สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

วงจรสังเคราะห์ความถี่แบบดิจิตอลโดยตรงที่สามารถโปรแกรมได้โดยใช้ FPGA

PROGRAMMABLE DIRECT DIGITAL FREQUENCY SYNTHESIZER USING FPGA

โดย

นายธนะสิน ปลุกใจ

นายวิรัตน์ วัฒนากร

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....
เลขทะเบียน..... 61991
วัน,เดือน,ปี..... 25 ก.ค. 2549

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

ภาควิชา
วิศวกรรมโทรคมนาคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

วงจรสังเคราะห์ความถี่แบบดิจิตอลโดยตรงที่สามารถโปรแกรมได้โดยใช้ FPGA

PROGRAMMABLE DIRECT DIGITAL FREQUENCY SYNTHESIZER USING FPGA

โดย

นายธนะสิน ปลุกใจ 45015010

นายวิรัตน์ วัฒนากร 45015034

อาจารย์ที่ปรึกษา

รศ.ดร. กอบชัย เดชหาญ

อ.ศรวัฒน์ ชิวปรีชา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **วงจรสังเคราะห์ความถี่แบบดิจิตอลโดยตรงที่สามารถโปรแกรมได้โดยใช้ FPGA**

PROGRAMMABLE DIRECT DIGITAL FREQUENCY SYNTHESIZER USING FPGA

ผู้จัดทำ

1.นายธนะสิน ปลุกใจ 45015010

2.นายวิรัตน์ วัฒนากร 45015034

(.....) อาจารย์ที่ปรึกษา

รศ.ดร. กอบชัย เดชหาญ

(.....) อาจารย์ที่ปรึกษา

อาจารย์ ศรวัฒน์ ชิวปรีชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรสังเคราะห์ความถี่แบบดิจิตอลโดยตรงที่สามารถโปรแกรมได้โดยใช้ FPGA

PROGRAMMABLE DIRECT DIGITAL FREQUENCY SYNTHESIZER USING FPGA

โดย นายธนะสิน ปลุกใจ 45015010

นายวิรัตน์ วัฒนากร 45015034

อาจารย์ที่ปรึกษา รศ.ดร. กอบชัย เดชหาญ

อาจารย์ศรวัฒน์ ชิวปรีชา

บทคัดย่อ

โครงงานนี้นำเสนอการสร้าง วงจรสังเคราะห์ความถี่แบบดิจิตอลโดยตรงที่สามารถโปรแกรมได้โดยใช้ FPGA (Programmable Direct Digital Frequency Synthesizer using FPGA : DDS) ที่สามารถโปรแกรมได้และสามารถกำหนดความถี่ของสัญญาณได้ โดยการรับค่าจากคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม หรือคีย์บอร์ดผ่านทาง MCS-51 ไปยังอุปกรณ์ FPGA (Field Programmable Gate Array) โดยอธิบายพฤติกรรมการทำงานของวงจรด้วยภาษา VHDL (Very High Speed Integrated Circuit Hardware Description Language)

ABSTRACT

This project presents an implementation of programmable direct digital frequency synthesizer using FPGA. The frequency of signal can be assingned by frequency control words from computer via serial port or keyboard via MCS-51 to FPGA. The behavior of the designed circuit can be described by using VHDL.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของปริญญานิพนธ์	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 ทฤษฎีการสังเคราะห์สัญญาณดิจิตอลโดยตรง	2
2.2 การเขียนภาษา VHDL	3
2.2.1 Terminology และ Convention	3
2.2.2 การออกแบบจากบนลงล่าง	6
2.2.3 ภาษา VHDL และ ส่วนประกอบต่างๆของภาษา	7
2.2.3.1 หน่วยการออกแบบเอนทิตี	8
2.2.3.2 หน่วยการออกแบบสถาปัตยกรรม	9
2.2.3.3 หน่วยการออกแบบแพ็กเก็ต	12
2.2.3.4 หน่วยการออกแบบโครงแบบ	13
2.2.4 ชุดคำสั่งลำดับ (Sequential Statements)	14
2.2.4.1 Process Statement	14
2.2.4.2 Wait Statement	15
2.2.4.3 IF-THEN-ELSE statement	16
2.2.4.4 CASE Statement	16
2.3 การบีบอัดควอแดรนท์ (Quadrant compression)	17
2.3.1 ขั้นตอนวิธีซันเดอร์แลนด์ (Sunderland algorithm)	20
บทที่ 3 การออกแบบและการสร้าง	21
3.1 การออกแบบวงจรเชิงเลขด้วยอุปกรณ์ FPGA	21
3.2 การออกแบบโดยใช้ภาษาอธิบายการทำงานของฮาร์ดแวร์	22
3.2.1 การจำลองการทำงานของวงจร (Simulation)	22
3.2.2 การสังเคราะห์วงจร	22
3.2.3 การแบ่งวงจร (Partitioning)	23
3.2.4 การวางอุปกรณ์ (Placement)	23
3.2.5 การเชื่อมต่อสัญญาณ (Routing)	23
3.2.6 การโปรแกรมอุปกรณ์ FPGA (Configuration)	23
3.3 FPGA ตระกูล ACEX1K	24
3.4 Direct Digital Frequency Synthesizer (DDS)	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 การออกแบบและการสร้าง 28
3.4.2 การสร้างสัญญาณรูปคลื่นไซน์ 29
3.5 รายละเอียดการออกแบบ 42
3.6 วงจรแปลงระดับแรงดัน 46
3.6.1 หลักการทำงาน 46
3.7วงจรแปลงระดับสัญญาณดิจิตอลเป็นอนาล็อก 47
3.8 การออกแบบวงจรและลายวงจร 48
3.8.1 การออกแบบวงจรและลายวงจรส่วนของวงจรแปลงดิจิตอลเป็นอนาล็อก 48
3.8.2 การออกแบบวงจรและลายวงจรส่วนของวงจรส่งข้อมูลผ่านพอร์ตอนุกรม โดยใช้MCS-51 49
บทที่ 4 การทดลองและผลการทดลอง 51
4.1 การควบคุมการใช้งานการเชื่อมต่อผ่านพอร์ตอนุกรม 51
4.2 การออกแบบวงจรส่วนต่างๆ โดยใช้ภาษา VHDL 54
4.2.1 ส่วนของวงจร DIV 1000 54
4.2.2 ส่วนของวงจร LATCH 55
4.2.3 ส่วนของวงจร ONEPULSE 56
4.2.4 ส่วนของวงจร SERIAL_COMMUNICATION 57
4.2.5 ส่วนของวงจร SELECT INPUT 58
4.2.6 ส่วนของวงจร LFSR 59
4.2.7 ส่วนของวงจร DDS 60
4.2.8 ส่วนของวงจร SUNDERLANS 61
4.2.9 ส่วนของวงจร MUX_OUTPUT 62
4.3 ผลการทดสอบการทำงานของ DDS Oscillator 63
4.3.1 การทดสอบการกำเนิดสัญญาณซายน์ที่ความถี่ต่างๆ 63
4.3.2 การทดสอบการกำเนิดสัญญาณสามเหลี่ยมที่ความถี่ต่างๆ 69
4.3.3 การทดสอบการกำเนิดสัญญาณสามเหลี่ยมที่ความถี่ต่างๆ 74
4.3.4 การทดสอบการกำเนิดสัญญาณฟันเลื่อยที่ความถี่ต่างๆ 79
4.4 ผลการทดสอบการทำงานของ SUNDERLAND 85
4.4.1 การทดสอบการกำเนิดสัญญาณซายน์ที่ความถี่ต่างๆ 85
4.4.2 การทดสอบการกำเนิดสัญญาณสามเหลี่ยมที่ความถี่ต่างๆ 93
4.5 การทดสอบสัญญาณแบบสุ่ม 101
บทที่ 5 บทวิจารณ์และสรุป 102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

กิตติกรรมประกาศ

บรรณานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 ตัวอย่างการเก็บข้อมูลของสัญญาณไซน์ (Sine Wave) ของ 1 คาบเวลา	2
รูปที่ 2.2 แสดงขั้นตอนการออกแบบจากบนลงล่าง	6
รูปที่ 2.3 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี้	8
รูปที่ 2.4 แสดงรูปแบบของ RS_flipflop	8
รูปที่ 2.5 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม	9
รูปที่ 2.6 แสดงหน่วยการออกแบบสถาปัตยกรรมของ RS_flipflop	10
รูปที่ 2.7 แสดงโครงสร้างภายในสถาปัตยกรรมของ RS_flipflop	10
รูปที่ 2.8 แสดงหน่วยการออกแบบสถาปัตยกรรมของ RS_flipflop ในลักษณะโครงสร้าง	11
รูปที่ 2.9 แสดงหน่วยการออกแบบสถาปัตยกรรมของ RS_flipflop ในลักษณะพฤติกรรม	11
รูปที่ 2.10 แสดงหน่วยการออกแบบสถาปัตยกรรมของ RS_flipflop ในลักษณะผสม	12
รูปที่ 2.11 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต	13
รูปที่ 2.12 แสดงโครงสร้างโดยทั่วไปของบอดี้แพ็กเก็ต	13
รูปที่ 2.13 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ	14
รูปที่ 2.14 การสังเคราะห์สัญญาณซายน์เต็มรูปด้วยเทคนิคบีบอัดควอแดรนท์	18
รูปที่ 2.15 ขั้นตอนวิธีผลต่างซายน์-เฟส	19
รูปที่ 2.16 ขั้นตอนวิธีซันเดอร์แลนด์	20
รูปที่ 3.1 ลักษณะของตัว FPGA และการนำไปใช้งาน	21
รูปที่ 3.2 โครงสร้างภายในของ FPGA ตระกูล ACEX1k	25
รูปที่ 3.3 แสดง Block Diagram โครงสร้างของวงจรสังเคราะห์ความถี่ที่ใช้หน่วยความจำเป็นค่าขนาดของสัญญาณเอาไว้	25
รูปที่ 3.4 วงจรสร้างสัญญาณซายน์แบบเชิงเลข	26
รูปที่ 3.5 แสดงการกำเนิดสัญญาณซายน์แบบดิจิตอล ที่มีค่า Phase Increment เท่ากับ 55	27
รูปที่ 3.6 Phase Increment เท่ากับ 55	28
รูปที่ 3.7 ขั้นตอนการทำงานของ ตัวกำเนิดสัญญาไซน์	29
รูปที่ 3.8 ผลจากค่าในตารางมาวาดกราฟ	41
รูปที่ 3.9 สถาปัตยกรรมของชิปวงจรรวมเพื่อสังเคราะห์ความถี่แบบดิจิตอลที่ใช้หน่วยความจำ	42
รูปที่ 3.10 โครงสร้างของวงจรสร้างเฟสขนาด 32 บิต	43
รูปที่ 3.11 โครงสร้างของวงจรบวก 4 บิต แบบริปเปิลแคร์รี่	44
รูปที่ 3.12 ขั้นตอนวิธีโมดิฟายซันเดอร์แลนด์	45
รูปที่ 3.13 การสังเคราะห์สัญญาณรูปฟันเลื่อย	45
รูปที่ 3.14 วงจรแปลงระดับแรงดัน	46
รูปที่ 3.15 วงจรแปลงระดับสัญญาณดิจิตอลเป็นอนาล็อก	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 3.16 วงจรแปลงดิจิตอลเป็นอนาล็อกและลายวงจรแปลงดิจิตอลเป็นอนาล็อก	48
รูปที่ 3.17 วงจรส่งข้อมูลผ่านพอร์ตอนุกรมและลายวงจรส่งข้อมูลมาจากMCS-51	49
รูปที่ 3.18 ภาพถ่ายอุปกรณ์และวงจรใช้งานจริง	50
รูปที่ 4.1 แสดงเมนูหลักของโปรแกรมการใช้งานดิจิตอลออสซิลเลเตอร์	52
รูปที่ 4.2 แสดงเมนู Advisor	53
รูปที่ 4.3 แสดงเมนู Project Create	53
รูปที่ 4.4 สัญลักษณ์ของส่วนวงจร DIV 1000	54
รูปที่ 4.5 ผลการจำลองการทำงานส่วนของวงจร DIV 1000	54
รูปที่4.6 สัญลักษณ์ของส่วนวงจร LATCH	55
รูปที่ 4.7 ผลการจำลองการทำงานของส่วนวงจร LATCH	55
รูปที่ 4.8 สัญลักษณ์ของส่วนวงจร ONEPULSE	56
รูปที่ 4.9 ผลการจำลองการทำงานของส่วนวงจร ONEPULSE	56
รูปที่ 4.10 สัญลักษณ์ของส่วนวงจร SERIAL_COMMUNICATION	57
รูปที่ 4.11 ผลการจำลองการทำงานของส่วนวงจร SERIAL_COMMUNICATION	57
รูปที่ 4.12 สัญลักษณ์ของส่วนวงจร SELECT_INPUT	58
รูปที่ 4.13 ผลการจำลองการทำงานของส่วนวงจร SELECT_INPUT	58
รูปที่ 4.14 สัญลักษณ์ของส่วนวงจร LFSR	59
รูปที่ 4.15 ผลการจำลองการทำงานของส่วนวงจร LFSR	59
รูปที่ 4.16 สัญลักษณ์ของส่วนวงจร DDS	60
รูปที่ 4.17 ผลการจำลองการทำงานของส่วนวงจร DDS	60
รูปที่ 4.18 สัญลักษณ์ของส่วนวงจร SUNDERLANS	61
รูปที่ 4.19 ผลการจำลองการทำงานของส่วนวงจร SUNDERLANS	61
รูปที่ 4.20 สัญลักษณ์ของส่วนวงจร MUX_OUTPUT	62
รูปที่ 4.21 ผลการจำลองการทำงานของส่วนวงจร MUX_OUTPUT	62
รูปที่4.22 การทดสอบการกำเนิดสัญญาณซายน์ที่ความถี่ต่างๆ	68
รูปที่4.23 การทดสอบการกำเนิดสัญญาณสามเหลี่ยมที่ความถี่ต่างๆ	73
รูปที่4.24 การทดสอบการกำเนิดสัญญาณสี่เหลี่ยมที่ความถี่ต่างๆ	79
รูปที่4.25 การทดสอบการกำเนิดสัญญาณฟันเลื่อยที่ความถี่ต่างๆ	84
รูปที่4.26 แสดงสัญญาณซายน์ที่สร้างจาก SUNDERLAND ที่ความถี่ต่างๆ	93
รูปที่4.27 แสดงสัญญาณสามเหลี่ยมที่สร้างจาก SUNDERLAND ที่ความถี่ต่าง	101
รูปที่4.28 การทดสอบการกำเนิดสัญญาณสุ่ม	101
รูปที่ 4.29 สเปกตรัมองค์ประกอบของสัญญาณซายน์ 10 กิโลเฮิรตซ์	102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ตารางแสดงการเก็บข้อมูลของสัญญาณไซน์ (Sine Wave)	3
ตารางที่ 2.2 ความสัมพันธ์ของบิตเอ็มเอสบีกับควอแดรนท์และสัญญาณซายน์	18
ตารางที่ 3.1 คุณลักษณะต่างๆของชิป FPGA ที่ใช้กับบอร์ดในกลุ่ม POWER ACEX1K SERIES	25
ตารางที่ 3.2 การเก็บค่าขนาด (Amplitude) ในหน่วยความจำ	30
ตารางที่ 4.1 รูปแบบการส่งข้อมูลผ่านทางพอร์ตอนุกรม	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

วงจรสังเคราะห์ความถี่แบบดิจิตอล (Direct digital frequency synthesizer: DDS) ถูกคิดค้นและนำเสนอเป็นครั้งแรกในปี ค.ศ 1971 โดย J. Tierney โดยมีหลักการคือใช้วงจรดิจิตอลสร้างเฟสเพื่อนำมาอ้างอิงของรูปสัญญาณ (Waveform) ที่ต้องการสังเคราะห์ ซึ่งนิยมใช้หน่วยความจำสำหรับเก็บรูปสัญญาณดิจิตอล (Direct pattern) ของสัญญาณที่ต้องการสังเคราะห์ ดัง นั้นในระยะแรกๆงานเกี่ยวกับวงจรสังเคราะห์ความถี่แบบดิจิตอลจะมุ่งเน้นในเทคนิคการบีบอัดข้อมูลในหน่วยความจำเพื่อลดขนาดของหน่วยความจำ แต่ยังคงรักษาคุณภาพของสัญญาณให้อยู่ในระดับที่นำไปใช้งานได้ แต่วงจรเพื่อสังเคราะห์ความถี่แบบดิจิตอลแบบที่ใช้หน่วยความจำมีข้อจำกัดในเรื่องของเวลาในการเข้าถึงหน่วยความจำมากขนาดวงจร และการกินกำลังงาน

เนื้อหาในบทนี้จะเกี่ยวกับทฤษฎีและหลักการทำงานของวงจรสังเคราะห์ความถี่แบบดิจิตอลโดยจะแบ่งออกเป็นสองประเภทหลักๆคือแบบใช้ตารางเปิดดู (Look-up table) และแบบวิธีของซันเดอร์แลนด์ (Sunderland)

1.2 ความมุ่งหมายและวัตถุประสงค์ของปริญญานิพนธ์

ในปริญญานิพนธ์กล่าวถึงวงจรกำเนิดสัญญาณดิจิตอลโดยใช้การอธิบายพฤติกรรมการทำงานของวงจรด้านภาษา VHDL (Very High Speed Integrated Circuit Hardware Description Language) ซึ่งปริญญานิพนธ์นี้มีจุดประสงค์ดังนี้

1.เพื่อศึกษาทฤษฎีและหลักการการคำนวณของวงจรกำเนิดสัญญาณดิจิตอล (Direct Digital Frequency Synthesis)

2.เพื่อศึกษาการเขียนภาษาวีเอชดีแอลในการออกแบบระบบฮาร์ดแวร์ดิจิตอล ซึ่งเริ่มตั้งแต่การออกแบบแก้ไขตรวจสอบ จำลองการทำงาน จนถึงสังเคราะห์วงจร

3. เพื่อศึกษาโปรแกรมต่างๆที่ใช้ในการออกแบบฮาร์ดแวร์ดิจิตอล เช่น MAXPLUS+II

4. เพื่อศึกษาการเชื่อมต่อคอมพิวเตอร์กับเอฟพีจีเอผ่านทางพอร์ตอนุกรม

5. เพื่อศึกษาการใช้งาน MCS-51 ที่ใช้เชื่อมต่อกับเอฟพีจีเอผ่านทางพอร์ตอนุกรม

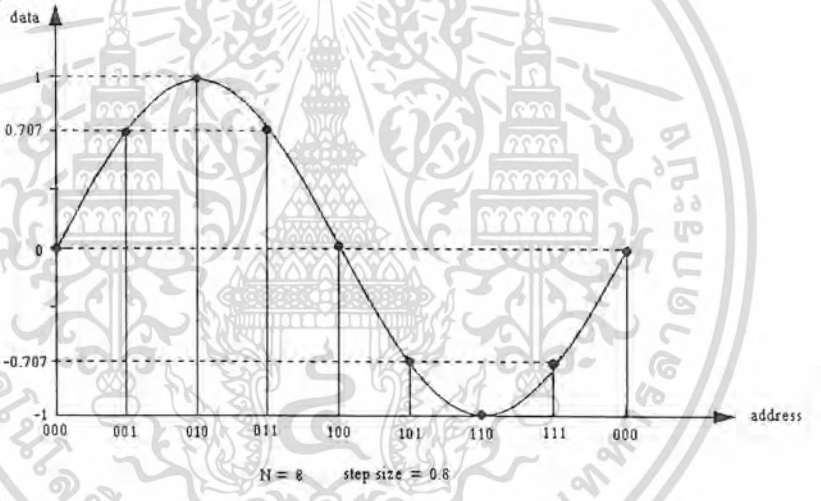
6. สามารถสร้างวงจรกำเนิดสัญญาณซายน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 ทฤษฎีการสังเคราะห์สัญญาณดิจิตอลโดยตรง (Direct Digital Frequency Synthesizer : DDS)

แนวคิดในการสร้างรูปคลื่นแบบไดเร็กดิจิตอลซินธิไซเซอร์หรือ DDS คือการกำเนิดรูปคลื่นโดยอาศัยวิธีการทางดิจิตอลซึ่งในเครื่องกำเนิดความถี่ทั่วๆไปนั้นจะใช้วิธีการทางอนาล็อกเฟสล็อกลูป หรือการกำเนิดรูปคลื่นโดยใช้คริสตอล การสร้างรูปคลื่นแบบไดเร็กดิจิตอลซินธิไซเซอร์จะเก็บข้อมูลของสัญญาณที่จะกำเนิดขึ้นมานั้นให้ครบคาบของสัญญาณซึ่งถ้าหากเรายิ่งเก็บข้อมูลของสัญญาณจำนวนมากเท่าไรก็จะทำให้สัญญาณที่ผลิตออกมาใกล้เคียงกับความเป็นจริงมากเท่านั้น แต่จะมีข้อเสียคือจะสิ้นเปลืองเนื้อที่ในหน่วยความจำมากตามไปด้วย



รูปที่ 2.1 ตัวอย่างการเก็บข้อมูลของสัญญาณไซน์ (Sine Wave) ของ 1 คาบเวลา

ซึ่งการเก็บข้อมูลจะต้องใช้การสุ่ม (Sampling) ข้อมูลบนสัญญาณแต่ละจุดด้วยเวลาที่เท่ากันทุกจุดดังนั้นจุดบนสัญญาณที่ต้องการเก็บคือ 360/จำนวนข้อมูล เช่นต้องการเก็บข้อมูล 1024 ค่า ดังนั้นจะต้องทำการเก็บข้อมูล สามารถทำเป็นตารางได้ดังนี้การเก็บข้อมูลจำนวน N ค่าพิจารณาได้จากตารางที่ 2.1 หลังจากที่เราได้ข้อมูลมาแต่ละจุดแล้ว จะต้องมาทำการจัดค่า (Quantization) ให้มีค่าเป็นทางดิจิตอลเมื่อเราได้ข้อมูลที่มีค่าเป็นเลขฐานสองแล้วก็จะนำค่าข้อมูลเหล่านี้ไปเก็บในหน่วยความจำโดยการจัดเรียงกันไปคือค่าของสัญญาณจุดแรกบนสัญญาณจะถูกเก็บที่ตำแหน่ง (Address) แรกค่าของข้อมูลที่สองจะถูกเก็บในตำแหน่ง (Address) ถัดไปจนครบหมดทุกค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากสัญญาณไซน์แล้วเราสามารถสร้างสัญญาณชนิดอื่นได้อีกมากโดยใช้โปรแกรมคำนวณค่าของสัญญาณต่างๆโดยการคำนวณค่าของสัญญาณแต่ละจุด โดยใช้หลักการเดียวกับการกำเนิดคลื่นไซน์ที่ได้กล่าวมาข้างต้นนั่นเอง

ตารางที่ 2.1 ตารางแสดงการเก็บข้อมูลของสัญญาณไซน์ (Sine Wave)

ข้อมูลที่	เฟส	ข้อมูล
0	$0.\pi/N$	f(0)=sin(0. π /N)
1	$1.\pi/N$	f(1)=sin(1. π /N)
2	$2.\pi/N$	f(2)=sin(2. π /N)
3	$3.\pi/N$	f(3)=sin(3. π /N)
*	*	*
*	*	*
*	*	*
*	*	*
N – 3	(N – 3). π /N	f(N - 3)=sin((N – 3). π /N)
N – 2	(N – 2). π /N	f(N - 2)=sin((N – 2). π /N)
N – 1	(N – 1). π /N	f(N - 1)=sin((N – 1). π /N)

2.2 การเขียนภาษา VHDL

2.2.1 Terminology และ Convention

การเขียนรูปแบบของระบบดิจิตอลด้วยภาษา VHDL นั้น จะมีศัพท์เทคนิคเฉพาะ ฉะนั้นในส่วนนี้จะเป็นการบรรยาย และอธิบายศัพท์บางคำที่จะต้องพบในรายงานชุดนี้

ลักษณะของรูปแบบ (modal styles) : ลักษณะการเขียนรูปแบบ (model) ด้วยภาษา VHDL สามารถแบ่งได้เป็น

- Behavioral Model : หรือที่เรียกอีกอย่างว่า algorithmic description เป็นรูปแบบที่บรรยายพฤติกรรมของระบบดิจิตอล ในส่วนที่บรรยายมีโครงสร้างคล้ายกับภาษาชั้นสูง (high level language) ทั่วไป เช่น PASCAL หรือ C เป็นต้น ในการจำลองการทำงาน (simulation) คำสั่งแต่ละคำสั่ง (statement) จะถูกประเมินผลเป็นไปตามลำดับ (sequential) จากบนลงล่าง ยกเว้นในกรณีของคำสั่ง LOOP หรือการเรียกใช้โปรแกรมย่อยรูปแบบลักษณะนี้จะไม่ให้รายละเอียดที่เกี่ยวกับการผลิต หรือโครงสร้างของ Hardware แต่ในทางตรงข้ามที่รายละเอียดเกี่ยวกับความสัมพันธ์ระหว่าง input กับ output ที่ดี

- Dataflow Model : เรียกอีกอย่างหนึ่งได้ว่า " Register transfer level " (RTL) เป็นรูปแบบที่ถูกเขียนขึ้น เพื่อจุดประสงค์ที่จะใช้เป็นเครื่องมือสำหรับสังเคราะห์วงจรอัตโนมัติ รูปแบบลักษณะนี้ส่วนใหญ่จะเป็น procedural constructs และ functional operators

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Structural Model : เป็นรูปแบบที่แสดงการเชื่อมต่อกันระหว่างอุปกรณ์ต่างๆ ที่ประกอบกันขึ้นเป็นวงจรหรือระบบดิจิตอล และสามารถเรียกอีกอย่างได้ว่า " netlist representation" เป็นการเขียนที่แสดงให้เห็นโครงสร้างของ hardware

- Mixed – Level Model : จากคุณสมบัติที่อ่อนตัวของภาษา VHDL จึงสามารถที่จะเขียนรูปแบบโดยใช้ลักษณะต่างๆ บรรยายวงจรหรือระบบดิจิตอลเดียวกันได้ ฉะนั้นรูปแบบเช่นนี้จึงมีการเขียนแบบผสม

Concurrency ในภาษา VHDL นั้น ชุดคำสั่งจะทำงานในเวลาเดียวกันและอิสระต่อกัน ลักษณะเช่นนี้เป็นคุณสมบัติที่เป็นความจริงทางฟิสิกส์ของวงจรอิเล็กทรอนิกส์ ชุดคำสั่งนี้เรียกว่า "concurrent statement " และจะทำงานก็ต่อเมื่อมีการเปลี่ยนแปลงค่าของสัญญาณ

Sequential : นอกจากความสามารถที่ชุดคำสั่งจะทำงานแบบ concurrent แล้วบางครั้งการเขียนรูปแบบในลักษณะที่บรรยายพฤติกรรมของวงจร มีความจำเป็นที่จะต้องให้ชุดคำสั่งทำงานเป็นลำดับขั้นเรียงกันจากบนลงล่าง อย่างเช่นการเขียนแบบ behavioral model เป็นต้น ชุดคำสั่งที่เป็น sequential นี้จะใช้ในโปรแกรมย่อย (subprogram)และ process statement

Driver : สัญญาณต่างๆ (signal) ใน VHDL นั้นจะถูกควบคุมด้วยตัวขับหรือ " driver " สัญญาณเหล่านี้จะรับค่าใหม่ (ระดับของสัญญาณ) ได้ด้วยตัวขับนี้เอง

Transaction :การเกิด transaction กับ signal นั้นจะเกิดขึ้นเมื่อมีการกำหนดค่าๆหนึ่งให้กับ signal signal นั้น ค่าใหม่ที่ signal ได้รับอาจจะมีผลหรือไม่มีผลทำให้เกิดการเปลี่ยนแปลงของระดับสัญญาณ (event) เช่นการเปลี่ยนจากค่า logic '0' เป็นค่า logic '0' เป็นต้น

Event : คือการเปลี่ยนระดับค่าของ SIGNAL จากระดับหนึ่งไประดับอื่น อย่างเช่น ในระบบดิจิตอลการเปลี่ยนจาก Logic '0' เป็นค่า Logic '1' หรือในทางตรงกันข้ามถือว่า SIGNAL นั้นเกิด "event" ฉะนั้นจะเห็นได้ว่า การที่จะเกิด event ได้นั้นจะต้องเกิด transaction ไม่จำเป็นต้องเกิด event ทุกครั้ง

Sensitivity List : คือรายชื่อของ signal ต่างๆ ที่มีผลทำให้เกิดการทำงานของ concurrent statement เมื่อเกิด event ขึ้นกับ signal ตัวใดตัวหนึ่งหรือหลายตัวพร้อมกันในรายชื่อนั้น

- Object : ในภาษา นั้นคำว่า ใช้เขียนเพื่อบ่งบอกถึงองค์ประกอบส่วนหนึ่งของรูปแบบ ซึ่งเปรียบได้เหมือนกับภาชนะที่มีไว้สำหรับบรรจุค่าต่างๆ สามารถแบ่งออกได้เป็นสามชั้น (class) ด้วยกันคือ

CONSTANT : ได้แก่ object ประเภทหนึ่งที่เมื่อกำหนดค่าเริ่มต้นให้แล้วจะคงค่านั้นไว้ตลอด ไม่สามารถดัดแปลงหรือแก้ไขได้ สามารถประกาศใช้ได้ในส่วนประกาศต่างๆ ของรูปแบบ (model)

SIGNAL : หมายถึง object ประเภทหนึ่งที่สามารถกำหนดค่าที่สัมพันธ์กับเวลาให้ได้นั้น หมายความว่าSIGNAL สามารถรับค่าได้เพียงค่าเดียวเท่านั้นในขณะเวลาหนึ่ง SIGNAL จะรับค่าๆหนึ่งได้จากขับสัญญาณหรือ driver ซึ่งตัวขับนี้อาจจะเก็บค่าในอนาคตสำหรับ SIGNAL ไว้ด้วย SIGNAL สามารถประกาศใช้ได้ในส่วนที่เป็น sequential body เท่านั้น ดังนั้น SIGNAL จึงสามารถถูกนำไปใช้ตลอดโครงสร้างของรูปหรือที่เรียกว่า global object

VARIABLE : หรือตัวแปรได้แก่ object ที่สามารถกำหนดค่าใด ๆ ให้ได้และสามารถที่จะเปลี่ยนแปลงค่าได้ตลอดการจำลองการทำงาน แต่จะเก็บค่าเพียงค่าเดียวเท่านั้นในขณะเวลาหนึ่ง เนื่องจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VARIABLE สามารถประกาศใช้ได้ในส่วนที่เป็น sequential body เท่านั้นอันได้แก่ส่วนประกอบของ PROCESS, FUNCTION หรือ PROCEDURE ดังนั้น VARIABLE จึงสามารถนำไปใช้ได้เฉพาะในขอบเขตที่ถูกประกาศใช้เท่านั้น (local object)

- ประเภทของ object ที่กำหนดไว้แล้ว (predefined type) : ได้แก่ TYPE ที่กำหนดไว้ใน package ชื่อ STANDARD และกำหนดโดย IEEE ว่าจะต้องมีในระบบที่ใช้พัฒนา VHDL ฉะนั้นจึงไม่จำเป็นต้องประกาศใช้ในทุกรูปแบบที่เขียนขึ้น TYPE ประเภทนี้ได้แก่

  1) BOOLEAN คือกลุ่มของค่า FALSE และ TURE
  2) BIT คือกลุ่มของค่า '0' และ '1'
  3) INTEGER คือกลุ่มของค่า –214748347 ถึง 214748347
  4) PEAK คือกลุ่มของค่า -1.0E38 ถึง 1.5E38
  5) CHARACTER คือกลุ่มของค่าพยัญชนะ 'A' – 'Z', 'a' – 'z' อักษรหรือเครื่องหมายพิเศษและตัวอักษรควบคุม
  6) TIME ได้แก่หน่วยเวลาที่มีค่าพื้นฐานเป็นวินาที (second ย่อด้วย s หรือ S)
  7) SEVERITY LEVEL คือกลุ่มของค่า NOTE, WARING, ERROR, FAILURE ส่วนต่างในการเขียน VHDL

การเขียนรูปแบบหรือ modeling ด้วยภาษา VHDL มีความจำเป็นที่จะต้องแนะนำให้รู้จักกับส่วนต่าง ๆ ของแบบ (design units) ที่ใช้ภาษาเสียก่อน และนี่ก็เป็นขั้นตอนแรกที่สำคัญที่สุดของการศึกษาศึกษาเรียนรู้การใช้ภาษา VHDL เขียนรูปแบบบรรยายระบบดิจิตอลในมุมมองของการออกแบบลักษณะ Top-Down Design นอกจากนั้นการที่จะเข้าใจในเรื่องของโครงสร้าง และส่วนต่าง ๆ ของรูปแบบ VHDL ให้ถูกต้องเสียก่อน

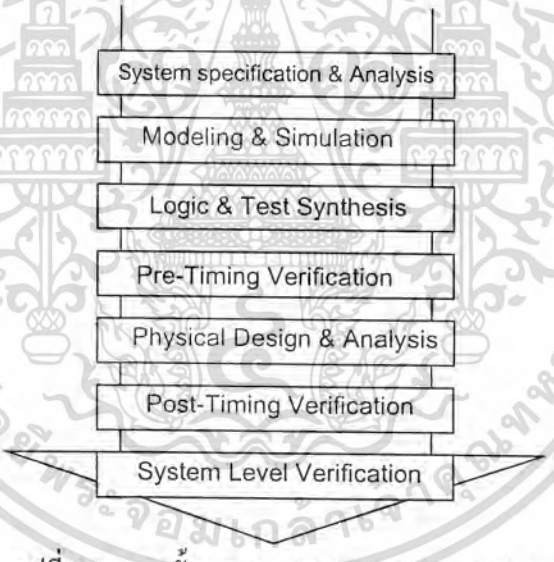
การออกแบบวงจรเชิงเลข (Digital Circuit) นั้น ในปัจจุบันก้าวหน้าไปอย่างมากโดยการใช้ภาษาบรรยายการทำงานของวงจร (Hardware Description Language : HDL) ซึ่งเป็นภาษาที่ใช้สำหรับออกแบบฮาร์ดแวร์ โดยภาษาที่เป็นมาตรฐานสากลเช่น Verilog หรือ VHDL (VHSIC Hardware Description Language (VHSIC : Very High Speed Integrated Circuit)) หรือภาษาที่ไม่เป็นมาตรฐานเช่น AHDL (Altera Hardware Description Language) หรือ PHDL (Philips Hardware Description Language) เป็นต้น มาบรรยายการทำงานของวงจรที่ได้ออกแบบไว้ ซึ่งในปริญานิพนธ์นี้ได้ใช้ภาษา VHDL มาทำการออกแบบวงจร Digital Oscillator ทำให้ลดความยุ่งยากในการนำเอาอุปกรณ์มาเชื่อมต่อให้เป็นวงจร รวมทั้งลดเวลาที่ใช้ในการออกแบบและทดสอบการทำงาน ซึ่งมีความแตกต่างเป็นอย่างมากเมื่อเปรียบเทียบกับการออกแบบในอดีตที่ผ่านมา คือผู้ออกแบบจะต้องนำเอาอุปกรณ์แต่ละตัวที่ทำการออกแบบไว้ มาทำการต่อทดลองในแผงวงจรจริง และทำการทดสอบวงจรเพื่อหาข้อผิดพลาด ซึ่งต้องใช้เวลานานกับการแก้ปัญหาแต่ละอย่างที่เกิดขึ้น แต่ในการออกแบบด้วยภาษา VHDL ผู้ออกแบบเพียงแต่เขียนซอสโค๊ด (Source Code) บรรยายการทำงานของวงจร หลังจากนั้นก็ทำการคอมไพล์ (Compile) แล้วจำลองการทำงาน (Simulate) ดูว่าได้ฟังก์ชั่นการทำงานและไทม์มิ่ง (Timing)ตามที่ต้องการหรือไม่จากนั้นก็นำซอสโค๊ดที่ได้ไปทำการสังเคราะห์ด้วยโปรแกรมสังเคราะห์ (Synthesis Tool) สุดท้ายนำวงจรที่ได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการแม็ป (Map) ลงไปยัง FPGA (Field Programmable Gate Array) เพื่อเป็นชิป (Chip) ต้นแบบสำหรับการนำไปทดสอบการทำงาน

2.2.2 การออกแบบจากบนลงล่าง

ในการพัฒนาวงจรรวมเชิงเลขขนาดใหญ่ที่มีความซับซ้อน ผู้ออกแบบมักจะมองการออกแบบให้อยู่ในรูปของบล็อกไดอะแกรมก่อน จากนั้นจึงวิเคราะห์ให้ลึกถึงรายละเอียดต่อไป ซึ่งภาษา VHDL นั้นอนุญาตให้อธิบายการทำงานของแต่ละบล็อก และวิเคราะห์การทำงาน แก้ไขและปรับปรุงการทำงานจากผลที่วิเคราะห์ เพื่อให้ได้การทำงานตามที่ต้องการ โดยการออกแบบในลักษณะนี้เรียกว่า หลักการออกแบบจากบนลงล่าง (Top-Down Design) ซึ่งถ้าเปรียบเทียบกับการออกแบบจากล่างขึ้นบน (Bottom-Up Design) จะเห็นได้ว่าการออกแบบจากล่างขึ้นบนจะใช้เวลาในการออกแบบมากกว่าเพราะเป็นการวาดวงจรด้วยอุปกรณ์ต่างๆ (Schematic Capture) ที่ประกอบกันเข้าเป็นวงจรที่ต้องการออกแบบ จำลองการทำงาน ตรวจสอบความถูกต้อง ซึ่งใช้เวลามาก และถ้าวงจรที่ต้องการออกแบบมีความซับซ้อนก็จะเป็นเรื่องที่ยากมากในการออกแบบลักษณะนี้ ดังนั้นการใช้ภาษา VHDL กับหลักการออกแบบจากบนลงล่างจึงเป็นวิธีการที่เหมาะสมสำหรับการออกแบบและพัฒนาวงจรที่มีความซับซ้อนมากขึ้น ทั้งยังช่วยลดเวลาและค่าใช้จ่ายในการออกแบบ



รูปที่ 2.2 แสดงขั้นตอนการออกแบบจากบนลงล่าง

จากรูปที่ 2.2 แสดงให้เห็นถึงขั้นตอนการออกแบบจากบนลงล่างทั้งนี้ในทางปฏิบัติอาจจะมีข้อแตกต่างไปจากนี้บ้างเล็กน้อย โดยขั้นตอนของการออกแบบจากบนลงล่างมีรายละเอียด ดังนี้

1. ขั้นตอนการสร้างข้อกำหนดของความต้องการ และวิเคราะห์ระบบ เพื่อหาแนวความคิดและ หลักการ (Idea and Concept) ในการแก้ปัญหา

2. ขั้นตอนการเขียนรูปแบบของระบบที่ต้องการออกแบบโดยใช้ภาษา VHDL สำหรับ บรรยาย พฤติกรรมการทำงาน พร้อมทั้งจำลองการทำงาน เพื่อเปรียบเทียบและตรวจสอบความถูกต้องกับข้อกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.ขั้นตอนการสังเคราะห์ซึ่งจะต้องทำการกำหนดเทคโนโลยีที่จะมารองรับวงจรที่ออกแบบและระบบช่วยออกแบบจะทำการสังเคราะห์วงจรที่ได้จากรูปแบบที่เขียนขึ้นให้อยู่ในรูปของวงจรที่ประกอบด้วยอุปกรณ์อิเล็กทรอนิกส์ หรือวงจรในระดับเกท (Gate Level) และการเชื่อมต่อกันของอุปกรณ์เหล่านั้น หรือไม่ก็อยู่ในรูปของเน็ทลิสต์ (Net list) ที่สามารถนำไปผลิตลงบนอุปกรณ์อื่นได้

4. หลังจากการสังเคราะห์วงจรให้อยู่ในระดับเกทหรือเน็ทลิสต์แล้ว ข้อมูลที่ได้นอกจากจะเป็นข้อมูลสำหรับจำลองการทำงานในเรื่องของความถูกต้องของฟังก์ชั่นแล้ว ยังมีข้อมูลที่ เกี่ยวกับเวลาด้วย ซึ่งจากความจริงที่ว่า อุปกรณ์อิเล็กทรอนิกส์ทุกชิ้นจะมีเวลาหน่วงของ การเคลื่อนผ่าน (Propagation Delay time) เสมอ ถึงแม้ว่าจะเป็นเวลาที่น้อยมากในระดับนาโนวินาที แต่ถ้าภายในวงจรหนึ่งประกอบด้วยเกทของฟังก์ชั่นต่างๆจำนวน 10,000 เกท ขึ้นไป เวลาดังกล่าวนี้จะสะสมกันมากขึ้นจนอาจจะทำให้การทำงานของวงจรทั้งหมดผิดไป หรือไม่สามารถทำงานในย่านความถี่สัญญาณนาฬิกาสูงๆได้

5. ขั้นตอนของการผลิตเป็นวงจรจริง (Technology and Device Mapping) โดยนำข้อมูลที่ได้ จากการสังเคราะห์มาผลิต ซึ่งอาจจะอยู่ในรูปของอุปกรณ์ FPGA หรือวงจรรวม ASIC

6. หลังจากที่ได้วงจรจริงมาแล้วยังต้องมีความจำเป็นที่จะต้องตรวจสอบการทำงานที่คำนึงถึงเวลาด้วยเพื่อความถูกต้องของวงจรครั้งสุดท้ายก่อนที่จะนำไปรวมเข้ากับอุปกรณ์อื่นๆให้เป็นระบบ เพราะในขั้นตอนนี้วงจรที่ออกแบบจะประกอบด้วยอินพุทและเอาท์พุทแพด (Pad) ซึ่งเป็นจุดต่อสำหรับรับและส่งสัญญาณกับภายนอก

7. หลังจากที่นำวงจรที่ออกแบบรวมเข้ากับอุปกรณ์อื่นๆให้เป็นระบบแล้วนั้น จะต้องทดสอบการทำงานรวมทั้งระบบร่วมกับอุปกรณ์อื่นๆอีกครั้ง ซึ่งเป็นการทดสอบการทำงานจริงขั้นสุดท้าย

2.2.3 ภาษา VHDL และ ส่วนประกอบต่างๆของภาษา

วิวัฒนาการของภาษา VHDL นั้นเริ่มต้นประมาณปี ค.ศ. 1981 โดยที่กระทรวงกลาโหมสหรัฐอเมริกา หรือ DOD (Department of Defence) ได้ทำการพัฒนาโครงการที่มีชื่อว่า VHSIC ซึ่งเป็นการพัฒนาโปรแกรมซึ่งจัดเป็นภาษาระดับสูงเช่นเดียวกับภาษาC หรือ Pascal แต่สามารถบรรยายพฤติกรรมการทำงานของวงจรเชิงเลข หรือโครงสร้างของวงจรได้ ทั้งนี้เพื่อให้สามารถออกแบบและสร้างวงจรรวมได้รวดเร็วขึ้น

ในการเขียนรูปแบบบรรยายระบบเชิงเลขในลักษณะของการออกแบบจากบนลงล่างจะต้องทำความเข้าใจในเรื่องของโครงสร้างและส่วนประกอบต่างๆของรูปแบบภาษา VHDL เสียก่อนซึ่งส่วนประกอบที่สำคัญและเป็นพื้นฐานของการเขียนมี 4 หน่วย คือ

- หน่วยการออกแบบเอนทิตี (Entity Design unit)
- หน่วยการออกแบบสถาปัตยกรรม (Architecture Design unit)
- หน่วยการออกแบบแพ็กเก็ต (Package Design unit)
- หน่วยการออกแบบโครงแบบ (Configuration Design unit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.1 หน่วยการออกแบบเอนทิตี

หน่วยการออกแบบนี้เป็นส่วนที่ใช้สำหรับติดต่อระหว่างภายนอกกับรูปแบบที่เขียนขึ้นโดยเป็นการกำหนดจุดเชื่อมต่อของรูปแบบ กำหนดทิศทางการไหลของสัญญาณ และประเภทของค่าที่สามารถกำหนดให้กับสัญญาณตามจุดต่างๆของข้อมูลที่ไหลผ่านจุดต่อเหล่านั้น รูปที่ 2.3 แสดงให้เห็นถึงโครงสร้างของหน่วยการออกแบบเอนทิตี

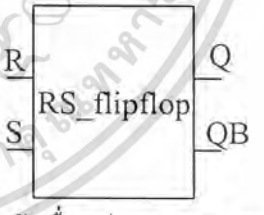
```
Entity component_name is
    Input and Output ports
    Physical and other parameters
End [component_name];
```

รูปที่ 2.3 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี้

ส่วนนี้จะขึ้นต้นด้วยคำว่า Entity และ is ระหว่างคำทั้งสองคำเป็นส่วนสำหรับชื่อของรูปแบบที่ต้องการจะเขียน(component_name)หลังจากนั้นจะตามด้วยส่วนที่ใช้กำหนดช่องทางเข้าและออกของข้อมูล(input-output) รวมทั้งพารามิเตอร์อื่นๆและที่สำคัญคือหน่วยการออกแบบเอนทิตีจะต้องปิดท้ายด้วยคำว่าEndและเครื่องหมายอัฒภาคเสมอ (;)

```
Entity RS_ff is
    port(S,R : in bit;
         Q,QB : out bit);
End RS_ff;
```

(a) หน่วยการออกแบบเอนทิตีในรูปของภาษา VHDL



(b) มุมมองของตัวเชื่อมประสาน (Interfacing)

รูปที่ 2.4 แสดงรูปแบบของ RS_flipflop

ในรูปที่ 2.4 เป็นหน่วยการออกแบบเอนทิตีที่บรรยายอุปกรณ์ชื่อ RS_flipflop ในส่วนหัวของเอนทิตีมีการกำหนดจุดต่อ 4 จุด ภายใต้ชุดคำสั่ง portโดยที่ 2 จุดแรกเป็นจุดให้ข้อมูลไหลผ่านเข้า ได้แก่ R, S ซึ่งกำหนดด้วยทิศทางการติดต่อกับโลกภายนอกเป็นการไหลเข้าของข้อมูล (in) ส่วนจุดเอาท์พุทเป็นจุดให้ข้อมูลไหลออก ได้แก่ Q,QB ซึ่งกำหนดด้วยทิศทางการติดต่อกับภายนอกเป็นการไหลออก (out) ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภทของข้อมูลที่จะไหลเข้าและออกนั้นเป็นประเภท bit ที่สามารถมีค่าได้เพียงสองค่าเท่านั้น คือ "0" และ "1" เท่านั้น

2.2.3.2 หน่วยการออกแบบสถาปัตยกรรม

หน่วยการออกแบบสถาปัตยกรรมคือส่วนที่ใช้เขียนบรรยายพฤติกรรมของรูปแบบในมุมมองของการจำลองการทำงาน พฤติกรรมต่างๆที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้าและออกตรงช่องทาง ตลอดจนพารามิเตอร์ต่างๆที่กำหนดในหน่วยการออกแบบเอนทิตี รูปที่ 2.5 แสดงให้เห็นถึงโครงสร้างของหน่วยการออกแบบสถาปัตยกรรม

```
Architecture identifier of component_name is
        [declaration]
    Begin
        Specification of the functionality
        of the component in terms of its
        input lines and as influenced by
        physical and other parameters
End [identifier];
```

รูปที่ 2.5 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม

ส่วนของหน่วยการออกแบบสถาปัตยกรรมเริ่มต้นด้วยคำว่า Architecture และตามด้วยชื่อ (identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่งที่แสดงให้เห็นว่า Architecture นั้นใช้บรรยายหน่วยการออกแบบเอนทิตีใดๆ (of <entity design unit> is) ส่วนที่อยู่ระหว่าง Architecture และ Begin เป็นพื้นที่ส่วนประกาศหน่วยของสถาปัตยกรรมกำหนด (Architecture declaration area) ที่เป็นส่วนเพื่อเลือก (Option) ในบริเวณนี้สามารถใช้เขียนประกาศกำหนดค่าต่างๆที่จะนำไปใช้ภายในสถาปัตยกรรมนั้นได้ อาทิเช่น ประเภท (Type) ต่างๆ (ตัวอย่างเช่น bit, bit_vector), สัญญาณ (signal), ค่าคงที่ (constant), โปรแกรมย่อย (ได้แก่ function และ procedure) และอุปกรณ์ (component) ส่วนที่ใช้บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้าและไหลออกของรูปแบบ (สัญญาณที่กำหนดในชุดคำสั่ง port) นั้นจะถูกบรรยายในบริเวณเนื้อที่ระหว่างคำว่า Begin กับ End ของหน่วยการออกแบบสถาปัตยกรรม และนอกจากนั้นชุดคำสั่งทุกคำสั่งที่อยู่ภายในบริเวณนี้จะเป็นชุดคำสั่งแบบแข่งขนาน (Concurrent statement) เท่านั้น คือทุกๆ statement จะทำงานพร้อมกัน ลำดับก่อนหลังจะ ไม่มีผลต่อการทำงานของรูปแบบ หน่วยการออกแบบสถาปัตยกรรมจะต้องปิดท้ายด้วยคำสั่ง End และชื่อของสถาปัตยกรรมนั้นๆ โดยทั่วไปการเขียนรูปแบบระบบเชิงเลขด้วยภาษา VHDL สามารถเขียนได้ในลักษณะต่างๆ ดังนี้

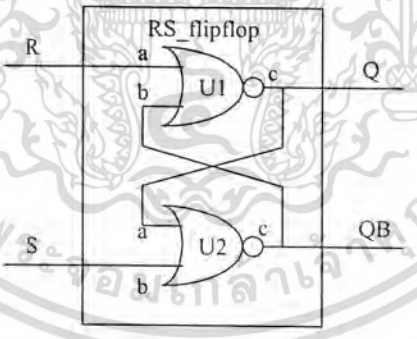
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ลักษณะการไหลของข้อมูล (Dataflow style)
- ลักษณะพฤติกรรม (Behavioral style)
- ลักษณะโครงสร้าง (Structural style)
- ลักษณะผสม (Mixed Model style)

```
Architecture dataflow of RS_ff is
        Begin
                Q  <=  not(QB or R);
                QB <= not(Q or S);
End dataflow;
```

รูปที่ 2.6 แสดงหน่วยการออกแบบสถาปัตยกรรมของ RS_flipflop ตามฟังก์ชั่นบูลีน $Q = \overline{QB + R}$ และ $QB = \overline{Q + S}$

รูปที่ 2.6 ส่วนที่บรรยายความสัมพันธ์ระหว่างข้อมูลที่ไหลเข้า (R, S) กับข้อมูลที่ไหลออก (Q, QB) ประกอบด้วยชุดคำสั่งแบบแข่งขนาน 2 ชุด ซึ่งเขียนเป็นประเภทการไหลของข้อมูล หรือเรียกว่า ระดับการถ่ายโอนข้อมูลระหว่างรีจิสเตอร์ (RTL : Register Transfer Level)



รูปที่ 2.7 แสดงโครงสร้างภายในสถาปัตยกรรมของ RS_flipflop

รูปที่ 2.7 เป็นหน่วยการออกแบบสถาปัตยกรรมของ RS_flipflop ในลักษณะโครงสร้างซึ่งเปรียบเสมือนการนำอุปกรณ์ที่มีอยู่ในไลบรารี่ (Library) มาต่อเป็นวงจรตามต้องการโดยใช้นอร์เกท 2 อินพุท (nor2) จำนวนสองตัวมาสร้างตามฟังก์ชั่นบูลีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Architecture struc of RS_ff is
        component nor2
            port(a,b : in bit;
                 c :out bit);
        end component;

    Begin
        U1 : nor2 port map(R,QB,Q);
        U2 : nor2 port map(S,Q,QB);
End struc;
```

รูปที่ 2.8 แสดงหน่วยการออกแบบสถาปัตยกรรมของ RS_flipflop ในลักษณะโครงสร้าง

```
Architecture behave of RS_ff is
    Begin
        process(R,S)
            begin
                if R='0' and S='1' then
                Q  <= '1';
                QB <= '0';
                elsif R='1' and S='0' then
                Q  <= '1';
                QB <= '0';
                end if;
        end process;
End behave;
```

รูปที่ 2.9 แสดงหน่วยการออกแบบสถาปัตยกรรมของ RS_flipflop ในลักษณะพฤติกรรม

รูปที่ 2.9 เป็นการเขียนบรรยายการทำงานของรูปแบบในลักษณะพฤติกรรม ซึ่งจะเห็นได้ว่ามีลักษณะที่เหมือนกับการเขียนโปรแกรมทั่วไป โดยจะต้องมีการใช้งานส่วนที่เรียกว่า process และการทำงานของรูปแบบจะขึ้นอยู่กับการเปลี่ยนแปลงของสิ่งที่อยู่ภายใน process (อินพุท R, S) ซึ่งเรียกว่า Sensitivity list การเขียนในลักษณะนี้ลำดับก่อนหลังของชุดคำสั่งจะมีผลต่อการทำงานของรูปแบบที่เขียนขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Architecture mixed of RS_ff is
        component nor2
                port(a,b : in bit;
                        c : out bit);
        end component;

Begin
        U1 : nor2 port map(R,QB,Q);
        QB <= not(Q or S);
End mixed;
```

รูปที่ 2.10 แสดงหน่วยการออกแบบสถาปัตยกรรมของ RS_flipflop ในลักษณะผสม

ไม่ว่าจะเขียนบรรยายส่วนของสถาปัตยกรรมของ RS_flipflop ในลักษณะของพฤติกรรม การไหลของข้อมูล โครงสร้าง หรือผสมที่นำเอาแต่ละลักษณะมาเขียนไว้ในส่วนของสถาปัตยกรรมก็ตามต่างก็มีพฤติกรรมเดียวกัน และจะให้ผลลัพธ์จากการจำลองการทำงานที่เหมือนกัน ซึ่งถือว่าเป็นข้อดีของภาษา VHDL

2.2.3.3 หน่วยการออกแบบแพ็กเก็ต

ข้อมูลต่างๆ ตลอดจนโปรแกรมย่อยที่เป็นประโยชน์ต่อการเขียนรูปแบบบรรยายระบบเชิงเลขสามารถเก็บไว้ในส่วนของแพ็กเก็ตได้และข้อมูลเหล่านี้สามารถเรียกไปใช้ได้โดยหน่วยการออกแบบเอนทิตีหน่วยการออกแบบสถาปัตยกรรมหรือจากหน่วยการออกแบบแพ็กเก็ตอื่นๆ โดยปกติแล้วแพ็กเก็ตจะแบ่งออกเป็น 2 ส่วน คือ การประกาศแพ็กเก็ต (Package Declaration) และส่วนของบอดี้แพ็กเก็ต (Package Body) เนื่องจากแพ็กเก็ตถูกสร้างขึ้นเป็นส่วนแยกต่างหากออกจากรูปแบบที่กำลังเขียนอยู่ ฉะนั้นการที่จะนำแพ็กเก็ตไปใช้นั้นจะต้องมีการเชื่องโยงหรืออ้างอิงเสียก่อน ซึ่งในภาษา VHDL สามารถกระทำได้ด้วยชุดคำสั่ง USE

Package Declaration

ส่วนที่มีความสำคัญที่สุดของแพ็กเก็ต(ถ้ามองในแง่ของการนำไปใช้จากภายนอก)ได้แก่ส่วนการประกาศแพ็กเก็ต เพราะจะเป็นส่วนที่กำหนดชื่อของสิ่งที่ประกาศอยู่ภายในแพ็เก็ตสำหรับนำไปใช้ภายนอกตัวของแพ็กเก็ตเองสิ่งใดๆที่ถูกประกาศไว้ในส่วนของบอดี้แพ็กเก็ตแต่ไม่ได้ถูกประกาศไว้ในส่วนการประกาศแพ็กเก็ตจะไม่สามารถถูกนำค่าและพฤติกรรมไปใช้ส่วนนอกได้ ซึ่งสามารถเปรียบเทียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ดับสิ่งที่ประกาศไว้ในส่วนของการประกาศเอนทิตีคือจุดเชื่อมต่อหรือพอร์ทที่มีหน้าที่ติดต่อกับโลกภายนอกฉะนั้นโดยทั่วไปแล้วแพ็กเก็ตสามารถสร้างขึ้นได้โดยไม่จำเป็นต้องมีส่วนบอดี้และยังสามารถถูกนำไปใช้จากรูปแบบภายนอกได้ เช่น ใช้สำหรับประกาศชนิด (Type) หรือสัญญาณเช่นเดียวกันกับส่วนบอดี้แพ็กเก็ตที่ไม่จำเป็นต้องมีส่วนของการประกาศแพ็กเก็ต แต่แพ็กเก็ตนั้นจะไม่สามารถถูกนำไปใช้จากรูปแบบอื่นได้

```
Package package_name is
   Package_declaration_part
End package_name;
```

รูปที่ 2.11 แสดงโครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต

Package body

โครงสร้างที่ประกอบด้วยคำสั่งต่างๆในรูปของคำสั่งลำดับ (Sequence)ที่ใช้บรรยายฟังก์ชั่นการทำงานของโปรแกรมย่อย(Subprogram)ทั้งหลายที่ชื่อของโปรแกรมย่อยนั้นๆที่ถูกประกาศไปในส่วนของการประกาศแพ็กเก็ตแล้วจะถูกเก็บไว้ในส่วนบอดี้แพ็กเก็ตทั้งนี้รวมทั้งการกำหนดค่าคงที่ต่างๆอันได้แก่ค่าคงที่ที่ถูกประกาศชื่อก่อนในส่วนของการประกาศแพ็กเก็ตแต่ถูกกำหนดค่าในส่วนของบอดี้แพ็กเก็ต ฉะนั้นส่วนบอดี้แพ็กเก็ตจึงไม่จำเป็นต้องมีถ้าในส่วนของการประกาศแพ็กเก็ตไม่มีการประกาศชื่อที่เป็นโปรแกรมย่อยหรือค่าคงที่ การเขียนบอดี้แพ็กเก็ตนั้นเป็นไปตามกฎเกณฑ์ที่แสดงในรูปที่ 2.12

```
Package body package_name is
   declarative part
End package_name;
```

รูปที่ 2.12 แสดงโครงสร้างโดยทั่วไปของบอดี้แพ็กเก็ต

2.2.3.4 หน่วยการออกแบบโครงแบบ

ดังที่ทราบกันแล้วว่ารูปแบบหนึ่งของระบบดิจิตอล ไม่ว่าจะเป็นอะไร จะมีหน่วยการออกแบบเอนทิตีได้ เพียงหนึ่งหน่วยเท่านั้น แต่ในขณะที่ หน่วยการออกแบบเอนทิตี หนึ่งหน่วยนี้อาจจะมีสถาปัตยกรรม ที่เป็นหน่วยรอง ได้หลายหน่วย ดังนั้นจะต้องมีหน่วยการออกแบบโครงแบบมาเพื่อกำหนดการใช้โครงแบบ (Configuration) ประกอบเอนทิตีกับหน่วยการออกแบบสถาปัตยกรรมหน่วยไหนเข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Configuration identifier of entity_name is
        Configuration declarative_part
End;
```

รูปที่ 2.13 แสดงโครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ

2.2.4 ชุดคำสั่งลำดับ (Sequential Statements)

ภาษา VHDL สามารถใช้เขียนรูปแบบ (modeling) บรรยายระบบดิจิตอลในลักษณะของ behavioral description ที่โครงสร้างภายในประกอบด้วย sequential statement การศึกษาในรายละเอียดของโครงสร้างดังกล่าว สำหรับ software engineering ที่มีความคุ้นเคยกับการเขียน โปรแกรมด้วยภาษาชั้นสูง อาทิเช่น C หรือ PASCAL อยู่ก่อนแล้ว จะสามารถเข้าใจโครงสร้างแบบ sequential ได้ง่าย เพียงแต่ต้องทำความเข้าใจเกี่ยวกับลักษณะการทำงานของ hardware เพิ่มเติม ในภาษา VHDL มีคำสั่งดังต่อไปนี้

- WAIT statement
- VARIABLE assignment
- Signal assignment
- IF-THEN-ELSE statement
- CASE statement
- Loops
- NEXT statement
- EXIT statement
- RETURN statement
- NULL statement
- Procedure call
- ASSERTION statement

จะกล่าวเฉพาะ statement ที่สำคัญๆ เพื่อความเข้าใจในการทำงานแบบ sequential เท่านั้น ตามที่เคยกล่าวมาแล้วว่าภาษา VHDL เป็นภาษาที่มีคุณสมบัติเป็นแบบแข่งขนานนั้น คือ ชุดคำสั่งภายในตัวโครงสร้างจะเป็นชุดคำสั่งแบบแข่งขนาน เช่นเดียวกับภาษา ADA ชุดคำสั่งลำดับหรือ sequential statement ที่เรียกว่า อันได้แก่ process statement

2.2.4.1. Process Statement

หัวใจสำคัญของ Concurrent shell ที่ทำให้สามารถเขียน VHDL model เพื่อบรรยายพฤติกรรมของระบบดิจิตอลอิเล็กทรอนิกส์ในลักษณะ behavioral description ได้แก่คำสั่ง process ที่โครงสร้างภายในจะประกอบด้วยชุดคำสั่งแบบลำดับเท่านั้น ชุดคำสั่งเหล่านี้จะทำงานเป็นลำดับจากบนลงล่าง เมื่อ PROCESS ถูกกระตุ้นให้ทำงาน

เอกสารนี้มีไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำ PROCESS ในบรรทัดแรกของโครงสร้าง แสดงถึงจุดเริ่มต้นของชุดคำสั่ง process ในบางกรณีใน architecture หนึ่งอาจจะมีโครงสร้างของชุดคำสั่ง process หลายชุดได้ คำ END PROCESS บอกถึงจุดสิ้นสุดของชุดคำสั่ง process เมื่อคำสั่ง END PROCESS ถูกปฏิบัติแล้วชุดคำสั่ง process จะหยุดการทำงานลงชั่วขณะ (แต่ยังคง active อยู่ตลอดเวลา) จนกว่าจะมีสัญญาณอย่างน้อยตัวหนึ่งใน sensitivity list เกิด event ขึ้นอีก ชุดคำสั่ง process เป็นชุดคำสั่งแบบแข่งขนาน นั้นหมายความว่า โดยปกติแล้วชุดคำสั่ง process จะทำงานตลอดเวลา การที่ชุดคำสั่งทั้งหลายอยู่ภายในบล็อกของชุดคำสั่ง process เป็นชุดคำสั่งแบบลำดับ นั้นถ้าชุดคำสั่ง process ใดที่ไม่มี sensitivity list เป็นตัวควบคุมการทำงาน จะทำให้เกิดการทำงานที่เปรียบเสมือนว่าเป็นวงรอบ (loop) ที่ไม่รู้จบขึ้น วิธีการป้องกันการเกิดเหตุการณ์เช่นนี้ คือการเติม wait statement ลงในส่วนของชุดคำสั่ง process

2.2.4.2. Wait Statement

ชุดคำสั่ง process สามารถมี sensitivity list ได้เพียงอันเดียว หมายความว่าชุดคำสั่ง process จะถูกกระตุ้นได้จากการที่สัญญาณหนึ่งในรายชื่อที่เกิด event ขึ้นเท่านั้น หลังจากที่ถูกกระตุ้นแล้วคำสั่งทั้งหลายที่อยู่ภายในจะทำงานแบบลำดับลงมาจนกระทั่งหมด และชุดคำสั่ง process จะหยุดการทำงานชั่วคราวจนกว่าจะมี event เกิดขึ้นอีกกับสัญญาณตัวใดตัวหนึ่งในรายชื่อนั้นอีก ถ้ากรณีที่ภายในรายชื่อประกอบด้วยสัญญาณหลายตัว และเกิด event ขึ้นในเวลาเดียวกัน จะมีสัญญาณเพียงตัวเดียวจากทั้งหมดเท่านั้น ที่กระตุ้นการทำงาน ของชุดคำสั่ง process ซึ่งไม่สามารถที่จะบอกได้ว่าเป็นตัวใด ดังนั้นการใช้ชุดคำสั่ง process ร่วมกับ sensitivity list จึงมีขีดจำกัดอยู่มากในภาษา VHDL มีวิธีการหลีกเลี่ยงปัญหาเช่นนี้โดยใช้ชุดคำสั่ง wait statement

คำสั่ง wait statement นั้นให้ความคล่องตัวกว่า เพราะ wait statement มีข้อดีคือ ประการแรกสามารถที่จะกำหนดลงตรงตำแหน่งใดๆ ภายในโครงสร้างของชุดคำสั่งลำดับได้ เพื่อระงับการทำงานภายในชุดคำสั่ง process ตรงตำแหน่งที่ wait statement อยู่ ส่วนการใช้ sensitivity list เป็นตัวควบคุมการทำงานชุดคำสั่ง process จะหยุดตรงตำแหน่งสุดท้ายของชุดคำสั่ง process เท่านั้นประการที่สอง สามารถที่จะกำหนด wait statement ได้หลายๆอันในแต่ละตำแหน่ง ตามความต้องการโดยไม่จำกัด ส่วนประการสุดท้าย wait statement มีหลายรูปแบบ นั่นหมายความว่าสามารถที่จะใช้ควบคุม PROCESS ได้หลายลักษณะ แต่มีสิ่งที่ต้องจำไว้ว่า wait statement ไม่สามารถที่จะใช้ร่วมกับ sensitivity list ภายใน PROCESS เดียวกันได้ ฉะนั้นผู้เขียนรูปแบบจึงต้องตัดสินใจก่อนว่าจะใช้อะไร ทั้งนี้ขึ้นอยู่กับระบบดิจิตอลที่จะเขียนบรรยายและประสบการณ์ในการเขียน

ในภาษา VHDL สามารถใช้ wait statement ได้ 4 แบบคือ

- WAIT ON signal_list --signal sensitivity
- WAIT UNTIL condition; --condition
- WAIT FOR time; --timeout
- WAIT; --forever

แต่ละอย่างมีความหมายและวิธีใช้ที่แตกต่างกันตามตัวอย่างต่อไปนี้

- WAIT ON clock, clear, preset, d;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งจะหยุดการทำงานของชุดคำสั่งลำดับ ไว้จนกว่าจะเกิด event ขึ้นที่สัญญาณ clock หรือ clear หรือ present หรือ d

- WAIT UNTIL (clock = "1") ;

การทำงานของชุดคำสั่งลำดับจะหยุดที่ตำแหน่งนี้ และจะทำงานต่อไปเมื่อสัญญาณ clock เกิด event และ boolean expression เป็น TRUE ทุกครั้งเมื่อลำดับการทำงานถูกหยุดตรงจุดนี้ และก่อนที่จะทำต่อไปได้นั้นต้องตรวจสอบว่ามี event เกิดขึ้นบนสัญญาณ clock หรือเปล่า และสัญญาณ clock มีค่าเป็น "1" ฉะนั้น boolean expression จะต้องเป็นจริง

- WAIT FOR 10 NS;

คำสั่งนี้เป็นการรอหรือหยุดในขณะที่จำลองการทำงาน (simulation) จนกว่าเวลาการจำลองจะล่วงเลยไปแล้ว 10 ns. จะมีผลทำให้ PROCESS เริ่มทำงานต่อไปนี้

- WAIT;

เป็นคำสั่งให้ PROCESS หยุดทำงานตลอดไป โดยไม่มีการเริ่มต้นการทำงานใหม่ บางครั้งการเขียนรูปแบบบรรยายการทำงานของระบบดิจิตอล มีความจำเป็นที่ต้องใช้คำสั่งนี้ เพื่อหยุดการทำงานของอย่างถาวรนอกจากนั้นยังสามารถใช้ควบคุม (ON,UNTIL และ FOR) ร่วมกันใน wait statement ได้ส่วนใหญ่ของ VHDL analyzer จะตรวจสอบเพื่อให้เป็นที่แน่ใจว่าใน PROCESS มี sensitivity list หรือ wait statement อย่างใดอย่างหนึ่ง เพื่อป้องกันปัญหาอันอาจจะเกิดขึ้นได้

2.2.4.3. IF-THEN-ELSE statement

เป็นชุดคำสั่งพื้นฐานที่ใช้สอบถาม เพื่อการตัดสินใจกระทำอะไรบางสิ่งบางอย่าง

IF statement

Format : IF condition THEN
Sequential _statement
END IF;

2.2.4.4. CASE Statement

โครงสร้างอีกอันหนึ่งที่เป็นลำดับ และมีความคล้ายคลึงกับ IF-THEN-ELSE คือ CASE statement เพราะใช้เป็นคำสั่งเลือกหนทางปฏิบัติ ตามข้อแม้ (condition) ที่กำหนดให้

คำสั่ง CASE และ END CASE กำหนดจุดเริ่มต้นและจุดสิ้นสุด คำสั่งWHENใช้สำหรับกำหนดตัวเลือกที่จะนำมาเปรียบเทียบกับ expression ตัวเลือกใดเป็นไปตาม expression ที่กำหนด PROCESS จะเริ่มต้นทำงานที่ชุดคำสั่งลำดับที่ตามมาจนกระทั่งคำสั่งสุดท้ายของตัวเลือกนั้นๆ และจะออกจาก CASE statement โดยไม่ทำหนทางเลือกอื่นๆ ที่ยังคงเหลืออยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ชุดคำสั่งแบบแข่งขนาน (Concurrent Statement)

ภาษา VHDL เป็นภาษาที่มีการทำงานในลักษณะแข่งขนาน concurrency หรือสามารถที่จะมองชุดคำสั่งแบบแข่งขนานแต่ละอันเป็น PROCESS ที่เชื่อมต่อกันด้วย signal แต่ละ PROCESS ทำงานอิสระไม่ขึ้นต่อกัน ที่เรียกว่า asynchronous ดังนั้น concurrent statement ส่วนใหญ่จึงสามารถเขียนแทนได้ด้วย PROCESS statement

ชุดคำสั่งแบบ sequential ไม่สามารถที่จะใช้ในรูปของชุดคำสั่งแบบแข่งขนานได้ มีบางคำที่สามารถใช้ใน VHDL ได้ทั้งสองรูปแบบ เช่น signal assignment เป็นต้น ชุดคำสั่งที่ใช้ในโครงสร้างแบบ concurrent ซึ่งมีทั้งหมดได้แก่

1) Signal assignment statement
2) Component instantiation statement
3) Assert statement
4) Generate statement
5) Process statement
6) Procedure statement

Block statement

2.3 การบีบอัดควอแดรนท์ (Quadrant compression)

การบีบอัดควอแดรนท์ใช้หลักการของสมการ (Symmety) ของสัญญาณซายน์ในแต่ละควอแรนท์ นั่นคือสำหรับ $0 \le a \le 90$ แล้วจะได้ว่า

$$\begin{aligned} \sin(90-a) &= \sin(90+a), \\ \sin(270-a) &= \sin(270+a), \\ \sin a &= -\sin(a), \\ \sin a &= -\sin(180+a) \end{aligned} \tag{2.1}$$

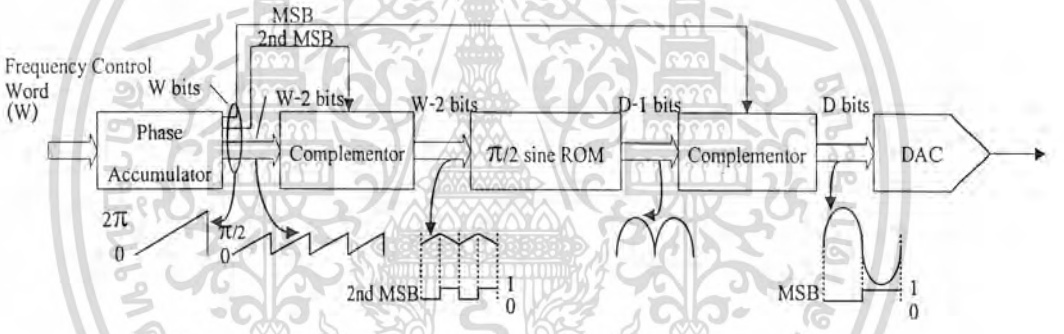
จากสมการ (2.1) จะเห็นได้ว่าเราสามารถหาค่าซายน์สำหรับทุกควอแดรนท์ (0 < a < 360) ได้จากค่า a ของควอแดรนท์แรกเพียงควอแดรนท์เดียว (0 < a < 90) โดยในการออกแบบจริงนั้นเราจะนำบิตที่มีนัยสำคัญสูงสุด (MSB) หรือเรียกว่าบิตเอ็มเอสบี 2 บิตแรกมากำหนดควอแดรนท์ของสัญญาณซายน์ที่จะสังเคราะห์ ความสำคัญของบิตเอ็มเอสบีกับควอแดรนท์และค่าแอมปลิจูดสัญญาณซายน์เราสามารถสรุปได้ดังตารางที่ 2.2 และ รูปที่ 2.14 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
61991

ตารางที่ 2.2 ความสัมพันธ์ของบิตเอ็มเอสบีกับควอแดรนท์และสัญญาณซายน์

Quadrant	MSB	MSB-1	Sine
Quadrant I	0	0	½ + ½ sin(I)
Quadrant I I	0	1	½ + ½ sin(I complemented)
Quadrant I	1	0	½ - ½ sin(I complemented)
Quadrant I ∨	0	1	½ + ½ sin(I)

* I เป็นดัชนี (Index) เริ่มจาก 0 ถึง 2^{w-2}-1



รูปที่ 2.14 การสังเคราะห์สัญญาณซายน์เต็มรูปด้วยเทคนิคบีบอัดควอแดรนท์

จากรูป 2.14 หน่วยความจำ (ROM) จะเก็บแอมปลิจูดของสัญญาณซายน์ในช่วง 0 ถึง π/2 บิตเอ็มเอสบี 2 บิต บนจะถูกใช้สำหรับกำหนดควอแดรนท์ของสัญญาณซายน์และบิตที่เหลือ W-2 บิต จะถูกใช้สำหรับชี้ตำแหน่ง ของหน่วยความจำเพื่อดึงแอมปลิจูดสัญญาณซายน์ที่ต้องการออกมา บิตเอ็มเอสบีบนสุด (MSB) ทำหน้าที่กำหนดเครื่องหมาย (Sinc) ของสัญญาณเอาต์พุตและบิตเอ็มเอสบีที่สอง (2^{nd} MSB) กำหนดว่าแอมปลิจูดของสัญญาณเอาต์พุตเป็นขาขึ้นหรือขาลงกล่าวคือในควอแดรนท์ที่ 1 และ 3 สัญญาณเอาต์พุตของสัญญาณสร้างเฟส (Phase accumulator) จะถูกนำมาใช้โดยตรงส่วนควอแดรนท์ที่ 2 และ 4 นั้นสัญญาณเอาต์พุตของวงจรสร้างเฟสจะถูกคอมพรีเมนต์ (Complemented) สัญญาณแอมปลิจูดเอาต์พุตที่ออกจากหน่วยความจำจะมีลักษระเป็นสัญญาณเต็มลูกคลื่น (Full wave) ซึ่งสามารถเปลี่ยนเป็นสัญญาณซายน์โดยการคูณแอมปลิจูดของสัญญาณที่อยู่ในช่วง π ถึง 2π ด้วย –1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าวิธีการบีบอัดควอแดรนท์นี้สามารถใช้ข้อมูลของแอมปลิจูดของสัญญาณซายน์ทั้งสี่ควอแดรนท์หรือสัญญาณเต็มลูกคลื่นได้ นั่นหมายถึงวิธีการบีบอัดควอแดรนท์นั้นสามารถรดขนาดของหน่วยความจำได้ถึง 75 เปอร์เซ็นต์

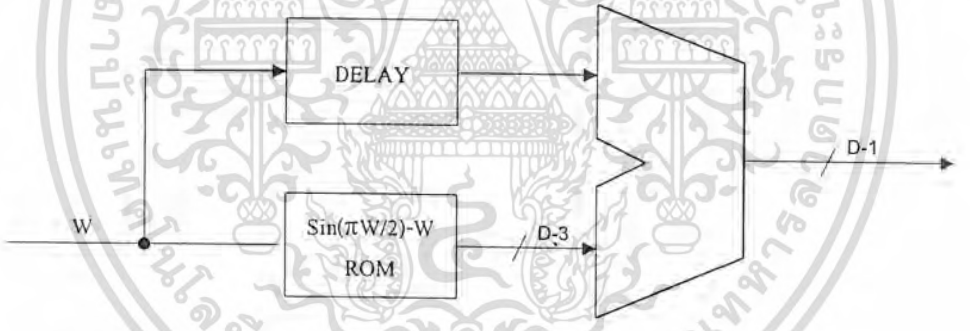
ด้วยเหตุนี้วิธีการบีบอัดควอแดรนท์จึงมักจะถูกนำมาใช้สำหรับออกแบบวงจรสังเคราะห์ความถี่แบบที่ใช้หน่วยความจำอยู่เสมอ ซึ่งในการออกแบบจริงๆแล้ว นอกจากการบีบอัดควอแดรนท์แล้วมักจะมีการบีบอัดข้อมูลเพิ่มเติมต่อไปอีก สรุปเป็นวิธีต่างๆได้ดังนี้

ขั้นตอนวิธีผลต่างซายน์-เฟส (Sine-phase difference algorithm)

ขั้นตอนวิธีผลต่างซายน์-เฟสนี้หน่วยความจำจะเก็บค่าของฟังก์ชัน $f(W)$ ซึ่งเป็นค่าผลต่างของแอมปลิจูดซายน์กับค่าเฟสที่เฟสนั้นๆฟังก์ชัน $f(W)$ สามารถเขียนเป็นสมการได้ดังนี้

$$f(W) = \sin\left(\frac{\pi W}{2}\right) - W \tag{2.2}$$

สัญญาณซายน์ที่ต้องการจะถูกสังเคราะห์โดยการนำค่าแอมปลิจูดจากหน่วยความจำมาบวกกับค่าของเฟสที่ต้องการสังเคราะห์ ดังแสดงในรูปที่ 2.15



รูปที่ 2.15 ขั้นตอนวิธีผลต่างซายน์-เฟส

วิธีการเก็บค่าของ $f(W)$ แทนการเก็บค่า $\sin(\pi W/2)$ สามารถลดจำนวนบิตของแอมปลิจูดได้ 2 บิต เพราะ

$$\max\left[\sin\left(\frac{\pi W}{2}\right) - W\right] \approx 0.21 \max\left[\sin\left(\frac{\pi W}{2}\right)\right] \tag{2.3}$$

จะเห็นว่าในขั้นตอนวิธีผลต่างซายน์-เฟสนี้จะมีฮาร์ดแวร์ของวงจร (Adder) สำหรับทำฟังก์ชั่น

$$\left[\sin\left(\frac{\pi W}{2}\right) - W\right] + W \tag{2.4}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการที่ขั้นตอนวิธีผลต่างชายน์-เฟสสามารถลดจำนวนบิตของแอมปลิจูดเอาต์พุตได้ นั่นคือขนาดของหน่วยความจำจะลดลง เป็นผลให้วงจรสังเคราะห์ความถี่จะสามารถทำงานได้เร็วขึ้น

2.3.1 ขั้นตอนวิธีซันเดอร์แลนด์ (Sunderland algorithm)

ขั้นตอนวิธีซันเดอร์แลนด์ ถูกพัฒนาต่อจากขั้นตอนวิธีฮัทชิสัน (Hutchison algorithm) ขั้นตอนวิธีซันเดอร์แลนด์มีหลักการทำงานโดยแบ่งสัญญาณเฟสที่ได้รับจากวงจรสร้างเฟสออกเป็นสามส่วน (a, b, และ c) ดังนี้

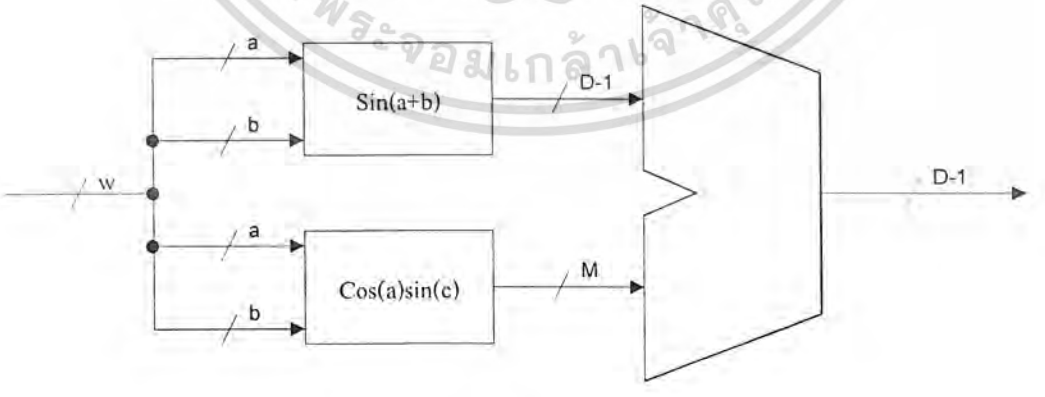
$$\sin(x) = \sin(a+b+c) \tag{2.5}$$

$$\sin(a+b+c) = \sin(a+b)\cos c + \cos a \cos b \sin c - \sin a \sin b \sin c \tag{2.6}$$

$$\sin(a+b+c) \approx \sin(a+b) + \cos a \sin b \tag{2.7}$$

และสัญญาณ $\sin(a+b+c)$ สามารถกระจายออกได้ดังสมการที่ (2.6) และประมาณค่าได้ดังสมการที่ (2.7) ตามลำดับ ซึ่งขั้นตอนวิธีซันเดอร์แลนด์นี้สามารถบีบอัดหน่วยความจำได้ประมาณ 11 เท่า

จากสมการที่ (2.7) เราสามารถนำขั้นตอนวิธีซันเดอร์แลนด์มาสร้างเป็นฮาร์ดแวร์ได้โดยแบ่งหน่วยความจำออกเป็นสองส่วนคือหน่วยความจำหยาบ (Coarse ROM) และหน่วยความจำละเอียด (Find ROM) โดยหน่วยความจำหยาบจะเก็บค่าแอมปลิจูดของ $\sin(a+b)$ และหน่วยความจำละเอียดจะเก็บค่าแอมปลิจูดของ cos a sin c ดังแสดงในรูปที่ 2.16



รูปที่ 2.16 ขั้นตอนวิธีซันเดอร์แลนด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

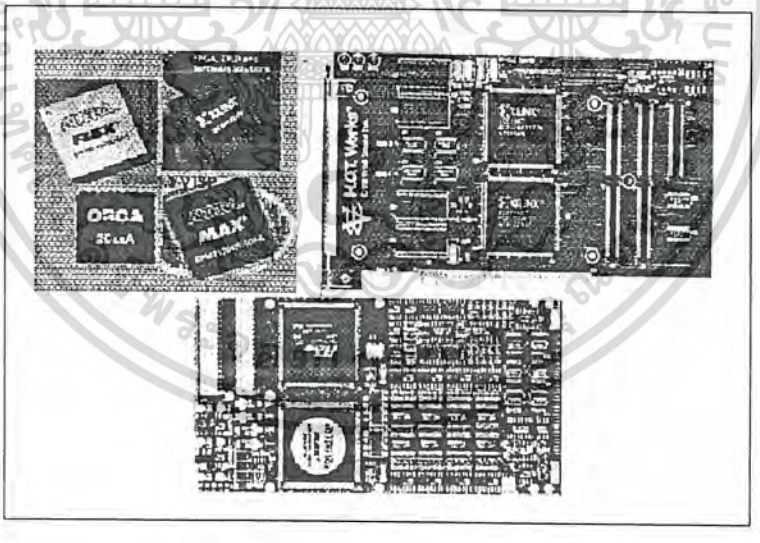
บทที่ 3

การคำนวณและการสร้าง

3.1 การออกแบบวงจรเชิงเลขด้วยอุปกรณ์ FPGA

อุปกรณ์ FPGA (Field Programmable Gate Array) เป็นอุปกรณ์ที่ใช้ในการโปรแกรมวงจรที่ได้ออกแบบลงไปเพื่อให้อุปกรณ์ FPGA มีฟังก์ชั่นการทำงานตามที่ออกแบบไว้ ในการทำ FPGA ซึ่งเป็นวิธีการออกแบบ IC (Integrated Circuit) แบบ Semicustom อีกวิธีหนึ่ง เมื่อเทียบกับการทำ ASICs (Application Specific Integrated Circuits) แล้วนั้นก็มีทั้งข้อดีและข้อเสีย คือ การทำ FPGA จะมีข้อจำกัดในด้านขนาดของวงจรเพราะภายในอุปกรณ์ FPGA จะมีจำนวนเกท (Gate) ให้ใช้จำนวนจำกัดและการทำ FPGA ก็เหมาะสำหรับการทำผลิตภัณฑ์ต้นแบบหรือเพื่อผลิตในปริมาณต่ำ ส่วนข้อดีของการทำ FPGA ก็คือระยะเวลาที่ใช้ในการทำตั้งแต่เขียนรหัส (Code) อธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลด (Download) นั้นน้อยกว่าการทำ ASIC มากและการตรวจสอบหรือแก้ไขการออกแบบก็ทำได้สะดวก

การทำ FPGA ในปัจจุบันมีประสิทธิภาพและความสะดวกมากขึ้น ทั้งนี้ก็เนื่องจากทางบริษัทผู้ผลิตอุปกรณ์ FPGA ได้เพิ่มความสามารถของอุปกรณ์ FPGA โดยเพิ่มจำนวนองค์ประกอบภายในหรือปรับปรุงโครงสร้างสถาปัตยกรรมภายในและยังได้เพิ่มประสิทธิภาพของซอฟต์แวร์ที่ใช้ทำ PPR (Partitioning Placement and Routing) สำหรับอุปกรณ์นั้นๆ ด้วย ลักษณะของตัวFPGAและการนำไปใช้งานแสดงดังใน รูปที่ 3.1



รูปที่ 3.1 ลักษณะของตัว FPGA และการนำไปใช้งาน

สำหรับตัวอุปกรณ์ FPGA นั้นก็มีโครงสร้างพื้นฐานเทคโนโลยีที่ใช้สร้างตลอดจนเทคนิควิธีการโปรแกรมที่แตกต่างกันสำหรับผู้ผลิตแต่ละราย นอกจากนั้นอุปกรณ์ FPGA ของแต่ละผู้ผลิตก็มีโครงสร้างและความสามารถที่แตกต่างกันบางส่วน ในการใช้งานนั้นอุปกรณ์ FPGA สามารถนำไปประยุกต์ใช้งานได้ เช่น การประมวลผลสัญญาณเชิงเลข (DSP : Digital Signal Processing) การออกแบบไมโครคอนโทรลเลอร์ เป็นต้นโดยมีขั้นตอนในการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบโดยใช้ภาษาอธิบายการทำงานของฮาร์ดแวร์

ในการออกแบบวงจรเชิงเลขนั้นทำได้โดยการวาดวงจรหรือใช้ภาษาอธิบายฮาร์ดแวร์ในขั้นตอนนี้เป็นขั้นตอนที่ไม่แตกต่างกันระหว่างการออกแบบด้วย FPGA และ ASIC ในกรณีที่ใช้ภาษาอธิบายฮาร์ดแวร์ แต่ในกรณีที่ออกแบบโดยวิธีการวาดวงจรจะแตกต่างกันโดยที่การทำวิธีนี้จะต้องคำนึงถึงเทคโนโลยีที่จะใช้ซึ่งแต่ละเทคโนโลยีก็มีความแตกต่างกันไป จะเห็นได้ว่าการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์ ทำได้สะดวกกว่าเพราะการทำด้วยวิธีนี้ไม่ต้องคำนึงถึงเทคโนโลยีที่จะใช้ (Technology independence) และที่สำคัญการออกแบบด้วยวิธีนี้สามารถที่จะแก้ไขโมเดล (Model) หรือเปลี่ยนแปลงเทคโนโลยีได้สะดวกกว่าเพราะไม่ต้องวาดวงจรใหม่ นั่นคือการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์จะทำให้โมเดลที่ได้ไม่ขึ้นกับเทคโนโลยี

ในการเขียนโค้ดสิ่งที่ต้องคำนึงถึงคือเขียนอย่างไรจึงจะสามารถสังเคราะห์เป็นวงจรได้และให้คุณสมบัติของวงจรตามที่กำหนด เพราะลักษณะการเขียนโค้ดจะมีผลโดยตรงกับวงจรที่ได้ เนื่องจากในการสังเคราะห์วงจรนั้นซอร์ฟแวร์สังเคราะห์วงจร (Synthesis Tools) จะทำการสังเคราะห์ตามโค้ดที่เขียน ถ้าอธิบายการทำงานของวงจรเดียวกันแต่เขียนโค้ดในลักษณะที่ต่างกันเมื่อสังเคราะห์แล้วจะได้วงจรที่ต่างกัน และจากวงจรที่ต่างกัน เมื่อนำไปทำต้นแบบด้วย FPGA หรือการทำ ASIC แล้วจะได้ไอซีที่มีคุณสมบัติต่างกันทั้งในด้านของขนาดหรือความเร็ว (Area and Time) ส่วนการเขียนโค้ดลักษณะใดเพื่อให้ได้ผลลัพธ์ที่ดีที่สุดนั้นก็ขึ้นอยู่กับประสบการณ์ในการออกแบบ

3.2.1 การจำลองการทำงานของวงจร (Simulation)

ขั้นตอนนี้เป็นขั้นตอนที่สำคัญเพราะเป็นขั้นตอนที่ใช้ตรวจสอบฟังก์ชั่นการทำงานของวงจรว่าถูกต้องหรือไม่ มีข้อผิดพลาดตรงไหน เพื่อที่จะได้ทำการแก้ไขให้ถูกต้อง ในขั้นตอนนี้จะใช้ซอร์ฟแวร์สำหรับทำการจำลองการทำงานของวงจร เช่น V-System และ Model Sim ของบริษัท Model Technology

3.2.2 การสังเคราะห์วงจร

ในขั้นตอนนี้จะใช้ซอร์ฟแวร์สังเคราะห์วงจร (Synthesis tools) ทำการสังเคราะห์โค้ดเพื่อให้ได้เป็นวงจรขึ้นมา แต่ต้องตรวจสอบด้วยว่าซอร์ฟแวร์นั้นๆสนับสนุนเทคโนโลยี FPGA (FPGA Library) ที่ต้องการใช้หรือไม่โดย FPGA ที่นิยมใช้งานเช่นของบริษัท Xilinx ตระกูล XC4000 และบริษัท Altera ตระกูล FLEX 10 K ซอร์ฟแวร์สังเคราะห์วงจรที่นิยมใช้เช่นโปรแกรม Leonardo Spectrum ของบริษัท Exemplar Logic ซึ่งในขั้นตอนนี้ซอร์ฟแวร์สังเคราะห์วงจรจะแปลงโค้ดและทำการออปติไมซ์ (Optimization)เพื่อให้ได้วงจรตามเทคโนโลยีที่เลือกใช้ นอกจากนี้ยังสามารถกำหนดข้อบังคับสำหรับวงจรได้เช่น ข้อบังคับในเรื่องของเวลา (Time Constraints) หรือข้อบังคับในเรื่องของพื้นที่ ซึ่งข้อบังคับเหล่านี้จะถูกนำไปใช้ในขั้นตอนออปติไมซ์เพื่อให้วงจรที่ได้เป็นไปตามที่กำหนด ส่วนสำคัญในการออปติไมซ์คือการเทียบ (Mapping) วงจรให้เข้ากับเทคโนโลยีที่ใช้เพื่อให้ได้วงจรที่เหมาะสมกับโครงสร้างสถาปัตยกรรมภายในอุปกรณ์ FPGA ในกรณีของ Xilinx ตระกูล XC4000 และ Altera ตระกูล FLEX 10 K จะเทียบโดยใช้วิธี LUT (Look Up Table)เมื่อทำการสังเคราะห์วงจรเสร็จแล้วซอร์ฟแวร์สังเคราะห์วงจรก็

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีการรายงานผลว่าวงจรที่ออกแบบไปนั้นเป็นอย่างไร เช่น มีความหน่วง (Delay) เท่าไรใช้ทรัพยากรต่างๆใน FPGA อะไรบ้าง เป็นต้น

3.2.3 การแบ่งวงจร (Partitioning)

ขั้นตอนนี้เป็นการแบ่งวงจรที่ได้จากการสังเคราะห์ให้เป็นส่วนย่อยๆ สำหรับลงใน CLBs, IOBs หรือองค์ประกอบอื่นๆ ภายในอุปกรณ์ FPGA สำหรับเกณฑ์ที่ใช้ในการแบ่งคือให้แต่ละส่วนที่จะแยกออกจากกันมีจำนวนสัญญาณที่เชื่อมต่อระหว่างกันน้อยที่สุดเท่าที่จะทำได้เพื่อช่วยลดความหนาแน่นในตอนทำการเชื่อมต่อสัญญาณ (Routing) ในขั้นตอนนี้จะใช้ซอร์ฟแวร์ทำโดยซอร์ฟแวร์จะเทียบส่วนประกอบของวงจรเช่น เกท (Gate), ฟลิปฟลอป (Flipflop) ลงในทรัพยากรต่างๆ ที่มีอยู่ภายในอุปกรณ์ FPGA (CLBs, IOBs, BUFT) และ (Edge Decoder) หลังจากทำขั้นตอนนี้เสร็จแล้วสามารถที่จะทราบว่าวงจรใช้จำนวนทรัพยากรภายในอุปกรณ์ FPGA ไปเท่าไร ส่วนซอร์ฟแวร์ที่ใช้ในขั้นตอนนี้ขึ้นอยู่กับตัว FPGA ที่ใช้งานเช่น FPGA ของบริษัท Xilinx จะใช้ Xilinx Foundation Series 2.1i ซึ่งซอร์ฟแวร์ตัวนี้จะรวมเอาซอร์ฟแวร์ย่อยอื่นๆอีก เพื่อให้การทำ PPR (Partitioning, Placement and Routing) เป็นไปอย่างต่อเนื่อง ส่วน FPGA ของบริษัท Altera จะใช้ Altera MAX+II

3.2.4 การวางอุปกรณ์ (Placement)

ขั้นตอนนี้เป็นการเลือกทำเลที่ตั้งของแต่ละส่วนของวงจรที่ผ่านการแบ่งวงจร (Partitioning) มาแล้วว่าควรจะอยู่ในตำแหน่งใดในอุปกรณ์ FPGA เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด เช่นวงจรส่วนไหนควรอยู่ใกล้กันเพื่อจะได้ค้นหาเส้นทาง (Route) ได้ง่ายหรือช่วยลดความหน่วง

จะเห็นได้ว่าตำแหน่งภายในอุปกรณ์ FPGA นั้นมีความสำคัญเพราะถ้าจัดวางวงจรลงในตำแหน่งที่ไม่เหมาะสมแล้วจะทำให้ความหน่วงเพิ่มขึ้นหรือตัว Router ทำการค้นหาเส้นทางสัญญาณได้ไม่หมด

3.2.5 การเชื่อมต่อสัญญาณ (Routing)

ในขั้นตอนนี้เป็นการเชื่อมต่อสัญญาณระหว่างองค์ประกอบต่างๆ ภายในอุปกรณ์ FPGA เช่น ระหว่าง CLBs หรือระหว่าง CLBs กับ IOBs ขั้นตอนนี้จะทำต่อเนื่องจากการวางอุปกรณ์ ในกรณีที่ทำการวางอุปกรณ์ไว้ไม่ดีซอร์ฟแวร์ก็จะทำการเชื่อมต่อสัญญาณได้ไม่หมดหรือเกิดความหน่วงเกินค่าที่กำหนดในข้อบังคับโดยสามารถทำขั้นตอนนี้ได้โดยใช้ซอร์ฟแวร์เช่นกันหรือทำการเชื่อมต่อสัญญาณด้วยตัวเอง (Manual Layout) ก็ได้แต่ทางที่ดีควรใช้ซอร์ฟแวร์ทำดีกว่าโดยให้ทำการค้นหาเส้นทางหลายๆครั้งเพื่อหาครั้งที่ดีที่สุด นอกจากนั้นการกำหนดข้อบังคับทางเวลา (Time Constraints) จะช่วยให้ผลที่ได้จากการทำการเชื่อมต่อสัญญาณดีขึ้นได้

3.2.6 การโปรแกรมอุปกรณ์ FPGA (Configuration)

หลังจากที่วงจรผ่านขั้นตอนต่างๆจนกระทั่งผ่านการทำ PPR (Partitioning, Placement and Routing) แล้วนั้น ถึงตอนนี้ก็สามารถที่จะดาวน์โหลดลงในอุปกรณ์ FPGA ได้แล้วในการดาวน์โหลดนี้ก่อนอื่นต้องแปลงแบบวงจรรวมที่ได้ให้เป็นข้อมูลวงจร (Configuration data) ซึ่งอยู่ในรูปของบิทสตรีม (Bit-Stream) ก่อนแล้วจึงดาวน์โหลดลงไปเพื่อให้อุปกรณ์ FPGA มีฟังก์ชั่นการทำงานตามวงจรที่ออกแบบไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่อธิบายมาทั้งหมดจะเห็นได้ว่าการออกแบบเพื่อทำ FPGA นั้น ทำได้สะดวกกว่าการทำ ASIC มากเพราะใช้เวลาน้อยกว่ามาก ส่วนสำคัญที่ใช้ในการทำ FPGA คือซอร์ฟแวร์ที่ใช้ตั้งแต่การเขียนโค้ดอธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลดลงในอุปกรณ์ FPGA ซึ่งซอร์ฟแวร์ที่ใช้ต้องเป็นซอร์ฟแวร์ที่ใช้ทำงานต่อเนื่องกัน

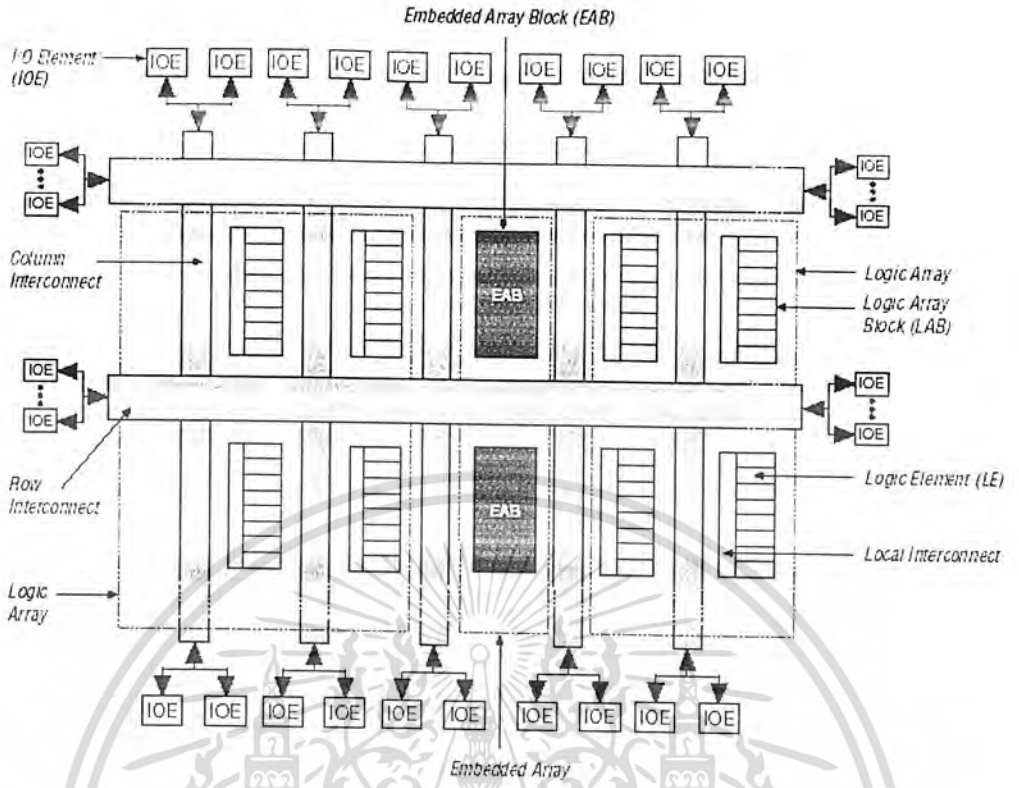
3.3 FPGA ตระกูล ACEX1K

FPGA ตระกูล ACEX1K ของบริษัท ALTERA เป็น FPGA ที่มีโครงสร้างภายในเป็นแบบ SRAM Based FPGA ใช้เทคโนโลยีในการโปรแกรมเหมือนกับหน่วยความจำแบบ SRAM (Static Ram) ทำให้การโปรแกรมสามารถทำซ้ำได้โดยไม่จำกัดจำนวนครั้ง และใช้เวลาโปรแกรมน้อยมาก (ระดับ nsec) การโปรแกรมทำได้ง่ายเช่นเดียวกับการเขียน SRAM ทั่วไป แต่มีข้อเสียคือ มันต้องการไฟเลี้ยงในการเก็บวงจรที่ออกแบบไว้

Feature	POWER ACEX1K SERIES		
	POWER ACE1K-10	POWER ACE1K-30	POWER ACE1K-50
ACEX1K Part Nuber	EP1K10TC144	EP1K30TC144	EP1K50TC144
Typical gates	10,000	30,000	50,000
Speed Grade	-3	-3	-3
Maximum system gate	56,000	119,000	199,000
Logic elements(Les)	576	1,728	2,880
Embedded Array Block (EABs)	3	6	10
Total RAM bits (Internal memory)	12,288	24,576	40,960
Multi Volt I/O Support	2.5V,3.3V,5.0V	2.5V,3.3V,5.0V	2.5V,3.3V,5.0V
Maximum Frequency	180MHz	180MHz	180MHz
User I/O pins	102	102	102
Package	TQFP144	TQFP144	TQFP144

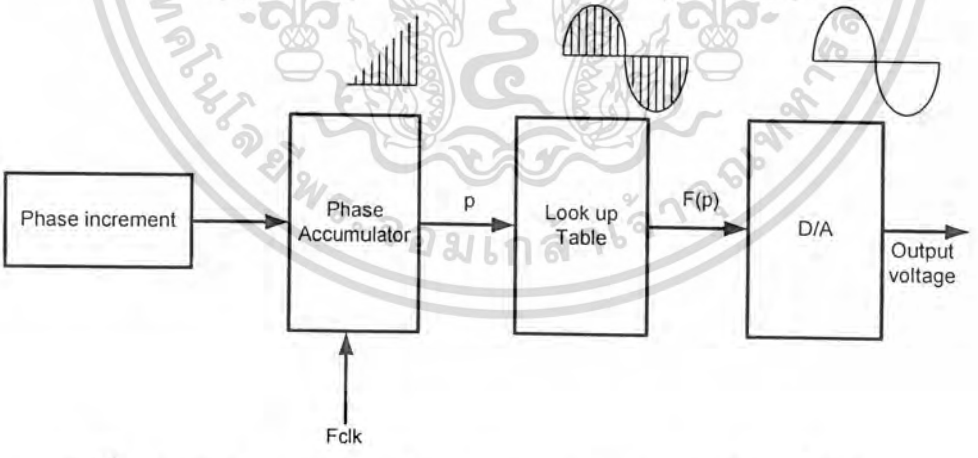
ตารางที่ 3.1 คุณลักษณะต่างๆของชิป FPGA ที่ใช้กับบอร์ดในกลุ่ม POWER ACEX1K SERIES

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 โครงสร้างภายในของ FPGA ตระกูล ACEX1k

3.4 Direct Digital Frequency Synthesizer (DDS)

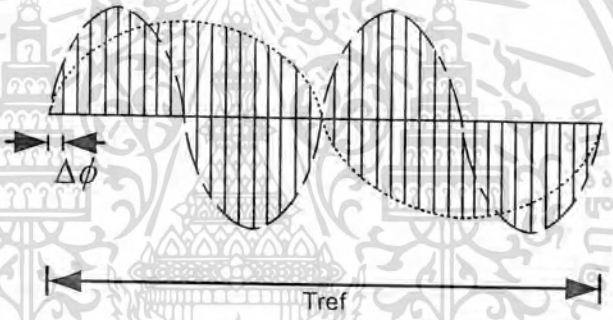


รูปที่ 3.3 แสดง Block Diagram โครงสร้างของวงจรสังเคราะห์ความถี่ที่ใช้หน่วยความจำเป็นค่าขนาดของสัญญาณเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรสังเคราะห์ความถี่ดิจิตอลแบบโดยตรง มีโครงสร้างหลัก ๆ ตามรูป จะอาศัยหน่วยความจำเพื่อเก็บค่าขนาดของสัญญาณที่เฟสต่าง ๆ เอาไว้ ซึ่งรูปสัญญาณที่เก็บเอาไว้ อาจจะเป็นรูปสัญญาณซายน์ (Sine) สัญญาณรูปฟันเลื่อย (sawtooh) หรือรูปสัญญาณอื่น ๆ ที่ผู้ใช้งานได้ทำการออกแบบไว้ เมื่อต้องการรูปสัญญาณออกมาที่เอาท์พุต ก็จะทำการเรียกข้อมูลที่เฟสต่าง ๆ ที่เก็บเอาไว้ออกมา เพื่อป้อนให้กับวงจรแปลงสัญญาณดิจิตอลเป็นอนาลอก ดังรูปที่ 3.3

หากต้องการจะเปลี่ยนความถี่ของสัญญาณก็ทำได้โดย การเปลี่ยนอัตราการเรียกขอมูลออกจากหน่วยความจำ (เป็นความเร็วของสัญญาณนาฬิกา) หรือเปลี่ยนลำดับการเรียกข้อมูลที่เฟสต่าง ๆ เพื่อให้จำนวนของ ข้อมูลในหนึ่งคาบของสัญญาณเปลี่ยนไป ดังเช่นในรูปที่ 3.4 จะเห็นว่ามีรูปของสัญญาณสองความถี่ รูปสัญญาณที่มีความถี่สูงกว่า จะมีจำนวนข้อมูลที่ถูกเรียกออกมาน้อยกว่ารูปสัญญาณที่มีความถี่ต่ำกว่า



รูปที่ 3.4 วงจรสร้างสัญญาณซายน์แบบเชิงเลข

มีข้อสังเกตคือ ระยะ ระหว่างข้อมูลที่ถูกเรียกออกมาของสัญญาณทั้งสองความถี่มีค่าเท่ากัน ดังนั้นความละเอียดในการเปลี่ยนความถี่จะขึ้นอยู่กับขนาดของหน่วยความจำที่ใช้เก็บข้อมูล

หลักการของวงจรสังเคราะห์ความถี่ดิจิตอลแบบโดยตรง (Direct Digital Frequency Synthesizer) นั้นจะมีสัญญาณนาฬิกา (Fclk) นำมาป้อนให้กับส่วนของ Phase Accumulator ซึ่งวงจรนี้จะทำการบวกวนซ้ำด้วยค่าที่รับเข้ามาจากส่วน เพิ่มเฟส ผลของการบวกจะได้ค่าออกมาเป็น เฟสสะสม P ค่าสะสมเฟส P นี้จะมีลักษณะเพิ่มขึ้นด้วยจำนวนเท่ากันทุกครั้งที่ทำการบวก ตามจังหวะสัญญาณนาฬิกา อ้างอิง (Fclk) ค่าเฟสสะสม P จะนำไปชี้ในตารางเปิดดู (Look up table) ซึ่งได้จัดเก็บค่าขนาดของรูปสัญญาณไว้แล้ว ซึ่งจะได้สัญญาณ F(P) ออกมา ซึ่งเป็นค่าแบบดิจิตอล มาทำการเปลี่ยนเป็นค่า อนาลอก ด้วยวงจร Digital – to – Analog Converter (DA) ก็จะได้ Output voltage เป็นสัญญาณแบบที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่ที่สามารถกำเนิดได้จากโครงสร้างนี้ คือ

$$\text{Fout (min)} = \text{Fclk} / 2^n$$

$$\text{Fout} = (\text{Frequency Input} \times \text{Fclk}) / 2^n \quad (3.1)$$

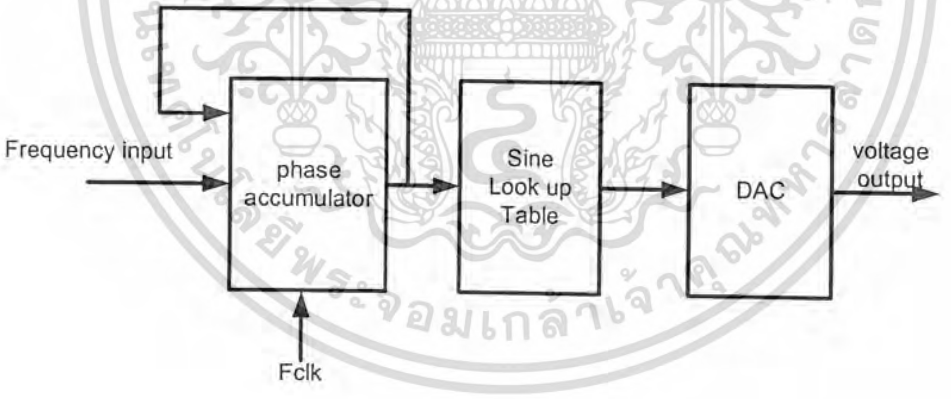
เมื่อ

Fout = Frequency Output

Fclk = Frequency Clock

N = จำนวน bit ของ Phase Accumulator

ถ้าค่า N มีค่ามาก Step size ของความถี่จะมีความละเอียดมากขึ้นด้วย

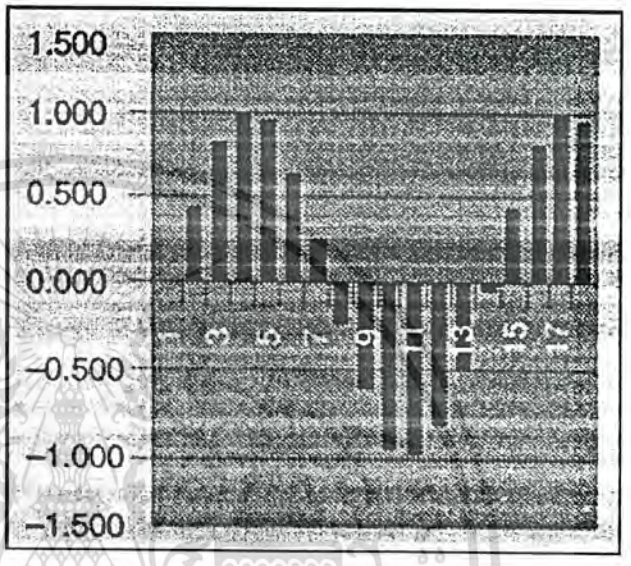


รูปที่ 3.5 แสดงการกำเนิดสัญญาณซายน์แบบดิจิตอล ที่มีค่า Phase Increment เท่ากับ 55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Phase increment register : 55

Time	Phase	Sinus
0	0	0.000
1	55	0.462
2	110	0.819
3	165	0.991
4	220	0.940
5	275	0.676
6	330	0.259
7	385	-0.216
8	440	-0.643
9	495	-0.924
10	550	-0.996
11	605	-0.843
12	660	-0.500
13	715	-0.044
14	770	-0.423
15	825	0.793
16	880	0.985
17	935	0.954

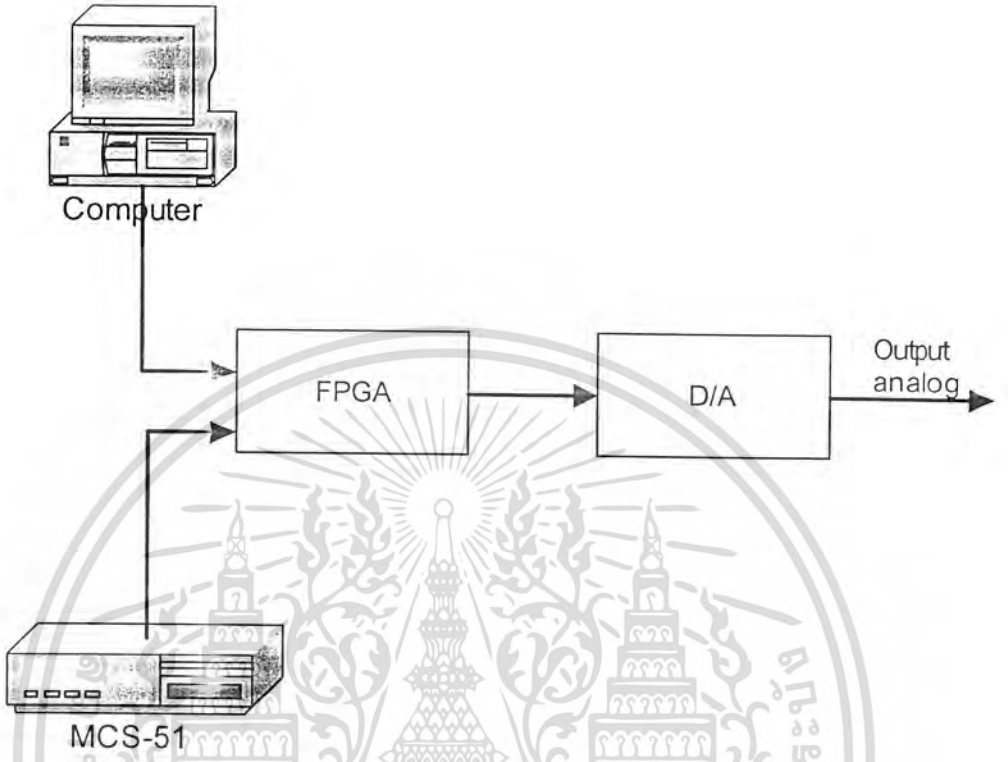


รูปที่ 3.6 Phase Increment เท่ากับ 55

3.4.1 การออกแบบและการสร้าง

การออกแบบตัวกำเนิดสัญญาณไซน์แบบดิจิตอลด้วย FPGA เป็นการออกแบบโดยการนำเอา Personal Computer (PC) และ MCS-51 กับ FPGA ต่อทำงานร่วมกัน ดัง รูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 ขั้นตอนการทำงานของ ตัวกำเนิดสัญญาณไซน์

3.4.2 การสร้างสัญญาณรูปคลื่นไซน์

ตารางที่ 3.2 การเก็บค่าขนาด (Amplitude) ในหน่วยความจำ

Phase	Amplitude
0	10000000
1	10000010
2	10000100
3	10000110
4	10001000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Phase	Amplitude
5	10001011
6	10001101
7	10001111
8	10010001
9	10010011
10	10010110
11	10011000
12	10011010
13	10011100
14	10011110
15	10100000
16	10100011
17	10100101
18	10100111
19	10101001
20	10101011
21	10101101
22	10101111
23	10110001
24	10110011
25	10110101
26	10110111
27	10111001
28	10111011
29	10111101
30	10111111
32	11000001
33	11000011
34	11000101
35	11000111
36	11001001
37	11001010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Phase	Amplitude
38	11001100
39	11001110
40	11010000
41	11010001
42	11010011
43	11010101
44	11010110
45	11011000
46	11011010
47	11011011
48	11011101
49	11011110
50	11100000
51	11100001
52	11100011
53	11100100
54	11100101
55	11100111
56	11101000
57	11101001
58	11101010
59	11101100
60	11101101
61	11101110
62	11101111
63	11110000
64	11110001
65	11110010
66	11110011
67	11110100
68	11110101
69	11110110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Phase	Amplitude
70	11110111
71	11110111
72	11111000
73	11111001
74	11111001
75	11111010
76	11111011
77	11111011
78	11111100
79	11111100
80	11111101
81	11111101
82	11111101
83	11111110
84	11111110
85	11111110
86	11111111
87	11111111
88	11111111
89	11111111
90	11111111
91	11111111
92	11111111
93	11111111
94	11111111
95	11111111
96	11111111
97	11111110
98	11111110
99	11111110
100	11111101
101	11111101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Phase	Amplitude
102	11111101
103	11111100
104	11111100
105	11111011
106	11111011
107	11111010
108	11111001
109	11111001
110	11111000
111	11110111
112	11110111
113	11110110
114	11110101
115	11110100
116	11110011
117	11110010
118	11110001
119	11110000
120	11101111
121	11101110
122	11101101
123	11101100
124	11101010
125	11101001
126	11101000
127	11100111
128	11100101
129	11100100
130	11100011
131	11100001
132	11100000
133	11011110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Phase	Amplitude
134	11011101
135	11011011
136	11011010
137	11011000
138	11010110
139	11010101
140	11010011
141	11010001
142	11010000
143	11001110
144	11001100
145	11001010
146	11001001
147	11000111
148	11000101
149	11000011
150	11000001
151	10111111
152	10111101
153	10111011
154	10111001
155	10110111
156	10110101
157	10110011
158	10110001
159	10101111
160	10101101
161	10101011
162	10101001
163	10100111
164	10100101
165	10100011

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Phase	Amplitude
166	10100000
167	10011110
168	10011100
169	10011010
170	10011000
171	10010110
172	10010011
173	10010001
174	10001111
175	10001101
176	10001011
177	10001000
178	10000110
179	10000100
180	10000010
181	10000000
182	01111101
183	01111011
184	01111001
185	01110111
186	01110100
187	01110010
188	01110000
189	01101110
190	01101100
191	01101001
192	01100111
193	01100101
194	01100011
195	01100001
196	01011111
197	01011100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Phase	Amplitude
198	01011010
199	01011000
200	01010110
201	01010100
202	01010010
203	01010000
204	01001110
205	01001100
206	01001010
207	01001000
208	01000110
209	01000100
210	01000010
211	01000000
212	00111110
213	00111100
214	00111010
215	00111000
216	00110110
217	00110101
218	00110011
219	00110001
220	00101111
221	00101110
222	00101100
223	00101010
224	00101001
225	00100111
226	00100101
227	00100100
228	00100010
229	00100001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Phase	Amplitude
230	00011111
231	00011110
232	00011100
233	00011011
234	00011010
235	00011000
236	00010111
237	00010110
238	00010101
239	00010011
240	00010010
241	00010001
242	00010000
243	00001111
244	00001110
245	00001101
246	00001100
247	00001011
248	00001010
249	00001001
250	00001000
251	00001000
252	00000111
253	00000110
254	00000110
255	00000101
256	00000100
257	00000100
258	00000011
259	00000011
260	00000010
261	00000010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Phase	Amplitude
262	00000010
263	00000001
264	00000001
265	00000001
266	00000000
267	00000000
268	00000000
269	00000000
270	00000000
271	00000000
272	00000000
273	00000000
274	00000000
275	00000000
276	00000000
277	00000001
278	00000001
279	00000001
280	00000010
281	00000010
282	00000010
283	00000011
284	00000011
285	00000100
286	00000100
287	00000101
288	00000110
289	00000110
290	00000111
291	00001000
292	00001000
293	00001001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

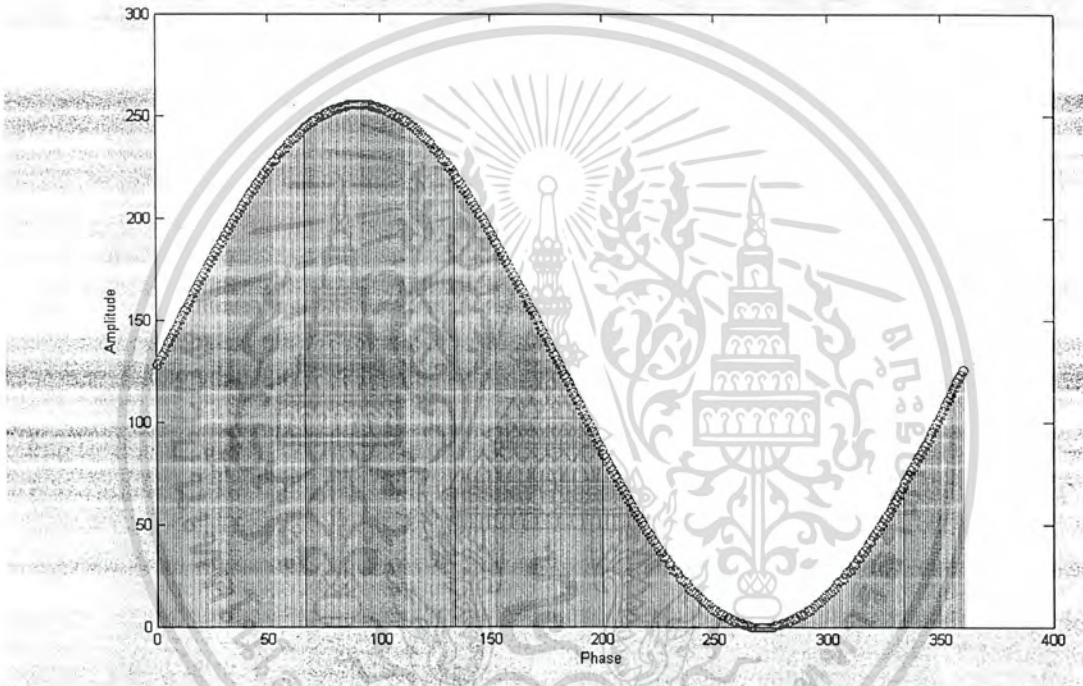
Phase	Amplitude
294	00001010
295	00001011
296	00001100
297	00001101
298	00001110
299	00001111
300	00010000
301	00010001
302	00010010
303	00010011
304	00010101
305	00010110
306	00010111
307	00011000
308	00011010
309	00011011
310	00011100
311	00011110
312	00011111
313	00100001
314	00100010
315	00100100
316	00100101
317	00100111
318	00101001
319	00101010
320	00101100
321	00101110
322	00101111
323	00110001
324	00110011
325	00110101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Phase	Amplitude
326	00110110
327	00111000
328	00111010
329	00111100
330	00111110
331	01000000
332	01000010
333	01000100
334	01000110
335	01001000
336	01001010
337	01001100
338	01001110
339	01010000
340	01010010
341	01010100
342	01010110
343	01011000
344	01011010
345	01011100
346	01011111
347	01100001
348	01100011
349	01100101
350	01100111
351	01101001
352	01101100
353	01101110
354	01110000
355	01110010
356	01110100
357	01110111

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

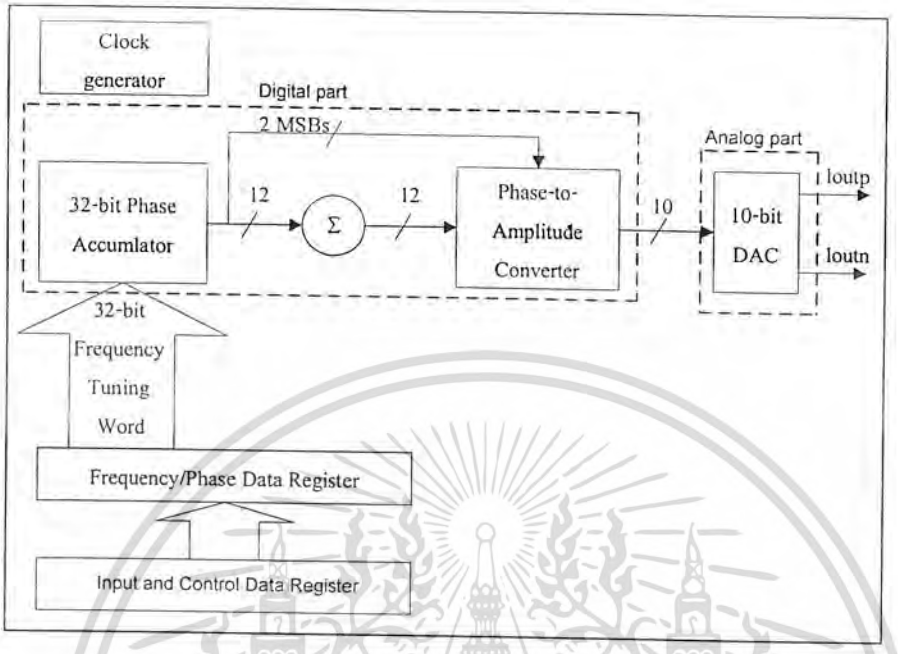
Phase	Amplitude
358	01111001
359	01111011
360	01111101



รูปที่ 3.8 ผลจากค่าในตารางมาวาดกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 รายระเอียดการออกแบบ



รูปที่ 3.9 สถาปัตยกรรมของชิปวงจรรวมเพื่อสังเคราะห์ความถี่แบบดิจิตอลที่ใช้หน่วยความจำ

วงจรรวมเพื่อสังเคราะห์ความถี่แบบดิจิตอลที่ใช้หน่วยความจำมีสถาปัตยกรรมตามรูปที่ 3.9 โดยจะแบ่งออกเป็นส่วนหลักๆคือส่วนดิจิตอล (Digital part) และส่วนอนาลอก (Analog part) โดยส่วนดิจิตอลจะประกอบด้วยวงจรสร้างเฟสขนาด 32 บิต และวงจรเปลี่ยนเฟสเป็นแอมปลิจูด และส่วนประกอบหลักของส่วนอนาลอกคือวงจรแปลงดิจิตอลเป็นวงจรขนาด 12 บิต เราสามารถสรุปรายระเอียดการออกแบบวงจรแยกตามส่วนประกอบหลักๆได้ดังนี้

ส่วนดิจิตอล (Digital part)

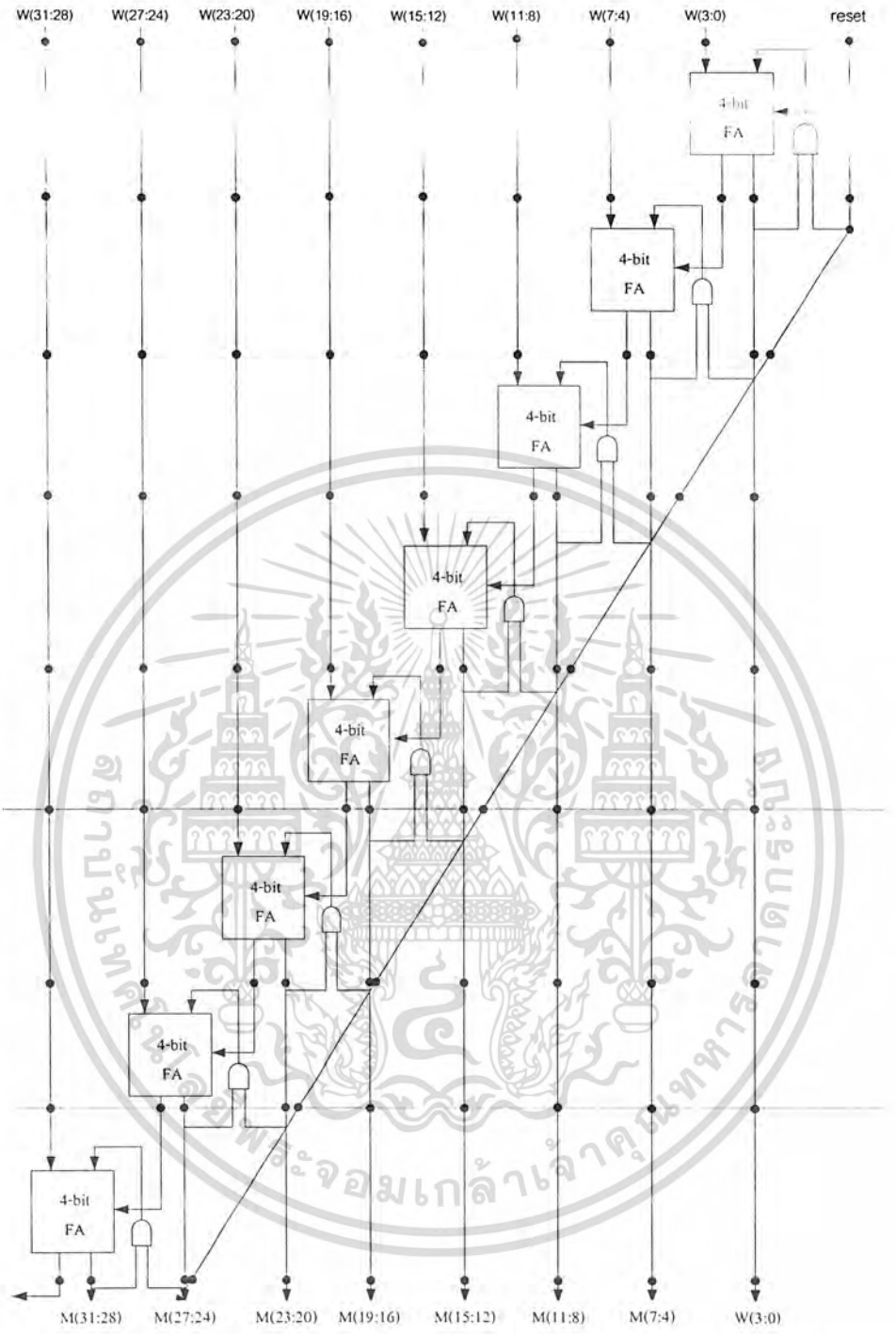
วงจรในส่วนดิจิตอลได้ถูกออกแบบด้วยวิธีการออกแบบจากบนลงล่าง (Top-down design) ซึ่งมีขั้นตอนคือเริ่มจากการกำหนดข้อกำหนด (Specification) ของวงจรและเมื่อได้ข้อกำหนดของวงจรแล้วก็จะทำการเขียนโค้ดคำสั่งบรรยายพฤติกรรมการทำงานของวงจรด้วยภาษาวีเอชดีแอล เสร็จแล้วนำไปสังเคราะห์

วงจรสร้างเฟส 32 บิต (32-bit phase accumulator)

วงจรสร้างเฟส 32 บิตมีโครงสร้างดังรูปที่ 3.10 ประกอบด้วยวงจรบวกขนาด 4 บิต (4-bit full adder) จำนวน 8 ชุดมาต่อกันในลักษณะสายท่อ (Pipeline) เพื่อเพิ่มความเร็วในการทำงานของวงจร นอกจากนี้ยังมี

การนำสัญญาณรีเซตแบบสายท่อมาใช้เพื่อเคลียร์ค่าของรีจีสเตอร์เฟส โดยที่จุดดำๆในรูปที่ 3.10 เป็นรีจีสเตอร์สำหรับปรับเวลา (Pre-skewing and De-skewing registers)

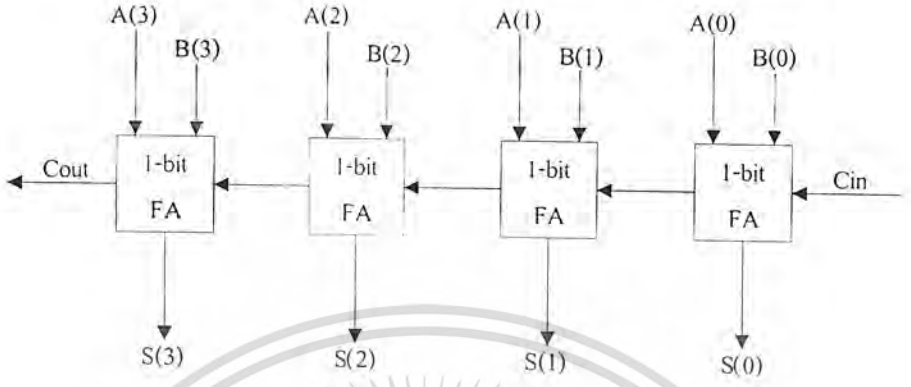
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 โครงสร้างของวงจรสร้างเฟสขนาด 32 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรบวกขนาด 4 บิตที่นำมาใช้ออกแบบเป็นวงจรสร้างเฟส 32 บิต มีโครงสร้างเป็นวงจรบวกแบบริปเปิลแคร์รี่ (Ripple-carry adder) ดังแสดงในรูปที่ 3.11



รูปที่ 3.11 โครงสร้างของวงจรบวก 4 บิต แบบริปเปิลแคร์รี่

วงจรเปลี่ยนเฟสเป็นแอมปริจูด (Phase-to-amplitude converter)

วงจรเปลี่ยนเฟสเป็นแอมปริจูดสามารถสังเคราะห์รูปสัญญาณได้ทั้งหมด 4 รูปสัญญาณ ได้แก่ รูปซายน์ รูปแรมป์ รูปฟันเลื่อย และสัญญาณแบบสุ่ม วิธีบีบอัดควอแดรนท์ ได้ถูกนำมาใช้ในการสังเคราะห์สัญญาณทุกสัญญาณ นอกจากวิธีการบีบอัดควอแดรนท์แล้วเราสามารถอธิบายขั้นตอนการสังเคราะห์สัญญาณแต่ละรูปสัญญาณได้ดังนี้

รูปซายน์ (Sine)

ใช้หลักการของการเก็บรูปสัญญาณไว้ในหน่วยความจำแบบอ่านอย่างเดียว (ROM) โดยบีบอัดข้อมูลเพื่อลดขนาดของหน่อยความจำด้วยขั้นตอนวิธีโมดิฟายซันเดอร์แลนด์ (Modified Sunderland algorithm)

ในขั้นตอนวิธีซันเดอร์แลนด์ค่าของเฟสขนาด 12 บิต (ϕ) ที่ถูกสร้างจากวงจรสร้างเฟสจะแบ่งออกเป็น 3 ส่วน เท่าๆกัน ส่วนละ 4 บิต (a,b,c) ดังสมการที่ 3.2

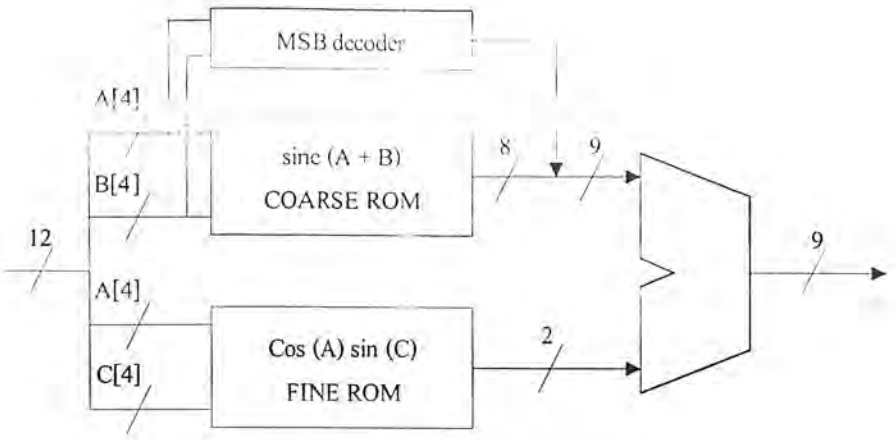
$$\phi = a + b + c \tag{3.2}$$

และจากความสัมพันธ์ของฟังชั่น $\sin(\phi) = \sin(a+b+c)$ เราสามารถประมาณค่าของฟังก์ชั่น $\sin(\phi)$ ได้ดังสมการที่ (3.3) และ (3.4) ตามลำดับ

$$\sin(a+b+c) = \sin(a+b)\cos c + \cos a \cos b \sin c - \sin a \sin b \sin c \tag{3.3}$$

$$\sin(a+b+c) \approx \sin(a+b) + \cos a \sin b \tag{3.4}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



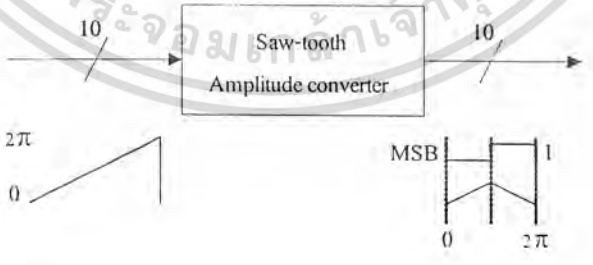
รูปที่ 3.12 ขั้นตอนวิธีโมดิฟายชันเดอร์แลนด์

สมการที่ 3.4 จะถูกใช้ในการออกแบบเป็นวงจรจริง โดย $\sin(a+b)$ จะถูกเก็บในหน่วยความจำหยาบ (Coarse ROM) และ cos a sin c จะถูกเก็บในหน่วยความจำละเอียด (Fine ROM) สัญญาณเอาต์พุตของทั้งหน่วยความจำหยาบและหน่วยความจำละเอียดจะถูกนำมารวมกันเพื่อสร้างเป็นสัญญาณซายน์ ดังแสดงในรูปที่ 3.12 โดยที่สัญญาณเอาต์พุตของวงจรเปลี่ยนเฟสเป็นแอมปลิจูดด้วยขั้นตอนวิธีชันเดอร์แลนด์จะมีขนาด 11 บิต ซึ่งจะถูกนำไปดีโคดควอแดรนท์และสร้างเป็นสัญญาณแอมปลิจูดเอาต์พุตของสัญญาณซายน์ขนาด 12 บิต ต่อไป

รูปแรมป์ (Ramp)

สัญญาณรูปแรมป์นั้นสามารถสังเคราะห์ได้โดยตรงจากสัญญาณเอาต์พุทของวงจรสร้างเฟส โดยจะดึงเฉพาะส่วนบิตเอ็มเอสบี 12 บิต แรกมาเป็นสัญญาณเอาต์พุตของวงจร เปลี่ยนเฟสเป็นแอมปลิจูดเพื่อป้อนให้กับวงจรแปลงเป็นดิจิตอลเป็นอนาลอกต่อไป

รูปฟันเลื่อย (Saw-tooth)



รูปที่ 3.13 การสังเคราะห์สัญญาณรูปฟันเลื่อย

สัญญาณรูปฟันเลื่อยสามารถสังเคราะห์ได้จากบิตเอ็มเอสบี 10 บิตแรกที่ได้จากเอาต์พุตของวงจรสร้างเฟส โดยถ้าบิตเอ็มเอสบีบนสุดมีค่าเท่ากับ 0 สัญญาณ 10 บิต จากวงจรสร้างเฟสจะถูกส่งต่อโดยตรงไปยังวงจรแปลงเป็นอนาลอก แต่ถ้าบิตเอ็มเอสบีบนสุดมีค่าเท่ากับ 1 สัญญาณ 10 บิต จากวงจรสร้างเฟสจะถูกคอมพลีเมนต์ก่อนที่จะส่งต่อไปยังวงจรแปลงดิจิตอลเป็นอนาลอก

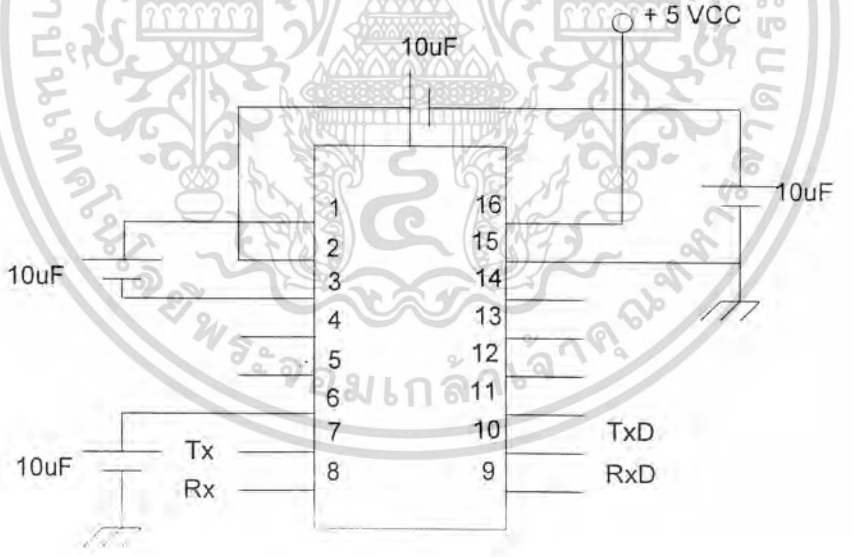
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 วงจรแปลงระดับแรงดัน

ในการเชื่อมต่อวงจรที่ทำงานระดับแรงดันแบบทีทีแอล เข้ากับพอร์ต RS-232ของเครื่องคอมพิวเตอร์ที่มีระดับแรงดัน -15 โวลต์ ถึง +15 โวลต์ นั้นจะต้องมีวงจรพิเศษเพื่อทำการเปลี่ยนแปลงระดับแรงดันให้เหมาะสมซึ่งในที่นี้เราได้เลือกใช้ MAX 232 ซึ่งใช้อุปกรณ์ประกอบจากภายนอกน้อย คือ ใช้ C เพียง 5 ตัวเท่านั้น

3.6.1 หลักการทำงาน

จากวงจรจะใช้ C ที่ต่อระหว่างขา 1 กับ 3, ระหว่างขา 4 กับ 5, ขา 2 กับ 6 เป็นตัวกำหนดระดับแรงดันที่ใช้ในการเชื่อมต่อโดยขา R11และ R10 จะเป็นขาที่รับระดับแรงดัน -15 โวลต์ ถึง +15 โวลต์ และแปลงออกเป็นแรงดัน 0 โวลต์ และ +5 โวลต์ ตามลำดับ ออกที่ขา R10 และ R20 ส่วนขา T10และ T11จะรับแรงดันที่เป็น 0 โวลต์ และ +5 โวลต์ ตามลำดับ แปลงเป็นระดับแรงดัน -10 โวลต์ +10 โวลต์ ออกที่ขา 7 และ 10

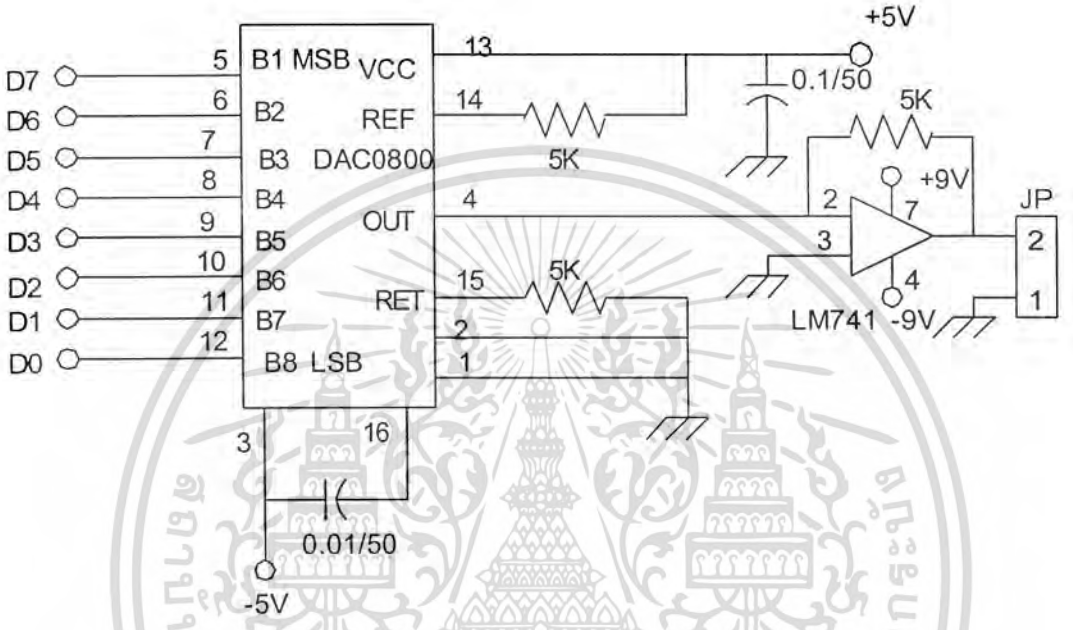


รูปที่ 3.14 วงจรแปลงระดับแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 วงจรแปลงระดับสัญญาณดิจิตอลเป็นอนาล็อก

เป็นส่วนที่นำสัญญาณ Output จาก FPGA ที่เป็นสัญญาณแบบดิจิตอล ขนาด 8 bit มาแปลงให้สัญญาณ Analog โดยใช้ IC DAC 0800 วงจรใช้งานแสดงดังรูป

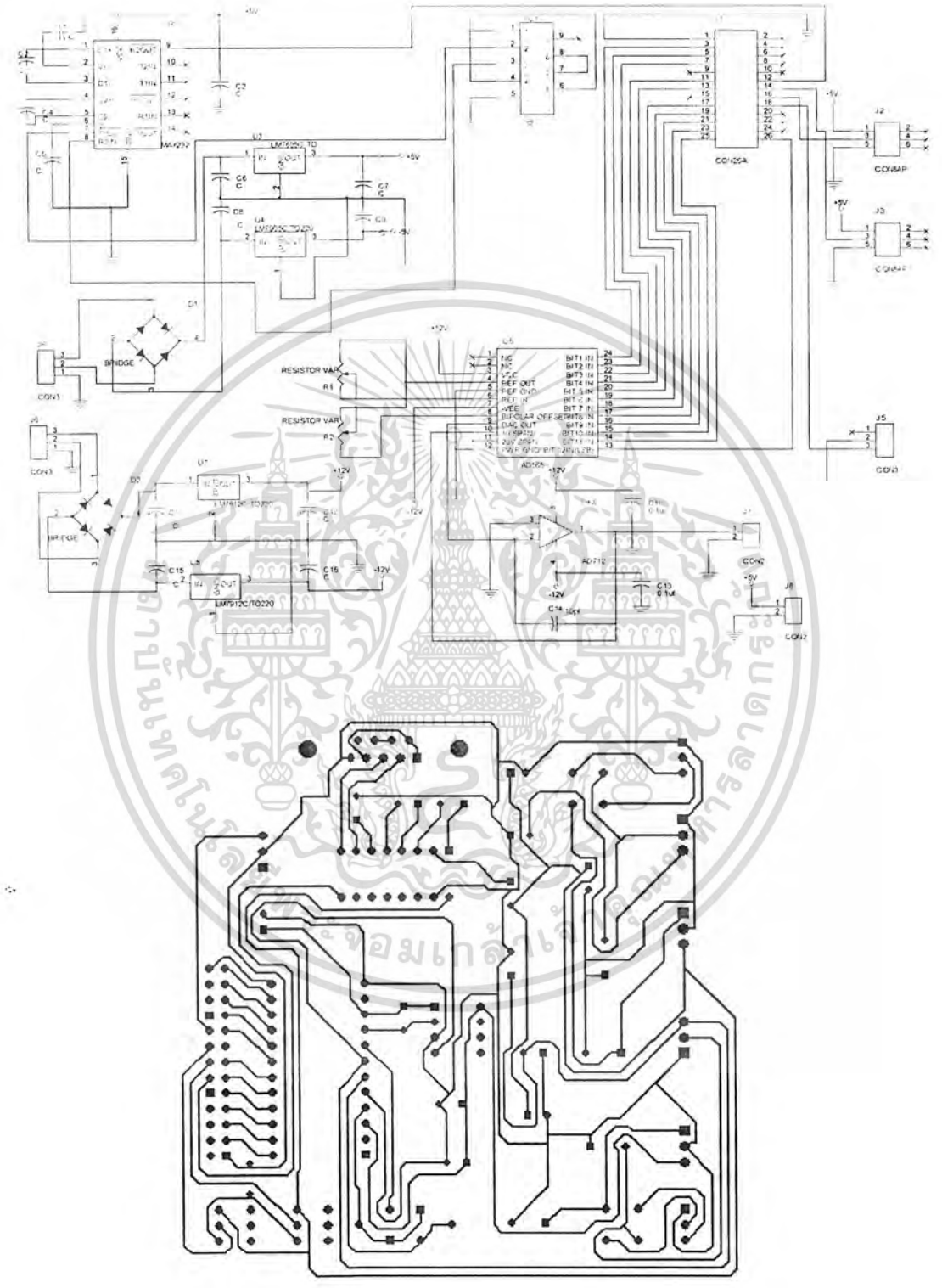


รูปที่ 3.15 วงจรแปลงระดับสัญญาณดิจิตอลเป็นอนาล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8 การออกแบบวงจรและลายวงจร

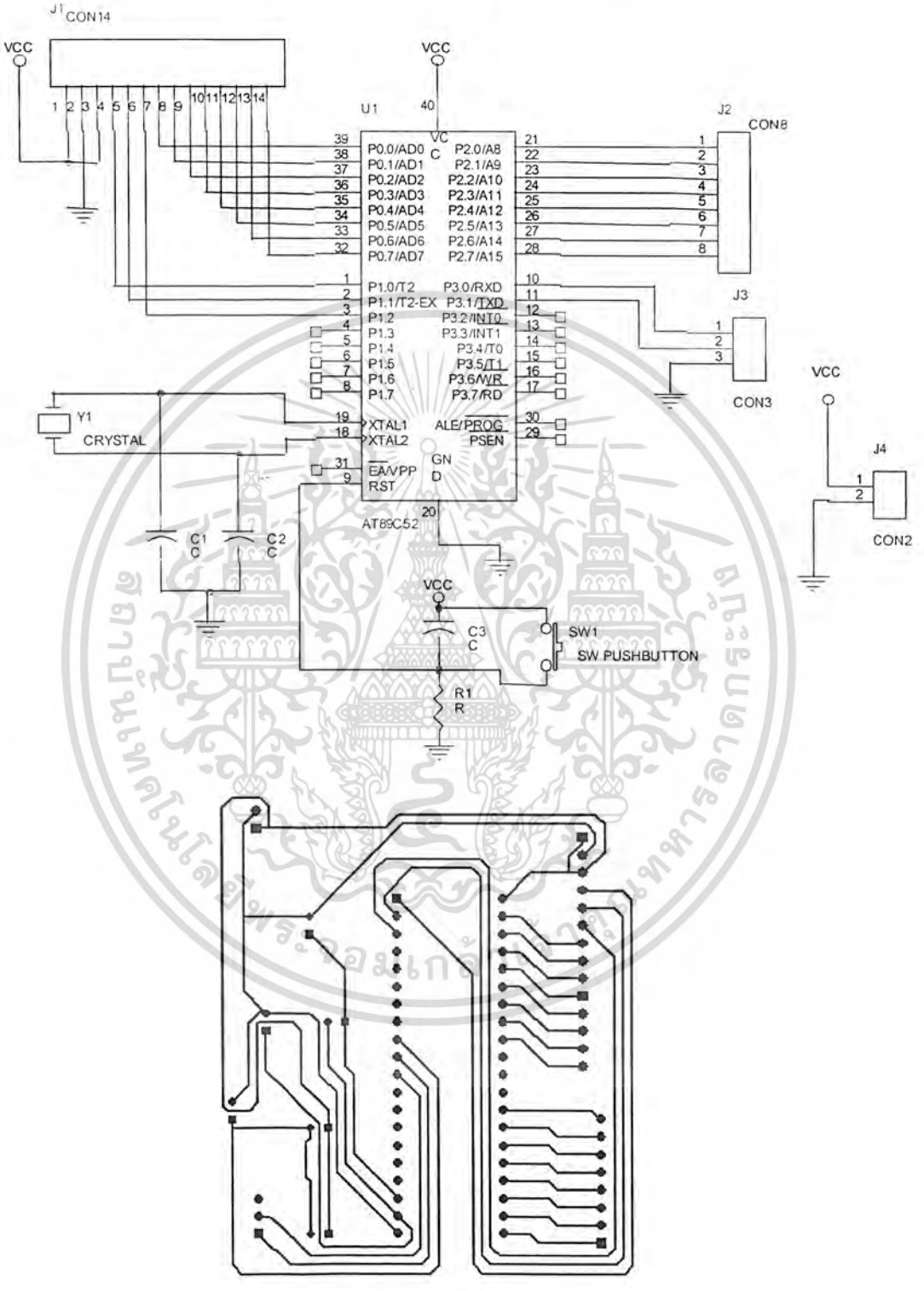
3.8.1 การออกแบบวงจรและลายวงจรส่วนของวงจรแปลงดิจิตอลเป็นอนาล็อก



รูปที่ 3.16 วงจรแปลงดิจิตอลเป็นอนาล็อกและลายวงจรแปลงดิจิตอลเป็นอนาล็อก

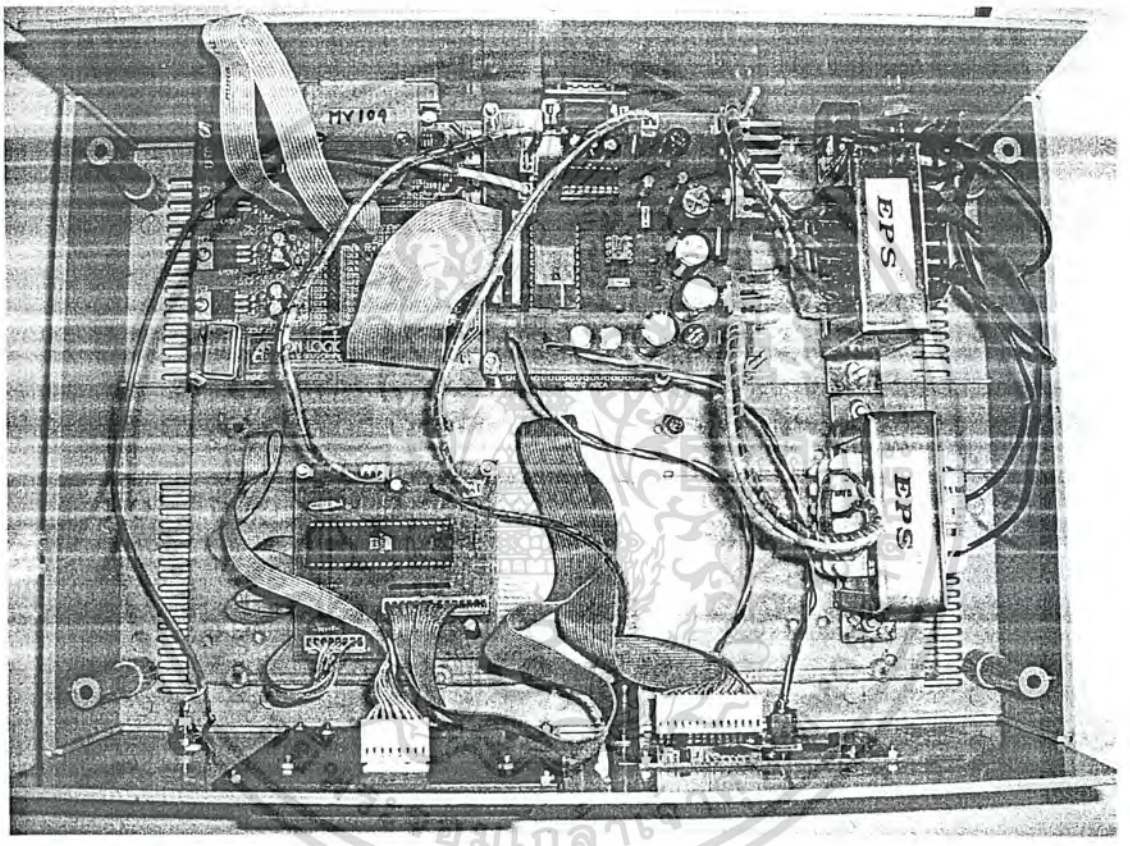
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.8.2 การออกแบบวงจรและลายวงจรส่วนของวงจรส่งข้อมูลผ่านพอร์ตอนุกรมโดยใช้MCS-51



รูปที่ 3.17 วงจรส่งข้อมูลผ่านพอร์ตอนุกรมและลายวงจรส่งข้อมูลมาจากMCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 ภาพถ่ายอุปกรณ์และวงจรใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทำงาน

การออกแบบวงจรกำเนิดสัญญาณไซน์แบบดิจิตอล ประกอบด้วย 2 ส่วนคือ ส่านของโครงสร้างวงจรกำเนิดสัญญาณแบบดิจิตอลโดยวิธีการใช้ตารางเปิดดู (Look-up table) และส่วนของโครงสร้างวงจรกำเนิดสัญญาณแบบดิจิตอลโดยใช้วิธีการของซันเดอร์แลนด์ (Sunderlans Algorithm) จะใช้การอธิบายการพฤติกรรมการทำงานของวงจร ด้วยภาษา VHDL และการจำลองการทำงาน

4.1 การควบคุมการใช้งานการเชื่อมต่อผ่านพอร์ตอนุกรม

การควบคุมการทำงานของบอร์ดสามารถทำได้สองวิธี (1) ใช้โปรแกรมคอมพิวเตอร์ควบคุมการทำงานผ่านพอร์ตอนุกรม (RS-232) และ (2) ใช้MCS-51 ควบคุมการทำงานผ่านทางพอร์ตอนุกรม ซึ่งจะกล่าวถึงรายละเอียดของการทำงานในแต่ละแบบในหัวข้อถัดไป

ในการควบคุมการทำงานในแต่ละวิธีมีการรูปแบบการทำงานดังต่อไปนี้

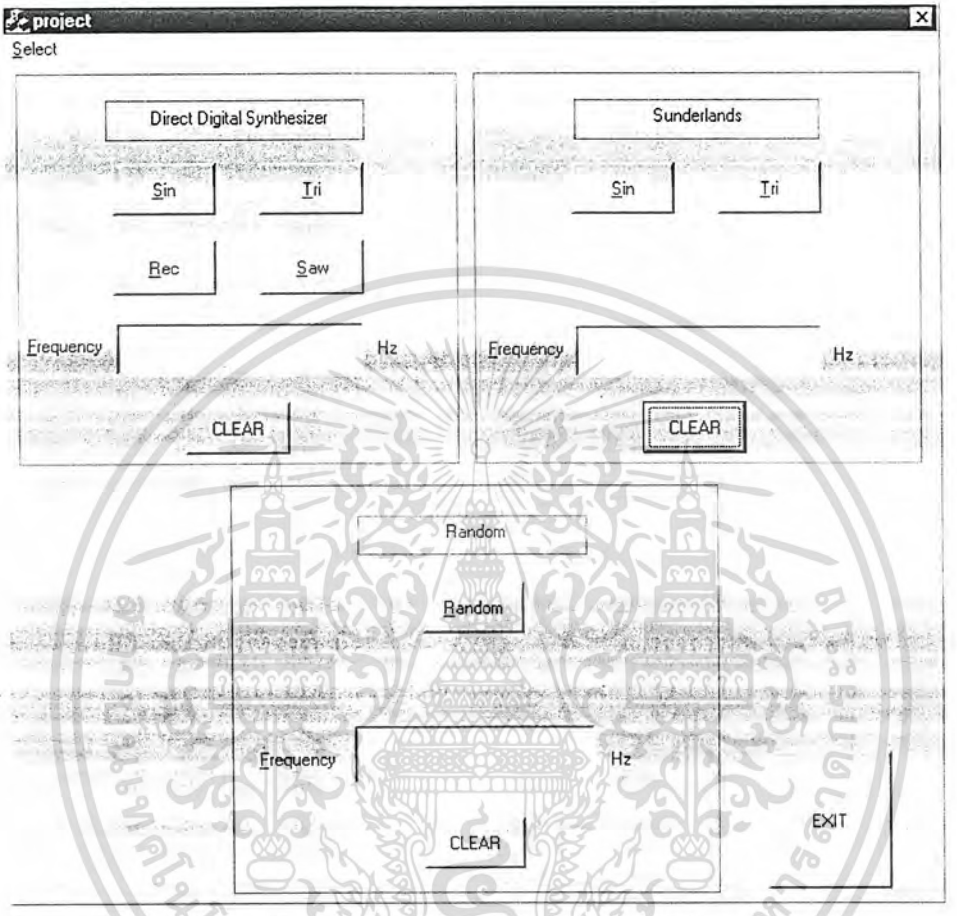
Address(hex)	Register Name	Description
01	RegINC0	Bit7-Bit0 of frequency control word(W) $fout = (f_{clk} * W) / 2^{32}$ Where fout is output frequency Fclk is MCLK frequency W is 32-frequency control word
02	RegINC1	Bit15-Bit8 of frequency control word(W)
03	RegINC2	Bit23-Bit16 of frequency control word(W)
04	RegINC3	Bit31-Bit24 of frequency control word(W)
05	RegCTRL	DDS control register used for controlling DDS operation 0000000 Sinusindom(Look up table) 0000001 Rectangular(Look up table) 0000010 Tringular(Look up table) 0000011 Sawthooth(Look up table) 0000100 Sinusindom(Sunderlans) 0000101 Tringular(Sunderlans) 0000110 Random(LFSR)

ตารางที่ 4.1 รูปแบบการส่งข้อมูลผ่านทางพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการใช้งานการเชื่อมต่อผ่านพอร์ตอนุกรมโดยใช้ วิชวลซี

การทำงานของโปรแกรมที่เขียนด้วยวิชวลซี มีความสะดวกในการเชื่อมต่อผ่านพอร์ตอนุกรมเรา จึงนำมาประยุกต์ใช้งานร่วมกับ Oscillator



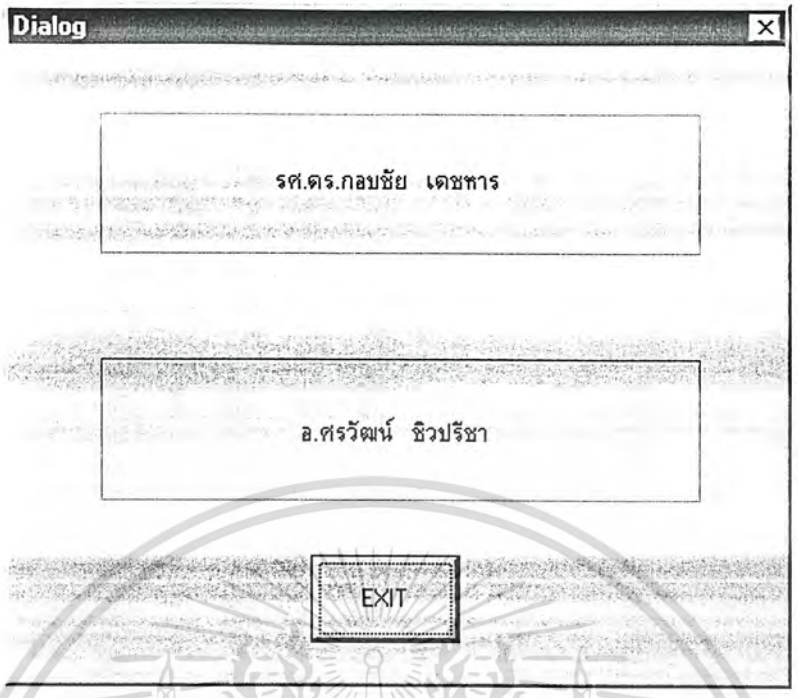
รูปที่ 4.1 แสดงเมนูหลักของโปรแกรมการใช้งานดิจิตอลออสซิลเลเตอร์

จากรูปที่ 4.1 จะแสดงวิธีการเรียกใช้ออสซิลเลเตอร์โดยจะแบ่งเป็นสามส่วนหลักดังนี้

1. ส่วนของ Direct Digital Synthesizer ใช้สำหรับเรียกใช้การสร้างสัญญาณแบบตารางเปิดดู
2. ส่วนของ Sunderlands ใช้สำหรับเรียกใช้โดยวิธีการสร้างแบบซันเดอร์แลนด์
3. ส่วนของ Random ใช้สำหรับเรียกใช้โดยการสร้างสัญญาณสุ่ม

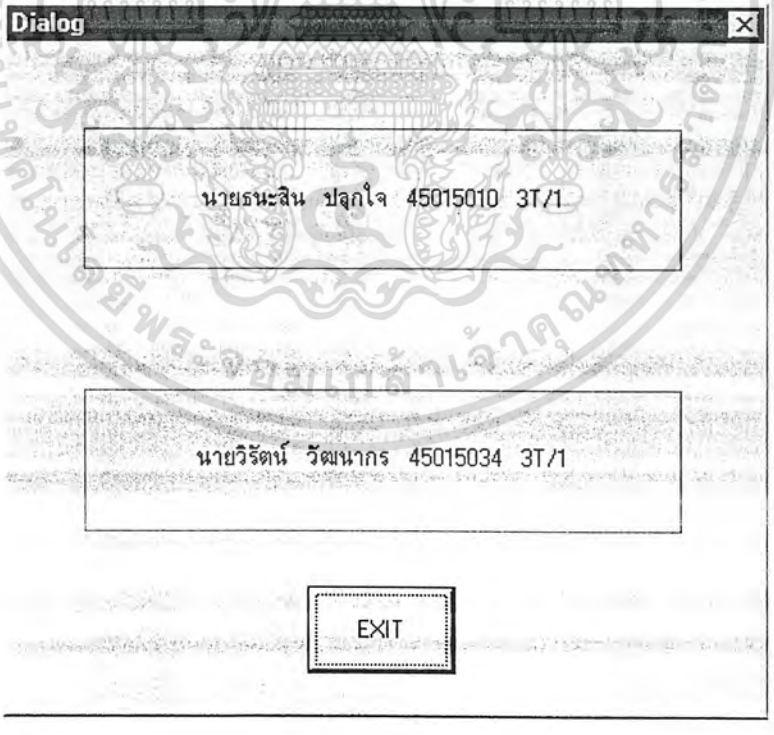
ในการเชื่อมต่อเมนูย่อยจะเลือกที่ Select จะแสดงเมนูของ Advisor และ Project Create ขึ้นมาดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงเมนู Advisor

จากรูปที่ 4.2 แสดงรายละเอียดของอาจารย์ที่ปรึกษาของการทำโปรเจ็ก



รูปที่ 4.3 แสดงเมนู Project Create

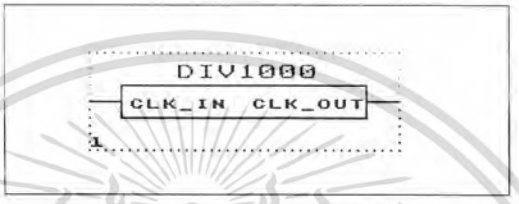
จากรูปที่ 4.3 แสดงรายละเอียดของผู้จัดทำของโปรเจ็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การออกแบบวงจรส่วนต่างๆ โดยใช้ภาษา VHDL

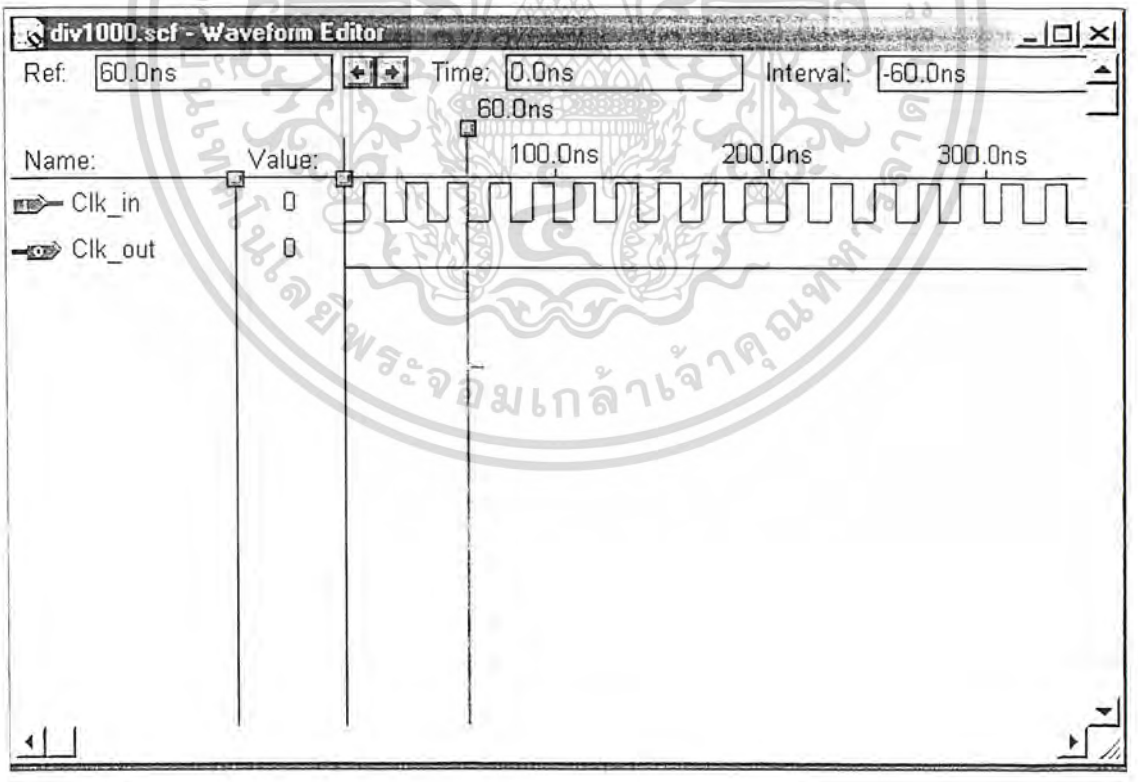
4.2.1 ส่วนของวงจร DIV 1000

ทำหน้าที่เป็นโมดูลของวงจรหารความถี่ เนื่องจากภายในบอร์ด ACEX1K-50 ใช้แหล่งกำเนิดแบบ Module Oscillator ความถี่ 9.6MHz แต่ความถี่ที่ได้จะใช้เป็น Baud Rate ที่เราต้องการในการรับ มีค่าเท่ากับ 9600MHz เพราะฉะนั้นเราจะต้องทำการหารความถี่ที่ได้จาก Module Oscillator ลงให้เหลือเพียง 9600 MHz ซึ่งค่าของตัวหารจะเท่ากับ 9.6MHz / 9600 Hz เท่ากับ 1000
มีลักษณะดังรูป



รูปที่4.4 สัญลักษณ์ของส่วนวงจร DIV 1000

จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงาน (simulation)ได้ดังนี้

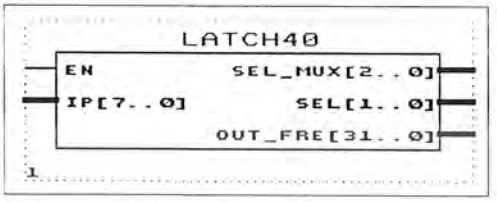


รูปที่ 4.5 ผลการจำลองการทำงานส่วนของวงจร DIV 1000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

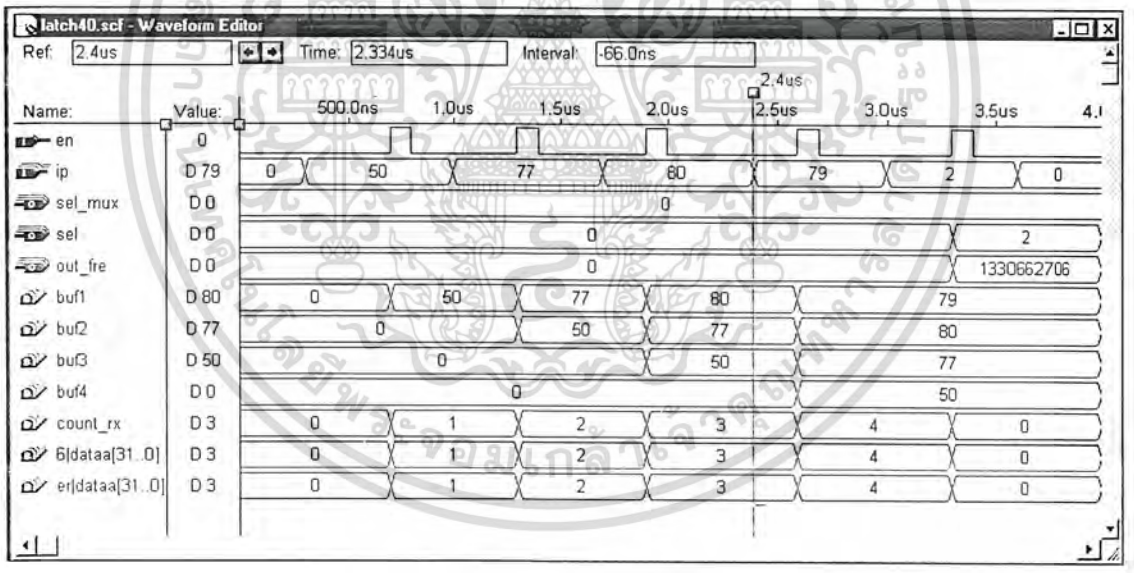
4.2.2 ส่วนของวงจร LATCH

เป็นโมดูลของวงจร D Flip Flop ทำหน้าที่ Latch ค่าข้อมูลที่รับมาได้ และส่งค่าที่ได้รับมาให้กับส่วนต่างๆ ที่จะนำไปใช้งาน มีลักษณะดังรูป



รูปที่4.6 สัญลักษณ์ของส่วนวงจร LATCH

จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงาน (simulation) ได้ดังนี้

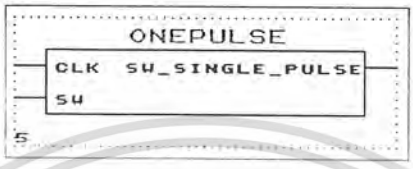


รูปที่ 4.7 ผลการจำลองการทำงานของส่วนวงจร LATCH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

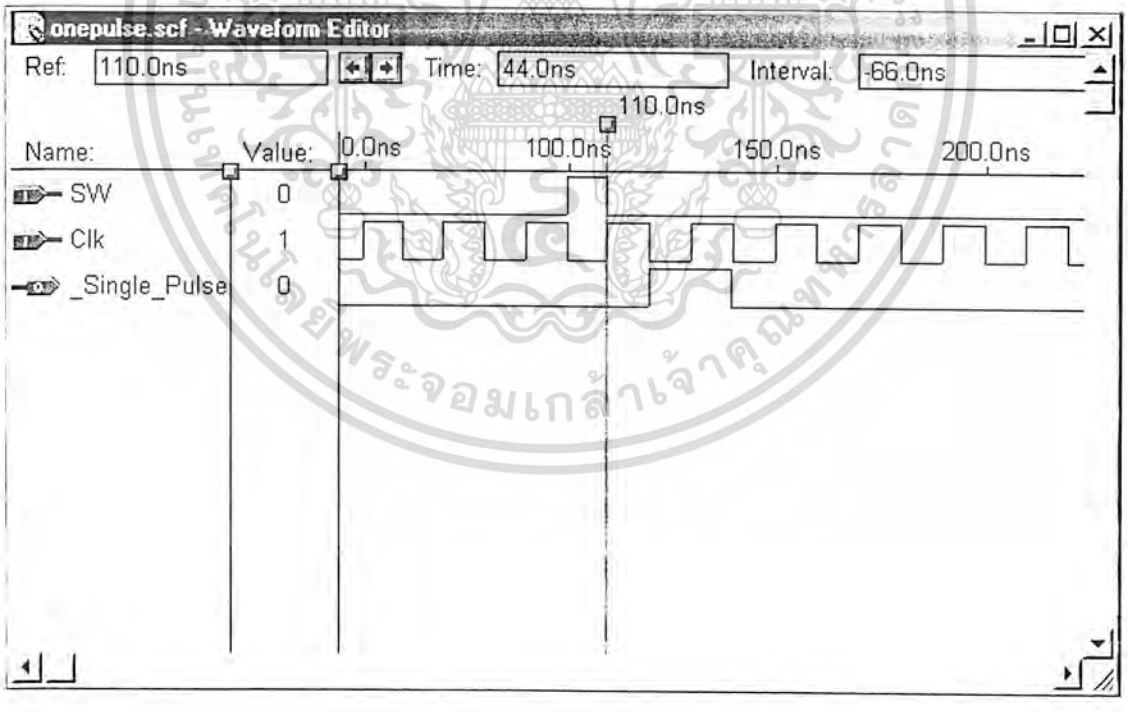
4.2.3 ส่วนของวงจร ONEPULSE

เป็นโมดูลรับค่าจากส่วนของ RX_EN ของส่วน SERIAL_COMMUNICATION โมดูลนี้จะทำการ Debounce สัญญาณที่รับและให้ Pulse ออกมา 1 ลูกเท่านั้น ขนาดของ Pulse จะเท่ากับความถี่ที่ต่ออยู่กับอินพุต CLK ซึ่งมีลักษณะดังรูป



รูปที่ 4.8 สัญลักษณ์ของส่วนวงจร ONEPULSE

จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงาน (simulation) ได้ดังนี้

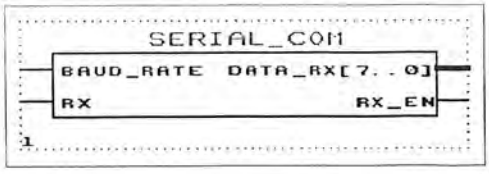


รูปที่4.9 ผลการจำลองการทำงานของส่วนวงจร ONEPULSE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

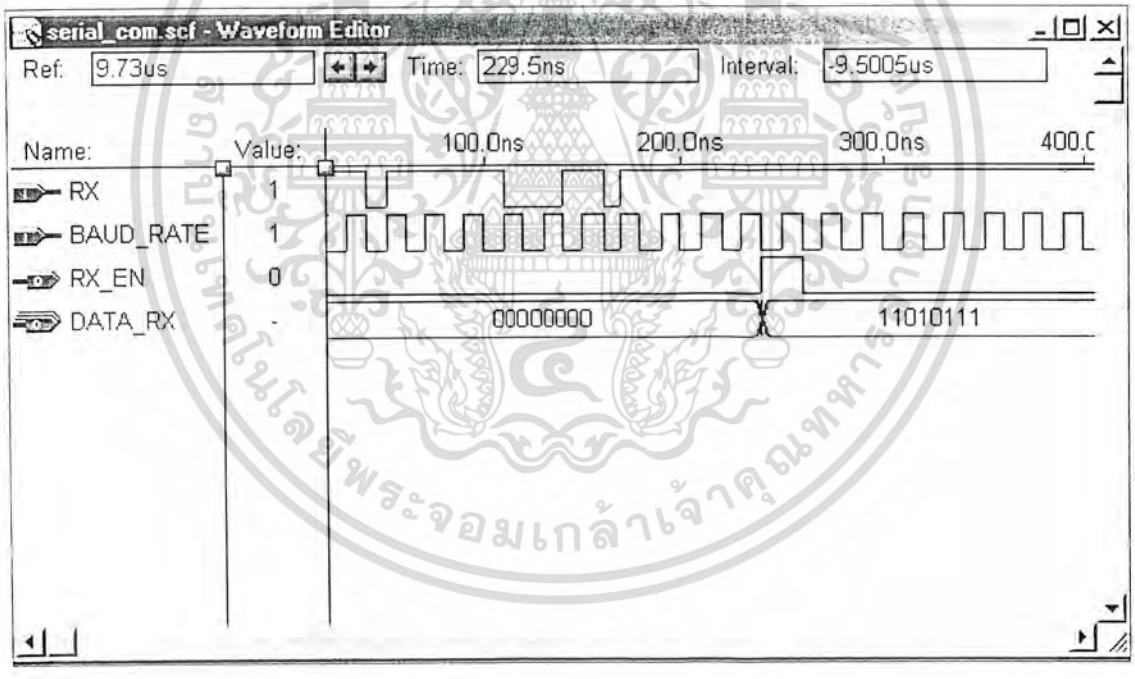
4.2.4 ส่วนของวงจร SERIAL_COMMUNICATION

เป็นโมดูลของวงจรรับ-ส่งข้อมูลแบบอนุกรม ที่รับค่ามาจากพอร์ตอนุกรมของคอมพิวเตอร์ และผ่านวงจรปรับระดับแรงดันก่อนที่จะส่งเข้ามาที่เอฟพีจีเอ มีลักษณะดังรูป



รูปที่4.10 สัญลักษณ์ของส่วนวงจร SERIAL_COMMUNICATION

จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงาน (simulation) ได้ดังนี้

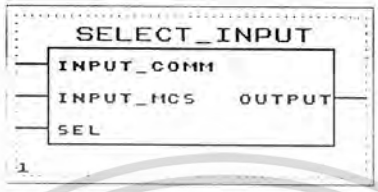


รูปที่ 4.11 ผลการจำลองการทำงานของส่วนวงจร SERIAL_COMMUNICATION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

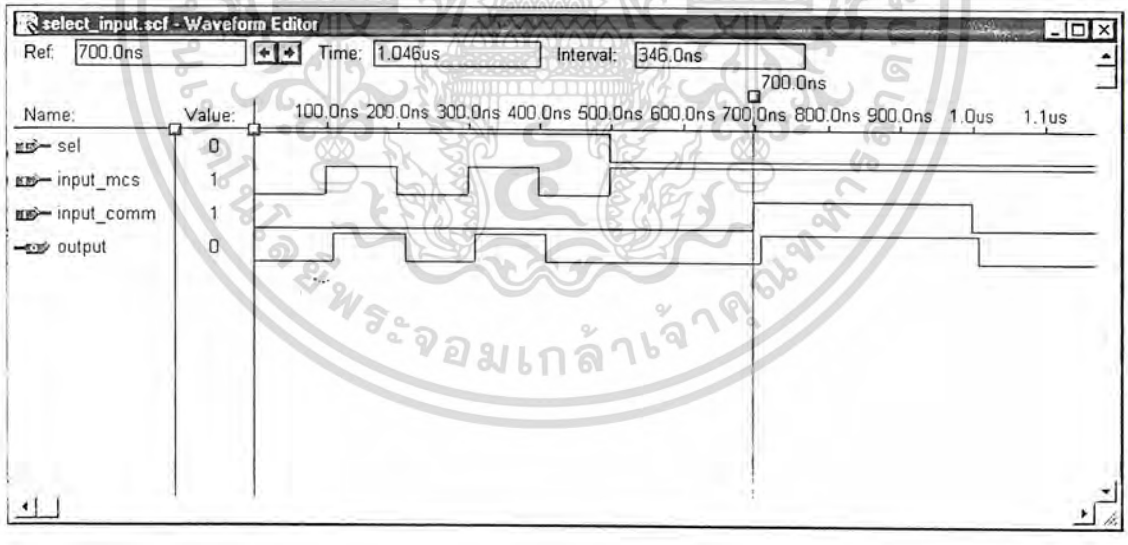
4.2.5 ส่วนของวงจร SELECT_INPUT

เป็นโมดูลที่ทำหน้าที่เลือกอินพุทที่รับเข้ามาในส่วนของพอร์ตอนุกรมว่ามาคอมพิวเตอร์หรือมาจากMCS-51 มีลักษณะดังรูป



รูปที่ 4.12 สัญลักษณ์ของส่วนวงจร SELECT_INPUT

จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงาน (simulation) ได้ดังนี้

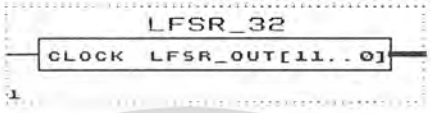


รูปที่ 4.13 ผลการจำลองการทำงานของส่วนวงจร SELECT_INPUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

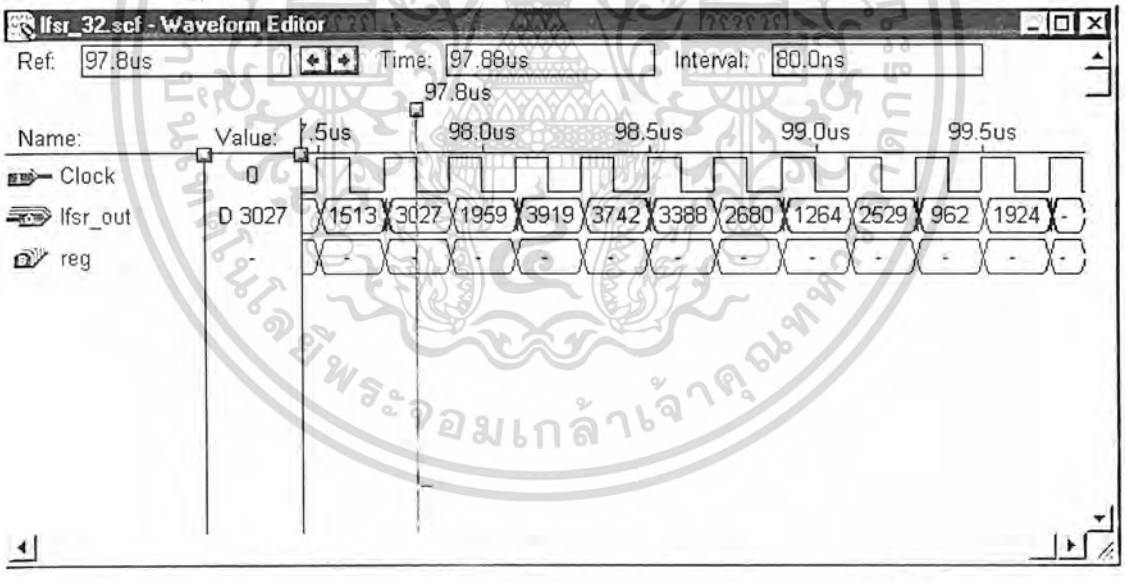
4.2.6 ส่วนของวงจร LSFR

เป็นโมดูลของ LFSR ทำหน้าที่ผลิตสัญญาณ RANDOM โดยใช้โครงสร้างของ Linear Feedback Shift Register มีลักษณะดังรูป



รูปที่ 4.14 สัญลักษณ์ของส่วนวงจร LFSR

จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงาน (simulation) ได้ดังนี้

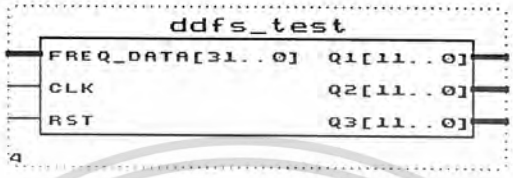


รูปที่ 4.15 ผลการจำลองการทำงานของส่วนวงจร LFSR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

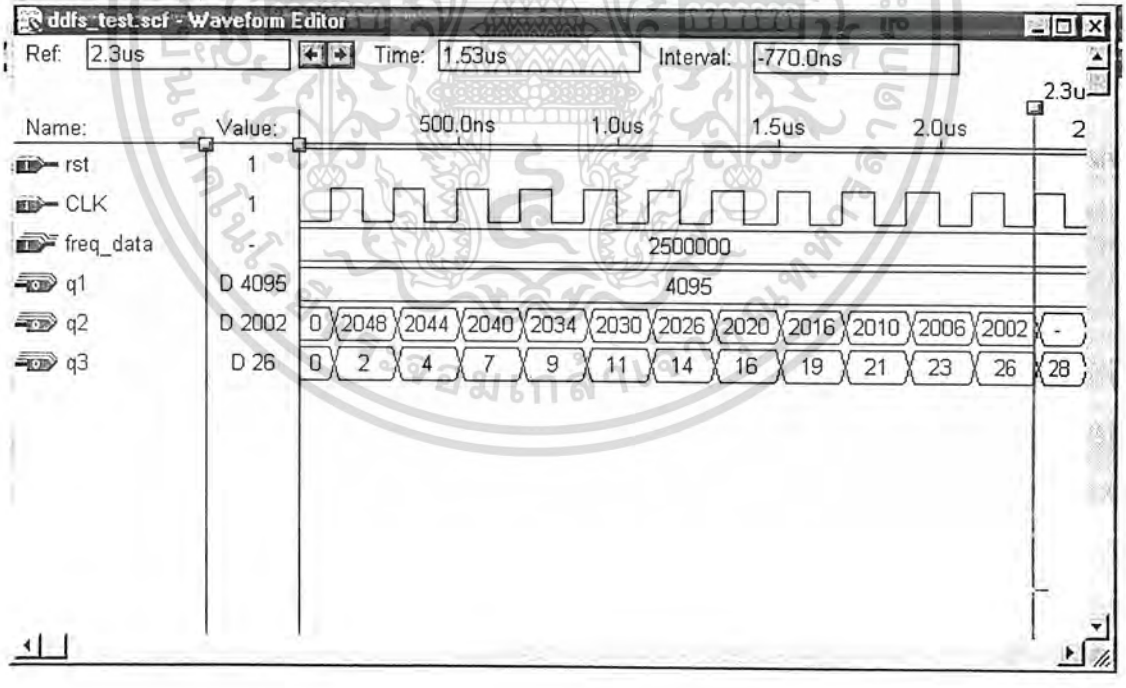
4.2.7 ส่วนของวงจร DDS

เป็นโมดูลที่ทำหน้าที่ผลิตสัญญาณซายน์,สามเหลี่ยม,สี่เหลี่ยมและฟันเลื่อยโดยใช้วิธีตารางเปิดดู (Lookup table)



รูปที่ 4.16 สัญลักษณ์ของส่วนวงจร DDS

จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงาน (simulation) ได้ดังนี้

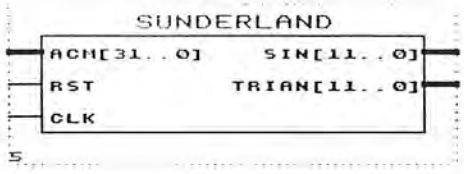


รูปที่ 4.17 ผลการจำลองการทำงานของส่วนวงจร DDS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

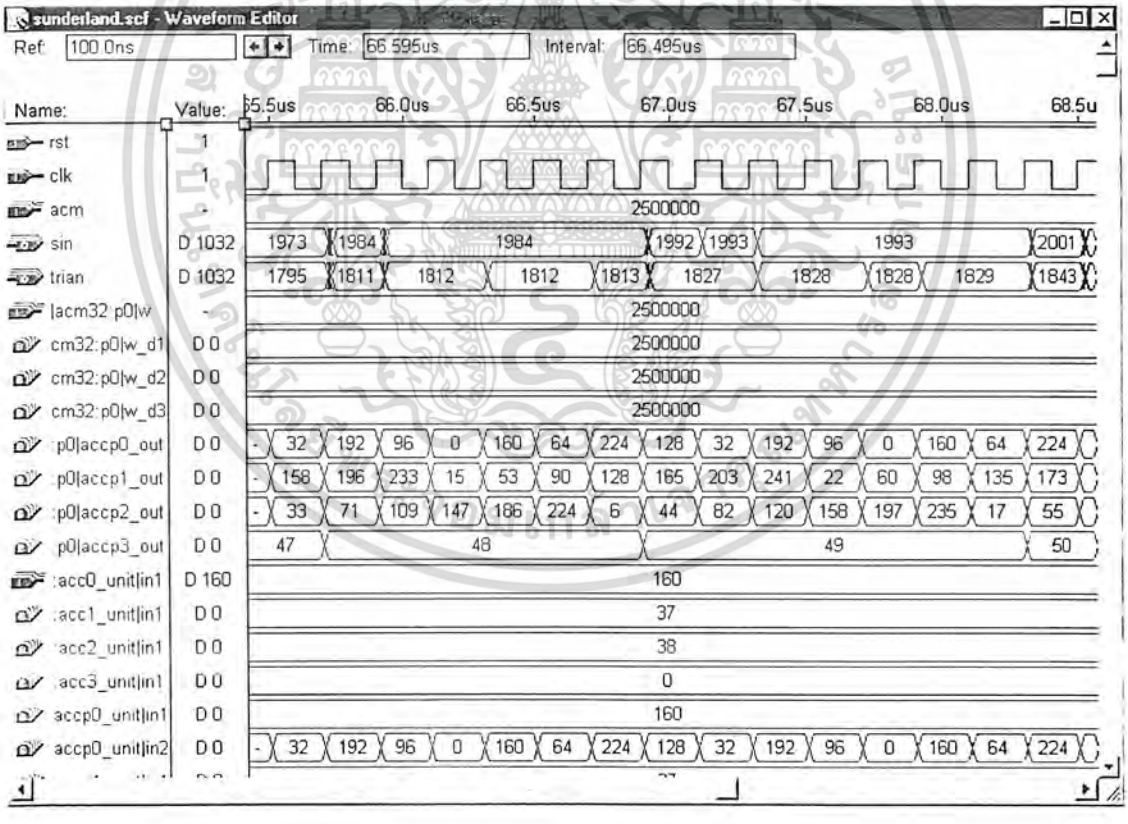
4.2.8 ส่วนของวงจร SUNDERLANS

เป็นโมดูลที่ทำหน้าที่ผลิตสัญญาณซายน์,สามเหลี่ยมโดยใช้วิธีของ sunderlans algorithms



รูปที่ 4.18 สัญลักษณ์ของส่วนวงจร SUNDERLANS

จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงาน (simulation) ได้ดังนี้

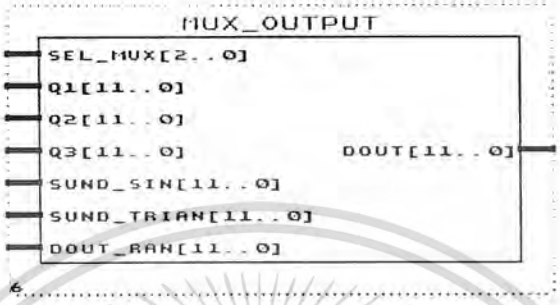


รูปที่ 4.19 ผลการจำลองการทำงานของส่วนวงจร SUNDERLANS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

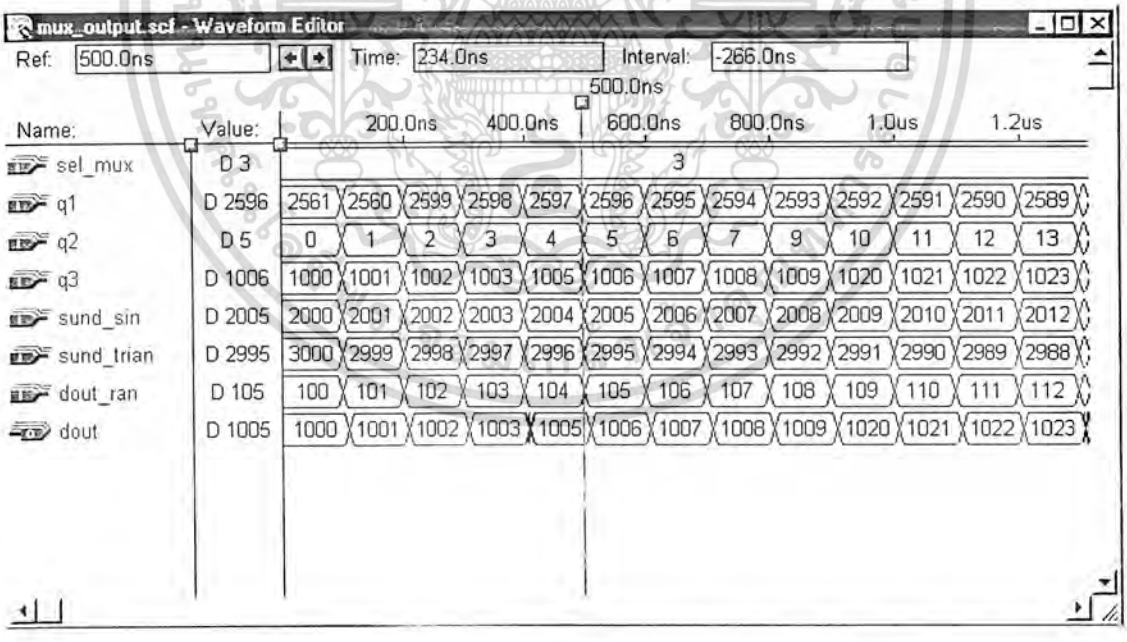
4.2.9 ส่วนของวงจร MUX_OUTPUT

เป็นโมดูลที่ทำหน้าที่ทำการเลือกว่าจะให้เอาต์พุทระหว่าง วิธีใช้ตารางเปิดดู, วิธี sunderlans หรือ ของ LFSR ออกที่เอาต์พุทตามที่ต้องการ



รูปที่ 4.20 สัญลักษณ์ของส่วนวงจร MUX_OUTPUT

จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงาน (simulation) ได้ดังนี้

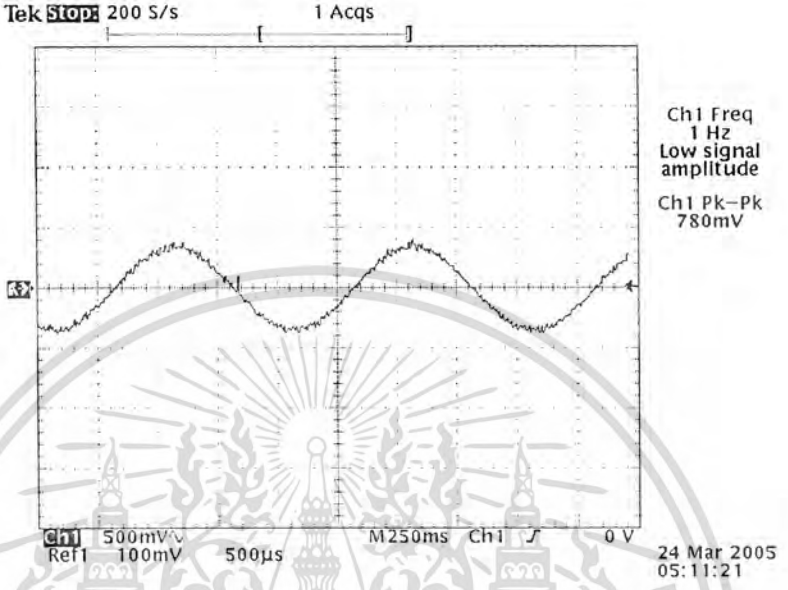


รูปที่ 4.21 ผลการจำลองการทำงานของส่วนวงจร MUX_OUTPUT

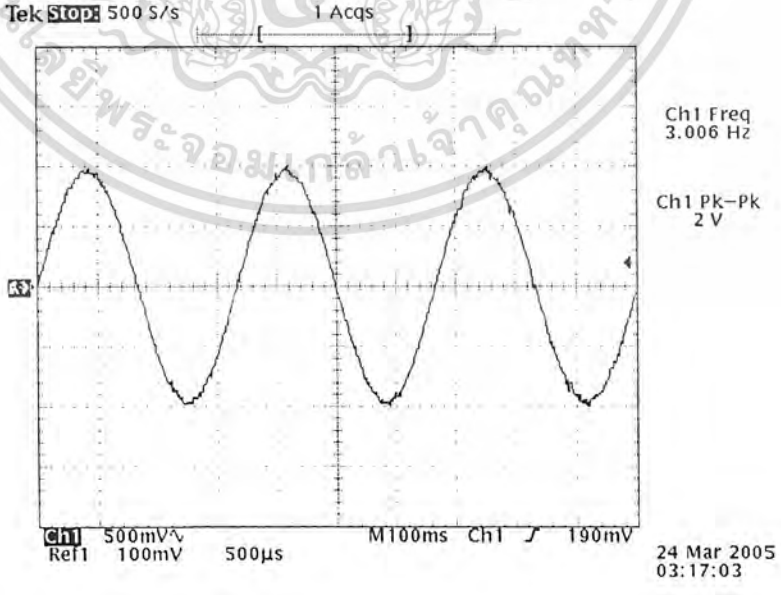
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดสอบการทำงานของ DDS OSCILATOR

4.3.1 การทดสอบการกำเนิดสัญญาณซายน์ที่ความถี่ต่างๆดังนี้

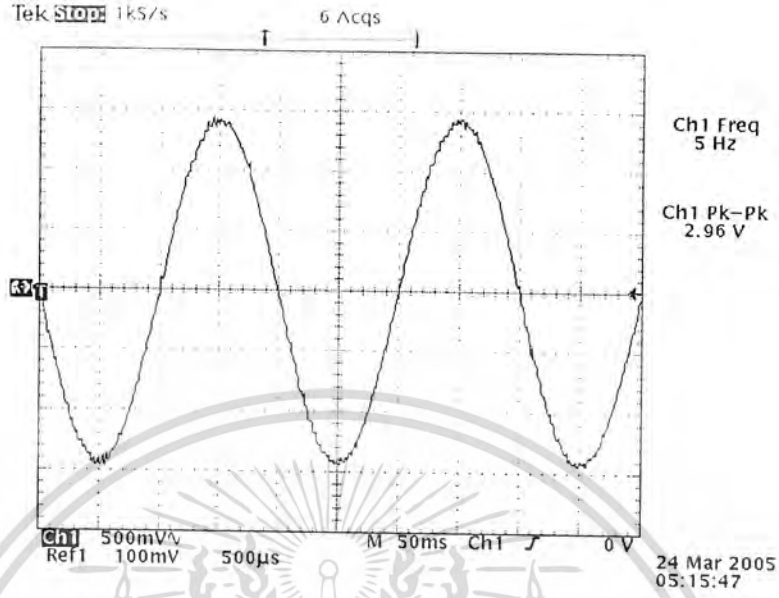


ฑ) รูปแสดงสัญญาณซายน์ที่ความถี่ 1 Hz

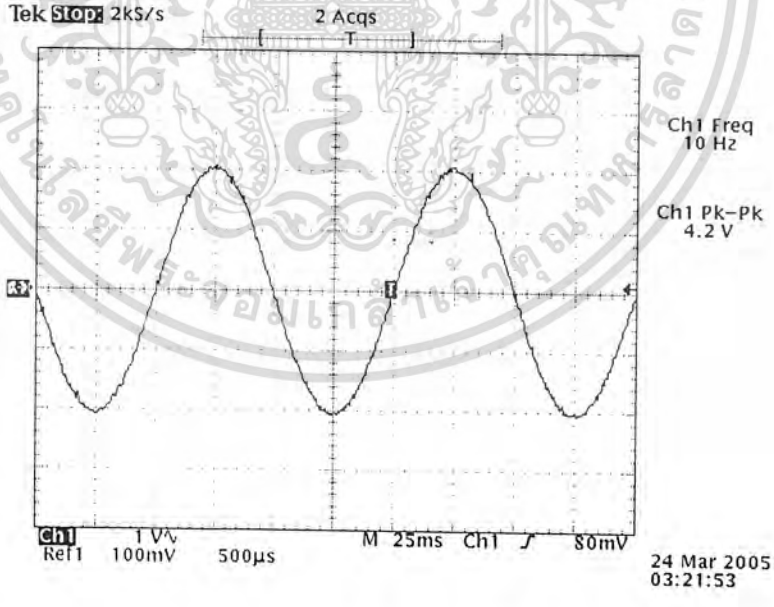


ฒ) รูปแสดงสัญญาณซายน์ที่ความถี่ 3 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

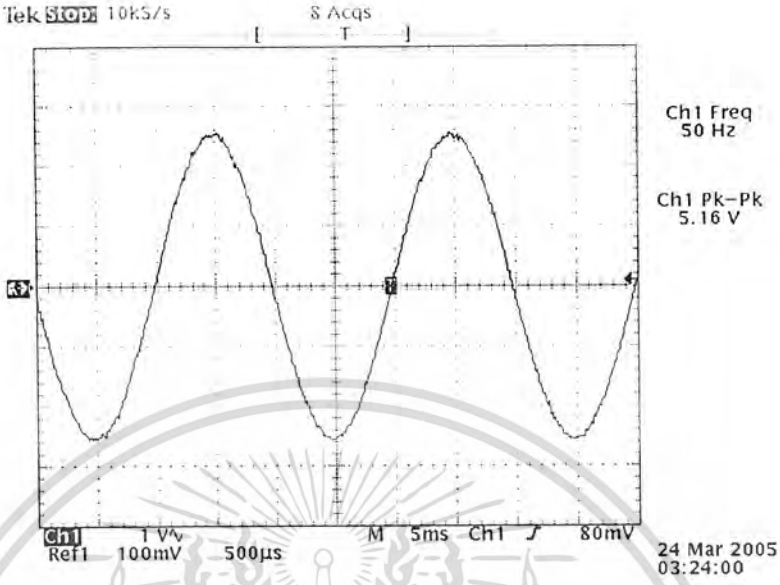


ค) รูปแสดงสัญญาณซายน์ที่ความถี่ 5 Hz

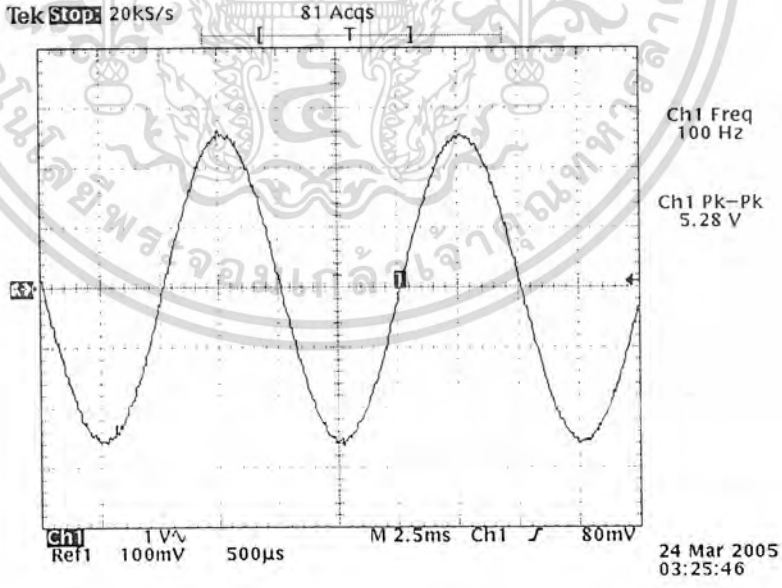


ง) รูปแสดงสัญญาณซายน์ที่ความถี่ 10 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

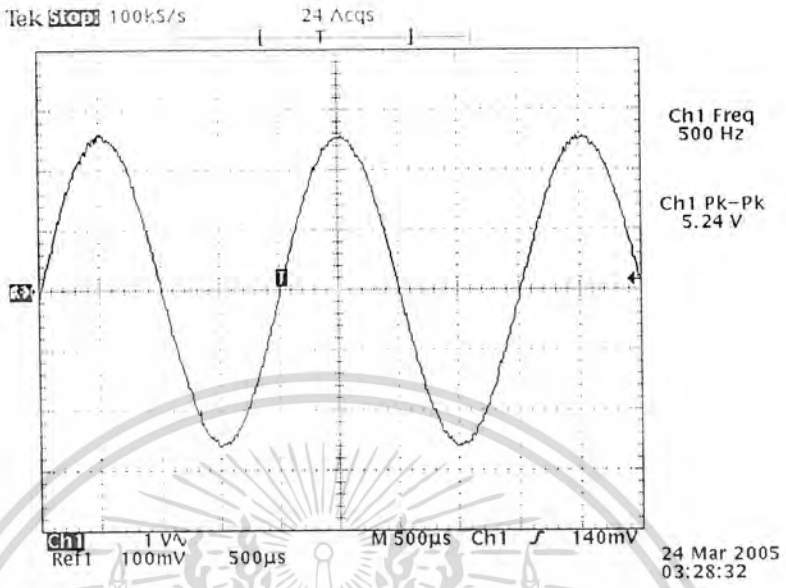


จ) รูปแสดงสัญญาณซายน์ที่ความถี่ 50 Hz

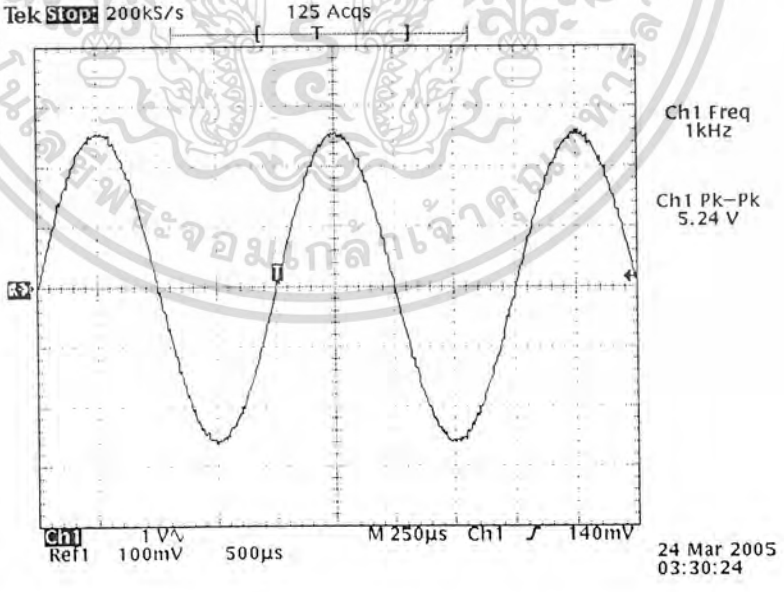


ฉ) รูปแสดงสัญญาณซายน์ที่ความถี่ 100 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

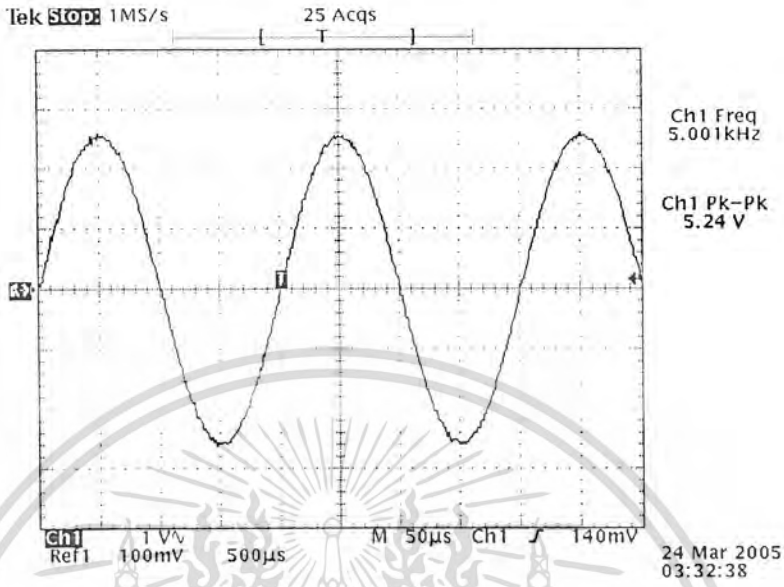


จ) รูปแสดงสัญญาณซายน์ที่ความถี่ 500 Hz

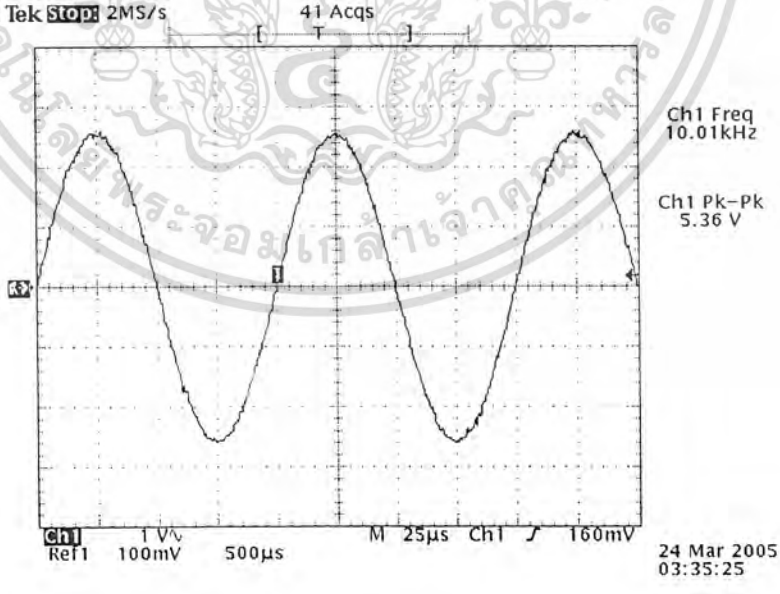


ช) รูปแสดงสัญญาณซายน์ที่ความถี่ 1 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

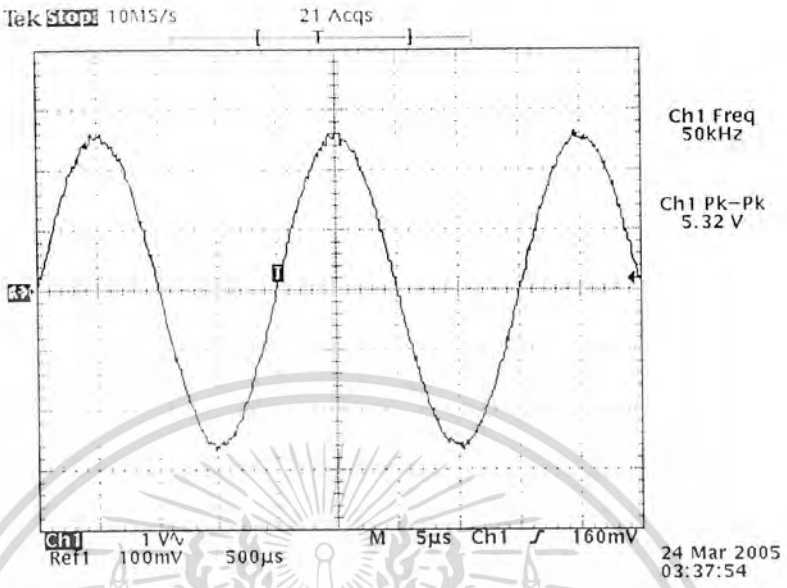


ช) รูปแสดงสัญญาณซายน์ที่ความถี่ 5 kHz

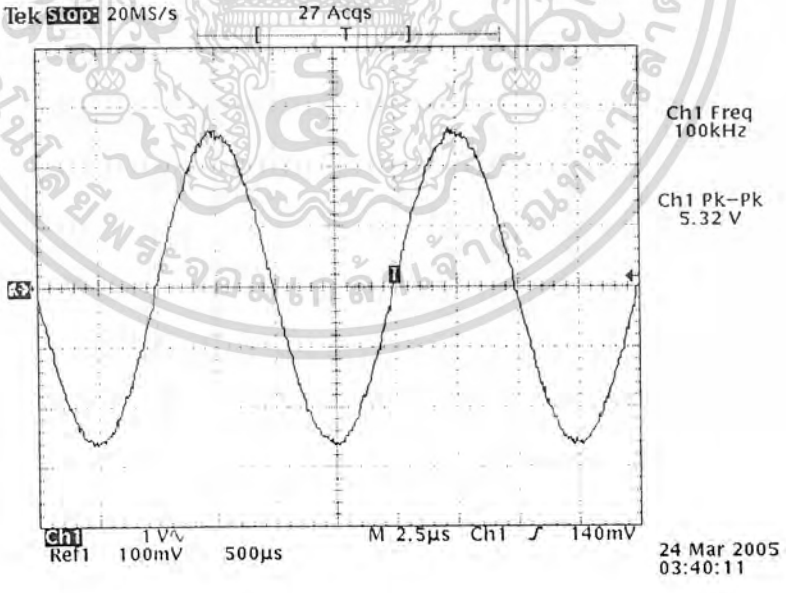


ฌ) รูปแสดงสัญญาณซายน์ที่ความถี่ 10 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ญ) รูปแสดงสัญญาณซายน์ที่ความถี่ 50 kHz

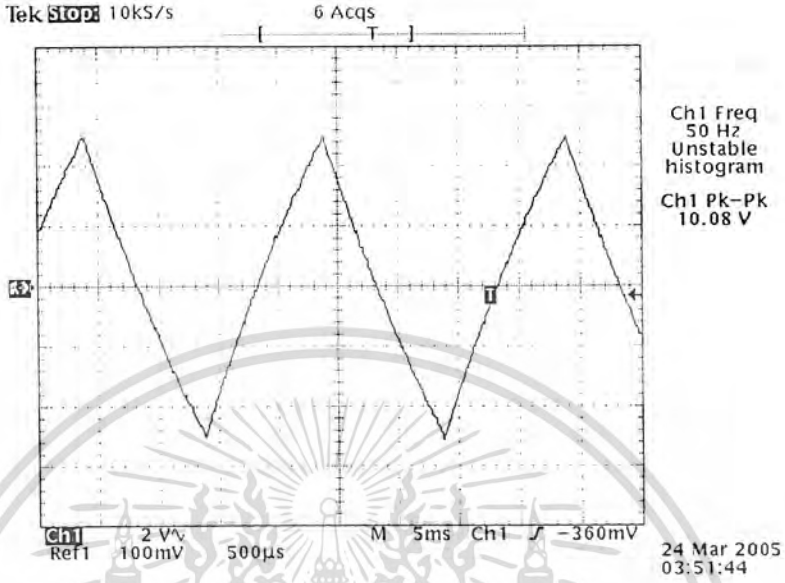


ฎ) รูปแสดงสัญญาณซายน์ที่ความถี่ 100 kHz

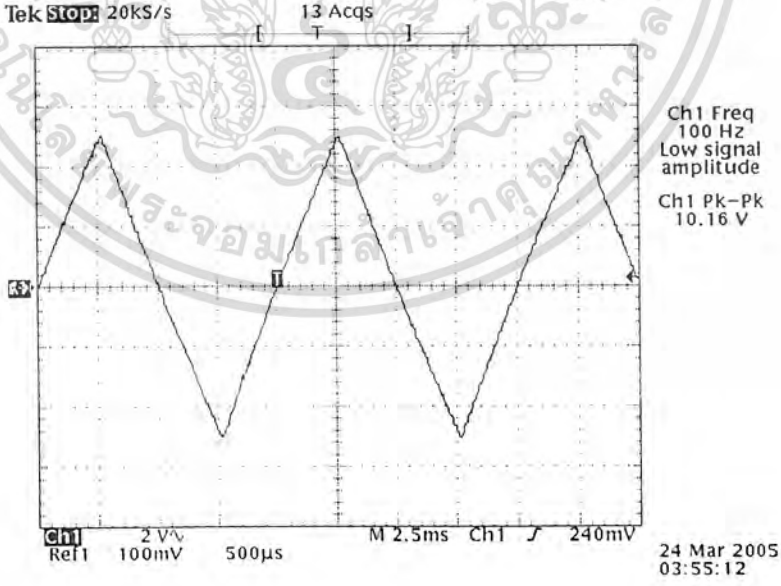
รูปที่ 4.22 แสดงสัญญาณซายน์ที่ความถี่ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 การทดสอบการกำเนิดสัญญาณสามเหลี่ยมที่ความถี่ต่างๆดังนี้

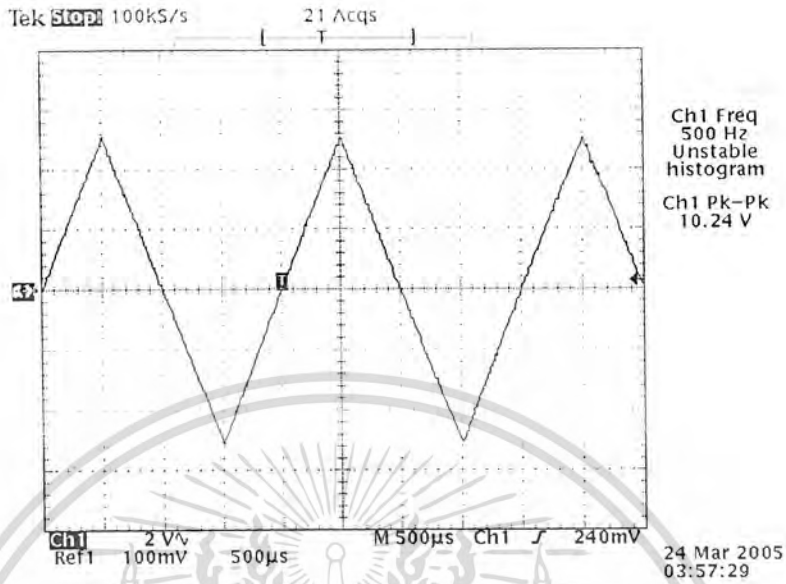


ก) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 50 Hz

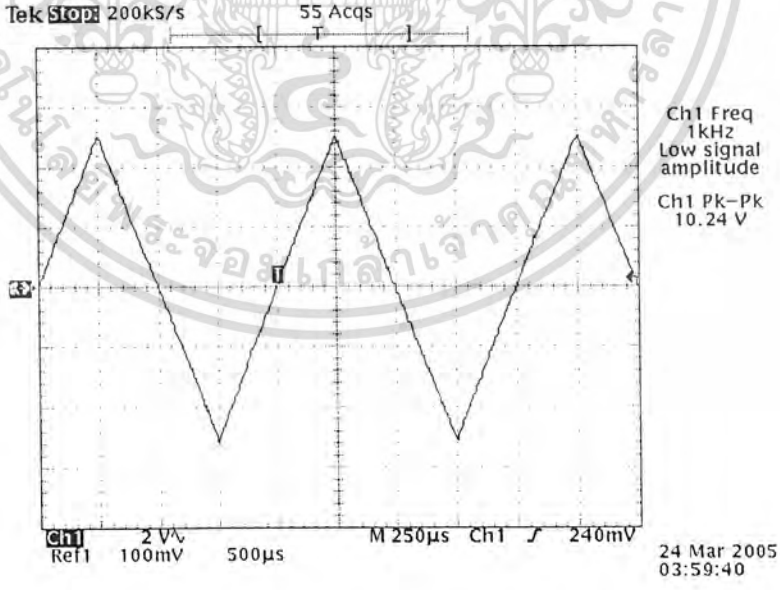


ข) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 100 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

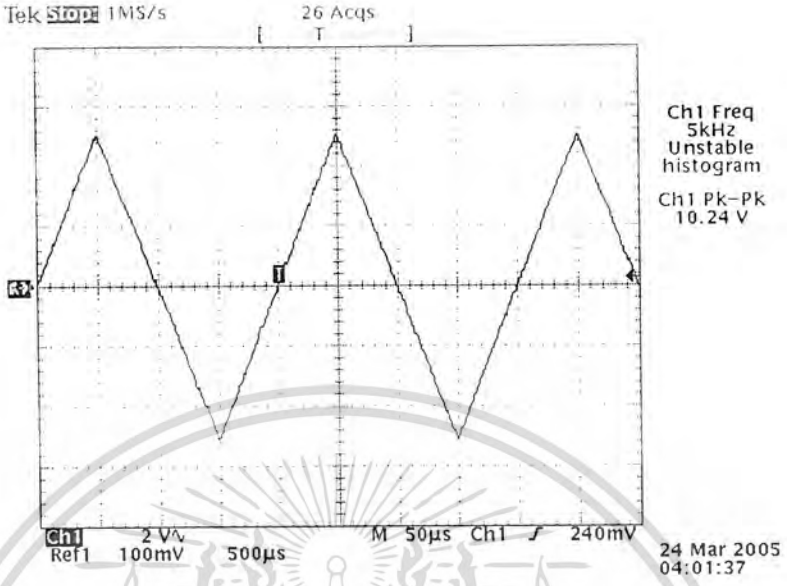


ค) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 500 Hz

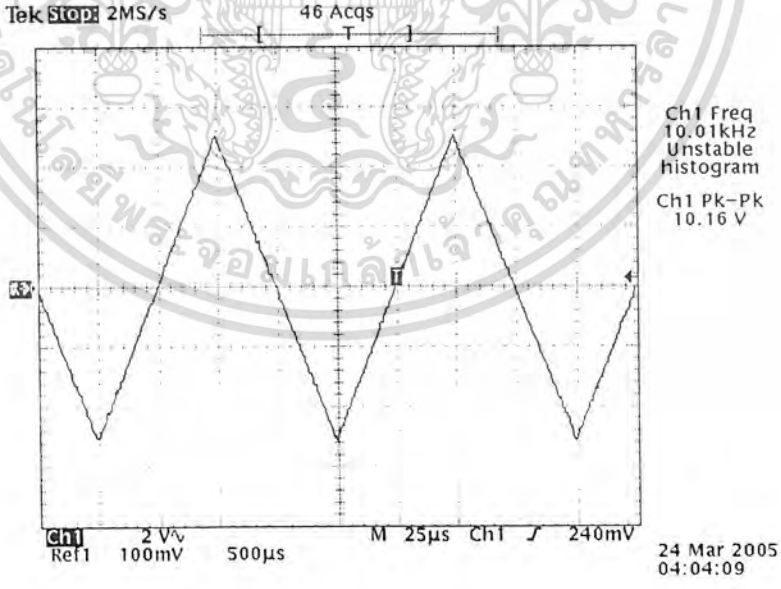


ง) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 1 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

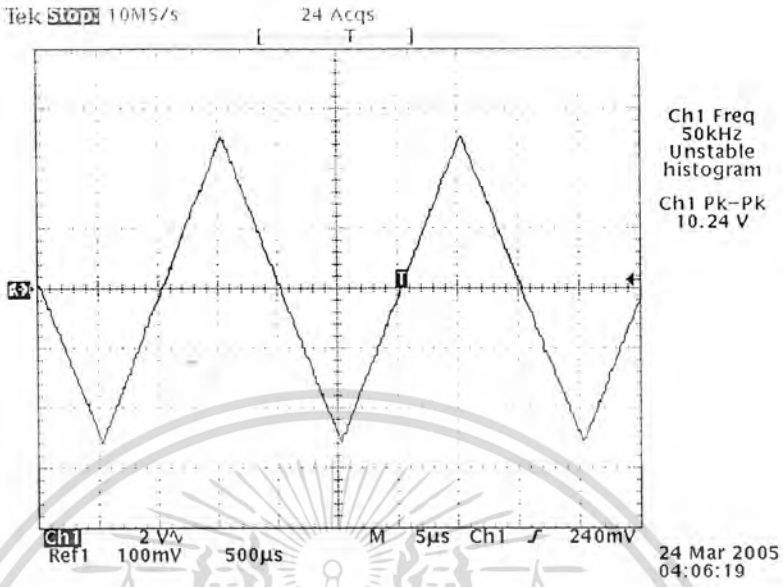


จ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 5 kHz

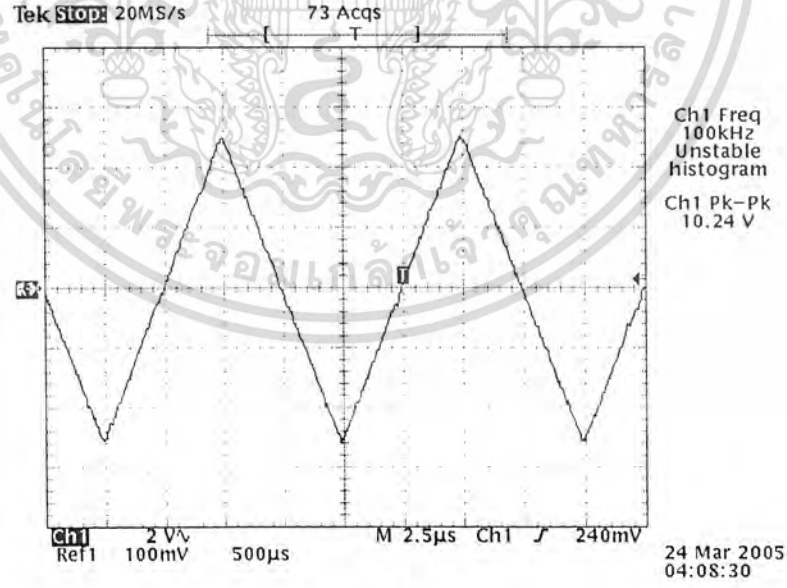


ฉ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 10 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

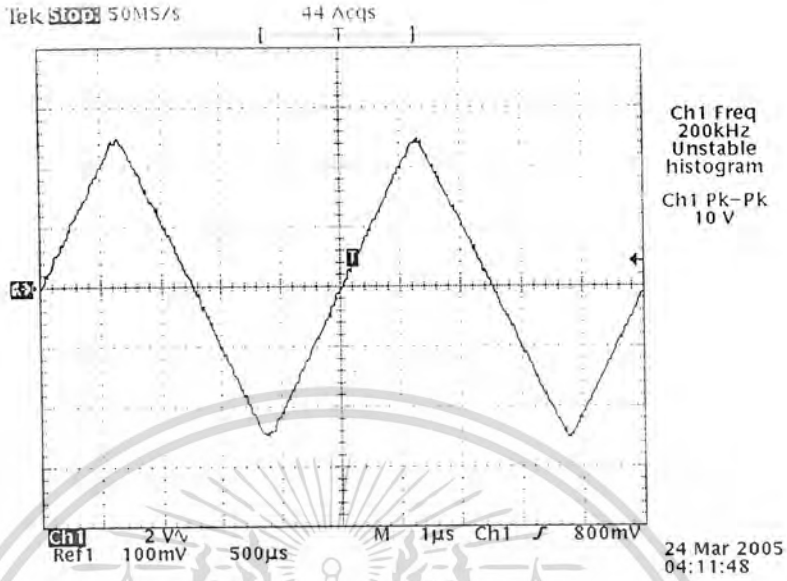


ช) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 50 kHz

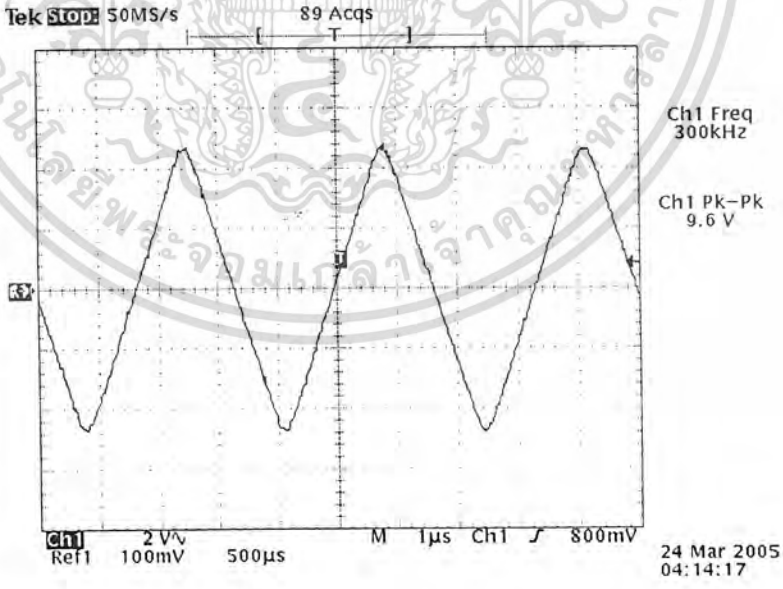


ซ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 100 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ฌ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 200 kHz

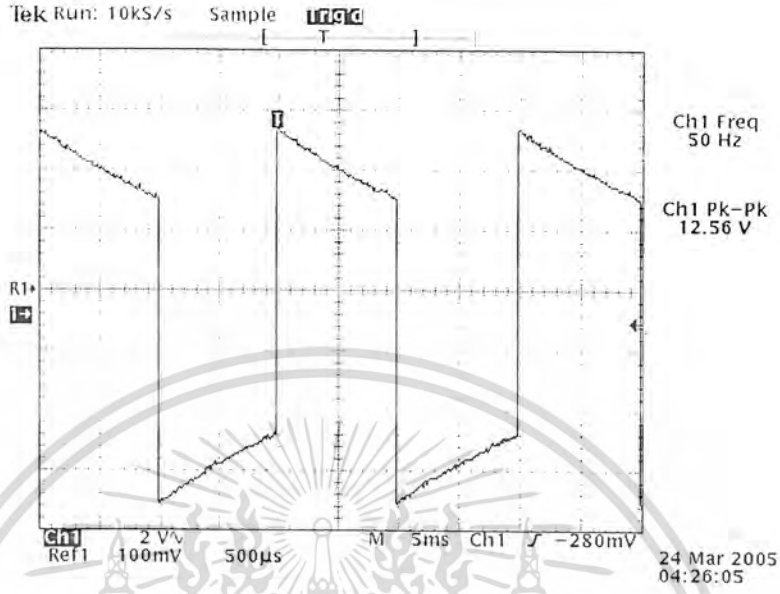


ญ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 300 kHz

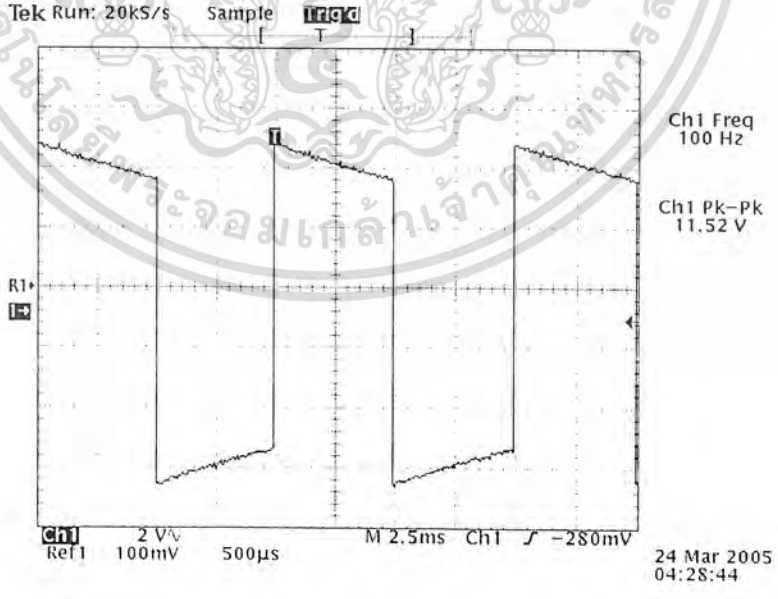
รูปที่ 4.23 แสดงสัญญาณสามเหลี่ยมที่ความถี่ต่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 การทดสอบการกำเนิดสัญญาณสี่เหลี่ยมที่ความถี่ต่างๆดังนี้

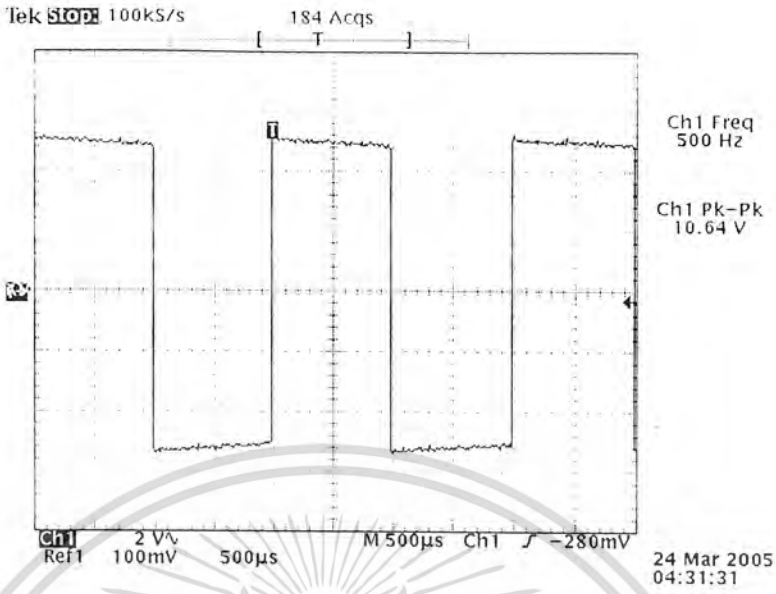


ก) รูปแสดงสัญญาณสี่เหลี่ยมที่ความถี่ 50 Hz

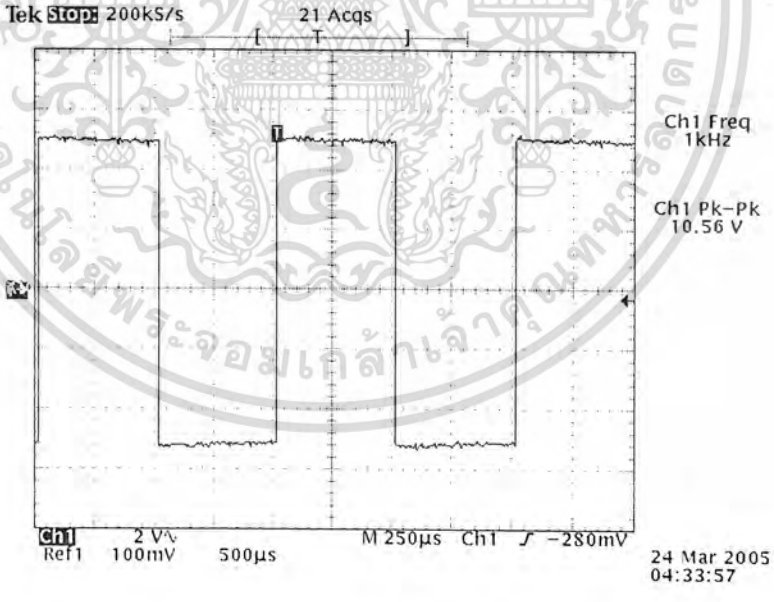


ข) รูปแสดงสัญญาณสี่เหลี่ยมที่ความถี่ 100 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

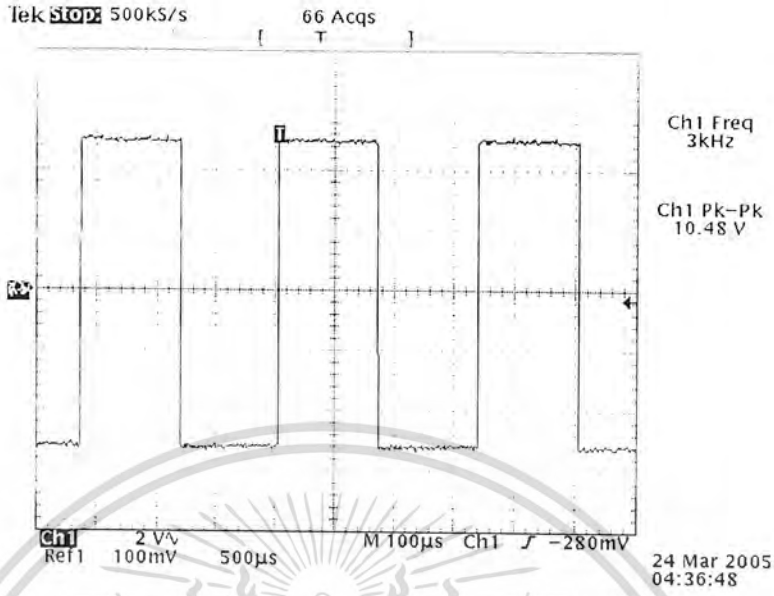


ค) รูปแสดงสัญญาณสี่เหลี่ยมที่ความถี่ 500 Hz

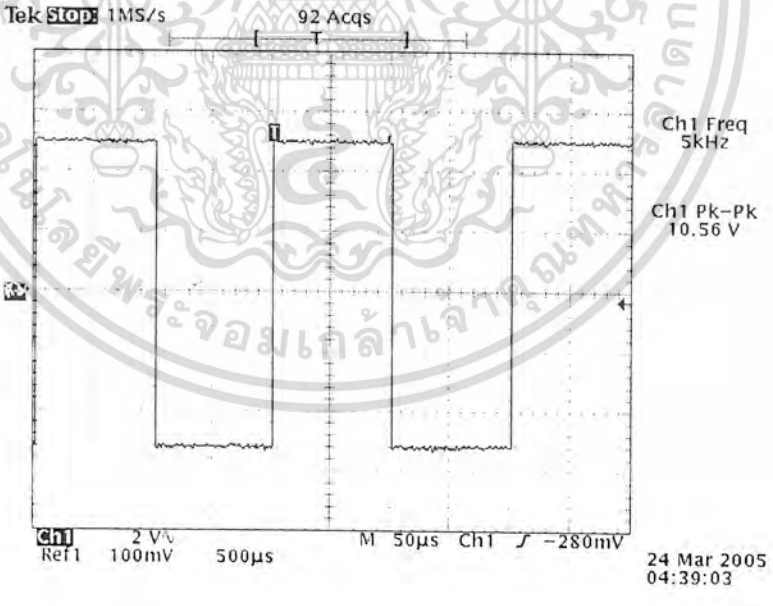


ง) รูปแสดงสัญญาณสี่เหลี่ยมที่ความถี่ 1 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

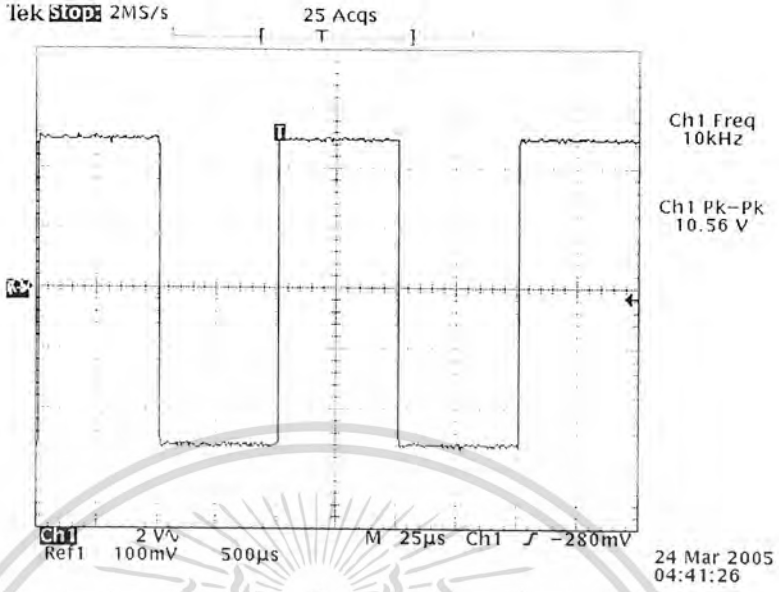


จ) รูปแสดงสัญญาณสี่เหลี่ยมที่ความถี่ 3 kHz

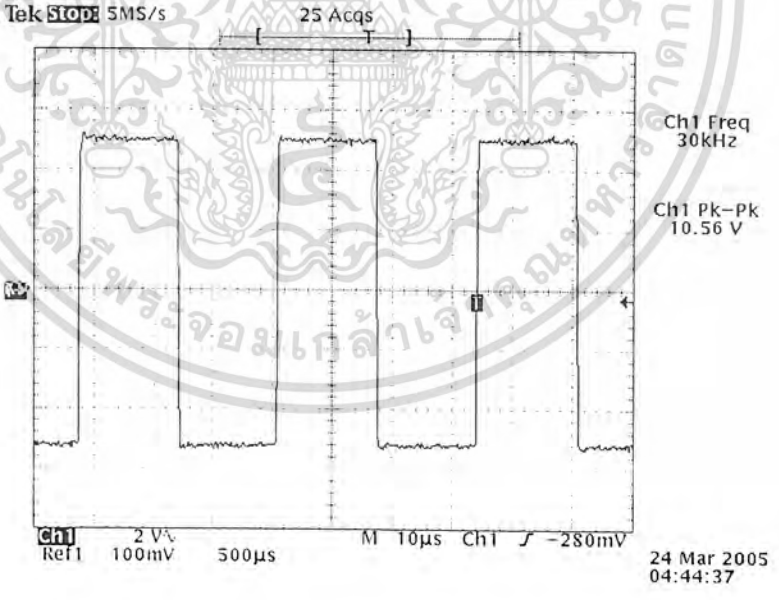


ฉ) รูปแสดงสัญญาณสี่เหลี่ยมที่ความถี่ 5 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

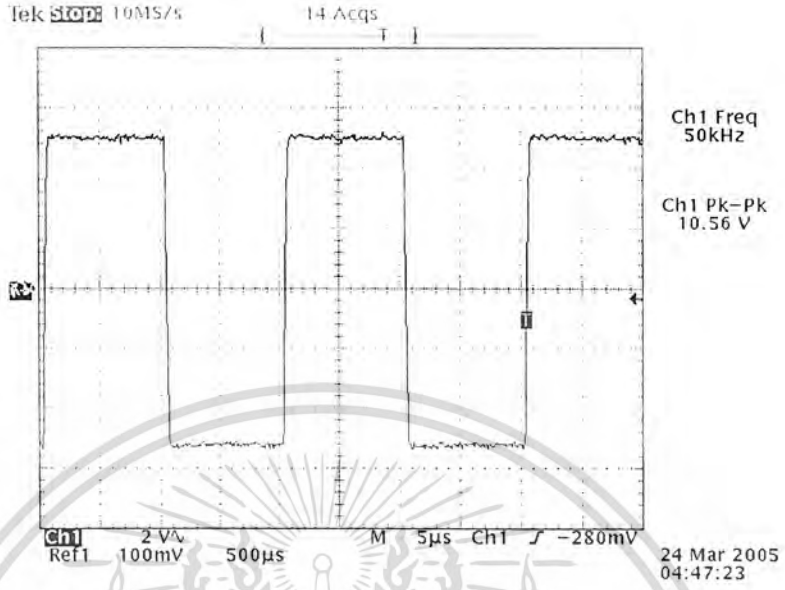


ช) รูปแสดงสัญญาณสี่เหลี่ยมที่ความถี่ 10 kHz

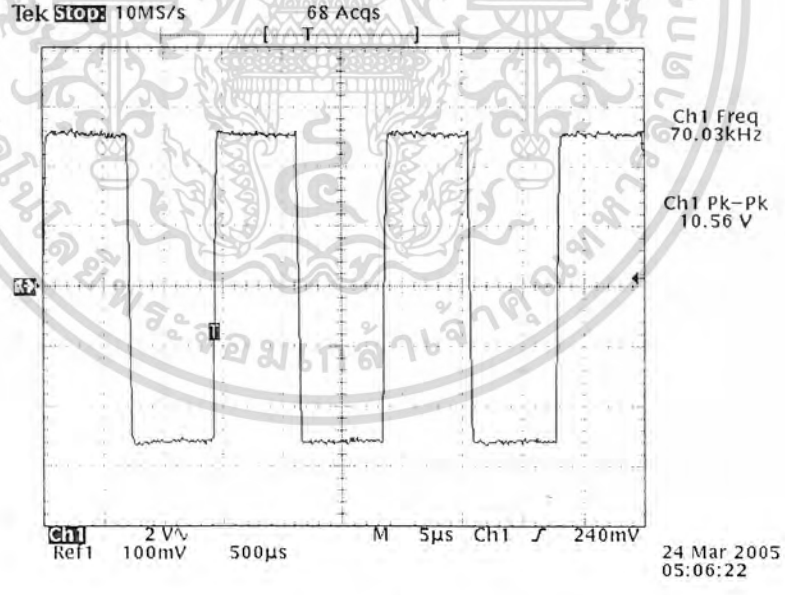


ซ) รูปแสดงสัญญาณสี่เหลี่ยมที่ความถี่ 30 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

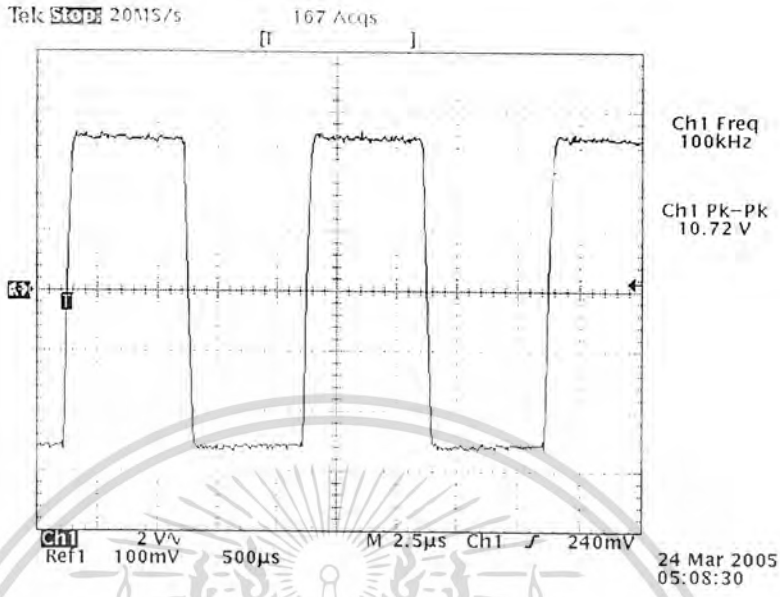


ฌ) รูปแสดงสัญญาณสี่เหลี่ยมที่ความถี่ 50 kHz



ญ) รูปแสดงสัญญาณสี่เหลี่ยมที่ความถี่ 70 kHz

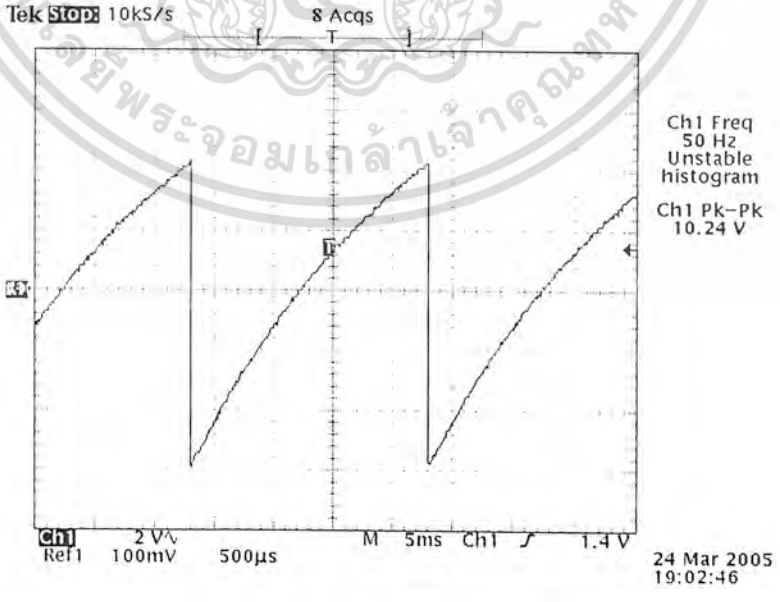
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ฎ) รูปแสดงสัญญาณสี่เหลี่ยมที่ความถี่ 100 kHz

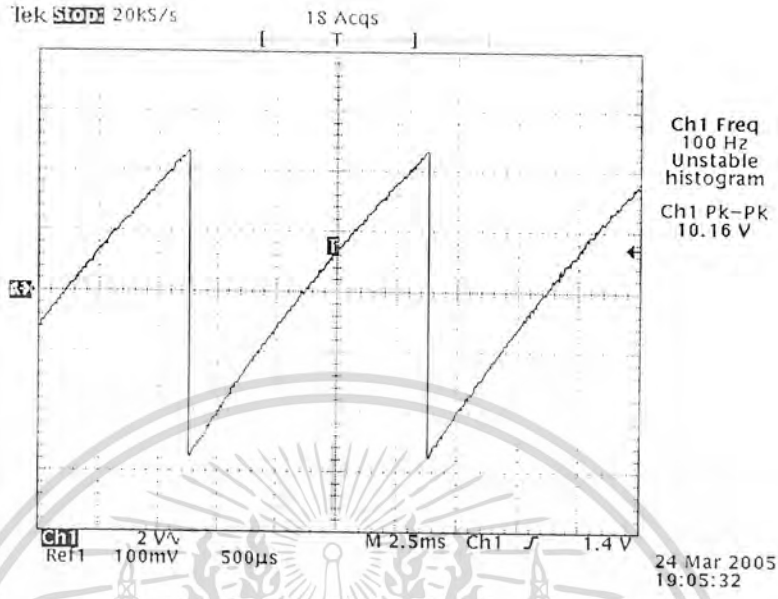
รูปที่ 4.24 แสดงสัญญาณสี่เหลี่ยมที่ความถี่ต่าง

4.3.4 การทดสอบการกำเนิดสัญญาณฟันเลื่อยที่ความถี่ต่างๆดังนี้

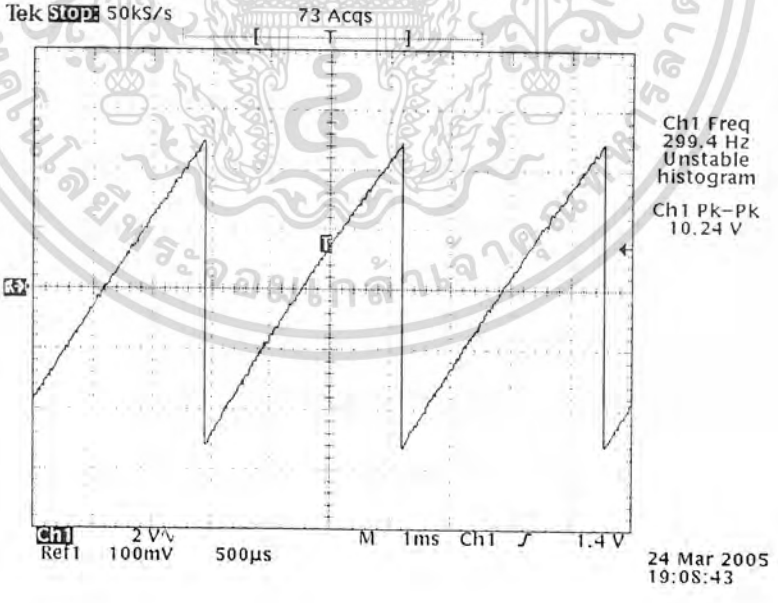


ก) รูปแสดงสัญญาณฟันเลื่อยที่ความถี่ 50 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

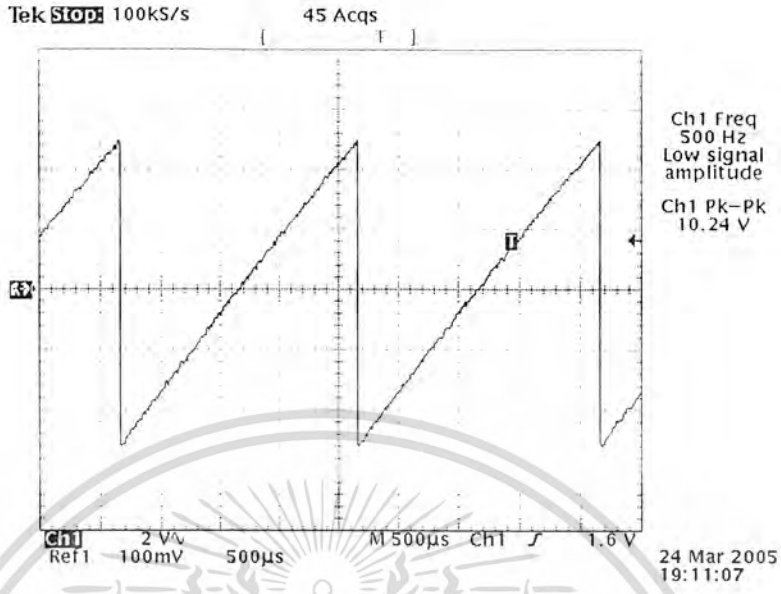


ข) รูปแสดงสัญญาณฟันเลื่อยที่ความถี่ 100 Hz

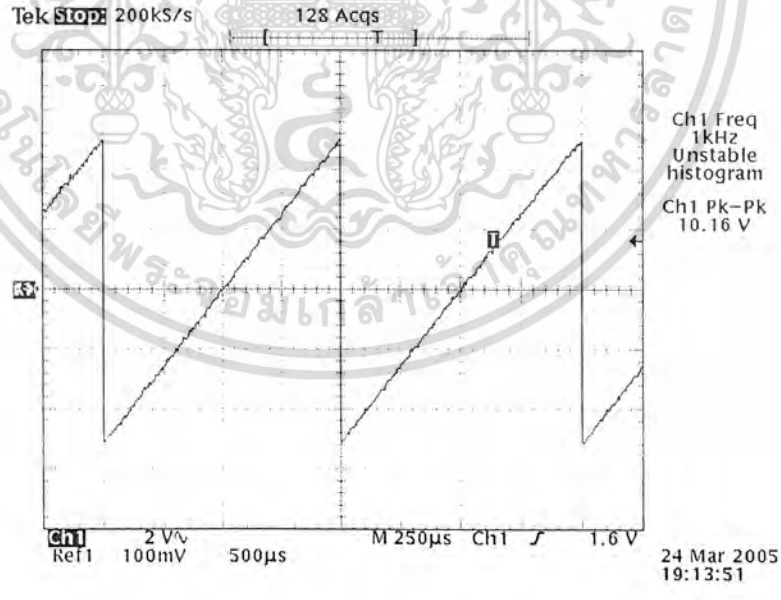


ค) รูปแสดงสัญญาณฟันเลื่อยที่ความถี่ 300 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

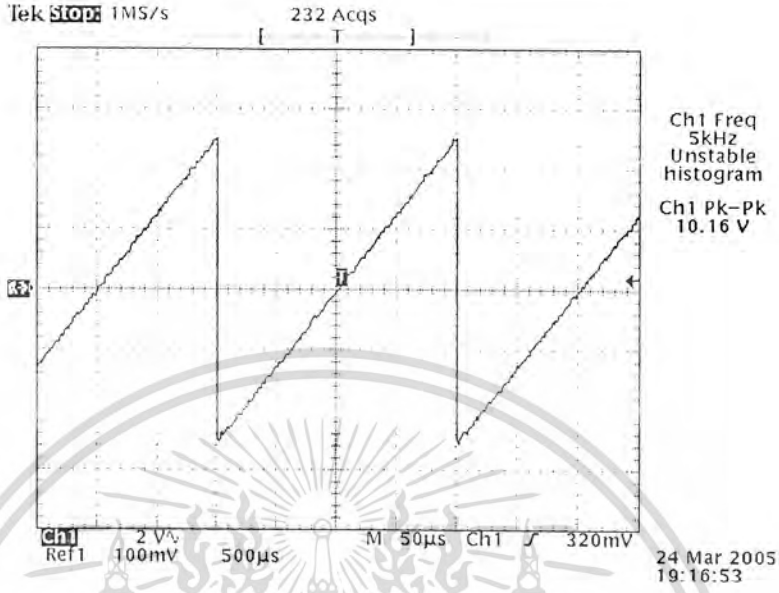


ง) รูปแสดงสัญญาณฟันเลื่อยที่ความถี่ 500 Hz

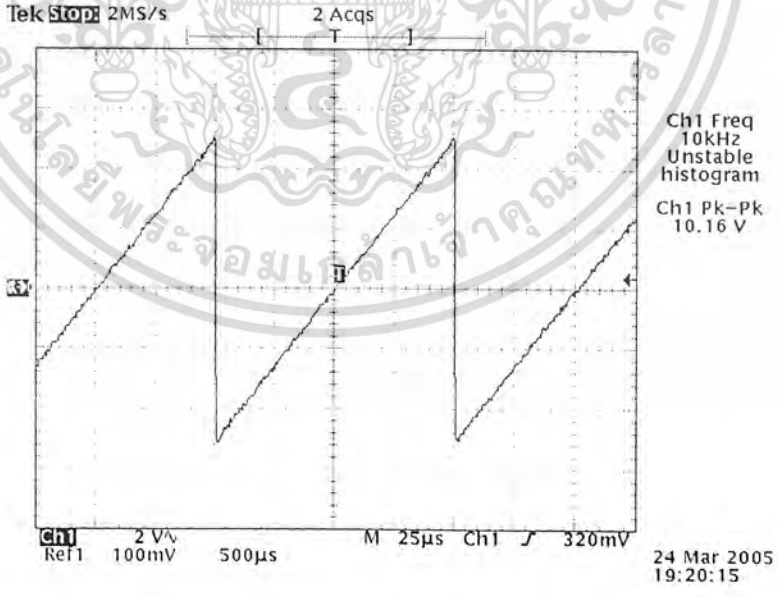


จ) รูปแสดงสัญญาณฟันเลื่อยที่ความถี่ 1 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

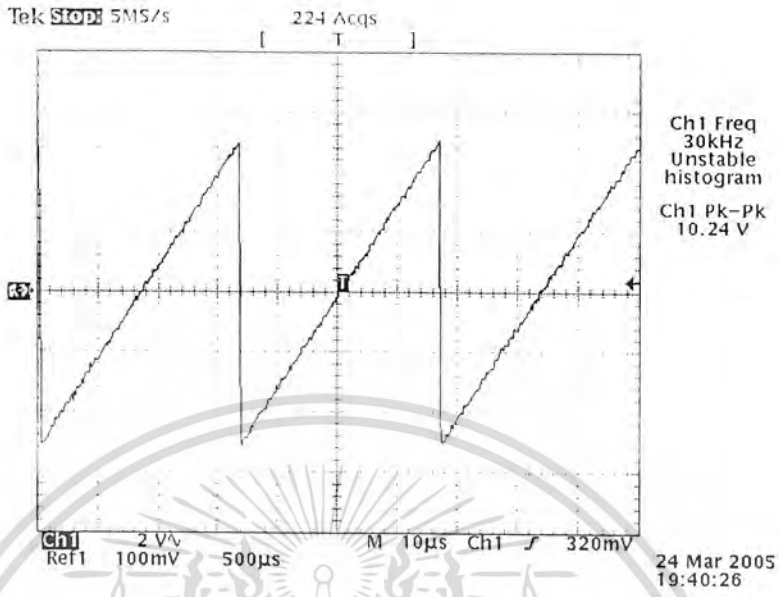


ฉ) รูปแสดงสัญญาณฟันเลื่อยที่ความถี่ 5 kHz

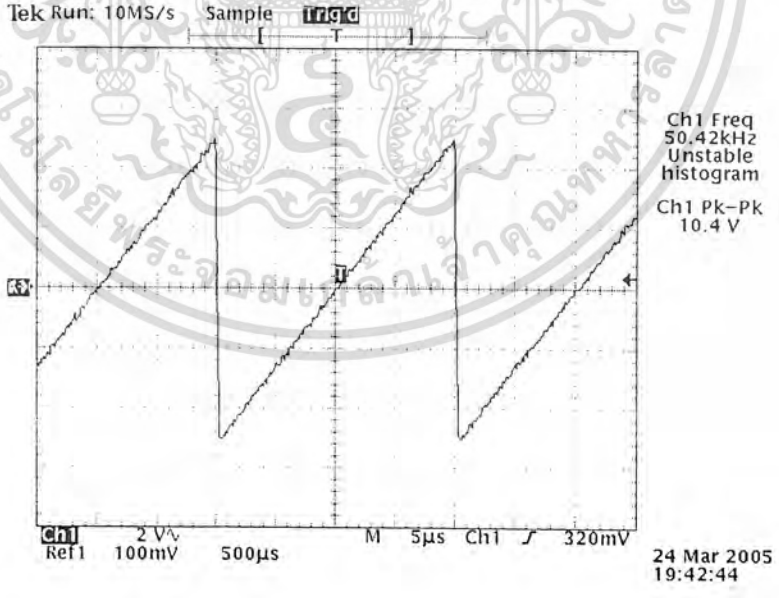


ช) รูปแสดงสัญญาณฟันเลื่อยที่ความถี่ 10 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

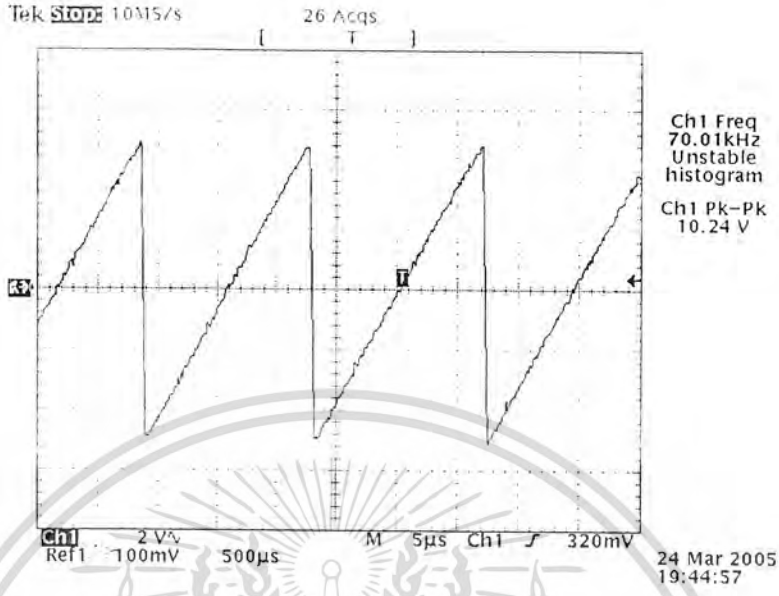


ช) รูปแสดงสัญญาณฟันเลื่อยที่ความถี่ 30 kHz

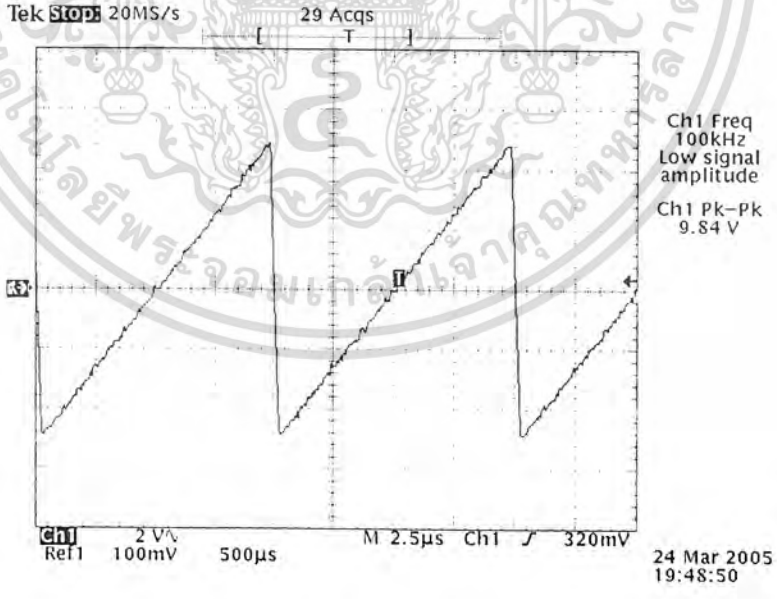


ฌ) รูปแสดงสัญญาณฟันเลื่อยที่ความถี่ 50 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ญ) รูปแสดงสัญญาณฟันเลื่อยที่ความถี่ 70 kHz



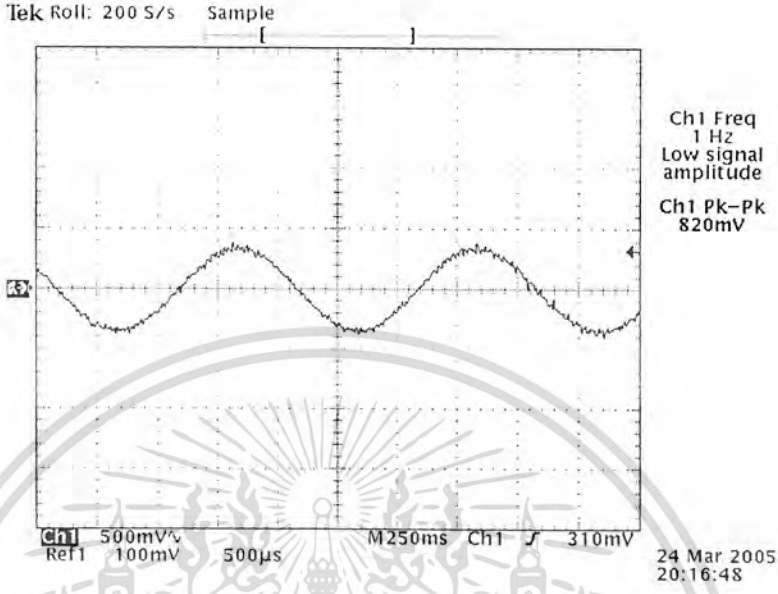
ฎ) รูปแสดงสัญญาณฟันเลื่อยที่ความถี่ 100 kHz

รูปที่ 4.25 แสดงสัญญาณฟันเลื่อยที่ความถี่ต่าง

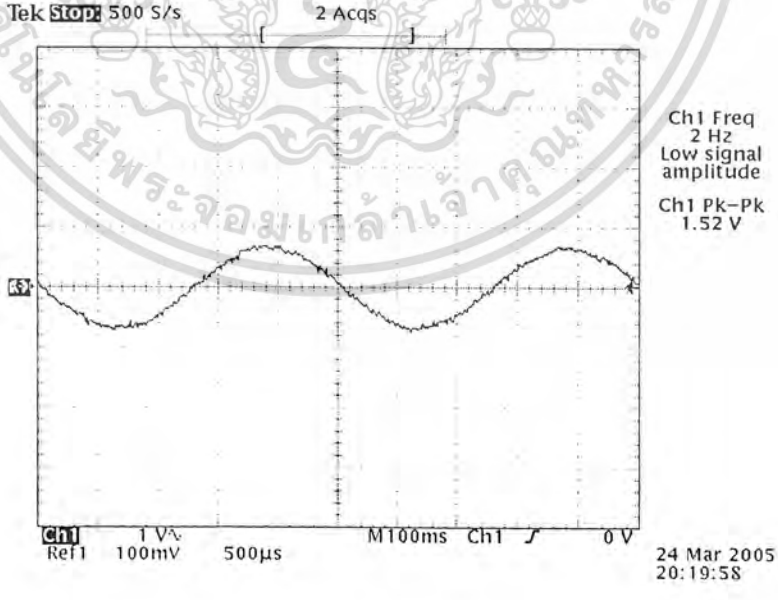
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการทดสอบการทำงานของ SUNDERLAND

4.4.1 การทดสอบการกำเนิดสัญญาณซายน์ที่ความถี่ต่างๆดังนี้

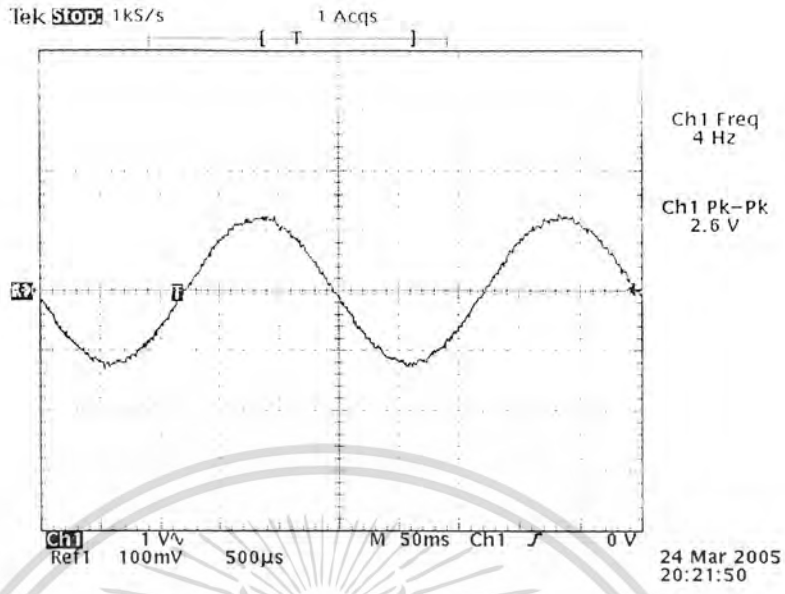


ก) รูปแสดงสัญญาณซายน์ที่ความถี่ 1 Hz

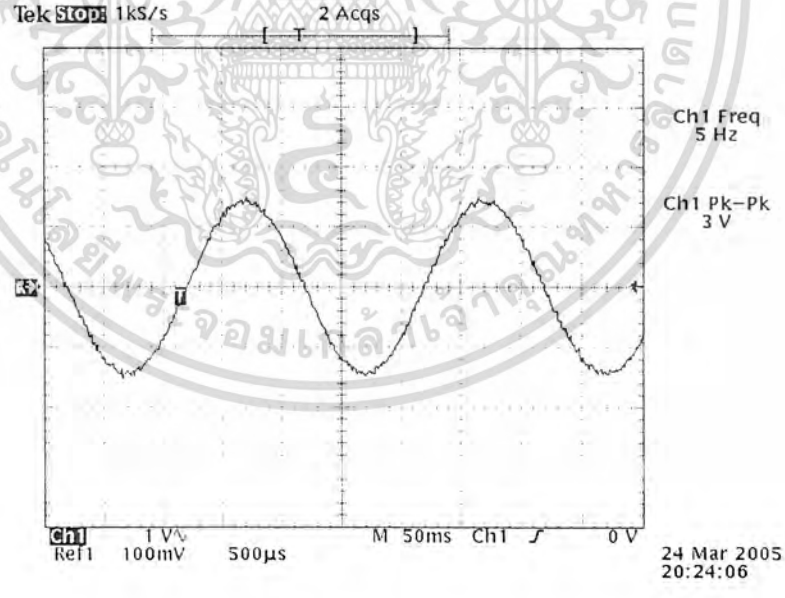


ข) รูปแสดงสัญญาณซายน์ที่ความถี่ 2 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

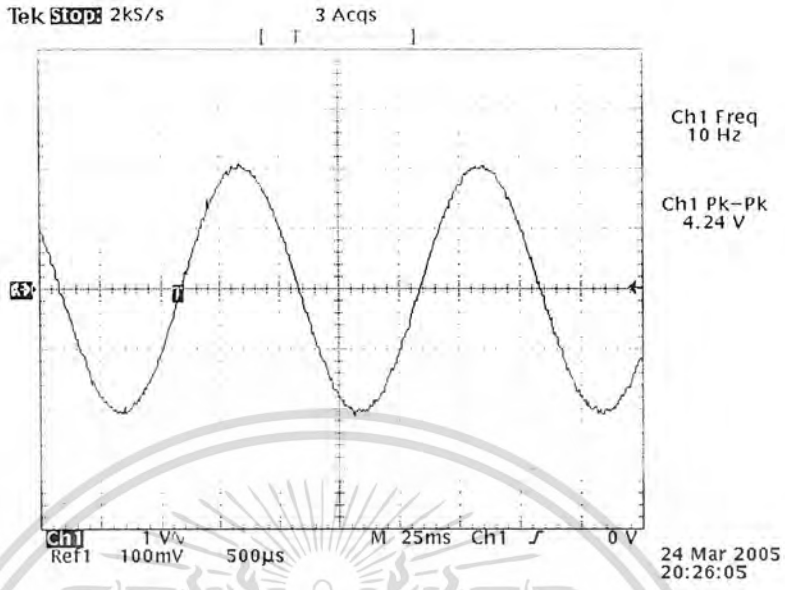


ค) รูปแสดงสัญญาณซายน์ที่ความถี่ 4 Hz

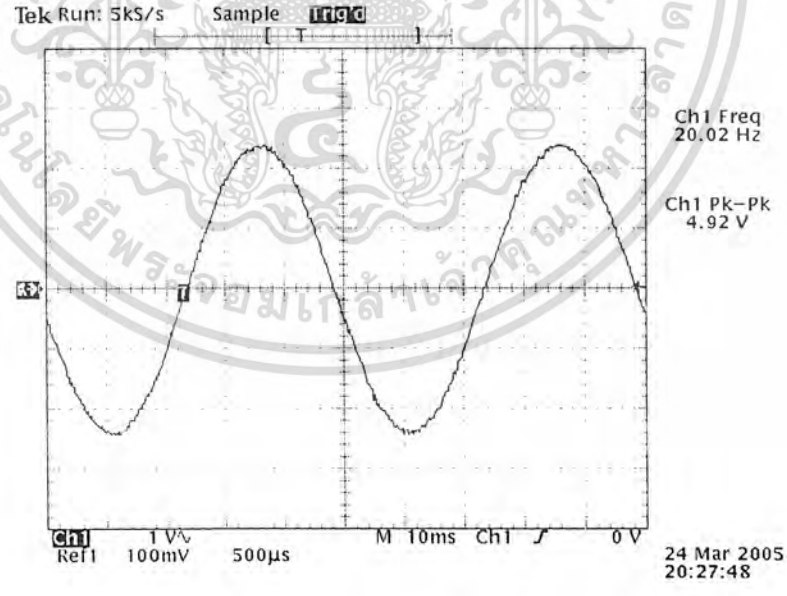


ง) รูปแสดงสัญญาณซายน์ที่ความถี่ 5 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

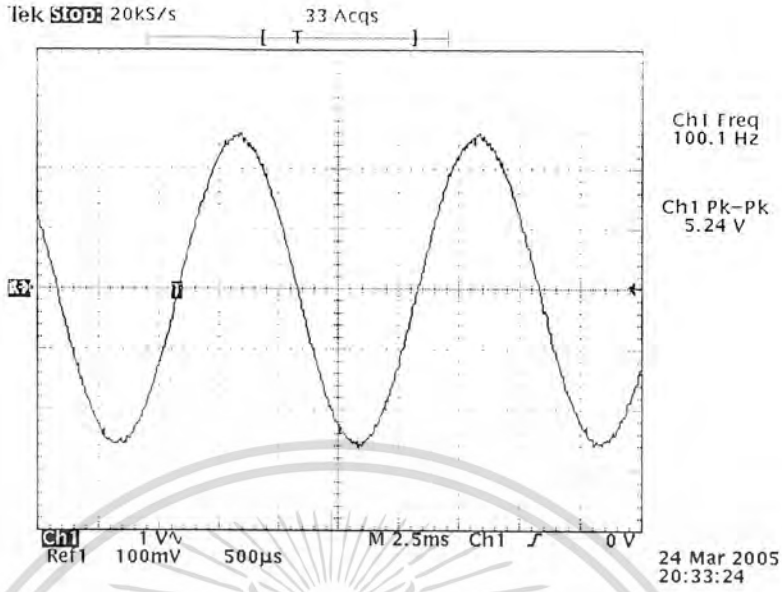


จ) รูปแสดงสัญญาณซายน์ที่ความถี่ 10 Hz

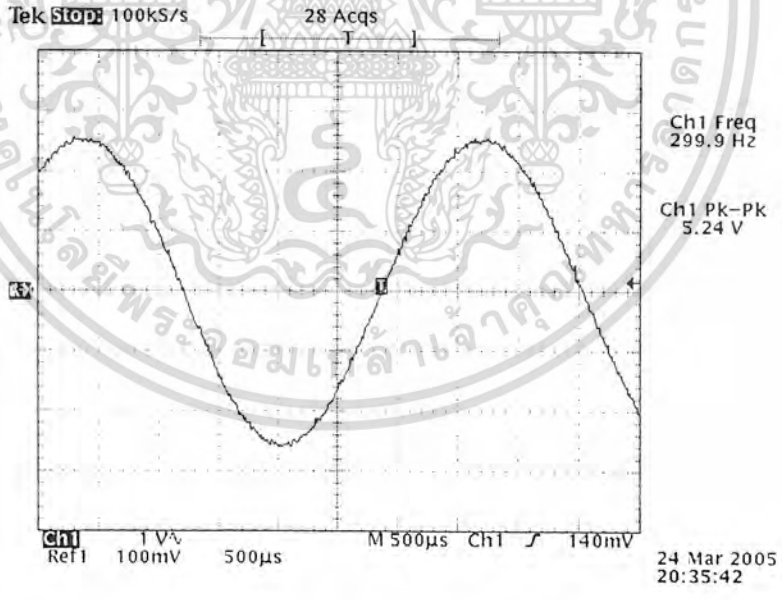


ฉ) รูปแสดงสัญญาณซายน์ที่ความถี่ 20 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

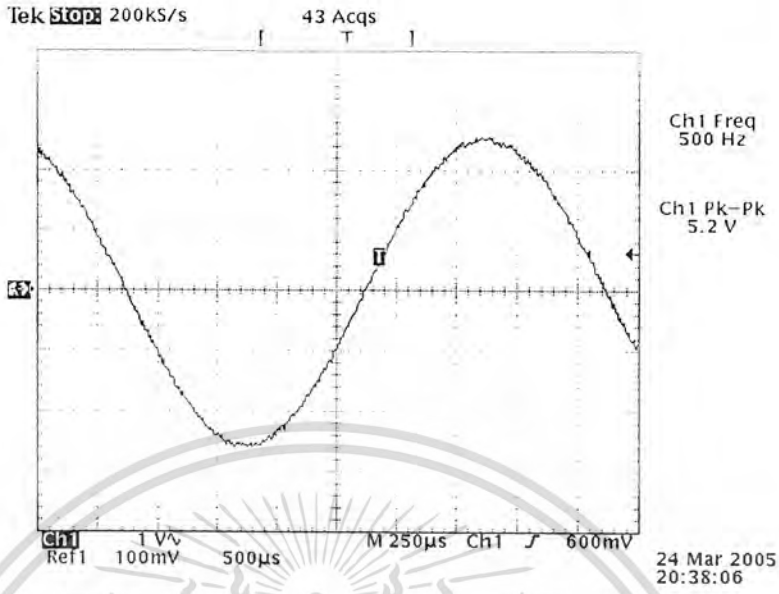


ช) รูปแสดงสัญญาณซายน์ที่ความถี่ 100 Hz

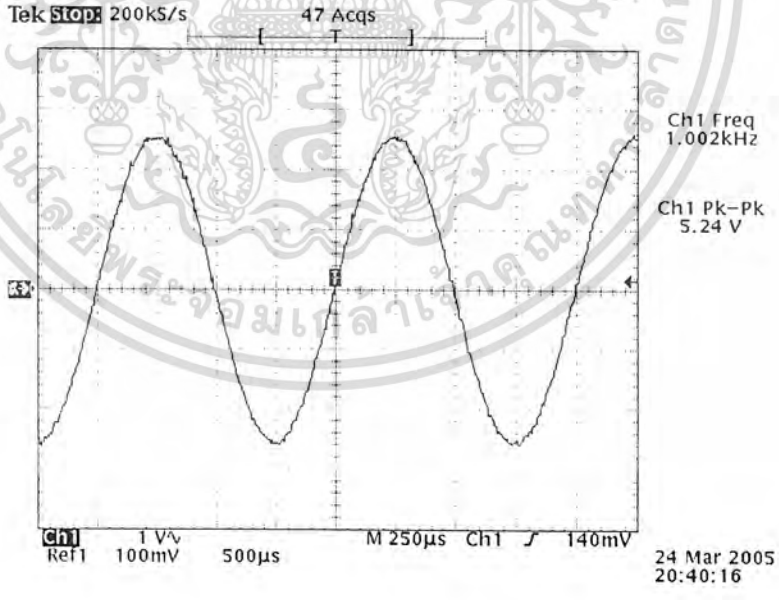


ช) รูปแสดงสัญญาณซายน์ที่ความถี่ 300 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

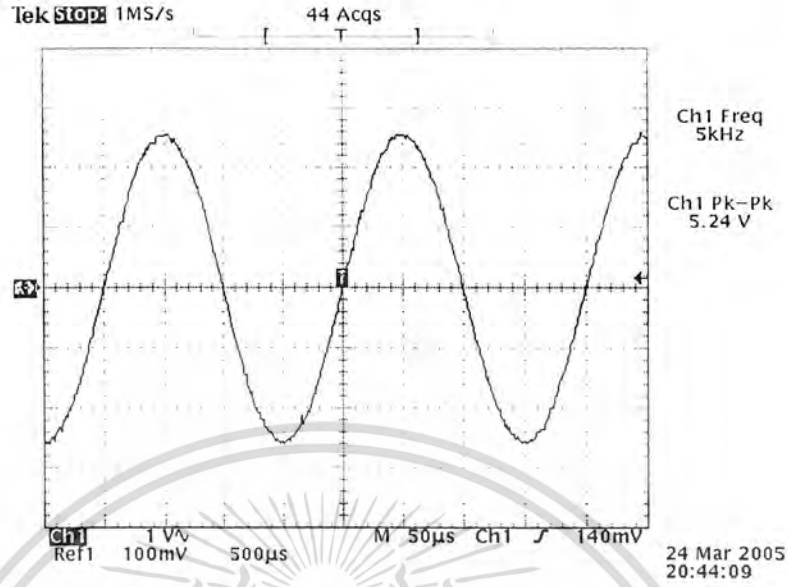


ฌ) รูปแสดงสัญญาณซายน์ที่ความถี่ 500 Hz

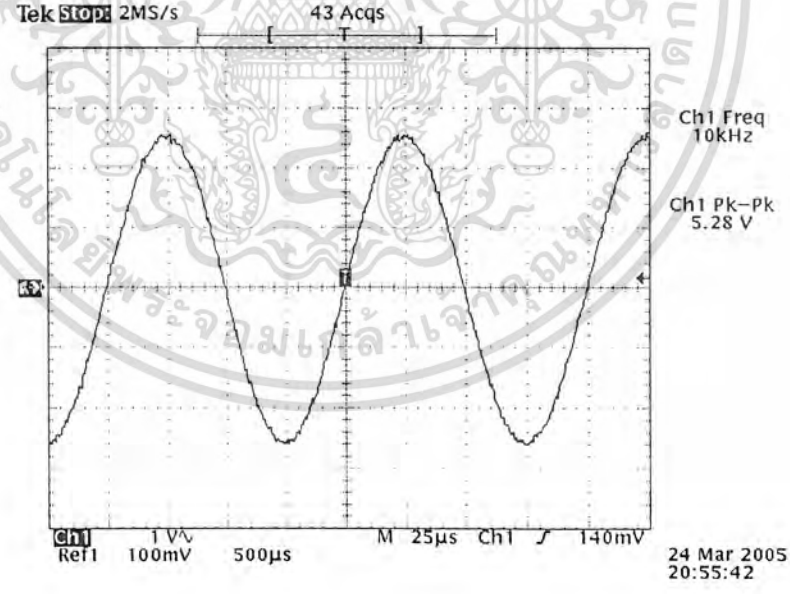


ญ) รูปแสดงสัญญาณซายน์ที่ความถี่ 1 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

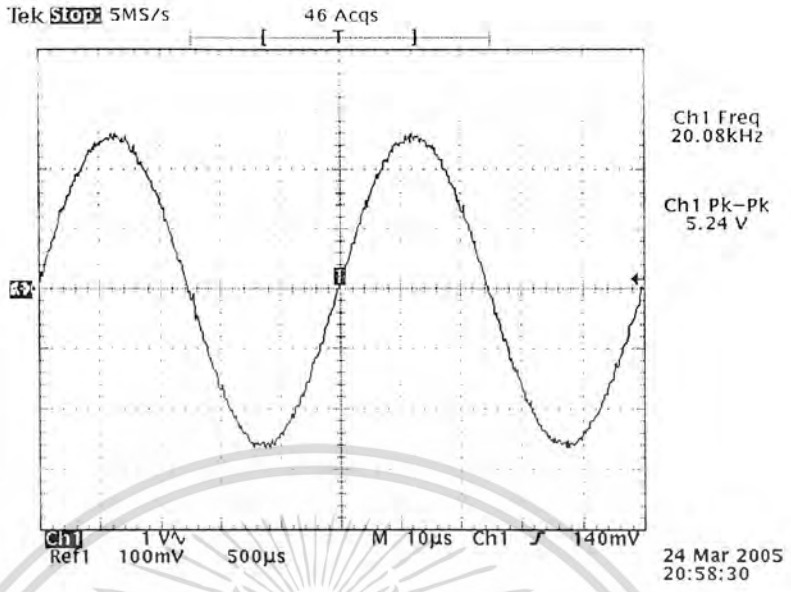


ฎ) รูปแสดงสัญญาณซายน์ที่ความถี่ 5 kHz

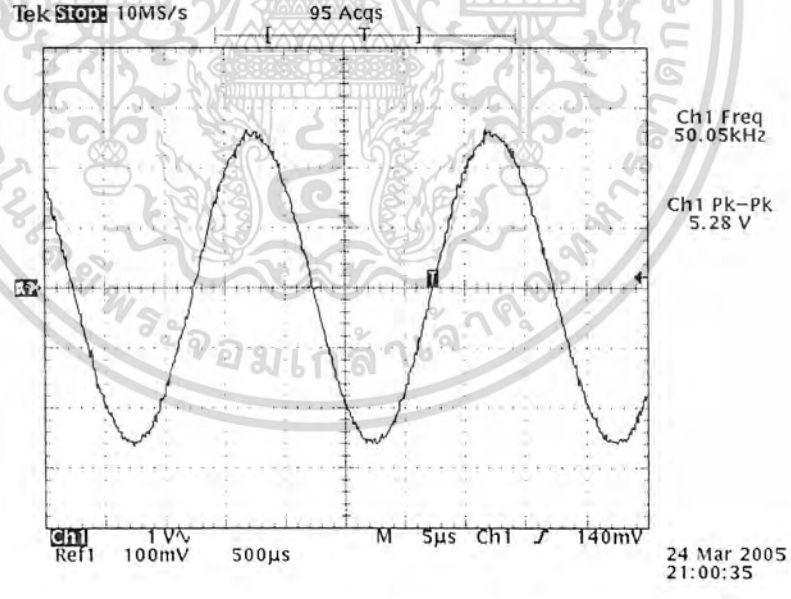


ฐ) รูปแสดงสัญญาณซายน์ที่ความถี่ 10 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

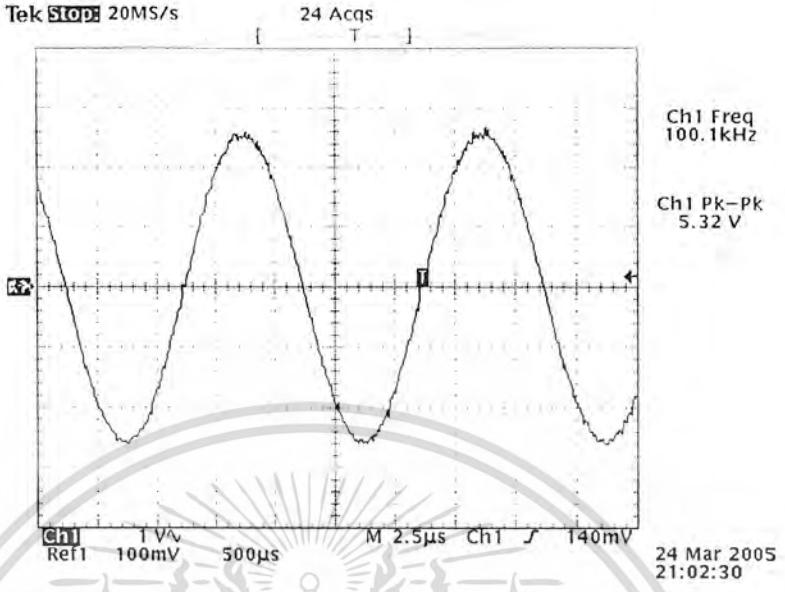


๓) รูปแสดงสัญญาณซายน์ที่ความถี่ 20 kHz

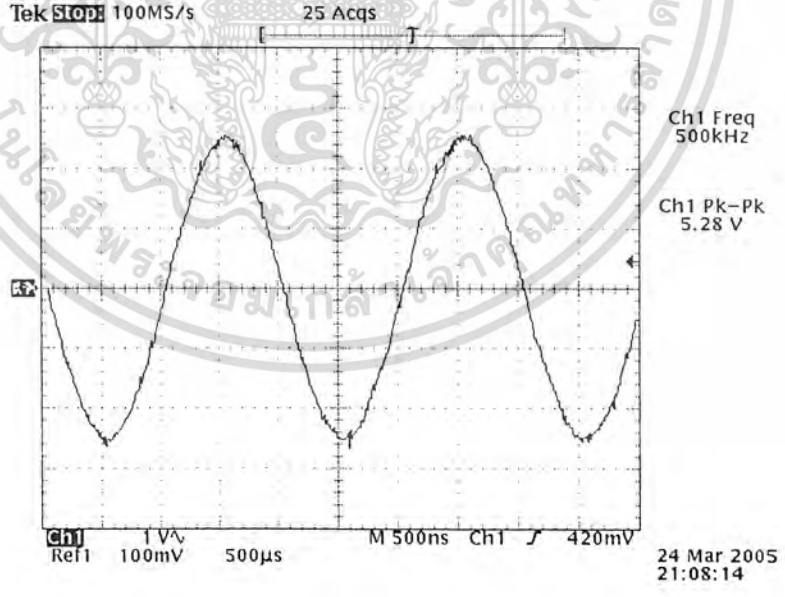


๗) รูปแสดงสัญญาณซายน์ที่ความถี่ 50 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

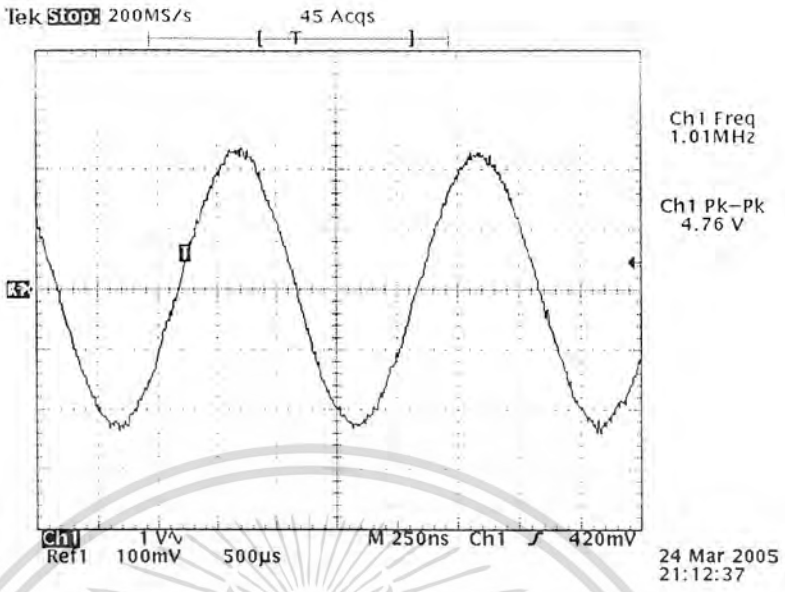


ฌ) รูปแสดงสัญญาณซายน์ที่ความถี่ 100 kHz



ค) รูปแสดงสัญญาณซายน์ที่ความถี่ 500 kHz

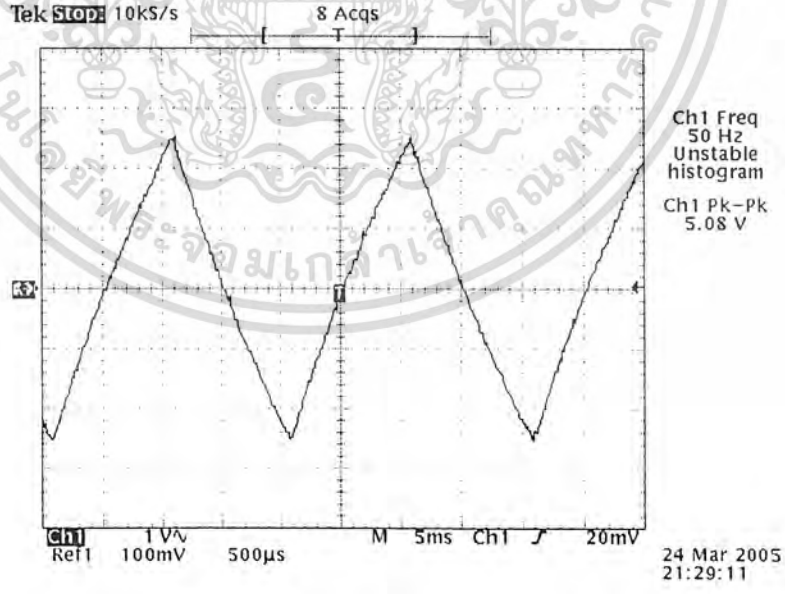
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ต) รูปแสดงสัญญาณซายน์ที่ความถี่ 1 MHz

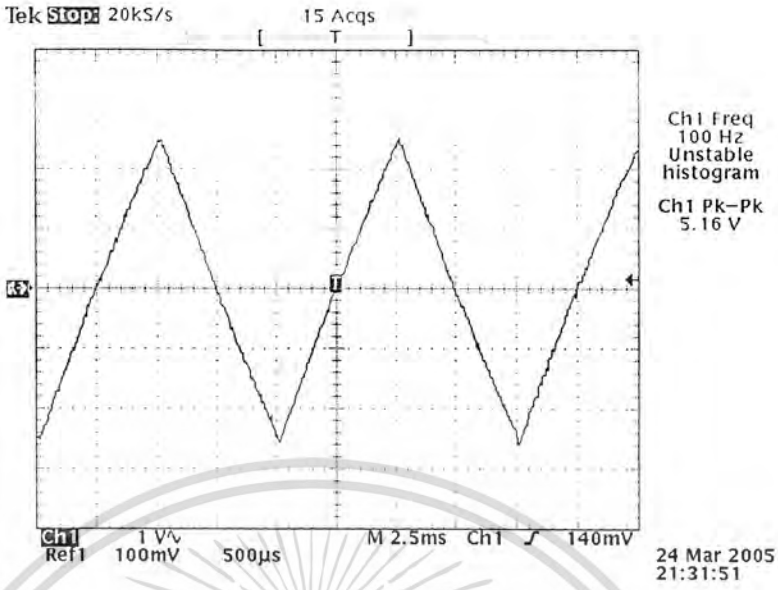
รูปที่ 4.26 แสดงสัญญาณซายน์ที่สร้างจาก Sunderland ที่ความถี่ต่างๆ

4.4.2 การทดสอบการกำเนิดสัญญาณสามเหลี่ยมที่ความถี่ต่างๆดังนี้

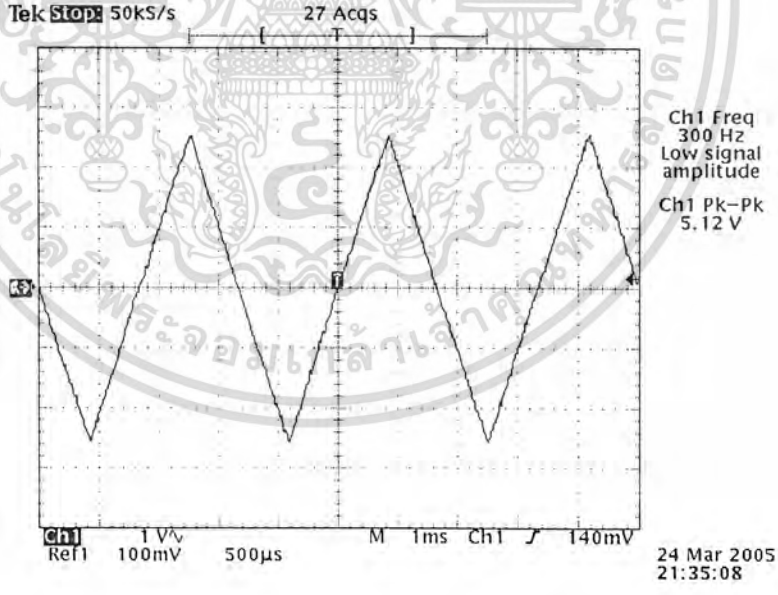


ก) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 50 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

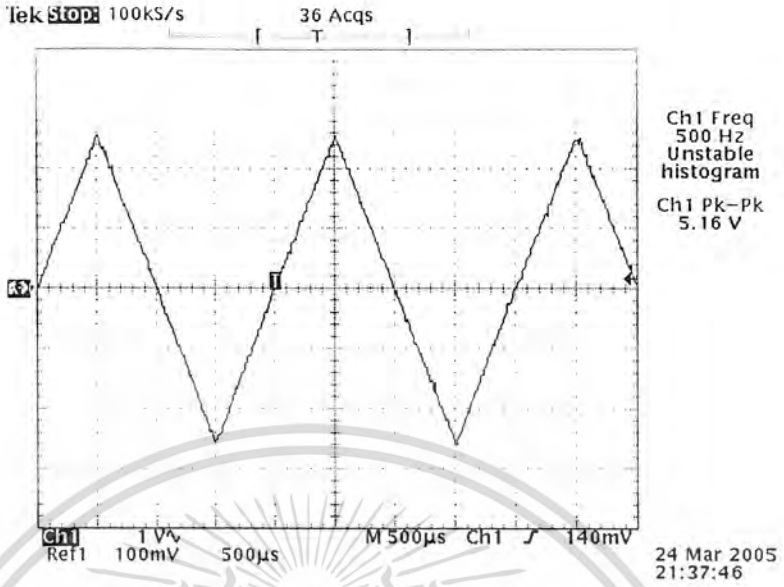


ข) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 100 Hz

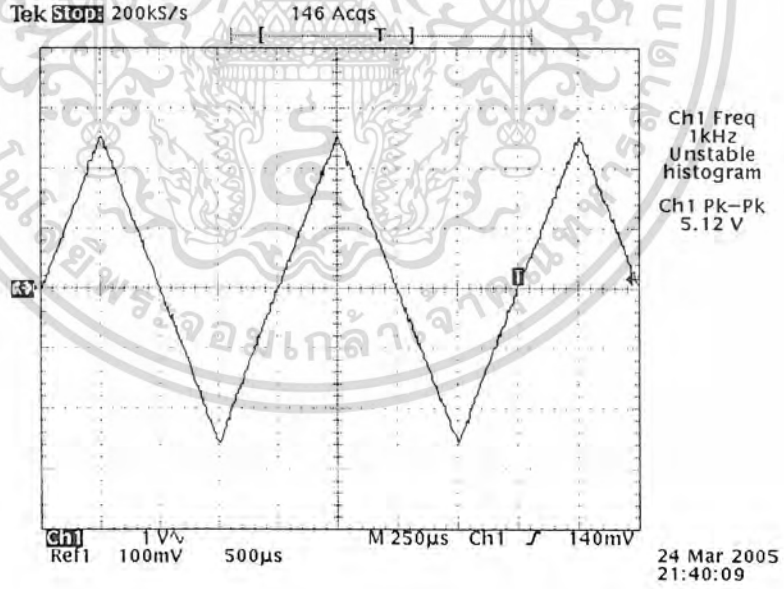


ค) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 300 Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

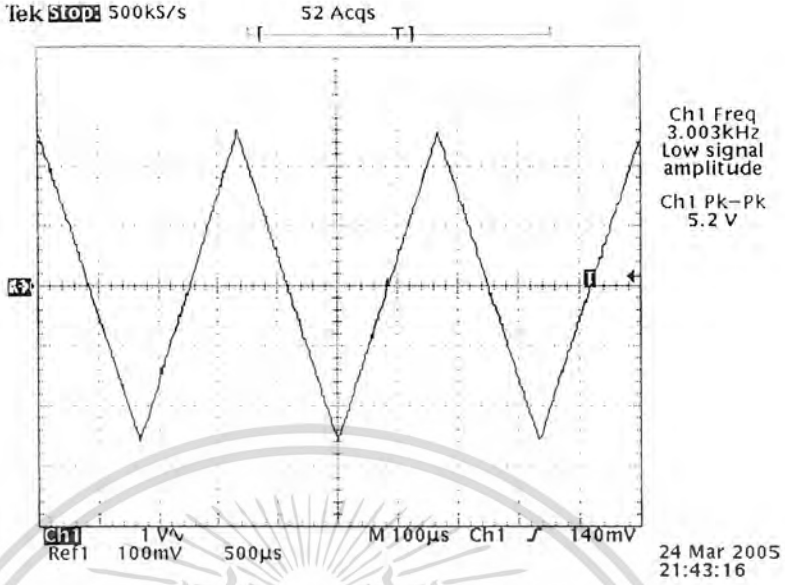


ง) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 500 Hz

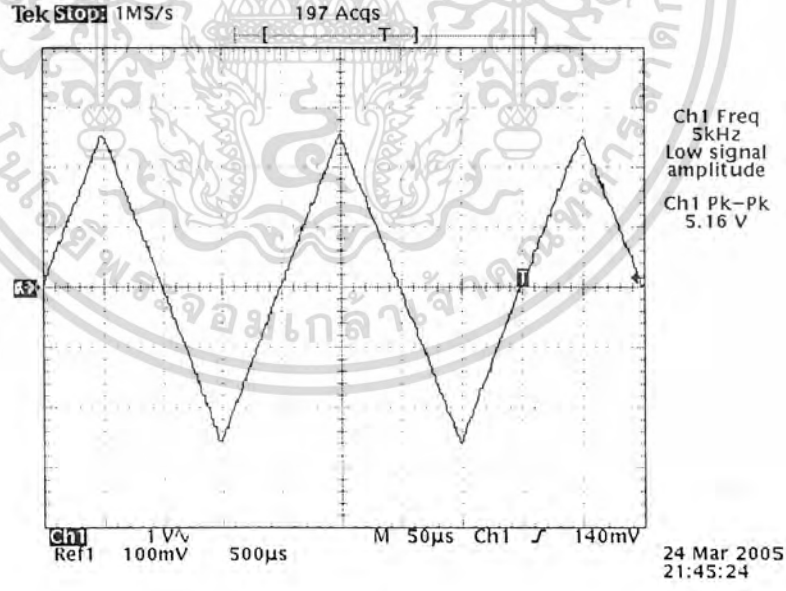


จ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 1 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

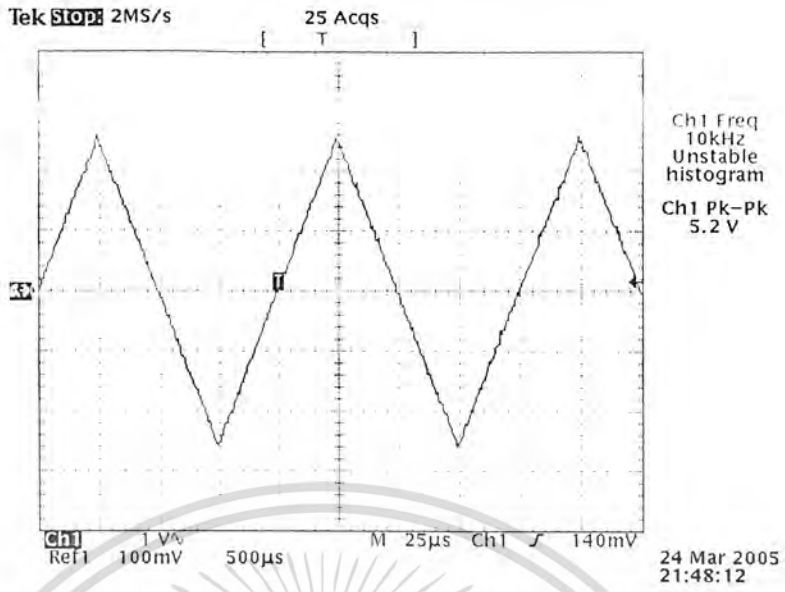


ฉ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 3 kHz

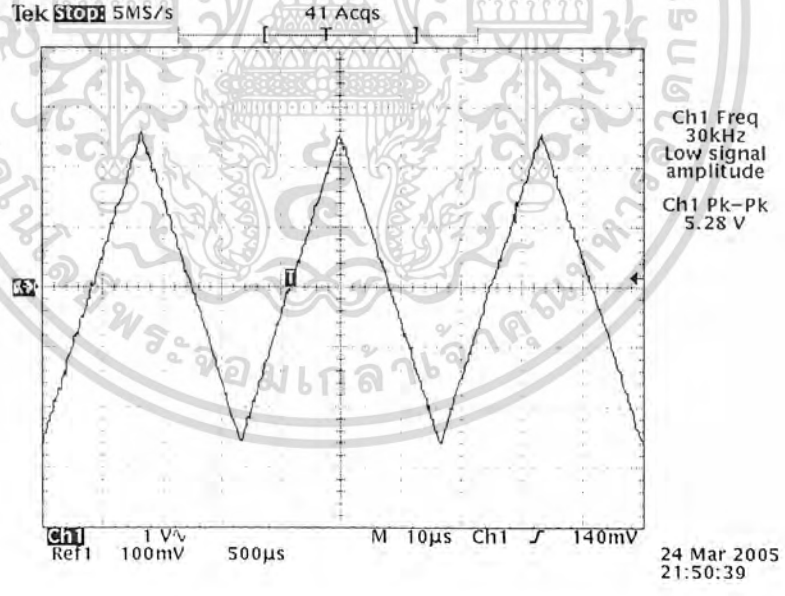


ช) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 5 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

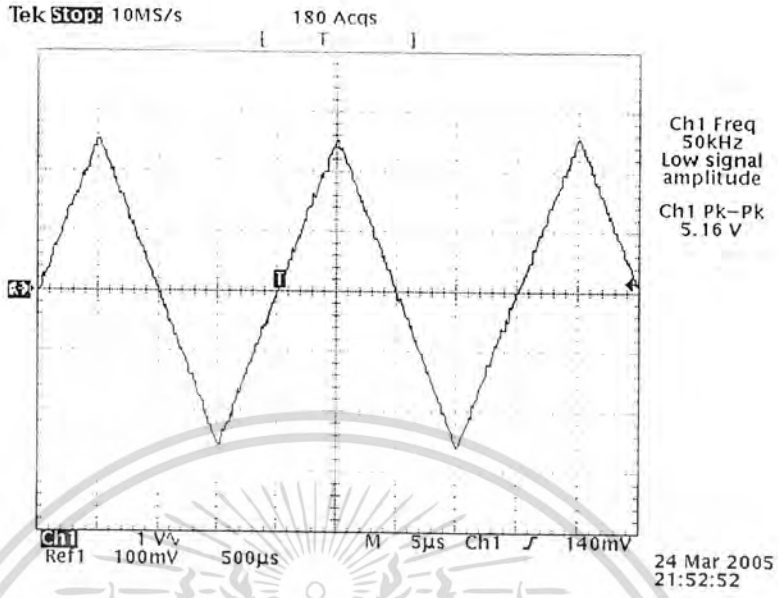


ฉ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 10 kHz

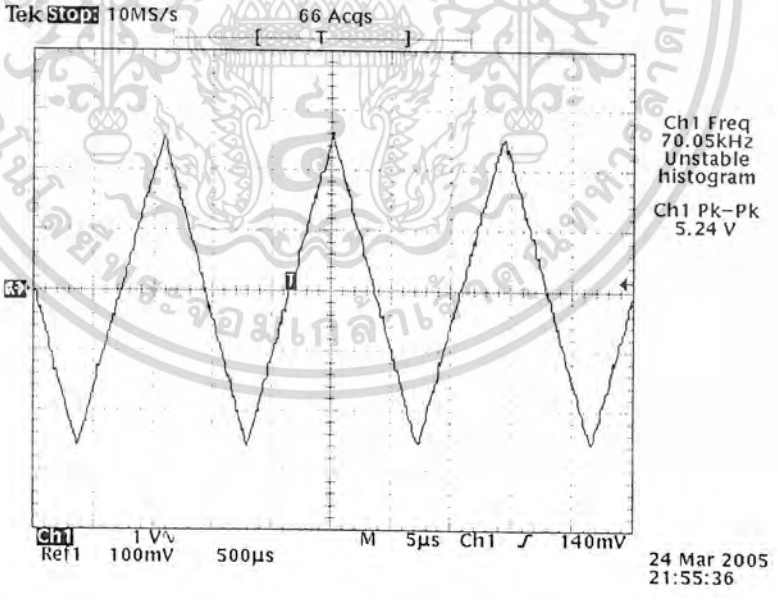


ช) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 30 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

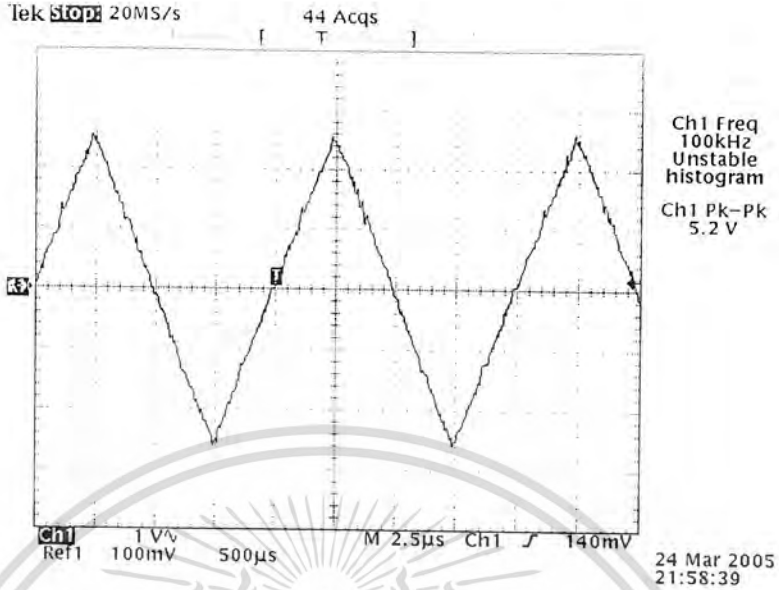


ช) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 50 kHz

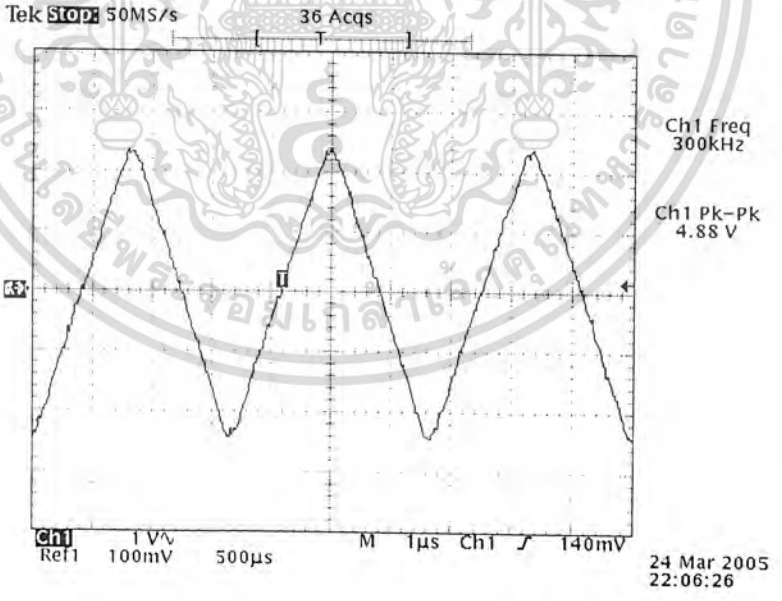


ฌ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 70 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

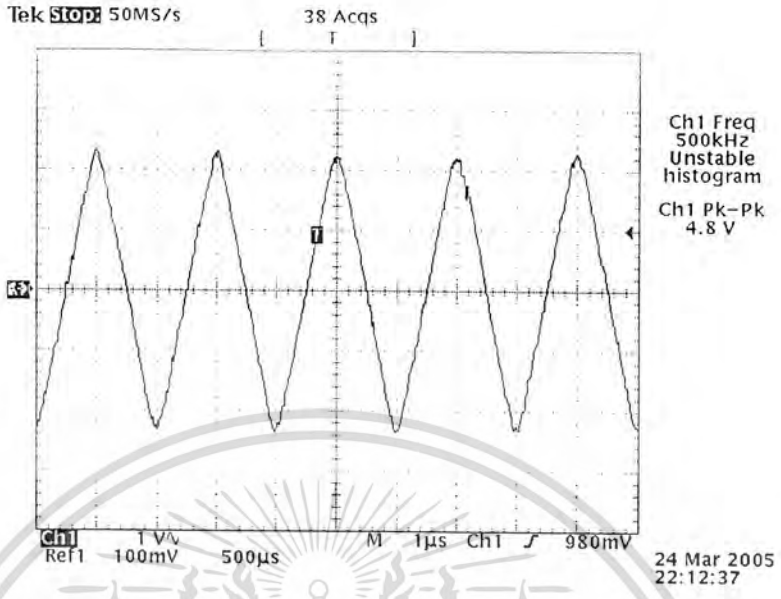


ญ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 100 kHz

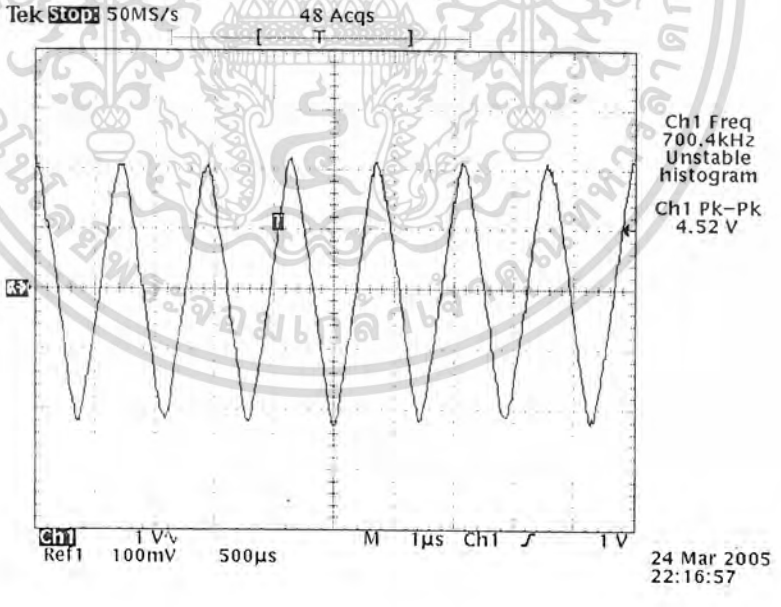


ฎ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 300 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

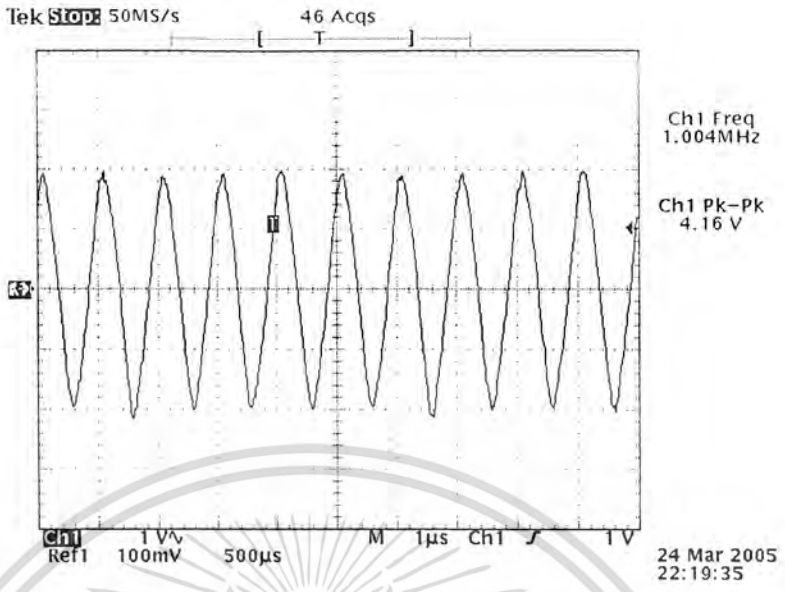


ฎ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 500 kHz



ฐ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 700 kHz

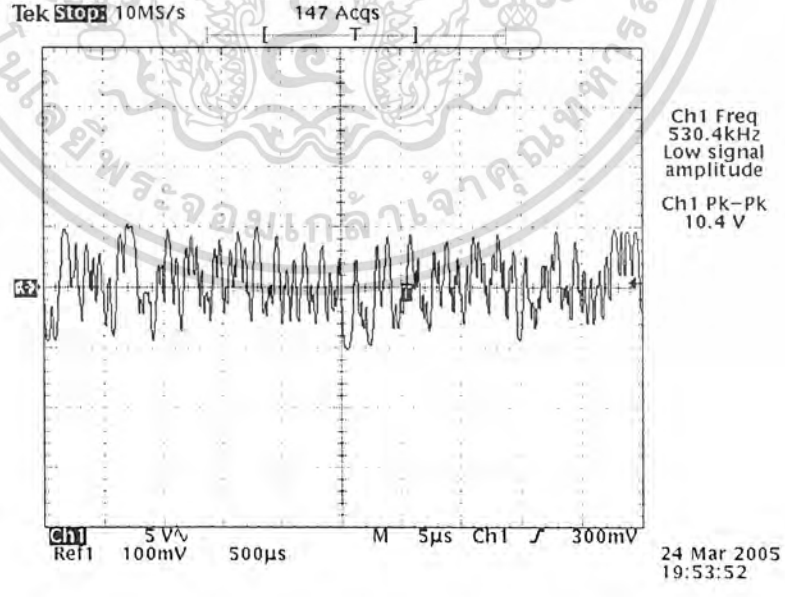
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ฑ) รูปแสดงสัญญาณสามเหลี่ยมที่ความถี่ 1 MHz

รูปที่ 4.27 แสดงสัญญาณสามเหลี่ยมที่สร้างจาก Sunderland ที่ความถี่ต่างๆ

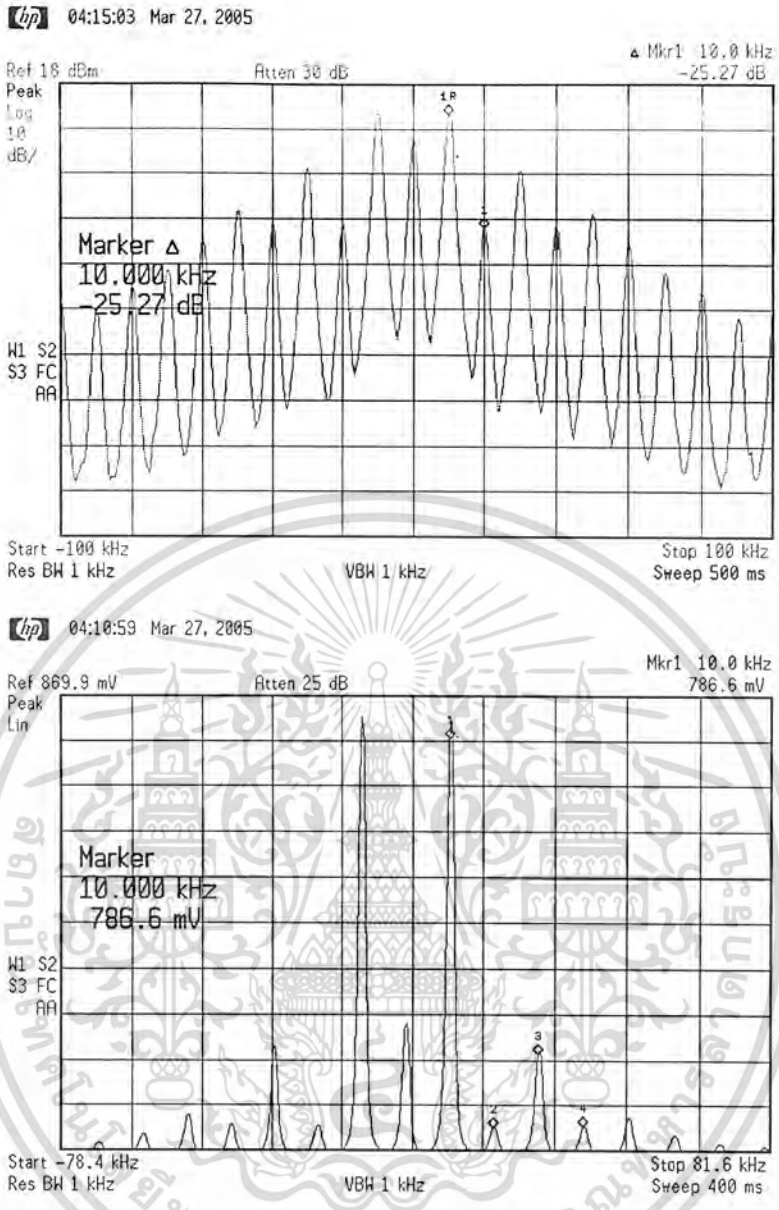
4.5 การทดสอบการกำเนิดสัญญาณสุ่มที่ความถี่ต่างๆดังนี้



ก) รูปแสดงสัญญาณแบบสุ่ม

รูปที่ 4.28 แสดงรูปสัญญาณสุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.29 สเปกตรัมองค์ประกอบของสัญญาณซายน์ 10 กิโลเฮิรตซ์

จากผลการทดลองในการกำเนิดสัญญาณ โดยใช้วิธี Direct Digital Frequency Synthesizer :DDS จะใช้หลักการสร้างสองวิธี วิธีแบบแรกวิธีใช้ตารางเปิดดูได้สัญญาณรูปซายน์,สี่เหลี่ยม,สามเหลี่ยม, ฟันเลื่อย และสัญญาณสุ่ม ส่วนวิธีของซันเดอร์แลนด์จะได้สัญญาณซายน์และสามเหลี่ยม ซึ่งวิธีนี้จะสามารถผลิตความถี่ได้สูงกว่าแบบแรกมาก

ความถี่ของสัญญาณนาฬิกาที่ป้อนให้กับเอฟพีจีเอที่สังเคราะห์ความถี่คือ 9.6 เมกะเฮิรตซ์ และเอฟพีจีเอสังเคราะห์ความถี่สามารถกำเนิดสัญญาณซายน์วิธีใช้ตารางเปิดดู ได้ความถี่ตั้งแต่ 0 ถึง 100 กิโลเฮิรตซ์ และวิธีของซันเดอร์แลนด์ ได้ความถี่ตั้งแต่ 0 ถึง 1 เมกะเฮิรตซ์ โดยวิธีของซันเดอร์แลนด์ มีค่าแอสเฟดีอาร์ (SFDR) ประมาณ –25 เดซิเบลที่ความถี่ 10 กิโลเฮิรตซ์ และมีค่า ทีเอชดี (THD) ประมาณ 26 เปอร์เซ็นต์ หรือประมาณ –11 เดซิเบล

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและวิจารณ์

ทฤษฎีการออกแบบวงจรสังเคราะห์ความถี่ดิจิตอลโดยตรง (Direct Digital Frequency synthesizer : DDS) นั้นจะอาศัยการเก็บค่าขนาดของสัญญาณไว้ในตารางเปิดดู(Lookup table) แล้วใช้การชี้ตำแหน่งในการกำเนิดสัญญาณออกมา ซึ่งในการออกแบบค่าที่เก็บในตารางเปิดดูนั้นจะเก็บค่าขนาดของสัญญาณเพียง 1 ใน 4 คาบท่านั้น ของสัญญาณ 1 คาบเท่านั้น แล้วอาศัยการจัดความสัมพันธ์ของอินพุทเพื่อให้ได้เอาท์พุทเป็นสัญญาณหนึ่งคาบ ส่วนในการสร้างวิธีของซันเดอร์แลนด์จะแบ่งเฟสออกเป็น 3 ส่วนเท่าๆกันส่วนละ4บิต ($\phi = a+b+c$) และอาศัยความสัมพันธ์ของฟังก์ชั่น $\sin(\phi) = \sin(a+b+c)$ ทำให้สามารถประมาณค่าของฟังก์ชั่น $\sin(\phi)$ ได้ดังสมการ $\sin(\phi) \approx \sin(a+b) + \cos a \sin b$

ในการออกแบบวงจรสังเคราะห์ความถี่ดิจิตอลโดยตรงจะใช้ Control word 32 บิตโดยพิจารณาพบว่าความถี่ที่สามารถกำเนิดได้ต่ำสุดตามสมการ Fout (min)=Fclk/2^n และความถี่สูงสุดที่สามารถกำเนิดได้ดังสมการ Fout =(Frequency Input x Fclk)/2^n นั่นคือความถี่ที่กำเนิดต่ำสุดได้ 1 เฮิรตซ์ โดยความถี่จะสามารถเพิ่มที่ละ 1 เฮิรตซ์ไปจนถึงความถี่สูงสุดที่กำเนิดได้คือ 4294967296 เฮิรตซ์ แต่ในการทดลองข้างต้นพบว่า วิธีการสร้างแบบตารางเปิดดูความถี่ต่ำสุดที่กำเนิดได้คือ 1 เฮิรตซ์ และความถี่สูงสุดที่สามารถกำเนิดได้คือ 100 กิโลเฮิรตซ์ วิธีการสร้างแบบซันเดอร์แลนด์ความถี่ต่ำสุดที่กำเนิดได้คือ 1 เฮิรตซ์ และความถี่สูงสุดที่สามารถกำเนิดได้คือ 1 เมกะเฮิรตซ์

สำหรับการประยุกต์ใช้งานวงจรกำเนิดสัญญาณแบบดิจิตอลนั้นสามารถนำสัญญาณมาเข้ารหัสเอฟเอสเค (FSK) หรือการมอดูเลทสัญญาณแบบคิวเอเอ็ม (QAM) รวมทั้งระบบงานทั่วๆไปที่ไปที่ต้องการสัญญาณพื้นฐาน อาทิเช่น สัญญาณซายน์,สัญญาณสามเหลี่ยม,สัญญาณสี่เหลี่ยม หรือสัญญาณฟันเลื่อยเป็นต้น และอาจออกแบบให้ได้สัญญาณเฉพาะ

ในการศึกษาโครงงานนี้ ทำให้มีความรู้เกี่ยวกับทฤษฎีต่างๆมาใช้งานร่วมกัน รวมทั้งวงจรออกแบบที่สามารถนำมาสร้างเป็นฮาร์ดแวร์ด้วยภาษาวีเอชดีแอล ที่มีความยืดหยุ่นในการออกแบบสูง การใช้งาน MCS-51 ในการใช้งานร่วมกับวงจรที่ออกแบบ ตลอดจนการใช้งานร่วมกับคอมพิวเตอร์โดยใช้ภาษาวิชวลซี ในการช่วยในการเชื่อมต่อ (Interface) ซึ่งสิ่งเหล่านี้เหมาะสมแก่การศึกษาค้นคว้าและพัฒนาอย่างยิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์นี้สำเร็จลุล่วงได้ดีด้วยความกรุณาของ รศ.ดร.กอบชัย เดชหาญและอาจารย์ศรวัฒน์ ชิวปรีชา อาจารย์ที่ปรึกษา ผู้คอยเอาใจใส่ดูแลและเมตตาแก่ศิษย์ด้วยดีเสมอมา คณะผู้จัดทำรู้สึกซาบซึ้งในความกรุณาเป็นอย่างยิ่งและขอบพระคุณอย่างสูง

ขอบคุณห้องปฏิบัติการสำหรับสถานที่ เครื่องมือ และอุปกรณ์ในการทำวิทยานิพนธ์ฉบับนี้ ขอบคุณพระคุณอาจารย์ทุกๆท่านที่ประสิทธิ์ประสาทวิชาความรู้ต่างๆให้แก่ศิษย์ทั้งทางตรงและทางอ้อมตลอดจนกำลังใจ คำแนะนำเกี่ยวกับปัญหาต่างๆ และความช่วยเหลือทุกอย่างระหว่างทำปริญญานิพนธ์นี้

สุดท้ายนี้ขอกราบขอบคุณพระคุณบิดา มารดา ที่ให้ความสำคัญกับการศึกษาของลูกและให้การสนับสนุนเอาใจใส่ดูแลด้วยดีเสมอมารวมทั้งกำลังใจอันยิ่งใหญ่อย่างหาที่เปรียบมิได้

ธนะสิน ปลุกใจ
วิรัตน์ วัฒนากร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1.D.L.Perry, "VHDL",McGraw-Hill Compony,Inc,volume3,1998.

2.B. Goldberg, "Direct techniques in frequency synthsis", New York:McGraw-Hill,1996.

3.V.F Kroupa, "Direct digital frequency synthesizers",IEEE Press,1999.

4.D.L.Perry, "VHDL",New York:McGraw-Hill,1995.

5.J.G.Proakis and D.G.Manolakis, "Introduction to Digital Signal Processing",Macmillan publisling,1988, p372-376.

6.มนัส สังวรศิลป์,วรรัตน์ ภัทรอมรกุลม, "คู่มือการใช้งาน MATLAB ฉบับสมบูรณ์", สำนักพิมพ์อินโฟเพรส,กรุงเทพ,2543.

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้