ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อ การออกแบบและสร้างวงจร ASICs ของตัวเข้ารหัสและถอดรหัสเสียงเวกเตอร์ควอนไตซ์

An ASICs Design and Implementation of a Vector Quantization Speech Encoder and Decoder

ชื่อนักศึกษา 1. นายบัณฑิตศักดิ์ ศรีโสภา รหัสประจำตัว 42035339

2. นายวิทยา บุตรโสภา รหัสประจำตัว 42035350

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์กิติพงค์ มะโน

อาจารย์ที่ปรึกษาร่วม อาจารย์พีระวุฒิ สุวรรณจันทร์

คณะกรรมการสอบปริญญานิพนธ์		ลายมือชื่อ
1. อาจารย์กิติพงค์	มะโน	.....
2. อาจารย์พีระวุฒิ	สุวรรณจันทร์	.....
3. อาจารย์สุชิน	อาจหาญ	.....
4. อาจารย์อำพล	ทองระอา	.....
5. อาจารย์สุระชัย	พิมพ์สาลี	.....

วัน/เดือน/ปีที่สอบ วันเสาร์ที่ 9 ธันวาคม พ.ศ. 2543 เวลา 09.15 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.วิสุทธิ์ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 9 เดือน ธ.ค. พ.ศ. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปริญญานิพนธ์

การออกแบบและสร้างวงจร ASICs ของตัวเข้ารหัส และถอดรหัสเสียงเวกเตอร์ควอนไตซ์

AN ASICs DESIGN AND IMPLEMENTATION OF A VECTOR QUANTIZATION SPEECH ENCODER AND DECODER

T040180

นายบัณฑิตศักดิ์ ศรีโสภา
นายวิทยา บุตรโสภา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

ปก.
LB2611
.T43
2543

เลขหมู่.....
เลขทะเบียน 40180
วัน, เดือน, ปี 17 ส.ค. 2544

b 11092518
i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง การออกแบบและสร้างวงจร ASICs ของตัวเข้ารหัสและถอดรหัสเสียง
เวกเตอร์ควอนไตซ์
An ASICs design and implementation of a vector quantization
Speech encoder and decoder

วัตถุประสงค์

1. เพื่อศึกษาหลักการออกแบบวงจรรวมเฉพาะกิจ
2. เพื่อศึกษาเครื่องมือที่จะสามารถใช้ในการออกแบบวงจรรวมเฉพาะกิจ
3. เพื่อออกแบบและสร้าง Layout ของวงจรรวม
4. เพื่อที่สร้างตัว Layout วงจรรวมของตัวเข้ารหัสและถอดรหัสข้อมูลเสียงที่ใช้วิธี ควอนไตซ์เวคเตอร์

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ศึกษาและเข้าใจถึงการออกแบบวงจรรวมเฉพาะกิจ
2. ได้เข้าใจถึงหลักการออกแบบสถาปัตยกรรมและสร้างฮาร์ดแวร์ของตัวประมวลผลสำหรับค้นหาข้อมูลใน โคดบุ๊ค
3. ได้ศึกษาและเข้าใจถึงการออกแบบตัวเข้ารหัสและถอดรหัสข้อมูลเสียงที่ใช้ วิธี ควอนไตซ์เวคเตอร์
4. เข้าใจถึงการใช้เครื่องมือ L-Edit ของบบริษัท Tanner Tool ที่ใช้ในการออกแบบ
5. ได้ Layout จากการออกแบบวงจรตัวเข้ารหัสและถอดรหัสข้อมูลเสียงที่ใช้วิธีควอนไตซ์-เวคเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	การออกแบบและสร้างวงจร ASICs ของตัวเข้ารหัสและถอดรหัสเสียงเวกเตอร์ควอนไตซ์
นักศึกษา	นายบัณฑิตศักดิ์ ศรีโสภา
	นายวิทยา บุตรโสภา
อาจารย์ที่ปรึกษา	อาจารย์กิติพงค์ มะโน
อาจารย์ที่ปรึกษาร่วม	อาจารย์พีระวุฒิ สุวรรณจันทร์
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชา	อิเล็กทรอนิกส์และคอมพิวเตอร์
ปีการศึกษา	2543

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เสนอการออกแบบและสร้างวงจรรวมเฉพาะงานของตัวเข้ารหัสเสียงโดยใช้วิธีเวกเตอร์ควอนไตซ์มาตราฐานโดยใช้เทคโนโลยี ASIC (Application Specification Integrated Circuits) ของบริษัทอาคาเทลไมโครอิเล็กทรอนิกส์โดยใช้เซลมาตราฐานชื่อ MTC05 ในขบวนการออกแบบและสร้างได้ออกแบบโดยใช้ภาษา VHDL และพัฒนาเป็นเลย์เอาต์ของชิพเดี่ยวโดยใช้โปรแกรม Tanner Tools Pro Leonardo และ Xilinx Foundation program. โดยชิปมีขนาด 2.9 x 2.4 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	An ASIC design and Implementation of A vector quantization speech encoder and decoder
Students	Mr. Bandidsak Srisopa
	Mr. Vittaya Budsopa
Advisor	Mr. Kitipong Mano
Co-Advisor	Mr. Peerawut Suwanjan
Education Level	Bachelor of Science in Industrial Education
Program in	Electronics and Computer
Academic Year	2000

ABSTRACT

This thesis presents a semi-custom ASIC (Application Specification Integrated Circuits) design and Implementation of a vector quantization speech encoder and decoder. The encoder and decoder implemented in this paper meets the requirements of vector quantization standard, and is used 0.5 um MTC05 standard cell of Alcatel microelectronic cooperation. The encoder and decoder system has been design in VHDL (VHSIC Hardware Description Language) and developed as a single chip by Tanner Tools Pro, Leonardo and Xilinx Foundation program. The Chip size is 2.9 x 2.4 cm.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี เนื่องจากความกรุณาของอาจารย์ที่ปรึกษาคืออาจารย์กิติพงค์ มะโน ที่คอยให้แนะนำและความช่วยเหลือต่างๆ ทั้งให้คำปรึกษาและคำแนะนำต่างๆ ในทุกๆเรื่องและขอขอบพระคุณอาจารย์ในภาควิชาที่ให้ความช่วยเหลือตลอดในการทำโครงงานเต็มที่ ทั้งด้านเวลา สถานที่ เครื่องมืออุปกรณ์ต่างๆ จึงขอกราบขอบพระคุณไว้ ณ ที่นี้

ขอขอบพระคุณ NECTEC ที่ให้ความอนุเคราะห์ในเรื่องการตอบปัญหาต่างๆและข้อมูลที่ใช้ ในการทำโครงงานและขอขอบพระคุณ คุณพ่อ คุณแม่ ผู้ให้กำเนิดและให้โอกาสในการศึกษาตลอดจนเพื่อนๆ และผู้ที่เกี่ยวข้องทุกท่านที่ให้คำแนะนำและคอยเป็นกำลังใจในการทำงาน จนทำให้โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ขีดความสามารถของโครงงาน	2
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 กล่าวนำ	4
2.2 การควอนไตซ์เวกเตอร์แบบมาตรฐาน	4
2.2.1 หลักการควอนไตซ์เวกเตอร์แบบมาตรฐาน	4
2.2.2 การคำนวณค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน	6
2.2.3 การออกแบบโค้ดบุ๊ค	8
2.3 Top – Down Design	11
2.4 หลักการออกแบบวงจรรวม	15
2.4.1 วงจรรวมชนิดต่างๆ	16
2.4.2 วงจรรวมประเภท Full Custom	17
2.4.3 วงจรรวมประเภท Semi Custom	18
2.4.4 Design Flow	20
2.5 ภาษา VHDL	20
2.5.1 โครงสร้างของภาษา VHDL	24
2.5.2 ลักษณะของการเขียนบรรยายด้วยภาษา VHDL	25
2.6 เครื่องมือที่ใช้ในการออกแบบวงจรรวม	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 3 การออกแบบ การสร้างและการทำงาน	28
3.1 การใช้ Tanner Tool	28
3.2 ส่วนประกอบต่างๆ ของตัวเข้ารหัสและถอดรหัสข้อมูลเสียง	30
3.2.1 ส่วนประกอบภาค DATA_BUFFER	31
3.2.2 ส่วนประกอบภาค CB_BUFFER	32
3.2.3 ส่วนประกอบภาค SUB_ABS	33
3.2.4 ส่วนประกอบภาค ADDER	34
3.2.5 ส่วนประกอบภาค COMPARATOR	35
3.2.6 ส่วนประกอบภาค ADDRESS & INDEX	36
3.2.7 ส่วนประกอบภาค CONTROL	37
3.2.8 ส่วนประกอบภาค SERIAL_OUT	39
3.3 การสร้าง PADFRAME	40
3.3.1 ขั้นตอนการสร้าง PADFRAME	40
3.3.2 การตั้งชื่อ Label ของ PAD	40
3.4 การทำ Block Place & Route (BPR)	41
บทที่ 4 การทดลองและผลการทดลอง	48
4.1 กล่าวนำ	48
4.2 การจำลองการทำงานของวงจรรวมเฉพาะกิจตัวเข้ารหัสและถอดรหัสข้อมูลเสียง	48
4.2.1 ผลการจำลองการทำของสัญญาณของภาค DATA_BUFFER	48
4.2.2 ผลการจำลองการทำงานสัญญาณของภาค COMPARATOR	50
4.2.3 ผลการจำลองการสัญญาณของภาค SERIAL_OUT	51
4.2.4 ผลการจำลองการสัญญาณของภาค CB_BUFFER	52
4.2.5 ผลการจำลองการสัญญาณของภาค ABS_SUB	53
4.2.6 ผลการจำลองการสัญญาณของภาค ADDER	55
4.2.7 ผลการจำลองการสัญญาณของภาค ADDRESS&INDEX	56
4.2.8 ผลการจำลองการสัญญาณของภาค CONTROL	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา	59
5.1 บทสรุป	59
5.2 ปัญหาและแนวทางการแก้ไข	59
5.3 แนวทางในการพัฒนา	59
ภาคผนวก ก รายละเอียดวงจร	60
ภาคผนวก ข โปรแกรม VHDL	67
บรรณานุกรม	77
ประวัติผู้แต่ง	78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 การควอนไตซ์เวกเตอร์แบบมาตรฐาน	5
รูปที่ 2.2 การจัดเก็บโค้ดบุ๊คในหน่วยความจำ	6
รูปที่ 2.3 วิธีการของ LBG โดยการประยุกต์จากวิธีการของ Lloyd	8
รูปที่ 2.4 ขั้นตอนของ Top – Down Design	12
รูปที่ 2.5 Dataflow Description ของ Multiply – Accumulate Function	14
รูปที่ 2.6 วงจรรวมประเภทต่างๆ	15
รูปที่ 2.7 แสดงแผนผังการแบ่งวงจรรวม	16
รูปที่ 2.8 (ก) วงจรอินเวอร์เตอร์	17
รูปที่ 2.8 (ข) เลย์เอาต์ของอินเวอร์เตอร์	17
รูปที่ 2.9 เป็น Standard Cell วงจร D flip-flop	19
รูปที่ 2.10 ผังการออกแบบ ASIC	21
รูปที่ 2.11 แสดงแผนผังการทำงาน Tanner Tools	27
รูปที่ 3.1 แสดงขั้นตอนการดำเนินการทำ Mask Layout	29
รูปที่ 3.2 แผนผังการทำงานของตัวเข้ารหัสและถอดรหัสเสียง	30
รูปที่ 3.3 (ก) Standard Cell DATA_BUFFER	31
รูปที่ 3.3 (ข) ผังงานของ DATA_BUFFER	32
รูปที่ 3.4 (ก) Standard Cell CB_BUFFER	33
รูปที่ 3.4 (ข) ผังงานของ CB_BUFFER	33
รูปที่ 3.5 (ก) Standard Cell ABS_SUB	34
รูปที่ 3.5 (ข) ผังงานของ ABS_SUB	34
รูปที่ 3.6 (ก) Standard Cell ADDER	35
รูปที่ 3.6 (ข) ผังงานของ ADDER	35
รูปที่ 3.7 (ก) Standard Cell COMPARATOR	36
รูปที่ 3.7 (ข) ผังงานของ COMPARATOR	36
รูปที่ 3.8 (ก) Standard Cell ADDRESS&INDEX	37
รูปที่ 3.8 (ข) ผังงานของ ADDRESS&INDEX	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 3.9 (ก) Standard Cell CONTROL	38
รูปที่ 3.9 (ข) ผังงานของ CONTROL	38
รูปที่ 3.10 (ก) Standard Cell SERIAL_OUT	39
รูปที่ 3.10 (ข) ผังงานของ SERIAL_OUT	39
รูปที่ 3.11 การตั้งชื่อ PAD	41
รูปที่ 3.12 ขั้นตอนการทำ Block Place&Route	42
รูปที่ 3.13 ผลของการ Initialization ในขั้นตอนการทำ BPR	43
รูปที่ 3.14 การทำ Floorplan ของวงจรเข้ารหัสถอดรหัสสัญญาณเสียง	44
รูปที่ 3.15 แผนผังการตรวจสอบความถูกต้องของ Layout	45
รูปที่ 3.16 ผลการทำ Block Routing	46
รูปที่ 4.1 ผังงานของ DATA_BUFFER	49
รูปที่ 4.2 ผลการเลียนแบบการทำงานของภาค DATA_BUFFER	49
รูปที่ 4.3 ผังงานของ COMPARATOR	50
รูปที่ 4.4 ผลการเลียนแบบการทำงานของภาค COMPARATOR	51
รูปที่ 4.5 ผังงานของ SERIAL_OUT	51
รูปที่ 4.6 ผลการเลียนแบบการทำงานของภาค SERIAL	52
รูปที่ 4.7 ผังงานของ CB_BUFFER	52
รูปที่ 4.8 ผลการเลียนแบบการทำงานของภาค CB_BUFFER	53
รูปที่ 4.9 ผังงานของ ABS_SUB	54
รูปที่ 4.10 ผลการเลียนแบบการทำงานของภาค ABS_SUB	54
รูปที่ 4.11 ผังงานของ ADDER	55
รูปที่ 4.12 ผลการเลียนแบบการทำงานของภาค ADDER	55
รูปที่ 4.13 ผังงานของ ADDRESS&INDEX	56
รูปที่ 4.14 ผลการเลียนแบบการทำงานของภาค ADDRESS&INDEX	56
รูปที่ 4.15 ผังงานของ CONTROL	57
รูปที่ 4.16 ผลการเลียนแบบการทำงานของภาค CONTROL	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ความก้าวหน้าของอุตสาหกรรมอิเล็กทรอนิกส์ในปัจจุบันมีความเจริญก้าวหน้าทางด้านการพัฒนาอุปกรณ์ต่างๆ ทางอิเล็กทรอนิกส์ได้อย่างมากมายซึ่งทำให้ลดการสิ้นเปลืองพลังงานและขนาด ในขณะเดียวกันนี้ก็ได้มีการเพิ่มประสิทธิภาพและระดับความน่าเชื่อถือได้ของวงจรที่สูงขึ้น เห็นได้ชัดเจนจากเทคโนโลยีไมโครโปรเซสเซอร์และหน่วยความจำปัจจุบันและในทุกๆ ครั้งที่ได้มีการพัฒนาขึ้นจะทำให้เกิดช่องว่างวงจรรวมและไอซีมาตราฐานมากขึ้นในการพัฒนานั้นจะมีการเพิ่มความหนาแน่นและจำนวนของฟังก์ชั่นลอจิกที่เหมาะสม นักออกแบบอุปกรณ์ทางด้านดิจิตอลได้พิจารณาถึงการผลิตให้ได้ขนาดมากๆ และการผลิตวงจรรวม (Appiication Specific Integrated Circuit : ASIC) ซึ่งวงจรรวมนั้นจะแบ่งตามการสร้างออกเป็นสองกลุ่มใหญ่ๆ คือ ฟิลด์โปรแกรม - เมเบิล (Field programmable) และ มาร์คโปรแกรมเมเบิล (Mask programmable) ซึ่งการออกแบบเช่นนี้จะทำให้ประสิทธิภาพของวงจรดีมากยิ่งขึ้นและความผิดเพี้ยนของสัญญาณน้อยลง ใช้พลังงานงานที่ต่ำและถ้าเป็นการออกแบบในทางด้านการสื่อสารก็จะทำให้ความละเอียดในการส่งโดยเฉพาะอย่างยิ่งในด้านการสื่อสารข้อมูลดีมากขึ้น

ดังนั้นโครงงานนี้จึงได้ออกแบบและสร้างวงจรรวมเฉพาะกิจของตัวเข้ารหัสและถอดรหัสข้อมูลเสียงที่ใช้วิธีของเวคเตอร์ควอนไตซ์ แบบมาตรฐานช่วยในการสุ่มสัญญาณทำให้มีการผิดเพี้ยนของสัญญาณน้อยลงและสามารถทำงานได้ที่เวลาจริงมากขึ้น โครงงานชิ้นนี้สามารถใช้ได้กับข้อมูลดิจิตอลขนาด 8 บิต และสามารถค้นหาข้อมูลภายในโค้ดบุ๊คที่มีขนาด 64×5 บิตได้ การประมวลผลจะเป็นแบบไปป์ไลน์ ซึ่งจะช่วยให้การทำงานเร็วขึ้น ในการออกแบบได้ใช้ภาษา VHDL บรรยายการทำงานของฮาร์ดแวร์ที่แสดงถึงความสัมพันธ์ระหว่างสัญญาณอินพุตกับสัญญาณเอาต์พุต และมีการสร้าง Layout โดยใช้ Tanner Tools ซึ่งเป็นซอฟต์แวร์ที่สามารถออกแบบทั้งในระดับ Netlist โดยการออกแบบ Schematic Diagram จำลองการทำงานระดับเกต (Gate-Level Simulation) และในระดับวงจร (Circuit-Level Simulation) และในระดับ Mask Layout โดยเริ่มจากการทำ Library Standard Cell เพื่อการ Place & Route ของวงจรที่ได้ จากนั้นทำการ Layout จาก Netlist File ที่ได้จากการออกแบบวงจรจากการใช้ภาษา VHDL โดยนำงานออกแบบตัวเข้ารหัสและถอดรหัสข้อมูลเสียงที่ใช้วิธีของเวคเตอร์ควอนไตซ์นี้มาทดสอบจำลองการทำงานด้วยโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

T-Spice ในปัจจุบันการออกแบบด้วยวิธีการขั้นสูงโดยใช้ภาษา VHDL หรือ Verilog แล้วนำไปสังเคราะห์ลงบนอุปกรณ์ที่เป็นฟิลด์โปรแกรมเมเบิล (Field programmable) เช่น FPGA และ CPLD จะพบว่ายังไม่เหมาะสมที่จะนำไปใช้งานจริงเนื่องจากอุปกรณ์จำพวกนี้มีข้อจำกัดในการนำไปใช้งานบางอย่างที่ต้องการประสิทธิภาพของวงจรเช่น ความเร็ว และการสร้างวงจรรวมจึงเป็นทางเลือกหนึ่งที่เป็นสิ่งจำเป็นและปัจจุบันเครื่องที่ใช้ก็มีราคาที่ถูกและหาได้

1.2 ขีดความสามารถของโครงงาน

1. สามารถใช้กับข้อมูลดิจิตอลขนาด 8 บิต
2. สามารถค้นหาข้อมูลในโค๊ดบุ๊คขนาด 64 ×5 บิตได้
3. สามารถเข้ารหัสและถอดรหัสสัญญาณเสียงได้
4. สามารถรับส่งข้อมูลในอัตราบอด 9,600 บิตต่อวินาที
5. ความถี่ที่ใช้ในการสุ่มสัญญาณ 8 กิโลเฮิรตซ์
6. ออกแบบด้วยเทคโนโลยี Stanard cell 0.5 ไมครอนหรือดีกว่า
7. สามารถเลียนแบบการทำงานของวงจรรวม
8. สร้าง Layout ที่เป็นต้นแบบของวงจรรวมตัวเข้ารหัสและถอดรหัสด้วยวิธีของเวคเตอร์ควอนไตซ์

1.3 เนื้อหาโดยสังเขป

เนื้อหาของปริญญานิพนธ์ฉบับนี้ ได้กล่าวถึงการออกแบบและสร้างตัวเข้ารหัสและถอดรหัสด้วยวิธีของเวคเตอร์ควอนไตซ์และใช้ภาษา VHDL บรรยายการทำงานของส่วนที่ออกแบบเป็นวงจรเชิงเลขเสร็จแล้วจึงนำไปทำกระบวนการสังเคราะห์ให้เป็นระดับเกตและระดับวงจรเพื่อนำไปทำกระบวน Place & Route และทำให้เป็น Layout หรือ mask สำหรับส่งไปเจือสาร ซึ่งการทำโครงงานทั้งหมดนี้จะใช้ทั้งในทฤษฎีของการออกแบบวงจรรวมเฉพาะกิจ และการใช้โปรแกรมของบริษัท Tanner เป็นเครื่องมือในการออกแบบ

บทที่ 2 ทฤษฎีและหลักการ กล่าวถึงการควอนไตซ์เวกเตอร์แบบมาตรฐาน การออกแบบโค้ดบุ๊ค การออกแบบ Top – Down Design โครงสร้างของภาษา VHDL ลักษณะการบรรยายด้วยภาษา VHDL การบรรยายเชิงพฤติกรรม การบรรยายเชิงข้อมูล การบรรยายเชิงโครงสร้าง ทฤษฎีและหลักการของการออกแบบวงจรรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การออกแบบ การสร้างและการทำงาน กล่าวถึงวงจรเข้ารหัสและถอดรหัส การทำงานในแต่ละภาค ขั้นตอนการใช้เครื่องมือสำหรับออกแบบวงจร โครงสร้างการออกแบบวงจรรวมในระดับ ASIC การเลียนแบบการทำงานของวงจรเพื่อทดสอบ

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงการนำ Layout ที่ได้มาในแต่ละส่วนย่อยมาทำกระบวนการการจำลองการ ในแต่ละบล็อกจากนั้นก็ทำการรวม Cell Layout ในแต่ละบล็อกที่ได้ไปทำกระบวนการการจำลองการบล็อกรวมทั้งหมด

บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา กล่าวถึงการสรุปผลและอภิปรายเกี่ยวกับความสามารถ ประสิทธิภาพของตัวเข้ารหัสและถอดรหัสข้อมูลเสียงที่ใช้วิธีของเวคเตอร์ควอนไตซ์ ปัญหาที่เกิดขึ้นระหว่างการทำโครงงาน แนวทางการแก้ไขปัญหาที่เกิดขึ้น และข้อเสนอแนะแนวทางในการพัฒนาปรับปรุงโครงงานให้มีประสิทธิภาพมากยิ่งขึ้น

ในภาคผนวกแสดงรายละเอียดของโปรแกรมและรายการอุปกรณ์ต่างๆ ที่ใช้ในการจัดทำโครงงานมีดังนี้

ภาคผนวก ก รายละเอียดของวงจร

ภาคผนวก ข ภาษา VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 กล่าวนำ

สำหรับทฤษฎีที่ใช้ในการออกแบบโครงงานนี้ประกอบด้วย ทฤษฎีของเวกเตอร์ควอนไตซ์ การออกแบบโค้ดบุ๊ค การออกแบบ Top-Down Design ทฤษฎีและหลักการ VLSI & ASIC และหลักการใช้เครื่องมือซึ่งจะนำเสนอเป็นส่วนๆ ดังต่อไปนี้

2.2 การควอนไตซ์เวกเตอร์แบบมาตรฐาน

2.2.1. หลักการควอนไตซ์เวกเตอร์แบบมาตรฐาน

การควอนไซต์เวกเตอร์แบบมาตรฐาน เป็นการควอนไตซ์เวกเตอร์แบบแรกสุดที่ได้มีการนำไปประยุกต์ใช้งานในการเข้ารหัสสัญญาณเสียงและเป็นพื้นฐานในการพัฒนาการควอนไตซ์เวกเตอร์ในเวลาต่อมา ซึ่งวิธีการควอนไตซ์เวกเตอร์แบบมาตรฐานนี้ได้นำเสนอวิธีการออกแบบโค้ดบุ๊คซึ่งเป็นที่รู้จักกันดี คือวิธีการแบบ LBG (LBG Algorithms) หรือในบางครั้งเราเรียกว่าวิธีการแบบ Splitting Algorithms ซึ่งได้มีพัฒนามาจากวิธีการของ Lloyd (Lloyd Algorithms) หลักการควอนไตซ์เวกเตอร์แบบมาตรฐาน

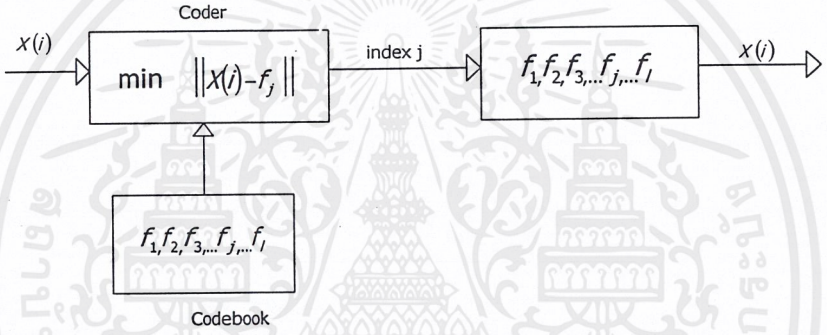
รูปที่ 2.1 เป็นหลักการพื้นฐานของการควอนไตซ์เวกเตอร์แบบมาตรฐาน วงจรเครื่องส่ง (Transmitter) ประกอบไปด้วยส่วนที่เป็นโค้ดบุ๊คเวกเตอร์ที่มี N มิติ ซึ่ง N คือ จำนวนของตัวอย่าง (Samples) หรืออีลีเมนต์ (Element) ในหนึ่งเวกเตอร์และมีจำนวน L เวกเตอร์คือ f_i $\{ j = 1,2,3,\ldots,L\}$ โดยเป็นส่วนที่มีความสำคัญเนื่องจากโค้ดบุ๊คจะต้องเป็นที่เก็บคุณลักษณะของเสียงไว้ทั้งหมดและการจัดโค้ดบุ๊คเวกเตอร์ในหน่วยความจำได้จัดในลักษณะของอะเรย์ 2 มิติ ซึ่งแสดงได้ดังรูปที่ 2.2 และจากรูปที่ 2.1 ในขบวนการเริ่มต้นสัญญาณอินพุตเวกเตอร์ $X(i)$ จะถูกสุ่มเข้ามาจำนวน N แซมเปิ้ล (เท่ากับมิติของโค้ดบุ๊คหรือหนึ่งเวกเตอร์) โดยวงจรเข้ารหัสจะแทนสัญญาณเวกเตอร์ $X(i)$ ด้วยค่าเวกเตอร์ f_j จากโค้ดบุ๊ค ซึ่งจะมีเพียงหนึ่งเวกเตอร์ที่มีคุณลักษณะใกล้เคียงกับสัญญาณอินพุตเวกเตอร์ $X(i)$ มากที่สุด โดยการหาค่าต่ำสุด (Minimize the Euclidean Distance) ซึ่งมีเงื่อนไขดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\|X(i)-f_j\|<\|X(i)-f_k\| \qquad (2.1)$$

โดยที่ $k=1,2,3,\ldots,L \qquad ; k \neq j$

และจะได้ค่าดัชนี (Index) ของเวกเตอร์จากโค้ดบุ๊คคือ j มาทำการถอดรหัสจากโค้ดบุ๊คที่เครื่องรับ โดยที่โค้ดบุ๊คที่เครื่องส่งและเครื่องรับจะต้องมีข้อมูลที่เหมือนกันหลังจากนั้นจะได้สัญญาณเอาต์พุตเวกเตอร์ที่ถูกสร้างขึ้นใหม่คือ เวกเตอร์ $X(i)$ ซึ่งมีค่าเท่ากับเวกเตอร์ f_j ที่เก็บอยู่ในโค้ดบุ๊คนั้นเอง โดยที่เวกเตอร์ f_j จะต้องมีค่าใกล้เคียงกับอินพุตเวกเตอร์ $X(i)$ มากที่สุด



รูปที่ 2.1 การควอนไตซ์เวกเตอร์แบบมาตรฐาน

ค่า Parameter ที่สำคัญอีกประการหนึ่งของควอนไตซ์เวกเตอร์ คือ ความละเอียด (Resolution) ของจำนวนบิตที่ใช้ในหนึ่งอีลีเมนต์ของเวกเตอร์สามารถคำนวณได้ดังนี้

$$R = \frac{\log_2 L}{N} \quad \text{(Bit /element)} \qquad (2.2)$$

สมมติถ้ากำหนดให้โค้ดบุ๊คมีมิติ $N=5$ และจำนวนเวกเตอร์ในโค้ดบุ๊ค $L=64$ จะได้

$$R = \frac{\log_2 64}{5} = \frac{6}{5} = 1.2 \quad \text{(Bit /element)} \qquad (2.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$f_{1(1)}$	$f_{1(2)}$	...	$f_{1(N)}$
$f_{2(1)}$	$f_{2(2)}$	...	$f_{2(N)}$
$f_{3(1)}$	$f_{3(2)}$	...	$f_{3(N)}$
.	.	.	.
.	.	.	.
.	.	.	.
$f_{L(1)}$	$f_{L(2)}$	...	$f_{L(N)}$

รูปที่ 2.2 การจัดเก็บโค้ดบุ๊คในหน่วยความจำ

นั่นหมายความว่า ในหนึ่งแซมเปิ้ลของการสุ่มสัญญาณที่ส่งไปยังเครื่องรับต้องการข้อมูลเพียง 1.2 บิต จะสังเกตเห็นว่า ความละเอียดของตัวแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลจะไม่มีผลต่อบิตเรตในการส่งข้อมูล ถ้าหากเรานำหลักการควอนไตซ์เวกเตอร์ไปประยุกต์ใช้ในการเข้ารหัสสัญญาณเสียงจากสมการที่ 2.2 สามารถคำนวณหาค่าบิตเรตในการส่งข้อมูลได้ดังนี้

$$\text{Bitrate} = R \times Fs \quad \text{(bit/sec)} \tag{2.4}$$

โดยที่

Fs = ความถี่ในการสุ่มสัญญาณ (Hz)

ถ้าอัตราการสุ่มสัญญาณ Fs = 8 kHz มิติของโค้ดบุ๊ค N = 5 (หมายถึงการเข้ารหัสในแต่ละครั้งใช้สัญญาณจำนวน 5 ตัวอย่าง) และจำนวนของเวกเตอร์ L = 64 นั่นหมายความว่าค่าดัชนีเวกเตอร์ j ที่ส่งไปยังเครื่องรับจำนวน 6 บิต สามารถสร้างสัญญาณเอาต์พุตเวกเตอร์ $X(i)$ ได้จำนวน 5 โดยที่ ดังนั้นจำนวนบิตเรตในการส่งข้อมูลจะได้เป็น

$$\text{Bitrate} = 1.2 \times 8000 = 9600 \text{ bps} \tag{2.5}$$

2.2.2 การคำนวณค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน

การวัดคุณภาพของเสียงที่ได้จากการควอนไตซ์เวกเตอร์กระทำได้โดยการทดสอบหาค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (SNRseg หรือ Segment Signal to Noise Ratio) ระหว่างสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุต X *(i)* และสัญญาณเอาต์พุต *X (n)* โดยทำการแบ่งสัญญาณ *X (n)* เป็นเฟรม พร้อมทั้งทำการคำนวณครั้งละหนึ่งเฟรม ซึ่งเขียนแทนด้วย SNRm และในแต่ละเฟรมจะใช้จำนวนตัวอย่าง 128 ตัวอย่างดังนี้

$$SNRm = 10\log_{10}\frac{\sum_{n=1}^{128}\{X(n)\}^2}{\sum_{n=1}^{128}\{X(n)-X(n)\}^2} \qquad (2.6)$$

เมื่อ $X(i)$ = สัญญาณอินพุต

$X(n)$ = สัญญาณที่ผ่านการควอนไตซ์เวกเตอร์

หลังจากนั้นจะนำเอาค่า SNRm ทั้งหมดที่ได้จากสมการที่ (2.11) มาทำการหาค่าเฉลี่ย ซึ่งเขียนแทนด้วย SNRseg ดังนี้

$$\text{SNRseg} = \frac{1}{S}\sum_{m=1}^{s} SNRm(dB) \qquad (2.7)$$

S คือ จำนวนของเฟรมทั้งหมด

สำหรับการคำนวณในแต่ละเฟรม ถ้าหาก $\sum_{n=1}(X(i))^2 < 32\times10^4$ (หรือต่ำกว่า 43 dB ของสัญญาณ Sinusoidal ที่แปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลแปลงได้เต็มสเกล) เราจะทำการตัดการคำนวณในเฟรมนี้ออกไป ทั้งนี้เนื่องจากสัญญาณที่เกิดขึ้นในระดับนี้ส่วนใหญ่จะเป็นสัญญาณรบกวน (Noise) ที่เกิดจากขบวนการควอนไตซ์ของตัวแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล เพราะสัญญาณเสียงที่พูดได้จะต้องมีพลังงาน (Energy) ที่มากกว่านี้

ในอดีตรูปแบบของสัญญาณไฟฟ้าโดยมากมักอยู่ในรูปสัญญาณแอนะลอก การนำเอาสัญญาณไฟฟ้ามาประมวลผลเพื่อให้เกิดรูปแบบที่ต้องการนั้น ต้องใช้อุปกรณ์ทางแอนะลอก แต่ปัจจุบันนี้เทคโนโลยีทางด้านดิจิตอลก้าวหน้าไปมาก ทำให้การประมวลผลสัญญาณทางดิจิตอลสามารถทำได้อย่างรวดเร็ว และมีประสิทธิภาพ

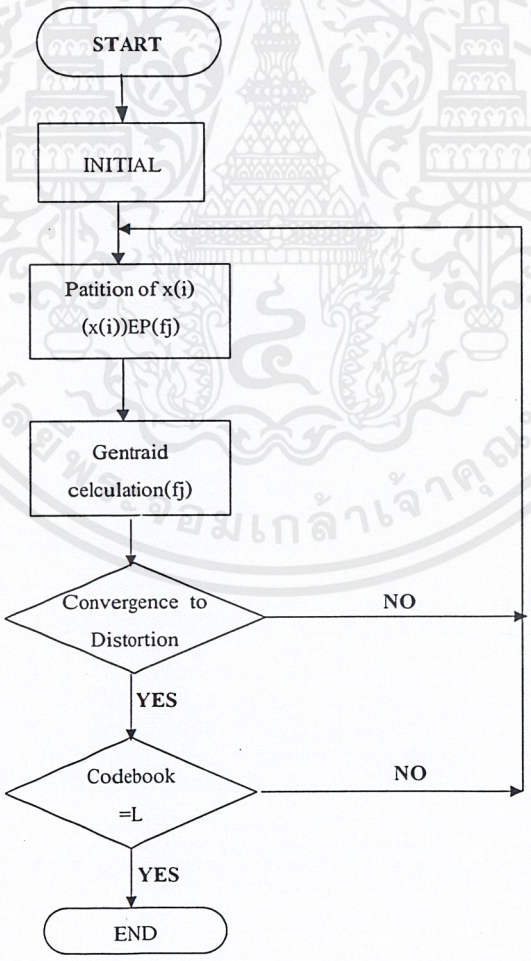
ดังนั้นการแปลงรูปแบบสัญญาณ (Conversion) จึงมีความจำเป็นในการแปลงสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอนะลอกให้เป็นสัญญาณดิจิตอล โดยอุปกรณ์การแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล จะถูกประมวลผลโดยตัวประมวลผลสัญญาณดิจิตอล เช่น คอมพิวเตอร์ เป็นต้น จากผลลัพธ์ที่ได้อาจถูกนำมาแสดงผลโดยตรงเลย หรืออาจถูกแปลงให้อยู่ในรูปของสัญญาณแอนะลอกที่ใช้งานได้ การที่จะแปลงสัญญาณแอนะลอกเป็นดิจิตอลได้นั้น สามารถทำได้โดยใช้อุปกรณ์แปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก สำหรับระบบที่มีการประมวลผลข้อมูลทางดิจิตอล

2.2.3 การออกแบบโค้ดบุ๊ค

ขั้นตอนการออกแบบจะนำเอาสัญญาณเสียงต้นฉบับที่ใช้ในการออกแบบโค้ดบุ๊ค $\{X(i)\}$ (Training Vector) มาทำการแยกกลุ่ม (Clustering) เพื่อหาคุณลักษณะของสัญญาณเสียงที่เหมาะสมที่สุดและให้มีความผิดพลาด (Distortion) น้อยที่สุด โดยกำหนดให้โค้ดบุ๊คเวกเตอร์ที่ต้องการออกแบบมีจำนวน L เวกเตอร์ ในแต่ละเวกเตอร์มีจำนวนอีลีเมนต์เท่ากับ N ซึ่งจากวิธีการของ LBG สามารถเขียนแผนผังงานการออกแบบโค้ดบุ๊คได้ดังรูปที่ 2.3



รูปที่ 2.3 วิธีการของ LBG โดยการประยุกต์จากวิธีการของ Lloyd

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งขั้นตอนการออกแบบโค้ดบุ๊คมีขั้นตอนการออกแบบดังนี้

กำหนดให้โค้ดบุ๊คเวกเตอร์ที่ต้องการมีจำนวน L เวกเตอร์ในแต่ละเวกเตอร์ มีจำนวนอีลีเมนต์เท่ากับ N

$P(f_j)$ = Voronoi Cell โดยมีโค้ดบุ๊คเวกเตอร์ f_j เป็นค่า Cenyriod ของ $P(f_j)$, $\{j = 1,2,3,\ldots L\}$

$\{X(i)\}$ = สัญญาณเสียงต้นฉบับสำหรับการออกแบบโค้ดบุ๊คโดย $\{i = 1,2,3,\ldots M\}$ และ M คือ จำนวนของเวกเตอร์ใน $\{X(i)\}$

α = เวกเตอร์สำหรับการ Spliting โดยจำนวนอีลีเมนต์เท่ากับ N ในที่นี้ใช้ค่า $\alpha = [0.01,0.01,\ldots 0.01]^t$

ε = Theshold เลือกค่า $\varepsilon = 0.005$

ขั้นตอนการออกแบบ

1. กำหนดให้ Iteration $m = 0$
2. ติดตั้งโค้ดบุ๊คเวกเตอร์ $f_{j(m)}$ ที่ $j = 1$ โดยหาค่า Centroid ของสัญญาณเสียงต้นฉบับ $\{X(i)\}$ ได้จาก

$$f_{j(m)} = \frac{1}{M}\sum_{i=1}^{M} x(i) \tag{2.8}$$

3. ทำการแยกโค้ดบุ๊คเวกเตอร์ $f_{j(m)}$ ออกเป็น 2 ส่วน โดยการบวกและลบด้วยเวกเตอร์ซึ่งมีจำนวนอีลีเมนต์เท่ากับโค้ดบุ๊คเวกเตอร์ และจะได้โค้ดบุ๊คเวกเตอร์ใหม่เป็น 2 เท่าของจำนวน เวกเตอร์เดิมดังนี้

$$f_{j(m)}^{new} = f_{j_{(m)}}^{old} - \alpha$$

$$f_{j+1(m)}^{new} = f_{j_{(m)}}^{old} - \alpha \tag{2.9}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. กำหนดให้ m = m+1

5. แยกสัญญาณเสียงต้นฉบับ $\{X(i)\}$ ให้เข้ากลุ่มของ $P(f_{j(m)})$ ดังนี้

$$x(i) \in p\left(f_{j(m)}\right) if \; X(i) = f_j$$

โดยที่ $\{i = 1,2,3,...,M\}$

$\{j = 1,2,3,...,L\}$;$k \neq j$ (2.10)

6. คำนวณหาค่า Centroid ในแต่ละ $P(f_{j(m)})$ โดยการนำเอา $\{X(i)\}$ ที่ถูกจัดกลุ่มในขั้นที่ 5 มาทำการคำนวณดังนี้

$$f_{j(m)} = \frac{1}{M_{jx(i)\in P(f_{j(m)})}}$$

$$f_{j(m)} = \frac{1}{M}\sum_i x(i) \quad (2.11)$$

โดยที่ $\{j = 1,2,3,...L\}$

เมื่อ

M_j คือ จำนวนของ training vector ในแต่ละ $p(f_{j(m)})$

7. คำนวณค่าความเพี้ยนเฉลี่ย (Average Distrotion) $D(m)$ รวมทั้งหมดของโค้ดบุ๊คเวกเตอร์ $f_{j(m)}$

$$D(m) = \sum_{j=1}^{L} \frac{M_j}{M} D_{j(m)} \quad (2.12)$$

โดยที่ $\{j = 1,2,3,...L\}$

โดยที่ $D_{j(m)}$ คือความเพี้ยนเฉลี่ยในแต่ละ $P(f_{j(m)})$ ซึ่งเกิดจากความเปรียบเทียบกันระหว่างสัญญาณเสียงต้นฉบับ $\{X(i)\}$ และโค้ดบุ๊คเวกเตอร์ $f_{j(m)}$ ในแต่ละ $P(f_{j(m)})$

$$D_{j(m)} = \frac{1}{M_{jx(i)\in P\left(f_{j(m)}\right)}} \left\| x(i) - f_{j(m)} \right\|^2 \quad (2.13)$$

โดยที่ $\{j = 1,2,3,...L\}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ตรวจสอบค่าความผิดเพี้ยนรวมทั้งหมดเทียบกับค่า Theshold

$$\frac{D(m-1)-D(m)}{D(m-1)} < \varepsilon \qquad (2.14)$$

เมื่อ $D(m-1)$ คือค่าความผิดเพี้ยนที่เกิดขึ้นในครั้งก่อน ถ้าข้อกำหนดจากสมการในข้อที่ 8 ไม่เป็นจริง ให้กลับไปทำในขั้นตอนที่ 4 และข้อกำหนดจากสมการในข้อที่ 8 เป็นจริงจะกำหนดให้ m = 0 และกระจายโค้ดบุ๊คเวกเตอร์ f_j ในแต่ละ $P(f_j)$ ออกเป็น 2 เวกเตอร์โดยการบวกและลบด้วยเวกเตอร์ เช่นเดียวกับขั้นตอนที่ 3 และให้กลับไปกระทำในขั้นตอนที่ 4 โดยการกระทำในแต่ละรอบจำนวนของโค้ดบุ๊คเวกเตอร์ f_j จะมีค่าเพิ่มขึ้นเป็น 2 เท่า ซึ่งจะได้โค้ดบุ๊คเวกเตอร์ f_j ในแต่ละ $P(f_j)$ ในแต่ละ $P(f_j)$ จำนวน L เวกเตอร์ อันเป็นการสิ้นสุดการออกแบบโค้ดบุ๊ค

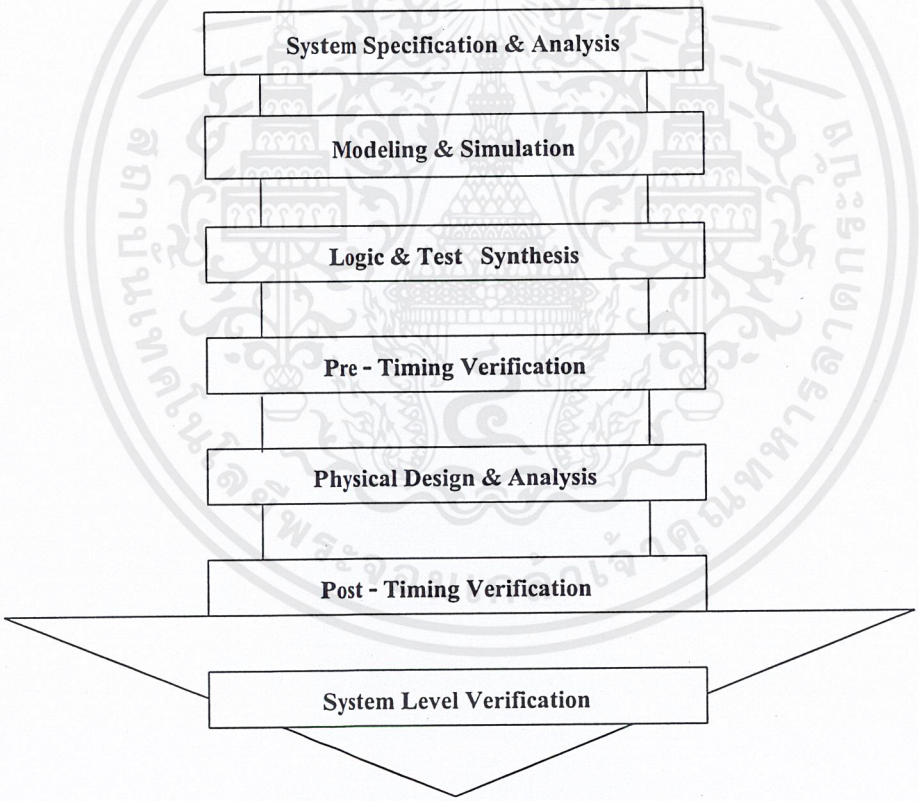
2.3. Top – Down Design

การนำภาษา VHDL มาใช้กำหนดและบรรยายพฤติกรรมฟังก์ชั่นการทำงานของฮาร์ดแวร์ในระบบดิจิตอลนั้น คือความอ่อนตัวของภาษาที่สามารถจำลองการทำงานจากหลักการของรูปแบบ (Simulate Conceptual Designs) แต่ในขณะเดียวกันก็สามารถจำลองการทำงานของฮาร์ดแวร์ ที่ให้รายละเอียดเกี่ยวกับเวลาอย่างถูกต้อง (Timing Based Simulation) และจากโครงสร้างของภาษายังสามารถจำลองการทำงานในรูปของลำดับชั้น (Hierarchy of Simulation Levels) ความสามารถดังกล่าวนี้ จึงช่วยให้การออกแบบสามารถที่จะเขียนรูปแบบบรรยายจากระดับบนสุดของวงจรที่อยู่ในรูปสังเขป (High Level of Abstraction) ลงสู่รายละเอียดในระดับล่างของวงจรได้ เช่น Gate Level เป็นต้น ในเวลานั้นวงการอุตสาหกรรมไมโครอิเล็กทรอนิกส์ ตลอดจนสถาบันวิจัยและศึกษา กำลังพัฒนาภาษาที่ใช้สำหรับการสังเคราะห์วงจรแบบอัตโนมัติ เพื่อลดเวลาในการพัฒนาวงจรลง ภาษา VHDL จึงถูกนำเข้าพิจารณาในโครงการนี้ด้วย โดยเพิ่มขีดความสามารถของภาษาขึ้น นอกเหนือจากเป็นภาษาที่ใช้สำหรับสังเคราะห์วงจร (Synthesis Language)

การเริ่มต้นด้วยวิธีการเขียนรูปแบบจากแนวความคิดอย่างสังเขป พร้อมทั้งการจำลองการทำงานของรูปแบบที่เขียนขึ้นเพื่อตรวจสอบความถูกต้อง ประกอบกับการกลั่นกรองเพื่อเพิ่มเติมรายละเอียดลงสู่ระบบดิจิตอลที่สมบูรณ์ในรูปของวงจรไฟฟ้าที่แต่ละขั้นนั้นเป็นขบวนการของ Top-Down Design การที่เริ่มต้นด้วยการเขียนรูปแบบในระดับบน (Top-Level) ของแนวความคิดอย่างสังเขปนั้น วิศวกรออกแบบสามารถที่จะพัฒนาสภาพแวดล้อมต่าง ๆ เพื่อการตรวจสอบการทำงานของวงจร (Test Environment) ได้ตั้งแต่ในระยะแรก ๆ ของการออกแบบ เพื่อใช้สำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรวจสอบความถูกต้องของแนวความคิดกับสิ่งที่ต้องการจริงหรือ Specification ของงาน ดังนั้น จึงเป็นไปได้ยากที่ในระดับล่างลงมาโดยที่ได้เพิ่มรายละเอียดของรูปแบบให้มากขึ้นตามลำดับจะเกิดข้อผิดพลาด หรือเบี่ยงเบนไปจากจุดประสงค์เดิม เพราะในแต่ละขั้นตอนย่อยจะมีการสร้างสภาพแวดล้อมเพื่อการตรวจสอบขึ้นใหม่ โดยอ้างอิงจากระดับที่อยู่สูงกว่าขึ้นไปเสมอ จากรูปที่ 2.4 แสดงให้เห็นขั้นตอนของการออกแบบในลักษณะของ Top-Down Design ทั้งนี้ในทางปฏิบัติอาจจะมีข้อแตกต่างไปจากนี้บ้างเล็กน้อย ก็เนื่องมาจากขั้นตอนของการผลิต (Implementation) สามารถกระทำได้ในหลาย ๆ เทคโนโลยี เช่น Programmable Logic Devices อันได้แก่ PLA, FPGAs หรือ CPLD เป็นต้น นอกจากนั้นยังมี Semi-Custom IC (Gate Array, Standard Cell) และ Full Custom IC



รูปที่ 2.4 ขั้นตอนของ Top-Down Design

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 ขบวนการออกแบบโดยใช้วิธี Top-Down Design

ขั้นตอนการออกแบบ Top-Down Design มีรายละเอียดดังนี้

1. System Specification and Analysis ขั้นตอนของการสร้างข้อกำหนดของความต้องการ (Specification) และวิเคราะห์ระบบ เพื่อหาแนวความคิดและหลักการ (Idea and Concept) ในการแก้ปัญหา

2. Modeling and Simulation การเขียนรูปแบบของระบบที่ต้องการจะออกแบบ โดยภาษา VHDL หรือ HDL อื่น ๆ จากแนวความคิดอย่างสังเขปที่ได้ สำหรับบรรยายพฤติกรรมการทำงาน พร้อมทั้งจำลองการทำงานเพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด

3. Logic and Test Synthesis หลังจากที่ได้หลักการขั้นต้นพร้อมกับแนวความคิดที่ผ่านการตรวจสอบแล้ว หลักการนี้จะถูกเพิ่มเติมในรายละเอียดลงมาเป็นลำดับขั้นตอนที่เหมือนกัน คือ Modeling and Simulation จนกระทั่งอยู่ในระดับที่จะนำไปผลิตวงจร หรือสังเคราะห์ (Synthesis) ในขั้นตอนนี้เอง เทคโนโลยีที่จะมารองรับวงจรออกแบบจะถูกกำหนดขึ้นและระบบช่วยการออกแบบจะสังเคราะห์วงจรที่ได้จากรูปแบบที่เขียนขึ้น ให้อยู่ในรูปของวงจรที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ และการเชื่อมต่อระหว่างกันของอุปกรณ์เหล่านั้น หรือไม่ก็อยู่ในรูปของ Netlist ที่สามารถนำไปผลิตลงบนอุปกรณ์อื่นได้ นอกจากนั้นการผลิตบางเทคโนโลยี อาทิเช่น Gate Array หรือ Standard Cell และ Full Custom IC อาจจะมีความจำเป็นที่ต้องสร้างโครงสร้างของวงจรใหม่หลังจากที่สังเคราะห์ครั้งแรกแล้ว เพื่อความสะดวกต่อการตรวจสอบการทำงานหลังจากที่ผลิตเป็นวงจรต้นแบบแล้ว หรือที่เรียกว่า "Design For Test" พร้อมทั้งข้อมูลในการตรวจสอบจะถูกกำหนดในขั้นตอนนี้

4. Pre-Timing Verification หลังจากการสังเคราะห์วงจรให้อยู่ในรูป Gate-Level หรือ Netlist แล้วข้อมูลที่ได้จากผู้ผลิตอุปกรณ์วงจรนั้น นอกจากจะเป็นข้อมูลสำหรับจำลองการทำงานในเรื่องของความถูกต้องของฟังก์ชัน (Functional Simulation) แล้ว ยังมีข้อมูลที่เกี่ยวกับเวลาด้วย ซึ่งเป็นความจริงที่ว่า อุปกรณ์ทางอิเล็กทรอนิกส์ทุกชิ้นจะมี Propagation Delay เสมอ ถึงแม้ว่าจะเป็นเวลาที่น้อยมากในระดับ nanosecond (10^{-9} second) แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกตของฟังก์ชันต่าง ๆ จำนวน 10,000 เกทขึ้นไป เวลาดังกล่าวนี้จะสะสมกันมากขึ้น จนอาจจะทำให้การทำงานของวงจรรวมทั้งหมดผิดไป หรือไม่สามารถทำงานในย่านความถี่สัญญาณนาฬิกาที่สูง

5. Physical Design and Analysis คือขั้นตอนของการผลิตเป็นวงจรจริง (Technology and Device Mapping) โดยนำข้อมูลที่ได้จากการสังเคราะห์มาผลิต ซึ่งอาจจะอยู่ในรูปของแผงวงจรไฟฟ้าที่ประกอบด้วยอุปกรณ์หลายๆ ชิ้น หรืออยู่ในรูปของวงจรรวมเฉพาะงาน (ASIC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

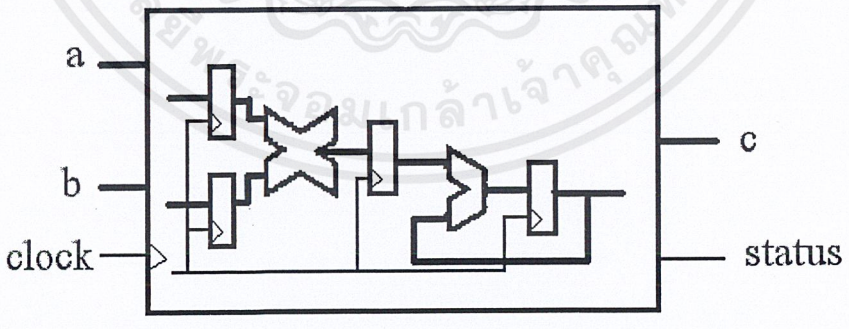
6. Post-Timing Verification หลังจากที่ได้วงจรจริงมาเรียบร้อยแล้ว ยังต้องมีความจำเป็นที่ต้องตรวจสอบการทำงานที่คำนึงถึงเวลาด้วย เพื่อความถูกต้องของวงจรครั้งสุดท้ายก่อนที่จะนำไปรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอล เพราะในขั้นตอนนี้วงจรที่ออกแบบ จะประกอบด้วย Input และ Output Pad ซึ่งเป็นจุดต่อสำหรับรับและส่งสัญญาณกับภายนอก

7. System Level Verification หลังจากที่นำวงจรที่ออกแบบรวมเข้ากับอุปกรณ์อื่นๆ ให้เป็นระบบดิจิตอลแล้วนั้น จะต้องทดสอบการทำงานรวมทั้งระบบร่วมกับอุปกรณ์อื่นๆ อีกครั้งเป็นการควบคุมคุณภาพของผลิตภัณฑ์

ในตัวอย่างของ multiply accumulate algorithm แสดงให้เห็นขบวนการออกแบบในลักษณะของ Top-Down Design ได้โดยการใช้ Multiply Accumulate Function จุดประสงค์แรกก็คือ การเขียนรูปแบบของฟังก์ชั่นในระดับบนสุดด้วยภาษา VHDL และจำลองการทำงานของรูปแบบที่ได้เพื่อตรวจสอบความถูกต้องซึ่งฟังก์ชันนี้อาจจะเป็น Discrete Fourier Transform (DFT) ก็ได้ หลังจากที่ได้ฟังก์ชั่นที่ต้องการแล้วและมีชื่อว่า “multiply _accum” สามารถนำไปใช้ได้ในรูปของ Function Call คือ

```
C <= multiply_accum(a,b);                    (2.15)
```

ถ้าฟังก์ชั่นเป็นที่พอใจแล้วในขั้นต่อไปจะเป็นการเริ่มของการแปลงแนวความคิดไปสู่การสร้างวงจรจริง ขบวนการดังกล่าวเกิดขึ้นโดยการเพิ่มรายละเอียดให้กับแนวความคิดเดิม รูปที่ 2.5 แสดงให้เห็นวิธีการบรรยายแบบ Dataflow (Dataflow Description) ของ Multiply Accumulate Function



รูปที่ 2.5 Dataflow Description ของ Mutiply-Accumulate Function

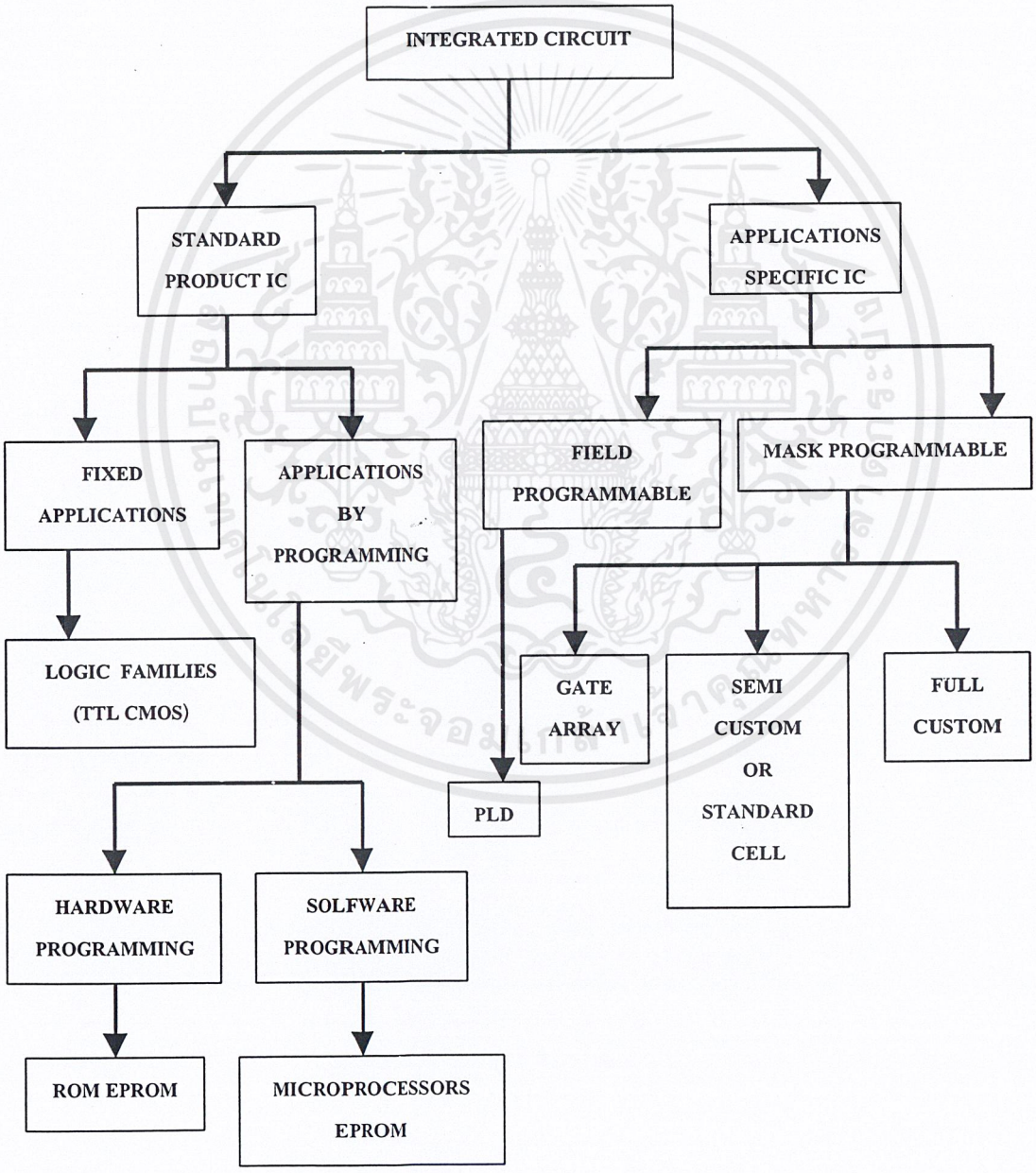
จาก Dataflow Format ที่ได้นี้ ขบวนการต่อไปคือการสร้างวงจรให้อยู่ในรูปของอุปกรณ์พื้นฐานหรือ Gate-Level Implementation ซึ่งวิธีนี้อาจใช้วิธีการสังเคราะห์อัตโนมัติ ผลลัพธ์ที่ได้จากการสังเคราะห์จะเป็นการแสดงภาพวงจรด้วยเกตของฟังก์ชันต่างๆ หรือในรูปของ Netlist และจะเป็นการกำหนดเทคโนโลยีจำเพาะที่จะนำไปผลิตในภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติหลักของภาษา VHDL คือสามารถที่จะบรรยายฮาร์ดแวร์ ได้ในทุกๆ ระดับของภาพรวมทั้งระบบ ฉะนั้น ผู้ออกแบบจึงสามารถใช้เครื่องมือ (ภาษา) เพียงอันเดียวในการบรรยายทั้งระบบ ซึ่งก็เช่นเดียวกันกับเครื่องมือจำลองการทำงาน

2.4 หลักการออกแบบวงจรรวม Application Specific Integrated Circuit

นับตั้งแต่กลางคริสศตวรรษที่ 20 ได้มีการพัฒนาวงจรรวมขนาดเล็กระดับ Small Scale



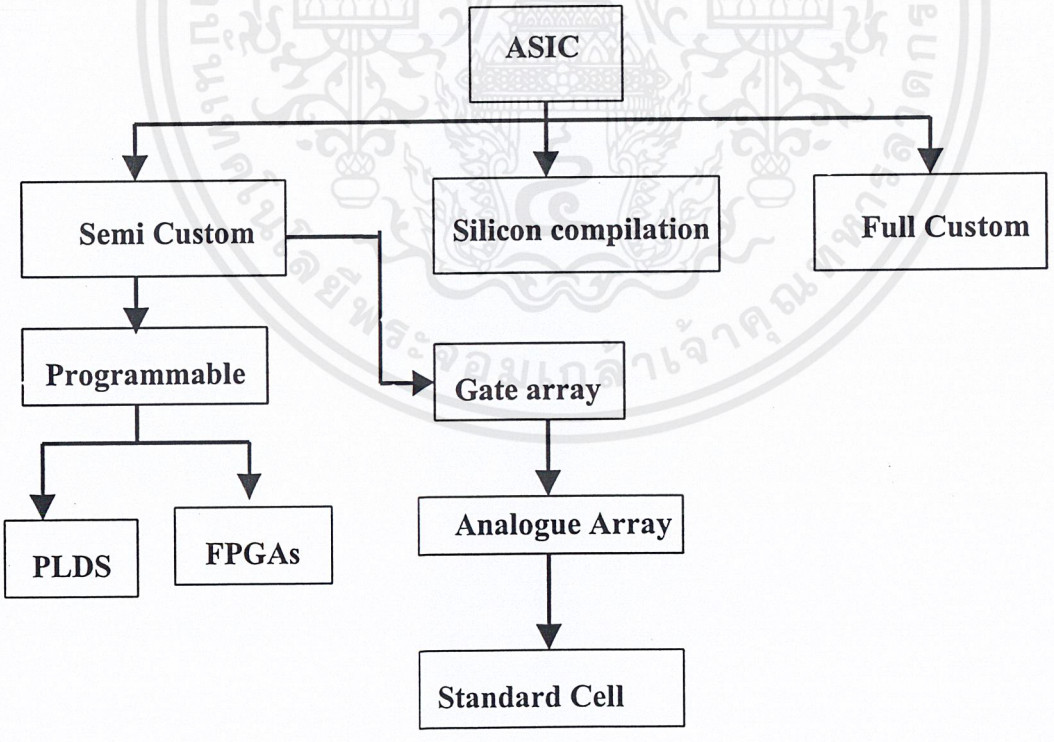
รูปที่ 2.6 วงจรรวมประเภทต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Integrated (SSI) จนกระทั่งขึ้นมาถึงวงจรรวมขนาดใหญ่มากระดับ Very Large Scale Integrated (VLSI) ในปัจจุบัน มีการแบ่งวิธีการออกแบบและใช้งานวงจรรวมออกเป็น 2 แบบใหญ่ คือ Standard Product Integrated Circuit (SPIC) และ Application Specific Integrated Circuit (ASIC)ดังรูปที่ 2.6 และปริญญานิพนธ์ฉบับนี้จะกล่าวถึงวงจรรวมขนาดใหญ่มากชนิด ASIC ซึ่งเป็นวงจรรวมที่ถูกออกแบบและใช้งานโดยผู้ใช้เป็นผู้กำหนดหน้าที่การทำงานของวงจรรวมนั้นตามความต้องการ วงจรรวม ASIC เกิดจากการที่ผู้ใช้ต้องการวงจรรวมที่มี ลักษณะแตกต่างไปจากที่มีอยู่ในท้องตลาด การออกแบบสร้างอาจใช้วิธีการนำวงจรรวมพื้นฐานตั้งแต่ขนาดเล็กถึงขนาดใหญ่เข้ามาบรรจุในวงจรรวมตัวเดียวกัน หรือออกแบบเองตั้งแต่การนำทรานซิสเตอร์ประกอบเป็นวงจรขนาดเล็ก และประกอบเข้ากันเป็นวงจรรวมก็ได้

2.4.1 วงจรรวม ASIC ชนิดต่าง ๆ

การใช้งานวงจรรวม ASIC ในเชิงพาณิชย์ จำเป็นต้องใช้วงจรรวม ASIC แบบ Mask Programmable เนื่องจากต้นทุนต่อ 1 ตัวต่ำกว่าวงจรรวมแบบ Standard Products ในกรณีที่ปริมาณการผลิตสูงนับพันนับหมื่นตัวขึ้นไป ยกตัวอย่างเช่น วงจรรวม EPLD ตัวหนึ่งอาจมีราคาสูงถึงหนึ่ง



รูปที่ 2.7 ผังการแบ่งวงจรรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

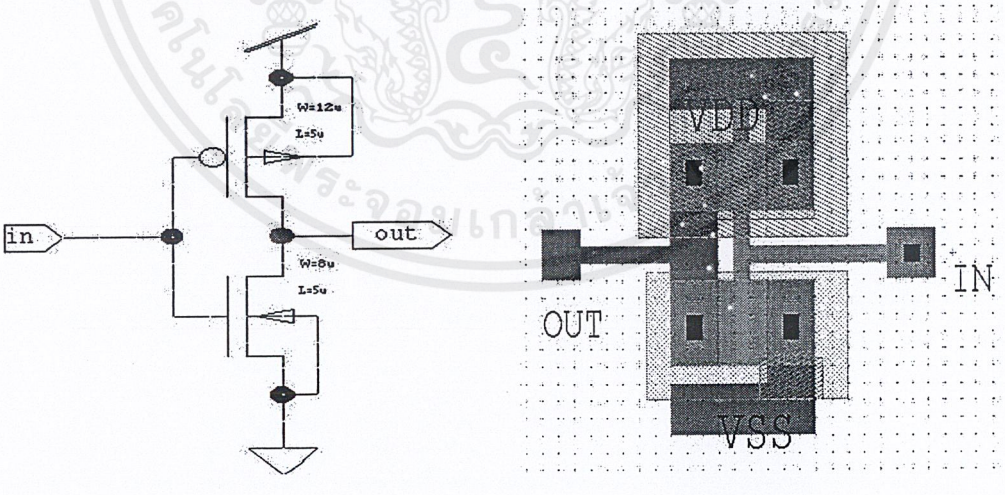
พันบาท ในขณะที่ถ้าผลิตวงจรรวมที่มีคุณสมบัติเหมือนกันทุกประการโดยใช้ Mask Programmable แล้ว ราคาตัวหนึ่งจะลดลงเหลือเพียงไม่ถึง 100 บาท การใช้วงจรรวมแบบ Mask Programmable จึงมีบทบาทสำคัญในการผลิตสินค้าอิเล็กทรอนิกส์ในเชิงพาณิชย์ในปัจจุบัน

วงจรรวม Mask Programmable ASIC นี้ หลังจากผู้ใช้ออกแบบและตรวจสอบการทำงานจนเป็นที่พอใจแล้ว ต้องส่งให้ผู้ผลิตทำการเจือสาร ไม่สามารถโปรแกรมได้ด้วยตนเองเหมือนวงจรรวมแบบ Field Programmable ช่วงเวลาการผลิตออกมาใช้งานจึงใช้เวลานับเดือนและมีค่าใช้จ่ายเบื้องต้นในการเจือสารสูง วงจรรวมแบบ Mask Programmable ASIC ในปัจจุบันมีการแบบออกเป็น 3 กลุ่ม คือ Full Custom , Silicon Compilation และ Semi-custom ในทั้ง 3 กลุ่มนี้มี เพียงกลุ่มของ Semi-custom เท่านั้นที่ยังมีการแบ่งแยกออกไปได้อีกดังแสดงในแผนผังรูปที่ 2.7

2.4.2 วงจรรวม Full custom

วงจรรวม Full custom นี้ผู้ใช้เป็นผู้ออกแบบเองทั้งหมด ตั้งแต่ระดับวงจรเชิงเลขจนกระทั่งถึงระดับล่างสุดจนทำเป็นเลเอาต์ออกมาแล้วจึงส่งข้อมูลการออกแบบไปยังผู้ผลิตเพื่อนำไปเจือสาร จะจัดอยู่ในรูปของไฟล์จุด GDS II หรือ CIF

ในที่นี้จะยกตัวอย่างการออกแบบในลักษณะแบบที่เป็น Full custom ในแบบอนาล็อกโดยใช้ ซีมอสสร้างเป็นวงจร invertor และในแบบที่เป็นเลเอาต์ได้ดังรูปที่ 2.8 (ก)และรูปที่ 2.8(ข)



(ก) (ข)

รูปที่ 2.8 (ก) วงจรอินเวอร์เตอร์ (ข) เลย์เอาต์ของอินเวอร์เตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

40180

จากในรูปที่ 2.8 (ก) และ (ข) เป็นลักษณะของการออกแบบในแบบที่เป็น Full Custom ก็คือต้องออกแบบจากในระดับล่างที่เป็นทรานซิสเตอร์เสียก่อนดังในรูป (ก) จากนั้นก็มาออกแบบในลักษณะที่เป็นส่วนของ Layout และในการทำ Layout นั้นจะต้องมีปัจจัยในการวาดด้วยก็คือกฎการออกแบบ (Design Rules) ซึ่งเป็นจะเป็นส่วนหนึ่งของเทคโนโลยีที่ใช้ในผลิตเช่น เทคโนโลยีการผลิต CMOS 0.5 ไมครอนของโรงงาน Alcatal Microelectrincs ในการออกแบบของFull Custom นั้นผู้ออกแบบจำเป็นต้องมีความรู้ในเรื่องของพื้นฐาน CMOS ว่ามีโครงสร้างอย่างไรเพราะในการวาด Layout ใน Full Customนั้นต้องวาดจากระดับที่เป็นชั้นสารที่ได้มาจากโครงสร้างของN-MOS และP-MOSสารเหตุที่เลือกใช้เทคโนโลยีของสารกึ่งตัวนำนั้นก็เพราะเป็นเทคโนโลยีที่ผลิตได้ง่าย และยังมีข้อดีก็คือการประหยัดพลังงานกล่าวคือในปัจจุบันจะใช้เทคโนโลยีสารกึ่งตัวนำมาเป็นเทคโนยีที่ใช้ในการออกแบบและผลิตวงจรรวม ASIC

2.4.3 วงจรรวมประเภท Semi Custom

วงจรรวมในประเภทนี้ผู้ผลิตได้ทำการรวบรวมวงจรพื้นฐานต่างไว้ในวงจรรวมเพียงตัวเดียวกัน ซึ่งผู้ใช้จะเป็นผู้ที่ทำการเชื่อมโยงวงจรพื้นฐานที่มีอยู่ให้เป็นวงจรรวมที่มีขนาดใหญ่มากที่สามารถงานได้ตามความต้องการ โดยที่จะใช้เครื่องคอมพิวเตอร์เป็นเครื่องมือที่ช่วยในการออกแบบซึ่งในวงจรรวมASIC แบบนี้สามารถแบ่งออกเป็น 3 กลุ่มได้แก่

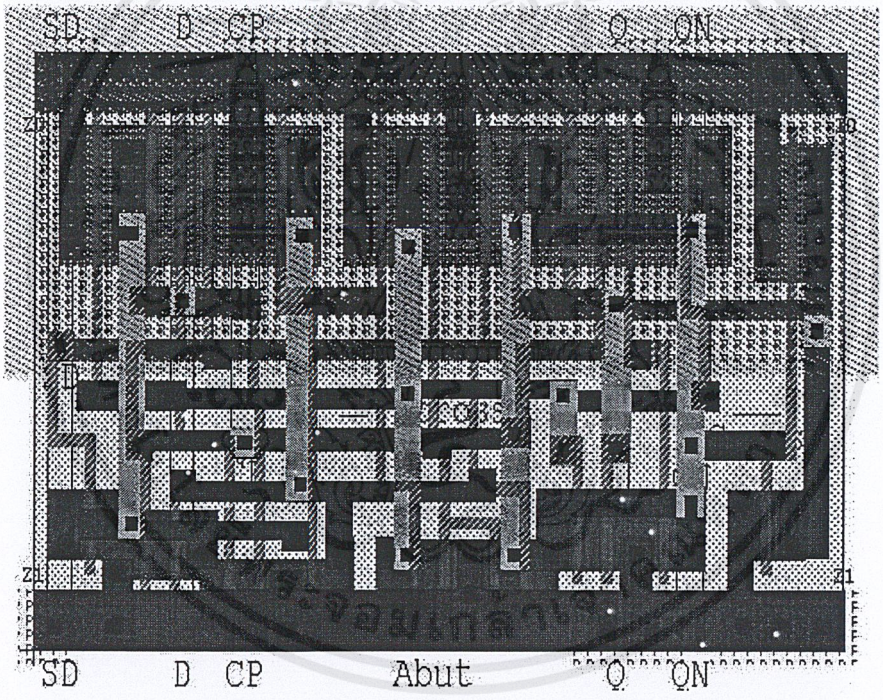
1. Programmable Logic Devices (PLD) วงจรรวม ASIC ในกลุ่มนี้ผู้ผลิตได้บรรจุวงจรพื้นฐานประเภทเกท (Gate) แบบต่างๆเช่นAND, OR , INVERTER ไว้ในวงจรรวมเพียงตัวเดียว ผู้ใช้สามารถออกแบบเชื่อมโยงวงจรประเภทนี้เข้าเป็นวงจรเชิงเลขขนาดใหญ่ที่ทำงานได้ตามต้องการได้และเมื่อออกแบบเสร็จแล้วก็สามารถผลิตขึ้นใช้ เองได้โดยไม่ต้องสั่งให้ผู้ผลิตทำการเจือสารแต่อย่างใด ระยะเวลานับจากการออกแบบจนกระทั่งผลิตขึ้นใช้งานจึงสั้นที่สุดเมื่อเทียบกับวงจรรวม ASIC ในกลุ่มอื่นๆ จึงเหมาะสมที่จะใชพัฒนาต้นแบบหรือผลิตสินค้าที่มีจำนวนขายเป็นสิบเป็นร้อยตัวเท่านั้น

2. Gate Array วงจร ASIC ประเภทนี้ผู้ผลิตจะบรรจุวงจรพื้นฐานประเภทเกท แบบต่างๆ ไว้ในวงจรรวมตัวเดียว และวงจรเกทบางส่วนจะต่อสายภายในไว้เป็นกลุ่ม วงจรเชิงเลขที่ใช้งานกันทั่วไปได้แก่วงจร Latch วงจร Decoder วงจร Register เป็นต้น กลุ่มของวงจรเชิงเลขที่รวบรวมบรรจุไว้ในวงจร Gate Array นี้เรียกว่า Macrocell หรือ Logic-Cell ที่ผู้ใช้มีหน้าที่ออกแบบต่อสายเชื่อมโยงระหว่าง Macrocell แต่ละตัวเข้าด้วยกันให้เป็นวงจรเชิงเลขขนาดใหญ่ที่ทำงานได้ตามต้องการแล้วส่งข้อมูลการออกแบบนี้ไปให้ผู้ผลิตทำการเจือสารต่อไป การออกแบบวงจร Gate Array นี้ผู้ใช้ต้องมีข้อมูลที่เกี่ยวกับตัววงจรรวม Gate Array และ Macrocell ที่บรรจุอยู่ภายในภายวงจรรวมตัวนั้นซึ่งตามปกติผู้ผลิตจะส่งมาให้โดยไม่คิดค่าใช้จ่าย เนื่องจากวงจรรวมประเภทนี้ต้องทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเจือสารโดยผู้ผลิตหลังการออกแบบทำให้มีต้นทุนการผลิตสูงเป็นเงินหลายแสนบาทในการผลิตในแต่ละครั้งจึงเหมาะสำหรับการออกแบบในเชิงพาณิชย์ที่มีปริมาณผลิตตั้งแต่หนึ่งพันตัวขึ้นไป

3. Standard Cell ปัญหาการใช้พื้นที่ซิลิคอนอย่างมากมายสำหรับการเชื่อมโยงวงจรของวงจรรวม Gate array ทำให้ เกิดขีดจำกัดในความซับซ้อนของวงจรที่จะใช้ ประกอบกับผู้ใช้จำนวนมากต้องการรวบรวมวงจรรวมมาตราฐาน อาทิ วงจรรวมตระกูล 7400 วงจรรวมตระกูล 4000 จำนวนหลายตัวเข้าเป็นวงจรรวมASIC เพียงตัวเดียว ทำให้เกิดวงจรรวม ASIC แบบ Standard cell ขึ้น วงจรรวม Standard cell นี้ ผู้ใช้ เป็นผู้เลือกกลุ่มวงจรทำหน้าที่ต่าง ๆ เช่น เกท ฟลิปฟลอป ตัวนับ ตัวเลื่อน หน่วยความจำ หรือแม้ กระทั่งไมโครโปรเซสเซอร์จากแฟ้มข้อมูล (Library) ของผู้ผลิต ซึ่งอาจจะเป็นแฟ้มข้อมูล



รูปที่ 2.9 Standard Cell วงจร D flip-flop

คอมพิวเตอร์ หรือสมุดเหมาะสมสำหรับวงจรเชิงเลขที่มีความซับซ้อน การผลิตวงจรรวม Standard cell จะมีต้นทุนสูงกว่าวงจรรวม Gate array และใช้เวลาในการออกแบบและเจือสารยาวกว่าสองถึงสามเท่าตัว วงจรรวม Standard cell จึงเหมาะสมกับการใช้งานในเชิงพาณิชย์ที่มีปริมาณการผลิตนับหมื่นตัวขึ้นไป บทบาทของวงจรรวม Standard cell จะมีเพิ่มมากขึ้นในอนาคตอันใกล้ เมื่อต้นทุนการเจือสารลดลง และวงจรรวม Programmable Gate array สามารถบรรจุจำนวนเกทได้มากขึ้นจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระทั่งทัดเทียมกับมวลรวมGate array ในปัจจุบัน รูปที่ 2.9 แสดงตัวอย่าง Standard cell จากแฟ้มข้อมูลของผู้ผลิตรายหนึ่งผู้ใช้จะนำข้อมูล Standard cell มาออกแบบวงจรที่ตนต้องการ แล้วส่งผลการออกแบบให้ผู้ผลิตทำการเจือสาร ซึ่งแผ่นซิลิคอนที่เจือสารแล้วจะประกอบด้วยพื้นที่ของ cell ต่าง ๆ

2.4.4 Design Flow

ขั้นตอนในการออกแบบและสร้างวงจรรวม ASICในประเภทที่เป็น Semi Custom นั้นจะมีการนำคอมพิวเตอร์เป็นเครื่องมาช่วยในการออกแบบ ซึ่งในการออกแบบ ในการออกแบบวงจรรวม ASIC ในประเภท Semi Custom ที่ใช้คอมพิวเตอร์มาช่วยจะประกอบด้วยขั้นตอนที่สำคัญดังแสดงให้เห็นในรูปที่ 2.10 ซึ่งมีรายละเอียดดังนี้

1. Design Entry ผู้ออกแบบจะเริ่มต้นด้วยการกำหนดรายละเอียดหน้าที่การทำงานของวงจรที่ตนเองต้องการ (Function Specification) แล้วจึงแปลงรายละเอียดนี้ให้อยู่ในลักษณะที่คอมพิวเตอร์เข้าใจซึ่งจะใช้ ภาษาพรรณนาฮาร์ดแวร์ (Hardware Description Language หรือ HDL)

2. Logic Synthesis จะใช้ภาษา VHDLหรือ Verilog หรืออาจจะใช้เป็น Schematic มาทำการสังเคราะห์ออกมาให้ได้เป็นไฟล์ Netlist โดยใช้เครื่องมือในการสังเคราะห์

3. System partitioning เป็นการตรวจสอบขนาดของวงจรรวมที่ได้มาการ Netlist

4. Prelayout simulation เป็นการนำเอาไฟล์ Netlist ที่ได้มาการแปลงของ VHDL มาเปรียบเทียบความถูกต้องของวงจรที่ได้ออกแบบโดยจำลองการทำงาน

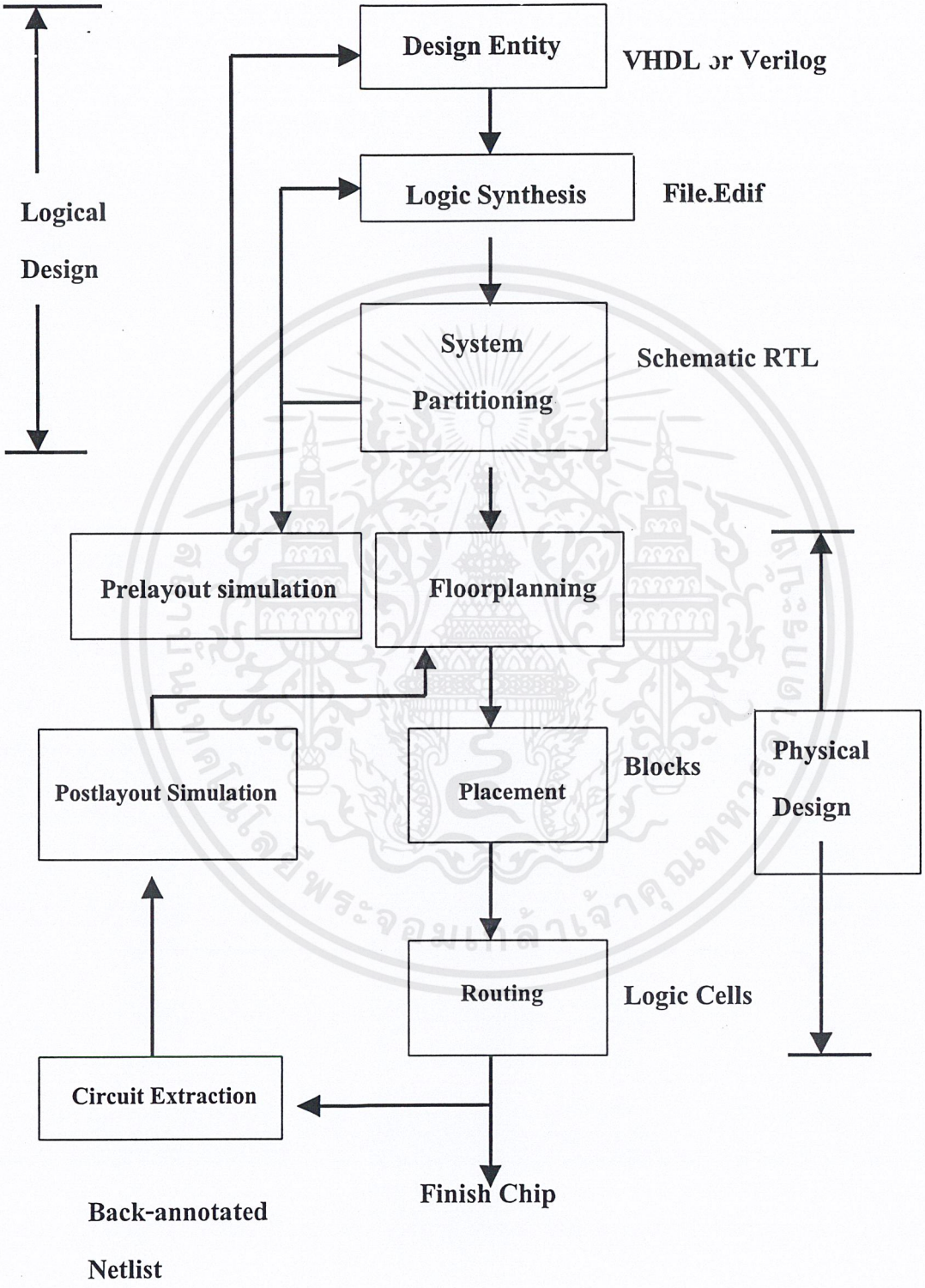
5. Floorpalannig เป็นการร่างการวาง Block ต่างๆ ที่ทำเป็นชิฟให้มีความเหมาะสม

6. Placecment&routing เป็นขั้นตอนการจัดเรียงพื้นที่ต่างของ Cell และทำการเชื่อมต่อจุดต่างๆ

2.5 ภาษา VHDL

ภาษา VHDL เริ่มประมาณปี ค.ศ.1981 โดยที่กระทรวงกลาโหมสหรัฐอเมริกา หรือ Department Of Defense (DOD) เห็นว่าอุปกรณ์อิเล็กทรอนิกส์และคอมพิวเตอร์ที่ใช้ในกิจการทางทหาร เป็นอุปกรณ์ที่ได้รับการพัฒนามาเมื่อประมาณ 20 ปีก่อน ในขณะนั้นการพัฒนาอุปกรณ์อิเล็กทรอนิกส์เป็นไปอย่างล่าช้า เพราะเทคโนโลยีทางด้านไมโครอิเล็กทรอนิกส์ ได้รับการพัฒนาไปอย่างรวดเร็ว ดังที่เห็นได้ว่ามีวงจรดิจิตอลอิเล็กทรอนิกส์หลายวงจร ที่แต่เดิมถูกสร้างขึ้นมาจากชิ้นส่วนอุปกรณ์อิเล็กทรอนิกส์จำนวนหลายชิ้นประกอบกันอยู่บนแผงวงจรไฟฟ้า ที่มีขนาดใหญ่ แต่ในปัจจุบัน สามารถที่จะใช้เทคโนโลยีการออกแบบและผลิตวงจรรวมขนาดใหญ่มาก รวมอุปกรณ์ต่าง ๆ เหล่านั้นให้อยู่บนชิ้นอุปกรณ์สารกึ่งตัวนำ ที่มีขนาดประมาณ 1-2 ตร.ซม. ได้ ซึ่งเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 ฝังการออกแบบ ASIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลให้ประสิทธิภาพในการทำงานของวงจรสูงขึ้น ตลอดจนความน่าเชื่อถือและความคงทนต่อสภาพแวดล้อมสูง ขณะเดียวกันนั้นในวงการทหารได้มีการนำระบบคอมพิวเตอร์และอิเล็กทรอนิกส์มาใช้ในระบบอาวุธอย่างแพร่หลาย ดังนั้นอุปกรณ์ที่มีใช้อยู่จึงไม่เหมาะสมกับเทคโนโลยีด้านอาวุธของประเทศคู่แข่งขัน การที่จะเปลี่ยนอุปกรณ์ใหม่เป็นสิ่งที่ต้องใช้งบประมาณมาก และก็จะประสบกับปัญหาเช่นเดิมคือ อุปกรณ์ใหม่ได้รับการพัฒนามานานแล้วเช่นกัน เพราะในขณะนั้นขั้นตอนของการออกแบบ การผลิต และการตรวจสอบวงจรต้นแบบ เป็นขบวนการที่ต้องใช้วิศวกร และเวลาสำหรับดำเนินการมาก ฉะนั้นทาง DOD จึงตั้งโครงการขึ้นมาเพื่อศึกษา วิธีการที่จะช่วยพัฒนาวงจรอิเล็กทรอนิกส์ โดยเฉพาะอย่างยิ่งวงจรระบบดิจิตอล ให้สามารถนำไปผลิตได้เร็วขึ้น และโครงการดังกล่าวมีชื่อว่า Very High Speed Integrated Circuits หรือ VHSIC ในระยะแรกนั้นโครงการเป็นความลับของประเทศและอยู่ในความดูแลควบคุมของ United States International Traffic and Arms Regulations (ITAR) ในปี ค.ศ. 1983 ตามคำแนะนำของคณะทำงาน (Woods Hole workshop) ทาง DODได้กำหนดความต้องการมาตรฐานของภาษาที่ใช้สำหรับบรรยายพฤติกรรมของ วงจรหรือฮาร์ดแวร์ของระบบสำหรับโครงการ VHSIC ซึ่งมีสาระสำคัญพอสรุปได้ดังนี้

1. ต้องเป็นภาษาที่นำไปเขียนรูปแบบระบบดิจิตอล และมีคุณสมบัติที่สามารถจะเข้าใจได้ทั้งคนและเครื่องโดยไม่ต้องมีการแปลหรือเปลี่ยนแปลงอีก
2. สามารถนำไปใช้เป็นเอกสารประกอบโครงการได้ (Project Documentation)
3. ต้องเป็นภาษาที่เขียนขึ้นสำหรับใช้จำลองการทำงานของวงจร (Simulation Language)

ฉะนั้นภาษาดังกล่าวนี้จึงเป็นภาษาโปรแกรมระดับสูง (High Level Language) เช่นเดียวกับภาษา PASCAL, FORTRAN และ ADA ซึ่งในทางวิศวกรรมการออกแบบฮาร์ดแวร์ เรียกว่า Hardware Description Language หรือ HDL ดังนั้นภาษามาตรฐานนี้จึงมีชื่อว่า VHSIC-HDL หรือ VHDL นั้นเอง

โครงการ DOD ได้มอบหมายให้บริษัท IBM, Texas Instruments และ Intermetrics เป็นผู้ศึกษาและพัฒนา การดำเนินงานได้กระทำไปอย่างต่อเนื่อง และได้ผลเป็นที่น่าพอใจ จนกระทั่งปี ค.ศ. 1985 ทาง ITAR ได้ยกเลิกข้อจำกัดในการถ่ายทอดเทคโนโลยีทางการทหารออกจากโครงการนี้ ดังนั้น VHDL จึงเริ่มเป็นที่รู้จักกันโดยทั่วไป จนกระทั่งทาง IEEE จึงได้รับภาษานี้เข้ามาศึกษาประมาณปี ค.ศ. 1987 ได้ยอมรับกำหนดมาตรฐานของภาษา โดยให้ชื่อว่า IEEE 1076-1987 และมีชื่อเรียกว่า VHDL มาตรฐานนี้ได้รับการปรับปรุงจนปัจจุบัน ได้ชื่อว่า IEEE 1076-1993 หรือ VHDL 1993

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Terminology and Conventions

การเขียนรูปแบบของระบบดิจิตอลด้วยภาษา VHDL นั้นจะมีศัพท์เทคนิคเฉพาะ ฉะนั้นจะอธิบายศัพท์บางคำที่จะต้องพบ

1. ลักษณะของรูปแบบ (model styles) : ลักษณะของการเขียนรูปแบบด้วยภาษา VHDL สามารถแบ่งได้เป็น

1) Behavioral Model หรือเรียกอีกอย่างได้ว่า Algorithmic description เป็นรูปแบบที่บรรยายพฤติกรรมของระบบดิจิตอล ในส่วนที่บรรยายมีโครงสร้างคล้ายกับภาษาชั้นสูง (High Level Language) ทั่วๆ ไป เช่น PASCAL หรือ ภาษา C เป็นต้น ในการจำลองการทำงานคำสั่งแต่ละคำสั่ง (Statement) จะถูกประเมินผลเป็นไปตามลำดับ (Sequential) จากบนลงล่าง ยกเว้นในกรณีของคำสั่ง LOOP หรือการเรียกใช้โปรแกรมย่อย รูปแบบนี้จะไม่ให้รายละเอียดที่เกี่ยวกับผลิตหรือโครงสร้างของฮาร์ดแวร์ แต่ในทางตรงข้ามจะให้รายละเอียดเกี่ยวกับความสัมพันธ์ระหว่างอินพุตกับเอาต์พุต

2) Dataflow Model เรียกอีกอย่างหนึ่งได้ว่า "Register Transfer Level" (RTL) เป็นรูปแบบที่ถูกเขียนขึ้นเพื่อจุดประสงค์ที่จะใช้เครื่องมือสำหรับสังเคราะห์วงจรอัตโนมัติรูปแบบลักษณะนี้ส่วนใหญ่จะเป็น Procedural Constructs และ Functional Operators

3) Structural Model เป็นรูปแบบที่แสดงการเชื่อมต่อกันระหว่างอุปกรณ์ต่างๆ ที่ประกอบกันขึ้นเป็นวงจรหรือระบบดิจิตอลและสามารถเรียกอีกอย่างได้ว่า "Netlist Representation" เป็นการเขียนที่แสดงให้เห็นโครงสร้างของฮาร์ดแวร์

4) Mixed-Level Model จากความอ่อนตัวของภาษา VHDL จึงสามารถที่จะเขียนรูปแบบโดยใช้ลักษณะต่างๆ ที่กล่าวมาแล้วข้างต้น บรรยายวงจรหรือระบบดิจิตอลเดียวกันได้ฉะนั้นรูปแบบเช่นนี้จึงมีการเขียนแบบผสม

2. Concurrency ในภาษา VHDL นั้นชุดคำสั่งแต่ละชุดจะทำงานในเวลาเดียวกันและอิสระต่อกันซึ่งเป็นคุณสมบัติที่เป็นความจริงทางฟิสิกส์ของวงจรอิเล็กทรอนิกส์ ชุดคำสั่งนี้เรียกว่า "Concurrent Statement" และจะทำงานก็ต่อเมื่อมีการเปลี่ยนแปลงค่าของสัญญาณ

3. Sequential นอกจากความสามารถที่จะทำงานแบบ Concurrent แล้ว บางครั้งการเขียนรูปแบบในลักษณะที่บรรยายพฤติกรรมของวงจรมีความจำเป็นที่จะต้องให้ชุดคำสั่งทำงานเป็นลำดับขั้นเรียงกันจากบนลงล่าง อย่างเช่นการเขียนแบบ Behavioral Model เป็นต้น ชุดคำสั่งที่เป็น Sequential นี้จะใช้ในโปรแกรมย่อยและ Process Statement

4. Driver สัญญาณต่างๆ ใน VHDL นั้นจะถูกควบคุมด้วยตัวขับหรือ "Driver" สัญญาณเหล่านี้จะรับค่าใหม่ (ระดับของสัญญาณ) ได้ด้วยตัวขับนี้เอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Transaction การเกิด Transaction กับ Signal นั้นเกิดขึ้นเมื่อมีการกำหนดค่าๆ หนึ่งให้กับสัญญาณนั้น ค่าใหม่ที่สัญญาณได้รับอาจจะมีหรือไม่มีผลทำให้เกิดการเปลี่ยนแปลงของระดับสัญญาณ (event) เช่นการเปลี่ยนจากค่า Logic '0' เป็นค่า Logic '1' เป็นต้น

6. Event คือการเปลี่ยนระดับค่าของ SIGNAL จากระดับหนึ่งไปสู่ระดับอื่น อย่างเช่นในระบบดิจิตอลการเปลี่ยนจาก Logic '0' เป็น Logic '1' หรือในทางตรงกันข้ามถือว่า SIGNAL นั้นเกิด "Event" ฉะนั้นจะเห็นได้ว่า การที่จะเกิด Event ได้นั้นจะต้องเกิด Transaction ด้วย แต่ในทางตรงข้ามการเกิด Transaction ไม่จำเป็นต้องเกิด Event ทุกครั้ง

7. Sensitivity List คือรายชื่อของ SIGNAL ต่างๆ ที่มีผลให้เกิดการทำงานของ Concurrent Statement เมื่อเกิด Event ขึ้นกับ SIGNAL ตัวใดตัวหนึ่งหรือหลายตัวพร้อมกันในรายชื่อนั้น

8. Objects ในภาษา VHDL นั้นคำว่า Object ใช้เขียนเพื่อบ่งบอกถึงองค์ประกอบส่วนหนึ่งของรูปแบบ ซึ่งเปรียบได้เหมือนกับภาชนะที่มีไว้สำหรับบรรจุค่าต่างๆ สามารถแบ่งออกได้เป็นสามชั้นด้วยกันคือ

1) CONSTANT ได้แก่ Object ประเภทหนึ่งที่เมื่อกำหนดค่าเริ่มต้นให้แล้วจะคงค่านั้นไว้ตลอด ไม่สามารถดัดแปลงหรือแก้ไขได้ สามารถประกาศใช้ได้ในส่วนที่เป็นส่วนประกาศของรูปแบบ

2) SIGNAL หมายถึง Object ประเภทหนึ่งที่สามารถกำหนดค่าที่สัมพันธ์กับเวลาให้ได้นั้นหมายความว่า SIGNAL สามารถรับค่าได้เพียงค่าเดียวเท่านั้นในขณะเวลาหนึ่ง SIGNAL จะรับค่าๆ หนึ่งได้จากตัวขับสัญญาณหรือ Driver ซึ่งตัวขับนี้อาจจะเก็บค่าในอนาคตสำหรับ SIGNAL ไว้ด้วย SIGNAL สามารถประกาศใช้ได้ในส่วนที่เป็นเนื้อที่ของ Concurrent Body เท่านั้น ดังนั้น SIGNAL จึงสามารถถูกนำไปใช้ได้ตลอดโครงสร้างของรูปแบบหรือที่เรียกว่า Gobal Object

3) VARIABLE หรือตัวแปรได้แก่ Object ที่สามารถกำหนดค่าใดๆ ได้และสามารถที่จะเปลี่ยนแปลงค่าได้ตลอดการจำลองการทำงาน แต่จะเก็บค่าเพียงค่าเดียวเท่านั้น เนื่องจาก VARIABLE สามารถประกาศใช้ได้ในส่วนที่เป็นส่วนประกาศของ PROCESS, FUNCTION หรือ PROCEDURE ดังนั้น VARIABLE จึงสามารถนำไปใช้ได้เฉพาะในขอบเขตที่ถูกประกาศใช้

2.5.1 โครงสร้างของภาษา VHDL

VHDL นั้นประกอบด้วยส่วนต่างๆ ที่สำคัญและเป็นพื้นฐานของการเขียนรูปแบบระบบดิจิตอลที่สำคัญ 4 หน่วยคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Entity Design Unit
2. Architecture Design Unit
3. Package Design Unit
4. Configuration Design Unit

2.5.2 ลักษณะของการเขียนบรรยายด้วยภาษา VHDL

การบรรยายเชิงพฤติกรรม เป็นการบรรยายลักษณะการเปลี่ยนแปลงของข้อมูลในรูปแบบของอัลกอริธึม สำหรับการคำนวณผลลัพธ์ที่เกิดขึ้นสืบเนื่องมาจากการเปลี่ยนแปลงสภาวะของข้อมูลที่เข้ามาโดยไม่คำนึงถึงว่าลักษณะโครงสร้างหรือความสัมพันธ์ของอุปกรณ์ที่อยู่ภายใน

การบรรยายเชิงข้อมูล เป็นการบรรยายถึงการเลื่อนไหลของข้อมูลผ่านรีจิสเตอร์และบัสของระบบ เป็นระดับขั้นการบรรยายที่อยู่ตรงกลางระหว่างการบรรยายเชิงพฤติกรรมและการบรรยายเชิงโครงสร้าง เครื่องมือที่ใช้ในการควบคุมการเลื่อนไหลของข้อมูลได้แก่ Conditional Selected และ Guarded

การบรรยายเชิงโครงสร้าง การบรรยายการทำงานของระบบในเชิงโครงสร้างจะต้องแสดงรายการของอุปกรณ์ทั้งหมดที่ใช้ในระบบและต้องกำหนดการเชื่อมต่อระหว่างอุปกรณ์ต่าง ๆ ด้วย เพราะว่าการบรรยายในระดับนี้เป็นการบรรยายที่ใกล้เคียงลักษณะของฮาร์ดแวร์จริงที่สุด

2.6 เครื่องมือที่ใช้ออกแบบวงจรรวม

การออกแบบวงจรรวมโดยใช้ Tanner Tools ซึ่งเป็นซอฟแวร์ที่สามารถออกแบบได้ทั้งในระดับ Netlist โดยการออกแบบ Schematic Diagram จำลองการทำงานระดับเกต (Gate-Level Simulation) กับในระดับวงจร (Circuit-Level Simulation) และในระดับ Mask Layout โดยเริ่มจากทำ Library Standard Cell เพื่อการ Place&Route ของวงจรที่ได้ จากนั้นทำ Layout จาก Netlist File ที่ได้จากการออกแบบวงจรจากการใช้ภาษา VHDL โดยนำงานออกแบบวงจรรวมเฉพาะกิจตัวเข้ารหัสและถอดรหัสข้อมูลเสียงที่ใช้วิธีควอนไตซ์เวคเตอร์ มาทำการทดสอบการ Simulation กับ T-Spice ปัจจุบันสามารถทำการออกแบบ ด้วยวิธีการออกแบบขั้นสูง โดยใช้ VHDL หรือVerilog แล้วทำการสังเคราะห์ลงบน FPGA แล้ว แต่ยังไม่เหมาะสมที่จะใช้งานจริง เนื่องจาก FPGA มีข้อจำกัดในการนำไปใช้งานบางอย่างที่ต้องการประสิทธิภาพของวงจร เช่น ความเร็ว การสร้างวงจรรวมซึ่งเป็นสิ่งที่จำเป็น แต่ปัจจุบันสามารถทำได้โดยเครื่องมือที่มีราคาถูกและประสิทธิภาพเพียงพอที่จะใช้ในการออกแบบอย่างเช่น Tanner Tools ™ [3]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 การใช้ Tanner Tools™ System

Tanner Tools เป็นซอฟแวร์ที่ช่วยในการออกแบบวงจรรวมได้อย่างสมบูรณ์บนเครื่องคอมพิวเตอร์ PC ประกอบด้วย 3 ส่วนหลัก ๆ แสดงดังรูปที่ 2.11

ส่วนที่ 1 ส่วนจำลองการทำงานของวงจร

T-Spice เป็นส่วนจำลองการทำงานของวงจรอนาล็อกและดิจิตอล

W-Edit เป็นตัวแสดงสัญญาณ Waveform ที่ได้จาก T-Spice

Gate-Sim ใช้จำลองการทำงานของวงจรดิจิตอลร่วมกับค่า Timing Delays ในการทำ Static Timing Analysis โดยสามารถจำลองการทำงานของวงจรที่มีขนาดใหญ่

ส่วนที่ 2 ส่วนออกแบบในระดับ Netlist

S-Edit เป็นซอฟแวร์ที่ช่วยใช้สร้างและแก้ไข Circuit Schematics ของวงจรได้ทั้งอนาล็อกและดิจิตอล โดยมีความสัมพันธ์กับซอฟแวร์อื่น ๆ

LVS เป็นเครื่องมือที่ใช้เปรียบเทียบ SPICE Netlist ที่ได้จาก Schematics กับ SPICE Netlist ที่ถูก Extract จาก Layout เพื่อให้ทราบว่า Layout ที่ออกแบบเป็นวงจรเดียวกัน

ส่วนที่ 3 ส่วนออกแบบในระดับ Layout

L-Edit เป็น Low-Level, Full Custom Mask Editor ซึ่งไม่สามารถทำแปลงเลเยอร์ของ Layout ได้โดยอัตโนมัติ (Automatic Layer Transformations) ได้ L-Edit สามารถอ่านและบันทึกในรูปแบบของ Standard Mask Layout Interchange Formats (GDS II และ CIF) หรือโครงสร้างข้อมูลของ Tanner (TDB = Tanner Database) ซึ่งสามารถทำได้เร็วกว่าสองแบบข้างต้นโดยโครงสร้างข้อมูลของ TDB จะถูกใช้ในระหว่างกระบวนการออกแบบเพื่อเรียกใช้และเก็บข้อมูลจากการออกแบบ ส่วน CIF กับ GDS II จะใช้ในการถ่ายทอดข้อมูลไปยังโรงงานผลิตชิพ หรือ CAD Tools ตัวอื่น ๆ ในส่วนของการกำหนดขนาดของช่องไฟใน L-Edit Grid Option สามารถใช้ได้กับกฎการออกแบบได้ทั้งแบบแลมด้าและไมครอน

L-Edit/SPR จะสร้าง Layout สำหรับ Standard Cell Design รวมถึงการทำ Placement, Routing และ Pad Routing

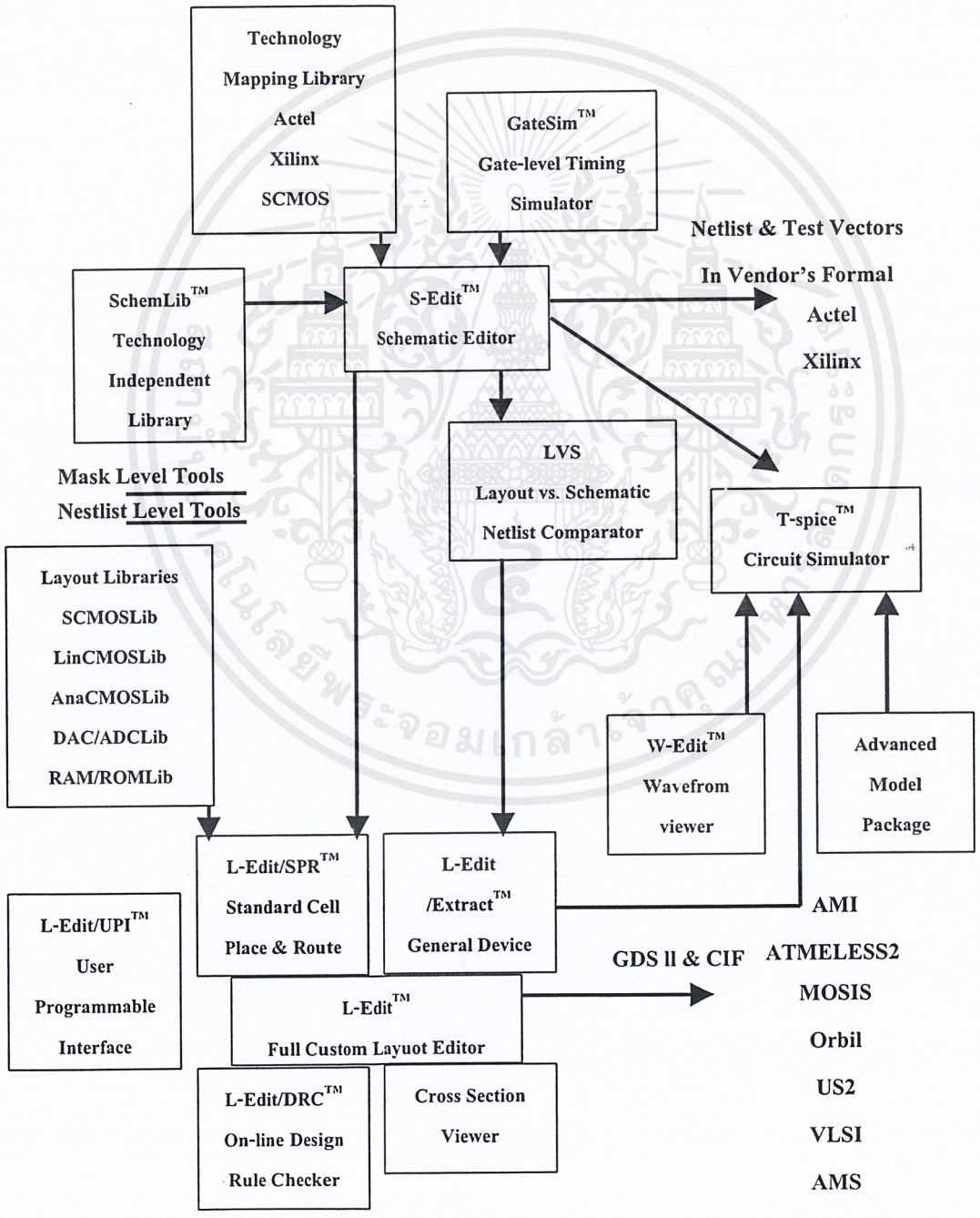
L-Edit/Extract จะสร้าง SPICE-Compatible Circuit Netlist จาก L-Edit Layout

L-Edit/DRC จะมีส่วนของ User Programmable Rules เช่น Minimum Width, Exact Width เป็นต้น และมีส่วนของ Error Browser, Object Browser สำหรับทำ Rule-Checking Error และสนับสนุนโครงสร้างแบบ 45 องศา และ 90 องศา

นอกจากนี้ยังมี ASIC Library ของ MOSIS และ ORBIT [4] ให้ใช้ในการออกแบบ ซึ่ง Library ทั้งหมดประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. SCMOSLib : ดิจิตอล ลอจิกและ I/O Pads
2. DAC/ADCLib : วงจร DAC และ ADC
3. Lin CMOSLib : วงจรเชิงเส้น
4. Ana CMOSLib : วงจรอนาล็อก
5. RAM/ROMLib : ใช้สร้าง Layout ของ RAM หรือ ROM โดย Library ทั้งหมดสามารถใช้กับเทคโนโลยี 2.0 ,1.2,1.0 และ 0.8 ไมครอน



รูปที่ 2.11 แผนผังการทำงาน Tanner Tools

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ การสร้าง และการทำงาน

ในการออกแบบและการสร้างออกแบบและสร้างวงจรรวมเฉพาะกิจตัวเข้ารหัสและถอดรหัสข้อมูลเสียงที่ใช้วิธีควอนไตซ์เวคเตอร์ นั้นเป็นวิธีการออกแบบโดยใช้ VHDL อธิบายการทำงานของระบบทั้งหมดแล้วจึงสังเคราะห์ออกมาเป็นวงจรโดยใช้ซอฟต์แวร์ที่ใช้สังเคราะห์วงจร ดังนั้นในบทนี้ จึงนำส่วนที่ออกแบบไว้แล้วสร้าง Mask Layout ด้วยเทคโนโลยี การผลิตแบบ CMOS 0.5 ไมครอน ของโรงงาน Alcatel Microelectronic

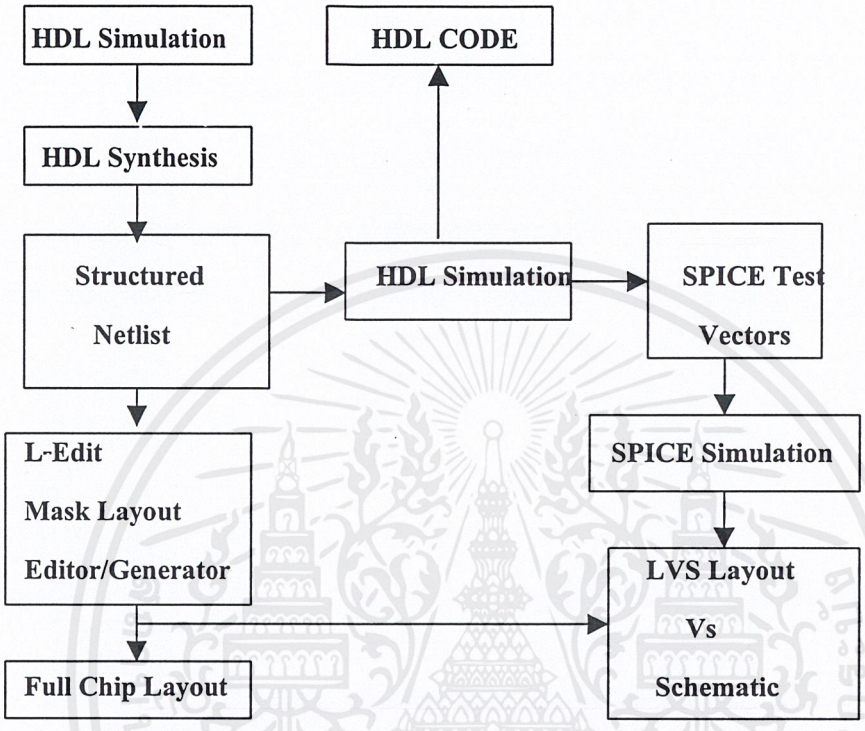
3.1 การใช้ Tanner Tool

การออกแบบวงจรรวมโดยใช้ Tanner Tools ซึ่งเป็นซอฟแวร์ที่สามารถออกแบบได้ทั้งในระดับ Netlist โดยการออกแบบ Schematic Diagram จำลองการทำงานระดับเกต (Gate-Level Simulation) กับในระดับวงจร (Circuit-Level Simulation) และในระดับ Mask Layout โดยเริ่มจากทำ Library Standard Cell เพื่อการ Place&Route ของวงจรที่ได้ จากนั้นทำ Layout จาก Netlist File ที่ได้จากการออกแบบวงจรจากการใช้ภาษา VHDL โดยนำงานออกแบบวงจรรวมเฉพาะกิจตัวเข้ารหัสและถอดรหัสข้อมูลเสียงที่ใช้วิธีควอนไตซ์เวคเตอร์มาทำการทดสอบการ Simulation กับ T-Spice

ปัจจุบันสามารถทำการออกแบบ ด้วยวิธีการออกแบบขั้นสูง โดยใช้ VHDL หรือVerilog แล้วทำการสังเคราะห์ลงบน FPGA แล้ว แต่ยังไม่เหมาะสมที่จะใช้งานจริง เนื่องจาก FPGA มีข้อจำกัดในการนำไปใช้งานบางอย่างที่ต้องการประสิทธิภาพของวงจร เช่น ความเร็ว การสร้างวงจรรวมซึ่งเป็นสิ่งที่จำเป็น แต่ปัจจุบันสามารถทำได้โดยเครื่องมือที่มีราคาถูกและประสิทธิภาพเพียงพอที่จะใช้ในการออกแบบอย่างเช่น Tanner Tools ™ [3]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 ขั้นตอนในการออกแบบ



รูปที่ 3.1 ขั้นตอนการดำเนินการทำ Mask Layout

ขั้นตอนออกแบบและการสร้าง Layout ในแต่ละส่วน

1. สร้าง Standard Cell สำหรับ Place&Route จาก Black Box GDSII File ที่อยู่ใน Design Kit ของโรงงานด้วยโปรแกรม L-Edit

2. นำ Netlist ของวงจรแต่ละส่วนที่ได้มาทำ Standard Cell Place&Route

3. การ Extract Layout ของ Block ที่ได้จากการ Place&Route

4. การตรวจสอบความถูกต้องของ Layout ที่ได้จากทำ DRC

5. การสร้าง Padframe และทำ Block Place&Route (BPR)

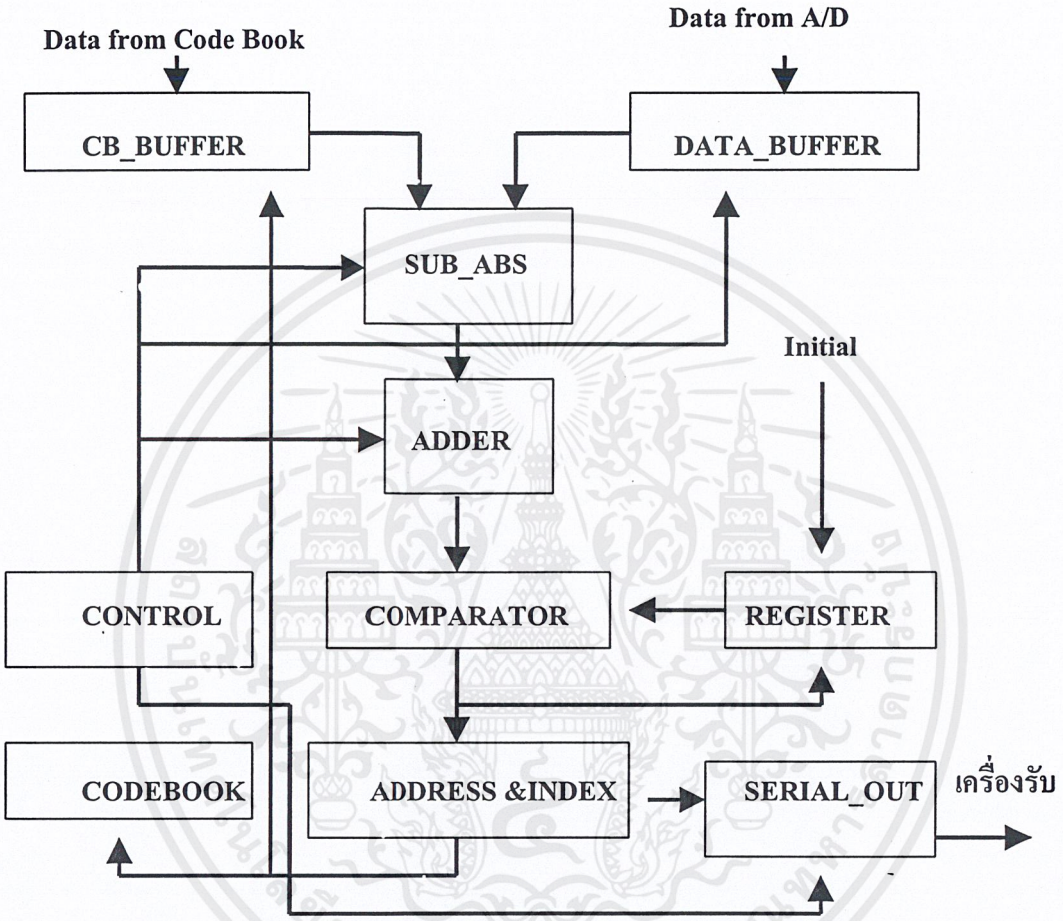
6. การ Extract Layout เพื่อตรวจสอบหลังการทำ Block Routing

7. การตรวจสอบความถูกต้องของ Layout ด้วยการ DRC&ERC

ขั้นตอนการทำงานตั้งแต่การรับ Netlist ของไฟล์ที่มีนามสกุล VHD แล้วแปลงให้เป็นไฟล์ที่มี นามสกุล EDF นั้น เพื่อดำเนินการ Place&Route สร้าง Mask Layout โดยใช้ L-Edit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ส่วนประกอบต่าง ๆ ของ ตัวเข้ารหัสและถอดรหัสข้อมูลเสียง



รูปที่ 3.2 แผนผังการทำงานของตัวเข้ารหัสและถอดรหัสเสียง

ตัวเข้ารหัสและถอดข้อมูลเสียง เป็นการนำสัญญาณจากตัวแปลงสัญญาณ แอนะลอกเป็นสัญญาณดิจิตอลมาเปรียบเทียบกับสัญญาณจากโค้ดบุ๊ค ว่ามีค่าใกล้เคียงกับสัญญาณในโค้ดบุ๊คตำแหน่งใดมากที่สุด แล้วทำการส่งตำแหน่งของสัญญาณชุดนั้นออกไปทางด้านเครื่องรับ ซึ่งมีหลักการทำงาน โดยภาค DATA_BUFFER จะรับสัญญาณมาจากตัวแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลมาเก็บไว้และนำสัญญาณจาก DATA_BUFFER จำนวน 5 อีลีเมนต์มาลบกับสัญญาณจากโค้ดบุ๊คจำนวน 5 อีลีเมนต์ ซึ่งเก็บอยู่ใน CB_BUFFER เป็นการลบแบบไม่คิดเครื่องหมาย จากนั้นนำผลลัพธ์ที่ได้จากการลบจำนวน 5 อีลีเมนต์มาบวกกันโดยภาค ADDER ผลลัพธ์ที่ได้จากการบวกนำไปเปรียบเทียบกับค่าที่เก็บไว้ในรีจิสเตอร์ ซึ่งในครั้งแรกค่าในรีจิสเตอร์จะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

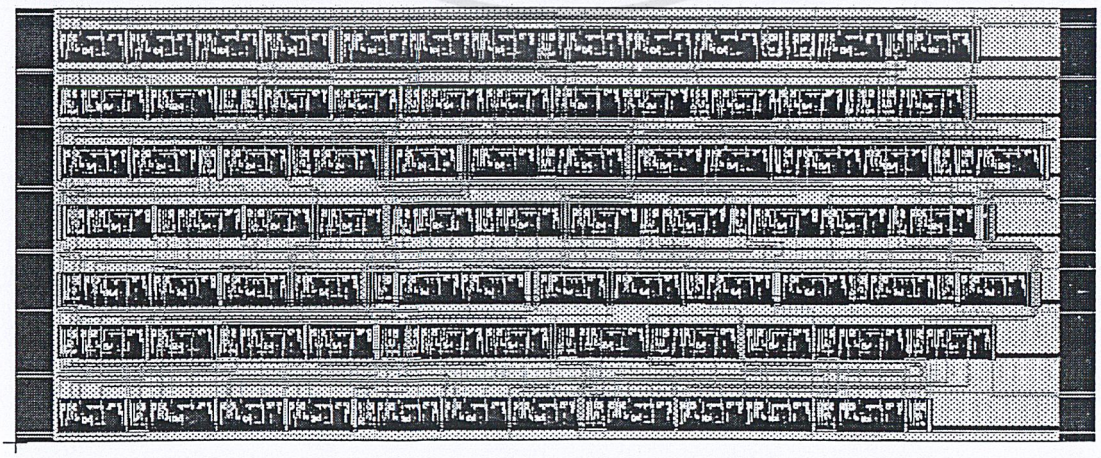
กำหนดเป็นค่าสูงสุด หากผลลัพธ์ที่ได้จากการบวกมีค่าน้อยกว่าค่าที่เก็บอยู่ในรีจิสเตอร์ ภาค COMPARATOR จะส่งค่านี้ไปเก็บไว้แทนค่าเดิม แต่ถ้าค่าที่ได้จาก ภาค ADDER มีค่ามากกว่าค่าในรีจิสเตอร์ รีจิสเตอร์ก็จะเก็บค่าเดิมไว้ และทุกครั้งที่ผลลัพธ์จากการเปรียบเทียบมีค่าน้อยกว่าค่าเดิมในรีจิสเตอร์ ตัวเปรียบเทียบก็จะส่งสัญญาณให้ภาค ADDRESS & INDEX จำตำแหน่งของค่านั้นไว้

ตัวประมวลผลสำหรับค้นหาข้อมูลในโค้ดบุ๊ค มีจำนวนรอบในการทำงานดังกล่าวข้างต้น จำนวน 64 ครั้งต่อสัญญาณอินพุตหนึ่งชุด ซึ่งมีรอบการทำงานเท่ากับจำนวนเวกเตอร์ของตัวโค้ดบุ๊ค และเมื่อทำงานครบ 64 ครั้ง ก็จะส่งตำแหน่งของข้อมูลที่มีค่าใกล้เคียงกับสัญญาณอินพุตมากที่สุดไปยังภาค SERIAL_OUT และภาค SERIAL_OUT จะส่งสัญญาณดังกล่าวแบบอนุกรมไปยังเครื่องรับต่อไป ซึ่งการทำงานทั้งหมดจะถูกควบคุมโดยภาค CONTROL

จากในรูปที่ 3.2 ได้แสดงส่วนของวงจรรวมทั้งหมดซึ่งประกอบไปด้วย DATA_BUFFER CB_BUFFER , SUB_ABS, ADDER, COMPARATOR, ADDRESS &INDEX, CONTROL, SERIAL และในแต่ละส่วนนั้นจะมีส่วนประกอบย่อย ๆต่าง ๆดังนี้

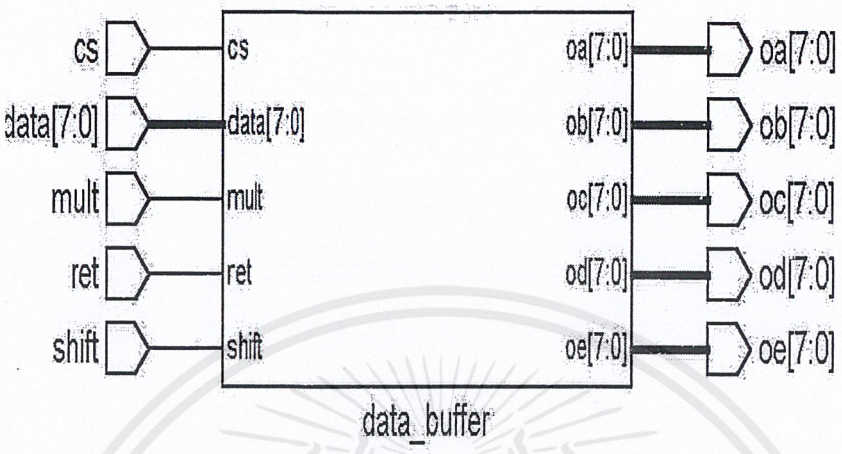
3.2.1 ส่วนประกอบ DATA_BUFFER

เป็นตัวเก็บสัญญาณอินพุตขนาด 5 ไบต์จำนวน 2 ชุด มีการเลื่อนข้อมูลเข้าครั้งละ 1 ไบต์โดยขา SHIFT ความถี่ 8 กิโลเฮิร์ต ในหนึ่งชุดจะทำการเลื่อนข้อมูลเข้าจำนวน 5 ไบต์เมื่อข้อมูลชุดแรกเข้ามาครบแล้ว สัญญาณจากขา CS จะควบคุมให้ตัวเก็บข้อมูลอีกชุดหนึ่งเลื่อนข้อมูลเข้ามาเก็บจนครบ 5 ไบต์เช่นกันและขาสัญญาณ MULTI จะเลือกข้อมูลชุดแรกออกไปทำการลบที่ภาค SUB_ABS ซึ่งจะสลับกันทำงานเช่นนี้เรื่อยๆ แสดง Layout ได้ ดังรูปที่ 3.3 ที่ได้จากการทำ Place&Route



(ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



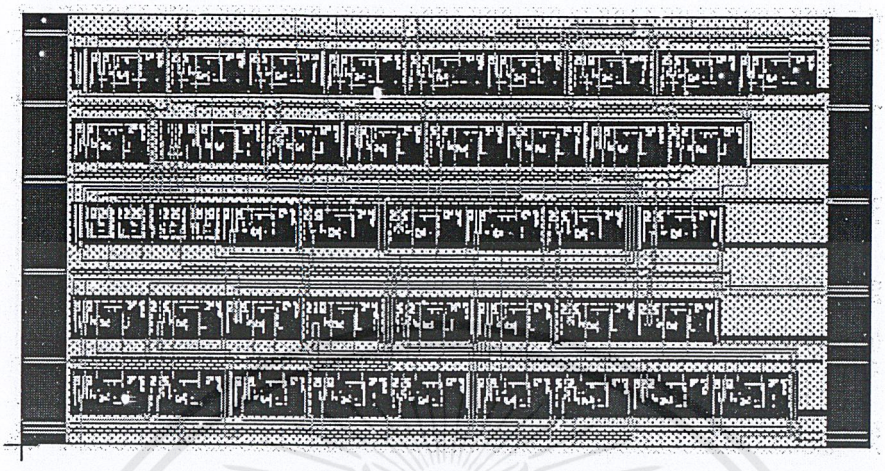
(ข)

รูปที่ 3.3 (ก) Standard Cell DATA_BUFFER (ข) ผังงานของ DATA_BUFFER

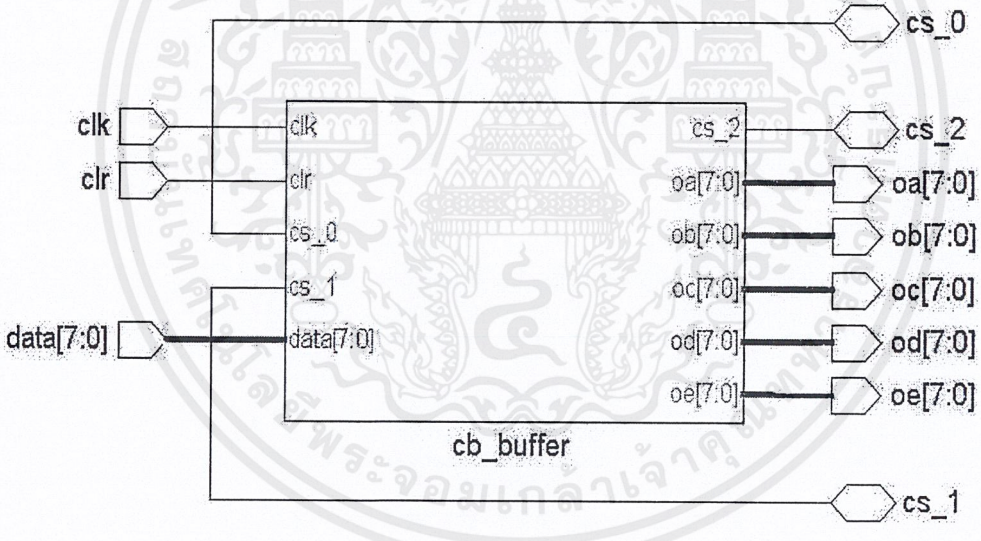
3.2.2 ส่วนประกอบของภาค CB_BUFFER

เป็นตัวรับสัญญาณจากโค้ดบุ๊ค มีขา DATA และขา Chip Select (CS) เป็นตัวกำหนดการทำงาน โดยที่ข้อมูลจะส่งมาที่ BUFFER ทุกตัวพร้อมกัน มี BUFFER อยู่ทั้งหมด 5 ตัว ซึ่งที่ตัว BUFFER จะกำหนดค่าไว้ที่ขา WR มีค่า 00, 01, 02, 03 และ 04 ตามลำดับ การทำงานจะขึ้นอยู่กับค่าของขา CS ถ้าขา CS มีค่า 02 ก็แสดงว่า BUFFER ตัวที่สามที่ถูกกำหนดค่าไว้ที่ 02 จะเป็นตัวเก็บข้อมูลไว้ ซึ่งที่ขา CS จะมีค่า 00, 01, 02, 03 และ 04 เช่นกัน ซึ่งค่าของ CS จะถูกกำหนดโดย CONTROL อีกทีหนึ่ง ฉะนั้น BUFFER จะรับข้อมูลจาก DATA ได้ก็ต่อเมื่อ ค่า WR ที่กำหนดไว้มีค่าเท่ากับค่าของ CS และ BUFFER จะสลับกันทำงานก็ต่อเมื่อ ค่าที่ขา CS เปลี่ยนไปซึ่งจะทำงานพร้อมกับการดึงข้อมูลจาก BUFFER ไปใช้งานด้วย ซึ่งจะมีขา CLK ต่อให้กับ BUFFER ทุกตัว ส่วนขา CLR จะทำงาน ก็ต่อเมื่อมีการส่งข้อมูลครบ 64 ครั้ง ก็จะเคลียร์ BUFFER ทุกตัวให้ว่างเพื่อเริ่มต้นรับค่าใหม่ดังรูปที่ 3.4 ได้แสดงถึง Standard Cell และผังงานของ CB_BUFFER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



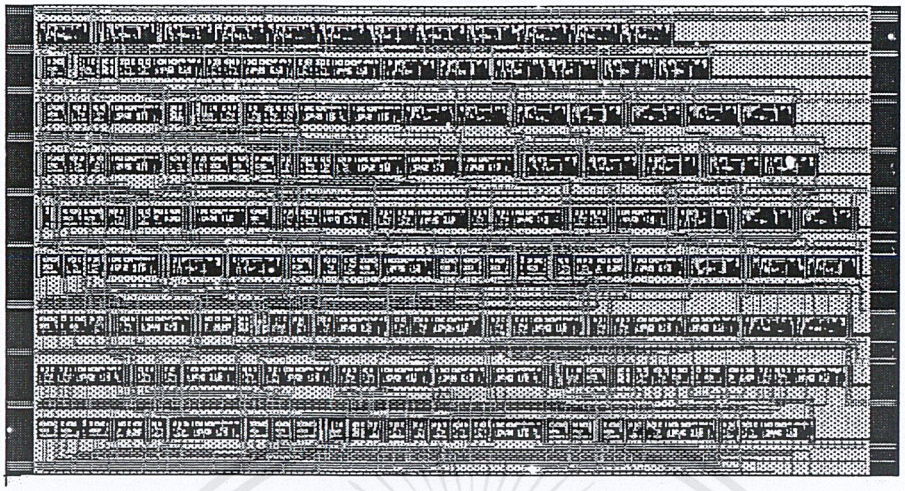
(ข)

รูปที่ 3.4 (ก) Standard Cell CB_BUFFER (ข) ผังงานของ CB_BUFFER

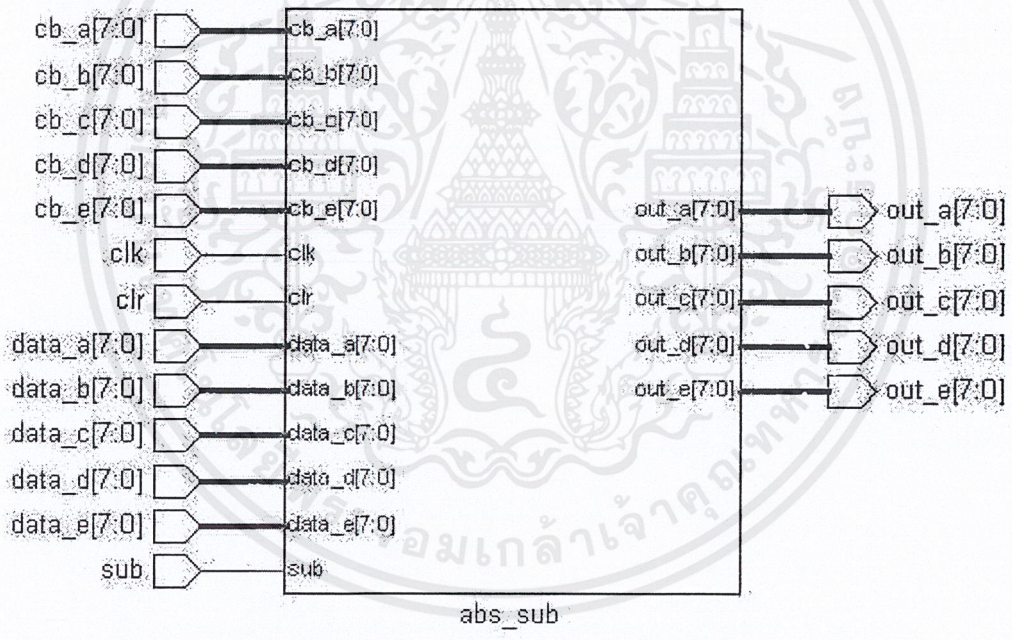
3.2.3 ส่วนประกอบของ SUB_ABS

ภาค SUB_ABS ทำหน้าที่ในการลบโดยทำการลบสัญญาณที่มาจากอินพุตกับสัญญาณจากโค้ดบุ๊คโดยไม่มีการคิดเครื่องหมาย ซึ่งจะมีสัญญาณอยู่ 5 อีลีเมนต์โดยวงจร SUB_ABS จะทำงานเมื่อขา SUB มีสถานะเป็น High ดังรูปในที่ 3.5 จะแสดงให้เห็นถึงผังงานและ Standard Cell

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



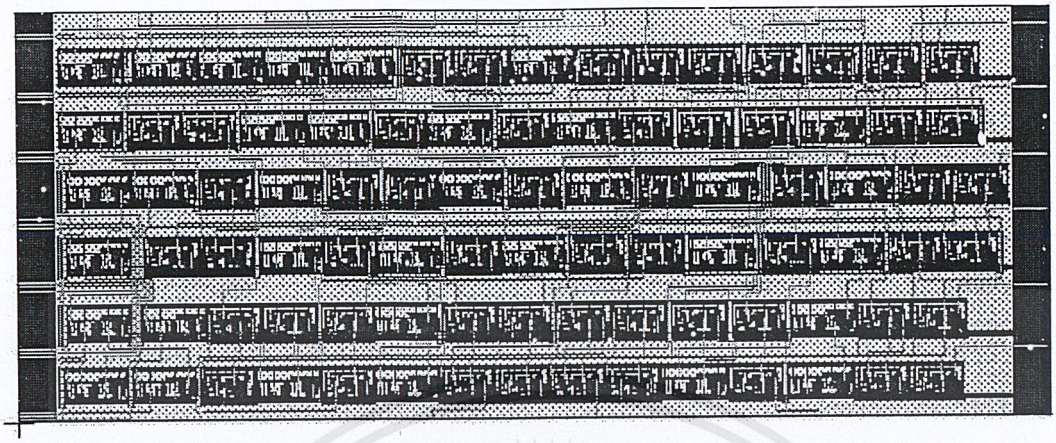
(ข)

รูปที่ 3.5 (ก) Standard Cell ASB_SUB (ข) ผังงานของ ABS_SUB

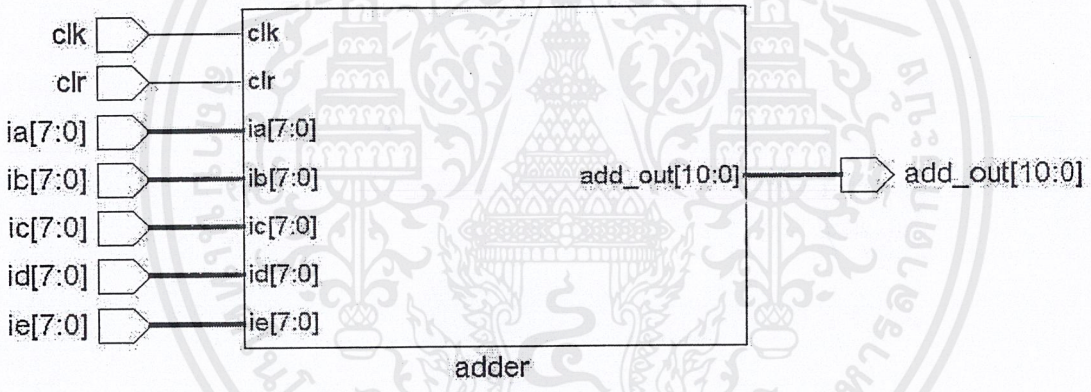
3.2.4 ส่วนประกอบของ ภาค ADDER

ในรูปที่ 3.6 เป็นภาค ADDER เป็นภาคที่ทำหน้าที่ในการบวกข้อมูลที่ได้จากภาค SUB_ABS โดยมีข้อมูล 5 อีลีเมนต์มาป้อนเข้าวงจรบวก โดยแต่ละอีลีเมนต์ให้ชื่อว่า IA, IB, IC, ID และ IE ซึ่งเป็นข้อมูลขนาด 8 บิต การทำงานจะให้ข้อมูลบวกกัน 2 ชุดคือ IE บวกกับข้อมูลจาก ID และ IC บวกกับ IB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

รูปที่ 3.6 (ก) Standard Cell ADDER (ข) ผังงานของ ADDER

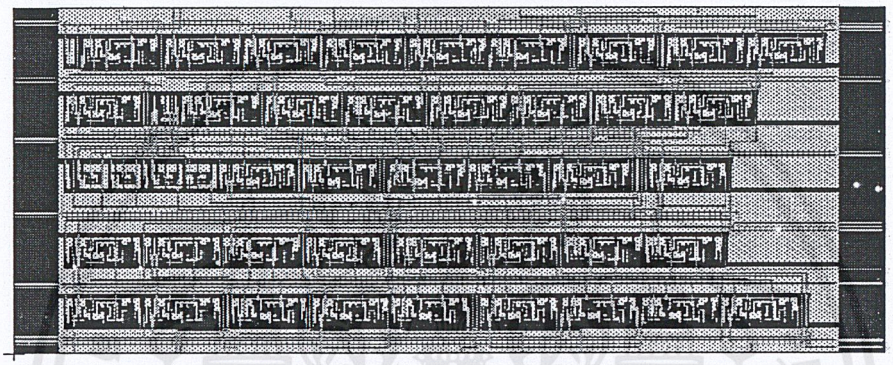
ซึ่งจะได้ผลลัพธ์ที่มีขนาด 9 บิต ซึ่งเป็นการเผื่อบิตตัวทดไว้ จากนั้นนำผลลัพธ์ที่ได้ทั้งสองค่ามาบวกกันต่อไป ซึ่งจะได้ผลลัพธ์อีกค่าหนึ่งซึ่งมีขนาด 10 บิตเป็นการเผื่อบิตตัวทดไว้เช่นกัน จากนั้น ผลลัพธ์ที่ได้นำไปบวกกับข้อมูลจาก IA ซึ่งจากรูปที่ 3.6 ภายในบล็อก IA จะต่อผ่าน BUFFER 2 ตัวด้วยกันผลลัพธ์ที่ได้จากการบวกจะกำหนดให้มีขนาด 11 บิตซึ่งเป็นการเผื่อบิตตัวทดไว้ ผลลัพธ์ที่ ได้จากภาค ADDER จะส่งต่อไปยังภาค COMPARATOR ต่อไป

3.2.5 ส่วนประกอบภาค COMPARATOR

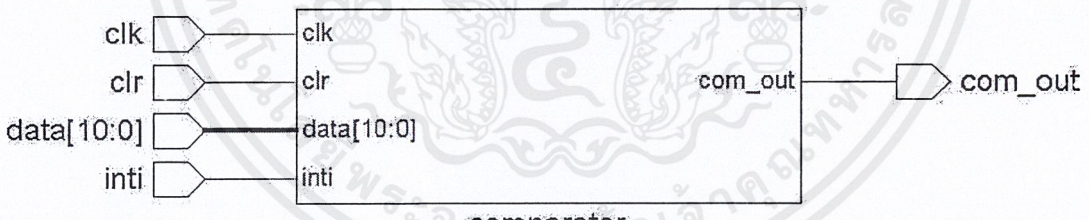
ในรูปที่ 3.7 เป็นภาคที่ทำหน้าที่เปรียบเทียบค่าที่ได้จากภาค ADDER แต่ละเวกเตอร์กับค่าที่เก็บไว้ใน รีจิสเตอร์ การทำงานจะควบคุมด้วยภาค CONTROL ซึ่งจะกำหนดค่าแรก ให้กับรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในภาค COMPARATOR ให้เป็นค่าสูงสุด เมื่อภาคเปรียบเทียบได้รับค่ามาจากวงจรบวกก็นำค่าที่ได้นี้ไปเปรียบเทียบกับค่าสูงสุดที่ถูกกำหนดไว้ คือ "7FF" ถ้าค่าที่ได้จากวงจรบวกมีค่าน้อยกว่าค่าในวงจรเปรียบเทียบ วงจรเปรียบเทียบก็จะนำค่าจากวงจรบวกไปเก็บไว้แทนที่ค่าเดิม แต่ถ้าค่าจากวงจรบวกมีค่ามากกว่า วงจรเปรียบเทียบก็จะเก็บค่าเดิมไว้ ซึ่งในการทำงานทั้งหมดจะเปรียบเทียบกัน 64 ครั้งต่อสัญญาณอินพุตหนึ่งสัญญาณ ค่าที่ได้ก็จะมีค่าใกล้เคียงกับสัญญาณจากอินพุตมากที่สุด



(ก)



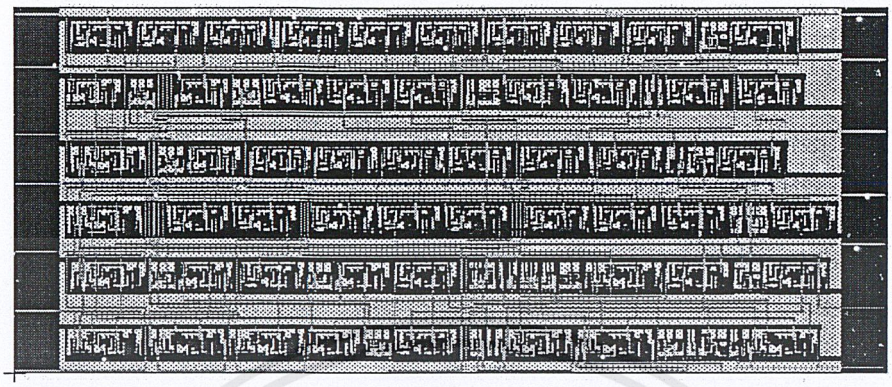
(ข)

รูปที่ 3.7 (ก) Standard Cell COMPARATOR (ข) ผังงานของ COMPARATOR

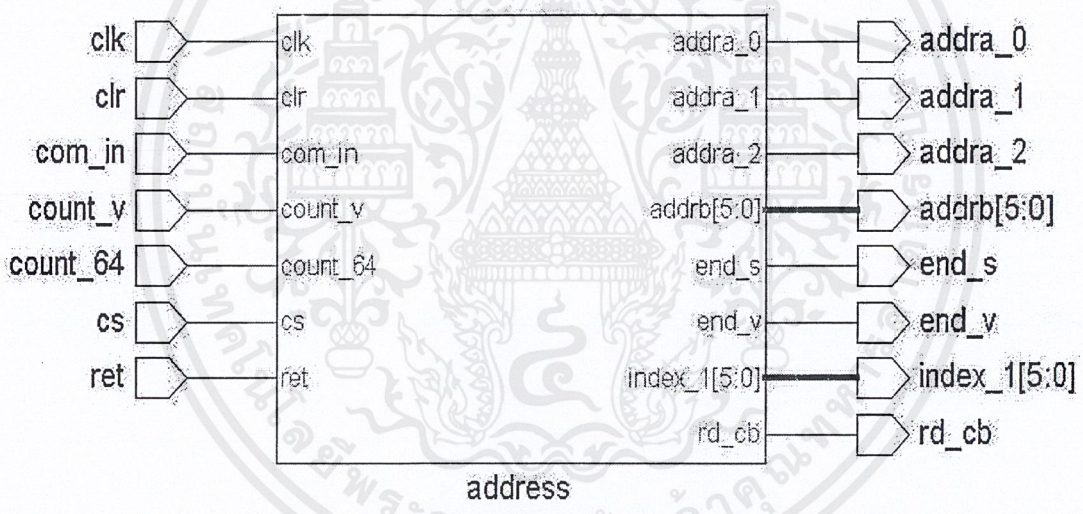
3.2.6 ภาค ADDRESS & INDEX

ในรูปที่3.8เป็นภาคที่ทำหน้าที่ส่งตำแหน่งของข้อมูลไปเปิดโค้ดบุ๊ค และส่งสัญญาณค่าดัชนีไปยังภาคต่อไป ซึ่งจะมีวงจรนับอยู่ 2 ชุด คือวงจรนับ 64 กับวงจรนับ 5 ซึ่งวงจรนับ 64 จะนับจำนวนเวกเตอร์ของข้อมูลในโค้ดบุ๊คโดยควบคุมจากภาค CONTROL เมื่อมีสัญญาณพัลส์เข้ามาลูกแรกวงจรนับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

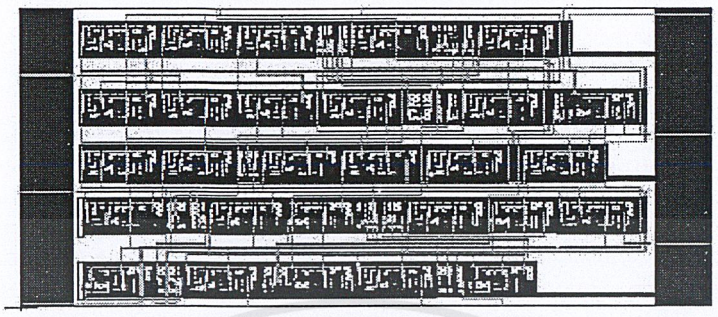
รูปที่ 3.8 (ก) Standard Cell ADDRESS&INDEX (ข) ผังงานของ ADDRESS&INDEX

จะเพิ่มค่าครั้งละ 1 และมี BUFFER 4 ตัวทำหน้าที่หน่วงเวลาเพื่อให้เวลาในการทำงานของวงจรอื่นๆ ทำงานกันได้ทันเวลาพอดี และมีสัญญาณมาจากภาค COMPARATOR ควบคุมการจดจำค่าดัชนี และจะส่งค่าดัชนีออกไปเมื่อขา CS อยู่ในสถานะเป็น High

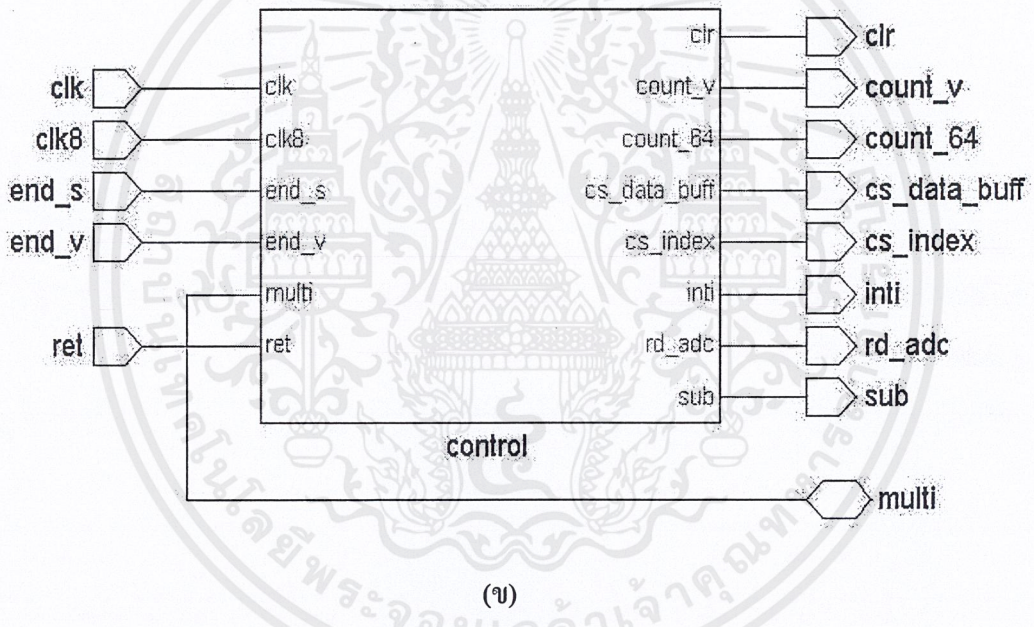
3.2.7 ส่วนประกอบภาค CONTROL

ในรูปที่ 3.9 นั้นเป็นภาคที่ทำหน้าที่ควบคุมเวลาการทำงานของภาคอื่นๆ โดยสัญญาณที่ ป้อนให้ภาค CONTROL จะมีสัญญาณนาฬิกา 2 สัญญาณคือ ความถี่ 5.5 เมกะเฮิร์ต และความถี่ 8 กิโลเฮิร์ตและรับสัญญาณจากภาค ADDRESS&INDEX ซึ่งเป็นสัญญาณสิ้นสุดการทำงานแต่ละ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก)



(ข)

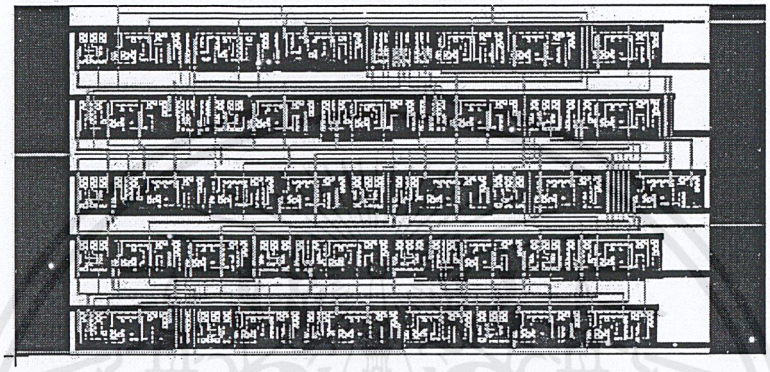
รูปที่ 3.9 (ก) Standard Cell CONTROL (ข) ผังงานของ CONTROL

เวกเตอร์ของสัญญาณอินพุต เมื่อสัญญาณ RET เป็น "1" ภาค CONTROL จะส่งสัญญาณไปควบคุมการทำงานของภาคต่างๆ และเมื่อภาค CONTROL ได้รับสัญญาณสิ้นสุดการทำงานจากขา END_S ภาค CONTROL จะส่งสัญญาณไปให้ภาค SUB_ABS และภาค ADDER หยุดการทำงาน และนับเวลาจากเริ่มการทำงานจนครบ 625 ไมโครวินาที ภาค CONTROL จะส่งสัญญาณให้ภาค SUB_ABS และภาค ADDER เริ่มการทำงานอีกครั้งหนึ่ง

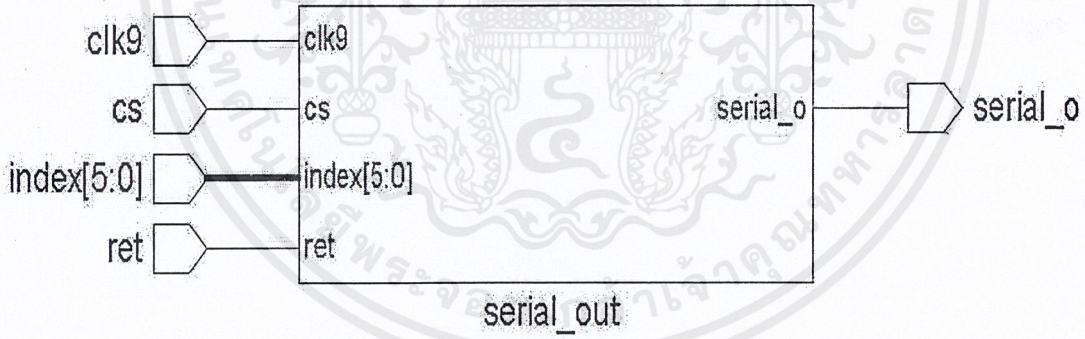
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.8 ส่วนประกอบภาค SERIAL_OUT

เป็นภาคที่ส่งตำแหน่งของข้อมูลไปยังเครื่องรับ มีขาที่ใช้งานคือ ขา CLK 9 กิโลเฮิร์ต ซึ่งจะจ่ายให้กับอุปกรณ์ในภาคนี้ ส่วนขา CS จะใช้ควบคุมการทำงานของแต่ละบล็อกโดยส่งมาจากภาค CONTROL



(ก)



(ข)

รูปที่ 3.10 (ก) Standard Cell SERIAL_OUT (ข) ผังงานของ SERIAL_OUT

ส่วน INDEX จะถูกส่งมารออยู่ที่บล็อก SE3 และ SE4 ถ้าขา CS มีค่าเป็น '1' INDEX จะถูกนำเข้าไปเก็บใน SE3 และเมื่อ CLK เข้ามาอีกลูกหนึ่ง SERIAL SE5 จะนำข้อมูลเข้าไปเพื่อส่งไปยังเครื่องรับต่อไป แต่ถ้า CS มีค่าเป็น '0' ก็จะสลับการทำงานเป็น SE4 รับข้อมูลเข้าไป และเมื่อสัญญาณ CLK เข้ามาอีกลูกหนึ่ง SERIAL SE6 ก็จะรับข้อมูลเพื่อส่งไปยังเครื่องรับต่อไป การทำงานจะสลับกันไปตามสัญญาณจาก CLK ที่เข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การสร้าง Padframe

ก่อนการสร้าง Padframe จำเป็นต้องทราบชนิดและวิธีการใช้งานของ Pad ต่าง ๆ ของโรงงานก่อนประกอบด้วย Pad ต่าง ๆ ดังนี้

1. Pad ที่จะใช้กับชิพต้องการแรงดันคงที่ ในบางกรณีที่ชิพที่ต้องการแรงดันที่มีความเที่ยงตรงสูง (Noise ต่ำ) เช่น วงจรอนาล็อกหรือชิพบางตัวที่ไวต่อการเปลี่ยนแปลง

1) VDDCO ใช้เฉพาะจ่ายแรงดันเข้าไปภายใน

2) VSSCO ใช้เฉพาะเป็นกราวด์ของวงจรภายใน

2. Pad ชนิดพิเศษ ได้แก่ Pad ดังต่อไปนี้

1) VDD จะรวมไฟเลี้ยงทั้ง 3 วงจรรวมกัน

2) VSS จะรวมกราวด์ทั้ง 3 วงจรรวมกัน รวมไปถึงกราวด์ Core และ Substrate Biasing

3) VDDI1 ต่อมาจากไฟเลี้ยงภายในวงที่มีขนาดกว้าง

4) VSSI1 ต่อมาจากกราวด์ภายในวงที่มีขนาดกว้าง

5) VDDI2 ต่อมาจากไฟเลี้ยงภายในวงที่มีขนาดแคบ

6) VSSI2 ต่อมาจากกราวด์ภายในวงที่มีขนาดแคบ

3.3.1 ขั้นตอนการสร้าง Padframe

เมื่อได้ Floorplan ของชิพก็จะทำการเลือก Pad ที่จะใช้งาน โดยเลือก Pad Output ทั้งหมดที่สามารถจ่ายกระแสได้ 8 mA เพื่อง่ายต่อการใช้งาน รายละเอียดของ Pad ที่เลือกใช้มี ดังนี้

1. B8CR เป็น Pad Output แบบ C3MOS

2. BD8C เป็น Pad Input/Output แบบ CMOS

3. IBUF เป็น Pad Input

4. CORNER เป็น Pad ที่ใช้ต่อมุมของ Pad Frame

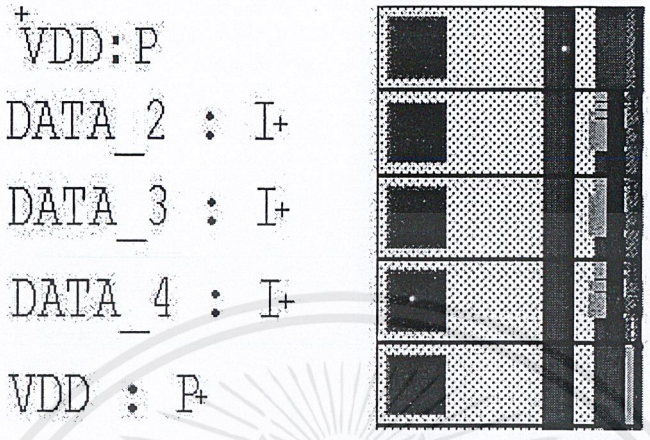
5. VDD เป็น Pad ที่ใช้ต่อกับ Vdd ภายนอกเลี้ยงวงจรทั้งหมด

6. VSS เป็น Pad ที่ใช้ต่อกับ Ground ภายนอก

3.3.2 การตั้งชื่อ Label ของ Pad

เมื่อเลือก Pad และสร้าง Pad Frame เสร็จเรียบร้อยแล้ว ให้ตั้งชื่อ Pad เพื่อให้ทางโปรแกรม ERC ของโรงงานสามารถทดสอบได้ โดยสร้าง Label ด้วย Port บน Layer GDS II โดยที่ Pad Input หรือ Output ให้ตั้งชื่อโดยมี ":I" ต่อท้ายเช่น PAD8 : I เป็นต้น สำหรับ Pad ของแหล่งจ่ายไฟทุกชนิดให้ตั้งชื่อโดยมี ":P" ต่อท้าย เช่น VDDB2:P เป็นต้น สำหรับ Pad ของ Ground ทุกชนิดให้ตั้งชื่อโดยมี ":G" ต่อท้ายเช่น VSSDP : G เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 การตั้งชื่อ PAD

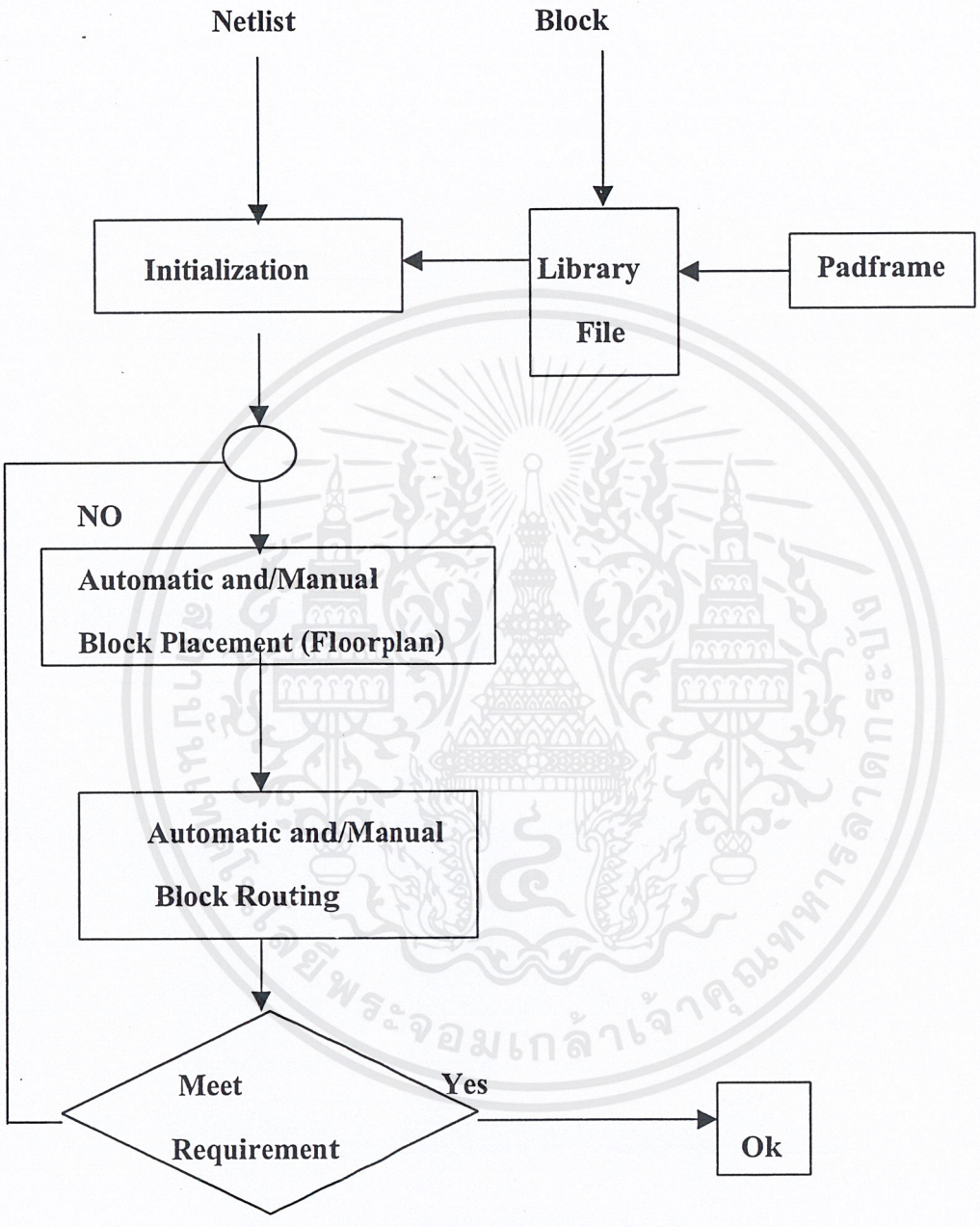
3.4 การทำ Block Place & Route (BPR)

L-Edit สามารถทำ Block Place & Route ได้โดยมีขั้นตอนดังรูป ซึ่งแบ่งได้เป็น 3 ขั้นตอนหลัก ๆ คือแผนผังขั้นตอนการทำงานนั้นจะแสดงในรูปที่ 3.12

1. Initialization เมื่อกำหนดพารามิเตอร์เรียบร้อยแล้ว โปรแกรม L-Edit จะอ่านNetlist เพื่อทำการ Initialize ข้อมูลเพื่อทำ BPR จะได้บล็อกต่าง ๆดังรูปที่ 3.13 ซึ่ง BLOCK ต่างๆ จะถูกจัดวางที่ขอบด้านล่างของ Pad Frame และมี Routing Guide ปรากฏขึ้นเพื่อแสดงการเชื่อมต่อของสายสัญญาณภายในชิปเพื่อใช้เป็นแนวทางในการปรับแต่บล็อกเพื่อใช้เชื่อมต่อสัญญาณอย่างเหมาะสม

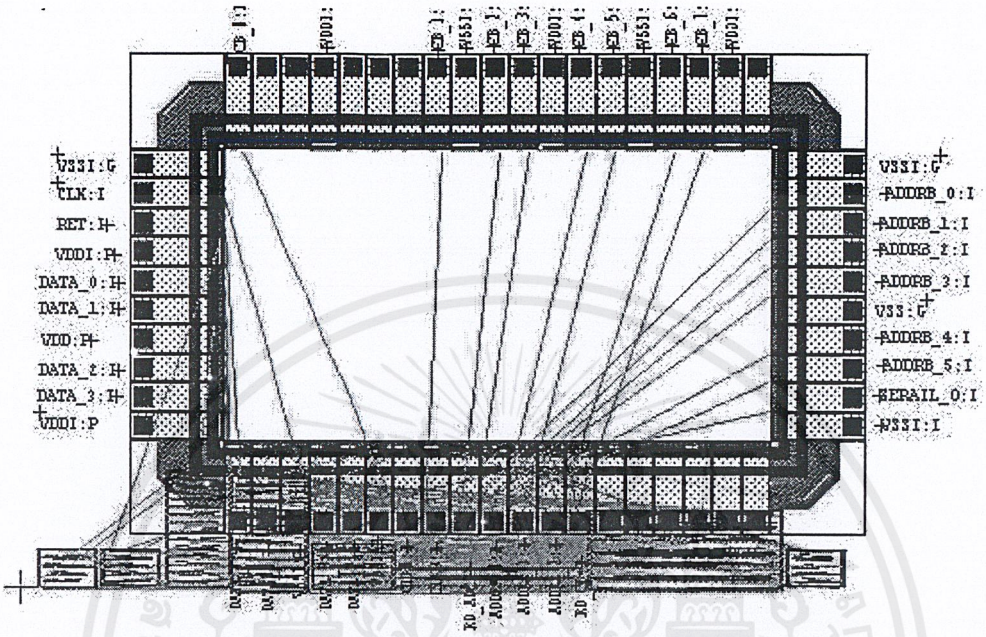
2. Block Placement (Floorplan) คือการวางบล็อกต่างๆ ให้อยู่ในกรอบภายใน Pad Frame ปรับด้านและตำแหน่งของบล็อกต่าง ๆ โดยคำนึงถึงพื้นที่ การเชื่อมต่อสัญญาณระหว่างบล็อกกับ Pad และปัจจัยที่กำหนดเพื่อให้ได้ Floorplan ที่เหมาะสมการทำ Block Placement หรือ Floorplan เกี่ยวข้องกับขั้นตอน SPR ของบล็อกต่าง ๆ เนื่องจากถ้าไม่สามารถสร้าง Floorplan ที่เหมาะสมจากขนาดของบล็อกที่มีอยู่ได้ ก็ต้องกลับไปทำขั้นตอน SPR เพื่อปรับขนาดของบล็อกใหม่ ซึ่งจำนวนการทำซ้ำในแต่ละขั้นตอนนี้จะขึ้นอยู่กับเงื่อนไขของพื้นที่ของชิพว่าถูกจำกัดไว้ที่ขนาดเท่าไรและอาจต้องย้อนกลับไปถึงขั้นตอนการออกแบบวงจรถ้าไม่สามารถปรับให้เป็นไปตามต้องการแสดงดังรูปที่ 3.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 ขั้นตอนการทำ Block Place&Route

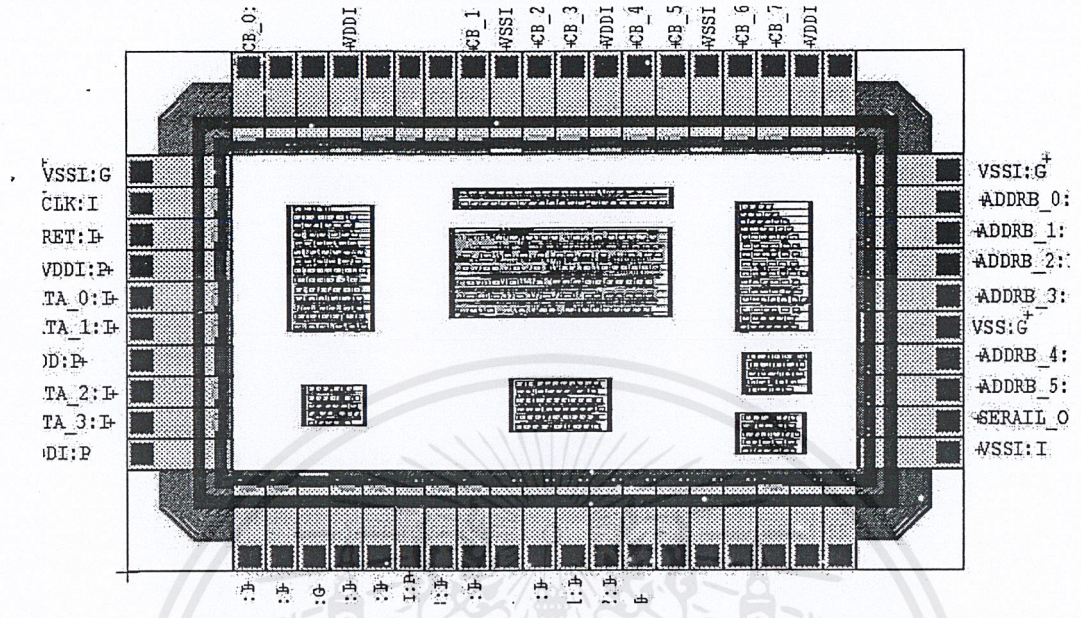
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 ผลของการ Initialization ในขั้นตอนทำ BPR

3. Block Routing หลังจากการ Floorplanจะเป็นขั้นตอนเชื่อมต่อสัญญาณต่างๆเข้าด้วยกันโดยพิจารณาจากประเภทและคุณสมบัติของสัญญาณแต่ละเส้นเพื่อใช้เป็นข้อมูลในการกำหนดชนิดของเลเยอร์และขนาดของสัญญาณเนื่องคุณสมบัติทางไฟฟ้าที่แตกต่างกันเช่น ค่าของ capacitance และ resistance ที่มีผลกับDelay ของสัญญาณทุกเส้น,Maximum Current Density ซึ่งเป็นปัจจัยสำคัญที่กำหนดขนาดของสัญญาณ Power เป็นต้นในการเชื่อมต่อสัญญาณ (Routing) สามารถทำได้ทั้งแบบอัตโนมัติโดยใช้ Automatic Routing ซึ่งใช้กับสัญญาณที่ไม่มีเงื่อนไขบังคับมากนักและทำเองโดยใช้ Assist Manual Route กับสัญญาณที่สำคัญเช่น Clock, Bus และPowerขั้นตอน Block Routing จะมีความสำคัญมากกับชิพที่มีเงื่อนไขของเวลามาบังคับมากๆเช่น ชิพที่มีความเร็วในการทำงานสูงและจะยิ่งสำคัญมากขึ้นเมื่อใช้เทคโนโลยี การผลิตที่เล็กลงซึ่งการทำซ้ำในขั้นตอน SPR อาจจะต้องทำรวมมาถึงขั้นตอนนี้ด้วยถ้าไม่สามารถเป็นไปตามเงื่อนไขของเวลาได้แต่เนื่องจากชิพตัวเข้ารหัสและถอดรหัสตัวนี้ไม่มีเงื่อนไขนี้เพื่อต้องการความถูกต้องเป็นสำคัญในส่วนนี้จะแสดงในรูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 การทำ Floorplanของวงจรเข้ารหัสถอดรหัสสัญญาณเสียง

4. การ Extract Layout ของชิพตัวเข้ารหัสและถอดรหัสสัญญาณเสียงเพื่อตรวจสอบหลังการทำ Block Routing ขั้นตอนนี้จะพิจารณา Block และ Pad ต่าง ๆ เป็น subcircuit เพื่อให้สามารถ Extract Block และ Pad ทั้งหมดออกมาเป็น subcircuit ที่มีการเรียงลำดับ port จากน้อยไปมากได้

5. การตรวจสอบความถูกต้องของ Layout หลังการทำ Block Routing เนื่องจาก port ของ subcircuit ใน netlist ที่ได้จากการ Extract มีการเรียงลำดับจากน้อยไปมาก จึงต้องแก้ไขใน VHDL netlist ให้เรียงลำดับเช่นเดียวกัน และทำการตรวจสอบดังนี้

1) สร้าง element description file เพื่อใช้อธิบายลักษณะของ Block และPad แต่ละแบบ

เช่น Pad output => || d padout a z AddP GndP

Block Data_path => || d data_path acc_sim_0 acc_sim_1 acc_sim_2……

2) แปลง VHDL netlist ให้เป็น SPICE netlist

3) save เก็บไว้เพื่อใช้เป็น netlist จากschematic

4) ใช้โปรแกรม L-Edit/Extract layout เพื่อให้ได้ netlist จาก Layout

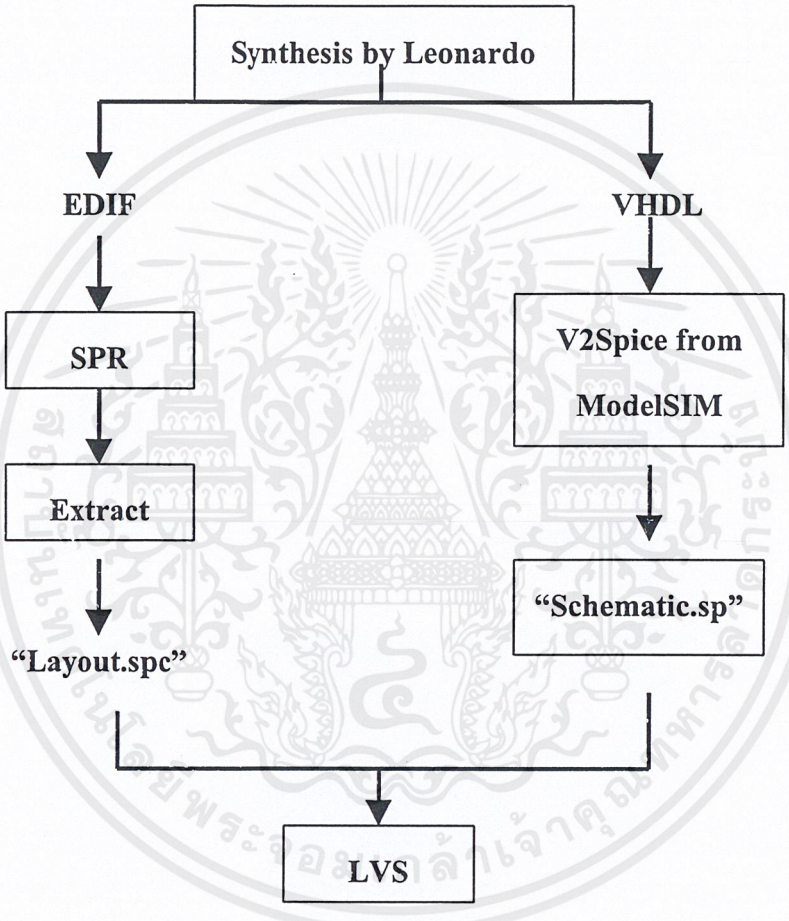
5) สร้าง LVS setup file

6)ใช้โปรแกรม LVS เพื่อเปรียบเทียบความเท่ากันของ netlist ระหว่าง Layout และ schematic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

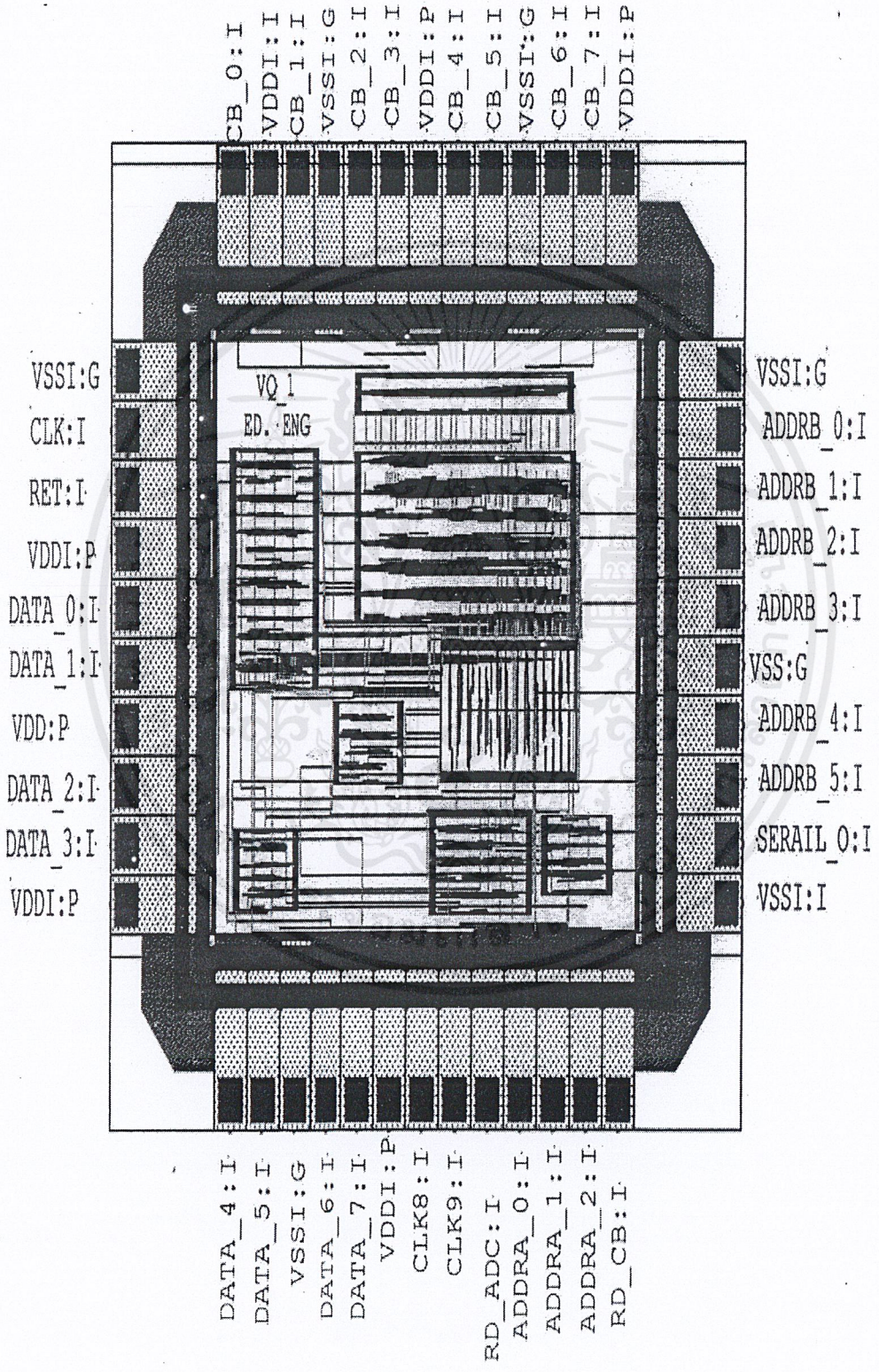
7) Run โปรแกรม LVS

ถ้า Netlist ทั้งสองเท่ากันจะรายงานว่า “Circuit are equal” นั้นหมายถึงเลเอาต์ที่ออกแบบเป็นวงจรเดียวกับวงจรที่ออกแบบไว้และในรูปที่ 3.15 นั้นเป็นแผนผังขั้นตอนในการตรวจสอบความถูกต้อง ดูจากแผนผังแล้วจะพบว่าจะใช้โปรแกรมในการตรวจสอบคือ LVS



รูปที่ 3.15 แผนผังในการตรวจสอบความถูกต้องของ Layout

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 ผลของการทำขั้นตอน Block Route

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจากในรูปที่ 3.16 สามารถที่จะสรุปได้ว่าบล็อกแต่ละบล็อกทั้งหมดนั้นมีขนาดของความกว้างและขนาดของยาวเท่าใดและมี Standard Cell ที่ใช้ในแต่ละบล็อกเท่าใดและผลสรุปทั้งหมดนี้จะอยู่ ในตาราง BPR Chip ตารางที่ 3.1

ตารางที่ 3.1 BPR CHIP

BPR CHIP	Number Of standard Cell	Number Of Signal in netlist	Core Size in LU	Core Area (LU^2)	Length of Nets in Core	Generated Visa in Core
CB_BUFFER	49	115	995.7x 102.6	102158.82	23001.7 LU	368
DATA_BUFFER	124	188	408.95x 606	247823.7	55646.4 LU	1153
ABS_SUB	208	449	1008.65 x 430.4	434122.96	114480.7 LU	1385
ADDER	92	222	356.35 x 621.7	221542.795	31079.6 LU	694
COMPARATOR	52	91	320.25 x 204.7	65555.175	10893.7 LU	245
ADDRESS& INDEX	85	120	465.15 x 255.4	118543.91	19592.75 LU	369
CONTROL	46	72	304.9 x 202.2	61650.78	9631.5 LU	208
SERIAL_OUT	51	73	321.1 x 197.2	63320.92	11511.5 LU	259
Total CHIP	707	180	2940.00x 2490.00	7320600.00	-	555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 กล่าวนำ

ออกแบบและสร้างวงจรรวมเฉพาะกิจตัวเข้ารหัสและถอดรหัสข้อมูลเสียงโดยใช้โปรแกรม LEDIT Tanner tool ได้สร้างขึ้นโดยอาศัยหลักการควอนไตซ์เวกเตอร์ เพื่อใช้ในการเข้ารหัสสัญญาณเสียง จากบทที่ 1-3 ได้กล่าวถึงแนวความคิด ทฤษฎี การออกแบบและการสร้างวงจรรวมเฉพาะกิจตัวเข้ารหัสและถอดรหัสข้อมูลเสียงที่ใช้วิธีควอนไตซ์เวคเตอร์ โดยใช้ภาษา VHDL และโปรแกรม L-Edit ดังรูปวงจรในบทที่ 3 นั้น ส่วนในบทนี้จะกล่าวถึงการทดสอบการทำงานและผลการทดลองของวงจรรวมเฉพาะกิจตัวเข้ารหัสและถอดรหัสข้อมูลเสียงที่ใช้วิธีควอนไตซ์เวคเตอร์ โดยได้ลักษณะผลการทดลองออกเป็น 2 ส่วนคือ 1. ในส่วนที่ใช้ VHDL เลียนแบบการทำงาน (Simulate) สัญญาณต่างๆ โดยใช้โปรแกรม Logic Simulator ของ Xilinx Foundation F1.5เป็นในส่วนของ Extract Layout

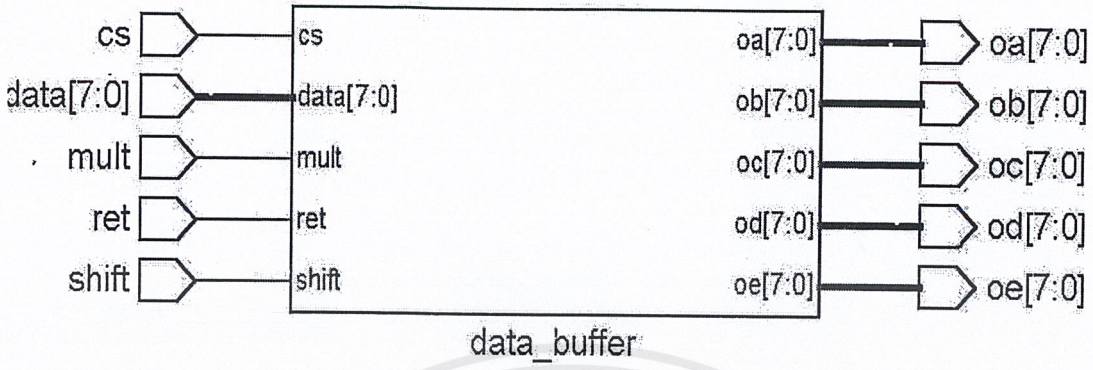
4.2 การจำลองการทำงานวงจรรวมเฉพาะกิจตัวเข้ารหัสและถอดรหัสข้อมูลเสียง

จากการออกแบบและสร้างวงจรรวมเฉพาะกิจตัวเข้ารหัสและถอดรหัสข้อมูลเสียงที่ใช้วิธีควอนไตซ์เวคเตอร์ ได้แบ่งการทำงานออกเป็นบล็อกต่างๆ ดังนี้ คือ DATA_BUFFER, B_BUFFER, SUB_ABS, ADDER, COMPARATOR, CONTROL, ADDRESS&INDEX และ SERIAL_OUT ซึ่งสัญญาณแต่ละตัวก็จะมีหน้าที่การทำงานและผลการทำงานที่แตกต่างกัน

4.2.1 ผลการจำลองการทำงานสัญญาณของภาค DATA_BUFFER

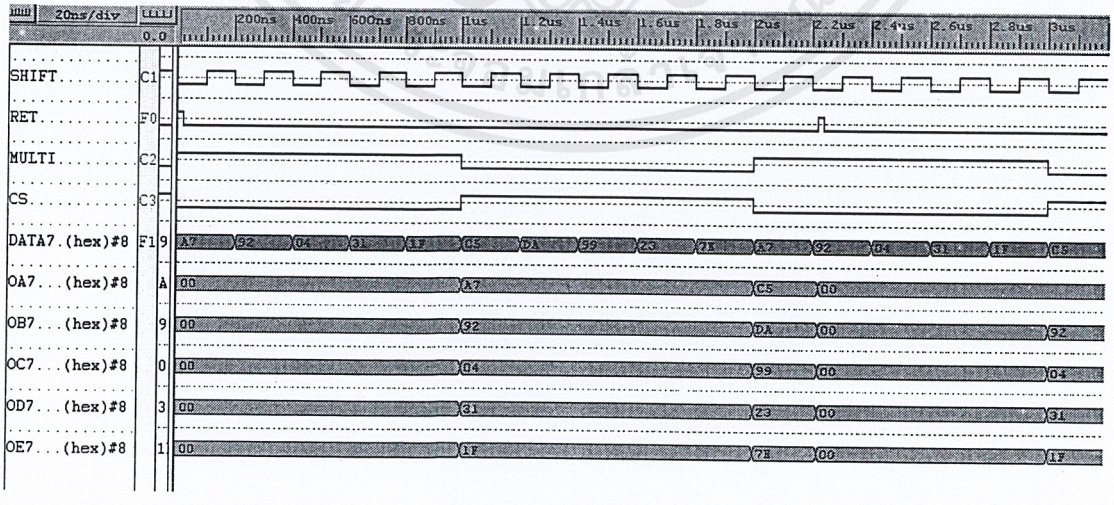
จากการทดลองเลียนแบบการทำงานของผังงาน DATA_BUFFER ในรูปที่ 4.1 โดยใช้โปรแกรม Logic Simulater ของ Xilinx Foundation F1.5 ได้ผลการทดลองดังรูปที่ 4.2 จะเห็นว่าเมื่อมีสัญญาณข้อมูลขนาด 8 บิต (เท่ากับ 1 อีลีเมนต์) เข้ามาที่ขา DATA[7:0] ข้อมูลก็จะถูกเลื่อน (shift) ไปเก็บไว้ยังตำแหน่งต่างๆ ของ buffer ชุดแรก โดยข้อมูลที่เข้ามาเป็นอีลีเมนต์แรก จะถูกนำไปเก็บไว้ที่รีจิสเตอร์ตำแหน่งล่างสุด คือ OA7 ส่วนข้อมูลในอีลีเมนต์ถัดไปก็จะถูกนำไปเก็บที่ตำแหน่ง OB7, OC7, OD7 และ OE7 ซึ่งเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 ผังงานของ DATA_BUFFER

รีจิสเตอร์ตำแหน่งสูงสุดตามลำดับ จนเมื่อมีข้อมูลเก็บใน buffer ชุดแรกจนครบ 5 อีลีเมนต์แล้ว สัญญาณ MULTI ก็จะเปลี่ยนสถานะจาก high เป็น low เพื่อทำการเลือกให้ข้อมูลที่อยู่ใน buffer ชุดแรกไปแสดงผลยังเอาต์พุต ในขณะเดียวกัน สัญญาณ CS ก็จะเปลี่ยนสถานะจาก low เป็น high เพื่อเลือกให้ buffer อีกชุดหนึ่งรับข้อมูลเวกเตอร์ถัดไปเข้ามาเก็บใน buffer เช่นเดียวกับ buffer ชุดแรก จนครบ 5 เวกเตอร์อีกครั้งหนึ่งแล้วจึงนำข้อมูลใน buffer นั้น ส่งผ่านไปยังเอาต์พุตพร้อมๆ กัน ซึ่งจะทำงานในลักษณะเช่นนี้เรื่อยๆ ไป แต่ถ้าสัญญาณ RET เป็น high เมื่อใด ก็จะทำการเคลียร์ค่าข้อมูลที่มีอยู่ใน buffer นั้น ให้เป็นศูนย์ จนกระทั่งสัญญาณ RET เปลี่ยนสถานะเป็น low ก็จะเริ่มทำการรับข้อมูลเข้ามาเก็บใน buffer อีกครั้งในคล็อกการทำงานถัดไป

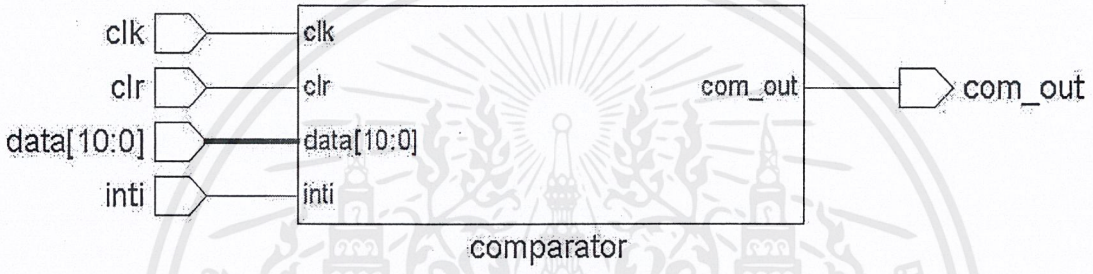


รูปที่ 4.2 ผลการเลียนแบบการทำงานของภาค DATA_BUFFER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ผลการจำลองการทำงานสัญญาณของภาค COMPARATOR

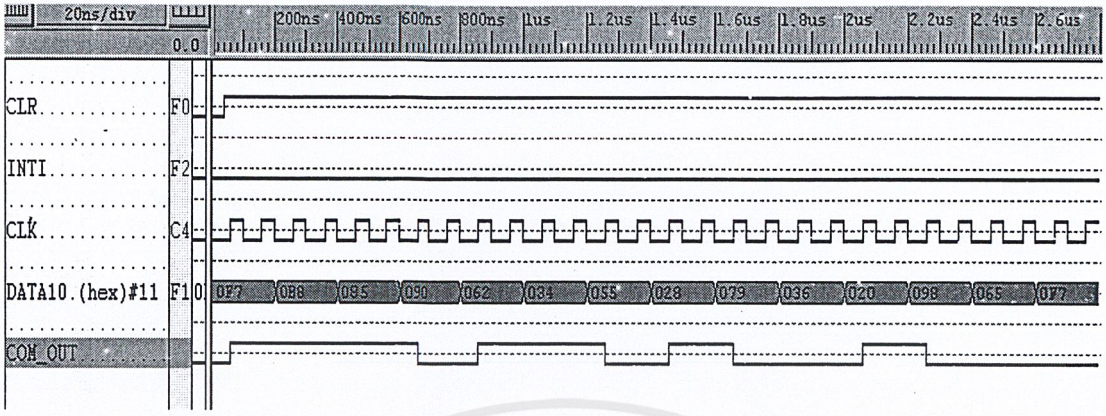
ในรูปที่ 4.3 เป็นผังงานที่ใช้ในการทดลองเลียนแบบการทำงานของสัญญาณภาค COMPARATOR โดยมีขา INTI, CLR, CLK และ DATA[10:0] เป็นสัญญาณอินพุต และมีขา COM_OUT เป็นสัญญาณเอาต์พุต ก่อนการทดลองเลียนแบบการทำงาน จะต้องต่อขาสัญญาณภายนอกให้กับภาค COMPARATOR แล้วจึงเลียนแบบการทำงานด้วยโปรแกรม Logic Simulater ซึ่งแสดงผลการทดลองเลียนแบบสัญญาณการทำงานได้ดังรูปที่ 4.4



รูปที่ 4.3 ผังงานของ COMPARATOR

เมื่อเริ่มต้นการทำงาน จะกำหนดให้สัญญาณ INTI เป็น high เพื่อกำหนดค่าเริ่มต้นภายในรีจิสเตอร์ให้เป็นค่าสูงสุด ซึ่งมีค่าเท่ากับ '7FF' เมื่อมีสัญญาณข้อมูลขนาด 10 บิต เข้ามาที่ขา DATA[10:0] ตัว COMPARATOR ก็จะนำสัญญาณข้อมูลที่เข้ามาไปเปรียบเทียบกับสัญญาณข้อมูลที่เก็บอยู่ภายในรีจิสเตอร์ โดยถ้าค่าข้อมูลที่รับเข้ามานั้น มีค่าน้อยกว่าหรือเท่ากับค่าที่เก็บอยู่เดิมในรีจิสเตอร์ตัว COMPARATOR ก็จะทำการเก็บค่าข้อมูลตัวนั้นไว้ในรีจิสเตอร์แทนค่าเดิมที่มีอยู่ เพื่อใช้สำหรับเปรียบเทียบกับสัญญาณข้อมูลที่จะเข้ามาเป็นอันดับต่อไป พร้อมกันนี้ ก็จะส่งสัญญาณ high ออกทางขาเอาต์พุต COM_OUT เพื่อบอกกับภาค ADDRESS & INDEX ว่าข้อมูลตำแหน่งนั้นมีค่าใกล้เคียงกับสัญญาณข้อมูลอินพุตที่รับเข้ามาจากอุปกรณ์แปลงสัญญาณแอนะลอกเป็นดิจิตอลพร้อมทั้งให้เก็บตำแหน่งนั้นไว้สำหรับการอ่านค่าดัชนีในโค้ดบุ๊คต่อไป แต่ถ้าข้อมูลที่เข้ามานั้นมีค่ามากกว่า ค่าในรีจิสเตอร์ก็จะไม่มีการเปลี่ยนแปลงจะเก็บค่าตำแหน่งเดิมไว้ แล้วส่งสัญญาณ low ออกทางขาเอาต์พุต COM_OUT บอกกับภาค ADDRESS & INDEX ว่าค่าตำแหน่งที่รับเข้ามานั้นไม่ได้ใกล้เคียงกับสัญญาณข้อมูลอินพุตที่รับมาจากอุปกรณ์แปลงสัญญาณแอนะลอกเป็นดิจิตอล

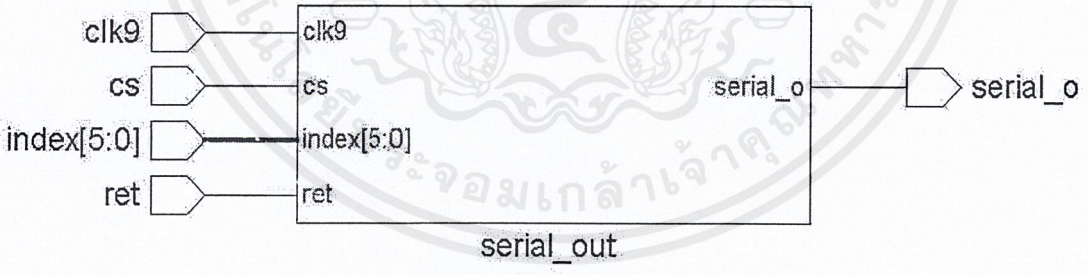
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 ผลการเลียนแบบการทำงานของภาค COMPARATOR

4.2.3 ผลการจำลองการทำงานสัญญาณของภาค SERIAL_OUT

ภาค SERIAL_OUT เป็นภาคที่ทำหน้าที่ส่งข้อมูลที่เป็น INDEX ออกไปยังภาครับในรูปแบบของสัญญาณแอนะลอก โดยส่งเรียงกันไปทีละบิต ลักษณะการต่อสัญญาณเพื่อทำการทดลองเลียนแบบการทำงานก็มีลักษณะเช่นเดียวกับ DATA_BUFFER และ COMPARATOR โดยการต่อสายสัญญาณภายนอกให้กับภาค SERIAL_OUT มีลักษณะดังรูปที่ 4.5

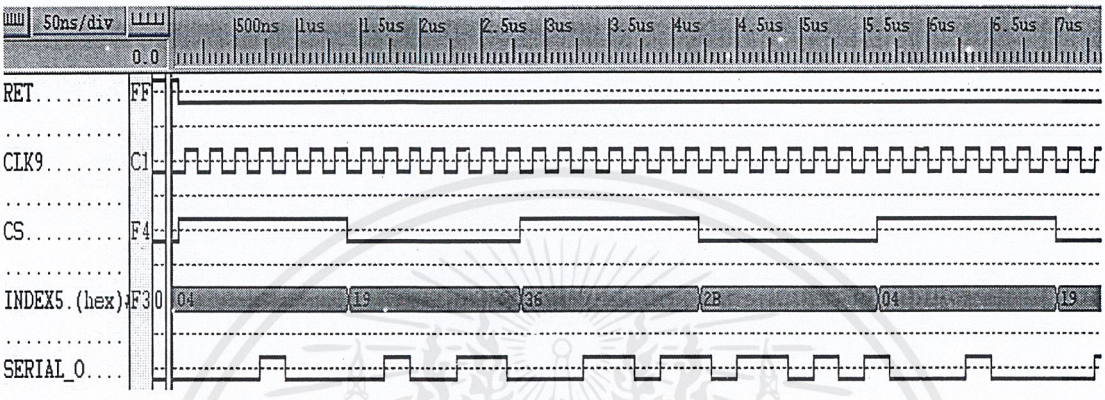


รูปที่ 4.5 ผังงานของ SERIAL_OUT

จากรูปที่ 4.6 เมื่อเริ่มต้นการทดลอง จะต้องให้สัญญาณ RET เป็น high เพื่อทำการเคลียร์ข้อมูลที่อยู่ใน buffer ของ SERIAL_OUT ก่อนการทำงานของภาค SERIAL_OUT คือจะต้องส่งข้อมูลขนาด 6 บิต ในลักษณะอนุกรมออกทางด้านเอาต์พุตโดยมีบิตเรตในการส่งข้อมูล 9,600 บิตต่อวินาที เมื่อมีสัญญาณอินพุตเข้ามาที่ขา INDEX[5:0] ชิฟรีจิสเตอร์ภายในของภาค SERIAL_OUT ซึ่งมี 2 ตัว จะสลับกันส่งสัญญาณออกมายังเอาต์พุต ในขณะที่สัญญาณ CS มีการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

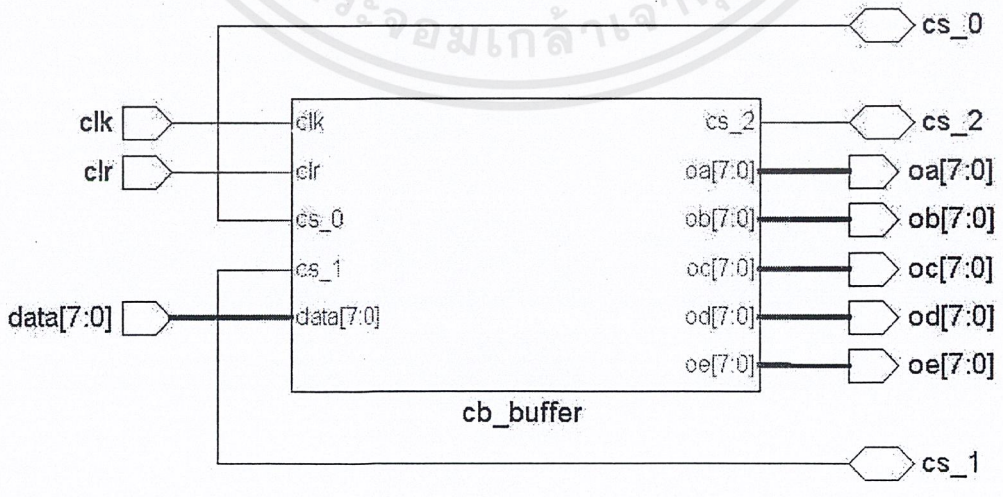
สภาวะจาก high เป็น low หรือเปลี่ยนจาก low เป็น high สังเกตได้จากรูปที่ 4.6 โดยที่ภาค SERIAL_OUT จะส่งข้อมูลของบิตนัยสำคัญต่ำสุดออกไปก่อน แล้วจึงตามด้วยบิตถัดไปจนครบทั้ง 6 บิต



รูปที่ 4. 6 ผลการเลียนแบบการทำงานของอุปกรณ์ SERIAL_OUT

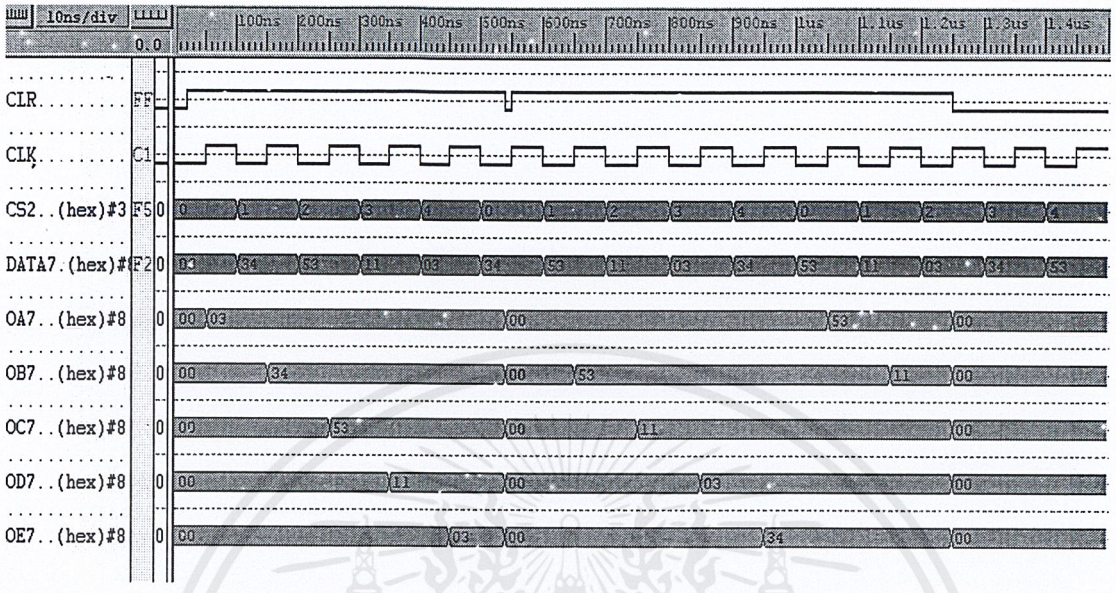
4.2.4 ผลการจำลองการทำงานสัญญาณของภาค CB_BUFFER

ในรูปที่ 4.7 เป็นผังงานที่ใช้ในการทดลองเลียนแบบการทำงานของสัญญาณภาค CB_BUFFER โดยมีขาสัญญาณอินพุต ดังนี้คือ CS, DATA, CLK, CLR และมีขาเอาต์พุตสัญญาณดังนี้คือ OA, OB, OC, OD, OE ในการเลียนแบบการทำงานของวงจรนี้จะใช้โปรแกรม LOGIC SIMULATOR ของ Xilinx Foundation F 1.5



รูปที่ 4.7 ผังงานของ CB_BUFFER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



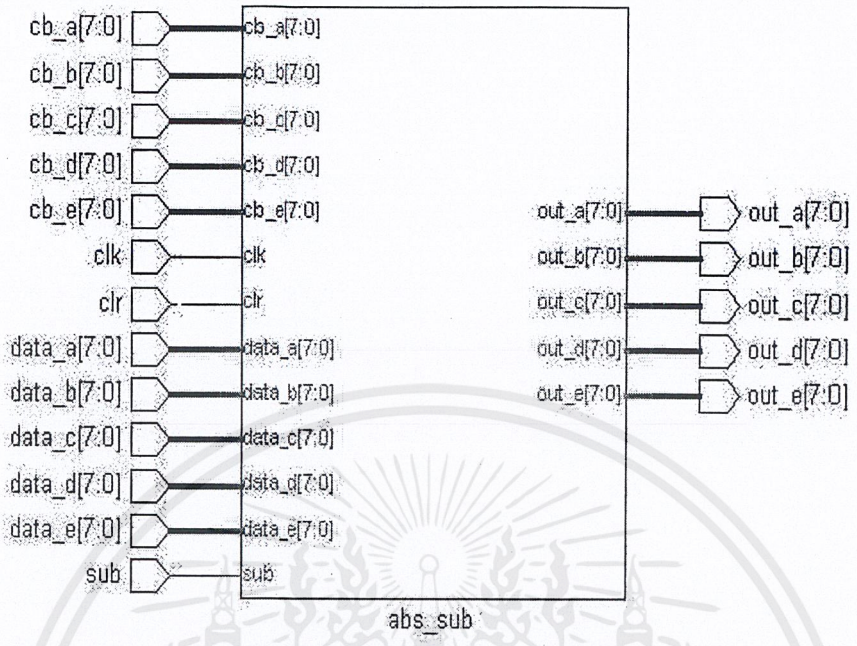
รูปที่ 4.8 ผลการเลียนแบบการทำงานของภาค CB_BUFFER

จากผลการทดลองดังรูปที่ 4.8 จะเห็นว่าเมื่อมีสัญญาณข้อมูลขนาด 8 บิต (ได้จาก codebook มีค่าเท่ากับ 1 อิลีเมนต์) เข้ามาที่ขาของ DATA[0:7] ข้อมูลจะถูกเลื่อนไปเก็บไว้ในตำแหน่งต่างๆใน Buffer โดยข้อมูลที่เข้ามาในอิลีเมนต์แรกสุดนั้นจะถูกเก็บที่ตำแหน่ง OA และในส่วนของข้อมูลตัวต่อไปก็จะถูกเก็บในแต่ละตำแหน่งถัดมาทั้งนี้แล้วในการส่งข้อมูลมาเก็บในแต่ละได้นั้นจะต้องถูกสัญญาณที่ที่ขา CS ควบคุมเมื่อ CS ได้รับข้อมูลชุดแรกเข้ามาก็จะสั่งให้นำข้อมูลใน DATAตัวแรกมาเก็บจะทำงานอย่างนี้ไปจนกว่าขา CS จะได้รับข้อมูล 5 ชุดเมื่อครบแล้วก็ส่งออกไปและขา CLR จากสภาวะที่ High จะเปลี่ยนเป็น Low เพื่อ ที่จะเก็บข้อมูลในชุดต่อไปจาก Codebook ที่ขาDATA จะสังเกตเห็นว่าจะทำงานต่างจาก DATA_BUFFER ตรงที่การเก็บข้อมูลใน Buffer เพราะใน DATA_BFFER นั้นจะต้องเก็บข้อมูลให้ครบ 5 ชุดก่อนจึงส่งข่อมูลออกไป

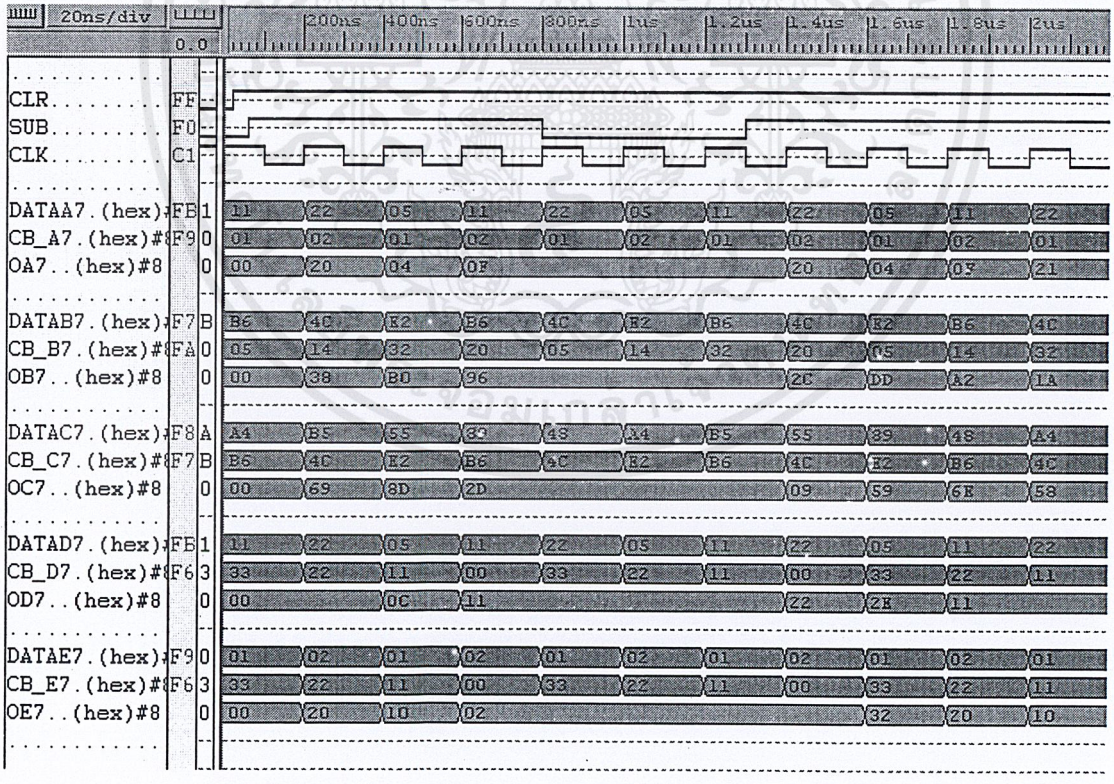
4.2.5 ผลการจำลองการทำงานสัญญาณของภาค ABS_SUB

ภาค ABS_SUB เป็นภาคที่ทำหน้าที่ ในการรับข้อมูลจากภาค DATA_BUFFER จำนวน 5 อิลีเมนต์และข้อมูลใน Codebook จำนวน 5 อิลีเมนต์ ซึ่งเก็บไว้ใน CB_BUFFER มาทำการลบกันในแบบที่ไม่คิดเครื่องหมาย เพื่อให้ข้อมูลตรงกันที่น้อยที่สุดจะมีลักษณะของผังงานดังรูปที่ 4.9 และในรูปที่ 4.10จะเป็นผลการเลียนแบบการทำงานของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 ผังงานของ ABS_SUB

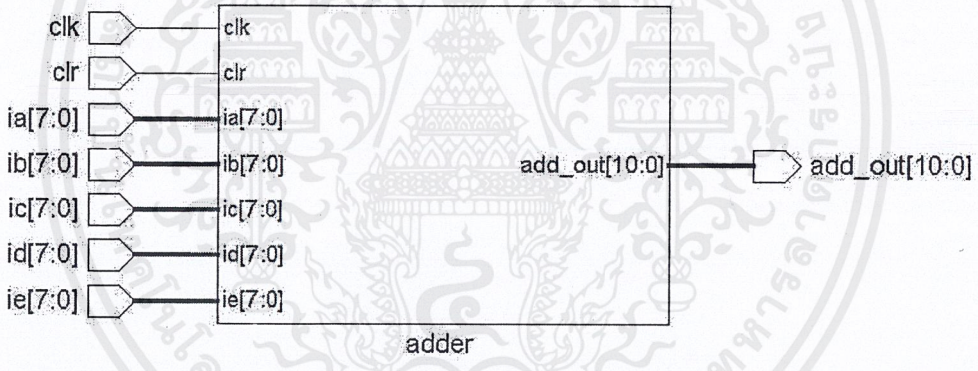


รูปที่ 4.10 ผลการเลียนแบบการทำงานของภาค ABS_SUB

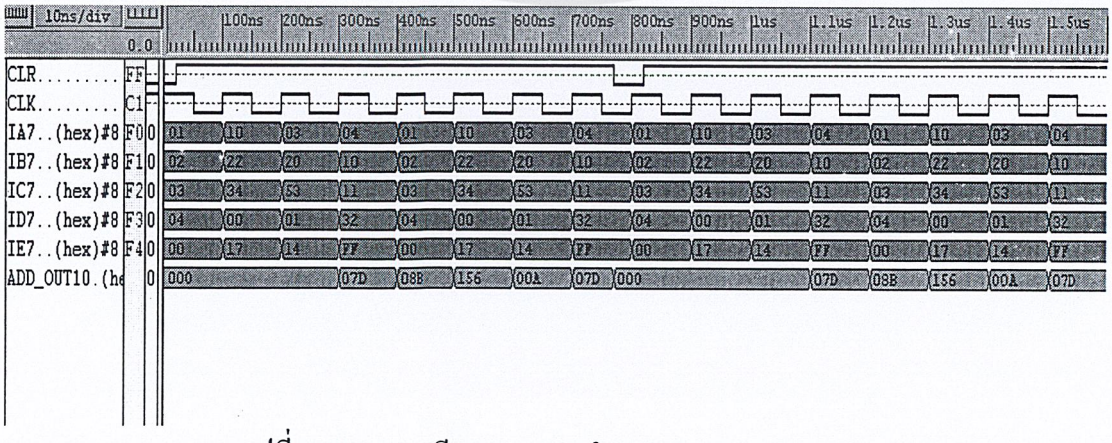
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.6 ผลการจำลองการทำงานสัญญาณของภาค ADDER

ในส่วนของภาค ADDERนั้นเป็นภาคที่ทำหน้าที่ในการนำข้อมูลที่ได้จากการลบแบบไม่คิดเครื่องหมายจากภาค ABS_SUB ซึ่งอินพุตที่ได้ จะมีข้อมูล 5 อีลีเมนต์มาป้อนเข้าวงจรบวก โดยแต่ละอีลีเมนต์ให้ชื่อว่า IA, IB, IC, ID และ IE ซึ่งเป็นข้อมูลขนาด 8 บิต การทำงานจะให้ข้อมูลบวกกัน 2 ชุดคือ IE บวกกับข้อมูลจาก ID และ IC บวกกับ IB ซึ่งจะได้ผลลัพธ์ที่มีขนาด 9 บิต ซึ่งเป็นการเผื่อบิตตัวทดไว้ จากนั้นนำผลลัพธ์ที่ได้ทั้งสองค่ามาบวกกันต่อไป ซึ่งจะได้ผลลัพธ์อีกค่าหนึ่งซึ่งมีขนาด 10 บิตเป็นการเผื่อบิตตัวทดไว้เช่นกัน จากนั้นผลลัพธ์ที่ได้นำไปบวกกับข้อมูลจาก IA จะต่อผ่าน BUFFER 2 ตัวด้วยกันผลลัพธ์ที่ได้จากการบวกจะกำหนดให้มีขนาด 11 บิตซึ่งเป็นการเผื่อบิตตัวทดไว้ ผลลัพธ์ที่ได้จากภาค ADDER จะส่งต่อไปยังภาค COMPARATOR ต่อไปซึ่งรายละเอียดผังงานของภาค ADDERจะแสดงในรูปที่ 4.11 และในรูปที่ 4.12จะแสดงถึงผลการจำลองการทำงานในภาค ADDER



รูปที่ 4.11 ผังงานภาค ADDER

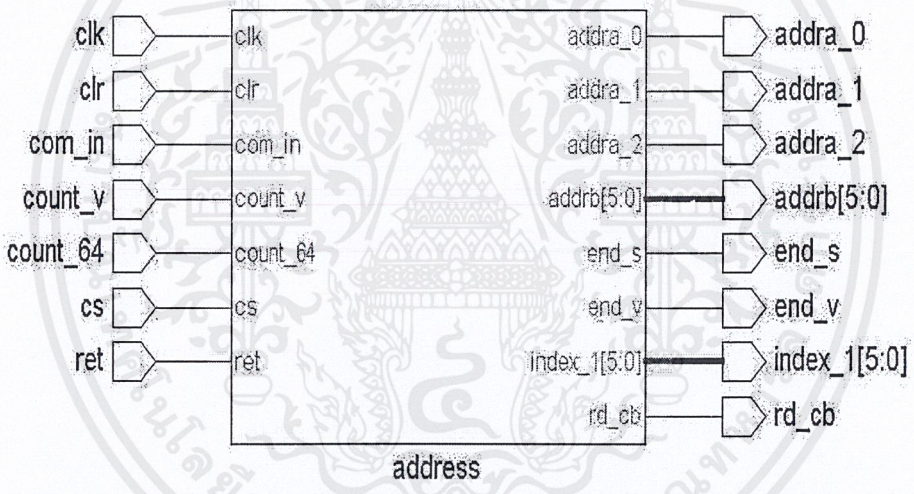


รูปที่ 4.12 ผลการเลียนแบบการทำงานของภาค ADDER

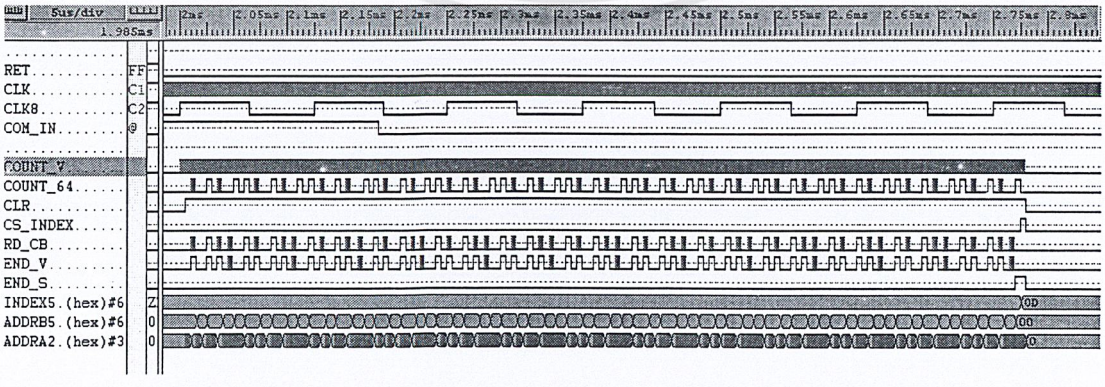
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.7 ผลการจำลองการทำงานสัญญาณของภาค ADDRESS&INDEX

ในรูปที่ 4.13 เป็นผังงานซึ่งเป็นภาคที่ทำหน้าที่ส่งตำแหน่งของข้อมูลไปเปิดโค้ดบุ๊ค และส่งสัญญาณค่าดัชนี ไปยังภาคต่อไป ซึ่งจะมีวงจรนับอยู่ 2 ชุด คือวงจรนับ 64 กับวงจรนับ 5 ซึ่งวงจรนับ 64 จะนับจำนวนเวกเตอร์ของข้อมูลในโค้ดบุ๊คโดยควบคุมจากภาค CONTROLเมื่อมี สัญญาณพัลส์เข้ามาลูกแรกวงจรนับ จะเพิ่มค่าครั้งละ 1 และมี BUFFER 4 ตัวทำหน้าที่หน่วงเวลาเพื่อให้เวลาในการทำงานของวงจรอื่นๆทำงานกันได้ทันเวลาพอดีและมีสัญญาณมาจากภาค COMPARATOR ควบคุมการจดจำค่าดัชนี และจะส่งค่าดัชนีออกไปเมื่อขา CS อยู่ในสถานะเป็น High และในรูปที่ 4.14 เป็นผลการจำลองการทำงานของภาค ADDRESS&INDEX



รูปที่ 4.13 ผังงานของภาค ADDRESS&INDEX

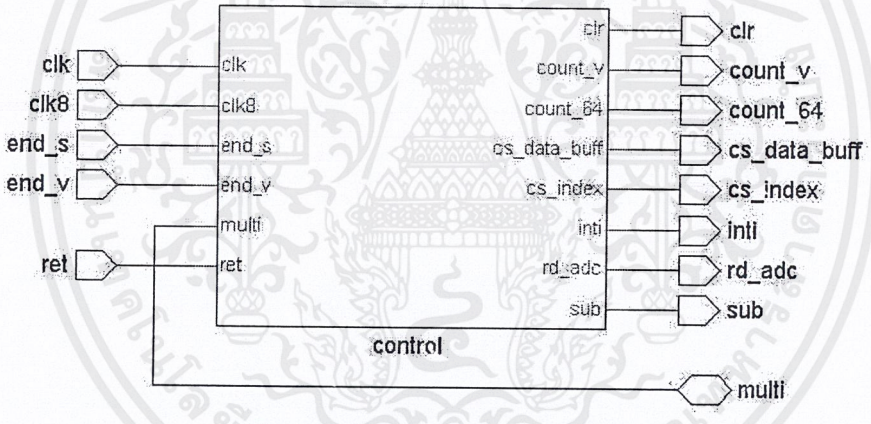


รูปที่ 4.14 ผลการเลียนแบบการทำงานของภาค ADDRESS&INDEX

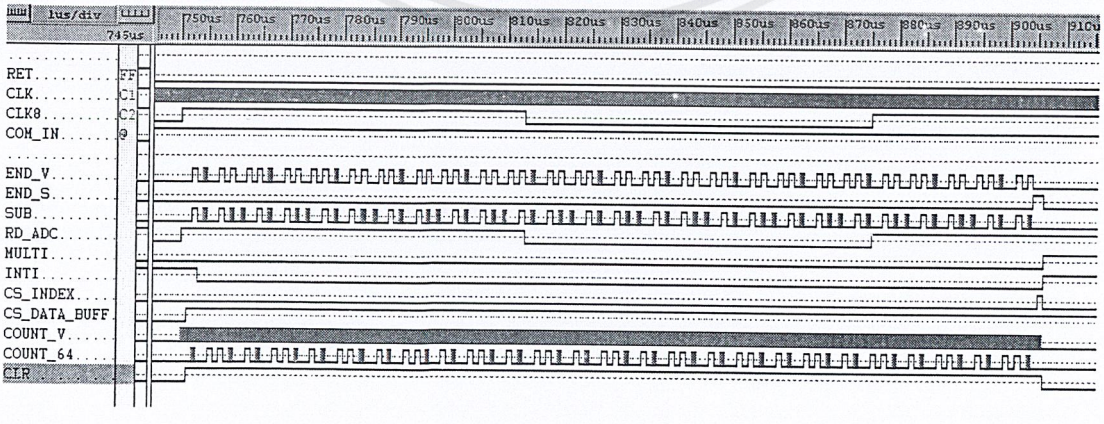
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.8 ผลการจำลองการทำงานสัญญาณของภาค CONTROL

ในรูปที่ 4.15 นั้นเป็นผังงานของภาคที่ทำหน้าที่ควบคุมเวลาการทำงานของภาคอื่นๆ โดยสัญญาณที่ป้อนให้ภาค CONTROL จะมีสัญญาณนาฬิกา 2 สัญญาณคือ ความถี่ 5.5 เมกะเฮิร์ต และความถี่ 8 กิโลเฮิร์ตและรับสัญญาณจากภาค ADDRESS&INDEX ซึ่งเป็นสัญญาณสิ้นสุดการทำงานแต่ละเวกเตอร์ของสัญญาณอินพุต เมื่อสัญญาณ RET เป็น '1' ภาค CONTROL จะส่งสัญญาณไปควบคุมการทำงานของภาคต่างๆ และเมื่อภาค CONTROL ได้รับสัญญาณสิ้นสุดการทำงานจากขา END_S ภาค CONTROL จะส่งสัญญาณไปให้ภาค SUB_ABS และภาค ADDER หยุดการทำงานและนับเวลาจากเริ่มการทำงานจนครบ 625 ไมโครวินาที ภาค CONTROL จะส่งสัญญาณให้ภาค SUB_ABS และภาค ADDER เริ่มการทำงานอีกครั้งหนึ่ง และผลการเลียนแบบการทำงานแสดงไว้ในรูปที่ 4.16



รูปที่ 4.15 ผังงานของภาค CONTROL



รูปที่ 4.16 ผลการเลียนแบบการทำงานของภาค CONTROL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผลการทดลอง

เมื่อนำผังงานในแต่ละผังงานมารวมแล้วเราสามารถหาผลการทำงานจริงได้แต่ตอนนี้ทำได้แค่เพียงตรวจสอบความถูกต้องของชิ้นงานได้เท่านั้นและยังไม่สามารถทดสอบใช้งานจริงได้อันเนื่องจากชิ้นงานที่ออกแบบยังไม่ได้นำไปเจือสารจริง ๆ ดังนั้นจึงไม่สามารถแสดงผลการทดลองจริงได้ แต่อย่างไรก็ตามในการทดลองในแต่ละผังงานของแต่ละภาคเป็นสิ่งที่ยืนยันได้ว่าวงจรนั้นสามารถทำงานได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป ปัญหา แนวทางการแก้ไขและพัฒนา

5.1 บทสรุป

จากที่ได้ทำการออกแบบและสร้างวงจรเข้ารหัสและถอดรหัสสัญญาณเสียงโดยใช้วิธีควอนไตซ์เวคเตอร์โดยออกแบบในลักษณะที่เป็นวงจรรวม ASIC ที่ใช้ Standard Cell มาทำการออกแบบและใช้โปรแกรม L-Edit เป็นเครื่องมือทำการออกแบบ ผลที่ได้เป็นไปตามวัตถุประสงค์ที่วางไว้คือ เข้าใจถึงหลักการทำงานของตัววงจรเข้ารหัสและถอดรหัสสัญญาณเสียง , การใช้ภาษา VHDL และได้รู้ถึงหลักการออกแบบวงจรรวม ที่ใช้ Standard Cell และสามารถใช้โปรแกรม L-Edit เป็นเครื่องช่วยในการออกแบบ และยังมีผลการจำลองการทำงานของวงจรที่เป็น Layout

5.2 ปัญหาและแนวทางการแก้ไข

1. ปัญหา ในการตรวจสอบวงจรที่เป็น Layout กับ Netlistว่าตรงตามกับ VHDLที่แปลงให้อยู่ในรูปของ Schematic.Sp หรือไม่นั้นยังทดสอบไม่ได้

แนวทางการแก้ปัญหา ศึกษาในเรื่องของการทดสอบใหม่และเรียนรู้การใช้โปรแกรม

2. ปัญหา ในการ Extract Layout จะเห็นว่ามีพอร์ตของวงจรจะถูกเรียงจากน้อยไปหามาก ซึ่งแต่กต่างจาก Netlist ที่แปลงจาก VHDL ทำให้ไม่สามารถ LVS ได้ถูกต้อง

แนวทางการแก้ไข อาจแก้ไขโดยการเปลี่ยนชื่อ พอร์ต ให้เรียงลำดับให้ตรงกัน

3. ปัญหา ไม่สามารถทดสอบผลการทดลองจริงได้เนื่องจากยังไม่ได้ทำการเจือสาร

แนวทางการแก้ไข ทำการทดสอบอย่างละเอียดในแต่ละบล็อกและหางบประมาณไปทำการเจือสาร

5.3 แนวทางในการพัฒนา

1) ในการออกแบบและสร้างวงจรเข้ารหัสและถอดรหัสสัญญาณเสียงที่ใช้ประเภท Standard Cell ควรที่จะเพิ่มในส่วนของวงจรตัวแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอลตัวเข้าไปในส่วนของชิฟด้วย

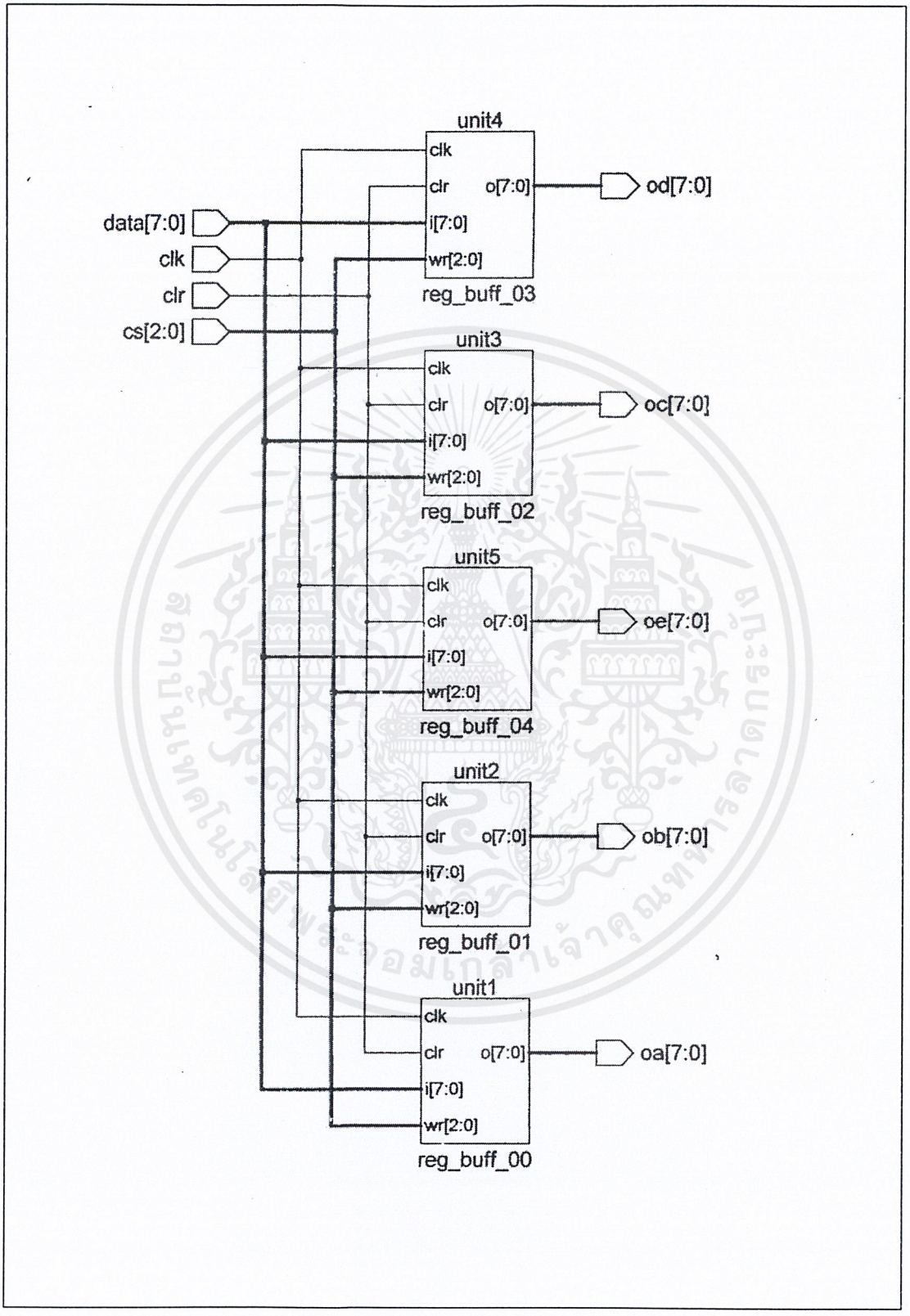
2) ให้ลองเปลี่ยนเทคโนโลยีของ Standard Cell ให้เล็กลงไปอีกปกติที่ใช้คือ 0.5 ไมครอน

3) ในการจัดวางเรียง Block ควรวางให้เกิดค่าของ Resister และ Capasister น้อยที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

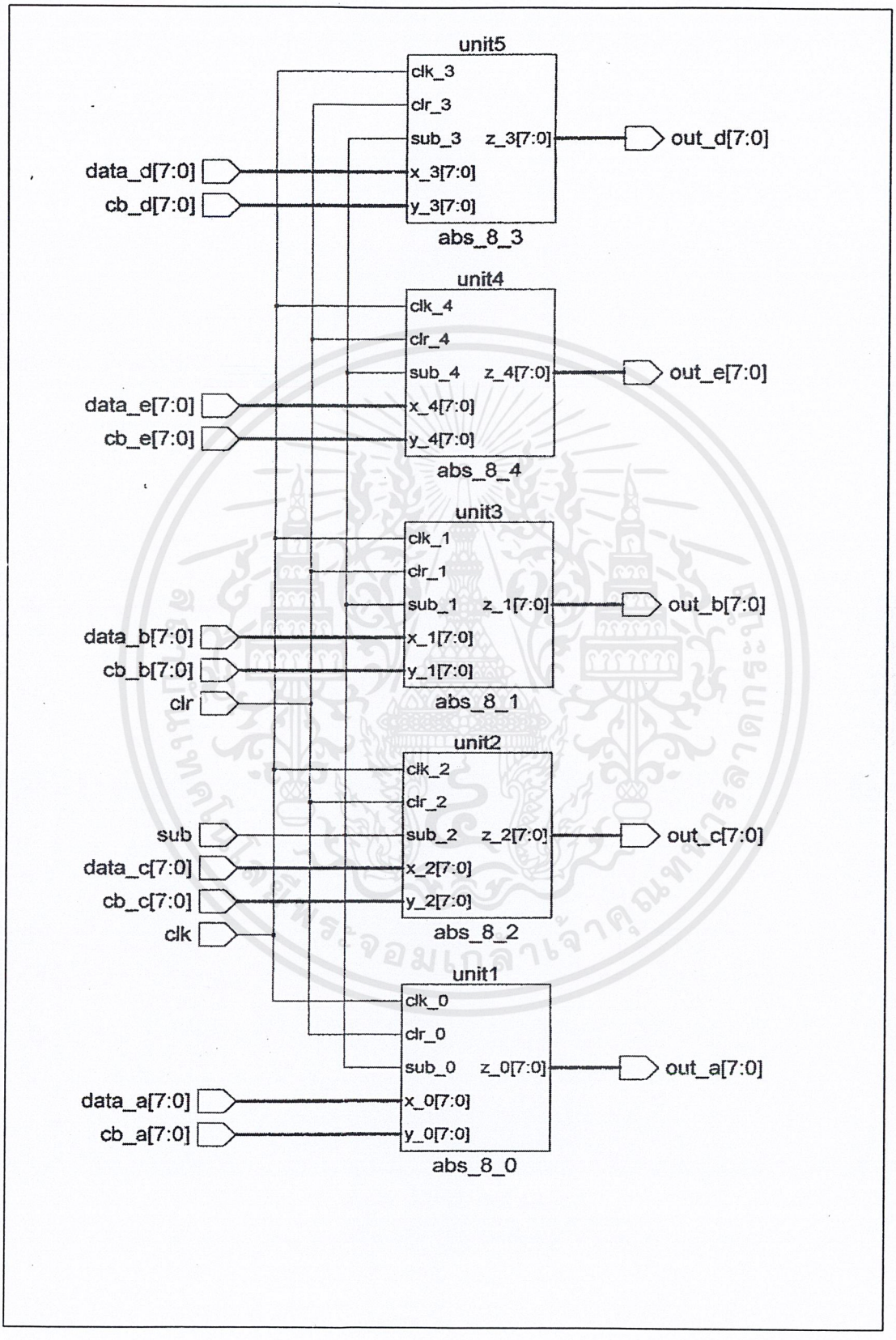
ภาคผนวก ก
รายละเอียดของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



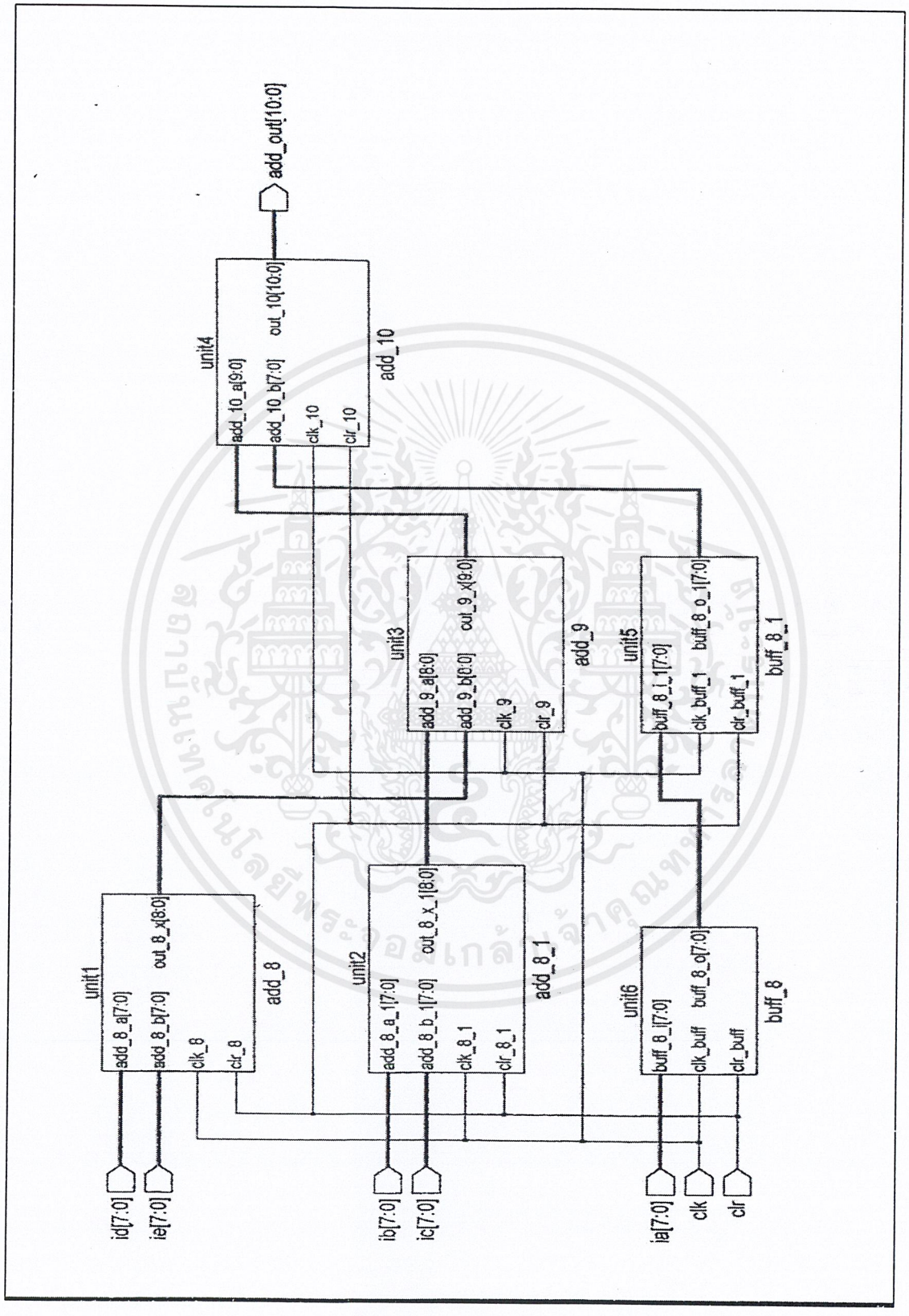
รูปที่ ก.1 CB_BUFFER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



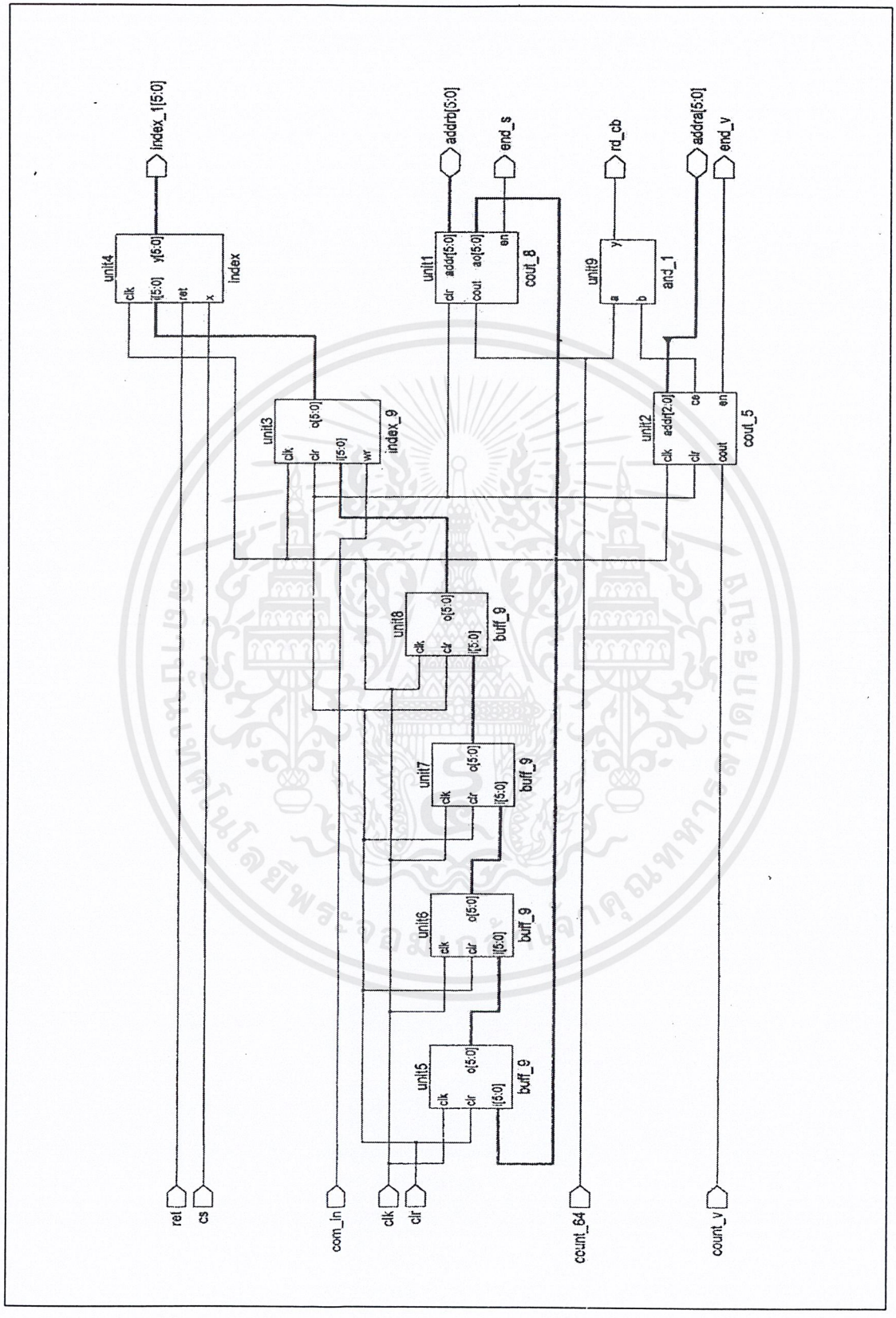
รูปที่ ก.2 ABS_SUB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



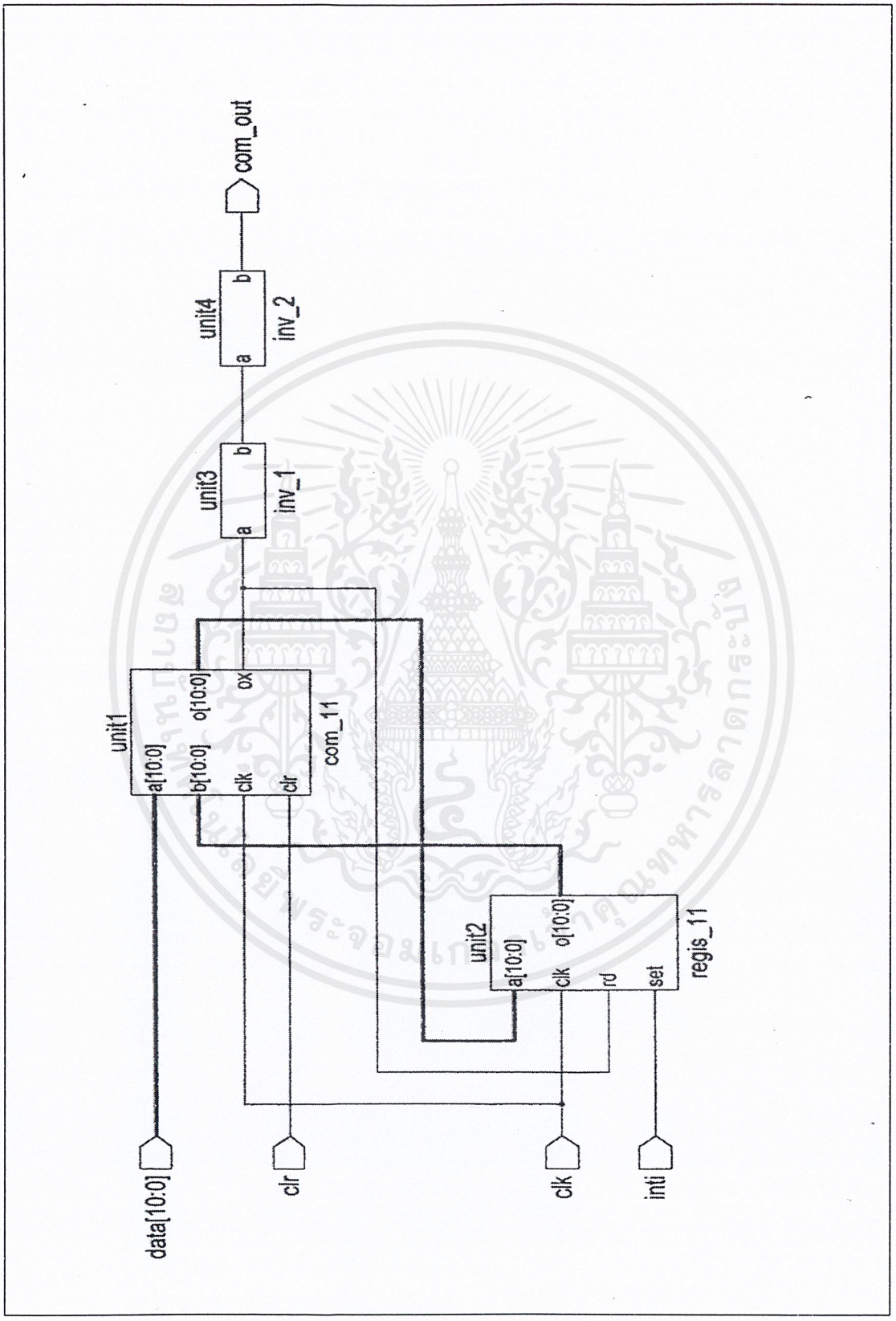
รูปที่ ก.3 ADDER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



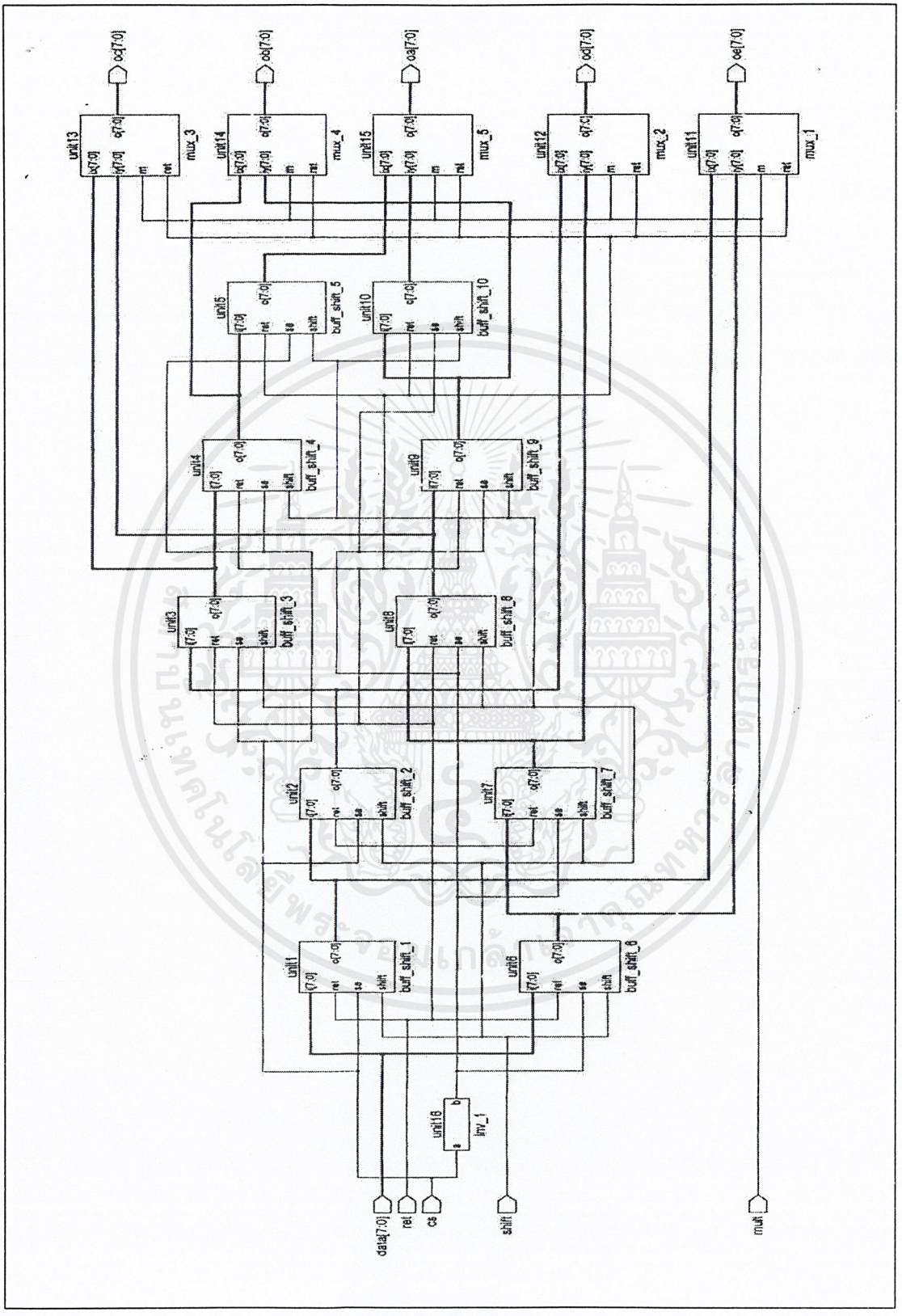
รูปที่ ก.4 ADDRESS&INDEX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.5 COMPARATOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.6 DATA_BUFFER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ภาษา VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
  component abs_sub
   port (
      data_a : IN std_logic_vector (7 DOWNTO 0) ;
      data_b : IN std_logic_vector (7 DOWNTO 0) ;
      data_c : IN std_logic_vector (7 DOWNTO 0) ;
      data_d : IN std_logic_vector (7 DOWNTO 0) ;
      data_e : IN std_logic_vector (7 DOWNTO 0) ;
      cb_a : IN std_logic_vector (7 DOWNTO 0) ;
      cb_b : IN std_logic_vector (7 DOWNTO 0) ;
      cb_c : IN std_logic_vector (7 DOWNTO 0) ;
      cb_d : IN std_logic_vector (7 DOWNTO 0) ;
      cb_e : IN std_logic_vector (7 DOWNTO 0) ;
      clk : IN std_logic ;
      clr : IN std_logic ;
      sub : IN std_logic ;
      out_a : OUT std_logic_vector (7 DOWNTO 0) ;
      out_b : OUT std_logic_vector (7 DOWNTO 0) ;
      out_c : OUT std_logic_vector (7 DOWNTO 0) ;
      out_d : OUT std_logic_vector (7 DOWNTO 0) ;
      out_e : OUT std_logic_vector (7 DOWNTO 0)) ;
  end component;
  component adder
    port (
      ie : IN std_logic_vector (7 DOWNTO 0) ;
      id : IN std_logic_vector (7 DOWNTO 0) ;
      ic : IN std_logic_vector (7 DOWNTO 0) ;
      ib : IN std_logic_vector (7 DOWNTO 0) ;   ia : IN std_logic_vector (7 DOWNTO 0) ;
clk : IN std_logic ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    clr : IN std_logic ;
    add_out : OUT std_logic_vector (10 DOWNTO 0)) ;
end component ;
component data_buffer
  port (
    shift : IN std_logic ;
    ret : IN std_logic ;
    mult : IN std_logic ;
    cs : IN std_logic ;
    data : IN std_logic_vector (7 DOWNTO 0) ;
    oe : OUT std_logic_vector (7 DOWNTO 0) ;
    od : OUT std_logic_vector (7 DOWNTO 0) ;
    oc : OUT std_logic_vector (7 DOWNTO 0) ;
    ob : OUT std_logic_vector (7 DOWNTO 0) ;
    oa : OUT std_logic_vector (7 DOWNTO 0)) ;
end component ;
component cb_buffer
  port (
    data : IN std_logic_vector (7 DOWNTO 0) ;
    cs_0 : INOUT std_logic ;
    cs_1 : INOUT std_logic ;
    cs_2 : INOUT std_logic ;
    clk : IN std_logic ;
    clr : IN std_logic ;
    oe : OUT std_logic_vector (7 DOWNTO 0) ;
    od : OUT std_logic_vector (7 DOWNTO 0) ;
    oc : OUT std_logic_vector (7 DOWNTO 0) ;
    ob : OUT std_logic_vector (7 DOWNTO 0) ;
    oa : OUT std_logic_vector (7 DOWNTO 0)) ;
end component ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
component control
  port (
    clk8 : IN std_logic ;
    clk : IN std_logic ;
    ret : IN std_logic ;
    end_v : IN std_logic ;
    end_s : IN std_logic ;
    clr : OUT std_logic ;
    sub : OUT std_logic ;
    inti : OUT std_logic ;
    count_v : OUT std_logic ;
    count_64 : OUT std_logic ;
    rd_adc : OUT std_logic ;
    multi : INOUT std_logic ;
    cs_data_buff : OUT std_logic ;
    cs_index : OUT std_logic) ;
end component ;
component comparator
  port (
    data : IN std_logic_vector (10 DOWNTO 0) ;
    clk : IN std_logic ;
    clr : IN std_logic ;
    inti : IN std_logic ;
    com_out : OUT std_logic) ;
end component ;
component address
  port (
    clk : IN std_logic ;
    clr : IN std_logic ;
    count_64 : IN std_logic ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        count_v : IN std_logic ;
        ret : IN std_logic ;
        cs : IN std_logic ;
        com_in : IN std_logic ;
        addrb : OUT std_logic_vector (5 DOWNTO 0) ;
        addra_0 : OUT std_logic ;
        addra_1 : OUT std_logic ;
        addra_2 : OUT std_logic ;
        rd_cb : OUT std_logic ;
        end_v : OUT std_logic ;
        end_s : OUT std_logic ;
        index_1 : OUT std_logic_vector (5 DOWNTO 0)) ;
    end component ;
    component serial_out
      port (
        ret : IN std_logic ;
        clk9 : IN std_logic ;
        cs : IN std_logic ;
        index : IN std_logic_vector (5 DOWNTO 0) ;
        serial_o : OUT std_logic) ;
    end component ;
end k ;
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity Chip is
    port (
      clk : IN std_logic ;
      clk9 : IN std_logic ;
      clk8 : IN std_logic ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ret : IN std_logic ;
    data : IN std_logic_vector (7 DOWNTO 0) ;
    cb : IN std_logic_vector (7 DOWNTO 0) ;
    addra_0 : OUT std_logic ;
    addra_1 : OUT std_logic ;
    addra_2 : OUT std_logic ;
    addrb : OUT std_logic_vector (5 DOWNTO 0) ;
    rd_cb : OUT std_logic ;
    serial_o : OUT std_logic ;
    rd_adc : OUT std_logic) ;
end Chip ;
architecture Chip_arch of test is
  for unit1 : work.k.abs_sub use entity work.abs_sub(INTERFACE);
  for unit2 : work.k.adder use entity work.adder(INTERFACE);
  for unit3 : work.k.data_buffer use entity work.data_buffer(INTERFACE);
  for unit4 : work.k.cb_buffer use entity work.cb_buffer(INTERFACE );
  for unit5 : work.k.control use entity work.control(INTERFACE);
  for unit6 : work.k.comparator use entity work.comparator(INTERFACE);
  for unit7 : work.k.address use entity work.address(INTERFACE);
  for unit8 : work.k.serial_out use entity work.serial_out(INTERFACE);
  signal int_data_0,int_data_1,int_data_2,int_data_3,int_data_4,int_data_5,
int_data_6,int_data_7,int_cb_0,int_cb_1,int_cb_2,int_cb_3,int_cb_4,int_cb_5,int_cb_6,int_cb_7,
int_ret,int_clk8,int_clk,int_clk9,int_rd_cb,int_rd_adc,int_addra_0,int_addra_1,int_addra_2,
int_addrb_0,int_addrb_1,int_addrb_2,int_addrb_3,int_addrb_4,int_addrb_5,int_serial_o,oa1_7,
oa1_6,oa1_5,oa1_4,oa1_3,oa1_2,oa1_1,oa1_0,ob1_7,ob1_6,ob1_5,ob1_4,ob1_3,ob1_2,ob1_1,
ob1_0,oc1_7,oc1_6,oc1_5,oc1_4,oc1_3,oc1_2,oc1_1,oc1_0,od1_7,od1_6,od1_5,od1_4,od1_3,
od1_2,od1_1,od1_0,oe1_7,oe1_6,oe1_5,oe1_4,oe1_3,oe1_2,oe1_1,oe1_0,data_a1_7,data_a1_6,
data_a1_5,data_a1_4,data_a1_3,data_a1_2,data_a1_1,data_a1_0,data_b1_7,data_b1_6,data_b1_5
,data_b1_4,data_b1_3,data_b1_2,data_b1_1,data_b1_0,data_c1_7,data_c1_6,data_c1_5,data_c1_
4, data_c1_3, data_c1_2, data_c1_1,data_c1_0, data_d1_7, data_d1_6, data_d1_5, data_d1_4,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
data_d1_3,data_d1_2,data_d1_1,data_d1_0,data_e1_7,data_e1_6,data_e1_5,ata_e1_4,
data_e1_3, data_e1_2,data_e1_1,data_e1_0,cb_a_7,cb_a_6,cb_a_5, cb_a_4, cb_a_3, cb_a_2,
cb_a_1, cb_a_0, cb_b_7, cb_b_6, cb_b_5,cb_b_4, cb_b_3, cb_b_2, cb_b_1, cb_b_0, cb_c_7,
cb_c_6, cb_c_5, cb_c_4,b_c_3, cb_c_2, cb_c_1, cb_c_0, cb_d_7, cb_d_6, cb_d_5, cb_d_4,
cb_d_3,cb_d_2, cb_d_1, cb_d_0, cb_e_7, cb_e_6, cb_e_5, cb_e_4, cb_e_3, cb_e_2,cb_e_1,
cb_e_0, add_out1_10, add_out1_9, add_out1_8, add_out1_7,add_out1_6, add_out1_5,
add_out1_4, add_out1_3, add_out1_2, add_out1_1,add_out1_0, end_v1, end_s1, clr1, sub1, inti1,
count_v1, count_641,mutil1, cs_data_buff1, cs_index1, com_in1, index_5, index_4, index_3,
index_2, index_1, index_0  : std_logic ;

begin
  unit1 : work.k.abs_sub port map ( data_a(7)=>data_a1_7,
     data_a(6)=>data_a1_6, data_a(5)=>data_a1_5, data_a(4)=>data_a1_4,
     data_a(3)=>data_a1_3, data_a(2)=>data_a1_2, data_a(1)=>data_a1_1,
     data_a(0)=>data_a1_0, data_b(7)=>data_b1_7, data_b(6)=>data_b1_6,
     data_b(5)=>data_b1_5, data_b(4)=>data_b1_4, data_b(3)=>data_b1_3,
     data_b(2)=>data_b1_2, data_b(1)=>data_b1_1, data_b(0)=>data_b1_0,
     data_c(7)=>data_c1_7, data_c(6)=>data_c1_6, data_c(5)=>data_c1_5,
     data_c(4)=>data_c1_4, data_c(3)=>data_c1_3, data_c(2)=>data_c1_2,
     data_c(1)=>data_c1_1, data_c(0)=>data_c1_0, data_d(7)=>data_d1_7,
     data_d(6)=>data_d1_6, data_d(5)=>data_d1_5, data_d(4)=>data_d1_4,
     data_d(3)=>data_d1_3, data_d(2)=>data_d1_2, data_d(1)=>data_d1_1,
     data_d(0)=>data_d1_0, data_e(7)=>data_e1_7, data_e(6)=>data_e1_6,
     data_e(5)=>data_e1_5, data_e(4)=>data_e1_4, data_e(3)=>data_e1_3,
     data_e(2)=>data_e1_2, data_e(1)=>data_e1_1, data_e(0)=>data_e1_0,
     cb_a(7)=>cb_a_7, cb_a(6)=>cb_a_6, cb_a(5)=>cb_a_5, cb_a(4)=>cb_a_4,
     cb_a(3)=>cb_a_3, cb_a(2)=>cb_a_2, cb_a(1)=>cb_a_1, cb_a(0)=>cb_a_0,
     cb_b(7)=>cb_b_7, cb_b(6)=>cb_b_6, cb_b(5)=>cb_b_5, cb_b(4)=>cb_b_4,
     cb_b(3)=>cb_b_3, cb_b(2)=>cb_b_2, cb_b(1)=>cb_b_1, cb_b(0)=>cb_b_0,
     cb_c(7)=>cb_c_7, cb_c(6)=>cb_c_6, cb_c(5)=>cb_c_5, cb_c(4)=>cb_c_4,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cb_c(3)=>cb_c_3, cb_c(2)=>cb_c_2, cb_c(1)=>cb_c_1, cb_c(0)=>cb_c_0,
cb_d(7)=>cb_d_7, cb_d(6)=>cb_d_6, cb_d(5)=>cb_d_5, cb_d(4)=>cb_d_4,
cb_d(3)=>cb_d_3, cb_d(2)=>cb_d_2, cb_d(1)=>cb_d_1, cb_d(0)=>cb_d_0,
cb_e(7)=>cb_e_7, cb_e(6)=>cb_e_6, cb_e(5)=>cb_e_5, cb_e(4)=>cb_e_4,
cb_e(3)=>cb_e_3, cb_e(2)=>cb_e_2, cb_e(1)=>cb_e_1, cb_e(0)=>cb_e_0,
clk=>int_clk, clr=>clr1, sub=>sub1, out_a(7)=>oa1_7, out_a(6)=>oa1_6,
out_a(5)=>oa1_5, out_a(4)=>oa1_4, out_a(3)=>oa1_3, out_a(2)=>oa1_2,
out_a(1)=>oa1_1, out_a(0)=>oa1_0, out_b(7)=>ob1_7, out_b(6)=>ob1_6,
out_b(5)=>ob1_5, out_b(4)=>ob1_4, out_b(3)=>ob1_3, out_b(2)=>ob1_2,
out_b(1)=>ob1_1, out_b(0)=>ob1_0, out_c(7)=>oc1_7, out_c(6)=>oc1_6,
out_c(5)=>oc1_5, out_c(4)=>oc1_4, out_c(3)=>oc1_3, out_c(2)=>oc1_2,
out_c(1)=>oc1_1, out_c(0)=>oc1_0, out_d(7)=>od1_7, out_d(6)=>od1_6,
out_d(5)=>od1_5, out_d(4)=>od1_4, out_d(3)=>od1_3, out_d(2)=>od1_2,
out_d(1)=>od1_1, out_d(0)=>od1_0, out_e(7)=>oe1_7, out_e(6)=>oe1_6,
out_e(5)=>oe1_5, out_e(4)=>oe1_4, out_e(3)=>oe1_3, out_e(2)=>oe1_2,
out_e(1)=>oe1_1, out_e(0)=>oe1_0);

unit2 : work.k.adder port map ( ie(7)=>oe1_7, ie(6)=>oe1_6,
ie(5)=>oe1_5, ie(4)=>oe1_4, ie(3)=>oe1_3, ie(2)=>oe1_2, ie(1)=>oe1_1,
ie(0)=>oe1_0, id(7)=>od1_7, id(6)=>od1_6, id(5)=>od1_5, id(4)=>od1_4,
id(3)=>od1_3, id(2)=>od1_2, id(1)=>od1_1, id(0)=>od1_0, ic(7)=>oc1_7,
ic(6)=>oc1_6, ic(5)=>oc1_5, ic(4)=>oc1_4, ic(3)=>oc1_3, ic(2)=>oc1_2,
ic(1)=>oc1_1, ic(0)=>oc1_0, ib(7)=>ob1_7, ib(6)=>ob1_6, ib(5)=>ob1_5,
ib(4)=>ob1_4, ib(3)=>ob1_3, ib(2)=>ob1_2, ib(1)=>ob1_1, ib(0)=>ob1_0,
ia(7)=>oa1_7, ia(6)=>oa1_6, ia(5)=>oa1_5, ia(4)=>oa1_4, ia(3)=>oa1_3,
ia(2)=>oa1_2, ia(1)=>oa1_1, ia(0)=>oa1_0, clk=>int_clk, clr=>clr1,
add_out(10)=>add_out1_10, add_out(9)=>add_out1_9, add_out(8)=>
add_out1_8, add_out(7)=>add_out1_7, add_out(6)=>add_out1_6, add_out(5)
=>add_out1_5, add_out(4)=>add_out1_4, add_out(3)=>add_out1_3,
add_out(2)=>add_out1_2, add_out(1)=>add_out1_1, add_out(0)=>add_out1_0   );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit3 : work.components.data_buffer port map ( shift=>int_cik8, ret=>int_ret,
  mult=>mutil1, cs=>cs_data_buff1,
  data(7)=>int_data_7, data(6)=>int_data_6,data(5)=>int_data_5, data(4)=>int_data_4,
  data(3)=>int_data_3, data(2)=>int_data_2,data(1)=>int_data_1, data(0)=>int_data_0,
  oe(7)=>data_e1_7, oe(6)=>data_e1_6,
  oe(5)=>data_e1_5, oe(4)=>data_e1_4, oe(3)=>data_e1_3, oe(2)=>data_e1_2,
  oe(1)=>data_e1_1, oe(0)=>data_e1_0, od(7)=>data_d1_7, od(6)=>data_d1_6,
  od(5)=>data_d1_5, od(4)=>data_d1_4, od(3)=>data_d1_3, od(2)=>data_d1_2,
  od(1)=>data_d1_1, od(0)=>data_d1_0, oc(7)=>data_c1_7, oc(6)=>data_c1_6,
  oc(5)=>data_c1_5, oc(4)=>data_c1_4, oc(3)=>data_c1_3, oc(2)=>data_c1_2,
  oc(1)=>data_c1_1, oc(0)=>data_c1_0, ob(7)=>data_b1_7, ob(6)=>data_b1_6,
  ob(5)=>data_b1_5, ob(4)=>data_b1_4, ob(3)=>data_b1_3, ob(2)=>data_b1_2,
  ob(1)=>data_b1_1, ob(0)=>data_b1_0, oa(7)=>data_a1_7, oa(6)=>data_a1_6,
  oa(5)=>data_a1_5, oa(4)=>data_a1_4, oa(3)=>data_a1_3, oa(2)=>data_a1_2,
  oa(1)=>data_a1_1, oa(0)=>data_a1_0);

unit4 : work.components.cb_buffer port map ( data(7)=>int_cb_7, data(6)=>
  int_cb_6, data(5)=>int_cb_5, data(4)=>int_cb_4, data(3)=>int_cb_3, data(2)=>int_cb_2,
  data(1)=>int_cb_1, data(0)=>int_cb_0, cs_0=>int_addra_0, cs_1=>int_addra_1, cs_2=>
  int_addra_2, clk=>int_clk, clr=>clr1, oe(7)=>cb_e_7, oe(6)=>cb_e_6, oe(5)=>
  cb_e_5, oe(4)=>cb_e_4, oe(3)=>cb_e_3, oe(2)=>cb_e_2, oe(1)=>cb_e_1,
  oe(0)=>cb_e_0, od(7)=>cb_d_7, od(6)=>cb_d_6, od(5)=>cb_d_5, od(4)=>
  cb_d_4, od(3)=>cb_d_3, od(2)=>cb_d_2, od(1)=>cb_d_1, od(0)=>cb_d_0,
  oc(7)=>cb_c_7, oc(6)=>cb_c_6, oc(5)=>cb_c_5, oc(4)=>cb_c_4, oc(3)=>
  cb_c_3, oc(2)=>cb_c_2, oc(1)=>cb_c_1, oc(0)=>cb_c_0, ob(7)=>cb_b_7,
  ob(6)=>cb_b_6, ob(5)=>cb_b_5, ob(4)=>cb_b_4, ob(3)=>cb_b_3, ob(2)=>
  cb_b_2, ob(1)=>cb_b_1, ob(0)=>cb_b_0, oa(7)=>cb_a_7, oa(6)=>cb_a_6,
  oa(5)=>cb_a_5, oa(4)=>cb_a_4, oa(3)=>cb_a_3, oa(2)=>cb_a_2, oa(1)=>
  cb_a_1, oa(0)=>cb_a_0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unit5 : work.components.control port map ( clk8=>int_clk8, clk=>int_clk, ret=>int_ret,
end_v=>end_v1, end_s=>end_s1, clr=>clr1, sub=>sub1, inti=>inti1,count_v=>count_v1,
count_64=>count_641, rd_adc=>int_rd_adc, multi=>mutil1,cs_data_buff=>cs_data_buff1,
cs_index=>cs_index1);

  unit6 : work.components.comparator port map ( data(10)=>add_out1_10,
    data(9)=>add_out1_9, data(8)=>add_out1_8, data(7)=>add_out1_7, data(6)
    =>add_out1_6, data(5)=>add_out1_5, data(4)=>add_out1_4, data(3)=>
    add_out1_3, data(2)=>add_out1_2, data(1)=>add_out1_1, data(0)=>
    add_out1_0, clk=>int_clk, clr=>clr1, inti=>inti1, com_out=>com_in1);

  unit7 : work.components.address port map ( clk=>int_clk, clr=>clr1, count_64 =>count_641,
count_v=>count_v1, ret=>int_ret, cs=>cs_index1, com_in=>  com_in1, addrb(5)=>int_addrb_5,
addrb(4)=>int_addrb_4, addrb(3)=>int_addrb_3,addrb(2)=>int_addrb_2, addrb(1)=>int_addrb_1,
addrb(0)=>int_addrb_0, addra_0=>  int_addra_0, addra_1=>int_addra_1, addra_2=>int_addra_2,
rd_cb=>int_rd_cb,  end_v=>end_v1,  end_s=>end_s1,  index_1(5)=>index_5,  index_1(4)=>
index_4,index_1(3)=>index_3,index_1(2)=>index_2,index_1(1)=>index_1,index_1(0)=>
index_0);

  unit8 : work.components.serial_out port map ( ret=>int_ret, clk9=>int_clk9, cs=>
    mutil1, index(5)=>index_5, index(4)=>index_4, index(3)=>index_3,
    index(2)=>index_2, index(1)=>index_1, index(0)=>index_0, serial_o=>
    int_serial_o);
end Chip_arch ;
```

รูปที่ ข.1 โปรแกรม VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ดาบชัย พัวพานิช์และคณะ. “การพัฒนาตัวเข้ารหัส เสียงโดยวิธีการควอนไตซ์เวกเตอร์แบบมาตรฐานและอุปกรณ์ FPGAs.” ปริญญานิพนธ์ครุศาสตร์อุตสาหกรรม,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2540

บุญยอด แสงคุณธรรมและคณะ. “การเข้ารหัสเสียงโดยวิธีการควอนไตซ์เวกเตอร์ด้วยอุปกรณ์ FPGAs.” ปริญญานิพนธ์ครุศาสตร์อุตสาหกรรม,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2540

ปราณี ทิพย์สุวรรณ์และคณะ. “ตัวประมวลผลสำหรับค้นหาข้อมูลในโค้ดบุ๊ค” ปริญญานิพนธ์ครุศาสตร์อุตสาหกรรม,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2542

Xilink. The Programmable Logic data Book :3 rd ed. Xilink Inc. 1994

L-Edit™ User Guide Volume 1:Layout Edit, for Windows 95/98/NT, Tanner Tools

L-Edit™ User Guide Volume 2:Placement and Routing, for Windows 95/98/NT, Tanner Tools

L-Edit™ User GuideVolume 3:Design Verification, for Windows 95/98/NT, Tanner Tools

L-Edit™ User GuideVolume 4:User-Programmable Interface, for Windows 95/98/NT, Tanner Tools

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายบัณฑิตศักดิ์ ศรีโสภา
วันเดือนปีเกิด	16 กันยายน พ.ศ. 2520
สถานที่เกิด	จังหวัด ชลบุรี
ภูมิลำเนาเดิม	38/1 หมู่ 13 ตำบล วังหว้า อำเภอ แกลง จังหวัด ระยอง 21110
ที่อยู่ปัจจุบัน	300/53 หมู่บ้านรุ่งอรุณ 1 ถนนฉลองกรุง แขวงลำปาทิว เขตลาดกระบัง
โทรศัพท์	(02) 7374452-3 ต่อ 106
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนอนุบาลระยอง
มัธยมศึกษา	โรงเรียนระยองวิทยาคม
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคสัตหีบ
ประกาศนียบัตรวิชาชีพชั้นสูง	สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
ปริญญาตรี	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
ทุนการศึกษา	-
คติพจน์	ประสบการณ์หาซื้อไม่ได้ด้วยเงินตรา แต่ได้มา จากการลงมือกระทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้ทำปริญญานิพนธ์	นายวิทยา บุตรโสภา
วันเดือนปีเกิด	17 มิถุนายน พ.ศ. 2521
สถานที่เกิด	จังหวัดสระบุรี
ภูมิลำเนาเดิม	38/1 หมู่ 13 ตำบล บ้านครัว อำเภอบ้านหมอ
	จังหวัดสระบุรี 18270
ที่อยู่ปัจจุบัน	300/108 หมู่บ้านรุ่งอรุณ 1 ถนนฉลองกรุง
	แขวงลำปาทิว เขตลาดกระบัง
โทรศัพท์	(02) 7373897
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนราษฎร์ศึกษา
มัธยมศึกษา	โรงเรียนราษฎร์ศึกษา
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคสระบุรี
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคท่าหลวงซีเมนต์ไทยอนุสรณ์
ปริญญาตรี	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
	ภาควิชาครุศาสตร์วิศวกรรม
	คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
ทุนการศึกษา	กู้ยืมรัฐบาล
คติพจน์	เมื่อมีสมาธิก่อให้เกิดสติ เมื่อมีสติก่อให้เกิดปัญญา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้