



การประยุกต์ใช้งานการประมวลผลสัญญาณเชิงเลข  
( DIGITAL SIGNAL PROCESSING APPLICATION )



บริบทของงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง

632542

ปริญญาโทสำหรับภาคการศึกษาที่ 2 ปีการศึกษา 2535  
ภาควิชาวิศวกรรมอิเล็กทรอนิกส์  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง


เรื่อง การประยุกต์ใช้งานการประมวลผลสัญญาณดิจิทัล  
( Digital Signal Processing Application )

ผู้จัดทำ

น.ส.กนกวรรณ แซ่เอ็ง	321001
น.ส.ธิดาพร พิทักษ์พรพันธุ์	321122
น.ส.เอื้อมพร จารุกิตติยพันธ์	321430

อาจารย์ที่ปรึกษา

รศ.ดร.มนัส สังวรศิลป์

(..........)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง

032542

## การประมวลผลสัญญาณเชิงเลข

( Digital Signal Processing : DSP )

นางสาว กนกวรรณ แซ่เซ็ง	321001
นางสาว ธิดาพร พิทักษ์พรพันธุ์	321122
นางสาว เอื้อมพร จารุกิตติยนต์	321430

รต.ดร. มนัส สังวรศิลป์ อาจารย์ที่ปรึกษา  
ภาคการศึกษาที่ 2 ปีการศึกษา 2535

### บทคัดย่อ

ปัจจุบันนี้ ได้มีการศึกษาวิจัยทฤษฎีการประมวลผลสัญญาณเชิงเลข และนำไปประยุกต์ใช้งานอย่างกว้างขวาง เนื่องจากเทคโนโลยีในการผลิตไอซีสามารถสร้างหน่วยประมวลผลสัญญาณเชิงเลข บรรจุในชิปตัวเดียวได้ ทำให้สะดวกและต้นทุนในการผลิตลดลง

ปริญญาโทฉบับนี้มีจุดมุ่งหมายเพื่อศึกษาและนำทฤษฎีการประมวลผลสัญญาณเชิงเลขไปประยุกต์ใช้งาน โดยเริ่มต้นจากการศึกษาการทำงานของ Microcontroller TMS 32010 ของ บริษัท TEXAS INSTRUMENTS ที่ถูกออกแบบมาเพื่องานการประมวลผลสัญญาณเชิงเลขโดยเฉพาะแล้วนำมาสร้างเป็นชุดทดลองที่สามารถทำงานได้โดยลำพัง แต่มีการออกแบบให้เชื่อมต่อกับเครื่องไมโครคอมพิวเตอร์ในการถ่ายเทข้อมูลและโปรแกรมผ่าน RAM PACK เพื่อความสะดวกในการพัฒนาโปรแกรม จากนั้นเราจะนำชุดทดลองที่สร้างขึ้น มาประยุกต์ใช้ในการวิเคราะห์สัญญาณต่าง ๆ ในเชิงความถี่ โดยใช้ทฤษฎีการแปลงฟูเรียร์เติมหน่วย (DFT) และการแบ่งหรือจัดกลุ่มสมการทางคณิตศาสตร์ เพื่อลดเวลาและขั้นตอนในการคำนวณ วิธีการนี้เรียกว่า การแปลงฟาสต์ฟูเรียร์ (Fast Fourier Transform) และใช้ทฤษฎีการแปลงกลับของการแปลงฟาสต์ฟูเรียร์ที่เรียกว่าการแปลงฟาสต์ฟูเรียร์ผกผัน (Inverse Fast Fourier Transform) ในการเปลี่ยนสัญญาณเชิงความถี่ให้อยู่ในโดเมนเวลา ซึ่งเราสามารถประยุกต์ใช้ในการสังเคราะห์สัญญาณที่ประกอบด้วยความถี่ตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Digtal Signal Processing

KANOKWAN	SAEHENG	321001
THIDAPORN	PITAKPORN PAN	321122
AURMPORN	JARUKITTIYON	321430

### Abstract

Nowaday Digital Signal Processing Theorem has been reserched, developed , and widespread applied because the IC production technology can build a digital signal processor in a LSI single chip , that causes comfort and reduction of the cost of production.

This thesis has aimed for studying and applying Digtal Signal Processing Theorem. First , we had begun with studying the operation of the microcontroller TMS 32010 that was designed specific for digital signal processing. Then we made a stand-alone experiment set ,which was designed to interface with a microcomputer to transfer data and program with a rampack , so we can develop the program more comfortable. Later we applied the experiment set for analysis any signal in frequency domain by using Discrete Fourier Transform Theorem and grouping mathematics equation for decreasing time and stage in the process called Fast Fourier Transform , and used the inversion of this method called Inverse Fast Fourier Transform for transferring the signal in frequency domain to time domain. From this method , we can sythesis the various frequency signal.



# สารบัญ

หน้า

บทที่ 1	หลักการประมวลผลสัญญาณดิจิทัล	1
1.1	ทฤษฎีของการแปลงสัญญาณ	1
1.1.1	ทฤษฎีของการสุ่มข้อมูล	2
1.1.2	ทฤษฎีการควินาโตซ์	3
1.1.3	วงจรรวมเปิดแอนด์โฮลด์	5
1.1.4	วงจรมีการแปลงสัญญาณอนาล็อกเป็นดิจิทัล	7
1.1.5	วงจรมีการแปลงสัญญาณดิจิทัลเป็นอนาล็อก	9
1.2	ระบบสัญญาณเวลาไม่ต่อเนื่อง	10
1.2.1	ตัวประมวลผลสัญญาณ	10
1.2.2	สัญญาณเวลาไม่ต่อเนื่อง	12
1.2.3	การกระทำต่อสัญญาณเวลาไม่ต่อเนื่อง	17
1.2.4	ระบบประมวลผลเชิงเส้นและไม่เปลี่ยนแปลงตามเวลา	21
1.2.5	การกรองความถี่โดยใช้ตัวกรองดิจิทัล	23
1.2.6	การสุ่มข้อมูล	26
1.2.7	การสังเคราะห์ด้วยความละเอียดสูง	29
บทที่ 2	สถาปัตยกรรมของ TMS32010	33
2.1	คุณสมบัติ	34
2.2	สัญญาณเข้าและรายละเอียด	35
2.3	สถาปัตยกรรม	39
2.4	ฟังก์ชันบล็อกอะแดปต์	39
2.5	การจัดหน่วยความจำ	43
2.6	หน่วยประมวลผลทางคณิตศาสตร์	47
2.7	การควบคุมระบบ	47
2.8	ฟังก์ชันอินพุทและเอาต์พุท	49
2.9	การขออินเทอร์รัพท์	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า

บทที่ 3 การออกแบบวงจรและการทำงาน.....	52
3.1 ส่วนประมวลผล.....	52
3.1.1 ตัวประมวลผล.....	52
3.1.2 วงจรสร้างสัญญาณรีเซตระบบ.....	53
3.1.3 วงจรสร้างสัญญาณนาฬิกา.....	53
3.1.4 วงจรสร้างสัญญาณ ~BIO.....	54
3.1.5 การสร้างสัญญาณฮ่านเขียนพอร์ทและ หน่วยความจำ.....	54
3.1.6 วงจรกำหนดความถี่สุ่ม.....	56
3.2 ส่วนแปลงสัญญาณอนาลอกเป็นดิจิตอล.....	58
3.3 ส่วนแปลงสัญญาณดิจิตอลเป็นอนาลอก.....	61
3.4 ส่วนหน่วยความจำ.....	63
บทที่ 4 การแปลงฟูเรียร์ของสัญญาณ.....	66
4.1 หลักการและที่มาของ DFT.....	66
4.1.1 DIT FFT.....	66
4.1.2 DIT FFT Butterfly.....	72
4.1.3 การทำ Scaling.....	73
4.1.4 การทำ Bit-Reverse.....	75
4.1.5 การแปลงผกผันของ FFT.....	75
4.2 คุณสมบัติบางประการของ DFT.....	76
4.3 ความผิดพลาดจากการทำ FFT.....	77
บทที่ 5 ผลการทดลองและสรุปผลการทดลอง.....	79
- การทดลอง.....	79
- สรุปผลการทดลอง.....	104
ภาคผนวก ก โปรแกรมย่อยในการทำ FFT และ IFFT.....	105
ภาคผนวก ข การหาขนาดของจำนวนเชิงซ้อนโดยใช้ TMS32010.....	123
ภาคผนวก ค การติดต่อกับคอมพิวเตอร์.....	125

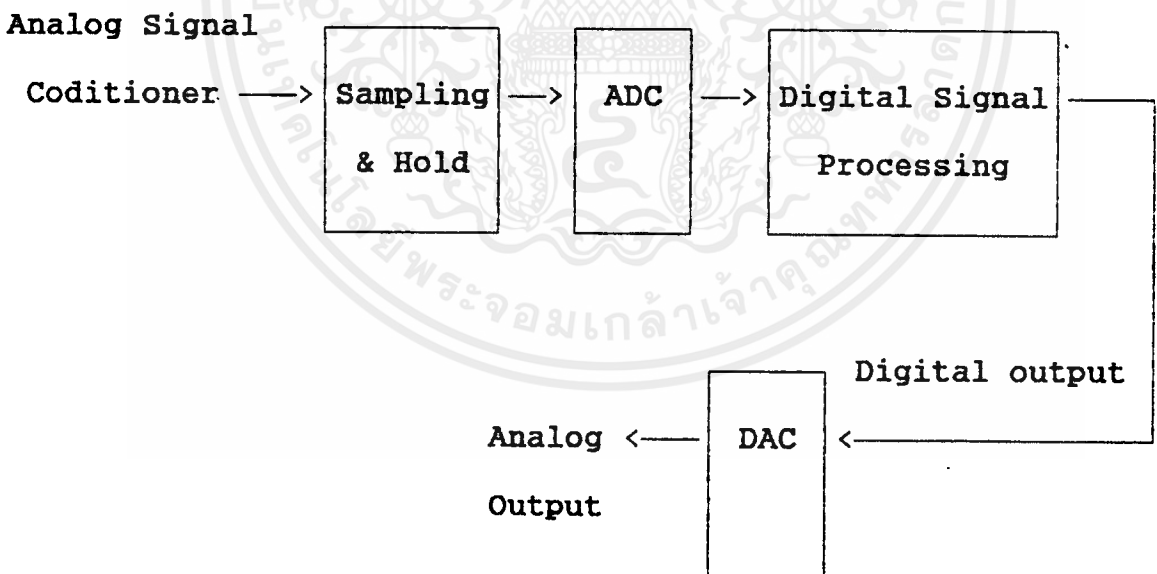
บทที่ 1 หลักการประมวลผลสัญญาณดิจิทัล (DSP)

ขั้นตอนการประมวลผลสัญญาณของ DSP สามารถมองเป็นภาพคร่าวๆ ได้ดังนี้

- 1) การ Sampling คือ การสุ่มสัญญาณอนาลอกที่เข้ามาเป็นอินพุตของระบบ
- 2) แปลงค่าของสัญญาณที่สุ่มมาให้เป็นดิจิทัล
- 3) ประมวลข้อมูลดิจิทัลนั้นตามอัลกอริทึมทางคณิตศาสตร์
- 4) หากเข้าที่พุดต้องการสัญญาณที่เป็นอนาลอก ก็จะต้องทำการแปลงผลลัพธ์ที่ได้

จากดิจิทัลเป็นอนาลอก

ขั้นตอนต่างๆ ดังกล่าว สามารถเขียนเป็นแผนภาพได้ดังรูป 1.1 ซึ่งในการแปลงสัญญาณเพื่อทำให้เป็นสัญญาณดิจิทัล เพื่อการประมวลผล และการแปลงสัญญาณกลับจากดิจิทัลเป็นอนาลอก อาศัยทฤษฎีการแปลงสัญญาณ ส่วนการประมวลผลนั้นจะพิจารณาในระบบสัญญาณเวลาไม่ต่อเนื่อง ซึ่งจะกล่าวต่อไปดังนี้



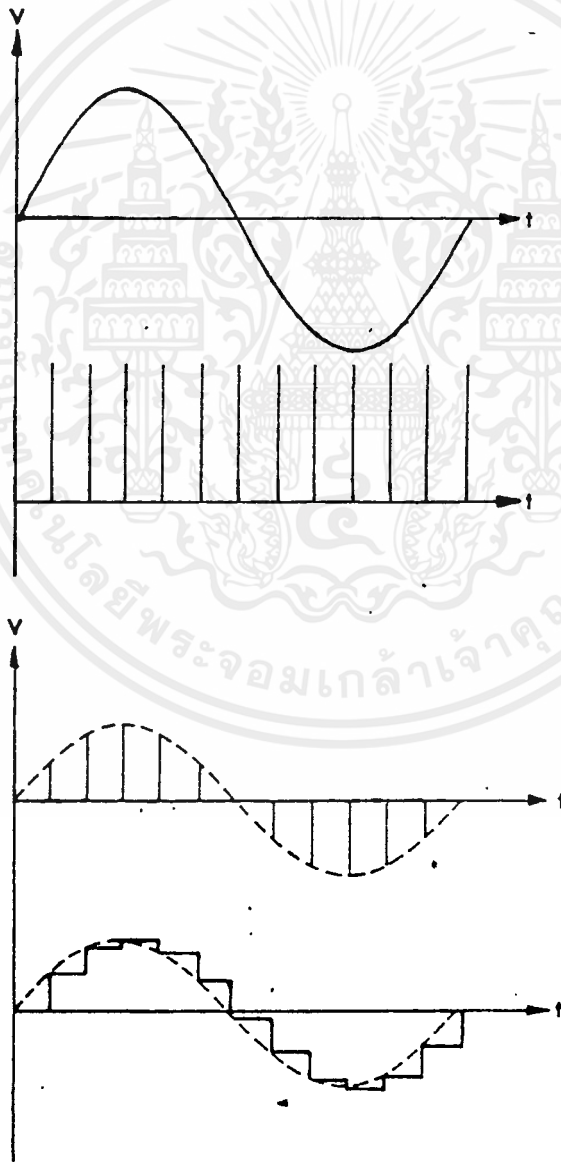
รูปที่ 1.1 แผนภาพแสดงการทำงานในการประมวลผลสัญญาณดิจิทัล

1.1) ทฤษฎีของการแปลงสัญญาณ (Data Aquisition and Conversion)

รายละเอียดทางทฤษฎีสามารถอธิบายเป็นหัวข้อต่างๆ ได้ดังนี้  
 เอกสารนี้เป็นเอกสารที่เผยแพร่ในอินเทอร์เน็ตเพื่อการศึกษาเท่านั้น ไม่ใช่ว่าจะนำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.1 ทฤษฎีของการสุ่มข้อมูล (Sampling)

ในการสุ่มข้อมูลนั้นสัญญาณอนาลอกจะถูกสุ่มเป็นระยะๆ ดังที่ กลุ่มของสัญญาณที่สุ่มจะแทนด้วยขนาดความสูง ซึ่งเกิดจากการตัดต่อสัญญาณอนาลอกด้วยระยะเวลาอันสั้น ผลของการสุ่มด้วยความเร็ว จะเหมือนกับการดูขบวนสัญญาณพัลซ์ กับสัญญาณอนาลอก ซึ่งจะได้สัญญาณที่มีมอดูเลท (Modulate) ระหว่างขบวนพัลซ์กับสัญญาณอนาลอก ดังแสดงในรูปที่ 1.2 และถ้าเรามีการโพลต์สัญญาณที่สุ่มได้เอาไว้ เราก็จะได้สัญญาณ ดังในรูปที่ 1.2ง

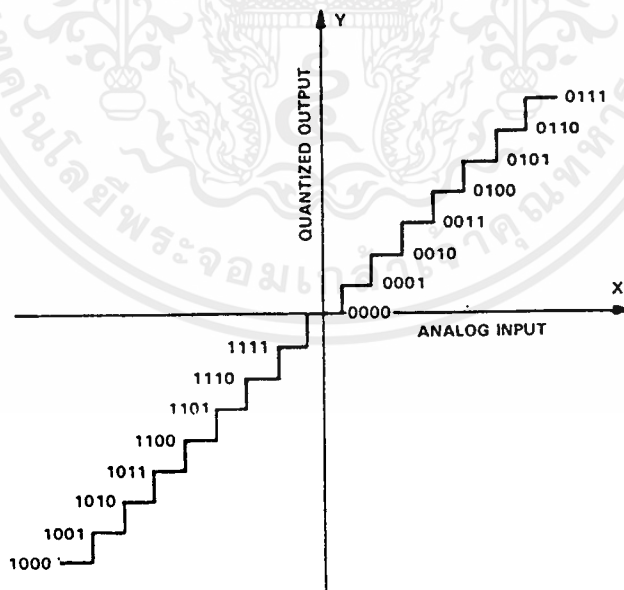


รูปที่ 1.2 แสดงการสุ่มสัญญาณ

อัตราการสุ่มสัญญาณ หรือความถี่ของการสุ่มสัญญาณควรมีค่าเท่าใด ที่ข้อมูลที่สุ่มได้ จะเป็นตัวแทนที่ดีของสัญญาณต่อเนื่องนั้น คำตอบคือ ขึ้นอยู่กับความถี่ของสัญญาณอนาลอก และจากทฤษฎีการสุ่ม (Nyquist Theorem) กล่าวไว้ว่า " ถ้าสัญญาณต่อเนื่อง ที่มี ความถี่ฮาร์โมนิก (Harmonic Frequency) ไม่เกิน  $f$  แล้ว สัญญาณดังกล่าวสามารถ เปลี่ยนกลับมาเช่นเดิม โดยไม่สูญเสียรายละเอียด หรือผิดเพี้ยนไป ถ้าอัตราการสุ่ม มากกว่า  $2f$  "

### 1.1.2 ทฤษฎีการควอนไทซ์ (Quantizing Theorem)

การควอนไทซ์ เป็นขบวนการที่แปลงสัญญาณอนาลอก ว่าเป็นข้อมูลทางดิจิทัลที่เป็น สัดส่วนกับสัญญาณอนาลอก เช่นอยู่ในรูปของรหัสไบนารี เป็นต้น หากเรานำเข้าที่พิกัดที่ได้จากการควอนไทซ์ และขนาดของสัญญาณอนาลอก มาเขียนเป็นกราฟ ก็จะได้กราฟ ควอนไทซ์ทรานสเฟอร์ฟังก์ชัน (Quantize Transfer Function) ดังแสดงในรูปที่ 1.3



รูปที่ 1.3 แสดงทรานสเฟอร์ฟังก์ชันของการควอนไทซ์ 4 บิต

จุดสำคัญของการควอนไทซ์นั้น ได้แก่ <sup>การละเอียด</sup> ไรโซลูชัน (Resolution) ของตัวควอนไทซ์ ซึ่งจะสามารถกำหนดได้จากจำนวนบิตของเข้าที่พิกัดรหัสดิจิทัลให้ หรือจากกราฟพุ่มที่ 1.3 ก็คือส่วนกลับของความกว้างของขั้นบันไดทางแกนนอนนั่นเอง

ค่าความกว้างของขั้นบันไดนี้จะสามารถคำนวณได้จาก

$$Q = \text{FSR} / 2^n$$

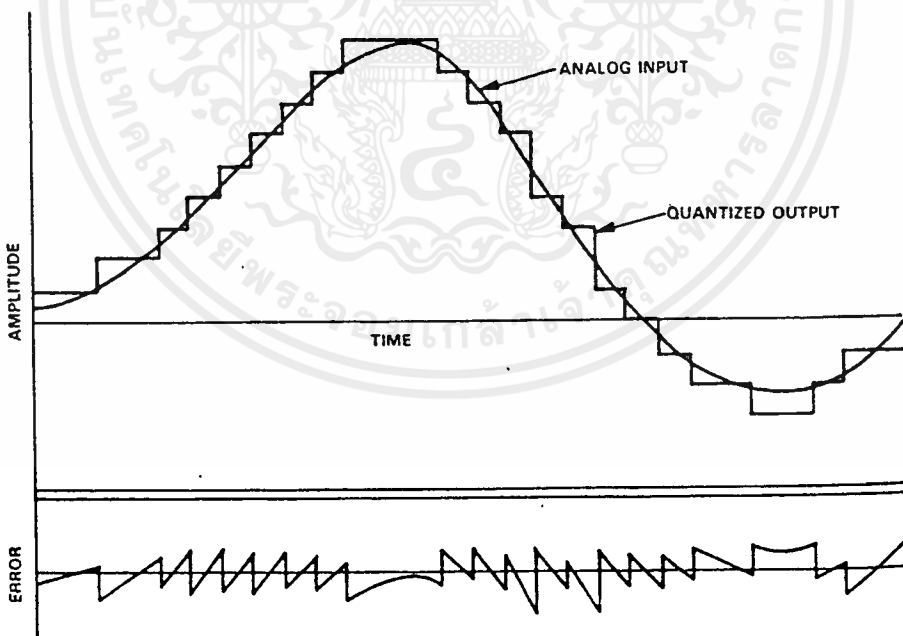
โดยที่

$Q$  : ค่าความกว้างของบันไดทางแกนนอน

$\text{FSR}$  : ช่วงเต็มสเกลของสัญญาณอนาล็อก (Full Scale Range)

$n$  : จำนวนบิตของรหัสดิจิทัล

จากสมการข้างต้นจะเห็นว่าเมื่อจำนวนบิตมากขึ้น ขนาดของ  $Q$  ก็จะลดลงและถ้าเรานำสัญญาณอนาล็อกใดๆ มาทำการควันไตซ์ จะเห็นได้ว่าเมื่อนำเอาผลที่ได้จากการควันไตซ์ มาเปรียบเทียบกับสัญญาณอนาล็อกนั้นแล้ว ก็จะพบว่ามีความผิดพลาดเกิดขึ้น ซึ่งเราจะเรียกความผิดพลาดนี้ว่า ความผิดพลาดควันไตซ์ (Quantizing Error) ดังแสดงในรูปที่ 1.4

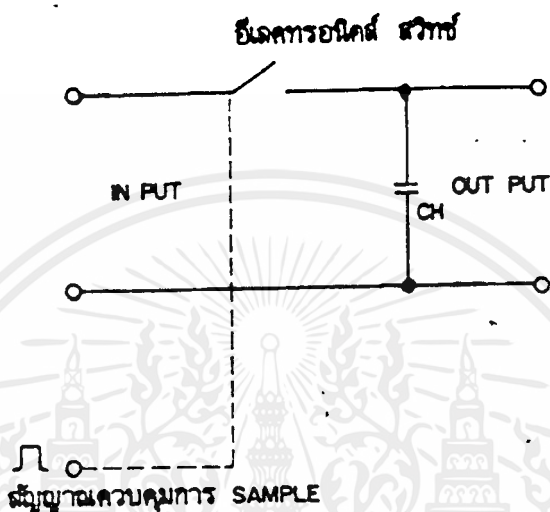


รูปที่ 1.4 แสดงความผิดพลาดควันไตซ์

ความผิดพลาดนี้เป็นธรรมชาติของการควันไตซ์ ซึ่งจะทำให้การแก้ไขไม่ได้ แต่เราจะลดค่าความผิดพลาดนี้ได้ หรือ ก็โดยการเพิ่มจำนวนบิตของการควันไตซ์ ซึ่ง ค่าความผิดพลาดนี้เอง ที่จะเป็นตัวกำหนดความแม่นยำในการใช้งาน DSP ของเอกสารทุกครั้งที่มีการนำไปใช้

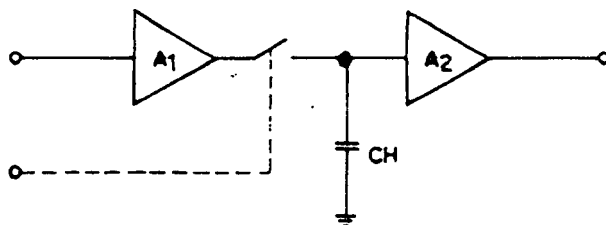
1.1.3 วงจรแชนเปิลแอนด์โฮลด์

วงจรแชนเปิลแอนด์โฮลด์ โดยพื้นฐานแล้ว จะเป็นวงจรหรือ อุปกรณ์เก็บแรงดัน (Voltage Memory) ซึ่งใช้อุปกรณ์ที่สำคัญคือ ตัวเก็บประจุ ดังานรูปที่ 1.5



รูปที่ 1.5 แสดงวงจรพื้นฐานของการแชนเปิลแอนด์โฮลด์

อิลคทรอนิกส์สวิทช์จะต่อสัญญาณแรงดันเข้ากับตัวเก็บประจุ ซึ่งสวิทช์จะถูกควบคุมจากแชนบลิ้งพัลส์ ช่วงเวลาการตัดสวิทช์ และเวลาในการประจุ จนแรงดันถึงค่าที่สุ่มมานั้น จะเรียกว่า อเพอเจอร์ไทม์ (Aperture Time) ของแชนบลิ้งแอนด์โฮลด์

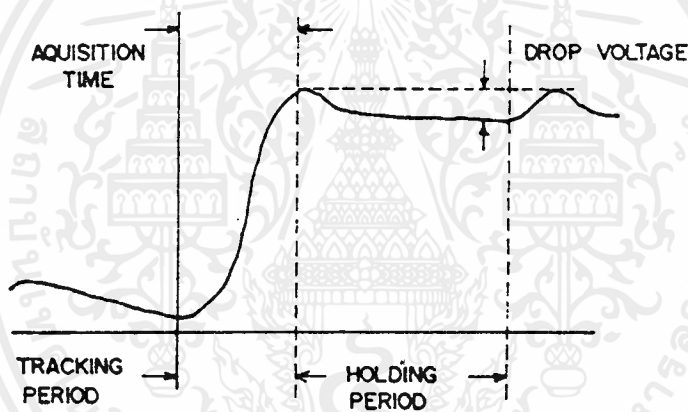


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและข้อมูลในสิ่งนี้ด้วยเอกสารเหล่านี้เป็นการนำไปใช้

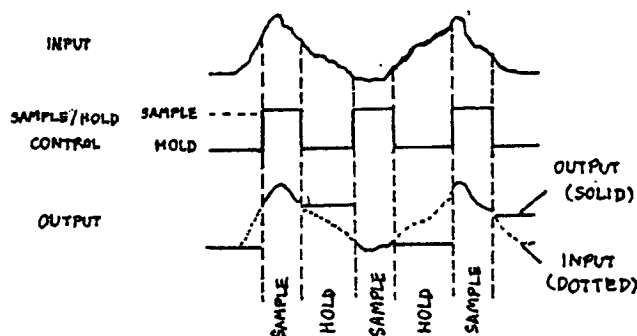
รูปที่ 1.6 แสดงวงจรแชนเปิลแอนด์โฮลด์ที่ใกล้เคียงกับวงจรที่ใช้งานจริง

านรูปที่ 1.6 โดยการเพิ่มเติมบัฟเฟอร์แอมพลิฟายเออร์ (Buffer Amplifier) เข้าทางส่วนอินพุท และเข้าที่พุทของวงจรมินิแอมพลิฟายเออร์ทางด้านอินพุท จะช่วย ท้าให้วงจรมินิแอมพลิฟายเออร์มีอิมพีแดนซ์สูง สะดวกต่อการใช้งาน และสามารถเพิ่มกระแสเพื่อทำ การประจุตัวเก็บประจุได้เร็วขึ้น ส่วนแอมพลิฟายเออร์ทางด้านเข้าที่พุท จะช่วยทำให้ เข้าที่พุทมีอิมพีแดนซ์ต่ำ สามารถขับเอาสัญญาณได้ง่าย ส่วนสำคัญที่ต้องพิจารณาคือ จะต้อง ใช้งานแอมพลิฟายเออร์ที่กินกระแสต่ำ เพื่อให้ดึงกระแสจากตัวเก็บประจุ ในช่วงโวลต์ สัญญาณน้อยที่สุด มิฉะนั้นจะเกิดการลดลง (Drop) ของแรงดัน ดังแสดงในรูปที่ 1.7

ส่วนรูปที่ 1.8 แสดงรูปคลื่นสัญญาณที่เกิดจากวงจรแซมเปิลแอนด์โฮลด์ในทางอุดมคติ



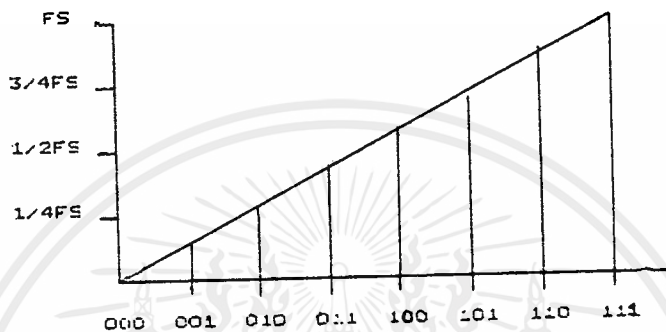
รูปที่ 1.7 แสดงการตกของแรงดันที่โวลต์ไว้



รูปที่ 1.8 แสดงรูปคลื่นสัญญาณจากวงจรแซมเปิลแอนด์โฮลด์ในทางอุดมคติ

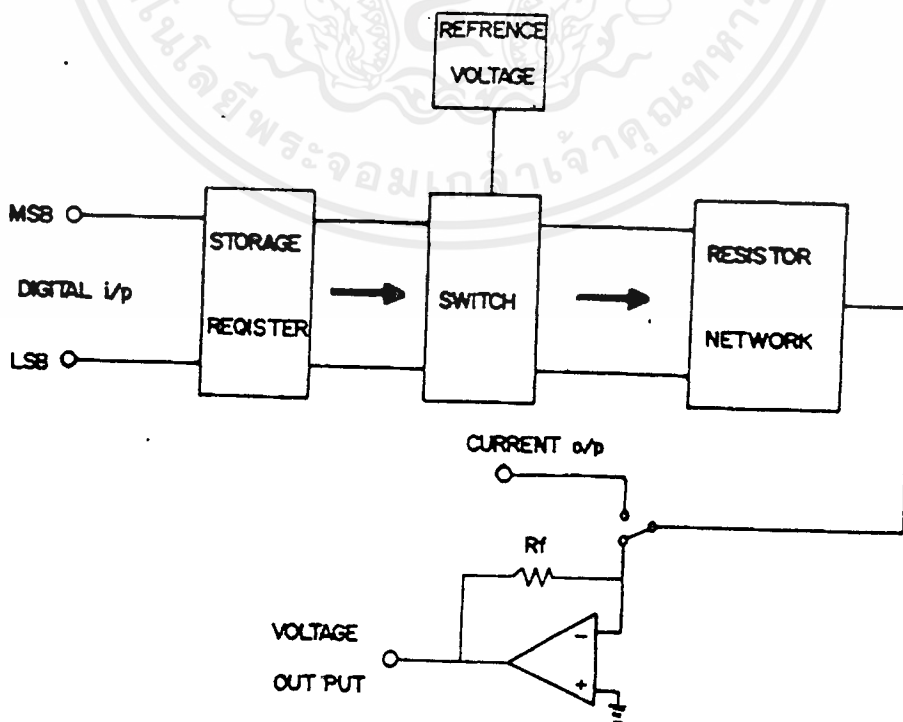
1.1.4 วงจรเปลี่ยนสัญญาณดิจิตอลให้เป็นอนาลอก (DAC)

ดีเอซี เป็นอุปกรณ์สำคัญอย่างหนึ่ง ที่ทำหน้าที่เอาคอมพิวเตอร่าบาส์ร่วมกับอุปกรณ์หรือวงจรอนาลอกอื่นๆ รูปที่ 1.9 จะแสดงทรานสเฟอร์ฟังก์ชันของ ดีเอซี 3บิท จะเห็นว่าวาร์หัสติจิตอลอินพุท 1 เวิร์ด จะเปลี่ยนเป็นแรงดันอนาลอก 1 ค่า



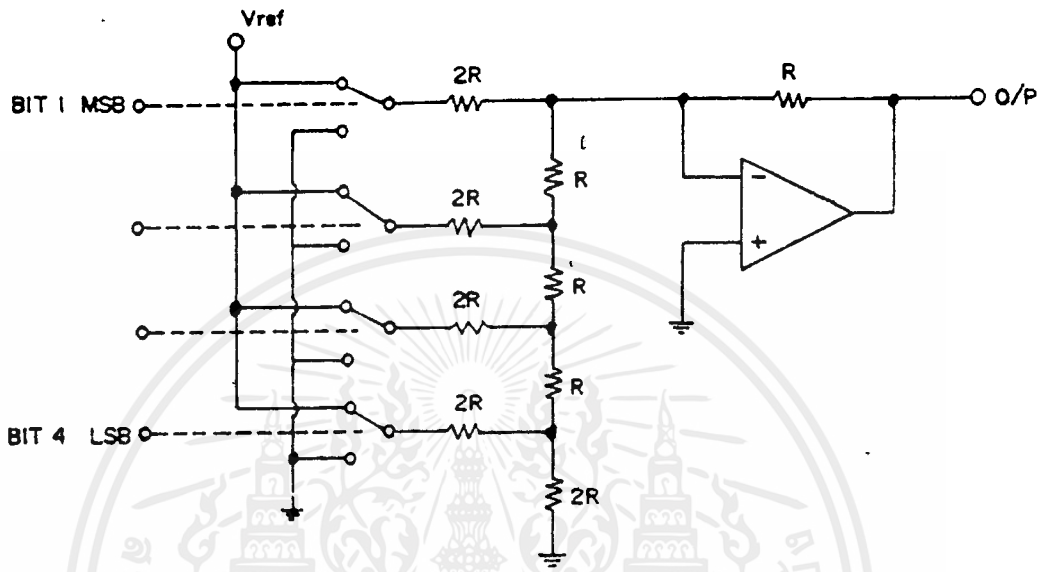
รูปที่ 1.9 แสดงทรานสเฟอร์ฟังก์ชันของดีเอซีขนาด 3 บิท

ลักษณะการจัดวงจรดีเอซี จะเป็นลักษณะดังรูปที่ 1.10



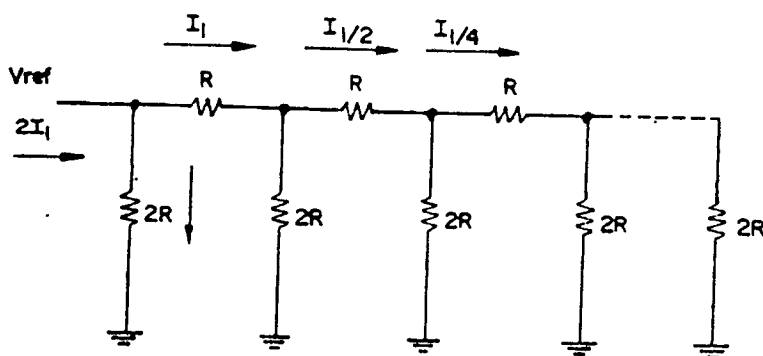
รูปที่ 1.10 แสดงบล็อกไดอะแกรมของ ดีเอซี

วงจรดีเอซี ที่ใช้กันมีหลายแบบ แต่ตอนนี้จะขอกล่าวถึง วงจรดีเอซีที่เป็นแบบตัวต้านทานชั้นบันได (R-2R ladder) ซึ่งเป็นแบบที่นิยมใช้กันมาก ลักษณะของวงจรแสดงในรูปที่ 1.11



รูปที่ 1.11 แสดงวงจรดีเอซีขนาด 4 บิต แบบ R-2R ladder .

ในวงจรนี้ สวิตช์จะตัดต่อให้แรงดันอ้างอิง ต่อเข้ากับวงจรชั้นบันได จะเห็นว่า สวิตช์อินพุตรีซิสเตอร์ (2R) เมื่อมองเข้าไป จะเห็นคู่ของรีซิสเตอร์ระหว่างจุดต่อ R-2R ที่ติดกัน กระแสจะถูกบั่นทอนไปอัตรา 2/1 ซึ่งสอดคล้องกับรหัสาบารี่ดังรูปที่ 1.12



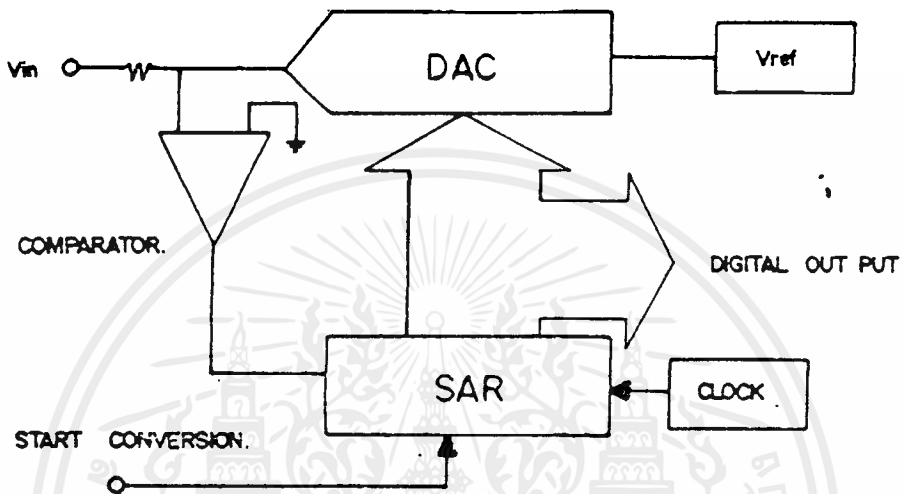
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งขอสงวนสิทธิ์ในข้อมูลและเนื้อหาที่ปรากฏในเอกสารฉบับนี้ที่มีการนำไปใช้

รูปที่ 1.12 แสดงการแบ่งแรงดันจากแรงดันอ้างอิง



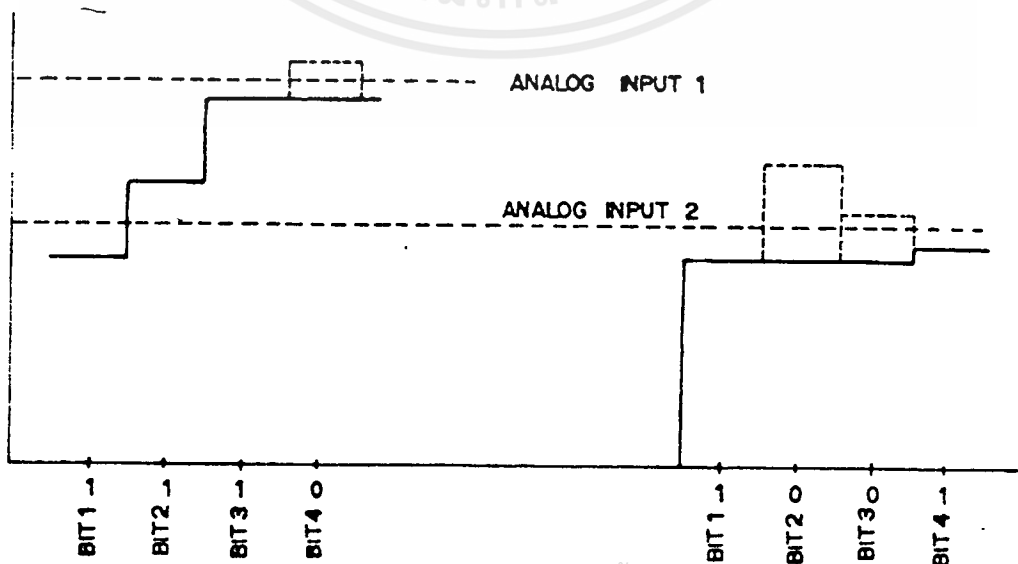
### 1.1.5 วงจรเปลี่ยนสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอล

วงจรเอ็ดซีส่วนใหญ่ จะอยู่ในรูปของไอซี และแบบที่นิยมมาช้คือ แบบการประมาณ ซัดเซสซีฟ (Successive Approximation) ซึ่งนิยมใช้งานที่ที่ต้องการความเร็วสูง และปานกลาง วงจรแบบนี้ คล้ายกับเดคาร์ทอร์ที่ทำงานในลักษณะป้อนกลับ



รูปที่ 1.13 แสดงบล็อกไดอะแกรม ของตัวแปลงแบบซัดเซสซีฟ

ในรูปที่ 1.13 แสดงฟังก์ชันต่างๆ ในเอ็ดซี คอมพาราเตอร์ จะคอยเปรียบเทียบ เอาท์พุทจากเอ็ดซี กับสัญญาณอนาลอกอินพุท เอาท์พุทที่ได้จะควบคุม รีจิสเตอร์ของตัว ประมาณแบบซัดเซสซีฟ การทำงานของ SAR จะเป็นดังนี้ (ดูรูป 1.14)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

รูปที่ 1.14 แสดงฟังก์ชันมิงไดอะแกรมของ เอ็ดซีแบบ SAR ที่มีการนำไปใช้

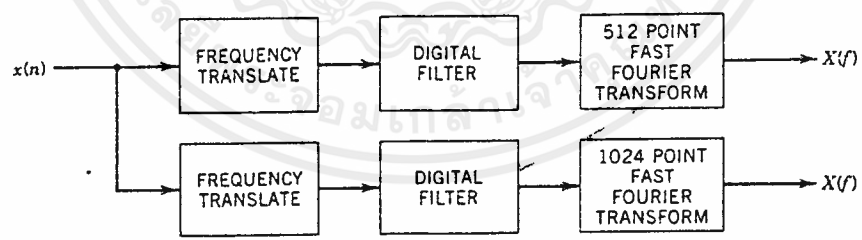
เมื่อสัญญาณนาฬิกาเข้ามา 1 ลูก ก็จะทำให้บิตสูงสุด เป็น "1" ส่วนบิตอื่น ๆ ยังคงเป็น "0" ดีเอชจะเปลี่ยนเอาต์พุตของ SAR ว่าเป็นสัญญาณอนาล็อกเปรียบเทียบกับสัญญาณอนาล็อกอินพุต ถ้าผลการเปรียบเทียบที่ตัวเปรียบเทียบ น้อยกว่าอินพุต ก็ดึงบิตนั้นเป็น "1" หากผลรวมของสองบิต มีค่ามากกว่า ก็ทำให้บิตนั้นเป็น "0" แต่ถ้า น้อยกว่าก็ดึงค่า "1" เอาไว้ แล้วทำการทดสอบบิตถัดไปด้วยวิธีดังกล่าว จนครบทุกบิต หรือจนกว่าเอาต์พุตจะต่างจาก ค่าแรงดันอินพุต ไม่เกิน "1" LSB

ข้อจำกัดประการหนึ่งสำหรับการทำงานคือ อนาล็อกอินพุตจะต้องคงที่ ในช่วงเวลา ที่ทำการเปลี่ยนสัญญาณ โดยเปลี่ยนแปลงได้ไม่เกิน 1/2 LSB ในช่วงสุดท้ายของการ เปลี่ยนสัญญาณดิจิตอล เอาต์พุตจะออกมาขนานกันทุกบิต

1.2) ระบบสัญญาณเวลาไม่ต่อเนื่อง (Discrete - Time Signal System)

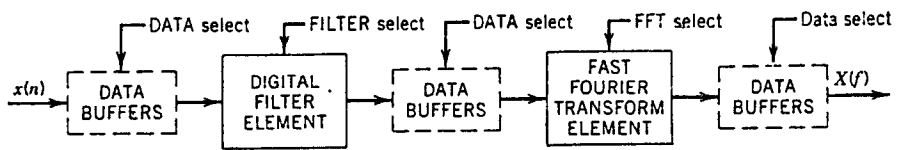
1.2.1 ตัวประมวลผลสัญญาณ

ในชุดเริ่มต้นของ DSP หน่วยประมวลผลกลางจะถูกออกแบบมา เพื่อการทำงาน อย่างหนึ่งโดยเฉพาะ ดังรูป 1.15



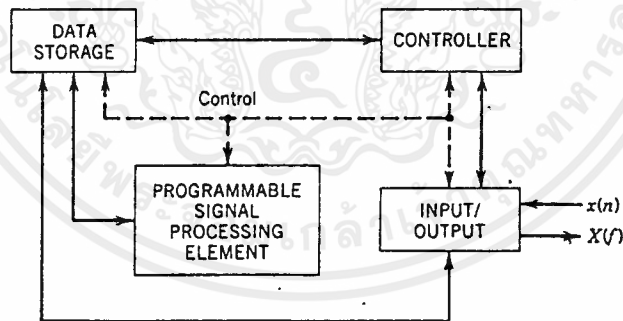
รูปที่ 1.15 แสดงการนำ DSP ทำงานเฉพาะอย่าง

ในแต่ละส่วน จะไม่สามารถโปรแกรมมาหมดได้ แต่จะมีเพียงการควบคุมแบบง่าย ๆ เท่านั้น เช่น การควบคุมการปิด-เปิด ต่อจากนั้นมา ระบบ DSP ได้ก้าวหน้าขึ้น โดยแต่ละส่วนสามารถจะเลือกได้ว่า จะทำงานเป็นอัลกอริทึมใด ซึ่งจะทำให้ระบบมีความยืดหยุ่น ยิ่งขึ้นดังรูป 1.16



รูปที่ 1.16 แสดงการำใช้งาน DSP โดยสามารถเลือกส่วนต่าง ๆ ได้

ความก้าวหน้าเกิดขึ้นเร็วมากเมื่อ เทคโนโลยีของไมโครโปรเซสเซอร์ ถูกสร้าง  
 ำที่ทำงานได้เร็วยิ่งขึ้น ทำให้การประมวลผลในระบบ DSP เปลี่ยนมาใช้เป็นระบบ  
 ไมโครโปรเซสเซอร์ ดังรูป 1.17



รูปที่ 1.17 แสดงการำใช้ไมโครโปรเซสเซอร์แบบโปรแกรมได้

ระบบ DSP ที่ำใช้ไมโครโปรเซสเซอร์ จะมีข้อดีคือ มีความยืดหยุ่นสูงมากกล่าวคือ  
 จะสามารถทำงานได้หลาย ๆ อย่าง ตามแต่ อัลกอริทึมที่โปรแกรมำให้

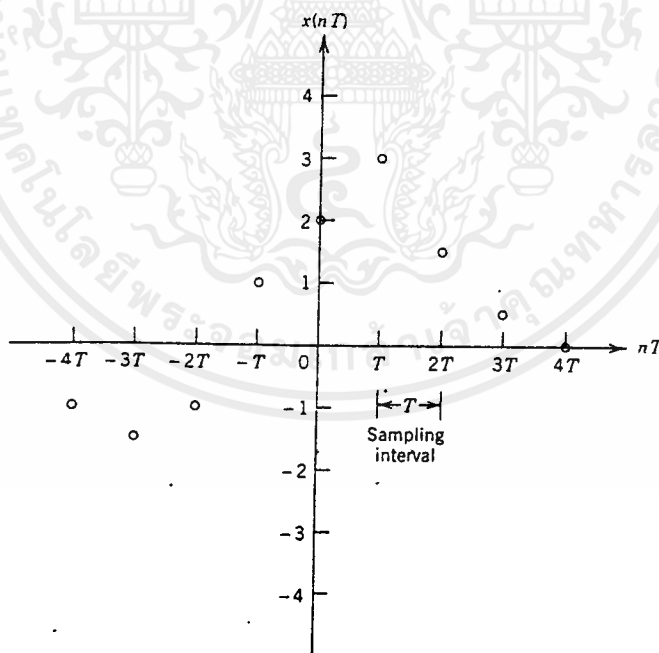
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2.2 สัญญาณเวลาไม่ต่อเนื่อง (Discrete-Time Signal)

ในระบบสัญญาณเวลาไม่ต่อเนื่อง สัญญาณจะถูกกำหนดโดย

$$\{X(nT)\} \quad ; \quad n = \{\dots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \dots\}$$

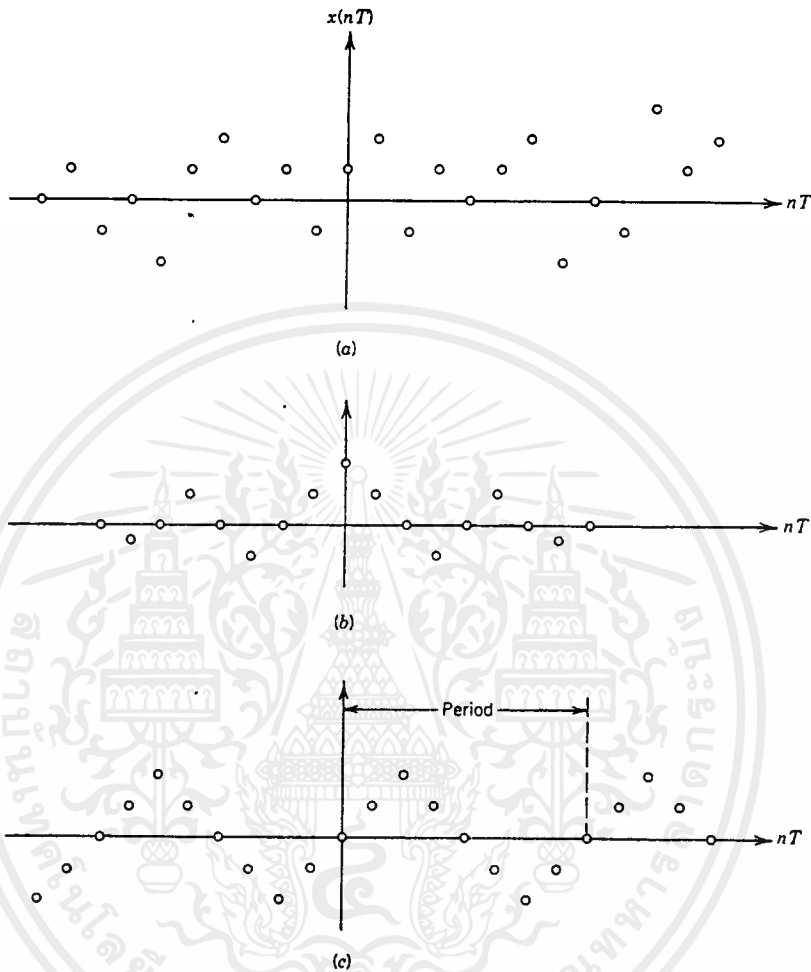
เมื่อ  $T$  คือค่าของช่วงห่างของเวลา ในระหว่างการเก็บข้อมูล (Sampling Interval) ค่าของ  $\{X(nT)\}$  มาจากค่าของข้อมูล ที่ถูกเก็บเข้ามาเรื่อย ๆ ซึ่งจะแทนค่าของปรากฏการณ์ที่เกิดขึ้นอย่างต่อเนื่อง  $a(t)$  ดังรูป 1.18



รูปที่ 1.18 แสดงสัญญาณ เวลาไม่ต่อเนื่องของ  $x$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะสัญญาณอนาล็อกรูปแบบต่าง ๆ ที่ถูกสุ่มเข้ามา จะแสดงดังรูป 1.19

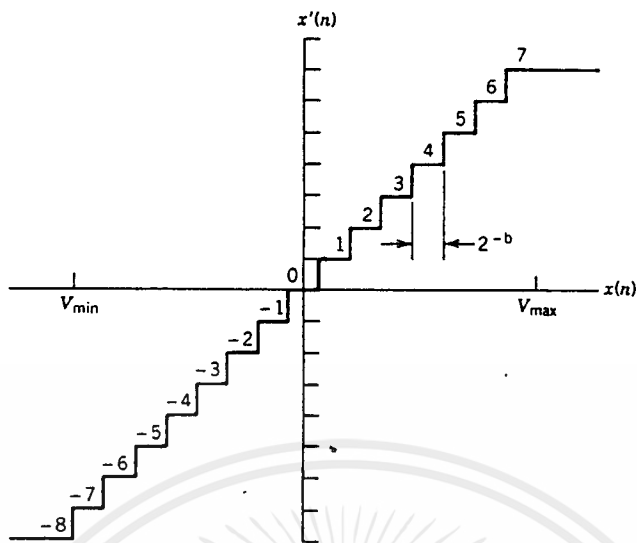


รูปที่ 1.19 แสดงการสุ่มสัญญาณรูปแบบต่าง ๆ เข้ามา

เนื่องจากสัญญาณอนาล็อกนั้น จะมีความต่อเนื่องไม่สิ้นสุด แต่ในระบบดิจิทัลจะไม่สามารถทำได้ ดังนั้นจึงต้องเปลี่ยนข้อมูลอนาล็อกที่มีอยู่ให้มีค่าที่สิ้นสุดก่อน โดยการควอนไทซ์ (Quantization: Q) เราจะได้  $\{X'(nT)\}$  คือสัญญาณที่ถูกควอนไทซ์แล้ว

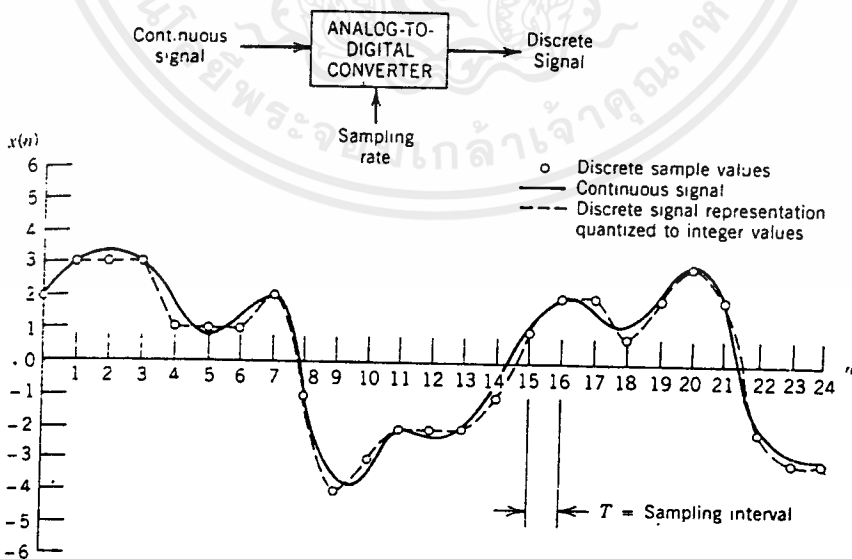
$$\{X'(nT)\} = Q \{X(nT)\}$$

เอกสารนี้เมื่อ Q กำหนดโดยกราฟดังรูป 1.20 วิชาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.20 แสดงการควอนไทซ์สัญญาณ A/D

เมื่อสัญญาณไม่ต่อเนื่องผ่านการควอนไทซ์แล้ว ก็จะได้สัญญาณดังตัวอย่างรูป 1.21



$$x(n) = \{ \dots x(-2), x(-1), 2, 3, 3, 3, 1, 1, 1, 2, -1, -4, -3, -2, -2, -2, -1, 1, 2, 2, 1, 2, 3, 2, -2, -3, -3, x(25), x(26), \dots \}$$

รูปที่ 1.21 แสดงการควอนไทซ์สัญญาณอนาลอกอินพุท

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอนเท่านั้น การใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และค่าความแตกต่างระหว่าง สัญญาณอินพุทและ สัญญาณที่ได้จากการควอนไทซ์ เราจะเรียกว่า ค่าควอนไทซ์เช็ซันเออเอรอร์ (Quantization Error) จะกำหนดโดยสมการ

$$e(nT) = x'(nT) - x(nT)$$

เราจะกำหนดสัญญาณพื้นฐานต่าง ๆ  $\{x(nT)\}$  ได้คือ

### 1.2.2.1 สัญญาณกระตุ้นหนึ่งหน่วย (Unit Impulse Sequence)

จะถูกกำหนดโดย

$$\delta(nT) = \begin{cases} 1 & \text{เมื่อ } n = 0 \\ 0 & \text{เมื่อ } n \neq 0 \end{cases}$$

ดังรูป 1.22



รูปที่ 1.22 แสดงสัญญาณกระตุ้นหนึ่งหน่วย

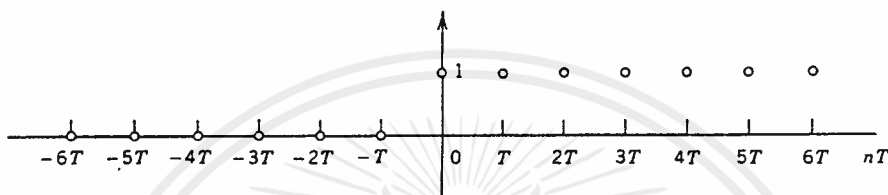
สัญญาณกระตุ้นหนึ่งหน่วยนั้น มีประโยชน์มากในการวัด การตอบสนองของระบบ และเราสามารถกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ  $x(nT) = \sum_{k=-\infty}^{\infty} x(nT) \delta(nT - kT)$  นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.2.2 สัญญาณขั้นหนึ่งหน่วย (Unit Step Sequence) จะถูกกำหนดโดย

$$u(nT) = \begin{cases} 1 & \text{เมื่อ } n \geq 0 \\ 0 & \text{เมื่อ } n < 0 \end{cases}$$

ดังรูป 1.23



รูปที่ 1.23 แสดงสัญญาณขั้นหนึ่งหน่วย  $u(nT)$

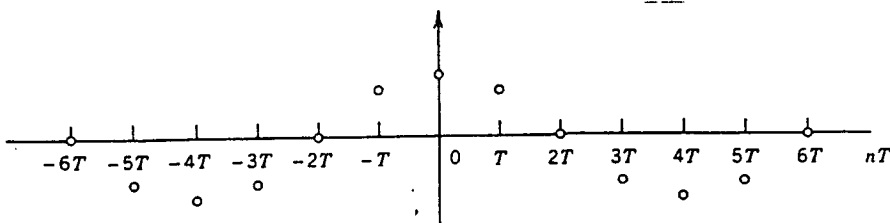
และจะได้ว่า

$$u(nT) - u(nT-1) = \delta(nT)$$

1.2.2.3 สัญญาณรูปคลื่นไซน์ (Sinusoidal Sequence) จะถูกกำหนดโดย

$$x(nT) = \sin ( 2\pi f nT + \phi )$$

เมื่อ  $f$  คือความถี่ของการแกว่ง และ ค่า  $\phi$  คือค่าของเฟสดังแสดงในรูป 1.24



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งหารูปที่ 1.24 แสดงรูปคลื่นไซน์  $\cos(2\pi n/\delta)$  รังที่มีการนำไปใช้

### 1.2.3 การกระทำต่อสัญญาณเวลาไม่ต่อเนื่อง

การกระทำต่อสัญญาณก็คือ การเปลี่ยนรูปแบบของสัญญาณอินพุต ำให้ได้สัญญาณเอาต์พุตอื่น ๆ เราจะกำหนด

$$y(nT) = R[x(nT)]$$

เมื่อ  $y(nT)$  คือ เอาต์พุตของระบบ  
 $x(nT)$  คือ อินพุตของระบบ  
 และ  $R$  คือ วิธีการประมวลผลใด ๆ

และเมื่อเราทำ  $T = 1$  เราจะแทนสมการได้เป็น

$$y(n) = R[x(n)]$$

ในระบบประมวลผลสัญญาณไม่ต่อเนื่องนี้ ถ้าเป็นประมวลผลแบบเวลาจริง เราจะได้ว่าเอาต์พุตที่เวลา  $nT$  ใด ๆ จะไม่สามารถนำข้อมูลอินพุตที่เวลา มากกว่า  $nT$  มาคำนวณได้ เพราะข้อมูลนั้นจะยังไม่เกิดขึ้น

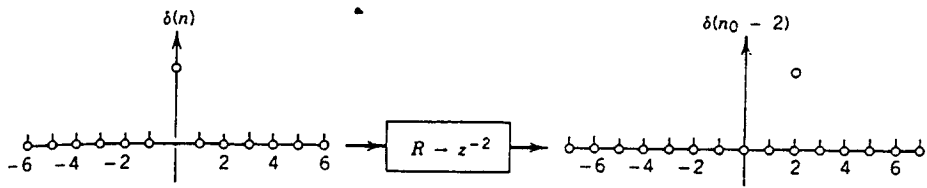
เราจะกำหนดการกระทำ ( $R$ ) ขึ้นพื้นฐาน ต่อสัญญาณไม่ต่อเนื่องดังนี้

1.2.3.1 การเลื่อนข้อมูล (Shift Operation) จะเป็นการเลื่อนค่าของสัญญาณอินพุตไปตามแกนเวลา  $nT$  กำหนดโดยสมการ

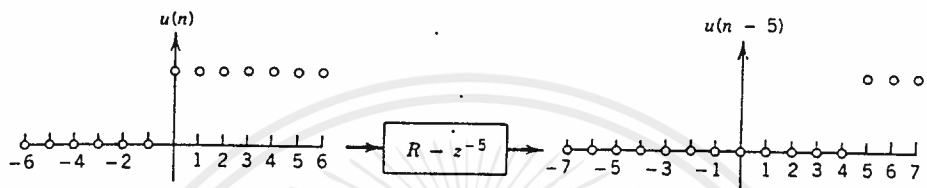
$$y(n) = z^{-m} x(n) = x(n-m)$$

ดังตัวอย่างในรูป 1.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



a.  $z^{-2}\delta(n) = \delta(n - 2)$



b.  $z^{-5}u(n) = u(n - 5)$

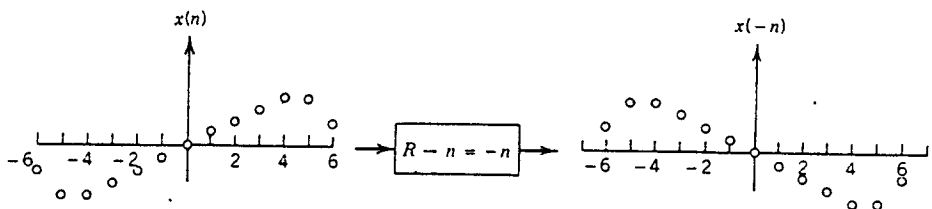
รูปที่ 1.25 แสดงการเลื่อนข้อมูล

1.2.3.2 การกลับค่า (Transpose Operation) เป็นการกลับค่าของสัญญาณ

ในแกนของ nT จะกำหนดโดยสมการ

$$y(n) = x(-n)$$

ดังตัวอย่าง รูป 1.26



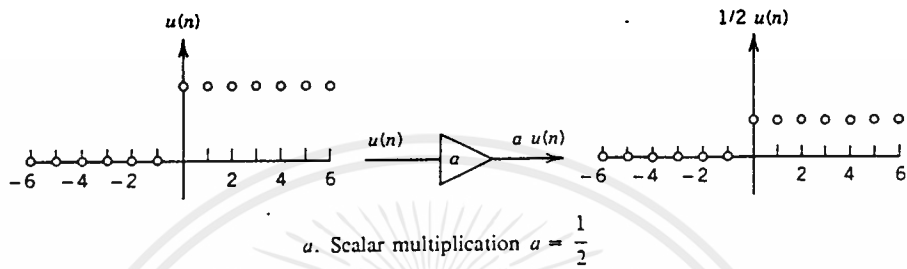
รูปที่ 1.26 แสดงการกลับค่าข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.3.3 การคูณสัญญาณด้วยค่าคงที่คงตัว (Scalar Multiplication - Operation) ก็คือการคูณสัญญาณด้วยค่าคงตัวนั่นเอง กำหนดโดย

$$y(n) = ax(n)$$

ดังรูป 1.27

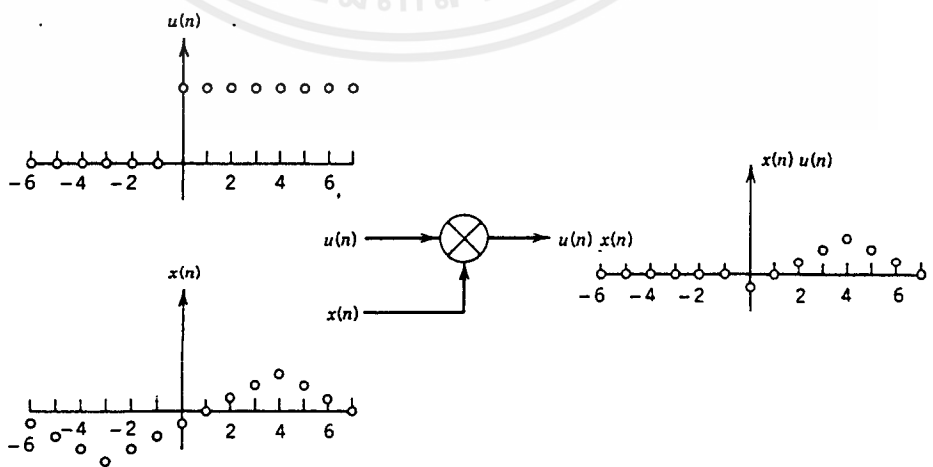


รูปที่ 1.27 แสดงการคูณสัญญาณด้วยค่าคงตัว

1.2.3.4 การคูณสัญญาณด้วยสัญญาณ (Signal (Vector) Multiplication - Operation) ซึ่งกำหนดโดย

$$y(n) = w(n)x(n)$$

ดังรูป 1.28



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามรูปที่ 1.28 แสดงการคูณสัญญาณด้วยสัญญาณทุกครั้งที่มีการนำไปใช้

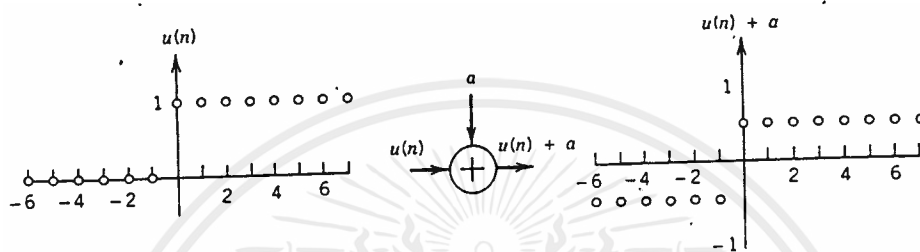
อาจเรียกได้ว่าเป็นการมอดดูเลขสัญญาณ (Modulation) นั้นเอง

1.2.3.5 การบวกสัญญาณด้วยค่าคงตัว (Scalar Addition Operation)

ใช้เป็นการเลื่อนระดับเฉลี่ยของสัญญาณ กำหนดโดย

$$Y(n) = a + x(n)$$

ดังรูป 1.29

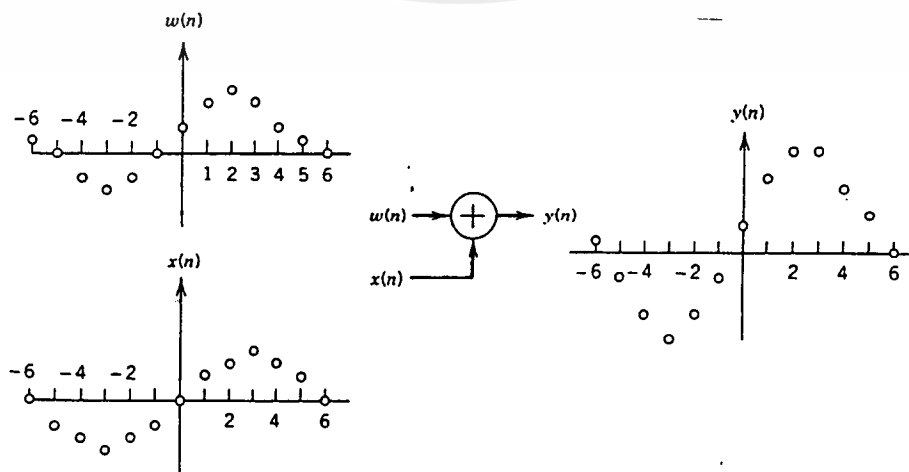


รูปที่ 1.29 แสดงการบวกสัญญาณด้วยค่าคงตัว

1.2.3.6 การบวกสัญญาณด้วยสัญญาณ (Signal (Vector) Addition Operation) กำหนดโดย

$$y(n) = x(n) + w(n)$$

ดังรูป 1.30



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามทำซ้ำหรือดัดแปลงเอกสารฉบับนี้โดยไม่ขออนุญาตทุกครั้งที่มีการนำไปใช้

รูปที่ 1.30 แสดงการบวกสัญญาณด้วยสัญญาณ

### 1.2.4 ระบบประมวลผลที่มีความเป็นเชิงเส้นและไม่เปลี่ยนแปลงตามเวลา (Linear Time-invariant Digital System)

ทฤษฎีความเป็นเชิงเส้นและไม่เปลี่ยนแปลงตามเวลา เป็นพื้นฐานที่สำคัญมาในระบบ DSP โดยระบบ DSP จะกระทำต่อสัญญาณที่เข้ามาด้วยอัลกอริทึมที่กำหนด แล้วได้เอาท์พุทออกมา จากที่เรากำหนดให้  $R$  คือการประมวลผลใด ๆ จะได้

$$y(n) = R[x(n)]$$

การที่ระบบจะมีความเป็นเชิงเส้นและไม่เปลี่ยนแปลงตามเวลาได้นั้น ระบบจะต้องถูกกำหนดด้วยการตอบสนอง ของระบบสัญญาณกระตุ้นหนึ่งหน่วย  $h(n)$

$$h(n) = R[\delta(n)]$$

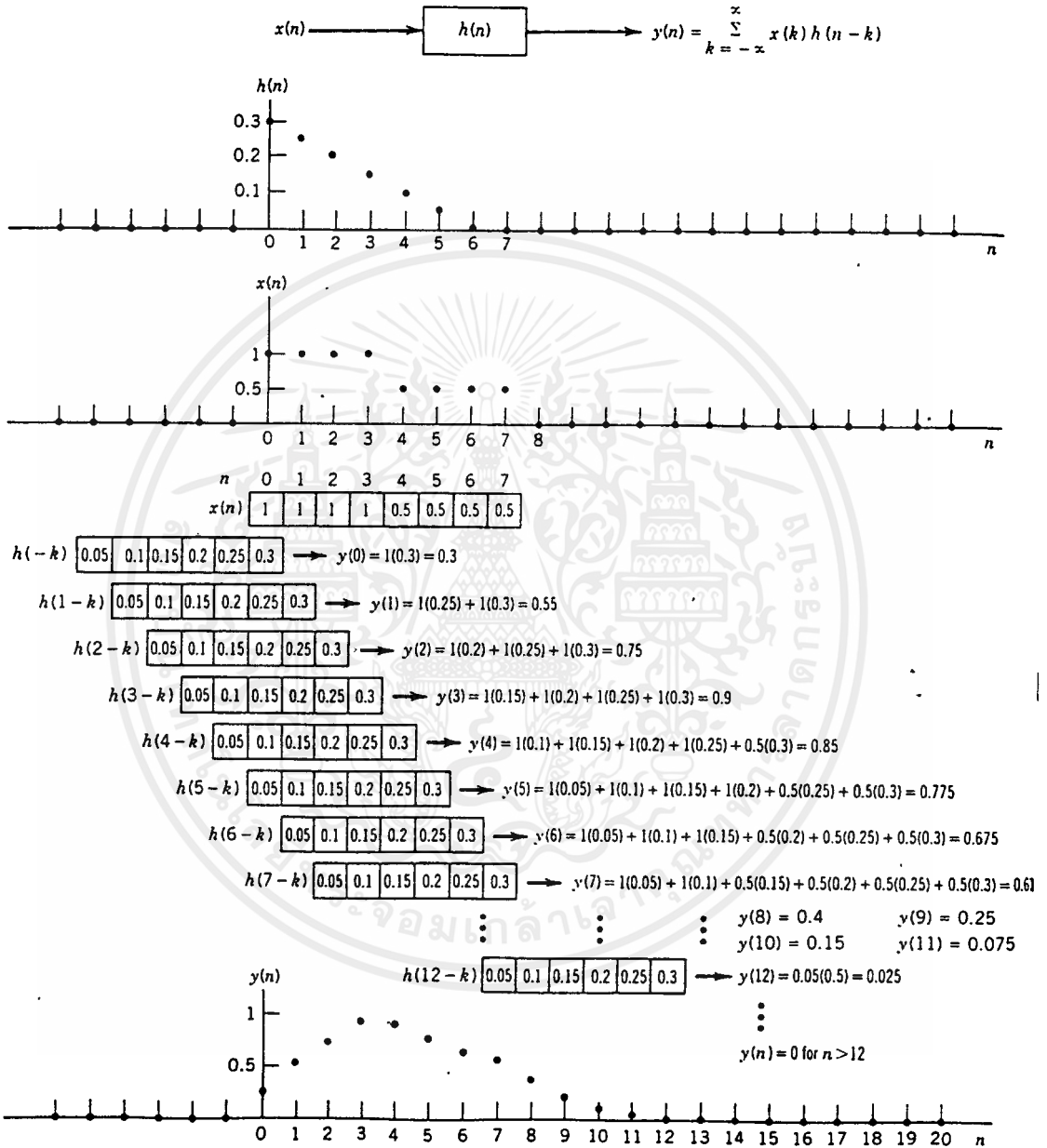
ดังนั้น เราจะหา  $y(n)$  ได้คือ

$$y(n) = R[x(n)] = \sum_{k=-\infty}^{\infty} x(k)h(n-k)$$

หรือ

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

ตัวอย่างเช่น จากรูป 1.31



รูป 1.31 แสดงการตอบสนองของระบบเมื่อได้รับสัญญาณกระตุ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.4.1 ความเป็นเชิงเส้น (Linearity) ระบบจะมีความเป็นเชิงเส้นได้ ก็ต่อเมื่อกำหนดค่าที่

$$y_1(n) = R[a_1x_1(n) + a_2x_2(n)]$$

และ

$$y_2(n) = a_1R[x_1(n)] + a_2R[x_2(n)]$$

ดังนั้นความเป็นเชิงเส้นของสมการ จะเกิดขึ้นเมื่อ

$$y_1(n) = y_2(n)$$

1.2.4.2 ความไม่เปลี่ยนแปลงตามเวลา (Time-Invariance) จะเกิดขึ้นเมื่อ

$$R[x(n-m)] = z^{-m} R[x(n)]$$

### 1.2.5 การกรองความถี่โดยใช้ดิจิทัล (Digital Filter)

ดิจิทัลฟิลเตอร์นี้ เป็นพื้นฐานการทำงาน ในระบบ DSP ที่สำคัญมากอย่างหนึ่ง อัลกอริทึมของดิจิทัลฟิลเตอร์ จะมาจากสมการ ความแตกต่างเชิงเส้นสัมประสิทธิ์คงที่ (Linear, Constant Coefficient Difference Equation)

$$\sum_{k=0}^k D_k y(n-k) = \sum_{l=0}^l C_l x(n-l)$$

เมื่อค่า  $D_k$  ,  $C_l$  คือค่าสัมประสิทธิ์ที่กำหนดคุณสมบัติของระบบ

และ  $y(n)$  ,  $x(n)$  คือค่าเอาต์พุตและอินพุตของระบบตามลำดับ

จากสมการข้างต้นเราสามารถเขียนใหม่ได้เป็น

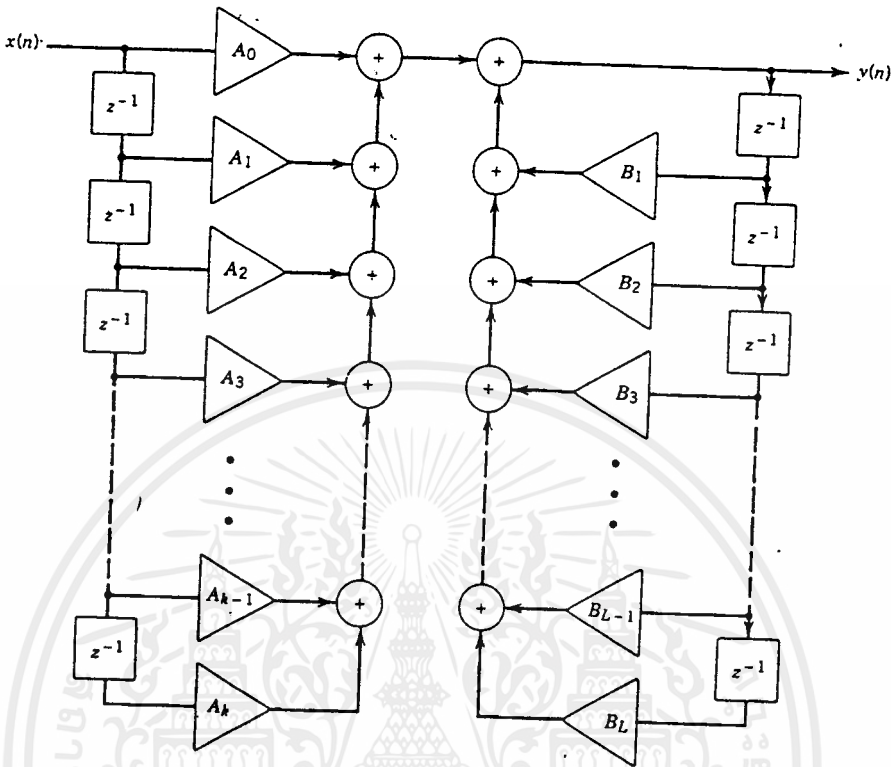
$$y(n) = \sum_{l=0}^l A_l x(n-l) - \sum_{k=1}^k B_k y(n-k)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$A_l = C_l/D_0 \quad \text{และ} \quad B_k = D_k/D_0$$

ซึ่งจะแสดงได้ด้วย รูป 1.32



รูปที่ 1.32 แสดง Direct form linear constant coefficient difference equation implementation ของสมการข้างบน

ดังนั้น การเปลี่ยนคุณสมบัติของฟิลเตอร์ ก็คือการเปลี่ยนค่าคงตัวนี้เอง ดิจิตอลฟิลเตอร์จึงแยกเป็นสองแบบ คือ

1.2.5.1 ฟิลเตอร์ที่มีการตอบสนองไม่สิ้นสุด (Infinite Impulse Response Filter: IRR) กล่าวคือ เอาท์พุท ( $y(n)$ ) ของฟิลเตอร์ จะถูกคำนวณจากอินพุท ( $x(n)$ ) ในปัจจุบัน และในอดีต รวมกับเอาท์พุทในอดีต ดังนั้นสมการที่ใช้ก็คือ

$$y(n) = \sum_{l=0}^1 A_l x(n-l) - \sum_{k=1}^k B_k y(n-k)$$

จากการกำหนดเอาท์พุทของระบบด้วยสมการ จะเห็นว่า เอาท์พุทที่ได้อาจเกิดการแกว่ง (Oscillate) ขึ้นได้แม้อินพุทจะหมดไปแล้วก็ตาม อดให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะกำหนดค่าทรานเฟอร์ฟังก์ชัน (Transfer Function) ในระนาบของ  $z$   $H(z)$  ได้คือ

$$H(z) = \frac{\sum_{i=0}^M A_i z^{-i}}{1 + \sum_{i=1}^N B_i z^{-i}}$$

อันดับของโพลเตอร์จะกำหนดด้วย ค่ากำลังมากที่สุดของ  $z$  ใน  $H(z)$

1.2.5.2 โพลเตอร์ที่มีการตอบสนองสั้นสุด (Finite Impulse Response Filter: FIR) ค่าเอาต์พุทของโพลเตอร์จะได้จากการคำนวณ รวมค่าของอินพุท  $x(n)$  ในปัจจุบันและในอดีต และจากสมการมาตรฐานของโพลเตอร์ เราจะได้ว่าค่า  $B_k$  นั้น เท่ากับศูนย์ ดังนั้นเราจะเขียนสมการของ FIR ได้เป็น

$$y(n) = \sum_{i=0}^1 A_1 x(n-1)$$

เราจะเห็นได้ว่า ค่าการตอบสนองต่อสัญญาณกระตุ้น หนึ่งหน่วย ก็คือ  $A_1$  นั่นเอง

$$h(n) = \sum_{i=0}^1 A_1 \delta(n-1)$$

จะได้

$$h(n) = A_1 n$$

แล้วเราสามารถเขียน สมการการตอบสนองในระนาบของ  $z$  ได้คือ

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}$$

เมื่อ  $n-1$  คือ ค่าลำดับที่ของโพลเตอร์

FIR โพลเตอร์นี้จะมีฟังก์ชัน ที่ขึ้นกับอินพุทเพียงอย่างเดียว ดังนั้นเมื่ออินพุทหมดไป เอาต์พุทจะลดลงด้วย จากลักษณะของโพลเตอร์นี้เอง ทำให้เราสามารถที่จะ กำหนดค่าคงตัว  $(A_1)$  ให้มีคุณสมบัติเฟสเชิงเส้น (linear Phase) ได้โดยง่าย ซึ่งจะเป็น ประโยชน์ในการใช้งานต่าง ๆ เช่น งานการาชังานเกี่ยวกับเสียงพูด

1.2.6 การสุ่มข้อมูล

การสุ่มสัญญาณที่ต่อเนื่อง (a(t)) เข้ามาเป็นสัญญาณที่ามต่อเนื่อง (x(nT)) คือ

$$x(n) = a(t) \Big|_{t=nT} \quad \text{เมื่อ } n \text{ เป็นจำนวนเต็ม และ } -\infty < n < \infty$$

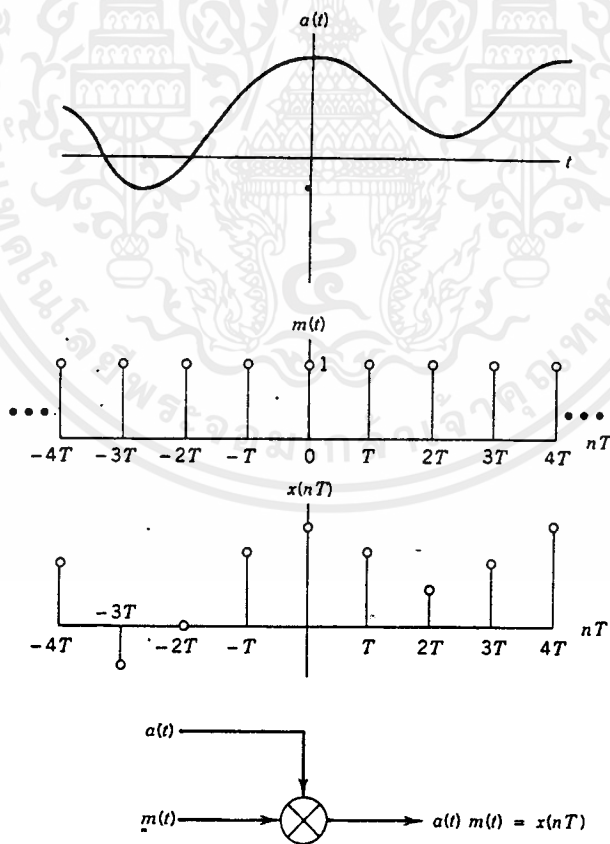
หรืออาจบอกได้ว่า การสุ่มเป็นการคูณสัญญาณที่ต่อเนื่อง ด้วยขบวนสัญญาณกระตุ้นหนึ่งหน่วย ((nT)) ซึ่งก็จะได้

$$x(nT) = a(t) * m(t)$$

เมื่อ

$$m(t) = \sum_{n=-\infty}^{\infty} \delta(t-nT)$$

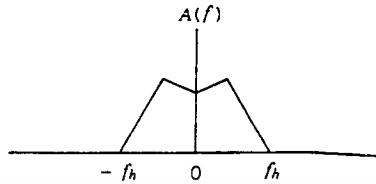
ดังรูป 1.33



รูปที่ 1.33 แสดงการสุ่มสัญญาณและ การมอดูเลตสัญญาณที่เกิดจากสัญญาณ

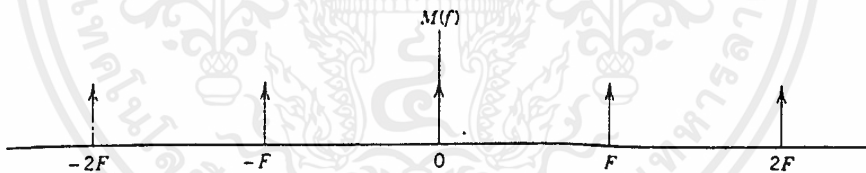
เอกสารนี้เป็นเอกสารที่สง กระตุ้นหนึ่งหน่วย เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเรากำหนดค่าสเปกตรัม (Spectrum) ของ  $a(t)$  ดังรูป 1.34



รูปที่ 1.34 แสดงสเปกตรัมความถี่ของสัญญาณอินพุท

เมื่อ  $f_n$  คือ ความถี่สูงสุดของสัญญาณ  $a(t)$  และสเปกตรัมของสัญญาณ  $m(t)$  จะเป็นดังรูป 1.35



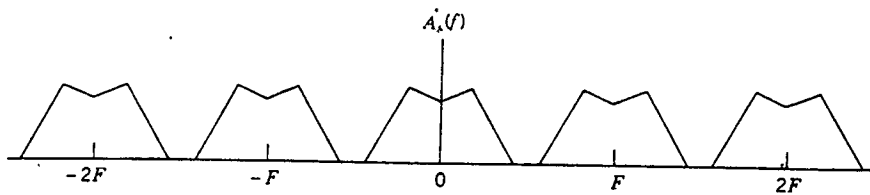
รูปที่ 1.35 แสดงสัญญาณการสุ่มข้อมูลที่จะนำไปมอดูเลต

เมื่อ  $F =$  ความถี่ของการสุ่ม  $= 1/T$  และค่าสเปกตรัม ของสัญญาณ  $x(nT)$  จะได้จากสมการ

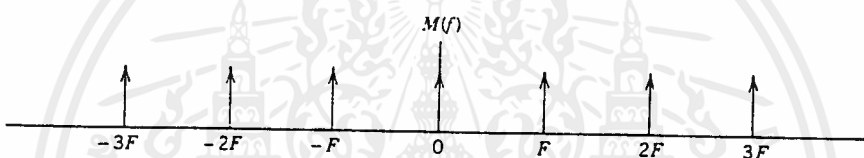
$$X(f) = A(f) * M(f)$$

เมื่อ  $X(f), A(f)$  และ  $M(f)$  คือสเปกตรัมของสัญญาณ จากข้างต้นเราจะได้  $X(f)$  ดังรูป 1.36 ถ้าค่า  $F > 2f_n$

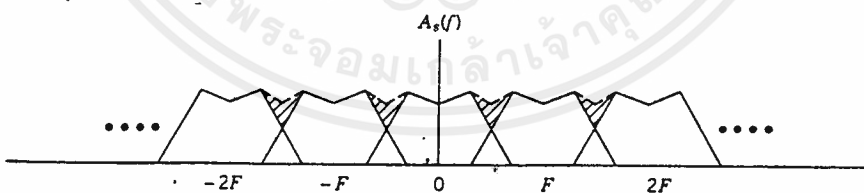
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.36 แสดงสเปกตรัมสัญญาณเอาท์พุท ที่  $F > 2f_h$   
และถ้าค่า  $F < 2f_h$  ดังรูป 1.37



รูปที่ 1.37 แสดงสัญญาณการสุ่มข้อมูลที่จะนำไปมอดูเลต ที่  $F > 2f_h$   
จะได้สเปกตรัมสัญญาณเอาท์พุท ดังรูป 1.38



รูปที่ 1.38 แสดงสเปกตรัมสัญญาณเอาท์พุท ที่  $F < 2f_h$

จะเห็นได้ว่าสเปกตรัมมีการซ้อนทับกัน (Aliasing) ซึ่งเป็นเหตุให้ข้อมูลที่ส่งมา  
ว่าที่  $(X(nT))$  ไม่สามารถแทนสัญญาณ  $a(t)$  ได้ ซึ่งค่า  $2f_h$  นี้จะ  
เรียกว่า Nyquist Rate สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.6.1 ความถี่ฐาน (Normalize Frequency)

ค่าของความถี่ฐาน จะเป็นค่าของเลขจำนวนจริง ตั้งแต่ 0 ถึง 1 โดยค่าของความถี่ฐานที่เป็น 1 จะแทนด้วยความถี่ของ  $a(t)$  เท่ากับความถี่ของการสุ่ม ( $f$ ) และในระบบดิจิตอลซิกแนลโปรเซสซิงนี้ ค่าของความถี่ที่ใช้ในการคำนวณต่าง ๆ ก็จะอยู่ในรูปของความถี่ฐานทั้งสิ้น เช่น สัญญาณ  $a(t)$  ที่มีความถี่ฐานเท่ากับ 0.25 เมื่อความถี่ในการสุ่ม เท่ากับ 10 KHz

$$\begin{aligned} \text{ดังนั้นเราจะได้ความถี่ ของ } a(t) &= 0.25 * 10 \\ &= 2.5 \text{ KHz เป็นต้น} \end{aligned}$$

1.2.7 การสังเคราะห์ด้วยความละเอียดสูง (Precision digital signal Generation)

สัญญาณซึ่งเป็นสิ่งพื้นฐานทางอิเล็กทรอนิกส์ และระบบ DSP อาจสร้างขึ้นได้โดยหลายวิธี แตกต่างกันออกไป ตามความต้องการใช้งาน แต่การสร้างสัญญาณที่มีความถูกต้องเที่ยงตรง โดยเฉพาะสัญญาณคลื่นรูปขายนั้นค่อนข้างยากและมีความซับซ้อน

ในระบบ DSP เราสามารถจะสร้างสัญญาณได้หลายวิธี ในที่นี้เราจะกล่าวถึงเพียงบางวิธีเท่านั้น โดยจะยกตัวอย่างการสร้าง สัญญาณคลื่นรูปขายน้

1.2.7.1 วิธีมองค่าจากตารางโดยตรง (Direct Table Lookup Method)

โดยเราจะกำหนดค่า Sine ของมุมต่าง ๆ รอบวงกลมหนึ่งหน่วย ทั้งหมด N มุม ทั่วเท่า ๆ กัน ซึ่งจะถูเก็บเป็นค่าในตาราง และได้อยู่ในรูปแบบดังนี้

$$S[n] = \text{Sin}(n * 360^\circ / N) \quad ; \quad n = 0, 1, 2, \dots, (N-1)$$

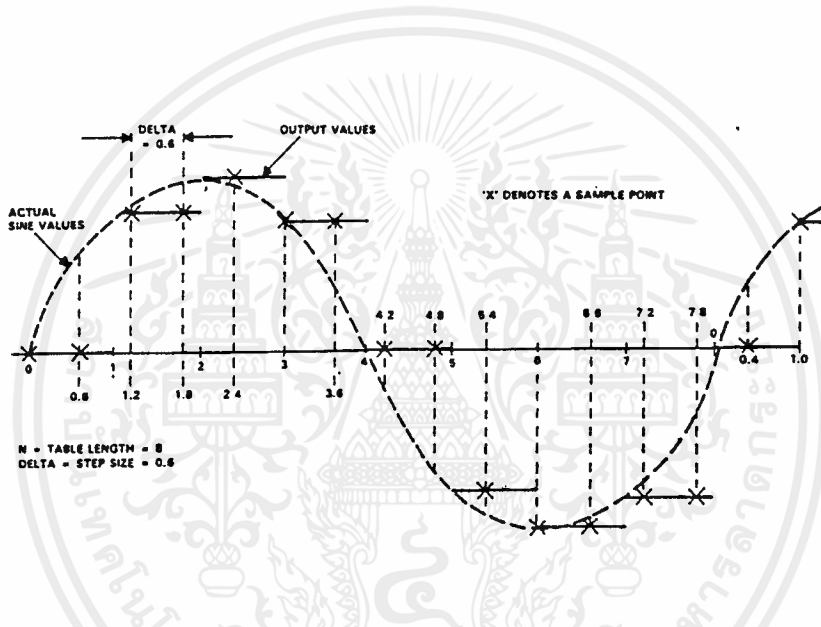
- เมื่อ  $S[n]$  คือ ค่าที่จะเก็บในตาราง
- $n$  คือ ค่าของตัวชี้
- $N$  คือ จำนวนของ  $S[n]$

สัญญาณ Sine จะถูกสร้างขึ้น โดยการกวาดค่าต่าง ๆ ในตาราง  $S[n]$  ออกมาโดยาชตัวชี้  $n$  ที่มีค่าเพิ่มขึ้นเรื่อย ๆ อย่างคงที่ และเมื่อเกิน 360 ก็จะกลับมาเริ่มต้นใหม่ซึ่งการเพิ่มขึ้นนี้ เราจะได้กำหนดด้วยค่า  $\Delta$  ซึ่งเป็นจำนวนจริง โดยดังนั้นเราจะได้อะไรที่  $(S[n])$  คือ

$$S[\text{mod}(k \cdot \Delta, N)] \quad ; \quad k = 1, 2, 3, 4, \dots$$

เมื่อ  $\text{mod}(A, B)$  กำหนดโดยมีค่าเท่ากับค่าที่เหลืออยู่ของ  $A/B$  เพื่อที่จะได้ผลลัพธ์เป็นเลขจำนวนเต็ม

การมอดุเลนั้นก็จะทำให้เกิดการวนไปรอบ ๆ ตาราง และจะได้สัญญาณเอาต์พุต ดังรูป 1.39 เมื่อค่า  $N=8$



รูปที่ 1.39 แสดงสัญญาณเอาต์พุตจากการใช้วิธี Direct Table Lookup

จะเห็นว่า สัญญาณเอาต์พุตที่ได้มีค่าประมาณใกล้เคียง สัญญาณชายน ซึ่งหากเราต้องการค่าที่แม่นยำมากขึ้น ก็จะทำให้ได้โดยการเพิ่มค่า  $N$  ในตาราง ยิ่งเพิ่มมากขึ้นก็จะทำให้ได้เอาต์พุตที่ ใกล้เคียงสัญญาณชายนมากขึ้น ความถี่ของสัญญาณเอาต์พุตที่ได้จะขึ้นอยู่กับตัวแปร 2 ตัว คือ

- 1) ค่าช่วงเวลาการสุ่ม ซึ่งคือช่วงเวลาระหว่าง  $S[n]$  และ  $S[n+1]$
- 2) ค่าการกระโดด Delta

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$f = \frac{\text{Delta}}{t * N} \quad \text{Hz}$$

และจากทฤษฎีของ Nyquist จะได้ว่า อย่างน้อยจะต้องมีข้อมูล 2 ค่า เกิดขึ้น  
านแต่ละคาบเวลา ของสัญญาณชายน์ ดังนี้

$$\text{Delta} < N/2$$

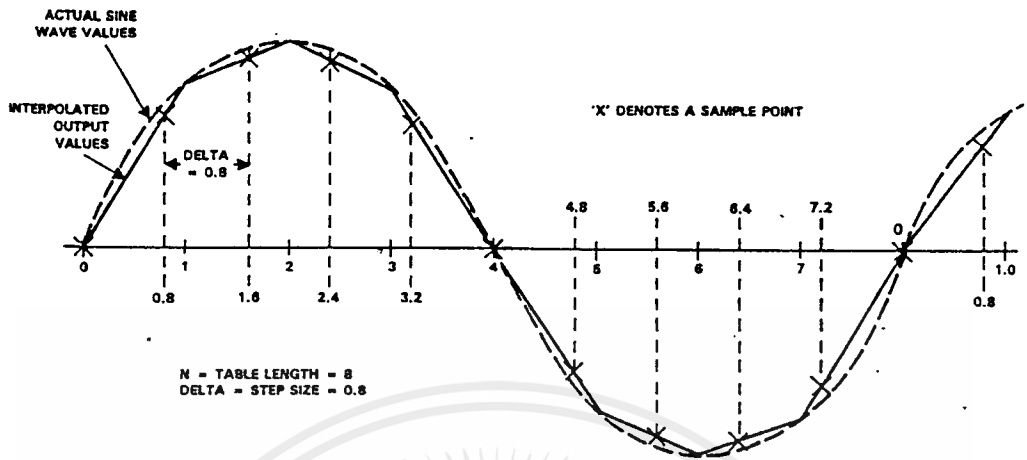
จากตัวอย่าง รูป 1.39 N มีค่าเท่ากับ 8 ค่า Delta = 0.6 ถ้าการสุ่ม  
8 ครั้งเกิดขึ้นใน 1 ms ดังนั้น  $t = 0.000125$  s จะได้

$$f = 0.6 / (8 * 0.000125) = 600 \quad \text{Hz}$$

1.2.7.2 วิธีมองค่าจากตารางโดยตรงแล้วเติมเต็ม (Direct Table Lookup  
Linear Interpolation Method) วิธีการนี้ เป็นวิธีที่พัฒนาต่อมา จากวิธีแรก  
(1.2.7.1) ในวิธีการแรกจะเห็นว่า เมื่อค่าของ Delta ไม่ได้เป็นค่าของจำนวนเต็ม  
ค่าความผิดพลาดจะเกิดขึ้นมา เมื่อที่ตำแหน่งการสุ่มนั้น ๆ ไม่มีตาราง จึงต้องนำเอา  
ค่าในตารางที่ตำแหน่งอื่น ๆ มาแทน ซึ่งเราจะแก้ปัญหานี้ได้ โดยการสร้างตารางขึ้นมา  
2 ตาราง ตารางแรกจะเหมือนกับในวิธีก่อน ส่วนตารางที่ 2 นั้น จะเป็นค่าความ  
แตกต่างระหว่างข้อมูล  $S[n]$  และ  $S[n+1]$  กำหนดได้โดย

$$T[n] = S[n+1] - S[n]$$

ซึ่งอัลกอริทึมนี้จะแสดงได้ดังรูป 1.40



รูปที่ 1.40 แสดงสัญญาณเอาต์พุตจากการใช้วิธี (Direct Table Lookup Linear Interpolation Method)

จากรูป 1.40 จะเห็นได้ว่า เมื่อค่าตำแหน่งการสุ่มไม่มีในตาราง S[n] ค่าของเอาต์พุต จะได้จาก การ คำนวณค่าความชันของสมการเส้นตรง และจะกำหนดเอาต์พุตได้ด้วย

$$\sin(360 (I+D)/N) = \sin(360 I/N) + D[\sin(360 (I+1)/N) - \sin(360 I/N)]$$

จะได้เป็น

$$S[n+D] = S[n] + DT(n)$$

เมื่อ D เป็นค่าของเลขจำนวนจริง  $0 < D < 1$

ซึ่งวิธีนี้จะทำให้เอาต์พุตที่ได้จะมีความถูกต้องกว่าวิธีแรก

บทที่ 2 สถาปัตยกรรมของ TMS32010

TMS32010 ถูกออกแบบโดยใช้สถาปัตยกรรมแบบฮาร์ดแวร์ เป็นดิจิทัลซิกแนลโปรเซสซิงที่มีความสามารถทำงานนฟังก์ชันทางฮาร์ดแวร์ได้ และมีข้อดีหลายอย่าง เช่น มีความเร็วในการคูณเลข 16 บิต 2 ชุดได้ภายในเวลาเพียง 200 นาโนเซค มีบารีเลชิฟเตอร์ 0 - 16 บิต และฮาร์ดแวร์ที่สำคัญคือ ออกซิลลารีรีจิสเตอร์ (Auxillary Register) สำหรับการเข้าหาหน่วยความจำข้อมูลแบบทางอ้อม สำหรับรายละเอียดต่างๆ ที่อธิบายในบทนี้มีดังนี้

- 2.1 คุณสมบัติ
- 2.2 สัญญาณขา และรายละเอียดของสัญญาณ
- 2.3 สถาปัตยกรรม
- 2.4 ฟังก์ชันบล็อกไดอะแกรม (Function Block Diagram)
- 2.5 การจัดหน่วยความจำ
- 2.6 หน่วยประมวลผลทางคณิตศาสตร์
- 2.7 การควบคุมระบบ
- 2.8 ฟังก์ชันอินพุทและเอาต์พุท
- 2.9 การขออินเทอร์พ (Interrupt)

## 2.1 คุณสมบัติ

ไอซี TMS32010 มีคุณสมบัติดังต่อไปนี้

- วัฏจักรการทำงานต่อ 1 คำสั่ง จะใช้เวลาดังนี้:

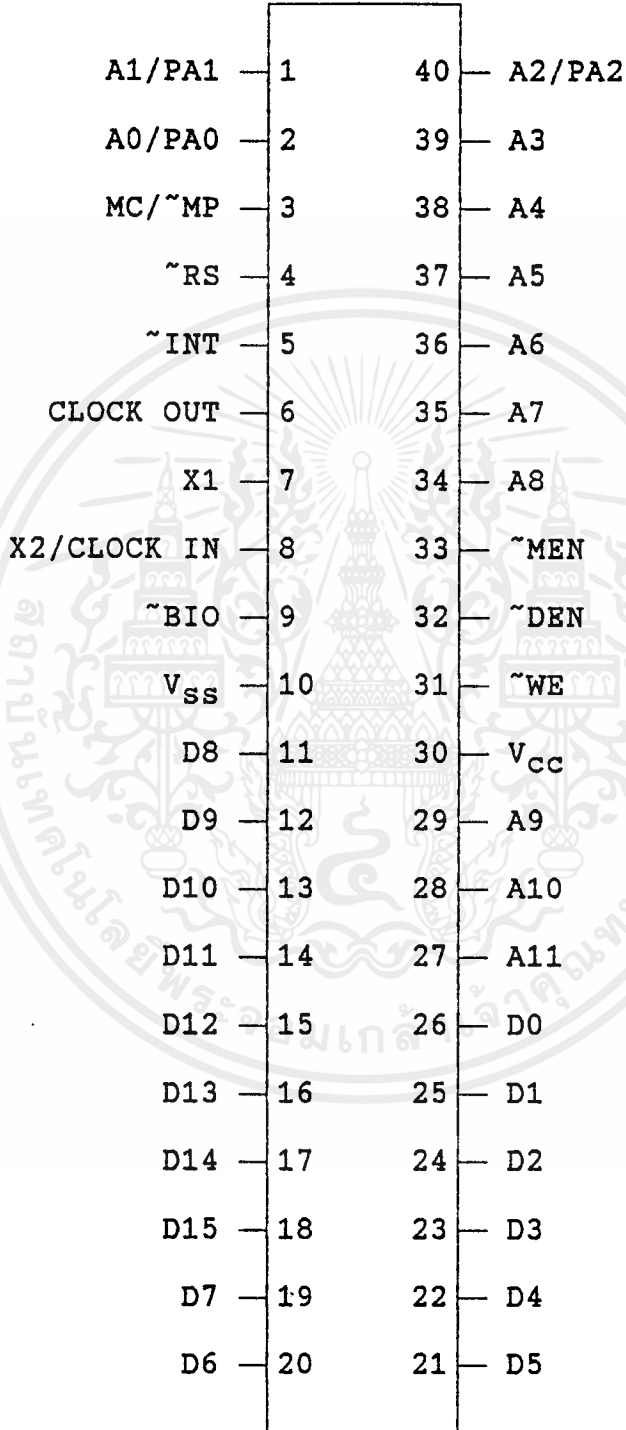
160 nS สำหรับ TMS32010-25/C10-25/C15-25/C17-25

200 nS สำหรับ TMS32010/C10/C11/C15/E15/C17/E17

280 nS สำหรับ TMS32010-14

- มีหน่วยความจำแบบแรม (RAM) ภายในไอซีขนาด 144/256 เวิร์ด (Words)
- มีหน่วยความจำแบบรอม (ROM) ภายในไอซีขนาด 1.5/4 กิโลเวิร์ด
- สามารถต่อหน่วยความจำภายนอกเพิ่มได้ถึง 4 กิโลเวิร์ด
- มีบัสข้อมูลแบบสองทาง ขนาด 16 บิต และมีความเร็วในการรับส่งข้อมูล 50 เมกะไบท์ต่อวินาที (Megabyte per second:Mbps)
- มีหน่วยประมวลผลทางคณิตศาสตร์ (Arithmetic Logic Unit:ALU) และรีจิสเตอร์แอดคิวมูลเอเตอร์ (Accumulator Resister) ขนาด 32 บิต
- สามารถทำคำสั่งในการคูณเลขขนาด 16 บิต 2 ชุด ภายใน 1 วัฏจักรการทำงาน และได้ผลลัพธ์เป็นเลขขนาด 32 บิต
- มีตัวเลื่อน (Shifter) ขนาด 0-16 บิต
- มีตัวกำเนิดสัญญาณนาฬิกา (Clock) ภายในตัวเอง
- มีพอร์ต (Port) สำหรับติดต่อสำหรับติดต่ออุปกรณ์ภายนอกได้ 8 พอร์ต
- มีไมโครโปรเซสเซอร์ร่วม สำหรับ TMS320C17/E17
- ใช้งานเลี้ยงไอซีขนาด 5 โวลท์
- เป็นไอซีขนาด 40 ขา
- ผลิตโดยเทคโนโลยีแบบ NMOS สำหรับ TMS32010/11 และแบบ CMOS สำหรับ TMS320C10/C15/E15/C17/E17

2.2 สัญญาณขา และรายละเอียดของสัญญาณ



รูปที่ 2.1 แสดงขาของไอซี เบอร์ TMS32010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับรายละเอียดของสัญญาณขาของ TMS32010 เป็นดังตารางที่ 2.1

สัญญาณ	สถานะ	รายละเอียด
A11 MSB A10 A9 A8 A7 A6 A5 A4 A3 A2/PA2 A1/PA1 A0/PA0	0	เป็นแอดเดรสบัสขนาด 12 บิต จาก A11(บิตสูง) ถึง A0 (บิตต่ำ) และเป็นพอร์ทแอดเดรสขนาด 3 บิต จากบิตPA2 ถึง PA0 บกติบัสแอดเดรสจะทำงานอยู่เสมอ ยกเว้นขณะกำลังทำงานในคำสั่ง IN และ OUT บิต A2 ถึง A0 จะถูกใช้เพื่อพอร์ทแอดเดรส
D15 MSB D14 D13 D12 D11 D10 D9 D8 D7 D6 D5	I/O/Z	เป็นบัสข้อมูลแบบขนาน ขนาด 16 บิต บัสข้อมูลจะอยู่ในสภาวะไฮอิมพีแดนซ์เสมอ ยกเว้นในช่วงที่ ~WEทำงาน (เป็นลอจิก(logic)"0")

D5 นี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D4 D3 D2 D1 D0 LSB		
$\sim$ BIO	I	เป็นอินพุตที่ถูควบคุมโดยคำสั่ง BIOZ ถ้ามีลอจิก "0" จะ าซูปกรณ์ที่แอดเดรสระบุ
$\sim$ DEN	O	เป็นตัวกำหนดการรับข้อมูลจากอุปกรณ์อินพุตเมื่อมีลอจิก"0" จะทำการรับข้อมูลจากบัสข้อมูล และจะมีลอจิก"0"เฉพาะ ในช่วงวัฏจักรแรกของคำสั่ง IN เท่านั้น อนึ่งขณะที่ $\sim$ DEN มีลอจิก"0"นั้น $\sim$ WE และ $\sim$ MENจะไม่ทำงาน(ลอจิก"1")
$\sim$ INT	I	เป็นสัญญาณอินเทอร์รัพ (Interrupt Signal) ทำงานที่ ขอบขาลง (Negative-going Edge)
MC/ $\sim$ MP	I	เป็นขาที่าใช้เลือกโหมดของหน่วยความจำ ที่ลอจิก"1"จะาใช้ เป็นโหมดไมโครคอมพิวเตอร์ (Microcomputer) ซึ่งจะ าใช้หน่วยความจำภายใน 1.5 กิโลเฮิร์ต และต่อหน่วย ความจำภายนอกได้ 2.5 กิโลเฮิร์ต ส่วนโหมดไมโครโปร เซสเซอร์(ลอจิก"0") จะาใช้เพียงหน่วยความจำภายนอก ขนาด 4 กิโลเฮิร์ต มีแอดเดรสตั้งแต่ 0 ถึง 4095
$\sim$ MEN	O	เป็นสัญญาณขอาใช้หน่วยความจำ จะทำงาน(ลอจิก"0")ใน ทุกรอบของแมชชีนไซเคิล(Machine Cycle) ยกเว้นเมื่อ $\sim$ WEและ $\sim$ DENทำงาน สัญญาณนี้จะถูควบคุมโดยคำสั่งเพท (Fetch) จากโปรแกรมนหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต  
 ใว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\sim$ RS	I	เป็นสัญญาณรีเซ็ต(Reset) ทำงานเมื่อลอจิก"0" มีค่าค้างอยู่ 5 สัญญาณนาฬิกา เมื่อทำงานแล้วจะมีผลทำให้ $\sim$ DEN, $\sim$ WE, $\sim$ MEN ถูกบังคับให้มีลอจิก "1" บัสข้อมูลเป็นไอเอ็มพีแดนซ์ แอดเดรสบัสและ โปรแกรมเคาท์เตอร์ (Program Counter) ถูกเคลียร์
$\sim$ WE	O	เป็นสัญญาณการเขียนข้อมูลเอาต์พุต ซึ่งจะทำงานที่ลอจิก "0" ข้อมูลจะถูกส่งจากอุปกรณ์เอาต์พุต มายังบัสข้อมูล สัญญาณนี้จะทำงานเฉพาะในคำสั่ง OUT และในคำสั่ง TBLW
CLKOUT	O	สัญญาณนาฬิกาเอาต์พุตของระบบ ซึ่งมีความถี่เท่ากับ 1 ใน 4 ของความถี่ CLKIN
V <sub>CC</sub>	I	ไฟเลี้ยง 5 โวลต์
V <sub>SS</sub>	I	กราวด์
X1	O	ขาคริสตอล เอาต์พุตสำหรับตัวกำเนิดความถี่ภายใน
X2/CLKIN	I	เป็นขาอินพุตให้แกตัวกำเนิดความถี่ทั้งภายในและภายนอก

( I : อินพุต , O : เอาต์พุต , Z : ไอเอ็มพีแดนซ์ (High-impedance) )

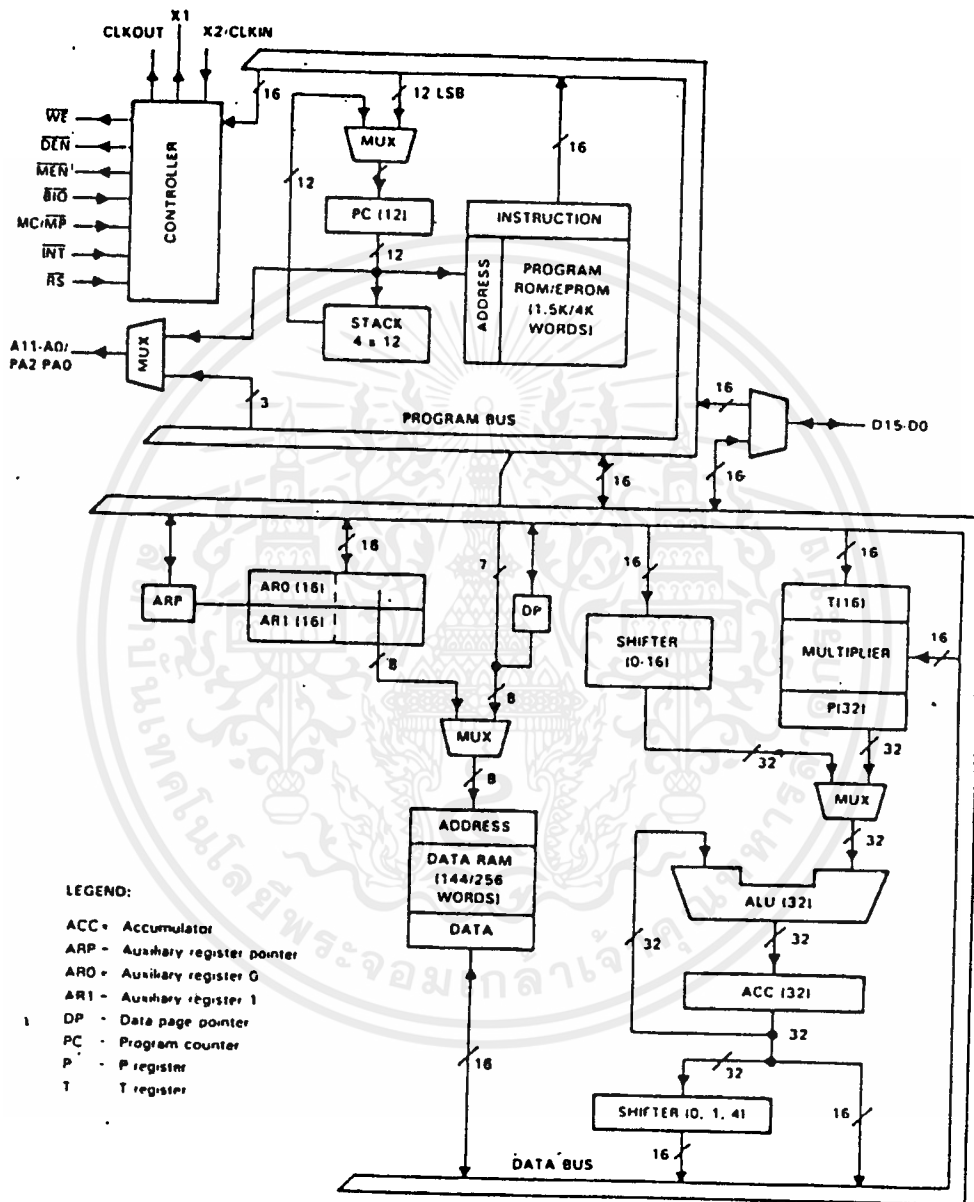
### 2.3 สถาปัตยกรรม

ดังที่ได้กล่าวมาแล้วว่า TMS32010 ออกแบบโดยใช้เทคโนโลยีของฮาร์ดแวร์ เพราะต้องการความเร็ว และวงจรมีความซับซ้อน เทคโนโลยีแบบนี้ หน่วยความจำที่เก็บ โปรแกรม และหน่วยความจำที่เก็บข้อมูลจะแยกกันอยู่คนละที่ จึงสามารถใช้คำสั่ง เพท และ คำสั่ง เอกซิดิวพร้อมกันได้ รวมทั้งยังสามารถรับส่ง ระหว่างโปรแกรมและข้อมูลได้ สามารถใช้โปรแกรมที่ถูกเก็บอยู่ในหน่วยความจำแบบแรมมาได้ ซึ่งจะเป็นความสะดวกมากขึ้น เนื่องจากไม่ต้องใช้หน่วยความจำแบบรอม TMS32010 ประกอบด้วยหน่วยประมวลผล ทางคณิตศาสตร์และแอดดิทิวเลเตอร์ขนาด 32 บิต เพื่อใช้ในการคำนวณที่ต้องการความ แม่นยำแบบ ดับเบิลเพรซิซัน(Double-Precision), ทุ-คอมพลีเมนต์ อะริทเมติก (Two's Complement Arithmetic) หน่วยประมวลผลทางคณิตศาสตร์จะประมวลผล ข้อมูลขนาด 16 บิต และสามารถทำงานแบบบูลีน (Boolean) ส่วนแอดดิทิวเลเตอร์จะมี ขนาด 32 บิต ทำหน้าที่เก็บข้อมูลจากหน่วยประมวลผลทางคณิตศาสตร์ และยังสามารถ ส่งข้อมูลกลับไปให้หน่วยประมวลผลทางคณิตศาสตร์ แอดดิทิวเลเตอร์นี้ถูกแบ่งเป็นสอง เวิร์ด คือ เวิร์ดสูง (บิต 31 ถึง บิต 16) และ เวิร์ดต่ำ (บิต 15 ถึง บิต 0) มัลติพลายเออร์ (Multiplier) จะใช้สำหรับการคูณเลข 16 บิต และได้ผลออกมาเป็น เลขขนาด 32 บิต ภายใน 1 วัชเคิลการทำงาน มัลติพลายเออร์จะประกอบด้วย 3 ส่วน ได้แก่ รีจิสเตอร์ T ,รีจิสเตอร์ P และ มัลติพลายเออร์อะเรย์ (Multiplier Array) บารีลชิฟเตอร์ จะทำการชิฟทางซ้าย 0 ถึง 16 บิต สำหรับเวิร์ดข้อมูลที่อยู่ในเมมโมรี และเข้าไปเก็บไว้ใน ALU ส่วนในแอดดิทิวเลเตอร์สามารถชิฟทางซ้ายได้ 0,1,4 บิต และเก็บผลไว้ในแอดดิทิวเลเตอร์เวิร์ดสูง นอกจากนี้ยังมีสแตค (Stack) 4 ระดับ สำหรับเก็บค่าโปรแกรมเคาน์เตอร์ ระหว่างการขออินเทอร์รัพ และการเรียกใช้ซับรูทีน (Sub Routine)

### 2.4 ฟังก์ชันบล็อกไดอะแกรม

ฟังก์ชันบล็อกไดอะแกรมที่แสดงในบทนี้ จะแสดงถึงการทำงานของ TMS32010 ทาง เติ่นต่างๆ ใน TMS และ ขาสัญญาณสำหรับอินเทอร์เฟสกับอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 แสดงผังชั้นบล็อกไออะแกรมของ TMS32010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.2 รายละเอียดของฮาร์ดแวร์ภายใน TMS32010

ชื่อเรียก	สัญลักษณ์	ลักษณะและการทำงาน
Accumulator	ACC	เป็นแอดดิทิวเมเตอร์ 32 บิต แบ่งเป็นสองส่วน คือ เวิร์ดสูง (บิตที่ 16 ถึง 31) และเวิร์ดต่ำ (บิตที่ 0 ถึง 15) ใช้สำหรับเก็บผลลัพธ์จาก ALU
Arithmetic logic Unit	ALU	เป็น ALU 32 บิต แบบทู-คอมพลีเมนต์ มี input port และ output port 32 บิต ที่เชื่อมต่อกับ ACC
Auxiliary register	ARO, AR1	16 บิตรีจิสเตอร์ เก็บค่าเพื่อการเข้าถึง Address และค่าสำหรับการนับลูบ 9 บิตท้ายของแต่ละรีจิสเตอร์ ทำหน้าที่ Counter ให้นับขึ้นหรือลงได้
Auxiliary register Pointer Central	ARP	เป็นรีจิสเตอร์ 1 บิต สำหรับบอกสถานะการเลือกใช้งาน ARO หรือ AR1
Arithmetic Logic Unit	CALU	กลุ่มของ ALU, ตัวคูณ, ACC และ shifter

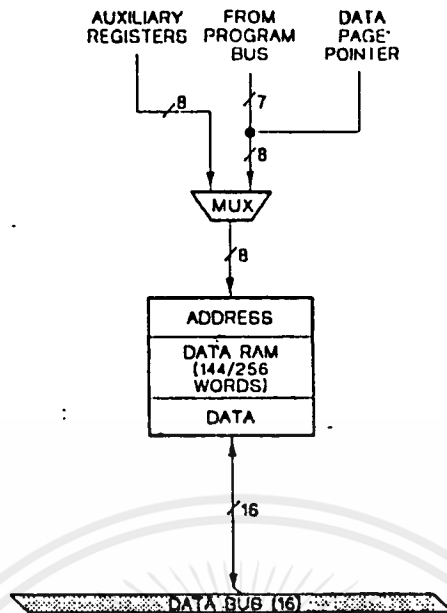
Data Memory Page Pointer	DP	รีจิสเตอร์ขนาด 1 บิต บอกการเลือก Page ของ Data Memory Address เป็น 8 บิต เพื่อชี้ไปยัง Address
Data RAM	—	ดาต้าแรมภายในชิพซึ่งมี 144 หรือ 256 เวิร์ด
Interrupt Flag	INTF	แฟลกซ์ขนาด 1 บิต บอกการขออินเทอร์รัพ
Interrupt Mode	INTM	สแตตัสบิต ซึ่งเป็นเครื่องหมายให้อินเทอร์รัพ-แฟลกซ์
Multiplier	MULT	ตัวคูณแบบ parallel 16*16 บิต
Overflow Flag	OV	สแตตัสบิต บอกการเกิด overflow ของการคำนวณเลข
Overflow Mode	OVM	สแตตัสบิต บอกการ saturate ของการคำนวณผลลบ
P Register	P	รีจิสเตอร์ 32 บิต เก็บผลการคูณ
Program Bus	P(15-0)	16 บิตบัส ของโปรแกรมเมมโมรี่
Port Address	PA(2-0)	Port Address ขนาด 3 บิต (เป็น 3 บิตท้ายของ Address) สำหรับโหมด I/O
Program Counter	PC(11-0)	รีจิสเตอร์ 12 บิต เก็บค่า Address ของโปรแกรมเมมโมรี่ สำหรับคำสั่งถัดไป

Shifter	-	มีสองตัวคือ บาร์เรลชิฟเตอร์ 16 บิต สำหรับข้อมูลของดาต้าเข้า ALU และพาราเรล-ชิฟเตอร์สำหรับ shift ข้อมูล 0,1,4 ตำแหน่ง ทำให้ ACC ท้าให้เกิดบิตสูงในดาต้าแรม
Stack	-	อาร์ดแวย์แสดขนาด 4*12 เก็บค่า PC ระหว่างการอินเทอร์รัพ หรือการ call
Status Register	ST	สแตตัสรีจิสเตอร์ขนาด 16 บิต
T Register	T	รีจิสเตอร์ 16 บิต เก็บตัวทวนการคูณ

## 2.5 การจัดหน่วยความจำ

ในหัวข้อนี้จะทำให้รายละเอียดเกี่ยวกับ หน่วยความจำโปรแกรม , หน่วยความจำข้อมูล การจัดผังของหน่วยความจำ ออกซิลารี่รีจิสเตอร์ และ โหมดการอ้างหน่วยความจำ หน่วยความจำข้อมูล

หน่วยความจำข้อมูลประกอบด้วยแรม 144/256 เวิร์ด 16 บิตบนชิพ (on chip RAM) ดังรูป 2.3 ในกรณีที่ใช้หน่วยความจำข้อมูลแบบขยาย ข้อมูลจะถูกดำเนินการบนออฟชิพแรม (off chip RAM) และจะถูกอ่านเข้าไปในออนชิพแรมเมื่อต้องการคำสั่งที่ใช้คือ TBLR/TBLW และ IN/OUT ตัวอย่างเช่น คำสั่ง TBLW จะเป็นการส่งข้อมูลจากหน่วยความจำภายนอก เข้ามายังหน่วยความจำข้อมูล และคำสั่ง IN จะเป็นการอ่านข้อมูลจากพอร์ทภายนอก เข้ามายังหน่วยความจำข้อมูลภายใน



รูปที่ 2.3 แสดงหน่วยความจำข้อมูลแบบออนชิพ

### หน่วยความจำโปรแกรม

เป็นหน่วยความจำที่ใช้เก็บชุดของคำสั่ง โดยสำหรับใน TMS32010 จะมีหน่วยความจำภายใน หรือที่เรียกว่า "มาสค์รอม" (Mask ROM) มีขนาด 1.5 กิโลไบต์

ในการใช้งานหน่วยความจำโปรแกรม เราสามารถเลือกได้ 2 โหมด คือ โหมดไมโครคอมพิวเตอร์ และ โหมดไมโครโปรเซสเซอร์ โดยในการออกแบบวงจรเราสามารถเลือกได้จากขา MC/~MP

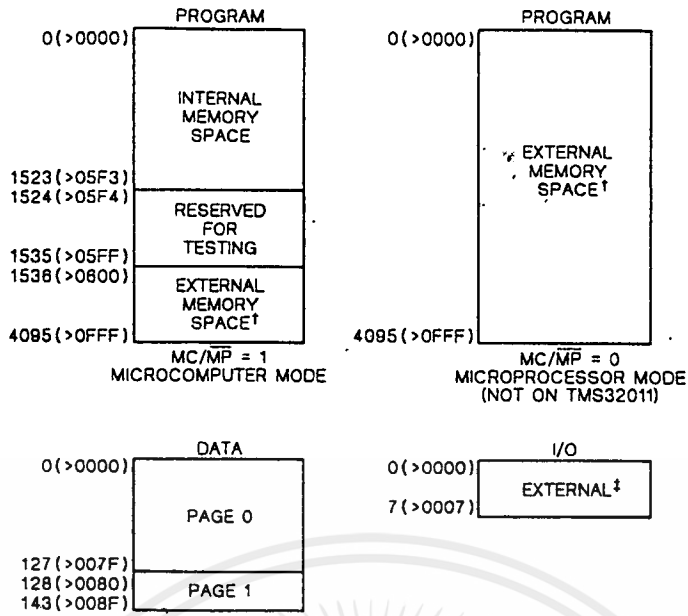
- โหมดไมโครคอมพิวเตอร์: จะใช้หน่วยความจำรอมภายในโดยเริ่มจากแอดเดรสที่ 0 ถึง 1523 แต่แอดเดรสที่ 1524 ถึงแอดเดรสที่ 1535 จะสงวนไว้โดยทางบริษัท และจากแอดเดรสที่ 1536 ถึงแอดเดรสที่ 4095 จะเป็นหน่วยความจำที่เราต้องต่อเพิ่มไว้ภายนอก เป็นขนาด 2.5 กิโลไบต์

- โหมดไมโครโปรเซสเซอร์: จะใช้หน่วยความจำภายนอกทั้งหมด 4 กิโลไบต์ ตั้งแต่แอดเดรสที่ 0 จนถึง แอดเดรสที่ 4095

### การจัดหน่วยความจำ

TMS32010 ได้ทำการแบ่งส่วนแอดเดรสไว้ 3 ส่วน สำหรับเก็บหน่วยความจำโปรแกรม หน่วยความจำข้อมูล และอินพุท/เอาต์พุท ดังแสดงในรูป 2.4 ซึ่งแสดงการจัดเก็บทั้ง 2 โหมด

มีให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



†Not available on the TMS32011

‡Ports 0 and 1 dedicated to internal control register on TMS32011.

## รูปที่ 2.4 แสดงลักษณะการจัดหน่วยความจำภายใน TMS32010

### ออกซิลลารีรีจิสเตอร์

ใน TMS32010 จะมีรีจิสเตอร์ประเภทหนึ่งที่เรียกว่ารีจิสเตอร์ออกซิลลารี หรือ AR ซึ่งแบ่งออกเป็น 2 ตัว คือ AR0 และ AR1 รีจิสเตอร์ AR มักใช้ในการอ้างหน่วยความจำแบบทางอ้อม (Indirect Addressing Mode) สำหรับคำสั่งที่เกี่ยวกับการอ้างหน่วยความจำแบบทางอ้อมนั้น จะต้องใส่โอเปอเรนด์ (Operand) ซึ่งเป็นตำแหน่งของหน่วยความจำของข้อมูลที่ 8 บิตท้ายของ AR ด้วย

เนื่องจากรีจิสเตอร์ AR มีอยู่ 2 ตัว ดังนั้นเราสามารถเลือกเข้าได้โดย ออกซิลลารีรีจิสเตอร์พอยน์เตอร์ (Auxillary Register Pointer: ARP) ซึ่งจะเป็นบิตหนึ่งภายในรีจิสเตอร์บอกสถานะ (Status Register)

### โหมดการอ้างหน่วยความจำ

วิธีการอ้างหน่วยความจำของ TMS32010 สามารถแบ่งออกได้เป็น 3 แบบคือ

1) การอ้างหน่วยความจำโดยตรง (Direct Addressing Mode)

เอกสารสำหรับการอ้างแอดเดรสของหน่วยความจำโดยตรงนั้นภายในคำสั่งจะประกอับไปด้วยแอดเดรสของหน่วยความจำข้อมูล (Data memory Address : DMA) ที่ 7 บิตท้าย

ของตัวคำสั่ง และบิตที่ 7 จะเป็นค่าพอยน์เตอร์ที่ชี้บอกหน้าของหน่วยความจำข้อมูล (Data Memory Page Point : DP) ซึ่งจะเป็นตัวเลือกหน้า ของหน่วยความจำข้อมูล

การอ้างแอดเดรสของหน่วยความจำโดยตรงสามารถทำได้สำหรับทุกคำสั่ง ยกเว้น คำสั่งต่อไปนี้

- CALL
- BANZ
- คำสั่งที่นำค่าโอเปอร์แรนด์ทันที (Immediate Operand)
- คำสั่งที่นำมีโอเปอร์แรนด์

2) การอ้างแอดเดรสของหน่วยความจำโดยทางอ้อม (Indirect Addressing Mode)

จะเป็นการอ้างถึงตำแหน่งของหน่วยความจำข้อมูลที่ 7 บิตล่างของคำสั่ง และที่บิตที่ 7 จะเป็นค่าของ ARP การอ้างแอดเดรสของหน่วยความจำโดยทางอ้อม จะไม่มีการเลือกหน้าของหน่วยความจำข้อมูล ARP ก็จะเป็นตัวเลือกว่าเป็น ARO หรือ AR1 AR สามารถเพิ่มหรือลดลงได้โดยอัตโนมัติ เมื่อเราได้มีการนำคำสั่ง ที่อ้างแอดเดรสของหน่วยความจำโดยทางอ้อม

สำหรับการนำคำสั่งประเภทอ้างแอดเดรสของหน่วยความจำโดยทางอ้อม จะมี AR อยู่ในคำสั่งเป็นจำนวน 9 บิต ซึ่งสามารถถูกไหลได้ โดย คำสั่ง LAR , LARK และสามารถเปลี่ยนค่าได้โดยคำสั่ง MAR

การอ้างแอดเดรสของหน่วยความจำโดยทางอ้อม จะใช้สัญลักษณ์ดังต่อไปนี้ สำหรับการเขียนภาษาแอสเซมบลีของ TMS32010

- "\*" : จะใช้ค่าของ AR สำหรับชี้ตำแหน่งหน่วยความจำข้อมูล
- "\*" : จะใช้ค่าของ AR สำหรับชี้ตำแหน่งหน่วยความจำข้อมูล และจะลดค่าของ AR ลงไป 1 หลังจากทำคำสั่งเสร็จ
- "\*" : จะใช้ค่าของ AR สำหรับชี้ตำแหน่งหน่วยความจำข้อมูล และจะเพิ่มค่าของ AR ขึ้นไป 1 หลังจากทำคำสั่งเสร็จ

3) การอ้างแอดเดรสของหน่วยความจำโดยทันที (Immediate Addressing Mode)

ในชุดคำสั่งของ TMS32010 จะมีอยู่ 5 คำสั่ง ที่ใช้ในการแอดเดรสของหน่วยความจำโดยทันที ซึ่งคำสั่งเหล่านี้จะทำงานได้ภายใน 1 วัฏจักรคำสั่ง โดยคำสั่งเหล่านี้

ได้แก่

LARK

LACK

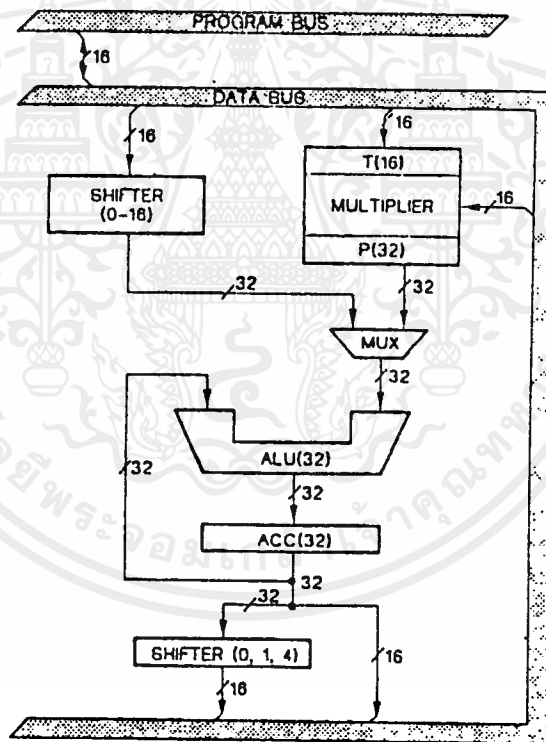
LARP

LDPK

MPYK

### 2.6 หน่วยประมวลผลทางคณิตศาสตร์กลาง

หน่วยประมวลผลทางคณิตศาสตร์กลาง ประกอบด้วยหน่วยการคูณเลข 16 บิต แบบขนาน หน่วยประมวลผลทางคณิตศาสตร์ แอดดีทิวเมเตอร์ขนาด 32 บิต และตัวเลื่อนข้อมูล 2 ชุด ซึ่งบล็อกไดอะแกรมแสดง ในรูปที่ 2.5



รูปที่ 2.5 แสดงบล็อกไดอะแกรม ของหน่วยประมวลผลทางคณิตศาสตร์กลาง (CALU)

### 2.7 การควบคุมระบบ

การควบคุมระบบของ TMS32010 ทำโดยใช้ โปรแกรมเคาท์เตอร์ แสดงสัญญาณรีเซ็ตจากภายนอก สัญญาณอินเทอร์รัพ และสแตตัสรีจิสเตอร์  
 เอกสารนี้เป็นเอกสารลิขสิทธิ์ของ Texas Instruments เพื่อใช้ในการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โปรแกรมเดาทเตอร์และ สแตก จะทำงานในคำสั่งที่เกี่ยวกับการเรียกโปรแกรมย่อย การอินเทอร์รัพ และคำสั่งอ่าน/เขียนตาราง โปรแกรมเดาทเตอร์ซึ่งมี 12 บิต จะทำการเก็บค่าแอดเดรสของหน่วยความจำโปรแกรม ที่กำลังทำงานอยู่ TMS32010 จะทำการอ่านหน่วยความจำโปรแกรมในแอดเดรสที่โปรแกรมเดาทเตอร์ระบุ และทำการเพิ่มค่าโปรแกรมเดาทเตอร์บางบิต สำหรับการเพทคำสั่งต่อไป โปรแกรมเดาทเตอร์จะมีค่าเป็นศูนย์ เมื่อสัญญาณรีเซ็ตทำงาน

- สัญญาณรีเซ็ต เป็นการอินเทอร์รัพแบบ นอน-มาสเคเบิล (Non-maskable) เพื่อเป็นการทำให้ค่าเริ่มต้นแก่ TMS32010 การรีเซ็ต TMS32010 จะเกิดผลตามมาดังนี้

- สัญญาณ ~DEN , ~WE และ ~MEN จะมีค่าเป็นระดับลอจิก "1"
- บัสข้อมูลจะเป็นไฮอิมพีแดนซ์ โปรแกรมเดาทเตอร์จะถูกเซ็ตเป็นศูนย์ และแอดเดรสบัส (A11-A0) จะมีค่าเป็นศูนย์ทั้งหมด หลังจากสัญญาณนาฬิกาถูกถัดไปเมื่อสัญญาณรีเซ็ตเป็นระดับลอจิก "0"

- ไม่สามารถขออินเทอร์รัพได้ และอินเทอร์รัพแฟลกรีจิสเตอร์ ถูกเซ็ตเป็นศูนย์

- สเตตัสรีจิสเตอร์ ประกอบด้วยรีจิสเตอร์บอกสถานะ 5 บิต ได้แก่ ออกซิลลารี-รีจิสเตอร์ (ARP) , พอยเตอร์บอกหน้าหน่วยความจำข้อมูล (DP) , อินเทอร์รัพโหมดบิต (INTM) , โอเวอร์โพลว์แฟลก (OV) และ โอเวอร์โพลว์โหมดบิต (OVM) แสดงดังในรูปที่ 2.6

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OV	OVM	INTM	1	1	1	1	ARP	1	1	1	1	1	1	0	DP

รูปที่ 2.6 แสดงรีจิสเตอร์บอกสถานะของ TMS32010

รีจิสเตอร์บอกสถานะสามารถเก็บเข้า และโหลด จากหน่วยความจำได้โดยคำสั่ง SST และ LST ตามลำดับ

ARP ใช้สำหรับการอ้างแอดเดรสของหน่วยความจำโดยทางอ้อม สำหรับบิตนี้จะมีผลเมื่อเกิดการอ้างแอดเดรสของหน่วยความจำโดยทางอ้อม และ คำสั่ง LARP, MAR, LST ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DP เป็นพอยน์เตอร์ที่ชี้บอกหน้าของหน่วยความจำข้อมูล มักจะชี้ด้านคำสั่งที่มีการอ้างแอดเดรสของหน่วยความจำข้อมูลโดยตรง โดยจะไว้บออยู่ที่ 7 ของออบเจกต์ของคำสั่ง ถ้า DP มีค่าเป็น "0" ก็จะหมายถึงหน่วยความจำข้อมูล 128 เวิร์ดแรก หรือเรียกว่าหน้าที่ศูนย์ และถ้า DP มีค่าเป็น "1" ก็จะหมายถึงหน่วยความจำข้อมูลตำแหน่งที่ 128 จนถึงตำแหน่งที่ 143 หรือเรียกว่าเป็นหน้าที่หนึ่ง ค่าของ DP จะเปลี่ยนค่าได้โดยคำสั่ง LST , LDP และ LDPK

INTM ถ้าเราทำการอินทราเปลอินเทอร์รัพโดยทำให้บิตนี้เป็น "0" เมื่อเกิดการอินเทอร์รัพขึ้น ก็จะทำให้บิตนี้เป็น "1" (ก่อนที่จะเข้าโปรแกรมตอบสนองการอินเทอร์รัพ)

บิตนี้สามารถทำการเซ็ท และรีเซ็ทได้โดยคำสั่ง DINT และ EINT ตามลำดับ ส่วนคำสั่งการรีเซ็ท ที่ขา RS ก็จะมีผลทำให้บิตนี้มีค่าเป็น "1"

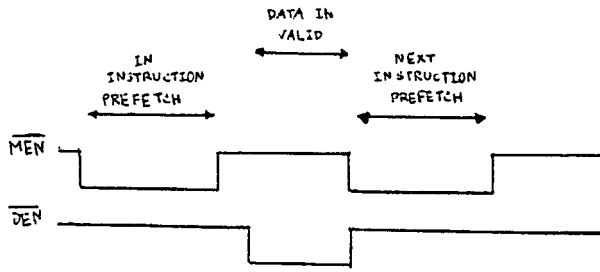
OV เป็นบิตบอกสถานะของการเกิดโอเวอร์โพล์ ถ้าบิตนี้เป็น "0" จะแสดงว่าแอดดิทิวเลเตอร์ไม่เกิดโอเวอร์โพล์ แต่ถ้าบิตนี้เป็น "1" จะแสดงว่าแอดดิทิวเลเตอร์เกิดการโอเวอร์โพล์ เราสามารถเคลียร์บิตนี้ได้โดยคำสั่ง BV , LST หรือการรีเซ็ทที่ขา  $\sim$ RS

OVM เป็นบิตบอกสถานะของโหมตโอเวอร์โพล์ ถ้าบิตนี้เป็น "0" จะเป็นการดีสเอเบิลการโอเวอร์โพล์ แต่ถ้าบิตนี้เป็น "1" ก็จะเป็นการอินทราเปลการโอเวอร์โพล์ เราสามารถเซ็ท หรือรีเซ็ทบิตนี้ได้โดยคำสั่ง SOVM และ ROVM ตามลำดับ นอกจากนี้คำสั่ง LST ก็มีผลต่อบิตนี้ด้วย

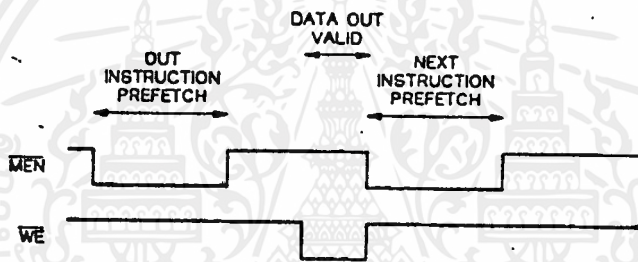
## 2.8 ฟังก์ชันอินพุทและเอาต์พุท

ฟังก์ชันอินพุท และเอาต์พุท ใช้สำหรับติดต่อกับอุปกรณ์ภายนอก ทางบัสข้อมูลขนาด 16 บิต โดยใช้คำสั่ง IN และ OUT พอร์ตอินพุทและเอาต์พุท จะอยู่ที่สามบิตต่ำของแอดเดรสบัส (PA2-PA0) ข้อมูลจะถูกรับ-ส่งระหว่างบัสข้อมูลและหน่วยความจำข้อมูล โดยคำสั่ง  $\sim$ DEN และ  $\sim$ WE บัสข้อมูลสองทิศทางภายนอกจะอยู่ในสภาวะไฮอิมพีแดนซ์เสมอ ยกเว้นเมื่อสัญญาณ  $\sim$ WE ทำงาน (ลอคจิก"0") หรือระหว่างคำสั่ง IN ,  $\sim$ WE จะทำงานระหว่างไฮเดิลแรกของคำสั่ง OUT และไฮเดิลที่สองของคำสั่ง TBLW และเนื่องจากแอดเดรสของพอร์ตมี 3 บิต จึงสามารถอ้างพอร์ตได้ทั้งหมด 8 พอร์ต และเพื่อความเข้าใจสำหรับการทำงานของอินพุทและเอาต์พุทจะแสดงดังรูป

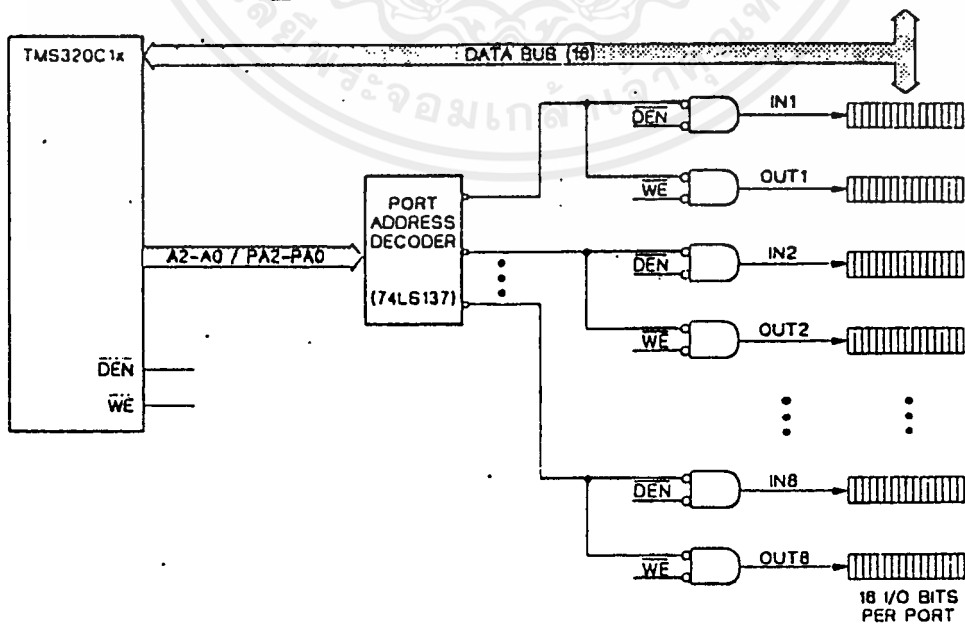
เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของบริษัทฯ เพื่อการวิจัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 จะแสดงไทม์มิงไดอะแกรม ของคำสั่งอินพุท



รูปที่ 2.8 จะแสดงไทม์มิงไดอะแกรม ของคำสั่งเอาท์พุท



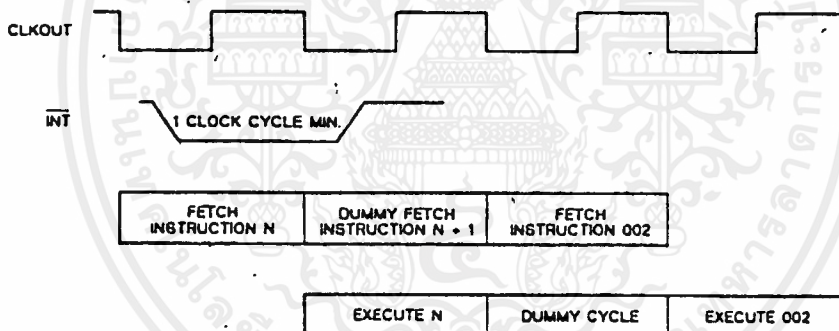
รูปที่ 2.9 แสดงการอินเทอร์เฟส กับอุปกรณ์ภายนอกของ TMS32010

## 2.9 การขออินเทอร์รัพ

สัญญาณอินเทอร์รัพ จะทำงานที่ขอบขาลงของระดับลอจิก เมื่อสัญญาณอินเทอร์รัพอยู่ในสถานะอีนาเบิล (enable) หรือยอมให้มีการอินเทอร์รัพได้ (INTM เป็นศูนย์) การอินเทอร์รัพจะขึ้นอยู่กับระดับลอจิกของขา  $\overline{INT}$  แต่ถ้าอยู่ในสถานะดิสเอเบิล (disable) INTM จะมีค่าเป็นลอจิก "1" จะไม่ยอมให้มีการอินเทอร์รัพว่าลอจิกของขา  $\overline{INT}$  จะเป็นค่าใดก็ตาม

การอินเทอร์รัพจะถูกหน่วงเวลาในกรณีต่อไปนี้

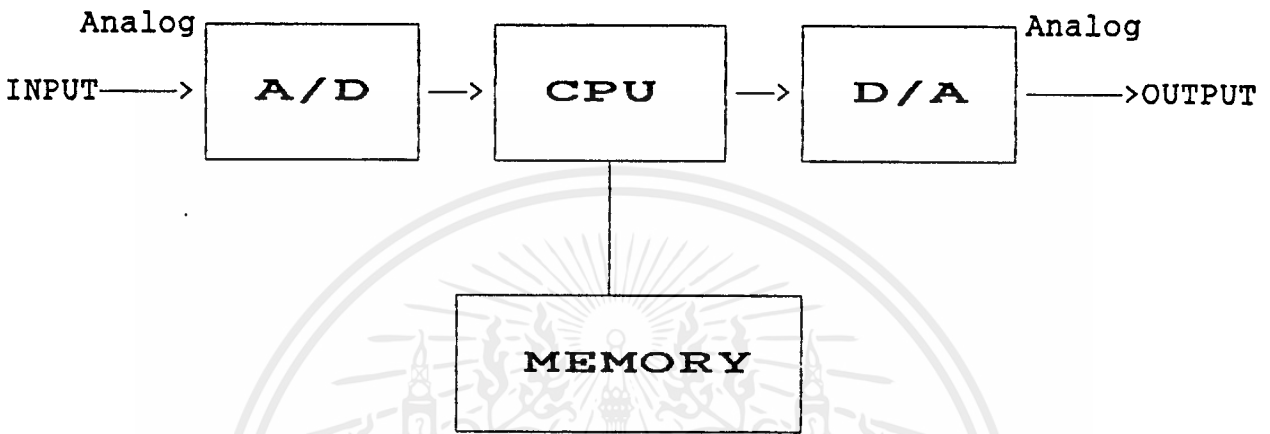
- จนกว่าจะจบคำสั่งสุดท้ายของไอซีเคิล
- จนกว่าคำสั่งที่ต่อจากคำสั่ง MPY หรือ MPYK จะทำงานเสร็จสมบูรณ์
- จนกว่าคำสั่งที่ต่อจากคำสั่ง RINT ทำงาน



รูปที่ 2.10 แสดงไทม์มิง ของการอินเทอร์รัพ

บทที่ 3 การออกแบบวงจรและการทำงาน

การออกแบบวงจร เราจะเริ่มต้นจากแผนผังเบื้องต้นของระบบประมวลผลสัญญาณดิจิตอลดังรูป



รูปที่ 3.1 ระบบประมวลผลสัญญาณดิจิตอล

จากแผนผังรูป 3.1 เราสามารถออกแบบโดยแยกวงจรออกเป็น 4 ส่วน ดังนี้

- ส่วนประมวลผล (CPU)
- ส่วนแปลงสัญญาณอนาลอกเป็นดิจิตอล (A/D)
- ส่วนแปลงสัญญาณดิจิตอลเป็นอนาลอก (D/A)
- ส่วนหน่วยความจำ (MEMORY)

**3.1 ส่วนประมวลผล**

าววงจรส่วนประมวลผลจะประกอบด้วยวงจรส่วนต่าง ๆ ดังนี้

**3.1.1 ตัวประมวลผล**

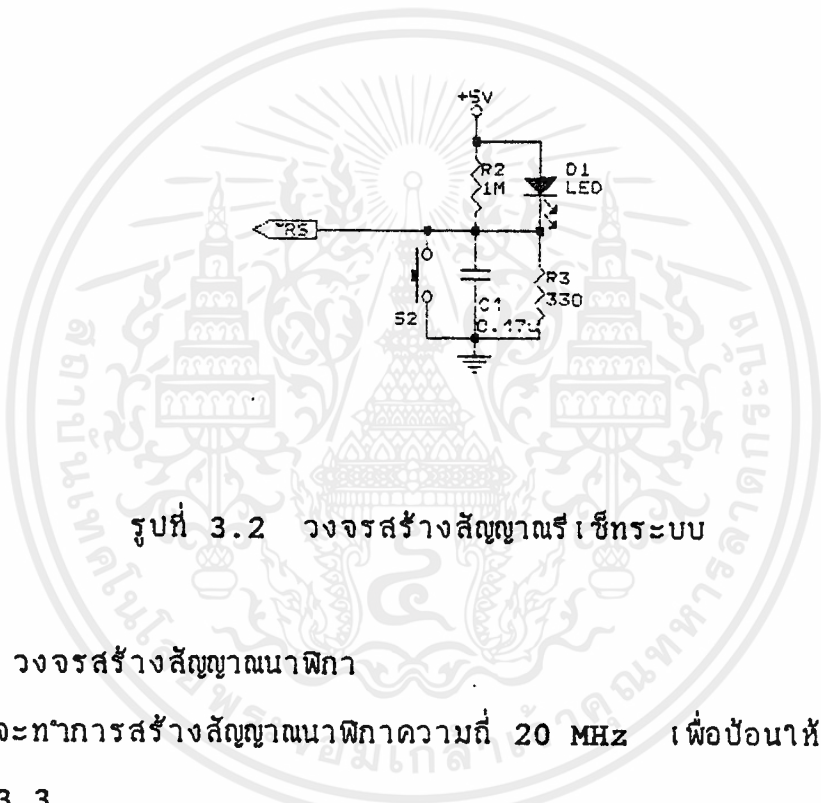
เราเลือกชิปไอซี TMS 32010 เป็นตัวประมวลผล ซึ่ง TMS 32010 จะมีหน่วยความจำภายในตัวเอง แต่เนื่องจากการใช้งานจริง การจะแก้ไขข้อมูลในหน่วยความจำภายในของ TMS 32010 ที่ทำได้ไม่สะดวก เราจึงออกแบบให้ใช้หน่วยความจำภายนอกแทน โดยเราจะเลือกให้ TMS 32010 ทำงานในโหมดไมโครโปรเซสเซอร์

### 3.1.2 วงจรสร้างสัญญาณรีเซ็ตระบบ

วงจรนี้จะทำการสร้างสัญญาณลอจิก 0 เพื่อรีเซ็ต TMS 32010 และทำให้ระบบกลับสู่จุดเริ่มต้น วงจรนี้จะทำงานก็ต่อเมื่อ

- เริ่มจ่ายไฟเลี้ยงให้กับระบบ (Power on Reset)
- กดปุ่มรีเซ็ต (Manual Reset)

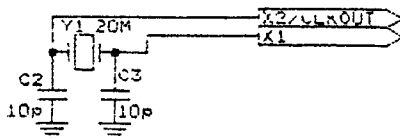
การทำงานของวงจร อาศัยหลักการทำงานของวงจร R-C Time Constant ดังรูปที่ 3.2



รูปที่ 3.2 วงจรสร้างสัญญาณรีเซ็ตระบบ

### 3.1.3 วงจรสร้างสัญญาณนาฬิกา

วงจรนี้จะทำการสร้างสัญญาณนาฬิกาความถี่ 20 MHz เพื่อป้อนให้กับ TMS 32010 วงจรดังรูปที่ 3.3

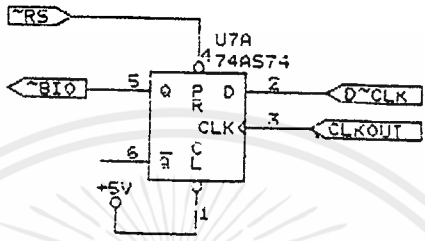


รูปที่ 3.3 วงจรสร้างสัญญาณนาฬิกาความถี่ 20 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 วงจรสร้างสัญญาณ  $\sim$ BIO

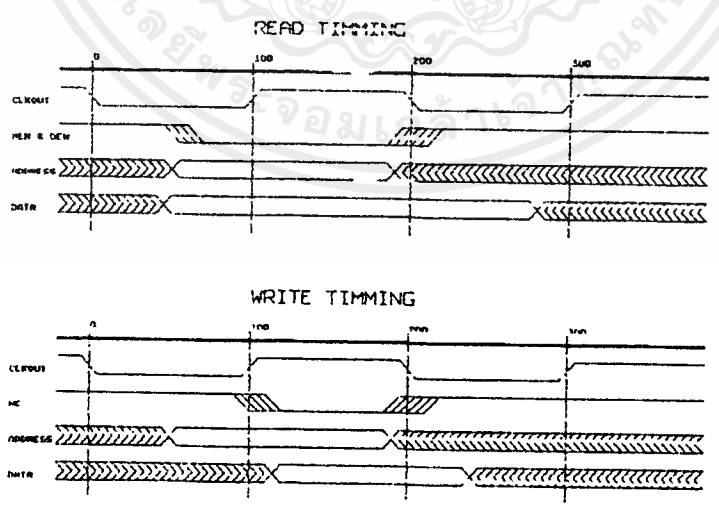
สัญญาณ  $\sim$ BIO เป็นสัญญาณอินพุทของ TMS 32010 ที่ส่งเข้าบเพื่อบอกให้ TMS 32010 รู้ว่ามีการส่งข้อมูลเข้ามาพร้อมแล้ว TMS32010 สามารถจะอ่านข้อมูลเข้า าบาเข้าได้ ในการออกแบบเราใช้สัญญาณนี้ ในการซิงโครไนซ์การทำงานของ TMS 32010 กับข้อมูลที่ส่งเข้ามา โดยาใช้วงจรดังรูปที่ 3.4



รูปที่ 3.4 วงจรสร้างสัญญาณ  $\sim$ BIO

3.1.5 การาใช้สัญญาณ  $\sim$ MEM ,  $\sim$ DEN ,  $\sim$ WE ในการอ่านและเขียนพอร์ทและ หน่วยความจา

จากการทำงานของ TMS 32010 ซึ่งมี Timing Diagram ดังรูปที่ 3.5



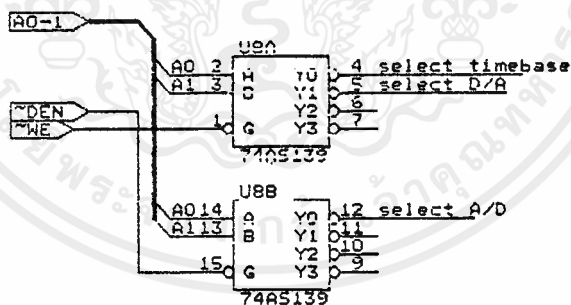
TMS32010  
TIMMING DIAGRAM

รูปที่ 3.5 Timing Diagram ของการอ่านและการเขียนหน่วยความจาและพอร์ท

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สัญญาณ  $\sim$ MEM จะเป็นลอจิก 0 เมื่อ TMS 32010 ต้องการจะอ่านหน่วยความจำ
- สัญญาณ  $\sim$ DEN จะเป็นลอจิก 0 เมื่อ TMS 32010 ต้องการจะอ่านพอร์ท
- สัญญาณ  $\sim$ WE จะเป็นลอจิก 0 เมื่อ TMS 32010 ต้องการจะเขียนหน่วยความจำหรือพอร์ท

เราสามารถแยกความแตกต่างของการเขียนหน่วยความจำและการเขียนพอร์ท โดยใช้ค่าแอดเดรส เนื่องจาก TMS 32010 มีพอร์ทอินพุทเอาต์พุท 8 พอร์ท ดังนั้นถ้าค่าแอดเดรสมากกว่า 8 แสดงว่า TMS32010 ต้องการเขียนข้อมูลลงในหน่วยความจำ แต่ถ้าค่าแอดเดรสน้อยกว่าหรือเท่ากับ 8 แสดงว่า TMS 32010 ต้องการส่งข้อมูลออกไปที่พอร์ท 1 ในวงจรมันนี้ออกแบบต้องการใช้พอร์ท 3 พอร์ท เป็นพอร์ทอินพุท 1 พอร์ท พอร์ทเอาต์พุท 2 พอร์ท จึงออกแบบไว้ดังรูปที่ 3.6

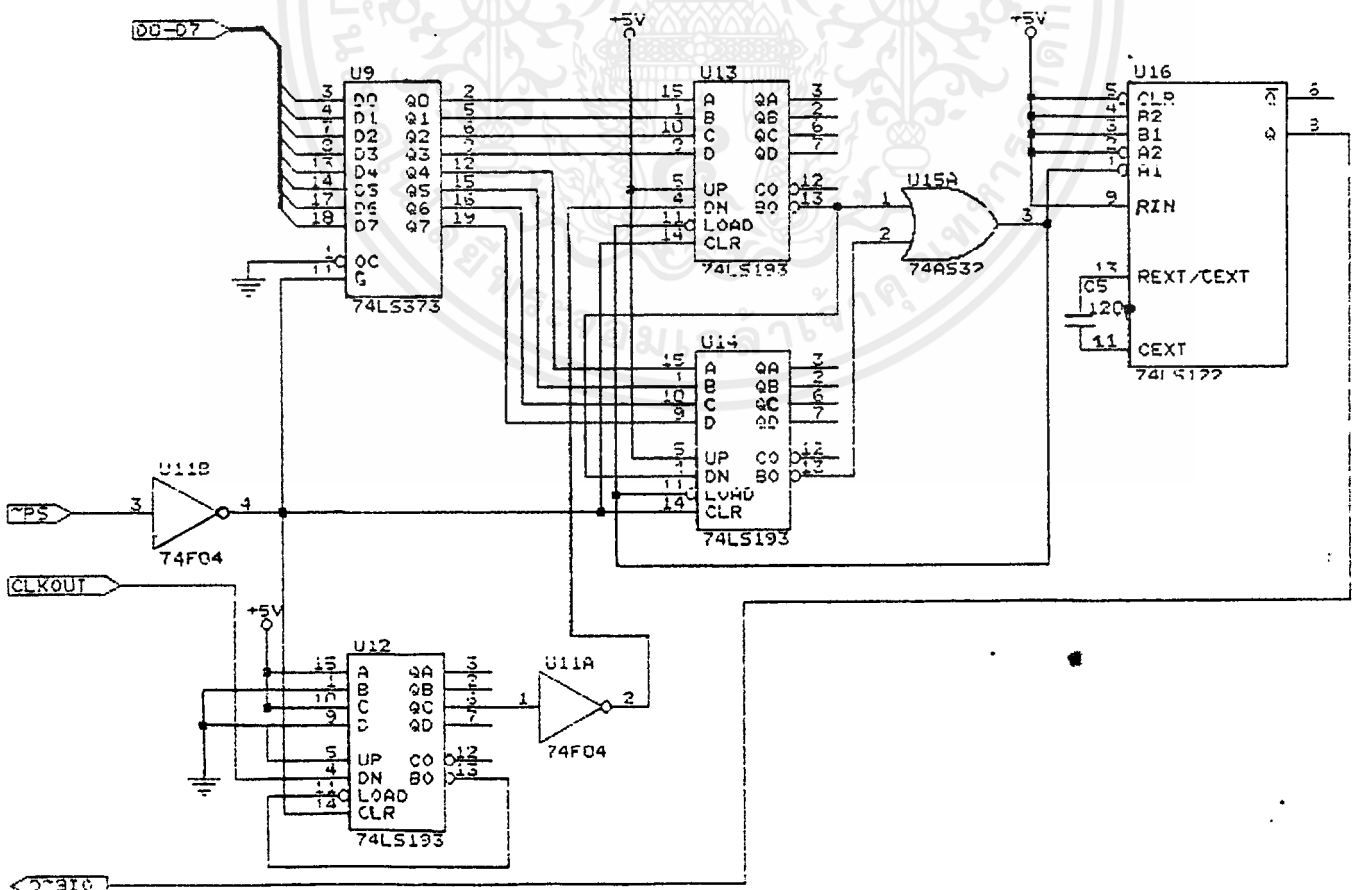


รูป 3.6 วงจร Decode พอร์ทอินพุท พอร์ทเอาต์พุท และหน่วยความจำ

3.1.6 วงจรกำหนดความถี่สุ่ม (Time Base)

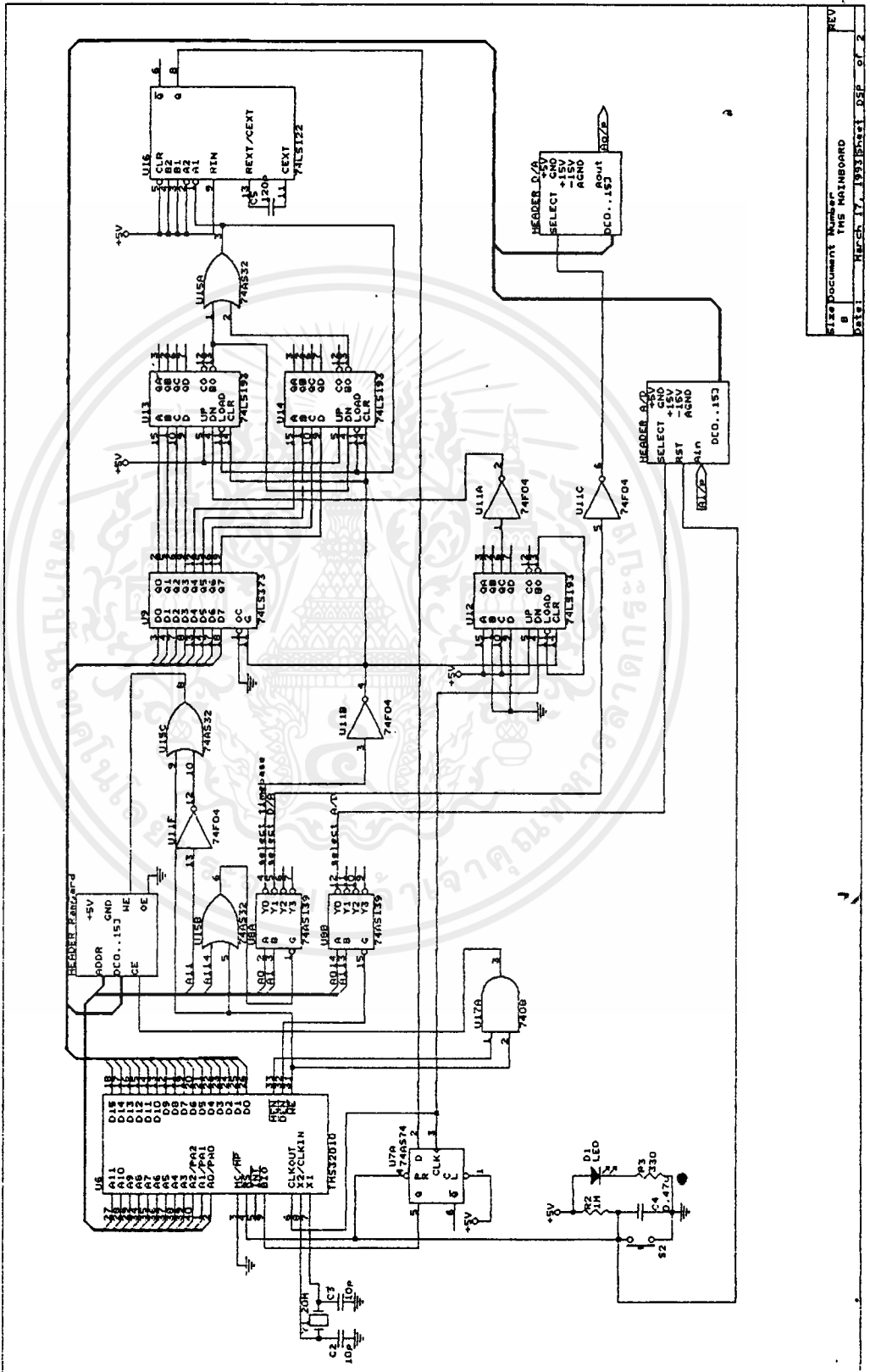
วงจรถูกออกแบบมาเพื่อใช้งานการกำหนดความถี่ที่ใช้งานการสุ่มข้อมูลให้กับ TMS32010 โดยอาศัยสัญญาณ  $\sim$ BIO ซึ่ง TMS 32010 สามารถตรวจสอบสัญญาณนี้ได้ การใช้งาน เราสามารถกำหนดค่าความถี่สุ่มได้โดยใช้ซอฟต์แวร์

การออกแบบวงจรเราเลือกไอซี 74LS193 ซึ่งเป็นวงจรมับถอยหลังที่สามารถกำหนดค่าเริ่มต้นได้เป็นเลขฐานสอง 4 บิต จำนวน 2 ตัว ต่อกันเพื่อขยายให้กำหนดค่าเริ่มต้นได้ 8 บิตและใช้งานเป็นวงจรถหารความถี่ ซึ่งสัญญาณอินพุตได้จากสัญญาณนาฬิกาของระบบมีขนาด 5 MHz ต่อกับไอซี 74LS193 อีกหนึ่งตัว ซึ่งถูกต่อเป็นวงจรถหาร 5 จะได้สัญญาณความถี่ 1 MHz ส่วนค่าตัวหารของวงจรถหารความถี่ กำหนดโดยไอซีซอฟต์แวร์ของ TMS 32010 ส่งค่ามาที่ไอซี 74LS373 ซึ่งทำหน้าที่ latch ค่าข้อมูล 8 บิตเพื่อป้อนเป็นค่าเริ่มต้น สัญญาณเอาต์พุตจากวงจรถหารความถี่ จะนำมาเป็นสัญญาณทริกไอซี 74LS122 (Monostable) เพื่อให้สร้างสัญญาณพัลส์ที่มีความกว้างคงที่ เพื่อเป็นสัญญาณอินพุตของวงจรถ่ายสัญญาณ  $\sim$ BIO จะได้วงจรดังรูปที่ 3.7



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.7 วงจรกำหนดความถี่สุ่ม



SIZE	Document Number	REV
B	THE MAINBOARD	
DATE:	REV: 05/17/1993	DATE: 05/17/1993

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 ส่วนแปลงสัญญาณอนาลอกเป็นดิจิทัล

ตัวแปลงสัญญาณ ADC ที่ใช้นางจร คือ ADC908 ซึ่งแปลงสัญญาณอนาลอกเป็นดิจิทัล 8 บิต ลักษณะการทำงานภายในของไอซีนี้ เป็นแบบ ซักเซสซิฟแอฟร็อกซิเมชัน (Successive Aproximation) ซึ่งมีคอนเวอร์ชันไทม์ (Conversion Time) เท่ากับ 6 ไมโครวินาที เพื่อทำงานร่วมกับการประมวลผลของ TMS จึงเลือกใช้ความเร็วในการแปลงน้อย

ไอซี ADC908 มี 2 โหมด คือโหมดยูนิโพลาร์ (Unipolar Mode) และโหมดไบโพลาร์ (Bipolar Mode) ในโครงการนี้จะเลือกใช้งานในโหมดไบโพลาร์ เพื่อรับสัญญาณอนาลอกอินพุตได้ทั้งบวกและลบเลย ขนาดสัญญาณไม่เกิน +10 Volt

การทำงานของ ADC908 แบ่งออกเป็น 3 โหมดคือ

1. รอมโหมด (ROM Mode) ใช้เวลาในการแปลงข้อมูล 7 ไมโครวินาที
2. สแตติกแรมโหมด (Static RAM Mode) เวลาในการแปลงข้อมูล เท่ากับ 6 ไมโครวินาที
3. สโลว์เมมโมรี่โหมด (Slow Memory Mode)

เนื่องจากการทำงาน RAM นั้นต้องใช้สัญญาณควบคุมมาก และในโหมด Slow Memory Mode จะทำให้ TMS คอยการแปลงสัญญาณนาน ดังนั้นการใส่ ADC908 ในโหมด ROM จะมีความเหมาะสมที่สุด โดยป้อนลอจิก "0" เข้าที่ขา  $\sim CS$  เพื่อเลือกให้ ADC908 ทำงาน และอ่านข้อมูลโดยการป้อนลอจิก "0" เข้าที่ขา  $\sim RD$  หลังจากที่เราทำการอ่านข้อมูลเสร็จแล้ว ADC908 ก็จะทำการแปลงข้อมูลชุดต่อไป และรองนกว่าเราจะใส่สัญญาณ  $\sim CS$  และ  $\sim RD$  อีกครั้งจึงปล่อยข้อมูลออกมา เนื่องจากช่วงเวลาที่ ADC908 ปล่อยข้อมูลเข้าสู่ดาต้าบัสของระบบ มีเวลานานเกินไปสำหรับการปฏิบัติงานของ TMS ดังนั้นจึงใช้ 74F373 มาเปิดเกตให้ดาต้าของ ADC908 เข้าสู่ดาต้าบัสของระบบเพียงในช่วงเวลาที่ต้องการ ซึ่งดาต้าจาก ADC908 นี้ จะเป็น 8 บิตล่าง ส่วนของมูล 8 บิตบนจะกำหนดค่าที่เท่ากับ "00000000" โดยใส่ 74LS244 ต่อขาได้อินพุตลงกราวด์

ส่วนแชนเนลแอนด์โฮลด์ ใช้ไอซีเบอร์ LF398 และเลือกค่าตัวเก็บประจุเพื่อหาที่มีความเร็วในการแชนเนลสูงสุด จากดาต้าบุคคือ 1 nF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

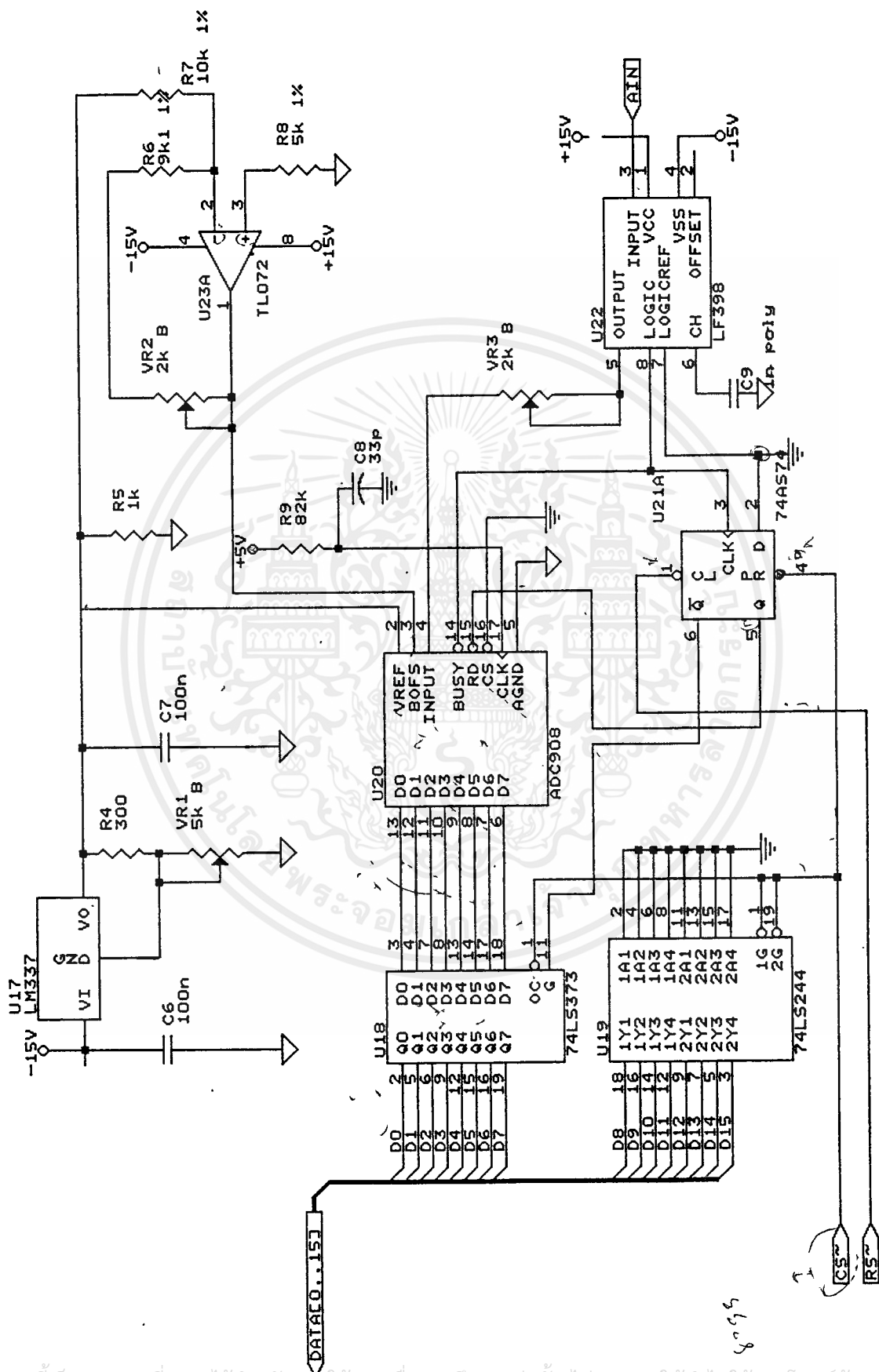
การควบคุมการทำงานเป็นลำดับคือ

-เมื่อทำการรีเซ็ต D-Flip Flop จะถูกเคลียร์ ซึ่งจะทําให้ ADC อยู่ในสภาวะอ่านข้อมูล คือสัญญาณ  $\sim$ BUSY จะเป็น "1" และจาก 74F373 ก็จะได้รับข้อมูลจาก ADC ไปได้

-เมื่อมีสัญญาณอ่านพอร์ต  $\sim$ PS เข้ามา ที่ระดับลอจิก "0" คาต้า 16 บิตจากเอาต์พุตของ 74F373 จะถูกส่งเข้าคาต้าบัส 74AS74 ซึ่งถูกพรีเซ็ต (Preset) Q จะเปลี่ยนจาก "0" เป็น "1" ADC ก็จะเริ่มทําการแปลงข้อมูลและทําให้สัญญาณ  $\sim$ BUSY ที่เปลี่ยนจาก "1" เป็น "0" อินพุตที่ขาลอจิกของ LF398 เพื่อให้ทําการรอลด์สัญญาณอนาลอกไว้ เมื่อ ADC แปลงข้อมูลเรียบร้อยแล้ว สัญญาณ  $\sim$ BUSY จะเปลี่ยนจาก "0" เป็น "1" ทําให้ 74AS74 รับข้อมูลจากขา D มาออกที่ขา Q ทําให้  $Q = 0$  และ  $\sim Q = 1$  ADC จะปล่อยข้อมูลที่แปลงออกมาระยะหนึ่งทําให้ 74LS373 เก็บไว้เตรียมพร้อมที่จะส่งออกเป็นคาต้าเอาต์พุตที่แอด TMS ในครั้งต่อไป

ตารางที่ 3.1 แสดงการทำงานของ ADC908 ในรอมโหมด

Input		Output		การทำงาน
$\sim$ CS	$\sim$ RD	$\sim$ BUSY	data bus	
L	↓ H→L	H	Z→D	อ่านข้อมูล
L	↑ L→H	↑ H→L	D→Z	รีเซ็ตและเริ่มแปลงข้อมูล
L	↓ H→L	L	Z	มาช้า
L	↑ L→H	L	Z	มาอนุญาตทําให้เกิดขึ้น



REV
Size Document Number
A TMS A/D PART
Date: September 3, 1992 Sheet DSP of 3

### 3.3 ส่วนแปลงสัญญาณดิจิทัลเป็นอนาลอก

วงจรนี้ทำหน้าที่ในการส่งข้อมูลออกจากระบบ โดยจะรับข้อมูลมาจาก TMS 32010 โดยานวงจรมีเราเลือกใช้ออสซิลเลเตอร์ DAC 0800 ซึ่งทำหน้าที่แปลงสัญญาณดิจิทัล 8 บิตให้เป็นสัญญาณอนาลอก และมีเซตลิ่งไทม์เท่ากับ 100 ns จากการทำงานของ DAC 0800 ซึ่งจะแปลงข้อมูลดิจิทัลที่อยู่ขาเอาต์พุตของ DAC0800 ดังนั้นเพื่อความต่อเนื่องของสัญญาณอนาลอกที่เอาต์พุต จึงจำเป็นต้องมีข้อมูลที่ขาเอาต์พุตของ DAC 0800 อยู่ตลอดเวลา แต่ข้อมูลที่มาจากขาเอาต์พุตของ TMS 32010 จะมีอยู่เพียงชั่วระยะเวลาหนึ่งเท่านั้น จึงจะต้องมีวงจรที่จะ latch ค่าข้อมูลนั้นไว้ ซึ่งเราเลือกใช้ออสซิลเลเตอร์ 74LS373 แล้วจึงต่อเอาต์พุตของ DAC 0800 มาเชื่อมต่อกับเอาต์พุตของ 74LS373 มาแปลงเป็นสัญญาณอนาลอก เนื่องจากการทำงานของ DAC 0800 จะทำให้เอาต์พุตในรูปแบบของกระแส ดังนั้นจึงต้องมีวงจรแปลงกระแสที่ได้เป็นแรงดันอีกครั้งหนึ่ง ซึ่งจะได้วงจรดังรูปที่ 3.10

เราสามารถคำนวณค่าต่าง ๆ ได้ดังนี้

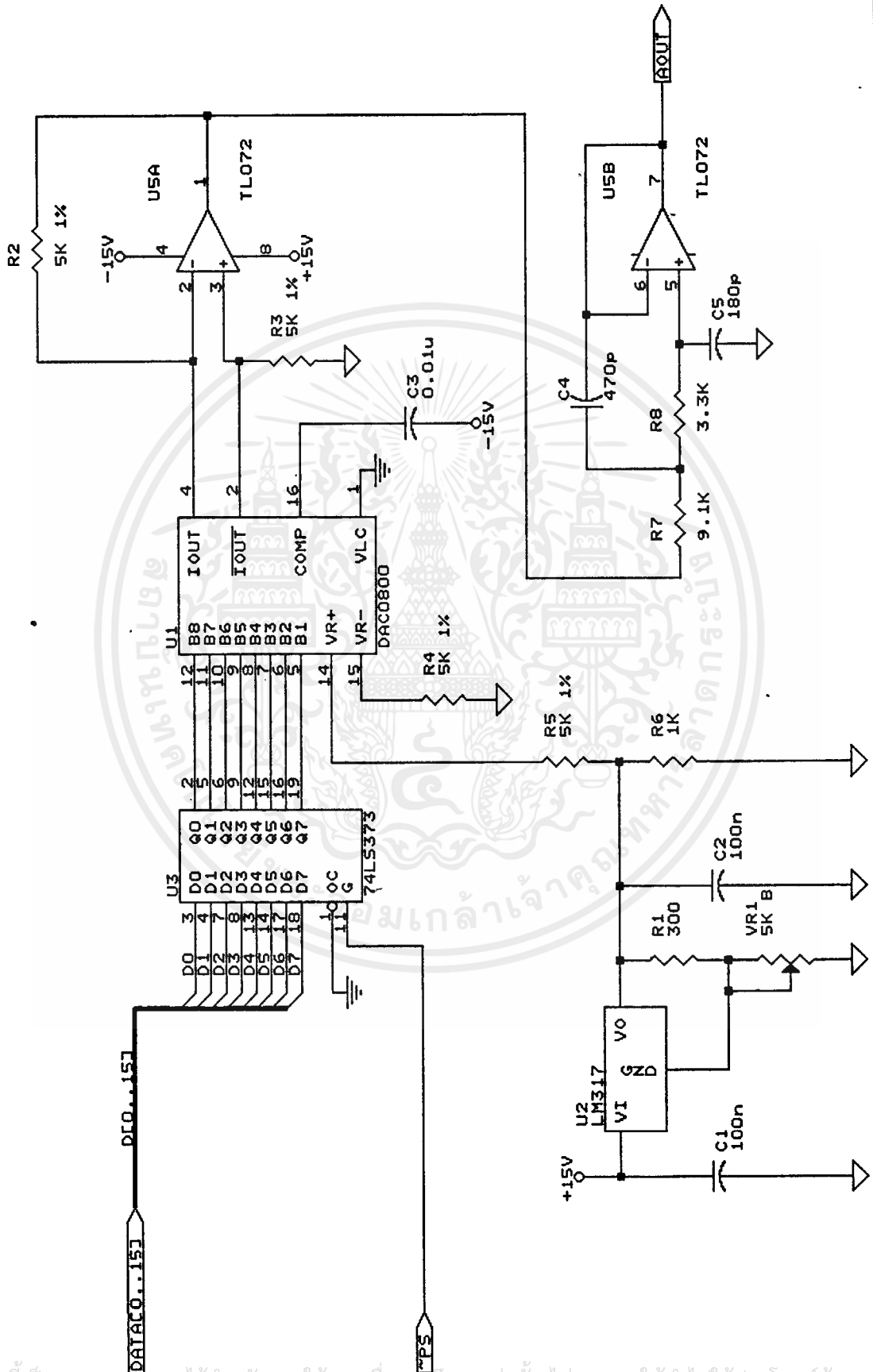
$$\begin{aligned}
 I_{FS} &= (V_{REF} / R_{REF}) (255/256) \\
 I_{FS} &= I_O + \tilde{I}_O \\
 V_{REF} &= 10.000 \text{ V} \\
 R_{REF} &= 5.000 \text{ K}\Omega \\
 R_{15} &= R_{REF} \\
 C_3 &= 0.01 \mu\text{F} \\
 V_{LC} &= 0 \text{ V}
 \end{aligned}$$

ในวงจรออปแอมป์ จะทำหน้าที่เป็นตัวแปลงกระแสให้เป็นแรงดัน โดยสัญญาณเอาต์พุตที่ได้จะอยู่ในรูปไบโพลาร์

สัญญาณ  $V_{REF}$  10.000 V สามารถสร้างได้โดยใช้ออสซิลเลเตอร์ LM317

สัญญาณอนาลอกเอาต์พุตที่ได้จากวงจรแปลงกระแสเป็นแรงดัน จะมีลักษณะเป็นขั้น ๆ ไม่ต่อเนื่องกัน ดังนั้นเราควรจะทำให้สัญญาณเอาต์พุตที่ได้มีลักษณะราบเรียบ โดยอาจใช้วงจร Low Pass Filter โดยตั้งความถี่คัตออฟที่ 100 KHz

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Size Document Number	REV
A	
TMS D/A PART	
Date: October 22, 1992	Sheet DSP of 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่มีให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 ส่วนหน่วยความจำ

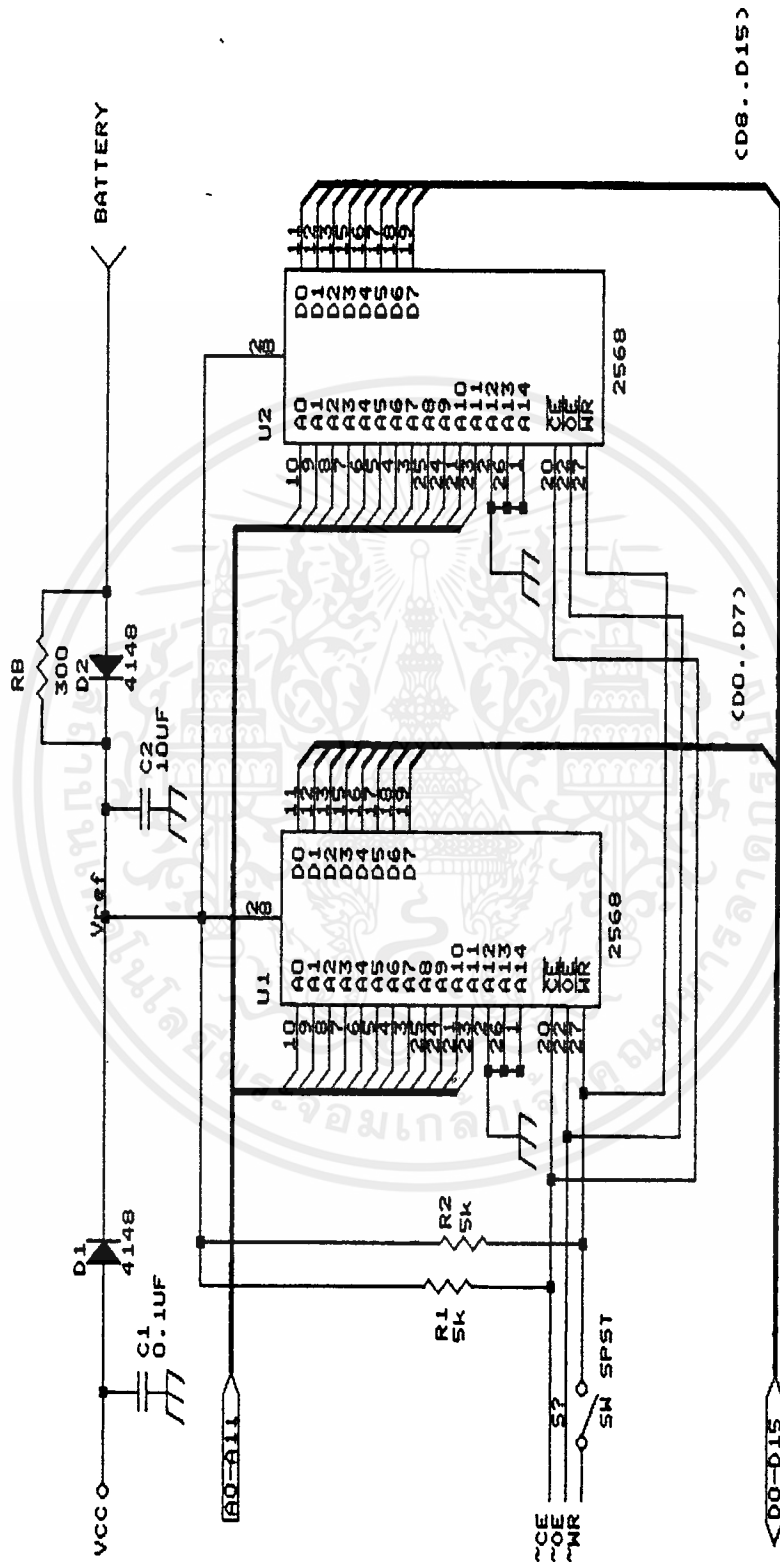
ในที่นี้เราหมายถึง หน่วยความจำภายนอก ซึ่งออกแบบมาเพื่อความสะดวกในการใช้งาน ดังนั้นเราจึงออกแบบให้อยู่ในลักษณะ RAM PACK ที่มี Battery ในตัวสำหรับ Backup ข้อมูล และมีสวิทช์ปิดเปิดสำหรับเลือกการทำงานในโหมด ROM หรือ RAM

เนื่องจาก TMS 32010 ทำงานด้วยความเร็วสูงและจะมีการอ่านหน่วยความจำในทุกรอบของการทำงาน ดังนั้นหน่วยความจำที่ใช้จะต้องมีความเร็วในการทำงานสูงด้วย ในที่นี้เราเลือกใช้อิซซีหน่วยความจำ MT5C2568 ซึ่งเป็นหน่วยความจำแบบสแตติก (Static RAM) ขนาด 32 กิโลไบต์ 8 บิต และมีเวลาในการเข้าถึงข้อมูลไม่เกิน 100 ns แต่เนื่องจาก TMS 32010 เป็นตัวประมวลผลขนาด 16 บิต และสามารถอ้างตำแหน่งของ หน่วยความจำภายนอกได้ 4 กิโลเวิร์ด ดังนั้นจึงต้องใช้อิซซี MT5C2568 2 ตัว มาต่อขนานกันเพื่อให้ได้ขนาด 16 บิต ซึ่งเราสามารถออกแบบวงจรได้ดังรูปที่ 3.11

นอกเหนือจากวงจรส่วนต่างๆ ดังได้กล่าวข้างต้น ซึ่งเป็นวงจรหลักในการทำงานแล้ว ในการพัฒนาโปรแกรม จะต้องมีการเขียนโปรแกรมลงใน RAM PACK และเนื่องจาก RAM PACK ที่เราออกแบบมีขนาด 16 บิต ดังนั้นในการพัฒนาโปรแกรมบนเครื่อง IBM PC และทำการเขียนข้อมูลลงใน RAM PACK เราจะต้องออกแบบวงจรที่เชื่อมต่อการระหว่างเครื่อง IBM PC และ RAM PACK

จากการศึกษาพบว่า เราสามารถเชื่อมต่อวงจรที่เราออกแบบกับ IBM PC ผ่านทาง I/O PORT ที่ว่างอยู่ โดยการออกแบบวงจรเป็นการติดต่อที่ SLOT ของเครื่อง IBM PC ในที่นี้เราเลือกใช้พอร์ท

เนื่องจากขาเอาต์พุตของ I/O PORT มีขนาด 8 บิต และ RAM PACK ที่เราออกแบบต้องการข้อมูลสำหรับขาแอดเดรส 12 บิต ขาด้าน 16 บิต และขาสัญญาณที่ใช้ในการควบคุมคือ  $\sim WE, \sim OE, \sim CE$  ดังนั้นเราจึงต้องใช้ I/O PORT ของ IBM PC ทั้งหมดจำนวน 4 พอร์ท ในการออกแบบเราเลือกใช้อิซซี 8255 จำนวน 1 ตัว ซึ่งมี 3 พอร์ท และอิซซี 74LS373 จำนวน 1 ตัว เป็นพอร์ทอีก 1 พอร์ท และอิซซี 74LS138 ซึ่งเป็น Decoder ร่วมกับ 74LS20 เป็น NAND GATE 4 อินพุต และ 74LS02 เป็น NOR GATE 2 อินพุต เป็นวงจรถอดรหัสสัญญาณแอดเดรสของ I/O PORT จาก IBM PC จะได้ว่าวงจรดังรูปที่ 3.12



Size	Document Number	REV
A	TMS RAMPACK	
Date:	May 20, 1988	Sheet DSP of 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



บทที่ 4 การแปลงฟูเรียร์ของสัญญาณ

ในการวิเคราะห์องค์ประกอบของสัญญาณอย่างละเอียด มักพิจารณาจากสัญญาณในโดเมนความถี่ เรียกว่า SPECTRUM ความถี่ของสัญญาณ ทำให้บอกได้ว่า สัญญาณหนึ่งๆ ประกอบด้วยความถี่อะไรบ้าง ความเข้มเท่าใด โดยที่สัญญาณที่วิเคราะห์ทำไมจำเป็นต้องมีความเป็นคาบในโดเมนเวลา ตัวอย่างสัญญาณที่จะต้องวิเคราะห์ทาง SPECTRUM เช่น เสียงทางดนตรี เสียงพูดของคน เสียงของเครื่องจักร ความเข้มของแสงชนิดต่างๆ

4.1 หลักการและที่มาของ DISCRETE FOURIER TRANSFORM (DFT)

การแปลงสัญญาณในโดเมนเวลาให้เป็นสัญญาณในโดเมนความถี่ โดยใช้การแปลงฟูเรียร์ สำหรับสัญญาณใดๆที่มีความต่อเนื่องทางเวลา จะได้ว่า

$$X(w) = \int_{-\infty}^{\infty} x(t)e^{-jwt} dt \quad (1)$$

โดยที่  $w$  เป็นตัวแปรทางความถี่  
 $t$  เป็นตัวแปรทางเวลา

สำหรับสัญญาณที่ถูกสุ่มมาเป็นช่วงๆ ในโดเมนเวลา ( $x(n)$ ) การแปลงฟูเรียร์จะต้องทำเป็นลำดับ เรียกการแปลงฟูเรียร์ของสัญญาณที่ถูกสุ่มมา (DISCRETE SIGNAL) นี้ว่าการทำฟูเรียร์แบบ DISCRETE (DISCRETE FOURIER TRANSFORM) ให้ผลตามสมการ

$$X(w) = \sum_{n=-\infty}^{\infty} x(n)e^{-jwn} \quad (2)$$

$w$  เป็น NORMALIZE FREQUENCY ซึ่งจะแปรค่าอยู่ในช่วง 0-2 ดังนั้น  $X(w)$  จะครบคาบที่ 2 ซึ่งเป็นเหตุผลด้วยว่า SPECTRUM ของสัญญาณที่ถูกสุ่มมาจะมีความเป็นคาบ

ในการแปลง DFT จะสมมุติให้สัญญาณที่สุ่มมามีการครบคาบ จากการสุ่ม  $N$  ครั้งพอดี สมการของ DFT  $N$  จุดจะได้เป็น

$$N-1$$

$$X(k) = \sum_{n=0}^{N-1} x(n)W_N^{nk} \quad (3)$$

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัด  $n=0$  เนื้อหาและต้องอ้างอิงถึง  $k=0, 1, 2, \dots, N-1$  นี้เป็นการนำไปใช้

เมื่อ  $w_N$  คือ PHASE หรือ TWIDDLE FACTOR

มีค่าเท่ากับ  $e^{-j2\pi/N}$

$k$  คือ ลำดับของช่วงความถี่

เปรียบเทียบกับสมการ (2) จะเห็นว่า  $w$  ในสมการ (2) เป็น  $2\pi k/N$  ในสมการ (3) คือเป็นค่าที่ถูกแบ่งไว้เป็นช่วงๆ จำนวน  $N$  ช่วง

ในสมการ (3) เราประมาณได้ว่า จะต้องมีการคูณกันจำนวน  $N^2$  ครั้งจากการทำ DFT  $N$  จุด เนื่องจากที่  $x(k)$  หนึ่งๆ จะต้องมีการคูณกันจำนวน  $N$  ครั้ง ทำให้เสียเวลาในการคำนวณมาก ทำให้การประมวลผลไม่เป็น REAL TIME การลดเวลาในการคำนวณ DFT ก็ทำได้จากการลดขั้นตอนการคูณ โดยแบ่งสมการฟูเรียร์พิจารณาเป็นกลุ่มย่อย เรียกรวมนี้ว่า การแปลงฟูเรียร์ (FAST FOURIER TRANSFORM: FFT) สามารถทำได้หลายวิธี เช่น การแบ่งกลุ่มตามอินพุตในโดเมนเวลา (DECIMAL IN TIME: DIT) แบ่งกลุ่มตามเข้าที่พุทในโดเมนความถี่ (DECIMAL IN FREQUENCY: DIF) เป็นต้น ในที่นี้ใช้ DFT แบบ DIT

#### 4.1.1 DIT FFT

จากสมการ (3) ถ้าทำการแยกเทอมดู , ได้ จะได้ว่า

$$x(k) = \sum_{n=0}^{(N/2)-1} x(2n)w_N^{2nk} + \sum_{n=0}^{(N/2)-1} x(2n+1)w_N^{2(n+1)k}$$

$$= \sum_{n=0}^{(N/2)-1} x(2n)w_N^{2nk} + w_N^k \sum_{n=0}^{(N/2)-1} x(2n+1)w_N^{2nk}$$

(4)

$$\text{พิจารณา } w_N^2 = e^{-j2\pi/N \cdot 2}$$

$$= e^{-j2\pi/(N/2)}$$

$$= w_{N/2}$$

ดังนั้น เราจะเขียนสมการ (4) ได้ใหม่เป็น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X(k) = \sum_{n=0}^{(N/2)-1} x(2n)W_{N/2}^{nk} + W_N^k \sum_{n=0}^{(N/2)-1} x(2n+1)W_{N/2}^{nk}$$

(5)

ซึ่งอาจมองได้เป็น

$$X(k) = Y(k) + W_N^k \cdot Z(k)$$

เมื่อ  $Y(k)$  เป็นผลของการกระทำในเทอมคู่

$Z(k)$  เป็นผลของการกระทำในเทอมคี่

สามารถเขียนเป็นแผนภาพให้เข้าใจดังรูปที่ 4.1

ถึงตอนนี้จำนวนครั้งในการคูณจะลดได้เป็น  $2(N/2)^2 + N$  เมื่อมองจากแผนภาพในรูปที่ 1 จะเห็นว่า Spectrum  $X(k)$  แบ่งเป็น 2 ช่วง คือ

ช่วงแรก  $0 \leq k \leq (N/2)-1$

ช่วงหลัง  $N/2 \leq k \leq (N-1)$

พิจารณาจากสมการ (5) จะได้ว่า

$$X(k) = \sum_{n=0}^{(N/2)-1} x(2n)W_{N/2}^{nk} + W_N^k \sum_{n=0}^{(N/2)-1} x(2n+1)W_{N/2}^{nk}$$

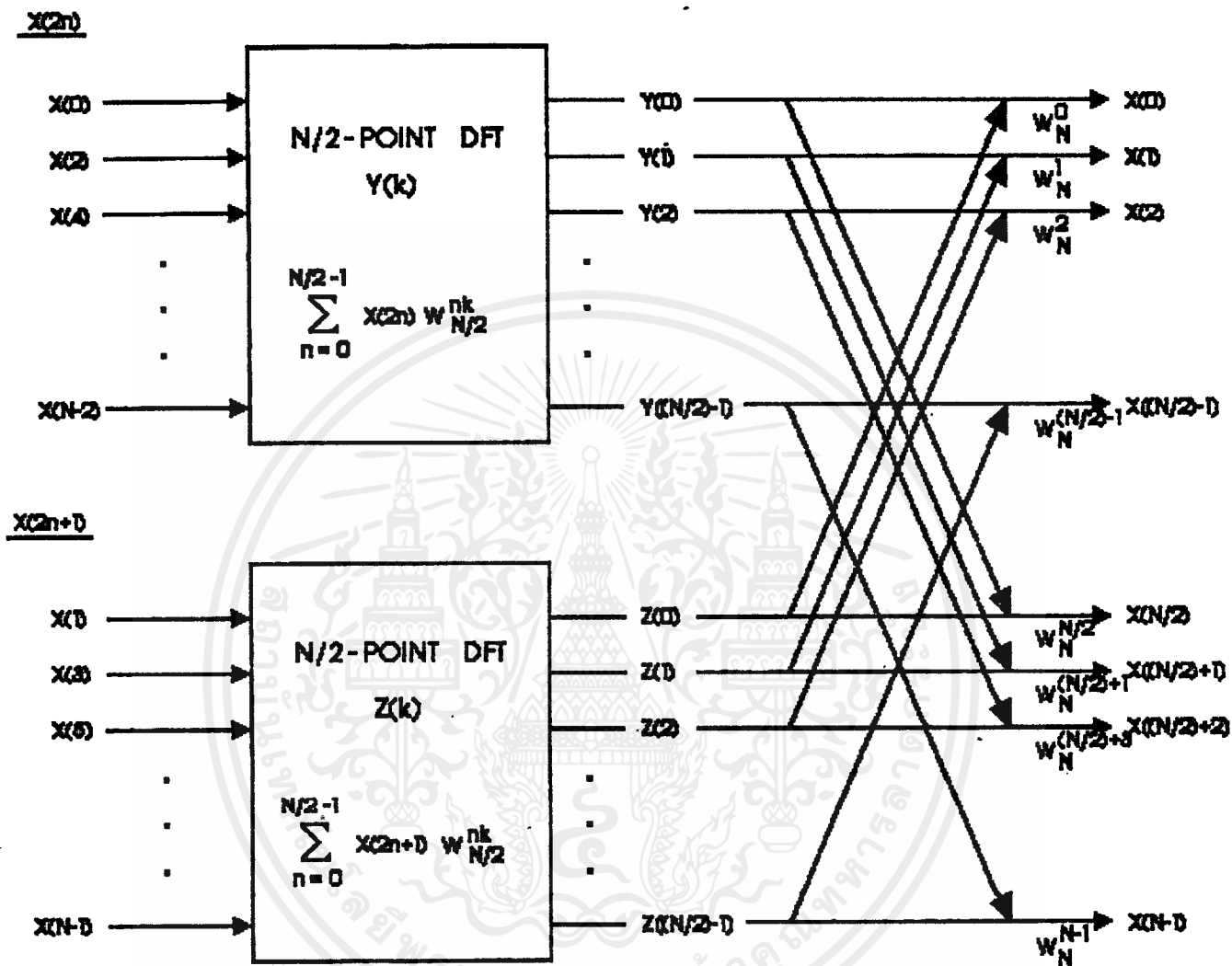
ที่  $0 \leq k \leq (N/2)-1$

ส่วนในช่วงหลังเมื่อแทนค่า

$$X(k+N/2) = \sum_{n=0}^{(N/2)-1} x(2n)W_{N/2}^{n(k+N/2)}$$

$(N/2)-1$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ...  $+ W_N^{k+(N/2)} \sum_{n=0}^{(N/2)-1} x(2n+1)W_{N/2}^{n(k+N/2)}$  โยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและ  $n=0$  อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 ขั้นตอนการแบ่งเทอมคู่,คี่ ของสัญญาณ input ในการทำ DIT FFT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่า

$$\begin{aligned} W_N^{(k+N/2)} &= W_{N/2}^{n(N/2)} \cdot W_{N/2}^{nk} \\ &= e^{-j2\pi n} \cdot W_{N/2}^{nk} \\ &= W_{N/2}^{nk} \end{aligned}$$

และ  $W_N^{(k+N/2)} = W_N^k \cdot e^{-j\pi}$

$$= -W_N^k$$

ทำให้ได้สมการในส่วนของครึ่งหลังเป็น

$$X(k+N/2) = \sum_{n=0}^{(N/2)-1} x(2n)W_{N/2}^{nk} - W_N^k \sum_{n=0}^{(N/2)-1} x(2n+1)W_{N/2}^{nk}$$

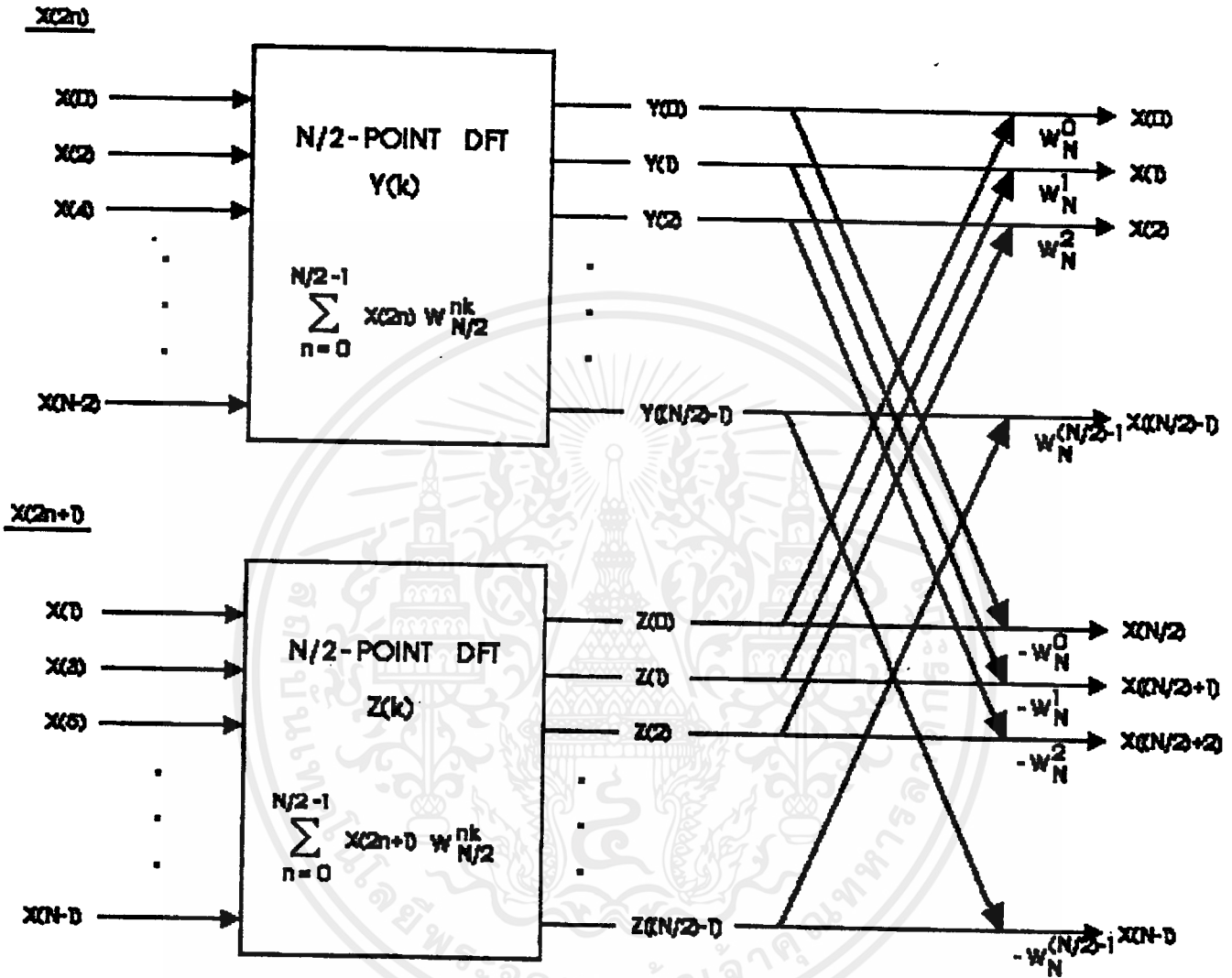
ที่  $0 \leq k \leq (N/2)-1$

เขียนนารูปของ  $Y(k)$  และ  $Z(k)$  ได้เป็น

$$X(k+N/2) = Y(k) - W_N^k \cdot Z(k)$$

ทำให้มองแผนภาพได้ง่ายขึ้นดังในรูปที่ 4.2

นอกจากนั้นเรายังสามารถจะแยก  $N/2$  point DFT ให้แยกย่อยลงไปได้อีก เป็น  $N/4, N/8, \dots$  point ได้อีกตามหลักการเดิม จนได้ DFT ส่วนที่น้อยที่สุดคือ DFT 2 point หรือ 2 จุด แผนภาพของการทำ DFT 2 จุด มีลักษณะคล้ายผีเสื้อ จึงเรียกแผนภาพนี้ว่า แผนภาพ Butterfly



รูปที่ 4.2 การแทนค่า  $x(k)$  ในเทอมคู่และคี่ [  $Y(k)$  และ  $Z(k)$  ]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 DIT FFT BUTTERFLY

จากการแทนค่าสมการในช่วงครึ่งแรกและครึ่งหลังของ DFT 2 point

(N=2)

จะได้  $X(k) = Y(k) + W_N^k \cdot Z(k)$

และ  $X(k+1) = Y(k) - W_N^k \cdot Z(k)$

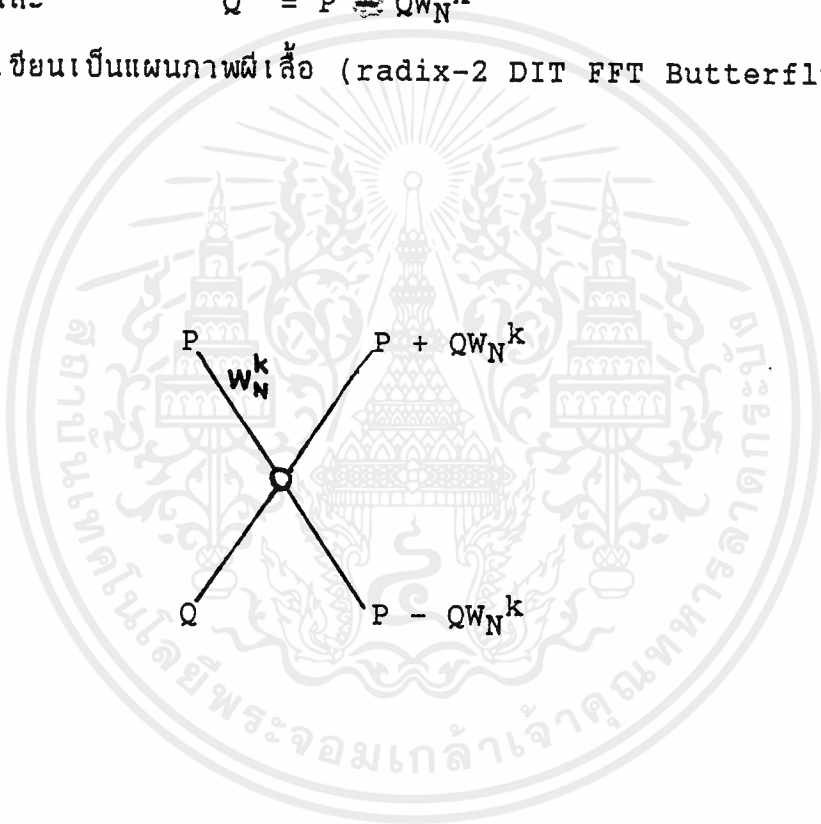
ซึ่งอาจมองเป็น

$$P' = P + QW_N^k$$

และ  $Q' = P - QW_N^k$

เขียนเป็นแผนภาพผีเสื้อ (radix-2 DIT FFT Butterfly Symbol) ได้

ดังรูปที่ 4.3



รูปที่ 4.3 Radix-2 DIT Butterfly Symbol

ในการทำ FFT N จุด เราแบ่งครึ่งทำ FFT ๒๒เรื่อยๆจนเหลือ 2 จุด ถ้าทำการแบ่งไปทั้งหมด M ครั้ง (N = 2<sup>M</sup>) นั่นคือเราจะต้องทำ Radix-2 Butterfly ทั้งหมด M Stage จำนวนของ Butterfly ทั้งหมดจะเท่ากับ N/2 log<sub>2</sub> N ชุด และมีการคูณกันทั้งหมด N log<sub>2</sub> N ครั้ง ซึ่งน้อยลงกว่าการทำ DFT ธรรมดา

แต่การทำ FFT ยังมีข้อควรพิจารณาอีกหลายประการ อันได้แก่ การทำ Scaling, การทำ Bit-Reverse งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 การทำ Scaling

จากการทำ Butterfly เราได้

$$P_{m+1} = P_m + W_N^k \cdot Q_m$$

$$Q_{m+1} = P_m - W_N^k \cdot Q_m$$

ซึ่ง  $W_N^k = e^{-j(2\pi/N)k}$   
 $= \cos(X) - j\sin(X)$  เมื่อ  $X = (2\pi/N)$

$P_m$  และ  $Q_m$  ในความเป็นจริงแล้วจะประกอบด้วยส่วนจริงและส่วนจินตภาพ คือ

$$P_m = PR + jPI$$

และ  $Q_m = QR + jQI$

ทำที่เราได้

$$\begin{aligned} P_{m+1} &= PR + jPI + (QR\cos(X) + QI\sin(X)) \\ &\quad + j(QI\cos(X) - QR\sin(X)) \\ &= (PR + QR\cos(X) + QI\sin(X)) \\ &\quad + j(PI + QI\cos(X) - QR\sin(X)) \end{aligned} \tag{7}$$

$$\begin{aligned} Q_{m+1} &= PR + jPI - (QR\cos(X) + QI\sin(X)) \\ &\quad - j(QI\cos(X) - QR\sin(X)) \\ &= (PR - QR\cos(X) - QI\sin(X)) \\ &\quad + j(PI - QI\cos(X) + QR\sin(X)) \end{aligned}$$

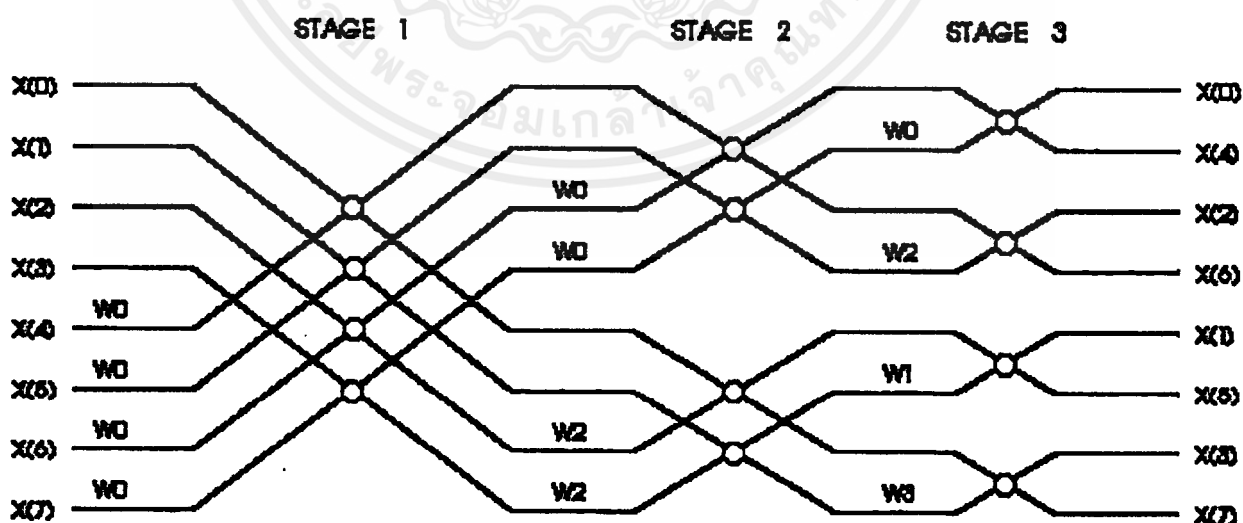
พิจารณาค่าสูงสุดที่จะเป็นได้ของ output ดูได้จากสมการ(7)คือประมาณ

$$1 + 1\sin 45 + 1\cos 45 = 2.414213562$$

ซึ่งเกิน 1 ทำให้เกิดการ overflow Butterfly ในแต่ละ stage จึงควรมีการ scale ค่าลงโดยการหาร 2 ดังนั้น ถ้ามี M stage output จะถูก scale ลงเท่ากับ  $2^M = N$  ในกรณีที่ทำการ FFT N จุด ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการคำนวณใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

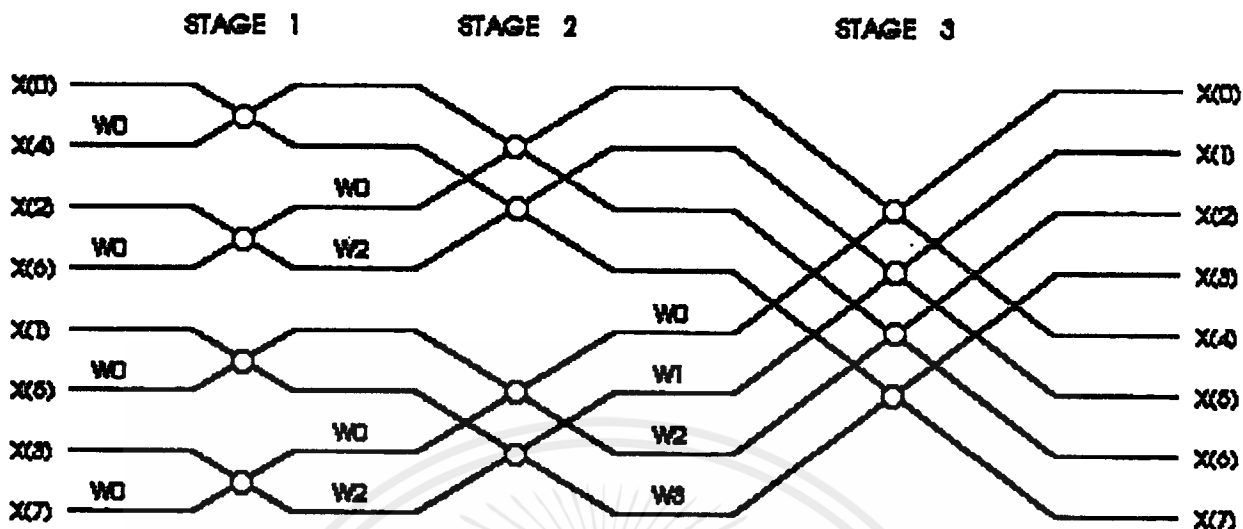
ตารางที่ 4.1 การเรียงและสลับลำดับ BIT ของ ลำดับ FFT 8 จุด

ลำดับปีท	เลขฐานสอง	Bit-Reverse	ลำดับกลับปีท
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



รูปที่ 4.4 แผนภาพการทํา FFT 8 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สําหรับการวิจัยและพัฒนาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แผนภาพการทำ FFT 8 จุด โดยทำ Bit-Reverse ที่ input

4.1.4 การทำ Bit-Reverse

หากพิจารณาการทำ FFT 8 จุด ในรูปที่ 4.4 จะเห็นว่า output ที่ได้จะสลับลำดับ โดยสังเกตว่าการเรียง Bit ในลำดับจะกลับกัน ดังตารางที่ 4.1 แต่ถ้าเรากลับลำดับ input เสียก่อน output ที่ได้ จะเรียงลำดับกันดังรูปที่ 4.5

4.1.5 การแปลงผกผันของ FFT (INVERSE FFT)

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นภายใต้แอสซิมเมตริกเดียวกันกับสมการ DFT ถ้าแทน  $k$  ด้วย  $n$  ,  $x(n)$  ด้วย  $X(k)$  และแทน  $w_N^{nk}$  ด้วย  $w_N^{-nk}$  แล้วก็จะได้ IDFT ของ  $X(k)$  ได้โดยง่าย

ในกระบวนการที่เราแปลงค่าต่อเนื่องกันคือ ทำ FFT แล้วนำผลไปแปลงค่า IFFT จากแอสซิมเมตริกเดียวกันหากมีการ Scaling ระหว่างขั้นตอนของ FFT นั่นคือ ผลลัพธ์ที่ได้  $= (1/N)X(k)$  ในขั้นตอนของ IFFT ก็จำเป็นต้องมีการ Scaling อีก

#### 4.2 คุณสมบัติบางประการของ DFT

เนื่องจาก DFT นั้น พัฒนามาจากการทำ FOURIER TRANSFORM จึงมีคุณสมบัติเช่นเดียวกับคุณสมบัติของ FOURIER TRANSFORM คุณสมบัติบางประการที่จะกล่าวถึงงานนี้ได้แก่

1) ความเป็นเชิงเส้น (LINEARITY)

ถ้า  $X_1(k)$  เป็น DFT ของ  $x_1(n)$

และ  $X_2(k)$  " "  $x_2(n)$

ถ้า  $x_3(n) = a_1x_1(n) + a_2x_2(n)$

จะได้ว่า DFT ของ  $x_3(n)$  คือ

$$X_3(k) = a_1X_1(k) + a_2X_2(k)$$

2) ความเป็นคาบ (PERIODICITY)

ทั้ง DFT และ IDFT จะมีความเป็นคาบเท่ากับจำนวนจุดของการกระทำ (N) เหตุผลเนื่องมาจากเทอม  $w_N^{nk}$

พิสูจน์ให้เห็นได้ว่า  $X(k) = X(k+N)$

ได้โดย 
$$X(k+N) = \sum_{n=0}^{N-1} x(n)w_N^{n(k+N)}$$

$$= \sum_{n=0}^{N-1} x(n)w_N^{nk}w_N^{nN}$$

เพราะว่า  $w_N^{nN} = e^{-j2\pi n} = 1$

เอกสารนี้เป็น... ดังนั้นก็จะได้ว่า  $X(k+N) = X(k)$  สำหรับ  $N$  IDFT  $x(n) = x(n+N)$  ค่า  
ก็พิสูจน์เหมือนกัน ห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) DFT ของลำดับคู่ (DFT OF EVEN AND ODD SEQUENCES)

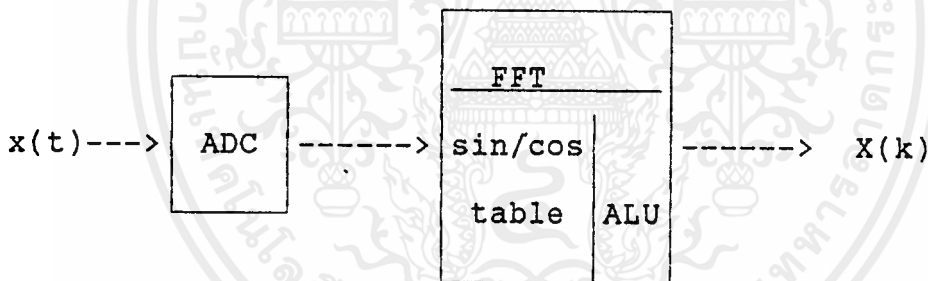
DFT ของลำดับคู่ซึ่งอยู่ในเทอมของ COSINE จะได้แต่ค่า REAL และลำดับคู่ซึ่งอยู่ในเทอมของ SINE จะได้แต่ค่า IMAGINARY

4) DFT สำหรับลำดับค่า REAL (DFT OF REAL SEQUENCE)

สำหรับลำดับซึ่งประกอบด้วยทุกค่าของ  $x(n)$  เป็น REAL NUMBER จะได้ DFT เป็นลักษณะสมมาตรคู่ (SYMMETRIC) ในแกน REAL และสมมาตรคี่ (ANTI-SYMMETRIC) ในแกน IMAGINARY ที่จุดกึ่งกลางของสมมาตร  $X(k) = X(N/2)$  เนื่องจากคุณสมบัติของการเป็น CONJUGATE ซึ่งกันและกันดังนี้

$$X(k) = X^*(N-k) \quad ; \quad k = 0, 1, 2, \dots, N-1$$

4.3 ERROR จากการทำ FFT



พิจารณาจากแผนภาพของระบบในการทำ FFT จะพบว่า ERROR สามารถเกิดขึ้นได้ตั้งแต่ INPUT SIGNAL ที่มี WHITE NOISE บน (ถ้ามีอยู่น้อยก็อาจตัดทิ้งได้) , ADC อันได้แก่ QUANTIZATION ERROR, SATURATION ERROR และ SAMPLING ERROR

QUANTIZATION ERROR เกิดจากการ SAMPLING ค่าอินพุตที่มีลักษณะต่อเนื่อง แต่ ADC ำให้ค่าเฉพาะระดับขั้นเท่านั้น ดังที่เคยกล่าวไว้แล้วก่อนหน้านี้

SATURATION ERROR เกิดจากการที่อินพุตเกินระดับการทำงานของ ADC ซึ่งอาจตัดทิ้งได้ในกรณีที่เรำให้อินพุตน้อยกว่าช่วงการทำงานของ ADC

SAMPLING ERROR เกิดจากความผิดพลาดของเวลาในการสุ่ม เช่น ำการสุ่มภายในช่วงเวลาของการสุ่มจริง (กรณีนี้ถ้ามีความผิดพลาดน้อยมาก ก็อาจตัดทิ้งได้เช่นกัน)

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ขั้นตอนการ PROCESS FFT ERROR อันเกิดจากการถูกจำกัดลำดับ ดังนั้นยิ่งทำ FFT ด้วยจำนวนจุด(N)มาก ความถูกต้องก็จะยิ่งมากขึ้น บทนิยามทำ  $N > 1024$  จุด ความยาวอันจำกัดของREGISTER ที่เก็บค่าแต่ละพจน์(FINITE WORD-LENGTH REGISTER) ได้แก่ค่า SINE, COSINE ที่เรา ASSIGN ว่าเป็นTABLE เมื่อPROCESSOR ทำการประมวลผลที่ALU ก็จะทำให้เกิดการปัดค่าในระหว่างการคูณขึ้นด้วย(ROUNDING) นอกจากนั้นการทำSCALING เพื่อไม่ให้เกิดOVERFLOW ทำให้ต้องตัดความผิดพลาดไปจากค่าเดิมด้วย

เนื่องจากการประเมินขอบเขตของความผิดพลาดที่จะยอมรับได้นั้น พิจารณาจาก ทฤษฎีความน่าจะเป็นและการกระจายค่าของตัวแปร ซึ่งมีความซับซ้อนจึงมาได้เพียงในที่นี้ แต่พึงระลึกไว้เสมอว่า ความผิดพลาดในขอบเขตที่ยอมรับได้นั้น เป็นสิ่งสำคัญเมื่อการทำ FFT ถูกนำไปใช้งานต่างๆ ซึ่งค่าที่ยอมรับได้ในงานต่างๆก็จะแตกต่างกันไป ตามความเหมาะสม

บทที่ 5 ผลการทดลองและสรุปผลการทดลองการทดลอง

เราสามารถนำชุดทดลองที่สร้างขึ้นมาประยุกต์ใช้ร่วมกับโปรแกรมต่าง ๆ ที่เราเขียน ทว่าการทดลองต่าง ๆ ดังนี้

1. วิเคราะห์หาสเปกตรัมความถี่ของสัญญาณต่าง ๆ โดยใช้โปรแกรมการแปลงฟูรีเยร์ และเปรียบเทียบผลที่ได้กับผลการทดลองที่ได้จากการใช้โปรแกรมภาษาซี โดยมีตัวอย่างสัญญาณที่ทำการวิเคราะห์ดังนี้

- สัญญาณไซน์ ขนาด 1 V<sub>pp</sub> ความถี่ 1 KHz และ 5 KHz
- สัญญาณสามเหลี่ยม ขนาด 1 V<sub>pp</sub> ความถี่ 1 KHz
- สัญญาณสี่เหลี่ยม ขนาด 1 V<sub>pp</sub> ความถี่ 1 KHz

ผลการทดลองแสดงในตารางที่ 1-4 และ รูปที่ 1-4

2. สังเคราะห์สัญญาณต่าง ๆ ที่มีความถี่ตามที่ต้องการ โดยใช้โปรแกรมการแปลงฟูรีเยร์ผกผัน และเปรียบเทียบผลที่ได้กับผลการทดลองที่ได้จากการใช้โปรแกรมภาษาซี ทว่าการทดลองโดยป้อนผลลัพธ์ที่ได้จากการทดลองข้อ 1 เป็นอินพุต ในการสังเคราะห์สัญญาณต่าง ๆ กลับคืนมา ผลการทดลองแสดงในตารางที่ 5-8 และ รูปที่ 5-8

ตารางที่ 1    เปรียบเทียบผลการวิเคราะห์สเปกตรัมของสัญญาณชานัน์ ขนาด 1 V<sub>pp</sub>  
ความถี่ 1 KHz โดยใช้ TMS 32010 และ โปรแกรมภาษาซี

k	TMS 32010			โปรแกรมภาษาซี		
	X <sub>r</sub>	X <sub>i</sub>	Mag	X <sub>r</sub>	X <sub>i</sub>	Mag
0	3936	0	3936**	3971.0	0.0	3971.0**
1	-448	-288	544	-375.1	-231.2	440.7
2	-32	0	32	-6.5	0.7	6.5
3	-64	-32	96	1.3	-8.9	9.0
4	0	0	0	5.5	-0.0	5.5
5	-32	0	32	5.7	-1.9	6.0
6	-64	0	64	0.2	1.8	1.8
7	-64	0	64	2.5	2.7	3.7
8	0	0	0	5.0	-4.0	6.4
9	0	-32	32	8.2	-8.7	11.9
10	0	0	0	-1.0	1.8	2.1
11	-32	-32	32	2.4	1.9	3.1
12	0	0	0	8.4	0.0	8.4
13	-32	-32	32	5.1	-0.6	5.1
14	0	-32	32	11.3	0.7	11.3
15	-32	-32	32	5.6	1.7	5.9
16	0	0	0	-1.0	0.0	1.0
17	0	-32	32	5.6	-1.7	5.9
18	32	0	32	11.3	-0.7	11.3

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

k	TMS 32010			โปรแกรมภาษาซี		
	$X_r$	$X_i$	Mag	$X_r$	$X_i$	Mag
19	64	-32	96	5.1	0.6	5.1
20	0	0	0	8.4	0.0	8.4
21	32	0	32	2.4	-1.9	3.1
22	0	0	0	-1.0	-1.8	2.1
23	0	0	0	8.2	8.7	11.9
24	0	0	0	5.0	4.0	6.4
25	0	-32	32	2.5	-2.7	3.7
26	0	0	0	0.2	-1.8	1.8
27	32	-32	32	5.7	1.9	6.0
28	0	0	0	5.5	-0.0	5.5
29	32	32	32	1.3	8.9	9.0
30	0	-32	32	-6.5	-0.7	6.5
31	-352	224	416	-375.1	231.2	440.7

หมายเหตุ \*\* ขนาดของ  $x(0)$  จะมีค่าสูงมากเมื่อเทียบกับที่ตำแหน่งอื่น ๆ ดังนั้น  
 เพื่อให้สามารถมองเห็นความแตกต่างของขนาดที่ตำแหน่งอื่น ๆ  
 เราจึงไม่นำค่า  $x(0)$  มา plot กราฟ

SPECTRUM x(k)



magnitude response of fft  
 max val= 16.64

รูปที่ 5.1 ผลการวิเคราะห์สเปกตรัมของสัญญาณชายน ขนาด 1 V<sub>pp</sub>  
 ความถี่ 1 KHz โดยาช TMS 32010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2 เปรียบเทียบผลการวิเคราะห์สเปกตรัมของสัญญาณชายน์ ขนาด 1 V<sub>pp</sub> ความถี่ 5 KHz โดยใช้ TMS 32010 และ โปรแกรมภาษาซี

k	TMS 32010			โปรแกรมภาษาซี		
	X <sub>r</sub>	X <sub>i</sub>	Mag	X <sub>r</sub>	X <sub>i</sub>	Mag
0	3840	0	3840**	3876.0	0.0	3876.0**
1	-32	-32	32	28.0	0.4	28.0
2	-32	-32	32	30.0	-4.3	30.3
3	-96	-32	96	19.6	-5.2	20.3
4	0	0	0	34.4	-12.4	36.6
5	-320	320	480	-255.9	342.3	427.4
6	-32	0	32	12.4	10.0	15.9
7	-32	0	32	11.2	11.1	15.8
8	0	0	0	24.0	4.0	24.3
9	32	0	32	18.0	-1.8	18.1
10	0	-32	32	18.2	4.0	18.7
11	0	-32	32	17.1	-5.5	18.0
12	0	0	0	17.5	-4.4	18.0
13	-32	0	32	16.5	8.3	18.5
14	-32	-32	32	23.2	1.6	23.3
15	0	0	0	17.2	4.9	17.9
16	32	0	32	28.0	0.0	28.0
17	32	-32	32	17.2	-4.9	17.9
18	32	-32	32	23.2	-1.6	23.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

k	TMS 32010			โปรแกรมภาษาซี		
	$X_r$	$X_i$	Mag	$X_r$	$X_i$	Mag
19	32	-32	32	16.5	-8.3	18.5
20	0	0	0	17.5	4.4	18.0
21	0	0	0	17.1	5.5	18.0
22	32	0	32	18.2	-4.0	18.7
23	32	0	32	18.0	1.8	18.1
24	0	-32	32	24.0	-4.0	24.3
25	32	0	32	11.2	-11.1	15.8
26	0	-32	32	12.4	-10.0	15.9
27	-256	-352	416	-255.9	-342.3	427.4
28	0	0	0	34.4	12.4	36.6
29	32	0	32	19.6	5.2	20.3
30	32	-32	32	30.0	4.3	30.3
31	64	0	64	28.0	-0.4	28.0

**หมายเหตุ** \*\* ขนาดของ  $X(0)$  จะมีค่าสูงมากเมื่อเทียบกับที่ตำแหน่งอื่น ๆ ดังนั้น  
 เพื่อให้สามารถมองเห็นความแตกต่างของขนาดที่ตำแหน่งอื่น ๆ  
 เราจึงไม่นำค่า  $X(0)$  มา plot กราฟ

SPECTRUM  $x(k)$



magnitude response of fft

FREQUENCY k

max val= 14.14

รูปที่ 5.2 ผลการวิเคราะห์สเปกตรัมของสัญญาณชานัน ขนาด 1 V<sub>pp</sub>  
 ความถี่ 5 KHz โดยใช้ TMS 32010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการทำ  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 3** เปรียบเทียบผลการวิเคราะห์สเปกตรัมของ สัญญาณสี่เหลี่ยม ขนาด 1 V<sub>pp</sub> ความถี่ 1 KHz โดยใช้ TMS 32010 และ โปรแกรมภาษาซี

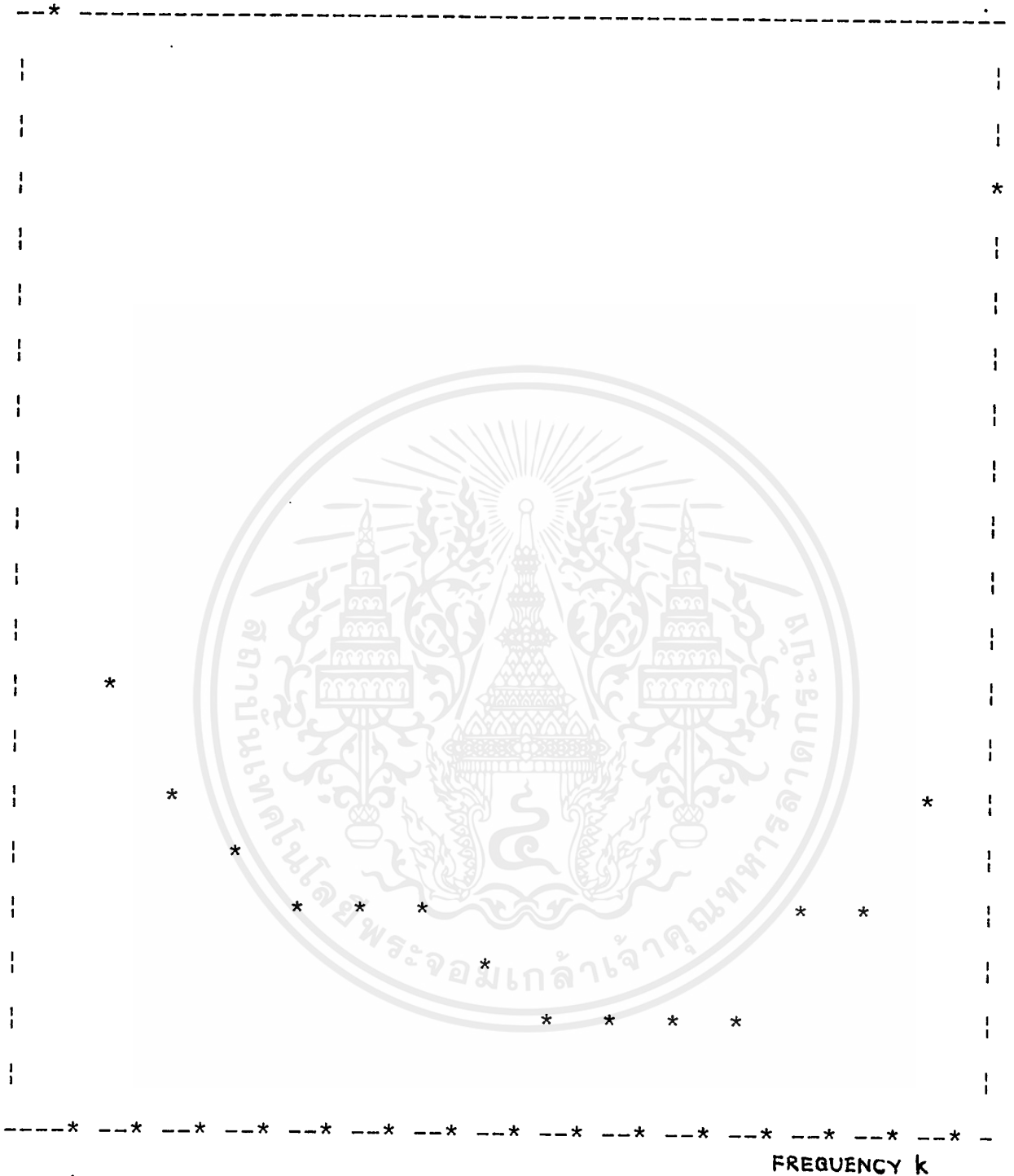
k	TMS 32010			โปรแกรมภาษาซี		
	X <sub>r</sub>	X <sub>i</sub>	Mag	X <sub>r</sub>	X <sub>i</sub>	Mag
0	3968	0	3968**	3985.0	0.0	3985.0**
1	-128	-640	672	-57.9	-578.5	581.4
2	0	0	0	0.9	-0.3	1.0
3	-128	-224	288	-57.8	-187.3	196.0
4	0	0	0	0.7	-0.7	1.0
5	-128	-128	160	-57.5	-105.8	120.4
6	0	0	0	0.3	-0.9	1.0
7	-128	-96	160	-57.1	-68.4	89.2
8	0	0	0	0.0	-1.0	1.0
9	-64	-96	96	-56.8	-45.7	72.9
10	0	0	0	-0.3	-0.9	1.0
11	-96	-64	96	-56.4	-29.6	63.7
12	0	0	0	-0.7	-0.7	1.0
13	-96	-64	96	-56.1	-16.7	58.6
14	0	0	0	-0.9	-0.3	1.0
15	-96	-32	96	-56.0	-5.4	56.2
16	0	0	0	-1.0	0.0	1.0
17	-64	0	64	-56.0	5.4	56.2
18	0	0	0	-0.9	0.3	1.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

k	TMS 32010			โปรแกรมภาษาซี		
	$X_r$	$X_i$	Mag	$X_r$	$X_i$	Mag
19	-64	32	96	-56.1	16.7	58.6
20	0	0	0	-0.7	0.7	1.0
21	-64	0	64	-56.4	29.6	63.7
22	0	0	0	-0.3	0.9	1.0
23	-64	32	96	-56.8	45.7	72.9
24	0	0	0	-0.0	1.0	1.0
25	-64	96	96	-57.1	68.4	89.2
26	0	0	0	0.3	0.9	1.0
27	-32	128	160	-57.5	105.8	120.4
28	0	0	0	0.7	0.7	1.0
29	-32	192	224	-57.8	187.3	196.0
30	0	0	0	0.9	0.3	1.0
31	-32	544	544	-57.9	578.5	581.4

**หมายเหตุ** \*\* ขนาดของ  $X(0)$  จะมีค่าสูงมากเมื่อเทียบกับที่ตำแหน่งอื่น ๆ ดังนั้น เพื่อให้สามารถมองเห็นความแตกต่างของขนาดที่ตำแหน่งอื่น ๆ เราจึงนำค่า  $X(0)$  มา plot กราฟ

SPECTRUM x(k)



magnitude response of fft

FREQUENCY k

max val= 20.40

รูปที่ 5.3 ผลการวิเคราะห์สเปกตรัมของ สัญญาณสี่เหลี่ยม ขนาด 1 V<sub>pp</sub> ความถี่ 1 KHz โดยใช้ TMS 32010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 4** เปรียบเทียบผลการวิเคราะห์สเปกตรัมของสัญญาณสามเหลี่ยม ขนาด 1 V<sub>pp</sub> ความถี่ 1 KHz โดยใช้ TMS 32010 และ โปรแกรมภาษาซี

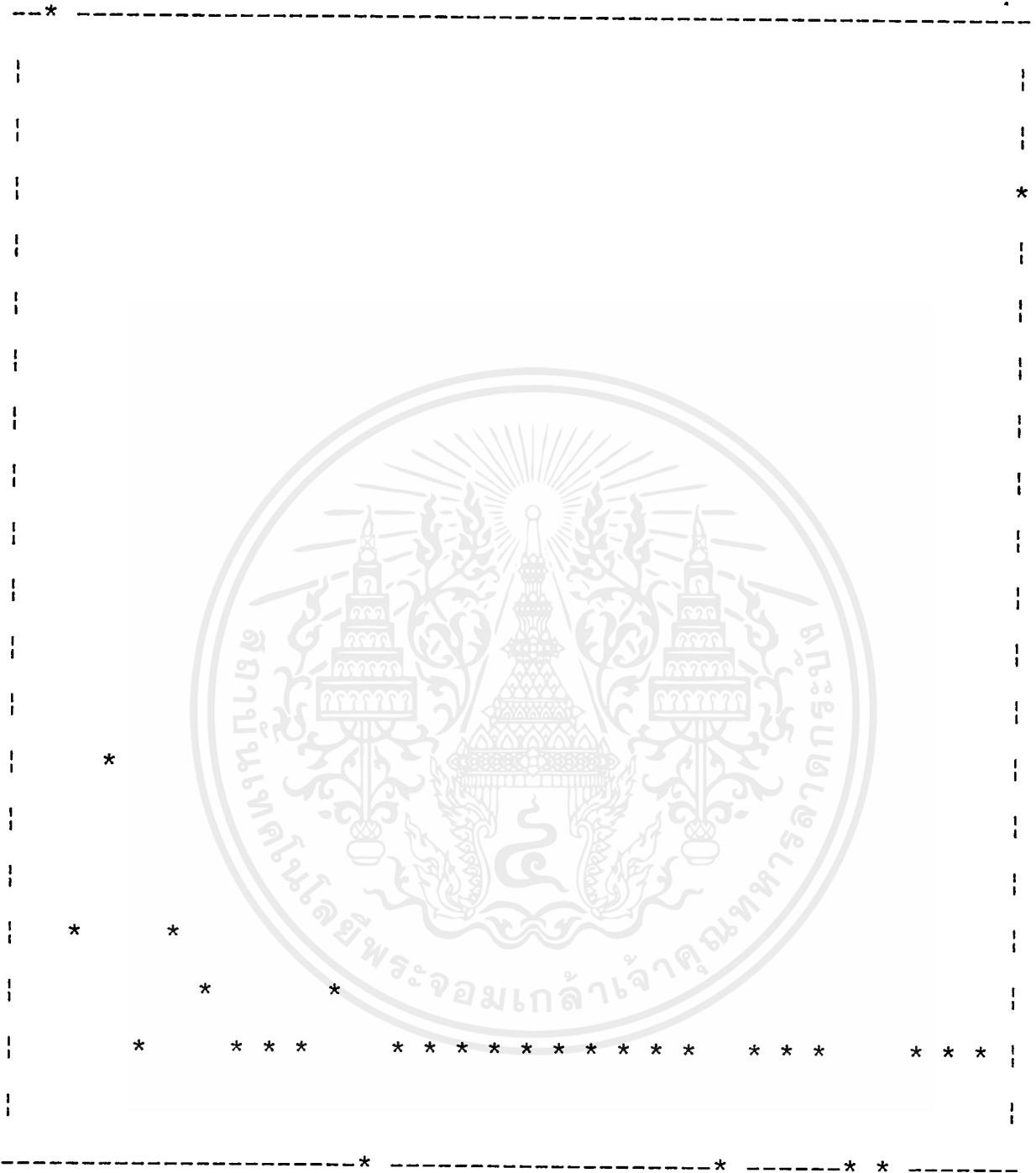
k	TMS 32010			โปรแกรมภาษาซี		
	X <sub>r</sub>	X <sub>i</sub>	Mag	X <sub>r</sub>	X <sub>i</sub>	Mag
0	3936	0	3936**	3977.0	0.0	3977.0**
1	96	-384	416	162.7	-317.7	357.0
2	-64	-32	96	-3.2	-6.1	6.9
3	-128	-32	160	-46.9	-0.4	46.9
4	-32	0	32	-7.7	-1.7	7.8
5	-64	-32	96	6.1	9.2	11.0
6	-64	0	64	-2.6	2.8	3.9
7	-32	-32	32	0.5	-14.5	14.6
8	-32	0	32	-10.0	-3.0	10.4
9	-32	-32	32	-8.2	2.8	8.6
10	-64	0	64	-10.3	3.1	10.8
11	0	0	0	2.1	8.6	8.9
12	-32	0	32	-6.2	0.2	6.2
13	-32	-32	32	-7.2	-9.9	12.3
14	0	-32	32	4.2	2.1	4.7
15	-32	-32	32	2.6	2.7	3.8
16	-32	0	32	-1.0	0.0	1.0
17	32	0	32	2.6	-2.7	3.8
18	0	32	32	4.2	-2.1	4.7

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปยังระบบออนไลน์หรือทำ  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

k	TMS 32010			โปรแกรมภาษาซี		
	$X_r$	$X_i$	Mag	$X_r$	$X_i$	Mag
19	0	32	32	-7.2	9.9	12.3
20	-32	0	32	-6.2	-0.2	6.2
21	0	-32	32	2.1	-8.6	8.9
22	0	0	0	-10.3	-3.1	10.8
23	-32	-32	32	-8.2	-2.8	8.6
24	-32	0	32	-10.0	3.0	10.4
25	32	32	32	0.5	14.5	14.6
26	0	0	0	-2.6	-2.8	3.9
27	0	0	0	6.1	-9.2	11.0
28	-32	0	32	-7.7	1.7	7.8
29	-32	-32	32	-46.9	0.4	46.9
30	0	-32	32	-3.2	6.1	6.9
31	160	288	352	162.7	317.7	357.0

หมายเหตุ \*\* ขนาดของ  $X(0)$  จะมีค่าสูงมากเมื่อเทียบกับที่ตำแหน่งอื่น ๆ ดังนั้น เพื่อให้สามารถมองเห็นความแตกต่างของขนาดที่ตำแหน่งอื่น ๆ เราจึงนำขนาด  $X(0)$  มา plot กราฟ

SPECTRUM x(k)



magnitude response of fft FREQUENCY k  
 max val= 12.37

**รูปที่ 5.4** ผลการวิเคราะห์สเปกตรัมของสัญญาณสามเหลี่ยม ขนาด 1 V<sub>pp</sub>  
 ความถี่ 1 KHz โดยใช้ TMS 32010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 5** เปรียบเทียบผลการสังเคราะห์สัญญาณชาชนัน ขนาด 1 V<sub>pp</sub>  
ความถี่ 1 KHz TMS 32010 และ โปรแกรมภาษาซี

k	TMS 32010			โปรแกรมภาษาซี		
	X <sub>r</sub>	X <sub>i</sub>	Mag	X <sub>r</sub>	X <sub>i</sub>	Mag
0	93	-12	93	93.0	-12.0	93.7
1	97	-12	97	90.5	13.6	91.5
2	106	-7	106	98.2	-0.3	98.2
3	115	-11	115	94.7	7.9	95.1
4	116	0	116	89.6	-3.4	89.7
5	122	-7	122	95.1	0.8	95.1
6	123	-5	123	100.8	5.7	101.0
7	137	-6	137	109.9	2.2	109.9
8	139	2	139	111.0	2.0	111.0
9	141	3	141	113.1	-0.7	113.1
10	145	-2	145	116.4	4.4	116.5
11	148	1	148	121.2	2.5	121.2
12	160	-1	160	125.2	3.4	125.2
13	153	-6	153	131.6	6.5	131.8
14	152	0	152	139.8	0.5	139.8
15	149	-10	149	141.7	5.1	141.8
16	149	8	149	149.0	8.0	149.2
17	143	6	143	150.2	-8.4	150.4
18	140	1	140	153.3	0.3	153.3

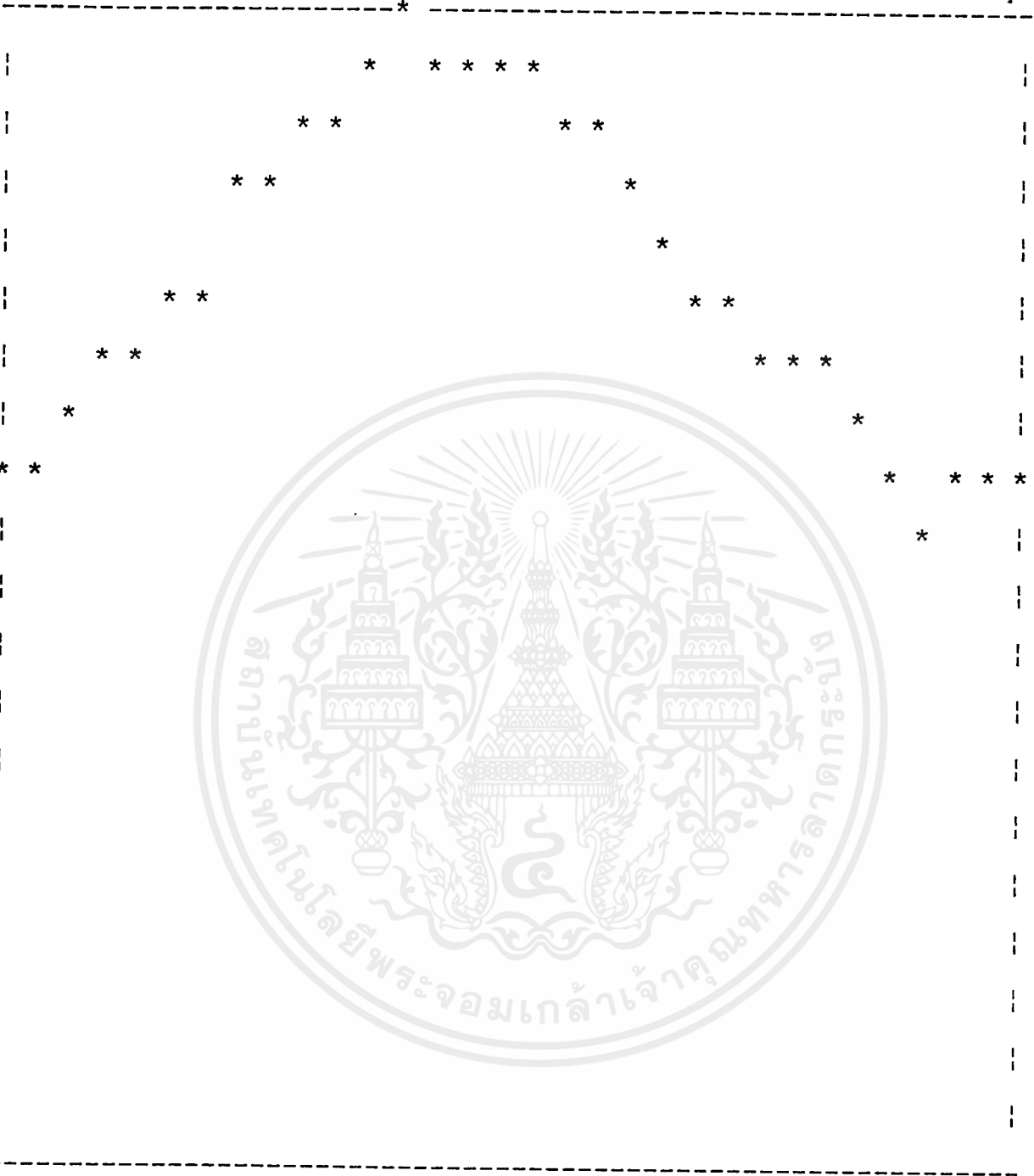
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้ประโยชน์ด้านอื่นใด

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

k	TMS 32010			โปรแกรมภาษาซี		
	$X_r$	$X_i$	Mag	$X_r$	$X_i$	Mag
19	133	7	133	153.3	-5.7	153.4
20	125	3	125	160.3	-0.5	160.3
21	122	3	122	148.6	1.3	148.6
22	117	5	117	145.1	-0.1	145.1
23	113	0	113	141.2	2.9	141.2
24	111	2	111	139.0	2.0	139.0
25	111	3	111	138.0	-4.4	138.1
26	101	6	101	123.9	-4.4	124.0
27	96	1	96	122.5	-4.7	122.6
28	89	-4	89	116.7	0.5	116.7
29	95	8	95	116.4	-8.6	116.8
30	98	0	98	106.1	-6.2	106.3
31	91	14	92	99.0	-10.3	99.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMPLITUDE X(t)



TIME t

magnitude response of fft

max val= 160.00

รูปที่ 5.5 ผลการสังเคราะห์สัญญาณชายน ขนาด 1 V<sub>pp</sub>  
 ความถี่ 1 KHz โดยใช้ TMS 32010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6 เปรียบเทียบผลการสังเคราะห์สเปกตรัมของสัญญาณชายัน ขนาด 1 V<sub>pp</sub> ความถี่ 5 KHz โดยใช้ TMS 32010 และ โปรแกรมภาษาซี

k	TMS 32010			โปรแกรมภาษาซี		
	X <sub>r</sub>	X <sub>i</sub>	Mag	X <sub>r</sub>	X <sub>i</sub>	Mag
0	105	-13	105	105.0	-13.0	105.8
1	88	-13	88	123.8	11.1	124.3
2	107	-9	107	146.2	2.9	146.2
3	137	-11	137	142.9	9.7	143.2
4	158	0	158	117.5	0.4	117.5
5	132	-8	132	- 95.3	- 6.5	95.5
6	113	-2	113	98.1	3.8	98.2
7	96	-5	96	117.9	3.0	118.0
8	99	5	99	143.0	5.0	143.0
9	119	4	119	143.7	-3.7	143.8
10	146	0	146	124.8	5.3	124.9
11	140	-1	140	99.7	0.7	99.7
12	120	-3	120	87.6	-2.4	87.6
13	96	-4	96	106.5	3.9	106.6
14	98	-5	98	131.1	6.6	131.3
15	114	-5	114	145.9	4.3	145.9
16	137	-1	137	137.0	-1.0	137.0
17	146	5	146	114.6	-4.3	114.7
18	131	7	131	98.5	-3.7	98.6

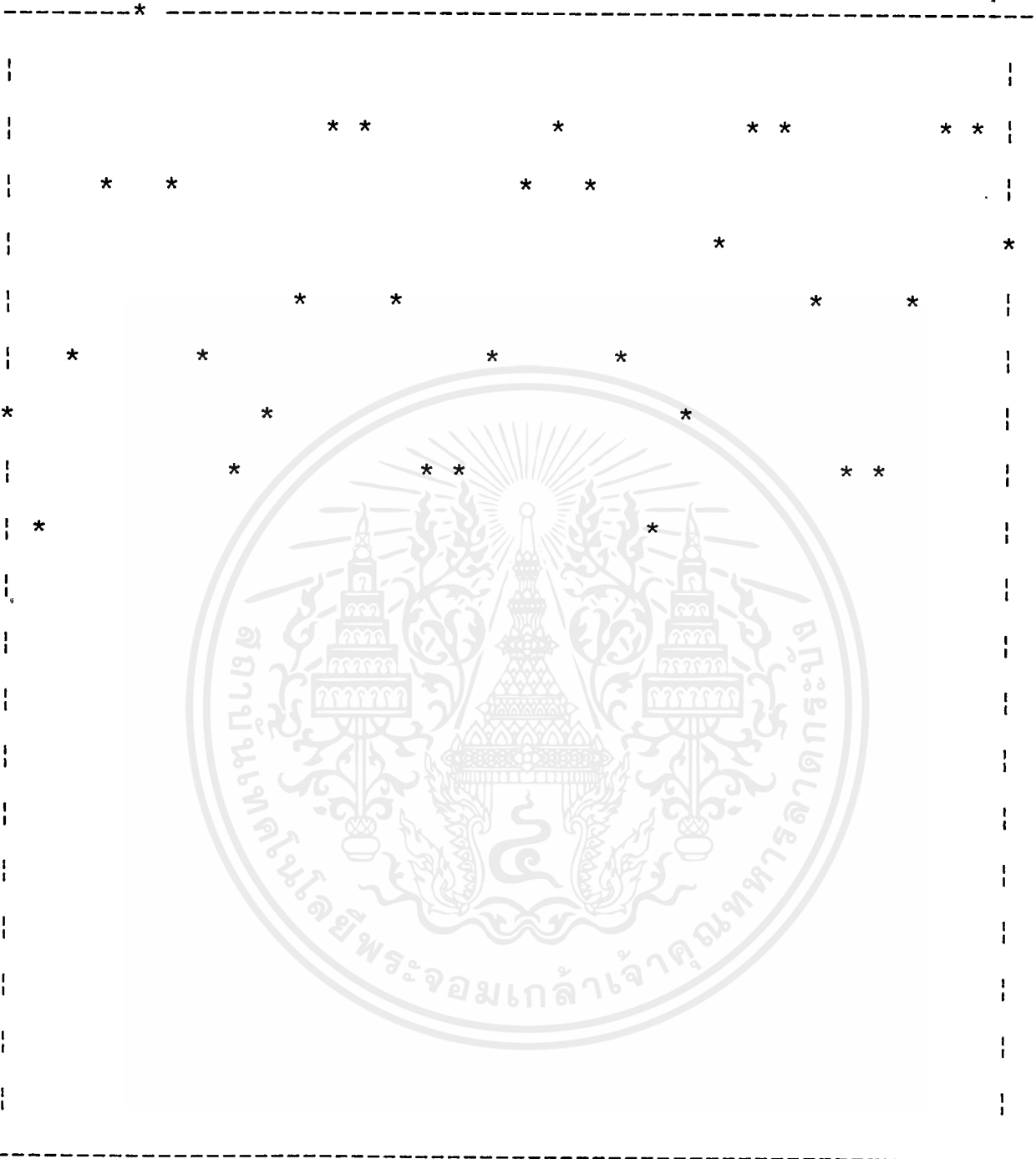
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบุคคลใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้วยวิธีใด ๆ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

k	TMS 32010			โปรแกรมภาษาซี		
	X <sub>r</sub>	X <sub>i</sub>	Mag	X <sub>r</sub>	X <sub>i</sub>	Mag
19	107	5	107	96.7	-3.9	96.8
20	87	-3	87	120.4	-2.4	120.4
21	100	2	100	140.9	-0.7	140.9
22	125	6	125	146.6	1.0	146.7
23	144	-3	144	119.5	3.7	119.6
24	143	5	143	99.0	5.0	99.1
25	119	4	119	97.7	-3.0	97.8
26	98	4	98	114.3	-0.5	114.3
27	96	7	96	132.5	-6.5	132.7
28	117	0	117	158.3	0.4	158.3
29	144	10	144	137.1	-9.7	137.4
30	146	3	146	107.9	-7.4	108.2
31	124	11	124	88.5	-11.1	89.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMPLITUDE X(t)



TIME t

magnitude response of fft

max val= 158.00

รูปที่ 5.6 ผลการสังเคราะห์สเปกตรัมของสัญญาณไซน์ ขนาด 1 V<sub>pp</sub>  
ความถี่ 5 KHz โดยาช TMS 32010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 7 เปรียบเทียบผลการสังเคราะห์สเปกตรัมของสัญญาณสี่เหลี่ยม ขนาด 1 V<sub>pp</sub>  
ความถี่ 1 KHz โดยาซ์ TMS 32010 และ โปรแกรมภาษาซี

k	TMS 32010			โปรแกรมภาษาซี		
	X <sub>r</sub>	X <sub>i</sub>	Mag	X <sub>r</sub>	X <sub>i</sub>	Mag
0	84	-10	84	84.0	-10.0	84.5
1	157	-10	157	89.1	7.8	89.5
2	153	-8	153	94.6	3.3	94.7
3	155	-11	155	94.6	0.5	94.6
4	150	-2	150	97.1	-1.4	97.1
5	152	-1	152	91.8	2.2	91.8
6	152	-4	152	95.0	1.3	95.0
7	153	-4	153	98.3	1.7	98.4
8	154	0	154	94.0	-0.0	94.0
9	149	-3	149	94.4	3.1	94.5
10	153	-2	153	95.0	2.9	95.0
11	156	-3	156	95.1	0.3	95.1
12	150	1	150	97.1	1.4	97.1
13	152	-1	152	92.4	9.9	92.9
14	153	-4	153	94.6	7.0	94.9
15	158	-8	158	89.9	8.7	90.3
16	164	10	164	164.0	10.0	164.3
17	91	10	91	158.8	-7.8	159.0
18	95	8	95	153.3	-3.3	153.3

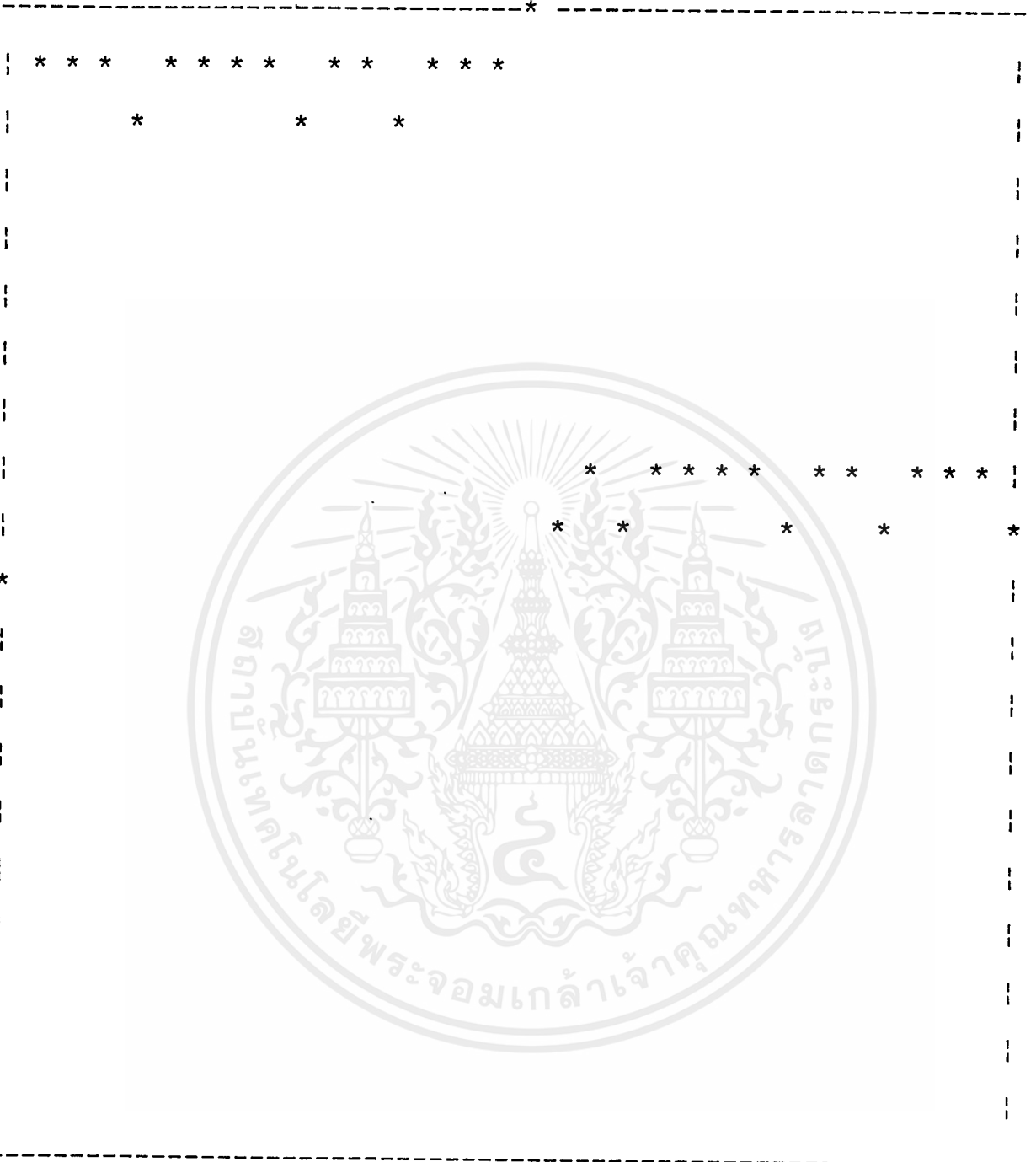
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

k	TMS 32010			โปรแกรมภาษาซี		
	$X_r$	$X_i$	Mag	$X_r$	$X_i$	Mag
19	93	11	93	153.3	-0.5	153.3
20	97	1	97	150.8	1.4	150.8
21	96	1	96	156.1	-2.2	156.1
22	96	4	96	152.9	-1.3	152.9
23	95	4	95	149.6	-1.7	149.6
24	94	0	94	154.0	0.0	154.0
25	99	3	99	153.5	-3.1	153.5
26	95	2	95	152.9	-2.9	153.0
27	92	3	92	152.8	-0.3	152.8
28	97	-2	97	150.8	-1.4	150.8
29	96	1	96	155.5	-9.9	155.8
30	95	4	95	153.3	-7.0	153.4
31	90	8	90	158.0	-8.7	158.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMPLITUDE x(t)



TIME (t)

magnitude response of fft

max val= 164.30

รูปที่ 5.7 ผลการสังเคราะห์สเปกตรัมของสัญญาณสี่เหลี่ยม ขนาด 1 V<sub>pp</sub>  
 ความถี่ 1 KHz โดยใช่ TMS 32010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 8** เปรียบเทียบผลการสังเคราะห์สเปกตรัมของสัญญาณสามเหลี่ยม ขนาด 1 V<sub>pp</sub>  
ความถี่ 1 KHz โดยใช้ TMS 32010 และ โปรแกรมภาษาซี

k	TMS 32010			โปรแกรมภาษาซี		
	X <sub>r</sub>	X <sub>i</sub>	Mag	X <sub>r</sub>	X <sub>i</sub>	Mag
0	108	-12	108	108.0	-12.0	108.6
1	132	-18	133	117.8	6.0	117.9
2	139	-10	139	126.1	2.6	126.1
3	145	-11	145	117.1	2.0	117.1
4	141	-2	141	126.0	-0.8	126.0
5	155	-2	155	112.7	6.8	112.9
6	149	-2	149	111.1	2.3	111.2
7	146	-7	146	105.7	6.3	105.9
8	144	4	144	100.0	-0.0	100.0
9	143	6	143	106.9	-0.0	106.9
10	136	2	136	93.3	1.0	93.3
11	132	-5	132	97.1	1.1	97.1
12	126	4	126	102.2	-2.0	102.2
13	125	-3	125	104.2	6.1	104.4
14	127	-3	127	107.8	3.4	107.9
15	124	-2	124	111.7	3.5	111.8
16	112	8	112	112.0	8.0	112.2
17	112	4	112	124.9	-0.7	124.9
18	109	4	109	127.5	-2.6	127.5

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปยังระบบออนไลน์ที่นอกเหนือจากนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

k	TMS 32010			โปรแกรมภาษาซี		
	$X_r$	$X_i$	Mag	$X_r$	$X_i$	Mag
19	105	7	105	125.6	-2.9	125.6
20	102	-2	102	126.0	4.8	126.0
21	97	2	97	133.0	-4.6	133.1
22	93	2	93	136.8	3.2	136.8
23	106	1	106	144.4	6.2	144.5
24	100	0	100	144.0	4.0	144.0
25	107	6	107	146.2	-5.1	146.3
26	112	2	112	149.0	-1.0	149.0
27	114	7	114	156.0	-0.2	156.0
28	126	-1	126	141.7	-2.0	141.8
29	117	3	117	145.8	-8.3	146.1
30	125	3	125	140.1	-9.1	140.4
31	118	6	118	134.0	-16.1	135.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMPLITUDE X(t)



magnitude response of fft

TIME t

max val= 155.01

รูปที่ 5.8 ผลการสังเคราะห์สเปกตรัมของสัญญาณสามเหลี่ยม ขนาด 1 V<sub>pp</sub>  
 ความถี่ 1 KHz โดยใช่ TMS 32010

สรุปผลการทดลอง

เมื่อเปรียบเทียบผลการทำ FFT บน TMS 32010 กับการใช้ภาษาซีซึ่งให้ ความแม่นยำกว่า จะพบว่าผลการทำ FFT บน TMS 32010 มีความผิดพลาด ซึ่งข้อ ผิดพลาดอาจเกิดได้จากสิ่งต่าง ๆ ตามที่ได้กล่าวไว้ในหัวข้อความผิดพลาดเนื่องจากการ ทำ FFT แต่หากเราจะไม่พิจารณาละเอียดก็สามารถอนุมานมาได้

การปฏิบัติงานทางด้านเสียง จะมีปัญหาและข้อบกพร่องเนื่องจาก

1. สัญญาณรบกวนจากการออกแบบและประกอบวงจร ส่วนที่รับค่า I/P ก่อนที่จะทำ ADC และส่วนที่เชื่อมต่อกับ DAC เพื่อเป็น O/P
2. การออกแบบการจัดสรรหน่วยความจำไม่ดีพอและยากแก่การพัฒนา เนื่องจากการประมวลผลของ FFT จำเป็นต้องใช้เวลาอยู่ช่วงหนึ่ง ดังนั้น หากไม่มี buffer memory เพื่อเก็บค่า O/P และค่า I/P จาก ADC จะทำให้การ Sampling ค่าที่จะนำมาคำนวณต่อเนื่อง

นอกจากนั้นการออกแบบ Ram pack ๖๕ ยังทำให้การพัฒนาโปรแกรมไม่ สะดวกและล่าช้า ในกรณีที่ต้องอ่านข้อมูลมาวิเคราะห์และตรวจสอบค่า

3. จำนวนบิตของ DIGITAL I/P และ O/P บกดี ADC และ DAC ที่ใช้กับงานทางเสียงได้ดีจะต้องมีขนาดใหญ่ เช่น 12 บิต หรือ 16 บิต แต่ขนาด 8 บิต ก็พอจะใช้กับเสียงพูดของคนปกติได้ แต่คุณภาพจะไม่ดีพอ

4. ขีดจำกัดความสามารถของ TMS 32010 เนื่องจาก internal ram มีค่าจำกัดเพียง 144 word ทำให้สามารถประมวลผล FFT ได้ไม่เกิน 32 จุด ทำให้ ค่าที่ได้จากการประมวลผลไม่ถูกต้องละเอียดพอ

**ภาคผนวก**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โปรแกรมย่อยในการทำ FFT และ IFFT

ฟังก์ชันหลักสำหรับการทำ FFT และ IFFT

1) การทำ Bit-Reverse

```
;BITREV PR,PI,QR,QI
      ZALH PR
      ADDS QR
      SACL PR
      SACH QR
      ZALH PI
      ADDS QI
      SACL PI
      SACH QI
; $END
```

2.1) การทำ DFT 4 จุด

```
;COMBO $MACRO R1,I1,R2,I2,R3,I3,R4,I4
; CALCULATE PARTIAL TERMS FOR R3,R4,I3,I4
      LAC R3,14 ; ACC = (1/4)(R3)
      ADD R4,14 ; ACC = (1/4)(R3+R4)
      SACH R3,1 ; R3 = (1/2)(R3+R4)
      SUB R4,15 ; ACC = (1/4)(R3+R4)-(1/2)(R4)
      SACH R4,1 ; R4 = (1/2)(R3-R4)
      LAC I3,14 ; ACC = (1/4)(I3)
      ADD I4,14 ; ACC = (1/4)(I3+I4)
      SACH I3,1 ; I3 = (1/2)(I3+I4)
      SUB I4,15 ; ACC = (1/4)(I3+I4)-(1/2)(I4)
      SACH I4,1 ; I4 = (1/2)(I3-I4)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำออกเผยแพร่ภายนอกได้  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น หากต้องการแก้ไขหรือเปลี่ยนแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; CALCULATE PARTIAL TERMS FOR R2,R4,I2,I4

LAC R1,14 ; ACC = (1/4)(R1)  
 ADD R2,14 ; ACC = (1/4)(R1+R2)  
 SACH R1,1 ; R1 = (1/2)(R1+R2)  
 SUB R2,15 ; ACC = (1/4)(R1+R2)-(1/2)(R2)  
 ADD I4,15 ; ACC = (1/4)[(R1-R2)+(I3-I4)]  
 SACH R2 ; R2 = (1/4)[(R1-R2)+(I3-I4)]  
 SUBH I4 ; ACC = (1/4)[(R1-R2)-(I3-I4)]  
 DMOV R4 ; I4 = R4 = (1/2)(R3-R4)  
 SACH R4 ; R4 = (1/4)[(R1-R2)-(I3-I4)]  
 LAC I1,14 ; ACC = (1/4)(I1)  
 ADD I2,14 ; ACC = (1/4)(I1+I2)  
 SACH I1,1 ; I1 = (1/2)(I1+I2)  
 SUB I2,15 ; ACC = (1/4)(I1+I2)-(1/2)(I2)  
 SUB I4,15 ; ACC = (1/4)[(I1-I2)-(R3-R4)]  
 SACH I2 ; I2 = (1/4)[(I1-I2)-(R3-R4)]  
 ADDH I4 ; ACC = (1/4)[(I1-I2)+(R3-R4)]  
 SACH I4 ; I4 = (1/4)[(I1-I2)+(R3-R4)]

; CALCULATE PARTIAL TERMS FOR R1,R3,I1,I3

LAC R1,15 ; ACC = (1/4)(R1+R2)  
 ADD R3,15 ; ACC = (1/4)[(R1+R2)+(R3+R4)]  
 SACH R1 ; R1 = (1/4)[(R1+R2)+(R3+R4)]  
 SUBH R3 ; ACC = (1/4)[(R1+R2)-(R3+R4)]  
 SACH R3 ; R3 = (1/4)[(R1+R2)-(R3+R4)]  
 LAC I1,15 ; ACC = (1/4)(I1+I2)  
 ADD I3,15 ; ACC = (1/4)[(I1+I2)+(I3+I4)]  
 SACH I1 ; I1 = (1/4)[(I1+I2)+(I3+I4)]  
 SUBH I3 ; ACC = (1/4)[(I1+I2)-(I3+I4)]

SACH I3 ; I3 = (1/4)[(I1+I2)-(I3+I4)]

; \$END

2.2) การทำ IDFT 4 จุด

; COMBO \$MACRO R1,I1,R2,I2,R3,I3,R4,I4

; CALCULATE PARTIAL TERMS FOR R3,R4,I3,I4

LAC R3,14 ; ACC = (1/4)(R3)  
 ADD R4,14 ; ACC = (1/4)(R3+R4)  
 SACH R3,1 ; R3 = (1/2)(R3+R4)  
 SUB R4,15 ; ACC = (1/4)(R3+R4)-(1/2)(R4)  
 SACH R4,1 ; R4 = (1/2)(R3-R4)  
 LAC I3,14 ; ACC = (1/4)(I3)  
 ADD I4,14 ; ACC = (1/4)(I3+I4)  
 SACH I3,1 ; I3 = (1/2)(I3+I4)  
 SUB I4,15 ; ACC = (1/4)(I3+I4)-(1/2)(I4)  
 SACH I4,1 ; I4 = (1/2)(I3-I4)

; CALCULATE PARTIAL TERMS FOR R2,R4,I2,I4

LAC R1,14 ; ACC = (1/4)(R1)  
 ADD R2,14 ; ACC = (1/4)(R1+R2)  
 SACH R1,1 ; R1 = (1/2)(R1+R2)  
 SUB R2,15 ; ACC = (1/4)(R1+R2)-(1/2)(R2)  
 SUB I4,15 ; ACC = (1/4)[(R1-R2)-(I3-I4)]  
 SACH R2 ; R2 = (1/4)[(R1-R2)-(I3-I4)]  
 ADDH I4 ; ACC = (1/4)[(R1-R2)+(I3-I4)]  
 DMOV R4 ; I4 = R4 = (1/2)(R3-R4)  
 SACH R4 ; R4 = (1/4)[(R1-R2)+(I3-I4)]  
 LAC I1,14 ; ACC = (1/4)(I1)  
 ADD I2,14 ; ACC = (1/4)(I1+I2)  
 SACH I1,1 ; I1 = (1/2)(I1+I2)  
 SUB I2,15 ; ACC = (1/4)(I1+I2)-(1/2)(I2)  
 ADD I4,15 ; ACC = (1/4)[(I1-I2)+(R3-R4)]  
 SACH I2 ; I2 = (1/4)[(I1-I2)+(R3-R4)]  
 SUBH I4 ; ACC = (1/4)[(I1-I2)-(R3-R4)]

```

SACH I4 ; I4 = (1/4)[(I1-I2)-(R3-R4)]
; CALCULATE PARTIAL TERMS FOR R1,R3,I1,I3
LAC R1,15 ; ACC = (1/4)(R1+R2)
ADD R3,15 ; ACC = (1/4)[(R1+R2)+(R3+R4)]
SACH R1 ; R1 = (1/4)[(R1+R2)+(R3+R4)]
SUBH R3 ; ACC = (1/4)[(R1+R2)-(R3+R4)]
SACH R3 ; R3 = (1/4)[(R1+R2)-(R3+R4)]
LAC I1,15 ; ACC = (1/4)(I1+I2)
ADD I3,15 ; ACC = (1/4)[(I1+I2)+(I3+I4)]
SACH I1 ; I1 = (1/4)[(I1+I2)+(I3+I4)]
SUBH I3 ; ACC = (1/4)[(I1+I2)-(I3+I4)]
SACH I3 ; I3 = (1/4)[(I1+I2)-(I3+I4)]
; $END

```

3)การทํา Butterfly โดยซ้ำ  $W = 1$

การทํางานจะมีค่าสิ่งเหมือนกันทั้ง DFT และ IDFT

```

;ZERO $MACRO PR,PI,QR,QI
; CALCULATE Re[P+Q] AND Re[P-Q]
LAC PR,15 ; ACC = (1/2)(PR)
ADD QR,15 ; ACC = (1/2)(PR+QR)
SACH PR ; PR = (1/2)(PR+QR)
SUBH QR ; ACC = (1/2)(PR+QR)-(QR)
SACH QR ; PR = (1/2)(PR-QR)
; CALCULATE Im[P+Q] AND Im[P-Q]
LAC PI,15 ; ACC = (1/2)(PI)
ADD QI,15 ; ACC = (1/2)(PI+QI)
SACH PI ; PR = (1/2)(PI+QI)
SUBH QI ; ACC = (1/2)(PI+QI)-(QI)

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ใช่ว่ากรณินี้ใดๆ ทั้งสิ้นห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\$END

4.1) การทำ Butterfly สำหรับ DFT โดยให้  $W = (1/\sqrt{2}) - j(1/\sqrt{2})$

```

;PIBY4 $MACRO PR,PI,QR,QI,W
    LT      W      ; T-REGISTER=W=COS(PI/4)=SIN(PI/4)
    LAC     QI,14   ; ACC = (1/4)(QI)
    SUB     QR,14   ; ACC = (1/4)(QI-QR)
    SACH    QI,1    ; QI = (1/2)(QI-QR)
    ADD     QR,15   ; ACC = (1/4)(QI+QR)
    SACH    QR,1    ; QR = (1/2)(QI+QR)
    LAC     PR,14   ; ACC = (1/4)(PR)
    MPY     QR      ; P-REGISTER = (1/4)(QI+QR)*W
    APAC    ; ACC = (1/4)[PR+(QI+QR)*W]
    SACH    PR,1    ; PR = (1/2)[PR+(QI+QR)*W]
    SPAC    ; ACC = (1/4)(PR)
    SPAC    ; ACC = (1/4)[PR-(QI+QR)*W]
    SACH    QR,1    ; QR = (1/2)[PR-(QI+QR)*W]
    LAC     PI,14   ; ACC = (1/4)(PI)
    MPY     QI      ; P-REGISTER = (1/4)(QI-QR)*W
    APAC    ; ACC = (1/4)[PI+(QI-QR)*W]
    SACH    PI,1    ; PI = (1/2)[PI+(QI-QR)*W]
    SPAC    ; ACC = (1/4)(PI)
    SPAC    ; ACC = (1/4)[PI-(QI-QR)*W]
    SACH    QI,1    ; QI = (1/2)[PI-(QI-QR)*W]
; $END
    
```

4.2) การทำ Butterfly สำหรับ IDFT โดยให้  $W = (1/\sqrt{2}) - j(1/\sqrt{2})$

```

;PIBY4 $MACRO PR,PI,QR,QI,W
    LT      W      ; T-REGISTER=W=COS(PI/4)=SIN(PI/4)
    LAC     QI,14   ; ACC = (1/4)(QI)
    
```

เอกสารนี้เป็น LAC ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADD      QR,14      ; ACC = (1/4)(QI+QR)
SACH     QI,1       ; QI  = (1/2)(QI+QR)
SUB      QR,15      ; ACC = (1/4)(QI-QR)
SACH     QR,1       ; QR  = (1/2)(QI-QR)
LAC      PR,14     ; ACC = (1/4)(PR)
MPY      QR         ; P-REGISTER = (1/4)(QI-QR)*W
SPAC     ; ACC = (1/4)[PR-(QI-QR)*W]
SACH     PR,1       ; PR  = (1/2)[PR-(QI-QR)*W]
APAC     ; ACC = (1/4)(PR)
APAC     ; ACC = (1/4)[PR+(QI-QR)*W]
SACH     QR,1       ; QR  = (1/2)[PR+(QI-QR)*W]
LAC      PI,14     ; ACC = (1/4)(PI)
MPY      QI         ; P-REGISTER = (1/4)(QI+QR)*W
APAC     ; ACC = (1/4)[PI+(QI+QR)*W]
SACH     PI,1       ; PI  = (1/2)[PI+(QI+QR)*W]
SPAC     ; ACC = (1/4)(PI)
SPAC     ; ACC = (1/4)[PI-(QI+QR)*W]
SACH     QI,1       ; QI  = (1/2)[PI-(QI+QR)*W]
;
$END

```

5.1) การทำ Butterfly สำหรับ DFT โดยใช้  $W = -j$

```

;PIBY2 $MACRO PR,PI,QR,QI

```

```

; CALCULATE Re[P+jQ] AND Re[P-jQ]

```

```

LAC      PI,15     ; ACC = (1/2)(PI)

```

```

SUB      QR,15     ; ACC = (1/2)(PI-QR)

```

```

SACH     PI        ; PI  = (1/2)(PI-QR)

```

```

ADDH     QR        ; ACC = (1/2)(PI-QR)+(QR)

```

```

SACH     QR        ; QR  = (1/2)(PI+QR)

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;      CALCULATE Im[P+jQ] AND Im[P-jQ]
LAC      PR,15      ; ACC = (1/2)(PR)
ADD      QI,15      ; ACC = (1/2)(PR+QI)
SACH     PR         ; PI  = (1/2)(PR+QI)
SUBH     QI         ; ACC = (1/2)(PR+QI)-(QI)
DMOV     QR         ; QR  -> QI
SACH     QR         ; QR  = (1/2)(PR-QI)
;      $END

```

5.2) การทำ Butterfly สำหรับ IDFT โดยใช้  $W = -j$

```

;PIBY2 $MACRO PR,PI,QR,QI
;      CALCULATE Re[P+jQ] AND Re[P-jQ]
LAC      PI,15      ; ACC = (1/2)(PI)
ADD      QR,15      ; ACC = (1/2)(PI+QR)
SACH     PI         ; PI  = (1/2)(PI+QR)
SUBH     QR         ; ACC = (1/2)(PI+QR)-(QR)
SACH     QR         ; QR  = (1/2)(PI-QR)
;      CALCULATE Im[P+jQ] AND Im[P-jQ]
LAC      PR,15      ; ACC = (1/2)(PR)
SUB      QI,15      ; ACC = (1/2)(PR-QI)
SACH     PR         ; PI  = (1/2)(PR-QI)
ADDH     QI         ; ACC = (1/2)(PR-QI)+(QI)
DMOV     QR         ; QR  -> QI
SACH     QR         ; QR  = (1/2)(PR+QI)
;      $END

```

6.1) การทำ Butterfly สำหรับ DFT โดยให้  $W = -(1/\sqrt{2}) - j(1/\sqrt{2})$

```

;PI3BY4 $MACRO PR,PI,QR,QI,W

LT W ; T-REGISTER=W=COS(PI/4)=SIN(PI/4)
LAC QI,14 ; ACC = (1/4)(QI)
SUB QR,14 ; ACC = (1/4)(QI-QR)
SACH QI,1 ; QI = (1/2)(QI-QR)
ADD QR,15 ; ACC = (1/4)(QI+QR)
SACH QR,1 ; QR = (1/2)(QI+QR)
LAC PR,14 ; ACC = (1/4)(PR)
MPY QI ; P-REGISTER = (1/4)(QI-QR)*W
APAC ; ACC = (1/4)[PR+(QI-QR)*W]
SACH PR,1 ; PR = (1/2)[PR+(QI-QR)*W]
SPAC ; ACC = (1/4)(PR)
SPAC ; ACC = (1/4)[PR-(QI-QR)*W]
MPY QR ; P-REGISTER = (1/4)(QI+QR)*W
SACH QR,1 ; QR = (1/2)[PR-(QI-QR)*W]
LAC PI,14 ; ACC = (1/4)(PI)
SPAC ; ACC = (1/4)[PI-(QI+QR)*W]
SACH PI,1 ; PI = (1/2)[PI-(QI+QR)*W]
APAC ; ACC = (1/4)(PI)
APAC ; ACC = (1/4)[PI+(QI+QR)*W]
SACH PI,1 ; QI = (1/2)[PI+(QI+QR)*W]

$END
    
```

6.2) การทำ Butterfly สำหรับ IDFT โดยให้  $W = -(1/\sqrt{2}) - j(1/\sqrt{2})$

```

;PI3BY4 $MACRO PR,PI,QR,QI,W

LT W ; T-REGISTER=W=COS(PI/4)=SIN(PI/4)
LAC QI,14 ; ACC = (1/4)(QI)
    
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SUB      QR,14      ; ACC = (1/4)(QI-QR)
SACH    QI,1       ; QI  = (1/2)(QI-QR)
ADD     QR,15      ; ACC = (1/4)(QI+QR)
SACH    QR,1       ; QR  = (1/2)(QI+QR)
LAC     PR,14      ; ACC = (1/4)(PR)
MPY     QR         ; P-REGISTER = (1/4)(QI+QR)*W
SPAC                    ; ACC = (1/4)[PR-(QI+QR)*W]
SACH    PR,1       ; PR  = (1/2)[PR-(QI+QR)*W]
APAC                    ; ACC = (1/4)(PR)
APAC                    ; ACC = (1/4)[PR+(QI+QR)*W]
MPY     QI         ; P-REGISTER = (1/4)(QI-QR)*W
SACH    QR,1       ; QR  = (1/2)[PR+(QI+QR)*W]
LAC     PI,14      ; ACC = (1/4)(PI)
SPAC                    ; ACC = (1/4)[PI-(QI-QR)*W]
SACH    PI,1       ; PI  = (1/2)[PI-(QI-QR)*W]
APAC                    ; ACC = (1/4)(PI)
APAC                    ; ACC = (1/4)[PI+(QI-QR)*W]
SACH    PI,1       ; QI  = (1/2)[PI+(QI-QR)*W]
;
$END

```

7.1)การทํา Butterfly สำหรับ DFT โดยใช่  $W = \cos(2\pi k/N) - j\sin(2\pi k/N)$

```
$BTRFL $MACRO PR,PI,QR,QI,WR,WI
```

```
;$CALCULATE QR*WR+QI*WI AND STORE INTO QI
```

```
LT      QI          ; T-REGISTER = QI
```

```
MPYK    WI          ; P-REGISTER = (1/16)(QI*WI)
```

```
ZAC                    ; ACC = 0
```

```
LTA     QR          ; ACC = (1/16)(QI*WI);LOAD T WITH QR
```

```
MPYK    WR          ; P-REGISTER = (1/16)(QR*WR)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่หรือใช้ซ้ำโดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

APAC                                ; ACC = (1/16)(QR*WR+QI*WI)
SACH  QR,4                          ; QR = (QR*WR+QI*WI)
LAC  QR,15                          ; ACC = (1/2)(QR*WR+QI*WI)
SACH  QR                             ; QR = (1/2)(QR*WR+QI*WI)
;CALCULATE  QI*WR-QR*WI AND STORE RESULT IN QR
MPYK  -WI                            ; P-REGISTER = (1/16)(-QR*WI)
ZAC                                       ; ACC = 0
LTA  QI                              ; ACC = (1/16)(-QR*WI);LOAD T WITH QI
MPYK  WR                             ; P-REGISTER = (1/16)(QI*WR)
APAC                                ; ACC = (1/16)(QI*WR-QR*WI)
SACH  QI,4                          ; QI = (QI*WR-QR*WI)
LAC  QI,15                          ; ACC = (1/2)(QI*WR-QR*WI)
SACH  QI                             ; QI = (1/2)(QI*WR-QR*WI)
;CALCULATE RE(P+JQ) & RE(P-JQ) STORE RESULT IN PR & QR
LAC  PR,14                          ; ACC = (1/4)PR
ADD  QR,15                          ; ACC = (1/4)[PR+(QR*WR+QI*WI)]
SACH  PR,1                          ; PR = (1/2)[PR+(QR*WR+QI*WI)]
SUBH  QR                             ; ACC = (1/4)[PR-(QR*WR+QI*WI)]
SACH  QR,1                          ; PR = (1/2)[PR-(QR*WR+QI*WI)]
;CALCULATE IM(P+JQ) & IM(P-JQ) STORE RESULT IN PI & QI
LAC  PI,14                          ; ACC = (1/4)PI
ADD  QI,15                          ; ACC = (1/4)[PI+(QI*WR-QR*WI)]
SACH  PI,1                          ; PR = (1/2)[PR+(QI*WR-QR*WI)]
SUBH  QI                             ; ACC = (1/4)[PR-(QI*WR-QR*WI)]
SACH  QI,1                          ; PR = (1/2)[PR-(QI*WR-QR*WI)]
;
$END

```

7.2) การทำ Butterfly สำหรับ IDFT โดยให้  $W = \cos(2\pi k/N) - j\sin(2\pi k/N)$

;BTRFL \$MACRO PR,PI,QR,QI,WR,WI

;CALCULATE QR\*WR+QI\*WI AND STORE INTO QI

LT QI ; T-REGISTER = QI

MPYK -WI ; P-REGISTER = (1/16)(-QI\*WI)

ZAC ; ACC = 0

LTA QR ; ACC = (1/16)(-QI\*WI);LOAD T WITH QR

MPYK WR ; P-REGISTER = (1/16)(QR\*WR)

APAC ; ACC = (1/16)(QR\*WR-QI\*WI)

SACH QR,4 ; QR = (QR\*WR-QI\*WI)

LAC QR,15 ; ACC = (1/2)(QR\*WR-QI\*WI)

SACH QR ; QR = (1/2)(QR\*WR-QI\*WI)

;CALCULATE QI\*WR+QR\*WI AND STORE RESULT IN QR

MPYK WI ; P-REGISTER = (1/16)(QR\*WI)

ZAC ; ACC = 0

LTA QI ; ACC = (1/16)(QR\*WI);LOAD T WITH QI

MPYK WR ; P-REGISTER = (1/16)(QI\*WR)

APAC ; ACC = (1/16)(QI\*WR+QR\*WI)

SACH QI,4 ; QI = (QI\*WR+QR\*WI)

LAC QI,15 ; ACC = (1/2)(QI\*WR+QR\*WI)

SACH QI ; QI = (1/2)(QI\*WR+QR\*WI)

;CALCULATE RE(P+JQ) & RE(P-JQ) STORE RESULT IN PR & QR

LAC PR,14 ; ACC = (1/4)PR

ADD QR,15 ; ACC = (1/4)[PR+(QR\*WR-QI\*WI)]

SACH PR,1 ; PR = (1/2)[PR+(QR\*WR-QI\*WI)]

SUBH QR ; ACC = (1/4)[PR-(QR\*WR+QI\*WI)]

SACH QR,1 ; QR = (1/2)[PR-(QR\*WR+QI\*WI)]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;CALCULATE IM(P+JQ) & IM(P-JQ) STORE RESULT IN PI & QI

```
LAC    PI,14          ; ACC = (1/4)PI
ADD    QI,15          ; ACC = (1/4)[PI+(QI*WR-QR*WI)]
SACH   PI,1           ; PR = (1/2)[PR+(QI*WR-QR*WI)]
SUBH   QI              ; ACC = (1/4)[PR-(QI*WR-QR*WI)]
SACH   QI,1           ; QI = (1/2)[PR-(QI*WR+QR*WI)]
;      $END
```

การเก็บผลลัพธ์ที่ได้ลงใน RAM

;STORE RESULT->RAM

```
LARK ARO,FDATIND      ;ARO INDEX XkREAL/IMG
LARK AR1,DATALL       ;LOOP 64 TIMES
STORFFT: LARP ARO
LAC DATINA
TBLW *+               ;WRITE XkR/I->pma
ADD ONEA
SACL DATINA
LARP AR1
BANZ STORFFT
;      $END
```

การหาค่าสัมบูรณ์ของเลขฐาน 16

```
CONST: LACK 80H
SACL IFNEG
LAC IFNEG,8
SACL IFNEG           ;(IFNEG) = 1000 0000 0000 0000 = 800H
LACK 10H
SACL INVRT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ZAC

SACL XOR

;BEGIN CALCULATE AND STORE FOR SCOPE DISPLAY

BSCP: LARK ARO,XOR

LARK AR1,63

N\_P: LARP 0

LAC \*

AND IFNEG

BZ POS

NEG: LAC INVRT,12 ; (INVRT) = 0000 0000 0001 0000 = 10H

SUB \*

SACL \*

POS: ZAC

LACK OFFH

AND \*

SACL \*+

MAR \*,AR1

BANZ N\_P

; \$END

การหาค่ารากที่สองที่เป็นบวกของจำนวนเชิงซ้อน

; FUNCTION FOR FIND SQUARE ROOT

SQRT: LARP ARO

LARK ARO,0H ;AR0 POINT Xr/i(k)

SAR ARO,COUNT1

LARP AR1

LARK AR1,31 ;AR1 POINT

S1: SAR AR1,COUNT

LARP ARO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LAR ARO, COUNT1 ;ARO POINT Xr/i(k)

ZAC

LT \*

MPY \*+

APAC

LT \*

MPY \*+

APAC

SAR ARO, COUNT1

SACL SUM ;SUM=Xr\*Xr + Xi\*Xi

LACK 80H

SACL HALF

SACL SFT

LARP AR1

LARK AR1, 6H ;AR1 COUNT FIND LOOP

S2: LAC SFT, 15

SACH SFT

ZAC

LT HALF

MPY HALF

APAC

SUB SUM

BZ EQQ

BGZ G

BLZ L

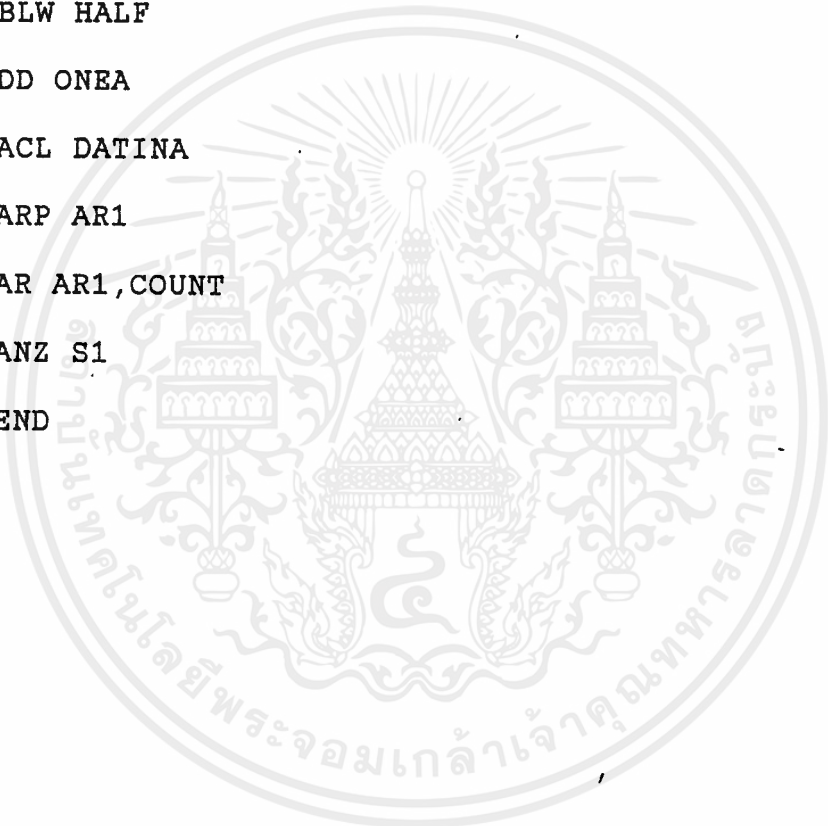
G: LAC HALF

SUB SFT

เอกสารนี้เป็น **SACL HALF** ที่จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B LL

L: LAC HALF  
ADD SFT  
SACL HALF  
LL: BANZ S2  
EQQ: LAC HALF,3  
SACL HALF  
LAC DATINA  
TBLW HALF  
ADD ONEA  
SACL DATINA  
LARP AR1  
LAR AR1,COUNT  
BANZ S1  
; \$END



โปรแกรมหลักในการทำ FFT 32 จุด

\*\*หมายเหตุ การรหัส C32 คอมพิวเตอร์โปรแกรมไม่สามารถเขียน source code เป็นการเรียกใช้ฟังก์ชันได้ จะต้องเขียนแต่ละคำสั่งโดยตรง

; BIT REVERSE I/P OF FFT 32 POINT

BITREV X1R,X1I,X16R,X16I

BITREV X2R,X2I,X8R,X8I

BITREV X3R,X3I,X24R,X24I

BITREV X5R,X5I,X20R,X20I

BITREV X6R,X6I,X12R,X12I

BITREV X7R,X7I,X28R,X28I

BITREV X9R,X9I,X18R,X18I

BITREV X11R,X11I,X26R,X26I

BITREV X13R,X13I,X22R,X22I

BITREV X15R,X15I,X30R,X30I

BITREV X19R,X19I,X25R,X25I

BITREV X23R,X23I,X29R,X29I

; CALCULATE FFT 32 POINT STAGE 1&2

COMBO X0R,X0I,X1R,X1I,X2R,X2I,X3R,X3I

COMBO X4R,X4I,X5R,X5I,X6R,X6I,X7R,X7I

COMBO X8R,X8I,X9R,X9I,X10R,X10I,X11R,X11I

COMBO X12R,X12I,X13R,X13I,X14R,X14I,X15R,X15I

COMBO X16R,X16I,X17R,X17I,X18R,X18I,X19R,X19I

COMBO X20R,X20I,X21R,X21I,X22R,X22I,X23R,X23I

COMBO X24R,X24I,X25R,X25I,X26R,X26I,X27R,X27I

COMBO X28R,X28I,X29R,X29I,X30R,X30I,X31R,X31I

; CALCULATE FFT 32 POINT STAGE 3

ZERO	X0R,X0I,X4R,X4I
PIBY4	X1R,X1I,X5R,X5I,W
PIBY2	X2R,X2I,X6R,X6I
PI3BY4	X3R,X3I,X7R,X7I,W
ZERO	X8R,X8I,X12R,X12I
PIBY4	X9R,X9I,X13R,X13I,W
PIBY2	X10R,X10I,X14R,X14I
PI3BY4	X11R,X11I,X15R,X15I,W
ZERO	X16R,X16I,X20R,X20I
PIBY4	X17R,X17I,X21R,X21I,W
PIBY2	X18R,X18I,X22R,X22I
PI3BY4	X19R,X19I,X23R,X23I,W
ZERO	X24R,X24I,X28R,X28I
PIBY4	X25R,X25I,X29R,X29I,W
PIBY2	X26R,X26I,X30R,X30I
PI3BY4	X27R,X27I,X31R,X31I,W

; CALCULATE FFT 32 POINT STAGE 4

ZERO	X0R,X0I,X8R,X8I	W0
BTRFL	X1R,X1I,X9R,X9I,C016,S016	W2
PIBY4	X2R,X2I,X10R,X10I,W	W4
BTRFL	X3R,X3I,X11R,X11I,C048,S016	W6
PIBY2	X4R,X4I,X12R,X12I	W8
BTRFL	X5R,X5I,X13R,X13I,C080,S080	W10
PI3BY4	X6R,X6I,X14R,X14I,W	W12
BTRFL	X7R,X7I,X15R,X15I,C112,S112	W14

เอกสารนี้เป็น ZERO ที่สงวนไว้สำหรับ X16R,X16I,X24R,X24I นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อฝ่ายวิชาการนำใบใช้

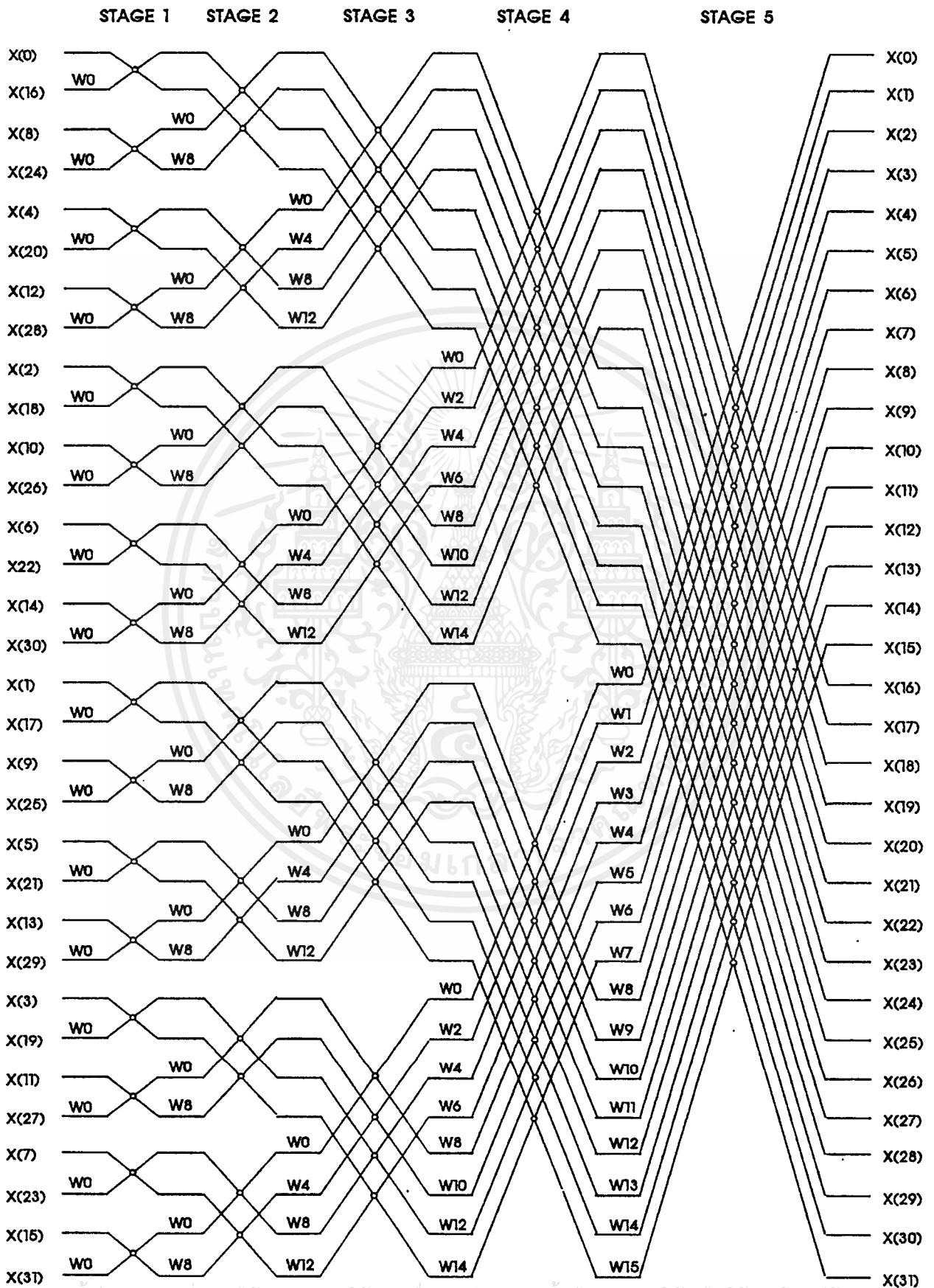
BTRFL	X17R,X17I,X25R,X25I,C016,S016	W2
-------	-------------------------------	----

PIBY4	X18R,X18I,X26R,X26I,W	w 4
BTRFL	X19R,X19IX,X27R,X27I,C048,S016	w 6
PIBY2	X20R,X20I,X28R,X28I	w 8
BTRFL	X21R,X21I,X29R,X29I,C080,S080	w 10
PI3BY4	X22R,X22I,X30R,X30I,W	w 12
BTRFL	X23R,X23I,X31R,X31I,C112,S112	w 14

; CALCULATE FFT 32 POINT STAGE 5

ZERO	X0R,X0I,X16R,X16I	w 0
BTRFL	X1R,X1I,X17R,X17I,C008,S008	w 1
BTRFL	X2R,X2I,X18R,X18I,C016,S016	w 2
BTRFL	X3R,X3I,X19R,X19I,C024,S024	w 3
PIBY4	X4R,X4I,X20R,X20I,W	w 4
BTRFL	X5R,X5I,X21R,X21I,C040,S040	w 5
BTRFL	X6R,X6I,X22R,X22I,C048,S048	w 6
BTRFL	X7R,X7I,X23R,X23I,C056,S056	w 7
PIBY2	X8R,X8I,X24R,X24I	w 8
BTRFL	X9R,X9I,X25R,X25I,C072,S072	w 9
BTRFL	X10R,X10I,X26R,X26I,C080,S080	w 10
BTRFL	X11R,X11I,X27R,X27I,C088,S088	w 11
PI3BY4	X12R,X12I,X28R,X28I,W	w 12
BTRFL	X13R,X13I,X29R,X29I,C104,S104	w 13
BTRFL	X14R,X14I,X30R,X30I,C112,S112	w 14
BTRFL	X15R,X15I,X31R,X31I,C120,S120	w 15

; \$END



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

การคำนวณหาขนาดของจำนวนเชิงซ้อนโดยใช้ TMS 32010

จากสูตรการหาขนาดของจำนวนเชิงซ้อน

กำหนด  $Z = a + bi$

จะได้ว่า  $|z| = \sqrt{a^2 + b^2}$

จะเห็นว่าในการหาขนาดของจำนวนเชิงซ้อน ประกอบด้วยขั้นตอนย่อย 3 ขั้นตอนคือ

- การหาค่ากำลังสองของจำนวนจริง
- การบวกของจำนวนจริง
- การหาค่ารากที่สองที่เป็นบวกของจำนวนจริงบวก

การหาค่ากำลังสองของจำนวนจริง

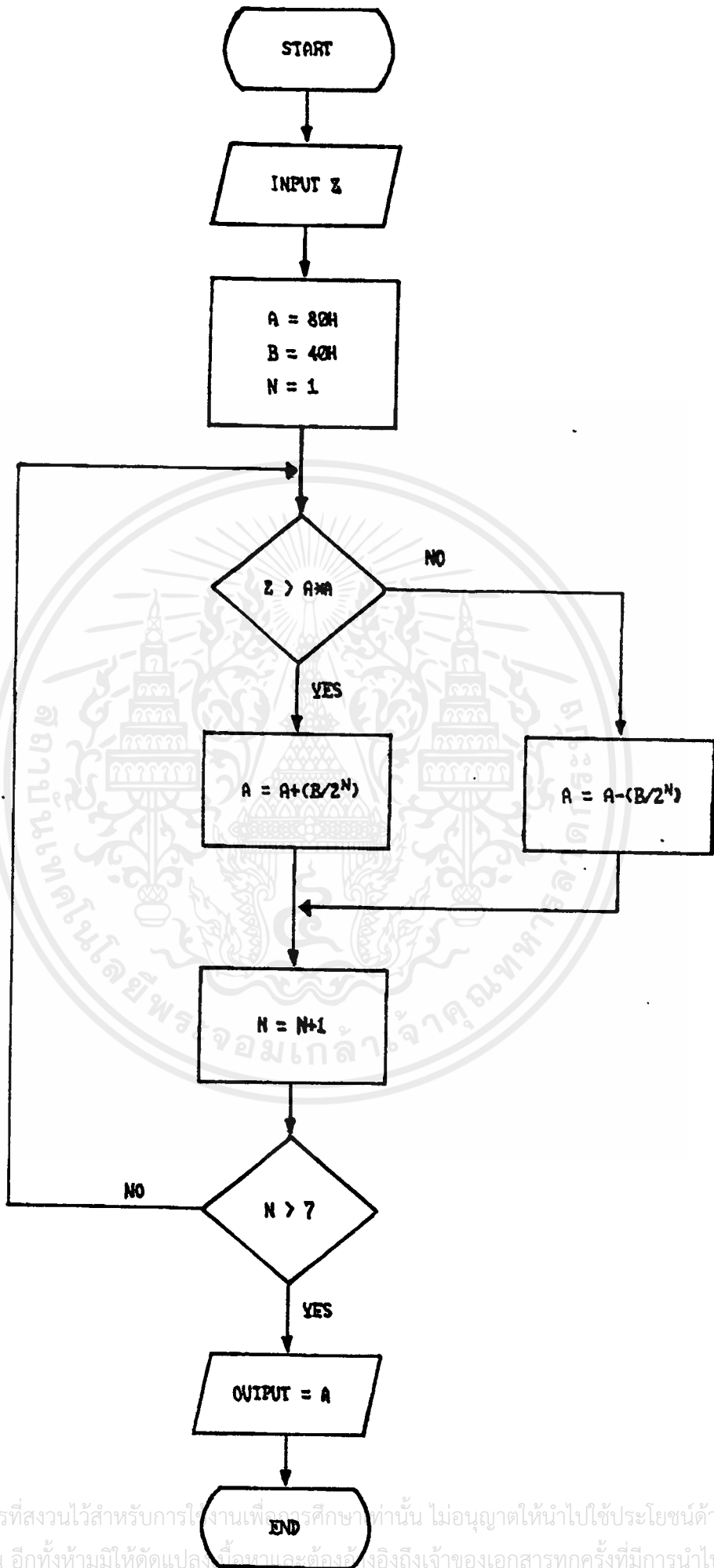
เนื่องจาก TMS 32010 ไม่มีคำสั่งในการหาค่ากำลังสอง แต่จากสมการทางคณิตศาสตร์ เรามาทราบว่า  $a^2 = a * a$  และ TMS 32010 มีคำสั่งในการคูณ แต่ถ้า  $a < 0$  อาจจะทำให้การคูณเกิด Overflow เราสามารถป้องกันปัญหานี้ได้ โดยใช้หลักการที่ว่า  $a * a = (-a) * (-a)$  ดังนั้นถ้า  $a < 0$  เราจะแทน  $a$  ด้วย  $-a$

การบวกของจำนวนจริง

TMS 32010 มีคำสั่งในการบวกอยู่แล้ว สามารถเรียกใช้ได้เลย

การหาค่ารากที่สองที่เป็นบวกของจำนวนจริงบวก

จากสมการทางคณิตศาสตร์ เรามาทราบว่า ถ้า  $a * a = z$  โดยที่  $a, z > 0$  จะได้ว่า  $a$  จะเป็นค่ารากที่สองที่เป็นบวกของ  $z$  ดังนั้นเราจะใช้หลักการทาง Numerical ในการประมาณค่า  $a$  และทำการคำนวณเปรียบเทียบ  $a * a$  กับ  $z$  เปลี่ยนค่า  $a$  ให้  $a * a$  เข้าใกล้  $z$  จนกว่าจะได้ผลลัพธ์ตามต้องการ สามารถเขียน Flow chart ได้ดังรูป  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก คการติดต่อกับ COMPUTERอธิบายการทำงานในการติดต่อกับ COMPUTER

สามารถแบ่งการทำงานหลักออกเป็น 3 ส่วน คือ

ส่วนส่งข้อมูลลง RAMPACK เริ่มโดยการเปิดพัลส์ที่ต้องการส่ง ส่ง CONTROL PORT 1 ที่ 8255 โดยให้ทุกพอร์ทเป็นเข้าที่พอร์ทพอร์ท อ่านค่าจากพัลส์ที่ละตัวอักษร ถ้าพบว่าเป็นหมายเหตุหรือช่องว่างจะข้ามไป ถ้าเป็น ADDRESS หรือ DATA จะเก็บเข้าตัวแปรในรูปแบบสิบหก ทว่าการส่ง ADDRESS HIGH BYTE ไปที่พอร์ท C, ส่ง ADDRESS LOW BYTE ไปที่ 373, ส่ง DATA HIGH BYTE และ LOW BYTE ไปที่พอร์ท A และ B ถ้าอ่าน DATA ได้ติดต่อกัน จะทำการเพิ่ม ADDRESS ขึ้นหนึ่งหน่วย อ่าน DATA นั้นและส่งเช่นเดิมอีกครั้ง ส่งจนจบพัลส์ แล้วทำการปิดพัลส์

ส่วนเปรียบเทียบเพื่อดูว่า ข้อมูลจากพัลส์และ RAM เพียงใดหรือไม่มีที่ตำแหน่งใด โปรแกรมนี้ต่างจากส่วนส่งที่ 8255 CONTROL PORT 1 ที่พอร์ท A และ B เป็นอินพุทพอร์ท พอร์ท C เป็นเข้าที่พอร์ทพอร์ท อ่าน DATA ของทั้งคู่มาเทียบกัน แสดง ADDRESS ที่เพียงใดบนจอ

ส่วนอ่านข้อมูลจากพัลส์ ที่อ่านข้อมูลที่คำนวณได้จาก RAM โดยใส่ ADDRESS เริ่มต้น และจบเป็นอินพุท ส่ง CONTROL PORT 1 ที่พอร์ท A และ B เป็นอินพุทพอร์ท พอร์ท C เป็นเข้าที่พอร์ทพอร์ท วนส่ง ADDRESS และอ่านข้อมูลเข้ามาแสดงที่หน้าจอ โดยเพิ่มค่า ADDRESS ขึ้นทุกครั้งจนถึง ADDRESS ท้าย

## บรรณานุกรม

1. TEXAS INSTRUMENTS, "First-Generation TMS32010 User's Guide",1988
2. คู่มือ IBM Technical Reference XT
3. Eugene L. Zuch Editor, "Data Acquisition and Conversion Handbook"
4. David J. Defatta , Joseph G. Lucas , William S. Hodgkiss ,  
"Digital signal processing : A System Design Approach" , JOHN  
WILEY & SONS , 1988
5. Louis Nashelsky Editor , "Introduction to Digital Technology"
6. ชนภัทร นันทจิวารัชย์ , สุชาติ ฝอลปกรณ์ , " ปรญญนิพนธ์ปีการศึกษา 2534  
ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สจล. เรื่องการประมวลผลสัญญาณ  
ดิจิทัล และการประยุกต์ใช้งาน "

