國立交通大學

電子工程學系 電子研究所 碩士班

碩士論文

低功率單晶片網路之拓樸與佈局規劃

Topology Generation and Floorplanning for Low Power Application-Specific Network-on-Chips

研 究 生： 李婉毓

指導教授： 江蕙如 博士

中華民國九十六年七月

低功率單晶片網路之拓樸與佈局規劃
Topology Generation and Floorplanning for Low Power Application-Specific Network-on-Chips

研究生：李婉毓　　Student：Wan-Yu Lee

指導教授：江蕙如　　Advisor：Iris Hui-Ru Jiang

國立交通大學
電子工程學系電子研究所
碩士論文

A Thesis
Submitted to Institute of Electronins
Department of Electronics Engineering
College of Electrical and Computer Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Electronics Engineering

July 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年七月

低功率單晶片網路之拓樸與佈局規劃

研究生：李婉毓　　　　　　　　指導教授：江蕙如 博士

國立交通大學

電子工程學系 電子研究所

摘要

隨著製程之進步，晶片上的核心數目與核心間的資料傳輸量急遽增加。傳統使用共享匯流排做為核心間的連接方式功效不彰。應用網路連接核心的單晶片網路因能大幅提升傳輸效率，是近年新興又熱門的研究領域。單晶片網路的效能可由功率、速度、面積這三方面來評估。功率及速度由網路拓樸及其所使用的路由器數目決定；面積則是與佈局有關。不同於以往，本論文提出新的單晶片網路設計流程—先完成與效能密切相關的拓樸設計而後再做佈局規劃，並且突破前人使用複雜而耗時的演算法的缺點。實驗結果證實，本論文中所產生的網路拓樸保證符合路由器數目的限制，並且保證決不會造成資料傳輸的交互等待因而引發系統停滯。更甚者，在使用與前人一樣甚至更少的路由器數目，並保有前兩項特點之下，仍能達成低功耗的目地。

TOPOLOGY GENERATION AND FLOORPLANNING FOR LOW POWER APPLICATION-SPECIFIC NETWORK-ON-CHIPS

Student: Wan-Yu Lee Advisor: Dr. Iris Hui-Ru Jiang

Institute of Electronics
Department of Electronics Engineering
National Chiao Tung University

Abstract

As the process advances into nanotechnology, the number of cores and the amount of communication on a chip are rapidly increasing. Using a micro-network, Network-on-Chip can overcome the communication inefficiency in the traditional shared bus communication architecture. The system performance of application-specific Network-on-Chips is mostly measured by power, timing, and area. Moreover, power and timing highly depend on how the network topology connects routers and cores and how many routers are used; area is simply determined by floorplanning. Unlike previous endeavors, in this thesis, we propose a new methodology to perform network topology generation before floorplanning. Moreover, our method can preserve the optimality of topology to floorplan. Our method not only simultaneously minimizes power, satisfies timing and area constraints, but also guarantees deadlock free. The results show using the same or less number of routers, this approach can achieve competitive power consumption and have the above guarantees.

Acknowledgements

I would like to express my gratitude to all those who gave me the possibility to complete the thesis. First of all, my sincerest appreciation goes to my advisor, Prof. Iris Hui-Ru Jiang, for her insightful suggestion and patient guidance in completing this thesis. Secondly, I would like to give my gratitude to the members of my thesis committee, Prof. Jing-Yang Jou, Prof. Hung-Ming Chen and Prof. Mango Chia-Tso Chao, for their precious suggestions and comments. Special thanks goes to Prof. Chen-Yi Lee and Dr. Tzu-Ming Liu for providing the H.264 benchmark.

Last but not least, I deeply show my greatest appreciation to my parents, family and friends for their invaluable support, advices and encouragement throughout my study years.

Wan-Yu Lee

National Chiao Tung University

July 2007

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

As technology advances into the nanometer era, the number of cores in a single chip and the communication complexity are rapidly increasing. Although the traditional shared bus communication architecture is simple and easy to implement, it allows only one core to transfer data at a time. This limitation may result in inefficient communication, especially when on-chip communication is extraordinarily dense. To tackle this inefficiency, Network-on-Chip (NoC), as shown in Figure 1.1, has been proposed, connecting cores on a chip by a micro-network [7]. In a micro-network, each core is connected to a router, and a router is connected to another, both through physical links. Each communication message is partitioned into packets and then transferred through a sequence of routers and physical links from the source core to its destination one (packet-switched). In addition, NoC adopts the globally asynchronous locally synchronous (GALS) manner, where the packets are transferred asynchronously between routers but synchronously within a router. Therefore, the NoC communication architecture can provide high bandwidth by pipelining the message [4].

Two of the main tasks affecting the system performance of NoC are network topology generation and floorplanning. The system performance is measured by

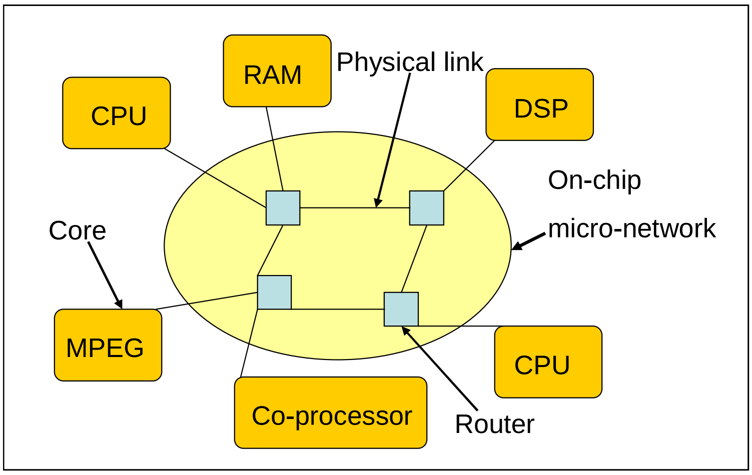


Figure 1.1: An example of Network-on-Chip [5]. Three main components in NoC are cores, routers, and physical links.

power, timing, and area. First of all, as can be seen in Figure 1.2, the power consumption contributed by elements in routers highlighted by a circle and by physical links is 75%, 25%, respectively, in 180 nm technology [12]. (For 100 nm technology, the ratios are 70%, 30%, respectively.) Secondly, the delay of one router consumes hundreds of clock cycles, while that of a physical link is subject to only one clock cycle. Hence, routers dominate power and delay. Finally, compared to the area of cores, the area of routers is negligibly small.

On the other hand, floorplan determines the physical locations of cores and routers, thus influencing the overall area and the length of physical links; the network topology indicates the overall connection between cores and routers, and between routers. The network topology of NoC can be classified into regular and irregular architectures. As demonstrated in Figure 1.3, the connection in the regular architecture is isomorphic, while that in the irregular one is not. The regular architecture, e.g., mesh and torus, has advantages of topology reuse and low design complexity, and is suitable for homogenous cores, e.g., general purpose CPUs, FPGAs, etc.

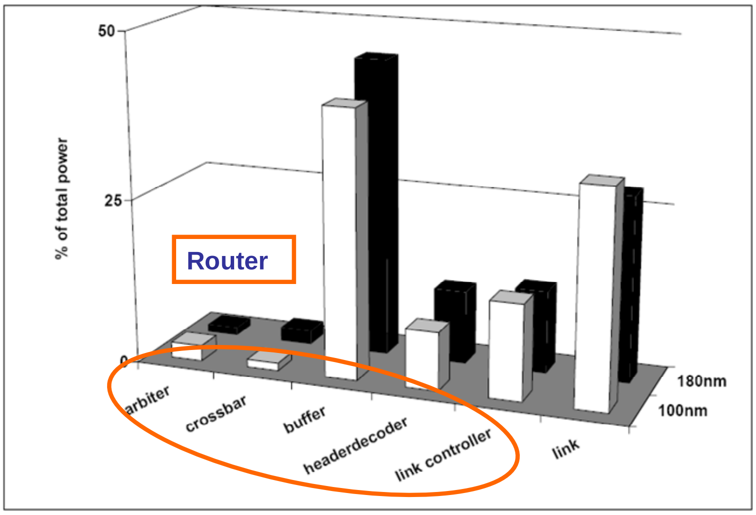


Figure 1.2: The elements in routers highlighted by a circle contribute over 70% power consumption [12].

However, the cores are heterogeneous, i.e., different in functions and sizes, in most designs. The irregular architecture, as known as application-specific or custom architecture, gives a tailored network topology for every design, often uses fewer routers, and also can offer better system performance than the regular one [3]. Thus, in this thesis, we focus on network topology generation and floorplanning for the irregular architecture.

Table 1.1 summarizes the impacts of network topology and floorplan of NoC on power, timing, and area of different components. The network topology determines the power, timing, and area resulted from routers. The floorplan influences the power, timing, and area from physical links, and the overall dimension over cores, routers, and physical links. As mentioned earlier, routers contribute much more power and delay, while cores mainly dominate area. Moreover, for low power designs, power and timing are of significant importance, while area is not tightly constrained. Hence, network topology is more critical than floorplan; in this thesis, we generate the network topology before floorplanning.

Table 1.1: The Relationship between Design Metrics and Network Topology and Floorplan

Design metrics	Power	Timing	Area
Network topology	Router	Router	Router
Floorplan	Physical link	Physical link	Core, Physical link

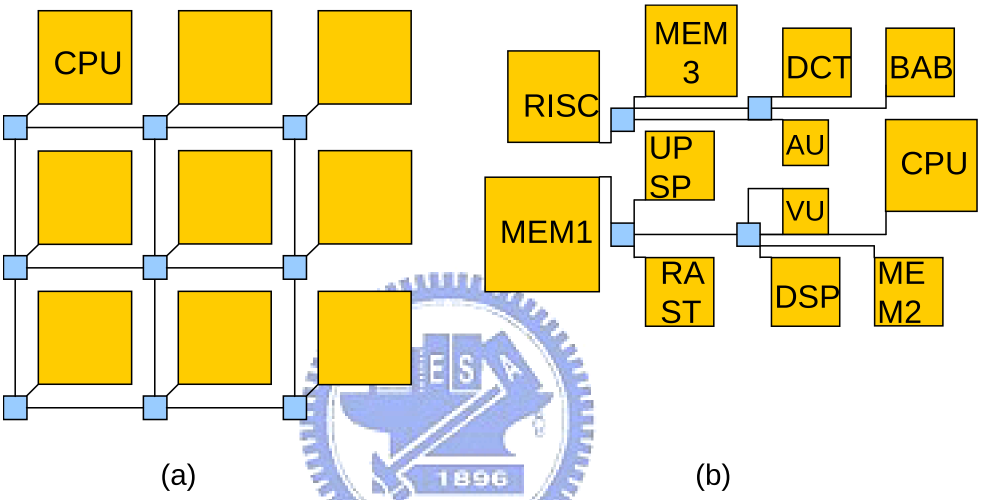


Figure 1.3: The network topology can be classified into (a) regular architecture, and (b) irregular architecture.

1.2 Previous Works

Recently, low power application-specific NoC has extensively been studied in literature [6], [11]—[14]. Srinivasan et al. proposed a two phase work—floorplanning first and then generating topology, using mixed integer linear programming (MILP) in both phases [11, 13]. In order to reduce the time complexity, they also proposed a fast heuristic for the second phase in [12]. However, these works handled floorplan first, and sacrificed some freedom in topology generation, which dominates power and timing. In addition, they cannot guarantee each communication trace is completed within the required number of routers. On the other hand, Murali et al. also presented a two-phase flow in [14]. They adopted simulated annealing during floorplanning, and clustering during topology generation. In their work, they assumed a variety of routers can be used. This assumption might be somewhat impractical. Moreover, either mixed integer linear programming or simulated annealing is very time consuming.

Although NoC provides high bandwidth communication, a bad network topology may induce deadlocks. A deadlock, caused by a cyclic data dependency between resources, may block messages to transfer toward their destinations, thus the system cannot proceed. Figure 1.4 shows an example. Edges in Figure 1.4(a) represent communication traces between cores, without cyclic data dependency between them. The resulting topology as shown in Figure 1.4(b) potentially incurs a deadlock (shown in dotted lines) caused by the communication traces (A, B), (B, E) and (E, F). To prevent potential deadlocks, the authors in [14] restricted the usage of physical links, such that no cycles exist in the network topology. However, they broke the potential cycles without considering the communication between cores, thus possibly losing optimality. [6] and [13] allocated more routers and physical links to provide alternatives, and then deadlocks can be removed. The alternatives were created during post processing, thus they cannot do without the penalties on power and timing.

1.3 Our Contribution

As mentioned in Section 1.1, network topology is more critical than floorplan. In this thesis, we thus propose a new two phase flow—topology generation and then floorplanning. In the first phase, network topology generation focuses on the power and timing issues on routers. This phase targets to minimize the number of routers used, to complete each communication trace within the required number

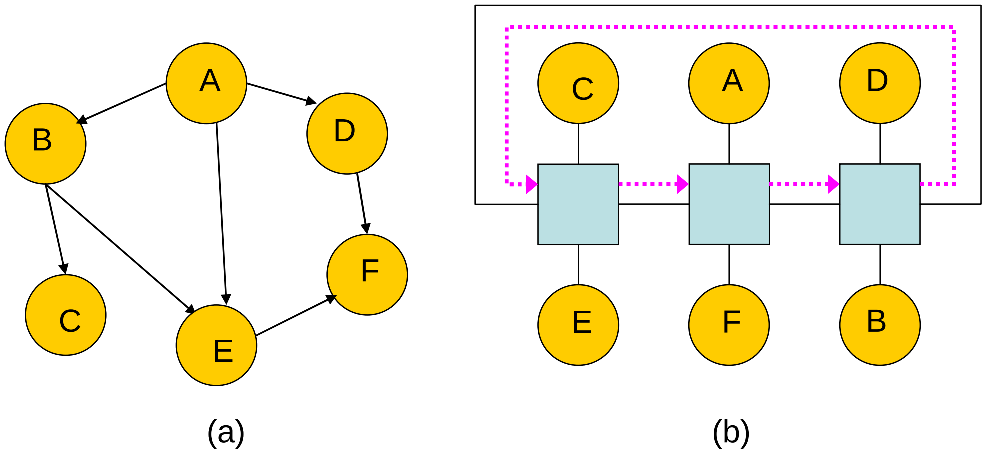


Figure 1.4: A bad topology may induce potential deadlocks. (a) Directed edges in the graph represent communication traces between cores A, B, C, D, E and F, without cyclic data dependency between them. (b) The resulting topology incurs a potential deadlock caused by the communication traces $(A,\ B)$, $(B,\ E)$ and $(E,\ F)$.

of routers, and to guarantee deadlock free. Because the most important issues are tackled during topology generation, in the second phase, floorplanning arranges the locations of cores, routers, and physical links by just flattening the topology. The goal of floorplanning is to minimize the power and timing of physical links and overall area. Our method can preserve the optimality of topology to floorplan. Furthermore, compared with previous work, we adopt partitioning-based approaches in both phases, thus improving the efficiency. Experimental results show that using the same or less number of routers, we can not only achieve competitive power consumption but also guarantee deadlock free and meet timing constraints.

1.4 Organization

The remainder of this thesis is organized as follows. Chapter 2 introduces the concept of NoC and power and timing models, as well as formulates our problem, Chapter 3 describes our methodology, Chapter 4 shows our results, and finally, Chapter 5 concludes this thesis.

Chapter 2

Preliminaries and Problem Definition

In this Chapter, we will introduce communication trace graphs, main components in NoC, and power and timing models, and definite our problem.

2.1 Communication Trace Graph

A communication Trace Graph (CTG) [6, 11, 13] is a directed acyclic graph used to describe a design. Figure 2.1 is a CTG with six nodes and seven edges. In a CTG, a node represents a core associated with its height and width. A directed edge is a communication trace from its source to destination associated with a pair (B, L) of its bandwidth B and latency constraint L. The bandwidth B states for the amount of data transferred by a communication trace measured in Mega-bits per second (Mb/s). On the other hand, the latency constraint L represents the maximum number of routers allowed for a single communication trace.

2.2 Main Components in NoC

As shown in Figure 1.1, three main components in NoC include cores, routers, and physical links. Cores can be processing elements or memories. In NoC, a core must connect to only one router and then communicates with some other cores through routers and physical links. Two cores cannot directly be connected. Con-

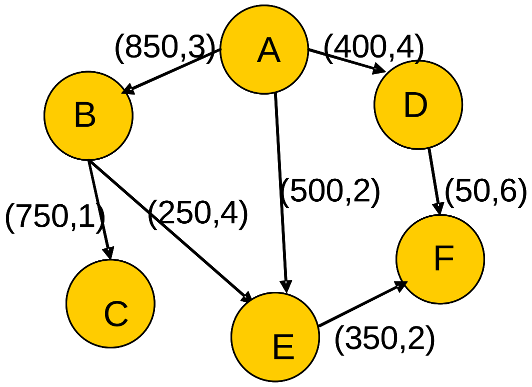


Figure 2.1: An example of CTG with six nodes and seven edges.

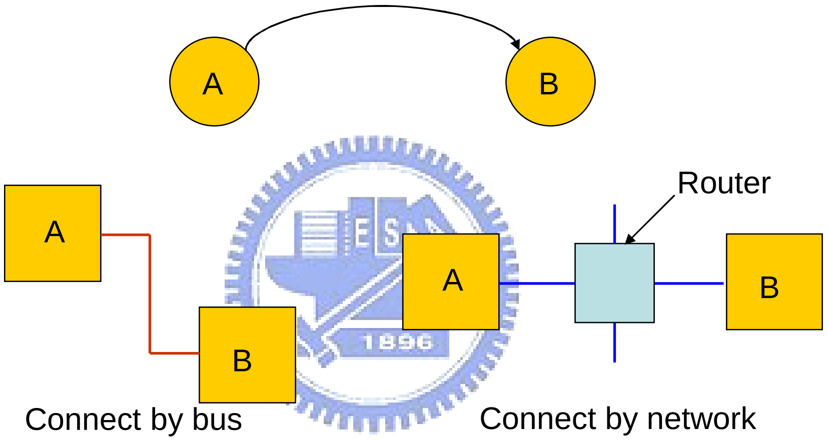


Figure 2.2: An example of connecting A and B by bus and network.

sequently, a communication trace passes at least one router. (When the source and destination cores are connected to the same router). A physical link connects a core to its router or two routers. The power and timing models of routers and physical links will be detailed later. Figure 2.2 shows the difference between the shared bus and micro-network communication architectures. In the shared bus architectures, cores communicate each other through a common bus.

2.3 Router Architecture

The router architecture specifies the number of ports R_p, the peak bandwidth B_{max}, and the power model of a router. The number of ports constrains how many physical links a router can support. The peak bandwidth is the maximum bandwidth allowed for each port. The power model indicates the power consumption of input and output ports of unit bandwidth (nW/Mb/s).

2.4 Physical Link Model

We use the Manhattan distance to measure the physical link length. The power model of a physical link is proportional to its length and the bandwidth transferred through the link. The area of a physical link is proportional to its link width.

2.5 Power Model

Assume n routers, physical links of total length L_p on a communication trace of bandwidth B, while the power model of routers P_i for input ports and P_o for output ports, and unit-length physical link power P_p. The total power consumption P_{total} of this communication trace is computed by the router power P_{router} plus the physical link power P_{link} as follows.

$$\begin{aligned}
P_{total} &= P_{router} + P_{link},\ where \\
P_{router} &= B \cdot (P_i + P_o) \cdot n\ (nW), \\
P_{link} &= B \cdot L_p \cdot P_p\ (nW).
\end{aligned}$$

For example, if the communication trace (A, D) in Figure 2.1 passes through 2 routers and total 1mm-long physical links (see Figure 2.3), while the input and

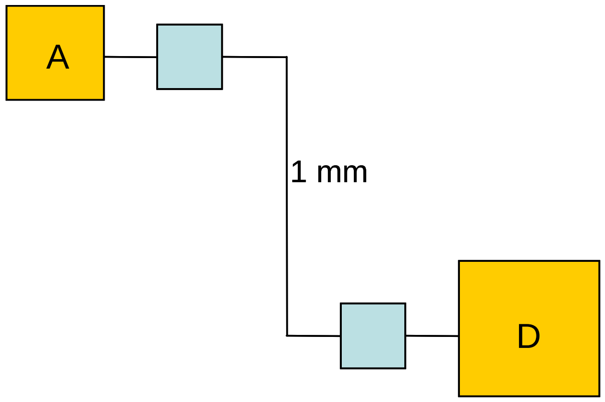


Figure 2.3: The communication trace $(A,\ D)$ in Figure 2.1 pass through 2 routers and total 1mm-long physical links.

output port power of routers are 300 (nW/Mb/s) and 65 (nW/Mb/s), respectively, and unit-length physical link power is 65 (nW/Mb/s/mm), then the power of the communication trace $(A,\ D)$ is:

$$P_{total} = P_{router} + P_{link} = 400{\cdot}(300 + 65){\cdot}2 + 400{\cdot}1{\cdot}65 = 318\ (\mu\text{W}).$$

2.6 Timing Model

Assume n routers, physical links of total length L_p on a communication trace under the clock period T_c, C clocks per router, and the unit-length physical link delay T_p. The total delay T_{total} is computed by the router delay T_{router} plus the physical link delay T_{link} as follows.

$$\begin{aligned} T_{total} &= T_{router} + T_{link},\ where \\ T_{router} &= n \cdot C \cdot T_c\ (ns), \\ T_{link} &= L_p \cdot T_p\ (ns). \end{aligned}$$

However, since T_{router} is usually hundreds times by T_{link}, we simplify the timing model by omitting T_{link} in T_{total}, but constrain the maximum distance L_{max}

allowed for a physical link to ensure its delay is less than one clock cycle. Thus, our timing model are as follows.

Total delay: $T_{total} = T_{router}$.

Maximum distance constraint: L_{max} (mm).

Using the same case in Section 2.5, when one router delay is equal to 100 clocks, and the clock period T_p is 3 ns, the delay of communication $(A,\ D)$ is:

$T_{total} = T_{router} = 2{\cdot}100{\cdot}3{=}600$ (ns), while $L_{max} = 6$ (mm).

2.7 Problem Definition

We formulate the topology and floorplan generation (TFG) problem as follows.

Problem: Topology and Floorplan Generation (TFG): Given a CTG, the router architecture, and the physical link model, find a deadlock-free network topology and floorplan with minimum power, subject to area, timing, and bandwidth constraints.

Figure 2.4 gives the inputs and outputs of the TFG problem. Assume a floorplan of height H and of width W(see Figure 2.5). The area constraints bound the aspect ratio (H/W), the overall area $(H{\cdot}W)$, and the link width of physical links. The timing constraints include the latency constraint on every communication trace in CTG, and the link length constraint of each physical link. The bandwidth constraints describe the router peak bandwidth. If there exists no deadlock-free topology meeting all latency constraints, we shall minimize the number of violations.

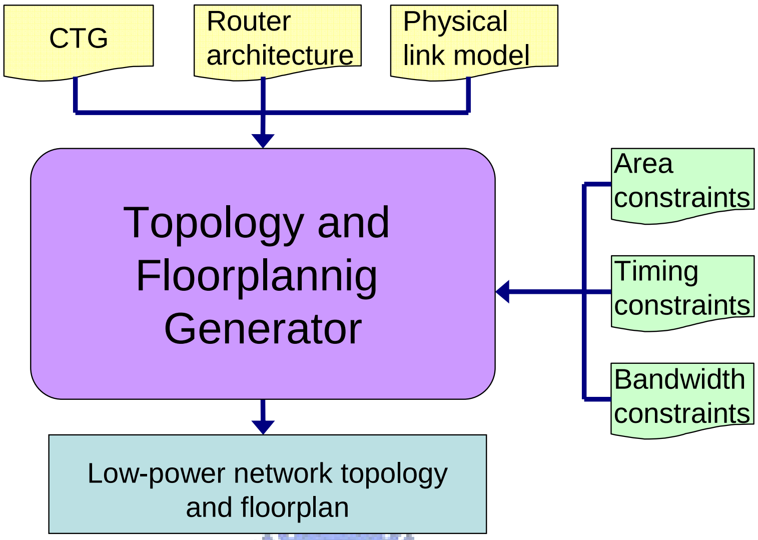


Figure 2.4: The inputs and outputs of the TFG problem.

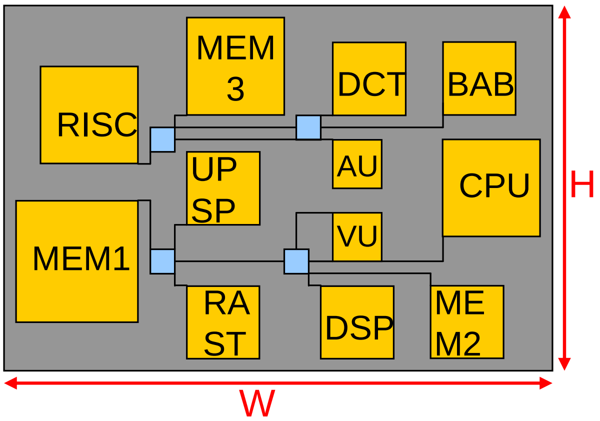


Figure 2.5: A floorplan of height H and of width W.

Chapter 3

New Methodology

As mentioned in Chapter 1, network topology is more critical than floorplan; in this thesis, we thus propose the TFG flow—topology generation and then floorplanning (see Figure 3.1). Our goal is to find a network topology that can reflect the input CTG. When a communication trace between two cores has high bandwidth and a tight latency, the closer these two cores in topology, the easier to reduce power and to meet latency constraints. So does floorplanning. For example, Figure 3.2(a) is the input CTG given by Figure 2.1, Figure 3.2(b) is the resulting network topology, Figure 3.2(c) shows the routing path of each trace, and Figure 3.2(d) shows the floorplan. We will detail topology generation and floorplanning in Chapter 3.1 and Chapter 3.2.

3.1 Phase I—Topology Generation

We formulate the topology generation (TG) problem as follows and propose the TG algorithm to solve it.

3.1.1 Problem Formulation of Topology Generation

Problem: Topology Generation (TG): Given a CTG $G = (V, E)$, the router architecture, find a deadlock-free topology N and assign every communication

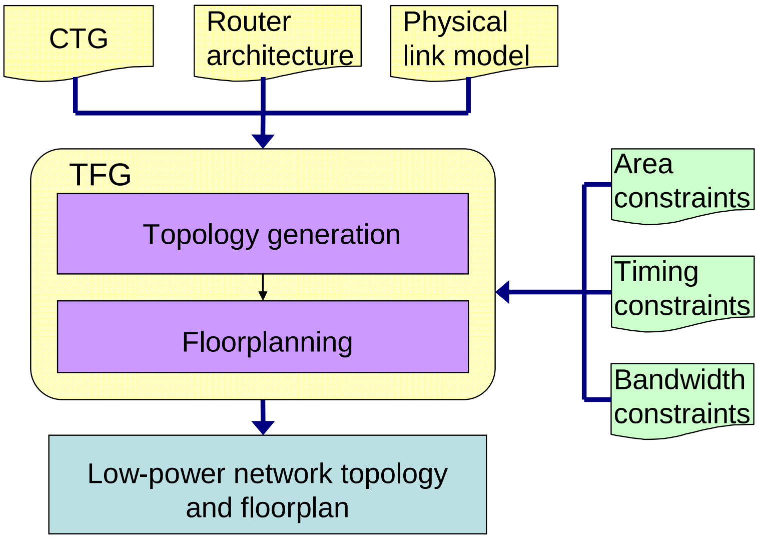


Figure 3.1: The overview of the TFG flow.

trace in G with a routing path in N such that the number of routers and router power consumption are minimized, latency and bandwidth constraints are satisfied.

The topology N is a directed graph with the leaves as cores, the internal nodes as routers, the edges as physical links. Each core is associated with its height and width from the CTG. Each edge in G corresponds to a path in N, a sequence of nodes and edges. Moreover, *deadlock free can be guaranteed if the subgraph of* N *induced by internal nodes is maintained acyclic.*

3.1.2 The TG Algorithm

The TG algorithm is listed in Figure 3.3. Line 1 initializes N by applying topological sort on G. Line 2 uses g to record the number of groups in N. Lines 3–5 incrementally generate topology N until the number of groups is not greater than the number of ports of a router. Line 6 finally connects the groups to a router. Line 7 accordingly assigns every communication trace in G a routing path in N. Line 8 outputs the topology.

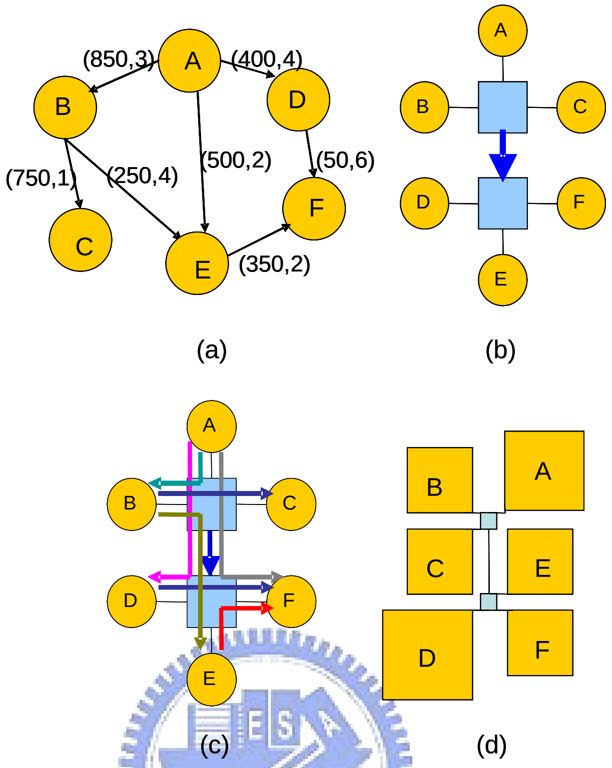


Figure 3.2: One example of TFG. (a) A CTG. (b) The network topology. (c) The network topology with path assignment. (d) The floorplan.

In line 4, for an edge of latency constraint 0 or 1, Check_Tight_Latency merges the source and destination nodes into a group and mounts them to a router. Then, in line 5, Partition_and_Merge applies min-cut partition on N and merges groups into a router if possible. Check_Tight_Latency and Partition_and_Merge maintain the subgraphs of the current N induced by internal nodes acyclic using the topological order obtained from line 1. When nodes are merged, the latency constraints on the related edges are updated as the original number minus one. If the related edges are also merged, the minimum of these updated latency constraints is assigned to the new edge. In addition, the bandwidths of these edges are accumulated to the new edge. Once the bandwidth of an edge exceeds B_{max}, it shall be split into multiple edges, and the bandwidth will be distributed over these new edges.

```
Algorithm: TG(G, R, N)
Input:    G=(V, E)                /* CTG */
          $R=(R_p, B_{max})$      /* router architecture */
Output:   N                       /* the topology of G */
N←Topological_Sort(G)
g←|V| /* # of groups in N */
while $g > R_p$ do
    (N, g)←Check_Tight_Latency(N, g, R)
    (N, g)←Partition_and_Merge(N, g, R)
N←Connect(N)
Assign_Routing_Path(G, N)
return N
```

Figure 3.3: The TG algorithm.

Figure 3.4(a) is the topologically sorted graph of the CTG in Figure 3.2(a), where the initial number of groups is six. After Check_Tight_Latency, B and C are merged, the related edges $(A,\ B)$ and $(B,\ E)$ are updated as the bold edges $(A,\ \{B,\ C\})$ and $(\{B,\ C\},\ E)$ in Figure 3.4(b). In Figure 3.4(c), during Partition_and_Merge, A is merged with B and C, the edges $(A,\ E)$ and $(\{B,\ C\},\ E)$ shown in dotted lines are also merged to the bold edge $(\{A,\ B,\ C\},\ E)$. The latency constraint of this new edge $L(\{A,\ B,\ C\},\ E)$ =min$\{L(A,\ E)-1, L(\{B,\ C\},\ E)\}$, while its bandwidth $B(\{A,\ B,\ C\},\ E) = B(A,\ E) + B(\{B,\ C\},\ E)$. Figure 3.5 illustrates another example.

3.2 Phase II—Floorplanning

We formulate the floorplanning problem as follows and propose the FP algorithm to solve it.

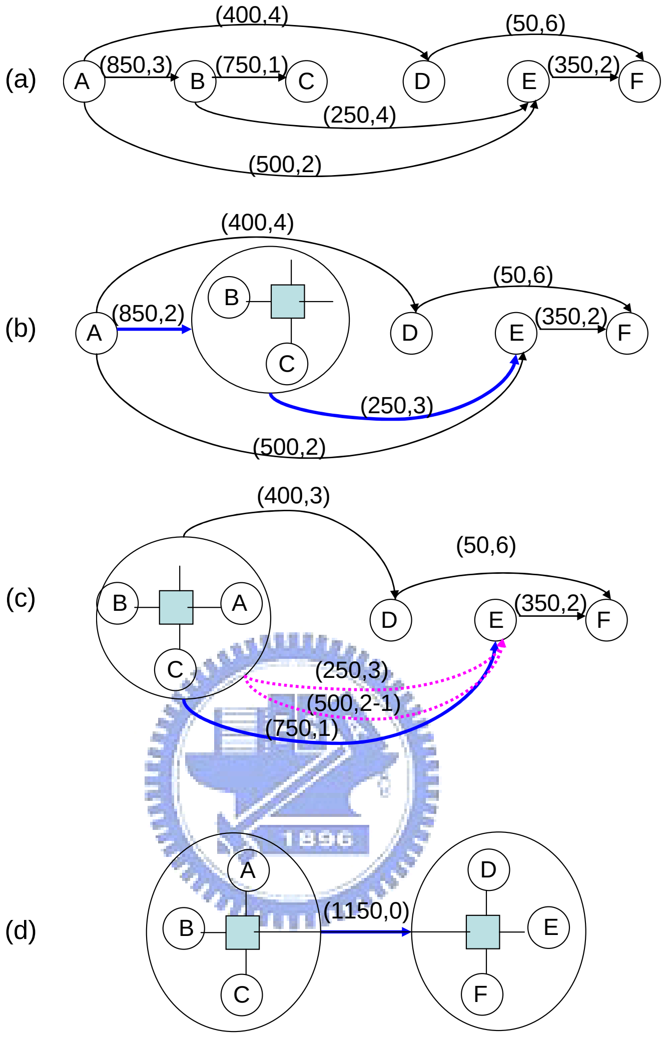


Figure 3.4: One example of the TG algorithm.

3.2.1 Problem Formulation of Floorplanning

Problem: Floorplanning (FP): Given the topology N generated in Phase I, the physical link model, area and timing constraints, find a floorplan F with minimum physical link power, such that area constraints and maximum distance constraint are satisfied.

3.2.2 The FP Algorithm

Inspired by [1], we generate the floorplan by flattening the topology, as shown in Figure 3.6. In line 1, Find_Corner chooses the beginning and the end routers of the longest path in N as two corners of the floorplan. The floorplan dimension is initialized by area constraints. Lines 2 and 3 decide the location of each non-corner router v according to the path length between v and corners in N and the dimensions of cores connected to routers on the path. Finally, line 4 flattens cores and refines the floorplan. L_{max} is used in Find_Location and Flatten. After the FP algorithm, the resulting floorplan of Figure 3.2(a) is shown in Figure 3.2(d).

Considering another example, Figure 3.7(b) is the topology of Figure 3.7(a). First of all, as shown in Figure 3.7(c), Find_Corner chooses routers A and D of path 1 and A and F of path 2 as corners of floorplan. Then, Find_Location decides the location of non-corner routers, B, C and E along the dotted lines. (see Figure 3.7(d)). Finally, Flatten arranges the physical location of each core and refines the floorplan. The resulting floorplan is shown in Figure 3.7(e).

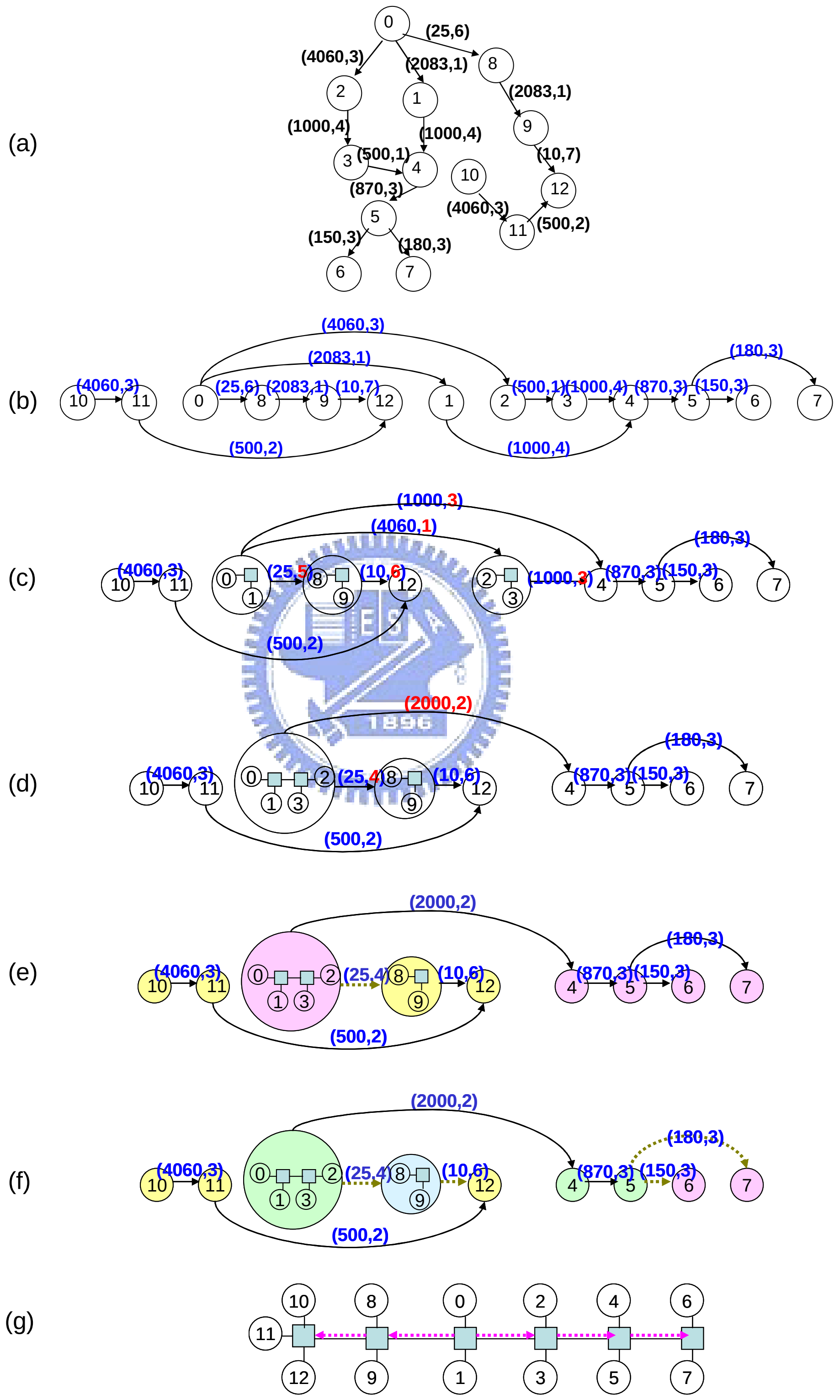
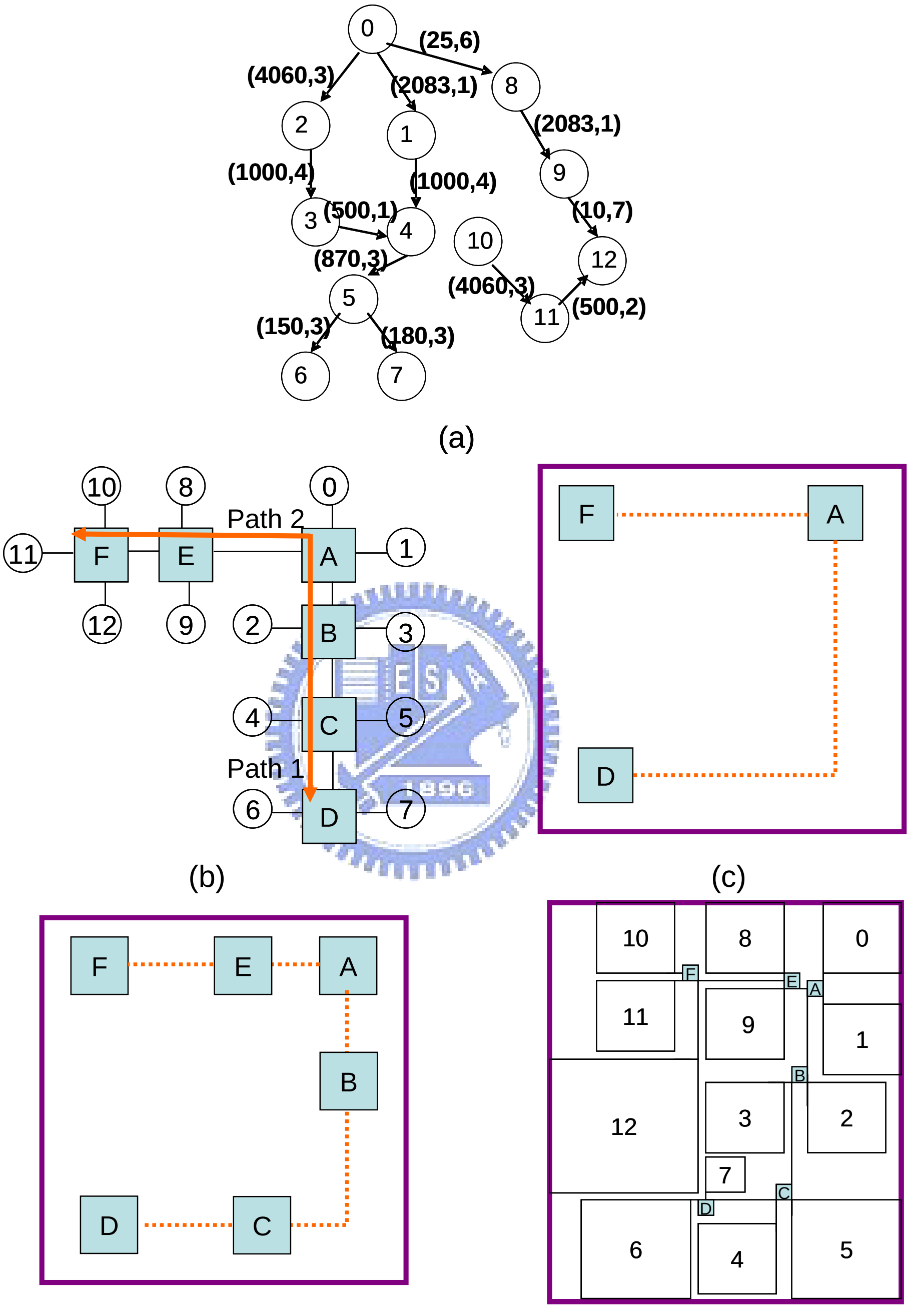


Figure 3.5: Another example of the TG algorithm. (a) The CTG of mp3 enc mp3 dec. (b)–(f) The process of the TG algorithm. (g) The resulting topology.

```
Algorithm: FP(N, $L_{max}$, A, F)
Input:   N          /* topology */
         $L_{max}$  /* max link length constraint*/
         A          /* area constraints */
Output:  F          /* floorplan */
F $\leftarrow$ Find_Corner(N)
foreach non-corner router v in N do
    F $\leftarrow$ F $\cup$ Find_Location(v)
F $\leftarrow$ F $\cup$ Flatten(N)
return F
```

Figure 3.6: The FP algorithm.



(a)

(b)

(c)

(d)

(e)

Figure 3.7: Another example of the FP algorithm. (a) The CTG of mp3 enc mp3 dec with bandwidth in Kb/s. (b) The corresponding topology. (c) Find_Corner chooses corner routers A, D, F. (d) Find_Location decides the location of non-corner routers B, C, E. (e) The resulting floorplan.

Chapter 4

Experimental Results

4.1 Benchmark Applications

We applied our algorithm on three benchmarks in [11] and the H.264/AVC video decoder in [9]. (Only these three benchmarks are illustrated in [11].) Table 4.1 lists the characteristics of benchmark CTGs in the number of nodes and in the number of edges, while Table 4.2 gives the node descriptions of the benchmarks.

4.2 Experimental Setup

We adopted the parameters generated by a cycle accurate power and performance model [2] in our experiments. These parameters are also used in [10, 11, 12]. Under 100nm technology and 3ns clock period, the power model of input port P_i and output port P_o is 328 nW/Mb/s and 65.5 nW/Mb/s, respectively. The unit-length physical link power P_p is 79.6 nW/Mb/s/mm.

Table 4.1: CTG Characteristics

Benchmark	Nodes	Edges
263 dec mp3 dec	14	15
263 enc mp3 dec	12	12
mp3 enc mp3 dec	13	12
H.264 BL@L4.1	8	8

Table 4.2: Node Descriptions

Node	263 dec mp3 dec	263 enc mp3 dec	mp3 enc mp3 dec	H.264 BL@L4.1
0	VLD	ME	FP	SYNTAX PARSER
1	IQ	DCT	FFT	CAVLC
2	IDCT	FP	FILTER	INTRA PREDICTION
3	MC	IDCT	MDCT	MOTION COMPENSATION
4	ADD	MC	ITER. ENC.1	RESIDUAL ADDER
5	MEM 1	VLE	ITER. ENC.2	DEQUANT.
6	MEM2	MEM	BIT RES 1	IDCT
7	HUFF 1	BIT RES 1	BIT RES 2	LOOP FILTER
8	HUFF 2	BIT RES 2	BIT RES 3	
9	BIT RES 1	IMDCT	BIT RES 4	
10	BIT RES 2	SUM	IMDCT	
11	IMDCT	BUF	SUM	
12	SUM		BUF	
13	BUF			

Table 4.3: Comparison between [11] and TFG using 4-port routers

Benchmark	[11]		TFG			
	# of routers	$\mathbf{P}_{\mathbf{total}}$ (μW)	# of routers	$\mathrm{P}_{\mathrm{router}}$ (μW)	$\mathrm{P}_{\mathrm{link}}$ (μW)	$\mathbf{P}_{\mathbf{total}}$ (μW)
263 dec mp3 dec	6	**13.9**	6	13	1.1	**14.1**
263 enc mp3 dec	5	**194.6**	5	135.6	20	**156.6**
mp3 enc mp3 dec	6	**10.9**	6	9.4	0.7	**10.1**
H.264 BL@L4.1	N/A	**N/A**	3	11.5	2.4	**13.9**

4.3 Discussion

The results of using 4-port routers are listed in Table 4.3. The second and third columns indicate the number of routers used and total power consumption from [11]. The fourth to seventh columns show our results, where P_{router}, P_{link}, and P_{total} represents router power, physical link power, and total power, respectively.

It can be seen that physical link power is far less than router power. Please note that using the same number of routers and achieving competitive power consumption, we can guarantee deadlock free, but [11] cannot. Figure 4.1 shows that we generated a deadlock free topology of the 263 dec mp3 dec benchmark (see Figure 4.1(d)), while [11] generated a topology with deadlocks (highlighted by circles in Figure 4.1(b)). Figure 4.1(c) shows the post-processed deadlock-free topology of Figure 4.1(b), where the revised part is highlighted by a rectangle. Although the cycles between routers can be broken by introduce more routers, power consumption increases and latency constraints may be violated. It can be seen that Figure 4.1(c) requires one more router than Figure 4.1(b). The communication trace $(2, 4)$ in Figure 4.1(c) passes three routers, but its latency constraint is only two. So did the 263 enc mp3 dec benchmark in Figure 4.2. Figure 3.7 and Figure 4.3 demonstrate our results on mp3 enc mp3 dec and H.264 BL@L4.1.

Table 4.4: Comparison between Mesh, [11] and TFG using 5-port routers

Benchmark	Mesh		[11]		TFG	
	# of routers /Ratio	P_{total} (μW) /Ratio	# of routers /Ratio	P_{total} (μW) /Ratio	# of routers /Ratio	P_{total} (μW) /Ratio
263 dec mp3 dec	14/1.0	22.3/1.0	5/0.36	11.9/0.53	5/0.36	10.2/0.46
263 enc mp3 dec	12/1.0	273.7/1.0	5/0.42	179.5/0.66	4/0.33	115.9/0.42
mp3 enc mp3 dec	13/1.0	18.0/1.0	5/0.38	8.6/0.48	5/0.38	7.9/0.44

Table 4.4 compares the results of using 5-port routers between mesh (regular architecture), [11] and TFG. The results of Mesh are obtained in [11]. In the mesh architecture, a router connects only one core and four routers, so the number of required routers is at least the number of nodes in a CTG. On the contrary, [11] and TFG can connect cores to routers and routers to routers more flexibly. Experimental results show that TFG outperforms Mesh and [11]. It can be seen that, on average, TFG can save almost 64% of routers and reduce 56% power consumption on these three benchmarks with respect to Mesh. Moreover, compared with [11], TFG uses the same or less number of routers, consumes obviously lower power, guarantees deadlock free, and satisfies latency constraints.

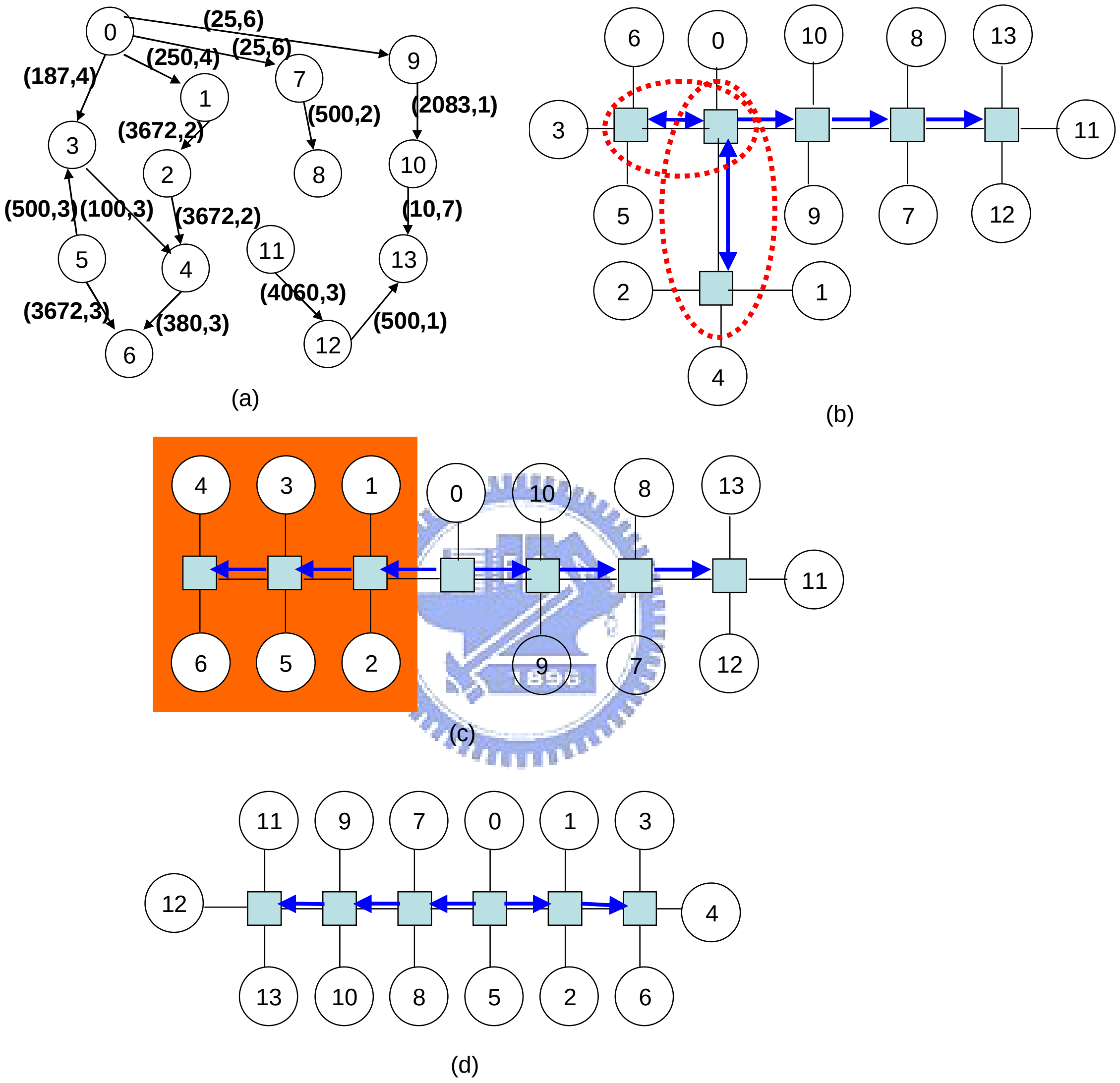
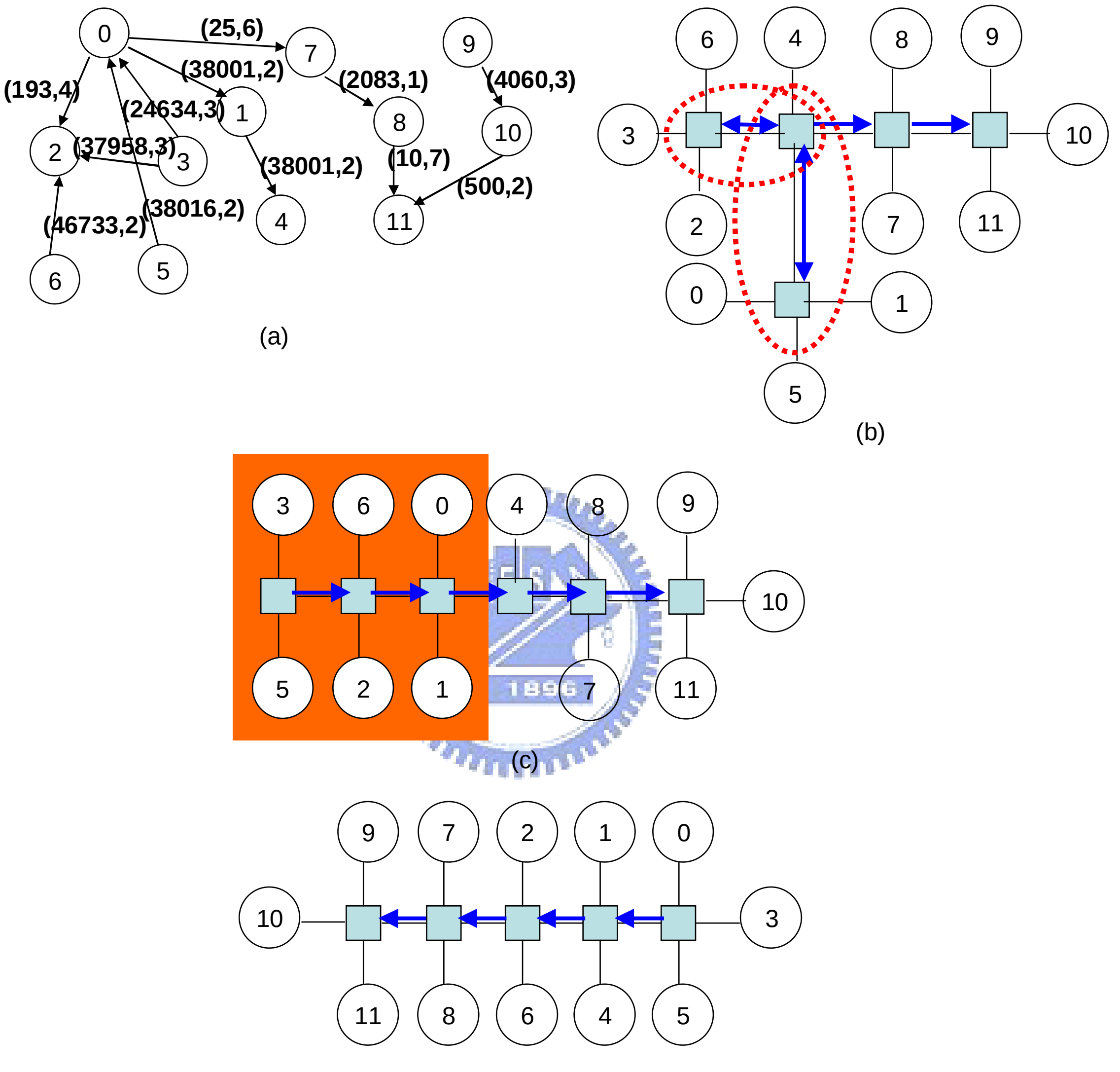


Figure 4.1: (a) The CTG of 263 dec mp3 dec with bandwidth in Kb/s. (b) The topology generated by [7], the deadlocks induced by traces (0, 1), (0, 3), (3, 4), and (4, 6) are highlighted by circles. (c) The post-processed deadlock-free topology of (b). (d) The deadlock-free topology generated by the TG algorithm.



(a)

(b)

(c)

(d)

Figure 4.2: (a) The 263 enc mp3 dec CTG with bandwidth in Kb/s. (b) The topology generated by [11], the deadlocks induced by traces (0, 2), (1, 4), (5, 0) are highlighted by circles. (c) The post-processed deadlock-free topology of (b). (d) The deadlock-free topology generated by the TG algorithm.

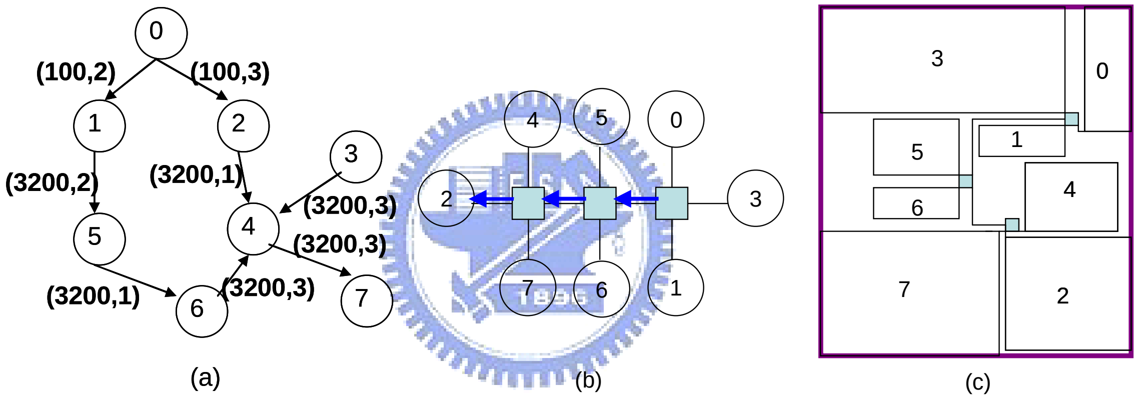


Figure 4.3: (a) The CTG and node description of H.264 BL@L4.1 with bandwidth in Mb/s. (b) The topology. (c) The floorplan.

Chapter 5

Conclusions

In this thesis, we proposed a two-phase flow—topology generation and then floorplanning, for low power application-specific NoCs. Unlike the time-consuming methods used in previous works, we adopted partition-based approaches in both phases. Experimental results showed that using the same or less number of routers, we can not only achieve competitive power consumption but also guarantee deadlock free and meet latency constraint. Moreover, with respect to Mesh, TFG can further save almost 64% of routers and reduce 56% power consumption on average.

Bibliography

[1] A. R. Agnihotri, S. Ono, and P. H. Madden. "Recursive Bisection Placement. Feng Shui 5.0 Implementation Details." In Proceedings of International Symposium on Physical Design, 2005, pp. 230–232.

[2] N. Banerjee, P. Vellanki, and K. S. Chatha. "A Power and Performance model for Network-on-Chip Architectures." In Proceedings of Design, Automation and Test in Europe Conference and Exhibition, 2004, Vol. 2, pp. 1250–1255.

[3] L. Benini. "Application Specific NoC Design." In Proceedings of Design, Automation and Test in Europe Conference and Exhibition, 2006, pp. 491–495.

[4] L. Benini and G. De Micheli. "Networks on Chips: A New SoC Paradigm." In IEEE Computer, 2002, Vol. 35 , No. 1, pp. 70–78.

[5] D. Bertozzi and L. Benini. "Xpipes. A Network-on-Chip Architecture for Gigascale Systems-on-Chip." In IEEE Circuits and Systems Magazine, 2004, Vol. 4, No. 2, pp. 18–31.

[6] W. J. Dally and C. L. Seitz. "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks." In IEEE Transactions on Computers, 1987, Vol. C-36, No. 5, pp. 547–553.

[7] W. J. Dally and B. Towels. "Route Packets, Not Wires: On-Chip Interconnection Networks." In Proceedings of Design Automation Conference, 2001, pp. 684–689.

[8] A. Jalabert, S. Murali, L. Benini, and G. De Micheli. "XpipesCompiler. A Tool for Instantiating Application Specific Network on Chips." In Proceedings of Design, Automation and Test in Europe Conference and Exhibition, 2004, Vol. 2, pp. 884–889.

[9] T.-M. Liu, T.-A. Lin, S.-Z. Wang, W.-P. Lee, K.-C. Hou, J.-Y. Yang and C.-Y. Lee. "An 865-μW H.264/AVC Video Decoder for Mobile Applications." In Proceedings of Asian Solid-State Circuit Conference, 2005, pp. 301–304.

[10] K. Srinivasan, K. S. Chatha, and G. Konjevod. "An Automated Technique for Topology and Route Generation for Application Specific on-Chip Interconnection Networks." In Proceedings of International Conference on Computer-Aided Design, 2005, pp. 231–237.

[11] K. Srinivasan, K. S. Chatha, and G. Konjevod. “Linear-Programming-Based Techniques for Synthesis of Network-on-Chip Architectures.” In IEEE Transactions on Very Large Scale Integration Systems, 2006, Vol. 14, No. 4, pp. 407–420.

[12] K. Srinivasan and K. S. Chatha.“A Low Complexity Heuristic for Design of Custom Network-on-Chip Architectures.” In Proceedings of Design, Automation and Test in Europe Conference and Exhibition, 2006, pp. 130–135.

[13] K. Srinivasan, K. S. Chatha, and G. Konjevod. “Application Specific Network-on-Chip Design with Guaranteed Quality Approximation Algorithms.” In Proceedings of Asia and South Pacific Design Automation Conference, 2007, pp. 184–190.

[14] S. Murali, P. Meloni, F. Angiolini, D.Atienza, S. Carta, L. Benini, G. De Micheli, and L. Raffo. “Designing Application-Specific Networks on Chips with Floorplan Information.” In Proceedings of International Conference on Computer-Aided Design, 2006, pp. 355–362.