A Combinatorial Switch Block Design Technique for Reconfigurable Interconnection Networks

by

Hongbing Fan

Ph.D, Shandong University, 1990

A Dissertation Submitted in Partial Fullfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in the Department of Department of Computer Science

We accept this dissertation as conforming
to the required standard

Dr. John Ellis, Supervisor (Department of Computer Science)

Dr. Jon Muzio, Departmental Member (Department of Computer Science)

Dr. Wendy Myrvold, Departmental Member (Department of Computer Science)

Dr. Fayez Gebali, Outside Member
(Department of Electrical and Computer Engineering)

Dr. Steve Wilton, External Examiner (University of British Columbia, Canada)

Abstract

Switch blocks are fundamental switch components in reconfigurable interconnection networks such as FPGA routing networks and circuit switching communication networks. A switch block consists of a set of terminals usually partitioned into some sides, and programmable switches joining terminals on different sides, it is used to implement various routing requirements through reconfiguring its switches. This dissertation presents a combinatorial design technique to tackle the connection topology design problem for a wide assortment of switch blocks including hyper-universal switch blocks and universal switch blocks. We prove that large switch blocks can be constructed by a finite number of prime switch blocks, and show how to design the prime switch blocks and how to combine them into large switch blocks. We demonstrate the design scheme by examining and designing in detail the commonly used 4-sided switch blocks, as well as the generic universal switch blocks.

Examiners:

Dr. John Ellis, Supervisor (Department of Computer Science)

Dr. Jon Muzio, Departmental Member (Department of Computer Science)

Dr. Wendy Myrvold, Departmental Member (Department of Computer Science)

Dr. Fayez Gebali, Outside Member
(Department of Electrical and Computer Engineering)

Dr. Steve Wilton, External Examiner (University of British Columbia, Canada)

Contents

List of Figures

List of Tables

Acknowledgements

First I would like to thank my supervisor J. Ellis for his continuous encouragement, patient guidance, and support in the course of my graduate study.

The work in this dissertation could not have been done without the collaboration and support from W. L. Wu at the Chinese University of Hong Kong and J. Liu at the University of Lethbridge. I thank them for their constructive collaboration and friendship.

I am indebted to J. Muzio and W. Myrvold for their crucial advice on my dissertation and valuable revision comments. I thank M. Fellows for his guidance on parameterized complexity theory, and J. Shallit for bring linear Diophantine equations to my attention.

Thanks also go to the University of Victoria, the Faculty of Graduate Studies, and the Department of Computer Science for providing the financial support.

Finally, my special thank goes to my wife and daughter for their understanding and support.

Dedication

I proudly dedicate this dissertation
to my parents.
I feel their understanding and support wherever I am.

Chapter 1

Introduction

Reconfigurable interconnection networks are the underlying hardware infrastructures of many electronic devices. On a small scale, the programmable logic devices such as *Field Programmable Gate Arrays* (FPGA's) are based on routing networks which implement requested connections among functional blocks/components through reconfiguring programmable switches [12, 9]. On a larger scale, circuit switching communication networks, such as traditional telephone systems, utilize programmable switching networks to connect different user terminals upon request [47].

Despite the differences in scale and technology, all reconfigurable interconnection networks share some features. First of all, a reconfigurable interconnection network basically consists of programmable switches and wire segments (or channels). Secondly, the function of a reconfigurable interconnection network is to implement requested connections (or routings) by configuring the programmable switches. And thirdly, a complex reconfigurable interconnection network is usually built up from simple switch modules (or switch components, or switch blocks), which are joined together by wire segments to form a certain network.

Switch modules are important reconfigurable/programmable components in reconfigurable interconnection networks. They affect not only the routing capability of reconfigurable interconnection networks, but also the area, time, and power efficiency. The simplest and most flexible switch modules

are *switch blocks*. A switch block consists of a set of terminals, usually partitioned into several *sides* (i.e., groups), and a set of programmable switches, each switch joining a pair of terminals on different sides. The commonly used crossbar can be seen as 2-sided switch blocks with input terminals on one side and output terminals on the other.

Switch block design has many levels from the high level *topological design*, i.e., determining the connection relations of terminals and switches, to the physical level of switch circuit and layout design. The topological design of switch blocks is important because it determines the *routing capability*, i.e., the ability of realizing a given connection requirement, and it affects the area and time efficiency of a reconfigurable interconnection network.

This dissertation focuses on the topological design of switch blocks. The main problem is to design a switch block with a given number of sides and a given number of terminals on each side and a given routing capability and with a minimum number of switches. We use a set of *routing requirements* to describe the *routing capability* of a switch block. A routing requirement for a k-sided switch block with d_i terminals on side i, where $i = 1, \ldots, k$, is a set of nets. A *net* or a *l-pin net* ($l \leq k$) is a connection request for l terminals on l different specified sides. The number of nets that specify side i in a routing requirement is at most $d_i, i = 1, \ldots, k$. A *feasible routing* (or *detailed routing*) of a routing requirement in the switch block is an assignment ON/OFF to switches such that all nets in the routing requirement are routed simultaneously. That is, every l-pin net is realized by a set of ON switches which connect t terminals on the t different specified sides, and no others, and a terminal is used by at most one net realization. A switch block is *routable* for a routing requirement if a feasible routing exists

in the switch block. A switch block is said to be *universal* if it routable for every routing requirement consisting of 2-pin nets. The universal switch block design problem is to design a universal switch block of k sides and w terminals on each side and with a minimum number of switches [14, 46, 26]. Similarly, a switch block is said to be *hyper-universal* if it is routable for every routing requirement consisting of multi-pin nets (i.e., all nets are allowed). The hyper-universal switch block design problem first was studied in [22, 23].

The main purpose of this dissertation is to establish a combinatorial design technique to tackle the topological design problem for a wide assortment of switch blocks. We transfer the switch block design problem to a graph design problem by viewing a k-sided switch block as a k-partite graph, i.e., a terminal as a vertex, a switch as an edge, and a side as a part. We give precise combinatorial definitions for the switch blocks, the routing requirements and the feasible routings, and develop a switch block design scheme, as well as use the design scheme to design some particularly interested switch blocks.

1.1 Motivation

Our study of switch blocks was originally motivated by an attempt to design better 4-sided switch blocks for FPGA routing architectures [42, 14, 37].

An FPGA consists of an array of logic blocks and a routing network. After fabrication, the routing network can be reconfigured to make various connections of the logic blocks via reconfiguring the programmable switches in the routing network. Since FPGA was first introduced in 1985 [9], the research and development on FPGA have been growing at a tremen-

dous speed with the advancement of *Very Large Scale Integration* (VLSI) technology More and more powerful FPGA products have come onto the market. For instance, the Xilinx Vertex-II Pro platform FPGA released in 2002, which is based on IBM's advanced 130nm and 9-layer copper process, contains up to 50,832 logic cells, 3,888K block RAM, four IBM PowerPC 405 processors, and sixteen multi-gigabit I/O transceivers [62]. Such an FPGA is capable of implementing a wide range of high performance system applications, including optical networking, wireless infrastructure, storage systems, industrial control, etc. Using FPGA's has become a risk-free, fast turn-around, and economical solution for many integrated circuit designs. There is no doubt that the development of more advanced FPGA's will continue.

The routing network or the reconfigurable interconnection network plays an important role in an FPGA, it is the infrastructure which realizes the different connections of functional blocks according to the routing requests made by applications. As with FPGA's, reconfigurable interconnection networks can also be used in reconfigurable *Systems-on-a-Chip* to implement different connections of components for different functions. The interconnection networks of circuit switching communication systems can also be viewed as reconfigurable interconnection networks [47].

Many issues are involved in the design of an on-chip reconfigurable interconnection network. The first is *routability*, which measures the ability to realize a given connection requirement. Routability depends on both the connection topology of a reconfigurable interconnection network and the routing algorithms, which are expected to make efficient use of the existing routing resources, i.e., switches and wire segments[12, 9].

The second issue is time efficiency. Despite many advantages, the speed of an FPGA implementation is usually slower than that of an application-specific integrated circuit design. The slower speed is largely due to the programmable switches used to make connections. Because they add resistance and capacitance to the connections, the time to reach the voltage threshold of gates is increased and clock frequency is decreased.

The third issue is area efficiency. Switches in the routing network take a large chip area [9, 33]. Reducing the number of switches is aimed at achieving the area efficiency. The fourth issue is power consumption. An FPGA implementation usually uses more power than an application-specific integrated circuit design. Reducing the power consumption is more important with mobile applications.

All these issues are hard when a routing networks gets large and complex because they interact and some are in conflict. Also these issues are involved at different levels of the design process, i.e., topological design, switch circuit design and layout design.

Switch blocks are the fundamental switch components in a routing network. They affect the performance of the routing network with respect to routability, area, time, and power efficiencies [12, 9, 14, 46, 51, 34]. Designing efficient switch blocks is fundamental and crucial in the design of routing networks. The basic design strategy is to optimize the design of switch blocks, i.e., to minimize the number of switches consistent with a certain routing capability. Although there is no theoretical proof that the use of optimal switch blocks results in a high global (chip level) routability, extensive experimental results have shown that switch blocks of high routing capability are good choices most of the time [42, 9, 14, 51, 24].

Switch blocks can also be used in the design of high performance circuit switching communication networks. The traditional circuit switching networks use crossbars as basic switch modules [47, 16, 4]. However, switch blocks provide more flexible and highly routable switch modules. For example, Yen et al.[64] proposed a polygonal switching network, which is a three stage switch module with a universal switch block [14, 46] at the center and a full crossbar on each side. Such a polygonal switching network is rearrangeable, i.e., it can realize all possible point-to-point connection requests. Moreover, if a polygonal switching network uses a hyper-universal switch block at the center, it will be rearrangeable for all possible group connection requests.

Switch blocks also have a potential application in the design of on-chip reconfigurable interconnection networks for reconfigurable Systems-on-a-Chip. Designing a reconfigurable routing network to implement different connections among components on a chip is one technique for designing reconfigurable Systems-on-a-Chip. However, in this application, generic switch blocks are necessary with respect to various types of connection requests.

Because of the existing and potential applications of switch blocks, it is important to have a systematic design method for designing a switch block with respect to any given number of sides and numbers of terminals on all sides and routing capability.

1.2 The Goals of This Dissertation

This dissertation has four goals. The first goal is to examine and classify switch block design problems for various types of switch blocks. We describe

the problem in general as follows. Given a *dimension specification*, i.e., the number of sides, and a *channel capacity* specification[1], i.e., the number of terminals on every side, and a *routing capability* specification, i.e., a set of routing requirements, the problem is to design a switch block satisfying (1) it has the specified dimension and channel capacities, (2) it is routable for every routing requirement specified by the routing capability specification, (3) it has the minimum number of switches satisfying (1) and (2), and in addition (4) it has a small flexibility, i.e., the maximum number of switches incident with a switch, and (5) a feasible routing in the switch block can be found efficiently. A switch block satisfying (1), (2) and (3) is said to be *optimal* (or an *optimal switch block*).

We classify switch blocks and switch block design problems according to the specifications. For convenience, we write (k, w)-*SB* for a k-sided switch block with w terminals on each side, and $(d_1, \ldots, d_k)^T$-*SB* for a k-sided switch block with d_i terminals on side $i, i = 1, \ldots, k$. The notions of universal [14] and hyper-universal [22] have been used previously. A universal (hyper-universal) switch block is one routable for very routing requirements consisting of 2-pin (multi-pin) nets. The universal (hyper-universal) switch block design problem studied previously is to design an optimal universal (hyper-universal) (k, w)-SB for every pair of k and w.

In this dissertation, we introduce the concept of a *P-universal switch block*, which means a switch block routable for every routing requirement consisting of nets from a allowed net pattern set P. A P-universal $(d_1, \ldots, d_k)^T$-SB design problem will be addressed. This is a more general switch block design problem, which covers the universal switch block design problem, as

[1]The number of terminals on a side is also called channel density, or channel width

well as the hyper-universal switch block design problem.

The second goal is to integrate the previous work on switch block designs [22, 24, 26, 23] and to establish a systematic design scheme to tackle various and more general switch block design problems. A decomposition design technique[2] for hyper-universal (k, w)-SB's was first proposed in [22]. We have proved a decomposition theorem, which says that, when k is fixed, a hyper-universal (k, w)-SB can be constructed by a disjoint union of some copies of a hyper-universal (k, p)-SB and one hyper-universal (k, r)-SB, where p is a function of k and there is a finite number of options for r. This implies that when designing hyper-universal (k, w)-SB's for every $w \geq 1$, we can first design hyper-universal (k, r)-SB's for some small integers r, called *prime* hyper-universal (k, w)-SB's. Then we use them to build *compound* hyper-universal (k, w)-SB's for all other w's by the disjoint union operation. This design scheme is called a *decomposition design scheme.* The advantages of the compound hyper-universal (k, w)-SB's are (1) the number of switches is linear in w, (2) they are nearly optimal as w grows, and (3) a feasible routing can be found efficiently. This decomposition design scheme was also used in designing universal (k, w)-SB's in [27, 26]. In this dissertation, we aim to generalize the decomposition design technique for a more general class of switch blocks. We achieve this goal by developing a decomposition design scheme for P-universal k-sided switch blocks with the number of terminals on each side is determined by a vector $(dw + c)$, where d and c are k-dimensional nonnegative integer vectors, and w is a scale variable.

[2]It is called reduction design scheme previous. We here call it decomposition design scheme

The third goal is to give a detailed proof of hyper-universality for a recently designed prime hyper-universal $(4, w)$-SB's, and to make a comparison with previous $(4, w)$-SB designs. The new design has been published in [24], but a detailed proof for hyper-universality has not yet been published. We use routing capability, i.e., the sets of routing requirements routable, to compare our new hyper-universal $(4, w)$-SB's with three previous $(4, w)$-SB designs, i.e., disjoint switch blocks [61], universal switch blocks [14], and Wilton's switch blocks [51]. We show that the hyper-universal $(4, w)$-SB's have the highest routing capability. The experiments with the hyper-universal switch blocks were done by Y. L. Wu and C. C. Cheung [24]. But we do not examine experimental issues in this dissertation.

The fourth goal is to give a detailed description for our universal (k, w)-SB designs, including a complete proof of a new decomposition theorem. The universal (k, w)-SB's for $k \geq 5$ design was studied by Shyu et al. [46]. We found bugs in their results. A short note with partial corrections was made in [27]. In [26], using the decomposition design technique we formally proved that the given (k, w)-SB's designs were optimal universal (k, w)-SB's for $k \leq 6$, or $k \geq 7$ and even w, and gave approximate universal (k, w)-SB's for other values of k and w. In designing prime universal (k, w)-SB's, an important problem is to determine the channel capacities of the prime universal switch blocks. It was shown in [26] that the channel capacities of prime universal (k, w)-SBs are $w = 1, 2, 3, \ldots, 2i + 1, \ldots, f_2(k)$. The function $f_2(k)$ is equal to the maximum degree of a non-decomposable regular graphs of k vertices. In [26] it was proved that $\frac{k+3-i}{3} \leq f_2(k) \leq k(k-1)/2$, and it was conjectured in that $f_2(k) = \frac{k+3-i}{3}$ where $1 \leq i \leq 6, k \geq 7$ and $i \equiv k \pmod 6$. In this dissertation, we give a proof of this conjecture and

its impact on universal switch block designs.

1.3 Contributions

The main contributions of this dissertation are:

1. combinatorial modelling for a more general class of switch blocks and switch block design problems,
2. more flexible routing capability specifications,
3. the enumeration of routing requirements by linear Diophantine equations,
4. more general decomposition theorems for routing requirements and switch block designs,
5. a decomposition design scheme for a wide assortment of switch blocks,
6. design and verification of hyper-universal $(4, w)$-SB's, and
7. a new decomposition theorem for universal (k, w)-SB's.

More specific details are as follows.

- We describe a combinatorial model for a more general class of switch block design problems. All the switch blocks studied previously are *regular switch blocks*, i.e., having the same number of terminals on all sides. We consider irregular switch blocks specified by $(d_1, \ldots, d_k)^T$-SB, and in particular, the $(dw+c)$-SB's where d and c are two given k-dimensional nonnegative integer vectors and $w \geq 1$ is an integer scale

variable. The $(dw + c)$-SB's can be used to design a $(d_1, \ldots, d_k)^T$-SB by choosing d, c and w such that $dw + c = (d_1, \ldots, d_k)^T$.

Three aspects of switch blocks are modelled: the switch blocks themselves, routing requirements, and feasible routings. It is natural to represent a $(d_1, \ldots, d_k)^T$-SB by a *k-partite graph* with terminals as vertices and switches as edges, i.e., a graph with vertex set partitioned into k parts labelled $1, 2, \ldots, k$, part i has d_i vertices, and each edge joining a pair of vertices in different parts.

We represent a l-pin net by a subset $N = \{i_1, \ldots, i_l\} \subseteq \{1, 2, \ldots, k\}$, and a routing requirement for a $(d_1, \ldots, d_k)^T$-SB by a multi-set of subsets $\{N_1, \ldots, N_t\}$ satisfying $N_i \subseteq \{1, 2, \ldots, k\}, i = 1, \ldots, t$ and

$$\sum_{j=1}^{t} |N_j \cap \{i\}| \leq d_i, i = 1, \ldots, k.$$

A feasible routing of $\{N_1, \ldots, N_t\}$ is a subgraph consisting of l components $T_1, \ldots, T_t$ such that each T_j is a tree of $|N_j|$ vertices and T_j has a vertex in part i if and only if $i \in N_j$. It is clear that when the switches corresponding to the edges of a tree are turned on, then the terminals corresponding to the vertices of the tree are connected. Using the above model, we reduce a switch block design problem to a graph design problem.

- As routing capability terminology, "universal" and "hyper-universal" have been used previously. In this dissertation, we propose a more flexible routing capability specification determined by a given set of subsets, called a *net pattern set*. Let P be a net pattern set, a routing requirement with nets choosing from P is called a *P-net routing requirement*. A switch block is said to be *P-universal* if it is routable for

every P-net routing requirement. For example, when P consists of all 2-pin nets, then a P-net routing requirement is equivalent to a 2-pin net routing requirement, and a P-universal switch block is equivalent to a universal switch block.

The new switch block design problem is as follows. Given a net pattern set P, and two k-dimensional nonnegative integer vectors d and c, design an optimal P-universal $(dw + c)$-SB for every $w \geq 1$.

- Computing the set of all P-net routing requirements for $(dw+c)$-SB's for every $w \geq 1$ is important in the design of P-universal $(dw+c)$-SB's because the set of routing requirements is used to test if a candidate design is P-universal. We model a P-net routing requirement as a nonnegative integer vector, called a *tight routing requirement vector*, satisfying a system of linear Diophantine equations. We use the known Hilbert basis algorithm [18] to compute the set of minimal solutions (the minimal is by component-wise less than ordering) of the system. Once the set of minimal solutions is known, we can use it to generate all other solutions effectively by taking nonnegative linear combinations. Graph theory approaches are used to enumerate all multi-pin net routing requirements for a $(4, w)$-SB in [23, 26]. The use of the system of linear Diophantine equations solves the problem of computability of the set of minimal routing requirements of (k, w)-SB for any fixed k, which was raised in [23, 21].

- Using the minimal solutions of the system of linear Diophantine equations, we prove a new decomposition theorem for routing requirements for $(dw + c)$-SB design, which says that a routing requirement for a

$(dw + c)$-SB can be decomposed into some routing requirements for (dp)-SBs and one routing requirement for a $(dr + c)$-SB, where p is determined by d and c, and r is determined by p and w, and the options for r for all $w \geq 1$ is finite. This permits the decomposition design scheme for P-universal $(dw+c)$-SB, in which we design the so-called prime P-universal (dp)-SB and P-universal $(dr + c)$-SB's for a finite number of r's, and then use them to build all other P-universal $(dw + c)$-SB's. This approach not only makes the switch block design manageable, but also produces linear sized switch blocks and an efficient feasible routing algorithm.

- The optimal hyper-universal $(2, w)$-SB's and $(3, w)$-SB's as well as an approximate $(4, w)$-SB design are given in [22]. We will use these switch block designs as examples to illustrate our decomposition design method. The prime hyper-universal $(4, w)$-SB's $(w = 1, \ldots, 7)$ are improved from the results of [22] to optimal designs for $w = 1, 2, 3, 4, 5$ and to near optimal $(4, w)$-HUSB's for $w = 6, 7$. With the new designs, the compound hyper-universal $(4, w)$-SB has $\lceil 6.34w \rceil$ switches.

- We prove a new decomposition theorem for universal (k, w)-SB's, by which a union of $w/2$ copies of optimal universal $(k, 2)$-SB is an optimal universal (k, w)-SB when w is even, and a union of a universal $(k, \frac{k+3-i}{3})$-SB and $\frac{3w-k-3+i}{6}$ copies of universal $(k, 2)$-SB forms a universal (k, w)-SB when $w \geq \frac{k+3-i}{3}$ is odd, where $1 \leq i \leq 6$ and $i \equiv k \pmod 6$. A detailed proof for optimal universal $(k, 2)$-SB is also given.

Some of the results in this dissertation have been published. The decomposition design scheme for hyper-universal (k, w)-SB's, including a design of hyper-universal $(4, w)$-SB with $\lceil 6.7w \rceil$ switches were published in [22, 23]. The improved prime hyper-universal $(4, w)$-SB's with experimental justifications were published in [24]. The fundamental results about hyper-universal switch blocks were published in [26], and new improvements on universal (k, w)-SB's with experimental justifications were presented in [25].

1.4 Outline

The rest of the dissertation is organized as follows. Chapter 2 describes the background and some previous results on switch block designs. Chapter 3 formally introduces the combinatorial models for switch blocks. Chapter 3 also provides the formal definitions and notations used in this dissertation. Chapter 4 presents the Hilbert basis approach to enumerating a set of minimal routing requirements. The sets of minimal routing requirements for $(3, w)$-SB's and $(4, w)$-SB's are enumerated explicitly. Chapter 5 presents a general decomposition design scheme with examples of designing optimal hyper-universal $(2, w)$-SB's and $(3, w)$-SB's. Chapter 6 focuses on designing general hyper-universal $(4, w)$-SB's, and Chapter 7 is about the universal (k, w)-SB designs, including the detailed proof of a new decomposition theorem.

Chapter 2
Background

In this chapter, we describe the background for switch block design problems and some of the previous work on switch block designs. Section 2.1 describe the switch blocks for FPGA routing architectures. Section 2.2 presents various switch blocks which have been studied, and Section 2.3 illustrates generic switch blocks in switching networks.

2.1 Field Programmable Gate Arrays and Switch Blocks

An FPGA contains an array of logic blocks, and some other blocks such as memory blocks and input/output blocks (aligned in row and column). These blocks are connected to a reconfigurable interconnection network, which consists of wire segments of various lengths and programmable switches connecting terminals of segment wires. In order to manage the network design, implementation and fabrication, a reconfigurable interconnection network is usually built of switch blocks joined together by wire segments to form a certain architecture.

2.1.1 Field Programmable Gate Arrays and Routing Networks

A typical FPGA architecture is the symmetric island style model, as shown in Figure. 2.1, which has been studied extensively. See, for example [42, 12, 9, 3, 14, 61, 56, 57]. Island style architectures are used in many FPGA's, including the XC4000 series FPGA's [61].

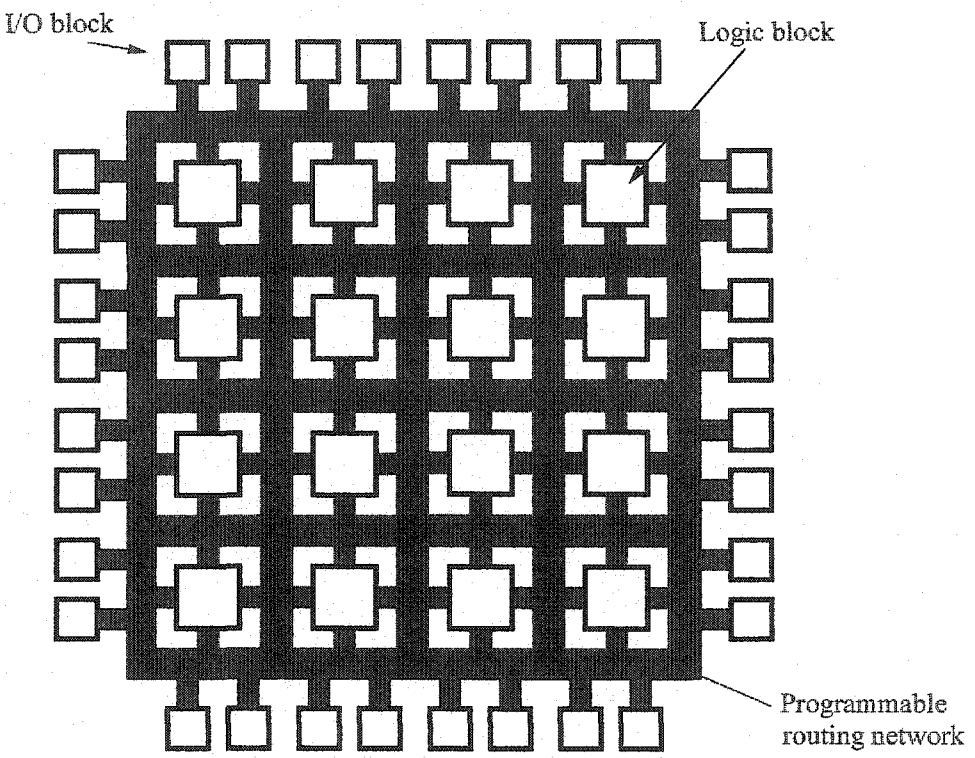


Figure 2.1: Island style FPGA diagram (from [43]).

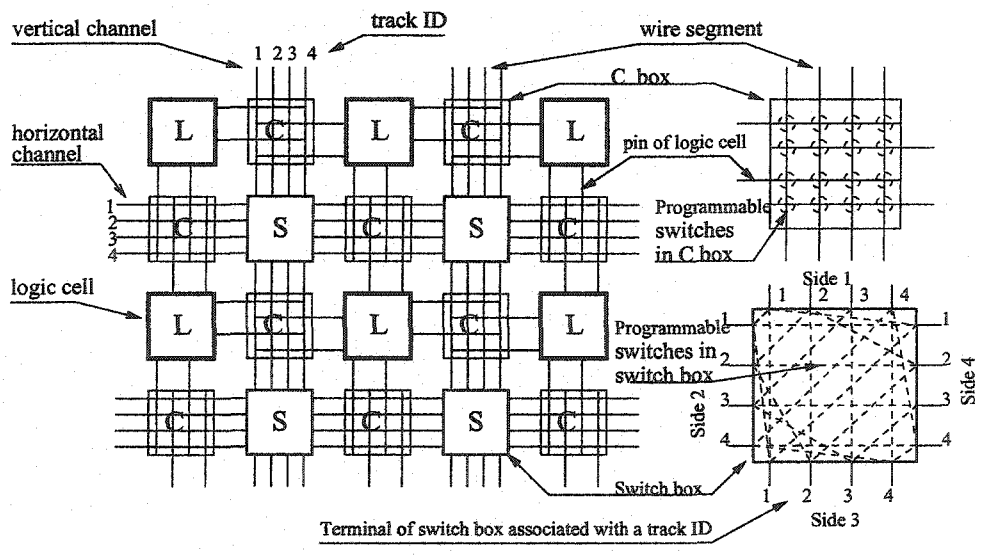


Figure 2.2: Switch blocks in an island-style FPGA.

Figure. 2.2 shows a more detailed diagram of an island style FPGA architecture, in which the logic blocks are labelled L, switch blocks are labelled S, and the connection blocks are labelled C. A *logic block* can be a single p-input *look up table* (p-LUT) capable of realizing every Boolean function with p inputs by reconfiguration, or a *clustered logic block* consisting of several p-LUTs. Logic blocks are separated by vertical and horizontal channels. There are w (called the *channel capacity*) prefabricated parallel wire segments (only short wire segments are depicted) running between each pair of adjacent L-blocks in both the vertical and horizontal channels. The wire segments in a vertical (or horizontal) channel are arranged in w vertical (or horizontal) tracks; each track within a channel is assigned a distinct integer in $\{1, \ldots, w\}$ as its track ID. There are connection blocks in the channel between adjacent L-blocks. A switch block is located at each intersection of a vertical and horizontal channel. In Figure 2.2, the switch block has four sides and four wire segment terminals on each side; programmable switches join certain terminals on different sides. It is a $(4, 4)$-SB. The switches in the switch blocks are used to connect wire segments, while switches in a C-block are used to connect a L-block to wire segments.

When an FPGA is used to implement a Boolean function represented as a Boolean network, a partitioning algorithm is used to decompose the Boolean network into some smaller subnetworks such that each of them can be implemented by a single logic block. Then a placement algorithm is employed to select a logic block for each subnetwork, and the input/output pins of the selected logic blocks are assigned to a set of disjoint groups, where each group called a *connection request*[1] as the pins in the group need

[1]It is called a net in VLSI design.

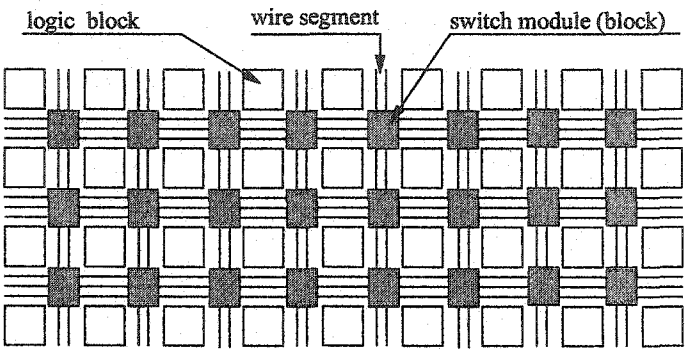
to be connected, and the set of groups (or connection requests) is called a *connection requirement.* Then a routing algorithm is called to choose wire segments and switches such that all the pins in a group are connected and pins in different groups are not connected. An assignment of wire segments and switches realizing the connection reqirement is called a *detailed routing* (or *realization*) of the connection requirement. A connection requirement is *routable* if a detailed routing exists. The term "routability" refers to the possibilities that a given connection requirement is realizable.

There are basically two kinds of routing algorithms [12, 9, 3, 15, 17, 13]. A *two stage router* first determines a route for each connection request, called a global routing, then determines the detailed routing according to the global routing. A *one stage router* does the routing successively by searching for available wire segments and switches for each connection request. Routing is a hard problem. It is known that determining the existence of a detailed routing with respect to a global routing is NP-complete [58].

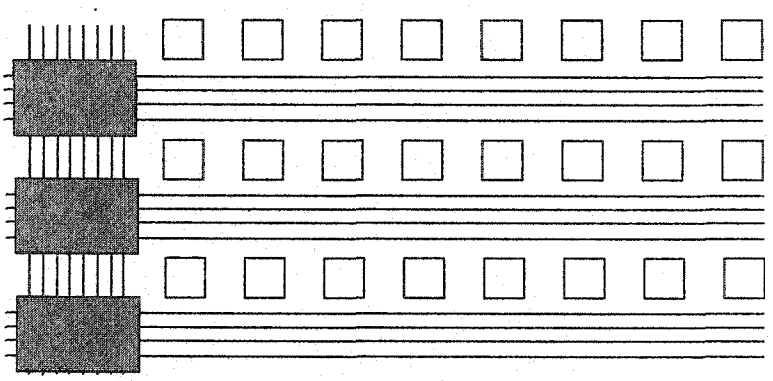
2.1.2 Routing Networks and Switch Blocks

We view the interconnection network formed by wire segments and switch blocks as the framework of a reconfigurable interconnection network, and the C-blocks as the interface between the logic blocks and the framework. A reconfigurable interconnection network model defines how switch blocks are connected by wire segments. Figure 2.3 shows conceptual diagrams of three commonly used models, i.e., the mesh model, the row-model, and the hierarchical model.

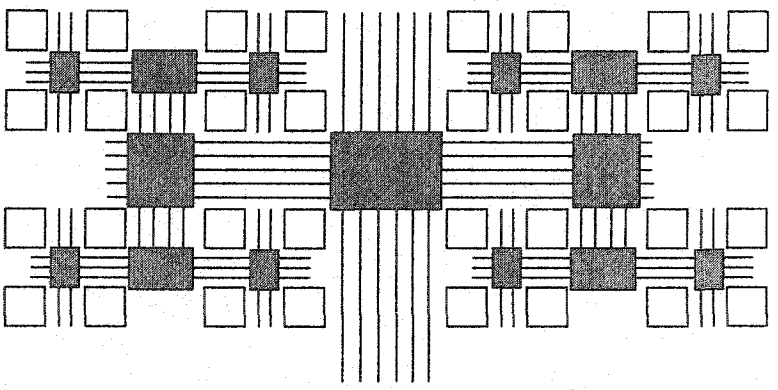
Since switch blocks are the main switching component in a routing net-



(a) a mesh model (island style)



(b) a row model



(c) a hierachical model (tree style)

Figure 2.3: Diagrams of a mesh model, a row-model, and a hierarchical model.

work and they have a great effect on routability and area efficiency, switch block design is very important for FPGA routing network design.

Designing switch blocks is sometimes associated with a routing scheme. For example, in the greedy routing architecture [59, 57], h-sided predetermined switch blocks are required. Even after switch blocks are designed, determining how to connect individual switch blocks to a reconfigurable interconnection network is another challenge because there are w factorial different ways to connect by wire segments the terminals in a side of a switch block to the terminals in a side of another switch blocks with channel capacity w. Determining how to make these connections such that the resulting reconfigurable interconnection network has the highest routability is a challenge [9].

The structure of a switch block defines the connections between terminals via switches. Two switch blocks are topologically equivalent or isomorphic if one can be changed to another by permuting terminals within the sides. Therefore, a topological design for a switch block defines a class of isomorphic switch blocks.

Routability and the number of switches are two important criteria in switch block design. High routability is one goal, and a small number of switches is another goal. A small number of switches is aimed at achieving good area efficiency. However high routability and a small number of switches are conflicting goals.

Rose and Brown examined the trade-off between chip level routability and area efficiency [42]. They introduced a metric called *flexibility* , denoted by F_s, which is the maximum number of switches which connect to a terminal in a switch block. They investigated the effect of flexibility on

routability, and observed that $F_s = 3, 4$ achieves good routability.

However, as there can still be many different switch blocks with the same flexibility, it is important to analyze the routability differences among them and to find optimal designs. This initiated the investigation of switch block designs [14].

Four types of 4-sided switch blocks designs have been proposed [61, 14, 51, 22]. The first type is the class of disjoint $(4, w)$-switch blocks, which are constructed from w complete $(4, 1)$-SB's. The Xilinx XC4000 series FPGA's use disjoint $(4, w)$-SB's. Figure 2.4(a) shows a disjoint $(4, 3)$-SB.

The second type is the class of Wilton's switch blocks [51]. A Wilton's $(4, w)$-SB has terminals $t_{i,j}, j = 0, \ldots, w-1$ on side $i, i = 0, \ldots, 3$. and switch set

$$\cup_{i=0}^{w-1} \{t_{0,i}t_{2,i}, t_{1,i}t_{3,i}, t_{0,i}t_{1,(w-i)}, t_{1,i}t_{2,(i+1)}, t_{2,i}t_{3,(2w-2-i)}, t_{3,i}t_{0,(i+1)}\} \quad (2.1)$$

where the second index is taken modulo w and a $t_{i,j}t_{h,k}$ denotes a switching joining terminals $t_{i,j}$ and $t_{h,k}$. Figure 2.4(b) depicts a Wilton's $(4, 3)$-SB.

The third type is a universal switch block. Chang et al. [14] first proposed the concept of a universal switch block, and gave a generation algorithm for universal $(4, w)$-SB's. They called a $(4, w)$-SB generated by the algorithm a symmetric universal $(4, w)$-SB, written $M_{4,w}$. They proved that an $M_{4,w}$ is an optimal universal $(4, w)$-SB, and has flexibility three and $6w$ switches. It was also proved that an $M_{4,w}$ is isomorphic to a disjoint union of $w/2$ copies of $M_{4,2}$ when w is even, and a disjoint union of $(w-1)/2$ copies of $M_{4,2}$ and one $M_{4,1}$ when w is odd. Figure 2.4(c) depicts a universal $(4, 3)$-SB (written $(4, 3)$-USB) isomorphic to the $M_{4,3}$.

The fourth type is a hyper-universal switch block, which was first pro-

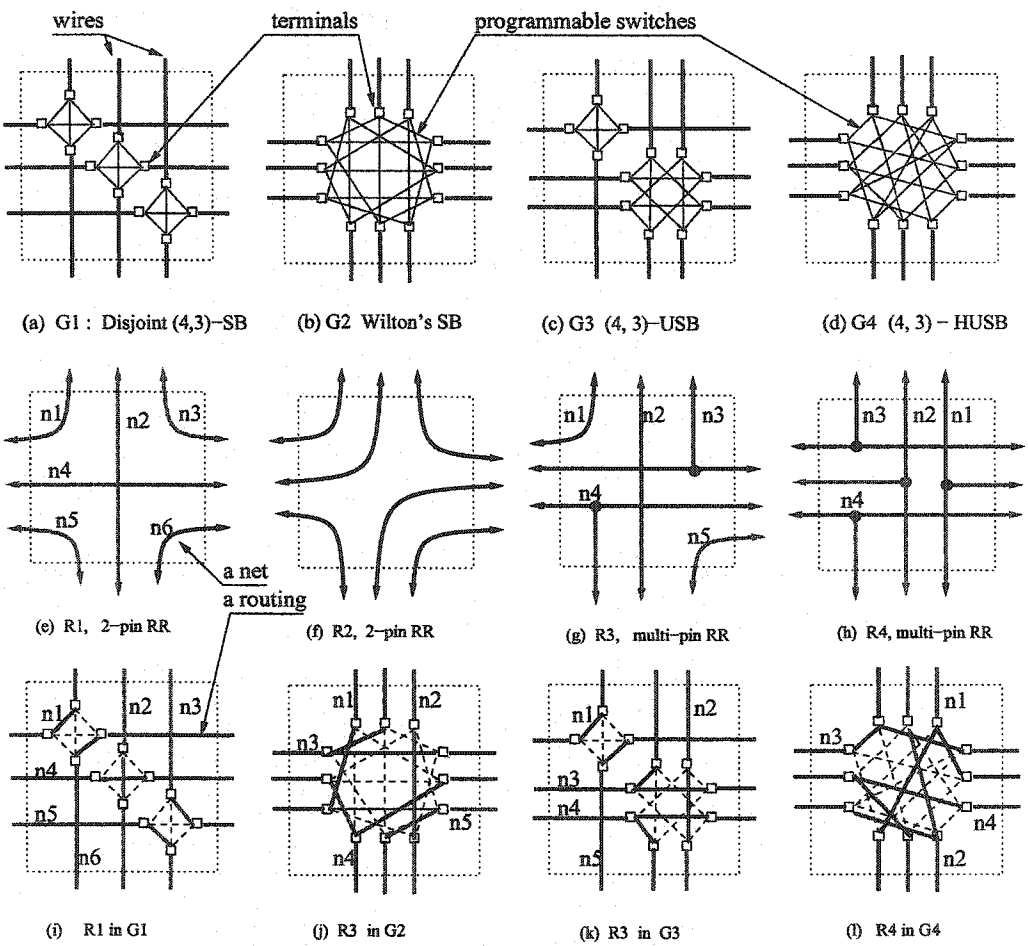


Figure 2.4: Routing requirements and feasible routings for four different $(4, 3)$-SB designs.

posed in [22]. The concept of hyper-universal switch blocks generalizes the concept of universal switch blocks by allowing multi-pin nets. Figure 2.4(d) presents a hyper-universal $(4,3)$-SB (written $(4,3)$-HUSB) [24].

It can be seen that all the four representative $(4,3)$-SB's in Figure 2.4 have flexibility three and eighteen switches. However, their routing capabilities are not the same. It can be checked that routing requirement R2 shown in Figure 2.4(l) is not routable in G2, and R4 is not routable in G3, and neither R3 nor R4 is routable in G1. But R1, R2, R3, and R4 are all routable in G4. This fact tells us that besides the number of switches and the flexibility, the connection topology is important to the routing capability of a switch block.

Switch blocks are designed for FPGA routing networks. It is important to test the routability of a routing network resulted by a given switch block designs. Experiments have been done by different research groups using VPR (versatile placement and routing) [7]. For a given class of switch blocks, and a given benchmark circuit, the experiment is to find the minimal channel capacity w such that a routing network can implement the benchmark circuit with VPR. This is repeated for all benchmark circuits, and then the sum of all these minimal channel capacities is used as a measure for the routability of the class of switch blocks. Results in [14] say that universal switch blocks are better than disjoint switch blocks. Results in [51] say that Wilton's switch blocks are better than disjoint switch blocks and universal switch blocks. Recent experiments have shown that the results are sensitive to the number of iterations. At 35 iterations, Wilton's switch blocks are the best, then universal switch blocks, hyper-universal switch blocks, and disjoint switch blocks. But at 100 iterations, hyper-universal

switch blocks turn out to be the best, the next are universal switch blocks, Wilton's switch blocks and disjoint switch blocks. However, for all benchmark circuits, universal switch blocks, hyper-universal switch blocks and Wilton's switch blocks are better than disjoint switch blocks.

Because the experimental results depend on many factors such as routing architecture, the placement algorithm, the routing algorithm, benchmark circuits, etc., it is hard to decide which type of switch blocks is of highest routability by experiment. However, from the theoretical point of view, the hyper-universal $(4, w)$-SB's do have advantages in both routing capability and structure because a hyper-universal $(4, w)$-SB with $w \geq 8$ can be constructed by the disjoint union of hyper-universal $(4, w)$-SB with smaller w. This property makes finding a feasible routing in hyper-universal $(4, w)$-SB's easier than in others and layout design of hyper-universal $(4, w)$-SB's is no harder than that of disjoint switch blocks [24].

2.2 Generic Switch Blocks

The switch blocks discussed in Section 2.1.2 have four sides and all sides have the same channel capacity. However, in practical routing networks, switch blocks with different numbers of sides may be used. For example, at the four corners of an island-style FPGA routing network, 2-sided switch blocks are used, and at the edge, 3-sided switch blocks are used. In the tree style, 6-way mesh, and 8-way mesh reconfigurable interconnection network models [46], switch blocks with more sides are used. Figure 2.5 shows some (k, w)-SB's with $k = 4, 5, 6$, and their routing requirements and feasible routings. On the other hand, the numbers of terminals on different sides may not be the same either. That is, a switch block may have different

channel capacities on different sides. Another variable factor is routing capability. Different routing capabilities may be required according to different sets of connection requirements.

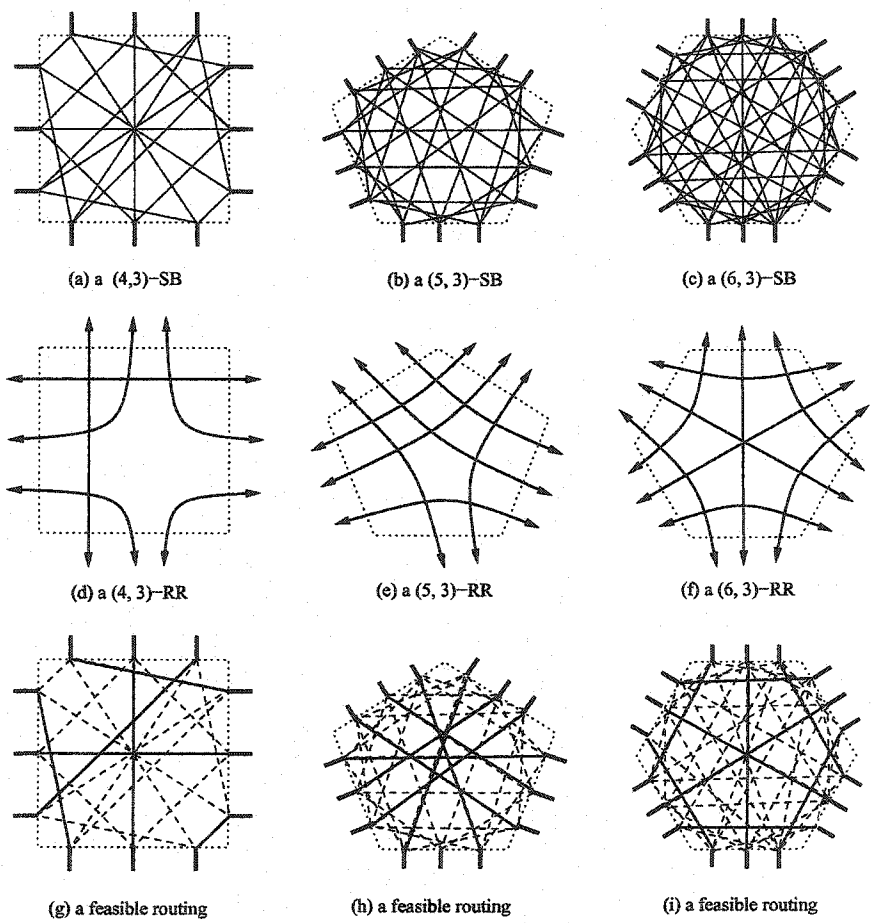


Figure 2.5: Examples of switch blocks with different dimensions.

Therefore, switch blocks with various dimension, channel capacity, and routing capability specifications should be considered. Only a few combinations of the three types of specifications such as universal $(4, w)$-SB's and hyper-universal universal $(4, w)$-SB's have been studied. More general classes of switch blocks have not yet been addressed.

2.2.1 Universal Switch Blocks and Hyper-Universal Switch Blocks

Shyu et al. [46] investigated the generic universal (k, w)-SB design problem and generalized the $(4, w)$-SB generation algorithm of [14] to a (k, w)-SB generation algorithm, which generates an $M_{k,w}$ for any pair of (k, w) with $k \geq 2, w \geq 1$. The main result in [46] was that an $M_{k,w}$ is an optimal universal (k, w)-SB. Unfortunately, the result was incorrect. A counterexample was given in [27].

Fan et al. examined the universal (k, w)-SB design problem using the decomposition design technique developed for hyper-universal (k, w)-SB's. In [27, 26], it was proved that an $M_{k,w}$ is an optimal universal (k, w)-SB's only for $k \leq 6$ or even w.

2.2.2 Crossbars via Switch Blocks

An $n \times m$ *crossbar* consists of n parallel input wires and m parallel output wires; they are placed orthogonally such that each input wire crosses every output wire, and programmable switches are placed at some cross-points to join the pairs of wires. A crossbar is called a full crossbar if there is a switch joining each pair of input and output wires, otherwise it is called a partial crossbar. It is clear that a full $n \times m$ crossbar has nm switches. Figure 2.6(a) shows a diagram of a full 12×6 crossbar where the vertical wires are inputs and the horizontal wires are outputs. A crossbar can be viewed as a 2-sided switch block: the input wire terminals are on one side and output wire terminals are on the other side, and two terminals are joined by a switch if and only if there is a switch joining the two wires of the terminals in the crossbar. See Figure 2.6(a) and (b) for examples.

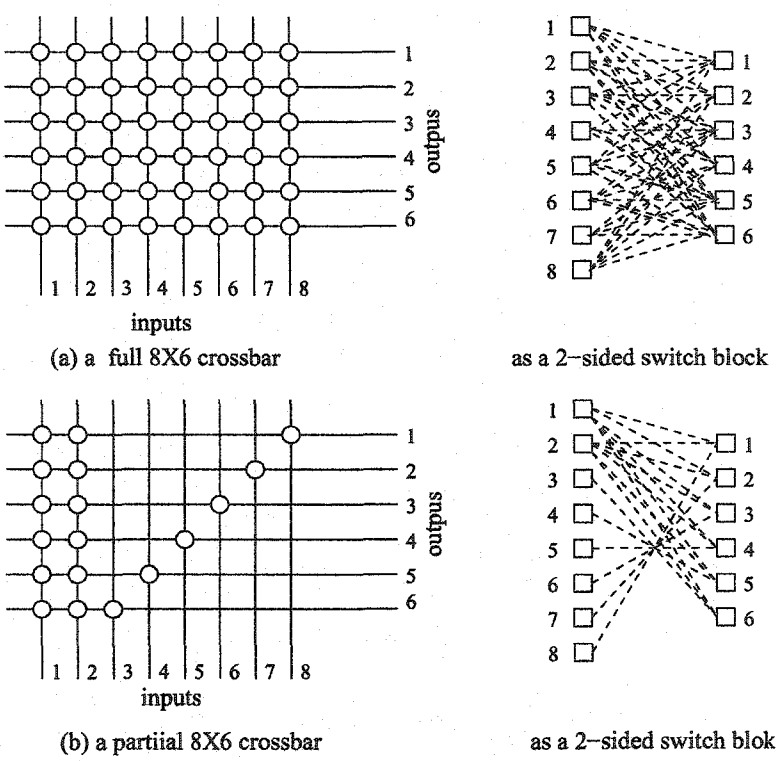


Figure 2.6: Crossbars and graph representations.

It is clear that a full $m \times n$ crossbar with $n \leq m$ can route an input signal to an n-subset of output wires in any given order. Particularly, a full $n \times n$ crossbar can permute the signals. On the other hand, if an $n \times (n/2)$ crossbar is used as an $(n, 1)$-switch module with input wires as terminals, it can route all multi-pin net routing requirements applied to the input terminals.

However, full crossbars are too expensive when n and m are large. The problem of designing crossbars with large m, n focuses on selecting cross-points where switches are going to be placed to satisfy certain routing specifications. There are basically three types of crossbar design problems. The first one is to design a partial $n \times m$ $(n \geq m)$ crossbar with a minimum number of switches and such that every group of m inputs can be routed to m outputs. Nakamura and Masson [36] showed that an optimal design has $(n - m + 1)m$ switches. Figure 2.6(b) shows an optimal 8×6 crossbar.

The second problem is called the *sparse crossbar* design problem. It is to design a partial $n \times n$ $(n \geq m)$ crossbar with a linear number of switches (in terms of n) and such that the percentage of routable routing vectors is at least a given r. A routing vector is an n-dimensional 0-1 vector $(x_1, \ldots, x_n)$, which is used to represent a selection of input terminals with $x_i = 1$ meaning the i-th input terminal is selected. A routing vector is routable if all the selected terminals can be routed to output terminals simultaneously.

The third problem is, given a number p, design a partial $n \times m$ $(n \geq m)$ crossbar of p switches which maximizes the number of routable routing vectors.

The second and the third problems were investigated in [31, 32], where algorithms for generating crossbars and a probabilistic based evaluation

method were proposed. Note that determining whether a selected set of input terminals can be routed to the output terminals in a partial $n \times m$ ($n \geq m$) crossbar can be done in time polynomial in n by using a max-flow algorithm.

The switch matrix studied in [54] is a generalization of crossbars. A switch matrix has two groups of wires placed orthogonally like a crossbar. There are two kind of switches, crossing switches like the switches in crossbar, and separating switches which are used to separate the wires. A switch matrix is a multi-stage 4-sided switch module. Although a switch matrix is not a switch block by our defintion, the decomposition design method can still be used to design switch matrices.

2.2.3 Switch Circuit and Layout Designs

All switches in a reconfigurable interconnection network are eventually implemented by circuits, and circuits are laid out for fabrication. Switch circuit design determines what types of switches are used. The layout design determines the floor plan of gates and the map of wires in each metal layer. Circuit design and layout design are the crucial steps in achieving area, time and power efficiencies.

There are basically three types of switches currently being used: a pass transistor switch, a multiplexer, and a tri-state buffer [12, 9]. A pass transistor switch is bidirectional, i.e. a signal can go from one end to another end of the switch in either direction. Both a multiplexer and a tri-state buffer are unidirectional switches, i.e., a signal can only go from one end to another end of the switch. Each of these switches has advantages and disadvantages. Strategies for using these switch circuits associated with certain

processing technologies are discussed in [8, 34]. A switch block layout with evaluations was presented in [45]. A switch block consisting of bidirectional switches can be modelled as a graph. A switch block consisting of unidirectional switches can be modelled as a directed graph. For convenience, we use bidirectional switches, i.e., a pass transistor switch as our basic switch model in the topological design of switch blocks. So we model a switch block as an undirected graph.

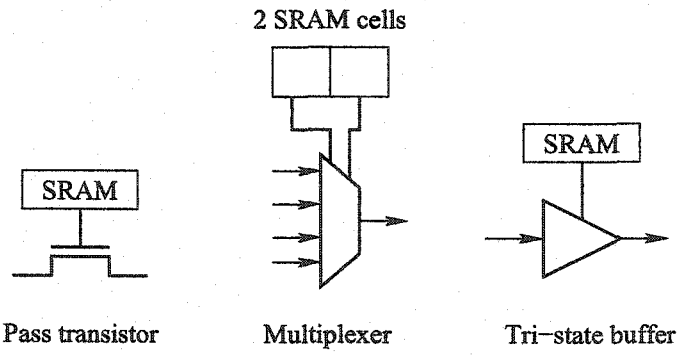


Figure 2.7: Three types of switch gates (From [9]).

2.3 Switching Networks and Switch Blocks

In this section, we examine the applications of switch blocks in circuit switching communication networks. Crossbars have been used widely in circuit switching networks. However, general switch blocks can also be used to build a circuit switching network.

2.3.1 Switching Networks

There are two types of message switching techniques being used in communication networks; circuit switching and packet switching [47]. *Circuit*

switching networks such as traditional telephone systems make a real connection path between communicating parties. *Packet switching networks*, such as the internet, transfer messages by sending packets, which carry the addresses of their destinations. A packet travels to its destination along some path which is not predetermined. Even though the packet switching model is widely used for communication networks, the circuit switching method is still a good solution when a high quality service is required and it is the main switching method for an on-chip switching network.

Similar to an FPGA routing network, a circuit switching network uses wires and switches. The wires may be optical fibers, or radio channels. Whatever techniques are used, the basic function of a reconfigurable interconnection network is to make pipeline connections between pairs of communication terminals. One of the design goals of the reconfigurable interconnection network is to implement as many point-to-point connections simultaneously as possible.

A directed switching network connecting n input terminals to n output terminals is commonly used. Such a network is said to be *rearrangeable* if it is routable for any routing requirement formed by a permutation between input and output terminals, that is, for any permutation p on $\{1, \ldots, n\}$, the switching network can be configured to connect input terminal i to $p(i), i = 1, \ldots, n$ simultaneously. Such a network is said to be *non-blocking* if there is a routing strategy such that it can route any routing requirement dynamically, i.e., without changing the previous connections to do the routing for later requests. It is clear that if a network is non-blocking it must be rearrangeable. But the inverse is not true. Clearly, an $n \times n$ crossbar is a non-blocking network.

One of the basic design problems is to design a rearrangeable multi-stage network with a small number of switches. This problem has been studied extensively ever since telephone systems were first built [16, 4, 47].

Clos [16] proposed the first three stage non-blocking network (known as a Clos network), which uses full crossbars of smaller sizes as basic switching modules and has $6n^{3/2}$ switches, n is the number of input terminals. Benes [4] proposed a rearrangeable network of $4nlog(2n)$ switches and $O(log\ n)$ stages. The well-known Banyan, Baseline, Butterfly, Delta, and Omega networks also have $O(log\ n)$ stages.

All the early switching networks used full crossbars as the basic switch modules. In other words, they use 2-sided switch blocks as the constructing switch modules.

Yen et al. [64] recently proposed a three-stage, one-sided, rearrangeable polygonal switching network, which uses a universal $(\sqrt{n}, \sqrt{n})$-SB as the main switch module and a full $\sqrt{n} \times \sqrt{n}$ crossbar attached to each side. The polygonal switching network has less than $2n^{3/2}$ switches.

Just as a universal (k, w)-SB offers a solution to the point-to-point connection in a polygonal switching network, a hyper-universal (k, w)-SB can be a solution to the many-to-many connection in a polygonal switching network. It can be seen that a polygonal switching network with a hyper-universal (k, w)-SB at the center is capable of realizing any connection requirement. Such a network is much simpler than the conference network for group communication network proposed in [63], which is based on crossbar switch modules.

We next use an example to illustrate the use of switch blocks in switching network design. Suppose that there are four towns labelled N, E, S, and

W, and each of these towns has three people, named N_i, E_i, S_i, and W_i, $i = 1, 2, 3$. The task is to design a central telephone switching office connecting these people, such that it enables group communication. That is, for every connection requirement, i.e., a partition of the twelve people, the system can be reconfigured to connect simultaneously the people of every group of the partition. For instance, $\{N_1, E_1, S_1\}, \{E_2, S_2, W_1\}, \{S_3, W_2, N_2\}, \{W_3, N_3, E_3\}$ is a partition of the twelve people. The system should be able to connect simultaneously N_1 and E_1 and S_1 of the first group, E_2 and S_2 and W_1 of the second group, S_3 and W_2 and N_2 of the third group, W_3 and N_3 and E_3 of the fourth group, and different groups are not connected. Note that we assume that a person is only allowed to be in one communication group at a time.

Plan 1: Use a 12×6 full crossbar. See Figure 2.8(a).

It is obvious that this design can realize all possible connection requirements. It has 72 switches.

Plan 2: Use a complete $(12, 1)$-SB. See Figure 2.8(b).

It is clear that a complete $(12, 1)$-SB is routable for every connection requirement. This design has 66 switches.

Plan 3: Use one universal $(4, 3)$-SB and four 3×3 full crossbars. See Figure 2.4(c).

Note that the universal $(4, 3)$-SB is a union of a universal $(4, 2)$-SB and a universal $(4, 1)$-SB. This plan uses $36+18 = 54$ switches. It is routable for every connection requirement with at most two people in a group because such a connection requirement induces a 2-pin net routing requirement for the $(4, 3)$-SB, and a feasible routing can be found in the switch block, and then the crossbars can be used to route the terminals to the corresponding

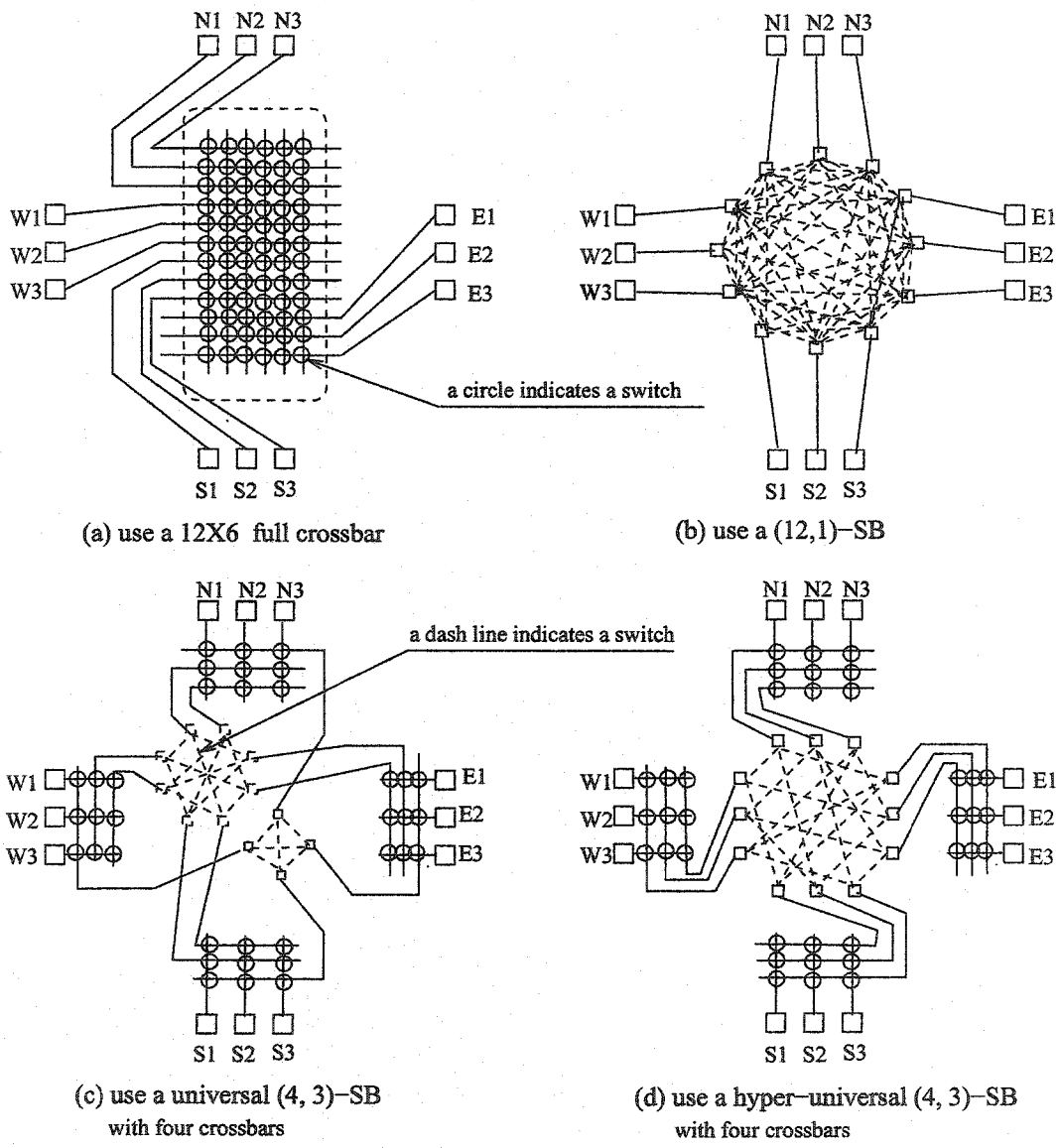


Figure 2.8: Examples of reconfigurable interconnection networks.

people.

However, this design can not route the connection requirement

$$\{N_1, E_1, S_1\}, \{E_2, S_2, W_1\}, \{S_3, W_2, N_2\}, \{W_3, N_3, E_3\}$$

because a feasible routing would result in a feasible routing of R4 in G3 in Figure 2.4(c) on the central $(4, 3)$-SB.

Plan 4: Use a hyper-universal $(4, 3)$-SB and four 3×3 full crossbars. See Figure 2.4(d).

This plan uses $36 + 18 = 54$ switches. It is routable for all connection requirements because any connection requirement induces a multi-pin routing requirement for the $(4, 3)$-SB and we can find a feasible routing for the routing requirement and use the four crossbars to connect to the people in each group.

Notice that Plan 4 uses the minimum number of switches of all plans being proposed. We can also consider other possible three stage designs. For instance, we can use one hyper-universal $(6, 2)$-SB at the center and six 2×2 crossbars but we have to design a hyper-universal $(6, 2)$-SB first. We can also use an irregular hyper-universal $(3, 4, 5)^T$-SB.

Clearly we can make many different choices among hyper-universal switch blocks with a total of twelve terminals. But first, we need know how to design hyper-universal switch blocks with various dimension and channel capacity specifications. We are going to address how to design such regular and irregular switch blocks in this dissertation.

Chapter 3

Switch Block Design Problems

In this chapter, we first describe combinatorial models for switch blocks, routing requirements, and feasible routings, then we describe various types of switch blocks according to their dimension and channel capacity and routing capability specifications. Finally, we define the switch block design problems considered in this dissertation. This chapter also serves as a reference for the formal terminology and notation.

3.1 Definitions from Graph Theory

We use standard terminology and notation from graph theory and hypergraph theory (see for example, [11, 5]) with a minor modification to fit our switch block design problem.

By a *graph* we mean a *multiple* graph, i.e., an ordered pair (V, E), where V is a finite *vertex* set and *edge* set E is a *multi-set* of 2-subsets of V. [1] For convenience, an edge $\{u, v\} \in E$ is sometimes denoted by uv. The definitions for a *path*, a *connected graph*, a *component*, a *connected graph*, a *cycle*, a *Hamiltonian cycle*, and a *tree* are available in [11].

We note that in a *multi-set*, repetition of an element is allowed; the number of replications of an element e in a multi-set A is called the *multiplicity* of e, denoted by $p_A(e)$ (or simply $p(e)$). Then $p_A(e) > 0$ if e is an

[1] The graph we considered here is allowed to have multiple edges, i.e., edges corresponding to the same 2-subset of vertices, but not allowed to have a loop edge.

element of A (written $e \in A$). We define $p_A(e) = 0$ if e is not an element of A (written $e \notin A$). We also note that operations on multi-sets take into account the multiplicities. A multi-set B is a *sub-multi-set* of a multi-set A (written $B \subseteq A$) if for every element $e \in B$, $p_B(e) \leq p_A(e)$. Two multi-sets A and B are equal (written $A = B$) if $A \subseteq B$ and $B \subseteq A$. When $B \subseteq A$ but $B \neq A$, we write $B \subset A$ and call B a *proper sub-multi-set* of A. The *union*, *intersection* and *difference* of multi-sets are defined by:

$A \cup B = \{e,\ p(e) = p_A(e) + p_B(e) \mid e \in A \text{ or } e \in B\}$,

$A \cap B = \{e,\ p(e) = \min\{p_A(e), p_B(e)\} \mid e \in A \text{ and } e \in B\}$,

$A \setminus B = \{e,\ p(e) = p_A(e) - p_B(e) \mid e \in A \text{ and } p_A(e) > p_B(e)\}$.

A multi-set is a *set* if the multiplicity of every element is one.

A graph is said to be *simple* (or a *simple graph*) if all its edges have multiplicity one. A graph is called *k-partite* if its vertex set can be partitioned into k *parts* such that no edges join vertices within a part.

A *hypergraph* is an ordered pair (V, E), where V is a *vertex set* and *edge set* E is a multi-set of subsets of V. [2] Let G be a hypergraph, we use $V(G)$ and $E(G)$ to denote the vertex set and edge set of G, respectively. Let $v \in V(G)$ be a vertex and $e \in E(G)$ be an edge. If e contains v, we write $v \in e$ and say e is *incident* with v. The *degree* of a vertex v in G is the number of edges incident with v, written $d_G(v)$. A hypergraph G is said to be *k-regular* (or *regular*) if the degrees of all its vertices are equal to k (identical). The *size* edge e (written $|e|$) is the cardinality of e, i.e., the number of vertices contained in e. When $|e| = 1$ we call e a *singleton* (or a singleton edge). A hypergraph is called a *k-graph* if all edges have size at

[2] The hypergraph we defined here is sometimes called a *multiple hypergraph* because it allows multiple edges, i.e., edges corresponding to the same subset of vertices.

most k. Then a 2-graph containing no singleton is a graph. For 2-graphs, we define a path, a component, a cycle, and a cut edge in the same way as for graphs by considering edges of size two. However, two singletons incident with the same vertex is considered as an even cycle. A hypergraph G' is a *sub-hypergraph* (or simply *subgraph*) of G (written $G' \subseteq G$) if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$, and a *spanning subgraph* if, in addition, $V(G') = V(G)$. When $G' \subseteq G$ and $G' \neq G$ we call G' a *proper sub-hypergraph* of G (written $G' \subset G$). In particular, when G' is a proper spanning k-regular (regular) sub-hypergraph of G we called G' a *k-factor* (*factor*) of G.

A regular hypergraph is said to be *non-decomposable* if it does not contain a k-factor for any $k \geq 1$, otherwise it is *decomposable*. A hypergraph is said to be *k-factor-free* if it does not contain a k-factor. Two hypergraphs G and G' are *isomorphic* if there is a bijection f from $V(G)$ to $V(G')$ such that $e = \{v_{i_1}, \ldots, v_{i_t}\} \in E(G)$ if and only if $f(e) = \{f(v_{i_1}), \ldots, f(v_{i_t})\} \in E(G')$, and $p_{E(G)}(e) = p_{E(G')}(f(e))$ for every $e \in E$; such a bijection f is called an *isomorphism* from G to G'.

3.2 Modelling of Switch Blocks

In this section, we formally define our combinatorial models for switch blocks, routing requirements and feasible routings. We have introduced these concepts in Chapter 1.

3.2.1 Dimension and Channel Capacity Specifications

The *dimension* of a switch block refers to the number of sides, and the *channel capacity* of a side refers to the number of terminals on the side.

A *k-sided switch block* is a switch block in which terminals are parti-

tioned into k sides and each switch joins a pair of terminals on different sides. We label the sides of a k-sided switch block by $1, 2, \ldots, k$. Suppose that the channel capacity of side i is d_i for $i = 1, \ldots, k$. Then we call the switch block a $(d_1, \ldots, d_k)^T$-SB, where $(d_1, \ldots, d_k)^T$ will denote a column vector.

A k-sided switch block with all sides having the same channel capacity w is abbreviated to a (k, w)-SB. We use *k-SB's* to denote the class of (k, w)-SB's with all $w \geq 1$. Let d and c be two k-dimensional nonnegative integer column vectors, then $(dw + c)$-SB's with all $w \geq 1$ define a class of switch blocks with channel capacities determined by vectors $(dw+c), w \geq 1$. When $d = (1, \ldots, 1)^T$ and $c = (0, \ldots, 0)^T$, a $(dw + c)$-SB is a (k, w)-SB and the class $(dw + c)$-SB's is the k-SB's.

3.2.2 Switch Block and Graph Representation

It is natural to represent a k-sided switch block by a k-partite simple graph with terminals as vertices and switches as edges and sides as parts. This model has been used by many authors, see for example [51, 14, 31].

Definition 3.2.1 *For a $(d_1, \ldots, d_k)^T$-SB, we denote the j-th terminal in side i by a vertex $v_{i,j}$ for $i = 1, \ldots, k$ and $j = 1, ..., d_i$. If there is a switch joining terminals $v_{i,j}$ and $v_{s,t}$, then we denote the switch by an edge $v_{i,j}v_{s,t}$. Then the $(d_1, \ldots, d_k)^T$-SB corresponds to a k-partite simple [3] graph G with vertex set partition $(V_1, \ldots, V_k)$, where $V_i = \{v_{i,j} | j = 1, \ldots, d_i\}, i = 1, \ldots, k$ and edge set E corresponding to the set of switches. We call it a representation graph of the switch block and denote it by $((V_1, \ldots, V_k), E)$.*

[3] Note that we assume that there is at most one switch joining a pair of terminals.

For convenience, we will use a switch block and its graph representation interchangeably. A complete $(d_1, \ldots, d_k)^T$-SB corresponds to the complete k-partite graph with vertex set partition $(V_1, \ldots, V_k)$, where $V_i = \{v_{i,j} | j = 1, \ldots, d_i\}$, $i = 1, \ldots, k$, and edge set

$$\{v_{i,j}v_{s,t} | 1 \leq i < s \leq k, j = 1, \ldots, d_i, t = 1, \ldots, d_s\}.$$

We use $K_{d_1,\ldots,d_k}$ to denote a complete $(d_1, \ldots, d_k)^T$-SB, and $K_{(k,w)}$ to denote a complete (k, w)-SB.

Let G be a $(d_1, \ldots, d_k)^T$-SB represented by graph $((V_1, \ldots, V_k), E)$, and G' be a $(d'_1, \ldots, d'_k)^T$-SB represented by graph $((V'_1, \ldots, V'_k), E')$. We say G and G' are *isomorphic* if there is an isomorphism f from $((V_1, \ldots, V_k), E)$ to $((V'_1, \ldots, V'_k), E')$ such that $f(V_i) = V'_i, i = 1, \ldots, k$. We define the *disjoint union* of G and G', written $G + G'$, to be a $(d_1 + d'_1, \ldots, d_k + d'_k)^T$-SB with graph representation $((V_1 \cup V'_1, \ldots, V_k \cup V'_k), E \cup E')$ and the terminals of G' follows the terminals of G on each side. A *disjoint union of t copies* of G (or simply t copies of G), written tG, is a disjoint union of t switch blocks where each of them is isomorphic to G. For example, $3K_{1,1,1}$ is the disjoint union of three copies of $K_{1,1,1}$, with graph representation $((V_1, V_2, V_3), E)$, where $V_i = \{v_{i,j} | j = 1, 2, 3\}, i = 1, 2, 3$ and $E = \{v_{i,j}v_{s,j} | 1 \leq i < s \leq 3, j = 1, 2, 3\}$.

3.2.3 Routing Requirements

A *routing requirement* is a set of nets, and a *l-pin net* (or a net) is a connection request which requires connecting l terminals on l different sides. We note that a 2-pin net indicates a point-to-point connection, and a *multi-pin net* (i.e, a l-pin net with $l > 2$) indicates a *multi-point connection* (also called *multiconnection*). There are three types of nets: a track-free net, a

track-fixed net, and a mixed net. In a *track-free net*, only the sides of the terminals are given, no specific terminals on the sides are given. A realization of a track-free net can use any available terminals on the specified sides. While in a *track-fixed net*, exact terminals are given. That is, a set of specific terminals are required to be connected. In a *mixed net*, some terminals are fixed and others are free.

Throughout this dissertation, we only consider track-free nets. Unless stated otherwise, by net we always mean a track-free net. We represent a track-free l-pin net specifying l different sides $i_1, i_2, \ldots, i_l$ by set $\{i_1, i_2, \ldots, i_l\}$. For instance, a 3-pin net that requests connecting three terminals on sides $1, 2$ and 3 is represented by $\{1, 2, 3\}$. With this representation of a net, a routing requirement can be expressed as a multi-set of subsets of side labels. For example, for a $(4, 4)$-SB, the multi-set of subsets

$$\{\{1, 2\}, \{1, 2\}, \{3, 4\}, \{2, 3, 4\}\}$$

represents a routing requirement which has four nets. The first net $\{1, 2\}$ requires connecting a terminal on side 1 to one on side 2, the second net $\{1, 2\}$ requires connecting another pair of terminals on sides 1 and 2 respectively. The third net $\{3, 4\}$ requires connecting a terminal on sides 3 to one on side 4, and the fourth net $\{2, 3, 4\}$ requires connecting three terminals which are on sides 2, 3 and 4, respectively.

Since in a feasible routing for a routing requirement a terminal is allowed to be used by at most one net realization, it is necessary for the existence of a feasible routing that the number of nets specifying a side is at most the number of terminals on that side. Therefore, we always assume that the number of nets specifying a side in a routing requirement is at most the

number of terminals on that side.

We formally define our representation of a routing requirement as follows.

Definition 3.2.2 *A routing requirement for a $(d_1, \ldots, d_k)^T$-SB, written $(d_1, \ldots, d_k)^T$-RR, is represented by a multi-set $\{N_1, N_2, \ldots, N_t\}$ with $N_i \subseteq \{1, \ldots, k\}, i = 1, \ldots, t$ and*

$$\sum_{j=1}^{t} |N_j \cap \{i\}| \leq d_i, i = 1, \ldots, k. \tag{3.1}$$

We call N_j a net *(or an $|N_j|$-pin net), and when $|N_i| = 1$, call N_i a singleton net. When $i \in N_j$ we say N_j is* incident with *(or* specifying*) side i.*

A $(d_1, \ldots, d_k)^T$-RR $\{N_1, N_2, \ldots, N_t\}$ is said to be tight, written $(d_1, \ldots, d_k)^T$-TRR, if the number of nets incident with side i is equal to the capacity $d_i, i = 1, \ldots, k$, i.e.,

$$\sum_{j=1}^{t} |N_j \cap \{i\}| = d_i, i = 1, \ldots, k. \tag{3.2}$$

A routing requirement for a (k, w)-SB is denoted by (k, w)-RR, and a tight routing requirement for a (k, w)-SB is (k, w)-TRR, where parameters k and w are called the dimension and the capacity, respectively. We use k-TRR to denote a (k, w)-TRR for some w.

*Let P be a set of subsets of $\{1, \ldots, k\}$. A $(d_1, \ldots, d_k)^T$-RR $\{N_1, N_2, \ldots, N_t\}$ is called a P-*net *routing requirement if $N_i \in P$ for $i = 1, \ldots, t$. We call P a* net pattern set*. In particular, a $\{S \subseteq \{1, \ldots, k\} | 1 \leq |S| \leq 2\}$-net routing requirement is called a 2-pin net routing requirement, an $\{S \subseteq \{1, \ldots, k\} | 1 \leq |S| \leq s\}$-net routing requirement is called an s-pin net routing requirement, and an $\{S \subseteq \{1, \ldots, k\} | 1 \leq |S| \leq k\}$-net routing require-*

ment is called a multi-pin net routing requirement. [4]

The representation of nets via subsets of side labels is similar to that of edges via subsets of vertices in a hypergraphs. An edge set is a multi-set of subsets of a vertex set. A routing requirement is a multi-set of subsets of side labels. Thus, the representation $\{N_1, N_2, \ldots, N_t\}$ of a routing requirement for a k-sided switch block can be modelled by a hypergraph with vertex set $\{1, \ldots, k\}$ and edge set $\{N_1, N_2, \ldots, N_t\}$.

In particular, a 2-pin net routing requirement corresponds to a 2-graph, an s-pin net routing requirement to an s-graph, and a multi-pin net routing requirement corresponds to a general hypergraph. A multi-pin (k, w)-TRR (k-TRR) corresponds to a w-regular (regular) hypergraph on k vertices. This combinatorial model enables us to use hypergraph theory in the study of routing requirements.

3.2.4 Feasible Routing

A feasible routing for a routing requirement in a switch block is an assignment ON/OFF to the switches, where the set of ON switches realize the requested connections. The edges corresponding to the ON switches in a feasible routing induce a subgraph in the representation graph, when each component corresponds to a net and each component is a tree.

Definition 3.2.3 *Let G be a $(d_1, \ldots, d_k)^T$-SB with graph representation $((V_1, \ldots, V_k), E)$, where $V_i = \{v_{i,j} | j = 1, \ldots, d_i\}, i = 1, \ldots, k$. Let $R = \{N_1, N_2, \ldots, N_t\}$ be a $(d_1, \ldots, d_k)^T$-RR. We say G has (or contains) a*

[4] We note that in the above definition of routing requirements, singleton nets are allowed. In practice, singleton nets do not occur. We introduce the singleton net for the convenience of mathematical manipulations.

feasible routing for R (or G is routable for R, or R is routable in G) if $((V_1, \ldots, V_k), E)$ *has a subgraph T consisting of t components $T_1, T_2, \ldots, T_t$ such that for each $i = 1, \ldots, t$, T_i is a tree of $|N_i|$ vertices and $|V(T_i) \cap V_j| = 1$ for every $j \in N_i$. We call T (or $\{T_1, T_2, \ldots, T_t\}$) a* feasible routing *of R in G, and T_i a routing (or a realization) of net N_i.*

We note that the channel capacity constraint (3.1) is a trivial necessary condition for the existence of a feasible routing. The realization of a singleton net corresponds to a single terminal, or an isolated vertex in the realization of the routing requirement.

Example 3.2.1 Figure 3.1(a) shows a $(4,4)$-SB, where each side has four terminals. Figure 3.1(b) shows a $(4,4)$-RR, which has seven nets: $N_1 = \{1,2\}$, $N_2 = \{1,2,4\}$, $N_3 = \{1,4\}$, $N_4 = \{2,3,4\}$, $N_5 = \{1,3\}$, $N_6 = \{2,3\}$, $N_7 = \{3,4\}$. Net N_2 is a 3-pin net, it requires two switches connecting three terminals on sides 1, 2, and 4, respectively. Figure 3.1(c) shows a feasible routing of the routing requirement.

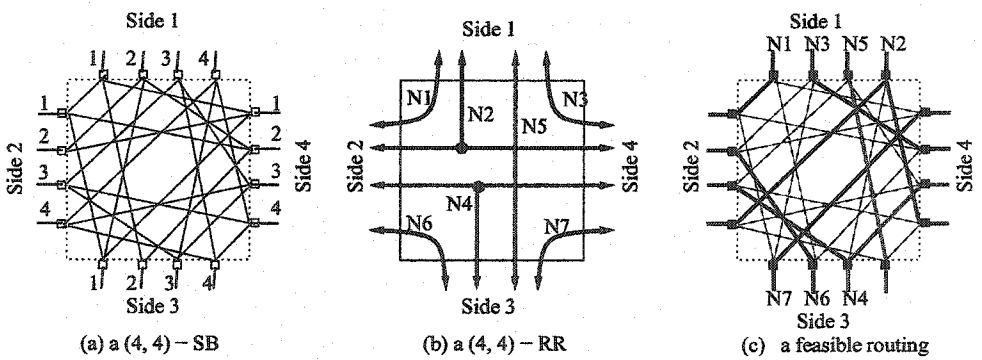


Figure 3.1: An example of a switch block, a routing requirement and a feasible routing.

3.3 Switch Block Design Problems

We consider three types of design specifications. A dimension specification describes the number of sides, a channel capacity specification describes the number of terminals on each side, and a routing capability specification describes a set of routing requirements which must be routable. We have three types of dimension and channel capacity specifications: $(d_1, \ldots, d_k)^T$-SB's, $(dw + c)$-SB's and (k, w)-SB's. There are three types of routing capability descriptions: hyper-universal, universal, and P-universal.

Definition 3.3.1 *A switch block ($(d_1, \ldots, d_k)^T$-SB, $(dw + c)$-SB, (k, w)-SB) is said to be hyper-universal (universal, P-universal) if it is routable for every multi-pin net (2-pin net, P-net) routing requirement ($(d_1, \ldots, d_k)^T$-RR, $(dw+c)$-RR, (k, w)-RR). We write HUSB, USB and P-USB for hyper-universal switch block, universal switch block, and P-universal switch block, respectively.*

A $(d_1, \ldots, d_k)^T$-HUSB (USB, P-USB) is said to be optimal *if it has the minimum number of switches of all $(d_1, \ldots, d_k)^T$-HUSB's (USB's, P-USB's). We denote by $min_{HUSB}(k, w)$ ($min_{USB}(k, w)$) the number of switches in an optimal (k, w)-HUSB ((k, w)-USB).*

According to these definitions, if two net pattern sets P and P' are such that $P \subseteq P'$ then a $(d_1, \ldots, d_k)^T$-P'-USB is also a $(d_1, \ldots, d_k)^T$-P-USB. Therefore, a $(d_1, \ldots, d_k)^T$-HUSB must be a $(d_1, \ldots, d_k)^T$-USB. But the reverse is not necessarily true. In Chapter 6, we will show that the symmetric $(4, w)$-USB from [14] is not a $(4, w)$-HUSB for $w \geq 3$.

Besides the above routing capabilities, we can also consider a switch block routable for a specific set of routing requirements $\mathcal{R}$, called an $\mathcal{R}$-

routable switch block. Table 3.1 summarizes the different types of switch blocks characterized by different dimensions, channel capacities, and routing capability specifications.

	$(d_1, \ldots, d_k)^T$-SB	(k, w)-SB	$(dw + c)$-SB
$\mathcal{R}$: a set of routing requirements	$\mathcal{R}$-routable $(d_1, \ldots, d_k)^T$-SB	$\mathcal{R}$-routable (k, w)-SB	$\mathcal{R}$-routable $(dw + c)$-SB
Hyper-universal	$(d_1, \ldots, d_k)^T$-HUSB	(k, w)-HUSB	$(dw + c)$-HUSB
Universal	$(d_1, \ldots, d_k)^T$-USB	(k, w)-USB	$(dw + c)$-USB
P-universal	$(d_1, \ldots, d_k)^T$-P-USB	(k, w)-P-USB	$(dw + c)$-P-USB

Table 3.1: Different types of switch blocks.

3.3.1 Switch Block Design Problems

Each type of switch block in Table 3.1 is associated with a switch block design problem. The existence of these types of switch blocks is obvious because a complete switch block satisfying the dimension and channel capacity constraints is always $\mathcal{R}$-routable (hyper-universal, universal, P-universal). The problem considered in this thesis is to design an optimal $\mathcal{R}$-routable (hyper-universal, universal, P-universal) switch block. By optimal we mean using a minimum number of switches. The switch block design problems associated with $(d_1, \ldots, d_k)^T$-SB are the problems of designing a single switch block with a minimum number of switches. The switch block design problem associated with a scale variable w is to design a class of optimal switch blocks for all $w \geq 1$. We are particularly interested in such problems due to the requirement of scalability in designs.

The k-HUSB's design problem: Given an integer $k \geq 1$, design an optimal (k, w)-HUSB for every $w \geq 1$.

The k-USB's design problem: Given an integer $k \geq 1$, design an optimal (k, w)-USB for every $w \geq 1$.

The $(dw + c)$-P-USB's design problem: Given a net pattern set P and two k-dimensional nonnegative vectors d and c, design an optimal $(dw + c)$-P-USB for every $w \geq 1$.

We note that a solution scheme for the $(dw+c)$-P-USB's design problem can be used to solve the $(d_1, \ldots, d_k)^T$-P-USB design problem by selecting appropriate nonnegative integer vectors d and c, and a scale integer $w_0 \geq 1$ such that

$$dw_0 + c = (d_1, \ldots, d_k)^T.$$

Once d and c are chosen, we can design a $(dw_0 + c)$-P-USB G using the design method for $(dw + c)$-P-USB's for $w = w_0$. The choice for d and c is not unique. Suppose nonnegative integer vectors d_1 and c_1, and a scale integer $w_1 \geq 1$ also satisfy $d_1 w_1 + c_1 = (d_1, \ldots, d_k)^T$. Then using the design method for $(d_1 w + c_1)$-P-USB's we can get a $(d_1 w_1 + c_1)$-P-USB G_1. Both G and G_1 are $(d_1, \ldots, d_k)^T$-P-USB's. We can choose the one with the smaller number of switches. The problem here is how to choose d and c such that the resulting $(d_1, \ldots, d_k)$-P-USB has the smallest possible number of switches. Our strategy for finding d and c is to maximize w, that is,

$max\{w \mid \text{over all nonnegative integer vectors } d \text{ and } c,\ dw+c = (d_1, \ldots, d_k)^T\}$.

3.4 Problem Simplifications

The main difficulty in designing an optimal hyper-universal switch block or universal switch block is due to the complexity of verifying a design. One reason is that there is usually a great number of routing requirements which need to be checked. Another reason is that it is not easy to find a

feasible routing in an arbitrary switch block for a routing requirement. In this section, we introduce two reductions to reduce the number of routing requirements that need to be checked.

3.4.1 Reductions

Let P be a net pattern set for a $(d_1, \ldots, d_k)^T$-SB. Consider the problem of designing an optimal $(d_1, \ldots, d_k)^T$-P-USB.

Let P' be the net pattern set obtained from P by adding singletons $\{i\}, i = 1, \ldots, k$, i.e., $P' = P \cup \{\{i\} \mid i = 1, \ldots, k\}$. Since a singleton net uses one terminal in a feasible routing, a switch block is a $(d_1, \ldots, d_k)^T$-P-USB if and only if it is a $(d_1, \ldots, d_k)^T$-P'-USB. Therefore, in the following, we always assume that a net pattern set contains singletons $\{i\}$ for $i = 1, \ldots, k$. With this assumption, we know that a tight P-net $(d_1, \ldots, d_k)^T$-RR always exists.

The first reduction says that in designing a hyper-universal (universal, P-universal) switch block we need only consider tight routing requirements. Recall that a tight $(d_1, \ldots, d_k)^T$-RR is one where the number of nets incident with side i is equal to d_i, $i = 1, \ldots, k$.

Let R be a P-net $(d_1, \ldots, d_k)^T$-RR. If R is not tight, then we can add some singletons into R such that the resulting $(d_1, \ldots, d_k)^T$-RR R' is a $(d_1, \ldots, d_k)^T$-TRR. It can be seen that a feasible routing of R' induces a feasible routing of R.

Theorem 3.4.1 *Let P be a net pattern set containing $\{i\}$ for $i = 1, \ldots, k$. A $(d_1, \ldots, d_k)^T$-SB is P-universal (hyper-universal, universal) if and only if it is routable for every P-net (multi-pin, 2-pin) $(d_1, \ldots, d_k)^T$-TRR.*

Proof. Since the set of all P-net (multi-pin net, 2-pin net) $(d_1, \ldots, d_k)^T$-TRR's is a subset of all P-net (multi-pin net, 2-pin net) $(d_1, \ldots, d_k)^T$-RR's, we need only show that a $(d_1, \ldots, d_k)^T$-SB is routable for every P-net (multi-pin net, 2-pin net) $(d_1, \ldots, d_k)^T$-RR provided it is routable for every P-net (multi-pin net, 2-pin net) $(d_1, \ldots, d_k)^T$-TRR.

Suppose that a $(d_1, \ldots, d_k)^T$-SB is routable for every tight P-net (multi-pin net, 2-pin net) $(d_1, \ldots, d_k)^T$-TRR. Let R be any P-net (multi-pin net, 2-pin net) $(d_1, \ldots, d_k)^T$-RR. If R is not tight, i.e, there exists a side i such that the number d'_i of nets in R incident with side i is less than d_i, we add $d_i - d'_i$ copies of singleton $\{i\}$ into R. Continuing this process until a P-net (multi-pin net, 2-pin net) $(d_1, \ldots, d_k)^T$-TRR R' is obtained. By the assumption, R' has a feasible routing F in the $(d_1, \ldots, d_k)^T$-SB. Removing all vertices of F corresponding to the added singletons, we obtain a feasible routing of R in the $(d_1, \ldots, d_k)^T$-SB. Therefore the $(d_1, \ldots, d_k)^T$-SB is routable for every P-net (multi-pin net, 2-pin net) $(d_1, \ldots, d_k)^T$-RR. □

The second reduction says that in designing a hyper-universal (universal) switch blocks we need only consider those routing requirements containing no $\{x\}$ and $\{y\}$ with $x \neq y$.

Definition 3.4.1 *A routing requirement is said to be a* primitive routing requirement, *written* PRR, *if it does not contain two singletons* $\{x\}$ *and* $\{y\}$ *such that* $x \neq y$. *Similarly, we call a hypergraph not containing two singleton edges* $\{x\}$ *and* $\{y\}$ *with* $x \neq y$ *a primitive hypergrpah.*

Let R be a routing requirement containing two singletons $\{x\}$ and $\{y\}$ with $x \neq y$, we call the transformation from R into $(R \setminus \{\{x\}, \{y\}\}) \cup \{\{x, y\}\}$ a *primitive operation* on $\{x\}$ and $\{y\}$. It is clear that a routing

requirement can be transformed to a primitive routing requirement via a some sequence of primitive operations.

Theorem 3.4.2 *A $(d_1, \ldots, d_k)^T$-SB is hyper-universal (universal) if and only if it is routable for every primitive $(d_1, \ldots, d_k)^T$-RR, and if and only if it is routable for every primitive $(d_1, \ldots, d_k)^T$-TRR.*

Proof. Since the set of all multi-pin net (2-pin net) $(d_1, \ldots, d_k)^T$-PRR's is a subset of all multi-pin net (2-pin net) $(d_1, \ldots, d_k)^T$-RR's, we need only show that a $(d_1, \ldots, d_k)^T$-SB is routable for every multi-pin net (2-pin net) $(d_1, \ldots, d_k)^T$-RR if it is routable for every multi-pin net (2-pin net) primitive $(d_1, \ldots, d_k)^T$-PRR.

Suppose that a $(d_1, \ldots, d_k)^T$-SB is routable for every multi-pin net (2-pin net) $(d_1, \ldots, d_k)^T$-PRR. Let R be any multi-pin net (2-pin net) $(d_1, \ldots, d_k)^T$-RR. If R is not primitive, i.e., there exist two singletons $\{x\}$ and $\{y\}$ such that $x \neq y$, we remove an $\{x\}$ and a $\{y\}$ from R and then add a new 2-pin net $\{x, y\}$, i.e., do a primitive operation on $\{x\}$ and $\{y\}$. Continue this process until a multi-pin net (2-pin net) $(d_1, \ldots, d_k)^T$-PRR R' is derived. By the assumption, R' has a feasible routing F in the $(d_1, \ldots, d_k)^T$-SB. Removing all edges of F corresponding to the added 2-pin nets, we obtain a feasible routing of R in the $(d_1, \ldots, d_k)^T$-SB. Therefore the $(d_1, \ldots, d_k)^T$-SB is routable for every multi-pin net (2-pin net) $(d_1, \ldots, d_k)^T$-RR.

Furthermore, by the above argument, we know that, if a $(d_1, \ldots, d_k)^T$-SB is routable for every multi-pin net (2-pin net) $(d_1, \ldots, d_k)^T$-PTRR, then it is routable for every multi-pin net (2-pin net) $(d_1, \ldots, d_k)^T$-TRR, and by Theorem 3.4.1 it is routable for every multi-pin net (2-pin net) $(d_1, \ldots, d_k)^T$-RR. □

By the above two theorems, when designing a hyper-universal switch block or a universal switch block, we need only consider tight routing requirements, or primitive routing requirements, or *primitive tight routing requirements* (*PTRR*). Note that we can not reduce to PRR's for P-USB design in general because when we join two singletons, a new 2-pin net might be added to the net pattern set.

3.4.2 Minimal Routing Requirements

Definition 3.4.2 *A (k, w')-TRR R' is said to be a proper* sub-tight routing requirement *of a (k, w)-TRR R if $R' \subset R$. A (k, w)-TRR R is said to be* minimal *if R does not have a proper sub-tight routing requirement. We write MTRR for "minimal tight routing requirement", MPTRR for "minimal primitive tight routing requirement", and k-MTRR (k-MPTRR) for a (k, w)-MTRR ((k, w)-MPTRR) for some value of w. We say two k-MPTRR's R and R' have the same* type *if their corresponding hypergraphs are isomorphic.*

The concept of k-MTRR is important to our switch block design technique. In terms of hypergraphs, a multi-pin net (k, w)-MTRR (k-MTRR) corresponds to a non-decomposable w-regular (regular) hypergraph on k vertices, and a k-MPTRR corresponds to a non-decomposable primitive regular hypergraphs on k vertices. We will show in Chapter 4 that when k is fixed, the number of multi-pin k-MTRR's is finite. That is, the number of non-decomposable regular hypergraphs on k vertices is finite. Let $f(k)$ denote the maximum number r such that there is a multi-pin net (k, r)-MTRR. Then when $w > f(k)$, a multi-pin net (k, w)-TRR always contains a proper sub-tight routing requirement. In other words, a multi-pin net

(k, w)-TRR can be decomposed into a disjoint union of k-MTRR's. This decomposition property of tight routing requirements plays an important role in our decomposition design technique for k-HUSB's.

3.4.3 Tight Routing Requirement Vectors

In [14], a routing requirement vector is used to represent a 2-pin net routing requirement for a $(4, w)$-SB. Let $n_{i,j}$ denote the multiplicity of $\{i, j\}$ in a $(4, w)$-RR, $1 \leq i < j \leq 4$. Then nonnegative integer vector

$$(n_{1,2}, n_{1,3}, n_{1,4}, n_{2,3}, n_{2,4}, n_{3,4})$$

is called a *routing requirement vector* of the $(4, w)$-RR, written $(4, w)$-RRV. Clearly, a nonnegative integer vector $(n_{1,2}, n_{1,3}, n_{1,4}, n_{2,3}, n_{2,4}, n_{3,4})$ is a $(4, w)$-RRV of a $(4, w)$-RR if and only if it satisfies

$$\begin{cases} n_{1,2} + n_{1,3} + n_{1,4} \leq w \\ n_{1,2} + n_{2,3} + n_{2,4} \leq w \\ n_{1,3} + n_{2,3} + n_{3,4} \leq w \\ n_{1,4} + n_{2,4} + n_{3,4} \leq w \end{cases} \tag{3.3}$$

In [46], a (k, w)-*RRV* for a (k, w)-SB is defined be a nonnegative integer vector

$$(n_{1,2}, \ldots, n_{1,k}, n_{2,3}, \ldots, n_{2,k}, \ldots, n_{k-1,k})$$

satisfying

$$\begin{cases} n_{1,2} + \ldots + n_{1,k} \leq w & \text{(number of nets incident with side 1)} \\ n_{1,2} + \ldots + n_{2,k} \leq w & \text{(number of nets incident with side 2)} \\ \ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ n_{1,k} + \ldots + n_{k-1,k} \leq w & \text{(number of nets incident with side k)} \end{cases} \tag{3.4}$$

Where $n_{i,j}$ is the multiplicity of $\{i,j\}$ in the corresponding 2-pin net (k,w)-RR.

For a 2-pin net (k,w)-TRR with singleton nets are allowed, let s_i denote the multiplicity of singleton $\{i\}, i = 1,\ldots,k$ in the (k,w)-TRR. Then the nonnegative integer vector

$$(s_1,\ldots,s_k,n_{1,2},\ldots,n_{1,k},n_{2,3},\ldots,n_{2,k},\ldots,n_{k-1,k})$$

is a solution to system of linear equations:

$$\begin{cases} s_1 + n_{1,2} + \ldots + n_{1,k} = w \\ s_2 + n_{1,2} + \ldots + n_{2,k} = w \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ s_k + n_{1,k} + \ldots + n_{k-1,k} = w. \end{cases} \tag{3.5}$$

We call nonnegative integer vector

$$(s_1,\ldots,s_k,n_{1,2},\ldots,n_{1,k},n_{2,3},\ldots,n_{2,k},\ldots,n_{k-1,k})^T$$

satisfying (3.5) a *tight* (k,w)*-RRV*, written (k,w)-TRRV. It is obvious that a (k,w)-TRRV corresponds to a (k,w)-TRR.

In general, for a $(d_1,\ldots,d_k)^T$-SB and a given net pattern set $P = \{S_1,S_2,\ldots,S_n\}$. We define the *incidence matrix* of P to be the $k \times n$ matrix $A = (a_{i,j})$ with $a_{i,j} = 1$ if $i \in S_j$ and 0 otherwise. We define a P-net $(d_1,\ldots,d_k)$-*TRRV* to be a nonnegative integer vector $x = (x_1,\ldots,x_n)^T$ satisfying

$$Ax = (d_1,\ldots,d_k)^T \tag{3.6}$$

It is clear that a P-net $(d_1,\ldots,d_k)$-TRRV corresponds to a P-net $(d_1,\ldots,d_k)$-TRR. Therefore, we can derive all P-net $(d_1,\ldots,d_k)$-TRR's via computing all nonnegative integer solutions to (3.6).

For example, consider a $(2,2,3)^T$-SB and net pattern set

$$P = \{\{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\}\}.$$

Then the incidence matrix of P is

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Then the $(2,2,3)^T$-TRR $\{\{1,3\},\{1,3\},\{2,3\},\{2\}\}$ corresponds to the $(2,2,3)^T$-TRRV $x = (0,1,0,0,2,1)^T$, which is a solution to system:

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ 3 \end{pmatrix}$$

3.5 A Design Strategy and an Example

The switch block design strategy for k-HUSB's (k-USB's) in this thesis is designed to solve the following three problems: (1) the enumeration of all multi-pin (2-pin) net (k,w)-PTRR's for every w, which are used to test if a (k,w)-SB is hyper-universal (universal), (2) creating a systematic method to construct a (k,w)-SB with a certain routing capacity for any given w, and (3) creating an algorithm to find a feasible routing for multi-pin (2-pin) (k,w)-RR's in the (k,w)-SB. The following example demonstrates the basic idea of our method.

Example 3.5.1 Designing an optimal $(4, w)$-USB for every $w \geq 1$.

We consider 2-pin net $(4, w)$-PTRR's in designing 4-USB's by Theorem 3.4.2. It can be checked that there are only two non-isomorphic 2-pin net 4-MPTRR's:

$$\{\{1,2\},\{3,4\}\}, \quad \{\{1\},\{1\},\{2,3\},\{3,4\},\{2,4\}\}.$$

Using the above 2-pin net 4-MPTRR's, we can generate all 2-pin net 4-MPTRR's by applying all permutations on $\{1,2,3,4\}$:
$\{\{1,2\},\{3,4\}\}, \{\{1,3\},\{2,4\}\}, \{\{1,4\},\{2,3\}\}$,
$\{\{1\},\{1\},\{2,3\},\{3,4\},\{2,4\}\}, \{\{2\},\{2\},\{1,3\},\{3,4\},\{1,4\}\}$,
$\{\{3\},\{3\},\{1,2\},\{2,4\},\{1,4\}\}, \{\{4\},\{4\},\{1,2\},\{2,3\},\{1,3\}\}$.

Therefore every 2-pin net $(4, w)$-PTRR can be decomposed into the disjoint union of the above 2-pin net 4-MPTRR's. In other words, all 2-pin net $(4, w)$-PTRR's can be obtained by combining the above 2-pin net $(4, 2)$-MPTRR's and 2-pin net $(4, 1)$-MPTRR's. Thus we can use the above $(4, 2)$-MPTRR's and $(4, 1)$-MPTRR's to generate all 2-pin net $(4, w)$-PTRR's for every w.

Since a 2-pin net $(4, w)$-PTRR can be decomposed into a multi-set of 2-pin net $(4, 1)$-MPTRR's and 2-pin net $(4, 2)$-MPTRR's, when w is even the number of 2-pin net $(4, 1)$-MPTRR's must be even, so pairing these $(4, 1)$-MPTRR's up and also including the existing $(4, 2)$-MPTRR's, we obtain $\frac{w}{2}$ 2-pin net $(4, 2)$-PTRR's. When w is odd, the number of 2-pin net $(4, 1)$-MPTRR's must be odd; pair these $(4, 1)$-MPTRR's up leaving one $(4, 1)$-MPTRR, so in total we obtain $\frac{w-1}{2}$ 2-pin net $(4, 2)$-PTRR's and one 2-pin net $(4, 1)$-PTRR. Hence the following *decomposition property* holds.

Proposition 3.5.1 *A 2-pin net $(4, w)$-PTRR can be decomposed into $\frac{w}{2}$ 2-*

pin net $(4,2)$*-PTRR's when* w *is even, and into* $\frac{w-1}{2}$ *2-pin net* $(4,2)$*-PTRR's and one 2-pin net* $(4,1)$*-PTRR when* w *is odd.*

Because of the above decomposition property, we can start by designing a $(4,2)$-USB and a $(4,1)$-USB. When w is even, a $(4,w)$-USB can be obtained by taking a disjoint union of $\frac{w}{2}$ copies of the $(4,2)$-USB because a 2-pin net $(4,w)$-PTRR can be decomposed into $\frac{w}{2}$ 2-pin net $(4,2)$-PTRR's, each of which has a feasible routing in one of the $\frac{w}{2}$ copies of the $(4,2)$-USB. Similarly, when w is odd, a disjoint union of $\frac{w-1}{2}$ copies of the $(4,2)$-USB and one copy of the $(4,1)$-USB gives a $(4,w)$-USB. We call the $(4,1)$-USB and $(4,2)$-USB *prime 4-USB's*, and a $(4,w)$-USB constructed from the prime 4-USB's a *compound k-USB.*

Figure 3.2 shows a prime $(4,1)$-USB G_1 and a prime $(4,2)$-USB G_2. It can be seen that G_1 is a $(4,1)$-USB as it is a complete $(4,1)$-SB. G_1 is an optimal $(4,1)$-USB because there must be at least one switch between any two sides. However, it is not obvious that G_2 is an optimal $(4,2)$-USB. We will show G_2 is an optimal $(4,2)$-HUSB in Chapter 6. Chang et al. [14] first gave an optimal $(4,2)$-USB, which is isomorphic to G_2. In [14], it was also proved that $6w$ is a lower bound for the number of switches in a $(4,w)$-USB. Since G_1 has six switches and G_2 has twelve switches, when w is even, $\frac{w}{2}$ copies of G_2 gives a $(4,w)$-USB with $6w$ switches, so it is optimal. When w is odd, the disjoint union of $\frac{w-1}{2}$ copies of G_2 and one G_1 gives a $(4,w)$-USB with $6w$ switches, so it is optimal.

Next we use an example to illustrate how to find a feasible routing for a given 2-pin net $(4,w)$-RR in a compound $(4,w)$-USB. Consider the compound $(4,4)$-USB obtained by taking the union of two copies of G_2. Figure 3.3(b) shows the $(4,4)$-USB obtained by two copies of G_2. Given

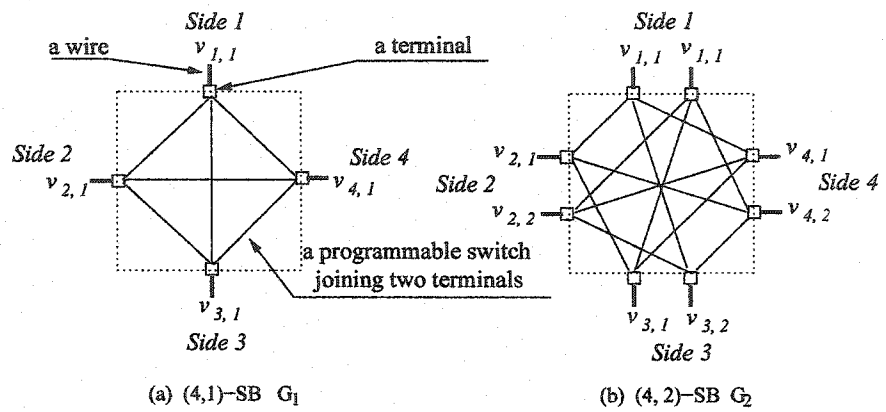


Figure 3.2: Prime universal 4-sided switch blocks.

a 2-pin net $(4,4)$-RR, R, for instance, $R = \{\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,3\}, \{3,4\}\}$, we find a feasible routing of R in the $(4,4)$-USB as follows.

Step 1. Convert R to a 2-pin net $(4,4)$-TRR:
$\{\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,3\}, \{3,4\}, \{1\}, \{2\}, \{4\}, \{4\}\}$.

Step 2. Convert the obtained 2-pin net $(4,4)$-TRR to a $(4,4)$-PTRR by a primitive operation: substituting $\{1\}, \{2\}$ by $\{1,2\}$, we obtain
$\{\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,3\}, \{3,4\}, \{1,2\}, \{4\}, \{4\}\}$.

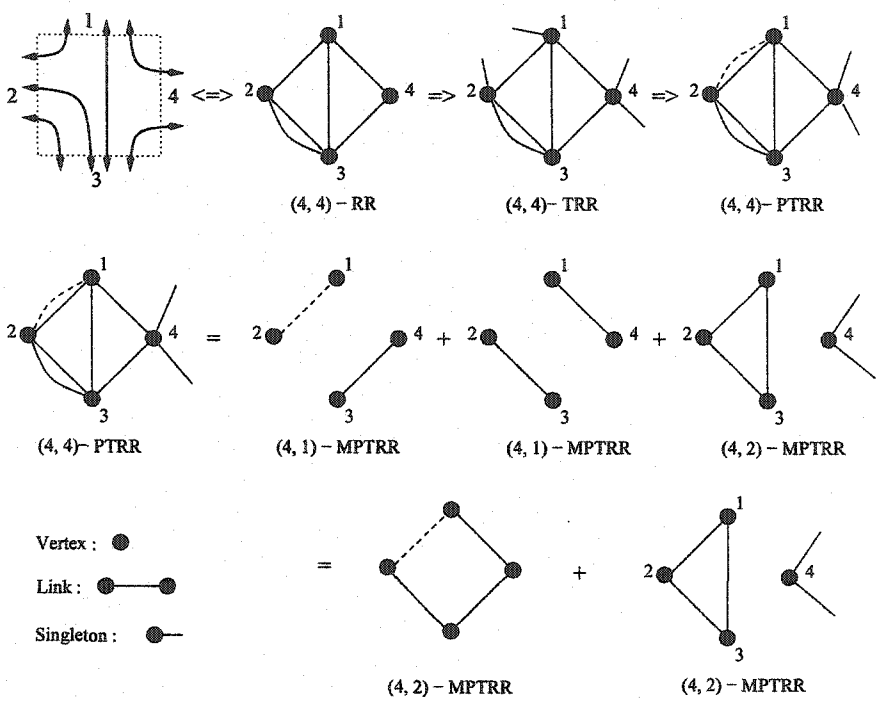
Step 3. Decompose the 2-pin net $(4,4)$-PTRR into 2-pin net 4-MPTRR's:
$\{\{1,2\}, \{3,4\}\} \cup \{\{1,4\}, \{2,3\}\} \cup \{\{1,2\}, \{2,3\}, \{1,3\}, \{4\}, \{4\}\}$.

Step 4. Combining the first two 2-pin net $(4,1)$-MPTRR's, we obtain two 2-pin net $(4,2)$-PTRR's:
$R_1 = \{\{1,2\}, \{3,4\}, \{1,4\}, \{2,3\}\}$,
$R_2 = \{\{1,2\}, \{2,3\}, \{1,3\}, \{4\}, \{4\}\}$.

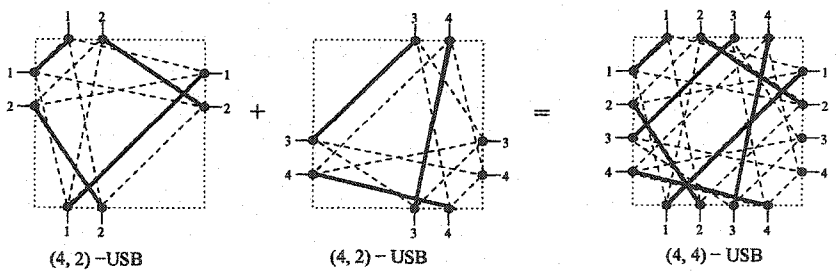
Step 5. Find a feasible routing for R_1 in the first copy of G_2 and a feasible routing for R_2 in the second copy of G_2 in the compound $(4,4)$-USB.

Step 6. Remove the switches and singletons corresponding to the added singletons and edges in Steps 1 and 2. A feasible routing of R in the compound $(4,4)$-USB is obtained.

Figure 3.3(a) shows the above transformations of these routing requirements from steps 1 to 4. Figure 3.3(b) shows the disjoint union of two copies of G_2 and the corresponding feasible routings of R_1 and R_2 in G_2.



(a) transformation and decomposition of MPTRRs.



(b) a compound (4, 4)–USB and a feasible routing.

Figure 3.3: An example of a compound $(4, 2)$-USB and a feasible routing.

This example illustrates how we solve the three problems at the beginning of this section. In summary, the scheme of designing for 4-USB's is as follows.

(1) Compute all 2-pin net 4-MPTRR's, which are used to generate all 2-pin net $(4, w)$-PTRR's for every w by combination, determine the channel capacities for prime 4-USB's. In this case, the channel capacities are 2 and 1, so prime 4-USB's are a $(4, 2)$-USB and a $(4, 1)$-USB.

(2) Design a prime $(4, 2)$-USB and a $(4, 1)$-USB and set up a *routing table* for the $(4, 2)$-USB recording the feasible routings of all 2-pin net $(4, 2)$-PTRR's in the $(4, 2)$-USB, and a routing table for the $(4, 1)$-USB recording the feasible routings of all 2-pin net $(4, 1)$-PTRR's in the $(4, 1)$-USB.

(3) Construct a compound $(4, w)$-USB for every $w \geq 1$ by a disjoint union of $\lfloor \frac{w}{2} \rfloor$ copies of the prime $(4, 2)$-USB (plus a prime $(4, 1)$-USB when w is odd).

(4) Find a feasible routing for a 2-pin net $(4, 4)$-RR R in the compound $(4, 4)$-USB by converting R to a 2-pin net $(4, 4)$-PTRR R', and then decomposing R' to 2-pin net 4-MPTRR's, composing them into $\lfloor \frac{w}{2} \rfloor$ 2-pin net $(4, 2)$-PTRR's (plus a $(4, 1)$-PTRR when w is odd), and then finding a feasible routing for each of the 2-pin net $(4, 2)$-PTRR's in one copy of the $(4, 2)$-USB and $(4, 1)$-USB by the routing tables.

We note that the 4-USB design problem was first studied by Chang et al. in [14]. They gave a $(4, w)$-SB generation algorithm and proved that the $(4, w)$-SB $M_{4,w}$ generated by the algorithm is a union of $\lfloor \frac{w}{2} \rfloor$ $(4, 2)$-SB's (plus a $(4, 1)$-SB when w is odd) and that $M_{4,w}$ is an optimal universal $(4, w)$-USB.

3.6 Conclusions and Future Work

In this chapter, using combinatorial models we gave definitions of a switch block, a track-free net, a routing requirement, and a feasible routing. We examined various classes of switch blocks and switch block design problems with respect to dimension, channel capacity and routing capacity specifications. We introduced the new concepts of irregular $(d_1, \ldots, d_k)^T$-SB's, $(dw + c)$-SB's and P-universal switch blocks, which are more general than the previously studied regular switch blocks, i.e., the universal (k, w)-SB's and hyper-universal (k, w)-SB's.

We also gave an example showing how we design universal $(4, w)$-SB's. We called our design method a *decomposition design scheme.* In this dissertation, we are going to generalize the decomposition design scheme to tackle the general P-universal $(dw + c)$-SB's design problem.

Our current model is based on pass transistor switches, which are bidirectional switches. However, unidirectional switches such as tri-state buffer switches are often used in practice. It is of practical value to develop a combinatorial model for nets, routing requirements and feasible routings for unidirectional switch based switch blocks. Combinatorial models for track-fixed nets and for mixed nets would also be interesting. Developing combinatorial models for specific switch modules such as switch matrices would also be a future work.

Historical notes:

Y. L. Wu introduced his work on switch block designs to me when I was

working in his research group at the Chinese University of Hong Kong in 1998. He wanted to find a feasible routing algorithm for the switch blocks given in [37], which are $(4, w)$-SB's routable for all routing requirements with tracks fixed on a given side. I found that finding a feasible routing for an arbitrary routing requirement with tracks fixed on a side in the switch blocks was not an easy task, so I considered an alternative switch module consisting of a complete $(2, w)$-SB attached to a side of a $(4, w)$-SB. A feasible routing can be found efficiently in this switch module provided the $(4, w)$-SB is routable for every track-free routing requirement and a feasible routing can be found easily. The problem is how to design such a $(4, w)$-HUSB with a small number of switches. In the summer of 1999, J. Liu invited me to visit him at the University of Lethbridge for a month. I proposed the problem to him. He was interested in the problem and we began working on it together.

We formally modelled a (k, w)-SB as a k-partite graph, a net as a subset of side labels, a routing requirement (previously called a global routing) as a multi-set of subsets corresponding to a regular hypergraph, and a feasible routing (used to be called a detailed routing) as a spanning forest with each component corresponding to a net, and the switch block (previously called a switch box) design problem as a graph design problem. Using these models, some progress on switch block designs was made. I proved that the number of minimal routing requirements is finite, and gave all 3-MPTRRs and 4-MPTRRs. J. Liu investigated the global routing from the point of 1-designs and hypergraphs, and gave a necessary condition for a global routing being minimal.

Furthermore I designed optimal $(3, w)$-HUSB's and approximate $(4, w)$-

HUSB's. I mentioned these results to Y. L. Wu after I started my graduate studies at the University of Victoria. He was interested in these results, and encourage me to generalize the method for $(4, w)$-HUSB to (k, w)-HUSB's. I proved that the method works when k is fixed, and it lead to a (k, w)-HUSB with a linear number of switches and an efficient feasible routing algorithm. Y. L. Wu suggested submitting a joint paper to ICCAD 2000 based on the results on (k, w)-HUSB's, $(3, w)$-HUSB's and $(4, w)$-HUSB's. J. Liu agreed and the three of us began to work together on this topic. Y. L. Wu did the background investigation. He noticed the work on universal switch blocks done by Chang et al. [14], and named our switch block "a hyper-universal switch box". The notion "decomposition design scheme" was previous called "reduction" design scheme. Our joint paper [22] was accepted to IEEE/ ACM International Conference on Computer-Aided Design (ICCAD), and I presented the paper in Nov. 2000 in San Jose. Our second joint paper [24] with a better $(4, w)$-HUSB design and experiments was accepted to 38th IEEE/ACM Design Automation Conference. Y. L. Wu and C. C. Cheung contributed the experimental results. I attended and presented the paper in June 2001 in Las Vegas.

The modelling for the irregular switch blocks and $(dw + c)$-SB's and the concept of P-universal switch blocks were done after my Ph.D candidacy exam, and were first proposed in this dissertation. The notion of a routing requirement vector (RRV) was first introduced by Chang et al. in [14]. In order to make use of linear Diophantine equations, I modified the routing vector to be a tight routing requirement vector in this dissertation. The computability of all k-MPTRR's for any fixed k is solved in this this dissertation.

Chapter 4

The Enumeration of Routing Requirements

In this chapter, we examine the problem of enumerating the set of all tight routing requirements for some types of switch blocks. This problem is important to our switch block design approach. We use the complete set of tight routing requirements to verify that a switch block design satisfies the required routing capability. We also use the decomposition property of tight routing requirements to construct compound switch blocks. Section 4.1 discusses systems of linear Diophantine equations, and Section 4.2 illustrates how we use the set of minimal solutions to a system of linear Diophantine equations to generate tight routing requirement vectors. In Section 4.4, we give a combinatorial approach to enumerate k-MPTRR's and then compute all k-MPTRR's for $k \leq 4$, which will be used in Chapters 5 and 6.

4.1 Systems of Linear Diophantine Equations

A system of linear equations, $Ax = b$, is called a *system of linear Diophantine equations* if A is an $m \times n$ integer matrix and b is an m-dimensional integer column vector and integer solutions $x = (x_1, \ldots, x_n)^T$ are requested. If $b = (0, \ldots, 0)^T$, the system is *homogeneous* [18, 30]. A solution $x = (x_1, \ldots, x_n)^T$ to $Ax = b$ is *non-zero* if $x \neq (0, \ldots, 0)^T$, i.e., at least one of x_i's is not equal to 0, and a *nonnegative integer solution* if all x_i's are nonnegative integers. In the following, unless stated otherwise, every system of linear equations considered is a system of linear Diophantine equations,

and every solution considered is a non-zero, nonnegative integer solution.

Let $x = (x_1, \ldots, x_n)^T$ and $y = (y_1, \ldots, y_n)^T$ be two vectors. Define $x \preceq y$ if $x_i \leq y_i$ for all $i = 1, \ldots, n$. A solution y to a system of linear Diophantine equations is said to be *minimal* (or a *minimal solution*) if there is no other solution x such that $x \preceq y$. The set of *minimal solutions* to a homogenous system of linear Diophantine equations is also called the *Hilbert basis* of the system of linear Diophantine equations.

Let x be a solution to $Ax = b$ and x' be a minimal solution to $Ax = b$ with $x' \preceq x$. Then $x - x'$ is a solution to $Ax = 0$. [1] Since $x = x' + (x - x')$, the solution x can be expressed as a sum of a minimal solution to $Ax = b$ and a solution to the homogenous system $Ax = 0$. A solution y to $Ay = 0$ can be expressed as a nonnegative linear combination of minimal solutions to $Ay = 0$ because if y is not a minimal solution to $Ay = 0$, then there is a minimal solution y' such that $y' \preceq y$ and $y' \neq y$. Hence $y = y' + (y - y')$ and $y - y'$ is a non-zero nonnegative integer solution to $Ay = 0$. If $y - y'$ is not a minimal solution, then there is a minimal solution y'' such that $y'' \preceq y - y'$. Then $y = y' + y'' + (y - y' - y'')$ and $y - y' - y''$ is a solution to $Ay = 0$. If $(y - y' - y'')$ is not a minimal solution, then continue until a minimal solution is derived. This process must stop in a finite number of steps as the integer solutions keep decreasing strictly and they are nonnegative. Therefore, any solution to $Ax = b$ can be expressed as the sum of a minimal solution to $Ax = b$ and some minimal solutions to the homogenous system $Ay = 0$.

[1] In the expression $Ax = 0$, the right side 0 is an abbreviation to $(0, \ldots, 0)^T$. We use 0 to denote a zero vector, i.e., a vector will all components equal to zero.

4.1.1 Finiteness of the Hilbert Basis

A system of linear Diophantine equations may have an infinite number of non-zero non-negative integer solutions. However, the set of non-zero non-negative **minimal** integer solutions to a system of linear Diophantine equations is a finite set. This can be derived from the well-quasi-ordered set theory. A *partially ordered set* (or *poset*) $(Q, \leq)$ is *well-quasi-ordered* if there is neither an infinite *descending chain* nor an infinite *anti-chain*, or, equivalently, for any infinite sequence $x_i, i = 1, 2, \ldots$ with $x_i \in Q$, there exist x_i and x_j such that $x_i \leq x_j$ and $i < j$. Since the set of minimal elements forms an anti-chain, **a well-quasi-ordered set always has a finite number of minimal elements.** See [29, 20] for more detailed descriptions about the properties of well-quasi-ordered sets.

For example, if $\mathcal{Z}$ denotes the set of nonnegative integers, then $(\mathcal{Z}, \leq)$ is a well-quasi-ordered set. This is because for any infinite sequence of nonnegative integers $x_i, i = 1, 2, \ldots$, if the integers in the sequence are unbounded, then there must exist an x_j such that $x_1 < x_j$ and $1 < j$; otherwise, there must exist x_i and x_j such that $x_i = x_j$ and $i < j$.

Lemma 4.1.1 (Higman [29, 20]) *Let $(Q, \leq)$ be a well-quasi-ordered set and Q^* be the set of all finite sequences formed by elements from Q. For $\mathbf{a} = [a_1, a_2, \ldots, a_n] \in Q^*$ and $\mathbf{b} = [b_1, b_2, \ldots, b_m] \in Q^*$, define $\mathbf{a} \preceq \mathbf{b}$ if $\mathbf{b}$ has a subsequence $b_{i_1}, \ldots, b_{i_n}$ such that $i_1 < i_2 < \ldots < i_n$ and $a_j \leq b_{i_j}$ for $j = 1, \ldots, n$. Then the partially ordered set $(Q^*, \preceq)$ is a well-quasi-ordered set.*

By Lemma 4.1.1 and the fact that $(\mathcal{Z}, \leq)$ is well-quasi-ordered, we know $(\mathcal{Z}^*, \preceq)$ is well-quasi-ordered. Let $\mathcal{Z}^n = \{(x_1, \ldots, x_n)^T | x_i \in \mathcal{Z}\}$. By the

fact that a subset of a well-quasi-ordered set is also well-quasi-ordered with ordering of the super set, we have $(\mathcal{Z}^n, \preceq)$ is well-quasi-ordered since $\mathcal{Z}^n \subseteq \mathcal{Z}^*$. Note that $[x_1, \ldots, x_n] \preceq [y_1, \ldots, y_n]$ if and only if $x_i \leq y_i$ for all $i = 1, \ldots, n$, i.e., $(x_1, \ldots, x_n)^T \preceq (y_1, \ldots, y_n)^T$.

Theorem 4.1.2 *The set of minimal solutions to a system of linear Diophantine equations is a finite set. The Hilbert basis of a homogenous linear Diophantine equations is a finite set.* [2]

Proof. Let $Ax = b$ be a system of linear Diophantine equations with $x = (x_1, \ldots, x_n)^T$. Then the set S of all non-zero nonnegative integer solutions to $Ax = b$ is a subset of $\mathcal{Z}^n$. Therefore $(S, \preceq)$ is a well-quasi-ordered set, and so that $(S, \preceq)$ has a finite number of minimal elements. That is, the system of linear Diophantine equations $Ax = b$ has a finite number of minimal solutions with respect to ordering $\preceq$. Similarly, there is a finite number of minimal solutions to a homogenous system of linear Diophantine equations. That is, the Hilbert basis of a homogenous system of linear Diophantine equations is a finite set. □

4.1.2 Enumerating the Set of Minimal Solutions

There are several known algorithms for computing the set of minimal solutions to a system of linear Diophantine equations. Contejean and Devie [18] surveyed some of them and gave an incremental algorithm. Pasechnik [38] recently presented another algorithm. Contejean and Devie's algorithm is simple and easy to implement. It can be described as follows.

[2] This theorem must have been proved a long time ago. We include it for self-containment of the thesis.

Let $Ax = 0$ be a homogenous system of linear Diophantine equations, where A is an $m \times n$ integer matrix. Let e_i denote the n-dimensional unit vector, i.e., the i-th component is 1 and the other components are 0.

Hilbert basis Algorithm

Input: $Ax = 0$

$\mathcal{P} := \{e_1, \ldots, e_n\}$;

$\mathcal{B} := \emptyset$;

while $\mathcal{P} \neq \emptyset$ **do**;

$\mathcal{B} := \mathcal{B} \cup \{x \in \mathcal{P} \mid Ax = 0\}$;

$\mathcal{L} := \{x \in \mathcal{P} \setminus \mathcal{B} \mid \forall s \in \mathcal{B}, x \not\geq s\}$;

$\mathcal{P} := \{x + e_i \mid x \in \mathcal{L}, (Ax)^T(Ae_i) < 0\}$;

end while

Output $\mathcal{B}$.

Even though the above algorithm looks simple, the proof of termination of the algorithm is complicated [18]. The above algorithm can be used to compute the set of all minimal solutions to $Ax = b$ by first applying the algorithm to compute all minimal solutions to the homogenous system $Ax - by = 0$ (i.e., $(A, b)(x^T, y)^T = 0$), and then choosing those solutions $(x^T, y)^T$ with $y = 1$.

4.2 Computing Tight Routing Requirement Vectors

In this section, we use a system of linear Diophantine equations to compute the set of all tight routing requirement vectors (TRRV) for a given type of switch block. Suppose a given a channel capacity vector $(d_1, \ldots, d_k)^T$

and a net pattern set $P = \{S_1, S_2, \ldots, S_n\}$, where $S_i \subseteq \{1, 2, \ldots, k\}$ and $\{\{i\} | i = 1, \ldots, k\} \subseteq P$. The problem is to compute all P-net $(d_1, \ldots, d_k)^T$-TRRV's. This problem is equivalent to solving a corresponding system of linear Diophantine equations (3.6), i.e.,

$$Ax = (d_1, \ldots, d_k)^T \tag{4.1}$$

where $A = (a_{i,j})_{k \times n}$ is the incidence matrix of P with $a_{i,j} = 1$ if $i \in S_j$ and 0 otherwise.

From the discussion in Section 4.1, we know that all P-net $(d_1, \ldots, d_k)^T$-TRRV's can be computed by determining all nonnegative integer solutions to the above system of linear equations.

4.2.1 Tight Routing Requirement Vectors for a Class of Switch Blocks

Let d and c be two k-dimensional nonnegative integer column vectors. In designing $(dw+c)$-P-USB's, we need to compute all P-net $(dw+c)$-TRRV's for every integer $w \geq 1$. We have seen that, for any fixed w, a $(dw + c)$-TRRV corresponds to a nonnegative integer solution $x = (x_1, \ldots, x_n)^T$ to the following system of linear Diophantine equations:

$$Ax = dw + c \tag{4.2}$$

However, solving (4.2) for every w is not effective. We use an alternative approach based on the Hilbert basis of a homogenous system of linear Diophantine equations.

In the following, we take both x and w as unknown variables and consider the following system of linear Diophantine equations

$$Ax - dw = c, \quad \text{i.e., } (A, -d)(x^T, w)^T = c \tag{4.3}$$

and its homogenous systems

$$Ax - dw = 0, \quad \text{i.e., } (A, -d)(x^T, w)^T = 0 \tag{4.4}$$

where $(x^T, w)^T$ is the column vector $(x_1, \ldots, x_n, w)^T$. By the arguments in Section 4.1, if the Hilbert basis of (4.4) is known, then for any fixed w, say $w = w_0$, we can use the Hilbert basis to generate all solutions of $Ax - dw_0 = 0$ by nonnegative linear combinations of the minimal solutions as follows. Suppose that the Hilbert basis of (4.4) consists of $(x_1^T, w_1)^T, \ldots, (x_t^T, w_t)^T$. Solve the linear Diophantine equation

$$w_1 y_1 + w_2 y_2 + \ldots + w_t y_t = w_0. \tag{4.5}$$

For each solution $(y_1, y_2, \ldots, y_t)^T$ to (4.5), $\sum_{j=1}^{t} y_j (x_j^T, w_j)^T$ is a solution to $Ax - dw_0 = 0$. Since a solution to $Ax - dw_0 = 0$ can always be decomposed into the union of minimal solutions, we can generate all solutions to $Ax - dw_0 = 0$. But the same solution may be generated more than once because the decomposition is not unique.

We have seen that every solution to $Ax - dw = c$ can be expressed as a sum of a minimal solution to $Ax - dw = c$ and a solution to $Ax - dw = 0$. The latter can be expressed as a nonnegative linear combination of minimal solutions to $Ax - dw = 0$. Therefore we can use the set of minimal solutions to $Ax - dw = c$ and the Hilbert basis of $Ax - dw = 0$ to generate all solutions to $Ax - dw = c$.

For a fixed $w = w_0$, to generate all solutions to $Ax - dw_0 = c$, we can find all the minimal solutions to $Ax - dw = c$, $X_j = (x_{j_1}, \ldots, x_{j_n}, w_j)^T, j = 1, \ldots, l$ with $w_j \leq w_0$ from the set of minimal solutions to $Ax - dw = c$. Then for each w_j, we can generate all solutions to $Ax - d(w_0 - w_j) = 0$

from the Hilbert basis of $Ax - dw = 0$ using the method described earlier. For each of them by adding X_j, we obtain a solution to $Ax - dw_0 = c$.

In summary, by using a Hilbert basis algorithm, we can compute the sets of all minimal solutions to (4.3) and to (4.4), and then use them to enumerate all P-net $(dw + c)$-TRRV's for every integer $w \geq 1$. Since a P-net $(dw + c)$-TRRV corresponds to a P-net $(dw + c)$-TRR, all P-net $(dw + c)$-TRR's can be enumerated via the above generation method.

In particular, when $d = (1, \ldots, 1)^T$ and $c = (0, \ldots, 0)^T$ and $P = \{S \mid 1 \leq |S| \leq k\}$, a minimal solution to $Ax - dw = 0$ corresponds to a multi-pin net k-MTRR, i.e., a non-decomposable regular hypergraph on k vertices. Therefore, for a fixed k, and using the algebraic method described above, we can enumerate the set of multi-pin net k-MTRR's, and then all k-MPTRR's (note that the set of k-MPTRR's is a subset of k-MTRR's). This solves the enumerability of the set of multi-pin net k-MTRR's (k-MPTRR's) for any fixed k.

4.3 Enumerating Minimal Routing Requirements

In this section, we discuss the enumeration of k-MPTRR's via combinatorics. In this method, the maximum degree of a non-decomposable hypergraph on k vertices is useful for bounding the search space.

Denote by $\mathcal{G}_s(k)$ the set of all s-pin net k-MPTRR's, where $2 \leq s \leq k$. We write $\mathcal{G}(k)$ for $\mathcal{G}_k(k)$, which is the set of multi-pin k-MPTRR's. When $s = 2$, $\mathcal{G}_2(k)$ is the set of 2-pin net k-MPTRR's. Since the set of s-pin net k-MTRR's corresponds to the set of minimal solutions to a corresponding system of linear Diophantine equations and the set of k-MPTRR's is a subset of k-MTRR's, by Theorem 4.1.2, we have the following theorem.

Theorem 4.3.1 *For any fixed k and $2 \leq s \leq k$, the number of s-pin net k-MTRR's is finite, and the number of all s-pin net k-MPTRR's is finite, i.e., $\mathcal{G}_s(k)$ is a finite set.*

Definition 4.3.1 *Define the function*

$$f_s(k) = \max\{\ r \mid \textit{there exists an } s\textit{-pin net } (k,r)\textit{-MPTRR}\}$$

In particular, we define

$$f(k) = \max\{\ r \mid \textit{there exists a multi-pin net } (k,r)\textit{-MPTRR}\},$$

i.e., $f(k) = f_k(k)$.

Since a primitive operation does not change the minimality of a tight routing requirement, applying a sequence of primitive operations to a minimal tight routing requirement yields a primitive tight routing requirement. So we have the following lemma.

Lemma 4.3.2 $f_s(k) = \max\{\ r \mid \textit{there exists an } s\textit{-pin net } (k,r)\textit{-MTRR}\}$

From the above definitions we have:

$$\mathcal{G}_2(k) \subseteq \mathcal{G}_3(k) \subseteq \ldots \subseteq \mathcal{G}_k(k) = \mathcal{G}(k),$$

$$f_2(k) \leq f_3(k) \leq \ldots \leq f_k(k) = f(k).$$

It is clear that $f_s(k)$ is equal to the maximum value of degrees of all regular s-graphs on k vertices. Since $\mathcal{G}_s(k)$ is a finite set by Theorem 4.3.1, $f_s(k)$ is well-defined and finite.

Theorem 4.3.3 *An s-pin net (k, r)-PTRR with $r > f_s(k)$ can always be decomposed into a disjoint union of s-pin net k-MPTRR's of capacities at most $f_s(k)$. In other words, all s-pin net (k, w)-PTRR's can be generated from s-pin net k-MPTRR's via disjoint unions.*

The value of $f(k)$ is useful in enumerating all multi-pin net k-MPTRR's and in designing k-HUSB's. We want to know the value of $f(k)$. P. Haxell proved that $f(k)$ has a super-polynomial lower bound [21].

Theorem 4.3.4 (P. Haxell, [21]) *There exists a constant $C > 1$ such that $f(k) \geq C^{\sqrt{k \log k}}$ for infinitely many values of k.*

We have been looking for an upper bound for $f(k)$. We recently found such a bound by using Pottier's upper bound on the sum of the components of a minimal solution $x = (x_1, \ldots, x_q)^T$ to a system of linear Diophantine equations $Ax = 0$ [41]:

$$\sum_{i=1}^{q} x_i \leq \prod_{i=1}^{p} (\sum_{j=1}^{q} |a_{i,j}| + 1), \tag{4.6}$$

where $A(a_{i,j})$ is a $q \times p$ integer matrix. By definition, $f(k)$ is maximum w such that there is a non-negative integer vector x such that $(x^T, w)^T$ is a minimal solution to

$$Bx - (1, \ldots, 1)^T w = 0 \tag{4.7}$$

where B is the incidence matrix of net pattern set $\{S \subseteq \{1, \ldots, k\} | 1 \leq |S| \leq k\}$. Since $(B, -(1, \ldots, 1)^T)$ is a $k \times 2^k$ matrix, the sum of the absolute values of each row is $2^{k-1} + 1$. Then the right side of (4.6) is $(2^{k-1} + 1)^k$. Consider a minimal solution to (4.7) with the last component equal to $f(k)$, we obtain

$$f(k) \leq (2^{k-1} + 1)^k. \tag{4.8}$$

Once $f(k)$ or an upper bound $b(k) \geq f(k)$ is known, we can enumerate all non-decomposable r-regular hypergraphs for $r = 1, \ldots, f(k)$, or for $r = 1, \ldots, b(k)$. This guarantees that we can derive all k-MPTRR's.

4.4 Minimal Routing Requirements of Small Dimensions

In this section, we enumerate all multi-pin net k-MPTRR's for $k \leq 4$. They will be used to design 2-HUSB's and 3-HUSB's in Chapter 5, and 4-HUSB's in Chapters 6. As in the last section, we useed $\mathcal{G}(k)$ to denote the set of all multi-pin net k-MPTRR's and $f(k)$ to denote the maximum r such that there is a multi-pin net (k, r)-MPTRR, i.e., $f(k)$ is the maximum value of degrees of all non-decomposable regular hypergraphs on k vertices. For convenience, we omit the notion "multi-pin net" in this section.

It is obvious that there is only one 1-MPTRR, i.e., $\{\{1\}\}$, so $\mathcal{G}(1) = \{\{\{1\}\}\}$ and $f(1) = 1$. It is also obvious that there is only one 2-MPTRR, i.e., $\{\{1, 2\}\}$. Therefore, $\mathcal{G}(2) = \{\{\{1, 2\}\}\}$ and $f(2) = 1$.

For $k = 3$, since $\{\{1, 2\}, \{2, 3\}, \{1, 3\}\}$ is a $(3, 2)$-MPTRR, $f(3) \geq 2$. We show that $f(3) \leq 2$. Suppose, on the contrary, that there is a $(3, r)$-MPTRR, say R, with $r \geq 3$. Then R does not contain $\{1, 2, 3\}$ because $\{\{1, 2, 3\}\}$ is a $(3, 1)$-TRR. We may assume that R contains a minimum number of subsets of all $(3, r)$-MPTRR's. By this assumption, we know that R does not contain two subsets N_1 and N_2 with $N_1 \cap N_2 = \emptyset$ because otherwise $(R \setminus \{N_1, N_2\}) \cup \{N_1 \cup N_2\}$ would be a $(3, r)$-MPTRR with a smaller number of subsets. Hence if R contains a subset of size 1, say $\{1\}$, then none of $\{2\}, \{3\}, \{2, 3\}$ is contained in R. Therefore R contains only the subsets of $\{1, 2\}, \{1, 3\}$ and $\{1\}$. It is clear that R could not be a $(3, r)$-PTRR because the number of subsets containing 1 in R is greater than the

$$\{\{1\},\{2,3\}\},\{\{2\},\{1,3\}\},$$
$$\{\{3\},\{1,2\}\},\{\{1,2,3\}\},$$
$$\{\{1,2\},\{2,3\},\{1,3\}\}$$

Table 4.1: A list of all 3-MPTRR's.

$$GR_1^1 = \{\{1,2,3,4\}\},$$
$$GR_2^1 = \{\{1,2\},\{3,4\}\},$$

$$GR_1^2 = \{\{1,2,3\},\{1,2,4\},\{3,4\}\},$$
$$GR_2^2 = \{\{1,2,3\},\{1,4\},\{2\},\{3,4\}\},$$
$$GR_3^2 = \{\{1,2\},\{3,1\},\{2,3\},\{4\},\{4\}\},$$

$$GR_1^3 = \{\{1,2,3\},\{1,2,4\},\{3,4,1\},\{2,3,4\}\},$$
$$GR_2^3 = \{\{1,2,3\},\{1,4\},\{2,4\},\{3,4\},\{1,2,3\}\}$$

Table 4.2: Different types of 4-MPTRR's.

number of subsets containing 2. Similarly, R contains neither $\{2\}$ nor $\{3\}$. Hence R contains only subsets of size two. Then R must contain all subsets of $\{1,2\},\{1,3\},\{2,3\}$. Therefore R contains $\{\{1,2\},\{1,3\},\{2,3\}\}$, which is a $(4,2)$-MPTRR. A contradiction follows. It follows that $f(3) \leq 2$, so that $f(3) = 2$. The list all 3-MPTRR's of $\mathcal{G}(3)$ is given in Table 4.1.

Next we consider $k = 4$. We say two k-MPTRR's R and R' have the same *type* if their corresponding hypergraphs are isomorphic. We first con-

struct a set of 4-MPTRR's, see Table 4.2, where GR_j^i denotes a jth type of $(4,i)$-MPTRR's. It can easily be checked that each GR_j^i's is a $(4,i)$-MPTRR. It can be seen that no two 4-MPTRR's in Table 4.2 are isomorphic. It can also be verified that Table 4.2 gives all different types of $(4,i)$-MPTRR's with $i \leq 3$. Next we show that $f(4) = 3$.

It is clear that $f(4) \geq 3$ since GR_2^3 is a $(4,3)$-MPTRR. Now we prove that $f(4) \leq 3$. Suppose, on the contrary, that there is a $(4,r)$-MPTRR, say R, with $r \geq 4$. Then R does not contain any 4-MPTRR of types in Table 4.2. We may assume that R is a $(4,r)$-MPTRR with a minimum number of subsets over all $(4,r)$-MPTRR's. Then the intersection of every pair of subsets in R is not empty because if $N_1, N_2 \in R$ such that $N_1 \cap N_2 = \emptyset$, then $(R \setminus \{N_1, N_2\}) \cup \{N_1 \cup N_2\}$ would be a $(4,r)$-MPTRR with a smaller number of subsets. Consequently, R does not contain a subset of size one since otherwise, say $\{1\} \in R$, then all subsets of R must contain 1, which implies that the number of subsets in R containing 1 would be greater than the number of subsets containing 2, which contradicts the fact that R is a tight $(4,r)$-RR. We have that R does not contain $\{1,2,3,4\}$ and R does not contain all of $\{1,2,3\}, \{2,3,4\}, \{3,4,1\}, \{4,1,2\}$. We next consider four cases according to the containment relations of R with subsets $\{1,2,3\}, \{2,3,4\}, \{3,4,1\}, \{4,1,2\}$.

Case 1: R contains three of $\{1,2,3\}, \{2,3,4\}, \{3,4,1\}, \{4,1,2\}$.

Without loss of generality, assume that R contains $\{1,2,3\}, \{2,3,4\}$, and $\{3,4,1\}$. Then R does not contain $\{1,4\}$ because $\{\{1,2,3\}, \{2,3,4\}, \{1,4\}\}$ is a $(4,2)$-MPTRR. Similarly, R does not contain $\{1,2\}$ or $\{2,4\}$. Therefore, R contains only the subsets of $\{1,2,3\}, \{2,3,4\}, \{3,4,1\}, \{3,4\}, \{2,3\}, \{1,3\}$. Let $x_i, i = 1, \ldots, 6$ be the multiplicities of these subsets in R. Then x_i's

are nonnegative integers and satisfy the following equations, in which each expression is the number of nets incident with a side:

$$x_1 + x_3 + x_6 = x_1 + x_2 + x_5 = \sum_{i=1}^{6} x_i = x_2 + x_3 + x_4 = r.$$

The only nonnegative integer solution of the above system of equations is $x_i = 0, i = 1, \ldots, 6$, which implies that $r = 0$, a contradiction. In fact, if all subsets in R contain a common element, that would also lead to a contradiction.

Case 2: R contains two of $\{1,2,3\}, \{2,3,4\}, \{3,4,1\}, \{4,1,2\}$.

Without loss of generality, assume that R contains $\{1,2,3\}$ and $\{2,3,4\}$. Then R does not contain $\{1,4\}$ because $\{\{1,2,3\}, \{2,3,4\}, \{1,4\}\}$ is a $(4,2)$-MPTRR. If R contains both $\{1,2\}$ and $\{1,3\}$, then R contains neither $\{3,4\}$ nor $\{2,4\}$, then R contains only the subsets of $\{1,2,3\}$, $\{2,3,4\}$, $\{1,2\}$, $\{1,3\}$ and $\{2,3\}$. Let $x_i, i = 1, \ldots, 5$ be the multiplicities of these subsets in R. Then

$$x_1 + x_3 + x_4 = x_1 + x_2 + x_3 + x_5 = x_1 + x_2 + x_4 + x_5 = x_2 = r.$$

This implies that $r = 0$, a contradiction. Therefore R does not contain both $\{1,2\}$ and $\{1,3\}$. Similarly, R does not contain both $\{4,2\}$ and $\{4,3\}$.

If R contains $\{1,2\}$, then R contains only the subsets of $\{1,2,3\}$, $\{2,3,4\}$, $\{1,2\}$, $\{2,3\}$ and $\{2,4\}$. All these subsets contain 2, a contradiction. Therefore R does not contain $\{1,2\}$. Similarly, R does not contain any of $\{1,3\}$, $\{4,2\}$ and $\{4,3\}$. Then R contains only the subsets of $\{1,2,3\}, \{2,3,4\}$, and $\{2,3\}$. This is a impossible as all these three subsets contain 2.

Case 3: R contains one of $\{1,2,3\}, \{2,3,4\}, \{3,4,1\}, \{4,1,2\}$.

We may assume that R contains $\{1,2,3\}$. Then R must contain at least one of $\{3,4\}, \{1,4\}$ and $\{2,4\}$ because R must contain nets inci-

dent with 4 and these three nets are the only available subsets containing 4. If R contains all of $\{3,4\},\{1,4\}$ and $\{2,4\}$, then R does not contain any of $\{1,2\},\{2,3\}$ and $\{1,3\}$. Now R contains only the subsets of $\{1,2,3\},\{3,4\},\{1,4\},\ \{2,4\}$. Let $x_i, i = 1,\ldots,4$ denote the multiplicities of $\{1,2,3\},\{3,4\},\ \{1,4\},\ \{2,4\}$ in R, respectively. Then the following equations must hold:

$$x_1 + x_3 = x_1 + x_4 = x_1 + x_2 = x_2 + x_3 + x_4 = r.$$

Therefore, $x_2 = x_3 = x_4 = \frac{r}{3} \geq 1$, and $x_1 = \frac{2r}{3} \geq 2$. Then R contains

$$\{\{1,2,3\},\{1,2,3\},\{3,4\},\{1,4\},\{2,4\}\},$$

which is a $(4,3)$-MPTRR. A contradiction follows.

If R contains exactly two of $\{3,4\},\{1,4\}$ and $\{2,4\}$, we may assume, without loss of generality, that R contains $\{3,4\}$ and $\{1,4\}$. Then R contains only the subsets of $\{1,2,3\},\{1,3\},\ \{3,4\},\{1,4\}$. Let $x_i, i = 1,\ldots,4$ denote the multiplicities of $\{1,2,3\},\{1,3\},\{3,4\},\ \{1,4\}$ in R, respectively. Then following equations must hold:

$$x_1 + x_2 + x_4 = x_1 = x_1 + x_2 + x_3 = x_3 + x_4 = r$$

It follows that $r = 0$, a contradiction.

If R contains exactly one of $\{3,4\},\{1,4\}$ and $\{2,4\}$. We may assume, without loss of generality, that R contains $\{3,4\}$. Then R contains only the subsets of $\{1,2,3\},\{3,4\},\{2,3\},\ \{1,3\}$. All these subsets contain 3, a contradiction.

Case 4: R does not contain any one of $\{1,2,3\},\{2,3,4\},\{3,4,1\},\{4,1,2\}$.

Then R contains at most three different types of subsets of size 2 since no two nets have empty intersections. Without loss of generality,

assume that R contains $\{1,2\},\{1,3\}$. Then R contains neither $\{3,4\}$ nor $\{2,4\}$. Then R must contain $\{1,4\}$. Then R contains only the subsets of $\{1,2\},\{1,3\},\{1,4\}$. But all these subsets contain 1, a contradiction follows.

In summary, every possible case leads to a contradiction. Therefore $f(4) = 3$, and Table 4.2 contains all possible types of 4-MPTRR's.

For $k \geq 5$, the computation of multi-pin net k-MPTRR's and $f(k)$ have not yet been proved formally. However, we have computed all 2-pin net k-MPTRR's for $k = 1, 2, \ldots, 8$. The exact value of $f_2(k)$ has also been derived, see Chapter 7.

4.5 Conclusions and Future Work

In this section, we presented a method of enumerating routing requirements via solving a system of linear Diophantine equations. The good news is that we can use existing Hilbert basis algorithms to enumerate all minimal, tight routing requirements, then use these to generate all tight P-net routing requirements. This shows that the set of all k-MPTRR's for any fixed k is enumerable.

The bad news is that using the known Hilbert basis algorithm to enumerate multi-pin net k-MPTRR's can not be done in polynomial time in k. We implemented the Contejean and Devie's algorithm to enumerate multi-pin net k-MTRR's. The computation terminated for for $k = 1, 2, 3, 4, 5$. But for $k = 6$, it ran for more than 24 hours without termination. One reason is that the number of variables in the corresponding system of linear Diophantine equations is exponential, i.e., 2^k. Another reason is that the algorithm needs a large amount memory to store a candidate set.

We also gave an upper bound for the function $f(k)$, which is the maxi-

mum r such that there is a multi-pin net (k, r)-MPTRR's, or equivalently the maximum degree over all non-decomposable regular hypergraphs on k vertices. As examples, we gave k-MPTRR's for $k \leq 4$, and proved that $f(1) = 1, f(2) = 1, f(3) = 2$ and $f(4) = 3$.

4.5.1 Future Work

The set of routing requirements plays an important role in the design of switch blocks and reconfigurable routing networks because any design is eventually used to implement the routing requirements. So the investigation of the generation of routing requirements and properties of minimal routing requirements, such as the decomposition property, of a set of routing requirements are important topics for switch module design.

With regards to the generation of routing requirements, we are interested in finding an efficient Hilbert basis algorithm suitable for this specific problem. It is possible that some other existing Hilbert basis algorithms might work better for this application. So further experiments are needed.

Generation of routing requirements can make use of the properties of routing requirements. For example, to generate all multi-pin net k-MPTRR's, we can use the value of $f(k)$ to reduce the search space. Determining the value of $f(k)$ or an upper bound and finding new combinatorial properties of a multi-pin net k-MPTRR are important in this respect.

Problem 1 *Determine the value of $f_s(k)$ for $k \geq 5$ and $k \geq s \geq 3$. In particular, determine the value of $f(k)$ for $k \geq 5$.*

Another class of open problems is counting the number of routing requirements determined by a routing capability specification. This number is

useful in designing a switching network. The logarithm of the number gives a lower bound of the number of switches needed in the switch block with the routing capability. Chang et al. [14] computed the number of different 2-pin net $(4, w)$-RR's by first showing that it is a polynomial in w, and then computing the value of the coefficients. It seems that the argument is also true for 2-pin net (k, w)-TRR's, but this has not yet been proven. The following counting problems are still open.

Problem 2 *(i) Determine the number of different 2-pin net (k, w)-TRR's for $k \geq 5$ and $w \geq 1$.*
(ii) Determine the number of different multi-pin net (k, w)-TRR's for $k \geq 3$ and $w \geq 1$.

Historical notes:

I proved the finiteness of the number of k-MPTRR's when we first needed that fact. However the result is covered by Higman's lemma. M. Fellows introduced Higman's lemma to me. The result is also covered by arguments from the Hilbert basis. J. Shallit brought the notion of linear Diophantine equations to my attention. I did the concrete proofs for k-MPTRR's for $k \leq 4$. The functions $f(k)$ and $f_s(k)$ were first proposed in joint work with J. Liu. Liu saw the problem of non-decomposable hypergraphs to be an interesting combinatorial problem. He also defined another function $g(k)$, the maximum degree of non-decomposable *uniform* (i.e., all the edges have the same size) regular hypergraphs on k vertices, and proved several properties about $g(k)$. Noticing that $f(4) = 3$, $f(3) = 2$, $f(2) = 1$, J. Liu conjectured that $f(k) = k - 1$. I talked with P. Hexall about $f(k)$

and the conjecture. She later sent me a proof that shows $f(k)$ has a super polynomial lower bound. We wrote a joint paper [21], which includes the background on $f(k)$, the finiteness of $f(k)$ and the values of $f(k)$ with $k \leq 4$, Hexall's lower bound theorem, and Liu's properties for minimal global routings. Finding on bound for $f(k)$ was proposed as an open problem.

I had been trying to find an upper bound for $f(k)$ for a year, until I met J. Shallit. He mentioned to me the possible usage of Hilbert bases of linear Diophantine equations. I researched this field and found Pottier's theorem in [18], from which I derived an upper bound for $f(k)$ (4.8).

Chapter 5

A Decomposition Design Scheme for Switch Blocks

In this chapter, we present a decomposition design scheme for designing a class of switch blocks. The whole scheme consists of

1. a routing requirement enumeration scheme, (see Chapter 4)
2. tight routing requirement decomposition and composition algorithms,
3. a prime switch block design scheme,
4. a switch block construction scheme, and
5. a feasible routing algorithm.

We will describe the decomposition design scheme for k-HUSB's using multi-pin net routing requirements. For convenience, we omit the "multi-pin net" in the "phrase a multi-pin net routing requirement". All the arguments are valid for other types of switch blocks, i.e. k-USB's, k-P-USB's, $(dw+c)$-HUSB's, $(dw+c)$-USB's, and $(dw+c)$-P-HUSB's.

5.1 Decomposition Theorems

Our switch block design technique depends on the decomposition properties of routing requirements. We have the following decomposition theorem by Theorem 4.3.1.

Theorem 5.1.1 (Decomposition Theorem I) *For any fixed $k \geq 0$, the number of minimal k-MPTRR's is finite, and any (k, w)-PTRR can be decomposed into a disjoint union of k-MPTRR's.*

Lemma 5.1.2 (Fan et al. [22, 23]) *Let $\{r_1, \ldots, r_t\}$ be a multi-set of positive integers, and p be the least common multiple of $r_1, \ldots, r_t$. If $r_1x_1 + \ldots + r_tx_t \geq tp - t + 1$, where $x_1, \ldots, x_t$ are nonnegative integers, then there are integers $y_1, \ldots, y_t$ such that $0 \leq y_i \leq x_i, i = 1, \ldots, t$ and $r_1y_1 + \ldots + r_ty_t = p$.*

Proof. By the pigeonhole principle, there is an integer $1 \leq i \leq t$ such that $r_ix_i \geq p$. Therefore, there is a positive integer y_i such that $y_i \leq x_i$ and $r_iy_i = p$ since p is a multiple of r_i. For any $j \neq i$, let $y_j = 0$. Then $0 \leq y_l \leq x_l$ for $l = 1, \ldots, t$ and $r_1y_1 + \cdots + r_ty_t = p$. □

Lemma 5.1.3 *Let $S = \{r_1, \ldots, r_t\}$ be a finite multi-set of positive integers. Then there exist an integer p_S and a finite nonnegative integer set D_S such that: for any given integer $w \geq 0$, there is an $r_w \in D_S$ such that if the nonnegative vector $(x_1, \ldots, x_t)$ satisfies $r_1x_1 + \ldots + r_tx_t = w$, then there exist $\frac{w-r_w}{p_S} + 1$ nonnegative integer vectors $(y_{i,1}, \ldots, y_{i,t}), i = 1, \ldots, \frac{w-r_w}{p_S} + 1$ such that*

$$\sum_{j=1}^{t} y_{i,j}r_j = \begin{cases} p_S, & i = 1, \ldots, \frac{w-r_w}{p_S}, \\ r_w, & i = \frac{w-r_w}{p_S} + 1 \end{cases} \tag{5.1}$$

and

$$\sum_{i=1}^{\frac{w-r_w}{p_S}+1} (y_{i,1}, \ldots, y_{i,t}) = (x_1, \ldots, x_t) \tag{5.2}$$

Proof. Let p_S be the least common multiple of $r_1, \ldots, r_t$, and let $D_S = \{0, 1, \ldots, tp_S - t\}$. We show that p_S and D_S satisfy the statements by induction on w.

When $w = 0$, then $(x_1, \ldots, x_t) = (0, \ldots, 0)$. Let $r_w = 0$ and $(y_{1,1}, \ldots, y_{1,t}) = (0, \ldots, 0)$. We prove the truth of the lemma for w by assuming the truth for all values less than w.

If $w \leq tp_S - t$, then we choose $r_w = w$. Then $r_w \in D_S$ and $\frac{w-r_w}{p_S} = 0$. For any nonnegative vector $(x_1, \ldots, x_t)$ with $r_1x_1 + \ldots + r_tx_t = w$, we choose $(y_{1,1}, \ldots, y_{1,t}) = (x_1, \ldots, x_t)$. Clearly the statements are satisfied.

Otherwise $w > tp_S - t$. Consider $w' = w - p_S$. Since $w' < w$, by the induction hypothesis, we know there exists an $r_{w'} \in D_S$, satisfying the statements with respect to w'. Let $r_w = r_{w'}$. We show that r_w satisfies the requirements with respect to w.

Let $(x_1, \ldots, x_t)$ be a nonnegative integer vector satisfying $r_1x_1 + \ldots + r_tx_t = w$. Since $w > tp_S - t$, by Lemma 5.1.2 we know there exists $(0, \ldots, 0)^T \preceq (y_{1,1}, \ldots, y_{1,t})^T \preceq (x_1, \ldots, x_t)^T$ such that $\sum_{j=1}^{t} r_j y_{1,j} = p_S$.

Now consider $(x_1 - y_{1,1}, \ldots, x_t - y_{1,t})$. Since $r_1(x_1 - y_{1,1}) + \ldots + r_t(x_t - y_{1,t}) = w - p_S = w' < w$, by the assumption about the value of w', there exist vectors $(y_{i,1}, \ldots, y_{i,t}), i = 2, \ldots, \frac{w'-r_{w'}}{p_S} + 2 = \frac{w-r_w}{p_S} + 1$ satisfying the conditions (5.1) and (5.2) with respect to $(x_1 - y_{1,1}, \ldots, x_t - y_{1,t})$. Now, $(y_{i,1}, \ldots, y_{i,t}), i = 1, \ldots, \frac{w-r_w}{p_S} + 1$, satisfy the conditions (5.1) and (5.2) with respect to $(x_1, \ldots, x_t)$.

□

Example 5.1.1 Let $S = \{1, 2\}$, then $p_S = 2$ and $D_S = \{0, 1\}$ satisfy the statements of Lemma 5.1.3.

Let w be any positive integer and $(0, 0) \preceq (x_1, x_2)$ with $x_1 + 2x_2 = w$. When w is odd, x_1 must be odd. Let $r_w = 1$ and

$$(y_{i,1}, y_{i,2}) = \begin{cases} (2,0), & i = 1, \ldots, \frac{x_1-1}{2}, \\ (0,1) & i = \frac{x_1-1}{2} + 1, \ldots, \frac{x_1-1}{2} + x_2 \\ (1,0) & i = \frac{x_1-1}{2} + x_2 + 1. \end{cases}$$

When w is even, x_1 is even. Let $r_w = 0$ and

$$(y_{i,1}, y_{i,2}) = \begin{cases} (2,0), & i = 1, \ldots, \frac{x_1}{2}, \\ (0,1) & i = \frac{x_1}{2} + 1, \ldots, \frac{x_1}{2} + x_2 \end{cases}$$

Then the above r_w and $(y_{i,1}, y_{i,2})$'s satisfy conditions (5.1) and (5.2).

It can been seen that the minimum p_S must be the least common multiple of $r_1, \ldots, r_t$. But D_S is not necessarily $\{0, 1, \ldots, tp_S - t\}$. The following example shows a smaller D_S.

Example 5.1.2 Let $S = \{1, 2, 3\}$. We show that $p_S = 6$ and $D_S = \{0, 1, 2, 3, 4, 5, 7\}$ satisfy the statements of Lemma 5.1.3. Let $w > 0$ and $x_1 + 2x_2 + 3x_3 = w$.

If $w = 6$ then let $r_w = 0$ and $(y_{1,1}, y_{1,2}, y_{1,3}) = (x_1, x_2, x_3)$;

else if $w \leq 7$, let $r_w = w$ and $(y_{1,1}, y_{1,2}, y_{1,3}) = (x_1, x_2, x_3)$;

else we have $x_1 + 2x_2 + 3x_3 \geq 8$. It suffices to show that there exists $(y_{1,1}, y_{1,2}, y_{1,3})$ satisfying $(y_{1,1}, y_{1,2}, y_{1,3}) \preceq (x_1, x_2, x_3)$ and $y_{1,1} + 2y_{1,2} + 3y_{1,3} = 6$.

If $x_3 \geq 2$, let $(y_{1,1}, y_{1,2}, y_{1,3}) = (0, 0, 2)$;

else if $x_1 \geq 6$, let $(y_{1,1}, y_{1,2}, y_{1,3}) = (6, 0, 0)$;

else if $x_2 \geq 3$, let $(y_{1,1}, y_{1,2}, y_{1,3}) = (0, 3, 0)$;

else if $x_3 \geq 1$ and $x_1 \geq 1$ and $x_2 \geq 1$, let $(y_{1,1}, y_{1,2}, y_{1,3}) = (1, 1, 1)$;

else if $x_1 \geq 2$ and $x_2 \geq 2$ and $x_3 = 0$, let $(y_{1,1}, y_{1,2}, y_{1,3}) = (2, 2, 0)$;

else if $x_1 \geq 4$ and $x_2 = 1$ and $x_3 = 0$, let $(y_{1,1}, y_{1,2}, y_{1,3}) = (4, 2, 0)$;

Else, one of the following must hold: $\{x_1 \leq 6, x_2 = 0, x_3 = 0\}$, $\{x_1 \leq 3, x_2 = 1, x_3 = 0\}$, $\{x_1 \leq 2, x_2 = 0, x_3 = 1\}$, $\{x_1 = 0, x_2 = 2, x_3 = 1\}$. In each of these cases, the value of $x_1 + 2x_2 + 3x_3$ is less than 8.

Theorem 5.1.4 (Decomposition Theorem II) *For any fixed $k > 0$, there exists a $p_k > 0$, and a finite nonnegative integer set D_k such that for any $w > 1$, there exists an $r_w \in D_k$ such that that every (k, w)-PTRR can be decomposed into one (k, r_w)-PTRR and $(w - r_w)/p_k$ (k, p_k)-PTRR's.*

Proof. By Theorem 5.1.1, for a fixed k, there is a finite number of k-MPTRR's, so that the set of capacities of k-MPTRR's, S, is a finite set. Let $S = \{r_1, \ldots, r_t\}$. Then there exist p_S and D_S satisfying the statements of Lemma 5.1.3. Let $p_k = p_S$. Let $D_k = D_S$. For an integer $w \geq 1$, by Lemma 5.1.3, there exists an $r_w \in D_k$ satisfying the statements of Lemma 5.1.3. By Theorem 5.1.1, any (k, w)-PTRR can be decomposed into k-MPTRR's. These k-MPTRR's can be regrouped according to their capacities, i.e., the k-MPTRR's with the same capacity are put in the same group. Let there be x_i (k, r_i)-MPTRR's. Then $r_1x_1 + \ldots + r_tx_t = w$. Therefore there exist $(y_{i,1}, \ldots, y_{i,t}), i = 1, \ldots, \frac{w-r_w}{p_k} + 1$ satisfying the conditions (5.1) and (5.2) with respect to $(x_1, \ldots, x_t)$. Then for each $i = 1, \ldots, t$, we partition the x_i (k, r_i)-MPTRR's into $\frac{w-r_w}{p_k} + 1$ groups $\mathcal{G}_{i,j}$, which contains $y_{i,j}$ (k, r_i)-MPTRR's, $j = 1, \ldots, \frac{w-r_w}{p_k} + 1$. Then $\cup_{i=1}^{t} \mathcal{G}_{i,j}$ is a (k, p_k)-PTRR when $j = 1, \ldots, \frac{w-r_w}{p_k}$, and a (k, r_w)-PTRR when $j = \frac{w-r_w}{p_k} + 1$.

□

Theorem 5.1.5 (Fan et al. [22, 23]) *For any fixed $k > 0$, there exists a $p_k > 0$ and a finite nonnegative integer set D_k such that for any $w > 0$,*

there exists an $r_w \in D_k$ such that the disjoint union of one (k, r_w)-HUSB and $(w - r_w)/p_k$ copies of (k, p_k)-HUSB's is a (k, w)-HUSB.

Proof. Consider a (k, w)-SB G which is a disjoint union of one (k, r_w)-HUSB G' and $(w - r_w)/p_k$ copies of a (k, p_k)-HUSB G''. By Theorem 5.1.4, any (k, w)-PTRR R can always be decomposed into one (k, r_w)-PTRR and $(w - r_w)/p_k$ (k, p)-PTRR's. Then the (k, r_w)-PTRR is routable in G', and each of the $(w - r_w)/p_k$ (k, p)-PTRR's is routable in one copy of G'', so that R is routable in G. Therefore, G is a (k, w)-HUSB. □

5.2 A Switch Block Design Scheme

The decomposition theorems described in the above section establish a foundation for our decomposition design scheme. By Theorem 5.1.5, we first design (k, w)-HUSB's for $w = p_k, d_1, d_2, \ldots, d_s$, called *prime k-HUSB's*, then we use the prime k-HUSB's to construct all other (k, w)-HUSB's. We refer to this design scheme as a *decomposition design scheme*, and to a (k, w)-HUSB obtained by a disjoint union of prime k-HUSB's as a *compound (k, w)-HUSB.*

A decomposition design scheme for k-HUSB's:

I. Enumerate k-MPTRR's $\mathcal{G}(k)$. Determine the set $S = \{r_1, \ldots, r_t\}$ of capacities of these k-MPTRR's. Note that $f(k) = \max\{r \mid r \in S\}$.

II. Determine p_k and $D_k = \{d_0, \ldots, d_s\}$, with $0 = d_0 < d_1 < \ldots < d_s$ satisfying the statements of Theorem 5.1.5. Note that p_k and $0 = d_0, \ldots, d_s$ can be minimized using properties of the k-MPTRR's.

III. Design a prime (k, r)-HUSB, $P(k, r)$, for each $r \in \{p_k, d_1, \ldots, d_s\}$, and set up a feasible routing table recording the feasible routings of all (k, r)-PRTT's in $P(k, r)$.

IV. For any given w, construct a (k, w)-HUSB as follows:

If $w \in \{p_k, d_1, \ldots, d_t\}$, then use the prime (k, w)-HUSB $P(k, w)$. Otherwise, choose the minimum q such that $w - qp_k \in \{d_0, \ldots, d_s\}$. Then the disjoint union of q copies of $P(k, p_k)$ and one $P(k, w - qp_k)$, i.e., $qP(k, p_k) + P(k, w - qp_k)$, is a (k, w)-HUSB by Theorem 5.1.5.

A routing algorithm for a compound (k, w)-HUSB:

Let $G = qP(k, p_k) + P(k, w - qp_k)$ be a compound (k, w)-HUSB, and R be a (k, w)-RR. The following procedure computes a feasible routing for R in G.

Step 1. Transform R into a (k, w)-TRR R' by adding singletons.

Step 2. Transform R' into a (k, w)-PTRR R'' by primitive operations.

Step 3. Decompose R'' into k-MPTRR's by the following procedure:

Decomposition procedure

Input R''

$\bar{R} := R''$; $\mathcal{G} := \mathcal{G}(k)$; $\mathcal{M} := \emptyset$;

while $\bar{R} \neq \emptyset$ **do**

Choose $M \in \mathcal{G}$;

while $M \subseteq \bar{R}$ **do**

$\mathcal{M} := \mathcal{M} \cup \{M\}$;

$\bar{R} := \bar{R} - \{M\}$;

end while

$\mathcal{G} := \mathcal{G} - \{M\}$;

end while

Output $\mathcal{M}$

Step 4. Compose the k-MPTRR's in $\mathcal{M}$ to form one $(k, w - qp_k)$-PTRR and q (k, p_k)-PTRR's as in the proof of Theorem 5.1.4.

Step 5. Use the feasible routing tables of $P(k, p_k)$ and $P(k, w - qp_k)$ to find a feasible routing for the $(k, w - qp_k)$-PTRR in $P(k, w - qp_k)$ and a feasible routing for each of the (k, p_k)-PTRR's in one copy of $P(k, p_k)$. In this way we obtain a feasible routing of R'' in $qP(k, p_k) + P(k, w - qp_k)$.

Step 6. By removing all realizations for nets in $R'' \setminus R$, we obtain a feasible routing of R in G.

We note that the above feasible routing algorithm is an exact algorithm with running time linear in w. It is clear that steps 1 and 2 take linear time. For step 3, since when k is fixed, the number of k-MPTRR's is finite and each is processed in at most time linear in the while loop, therefore step 3 uses linear time. Step 4 also takes linear time because there are a fixed number of possible combinations and each of them takes a linear time to check. Since it takes a constant time to find a feasible routing by a feasible routing table, it takes linear time to complete step 5. It is obvious that step 6 can be done in linear time. Therefore, finding a feasible routing for a given routing requirement in $qP(k, p_k) + P(k, w - qp_k)$ can be done in time linear in w. However, the linear constant depends on k, and could be exponential in k.

The above decomposition design scheme reduces the k-HUSB design problem to the problem of designing prime k-HUSB's. It is desirable to use optimal prime k-HUSB's. However, designing optimal prime k-HUSB's is a challenge. No effective method is known yet. It is not known either that a compound switch block constructed from optimal prime switch blocks is also an optimal switch block, though it is true in some cases. In practice, we

can use approximate prime k-HUSB's as building blocks. In this respect, the decomposition design scheme does not provide a scheme to solve the optimal k-HUSB design problem, but provides a scheme to derive approximate k-HUSB's. We note that $P(k, p_k)$ is the most important prime k-HUSB because it is the repeated part of a compound (k, w)-HUSB. Therefore, we pay much attention to the design of $P(k, p_k)$.

We define a class of (k, w)-SB's, which will be used frequently as candidate switch block designs below.

Definition 5.2.1 *For positive integers k and w, define $G(k, w)$ to be the (k, w)-SB represented by the k-partite graph $((V_1, \ldots, V_k), E)$ with*

$$V_i = \{v_{i,j} | j = 1, \ldots, w\}, i = 1, \ldots, k \tag{5.3}$$

$$E = \cup_{1 \le i < j \le k} \{v_{i,h} v_{j,h+(j-i)-1} | h = 1, \ldots, w\} \tag{5.4}$$

where the second index of the subscript is taken modulo w.

5.2.1 A General Reduction Design Scheme

Let $P = \{S_1, S_2, \ldots, S_n\}$ be a net pattern set for a k-sided switch block, where $S_i \subseteq \{1, 2, \ldots, k\}$ and $\{\{i\} \mid i = 1, \ldots, k\} \subseteq P$. Let d and c be two k-dimensional nonnegative integer vectors. We consider the problem of designing $(dw + c)$-P-USB for every $w \ge 1$.

Theorem 5.2.1 (Switch Block Construction Theorem) *Let d and c be two k-dimensional nonnegative integer vectors and P be a net pattern set for a k-sided switch block. Then there exist an integer $p > 0$ and a finite set of nonnegative integers D such that, for any $w \ge 1$, there is an $r_w \in D$ such that a disjoint union of one $(dr_w + c)$-P-USB and $(w - r_w)/p$ copies of a (dp)-P-USB is a $(dw + c)$-P-USB.*

Proof. It suffices to show that there exist integers $p > 0$ and $b > 0$ such that for any $w \geq 1$, there exists $r_w \leq b$ such that any P-net $(dw + c)$-TRR can be decomposed into one P-net $(dr_w + c)$-TRR and $(w - r_w)/p$ P-net (dp)-TRR's. We prove this by using $(dw + c)$-TRRV's.

Let $Ax - dw = c$ be the system of linear Diophantine equations corresponding to the P-net $(dw+c)$-TRRV's, where A is the incidence matrix of P. The homogeneous system of linear Diophantine equations $Ax - dw = 0$ has a finite number of minimal solutions by Theorem 4.1.2, $(x_1^T, w_1)^T, \ldots,$ $(x_t^T, w_t)^T$, and $Ax - dw = c$ has a finite number of minimal solutions, $(y_1^T, w_1')^T, \ldots, (y_l^T, w_l')^T$, where x_i^T's and y_i^T's are n-dimensional column nonnegative integer vectors. Then a solution to $Ax - dw = c$ can be expressed as a nonnegative linear combination of $(x_1^T, w_1)^T, \ldots, (x_t^T, w_t)^T$ and one $(y_i^T, w_i')^T$. Let S be the multi-set $\{w_1, \ldots, w_t\}$, and p_S and D_S be the integer and nonnegative integer set satisfying the statements of Lemma 5.1.3. Let $m' = \max\{r \mid r \in D_S\}$ and $m'' = \max\{w_1', \ldots, w_l'\}$.

For any integer $w_0 > 0$, if $w_0 > m' + m''$, let $(x_0^T, w_0)^T$ be a solution to $Ax - dw = c$. This vector can be expressed as $(x_0^T, w_0)^T = c_1(x_1^T, w_1) + \ldots + c_t(x_t^T, w_t)^T + (y_i^T, w_i')^T$, where $c_j \geq 0, j = 1, \ldots, t$ and $i \in \{1, \ldots, l\}$. Since $c_1 w_1 + \ldots, c_t w_t = w_0 - w_i' \geq w_0 - m'' > m'$, by Lemma 5.1.3, there exists $c_i' \leq c_i, i = 1, \ldots, t$ such that $c_1' w_1 + \ldots + c_t' w_t = p_S$. Now $(x'^T, w')^T = (x_0^T, w_0)^T - c_1'(x_1^T, w_1)^T - \ldots - c_t'(x_t^T, w_t)^T = (c_1 - c_1')(x_1^T, w_1)^T + \ldots + (c_t - c_t')(x_t^T, w_t)^T + (y_i^T, w_i')^T$ is a solution to $Ax - dw' = c$ where $w' = w_0 - p_S$. Continue this process until $w' \leq m' + m''$. Let $p = p_S, b = m' + m''$, and $r_{w_0} = w_0 - qp_S$, where q is the smallest positive integer such that $w_0 - qp_S \leq m' + m''$. Then a solution of $Ax - dw_0 = c$ can be expressed as a solution to $Ax - dr_{w_0} = c$ plus $(w_0 - r_{w_0})/p_S$ solutions to $Ax - dp_S = 0$.

That implies that every P-net $(dw_0 + c)$-TRR can be decomposed into a disjoint union of one P-net $(dr_{w_0} + c)$-TRR and $(w_0 - r_{w_0})/p$ P-net (dp)-TRR's. Let $p = p_S$ and $D = \{0, 1, ;m' + m''\}$. Then p and D satisfy the statements of the theorem. □

A decomposition design scheme for $(dw + c)$-P-USB's

I. Enumerate the sets of minimal solutions of the corresponding systems of linear Diophantine equations $Ax - dw = c$ and $Ax - dw = 0$, where A is the incidence matrix of P.

II. Compute an integer p and a finite integer set D satisfying the conditions of Theorem 5.2.1.

III. For each $r \in \{p\} \cup D$, design a prime $(dr+c)$-P-USB, $P(r)$, and set up a feasible routing table recording feasible routings for all $(dr+c)$-TRR's in $P(r)$.

IV. For any given $w \geq 1$, construct a compound $(dw+c)$-P-USB as follows: if $w \in \{p\} \cup D$, then use the prime $(dw + c)$-P-USB $P(w)$, otherwise choose the minimum q such that $w - qp \in D$. The disjoint union of q copies of $P(p)$ and one $P(w - qp)$, i.e., $qP(p) + P(w - qp)$, is a compound $(dw + c)$-P-USB.

5.3 Two-Sided and Three-Sided Hyper-Universal Switch Blocks

Example 3.5.1 shows a primary example of the decomposition scheme for 4-USB's. In this section, we use the decomposition design scheme to design optimal (k, w)-HUSB's for $k = 2, 3$ and all $w \geq 1$. We use $min_{HUSB}(k, w)$ $(min_{USB}(k, w))$ to denote the number of switches in an optimal (k, w)-HUSB (USB).

5.3.1 Two-Sided Hyper-Universal Switch Blocks

From the discussion in section 4.4, we have $\mathcal{G}(2) = \{\{\{1,2\}\}\}$, so that the set of capacities of 2-MPTRR's is $S = \{1\}$. Hence $p_2 = 1, D_S = \{0\}$ because every $(2,w)$-PTRR can be decomposed into a disjoint union of w $(2,1)$-PTRR's. Therefore, there is only one prime 2-HUSB, which is a $(2,1)$-HUSB. It is clear that a complete $(2,1)$-SB, isomorphic to $G(2,1)$, is an optimal $(2,1)$-HUSB.

Let $P(2,1) = G(2,1)$. Then $wG(2,1)$ is a $(2,w)$-HUSB. Since $wG(2,1)$ has w switches and a $(2,w)$-HUSB needs at least w switches to route the $(2,w)$-PTRR formed by w copies of $\{1,2\}$, $wG(2,1)$ is an optimal $(2,w)$-HUSB and $min_{HUSB}(2,w) = w$.

Theorem 5.3.1 *A $(2,w)$-SB is an optimal $(2,w)$-HUSB if and only if it is isomorphic to $wG(2,1)$. The number of switches in an optimal $(2,w)$-HUSB is w, i.e., $min_{HUSB}(2,w) = w$.*

Theorem 5.3.2 *For any pair of integers $k \geq 2$ and $w \geq 1$,*

$$min_{HUSB}(k,w) \geq min_{USB}(k,w) \geq \frac{k(k-1)}{2}w.$$

Proof. For each pair i,j with $1 \leq i < j \leq k$, let $R(i,j)$ be a 2-pin net (k,w)-TRR formed by w copies of $\{i,j\}$'s and w copies of $\{t\}$'s for each $t \in \{1,\ldots,k\} \setminus \{i,j\}$. A feasible routing of $R(i,j)$ must use at least w switches between V_i and V_j. Therefore, a (k,w)-USB contains at least $\binom{k}{2}w$ switches. Since a hyper-universal switch block must be a universal switch block, it follows that $min_{HUSB}(k,w) \geq min_{USB}(k,w) \geq \frac{k(k-1)}{2}w$. □

Theorem 5.3.3 *$G(k,1)$ is an optimal $(k,1)$-HUSB (USB) and $min_{HUSB}(k,1) = min_{USB}(k,1) = \frac{k(k-1)}{2}$.*

Proof. $G(k,1)$ is hyper-universal because it is a complete $(k,1)$-SB. $G(k,1)$ has $\frac{k(k-1)}{2}$ switches, so that $min_{USB}(k,1) \leq min_{HUSB}(k,1) \leq \frac{k(k-1)}{2}$. On the other hand, $min_{HUSB}(k,1) \geq min_{USB}(k,1) \geq \frac{k(k-1)}{2}$ by Theorem 5.3.2. Therefore, $min_{HUSB}(k,1) = min_{USB}(k,1) = \frac{k(k-1)}{2}$ and $G(k,1)$ is an optimal $(k,1)$-HUSB and an optimal $(k,1)$-USB as well. □

By the above theorem, we always choose the prime $(k,1)$-HUSB (USB) $P(k,1)$ to be $G(k,1)$.

5.3.2 Three-Sided Hyper-Universal Switch Blocks

The set of capacities of 3-MPTRR's is $S = \{1,2\}$ by Table 4.1. Then, by Example 5.1.1, we have $p_3 = 2$ and $D_3 = \{0,1,2\}$. So there are two prime 3-HUSB's, one is a $(3,1)$-HUSB and another is a $(3,2)$-HUSB. We have defined $P(3,1)$. We will show that $G(3,2)$ is an optimal $(3,2)$-HUSB, so we can choose the prime $P(3,2)$ to be $G(3,2)$.

Theorem 5.3.4 *$G(3,w)$ is an optimal $(3,w)$-HUSB and $min_{HUSB}(3,w) = 3w$ for every $w \geq 1$.*

Proof. It suffices to show that $G(3,w)$ is hyper-universal because $G(3,w)$ has $3w$ switches, which is a lower bound for $min_{HUSB}(3,w)$ by Theorem 5.3.2.

Consider $G(3,w)$ as a simple graph. Then $G(3,w)$ is 2-regular. We see that

$$v_{1,1}v_{2,1}v_{3,1}v_{1,w}v_{2,w}v_{3,w}\cdots v_{1,t}v_{2,t}v_{3,t}v_{1,t-1}v_{2,t-1}v_{3,t-1}\cdots v_{1,2}v_{2,2}v_{3,2}v_{1,1}$$

is a cycle of $G(3,w)$ using all vertices of $G(3,w)$, therefore $G(3,w)$ is a cycle.

Let $R = \{N_1, N_2, \ldots, N_l\}$ be a $(3,w)$-PTRR. Then $\sum_{i=1}^{l} |N_i \cap \{j\}| = w, j = 1,2,3$, by definition. Since $N_i \subseteq \{1,2,3\}$ is a non-empty set, N_i

must be equal to one of subsets

$$\{1,2,3\},\{1,2\},\{2,3\},\{1,3\},\{1\},\{2\},\{3\}.$$

We claim that the subsets in R can be ordered and the elements in each N_i can be ordered so that $1,2,3$ appear successively in cyclic order (we will say that R can be ordered to satisfy the ordering property). For example, if $R=\{\{1,2\},\{2,3\},\{1,3\}\}$, then R can be ordered as $\{2,3\},\{1,2\},\{3,1\}$, (or $\{\{1,2\},\{3,1\},\{2,3\}\}$) to satisfy the required ordering property. We prove this claim by induction on w.

When $w=1$, then a $(3,1)$-PTRR R must be one of

$$\{\{1,2,3\}\},\{\{1,2\},\{3\}\},\{\{3,1\},\{2\}\},\{\{2,3\},\{1\}\},\{\{1\},\{2\},\{3\}\}.$$

Clearly, R already satisfies the required ordering property. We assume that the claim is true for any $(3,n)$-PTRR for $1\leq n\leq w-1$ and show that the claim is true for any $(3,w)$-PTRR R.

Since the capacities of $(3,w)$-MPTRR's are 1 and 2, R contains a $(3,1)$-PTRR or a $(3,2)$-PTRR.

Case 1: R contains a $(3,1)$-PTRR R_1.

Let $R'=R\setminus R_1$. Then R' is a $(3,w-1)$-PTRR. By the induction hypothesis, R' can be reordered as required.

If $R_1=\{1,2,3\}$, without loss of generality, we assume that the element 1 is the first element in the ordering of R', then simply putting $\{1,2,3\}$ in front of the ordered R' will result in the desired ordering for R.

Suppose that R_1 contains a subset of two elements. Without loss of generality, let $R_1=\{\{1,2\},\{3\}\}$. If 1 or 2, say 2, is the first element in the ordered R', then the first net in the ordered R' is either $\{2\}$ or $\{2,3\}$. In

the first case, the last element is 1, so we can move $\{2\}$ to the end of the ordered R' to obtain a new desired ordering with 3 in the first position. In the latter case, we can move $\{2,3\}$ to the end of the ordered R' to obtain a new desired ordering with 1 in the first position. Therefore, we may assume that either 1 or 3 is in the first position in the ordered R'.

If 1 is the first element in the ordered R', we put $\{1,2\},\{3\}$ in front of the ordered R'. If 3 is the first element in the ordering of R', we put $\{3\},\{1,2\}$ in front of the ordered R'. In either case we obtain the desired ordering R.

Case 2: R contains no $(3,1)$-PTRR.

In this case, R must contain $R_2 = \{\{1,2\},\{3,1\},\{2,3\}\}$. If $R = R_2$, the statements are clearly true. Let $R' = R \setminus R_2$. Then R' is a $(3, w-2)$-PTRR with $w-2 \geq 1$, and which can be ordered to satisfy the ordering property by the induction hypothesis. Furthermore, we can assume that 1 appears first in the ordered R'.

Now by simply putting $\{1,2\},\{3,1\},\{2,3\}$ in front of the ordered R', a desired ordering R will be formed.

We have shown that R can be ordered such that 1,2,3 appear successively in cyclic order. Without loss of generality we assume that the ordered sequence of N_i's is $N_1, N_2, \ldots, N_l$ and 1 2 3 is the first segment in the ordering. Starting from $v_{1,1}$ of the Hamilton cycle

$$v_{1,1}v_{2,1}v_{3,1}v_{1,w}v_{2,w}v_{3,w}\ldots v_{1,t}v_{2,t}v_{3,t}v_{1,t-1}v_{2,t-1}v_{3,t-1}\ldots v_{1,2}v_{2,2}v_{3,2}v_{1,1},$$

we successively cut the section with $|N_i|$ vertices and let the path be $T(N_i)$ along the cycle. Since the cycle has $3w$ vertices and $\sum_{i=1}^{l}|N_i| = 3w$, each $T(N_i)$ is well-defined for $i = 1,\ldots,l$. Since the sequence $N_1, N_2, \ldots, N_l$

generates a sequence 1,2,3,1,2,3,...,1,2,3, $T(N_i)$ is a path of $|N_i|$ vertices and $|V(T(N_i)) \cup V_j| = 1$ if and only if $j \in V_i$.

This implies that $\{T(N_1), T(N_2), \ldots, T(N_l)\}$ is a feasible routing of $\{N_1, N_2, \ldots, N_l\}$ in $G(3, w)$. Therefore $G(3, w)$ is hyper-universal. □

The proof of the above theorem also gives a routing algorithm for $G(3, w)$. It is obvious that we can rearrange the subsets of a $(3, w)$-PTRR in the required ordering in time linear in w. Hence we can find a feasible routing in $G(3, w)$ in polynomial time.

We define the prime $(3, 2)$-HUSB $P(3, 2)$ to be $G(3, 2)$. Then a compound optimal $(3, w)$-HUSB can be obtained. That is, when w is even, $\frac{w}{2}G(3, 2)$ is a $(3, w)$-HUSB with $3w$ switches, so it is an optimal $(3, w)$-HUSB. When w is odd, $\frac{w-1}{2}G(3, 2) + G(3, 1)$ is a $(3, w)$-HUSB with $3w$ switches, so it is an optimal $(3, w)$-HUSB.

We note that $G(3, w)$ is an optimal $(3, w)$-HUSB which corresponds to a connected graph. Therefore, an optimal $(3, w)$-HUSB is not unique, up to isomorphism. In fact, when w is even the disjoint union $G(3, w_1)+G(3, w_2)+\ldots+G(3, w_t)$ is an optimal $(3, w)$-HUSB provided that $\sum_{i=1}^{t} w_i = w$ and all w_i's are even. When w is odd, $G(3, w_1) + G(3, w_2) + \ldots + G(3, w_t)$, it is an optimal $(3, w)$-HUSB provided that $\sum_{i=1}^{t} w_i = w$ and one of the w_i's is odd and all others are even. Conversely, since an optimal $(3, w)$-HUSB graph must induce a matching of w edges between any two sides, it is a graph of degree two, and so that it is a disjoint union of cycles. If it has more than one odd cycles, then it is not routable for the routing requirement consisting of $\lfloor \frac{w}{2} \rfloor$ copies of $\{\{1, 2\}, \{2, 3\}, \{1, 3\}\}$. Therefore we have the following theorem.

Theorem 5.3.5 *When w is even, a $(3, w)$-HUSB (USB) is optimal if and*

only if it is isomorphic to $G(3,w_1)+G(3,w_2)+\ldots+G(3,w_t)$ *where* $\sum_{i=1}^{t} w_i = w$ *and all* w_i*'s are even. When* w *is odd, a* $(3,w)$*-HUSB (USB) is optimal if and only if it is isomorphic to* $G(3,w_1)+G(3,w_2)+\ldots+G(3,w_t)$ *where* $\sum_{i=1}^{t} w_i = w$ *and one of the* w_i*'s is odd and all others are even.*

5.4 Lower and Upper Bounds

Theorem 5.3.2 gives a lower bound on the number of switches in a (k,w)-HUSB. That is, $min_{HUSB}(k,w) \geq \frac{k(k-1)}{2}w$. Next we give an asymptotic upper bound for $min_{HUSB}(k,w)$. The following theorem and its proof give a design for approximate (k,w)-HUSB's.

Theorem 5.4.1 *For any fixed positive integer* k*,*

$$min_{HUSB}(k,w) = O(w).$$

Proof. Use a complete (k,r)-SB $K_{(k,r)}$ as a prime (k,r)-HUSB. Then, by Theorem 5.1.5, we know that there exists p_k and $0 = d_0 < d_1 < d_2 < \ldots < d_s$ such that for any $w > 0$, there exists an integer q and a $r_w \in \{d_0,\ldots,d_s\}$ such that $qK_{(k,p_k)} + K_{(k,r_w)}$ is hyper-universal. Since the maximum value of capacities of k-MPTRR's is $f(k)$, the number of of different capacities of k-MPTRR's is at most $f(k)$, therefore by the proof of Lemma 5.1.2, we know $p_k \leq f(k)!$ and $d_s \leq f(k)f(k)!$. Then

$$\begin{aligned}
|E(qK_{(k,p_k)} + K_{(k,r_w)})| &= qk(k-1)p_k^2/2 + k(k-1)r_w^2/2 \\
&= (qp_k^2 + r_w^2)k(k-1)/2 \\
&= ((w-r_w)p_k + r_w^2)k(k-1)/2 \\
&\leq (wp_k + d_s^2 - d_s p_k)k(k-1)/2 \\
&\leq (wf(k)! + f^2(k)(f(k)!)^2 - f(k)(f(k)!)^2)k(k-1)/2 \\
&= O(w).
\end{aligned}$$

It follows that, when k is fixed, $min_{HUSB}(k, w) = O(w)$. □

This theorem claims that, when k is fixed, the number of switches of an optimal (k, w)-HUSB is linear in w, as compared to $O(w^2)$ for a complete (k, w)-HUSB. Similarly we can show that the compound $(dw + c)$-P-USBs have linear number of switches in w.

5.5 Conclusions and Future Work

In this chapter, we proved the main decomposition theorems for routing requirements. The decomposition theorems guarantee that a (k, w)-HUSB can be constructed by a disjoint union of some copies of a particular prime (k, p_k)-HUSB and a reminder (k, r)-HUSB. The decomposition design scheme says we can design prime (k, r)-HUSB's for a finite number of r's, and then use them to build all other (k, w)-HUSB's. This decomposition design scheme also works for k-USB's and $(dw + c)$-P-USB's. The advantages of the design method are: (1) a (k, w)-HUSB can be derived automatically once all prime designs are known, and (2) the number of switches in a compound (k, w)-HUSB is linear in w, and a feasible routing for a routing requirement in a compound (k, w)-HUSB can be found efficiently. However, designing optimal prime (k, w)-HUSB's is a challenging problem. The decomposition design does not guarantee optimal designs. Approximate prime k-HUSB's can be used.

5.5.1 Future Work

There are two problems in the decomposition design scheme, which are not yet solved completely. The first problem is, given a set $r_1, \ldots, r_t$, how to compute the minimum values of p_k and $d_1 < d_2 < \ldots < d_s$ satisfying the

conditions of Theorem 5.1.5. These values determine the channel capacities of prime k-HUSB's.

The second problem is how to design optimal prime k-HUSB's. In fact, the decomposition design method reduces the k-HUSB design problem to the prime k-HUSB design problem. We do not know how to design optimal prime switch blocks effectively. New methods for designing approximate prime switch blocks are needed. We are particularly interested in optimal $(k, 2)$-HUSBs for $k \geq 5$. We will prove that $G(k, 2)$ is universal in Chapter 7, but it is not yet known if $G(k, 2)$ is hyper-universal.

Regarding $(dw+c)$-HUSB's, we have proved the decomposition theorem for $(dw + c)$-HUSB's and gave a decomposition design scheme for $(dw + c)$-HUSB's. More concrete designs and examples can be done using this design scheme, for example, the design of rectangular switch blocks. A $(dw + c)$-HUSB design scheme can be used to design a $(d_1, \ldots, d_k)^T$-HUSB by choosing appropriate d, c and w. However, a method of choosing d, c and w needs to be developed.

Using a decomposition design scheme to design other kinds of switch blocks can also be considered. For example, prime switch matrices [54] can possibly be introduced and used to build large switch matrices. Directed switches and special routing for certain nets could also be considered.

Moreover, no work has been done on the design of switch blocks routable for a specific set of routing requirements, nor for designing reconfigurable interconnection network routable for a specific set of connection requirements. These two problems have a potential application in the design of on-chip networks for reconfigurable Systems-on-a-Chip.

Historical notes:

The decomposition design scheme was first proposed for $(4, w)$-HUSB's in [22] (joint with J. Liu and W. L. Wu) and further improved in [24, 23]. My main contributions were the $(4, w)$-TRR decomposition theorems, (k, w)-HUSB composition theorems, the determining that the number of switches in an optimal (k, w)-HUSB is linear (when k is fixed), the decomposition design scheme and the feasible routing scheme. These results were verified and improved and tested by J. Liu and W. L. Wu and their students. Using the sets of minimal solutions of systems of linear Diophantine equations to prove the decomposition theorem for $(dw + c)$-HUSB's and the generalized decomposition design scheme for $(dw+c)$-HUSB's were first proposed in this dissertation. I implemented Contejean and Devie's algorithm and designed a few classes of 3-sided and 4-sided irregular switch blocks, but this is not included in this dissertation.

Chapter 6

Four-Sided Hyper-Universal Switch Blocks

In this chapter, we apply the decomposition design scheme to the design of $(4, w)$-HUSB's. We focus on designing prime 4-HUSB's and present two groups of prime 4-HUSB's. We will give a detailed proof of the hyper-universality of the second group, which is an improvement on the first group. Unless stated otherwise, all routing requirements considered in this chapter are multi-pin net routing requirements.

6.1 The Decomposition Design Scheme

Using the decomposition design scheme described in Chapter 5, we can design $(4, w)$-HUSB's as follows.

Reduction design for 4-HUSB's

Step I. Enumerate all 4-MPTRR's. All the 4-MPTRR's are listed in Table 6.1. They are generated from Table 4.2 by applying the permutations of $\{1, 2, 3, 4\}$.

Step II. The set of capacities of 4-MPTRR's is $S = \{1, 2, 3\}$. From Example 5.1.2, we have $p_4 = 6$ and $D_4 = \{0, 1, 2, 3, 4, 5, 7\}$. Therefore, prime 4-HUSB's are $(4, r)$-HUSB's with $r = 1, 2, \ldots, 7$.

Step III. Design prime 4-HUSB's, $P(4, i) = G_i, i = 1, \ldots, 7$, where $G_1 = G(4, 1), G_2 = G(4, 2)$, and G_3, G_4, G_5 G_6 and G_7 are the switch blocks shown in Figure 6.1.

We draw the switch blocks in Figure 6.1 for a clear view for the structure

(4, 1)-MPTRR's:

$GR^1_{1,1} = \{\{1,2,3,4\}\}$	
$GR^1_{2,1} = \{\{1,2\},\{3,4\}\}$	$GR^1_{2,2} = \{\{1,3\},\{2,4\}\}$
$GR^1_{2,3} = \{\{1,4\},\{2,3\}\}$	
$GR^1_{3,1} = \{\{1\},\{2,3,4\}\}$	$GR^1_{3,2} = \{\{2\},\{1,3,4\}\},$
$GR^1_{3,3} = \{\{3\},\{1,2,4\}\}$	$GR^1_{3,4} = \{\{4\},\{1,2,3\}\},$

(4, 2)-MPTRR's:

$GR^2_{1,1} = \{\{1,2,3\},\{1,2,4\},\{3,4\}\}$	$GR^2_{1,2} = \{\{1,2,3\},\{2,3,4\},\{1,4\}\},$
$GR^2_{1,3} = \{\{1,2,4\},\{2,3,4\},\{1,3\}\}$	$GR^2_{1,4} = \{\{1,3,4\},\{2,3,4\},\{1,2\}\}$
$GR^2_{1,5} = \{\{1,2,3\},\{1,3,4\},\{2,4\}\}$	$GR^2_{1,6} = \{\{1,2,4\},\{1,3,4\},\{2,3\}\}$
$GR^2_{2,1} = \{\{1,2,3\},\{1,4\},\{2\},\{3,4\}\}$	$GR^2_{2,2} = \{\{1,2,4\},\{1,3\},\{2\},\{3,4\}\}$
$GR^2_{2,3} = \{\{2,3,4\},\{1,4\},\{2\},\{1,3\}\}$	$GR^2_{2,4} = \{\{1,2,3\},\{1,4\},\{3\},\{2,4\}\}$
$GR^2_{2,5} = \{\{1,4,3\},\{1,2\},\{3\},\{2,4\}\}$	$GR^2_{2,6} = \{\{2,4,3\},\{1,2\},\{3\},\{1,4\}\}$
$GR^2_{2,7} = \{\{2,4,3\},\{1,2\},\{4\},\{1,3\}\}$	$GR^2_{2,8} = \{\{1,4,3\},\{1,2\},\{4\},\{2,3\}\}$
$GR^2_{2,9} = \{\{1,2,4\},\{1,3\},\{4\},\{2,3\}\}$	$GR^2_{2,10} = \{\{1,2,4\},\{4,3\},\{1\},\{2,3\}\}$
$GR^2_{2,11} = \{\{1,4,3\},\{4,2\},\{1\},\{2,3\}\}$	$GR^2_{2,12} = \{\{2,1,3\},\{4,2\},\{1\},\{4,3\}\}$
$GR^2_{3,1} = \{\{1,2\},\{3,1\},\{2,3\},\{4\},\{4\}\}$	$GR^2_{3,2} = \{\{1,2\},\{4,1\},\{2,4\},\{3\},\{3\}\}$
$GR^2_{3,3} = \{\{1,3\},\{4,1\},\{3,4\},\{2\},\{2\}\}$	$GR^2_{3,4} = \{\{3,2\},\{4,3\},\{2,4\},\{1\},\{1\}\}$

(4, 3)-MPTRR's:

$GR^3_{1,1} = \{\{1,2,3\},\{1,2,4\},\{3,4,1\},\{2,3,4\}\}$
$GR^3_{2,1} = \{\{1,2,3\},\{1,4\},\{2,4\},\{3,4\},\{1,2,3\}\}$
$GR^3_{2,2} = \{\{2,3,4\},\{1,2\},\{1,3\},\{1,4\},\{2,3,4\}\}$
$GR^3_{2,3} = \{\{3,4,1\},\{2,1\},\{2,3\},\{2,4\},\{3,4,1\}\}$
$GR^3_{2,4} = \{\{4,1,2\},\{3,1\},\{3,2\},\{3,4\},\{4,1,2\}\}$

Table 6.1: All 4-MPTRR's.

and for the convenience of verification. It can be seen that G_3 contains vertex disjoint subgraphs G_1 and G_2; G_4 contains two vertex disjoint G_2's; G_5 contains vertex disjoint subgraphs G_2 and G_3; G_6 contains three vertex disjoint G_2's; and G_7 contains vertex disjoint subgraphs G_3 and G_4.

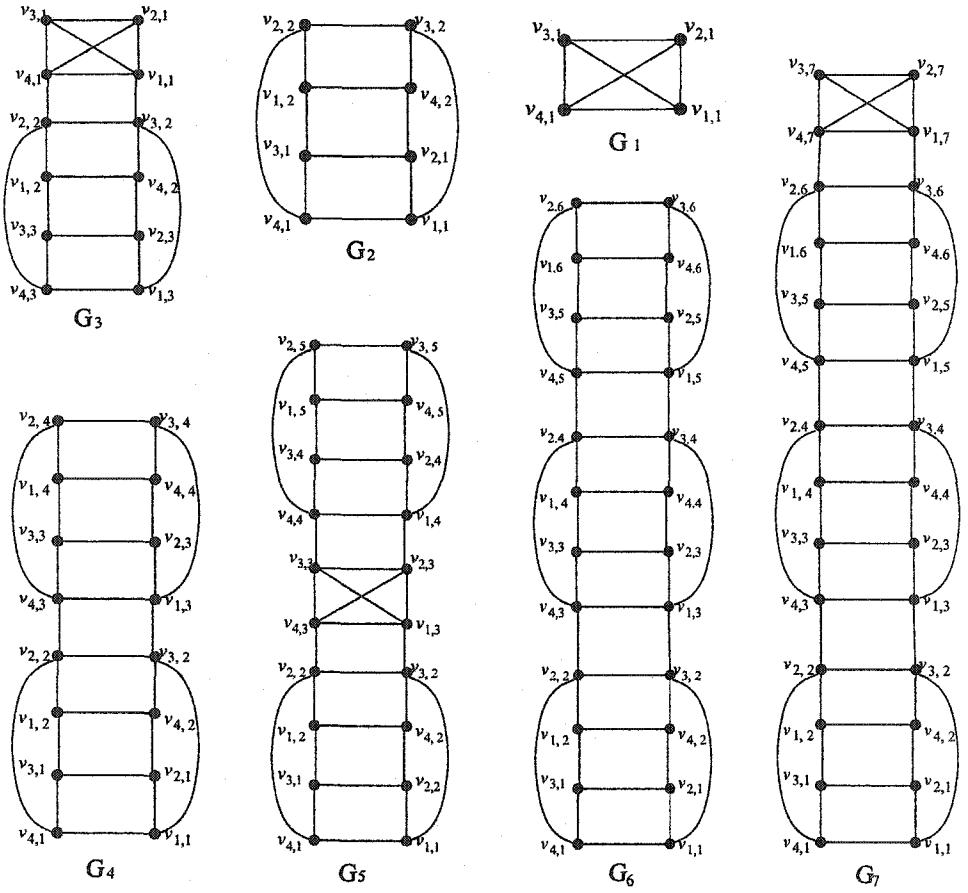


Figure 6.1: The prime 4-HUSB's.

Theorem 6.1.1 *G_i is hyper-universal for $1 \leq i \leq 7$.*

Proof. Let R be a $(4, i)$-PTRR. Then R is a disjoint union of $GR^j_{h,p}$'s as defined by Table 6.1. We show that R is routable in G_i.

Case 1: G_1.

$G_1 = G(4,1)$ is an optimal $(4,1)$-HUSB by Theorem 5.3.3.

Case 2: G_2.

We now show that $G_2 = G(4,2)$ is an optimal $(4,2)$-HUSB. Since G_2 has twelve switches equal to the lower bound on the number of switches in a $(4,2)$-HUSB by Theorem 5.3.2, it suffices to show that $G(4,2)$ is routable for every $(4,2)$-PTRR's. Note that there are eight $(4,1)$-MPTRR's and twenty-two $(4,2)$-MPTRR's, they generate $28+8+22 = 58$ $(4,2)$-PTRR's. The feasible routing table of all $(4,2)$-PTRR's is shown in Figure 6.2, Figure 6.3 and Figure 6.4.

Case 3: G_3.

If $R = GR_1^3$ or $GR_{h,p}^3$ for some h,p, it can be checked directly that G_3 contains a feasible routing for R (details are omitted). If R is a union of a $(4,1)$-MPTRR and a $(4,2)$-MPTRR, then G_3 has a feasible routing for R since G_3 contains disjoint subgraphs isomorphic to G_1 and G_2 respectively.

Case 4: G_4.

If R is a union of a $(4,1)$-MPTRR and a $(4,3)$-MPTRR, then it can be checked directly that G_4 contains a feasible routing for R (details are omitted). If R is a union of two $(4,2)$-PTRR's, G_4 contains a feasible routing for R since G_4 contains two disjoint G_2's.

Case 5: G_5.

G_5 contains disjoint G_2 and G_3. A $(4,5)$-PTRR can always be decomposed into a union of 4-MPTRR's with capacities $1,2$ and 3, these 4-MPTRR's can be regrouped into one $(4,2)$-PTRR and one $(4,3)$-PTRR, so that R is routable in G_5.

Case 6: G_6.

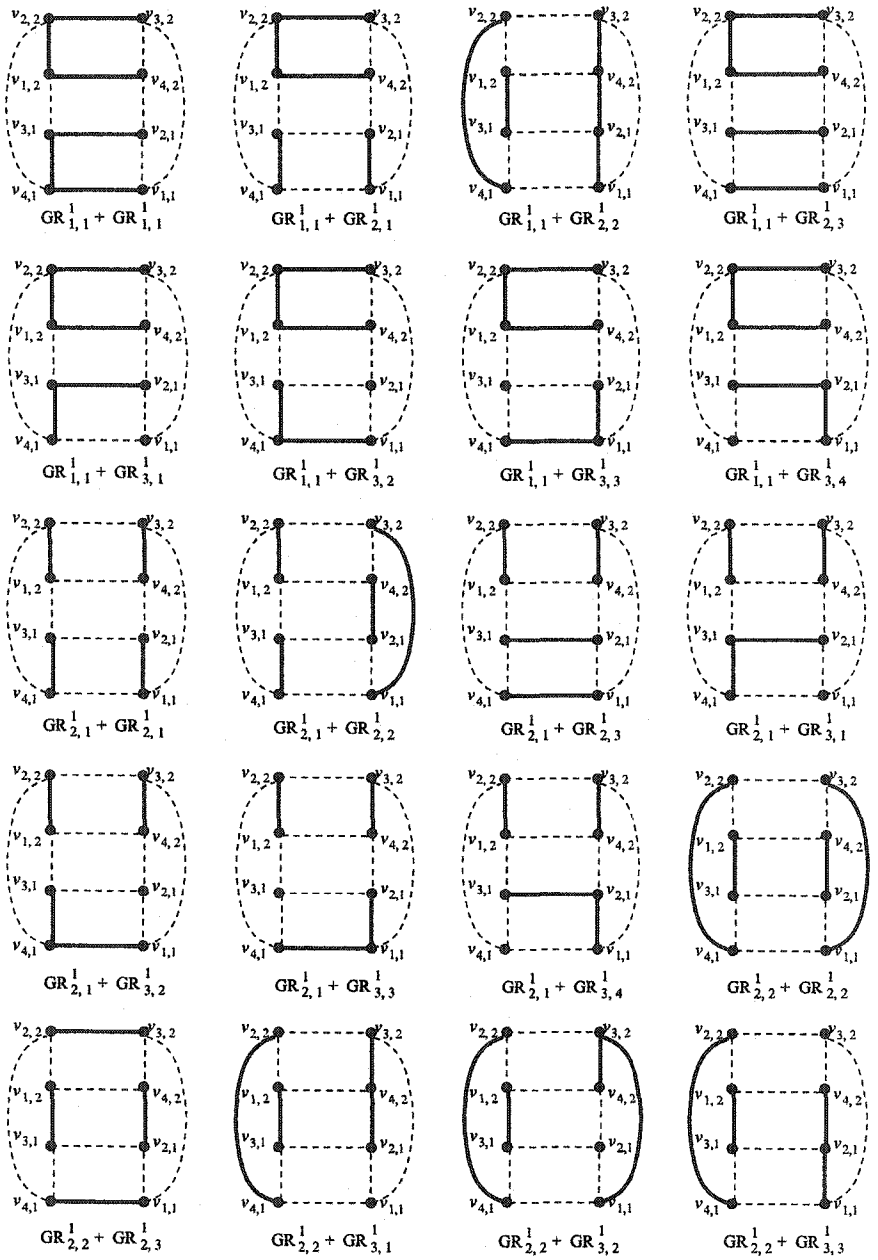


Figure 6.2: Feasible routings of $(4,2)$-PRTT's formed by two $(4,1)$-MPRTT's.

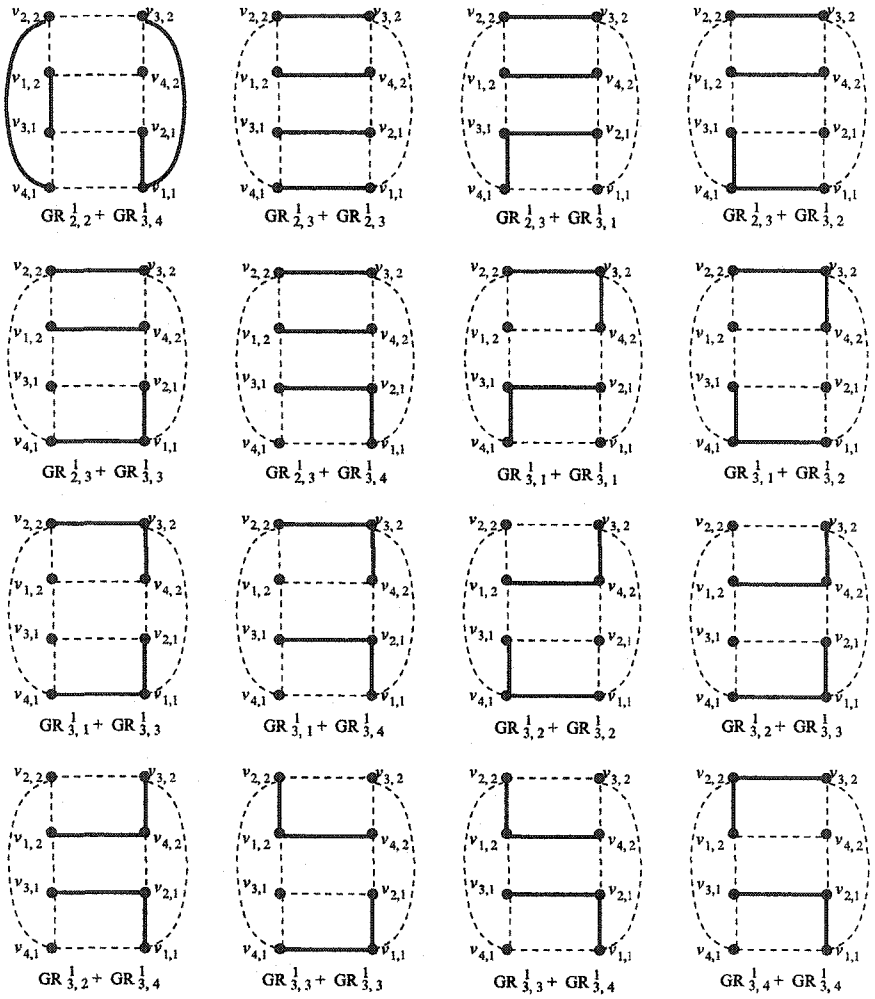


Figure 6.3: Feasible routings of $(4,2)$-PRTT's formed by two $(4,1)$-MPTRR's.

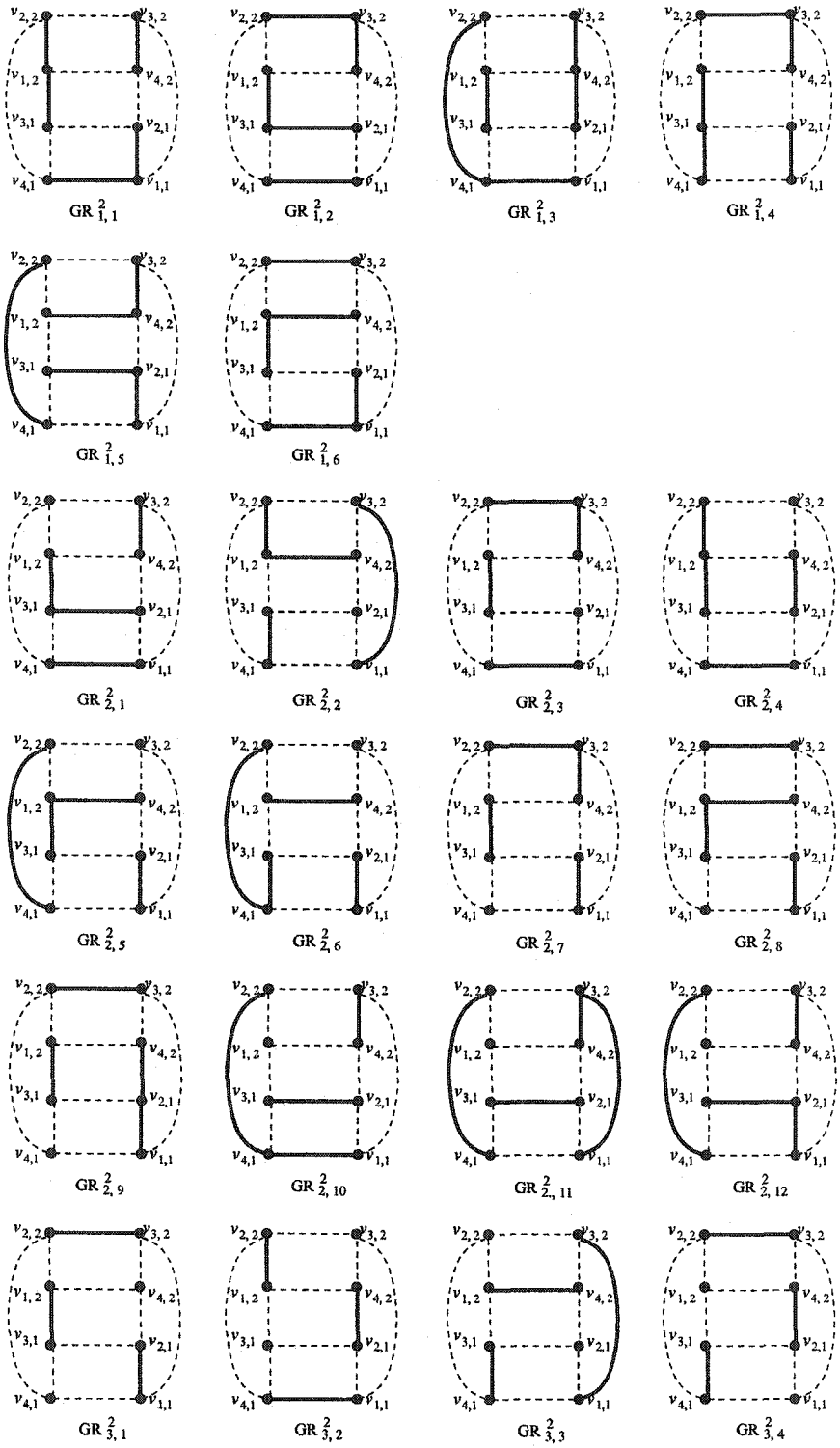


Figure 6.4: Feasible routings of $(4, 2)$-PRTT's formed by one $(4, 2)$-MPTRR.

If R is a union of three $(4,2)$-PTRR's, then G_6 contains a feasible routing of R because G_6 contains three vertex disjoint G_2's. If R is a union of two $(4,3)$-PTRR's, then it can be checked directly that G_6 contains a feasible routing for R.

Case 7: G_7.

Now R can always be decomposed into a $(4,3)$-PTRR and a $(4,4)$-PTRR, so that G_7 contains a feasible routing for R since G_7 contains vertex disjoint subgraphs G_3 and G_4. □

Step IV. Construct $(4,w)$-HUSB's by combining the prime 4-HUSB's G_i $(i = 1,2,\ldots,7)$. Define

$$F(w) = \begin{cases} hG_6 & \text{if } w = 6h, \\ (h-1)G_6 + G_7 & \text{if } w = 6h+1, \\ hG_6 + G_2 & \text{if } w = 6h+2, \\ hG_6 + G_3 & \text{if } w = 6h+3, \\ hG_6 + G_4 & \text{if } w = 6h+4, \\ hG_6 + G_5 & \text{if } w = 6h+5. \end{cases}$$

Since the G_i's $(i = 1,\ldots,7)$ are HUSB's, $F(w)$ is a $(4,w)$-HUSB. The number of switches in $F(w)$ is

$$|E(F(w))| = \begin{cases} \frac{20}{3}w & \text{if } w = 0(\text{mod } 6), \\ \frac{20}{3}w - \frac{2}{3} & \text{if } w = 1(\text{mod } 6), \\ \frac{20}{3}w - \frac{4}{3} & \text{if } w = 2(\text{mod } 6), \\ \frac{20}{3}w & \text{if } w = 3(\text{mod } 6), \\ \frac{20}{3}w - \frac{2}{3} & \text{if } w = 4(\text{mod } 6), \\ \frac{20}{3}w - \frac{4}{3} & \text{if } w = 5(\text{mod } 6). \end{cases}$$

Therefore, $F(w)$ is hyper-universal with $\lceil \frac{20}{3}w \rceil$ switches. Furthermore, we have $min_{HUSB}(4,1) = 6, min_{HUSB}(4,2) = 12$, $18 \leq min_{HUSB}(4,3) \leq$

$|E(G_3)| = 20$, $24 \leq min_{HUSB}(4,4) \leq |E(G_4)| = 26$, $30 \leq min_{HUSB}(4,5) \leq |E(G_5)| = 32$, $36 \leq min_{HUSB}(4,6) \leq |E(G_6)| = 40$ and $42 \leq min_{HUSB}(4,7) \leq |E(G_7)| = 46$. In the next section, we will give improved prime $(4,w)$-HUSB's which attain optimality for $w = 3,4,5$.

Since G_2 is an optimal $(4,2)$-HUSB with 12 switches, G_2 is an optimal $(4,2)$-USB. Therefore, $\frac{w}{2}G_2$ is an optimal $(4,w)$-USB when w is even, and otherwise $\frac{w-1}{2}G_2 + G_1$ is an optimal $(4,w)$-USB because it is universal and has $6w$ switches by Theorem 5.3.2. This completes the arguments of Example 3.5.1. We note that the symmetric $(4,w)$-USB $M_{4,w}$ [14] is isomorphic to $\frac{w}{2}G(4,2)$ when w is even, and $\frac{w-1}{2}G(4,2) + G(4,1)$ when w is odd.

6.2 Better Prime Switch Block Designs

Figure 6.5 depicts a new group of prime 4-HUSB's, $H_i, i = 1,\ldots,7$.

Theorem 6.2.1 *The following statements hold:*
(1) H_i is an optimum $(4,i)$-HUSB for $i = 1,2,3,4,5$.
(2) H_6 is a $(4,6)$-HUSB with 38 switches, within at most two of optimum.
(3) H_7 is a $(4,6)$-HUSB with 43 switches, within at most one of optimum.

In order to make the proof of the hyper-universality of the H_i's easy to verify, we draw H_i so that feasible routings can be checked easily. See Figure 6.6 for the new drawing of the H_i's.

$H_1 = G_1$ and $H_2 = G_2$ are known to be optimal HUSB's. Since every $(4,5)$-PTRR can be decomposed into a union of 4-PTRR's of capacities 1, 2, or 3 and they can be regrouped into one $(4,2)$-PTRR and one $(4,3)$-PTRR, $H_5 = H_2 + H_3$ is hyper-universal provided that H_3 is hyper-universal. Simi-

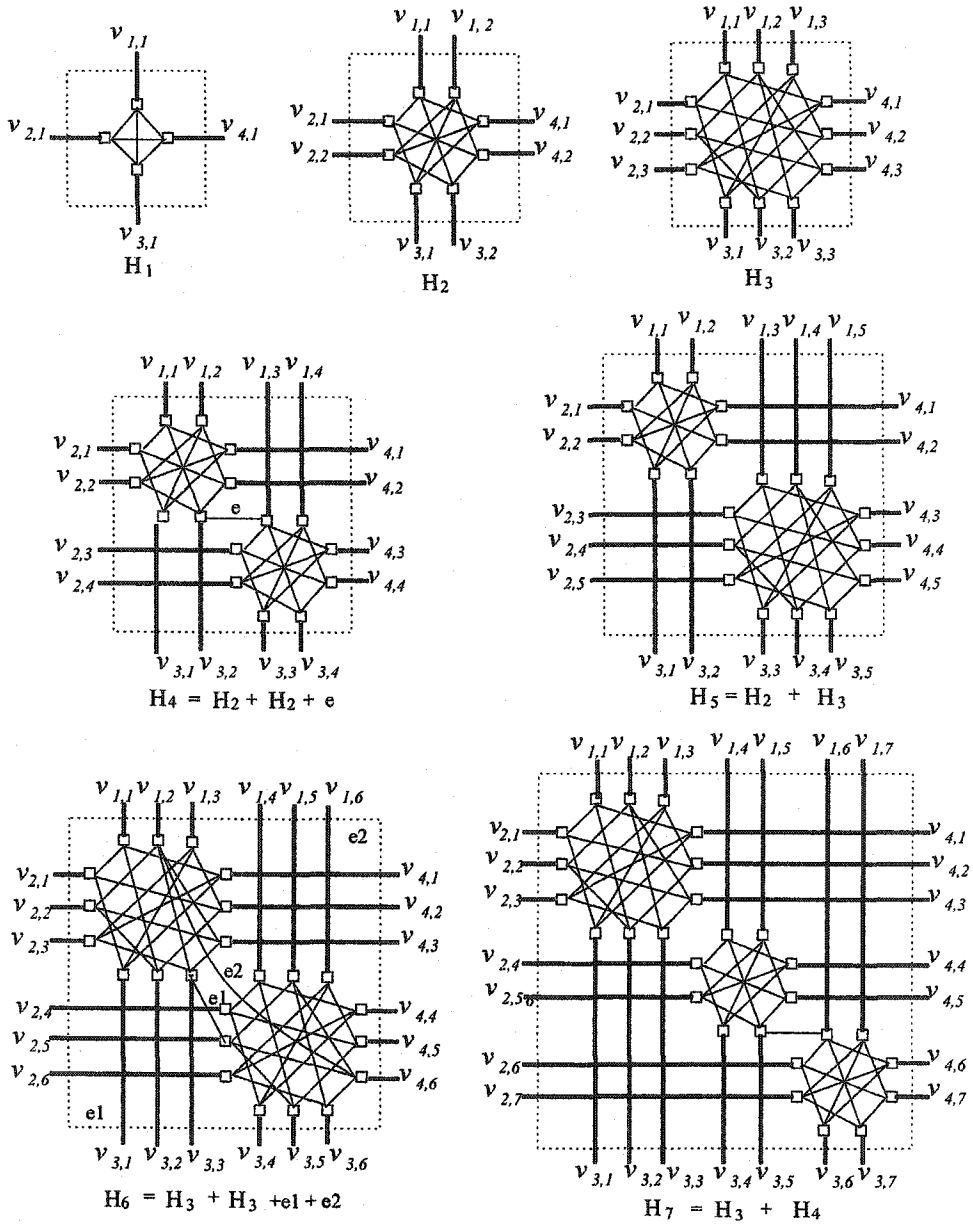


Figure 6.5: New prime 4-HUSB's.

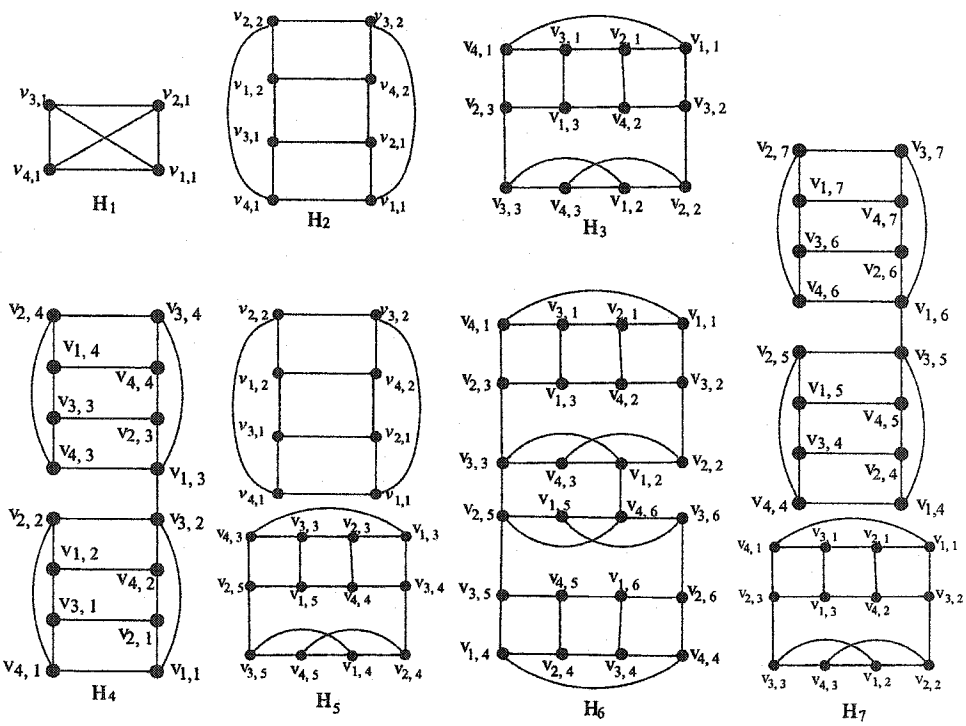


Figure 6.6: An alternative drawing of the prime 4-HUSB's.

larly, since a $(4,7)$-PTRR can always be decomposed into one $(4,3)$-PTRR and one $(4,4)$-PTRR, $H_7 = H_3 + H_4$ is hyper-universal provided that H_3 and H_4 are hyper-universal. Therefore, we need only show that H_3, H_4 and H_6 are hyper-universal.

H_3 is an optimal $(4,3)$-HUSB :

We first explain how H_3 was found. The hunt for an optimal $(4,3)$-HUSB started by examining all $(4,3)$-SB's with eighteen switches. Since any two sides must induce a matching of three edges, in the first round, there are $(3!)^6$ possible candidates. Since any three sides must induce a $(3,3)$-HUSB of nine switches which is known to be optimal, there are two choices, up to isomorphism, i.e., $G(3,3)$ and $G(3,2) + G(3,1)$. With this restriction on sides 1, 2 and 3, and a fixed matching between sides 3 and 4, there were $2 \times 6 \times 6 = 72$ candidates left in the second round. Then we used brute force to test these candidates one by one see if they are hyper-universal. H_3 passed the test.

Next we show that H_3 is hyper-universal. It suffices to show that H_3 has a feasible routing for any $(4,3)$-PTRR's obtained by combining the 4-MPTRR's in Table 6.1. Since there are eight $(4,1)$-MPTRR's, twenty-two $(4,2)$-MPTRR's and five $(4,3)$-MPTRR's, there are $8 \times 8 \times 8 + 8 + 5$ or 693 $(4,3)$-PTRR's to be considered. We find a feasible routing for each of these 693 $(4,3)$-PTRR's.

It can easily be seen that the side permutation $\sigma = (1,4)(2,3)$ induces an automorphism of H_3 (symmetric about the central vertical line). This implies that if R is routable in H_3 then all the $(4,3)$-PTRR's obtained by the permutation σ are also routable in H_3. Therefore, we only need to

consider those $(4,3)$-PTRR's which are not equivalent under σ. Let R be a $(4,3)$-PTRR.

Case 1: R is a $GR^3_{i,j}$.

Since $GR^3_{2,1}$ and $GR^3_{2,2}$ (and $GR^3_{2,3}$ and $GR^3_{2,4}$) are equivalent under σ, it suffices to consider $GR^3_{1,1}, GR^3_{2,1}$ and $GR^3_{2,3}$. The feasible routings of these $(4,3)$-PTRR's are given in Figure 6.7, in which the number at a vertex represents the label of the side containing the terminal.

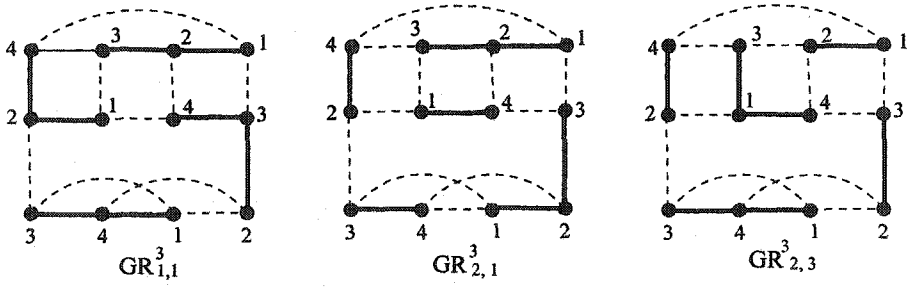


Figure 6.7: Feasible routings of $(4,3)$-MPTRR's in H_3

Case 2: R consists of the $GR^1_{i,j}$'s.

Subcase 2.1: R does not contain a $GR^1_{2,3}$.

In this case, we consider the three 4-cycles B_1, B_2 and B_3 of H_3, each of which has the vertex ordering $1,2,4,3$, see Figure 6.8(i). Such a 4-cycle is routable for a $GR^1_{i,j}$ except $GR^1_{2,3}$. Therefore, R has a feasible routing in H_3.

Subcase 2.2: $R = 3GR^1_{2,3}$.

A feasible routing is given in Figure 6.8(a).

Subcase 2.3: R contains exactly one $GR^1_{2,3}$.

If R contains $GR^1_{2,3} + GR^1_{2,2}$, $GR^1_{2,3} + GR^1_{3,2}$, $GR^1_{2,3} + GR^1_{2,1}$, or $GR^1_{2,3} + GR^1_{1,1}$, then we route R as shown in Figure 6.8(b),(c),(d), and (e), respectively, because the unused 4-cycle with vertex label order $1,2,4,3$ which is

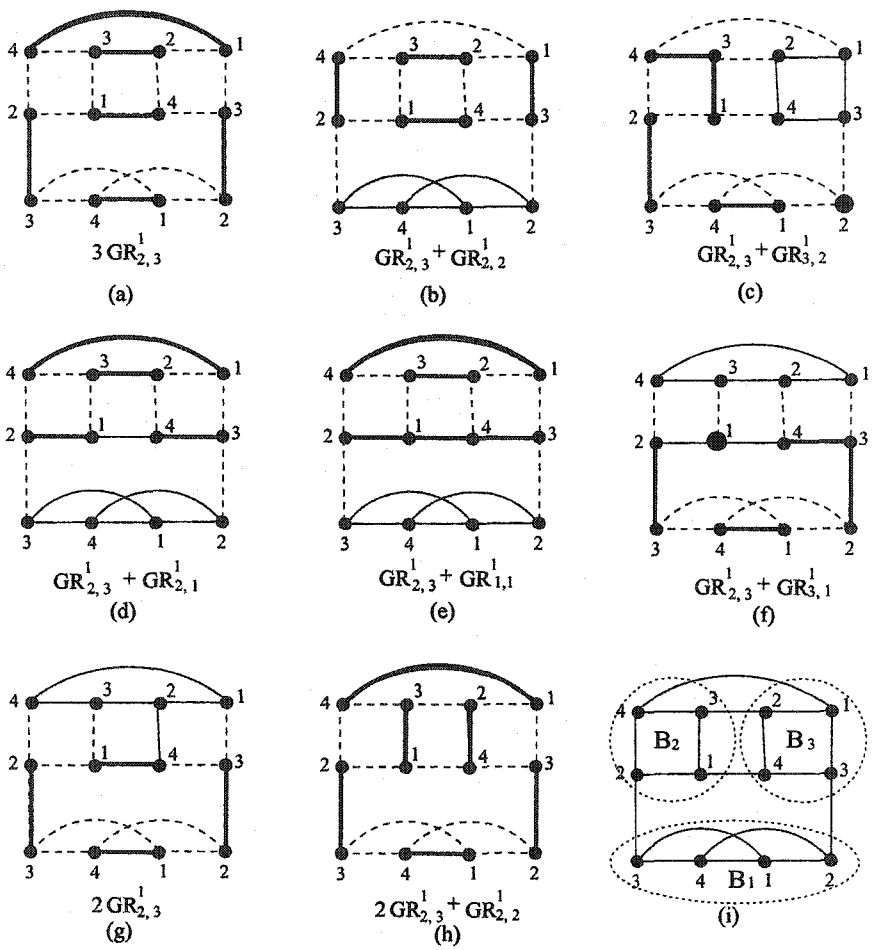


Figure 6.8: Feasible routings of $(4,3)$-PTRR's formed by three $(4,1)$-MPTRR's in H_3

routable for any $GR^1_{i,j}$ except $GR^1_{2,3}$. If R contains $GR^1_{2,3} + GR^1_{3,3}$, since $GR^1_{2,3} + GR^1_{3,2}$ is equivalent to $GR^1_{2,3} + GR^1_{3,3}$ under σ, R is routable in H_3. If R contains $GR^1_{2,3} + GR^1_{3,1}$, we may assume that R does not contain $GR^1_{2,3} + GR^1_{2,2}$ as it has been included in a previous case, then we can route R as in Figure 6.8(f) since the used 4-cycle with vertex label order 1,2,3,4, which is rouable for every any $GR^1_{i,j}$ except $GR^1_{2,2}$. If R contains $GR^1_{2,3}+GR^1_{3,4}$, then we know R is routable since $GR^1_{2,3}+GR^1_{3,4}$ is equivalent to $GR^1_{2,3} + GR^1_{3,1}$ under σ.

Subcase 2.4: R contains exactly two $GR^1_{2,3}$'s.

If we route $2GR^1_{2,3}$ as shown in Figure 6.8(g), then the unused 4-cycle has vertex label $1, 2, 3, 4$, which is routable for any of $GR^1_{i,j}$ except $GR^1_{2,2}$. Finally $R = 2GR^1_{2,3}+GR^1_{2,2}$, a feasible routing of R is shown in Figure 6.8(h).

Case 3: R contains a $GR^2_{i,j}$.

It suffices to consider the following $GR^2_{i,j}$'s which are not equivalent to each other under the permutation $\sigma = (1,4)(2,3)$:

$$GR^2_{1,1}, GR^2_{1,2}, GR^2_{1,3}, GR^2_{1,6}, GR^2_{2,1}, GR^2_{2,2},$$

$$GR^2_{2,3}, GR^2_{2,7}, GR^2_{2,8}, GR^2_{2,9}, GR^2_{3,1}, GR^2_{3,2}$$

If R does not contain $GR^1_{2,3}$, then we route R in H_3 as in Figure 6.9, note that the unused 4-cycle in these figures is routable for any $GR^1_{i,j}$ except $GR^1_{2,3}$. When R contains $GR^1_{2,3}$, then we route R as in Figure 6.10.

We conclude that H_3 is routable for all $(4,3)$-PTRR's. Hence H_3 is a $(4,3)$-HUSB. Since H_3 has 18 switches which is equal to the lower bound 6×3 on the number of switches of a $(4,3)$-HUSB, H_3 is an optimal $(4,3)$-HUSB.

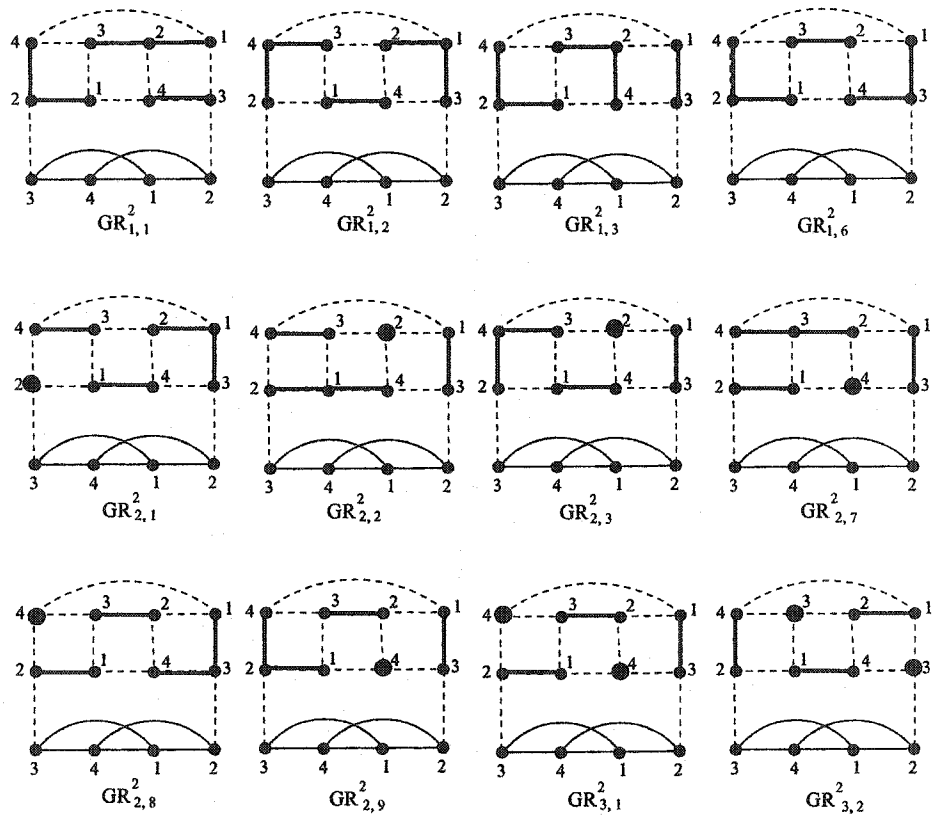


Figure 6.9: Feasible routings of $(4,3)$-PTRR's formed by one $(4,2)$-MPTRR and one $(4,1)$-MPTRR in H_3

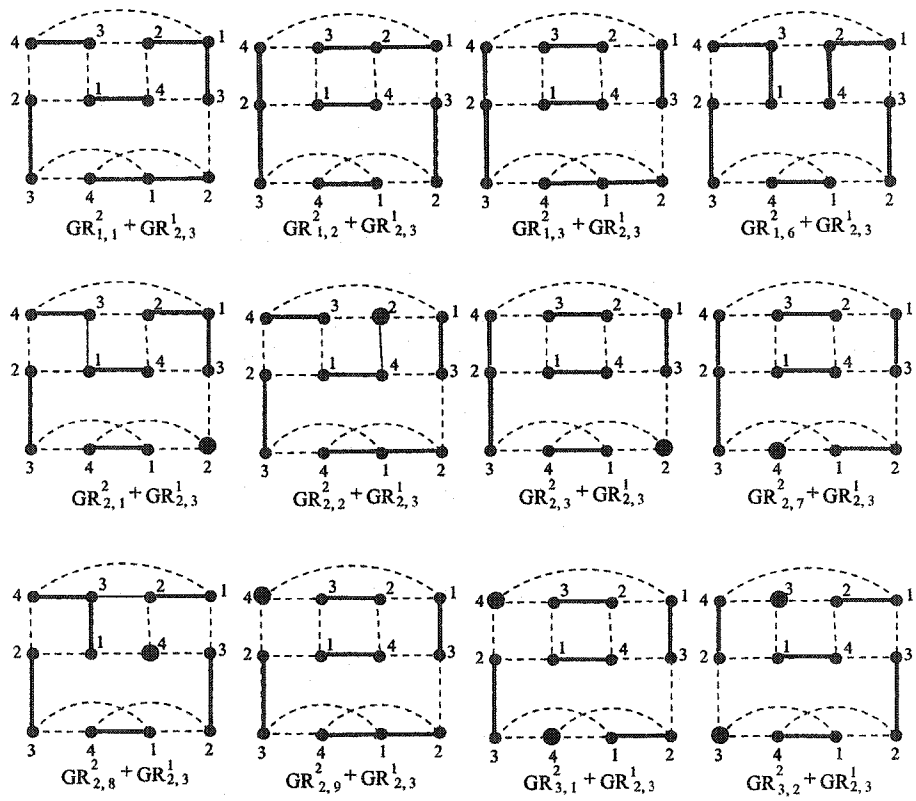


Figure 6.10: Continued feasible routings of $(4,3)$-PTRR's formed by one $(4,2)$-MPTRR and one $(4,1)$-MPTRR in H_3

H_4 is an optimal $(4,4)$-HUSB

Searching for an optimal $(4,4)$-HUSB started by testing candidate $(4,4)$-SB designs with 24 switches for hyper-universality. If such an optimal $(4,4)$-HUSB exists, then by Theorem 5.3.1, we know any two sides of G must induce a matching of four edges, so there are $4! = 24$ different matchings between any two sides. Hence there are total 24^6 different $(4,4)$-SB's. However, we only need to check non-isomorphic $(4,4)$-SB's, so we can choose a fixed matching between sides 1 and 2, sides 2 and 3, and sides 3 and 4. This reduces the number of candidate $(4,4)$-SB's to 24^3. Furthermore, any three sides of G must induce an optimal $(3,4)$-HUSB, which is isomorphic to either $G(4,4)$ or $G(4,2) + G(4,2)$ by Theorem 5.3.5. Therefore, we can only consider two matchings between sides 1 and 3, thus reducing the number of $(4,4)$-SB's to $2 \times 24 \times 24 = 1152$. We tested these candidates. This time we were not lucky, we tested every one of these 1152 $(4,4)$-SB's and none of them passed the test. That is, there exists a non-routable $(4,4)$-RR for each of them, thus we proved that no $(4,4)$-SB with 24 switches is hyper-universal. Detailed are omitted here. We then considered a $(4,4)$-SB with 25 switches. We started from adding a switch to two copies of H_2. We obtained H_4 and proved that H_4 is hyper-universal.

We now show that H_4 is hyper-universal. Let R be any $(4,4)$-PTRR which is a union of 4-MPTRR's. Since H_4 contains two copies of H_2 is a $(4,2)$-HUSB, Hence if R is a disjoint union of two $(4,2)$-PTRR's, it is routable in H_4. Therefore we need only consider the case when R is a union of a $GR^1_{i,j}$ and a $GR^3_{h,k}$.

If R does not contain $GR^1_{2,3}$, we can route the five $GR^3_{h,k}$'s first, as shown in Figure 6.11(a). Note that the unused 4-cycle in H_4 is routable for any

$GR^1_{i,j}$ except $GR^1_{2,3}$. If R contains a $GR^1_{2,3}$, then a feasible routing of R in H_4 is given in Figure 6.11(b). This proves that H_4 is hyper-universal.

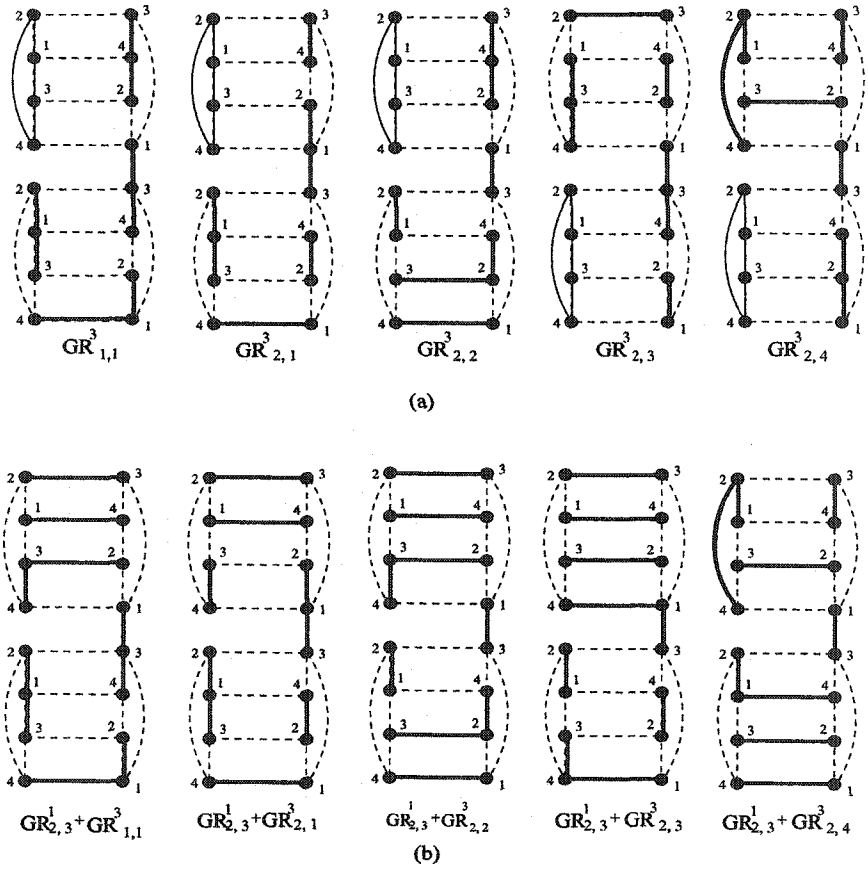


Figure 6.11: Feasible routings of $(4,4)$-PTRR's in H_4.

H_6 is hyper-universal

To prove that H_6 is hyper-universal we need to show that H_6 contains a feasible routing for every $(4,6)$-PTRR. Let R be such a $(4,6)$-PTRR. Then R can be decomposed into either two $(4,3)$-PTRR's or three $(4,2)$-

MPTRR's. In the first case, R is clearly routable in H_6 as H_6 contains two copies of H_3. In the second case, we show that R is also routable in H_6.

Let $R = R_1 + R_2 + R_3$, where each R_i $(i = 1, 2, 3)$ is a $(4, 2)$-MPTRR from Table 6.1.

For convenience, we partition the vertices of H_6 into six groups, every four vertices on the same horizontal line falling in the same group, and we label the groups from top to bottom. Let $L_{i,i+1}$ be the subgraph of H_6 induced by the vertices in ith and $(i+1)$th groups. Then H_6 has three disjoint subgraphs $L_{1,2}$, $L_{3,4}$ and $L_{5,6}$. By Figure 6.9, we see that $L_{1,2}$ and $L_{5,6}$ can route any minimal $(4, 2)$-PTRR from Table 6.1. Next we show that $L_{3,4}$ is routable for any $GR^2_{i,j}$. Again, we note that the permutation $\sigma = (1, 4)(2, 3)$ is an automorphism of H_6, so that it is sufficient to check those $GR^2_{i,j}$'s which are not equivalent under the permutation $\sigma = (1, 4)(2, 3)$. Figure 6.12 lists the feasible routings of these $GR^2_{i,j}$'s in $L(3, 4)$. In summary, we know that R_1, R_2, R_3 is routable in $L_{1,2}, L_{3,4}, L_{5,6}$ respectively. Therefore, R is routable in H_6.

Using $H_i, i = 1, 2, 3, 4, 5, 6, 7$ as prime 4-HUSB's, we construct a compound $(4, w)$-HUSB $H(w)$ as follows.

$$H(w) = \begin{cases} H_1 & \text{if } w = 1, \\ hH_6 & \text{if } w = 6h, \\ (h-1)H_6 + H_7 & \text{if } w = 6h+1, \\ hH_6 + H_2 & \text{if } w = 6h+2, \\ hH_6 + H_3 & \text{if } w = 6h+3, \\ hH_6 + H_4 & \text{if } w = 6h+4, \\ hH_6 + H_5 & \text{if } w = 6h+5. \end{cases}$$

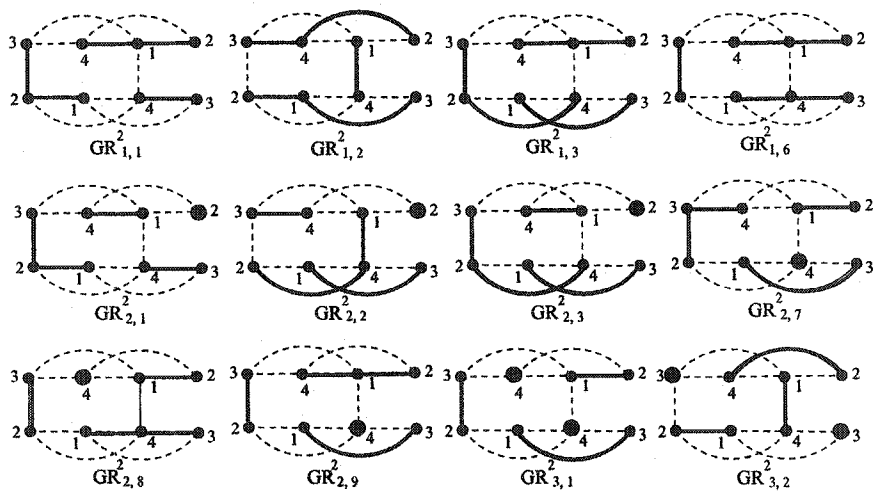


Figure 6.12: Feasible routings of $(4,6)$-PTTR's formed by three $(4,2)$-MPTRR's in H_6

The number of switches in $H(w)$ is equal to

$$|E(H(W))| = \begin{cases} 6 & \text{if } w = 1, \\ \frac{19}{3}w & \text{if } w = 0(mod\ 6), \\ \frac{19}{3}w - \frac{4}{3} & \text{if } w = 1(\text{mod } 6), \\ \frac{19}{3}w - \frac{2}{3} & \text{if } w = 2(\text{mod } 6), \\ \frac{19}{3}w - 1 & \text{if } w = 3(\text{mod } 6), \\ \frac{19}{3}w - \frac{1}{3} & \text{if } w = 4(\text{mod } 6), \\ \frac{19}{3}w - \frac{5}{3} & \text{if } w = 5(\text{mod } 6). \end{cases}$$

Therefore, $H(w)$ is hyper-universal with $\lceil \frac{19}{3}w \rceil$.

6.3 Comparison of Switch Block Designs

In this section, we compare the routing capabilities of disjoint $(4,w)$-SB's, $(4,w)$-USB, Wilton's $(4,w)$-SB's, and the $(4,w)$-HUSB's. We compare switch blocks with the same dimension and channel capacity. The com-

parisons can be done in three ways. The first way is to compare the set of routing requirements that are routable: a switch block G is said be better than switch block G' in terms of routing capability if the set of routing requirements routable in G properly contains that of G'. The second way is to compare the number of routing requirements routable: a switch block G is said to be better than switch block G' in terms of routing capacity if the number of routing requirements routable in G is greater than that of G'. The third way is to use the experimental methods: inserting a switch block design into an FPGA design platform, testing the benchmark circuits to compare the total number of tracks needed. The experimental comparisons for different designs have been done by several authors [14, 51, 24]. Our design of $(4, w)$-HUSB's was tested experimentally by Wu and Cheung [24], and the experimental results shows a 9% decrease in the number of tracks used as compared to that of disjoint $(4, w)$-SB's.

Next we compare the four $(4, w)$-SB designs in terms of routing capability. For convenience, we use $RC(G)$ to denote the set of tight routing requirements routable in switch block G.

When $w = 1$, all the four designs are isomorphic to a complete $(4, 1)$-SB. Furthermore, we can see that they are isomorphic to each other when $w = 2$ (see Figure 6.13). Therefore when $w = 1$ and 2, the $(4, 2)$-USB, Wilton's $(4, 2)$-SB and our $(4, 2)$-HUSB are isomorphic each other.

The four designs differ from each other when $w = 3$. The disjoint $(4, 3)$-SB is isomorphic to $3G(4, 1)$, and the symmetric $(4, 3)$-USB is isomorphic to $G(4, 2) + G(4, 1)$. Both Wilton's $(4, 3)$-SB and the $(4, 3)$-HUSB H_3 are connected.

It is clear that $RC(3G(4, 1))$ consists of those $(4, 3)$-TRR's which can

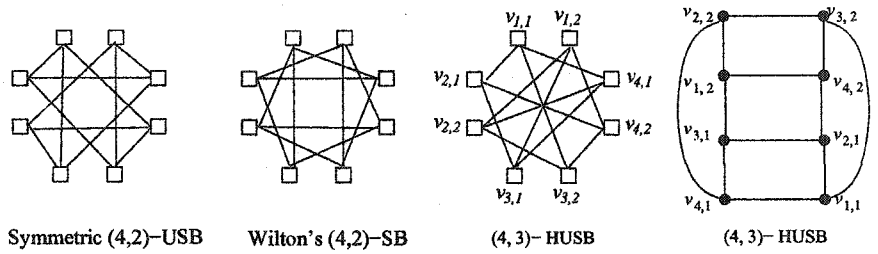


Figure 6.13: Symmetric $(4,2)$-USB, Wilton's $(4,2)$-SB and our$(4,2)$-HUSB.

be decomposed into three $(4,1)$-TRR's. $RC(G(4,2)+G(4,1))$ consists of those $(4,3)$-TRR's which can be decomposed into a $(4,2)$-TRR and a $(4,1)$-TRR. Since the union of two $(4,1)$-TRR's makes one $(4,2)$-TRR and there are $(4,2)$-TRR's which can not be decomposed into two $(4,1)$-TRR's, $RC(3G(4,1))$ is properly contained in $RC(G(4,2)+G(4,1))$. Since there are $(4,3)$-TRR's which can not be decomposed into a $(4,2)$-TRR and a $(4,1)$-TRR, $RC(G(4,2)+G(4,1))$ is properly contained in $RC(H_3)$.

Table 6.2 shows the routing properties of some $(4,3)$-TRR's in the four $(4,3)$-SB's. The reason that $GR^1_{2,1}+2GR^1_{2,3}$ is not routable in Wilton's $(4,3)$-SB is that in any feasible routing for $GR^1_{2,1}+2GR^1_{2,3}$, the two edges corresponding to $GR^1_{2,1}$ must be in a 4-cycle, but no such 4-cycle exists in Wilton's $(4,3)$-SB. Therefore, RC(Wilton's (4,3)-SB) is properly contained in $RC(H_3)$. From the table, we know that there are no containment relations between $RC(3G(4,1))$ and RC(Wilton's (4,3)-SB), $RC(G(4,2)+G(4,1))$ and RC(Wilton's (4,3)-SB).

The same containment relations also hold for the four $(4,w)$-SB designs with $w \geq 4$. That is, $RC(wG(4,1))$ is a proper subset of $RC(\frac{w-1}{2}G(4,2)+G(4,1))$ ($RC(\frac{w}{2}G(4,2))$ when w is even). $RC(\frac{w-1}{2}G(4,2)+G(4,1))$ is a proper subset of RC((4,w)-HUSB), and RC(Wilton's (4,w)-SB) is a proper

Routable in (4, 3)-SB?	Disjoint SB	Wilton's SB	USB	HUSB
$GR^1_{2,1} + 2GR^1_{2,3}$	YES	NO	YES	YES
$GR^2_{1,3} + GR^1_{1,2}$	NO	YES	YES	YES
$GR^3_{1,1}$	NO	YES	NO	YES

Table 6.2: Special routing instances.

subset of RC((4,w)-HUSB). RC(Wilton's (4,w)-SB) has no containment relations with $RC(wG(4,1))$ and with $RC(\frac{w-1}{2}G(4,2) + G(4,1))$.

Considering connectivity, the disjoint $(4,w)$-SB has w components, the $(4,w)$-USB has $\lceil w/2 \rceil$ components, and the $(4,w)$-HUSB has $\lceil w/6 \rceil$ components, and Wilton's $(4,w)$-SB is connected, i.e. it has one component.

Now consider feasible routing algorithms. We know that a feasible routing can be found in time linear in w for the compound $(4,w)$-HUSB's. But no linear time algorithm for determining the existence of a feasible routing of any given (4, 2)-TRR is known for the disjoint $(4,w)$-SB's, the $(4,w)$-USB's, or Wilton's $(4,w)$-SB's. The disadvantages of the $(4,w)$-HUSB's are that they use more switches than the of other three kinds of switch blocks.

6.4 Conclusions and Future Work

In this chapter, we applied the decomposition design scheme to the design of $(4,w)$-HUSB's. The prime 4-HUSB's are $(4,r)$-HUSB's with $r = 1,2,3,4,5,6,7$. We presented and verified a group prime 4-HUSB designs, which achieves optimality for $r = 1,2,3,4,5$ and near optimality for $r = 6$ and 7. For $w \geq 7$ we showed a compound $(4,w)$-HUSB with $\lceil 6.34w \rceil$ switches.

From the design and verification of the prime 4-HUSB's, we see that it is hard to verify that a switch block is a hyper-universal switch block when it is not a compound switch block. The difficulty is due to the lack of an efficient way to find a feasible routing for a given routing requirement. Besides, there are a great many routing requirements which need to be checked.

Since the prime $(4, 6)$-HUSB is used as the repeated component in a compound $(4, w)$-HUSB, it would be valuable to know an optimal $(4, 6)$-HUSB. We have tested some $(4, 6)$-SB's with 36 and 37 switches, but none was found to be hyper-universal. So the optimal $(4, 6)$-HUSB design problem is still open. We would like to know if H_6 is an optimal $(4, 6)$-HUSB.

Historical notes :

The $(4, w)$-HUSB design problem was the first concrete problem I have worked on in this area. The first prime design was obtained in the summer of 1999 (joint with J. Liu). In the summer of 2000, J. Liu invited me to the University of Lethbridge for the second time. We improved the prime $(4, w)$-HUSB's for $w = 3, 4, 5, 6, 7$. My main contribution is in identifying these switch blocks from a large candidate set and verifying the hyper-universality of these switch blocks. J. Liu completed and simplified the proof of H_3 by using the symmetry. The proofs for H_3 and H_4 and H_6 here are based on Liu's simple proof.

Chapter 7

Universal Switch Blocks

In this chapter, we apply the decomposition design scheme to design (k, w)-USB's. Because of the special properties of 2-pin net (k, w)-PTRR's, better results are achieved. Since we will only consider 2-pin net (k, w)-PTRR's in this chapter, unless stated otherwise, by a (k, w)-PTRR we mean a 2-pin net (k, w)-PTRR.

Section 7.1 presents a decomposition theorem for 2-pin net (k, w)-PTRR's. It says that any 2-pin net $(k, 2m)$-PTRR can be decomposed into m $(k, 2)$-PTRR's, and that the maximum capacity of 2-pin net k-MPTRR's, $f_2(k)$ (i.e., the maximum degree of non-decomposable) regular 2-graphs on k vertices, is equal to 2 when $3 \leq k \leq 6$, or to $\frac{k+3-i}{3}$ when $k \geq 7$, where $1 \leq i \leq 6$ and $i \equiv n \pmod 6$. A detailed proof of this decomposition theorem is given in Section 7.4. This decomposition theorem implies that prime k-USB's are (k, w)-USB's with $w = 1, 2, 3, 5, \ldots, 2j + 1, \ldots, \frac{k+3-i}{3}$, and a disjoint union of m copies of a $(k, 2)$-USB is a $(k, 2m)$-USB. For odd $w \geq \frac{k+3-i}{3}$, a disjoint union of m copies of a $(k, 2)$-USB and one $(k, \frac{k+3-i}{3})$-USB forms a $(k, 2m + \frac{k+3-i}{3})$-USB. Section 7.2 shows that $G(k, 2)$ is an optimal $(k, 2)$-USB, and Section 7.3 discusses approximate designs for prime (k, w)-USB's with $w = 3, 5, \ldots, \frac{k+3-i}{3}$.

7.1 Decomposition Theorems for Universal Switch Blocks

In using the decomposition design scheme to design (k, w)-USB's for every $w \geq 1$, we need to enumerate all k-MPTRR's and compute $f_2(k)$. The following theorem gives a characterization for k-MPTRR's, which can be used to enumerate all k-MPTRR's.

Theorem 7.1.1 *Let R be a 2-pin net (k, q)-PTRR and G be the q-regular 2-graph on k vertices corresponding to R. Then R is a (k, q)-MPTRR if and only if one of the following statements holds:*
(1) G is 1-regular.
(2) G is 2-regular and at least one component of G is an odd cycle.
(3) q (≥ 3) is odd and there is a 2-factor-free component C in the graph obtained from G by deleting all cut edges, such that C is incident with at least $q-1$ cut edges of G if C contains a singleton with multiplicity one, and otherwise, at least q cut edges.

Theorem 7.1.2

$$f_2(k) = \begin{cases} 1, & 1 \leq k \leq 2, \\ 2, & 3 \leq k \leq 6, \\ \frac{k+3-i}{3}, & k \geq 7 \end{cases} \tag{7.1}$$

where $1 \leq i \leq 6$ and $i \equiv k \pmod 6$.

The proofs of the above theorems are in Section 7.4 below. Next we consider the 2-pin k-MPTRR's for $k = 1, \ldots, 8$. For $k = 1, 2, 3, 4$, we can easily find all 2-pin k-MPTRR's from the multi-pin k-MPTRR's given in Chapter 4 and 6. For $k = 5, 6, 7, 8$, we use Theorems 7.1.1 and 7.1.2 to compute $\mathcal{G}_2(k)$. We enumerated all non-isomorphic regular 2-graphs of k vertices

with degree $r = 1, 2, 3, \ldots, f_2(k)$ for $k = 5, \ldots, 8$. To simplify the representation of $\mathcal{G}_2(k)$, we choose one representative from each type of k-MPTRR's, and give a complete list of the different types. We use kGR_i^r to denote the ith type of a k-MPTRR with capacity r. For example, a complete list of 3-MPTRR types is $3GR_1^1 = \{\{1,2\},\{3\}\}$ and $3GR_1^2 = \{\{1,2\},\{2,3\},\{1,3\}\}$. A complete list of 2-pin net 4-MPTRR's types is $4R_1^1 = \{\{1,2\},\{3,4\}\}$ and $4GR_1^2 = \{\{1,2\},\{2,3\},\{3,1\},\{4\},\{4\}\}$. Figure 7.1 shows the 2-graphs of different types of 2-pin net k-MPTRR's for $k = 1, \ldots, 8$.

The following two theorems follow directly from Theorem 7.1.2 and the decomposition design scheme in Chapter 5.

Theorem 7.1.3 *Let R be a 2-pin net (k,w)-PTRR, then the following statements hold:*

(i) When w is even, R can be decomposed into $w/2$ $(k,2)$-PTRR's.

(ii) When $1 \leq k \leq 6$ and w is odd, R can be decomposed into one $(k,1)$-PTRR and $(w-1)/2$ $(k,2)$-PTRR's.

(iii) When $k \geq 7$ and w is odd and $w > \frac{k+3-i}{3}$, R can be decomposed into one $(k, \frac{k+3-i}{3})$-PTRR and $\frac{3w-k-3+i}{6}$ $(k,2)$-PTRR's, where $1 \leq i \leq 6$ and $i \equiv k \pmod 6$.

(iv) When $k \geq 7$ and w is odd and $w \leq \frac{k+3-i}{3}$, there exists a (k,w)-MPTRR for every pair of k, w with $k \geq 7$ and odd $w \leq \frac{k+3-i}{3}$, where $1 \leq i \leq 6$ and $i \equiv k \pmod 6$.

Theorem 7.1.4 *The following statements hold:*

(i) When w is even, a disjoint union of $w/2$ copies of a $(k,2)$-USB forms a (k,w)-USB.

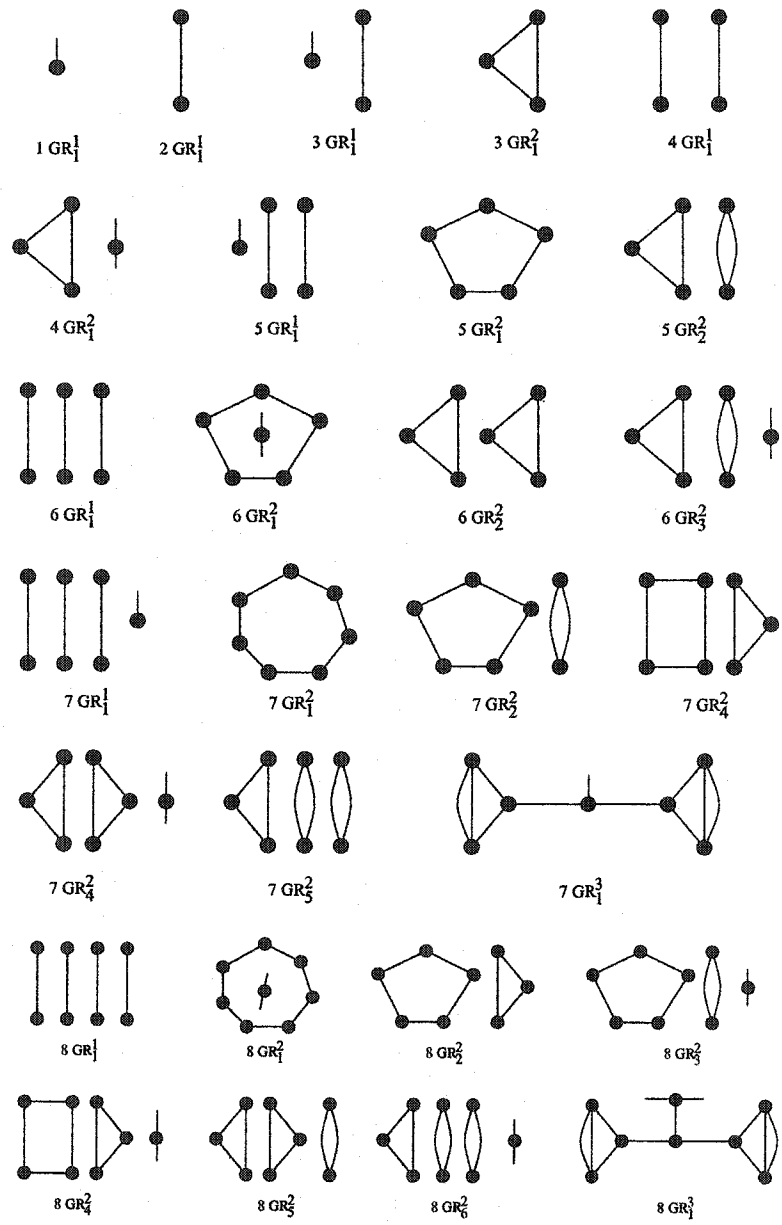


Figure 7.1: A complete list of the different types of 2-pin net k-MPTRR's for $k = 1, \ldots, 8$.

(ii) When $1 \leq k \leq 6$ *and* w *is odd, a disjoint union of one* $(k,1)$*-USB and* $(w-1)/2$ *copies of a* $(k,2)$*-USB forms a* (k,w)*-USB.*

(iii) When $k \geq 7$ *and* w *is odd and* $w > \frac{n+3-i}{3}$*, a disjoint union of one* $(k, \frac{k+3-i}{3})$*-USB and* $\frac{3w-k-3+i}{6}$ *copies of a* $(k,2)$*-USB forms a* (k,w)*-USB, where* $1 \leq i \leq 6$ *and* $i \equiv k \pmod 6$.

(iv) When $k \geq 7$ *and* w *is odd and* $w \leq \frac{k+3-i}{3}$*, the graph of a* (k,w)*-USB must be connected, where* $1 \leq i \leq 6$ *and* $i \equiv k \pmod 6$.

7.2 Prime Universal Switch Blocks with Capacities Two

By Theorem 7.1.4, when $1 \leq k \leq 6$ the prime k-USB's are (k,r)-USB's with $r = 1, 2$, when $k \geq 7$ the prime k-USB's are (k,r)-USB's with $r = 2$ and all odd numbers between 1 and $\frac{k+3-i}{3}$, where $1 \leq i \leq 6$ and $i \equiv k \pmod 6$. We know that the complete $(k,1)$-SB, $G(k,1)$, is an optimal $(k,1)$-USB for all k by Theorem 5.3.3. We will first show that $G(k,2)$ is an optimal $(k,2)$-USB for every $k \geq 2$ and then consider an approximate design for optimal prime (k,r)-USB's.

7.2.1 Optimal Universal Switch Blocks with Channel Capacity Two

By Definition 5.2.1, $G(k,2)$ is the k-partite graph $((V_1,\ldots,V_k),E)$ with $V_i = \{v_{i,j} | j = 1,2\}, i = 1,\ldots,k$ and

$$E = \cup_{1 \leq i < j \leq k} \{v_{i,h} v_{j,\ h+(j-i)-1} | h = 1,2\}$$

where the second index of the subscript is taken modulo 2.

Proposition 7.2.1 *Let* $i_1, i_2, \ldots, i_q$ *be* q *different integers with* $1 \leq i_j \leq k, j = 1,\ldots,q$. *Let* $G[V_{i_1}, V_{i_2}, \ldots, V_{i_q}]$ *denote the subgraph of* $G(k,2)$ *with vertex set* $V_{i_1} \cup V_{i_1} \cup \ldots \cup V_{i_q}$ *and edge set consisting of edges of* $G(k,2)$

with ends between V_{i_j} *and* $V_{i_{j+1}}$ *for* $j = 1, \ldots, q$, *where* $V_{i_{q+1}} = V_{i_1}$. *Then* $G[V_{i_1}, V_{i_2}, \ldots V_{i_q}]$ *is a Hamiltonian cycle if* q *is odd, and two disjoint cycles of equal length otherwise.*

Proof. $H = G[V_{i_1}, V_{i_2}, \ldots, V_{i_q}]$ is a 2-regular simple graph and thus, H consists of cycles.

A pair V_{i_j} and $V_{i_{j+1}}$ is called an increasing pair if $i_{j+1} > i_j$, and decreasing pair, otherwise. Let s and r be the numbers of increasing pairs and decreasing pairs of the sequence $V_{i_1}, V_{i_2}, \ldots, V_{i_q}, V_{i_1}$, respectively. Then, when starting at vertex $v_{i_1,t}$ and going along the cycle containing $v_{i_1,t}$, we will return to V_{i_1} at the vertex $v_{i_1,t+s-r}$. If $v_{i_1,t+s-r} \neq v_{i_1,t}$, then we can continue this process, until we end up at $v_{i_1,t}$. Now we have $t = t + m(s-r) (\text{mod } 2)$, and so that $m(s-r) = 0 \pmod 2$, where m is equal to either 1 or 2. Since $s + r = q$, if q is odd then $s - r$ is odd and $m = 2$ and H is a Hamiltonian cycle. Otherwise $m = 1$ and H consists of two cycles of length q. □

Theorem 7.2.2 $G(k, 2)$ *is an optimal* $(k, 2)$*-USB for every* $k \geq 2$.

Proof. Let R be any $(k, 2)$-PTRR. We show that R has a feasible routing in $G(k, 2)$. Let H be the 2-graph of R. Then each component of H is a cycle. Let $\{i_1, i_2\}, \ldots, \{i_q, i_1\}$ be any component of H. If h is odd, then $G[V_{i_1}, V_{i_2}, \ldots, V_{i_q}]$ is a Hamiltonian cycle of length $2q$ and it has two sets of h independent edges, any one of which can be used as a feasible routing of $\{i_1, i_2\}, \ldots, \{i_q, i_1\}$. If q is even, then $G[V_{i_1}, V_{i_2}, \ldots, V_{i_q}]$ is a union of two disjoint cycles of equal lengths and it has four sets of q independent edges. Two of which can be used as a feasible routing of $\{i_1, i_2\}, \ldots, \{i_q, i_1\}$. Thus R has a feasible routing in $G(k, 2)$, and therefore $G(k, 2)$ is universal.

Furthermore, $G(k,2)$ has $2\binom{k}{2}$ edges. Hence by Theorem 5.3.2, $G(k,2)$ is an optimal $(k,2)$-USB. □

Now we define

$$U(k,w) = \begin{cases} h\ G(k,2) & \text{if } w = 2h, \\ G(k,1) + h\ G(k,2) & \text{if } w = 2h+1. \end{cases} \tag{7.2}$$

Recall that $h\ G(k,2)$ denotes a disjoint union of h copies of $G(k,2)$.

Theorem 7.2.3 *$U(k,w)$ is an optimal (k,w)-USB when w is even, or when w is odd and $k \leq 6$.*

Proof. By Theorems 7.1.3 and 7.2.2, when w is even, every (k,w)-PTRR can be decomposed into $\frac{w}{2}$ $(k,2)$-PTRR's and each of which has a feasible routing in a $G(k,2)$. Therefore, $U(k,w)$ is universal. Since $|E(U(k,w))| = \binom{k}{2}w$ is equal to the lower bound on the number of switches in a $(k,2)$-USB by Theorem 5.3.2, $U(k,w)$ is an optimal (k,w)-USB.

When $k \leq 6$ and w is odd, by Theorem 7.1.4(ii) and Theorems 7.2.2, we know that $U(k,w)$ is universal. Since $U(k,w)$ has $\binom{k}{2}w$ switches, it is an optimal (k,w)-USB. □

In [46], Shyu et al. claimed that the (k,w)-SB, $M_{k,w}$, generated by their algorithm is an optimal (k,w)-USB. However, $M_{k,w}$ is isomorphic to $U(k,w)$ and $U(k,w)$ is not universal when $k \geq 7$ and w (≥ 3) is odd. For example, $U(7,3)$ is not routable for $7GR_1^3$ because $7GR_1^3$ can not be decomposed into one $(7,1)$-PTRR and one $(7,2)$-PTRR. In fact, $U(k,w)$ is a (k,w)-USB if and only if $2 \leq k \leq 6$ or w is even, see [27].

We find that designing optimal prime (k,w)-USB's for $k \geq 7$ and odd w is not an easy task. Therefore, we try to design prime (k,w)-USB's which are close to optimal.

7.3 Prime Universal Switch Blocks with Odd Capacities

We now consider designing prime (k, w)-USB's for $k \geq 7$ and odd w with $3 \leq w \leq f_2(k)$. We propose an iterative design method, in which we start from $G(k, 1)$ and add a $G(k, 2)$ and some extra switches such that the resulting switch block is universal and continue this process until a $(k, f_2(k))$-USB is derived. We here illustrate this method by constructing a $(k, 3)$-USB for $k = 7$, i.e., a $(7, 3)$-USB.

The basic idea of designing a $(7, 3)$-USB is to add some switches to $G(7, 1) + G(7, 2)$ such that the resulting $(7, 3)$-SB is universal. If a $(7, 3)$-PTRR is a disjoint union of a $(7, 1)$-PTRR and a $(7, 2)$-PTRR, then it is routable in $G(7, 1) + G(7, 2)$. Therefore, we consider adding the minimum number of switches between $G(7, 1)$ and $G(7, 2)$ (called *bridge switches*) such that the resulting $(7, 3)$-SB $\bar{U}(7, 3)$ is routable for all $(7, 3)$-MPTRR's, that is, routable for every $(7, 3)$-MPTRR isomorphic to $7GR_1^3$.

Let R be a $(7, 3)$-MPTRR. Then, in any routing for R in $\bar{U}(7, 3)$, at least one bridge switch is used. We consider a feasible routing of R which uses exactly one bridge switch. Suppose that $\{i_1, i_2\}$ is the net in R using the bridge switch in the feasible routing, and i_1 corresponds to a vertex v in $G(7, 1)$. Then we must use three independent edges in $G(7, 1) - \{v\}$ to route three mutually disjoint 2-pin nets in $R \setminus \{i_1, i_2\}$. We see that such an $\{i_1, i_2\}$ must corresponds to an edge in a triangle of the 2-graph of R. One of the smallest (in terms of number of edges) graphs on seven vertices which will always contain a triangle edge of any 2-graph isomorphic to a $(7, 3)$-MPTRR is given in Figure 7.2(b). We call this graph a *connection pattern graph*. The labels of the vertices and the orientations of the edges in the connection pattern graph are arbitrary. A directed edge (i, j) in the

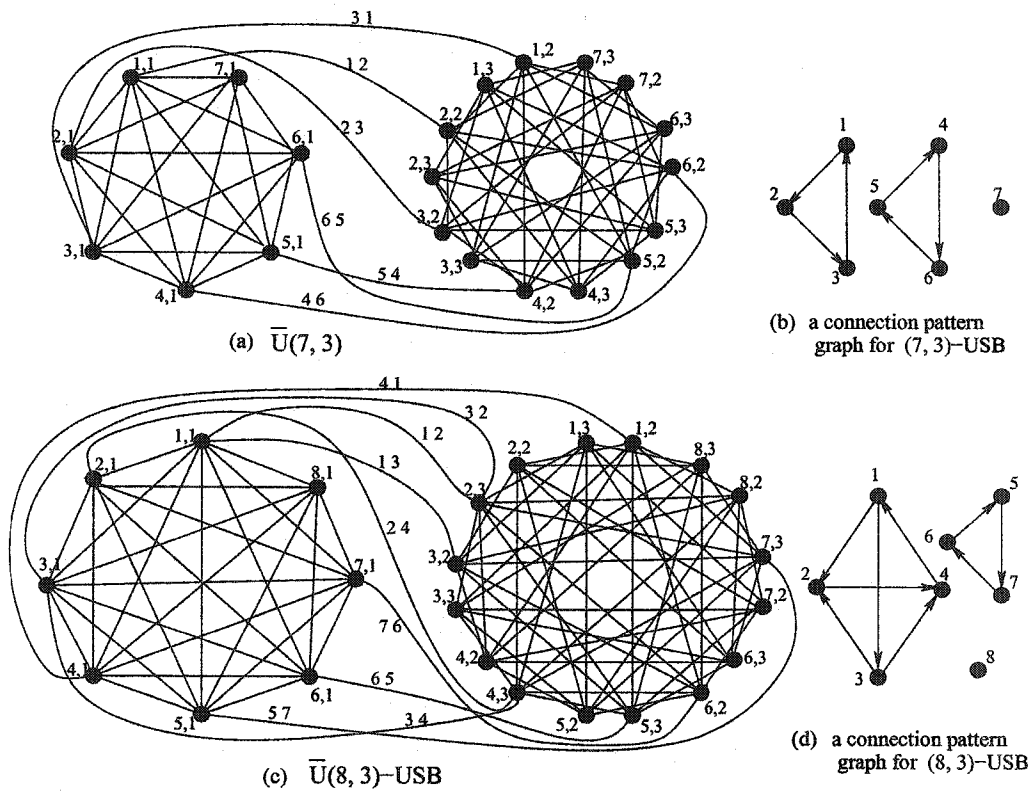


Figure 7.2: A (7,3)-USB and a (8,3)-USB.

connection pattern graph corresponds to a bridge switch joining a terminal on the i-th side of $G(7,1)$ to a terminal on the j-th side of $G(7,2)$. Using this connection pattern graph, we obtain a $(7,3)$-SB, $\bar{U}(7,3)$, as shown in Figure 7.2(a).

We next show that $\bar{U}(7,3)$ is universal. We need only show that $\bar{U}(7,3)$ is routable for every $(7,3)$-MPTRR R (recall that R is isomorphic to $7R_1^3$). Let $\{i_1, i_2\}$ be a net corresponding to a triangle edge in the connection pattern graph. Without loss of generality, assume that $v_{i_1,1}$ and $v_{i_2,2}$ are the two end terminals of the bridge switch corresponding to $\{i_1, i_2\}$ in $\bar{U}(7,3)$, and where i_1 corresponds to $v_{i_1,1}$.

Let N_1, N_2, N_3 be three mutually disjoint 2-pin nets in R, none containing i_1. We can find routings F_1 for N_1, N_2, N_3 in in $G(7,1) - v_{i_1,1}$. Now consider $R' = (R \setminus \{\{i_1, i_2\}, N_1, N_2, N_3, \{i_3\}) \cup \{\{i_2, i_3\}\}$, where $\{i_3\}$ is the singleton in R. Then R' corresponds to a 2-regular graph, which has a feasible routing F_2 in $G(7,2)$, such that a switch s' realizing $\{i_2, i_3\}$ is incident with $v_{i_2,2}$. Then $((F_1 \cup F_2) \setminus \{s'\}) \cup \{v_{i_1,1}v_{i_2,2}, \{v_{i_3,2}\}\}$ is a feasible routing of R. Therefore $\bar{U}(7,3)$ is a $(7,3)$-USB.

Similarly, we construct a $\bar{U}(8,3)$, see Figure 7.2(c) by the connection pattern graph shown in Figure 7.2(d). Using the prime $\bar{U}(7,3)$ and $\bar{U}(8,3)$ we construct $(7,w)$-USB's and $(8,w)$-USB's as follows. For $w = 2h+1$, define

$$\bar{U}(7, 2h+1) = \bar{U}(7,3) + \textstyle\sum_{i=1}^{h-1} G(7,2) \tag{7.3}$$

$$\bar{U}(8, 2h+1) = \bar{U}(8,3) + \textstyle\sum_{i=1}^{h-1} G(8,2). \tag{7.4}$$

Theorem 7.3.1 *Let w be an odd integer with $w \geq 3$. Then $\bar{U}(7,w)$ is a*

$(7,w)$-USB with approximation ratio $\frac{|E(\bar{U}(7,w))|}{min_{USB}(7,w)} \leq 1+\frac{6}{21w}$. *And* $\bar{U}(8,w)$ *is a* $(8,w)$*-USB with approximation ratio* $\frac{|E(\bar{U}(8,w))|}{min_{USB}(8,w)} \leq 1+\frac{9}{28w}$.

Proof. By our construction, we see that $\bar{U}(7,3)$ is a $(7,3)$-USB. By Theorem 7.1.2, $f_2(7) = 3$. Then by Theorem 7.1.3-(iii), every $(7,w)$-PTRR with odd $w \geq 5$ can be decomposed into $\frac{w-3}{2}$ $(7,2)$-PTRR's and one $(7,3)$-PTRR, and each $(7,2)$-PTRR has a feasible routing in a copy of $G(7,2)$, and the $(7,3)$-PTRR has a feasible routing in $\bar{U}(7,3)$. Therefore $\bar{U}(7,w)$ is universal. By Theorem (5.3.2), we have

$$|E(\bar{U}(7,w))| = \binom{7}{2}w + 6 = 21w + 6, \quad min_{USB}(7,w) \geq \binom{7}{2}w = 21w$$

Therefore,

$$\frac{|E(\bar{U}(7,w))|}{min_{USB}(7,w)} \leq 1+\frac{6}{21w}.$$

Similarly, it can be shown that $\bar{U}(8,w)$ is universal with approximation ratio

$$\frac{|E(\bar{U}(8,w))|}{min_{USB}(8,w)} \leq 1+\frac{9}{28w}.$$

□

We see that when w is large, the ratio is close to 1. Hence $\bar{U}(7,w)$ and $\bar{U}(8,w)$ are close to optimal when w is large.

For $k \geq 9$, we can use a similar method as long as the all k-MPTRR's are known. First, we need to design (k,w)-USB's for each odd integer $3 \leq w \leq f_2(k)$. The construction starts from $w = 3$. Construct a $(k,3)$-USB by adding bridge switches between $G(k,1)$ and $G(k,2)$ such that the resulting $(k,3)$-SB, written $\bar{U}(k,3)$, is routable for all $(k,3)$-MPTRR's. Then construct $\bar{U}(k,5)$ by adding bridge switches between $\bar{U}(k,3)$ and $G(k,2)$ such that the resulting $(k,5)$-SB is routable for all $(k,5)$-MPTRR's. We

continue this process until $\bar{U}(k, \frac{k+3-i}{3})$ is constructed, where $1 \leq i \leq 6$ and $i \equiv k \pmod 6$.

Then we use these universal switch blocks to build a (k, w)-USB for any odd w. When $w \leq \frac{k+3-i}{3}$, we use $\bar{U}(k, w)$ (recall that w is odd), otherwise $w > \frac{k+3-i}{3}$, let $\bar{U}(k, w)$ be the disjoint union of one $\bar{U}(k, \frac{k+3-i}{3})$ and $\frac{w-\frac{k+3-i}{3})}{2}$ copies of $G(k, 2)$. Then, by Theorem 7.1.3 we know that $\bar{U}(k, w)$ is universal. By Theorem 5.3.2, we have the approximation ratio of $\bar{U}(k, w)$

$$\frac{|E(G)|}{min_{USB}(k, w)} \leq \frac{\binom{k}{2}(w + (\frac{k+3-i}{3}))^2 - \frac{k+3-i}{3}))}{\binom{k}{2}w} \leq 1 + \frac{k(k-1)(k^2-k-1)}{w} \sim 1$$

when w is large and k is fixed. Therefore, $\bar{U}(k, w)$ is close to optimal when w is large.

7.3.1 Feasible Routing Algorithms

We now discuss finding feasible routings in the above compound k-USB's. Let $M = \frac{w-f_2(k)}{2}G(k, 2) + \bar{U}(k, f_2(k)$ be a compound (k, w)-USB obtained by the above decomposition design scheme and let R be any (k, w)-PTRR. We claim that there is a polynomial time (in k and w) algorithm to find a feasible routing of R in M. Let G' be the regular 2-graph corresponding to R. Since there is a polynomial time algorithm (in the number of vertices and edges) to find a 2-factor of a graph [35], there exists a polynomial time algorithm (in k and w) decomposing G' into a $f_2(k)$-factor and $\frac{w-f_2(k)}{2}$ 2-factors. The $f_2(k)$-factor is a $(k, f_2(k))$-PTRR, for which and a feasible routing can be found in $\bar{U}(k, f_2(k))$ in constant time. Each of the $\frac{w-f_2(k)}{2}$ 2-factors is a $(k, 2)$-PTRR, for which a feasible routing can be found in $G(k, 2)$ in constant time. Therefore we can find a feasible routing for R

in M in time polynomial in k and w. Recall that for a compound (k, w)-HUSB, when k is fixed, we can find a feasible routing for a multi-pin net (k, w)-PTRR in time linear in w, but the constant depends on k and may be exponential.

7.4 Proofs for Decomposition Theorems

In this section, we give detailed graph theoretical proofs for Theorems 7.1.1 and 7.1.2. All graphs considered are multiple graphs. Let $G = (V, E)$ be a graph, and $D, S \subseteq V$ be disjoint subsets. $G - D$ denotes the graph obtained from G by deleting all vertices in D and all edges incident with D, and $d_{G-D}(x)$ stands for the degree of vertex x in graph $G - D$. A *component* of a graph is a maximal connected subgraph. A component of $G - D$ is said to be an *odd component* if it has an odd number of vertices. Let $E_G(S, D)$ denote the set of edges of G having one end in S and the other in D, and $e_G(S, D)$ denote the number of edges in $E_G(S, D)$, namely $e_G(S, D) = |E_G(S, D)|$. Let $E' \subseteq E$, denote by $G - E'$ the subgraph of G obtained from G by removing all edges in E'. In particular, $G - e$ denotes the graph obtained from G by removing edge e. An edge e of graph G is called a *cut edge* if the number of components in $G - e$ is greater than that in G. The following known results will be used in our proofs.

Lemma 7.4.1 (Tutte, 1947, [50]) *A graph G has a 1-factor if and only if for any $S \subseteq V(G)$,*

$$o(G - S) \leq |S| \tag{7.5}$$

where $o(G - S)$ denotes the number of odd components of $G - S$.

Lemma 7.4.2 (Petersen, [40, 2]) *Every 3-regular graph without a cut edge has a 1-factor.*

Lemma 7.4.3 (Petersen, [40, 2, 11]) *Any $2k$-regular ($k \geq 2$) graph can be decomposed into k disjoint 2-factors. That is, a $2k$-regular graph is 2-factorable.*

Lemma 7.4.4 (The k-factor Theorem, [10]) *A graph G contains a k-factor if and only if*

$$k|D| - q(D,S) - \sum_{x \in S}(k - d_{G-D}(x)) \geq 0 \tag{7.6}$$

for all disjoint sets $D, S \subseteq V(G)$, where $q(D,S)$ denotes the number of components C in $G - D - S$ such that $e_G(S, V(C)) + k|V(C)|$ is odd.

The next lemma is due to Schonberger (1934) as stated in [2]. However, we proved the result before knowing of its earlier existence. We here provide our own proof using Tutte's 1-factor theorem.

Lemma 7.4.5 *Let G be a 3-regular graph without a cut edge, and let e_1, e_2 be any two edges of G. Then $G - \{e_1, e_2\}$ contains a 1-factor.*

Proof. Suppose $G - \{e_1, e_2\}$ does not contain a 1-factor. Then by Lemma 7.4.1, there exists an $S \subseteq V(G - \{e_1, e_2\})$ such that $o(G - \{e_1, e_2\} - S) > |S|$. Let $C_1, \ldots, C_t$ be the odd components of $G - \{e_1, e_2\} - S$ where $t = o(G - \{e_1, e_2\} - S)$. Then we have

$$t \geq |S| + 1, \quad 3|S| \geq e_G(C_1 \cup \ldots \cup C_t, S) \tag{7.7}$$

By Lemma 7.4.2 we have $o(G - S) \leq |S|$ since G has a 1-factor. We must have that either (1): e_1 or e_2 join two odd components of $G' - S$, or (2): e_1

and e_2 join two different odd components and an even component. Suppose (2) is the case and without loss of generality, assume that e_1 joins the odd component C_1 and an even component C, and e_2 joins C_2 and C. Since $e_G(C_1, V \setminus V(C_1))$ is odd and greater than one, $e_G(C_1, S) \geq 2$. Similarly, $e_G(C_2, S) \geq 2$. For $i \geq 3$, we have $e_G(C_i, S) \geq 3$ as G does not have a cut edge. Then by (7.7), we have $3|S| \geq e_G(C_1 \cup C_2 \cup C_3 \cup \cdots \cup C_t, S) \geq 2 + 2 + 3(t-2) = 3t - 2 \geq 3(|S|+1) - 2 = 3|S| + 1$, a contradiction.

Next we suppose (1) is the case, i.e., either e_1 or e_2 joins two odd components of $G - \{e_1, e_2\} - S$. Without loss of generality, we assume that e_1 joins C_1 and C_2. We prove the contradiction by five cases according to the location of e_2 in G.

Case 1: e_2 joins C_1 and C_2.

Then $e_G(C_1 \cup C_2, S)$ must be even and $e_G(C_1 \cup C_2, S) \geq 2$. If $e_G(C_1 \cup C_2, S) \geq 4$, then by (7.7), we have $3|S| \geq e_G(C_1 \cup \ldots \cup C_t, S) \geq 4 + 3(t-2) = 3t - 2 \geq 3(|S|+1) - 2 = 3|S| + 1$, a contradiction. Therefore, $e_G(C_1 \cup C_2, S) = 2$. Then by (7.7) we have $3|S| \geq e_G(C_1 \cup \ldots \cup C_t, S) \geq 2 + 3(t-2) = 3t - 4 \geq 3(|S|+1) - 4 = 3|S| - 1$. If there is a C_i with $i \geq 3$ and $e_G(C_i, S) \geq 5$, then $3|S| \geq e_G(C_1 \cup \ldots \cup C_t, S) \geq 2 + 3(t-3) + 5 \geq 3|S| + 1$, a contradiction. Therefore, $e_G(C_i, S) = 3$ for $i = 3, \ldots, t$. Then $3|S| \geq e_G(C_1 \cup \ldots \cup C_t, S) = 2 + 3(t-2) = 3t - 4 \geq 3(|S|+1) - 4 = 3|S| - 1$. This implies that $t = |S| + 1$ and that there is one more edge e incident with S. But since G has no cut edge, both ends of e must be in S. Simple counting shows that the number of vertices of G must be odd, which is a contradiction since G is 3-regular implies that G has an even number of vertices.

Case 2: e_2 joins one of C_1 and C_2 to a C_i for some i with $3 \leq i \leq t$.

Without loss of generality, suppose e_2 joins C_2 to C_3. Since $e_{G-\{e_1,e_2\}}(C_i, S)$ is odd and $|V(C_i)|$ is odd and G is 3-regular and G has no cut edge, there are an even number of edges between C_i and S for $i = 1, 3$ and an odd number of edges between C_2 and S and at least three edges between C_i and S for $i = 4, \ldots, t$, so that $e_G(C_1, S) \geq 2$, $e_G(C_3, S) \geq 2$, $e_G(C_2, S) \geq 1$, and $e_G(C_i, S) \geq 3$ for $i = 4, \ldots, t$. Then by (7.7), we have $3|S| \geq e_G(C_1 \cup \ldots \cup C_t, S) \geq 2+1+2+3(t-3) = 3t-4 \geq 3(|S|+1)-4 = 3|S|-1$. This implies $t = |S|+1$. We obtain a contradiction as in the argument at the end of Case 1.

Case 3: e_2 joins C_i and C_j with $3 \leq i < j \leq t$.

Without loss of generality, suppose that e_2 joins C_3 and C_4. By the previous arguments, we know that $e_G(C_i, S) \geq 2$ for $i = 1, 2, 3, 4$ and $e_G(C_i, S) \geq 3$ for $i = 4, \ldots, t$. If $e_G(C_i, S) \geq 4$ for some i with $1 \leq i \leq 4$, or $e_G(C_i, S) \geq 5$ for some i with $5 \leq i \leq t$, then by (7.7) we have $3|S| \geq e_G(C_1 \cup \ldots \cup C_t, S) \geq 3t-2 \geq 3(|S|+1)-2 = 3|S|+1$, a contradiction. Therefore, $e_G(C_i, S) = 2$ for $1 \leq i \leq 4$, and $e_G(C_i, S) = 3$ for $5 \leq i \leq t$. Then by (7.7) we have $3|S| \geq e_G(C_1 \cup \ldots \cup C_t, S) = 8+3(t-4) = 3t-4 \geq 3(|S|+1)-4 \geq 3|S|-1$, which implies $t = |S|+1$, again a contradiction by the argument at the end of Case 1.

Case 4: e_2 joins an odd component and an even component.

Suppose e_2 joins a C_{i_0} with $1 \leq i_0 \leq t$ and an even component C of $G - \{e_1, e_2\}$, where $|V(C)|$ is even.

If $3 \leq i_0 \leq t$ then we have $e_G(C, S) \geq 1$, and $e_G(C_{i_0}, S) \geq 2$. Then by (7.7) $3|S| \geq e_G(C_1 \cup \ldots \cup C_t \cup C, S) \geq 3(t-3)+2+2+2+1 = 3t-2 \geq 3(|S|+1)-2 = 3|S|+1$, a contradiction. If $i_0 = 1$, then $e_G(C_1, S) \geq 1$, and we have $3|S| \geq e_G(C_1 \cup \ldots \cup C_t \cup C, S) \geq 3(t-2)+2+1+1 = 3t-2 \geq 3|S|+1$,

a contradiction. Similarly, a contradiction follows if $i_0 = 2$.

Case 5: e_2 is not incident with any odd component.

Then e_2 joins two even components, or e_2 is in an even component, or e_2 joins S and a component of $G - S$, e_2 joins two vertices of S. In any of the above cases, we have $3|S| \geq e_G(G - S, S) > 3(t-2) + 2 + 2 = 3t - 2 \geq 3(|S| + 1) - 2 \geq 3|S| + 1$, a contradiction.

□

The following theorem describes a structural property of regular graphs which we have not found elsewhere.

Theorem 7.4.6 *Let G be a $(2r+1)$-regular graph ($r \geq 1$). If the number of cut edges in G is less than $2r+1$, then G has a 2-factor.*

Proof. We first prove the correctness for $r = 1$. Suppose G is a 3-regular graph with k cut edges with $k \leq 2$, we show that G contains a 2-factor.

Suppose $k = 0$, then G has a 1-factor by Lemma 7.4.2. Removing the 1-factor from G, we obtain a 2-factor of G, so that G contains a 2-factor.

Suppose $k = 1$, by the above argument it is suffices to show G has a 1-factor. Suppose on the contrary that G does not contain a 1-factor. Then by Lemma 7.4.1, there exists an $S \subseteq V(G)$ such that $o(G - S) > |S|$. We note that for each odd component C of $G - S$, $e_G(C, S)$ must be odd, and there is at most one odd component C with $e_G(C, S) = 1$, and for other components C, $e_G(C, S) \geq 3$ since G has only one cut edge. Then $3(o(G - S) - 1) + 1 \leq 3|S|$ or $3o(G - S) - 2 \leq 3|S|$. This implies that $o(G - S) \leq |S|$, a contradiction to $o(G - S) > |S|$. Therefore G has a 1-factor, and so that G has a 2-factor.

Next suppose $k = 2$. Suppose on the contrary that G does not contain

a 2-factor. After the deletion of the two cut edges from G, the resulting graph must have a component not containing a 2-factor, let it be H.

We see that the two cut edges must be both incident with H, for otherwise we can obtain a graph H' by joining two copies of H with one cut edge. H' has a 2-factor by the above argument for the case $k = 1$, which would induce a 2-factor of H, contradicting the choice of H. Let u_1 and u_2 be the two vertices in H incident with the two cut edges of G. Then u_1 and u_2 must be distinct since otherwise G would have more than two cut edges. Then we have $d_H(u_1) = d_H(u_2) = 2$. Let K be the 3-regular graph obtained from H by deleting u_1 and u_2 and adding edge e'_i joining the neighbors of u_i, $i = 1, 2$. Note that the two neighbors of a u_i must be distinct since otherwise G would have more than two cut edges. Then H contains a 2-factor if and only if K contains a 2-factor containing both e'_1 and e'_2, and if and only if K has a 1-factor containing neither e'_1 nor e'_2. Since K is 3-regular and does not contain a cut edge, by Lemma 7.4.5 we know K has a 1-factor, therefore, H has a 2-factor. This contradicts to the choice of H. The above contradiction implies that G has a 2-factor. Therefore the theorem is true for $r = 1$.

Next, we suppose that $r \geq 2$. We show that G has a 2-factor. By Lemma 7.4.4, it is suffices to show that, for any two disjoint sets $D, S \subseteq V(G)$,

$$2|D| - q(D, S) - \sum_{x \in S} (2 - d_{G-D}(x)) \geq 0. \tag{7.8}$$

where $q(D, S)$ denotes the number of components, C, of $G - D - S$ such that $e_G(V(C), S) + 2|V(C)|$ is odd.

Let $\Delta = 2|D| - q(D,S) - \sum_{x \in S}(2 - d_{G-D}(x))$, thus

$$\begin{aligned}
\Delta &= 2|D| - q(D,S) - \sum_{x \in S}(2 - d_{G-D}(x)) \\
&= 2|D| - q(D,S) - 2|S| + \sum_{x \in S} d_{G-D}(x) \\
&= 2|D| - q(D,S) - 2|S| + (2r+1)|S| - e_G(D,S) \\
&= 2|D| - 2|S| + (2r+1)|S| - e_G(D,S) - q(D,S),
\end{aligned}$$

Thus showing that G has a 2-factor is equivalent to showing that

$$\Delta = 2|D| - 2|S| + (2r+1)|S| - e_G(D,S) - q(D,S) \geq 0 \tag{7.9}$$

where $q(D,S)$ equals to the number of components, C, of $G - D - S$ such that

$$e_G(V(C),S) \text{ is odd. (this implies that } e_G(V(C),S) \geq 1) \tag{7.10}$$

Let A be the set of components C of $G - D - S$ such that $e_G(V(C),S)$ is odd and $e_G(V(C),D) = 0$. Let

$$A_1 = \{C \in A \mid e_G(V(C),S) = 1\}$$

$$A_2 = \{C \in A \mid e_G(V(C),S) \geq 3\}$$

Then for each $C \in A_1$, the unique edge joining C and S is a cut edge of G, so that

$$|A_1| \leq 2r \tag{7.11}$$

because G does not have more than $2r$ cut edges by the assumption of the theorem. It is obvious that $A_1 \cap A_2 = \emptyset$, so that $A = A_1 \cup A_2$ and $|A| = |A_1| + |A_2|$.

Let B be the set of components, C, of $G-D-S$ such that $e_G(V(C),S)$ is odd and $e_G(V(C),D) \geq 1$. Then for each $C \in B$, $e_G(V(C),S) \geq 1$. Since G is $(2r+1)$-regular, we have

$$(2r+1)|D| \geq e_G(D,S) + e_G(\cup_{C\in B}V(C),D) \geq e_G(D,S) + |B|. \quad (7.12)$$

Then $A \cap B = \emptyset$ and

$$q(D,S) = |A| + |B| = |A_1| + |A_2| + |B|. \quad (7.13)$$

Since G is $(2r+1)$-regular, we have

$$\begin{aligned}
(2r+1)|S| &= \textstyle\sum_{v\in S} d_G(v) \\
&\geq e_G(V(G)\setminus S, S) \\
&\geq e_G(D \cup (\cup_{C\in A_1\cup A_2\cup B}V(C)), S) \\
&= e_G(D,S) + e_G(\cup_{C\in A_1}V(C),S) \\
&\quad + e_G(\cup_{C\in A_2}V(C),S) + e_G(\cup_{C\in B}V(C),S) \qquad (7.14) \\
&\geq e_G(D,S) + |A_1| + 3|A_2| + |B| \\
&\geq e_G(D,S) + |A_1| + |A_2| + |B| \\
&\geq e_G(D,S) + |A| + |B| \\
&= e_G(D,S) + q(D,S).
\end{aligned}$$

There are three possible cases according to the relations of S and D.

Case 1: $|D| \geq |S|$.

By (7.9) and (7.14) we have

$$\Delta = 2(|D| - |S|) + [(2r+1)|S| - e_G(D,S) - q(D,S)] \geq 0.$$

Case 2: $|D| < |S|$ and $2|S| - 2|D| \leq |A| - (2r-1)$.

In this case we have $|A| \geq (2r-1) + 2(|S| - |D|) \geq 2r+1$. Then by (7.11) we yield

$$|A_2| = |A| - |A_1| \geq |A| - 2r \geq 2r + 1 - 2r = 1. \tag{7.15}$$

$$\begin{aligned}
\Delta &= 2|D| - 2|S| + (2r+1)|S| - e_G(D,S) - (|A| + |B|) \\
&= (2r+1)|S| - e_G(D,S) - (|A| + |B| + 2|S| - 2|D|) \\
&\geq (2r+1)|S| - e_G(D,S) - (|A| + |B| + |A| - (2r-1)) \quad \text{(by Case 2)} \\
&= (2r+1)|S| - e_G(D,S) - (2|A| + |B|) + (2r-1) \\
&= (2r+1)|S| - e_G(D,S) - 2(|A_1| + |A_2|) - |B| + (2r-1) \\
&= ((2r+1)|S| - e_G(D,S) - |A_1| - 3|A_2| - |B|) + 2r - 1 - |A_1| + |A_2| \\
&\geq 2r - 1 - |A_1| + |A_2| \quad \text{(by (7.14))} \\
&= (2r - |A_1|) + (|A_2| - 1) \\
&\geq 0 \quad \text{(by (7.11) and (7.15)).}
\end{aligned}$$

Case 3: $|D| < |S|$ and $2|S| - 2|D| > |A| - (2r-1)$.

In this case we have $2|S| - 2|D| - 1 \geq |A| - (2r-1)$.

If $|S| - |D| \geq 2$, then

$$\begin{aligned}
\Delta &= 2|D| - 2|S| + (2r+1)|S| - e_G(D,S) - (|A| + |B|) \\
&= 2|D| - 2|S| + (2r+1)|S| - e_G(D,S) - (|A| - (2r-1)) - |B| - (2r-1) \\
&\geq 2|D| - 2|S| + (2r+1)|S| - e_G(D,S) - (2|S| - 2|D| - 1) - |B| - (2r-1) \\
&= (2r-3)|S| - (e_G(D,S) + |B|) + 4|D| - (2r-2) \\
&\geq (2r-3)|S| - (2r+1)|D| + 4|D| - (2r-3) - 1 \quad \text{(by (7.12))} \\
&= (2r-3)(|S| - |D| - 1) - 1 \\
&\geq 0. \quad \text{(since } r \geq 2 \text{ and } |S| - |D| \geq 2.\text{)}
\end{aligned}$$

If $|A|$ is odd, then $2|S| - 2|D| - 2 \geq |A| - (2r-1)$ and

$$\begin{aligned}
\Delta &= 2|D| - 2|S| + (2r+1)|S| - e_G(D,S) - (|A| + |B|) \\
&= 2|D| - 2|S| + (2r+1)|S| - e_G(D,S) - (|A| - (2r-1)) - |B| - (2r-1) \\
&\geq 2|D| - 2|S| + (2r+1)|S| - e_G(D,S) - (2|S| - 2|D| - 2) - |B| - (2r-1) \\
&= (2r-3)|S| - (e_G(D,S) + |B|) + 4|D| - (2r-3) \\
&\geq (2r-3)|S| - (2r+1)|D| + 4|D| - (2r-3) \quad \text{(by (7.12))} \\
&= (2r-3)(|S| - |D| - 1) \\
&\geq 0 \quad \text{(since } r \geq 2 \text{ and } |D| < |S|.)
\end{aligned}$$

If $e_G(D,S) + |B| \leq (2r+1)|D| - 1$, then

$$\begin{aligned}
\Delta &= 2|D| - 2|S| + (2r+1)|S| - e_G(D,S) - (|A| + |B|) \\
&= 2|D| - 2|S| + (2r+1)|S| - e_G(D,S) - (|A| - (2r-1)) - |B| - (2r-1) \\
&\geq 2|D| - 2|S| + (2r+1)|S| - e_G(D,S) - (2|S| - 2|D| - 1) - |B| - (2r-1) \\
&= (2r-3)|S| - (e_G(D,S) + |B|) + 4|D| - (2r-2) \\
&\geq (2r-3)|S| - ((2r+1)|D| - 1) + 4|D| - (2r-2) \\
&\geq (2r-3)(|S| - |D| - 1) \\
&= 0 \quad \text{(since } r \geq 2 \text{ and } |D| < |S|.)
\end{aligned}$$

Otherwise, we have $|S| - |D| \leq 1$ (contrary to $|S| - |D| \geq 2$), which implies

$$|S| = |D| + 1 \tag{7.16}$$

(since $|D| < |S|$ in this case), and $|A|$ is even, and $e_G(D,S) + |B| \geq (2r+1)|D|$ (contrary to $e_G(D,S) + |B| \leq (2r+1)|D| - 1$), by which and (7.12) we have

$$e_G(D,S) + |B| = (2r+1)|D|. \tag{7.17}$$

If $|A| \leq 2r-1$, then we have

$$\begin{aligned}\Delta &= 2|D| - 2|D| - 2 + (2r+1)|D| + 2r + 1 - e_G(D,S) - (|A| + |B|) \\ &= (2r+1)|D| + 2r - 1 - e_G(D,S) - (|A| + |B|) \\ &= 2r - 1 - |A| \quad \text{(by (7.17))} \\ &\geq 0.\end{aligned}$$

Otherwise we have $|A| \geq 2r$. We are going to show that this case does not occur. First of all, by (7.17) and (7.14) and (7.16), we have $(2r+1)|D|+|A| = e_G(D,S)+|B|+|A| \leq (2r+1)|S| = (2r+1)|D|+2r+1$, so that $|A| \leq 2r+1$. Since $|A|$ is even, $|A| \leq 2r$. Then by the assumption $|A| \geq 2r$ we have

$$|A| = 2r \tag{7.18}$$

and by (7.17)

$$e_G(D,S) + |B| + |A| = (2r+1)|D| + 2r = (2r+1)|S| - 1 \tag{7.19}$$

This implies that $A_2 = \emptyset$ since otherwise $|A_2| \geq 1$ then by (7.14) and (7.19), $(2r+1)|S| \geq e_G(D,S)+|B|+|A_1|+3|A_2| = e_G(D,S)+|B|+|A|+2|A_2| = (2r+1)|S|-1+2|A_2| \geq (2r+1)|S|+1$, a contradiction follows. Therefore, $A_2 = \emptyset$, so that $A = A_1$ and $|A| = |A_1| = 2r$. Hence S and the components in A_1 are joined by exactly $2r$ cut edges and $e_G(\cup_{C\in A}V(C), S) = |A|$.

This also implies that for each $C \in B$ $e_G(C,S) = 1$ since otherwise we must have $e_G(C,S) \geq 3$ and $(2r+1)|S| \geq e_G(D \cup (\cup_{C\in A\cup B}V(C)), S) = e_G(D,S) + e_G(\cup_{C\in A}V(C),S) + e_G(\cup_{C\in B}V(C),S) \geq e_G(D,S) + |A| + |B| + 2 = (2r+1)|S| + 1$ by (7.19), which again is a contradiction. Hence $e_G(\cup_{C\in B}V(C)),S) = |B|$. Therefore

$$e_G(D \cup (\cup_{C\in A\cup B}V(C)), S) = e_G(D,S) + |A| + |B|. \tag{7.20}$$

Since $\sum_{v\in S} d_G(v) = (2r+1)|S| > (2r+1)|S|-1 = e_G(D,S)+|B|+|A| = e_G(D\cup(\cup_{C\in A\cup B}V(C)),S)$ by (7.19) and (7.20), there is one edge $e = uv \notin E_G(D\cup(\cup_{C\in A\cup B}V(C)),S)$, incident with S. Suppose that $u \in S$. If v is not in S, v must be in a component C' of $G-D-S$ such that $e_G(V(C'),S)$ is even. Then $e_G(V(C'),S) \geq 2$ and $(2r+1)|S| \geq e_G(D\cup(\cup_{C\in A\cup B}V(C)),S)+e_G(V(C'),S) \geq e_G(D,S)+|B|+|A|+2 = (2r+1)|S|+1$ by (7.20) and (7.19), a contradiction. Therefore, $v \in S$. Now $(2r+1)|S| = \sum_{v\in S} d_G(v) \geq e_G(D\cup(\cup_{C\in A\cup B}V(C)),S)+2 = e_G(D,S)+|B|+|A|+2 = (2r+1)|S|+1$ by (7.20) and (7.19), again a contradiction.

In all cases, we have proved that $\Delta \geq 0$, therefore, G has a 2-factor by Lemma 7.4.4. □

Next we move on to 2-graphs. Recall that a 2-graph is primitive if it has at most one vertex incident with singleton edges, no singletons $\{x\}$ and $\{y\}$ such that $x \neq y$. The following lemma is obvious.

Lemma 7.4.7 *Let G be a 2-graph, and $\bar{G}$ be a primitive 2-graph obtained from G by successively substituting two singletons $\{x\}$ and $\{y\}$ with $x \neq y$ by an edge $\{x,y\}$ until no such pair of singletons exist. Then G has a 2-factor if $\bar{G}$ does and G is 2-factorable if $\bar{G}$ is.*

The result of Lemma 7.4.3 can be generalized to 2-graphs.

Theorem 7.4.8 *A $2k$-regular 2-graph ($k \geq 2$) has a 2-factor and is 2-factorable.*

Proof. It suffices to show that a $2k$-regular 2-graph ($k \geq 2$) always has a 2-factor. Let G be a $2k$-regular 2-graph with $k \geq 2$. If G does not contain

a singleton, then by Lemma 7.4.3, G has a 2-factor. Next suppose that G contains singletons. Let $\bar{G}$ be a primitive 2-graph obtained from G by primitive operations as described in Lemma 7.4.7. Then $\bar{G}$ contains at most one vertex contained in singletons.

If $\bar{G}$ does not contain a singleton, then $\bar{G}$ has a 2-factor by Lemma 7.4.3, and so that G has a 2-factor by Lemma 7.4.7.

Suppose that $\bar{G}$ has a singleton $\{x\}$ with multiplicity $p(x) > 0$. Since $p(x) + 2e = 2k|V(\bar{G})|$ is even, $p(x)$ must be even, where e is the number of edges of size two in $\bar{G}$. If $p(x) = 2k$, then $\bar{G}-x$ is a $2k$-regular graph and has a 2-factor by Lemma 7.4.3, and so also that G has a 2-factor. Otherwise, we have $2 \leq p(x) \leq 2k-2$. Delete the $p(x)$ singletons $\{x\}$'s from $\bar{G}$ and add three new vertices x_1, x_2, x_3 and new edges xx_1, x_1x_2, x_1x_3 and x_2x_3 with multiplicities $p(x)$, $(2k - p(x))/2$, $(2k - p(x))/2$ and $(2k + p(x))/2$, respectively. We obtain a $2k$-regular graph G'. Again by Lemma 7.4.3, G' has a 2-factor, say F. If F does not contain edge xx_1, then the restriction of F on G' is a 2-factor of $\bar{G}$. Otherwise F contains edge xx_1, then the multiplicity of xx_1 in F must be two copies of x_2x_3. Now the restriction of F on G' plus two copies of $\{x\}$ is a 2-factor of $\bar{G}$. Therefore, $\bar{G}$ has a 2-factor, and so does G by Lemma 7.4.7. □

Proof of Theorem 7.1.1

In the following, we work on 2-graphs. Recall that a 2-graph is a hypergraph with edge size at most two. A 2-graph is a graph when it does not contains singleton edges. The definitions of a path, a component, a cycle, a cut edge in a 2-graph are the same as that in a graph by only considering

edges of size two (i.e. edges incident with two vertices). For convenient, two singletons incident with the same vertex is considered as an even cycle. We describe the Theorem 7.1.1 in terms of primitive 2-graphs as follows.

Theorem *A primitive q-regular 2-graph G is non-decomposable if and only if one of the following statements holds:*

(1) G is 1-regular.

(2) G is 2-regular and contains an odd cycle component.

(3) q (≥ 3) is odd and there is a 2-factor-free component C in the graph, obtained from G by deleting all cut edges, such that C is incident with at least $q-1$ cut edges of G if C contains a singleton with multiplicity one, and otherwise, at least q cut edges.

Proof. Let G be a primitive q-regular 2-graph. When $q = 1$, it is easy to see that G is non-decomposable if and only if G is a 1-regular 2-graph. When $q = 2$, then G consists of some disjoint cycles, or possible two singletons incident with a vertex. Since a cycle contains a 1-factor if and only if it is an even cycle, G is non-decomposable if and only if it contains an odd cycle component.

Next we suppose $q \geq 3$. By Theorem 7.4.8, we may suppose that $q = 2r+1, r \geq 1$. Consider the graph obtained from G by deleting all cut edges. It must contain a 2-factor-free component C since otherwise G would contain a 2-factor.

We show that C is incident with at least $2r$ cut edges of G if C has a singleton with multiplicity one, and otherwise at least $2r+1$ cut edges.

Let $e_1, \ldots, e_m$ be all the cut edges incident to C and let $v_1, \ldots, v_m$ be the vertices in C incident with $e_1, \ldots, e_m$ respectively. Note that $v_1, \ldots, v_m$ may not be all distinct.

If C has a singleton $\{x\}$ with multiplicity one. We construct a $(2r+1)$-regular graph C' as follows. First, for each e_i $(i = 1, \ldots, m)$, add three new vertices x_i, y_i, z_i, and edges $v_i x_i$, $(r)x_i y_i$, $(r)x_i z_i$, $(r+1)y_i z_i$, where the notation $(h)e$ means h copies of e. Second, delete $\{x\}$ and add three new vertices w_1, w_2, w_3, and edges xw_1, $(r)w_1w_2$, $(r)w_1w_3$, $(r+1)w_2w_3$. It can be seen that C' does not contain a 2-factor since otherwise C would contain a 2-factor. Hence C' is a 2-factor-free $(2r+1)$-regular graph with $m+1$ cut edges. Then by Theorem 7.4.6, we have $m+1 \geq 2r+1$, or $m \geq 2r$. Therefore C is incident with at least $2r = q-1$ cut edges of G.

Otherwise, C does not contain a singleton with multiplicity one. We construct a $(2r+1)$-regular graph C' from C as follows. For each v_i $(i = 1, \ldots, m)$, we add three new vertices x_i, y_i, z_i, and edges $v_i x_i$, $(r)x_i y_i$, $(r)x_i z_i, (r+1)y_i z_i$. Furthermore, if C has a singleton $\{v\}$ with odd multiplicity $p(v)$, then $3 \leq p(v) < q$. We delete $p(v)$ $\{v\}$s and add three new vertices y, z, w to and edges vy, yz, yw and zw with multiplicities $p(v)$, $\frac{q-p(v)}{2}$, $\frac{q-p(v)}{2}$ and $\frac{q+p(v)}{2}$, respectively. If C has a singleton $\{v\}$ with even multiplicity $p(v)$ (≥ 2), we delete $p(v)$ $\{v\}$'s and then add two new vertices y, z and new edges vy, vz, yz with multiplicities $\frac{p(v)}{2}$, $\frac{p(v)}{2}$ and $q - \frac{p(v)}{2}$ respectively.

Then C' is a 2-factor-free $(2r+1)$-regular graph with m cut edges. Therefore $m \geq 2r+1$ by Theorem 7.4.6, namely, C is incident with at least $2r+1 = q$ cut edges.

To complete the proof of Theorem 7.1.1, we need only show that condition (3) is also sufficient for G to be non-decomposable. Suppose that (3) holds for a component C of the cut edge deleted subgraph. Suppose on the contrary that G has a proper regular factor. Since q is odd, G must have

an even regular factor and hence a 2-factor F by Theorem 7.4.8. Observe that, as G is primitive, F does not contain any cut edge and if F contains a singleton $\{v\}$, it must contain two copies of it. This observation implies that the restriction F on $V(C)$ is a 2-factor of C. This is a contradiction. Therefore, G is non-decomposable. □

Proof of Theorem 7.1.2

Proof. By definition, $f_2(n)$ is the maximum degree of non-decomposable regular 2-graphs on n vertices. It is easy to check that $f_2(1) = 1$ and $f_2(2) = 1$. When $3 \leq n \leq 6$, it can be seen that $f_2(n) \geq 2$ because a disjoint union of a 3-cycle (a cycle of three vertices) and a 2-regular 2-graph of $n - 3$ vertices gives a non-decomposable 2-regular 2-graph of n vertices.

Let n_0 be the minimum integer such that $f_2(n_0) \geq 3$ and G_0 be a non-decomposable 3-regular 2-graph with n_0 vertices. By Lemma 7.4.7, we assume that G_0 is primitive. Then one of the following three cases must hold.

Case 1: G_0 contains no singletons.

By Theorem 7.1.1, G_0 has a 2-factor-free component C incident with at least three cut edges, each of the components that is joined by a cut edge to C has at least three vertices, therefore $n_0 \geq |V(C)| + 3 \times 3 \geq 10$.

Case 2: G_0 contains a singleton $\{x\}$ with multiplicity at least two.

If the multiplicity of $\{x\}$ is three, then delete x from G_0, and we obtain a 3-regular non-decomposable graph with $n_0 - 1$ vertices. Hence we have $n_0 - 1 \geq 10$ or $n_0 \geq 11$. If the multiplicity of $\{x\}$ is two, then we remove two copies of $\{x\}$, add two extra vertices y, z and edges xy, xz, $(2)zy$. We

obtain a non-decomposable 3-regular graph. Then we have $n_0 + 2 \geq 10$ or $n_0 \geq 8$.

Case 3: G_0 contains a singleton $\{x\}$ with multiplicity one.

We remove $\{x\}$ and add three extra vertices y, z, w and edges xy, yz, yw, $(2)zw$, obtaining a non-decomposable 3-regular graph. Then we have $n_0 + 3 \geq 10$ or $n_0 \geq 7$.

Therefore, n_0 is at least 7. This implies that $f_2(n) \leq 2$ when $3 \leq n \leq 6$. Therefore $f_2(n) = 2$ for $3 \leq n \leq 6$.

Next we show that $f_2(n) \geq \frac{n+3-i}{3}$ for $n \geq 7$, where $1 \leq i \leq 6$ and $i \equiv n \pmod 6$. We first define a $\frac{n+3-i}{3}$-regular 2-graph on n vertices for each $n \geq 7$ as follows.

Case 1: When $n = 6r + 4, r \geq 1$.

Let $M(6r+4, 2r+1)$ be the graph of n vertices $\{v_0\} \cup \{v_{i,j} | i = 1, 2, \ldots, 2r+1, j = 1, 2, 3\}$, edges $\{v_0 v_{i,1} | i = 1, 2, \ldots, 2r+1\} \cup (r)\{v_{i,1}v_{i,2}, v_{i,1}v_{i,3} | i = 1, 2, \ldots, 2r+1\} \cup (r+1)\{v_{i,2}v_{i,3} | i = 1, 2, \ldots, 2r+1\}$, where $(r)S$ denotes the multiple set of r copies of S (see Figure 7.3(a)). Then $M(6r+4, 2r+1)$ is a $(2r+1)$-regular graph with $2r+1$ cut edges. We have $2r+1 = \frac{n+3-4}{3}$.

Case 2: When $n = 6r + 1, r \geq 1$.

We construct a 2-graph $M(6r + 1, 2r + 1)$ by deleting one triangle $\{v_{2r+1,1}, v_{2r+1,2}, v_{2r+1,3}\}$ from the graph $M(6r + 4, 2r + 1)$, and adding a singleton edge $\{v_0\}$ (see Figure 7.3(b)), where a circle around a vertex indicates a singleton). Then $M(6r + 1, 2r + 1)$ has $n = 6r + 1$ vertices and $2r + 1 = \frac{(6r+1)+2}{3} = \frac{n+3-1}{3}$.

Case 3: When $n = 6r + 2, r \geq 1$.

We construct a 2-graph $M(6r+2, 2r+1)$ by deleting the edge $\{v_0\}$ from the graph $M(6r + 1, 2r + 1)$, adding a new vertex v_{6r+2}, an edge $v_0 v_{6r+2}$

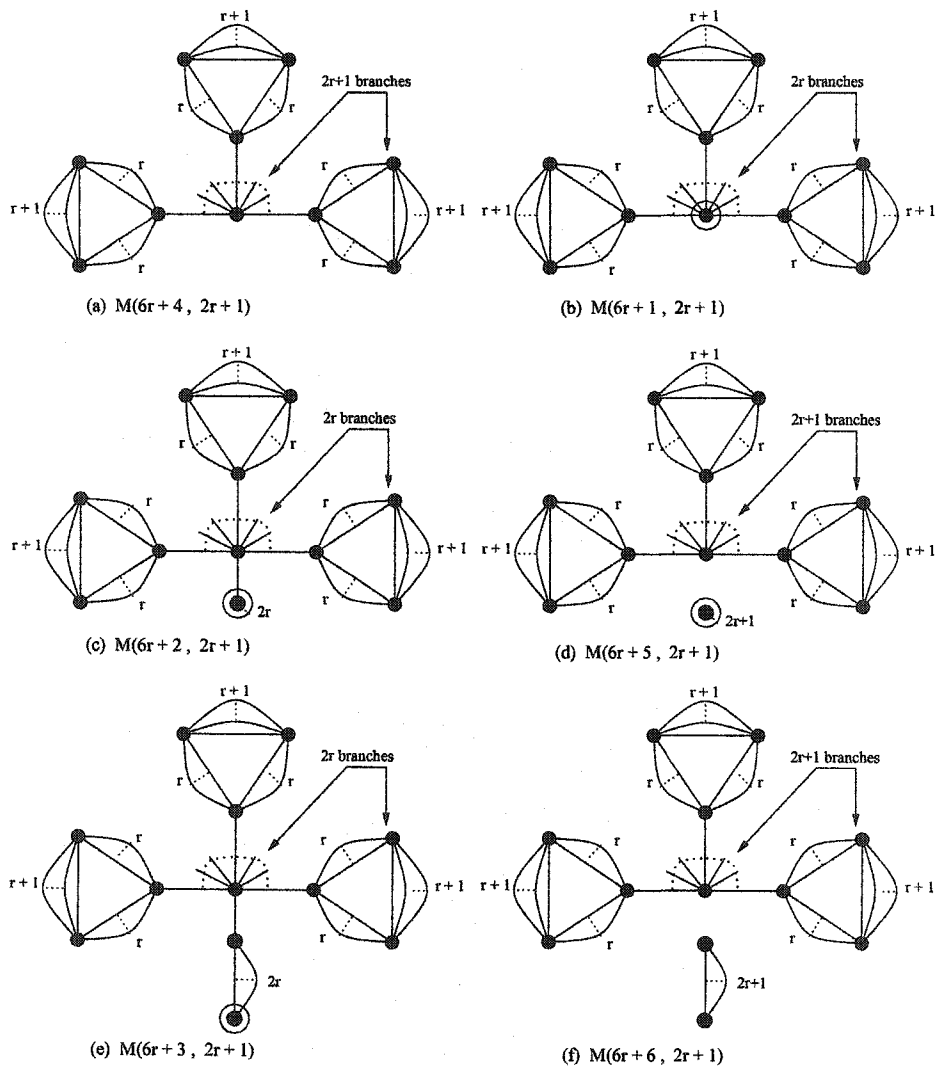


Figure 7.3: Non-decomposable $\frac{n+3-i}{3}$-regular 2-graphs

and $2r$ copies of $\{v_{6r+2}\}$ (see Figure 7.3(c)). Then $M(6r+2, 2r+1)$ has $n = 6r+2$ vertices and $2r+1 = \frac{(6r+2)+3-2}{3} = \frac{n+3-2}{3}$.

Case 4: When $n = 6r+5, r \geq 1$.

We construct a 2-graph $M(6r+5, 2r+1)$ from the graph $M(6r+4, 2r+1)$, by adding a new vertex v_{6r+5}, and adding $2r+1$ singletons of $\{v_{6r+5}\}$ (see Figure 7.3(d)). Then $M(6r+5, 2r+1)$ has $n = 6r+5$ vertices and $2r+1 = \frac{(6r+5)+3-5}{3} = \frac{n+3-5}{3}$.

Case 5: When $n = 6r+3, r \geq 1$.

We construct a 2-graph $M(6r+3, 2r+1)$ by removing all the edges of $\{v_{6r+2}\}$ from the graph $M(6r+2, 2r+1)$, adding a new vertex v_{6r+3}, an edge $\{v_{6r+3}\}$ and $2r$ copies of $v_{6r+2}v_{6r+3}$ (see Figure 7.3(e)). Then $M(6r+3, 2r+1)$ has $n = 6r+3$ vertices and $2r+1 = \frac{(6r+3)+3-3}{3} = \frac{n+3-3}{3}$.

Case 6: When $n = 6r+6, r \geq 1$.

We construct a 2-graph $M(6r+6, 2r+1)$ from the graph $M(6r+4, 2r+1)$, by adding new vertices v_{6r+5} and v_{6r+r}, adding $2r+1$ copies of $v_{6r+5}v_{6r+6}$ (see Figure 7.3(f)). Then $M(6r+6, 2r+1)$ has $n = 6r+6$ vertices and $2r+1 = \frac{(6r+6)+3-6}{3} = \frac{n+3-6}{3}$.

Figure 7.3 shows the graphs constructed above. It is easy to see that in the 2-graphs constructed in (a) through (f), the component induced by the vertex v_0 is 2-factor-free. Also, there are $(2r+1)$ cut edges incident to v_0. Therefore, all these 2-graphs are non-decomposable by Theorem 7.1.1. This proves that $f_2(n) \geq \frac{n+3-i}{3}$ where $1 \leq i \leq 6$ and $i \equiv n \pmod 6$.

Finally, we show that $f_2(n) \leq \frac{n+3-i}{3}$, $1 \leq i \leq 6$ and $i \equiv n \pmod 6$. First of all, by the above construction, we know that $M(7,3)$ is a non-decomposable 3-regular 2-graph, so that $f_2(7) \geq 3$. Since a non-decomposable

$f_2(n)$-regular 2-graph on n vertices plus an extra vertex contained in $f_2(n)$ singletons gives a non-decomposable $f_2(n)$-regular 2-graph on $n+1$ vertices, we have $3 \leq f_2(n) \leq f_2(n+1)$ when $n \geq 7$. Moreover, for $n \geq 7$, $f_2(n)$ must be odd by Theorem 7.4.8.

Let G be a non-decomposable $f_2(n)$-regular 2-graph on n vertices. By Lemma 7.4.7, we may assume that G is primitive. Therefore, G has at most one vertex incident with singletons. We construct a non-decomposable $f_2(n)$-regular graph G' as follows. If G does not have a singleton, then let $G' = G$. Otherwise let x be the vertex with singleton $\{x\}$. Let p be the multiplicity of $\{x\}$ in $E(G)$. Then $1 \leq p \leq f_2(n)$, we construct G' by the following cases.

Case 1: $p = f_2(n)$. Then x is an isolated vertex. Let $G' = G - x$.

Case 2: $p = 2m$. We remove p copies of $\{x\}$, add in new vertices y, z, m copies of xy, m copies of xz, and $f_2(n) - m$ copies of yz. Let G' be the resulted graph.

Case 3: Otherwise, i.e., $p = 2m + 1 < f_2(n)$. We remove p copies of $\{x\}$, add new vertices y, z, w, and $2m + 1$ copies of the edge xy, $\frac{f_2(n)-2m-1}{2}$ copies of yz and yw, and $2m+1$ copies of zw. Let G' be the resulted graph.

It can be checked that in each of the above cases the resulting graph G' is $f_2(n)$-regular. Since G is non-decomposable, by the arguments in the proof of Theorem 7.4.8, we know that G' is non-decomposable. By the construction, G' has at most $n + 3$ vertices.

By Theorem 7.1.1, G' has a 2-factor-free component C which is incident with at least $f_2(n)$ cut edges. Each of these cut edges joins C with a component of G' with at least 3 vertices. Then we have $3f_2(n) + |V(C)| \leq$

$|V(G')|$ and hence

$$f_2(n) \leq \frac{|V(G')| - |V(C)|}{3} \leq \frac{n+3-1}{3} = \frac{n+2}{3}. \tag{7.21}$$

Let $n = 6r + i$, where $r \geq 1$ and $1 \leq i \leq 6$ and $i \equiv n \pmod 6$. By (7.21) we have $f_2(n) \leq \frac{n+2}{3} = \frac{6r+i+2}{3} = 2r + 1 + \frac{i-1}{3}$. Since $\lfloor \frac{i-1}{3} \rfloor \leq 1$ and $f_2(n)$ is odd, it follows that $f_2(n) \leq 2r + 1 = \frac{n+3-i}{3}$. □

7.5 Conclusions and Future Work

In this chapter, we applied the decomposition design scheme to the design of universal (k, w)-USB's. We proved that the prime k-USB's are (k, r)-USB's with $r = 1, 2, 3, \ldots, 2j - 1, \ldots, f_2(k)$, where $f_2(k)$ is the maximum degree among all non-decomposable regular 2-graphs on k-vertices. Using graph theory, we proved that $f_2(k) = 2$ when $3 \leq k \leq 6$ and $\frac{k+3-i}{3}$ when $k \geq 7$ where $1 \leq i \leq 6$ and $i \equiv n \pmod 6$. We gave optimal $(k, 1)$-USB's and $(k, 2)$-USB's for all $k \geq 2$, and showed that m copies of an optimal $(k, 2)$-USB constitute an optimal $(k, 2m)$-USB.

For $k \geq 7$, designing optimal prime (k, r)-USB's with $r = 3, 5, \ldots, 2j - 1, \ldots, f_2(k)$ is a challenging problem and is still open. However, we proposed an iterative design method which starts from an optimal $(k, 1)$-USB and adds an optimal $(k, 2)$-USB together with some bridge edges to the latest constructed k-USB successively until all prime k-USB's are constructed. As pointed out in Chapter 5, we do not know if $G(k, 2)$ is hyper-universal for $k \geq 5$. If it is, the above inductive design scheme can applied to design prime k-HUSB's.

We have studied only the regular universal switch blocks. More work remains to be done on the design of irregular universal switch blocks.

The exact value of $f_2(k)$ gives an exact bound in searching for 2-pin net k-MPTRR's. The characterization theorem of k-MPTTR's can also be used in searching for k-MPTRR's. Enumerating all 2-pin net k-MPTRR's for $k \geq 9$ is necessary in designing k-USB's. Further work needs to be done on finding an efficient k-MPTRR enumeration algorithm.

Historical note:

Universal switch blocks have been studied previously by other authors [14, 46]. In [14], Chang et al. proposed the concept of universal switch blocks and presented an optimal $(4, w)$-USB. In [46], Shyu et al. generalized the switch block construction algorithm for any given k and w. They showed that the (k, w)-SB's, $M_{k,w}$, generated by their algorithm are optimal (k, w)-USB's for all k and w. Y. L. Wu brought [46] to my attention. He suggested that I check [46] to see if it might be helpful in designing hyper-universal switch blocks. I examined the universal switch blocks given [46] using our technique for hyper-universal switch blocks, and I found that their result was not correct. I gave a 2-pin net $(7, 3)$-RR which is not routable in $M_{7,3}$. This means that $M_{7,3}$ is not universal. I reported this discovery to J. Liu and Y. L. Wu. Wu invited me and J. Liu to the Chinese University of Hong Kong to continue the research on the switch block design problem. Using the decomposition design technique for universal switch block design, the main issue is the decomposition of regular 2-graphs. Zheng Sun, a graduate student at CUHK, mentioned to me Petersen's even factor theorem and the k-factor theorem. I showed that the same result holds for regular 2-graphs of even degrees. That is, a 2-pin net $(k, 2m)$-TRR can be decomposed into m $(k, 2)$-TRR's. Then I showed that our switch block candidate

design $G(k,2)$'s are universal for all $k \geq 2$ and that $G(k,2)$ is isomorphic to $M_{k,2}$. Therefore $mG(k,2)$ and $M_{k,2m}$ are isomorphic and they are both universal. A correction was published in [27], in which we use the claim for $M_{k,2}$ rather than using $G(k,2)$. I gave all 2-pin k-MPTRR's for $k = 1,\ldots,8$ and proved that $\frac{k+3-i}{3} \leq f_2(k) \leq \frac{k(k-1)}{2}$ [26]. J. Liu verified all the proofs and conjectured that $f_2(k) = \frac{k+3-i}{3}$. J. Liu suggested working on his conjecture on $f_2(k)$. We worked together through email, and we eventually proved the conjecture. The result of Theorem 7.4.6 was later improved and generalized to a stronger graph theorem by J. Liu and Guizhen Liu. The proofs presented in this dissertation were the original proofs to which I had contributed. The application of this result to (k,w)-USB's together with experimental justification (done by Y. L. Wu and C. C. Cheung) were published in [25].

Bibliography

1. A. A. Aggarwal and D. M. Lewis. Routing Architectures for Hierarchical Field Programmable Gate Arrays. In *Proceedings of the International Conference on Computer Design : VLSI in Computers and Processors*, pages 475–478, 1994.

2. J. Akiyama and M. Kano. Factors and factorization of graphs–a survey. *J. Graph Theory*, 9(1):67–86, 1985.

3. M. J. Alexander and G. Robins. New Performance FPGA Routing Algorithms. In *Proceedings of IEEE/ACM Design Automation Conference*, pages 562–567, 1995.

4. V. E. Benes. Optimal rearrangeable multistage connecting networks. *Bell System Technical Journal*, 43:1641–1656, 1964.

5. C. Berge. *Graphs and Hypergraphs.* Elsevier Science Science Publication Company Inc., New York, 1985.

6. V. Betz and J. Rose. Directional Bias and Non-Uniformity in FPGA Global Routing Architectures. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 652–659, 1996.

7. V. Betz and J. Rose. A New Packing, Placement and Routing Tool for FPGA Research. *Seventh International Workshop on Field-Programmable Logic and Applications (Available for download*

from http://www.eecg.toronto.edu/~jayar/software.html), pages 213–222, 1997.

8. V. Betz and J. Rose. Circuit design, transistor sizing and wire layout of FPGA interconnect. In *Custom Integrated Circuits*, pages 171 –174. IEEE, 1999.

9. V. Betz, J. Rose, and A. Morquardt. *Architecture and CAD for Deep-Submicron FPGAs.* Kluwer Academic Publishers, Boston MA, 1999.

10. B. Bollobas. *Extremal Graph Theory.* Academic Press, New York, 1978.

11. J. A. Bondy and U. S. R. Murty. *Graph Theory with Applications.* Macmillan Press, London, 1976.

12. S. Brown, R. Francis, J. Rose, and Z. Vranesic. *Field Programmable Gate Arrays.* Kluwer Academic Publishers, Boston MA, 1992.

13. S. Brown, J. Rose, and Z. G. Vranesic. A detailed router for field-programmable gate arrays. *IEEE Transactions on Computer-Aided Design*, 11:620–628, May 1992.

14. Y. W. Chang, D. F. Wong, and C. K. Wong. Universal switch modules for FPGA design. *ACM Transactions on Design Automation of Electronic Systems.*, 1(1):80–101, 1996.

15. Y. W. Chang, K. Zhu, and D. F. Wong. Timing-driven routing for symmetrical array-based FPGAs. *ACM Transactions on Design Automation of Electronic Systems.*, 5(3):433–450, January 2000.

16. C. Clos. A study of non-blocking switching networks. *Bell System Technical Journal*, 32:406–424, Mar. 1953.

17. J. Cong and L. He. Optimal wiresizing for interconnects with multiple sources. *ACM Transactions on Design Automation of Electronic Systems*, 1(4):478–511, Oct. 1996.

18. E. Contejean and H. Devie. An efficient incremental algorithm for solving systems of linear diophantine equations. *Inform. and Comput.*, 113(1):143–172, 1994.

19. A. DeHon. *Reconfigurable Architectures for General-Purpose Computing.* PhD thesis, Massachusetts Institute of Technology, September 1996.

20. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Monographs in Computer Science. Springer-Verlag, 1999.

21. H. Fan, P. Haxell, and J. Liu. The global routing—a combinatorial design problem. *submitted to Discrete Applied Mathematics*, 2000.

22. H. Fan, J. Liu, and Y. L. Wu. General Models for Optimum Arbitrary-Dimension FPGA Switch Box Designs. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 93–98, 2000.

23. H. Fan, J. Liu, and Y. L. Wu. General Models and a Reduction Design Technique for FPGA Switch Box Designs. *IEEE Transactions on Computers*, 52(1):21–30, Jan. 2003.

24. H. Fan, J. Liu, Y. L. Wu, and C. C. Cheung. On Optimum Switch Box Designs for 2-D FPGAs. *Proceedings of IEEE/ACM Design Automation Conference*, pages 203–208, Jun. 2001.

25. H. Fan, J. Liu, Y. L. Wu, and C. C. Cheung. On Optimum Designs of Universal Switch Blocks. *Lecture Notes in Computer Science*, 2438:142–151, 2002.

26. H. Fan, J. Liu, Y. L. Wu, and C. K. Wong. Reduction Design for Generic Universal Switch Blocks. *ACM Transactions on Design Automation of Electronic Systems*, (4):526–546, Oct. 2002.

27. H. Fan, Y. L. Wu, and Y. W. Chang. Comment on "General Universal Switch Blocks". *IEEE Transactions on Computers*, 51(1):93–95, Jan. 2002.

28. P. Hallschmid and S. Wilton. Detailed Routing Architectures for Embedded Programmable Logic IP Cores. In *ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 69–74, Monterey, CA, Feb. 2001.

29. G. Higman. Ordering by divisibility in abstract algebras. *Proc. London Math. Soc*, 2:326–336, 1952.

30. K. Ireland and M. Rosen. *A Classical Introduction to Modern Mumber Theory.* Springer-Verlag, New York, 1980.

31. G. Lemieux. *Efficient Interconnection Network Components for Programmable Logic Devices.* PhD thesis, University of Toronto, 2003.

32. G. Lemieux and D. Lewis. Using Sparse Crossbars within LUT Clusters. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, pages 59–68, 2001.

33. G. Lemieux and D. Lewis. Analytical Framework for Switch Block Design. *Lecture Notes in Computer Science*, 2438:122–131, 2002.

34. G. Lemieux and D. Lewis. Circuit Design of FPGA Routing Switches. In *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, pages 19–28, 2002.

35. L. Lovasz and M. D. Plummer. *Matching Theory.* Elsevier Science Publishers, New York, 1986.

36. S. Nakamura and G. M. Masson. Lower bounds on crosspoints in concentrators. *IEEE Transactions on Computers*, 31:1173–1179, 1982.

37. J. F. Pan, Y. L. Wu, G. Yan, and C. K. Wong. On the optimal four-way switch box routing structures of FPGA greedy routing architectures. *Integration, the VLSI Journal*, 25:137–159, 1998.

38. D. V. Pasechnik. On computing Hilbert bases via the Elliot-MacMahon algorithm. *Theoretical Computer Science*, 263:37–46, 2001.

39. M. Pedram, B. S. Nobandegani, and B. T. Preas. Architecture and Routability Analysis for Row-Based FPGAs. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 230–235. IEEE Computer Society Press, 1993.

40. J. Petersen. Die theorie der regulären graphs. *Acta Math*, 15:193–220, 1891.

41. L. Pottier. Minimal solutions of linear diophantine systems: bounds and algorithms,. In *Proceedings of the 4th International Conference on Rewriting Techniques and Applications*, pages 162–173, 1991.

42. J. Rose and S. Brown. Flexibility of Interconnection Structures for Field-Programmable Gate Arrays. *IEEE Journal of Solid State Circuits*, 26(3):277–282, March 1991.

43. Jonathan Rose, Abbas El Gamal, and Alberto Sangiovanni-Vincentelli. Architecture of Field-Programmable Gate Arrays. *Proceedings of the IEEE*, 81(7):1013–1029, July 1993.

44. Alberto Sangiovanni-Vincentelli, Abbas El Gamal, and Jonathan Rose. Synthesis Methods for Field Programmable Gate Arrays. *Proceedings of the IEEE*, 81(7):1057–1083, July 1993.

45. H. Schmit and V. Chandra. FPGA Switch Block Layout and Evaluation. In *Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 11–18, 2002.

46. M. Shyu, G. M. Wu, Y. D. Chang, and Y. W. Chang. Generic Universal Switch Blocks. *IEEE Transactions on Computers*, pages 348–359, 2000.

47. Andrew S. Tanenbaum. *Computer Networks.* Prentice-Hall, Inc., 1996.

48. E. Tau, D. Chen, I. Eslick, J. Brown, and A. DeHon. A First Generation DPGA Implementation. In *FPD'94 - Third Canadian Workshop of Field-Programmable Devices*, pages 138–143, May 1995.

49. S. Trimberger. *Field-Programmable Gate Array Technology.* Kluwer Academic Publishers, Boston MA, 1994.

50. W. T. Tutte. The factorization of linear graphs. *J. London Math. Soc.*, 22:107–111, 1947.

51. S. Wilton. *Architecture and Algorithms for Field-Programmable Gate Arrays with Embedded Memory.* PhD thesis, University of Toronto, 1997.

52. S. Wilton and R. Saleh. Progammable Logic IP Cores in SoC Design: Opportunities and Challenges. In *Proceedings of IEEE Custom Integrated Circuits Conference*, pages 63–66, 2001.

53. G. M. Wu and Y. W. Chang. Switch-Matrix Architecture and Routing for FPDs. In *Proceedings of the International Symposium on Physical Design*, pages 158–163, New York, 1998. ACM Press.

54. G. M. Wu and Y. W. Chang. Quasi-universal switch matrices for FPD design. *IEEE Transactions on Computers*, 48:1107–1122, Oct. 1999.

55. G. M. Wu, J. M. Lin, and Y. W. Chang. An algorithm for dynamically reconfigurable FPGA placement. In *Proceedings of IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pages 501–504. IEEE, 2001.

56. Y. L. Wu and D. Chang. On NP-completeness of 2-D FPGA routing architectures and a novel solution. *Proceedings of International Conference on Computer-Aided-Design*, pages 362–366, 1994.

57. Y. L. Wu and M. Marek-Sadowska. Routing for array type FPGAs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(5):506–518, May 1997.

58. Y. L. Wu, S. Tsukiyama, and M. Marek-Sadowska. On computational complexity of a detailed routing problem in two-dimensional FPGAs. In *Proceedings 4th Great Lakes Symp. VLSI*, pages 70–75, 1994.

59. Y. L. Wu, S. Tsukiyama, and M. Marek-Sadowska. Graph based analysis of 2-D FPGA routing. *IEEE Transactions on Computer-Aided Design*, 15(1):33–44, 1996.

60. Y.L. Wu and M. Marek-Sadowska. Graph Based Analysis of FPGA Routing. In *Proceedings of the European Design and Automation Conference with EURO-VHDL'93*, pages 104–109, September 1993.

61. Xilinx. Inc. the programmable logic data book. 1994.

62. In. Xilinx. Virtex-ii pro hand book. 2002.

63. Y. Y. Yang. A New Conference Network for Group Ommunicaiton. *IEEE Transactions on Computers*, 51(9):995–1009, 2002.

64. M.H. Yen, S.J. Chen, and S.H. Lan. A Three-Stage One-Sided Rearrangeable Polygonal Switching Network. *IEEE Transactions on Computers*, 50(11):1291–1294, Nov. 2001.

65. K. Zhu and D. F. Wong. On Channel Segmentation Design for Row-Based FPGAs. In *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, pages 26–29, 1992.

66. K. Zhu, D. F. Wong, and Y.-W. Chang. Switch Module Design with Application to Two-Dimensional Segmentation Design. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 480–485, 1993.

Index