

การปรับปรุงประสิทธิภาพการจัดวางและเชื่อมโยงของเซลเบส  
CELL-BASED PLACEMENT AND ROUTING OPTIMIZATION



นายทรงพล ใหม่สาลี  
MR. SONGPON MAISALEE

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2540

ISBN 974-622-072-1

เลขที่.....

เลขทะเบียน..... 29394

วัน, เดือน, ปี 26 ส.ค. 2541

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Cell-Based Placement and Routing Optimization
Student	Mr. Songpon Maisalee
Thesis Advisor	Mr. Banjong Piyathamrong
Level of Study	Master of Science in Computer Science and Information Technology King Mongkut's Institute of Technology Ladkrabang
Year	1997

### ABSTRACT

The development of microelectronics is an important factor in improving quality of life. Most of electrical equipments have integrated circuit as important component. A standard cell is one type of Application Specification Integrated Circuits or ASIC. Its physical layout design methodology is semi-custom design or cell-based design. The layout of the design consists of fixed cells, which were physically designed, placed in rows and routed. Both of placement and routing complexity are NP-Complete problem. In the past solving of physical layout design focused solely on minimizing the size of layout. Consequently, the resulted layout had inefficient performance. To accommodate the problem, this research work applies an hierarchical placement methodology and considers factors which impact the placement and routing, in order to assign an appropriate cost function under the simulated annealing algorithm. The designed layout is expected to work properly and meet the desired requirement.

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ดีด้วยความกรุณาจากท่านอาจารย์ บรรจง ปิยะธำรง ที่ได้ให้คำแนะนำและชี้แนะทางที่เป็นประโยชน์ในการแก้ปัญหาการทำวิจัยตลอดระยะเวลาที่ทำการศึกษาอยู่ ผู้เขียนขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้ ขอขอบคุณทุนอุดหนุนการทำวิทยานิพนธ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และคุณอัญชลี จันทร์นวล ที่ให้ความช่วยเหลือและเป็นกำลังใจจนสำเร็จลุล่วงไปด้วยดี



## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่	
1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	3
1.3 ทฤษฎีที่เกี่ยวข้องกับการวิจัย.....	4
1.4 ขอบเขตของการวิจัย.....	6
1.5 วิธีที่ใช้ในการดำเนินการวิจัย.....	8
1.6 ประโยชน์ที่จะได้รับ.....	8
2 การออกแบบวงจรรวมในระดับกายภาพของวงจรรวมแสดนคาร์ดเซล.....	9
2.1 คุณลักษณะเฉพาะการออกแบบระดับกายภาพของวงจรรวม แสดนคาร์ดเซล.....	9
2.2 ประเภทของเซลล์ที่ใช้ในวงจรรวมแสดนคาร์ดเซล.....	10
2.3 กฎการออกแบบ (Design Rules).....	11
2.4 ลักษณะทางกายภาพวงจรรวมแสดนคาร์ดเซล.....	12
2.5 ขั้นตอนการออกแบบวงจรรวมในระดับกายภาพ.....	13
2.6 ปัญหาการออกแบบทางกายภาพ.....	14
3 สิมูเลทเตดอัลเนลลิ่งอัลกอริทึม.....	17
3.1 ปัญหาทางเลือก (Combinatorial Optimization Problem).....	17
3.2 สิมูเลทเตดอัลเนลลิ่งอัลกอริทึม.....	17
3.3 ปัญหาการเดินทางของพนักงานขาย (TSP).....	21

## สารบัญ (ต่อ)

บทที่	หน้า
4 การจัดวาง .....	23
4.1 การออกแบบวงจรรวมแบบลำดับชั้น (Hierarchical Design).....	23
4.2 การจัดวางแบบลำดับชั้น (Hierarchical Placement).....	24
4.3 การคำนวณพื้นที่โดยประมาณ (Area Estimation).....	29
4.4 การจัดกลุ่มเซลล์ (Cell Clustering).....	31
4.5 การจัดกลุ่มเซลล์โดยใช้สัญญาณเน็ตลิ่งอัลกอริทึม.....	33
4.6 การวางผังเซลล์ (Cell Floorplanning).....	34
4.7 การวางผังเซลล์โดยใช้สัญญาณเน็ตลิ่งอัลกอริทึม.....	35
4.8 การวางเซลล์ (Cell Final Placement).....	36
4.9 การวางเซลล์โดยใช้สัญญาณเน็ตลิ่งอัลกอริทึม.....	37
5 การเชื่อมโยง.....	38
5.1 การกำหนดช่องทางเดินของสายสัญญาณ (Global Routing).....	38
5.2 การกำหนดทางเดินภายในช่องทาง (Detailed Routing).....	41
6 ทดสอบกับ MCNC Benchmark.....	43
6.1 MCNC Benchmark.....	43
7 บทสรุป.....	53
7.1 สรุปผลการวิจัย.....	53
7.2 ปัญหาที่เกิดขึ้นและข้อเสนอแนะ.....	54
เอกสารอ้างอิง.....	55
ภาคผนวก .....	58
ภาคผนวก ก. ผลงานวิจัยที่ได้รับการตีพิมพ์.....	59
ภาคผนวก ข. โปรแกรมการทดลอง.....	60
ภาคผนวก ค. MCNC Benchmark.....	97
ประวัติผู้เขียน.....	109

## สารบัญตาราง

	หน้า
6.1 วงจรทดสอบ.....	43
6.2 แสดงการเปรียบเทียบการจัดวางแบบ Hierarchical Design กับ Flat Design ของวงจร SC0.....	51
6.3 แสดงการเปรียบเทียบการจัดวางแบบ Hierarchical Design กับ Flat Design ของวงจร SC2.....	52



## สารบัญภาพ

	หน้า
1.1 แสดงการจำแนกประเภทของวงจรรวม.....	2
1.2 แสดงขั้นตอนการออกแบบวงจรรวม.....	3
1.3 แสดงกราฟแบบมีทิศทางและกราฟไม่มีทิศทาง.....	5
1.4 แสดงต้นไม้หรือ Tree.....	6
2.1 แสดงขั้นตอนการออกแบบระดับกายภาพของวงจรรวมแสดงนคาร์ดเซล.....	9
2.2 แสดงลักษณะทางกายภาพของฟิคเซล.....	10
2.3 แสดงกฎการออกแบบ.....	12
2.4 (a) แสดงกายภาพของแสดงนคาร์ดเซล.....	12
2.4 (b) แสดงกายภาพของแมคโครเซล.....	12
2.5 แสดงกายภาพของแสดงนคาร์ดเซลในลักษณะแถว.....	13
2.6 แสดงขั้นตอนการออกแบบในระดับกายภาพ.....	14
2.7 แสดงการจัดวางเซลลงบนเลเอาต์.....	15
3.1 แสดงอัลกอริทึม Simulated_Annealing().....	18
3.2 แสดงอัลกอริทึม Accept( $\Delta C_T$ ).....	19
3.3 กราฟแสดงฟังก์ชันเป้าหมายที่ได้จากสิมูลเลตเตดอัลเนลลิงอัลกอริทึม.....	20
4.1 การออกแบบวงจรรวมแบบลำดับชั้น.....	23
4.2 แสดงการจัดวางแบบลำดับชั้น.....	24
4.3 แสดงการใช้ฟิคเซล.....	25
4.4 แสดงคลิกที่คัลพารภายในวงจร.....	26
4.5 แสดง Half Perimeter ของสายสัญญาณ.....	27
4.6 แสดงการเชื่อมโยงระหว่างเซล.....	28
4.7 แสดงพื้นที่ของเลเอาต์.....	29
4.8 แสดงการจัดวางเซลภายในแถว.....	30
4.9 แสดงอัลกอริทึม Estimate_Area().....	30
4.10 แสดงการแบ่งกลุ่มโดยใช้คัทไลน์.....	31
4.11 แสดงการแบ่งกลุ่มในลักษณะ M-Way.....	32
4.12 แสดงการใช้คัทไลน์และคัทฟีด.....	33

## สารบัญญภาพ (ต่อ)

	หน้า
4.13 แสดงการวางผังเซล.....	35
4.14 แสดงการจัดวางเซล.....	36
5.1 แสดงแกนเน็ตที่ใช้ในการเชื่อมโยง.....	38
5.2 แสดงอัลกอริทึม Global_Routing().....	39
5.3 แสดง Multi-Point Net, MST และ MRST.....	39
5.4 แสดง Switchable Segment.....	40
5.5 แสดงการเชื่อมโยงสายสัญญาณ.....	41
5.6 แสดงผลลัพธ์ที่ได้จากการเชื่อมโยง.....	41
5.7 แสดงอัลกอริทึม Greedy_Channel_Route().....	42
5.8 แสดงการทำงานของ Greedy Route.....	42
6.1 แสดงขั้นตอนภายในโปรแกรม Automatic Place and Route.....	44
6.2 แสดงข้อมูลในไลบรารี.....	44
6.3 แสดงการคำนวณพื้นที่โดยประมาณ.....	45
6.4 (a) แสดงการกำหนดจำนวนแถวของเซล.....	46
6.4 (b) แสดงการกำหนดค่าอุณหภูมิเริ่มต้นและจำนวนทรานซิสต์.....	46
6.5 แสดงกราฟฟังก์ชันเป้าหมายของสิมูเลตเตดอัลเน็ตลิ่งอัลกอริทึม.....	47
6.6 (a) แสดงการจัดวางก่อนการแบ่งกลุ่ม.....	47
6.6 (b) แสดงการจัดวางหลังการแบ่งกลุ่ม.....	48
6.7 แสดงการวางผัง.....	48
6.8 (a) แสดงการจัดวาง.....	49
6.8 (b) แสดงการจัดวางพร้อมคัทไฟด์.....	49
6.9 แสดงการกำหนดช่องทาง.....	50
6.10 แสดงการเชื่อมโยงภายในแกนเน็ต.....	50
6.11 กราฟแสดงการเปรียบเทียบการจัดวางแบบ Hierarchical Design กับ Flat Design ของวงจร SC0.....	51
6.12 กราฟแสดงการเปรียบเทียบการจัดวางแบบ Hierarchical Design กับ Flat Design ของวงจร SC2.....	52

# บทที่ 1

## บทนำ

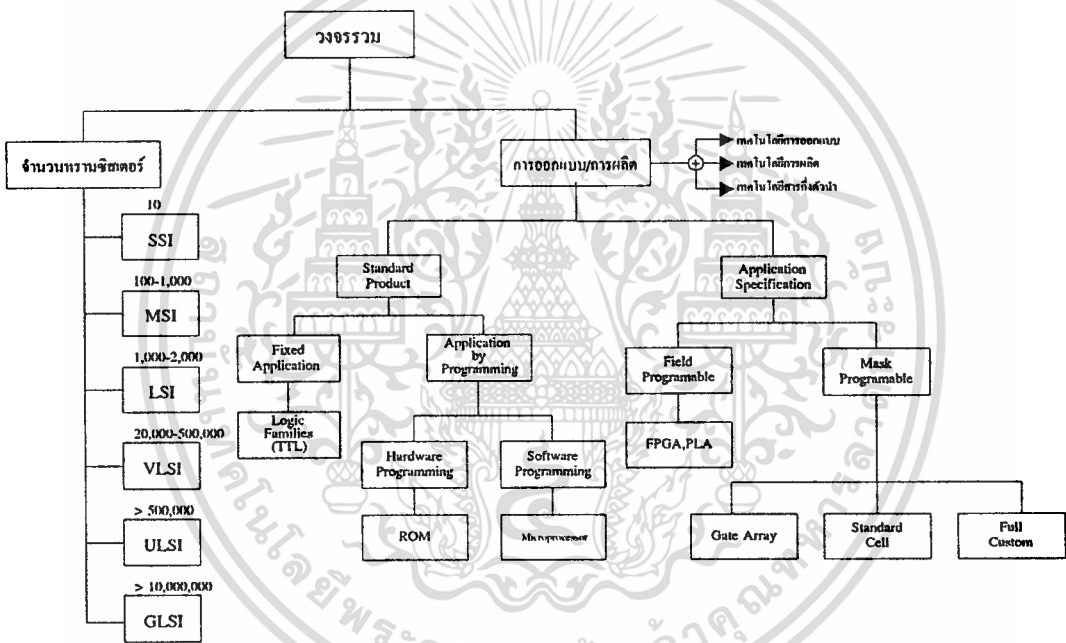
### 1.1 ความเป็นมาและความสำคัญของปัญหา

ในปีพ.ศ. 2490 นักฟิสิกส์ที่สถาบันวิจัยเบล ประเทศสหรัฐอเมริกา สามารถประดิษฐ์ทรานซิสเตอร์ (Transistor) จากสารกึ่งตัวนำได้สำเร็จ นับได้ว่าเป็นจุดเริ่มต้นของการพัฒนาเทคโนโลยีไมโครอิเล็กทรอนิกส์และสารกึ่งตัวนำ จนกระทั่งในปีพ.ศ. 2500 ที่บริษัท Texas Instrument [1] สามารถประดิษฐ์วงจรรวม (IC : Integrated Circuit) ขึ้นมาใช้งานได้เป็นผลสำเร็จ โครงสร้างภายในวงจรรวมเกิดจากการนำทรานซิสเตอร์หลาย ๆ ตัวมาเชื่อมต่อกันเป็นวงจรรวมขนาดเล็กบรรจุอยู่บนแผ่นซิลิคอน วิวัฒนาการของวงจรรวมเริ่มจากวงจรรวมขนาดเล็ก หรือ SSI (Small Scale IC) ภายในวงจรประกอบด้วยทรานซิสเตอร์ไม่กี่สิบตัว แต่ในปัจจุบันเป็นยุคของวงจรรวมขนาดใหญ่มากหรือ ULSI (Ultra Large Scale IC) ประกอบด้วยทรานซิสเตอร์ไม่ต่ำกว่าสิบล้านตัว อัตราความหนาแน่นของทรานซิสเตอร์ภายในวงจรรวมมีปริมาณมากกว่าหลายเท่าเมื่อเทียบกับวงจรรวมในยุคแรก

ไม่เพียงแต่วิทยุ โทรทัศน์ และเครื่องคอมพิวเตอร์เท่านั้นที่มีวงจรรวมเป็นส่วนประกอบที่สำคัญ อุปกรณ์เครื่องใช้ไฟฟ้าเกือบทุกประเภทประกอบด้วยวงจรรวมเป็นอุปกรณ์พื้นฐานที่สำคัญเช่นกัน เป็นการชี้ให้เห็นว่าวงจรรวมได้เข้ามามีอิทธิพลต่อความเป็นอยู่ของมนุษย์อย่างสมบูรณ์ การประยุกต์วงจรรวมไปใช้ในงานต่าง ๆ เป็นไปอย่างกว้างขวาง ขึ้นอยู่กับวัตถุประสงค์การใช้งาน วงจรรวมแต่ละประเภทที่ผลิตออกมาจึงมีความเหมาะสมในการใช้งานต่างกัน กระบวนการออกแบบและกระบวนการผลิตมีความซับซ้อนแตกต่างกัน เช่น วงจรรวมมาตรฐาน (Standard Product) ที่มีจำหน่ายในท้องตลาดทั่วไปเป็นวงจรรวมขนาดเล็ก การออกแบบไม่มีความซับซ้อนซึ่งแตกต่างจากวงจรรวมเฉพาะกิจ (ASIC : Application Specification Integrated Circuit) ที่ต้องการความซับซ้อนในการออกแบบ เนื่องจากความซับซ้อนของวงจร เช่น วงจรรวมที่ใช้ในการประมวลผลภาพ เป็นต้น แนวโน้มการผลิตวงจรรวมเฉพาะกิจนับวันเพิ่มมากขึ้น เนื่องจากสามารถเลือกวิธีการออกแบบได้หลายวิธี เช่น FPGA, Gate Array, Standard Cell และ Full Custom ดังแสดงในรูปที่ 1.1 [2] [3] และสามารถทำต้นแบบในการผลิตได้รวดเร็ว ต้นทุนในการผลิตต่ำ แม้ปริมาณการผลิตน้อย การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลิตวงจรรวมเฉพาะกิจถูกแบ่งออกเป็นขั้นตอนที่สำคัญ 2 ขั้นตอนคือ ขั้นตอนการออกแบบ (Designed) และขั้นตอนการเจือสาร (Fabrication) ดังแสดงในรูปที่ 1.2 [4] โดยเฉพาะในการออกแบบวงจรรวมในระดับกายภาพของเลเอาท์ (Physical Layout Design) ซึ่งเดิมออกแบบด้วยมือต้องใช้บุคลากรและเวลาเป็นจำนวนมาก ไม่สามารถรองรับกับงานที่ซับซ้อนได้จึงเปลี่ยนมาใช้คอมพิวเตอร์ช่วยออกแบบแทน เครื่องคอมพิวเตอร์ที่ใช้ในการออกแบบส่วนใหญ่เป็นเครื่องในระดับเวคสเตชัน (Workstation) ซอฟต์แวร์ (Software) ช่วยออกแบบมีราคาสูงโอกาสการศึกษาทดลองออกแบบวงจรรวมเป็นไปได้ลำบาก ไม่แพร่หลายต้องใช้ซอฟต์แวร์ช่วยออกแบบจากต่างประเทศ



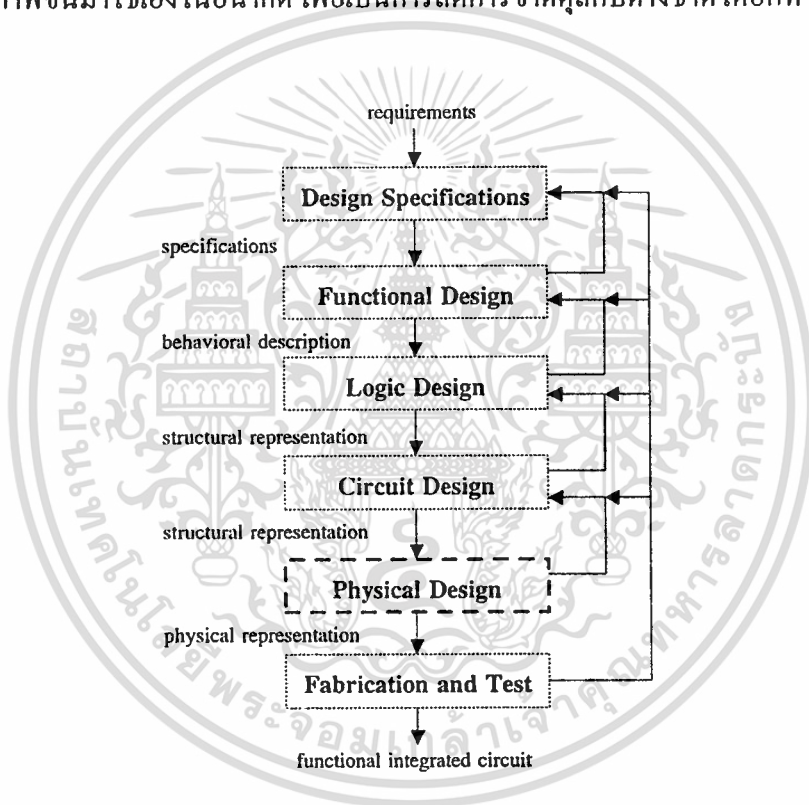
รูปที่ 1.1 แสดงการจำแนกประเภทของวงจรรวม

ปัจจุบันการศึกษาพัฒนาซอฟต์แวร์ช่วยออกแบบวงจรรวมภายในประเทศ มีผู้วิจัยค้นคว้าค่อนข้างน้อย เมื่อเปรียบเทียบกับประเทศอุตสาหกรรมอย่างเช่น ไต้หวัน ญี่ปุ่น เป็นต้น ในประเทศไทยนั้นช่วงเวลาที่ผ่านมามีส่วนใหญ่เป็นการพัฒนาซอฟต์แวร์ช่วยออกแบบในระดับกายภาพ (Layout Editor) ที่ใช้วิธีการออกแบบฟูลคัสตอม (Full Custom) [5] [6] ผู้ออกแบบต้องมีความรู้ทางด้านสารกึ่งตัวนำและเทคนิคการออกแบบทรานซิสเตอร์

สำหรับการออกแบบวงจรรวมเสตนเซลล์ (Standard Cell) หรือเซลล์เบสดีไซน์ (Cell-Based Design) [7] [8] [9] เป็นการออกแบบวงจรรวมที่ไม่ต้องอาศัยความรู้ทางด้านเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารกึ่งตัวนำและเทคนิคการออกแบบทรานซิสเตอร์เหมือนการออกแบบวงจรรวมแบบฟูลคัสตอม เนื่องจากการออกแบบวงจรรวมแบบแสดนคาร์ดเซลเป็นการออกแบบวงจรรวมโดยอาศัยโมดูลหรือเซลล์ที่มีการออกแบบไว้แล้วในไลบรารี (Library) มาใช้งาน

ด้วยเหตุผลข้างต้น ผู้วิจัยได้ศึกษาขั้นตอนและอัลกอริทึมที่ใช้ในการออกแบบวงจรรวมในระดับกายภาพสำหรับวงจรรวมแสดนคาร์ดเซล (Standard Cell Physical Layout) ซึ่งประกอบด้วยการจัดวาง (Placement) และการเชื่อมโยง (Routing) เพื่อเป็นแนวทางในการพัฒนาซอฟต์แวร์ช่วยออกแบบในระดับกายภาพของเลเอาท์สำหรับวงจรรวมแสดนคาร์ดเซลที่มีประสิทธิภาพขึ้นมาใช้เองในอนาคต เพื่อเป็นการลดการขาดดุลกับต่างชาติได้อีกทางหนึ่ง



รูปที่ 1.2 แสดงขั้นตอนการออกแบบวงจรรวม

## 1.2 วัตถุประสงค์ของการวิจัย

การออกแบบระดับกายภาพวงจรรวมแสดนคาร์ดเซล แบ่งขั้นตอนการออกแบบเป็น 2 ขั้นตอนคือ การจัดวางและการเชื่อมโยง ขั้นตอนการจัดวางเป็นการวางเซลล์ลงบนเลเอาท์ (Layout) ส่วนขั้นตอนการเชื่อมโยงเป็นการเชื่อมสายสัญญาณระหว่างเซลล์ที่ได้ภายหลังจากการจัดวาง การออกแบบระดับกายภาพวงจรรวมแสดนคาร์ดเซลที่มีจำนวนเซลล์ไม่มาก เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถออกแบบด้วยมือแทนได้ แต่สำหรับการออกแบบที่มีจำนวนเซลล์มาก ๆ จำเป็นต้องใช้เครื่องคอมพิวเตอร์เข้ามาช่วย เนื่องจากการออกแบบด้วยวิธีเดิมโดยใช้มือไม่สามารถรองรับความซับซ้อนของงานได้

ถึงแม้ปัจจุบันได้มีการนำคอมพิวเตอร์มาช่วยในการออกแบบระดับกายภาพแล้วก็ตาม เลเอาท์ที่ได้จากการออกแบบด้วยคอมพิวเตอร์ไม่สามารถกล่าวได้ว่าเป็นเลเอาท์ที่ดีที่สุด ทั้งนี้เนื่องจากจำนวนผลลัพธ์ที่เป็นไปได้ (Solution Space) มีเป็นจำนวนมากไม่สามารถตรวจสอบทุกคำตอบได้ภายในเวลาพหุนาม (Polynomial) ดังนั้นการแก้ปัญหาคาดวงและการเชื่อมต่อ จึงเป็นการหาผลลัพธ์ที่ดีที่สุดภายใต้เงื่อนไขที่กำหนดเท่านั้น

ปัญหาคาดวงและการเชื่อมต่อจัดเป็นปัญหาประเภทเอ็นพีคัมพลีท (NP-Complete : Non-Deterministic Polynomial Complete) [10] ปัญหาประเภทนี้ไม่มีอัลกอริทึม (Algorithm) ที่ใช้แก้ปัญหาได้โดยตรงภายในเวลาพหุนาม ต้องใช้วิธีการแปลงไปเป็นปัญหาอื่นที่สามารถแก้ได้ด้วยอัลกอริทึมที่ใช้เวลาพหุนามมาแทน คำตอบที่ได้จากปัญหาใหม่เปรียบเสมือนเป็นคำตอบของปัญหาเดิม

การแก้ปัญหาคาดวงและการเชื่อมต่อมักจะใช้อัลกอริทึม (Heuristic Algorithms) เป็นหลัก มุ่งเน้นการใช้เนื้อที่ของเลเอาท์ให้น้อยที่สุดเพียงวัตถุประสงค์เดียว มิได้คำนึงถึงวัตถุประสงค์อื่น ๆ เช่น ประสิทธิภาพทางไฟฟ้า ทำให้เลเอาท์ที่ได้ไม่สามารถใช้งานจริงได้ ดังนั้น ในการวิจัยนี้จึงได้พิจารณาวัตถุประสงค์ 5 ประการ ซึ่งประกอบด้วย Aspect Ratio, การใช้เนื้อที่ของเลเอาท์, ประสิทธิภาพทางไฟฟ้า, การเชื่อมต่อที่สมบูรณ์, เวลาที่ใช้ในการแก้ปัญหาและจำนวนเซลล์ไปพร้อม ๆ กัน เพื่อกำหนดฟังก์ชันเป้าหมายที่เหมาะสมสำหรับใช้ในการแก้ปัญหาคาดวงและการเชื่อมต่อ ซึ่งจะส่งผลให้เลเอาท์ที่ได้จากการแก้ปัญหาเป็นไปตามวัตถุประสงค์ของผู้ ออกแบบและมีประสิทธิภาพต่อการใช้งานจริง

### 1.3 ทฤษฎีที่เกี่ยวข้องกับการวิจัย

#### 1.3.1 คณิตศาสตร์การจัดลำดับและการจัดหมู่ (Combinatorial Mathematic)

##### 1.3.1.1 หลักการนับ (Principles of Counting)

ถ้าในการทำกิจกรรมใด ๆ แยกออกเป็น  $k$  ขั้นตอน โดยขั้นตอนแรกสามารถเลือกทำได้  $m_1$  วิธี และในขั้นตอนที่สองเลือกทำได้  $m_2$  วิธี และในขั้นตอนที่  $k$  สามารถเลือกทำได้  $m_k$  วิธี ดังนั้นจำนวนวิธีทั้งหมดที่ทำกิจกรรมนั้นได้เสร็จสิ้นเท่ากับ  $m_1, m_2, m_3, \dots, m_k$  วิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.3.1.2 การจัดลำดับ (Permutation)

ถ้ามีสิ่งของอยู่  $n$  สิ่ง ซึ่งแต่ละสิ่งมีลักษณะแตกต่างกัน การจัดลำดับ หมายถึง การนำสิ่งของทั้ง  $n$  สิ่งหรือ  $r$  สิ่ง ( $r \leq n$ ) มาจัดเรียงลำดับโดยถือลำดับเป็นสำคัญ จะได้  $\frac{n!}{(n-r)!}$  วิธี หรือ  $P_{n,r} = \frac{n!}{(n-r)!}$  วิธี

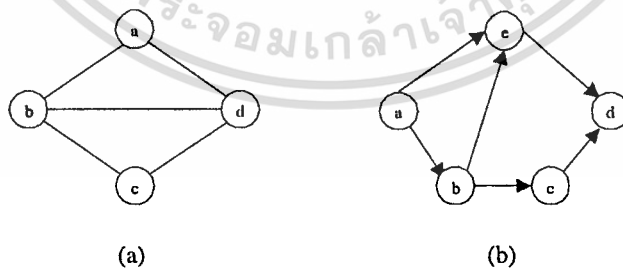
### 1.3.1.3 การจัดหมู่ (Combination)

ถ้ามีสิ่งของอยู่  $n$  สิ่ง ซึ่งแต่ละสิ่งมีลักษณะแตกต่างกัน การจัดหมู่ หมายถึงการนำสิ่งของ  $r$  สิ่ง ( $r < n$ ) มาจัดเป็นกลุ่มหรือหมู่ โดยไม่คำนึงถึงลำดับ แต่เน้นกลุ่มเป็นสำคัญจะได้  $\frac{n!}{(n-r)!r!}$  วิธีหรือ

$$C_{n,r} = \binom{n}{r} = \frac{n!}{(n-r)!r!} \text{ วิธี}$$

## 1.3.2 ทฤษฎีกราฟ (Graph Theory)

กำหนดให้กราฟ  $G = (V,E)$  ประกอบด้วยคู่ลำดับของเซต  $(V,E)$  โดยที่  $V$  คือ เซตของจุด (Vertex) ต่าง ๆ และ  $E$  คือเซตของเส้นเชื่อม (Edge) ระหว่างคู่ของจุดในกราฟ ตัวอย่างเช่น กราฟ  $G = (V,E)$  ในรูปที่ 1.3 (a) มี  $V = \{a,b,c,d\}$  และ  $E = \{ \{a,b\}, \{b,c\}, \{b,d\}, \{a,d\}, \{c,d\} \}$  เป็นกราฟที่ไม่มีทิศทาง (Undirected Graph) แทนเส้นเชื่อมของจุดด้วยเซตของคู่จุด ลำดับที่ของจุดไม่มีผลต่อกราฟ มีกราฟอีกประเภทหนึ่งเรียกว่ากราฟมีทิศทาง (Directed Graph) เช่นกราฟในรูปที่ 1.3 (b) จะสังเกตว่ามีลูกศรกำกับทิศทางของเส้นเชื่อมแต่ละเส้นในกราฟ ในกรณีนี้เราจะใช้คู่ลำดับของจุดแทนเส้นเชื่อมมีทิศทางเช่น (b,c)



รูปที่ 1.3 แสดงกราฟแบบมีทิศทางและกราฟไม่มีทิศทาง

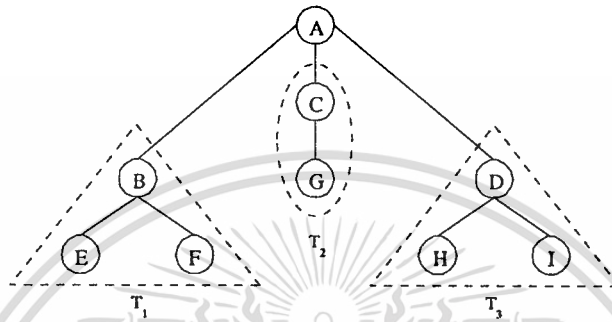
### 1.3.3 ต้นไม้ (Tree)

ต้นไม้คือ Connected Undirected Graph ที่ไม่มี Simple Circuit ดังแสดงในรูปที่ 1.4 ต้นไม้ประกอบด้วยสมาชิกที่เรียกว่าโหนด (Node) ที่

#### 1. ว่าง (ไม่มีโหนดในต้นไม้) หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. มีโหนดหนึ่งถือเป็นรากโหนด ส่วนโหนดที่เหลือแบ่งเป็นต้นไม้ย่อย  $T_1, T_2, T_3, \dots, T_k$  ( $k \geq 0$ ) โดยที่ต้นไม้ย่อยมีคุณสมบัติเป็นต้นไม้เช่นกัน ความสัมพันธ์ระหว่างรากโหนดและรากโหนดของต้นไม้ย่อยเป็นไปในลักษณะพ่อกับลูก คือรากโหนดเป็นพ่อและรากโหนดของต้นไม้ย่อยเป็นโหนดลูก



รูปที่ 1.4 แสดงต้นไม้หรือ Tree

#### 1.3.4 ทฤษฎีความซับซ้อน (Complexity Theory)

ปัญหาใดก็ตามที่สามารถแก้ได้ภายในเวลาพหุนามเราเรียกปัญหาประเภทนี้ว่าปัญหาพี (P-Problem : Polynomial Problem) ส่วนปัญหาใดก็ตามที่ไม่สามารถแก้ได้ภายในเวลาพหุนาม เราเรียกว่าปัญหาเอ็นพี (NP-Problem : Nondeterministic Polynomial Problem) ส่วนปัญหาเอ็นพี (NP) ใด ๆ ก็ตามที่สามารถแปลงจากปัญหาหนึ่งไปเป็นอีกปัญหาหนึ่งได้ภายในเวลาพหุนาม เราจะเรียกปัญหาประเภทนี้ว่าปัญหาเอ็นพีที่ค้อมพลีท (NP-Complete Problem) [11] [12]

#### 1.4 ขอบเขตของการวิจัย

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญของปัญหา วัตถุประสงค์ ทฤษฎีที่เกี่ยวข้อง ขอบเขตของการวิจัย วิธีที่ใช้ในการดำเนินการวิจัยและประโยชน์ที่จะได้รับ

บทที่ 2 การออกแบบวงจรรวมในระดับกายภาพของวงจรรวมแสดนคาร์ดเซล กล่าวถึง คุณลักษณะเฉพาะการออกแบบระดับกายภาพของวงจรรวมแสดนคาร์ดเซล ประเภทของเซลที่ใช้ในวงจรรวมแสดนคาร์ดเซล กฎการออกแบบ ลักษณะทางกายภาพวงจรรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงคาร์คเซล ขั้นตอนการออกแบบระดับกายภาพ และปัญหาการออกแบบทางกายภาพซึ่งประกอบด้วย ปัญหาการจัดวางและปัญหาการเชื่อมโยง

บทที่ 3 สิมูเลทเตดอัลเนลลิ่งอัลกอริทึม (Simulated Annealing Algorithm) กล่าวถึงปัญหาทางเลือก การทำงานของ สิมูเลทเตดอัลเนลลิ่งอัลกอริทึม การกำหนดสถานะที่เป็นไปได้ การกำหนดฟังก์ชันเป้าหมาย การเปลี่ยนสถานะ การกำหนดรูปแบบการลดลงของอุณหภูมิ และการประยุกต์นำเอาสิมูเลทเตดอัลเนลลิ่งมาใช้กับปัญหาการเดินทางของพนักงานขาย

บทที่ 4 การจัดวางแบบลำดับชั้น (Hierarchical Placement Design) กล่าวถึงหลักการพื้นฐานการออกแบบวงจรรวมแบบลำดับชั้น การประยุกต์การออกแบบลำดับชั้นมาใช้ในการจัดวาง โดยเริ่มจากการคำนวณโดยประมาณของวงจรรวม ตามด้วยการจัดกลุ่ม การวางผัง และการวางเซลล์โดย สิมูเลทเตดอัลเนลลิ่งอัลกอริทึม

บทที่ 5 การเชื่อมโยง (Routing) กล่าวถึง การกำหนดทางเดินของสายสัญญาณ การทำ Segment Decomposition การทำ Permutation Decomposition การกำหนด Switchable Segment และการกำหนดทางเดินภายในช่องทางโดยใช้ สิมูเลทเตดอัลเนลลิ่งอัลกอริทึม

บทที่ 6 ทดสอบกับ MCNC Benchmark พัฒนาโปรแกรมที่ใช้ในการจัดวางและการเชื่อมโยง (Automatic Place and Route) บน Microsoft Windows โดยใช้ Visual C++ เพื่อทดสอบกับ MCNC Benchmark ที่ใช้ใน Workshop และ Conference สำหรับการจัดวางและการเชื่อมโยง สำหรับแสดงคาร์คเซล

บทที่ 7 บทสรุป กล่าวถึง สรุปผลการวิจัยและข้อเสนอแนะ

และในส่วนสุดท้ายซึ่งเป็นภาคผนวกกล่าวถึงรายละเอียดของโปรแกรมที่ใช้ในงานวิจัยเพื่อความสะดวกของผู้ที่จะค้นคว้าต่อไป

## 1.5 วิธีที่ใช้ในการดำเนินการวิจัย

การดำเนินการวิจัยแบ่งออกเป็น 3 ขั้นตอน

ขั้นตอนที่ 1 เริ่มต้นด้วยการศึกษาวิธีการออกแบบในระดับกายภาพของเลเอาต์สำหรับวงจรรวมแอสแตนด์คาร์ดเซล ซึ่งประกอบด้วยการจัดวางและการเชื่อมโยง

ขั้นตอนที่ 2 นำหลักการออกแบบผังวงจรรวมแบบลำดับขั้นมาประยุกต์ใช้กับปัญหาการจัดวางและการเชื่อมโยง ภายใต้สมมติฐานเทคโนโลยีอัลติมาติค

ขั้นตอนที่ 3 นำอัลกอริทึมที่ได้จากขั้นตอนที่ 1 และ ขั้นตอนที่ 2 เขียนโปรแกรม “Automatic Place and Route” เพื่อทดสอบกับ MCNC Benchmark

## 1.6 ประโยชน์ที่จะได้รับ

1. เป็นการเพิ่มประสิทธิภาพการแก้ปัญหาของการจัดวางและการเชื่อมโยงสำหรับวงจรรวมแอสแตนด์คาร์ดเซลให้มีประสิทธิภาพสูงขึ้นและเป็นไปตามวัตถุประสงค์ทั้ง 5 ประการ คือ Aspect Ratio, การใช้เนื้อที่ของเลเอาต์, ประสิทธิภาพทางไฟฟ้า, การเชื่อมโยงที่สมบูรณ์, เวลาที่ใช้ในการแก้ปัญหาและจำนวนเซลล์

2. การออกแบบวงจรรวมเป็นงานที่ซับซ้อนมีหลายขั้นตอน ต้องใช้บุคลากรที่มีความรู้ความสามารถโดยเฉพาะ ในประเทศไทยยังขาดผู้มีความรู้ความสามารถทางด้านนี้อยู่เป็นจำนวนมาก เนื่องจากอุปกรณ์ที่ใช้ในการเรียนการสอนมีราคาสูง การแพร่กระจายการศึกษาอยู่ในวงจำกัด ประกอบกับมีผู้ศึกษาทางด้านนี้ไม่มากนัก เนื้อหาที่อยู่ในงานวิจัยนี้จะเป็นประโยชน์อย่างยิ่งสำหรับนักศึกษาและนักวิจัยที่สนใจการออกแบบวงจรรวมในระดับกายภาพ ซึ่งประกอบด้วยวิธีการแก้ปัญหาการจัดวางและการเชื่อมโยงของวงจรรวมแอสแตนด์คาร์ดเซล

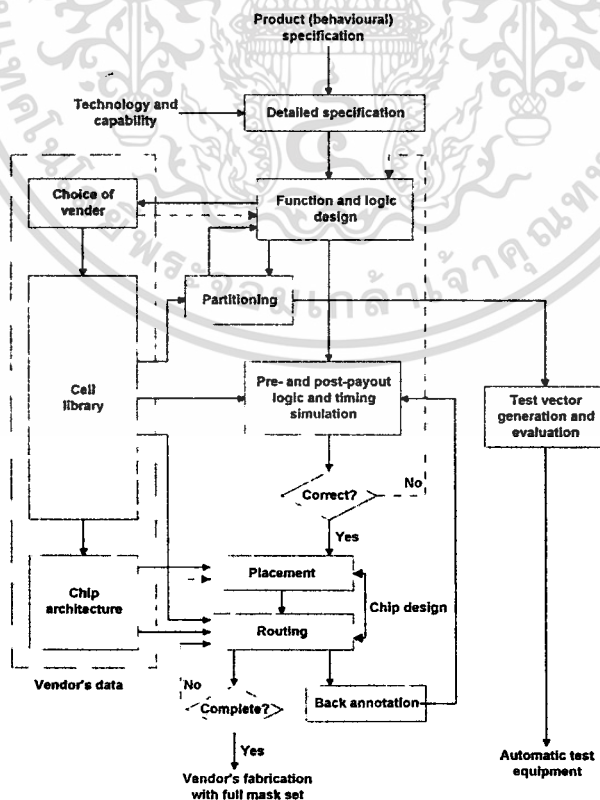
3. การออกแบบในระดับกายภาพของวงจรรวมต้องอาศัยคอมพิวเตอร์เข้ามาช่วยในการออกแบบ ซึ่งซอฟต์แวร์ส่วนใหญ่ที่ใช้ในปัจจุบันเป็นซอฟต์แวร์ที่ซื้อจากต่างประเทศทั้งสิ้น ถ้าเราสามารถพัฒนาซอฟต์แวร์ช่วยออกแบบในระดับกายภาพขึ้นมาใช้งานได้เอง จะเป็นการลดการขาดดุลกับต่างประเทศได้เป็นอย่างดี

## บทที่ 2

### การออกแบบวงจรรวมในระดับกายภาพของวงจรรวมแอสตนคาร์ดเซล

#### 2.1 คุณสมบัติเฉพาะการออกแบบระดับกายภาพของวงจรรวมแอสตนคาร์ดเซล

การออกแบบในระดับกายภาพของวงจรรวมแอสตนคาร์ดเซลใช้เทคนิคการออกแบบ เซมิคัสตอม (Semi-Custom Design) หรือเซลเบส (Cell-Based Design) ผู้ออกแบบไม่จำเป็นต้องออกแบบทางกายภาพในระดับเซลเอง อาศัยพรีดีไซน์ (Pre-Designed Cells) ที่ถูกออกแบบมาอย่างดีโดยผู้เชี่ยวชาญ ซึ่งต้องใช้ความรู้ความสามารถ ทักษะ และประสบการณ์อย่างสูง เพราะเป็นการออกแบบโครงสร้างและมิติในชั้นงานซิลิคอนโดยตรง ผู้ออกแบบเพียงแต่นำข้อมูลที่เป็นคุณลักษณะเฉพาะของเซลซึ่งจัดเก็บอยู่ในไลบรารี เช่น ขนาดความกว้างและความยาวของเซล ตำแหน่งอินพุทและเอาต์พุทหรือข้อมูลอื่น ๆ ที่จำเป็นมาใช้ในการจัดวางและเชื่อมโยงเท่านั้น ดังแสดงในรูปที่ 2.1



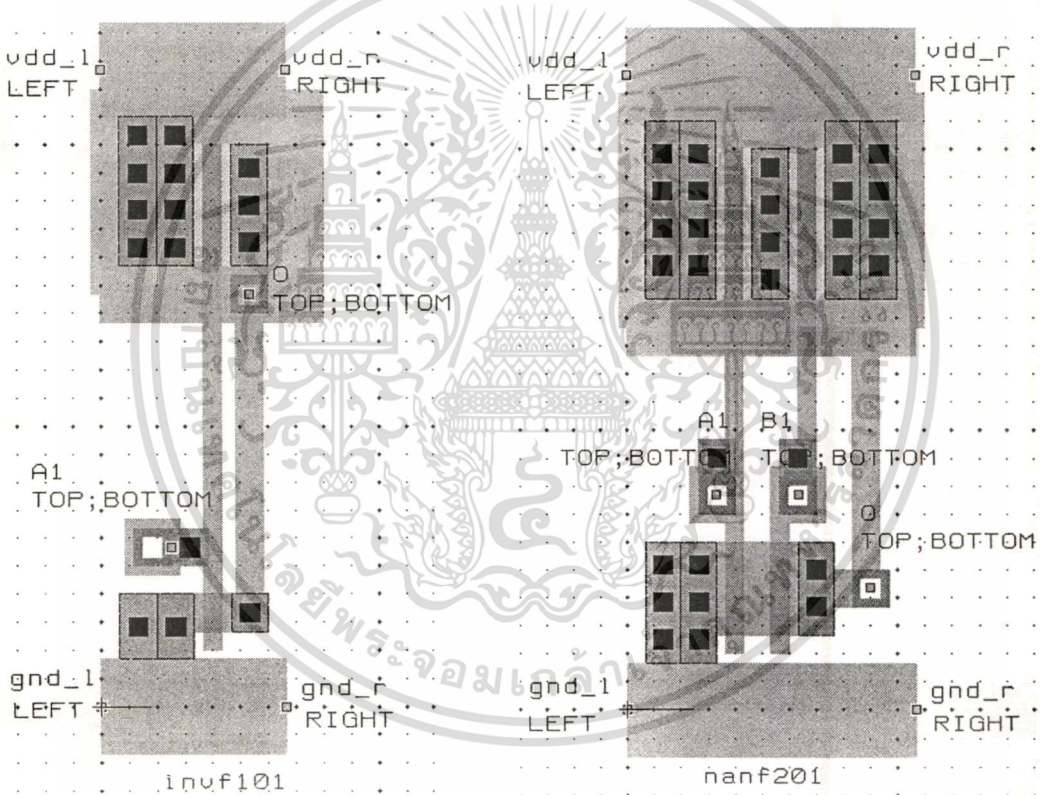
รูปที่ 2.1 แสดงขั้นตอนการออกแบบระดับกายภาพของวงจรรวมแอสตนคาร์ดเซล

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้หรือการสงวนลิขสิทธิ์ที่อื่นใดของผู้ที่พัฒนาโดยบริษัทเอกชนด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 ประเภทของเซลล์ที่ใช้ในวงจรรวมแสดนคาร์ดเซลล์

### 2.2.1 ฟิกเซลล์ (Fixed-Cell)

เซลล์ประเภทนี้มีคุณลักษณะที่สำคัญคือ มีขนาดความสูงคงที่แต่ความยาวเปลี่ยนแปลงไปตามชนิดของเซลล์ มีความเหมาะสมต่อการจัดวางในลักษณะแถวเนื่องจากมีจุดเชื่อมต่อของพาวเวอร์ (Power) และกราวด์ (Ground) อยู่ภายในเซลล์ทั้งด้านซ้ายและด้านขวามีอินพุตและเอาต์พุตอยู่ที่ด้านบนและด้านล่างของเซลล์ ทำให้เราสามารถเลือกจุดเชื่อมต่อที่ด้านบนหรือด้านล่างของเซลล์ก็ได้ ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 แสดงลักษณะทางกายภาพของฟิกเซลล์

คุณลักษณะที่สำคัญอีกประการหนึ่งของเซลล์ประเภทนี้ก็คือไม่สามารถแก้ไขเปลี่ยนแปลงทางกายภาพของเซลล์ได้ จึงเรียกเซลล์ประเภทนี้ว่า Hard Coded Cell ยกตัวอย่างเช่น Gates, Latches, Adders เป็นต้น ส่วนใหญ่เป็นกลุ่มวงจรในระดับ SSI และ MSI ซึ่งถือว่าเป็นเซลล์พื้นฐาน (Basic Cell) หรือแสดนคาร์ดเซลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.2 แมคโครเซลล์ (Macro-Cells)

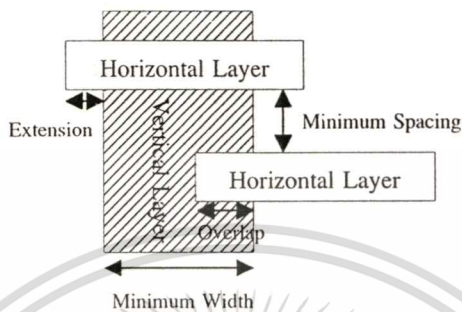
สำหรับการออกแบบที่ต้องอาศัยความซับซ้อนของวงจรแล้วการใช้ฟิสิกเซลล์เพียงอย่างเดียวซึ่งส่วนใหญ่เป็นเซลล์พื้นฐาน ไม่สามารถตอบสนองความต้องการบางประการของผู้ออกแบบได้ ดังนั้นจึงต้องอาศัยเซลล์ขนาดใหญ่ซึ่งเราเรียกว่า แมคโครเซลล์ ขนาดของเซลล์ไม่คงที่ทั้งความสูงและความยาวเปลี่ยนแปลงไป ยกตัวอย่างเช่น Microprocessor, RAM, ROM

### 2.3 กฎการออกแบบ (Design Rules)

ในการออกแบบในระดับกายภาพของวงจรรวม ต้องอาศัยกฎเกณฑ์ระหว่างนักออกแบบและโรงงานที่เจือสาร เพื่อให้ได้วงจรรวมที่ผลิตออกมาแล้วสามารถทำงานจริงได้ กฎเกณฑ์นี้เรียกว่ากฎการออกแบบ ในขั้นตอนการผลิตวงจรรวมนี้มีขั้นตอนอยู่มากมาย ตั้งแต่การทำหน้ากาก (Mask) การทำกระบวนการโฟโตริซิส (Photo Resist) การเจือสารเข้าไปในแต่ละชั้นสาร เป็นต้น ในแต่ละชั้นตอนมักจะมีผลผลิตเกิดขึ้นเสมอ เช่น หน้ากากแต่ละชุดมีการเสื่อมเกิดขึ้นที่ขอบของภาพ เมื่อนำวางซ้อนกันขอบของภาพที่ได้จากกระบวนการโฟโตริซิสก็เลยออกไป หรือในขั้นตอนการเจือสารที่ใช้ในการเจือได้แพร่เลยออกจากขอบที่กำหนดไว้ ปัญหาเหล่านี้อาจทำให้วงจรรวมเกิดการลัดวงจรหรือเสียหายจนทำงานไม่ได้ ดังนั้น เพื่อให้สามารถรับประกันได้ว่าวงจรรวมที่ผลิตขึ้นจะมีคุณสมบัติทางไฟฟ้า เช่น ค่าความต้านทาน ค่าความจุไฟฟ้า ถูกต้องตามที่ต้องการ จึงต้องมีการกำหนดมาตรฐานของแต่ละส่วนของวงจรรวมเข้าเป็นกฎเกณฑ์การออกแบบ แต่เนื่องจากเทคโนโลยีของขบวนการผลิตวงจรรวมได้พัฒนาไปอย่างรวดเร็วมาก ทำให้สามารถผลิตวงจรรวมที่มีขนาดเล็กลงได้ ดังนั้นขนาดมาตรฐานที่กำหนดไว้ในหน่วยวัดระยะทางจริง จึงต้องเปลี่ยนแปลงตามไปด้วย สร้างความยุ่งยากให้กับนักออกแบบวงจรรวมที่ต้องคอยเปลี่ยนกฎการออกแบบในเครื่องมือช่วยออกแบบให้อยู่ในรูปของหน่วยวัดระยะทางอิสระ ที่สามารถอ้างอิงถึงหน่วยวัดระยะทางจริงได้ และสามารถลดขนาดลงตามเทคโนโลยีที่เปลี่ยนแปลงไปได้ง่าย โดยไม่มีผลกระทบต่อกระบวนการออกแบบเลย เช่น กฎการออกแบบของ Mead-Conway ที่ใช้เทคโนโลยีแบบเอ็นเอ็มอส (NMOS) ซึ่งนิยมใช้กันอย่างแพร่หลาย และในภายหลังได้มีผู้คิดสร้างกฎการออกแบบในตนเองเดียวกันสำหรับเทคโนโลยีอื่น ๆ อีกมากมาย เช่น MOSIS CMOS 2.0 เป็นต้น

ดังนั้น การออกแบบในระดับกายภาพ ข้อมูลส่วนหนึ่งต้องอาศัยกฎการออกแบบ คือ ขนาดความกว้าง (Width) ระยะห่าง (Separation) ความยาวของส่วนเกิน (Extension) และความกว้างของส่วนคาบเกี่ยว (Overlap) ดังแสดงในรูปที่ 2.3 โดยมีหน่วยวัดระยะทางอิสระ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

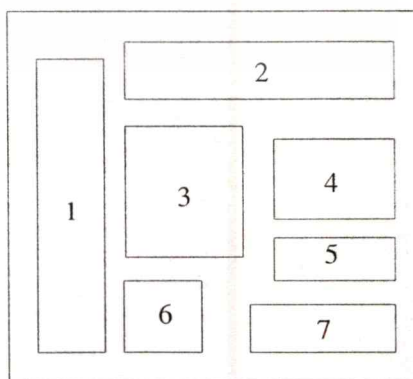
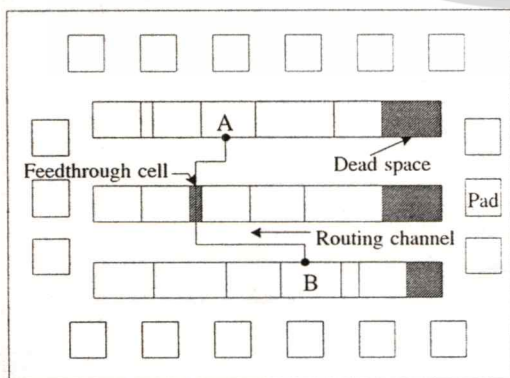
ตามกฎการออกแบบของ Mead-Conway คือค่า  $\lambda$  เป็นค่าที่ประมาณจากค่าผิดพลาดทางตำแหน่งที่มากที่สุดที่อาจเกิดขึ้นได้จากการใช้หน้ากากในขั้นตอนการผลิตวงจรรวม สำหรับค่า  $\lambda$  นั้นจะขึ้นกับความสามารถของเทคโนโลยีการผลิตวงจรรวม



รูปที่ 2.3 แสดงกฎการออกแบบ

### 2.4 ลักษณะทางกายภาพวงจรรวมแสดงตนดาร์เซลล์

ผัง (Floorplan) ทางกายภาพของวงจรรวมสามารถแบ่งออกได้เป็นสองประเภทใหญ่ ๆ คือ แสดงตนดาร์เซลล์เดี่ยว (Standard Cell Layout) และแมคโครเซลล์หรือบิลด์ดิ้งบล็อก (Building Block) ดังแสดงในรูปที่ 2.4 (a) และ (b) ในการออกแบบระดับกายภาพให้ได้ตามผังที่กำหนดทั้งแบบแสดงตนดาร์เซลล์และแมคโครเซลล์ ประกอบด้วยขั้นตอนที่สำคัญสองขั้นตอน คือ การจัดวางและการเชื่อมโยง สำหรับวิธีในการแก้ปัญหาทั้งสองของแต่ละแบบจะแตกต่างกัน เนื่องจากลักษณะของเซลล์ที่ใช้ในแสดงตนดาร์เซลล์เดี่ยวเป็นแบบฟิคเซลล์ ส่วนแมคโครเซลล์เดี่ยวใช้เซลล์ที่มีขนาดไม่คงที่ทั้งความกว้างและความสูง

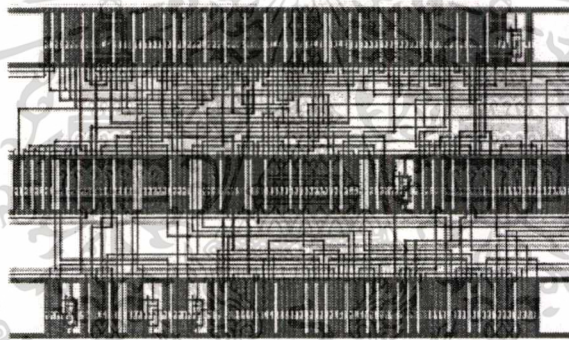


รูปที่ 2.4 (a) แสดงกายภาพของแสดงตนดาร์เซลล์

(b) แสดงกายภาพของแมคโครเซลล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับในการวิจัยนี้เป็นการออกแบบทางกายภาพวงจรรวมแสดนคาร์ดเซล โดยยึดถือแบบการวางผังของแสดนคาร์ดเซลเลเอาต์เท่านั้น เนื่องจากลักษณะทางเรขาคณิตของเซลประเภทนี้ มีขนาดความสูงคงที่ทุกเซล แต่จะมีความกว้างแตกต่างกันขึ้นอยู่กับประเภทของเซลและจุดเชื่อมต่อของอินพุทและเอาต์พุทอยู่ที่ด้านบนและด้านล่างของเซล ดังนั้น โครงสร้างทางกายภาพของวงจรรวมแสดนคาร์ดเซลที่มีความสูงคงที่ จึงใช้วิธีการจัดวางเซลในลักษณะแถวหรือ Row-Base เป็นหลัก ง่ายต่อการออกแบบ ส่วนพื้นที่ระหว่างแถวของเซลถูกใช้เป็นพื้นที่ในการเชื่อมโยงของสายสัญญาณ การเชื่อมโยงของสายสัญญาณเป็นการเชื่อมโยงในลักษณะสองเลขอร์ จุดเชื่อมต่อที่เป็นอินพุทและเอาต์พุทของวงจรจะอยู่ที่ขอบเขตของเลเอาต์ทั้งสองด้านซึ่งเป็น โครงสร้างพื้นฐานทางกายภาพวงจรรวมแสดนคาร์ดเซล ดังแสดงในรูปที่ 2.5

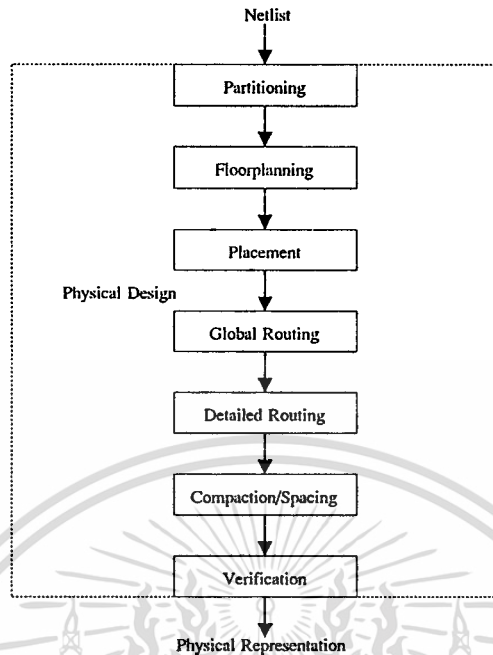


รูปที่ 2.5 แสดงกายภาพของแสดนคาร์ดเซลในลักษณะแถว

## 2.5 ขั้นตอนการออกแบบวงจรรวมในระดับกายภาพ

การออกแบบวงจรรวมในระดับกายภาพสำหรับวงจรรวมขนาดใหญ่จะแบ่งขั้นตอนออกเป็นหลายขั้นตอนดังแสดงในรูปที่ 2.6 โดยขั้นตอนแรกจะมีผลต่อขั้นตอนถัดมาเสมอ แต่สำหรับในการวิจัยนี้เป็นการวิจัยที่เน้นการจัดวางและการเชื่อมโยงสำหรับแสดนคาร์ดเซล ซึ่งการแก้ปัญหาการจัดวางและการเชื่อมโยงเป็นการแก้ปัญหาที่แยกจากกัน แต่มีความสัมพันธ์กัน นั่นหมายถึงเราต้องทำการจัดวางก่อนแล้วจึงทำการเชื่อมโยงทีหลัง ผลที่ได้จากการจัดวางจึงมีอิทธิพลโดยตรงต่อการเชื่อมโยง ถ้าเปรียบเทียบกับ การก่อสร้างอาคาร การจัดวางเปรียบเสมือนการวางฐานรากของอาคารที่ส่งผลต่อการสร้างอาคารชั้นต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงขั้นตอนการออกแบบในระดับกายภาพ

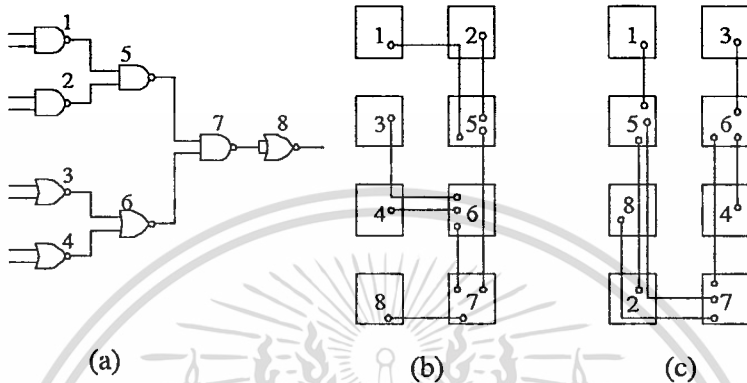
## 2.6 ปัญหาการออกแบบทางกายภาพ

### 2.6.1 ปัญหาการจัดวาง

การจัดวางเป็นการวางเซลล์บนเลเอาท์ในลักษณะ 2 มิติ (2-D Placement) ยกตัวอย่างในรูปที่ 2.7 (a) เป็นวงจรที่ประกอบด้วยเสตนคาร์ดเซลล์จำนวน 8 เซลล์ เพื่อนำมาออกแบบวงจรรวมเสตนคาร์ดเซลล์ในระดับกายภาพ โดยกำหนดให้บล็อกสี่เหลี่ยมหนึ่งบล็อกแทนเซลล์หนึ่งเซลล์ สมมติให้ระยะห่างระหว่างเซลล์ทั้งในแนวนอนและแนวตั้งห่างกันเท่ากับ 1 หน่วย เริ่มต้นจากการวางเซลล์ลงบนเลเอาท์ ซึ่งมีโอกาสเป็นไปได้หลายรูปแบบที่แตกต่างกัน ยกตัวอย่างในรูปที่ 2.7 (b) และ (c) การค้นหารูปแบบการวางเซลล์ที่ดีที่สุดจึงจำเป็นต้องมีเครื่องมือที่แสดงถึงความแตกต่างของการจัดวางเซลล์ในแต่ละแบบ ความยาวรวมของสายสัญญาณที่ใช้ในการเชื่อมโยงระหว่างเซลล์ สามารถนำมาเป็นเงื่อนไขหรือค่าฟังก์ชันเป้าหมายในการเปรียบเทียบการจัดวางในแต่ละแบบได้ ดังแสดงในรูปที่ 2.7 (a) ผลรวมของสายสัญญาณทั้งหมดเท่ากับ 10 หน่วย ส่วนในรูปที่ 2.7 (b) ผลรวมของสายสัญญาณทั้งหมดเท่ากับ 12 หน่วย ดังนั้นถ้ากำหนดให้

$$f(w) = \text{ฟังก์ชันเป้าหมาย}$$

เราสามารถให้ (W) เป็นตัววัดประสิทธิภาพของการจัดวางซึ่งในที่นี้คือระยะทางของสายสัญญาณระหว่างเซลล์ซึ่งคำนวณหาโดยวิธี Manhattan ( $\Delta X + \Delta Y$ ) เมื่อ  $\Delta X = |X_2 - X_1|$  และ  $\Delta Y = |Y_2 - Y_1|$



รูปที่ 2.7 แสดงการจัดวางเซลล์บนเลเอาท์

กำหนดปัญหาการจัดวางให้อยู่ในรูปแบบทางคณิตศาสตร์ ซึ่งประกอบด้วย เซ็ตของแสดนคาร์ดเซลล์  $C = \{c_1, c_2, c_3, \dots, c_n\}$  ที่มีขนาดความสูง  $H$  และความกว้าง  $W$  และเป็นเซ็ตของสายสัญญาณ (Signal Net)  $S = \{s_1, s_2, s_3, \dots, s_m\}$  ให้เซ็ตของตำแหน่ง  $L = \{l_1, l_2, l_3, \dots, l_n\}$  ที่เซลล์วางลงบนเลเอาท์ เมื่อ  $l_i = (x_i, y_i)$  เป็นคู่ลำดับของตำแหน่งบนแกน  $x$  และแกน  $y$  โดยที่  $0 \leq x \leq W$ ,  $0 \leq y \leq H$  เมื่อ  $W$  และ  $H$  เป็นขนาดความกว้างและความยาวของเลเอาท์ตามลำดับ

ถ้าสมมุติให้การแก้ปัญหาการจัดวางแบ่งขั้นตอนออกเป็น 2 ขั้นตอนคือ ขั้นตอนการแบ่งกลุ่มของเซลล์ออกเป็น  $r$  กลุ่ม ๆ ละ  $g_i$  เมื่อ  $1 \leq i < r$  พร้อมเลือกวางลงในแถวจำนวน  $r$  แถว และขั้นตอนการสลับตำแหน่งเซลล์ภายในแถวใด ๆ จะพบว่าขั้นตอนแรกสามารถทำได้ทั้งหมด

$$\binom{n}{g_1} \binom{n-g_1}{g_2} \dots \binom{n-g_{r-1}}{g_r} \quad r! \text{ วิธี}$$

ขั้นตอนที่สองทำได้  $g_i!$  วิธี ดังนั้นจำนวนทางเลือกทั้งหมดที่เป็นไปได้

$$\binom{n}{g_1} \binom{n}{g_2} \dots \binom{n-g_{r-1}}{g_r} \quad r! (g_1! g_2! g_3! \dots g_r!) \text{ วิธี}$$

ยกตัวอย่างการจัดวางแสดงตนคาร์ดเซลจำนวน 100 เซลออกเป็น 4 แถว แต่ละแถวประกอบด้วยเซล 25 เซล จำนวนทางเลือกที่เป็นไปได้ทั้งหมดเท่ากับ

$$\binom{100}{c_{25}} \binom{75}{c_{25}} \binom{50}{c_{25}} \binom{25}{c_{25}} \cdot 4! \cdot (25!) \text{ วิธี} > 10^{100}$$

ถ้าเราใช้เครื่องคอมพิวเตอร์ที่มีอยู่ในปัจจุบันจัดวางเซล 100 เซลให้ครบทั้ง  $10^{100}$  แบบ เราต้องใช้เวลาเป็นปี ๆ นั่นแสดงให้เห็นว่าปัญหาการจัดวางเป็นปัญหาที่ติดตามหาคำตอบที่ดีที่สุดจากทุกคำตอบซึ่งเป็นไปไม่ได้ ในปัจจุบันการแก้ปัญหาประเภทนี้ยังไม่มีอัลกอริทึมที่สามารถค้นหาคำตอบได้โดยตรง ต้องอาศัยอัลกอริทึมอื่นแทน ซึ่งเป็นการพิสูจน์ให้เห็นว่าปัญหาการจัดวางเป็นปัญหาประเภทเอ็นพีคัมพลีท (NP-Complete Problem)

### 2.6.2 ปัญหาการเชื่อมโยง

การเชื่อมโยงจะเกิดขึ้นได้ก็ต่อเมื่อเราทราบตำแหน่งที่แน่นอนของจุดเชื่อมต่อ นั่นหมายถึงจะต้องทราบตำแหน่งที่แน่นอนของเซลที่วางอยู่ภายในแถว การเชื่อมโยงเป็นปัญหาประเภทเอ็นพีคัมพลีท (NP-Complete) เช่นเดียวกับการจัดวาง โดยมีข้อกำหนดพื้นฐานที่สำคัญคือ จะต้องไม่เกิดการลัดวงจรขึ้นหลังจากที่มีการเชื่อมโยงและจะต้องเชื่อมโยงสายสัญญาณได้ครบถ้วน

การเชื่อมโยงจะมีประสิทธิภาพเพียงใด ส่วนหนึ่งขึ้นอยู่กับประสิทธิภาพของการจัดวาง อีกส่วนหนึ่งขึ้นอยู่กับประสิทธิภาพโดยตรงของการเชื่อมโยง ปัญหาการเชื่อมโยงมีลักษณะเหมือนปัญหาการเดินทางของพนักงานขาย (TSP) ที่ต้องเดินทางนำสินค้าไปขายยังเมืองต่าง ๆ โดยใช้ระยะทางในการเดินทางทั้งหมดสั้นที่สุด เป็นที่ทราบกันดีอยู่แล้วว่าปัญหาการเดินทางของพนักงานขายเป็นปัญหาประเภทเอ็นพีคัมพลีท ถ้าเราแทนเมืองต่าง ๆ ที่พนักงานต้องไปขายสินค้าด้วยเซลที่จะเชื่อมโยง และแทนระยะทางระหว่างเมืองต่าง ๆ ด้วยระยะทางระหว่างเซลที่วัดในลักษณะของ Rectilinear Distance ( $\Delta X + \Delta Y$ ) แล้วจะพบว่าปัญหาการเชื่อมโยงเปรียบเสมือนปัญหาการเดินทางของพนักงานขายเช่นกัน เป็นการแสดงให้เห็นว่าปัญหาการเชื่อมโยงเป็นปัญหาประเภทเอ็นพีคัมพลีทเช่นเดียวกัน แต่ความซับซ้อนของปัญหาการเชื่อมโยงมีมากกว่า เนื่องจากสามารถเลือกรูปแบบทางเดินการเชื่อมต่อของสายสัญญาณระหว่างเซลได้อีกด้วย

### บทที่ 3

#### สิมูเลทเตดอัลเนลลิ่งอัลกอริทึม

##### 3.1 ปัญหาทางเลือก (Combinatorial Optimization Problem)

ปัญหาทางเลือกหมายถึงปัญหาที่มีผลลัพธ์ที่เป็นไปได้อยู่เป็นจำนวนมาก บางปัญหาจำนวนผลลัพธ์ที่เป็นไปได้มีอัตราการเพิ่มในรูปของฟังก์ชันเอ็กโปเนนเชียล (Exponential) เมื่อจำนวนอินพุตของปัญหาเพิ่มขึ้น การเลือกหาผลลัพธ์ที่ดีที่สุดจากการเปรียบเทียบทุกสถานะไม่สามารถกระทำได้ ดังนั้น ต้องใช้วิธีค้นหาผลลัพธ์ที่ดีที่สุดจากบางส่วนของผลลัพธ์ที่เป็นไปได้ ซึ่งเราเรียกว่า Local Optimal Solution ผลลัพธ์ที่ได้จะมีคุณภาพเพียงใดขึ้นอยู่กับวิธีการว่าสามารถให้ผลลัพธ์ใกล้เคียงกับ Global Optimal Solution เพียงใด

วิธีการค้นหาผลลัพธ์ตามหลักการ Optimization Problem [13] [14] สามารถทำได้ โดยการกำหนดสถานะที่เป็นไปได้ (Solution Space) และฟังก์ชันเป้าหมาย (Cost Function) เพื่อเป็นค่าวัดเชิงปริมาณในการค้นหาผลลัพธ์ที่ให้ค่าฟังก์ชันเป้าหมายออกมาต่ำที่สุดภายในเงื่อนไขที่กำหนด เริ่มต้นด้วยการกำหนดสถานะหลังจากนั้นเป็นการทำซ้ำโดยเลือกสถานะใหม่จากสถานะรอบข้างของสถานะเดิม แล้วเปรียบเทียบค่าฟังก์ชันเป้าหมายของสถานะใหม่กับสถานะเดิม ถ้าค่าฟังก์ชันเป้าหมายใหม่น้อยกว่า (หรือมากกว่า) ค่าฟังก์ชันเป้าหมายเดิม เราจะยอมรับสถานะใหม่แล้วกลับไปทำซ้ำเช่นนี้จนครบทุกสถานะรอบข้าง จากหลักการข้างต้น เราเรียกว่า Local Search หรือ Iterative Improvement Algorithms นั่นเอง

##### 3.2 สิมูเลทเตดอัลเนลลิ่งอัลกอริทึม

เป็นฮิวริสติกอัลกอริทึม (Heuristic Algorithms) ที่เสนอโดย Kirkpatrick, Gelatt และ Vecchi โดยการประยุกต์จาก Statistical Physical Problems [15] มาเป็น Optimization Problems โดยการนำ Local Search Algorithms มาประยุกต์กับ Metropolis Algorithm ดังแสดงในรูปที่ 3.1 สามารถแก้ปัญหาลักษณะการจัดกลุ่มได้เป็นอย่างดี ง่ายต่อการนำไปใช้งาน และเหมาะสมต่อปัญหาที่มีขนาดใหญ่ ๆ แต่ก็มีข้อเสียในเรื่องระยะเวลาที่ใช้ในการแก้ปัญหา ปัจจุบันได้มีการวิจัยและศึกษาอัลกอริทึมนี้กันอย่างกว้างขวาง พยายามแก้ไขข้อเสียในเรื่องของเวลา โดยการเสนอวิธีการควบคุมอุณหภูมิแบบใหม่ ๆ [16] เช่น Quick Annealing,

Fast Simulated Annealing และ Very Fast Simulated Annealing เป็นต้น แต่ไม่ว่าจะใช้วิธีใดก็ตาม ระยะเวลาที่ใช้ในการคำนวณหาค่าฟังก์ชันเป้าหมายก็ยังคงเป็นปัจจัยพื้นฐานต่อระยะเวลาในการแก้ปัญหา ดังนั้นในงานวิจัยนี้ยังคงใช้ Boltzmann Annealing แต่มุ่งเน้นการลดระยะเวลาการคำนวณของฟังก์ชันเป้าหมายลง ก็เป็นอีกทางหนึ่งในการลดระยะเวลาในการแก้ปัญหาของสิมูเลทเตดอัลเนลลิ่งอัลกอริทึม

**Algorithm Simulated\_Annealing()**

initial solution (i)

initial temperature (T)

do

do

$j = \text{generate}(i)$

if  $\text{accept}(\Delta C, T)$  then

$i = j$

until cost is in equilibrium

reduce(T)

until cost cannot be reduced any further

**End**

รูปที่ 3.1 แสดงอัลกอริทึม Simulated\_Annealing()

**Algorithm** Accept( $\Delta C, T$ )

$\Delta C = \text{cost}(j) - \text{cost}(i)$

**if**  $\Delta C \leq 0$  **then** /\* new cost is less than or equal to the old cost \*/

**return**(ACCEPT) /\* accept the new configuration \*/

**else**

    randomly generate a number  $r$  between 0 and 1

**if**  $r < \exp\left(\frac{-\Delta C}{T}\right)$  **then** **return** (ACCEPT)

**else** **return**(REJECT)

**End**

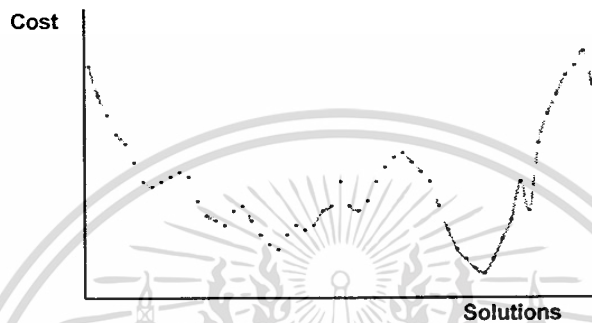
### รูปที่ 3.2 แสดงอัลกอริทึม Accept( $\Delta C, T$ )

การทำงานของอัลกอริทึมเริ่มต้นด้วยการกำหนดสถานะ (Initial Solution) และกำหนดอุณหภูมิ (Temperature) จากนั้นเป็นการทำซ้ำ (Iterative Improvement) ภายในลูป (Loop) ทั้งสองจนกระทั่งถึงจุดที่พอใจ ซึ่งเราจะได้ Local Optimal Solution ที่ใกล้เคียงกับ Global Optimal Solution เมื่อให้เวลาที่เหมาะสม

สำหรับขั้นตอนของการทำซ้ำของลูปในสุดหรือ Metropolis Algorithm เริ่มด้วยการเปลี่ยนสถานะใหม่โดยวิธีการสุ่มแล้วเปรียบเทียบค่าฟังก์ชันเป้าหมายของสถานะใหม่กับสถานะเดิม โดยให้  $\Delta C$  เป็นค่าความแตกต่างระหว่างค่าฟังก์ชันเป้าหมายเดิมกับค่าฟังก์ชันเป้าหมายใหม่ ถ้า  $\Delta C \leq 0$  แสดงว่า สถานะใหม่ดีกว่าสถานะเดิม แต่ถ้า  $\Delta C > 0$  การยอมรับสถานะใหม่หรือไม่ขึ้นอยู่กับค่าความน่าจะเป็นของ Boltzmann Distribution [17] ที่ได้จาก  $e^{-\Delta C/T}$  เปรียบเทียบกับค่าสุ่มที่อยู่ในช่วงศูนย์ถึงหนึ่ง ที่ได้จากการกระจายแบบยูนิฟอร์ม (Uniform) ถ้าค่าที่สุ่มออกมาน้อยกว่า  $e^{-\Delta C/T}$  เราจะยอมรับสถานะใหม่ ข้อพึงสังเกตในขณะค่า  $T$  สูงมาก ๆ การยอมรับการเปลี่ยนสถานะจะเกิดขึ้นก็ต่อเมื่อค่า  $\Delta C$  สูง แสดงว่ามีความแตกต่างกันมากระหว่างค่าฟังก์ชันเป้าหมายเดิมกับฟังก์ชันเป้าหมายใหม่ แต่เมื่อค่า  $T$  ลดลง โอกาสการยอมรับการเปลี่ยนสถานะจะมากขึ้นแม้ค่า  $\Delta C$  จะมีค่าเพียงเล็กน้อย แสดงว่ามีความแตกต่างกันน้อยมากระหว่างค่าฟังก์ชันเป้าหมายเดิมกับฟังก์ชันเป้าหมายใหม่ จากคุณลักษณะข้างต้นเราเรียกว่า Uphill Move ที่ทำให้สิบลูเลทเดออลด์เนลลิ่งอัลกอริทึม ต่างจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interactive Improvement Algorithms ทั่ว ๆ ไป เป็นการหลีกเลี่ยงผลลัพธ์ที่เป็น Local Optimal Solution และเมื่อนำค่าฟังก์ชันเป้าหมายที่ได้จากแต่ละสถานะมาเขียนกราฟ จะได้เส้นกราฟที่เหมือนเทือกเขา ดังแสดงในรูปที่ 3.3 ส่วนลูปนอกสุดเป็นการควบคุมการลดค่าของอุณหภูมิจนกระทั่งถึงจุดสิ้นสุด



รูปที่ 3.3 กราฟแสดงฟังก์ชันเป้าหมายที่ได้จากสิมูเลทเตดอัลเนลลิ่งอัลกอริทึม

การประยุกต์ใช้ สิมูเลทเตดอัลเนลลิ่งอัลกอริทึมเพื่อแก้ปัญหาใด ๆ ต้องพิจารณาถึงองค์ประกอบดังต่อไปนี้

### 1. การกำหนดสถานะที่เป็นไปได้

การกำหนดสถานะที่เป็นไปได้ เป็นการกำหนดเซตของผลลัพธ์ที่จะถูกเลือกด้วยค่าฟังก์ชันเป้าหมาย ดังนั้น ถ้าสามารถกำหนดเซตของสถานะที่เป็นไปได้ทั้งหมดแทบได้เท่าใด จำนวนของสถานะรอบข้างก็จะลดลงตาม ทำให้ระยะเวลาในการค้นหาสั้นลง

### 2. การกำหนดฟังก์ชันเป้าหมาย

ฟังก์ชันเป้าหมายหรือ Objective Function เป็นเครื่องมือวัดคุณภาพของสถานะ ในการเปลี่ยนสถานะแต่ละครั้งจำเป็นต้องคำนวณหาค่าฟังก์ชันเป้าหมายใหม่ เพื่อเปรียบเทียบกับค่าฟังก์ชันเป้าหมายเดิม ดังนั้นฟังก์ชันเป้าหมายที่กำหนดขึ้นควรเป็นฟังก์ชันที่ง่ายต่อการคำนวณและนำไปสู่วัตถุประสงค์ที่ต้องการ

### 3. การเปลี่ยนสถานะ

วิธีการเปลี่ยนสถานะที่นิยมนำมาประยุกต์ใช้มี 2 รูปแบบด้วยกันคือ การย้าย (Move) และการสลับที่ (Exchange) จะใช้รูปแบบใดขึ้นอยู่กับคุณลักษณะของปัญหาเป็นสำคัญ ยกตัวอย่างเช่นในการจัดวาง เราใช้วิธีการสลับที่ระหว่างเซลล์เป็นการเปลี่ยนสถานะ แต่ในการจัดกลุ่มจะใช้ทั้งการย้ายและการสลับที่ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4. การควบคุมอุณหภูมิ (Annealing Schedule)

Annealing Schedule หรือ Cooling Schedule เป็นการควบคุมค่าอุณหภูมิ ซึ่งประกอบด้วย อุณหภูมิเริ่มต้น อุณหภูมิที่ลดลง อุณหภูมิสุดท้าย และจำนวนทรานซิชัน (Transition) ตามทฤษฎีทางคณิตศาสตร์ของ Markov Chain [17] พิสูจน์ได้ว่าสมมุติเหตุอันเล็งอัลกอริทึม สามารถที่จะให้คำตอบออกมาเป็น Global Optimal Solution เมื่อมีการทดลองเปรียบเทียบสถานะใหม่กับสถานะเดิม ด้วยจำนวนครั้งเปรียบเทียบที่มากพอ สำหรับในการวิจัยนี้จะใช้ Annealing Schedule ดังต่อไปนี้

##### 1. อุณหภูมิเริ่มต้น

กำหนดให้ acceptance Ratio( $\chi_0$ ) = 0.8

$$\text{เมื่อ } \chi_0 = \frac{\text{number of accepted transitions}}{\text{number of proposed transitions}}$$

$$T_0 = \frac{(\Delta C^{(+)})}{\ln(\chi_0)}$$

##### 2. อุณหภูมิสิ้นสุด

สิ้นสุดการทำงานเมื่อค่าฟังก์ชันเป้าหมายไม่ลดลงเมื่อจบค่า T ใด ๆ

##### 3. จำนวนทรานซิชัน

ให้จำนวนทรานซิชันมากกว่า 100M เมื่อ M เป็นจำนวนของอินพุต

##### 4. การลดของอุณหภูมิลง

$$T_{k+1} = \alpha T_k \text{ เมื่อ } \alpha = 0.95$$

### 3.3 ปัญหาการเดินทางของพนักงานขาย (TSP)

การเดินทางของพนักงานขายเพื่อจำหน่ายสินค้าในจังหวัดต่าง ๆ นั้นจะเดินทางอย่างไรเพื่อใช้ทางเดินที่สั้นที่สุด อันจะนำไปสู่ผลกำไรของกิจการ การหาทางเดินที่ดีที่สุดนั้นเราสามารถนำหลักการของปัญหาทางเลือกและสมมุติเหตุอันเล็งอัลกอริทึมมาใช้ได้เป็นอย่างดี โดยกำหนดให้  $n$  เป็นจำนวนจังหวัดที่พนักงานขายจะต้องเดินทางผ่าน ลักษณะการเดินทางห้ามเดินทางย้อนกลับมาทางจังหวัดเดิม ยกเว้นจังหวัดที่เป็นจุดเริ่มต้น ซึ่งเป็นจังหวัดแรก ให้  $D = [d_{ij}]$  เป็นระยะทางระหว่างจังหวัด  $i$  และจังหวัด  $j$

จากปัญหาข้างต้นเรานำเอาสัญลักษณ์เตอแอลเนลถึงอัลกอริทึม มาแก้ปัญหาโดยกำหนดเงื่อนไขต่าง ๆ ดังต่อไปนี้

1. กำหนดสถานะที่เป็นไปได้ ( Solution Space (S) )

$$S = \{ \text{ทางเดินที่เป็นไปได้ทั้งหมดของ } n \text{ จังหวัด} \}$$

สมมติให้  $n = 4$  เป็นจำนวนจังหวัด ดังนั้นทางเดินที่เป็นไปได้  $|S| = (n-1)!$  ดังนี้

$$S = \{ \{ก,ข,ค,ง\}, \{ก,ข,ง,ค\}, \{ก,ค,ข,ง\}, \\ \{ก,ค,ง,ข\}, \{ก,ง,ข,ค\}, \{ก,ง,ค,ข\} \}$$

2. กำหนดสถานะรอบข้าง (Neighbourhood Structure)

$S_i = \{ j \neq S \mid \text{ทางเดินใหม่คือ } j \text{ ได้จากการเปลี่ยนเส้นทางระหว่างจังหวัดของทางเดินเดิมคือ } i \}$

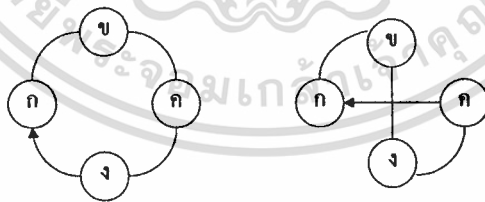
เช่น  $i = \{ก,ข,ง,ค\}$  ดังนั้นสถานะรอบข้างของ  $i$  คือ

$$N_i = \{ \{ก,ข,ค,ง\}, \{ก,ค,ข,ง\}, \{ก,ค,ง,ข\}, \{ก,ง,ข,ค\}, \{ก,ง,ค,ข\} \}$$

3. กำหนดการเปลี่ยนสถานะ (Generation Mechanism)

การเปลี่ยนสถานะใหม่ใช้หลักการของการเปลี่ยนลำดับที่ของทางเดินระหว่างจุด 2 จุด โดยในแต่ละวิธีมีทางเลือกของสถานะใหม่

$$|S_i| = (n-1)(n-2)$$



4. กำหนดค่าฟังก์ชันเป้าหมาย

$$f(\pi) = \sum_{i=1}^n d_i$$

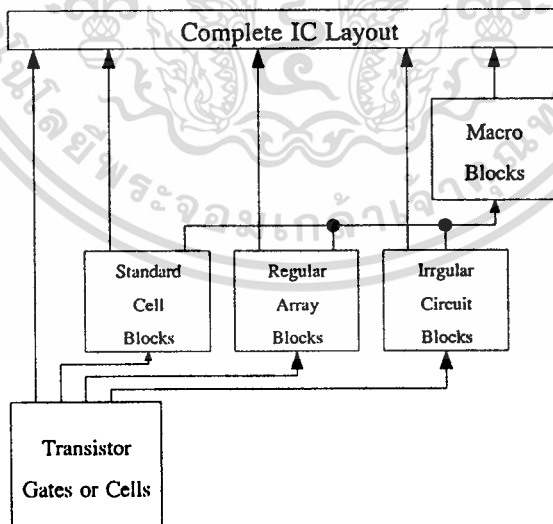
เมื่อ  $\pi$  แทนทางเดินทั้ง  $n$  จังหวัด

## บทที่ 4

### การจัดวาง

#### 4.1 การออกแบบวงจรรวมแบบลำดับชั้น (Hierarchical Design)

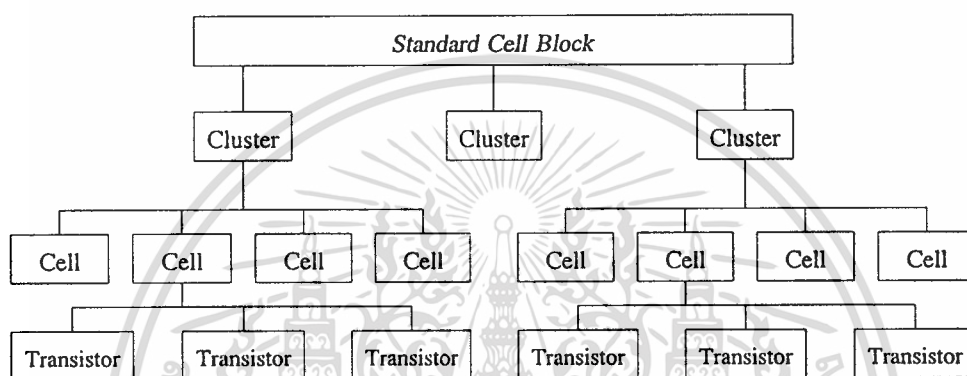
ปัจจุบันการออกแบบวงจรรวม ยึดหลักการออกแบบวงจรรวมแบบลำดับชั้น โดยเฉพาะในการออกแบบวงจรรวมขนาดใหญ่ โครงสร้างภายในประกอบด้วยฟังก์ชันบล็อกชนิดต่าง ๆ ทำงานร่วมกันดังแสดงในรูปที่ 4.1 ระดับล่างสุดเป็นทรานซิสเตอร์ ถัดจากนั้นเป็นฟังก์ชันบล็อกประเภทต่าง ๆ เช่น Standard Cell Blocks, Macro Blocks การออกแบบลักษณะเดิมเป็นการออกแบบชั้นเดียวหรือแบบเรียบ (Flat Design) ไม่สามารถตอบสนองความซับซ้อนของวงจรรวมได้ ซึ่งตรงข้ามกับการออกแบบในลักษณะลำดับชั้น ทั้งนี้เนื่องจากหลักการพื้นฐานที่สำคัญของการออกแบบลำดับชั้นก็คือการแบ่งแยกแล้วเอาชนะ เป็นการลดระยะเวลาในการแก้ปัญหาลง โดยแบ่งงานใหญ่ออกเป็นงานเล็กเพื่อให้สามารถจัดการกับปัญหาได้ง่ายขึ้น



รูปที่ 4.1 การออกแบบวงจรรวมแบบลำดับชั้น

## 4.2 การจัดวางแบบลำดับชั้น (Hierarchical Placement)

การประยุกต์ข้อดีของการออกแบบลำดับชั้นมาใช้ในการจัดวางของวงจรรวม แสดงตนคาร์ดเซล โดยแบ่งออกเป็น 2 ระดับคือ ระดับกลุ่ม (Cluster) และระดับเซล (Cell) ภายในกลุ่มแต่ละกลุ่มประกอบด้วยเซล ภายในแต่ละเซลประกอบด้วยทรานซิสเตอร์ ดังแสดงในรูปที่ 4.2



รูปที่ 4.2 แสดงการจัดวางแบบลำดับชั้น

แนวความคิดในการจัดวางแบบลำดับชั้น [18] [19] ได้แบ่งขั้นตอนการแก้ปัญหาการจัดวางออกเป็น 3 ลำดับคือจัดกลุ่มเซลโดยคำนึงถึงการจับกลุ่มที่เหมาะสมสำหรับการออกแบบเลเอาท์ที่เป็นแถว (Row-Based) เป็นหลัก จัดให้เซลที่มีความสัมพันธ์ในการเชื่อมต่ออยู่ในกลุ่มเดียวกันและระหว่างกลุ่มที่อยู่ติดกันต้องเป็นกลุ่มเซลที่มีความสัมพันธ์กัน โดยลดความสัมพันธ์ที่เกิดจากการข้ามกลุ่มลงให้น้อยที่สุด นำเซลที่จัดกลุ่มแล้วมาวางผังโดยพิจารณาผลกระทบที่มีต่อเนื้อที่ของเลเอาท์ ส่วนในลำดับสุดท้ายเป็นการวางเซลภายในกลุ่มโดยไม่มีการซ้อนทับระหว่างเซลเพื่อกำหนดตำแหน่งของเซลภายในแถวก่อนนำไปเชื่อมโยง

ฟังก์ชันเป้าหมายที่ใช้วัดประสิทธิภาพการจัดวางตั้งแต่ในอดีตที่ผ่านมาคือ ความยาวโดยประมาณของสายสัญญาณที่ใช้ในการเชื่อมโยง (Total Wire Length) ซึ่งฟังก์ชันเป้าหมายที่ใช้ความยาวของสายเพียงอย่างเดียวทำให้เลเอาท์ที่ได้จากการออกแบบมักเกิดปัญหาในเรื่องของประสิทธิภาพทางไฟฟ้า ดังนั้นในการวิจัยนี้ จึงพยายามนำวัตถุประสงค์ทั้ง 5 ประการที่กล่าวมาในบทที่ 1 มาพิจารณาถึงปัจจัยที่มีผลต่อเลเอาท์เพื่อเป็นการกำหนดฟังก์ชันเป้าหมายที่เหมาะสมต่อการออกแบบสำหรับแสดงตนคาร์ดเซลขึ้นมาใช้งานดังต่อไปนี้

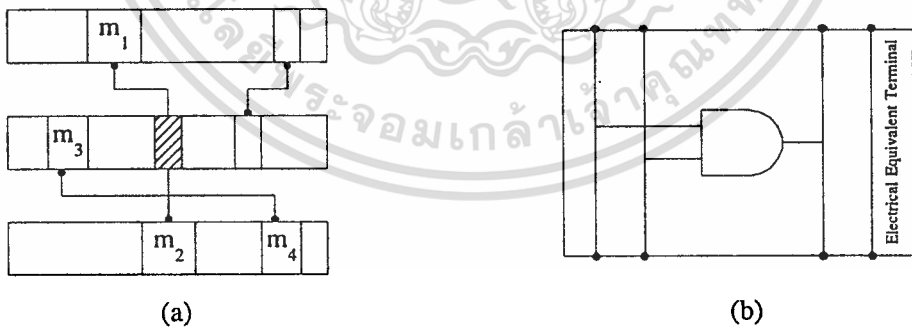
## 1. การใช้เนื้อที่ของเลเอาต์

การใช้เนื้อที่เลเอาต์มีประสิทธิภาพเพียงใด ขึ้นอยู่กับปัจจัยที่สำคัญดังต่อไปนี้ คือ การจัดวางเซลล์, จำนวนฟีดทรูเซลล์ (Feedthrough Cell) และเนื้อที่ว่างเปล่า (Dead Space) ท้ายแถว

ปัจจัยแรกคือการจัดวางเซลล์ เนื่องจากการเชื่อมโยงของแอสแตนคาร์ดเซลล์เป็นแบบ Channel Route เกิดจากการเชื่อมต่ออินพุตและเอาต์พุตของเซลล์อย่างน้อย 2 เซลล์หรือมากกว่า สายสัญญาณที่เชื่อมต่อถูกแบ่งออกเป็นเวกทิคัลเซ็กเมนต์ (Vertical Segment) และฮอริซซอลทัลเซ็กเมนต์ (Horizontal Segment) โดยที่ความยาวของฮอริซซอลทัลเซ็กเมนต์ ขึ้นอยู่กับความยาวของสายสัญญาณโดยตรง ถ้าสามารถลดความยาวของฮอริซซอลทัลเซ็กเมนต์ลงได้มากเท่าไรจำนวนแทรค (Track) ที่จะใช้ในการเชื่อมโยงจะลดลงตามไปด้วย ซึ่งจะส่งผลดีต่อการเชื่อมโยงและการใช้เนื้อที่ของเลเอาต์ในทางบวกมากกว่าทางลบ

ปัจจัยที่สองคือฟีดทรูเซลล์ ในกรณีการเชื่อมโยงระหว่างเซลล์ที่มีแถวของเซลล์วางอยู่ดังแสดงในรูปที่ 4.3 (a) เราไม่สามารถเชื่อมโยงสายสัญญาณทับลงบนเซลล์ ถ้าการเชื่อมโยงเป็นการเชื่อมโยงแบบสองเลเยอร์ จำเป็นต้องใช้ฟีดทรูเซลล์เพื่อเป็นช่องทางผ่านของสายสัญญาณซึ่งสามารถทำได้ 2 วิธีดังนี้

(1) ใช้ฟีดทรูพิน (Feedthrough Pin) ซึ่งเป็น Electrical Equivalent Terminal ที่อยู่ภายในเซลล์ ในกรณีนี้เซลล์ที่ออกแบบมาต้องมีฟีดทรูพินมาด้วยดังแสดงในรูปที่ 4.3 (b)



รูปที่ 4.3 แสดงการใช้ฟีดทรูเซลล์

(2) ใช้ฟีดทรูเซลล์เพิ่มเข้าไปในแถวซึ่งจะมีผลกระทบต่อเนื้อที่และ Aspect Ratio โดยตรง การลดจำนวนของฟีดทรูเซลล์จึงเป็นปัจจัยหนึ่งที่ส่งผลดีต่อเนื้อที่ของเลเอาต์เช่นกัน การกำหนดฟังก์ชันเป้าหมายที่ควบคุมฟีดทรูเซลล์สำหรับการออกแบบแอสแตนคาร์ดเซลล์ที่ไม่มี Electrical Equivalent Terminal จึงจำเป็นอย่างยิ่ง

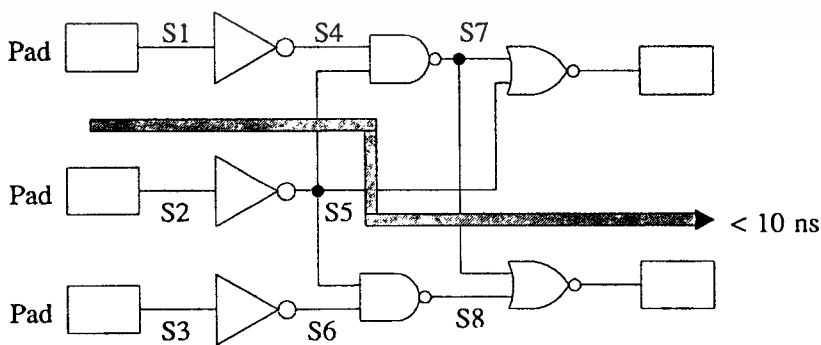
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจัยที่สามคือ พื้นที่ว่างเปล่าที่ขั้วแฉกเกิดขึ้นเนื่องจากแต่ละแฉกมีจำนวนเซลล์ไม่เท่ากัน แต่ละเซลล์มีขนาดความกว้างของเซลล์ไม่เท่ากัน จึงทำให้เนื้อที่ว่างเปล่าที่ขั้วแฉกไม่เท่ากัน การควบคุมพื้นที่ว่างเปล่าที่เหลืออยู่เท่า ๆ กันจะทำให้การใช้เนื้อที่ของเลเอาท์เป็นไปอย่างมีประสิทธิภาพ

## 2. ประสิทธิภาพทางไฟฟ้า

เป็นเรื่องน่าสนใจที่สำคัญอีกประการหนึ่งซึ่งเกี่ยวข้องกับการจัดวางเซลล์ประสิทธิภาพทางไฟฟ้าของวงจรขึ้นอยู่กับระยะทางการเชื่อมโยงของสายสัญญาณ เปรียบเสมือนการเดินทางโดยรถยนต์ที่วิ่งด้วยอัตราความเร็วคงที่ 20 กม./ชม. จากจังหวัด ก. ไปยังจังหวัด ข. ในการเดินทางมีเส้นทางให้เลือกสองเส้นทางด้วยกัน เส้นทางแรกระยะทางทั้งหมด 100 กม. เส้นทางที่สองระยะทาง 80 กม. ระยะเวลาที่ใช้ 50 และ 40 นาทีตามลำดับ แต่ถ้าการเดินทางมีเงื่อนไขเป็นการเดินทางไปสนามบินเพื่อโดยสารเครื่องบินไปต่างประเทศ เหลือเวลาเพียง 45 นาที เครื่องบินจะขึ้นจากสนามบิน ทำให้การเลือกเส้นทางในการเดินทางมีผลทันที

สำหรับในวงจรรวมเมื่อป้อนกระแสไฟเข้าสู่อินพุต กระแสไฟไหลผ่านเซลล์และสายสัญญาณที่ต่อกันเป็นวงจร เพื่อไปออกที่เอาต์พุต ต้องใช้เวลาในช่วงหนึ่ง เรียกว่า Propagation Delay เวลาที่ใช้มากที่สุดจะต้องไม่เกินเวลาที่กำหนด ดังแสดงในรูปที่ 4.4 ในการออกแบบเลเอาท์เพื่อแก้ปัญหาคritical path (Critical Path) ที่เกิดจาก Propagation Delay นั้น เงื่อนไขของความล่าช้าเกิดจาก สาเหตุหลายประการด้วยกันเช่น Intrinsic Delay ของ Device, Drive Factor, Temperature, Voltage และ Track Length ซึ่งมีผลสืบเนื่องมาจากการจัดวางและการเชื่อมโยง จึงจำเป็นที่ต้องนำเอาเงื่อนไขความยาวของสายสัญญาณมาเป็นส่วนหนึ่งของฟังก์ชันเป้าหมาย



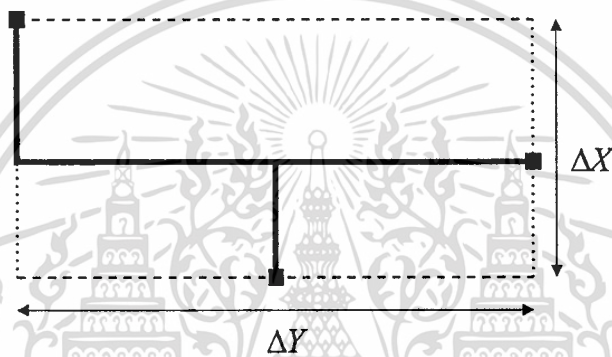
รูปที่ 4.4 แสดงคลิทิคัลพาทภายในวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมปัญหาคลิทธิคัลพารสามารถทำได้หลายวิธี [4] วิธีที่ง่ายและไม่ต้องอาศัยข้อมูลทางเทคนิคมากมายก็คือการใช้ความยาวของสายสัญญาณทุกเส้นที่อยู่ภายใต้คลิทธิคัลพารเป็นตัวควบคุม โดยกำหนดให้  $L(p)$  เป็น Critical Path Length เมื่อ  $p$  เป็นคลิทธิคัลพารใด ๆ

$$\text{เมื่อ } L(p) = \sum_{\forall n \in p} \text{half\_perimeter}_n$$

Half Perimeter ของสายสัญญาณใด ๆ หาได้จาก  $\Delta X + \Delta Y$  ของเส้นรอบเขตทั้งหมดที่เป็นของสายสัญญาณนั้น ดังแสดงในรูปที่ 4.5



รูปที่ 4.5 แสดง Half Perimeter ของสายสัญญาณ

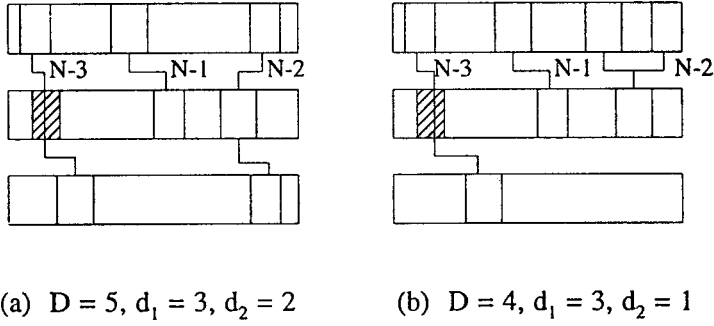
### 3. การเชื่อมโยงที่สมบูรณ์

การเชื่อมโยงของแสดนคาร์ดเซลเป็นการเชื่อมโยงภายในแกนเน็ล พื้นที่ในการเชื่อมโยงอยู่ระหว่างแถวของเซล ขนาดของพื้นที่ระหว่างแถวสามารถปรับเปลี่ยนได้โดยการขยับแถวของเซล ต่างจากการเชื่อมโยงของ Gate Array ที่มีขนาดของพื้นที่คงที่ ทำให้เกิดปัญหา Routing Congestion ทำให้การเชื่อมโยงไม่สมบูรณ์ร้อยละ การเกิด Routing Congestion สาเหตุหนึ่งสืบเนื่องมาจากไม่มีเนื้อที่เพียงพอต่อการเชื่อมโยงสำหรับวงจรรวมแสดนคาร์ดเซล ขึ้นอยู่กับการจัดวางเป็นหลัก ซึ่งมีโอกาสเกิด Routing Congestion ได้เช่นกัน วิธีการแก้ปัญหา Routing Congestion สามารถทำได้โดย

(1) การเชื่อมต่อของสายสัญญาณ ควรใช้จำนวนแกนเน็ลและจำนวนแพริคให้น้อยที่สุด ดังแสดงในรูปที่ 4.6 (a) และ (b) ถ้ากำหนดให้  $D$  เป็นผลรวมของสายสัญญาณทั้งหมด (Channel Density) เมื่อ

$$(D) = \sum_{i=1}^{r-1} d_i$$

$d_i$  = จำนวนของสายสัญญาณที่อยู่ภายในแกนเน็ลที่  $i$



รูปที่ 4.6 แสดงการเชื่อมโยงระหว่างเซลล์

(2) กระจายฮอริซซอลทัลเช็กเมนต์ ไปยังแชนเนลใกล้เคียงที่อยู่ติดกันที่มีจำนวนความหนาแน่นของแทร็ค (Track Density) น้อยกว่าซึ่งจะกล่าวโดยละเอียดในเรื่อง Global Routing

4. Aspect Ratio

ทุกแถวหลังจากจัดวางต้องมีความกว้าง (Width) ไม่เกิน Core Area ที่กำหนด เนื่องจากขณะที่มีการจัดวางเซลล์มีการสลับตำแหน่งของเซลล์ส่งผลกระทบต่อความกว้างของแถว การควบคุมความกว้างของแถวให้อยู่ภายใต้เงื่อนไขทำได้โดยกำหนดให้  $S$  เป็นความแตกต่างความกว้างของแถว ( $L_r$ ) กับความกว้างที่กำหนด ( $L_w$ )

$$\text{เมื่อ } S = |L_r - L_w|$$

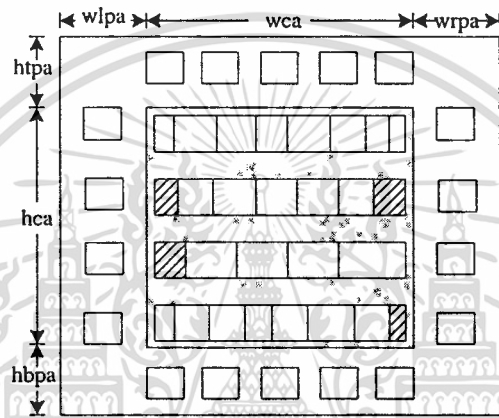
การควบคุมความสูง (Height) ขึ้นอยู่กับจำนวนแถวและพื้นที่ในการเชื่อมโยงก่อนทำการจัดวางและเชื่อมโยงได้มีการคำนวณพื้นที่โดยประมาณ (Area Estimate) มาก่อนซึ่งเป็นการกำหนด Aspect Ratio ในระดับหนึ่งแล้ว

5. เวลาและขนาดของปัญหา

ในขั้นตอนการจัดวางและการเชื่อมโยง (ยกเว้นการทำ Channel Route) ใช้สิมูเลทเตดอัลเนลลิจอัลกอริทึม มีความเหมาะสมต่อการแก้ปัญหาขนาดใหญ่ แต่มีข้อเสียของเวลาซึ่งขึ้นอยู่กับปัจจัยที่สำคัญ 2 ประการคือ การควบคุมอุณหภูมิและฟังก์ชันเป้าหมาย โดยพิจารณาใช้ฟังก์ชันเป้าหมายที่ใช้เวลาในการคำนวณน้อยที่สุดเท่าที่จะทำได้

### 4.3 การคำนวณพื้นที่โดยประมาณ (Area Estimation)

การคำนวณหาพื้นที่โดยประมาณของ Core Area ก่อนทำการจัดวางและการเชื่อมโยงจริง เพื่อตรวจสอบจำนวนแถวที่กำหนด สามารถให้เลเอาท์ใกล้เคียงตาม Aspect Ratio ที่ต้องการหรือไม่ ก่อนดำเนินการจัดวางและเชื่อมโยง โดยกำหนดให้พื้นที่เลเอาท์ถูกแบ่งออกเป็นสองส่วนคือ Core Area กับ Pad Area ดังแสดงในรูปที่ 4.7



รูปที่ 4.7 แสดงพื้นที่ของเลเอาท์

1. แบ่งพื้นที่เลเอาท์ออกเป็น 2 ส่วนคือ Core Area กับ Pad Area เมื่อ Core Area เป็นพื้นที่วางเซลล์ ส่วน Pad Area เป็นพื้นที่วาง Pad Frame ที่อยู่รอบ Core Area ดังนั้น

$$\text{ความกว้างเลเอาท์ (w)} = \text{ความกว้าง Core Area (wca)} + \text{ความกว้าง Pad Area (wpa)}$$

$$\text{ความสูงของเลเอาท์ (h)} = \text{ความสูง Core Area (hca)} + \text{ความสูง Pad Area (hpa)}$$

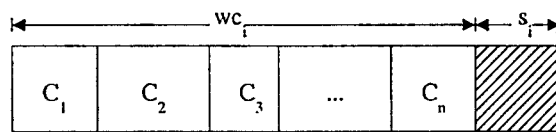
2. ความกว้างของ Pad Area แบ่งออกเป็น 2 ส่วนคือ ด้านซ้ายและด้านขวาของ Core Area ดังนั้น

$$\text{ความกว้าง Pad Area (wpa)} = \text{ความกว้างด้านซ้าย (wtpa)} + \text{ความกว้างด้านขวา (wrpa)}$$

3. ความสูงของ Pad Area แบ่งออกเป็น 2 ส่วนเช่นกันคือ ด้านบนและด้านล่างของ Core Area ดังนั้น

$$\text{ความสูง Pad Area (hpa)} = \text{ความสูงด้านบน (htpa)} + \text{ความสูงด้านล่าง (hbpa)}$$

4. ความกว้างของ Core Area ทุกแถวมีขนาดเท่ากัน แต่ภายในแต่ละแถวมีจำนวนเซลล์แตกต่างกัน ซึ่งทำให้ความกว้างของเซลล์ทั้งหมดในแถวและเนื้อที่ว่างเปล่าแตกต่างกัน



รูปที่ 4.8 แสดงการจัดวางเซลล์ภายในแถว

$$w_{ca} = r \cdot w_{c_i} = w_{c_i} + s_i$$

$$w_{c_i} = \sum_i^{n_i} c_i$$

5. ทุกเซลล์มีขนาดความสูงเท่ากัน กำหนดให้ความสูงของเซลล์ =  $h$
6. ความสูงของ Core Area ขึ้นกับจำนวนแถวคูณด้วยความสูงของเซลล์บวกด้วยความสูงของพื้นที่ในการเชื่อมโยง ดังนั้น

$$h_{ca} = r * h + \sum_{i=1}^{r-1} t d_i$$

เมื่อ  $t d_i$  หมายถึง ความหนาแน่นของแท่งภายในแกนแนล การคำนวณหาจำนวนแท่งในที่นี้จะไม่ใช่หลักการของ Stochastic Model [20] ซึ่งอาศัยหลักการของความน่าจะเป็นแต่จะใช้ อัลกอริทึม Left-Edge [21] ที่ง่ายต่อการเชื่อมโยงมาใช้แทนน่าจะทำให้ได้ค่าใกล้เคียงกับความเป็นจริงได้มากกว่าวิธีเดิมที่ใช้การคำนวณ โดยมีขั้นตอนดังแสดงในรูปที่ 4.9

**Algorithm Estimate\_Area()**

Place not Constrain

Route not Constrain

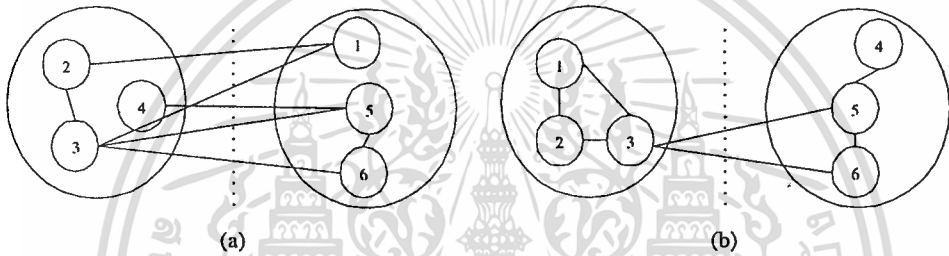
Core Estimate

**End**

รูปที่ 4.9 แสดงอัลกอริทึม Estimate\_Area()

#### 4.4 การจัดกลุ่มเซลล์ (Cell Clustering)

การจัดกลุ่ม เป็นขั้นตอนแรกของการจัดวางแบบลำดับชั้น นับว่ามีความสำคัญอย่างยิ่งต่อการจัดวางและการเชื่อมโยง ในการจัดกลุ่มของเซลล์ออกเป็นกลุ่มอย่างไรดีและอย่างไรเหมาะสมนับว่ามีใช้เรื่องง่าย การจัดกลุ่มแท้ที่จริงเป็นการแบ่งกลุ่ม (Partitioning) เป็นปัญหาประเภทเอ็นพีค้อมพลิต นักวิจัยพยายามศึกษาค้นคว้าวิธีการต่าง ๆ เพื่อนำเอาอัลกอริทึมเหล่านั้นมาประยุกต์ใช้ ส่วนใหญ่พบว่าใช้หลักการพื้นฐานของคัทไลน์ (Cut Line) ดังแสดงในรูปที่ 4.10 เป็นหลักทั้งสิ้น [22] [23] [24] โดยพิจารณาถึง



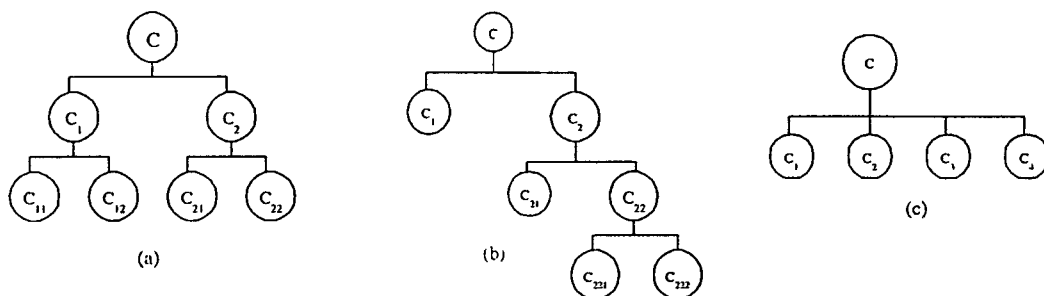
รูปที่ 4.10 แสดงการแบ่งกลุ่ม โดยใช้คัทไลน์

##### 1. ความสัมพันธ์

จัดเซลล์ที่มีความสัมพันธ์กันให้อยู่ภายในกลุ่มเดียวกัน เพื่อเป็นการลดความสัมพันธ์ระหว่างกลุ่มให้น้อยลง

##### 2. การแบ่งกลุ่มในลักษณะ M-Way

วิธีการแบ่งกลุ่มที่คุ้นเคยกันเป็นอย่างดีคือ Two-Way Partition ได้ถูกนำมาประยุกต์ใช้กับการแบ่งกลุ่มในลักษณะ M-Way Partition เพื่อให้จำนวนกลุ่มมากกว่า 2 กลุ่มขึ้นไป มีอยู่ 2 วิธี วิธีแรก ดังแสดงในรูปที่ 4.11 (a) แบ่งกลุ่มของเซลล์ออกเป็น 2 กลุ่ม แต่ละกลุ่มถูกแบ่งออกไปอีกจนได้จำนวนกลุ่มตามต้องการ วิธีที่สองคือ แบ่งกลุ่มของเซลล์ออกเป็น 2 กลุ่มแล้วเลือกแบ่งกลุ่มที่มีขนาดใหญ่กว่าเท่านั้นออกอีกจนได้จำนวนกลุ่มตามที่ต้องการ ดังแสดงในรูปที่ 4.11 (b) ข้อเสียก็คือการแบ่งครั้งต่อไปจะไม่นำเอาส่วนที่แบ่งแล้วมาพิจารณา โดยเฉพาะในวิธีที่สองเห็นได้อย่างชัดเจน ข้อเสียอีกประการหนึ่งก็คือ ถ้าจำนวนกลุ่มมากขึ้น จำเป็นต้องใช้เวลาในการแบ่งมากขึ้น



รูปที่ 4.11 แสดงการแบ่งกลุ่มในลักษณะ M-Way

มีวิธีการหนึ่งที่น่าจะเหมาะสมกว่าสำหรับการจัดกลุ่มเพื่อใช้กับการจัดวางแสดงตนคาร์คเซล ก็คือการแบ่งจำนวนกลุ่มออกเท่าจำนวนที่กำหนดตั้งแต่เริ่มแรกแล้วใช้สิมูเลทเตดอัลเนลลิ่งอัลกอริทึม แก้ปัญหาในการแบ่งกลุ่ม ซึ่งเป็นการมองปัญหาทั้งหมดแทนที่เป็นการมองปัญหาเฉพาะกลุ่มที่แบ่งเท่านั้น ดังแสดงในรูปที่ 4.11 (c)

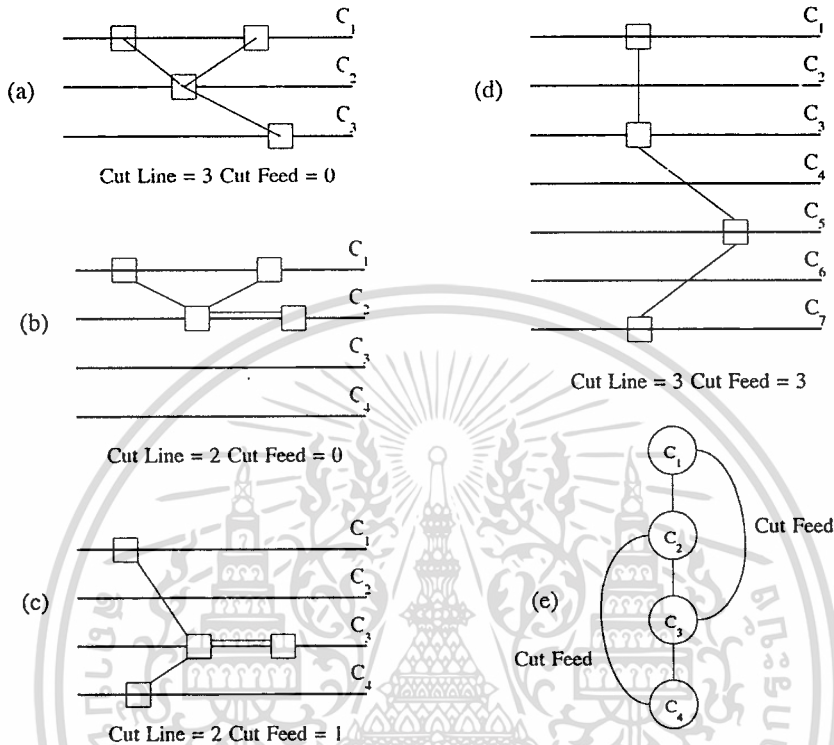
### 3. จำนวนกลุ่มที่ต้องการ

การจัดกลุ่มให้ได้ตามจำนวนกลุ่มที่กำหนดและให้เป็นไปตามธรรมชาติมากที่สุด โดยการกำหนดขนาดของกลุ่มที่ได้ภายหลังจากการแบ่งแล้วในแต่ละกลุ่มต้องมีจำนวนเซลมากกว่าหรือเท่ากับหนึ่งเสมอ

อย่างไรก็ตาม ในการวิจัยนี้เป็นการจัดกลุ่มเพื่อใช้กับการจัดวางแสดงตนคาร์คเซลที่เป็นแถว การนำหลักการของคัทไลน์ ดังแสดงในรูปที่ 4.10 มาประยุกต์ใช้โดยตรงจึงไม่มีความเหมาะสม พิจารณาจากรูปที่ 4.12 เป็นการแบ่งกลุ่มของเซลออกเท่าจำนวนแถวในรูปที่ 4.12 (a) และ (b) มีค่าคัทไลน์เท่ากับ 3 และ 2 ตามลำดับ ตามเกณฑ์การตัดสินใจการแบ่งกลุ่มโดยใช้คัทไลน์ การวางเซลในรูป 4.12 (b) จะดีกว่าในรูป 4.12 (a) เนื่องจากมีค่าคัทไลน์ต่ำกว่า ซึ่งในความเป็นจริงแล้ว ในการจัดวางเซลในลักษณะแถว การวางเซลทั้งสองรูปแบบไม่มีความแตกต่างกัน สามารถยอมรับได้ทั้งสองแบบ ในกรณีรูปที่ 4.12 (a) กลับเป็นผลดีต่อการเชื่อมโยงเสียอีก เนื่องจากการกระจายสายสัญญาณออกไป ลดโอกาสการเกิด Routing Congestion นั้นแสดงว่าการใช้คัทไลน์ในการแบ่งกลุ่มในกรณีนี้กลายเป็นผลเสียต่อการเชื่อมโยง การใช้คัทไลน์ (C) จึงไม่เหมาะสม จึงได้เสนอแนวความคิดใหม่โดยใช้คัทฟีด (CF : Cut Feed) แทน ดังแสดงในรูปที่ 4.12 (e) เมื่อ  $CF = C_{ij \neq i+1}$

การนำคัทฟีดมาเป็นเกณฑ์ในการแบ่งกลุ่มแทนคัทไลน์ก็ยังคงมีคุณสมบัติการจัดความสัมพันธ์ระหว่างกลุ่มและภายในกลุ่มเช่นเดิม แต่มุ่งเน้นการจัดกลุ่มที่มีจำนวนคัทฟีดให้

น้อยที่สุด ซึ่งเหมาะสมต่อการจัดวางในลักษณะแถวมากกว่า ดังแสดงได้จากรูปที่ 4.12 (c) และ (d) ตามลำดับ



รูปที่ 4.12 แสดงการใช้คัทไลน์และคัทฟีด

#### 4.5 การจัดกลุ่มเซลล์โดยใช้สิบลูเทเตอดัลเนตลิ่งอัลกอริทึม

แทน Circuit Network ด้วย Undirected Graph  $G(V,E)$  เมื่อ  $V$  เป็นเซตของเซลล์  $v_i \in V, V = \{v_1, v_2, v_3, \dots, v_n\}$  และ  $E$  เป็นเซตของด้าน  $e_i \in E, E = \{e_1, e_2, e_3, \dots, e_m\}$  ให้  $c_{ij}$  เป็นจำนวนด้านระหว่าง  $v_i$  กับ  $v_j$  และ  $i \neq j$  แบ่งกลุ่มเซลล์ทั้งหมดออกเป็น  $R$  กลุ่ม โดย

##### 1. กำหนดเซตของสถานะ

$$S = \{s \in S \mid \text{เป็นกลุ่มของเซลล์ } C \text{ กลุ่มที่เป็นไปได้ทั้งหมด} \}$$

## 2. การเปลี่ยนสถานะ

ใช้วิธีการสลับที่ระหว่างเซลล์จนกระทั่งค่าฟังก์ชันเป้าหมายคงที่จึงเปลี่ยนเป็นการเคลื่อนย้ายเซลล์ระหว่างกลุ่มแทนการสลับที่ระหว่างเซลล์จนค่าฟังก์ชันเป้าหมายลดลงและกลับมาใช้การสลับที่ระหว่างเซลล์แทน

## 3. ฟังก์ชันเป้าหมาย

$$f(n) = \alpha \sum_{j \neq i+1}^R CF_{ij} + \beta \sum_{i=1}^N CPL_i$$

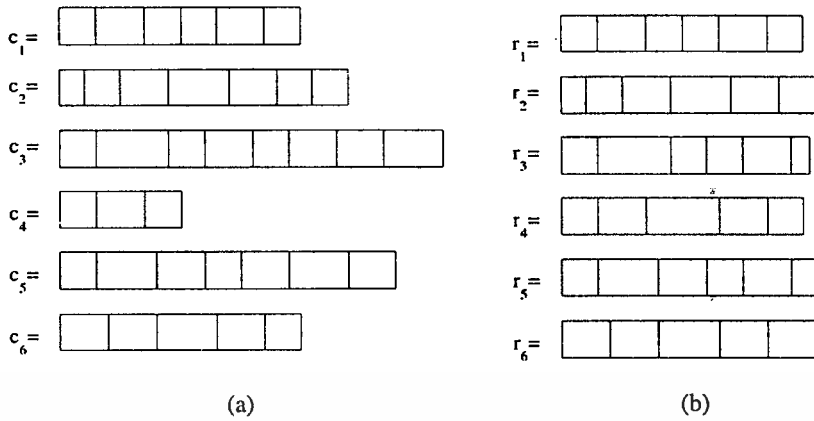
เมื่อ  $\alpha$ ,  $\beta$  เป็นค่าคงที่

CF = Cut Feed

CPL = Critical Path Length

### 4.6 การวางผังเซลล์ (Cell Floorplanning)

การวางผังเซลล์สำหรับแอสแตนคาร์ดเซลล์จะแตกต่างจากการวางผังของ Building Blocks โดยสิ้นเชิง เนื่องจากคุณลักษณะของเซลล์เป็นพิกเซลและจัดเรียงเป็นแถว ความกว้างของเลเอาท์ขึ้นอยู่กับผลรวมของเซลล์ที่อยู่ในแถวที่ยาวที่สุด ส่วนความสูงของเลเอาท์ขึ้นกับความสูงของเซลล์บวกกับความสูงของพื้นที่ในการเชื่อมโยง เราไม่สามารถทราบความสูงที่แท้จริงของพื้นที่ในการเชื่อมโยงได้อาศัยความสูงเฉลี่ยของแขนเนื้อที่ได้จากการคำนวณพื้นที่โดยประมาณ ภายหลังจากการแบ่งกลุ่มภายในแต่ละกลุ่มประกอบด้วยจำนวนเซลล์ไม่เท่ากัน ถ้านำเซลล์มาเรียงต่อกัน จะมีความกว้างไม่เท่ากันดังแสดงในรูปที่ 4.13 (a) พื้นที่ด้านท้ายของแต่ละแถวยกเว้นแถวที่ยาวที่สุดจะกลายเป็นเนื้อที่ว่างเปล่า (Dead Space) เพื่อเป็นการแก้ปัญหาพื้นที่ว่างเปล่าให้น้อยลงและมีความกว้างตามที่กำหนด ต้องมีการวางผังเพื่อปรับให้การวางเซลล์ทุกแถวมีพื้นที่ว่างเปล่าน้อยที่สุด และเฉลี่ยเท่ากัน ดังแสดงในรูปที่ 4.13 (b)



รูปที่ 4.13 แสดงการวางผังเซลล์

สำหรับการออกแบบกับเซลล์ที่ไม่มีพีครูพิน จำเป็นต้องแทรกพีครูเซลล์เข้าไปในแถว โดยไม่สนใจตำแหน่ง แต่จำเป็นต้องทราบทางเดินของสายสัญญาณเสียก่อน โดยใช้มินิมัสมสแพนนิ่งทรี (Minimum Spanning Tree) [25] เป็นตัวกำหนดทางเดินเพื่อให้ทราบจำนวนพีครูเซลล์โดยประมาณ จึงจะสามารถควบคุมพื้นที่ที่ท้ายแถวได้อย่างมีประสิทธิภาพ ถึงแม้ว่าการหาทางเดินที่แท้จริงจะเกิดขึ้นได้ก็ต่อเมื่อตำแหน่งของเซลล์ภายในแถวไม่ถูกเปลี่ยนแปลงแล้ว แต่นั่นไม่ใช่ประเด็นสำคัญ เนื่องจากขั้นตอนที่เหลือเป็นการจัดวางเซลล์ภายในแถวเท่านั้น ไม่มีการย้ายเซลล์ออกจากแถว ดังนั้น การวางผังในที่นี้เปรียบเสมือนเป็นการจัดวางผสมกับการทำ Global Route แบบหยายนั่นเอง เพื่อให้ความกว้างของเลเอาต์มีค่าใกล้เคียงกับ Aspect Ratio

#### 4.7 การวางผังเซลล์ใช้สัมมูลเหตุอัลเนลถึงอัลกอริทึม

กำหนดให้  $R$  เป็นเซตของกลุ่ม  $r_i \in R, R = \{r_1, r_2, r_3, \dots, r_n\}$  และ  $M_{r_i}$  เป็นเซตของเซลล์ในแถวใด ๆ  $m_i \in M_{r_i}, M_{r_i} = \{m_1, m_2, m_3, \dots, m_n\}$  ให้  $W$  เป็นเซตของความกว้างของแถว  $w_i \in W_{r_i}, W_{r_i} = \{w_1, w_2, w_3, \dots, w_n\}$  วางเซลล์ลงบนแถวแล้วให้มีเนื้อที่ว่างเปล่าน้อยที่สุดโดย

##### 1. กำหนดเซตของสถานะ

$$S = \{s \in S \mid \text{เป็นเซตของเซลล์บนแถวที่เป็นไปได้ทั้งหมด} \}$$

## 2. การเปลี่ยนสถานะ

เปลี่ยนสถานะใหม่โดยการเลือกเซลล์แบบสุ่มและสลัที่กัน สำหรับเซลล์ที่มีสายสัญญาณที่เกี่ยวข้องกับคลิทธิคัลพารจะไม่อนุญาตให้มีการสลัที่

## 3. ฟังก์ชันเป้าหมาย

$$f(n) = \sum_{i=1}^r s_i \quad f(n) = \text{เนื้อที่ว่างเปล่า}$$

เมื่อ  $s_i = |w_i - \bar{w}|$

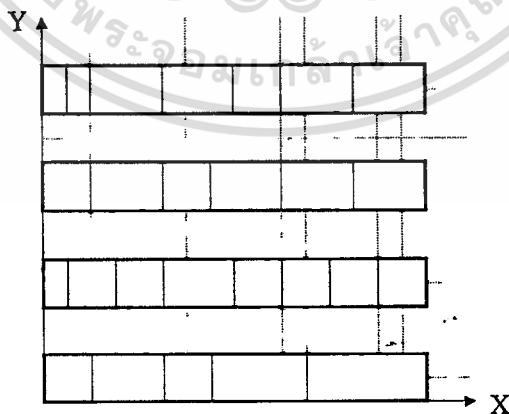
$w_i = \sum_{i=1}^n w_i$  เป็นความยาวรวมของเซลล์ที่อยู่แถวที่  $i$

$\bar{w} = \sum_{i=1}^r w_i$  เป็นความกว้างที่กำหนดที่ได้จากการประมาณ

## 4.8 การวางเซลล์ (Cell Final Placement)

การจัดวางเซลล์เป็นขั้นตอนสุดท้ายของการจัดวางแบบลำดับชั้น ภายหลังจากการจัดกลุ่มและวางผังดั่งที่กล่าวมาแล้ว เป็นการหาตำแหน่งที่ดีที่สุดบนเลเอาท์ที่ให้ความยาวโดยประมาณของสายสัญญาณทั้งหมดด้วยวิธี Half Perimeter ออกมาสั้นที่สุด ดังแสดงในรูปที่

4.14



รูปที่ 4.14 แสดงการจัดวางเซลล์

#### 4.9 การวางเซลโดยใช้สิบลูกเต๋าคอลเลกชันอัลกอริทึม

กำหนดให้กราฟ  $G(M,S)$  เป็น Undirected Graph เมื่อ  $M$  เป็นเซตของเซล  $m_i \in M$  ,  $M = \{m_1, m_2, m_3, \dots, m_n\}$  และ  $S$  เป็นเซตของสายสัญญาณ  $s_i \in S$  ,  $S = \{s_1, s_2, s_3, \dots, s_k\}$  แต่ละเซลประกอบด้วยเซตของสายสัญญาณ  $sm_i \subseteq S$  แต่ละสายสัญญาณประกอบด้วยเซตของเซล  $m_{s_i} \subseteq M$  วางเซลลงบนเลเอาต์บนตำแหน่ง  $(x_i, y_i)$  โดยไม่มีการซ้อนทับระหว่างเซลโดย

1. กำหนดเซตของสถานะที่เป็นไปได้

$$S = \{s \in S \mid \text{การจัดวางเซลที่เป็นไปได้ทั้งหมด}\}$$

2. การเปลี่ยนสถานะ

ใช้วิธีการสลับที่ระหว่างเซลภายในแถวเท่านั้น

3. ฟังก์ชันเป้าหมาย

$$f(n) = \sum_{i=1}^k w_i L_i \quad f(n) = \text{ความยาวของสายสัญญาณ}$$

เมื่อ  $L_i = \text{ความยาวของสายสัญญาณที่ } i$

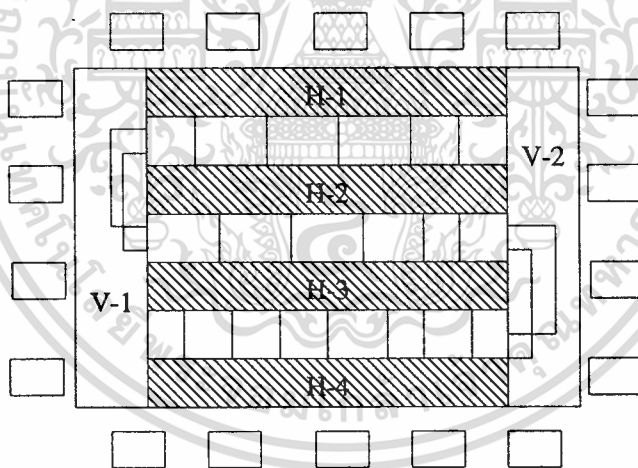
$w_i = \text{Weight ของสายสัญญาณที่ } i$

## บทที่ 5

### การเชื่อมโยง

การเชื่อมโยงเป็นปัญหาประเภทเอ็นพีคัมพลีท เช่นเดียวกับการจัดวาง ดังนั้น จึงได้แบ่งปัญหาการเชื่อมโยงออกเป็น 2 ขั้นตอนคือ Global Routing เป็นการกำหนดช่องทางเดินของสายสัญญาณ ควรใช้เซนเน็ลโคบ้าง และ Detail Routing เป็นการกำหนดทางเดินภายในเซนเน็ล ซึ่งแบ่งออกเป็น เวทิกัลเช็กเมนท์และฮอริซซอนทัลเช็กเมนท์

สำหรับการเชื่อมโยงสายสัญญาณบนเลเอาต์ในที่นี้เป็นการเชื่อมโยงในลักษณะสองมิติ พื้นที่สำหรับเชื่อมโยงแบ่งออกเป็น ฮอริซซอนทัลเซนเน็ล (Horizontal Channel) ซึ่งประกอบด้วย H-1, H-2, H-3, H-4 และ เวทิกัลเซนเน็ล (Vertical Channel) ประกอบด้วย V-1 และ V-2 เพื่อรองรับการเชื่อมโยงสายสัญญาณประเภทต่าง ๆ ดังแสดงในรูปที่ 5.1



รูปที่ 5.1 แสดงเซนเน็ลที่ใช้ในการเชื่อมโยง

#### 5.1 การกำหนดช่องทางเดินของสายสัญญาณ (Global Routing)

โครงสร้างของแอสตนคาร์ดเซลมีจุดเชื่อมต่อของอินพุท/เอาต์พุทที่อยู่ด้านบนและด้านล่างของเซล ถูกออกแบบมาเหมาะสมต่อการจัดวางในลักษณะแถว ผู้ใช้สามารถเลือกจุดเชื่อมต่อได้มากกว่า 1 จุดเสมอ จุดเหล่านี้จะเชื่อมต่อถึงกันโดยมีคุณสมบัติของ Electrical Equivalent การแก้ปัญหาการกำหนดช่องทางเดินของสายสัญญาณเป็นไปตามรูปที่ 5.2

**Algorithm Global\_Routing()**

For Each Net n

Identify the pin of Net n

Segment Decomposition

Identify Switable Segment

Optimize Track Density

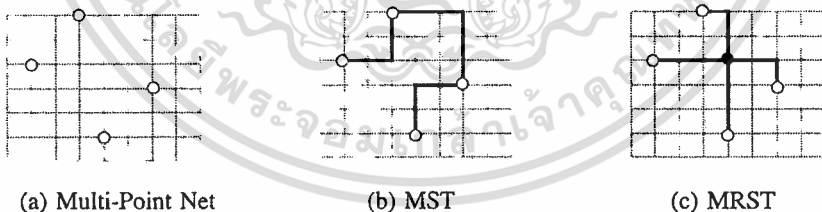
Endfor

**End**

รูปที่ 5.2 แสดงอัลกอริทึม Global\_Routing()

## 5.1.1 Segment Decomposition

สายสัญญาณที่มีจุดเชื่อมต่อระหว่างอินพุตหรือเอาต์พุตมากกว่า 2 จุด (Multi-Point Net) ดังแสดงในรูปที่ 5.3 (a) ต้องแบ่งออกเป็น Two-Point Segment โดยใช้ มินิมีมสแพนนิ่งทรี (MST : Minimum Spanning Tree) หรือมินิมีมเรคติลินเนียสแพนนิ่งทรี (MRST : Minimum Rectilinear Steiner Tree) [26] ดังแสดงในรูปที่ 5.3 (b) และ (c)



รูปที่ 5.3 แสดง Multi-Point Net, MST และ MRST

ความแตกต่างระหว่าง MST และ MRST อยู่ที่จำนวนจุดที่ใช้ในการเชื่อมต่อ และความยาวของสายสัญญาณ ความยาวของ MRST จะสั้นกว่าหรือเท่ากับความยาวของ MST เสมอ [25] ส่วนจำนวนจุดของ MRST จะมากกว่าหรือเท่ากับจำนวนจุดของ MST สาเหตุที่มากกว่า เนื่องจากการเพิ่มสไทเนอร์พอยท์ (Steiner Point) เข้าไป

การใช้ MRST หรือ MST เพื่อกำหนดทางเดินนั้นต้องคำนึงถึงลักษณะของ เอกสาร เลื่อนเข้าและเซลล์เป็นสำคัญ ในอดีตที่ผ่านมา การกำหนดทางเดินของสายสัญญาณจะใช้ MST ในการคำนวณ ไม่ว่าการมีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ MRST เพียงอย่างเดียวหนึ่ง แต่ในการวิจัยนี้นำมาใช้ทั้งคู่ เนื่องจากการออกแบบเลเอาท์ของแอสตาคารด์เซลล์ที่ไม่มีฟิเจอร์พิน การเพิ่มสโตนอร์พอยท์เปรียบเสมือนการเพิ่มฟิเจอร์เซลล์ ทำให้ต้องใช้เนื้อที่ของเลเอาท์มากขึ้น ดังนั้นการใช้ MRST จะเหมาะสมก็ต่อเมื่อสายสัญญาณนั้นเป็นคลิสิกัลพาร์ ซึ่งต้องใช้สายสัญญาณที่สั้นที่สุด ส่วนสายสัญญาณอื่น ๆ ใช้วิธีกำหนดทางเดินโดย MST จะมีความเหมาะสมมากกว่าเนื่องจากอัตราส่วนของ Worst-Case ระหว่าง MST และ MRST เป็นดังนี้ [26]

$$\frac{\text{cost}(MST)}{\text{cost}(MRST)} \leq \frac{3}{2}$$

### 5.1.2 Optimize Track Density

ขั้นตอนถัดมาเป็นการพิจารณา Two-Point Segment โค้งที่เป็นสวิชเช็กเมนต์ (Switchable Segment) คุณสมบัติของสวิชเช็กเมนต์เป็นเซ็กเมนต์ (Segment) ที่มีจุดเชื่อมต่อทั้งสองอยู่ในแนวระนาบเดียวกัน ดังแสดงในรูปที่ 5.4 ทำให้เราสามารถกระจายเซ็กเมนต์จากแขนเน็ลที่มีความหนาแน่นมากกว่าไปยังแขนเน็ลที่มีความหนาแน่นน้อยกว่าโดยใช้สิมูเลทเตดอัลเนลลิ่งอัลกอริทึม เมื่อ

1. กำหนดเซตของสถานะที่เป็นไปได้

$$S = \{ \text{ตำแหน่งของสวิชเช็กเมนต์ที่เป็นไปได้ทั้งหมด} \}$$

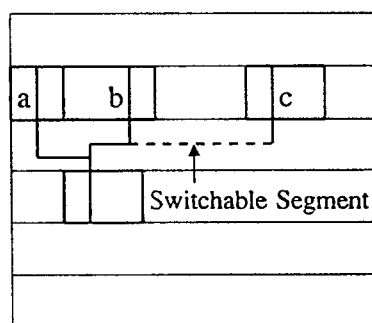
2. การเปลี่ยนสถานะ

สลัปที่สวิชเช็กเมนต์ไปยังแขนเน็ลที่อยู่บนหรือล่าง

3. ฟังก์ชันเป้าหมาย

$$f(n) = \sum_{i=1}^r d_i$$

เมื่อ  $d_i$  = จำนวนเซ็กเมนต์ที่อยู่ในแขนเน็ล  $i$

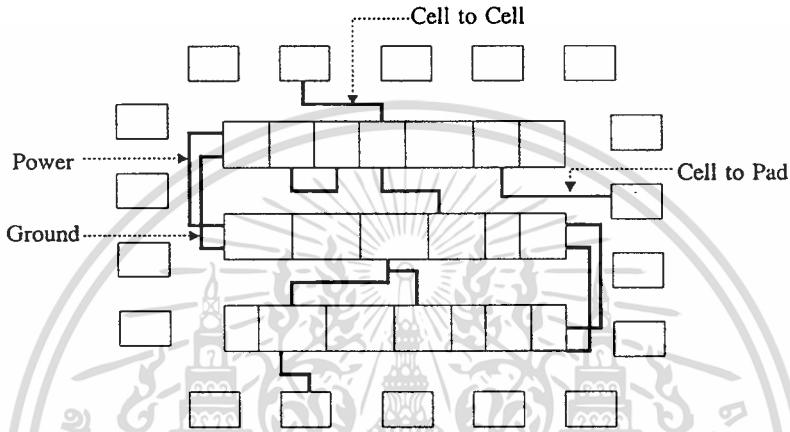


รูปที่ 5.4 แสดง Switchable Segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

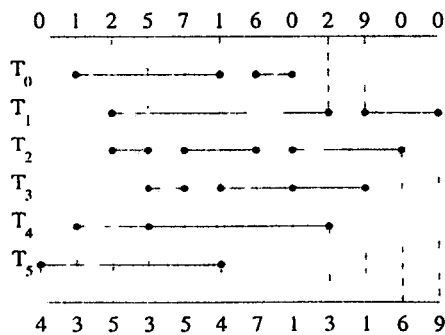
### 5.2 การกำหนดทางเดินภายในช่องทาง (Detailed Routing)

การเชื่อมโยงแบ่งออกเป็น 3 ส่วนคือ การเชื่อมโยงสายสัญญาณระหว่างแถวกับแถว, การเชื่อมโยงสายสัญญาณระหว่างแถวกับแพ็ด (Pad) และการเชื่อมโยงสายสัญญาณเพาเวอร์ (Power) และกราวด์ (Ground) ดังแสดงในรูปที่ 5.5



รูปที่ 5.5 แสดงการเชื่อมโยงสายสัญญาณ

การเชื่อมโยงสายสัญญาณในที่นี้เป็นการวางสายสัญญาณลงบนแพร์คที่อยู่ภายในแกนเนื้อโดยใช้การเชื่อมโยงแบบสองเลเยอร์ (Two Layer) โดยแบ่งเลเยอร์ออกเป็น เวอร์ติคัลเลเยอร์ (Vertical Layer) และฮอริซซอลทัลเลเยอร์ (Horizontal Layer) ระหว่างเลเยอร์เชื่อมต่อกันด้วยเวียร์ (Via) ดังนั้น จึงแบ่งสายสัญญาณออกเป็น เวอร์ติคัลเช็กเมนต์และฮอริซซอลทัลเช็กเมนต์ ในแต่ละเช็กเมนต์ต้องไม่มีการซ้อนทับกันของสายสัญญาณที่อยู่ในเลเยอร์เดียวกัน และต้องสามารถเชื่อมโยงสายสัญญาณ ได้สมบูรณ์ร้อยเปอร์เซ็นต์ ดังแสดงในรูปที่ 5.6



รูปที่ 5.6 แสดงผลลัพธ์ที่ได้จากการเชื่อมโยง

สำหรับในการวิจัยนี้เป็นการศึกษาการเชื่อมโยงสายสัญญาณระหว่างเซลล์กับเซลล์ภายในแถวโดยใช้ Greedy Channel Route Algorithm [27] ซึ่งมีขั้นตอนดังแสดงในรูปที่ 5.7 หลักการของอัลกอริทึมเป็นการเชื่อมโยงในลักษณะคอลัมน์จากซ้ายไปขวา ดังแสดงในรูปที่ 5.8 ไม่ต้องใช้ข้อริชชอลทล็ค็อนสเทรน (Horizontal Constrain) และเวอร์ทิคัลค็อนสเทรน (Vertical Constrain)

```

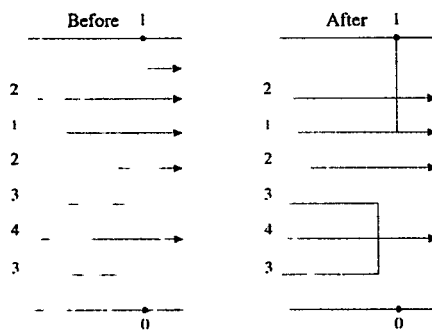
Algorithm Greedy_Channel_Route()

    initial channel width
    initial minimum jog length
    initial steady net constant

    for i = 1 to m
        for j = 1 to n
            make feasible top and bottom connections in minimal
            free up as many tracks as possible by collapsing split nets
            add jogs to reduce the range of split nets
            add jogs to raise rising nets and lower falling nets
            widen channel if infeasible top or bottom connections
            extend to next column
        end for
    end for

End
    
```

รูปที่ 5.7 แสดงอัลกอริทึม Greedy\_Channel\_Route()



รูปที่ 5.8 แสดงการทำงานของ Greedy Route

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 6

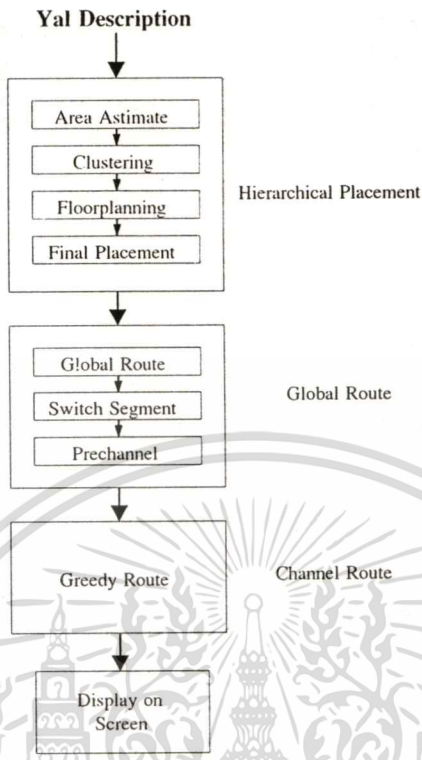
### ทดสอบกับ MCNC Benchmark

#### 6.1 MCNC Benchmark

ในการทดสอบอัลกอริทึมที่ให้แก่ปัญหาการจัดวางและการเชื่อมโยง ใช้วงจรทดสอบจาก MCNC ซึ่งเป็นวงจรทดสอบสำหรับการออกแบบในระดับกายภาพอย่างกว้างขวางสำหรับวงจรแอสแตนคาร์ดเซลที่นำมาเป็นข้อมูลนำเข้าเพื่อทดสอบโปรแกรม Automatic Place and Route ที่พัฒนาขึ้นจาก Visual C++ บน Microsoft Windows นั้น อธิบายอยู่ในรูปของ Yal Description สามารถดูรายละเอียดได้ในภาคผนวก ค. ซึ่งประกอบด้วยวงจรดังต่อไปนี้

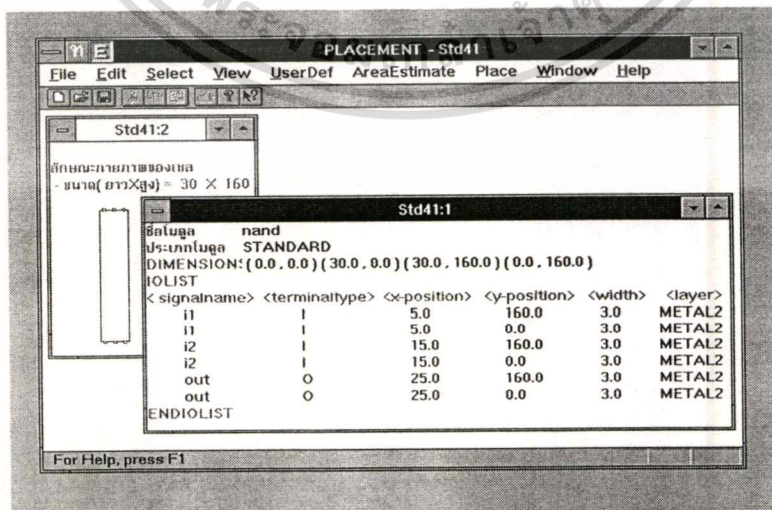
Benchmark	# Cell	# Net	# I/O	Pins/Net	Pins/Cell
SC0 (fract)	125	147	24	2.79	3.63
SC1 (primary 1)	752	985	81	3.35	3.97
SC2 (struct)	1,088	1,920	64	2.82	2.86
SC3 (primary 2)	2,907	3,136	107	3.83	3.99
SC5 (biomed)	6,417	5,742	97	3.88	3.47
SC6 (industry 2)	12,142	13,419	495	3.57	3.95
SC7 (industry 3)	15,059	21,940	375	3.1	4.52

ตารางที่ 6.1 แสดงวงจรทดสอบ



รูปที่ 6.1 แสดงขั้นตอนภายในโปรแกรม Automatic Place and Route

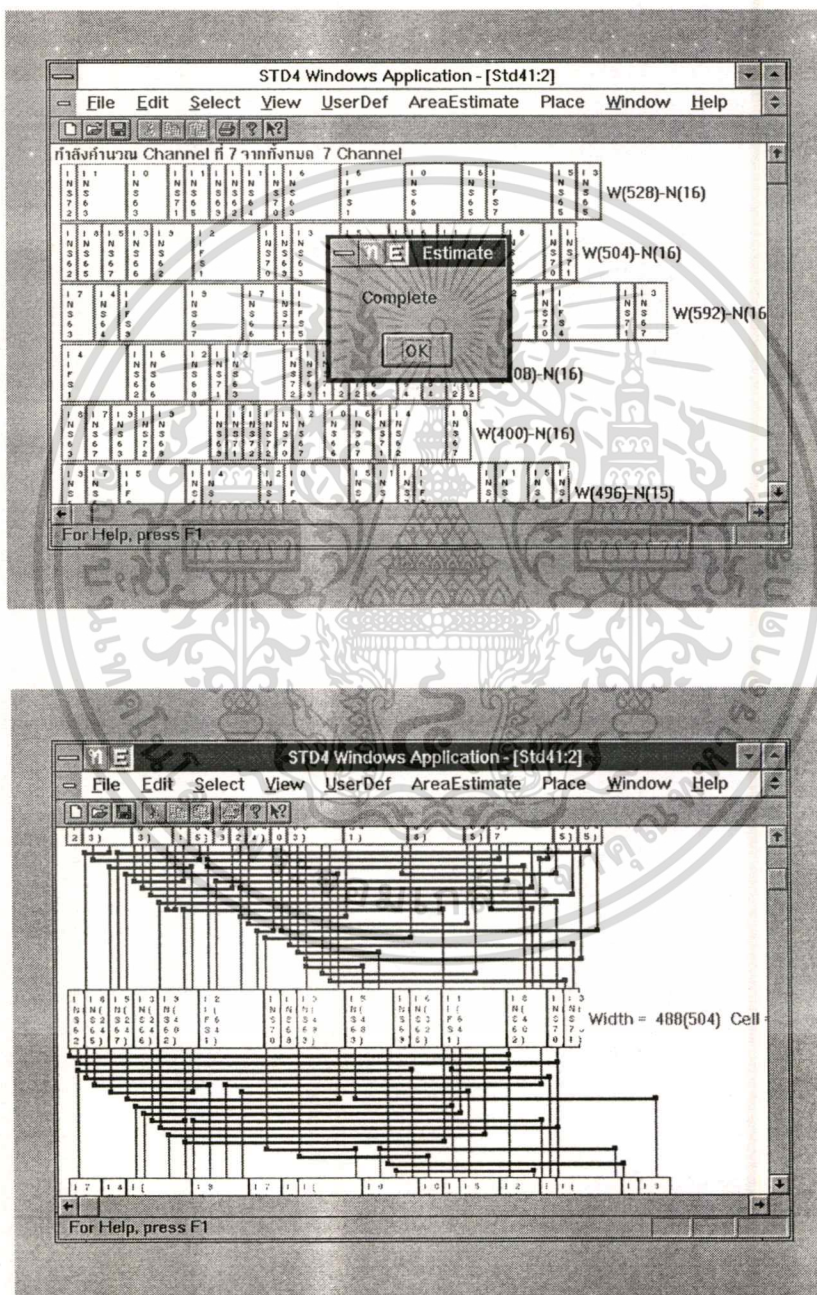
การทดสอบวงจร SCO เริ่มด้วยการไหลวงจรและไลบรารีที่เก็บขนาดและรายละเอียดจำนวนอินพุตเอาต์พุต พร้อมตำแหน่งที่อยู่บนเซลล์ ดังแสดงในรูปที่ 6.2



รูปที่ 6.2 แสดงข้อมูลในไลบรารี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

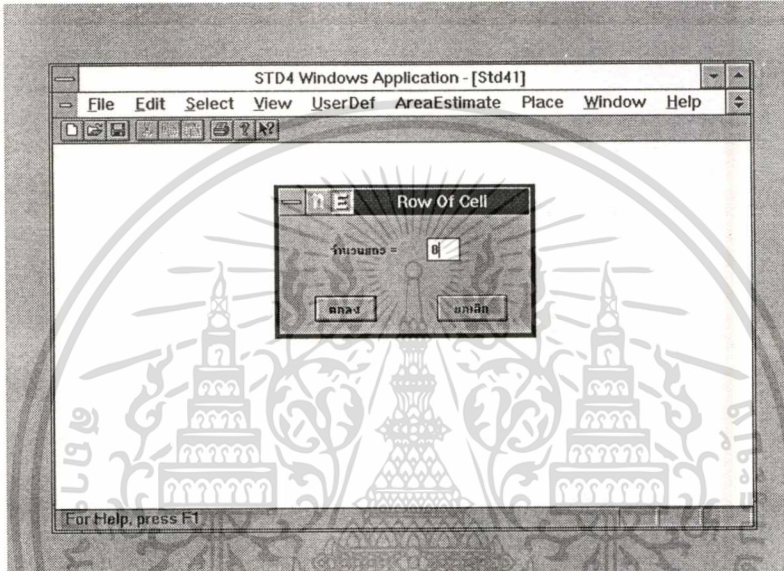
จากนั้นกำหนดจำนวนแถวของเซลล์สำหรับเลเอาท์ที่ต้องการ เพื่อใช้ในการคำนวณหาพื้นที่โดยประมาณ โดยการจัดวางและเชื่อมโยง โดยไม่มีข้อกำหนด ซึ่งผลลัพธ์ที่ได้จะเป็นค่าประมาณพื้นที่ของเลเอาท์และ Aspect Ratio ดังแสดงในรูปที่ 6.3 ก่อนทำการจัดวางและเชื่อมโยงจริง



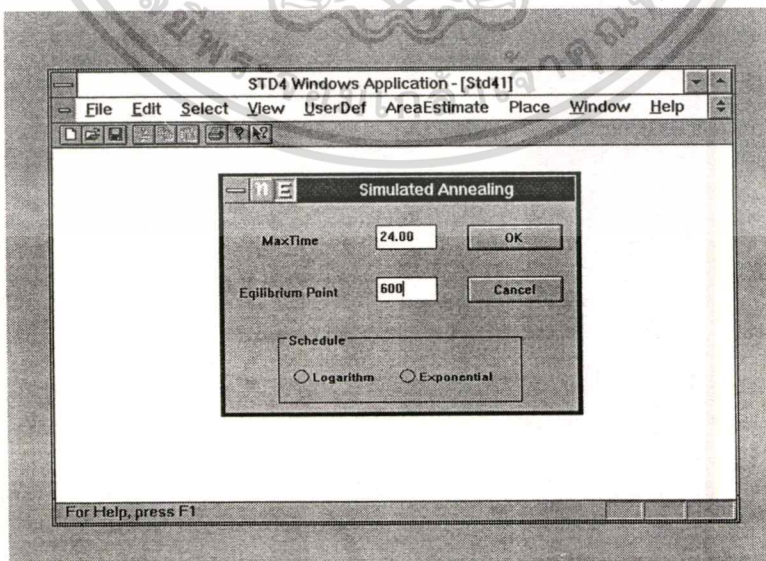
รูปที่ 6.3 แสดงการคำนวณพื้นที่โดยประมาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการจัดวางแบบลำดับชั้นเริ่มต้นจากการจัดกลุ่มของเซลล์ตามจำนวนแถวที่ระบุ พร้อมกำหนดค่าอุณหภูมิเริ่มต้นและจำนวนทรานซิสชันของสิมูเลทอัลเนลิ่งอัลกอริทึม ดังแสดงในรูปที่ 6.4 (a) และ (b) และเมื่อภายหลังจบการทำงานของสิมูเลทอัลเนลิ่งอัลกอริทึม สามารถตรวจสอบการทำงานของอัลกอริทึมเป็น Hill Climbing หรือไม่โดยพิจารณาจากรูปภาพของฟังก์ชันเป้าหมาย ดังแสดงในรูปที่ 6.5

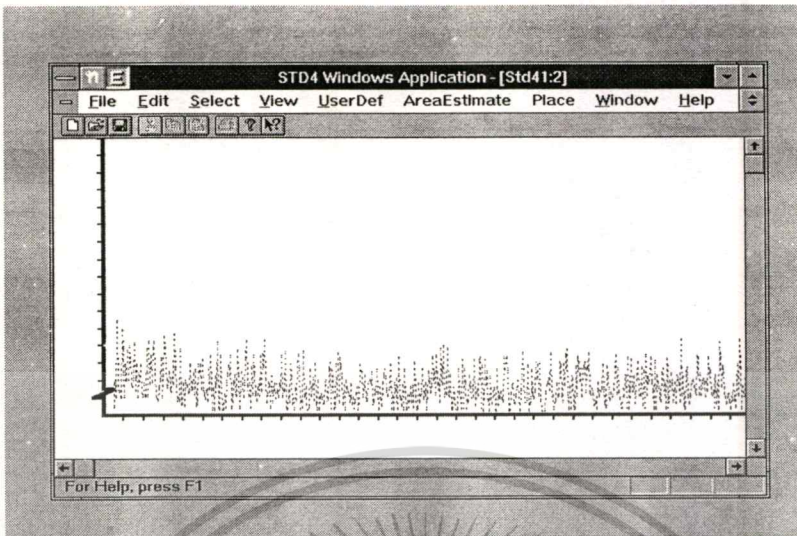


รูปที่ 6.4 (a) แสดงการกำหนดจำนวนแถวของเซลล์



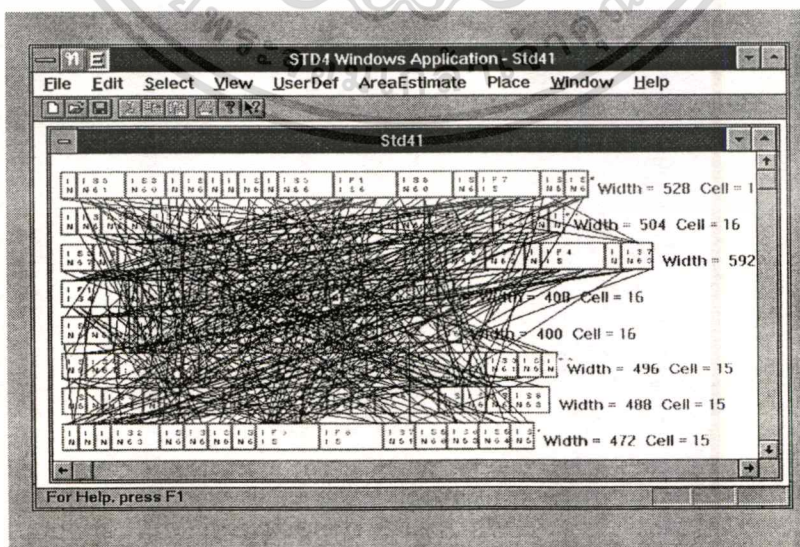
รูปที่ 6.4 (b) แสดงการกำหนดค่าอุณหภูมิเริ่มต้นและจำนวนทรานซิสชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



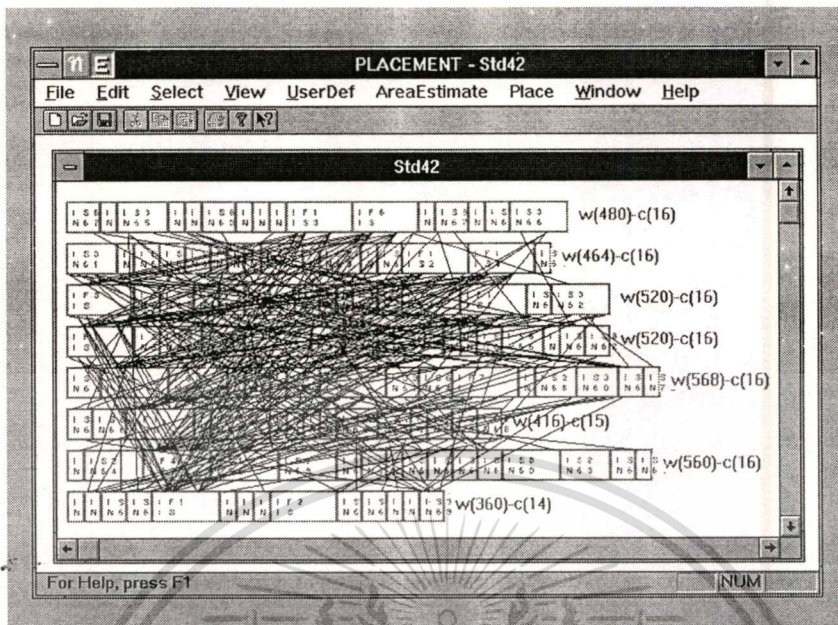
รูปที่ 6.5 แสดงกราฟฟังก์ชันเป้าหมายของสัญญาณดิจิตอลของสัญญาณ

จากการจัดกลุ่มแต่ละกลุ่มนั้นจะมีจำนวนเซลล์ไม่เท่ากัน เนื่องจากการจัดกลุ่มเป็นการจัดเซลล์ที่มีความสัมพันธ์กันให้อยู่ภายในกลุ่มเดียวกัน ดังแสดงในรูปที่ 6.6 (a) จะเห็นได้ว่าการจัดวางก่อนการแบ่งกลุ่มการกระจายของสายสัญญาณที่เชื่อมโยงระหว่างเซลล์จะกระจายมากกว่าการจัดวางหลังจากการจัดกลุ่มแล้ว ดังแสดงในรูปที่ 6.6 (b) ภายหลังจากการจัดกลุ่มแล้วจำเป็นต้องวางผังของเซลล์ใหม่เพื่อเป็นการลดเนื้อที่ว่างเปล่าและควบคุม Aspect Ratio เพื่อให้ได้เลเอาท์ตามที่ต้องการ ดังแสดงในรูปที่ 6.7

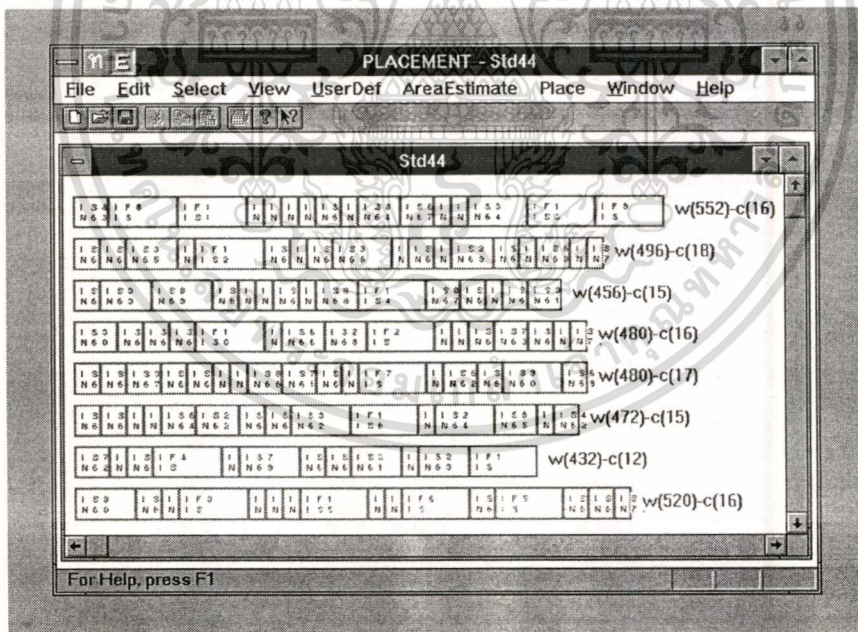


รูปที่ 6.6 (a) แสดงการจัดวางก่อนการแบ่งกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติเท่านั้น มิใช่ให้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



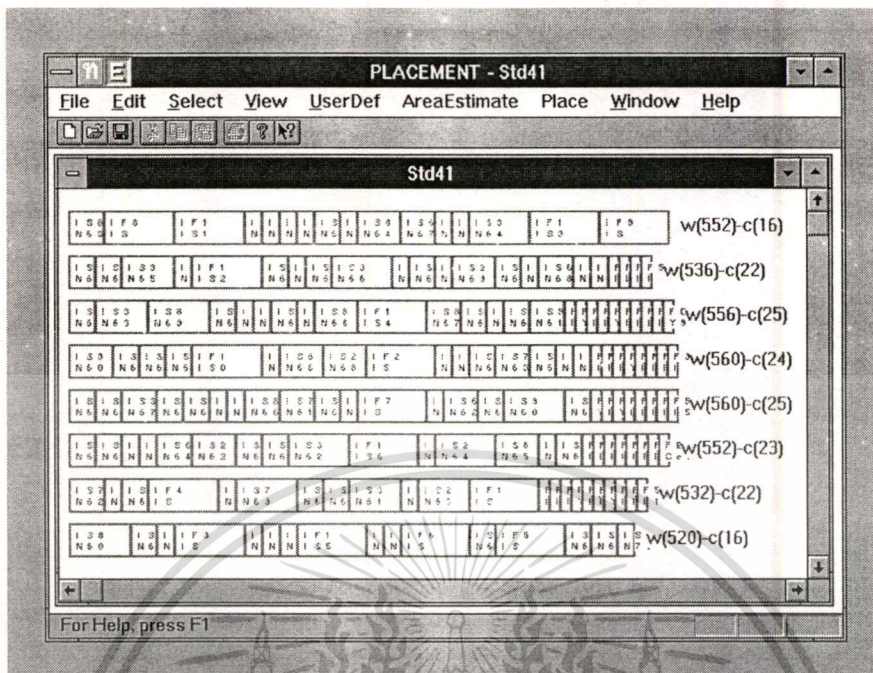
รูปที่ 6.6 (b) แสดงการจัดวางหลังการแบ่งกลุ่ม



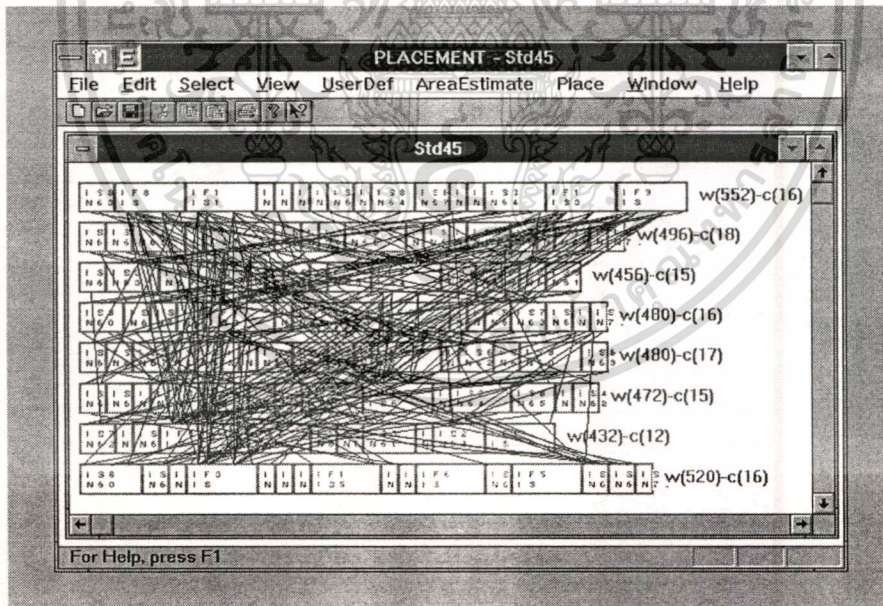
รูปที่ 6.7 แสดงการวางผัง

ขั้นตอนสุดท้ายเป็นการจัดวางเซลล์ภายในแถว เพื่อกำหนดตำแหน่งของเซลล์ในแถว นั้น ๆ ดังแสดงในรูปที่ 6.8 (a) เมื่อพิจารณาในรูปที่ 6.8 (b) จะเห็นได้ว่าการกระจายของสายสัญญาณเป็นระเบียบมากกว่าในรูปที่ 6.6 (b)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของบริษัทฯ ในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



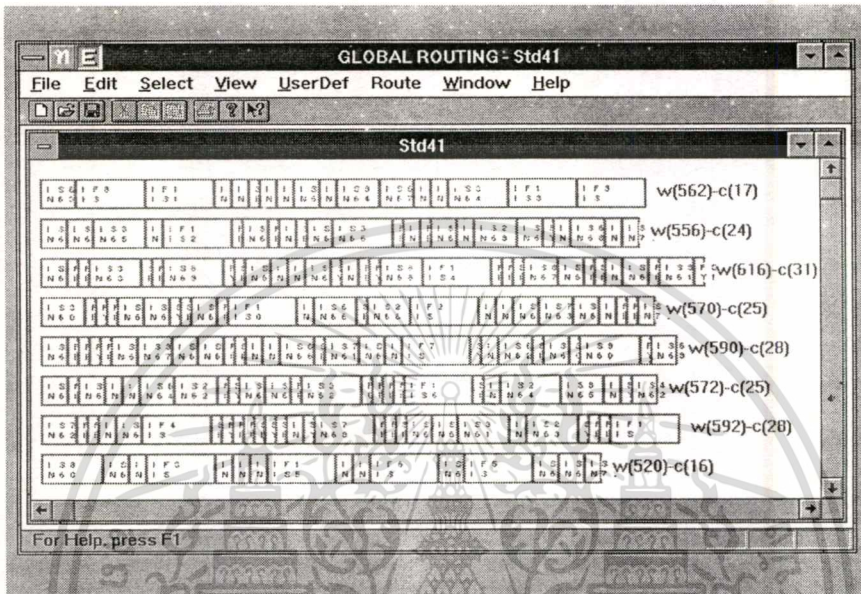
รูปที่ 6.8 (a) แสดงการจัดวาง



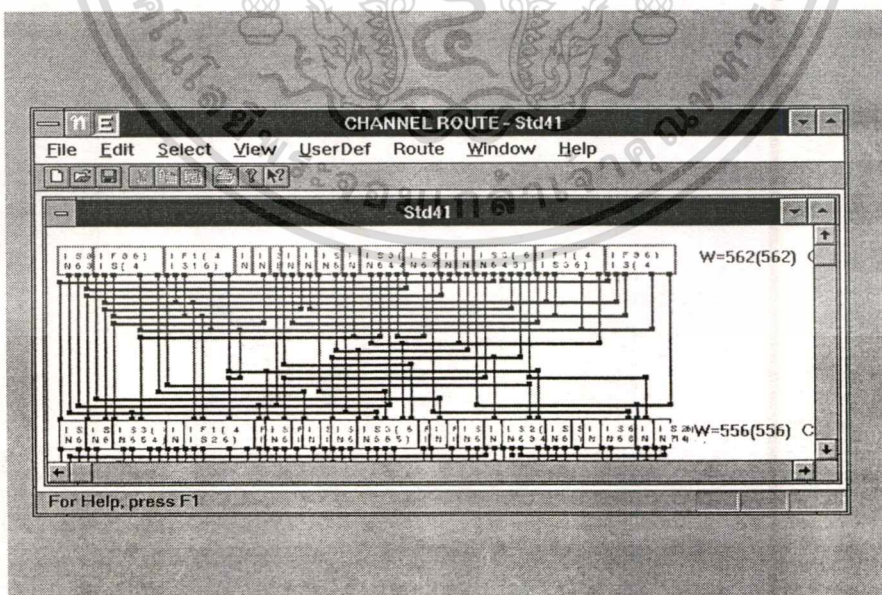
รูปที่ 6.8 (b) แสดงการจัดวางพร้อมคัทพีด

ส่วนการเชื่อมโยงจะนำเอา Placement List ที่ได้จากการจัดวางมากำหนดทางเดินซึ่งเรียกว่า Global Routing ในขั้นตอนนี้ต้องมีการเพิ่ม Feedthrough Cell ลงไปยังแถวตามเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งที่หาได้จากการกำหนดโดย MST หรือ MRST พร้อมปรับตำแหน่งของเซลล์อีกครั้งหนึ่ง ดังแสดงในรูปที่ 6.9 ตามด้วยการกำหนดทางเดินภายในแกนเน็ตด้วยอัลกอริทึมของ Greedy Channel Route ผลลัพธ์ดังแสดงในรูปที่ 6.10



รูปที่ 6.9 แสดงการกำหนดช่องทาง



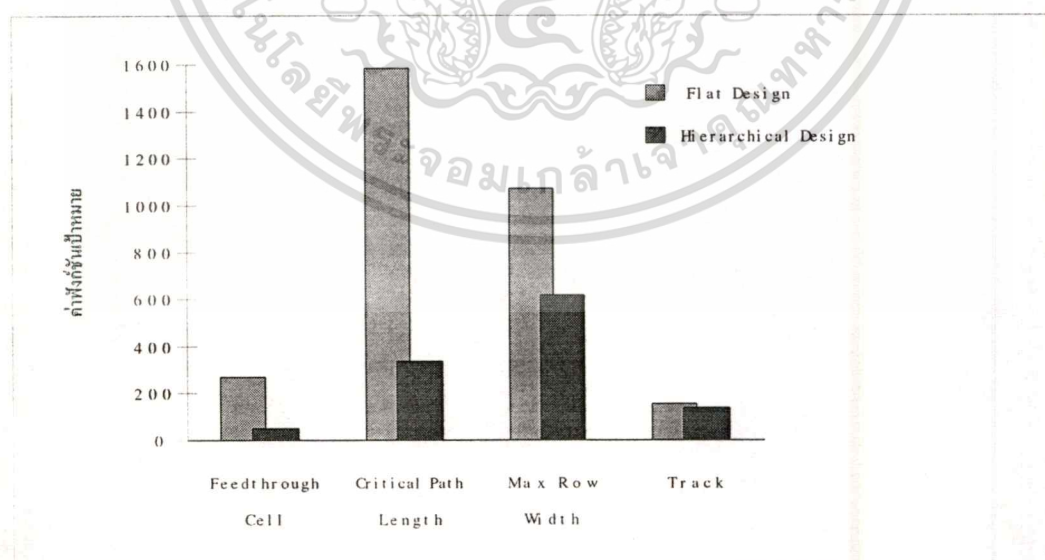
รูปที่ 6.10 แสดงการเชื่อมโยงภายในแกนเน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้เห็นถึงการจัดวางแบบลำดับชั้นมีประสิทธิภาพสูงกว่าการจัดวางแบบเดิม โดยกำหนดให้ความยาวของสายสัญญาณรวมเป็นฟังก์ชันเป้าหมายเพียงฟังก์ชันเดียว ส่วนการจัดวางแบบลำดับชั้นจะใช้หลักการตามที่กล่าวมาตั้งแต่ต้น โดยทดลองกับวงจร SCO โดยกำหนดให้จำนวนแถวของเซลล์เท่ากับ 8 แถว ซึ่งปรากฏว่าการจัดวางแบบลำดับชั้นจะให้จำนวนของพีครูซูล ความยาวของคลิสิกัลพาธ ความยาวของแถวที่ยาวที่สุดที่ทำให้เกิดเนื้อที่ว่างเปล่าท้ายแถวและจำนวนแทร็คน้อยกว่าการจัดวางแบบเดิมซึ่งพิจารณาเฉพาะเนื้อที่ของเลเอาต์เพียงอย่างเดียว ดังแสดงในตารางที่ 6.2 และเมื่อนำมาเปรียบเทียบกับกราฟจะได้ดังแสดงในรูปที่ 6.11

	Flat Design	Hierarchical Design
Feedthrough Cell	269 Cell	51 Cell
Critical Path Length	$1585 \times 10^{-6}$ m	$335 \times 10^{-6}$ m
Max Row Width	$1074 \times 10^{-6}$ m	$616 \times 10^{-6}$ m
Track	160	140

ตารางที่ 6.2 แสดงการเปรียบเทียบการจัดวางแบบ Hierarchical Design กับ Flat Design ของวงจร SCO



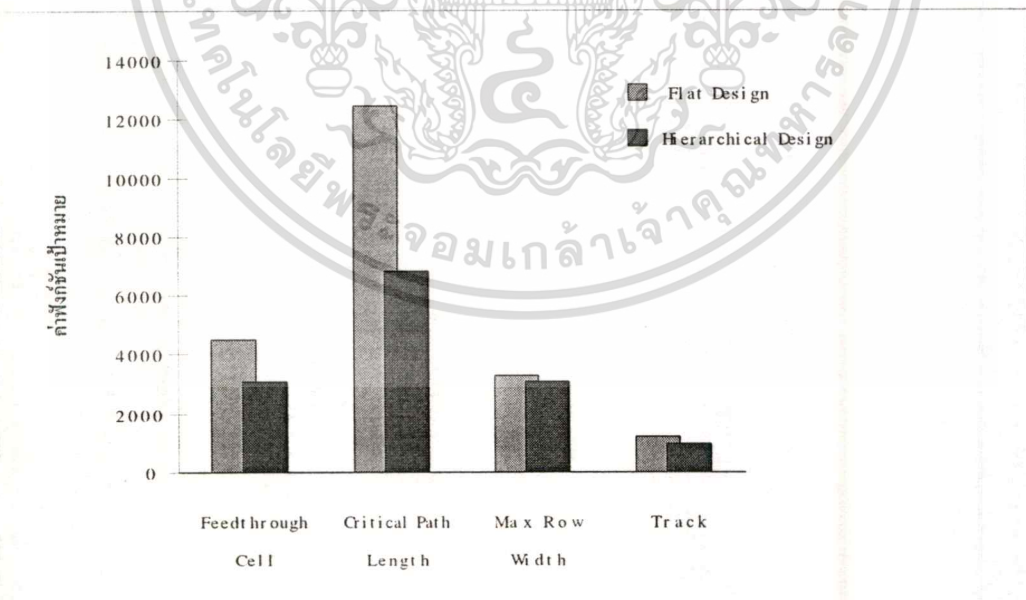
รูปที่ 6.11 กราฟแสดงการเปรียบเทียบการจัดวางแบบ Hierarchical Design กับ Flat Design ของวงจร SCO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการทดลองกับวงจรอื่นตามตารางที่ 6.1 ปรากฏว่าการทำงานของโปรแกรม ทดสอบ Automatic Place and Route สามารถรองรับการทำงานของวงจร SC2 ได้สูงสุด เนื่องจากเครื่องไมโครคอมพิวเตอร์ที่ใช้ทดสอบมีหน่วยความจำหลักเพียง 16 MB เท่านั้น ผลการทดลองจากการจัดวางเซลล์จำนวน 16 แถว ปรากฏว่าการจัดวางแบบลำดับชั้นมี ประสิทธิภาพสูงกว่าการจัดวางแบบเดิม ดังแสดงในตารางที่ 6.3 และกราฟแสดงการ เปรียบเทียบดังแสดงในรูปที่ 6.12

	Flat Design	Hierarchical Design
Feedthrough Cell	4517 Cell	3112 Cell
Critical Path Length	$12449 \times 10^{-6} \text{ m}$	$6809 \times 10^{-6} \text{ m}$
Max Row Width	$3292 \times 10^{-6} \text{ m}$	$3102 \times 10^{-6} \text{ m}$
Track	1200	993

ตารางที่ 6.3 แสดงการเปรียบเทียบการจัดวางแบบ Hierarchical Design กับ Flat Design ของวงจร SC2



รูปที่ 6.12 กราฟแสดงการเปรียบเทียบการจัดวางแบบ Hierarchical Design กับ Flat Design ของวงจร SC2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 7

### บทสรุป

#### 7.1 สรุปผลการวิจัย

การแก้ปัญหาการออกแบบในลักษณะกายภาพต้องคำนึงถึงรูปแบบของเลเอาท์และวิธีการออกแบบเป็นหลัก สำหรับวงจรรวมเสตนคาร์ดเซลรูปแบบของเลเอาท์ประกอบด้วยแถวของเซล ระหว่างแถวเป็นพื้นที่สำหรับการเชื่อมต่อสายสัญญาณ ซึ่งใช้วิธีการออกแบบเซมิคัสตอมหรือเซลเบส ผู้ออกแบบไม่ต้องออกแบบทางกายภาพในระดับเซลเอง เพียงแต่นำเซลที่อยู่ในไลบรารีมาจัดวางและเชื่อมต่อ

ในอดีตที่ผ่านมาการออกแบบเลเอาท์มุ่งเน้นการใช้เนื้อที่ของเลเอาท์เพียงวัตถุประสงค์เดียว เลเอาท์ที่ได้จากการออกแบบจึงไม่มีประสิทธิภาพต่อการใช้งาน ถึงแม้จะได้เนื้อที่ของเลเอาท์น้อยลงก็ตาม การพิจารณาวัตถุประสงค์ของเนื้อที่เพียงอย่างเดียวไม่เพียงพอ จึงได้นำเอาวัตถุประสงค์อื่นที่เกี่ยวข้องกับการแก้ปัญหาการออกแบบในระดับกายภาพมาพิจารณาร่วมด้วย โดยมีได้มุ่งเน้นวัตถุประสงค์ใดวัตถุประสงค์หนึ่งเป็นหลักซึ่งประกอบด้วย Aspect Ratio, การใช้เนื้อที่, คลิทธิคัลพาร, การเชื่อมต่อที่สมบูรณ์, เวลาที่ใช้ในการแก้ปัญหาและจำนวนเซลไปพร้อม ๆ กัน

ในการแก้ปัญหาการจัดวางได้ประยุกต์การออกแบบวงจรรวมแบบลำดับชั้นมาใช้ในการจัดวางแบบเดิมเพื่อให้เป็นการจัดวางแบบลำดับชั้น โดยแบ่งขั้นตอนออกเป็นการจัดกลุ่มเซล (Cell Clustering) การวางผังเซล (Cell Floorplanning) และการวางเซล (Cell Final Placement) โดยใช้สิมูเลทเตดอัลเนลลิจอัลกอริทึม เป็นอัลกอริทึมหลักในการค้นหาผลลัพธ์ในการแบ่งกลุ่มเซลได้เสนอแนวความคิดใหม่โดยใช้ทไฟดเป็นฟังก์ชันเป้าหมายแทนคัทไลน์ซึ่งเหมาะสมต่อการออกแบบเลเอาท์สำหรับเสตนคาร์ดเซลมากกว่า เป็นการกระจายสายสัญญาณเพื่อมิให้เกิด Routing Congestion พร้อมทั้งเพิ่มฟังก์ชันเป้าหมาย ความยาวของคลิทธิคัลพาร เพื่อควบคุมคลิทธิคัลพารมิให้เกิดปัญหาในเรื่องประสิทธิภาพทางไฟฟ้า ส่วนการวางผังเซล ได้นำเนื้อที่ว่างเปล่ามาเป็นฟังก์ชันเป้าหมาย เพื่อเป็นการควบคุมเนื้อที่ว่างเปล่าท้ายแถวให้เหลือน้อยที่สุดและใกล้เคียงกับ Aspect Ratio ที่คำนวณได้จากการคำนวณพื้นที่โดยประมาณ ซึ่งใช้การจัดวางและการเชื่อมต่อแบบง่ายและไม่มีข้อกำหนดแทนการคำนวณจากสูตรทางคณิตศาสตร์ ซึ่งส่งผลให้การคำนวณใกล้เคียงกับความเป็นจริงมากขึ้น หลังจากนั้น

เป็นขั้นตอนสุดท้ายคือการวางเซลล์ภายในแถว โดยใช้ความยาวรวมของสายสัญญาณที่ได้จากการประมาณด้วยวิธี Half Perimeter เป็นฟังก์ชันเป้าหมาย

ส่วนการแก้ปัญหาการเชื่อมโยงได้แบ่งขั้นตอนออกเป็น การกำหนดช่องทางเดินของสายสัญญาณ และการกำหนดทางเดินภายในช่องทาง ในการกำหนดช่องทางเดินของสายสัญญาณจะใช้ทั้ง MST และ MRST จะใช้ MRST ก็ต่อเมื่อสายสัญญาณนั้นเป็นคลิกคัลพาร นอกนั้นจะใช้ MST แทน ทั้งนี้เนื่องจากการใช้ MRST ต้องมีการเพิ่มซิทเนอร์พอยท์เข้าไป การเพิ่มซิทเนอร์พอยท์ เปรียบเสมือนการเพิ่มพีคเจอร์เซลล์เข้าไปในแถวซึ่งไม่เหมาะสมต่อการออกแบบเลเอาท์สำหรับแอสแตร์คาร์ดเซลล์

สำหรับขั้นตอนสุดท้ายเป็นการกำหนดทางเดินออกภายในช่องทางโดยสายสัญญาณจะถูกแบ่งเป็นเวอร์ติคัลเช็กเมนต์และฮอริซซอลทัลเช็กเมนต์ และกำหนดลงบนแทร็คที่อยู่ภายในแชนเนลโดยใช้ Greedy Channel Route Algorithm ซึ่งเป็นอัลกอริทึมที่มีการเชื่อมโยงในลักษณะคอลัมน์โดยไม่ต้องใช้เวกทิคัลค้อนสเตรนและฮอริซซอลทัลค้อนสเตรนแต่อย่างใด อีกทั้งสามารถขยายจำนวนแทร็คภายในแชนเนลได้เมื่อไม่พอ เป็นการรับประกันได้ว่าการเชื่อมโยงสามารถทำได้ครบสมบูรณ์ร้อยเปอร์เซ็นต์

จากการทดลองการจัดวางแบบลำดับชั้นกับวงจร SCO จะให้จำนวนพีคเจอร์เซลล์ความยาวของคลิกคัลพาร เนื้อที่ว่างเปล่าท้ายแถวและจำนวนแทร็คลดลงเมื่อเปรียบเทียบกับ การจัดวางแบบเดิมที่ใช้ฟังก์ชันเป้าหมายที่มุ่งเน้นเนื้อที่ของเลเอาท์เพียงประการเดียว

## 7.2 ปัญหาที่เกิดขึ้นและข้อเสนอแนะ

เนื่องจากการศึกษาการออกแบบเลเอาท์ในระดับกายภาพสำหรับวงจรรวมในประเทศไทยยังเป็นเรื่องใหม่มีผู้ให้ความสนใจน้อย หนังสือและบทความทางวิชาการที่กล่าวถึงวิธีการแก้ปัญหาการออกแบบเลเอาท์ในขั้นตอนต่าง ๆ หาได้ยากมากโดยเฉพาะบทความในการประชุมทางวิชาการเช่น Design Automation Conference เป็นบทความที่ถูกกล่าวอ้างเสมอแต่ไม่สามารถหาการศึกษาได้ ดังนั้น ควรมีการจัดหาหนังสือและบทความที่เป็นวารสารต่างประเทศที่กล่าวมาข้างต้น เพื่อให้ผู้ที่สนใจใช้เป็นพื้นฐานในการศึกษาทฤษฎีได้อย่างถูกต้อง

ในการวิจัยนี้โปรแกรมที่พัฒนาขึ้นยังไม่สามารถนำไปใช้กับการออกแบบจริงได้ทั้งหมด โดยเฉพาะการเชื่อมโยง จุดมุ่งหมายเพื่อใช้ทดสอบอัลกอริทึมและฟังก์ชันเป้าหมายเท่านั้น แต่ในอนาคตสามารถพัฒนาขึ้นมาใช้กับงานจริงได้ และเป็นการพัฒนาอุตสาหกรรมซอฟต์แวร์ภายในประเทศเพื่อให้มีการแข่งขันกับต่างประเทศต่อไป

## เอกสารอ้างอิง

- [1] Alexandria Virginia, "Understanding Computers the Chipmaker", The Editors of Time-Life Books, Time-Life Books., 1988.
- [2] Stanley L. Hurst, "Custom VLSI Microelectronics", Prentice hall International (UK) Ltd., 1992.
- [3] บวร ปภัสราทรม ประเสริฐ คันธมานนท์ และ สุเมธ อังคะศิริกุล, "เทคโนโลยีการออกแบบวงจรรวม", 2533.
- [4] William P. Swartz, "Automatic Layout of Analog and Digital Mixed Macro/Standard Cell Integrated Circuits", Yale University, May 1993.
- [5] John. P. Uyemura, "Physical Design of CMOS Integrated Circuits Using L-EDIT™", PWS Publishing Company., 1995.
- [6] ดร.สุชาย ชนวเสถียร และคนอื่น ๆ, การออกแบบวงจรรวม VLSI Design, (กรุงเทพฯ : ศรีอนันต์การพิมพ์, 2535)
- [7] John P. Huber and Mark W. Rosneck, "Successful ASIC Design the First Time Through", Van Nostrand Reinhold, 1991.
- [8] P. J. Hicks, "Semi-Custom IC Design and VLSI", Peter Peregrinus Ltd., 1983.
- [9] Malcolm R. Haskard, "An Introduction to Application Specific Integrated Circuits", Prentice Hall of Australia Pty Ltd., 1990.
- [10] Sadiq M. Sait and Habib Youssef, "VLSI Physical Design Automation", McGraw-Hill Book Company. 1995.
- [11] Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", The Benjamin/Cummings Publishing Company, Inc., 1993.
- [12] Sara Baase, "Computer Algorithms Introduction to Design and Analysis", Addison-Wesley Publishing Company, 1989.
- [13] P. R. Adby, M. A. H. Dempstek, "Introduction to Optimization Methods, London Chapman and Hall, 1974.
- [14] H. W. Leong University of Singapore and C. L. Lin University of Illinois, "Simulated Annealing and its Applications to VLSI Layout", Urbana, IL 61801, USA., pp.276-287.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- [15] S.Kirkpatrick, S.,C.D. Gelatt and M.P. Vecchi, "Optimization\* by Simulated Annealing", Science, Vol. 220, pp. 671-680, May 1983.
- [16] L. Ingber, "Simulated Annealing: Practice Versus Theory", Lester Ingber Research, McLean, VA, 1993.
- [17] Emile Aarts and Jan Korst, "Simulated Annealing and Boltzmann Machines", John Wiley & Sons, 1989.
- [18] Y. S. Cheung, Y. W. Lee and C. S. K. Yeung, "Hierarchical Simulated Annealing for Standard Cell Placement", Proc. ISIC-91, No.2, pp.17-22.
- [19] Wern-Jieh Sun and Carl Sechen, "Efficient and Effective Placement for Very Large Circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 14, No. 3, pp.349-359, March 1995.
- [20] Fadi J. Kurdahi and Alice C. Parker, "Techniques for Area Estimation of VLSi Layouts", IEEE Transactions on Computer-Aided Design, Vol. 8, No.1, January 1989.
- [21] Bryan T. Preas and Michael J. Lorenzetti, "Physical Design Automation of VLSI Systems", The Benjamin/Cummings Publishing Company, Inc. 1988.
- [22] Yen-Chuen Wei and Chung-Kuan Cheng, "Ratio Cut Partitioning for Hierarchical Designs", IEEE Transactions on Computer-Aided Design, vol.10, No.7, pp.911-921, July 1991.
- [23] Chung-Kuan Cheng and Yen-Chuen Wei, "An Improved Two-way Partitioning Algorithm with Stable Performance", IEEE Transactions on Computer-Aided Design vol.10 No.12, pp.1502-1511, December 1991.
- [24] Laura A. Sanchis, "Multiple-Way Network Partitioning", IEEE Transactions on Computers vol.38, No.1, pp.62-81, January 1989.
- [25] Michael Pecht, "Placement and Routing of Electronic Models", Marcel Dekker, Inc., 1993.
- [26] Jeff Griffith, Gabriel Robins, Jeffery S. Salowe and Tongtong Zhang, "Closing the Gap : Near-Optimal Stener Trees in Polynomial Time", Department of Computer Science, University of Virginia, Charlottesville, VA 22903-2442.

- [27] T. Ohtsuki, "Layout Design and Verification", Elsevier Science Publishing Company.Inc., 1986.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ภาคผนวก ก. ผลงานวิจัยที่ได้รับการตีพิมพ์

- [1] ทรงพล ใหม่สาธิตี, อาจารย์บรรจง ปิยธำรง, “การจัดวางและเชื่อมโยงแอสตนคาร์คเซลดโดยอัตโนมัติ”, การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 18, มหาวิทยาลัยเทคโนโลยีมหานคร หน้า 448 - 454, 22 - 24 พฤศจิกายน 2538.



## ภาคผนวก ข. โปรแกรมการทดลอง

```
// areaview.cpp : implementation file //

#include "stdafx.h"
#include "std4.h"
#include "std4doc.h"
#include "areaview.h"
#include "time.h"
#include "scmos.h"

void AreaView::EstimateWidth()
{
    int width;
    CString Name;
    NetworkList *TmpListHead;
    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();
    pDoc->EstimateCoreWidth = 0;

    // ตั้ง LINK NetworkListHead
    TmpListHead = pDoc->NetworkListHead;
    while (TmpListHead != NetworkListNull) {
        Name = TmpListHead->ModuleName;
        width = pDoc->WidthInLib(Name);
        if ( width > 0 )
            pDoc->EstimateCoreWidth += width;
        else {
            MessageBox(Name," Not Found In Library ",MB_OK);
            break;
        }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        TmpListHead = TmpListHead->Link;
    }
    width = pDoc->EstimateCoreWidth%pDoc->NumOfRow;
    pDoc->EstimateCoreWidth = (pDoc->EstimateCoreWidth/pDoc->NumOfRow) + width;
}

```

```

void AreaView::EstimateHeight(CDC* pDC)
{
    // Clear Old File
    remove("CHANNEL.$$$"); /* open file And Clear Data Initial Placement */
    //
    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();
    pDoc->EstimateCoreHeight = 0;
    Status = ChannelNo = 1;
    R1TmpRowHead = R2TmpRowHead = pDoc->RowListHead;
    while (R2TmpRowHead->Link != NULL) {
        // Draw Channel
        ViewChannel(pDC);
        FeedThrough = InChannel = OutChannel = 0;
        R2TmpRowHead = R1TmpRowHead->Link;
        // step 1. Find Signal In Channel
        ChanNel(1);    // 1 = Top
        ChanNel(2);    // 2 = Bottom
        // step 2. Left Edge Algorithm (Caculate Channel)
        LeftEdge();
        // Skip
        R1TmpRowHead = R2TmpRowHead;
        ChannelNo += 1;
        Status += 1;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pDoc->EstimateCoreHeight += pDoc->NumOfRow* pDoc->CellHeight;
}

void AreaView::LeftEdge()
{
    char          TrackNo,MinFound;
    int           X1,X2;
    HorSegmentList  TmpHorSegment,*SeedPoint,*BeginSort,*MinSegment,*Select;
    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();
    if (HorSegmentHead == NULL)
        return;
    PreLeftEdge();
    // Step 1. Sort By Location
    SeedPoint = HorSegmentHead;
    while ( SeedPoint->Link != NULL) {
        MinFound = 0;
        MinSegment = SeedPoint;
        X1 = SeedPoint->X1Location;
        BeginSort = SeedPoint->Link;
        while( BeginSort != NULL) {
            if ( BeginSort->X1Location < X1) {
                MinFound = 1;
                X1 = BeginSort->X1Location;
                MinSegment = BeginSort;
            }
            BeginSort = BeginSort->Link;
        }
        // step 2. Adjust Link List
        if ( MinFound ) {
            strcpy(TmpHorSegment.SignalName,SeedPoint->SignalName);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    TmpHorSegment.X1Location = SeedPoint->X1Location;
    TmpHorSegment.X2Location = SeedPoint->X2Location;
    strcpy(SeedPoint->SignalName,MinSegment->SignalName);
    SeedPoint->X1Location = MinSegment->X1Location;
    SeedPoint->X2Location = MinSegment->X2Location;
    strcpy(MinSegment->SignalName,TmpHorSegment.SignalName);
    MinSegment->X1Location = TmpHorSegment.X1Location;
    MinSegment->X2Location = TmpHorSegment.X2Location;
}
SeedPoint = SeedPoint->Link;
}
// step 3. Assigned To Track
TrackNo = 1;
SeedPoint = HorSegmentHead;
while ( SeedPoint != NULL) {
    if ( SeedPoint->TrackNo == 0 ) {
        SeedPoint->TrackNo = TrackNo;
        SeedPoint->BenchNo = 2;
        X1 = SeedPoint->X1Location;
        X2 = SeedPoint->X2Location;
        Select = SeedPoint->Link;
        while( Select != NULL) {
            if ( ( Select->X1Location == X1 ) && ( Select->X2Location
                == X2 ) && (Select->TrackNo == 0) ) {
                Select->TrackNo = -1; // OverLap
                Select->BenchNo = 3;
                X1 = Select->X1Location;
                X2 = Select->X2Location;
            }
}

```

```

else
    if ( ( Select->X1Location > X2 ) && (Select->
        TrackNo == 0) ){
        Select->TrackNo = TrackNo;
        Select->BenchNo = 2;
        X1 = Select->X1Location;
        X2 = Select->X2Location;
    }
    Select = Select->Link;
}
TrackNo += 1;
}
SeedPoint = SeedPoint->Link;
}
// step 4. Estimate Height
pDoc->EstimateCoreHeight += (TrackNo+1)*(int)VSpaceLayer1 ;
// step 5. Write To File
WriteChannel();
// step 5. Free List
FreeHorSegment();
}

```

```

// cluster.cpp : implementation file
//

#include "stdafx.h"
#include "std4.h"
#include "std4doc.h"
#include "cluster.h"

//-----
double ClusterView::Annealing(CDC* pDC)
{
    FILE *stream;
    int equilibrium,FixedCost;
    double oldcost,newcost,goodcost,goodcf;
    double T;
    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();
    // writ cost fro create graph
    if ((stream = fopen("COST,$$$", "wb")) == NULL)
    {
        fprintf(stderr, "Cannot open output file.\n");
        return 0;
    }
    T = pDoc->MaxTime;
    equilibrium = pDoc->MaxEqui;
    StableCost = FixedCost = 0;
    oldcost = newcost = goodcost = 0;
    if ( !T && !equilibrium)
        return 0;
    // Sm Algorithm start
    InitState());

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

goodcost = oldcost = CostFun(pDC);
goodcf = newcf;
fwrite(&oldcost, sizeof(oldcost), 1, stream);
GoodState();
while (T >= 1.0) {
    // satisfied point
    while (equilibrium != 0) {
        // equilibrium point
        ExchangeState();
        newcost = CostFun(pDC);
        pDoc->ViewRowArea(10, 20, pDC);
        if (newcost < goodcost) {
            goodcost = newcost;
            FixedCost = 0;
            StableCost = 0;
            if (CriticalAccept && (newcf < goodcf)) {
                goodcf = newcf;
                GoodState();
            }
        }
        else
            FixedCost++;
        if (FixedCost > 2*pDoc->MaxEqui)
            StableCost = 1;
        fwrite(&newcost, sizeof(newcost), 1, stream);
        pDoc->DispState(pDC ,T,equilibrium,newcost,goodcost);
        if (pDoc->AcceptState(newcost,oldcost,T))
            oldcost=newcost; // accept this state
        else
            OldState(); // not accept new state
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        equilibrium--;
    }

    equilibrium = pDoc->MaxEqui;
    //equilibrium = equilibrium-(equilibrium*0.5);
    T=pDoc->UpdateTemp(T);
}

pDoc->DispState(pDC ,T,equilibrium,newcost,goodcost);

fclose(stream); /* close file */

return goodcost;
}

```

```

double ClusterView::CostFun(CDC* pDC)
{
    double cost;
    char Pin,First;
    int XPosition,YPosition;
    long int CL;
    unsigned int CF;

    RECT Box;
    CString SignalName,InstanceName,Name;
    NetworkList *TmpNet;
    SignalList *TmpSignalList;
    RouteList *TmpRouteList;
    PinList *TmpPinList;

    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();
    TmpRouteList = pDoc->RouteListHead;
    if (InitCost) {
        InitCost = 0; // Off
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(TmpRouteList != NULL) {
    First = 1;
    Pin = 0;
    SignalName = TmpRouteList->SignalName;
    TmpPinList = PinListHead = pDoc->AllocPin(TmpRouteList->
        NumOfPin);
    while(Pin<TmpRouteList->NumOfPin) {
        pDoc->XYPlace(SignalName,&InstanceName,&
            XPosition,&YPosition,First);
        strcpy(TmpPinList->InstanceName,InstanceName);
        TmpPinList->Row = YPosition;
        pDoc->BoundingBox(XPosition,YPosition,First,&Box,TmpRouteList);
        TmpPinList = TmpPinList->Link;
        Pin += 1;
        First = 0;
    }
    TmpRouteList->NumOfFeed = CountFeed(PinListHead);
    pDoc->ClearPinList(PinListHead);
    TmpRouteList = TmpRouteList->Link;
}
}
else {
    RowList *RTmpRowHead;
    PlacementList *TmpPlaceList;
    for (char n=1; n<=2; n++) {
        if ( n == 1)
            RTmpRowHead = R1TmpRowHead;
        else
            RTmpRowHead = R2TmpRowHead;
        TmpPlaceList = RTmpRowHead->PlacementList;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while( TmpPlaceList != NULL)  {
    TmpNet = TmpPlaceList->NetworkLink;
    TmpSignalList = TmpNet->SignalListHead;
    while ( TmpSignalList != NULL) {
        SignalName = TmpSignalList->SignalName;
        TmpRouteList = pDoc->RouteListHead;
        while(TmpRouteList != NULL) {
            Name = TmpRouteList->SignalName;
            if (Name == SignalName)
                break;
            TmpRouteList = TmpRouteList->Link;
        }
        if ( TmpRouteList != NULL) {
            First = 1;
            Pin = 0;
            TmpPinList = PinListHead = pDoc->AllocPin
                (TmpRouteList->NumOfPin);
            while(Pin<TmpRouteList->NumOfPin) {
                pDoc->XYPlace(SignalName,
                    &InstanceName,
                    &XPosition,
                    &YPosition,First);
                strcpy(TmpPinList->InstanceName,
                    InstanceName);
                TmpPinList->Row = YPosition;
                pDoc->BoundingBox(XPosition,YPosition,
                    First,&Box,TmpRouteList)
            }
            Pin += 1;
            First = 0;
            TmpPinList = TmpPinList->Link;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    TmpRouteList->NumOfFeed = CountFeed
        (PinListHead);
    pDoc->ClearPinList(PinListHead);
    }
    TmpSignalList = TmpSignalList->Link;
}
    TmpPlaceList = TmpPlaceList->Link;
}
}
}
// Sum Cost
cost = 0.0;
CL = CF = 0;
TmpRouteList = pDoc->RouteListHead;
while(TmpRouteList != NULL) {
    CL += CriticalPath(TmpRouteList); // Cost For Wire Length
    CF += TmpRouteList->NumOfFeed;
    TmpRouteList = TmpRouteList->Link;
}
// Aspect Cost
DispCost(pDC,CL,CF);
newcf = (double)CF;
if (CL==0 )
    CriticalAccept = 1;
else
    CriticalAccept = 0;
cost = (double)(CL + CF);
return cost;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char ClusterView::CountFeed(PinList *PinListHead)
{
    char    NumOfFeed=0;
    PinList *FirstPin,*SecoundPin;
    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();
    FirstPin = PinListHead;
    pDoc->SortYLocation(PinListHead);
    while (FirstPin->Link != NULL) {
        SecoundPin = FirstPin->Link;
        if ( (SecoundPin->Row - FirstPin->Row) >= 2 )
            for ( char i= (int)FirstPin->Row+1; i<= (int)SecoundPin->Row-1; i++)
                NumOfFeed++;
        FirstPin = FirstPin->Link;
    }
    return NumOfFeed;
}

int ClusterView::CriticalPath(RouteList *TmpRouteList)
{
    int    Len;
    CString    Name1,Name2;
    char    found;
    CriticalNetList *Net;
    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();
    found = 0;
    Name1 = TmpRouteList->SignalName;
    Net = pDoc->CriticalNetListHead;
    while (Net != NULL) {
        Name2 = Net->SignalName;

```

```

if (Name1 == Name2) {
    found=1;
    break;
}
else
    Net = Net->Link;
}
if (found)
    if (TmpRouteList->Length > Net->MaximumLength)
        Len = (int)(TmpRouteList->Length - Net->MaximumLength);
    else
        Len = 0;
else
    Len = 0;
return Len;
}

```

```

// floorpln.cpp : implementation file
//

#include "stdafx.h"
#include "std4.h"
#include "std4doc.h"
#include "floorpln.h"
#include "math.h"

//-----
double FloorView::Anealing(CDC* pDC)
{
    FILE *stream;
    int equilibrium;
    double oldcost,newcost,goodcost;
    double T;
    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();

    // writ cost fro create graph
    if ((stream = fopen("COST.$$$", "wb")) == NULL)
    {
        fprintf(stderr, "Cannot open output file.\n");
        return 0;
    }

    T = pDoc->MaxTime;
    equilibrium = pDoc->MaxEqui;
    oldcost = newcost = goodcost = 0;
    if (!T && !equilibrium)
        return 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// Sm Algorithm start

InitState();

goodcost = oldcost = CostFun(pDC);

fwrite(&oldcost, sizeof(oldcost), 1, stream);

GoodState();

GoodRoute();

while (T >= 1.0) {
    // satisfied point

    while (equilibrium != 0) {
        // equilibrium point
        ExchangeState();
        newcost = CostFun(pDC);
        pDoc->ViewRowArea(10, 20, pDC);
        if (newcost < goodcost) {
            goodcost = newcost;
            GoodState();
            GoodRoute();
        }
        fwrite(&newcost, sizeof(newcost), 1, stream);
        pDoc->DispState(pDC ,T,equilibrium,newcost,goodcost);
        if (pDoc->AcceptState(newcost,oldcost,T))
            oldcost=newcost; // accept this state
        else
            OldState(); // not accept new state
        equilibrium--;
    }
    equilibrium = pDoc->MaxEqui;
    //equilibrium = equilibrium-(equilibrium*0.5);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        T=pDoc->UpdateTemp(T);
    }
    pDoc->DispState(pDC ,T,equilibrium,newcost,goodcost);
    fclose(stream); /* close file */
    return goodcost;
}

```

```
double FloorView::CostFun(CDC* pDC)
```

```

{
    double    cost;
    char      Pin,First;
    int       XPosition,YPosition,n;
    unsigned int  CF,CA;
    CString   SignalName,InstanceName,Name;
    RouteList *TmpRouteList;
    PinList   *TmpPinList;
    struct point *p;

    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();

    ClearFeed();
    TmpRouteList = pDoc->RouteListHead;
    while(TmpRouteList != NULL) {
        if ( TmpRouteList->NumOfPin > 1 ) {
            First = 1;
            n = Pin = 0;
            /* Zero out the entire point set. */
            (void) memset(&pts, 0, sizeof (pts));
            p = &(pts.a[0]);
            SignalName = TmpRouteList->SignalName;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(Pin<TmpRouteList->NumOfPin) {
    pDoc->XYPlace
        (SignalName,&InstanceName,&XPosition,&YPosition,First);

    //strcpy(TmpPinList->InstanceName,InstanceName);
    //TmpPinList->XLocation = XPosition;
    //TmpPinList->Row = YPosition;

    Pin += 1;
    First = 0;
    p -> x = XPosition;
    p -> y = (YPosition * pDoc->ChannelHeight) +
        (YPosition*pDoc->CellHeight);
    p -> r = YPosition;
    p -> pnum = ((pnum_t) n);
    p -> other_pnum = ((pnum_t) n);
    ++n;
    ++p;
    //TmpPinList = TmpPinList->Link;
}
//pDoc->ClearPinList(PinListHead);
pts.n = (pnum_t) n;
if ( Pin > 1)
    TmpRouteList->NumOfFeed =
        Do_Minimum_Spanning_Tree(&pts,SignalName,0);
}
TmpRouteList = TmpRouteList->Link;
}

```

```
// Sum Cost
```

```
cost = 0.0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CA = CF = 0;
TmpRouteList = pDoc->RouteListHead;
while(TmpRouteList != NULL) {
    //CL += (int)TmpRouteList->Length; // Cost For Wire Length
    CF += TmpRouteList->NumOfFeed;
    TmpRouteList = TmpRouteList->Link;
}
// Aspect Cost
cost = AspCost();
DispCost(pDC,CF,CA);
return cost;
}

double FloorView::AspCost()
{
    double head,deadspace;
    char first;
    RowList *TmpRowHead;
    FeedList *Feed;
    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();
    first = 1;
    Feed = FeedListHead;
    TmpRowHead = pDoc->RowListHead;
    while (TmpRowHead != NULL) {
        pDoc->RowWidth(TmpRowHead);
        if ( (TmpRowHead->RowNo >= 1) &&
            (TmpRowHead->RowNo < pDoc->NumOfRow-1) ) {
            // Add Feed
            TmpRowHead->RowWidth += Feed->FeedCount*pDoc->FeedSize;
            Feed = Feed->Link;

```

```

    }
    if (first) {
        first = 0;
        MaxWidth = TmpRowHead->RowWidth;
    }
    else
        if (TmpRowHead->RowWidth > MaxWidth)MaxWidth = (double)
            TmpRowHead->RowWidth;
        TmpRowHead = TmpRowHead->Link;
    }
    head = deadspace =0;
    TmpRowHead = pDoc->RowListHead;
    while (TmpRowHead != NULL) {
        if ( TmpRowHead->RowNo == 0 ||
            TmpRowHead->RowNo == pDoc->NumOfRow-1 )
            head += abs(TmpRowHead->RowWidth - MaxW);
        else
            deadspace += abs(TmpRowHead->RowWidth - MaxW);
        //deadspace += abs(TmpRowHead->RowWidth - 648);
        //deadspace += abs(TmpRowHead->RowWidth - maxwidth);
        //deadspace += abs(TmpRowHead->RowWidth - pDoc->EstimateCoreWidth);
        TmpRowHead = TmpRowHead->Link;
    }
    //return ( head*head + deadspace );
    //return ( maxwidth + deadspace );
    //return ( deadspace );
    //return ( MaxWidth );
    return ( sqrt(MaxWidth)*sqrt(MaxWidth) + sqrt(head + deadspace) );
    //return ( head + deadspace );
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// placevie.cpp : implementation file
//

#include "stdafx.h"
#include "std4.h"
#include "std4doc.h"
#include "placevie.h"
#include "math.h"

////////////////////////////////////
// PlaceView message handlers
// -----

double PlaceView::Anealing(CDC* pDC)
{
    FILE *stream;
    int equilibrium;
    double oldcost,newcost,goodcost;
    double T;
    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();

    // writ cost fro create graph
    if ((stream = fopen("COST.$$$", "wb")) == NULL)
    {
        fprintf(stderr, "Cannot open output file.\n");
        return 0;
    }

    T = pDoc->MaxTime;
    equilibrium = pDoc->MaxEqui;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

oldcost = newcost = goodcost = 0;
if ( !T && !equilibrium)
    return 0;

// Sm Algorithm start
InitState();

goodcost = oldcost = CostFun(pDC);
fwrite(&oldcost, sizeof(oldcost), 1, stream);
GoodState();

while (T >= 1.0) {
    // satisfied point
    while (equilibrium != 0) {
        // equilibrium point
        ExchangeState();
        newcost = CostFun(pDC);
        pDoc->ViewRowArea(10, 20, pDC);
        if (newcost < goodcost) {
            goodcost = newcost;
            GoodState();
        }
        fwrite(&newcost, sizeof(newcost), 1, stream);
        pDoc->DispState(pDC ,T,equilibrium,newcost,goodcost);
        if (pDoc->AcceptState(newcost,oldcost,T))
            oldcost=newcost; // accept this state
        else
            OldState(); // not accept new state
        equilibrium--;
    }
    equilibrium = pDoc->MaxEqui;
    //equilibrium = equilibrium-(equilibrium*0.5);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        T=pDoc->UpdateTemp(T);
    }
    pDoc->DispState(pDC ,T,equilibrium,newcost,goodcost);

    fclose(stream); /* close file */
    return goodcost;
}

```

```

double PlaceView::CostFun(CDC* pDC)
{
    double    cost;
    char      Pin,First;
    int       XPosition,YPosition;
    long int  CL,CRT;

    RECT      Box;
    CString   SignalName,InstanceName,Name;
    NetworkList *TmpNet;
    SignalList *TmpSignalList;
    RouteList *TmpRouteList;

    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();

    TmpRouteList = pDoc->RouteListHead;
    if (InitCost) {
        InitCost = 0; // Off
        while(TmpRouteList != NULL) {
            First = 1;
            Pin = 0;
            SignalName = TmpRouteList->SignalName;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while(Pin<TmpRouteList->NumOfPin) {
    pDoc->YPlace(SignalName,&InstanceName,
                &XPosition,&YPosition,First);
    pDoc->BoundingBox(XPosition,YPosition,First,
                    &Box,TmpRouteList);

    Pin += 1;
    First = 0;
}
TmpRouteList = TmpRouteList->Link;
}
}
else {
    RowList *RTmpRowHead;
    PlacementList *TmpPlaceList;

    RTmpRowHead = R1TmpRowHead;
    TmpPlaceList = RTmpRowHead->PlacementList;
    while( TmpPlaceList != NULL) {
        TmpNet = TmpPlaceList->NetworkLink;
        TmpSignalList = TmpNet->SignalListHead;
        while ( TmpSignalList != NULL) {
            SignalName = TmpSignalList->SignalName;
            TmpRouteList = pDoc->RouteListHead;
            while(TmpRouteList != NULL) {
                Name = TmpRouteList->SignalName;
                if (Name == SignalName)
                    break;
                TmpRouteList = TmpRouteList->Link;
            }

```

```

if ( TmpRouteList != NULL) {
    First = 1;
    Pin = 0;
    while(Pin<TmpRouteList->NumOfPin) {
        pDoc-> XYPlace(SignalName,
                        &InstanceName,
                        &Xposition,
                        &YPosition,First);
        pDoc->BoundingBox(XPosition,YPosition,
                           First, &Box,TmpRouteList);
        Pin += 1;
        First = 0;
    }
    TmpSignalList = TmpSignalList->Link;
}
TmpPlaceList = TmpPlaceList->Link;
}

// Sum Cost
cost = 0.0;
CRT= CL = 0;
TmpRouteList = pDoc->RouteListHead;
while(TmpRouteList != NULL) {
    CRT += CriticalPath(TmpRouteList);    // Cost For Wire Length
    CL += (int)TmpRouteList->Length;      // Cost For Wire Length
    TmpRouteList = TmpRouteList->Link;
}

DispCost(pDC,CL,CRT);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (CRT==0 )
    CriticalAccept = 1;
else
    CriticalAccept = 0;
cost = (double)(sqrt((double)CRT)*sqrt((double)CRT)+sqrt((double)CL));
return cost;
}
```



```

// globalvi.cpp : implementation file
//

#include "stdafx.h"
#include "std4.h"
#include "std4doc.h"
#include "globalvi.h"
#include "steiner.h"

////////////////////////////////////
// GlobalView drawing

void GlobalView::PinCluster()
{
    CString      SignalName,InstanceName;
    char         First,Pin;
    int          XPosition,YPosition;
    RouteList    *TmpRouteList;
    PinList      *TmpPinList;
    int          n,mst,mrst;
    struct point *pmst,*prst,*pt;

    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();

    TmpRouteList = pDoc->RouteListHead;
    while(TmpRouteList != NULL) {
        if ( TmpRouteList->NumOfPin > 1 ) {
            First = 1;
            n = Pin = 0;

            /* Zero out the entire point set. */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(void) memset(&ptsmst, 0, sizeof (ptsmst)); // for mst
pmst = &(ptsmst.a[0]);
(void) memset(&ptrst, 0, sizeof (ptrst)); // for rst
prst = &(ptrst.a[0]);
(void) memset(&ptrst, 0, sizeof (ptrst)); // for temp rst
pt = &(ptrst.a[0]);
SignalName = TmpRouteList->SignalName;
while(Pin<TmpRouteList->NumOfPin) {
    pDoc->XYPlace(SignalName,&InstanceName,
                  &XPosition,&YPosition,First);
    Pin += 1;
    First = 0;
    pmst->x= prst->x = pt->x = XPosition;
    pmst->y= (YPosition * pDoc->ChannelHeight)
            +(YPosition*pDoc->CellHeight);
    pmst->r = prst->r = pt->r = YPosition;
    pmst->pnum = prst->pnum = pt->pnum =
                ((pnum_t) n);
    pmst->other_pnum = pt->other_pnum =
                ((pnum_t) n);
    ++n;
    ++pmst; ++prst; ++pt;
}
ptsmst.n = ptrst.n = prst.n = (pnum_t) n;
if ( Pin >= 2 ) {
    if ( CriticalNet(SignalName) ) {
        mrst = Do_Kahng_Robins_Len(&ptrst);
        if ( RstInsertFeed(&ptrst,&ptrst,SignalName)) {
            mst = Do_Minimum_Spanning_Tree_Len
                    (&ptsmst);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        MstInsertFeed(&ptsmst,SignalName);
    }
}
else {
    Do_Minimum_Spanning_Tree_Len(&ptsmst);
    MstInsertFeed(&ptsmst,SignalName);
}
}
}
}
    TmpRouteList = TmpRouteList->Link;
}
pDoc->GoodState("GLOBAL.$$$");
}
int GlobalView::Do_Minimum_Spanning_Tree_Len(struct pset *pts)
{
    int          i,n,mst_len;
    struct edge  *ep;
    struct point *p1,*p2;
    //struct edge  m[MAX_TERMINALS - 1];

    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();

    Nmst = n = mst(pts, &m[0]);
    mst_len = 0;
    for (i = 0; i < n; i++) {
        ep = &m[i];
        p1 = &(pts -> a [ep -> p1]);
        p2 = &(pts -> a [ep -> p2]);
        mst_len += ep -> len;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    return (mst_len);
}

// * This routine computes an approximate Steiner Minimal Tree using
// * the Kahng-Robins heuristic, and plots the result.
int GlobalView::Do_Kahng_Robins_Len( struct pset *pts)
{
    int          i,n,save_n;
    struct edge  *ep;
    struct point *p1,*p2;
    int          kr_len;
    struct edge  m[MAX_POINTS];
    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();

    save_n = pts -> n;
    Nrst = n = kahng_robins (pts, 0, &m [0]);
    kr_len = 0;
    for (i = 0; i < n; i++) {
        ep = &m [i];
        p1 = &(pts -> a [ep -> p1]);
        p2 = &(pts -> a [ep -> p2]);
        kr_len += ep -> len;
    }
    return (kr_len);
}

```

```

void GlobalView::SwitchSegment(CString SignalName,CString InstName1,CString
InstName2)
{
    FILE *stream;
    SegmentList      NewSegment;

    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();

    if ((stream = fopen("SWSEGM.$$$", "ab")) == NULL) /* open file Initial
Placement */
    {
        fprintf(stderr, "Cannot open output file.\n");
        return;
    }

    strcpy(NewSegment.InstName1,InstName1);
    strcpy(NewSegment.InstName2,InstName2);
    strcpy(NewSegment.SignalName,SignalName);
    if (ChannelNo == pDoc->NumOfRow)
        ChannelNo -= 1;
        NewSegment.ChannelNo = NewSegment.AssignTo = ChannelNo;
    NewSegment.SwOrFx = 'S' ;
    NewSegment.X1 = NewSegment.X2 = 0;
    NewSegment.SegmentNo = N;

    fwrite(&NewSegment, sizeof(SegmentList), 1, stream);

    N++;
    fclose(stream); /* close file */
}

```

```

// trckview.cpp : implementation file
//

#include "stdafx.h"
#include "std4.h"
#include "trckview.h"
#include "std4doc.h"

double TrckView::Anealing(CDC* pDC)
{
    FILE *stream;
    int equilibrium;
    double oldcost,newcost,goodcost;
    double T;
    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();

    // writ cost fro create graph
    if ( (stream = fopen("COST.$$$", "wb")) == NULL )
    {
        fprintf(stderr, "Cannot open output file.\n");
        return 0;
    }

    T = pDoc->MaxTime;
    equilibrium = pDoc->MaxEqui;
    oldcost = newcost = goodcost = 0;
    if ( !T && !equilibrium)
        return 0;

    // Sm Algorithm start
    InitState();
    goodcost = oldcost = CostFun(pDC);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fwrite(&oldcost, sizeof(oldcost), 1, stream);
GoodState(); // Skip
while (T >= 1.0) {
    // satisfied point
    while (equilibrium != 0) {
        // equilibrium point
        ExchangeState();
        newcost = CostFun(pDC);
        //pDoc->ViewRowArea(10, 20, pDC);
        if (newcost < goodcost) {
            goodcost = newcost;
            GoodState();
        }
        fwrite(&newcost, sizeof(newcost), 1, stream);
        pDoc->DispState(pDC ,T,equilibrium,newcost,goodcost);
        if (pDoc->AcceptState(newcost,oldcost,T))
            oldcost=newcost; // accept this state
        else
            OldState(); // not accept new state
        equilibrium--;
    }
    equilibrium = pDoc->MaxEqui;
    //equilibrium = equilibrium-(equilibrium*0.5);
    T=pDoc->UpdateTemp(T);
}
pDoc->DispState(pDC ,T,equilibrium,newcost,goodcost);
fclose(stream); /* close file */
return goodcost;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

double TrckView::CostFun(CDC* pDC)
{
    double cost=0;
    SegList *TmpSeg,*FxSeg;

    CStd4Doc* pDoc = (CStd4Doc *)GetDocument();

    GetState();
    TmpSeg = SegHead;
    FxSeg = FixedSegHead;
    while( TmpSeg != NULL) {
        cost += (double)abs( (TmpSeg->SegNo+FxSeg->SegNo) -DBAR );
        TmpSeg = TmpSeg->Link;
        FxSeg = FxSeg->Link;
    }
    return cost;
}

```

```

// gredview.cpp : implementation file
//

#include "stdafx.h"
#include "std4.h"
#include "gredview.h"
#include "std4doc.h"
#include "scmos.h"

////////////////////////////////////
void GredView::GreedyRoute(char ChannelNo,CDC *pDC )
{
    int i = 1;
    ClearTrack();
    TmpTBarLine = TBarLineHead;
    TmpBBarLine = BBarLineHead;
    while ( TmpTBarLine != NULL) {
        // Clear VT & VB
        VT.X1 = VT.X2 = 0;
        strcpy(VT.SignalName,"NULL");
        VTEmpy.X1 = VTEmpy.X2 = 0;
        strcpy(VTEmpy.SignalName,"NULL");
        VB.X1 = VB.X2 = 0;
        strcpy(VB.SignalName,"NULL");
        VBEmpty.X1 = VBEmpty.X2 = 0;
        strcpy(VBEmpty.SignalName,"NULL");
        // Step By Step
        GreedyStep1( pDC );
        GreedyStep2( pDC );
        GreedyStep3( pDC );
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GreedyStep4( pDC );
GreedyStep6( pDC );
FreeCollap();
TmpTBarLine = TmpTBarLine->Link;
TmpBBarLine = TmpBBarLine->Link;
}
}

void GredView::AddDoglegs(CollapList *TmpCollapList,TrackList *Track,CString
TOrB)
{
char      TrackNo,Add;
int       Start,End;
VerList   Doglegs,Ove rLap;
TrackList *TmpTrack;
Add = 0;
Start = ROW;
End = Start + ( (InitialChannelWidth + 1 ) * (int)VSpaceLayer1 );

if ( TOrB == "TOP" )
    TmpTrack = TrackHead;

else
    TmpTrack = TrackLast;

TrackNo = 0;
while( TmpTrack != NULL ) {
    if ( !strcmp(TmpTrack->SignalName, "NULL") ) {
        TrackNo = TmpTrack->ChannelNo;
        if ( TOrB == "TOP" ) {
            Doglegs.X1 = Start + (TrackNo * (int)VSpaceLayer1 );
            Doglegs.X2 = End;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    else {
        Doglegs.X1 = Start;
        Doglegs.X2 = Start + (TrackNo * (int)VSpaceLayer1 );
    }
    OverLapInVerList(&OverLap,Doglegs.X1,Doglegs.X2);
    if ( !strcmp(OverLap.SignalName,"NULL") ) {
        Add = 1;
        break;
    }
    else if ( !strcmp(OverLap.SignalName,TmpCollapList->
        SignalName) ){
        if ( !OverLapOther(TmpCollapList->
            SignalName,TmpCollapList->
            X1,TmpCollapList->X2) ) {
            UpdateInVerList(TmpCollapList->
                SignalName,Doglegs.X1,Doglegs.X2);
            strcpy(TmpTrack->SignalName,TmpCollapList->
                SignalName);
            break;
        }
    }
}
}
if ( TOrB == "TOP" )
    TmpTrack = TmpTrack->Link;
else
    TmpTrack = TmpTrack->Prev;
}
if ( (Doglegs.X2 -Doglegs.X1) > (unsigned int)MinimumJogLength
    && TmpTrack != NULL && Add) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
AddToVerList(Doglegs.SignalName,Doglegs.X1,Doglegs.X2);  
strcpy(TmpCollapList->SignalName,"NULL");  
strcpy( Track->SignalName,"NULL");  
}  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ค. MCNC Benchmark

### รูปแบบ Yal Description

Copyright (C) 1987 by Bryan Preas and Ken Roberts.

All rights reserved.

Date: March 11, 1987

Modified: To accomodate rectilinear cells, power and ground nets with current and voltage restrictions, and the specification of critical net lengths.

The new YAL is not compatible with the old.

Jonathan Rose, November 18, 1987

#### I. General Comments

Comments are delimited by `"/* "` and `" */"`. Comments MAY extend across physical line boundaries.

White space is the delimiter between tokens. White space is one or more of any combination of space, tab, line feed, or carriage return.

The data is free format. A semicolon, `;`, is the logical line terminator. Any number (0 or more) of white space characters may separate a semicolon. A logical line can occupy more than one physical line (characters separated by carriage returns). Any number of logical lines can occupy a physical line.

"name"s are used for modules (cell definitions), cell instances

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(instantiations of modules), and signals (or nets). "name"s may contain any character except white space and semicolon. "name"s need not start with a letter. (e.g., "2" is a valid name). Case is significant; the name "A1" is different from the name "a1".

In the following descriptions, "|" means OR: choose one item from the list separated by the "|". Items between square brackets, "[" and "]", indicate optional fields on a logical line. Ellipsis, ". . .", (either horizontal or vertical) indicate that the preceding field or line should be repeated as necessary. Words between angle brackets, "<" and ">", are used as intermediate (non-terminal) definitions in the language. WORDS in upper case are keywords of the language.

"number"s are reals and are specified in microns.

Design rules are NOT specified in this language. Look in the text of circuit descriptions or cell libraries for design rules.

## II. Module Definition

A module is a definition of a circuit being laid out or a constituent (or primitive) cell. The definition for each circuit or primitive cell begins with "MODULE <modulename>". The definition ends with "ENDMODULE". A template of a module follows.

```
MODULE <modulename>;
    TYPE <moduletype>;
    DIMENSIONS <X1> <Y1> <X2> <Y2> .... <XN> <YN>
    IOLIST;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<signalname> <terminaltype> [<Xposition> | <side>
[ <Yposition> | <position> [<width> <layer>]]]
[CURRENT <current>] [VOLTAGE <voltage>];

```

```

.
.
.
ENDIOLIST;

```

NETWORK

```

<instancename> <modulename> <signalname> . . . ;

```

```

.
.
.
ENDNETWORK;

```

PLACEMENT

```

<instancename> <xlocation> <ylocation>
[<reflection>] [<rotation>];

```

```

.
.
.
ENDPLACEMENT;

```

CRITICALNETS;

```

<signalname> <maximumlength>;

```

```

.
.
.
ENDCRITICALNETS;

```

ENDMODULE;

**<modulename> ::= name**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<moduletype> ::= STANDARD | PAD | GENERAL | PARENT | FEEDTHROUGH

<width> ::= number

<height> ::= number

<signalname> ::= name

<terminaltype> ::= I | O | B | PI | PO | PB | F | PWR | GND

<side> ::= BOTTOM | RIGHT | TOP | LEFT

<layer> ::= PDIFF | NDIFF | POLY | METAL1 | METAL2

<xposition> ::= number

<yposition> ::= number

<position> ::= number

<instancename> ::= name

<xlocation> ::= number

<ylocation> ::= number

<Xi> ::= number

<Yi> ::= number

<current> ::= number

<voltage> ::= number

<maximumlength> ::= number

<reflection> ::= RFLNONE | RFLY

<rotation> ::= ROT0 | ROT90 | ROT180 | ROT270

The TYPE line is required for all modules. The allowable types are:

**STANDARD:** an internal, primitive cell that must be placed on a row or column.

**PAD:** a driver (pad) cell that should be placed around the periphery of the chip.

**GENERAL:** an internal, primitive general cell or building block.

**PARENT:** a higher level module to be laid out.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**FEEDTHROUGH:** a cell used to connect a signal across a row.

Whether or not a line (e.g., DIMENSIONS) or a section (NETWORK or PLACEMENT) within a module is required depends on the TYPE of module being defined. A TYPE and an IOLIST are always required.

A primitive cell (one to be used as a constituent of a layout:

TYPE = STANDARD, PAD, GENERAL, or FEEDTHROUGH) is assumed to have been laid out previously and therefore has the following data:

DIMENSIONS and IOLIST; <xposition> or <side>, <yposition> or <position>, <width> and <layer> are required for each of the IO <signalname>s within the IOLIST for primitive cells. NETWORK and PLACEMENT sections are ignored for primitive cells.

A feedthrough cell (TYPE = FEEDTHROUGH) should not appear in a NETWORK but is inserted into a layout by the layout system as necessary.

A module that defines a circuit to be laid out (TYPE = PARENT) must have an IOLIST and a NETWORK section. DIMENSIONS are ignored.

The DIMENSIONS line gives the co-ordinates of the the corners of the rectilinear box, in counter-clockwise order.

The IOLIST section defines the external connections or terminals of the module; this section should be included in definitions of all modules. Each line in this section defines a segment of connection to a higher level of hierarchy for the module. Each line must contain at least a <signalname> of an IO terminal and its <terminaltype>. Within a module being laid out (TYPE = PARENT) a

<signalname> on a terminal line in the IOLIST is in the same name space as the <signalname>s in the NETWORK of that same <module-name>. Each module has its own separate name space. Other fields may be required depending on the terminal and the module that contains it as described below.

If a primitive module (TYPE = STANDARD, PAD, GENERAL, or FEEDTHROUGH) is being defined, the lines within an IOLIST must contain the <xposition>, <yposition>, <width>, and <layer>. Terminals appear in the order in which they will be referenced in the NETWORK section of any higher level module. Electrically equivalent terminals (e.g., in a dual-ported primitive module) must have the same <signalname>s. (In this case binding between the terminal <signalname>s of a primitive module and the <signalname>s of the higher level module are performed in the order of unique signal names. Refer to the example in Section 3.)

It should be assumed that electrically equivalent terminals can be used as "feedthroughs" for the signals connected to them. If additional over-cell routing is allowed, i.e., for signals not connected to the cell, terminal locations for these feedthroughs will appear at the end of the IO terminal list. Valid <terminaltype>s are

"I" for input

"O" for output

"B" for bidirectional

"PI", "PO", "PB" for a pad terminal on a primitive,

where I, O, B are input, output, and

bidirectional, as above

"F" for an independent feedthrough location

"PWR" for VDD power input

"GND" for VSS ground input

Terminal locations are specified by <xposition> or <side>, <yposition> or <position>, <width> and <layer>, or LEFT. <xposition> and <yposition> give the coordinate of the center of the terminal in microns on the edge of the cell. To be compatible with the old YAL, terminals can also be specified as <side> and position. <side> may be BOTTOM, RIGHT, TOP or LEFT. <position> gives the coordinate of the center of the terminal in microns along the indicated <side> from the left or bottom of the module as appropriate. <width> specifies the dimension of the terminal along the indicated edge; it extends from <xposition> - <width>/2 to <xposition> + <width>/2 or <yposition> - <width>/2 to <yposition> + <width>/2 as appropriate. <layer> specifies the conductor layer of the terminal.

Fixed pad placement for higher level circuits (TYPE = PARENT) requires a slightly different interpretation of the fields and is specified in the following manner. In this case, the <side> and <position> specifiers must be used, rather than the <xposition> <yposition>. If the terminal is bound to the pad of an IO driver, then the <side> refers to the side of the chip and <position> refers to the position of the origin of that driver cell. <width> and <layer> are ignored. If no <position> is specified, then the pad is restricted to the indicated <side> but not to a <position> on the <side>. If no <side> is specified for a terminal bound to an IO driver then that IO driver may be placed on any <side>.

The NETWORK section defines the internal connectivity for the module. If a NETWORK section is included in the definition of a

primitive module (TYPE = STANDARD, PAD, GENERAL, or FEEDTHROUGH), it will be ignored. Each line defines an instance of a primitive module that is included in the circuit. Each instance definition has the following fields:

<instancename> is the name of the instance of a module. An instance has signal bindings (specified in this section) and a location and orientation to be determined by the layout system (specified in the PLACEMENT section).

<modulename> is the name of the module to which this instance is bound.

<signalname> . . . is a list of <signalname>s connected to the module, in the order determined by the module definition. For example, the nth <signalname> is bound to the nth UNIQUE terminal <signalname> of the primitive module definition. Connections to the feedthrough terminals are not included in this list; they should be determined by the layout system. Single terminal <signalname>s are used to specify an unconnected terminal. An unconnected terminal may also result if the list of <signalname>s on an instance definition is shorter than the list of unique <signalname>s in the IOLIST of the primitive module being bound.

The PLACEMENT section is used to report placement results, one logical line per instance. If a PLACEMENT section is included in the definition of a primitive module (TYPE = STANDARD, PAD, GENERAL, or FEEDTHROUGH), it will be ignored.

<instancename> is the cell instance, and must match an <instan-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

cename> in the NETWORK section. <xlocation> and <ylocation> specify the location of the (0, 0) point (in microns) of the module after any orientation change of the instance. <reflection> and <rotation> are optional; the default for <reflection> is RFLNONE (the normal or "defined" reflection); RFLY means reflected about the Y axis. The default for <rotation> is ROT0 (no rotation); ROT90, ROT180, ROT270 indicate a counterclockwise rotation for the appropriate number of degrees. Reflection is applied before rotation.

The CRITICALNETS section is used to specify which nets are performance-critical, and their maximum length. <signalname> is the critical net, and must match a <signalname> in the NETWORK section. <maximumlength> is a number in microns specifying how long the signal wire can be restricted to. Any placement program interested in performance optimization tries to meet this limit.

### III. Example 1: A Simple, Primitive Cell

This is a simple standard cell inverter. It has 1 input and 1 output. It is constructed on a 10-microns routing grid.

```

/*                                     */
/* Module name = INV                  */
/* Width = 20 microns, Height = 160 microns */
/*                                     */
MODULE INV;
TYPE STANDARD;
DIMENSIONS 0 0 20 0 20 160 0 160;
/*                                     */

```

```

/* Define the external connections: I1 and O1          */
/*                                                     */
IOLIST;
/* <signalname><terminaltype><xposition> <yposition><width><layer> */
I1      I           5.0      160.0    3.0  METAL2;
I1      I           5.0      0.0      3.0  METAL2;
O1      O          15.0      160.0    3.0  METAL2;
O1      O          15.0      0.0      3.0  METAL2;
ENDIOLIST;
ENDMODULE;

```



3 wide    3 wide

at 5.0    at 15.0

on METAL2 on METAL2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This MODULE would be referenced in a NETLIST section as follows:

```
/* <instancename> <modulename><signalname><signalname> */
CELLX      INV      SIGIN      SIGOUT;
```

Terminal I1 on <modulename> INV is bound to <signalname> SIGIN  
and terminal O1 is bound to <signalname> SIGOUT.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## วงจร SCO ที่อธิบายด้วย Yal Description

```

MODULE s109;
  TYPE PARENT;
  DIMENSIONS 0.002 0 0.002 0.002 0 0.002 0 0;
  IOLIST;
    reset I TOP;
    Phi1H I TOP;
    Phi2H I TOP;
    Clear I TOP;
    CD16 I TOP;
    CD15 I TOP;
    W O BOTTOM;
    Z O BOTTOM;
  ENDIOLIST;
NETWORK;
  INS620 i1s I74 II6;
  INS621 oi2s I35 I88 W;
  INS622 oai22s I21 I74 YD2 I72 II5;
  INS623 aoi32s I56 I291 YD3 I22 I307 I39;
  INS624 aoi33s I23 I307 YD3 I82 I291 YD4 I40;
  :
  IIFS8 dr2s II158 reset Phi1H Phi2H II162 YD5;
  IIFS9 dr2s II307 reset Phi1H Phi2H II311 YD12;
ENDNETWORK;
CRITICALNETS;
  I74 150;
  I28 200;
ENDCRITICALNETS;
ENDMODULE;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

ชื่อผู้เขียน	นายทรงพล ใหม่สาลี
วันเดือนปีเกิด	วันที่ 27 เมษายน พ.ศ. 2507
สถานที่เกิด	จังหวัดปราจีนบุรี
วุฒิการศึกษาระดับปริญญาตรี	วิทยาศาสตร์บัณฑิต สาขาวิชาสถิติ
สถานที่สำเร็จการศึกษา	มหาวิทยาลัยรามคำแหง
ปีสำเร็จการศึกษา	ปีการศึกษา 2529
สถานที่ทำงานปัจจุบัน	นักวิชาการคอมพิวเตอร์ 6ว กองวิชาการประมวลผลข้อมูล สำนักงานสถิติแห่งชาติ

