การแปลงฟูเรียร์แบบเร็ว ด้วย เอฟพีจีเอ

Fast Fourier Transform by using FPGA

โดย

นายธีรภัทร เชวงทรัพย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

การแปลงฟูเรียร์แบบเร็ว ด้วย เอฟพีจีเอ
Fast Fourier Transform by using FPGA

โดย

นาย ถิรภัทร เชวงทรัพย์

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแปลงฟูเรียร์แบบเร็ว ด้วย เอฟพีจีเอ

Fast Fourier Transform by using FPGA

โดย

นาย ถิรภัทร เชวงทรัพย์ 53011929

อาจารย์ที่ปรึกษา

ดร. สิรภพ ตู้ประกาย

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า การแสวงหากำไร หรือการอื่นใดไม่ให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

29/6/57

KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

7/8/57

KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

โครงงานปีการศึกษา 2556

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การแปลงฟูเรียร์แบบเร็ว ด้วย เอฟพีจีเอ

FAST FOURIER TRANSFORM BY USING FPGA

ผู้จัดทำ

นาย ถิรภัทร เชวงทรัพย์ 53011929

..... อาจารย์ที่ปรึกษา

(ดร. สิรภพ ตู้ประกาย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงได้ด้วยความกรุณาจาก ดร.สิรภพ ตู้ประกาย อาจารย์ที่ปรึกษาโครงงาน ที่ได้ให้คำเสนอแนะ แนวคิด ตลอดจนวิธีแก้และแนวทางแก้ไข ข้อบกพร่องต่างๆ มาโดยตลอด จนรายงานโครงงานเล่มนี้เสร็จสมบูรณ์ ผู้ขอกราบขอพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณ คุณพ่อและคุณแม่ ที่ได้ให้คำปรึกษาต่างๆรวมทั้งเป็นกำลังใจที่ดีเสมอมา

ขอขอบคุณ ทุนส่งเสริมและสนับสนุนห้องปฏิบัติการวิจัยและพัฒนา สำนักงานคณะกรรมการกิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมแห่งชาติที่สนับสนุนเครื่องมือวัด

ขอขอบคุณรุ่นพี่ที่ค่อยให้คำแนะนำ สำหรับการใช้เครื่องมือ อุปกรณ์ รวมถึง ให้ความเข้าใจและคำปรึกษาในการทำงานโครงงานครั้งนี้

และสุดท้ายของขอบคุณเพื่อนๆและน้องๆทุกคนที่ค่อยให้คำแนะตลอดการทำโครงงานชิ้นนี้จนสำเร็จลุล่วงไปได้

นาย ถิรภัทร เชว่งทรัพย์
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแปลงฟูเรียร์แบบเร็ว ด้วย เอฟพีจีเอ

FAST FOURIER TRANSFORM BY USING FPGA

โดย นาย ถิรภัทร เชวงทรัพย์ 53011929

อาจารย์ที่ปรึกษา ดร.สิรภพ ตู้ประกาย

บทคัดย่อ

โครงงานนี้เป็นการนำเทคโนโลยี FPGA มาประยุกต์ใช้งาน โดยสร้างฮาร์ดแวร์ที่สามารถแปลงสัญญาณในโดเมนเวลาให้เป็นสัญญาณในโดเมนความถี่ ซึ่งวิธีที่ใช้ก็คือ วิธีการแปลงฟูเรียร์อย่างเร็ว(FFT)

ABSTRACT

This project presents the using of FPGA technology application. By creating hardware that can convert the signal in time domain to the frequency domain with the fast Fourier transform.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		V
สารบัญตาราง		VII
บทที่ 1	**บทนำ**	**1**
1.1	ความเป็นมาและความสำคัญของปัญหา	1
1.2	วัตถุประสงค์	1
1.3	ขอบเขตของโครงงาน	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**2**
2.1	การแปลงฟูริเยร์แบบเร็ว	2
2.1.1	การคำนวณการแปลงฟูเรียร์เต็มหน่วยโดยตรง	2
2.1.2	การคำนวณการแปลงฟูเรียร์แบบเร็ว	3
2.1.2.1	การคำนวณการแปลงฟูเรียร์แบบเร็วโดยการลดทอนทางเวลา	3
2.1.2.2	แฟกเตอร์ตัวบิดและหน่วยคำนวณผีเสื้อ	5
2.1.2.3	การกลับบิตและภาระการคำนวณ	6
2.1.2.4	การคำนวณการแปลงฟูเรียร์แบบเร็วโดยการลดทอนทางความถี่	8
2.2	โปรแกรม MATLAB	11
2.2.1	ความสามารถของโปรแกรม MATLAB	12
2.2.2	ส่วนประกอบสำคัญของ MATLAB	12
2.3	FPGA (FIELD PROGRAMMABLE GATE ARRAYS)	14
2.4	ภาษา VHDL	15
2.4.1	ภาษา HDL (HARDWARE DESCRIPTION LANGUAGES)	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
	2.4.2 ข้อกำหนดภาษา VHDL	16
	2.4.3 การออกแบบระบบดิจิตอลแบบจากบนลงล่าง	17
บทที่ 3	**การออกแบบและการจัดทำโครงงาน**	**19**
	3.1 การออกแบบ	19
	3.1.1 การทดลองการจำลองสร้าง FREQUENCY SWEEP	19
	3.1.2 การทดลองออกแบบการสร้าง FIR FILTER	21
	3.1.3 การทดลองออกแบบการสร้าง FIR FILTER แบบ 32 TAB	22
	3.1.4 การทดลองออกแบบการสร้าง FFT ผ่านโปรแกรม MATLAB	22
	3.2 SIMULATE THE DESIGN	26
	3.2.1 การ CONFIG FFT MODEL	26
บทที่ 4	**ผลการทดลอง**	**34**
	4.1 ผลการสร้างการจำลองการสร้าง FREQUENCY SWEEP ผ่านโปรแกรม MATLAB	34
	4.2 ผลการทดลองออกแบบการสร้าง FIR FILTER ผ่านโปรแกรม MATLAB	39
	4.3 ผลการทดลองออกแบบการสร้าง FIR FILTER แบบ 32 TAP ผ่านโปรแกรม MATLAB	41
	4.4 ผลการทดลองออกแบบการสร้าง FFT ผ่านโปรแกรม MATLAB	43
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**47**
	5.1 สรุปผล	47
	5.2 ข้อเสนอแนะ	47
หนังสืออ้างอิง		48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่

2.1 การคำนวณการแปลงฟูเรียร์แบบเร็วโดยการลดทอนทางเวลา 4
2.2 กราฟไหลสัญญาณของการแปลง 2 จุด 5
2.3 กราฟการไหลของสัญญาณในหน่วยคำนวณผีเสื้อ 5
2.4 หน่วยคำนวณผีเสื้อที่ลดรูปแล้ว 6
2.5 การแปลง 8 จุดโดยใช้หน่วยคำนวณผีเสื้อที่ลดรูปแล้ว (DIT) 6
2.6 หน่วยคำนวณผีเสื้อของการแปลงฟูเรียร์โดยการลดทอนทางความถี่LCD 9
2.7 กราฟไหลสัญญาณของการคำนวณโดยการลดทอนทางความถี่ขนาด 8 จุด ครั้ง ภาคแรก 10
2.8 กราฟไหลสัญญาณแปลงฟูเรียร์เต็มหน่วยเร็วโดยการลดทอนทางความถี่ 11
2.9 แสดงหน้าตาของโปรแกรม MATLAB 11
2.10 แสดงโครงสร้างของส่วนการทำงาน SIMULINK ของโปรแกรม 13
2.11 พัฒนาการของภาษา VHDL 15
2.12 แสดงขอบเขตของการจำลองการทำงานและการสังเคราะห์เป็นวงจรของภาษา VHDL 17
2.13 ขั้นตอนการออกแบบระบบดิจิตอลแบบจากบนลงล่าง 18
3.1 แสดงส่วนประกอบภายในของ MATLAB 19
3.2 โมเดลการทำ FREQUENCY SWEEP 19
3.3 บล็อกการทำงานย่อยภายในส่วนของ FREQSWEEP_HIL/HIL BLOCK 20
3.4 การเชื่อมต่อ FIR FILTER 21
3.5 การเชื่อมต่อ FIR FILTER แบบ 32 TAB 22
3.6 การเชื่อมต่อภายใน 32 TAB FIR 23
3.7 6-TAPS FIR FILTER 23
3.8 การเชื่อมต่อ SHIFT TAPS BLOCK 24
3.9 การออกแบบการสร้าง FFT ผ่านโปรแกรม MATLAB 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่		หน้า
3.10	บล็อคไดอะแกรมการออกแบบการทำงานของ FPGA	26
3.11	การสร้างโฟลเดอร์สำหรับบันทึกผลการทดลอง	27
3.12	เลือกฟังก์ชันใน MEGA CORE FUNCTION	27
3.13	ตั้งชื่อไฟล์ MEGA CORE FUNCTION	28
3.14	เลือกอุปกรณ์ FPGA EP4CE22F17C6N	28
3.15	ตั้งค่า PARAMETERIZE ใน IP TOOLBENCH	29
3.16	ตั้งค่า TRANSFORM LENGTH, DATA PRECISION, TWIDDLE PRECISION	29
3.17	ตั้งค่า I/O DATA FLOW	30
3.18	ตั้งค่า IMPLEMENTATION OPTION	30
3.19	ตั้งค่า SIMULATION	31
3.20	JTAG TEST BENCHES	31
3.21	การ SIMULATING IN THIRD-PARTY SIMULATION TOOLS โดยใช้ NATIVELINK	32
3.22	PIN PLANNER	32
3.23	การ PROGRAMMER ลง BOARD FPGA	33
4.1	ผลการตรวจสอบการทำงานของการสร้าง FREQUENCY SWEEP ที่วัดได้จาก SCOPE โดยปรับค่าความชันใน RAMP เท่ากับ 0.01	34
4.2	ผลการตรวจสอบการทำงานของการสร้าง FREQUENCY SWEEP ที่วัดได้จาก SCOPE โดยปรับค่าความชันใน RAMP เท่ากับ 0.02	35
4.3	ผลการตรวจสอบการทำงานของการสร้าง FREQUENCY SWEEP ที่วัดได้จาก SCOPE โดยปรับค่าความชันใน RAMP เท่ากับ 0.05	35
4.4	ผลการตรวจสอบการทำงานของการสร้าง FREQUENCY SWEEP ที่วัดได้จาก SCOPE โดยปรับค่าความชันใน RAMP เท่ากับ 0.5	36
4.5	ผลการตรวจสอบการทำงานของการสร้าง FREQUENCY SWEEP ที่วัดได้จาก SCOPE โดยปรับค่าความชันใน RAMP เท่ากับ 0.2	36
4.6	ผลการตรวจสอบการทำงานของการสร้าง FREQUENCY SWEEP ที่วัดได้จาก SCOPE โดยปรับค่าความชันใน RAMP เท่ากับ 0.1	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่		หน้า
4.7	ผลการตรวจสอบการทำงานของการสร้าง FREQUENCY SWEEP ที่วัดได้จาก SCOPE โดยปรับค่าความชันใน RAMP เท่ากับ 0.005	37
4.8	ผลการตรวจสอบการทำงานของการสร้าง FREQUENCY SWEEP ที่วัดได้จาก SCOPE โดยปรับค่าความชันใน RAMP เท่ากับ 0.002	38
4.9	ผลการตรวจสอบการทำงานของการสร้าง FREQUENCY SWEEP ที่วัดได้จาก SCOPE โดยปรับค่าความชันใน RAMP เท่ากับ 1	38
4.10	ผลการตรวจสอบการทำงานของการสร้าง FREQUENCY SWEEP ที่วัดได้จาก SCOPE โดยปรับค่าความชันใน RAMP เท่ากับ -0.01	39
4.11	สเปกตรัมของอินพุทของโมเดลที่ทำการทดลอง จากจุด TEST A	40
4.12	ผลการตรวจสอบการทำงานของการสร้าง FIR FILTER จากจุด TEST B	40
4.13	ผลการทดสอบการออกแบบการสร้าง FIR FILTER แบบ 32 TAB จากจุด TEST A	41
4.14	ผลการทดสอบคำสั่ง STEM(FIR_COEFF) ของ FIR FILTER แบบ 32 TAB	42
4.15	ผลการหาสเปกตรัมของสัญญาณเอาท์พุทจากจุด TEST B	42
4.16	ผลการทดลองผ่าน FFT	43
4.17	ผลการสร้าง FFT MEGA CORE FUNCTION	44
4.18	ผลการ COMPILATION ใน QUARTUS II	44
4.19	ผลการ COMPILATION ใน QUARTUS II	45
4.20	ผลการ PROGRAMMER ลงบอร์ด FPGA	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
2.1	การกลับบิตดัชนีข้อมูล	7
2.2	จำนวนการคูณจำนวนเชิงซ้อนในการแปลงฟูเรียร์เต็มหน่วย	7
3.1	ตารางการตั้งค่าบล็อกการทำงานย่อยภายในส่วนของ FREQSWEEP	20
3.2	ตารางการตั้งค่า SINE WAVE BLOCKS	22

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการวิเคราะห์ ออกแบบและประมวลผลสัญญาณต่างๆนั้น จะอยู่ในรูปแบบการวิเคราะห์เชิงคณิตศาสตร์ ซึ่งหากสัญญาณที่วิเคราะห์มีลักษณะการเปลี่ยนแปลงทางเวลาไปเหมือนฟังก์ชันใดฟังก์ชันหนึ่งในทางคณิตศาสตร์ ก็จะเรียกสัญญาณนั้นตามชื่อเฉพาะนั้นๆ เช่น สัญญาณรูปไซน์เป็นต้น ส่วยการวิเคราะห์สัญญาณที่มีลักษณะการเปลี่ยนแปลงทางเวลาในรูปแบบอื่นๆ นั้นจำเป็นที่จำต้องหาวิธีการอื่นเข้ามาช่วยในการวิเคราะห์ขอคุณสมบัติของสัญญาณ เพื่ออธิบายถึงความสัมพันธ์ระหว่างสัญญาณอินพุท และเอาต์พุท รวมถึงคุณสมบัติต่างๆของสัญญาณ ซึ่งวิธีการหนึ่งที่จะนำมาใช้ในการวิเคราะห์สัญญาณแบบนี้ก็คือ การทำการกระจายสัญญาณที่ปรากฏในโดเมนเวลา (Time domain) ออกเป็นส่วนประกอบย่อยที่มีคลื่นรูปไซน์ที่มีขนาด ความถี่และเฟสต่างๆ ตามความเหมาะสมที่จะประกอบเป็นสัญญาณนั้น ซึ่งทำให้เกิดการมองสัญญาณได้อีกรูปแบบหนึ่งคือ การมองสัญญาณในโดเมนความถี่ (Frequency domain) ซึ่งหนึ่งในวิธีการแปลงสัญญาณจากโดเมนเวลาให้สามารถมองได้ในโดเมนความถี่นั่นก็คือ การแปลงฟูเรียร์อย่างเร็ว (FFT) ซึ่งถูกนำมาประยุกต์ใช้อย่างกว้างขวางในงานด้านการประมวลผลภาพและการประมวลผลสัญญาณ

ในโครงงานฉบับนี้จึงได้คิดที่จะนำเทคโนโลยีของ FPGA มาใช้ในการสร้างอุปกรณ์ที่ทำการสร้างการแปลงสัญญาณจากโดเมนเวลาให้อยู่ในรูปของโดเมนความถี่ (Frequency domain) โดยใช้วิธีการแปลงฟูเรียร์อย่างเร็ว (Fast Fourier Transform: FFT)

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาการออกแบบการแปลงฟูเรียร์แบบเร็ว
1.2.2 จำลองการทำงานการแปลงฟูเรียร์แบบเร็วบนโปรแกรมแมทแลป
1.2.3 แสดงผลการแปลงฟูเรียร์แบบเร็วด้วยเอฟพีจีเอ

1.3 ขอบเขตของโครงงาน

โครงงานนี้เป็นการศึกษาการออกแบบและจำลองการแปลงฟูเรียร์แบบเร็วด้วยเอฟพีจีเอ และนำไปเทียบผลการจำลองในโปรแกรมแมทแลป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 การแปลงฟูเรียร์แบบเร็ว

การแปลงฟูเรียร์เต็มหน่วย (Discrete Fourier Transform, DFT)นับเป็นวิธีการที่สำคัญในการวิเคราะห์การออกแบบ และการสร้างแบบในการประมวลสัญญาณเต็มหน่วย มีการวิเคราะห์และประมวลสัญญาณจำนวนมากที่กระทำภายใต้แวดล้อมทางความถี่หรือโดเมนฟูเรียร์(Frequency Domain or Fourier Domain) การแปลงฟูเรียร์นี้สามารถที่จะคำนวณโดยทางตรงหรือใช้ขั้นตอนวิธีการคำนวณแบบ เร็ว (Fast Fourier Transform, FFT) ก็ได้

2.1.1 การคำนวณการแปลงฟูเรียร์เต็มหน่วยโดยตรง (Direct Computation DFT)

พิจารณา x(n) ซึ่งเป็นชุดแถวของสัญญาณเต็มหน่วยอินพุตที่ประกอบด้วย N จุดตัวอย่าง X(k) ก็คือชุดแถว ของสัญญาณที่ได้จากการแปลงฟูเรียร์ และจะมีจำนวน N จุดตัวอย่าง เช่นเดียวกัน

$$X(k)=\begin{cases}\sum_{n=0}^{N-1} x(n)e^{-\frac{j2\pi nk}{N}} & ;0\le k\le -1 \quad DFT \\ 0 & elsewhere\end{cases} \tag{2.1}$$

และ

$$x(n)=\begin{cases}\sum_{n=0}^{N-1} X(k)e^{\frac{j2\pi nk}{N}} & ;0\le k\le -1 \quad iDFT \\ 0 & elsewhere\end{cases} \tag{2.2}$$

จะสังเกตได้ว่า การคำนวณหา x(n) จาก X(k) หรือ Inverse DFT (iDFT) นั้นเป็นวิธีเดียวกันกับการคำนวณ X(k) จาก x(n) สำหรับการคำนวณแต่ละองค์ประกอบความถี่ X_k จะต้องใช้การคูณจำนวนเชิงซ้อน N ครั้ง และใช้การบวกจำนวนเชิงซ้อน (N-1) ครั้ง แต่ละชุดข้อมูลก็จะมี N ตัวอย่าง ดังนั้นออเดอร์ของการคูณ และการบวกจึงเขียนได้เป็น

$$O(\text{Multiplication}) = N^2 \tag{2.3}$$

$$O(\text{Addition}) = N(N-1)\approx N^2 \tag{2.4}$$

เนื่องจากมีการกำหนดมาก่อนแล้วถึงความเป็นวัฏจักร (Cyclic หรือ Periodic) การคำนวณ DFT จึงต้องอาศัยรากเอกภาพอันดับ N (Nth root of Unity) ในที่นี้เพื่อความสะดวกจึงให้

$$W_N=e^{-\frac{j2\pi}{N}} \tag{2.5}$$

โดยอาศัยคุณสมบัติสังยุกต์สมมาตร(Complex Conjugate Symmetry) และการสมคาบ (Periodicity) ของ $e^{-\frac{j2\pi nk}{N}}=W_N^{nk}$ สำหรับ n และ k

$$W_N^{k(N-n)}=W_N^{-nk}=(W_N^{nk})^* \tag{2.6}$$

และ

$$W_N^{nk}=W_N^{k(N+n)}=W_N^{n(k+N)} \tag{2.7}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติดังกล่าวเป็นคุณสมบัติที่มีประโยชน์ในการพัฒนาการแปลงฟูเรียร์แบบเร็วอันจะได้กล่าวต่อไป การประมวลข้อมูลจำนวนมาก ๆ โดยใช้วิธีคำนวณโดยตรงนั้นมักจะไม่นิยมเพราะสิ้นเปลืองเวลา

ในการ คำนวณมากเกินไป อย่างไรก็ตามการคำนวณโดยตรงมีโครงสร้างการคำนวณที่ไม่ซับซ้อน ทำให้สามารถที่จะสร้างวงจรคำนวณในแบบหน่วยแถว (Systolic Array) ได้ง่าย หรือสามารถที่จะใช้การประมวลแบบขนาน (Parallel Processing) ได้สะดวก

2.1.2 การคำนวณการแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform; FFT)

ในการคำนวณการแปลงฟูเรียร์เต็มหน่วยโดยตรงนั้นมีการสิ้นเปลืองเวลาในการคำนวณมาก มักไม่ได้รับ ความนิยม โดยเฉพาะเมื่อ N มีค่ามาก ดังนั้นจึงมีขั้นตอนวิธีจำนวนอื่นอีกมากที่ได้รับการพัฒนาจนใช้งาน ได้อย่างเกิดประสิทธิภาพเมื่อเทียบกับการคำนวณโดยตรง ขั้นตอนวิธีส่วนใหญ่จะขึ้นอยู่กับจำนวนข้อมูล ในชุดแถวหรือขนาดของบล็อค (Block Size, N) เป็นสำคัญ เช่น ขั้นตอนวิธีตัวประกอบปฐม (Prime Factor Algorithm, PFA) เหมาะสมกับ N ที่เป็นจำนวนปฐม (เช่น 2, 3, 5, 7, 11, 13, 17, 19 เป็นต้น) ขั้นตอนวิธีแยกตัวประกอบมักจะเหมาะสมกับ N ที่แยกตัวประกอบได้ หรือ Composite Block Length (เช่น 4 = 2x2, 6 = 3x2, 12 = 3x3x2, 15 = 5x3, 255 = 3x5x17 เป็นต้น) ขั้นตอนวิธีแยกตัวประกอบที่ลด การคำนวณได้สูงสุดเห็นจะได้แก่วิธีของ Cooley-Tukey ซึ่งเหมาะสมกับ N=2m (เช่น N = 2, 4, 8, 16, 32, 64, 128, 256, 512 , 1024 ... เป็นต้น)
วิธีของ Cooley-Tukey ได้ย่อยการคำนวณเป็นชุดเล็ก ๆ จำนวนมากที่แทบจะไม่ต้องอาศัยการคูณความซับซ้อนของการคำนวณจาก N^2 จึงกลายมาเป็น $N\log_2 N$ วิธีดังกล่าวนี้อาศัยการลดทอนทางเวลา (เริ่มการ แบ่งจากกลุ่มของข้อมูล, x(n) ก่อน) หรือการลดทอนทางความถี่(เริ่มการแบ่งจากกลุ่มของผลลัพธ์, X(k) ก่อน) ก็ได้

2.1.2.1 การคำนวณการแปลงฟูเรียร์แบบเร็วโดยการลดทอนทางเวลา (Decimation in Time Fast Fourier Transform; DIT FFT)

สำหรับชุดข้อมูล N = 2m เมื่อ m คือเลขจำนวนเต็มบวกใด ๆ x(n) เป็นอินพุทที่ดัชนีต่าง ๆ การแปลงฟูเรียร์เต็มหน่วย สามารถเขียนได้โดย

$$X(k)=\sum_{n=0}^{N-1} x(n)W_N^{nk} \quad ;0\le k\le -1 \tag{2.8}$$

$$X(k)=\sum_{\substack{n=0\\ n=even}}^{N-2} x(n)W_N^{nk}+\sum_{\substack{n=1\\ n=odd}}^{N-1} x(n)W_N^{nk} \tag{2.9}$$

ให้ n = 2r สำหรับผลบวกชุดแรก และให้ n =2r+1 สำหรับผลบวกชุดหลัง จะได้

$$X(k)=\sum_{r=0}^{\frac{N}{2}-1} x(2r)W_N^{2rk}+\sum_{r=1}^{\frac{N}{2}-1} x(2r+1)W_N^{(2r+1)k} \tag{2.10}$$

จาก $W_N^{(2r+1)k}=W_N^{2rk}.W_N^k$ และ $W_N^{2rk}=(W_N^2)^{rk}=e^{-\frac{j2\pi}{N}2rk}=e^{-\frac{j2\pi}{N/2}rk}=W_{N/2}^{rk}$ จะได้

$$X(k)=\sum_{r=0}^{\frac{N}{2}-1} x(2r)W_{N/2}^{rk}+W_N^k\sum_{r=1}^{\frac{N}{2}-1} x(2r+1)W_{N/2}^{rk} \tag{2.11}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อให้ $g(r)=x(2r)$ และ $h(r)=x(2r+1)$ จะได้

$$X(k)=\sum_{r=0}^{\frac{N}{2}-1} g(r)W_{N/2}^{rk}+W_N^k\sum_{r=1}^{\frac{N}{2}-1} h(r)W_{N/2}^{rk} \qquad 0\le k\le N-1 \tag{2.12}$$

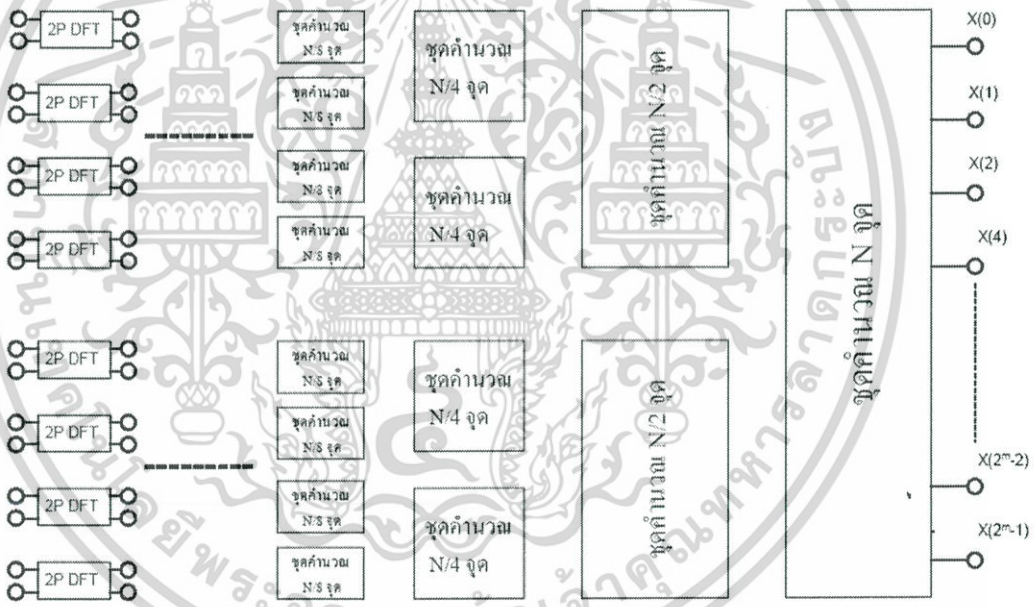
แม้ว่าดัชนี k ในสมการจะมีค่าจาก 0 ถึง N-1 แตละส่วนผลบวกก็จะใช้เพียงครึ่งเดียวเท่านั้น และเมื่อให้ G(k) และ H(k) เป็นผลการแปลงฟูเรียร์เต็มหน่วย ของ g(n) และ h(n) ตามลำดับ จะได้

$$X(k)=G(k)+W_N^kH(k) \qquad 0\le k\le \frac{N}{2} \tag{2.13}$$

เนื่องจากคาบของ G(k) และ H(k) มีค่าเป็น N/2 ค่าของ X(k) ที่เหลือจึงหาได้จาก

$$X(k)=G\left(k-\frac{N}{2}\right)+W_N^kH\left(k-\frac{N}{2}\right) \qquad \frac{N}{2}\le k\le N-1 \tag{2.14}$$

การคำนวณตามสมการ ที่(2.14) สามารถที่จะแสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 การคำนวณการแปลงฟูเรียร์แบบเร็วโดยการลดทอนทางเวลา

ในทำนองเดียวกันการคำนวณการแปลง G(k) และ H(k) ก็สามารถที่จะแยกย่อยไปได้อีกเรื่อย ๆ จนมีขนาดน้อยที่สุด คือ 2 จุด แสดงได้โดย

$$X(k)=\sum_{n=0}^{N-1} x(n)W_N^{nk}=\sum_{n=0}^{1} x(n)W_N^{nk} \tag{2.15}$$

ซึ่งอาจเขียนในรูปของเมตริกซ์ ได้ดังนี้

$$\begin{bmatrix}X(0)\\X(1)\end{bmatrix}=\begin{bmatrix}W_2^0 & W_2^0\\W_2^0 & W_2^1\end{bmatrix}\begin{bmatrix}x(0)\\x(1)\end{bmatrix} \tag{2.16}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

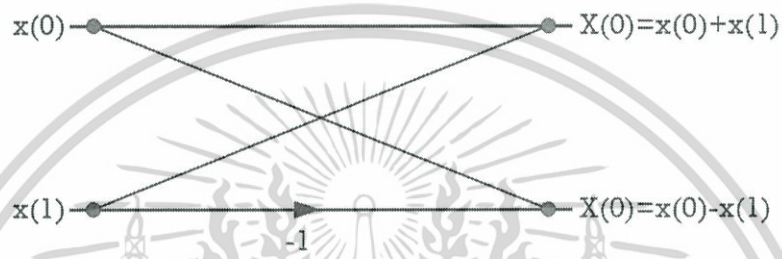
เมื่อ $W_2^0=e^0=1$ และ $W_2^1=e^{-j2\pi/2}=cos(\pi)+jsin(\pi)=-1$
หรือจะได้

$$X(0) = x(0)W_2^0 + x(1)W_2^0 = x(0) + x(1) \quad (2.17)$$

และ

$$X(1) = x(0)W_2^0 + x(1)W_2^1 = x(0) - x(1) \quad (2.18)$$

สามารถที่จะเขียนกราฟไหลสัญญาณ (Signal Flow Graph) ได้ดังรูป

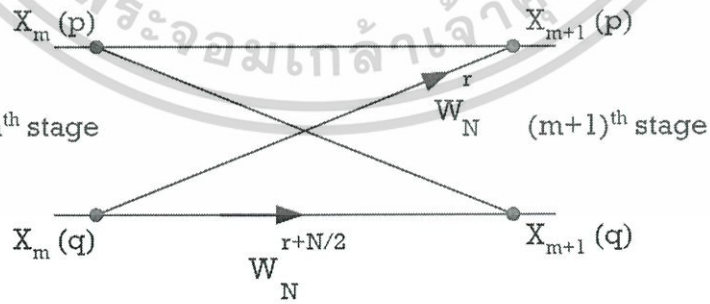


รูปที่ 2.2 กราฟไหลสัญญาณของการแปลง 2 จุด

สังเกตว่าในการลดทอนแต่ละครั้งผ่าน (Pass) ขนาดของบล็อคที่จะต้องทำการแปลงจะลดลงครึ่งหนึ่งเสมอ ดังนั้นจำนวนครั้งผ่านทั้งหมดจึงเป็น $\log_2 N$ ในแต่ละครั้งผ่าน จะมีการคูณ N ครั้ง (คือคูณด้วย W_N^k) จำนวนการคูณทั้งหมดจึงเป็น $N\log_2 N$ ซึ่งเมื่อเทียบกับ N^2 แล้วจะน้อยกว่ามาก

2.1.2.2 แฟกเตอร์ตัวบิดและหน่วยคำนวณผีเสื้อ (Twiddle Factor & Butterfly Unit)

การทำงานขั้นพื้นฐานของการแปลงก็คือการเรียงดัชนีใหม่การคูณด้วย W_N^k และการบวกกันของข้อมูล การคูณเสมือนทำให้ตำแหน่งของมูลย้ายไป จึงเรียก W_N^r ว่าเป็นแฟกเตอร์ตัวบิด (Twiddle Factor) การบวกกันเป็นคู่ ๆ มีลักษณะเป็นแบบเดิมเสมอและไขว้แทยงเหมือนตัวผีเสื้อ ดังจะแสดงให้ชัดเจน



รูปที่ 2.3 กราฟการไหลของสัญญาณในหน่วยคำนวณผีเสื้อ

$$X_{m+1}(p)=X_m(p)+W_N^r X_m(q) \quad (2.19)$$

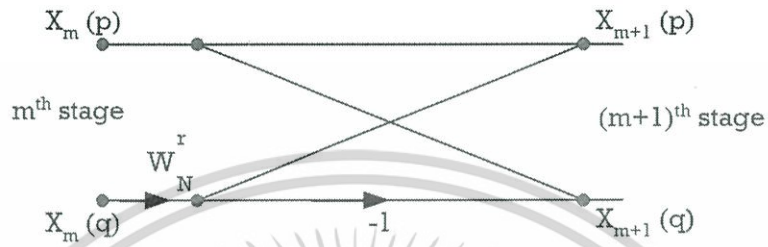
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X_{m+1}(q)=X_m(p)+W_N^{r+N/2} X_m(q) \quad (2.20)$$

จากสมการ

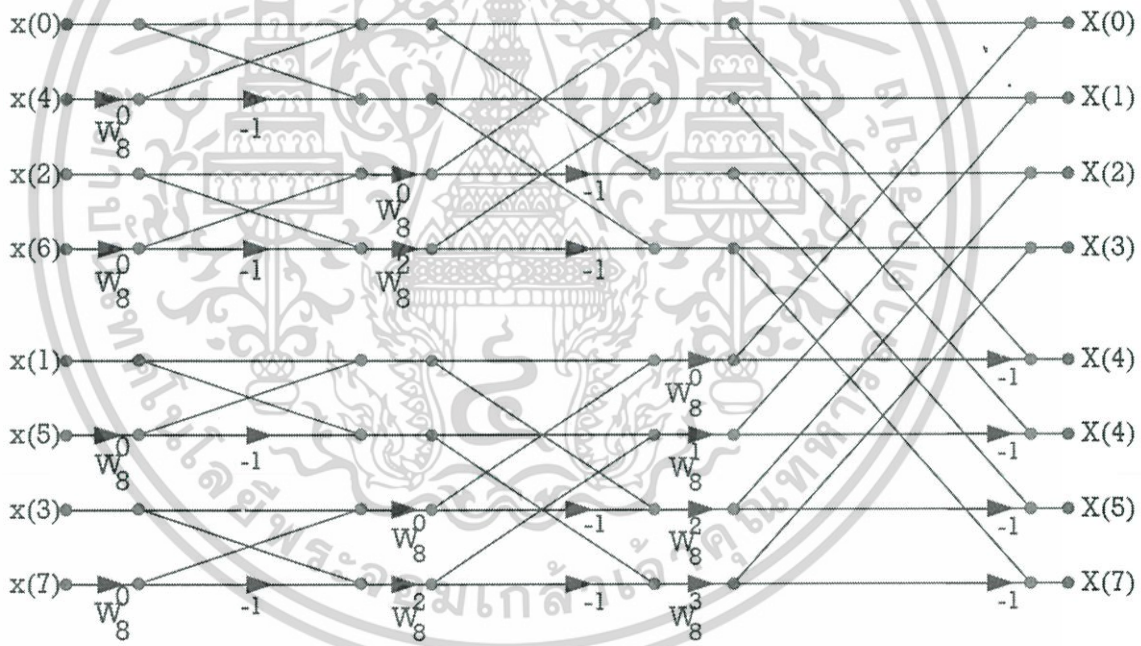
$$X_{m+1}(q)=X_m(p)+W_N^{r+N/2}X_m(q) \qquad (2.21)$$

จากสมการ (2.20) และสมการ (2.21) จะเขียนรูปหน่วยคำนวณผีเสื้อได้ใหม่เป็น



รูปที่ 2.4 หน่วยคำนวณผีเสื้อที่ลดรูปแล้ว

จะสังเกตเห็นว่าหน่วยคำนวณนี้ใช้การคูณ(จำนวนเชิงซ้อน)เพียง 1 ครั้ง จากตัวอย่างการแปลง 8 จุด จะเขียน กราฟไหลของสัญญาณได้เป็น



รูปที่ 2.5 การแปลง 8 จุดโดยใช้หน่วยคำนวณผีเสื้อที่ลดรูปแล้ว (DIT)

การคูณด้วย -1 เป็นการทำงานที่ง่ายเพราะเพียงแค่กลับเครื่องหมายเท่านั้น และยังมีการคูณอีกมาก (= 2^m- 1ครั้ง) ที่คูณด้วย 1 (เพราะว่า $W_N^0 = 1$)

2.1.2.3 การกลับบิตและภาระการคำนวณ (Bit Reversing & Computation Lond)

การแปลงฟูเรียร์เต็มหน่วยแบบเร็วโดยอาศัยการลดทอนทางเวลา ตำแหน่งข้อมูลทางด้านเอ้าพุทจะเรียงกันตามลักษณะปกติ แต่ตำแหน่งข้อมูลทางด้านอินพุทจะบิดเบือนไป (เช่น 0, 4, 2, 6, 1, 5, 3, 7 ในกรณีการแปลง 8 จุด) ก่อนที่จะเริ่มการคำนวณ จะต้องจัดวางข้อมูลในตำแหน่งที่เหมาะสมก่อน การจัดตำแหน่งนี้มีกฎเกณฑ์ที่เรียกว่าการกลับบิต ดังจะยกตัวอย่างตามตารางข้างล่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 การกลับบิตดัชนีข้อมูล

ตำแหน่ง	รหัสฐานสอง	รหัสที่กลับบิต	ดัชนีข้อมูล
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

ในการแปลงฟูเรียร์เต็มหน่วยแบบเร็ว N จุด หากใช้ไมโครโปรเซสเซอร์จะต้องเสียส่วนเก็บข้อมูล 5N ตำแหน่ง (2N สำหรับเก็บข้อมูลอินพุท 2N สำหรับข้อมูลเอ้าท์พุท และ N สำหรับแฟกเตอร์ตัวบิต) และสามารถลงเหลือ 3N ตำแหน่ง หากเป็นการคำนวณ แบบแทนที่ (Inplace Computation) เพราะว่าบางครั้งกาหไม่ได้นำข้อมูลอินพุทไปใช้ต่อ ตำแหน่งนั้นก็จะใช้เก็บผลการคำนวณในแต่ละรอบจนได้เอ้าท์พุทให้สังเกตว่าแม้ข้อมูลอินพุทจะเป็นจำนวนจริง แต่ในระหว่างการคำนวณจะทำให้ผลลัพท์ เป็นจำนวนเชิงซ้อนเสมอ สำหรับ N ตำแหน่ง ส่วนของส่วนเก็บข้อมูลแฟคเตอร์ตัวบิต N/2 ใช้สำหรับเก็บ cos nθ และ N/2 ใช้สำหรับเก็บ sin nθ เมื่อ $\theta = \frac{2\pi}{N}$ และ n = 0.....N/2-1

การแปลงฟริเยร์เต็มหน่วยแบบเร็ว N จุด จะมีครั้งผ่านทั้งหมด $\log_2 N$ ครั้งผ่าน มีหน่วยคำนวณผีเสื้อทั้งหมด $^N/_2 \log_2 N$ หน่วย อาศัยการคูณจำนวนเชิงซ้อนทั้งหมด $N \log_2 N$ ครั้ง ตารางข้างล่างนี้แสดงให้เห็นภาระการคูรเมื่อเทียบกับการคำนวณโดยตรงที่ใช้การคูณ N^2 ครั้ง

ตารางที่ 2.2 จำนวนการคูณจำนวนเชิงซ้อนในการแปลงฟูเรียร์เต็มหน่วย

จำนวนจุด(N)	N^2	$N \log_2 N$
8	64	24
16	256	64
64	4,096	384
256	65,536	2,048
512	262,144	4,608
1,024	1,048,576	10,240

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.4 การคำนวณการแปลงฟูเรียร์แบบเร็วโดยการลดทอนทางความถี่ (Decimation in Frequency Fast Fourier Transform; DIF FFT)

ชุดข้อมูล $N=2^m$ เมื่อ m คือเลขจำนวนเต็มบอกใดๆ x(n) เป็นอินพุทที่ดัชนีต่างๆ การแปลงฟูเรียร์เต็มหน่วยสามารถเขียนได้โดย

$$X(k)=\sum_{n=0}^{N-1} x(n)W_N^{nk}, \quad 0\le k\le N-1$$

$$=\sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{nk}+\sum_{n=\frac{N}{2}}^{N-1} x(n)W_N^{nk} \tag{2.22}$$

เมื่อให้ $r = n - {}^{N}/_{2}$ จะได้ $n = r + {}^{N}/_{2}$

ดังนั้น

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{nk}+\sum_{r=0}^{\frac{N}{2}-1} x\left(r+\frac{N}{2}\right)W_N^{\left(r+\frac{N}{2}\right)k} \tag{2.23}$$

เพราะว่า $W_N^{\frac{n}{2}k} = e^{-i\pi k} = (-1)^k$ โดยที่ทั้ง r และ n เป็นดัชนีอันดับเดียวกัน จึงได้

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1}\left\{x(n)+(-1)^k x\left(n+\frac{N}{2}\right)\right\}W_N^{nk} \tag{2.24}$$

จัดเป็นกลุ่มเทอมคู่และเทอมคี่ ในช่วง: $0 \le r \le {}^{N}/_{2} - 1$

สำหรับเทอมคู่ k = 2r

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1}\left\{x(n)+(-1)^k x\left(n+\frac{N}{2}\right)\right\}W_N^{nk} \tag{2.24}$$

$$X(2r)=\sum_{n=0}^{\frac{N}{2}-1}\left\{x(n)+(-1)^{2r} x\left(n+\frac{N}{2}\right)\right\}W_N^{2nr}$$

$$=\sum_{n=0}^{\frac{N}{2}-1}\left\{x(n)+x\left(n+\frac{N}{2}\right)\right\}W_N^{nr} \tag{2.25}$$

สำหรับเทอมคี่ k = 2r + 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X(2r+1)=\sum_{n=0}^{\frac{N}{2}-1}\left\{x(n)+(-1)^{2r+1}x\left(n+\frac{N}{2}\right)\right\}W_N^{n(2r+1)}$$

$$=\sum_{n=0}^{\frac{N}{2}-1}\left\{x(n)+x\left(n+\frac{N}{2}\right)\right\}W_N^{n}W_{\frac{N}{2}}^{nr} \qquad (2.26)$$

โดยกำหนดให้

$$g(n)=x(n)+x(n+\frac{N}{2}) \qquad (2.27)$$

$$h(n)=x(n)-x(n+\frac{N}{2}) \qquad (2.28)$$

ดังนั้น

$$X(2r)=\sum_{n=0}^{\frac{N}{2}-1}g(n)W_{\frac{N}{2}}^{nr} \qquad 0\le r\le\frac{N}{2} \qquad (2.29)$$

$$X(2r+1)=\sum_{n=0}^{\frac{N}{2}-1}h(n)W_N^{n}W_{\frac{N}{2}}^{nr} \qquad 0\le r\le\frac{N}{2} \qquad (2.30)$$

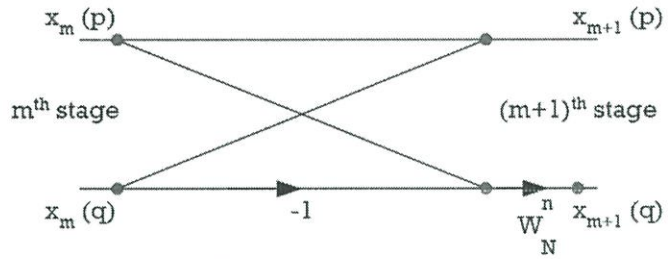
จะเห็นได้ว่า X(k) ซึ่งเป็นการแปลง N จุด ได้จากการแปลง N/2 จุดจากตัวอย่าง g(n) และ h(n) W_N^n การกระทำสามารถที่จะแบ่งครึ่งออกไปได้อีกเรื่อยๆจนเหลือ 1 คู่ ในขั้นตอนสุดท้าย แต่ละคู่ของการแปลงสามารถที่จะเขียนเป็นหน่วยคำนวณผีเสื้อได้เช่นเดียวกัน

$$x_{m+1}(p)=x_m(p)+x_m(q) \qquad (2.31)$$

$$x_{m+1}(q)=\{x_m(p)-x_m(q)\}W_N^n \qquad (2.32)$$

ซึ่งสามารถเขียนกราฟไหลสัญญาณได้ดังรูป

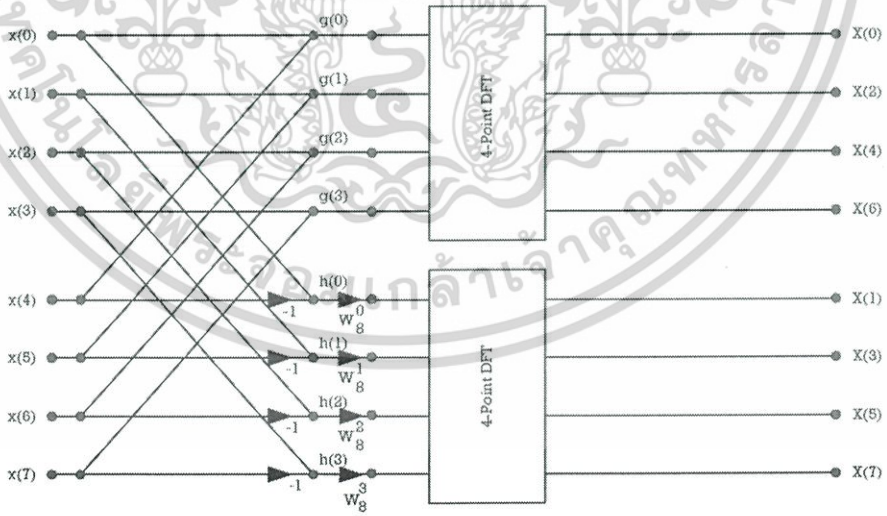
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 หน่วยคำนวณผีเสื้อของการแปลงฟูเรียร์โดยการลดทอนทางความถี่

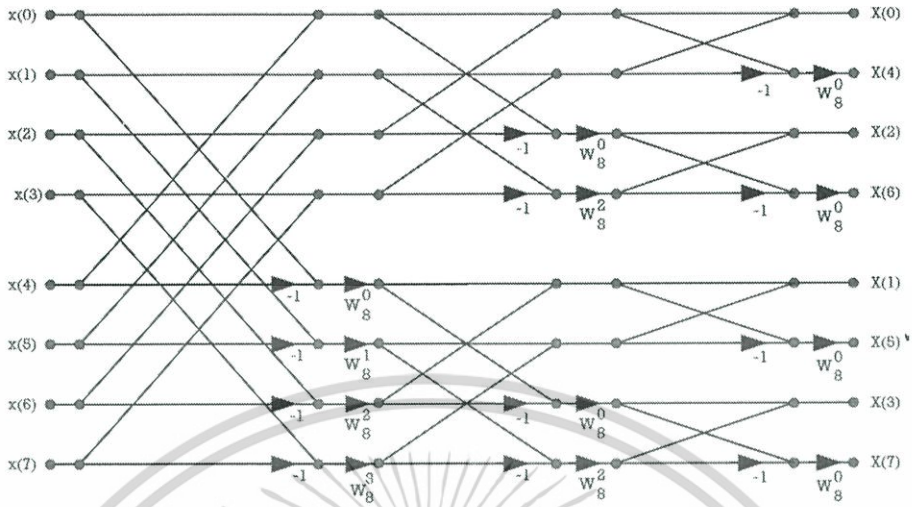
จำนวนครั้งผ่านที่ต้องการคือ $\log_2 N$ โดยแต่ละครั้งผ่านจะมีการคำนวณในผีเสื้อ N/2 หน่วย ดังนั้นจะสังเกตเห็นได้ว่าการแปลงฟูริเยร์ไม่ว่าจะอาศัยการลดทอนทางเวลาหรือการลดทอนทางความถี่โครงสร้าง การคำนวณจะใกล้เคียงกันมาก ยกตัวอย่างการแปลงฟูริเยร์เต็มหน่วย 8 จุดโดยการลดทอนทางความถี่ จะสามารถแบ่งครั้งผ่านได้เป็น 3 ครั้งผ่าน ครั้งผ่านแรกจะทำงานกับข้อมูลทั้งหมด แฟกเตอร์ตัวบิดที่ใช้คือ w_8^0, w_8^1, w_8^2 และ w_4^3โดยที่ ข้อมูลที่เข้ามานั้นจัดลำดับอย่างปกติ ในครั้งผ่านที่ 2 จะแบ่งผลที่ได้มาเป็น 2 กลุ่ม กลุ่มละ 4 จุดข้อมูล

การแปลงฟูริเยร์เต็มหน่วย 4 จุดนั้นสามารถแยกย่อยเป็นการแปลงฟูริเยร์เต็มหน่วย 2 จุด 2 ชุด แฟกเตอร์ตัว บิดที่ใช้คือ w_4^0 และ w_4^1 ที่เหลือจึงการคำนวณการแปลง 2 ชุด ทั้งหมด 4 ชุด เมื่อใช้ความสัมพันธ์ $w_4^1 = w_8^2, w_4^0 = w_8^0$ และ $w_2^0 = w_8^0$ กราฟไหลสัญญาณของการแปลงฟูริเยร์เต็มหน่วย 8 จุดโดยการลดทอนทางความถี่จะแสดงได้ดังรูป พึงสังเกตว่าในการแปลงฟูริเยร์เต็มหน่วยเร็วโดยการลดทอนทางความถี่นั้น แม้ว่าข้อมูลทางด้านอินพุทจะ เรียงมาอย่างปกติ แต่การแยกดัชนีคู่และดัชนีคี่ในแต่ละครั้งผ่านจะทำให้เกิดการสลับตำแหน่ง สุดท้ายแล้ว ข้อมูลที่ได้ทางเอ้าท์พุทจะเรียงกันในลักษณะกลับบิทซึ่งจะต้องมีการสลับคืนจึงจะได้การเรียนแบบธรรมชาติ



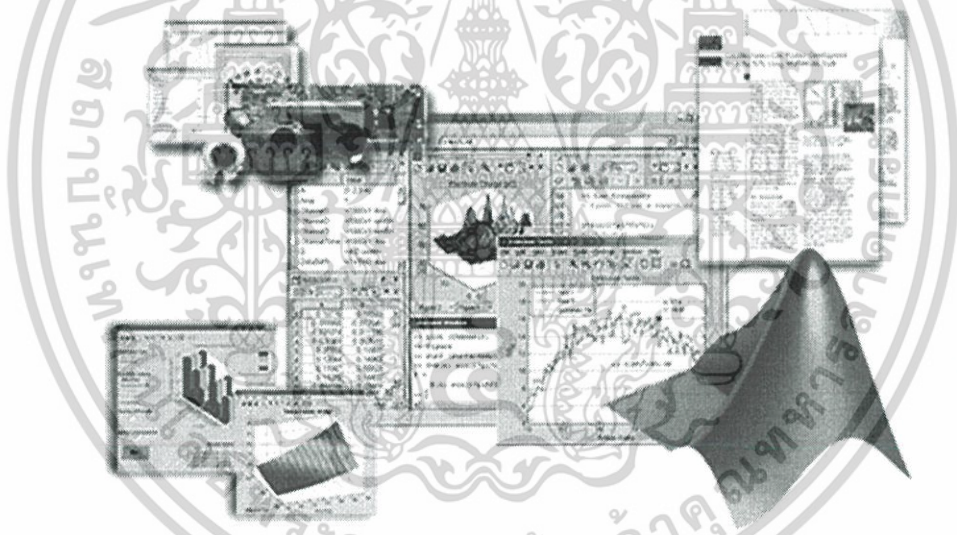
รูปที่ 2.7 กราฟไหลสัญญาณของการคำนวณโดยการลดทอนทางความถี่ขนาด 8 จุดครั้งผ่านแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 กราฟไหลสัญญาณแปลงฟูเรียร์เต็มหน่วยเร็วโดยการลดทอนทางความถี่ขนาด 8 จุด

2.2 โปรแกรม Matlab



รูปที่ 2.9 แสดงหน้าตาของโปรแกรม Matlab

Matlab เป็นโปรแกรมสำหรับการคำนวณเชิงตัวเลข และ Visualization ที่มีประสิทธิภาพสูง ชื่อของโปรแกรม “Matlab” ย่อมาจากคำเต็มว่า MATrix LABoratory และเป็นเครื่องหมายการค้าของบริษัท MathWorks ซึ่งการทำงานภายในโปรแกรม Matlab อยู่บนพื้นฐานของการคำนวณทางเมตริกซ์ (Matrix Manipulation and Computation) เป็นแกนหลัก โปรแกรม Matlab สามารถทำงานแบบโต้ตอบ (interactive) ซึ่งคล้ายๆ กับ ภาษา Basic ในโปรแกรม QBasic และแบบ compiled mode คล้ายๆ กับภาษา C และ Pascal นอกจากนี้เราสามารถใช้ Matlab เป็นเสมือนเครื่องคิดเลข ทำการคำนวณทางคณิตศาสตร์ได้ทันที โดยรูปร่างของโปรแกรมสามารถดูได้จากรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโปรแกรมแมทแลปสามารถทำได้ง่าย เมื่อเทียบกับภาษาโปรแกรมอื่นๆ เราสามารถสร้างและกำหนดค่าตัวแปรได้ทันทีโดยไม่ต้องประกาศตัวแปรก่อน ซึ่งสะดวกในการเขียนโปรแกรม เพราะผู้ใช้ไม่ต้องกังวลในเรื่องโครงสร้างของภาษา เช่น ในเรื่องการต้องประกาศตัวแปรก่อนใช้งาน แต่มาสนใจอัลกอริทึมในการแก้โจทย์ปัญหาเป็นหลัก ข้อมูลไม่ว่าจะเป็นตัวเลข หรือตัวอักษร (strings) จะถูกจัดเก็บในรูปแบบของแถว และหลัก หรือ array ซึ่งก็คือ matrix นั่นเอง เช่น จำนวนสเกลล่าร์ (scalar) จะถูกแทนด้วยเมตริกซ์ขนาด 1x1 ข้อมูลที่เป็นเวกเตอร์จะถูกแทนที่ด้วยเมตริกซ์ที่มีเพียง 1 แถว ในกรณีที่เป็นเวกเตอร์แบบแถว (Row vector) หรือ ถูกแทนที่ด้วยเมตริกซ์ที่มีเพียง 1 หลัก ในกรณีที่เป็นเวกเตอร์แบบหลัก (Column vector) เป็นต้น การที่ Matlab ถูกออกแบบมาให้มีการทำงานภายในเช่นนี้ ทำให้การเขียนโปรแกรมแก้โจทย์ปัญหาที่มีลักษณะของ vector และ matrix เป็นเรื่องง่าย

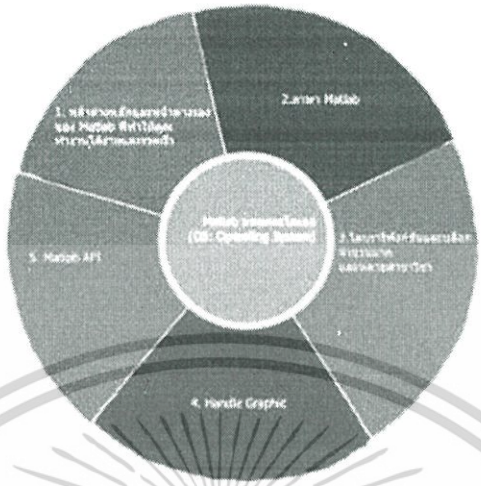
2.2.1 ความสามารถของโปรแกรม Matlab

1) Matlab เป็นโปรแกรมเพื่อการคำนวณและแสดงผลได้ทั้งตัวเลข และรูปภาพ ซึ่งมีประสิทธิภาพสูงสามารถทำการเขียนกราฟทั้ง 2 มิติ และ 3 มิติ ได้อย่างง่ายดาย และมีประสิทธิภาพ
2) เราสามารถควบคุมการทำงานของ Matlab ด้วยชุดคำสั่ง (command line) และยังสามารถรวบรวมชุดคำสั่งเป็นโปรแกรม (script file) ได้ด้วย
3) ลักษณะการเขียนโปรแกรมใน Matlab จะใกล้เคียงการเขียนสมการคณิตศาสตร์ที่เราคุ้นเคย จึงง่ายกว่าการเขียนโปรแกรมด้วยภาษาชั้นสูง เช่น ภาษา C, Pascal, Fortran และอื่นๆ
4) Matlab มีฟังก์ชั่นสำเร็จรูป (built-in function) เพื่อทำงานเฉพาะทางมากมาย นอกจากนี้ผู้ใช้ยังสามารถเขียนฟังก์ชั่นขึ้นมาใหม่โดยใช้ประโยชน์จากฟังก์ชั่นที่มีอยู่เดิมได้เพื่อให้เหมาะสมกับงานของผู้ใช้แต่ละกลุ่มสำหรับผู้ใช้ที่ต้องการใช้งานเฉพาะทางขั้นสูง เช่น งานด้าน Control, Image Processing, Artificial Neural Network หรืออื่นๆ Matlab ก็มี toolbox หรือชุด function พิเศษ เพื่อทำงานเฉพาะทางนั้นๆ ด้วย
5) Matlab สามารถเชื่อมโยงหรือส่งข้อมูลแบบ Dynamic Link กับโปรแกรมอื่นๆ ได้ เช่น Excel หรือโปรแกรมที่เขียนขึ้นเองจากภาษา C หรือ Visual Basic ที่ร่วมทำงานอยู่บนระบบปฏิบัติการ Windows

2.2.2 ส่วนประกอบสำคัญของ Matlab

Matlab ได้ออกแบบมาเพื่อสนับสนุนการทำงานของผู้ใช้อยู่ 5 ส่วน เพื่อช่วยในการวิเคราะห์ข้อมูล แสดงผลข้อมูล เชื่อมต่อกับสิ่งต่างๆ ภายนอกได้อย่างรวดเร็วและมีประสิทธิภาพสูงสุด ดังรูปที่ 2.10

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดงส่วนประกอบภายในของ Matlab

1) หน้าต่างหลักและหน้าต่างรองของ Matlab หรือที่เรียกกันว่า Matlab Desktop Environment ช่วยให้ผู้ใช้ทำงานได้ง่ายและรวดเร็วมากยิ่งขึ้น ในส่วนนี้ประกอบด้วยชุดเครื่องมือที่ช่วยให้เราสามารถใช้ฟังก์ชันก์และไฟล์ต่างๆ ด้วยเครื่องมือแบบรูปภาพ (GUI) ประกอบด้วยหน้าต่างย่อย Current Folder, Command Windows, Command History และ Workspace

2) ภาษา Matlab ใช้สำหรับเขียนแอพพลิเคชั่นหรือฟังก์ชันของคุณเอง ไว้ใช้งานโดยเฉพาะ Matlab เวอร์ชัน 2013a มีแอพพลิชั่นให้ดาวโหลดเพิ่มเติมได้ฟรีบนเว็บไซต์ www.mathworks.com โดยใช้ Matlab Editor เขียนในรูปฟังก์ชัน m-file พร้อมเครื่องมือตรวจสอบบัคในโปรแกรม

3) ไลบรารีฟังก์ชันและบล็อกไดอะแกรม จำนวนมากและครอบคลุมหลายสาขาวิชา ส่วนนี้จะรวบรวมฟังก์ชัน m-file หรือ mdl ของ Simulink เป็นไฟล์ย่อยๆ ไว้ โดยแต่ละไฟล์จะเป็นไฟล์ที่สร้างขึ้นมาเพื่อใช้กำหนดลักษณะในการคำนวณ เราเรียกว่า อัลกอลิทึม(Algorithms) แบบต่างๆ เริ่มจากฟังก์ชันง่ายๆ เช่นการบวก ฟังก์ชันตรีโกณมิติพื้นฐาน เช่น sine, cos, tan ไปจนถึงฟังก์ชันที่มีความซับซ้อนมีขั้นตอนในการคำนวณมากๆ เช่นการหาอินเวิร์สของเมตริกซ์ การหาค่าสมการเชิงซ้อน หรือการหารูทโลคัส เป็นต้น โดยส่วนนี้ประกอบด้วยฟังก์ชันพื้นฐาน เช่น บวล ลบ คูณ หาร sine, cost, log, x^2 และฟังก์ชันเฉพาะสาขาวิชา ซึ่งเราเรียกว่าทูลบ็อกซ์ (Toolbox) เช่น Control System, Bioinformatic ,Signal Processing, Fuzzy Logic , Aerospace, Image Processing, Econometrics และอื่นๆ

4) Handle Graphics เป็นไลบรารีฟังก์ชันใน Matlab สำหรับแสดงผลข้อมูล เป็นกราฟฟิก รูปภาพ เสียง วิดีโอ พร้อมด้วย ไลบรารี Guide เป็นเครื่องมือสำหรับสร้างแอพพลิเคชั่นแบบ GUI (Griphic User Interface) โดยเลียนแบบการเขียนโปรแกรมจาก Visual Basic ดังนั้นทำให้คุณสามารถนำอัลกอไปพัฒนาแอพพลิเคชั่นบน Matlab เพื่อใช้งานเองหรือแจกจ่ายได้อย่างรวดเร็ว

เอกสารนี้มีลิขสิทธิ์ ไม่ว่ากรณีใดๆ ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ประโยชน์ทางการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

5) Matlab API (API: Application Program Interface) เป็นส่วนติดต่อกับภาษาคอมพิวเตอร์อื่นๆ โปรแกรมภายนอก นามสกุลไฟล์ต่างๆ และสามารถพัฒนาให้เชื่อมต่อกับฮาร์ดแวร์ได้ง่าย

2.3 FPGA (Field Programmable Gate Arrays)

เป็นอุปกรณ์วงจรรวม(Integrated-Circuit Device) ที่ผู้ใช้สามารถจะทำการออกแบบวงจรหรือลอจิกการทำงานภายในทั้งหมด และทำการคอนฟิกูเรชั่นเองได้ โดยไม่ต้องส่งไปที่โรงงานผลิต FPGA จัดเป็น อุปกรณ์สารกึ่งตัวนำชนิดโปรแกรมได้ที่มีโครงข่ายการเชื่อมต่อภายในแบบแมตริกซ์ โครงสร้างภายในของ FPGA นั้นสามารถโปรแกรมให้มีหน้าที่การทำงานเหมือนลอจิกเกตพื้นฐาน เช่น AND, OR, XOR, NOT หรือรวมกันหลายๆ ชนิด (combinational logic) เพื่อให้ทำหน้าที่ที่มีความซับซ้อนเพิ่มขึ้น เช่น decoders หรือฟังก์ชันทางตณิตศาสตร์ ใน FPGAs ทั่วไปนอกจากจะประกอบด้วยส่วนของวงจรลอจิกแบบโปรแกรมได้แล้ว จะยังมีบล็อกของหน่วยความจำ ซึ่งอาจจะสร้างด้วยฟลิบฟลอปอย่างง่าย หรือใช้พื้นที่ของสารกึ่งตัวนำสร้างเป็นหน่วยความจำจริงๆ อยู่ภายในก็ได้

ในการออกแบบวงจรดิจิตอลอิเล็กทรอนิกส์ ที่มี FPGA อยู่บนแผงวงจรด้วยนั้น จะช่วยให้ผู้ออกแบบสามารถลดขนาดของแผงวงจร รวมทั้งสามารถออกแบบได้รวดเร็ว ไม่ต้องทดสอบรายละเอียดภายในให้เสร็จสมบูรณ์ 100 % ก็สามารถออกแบบแผงวงจรได้ เมื่อได้รับแผงวงจรและประกอบอุปกรณ์ต่างๆ เสร็จแล้ว จึงค่อยกำหนดหน้าที่การทำงานของ FPGA ได้ในภายหลัง ต่างจากการออกแบบด้วยลอจิกเกตขนาดเล็ก ที่ต้องออกแบบทางเดินของลายทองแดงให้เสร็จสมบูรณ์ก่อนและไม่สามารถแก้ไขได้ในภายหลัง นอกจากนี้ การใช้งาน FPGA สามารถโปรแกรมการทำงานได้ในทุกขณะแม้แต่ขณะที่ส่งมอบงานแล้ว ก็ยังสามารถเข้าไปแก้ไขวงจรได้โดยง่ายดาย จึงเป็นที่มาของคำว่า "field programmable" ซึ่งก็หมายถึงโปรแกรมได้ในภาคสนามหรือที่หน้างานนั่นเอง อย่างไรก็ตามข้อกำหนด (Configuration) ของ FPGA จะหายไปหลังจากปิดไฟเลี้ยง ดังนั้น จะต้องมีหน่วยความจำภายนอก (Flash) มาคอยรักษาข้อกำหนดของ FPGA ไว้ ซึ่ง FPGA จะมีกระบวนการอ่านข้อกำหนดนั้นโดยอัตโนมัติหลังจากได้รับไฟเลี้ยง

การทำงานของ FPGAs จะยังมีความเร็วที่ด้อยกว่าapplication-specific integrated circuit (ASIC) , และเมื่อเปรียบเทียบขนาดทางกายภาพ พบว่าจะมีความหนาแน่นของวงจรที่น้อยกว่ารวมทั้งใช้กำลังงานมากกว่า ASIC อย่างไรก็ตาม FPGA มีข้อได้เปรียบตรงที่ใช้เวลาในการพัฒนาผลิตภัณฑ์ (time to market) ที่น้อยกว่า สามารถแก้ไขวงจรได้หลังจากที่ใช้งานจริงในภาคสนาม , และมีค่าแรงในการดำเนินการที่ต่ำกว่า (non-recurring engineering) . นอกจากนี้ ยังมี FPGA ชนิดที่โปรแกรมได้ครั้งเดียว (OTP) ซึ่งมีราคาที่ต่ำกว่าโดย FPGA ชนิดนี้เมื่อโปรแกรมแล้วจะคล้ายกับ ASIC นอกจากนี้ยังมีการรวมหน่วยความจำ config เข้าไว้ในอุปกรณ์ FPGA ซึ่งจะยังคงอยู่แม้ปิดไฟเลี้ยง เรียกว่า Complex programmable logic devices

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ภาษา VHDL

VHDL ย่อมาจากคำว่า VHSIC Hardware Description Language (VHSIC :Very High Speed Integrated Circuit) เป็นภาษาโปรแกรมระดับสูง (High Level Language) ที่ใช้สำหรับการออกแบบฮาร์ดแวร์ในระบบดิจิตอล ตัวของภาษาสามารถบรรยายพฤติกรรมการทำงานในรูปของลำดับชั้น (Hierarchy) และ สามารถเขียนได้หลายรูปแบบ ด้วยเหตุผลนี้จึงทำให้ภาษา VHDL เป็นเครื่องมือที่ใช้ออกแบบตั้งแต่ขั้นตอนบนสุดคือ แนวความคิดที่จะแก้ปัญหา ลงไปทีละขั้นจนถึงขั้นตอนของการสร้างวงจรจริง และตัวภาษาก็เปิดโอกาสให้ได้พัฒนาและจำลองการทำงานของรูปแบบฟังก์ชันการทำงานของวงจรอย่างสังเขป โดยยังไม่ต้องคำนึงถึง รายละเอียดเกี่ยวกับโครงสร้างวงจรจริง นอกจากนั้น VHDL ยังเป็นภาษาที่สนับสนุนลักษณะต่างๆ ของระบบดิจิตอลที่มี ความซับซ้อนได้ ทั้งหมด วิวัฒนาการของภาษา VHDL เริ่มต้นประมาณปี ค.ศ. 1981 เมื่อกระทรวงกลาโหมสหรัฐอเมริกา หรือ DoD (Department of Defense) ได้พยายามปรับปรุงอุปกรณ์อิเล็กทรอนิกส์และคอมพิวเตอร์ที่ใช้ในกิจการทางทหาร ให้มีความทันสมัยมากขึ้นประกอบกับเทคโนโลยีทางด้านไมโคร-อิเล็กทรอนิกส์มีการพัฒนาไปอย่างรวดเร็วดังจะเห็นได้ จากการนำวงจรดิจิตอลหลายๆ วงจรมาทำการผลิตอยู่บนแผ่นซิลิกอนที่มีพื้นที่เพียง 1- 2 ตารางเซนติเมตรเท่านั้น ซึ่ง เป็นผลให้ประสิทธิภาพในการทำงานของวงจรสูงขึ้นตลอดจนความน่าเชื่อถือในการทำงานและความคงทนต่อสภาพ แวดล้อมสูง ในปี 1986 ภาษา VHDL ได้เริ่มมีการปรับปรุงภาษา VHDL เพื่อให้สามารถกำหนดเป็นมาตรฐานของ IEEE โดยสามารถประกาศเป็นมาตรฐานได้ในเดือนธันวาคมปี 1987 โดยอยู่ในหมวด IEEE 1076 – 1987 หลังจากนั้นก็ได้มีการพัฒนาปรับปรุงอย่างต่อเนื่องโดยได้มีประกาศปรับปรุงอีกครั้งในปี 1993 ซึ่งเรียกว่า IEEE 1076 – 1993 โดยได้มีการเพิ่มเติม Syntax พิเศษเพื่อให้ผู้ใช้สามารถใช้งานได้สะดวกมากยิ่งขึ้น จนถึงปัจจุบันนี้ ก็มี VHDL ออกมาหลายๆรุ่นแล้วเช่น VHDL 2006 และมาตรฐาน IEEE 1076-2008 ถูกตีพิมพ์ในเดือนมกราคม 2009 สำหรับขีดความสามารถในการออกแบบโดยใช้ภาษา VHDL นั้นสามารถออกแบบได้เฉพาะวงจรที่มีลักษณะเป็น Digital เท่านั้นส่วนวงจรที่เป็น Analog ในขณะนี้ยังไม่สามารถออกแบบได้โดยการใช้ภาษา VHDL

รูปที่ 2.11 พัฒนาการของภาษา VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1. ภาษา HDL (Hardware Description Languages)

HDL เป็นภาษาที่มีวัตถุประสงค์เพื่อใช้สำหรับการพัฒนาระบบอิเล็กทรอนิกส์ ซึ่งในปัจจุบันยังครอบคลุมเฉพาะระบบที่เป็นดิจิตอล ยังไม่ครอบคลุมระบบอะนาลอก มาตรฐานของภาษาต้องมีความสามารถดังต่อไปนี้

1.เขียนขึ้นเพื่อใช้ออกแบบระบบดิจิตอลได้

2.ใช้จำลองการทำงานของวงจรได้ (Simulation)

3.สามารถใช้สังเคราะห์วงจรได้ (Synthesis)

4.สามารถทดสอบการทำงานได้ (Testing)

5.ใช้เป็นเอกสารประกอบโครงการได้ (Documentation)

HDL เป็นภาษาที่ใช้ในการออกแบบทางด้านฮาร์ดแวร์ไม่ใช้ภาษาสำหรับการโปรแกรม (Program language) อย่างเช่นภาษา ซี เบสิก ADA หรือ ปาสคาล มีการพัฒนา HDL ขึ้นมาหลายภาษาที่สำคัญๆได้แก่ VHDL และ Verilog สำหรับ VHDL มีพื้นฐานมาจากภาษา ADA ส่วน Verilog มีพื้นฐานมาจากภาษาซี ทั้งสองภาษานี้มีผู้ใช้กันอย่างกว้างขวางดังนั้นผู้ที่พัฒนาเครื่องมือสำหรับการออกแบบระบบดิจิตอลจึงทำให้เครื่องมือนั้นรองรับการทำงานได้ทั้งสองภาษา

2.4.2. ข้อกำหนดของภาษา VHDL

เครื่องมือเพื่อการออกแบบระบบอิเล็กทรอนิกส์แบบอัตโนมัติ หรือ EDA (Electronic Design Automation) ส่วนใหญ่รองรับการท างานของ VHDL ได้ ซึ่งสามารถน าไปประยุกต์ใช้กับการพัฒนาฮาร์ดแวร์ระบบดิจิตอลได้หลายๆระดับตั้งแต่ใช้พัฒนาเป็นระบบ (System) ระดับบอร์ด (Boards) หรือการออกแบบให้อยู่ในรูปของไอซี ซึ่งก็ท าได้หลายๆชนิดเช่นกัน คือ ไอซีชนิด ASIC หรือไอซีประเภทโปรแกรมได้ (Programmable Logic Device PLD) ได้แก่ CPLD และ FPGA เป็นต้น ดังนั้นภาษา VHDL จึงมีความสามารถดังต่อไปนี้

1.สามารถออกแบบได้หลายระดับจากระดับพฤติกรรมจนถึงระดับเกต (Behavioral to Gate Level)

2.สนับสนุนการออกแบบแบบลำดับขั้น (Hierarchy Design)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

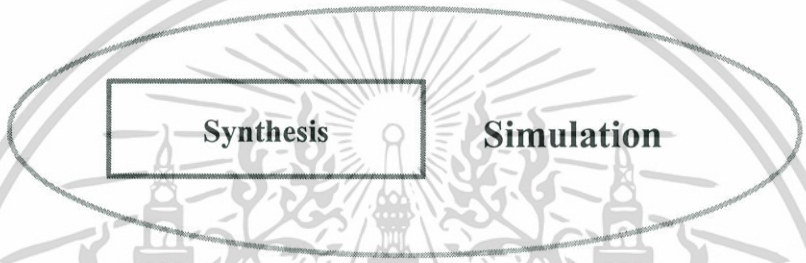
3.สนับสนุนระบบไลบรารี (Library Support)

4.สามารถควบคุมเวลาได้

5.สามารถทำงานได้ทั้งแบบขนานและแบบลำดับ (Concurrent และ Sequential)

6.สามารถกำหนดชนิด (Type) ของสัญญาณหรือข้อมูลได้

ภาษา VHDL มีความสามารถของ HDL ทุกประการเพียงแต่ความสามารถในการสังเคราะห์ เป็นวงจรจริง (Synthesis) ยังได้ไม่เท่ากับความสามารถในการจำลองการทำงานของวงจร (Simulation)

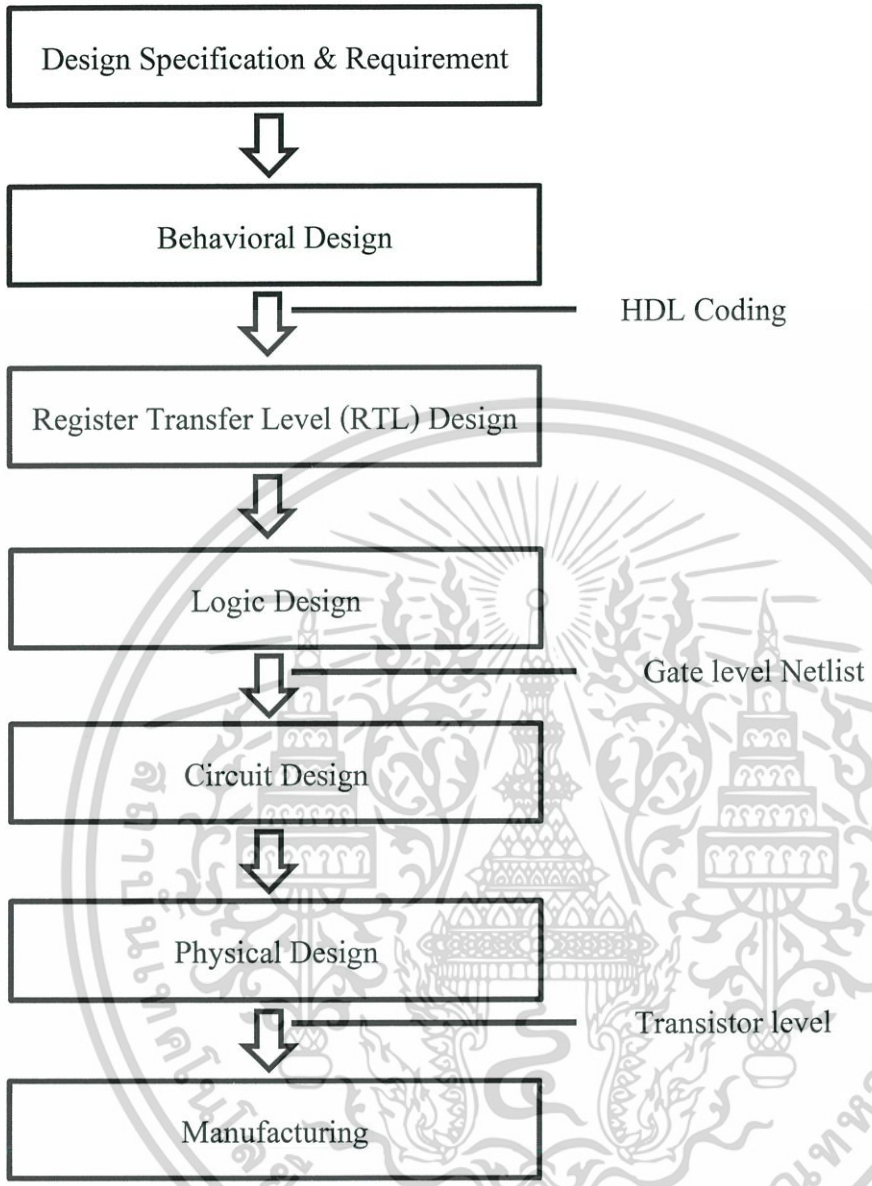


รูปที่ 2.12 แสดงขอบเขตของการจำลองการทำงานและการสังเคราะห์เป็นวงจรของภาษา VHDL

2.4.3. การออกแบบระบบดิจิตอลแบบจากบนลงล่าง (Top-Down)

กระบวนการออกแบบลักษณะจากบนลงล่าง มีขั้นตอนตามรูปที่ 2.13 โดยเริ่มต้นด้วยการกำหนดขอบเขตของงานที่ต้องการออกแบบ จากขอบเขตที่ได้จึงกำหนดเป็นฟังก์ชันการทำงานหรือพฤติกรรมการทำงานของระบบ ในขั้นตอนนี้สามารถทดสอบการทำงานได้แล้ว โดยการเขียนเป็นภาษา HDL แล้วทดสอบว่าฟังก์ชันดังกล่าวเป็นไปตามขอบเขตที่วางไว้หรือไม่ อีกทั้งในขั้นตอนนี้สามารถกำหนดค่าเวลาต่างๆของวงจรได้ เพื่อตรวจสอบการตอบสนองของระบบว่าอยู่ในขอบเขตของเวลาที่วางไว้หรือไม่ แต่ต้องไม่ลืมว่าการเขียนเป็นภาษาในขั้นตอนนี้อาจจะนำไปสังเคราะห์เป็นของจริงไม่ได้ แต่ก็ทำให้ทราบได้ว่าฟังก์ชันที่ออกแบบไว้สามารถให้ผลการทำงานตรงกับที่ต้องการหรือไม่ ขั้นตอนต่อไปเป็นการแปลงฟังก์ชันให้มีรายละเอียดมากขึ้นจนถึงระดับรีจิสเตอร์ ระดับหน่วยความจำ ระดับการประมวลผล และ ระดับ State machine ขั้นตอนต่อไปจะเป็นการแปลงสงไปในรายละเอียดมากขึ้นเป็นระดับโลจิกเกต ซึ่งทั้งสองขั้นตอนข้างต้นนี้สามารถจะเขียนและทดสอบด้วยภาษา HDL ได้ และการสังเคราะห์เป็นของจริงก็ทำได้มากกว่าในขั้นตอนแรก จากขั้นตอนนี้ไปจะเป็นขั้นตอนสำหรับการนำไปสังเคราะห์เป็นไอซี ASIC โดยการลงรายละเอียดของเกตให้ไปสู่ระดับวงจรทรานซิสเตอร์และระดับกายภาพ (Physical Design)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



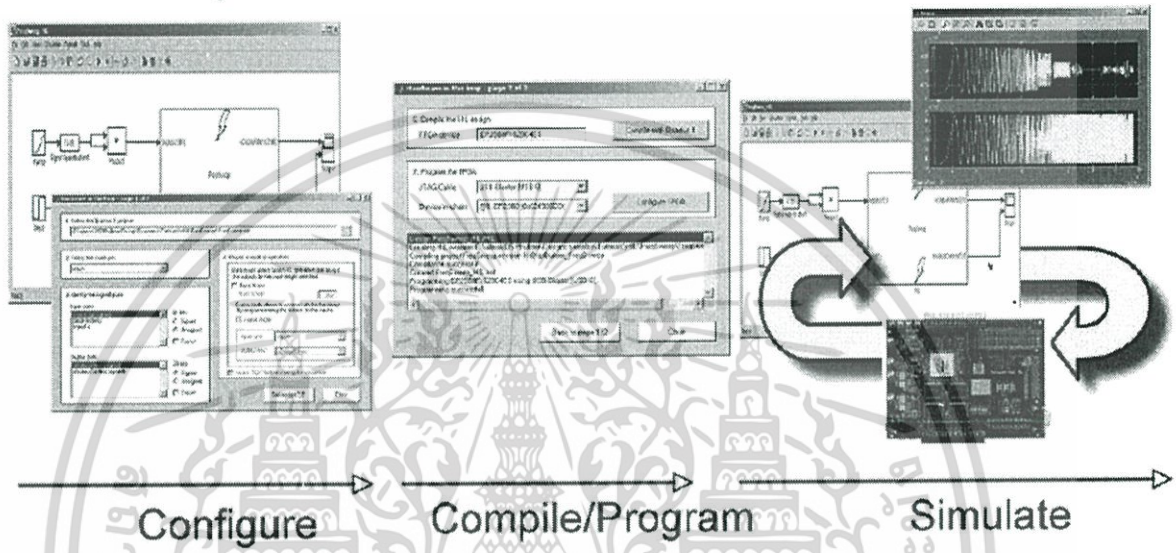
รูปที่ 2.13 ขั้นตอนการออกแบบระบบดิจิตอลแบบจากบนลงล่าง

การออกแบบตามขั้นตอนที่กล่าวมานี้ เรียกว่า “การออกแบบจากบนลงล่าง หรือ Top-down Design” ซึ่งมีข้อดีที่สามารถทดสอบการทำงานของระบบได้ก่อนว่าถูกต้องตรงกับความต้องการหรือไม่ อีกทั้งในแต่ละขั้นตอนยังสามารถปรับปรุงแก้ไขรายละเอียดเพิ่มเติมได้ ส่วนการออกแบบอีกลักษณะหนึ่งเรียกว่า “การออกแบบจากล่างขึ้นบน หรือ Bottom-up Design” ซึ่งเริ่มด้วยการเขียนโลจิกไดอะแกรมแล้วประกอบเป็นวงจรเสร็จแล้วจึงทดสอบการทำงาน การออกแบบลักษณะนี้ถ้าเป็นระบบขนาดเล็กสามารถทำได้สะดวกและรวดเร็ว แต่ถ้าเป็นระบบขนาดใหญ่กว่าจะรู้ผลว่าระบบทำงานถูกต้องหรือไม่ต้องใช้เวลามากและอาจมีความผิดพลาดได้ง่าย ดังนั้นการออกแบบระบบขนาดใหญ่จึงนิยมใช้วิธี Top-Down Design หรือใช้ทั้งสองวิธีร่วมกันก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการจัดทำโครงงาน

ทำการสร้างโมเดลแบบจำลองการทำงานกระบวนการประมวลผลโดยใช้ฟังก์ชัน Simulink ของโปรแกรม ผ่านบอร์ด เอฟพีจีเอ โดยการสร้างการแปลงฟูเรียร์แบบเร็วสามารถแบ่งส่วนการทำงานได้ดังรูปที่ 3.1

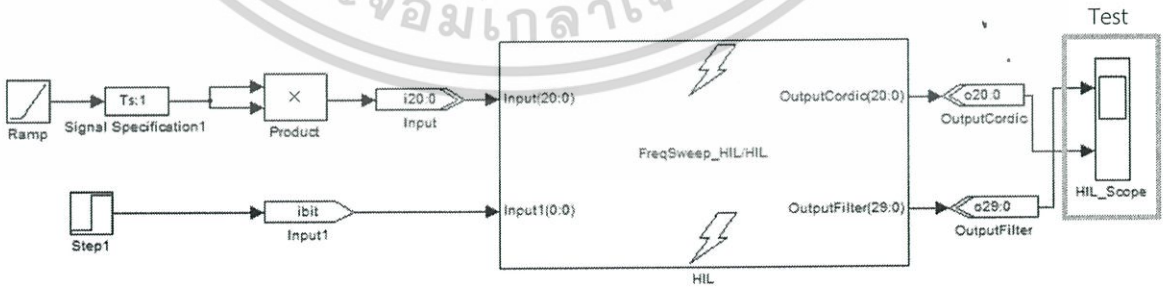


รูปที่ 3.1 แสดงโครงสร้างของส่วนการทำงาน Simulink ของโปรแกรม ผ่านบอร์ด เอฟพีจีเอ

3.1 การออกแบบ

3.1.1 การทดลองสร้างการจำลองการสร้าง Frequency Sweep ผ่านโปรแกรม MATLAB

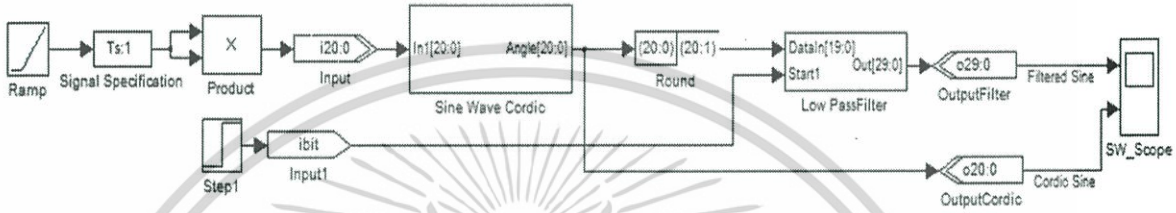
ทำการต่อโมเดลการสร้าง Frequency Sweep ตามรูปที่ 3.2



รูปที่ 3.2 โมเดลการทำ Frequency Sweep

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.2 ในส่วนของ FreqSweep_HIL/HIL Block นั้นจะประกอบด้วยการทำงานรวมกันของ ฟังก์ชั่นการทำงานต่างๆตามรูปที่ 3.3 โดยจะให้ผลที่จะกวาดความถี่ไปในย่านความถี่ที่ต้องการ



รูปที่ 3.3 บล็อกการทำงานย่อยภายในส่วนของ FreqSweep_HIL/HIL Block

ตารางที่ 3.1 ตารางการตั้งค่าบล็อกการทำงานย่อยภายในส่วนของ FreqSweep

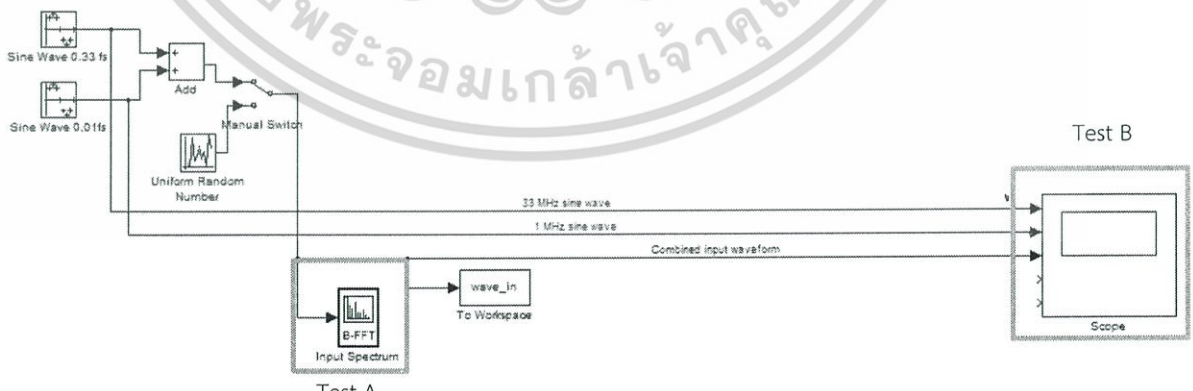
ชื่อบล็อก	การตั้งค่า
Ramp	Slope: 0.01,Start: 0, Initial output: 0, เลือก interpret vector parameter as 1-D
Signal Specification	Minimum: [], Maximum: [], Data type, Inherit: auto, Dimensions(-1 for inherited):-1, Variable-size signal: Inherit, Simple time(-1 for inherited): 1 Signal type: auto, Sampling mode auto
Product	Number of inputs: 2; Multiplication: Element-wise(.*), Simple time(-1 for inherited): -1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input(Altera Blockset)	Bus Type: Unsigned Integer, [Number Of Bits].[]:21
Round(Altera Blockset)	Bus Type: Signed Integer, [Number Of Bits].[]:21
Step	Step time: 1, Initial value:0, Final value: 1, Sample time: 1 เลือก Interpret vector parameters as 1-D
Input1	Bus Type: Single bit
OutputFilter(Altera Blockset)	Bus Type: Signed Integer, [Number Of Bits].[]:30 External type: Inferred
OutputCordic(Altera Blockset)	Bus Type: Signed Integer, [Number Of Bits].[]:21 External type: Inferred

3.1.2 การทดลองออกแบบการสร้าง FIR Filter ผ่านโปรแกรม MATLAB

ทำการต่อโมเดลการสร้าง FIR ตามรูปที่ 3.4



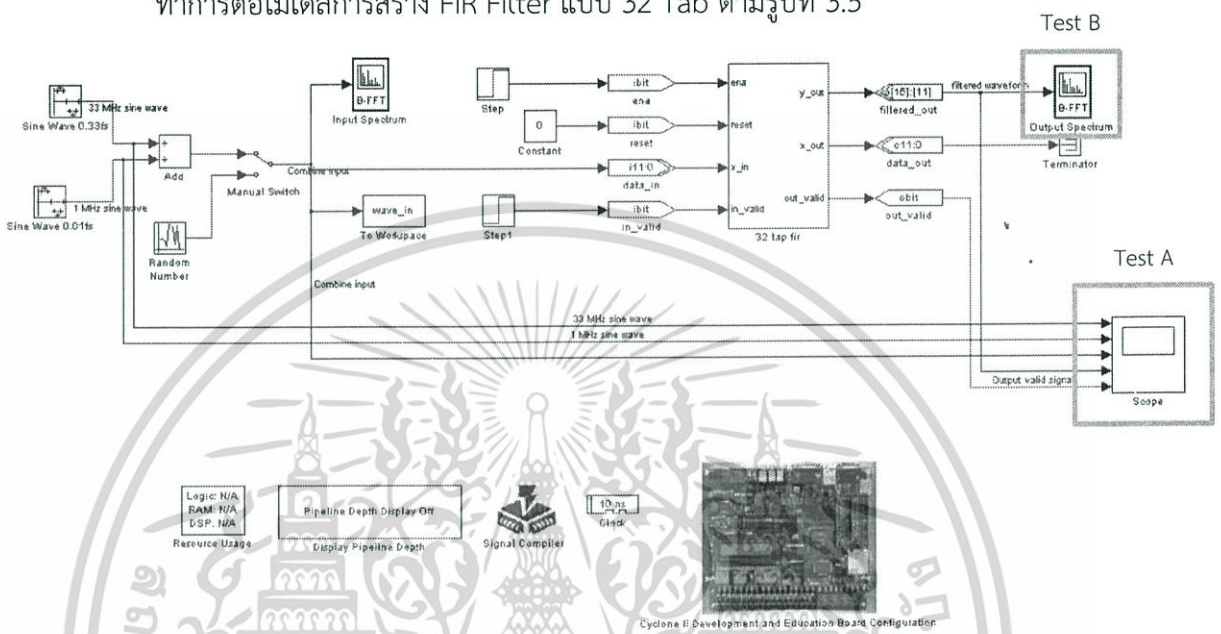
รูปที่ 3.4 การเชื่อมต่อ FIR Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การทดลองออกแบบการสร้าง FIR Filter แบบ 32 Tab ผ่านโปรแกรม

MATLAB

ทำการต่อโมเดลการสร้าง FIR Filter แบบ 32 Tab ตามรูปที่ 3.5



รูปที่ 3.5 การเชื่อมต่อ FIR Filter แบบ 32 Tab

ตารางที่ 3.2 ตารางการตั้งค่า Sine Wave Blocks

Sine Wave 0.33fs

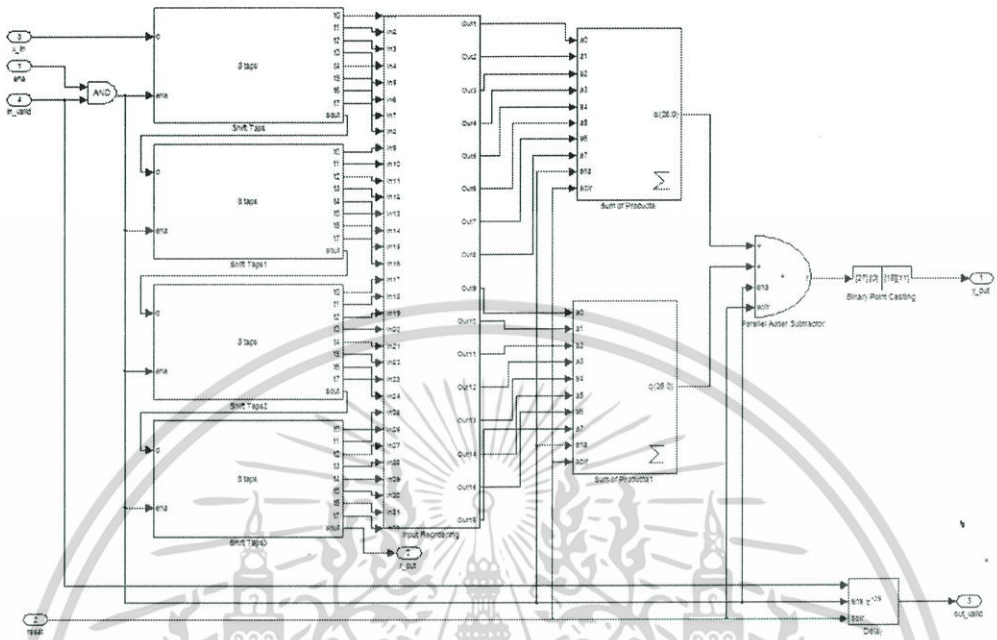
Sine type	Sample based
Amplitude	2^9–1
Samples per period	3
Sample time	sample time

Sine Wave 0.01fs

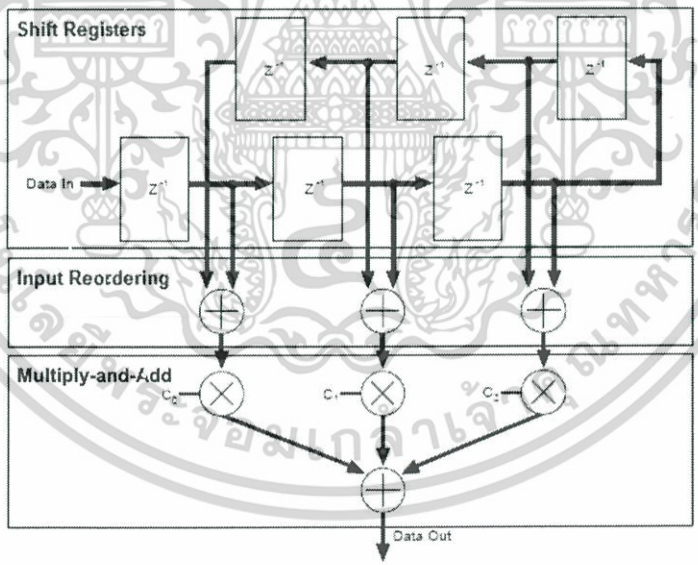
Sine type	Sample based
Amplitude	2^9–1
Samples per period	100
Sample time	sample time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในบล็อก 32 Tab FIR จะสร้างจากไดอะแกรมตามรูปที่ 3.6



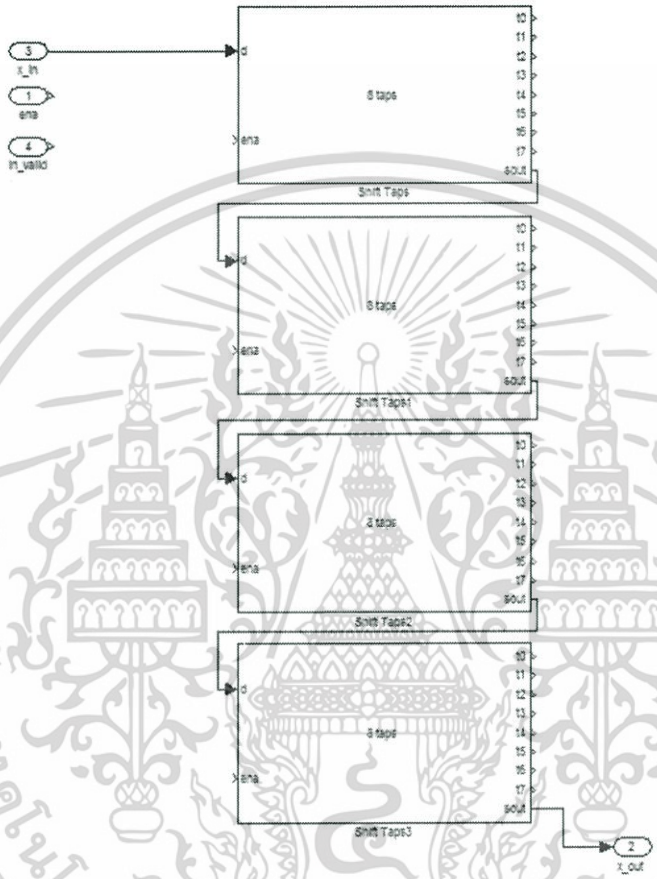
รูปที่ 3.6 การเชื่อมต่อภายใน 32 Tab FIR



รูปที่ 3.7 **6-Taps FIR Filter**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในการทำ FIR Filter นั้นจะแบ่งการทำงานออกเป็น 3 ขั้นตอนด้วยกันคือ Shift Taps, Sum of Products และ Parallel Adder Subtractor โดย Shift Taps สามารถสร้างได้ดังรูปที่ 3.8



รูปที่ 3.8 การเชื่อมต่อ Shift Taps Block

Connect the Shift Taps blocks together in the following manner:

เชื่อมอินพุต x_IN เข้าที่ d port ของ Shift Taps1

เชื่อม S out ของ Shift Taps1 เข้าที่ d port ของ Shift Taps2

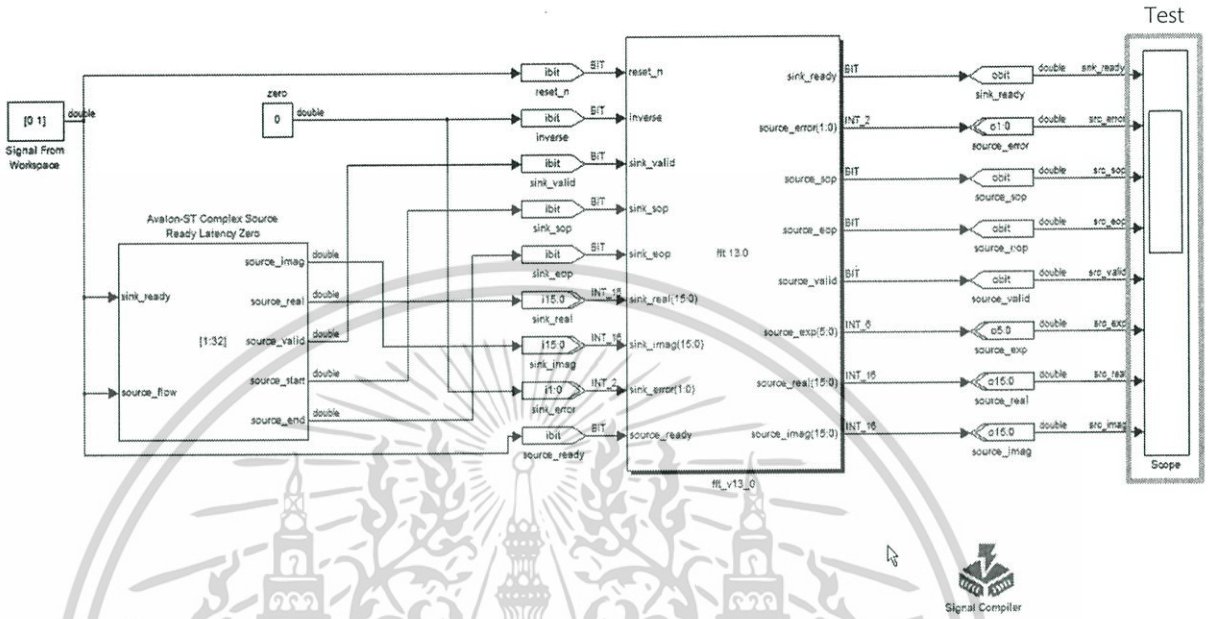
เชื่อม S out ของ Shift Taps2 เข้าที่ d port ของ Shift Taps3

เชื่อม S out ของ Shift Taps3 เข้าที่ d port ของ Shift Taps4

เชื่อม S out ของ Shift Taps4 เข้าที่ x_out

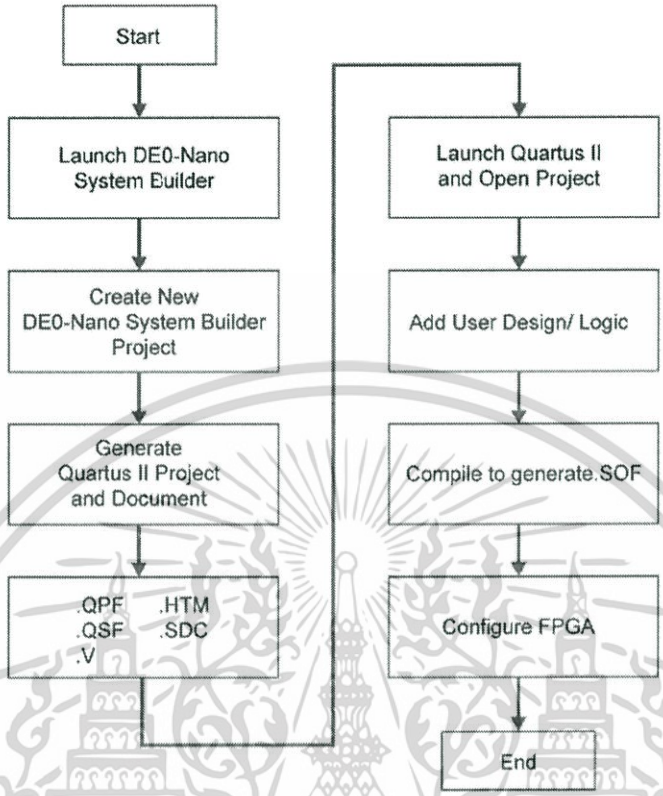
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 การทดลองออกแบบการสร้าง FFT ผ่านโปรแกรม MATLAB



รูปที่ 3.9 การออกแบบการสร้าง FFT ผ่านโปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 บล็อคไดอะแกรมการออกแบบการทำงานของ FPGA

3.2 Simulate the Design แบ่งออกเป็น 3 ขั้นตอน

Simulate in the MATLAB Software

Simulate with IP Functional Simulation Models

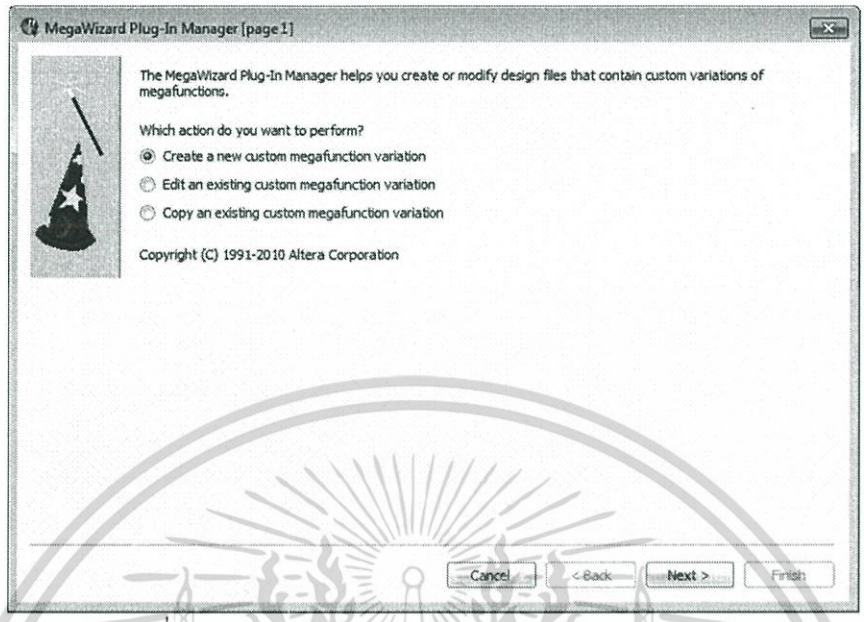
Simulating in Third-Party Simulation Tools Using NativeLink

3.2.1 การ Config FFT Model

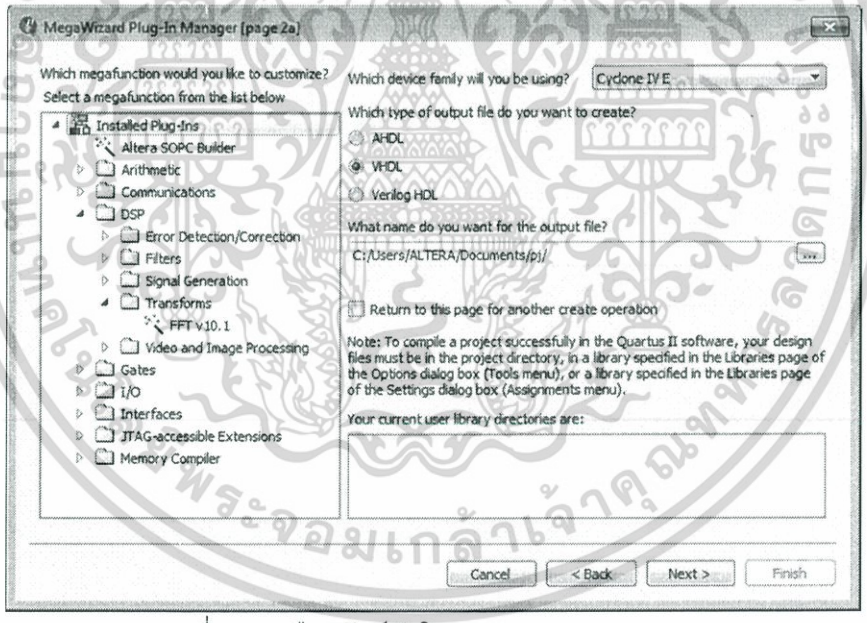
ขั้นตอนการติดตั้ง

1. เลือก ฟังก์ชัน และ ตั้งชื่อโปรแกรม
2. เลือกภาษา VHDL ในการบันทึกโปรแกรม
3. เลือก Target Device Family CYCLONE IV E EP4CE22F17C6N
4. ตั้งค่า Parameter
5. สร้างฟังก์ชันใน IP Toolbench

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

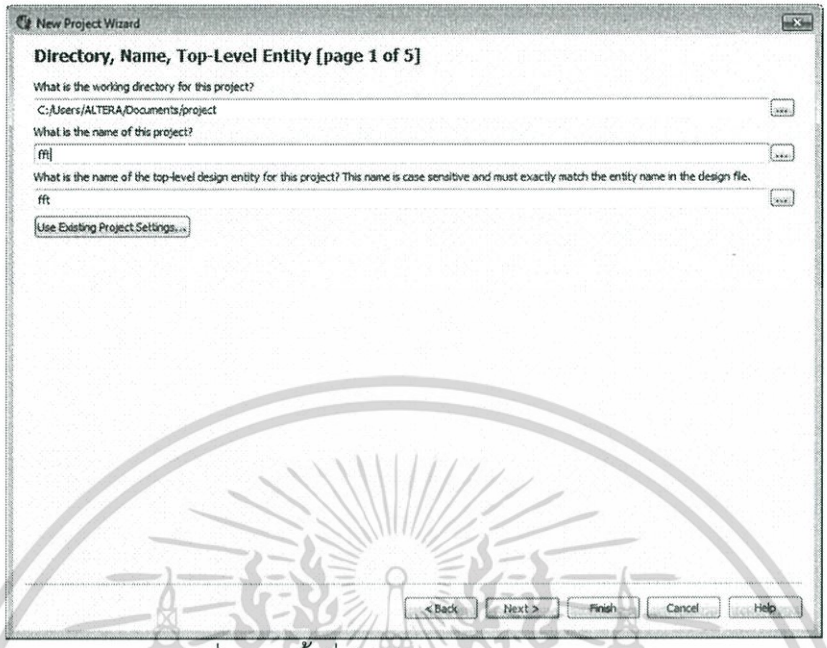


รูปที่ 3.11 การสร้างโฟลเดอร์สำหรับบันทึกผลการทดลอง

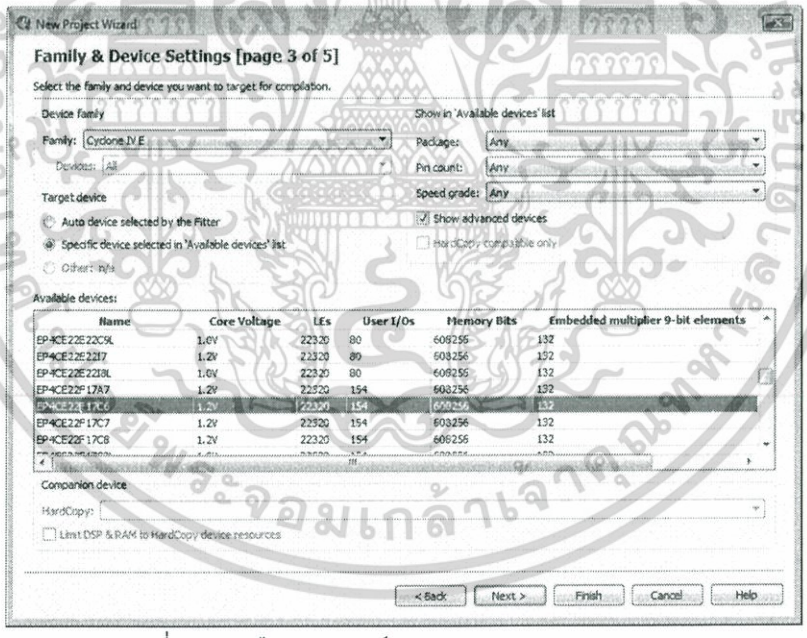


รูปที่ 3.12 เลือกฟังก์ชันใน Mega core function

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



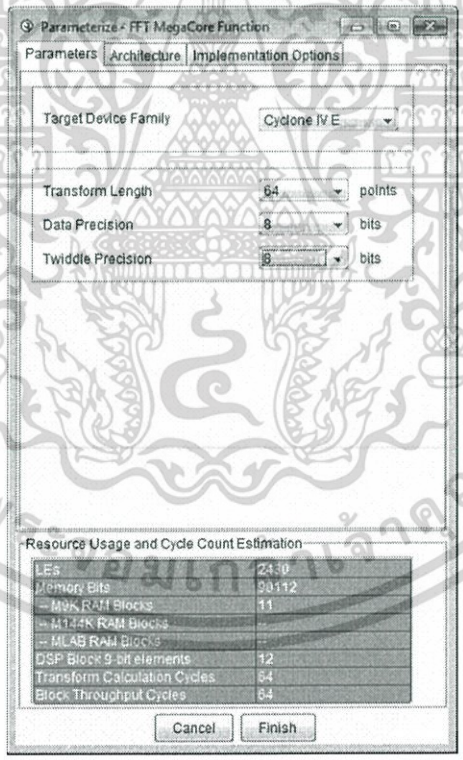
รูปที่ 3.13 ตั้งชื่อไฟล์ Mega core function



รูปที่ 3.14 เลือกอุปกรณ์ FPGA EP4CE22F17C6N

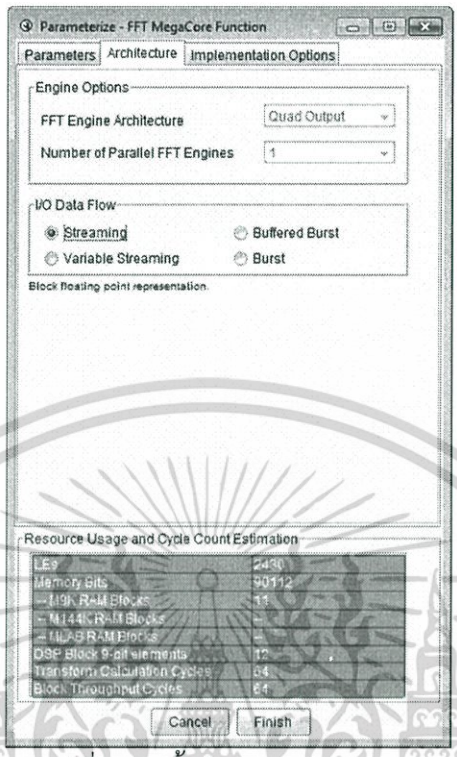
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.15 ตั้งค่า Parameterize ใน IP Toolbench

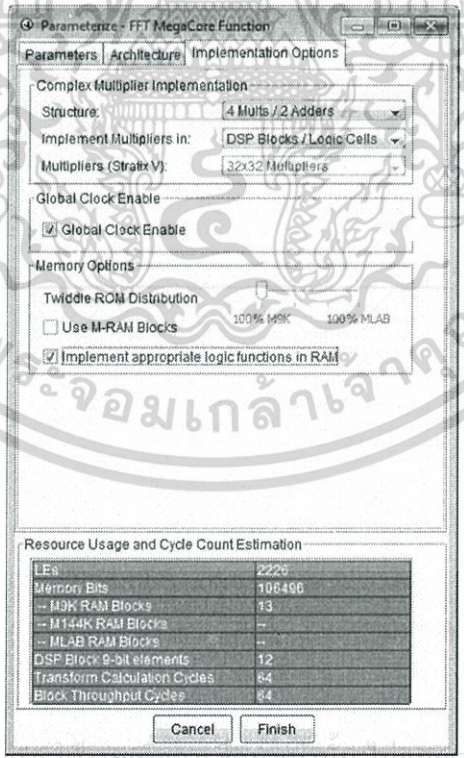


รูปที่ 3.16 ตั้งค่า Transform Length, Data Precision, Twiddle Precision

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

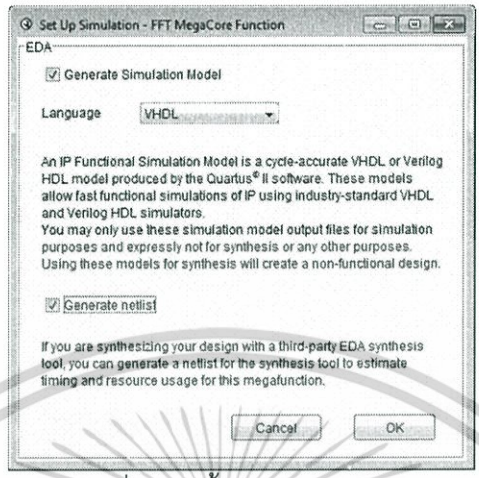


รูปที่ 3.17 ตั้งค่า I/O Data Flow

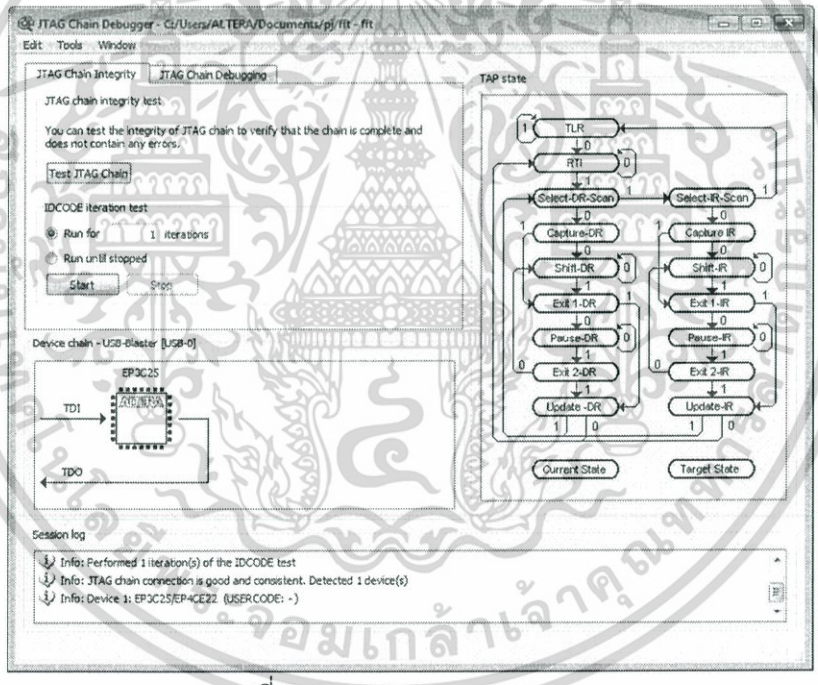


รูปที่ 3.18 ตั้งค่า Implementation Option

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

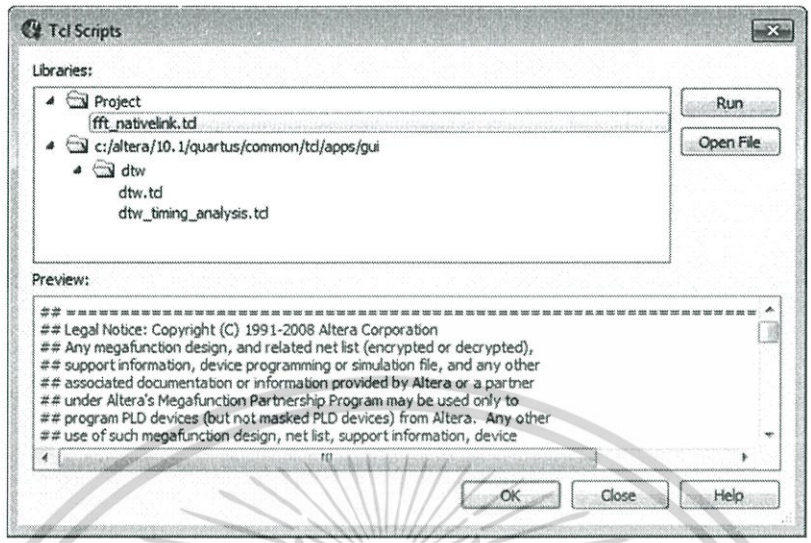


รูปที่ 3.19 ตั้งค่า Simulation

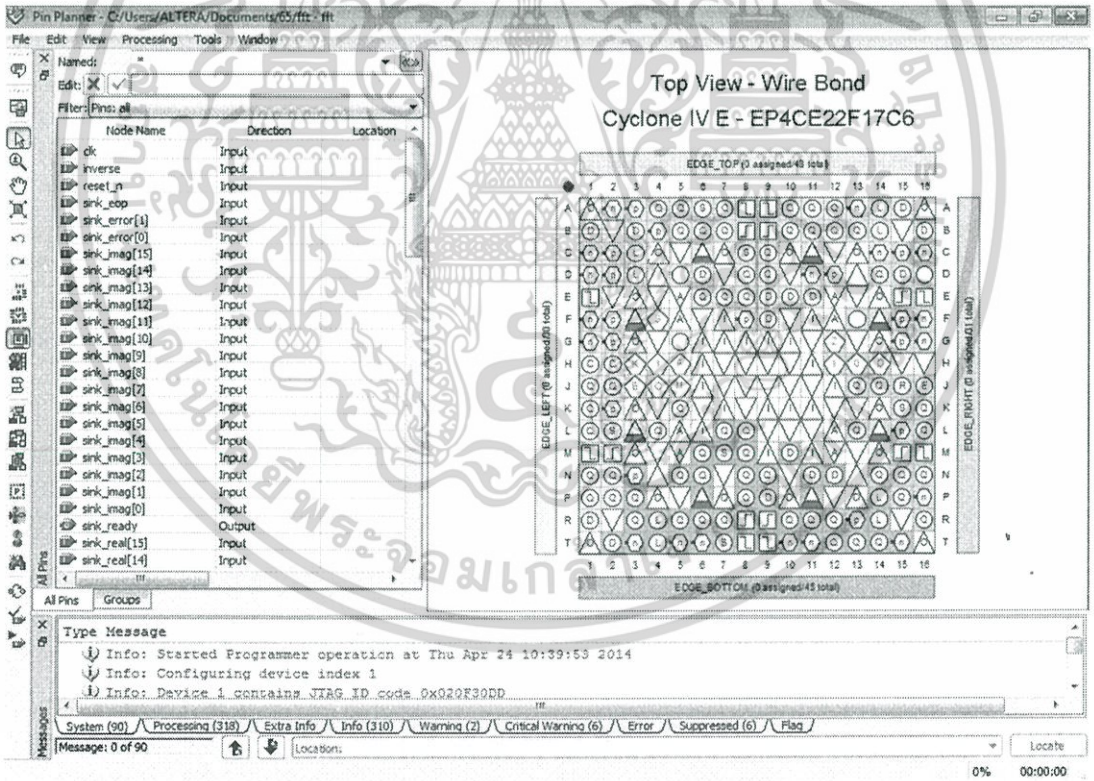


รูปที่ 3.20 JTAG Test Benches

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

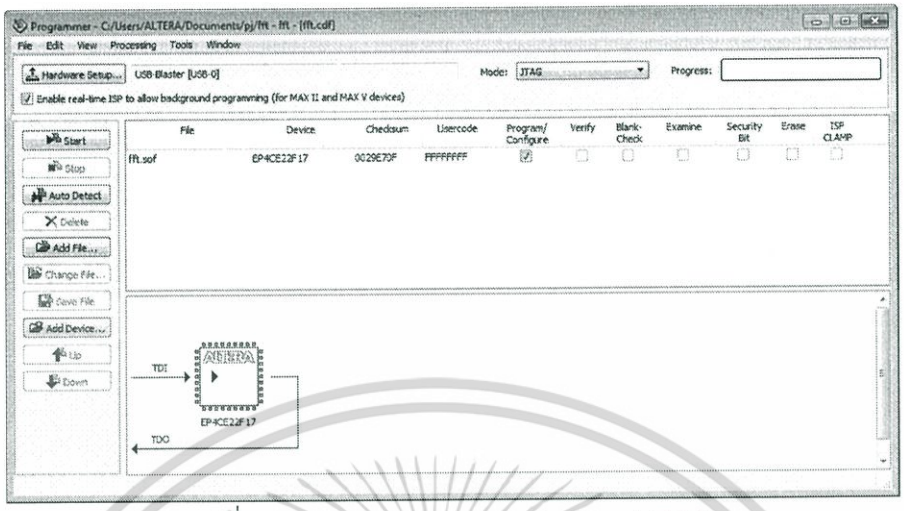


รูปที่ 3.21 การ Simulating in Third-Party Simulation Tools โดยใช้ NativeLink



รูปที่ 3.22 Pin Planner

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 การ Programmer ลง Board FPGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

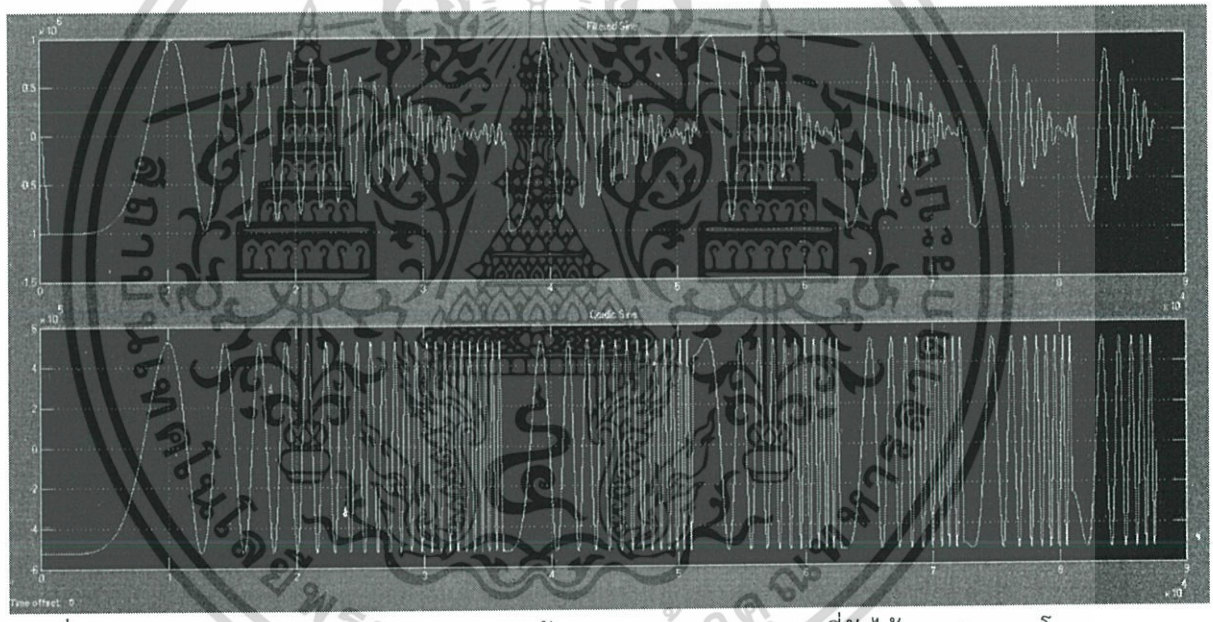
บทที่ 4
ผลการทดลอง

ในบทนี้จะกล่าวถึงการดำเนินงานโดยประมวลความรู้ที่ศึกษามาทั้งหมดมาประกอบกัน โดยผลการทดลองจะกล่าวถึงผลที่ได้จากการทดลองตามการทดลองในบทที่ 3 โดยค่าที่ได้จะออกมาในรูปแบบของกราฟทั้งหมด

4.1 ผลการสร้างการจำลองการสร้าง Frequency Sweep ผ่านโปรแกรม MATLAB จากรูปที่ 3.2

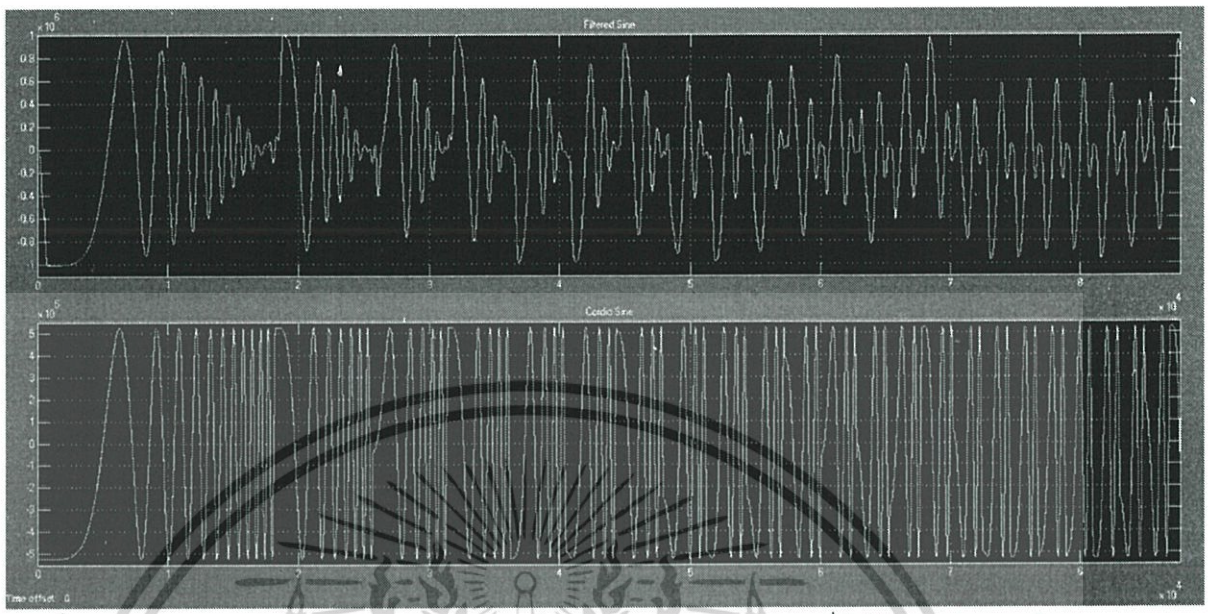
วัตถุประสงค์ของการทดลอง

1. ทดสอบการสร้าง Frequency Sweep
2. ตรวจสอบผลของข้อมูลที่ได้รับจากการ Simulink
3. สามารถนำข้อมูลที่ได้รับไปใช้งานได้ถูกต้อง

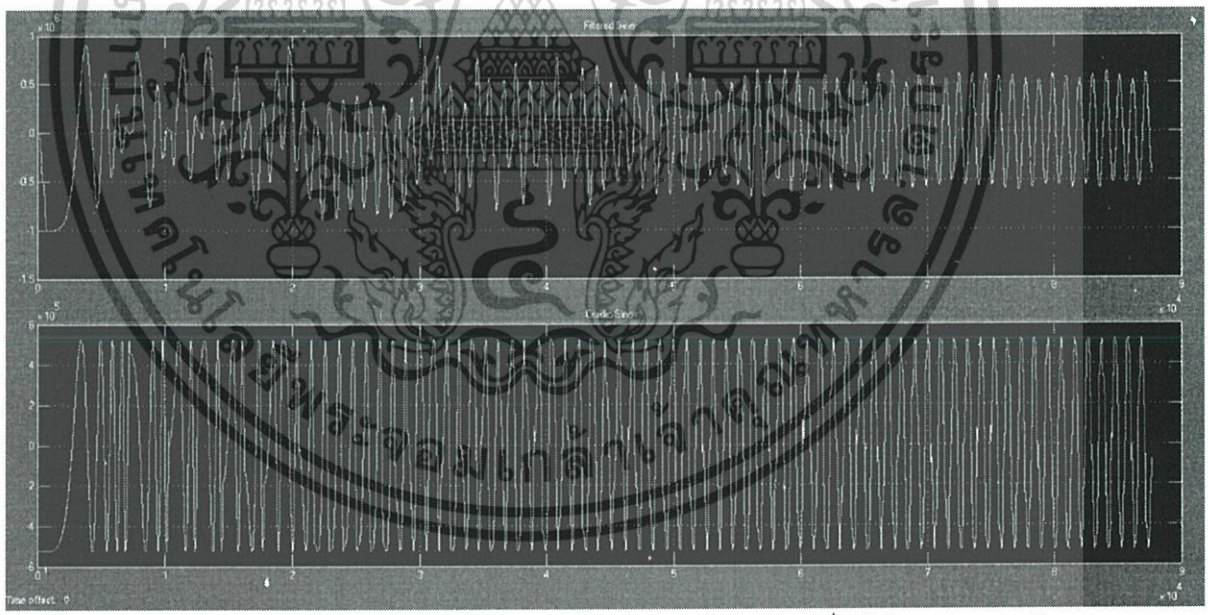


รูปที่ 4.1 ผลการตรวจสอบการทำงานของการสร้าง Frequency Sweep ที่วัดได้จาก Scope โดยปรับค่าความชันใน Ramp เท่ากับ 0.01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

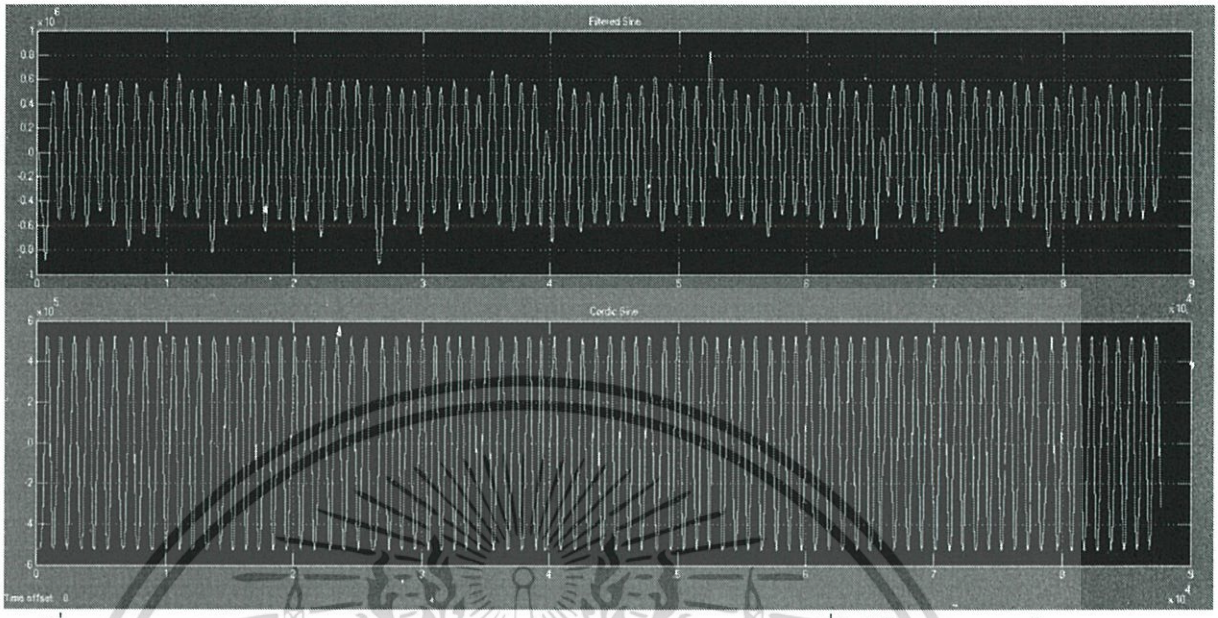


รูปที่ 4.2 ผลการตรวจสอบการทำงานของการสร้าง Frequency Sweep ที่วัดได้จาก Scope โดยปรับค่าความชันใน Ramp เท่ากับ 0.02

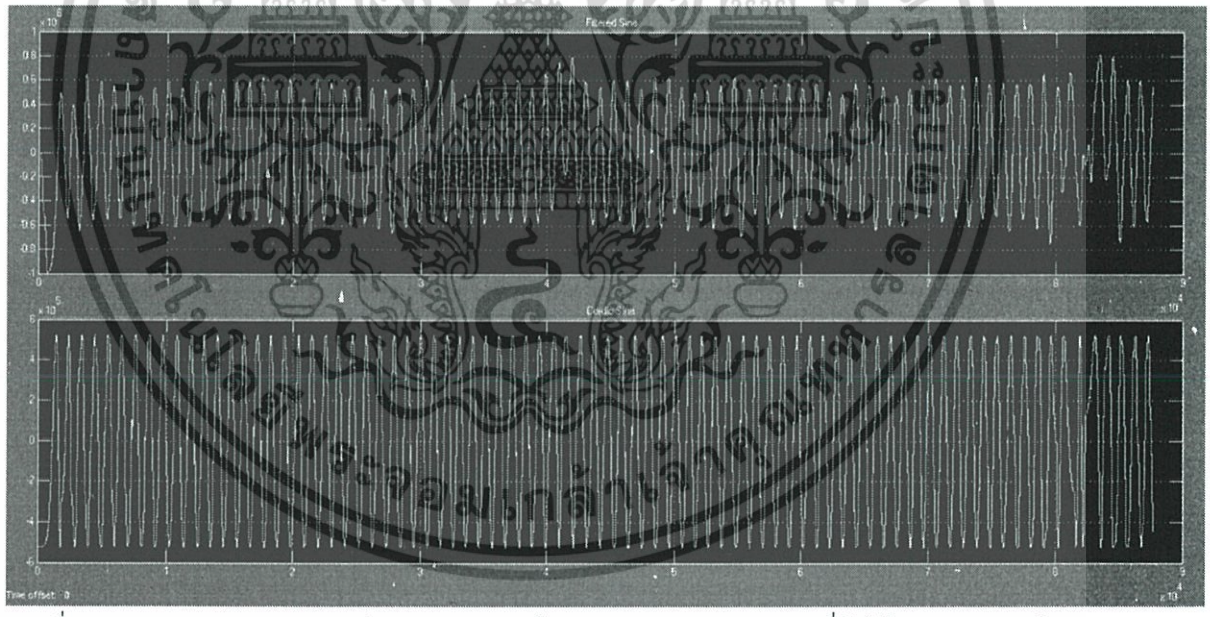


รูปที่ 4.3 ผลการตรวจสอบการทำงานของการสร้าง Frequency Sweep ที่วัดได้จาก Scope โดยปรับค่าความชันใน Ramp เท่ากับ 0.05

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

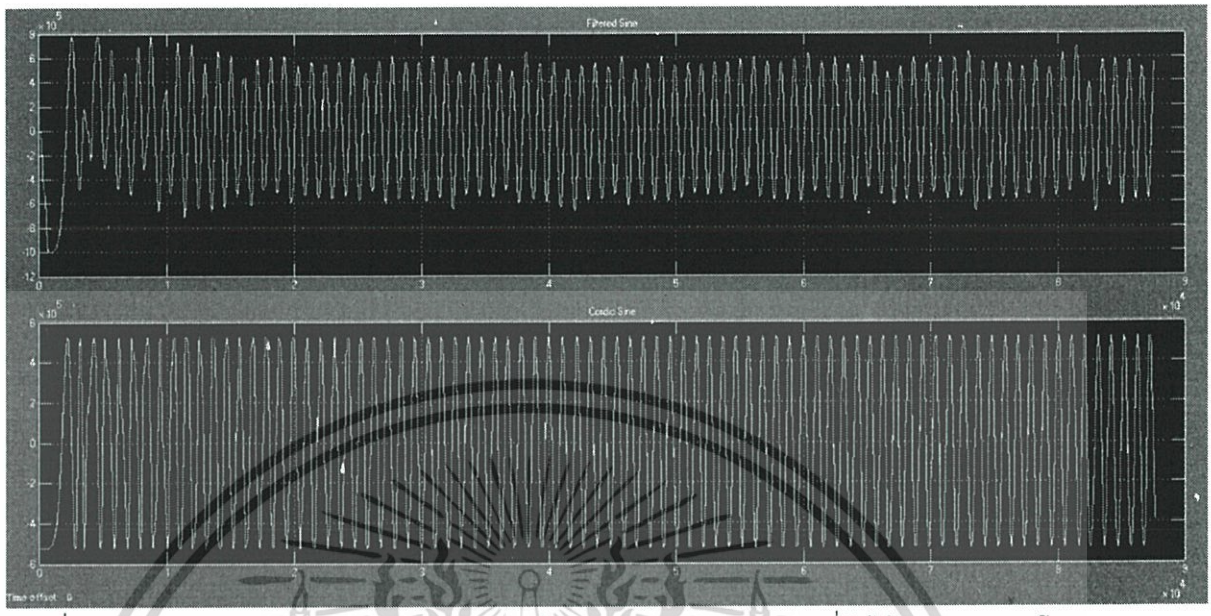


รูปที่ 4.4 ผลการตรวจสอบการทำงานของการสร้าง Frequency Sweep ที่วัดได้จาก Scope โดยปรับค่าความชันใน Ramp เท่ากับ 0.5

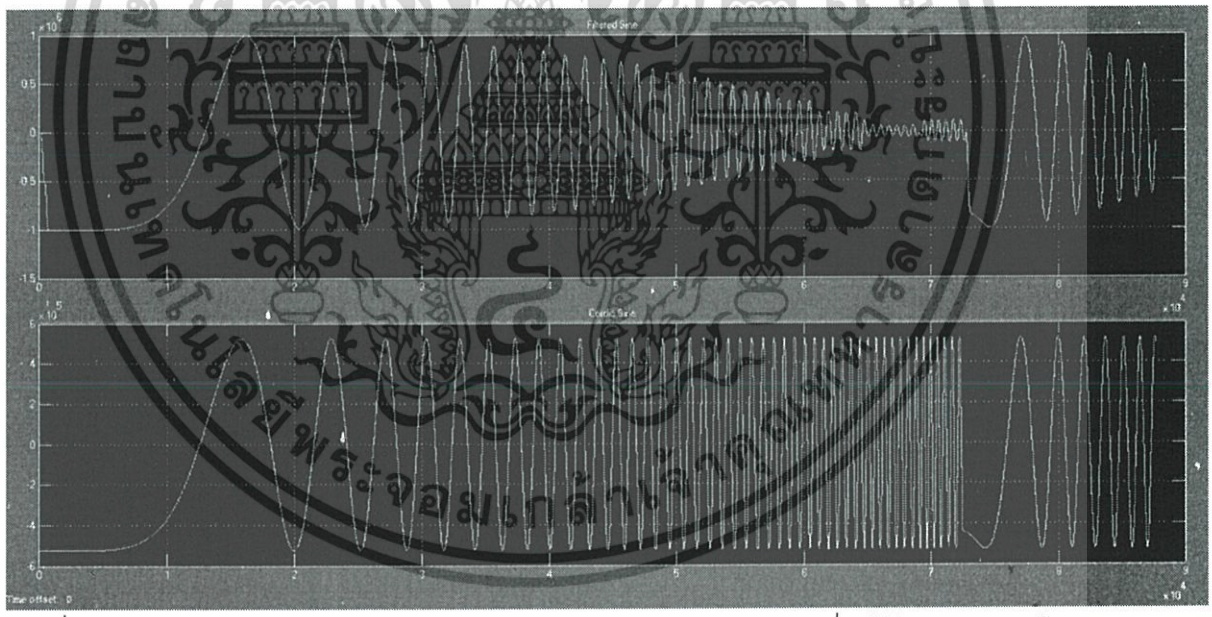


รูปที่ 4.5 ผลการตรวจสอบการทำงานของการสร้าง Frequency Sweep ที่วัดได้จาก Scope โดยปรับค่าความชันในRamp เท่ากับ 0.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

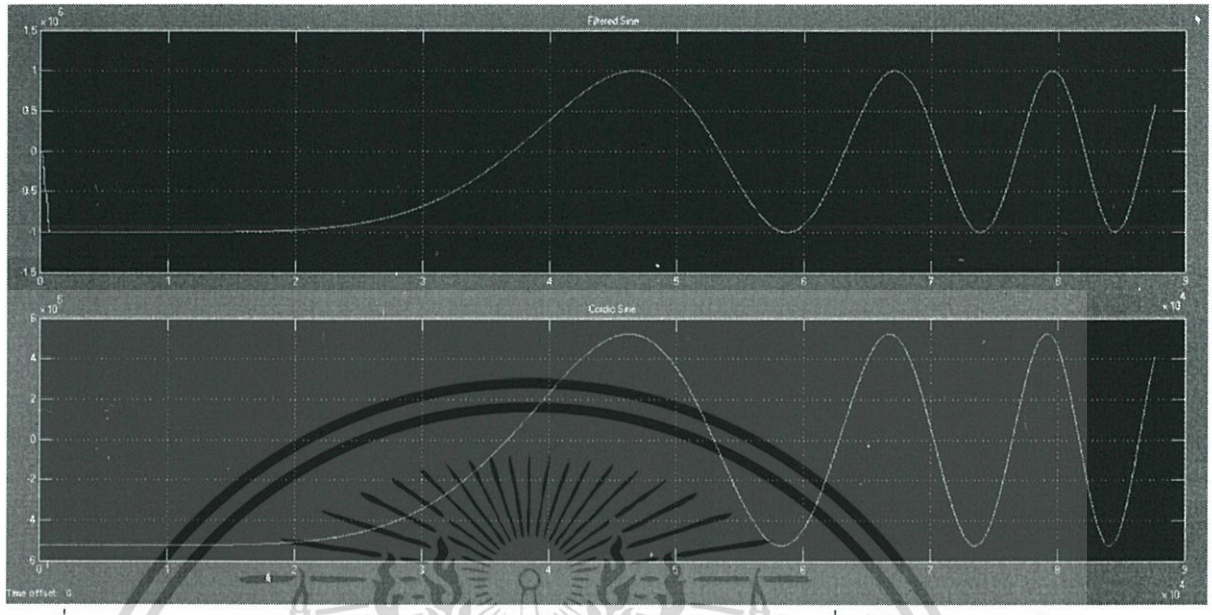


รูปที่ 4.6 ผลการตรวจสอบการทำงานของการสร้าง Frequency Sweep ที่วัดได้จาก Scope โดยปรับค่าความชันใน Ramp เท่ากับ 0.1

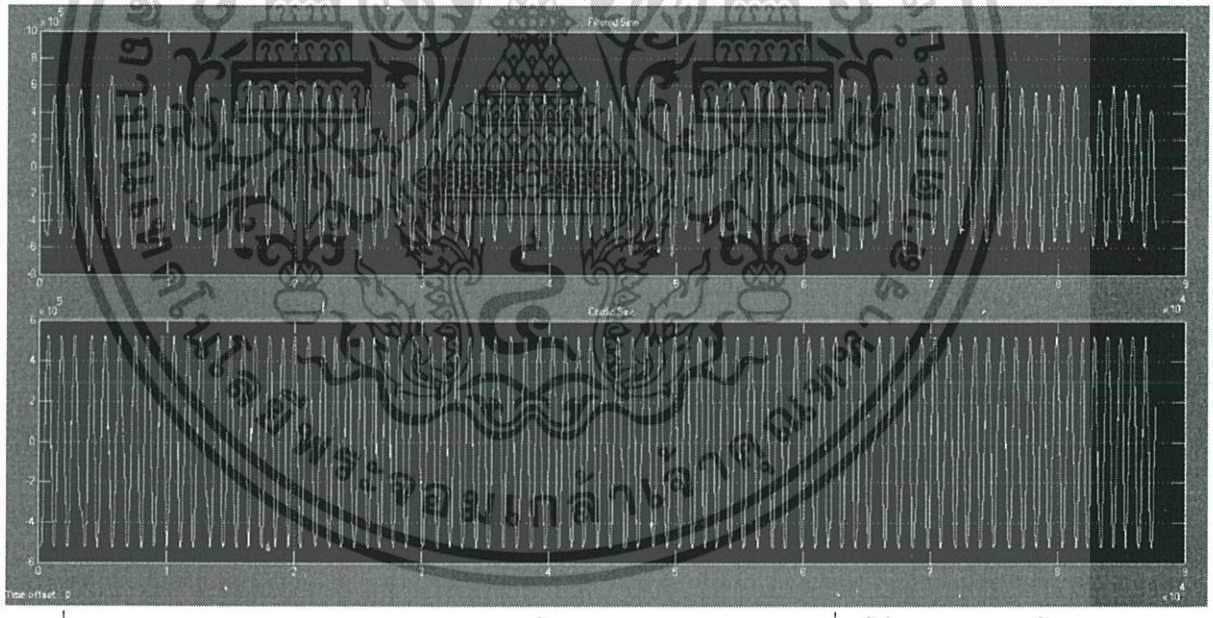


รูปที่ 4.7 ผลการตรวจสอบการทำงานของการสร้าง Frequency Sweep ที่วัดได้จาก Scope โดยปรับค่าความชันในRamp เท่ากับ 0.005

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

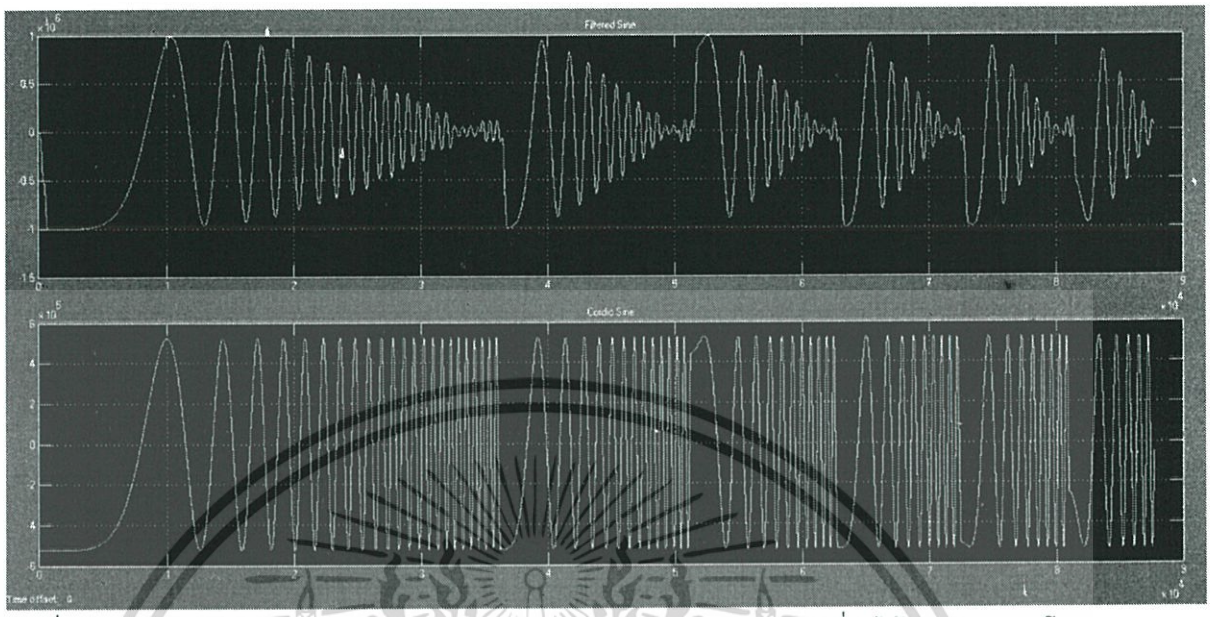


รูปที่ 4.8 ผลการตรวจสอบการทำงานของการสร้าง Frequency Sweep ที่วัดได้จาก Scope โดย ปรับค่าความชันในRamp เท่ากับ 0.001



รูปที่ 4.9 ผลการตรวจสอบการทำงานของการสร้าง Frequency Sweep ที่วัดได้จาก Scope โดย ปรับค่าความชันในRamp เท่ากับ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 ผลการตรวจสอบการทำงานของการสร้าง Frequency Sweep ที่วัดได้จาก Scope โดยปรับค่าความชันในRamp เท่ากับ -0.01

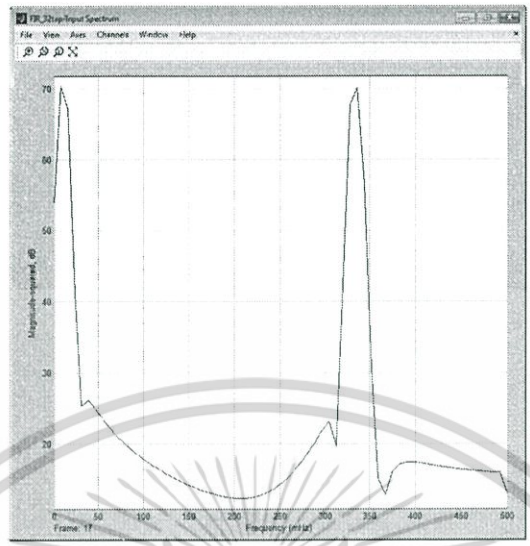
ซึ่งจากผลการทดลองพบว่า ยิ่งปรับค่าความชันใน Ramp เพิ่มขึ้น กราฟ Filtered Sine ก็จะมีความชิดกันของยอดคลื่นมากขึ้น ในขณะเดียวกัน Cordic Sine ก็มีความถี่ที่เพิ่มขึ้นเช่นกัน การทำเช่นนี้ เพื่อเป็นการกำหนดขนาด บอกจุดเริ่มต้นและจุดสิ้นสุดของข้อมูล หรือสัญญาณ ที่จะนำไปใช้ในการคำนวณ FFT เพื่อจะได้ออกแบบการคำนวณให้เหมาะสมกับข้อมูลได้ คล้ายกับการแบ่งข้อมูลออกเป็นห้วงๆนั้นเอง

4.2 ผลการทดลองออกแบบการสร้าง FIR Filter ผ่านโปรแกรม MATLAB ตามรูปที่ 3.4

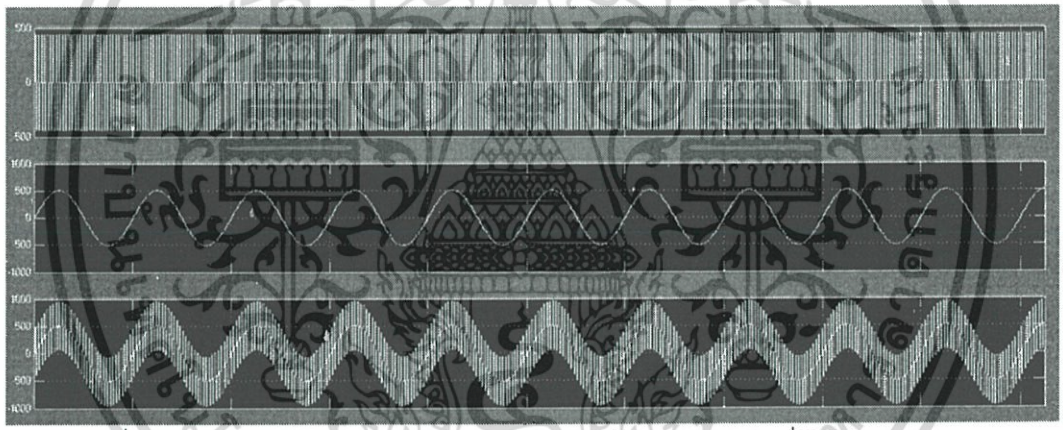
วัตถุประสงค์ของการทดลอง

1. เพื่อทดสอบการสร้าง FIR Filter ผ่านโปรแกรม MATLAB
2. เพื่อทดสอบการทำงานของ FIR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 สเปกตรัมของอินพุทของโมเดลที่ทำการทดลอง จากจุด Test A



รูปที่ 4.12 ผลการตรวจสอบการทำงานของการสร้าง FIR Filter ที่วัดได้จาก Scope จากจุด Test B เมื่อสเปกตรัมของอินพุทเป็นดังรูปที่ 4.11

จากผลการทดลองออกแบบสร้าง FIR Filter พบว่า เมื่อนำสัญญาณอินพุทที่มีสเปกตรัมดังรูปที่ 4.11 มาผ่าน FIR Filter ที่ออกแบบจะพบว่า ผลที่ได้จะออกมาดังรูปที่ 4.12 โดยหน้าที่ของ FIR Filter คือการนำสัญญาณอินพุตที่ได้รับมาผ่านระบบโดย ระบบจะนำแค่อินพุตมาประมวลผล ไม่มีการทำ loop back ของสัญญาณเอาท์พุต ดังการทดลองนี้คือนำสัญญาณอินพุตมาผ่านการคูณด้วยสัญญาณ Sine โดยการคูณแบบจุดต่อจุด ไม่มีการนำเอาผลของเอาท์พุตมารวม

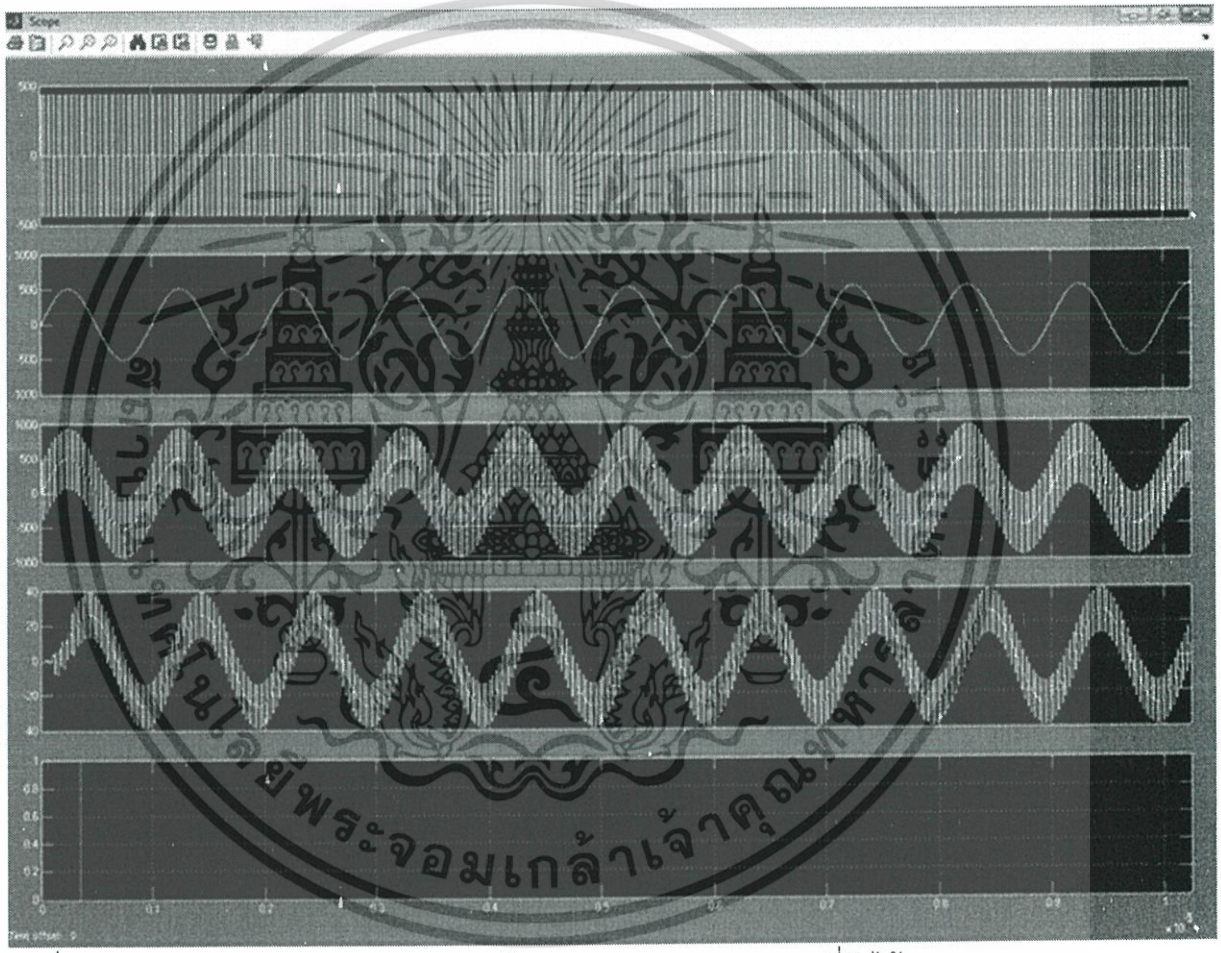
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดลองออกแบบการสร้าง FIR Filter แบบ 32 Tab ผ่านโปรแกรม MATLAB ตามรูปที่ 3.5

วัตถุประสงค์ของการทดลอง

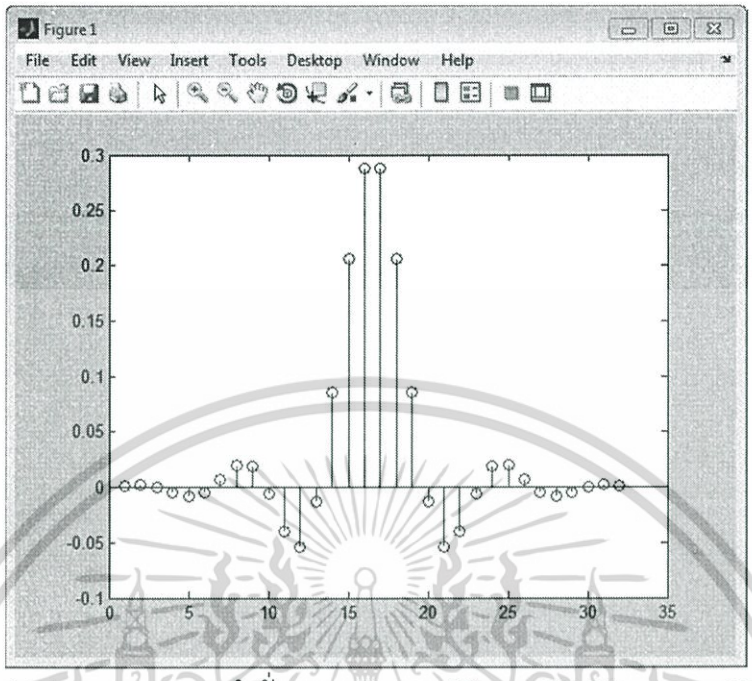
1. เพื่อทดสอบการออกแบบการสร้าง FIR Filter แบบ 32 Tab ผ่านโปรแกรม MATLAB

2. เพื่อดูการทำงานการทำ FFT ในการสร้าง FIR Filter แบบ 32 Tab

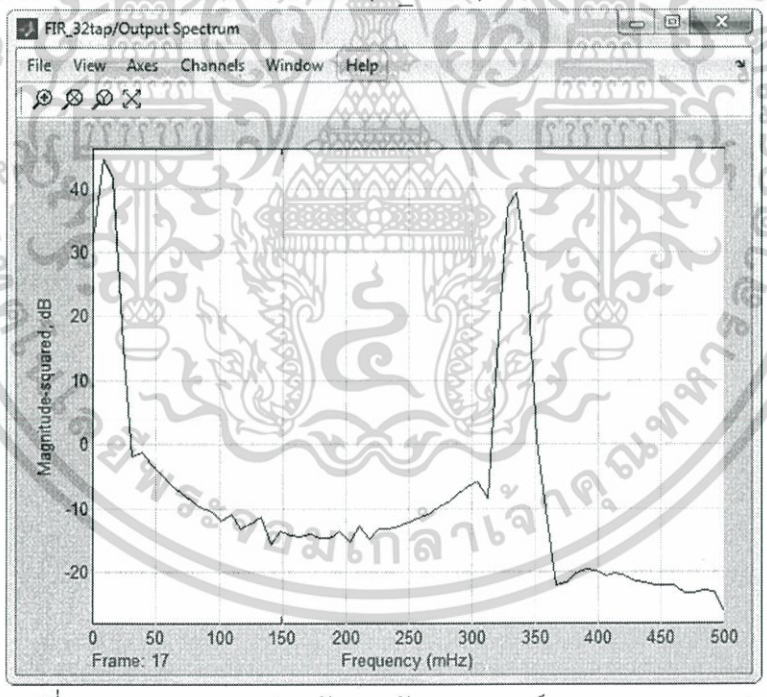


รูปที่ 4.13 ผลการทดสอบการออกแบบการสร้าง FIR Filter แบบ 32 Tab ที่วัดได้จาก Scope จากจุด Test A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



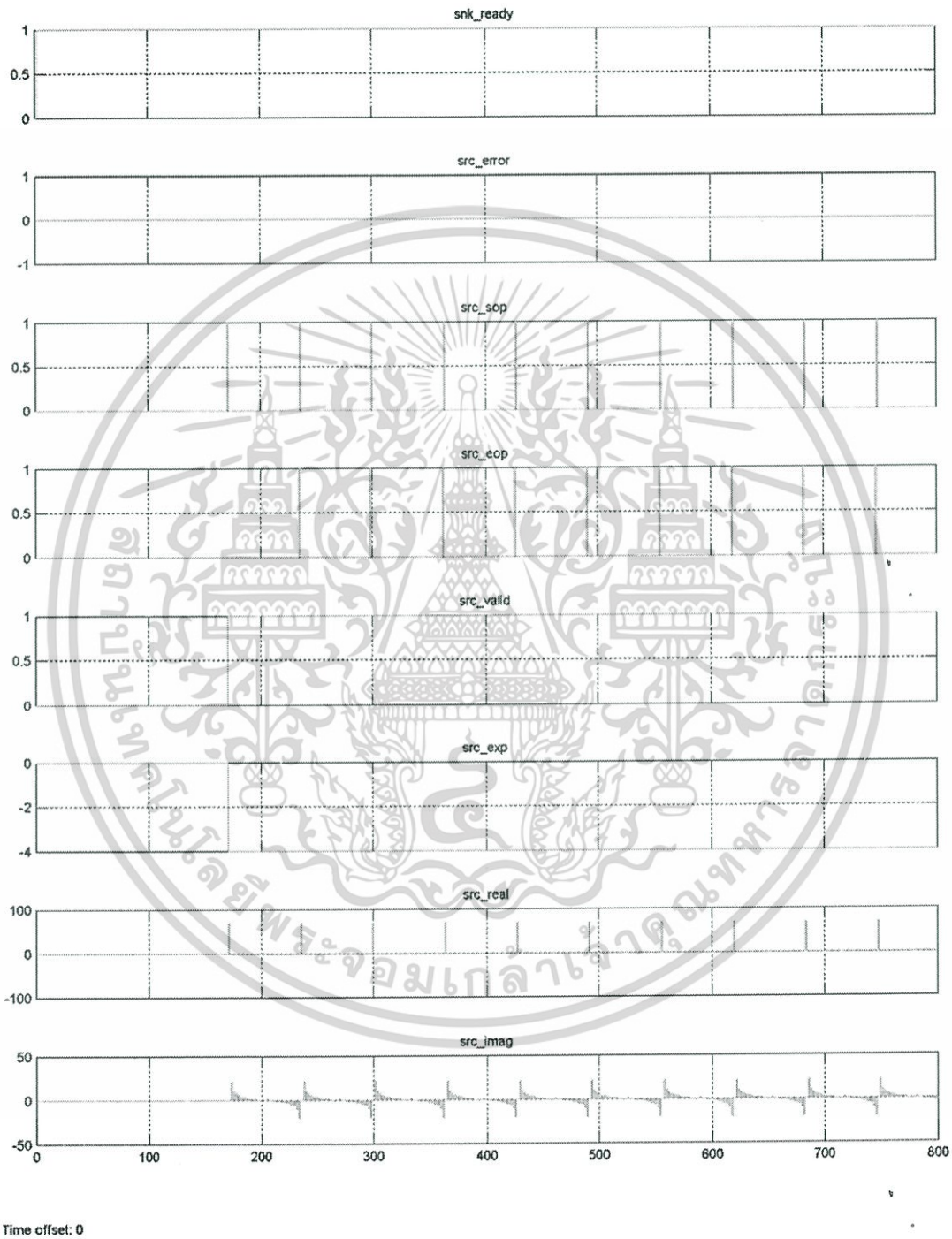
รูปที่ 4.14 ผลการทดสอบคำสั่ง stem(fir_coeff) ของ FIR Filter แบบ 32 Tab



รูปที่ 4.15 ผลการหาสเปกตรัมของสัญญาณเอาท์พุท จากจุด Test B

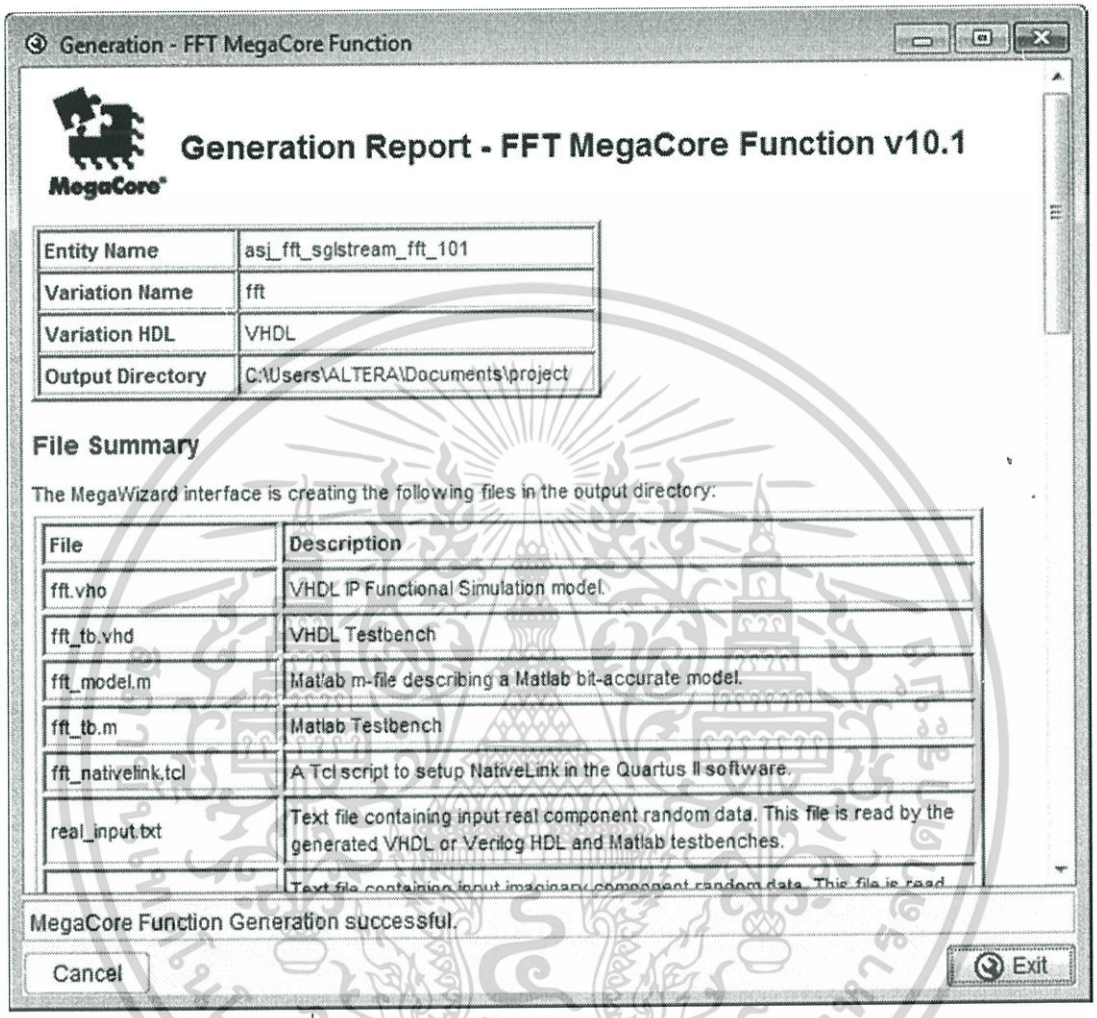
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการทดลองออกแบบการสร้าง FFT ผ่านโปรแกรม MATLAB ตามรูปที่ 3.9

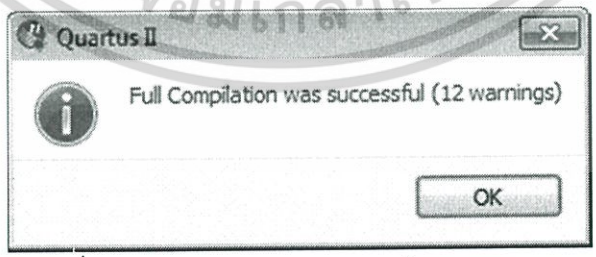


รูปที่ 4.16 ผลการทดลองผ่าน FFT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 ผลการสร้าง FFT Mega Core Function



รูปที่ 4.18 ผลการ Compilation ใน Quartus II

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.19 ผลการ Compilation ใน Quartus II

รูปที่ 4.20 ผลการ Programmer ลงบอร์ด FPGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองออกแบบการสร้าง FFT Mega Core Function ผ่านโปรแกรม MATLAB โดยสร้างสัญญาณที่รู้ค่าความถี่ขึ้นมา แล้วนำสัญญาณเข้าสู่ระบบ FFT ที่ออกแบบตามบทที่ 3 จะได้สัญญาณเอาท์พุตดังรูปที่ 4.16 หลังจากนั้นจึงทำการ Compilation ในโปรแกรม Quartus II เพื่อทดสอบโปรแกรมลง บอร์ด FPGA ซึ่งจากรูปที่ 4.17-4.20 พบว่า การ Compilation ประสบผลสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการทดลองโครงงาน

โครงงานนี้เป็นการทดลองสร้างสัญญาณ Frequency Sweep เพื่อทำการทดสอบการทำงานของ FPGA ร่วมกับ โปรแกรม MATLAB ซึ่งเมื่อทดสอบการทำงานร่วมกันได้แล้วจึงจะทำการทดลองสร้าง FIR Filter เพื่อหาความสัมพันธ์ในกระบวนการสร้าง การแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform: FFT) ซึ่งเมื่อทดสอบการทำงานรวมกันของโปรแกรมรวมถึงศึกษาการทำงานของการสร้าง FFT แล้วจึงทำการทดลองสร้างการทำ FFT จาก **FFT Mega Core Function** ตามรูปที่ 3.9 จึงได้ผลดังรูปที่ 4.16

5.2 ข้อเสนอแนะ

1. ในการทดลองจำเป็นต้องใช้ บอร์ด FPGA ควรตรวจเช็คการทำงานของบอร์ด FPGA ให้พร้อมใช้งานอยู่เสมอ
2. ข้อมูลที่ทำการทดลองยังทำได้ไม่ครบถ้วน เนื่องจากติด error ของ software บางประการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] ผศ.ดร.ปริญญา สงวนสัตย์. คู่มือการใช้ Matlab ฉบับสมบูรณ์. พิมพ์ครั้งที่ 1. นนทบุรี : ไอดีซี , 2556

[2] Vinay K. ingle and John G. Proakis. Digital Signal Processing Using MATLAB. United State. : Thomson, 2007

[3] K.R. Rao, D.N. Kim and J.J. Hwang. Fast Fourier Transform: Algorithms And Applications. London. : Springer, 2010

[4] Alan V. Oppenheim. The Fast Fourier Transform And Its Applications. London. : Springer, 1988

[5] Douglas F. Elliott and K. Ramamohan Rao. Fast Transform: Algorithms, Analyses, Applications. London. : Academic Press, 1982

[6] Vinay K. ingle and John G. Proakis. Digital Signal Processing Using MATLAB. United State. : Thomson, 2007

[7] Sanjit K. Mitra. Digital signal processing laboratory using MATLAB. Boston : WCB/McGraw-Hill, 2000

[8] Altera Corporation, "DSP Builder User Guide", version 9.0, 2009

[9] Altera Corporation, "FFT MegaCore Function User Guide", version 13.0, 2013

[10] Altera Corporation, "Nios II Processor Reference Handbook", version 13.1, 2014

[11] Quartus II and EDA Tools Discussion [ออนไลน์] เข้าถึงได้จาก : http://www.alteraforum.com/forum/forumdisplay.php?f=7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้