是由 $\{g_k\}$ 所生成, 我們叫它為高頻濾波器, 將 H 和 G 堆在一起就形成了小波轉換矩陣

$$W = \begin{bmatrix} H \\ G \end{bmatrix}$$

其中 W 的維度是 $N \times N$ 且

$$H = \frac{1}{2} \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & & & & \\ & & h_0 & h_1 & h_2 & h_3 & & \\ & & & h_0 & h_1 & h_2 & h_3 & \\ & & & & h_0 & h_1 & h_2 & h_3 \\ & & & & & \vdots & \vdots & \\ h_2 & h_3 & & & & h_0 & h_1 & \end{pmatrix}_{\frac{N}{2} \times \frac{N}{2}} \quad G = \frac{1}{2} \begin{pmatrix} g_0 & -g_1 & g_2 & -g_3 & & & & \\ & & g_0 & -g_1 & g_2 & -g_3 & & \\ & & & g_0 & -g_1 & g_2 & -g_3 & \\ & & & & g_0 & -g_1 & g_2 & -g_3 \\ & & & & & \vdots & \vdots & \\ g_2 & -g_3 & & & & g_0 & -g_1 & \end{pmatrix}_{\frac{N}{2} \times \frac{N}{2}}$$

因此 DWT 的分解過程相當於一個矩陣乘法

$$Wv = \begin{bmatrix} u \\ w \end{bmatrix}$$

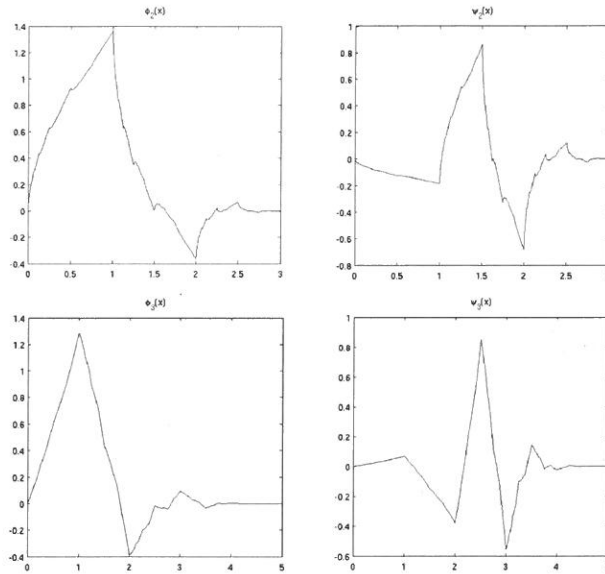
所以 DWT 的合成過程為

$$v = W^{-1} \begin{bmatrix} u \\ w \end{bmatrix}$$

其中 $W^{-1} = 2W^T$ 在正交的小波中。因為 W 和 W^{-1} 的向量計算量和小波的係數長度有關, 其計算量為 $O(N)$, 因此稱為快速小波轉換。

前面分解合成的過程是理論說明, 在實作上我們是把 H 和 G 的列向量交錯排列, 再以每個列向量對 v 做向量乘法, 這樣只需要一個暫存空間就可以把結果直接覆寫在的儲存空間, 然後將 v 的奇數列排在前半部, 偶數列排在後半部, 回到 $(u, w)^T$ 的型態, 就完成分解一次的 DWT。這個步驟可以重複地執行, 直到滿意的解析度為止。而 DWT

的合成過程，在實作上一樣先把 H 和 G 的列向量交錯排列，再將它們做轉置，當然這裡的 $(u, w)^T$ 也要交錯排列，最後將先前轉置的矩陣與交錯排列的 $(u, w)^T$ 做向量乘法。



圖一：此圖展示 Daubechies 小波的自格函數與小波，上面是階數為 $p=2$ ，下面是階數為 $p=3$ 左邊是自格函數，右邊是小波函數。

四、正確度分析

讀者如果對 WSC 方法細節有興趣，可參考[4]。在此我們只是簡單的介紹且針對我們的實驗過程。一個神經訊號在實驗中被記錄的，稱之為一個訊號事件。在我們的實驗中，用 $N=32$ 個數據紀錄一個訊號事件，記為 $s' = (s'_1, s'_2, \dots, s'_{32})$ 。經由小波轉換，做了四層，稱為小波係數且記為 $w^i = (u_{1,0}^i, u_{1,1}^i, w_{1,0}^i, w_{1,1}^i, \dots, w_{4,0}^i, \dots, w_{4,15}^i)$ 。我們分別將以上小波係數投影到平面上，因此可有 $N(N-1)/2 = 496$ 個平面，其投影是選定任兩個係數 $w_{j',k'}^i$ 及 $w_{j'',k''}^i$ ，再從其中找到最清楚顯示出點群集的圖。在我們發展的程式中，WSC 方法的最後步驟建立在相關的一個圖表分類界面軟體，允許使用者依視覺上瀏覽作各種預

測和手工限定群[4]。這相較原論文用統計方法，找到投影二座標，是較花時間，但卻是較精準的。

因為雜訊比在神經訊號未有標準定義，我們定義一個指標 SDN，這個量有點類似一般的雜訊比(signal to noise ratio, SNR)用來標示和正確度的關係。其定義方式為，每一次實驗時，在使用者手工限定群後，將所選定的每個訊號群組 s^i ，求每一點的平均值後形成的波形，再從波形找出其振幅 d_i ，最後取數據中最小的 d_i 當成訊號的強度，記做 \bar{S} 。在實驗時，雖然 s^i 此時有被雜訊干擾，但取平均後看作不受干擾的。每次用來作為干擾用的雜訊 n_i ，在所有動作完成 (m 次) 後計算其平均 μ_n ，計算方式為一般的標準差 (standard deviation)

$$\sqrt{\frac{1}{m} \sum_{i=1}^m (n_i - \mu_n)^2}$$

所得之值記為 \bar{N} 。則定義 SDN 的計算為

$$SDN = \frac{\bar{S}}{\bar{N}}$$

正確度的計算方式是在資料分析前，將每一個已知的資料依序作標記(如果決定用 4 個神經元，那在複製擴大資料後，以 1 千筆為單位，分別依序由前至後加上 1、2、3、4 作為標記)，而在分析後所確定群組，以所占比率最大的當正確標準。以此類推分完所有群組後，將正確點數的總和除上被分析的訊號總點數，所得到的百分比值，作為此次實驗的正確度。在以下表格 1 中每個 SDN 所對應的正確度是由 60 次實驗後並記錄正確度後，分別將最高和最低的 6 組數據排除後，計算其平均值得來。而為求實驗數據的客觀性，找來 6 位不同的實驗者分別以視覺上瀏覽作各種預測。

為了更清楚的了解我們的測試方法，我們選用主成份分析 (PCA) 來做比較，先由實際實驗取得神經訊號(signal)且雜訊(noise)數據亦來自於實際實驗中，共取得的 500 筆數據。其過程為

- Step 1. 隨機取出 4 至 7 個神經訊號作為此測試的數據。採用此方法的原因為希望能模擬實際情況中，探針附近探測出有多少神經訊號產生的未知性，且在一定程度上使樣點數不至於過少。
- Step 2. 將第 1 步中取出的樣點各自複製產生 1000 筆資料，依序複製後產生 4000 至 7000 筆資料。
- Step 3. 請使用者設定此組實驗等，雜訊大小要為實際雜訊數據大小的幾倍。

- Step 4. 由 500 筆雜訊數據，每次皆隨機取出其中 1 筆，並在乘上第 3 步所設定的倍數後將其與第 3 步中所產生的資料依序每次取一筆作相加的動作，重覆直到所有資料都作過此動作，用以模擬實際實驗時雜訊的干擾。
- Step 5. 應用 PCA 和 WSC 方法，在其建立的相關的一個圖表分類界面上，由 6 位使用者依視覺上瀏覽作各種預測和手工限定群後，經由電腦程式分析判定每次測試的正確度。
- Step 6. 在經過每組 60 次的實驗後計算出測試平均的正確度，並記錄於下列表格與圖示。

在實驗結果發現，當 SDN 值低於約 7.5 時，將使得 PCA 分析的正確度低於 60%，而小波分析卻仍有約 65% 的正確度。在 SDN 值大於 7.56 時，PCA 分析的正確度雖稍高於小波分析，但卻幾乎相同(正確度差距小於 0.5%)，而在此值之下，小波分析分析的正確度都高於 PCA 分析，甚至在 SDN 值約為 7.543 時，兩者的正確度相差近 10%。由此可看出，當我們在做實驗而結果能夠具有六成以上的正確度時，使用小波分析法能可能為較優的選擇。至於考量到實驗結果的正確度低於 60% 時，實驗數值的信賴度過低而不被接受的可能性，表格中僅列出 SDN 值約 7.51 以上，兩者皆具六成以上正確度之數據。我們應用以上的結果在真實實驗資料 Tsai et al. (2004)[7]上，其結果請參看圖 2。而我們利用以上所提出的方法作分析，並在計算出 SDN 值後，對照表格一求出其正確度為 99%，因此相當可信賴。

五、結論與討論

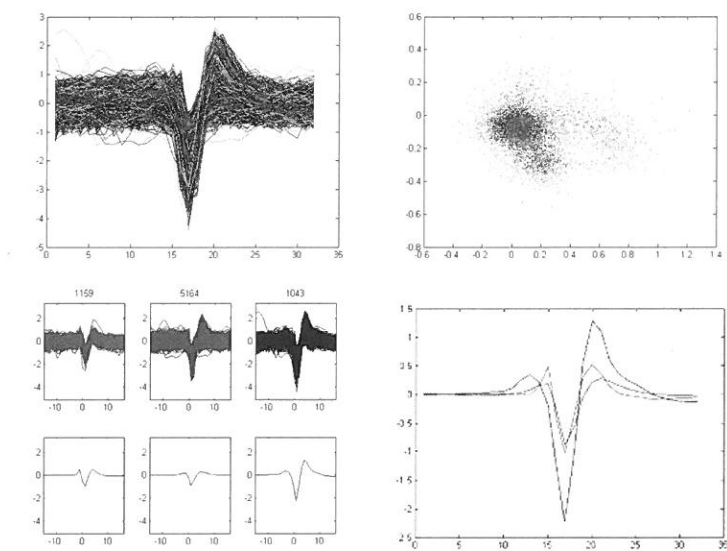
本論文建構了一個由神經訊號和雜訊的比值與正確率的對照表，提供了一個用以判斷實驗準確度的方法。因為實驗結果必須考慮到可信度的情形下，而從表格中看來小波分析在某種程度上對神經訊號的分析能力優於 PCA 分析。我們必需強調，對於同一張以神經訊號所產生的點圖，每一位實驗者在視覺上瀏覽所作的各種預測皆不盡相同，即便是同一位實驗者，每次圈選所產生的結果亦不盡相同。

雖然本論文提供一個判斷實驗準確度的表，但推得的方式依然有很大的改進空間，還是有一些可以再探討的地方，例如，(i) 我們可以發現依本論文方法所建構出來的神經訊號點圖，每一區塊形狀大小皆非相似，而如何改變每個訊號干擾後，能形成不同形狀的點圖，用以更接近模擬真實的情況。(ii) 另外在實驗中發現，愈分散愈不明顯的點

主成份分析法(PCA)		小波分析法(WSC)	
<i>SDN</i>	正確度 %	<i>SDN</i>	正確度 %
7.5807	98.57	7.5812	98.03
7.5794	97.26	7.5780	95.97
7.5756	95.44	7.5748	94.13
7.5697	91.99	7.5696	91.76
7.5641	90.22	7.5651	89.79
7.5607	83.33	7.5614	84.51
7.5559	77.37	7.5577	79.66
7.5514	71.26	7.5533	77.59
7.5473	68.01	7.5492	75.66
7.5429	65.26	7.5446	74.49
7.5376	64.49	7.5393	72.52
7.5285	63.69	7.5349	70.46
7.5206	62.53	7.5277	68.73
7.5121	60.78	7.5235	66.29

表一：此表是實驗結果，當 **SDN** 值低於約 **7.5**，將使得 **PCA** 分析的正確度低於 **60%**，而小波分析卻仍有 **65%** 的正確度。

集將造成愈高的振幅，代表是較強烈的訊號，但為何點集卻如此分散和不明顯，這將造成人爲判斷時反而容易錯失最強烈的訊號，造成此情況的原因是值得探討的問題。(iii) 我們用 SDN 值來作為正確度的判斷依據，但在實驗過程中發現，如果訊號分析出來的點圖，不以常理判斷，而隨便選圈時，則所建構出的表格將不再適用。比如，以肉眼可以明顯分辨出至少有 2 群以上訊號點集中處，卻將所有的訊號當作 1 群來作分類時，所得到的 SDN 值將比按照常理判別時來的高，這將造成查表後反而得到較高的正確度的情況。此種情況的發生，是由於所圈選的點愈分散，振幅愈大(由(ii))，造成 SDN 的值愈大。最後因爲本論文所提供之正確度對照表有(iii)的情況，所以如何尋求更好的解決方法，使結果在任何情況下都能適用，亦是一個值得深思的問題。諸如上述的情況，仍可再做更進一步的研究探討。另外對於審稿人提出的意見，例如雜訊中只有較大、較高的波峰與波谷造成訊號辨別的困擾，是否可用雜訊中較大(如最大的 25%)的波峰的值作標準差的計算，以作分母，是非常有用且實質的建議。



圖二：左上是含雜訊干擾的神經訊號圖，右上是 WSC 所產生的 496 張圖中，依視覺瀏覽所選出具有最多堆數且較明顯之圖，左下是選擇的各個群組的訊號圖，並求取平均後在每個群組的下方以單一訊號代表，右下是將各個群組的代表訊號圖疊合顯示。

六、參考資料

- [1] C.K Chui AND J.Z. Wang, A cardinal spline approach to wavelets, Proc. Amer. Math. Soc., 113(1991), pp. 785-793.
- [2] A. Cohen, I. Daubechies and J. Feauveau, Biorthogonal bases of compactly supported wavelets, Comm. Pure Appl. Math., 45(1992), pp. 485-560.
- [3] I. Daubechies, Orthonormal bases of compactly supported wavelets, Comm. Pure Appl. Math., 41(1988), pp. 909-996.
- [4] J.C. Letelier and P.P. Weber, Spike sorting based on discrete wavelet transform coefficients. J. Neurosci. Methods 101(2000) pp. 93-106.
- [5] M.A. Nicolelis, Brain-machine interfaces to restore motor function and probe neural circuits. Nat Rev Neurosci. 4(5)(2003): pp. 417-22.
- [6] W.C. Shann, First concepts of wavelets (凌波初步), 台北市：全華, 1998.
- [7] M.L. Tsai, C.C. Kuo, W.Z. Sun and C.T. Yen, Differential morphine effect on short- and long-latency laser evoked cortical responses of the rat, Pain 110 (2004) pp. 665-674.
- [8] 黃基礎、史金燾、施河，人體生理學 Human Physiology，藝軒圖書出版社，台北市。

received October 30, 2006

revised December 12, 2006

accepted December 23, 2006

Accuracy of Wavelet-based Spike Classifier

Yuan Li

Department of Mathematics, Fu Jen Catholic University, Taipei, R.O.C

Meng-Li Tsai

Department of Biomachtronic Engineering, National Ilan University, Ilan, Taiwan, R.O.C.

Wei-Chang Shann

Department of Mathematics, National Central University, Chung-Li, R.O.C.

Chien-Chang Yen

Department of Mathematics, Fu Jen Catholic University, Taipei, R.O.C

Abstract

Wavelet-based Spike Classifier(WSC) is one of wavelet methods which has been applied on classifications of spikes. This paper presents a study of accuracy for the forementioned method by constructing a accuracy verse signal to noise table. The result shows they have a strong correlation and has been used to estimate the accuracy for experiments.