Effective Network Grid Synthesis and Optimization for High Performance Very Large Scale Integration System Design

February 2008

Waseda University
Graduate School of Information, Production and Systems
Major in Information, Production and Systems Engineering
High-Level Verification Technologies

Yun Yang

DEDICATION

I would like to dedicate this thesis to my loving parents, Mr. Banghua YANG and Mrs. Yuzhen WANG, and my nice sisters, Mrs. Ling YANG and Mrs. Yingzi YANG, for their helpful advice and durative support in my last thirty year study life, and for their persistent encouragement and kindly understanding in my future academic career expectation. Also, I would like to thank all those who help me, inspire me, and support me in life, study, and research of my past two-decade struggle experience.

Abstract

Yun YANG

Network grid is one of the core elements in the VLSI processing systems. With the rapidly developed requirements of operation frequency, integration density and system scale, many research targets are presented for high performance VLSI system design. Effective network synthesis methods and proper global optimization algorithms are recent design challenges for the VLSI SoC system research.

One effective network grid design in this thesis is power/ground (P/G) distribution network synthesis in the VLSI circuit layout. Instead of the conventional SLP algorithm, the novel "Grid Synthesis using Genetic Algorithm (GGA)" is proposed to avoid the linear transformation loss and the local optimization demerit, inspired by the biological evolution analogy. And the "Hybrid Grid Synthesis based on Genetic Algorithm (HGGA)" is also proposed to deal with the dynamic signal situation and increase the genetic operation speed. Experimental results show that new P/G network layout area is smaller than previous algorithms, as the fittest survival theory in the nature. Also, the GGA/HGGA method can be applied for all P/G network synthesis problems because it can get the results directly, no matter whether these problems are linear or nonlinear. Thus the proposed GGA and HGGA methods are efficient for VLSI power/ground network synthesis to realize the global network optimization.

Another VLSI network grid design introduced in this thesis is the high performance systolic array construction. Traditional systolic arrays for matrix multiplication are limited by the deficiencies of throughput latency and utilization rate. The proposed systolic array methods are focused on extra network connection wires among the processing elements (PEs). By skillful connection wire adjustment and careful

computation sequence allocation, the processing efficiency is improved rapidly and the operation speed is increased to fully global pipeline operation. Two concurrent configuration algorithms for systolic network grid, including “Super Pipeline” and “Jumping Systolic Array (JSA)”, are proposed for high performance VLSI system design. The “Super Pipeline” design method is basic algorithm and is suitable for small scale network adjustment, which is emphasized in throughput latency and utilization rate by fast operation speed, compact array area and fully pipeline thread. For large scale array situation, another “Jumping Systolic Array (JSA)” method is proposed by the focuses on expansion ability, computation stabilization and global concurrence to deduce the general scale concurrent relationship. Various planar systolic array topologies, such as square topology and hexagonal topology, have been studied to construct the appropriate systolic array and realize high performance matrix multiplication. Experimental results prove that the proposed network grid design methods can realize fully concurrent processing, fast operation speed and compact array area with domination over other systolic architectures in band width, matrix complexity, or expansion capability, etc.

In conclusion, the novel GGA and HGGA algorithms are proposed to the power/ground (P/G) network synthesis and optimization, and the skillful connection adjustment methods are realized to the systolic array efficiency improvement. Based on these new design methods, the network grid synthesis with high performance is easy to be implemented for recent VLSI system design.

Keywords: *SLP algorithm, GGA method, HGGA method, gene disturbance, hybrid-SLP, P/G network synthesis, global optimization, matrix multiplication, super pipeline, jumping systolic array, fully concurrent operation, network grid design.*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

VLSI: Very Large Scale Integration

EDA: Electronic Design Automation

SOC: System on Chip

P/G: Power/Ground Network Grid

GGA: Grid Synthesis using Genetic Algorithm

HGGA: Hybrid Grid Synthesis based on Genetic Algorithm

PAD: Power/Ground Source Pad in SoC Chip Layout

DECAP: Decoupling Capacitance

KCL: Kirchoff's Current Law

KVL: Kirchoff's Voltage Law

MNA: Modified Nodal Analysis

ODE: Ordinary Differential Equation

AWE: Asymptotic Waveform Evaluation Algorithm

PRIMA: Passive Reduced-order Interconnect Macromodeling Algorithm

HIPRIME:	Hierarchical Passivity preserved Interconnect Macromodeling Engine
PADÉ:	Padé approximants for P/G Network Simulation
KRYLOV:	Krylov Subspace Method
IEKS:	Improved Extended Krylov Subspace
RLKC:	Resistance, Capacitance, Inductance and Mutual Inductance
ALM:	Augmented-Lagrangian Method
CGM:	Conjugate-Gradient Method
FDM:	Feasible-Direction Method
LP:	Linear Programming
SLP:	Sequence of Linear Programming
SQP:	Sequential Quadratic Programming
CG:	Conjugate Gradient Method
ICG:	Improved Conjugate Gradient Method
FE:	Forward Euler Method
BE:	Backward Euler Method
DGGA:	Dynamic Signal Manipulation in Genetic Algorithm

HYBRID-SLP:	Hybrid-SLP Method with Genetic Algorithm
TME:	Trapezoidal Modified Euler Method
TLM-ADI:	Transmission-Line-Modeling Alternating-Direction-Implicit
SA:	Simulated Annealing
LIM:	Latency Insertion Method
DFM:	Design For Manufacture
DSP:	Digital Signal Processing
PE:	Processing Element
BMMSA:	Band Matrix Multiplication Systolic Array
JSA:	Jumping Systolic Array
DFT:	Discrete Fourier Transform
FFT:	Fast Fourier Transform
FPGA:	Field Programmable Gate Array
DCT:	Discrete Cosine Transform
DST:	Discrete Sine Transform
IDCT:	Inverse Discrete Cosine Transform

IDST:	Inverse Discrete Sine Transform
CDMA:	Code Division Multiple Access
RSA:	Rivest-Shamir-Adleman Algorithm

Chapter 1

INTRODUCTION

1.1 Network Grid General Architecture

With the rapid progress of consumer electronics technology, the VLSI chip design and the EDA tools research become more and more important for high performance electronics product manufacture. Recently, large scale, high speed and low power VLSI system development become the research hot-spot. Many scientists and engineers focus on the next generation system VLSI chip design, or the complexity EDA tools development, such as software/hardware codesign, systematic synthesis and high-level verification. Thus deep characteristic exploration and further performance study are necessary to be performed for new VLSI system research advancement.

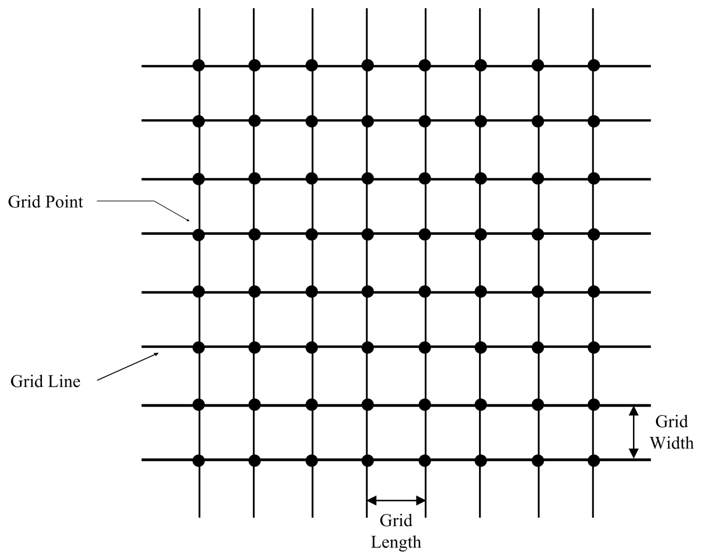


Figure 1.1: The general network grid architecture illustration.

The "Network Grid" in high performance VLSI system design means large scale network connection in the chip manufacture or EDA tools development. As shown in basic network grid architecture in Figure 1.1, the crossover wires compose the mesh network grid style. The mesh wires are named as "Grid Line", and the crossing points are "Grid Point". Also, the distances between neighboring grid line represent the "Grid Length" and the "Grid Width" in corresponding horizontal and vertical directions, respectively. In practical VLSI system design, the grid wires indicate the power lines, ground lines, interconnection wires, or other circuit connection network.

1.2 Power/Ground Network Grid

The global interconnection network in the VLSI system, such as the power/ground (P/G) network as one of the typical network grid architectures, delivers the signal to each chip inner element and becomes more difficult to be designed based on the super large network scale. Also, the durative dimension reduction with rapid chip speed increase will influence the practical circuit performance by the capacitance, inductance, crosstalk, or other parasitic effects. Then how to analyze and synthesize the global network grid in VLSI chip design becomes the active research point, and it was considered as one of the "TOP Ten Physical Design Frontier Research Problems in 21 Century" proposed by the ACM ISPD conference in 1999 [9].

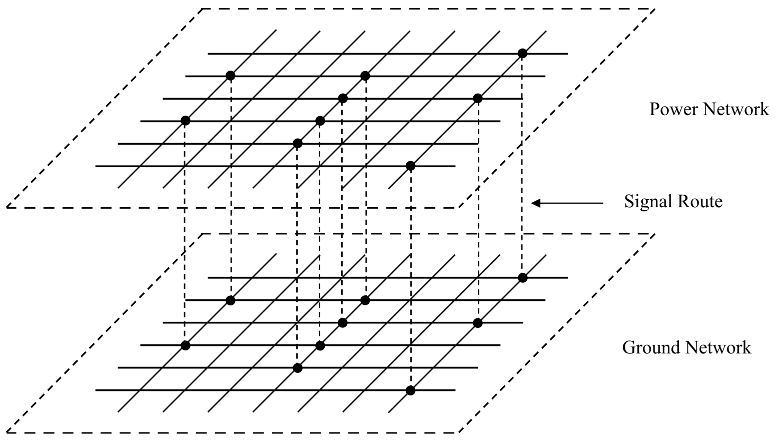


Figure 1.2: The typical power/ground (P/G) network architecture.

Figure 1.2 describes the typical power/ground network architecture. In the recent VLSI design, there are many layers for the power and ground network layout. The power mesh network transmits the source signal to circuit elements and enables the VLSI system operation. The ground mesh network provides system ground electrical level and keeps the circuit operation stabilization. The current signals flow from the power mesh network to the ground mesh network through the inner multi-layer circuit elements, by the performance driving for the VLSI circuit elements. The wire size, mesh density, connection arrangement, or other network characteristics, can also determine P/G network efficiency and related circuit performance.

1.3 Systolic Array Network Grid

Another example of the "Network Grid" design is the systolic array network in the high speed VLSI implementation, fast DSP element processing and high performance computer design [10]. "Systolic array" means large data pipeline processing architecture as shown in Figure 1.3, where the processing elements (PEs) and their connection wires compose the systolic array network. In practical VLSI system, the processing elements represent operation devices, processing unit, or other computation elements. Also, the connection wires mean data routes, result routes, or other signal routes.

In VLSI system the connection wires among the array processing elements can determine the operation speed and the result accuracy. There are many types of connection architectures, such as two-dimension array in Figure 1.3(a) and three-dimension architecture in Figure 1.3(b). In general, the systolic array network and the system computation are processed with highly concurrent pipeline thread. Different array architectures are corresponded to specific operation characteristics. Also, the adjustment and allocation of the systolic array architecture can increase the system processing efficiency and reduce the chip area for more compact VLSI system design. Thus the suitable architecture design of systolic array network is one of the important challenges for high performance VLSI chip implementation.

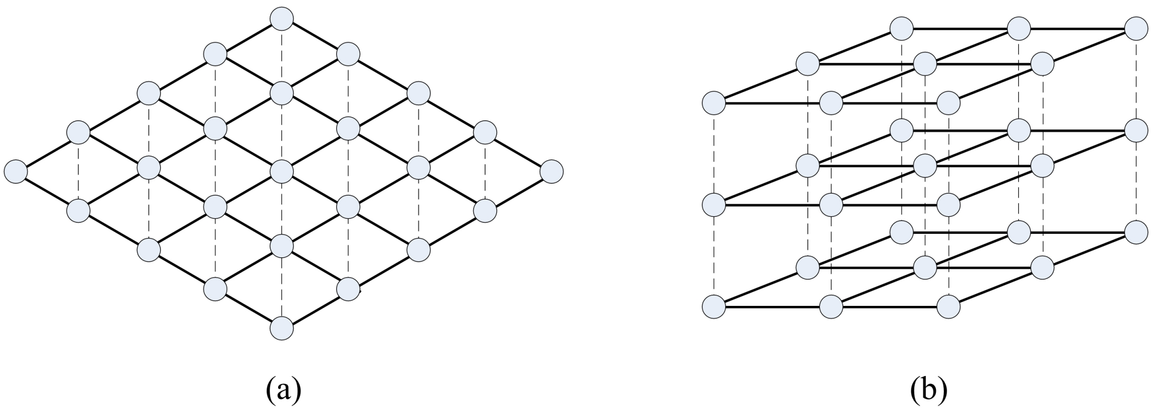


Figure 1.3: The systolic array network architecture examples. (a) two-dimension array, (b) three-dimension array.

1.4 Our Contributions

In this thesis we analyze the P/G network grid characteristics. To construct the high performance power or ground grid networks, we study network connection method and try to find the detailed relationship of each wire segment with network current and node voltage in the mesh network. Based on the research of IR-drop voltage loss in P/G distribution network, a novel network construction method of "Grid Synthesis using Genetic Algorithm (GGA)" is proposed to realize the network grid synthesis optimization. Under the basic genetic algorithm characteristic of global optimization and colony evolution, the GGA method can jump out of local minimal snare and realize global optimization operation. Also, the suitable and minimal P/G network grid can be constructed to reduce the P/G network layout area in the VLSI circuit design. In addition, the "Hybrid Grid Synthesis based on Genetic Algorithm (HGGA)" is also proposed to deal with the dynamic signal influence, caused by the DECAP and inductance in the P/G network grid. The gene disturbance, the Trapezoidal Modified Euler (TME) and the hybrid SLP-Genetic method have been used to construct the dynamic P/G network grid and realize the nonlinear optimization object. The test benches of some P/G network grid examples are given to prove the GGA/HGGA method efficiency. Also, some additional discussions of grid performance analysis and network synthesis are contributed to next era network grid research.

Another improvement in this thesis is the jumping systolic array method introduction for high performance systolic array design. Under careful study for existed systolic array architectures, we found that the processing element connection wires can influence final operation speed and output data accuracy. Also, the input and output data matrices can be condensed to more compact size by skillful connection design and data allocation. Based on these information, we proposed the "Super Pipeline" and "Jumping Systolic Array (JSA)" algorithms to improve the system operation efficiency and decrease the processing chip area. Basic idea of these systolic methods is the connection wire adjustment among the processing elements. The different wire link methods enable data jumping in the operation system and improve the processing efficiency. The trade-off for these methods is the complexity connection networks and the enhanced element driver requirements. This connection wire adjustment method can also be used in other network grid architectures by general analogy and ingenious design. Also, the related systolic array designs can be enlightened by the ideas of data jumping and connection adjustment.

1.5 Thesis Organization

The rest parts of this thesis are organized as follows. The detailed explanation and recent research achievements of the network grid are given in Chapter 2. Also, Chapter 3 describes novel "Grid Synthesis using Genetic Algorithm (GGA)" and "Hybrid Grid Synthesis based on Genetic Algorithm (HGGA)" for high performance power/ground (P/G) network synthesis and optimization. Chapter 4 introduces "Super Pipeline" and "Jumping Systolic Array (JSA)" methods for fully concurrent systolic system design. The final Chapter 5 concludes the whole thesis and expects the network grid synthesis in the future.

Chapter 2

BACKGROUND AND RECENT RESEARCH

Recently, the VLSI circuit dimension decreases rapidly to nanometer level and the chip speed increases greatly to GHz level. The global network grid performance becomes one of the main factors to influence the final design results. The chip design method should be converted from the device-center mode to the network-center mode. Thus how to manipulate the high performance chip system by the network grid consideration is the research hotspot now. The specific research area is focused on global network analysis, grid system optimization and connection wire improvement. In the following sections, some classical algorithms and recent achievements are given to introduce the network grid research background.

2.1 Global Network Grid Synthesis Design

2.1.1 Power/Ground Network Synthesis Target

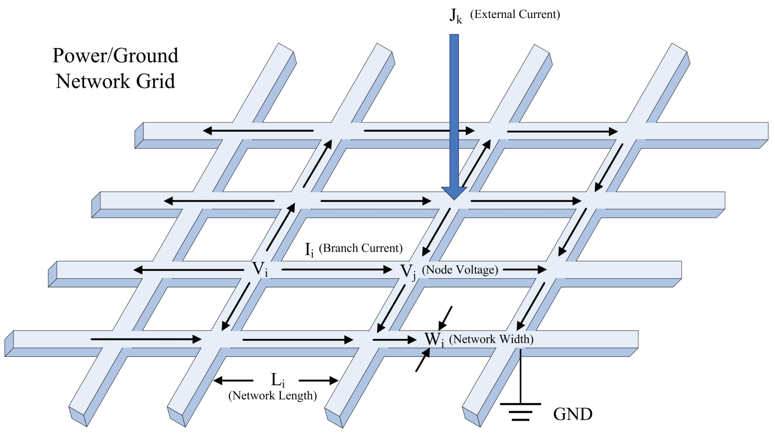


Figure 2.1: The typical power/ground (P/G) network grid architecture [1–3].

For the power/ground (P/G) network construction in network grid synthesis, the dimension reduction and operation speed increase will influence the whole network performance. Large IR-drop, Ldi/dt noise, Cdv/dt current, electromigration and crosstalk effects also disturb the circuit function, or even cause the chip out of work. So the suitable P/G network design method is important for recent VLSI system.

In addition, the P/G network area occupies large parts in recently high speed chip layout. The large network area can decrease the global parasitic effects and reduce the system noise influences. However, the large network grid will also increase the chip routing area and occupy many portions of the chip layouts. Thus it is important to find the suitable relationship between the P/G network performance and the chip layout area for better network grid synthesis design.

Figure 2.1 describes the typical power/ground (P/G) network grid architecture [1–3]. The current flows into the network from external PAD. Each network crossing point is named as "Node" with the corresponding "Node Voltage (V_i, V_j, ...)". The network segment between two neighboring nodes is called as "Branch" and its flowed current is "Branch Current" as I_i in Figure 2.1. The branch width and length are showed with W_i and L_i. Also, the "External Current J_k" provides the network power to drive the whole circuit and enable the network operation.

Consequently, the P/G network grid synthesis is how to realize the suitable power and ground network to satisfy the VLSI chip function requirements. Also, the P/G layout area should be designed skillfully to use the minimal chip area for the network global optimization target.

2.1.2 Systolic Array Network Synthesis Target

For another global network grid synthesis, the systolic array network design is mainly focused on the throughput latency, average utilization rate and system operation speed in the processing element array. The data matrix organization, the processing array architecture, and the connection network construction method, can influence the whole system operation efficiency. Compact throughput latency and fully utilization rate are used to increase the systolic array operation speed. However, system efficiency is difficult to be improved under common design method. How to build better systolic array network becomes one of the challenge research targets for recent high performance fast speed VLSI system design.

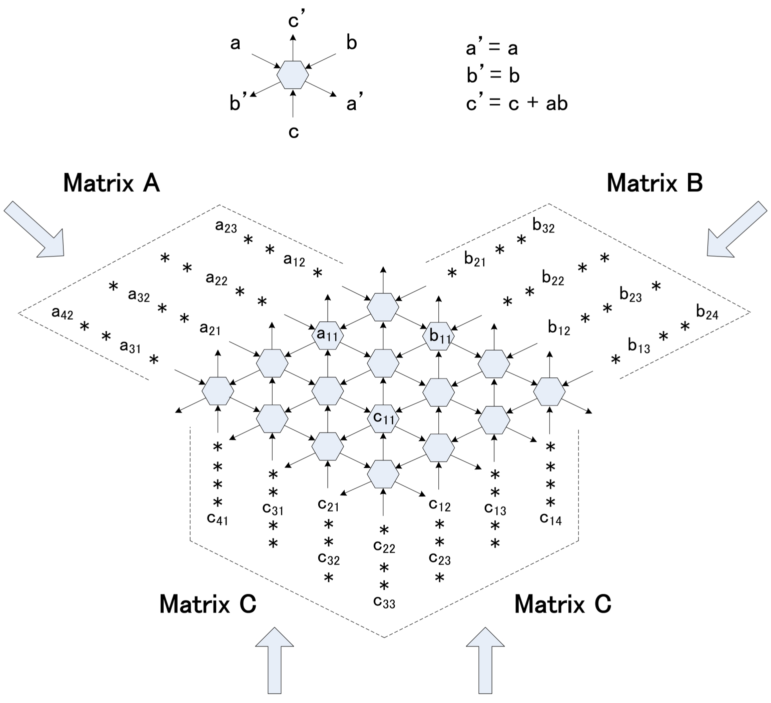


Figure 2.2: The typical two-dimension systolic array network architecture [4–6].

As the illustration in Figure 2.2, the systolic array architecture is given to show the basic network connection style and describe the processing element operation efficiency [4–6]. Each hexagon element in the array means a processing unit, where the multiplication function is realized by the equation "$c' = c + ab$" (a', b', c' are output data, and a, b, c mean the input data). Three data matrices input the systolic array and the elements a_{ij} (or b_{ij}, c_{ij}) mean the matrix data. The flag '$*$' is the latency cycle in the multiplication operation. So the typical two-dimension systolic array has only 33% processing element active in each cycle. The throughput latency is just about $3n$ without high processing efficiency, where n is the systolic array scale.

As a result, common systolic array requires the system efficiency improvement and processing speed increase. Many efforts and new algorithms are introduced to improve the array system performance. Also, this thesis provides new super pipeline and data jumping methods to quicken the system processing speed and realize the high performance systolic array network grid design.

2.2 Power/Ground Network Classical Algorithms

With frequency increase and dimension reduction for VLSI chip, the power/ground network IR-drop, Ldi/dt noise, Cdv/dt current, electromigration, or other parasitic effects, will cause the voltage transmission loss and influence the VLSI circuit function. Thus many classical algorithms are proposed to analyze, synthetize or optimize the P/G network grid in the recent years.

2.2.1 Power/Ground Network Analysis Algorithms

To realize the power/ground network grid optimization, the careful network analysis and grid computation for node voltages and branch currents is necessary to be developed. Recent researches mainly aim at the complex architecture and large scale network analysis with the order-reduction algorithms and linear transmission methods to reduce the network complexity and simplify the analysis procedure.

The common P/G network analysis methods, such as the decoupling capacitance and the topology allocation, are emphasized on the direct network simulation and performance study, which are suitable for the small size network situation. For large scale P/G network, the model order reduction algorithms are proposed to speed up the circuit analysis. Introduced from the famous “Asymptotic Waveform Evaluation (AWE)” method, a linear RLC circuit response approximation mode is proposed to deal with the general RLC interconnect situation [11]. Also, the “Passive Reduced-order Interconnect Macromodeling Algorithm (PRIMA)”, is proposed to improve the order reduction model accuracy under the consideration of inductance effects and macromodel passivity [12]. In addition, an "Improved Extended Krylov Subspace (IEKS)" method is used to construct the Norton equivalence circuit under the basic PRIMA reduced-order algorithm to resolve the network analysis results. This HiPRIME algorithm is a general hierarchical analysis methodology and can be used to analyze the RLKC power network grid efficiently [13].

Another feasible method to simulate the large size power delivery network is the hierarchical analysis algorithm [14]. The novel macromodeling method is used to partition the power distribution grid. Also, the numerical sparseness technique of the macromodeling port admittance matrices can ensure the network conservative approximation by using a "0-1" integer linear programming formulation. The power grids up to the increasing size of million-node level network can be analyzed by the efficient hierarchical algorithm under the suitable run-time and memory consumption. In addition, other order reduction method for network grid size is multigrid-like power grid analysis technique [15]. This method removes some grid nodes inside the network and replaces them by the peripheral equivalent nodes. So the reduced coarser structure can be analyzed fast and efficiently with few network scale. The final solution can be mapped back to the original power grid after the reduced equivalent grid is solved to get the grid parameters. Also, both static DC analysis and transient analysis can use the multigrid-like technique with high efficiency and acceptable accuracy.

Recently, the statistical approach methods are introduced into power grid analysis, such as the "Random Walks" algorithm [16–21]. The usual random-walk method bases on the grid node and starts the next step walk by the probability ratio. The continued walk traverses the power grid and analyzes the node information by the localizing computation. Also, the improved hierarchical method is used to build the multilevel and virtual-layer hierarchy instead of the basic single-level hierarchical method. The capacitor and inductor influences can also be analyzed by the statistical random-walk method to deal with the large scale network transient analysis efficiently, together with static DC precise analysis for million node level power distribution grid.

2.2.2 Power/Ground Network Synthesis Algorithms

The basic idea of power/ground network synthesis algorithm is focused on the network grid construction with layout area optimization and process time speedup. The common solution method is abstracting the P/G network to several kinds of linear or nonlinear systems under the constraints of signal integrity, circuit relationship, parasitic effects, etc. By different transmission methods, the nonlinear system can be converted to linear system, or complexity linear system can be replaced by simple linear system. Simultaneously, the system accuracy and operation efficiency can keep stable under the considerable value range. Finally, the simplified system equations and equivalent constraint conditions are solved together to design the P/G network grid with global layout area optimization.

Common P/G network synthesis includes the static and dynamic situation. The static synthesis means the network design with mainly consideration in the IR-drop and electromigration influence. For the dynamic case, Ldi/dt noise, Cdv/dt current, inductance-capacitance interaction, or other parasitic effects, are considered to deal with the network optimization and the system constraints. In addition, the topology design also influences the P/G network synthesis efficiency and optimization ratio. Thus in practical network grid design, the topology design, wire sizing and decoupling capacitance allocation, are used to improve the P/G network grid synthesis.

Because the static P/G network sizing neglects the Ldi/dt noise and the Cdv/dt current, the grid optimization mainly focuses on the voltage loss caused by the P/G network segment resistance. To solve the P/G network grid, the usual method is building the nonlinear equation with linear constraints to get the optimal grid area. In the equation solution procedure, many algorithms have been developed to deal with the problems, such as "Augmented-Lagrangian Method (ALM)" [22], "Conjugate-Gradient Method (CGM)" [23], "Feasible-Direction Method (FDM)" [24], SLP equivalent model or linear transformation algorithm [25, 26] and penalty nonlinear algorithm [27]. These algorithms convert the P/G network area sizing problem to the constraint nonlinear programming problem and solve the network grid optimization problem with fixed P/G network topology and resistor-only model.

The transient P/G network sizing considers the effects of resistance, capacitance and inductance in each segment. Because Ldi/dt noise, Cdv/dt current, IR drop and electromigration can cause the whole P/G network instability, the comprehensive effects and synthesis influences should be studied to design the suitable transient P/G network. Several algorithms are used to reduce the transient noise and optimize the global layout area under these circuit restraints. Some methods redesign the decoupling capacitance allocation and budgeting [28–31]. Other algorithms use the wire sizing based on the time-domain sensitivity to reach the P/G network optimization [8], or reduce the transient noise for P/G network grid optimization by the decap leakage current model consideration [32]. Because of the rapid increasing complexity of P/G network, many methods are used to reduce the model order and increase the process speed, including PRIMA [12], HiPRIME [13], Krylov [33], and Padé [34]. In addition, further P/G distribution network design methods have been proposed to reduce power noise [35], explore 3-D P/G grid model [36], study on-chip inductance effects [37], handle early power grid verification inductance [38], etc.

The topology allocation and floorplanning design also contribute to the P/G network synthesis and optimization. The Shiyou-Zhao method considers the decoupling capacitance (DECAP) allocation to suppress the power supply noise by floorplanning methodology [39]. The congestion-aware topology optimization algorithm divides the power grid into rectangular subgrids or tiles, and solve the congestion penalty function by efficient table lookup scheme, until the chip-wiring area global optimization [40]. Also, the partition-based algorithm uses the successive partitioning and grid-refinement scheme to realize the optimal power distribution network. By the idea of power grid locality properties, the distant nodes and sources effects are modeled more coarsely to enhance the convergence of the iterative conjugate-gradient-based solution [41]. In addition, the multigrid-based technique is proposed to deal with the on-chip power-supply network optimization problem by much coarser grid equivalence and back-mapping process resolution [42].

2.2.3 Other Considerations for Power/Ground Network

For the rapid increasing P/G network scale and frequency, the simple circuit model is not enough to describe the network grid performance. Then many new circuit models are given for precise P/G network analysis and synthesis. For example, the RLKC power delivery model is used to describe the power grid parasitic effects and realize the optimal P/G network synthesis [13]. Recently, another transmission line model (TLM) is used to describe the power grid in the chip manufacture, because of the TLM model integrated representation capability for the IR-drop, Ldi/dt noise, electromigration, or other parasitic effects [43]. Also, the alternating-direction-implicit (ADI) method is used to solve the TLM modeled P/G network grid much rapidly by linear time-space complexity and stable operation accuracy. But the TLM-ADI algorithm is just suitable for regular P/G network grid and the generic network situation needs further research to realize the effective P/G network grid synthesis.

The interaction of different elements and network in practical chip is necessary to be considered together. For example, the clock skew design and the P/G network power fluctuation can influence the corresponding circuit performance each other. So the synthesis consideration in the clock routing and power distribution is important for recent high performance large scale P/G network grid design [44]. Also, the co-synthesis of floorplan and P/G network is important to recent technology advance [45], where the B^*-tree floorplan representation is adopted and the "Simulated Annealing (SA)" method is used to solve the P/G mesh network. The IR-drop power integrity driven design methodology is realized by the macro current source modeling and the power mesh constraints. The floorplan and power integrity co-synthesis flow is effective for fast design convergence and P/G system optimization synthesis.

Table 2.1: Power/Ground network grid classical algorithms for synthesis design.

Characteristics	**Realization Methods**	**Authors**	**Publications**
Static IR-Drop Electromigration	Augmented Lagrangian	S. Chowdhury M. A. Breuer	IEEE TCAS, 1987 [22]
	Sequence of Linear Programming (SLP)		IEEE TCAD, 1988 [46]
	Conjugate Gradient		DAC, 1989 [23]
	Feasible Direction Method (FDM)	R. Dutta M. M. Sadowska	DAC, 1989 [24]
	Nonlinear Equation Solving, (Penalty Method, Network Merge)	X. H. Wu, X. L. Hong, Y. C. Cai, etc	IEEE TCAD, 2004 [27]
	Equivalent Model SLP	S. X.-D. Tan, R. C.-J. Shi	IEEE TCAD, 2003 [25] IEEE TCAD, 2003 [26]
Dynamic IR-Drop Wire Sizing	Congestion Driven, Hierarchical Analysis	H. H, Su, S. S. Sapatnekar	IEEE TDTC, 2003 [7] IEEE TCAD, 2004 [8]
Dynamic IR-Drop Decoupling Capacitance Topology Placement	Decoupling Capacitance, Placement	G. Bai, S. Bobba, I. N. Hajj	ICCAD, 2000 [28]
	Decoupling Capacitance, Hot Spot Analysis	H. H. Chen, D. D. Ling	DAC, 1997 [47]
	Decoupling Capacitance, Budget and Floorplanning	S. Y. Zhao, K. Roy, C.-K. Koh	DAC, 1997 [39]
Super Large Scale P/G Network Synthesis	Congestion-aware Topology Optimization	J. Singh, S. S. Sapatnekar	IEEE TCAD, 2005 [40]
	Partition-based Algorithm		IEEE TCAD, 2006 [41]
	Multigrid-based Technique	K. Wang, M. Marek-Sadowska	IEEE TCAD, 2005 [42]

Table 2.2: Power/Ground network grid classical algorithms for system analysis.

Characteristics	**Realization Methods**	**Authors**	**Publications**
Large Scale P/G Network Simulation	Asymptotic Waveform Evaluation (AWE)	L. T. Pillage, R. A. Rohrer	IEEE Trans. CAD, 1990 [11]
	Passive Reduced-order Interconnect Macromodeling Algorithm (PRIMA)	Altan Odabasioglu, Mustafa Celik, L. T. Pillage	IEEE Trans. CAD, 1998 [12]
	Padé approximants	J. C. Shah, A. A. Younis, S. S. Sapatnekar, M. M. Hassoun	IEEE Trans. CASII, 1998 [34]
RLKC Power Delivery Network Analysis	Preconditioned Krylov-Subspace Iterative Methods	T. H. Chen, C. C. P. Chen	DAC, 2001 [48]
	Hierarchical/Passivity Reserved Interconnect Macromodeling Engine (HiPRIME)	Y. M. Lee, Y. H. Cao, C. C. P. Chen, etc	IEEE Trans. CAD, 2005 [13]
Super Large Scale P/G Network Grid Analysis	Multigrid-Like Technique	J. N. Kozhaya, S. R. Nassif, F. N. Najm	IEEE Trans. CAD, 2002 [15]
	Hierarchical Analysis Method	M. Zhao, R. V. Panda, S. S. Sapatnekar, D. Blaauw	IEEE Trans. CAD, 2002 [14]
Other P/G Network Analysis Mode	Random Walks	Haifeng Qian, Sani R. Nassif, Sachin S. Sapatnekar	IEEE Trans. CAD, 2005 [17]
	Transmission-Line-Modeling Alternating-Direction-Implicit Method (TLM-ADI)	Y. M. Lee, C. C. P. Chen	IEEE Trans. CAD, 2002 [43]

Recently, the P/G network grid synthesis has been extended to further research challenge and co-design system implementation. such as power and timing optimization in voltage island aware floorplanning [49], via stapling 3-D IC design with simultaneous power and thermal integrity consideration [50], power delivery networks optimization with thermal reliability integrity [51], static timing analysis considering both IR and Ldi/dt drops for power supply variations [52], fast decap allocation based on algebraic multigrid [53], power grid physics and implications for CAD [54], fast traversal algorithms for IR drop analysis in extremely large size power grid [55], parallel-distributed "Latency Insertion Method (LIM)" with effective frequency-dependency modeling [56], dual VDD power distribution circuits [57], power grids analysis and verification considering process-induced leakage-current variations [58], or other P/G network synthesis algorithms.

Many classical P/G synthesis and analysis algorithms are shown in Table 2.1 and Table 2.2 to illustrate the P/G network grid development period and emphasize the representative algorithms. Also, the detailed introduction is given in this chapter and the following section descriptions. The next step research targets, such as faster processing speed, larger chip scale, smaller circuit dimension, nanometer interconnect, DFM research, or other P/G network grid synthesis improvements, are expected for the future P/G network design and manufacture.

2.3 Systolic Array Network Related Research

The rapidly increased requirements for large scale data operation can promote the development of digital signal processing (DSP) systems. Fast computation speed, large scale processing capability, operation accuracy with stabilization, or other system pipelined merits, are necessary for high speed VLSI system implementation. As one of the effective DSP architectures, the systolic array is used widely in the related product application and consumer electronics market. Also, the systolic network grid is the core element to determine the whole system operation efficiency. Many researches are employed in array architecture improvement and network grid optimization to high performance DSP chip design and VLSI system manufacture.

2.3.1 Systolic Array Network Development History

With the merits of highly concurrent process, large throughput latency and global pipelined operation, systolic array architectures are widely used in high speed VLSI implementation, fast DSP element processing and high performance computer design [59,60]. Basic systolic architecture is the global connection grid among the processing elements (PEs) in the DSP system as shown in Figure 2.2. The data input and output are operated with highly pipelined thread, and system processing function is realized by skillful processing element allocation and careful connection wire adjustment.

Since the systolic array introduction in the late 1970's guided by H. T. Kung, many systolic array architectures are implemented to increase the VLSI operation speed [6, 61–66]. Other systolic adjustments, such as error operation diagnosis, fault tolerance design, and processing time optimization, are realized to improve traditional systolic array performance from 1980's to 1990's [67–74]. Moreover, in the recent decade, the systolic array application is enlarged greatly in the high performance VLSI processor systems implementation [75–80]. Also, the booming consumer electronics market requires much rapidly pipeline processing chip and many practical electronics products use the systolic array architecture for mass information processing [81–86].

2.3.2 Systolic Array Grid Application Description

Basic systolic array architecture is composed of regular processing element (PE) arrangement and highly concurrent data computation [59]. The highly concurrent data matrix and the fully pipeline operation enable the mass data processing and high performance VLSI operation. To improve the systolic array computation accuracy, the algorithm-based fault tolerant matrix method is introduced to verify the operation result and keep the final data operation efficiency [61]. Also, the concurrent error diagnosable systolic arrays, are proposed to satisfy the corresponding VLSI system design requirements, and the new systolic array architecture with better throughput latency and utilization rate is given for effective systolic chip design [6]. Many further modified architectures are proposed to design the suitable circuit configuration for various systolic arrays, such as planar VLSI systolic array [62], optimal linear arrays for matrix multiplication [63], transitive closure systolic array [64], large fixed size systolic array [65], and bit recursive systolic array design [66].

Other types of array configurations are developed to enlarge the application range of systolic array architecture and satisfy the different DSP design requirements. For example, the fraction dimensional systolic array is given for non-integrity multiplication operation or pseudo-three-dimensional network, where the matrix-vector and matrix-matrix multiplication are computed to achieve better throughput rate than conventional systolic array under same processing elements [67]. Also, the parallel matrix multiplication design of cylindrical array and the two-layered mesh array describes the combinatorial systematic design procedure. The different operation layer and the cylindrical connection enable the effective matrix mesh architecture design for multiplication computation and parallel algorithm realization [68]. In addition, the special size array architectures are useful for some VLSI system implementation, such as the triangular systolic array architecture application. The transformation pipelined implementation uses the triangular array and the different processing element types are selected to achieve the highly concurrent computation by operation sequence allocation and processing element combination [69]. Furthermore, the diversity systolic array architectures, such as iteration vector combination [70], matrix-vector systolic multiplication [71], triangular systolic array [72], redundancy systolic array [73], and pyramid systolic array architecture [74], can provide broad design possibility and various application range for high performance VLSI system implementation.

Moreover, most recently systolic array developments are focused on the VLSI system application. For example, the multidimensional systolic array is introduced to discrete Fourier transform (DFT) and fast Fourier transform (FFT). By the combination of two distinct semi-systolic arrays into one truly systolic array, the resulting systolic array accepts the input data stream and produces the output stream in the array boundary with no intermediate spectrum transposition and simpler regular processing element connections [75]. Also, the Montgomery multiplication concurrent array in the cryptography and the cryptosystem uses the semi-systolic architecture for concurrent error detection and fault diagnosis tolerance. The bit-parallel architecture by $GF(2^m)$ enables the Montgomery multiplier with concurrent error detection capability and few extra space overhead, compared with basic Montgomery multiplier [76]. Other recent systolic array improvements include Euclid systolic array [77], digit-serial systolic array [78], wide I/O band-width systolic array [79], and cubic matrix systolic architecture [80], etc.

Additionally, the practical consumer electronics product requires fast operation speed, compact layout area, and error diagnosis robustness for systolic array application. The widespread field-programmable gate array (FPGA) system uses the novel systolic array-based architecture to realize the time multiplication and arithmetic pipeline design. The power consumption and adaptive algorithm decoder are also implemented by the pair architecture of systolic arrays [81]. Other efficient VLSI designs for the computation of discrete cosine transform (DCT), discrete sine transform (DST), inverse discrete cosine transform (IDCT) and inverse discrete sine transform (IDST), also use the systolic algorithms for superior performance system. The hardware complexity, processing speed and I/O costs are improved and the regularity, modularity, pipelining capability, and local connectivity are approached for efficient VLSI system design and DSP processing implementation [82]. Also, many systolic array considerations for consumer electronics applications to our modern enjoyable life, are presented in recent years, including CDMA array implementation [83], video code/decode systolic VLSI circuit [84], 2-D DFT pipelined systolic array [85], and RSA systolic array architecture [86], etc.

2.3.3 Matrix Multiplication Systolic Array

In this thesis we develop new two-dimension systolic array architectures for the matrix multiplication and optimize the array efficiency on throughput latency, utilization rate and operation speed.

The multiplication operation is the basic algorithm for the VLSI design and the DSP system. Because the speed of multiplication usually limits the operation rate of the VLSI/DSP and confines the application of the whole systems, many people studied the multipliers and proposed lots of algorithms, architectures, and technologies to improve the multiplier functions. Modified Booth multiplier is one of the fundamental multiplication structure [87], and the pipelined architectures can also increase the speed and efficiency of the multiplication operation [88, 89]. In addition, the systolic array architecture is introduced for matrix multiplications operation to satisfy the requirements of concurrent operation and pipeline process, which means many computations are operated synchronously [6, 59].

The fundamental two-dimension systolic array architectures for matrix multiplication are introduced mainly by Kung-Leiserson [59] and Huang-Abraham [61]. The regular systolic array allocation for the processing elements enables fast pipeline computation and compact VLSI system area. The data matrices keep the operation accuracy and realize the multiplication efficiency. To satisfy the operation requirements, the matrices are loose and the utilization rate is not fully concurrent.

To increase the matrix multiplication efficiency, many advanced systolic array configurations are developed to achieve more fully concurrent operation. For example, Li proposed the space-time transformation for better systolic array design. The parallel systolic arrays are assembled together to reduce the data flow latency, minimize the operation space time, and enable the concurrent error detection [90]. Another fault-tolerant design also considers the optimal systolic array approach for VLSI processors. The mapping theory enables the error diagnosis and optimal space time. The systolic throughput is also improved and the multiplier performance is advanced to better VLSI implementation [91], similarly to the systematic fault-tolerant approach design [92]. Additional latency reduction algorithm is realized by the data matrices sequence adjustment, where the input multiplication data are checked and the previous time allocation is used under the special algorithm to enable the final operation accuracy for the compact matrix multiplication [93–95].

There are many planar systolic array topologies, such as linear connection, square topology, and hexagonal topology. To optimize the diverse systolic array efficiency, many methods are introduced and implemented. One conventional method is the processing element variety, where the different element types are used to realize complex systolic function and the suitable element organization enables the final computation feasibility [69]. The above data matrices adjustment is another effective systolic design method. The matrix sequence can be rearranged to get the suitable positions for next step data input into the systolic array by fast matrix multiplication with few throughput latency and better utilization rate [94]. Moreover, the broadcast vector propagation provides the connection wire expansion method to increase the systolic concurrent range and raise the global pipeline possibility. The systolic array grid extension and small area connection interlacement are useful for the high performance systolic array implementation and more fully pipeline thread design [96, 97].

2.4 Network Grid Background Conclusion

The process technology scaling and the operation frequency increasing have influenced the VLSI circuit performance. The connection network in the chip design, as one of the important circuit elements, should be considered carefully in the optimization and operation improvement to satisfy the recent SoC design challenges.

In this thesis many researches have been introduced for the network grid design, including the power/ground network analysis or synthesis algorithms, and the systolic array performance improvement methods. The merits and demerits of these methods are also described in detail and the possible improvement directions are also given for the next step research schedule.

The following chapters will propose new methods for high performance network grid design in VLSI system, such as the genetic algorithm or hybrid genetic method for P/G network grid synthesis, and the super pipeline or data jumping methods for two-dimension matrix multiplication. Based on the conventional network algorithms, the novel grid synthesis methods can optimize the network layout area and increase the global operation efficiency for recent VLSI system design.

Chapter 3
POWER/GROUND NETWORK GRID SYNTHESIS

3.1 Power/Ground Network Grid Challenges

The power/ground (P/G) network grid in recent VLSI chip design faces the restrictions of power signal propagation fluctuation as shown in Figure 3.1, which is caused by IR drop, Ldi/dt, Cdv/dt, electromigration, or other parameter effects. To satisfy the P/G network grid design requirements, many efforts are used to counteract the fluctuation effects, such as wire sizing, decoupling capacitance, or topology adjustment [7,8]. This thesis proposed the novel P/G network grid synthesis methods with global optimization and colony computation for high performance network design.

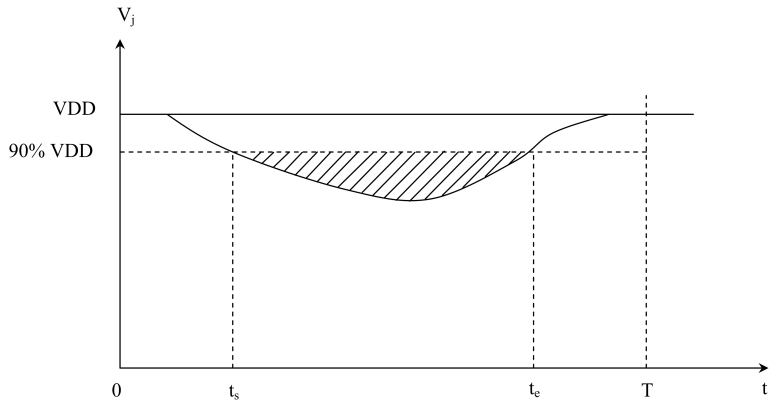


Figure 3.1: Power signal propagation fluctuation effects [7,8].

In this section we propose the novel "Grid Synthesis using Genetic Algorithm (GGA)" to realize global optimization for P/G network synthesis with IR drop consideration. Also, we propose the "Hybrid Grid Synthesis based on Genetic Algorithm (HGGA)" to deal with dynamic signal situation with Ldi/dt noise and Cdv/dt decoupling capacitance (DECAP) current in P/G distribution network.

3.2 *Grid Synthesis using Genetic Algorithm (GGA)*

3.2.1 Power/Ground Network Grid Problem Formulation

One example of P/G network $G = \{N, B\}$ with n nodes $N = \{1,\ldots,n\}$ and b branches $B = \{1,\ldots,b\}$ is shown in Figure 2.1. In B the segment between nodes i_1 and i_2 is the branch i with length l_i and width w_i, respectively. If we suppose ρ is the sheet resistivity, then the resistance r_i of branch i is [1, 25, 26]:

$$r_i = (V_{i1} - V_{i2})/(I_i) = \rho(l_i)/(w_i). \tag{3.1}$$

3.2.1.1 P/G Network Grid Objective Function

Our aim of the P/G network sizing can be realized by minimizing the total P/G routing area in terms of voltages, currents, and lengths of branches as below [25, 26]:

$$f(V, I) = \sum_{i \in B} l_i w_i = \sum_{i \in B} \frac{\rho I_i l_i^2}{V_{i1} - V_{i2}}. \tag{3.2}$$

In this thesis we suppose the length of each branch is constant because the P/G network connection is decided previously by the designers. Thus the routing area is related with the branch current variables I and node voltage variables V.

3.2.1.2 P/G Network Grid Synthesis Constraints

To satisfy the feasibility and reliability of the P/G network grid, its routing area is subject to the following synthesis constraints:

1. *Voltage IR drop constraints*: On the P/G network the voltage drop (IR drop) from PAD to leaf nodes should not exceed the restriction bounds, otherwise the chip cannot work correctly. The detailed voltage constraints are [25, 26]:

$$\begin{aligned} V_{i \in N} &\geq V_{min}, && \text{for power network,} \\ V_{i \in N} &\leq V_{max}, && \text{for ground network,} \end{aligned} \tag{3.3}$$

where V_{min} and V_{max} are the lower and upper restriction bounds, which are decided by the given layer parameters.

2. *Minimum width constraints*: The P/G branch widths have minimal restriction bounds because of the technological restrictions on the layers where the P/G branches exist. These bounds are described as follows [25, 26]:

$$w_{i \in B} = \frac{\rho l_i I_i}{V_{i1} - V_{i2}} \geq w_{i(min)}, \tag{3.4}$$

where the parameter $w_{i(min)}$ is width bound.

3. *Electromigration constraints*: In the P/G network electromigration is caused by high current density and has an upper limit. In the routing layer with fixed thickness, the detailed limits are [25, 26]:

$$\begin{aligned} &\frac{|I_i|}{w_i} \leq \sigma, \quad \text{or} \\ &|V_{i1} - V_{i2}| \leq \rho l_i \sigma, \end{aligned} \tag{3.5}$$

where σ is the current density for fixed thickness.

4. *Equal width constraints (coupling constraints)*: To satisfy the demands of chip cost and performance, most adjacent P/G branch segments should not be changed largely. Thus all the segments in a chain can be considered to share the same width. The constraint can be expressed as $w_i = w_j$, or [25, 26]

$$\frac{V_{i1} - V_{i2}}{l_i I_i} = \frac{V_{j1} - V_{j2}}{l_j I_j}. \tag{3.6}$$

5. *Kirchoff's current law (KCL)*: The current influx and efflux in each node of the P/G network should keep balance. That is [25, 26]:

$$\sum_{i \in B(j)} I_i = 0, \tag{3.7}$$

where $B(j)$ is the set of branches to node j.

The P/G network grid optimization problem is to get the minimal P/G routing area under the constraints of voltage IR drop, minimum width, electromigration, equal width and Kirchoff's current law. From Eq. (3.2) to Eq. (3.7) the formulated problem is a nonlinear problem with its nonlinear constraints.

3.2.2 Conventional SLP Optimization Methods

3.2.2.1 Basic Two-Phase SLP Algorithm

In 1989 Chowdhury proposed basic SLP (Sequence of Linear Programming) method to solve the P/G problem [23]. The method separates the original problem P into two phases ($P-V$, $P-I$), and only considers one variable in each phase, then iteratively solves $P-V$ and $P-I$ until the area result reaches the minimal value.

1. *Phase one (P-V)*: In this phase all branch currents are fixed and all node voltages are variables [25, 26]:

$$f(V) = \sum_{i \in B} (\rho I_i l_i^2) \times (\frac{1}{V_{i1} - V_{i2}}). \tag{3.8}$$

   Because phase $P-V$ is still nonlinear problem, it is further relaxed by Taylor's expansion around the initial solution if the feasible result exists. Then the original problem P is converted to linear problem and can be back solved by linear programming (LP). The final objective function is [25, 26]:

$$\begin{aligned} f_V(V) \ &= \ \sum_{i \in B} \frac{2\alpha_i}{(V_{i1}^0 - V_{i2}^0)} \quad - \\ &\quad \sum_{i \in B} \frac{\alpha_i}{(V_{i1}^0 - V_{i2}^0)^2}(V_{i1} - V_{i2}), \end{aligned} \tag{3.9}$$

   where the parameter $\alpha_i = \rho I_i l_i^2$.

   The optimization constraints are Eq. (3.3) - Eq. (3.7) with following equations:

$$\frac{V_{i1} - V_{i2}}{I_i} \geq 0, \tag{3.10}$$

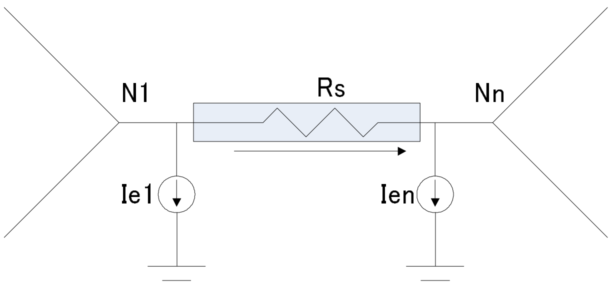


Figure 3.2: Equivalent circuit model for the series-resistor chain.

$$\xi sign(I_i)(V_{i1}^0 - V_{i2}^0) \leq sign(I_i)(V_{i1} - V_{i2}), \tag{3.11}$$

where I_i is the given constant and $\xi \in (0, 1)$.

2. *Phase two (P-I)*: In this phase the voltages in each node are fixed and the branch currents become the variables. Similarly, the objective function can be expressed as follows [25, 26]:

$$f_I(I) = \sum_{i \in B} \beta_i I_i, \tag{3.12}$$

where the parameter $\beta_i = (\rho l_i^2)/(V_{i1} - V_{i2})$, and the related constraints are given in Eq. (3.4), Eq. (3.6) and Eq. (3.7).

3.2.2.2 Equivalent Model SLP Algorithm

Based on the two-phase SLP method, Tan and Shi proposed new SLP algorithm with equivalent circuit modeling [25, 26]. By exploring the architecture of P/G network grid and the relationship of all node voltages with branch currents, Tan and Shi reduce the P/G network complexity by using the equivalent circuit modeling and apply the basic two-phase SLP method to realize the optimal routing area of the P/G network grid more efficiently.

Shown in Figure 3.2 the equivalent resistance R_s is the sum of all the resistances in series. And the equivalent end current I_{e1} and I_{en} replace the contributions from all the current sources. Then the P/G network grid complexity is reduced rapidly and the synthesis equations can be solved more quickly. Once the voltages at end nodes are known, the intermediate node voltages can be back solved step by step [25,26]:

$$R_s = \sum_{i=1}^{n-1} R_i, \tag{3.13}$$

$$I_{e1} = \sum_{i=1}^{n-2} \frac{\sum_{j=i+1}^{n-1} R_j}{R_s} I_i, \tag{3.14}$$

$$I_{en} = \sum_{i=1}^{n-2} \frac{\sum_{j=1}^{i} R_j}{R_s} I_i, \tag{3.15}$$

$$V_{i+1} = V_i - \frac{R_i}{R_s} V_s - R_i I_{ei}, \tag{3.16}$$

$$I_{e(i+1)} = I_{ei} - I_i. \tag{3.17}$$

After the reduced equivalent network is constructed, the SLP algorithm is used with new constrains as the following relationship [25,26]:

$$\begin{aligned} |V_{s1} - V_{s2}| &\leq \frac{\rho l_s \sigma}{1 + \frac{I_{e1}}{I_s}}, && \text{for power network,} \\ |V_{s1} - V_{s2}| &\leq \frac{\rho l_s \sigma}{1 + \frac{I_{en}}{I_s}}, && \text{for ground network.} \end{aligned} \tag{3.18}$$

Then the new phase $P - V$ is minimizing the function Eq. (3.9) under the constraints Eq. (3.3), Eq. (3.4), Eq. (3.6), Eq. (3.11), and Eq. (3.18) by using the SLP method. Similarly, the new phase $P - I$ can get the minimal objective function Eq. (3.12) under the constraints of Eq. (3.4), Eq. (3.6), Eq. (3.7), and Eq. (3.18).

From an initial solution the Tan-Shi equivalent model SLP algorithm iteratively solves two new linear programming problems: $P - V$, then $P - I$, until the ultimate result is reached.

3.2.3 Power/Ground Network GGA Method

Developing new method is essential because the SLP method can not reach the minimal result under the normal situation. In addition the linear transformation of the

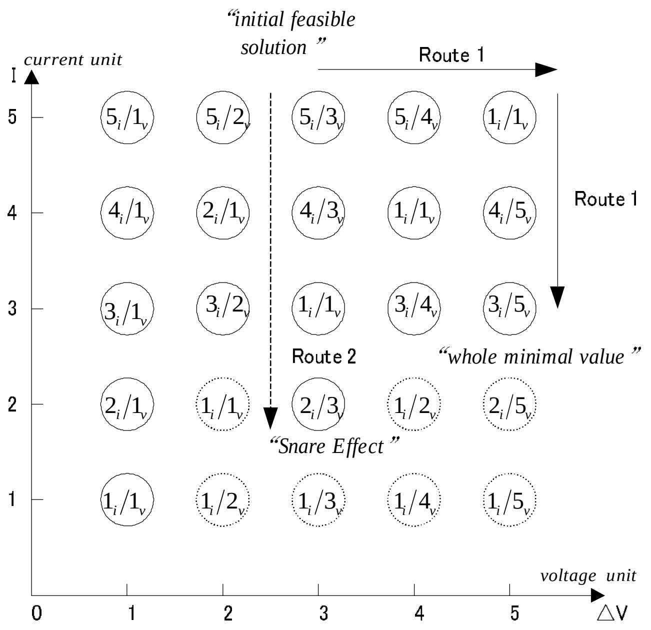


Figure 3.3: Snare effect in Tan-Shi equivalent model SLP algorithm. (1) X_i/Y_v means X unit current with Y unit voltage. (2) The dashed circles mean these values not exist. (3) Route1: ($5_i/3_v \Rightarrow 5_i/4_v \Rightarrow 1_i/1_v \Rightarrow 4_i/5_v \Rightarrow 3_i/5_v$). (4) Route2: ($5_i/3_v \Rightarrow 4_i/3_v \Rightarrow 1_i/1_v \Rightarrow 2_i/3_v$).

SLP method also has some problems. Based on the analogy between the P/G network optimization and the rapid colony evolution in biology science, we propose new "Grid Synthesis using Genetic Algorithm (GGA)" to solve the P/G network global optimization problems.

In this section, we first point out the deficiencies in Tan-Shi equivalent model SLP algorithm. Then we describe new GGA method in detail and show its process flow.

3.2.3.1 Deficiencies of SLP Algorithm

1. *Snare Effect*: Tan-Shi equivalent model SLP algorithm cannot reach minimal P/G routing area in practical circuits. From equation Eq. (3.2) we can get:

$$f(V,I) = \textstyle\sum_{i\in B} l_i w_i = \sum_{i\in B} \frac{\rho I_i l_i^2}{V_{i1}-V_{i2}}$$
$$= \textstyle\sum_{i\in B} \rho l_i^2 \frac{I_i}{V_{i1}-V_{i2}} = \sum_{i\in B} kg(\Delta V, I), \tag{3.19}$$

where $k = \rho l_i^2$, $g(\Delta V, I) = \frac{I_i}{\Delta V_i} = \frac{I_i}{V_{i1} - V_{i2}}$.

Figure 3.3 shows the optimization flag $g(\Delta V, I)$ in different routes. If all the values in the route exist, Tan-Shi equivalent model SLP algorithm can get the minimal result. However, in practical circuits the values of ΔV and I are subject to constraints Eq. (3.3), Eq. (3.4), Eq. (3.6), Eq. (3.7), Eq. (3.11) and Eq. (3.18). Consequently, many values in the routes do not exist and Tan-Shi equivalent model SLP algorithm cannot reach real P/G network grid optimization.

For example, route2 path is $(5_i/3_v \Rightarrow 4_i/3_v \Rightarrow 1_i/1_v \Rightarrow 2_i/3_v)$. Because the values around $(2_i/3_v)$ do not exist, the route2 cannot go ahead and the final result is $(2_i/3_v)$. In fact the global minimal value is $(3_i/5_v)$ if we choose route1 $(5_i/3_v \Rightarrow 5_i/4_v \Rightarrow 1_i/1_v \Rightarrow 4_i/5_v \Rightarrow 3_i/5_v)$. We name these local minimum situations as the "*Snare Effect*". Thus the new method for global optimization is required.

2. *The solution space limit ξ*: In phase $P - V$, the original problem is nonlinear programming problem in terms of V shown in Eq. (3.8). To replace it to linear programming problem the sequence-of-linear-programming (SLP) method was suggested by Tan and Shi [25, 26]. Using the Taylor expansion the objective function can be transformed to Eq. (3.9). However the solution space limit $\xi \in (0, 1)$ in Eq. (3.11) restricts the optimization step and increases the sizing complexity and time. Thus the limit should be improved to eliminate the Taylor expansion disadvantages.

3. *$P - I$ phase's problem in practice*: In the individual element situation the $P - I$ phase of Eq. (3.12) is linear programming problem, but elements in the P/G network is involved with adjacent circuits. Therefore the phase $P - I$ is not always linear. For instance in Figure 3.4, $I_B = I_C - I_2$. When I_C decreases I_B also reduces its value, and $f_I(I)$ value in Eq. (3.12) changes in the same direction. However when I_C drops its value below I_2, $I_B < 0$ and the absolute value of I_B

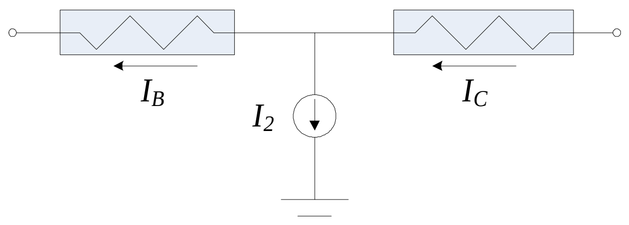


Figure 3.4: $P - I$ phase problem illustration in practice circuit model.
$I_B = I_C - I : I_C\searrow\Rightarrow|I_B|\searrow\Rightarrow P/G\ Area\searrow, (I_B > 0)$
$I_C\searrow\Rightarrow|I_B|\nearrow\Rightarrow P/G\ Area\nearrow or\searrow.(I_B < 0)$
$\Longrightarrow P - I$ phase is no longer linear programming problem.

increases with the I_C reduction. Then the P/G network area, which is related with the absolute value of branch currents in $P - I$ phase, increases its value when I_C is reduced. But the correct linear programming characteristic must own the fixed change direction for the network area and the current I_C. If I_C rises, the network area increases (or decreases). Reversely, when I_C drops, the area also decreases (or increases). Thus Tan-Shi equivalent model SLP algorithm loses the linear programming characteristic in network grid optimization and cannot deal with whole network current interaction. As a result, the phase $P - I$ should be modified and improved by better P/G network sizing method.

3.2.3.2 GGA Method Description

In the nature world the species evolution is rapid and optimal under the environment constraints. And the gene selection and mutation rules define the characteristics of each successful species. Similarly, the P/G network grid optimization is finding the optimal P/G sizing area under several constraints. Inspired by the analogy between biology evolution and P/G network optimization, we propose the novel method of "Grid Synthesis using Genetic Algorithm (GGA)" for P/G network grid synthesis.

Basic gene algorithm was presented by J. H. Holland in 1960's as shown in Figure 3.5 [98]. By the individual encoding, the processing targets can be expressed by

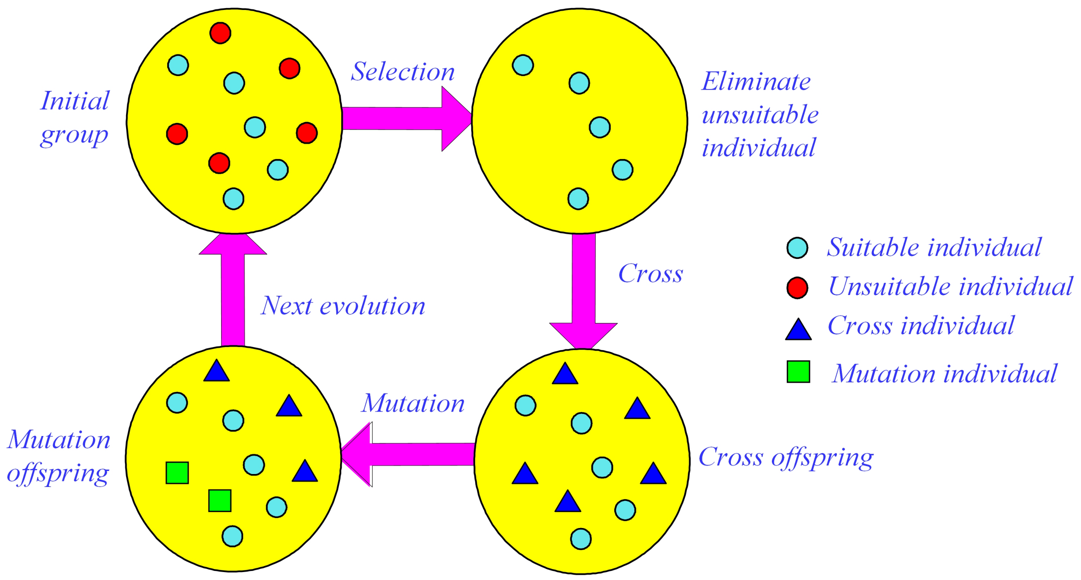


Figure 3.5: Basic genetic algorithm evolution schematic diagram.

the binary coding. The system optimization can be realized by the colony evolution operation. Also, common genetic evolution includes several steps, such as crossover, mutation, selection and new generation. By these sequent operation, the suitable individuals under the environment constraints can survive to deduce the best evolution generation, which is the global optimization results as the analogy with the P/G network grid synthesis design.

We change the individual genetic code to satisfy the demands of the network grid and reduce the process complexity. In our method only the independent variables are coded. Other branch currents and nodal voltages are related with the independent variables and can be computed based on the P/G network configuration. Then we let the P/G gene “crossover” and “mutation” under the restrictions of “environment selection”. After many generation genetic evolution we can get the minimal P/G network layout area. Since the benefits of colony search, fast convergence, and global optimization, the GGA method can approach the minimal P/G network area if the evolution generations are large enough. In addition, another merit is that the GGA method can process the objective function directly and avoid the error in the trans-

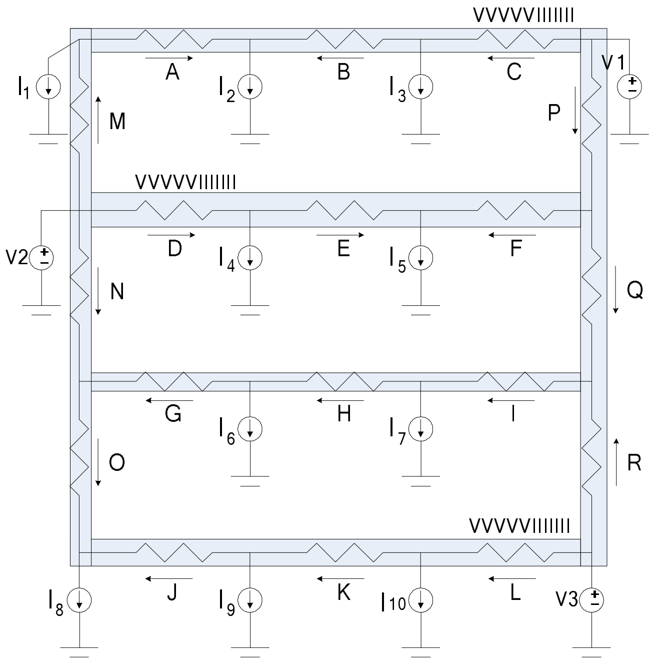


Figure 3.6: The network grid example of 4×4 P/G network and independent variables with the code "$VVVVVIIIIIII$" in the network, where V means the voltage bit and I means the current bit.

formation from nonlinear problem to linear problem, which is the crucial obstacle in the recent research of the P/G network grid optimization.

As an illustration Figure 3.6 shows a 4×4 P/G network grid example with 3 segments in each chain [25, 26]. There are 18 branches from 'A' to 'R', as well as 16 nodes in the P/G network. In addition, three voltage sources and ten inner current outputs compose the whole P/G network. The detailed explanation of the new GGA P/G network optimization algorithm is shown as follows:

1. *Find the independent variables*:

   Based on Tan-Shi equivalent model SLP Algorithm, the widths of all the branches in a chain are identical [25,26]. Thus each branch in a chain can be expressed by one independent variable. For an $m \times n$ P/G network with m rows, n columns and several strips, the numbers of independent variables are $(m-1)\times(strip-1)$.

For example, in Figure 3.6 the top route of P/G network grid has three segments A, B and C. We select the segment C as the independent variable. Similarly, other variables are segment D and segment L in the whole P/G network.

2. *Set the genetic codes of independent variables*:
   After the independent variables are found, the genetic codes are given to prepare the genetic algorithm. Each branch includes two parts: the branch current I_i and the voltage difference DV_i between its two nodes. The values of I_{max} and DV_{max} are determined by the given chip parameters. In GGA method the I_{max} means the summation of all the branch currents flowing out of circuit, and the DV_{max} is the maximal voltage difference in the P/G network. In the P/G network segments each part is signed and defined by the binary digits. And the appropriate bit distribution is crucial for the result precision and process time. For the ±3.0V voltage source, the common switch threshold is 0.3V, which is about 1/10 of the source voltage. Thus we can use five bits to code the voltage difference DV_i. Then each step of the voltage difference is $(DV_{max}/16)$ with the maximal period $(-DV_{max})\sim(+DV_{max})$ in each segment. Similarly, the 1% variation for the branch current I_i is suitable for the evolution requirements of precision and time. Thus we use the seven bit coding for the branch current I_i. Then the scope for I_i is $(-I_{max})\sim(+I_{max})$ and its scale is $(I_{max}/64)$. By the combination of I_i and DV_i we can get the final 12 bits genetic codes "$VVVVVIIIIIII$" for the independent variables in the whole P/G network.

3. *Population initialization*:
   Each feasible result of the P/G network grid is called "individual" which has the analogy with single creature in the nature. And just as many creatures compose the colony, several individuals comprise the "population". From the independence tests, some initial feasible results are obtained and set as the population. The optimal P/G network grid synthesis is evolved from the population pool

because most suitable genes are included in these feasible results or developed from the pool gene storage.

4. *Crossover*:

   In the propagation the genes in same place of different individuals are exchanged to get the next generation. This transform period is named as "crossover". Two steps compose this process:

   (a) *Grouping*: For population crossover these individuals are randomly divided into some groups which contain two individuals as the combination of male and female.

   (b) *Crossing*: After grouping each group's genes are exchanged at random. This is called "crossing". In GGA method the exchange method is updated to satisfy the P/G network design requirements. At first one random value 'n' is generated, which is not more than the genetic codes bit number. Then two initializations in one group exchange their last 'n' bits in terms of the crossover probability P_c(25%∼75%). Unlike the random crossing for basic gene algorithm, the directional end bit crossing of the GGA method can reduce the evolution complexity and increase the process speed.

5. *Mutation*:

   Mutation is the necessary step which changes the special positions of individual genetic codes. Because not all new genes can be created by the crossover, several code segments in special position alter to engender the new offspring. The GGA method firstly gets the special positions in the genetic codes at random, then inverses the selected position value under the restrictions of mutation probability P_m whose typical value is between 1% to 20%. The mutation step of GGA method can guarantee the searching completeness and reach the proper results. The selection of P_m can also influence the evolution speed and complexity.

6. *Compute the whole P/G network parameter*:

   In spite of the width equality, the P/G network is subject to $Kirchoff$'s current law. Thus the branch currents and nodal voltages can be calculated from the known individual genetic codes, such as the 4×4 P/G network in Figure 3.6 where the branch B current is $I_B = I_C - I_2$, and its voltage difference $DV_B = DV_C \times (L_B/L_C) \times (I_B/I_C)$. In the same way the P/G network area and other branch parameters can be acquired under the P/G network restrictions.

7. *Selection*:

   Based on the survival of the fittest theory the next generation is selected under the environment condition. If the individual's adaptive ability is higher than others, its genes can be preserved and inherited to offspring in all possibilities. The next generation satisfies the surroundings more than its parents. In the P/G network problem the new generation is better result for the area optimization. The environment condition is the P/G network restrictions. In GGA method the selection is subject to the limits of Equations (3.3)-(3.7) and avoids the difficult transformation from nonlinear problem to linear problem. Thus the GGA selection is the attempt and approach to the optimal result as follows:

   (a) *Individual selection*: After the crossover and mutation in each group, the GGA method selects those new results under Equations (3.3)-(3.7) limits. If the results are suitable for restrictions, they are preserved into population gene pool. Otherwise, those unsuccessful results are rejected and their parents are forced to reproduce again, until the suited results are created. If the propagation times exceed the maximum number M_i, the individual selection stops and the parents in the group are kept. These suitable parent genes are used to attend the next generation propagation.

   (b) *Population selection*: For the given group results the GGA method calculates each group's whole network area. If it is more optimal result than the last evolution, the area and related parameters in each segment are recorded as this evolution computing result. Otherwise, this selection fails and GGA method comes back to continue crossing and repeats the above steps. The whole P/G network is evolved again to get the new generation.

8. *Convergence to the global optimum*:
   This step gives an ending flag to reach the global P/G network layout optimum. There are two flags to determine end points when the evolution finishes.

   (a) *Global optimum*: If the global P/G network area keeps stable and just swings in a given little range A_s, the P/G network can be regarded to reach the optimum. Then GGA method finishes and sends out the result.

   (b) *Evolution maximum*: If the network cannot get the steady result and the global optimization is not convergent, the evolution period will come into the endless circle. Thus setting the maximum evolution number M_e is necessary. When the evolution numbers exceed M_e, the GGA method's optimization process stops and the final result is the optimal result acquired in the previous generations, which is considered as the global optimum result of the whole P/G network grid.

   If the above flags cannot be reached, GGA method returns to grouping and iterates the above related steps.

3.2.3.3 GGA Method Procedure Flow

The flow chart of entire GGA method procedure is shown in the following Figure 3.7. And the chart gives more detailed explanation of GGA method process which is discussed in the above sections.

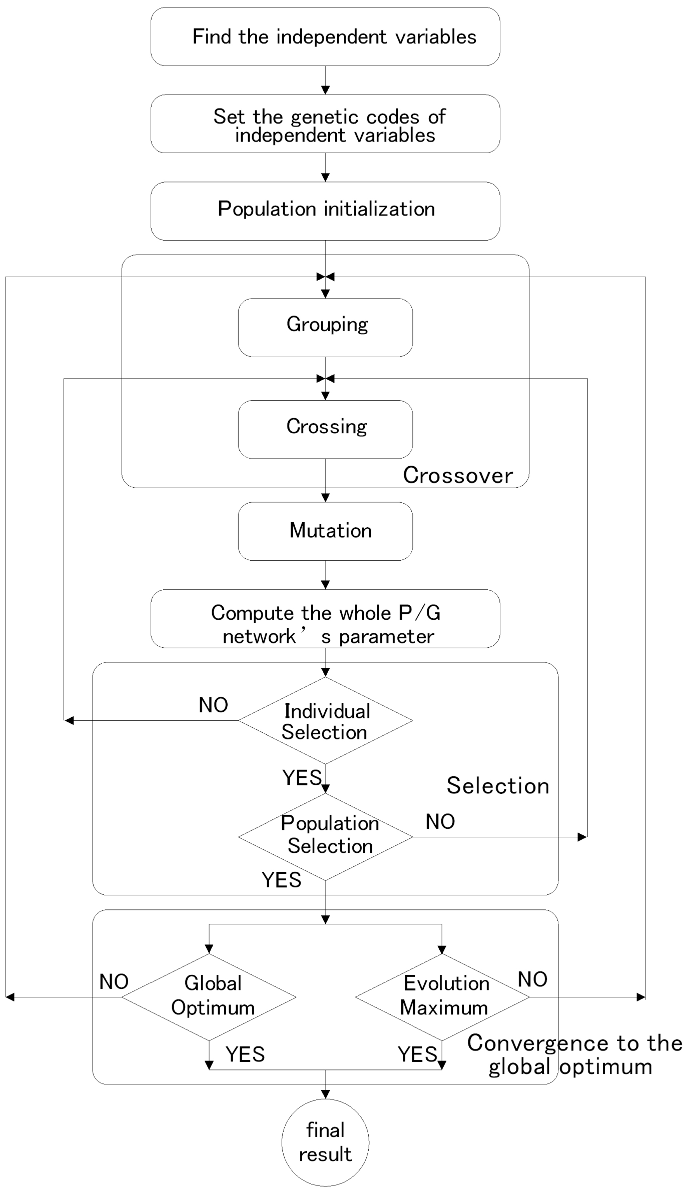

Find the independent variables
Set the genetic codes of independent variables
Population initialization
Grouping
Crossing
Crossover
Mutation
Compute the whole P/G network' s parameter
Individual Selection
NO
YES
Selection
Population Selection
NO
YES
Global Optimum
NO
YES
Evolution Maximum
NO
YES
Convergence to the global optimum
final result

Figure 3.7: The procedure flow of GGA method.

3.2.4 Large Scale Power/Ground Network Design

The previous section describes the small scale 4×4 P/G network grid synthesis and optimization by the proposed GGA method. If the network layout size is expanded, the computation complexity and the operation time are increased rapidly, which influence the basic GGA method efficiency. Thus further research and development should be focused on the P/G network scale expansion effects.

Compared with the basic P/G network case, the large scale P/G network grid synthesis emphasizes on the increased device number in each network wire, the pipeline computation network line enhancement, and large scale P/G mesh effect. In addition, the power sources and the PAD allocation can influence the P/G network grid current-voltage data relationship.

For $m \times n$ large scale network case, each distribution line drives several circuit devices, and many parallel network lines are listed in the whole chip layout. This P/G network grid impedes the element parameter independence. If one parameter varies, other segment currents or node voltages are changed to engender the new P/G network grid information data combination. This global relationship causes the encoding difficulty and reduce the evolution efficiency. Thus we should allocate the node information skillfully and avoid the operation error cumulation. The improved coding method and constraint selection design is the necessary research targets in the large scale P/G network grid synthesis design.

The mesh effect in the large scale P/G network grid is also important design challenge as shown in Figure 3.8. The P/G network grid includes the horizontal line and the vertical line. These crossover network composes the grid mesh as the fishnet hole style. Each mesh network carries several circuit devices, and the currents flow through the whole mesh network to distribute the large input current. The global current distribution increases the GGA method encoding difficulty, and impedes the global evolution efficiency. The multigrid method combines the neighboring network segments together to reduce the synthesis complexity [42]. For GGA method, the mesh grid coding allocation and the independent variable selection are core design efforts to improve the global optimization efficiency in the P/G network grid synthesis. Careful genetic algorithm design determines final network sizing area, operation speed, processing efficiency, or other evolution performance.

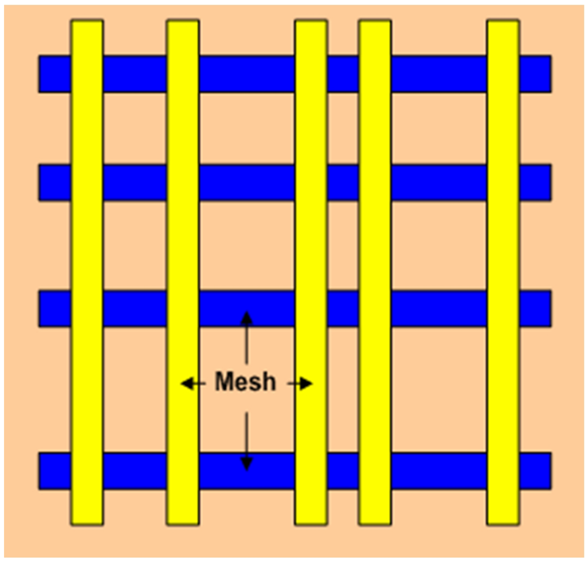


Figure 3.8: Power/Ground network grid mesh effect.

The common power source PAD positions are in the symmetric allocation in the P/G network grid, because of the global network balance. If the sources are in the middle part of the P/G network grid, the equivalent model and network equation computation can resolve the middle sources by the boundary equivalent power sources. Without the generality, we use four corner power sources to express the whole network source situation as shown in Figure 3.9.

To enable the large scale P/G network grid GGA coding, the partition process should be applied firstly to divide the network to typical GGA method network style given in Figure 3.6. For example, by the imaginal line in Figure 3.9, the original mesh network can be partitioned into four independent P/G network parts with left and right vertical P/G network line. By the genetic encoding and independent variable searching, the sub-network can be designed to get the global optimization sizing. In Figure 3.9, the independent variable number is $m \times (p-1) + q$, where m is horizontal P/G wire number, p is the vertical network line number. Each sub-network horizontal wire contains several n_i vertical device interface, the additional q means the vertical P/G network independent variables. The encoding method can also be modified to

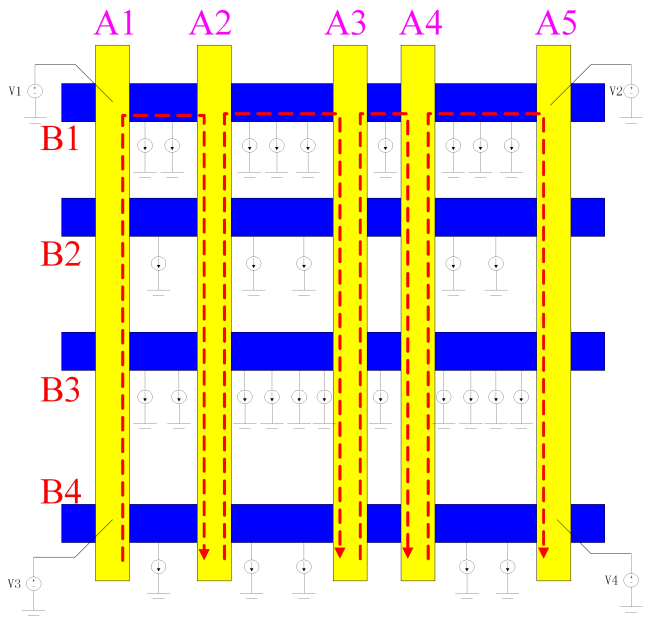


Figure 3.9: Large scale P/G network grid partition process.

distinguish the current coding and voltage coding style. Thus the genetic coding number can be reduced rapidly to decrease the genetic process complexity. Also, the evolution speed, optimization efficiency, operation stabilization, result floatability, or other genetic operation parameters, can be improved to enable the large scale GGA method application and global P/G network grid synthesis optimization.

The resolve procedure in the divided P/G network is also important for result computation and system colony evolution. The boundary connection among the sub-network is used to recover original P/G network information after genetic operation reaches global sub-network optimization. The equivalent boundary power sources can also be resolved to return original network positions for P/G network synthesis. The sub-network combination loss should be considered to avoid the result deviation and ensure the optimization correctness. As a result, the sub-network partition method, the equivalent model order reduction, the divided connection part combination, and the resolved element information recovery, provide the complement methods to enable GGA method process in large scale P/G network grid synthesis design situation.

3.2.5 Experimental Results and Discussions

Using the proposed GGA method, we can optimize the practical P/G network, and get the process time, the network area and the evolution efficiency. Because the linear transformation in Tan-Shi equivalent model SLP algorithm is not completely successful, we use the idea of searching the solution space step by step and get the accurate result to compare with GGA method. We test the result of the 4×4 P/G network grid in Figure 3.6. In this network the input voltage source is 3.0V and the genetic code is 12 bits "$VVVVVIIIIII$" for each segment. Then the largest segment current is 0.1A and the highest voltage drop for each segment is 0.05V. We also select the crossover rate 50% and the mutation rate 2% with the initial population size 30. After the computation we can get several P/G network areas because the GGA results float in each evolution as the diversity of species in the nature. The generation number varies between 100 and 1000 mostly and no more than 10000 in each evolution group. Consequently, based on these results we can select the minimal area and get the final P/G network optimization size shown in Table 3.1.

Table 3.1: The optimization results for 4×4 P/G network grid.

Methods		**Power/ground network area (um^2)**	**Process time(s)**
Tan-Shi equivalent model SLP (step search)		201.5382	0.015
GGA Method	Test1	109.0300 (near optimal size & final result)	16.688
	Test2	140.6160	15.109
	Test3	134.9440	16.203
	Test4	126.9540	18.375
	Test5	137.0550	16.640

In Table 3.1, it is obvious that GGA method are better than Tan-Shi equivalent model SLP. If we use Tan-Shi equivalent model SLP, the P/G network area is constant and the linear transformation is not successful. By the GGA method, we can get the minimal area $109.0300um^2$ and the improvement rate is 45%. Though the GGA behavior is affected by probability, we can get the minimal area by several experiments and the final result is more close to the global optimum than Tan-Shi equivalent model SLP, which can not realize the global optimization in some cases. But the process time for GGA method is longer than Tan-Shi equivalent model SLP. The average process time is 100∼1000 times than SLP algorithm. Thus in the small scale P/G network, the GGA method can realize the better global P/G network sizing optimization with the trade-off of process time cost.

If analyze the time cost of the proposed GGA method, we can find that it is mainly determined by the core evolution times. There are two evolution factors to affect the GGA method core evolution times. One is outside evolution factor e_{out} and the other one is inside evolution factor e_{in}. The former controls the group numbers and the latter decides the evolution times in each group. If these two factors vary, the process time will be changed. The detailed relationship is shown in Table 3.2. If the evolution factors are low the process time is short, but it is not easy to get the minimal area. On the contrary, if the evolution factors are high, the global optimization is easy, but the process time is much larger. The different combinations of the two factors can also influence the final results. Thus the proper selection of the evolution factors is the key point for the global P/G optimization. In this thesis we use the 10000 outside evolution factor and 10000 inside evolution factor. The result is better than Tan-Shi equivalent model SLP and more close to the global optimum. In addition the process time is not large and can be accepted for the P/G network grid synthesis design.

For Tan-Shi equivalent model SLP algorithm the complexity and process time are based on the scale and network architecture. If P/G network nodes increase, the complexity and time for SLP algorithm are determined by segment numbers and node

Table 3.2: The evolution results for 4×4 P/G network grid.

Outside evolution factor e_{out}	**Inside evolution factor** e_{in}	**Optimal power/ ground network area** (um^2)	**Optimal process time (s)**
100	100	152.671	0.2030
100	1000	109.030	1.6720
100	10000	109.030	18.594
1000	100	148.399	0.1400
1000	1000	135.730	1.8430
1000	10000	109.030	12.985
10000	100	133.513	0.1880
10000	1000	109.030	1.5780
10000	10000	109.030	14.703

quantities, which are linear with the square of P/G network size. Thus the process time of Tan-Shi equivalent model SLP algorithm increases rapidly. In addition, the snare effect and linear transformation can also restrict its efficiency and accuracy.

For the GGA method, though it is also necessary to calculate all branch currents and node voltages in P/G network grid, there are many advantages and merits to use this new global optimization method. Firstly, the SLP method computes each segment results step by step with ordinal exchange process of two phases $P-V$ and $P-I$. But in the GGA case each segments can realize their own evolution computation directly without the partition of computation phases. Secondly, the equivalent circuit modeling in Tan-Shi SLP algorithm is not completely successful. If the current directions are not same in a chain, the nodes can not be suppressed and the P/G segments can not be equivalent to simple circuit model. Then the SLP computation and back solved process become much more complex. For the GGA method, each chain has its own independent variable and the current directions can not influence the value establishment for each segment, once the genetic codes are decided. Thirdly, in the large scale network the GGA method has many evolution segments, then the whole system evolution speed will be accelerated rapidly because of the higher parallel global evolvement. There is an analogy between the P/G network grid optimization and the biology evolution history in the earth. The evolution time from the single cell animals to human being is almost ten times longer than the evolution period from the multicellular animals to human being, because the multicellular animals have much more evolution parts and can evolve together under the given natural selection. Thus all these advantages of GGA method can partly counteract the rapid increase of the computation complexity in large scale P/G network. Consequently, the proposed GGA method can reach the final global optimization results faster than SLP algorithm in the situation of the same P/G network grid scale.

The values of evolution factors are also important to determine the process time and P/G network layout area. If the evolution factors are limited improperly, the P/G network evolution can not easily reach the global optimum. And if the evolution factors are too large the process time will be extended intolerantly. Thus the suitable balance between the evolution factors and the P/G network grid optimization becomes the significant problem. In our experiments the common evolution factors are 10000 outside evolution factor e_{out} and 10000 inside evolution factor e_{in}, which is enough for the P/G network grid synthesis below the 100×100 scale. For the larger P/G network the detailed exploration of evolution factor effects should be carried on in the future work to understand the detailed factor relationship with the global P/G network grid optimal synthesis design.

Based on the above merits of direct computation, genetic variable independence and whole system evolution, the GGA method can optimize the large scale P/G network grid more efficiently than the SLP algorithm with high possibility. The

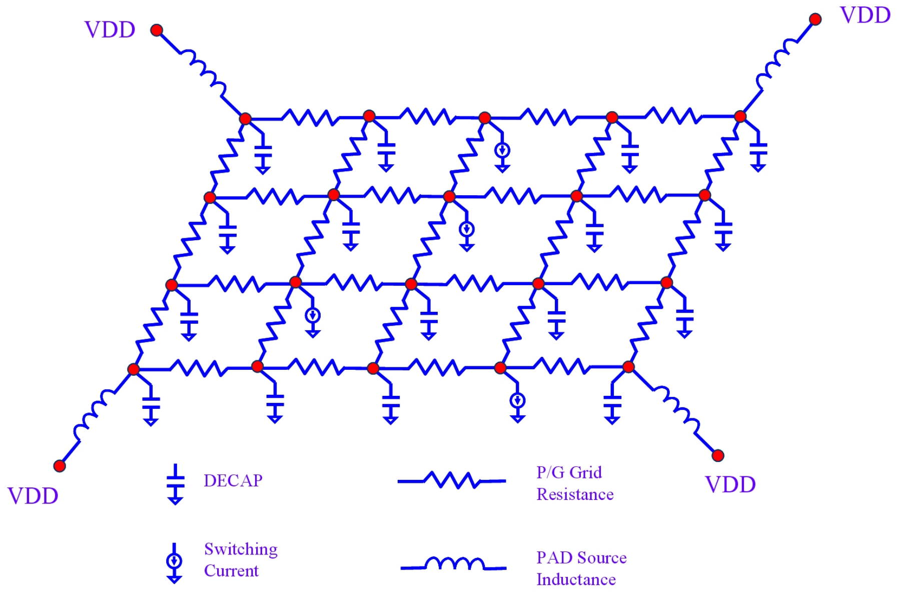


Figure 3.10: The VLSI P/G network model with dynamic signal consideration.

skillful selection of evolution factors can also ensure the P/G network sizing. For example, we experiment the 20×20 P/G network. If using the SLP algorithm, the process time increase more than 10 times because the network nodes increase from 16 to 400. But the process time for GGA method is still between the 15 seconds and 20 seconds with the evolution factors (10000, 10000). In addition, the P/G network layout area is also more close to the global optimization than the SLP algorithm.

3.3 Hybrid Grid Synthesis based on Genetic Algorithm (HGGA)

3.3.1 P/G Network Synthesis with Dynamic Signal

In Figure 3.10 the P/G network synthesis with dynamic signal consideration is required to get the global layout area, which includes wire sizing area and decoupling capacitance (DECAP) area [8, 29]. The corresponding area cost objective function is defined as follows [29, 42]:

$$F = \sum_{i=1}^{P} l_i w_i + \alpha \sum_{j=1}^{N} Cd_j. \tag{3.20}$$

where α is weight factor of two layout area cost of sizing area and DECAP area. P is network wire segment number, and N is the number of DECAP insertion nodes. l_i and w_i give length and width of segment i. Cd_j is DECAP value of node j. Because of layout area optimization target by given fixed P/G network topology, the distance l_i of each node is determined and the optimization objective is focused on segment width sizing and DECAP budgeting minimization.

For P/G network design with dynamic signal, the area cost objective function can be converted further by segment value combination of branch current $I_i(t)$ and node voltage $V_i(t)$. The wire sizing part can be replaced by resistance r_i relation function [25] of $r_i = (V_{i1}(t) - V_{i2}(t))/I_i(t) = \rho_s(l_i)/(w_i)$. The DECAP budgeting part can also be represented by Backward Euler (BE) formula for capacitor mode as [17]:

$$V_j'(t) = V_j(t) + \frac{h}{C} I_{Cj}'(t). \tag{3.21}$$

where h means timestep and C is DECAP capacitor value. $V_j'(t)$ is present time step node voltage and $V_j(t)$ is last time step voltage value. Also, DECAP flow current value in present time step is defined by $I_{Cj}'(t)$ as Eq. (3.21).

Based on resistance equivalence and Backward Euler transformation, P/G network synthesis objective function can be expressed by Eq. (3.22) as follows:

$$f(V, I) = \sum_{i=1}^{P} \frac{\rho_s l_i^2 I_i(t)}{V_{i1}(t) - V_{i2}(t)} + \alpha \sum_{j=1}^{N} \frac{h I_{Cj}'(t)}{V_j'(t) - V_j(t)}. \tag{3.22}$$

The objective function of area cost is converted to current/voltage combination relation. The global geometric minimization can be equivalent to circuit parameter optimal design. Also, the segment circuit parameters, such as branch current $I_i(t)$ and node voltage $V_i(t)$, vary under different time step and requires further design consideration.

The P/G network synthesis is realized under many circuit constraints, such as wire sizing constraints, DECAP constraints, and circuit system constraints [25, 42].

The corresponding constraints are also improved under the characteristic adjustment for dynamic signal circuit as follows:

- *Wire sizing constraints*: The suitable P/G network distribution has many constraints, including IR drop constraints, minimum width constraints, and current density constraints [25]. For example, voltage floating from PADs to each crossing node should be limited under specific range. The network wire widths can not be too small because of production obstacle and design problem. Also, each wire segment has maximal current constraint, otherwise too large current would impair segment performance and circuit function.

- *Decoupling capacitance (DECAP) constraints*: Each DECAP size has lower bound $Cd_{j(min)}$ defined by estimated intrinsic DECAP value of node voltage. The DECAP size upper bound $Cd_{j(max)}$ also exists by capacitance space allocation [17,25,42]. The specific DECAP constraints and dynamic circuit equations by Backward Euler (BE) model are described as follows:

$$Cd_{j(min)} \leq \frac{hI'_{Cj}(t)}{V'_j(t) - V_j(t)} \leq Cd_{j(max)}. \tag{3.23}$$

- *Circuit system constraints*: The Modified Nodal Analysis (MNA) formulation can give P/G network relation of node voltages and flow currents. The Ordinary Differential Equation (ODE) in Eq. (3.24), which includes circuit KVL and KCL relation together, describes special circuit system constraints in detail [17,42].

$$G\vec{V}(t) + C\dot{\vec{V}}(t) = \vec{I}(t). \tag{3.24}$$

  where G means conductance matrix and C is capacitor matrix. The Node voltage vector $\vec{V}(t)$ and the flow current vector $\vec{I}(t)$ give the relevant dynamic circuit relationship.

Consequently, the P/G network synthesis design can be transformed to find optimal wire width and allocate minimal decoupling capacitance to achieve global optimization of objective function $f(V, I)$ in Eq. (3.22) and maintain the dynamic circuit function under P/G network constraints.

3.3.2 Dynamic Signal Manipulation in Genetic Algorithm

Basic GGA method can improve the P/G network synthesis efficiency and realize the layout area optimization. The circuit parameters, such as node voltages and branch currents, can be replaced by the genetic coding. And the population evolution can simulate the P/G network design and achieve the global sizing optimization [2]. However, for dynamic signal consideration, the complex parameter interaction, such as Ldi/dt noise and Cdv/dt DECAP current, will cause the functional problems and increase the synthesis difficulty. The basic genetic method can not deal with Ldi/dt and Cdv/dt easily. Here we propose a novel gene disturbance method to realize the simulation of inductance and DECAP based on the biology evolution theory.

The algorithm flow for the "Dynamic Signal Manipulation in Genetic Algorithm (DGGA)" is shown in Figure 3.11, where the genetic coding is allocated for node voltages and branch currents of each wire segment at first. Given the initial population group, the individuals with genetic coding can exchange the coding segment to get the crossing offsprings. And the specific coding bits can vary with the suitable probability to realize the genetic mutation process. In addition, the novel gene disturbance process is inserted into the basic genetic method to simulate the dynamic signal of Ldi/dt and Cdv/dt. After the crossing, mutation and disturbance, new generation can be created to express new circuit parameters. Consider the circuit constraints as the selection environment, new generation can be selected to get the suitable offsprings as good circuit parameters. These selected generation can be used as the initial group for next evolution. By continued evolution, the generation can be improved to satisfy the environment constraints, or the circuit parameters can be

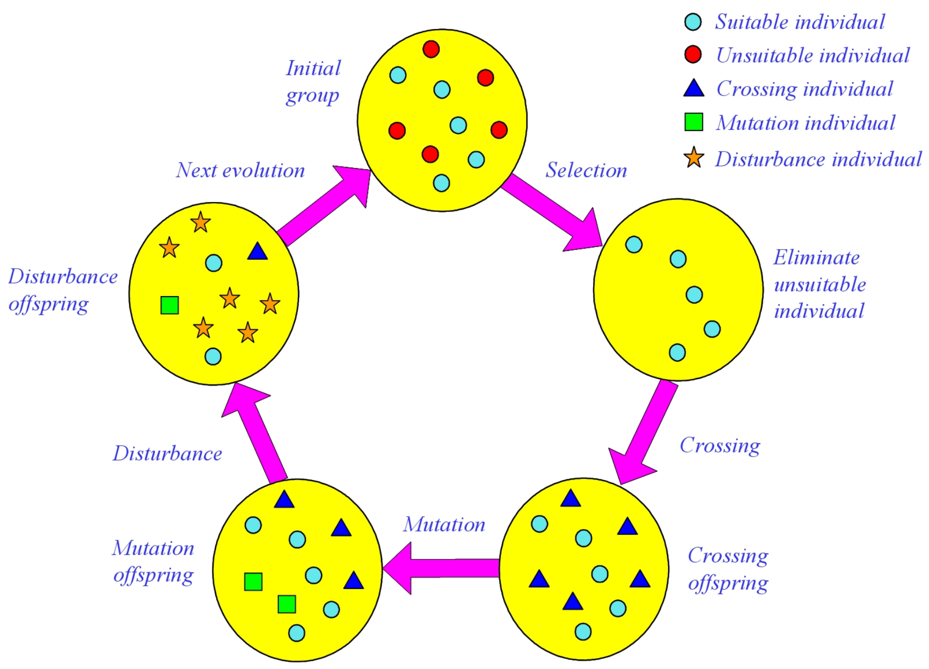


Figure 3.11: The algorithm flow of "Dynamic Signal Manipulation in Genetic Algorithm (DGGA)" with additional gene disturbance operation.

minimized to reduce the P/G network layout area. When the evolution enters the required endpoint, the genetic process has finished and P/G network can realize the global area optimization, because of the genetic theory for survival of the fittest.

The core improvement for the dynamic signal process is the introduction of gene disturbance method. In biology science, the gene disturbance means the individual variation under survival environment pressures. Different from basic gene mutation, which changes the genetic codes by random directions, the gene disturbance only happens under the environmental induction and varies with the expected reason. Also, the gene disturbance will not change the original individual gene largely, which is just boundary supplement or partial dissimilation for basic biological species. For example, human being has been evolved from pithecanthrope in several million years ago. Since the hominid walks from east African under recent anthropology research, different migration direction and survival environment have caused the human gene disturbance and different homo sapiens generation. For three main modern human races, the Caucasian race is from Europe, the Mongoloid race moves distantly to Asia, and the Negroid race stays in African. As a result, gene disturbance effect causes small human race differences, such as skin color, hair style, face shape, sport ability, etc.

Compared with biology gene disturbance, the dynamic signal of Ldi/dt and Cdv/dt in the P/G network can be considered as small value variance of circuit parameters. The P/G network synthesis can be realized based on "Dynamic Signal Manipulation in Genetic Algorithm (DGGA)" with additional gene disturbance operation. As shown in Figure 3.11, the initial group comes from genetic parameter coding, such as node voltage, branch current, and inductance current. After the selection process, the unsuitable individual is eliminated. The suitable individual enters the crossing process to get next generation offspring. Also, the mutation process enables the genetic coding complexity and avoids the good result missing. The next disturbance process is manipulating the P/G network dynamic signal by value variances of genetic codes at several end bits in the coding segment. The modified bit segment can express the variance range of di/dt and dv/dt. Thus the additional coding adjustments can work together with the genetic process of crossing and mutation to simulate the P/G network synthesis with the dynamic signal consideration.

3.3.3 Trapezoidal Modified Euler Method

For P/G network synthesis with dynamic signal consideration, node voltages and branch currents vary with different time point, unlike IR drop P/G network case. So different timestep coding should satisfy the constraint requirements and express the corresponding circuit parameters.

Common solution method is using the Backward Euler (BE) method to divide the circuit waveform and approximate the results [17]. To increase the simulation accuracy and satisfy the dynamic signal process requirements, the Trapezoidal Modified Euler (TME) method is used in this paper. The Modified Euler method is also named as Predictor-Corrector Euler method [99]. This implicit Euler method predicts next time point function value by basic Forward Euler (FE) method as Eq. (3.25).

$$y_{i+1}^p = y_i + hf(x_i, y_i). \tag{3.25}$$

where h means the timestep and y_{i+1}^p gives the predicted function result in next time point. Also, $f(x_i, y_i)$ describes the specific objective function relation [99].

The predictor value can be used in next real Euler method operation to correct the estimated Euler value and improve the simulation efficiency as shown in Eq. (3.26).

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, y_{i+1}^p)]. \tag{3.26}$$

where the trapezoidal method is also used to increase the approximation accuracy further [99].

From Eq. (3.25) and Eq. (3.26), the final Trapezoidal Modified Euler (TME) method relation can be given in Eq. (3.27).

$$y_{i+1} = y_i + \frac{h}{2}[f(x_i, y_i) + f(x_{i+1}, y_i + hf(x_i, y_i))]. \tag{3.27}$$

Thus the difficult implicit Euler iterative solution processing can be avoided with acceptable simulation accuracy [99].

Based on TME Euler method in Eq. (3.27) and circuit MNA formulation in Eq. (3.24), the detailed values of voltage and current can be derived to express the dynamic signal influences. Because original large size network can be compressed to core P/G grid mode [42], reduced network in Figure 3.12 is used to describe specific circuit performance.

Around reduced network corner node, the inductance current from outside PAD can be predicted as follows:

$$I_{Lj}^p(t) = I_{Lj}(t) + \frac{h}{L}(VDD - V_j(t)). \tag{3.28}$$

The MNA circuit system constraint of ODE problem given in Eq. (3.24) can deduce the predicted value of DECAP flowing current $I_{Cj}^p(t)$ as Eq. (3.29), where

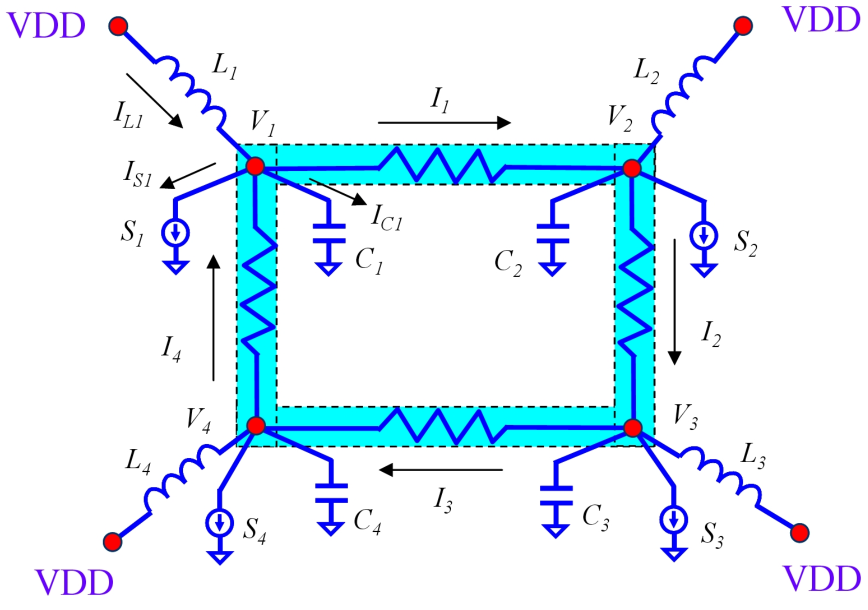


Figure 3.12: The core P/G grid model for Trapezoidal Modified Euler (TME) method.

$I_{j(left)}(t)$ and $I_{j(right)}(t)$ mean current values of left branch and right branch around given DECAP node, such as left current I_4 and right current I_1 of left upper corner node with voltage V_1 in Figure 3.12. $I_{Sj}(t)$ is the switching current of practical circuit device connected to P/G distribution network.

$$I^p_{Cj}(t) = (I^p_{Lj}(t) + I_{j(left)}(t)) - (I_{j(right)}(t) + I_{Sj}(t)). \tag{3.29}$$

The Backward Euler (BE) method can also be used to describe DECAP current and voltage relation. The predictor node voltage of inserting DECAP can be formulated as:

$$V^p_j(t) = V_j(t) + \frac{h}{C} I^p_{Cj}(t). \tag{3.30}$$

By simultaneous equations of Eq. (3.27) and Eq. (3.28), practical next timestep inductance current $I'_{Lj}(t)$ can be derived as Eq. (3.31), where L is inductance value connected to PAD, and VDD is source power of circuit system. The definitions of present timestep, next timestep and predicted parameter values are similar to above setting as Eq. (3.21) and Eq. (3.25).

$$I'_{Lj}(t) = I_{Lj}(t) + \frac{h}{2L}[(VDD - V_j(t)) + (VDD - V_j^p(t))]. \qquad (3.31)$$

In whole P/G network, node voltages and branch currents can be resolved by practical parameter values and equivalent circuit expansion based on TME Euler Method and MNA formulation. Thus different timestep values in P/G network with dynamic signal consideration can be computed easily to achieve the global layout area optimization.

3.3.4 Hybrid-SLP Method with Genetic Algorithm

Efficient P/G network synthesis requires high process accuracy and large scale network operation. The "Dynamic Signal Manipulation in Genetic Algorithm (DGGA)" can realize dynamic signal simulation and global layout area optimization without nonlinear-linear transformation loss. However, basic genetic method spends much longer time to deal with individual coding, environmental selection and crossing/mutation operation, etc. While SLP method can convert the nonlinear relation of objective function and constraints to linear function with fast process speed, but partly transformation loss [25]. Consider both method merits, the "Hybrid-SLP Method with Genetic Algorithm", including global area optimization and fast process speed, is proposed to deal with dynamic signal process in P/G network design.

The specific hybrid idea is described in Figure 3.13 in detail. The mesh plane means the branch current and node voltage combination with X direction for current and Y direction for voltage. The local minimal region $R1$ and global minimal region $R2$ are local and global layout area optimization positions, respectively. The constraint regions $(C1, C2, C3)$ give the P/G network synthesis constraints with current and voltage divergency process. From the start point, the optimization route travels by the genetic method and avoid the circuit constraints. When the route falls into the snare of local minimization area, the genetic method enables it to jump out of

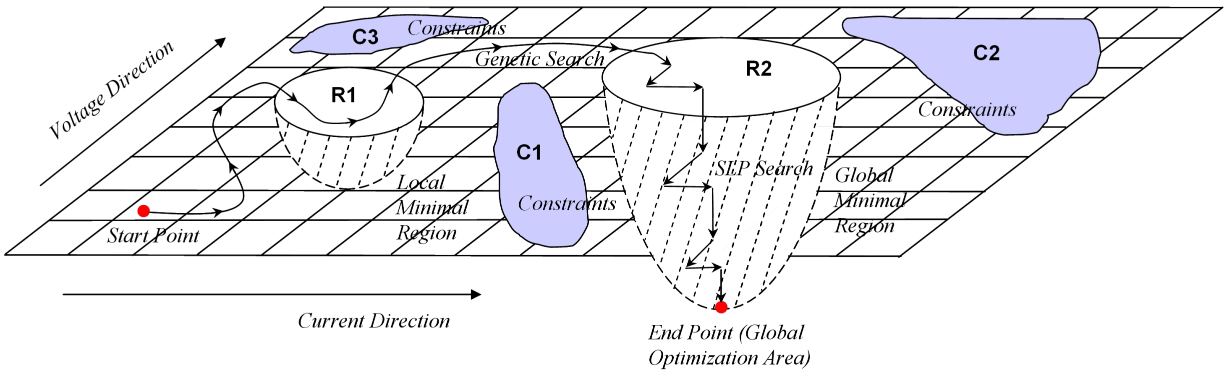


Figure 3.13: The hybrid-SLP method with genetic algorithm for Power/Ground network grid synthesis.

the local snare and progresses toward the global minimal region again. Based on the genetic characteristics of global optimization, the route can reach global minimization area with great possibility. Also, because of genetic coding variance reason, the end part searching of the optimization route is not fast, even repeats the jumping in/out process to increase the operation time and decrease the evolution efficiency. The fast end part searching method of SLP algorithm is used to replace the genetic method when the search route reaches the border of global minimization region. By the divergency process of current, voltage and dv/di dynamic signal parameters, the SLP method can realize fast searching speed and fall into global minimization region quickly. Because searching route already reaches global minimization region, the SLP searching route is limited under snare region in avoidance of transformation loss and searching obstacle of traditional SLP searching problem. Thus the searching route can fall into the end point of global optimization area much faster than the floating search process in basic genetic method.

Based on genetic global optimization and fast SLP endpoint searching, the hybrid-SLP method can combine both merits to realize efficient P/G network design, as recent examples of hybrid SLP/SQP/CG combinations with basic genetic method in real-time process system [100]. The final SLP searching result can approach the genetic minimal point with less evolution times and fast process speed. And the specific switch positions to SLP method are determined by the estimation of genetic evolution times and the observation of endpoint value floating. The procedure flow of “Hybrid Grid Synthesis based on Genetic Algorithm (HGGA)” is given in Figure 3.14.

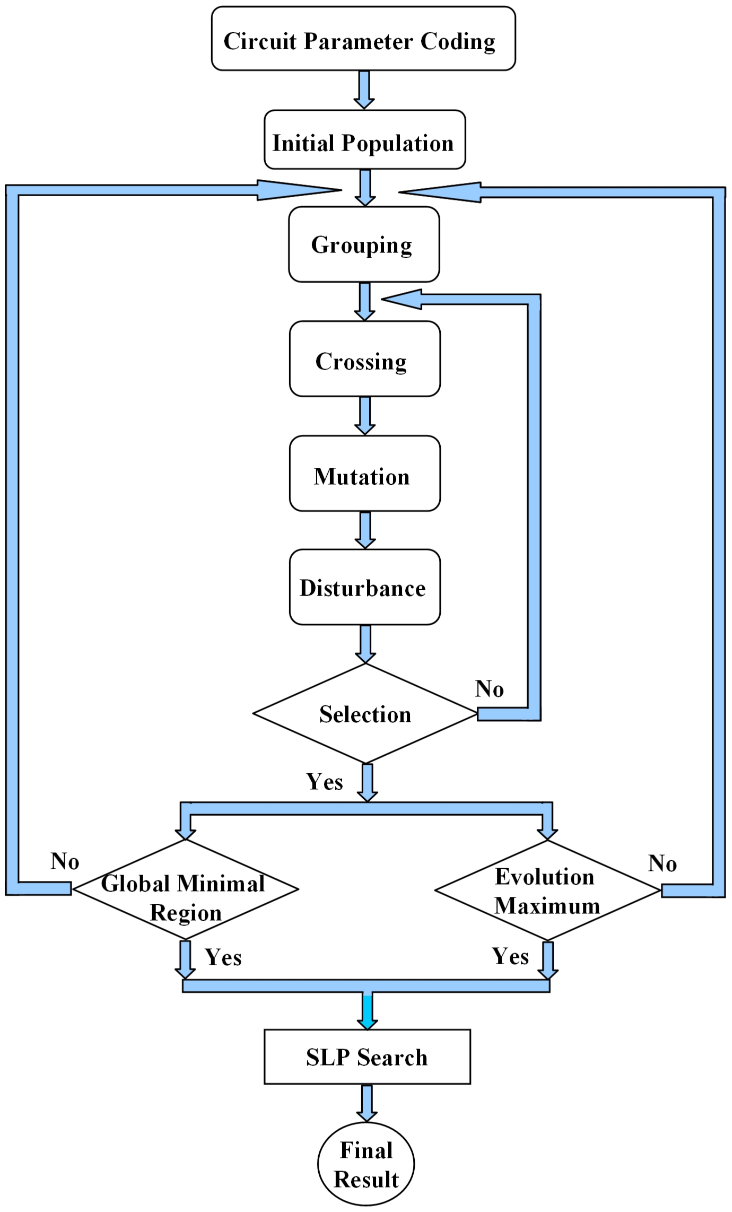


Figure 3.14: The procedure flow of “Hybrid Grid Synthesis based on Genetic Algorithm (HGGA)”.

3.3.5 Experimental Results and Discussions

Under a Linux PC with 2.40-GHz Pentium CPU and 1.00-GB memory, we implement the "Hybrid Grid Synthesis based on Genetic Algorithm (HGGA)" by dynamic signal manipulation (DGGA), trapezoidal modified Euler method (TME) and hybrid-SLP method in C++ program. The circuit parameters, such as sheet resistivity (ρ_s), maximal current density (J_{max}), minimum wire width (W_{min}) / pitch (p_{min}) are obtained from ITRS 2001 [101]. The P/G network uses 130-nm technology and 1.2V VDD supply. The voltage floating constraint is 90% VDD, or 1.08V voltage drop limits.

For genetic algorithm, we use 12 bit codes to express node voltage, branch current and DECAP current, respectively. The highest node voltage is 1.2V, and the largest branch current is 0.1A, together with maximal DECAP current 0.005A in genetic coding allocation. The crossover rate is 50% and the mutation rate is 2%. We also select 75% disturbance rate to deal with the dynamic signal. The initial population size is 30. Because we use SLP hybrid method to raise endpoint searching speed, generation number can be decreased rapidly. Also, the weight factor α of sizing area and DECAP area selects the value '1' to compose whole DGGA area.

We also focus on the 2×2 corn network equivalent model (Figure 3.12) from 4×5 P/G network mesh reduction as shown in Figure 3.10. Also, TME duration time is 0.2ns, and original DECAP values select $1pF \sim 200pF$ capacitance with several

Table 3.3: The HGGA evolution results of 4×5 P/G network model.

	P/G network area					**HGGA process time**			
	DGGA (μm^2)			Hybrid SLP (μm^2)		Whole	DGGA	SLP	SLP
	DGGA area	Sizing area	DECAP area	Final area	SLP improve	time (s)	time (s)	time (s)	time ratio
Test 1	7189.487	2115.618	5073.868	6643.176	7.5987%	8.778023	8.776444	0.001579	0.01799%
Test 2	7179.321	1509.225	5670.096	6568.945	8.5019%	6.188068	6.186131	0.001937	0.03130%
Test 3	5416.747	1510.229	3906.518	5392.726	0.4435%	10.606681	10.606458	0.000223	0.00210%
Test 4	7565.910	1832.957	5732.952	7011.543	7.3272%	3.273172	3.271272	0.001900	0.05805%
Test 5	6721.434	2516.722	4204.711	5850.134	12.963%	7.803567	7.796284	0.007283	0.09333%

Table 3.4: Comparisons of P/G network layout area and process time with different evolution factors for HGGA method.

Algorithms **(e1, e2, e3)**	**Minimal** **area (μm^2)**	**Maximal** **area (μm^2)**	**Average** **area (μm^2)**	**Average** **area ratio**	**Minimal** **time (s)**	**Maximal** **time (s)**	**Average** **time (s)**	**Average** **time ratio**
HGGA $(100, 30, 100)$	5392.726	7204.860	6625.825	1.0000	3.273172	10.606681	6.774586	1.0000
DGGA $(100, 30, 100)$	5735.697	7619.649	6827.714	1.0305	3.949530	9.736464	6.487729	0.9577
DGGA $(300, 100, 300)$	4670.848	7046.013	6213.138	0.9377	33.70503	90.59387	67.21230	9.9212
DGGA $(1000, 300, 1000)$	2943.870	6067.987	5009.532	0.7561	366.7305	1004.7524	784.5841	115.813

hundred pH PAD inductance. dv/dt and di/dt can be solved by gene disturbance and TME model. Larger P/G network can be expanded by multigrid-based technique [42], but run time will not increase linearly with network scale because of genetic merits of global evolution and system pipeline.

Because the genetic algorithm is not deterministic method with the results variation of layout area and process time around final data values, five typical test bench examples are given to describe the HGGA algorithm efficiency in Table 3.3. Test 1 means the evolution results in common layout area case, which happens with large probability and closes to average network area. The basic DGGA area is 7189.487μm^2 by two part combination of 2115.618μm^2 sizing area and 5073.868μm^2 DECAP area. After the hybrid SLP process, the final area is 6643.176μm^2 and the SLP area reduction is 7.5987%. For HGGA process time, whole time is 8.778023s with 8.776444s DGGA time and 0.001579s SLP time. Thus the SLP method can spend just 0.01799% time cost to realize 7.5987% area reduction. Also, Test 2 case is common process time case. Test 3 and Test 4 give the HGGA evolution situation of minimal layout area and minimal process time. The maximal SLP improvement of 12.963% area reduction is shown in Test 5.

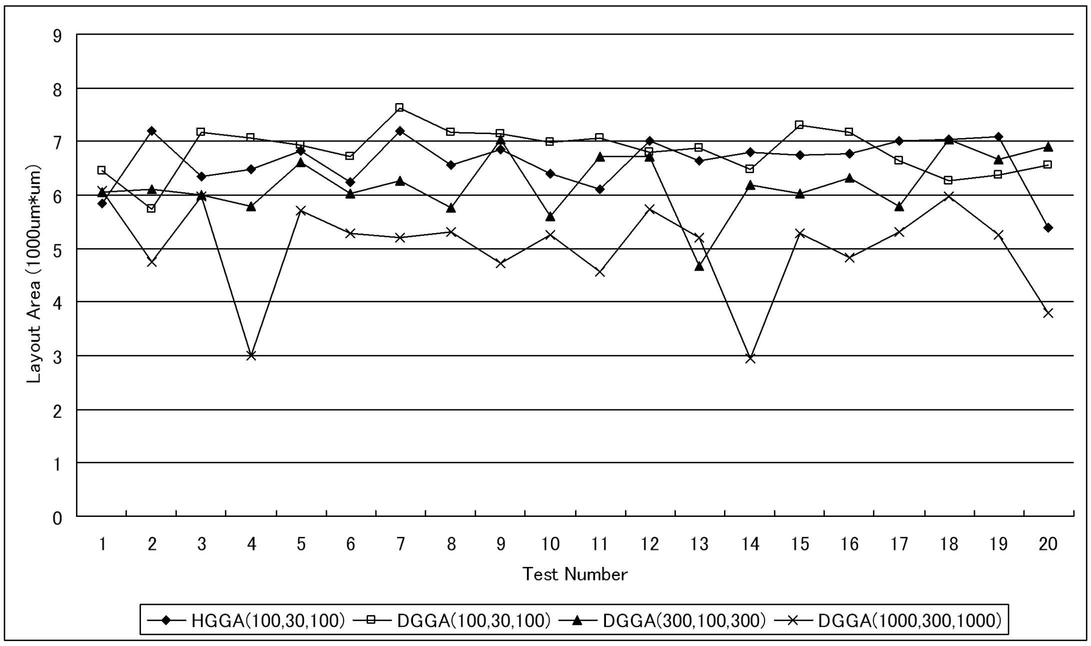


Figure 3.15: The HGGA layout area test comparisons.

The comparisons of P/G network layout area and process time with different evolution factors are given in Table 3.4. Factor $e1$ is the population grouping number for evolution exterior process, factor $e2$ means the break time value when the evolution can not reach the results. Also, factor $e3$ gives the interior evolution times in each population grouping process. Different evolution factors can cause the value variation for layout area and process time. By performance illustrations in Figure 3.15, Figure 3.16 and Table 3.4, The HGGA method can get $6625.825\mu m^2$ average area and 6.774586s average time by the evolution factors of 100($e1$), 30($e2$) and 100($e3$). The DGGA method with same evolution factors can get the synthesis results with 1.0305 average area ratio and 0.9577 average time ratio, but the value fluctuation range is large and can not reach the stable results easily. If the evolution factors have been extended, the layout area can be reduced, but the process time will increase rapidly, even 115.813 ratio of average time cost to just get 0.7561 ratio of average area improvement for DGGA(1000,300,1000). Based on the trade-off between layout area, process time and computation stability, the proposed HGGA algorithm is

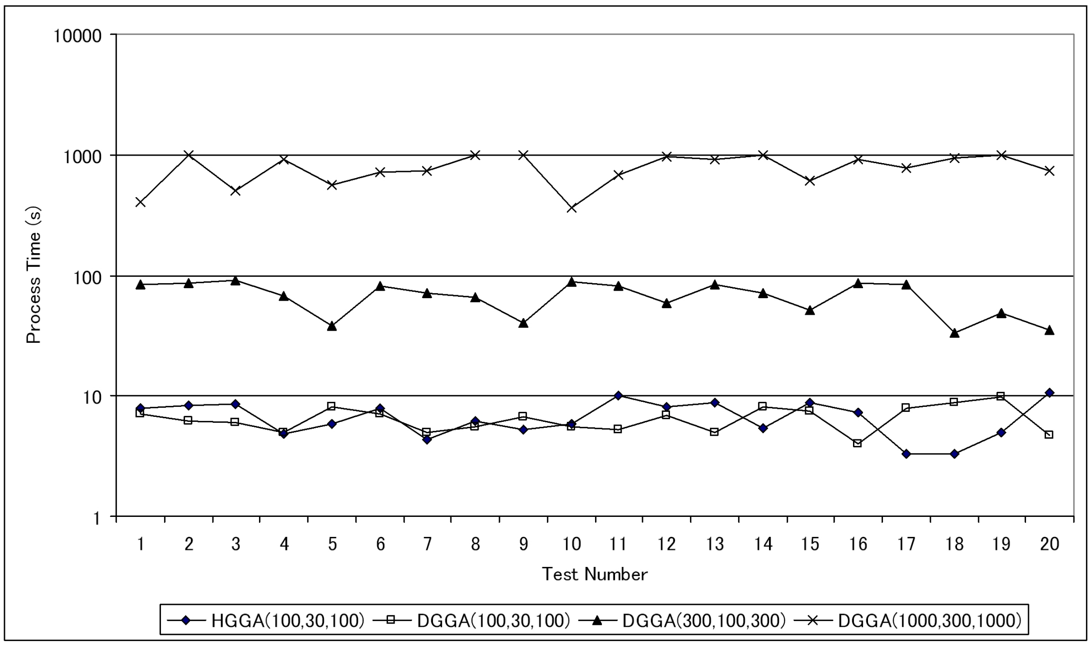


Figure 3.16: The HGGA process time test comparisons.

better Power/Ground network synthesis method with local minimization avoidance merit and fast endpoint searching capability.

Other network synthesis methods, such as SLP/SQP/ICG algorithms, can design the P/G network with high optimization speed, but dynamic process demerit and local minimization snare will influence the P/G network synthesis efficiency. Also, recent statistic random-walk method wastes many unnecessary walk process and can not ensure each walk route accuracy, even if this method can deal with some dynamic signal cases with high efficiency. Compared with these traditional synthesis methods, HGGA method uses genetic algorithm and TME method to solve the dynamic signal simulation, and the hybrid-SLP process combines both performance merits of SLP method and genetic algorithm. The HGGA method can also solve the optimization target directly without detailed considerations for objective functions and constraint conditions. Thus the proposed HGGA method is suitable for VLSI optimization problems and can be extended to other dynamic signal process domain.

3.4 Power/Ground Network Grid Synthesis Conclusion

The sequence-of-linear-programming (SLP) algorithm is the common method to solve the global P/G network grid optimization. But it can not reach the really global optimum in some cases and faces many problems, such as the snare effect and inadequate linear transformation demerits. To get rid of the SLP method deficiency, we introduce the evolution idea from the biology world and propose the GGA method in this thesis. This method enables the P/G network sizing area more close to the global optimum and its optimization process is more efficient with the growing scale of P/G network grid. Experimental results show that the proposed method is better than the conventional SLP method especially in the global optimization.

In addition, this thesis also proposes an efficient design algorithm for power/ground (P/G) network synthesis with dynamic signal consideration, which is mainly caused by Ldi/dt noise and Cdv/dt decoupling capacitance (DECAP) current in the distribution network. Instead of traditional SLP/SQP/ICG algorithm and statistic random-walk method, the gene disturbance process is proposed to simulate the dynamic signal, and the Trapezoidal Modified Euler (TME) method is introduced to realize the precise dynamic timestep process. The hybrid-SLP method is also used to reduce the genetic execute time and increase the network synthesis efficiency. Experimental results of power distribution network have proved the performance improvements for the HGGA method in layout area optimization and synthesis operation speed.

Future work will be focused on decreasing the process time further in the optimization and find the detailed relationship between the evolution factors and the process complexity. More large and complex P/G network grid should be tested to demonstrate the benefits of the proposed GGA/HGGA method. In addition, the GGA and HGGA methods can also be used in the P/G grid synthesis of 3-D network architecture and nanometer interconnect research in recent low power SoC design.

Chapter 4

SYSTOLIC ARRAY NETWORK GRID SYNTHESIS

4.1 Systolic Array Grid Problem Formulation

The basic element for matrix multiplication is processing element (PE), which calculates the multiplication operation and transfers data to realize the matrix multiplication function. Figure 4.1 gives the PE connection types including square topology and hexagonal topology, where a, b, and c mean multiplicator, multiplicand and multiplication-add operation result, respectively. The PE function is multiplying the input data a, b, and adding the operation data c to get new multiplication-add result c'. Also, a and b are sent to output port without change to get a' and b' as shown in Figure 4.1 [4].

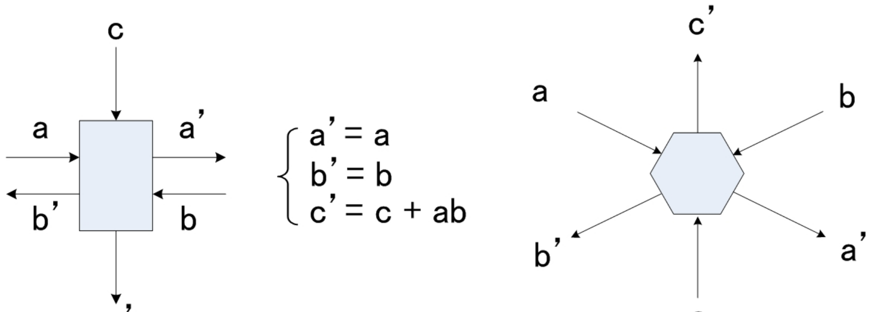


Figure 4.1: Processing element (PE) operation for matrix multiplication. (a) square topology, (b) hexagonal topology.

Consider the common matrix multiplication with input $n \times n$ matrices $A = (a_{ik})$ and $B = (b_{kj})$ to result in the output data matrix $C = (c_{ij})_{n\times n}$. The matrix operation equation can be described as Eq. (4.1) [4, 5].

$$[A]\bullet[B] = [C], \tag{4.1}$$

Or the matrices can be expanded as Eq. (4.2).

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \bullet \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix}. \tag{4.2}$$

Then the matrix multiplication result $C = (c_{ij})_{n\times n}$ can be computed by data combination and concurrent operation of the corresponding matrix elements. The specific characteristics of data arrangement and matrix symmetry can also determine result correctness and processing speed for the given multiplication operation.

According to systolic array architecture, system concurrent operation and highly pipeline characteristics enable it to be the suitable processing configuration for matrix multiplication. The systolic array matrix multiplication in Eq. (4.2) can also be expressed by the multiplication operation as Eq. (4.3) [4, 94].

$$c_{ij} = \sum_{k=1}^{n} a_{ik}b_{kj}, \qquad i, j = 1, 2, ..., n. \tag{4.3}$$

The detailed multiplication accumulation relationship is shown in Eq. (4.4).

$$c_{ij}^{(k)} = c_{ij}^{(k-1)} + a_{ik}b_{kj}, \quad \text{where} \quad k = 1, 2, ..., n, \quad \text{and} \quad c_{ij}^{(0)} = 0. \tag{4.4}$$

Then the systolic iterative algorithm of matrix multiplication computation can be given as the following pseudo code procedure [91, 94].

Algorithms 4.1 : (***Matrix multiplication systolic array***)

```
for $k = 1$ to $n$
    for $j = 1$ to $n$
for $i = 1$ to $n$
    begin
        $a(i,j,k) = a(i,j-1,k)$
        $b(i,j,k) = b(i-1,j,k)$
        $c(i,j,k) = c(i,j,k-1) + a(i,j,k) \times b(i,j,k)$
    end
```

where

$$a(i,0,k) = a_{ik}, \;\; b(0,j,k) = b_{kj}, \;\; c(i,j,0) = c_{ij}^{0} = 0.$$

For the practical systolic arrays, there are many array connection topologies to realize the highly concurrent operation and global pipeline computation [4, 75], such as linear connection, cubic array, triangular connection, square topology, and hexagonal topology as shown in Figure 4.2. As a result, the different array topologies are corresponded to the relative VLSI processing systems or matrix multiplication arrays with the diverse system pipeline characteristics.

In the systolic array design, the "throughput latency" means the duration period from the begin time point when first input data are sent into the array, to the end time point when the last results are sent out of the array. The "average utilization rate" is the average active PE per-cycle to total PEs, or the ratio of average active PE number to total array PE number in each cycle [4,6]. Based on the elementary systolic architecture, many developments and advancements are exerted to reduce throughput latency, increase average utilization rate, enhance bandwidth ability, fortify data transmission robustness, or other system characteristic improvements. These new systolic arrays enable highly data pipeline and fast operation speed for VLSI processor and matrix multiplication. Consequently, we focus on 2-D systolic array improvement and try to increase the matrix multiplication operation efficiency to allow high performance pipelined operation.

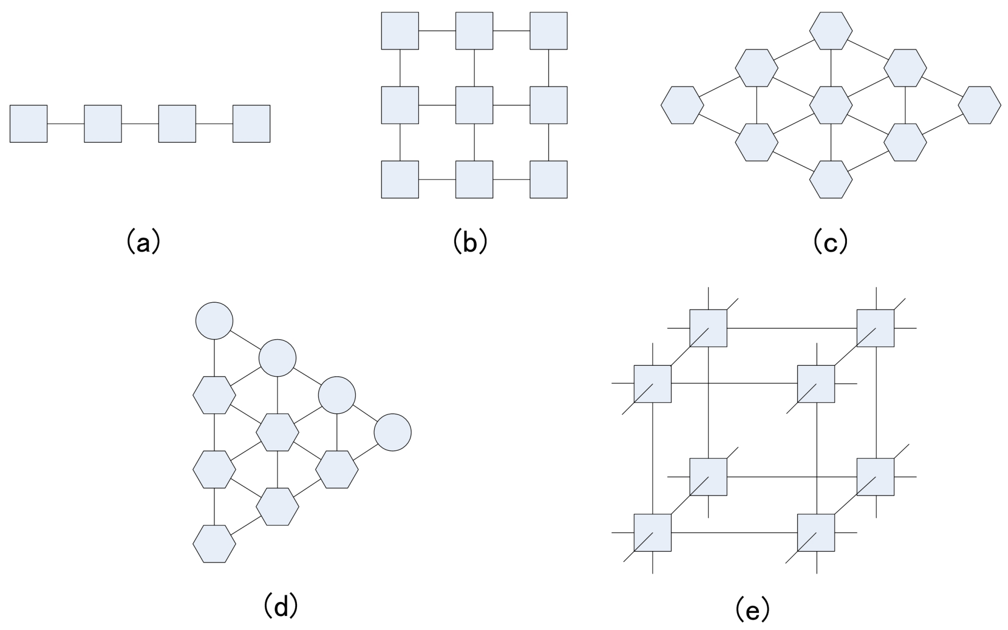


Figure 4.2: The different systolic array connection topologies. (a) linear connection, (b) square topology, (c) hexagonal topology, (d) triangular connection, (e) cubic array.

4.2 Super Pipeline Systolic Array Grid Algorithm

4.2.1 Band Matrix Multiplication Systolic Array (BMMSA)

In the large-scale scientific computation, the sparse data matrices are widely used to construct the highly pipeline processing array. Many sparse matrices can be converted to band matrices. Thus the research of band matrix multiplication systolic array (BMMSA) is important for recent DSP processing system implementation. If matrix A and matrix B are the band matrices with bandwidth W_A and W_B, the product matrix C is also a band matrix. Then its bandwidth W_C can be expressed as [4]:

$$W_C = W_A + W_B - 1. \tag{4.5}$$

By spreading the given Eq. (4.5) we can get the detailed relationship of the BMMSA input and output values. Suppose matrix A and matrix B have the same bandwidth 4, then the bandwidth of output matrix C is "$4 + 4 - 1 = 7$". If the bandwidth is other value, we can change the band size of the hexagonal topology array and realize the matrix multiplier operation by the same method.

When the BMMSA design in Figure 2.2 is in operation, only 5 or 6 processing elements (PEs) are in work. The average utilization rate is just about 1/3, and the throughput latency of the BMMSA with n-rank input data matrix is "$3n + min(W_A, W_B)$". In the equation W_A and W_B are two operand matrices bandwidths, and $min(W_A, W_B)$ means the cost before the BMMSA is stable. If $n >> W_A$ or $n >> W_B$, the time steps are approximately $3n$.

In the large scale concurrent system the traditional Kung-Leiserson BMMSA is wasteful architecture. To realize higher performance and more efficient multiplier in recent VLSI system design, we compress the data of the input matrices and increase the operation speed of each step in this thesis. Skillful cell arranging enables the final computation correctness and provides higher multiplication cell efficiency. In addition, new network grid connection architectures also increase the global processing speed and enhance the system operation efficiency.

4.2.2 Super Pipeline Array Design Methods

It is obvious that the matrix array in the Figure 2.2 is sparse and easy to be compressed to the compact operation style. Thus if the data matrices can be condensed, its cell efficiency will increase greatly. Moreover, the operation for different steps can be processed in the same time, even faster than common pipeline operation. These ideas can be called as the "Super Pipeline" matrix multiplication design method.

After the matrix C is condensed, the matrix A and matrix B become more compact processing array. Based on the Eq. (4.5) the array elements in matrix A and matrix B are rearranged and many new elements are added into the input matrices. The cell operation sequence will also be adjusted, otherwise the multiplicator and multiplicand may be lagged to influence the computation. In this situation the laggard data must be moved up to realize the correct multiplication operation as Figure 4.3.

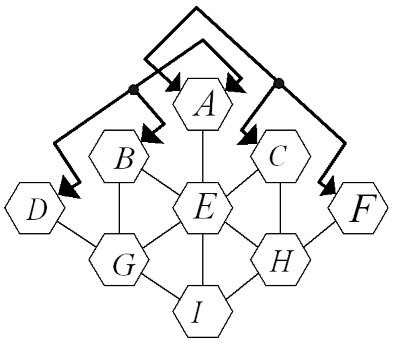


Figure 4.3: Move up the matrix data laggard numbers.

If the matrix cell arrangement can not solve the data unmatched situation, we add some additional processing cells to preserve the data temporarily and satisfy next step computation as shown in Figure 4.4.

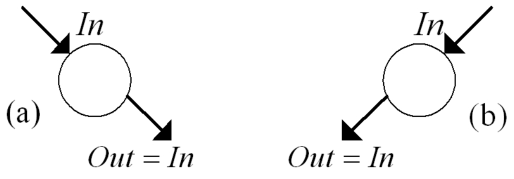


Figure 4.4: Additional data-store cells. (a) move right, (b) move left.

If moving up the laggard numbers and adding additional cells can not realize the correct multiplication function for the compressed matrices, the operation sequence can be changed to satisfy the Eq. (4.2) multiplication relationship and achieve the correct system operation. For example, if the data c_{33} is calculated in the given matrix multiplication, the traditional Kung-Leiserson BMMSA operation sequence is "$a_{31}b_{13}$ => $a_{32}b_{23}$ => $a_{33}b_{33}$ => $a_{34}b_{43}$", but our proposed super pipeline design operation sequence is "$a_{31}b_{13}$ => $a_{34}b_{43}$ => $a_{33}b_{33}$ => $a_{32}b_{23}$" just like the data "jumping" in the whole systolic array as shown in Figure 4.5. Then the matrix data are fully pipelined to realize the global concurrent multiplication operation.

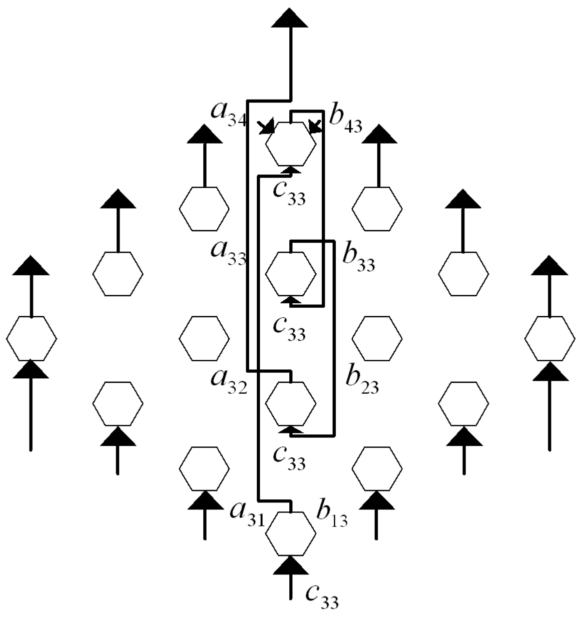


Figure 4.5: Change the operation sequence in the matrix multiplication.

Based on the proposed idea of "Super Pipeline", three different BMMSA (Band Matrix Multiplication Systolic Array) architectures are implemented to construct the new matrix multiplication processing array. In the wide-banded matrices situation, these BMMSA designs can realize higher cell efficiency and faster processing speed than the traditional Kung-Leiserson BMMSA design.

4.2.3 Super Pipeline Array Architectures

Through matrix compressing and sequence readjusting, we propose three BMMSA designs in wide-banded matrices situation. The detailed configurations and the corresponding characteristics for new multiplication are shown as follows.

4.2.3.1 Kung-Leiserson BMMSA

This BMMSA design is the traditional matrix multiplication design and its characteristics are given as follows [4]. It is obvious that its efficiency is low because in each step only 1/3 PEs are in operation as shown in Figure 2.2.

1. The involved number of the PE is "w_1w_2" (w_1 and w_2 are the bandwidths of the two operand matrices).

2. Every three cycles the data are pushed into the array.

3. Each PE is active every three cycles. Thus the average utilization rate is about 33% in full stream.

4. The throughput latency is "$3n + min(w_1, w_2)$".

5. The maximal input bandwidth is about "$2(w_1 + w_2)/3$" per cycle.

4.2.3.2 Two-Space-Parallel BMMSA

The distance of each neighboring element in this BMMSA design is two spaces. And the laggard numbers are moved up to satisfy the matrix multiplication requirements. Thus its operation efficiency is improved greatly and its multiplication speed is increased rapidly. Its characteristics are summarized as follows (Figure 4.6):

1. The organization of the data is similar to that in the Kung-Leiserson BMMSA.

2. The involved number of the PE is "w_1w_2".

3. Each neighboring element distance is two spaces. Every two cycles the data are pushed into the array.

4. Each PE is used every two cycles. Thus the average utilization rate is about 50% in full stream.

5. The throughput latency is "$2n + min(w_1, w_2)$".

6. The maximal input bandwidth is about "$3(w_1 + w_2)/2$" per cycle.

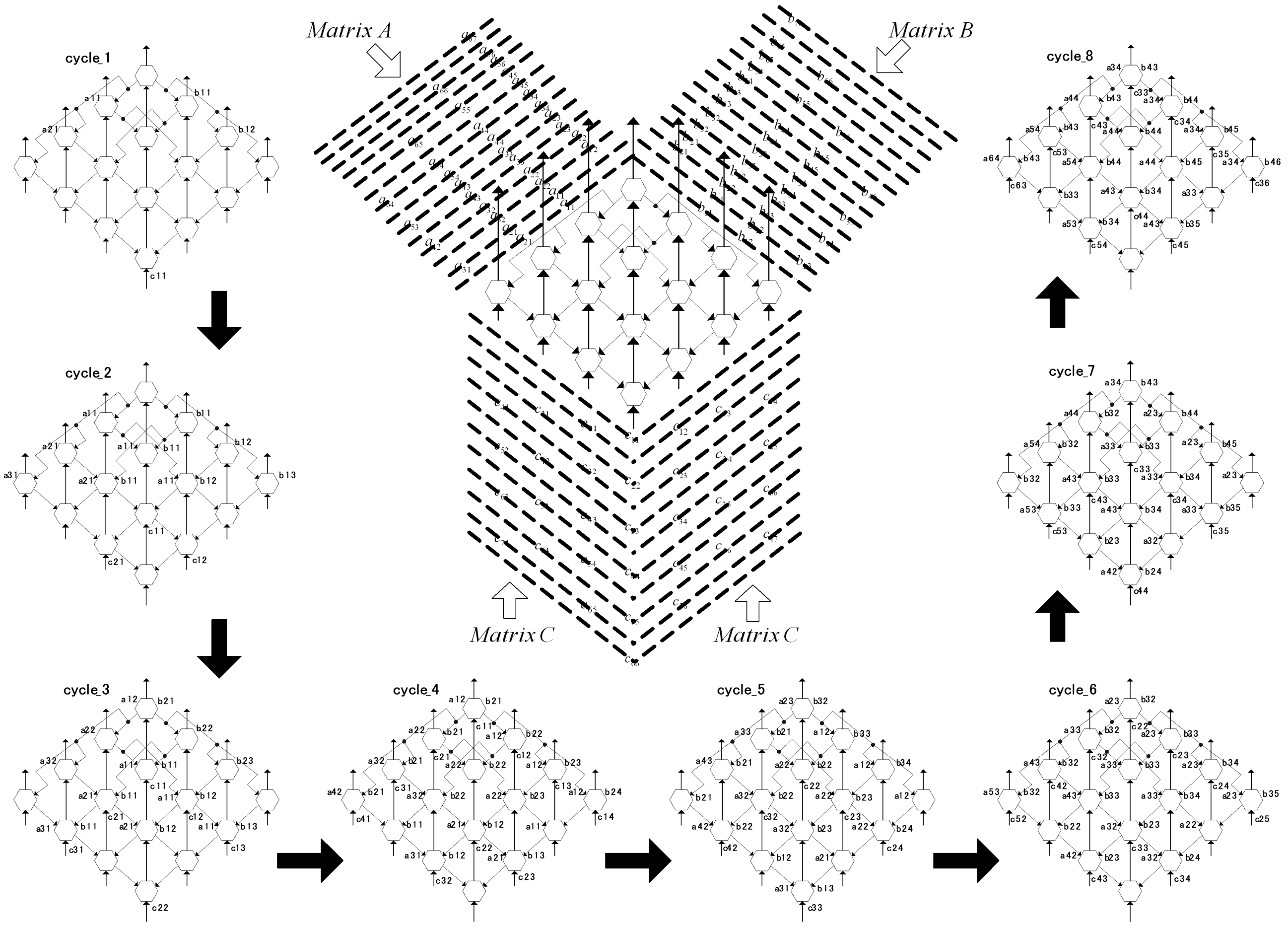


Figure 4.6: Pipelined configurations of the Two-Space-Parallel BMMSA design and its first eight cycle operation.

4.2.3.3 One-Space-Parallel BMMSA

In this BMMSA design the input data matrices are compressed further and the matrix corresponding distance of each neighboring element is one space. But it is difficult to satisfy the multiplication requirements in this situation. Thus the additional cells must be added with the operation sequence adjustments. After these modifications we can get the detailed matrix multiplication characteristics as follows (Figure 4.7):

1. The organization of the data is similar to that in the Kung-Leiserson BMMSA.

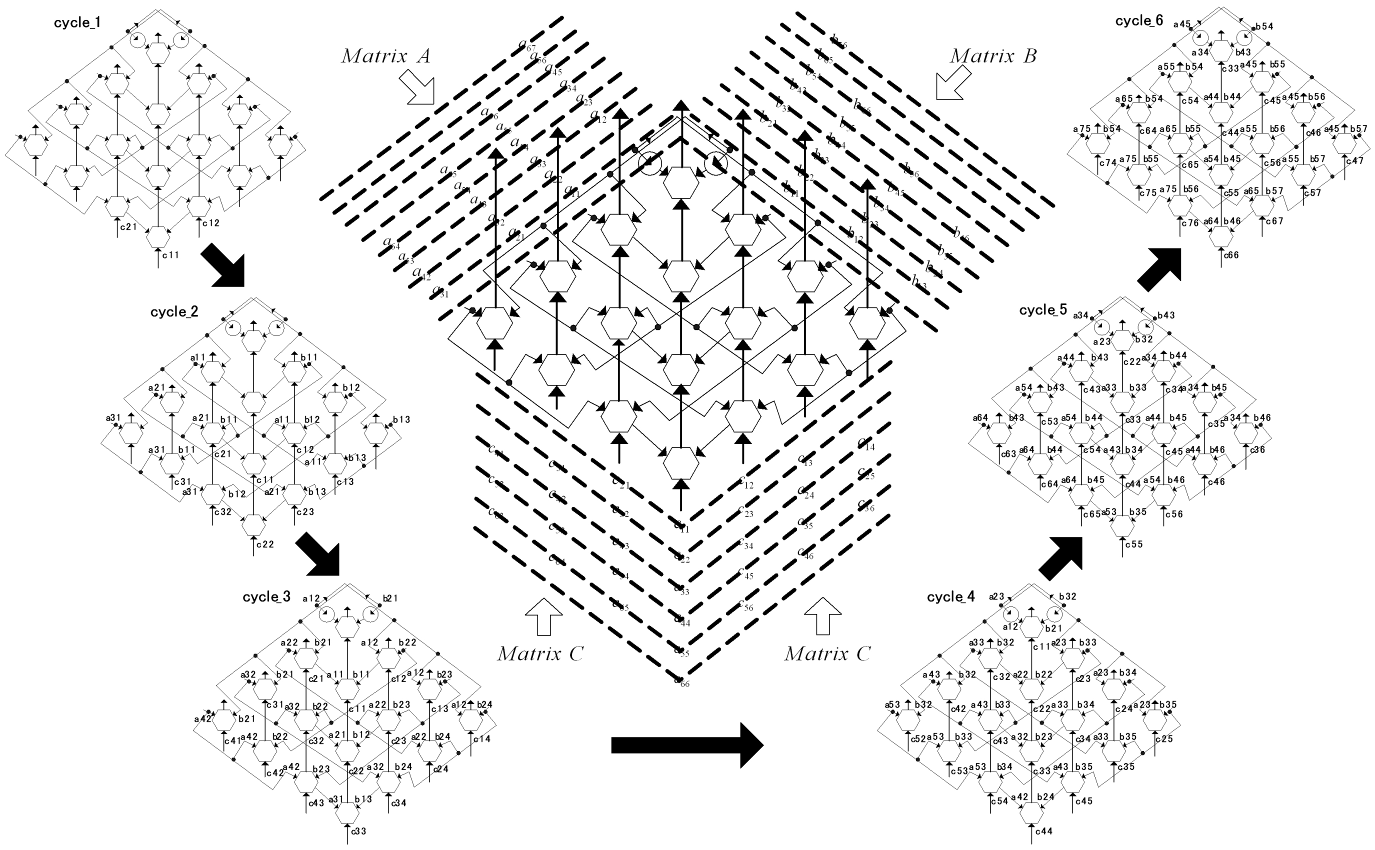


Figure 4.7: Pipelined configurations of the One-Space-Parallel BMMSA design and its first six cycle operation.

2. The involved number of the PE is "$w_1 w_2 + 2$".

3. Each neighboring element distance is one space. In every cycle the data are pushed into the array.

4. Almost all PEs are active and the average utilization rate is about 90% in full stream situation.

5. The throughput latency is "$9n/8 + min(w_1, w_2)$".

6. The maximal input bandwidth is about "$2(w_1 + w_2)$" per cycle.

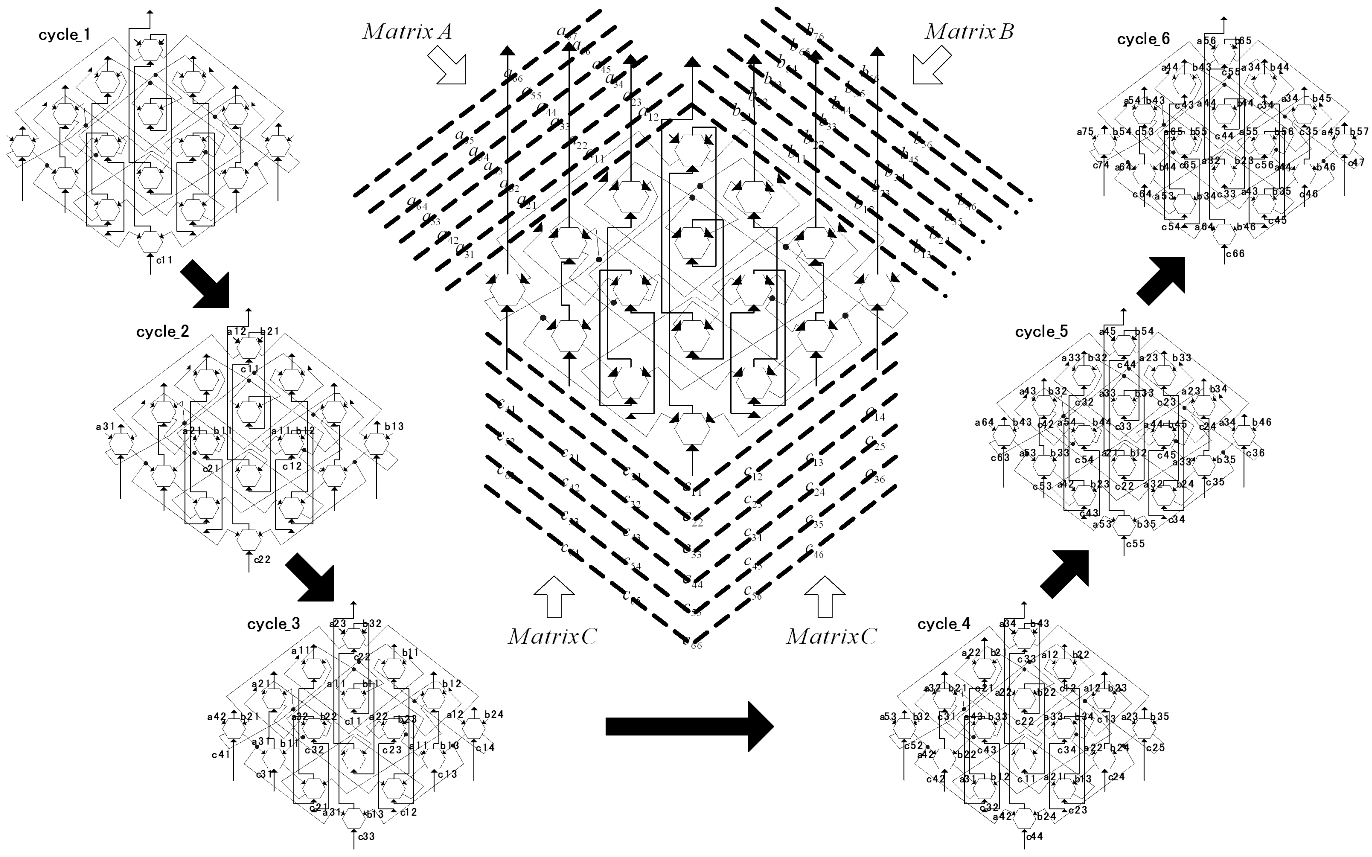


Figure 4.8: Pipelined configurations of the One-Space-Jumping BMMSA design and its first six cycle operation.

4.2.3.4 One-Space-Jumping BMMSA

In this BMMSA design the input matrices are also compressed into one space. And the operation sequence waves just like data “jumping” from one part to another parts to realize the super pipeline operation as shown in Figure 4.5. Though its pipelined configurations are more complex, the data matrix can be more compact and the additional cells can be removed by skillful sequence arrangement (Figure 4.8):

1. The organization of the data is similar to that in the Kung-Leiserson BMMSA.

2. The involved number of the PE is “w_1w_2”.

3. Each neighboring element distance is one space. In every cycle the data are pushed into the array.

4. Almost all PEs are active and the average utilization rate is about 100% in full stream situation.

5. The throughput latency is "$n + min(w_1, w_2)$".

6. The maximal input bandwidth is about "$2(w_1 + w_2)$" per cycle.

4.2.4 Super Pipeline Array Experimental Results

It is obvious that the proposed new BMMSA designs have better cell efficiency and faster operation speed than the Kung-Leiserson BMMSA. Moreover, drive capabilities are not similar. Table 4.1, Table 4.2, and Table 4.3 show their detailed characteristics in each cycle. And the more systolic array information about PE utilization rate in different cycle is given in Figure 4.9. All the corresponding attributes, such as utilization rate, throughput latency and data bandwidth, are also listed in Table 4.4, where w_1 and w_2 are the bandwidths of the two operand matrices, the average utilization equals to the division value between the average active PE per cycle and the total PE number, and $min(w_1, w_2)$ means the cost before the BMMSA is stable.

Table 4.1: Various BMMSA PE utilization value according to different cycles.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Kung-Leiserson	0	0	1	3	5	5	6	5	5	6
Two-Space-Parallel	0	0	5	10	6	10	6	10	6	10
One-Space-Parallel	0	6	15	16	16	16	16	16	16	16
One-Space-Jumping	0	3	9	15	16	16	16	16	16	16

Table 4.2: Various PE utilization rate according to different cycles (%).

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Kung-Leiserson	0	0	6.25	18.75	31.25	31.25	37.5	31.25	31.25	37.5
Two-Space-Parallel	0	0	31.25	62.5	37.5	62.5	37.5	62.5	37.5	62.5
One-Space-Parallel	0	33.33	83.33	88.89	88.89	88.89	88.89	88.89	88.89	88.89
One-Space-Jumping	0	18.75	56.25	93.75	100	100	100	100	100	100

Table 4.3: Drive capability according to different BMMSA designs.

Drive numbers	**Drive one PE**	**Drive two PEs**	**Drive three PEs**	**Drive four PEs**	**Total**
Kung-Leiserson	48	0	0	0	48
Two-Space-Parallel	38	2	2	0	42
One-Space-Parallel	24	0	6	2	32
One-Space-Jumping	32	2	4	0	38

These BMMSA designs have their own characteristics and satisfy the different array situation as shown in Figure 4.9 and Table 4.4. The traditional Kung-Leiserson BMMSA design is suitable for the common systolic array situation. And the Two-Space-Parallel BMMSA design has higher efficiency with unstable PE utilization rate (40%∼60%). For the One-Space-Parallel BMMSA design case, its cell efficiency and array stableness are much better, but two additional PEs increase the array complexity and influence its drive capability. Finally the One-Space-Jumping BMMSA design can realize the maximal utilization rate (100%) and minimal throughput latency ("$n + min(w_1, w_2)$") by jumping its data from one part to another parts to create the super pipeline multiplication operation. Though its configurations are the most complex in these BMMSA designs, it can spend only 1/3 time to complete the operation and reach the stable state much faster than the traditional Kung-Leiserson BMMSA design.

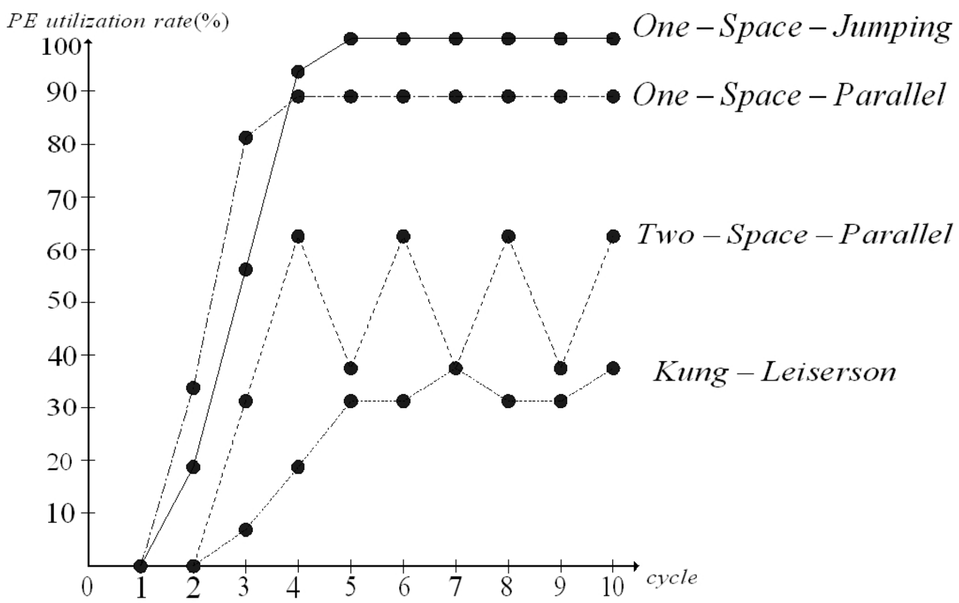


Figure 4.9: PE utilization rate for four BMMSA designs in different cycle.

Table 4.4: Performance comparisons of four BMMSA designs.

	Kung-Leiserson	**Two-Space-Parallel**	**One-Space-Parallel**	**One-Space-Jumping**
PE Scale	w_1w_2	w_1w_2	$w_1w_2 + 2$	w_1w_2
Input Bandwidth	$2(w_1 + w_2)/3$	$3(w_1 + w_2)/2$	$2(w_1 + w_2)$	$2(w_1 + w_2)$
Throughput Latency	$3n + min(w_1, w_2)$	$2n + min(w_1, w_2)$	$9n/8 + min(w_1, w_2)$	$n + min(w_1, w_2)$
Average Utilization Rate	33%	50%	90%	100%

4.3 Jumping Systolic Array (JSA) Grid Algorithm

Based on the elementary systolic architecture, many developments and advancements are exerted to reduce throughput latency, increase average utilization rate, enhance bandwidth ability, fortify data transmission robustness, or other characteristic improvements. These new systolic arrays enable more highly data pipeline and faster operation speed for VLSI processor and matrix multiplication, and advance system LSI design development. In this thesis, we focus on the large scale two-dimension systolic array improvement and try to increase the matrix multiplication operation efficiency by the proposed "Jumping Systolic Array (JSA)" method to allow high performance pipelined multiplication operation.

4.3.1 Jumping Systolic Array Procedure Description

For the three computation direction systolic array as **Algorithms 4.1**, the Euclidean space axes of $i(1,0,0)$, $j(0,1,0)$ and $k(0,0,1)$ are basic concurrent operation vectors. To optimize the systolic concurrent characteristics, the common 2-D systolic array designs focus on array element modification, peripheral element supplement, or data matrix adjustment [69, 90, 94]. According to our consideration, new linked wire modifications among processing elements enable array data jumping in concurrent computation and space time optimization to improve the system operation efficiency.

4.3.1.1 Forward/Backward Jumping Systolic Array

Consider the array data jumping idea between processing elements, which means that the PE operation data can jump in the systolic array according to specific rules and suitable sequences, the interim computation results will be sent to the next element as fast as possible. The jumping linked wire can be used to quicken the interim results into next step computation as shown in Figure 4.10, where linked wires jump across the following PEs and connect several distance later processing elements. There are some jumping connections which pass data back to the previous PEs and compute concurrent systolic operation ahead. Based on the swing wire connection method, we consider this type as "Forward/Backward Jumping Systolic Array" and the detailed array data jumping method is described in **Algorithms 4.2** by the result output dimension case.

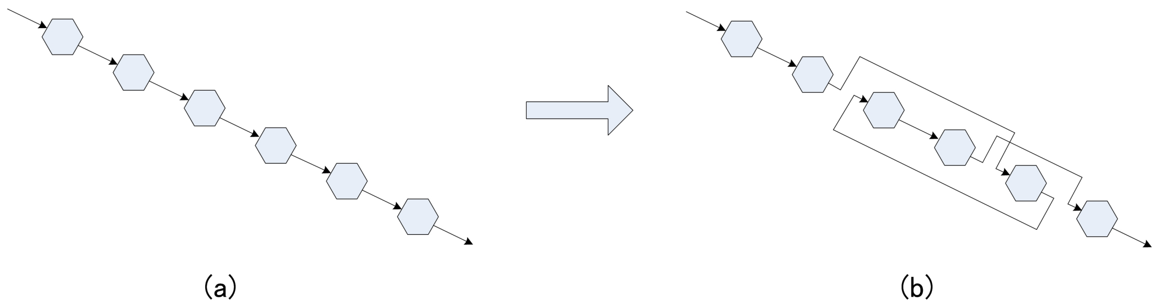


Figure 4.10: Array data jumping for forward/backward jumping systolic array. (a) original systolic array, (b) forward/backward jumping systolic array.

Algorithms 4.2 : (***Forward/Backward jumping systolic array***)

```
for $k = 1$ to $n$
    for $j = 1$ to $n$
        for $i = 1$ to $n$
            begin
                $a(i,j,k) \ = \ a(i,j-1,k)$
                $b(i,j,k) \ = \ b(i-1,j,k)$
                $c(i,j,k) \ = \ c(i,j,k-w) \ + \ a(i,j,k) \times b(i,j,k)$
            end
```

where

$$a(i,0,k) \ = \ a_{ik}, \ \ b(0,j,k) \ = \ b_{kj}, \ \ c(i,j,0) \ = \ c_{ij}^{0} \ = \ 0.$$

Parameter $w \in [-n, n]$ is the former jumping positions in whole systolic direction. If w is negative, the backward jumping occurs in the systolic array.

4.3.1.2 Spreading Jumping Systolic Array

Another jumping systolic array architecture other than the forward/backward jumping method is "Spreading Jumping Systolic Array" as shown in Figure 4.11, where the output data element can drive several later PEs to enable highly concurrent systolic operation based on suitable array data jumping process. The differences between spreading and broadcast are array connection method and data matrix combination. The spreading method uses irregular broadcast point and PE connection can be divided to several independent multi-driver regions, other than broadcast propagation vectors which are derived from uniform broadcast point and are continuous in data propagation direction. The spreading method can also extent PE linked wires around one middle spreading point and differs from single propagation direction broadcast vectors. Because of wire spreading connection style, this jumping method is called as "Spreading Jumping Systolic Array" as shown in **Algorithms 4.3** by the data in-out dimension case.

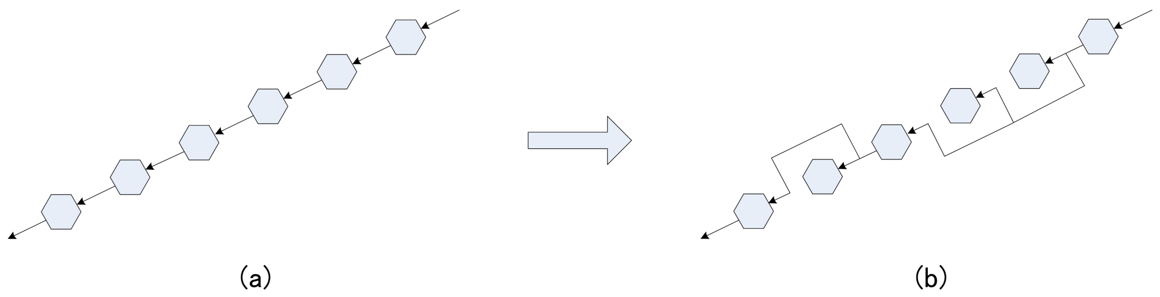


Figure 4.11: Array data jumping for spreading jumping systolic array. (a) original systolic array, (b) spreading jumping systolic array.

Algorithms 4.3 : (*Spreading jumping systolic array*)

```
for $k = 1$ to $n$
    for $j = 1$ to $n$
        for $i = 1$ to $n$
```

```
begin
    $a(i,j,k) = a(i,j-u,k)$
    $b(i,j,k) = b(i-v,j,k)$
    $c(i,j,k) = c(i,j,k-1) + a(i,j,k) \times b(i,j,k)$
end
```

where

$$a(i,0,k) = a_{ik}, \;\; b(0,j,k) = b_{kj}, \;\; c(i,j,0) = c_{ij}^{0} = 0.$$

The given parameters $u \in [1,n]$ and $v \in [1,n]$ mean the spreading positions when they are variable, or the irregular spreading systolic array output port positions if they are constant values.

4.3.1.3 Jumping Systolic Array Combination

Based on the basic highly concurrent analysis of forward/backward jumping systolic array and spreading jumping systolic array, the common jumping systolic array can be considered as two jumping architecture combination as shown in Figure 4.12, where two-dimension triangular connection topology is given as the detailed analysis example. The triangular array example uses three borders to express the jumping combination procedure, including the left vertical border for the forward/backward interim output results and the top/down inclined border for the spreading multiplication data transmission. The following **Algorithms 4.4** gives the specific description for the jumping systolic array combination, where the parameters u, v mean the spreading jumping positions of multiplication transmission data and another parameter w shows the forward/backward jumping interim output flow.

Algorithms 4.4 : (*Jumping systolic array combination*)

```
for $k = 1$ to $n$
    for $j = 1$ to $n$
        for $i = 1$ to $n$
```

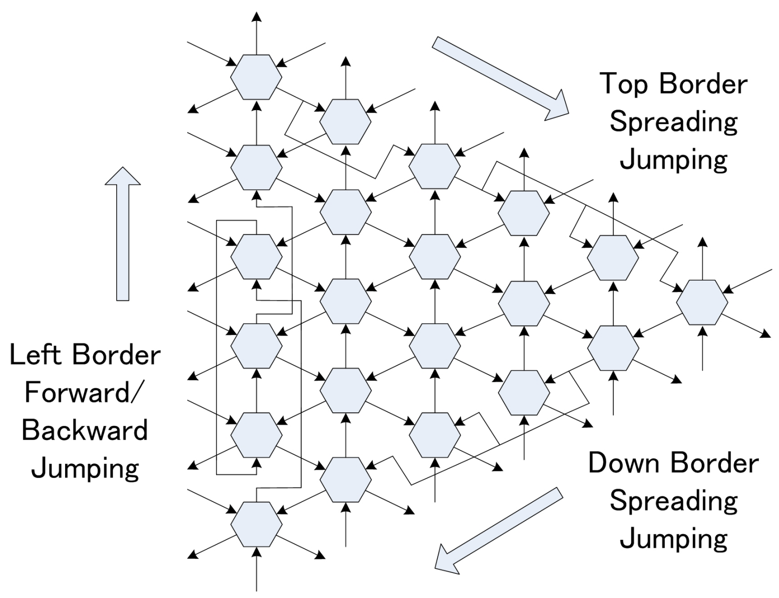


Figure 4.12: The architecture example of jumping systolic array combination of 2-D triangular connection topology.

```
begin
    $a(i,j,k) = a(i,j-u,k)$
    $b(i,j,k) = b(i-v,j,k)$
    $c(i,j,k) = c(i,j,k-w) + a(i,j,k) \times b(i,j,k)$
end
```

where

$$a(i,0,k) = a_{ik}, \;\; b(0,j,k) = b_{kj}, \;\; c(i,j,0) = c_{ij}^{0} = 0.$$

The given parameters $u \in [1,n]$, $v \in [1,n]$ and $w \in [-n,n]$ mean the former forward/backward result positions, or the spreading data element positions.

4.3.2 Two-Dimension Square Topology Jumping Systolic Array

The traditional 2-D square topology systolic array is illustrated as shown in Figure 4.13, with two data input sequences from two opposite sides of processing element (PE) and one interim result systolic sequence in vertical direction [90,91,94]. To com-

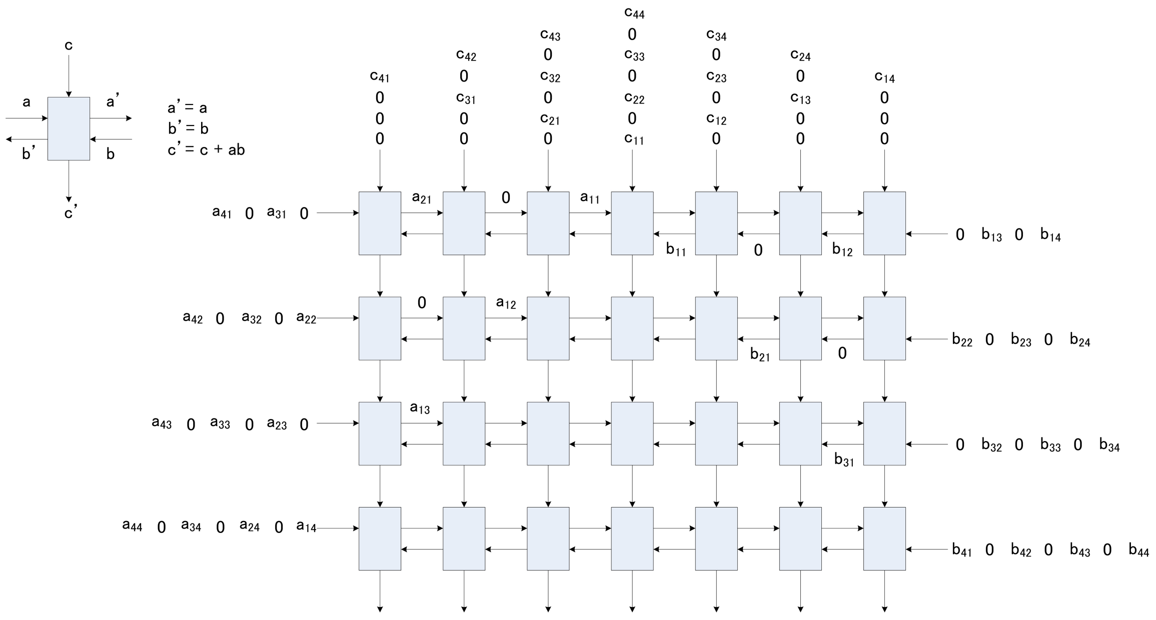


Figure 4.13: The traditional 2-D square topology systolic array architecture.

pute the matrix multiplication, the data supply latency is one cycle, and PEs are idle every other cycle. The planar square topology characteristics can be described as following characteristics statement.

The Traditional Square Array

1. The organization of three matrices data are shown in Figure 4.13 with two horizontal opposite sequences and one vertical systolic sequence.

2. Each set of pipelines (For A, B, and C of three direction matrices, respectively) is divided into two groups. In other words, each pipeline pumps data into the systolic array once every other cycle.

3. Each PE is active once every other cycle. Thus, if the multiplication operation is in full stream, the per-cycle PE utilization rate is about 50 percent.

4. The throughput latency is $T = 2n + t_{sat}$, where n is the data matrix size and t_{sat} is the stable time before systolic array reaches full stream operation.

Based on the proposed jumping systolic array method, new planar square topology systolic array can be designed by data matrices compression and spreading jumping systolic array as shown in Figure 4.14, where the left part matrix data jump to right half part with respect to the middle processing element, and the right part matrix data also jump to left half part by the spreading connection method. Then multiplication parameters appear to the required PE input ports ahead of the schedule together, and the system operation is accelerated to increase PE utilization rate and reduce throughput latency. Thus the proposed 2-D square topology jumping systolic array characteristics are summarized as follows.

The Proposed Square Jumping Array

1. The arrangement and progression of data matrices A, B, and C are similar to the traditional 2-D square topology with no dummy data.

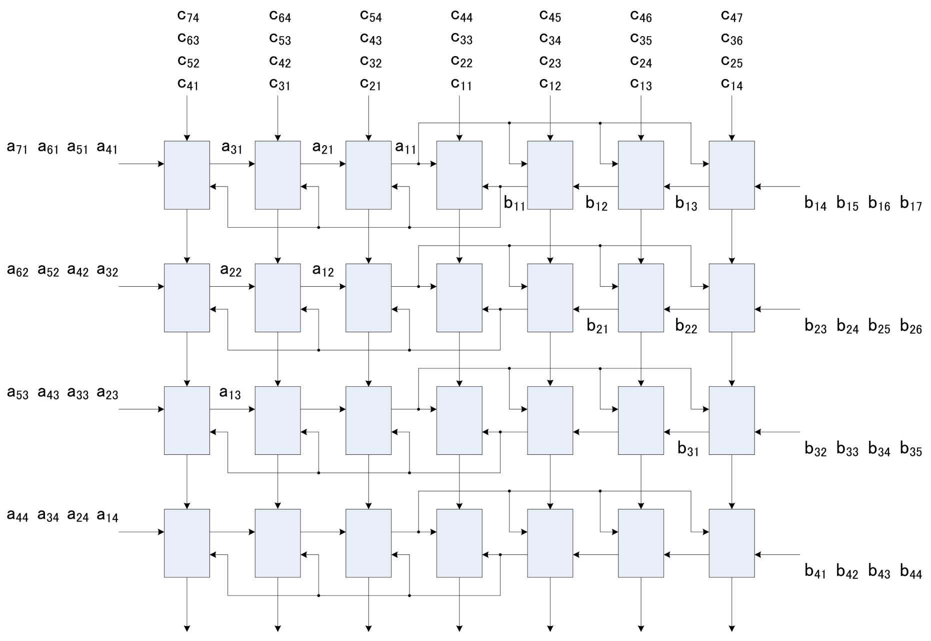
2. Each set of pipelines has no grouping subdividing. Or each pipeline pumps data into systolic array every cycle without traditional cycle waste.

3. Each PE is active once the data flows reach full stream. Then the per-cycle PE utilization rate is about 100 percent.

4. The throughput latency is $T = n + t_{sat}$, where t_{sat} is the stable time before full stream operation.

Algorithms 4.5 : (*Proposed 2-D square jumping systolic array*)

```
for $k = 1$ to $n$
    for $j = 1$ to $n$
        for $i = 1$ to $n$
            begin
                if $(j \leq [n/2])$        $a(i,j,k) = a(i,j-1,k)$
```



c_{74} c_{63} c_{52} c_{41}
c_{64} c_{53} c_{42} c_{31}
c_{54} c_{43} c_{32} c_{21}
c_{44} c_{33} c_{22} c_{11}
c_{45} c_{34} c_{23} c_{12}
c_{46} c_{35} c_{24} c_{13}
c_{47} c_{36} c_{25} c_{14}
a_{71} a_{61} a_{51} a_{41}
a_{31}
a_{21}
a_{11}
a_{62} a_{52} a_{42} a_{32}
a_{22}
a_{12}
a_{53} a_{43} a_{33} a_{23}
a_{13}
a_{44} a_{34} a_{24} a_{14}
b_{11}
b_{12}
b_{13}
b_{14} b_{15} b_{16} b_{17}
b_{21}
b_{22}
b_{23} b_{24} b_{25} b_{26}
b_{31}
b_{32} b_{33} b_{34} b_{35}
b_{41} b_{42} b_{43} b_{44}

Figure 4.14: The proposed 2-D square topology jumping systolic array.

```
        else    $a(i,j,k) = a(i,[n/2],k)$
        if $(i \leq [n/2])$    $b(i,j,k) = b(i-1,j,k)$
        else    $b(i,j,k) = b([n/2],j,k)$
        $c(i,j,k) = c(i,j,k-1) + a(i,j,k) \times b(i,j,k)$
    end
```

where

$$a(i,0,k) = a_{ik}, \;\; b(0,j,k) = b_{kj}, \;\; c(i,j,0) = c_{ij}^0 = 0,$$

The given value $[n/2]$ in the algorithm means the middle positions of horizontal spreading jumping sequences.

4.3.3 Two-Dimension Hexagonal Topology Jumping Systolic Array

The hexagonal systolic array is general two-dimension topology for practical pipeline operation because of its highly concurrent computation ability. In this section, new systolic architectures have been studied based on traditional hexagonal systolic array and the proposed jumping systolic architecture.

4.3.3.1 Traditional Two-Dimension Hexagonal Topology

Some traditional 2-D hexagonal topologies have been proposed by Kung-Leiserson and Huang-Abraham (Figure 4.15) [4, 6, 96]. To realize highly system pipeline operation, the processing element allocation and data matrix combination are important to construct the hexagonal topology. For Kung-Leiserson architecture, input data matrices and output result matrix are inserted from three different directions (Figure 4.15(a)), and Huang-Abraham array sends data matrices from one similar direction (Figure 4.15(b)). Thus the I/O congestion difference determines the specific applications for these two traditional hexagonal topologies as wide-banded matrices and narrow-banded matrices, respectively. The attribute of Kung-Leiserson array is shown in Section 4.2.3.1. To compare the hexagonal topology characteristics, we summary Kung-Leiserson array again together with Huang-Abraham array as follows [6].

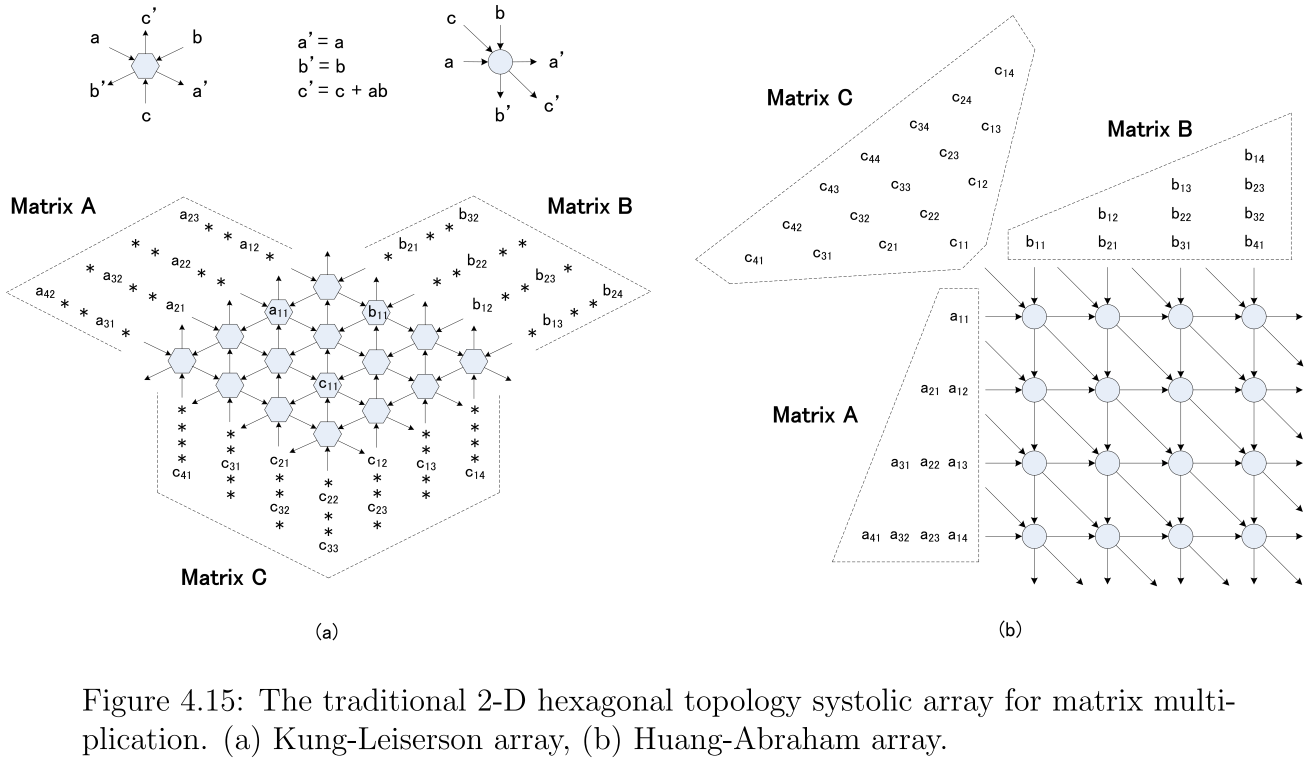


Figure 4.15: The traditional 2-D hexagonal topology systolic array for matrix multiplication. (a) Kung-Leiserson array, (b) Huang-Abraham array.

The Traditional Kung-Leiserson Array

1. The arrangement and progression of data matrices A, B, and C compose the hexagonal architectures and the import/export data matrices with highly system concurrent operation.

2. Each set of pipelines has three subgroup dividing, and in each cycle only one group sends matrix data into the array. Or each pipeline pumps data into the systolic array every three cycles if ignoring original cycle waste.

3. Once every three cycles each PE is active if data flows reach full stream. Then per-cycle PE utilization rate is about 33 percent.

4. The throughput latency is $T = 3n + t_{sat}$, where stable time t_{sat} means the redundant time before full stream operation.

The Traditional Huang-Abraham Array

1. The organization of data matrices A, B, and C is similar to Kung-Leiserson array. But the input/output systolic array direction is one similar orientation, not like three direction situation for the Kung-Leiserson array.

2. Each set of pipelines has no subgroup dividing, or each pipeline pumps data into systolic array every cycle if ignoring original cycle waste.

3. Once every cycle each PE is active if data flows reach full stream. Then per-cycle PE utilization rate is about 100 percent.

4. The throughput latency is $T = n + t_{sat}$, where stable time t_{sat} is the redundant time before full stream operation.

4.3.3.2 Redundant Dummy Latency Hexagonal Topology

The traditional hexagonal topology systolic array computes the results with highly pipelined thread and concurrent operation. However, the original Kung-Leiserson array wastes many PE cycles and has about 33% utilization rate. The improved systolic array, Huang-Abraham architecture, can realize 100% utilization rate by crowd input/output flow, which makes it suitable for narrow banded matrices application, compared with wide banded Kung-Leiserson array [6]. Then many architectures are introduced to increase the wide banded matrix systolic array efficiency. One popular method is matrix data sequence adjustment, but input data sequence changes influence the operation stream efficiency and impede the system robustness [94]. Another common used method is processing element variety, which causes unnecessary systolic array complexity [69]. Our previous research in this thesis former section proposes the super pipeline hexagonal topology, but the output result data sequence varies frequently and causes it no easy to be extended to large size systolic array [4]. Thus in large scale hexagonal topology situation, new hexagonal topology is necessary to be developed for the usually wide banded matrix application.

Based on the proposed data jumping idea, new hexagonal topology is given as shown in Figure 4.16, where output result directions keep unchanged for easy array size expansion. The matrix data jumping happens just in the tilted data systolic directions. By systolic array arrangement, the required matrix data appear in the latter PE input ports synchronously. The redundant dummy latency as array outside connection elements in Figure 4.16 can adjust operation sequences and satisfy systolic computation requirements. Thus regular data jumping arrangement and unchanged result output sequence enable the systolic array large scale implementation and expand the wide banded matrix application, as **Algorithms 4.6** descriptions.

The Proposed Redundant Dummy Latency Jumping Hexagonal Array

1. The array architecture is similar to traditional Kung-Leiserson array. But the

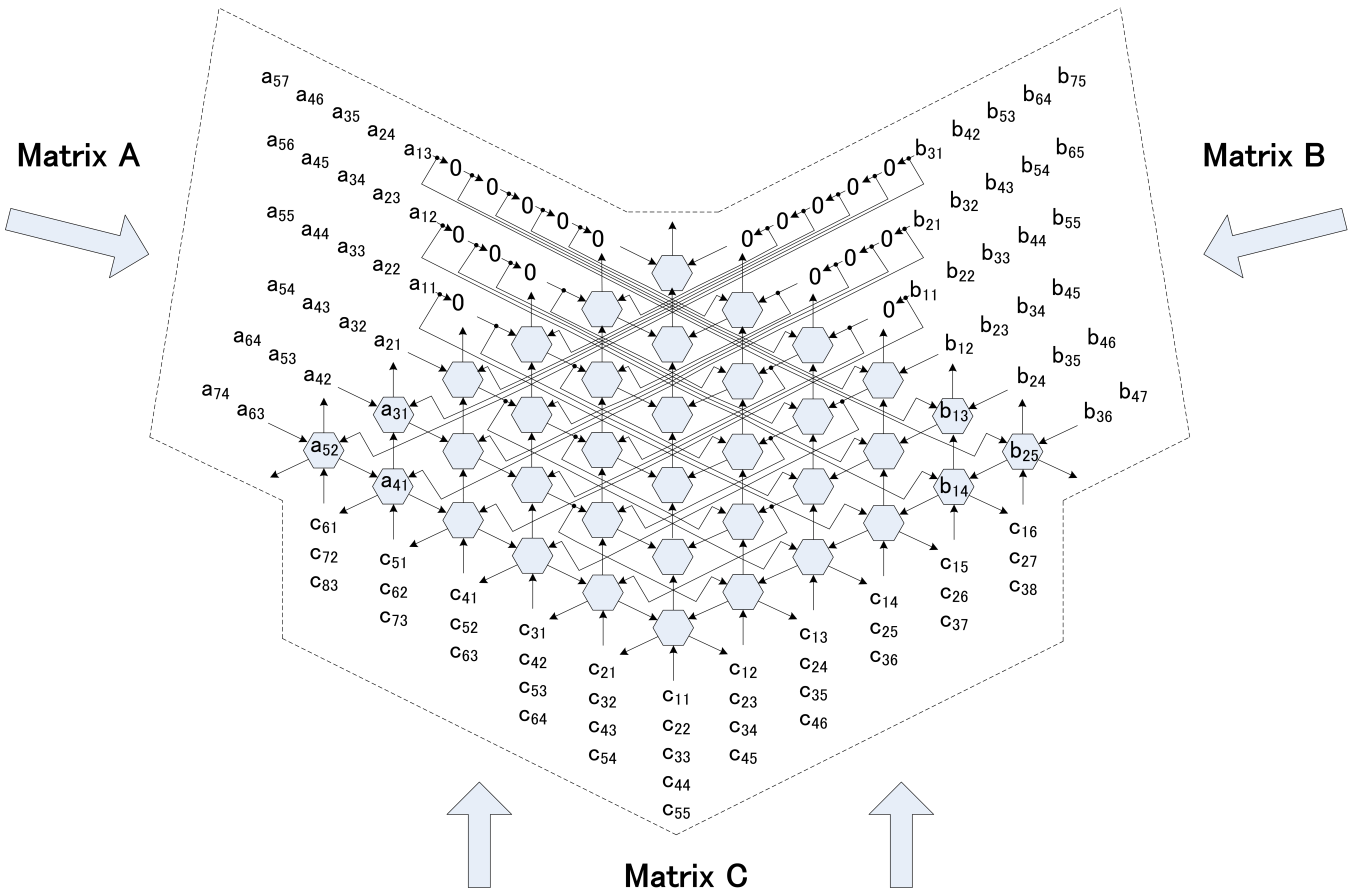


Figure 4.16: The new 2-D hexagonal topology jumping systolic array with redundant dummy latency for matrix multiplication operation.

input/output data matrices are more compact than Kung-Leiserson array. Some redundant dummy latencies, such as '0' positions outside PE array in Figure 4.16, are used to satisfy the operation sequence.

2. The matrix data appear to some PE input ports synchronously in the same tilted data systolic direction, and the regular linked wire connections ensure system systolic computation correctness and array size expansion ability.

3. Each set of pipelines has no subgroup dividing, or each pipeline pumps data into the systolic array every cycle if ignoring original cycle waste.

4. Once every cycle each PE is active if the data flows reach full stream. Then per-cycle PE utilization rate is about 100 percent.

5. The throughput latency is $T = n + t_{sat}$, where stable time t_{sat} is the redundant time before full stream operation.

Algorithms 4.6 :

(***Proposed 2-D redundant dummy latency jumping hexagonal array***)

```
for $k = 1$ to $n$
    for $j = 1$ to $n$
        for $i = 1$ to $n$
            begin
                if $(j > i)$    $a(i,j,k) = a(i, j - 2(j - i), k)$
                else    $a(i,j,k) = a(i, j - 1, k)$
                if $(i > j)$    $b(i,j,k) = b(i - 2(i - j), j, k)$
                else    $b(i,j,k) = b(i - 1, j, k)$
                $c(i,j,k) = c(i,j,k-1) + a(i,j,k) \times b(i,j,k)$
            end
```

where

$$a(i, 0, k) = a_{ik},\ \ b(0, j, k) = b_{kj},\ \ c(i, j, 0) = c^0_{ij} = 0,$$

The n means total systolic array scale value, and whole PE number is $n \times n$. Then, i means the tilted PE direction from left upper to right lower and varies from 1 to n with the start value 1 in the most right upper row. Similarly, j is the tilted PE direction from right upper to left lower and the start value 1 lies in the most left upper row. Thus $(j > i)$ means the right part of hexagonal topology excluding the middle PE row, and $(i > j)$ is the left hexagonal array part (Figure 4.16).

4.3.3.3 Compact Parallel Flow Hexagonal Topology

The redundant dummy latency hexagonal topology can realize fully pipeline systolic operation, and its algorithm is regular and can be extended to large scale systolic array easily. However, the additional redundant dummy latency requires the extra peripheral registers to store the dummy latency and satisfy the system computation requirements. So the system complexity is increased and the unnecessary hardware resource is larger. Thus new hexagonal topology without the redundant dummy latency and the extra peripheral registers are required to implement better systolic system. In this section, the compact parallel flow hexagonal topology as shown in Figure 4.17 is proposed to increase the system concurrent efficiency with reduced extra hardware resource. As a result, new systolic architecture has better large size array characteristics than our previous super pipeline systolic architecture [4] by unchanged result output direction, and the compact parallel flow architecture also decreases the jumping systolic array processing element area.

The architecture of the proposed compact parallel flow hexagonal topology is obtained from basic wide banded 2-D hexagonal topology. The forward/backward data jumping and the spreading data jumping are used together to keep the pipeline flow correctness. The swing back method reduces the linked wire complexity and removes the extra peripheral registers. Though the connection topology is more difficult and the matrix scale expansion is not easy as redundant dummy latency architecture, the proposed hexagonal architecture is much suitable architecture for the compact hardware resource and small PE area situation. The following algorithm description gives the specific summarization for the compact parallel flow hexagonal topology as the illustration in Figure 4.17.

The Proposed Compact Parallel Flow Jumping Hexagonal Array

1. The array architecture is based on the traditional Kung-Leiserson array and the redundant dummy latency hexagonal topology. More compact matrix data

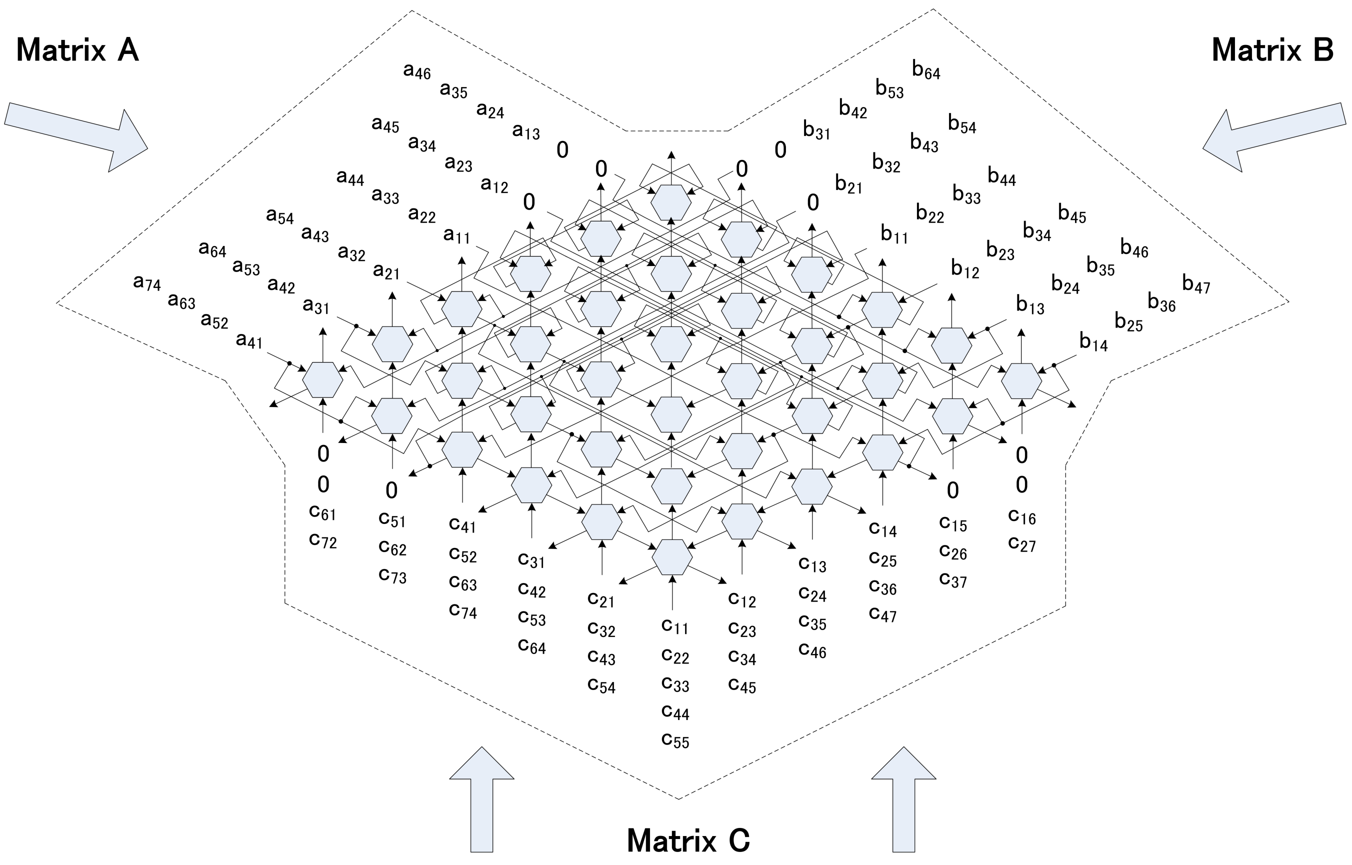


Figure 4.17: The proposed 2-D hexagonal topology jumping systolic array with compact parallel flow for matrix multiplication operation.

are also supplied to realize highly concurrent computation without redundant dummy latency and additional peripheral register.

2. The forward/backward jumping systolic architecture is used to adjust system matrix data sequences. The spreading jumping systolic array is also applied together to satisfy the system operation requirements.

3. Each set of pipelines has no subgroup dividing, or each pipeline pumps data into the systolic array every cycle if ignoring original cycle waste.

4. Once every cycle each PE is active if the data flows reach full stream. Then per-cycle PE utilization rate is about 100 percent.

5. The throughput latency is $T = n + t_{sat}$, where stable time t_{sat} is the redundant time before full stream operation.

4.3.4 Jumping Systolic Array Experimental Results

The proposed jumping systolic array (JSA) algorithm can increase the systolic efficiency and realize the compact array operation. Many ideal analyses and experimental results are performed to prove the JSA algorithm efficiency and find better systolic array compression methods. Thus the corresponding performance discussions for the multiplication system pipeline are given for further systolic array development and other connection topology designs.

4.3.4.1 Performance Comparisons

The detailed 2-D systolic array characteristic comparisons are given in Table 4.5, where n is the matrix width of the systolic array, t_{sat} is the redundant time before full stream operation, and Reg is the additional register number in redundant dummy latency hexagonal array. The proposed square jumping systolic array can realize smaller throughput latency and increase the average utilization rate to 100%. Reversely, the traditional square array requires longer time to reach system stabilization than square jumping array because of the incompact input/output data matrices. For the hexagonal topology, the proposed redundant dummy latency array requires additional register as Reg to store the dummy latency. The throughput latency for new jumping architectures is much smaller than traditional Kung-Leiserson systolic array. The average utilization rate also increases to 100% instead of original 33% rate. Though complex linked wires introduction, the expansion ability for proposed systolic architectures is also improved because of the result output route invariability. Also, the stabilization speed increases rapidly with much highly pipelined data flow.

Table 4.5: Performance comparisons of different two-dimension systolic array topologies for wide-banded matrix situation.

	Square topology		Hexagonal topology		
	2-D square array	square jumping array	Kung-Leiserson	redundant dummy latency	compact parallel flow
PE number	n^2	n^2	n^2	$n^2 + Reg$	n^2
Throughput latency	$2n + t_{sat}$	$n + t_{sat}$	$3n + t_{sat}$	$n + t_{sat}$	$n + t_{sat}$
Average utilization rate	50%	100%	33%	100%	100%
Expansion ability	$easy$	$easy$	$easy$	$normal$	$difficult$
Stabilization speed	$fair$	$fast$	$slow$	$fair$	$fast$

The characteristics of various two-dimension systolic array topologies are listed in Table 4.5 and Table 4.6, where *opposite* means that matrix data input/output direction lies in three opposite directions, *isodirection* means that matrix data input/output direction lies in one similar isogonic direction, *lateral* means that matrix data input/output direction lies in three lateral directions. The system difference and concurrent performance are also given for further comparison and evaluation. For example, the proposed jumping array is based on traditional wide-banded systolic array with opposite data directions, and the systolic array is complex by small array area trade-off. The expansion ability is normal and the stabilization speed is fast. As a result, compared with other systolic array architectures, the proposed jumping systolic array is better two-dimension systolic array architectures with synthesis performance merits in wide-banded array. Table 4.5 and Table 4.6 give the detailed performance descriptions and characteristic comparisons.

Based on the performance comparisons of various 2-D systolic array topologies in Table 4.5 and Table 4.6, the proposed jumping systolic array can be considered as the optimal architecture design for 2-D matrix multiplication, where the "optimal" means large performance improvement in average utilization rate or PE operation efficiency.

Table 4.6: Comparisons of characteristic and diagnosis performance for various two-dimension systolic array topologies.

	Kung-Leiserson	Huang-Abraham	Milentijevic-Milovanovic	Element-Change	Super-Pipeline	Jumping-Array
Systolic array characteristics	Traditional wide-banded	Improved narrow-banded	Data matrix adjustment	Array element variety	Super pipeline	Jumping systolic array
Data dimension	*opposite*	*isodirection*	*opposite*	*lateral*	*opposite*	*opposite*
Complexity	*simple*	*simple*	*complex*	*fair*	*complex*	*complex*
Array area	*large*	*small*	*medium*	*medium*	*medium*	*small*
Expansion ability	*easy*	*easy*	*difficult*	*normal*	*difficult*	*normal*
Stabilization speed	*slow*	*fast*	*fair*	*fair*	*fast*	*fast*

Other wide-banded systolic array architecture with opposite data direction can only realize partly PE operation efficiency, such as 33% rate for hexagonal topology and 50% rate for square topology. However, for the proposed JSA architecture, the average utilization rate can realize 100% operation rate after the array enters the stable state as shown in Figure 4.14, Figure 4.16 and Figure 4.17. Though the additional factors of complex connection, wire extension and driver capability are introduced, the proposed JSA algorithm can realize the optimal utilization rate or fully PE operation to increase the computation efficiency of matrix multiplication greatly.

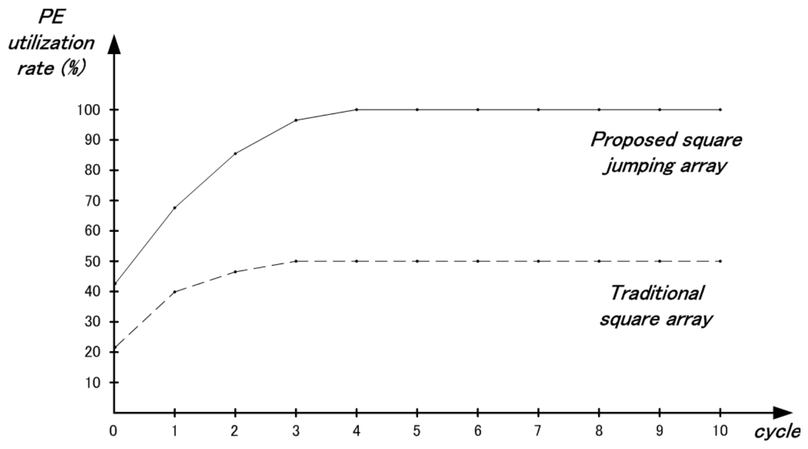
4.3.4.2 Speed Discussions

The jumping systolic array algorithm can speed up operation efficiency and increase array PE utilization rate in similar concurrent cycles. The detailed performance comparisons are given in Table 4.7, where the square array example uses 4×7 processing elements (28 PEs), the hexagonal array example uses 6×6 processing elements (36 PEs), and *Cycle* means different systolic array operation steps. Figure 4.18 describes the JSA algorithm efficiency for various two-dimension systolic topologies. The square array example is based on Figure 4.13 and Figure 4.14 with 4×7 processing elements

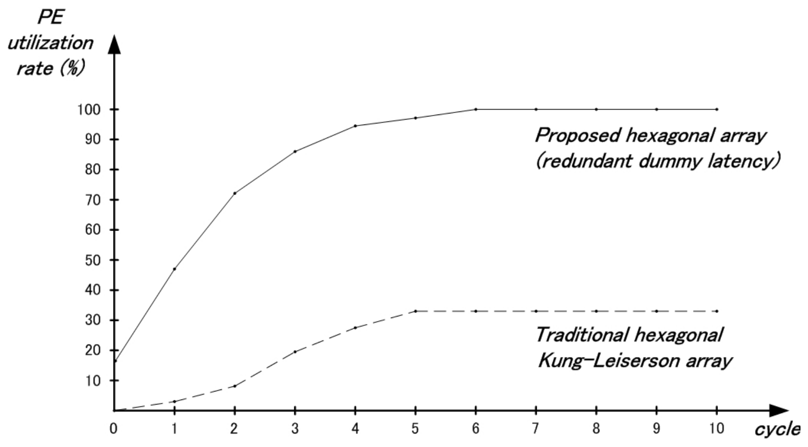
Table 4.7: Comparisons of PE utilization rate according to different cycles for various two-dimension systolic array topologies.

	Cycle	0	1	2	3	4	5	6	7	8	9	10
2-D square array	number	6	11	13	14	14	14	14	14	14	14	14
	rate(%)	21.43	39.29	46.43	50	50	50	50	50	50	50	50
square jumping array	number	12	19	24	27	28	28	28	28	28	28	28
	rate(%)	42.86	67.86	85.71	96.43	100	100	100	100	100	100	100
Kung-Leiserson	number	0	1	3	7	10	12	12	12	12	12	12
	rate(%)	0	2.78	8.33	19.44	27.78	33.33	33.33	33.33	33.33	33.33	33.33
redundant dummy latency	number	6	17	26	31	34	35	36	36	36	36	36
	rate(%)	16.67	47.22	72.22	86.11	94.44	97.22	100	100	100	100	100
compact parallel flow	number	0	17	26	31	34	35	36	36	36	36	36
	rate(%)	0	47.22	72.22	86.11	94.44	97.22	100	100	100	100	100

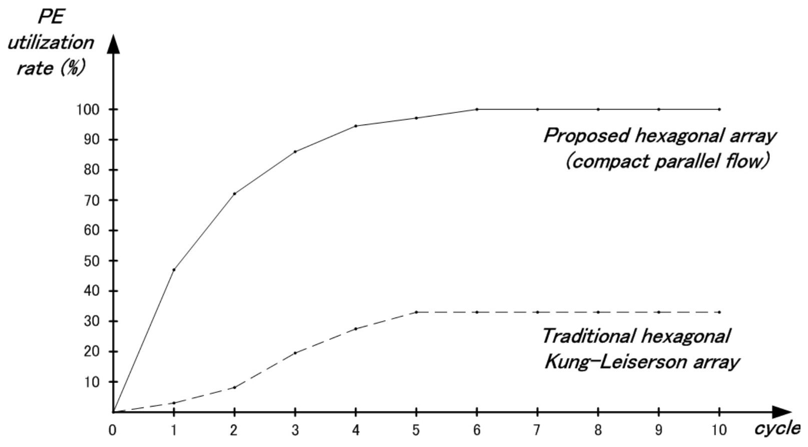
(28 PEs). The hexagonal array examples are derived from Figure 4.15, Figure 4.16, and Figure 4.17 by the 6×6 processing elements (36 PEs). Then PE utilization rate of new jumping systolic architectures increases rapidly, commonly two times for square array and three times for hexagonal array. The stabilization speed is fast and array system reaches high efficiency operation quickly. These results and improvements are based on the traditional wide-banded Kung-Leiserson array. Though other systolic array can also realize high efficiency system operation in some cases, the corresponding design constraints discussed in anterior sections, such as narrow-banded matrix for Huang-Abraham array, data compression complexity for Milentijevic-Milovanovic method and system expansion ability for Super-Pipeline architecture, limit practical systolic array application and reduce the system operation performance. Thus the proposed JSA method can improve the wide-banded matrix operation and promote the correlative two-dimension systolic array system design.



(a) square jumping array.



(b) redundant dummy latency hexagonal array.



(c) compact parallel flow hexagonal array.

Figure 4.18: The performance comparisons of PE utilization rate of two-dimension systolic array topologies.

Together with the benefits of rapid process speed and fully concurrent operation, the proposed JSA algorithm also introduced the fan-out and wire delay factors. Table 4.8 describes the systolic array driver capability, or the fan-out information for various 2-D systolic array topologies. From these data the systolic array elements are required to drive more subsequent PEs to realize fully pipelined operation, such as four PE driver in square jumping array and two or three PE driver combination in compact parallel flow jumping hexagonal array. Table 4.9 also gives the different systolic array wire length information, where the PE distance equals to PE size as the unit for wire length. Traditional square and hexagonal systolic array have no PE detouring, but the proposed JSA algorithm requires additional connection detouring to enable the concurrent computation. Thus the wire length has been increased to keep the operation accuracy, commonly from two times to five times length extension as shown in Table 4.9.

From Table 4.5, throughput latency is used to measure the number of clock cycle count and compare the systolic array efficiency. In each systolic clock, the input data matrices are sent into PE array and the output results are sent out of PE array step by step. In the JSA algorithm, we use jumping wires and the length of which might be long compared to the conventional systolic arrays. So the clock period of JSA might be slightly larger than the conventional systolic arrays, but the difference might not be so large with the placement optimization, the buffer optimization and the pipeline architecture optimization. Consequently, for common LSI design, the JSA clock period or the systolic array performance can be compared and be determined mainly by the throughput latency and the average utilization rate as other systolic array research discussions [6, 90, 91, 93, 94].

For very high speed VLSI system, the increased fan-out complexity and wire length maybe introduce wire delay problem and cause signal transmission mismatch. The multi-drive PE needs to drive more input gate capacitance, which increases signal transmission time and decreases systolic operation efficiency. The extended wire length also causes the data delay of signal transmission among PE connection wires. If the wire delay caused by fan-out and wire length is too large, the data signal can not be sent to each PE input ports in time and the signal mismatch error maybe happen. Because systolic array requires that each input data and output data can be arranged with specific sequence and careful match, the input/output data mismatch maybe influence whole array operation and cause PE function failure. If the mismatch of fan-out and wire delay happens, the upper bound for clock frequency and wire delay can be evaluated to ensure the JSA operation function. Consider recent interconnection development, the multi-GHz sub-micron LSI SoC chip design needs more detailed discussions for wire delay and clock frequency. Thus our future work will focus on further adjustment and attachment to improve JSA architecture for practical SoC design. The additional buffer can be inserted in PE connection wire to increase the signal transmission ability and minimize the wire delay, as the clock delay minimal method [102]. Also, wire layout width can be adjusted to give broad signal transmission route and enable fast data signal transmission, such as EWA wire-sizing algorithm [103].

Table 4.8: Comparisons of systolic array driver capability for various two-dimension systolic array topologies.

Systolic driver capability	Drive one PE	Drive two PEs	Drive three PEs	Drive four PEs	Total PE number
Traditional square array	84	0	0	0	84
Square jumping array	52	0	0	8	84
Kung-Leiserson array	108	0	0	0	108
Redundant dummy latency	66	30	0	0	126
Compact parallel flow	64	16	4	0	108

Table 4.9: Comparisons of array wire length for various two-dimension systolic array topologies.

Array wire length	No PE detour	Detour one PE	Detour two PEs	Detour three PEs	Detour four PEs	Detour five PEs	Detour six PEs	Detour eight PEs	Detour ten PEs	Total wire length (unit: PE size)
Traditional square array	69	0	0	0	0	0	0	0	0	69
Square jumping array	45	24	0	0	0	0	0	0	0	101
Kung-Leiserson array	85	0	0	0	0	0	0	0	0	85
Redundant dummy latency	55	0	10	0	8	0	6	4	2	395
Compact parallel flow	18	12	26	2	6	2	0	0	0	290

4.3.4.3 Complexity Analysis

With the operation efficiency improvement, JSA architecture will raise PE driver requirements and increase wire length complexity. The system complexity characteristic analysis and discussion are necessary for high performance systolic array design. As illustrated in Table 4.8, the multi-drive PEs information are listed to show the introduced driver ability complexity, where several next PE driver capability is required and total drive PE number is increased for redundant dummy latency array. Also, the wire length complexity is given in Table 4.9. The additional PE detouring needs more wire length to satisfy the concurrent operation requirements, such as about double wire length for square jumping array and about five times wire length for redundant dummy latency array. More complex compact parallel flow can further decrease total wire length from 395 units to 290 units in Table 4.9 case. When the array size is enlarged, PE drive number and total wire length are increased together. Thus high complexity requirements are necessary to be considered for the proposed JSA architecture design.

Table 4.10: Comparisons of systolic connection complexity in the result output direction for various two-dimension array topologies.

Connection Complexity	1×0	1×1	2×0	2×1	2×2	3×1	3×2	3×3	4×1	4×4	5×5	Total Crossing
Traditional Square Array	0	0	0	0	0	0	0	0	0	0	0	0
Square Jumping Array	24	0	0	0	0	0	0	0	0	0	0	24
Kung-Leiserson Array	0	0	0	0	0	0	0	0	0	0	0	0
Redundant Dummy Latency	6	3	2	2	3	0	2	3	0	3	1	141
Compact Parallel Flow	2	6	4	2	1	4	0	3	2	1	0	87

The connection complexity is another systolic array complexity characteristic and influences the jumping systolic array efficiency together. The correlated experimental data from systolic array examples in this thesis are listed in Table 4.10, where the square or hexagonal array examples are also similar to Table 4.7, and “Total Crossing” means the total connection crossing point number. Also, the $m \times n$ in top table line of “Connection Complexity” means m lines intersect n lines. If $m \times 0$ or $0 \times n$, no input data crossing lines exist, but the crossing point number can be considered as $m \times 1$ or $1 \times n$ case. In the result output dimension the crossing lines are recorded and the intersection points are used to describe the connection complexity in the proposed jumping systolic arrays. The traditional square and hexagonal arrays have no crossing line problem, but the proposed JSA algorithm introduces the connection complexity (such as the systolic array examples of Figure 4.14, Figure 4.16, and Figure 4.17) as the compensation of highly concurrent operation and better systolic efficiency. Thus careful systolic array design and precise performance trade-off are important for the jumping array idea application.

As a result, the proposed jumping systolic architectures can improve the operation efficiency with the system compensation of driver capability, wire length and connection difficulty based on the complexity performance analysis. For two hexagonal array types, the redundant dummy latency architecture is regular and easy to be extended to large size array with the demerits of driver capability increase and connection line complexity as Table 4.8, Table 4.9 and Table 4.10 illustrations. However, another hexagonal array of compact parallel flow, can realize the system jumping operation by compact systolic array and small processing element area. But the expansion ability for compact parallel flow architecture is not easy, compared with redundant dummy latency architecture. Thus the two hexagonal arrays are suitable for different systolic application requirements.

4.4 Systolic Array Network Grid Synthesis Conclusion

Based on the traditional Kung-Leiserson BMMSA configuration, we proposed three new BMMSA designs in the wide-banded matrices case. Through the novel idea of "Super Pipeline", these new systolic arrays can realize higher average utilization rate and lower throughput latency by adjusting their operation sequences. Moreover, we also add the additional PEs to the "One-Space-Parallel BMMSA". Furthermore, the "One-Space-Jumping BMMSA" can realize the maximal average utilization rate (100%) and the minimal throughput latency ("$n + min(w_1, w_2)$") by data jumping. These new matrix multiplication designs also spend less processing time and become stable more rapidly than the traditional Kung-Leiserson BMMSA before the systems come into the full stream state. Finally these new BMMSA design ideas can also be introduced to other types of systolic structures.

The further research of large scale systolic array improvement in this thesis, is the high performance jumping systolic algorithm in two-dimension systolic array design for matrix multiplication operation. The systolic array research is focused on the processing speed increasement and PE utilization rate improvement under connection wire adjustment for various systolic topologies. Three new jumping systolic arrays, including square jumping array, redundant dummy latency jumping hexagonal array and compact parallel flow jumping hexagonal array, are proposed to realize high efficiency two-dimension systolic array design. These new systolic arrays are also based on the traditional wide-banded Kung-Leiserson architecture and send out computation results by fully concurrent operation. Though the JSA method increases PE driver requirements and causes connection complexity, the operation efficiency improvement is worthy for high performance systolic array design. The jumping systolic architectures also dominate other systolic arrays on the data band width, matrix adjustment complexity and array expansion capability. Future work of jumping systolic array research will be focused on other systolic array topology developments, or new systolic architecture applications of practical consumer electronics products, such as FFT, DCT, IDCT, FPGA, etc. Also, next step researches of buffer insertion, wire width and placement optimization will enable JSA algorithm to satisfy next era LSI SoC chip design with very high speed and ultra integration density.

Chapter 5

CONCLUSION

5.1 Power/Ground Network Grid Synthesis

As one of the core elements in the global interconnect network, the power/ground (P/G) network grid synthesis and optimization methods are discussed in this thesis. Based on the analysis of conventional synthesis algorithms and optimization methods, the design problems and research challenges are summarized to deal with the effective P/G network grid synthesis and global optimization design.

Several types of traditional SLP P/G network synthesis algorithms are discussed in this thesis. By the detailed analysis of the step splitting optimization and the nonlinear transformation, the linear-nonlinear conversion loss problem is proposed and new non-loss synthesis method is necessary to be developed for high performance network grid global optimization.

In the static P/G network grid optimization, which only considers the IR drop voltage fluctuation, the novel “Grid Synthesis using Genetic Algorithm (GGA)” is introduced to the high performance network grid synthesis. Also, for the P/G network with dynamic signal consideration, the gene disturbance can simulate the dynamic noise and construct the genetic evolution model. Based on the analogy of biology evolution period, the conventional genetic algorithm is modified to satisfy the large scale P/G network grid computation characteristics. By the effective data search and genetic coding allocation for the independent variables, various network grid parameters can be expressed by small scale ‘0’ and ‘1’ coding. Under the environment restraints of circuit equations and boundary conditions, the colony evolution is performed to get the environment suitable solution and realize the global convergence optimization for the P/G network grid. Because of the "Global Optimization" and "Target Insensitivity", the diverse independent variables can evolve themselves concurrently, and the specific target function is not considered deeply in the GGA computation method. For the improved HGGA method, the SLP endpoint searching capability can be combined with the dynamic genetic optimization and increase the evolution speed. As a result, the GGA and HGGA methods can avoid the nonlinear-linear transformation demerits, and realize the global optimization synthesis for static and dynamic signal consideration in the P/G network grid design.

5.2 Systolic Array Network Grid Synthesis

Systolic array architecture is widely used in recent VLSI system design and DSP chip implementation. By the benefits of global pipeline, concurrent operation and compact area, the systolic array also becomes one of the core processing elements to influence the system efficiency greatly. Traditional systolic array is restricted by throughput latency and utilization rate by the matrix computation requirements. Thus effective systolic array design is important for high performance VLSI system.

Instead of previous systolic array improvement methods of matrix modification, processing element variance, or other array change methods, this thesis is focused on the network grid connection among the processing element array. By the skillful network grid adjustments, the matrix data can be computed with fully pipeline and global concurrent. Thus the system operation efficiency can be improved rapidly and the global concurrent operation can be realized easily.

The basic network grid connection adjustment method is named as "Super Pipeline" in this thesis, which is suitable for small scale systolic array and the given processing element topology. Careful connection wire adjustment, peripheral element attachment, and computation sequence modification, enable the global pipeline with the compact systolic array layout area. The throughput latency is improved greatly and the utilization rate also increases to fully concurrent operation. The design requirements of connection complexity and driver ability are increased greatly as the trade-off problems. But the great performance improvement is worthy of these connection adjustment difficulties and processing element allocation complexity for better systolic array construction and effective global pipeline operation realization.

The improved connection adjustment method for systolic network grid is proposed as “Jumping Systolic Array (JSA)”, which is focused on the large scale systolic array and the general pipeline relationship. The small scale “Super Pipeline” method is difficult to be extended to large scale array size because of the unordered connection adjustment sequences. The proposed JSA method can solve the obstacle by keeping the output data sequence stable and just changing the input data sequences. Also, the JSA method emphasizes the array expansion ability, the operation stabilization speed, and the processing array area, in addition to the basic systolic array characteristics of throughput latency and utilization rate. More systolic topologies, such as square array and hexagonal array, are studied in this thesis to extend the JSA method application. Fully pipeline operation, compact PE layout area, and massive matrix computation, are the typical JSA algorithm merits. The connection wire complexity, crossing point existence, and large driver ability requirements are also the design trade-off. Consequently, fully pipeline and global concurrence of large scale systolic array are the core contribution for the proposed JSA method and can be used to improve the practical systolic array system performance greatly.

5.3 Network Grid Comprehensive Design

Common network grid synthesis considers the network design methods separately, such as the P/G network optimization and systolic array design. With the VLSI system development, fast processing speed and high operation efficiency are required for recent consumer electronics application. Moreover, the diverse network grid interaction is necessary to be considered deeply to realize the network grid synthesis and the global layout optimization.

The P/G network grid synthesis is focused on the network wire sizing under the given circuit relationship constraints. The connection width enlargement/reduction processing can influence the network synthesis efficiency. Also, other adjacent network can cause the signal instability and influence the circuit parameters. So the global network co-design method is necessary to be considered for effective network synthesis design. In addition, the circuit devices connected to the P/G network can introduce the additional current or other signal insertion. Then the global P/G network efficiency can also be influenced partly. As a result, the system synthesis consideration is important for the network grid optimization and SoC system implementation.

For the systolic array design, the network grid connection method is emphasized deeply for fully data pipeline and global concurrent operation. Despite of the connection adjustment improvement, the element characteristics are also important for the system pipeline operation, such as the driver capability, the wire detouring and the data flow direction. So the system co-design of network grid and processing element is the necessary synthesis approach for the effective VLSI chip design. Other network connection methods, including the crossing point allocation and the PAD connection method, are also necessary to be studied carefully, together with the systolic network adjustment. The general design consideration of global processing element and interconnection network grid will contribute to final network grid synthesis and high performance VLSI system implementation.

5.4 Future Work

The rapidly developed VLSI system requires further design improvement in grid synthesis and global optimization. The difficult network grid synthesis is recent design challenge and becomes the core part of high performance VLSI implementation. Many future works are necessary to enhance the system operation efficiency by the detailed considerations of P/G distribution network, systolic array topology, and other related network grid interaction.

The future research for P/G network grid will focus on the network co-design development as former section descriptions. Also, the recent nanometer level chip design requires multi-layer layout and complexity routing, which provide new challenges of three-dimension P/G network grid synthesis and multi-layer global sizing optimization. Other nanometer research expectation also focus on the reduction interconnect effects, such as quantum line effect for very narrow network wire situation, and DFM manufacture research based on nanometer design mismatch problems.

The next step research for systolic array network can also be extended to the three-dimension systolic topology development. The practical consumer electronics application is another research improvement direction. Additionally, many connection topologies, including triangular array, fractional array, and error diagnosis attachment array, are possible to be developed by using network grid adjustment and global data jumping proposed in this thesis.

The synthesis design of the network grid and other circuit elements is also necessary to be considered as future research challenges. Embedded processing system, software hardware co-design, periphery elements interaction, PAD interconnection, or other recent research topics, can also be used to expand the network grid synthesis range. In conclusion, we faces many synthesis research challenges now, and future network grid works are expectable to advance the next era VLSI system development.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Professor Shinji Kimura, for his helpful discussion and constructive instruction in my research and study. Also, I want to express my sincere acknowledgement to his support for my Doctor thesis research.

I show my thankfulness to Professor Yasuaki Inoue for his valuable advices and discussions for my paper advice and research instruction. Also, I studied many knowledge in analog circuit and chip design by his generous guidance.

I also give my acknowledgement to Professor Satoshi Goto for his financial support for my Doctor research period, to Professor Takeshi Yoshimura for his suggestion in my research topic, and to Professor Takahiro Watanabe for the assistance in my oversea life in JAPAN.

I want to thank the excellent research collaboration with Mr. Zhangcai Huang, Mr. Jun Pan, Ms. Hong Yu, Mr. Jia Guo, Mr. Yangxing Liu, Mr. Yang Song, Mr. Xingwen Xu, Ms. Jia Chen, Ms. Qian Tian, Ms. Taeko Matsunaga, Mr. Liangwei Ge, and Mr. Song Chen. Their discussions are fruitful and necessary for my research and study.

I also thank Mr. Chengjie Zang, Mr. Lei Chen, Mr. Weijie Xing, Mr. Chongyang Zhang, Mr. Qilun Song, Mr. Jiajing Liu, Mr. Jia Lei, Ms. Jingji Zheng, Ms. Mengru Wang, Ms. Yuan Gao, Ms. Xin Man, and Ms. Yinhua Li, for their kind help and nice advices in my daily life and laboratory study. Their friendly relationship gives me happy research environment and I enjoy all my friends in Waseda University in the past three and half years.

BIBLIOGRAPHY

[1] Ting-Yuan Wang and Charlie Chung-Ping Chen. Optimization of the power/ground network wire-sizing and spacing based on sequential network simplex algorithm. In *IEEE International Symposium on Quality Electronic Design (ISQED)*, pages 157–162, March 2002.

[2] Yun Yang, Atsushi Kurokawa, Yasuaki Inoue, and Wenqing Zhao. Efficient large scale integration power/ground network optimization based on grid genetic algorithm. *IEICE Trans. Fundamentals*, E88-A(12):3412–3420, December 2005.

[3] Yun Yang, Atsushi Kurokawa, and Yasuaki Inoue. The efficient grid genetic algorithm used in VLSI static power/ground network optimization. In *18th Workshop on Circuits and Systems in Karuizawa*, pages 37–42, April 2005.

[4] Yun Yang, Wenqing Zhao, and Yasuaki Inoue. High-performance systolic arrays for band matrix multiplication. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, volume 2, pages 1130–1133, May 2005.

[5] Yun Yang and Wenqing Zhao. Design of three high-performance concurrent systolic arrays for band matrix multiplication. *Chinese Journal of Electronics*, 14(4):559–563, October 2005.

[6] Shek-Wayne Chan and Chin-Long Wey. The design of concurrent error diagnosable systolic arrays for band matrix multiplications. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 7(1):21–37, January 1988.

[7] Sachin S. Sapatnekar and Haihua Su. Analysis and optimization of power grids. *IEEE Trans. Design & Test of Computers*, 20(3):7–15, May-June 2003.

[8] Haihua Su, Jiang Hu, Sachin S. Sapatnekar, and Sani R. Nassif. A methodology for the simultaneous design of supply and signal networks. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 23(12):1614–1624, December 2004.

[9] Jeff Parkhurst, Naveed Sherwani, Sury Maturi, Dana Ahrams, and Eli Chiprout. SRC physical design top ten problem. In *ACM International Symposium on Physical Design (ISPD)*, pages 55–58, April 1999.

[10] Harry Jordan and Gita Alaghband. *Fundamentals of Parallel Processing.* Prentice-Hall, 2003.

[11] Lawrence T. Pillage and Ronald A. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 9(4):352–366, April 1990.

[12] Altan Odabasioglu, Mustafa Celik, and Lawrence T. Pileggi. PRIMA: passive reduced-order interconnect macromodeling algorithm. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 17(8):645–654, August 1998.

[13] Yu-Min Lee, Yahong Cao, Tsung-Hao Chen, Janet Meiling Wang, and Charlie Chung-Ping Chen. HiPRIME: hierarchical and passivity preserved interconnect macromodeling engine for RLKC power delivery. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 24(6):797–806, June 2005.

[14] Min Zhao, Rajendran V. Panda, Sachin S. Sapatnekar, and David Blaauw. Hierarchical analysis of power distribution networks. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 21(2):159–168, February 2002.

[15] Joseph N. Kozhaya, Sani R. Nassif, and Farid N. Najm. A multigrid-like technique for power grid analysis. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 21(10):1148–1160, October 2002.

[16] Haifeng Qian, Sani R. Nassif, and Sachin S. Sapatnekar. Early-stage power grid analysis for uncertain working modes. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 24(5):676–682, May 2005.

[17] Haifeng Qian, Sani R. Nassif, and Sachin S. Sapatnekar. Power grid analysis using random walks. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 24(8):1204–1224, August 2005.

[18] Weikun Guo, Sheldon X.-D. Tan, Zuying Luo, and Xianlong Hong. Partial random walks for transient analysis of large power distribution networks. *IEICE Trans. Fundamentals*, E87-A(12):3265–3272, December 2004.

[19] Le Kang, Yici Cai, Jin Shi, and Xianlong Hong. In-depth experimental study of power grid network analysis using random walks algorithm. In *IEEE International Conference on Communications, Circuits and Systems (ICCCAS)*, volume 4, pages 2401–2405, June 2006.

[20] Le Kang, Yici Cai, Yi Zou, Jin Shi, Xianlong Hong, and Sheldon X.-D. Tan. Fast decoupling capacitor budgeting for power/ground network using random walk approach. In *Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 751–756, January 2007.

[21] Peng Li. Statistical sampling-based parametric analysis of power grids. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 25(12):2852–2867, December 2006.

[22] Salim U. Chowdhury and Melvin A. Breuer. Minimal area design of power/ground nets having graph topologies. *IEEE Trans. Circuits and Systems*, 34(12):1441–1451, December 1987.

[23] Salim U. Chowdhury. Optimum design of reliable IC power networks having general graph topologies. In *ACM/IEEE Design Automation Conference (DAC)*, pages 787–790, June 1989.

[24] Robi Dutta and Malgorzata Marek-Sadowska. Automatic sizing of Power/Ground (P/G) networks in VLSI. In *ACM/IEEE Design Automation Conference (DAC)*, pages 783–786, June 1989.

[25] Sheldon X.-D. Tan and C.-J. Richard Shi. Efficient very large scale integration power/ground network sizing based on equivalent circuit modeling. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 22(3):277–284, March 2003.

[26] Sheldon X.-D. Tan, C.-J. Richard Shi, and Jyh-Chwen Lee. Reliability-constrained area optimization of VLSI power/ground networks via sequence of linear programmings. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 22(12):1678–1684, December 2003.

[27] Xiaohai Wu, Xianlong Hong, Yici Cai, Zuying Luo, Chung-Kuan Cheng, Jun Gu, and Wayne Dai. Area minimization of power distribution network using efficient nonlinear programming techniques. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 23(7):1086–1094, July 2004.

[28] G. Bai, S. Bobba, and I. N. Hajj. Simulation and optimization of the power distribution network in VLSI circuits. In *ACM/IEEE International Conference on Computer Aided Design (ICCAD)*, pages 481–486, November 2000.

[29] Hang Li, Jeffrey Fan, Zhenyu Qi, Sheldon X.-D. Tan, Lifeng Wu, Yici Cai, and Xianlong Hong. Partitioning-based approach to fast on-chip decoupling

capacitor budgeting and minimization. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 25(11):2402–2412, November 2006.

[30] Jingjing Fu, Zuying Luo, Xianlong Hong, Yici Cai, Sheldon X.-D. Tan, and Zhu Pan. A fast decoupling capacitor budgeting algorithm for robust on-chip power delivery. *IEICE Trans. Fundamentals*, E87-A(12):3273–3280, December 2004.

[31] Andrew B. Kahng, Bao Liu, and Sheldon X.-D. Tan. Efficient decoupling capacitor planning via convex programming methods. In *ACM International Symposium on Physical Design (ISPD)*, pages 102–107, April 2006.

[32] Yici Cai, Jingjing Fu, Xianlong Hong, Sheldon X.-D. Tan, and Zuying Luo. Power/ground network optimization considering decap leakage currents. *IEEE Trans. Circuits and Systems II: Express Briefs*, 53(10):1012–1016, October 2006.

[33] Janet M. Wang and Tuyen V. Nguyen. Extended Krylov subspace method for reduced order analysis of linear circuits with multiple sources. In *ACM/IEEE Design Automation Conference (DAC)*, pages 247–252, June 2000.

[34] Jatan C. Shah, Ahmed A. Younis, Sachin S. Sapatnekar, and Marwan M. Hassoun. An algorithm for simulating power/ground networks using Padé approximants and its symbolic implementation. *IEEE Trans. Circuits and Systems II: Express Briefs*, 45(10):1372–1382, October 1998.

[35] Chao-Yang Yeh and Malgorzata Merek-Sadowska. Timing-aware power-noise reduction in placement. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 26(3):527–541, March 2007.

[36] Jin Shi, Yici Cai, Sheldon X.-D. Tan, Jeffrey Fan, and Xianlong Hong. Pattern-based iterative method for extreme large power/ground analysis. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 26(4):680–692, April 2007.

[37] Atsushi Muramatsu, Masanori Hashimoto, and Hidetoshi Onodera. Effects of on-chip inductance on power distribution grid. In *ACM International Symposium on Physical Design (ISPD)*, pages 63–69, April 2005.

[38] Nahi H. Abdul Ghani and Farid N. Najm. Handling inductance in early power grid verification. In *ACM/IEEE International Conference on Computer Aided Design (ICCAD)*, pages 127–134, November 2006.

[39] Shiyou Zhao, Kaushik Roy, and Cheng-Kok Koh. Decoupling capacitance allocation and its application to power-supply noise-aware floorplanning. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 21(1):81–92, January 2002.

[40] Jaskirat Singh and Sachin S. Sapatnekar. Congestion-aware topology optimization of structured power/ground networks. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 24(5):683–695, May 2005.

[41] Jaskirat Singh and Sachin S. Sapatnekar. Partition-based algorithm for power grid design using locality. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 25(4):664–677, April 2006.

[42] Kai Wang and Malgorzata Marek-Sadowska. On-chip power-supply network optimization using multigrid-based technique. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 24(3):407–417, March 2005.

[43] Yu-Min Lee and Charlie Chung-Ping Chen. Power grid transient simulation in linear time based on transmission-line-modeling alternating-direction-implicit method. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 21(11):1343–1352, November 2002.

[44] Wai-Ching Douglas Lam, Cheng-Kok Koh, and Chung-Wen Albert Tsao. Power supply noise suppression via clock skew scheduling. In *IEEE International Symposium on Quality Electronic Design (ISQED)*, pages 355–360, March 2002.

[45] Chen-Wei Liu and Yao-Wen Chang. Floorplan and power/ground network co-synthesis for fast design convergence. In *ACM International Symposium on Physical Design (ISPD)*, pages 86–93, April 2006.

[46] Salim U. Chowdhury and Melvin A. Breuer. Optimum design of IC power/ground nets subject to reliability constraints. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 7(7):787–796, July 1988.

[47] Howard. H. Chen and David D. Ling. Power supply noise analysis methodology for deep-submicron VLSI chip design. In *ACM/IEEE Design Automation Conference (DAC)*, pages 638–643, June 1997.

[48] Tsung Hao Chen and Charlie Chung Ping Chen. Efficient large-scale power grid analysis based on preconditioned krylov-subspace iterative methods. In *ACM/IEEE Design Automation Conference (DAC)*, pages 559–562, June 2001.

[49] Wan-Ping Lee, Hung-Yi Liu, and Yao-Wen Chang. Voltage island aware floorplanning for power and timing optimization. In *ACM/IEEE International Conference on Computer Aided Design (ICCAD)*, pages 389–394, November 2006.

[50] Hao Yu, Joanna Ho, and Lei He. Simultaneous power and thermal integrity driven via stapling in 3D ICs. In *ACM/IEEE International Conference on Computer Aided Design (ICCAD)*, pages 802–808, November 2006.

[51] Ting-Yuan Wang, Ieng-Liang Tsai, and Charlie Chung-Ping Chen. Power-delivery networks optimization with thermal reliability integrity. In *ACM International Symposium on Physical Design (ISPD)*, pages 124–131, April 2004.

[52] Sanjay Pant and David Blaauw. Static timing analysis considering power supply variations. In *ACM/IEEE International Conference on Computer Aided Design (ICCAD)*, pages 365–371, November 2005.

[53] Cheng Zhuo, Jiang Hu, Min Zhao, and Kangsheng Chen. Fast decap allocation based on algebraic multigrid. In *ACM/IEEE International Conference on Computer Aided Design (ICCAD)*, pages 107–111, November 2006.

[54] Sanjay Pant and Eli Chiprout. Power grid physics and implications for CAD. In *ACM/IEEE Design Automation Conference (DAC)*, pages 199–204, July 2006.

[55] Yu Zhong and Martin D. F. Wong. Fast algorithms for IR drop analysis in large power grid. In *ACM/IEEE International Conference on Computer Aided Design (ICCAD)*, pages 351–357, November 2005.

[56] Takayuki Watanabe, Yuichi Tanji, Hidemasa Kubota, and Hideki Asai. Parallel-distributed time-domain circuit simulation of power distribution networks with frequency-dependent parameters. In *Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 832–837, January 2006.

[57] Sarvesh H. Kulkarni and Dennis Sylvester. Power distribution techniques for dual VDD circuits. In *Asia and South Pacific Design Automation Conference (ASPDAC)*, pages 838–843, January 2006.

[58] Imad A. Ferzli and Farid N. Najm. Analysis and verification of power grids considering process-induced leakage-current variations. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 25(1):126–143, January 2006.

[59] H. T. Kung and C. E. Leiserson. *Algorithms for VLSI processor arrays.* Addison-Wesley, 1980.

[60] Harry Jordan and Gita Alaghband. *Fundamentals of parallel processing.* Prentice-Hall, 2003.

[61] Kuang-Hua Huang and Jacob A. Abraham. Algorithm-based fault tolerance for matrix operations. *IEEE Trans. Computers*, C-33(6):518–528, June 1984.

[62] Guo-Jie Li and Benjamin W. Wah. The design of optimal systolic arrays. *IEEE Trans. Computers*, C-34(1):66–77, January 1985.

[63] P. J. Varman and I. V. Ramakrishnan. Synthesis of an optimal family of matrix multiplication algorithms on linear arrays. *IEEE Trans. Computers*, C-35(11):989–996, November 1986.

[64] Sun-Yuan Kung, Sheng-Chun Lo, and Paul S. Lewis. Optimal systolic design for the transitive closure and the shortest path problems. *IEEE Trans. Computers*, C-36(5):603–614, May 1987.

[65] Henry Y. H. Chuang and Ling Chen. A general approach to solving arbitrarily large problems in a fixed size systolic array. In *21st Annual Hawaii Int. Conf. System Sciences, Kailua-Kona, HI, USA (HICSS)*, pages 195–204, January 1988.

[66] Kuojuey R. Liu and K. Yao. A systematic approach to bit recursive systolic array design. In *Int. Conf. Systolic Arrays, San Diego, CA, USA (ICSA)*, pages 685–694, May 1988.

[67] S. P. S. Lam. Matrix and vector multiplications using a $2\frac{1}{2}$–dimensional systolic array. *Electronics Letters*, 26(18):1453–1454, August 1990.

[68] Jong-Chuang Tsay and Sy Yuan. Some combinatorial aspects of parallel algorithm design for matrix multiplication. *IEEE Trans. Computers*, 41(3):355–361, March 1992.

[69] KuoJuey Ray Liu, Shih-Fu Hsieh, and Kung Yao. Systolic block householder transformation for RLS algorithm with two-level pipelined implementation. *IEEE Trans. Signal Processing*, 40(4):946–958, April 1992.

[70] H. V. Jagadish and T. Kailath. A family of new efficient arrays for matrix multiplication. *IEEE Trans. Computers*, 38(1):149–155, January 1989.

[71] Chein-Wei Jen and Ding-Ming Kwai. Data flow representation of iterative algorithms for systolic arrays. *IEEE Trans. Computers*, 41(3):351–355, March 1992.

[72] Malek G. M. Hussain and Mansour Jaragh. A triangular systolic array for the discrete-time deconvolution. *IEEE Trans. Circuits and Systems*, 36(4):622–628, April 1989.

[73] J. J. Wang and C. W. Jen. Redundancy design for a fault tolerant systolic array. *IEE Proc. Computers and Digital Techniques*, 137(3):218–226, May 1990.

[74] Xiaqi Liu and Howard Fan. Schur type algorithms for spatial LS estimation with highly pipelined architectures. *IEEE Trans. Signal Processing*, 41(10):3010–3023, October 1993.

[75] Hyesook Lim and Jr. Earl E. Swartzlander. Multidimensional systolic arrays for the implementation of discrete fourier transforms. *IEEE Trans. Signal Processing*, 47(5):1359–1370, May 1999.

[76] Che-Wun Chiou, Chiou-Yng Lee, An-Wen Deng, and Jim-Min Lin. Concurrent error detection in Montgomery multiplication over $GF(2^m)$. *IEICE Trans. Fundamentals*, E89-A(2):566–574, February 2006.

[77] Jyh-Huei Guo and Chin-Liang Wang. Systolic array implementation of euclid's algorithm for inversion and division in $GF(2^m)$. *IEEE Trans. Computers*, 47(10):1161–1167, October 1998.

[78] Chang Hoon Kim, Chun Pyo Hong, and Soonhak Kwon. A digit-serial multiplier for finite field $GF(2^m)$. *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, 13(4):476–483, April 2005.

[79] A.K. Oudjida, S. Titri, and M. Hamarlain. N latency 2N I/O-bandwidth 2D-array matrix multiplication algorithm. In *8th IEEE Int. Conf. Electronics, Circuits and Systems, Malta (ICECS)*, pages 597–600, September 2001.

[80] S. Belkacemi, K. Benkrid, D. Crookes, and A. Benkrid. Design and implementation of a high performance matrix multiplier core for Xilinx Virtex FPGAs. In *6th Int. Workshop Comput. Archi. Machine Perception, New Orleans, LA, USA (CAMP)*, pages 156–159, May 2003.

[81] Man Guo, M. Omair Ahmad, M. N. S. Swamy, and Chunyan Wang. FPGA design and implementation of a low-power systolic array-based adaptive Viterbi decoder. *IEEE Trans. Circuits and Systems I: Regular Papers*, 52(2):350–365, February 2005.

[82] Doru Florin Chiper, M. N. S. Swamy, M. Omair Ahmad, and Thanos Stouraitis. Systolic algorithms and a memory-based design approach for a unified architecture for the computation of DCT/DST/IDCT/IDST. *IEEE Trans. Circuits and Systems I: Regular Papers*, 52(6):1125–1137, June 2005.

[83] Chung-Chin Lu, Jau-Yuan Hsu, and Chih-Chung Cheng. Systolic array implementation of a real-time symbol-optimum multiuser detection algorithm. *IEEE Trans. Communications*, 53(10):1718–1728, October 2005.

[84] Pol Lin Tai, Chii Tung Liu, and Jia Shung Wang. A unified systolic array design for kernel functions of video compression. *IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing*, 48(5):523–531, May 2001.

[85] Pramod Kumar Meher. Design of a fully-pipelined systolic array for flexible transposition-free VLSI of 2-D DFT. *IEEE Trans. Circuits and Systems II: Express Briefs*, 52(2):85–89, February 2005.

[86] Nadia Nedjah and Luiza de Macedo Mourelle. Three hardware architectures for the binary modular exponentiation: sequential, parallel, and systolic. *IEEE Trans. Circuits and Systems I: Regular Papers*, 53(3):627–633, March 2006.

[87] Louis P. Rubinfield. A proof of the modified booth's algorithm for multiplication. *IEEE Trans. Computers*, C-24(10):1014–1015, October 1975.

[88] Masato Nagamatsu, Shigeru Tanaka, Junji Mori, Katsusi Hirano, Tatsuo Noguchi, and Kazuhisa Hatanaka. A 15-ns 32×32-b CMOS multiplier with an improved parallel structure. *IEEE Trans. Solid-State Circuits*, 25(2):494–497, April 1990.

[89] Fang Lu and Henry Samueli. A 200 MHz CMOS pipelined multiplier-accumulator using a quasi-domino dynamic full-adder cell design. *IEEE Trans. Solid-State Circuits*, 28(2):123–132, February 1993.

[90] H. F. Li, C. N. Zhang, and R. Jayakumar. Space-time mapping, latency of data flow and concurrent error detection in systolic arrays. *IEE Proc. Computers and Digital Techniques*, 140(1):33–44, January 1993.

[91] C. N. Zhang, T. M. Bachtiar, and W. K. Chou. Optimal fault-tolerant design approach for VLSI array processors. *IEE Proc. Computers and Digital Techniques*, 144(1):15–21, January 1997.

[92] M. O. Esonu, A. J. Al-Khalili, S. Hariri, and D. Al-Khalili. Fault-tolerant design methodology for systolic array architectures. *IEE Proc. Computers and Digital Techniques*, 141(1):17–28, January 1994.

[93] I. Z. Milovanovic, E. I. Milovanovic, I. Z. Milentijevic, and M. K. Stojcev. Designing of processor-time optimal systolic arrays for band matrix-vector multiplication. *Computers Math. Applic.*, 32(2):21–31, February 1996.

[94] I. Z. Milentijevic, I. Z. Milovanovic, E. I. Milovanovic, and M. K. Stojcev. The design of optimal planar systolic arrays for matrix multiplication. *Computers Math. Applic.*, 33(6):17–35, June 1997.

[95] I. Z. Milentijevic, I. Z. Milovanovic, E. I. Milovanovic, and M. K. Stojcev. Designing hexagonal systolic array by composite mappings. *Facta Universitatis (Nis), Ser. Math. Inform.*, (12):283–296, December 1997.

[96] Jong-Chuang Tsay and Pen-Yuang Chang. Design of efficient regular arrays for matrix multiplication by two-step regularization. *IEEE Trans. Parallel and Distributed Systems*, 6(2):215–222, February 1995.

[97] Jong-Chuang Tsay and Pen-Yuang Chang. Some new designs of 2-D array for matrix multiplication and transitive closure. *IEEE Trans. Parallel and Distributed Systems*, 6(4):351–362, April 1995.

[98] John Henry Holland. *Adaption in Natural and Artificial Systems.* Univ Michigan Press, 1975.

[99] Richard L. Burden and J. Douglas Faires. *Numerical Analysis (Seven Edition).* Brooks Cole, Pacific Grove, CA, 2000.

[100] Xiaolin Hu, Zhangcan Huang, and Zhongfan Wang. Hybridization of the multi-objective evolutionary algorithms and the gradient-based algorithms. In *IEEE Congress on Evolutionary Computation (CEC)*, pages 870–877, December 2003.

[101] Semiconductor industry association. *International Technology Roadmap for Semiconductors*, 2001. [Online]. Available: http://public.itrs.net.

[102] Xuan Zeng, Dian Zhou, and Wei Li. Buffer insertion for clock delay and skew minimization. In *ACM International Symposium on Physical Design (ISPD)*, pages 36–41, April 1999.

[103] Rony Kay and Lawrence T. Pileggi. Ewa: efficient wiring-sizing algorithm for signal nets and clock nets. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 17(1):40–49, January 1998.

PUBLICATIONS

Journal Paper

1. **Yun Yang** and Shinji Kimura, "The optimal architecture design of two-dimension matrix multiplication jumping systolic array," *IEICE Trans. on Fundamentals (IEICE-A)*, 11 pages, vol.E91-A, no.4, Apr. 2008. (to be published)

2. **Yun Yang**, Atsushi Kurokawa, Yasuaki Inoue, and Wenqing Zhao, "Efficient large scale integration power/ground network optimization based on grid genetic algorithm," *IEICE Trans. on Fundamentals (IEICE-A)*, vol.E88-A, no.12, pp.3412-3420, Dec. 2005.

3. **Yun Yang** and Wenqing Zhao, "Design of three high-performance concurrent systolic arrays for band matrix multiplication," *Chinese Journal of Electronics (CJE)*, vol.14, no.4, pp.559-563, Oct. 2005.

4. Zhangcai Huang, Yasuaki Inoue, Hong Yu, Jun Pan, **Yun Yang**, Quan Zhang, and Shuai Fang, "Behavioral circuit macromodeling and analog LSI implementation for automobile engine intake system," *IEICE Trans. on Fundamentals (IEICE-A)*, vol.E90-A, no.4, pp.732-740, Apr. 2007.

5. Atsushi Kurokawa, Akira Kasebe, Toshiki Kanamoto, **Yun Yang**, Zhangcai Huang, Yasuaki Inoue, and Hiroo Masuda, "Formula-based method for capacitance extraction of interconnects with dummy fills," *IEICE Trans. on Fundamentals (IEICE-A)*, vol.E89-A, no.4, pp.847-855, Apr. 2006.

6. Zhangcai Huang, Atsushi Kurokawa, **Yun Yang**, Hong Yu, and Yasuaki Inoue, "Modeling the influence of input-to-output coupling capacitance on CMOS inverter delay," *IEICE Trans. on Fundamentals (IEICE-A)*, vol.E89-A, no.4, pp.840-846, Apr. 2006.

7. Atsushi Kurokawa, Masanori Hashimoto, Akira Kasebe, Zhangcai Huang, **Yun Yang**, Yasuaki Inoue, Ryosuke Inagaki, and Hiroo Masuda, "Second-order polynomial expressions for on-chip interconnect capacitance," *IEICE Trans. on Fundamentals (IEICE-A)*, vol.E88-A, no.12, pp.3453-3462, Dec. 2005.

International Conference

8. **Yun Yang**, Jia Guo, Yasuaki Inoue, and Hong Yu, "A large current and high speed laser diode driver using switch position modification," *International Conference on Communications, Circuits and Systems (ICCCAS)*, Guilin, China, pp.2319-2323, June 2006.

9. **Yun Yang**, Wenqing Zhao, and Yasuaki Inoue, "High-performance systolic arrays for band matrix multiplication," *IEEE International Symposium on Circuits and Systems (ISCAS)*, Kobe, Japan, pp.1130-1133, May 2005. ***(2005 Excellent Student Award of the IEEE Fukuoka Section)***

10. Hong Yu, Yasuaki Inoue, Xiaochuan Hu, Kazutoshi Sako, and **Yun Yang**, "An effective implementation of the compound element pseudo-transient algorithm on SPICE3," *IEEJ International Analog VLSI Workshop (IEEJ-2006)*, Hangzhou, China, Aug. 2006.

11. **Yun Yang** and Shinji Kimura, "Efficient hybrid genetic grid algorithm for power/ground network synthesis with dynamic signal consideration," *DAC 2008*, June 2008. (Under review)

Domestic Workshop

12. **Yun Yang** and Shinji Kimura, "Optimal planar jumping systolic array design for matrix multiplication," *The 20th Workshop on Circuits and Systems in Karuizawa (KARUIZAWA-2007)*, Karuizawa, Japan, pp.343-348, Apr. 2007.

13. **Yun Yang**, Jia Guo, and Yasuaki Inoue, "An effective large current and high speed laser diode driver circuit design," *The 19th Workshop on Circuits and Systems in Karuizawa (KARUIZAWA-2006)*, Karuizawa, Japan, pp.383-386, Apr. 2006.

14. **Yun Yang**, Atsushi Kurokawa, and Yasuaki Inoue, "The efficient grid genetic algorithm used in VLSI static power/ground network optimization," *The 18th Workshop on Circuits and Systems in Karuizawa (KARUIZAWA-2005)*, Karuizawa, Japan, pp.37-42, Apr. 2005.

15. Jia Guo, **Yun Yang**, and Yasuaki Inoue, "A study on the super-high-speed CMOS large current driver circuit," *The 2005 IEEJ Electronics, Information and Systems Society Workshop (IEEJ-SECTION-C)*, Kitakyushu, Japan, pp.735-736, Sep. 2005. (in Japanese)

16. Atsushi Kurokawa, Akira Kasebe, Toshiki Kanamoto, **Yun Yang**, Zhangcai Huang, Yasuaki Inoue, and Hiroo Masuda, "Analytical formula-based method for capacitance extraction of interconnects with dummy fills," *The 18th Workshop on Circuits and Systems in Karuizawa (KARUIZAWA-2005)*, Karuizawa, Japan, pp.19-24, Apr. 2005. (in Japanese)

Book & Thesis

17. **Yun Yang** and Shinji Kimura, "Effective Network Grid Synthesis and Optimization for High Performance Very Large Scale Integration System Design," *Doctor Thesis, Graduate School of Information, Production and Systems, Waseda University*, Kitakyushu, Japan, March 2008.

18. **Yun Yang**, Wenqing Zhao, Pushan Tang, and Xuan Zeng, "The GHz global interconnect network synthesis research and development," *Master Thesis, Department of Micro-Electronics Engineering, Fudan University*, Shanghai, China, June 2004. (in Chinese)

19. **Yun Yang**, Xinping Qu, and Binzhong Li, "The YBTO superconducting film design and Josephson effect research," *Bachelor Thesis, Department of Electronics Engineering, Fudan University*, Shanghai, China, June 1998. (in Chinese)

Patent

20. **Yun Yang** and Wenqing Zhao, "A new two-space-parallel high-performance systolic arrays for band matrix multiplication," *China Patent 200410016350.3*, 2005. (pending)

21. **Yun Yang** and Wenqing Zhao, "A new one-space-parallel high-performance systolic arrays for band matrix multiplication," *China Patent 200410016352.2*, 2005. (pending)

22. **Yun Yang** and Wenqing Zhao, "A new one-space-jumping high-performance systolic arrays for band matrix multiplication," *China Patent 200410016353.7*, 2005. (pending)

Award

23. **2005 Excellent Student Award of the IEEE Fukuoka Section** for *(High-Performance Systolic Arrays for Band Matrix Multiplication, in Proc. of 2005 IEEE International Symposium on Circuits and Systems, Vol.2, pp.1130-1133.)*