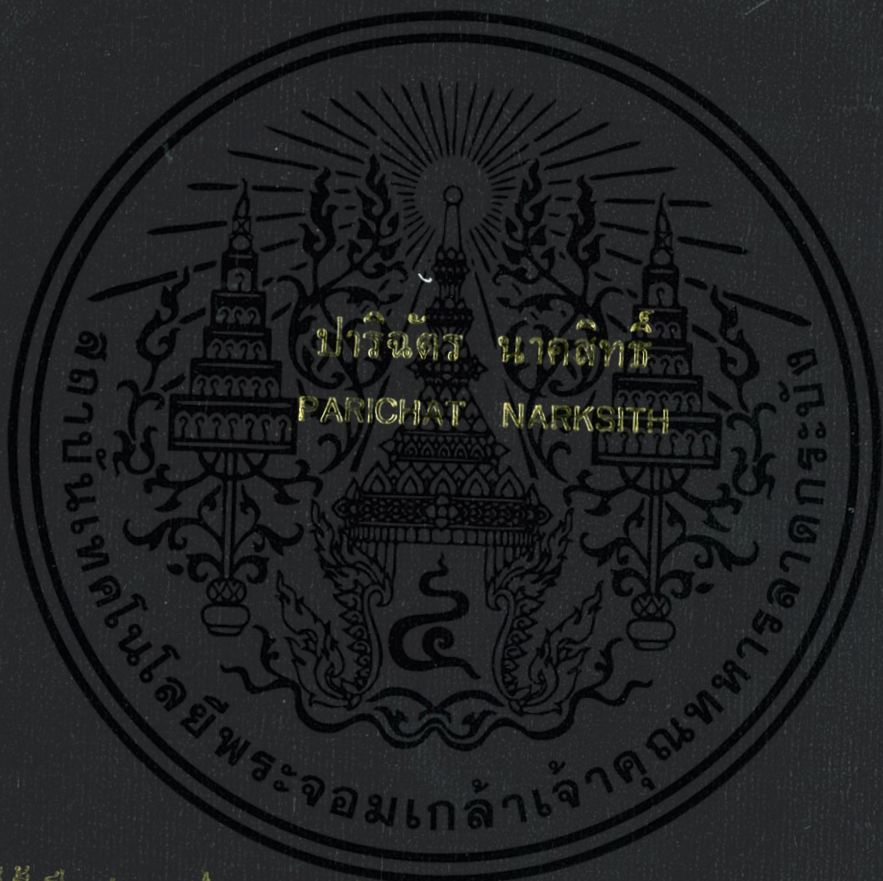


การปรับสวิตช์บนเครือข่ายรuffle-เอ็กซ์เชนจ์แบบมีลำดับชั้นสำหรับการ
แลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง

SWITCH ADJUSTING ON HIERARCHICAL SHUFFLE-EXCHANGE
NETWORKS FOR ALL-TO-ALL PERSONALIZED EXCHANGE



วิทยานิพนธ์เป็นส่วนหนึ่งของการศึกษาคณะวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

KMITL-2013-SC-M-002-024

การปรับสวิตช์บนเครือข่ายซัพเฟิลเอ็กซ์เชนจ์แบบมีลำดับชั้นสำหรับการ
แลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง

**SWITCH ADJUSTING ON HIERARCHICAL SHUFFLE-EXCHANGE
NETWORKS FOR ALL-TO-ALL PERSONALIZED EXCHANGE**



วิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์

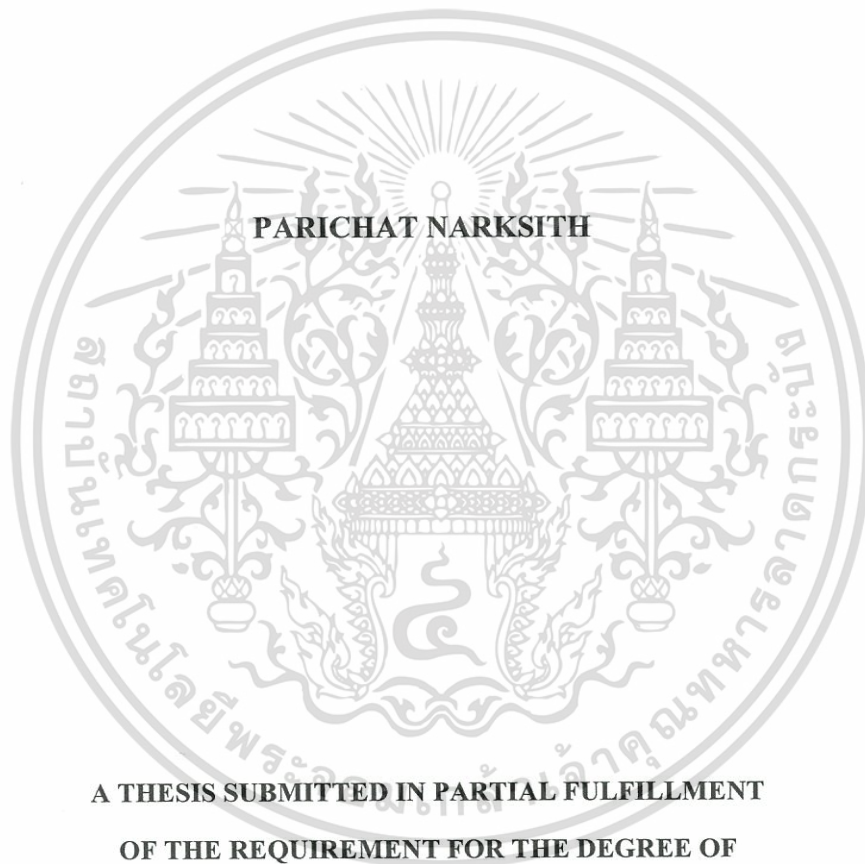
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2556

KMITL-2013-SC-M-002-024

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**SWITCH ADJUSTING ON HIERARCHICAL SHUFFLE-EXCHANGE
NETWORKS FOR ALL-TO-ALL PERSONALIZED EXCHANGE**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE IN COMPUTER SCIENCE**

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2013

KMITL-2013-SC-M-002-024

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2013

FACULTY OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์

การปรับสวิตช์บนเครือข่ายซัพเฟิลเอ็กซ์เชนจ์แบบมีลำดับชั้นสำหรับ
การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง
Switch Adjusting on Hierarchical Shuffle-Exchange Networks
for All-to-All Personalized Exchange

นักศึกษา

นางสาวปาริฉัตร นาคสิทธิ์

รหัสประจำตัว

54650803

ปริญญา

วิทยาศาสตรมหาบัณฑิต





สาขาวิชา

วิทยาการคอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์

รศ.ดร.จิรพร วีระพันธุ์

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม -

คณะกรรมการสอบวิทยานิพนธ์	ลายมือชื่อ
ดร.วรางคณา กิมปาน	
ผศ.ดร.ศรัณย์ อินทโกสม	
ดร.เฉลิมศักดิ์ เลิศวงศ์เสถียร	
รศ.ดร.จิรพร วีระพันธุ์	

วัน / เดือน / ปี ที่สอบ 16 พฤษภาคม พ.ศ. 2556 เวลา 15.00 – 18.00 น.
สถานที่สอบ ณ ห้อง 304 ชั้น 3 อาคารปฏิบัติการใหม่



(รองศาสตราจารย์ ดร.ศุภณัฐ ธีระพัฒน์)
คณบดีคณะวิทยาศาสตร์

วันที่ ๒3 เดือน พฤษภาคม พ.ศ. ๕6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	การปรับสวิตช์บนเครือข่ายซัพไฟลอิเล็กทรอนิกส์แบบมีลำดับชั้น สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบ เจาะจง
นักศึกษา	นางสาวปาริฉัตร นาคสิทธิ์
รหัสประจำตัว	54650803
ปริญญา	วิทยาศาสตรมหาบัณฑิต
สาขาวิชา	วิทยาการคอมพิวเตอร์
พ.ศ.	2556
อาจารย์ที่ปรึกษาวิทยานิพนธ์	รศ.ดร.จิรพร วีระพันธุ์

บทคัดย่อ

ขั้นตอนการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงดั้งเดิมนั้น ถูกนำมาใช้
ในเครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบอพลวัต (เช่น เครือข่ายไฮเปอร์คิวบ์
เครือข่ายเมฆ และเครือข่ายทอรัส เป็นต้น) และต่อมากการแลกเปลี่ยนข้อมูลประเภทดังกล่าวถูกเสนอ
ขึ้นสำหรับเครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบพลวัต หรือเครือข่ายที่มีการ
ติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ (เช่น เครือข่ายโอเมก้าหรือซัพไฟลอิเล็กทรอนิกส์
เซนจ์ เครือข่ายฟิลิป เครือข่ายคิวบ์ และเครือข่ายเบสไลน์ เป็นต้น) ซึ่งมีหน่วยประมวลผลภายใน
เครือข่ายเท่ากับ $N = 2^n$ ($= 4, 8, 16, 32, \dots$) จนกระทั่งในปัจจุบัน งานวิจัยด้านนี้ได้มุ่งเน้นไปที่การ
ออกแบบการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงสำหรับเครือข่ายซัพไฟลอิเล็กทรอนิกส์
เซนจ์ ที่มีหน่วยประมวลผลภายในเครือข่าย (N) เป็นจำนวนคู่ ($= 4, 6, 8, 10, \dots$) แต่อย่างไรก็ตาม
ขั้นตอนวิธีเหล่านั้น ไม่สามารถนำมาประยุกต์ใช้กับเครือข่ายซัพไฟลอิเล็กทรอนิกส์เซนจ์แบบมีลำดับชั้น
โดยตรงได้ งานวิจัยนี้จึงเสนอเครือข่ายซัพไฟลอิเล็กทรอนิกส์เซนจ์แบบมีลำดับชั้น พร้อมกับฟังก์ชันการปรับ
สวิตช์ โดยการดำเนินการบนวงจรการเลื่อนบิตภายในหน่วยประมวลผลด้วยเวลา $O(1)$ สำหรับการ
แลกเปลี่ยนข้อมูลประเภทดังกล่าว รวมทั้งการติดต่อสื่อสารประเภทหนึ่งไปหนึ่งแบบขนานอื่นๆ ที่
สำเร็จได้ สุดท้ายการปรับสวิตช์บนเครือข่ายซัพไฟลอิเล็กทรอนิกส์เซนจ์แบบมีลำดับชั้นจะถูกทดสอบความ
ถูกต้องสำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง

คำสำคัญ : การปรับสวิตช์, เครือข่ายซัพไฟลอิเล็กทรอนิกส์เซนจ์แบบมีลำดับชั้น, เครือข่ายที่มีการ
ติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ, การแลกเปลี่ยนข้อมูลประเภททั้งหมดไป
ทั้งหมดแบบเจาะจง, การติดต่อสื่อสารประเภทหนึ่งไปหนึ่งแบบขนาน

Thesis Title	Switch Adjusting on Hierarchical Shuffle-Exchange Networks for All-to-All Personalized Exchange
Student	Miss Parichat Narksith
Student ID	54650803
Degree	Master of Science
Program	Computer Science
Year	2013
Thesis Advisor	Assoc. Prof. Dr. Jeeraporn Werapun

ABSTRACT

The original all-to-all personalized exchange algorithms were introduced for static interconnection networks (i.e., hypercube, mesh, and torus). Later, the all-to-all personalized exchange methods were extended for dynamic interconnection networks or MINs (Multistage Interconnection Networks i.e., omega (or shuffle exchange), flip, cube, and baseline) with $N = 2^n$ ($= 4, 8, 16, 32, \dots$). Recently, the generalized all-to-all personalized exchange algorithm was designed for shuffle-exchange networks with even N ($= 4, 6, 8, 10, \dots$). However, for a hierarchical shuffle-exchange network, those existing algorithms cannot be directly applied. This paper introduces such a hierarchical MIN along with switch-adjusting function, operated by shift logic in $O(1)$, for applying all-to-all personalized exchange and other parallel point-to-point communications. Finally, correctness of our switch adjusting on hierarchical shuffle-exchange networks is verified for all-to-all personalized exchange permutations.

Keywords : switch adjusting, hierarchical shuffle-exchange networks, multistage interconnection networks, all-to-all personalized exchange, parallel point-to-point communications

กิตติกรรมประกาศ

วิทยานิพนธ์นี้ ตั้งแต่เริ่มต้นจนสำเร็จลุล่วงนั้น นับว่าเป็นความภาคภูมิใจของผู้วิจัยเป็นอย่างมาก หากคำพิงผู้วิจัยเพียงผู้เดียวคงไม่สามารถดำเนินงานจนประสบความสำเร็จเช่นนี้ได้ ทั้งนี้เพราะได้รับความกรุณาจากบุคคลหลายๆ ท่านดังนี้

ขอขอบพระคุณ รองศาสตราจารย์ ดร.จิรพร วีระพันธุ์ อาจารย์ที่ปรึกษา ที่ได้สละเวลาให้คำแนะนำ ดูแลเอาใจใส่ รวมทั้งให้ข้อคิด ความรู้ที่มีประโยชน์ที่ใช้ประกอบการทำงานวิจัยจนสำเร็จลุล่วง

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ศรัณย์ อินทโกสุม ดร.วรางคณา กัมปาน และ ดร.เฉลิมศักดิ์ เลิศวงศ์เสถียร คณะกรรมการสอบวิทยานิพนธ์ ที่กรุณาให้คำแนะนำตลอดจนข้อชี้แนะจนในที่สุดทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลงได้

ขอขอบคุณ คุณเสกฐันท์ ทองสุวรรณ ที่คอยช่วยเหลือในทุกๆ เรื่อง และเป็นกำลังใจให้กันเสมอมา

ท้ายสุดขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่คอยห่วงใย เป็นกำลังใจ ดูแลและผลักดันสนับสนุนด้านการศึกษาและทุกๆ ด้านด้วยดีตลอดมา

ประโยชน์อันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้แก่คุณพ่อ คุณแม่ อาจารย์ทุกท่านซึ่งเป็นที่เคารพรักยิ่ง ตลอดจนญาติพี่น้อง และเพื่อนๆ ทุกคน

ปาริฉัตร นาคสิทธิ์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญรูป.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	3
1.3 สมมติฐานของการศึกษา.....	3
1.4 ขอบเขตการวิจัย.....	3
1.5 ขั้นตอนการศึกษาและการดำเนินงานวิจัย.....	4
1.6 ประโยชน์ที่คาดว่าจะได้รับ.....	4
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	6
2.1 เครื่องข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ.....	6
2.1.1 องค์ประกอบของเครื่องข่าย MIN.....	7
2.1.1.1 หน่วยประมวลผล.....	7
2.1.1.2 สวิตช์.....	7
2.1.1.3 สเตจ.....	8
2.1.1.4 รูปแบบการเชื่อมต่อระหว่างสเตจ.....	9
2.1.2 ประเภทของเครื่องข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ.....	9
2.1.2.1 เครื่องข่ายโอเมก้าหรือซฟเฟิลเอ็กซ์เชนจ์.....	10
2.1.2.2 เครื่องข่ายฟิลิป.....	12
2.1.2.3 เครื่องข่ายคิวบ์.....	14
2.1.2.4 เครื่องข่ายเบสไลน์.....	17
2.2 ฟังก์ชันการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง.....	19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา แIVต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.2.1 สี่เหลี่ยมละติน (Latin Square).....	19
2.2.2 การแลกเปลี่ยนข้อมูลประเภททั้งหมด ไปทั้งหมดแบบเจาะจงบนเครือข่าย MIN.....	20
บทที่ 3 การปรับสวิตช์บนเครือข่ายซัพไฟลเธิ์กซ์เซนจ์แบบมีลำดับชั้นสำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง	26
3.1 โครงสร้างของเครือข่ายซัพไฟลเธิ์กซ์เซนจ์แบบบางส่วน (PSEN)	26
3.2 การปรับสวิตช์บนเครือข่ายซัพไฟลเธิ์กซ์เซนจ์แบบบางส่วน (PSEN).....	31
3.2.1 การปรับสวิตช์บนเครือข่าย PSEN	32
3.2.2 ความแตกต่างระหว่างการปรับสวิตช์สำหรับเครือข่าย MIN ที่มีอยู่กับการปรับสวิตช์สำหรับเครือข่าย PSEN	32
3.3 โครงสร้างของเครือข่ายซัพไฟลเธิ์กซ์เซนจ์แบบมีลำดับชั้น (HSEN)	35
3.4 การปรับสวิตช์บนเครือข่ายซัพไฟลเธิ์กซ์เซนจ์แบบมีลำดับชั้น (HSEN).....	36
3.4.1 การปรับสวิตช์บนเครือข่าย HSEN.....	36
3.4.2 ความแตกต่างระหว่างการปรับสวิตช์สำหรับเครือข่าย MIN ที่มีอยู่กับการปรับสวิตช์สำหรับเครือข่าย HSEN.....	37
บทที่ 4 การวิเคราะห์ความถูกต้องและความซับซ้อนด้านเวลาของการปรับสวิตช์บนเครือข่าย PSEN และ HSEN.....	48
บทที่ 5 บทสรุปและแนวทางการพัฒนางานวิจัย.....	53
5.1 บทสรุป	53
5.2 แนวทางการพัฒนางานวิจัย	54
เอกสารอ้างอิง	55
ภาคผนวก.....	56
ประวัติผู้เขียน	64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 การเชื่อมต่อภายในสวิทช์บนเครือข่าย MIN ที่ดำเนินการสำเร็จด้วย $\rho = (4\ 5\ 6\ 7\ 0\ 1\ 2\ 3)$	18
3.1 การเชื่อมต่อภายในสวิทช์บนเครือข่าย MIN ด้วย $\rho = (4\ 5\ 6\ 7\ 0\ 1\ 2\ 3)$	30
3.2 การเชื่อมต่อภายในสวิทช์บนเครือข่าย PSEN ด้วย $\rho = (4\ 0\ 5\ 1\ 6\ 2\ 7\ 3)$	31
3.3 การปรับสวิทช์บนเครือข่าย PSEN ที่ดำเนินการสำเร็จสำหรับ $\rho = (4\ 0\ 5\ 1\ 6\ 2\ 7\ 3)$ ด้วยวิธีการปรับสวิทช์ที่เสนอ.....	33
3.4 การปรับสวิทช์บนเครือข่าย HSEN ที่ดำเนินการสำเร็จสำหรับ $\rho = (4\ 12\ 6\ 14\ 0\ 8\ 2\ 10\ 5\ 13\ 7\ 15\ 1\ 9\ 3\ 11)$ ด้วยเงื่อนไขการปรับสวิทช์ที่เสนอ.....	38
3.5 ผลการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จทั้ง N รอบ.....	47



สารบัญรูป

รูปที่	หน้า
2.1 การเชื่อมต่อเพื่อการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ	7
2.2 การเชื่อมต่อภายในสวิตช์ขนาด 2×2	8
2.3 เชื่อมต่อภายในสวิตช์เพื่อกระจายข้อมูลภายในสวิตช์ขนาด 2×2	8
2.4 เครือข่ายซัพเฟิลเอ็กซ์เชนจ์ ($N = 8$).....	9
2.5 รูปแบบการเชื่อมต่อระหว่างสเตจแบบเพอร์เฟคซัพเฟิล สำหรับ $N = 8$	10
2.6 รูปแบบการเชื่อมต่อระหว่างสเตจภายในเครือข่ายซัพเฟิลเอ็กซ์เชนจ์ ($N = 8$)	11
2.7 การสื่อสารข้อมูลระหว่างหน่วยประมวลผล 3 ไป 7 สำหรับเครือข่ายซัพเฟิลเอ็กซ์เชนจ์ ($N = 8$).....	12
2.8 รูปแบบการเชื่อมต่อระหว่างสเตจแบบอินเวิร์ตเพอร์เฟคซัพเฟิล สำหรับ $N = 8$	12
2.9 รูปแบบการเชื่อมต่อระหว่างสเตจภายในเครือข่ายฟลิป ($N = 8$)	13
2.10 การสื่อสารข้อมูลระหว่างหน่วยประมวลผล 3 ไป 7 สำหรับเครือข่ายฟลิป ($N = 8$)	14
2.11 รูปแบบการเชื่อมต่อระหว่างสเตจ ISC_1 ภายในเครือข่ายคิวบ์ ($N = 8$).....	15
2.12 รูปแบบการเชื่อมต่อระหว่างสเตจ ISC_2 ภายในเครือข่ายคิวบ์ ($N = 8$).....	15
2.13 รูปแบบการเชื่อมต่อระหว่างสเตจ ISC_3 ภายในเครือข่ายคิวบ์ ($N = 8$).....	15
2.14 รูปแบบการเชื่อมต่อระหว่างสเตจภายในเครือข่ายคิวบ์ ($N = 8$).....	16
2.15 การสื่อสารข้อมูลระหว่างหน่วยประมวลผล 3 ไป 7 สำหรับเครือข่ายคิวบ์ ($N = 8$).....	17
2.16 รูปแบบการเชื่อมต่อระหว่างสเตจภายในเครือข่ายเบสไลน์ ($N = 8$).....	18
2.17 การสื่อสารข้อมูลระหว่างหน่วยประมวลผล 3 ไป 7 สำหรับเครือข่ายเบสไลน์ ($N = 8$).....	18
2.18 การปรับสวิตช์บนเครือข่ายซัพเฟิลเอ็กซ์เชนจ์ ($N = 8$) ที่ดำเนินการสำเร็จด้วย $p = (4\ 5\ 6\ 7$ $0\ 1\ 2\ 3)$	21
2.19 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายซัพเฟิลเอ็กซ์เชนจ์ ($N = 8$) ที่ดำเนินการสำเร็จทั้ง N รอบ	22
2.20 การเชื่อมต่อภายในสวิตช์บนเครือข่ายฟลิป ($N = 8$) ที่ดำเนินการสำเร็จด้วย $p = (4\ 5\ 6\ 7\ 0$ $1\ 2\ 3)$	23
2.21 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายฟลิป ($N = 8$) ที่ ดำเนินการสำเร็จทั้ง N รอบ.....	23
2.22 การเชื่อมต่อภายในสวิตช์บนเครือข่ายคิวบ์ ($N = 8$) ที่ดำเนินการสำเร็จด้วย $p = (4\ 5\ 6\ 7\ 0$ $1\ 2\ 3)$	24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และVIIของอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญญรูป (ต่อ)

รูปที่	หน้า
2.23 การแลกเปลี่ยนข้อมูลประเภททั้งหมด ไปทั้งหมดแบบเจาะจงบนเครือข่ายคิวบ์ ($N=8$) ที่ดำเนินการสำเร็จทั้ง N รอบ.....	24
2.24 การแลกเปลี่ยนข้อมูลประเภททั้งหมด ไปทั้งหมดแบบเจาะจงบนเครือข่ายเบสไลน์ ($N=8$) ที่ดำเนินการสำเร็จทั้ง N รอบ.....	25
3.1 รูปแบบการเชื่อมต่อระหว่างสแตจภายในเครือข่าย PSEN ($N=8$).....	27
3.2 การหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย PSEN ($N=8$) ที่ดำเนินการไม่สำเร็จ ($id: 3 \rightarrow id: 7$).....	28
3.3 การหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย PSEN ($N=8$) ที่ดำเนินการสำเร็จ ($id: 7 \rightarrow id: 3$).....	29
3.4 การแลกเปลี่ยนข้อมูลประเภททั้งหมด ไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN ($N=8$) ที่ไม่สำเร็จเนื่องจากการเรียงสับเปลี่ยน	30
3.5 การแลกเปลี่ยนข้อมูลประเภททั้งหมด ไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN ($N=8$) ที่ไม่สำเร็จเนื่องจากเกิดการขัดแย้งในสวิตช์	31
3.6 เงื่อนไขการเชื่อมต่อภายในสวิตช์สำหรับการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย MIN อื่นๆ.....	33
3.7 เงื่อนไขการปรับสวิตช์สำหรับการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย PSEN	33
3.8 การแลกเปลี่ยนข้อมูลประเภททั้งหมด ไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN ($N=8$) ที่ดำเนินการสำเร็จด้วยเงื่อนไขการปรับสวิตช์ที่เสนอ.....	34
3.9 การแลกเปลี่ยนข้อมูลประเภททั้งหมด ไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN ($N=8$) ที่ดำเนินการสำเร็จทั้ง N รอบ.....	34
3.10 โครงสร้างของเครือข่ายซัพเฟิลเอ็กซ์เชนจ์ ($N=16$).....	35
3.11 โครงสร้างของเครือข่ายซัพเฟิลเอ็กซ์เชนจ์แบบมีลำดับชั้น ($N=16$).....	36
3.12 เงื่อนไขการปรับสวิตช์สำหรับการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย HSEN	37
3.13 การปรับสวิตช์บนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จสำหรับ $\rho = (4\ 12\ 6\ 14\ 0\ 8\ 2\ 10\ 5\ 13\ 7\ 15\ 1\ 9\ 3\ 11)$	39

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.14 การสื่อสารรอบที่ 1 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (0\ 8\ 2\ 10\ 4\ 12\ 6\ 14\ 1\ 9\ 3\ 11\ 5\ 13\ 7\ 15)$	39
3.15 การสื่อสารรอบที่ 2 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (1\ 9\ 3\ 11\ 5\ 13\ 7\ 15\ 0\ 8\ 2\ 10\ 4\ 12\ 6\ 14)$	40
3.16 การสื่อสารรอบที่ 3 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (2\ 10\ 0\ 8\ 6\ 14\ 4\ 12\ 3\ 11\ 1\ 9\ 7\ 15\ 5\ 13)$	40
3.17 การสื่อสารรอบที่ 4 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (3\ 11\ 1\ 9\ 7\ 15\ 5\ 13\ 2\ 10\ 0\ 8\ 6\ 14\ 4\ 12)$	41
3.18 การสื่อสารรอบที่ 5 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (4\ 12\ 6\ 14\ 0\ 8\ 2\ 10\ 5\ 13\ 7\ 15\ 1\ 9\ 3\ 11)$	41
3.19 การสื่อสารรอบที่ 6 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (5\ 13\ 7\ 15\ 1\ 9\ 3\ 11\ 4\ 12\ 6\ 14\ 0\ 8\ 2\ 10)$	42
3.20 การสื่อสารรอบที่ 7 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (6\ 14\ 4\ 12\ 2\ 10\ 0\ 8\ 7\ 15\ 5\ 13\ 3\ 11\ 1\ 9)$	42
3.21 การสื่อสารรอบที่ 8 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (7\ 15\ 5\ 14\ 3\ 11\ 1\ 9\ 6\ 14\ 4\ 12\ 2\ 10\ 0\ 8)$	43
3.22 การสื่อสารรอบที่ 9 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (8\ 0\ 10\ 2\ 12\ 4\ 14\ 6\ 9\ 1\ 11\ 3\ 13\ 5\ 15\ 7)$	43

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.23 การสื่อสารรอบที่ 10 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (9\ 1\ 11\ 2\ 13\ 5\ 15\ 7\ 8\ 0\ 10\ 2\ 12\ 4$ $14\ 6)$	44
3.24 การสื่อสารรอบที่ 11 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (10\ 2\ 8\ 0\ 14\ 6\ 12\ 4\ 11\ 3\ 9\ 1\ 15\ 7$ $13\ 5)$	44
3.25 การสื่อสารรอบที่ 12 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (11\ 3\ 9\ 1\ 15\ 7\ 13\ 5\ 10\ 2\ 8\ 0\ 14\ 6$ $12\ 4)$	45
3.26 การสื่อสารรอบที่ 13 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (12\ 4\ 14\ 6\ 8\ 0\ 10\ 2\ 13\ 5\ 15\ 7\ 9\ 1$ $11\ 3)$	45
3.27 การสื่อสารรอบที่ 14 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (13\ 5\ 15\ 7\ 9\ 1\ 11\ 3\ 12\ 4\ 14\ 6\ 8\ 0$ $10\ 2)$	46
3.28 การสื่อสารรอบที่ 15 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (14\ 6\ 12\ 4\ 10\ 2\ 8\ 0\ 15\ 7\ 13\ 5\ 11\ 3$ $9\ 1)$	46
3.29 การสื่อสารรอบที่ 16 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบน เครือข่าย HSEN ($N = 16$) ที่ดำเนินการสำเร็จด้วย $\rho = (15\ 7\ 13\ 5\ 11\ 3\ 6\ 1\ 14\ 6\ 12\ 4\ 10\ 2$ $8\ 0)$	47

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบคอมพิวเตอร์ในปัจจุบันประกอบไปด้วยหน่วยประมวลผลหลายหน่วยประมวลผล ซึ่งการเชื่อมต่อระหว่างหน่วยประมวลผลภายในเครือข่าย (Interconnection Networks) มีผลต่อ ประสิทธิภาพของการประมวลผลแบบขนาน (Parallel Processing) และการสื่อสารแบบขนาน ระหว่างหน่วยประมวลผลเหล่านั้น จะสื่อสารกันผ่านการเชื่อมต่อระหว่างหน่วยประมวลผลภายใน เครือข่ายซึ่งอาจจะต้องใช้เวลามาก โดยขึ้นอยู่กับประเภทของการเชื่อมต่อ

การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง (All-to-All Personalized Exchange) ระหว่างหน่วยประมวลผลภายในเครือข่าย เป็นรูปแบบการสื่อสารหนึ่งที่สำคัญอย่าง มากสำหรับการสื่อสารระหว่างหน่วยประมวลผลภายในระบบการประมวลผลแบบขนานและแบบ กระจาย (Parallel and Distributed Computing Systems) โดยเฉพาะอย่างยิ่งในการประมวลผลฟาสต์ ฟูเรียร์ทรานสฟอร์ม (FFT: Fast Fourier Transform) การทรานสโพสต์เมทริกซ์ (Matrix Transposition) [1-5] และการค้นหาข้อมูลในตารางแบบกระจาย (Distributed Table Lookup) [3] การทำงานของรูปแบบการสื่อสารดังกล่าวนี้ คือ ทุกๆ หน่วยประมวลผล (N หน่วยประมวลผล) ภายในระบบ จะส่งข้อความสำหรับการสื่อสารที่แตกต่างกัน (Distinct) ไปยังปลายทาง (Destination) พร้อมๆ กัน (Parallel Send/Receive) N รอบ โดยสุดท้ายแล้ว ทุกๆ หน่วยประมวลผล จะได้รับข้อความจากหน่วยประมวลผลอื่นทั้งหมดในระบบ ส่วนการแลกเปลี่ยนข้อมูลระหว่าง หน่วยประมวลผลภายในเครือข่ายอีกแบบหนึ่ง คือ การแลกเปลี่ยนข้อมูลประเภททั้งหมดไป ทั้งหมดแบบกระจายข้อมูล (All-to-All Broadcast) ซึ่งแต่ละหน่วยประมวลผลจะส่งข้อความออกไป ยังหน่วยประมวลผลอื่นในระบบ โดยข้อความที่ปลายทางได้รับจากต้นทาง (Source) ซึ่งหน่วย ประมวลผลหน่วยใดหน่วยหนึ่งนั้น จะเป็นข้อความที่เหมือนกันทั้งหมด

การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง เริ่มแรกถูกนำเสนอขึ้น สำหรับเครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบอพลวัต (Static Interconnection Networks) เช่น เครือข่ายไฮเปอร์คิวบ์ (Hypercube Network) เครือข่ายเมช (Mesh Network) และเครือข่ายทอรัส (Torus Network) เป็นต้น ถึงแม้ว่าอัลกอริทึมสำหรับการแลกเปลี่ยน ข้อมูลของเครือข่ายไฮเปอร์คิวบ์มีความซับซ้อนด้านเวลาที่ดีที่สุด แต่เครือข่ายนี้ก็ยังมีข้อจำกัดเรื่อง การขยายขนาดเครือข่าย และในทางกลับกัน เครือข่ายเมชและทอรัสมีข้อดีในเรื่องการขยายขนาด แต่ความซับซ้อนด้านเวลาในการแลกเปลี่ยนข้อมูลสูงกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมาการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง ถูกเสนอขึ้นสำหรับเครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบพลวัต (Dynamic Interconnection Network) หรือเรียกว่า เครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ (Multistage Interconnection Network: MIN) โดย Yang และ Wang [3] ได้เสนออัลกอริทึมที่ดีที่สุดเพื่อการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงสำหรับเครือข่าย MIN ขึ้นเป็นครั้งแรกในปี 2000 และยังได้กล่าวไว้ว่า เครือข่ายดังกล่าวเหมาะสำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง เนื่องจากความสามารถในการขยายขนาดของเครือข่ายที่ดีกว่าและใช้เวลาในการสื่อสารภายในเครือข่ายที่น้อยกว่าเครือข่ายแบบพลวัต ตัวอย่างเครือข่าย MIN ได้แก่ เครือข่ายโอเมก้า (Omega Network) หรือชัฟเฟิลเอ็กซ์เชนจ์ (Shuffle-Exchange Network) เครือข่ายฟลิป (Flip Network) เครือข่ายคิวบ์ (Cube Network) และเครือข่ายเบสไลน์ (Baseline Network) เป็นต้น เครือข่าย MIN แต่ละแบบประกอบไปด้วยหน่วยประมวลผล N หน่วย โดยที่ $N = 2^n$ ($= 4, 8, 16, 32, \dots$) อัลกอริทึมดังกล่าวพัฒนาบนพื้นฐานสี่เหลี่ยมละติน (Latin Square) เพื่อการปรับสวิตช์ที่รวดเร็ว สี่เหลี่ยมละตินประกอบไปด้วยเซตการเรียงสับเปลี่ยนของการติดต่อสื่อสารระหว่างคู่ต้นทางกับปลายทางที่เป็นไปได้ (A Set of Admissible Permutations) ซึ่งเป็นสับเซตของการเรียงสับเปลี่ยนทั้งหมด $N!$ แต่อย่างไรก็ตามอัลกอริทึมสำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงที่เคยถูกเสนอขึ้นไม่สามารถนำมาใช้กับเครือข่ายชัฟเฟิลเอ็กซ์เชนจ์แบบทั่วไป (Generalized Shuffle-Exchange Network: GSEN) [5] ได้ เครือข่ายดังกล่าวเป็นเครือข่าย MIN ที่มีหน่วยประมวลผลภายในเครือข่ายเป็นจำนวนคู่ ($= 4, 6, 8, 10, \dots$) ซึ่งปัจจุบัน Chou และ Chen [5] ได้มุ่งเน้นการพัฒนาอัลกอริทึมสำหรับเครือข่ายดังกล่าว

วิธีการสำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงที่มีอยู่ และเซตการเรียงสับเปลี่ยนของการติดต่อสื่อสารของคู่ต้นทางกับปลายทางแบบขนาน (Parallel Point-to-Point Permutations) อาจมีความเฉพาะกับบางเครือข่าย เช่น เครือข่ายชัฟเฟิลเอ็กซ์เชนจ์ เครือข่ายฟลิป เครือข่ายคิวบ์ และเครือข่ายเบสไลน์ เป็นต้น แต่อย่างไรก็ตาม อัลกอริทึมดังกล่าวไม่สามารถนำมาประยุกต์ใช้กับเครือข่ายชัฟเฟิลเอ็กซ์เชนจ์แบบมีลำดับชั้น (Hierarchical Shuffle-Exchange Network: HSEN) ได้โดยตรง เนื่องจากเครือข่ายดังกล่าวถูกสร้างขึ้นด้วยโครงสร้างพิเศษที่แตกต่างจากรูปแบบเดิมที่ไม่ต้องมีการปรับสวิตช์ (Switch Adjusting) ก่อนทำการหาเส้นทางด้วยตนเอง (Self-Routing) โดยเครือข่าย HSEN เป็นเครือข่ายที่สร้างขึ้นจากการรวมระบบเล็ก (Smaller Subsystems) ขึ้นเป็นระบบใหญ่ ซึ่งเป็นวิธีการที่เหมาะสมสำหรับการขยายขนาดเครือข่ายของระบบการประมวลผลแบบขนานที่ใช้งานอยู่จริง เนื่องจากการสร้างระบบใหม่ให้ได้ขนาดตามความต้องการที่เพิ่มขึ้นอย่างรวดเร็วในปัจจุบันนั้น ต้องใช้เวลามากกว่าและมีค่าใช้จ่ายที่สูงกว่าการนำระบบเดิมที่มีอยู่แล้วมาเพิ่มขนาดเครือข่าย แต่เครือข่าย HSEN ต้องการเงื่อนไขที่เหมาะสมในการปรับสวิตช์สำหรับเครือข่ายที่มีโครงสร้างพิเศษนี้ ดังนั้นในงานวิจัยนี้จึงเสนอฟังก์ชันการปรับสวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเครือข่าย HSEN โดยใช้การเลื่อนบิตทางฮาร์ดแวร์ (Shift Logic) สำหรับการประยุกต์ใช้การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง รวมทั้งฟังก์ชันการติดต่อสื่อสารแบบขนานอื่นๆ บนเครือข่าย HSEN ซึ่งมีความซับซ้อนด้านเวลาเท่ากับ $O(1)$

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

วิทยานิพนธ์นี้มีวัตถุประสงค์เพื่อศึกษาและออกแบบฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงระหว่างหน่วยประมวลผลภายในเครือข่ายซีฟเฟิลเอ็กซ์เชนจ์แบบมีลำดับชั้น (HSEN) ที่มีหน่วยประมวลผลเท่ากับ N ซึ่งประกอบไปด้วยเครือข่ายย่อย คือ เครือข่ายซีฟเฟิลเอ็กซ์เชนจ์แบบบางส่วน (Partial Shuffle-Exchange Network: PSEN) ที่มีหน่วยประมวลผลเท่ากับ $N/2$ จำนวน 2 เครือข่ายย่อย โดยฟังก์ชันการปรับสวิตช์สำหรับเครือข่ายดังกล่าวนี้ สามารถทำให้การสื่อสารระหว่างคู่ต้นทางกับปลายทางสามารถสื่อสารกันได้อย่างถูกต้องสำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงและเซตการเรียงสับเปลี่ยนของคู่ต้นทางกับปลายทางแบบขนานบนเครือข่าย HSEN

1.3 สมมติฐานของการศึกษา

ฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงระหว่างหน่วยประมวลผลภายในเครือข่าย MIN ที่มีอยู่ อาทิ เครือข่าย โอเมก้าหรือซีฟเฟิลเอ็กซ์เชนจ์ เครือข่ายฟิลิป เครือข่ายคิวบ์ และเครือข่ายเบสไลน์ ไม่สามารถปรับใช้กับเครือข่ายซีฟเฟิลเอ็กซ์เชนจ์แบบมีลำดับชั้น (HSEN) โดยตรงได้ เนื่องจากเมื่อประยุกต์ใช้แล้วทำให้การแลกเปลี่ยนข้อมูลระหว่างหน่วยประมวลผล จากต้นทางไม่สามารถส่งถึงปลายทางที่ถูกต้องได้ ดังนั้นงานวิจัยนี้จึงออกแบบฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN เพื่อให้สามารถแลกเปลี่ยนข้อมูลระหว่างหน่วยประมวลผลต้นทางและปลายทางได้ถูกต้อง

1.4 ขอบเขตการวิจัย

วิทยานิพนธ์นี้มีขอบเขตการวิจัยเพื่อออกแบบฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายซีฟเฟิลเอ็กซ์เชนจ์แบบมีลำดับชั้น (HSEN) ที่มีหน่วยประมวลผลเท่ากับ $N = 2^n$

1.5 ขั้นตอนการศึกษาและดำเนินงานวิจัย

วิทยานิพนธ์นี้มีขั้นตอนการศึกษาและการดำเนินงานวิจัย ดังนี้

- 1) ศึกษาเครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ (MIN)
- 2) ศึกษาฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย MIN อาทิ เครือข่ายโอเมก้าหรือซัพเฟิลเอ็กซ์เซนจ์ เครือข่ายฟิลิป เครือข่ายคิวบ์ และเครือข่ายเบสไลน์ เป็นต้น
- 3) ตั้งสมมติฐาน โดยคาดว่าฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายแบบ MIN ที่มีอยู่ เช่น เครือข่ายโอเมก้าหรือซัพเฟิลเอ็กซ์เซนจ์ เครือข่ายฟิลิป เครือข่ายคิวบ์ และเครือข่ายเบสไลน์ ไม่สามารถปรับใช้กับเครือข่าย HSEN โดยตรงได้
- 4) ออกแบบฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN ให้สามารถแลกเปลี่ยนข้อมูลระหว่างหน่วยประมวลผลต้นทางและปลายทางได้ถูกต้อง เพื่อเป็นเครือข่ายย่อยสำหรับการสร้างเครือข่าย HSEN
- 5) ออกแบบเครือข่าย HSEN พร้อมตั้งสมมติฐาน โดยคาดว่าฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย MIN ที่มีอยู่ และเครือข่าย PSEN ที่ออกแบบใหม่นั้น ไม่สามารถปรับใช้กับเครือข่าย HSEN ได้โดยตรง
- 6) ออกแบบฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN ให้สามารถแลกเปลี่ยนข้อมูลระหว่างหน่วยประมวลผลต้นทางและปลายทางได้ถูกต้อง
- 7) พิสูจน์ความถูกต้องของฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN และ HSEN สรุปผล พร้อมเสนอแนวทางการพัฒนางานวิจัย
- 8) จัดทำวิทยานิพนธ์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

วิทยานิพนธ์นี้มีประโยชน์ที่คาดว่าจะได้รับ ดังนี้

- 1) สามารถออกแบบฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN ให้สามารถแลกเปลี่ยนข้อมูลระหว่างหน่วยประมวลผลต้นทางและปลายทางได้ถูกต้อง เพื่อเป็นเครือข่ายย่อยสำหรับสร้างระบบ HSEN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) สามารถออกแบบฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN ให้สามารถแลกเปลี่ยนข้อมูลระหว่างหน่วยประมวลผลต้นทางและปลายทางได้ถูกต้อง

3) สามารถนำความรู้และแนวคิดที่ได้จากการศึกษาและออกแบบฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN นี้ไปประยุกต์ใช้ เพื่อออกแบบฟังก์ชันการปรับสวิตช์สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย MIN แบบมีลำดับชั้นอื่นๆ ได้

4) แนวคิดการสร้างเครือข่าย HSEN และฟังก์ชันการปรับสวิตช์สำหรับเครือข่ายดังกล่าวที่เสนอในงานวิจัยนี้ สามารถเป็นแนวทางการพัฒนาการออกแบบสถาปัตยกรรมคอมพิวเตอร์แบบขนานที่สามารถลดเวลาและค่าใช้จ่ายในการขยายระบบให้มีขนาดเพียงพอต่อความต้องการที่เพิ่มขึ้นอย่างรวดเร็วในปัจจุบันได้



บทที่ 2

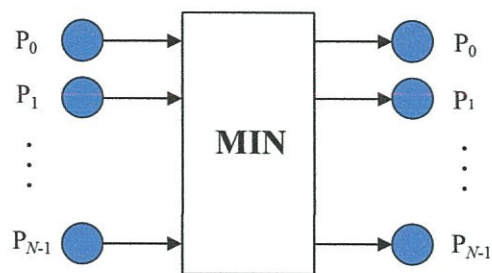
ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีที่เกี่ยวข้องกับสถาปัตยกรรมคอมพิวเตอร์แบบขนาน โดยเน้นไปที่เครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบพลวัต (Dynamic Interconnection Network) หรือเรียกว่า เครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ (Multistage Interconnection Network: MIN) จากนั้นจะกล่าวถึงงานวิจัยที่เกี่ยวข้อง ซึ่งจะเกี่ยวข้องกับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง (All-to-all Personalized Exchange) ระหว่างหน่วยประมวลผลภายในเครือข่าย MIN

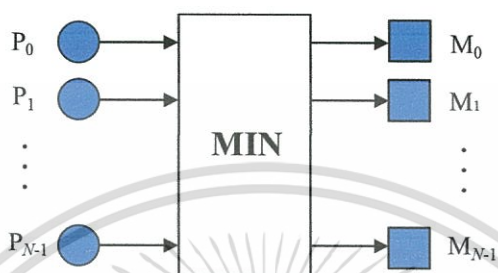
2.1 เครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ

การติดต่อสื่อสารระหว่างหน่วยประมวลผล (Interconnection Network) สามารถแบ่งออกได้เป็น 2 ประเภท คือ ประเภทพลวัต (Static) และประเภทพลวัต (Dynamic) สำหรับระบบเครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบพลวัต (Static Interconnection Network) จะมีการเชื่อมต่อระหว่างหน่วยประมวลผลเป็นแบบจุดต่อจุด ซึ่งถูกออกแบบไว้ตั้งแต่ขั้นตอนการออกแบบโมเดล (Model) ของระบบคอมพิวเตอร์แบบขนาน และจะใช้การส่งข้อความในการติดต่อสื่อสารระหว่างกัน ส่วนเครือข่ายการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบพลวัต หรือ MIN นั้น จะมีการสร้างรูปแบบการติดต่อสื่อสารที่สามารถปรับเปลี่ยนไปตามการติดต่อสื่อสารที่ต้องการขณะประมวลผลได้ ซึ่งการติดต่อสื่อสารประเภทนี้มีสวิตช์ (Switch) ทำหน้าที่เป็นตัวกำหนดรูปแบบการเชื่อมต่อ เพื่อสร้างเส้นทางสำหรับการติดต่อสื่อสารระหว่างหน่วยประมวลผลภายในเครือข่าย การกำหนดรูปแบบการเชื่อมต่อของสวิตช์ (Communication Pattern) นั้น จะเป็นไปตามฟังก์ชันการติดต่อสื่อสารระหว่างต้นทางและปลายทางที่ระบุ โดยโปรแกรมแบบขนาน (Parallel Program) และการเชื่อมต่อภายในสวิตช์ (Switch Setting) ที่เหมาะสมกับเครือข่าย MIN แต่ละประเภท เพื่อให้การสื่อสารข้อความจากต้นทาง (Source: S) ไปยังปลายทาง (Destination: D) ที่ถูกต้องได้

เครือข่าย MIN ถูกนำไปประยุกต์ใช้จริงกับระบบคอมพิวเตอร์แบบขนานชนิดต่างๆ อย่างกว้างขวางในการเชื่อมต่อระหว่างหน่วยประมวลผลกับหน่วยประมวลผล (Processor to Processor) และหน่วยประมวลผลกับหน่วยความจำ (Processor to Memory) ดังแสดงในรูปที่ 2.1 (ก) และ (ข) ตามลำดับ



(ก) การเชื่อมต่อระหว่างหน่วยประมวลผลกับหน่วยประมวลผล



(ข) การเชื่อมต่อระหว่างหน่วยประมวลผลกับหน่วยความจำ

รูปที่ 2.1 การเชื่อมต่อเพื่อการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ

รูปแบบการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ ยังสามารถออกแบบให้แตกต่างกันได้หลายรูปแบบ โดยขึ้นอยู่กับองค์ประกอบของเครือข่าย MIN เช่น การเลือกใช้สวิตช์ขนาดต่างๆ และชนิดของรูปแบบการเชื่อมต่อระหว่างสเตจ (Inter Stage Connection Pattern: ISC) ดังนี้

2.2.1 องค์ประกอบของเครือข่าย MIN

เครือข่าย MIN มีองค์ประกอบที่สำคัญ ได้แก่ หน่วยประมวลผล (Processing Elements: PEs) สวิตช์ สเตจ (Stage) และรูปแบบการเชื่อมต่อระหว่างสเตจ (ISC) ซึ่งมีรายละเอียดดังต่อไปนี้

2.2.1.1 หน่วยประมวลผล

หน่วยประมวลผลภายในเครือข่าย MIN ในงานวิจัยนี้ จะมีจำนวนเท่ากับ $N = 2^n$ ซึ่งเท่ากับจำนวนพอร์ตข้อมูลเข้าและออก (Input/Output Port) โดยที่ $n = 2, 3, 4, \dots$

2.2.1.2 สวิตช์

ขนาดของสวิตช์ (Switch Size) ที่นิยมใช้นั้น จะมีจำนวนพอร์ตข้อมูลเข้าและออกของสวิตช์ที่เท่ากัน คือ $k = 2^m$ เมื่อ $1 \leq m \leq n$ เช่น สวิตช์ขนาด 2×2 , 4×4 หรือ 8×8 เป็นต้น การเชื่อมต่อภายในสวิตช์สามารถปรับเปลี่ยนได้หลายรูปแบบ ได้แก่ การปรับสวิตช์ให้มี

การเชื่อมต่อแบบต่อตรง (Straight) ดังรูปที่ 2.2 (ก) และแบบสลับเปลี่ยน (Exchange) ดังรูปที่ 2.2 (ข) หรือการเชื่อมต่อภายในสวิตช์เพื่อกระจายข้อมูล (Broadcasting) ดังรูปที่ 2.3 ตามลำดับ



รูปที่ 2.2 การเชื่อมต่อภายในสวิตช์ขนาด 2×2



(ก) กระจายลง (Lower Broadcast)

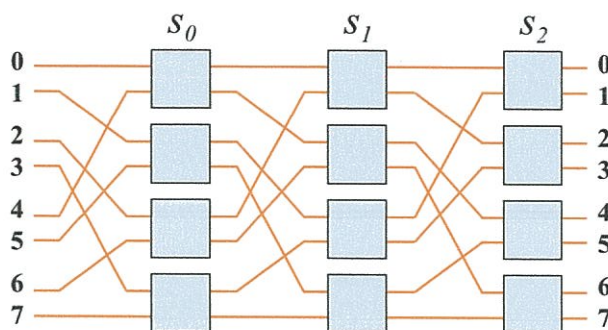
(ข) แบบกระจายขึ้น (Upper Broadcast)

รูปที่ 2.3 การเชื่อมต่อภายในสวิตช์เพื่อกระจายข้อมูลภายในสวิตช์ขนาด 2×2

การเชื่อมต่อภายในสวิตช์สามารถกำหนดการเชื่อมต่อแบบ 1-ต่อ-1 (1-to-1) ได้ทั้งหมด $k!$ เมื่อกำหนดให้สวิตช์มีขนาด $k \times k$ ยกตัวอย่างเช่น สวิตช์ขนาด 4×4 (ข้อมูลเข้าจะมีหมายเลขเป็น 00, 01, 10 และ 11) จะมีรูปแบบที่เป็นได้ทั้งหมด $4! = 24$ แบบ การใช้สวิตช์ขนาดต่างๆ นั้น มีผลต่อจำนวนสเตจภายในเครือข่าย ถ้าขนาดของสวิตช์มีขนาดใหญ่ ก็จะทำให้มีจำนวนสเตจน้อยลง และทำให้การติดต่อสื่อสารระหว่างหน่วยประมวลผลนั้นทำได้รวดเร็วขึ้น แต่ต้องแลกกับราคาของสวิตช์ (Switch Cost) ที่สูงขึ้นตามไปด้วย

2.1.1.3 สเตจ

กำหนดให้สวิตช์มีขนาด $k \times k$ และกำหนดให้ขนาดของข้อมูลเข้าและข้อมูลออก (หน่วยประมวลผลในเครือข่าย) เท่ากับ N จะได้จำนวนสเตจเท่ากับ $\log_k N$ ตัวอย่างเช่น กำหนดให้ $N = 8$ และใช้สวิตช์ขนาด 2×2 จะได้ว่า เครือข่ายนี้จะมีจำนวนสเตจเท่ากับ $\log_2 8 = 3$ สเตจ และแต่ละสเตจจะมีจำนวนสวิตช์เท่ากับ $N/2 = 8/2 = 4$ ดังรูปที่ 2.4 แสดงตัวอย่างเครือข่ายโอเมก้า (Omega Network) หรือชัฟเฟิลเอ็กซ์เชนจ์ (Shuffle-Exchange Network) ขนาด $N = 8$ และใช้สวิตช์ขนาด 2×2



รูปที่ 2.4 เครื่องข่ายฟีลด์เอ็กซ์เชนจ์ ($N = 8$)

2.1.1.4 รูปแบบการเชื่อมต่อระหว่างสเตจ

รูปแบบการเชื่อมต่อระหว่างสเตจ หรือ ISC เป็นรูปแบบการเชื่อมต่อของแต่ละสเตจภายในเครื่องข่าย MIN ซึ่งเป็นการเชื่อมต่อระหว่างข้อมูลเข้ากับสวิตช์ในสเตจแรก สวิตช์กับสวิตช์ หรือ สวิตช์ในสเตจสุดท้ายกับข้อมูลออก โดยเครื่องข่าย MIN แต่ละประเภทจะมี ISC ที่แตกต่างกัน และสามารถแบ่งประเภทเครื่องข่ายตาม ISC ได้ ซึ่งจะกล่าวต่อไปในรายละเอียดของเครื่องข่ายแต่ละประเภท

2.2.2 ประเภทของเครื่องข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ

เครื่องข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ หรือ MIN มีอยู่หลายประเภท แต่ละประเภทถูกแบ่งโดยรูปแบบการเชื่อมต่อระหว่างสเตจหรือ ISC ภายในเครื่องข่าย ซึ่งภายในเครื่องข่ายหนึ่งอาจใช้รูปแบบการเชื่อมต่อระหว่างสเตจเพียงรูปแบบเดียว หรือ บางเครื่องข่ายอาจมีรูปแบบการเชื่อมต่อระหว่างสเตจภายในเครื่องข่ายเดียวกันหลายแบบ เช่น เครื่องข่ายคิวบ์ เป็นต้น ในหัวข้อนี้จะกล่าวถึงตัวอย่างเครื่องข่าย MIN [6] 4 เครื่องข่าย ได้แก่ เครื่องข่ายโอเมก้าหรือฟีลด์เอ็กซ์เชนจ์ เครื่องข่ายฟลิป (Flip Network) เครื่องข่ายคิวบ์ (Cube Network) และเครื่องข่ายเบสไลน์ (Baseline Network)

เครื่องข่าย MIN แต่ละเครื่องข่ายในงานวิจัยนี้ มีรายละเอียดดังนี้

- 1) จำนวนหน่วยประมวลผลภายในเครื่องข่าย กำหนดให้เป็น $N = 2^n$ จะได้บิตหมายเลขหน่วยประมวลผล (id) n บิต คือ $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$
- 2) สวิตช์ที่ใช้ในเครื่องข่ายกำหนดให้เป็นสวิตช์ขนาด 2×2 ($k = 2$)
- 3) จำนวนสเตจภายในเครื่องข่าย หาได้จาก $\log_k N = n$ สเตจ (สเตจ $s_0 s_1 s_2 \dots s_{n-1}$)
- 4) จำนวนสวิตช์ต่อสเตจ หาได้จาก $N/k = N/2$

2.1.2.1 เครื่องข่ายโอเมก้าหรือซฟเฟิลเอ็กซ์เชนจ์

เครือข่ายโอเมก้าหรือเครือข่ายซฟเฟิลเอ็กซ์เชนจ์เป็นเครือข่ายที่มีรูปแบบการเชื่อมต่อระหว่างสแตจ (ISC) เพียงรูปแบบเดียว และมีฟังก์ชันการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเอง (Self-Routing) ดังนี้

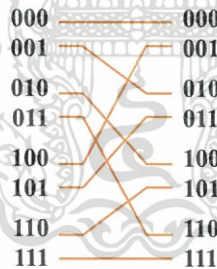
1) รูปแบบการเชื่อมต่อระหว่างสแตจ

ภายในเครือข่ายซฟเฟิลเอ็กซ์เชนจ์มีรูปแบบการเชื่อมต่อระหว่างสแตจ แบ่งออกเป็น 2 ส่วน ดังนี้

- การเชื่อมต่อระหว่างข้อมูลนำเข้ากับสวิตช์ในสแตจ s_0 การเชื่อมต่อระหว่างสวิตช์ในสแตจ s_0 กับ s_1 จนกระทั่งการเชื่อมต่อระหว่างสวิตช์ในสแตจ s_{n-2} กับ s_{n-1} นั้น จะมีรูปแบบการเชื่อมต่อที่เรียกว่า เพอร์เฟกซฟเฟิล (Perfect Shuffle) รูปแบบการเชื่อมต่อนี้สามารถทำได้โดยการนำหมายเลขบิต id หน่วยประมวลผลต้นทาง $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ เลื่อนบิตไปทางซ้ายแบบวนกลับ 1 บิต (1-Bit Circular Left-Shifting) ซึ่งทำให้บิตที่มีนัยสำคัญมากที่สุด (Most Significant Bit) ถูกย้ายตำแหน่งไปอยู่ในตำแหน่งที่มีนัยสำคัญน้อยสุด (Least Significant Bit) จะได้บิตหมายเลขปลายทาง (รูปที่ 2.5) คือ

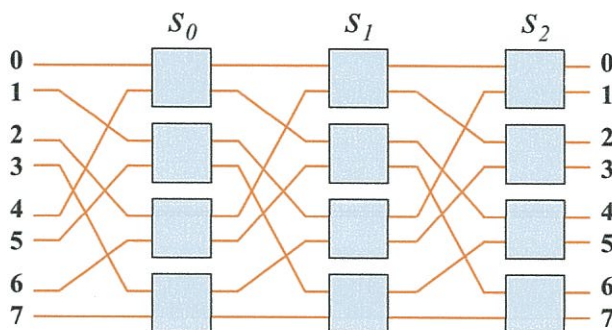
$$b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 \xrightarrow{\text{ISC}} b_{n-2}b_{n-3}b_{n-4} \dots b_1b_0b_{n-1}$$

- การเชื่อมต่อระหว่างสแตจ s_{n-1} กับข้อมูลออก เป็นการต่อตรง



รูปที่ 2.5 รูปแบบการเชื่อมต่อระหว่างสแตจแบบเพอร์เฟกซฟเฟิล สำหรับ $N = 8$

รูปแบบการเชื่อมต่อระหว่างสแตจทั้งหมดภายในเครือข่ายซฟเฟิลเอ็กซ์เชนจ์ ($N = 8$) แสดงได้ดังรูปที่ 2.6



รูปที่ 2.6 รูปแบบการเชื่อมต่อระหว่างสแตจภายในเครือข่ายซัพเฟิลเอ็กซ์เชนจ์ ($N = 8$)

2) การหาเส้นทางจากต้นทางไปปลายทางด้วยตนเอง

การหาเส้นทางด้วยตนเองหรือการสื่อสารข้อมูลประเภทหนึ่งไปหนึ่ง (Point-to-Point Data Routing) นี้ ใช้การดำเนินการฟังก์ชันการเชื่อมต่อภายในสวิตช์ เพื่อให้การสื่อสารข้อมูลจากหน่วยประมวลผลต้นทางสามารถส่งไปถึงปลายทางที่ถูกต้องได้ โดยจะพิจารณาหมายเลขบิต id หน่วยประมวลผลทั้งต้นทางและปลายทางด้วยการดำเนินการ XOR ดังนี้

หากบิต b_i ของต้นทางและบิต d_i ของปลายทางมีค่าต่างกัน กำหนดให้สวิตช์ในสแตจที่สัมพันธ์กับตำแหน่งของบิตที่ i นั้น ถูกปรับเป็นแบบสลับเปลี่ยน ซึ่งจะแทนด้วย “x” และในทางกลับกัน หากคู่อันดับที่ i ไม่ต่างกัน สวิตช์ในสแตจดังกล่าวนั้นจะถูกปรับเป็นแบบต่อตรง ซึ่งแทนด้วย “-”

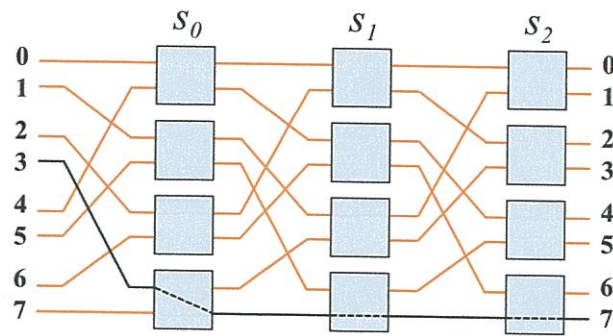
ฟังก์ชันการเชื่อมต่อภายในสวิตช์เพื่อการสื่อสารข้อมูลสำหรับเครือข่ายซัพเฟิลเอ็กซ์เชนจ์ สามารถทำได้ดังตัวอย่าง 2.1 ดังนี้

ตัวอย่าง 2.1 การสื่อสารข้อมูลภายในเครือข่ายซัพเฟิลเอ็กซ์เชนจ์ ($N = 8$) ระหว่างหน่วยประมวลผลต้นทาง $id: 3$ ไปหน่วยประมวลผลปลายทาง $id: 7$

ต้นทาง	3: 0 1 1
ปลายทาง	7: <u>1 1 1</u>
	<u>x - -</u>

จะได้รูปแบบการเชื่อมต่อภายในสวิตช์ในสแตจ s_0, s_1, s_2 คือ “x - -” ดังรูปที่

2.7



รูปที่ 2.7 การสื่อสารข้อมูลระหว่างหน่วยประมวลผล 3 ไป 7 สำหรับ
เครือข่ายซฟเฟิลเอ็กซ์เชนจ์ ($N = 8$)

2.1.2.2 เครือข่ายฟลิป

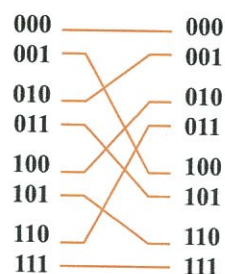
เครือข่ายฟลิปเป็นเครือข่ายที่มีรูปแบบการเชื่อมต่อระหว่างสเตจ (ISC) เพียงรูปแบบเดียว และมีฟังก์ชันสำหรับการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองดังนี้

1) รูปแบบการเชื่อมต่อระหว่างสเตจ

ภายในเครือข่ายฟลิปจะใช้รูปแบบการเชื่อมต่อระหว่างสเตจที่ตรงข้ามกันกับเครือข่ายซฟเฟิลเอ็กซ์เชนจ์ แต่สามารถแบ่งออกเป็น 2 ส่วนได้เช่นเดียวกัน คือ

- การเชื่อมต่อระหว่างข้อมูลเข้ากับสเตจ s_0 เป็นการต่อตรง
- การเชื่อมต่อระหว่างสวิตช์ในสเตจ s_0 กับ s_1 , การเชื่อมต่อระหว่างสวิตช์ในสเตจ s_1 กับ s_2 จนกระทั่งการเชื่อมต่อระหว่างสวิตช์ในสเตจ s_{n-1} กับข้อมูลออกนั้น จะมีรูปแบบการเชื่อมต่อที่เรียกว่า อินเวอร์ตเพอร์เฟกซ์ฟลิป (Inverted Perfect Shuffle) รูปแบบการเชื่อมต่อนี้สามารถทำได้โดยการนำหมายเลขบิต id หน่วยประมวลผลต้นทาง $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ เลื่อนบิตไปทางขวาแบบวนกลับ 1 บิต (1-Bit Circular Right-Shifting) ซึ่งทำให้บิตที่มีนัยสำคัญน้อยสุดถูกย้ายตำแหน่งไปอยู่ในตำแหน่งที่มีนัยสำคัญมากที่สุด จะได้บิตหมายเลขปลายทาง (รูปที่ 2.8) คือ

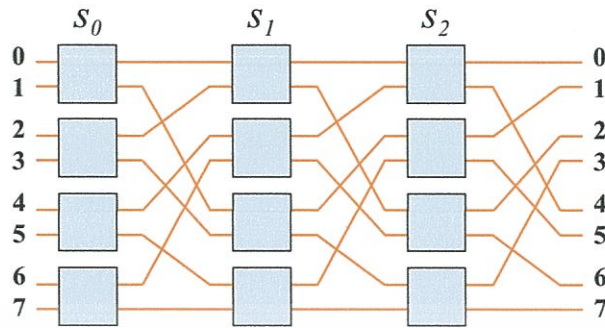
$$b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 \xrightarrow{\text{ISC}} b_0b_{n-1}b_{n-2} \dots b_3b_2b_1$$



รูปที่ 2.8 รูปแบบการเชื่อมต่อระหว่างสเตจแบบอินเวอร์ตเพอร์เฟกซ์ฟลิป สำหรับ $N = 8$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบการเชื่อมต่อระหว่างสแตจทั้งหมดภายในเครือข่ายฟลิป ($N = 8$) แสดงไว้ดังรูปที่ 2.9



รูปที่ 2.9 รูปแบบการเชื่อมต่อระหว่างสแตจภายในเครือข่ายฟลิป ($N = 8$)

2) การหาเส้นทางจากต้นทางไปปลายทางด้วยตนเอง

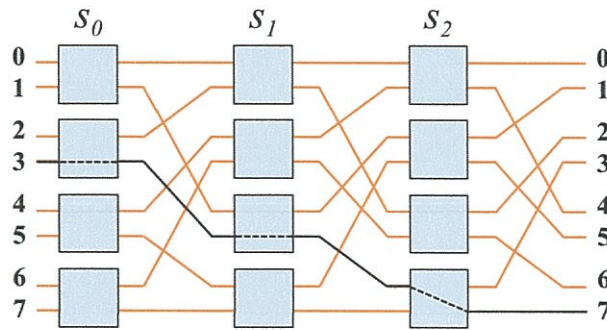
การหาเส้นทางด้วยตนเองเพื่อการสื่อสารข้อมูลของเครือข่ายนี้ สามารถพิจารณาหมายเลขบิต id หน่วยประมวลผลทั้งต้นทางและปลายทางด้วยการดำเนินการ XOR ได้เช่นเดียวกับเครือข่ายฟีลด์เอ็กซ์เชนจ์

ฟังก์ชันการเชื่อมต่อภายในสวิตช์เพื่อการสื่อสารข้อมูลสำหรับเครือข่ายฟลิปสามารถทำได้ดังตัวอย่าง 2.2 ซึ่งมีการเชื่อมต่อภายในสวิตช์แต่ละสแตจกลับตำแหน่งเป็นตรงกันข้ามกับเครือข่ายฟีลด์เอ็กซ์เชนจ์ ดังนี้

ตัวอย่าง 2.2 การสื่อสารข้อมูลภายในเครือข่ายฟลิป ($N = 8$) ระหว่างหน่วยประมวลผลต้นทาง $id: 3$ ไปหน่วยประมวลผลปลายทาง $id: 7$

ต้นทาง	3: 0 1 1
ปลายทาง	7: 1 1 1
	<u>x - -</u>

ข้อสังเกต เมื่อได้รูปแบบการเชื่อมต่อภายในสวิตช์จากการเปรียบเทียบบิตหมายเลขต้นทางกับปลายทางแล้ว จะต้องกลับรูปแบบที่ได้เป็นตรงกันข้ามทุกตำแหน่ง ดังนั้นจะได้รูปแบบการเชื่อมต่อภายในสวิตช์สำหรับสแตจ s_0, s_1, s_2 คือ “- - x” ดังรูปที่ 2.10



รูปที่ 2.10 การสื่อสารข้อมูลระหว่างหน่วยประมวลผล 3 ไป 7 สำหรับเครือข่ายฟิลิป ($N=8$)

2.1.2.3 เครือข่ายคิวบ์

เครือข่ายคิวบ์เป็นเครือข่ายที่มีรูปแบบการติดต่อสื่อสารระหว่างหน่วยประมวลผลในแต่ละสเตจหลายรูปแบบ แต่มีฟังก์ชันการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเอง เช่นเดียวกับเครือข่ายฟิลิปดังนี้

1) รูปแบบการเชื่อมต่อระหว่างสเตจ

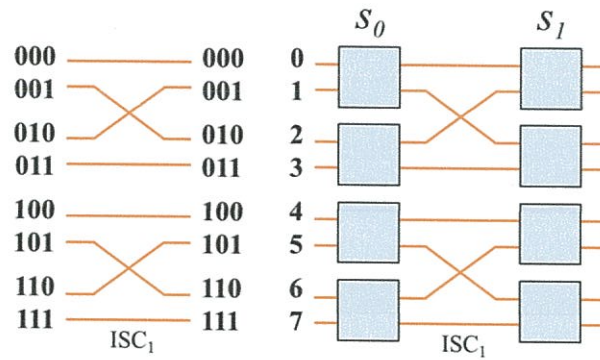
ภายในเครือข่ายคิวบ์มีรูปแบบการเชื่อมต่อระหว่างสเตจ หรือ ISC ที่แตกต่างกันสำหรับแต่ละการเชื่อมต่อระหว่างสเตจ ซึ่งสามารถแบ่งได้ดังนี้

- การเชื่อมต่อระหว่างข้อมูลเข้ากับสเตจ s_0 เป็นการต่อตรง
- การเชื่อมต่อระหว่างสเตจ s_{i-1} กับ s_i ($i = 1, 2, 3, \dots, n-1$) ซึ่งจะเรียกว่า ISC_i เป็นการสลับตำแหน่งบิตหมายเลขหน่วยประมวลผลต้นทาง $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ ระหว่างบิตที่มีนัยสำคัญน้อยสุด (b_0) กับบิต b_i (รูปที่ 2.11 2.12 และ 2.13) คือ

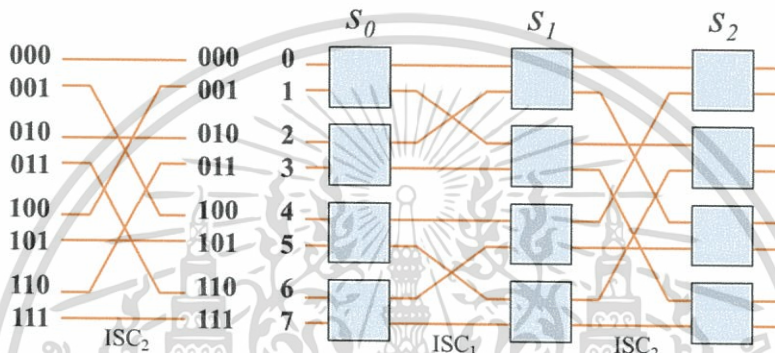
$$\begin{aligned}
 & b_{n-1}b_{n-2}b_{n-3} \dots b_2(b_1b_0) \xrightarrow{ISC_1} b_{n-1}b_{n-2} \dots b_2(b_0b_1) \\
 & b_{n-1}b_{n-2}b_{n-3} \dots b_3(b_2b_1b_0) \xrightarrow{ISC_2} b_{n-1}b_{n-2} \dots b_3(b_0b_1b_2) \\
 & \dots \\
 & (b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0) \xrightarrow{ISC_{n-1}} (b_0b_{n-2}b_{n-3} \dots b_1b_{n-1})
 \end{aligned}$$

- การเชื่อมต่อระหว่างสเตจ s_{n-1} กับข้อมูลออก ซึ่งจะเรียกว่า ISC_n เป็นรูปแบบการเชื่อมต่อที่เรียกว่า อินเวิร์ตเพอร์เฟคซัพเฟิล เช่นเดียวกับเครือข่ายฟิลิป สามารถทำได้โดยการนำบิตหมายเลขหน่วยประมวลผลต้นทาง $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ เลื่อนบิตไปทางขวาแบบวนกลับ 1 บิต จะได้บิตหมายเลขปลายทาง (รูปที่ 2.14) คือ

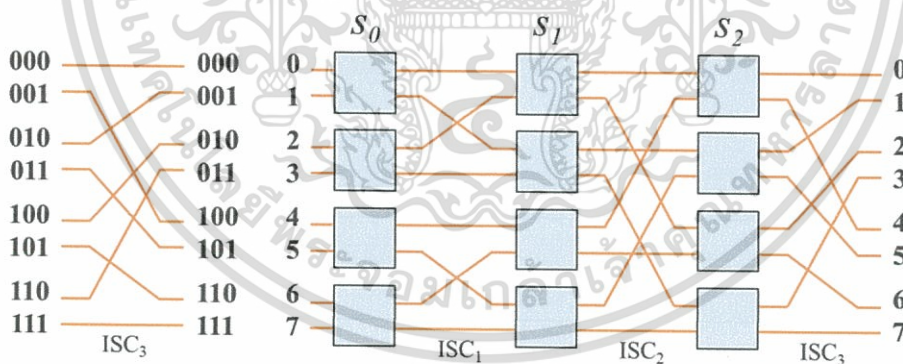
$$b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 \xrightarrow{ISC_n} b_0b_{n-1}b_{n-2} \dots b_3b_2b_1$$



รูปที่ 2.11 รูปแบบการเชื่อมต่อระหว่างสแตจ ISC_1 ภายในเครือข่ายคิวิบ์ ($N=8$)

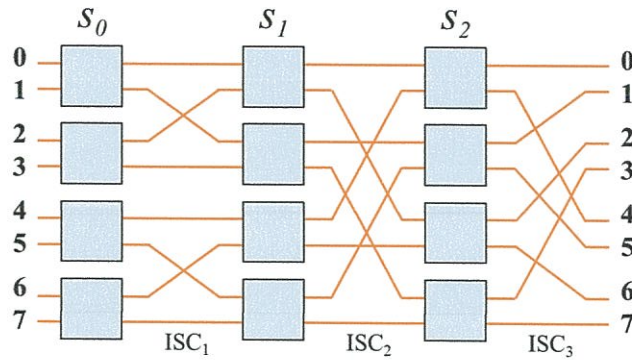


รูปที่ 2.12 รูปแบบการเชื่อมต่อระหว่างสแตจ ISC_2 ภายในเครือข่ายคิวิบ์ ($N=8$)



รูปที่ 2.13 รูปแบบการเชื่อมต่อระหว่างสแตจ ISC_3 ภายในเครือข่ายคิวิบ์ ($N=8$)

รูปแบบการเชื่อมต่อระหว่างสแตจทั้งหมดภายในเครือข่ายคิวิบ์ ($N=8$) แสดง
ได้ดังรูปที่ 2.14



รูปที่ 2.14 รูปแบบการเชื่อมต่อระหว่างสเตจภายในเครือข่ายคิวบ์ ($N=8$)

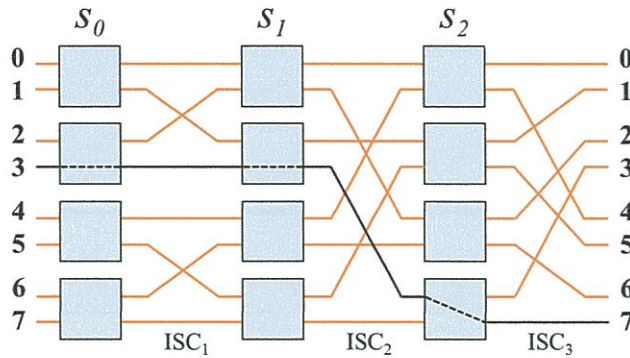
2) การหาเส้นทางจากต้นทางไปปลายทางด้วยตนเอง

การสื่อสารข้อมูลนี้ สามารถทำได้โดยการดำเนินการ XOR ระหว่างหมายเลขบิต id หน่วยประมวลผลทั้งต้นทางและปลายทางเช่นเดียวกับเครือข่ายซัพเฟิลเอ็กซ์เชนจ์และฟลิป และสำหรับฟังก์ชันการเชื่อมต่อภายในสวิตช์เพื่อการสื่อสารข้อมูลสำหรับเครือข่ายคิวบ์ สามารถทำได้ดังตัวอย่าง 2.3 ซึ่งมีการเชื่อมต่อภายในสวิตช์เช่นเดียวกับเครือข่ายฟลิป (มีการเชื่อมต่อภายในสวิตช์แต่ละสเตจกลับตำแหน่งเป็นตรงกันข้ามกับเครือข่ายซัพเฟิลเอ็กซ์เชนจ์) ดังนี้

ตัวอย่าง 2.3 การสื่อสารข้อมูลภายในเครือข่ายคิวบ์ ($N=8$) ระหว่างหน่วยประมวลผลต้นทาง $id: 3$ ไปหน่วยประมวลผลปลายทาง $id: 7$

ต้นทาง	3: 0 1 1
ปลายทาง	7: 1 1 1
	<u> </u> x - -

ข้อสังเกต เมื่อได้รูปแบบการเชื่อมต่อภายในสวิตช์จากการเปรียบเทียบบิตหมายเลขต้นทางกับปลายทางแล้ว จะต้องกลับรูปแบบที่ได้เป็นตรงกันข้ามทุกตำแหน่งเหมือนกับการเชื่อมต่อภายในสวิตช์สำหรับเครือข่ายฟลิป ดังนั้นจะได้รูปแบบการเชื่อมต่อภายในสวิตช์สำหรับสเตจ s_0, s_1, s_2 คือ “- - x” ดังรูปที่ 2.15



รูปที่ 2.15 การสื่อสารข้อมูลระหว่างหน่วยประมวลผล 3 ไป 7 สำหรับเครือข่ายคิวบ์ ($N = 8$)

2.1.2.4 เครือข่ายเบสไลน์

เครือข่ายเบสไลน์เป็นเครือข่ายที่มีรูปแบบการเชื่อมต่อระหว่างสเตจ (ISC) แตกต่างกันในแต่ละสเตจ และมีฟังก์ชันสำหรับการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองดังนี้

1) รูปแบบการเชื่อมต่อระหว่างสเตจ

ภายในเครือข่ายเบสไลน์จะใช้ ISC ที่แตกต่างกันสำหรับแต่ละการเชื่อมต่อระหว่างสเตจ ซึ่งสามารถแบ่งได้ดังนี้

- การเชื่อมต่อระหว่างข้อมูลเข้ากับสเตจ s_0 และสเตจ s_{n-1} กับข้อมูลออกเป็นการต่อตรง
- การเชื่อมต่อระหว่างสเตจ s_{i-1} กับ s_i ($i = 1, 2, 3, \dots, n-1$) ซึ่งจะเรียกว่า ISC_i เป็นการเลื่อนตำแหน่งบิตหมายเลขหน่วยประมวลผลต้นทาง $(b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0)$ $n - i + 1$ บิตแรกไปทางขวาแบบวนกลับ 1 บิต ดังรูปที่ 2.11 2.12 และ 2.13 คือ

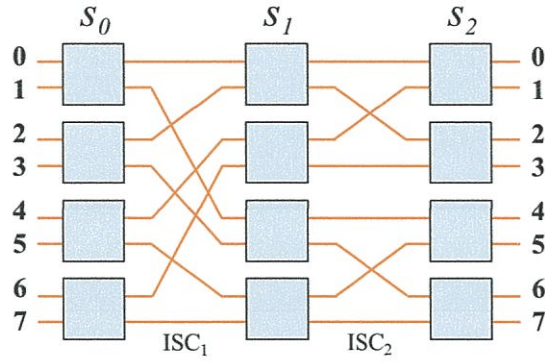
$$(b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0) \xrightarrow{ISC_1} (b_0b_{n-1}b_{n-2} \dots b_3b_2b_1)$$

$$b_{n-1}(b_{n-2}b_{n-3} \dots b_2b_1b_0) \xrightarrow{ISC_2} b_{n-1}(b_0b_{n-2} \dots b_3b_2b_1)$$

...

$$b_{n-1}b_{n-2}b_{n-3} \dots b_2(b_1b_0) \xrightarrow{ISC_{n-1}} b_{n-1}b_{n-2}b_{n-3} \dots b_2(b_0b_1)$$

รูปแบบการเชื่อมต่อระหว่างสเตจทั้งหมดภายในเครือข่ายเบสไลน์ ($N = 8$) แสดงได้ดังรูปที่ 2.16



รูปที่ 2.16 รูปแบบการเชื่อมต่อระหว่างสเตจภายในเครือข่ายเบสไลน์ ($N=8$)

2) การหาเส้นทางจากต้นทางไปปลายทางด้วยตนเอง

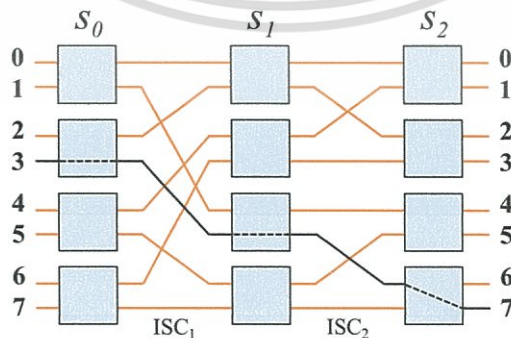
การหาเส้นทางด้วยตนเองเพื่อการสื่อสารข้อมูลของเครือข่ายนี้ สามารถพิจารณาหมายเลขบิต id หน่วยประมวลผลทั้งต้นทางและปลายทางด้วยการดำเนินการ XOR แต่สามารถทำได้เพียงบางคู่ต้นทางกับปลายทางเท่านั้น ซึ่งทำได้เช่นเดียวกับเครือข่ายคิวบ์ ดังตัวอย่าง 2.4 ดังนี้

ตัวอย่าง 2.4 การสื่อสารข้อมูลภายในเครือข่ายเบสไลน์ ($N=8$) ระหว่างหน่วยประมวลผลต้นทาง $id: 3$ ไปหน่วยประมวลผลปลายทาง $id: 7$

ต้นทาง	3: 0 1 1
ปลายทาง	7: 1 1 1
	<u> </u>
	x - -

จะได้รูปแบบการเชื่อมต่อภายในสวิตช์สำหรับสเตจ s_0, s_1, s_2 คือ “- - x” ดังรูป

ที่ 2.17



รูปที่ 2.17 การสื่อสารข้อมูลระหว่างหน่วยประมวลผล 3 ไป 7 สำหรับเครือข่ายเบสไลน์ ($N=8$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถึงแม้ว่าการหาเส้นทางด้วยตนเองบนเครือข่ายเบสไลน์จะทำได้เพียงบางคู่ต้นทาง-ปลายทางเท่านั้น แต่สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายนี้ สามารถทำได้ซึ่ง [1] ได้เสนอไว้ และจะกล่าวในหัวข้อถัดไป

2.2 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง

การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงระหว่างหน่วยประมวลผลภายในเครือข่าย สามารถพิจารณาว่าเป็นการสื่อสารข้อมูลประเภทหนึ่งไปหนึ่งแบบขนาน (Parallel Point-to-Point Data Routing) ระหว่างคู่ต้นทาง (S) - ปลายทาง (D) ทั้งหมดภายในเครือข่ายได้ การสื่อสารนี้ต้องอาศัยการเชื่อมต่อภายในสวิตช์สำหรับการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเอง การเชื่อมต่อภายในสวิตช์ดังกล่าวสามารถหาได้จากการดำเนินการ XOR ระหว่างหมายเลขบิต id หน่วยประมวลผลทั้งต้นทางและปลายทางดังตัวอย่างที่ 2.1 - 2.4 สำหรับเครือข่ายฟิสิกส์เชนจ์ฟิลิป คิวบ์ และเบสไลน์ตามลำดับ สำหรับตัวอย่างที่จะกล่าวต่อไปในหัวข้อนี้ จะเป็นตัวอย่างการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย MIN ขนาด $N = 8$

2.2.1 สี่เหลี่ยมละติน (Latin Square)

สี่เหลี่ยมละติน [1] สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย MIN คือ ตารางที่มีขนาด $N \times N$ โดยที่แต่ละแถวและแต่ละหลักจะมีตัวเลขที่แตกต่างกันคือ $0, 1, 2, \dots, N-1$ ซึ่งเป็นการเรียงสับเปลี่ยน (Permutation: ρ) ของการติดต่อสื่อสารระหว่างต้นทางและปลายทาง แต่ละแถวหมายถึงปลายทางของการสื่อสาร เช่น

$$L_1 = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 0 & 3 & 2 & 5 & 4 & 7 & 6 \\ 2 & 3 & 0 & 1 & 6 & 7 & 4 & 5 \\ 3 & 2 & 1 & 0 & 7 & 6 & 5 & 4 \\ 4 & 5 & 6 & 7 & 0 & 1 & 2 & 3 \\ 5 & 4 & 7 & 6 & 1 & 0 & 3 & 2 \\ 6 & 7 & 4 & 5 & 2 & 3 & 0 & 4 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$

$$L_2 = \begin{bmatrix} 0 & 4 & 2 & 6 & 1 & 5 & 3 & 7 \\ 1 & 5 & 3 & 7 & 0 & 4 & 2 & 6 \\ 2 & 6 & 0 & 4 & 3 & 7 & 1 & 5 \\ 3 & 7 & 1 & 5 & 2 & 6 & 0 & 4 \\ 4 & 0 & 6 & 2 & 5 & 1 & 7 & 3 \\ 5 & 1 & 7 & 3 & 4 & 0 & 6 & 2 \\ 6 & 2 & 4 & 0 & 7 & 3 & 5 & 1 \\ 7 & 3 & 5 & 1 & 6 & 2 & 4 & 0 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างความหมายของการเรียงสับเปลี่ยน $\rho = (1\ 0\ 3\ 2\ 5\ 4\ 7\ 6)$ ในแถวที่ 2 ของสี่เหลี่ยมละติน L_1 ข้างต้น คือ

ต้นทางหมายเลข 0 จะจับคู่ส่งข้อมูลไปยังปลายทางหมายเลข 1
 ต้นทางหมายเลข 1 จะจับคู่ส่งข้อมูลไปยังปลายทางหมายเลข 0
 ต้นทางหมายเลข 2 จะจับคู่ส่งข้อมูลไปยังปลายทางหมายเลข 3
 ต้นทางหมายเลข 3 จะจับคู่ส่งข้อมูลไปยังปลายทางหมายเลข 2
 ต้นทางหมายเลข 4 จะจับคู่ส่งข้อมูลไปยังปลายทางหมายเลข 5
 ต้นทางหมายเลข 5 จะจับคู่ส่งข้อมูลไปยังปลายทางหมายเลข 4
 ต้นทางหมายเลข 6 จะจับคู่ส่งข้อมูลไปยังปลายทางหมายเลข 7
 ต้นทางหมายเลข 7 จะจับคู่ส่งข้อมูลไปยังปลายทางหมายเลข 6

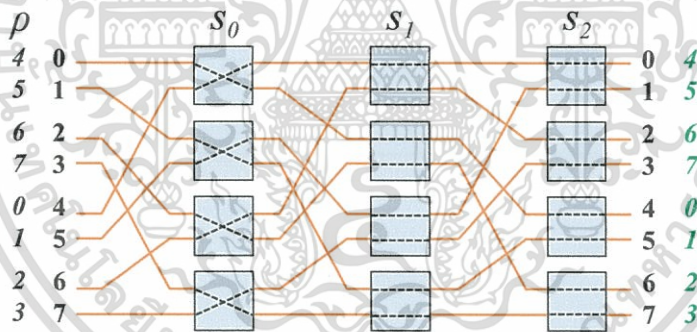
2.2.2 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย MIN

ในหัวข้อนี้จะแสดงการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย MIN ขนาด $N = 8$ ทั้ง 4 เครือข่าย คือ เครือข่ายซฟเฟิลเอ็กซ์เชนจ์ ฟลิป คิวบ์ และเบสไลน์ ดังตัวอย่างที่ 2.5 - 2.8 ตามลำดับ โดยที่ตัวอย่างที่ 2.5 - 2.7 จะนำการเรียงสับเปลี่ยนจากสี่เหลี่ยมละติน L_1 ในหัวข้อ 2.2.1 มาประยุกต์ใช้ โดยจะแสดงตัวอย่างการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองโดยใช้การเรียงสับเปลี่ยนจากสี่เหลี่ยมละติน L_1 รูปแบบหนึ่งสำหรับการสื่อสาร 1 รอบก่อนคือ $\rho = (4\ 5\ 6\ 7\ 0\ 1\ 2\ 3)$ เพื่อแสดงการนำไปปรับการเชื่อมต่อภายในสวิตช์บนเครือข่ายเหล่านี้ขณะสื่อสารข้อมูลจากต้นทาง (S) ไปปลายทาง (D) จากนั้นจึงแสดงการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายทั้งสาม ส่วนตัวอย่างที่ 2.8 เป็นการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายเบสไลน์ด้วยสี่เหลี่ยมละติน L_2

ตัวอย่างที่ 2.5 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายซฟเฟิลเอ็กซ์เชนจ์ เครือข่ายนี้จะนำผลลัพธ์ที่ได้จากการดำเนินการ $S \text{ XOR } D (r_{n-1}r_{n-2}r_{n-3} \dots r_2r_1r_0)$ ดังตารางที่ 2.1 ไปปรับการเชื่อมต่อภายในสวิตช์ในสเตจ $s_0, s_1, s_2, \dots, s_{n-3}, s_{n-2}, s_{n-1}$ ดังรูปที่ 2.18 และการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายนี้ดำเนินการดังรูปที่ 2.19

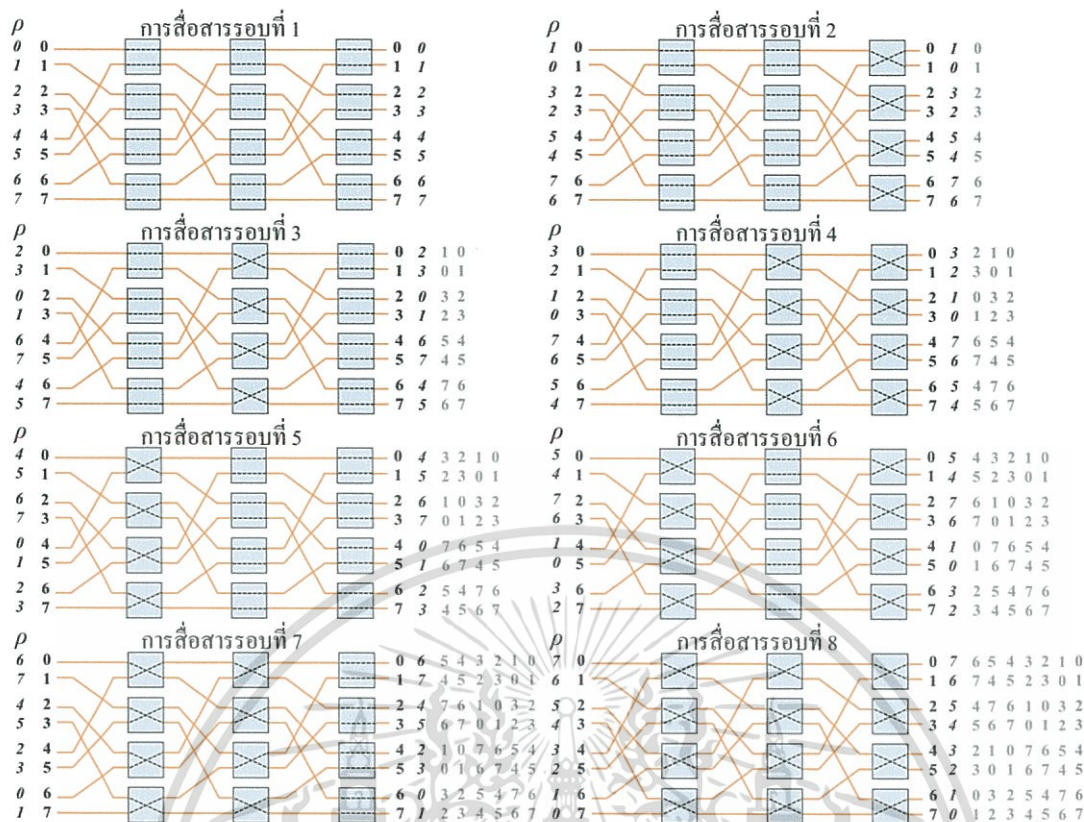
ตารางที่ 2.1 การเชื่อมต่อภายในสวิตช์บนเครือข่าย MIN ที่ดำเนินการสำเร็จด้วย $p = (4\ 5\ 6\ 7\ 0\ 1\ 2\ 3)$

ต้นทาง (S)		ปลายทาง (D)		$S \text{ XOR } D$	การเชื่อมต่อภายในสวิตช์
ฐานสิบ	ฐานสอง	ฐานสิบ	ฐานสอง		
0	000	4	100	100	x--
1	001	5	101	100	x--
2	010	6	110	100	x--
3	011	7	111	100	x--
4	100	0	000	100	x--
5	101	1	001	100	x--
6	110	2	010	100	x--
7	111	3	011	100	x--



รูปที่ 2.18 การเชื่อมต่อภายในสวิตช์บนเครือข่ายซฟเฟิลเอ็กซ์เชนจ์ ($N = 8$) ที่ดำเนินการสำเร็จด้วย $p = (4\ 5\ 6\ 7\ 0\ 1\ 2\ 3)$

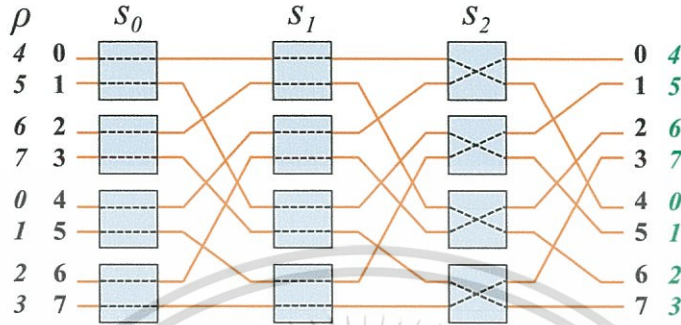
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



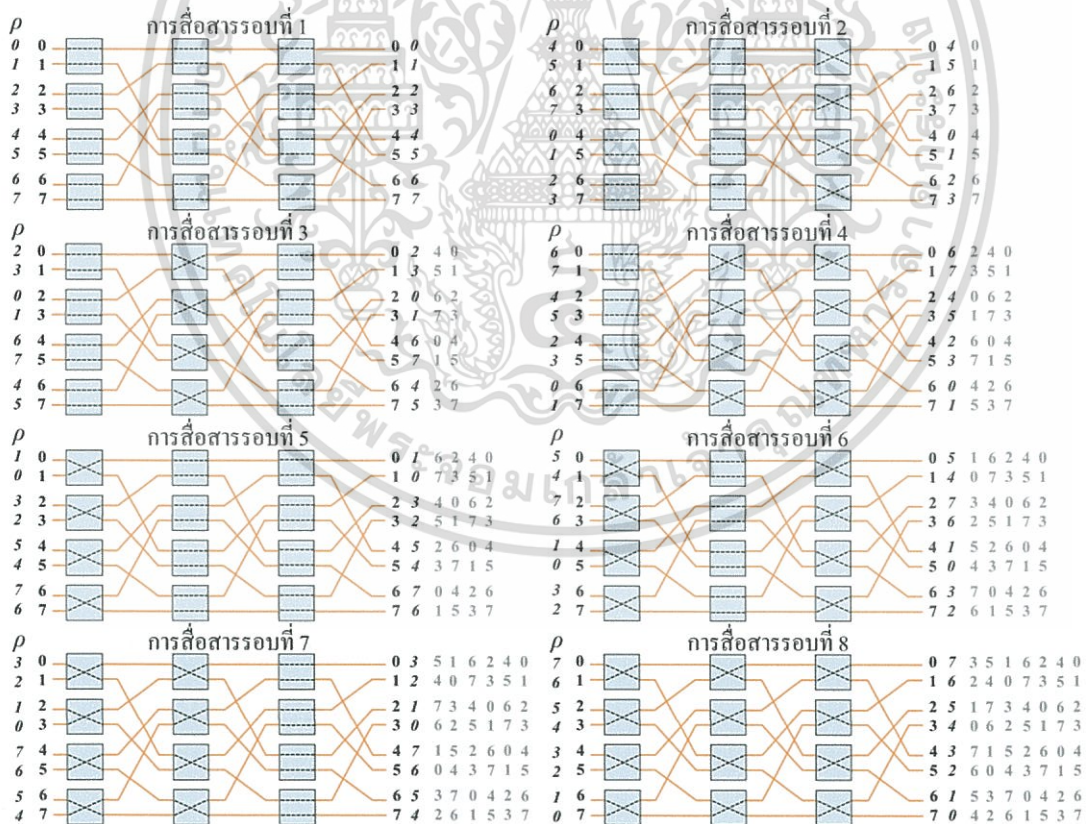
รูปที่ 2.19 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย วัฟเฟิลเอ็กซ์เชนจ์ ($N=8$) ที่ดำเนินการสำเร็จทั้ง N รอบ

จากรูปที่ 2.19 แต่ละรอบการสื่อสารนั้นหน่วยประมวลผลต้นทางจะส่งข้อมูลไปยังหน่วยประมวลผลปลายทางตามสี่เหลี่ยมละติน L_1 สำหรับการสื่อสารรอบที่ 1 อธิบายได้ว่าหน่วยประมวลผล $id \#0$ ส่งข้อความไปหน่วยประมวลผล $id \#0$, หน่วยประมวลผล $id \#1$ ส่งข้อความไปหน่วยประมวลผล $id \#1$, ..., หน่วยประมวลผล $id \#7$ ส่งข้อความไปหน่วยประมวลผล $id \#7$ ส่วนในการสื่อสารรอบที่ 2 นั้น หน่วยประมวลผล $id \#0$ ส่งข้อความไปหน่วยประมวลผล $id \#1$, หน่วยประมวลผล $id \#1$ ส่งข้อความไปหน่วยประมวลผล $id \#0$, ..., หน่วยประมวลผล $id \#7$ ส่งข้อความไปหน่วยประมวลผล $id \#6$ (ตัวเลขสีเทาด้านขวาของการสื่อสารในแต่ละรอบ หมายถึง หมายเลข id ของหน่วยประมวลผลที่ได้ส่งข้อความมายังหน่วยประมวลผลนั้นๆ แล้ว) และในรอบการสื่อสารอื่นๆ ก็จะมีการดำเนินการลักษณะเดียวกันนี้ เมื่อครบ N รอบ (การสื่อสารรอบที่ 8) ทุกๆ หน่วยประมวลผลจะได้รับข้อมูลจากหน่วยประมวลผลภายในเครือข่ายครบทุกหน่วยประมวลผล

ตัวอย่างที่ 2.6 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายฟิลิป เครือข่ายนี้จะนำผลลัพธ์ที่ได้จากการดำเนินการ $S \text{ XOR } D (r_{n-1}r_{n-2}r_{n-3} \dots r_2r_1r_0)$ จากตารางที่ 2.1 ไปปรับการเชื่อมต่อภายในสวิตช์ในสเตจ $s_{n-1}, s_{n-2}, s_{n-3}, \dots, s_2, s_1, s_0$ ดังรูปที่ 2.20 และการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายนี้ สามารถดำเนินการดังรูปที่ 2.21



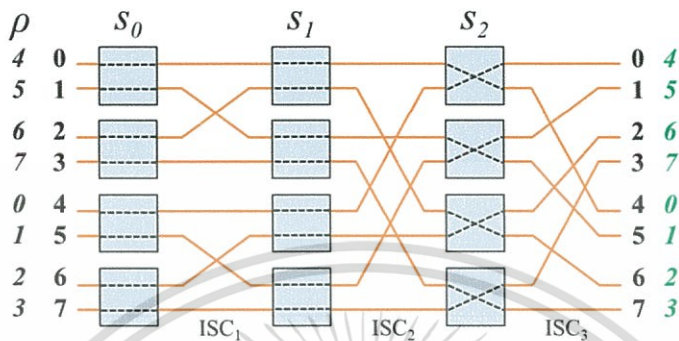
รูปที่ 2.20 การเชื่อมต่อภายในสวิตช์บนเครือข่ายฟิลิป ($N = 8$) ที่ดำเนินการสำเร็จด้วย $\rho = (4 \ 5 \ 6 \ 7 \ 0 \ 1 \ 2 \ 3)$



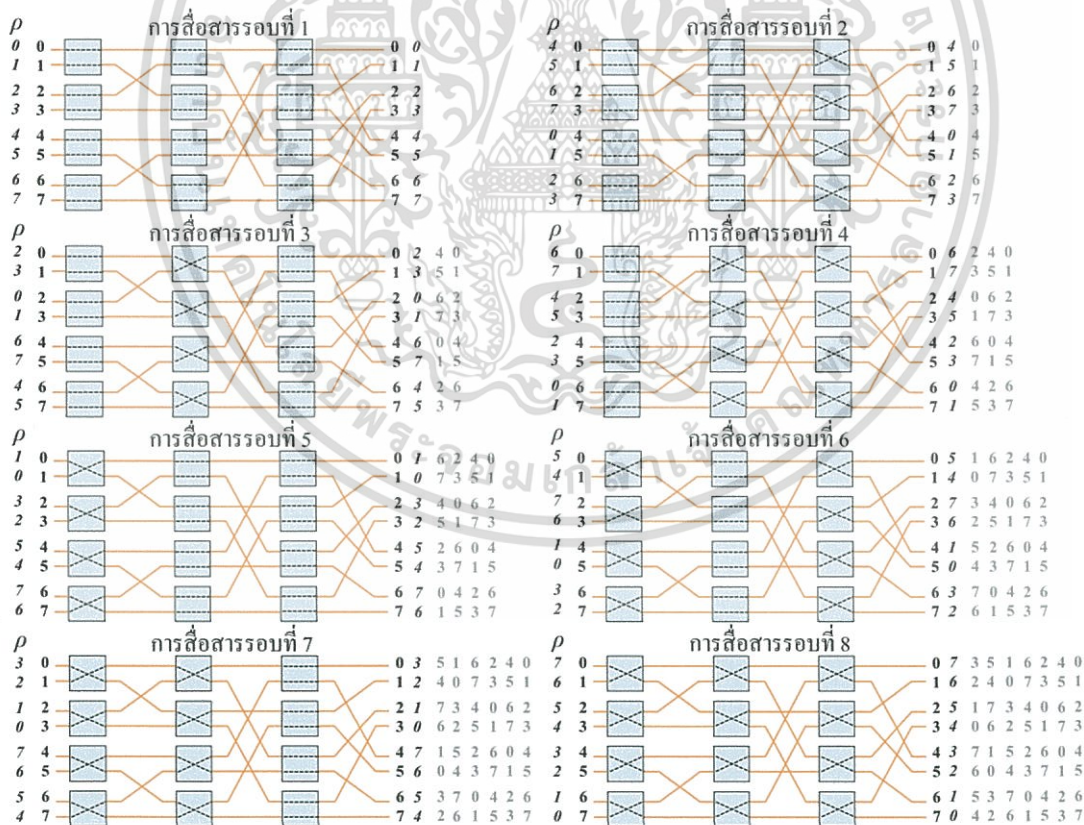
รูปที่ 2.21 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายฟิลิป ($N = 8$) ที่ดำเนินการสำเร็จทั้ง N รอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2.7 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายคิวบ์ เครือข่ายนี้จะนำผลลัพธ์ที่ได้จากการดำเนินการ $S \text{ XOR } D$ จากตารางที่ 2.1 ไปปรับการเชื่อมต่อภายในสวิตช์ในสเตจ $s_{n-1}, s_{n-2}, s_{n-3}, \dots, s_2, s_1, s_0$ เช่นเดียวกับเครือข่ายฟลิป ดังรูปที่ 2.22 และการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายนี้ดำเนินการดังรูปที่ 2.23



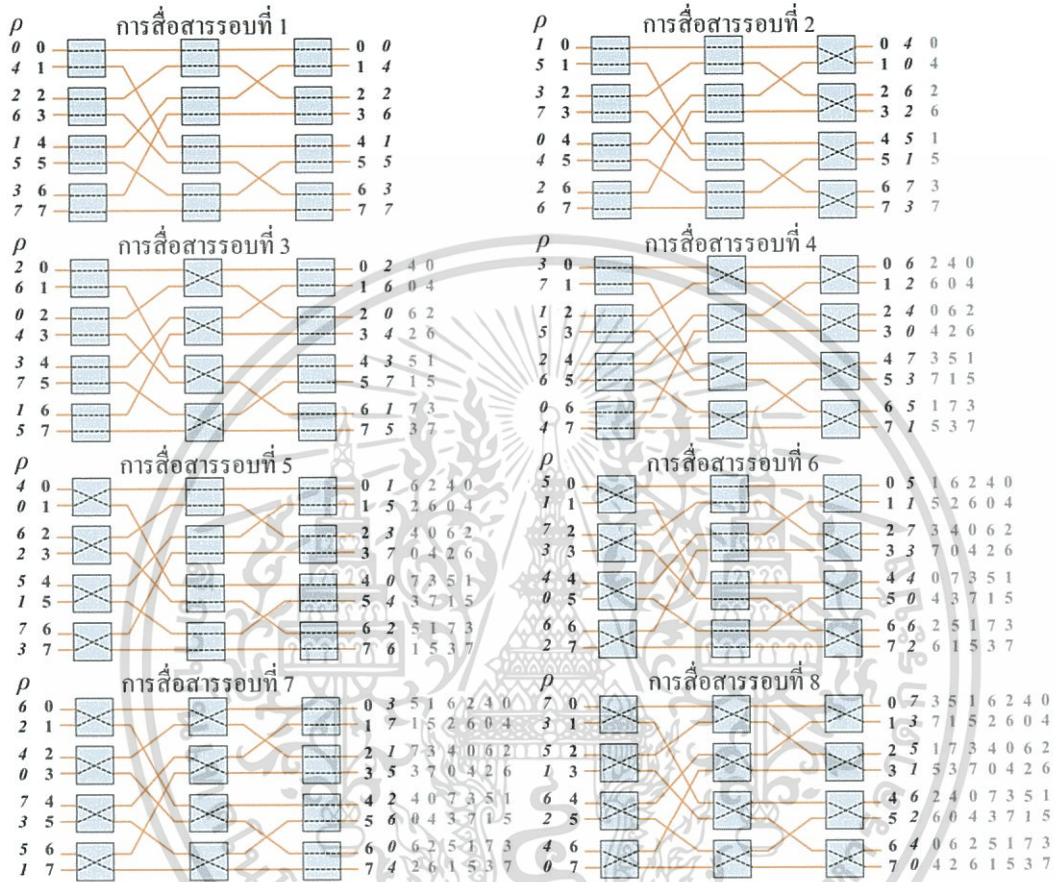
รูปที่ 2.22 การเชื่อมต่อภายในสวิตช์บนเครือข่ายคิวบ์ ($N = 8$) ที่ดำเนินการสำเร็จด้วย $\rho = (4\ 5\ 6\ 7\ 0\ 1\ 2\ 3)$



รูปที่ 2.23 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายคิวบ์ ($N = 8$) ที่ดำเนินการสำเร็จทั้ง N รอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2.8 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายเบสไลน์
 เครือข่ายนี้ไม่สามารถดำเนินการหาเส้นทางจากต้นทาง ไปปลายทางด้วยตนเองได้ ดังนั้นจะแสดง
 เฉพาะการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายนี้ด้วยการเรียง
 สับเปลี่ยนจากสี่เหลี่ยมละติน L_2 ซึ่งดำเนินการได้ดังรูปที่ 2.24



รูปที่ 2.24 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายเบสไลน์
 ($N=8$) ที่ดำเนินการสำเร็จทั้ง N รอบ

บทที่ 3

การปรับสวิตช์บนเครือข่ายซฟเฟิลเอ็กซ์เชนจ์แบบมีลำดับชั้น สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมด

แบบเจาะจง

งานวิจัยนี้เสนอการปรับสวิตช์ (Switch Adjusting) บนเครือข่ายซฟเฟิลเอ็กซ์เชนจ์แบบมีลำดับชั้น (Hierarchical Shuffle-Exchange Network: HSEN) สำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง โดยที่โครงสร้างของเครือข่ายแบบมีลำดับชั้นดังกล่าวเป็นเครือข่ายที่มีโครงสร้างพิเศษขนาด $N = 2^n$ ประกอบไปด้วยเครือข่ายย่อย คือ เครือข่ายซฟเฟิลเอ็กซ์เชนจ์แบบบางส่วน (Partial Shuffle-Exchange Network: PSEN) ขนาด $N/2$ ซึ่งเครือข่าย PSEN นี้เป็นเครือข่ายที่ถูกปรับโครงสร้างให้เหมาะสำหรับการขยายขนาดเครือข่าย โดยปรับจากเครือข่ายโอเมก้า (Omega Network) หรือซฟเฟิลเอ็กซ์เชนจ์ (Shuffle-Exchange) และเนื้อหาในบทนี้จะกล่าวถึงโครงสร้างของทั้งเครือข่ายย่อยและเครือข่ายแบบมีลำดับชั้น พร้อมทั้งการปรับสวิตช์เพื่อการสื่อสารข้อมูลจากต้นทางไปปลายทางสำหรับเครือข่ายดังกล่าว ทั้งประเภทการค้นหาเส้นทางด้วยตนเอง (Self-Routing) หรือการสื่อสารข้อมูลประเภทหนึ่งไปหนึ่ง (Point-to-Point Data Routing) และการสื่อสารข้อมูลประเภทหนึ่งไปหนึ่งแบบขนาน (Parallel Point-to-Point Data Routing) ที่ครอบคลุมถึงการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงอีกด้วย

3.1 โครงสร้างของเครือข่ายซฟเฟิลเอ็กซ์เชนจ์แบบบางส่วน (PSEN)

เครือข่ายซฟเฟิลเอ็กซ์เชนจ์แบบบางส่วน (PSEN) เป็นเครือข่ายย่อยพื้นฐาน (Basic Building Block) สำหรับการสร้างเครือข่ายซฟเฟิลเอ็กซ์เชนจ์แบบมีลำดับชั้น (HSEN) โดยเครือข่าย PSEN มีโครงสร้างที่ปรับมาจากเครือข่ายซฟเฟิลเอ็กซ์เชนจ์ เพื่อลดความซับซ้อนของโครงสร้างเครือข่ายและสะดวกสำหรับการขยายขนาดเครือข่ายจากเครือข่ายเดิมที่ใช้งานอยู่ เนื่องจากการสร้างเครือข่ายใหม่เพื่อให้ได้ขนาดตามความต้องการใช้งานที่เพิ่มขึ้นและเทคโนโลยีที่เปลี่ยนแปลงอย่างรวดเร็ว นั้น มีค่าใช้จ่ายสูงและใช้เวลานานกว่า ซึ่งรายละเอียดโครงสร้างของเครือข่าย PSEN มีรายละเอียดเช่นเดียวกับเครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ (Multistage Interconnection Network: MIN) อื่นๆ (จากหัวข้อ 2.1.2 ในบทที่ 2) นั่นคือ

- 1) จำนวนหน่วยประมวลผลภายในเครือข่าย กำหนดให้เป็น $N = 2^n$ จะได้บิตหมายเลขหน่วยประมวลผล (*id*) n บิต คือ $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2) สวิตช์ที่ใช้ในเครือข่ายกำหนดให้เป็นสวิตช์ขนาด 2×2 ($k = 2$)
- 3) จำนวนสแตจภายในเครือข่าย หาได้จาก $\log_k N = n$ สแตจ ($s_0, s_1, s_2, \dots, s_{n-1}$)
- 4) จำนวนสวิตช์ต่อสแตจ หาได้จาก $N/k = N/2$

เนื่องจากเครือข่าย PSEN ถูกปรับมาจากเครือข่ายซัพเฟิลเอ็กซ์เชนจ์ ดังนั้นโครงสร้างส่วนใหญ่จะยังคงโครงสร้างเครือข่ายซัพเฟิลเอ็กซ์เชนจ์ นั่นคือ เป็นเครือข่ายที่มีรูปแบบการเชื่อมต่อระหว่างสแตจ (ISC) เพียงรูปแบบเดียว แต่มีส่วนที่แตกต่าง คือ การเชื่อมต่อระหว่างข้อมูลเข้ากับสวิตช์ในสแตจแรก (s_0) จะต่อตรง ซึ่งสำหรับเครือข่ายซัพเฟิลเอ็กซ์เชนจ์ จะต่อแบบเพอร์เฟกซ์เฟิล (Perfect Shuffle) รายละเอียดของรูปแบบการเชื่อมต่อระหว่างสแตจภายในเครือข่าย PSEN และการประยุกต์ใช้ฟังก์ชันการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเอง (Self-Routing) ของเครือข่ายซัพเฟิลเอ็กซ์เชนจ์มีดังนี้

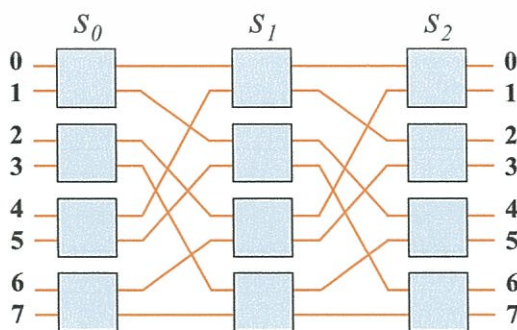
1) รูปแบบการเชื่อมต่อระหว่างสแตจ (ISC)

ภายในเครือข่าย PSEN มีรูปแบบการเชื่อมต่อระหว่างสแตจ แบ่งออกเป็น 2 ส่วน คือ

- การเชื่อมต่อระหว่างข้อมูลนำเข้ากับสวิตช์ในสแตจ s_0 และการเชื่อมต่อระหว่างสแตจ s_{n-1} กับข้อมูลออกเป็นการต่อตรง
- การเชื่อมต่อระหว่างสวิตช์ในสแตจ s_0 กับ s_1 การเชื่อมต่อระหว่างสวิตช์ในสแตจ s_1 กับ s_2 จนกระทั่งการเชื่อมต่อระหว่างสวิตช์ในสแตจ s_{n-2} กับ s_{n-1} นั้น จะมีรูปแบบการเชื่อมต่อที่เรียกว่า เพอร์เฟกซ์เฟิล (Perfect Shuffle) รูปแบบการเชื่อมต่อนี้สามารถทำได้โดยการนำบิตหมายเลขหน่วยประมวลผลต้นทาง $b_{n-1}, b_{n-2}, b_{n-3}, \dots, b_2, b_1, b_0$ เลื่อนบิตไปทางซ้ายแบบวนกลับ 1 บิต (1-Bit Circular Left-Shifting) ซึ่งทำให้บิตที่มีนัยสำคัญมากที่สุด (Most Significant Bit) ถูกย้ายตำแหน่งไปอยู่ในตำแหน่งที่มีนัยสำคัญน้อยสุด (Least Significant Bit) จะได้บิตหมายเลขปลายทาง (รูปที่ 2.5 ในบทที่ 2) คือ

$$b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 \xrightarrow{\text{ISC}} b_{n-2}b_{n-3}b_{n-4} \dots b_1b_0b_{n-1}$$

รูปแบบการเชื่อมต่อระหว่างสแตจทั้งหมดภายในเครือข่าย PSEN ขนาด $N = 8$ แสดงได้ดังรูปที่ 3.1



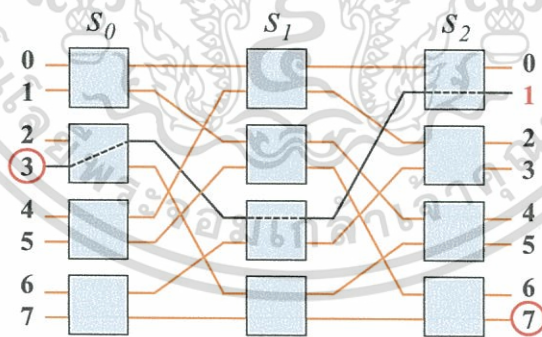
รูปที่ 3.1 รูปแบบการเชื่อมต่อระหว่างสแตจภายในเครือข่าย PSEN ($N = 8$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

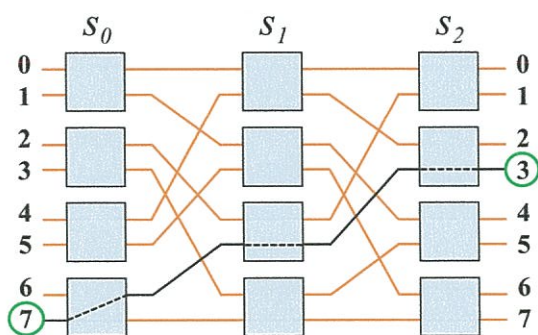
2) การหาเส้นทางจากต้นทางไปปลายทางด้วยตนเอง (Self-Routing)

จากเครือข่าย MIN อื่นๆ (ในบทที่ 2) จะเห็นได้ว่า เครือข่ายทั้งสามเครือข่ายนั้น (ซัพเฟิล เอ็กซ์เชนจ์ ฟลิป และคิวบ์) มีการหาเส้นทางด้วยตนเองเพื่อการสื่อสารข้อมูลโดยการพิจารณาหมายเลขบิต id หน่วยประมวลผลทั้งต้นทางและปลายทางด้วยการดำเนินการ XOR ดังนั้นจะแสดงการดำเนินการหาเส้นทางด้วยตนเองด้วยวิธีการเดียวกันบนเครือข่าย PSEN ดังตัวอย่างที่ 3.1

ตัวอย่าง 3.1 การหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย PSEN ($N = 8$) ระหว่างหน่วยประมวลผลต้นทาง $id: 3$ ไปหน่วยประมวลผลปลายทาง $id: 7$ ด้วยการดำเนินการ XOR



รูปที่ 3.2 การหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย PSEN ($N = 8$) ที่ดำเนินการไม่สำเร็จ ($id: 3 \rightarrow id: 7$)

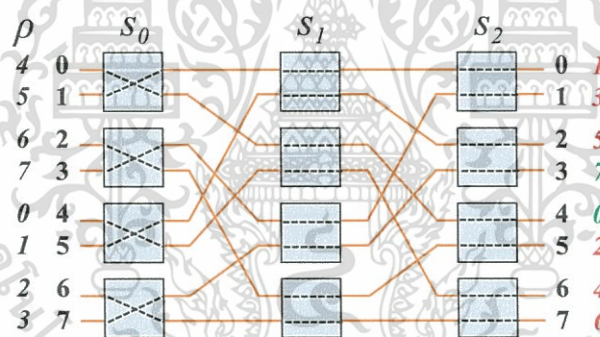


รูปที่ 3.3 การหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย PSEN ($N = 8$) ที่ดำเนินการสำเร็จ ($id: 7 \rightarrow id: 3$)

จากตัวอย่าง 3.1 จะเห็นว่า การหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย PSEN โดยการดำเนินการ XOR ระหว่างหมายเลขบิต id หน่วยประมวลผลทั้งต้นทางและปลายทาง ให้ผลลัพธ์การหาเส้นทางที่สำเร็จสำหรับบางคู่ต้นทางกับปลายทางเท่านั้น (เช่น รูปที่ 3.3 $id: 7 \rightarrow id: 3$) ดังนั้นหากใช้วิธีดำเนินการ XOR ดังกล่าวสำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง จะไม่สามารถสื่อสารข้อมูลได้สำเร็จ ดังรูปที่ 3.4 ซึ่งเป็นตัวอย่างการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง สำหรับการเรียงสับเปลี่ยน (Permutation) $\rho = (4\ 5\ 6\ 7\ 0\ 1\ 2\ 3)$ ซึ่งเป็น การเรียงสับเปลี่ยนที่เครือข่าย MIN อื่นๆ (จากบทที่ 2) สามารถใช้เพื่อสื่อสารระหว่างคู่ต้นทางกับปลายทางได้สำเร็จ (การเชื่อมต่อภายในสวิตช์ทำได้ตามตารางที่ 3.1) และรูปที่ 3.5 เป็นอีกตัวอย่างของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง โดยการเรียงสับเปลี่ยน $\rho = (4\ 0\ 5\ 1\ 6\ 2\ 7\ 3)$ ซึ่งได้จากรูปที่ 3.4 (การเชื่อมต่อภายในสวิตช์ทำได้ตามตารางที่ 3.2) ตัวอย่างในรูปที่ 3.5 นี้เป็นตัวอย่างการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงที่ไม่สำเร็จบนเครือข่าย PSEN เนื่องจากเกิดการขัดแย้งในสวิตช์ (Switch Conflict) กล่าวคือ เกิดการใช้พอร์ต (Out-Port) ของสวิตช์ร่วมกันในขณะเวลาหนึ่งของการสื่อสาร (เนื่องจากการเชื่อมต่อภายในสวิตช์ไม่สามารถเป็นแบบ k ต่อ 1 (Many-to-One) ได้) จึงไม่สามารถสื่อสารจากต้นทางไปปลายทางได้สำเร็จ ดังนั้นเครือข่าย PSEN ที่มีโครงสร้างพิเศษจึงจำเป็นต้องมีวิธีการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองที่สามารถทำให้การสื่อสารสามารถทำได้สำเร็จ (คือ การสื่อสารระหว่างต้นทางกับปลายทางทำได้ถูกต้องตามการเรียงสับเปลี่ยนที่เหมาะสมสำหรับเครือข่าย พร้อมทั้งไม่เกิดการขัดแย้งภายในสวิตช์ขณะสื่อสาร)

ตารางที่ 3.1 การเชื่อมต่อภายในสวิตช์บนเครือข่าย MIN ด้วย $\rho = (4\ 5\ 6\ 7\ 0\ 1\ 2\ 3)$

ต้นทาง (S)		ปลายทาง (D)		S XOR D	การเชื่อมต่อภายในสวิตช์
ฐานสิบ	ฐานสอง	ฐานสิบ	ฐานสอง		
0	000	4	100	100	x--
1	001	5	101	100	x--
2	010	6	110	100	x--
3	011	7	111	100	x--
4	100	0	000	100	x--
5	101	1	001	100	x--
6	110	2	010	100	x--
7	111	3	011	100	x--

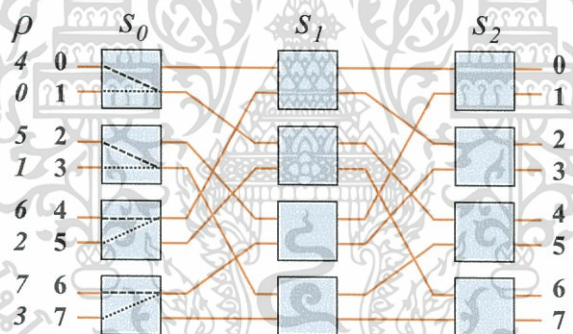


รูปที่ 3.4 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN ($N=8$) ที่ไม่สำเร็จเนื่องจากการเรียงสับเปลี่ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 การเชื่อมต่อภายในสวิตช์บนเครือข่าย PSEN ด้วย $\rho = (4\ 0\ 5\ 1\ 6\ 2\ 7\ 3)$

ต้นทาง (S)		ปลายทาง (D)		$S \text{ XOR } D$	การเชื่อมต่อภายในสวิตช์
ฐานสิบ	ฐานสอง	ฐานสิบ	ฐานสอง		
0	000	4	100	100	x --
1	001	0	000	001	-- x
2	010	5	101	111	x x x
3	011	1	001	010	- x -
4	100	6	110	010	- x -
5	101	2	010	111	x x x
6	110	7	111	001	-- x
7	111	3	011	100	x --



รูปที่ 3.5 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN ($N = 8$) ที่ไม่สำเร็จเนื่องจากเกิดการขัดแย้งในสวิตช์

3.2 การปรับสวิตช์บนเครือข่ายซฟเฟิลเอ็กซ์เชนจ์แบบบางส่วน (PSEN)

การเรียงสับเปลี่ยนที่สามารถทำให้การสื่อสารระหว่างคู่ต้นทางกับปลายทางบนเครือข่ายสำเร็จได้นั้น เป็นสิ่งที่สำคัญสำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง แต่ถึงแม้ว่าจะเลือกใช้การเรียงสับเปลี่ยนที่เหมาะสมสำหรับเครือข่ายแล้วก็ตาม แต่ฟังก์ชันการปรับสวิตช์ (สำหรับการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเอง) ที่เหมาะสมเพื่อให้การสื่อสารบนเครือข่ายสำเร็จได้ก็มีความสำคัญเช่นเดียวกัน จะเห็นได้จากรูปที่ 3.5 ซึ่งเลือกใช้การเรียงสับเปลี่ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่สามารถทำให้การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN สำเร็จได้ แต่การปรับสวิตช์เพื่อให้การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN โดยการดำเนินการ XOR ระหว่างหมายเลขบิต id หน่วยประมวลผลทั้งต้นทางและปลายทางดังเช่นเครือข่าย MIN อื่นๆ นั้นไม่สามารถทำได้สำเร็จ ดังนั้นเครือข่าย PSEN จึงต้องการการปรับสวิตช์ที่เหมาะสมเพื่อให้การสื่อสารดังกล่าวสำเร็จได้

3.2.1 การปรับสวิตช์บนเครือข่าย PSEN

กำหนดให้หมายเลขบิต id หน่วยประมวลผลต้นทาง (S) n บิต คือ $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ และหมายเลขบิต id หน่วยประมวลผลปลายทาง (D) n บิต คือ $d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$ แล้ว การปรับสวิตช์บนเครือข่าย PSEN สามารถทำได้ดังนี้

หมายเลขบิต id หน่วยประมวลผลต้นทาง n บิต จะถูกเลื่อนไปทางขวาแบบวนกลับ 1 บิต นั่นคือ

$$b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 \xrightarrow{\text{ISC}} b_0b_{n-1}b_{n-2} \dots b_3b_2b_1$$

เมื่อได้หมายเลขบิต id หน่วยประมวลผลต้นทาง n บิตที่ทำการเลื่อนบิตแล้ว (S') จึงสามารถนำไปใช้ในการกำหนดเงื่อนไขเพื่อดำเนินการ XOR เพื่อปรับสวิตช์ในสแตจที่เกี่ยวข้องสำหรับการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย PSEN ได้ ดังรูปที่ 3.7

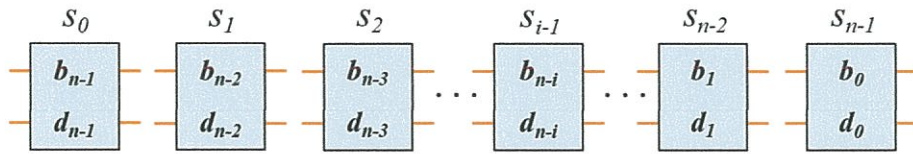
3.2.2 ความแตกต่างระหว่างการปรับสวิตช์สำหรับเครือข่าย MIN ที่มีอยู่กับการปรับสวิตช์สำหรับเครือข่าย PSEN

ความแตกต่างระหว่างการปรับสวิตช์สำหรับเครือข่าย MIN ที่มีอยู่กับการปรับสวิตช์สำหรับเครือข่าย PSEN แสดงดังรูปที่ 3.6 และ 3.7 นั่นคือ ในรูปที่ 3.6 เป็นเงื่อนไขการเชื่อมต่อภายในสวิตช์สำหรับเครือข่าย MIN อื่นๆ โดยที่แต่ละสวิตช์จะถูกปรับการเชื่อมต่อตามผลลัพธ์การ XOR ระหว่าง S และ D โดยตรง นั่นคือ

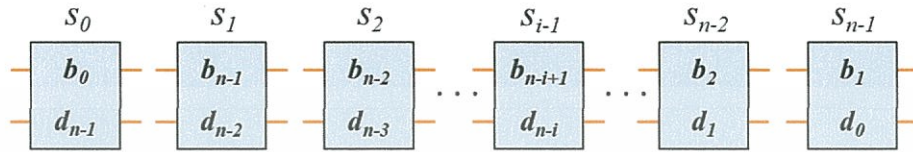
สวิตช์ในแต่ละสแตจ s_{i-1} โดยที่ $i = 1, 2, \dots, n$ จะถูกปรับเป็นต่อตรง (Straight: “-”) หาก $b_{n-i} = d_{n-i}$ แต่ถ้า $b_{n-i} \neq d_{n-i}$ สวิตช์ในสแตจดังกล่าวจะถูกปรับเป็นสับเปลี่ยน (Exchange: “x”)

ส่วนในรูปที่ 3.7 เป็นเงื่อนไขการปรับสวิตช์สำหรับเครือข่าย PSEN โดยที่แต่ละสวิตช์จะถูกปรับตามผลลัพธ์การ XOR ระหว่าง S' ที่เลื่อนขวา 1 บิตแล้ว ($S': b_0b_{n-1}b_{n-2} \dots b_3b_2b_1$) กับ D นั่นคือ

สวิตช์ในแต่ละสแตจ s_{i-1} ($i = 1, 2, \dots, n$) จะถูกปรับเป็นต่อตรง (-) หาก $b_{(n+1) \bmod n} = d_{n-i}$ แต่ถ้าไม่เท่ากัน สวิตช์ในสแตจดังกล่าวจะถูกปรับเป็นสับเปลี่ยน (x)



รูปที่ 3.6 เงื่อนไขการเชื่อมต่อภายในสวิตช์สำหรับการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย MIN อื่นๆ



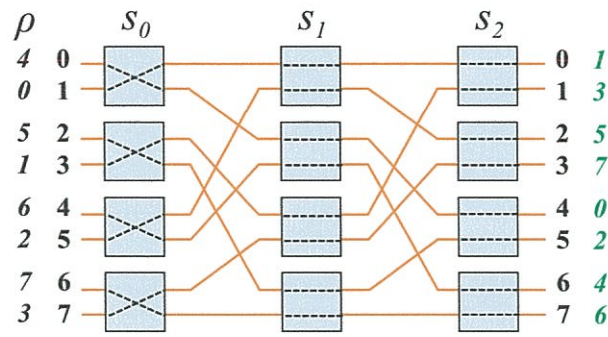
รูปที่ 3.7 เงื่อนไขการปรับสวิตช์สำหรับการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย PSEN

ตารางที่ 3.3 เป็นการปรับสวิตช์บนเครือข่าย PSEN ($N=8$) ด้วยฟังก์ชันการปรับสวิตช์ที่นำเสนอ (ดูรูปที่ 3.7) โดยใช้ $\rho = (4\ 0\ 5\ 1\ 6\ 2\ 7\ 3)$ ซึ่งสามารถทำให้การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN สำเร็จได้ ดังรูปที่ 3.8

ตารางที่ 3.3 การปรับสวิตช์บนเครือข่าย PSEN ที่ดำเนินการสำเร็จสำหรับ $\rho = (4\ 0\ 5\ 1\ 6\ 2\ 7\ 3)$ ด้วยวิธีการปรับสวิตช์ที่เสนอ

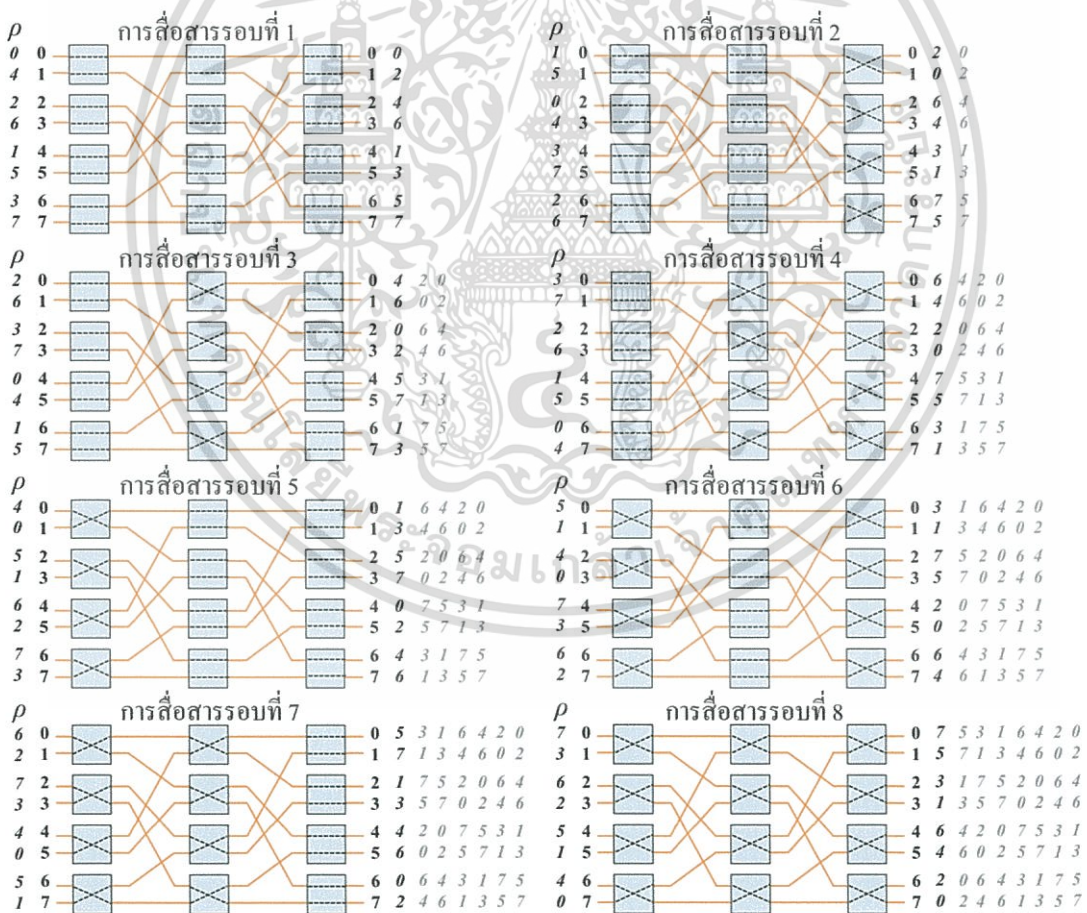
ต้นทาง (S)			ปลายทาง (D)		$S' \text{ XOR } D$	การปรับสวิตช์ (Switch Adjusting)
ฐานสิบ	ฐานสอง	เลื่อนบิต	ฐานสิบ	ฐานสอง		
0	000	000	4	100	100	x--
1	001	100	0	000	100	x--
2	010	001	5	101	100	x--
3	011	101	1	001	100	x--
4	100	010	6	110	100	x--
5	101	110	2	010	100	x--
6	110	011	7	111	100	x--
7	111	111	3	011	100	x--

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN ($N = 8$) ที่ดำเนินการสำเร็จด้วยเงื่อนไขการปรับสวิตช์ที่เสนอ

รูปที่ 3.9 เป็นรูปแบบการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN ($N = 8$) ที่สำเร็จ ทั้ง N รอบการสื่อสาร ซึ่งจะทำให้ทุกหน่วยประมวลผลได้รับข้อมูลจากทุกๆ หน่วยประมวลผลในเครือข่ายครบทุกหน่วย



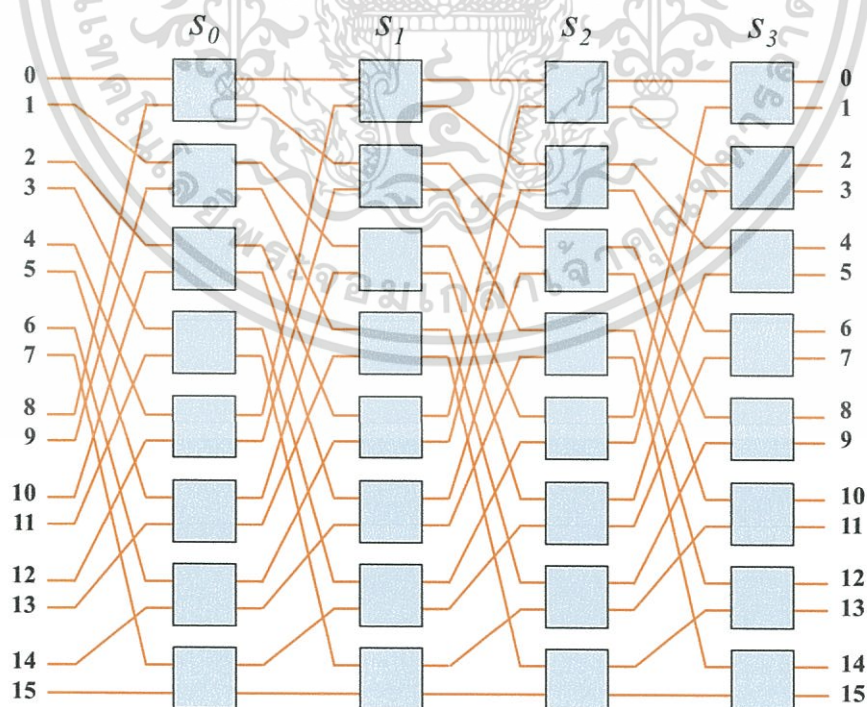
รูปที่ 3.9 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย PSEN ($N = 8$) ที่ดำเนินการสำเร็จทั้ง N รอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โครงสร้างของเครือข่ายซฟเฟิลเอ็กซ์เซนจ์แบบมีลำดับชั้น (HSEN)

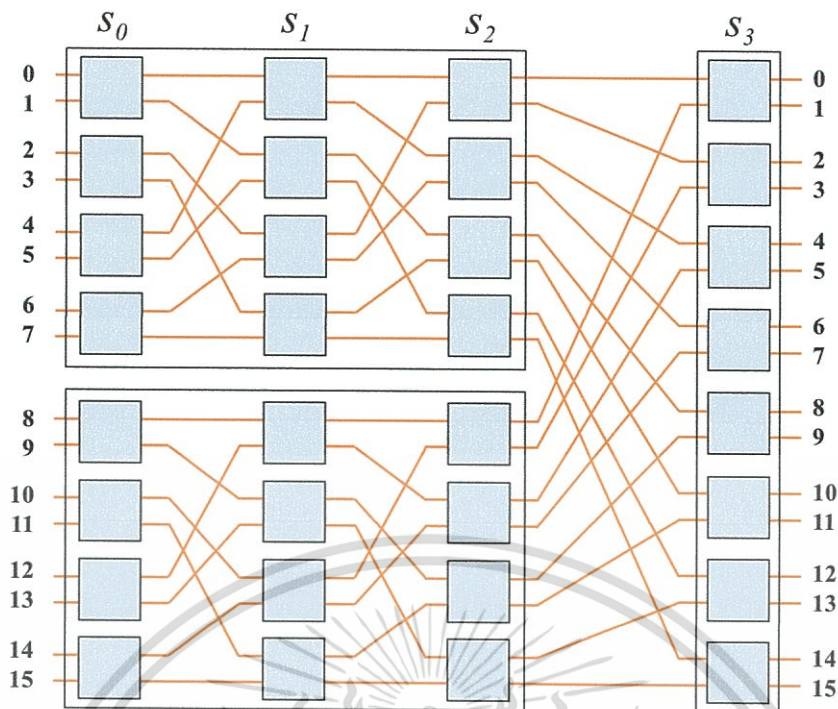
เครือข่ายซฟเฟิลเอ็กซ์เซนจ์แบบมีลำดับชั้น (HSEN) สามารถสร้างขึ้น โดยการนำเครือข่ายย่อย PSEN ที่ออกแบบไว้ในหัวข้อ 3.2 ซึ่งเป็นเครือข่ายเดิมที่ใช้งานอยู่มาประกอบกันเป็นเครือข่ายใหม่ที่ใหญ่ขึ้น เนื่องจากการสร้างเครือข่ายซฟเฟิลเอ็กซ์เซนจ์ใหม่ทั้งระบบนั้น (รูปที่ 3.10) มีค่าใช้จ่ายที่ค่อนข้างสูง และใช้เวลาในการสร้างมากกว่าการรวมเครือข่ายเดิมที่มีอยู่ (รูปที่ 3.11) ให้มีขนาดที่ใหญ่ขึ้นและเพียงพอต่อความต้องการใช้งานที่มากขึ้น เช่น หากเดิมมีเครือข่าย PSEN ($N = 8$) อยู่ 2 เครือข่าย และต้องการเครือข่ายใหม่ที่ใหญ่ขึ้น โดยต้องการให้มีหน่วยประมวลผลภายในเครือข่ายเท่ากับ $N = 16$ สามารถทำได้โดยการเชื่อมต่อ N พอร์ต (ในสแตจสุดท้าย) เท่านั้น ในขณะที่การสร้างเครือข่ายใหม่ให้มีขนาดเท่ากับ $N = 16$ จะต้องทำการเชื่อมต่อ $N \times \log_2 N$ พอร์ต (ในทุกๆ สแตจ) สำหรับการเชื่อมต่อระหว่างสแตจ หรือ ISC ภายในเครือข่าย HSEN นั้น รูปแบบการเชื่อมต่อ $ISC_1 - ISC_{n-2}$ มีการเชื่อมต่อเช่นเดียวกับการเชื่อมต่อภายในเครือข่าย PSEN ($N/2 = 8$) และรวมเข้ากับ ISC_{n-1} ใหม่ สำหรับการเชื่อมต่อระบบย่อยทั้งสองที่มีหน่วยประมวลผลเท่ากับ $N/2 (= 8)$ ให้เป็นเครือข่ายใหม่ที่มีหน่วยประมวลผลเท่ากับ $N (= 16)$ โดยใช้รูปแบบการเชื่อมต่อแบบเพอร์เฟกซ์ฟเฟิล (Perfect Shuffle) สำหรับสแตจสุดท้ายทั้ง n บิต

หมายเหตุ จำนวนสวิตช์ที่ใช้ทั้งหมดในเครือข่ายทั้ง 2 แบบ (รูปที่ 3.10 และรูปที่ 3.11) มีจำนวนเท่ากัน คือ $(N/2) \times \log_2 N$ สวิตช์



รูปที่ 3.10 โครงสร้างของเครือข่ายซฟเฟิลเอ็กซ์เซนจ์ ($N = 16$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 โครงสร้างของเครือข่ายฟีลด์เอ็กซ์เชนจ์แบบมีลำดับชั้น ($N=16$)

3.4 การปรับสวิตช์บนเครือข่ายฟีลด์เอ็กซ์เชนจ์แบบมีลำดับชั้น (HSEN)

การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายฟีลด์เอ็กซ์เชนจ์แบบมีลำดับชั้น (HSEN) จะเกิดปัญหาเช่นเดียวกับเครือข่ายเครือข่ายฟีลด์เอ็กซ์เชนจ์แบบบางส่วน (PSEN) นั่นคือ อาจเกิดการขัดแย้งในสวิตช์ขณะสื่อสารข้อมูล และการส่งข้อมูล ไปไม่ถึงปลายทางที่ถูกต้อง (Mismatch Destintions) ซึ่งเกิดจากการเรียงสับเปลี่ยนสำหรับการจับคู่ต้นทางกับปลายทางเพื่อสื่อสารข้อมูล และการปรับสวิตช์ที่ไม่ถูกต้องสำหรับเครือข่าย HSEN ที่มีโครงสร้างพิเศษ ดังนั้นเพื่อให้การสื่อสารข้อมูลแบบขนานภายในเครือข่าย HSEN สามารถทำได้สำเร็จนั้น ต้องมีการดำเนินการที่เหมาะสมกับหมายเลขบิตต้นทาง (S) เพื่อให้สามารถหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองได้

3.4.1 การปรับสวิตช์บนเครือข่าย HSEN

กำหนดให้หมายเลขบิต id หน่วยประมวลผลต้นทาง (S) n บิต คือ $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ และหมายเลขบิต id หน่วยประมวลผลปลายทาง (D) n บิต คือ $d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$ แล้ว การปรับสวิตช์บนเครือข่าย HSEN สามารถทำได้ดังนี้

หมายเลขบิต id หน่วยประมวลผลต้นทาง n บิต จะถูกเลื่อนบิต 2 ครั้ง ตามลำดับ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1) เลื่อนบิตไปทางขวาแบบวนกลับ 1 บิต สำหรับ $n-1$ บิตของเครือข่ายย่อย จะได้

$$b_{n-1}(b_{n-2}b_{n-3} \dots b_2b_1b_0) \xrightarrow{\text{เลื่อนขวา}} b_{n-1}(b_0b_{n-2} \dots b_3b_2b_1)$$

- 2) เลื่อนบิตไปทางซ้ายแบบวนกลับ 1 บิต สำหรับ n บิตของทั้งเครือข่ายที่ผ่านการเลื่อนบิตจากขั้นตอน 1) แล้ว จะได้

$$(b_{n-1}b_0b_{n-2} \dots b_3b_2b_1) \xrightarrow{\text{เลื่อนซ้าย}} (b_0b_{n-2}b_{n-3} \dots b_2b_1b_{n-1})$$

เมื่อได้หมายเลขบิต id หน่วยประมวลผลคั่นทาง n บิตที่ทำการเลื่อนบิตทั้งสองครั้งแล้ว (S') จึงสามารถดำเนินการ XOR เพื่อปรับสวิตช์ในสแตจที่เกี่ยวข้องสำหรับการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย HSEN ได้ ดังรูปที่ 3.12

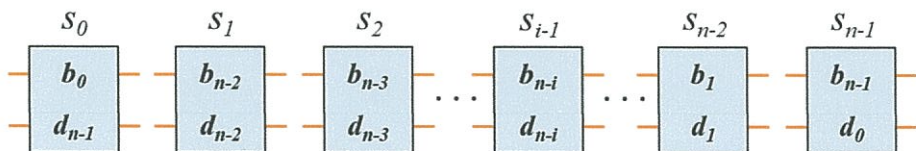
3.4.2 ความแตกต่างระหว่างการปรับสวิตช์สำหรับเครือข่าย MIN ที่มีอยู่กับการปรับสวิตช์สำหรับเครือข่าย HSEN

ความแตกต่างระหว่างการปรับสวิตช์สำหรับเครือข่าย MIN ที่มีอยู่กับการปรับสวิตช์สำหรับเครือข่าย HSEN แสดงดังรูปที่ 3.6 และ 3.12 นั่นคือ ในรูปที่ 3.6 เป็นเงื่อนไขการเชื่อมต่อภายในสวิตช์สำหรับเครือข่าย MIN อื่นๆ โดยจะปรับการเชื่อมต่อตามผลลัพธ์การ XOR ระหว่าง S และ D โดยตรง นั่นคือ

สวิตช์ในแต่ละสแตจ s_{i-1} โดยที่ $i = 1, 2, \dots, n$ จะถูกปรับเป็นต่อตรง (-) หาก $b_{n-i} = d_{n-i}$ แต่ถ้า $b_{n-i} \neq d_{n-i}$ สวิตช์ในสแตจดังกล่าวจะถูกปรับเป็นสับเปลี่ยน (x)

ส่วนเงื่อนไขการปรับสวิตช์สำหรับเครือข่าย HSEN ซึ่งแต่ละสวิตช์จะถูกปรับตามผลลัพธ์การ XOR ระหว่าง S' ที่เลื่อนบิตแล้ว ($b_0b_{n-2}b_{n-3} \dots b_2b_1b_{n-1}$) คือ S' กับ D นั่นคือ

สวิตช์ในแต่ละสแตจ s_{i-1} โดยที่ $i = 1, 2, \dots, n$ จะถูกปรับเป็นต่อตรง (-) หาก b_{n-i} (ของ S') = d_{n-i} แต่ถ้า $b_{n-i} \neq d_{n-i}$ สวิตช์ในสแตจดังกล่าวจะถูกปรับเป็นสับเปลี่ยน (x) ยกเว้น $i = 1$ และ $i = n$ โดยสวิตช์ในสแตจ s_0 ($i = n$) จะถูกปรับจากผลการ XOR ระหว่าง b_0 กับ d_{n-1} และสวิตช์ในสแตจ s_{n-1} ($i = 1$) จะถูกปรับจากผลการ XOR ระหว่าง b_{n-1} กับ d_0 ดังรูปที่ 3.12



รูปที่ 3.12 เงื่อนไขการปรับสวิตช์สำหรับการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเองบนเครือข่าย HSEN

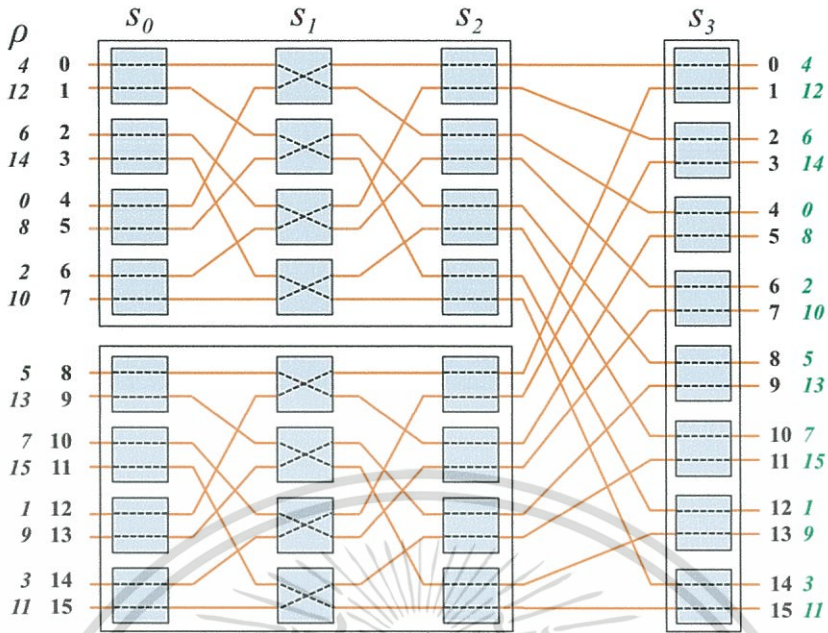
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง 3.2 การแลกเปลี่ยนข้อมูลประเภททั้งหมด ไปทั้งหมดแบบเจาะจงบนเครือข่ายฟีลด์เอ็กซ์เซนต์แบบมีลำดับชั้น ($N = 16$) ด้วยการเรียงสับเปลี่ยน $\rho = (4\ 12\ 6\ 14\ 0\ 8\ 2\ 10\ 5\ 13\ 7\ 15\ 1\ 9\ 3\ 11)$ คือ $S: 0 \rightarrow D: 4$ $S: 1 \rightarrow D: 12$ $S: 2 \rightarrow D: 6$... $S: 14 \rightarrow D: 3$ $S: 15 \rightarrow D: 11$ โดยตารางที่ 3.4 เป็นการปรับสวิตช์ตามวิธีการที่นำเสนอ (รูปที่ 3.12) และรูปที่ 3.13 แสดงการปรับสวิตช์บนเครือข่าย HSEN ตามตารางที่ 3.4 ซึ่งเป็นรอบการสื่อสารหนึ่งของการสื่อสารดังกล่าว

ตารางที่ 3.4 การปรับสวิตช์บนเครือข่าย HSEN ที่ดำเนินการสำเร็จสำหรับ $\rho = (4\ 12\ 6\ 14\ 0\ 8\ 2\ 10\ 5\ 13\ 7\ 15\ 1\ 9\ 3\ 11)$ ด้วยเงื่อนไขการปรับสวิตช์ที่เสนอ

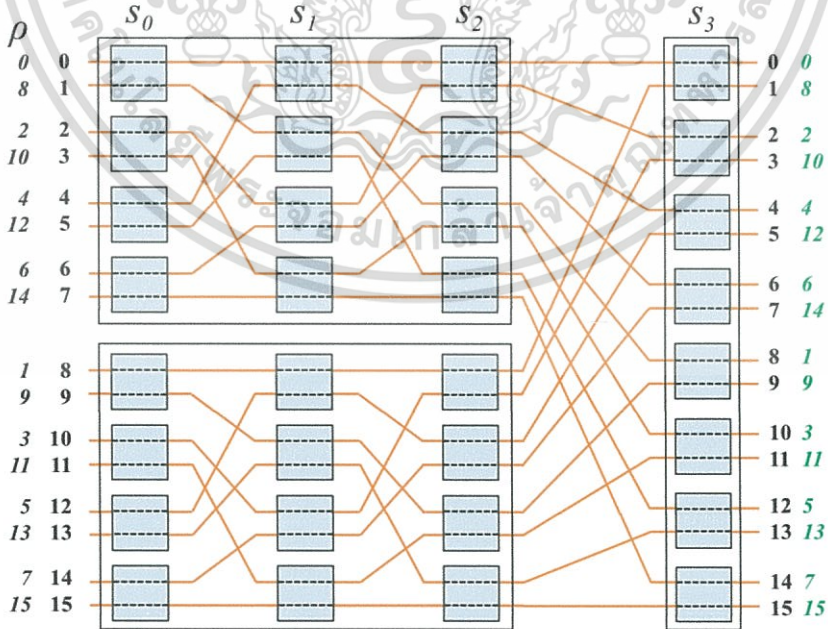
ต้นทาง (S)			ปลายทาง (D)		S' XOR D	การปรับสวิตช์ (Switch Adjusting)
ฐานสิบ	ฐานสอง	เลื่อนบิต	ฐานสิบ	ฐานสอง		
0	0000	0000	4	0100	0100	-x--
1	0001	1000	12	1100	0100	-x--
2	0010	0010	6	0110	0100	-x--
3	0011	1010	14	1110	0100	-x--
4	0100	0100	0	0000	0100	-x--
5	0101	1100	8	1000	0100	-x--
6	0110	0110	2	0010	0100	-x--
7	0111	1110	10	1010	0100	-x--
8	1000	0001	5	0101	0100	-x--
9	1001	1001	13	1101	0100	-x--
10	1010	0011	7	0111	0100	-x--
11	1011	1011	15	1111	0100	-x--
12	1100	0101	1	0001	0100	-x--
13	1101	1101	9	1001	0100	-x--
14	1110	0111	3	0011	0100	-x--
15	1111	1111	11	1011	0100	-x--

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

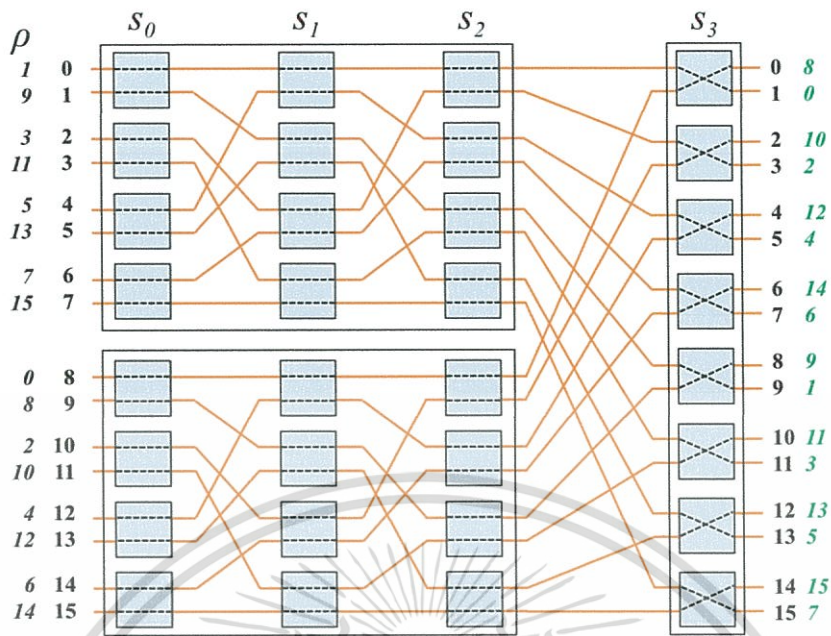


รูปที่ 3.13 การปรับสวิตช์บนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จสำหรับ $\rho = (4\ 12\ 6\ 14\ 0\ 8\ 2\ 10\ 5\ 13\ 7\ 15\ 1\ 9\ 3\ 11)$

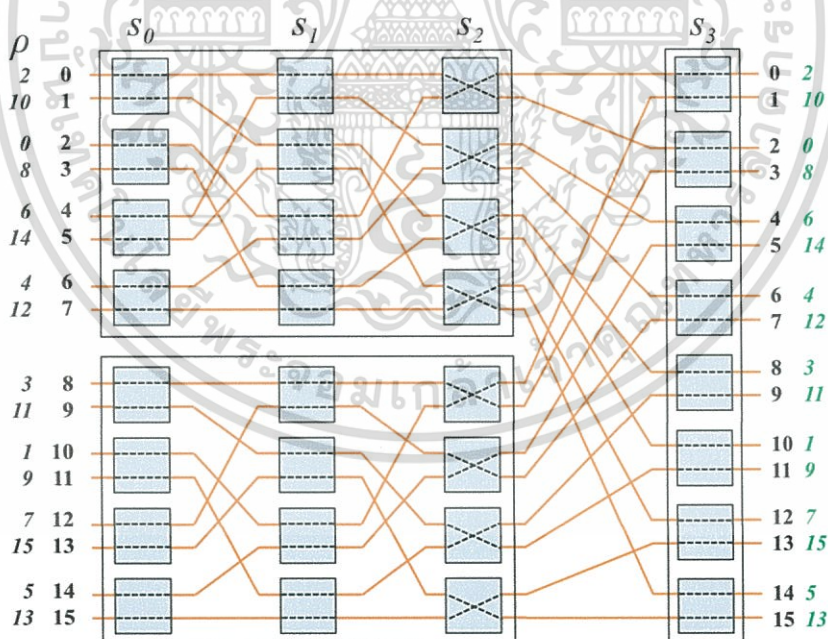
ตัวอย่าง 3.2 การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่ายฟีลด์เอ็กซ์เซนจ์แบบมีลำดับชั้น ($N=16$) ทั้ง N รอบ



รูปที่ 3.14 การสื่อสารรอบที่ 1 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (0\ 8\ 2\ 10\ 4\ 12\ 6\ 14\ 1\ 9\ 3\ 11\ 5\ 13\ 7\ 15)$

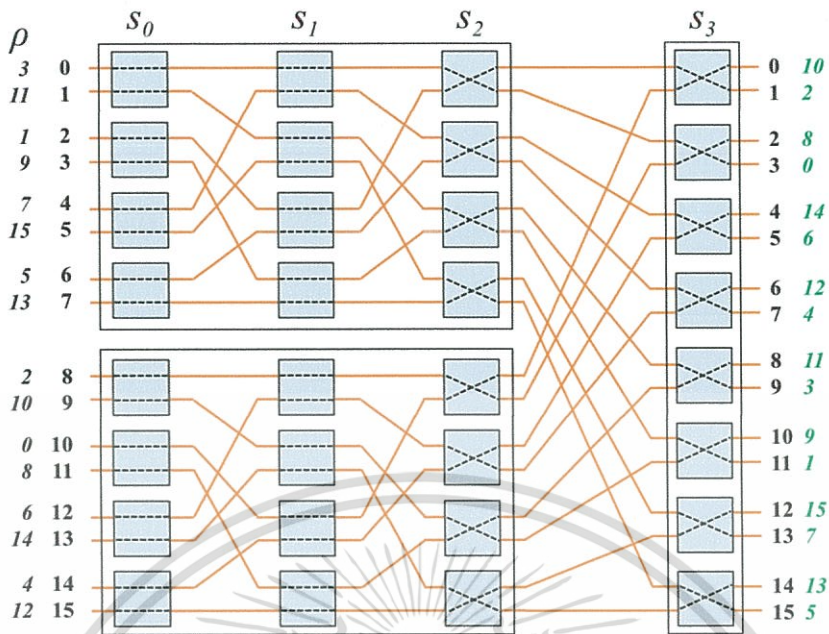


รูปที่ 3.15 การสื่อสารรอบที่ 2 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (1\ 9\ 3\ 11\ 5\ 13\ 7\ 15\ 0\ 8\ 2\ 10\ 4\ 12\ 6\ 14)$

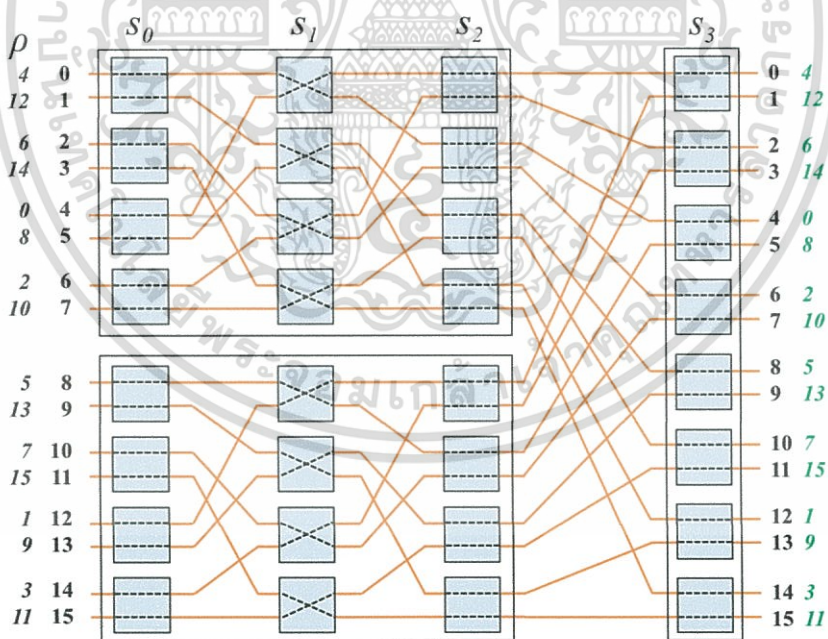


รูปที่ 3.16 การสื่อสารรอบที่ 3 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (2\ 10\ 0\ 8\ 6\ 14\ 4\ 12\ 3\ 11\ 1\ 9\ 7\ 15\ 5\ 13)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

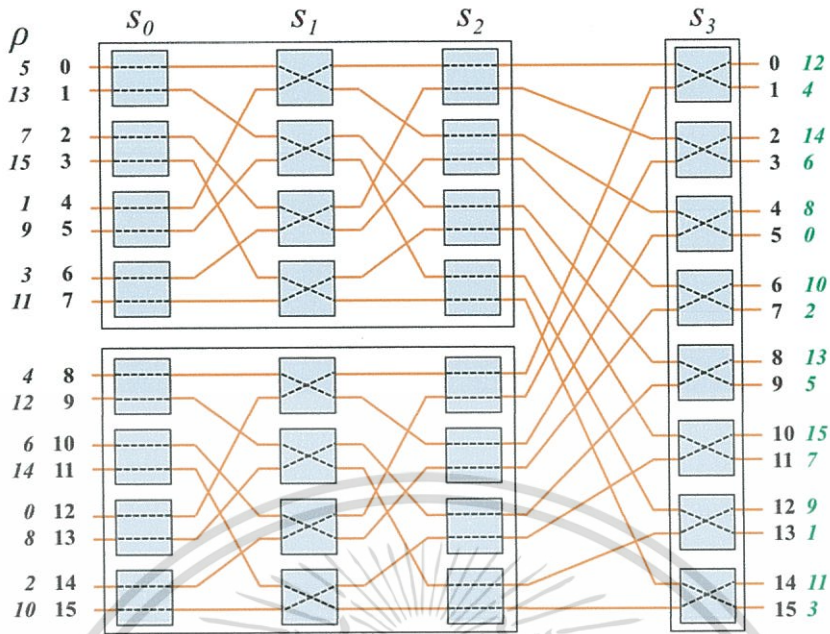


รูปที่ 3.17 การสื่อสารรอบที่ 4 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (3\ 11\ 1\ 9\ 7\ 15\ 5\ 13\ 2\ 10\ 0\ 8\ 6\ 14\ 4\ 12)$

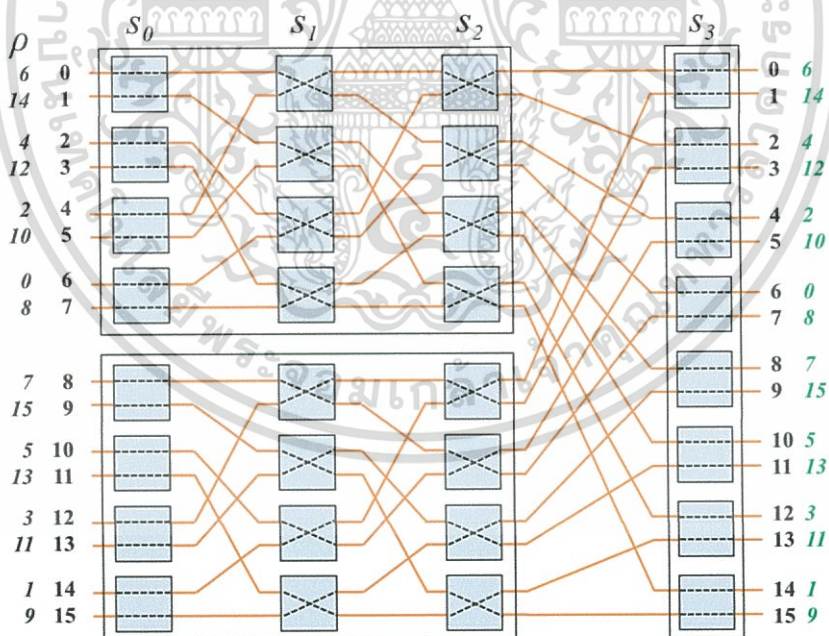


รูปที่ 3.18 การสื่อสารรอบที่ 5 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (4\ 12\ 6\ 14\ 0\ 8\ 2\ 10\ 5\ 13\ 7\ 15\ 1\ 9\ 3\ 11)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

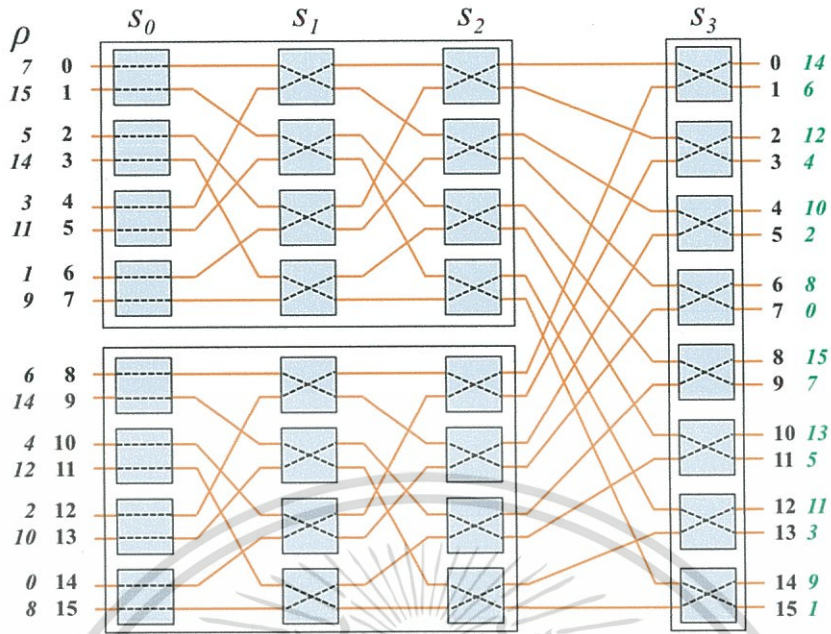


รูปที่ 3.19 การสื่อสารรอบที่ 6 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (5\ 13\ 7\ 15\ 1\ 9\ 3\ 11\ 4\ 12\ 6\ 14\ 0\ 8\ 2\ 10)$

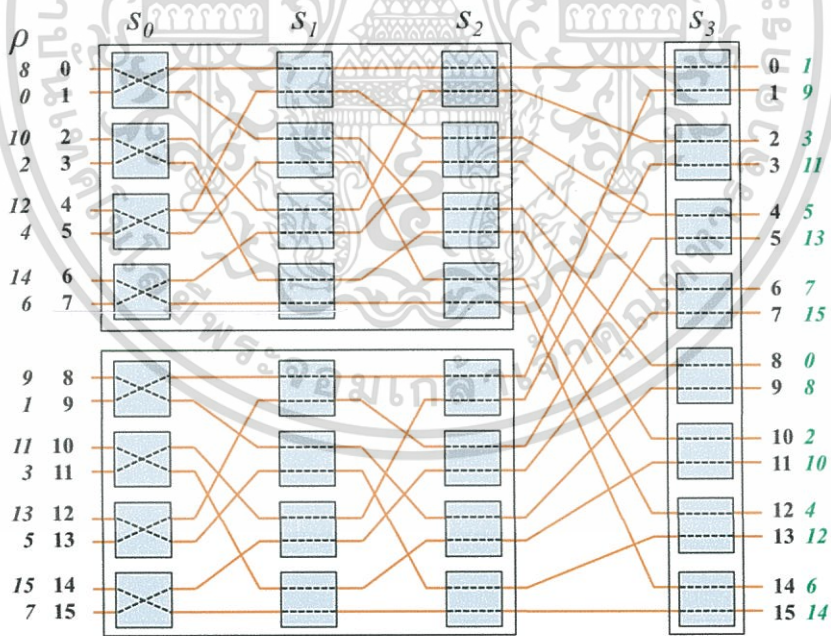


รูปที่ 3.20 การสื่อสารรอบที่ 7 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (6\ 14\ 4\ 12\ 2\ 10\ 0\ 8\ 7\ 15\ 5\ 13\ 3\ 11\ 1\ 9)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

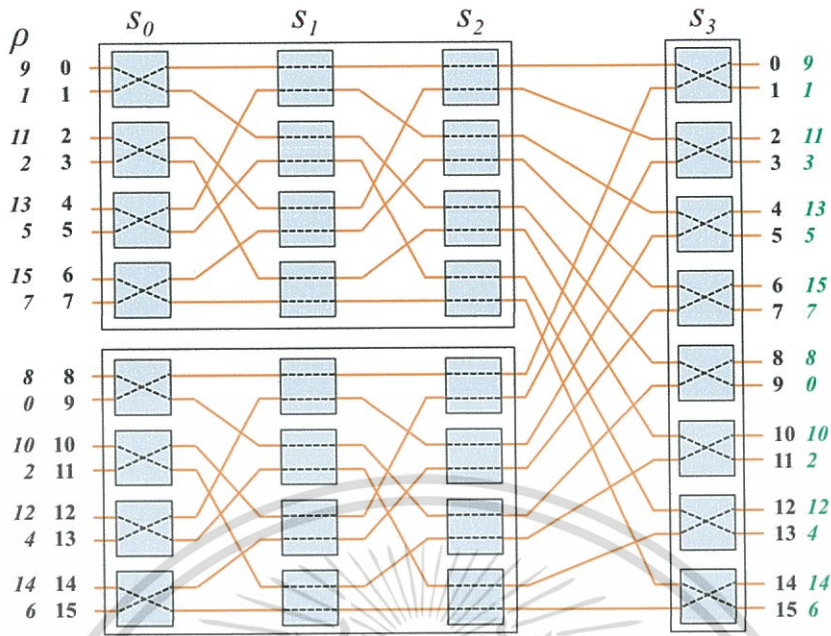


รูปที่ 3.21 การสื่อสารรอบที่ 8 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (7\ 15\ 5\ 14\ 3\ 11\ 1\ 9\ 6\ 14\ 4\ 12\ 2\ 10\ 0\ 8)$

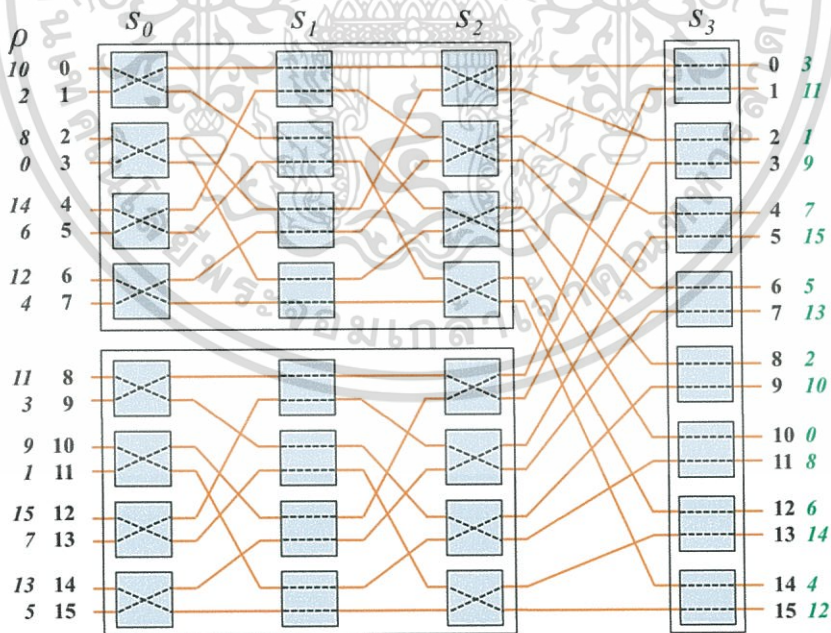


รูปที่ 3.22 การสื่อสารรอบที่ 9 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (8\ 0\ 10\ 2\ 12\ 4\ 14\ 6\ 9\ 1\ 11\ 3\ 13\ 5\ 15\ 7)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

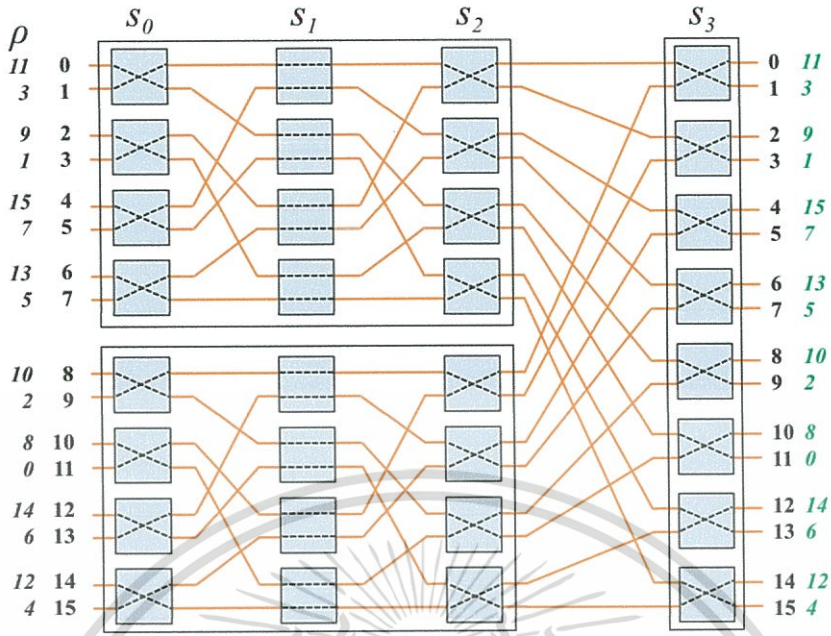


รูปที่ 3.23 การสื่อสารรอบที่ 10 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบ
 เจาจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (9 \ 1 \ 11 \ 2 \ 13 \ 5 \ 15 \ 7 \ 8 \ 0 \ 10 \ 2 \ 12 \ 4$
 $14 \ 6)$

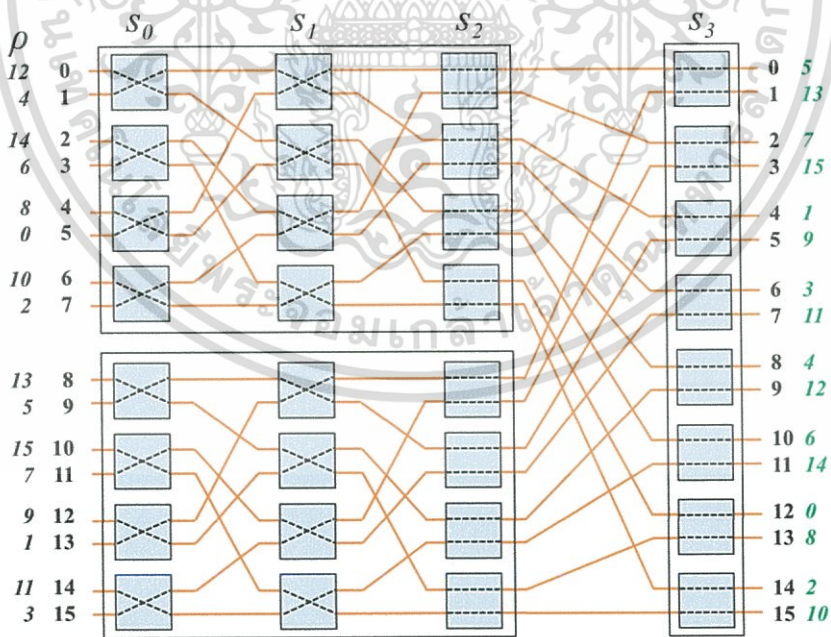


รูปที่ 3.24 การสื่อสารรอบที่ 11 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบ
 เจาจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (10 \ 2 \ 8 \ 0 \ 14 \ 6 \ 12 \ 4 \ 11 \ 3 \ 9 \ 1 \ 15 \ 7$
 $13 \ 5)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

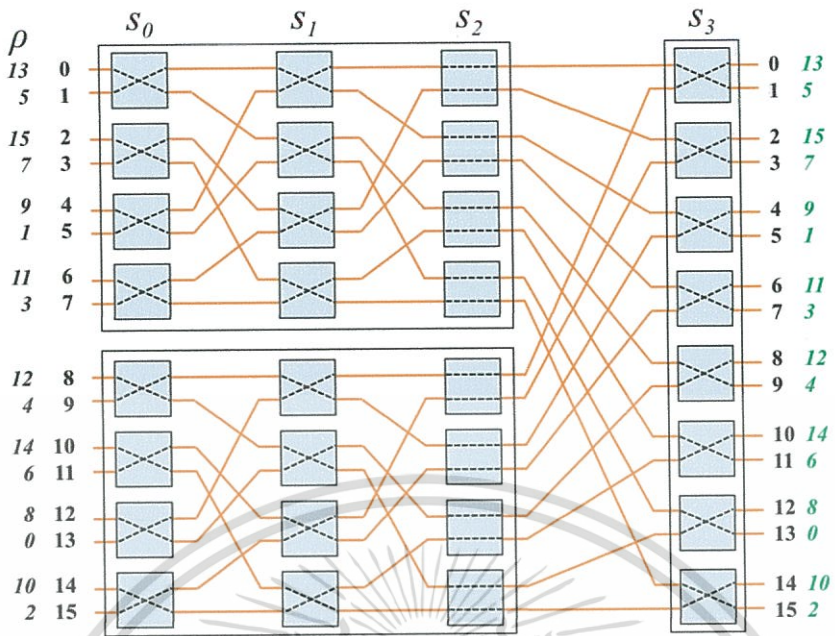


รูปที่ 3.25 การสื่อสารรอบที่ 12 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบ
 เจาจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (11\ 3\ 9\ 1\ 15\ 7\ 13\ 5\ 10\ 2\ 8\ 0\ 14\ 6\ 12\ 4\ 15$
 $12\ 4)$

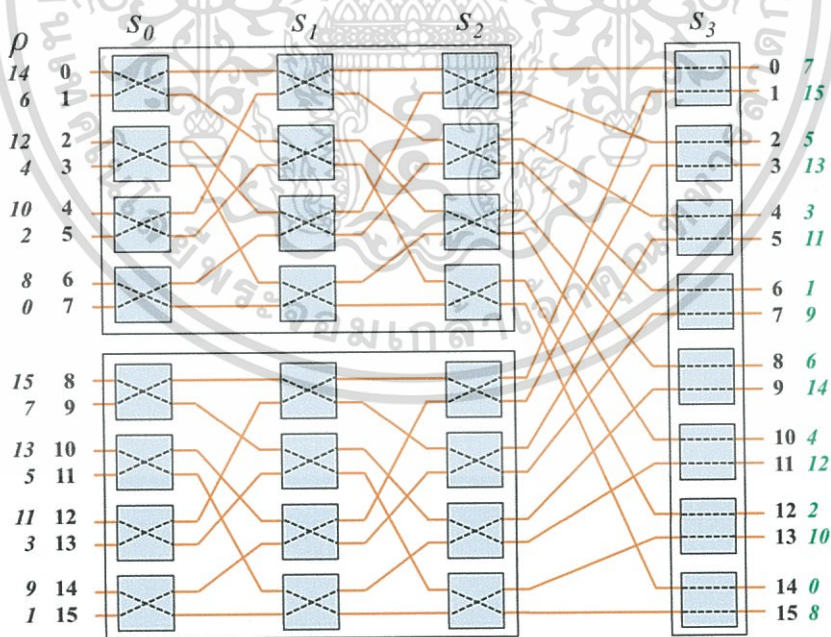


รูปที่ 3.26 การสื่อสารรอบที่ 13 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบ
 เจาจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (12\ 4\ 14\ 6\ 8\ 0\ 10\ 2\ 13\ 5\ 15\ 7\ 9\ 1$
 $11\ 3)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



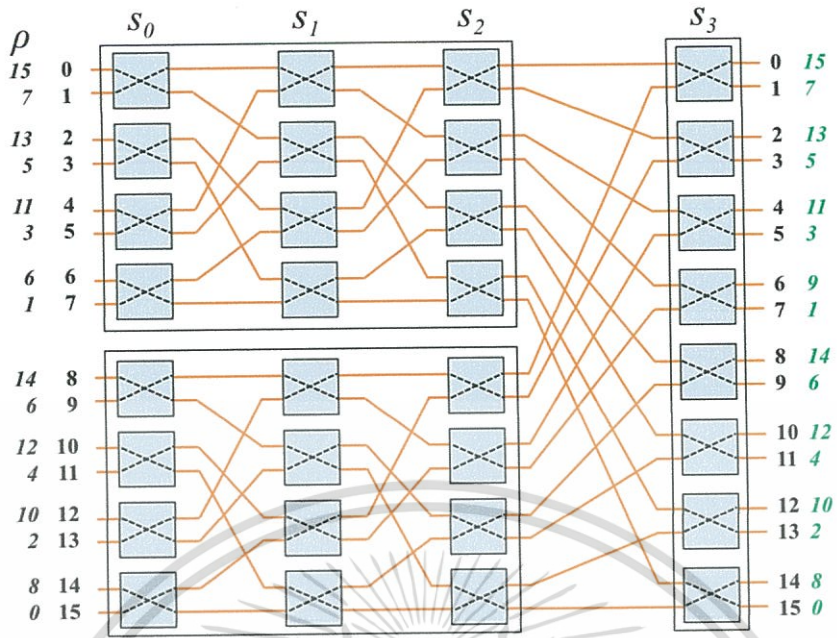
รูปที่ 3.27 การสื่อสารรอบที่ 14 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบ
 เเจาะจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (13\ 5\ 15\ 7\ 9\ 1\ 11\ 3\ 12\ 4\ 14\ 6\ 8\ 10\ 2)$



รูปที่ 3.28 การสื่อสารรอบที่ 15 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบ
 เเจาะจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (14\ 6\ 12\ 4\ 10\ 2\ 8\ 0\ 15\ 7\ 13\ 5\ 11\ 3$

9 1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.29 การสื่อสารรอบที่ 16 ของการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบ
 เจะจงบนเครือข่าย HSEN ($N=16$) ที่ดำเนินการสำเร็จด้วย $\rho = (15\ 7\ 13\ 5\ 11\ 3\ 6\ 1\ 14\ 6\ 12\ 4\ 10\ 2\ 8\ 0)$
 8 0)

ตารางที่ 3.5 ผลการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบจะจงบนเครือข่าย HSEN
 ($N=16$) ที่ดำเนินการสำเร็จทั้ง N รอบ

หน่วยประมวลผล		รอบ															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	8	2	10	4	12	6	14	1	9	3	11	5	13	7	15
2	2	1	9	3	11	5	13	7	15	0	8	2	10	4	12	6	14
3	3	2	10	0	8	6	14	4	12	3	11	1	9	7	15	5	13
4	4	3	11	1	9	7	15	5	13	2	10	0	8	6	14	4	12
5	5	4	12	6	14	0	8	2	10	5	13	7	15	1	9	3	11
6	6	5	13	7	15	1	9	3	11	4	12	6	14	0	8	2	10
7	7	6	14	4	12	2	10	0	8	7	15	5	13	3	11	1	9
8	8	7	15	5	13	3	11	1	9	6	14	4	12	2	10	0	8
9	9	8	0	10	2	12	4	14	6	9	1	11	3	13	5	15	7
10	10	9	1	11	3	13	5	15	7	8	0	10	2	12	4	14	6
11	11	10	2	8	0	14	6	12	4	11	3	9	1	15	7	13	5
12	12	11	3	9	1	15	7	13	5	10	2	8	0	14	6	12	4
13	13	12	4	14	6	8	0	10	2	13	5	15	7	9	1	11	3
14	14	13	5	15	7	9	1	11	3	12	4	14	6	8	0	10	2
15	15	14	6	12	4	10	2	8	0	15	7	13	5	11	3	9	1
16	16	15	7	13	5	11	3	9	1	14	6	12	4	10	2	8	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การวิเคราะห์ความถูกต้องและความซับซ้อนด้านเวลาของการ ปรับสวิตช์บนเครือข่าย PSEN และ HSEN

ในงานวิจัยเรื่องการปรับสวิตช์บนเครือข่ายซัฟเฟิลเอ็กซ์เชนจ์แบบมีลำดับชั้นสำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง (Switch Adjusting on Hierarchical Shuffle-Exchange Networks for All-to-All Personalized Exchange) ซึ่งได้นำเสนอวิธีการปรับสวิตช์เพื่อให้ได้ผลลัพธ์การสื่อสารข้อมูลที่ต้องการไว้ในบทที่ 3 สำหรับเนื้อหาในบทนี้ จะกล่าวถึงการวิเคราะห์ความถูกต้องและความซับซ้อนด้านเวลาของการปรับสวิตช์ ทั้งของเครือข่ายซัฟเฟิลเอ็กซ์เชนจ์แบบบางส่วน (Partial Shuffle-Exchange Network: PSEN) และเครือข่ายซัฟเฟิลเอ็กซ์เชนจ์แบบมีลำดับชั้น (Hierarchical Shuffle-Exchange Network: HSEN) ซึ่งมีเครือข่าย PSEN เป็นเครือข่ายย่อย

4.1 การพิสูจน์ความถูกต้องของการปรับสวิตช์บนเครือข่าย PSEN และ HSEN

เพื่อให้การพิสูจน์ความถูกต้องของการปรับสวิตช์สามารถเข้าใจได้ง่ายขึ้น จะแสดงการพิสูจน์ความถูกต้องของการปรับสวิตช์บนเครือข่ายโอเมก้า (Omega Network) หรือ ซัฟเฟิลเอ็กซ์เชนจ์ (Shuffle-Exchange Network: SEN) ก่อน แล้วจึงแสดงการพิสูจน์ความถูกต้องสำหรับเครือข่าย PSEN และ HSEN ดังทฤษฎีบทที่ 1 - 3 ตามลำดับ

กำหนดให้เครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ (Multistage Interconnection Network: MIN) ประเภทบนพื้นฐานเครือข่ายซัฟเฟิลเอ็กซ์เชนจ์ (Shuffle Exchange-Based) เช่น เครือข่าย SEN PSEN และ HSEN มีหน่วยประมวลผลเท่ากับพอร์ตข้อมูลเข้าและออก (Input/Output Ports) ภายในเครือข่ายเท่ากับ N หน่วยประมวลผล มีสเตจเท่ากับ $n = \log_2 N$ สเตจ และหมายเลขบิต id ของหน่วยประมวลผล (จากพอร์ตข้อมูลเข้าไปพอร์ตข้อมูลออก) แทนอยู่ในรูปเลขฐานสอง n บิต คือ

$S: b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ แทนหมายเลขบิต id ของหน่วยประมวลผลต้นทาง (Source)

$D: d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$ แทนหมายเลขบิต id ของหน่วยประมวลผลปลายทาง (Destination)

การปรับสวิตช์ในแต่ละสเตจบนเครือข่าย SEN สามารถทำได้โดยการพิจารณาหมายเลขบิต id ของหน่วยประมวลผลต้นทางและปลายทางแต่ละคู่ ด้วยการดำเนินการ XOR สำหรับเครือข่าย SEN เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี้ คู่บิตที่มีนัยสำคัญสูงสุด (Most Significant Bit) ของ S และ D จะถูกพิจารณาก่อน (ในสแตจ s_0) จากนั้นคู่บิตถัดไปที่มีนัยสำคัญต่ำลงมาก็จะปรับสวิตช์ในสแตจถัดไป (ในสแตจ s_j) และสุดท้ายการดำเนินการ XOR ระหว่างคู่บิตที่มีนัยสำคัญน้อยสุด (Least Significant Bit) จะปรับสวิตช์ในสแตจ s_{n-1} หลังจากปรับสวิตช์ในแต่ละสแตจ (ให้เป็นต่อตรง (-) หรือสับเปลี่ยน (x)) บิตที่ถูกพิจารณานั้น จะถูกเลื่อนบิตโดยรูปแบบการเชื่อมต่อภายในเครือข่าย หรือ ISC แบบเลื่อนซ้ายแบบวนกลับ 1 บิต และเมื่อหมายเลขบิต id ของหน่วยประมวลผลต้นทาง n บิต ผ่านเข้าไปยัง n สวิตช์ของ n สแตจตามลำดับ หมายเลขบิต id ของหน่วยประมวลผลต้นทาง n บิตนั้น จะถูกแทนด้วยหมายเลขบิต id ของหน่วยประมวลผลปลายทาง

ทฤษฎีบทที่ 1 บนเครือข่ายซัพเฟิลเอ็กเชนจ์ (SEN) เมื่อหมายเลขบิต id ของหน่วยประมวลผลต้นทาง S : $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ ผ่านเข้าไปยังการเชื่อมต่อภายในเครือข่าย หรือ ISC ทั้งหมด n การเชื่อมต่อและ n สวิตช์ (ใน n สแตจ) หมายเลขบิต id ของ S จะถูกแปลงเป็นหมายเลขบิต id ของหน่วยประมวลผลปลายทาง D : $d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$

การพิสูจน์ เมื่อหมายเลขบิต id ของหน่วยประมวลผลต้นทาง (S) ผ่านเข้าไปในแต่ละสแตจ ($s_0 - s_{n-1}$) ตาม ISC (เมื่อผ่านแต่ละสแตจแล้วจะถูกเลื่อนบิตไปทางซ้าย) และสวิตช์ในแต่ละสแตจจะถูกปรับ (เป็นต่อตรง (-) หรือสับเปลี่ยน (x)) จากผลลัพธ์ของการดำเนินการ XOR ระหว่างคู่บิตของ S และ D (ดูการปรับสวิตช์ที่ รูปที่ 3.6 ในบทที่ 3) เพื่อเชื่อมต่อจากพอร์ตข้อมูลเข้า (S) ไปยังพอร์ตข้อมูลออก (D) สามารถแสดงเป็นขั้นตอนได้ดังนี้

$$\begin{array}{l}
 \text{พอร์ตข้อมูลเข้า } - s_0: \quad b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 \xrightarrow{\text{ISC}} b_{n-2}b_{n-3} \dots b_1b_0b_{n-1} \\
 \text{สวิตช์(ใน } s_0): \quad b_{n-2}b_{n-3} \dots b_1b_0b_{n-1} \xrightarrow{\text{sw}} b_{n-2}b_{n-3} \dots b_1b_0d_{n-1} \\
 \text{จากสแตจ } s_0 - s_1: \quad b_{n-2}b_{n-3} \dots b_1b_0d_{n-1} \xrightarrow{\text{ISC}} b_{n-3}b_{n-4} \dots b_0d_{n-1}b_{n-2} \\
 \text{สวิตช์(ใน } s_1): \quad b_{n-3}b_{n-4} \dots b_0d_{n-1}b_{n-2} \xrightarrow{\text{sw}} b_{n-3}b_{n-4} \dots b_0d_{n-1}d_{n-2} \\
 \dots \\
 \text{จากสแตจ } s_{n-2} - s_{n-1}: \quad b_1b_0d_{n-1}d_{n-2} \dots d_3d_2d_1 \xrightarrow{\text{ISC}} b_0d_{n-1}d_{n-2} \dots d_3d_2b_1 \\
 \text{สวิตช์(ใน } s_{n-2}): \quad b_0d_{n-1}d_{n-2} \dots d_3d_2b_1 \xrightarrow{\text{sw}} b_0d_{n-1}d_{n-2} \dots d_3d_2d_1 \\
 \text{จากสแตจ } s_{n-1} - s_n: \quad b_0d_{n-1}d_{n-2} \dots d_3d_2d_1 \xrightarrow{\text{ISC}} d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1b_0 \\
 \text{สวิตช์(ใน } s_{n-1}): \quad d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1b_0 \xrightarrow{\text{sw}} d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0
 \end{array}$$

ดังนั้นสามารถพิสูจน์ได้ว่าการหาเส้นทางจากต้นทางไปปลายทางด้วยตนเอง (Self-Routing) บนเครือข่าย SEN นี้ จาก S : $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ (ณ พอร์ตข้อมูลเข้า) สามารถสื่อสารข้อมูลไปยัง D : $d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$ ได้อย่างถูกต้อง \square

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทฤษฎีบทที่ 2 บนเครือข่ายซัพเฟิลเอ็กเซนจ์แบบบางส่วน (PSEN) กับการปรับสวิตช์ที่เสนอนั้น เมื่อหมายเลขบิต id ของหน่วยประมวลผลต้นทาง $S: b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ ผ่านเข้าไปยังการเชื่อมต่อภายในเครือข่าย หรือ ISC ทั้งหมด $n-1$ การเชื่อมต่อและ n สวิตช์ (ใน n สเตจ) หมายเลขบิต id ของ S จะถูกแปลงเป็นหมายเลขบิต id ของหน่วยประมวลผลปลายทาง $D: d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$

การพิสูจน์ เมื่อหมายเลขบิต id ของหน่วยประมวลผลต้นทาง (S) ผ่านเข้าไปในแต่ละสเตจ ($s_0 - s_{n-1}$) ตาม ISC (เมื่อผ่านแต่ละสเตจแล้วจะถูกเลื่อนบิตไปทางซ้าย) และสวิตช์แต่ละสเตจจะถูกปรับ (เป็นต่อตรง (-) หรือสับเปลี่ยน (x)) จากผลลัพธ์ของการดำเนินการ XOR ระหว่างคู่อันดับของ S (ที่ทำการเลื่อนบิตแล้ว) กับ D (ดูการปรับสวิตช์ที่ รูปที่ 3.7 ในบทที่ 3) เพื่อเชื่อมต่อจากพอร์ตข้อมูลเข้า (S) ไปยังพอร์ตข้อมูลออก (D) สามารถแสดงเป็นขั้นตอนได้ดังนี้

$$\begin{array}{l}
 \text{สวิตช์(ใน } s_0): \quad b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 \xrightarrow{\text{sw}} b_{n-1}b_{n-2} \dots b_2b_1d_{n-1} \\
 \text{จากสเตจ } s_0 - s_1: \quad b_{n-1}b_{n-2} \dots b_2b_1d_{n-1} \xrightarrow{\text{ISC}} b_{n-2}b_{n-3} \dots b_1d_{n-1}b_{n-1} \\
 \text{สวิตช์(ใน } s_1): \quad b_{n-2}b_{n-3} \dots b_1d_{n-1}b_{n-1} \xrightarrow{\text{sw}} b_{n-2}b_{n-3} \dots b_1d_{n-1}d_{n-2} \\
 s_1 - s_2: \quad b_{n-2}b_{n-3} \dots b_1d_{n-1}d_{n-2} \xrightarrow{\text{ISC}} b_{n-3}b_{n-4} \dots d_{n-1}d_{n-2}b_{n-2} \\
 \dots \\
 \text{จากสเตจ } s_{n-3} - s_{n-2}: \quad b_2b_1d_{n-1}d_{n-2} \dots d_3d_2 \xrightarrow{\text{ISC}} b_1d_{n-1}d_{n-2} \dots d_3d_2b_2 \\
 \text{สวิตช์(ใน } s_{n-2}): \quad b_1d_{n-1}d_{n-2} \dots d_3d_2b_2 \xrightarrow{\text{sw}} b_1d_{n-1}d_{n-2} \dots d_3d_2d_1 \\
 \text{จากสเตจ } s_{n-2} - s_{n-1}: \quad b_1d_{n-1}d_{n-2} \dots d_3d_2d_1 \xrightarrow{\text{ISC}} d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1b_1 \\
 \text{สวิตช์(ใน } s_{n-1}): \quad d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1b_1 \xrightarrow{\text{sw}} d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0
 \end{array}$$

จากการผ่านสวิตช์ครั้งสุดท้าย จะได้ $d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$ ซึ่งเป็นหมายเลข id ของหน่วยประมวลผลปลายทาง ดังนั้นเป็นการพิสูจน์ว่าการปรับสวิตช์บนเครือข่าย PSEN ที่เสนอในงานวิจัยนี้นั้น สามารถทำให้การหาเส้นทางด้วยตนเอง (Self-Routing) หรือการสื่อสารข้อมูลประเภทหนึ่งไปหนึ่ง (Point-to-Point Data Routing) บนเครือข่าย PSEN สำเร็จได้

ทฤษฎีบทที่ 3 บนเครือข่ายซัพเฟิลเอ็กเซนจ์แบบมีลำดับชั้น (HSEN) กับการปรับสวิตช์ที่เสนอนั้น เมื่อหมายเลขบิต id ของหน่วยประมวลผลต้นทาง $S: b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ ผ่านเข้าไปยังการเชื่อมต่อภายในเครือข่าย หรือ ISC ทั้งหมด $n-1$ การเชื่อมต่อและ n สวิตช์ (ใน n สเตจ) หมายเลขบิต id ของ S จะถูกแปลงเป็นหมายเลขบิต id ของหน่วยประมวลผลปลายทาง $D: d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$

การพิสูจน์ เครือข่าย HSEN มีการปรับสวิตช์ที่แตกต่างจากเครือข่าย PSEN (หัวข้อ 3.4 ในบทที่ 3) เครือข่าย HSEN ที่มีหน่วยประมวลผลภายในเครือข่ายเท่ากับ N และมี $n = \log_2 N$ สเตจ และประกอบไปด้วยเครือข่าย PSEN 2 เครือข่าย ซึ่งมีหน่วยประมวลผลเท่ากับ $N/2$ และ $\log_2 N/2 =$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$n-1$ สเตจ เมื่อหมายเลขบิต id ของหน่วยประมวลผลต้นทาง (S) ผ่านเข้าไปในแต่ละสเตจ ($s_0 - s_{n-2}$) ของเครือข่ายย่อย ($N/2 = 2^{n-1}$) ตาม $ISC_1 - ISC_{n-2}$ (เมื่อผ่านแต่ละสเตจแล้วจะถูกเลื่อนบิตไปทางขวา 1 บิต สำหรับ $n-1$ บิตของเครือข่ายย่อย) เพื่อทำการเชื่อมต่อเครือข่ายย่อยให้เป็นเครือข่ายใหญ่ ($N = 2^n$) จะเชื่อมต่อด้วย ISC_{n-1} (เลื่อนบิตไปทางซ้าย 1 บิต สำหรับ n บิตของทั้งเครือข่าย) และสวิตช์แต่ละสเตจจะถูกปรับ (เป็นต่อตรง (-) หรือสับเปลี่ยน (x)) จากผลลัพธ์ของการดำเนินการ XOR ระหว่างคูบิตของ S (ที่ทำการเลื่อนบิตแล้ว) กับ D (ดูการปรับสวิตช์ที่ รูปที่ 3.12 ในบทที่ 3) เพื่อเชื่อมต่อจากพอร์ตข้อมูลเข้า (S) ไปยังพอร์ตข้อมูลออก (D) สามารถแสดงเป็นขั้นตอนได้ดังนี้

$$\begin{array}{l}
 \text{สวิตช์(ใน } s_0\text{):} \quad b_{n-1}b_{n-2}b_{n-3}\dots b_2b_1b_0 \xrightarrow{\text{sw}} b_{n-1}b_{n-2}\dots b_2b_1d_{n-1} \\
 \text{จากสเตจ } s_0 - s_1: \quad b_{n-1}(b_{n-2}b_{n-3}\dots b_2b_1d_{n-1}) \xrightarrow{\text{ISC}_1} b_{n-1}(b_{n-3}\dots b_1d_{n-1}b_{n-2}) \\
 \text{สวิตช์(ใน } s_1\text{):} \quad b_{n-1}(b_{n-3}\dots b_1d_{n-1}b_{n-2}) \xrightarrow{\text{sw}} b_{n-1}(b_{n-3}\dots b_1d_{n-1}d_{n-2}) \\
 \text{จากสเตจ } s_1 - s_2: \quad b_{n-1}(b_{n-3}\dots b_1d_{n-1}d_{n-2}) \xrightarrow{\text{ISC}_2} b_{n-1}(b_{n-4}\dots d_{n-1}d_{n-2}b_{n-3}) \\
 \dots \\
 \text{จากสเตจ } s_{n-3} - s_{n-2}: \quad b_{n-1}(b_1d_{n-1}\dots d_4d_3d_2) \xrightarrow{\text{ISC}_{n-1}} b_{n-1}(d_{n-1}d_{n-2}\dots d_3d_2b_1) \\
 \text{สวิตช์(ใน } s_{n-2}\text{):} \quad b_{n-1}(d_{n-1}d_{n-2}\dots d_3d_2b_1) \xrightarrow{\text{sw}} b_{n-1}(d_{n-1}d_{n-2}\dots d_3d_2d_1) \\
 \text{จากสเตจ } s_{n-2} - s_{n-1}: \quad (b_{n-1}d_{n-1}d_{n-2}\dots d_3d_2d_1) \xrightarrow{\text{ISC}_n} (d_{n-1}d_{n-2}d_{n-3}\dots d_2d_1b_{n-1}) \\
 \text{สวิตช์(ใน } s_{n-1}\text{):} \quad (d_{n-1}d_{n-2}d_{n-3}\dots d_2d_1b_{n-1}) \xrightarrow{\text{sw}} (d_{n-1}d_{n-2}d_{n-3}\dots d_2d_1d_0)
 \end{array}$$

จากการปรับสวิตช์บนเครือข่าย HSEN ที่นำเสนอในงานวิจัยนี้ ทำให้ได้ $d_{n-1}d_{n-2}d_{n-3}\dots d_2d_1d_0$ ซึ่งเป็นหมายเลข id ของหน่วยประมวลผลปลายทาง ดังนั้นเป็นการพิสูจน์ว่าการปรับสวิตช์บนเครือข่าย HSEN นี้ สามารถทำให้การหาเส้นทางด้วยตนเองหรือการสื่อสารข้อมูลประเภทหนึ่งไปหนึ่งบนเครือข่าย HSEN สำเร็จได้

4.2 การวิเคราะห์ความซับซ้อนด้านเวลาของการปรับสวิตช์บนเครือข่าย PSEN และเครือข่าย HSEN

ความซับซ้อนด้านเวลาของการปรับสวิตช์บนเครือข่าย PSEN และเครือข่าย HSEN สามารถวิเคราะห์ได้ดังทฤษฎีบทย่อยที่ 1 และ 2

ทฤษฎีบทย่อยที่ 1 ความซับซ้อนด้านเวลาของการดำเนินการปรับสวิตช์บนเครือข่าย PSEN สำหรับหน่วยประมวลผล N หน่วยประมวลผล (ในระบบการประมวลผลแบบขนาน (Parallel)) เท่ากับ $O(1)$

การพิสูจน์ ฟังก์ชันการปรับสวิตช์บนเครือข่าย PSEN ดำเนินการบนพื้นฐานการเลื่อนบิตไปทางขวาแบบวนกลับ 1 บิต (n บิตของ S : $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ ในรีจิสเตอร์ (Register) ขนาด n บิต) ขั้นตอนการเลื่อนบิตสามารถทำได้โดยวงจรการเลื่อนบิตภายในหน่วยประมวลผล (Shift Logic in CPU) ภายในหนึ่งหน่วยเวลา และสำหรับระบบการประมวลผลแบบขนาน (N หน่วยประมวลผล) ทุกๆ N วงจรการเลื่อนบิตภายในหน่วยประมวลผล จะทำภายในหนึ่งหน่วยเวลา ดังนั้นความซับซ้อนด้านเวลาของการดำเนินการปรับสวิตช์บนเครือข่าย PSEN คือ $O(1)$

ทฤษฎีบทย่อยที่ 2 ความซับซ้อนด้านเวลาของการดำเนินการปรับสวิตช์บนเครือข่าย HSEN สำหรับหน่วยประมวลผล N หน่วยประมวลผล (ในระบบการประมวลผลแบบขนาน) เท่ากับ $O(1)$

การพิสูจน์ ฟังก์ชันการปรับสวิตช์บนเครือข่าย HSEN ดำเนินการบนพื้นฐานการเลื่อนบิต 2 ครั้ง นั่นคือ การเลื่อนบิตไปทางขวาแบบวนกลับ 1 บิต ($n-1$ บิตแรกๆของ S : $b_{n-1}(b_{n-2}b_{n-3} \dots b_2b_1b_0)$) และการเลื่อนบิตไปทางซ้ายแบบวนกลับ 1 บิต (n บิตของ S ที่ผ่านการเลื่อนบิต $n-1$ บิตแรกแล้ว) แต่ละขั้นตอนของการเลื่อนบิตสามารถทำได้โดยวงจรการเลื่อนบิตภายในหน่วยประมวลผล ภายในหนึ่งหน่วยเวลาเท่านั้น และสำหรับระบบการประมวลผลแบบขนาน (N หน่วยประมวลผล) ทุกๆ N วงจรการเลื่อนบิตภายในหน่วยประมวลผล จะทำภายในสองหน่วยเวลา ดังนั้นความซับซ้อนด้านเวลาของการดำเนินการปรับสวิตช์บนเครือข่าย HSEN คือ $O(1)$

บทสรุปและแนวทางการพัฒนางานวิจัย

เนื้อหาในบทนี้เป็นบทสรุปของงานวิจัยเรื่องการปรับสวิตช์บนเครือข่ายซัพเฟลเอ็กซ์เชนจ์แบบมีลำดับชั้นสำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจง (Switch Adjusting on Hierarchical Shuffle-Exchange Networks for All-to-All Personalized Exchange) พร้อมทั้งเสนอแนวทางในการพัฒนางานวิจัย

5.1 บทสรุป

งานวิจัยนี้ได้ศึกษาและออกแบบฟังก์ชันการปรับสวิตช์บนเครือข่ายซัพเฟลเอ็กซ์เชนจ์แบบบางส่วน (Partial Shuffle-Exchange Network: PSEN) และเครือข่ายซัพเฟลเอ็กซ์เชนจ์แบบมีลำดับชั้น (Hierarchical Shuffle-Exchange Network: HSEN) โดยเครือข่าย PSEN จะเป็นเครือข่ายย่อยสำหรับเครือข่าย HSEN และเป็นเครือข่ายที่ปรับโครงสร้างมาจากเครือข่ายโอเมก้า (Omega Network) หรือ เครือข่ายซัพเฟลเอ็กซ์เชนจ์ (Shuffle-Exchange Network) เพื่อลดความซับซ้อนของเครือข่ายและสะดวกสำหรับการขยายขนาดเครือข่ายเดิมที่ใช้งานอยู่ให้เพียงพอต่อความต้องการ ซึ่งเป็นการประหยัดค่าใช้จ่ายและใช้เวลาในการสร้างเครือข่ายใหม่น้อยกว่า การปรับสวิตช์สำหรับเครือข่าย PSEN และ HSEN เป็นการปรับสวิตช์ที่สามารถทำให้การหาเส้นทางด้วยตนเอง (Self-Routing) หรือการสื่อสารข้อมูลประเภทหนึ่งไปหนึ่ง (Point-to-Point Data Routing) ทำได้สำเร็จ ซึ่งมีผลทำให้การแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงสำเร็จได้ เนื่องจากการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงเป็นการแลกเปลี่ยนข้อมูลประเภทหนึ่งที่มีความสำคัญในระบบการประมวลผลแบบขนานและแบบกระจาย (Parallel and Distributed System) หากไม่มีการปรับสวิตช์บนเครือข่าย HSEN ที่เสนอในงานวิจัยนี้ การสื่อสารข้อมูลภายในเครือข่ายนี้จะไม่สามารถทำได้ คือ ไม่สามารถส่งข้อมูลจากต้นทางไปปลายทางที่ถูกต้องได้ แม้ว่า จะประยุกต์ใช้การเรียงสับเปลี่ยนหรืออัลกอริทึมสำหรับการแลกเปลี่ยนข้อมูลประเภททั้งหมดไปทั้งหมดแบบเจาะจงที่มีอยู่ก็ตาม และสุดท้ายมีการพิสูจน์ความถูกต้องของการปรับสวิตช์ที่เสนอในงานวิจัยนี้ ซึ่งถูกต้อง นั่นคือ จากการปรับสวิตช์ เมื่อหมายเลขบิต id ของหน่วยประมวลผลต้นทาง $S: b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ ผ่านเข้าไปยังการเชื่อมต่อภายในเครือข่าย (ISC) แล้ว จะถูกแปลงเป็นหมายเลขบิต id ของหน่วยประมวลผลปลายทาง $D: d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$ ได้อย่างถูกต้อง ซึ่งหมายความว่า การสื่อสารดังกล่าวประสบความสำเร็จ สามารถส่งข้อมูลจากต้นทางไปปลายทางที่ถูกต้องได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 แนวทางการพัฒนางานวิจัย

เครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบพลวัต (Dynamic Interconnection Network) หรือเรียกว่า เครือข่ายที่มีการติดต่อสื่อสารระหว่างหน่วยประมวลผลแบบหลายสเตจ (Multistage Interconnection Network: MIN) อื่นๆ จะถูกศึกษาและออกแบบเป็นเครือข่ายที่มีลำดับชั้น ซึ่งมาพร้อมกับฟังก์ชันการปรับสวิตช์บนเครือข่ายเหล่านั้น เพื่อเป็นทางเลือกสำหรับการนำไปใช้กับระบบการประมวลผลแบบขนานที่ใช้งานอยู่จริง โดยสามารถลดเวลาและค่าใช้จ่ายของการสร้างเครือข่ายใหม่ให้เพียงพอต่อความต้องการใช้งานได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Y. Yang and J. Wang, "Optimal all-to-all personalized exchange in self-routable multistage networks," IEEE Trans. Parallel and Distributed System, vol. 11, no. 3, pp. 261-274, 2000.
- [2] Y. Yang and J. Wang, "Optimal all-to-all personalized exchange in a class of optical multistage networks," IEEE Trans. Parallel and Distributed System, vol. 12, no. 6, pp. 567-582, 2001.
- [3] A. Massini, "All-to-all personalized communication on multistage network," Discrete Appl. Math. vol.128, no. 2, pp. 435-446, 2003.
- [4] V.W. Liu, C. Chen, and R.B. Chen, "Optimal all-to-all personalized exchange in d-nary banyan multistage networks," J. Comb. Optim., vol. 14, no. 3, pp. 131-142, 2007.
- [5] W.Y. Chou and C. Chen, "All-to-all personalized exchange in generalized shuffle-exchange networks," J. Theoretical Computer Science, vol. 411, pp. 1669-1684, 2010.
- [6] C-L. Wu and T-Y. Feng, "On a class of multistage interconnection networks," IEEE Trans. Computers, vol. c-29, no. 8, pp. 694-702, 1980.
- [7] N.A.M. Yunus and M. Othman, "Shuffle exchange network in multistage interconnection network: a review and challenges," International Journal of Computer and Electrical Engineering, vol. 3, no. 5, pp. 724-728, 2011.
- [8] จีรพร วีระพันธุ์. เอกสารประกอบการสอนวิชาสถาปัตยกรรมคอมพิวเตอร์แบบขนาน. กรุงเทพฯ: ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2555.

The 10th International Joint Conference on Computer Science and Software Engineering (JCSSE2013)

57

The next decade of Computer Science & Software Engineering



May 29th – 31st, 2013

Department of Computer Science, Mahasarakham University
Maha Sarakham, THAILAND



Switch Adjusting on Hierarchical Shuffle-Exchange Networks for All-to-All Personalized Exchange

Parichat Narksith

Department of Computer Science, Faculty of Science
King Mongkut's Institute of Technology Ladkrabang
(KMITL)
Ladkrabang, Bangkok, 10520, Thailand
s4650803@kmitl.ac.th

Jeeraporn Werapun

Department of Computer Science, Faculty of Science
King Mongkut's Institute of Technology Ladkrabang
(KMITL)
Ladkrabang, Bangkok, 10520, Thailand
ksjeerap@kmitl.ac.th

Abstract—The original all-to-all personalized exchange algorithms were introduced for static interconnection networks (i.e., hypercube, mesh, and torus). Later, the all-to-all personalized exchange methods were extended for dynamic interconnection networks or MINs (Multistage Interconnection Networks i.e., omega (or shuffle exchange), flip, cube, and baseline) with $N = 2^n$ ($= 4, 8, 16, 32, \dots$). Recently, the generalized all-to-all personalized exchange algorithm was designed for shuffle-exchange networks with even N ($= 4, 6, 8, 10, \dots$). However, for a hierarchical shuffle-exchange network, those existing algorithms cannot be directly applied. This paper introduces such a hierarchical MIN along with switch-adjusting function, operated by shift logic in $O(1)$, for applying all-to-all personalized exchange and other parallel point-to-point permutations successfully. Finally, correctness of our switch adjusting on hierarchical shuffle-exchange networks is verified for all-to-all personalized exchange permutations.

Keywords—switch adjusting; hierarchical shuffle-exchange networks; dynamic interconnection networks; multistage interconnection networks; all-to-all personalized exchange; parallel point-to-point permutations

I. INTRODUCTION

At the present time, computer systems have many processors. Interconnection network between the processors also has effect on performance of parallel processing. The parallel communication between the processors through the interconnection network may require more time. All-to-all personalized exchange is one of frequently used parallel communications which is needed in applications of parallel and distributed computing systems, especially in fast Fourier transform (FFT), matrix transposition applications [1-5], and distributed table lookup [3]. In this type of communication, every processor in the system concurrently sends a distinct message to other processors in each time unit. At last, all processors receive the messages from all other processors, while in other class, all-to-all broadcast, each processor sends the same message to all other processors. Originally, all-to-all personalized exchange has been introduced for static interconnection networks, for instance, hypercubes, meshes, and torus. Even though the optimal complete exchange algorithm for hypercubes achieves optimal time complexity, but high-dimensional hypercubes have poorly scalable

because of their unbounded node degrees. On the other hand, meshes and torus have constant node degrees and better scalability but their communication algorithms require more time complexity.

Later, dynamic interconnection networks or multistage interconnection networks (MINs) are introduced. Yang and Wang [1] firstly proved that their all-to-all personalized exchange algorithm on MINs is optimal. The advantages of MIN implementation are their scalability and shorter communication delay. For a class of MINs (i.e., omega (or shuffle exchange), flip, cube, and baseline) with $N = 2^n$ ($= 4, 8, 16, 32, \dots$) processors, that new algorithm was proposed, based on a special Latin Square. Such an algorithm consists of a set of admissible permutations (a subset of the entire $N!$ permutations, which can be realized by a basic MIN) of a unique path MIN for fast switch setting. However, previous all-to-all personalized exchange algorithms, which successfully work on a network with unique path property, presented by [1] and [3] cannot be used on generalized shuffle-exchange networks (GSENs) [5], a type of MINs with even N ($= 4, 6, 8, 10, \dots$) due to GSENs do not have the unique path property. Recently, the study of Chou and Chen [5] focuses on designing the generalized all-to-all personalized exchange algorithm for GSENs.

Those existing all-to-all personalized exchange methods and parallel point-to-point permutations may reach a set of some realizable permutations by a particular MIN (i.e., shuffle exchange, flip, cube, and baseline). However, those algorithms cannot be directly applied on hierarchical shuffle-exchange networks (HSENs) without switch adjusting before self-routing. Such a hierarchical MIN architecture is constructed by compositing the smaller subsystems into the larger system which is suitable for scalable systems in real parallel engines since system reconfiguration is higher cost and requires more time than subsystems integration. The HSEN networks need the switch adjusting for some permutations which are realizable in these networks. Hence, this paper proposes switch adjusting for successful self-routing, operated by shift logic in $O(1)$ time, for applying all-to-all personalized exchange and parallel point-to-point permutations on the HSEN networks.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The rest of this paper is organized as follows: Section II mentions related work that includes a class of MINs, which are omega, flip, and cube networks together with their self-routing. Section III presents the structure of the hierarchical shuffle-exchange networks (HSENs) along with our switch-adjusting function for all-to-all personalized exchange and other parallel point-to-point permutations. Section IV provides the proof of correctness for successful self-routing on HSENs. Finally, the conclusion and the future work are discussed in the last section.

II. RELATED WORK

A class of multistage interconnection networks (MINs) are reviewed in Section A and then all-to-all personalized exchange on MINs are mentioned in Section B, respectively.

A. A Class of Multistage Interconnection Networks

First, the parameters for a class of MINs [6] (omega, flip, and cube) used in this paper are defined as follows: a network has N processors (N input/output ports). Each processor id (of N processors) can be represented as n -bit, $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$. A number of stages are equal to $n = \log_2 N$ stages and each stage contains $N/2$ of 2×2 switches. Each switching element has two inputs and two outputs. The switch can be set to straight (-) or exchange (x) connections, as shown in Fig. 1. Each of different networks has a unique inter stage connection (ISC) pattern or the interconnection links between switching elements, as detailed in the following networks.

1) Omega network

The omega network or shuffle-exchange network uses the perfect shuffle pattern for ISC between inputs and switches in stage s_0 , switches in stage s_0 and s_1 , ..., and switches in stage s_{n-2} and s_{n-1} , as displayed in Fig. 2 (a) for $N = 8$. The perfect shuffle ISC performs 1-bit circular left-shifting as:

$$b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 \xrightarrow{\text{ISC}} b_{n-2}b_{n-3}b_{n-4} \dots b_1b_0b_{n-1}$$

2) Flip network

The flip network uses the inverted perfect shuffle pattern for ISC between switches in stage s_0 and s_1 , switches in stage s_1 and s_2 , ..., and switches in stage s_{n-1} and outputs (Fig. 2 (b)). The inverted perfect shuffle ISC can be achieved by 1-bit circular right-shifting as:

$$b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 \xrightarrow{\text{ISC}} b_0b_{n-1}b_{n-2} \dots b_3b_2b_1$$

3) Cube network

The cube network, presented in Fig. 2 (d), uses n different ISC patterns within the network. Let ISC_1 refers to ISC between switches in stage s_0 and s_1 , ISC_2 stands for ISC between switches in stage s_1 and s_2 , ..., and ISC_n represents ISC between switches in stage s_{n-1} and outputs. There are the least significant bit and b_i inversing ($i = 1, 2, \dots, n-1$) for ISC_i , and 1-bit circular right-shifting for ISC_n , as follows:

$$\begin{aligned} b_{n-1}b_{n-2}b_{n-3} \dots b_2(b_1b_0) &\xrightarrow{\text{ISC}_1} b_{n-1}b_{n-2} \dots b_2(b_0b_1) \\ b_{n-1}b_{n-2}b_{n-3} \dots b_3(b_2b_1b_0) &\xrightarrow{\text{ISC}_2} b_{n-1}b_{n-2} \dots b_3(b_0b_1b_2) \\ &\dots \\ (b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0) &\xrightarrow{\text{ISC}_{n-1}} (b_0b_{n-2}b_{n-3} \dots b_1b_{n-1}) \\ b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 &\xrightarrow{\text{ISC}_n} b_0b_{n-1}b_{n-2} \dots b_3b_2b_1 \end{aligned}$$

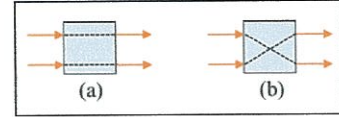


Fig. 1. A 2×2 switching element: (a) Straight connection and (b) Exchange connection.

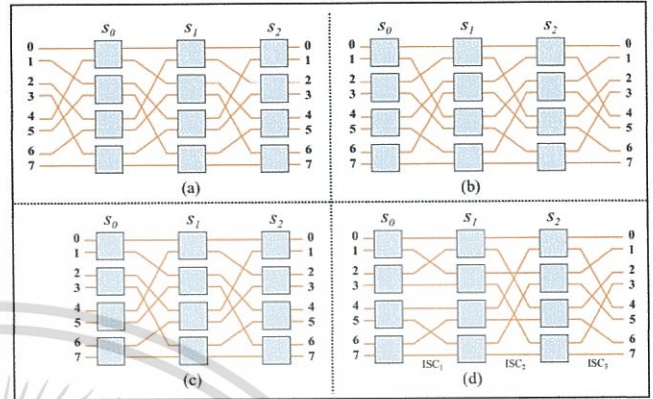


Fig. 2. MINs ($N = 8$) using 2×2 switches: (a) Omega network, (b) Flip network, (c) Partial shuffle-exchange network, and (d) Cube network.

4) Partial shuffle-exchange network (PSEN)

For the PSEN network, we modify from the shuffle-exchange (or omega) network to be the basic building block of our hierarchical shuffle-exchange networks (in Section III). ISC within PSEN is the perfect shuffle pattern which is the same as omega, but PSEN does not include the ISC between inputs and switches in stage s_0 . Therefore, PSEN uses perfect shuffle pattern for ISC between switches in stage s_0 and s_1 , stage s_1 and s_2 , ..., and stage s_{n-2} and s_{n-1} (see Fig. 2 (c)).

B. All-to-All Personalized Exchange on MINs

All-to-all personalized exchange communication can be considered as parallel point-to-point data routing between all pairs of source (S) and destination (D). This communication requires a switch setting for self-routing on a MIN. Switch setting can obtain from the XOR operation of two processor id (S and D). If b_i ($i = 0, 1, \dots, n-1$) of S and D is different, the corresponding switch is set to exchange (x); otherwise that switch is set to straight (-). Fig. 3 illustrates parallel point-to-point data routing for a permutation realized by the omega network, which is used for applying on MINs, described in previous section. The permutation, $\rho = \phi_1 II = (4\ 5\ 6\ 7\ 0\ 1\ 2\ 3)$, is obtained from Latin Square, proposed in [1]. This ρ means input 4 is mapped to output 0, input 5 is mapped to output 1, input 6 is mapped to output 2, and so on. The result from XOR, $r_{n-1}r_{n-2}r_{n-3} \dots r_2r_1r_0$ is brought to set the switches in stages s_{n-1} , s_{n-2} , ..., s_1 , and s_0 , for flip and cube networks. For omega network and PSEN, the switches in stages s_0 , s_1 , ..., s_{n-2} , and s_{n-1} are set, respectively. After XOR operation for ρ (for $N = 8$ in Table I), every couple of S and D gives a result as "x -", which means the switches in stage s_0 , s_1 , and s_2 are set to "exchange-straight-straight" for omega and PSEN. On the other hand, the corresponding switches are set to "- x", "straight-straight-exchange" for flip and cube network.

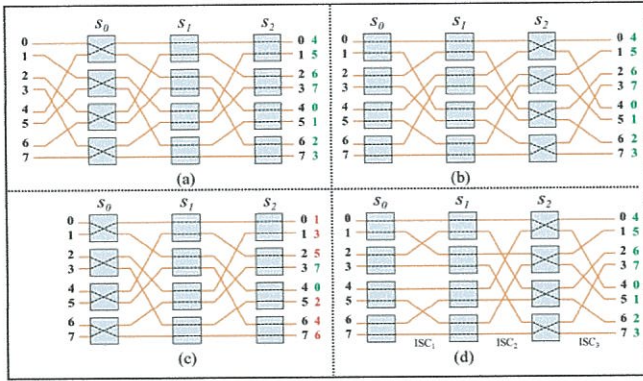


Fig. 3. Switch setting for $\rho = (4\ 5\ 6\ 7\ 0\ 1\ 2\ 3)$ on MINs ($N = 8$).

TABLE I. SUCCESSFUL SWITCH SETTING ON MINs ($N = 8$) FOR PERMUTATION "4 5 6 7 0 1 2 3"

Source		Destination		$S \text{ XOR } D$	Switch setting
dec	bin	dec	bin		
0	000	4	100	100	x--
1	001	5	101	100	x--
2	010	6	110	100	x--
3	011	7	111	100	x--
4	100	0	000	100	x--
5	101	1	001	100	x--
6	110	2	010	100	x--
7	111	3	011	100	x--

From Fig. 3, the results show that only PSEN network ($N = 8$) cannot connect from S to D properly when using the existing switch-setting function. Obviously, data routing is correct only input 0 to output 4 and input 7 to output 3. Hence, an appropriate switch-adjusting function for parallel point-to-point permutations is needed for PSENS.

III. SWITCH ADJUSTING ON HSENS FOR ALL-TO-ALL PERSONALIZED EXCHANGE

In this section, we introduce the hierarchical shuffle-exchange network (HSEN) along with our switch-adjusting function by using PSEN as our basic building block.

A. Switch Adjusting on PSENS

Successful all-to-all communication has to communicate by using admissible parallel point-to-point permutations, which can be realized by using $(S \text{ XOR } D)$ switch setting on MINs. The PSEN network also can realize a set of those permutations. For example, the permutation, $\rho = (1\ 3\ 5\ 7\ 0\ 2\ 4\ 6)$, is a realized permutation of PSEN as other MINs. Switches in each stage can be set, as illustrated in Table II.

TABLE II. UNSUCCESSFUL SWITCH SETTING ON PSEN ($N = 8$) FOR PERMUTATION "1 3 5 7 0 2 4 6"

Source		Destination		$S \text{ XOR } D$	Switch setting
dec	bin	dec	bin		
0	000	4	100	100	x--
1	001	0	000	001	--x
2	010	5	101	111	xxx
3	011	1	001	010	-x-
4	100	6	110	010	-x-
5	101	2	010	111	xxx
6	110	7	111	001	--x
7	111	3	011	100	x--

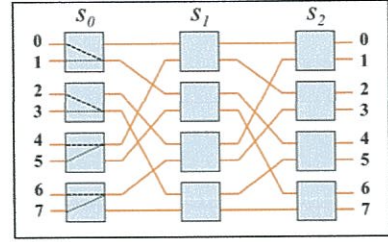


Fig. 4. Unsuccessful parallel point-to-point routing for $\rho = (1\ 3\ 5\ 7\ 0\ 2\ 4\ 6)$ on PSEN ($N = 8$) with some conflict switches at stage s_0 .

Fig. 4 shows unsuccessful parallel point-to-point routing for $\rho = (1\ 3\ 5\ 7\ 0\ 2\ 4\ 6)$ on PSEN ($N = 8$) because of conflict in some switches at stage s_0 . Even though the realized permutation is used, but conflict may occur on PSENS. In order to solve that conflict problem (Fig. 4) and mismatch between S and D (reported in Fig. 3 (c)), we introduce a switch-adjusting function for accomplishment self-routing on PSENS (such as all-to-all personalized exchange and other parallel point-to-point permutations of $N!$ permutations).

For a successful parallel point-to-point routing between S and D , where $S: b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ and $D: d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$, n -bit of processor id of S are shifted by 1-bit circular right-shifting to adjust corresponding switches as:

$$b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 \xrightarrow{\text{shift R}} b_0b_{n-1}b_{n-2} \dots b_3b_2b_1$$

After shifting S , n bits of shifted S and n bits of D will be operated by using XOR operation (to adjust switches in each of n stages from S to D) before applying $\rho = (1\ 3\ 5\ 7\ 0\ 2\ 4\ 6)$ correctly, as defined in Table III. Fig. 6 shows the successful parallel point-to-point routing for $\rho = (1\ 3\ 5\ 7\ 0\ 2\ 4\ 6)$ and Fig. 7 displays all results of applying all-to-all personalized exchange permutations on the PSEN network ($N = 8$).

The different between the existing switch setting and our switch adjusting is defined in Fig. 5. The block diagram in Fig. 5 (a) shows the switch setting in self-routing on shuffle-exchange networks, where each switch is set from the result of $S \text{ XOR } D$. In each stage s_{i-1} ($i = 1, 2, \dots, n$), if b_{n-i} (of S) = d_{n-i} (of D) then corresponding switch is set to straight (-); otherwise set to exchange (x). Fig. 5 (b) illustrates the result of the proposed switch adjusting for self-routing on PSENS. Each switch (in stage s_{i-1}) is set according to the result of (shift) $S \text{ XOR } D$, where (shift) $S = b_0b_{n-1}b_{n-2} \dots b_3b_2b_1$. In stage s_{i-1} , if $b_{(n+i) \bmod n} = d_{n-i}$ then the corresponding switch is set to straight (-); otherwise set to exchange (x), respectively.

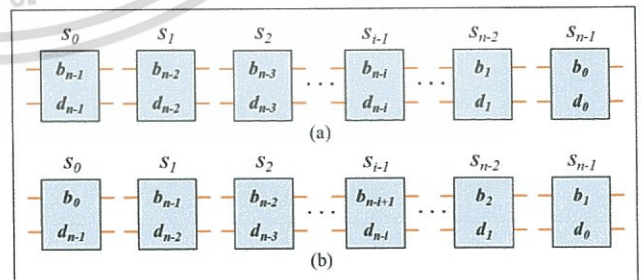


Fig. 5. (a) The existing switch setting on shuffle-exchange networks and (b) The proposed switch adjusting on PSEN networks.

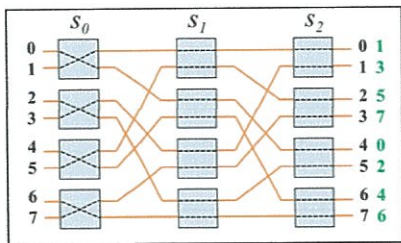


Fig. 6. Successful parallel point-to-point routing for $\rho = (1\ 3\ 5\ 7\ 0\ 2\ 4\ 6)$ on PSEN ($N = 8$) by using our switch-adjusting function.

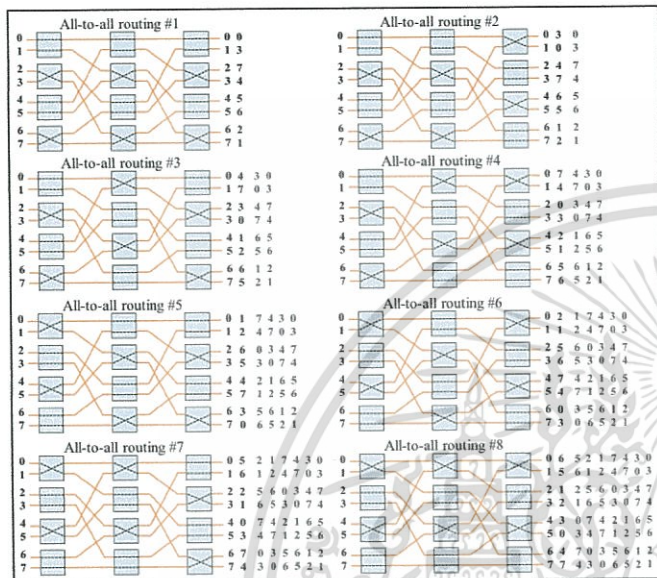


Fig. 7. A pattern of all-to-all personalized exchange on PSEN ($N = 8$).

TABLE III. SUCCESSFUL SWITCH ADJUSTING ON PSEN ($N = 8$) FOR PERMUTATION “13570246”

Source			Destination		(shift) S	Switch
<i>dec</i>	<i>bin</i>	<i>shifted</i>	<i>dec</i>	<i>bin</i>	XOR D	adjusting
0	000	000	4	100	100	x--
1	001	100	0	000	100	x--
2	010	001	5	101	100	x--
3	011	101	1	001	100	x--
4	100	010	6	110	100	x--
5	101	110	2	010	100	x--
6	110	011	7	111	100	x--
7	111	111	3	011	100	x--

B. Structure of Hierarchical Shuffle-Exchange Network

The hierarchical shuffle-exchange network (HSEN) is built by integrating the smaller subsystems into the larger system since the system reconstruction (Fig. 8 (a)) is higher cost and takes longer time than the subsystems integration (Fig. 8 (b)). For example, there are two existing PSENs ($N = 8$) that are available to be integrated into the larger network ($N = 16$) by connecting N ports (in the last stage), while the reconstructing new system ($N = 16$) have to reconnect $N \times \log_2 N$ ports (in all stages). Patterns $ISC_1 - ISC_{n-2}$ of the HSEN are the same as ISC applied in PSEN ($N/2 = 8$) plus the new ISC_{n-1} for connecting two subsystems of $N/2 (= 8)$ for $N (=16)$ with the perfect shuffle pattern (for all n bits in the last stage).

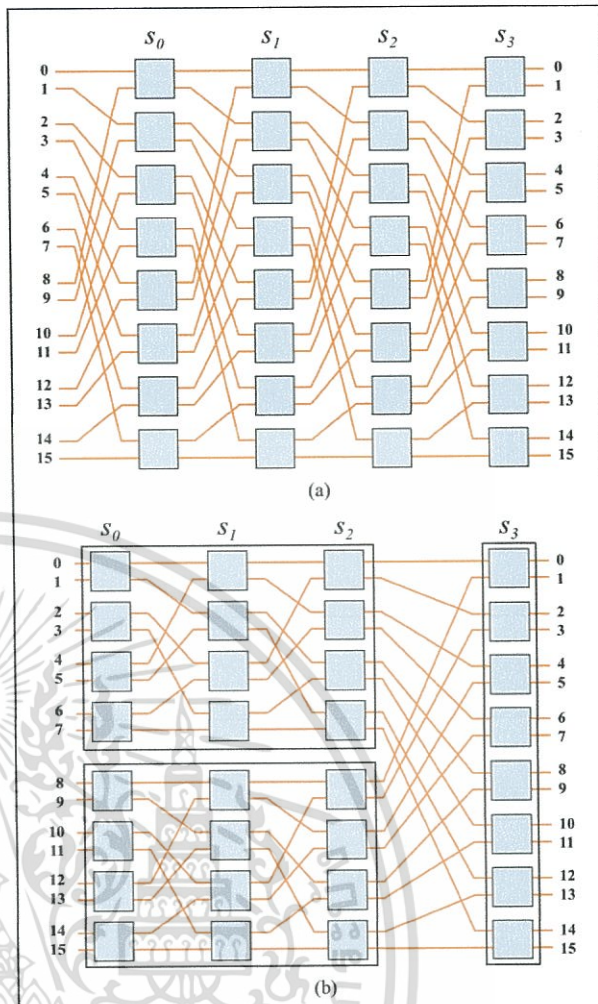


Fig. 8. Structure of two MINs ($N = 16$) using 2×2 switches: (a) Shuffle-exchange network (SEN) and (b) Hierarchical shuffle-exchange network (HSEN).

C. Switch Adjusting on HSENs

For HSEN networks, like PSEN networks, applying some all-to-all personalized exchange permutations may cause conflict in some switches and mismatch some pairs of S and D . Thus, for a successful parallel routing (between S and D), n -bit processor id of S are shifted two steps before adjusting switches by using XOR operation: 1) apply 1-bit circular right-shifting (for $(n-1)$ bits of its subsystem) and 2) apply another shift with 1-bit circular left-shifting (for n bits of the system), as follows:

$$\begin{aligned}
 & b_{n-1}(b_{n-2}b_{n-3} \dots b_2b_1b_0) \xrightarrow{\text{shift R}} b_{n-1}(b_0b_{n-2} \dots b_3b_2b_1) \\
 & (b_{n-1}b_0b_{n-2} \dots b_3b_2b_1) \xrightarrow{\text{shift L}} (b_0b_{n-2}b_{n-3} \dots b_2b_1b_{n-1})
 \end{aligned}$$

Fig. 9 illustrates the result of the proposed switch adjusting for self-routing on HSEN networks. Each switch is set according to the result of (shift) S XOR D , where (shift) $S = b_0b_{n-2}b_{n-3} \dots b_2b_1b_{n-1}$. In stage s_{n-i} (where $i = 1, 2, \dots, n-1, n$), except $i = 1$ and $i = n$, if b_{n-i} (of shifted S) = d_{n-i} (of D) then the corresponding switch is set to straight (-); otherwise set to exchange (x). Similar adjusting is performed in stage s_0 , where

$i = n$ (by using b_0 XOR d_{n-1}) and in stage s_{n-i} , where $i = 1$ (by using b_{n-1} XOR d_0).

For example, a permutation, $\rho = (0\ 8\ 11\ 3\ 7\ 15\ 12\ 4\ 5\ 13\ 14\ 6\ 2\ 10\ 9\ 1)$, is a realized parallel point-to-point permutation of HSEN ($N = 16$). Table IV defines results of our switch-adjusting function for adjusting switches on HSEs. After shifting S (as defined in Fig. 9 (block diagram)), n bits of (shift) S and n bits of D will be operated by using XOR operation for $\rho = (0\ 8\ 11\ 3\ 7\ 15\ 12\ 4\ 5\ 13\ 14\ 6\ 2\ 10\ 9\ 1)$ for successful parallel point-to-point routing, as shown in Fig. 10. Note that the results of all-to-all personalized exchange permutations, generated from that permutation, are also verified correctly.

TABLE IV. SUCCESSFUL SWITCH ADJUSTING ON HSEN ($N = 16$) FOR PERMUTATION “0 8 11 3 7 15 12 4 5 13 14 6 2 10 9 1”

Source			Destination		(shift) S XOR D	Switch adjusting
dec	bin	shifted	dec	bin		
0	0000	0000	0	0000	0000	----
1	0001	1000	15	1111	0111	-xxx
2	0010	0010	12	1100	1110	xxxx-
3	0011	1010	3	0011	1001	x--x
4	0100	0100	7	0111	0011	--xx
5	0101	1100	8	1000	0100	-x--
6	0110	0110	11	1011	1101	xx-x
7	0111	1110	4	0100	1010	x-x-
8	1000	0001	1	0001	0000	----
9	1001	1001	14	1110	0111	-xxx
10	1010	0011	13	1101	1110	xxxx-
11	1011	1011	2	0010	1001	x--x
12	1100	0101	6	0110	0011	--xx
13	1101	1101	9	1001	0100	-x--
14	1110	0111	10	1010	1101	xx-x
15	1111	1111	5	0101	1010	x-x-

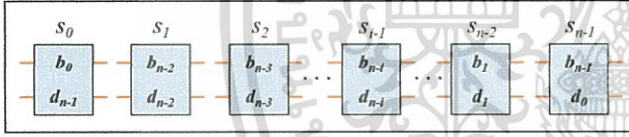


Fig. 9. The proposed switch adjusting on HSEN networks.

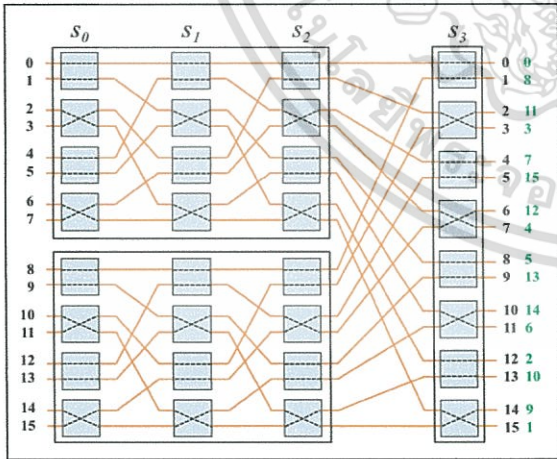


Fig. 10. Successful parallel point-to-point routing for $\rho = (0\ 8\ 11\ 3\ 7\ 15\ 12\ 4\ 5\ 13\ 14\ 6\ 2\ 10\ 9\ 1)$ on HSEN ($N = 18$) by using our switch-adjusting function.

IV. CORRECTNESS AND COMPLEXITY ANALYSIS

Correctness of the proposed switch-adjusting functions on PSEN and HSEN networks are proved in Section A and their time complexity analysis is provided in Section B.

A. Proof of Correctness

To simplify the proof, we will show correctness of self-routing on the shuffle-exchange network (SEN) first and then correctness of switch adjusting on our PSEN and HSEN networks in Theorem 1 - 3, respectively.

Let shuffle exchange-based MIN networks (i.e., SENs, PSENs, and HSENs) has N processor elements (PEs), $n = \log_2 N$ stages. Each processor id (from input ports to output ports) can be presented in binary as:

S : $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ be a (source) processor id .

D : $d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$ be a (destination) processor id .

Setting switches in each stage on SEN networks can be achieved by considering each pair (S and D) of n -bit processor id with XOR operation. For this MIN, a couple of the highest significant bits are considered first (in stage s_0), then the next (in stage s_1), and the last is the least significant bits (in stage s_{n-1}). After switch setting ($- / x$) in each stage, the corresponding binary bits are shifted by ISC (1-bit circular left-shifting). When that binary number (of S) passes through n switches and n stages alternately, the n -bit binary result will represent the (destination) processor id (D).

Theorem 1: On SEN networks, the (source) processor S : $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$, passing through n ISC connections and n switches (in n stages), is converted to the (destination) processor D : $d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$.

Proof. When the (input) binary number of S passes through each stage ($s_0 - s_{n-1}$), according to ISC (circular left-shift), and each switch is set ($- / x$) by using switch setting from S XOR D (see that pattern in Fig. 5 (a)) to connect from in -port to the corresponding out -port step-by-step, as follows:

$$\begin{aligned}
 in - s_0: & \quad b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 \xrightarrow{ISC} b_{n-2}b_{n-3} \dots b_1b_0b_{n-1} \\
 sw(in\ s_0): & \quad b_{n-2}b_{n-3} \dots b_1b_0b_{n-1} \xrightarrow{sw} b_{n-2}b_{n-3} \dots b_1b_0d_{n-1} \\
 s_0 - s_1: & \quad b_{n-2}b_{n-3} \dots b_1b_0d_{n-1} \xrightarrow{ISC} b_{n-3}b_{n-4} \dots b_0d_{n-1}b_{n-2} \\
 sw(in\ s_1): & \quad b_{n-3}b_{n-4} \dots b_0d_{n-1}b_{n-2} \xrightarrow{sw} b_{n-3}b_{n-4} \dots b_0d_{n-1}d_{n-2} \\
 & \quad \dots \\
 s_{n-2} - s_{n-1}: & \quad b_1b_0d_{n-1}d_{n-2} \dots d_3d_2d_1 \xrightarrow{ISC} b_0d_{n-1}d_{n-2} \dots d_3d_2b_1 \\
 sw(in\ s_{n-2}): & \quad b_0d_{n-1}d_{n-2} \dots d_3d_2b_1 \xrightarrow{sw} b_0d_{n-1}d_{n-2} \dots d_3d_2d_1 \\
 s_{n-1} - s_n: & \quad b_0d_{n-1}d_{n-2} \dots d_3d_2d_1 \xrightarrow{ISC} d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1b_0 \\
 sw(in\ s_{n-1}): & \quad d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1b_0 \xrightarrow{sw} d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0
 \end{aligned}$$

Thus, this proves that self-routing from S : $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ (at the input port) can be set to the correct D : $d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$ (at the output port) \square

Theorem 2: On PSEN networks with our switch adjusting, the (source) processor S : $b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$, passing through $n-1$ ISC connections and n switches (in n stages), is converted to the (destination) processor D : $d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$.

Proof. When the (input) binary S passes through each stage ($s_0 - s_{n-1}$), according to ISC (circular left-shift), and each switch is set ($- / x$) by using our switch adjusting from (shift) S

XOR D (see that pattern in Fig. 5 (b)) to connect from *in*-port to the corresponding *out*-port step-by-step, as follows:

$$\begin{aligned}
 sw(\text{in } s_0): & b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 \xrightarrow{sw} b_{n-1}b_{n-2} \dots b_2b_1d_{n-1} \\
 s_0 - s_1: & b_{n-1}b_{n-2} \dots b_2b_1d_{n-1} \xrightarrow{ISC} b_{n-2}b_{n-3} \dots b_1d_{n-1}b_{n-1} \\
 sw(\text{in } s_1): & b_{n-2}b_{n-3} \dots b_1d_{n-1}b_{n-1} \xrightarrow{sw} b_{n-2}b_{n-3} \dots b_1d_{n-1}d_{n-2} \\
 s_1 - s_2: & b_{n-2}b_{n-3} \dots b_1d_{n-1}d_{n-2} \xrightarrow{ISC} b_{n-3}b_{n-4} \dots d_{n-1}d_{n-2}b_{n-2} \\
 & \dots \\
 s_{n-3} - s_{n-2}: & b_2b_1d_{n-1}d_{n-2} \dots d_3d_2 \xrightarrow{ISC} b_1d_{n-1}d_{n-2} \dots d_3d_2b_2 \\
 sw(\text{in } s_{n-2}): & b_1d_{n-1}d_{n-2} \dots d_3d_2b_2 \xrightarrow{sw} b_1d_{n-1}d_{n-2} \dots d_3d_2d_1 \\
 s_{n-2} - s_{n-1}: & b_1d_{n-1}d_{n-2} \dots d_3d_2d_1 \xrightarrow{ISC} d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1b_1 \\
 sw(\text{in } s_{n-1}): & d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1b_1 \xrightarrow{sw} d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0
 \end{aligned}$$

Clearly, the last passing gives $d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$, which is exactly the same as the n -bit (destination) processor *id*. Therefore, this proves that our switch-adjusting function on PSEN networks can achieve the correct point-to-point data routing from S to D \square

Theorem 3: On HSEN networks with our switch adjusting, the (source) processor $S: b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$, passing through $n-1$ ISC connections and n switches (in n stages), is converted to the (destination) processor $D: d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$.

Proof. The HSEN network uses a different switch-adjusting function from that of the PSEN, as defined in Section III.C. The HSEN network (N processors and $n = \log_2 N$ stages) consists of two PSEs ($N/2$ processors and $\log_2 N/2 = n-1$ stages). When the (input) binary number passes through each stage ($s_0 - s_{n-2}$) for subsystems ($N/2 = 2^{n-1}$), according to $ISC_1 - ISC_{n-2}$ (1-bit circular left-shift for $n-1$ bits) to connect to the larger system ($N = 2^n$) by using last ISC_{n-1} (1-bit circular left-shift for n bits), and each particular switch is set ($- / x$) by using switch adjusting from (shift) S XOR D (see that pattern in Fig. 9) to connect from *in*-port to the corresponding *out*-port, as follows:

$$\begin{aligned}
 sw(\text{in } s_0): & b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0 \xrightarrow{sw} b_{n-1}b_{n-2} \dots b_2b_1d_{n-1} \\
 s_0 - s_1: & b_{n-1}(b_{n-2}b_{n-3} \dots b_2b_1d_{n-1}) \xrightarrow{ISC_1} b_{n-1}(b_{n-3} \dots b_1d_{n-1}b_{n-2}) \\
 sw(\text{in } s_1): & b_{n-1}(b_{n-3} \dots b_1d_{n-1}b_{n-2}) \xrightarrow{sw} b_{n-1}(b_{n-3} \dots b_1d_{n-1}d_{n-2}) \\
 s_1 - s_2: & b_{n-1}(b_{n-3} \dots b_1d_{n-1}d_{n-2}) \xrightarrow{ISC_2} b_{n-1}(b_{n-4} \dots d_{n-1}d_{n-2}b_{n-3}) \\
 & \dots \\
 s_{n-3} - s_{n-2}: & b_{n-1}(b_1d_{n-1} \dots d_3d_2d_1) \xrightarrow{ISC_{n-1}} b_{n-1}(d_{n-1}d_{n-2} \dots d_3d_2b_1) \\
 sw(\text{in } s_{n-2}): & b_{n-1}(d_{n-1}d_{n-2} \dots d_3d_2b_1) \xrightarrow{sw} b_{n-1}(d_{n-1}d_{n-2} \dots d_3d_2d_1) \\
 s_{n-2} - s_{n-1}: & (b_{n-1}d_{n-1}d_{n-2} \dots d_3d_2d_1) \xrightarrow{ISC_n} (d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1b_{n-1}) \\
 sw(\text{in } s_{n-1}): & (d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1b_{n-1}) \xrightarrow{sw} (d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0)
 \end{aligned}$$

Applying the proposed switch adjusting yields $d_{n-1}d_{n-2}d_{n-3} \dots d_2d_1d_0$, which is exactly the same as the n -bit (destination) processor *id*. Thus, this proves that our switch-adjusting function on HSEN networks connects to the correct point-to-point data routing from S to D \square

B. Time Complexity Analysis

Time complexity of our switch-adjusting functions on PSEN and HSEN networks are analyzed in Lemma 1 - 2.

Lemma 1: Time complexity of applying our switch-adjusting function on PSEN networks for N PEs (in parallel) is $O(1)$.

Proof. The switch-adjusting function on PSEs is based on 1-bit circular right-shifting (of $S: b_{n-1}b_{n-2}b_{n-3} \dots b_2b_1b_0$ in n -bit register). The shifting process can be performed by shift logic in CPU in one time unit and hence in parallel (N PEs), all N shift logic are shifted in one time unit. Therefore, time complexity of applying the switch-adjusting function on PSEs is $O(1)$ \square

Lemma 2: Time complexity of applying our switch-adjusting function on HSEN networks for N PEs (in parallel) is $O(1)$.

Proof. The switch-adjusting function on HSEs is based on two operations, which are 1-bit circular right-shifting (of the first $(n-1)$ bits of $S: b_{n-1}(b_{n-2}b_{n-3} \dots b_2b_1b_0)$) and 1-bit circular left-shifting (of n -bit result from the previous shift). Each shifting process can be performed by shift logic in CPU in one time unit only, and hence two time units required for that switch adjusting. In parallel (N PEs), all N shift logic are shifted in two time units. Clearly, time complexity of applying the switch-adjusting function on HSEs is $O(1)$ \square

V. CONCLUSION

This paper proposes the switch-adjusting function on hierarchical shuffle-exchange networks (HSEs) for applying all-to-all personalized exchange, a class of all-to-all communication which frequently used in parallel and distributed systems. Such a hierarchical MIN architecture is constructed in reasonable cost and time, which is suitable for scalable systems in real parallel engines. However, the proposed hierarchical design without switch adjusting before self-routing cannot apply some parallel point-to-point permutations and existing all-to-all personalized exchange algorithms. Therefore, by proof of correctness, we can verify that our hierarchical HSEN networks along with proposed switch-adjusting function perform the correct self-routing for all-to-all personalized exchange permutations. In our future work, other hierarchical MIN networks with appropriate switch-adjusting functions will be studied for successful all-to-all personalized exchange permutations.

REFERENCES

- [1] Y. Yang and J. Wang, "Optimal all-to-all personalized exchange in self-routable multistage networks," IEEE Trans. Parallel and Distributed System, vol. 11, no. 3, pp. 261-274, March 2000.
- [2] Y. Yang and J. Wang, "Optimal all-to-all personalized exchange in a class of optical multistage networks," IEEE Trans. Parallel and Distributed System, vol. 12, no. 6, pp. 567-582, June 2001.
- [3] A. Massini, "All-to-all personalized communication on multistage network," Discrete Appl. Math. 128 (2), 2003, pp. 435-446.
- [4] V.W. Liu, C. Chen, and R.B. Chen, "Optimal all-to-all personalized exchange in d -nary banyan multistage networks," J. Comb. Optim., vol. 14, pp. 131-142, March 2007.
- [5] W.Y. Chou and C. Chen, "All-to-all personalized exchange in generalized shuffle-exchange networks," J. Theoretical Computer Science, vol. 411, pp. 1669-1684, 2010.
- [6] C-L. Wu and T-Y. Feng, "On a class of multistage interconnection networks," IEEE Trans. Computers, vol. c-29, no. 8, pp. 694-702, August 1980.
- [7] N.A.M. Yunus and M. Othman, "Shuffle exchange network in multistage interconnection network: a review and challenges," International Journal of Computer and Electrical Engineering, vol. 3, no. 5, pp. 724-728, October 2011.

ประวัติผู้เขียน

ชื่อ – สกุล นางสาวปาริฉัตร นาคสิทธิ์
 วัน เดือน ปีเกิด 5 พฤษภาคม 2531
 ที่อยู่ 66/2 หมู่ที่ 4 ต.บ้านหลวง อ.คอนท่อม จ.นครปฐม 73150

ประวัติการศึกษา

2552 วิทยาศาสตร์บัณฑิต เกียรตินิยมอันดับสอง สาขาวิทยาการคอมพิวเตอร์
 คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้