

การศึกษาบอร์ดประมวลผลสัญญาณดิจิทัลและการประยุกต์ใช้งาน  
DIGITAL SIGNAL PROCESSING (DSP) BOARD STUDY  
AND ITS APPLICATION



โดย  
น.ส. จิตติมา ตั้งเลิศปัญญา  
น.ส. ชนัดดา พงษ์พัฒน์

เลขที่.....  
เลขทะเบียน..... 42502  
วัน, เดือน, ปี 24 พ.ค. 2545

.b.....
.i.....

ปฏิญญานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิศวกรรมระบบควบคุม  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านกำไร  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาบอร์ดประมวลผลสัญญาณดิจิทัลและการประยุกต์ใช้งาน  
DIGITAL SIGNAL PROCESSING (DSP) BOARD STUDY  
AND ITS APPLICATION



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิศวกรรมระบบควบคุม  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2543

ภาควิชา วิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การศึกษาบอร์ดประมวลผลสัญญาณดิจิทัลและการประยุกต์ใช้งาน

(Digital Signal Processing : DSP)

ผู้จัดทำ

1. น.ส. จิตติมา ตั้งเลิศปัญญา
2. น.ส. ชนัดดา พงษ์พัฒน์

  
..... อาจารย์ที่ปรึกษา  
(อาจารย์สว่าง เลิศศิริสุนทร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาบอร์ดประมวลผลสัญญาณดิจิทัลและการประยุกต์ใช้งาน  
DIGITAL SIGNAL PROCESSING (DSP) BOARD STUDY  
AND ITS APPLICATION

โดย

น.ส. จิตติมา ตั้งเลิศปัญญา

น.ส. ชนัดดา พงษ์พัฒน์

อาจารย์ที่ปรึกษา

อาจารย์สว่าง

เลิศศิริสุนทร

ปีการศึกษา 2543

**บทคัดย่อ**

วิทยานิพนธ์ฉบับนี้ เป็นการศึกษาโครงสร้างสถาปัตยกรรมภายใน และการใช้งานของดีเอสพี (Digital Signal Processing : DSP) บนบอร์ดดีเอสเค (DSP Starter Kit : DSK) โดยจะเน้นในการเชื่อมต่อสัญญาณอนาล็อกผ่านพอร์ตอนุกรมของดีเอสพี และวงจรเชื่อมต่อสัญญาณอนาล็อก (Analog Interface circuit : AIC) การใช้งานของบอร์ดจะนำมาประยุกต์เป็นตัวควบคุมแบบพีไอดี (PID) ในการควบคุมความเร็วของมอเตอร์กระแสตรงให้มีความเร็วคงที่ ถึงแม้ว่า โหลดจะเปลี่ยนแปลง

**Abstract**

This thesis is the study of the architecture of DSP and its application on the DSK (DSP Starter Kit) board. The study emphasizes the analog interface through serial port of DSP and Analog interface circuit (AIC). The DSP board application is using it as a PID controller in order to control the constant speed of DC motor though its loads are varied.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อ	I
สารบัญ	II
สารบัญรูปภาพ	IV
สารบัญตาราง	VI
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	3
2.1 สถาปัตยกรรม TMS320C31	3
2.1.1 หน่วยประมวลผลกลาง	3
2.1.2 การจัดหน่วยความจำของ TMS320C31	6
2.1.3 การจัดการบัสภายใน	7
2.1.4 อุปกรณ์รอบนอก	7
2.2 โครงสร้างของ TMS320C31 ดีเอสเค	18
2.2.1 โครงสร้างของดีเอสเค	18
2.2.2 ภาพรวมของดีเอสเค	18
2.3 วงจรเชื่อมต่อสัญญาณอนาล็อก	20
2.4 มอเตอร์กระแสตรง	24
2.5 ตัวควบคุมแบบพีไอดี	27
บทที่ 3 การออกแบบการทดลอง	30
3.1 การทดลองการเชื่อมต่อสัญญาณอนาล็อก	30
3.2 การออกแบบวงจรในการควบคุมมอเตอร์แบบป้อนกลับ	32
บทที่ 4 ผลการทดลอง	35
4.1 การทดลองการใช่วงจรเชื่อมต่อสัญญาณอนาล็อก	35
4.1.1 การทดลองการรับค่าเข้าบอร์ดดีเอสเค และส่งค่าเดิม ออกจากบอร์ดดีเอสเค	35
4.1.2 การทดลองผลของความถี่สัญญาณอินพุตที่มีผลต่อ การล้าหลังของสัญญาณ	39
4.2 การทดลองหาค่าพารามิเตอร์ของมอเตอร์กระแสตรง	42
4.2.1 การทดลองวัดค่า $R_s$	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2	การทดลองการวัดค่า $L_u$	43
4.2.3	การคำนวณหาค่า $K\Phi$	44
4.2.4	การทดลองการวัดค่า $\tau_m$	44
4.3	การทดลองเขียนโปรแกรมควบคุมแบบพี ไอ และดี	45
4.3.1	การเขียน โปรแกรมควบคุมแบบพี	45
4.3.2	การเขียน โปรแกรมควบคุมแบบไอ	47
4.3.3	การเขียน โปรแกรมควบคุมแบบดี	48
4.3.4	การเขียน โปรแกรมควบคุมแบบพี ไอดี	50
4.4	ผลการทดลองในการควบคุมความเร็วมอเตอร์	52
4.4.1	ผลการทดลองในการควบคุมความเร็วมอเตอร์ด้วย ตัวควบคุมแบบพี	52
4.4.2	ผลการทดลองในการควบคุมความเร็วมอเตอร์ด้วย ตัวควบคุมแบบพี ไอ	52
4.4.3	ผลการทดลองในการควบคุมความเร็วมอเตอร์ด้วย ตัวควบคุมแบบพี ไอ	53
4.4.4	ผลการทดลองในการควบคุมความเร็วมอเตอร์ด้วย ตัวควบคุมแบบพีดี	54
4.4.5	ผลการทดลองในการควบคุมความเร็วมอเตอร์ด้วย ตัวควบคุมแบบพี ไอดี	54
บทที่ 5	บทวิจารณ์และสรุป	56
	ภาคผนวก	
	กิตติกรรมประกาศ	
	หนังสืออ้างอิง	

## สารบัญรูปภาพ

	หน้า
รูปที่ 1-1 แสดงการเชื่อมต่อสัญญาณอนาล็อก	2
รูปที่ 1-2 แสดงการควบคุมมอเตอร์กระแสตรงวงปิด	2
รูปที่ 2-1 แสดงบล็อกไดอะแกรมของ TMS320C31	4
รูปที่ 2-2 แสดงหน่วยประมวลผลกลาง	5
รูปที่ 2-3 แสดงการจัดการหน่วยความจำ	6
รูปที่ 2-4 แสดงอุปกรณ์รอบนอกกับบัส และสัญญาณที่เกี่ยวข้อง	7
รูปที่ 2-5 แสดงตำแหน่งของรีจิสเตอร์ตัวจับเวลา	8
รูปที่ 2-6 แสดงสัญญาณนาฬิกาของสัญญาณพัลซ์ และสัญญาณนาฬิกา	9
รูปที่ 2-7 แสดงตำแหน่งของรีจิสเตอร์ของพอร์ตอนุกรม	17
รูปที่ 2-8 แสดงส่วนประกอบพื้นฐานของซีเอสเค	19
รูปที่ 2-9 แสดงบล็อกไดอะแกรมของวงจรเชื่อมต่อสัญญาณอนาล็อก	20
รูปที่ 2-10 แสดงรูปแบบของบิตในการส่งข้อมูลในการส่งของ DR, DX	22
รูปที่ 2-11 แสดงบิตควบคุม D0,D1 ของการส่งข้อมูลในแต่ละแบบ	22
รูปที่ 2-12 แสดงบิตควบคุมการขยาย	24
รูปที่ 2-14 แสดงบล็อกไดอะแกรมของมอเตอร์	26
รูปที่ 2-15 แสดงระบบวงปิดที่ใช้ตัวควบคุม	27
รูปที่ 2-16 แสดงบล็อกไดอะแกรมของพีไอดี	28
รูปที่ 3-1 แสดงการเชื่อมต่อระหว่างอุปกรณ์เชื่อมต่อสัญญาณอนาล็อก และ TLC320C31	31
รูปที่ 3-2 แสดงชุดวงจรขับมอเตอร์และชุดมอเตอร์/เจนเนอเรเตอร์	32
รูปที่ 3-3 แสดงชุดทดลองของแทคโคเจนเนอเรเตอร์	33
รูปที่ 3-4 แสดงชุดโหลด	33
รูปที่ 3-5 แสดงวงจรในการป้อนกลับและวงจรขยายเพื่อการหาค่าผิดพลาด	34
รูปที่ 3-6 แสดงวงจรขยายสัญญาณออกจากบอร์ด	34
รูปที่ 4-1 แสดงผังการทำงานของโปรแกรมรับค่าและส่งค่าเดิมออก	35
รูปที่ 4-2 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 300 Hz	39
รูปที่ 4-3 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 600 Hz	39
รูปที่ 4-4 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 900 Hz	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4-5 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 1200 Hz	40
รูปที่ 4-6 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 1500 Hz	41
รูปที่ 4-7 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 1200Hz	41
รูปที่ 4-8 แสดงบล็อกการทดลองมอเตอร์	42
รูปที่ 4-9 แสดงความสัมพันธ์ระหว่าง $V_a, I_a$ ในการหาค่า $\tau_c$	43
รูปที่ 4-10แสดงความสัมพันธ์ระหว่าง $V_a, I_a$ ในการหาค่า $\tau_m$	44
รูปที่ 4-11แสดงบล็อก ไดอะแกรมในการทดลองเขียนตัวควบคุมแบบพี ไอ และดี	45
รูปที่ 4-12แสดงฟังก์ชันในการเขียน โปรแกรมตัวควบคุมแบบพี	45
รูปที่ 4-13แสดงผลของตัวควบคุมแบบพีจากการทดลองจริง เมื่อให้ค่า $K_p$ เท่ากับ 2	46
รูปที่ 4-14แสดงผลของตัวควบคุมแบบพีจาก โปรแกรมMATLAB เมื่อค่า $K_p$ เท่ากับ 2	46
รูปที่ 4-15แสดงฟังก์ชันในการเขียน โปรแกรมตัวควบคุมแบบไอ	47
รูปที่ 4-16แสดงผลของตัวควบคุมแบบ ไอจากการทดลองจริง เมื่อให้ค่า $K_i$ เท่ากับ 1000	47
รูปที่ 4-17แสดงผลของตัวควบคุมแบบ ไอจากโปรแกรม MATLAB เมื่อให้ค่า $K_i$ เท่ากับ 1000	48
รูปที่ 4-18แสดงฟังก์ชันในการเขียน โปรแกรมตัวควบคุมแบบดี	48
รูปที่ 4-19 แสดงผลของตัวควบคุมแบบดีจาก โปรแกรม MATLAB เมื่อให้ค่า $K_d$ เท่ากับ 0.0125	49
รูปที่ 4-20แสดงผลของตัวควบคุมแบบดีจาก โปรแกรม MATLAB เมื่อให้ $K_d$ เท่ากับ 0.125	49
รูปที่ 4-21แสดงฟังก์ชันในการเขียน โปรแกรมตัวควบคุมแบบพีไอดี	50
รูปที่ 4-22แสดงผลการทดลองการเขียนโปรแกรมแบบพีไอดี เมื่อใช้ค่า $K_p=1$ $K_i=10$ $K_d=0.0125$	51
รูปที่ 4-23แสดงผลของตัวควบคุมแบบพีไอดี จากโปรแกรมMATLAB เมื่อใช้ค่า $K_p=1$ $K_i=10$ $K_d=0.0125$	51
รูปที่ 4-24แสดงผลการควบคุมความเร็วมอเตอร์ด้วยตัวควบคุมแบบพี	52
รูปที่ 4-25แสดงผลการควบคุมความเร็วมอเตอร์ด้วยตัวควบคุมแบบ ไอ	52
รูปที่ 4-26แสดงผลการควบคุมความเร็วมอเตอร์ด้วยตัวควบคุมแบบพีไอ	53
รูปที่ 4-27แสดงผลการควบคุมความเร็วมอเตอร์ด้วยตัวควบคุมแบบพีดี	54
รูปที่ 4-28แสดงผลการควบคุมความเร็วมอเตอร์ด้วยตัวควบคุมแบบพีไอดี	55

## สารบัญตาราง

	หน้า
ตารางที่ 2-1 แสดงรีจิสเตอร์ของหน่วยประมวลผลกลาง	5
ตารางที่ 2-2 แสดงบิตต่าง ๆ ของรีจิสเตอร์ควบคุมส่วนกลาง	10
ตารางที่ 2-3 แสดงบิตต่าง ๆ ของรีจิสเตอร์ควบคุมฝ่ายส่งของพอร์ตอนุกรม	13
ตารางที่ 2-4 แสดงบิตต่าง ๆ ของรีจิสเตอร์ควบคุมฝ่ายรับของพอร์ตอนุกรม	14
ตารางที่ 2-5 แสดงบิตต่าง ๆ ของรีจิสเตอร์ควบคุมตัวจับเวลาของฝ่ายรับ และ ส่งพอร์ตอนุกรม	15
ตารางที่ 2-6 แสดงรูปแบบการส่งแบบ secondary	23
ตารางที่ 2-7 แสดงผลตอบสนองต่อตัวควบคุมแบบพี ไอ และดี	28
ตารางที่ 4-1 แสดงผลการหาค่า R	42
ตารางที่ 4-2 แสดงความเร็วของมอเตอร์จากการควบคุมแบบต่างๆ เมื่อมีการเปิดสวิตช์โหลด	55

## บทที่ 1

### บทนำ

ดีเอสพี (Digital Signal Processing : DSP) มีลักษณะเป็นไมโครโปรเซสเซอร์ (Microprocessor) ที่มีความเร็วสูง ที่มีชุดคำสั่งพิเศษ และมีสถาปัตยกรรมภายในที่เหมาะสมสำหรับการประมวลสัญญาณ ซึ่งจะเป็นการคำนวณทางตัวเลข ดีเอสพีนี้มีการประยุกต์ใช้งานอย่างแพร่หลาย ดังนี้

- ดิจิตอลพีวีเตอร์
- การประมวลผลทางรูปภาพ (image processing)
- การประมวลผลทางเสียงพูด วงจรจดจำเสียงพูด
- งานควบคุม เช่น การควบคุมเซอร์โว การควบคุมหุ่นยนต์
- โมเด็ม แฟกซ์ เทเลคอนเฟอเรนซ์ (teleconference)
- วงจรสังเคราะห์เสียง
- ฯลฯ

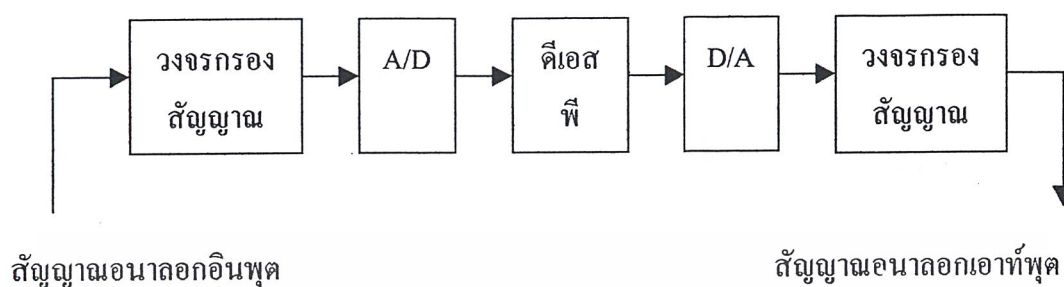
การที่ดีเอสพีเป็นที่นิยมอย่างแพร่หลายนั้น เป็นเพราะเป็นไมโครโปรเซสเซอร์ที่มีประโยชน์ใช้สอยคุ้มค่า เมื่อเทียบกับราคาที่ไม่แพง มีการสนับสนุนทั้งทางฮาร์ดแวร์ (Hardware) และทางซอฟต์แวร์ (Software) ซึ่งมีการพัฒนาอย่างต่อเนื่อง นอกจากนี้ การประมวลผลทางดิจิตอลจะสามารถนำมาประยุกต์ใช้งานได้หลากหลาย เนื่องจากสามารถที่จะโปรแกรมใหม่ได้เมื่อถูกนำไปใช้กับงานที่แตกต่างกันไป ซึ่งต่างจากการประมวลผลทางอนาล็อกซึ่งจะต้องมีอุปกรณ์ช่วยเป็นอย่างมาก เช่น การใช้ตัวต้านทาน ตัวเก็บประจุ ออปแอม และอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ทำให้มีความยืดหยุ่นต่ำ

ในโครงการนี้จะใช้บอร์ดดีเอสพี (DSP Starter Kit) ซึ่งจะประกอบด้วยไมโครโปรเซสเซอร์ดีเอสพี TMS320C31-50 ซึ่งเป็นรุ่น TMS320C3x ของบริษัท TEXAS INSTRUMENT เป็น DSP รุ่นอินตรินซิกขนาด 32 บิตแบบ CMOS สมรรถภาพสูงอยู่ในตระกูลของ TMS320 สามารถทำงานด้วยความเร็วสูงถึง 40 นาโนวินาทีต่อไซเคิล (ns /cycle) ประมวลผลได้ด้วยความเร็วถึง 60 ล้านคำสั่งอินตรินซิกต่อวินาที และ 30 ล้านคำสั่งต่อวินาที (million instructions per second :MIPS)

การประมวลสัญญาณอนาล็อก จะต้องทำโดยผ่านตัวเชื่อมต่อสัญญาณอนาล็อก (Analog interface circuit : AIC) รุ่น TLC32040 ซึ่งจะมีอยู่ในบอร์ดดีเอสพี และส่งไปประมวลผลที่

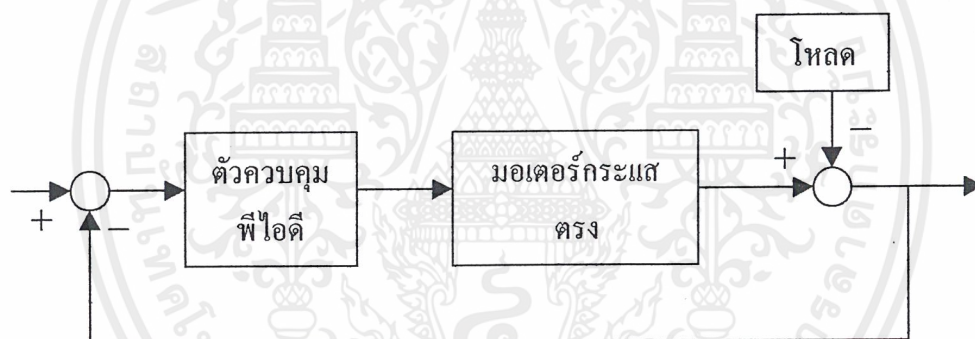
เอกสาร TMS320C31 โดยผ่านพอร์ตคอนโทรลเลอร์ของ C31 นี้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อสัญญาณอนาลอก จะเป็นดังรูปต่อไปนี้



รูปที่ 1-1 แสดงการเชื่อมต่อสัญญาณอนาลอก

โครงการนี้ใช้บอร์ดดีเอสเค ในการควบคุมความเร็วของมอเตอร์ให้คงที่ ถึงแม้ว่าโหลดจะเปลี่ยนแปลง ซึ่งจะใช้บอร์ดดีเอสเคเป็นตัวควบคุมแบบพีไอดี โดยจะเป็นการควบคุมวงปิดดังนี้



รูปที่ 1-2 แสดงการควบคุมมอเตอร์กระแสตรงวงปิด

มอเตอร์กระแสตรงต่อติดกับแทคโคเจนเนอเรเตอร์ (Tacho-generator) ซึ่งจะมีเอาต์พุตเป็นแรงดันที่แปรผันตามความเร็วในการหมุนของมอเตอร์ เมื่อโหลดซึ่งในโครงการนี้คือความต้านทานเปลี่ยนแปลง จะทำให้ความเร็วของมอเตอร์เปลี่ยนแปลง ซึ่งสามารถแก้ไขปัญหานี้ได้โดยใช้ตัวควบคุมแบบพีไอดี

## บทที่ 2

### ทฤษฎี

#### 2.1 สถาปัตยกรรม TMS320C3x

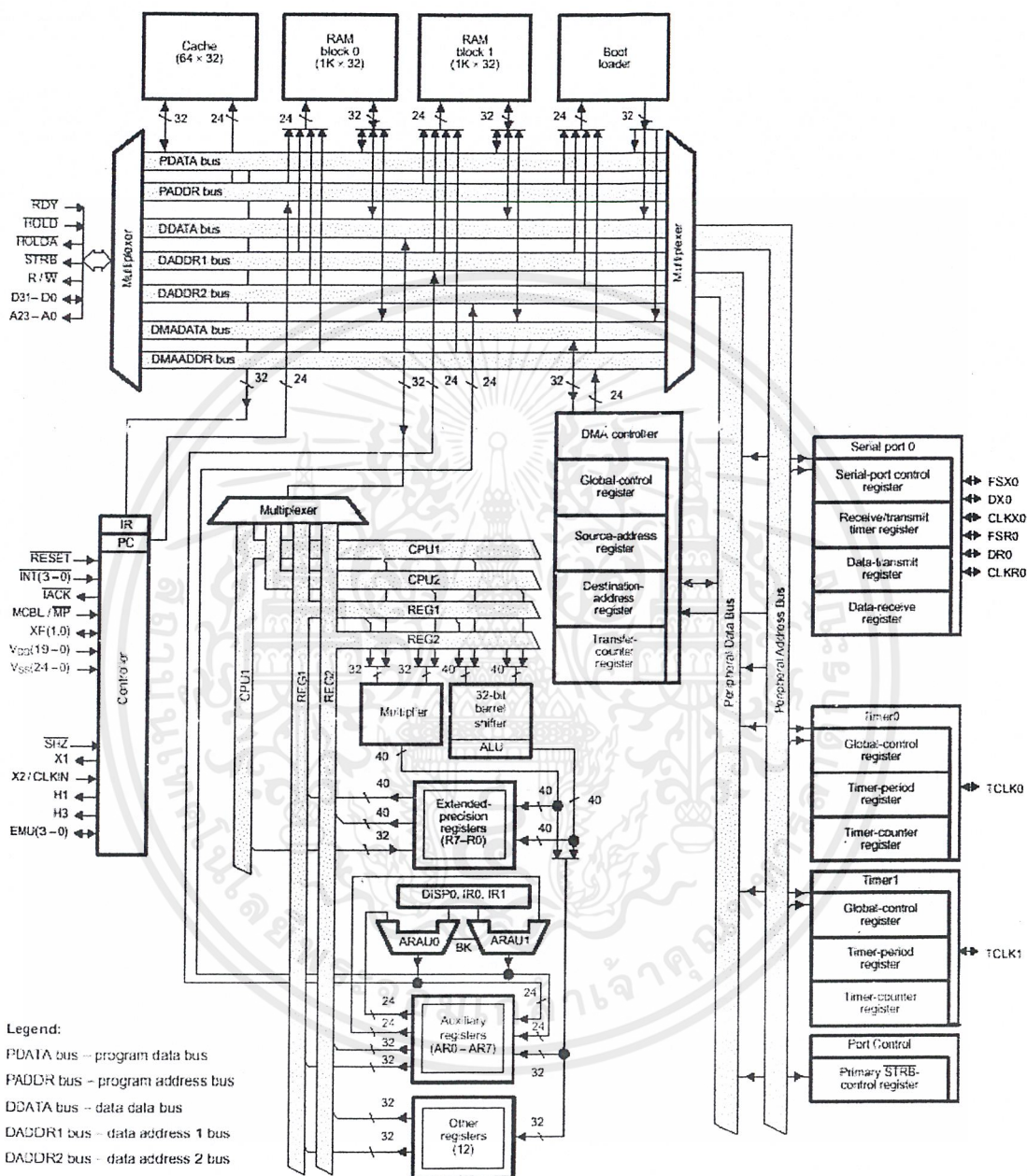
สถาปัตยกรรม TMS320C3x สามารถตอบสนองความต้องการพื้นฐานของขั้นตอนวิธีคำนวณที่ซับซ้อน และเน้นการแก้ปัญหาทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ หน่วยอิงครรชนีมีพิสัยแบบพลวัต (dynamic range) ที่กว้างและเที่ยงตรง, หน่วยความจำบนชิปขนาดใหญ่, ระบบแบบขนานคิกรูสูง, และตัวควบคุมการเข้าถึงหน่วยความจำโดยตรง ทำให้ TMS320C3x มีสมรรถภาพสูง รูปที่ 2-1 เป็นบล็อกไดอะแกรมของสถาปัตยกรรม TMS320C31

##### 2.1.1 หน่วยประมวลผลกลาง (Central Processing Unit : CPU)

ประกอบด้วยอุปกรณ์ต่างๆ ดังนี้

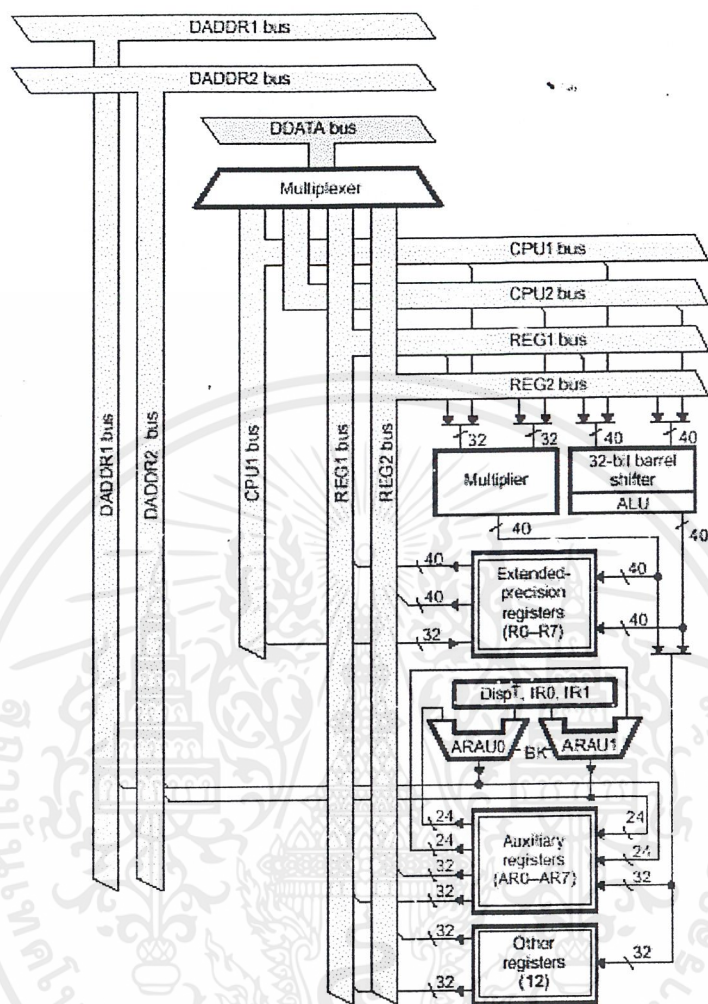
- ตัวคูณเลขอิงครรชนี/จำนวนเต็ม (Floating-point/integer multiplier) ตัวคูณจะทำการคูณแบบรอบเดียว โดยเป็นค่าจำนวนเต็ม 24 บิต และค่าอิงครรชนี 32 บิต
- หน่วยคำนวณและตรรกะ (Arithmetic logic unit : ALU) จัดการข้อมูลจำนวนเต็ม 32 บิต, ตรรกะ 32 บิต และอิงครรชนี 40 บิต แบบรอบเดียว สามารถคูณแบบขนาน และบวก/ลบเลขจำนวนเต็มหรือเลขอิงครรชนี 4 ตัวได้ในรอบเดียว
- ตัวเลื่อนหรือหมุนเวิร์ดข้อมูล (barrel shifter) 32 บิต
- บัสภายใน (CPU1/CPU2 และ REG1/REG2)
- หน่วยคำนวณรีจิสเตอร์ช่วย (Auxiliary register arithmetic units : ARAUs) ARAU0 และ ARAU1 สามารถสร้าง 2 ตำแหน่งในรอบเดียว หน่วยคำนวณรีจิสเตอร์ช่วย ทำงานแบบขนานด้วยตัวคูณและALU
- แฟ้มรีจิสเตอร์ของหน่วยประมวลผลกลาง (CPU register file) ชื่อของรีจิสเตอร์และฟังก์ชันที่กำหนด จะอยู่ในตาราง 2-1 ตามตารางจะบรรยายฟังก์ชันของแต่ละรีจิสเตอร์หรือกลุ่มของรีจิสเตอร์ไว้อย่างย่อๆ

รูปที่ 2-2 แสดงอุปกรณ์ต่างๆ ของหน่วยประมวลผลกลาง โดยจะตามหลังด้วยคำอธิบาย



รูปที่ 2-1 บล็อกไดอะแกรมของ TMS320C31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-2 หน่วยประมวลผลกลาง

ตาราง 2-1 แสดงรีจิสเตอร์ของหน่วยประมวลผลกลาง

Register name	Assigned Function
R0 – R7	Extended – precision register 0 – 7
AR0 – AR7	Auxiliary register 0 – 7
DP	Data – page pointer
IR0	Index register 0
IR1	Index register 1
BK	Block size
SP	System stack pointer

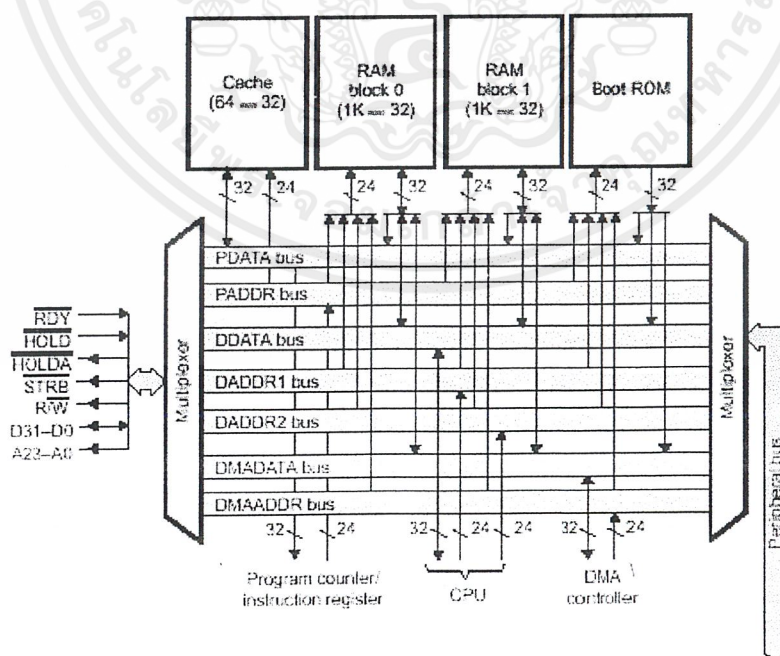
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ST	Status register
IE	CPU / DMA Interrupt enable
IF	CPU interrupt flags
IOF	I/O flags
RS	Repeat start address
RE	Repeat end address
RC	Repeat counter

### 2.1.2 การจัดหน่วยความจำของ TMS320C31

TMS320C3x มีพื้นที่ของหน่วยความจำทั้งหมดเป็น 16M (million) 32 บิต เวิร์ด โดยบรรจุ โปรแกรม, ข้อมูล และเนื้อหาของอินพุต/เอาต์พุตอยู่ภายใน ดังนั้นสามารถเก็บตาราง (table), สัมประสิทธิ์ (coefficient), รหัสของโปรแกรม (program code) หรือข้อมูล ไว้ได้ทั้งใน RAM หรือ ROM ก็ได้ ในกรณีนี้จะทำให้สามารถใช้หน่วยความจำได้มากที่สุด และ จัดพื้นที่หน่วยความจำได้ตามต้องการ

ใน TMS320C31 มี RAM 2 ตัว แต่ละตัวมีขนาด 1K x 32 บิต, ไม่มี ROM และมี Cache ขนาด 64 x 32 บิต เพื่อเก็บคำสั่งที่ใช้บ่อยๆ (ทำให้ทำงานได้ดีขึ้น)



รูปที่ 2-3 การจัดการหน่วยความจำ

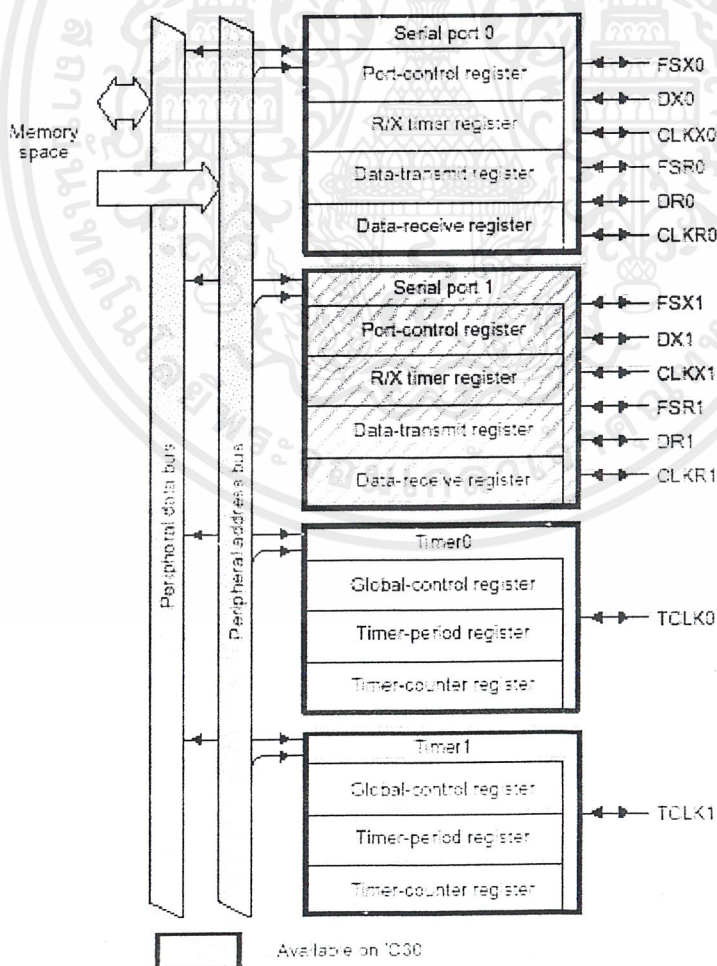
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3 การจัดการบัสภายใน

ส่วนใหญ่ TMS3320C3x จะมีสมรรถภาพสูงเพราะบัสภายใน และการแยกแบบขนานของ บัสของโปรแกรม (PADDR และ PDATA), บัสของข้อมูล (DADDR1, DADDR2 และ DDATA) และบัสของการเข้าถึงหน่วยความจำโดยตรง (DMAADDR และ DMADATA) จะสามารถไปนำ โปรแกรมมาแบบขนาน, การเข้าถึงข้อมูล และ การเข้าถึงหน่วยความจำโดยตรง โดยบัสเหล่านี้จะ ต่อกับพื้นที่ทั้งหมด (หน่วยความจำบนชิป, หน่วยความจำนอกชิป และ อุปกรณ์ภายนอกบนชิป) รูป ที่ 2-3 แสดงบัสภายในเหล่านี้ และการติดต่อกับบัสของหน่วยความจำบนชิปและนอกชิป

### 2.1.4 อุปกรณ์รอบนอก (Peripherals)

อุปกรณ์รอบนอกของ TMS320C31 ประกอบด้วยตัวจับเวลา (timer) 2 ตัว และพอร์ตอนุกรม (serial port) 1 ตัว รูปที่ 2-4 แสดงอุปกรณ์รอบนอกกับบัสและสัญญาณที่เกี่ยวข้อง



รูปที่ 2-4 แสดงอุปกรณ์รอบนอกกับบัสและสัญญาณที่เกี่ยวข้อง  
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 2.1.4.1 ตัวจับเวลา (Timer)

ตัวจับเวลา 2 ตัวของ TMS320C3x เป็นตัวจับเวลาอเนกประสงค์ (general-purpose timer) / ตัวนับเหตุการณ์ (event-counter) 32 บิตที่มีโหมดด้านสัญญาณ 2 ตัว และสัญญาณนาฬิกา (clocking) ภายในหรือภายนอก สามารถใช้ตัวจับเวลา เพื่อส่งสัญญาณให้ TMS320C3x หรือโลกภายนอกเป็นช่วงๆ ตามที่กำหนด หรือนับเหตุการณ์ภายนอก ตัวจับเวลาแต่ละตัวจะมีขาอินพุต/เอาต์พุต (I/O pin) ดังรูปที่ 2-4 ซึ่งสามารถเลือกใช้เป็นสัญญาณนาฬิกาอินพุต (input clock) ให้ตัวจับเวลา, สัญญาณนาฬิกาเอาต์พุต (output clock) หรือขาอินพุต/เอาต์พุตอเนกประสงค์ (general-purpose I/O pin)

ในแต่ละตัวจับเวลา จะมีรีจิสเตอร์การแทนข้อมูล 3 ชนิด ดังนี้

- รีจิสเตอร์ควบคุมส่วนกลาง (Global-control register) กำหนดโหมดการทำงาน (operating mode) ของตัวจับเวลา, ตรวจสอบสถานะของตัวจับเวลา และ ควบคุมฟังก์ชันต่างๆ ของขาอินพุต/เอาต์พุตของตัวจับเวลา
- รีจิสเตอร์คาบเวลา (Period register) ระบุความถี่ทางสัญญาณของตัวจับเวลา
- รีจิสเตอร์ตัวนับ (Counter register) เก็บค่าในขณะปัจจุบันของตัวนับที่เพิ่มค่า (incrementing counter) ซึ่งสามารถเพิ่มค่าตัวจับเวลา โดยการนับขอบขาขึ้นหรือขอบขาลงของตัวนับสัญญาณนาฬิกาอินพุต (input clock counter) ตัวจับเวลาทำให้เกิดการอินเทอร์รัพต์ภายในได้เมื่อค่าของรีจิสเตอร์ตัวนับมีค่าเท่ากับรีจิสเตอร์คาบเวลา โดยตัวสร้างสัญญาณพัลส์ (pulse generator) จะสร้างสัญญาณนาฬิกาภายนอก 2 ตัวคือ สัญญาณพัลส์ (pulse mode) หรือ สัญญาณนาฬิกา (clock mode)

ตำแหน่งของรีจิสเตอร์ของตัวจับเวลา แสดงในรูปที่ 2-5

Register	Peripheral address	
	Timer 0	Timer 1
Timer Global Control	808020h	808030h
Timer Counter	808024h	808034h
Timer Period	808028h	808038h

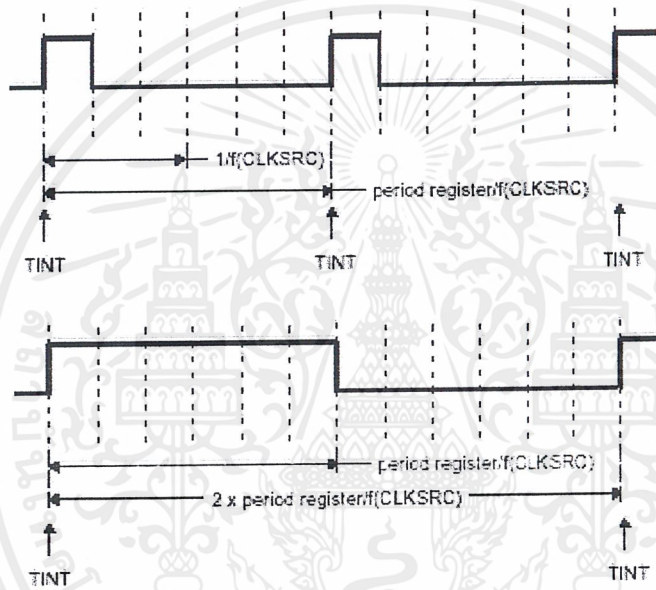
รูปที่ 2-5 แสดงตำแหน่งของรีจิสเตอร์ของตัวจับเวลา

อัตราของการส่งสัญญาณของตัวจับเวลากำหนดโดยความถี่ของสัญญาณนาฬิกาอินพุตของตัวจับเวลาและรีจิสเตอร์คาบเวลาของตัวจับเวลา ดังสมการข้างล่างโดยมีผลทั้งสัญญาณนาฬิกาของตัวจับเวลาภายในหรือภายนอกก็ได้

$$f(\text{pulse mode}) = f(\text{timer clock}) / \text{period register}$$

$$f(\text{clock mode}) = f(\text{timer clock}) / (2 \times \text{period register})$$

รูปที่ 2-6 แสดงสัญญาณนาฬิกาของสัญญาณพัลส์ และสัญญาณนาฬิกา ตามลำดับ



รูปที่ 2-6 แสดงสัญญาณนาฬิกาของสัญญาณพัลส์ และสัญญาณนาฬิกา

#### 2.1.4.2 พอร์ตอนุกรม (Serial Port)

ใน TMS320C31 มีพอร์ตอนุกรม 1 ตัว สามารถส่งผ่านข้อมูลได้ทั้ง 8,16,24 หรือ 32 บิตต่อเวิร์ด เป็นการส่งแบบสองทิศทาง สัญญาณนาฬิกาสำหรับพอร์ตอนุกรมสามารถสร้างจากภายในคือ serial port timer และ period register หรือสามารถสร้างจากภายนอกก็ได้ สัญญาณนาฬิกาภายในจะได้อัตราจากการหารความถี่ของ  $f(\text{H1})$  โหมดการส่งผ่านแบบต่อเนื่องสามารถทำได้โดยพอร์ตอนุกรมอนุญาตให้มีการส่งและรับเวิร์ด โดยไม่ต้องทำการซิงโครไนซ์ใหม่

ในแต่ละตัวของพอร์ตอนุกรมมีรีจิสเตอร์แทนที่ข้อมูล ดังนี้

##### 2.1.4.2.1 รีจิสเตอร์ควบคุมส่วนกลาง (Global Control Register) ใช้ในการควบคุมฟังก์ชัน

การทำงานของพอร์ตอนุกรม เป็นรีจิสเตอร์ขนาด 32 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2-2 แสดงบิตต่าง ๆ ของรีจิสเตอร์ควบคุมส่วนกลางของพอร์ตอนุกรม

บิต	ชื่อ	การทำงาน
0	RRDY	ถ้า RRDY=1 แสดงว่าบัฟเฟอร์ฝ่ายรับมีข้อมูลใหม่และพร้อมที่จะอ่าน โดยจะไหลผ่านเข้าไปในรีจิสเตอร์รับข้อมูล (Data Receive Register) ขอบขาขึ้นของสัญญาณจะไปเซ็ท RINT ถ้า RRDY=0 แสดงว่าบัฟเฟอร์ฝ่ายรับไม่มีข้อมูลใหม่เข้ามา จะเป็น 0 หลังจากทีบัฟเฟอร์ถูกอ่านแล้ว
1	XRDY	ถ้า XRDY=1 แสดงว่าบัฟเฟอร์ฝ่ายส่งได้เขียนข้อมูลบิตสุดท้ายลงในรีจิสเตอร์เลื่อนแล้ว และพร้อมที่จะเขียนข้อมูลใหม่ ขอบขาขึ้นจะเซ็ท XINT ถ้า XRDY=0 แสดงว่าบัฟเฟอร์ฝ่ายส่งยังไม่ได้เขียนข้อมูลบิตสุดท้ายลงในรีจิสเตอร์เลื่อน และยังไม่พร้อมสำหรับเวิร์คใหม่
2	FSXOUT	ถ้า RSXOUT=0 แล้วขา RSX เป็นขาอินพุต ถ้า RSXOUT=1 แล้วขา RSX เป็นขาเอาต์พุต
3	XSREMPY	ถ้า XSREMPY=0 แล้วรีจิสเตอร์เลื่อนของฝ่ายส่งยังว่างอยู่ ถ้า XSREMPY=1 แล้วรีจิสเตอร์เลื่อนของฝ่ายส่งไม่
4	RSRFULL	ถ้า RSRFULL=1 อาจทำให้ฝ่ายรับเกิดโอเวอร์รัน ในโหมดการทำงานต่อเนื่อง RSRFULL=1 เมื่อทั้งรีจิสเตอร์เลื่อนของฝ่ายรับและรีจิสเตอร์ข้อมูลของฝ่ายรับเต็ม ในโหมดการทำงานไม่ต่อเนื่อง RSRFULL=1 เมื่อทั้งรีจิสเตอร์เลื่อนของฝ่ายรับและรีจิสเตอร์ข้อมูลของฝ่ายรับเต็ม และขา FSR ได้รับข้อมูลใหม่ เมื่อข้อมูลถูกอ่านบิตนี้จะเซ็ทเป็น 0 ดังนั้นบิตนี้จะถูกรีเซ็ทโดยระบบเท่านั้น
5	HS	ถ้า HS=1 โหมดแฮนด์เซ็ทจะทำงาน ถ้า HS=0 โหมดแฮนด์เซ็ทจะไม่ทำงาน
6	XCLKSRCE	ถ้า XCLKSRCE=1 แล้วสัญญาณนาฬิกาฝ่ายส่งภายในถูกใช้ ถ้า XCLKSRCE=0 แล้วสัญญาณนาฬิกาฝ่ายส่งภายนอกถูกใช้
7	RCLKSRCE	ถ้า RCLKSRCE=1 แล้วสัญญาณนาฬิกาฝ่ายรับภายในถูกใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต	ชื่อ	การทำงาน
8	XVAREN	ถ้า RCLKSRCE=1 แล้วสัญญาณนาฬิกาฝ่ายรับภายนอกถูกใช้ ถ้า XVAREN=0 แล้วอัตราการส่งข้อมูลจะคงที่ โดยขา FSX จะทำงานในช่วงสุดท้ายของรอบสัญญาณนาฬิกาฝ่ายส่ง และ ไม่ทำงานก่อนการส่งจะเริ่ม ถ้า XVAREN=1 แล้วอัตราการส่งข้อมูลจะแปรค่าได้ โดยขา FSX จะทำงานตลอดเวลาที่มีการส่งข้อมูล
9	RVAREN	ถ้า RVAREN=0 แล้วอัตราการรับข้อมูลจะคงที่ โดยขา FSR จะทำงานในช่วงสุดท้ายของรอบสัญญาณนาฬิกาฝ่ายรับ และ ไม่ทำงานก่อนการรับ จะเริ่ม ถ้า RVAREN=1 แล้วอัตราการรับข้อมูลจะแปรค่าได้ โดยขา FSR จะทำงานตลอดเวลาที่มีการรับข้อมูล
10	XFSM	ถ้า XFSM=0 การทำงานในโหมดต่อเนื่องโดยเวิร์ดแรกของบล็อกรจะสร้างชิโคนัสเพลลาส์ ถ้า XFSM=1 การทำงานในโหมดมาตรฐานโดยทุกๆเวิร์ดจะสร้างชิโคนัสเพลลาส์
11	RFSM	ถ้า RFSM=0 การทำงานในโหมดต่อเนื่องโดยเวิร์ดแรกของบล็อกรจะสร้างชิโคนัสเพลลาส์ ถ้า RFSM=1 การทำงานในโหมดมาตรฐานโดยทุกๆเวิร์ดจะสร้างชิโคนัสเพลลาส์
12	CLKXP	ถ้า CLKXP=0 แล้วสัญญาณนาฬิกาฝ่ายส่งจะแอ็คทีฟไฮท์ ถ้า CLKXP=1 แล้วสัญญาณนาฬิกาฝ่ายส่งจะแอ็คทีฟโลว์
13	CLKRP	ถ้า CLKRP=0 แล้วสัญญาณนาฬิกาฝ่ายส่งจะแอ็คทีฟไฮท์ ถ้า CLKRP=1 แล้วสัญญาณนาฬิกาฝ่ายส่งจะแอ็คทีฟโลว์
14	DXP	ถ้า DXP=0 แล้ว DX จะแอ็คทีฟไฮท์ ถ้า DXP=1 แล้ว DX จะแอ็คทีฟโลว์
15	DRP	ถ้า DRP=0 แล้ว DR จะแอ็คทีฟไฮท์ ถ้า DRP=1 แล้ว DR จะแอ็คทีฟโลว์
16	FSXP	ถ้า FSXP=0 แล้ว FSX จะแอ็คทีฟไฮท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต	ชื่อ	การทำงาน
17	FSRP	ถ้า FSXP=1 แล้ว FSX จะแอสคทีฟโลว์ ถ้า FSRP=0 แล้ว FSR จะแอสคทีฟไฮท์ ถ้า FSRP=1 แล้ว FSR จะแอสคทีฟโลว์
19-18	XLEN	เป็นตัวบอกความยาวของเวิร์ดที่จะส่ง 00=18 บิต                      01=16 บิต 10=24 บิต                      11=32 บิต
21-20	RLEN	เป็นตัวบอกความยาวของเวิร์ดที่จะรับ 00=18 บิต                      01=16 บิต 10=24 บิต                      11=32 บิต
22	XTINT	ถ้า XTINT=0 แล้วการอินเทอร์รัพต์ของตัวจับเวลาฝ่ายส่งไม่ทำงาน ถ้า XTINT=1 แล้วการอินเทอร์รัพต์ของตัวจับเวลาฝ่ายส่งจะทำงาน
23	XINT	ถ้า XINT=0 แล้วการอินเทอร์รัพต์ของฝ่ายส่งไม่ทำงาน
24	RTINT	ถ้า RTINT=1 แล้วการอินเทอร์รัพต์ของฝ่ายส่งจะทำงาน ถ้า RTINT=0 แล้วการอินเทอร์รัพต์ของตัวจับเวลาฝ่ายรับไม่ทำงาน ถ้า RTINT=1 แล้วการอินเทอร์รัพต์ของตัวจับเวลาฝ่ายรับจะทำงาน
25	RINT	ถ้า RINT=0 แล้วการอินเทอร์รัพต์ของฝ่ายรับไม่ทำงาน ถ้า RTINT=1 แล้วการอินเทอร์รัพต์ของฝ่ายรับส่งจะทำงาน
26	XRESET	ถ้า XRESET=0 แล้วฝ่ายส่งของพอร์ตอนุกรมจะรีเซ็ต ถ้าต้องการออกจากการรีเซ็ตให้บิตนี้เป็น 1
27	RRESET	ถ้า RRESET=0 แล้วฝ่ายรับของพอร์ตอนุกรมจะรีเซ็ต ถ้าต้องการออกจากการรีเซ็ตให้บิตนี้เป็น 1
31-28	RESERVE	เป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.1.4.2.2 รีจิสเตอร์ควบคุมฝ่ายส่ง (FSX/DX/CLKX Port control) เป็นรีจิสเตอร์ขนาด 32 บิต ควบคุมการทำงานของพอร์ตอนุกรมในส่วนของ FSX,DXและ CLKX**

ตาราง 2-3 แสดงบิตต่าง ๆ ของรีจิสเตอร์ควบคุมฝ่ายส่งของพอร์ตอนุกรม

บิต	ชื่อ	การทำงาน
0	CLKXFUNC	ควบคุมการทำงานของ CLKX ถ้า CLKXFUNC=0 แล้ว CLKX จะเป็นพอร์ตอินพุตเอาต์พุต อเนกประสงค์ ถ้า CLKXFUNC=1 แล้ว CLKXจะเป็นขาพอร์ตอนุกรม
1	CLKXI/O	ถ้า CLKXI/O=0 แล้ว CLKX เป็นพอร์ตอินพุตอเนกประสงค์ ถ้า CLKXI/O=0 แล้ว CLKX เป็นพอร์ตเอาต์พุตอเนกประสงค์
2	CLKXDATOUT	ข้อมูลที่ออกจากขา CLKX
3	CLKXDATIN	ข้อมูลที่เข้ามาจากขา CLKX
4	DXFUNC	ควบคุมการทำงานของ DX ถ้า DXFUNC=0 แล้ว DX จะเป็นพอร์ตอินพุตเอาต์พุต อเนกประสงค์ ถ้า DXFUNC=1 แล้ว DXจะเป็นขาพอร์ตอนุกรม
5	DXI/O	ถ้า DXI/O=0 แล้ว DX เป็นพอร์ตอินพุตอเนกประสงค์ ถ้า DXI/O=0 แล้ว DX เป็นพอร์ตเอาต์พุตอเนกประสงค์
6	DXDATOUT	ข้อมูลที่ออกจากขา DX
7	DXDATIN	ข้อมูลที่เข้ามาจากขา DX
8	FSXFUNC	ควบคุมการทำงานของ FSX ถ้า FSXFUNC=0 แล้ว FSX จะเป็นพอร์ตอินพุตเอาต์พุต อเนกประสงค์ ถ้า FSXFUNC=1 แล้ว FSXจะเป็นขาพอร์ตอนุกรม
9	FSXI/O	ถ้า FSXI/O=0 แล้ว FSX เป็นพอร์ตอินพุตอเนกประสงค์ ถ้า FSXI/O=0 แล้ว FSX เป็นพอร์ตเอาต์พุตอเนกประสงค์
10	FSXDATOUT	ข้อมูลที่ออกจากขา FSX
11	FSXDATIN	ข้อมูลที่เข้ามาจากขา FSX
31-12	RESERVE	เป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.1.4.2.3 รีจิสเตอร์ควบคุมฝ่ายรับ (FSR/DR/CLKR Port control) เป็นรีจิสเตอร์ขนาด 32 บิต ควบคุมการทำงานของพอร์ตอนุกรมในส่วนของ FSR,DR และ CLKR**

ตาราง 2-4 แสดงบิตต่าง ๆ ของรีจิสเตอร์ควบคุมฝ่ายรับของพอร์ตอนุกรม

บิต	ชื่อ	การทำงาน
0	CLKRFUNC	ควบคุมการทำงานของ CLKR ถ้า CLKRFUNC=0 แล้ว CLKR จะเป็นพอร์ตอินพุตเอาท์พุต อเนกประสงค์ ถ้า CLKRFUNC=1 แล้ว CLKRจะเป็นขาพอร์ตอนุกรม
1	CLKRI/O	ถ้า CLKRI/O=0 แล้ว CLKR เป็นพอร์ตอินพุตอเนกประสงค์ ถ้า CLKRI/O=1 แล้ว CLKR เป็นพอร์ตเอาท์พุตอเนกประสงค์
2	CLKRDATOUT	ข้อมูลที่ออกจากขา CLKR
3	CLKRDATIN	ข้อมูลที่เข้ามาจากขา CLKR
4	DRFUNC	ควบคุมการทำงานของ DR ถ้า DRFUNC=0 แล้ว DR จะเป็นพอร์ตอินพุตเอาท์พุต อเนกประสงค์ ถ้า DRFUNC=1 แล้ว DRจะเป็นขาพอร์ตอนุกรม
5	DRI/O	ถ้า DRI/O=0 แล้ว DR เป็นพอร์ตอินพุตอเนกประสงค์ ถ้า DRI/O=1 แล้ว DR เป็นพอร์ตเอาท์พุตอเนกประสงค์
6	DRDATOUT	ข้อมูลที่ออกจากขา DR
7	DRDATIN	ข้อมูลที่เข้ามาจากขา DR
8	FSRFUNC	ควบคุมการทำงานของ FSR อเนกประสงค์ ถ้า FSRFUNC=1 แล้ว FSRจะเป็นขาพอร์ตอนุกรม
9	FSRI/O	ถ้า FSRI/O=0 แล้ว FSR เป็นพอร์ตอินพุตอเนกประสงค์ ถ้า FSRI/O=1 แล้ว FSR เป็นพอร์ตเอาท์พุตอเนกประสงค์
10	FSRDATOUT	ข้อมูลที่ออกจากขา FSR
11	FSRDATIN	ข้อมูลที่เข้ามาจากขา FSR
31-12	RESERVE	เป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.1.4.2.4 รีจิสเตอร์ควบคุมตัวจับเวลาของฝ่ายรับและฝ่ายส่ง (Receive/Transmit Cimer Control Register)** เป็นรีจิสเตอร์ขนาด 32 บิต ควบคุมการทำงานของตัวจับเวลา โดยบิต 5-0 จะควบคุมตัวจับเวลาของฝ่ายส่ง และบิต 11-6 ควบคุมการทำงานของตัวจับเวลาของฝ่ายรับ

ตาราง 2-5 แสดงบิตต่าง ๆ ของรีจิสเตอร์ควบคุมตัวจับเวลาของฝ่ายรับ และส่งของพอร์ตอนุกรม

บิต	ชื่อ	การทำงาน
0	XGO	เป็นบิตรีเซ็ตและเริ่มการทำงานของตัวนับเวลาของฝ่ายส่ง ถ้า XGO=1และไม่คงค่าตัวจับเวลา แล้วตัวนับเวลาจะเป็น 0 แล้วเริ่มทำการนับตามขอบขาขึ้นของสัญญาณนาฬิกา แล้ว XGO จะเคลียร์
1	$\overline{\text{XHLD}}$	ถ้า $\overline{\text{XHLD}}=0$ แล้วตัวนับเวลาไม่ทำงานและคงค่าปัจจุบันเอาไว้ เมื่อออกจาก $\overline{\text{XHLD}}$ ตัวนับเวลาจะนับต่อไป
2	XC/P	ถ้า XC/P=1 ฝ่ายส่งจะเป็นโหมดนาฬิกา การส่งสัญญาณของแฟลกสถานะ และเอาต์พุตภายนอกจะเป็น 50% ของคิวต์ไซ้เกิด (Duty cycle)
3	XCLKSRC	ถ้า CLKSRC=1 แล้วความถี่สัญญาณนาฬิกาภายในจะเท่ากับครึ่งหนึ่งของ CLKOUT ที่ใช้ในการนับขึ้นของตัวนับเวลา ถ้า CLKSRC=0 แล้วเราสามารถใช้นาฬิกาภายนอกจาก CLKX ในการเพิ่มค่าของตัวนับเวลา
4	RESERVE	เป็น 0
5	XTSTAT	บอกสถานะของตัวจับเวลาฝ่ายส่ง
6	RGO	เป็นบิตรีเซ็ตและเริ่มการทำงานของตัวนับเวลาของฝ่ายรับ ถ้า RGO=1และไม่คงค่าตัวจับเวลา แล้วตัวนับเวลาจะเป็น 0 แล้วเริ่มทำการนับตามขอบขาขึ้นของสัญญาณนาฬิกา แล้ว RGO จะเคลียร์
7	$\overline{\text{RHLD}}$	ถ้า $\overline{\text{RHLD}}=0$ แล้วตัวนับเวลาไม่ทำงานและคงค่าปัจจุบันเอาไว้ เมื่อออกจาก $\overline{\text{RHLD}}$ ตัวนับเวลาจะนับต่อไป
8	RC/P	ถ้า RC/P=1 ฝ่ายรับจะเป็นโหมดนาฬิกา การรับสัญญาณของแฟลกสถานะ และเอาต์พุตภายนอกจะเป็น 50% ของคิวต์ไซ้เกิด (Duty cycle)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9	RCLKSRC	ถ้า CLKSRC=1 แล้วความถี่สัญญาณนาฬิกาภายในจะเท่ากับครึ่งหนึ่งของ CLKOUT ที่ใช้ในการนับขึ้นของตัวนับเวลา ถ้า CLKSRC=0 แล้วเราสามารถใช้นาฬิกาภายนอกจาก CLKR ในการเพิ่มค่าของตัวนับเวลา
10	RESERVE	เป็น 0
11	RTSTAT	บอกสถานะของตัวจับเวลาฝ่ายรับ
31-12	RESERVE	เป็น 0

**2.1.4.2.5 รีจิสเตอร์ตัวจับเวลา/ตัวนับเวลาของฝ่ายรับและฝ่ายส่ง (Receive/Transmit Timer-Counter Register)** เป็นรีจิสเตอร์ขนาด 32 บิต โดยบิต 15-0 เป็นตัวจับเวลา/ตัวนับเวลาของฝ่ายส่ง และบิต 31-16 เป็นตัวจับเวลา/ตัวนับเวลาของฝ่ายรับ แต่ละตัวนับเวลาสามารถเคลียร์เป็น 0 ได้เมื่อทำการนับขึ้นไปจนเท่ากับค่าในรีจิสเตอร์คาบเวลา

**2.1.4.2.6 รีจิสเตอร์คาบเวลาของฝ่ายรับและฝ่ายส่ง (Receive/Transmit Timer-Period Register)** เป็นรีจิสเตอร์ขนาด 32 บิต โดยบิต 15-0 เป็นคาบเวลาของฝ่ายส่ง และบิต 31-16 เป็นคาบเวลาของฝ่ายรับ แต่ละรีจิสเตอร์สามารถให้ค่าคาบที่ต่างกันได้

**2.1.4.2.7 รีจิสเตอร์ข้อมูลฝ่ายส่ง (Data Transmit Register,DXR)** เป็นรีจิสเตอร์ขนาด 32 บิต เมื่อข้อมูลใน DXR ถูกโหลดแล้วฝ่ายส่งจะทำการโหลดข้อมูลลงไปในรีจิสเตอร์เลื่อนของฝ่ายส่ง (XSR)และทำการเลื่อนข้อมูลออกไป เวิร์ดข้อมูลจะไม่ถูกโหลดเข้าไปในรีจิสเตอร์เลื่อนจนกว่ารีจิสเตอร์จะว่าง เมื่อข้อมูลในรีจิสเตอร์ข้อมูลฝ่ายส่งถูกโหลดเข้าไปในรีจิสเตอร์เลื่อนของฝ่ายส่งแล้ว บิต XRDYจะเซ็ท เพื่อเป็นการบอกว่าพร้อมที่จะรับข้อมูลใหม่แล้ว การเลื่อนจะเป็นการเลื่อนไปทางซ้าย

**2.1.4.2.8 รีจิสเตอร์ข้อมูลฝ่ายรับ (Data Receive Register,DRR)** เป็นรีจิสเตอร์ขนาด 32 บิต เมื่อพอร์ตอนุกรมเป็นอินพุตฝ่ายรับจะเลื่อนบิตข้อมูลเข้าไปในรีจิสเตอร์เลื่อนของฝ่ายรับ เมื่อเข้าไปครบแล้วข้อมูลจะถูกโหลดเข้ามาในรีจิสเตอร์ข้อมูลฝ่ายรับ แล้วบิต RRDY จะเซ็ท ถ้ารีจิสเตอร์ข้อมูลฝ่ายรับยังไม่อ่านข้อมูล และรีจิสเตอร์เลื่อนของฝ่ายรับยังเต็มอยู่ ตัวรับจะไม่สนใจข้อมูลที่เข้ามาขณะนั้น นั่นคือ รีจิสเตอร์ข้อมูลของฝ่ายรับจะต้องอ่านข้อมูลก่อน เพื่อเป็นการอนุญาตให้ข้อมูลใหม่จากรีจิสเตอร์เลื่อนของฝ่ายรับส่งข้อมูลมาได้

ตำแหน่งของรีจิสเตอร์ของพอร์ตอนุกรมดังรูปที่ 2-7

รีจิสเตอร์	ตำแหน่ง
รีจิสเตอร์ควบคุมส่วนกลาง	808040h
รีจิสเตอร์ควบคุมฝ่ายส่ง	808042h
รีจิสเตอร์ควบคุมฝ่ายรับ	808043h
รีจิสเตอร์ควบคุมตัวจับเวลาของฝ่ายรับและฝ่ายส่ง	808044h
รีจิสเตอร์ตัวจับเวลา/ตัวนับเวลาของฝ่ายรับและฝ่ายส่ง	808045h
รีจิสเตอร์คาบเวลาของฝ่ายรับและฝ่ายส่ง	808046h
รีจิสเตอร์ข้อมูลฝ่ายส่ง	808048h
รีจิสเตอร์ข้อมูลฝ่ายรับ	80804Ch

รูปที่ 2-7 แสดงตำแหน่งของรีจิสเตอร์ของพอร์ตอนุกรม

#### 2.1.4.3 การเข้าถึงหน่วยความจำโดยตรง (Direct Memory Access : DMA)

ตัวควบคุมการเข้าถึงหน่วยความจำโดยตรงบนชิป สามารถอ่านหรือเขียนลงตำแหน่งต่างๆ ใน แผนที่หน่วยความจำ โดยไม่เข้าไปแทรกแซงการทำงานของหน่วยประมวลผลกลาง, ดังนั้น TMS320C3x สามารถอินเทอร์เฟสกับหน่วยความจำภายนอกและอุปกรณ์ภายนอก โดยผ่านทางหน่วยประมวลผลหลักซึ่งข้อมูลนั้นจะไม่ถูกลดทอนลง ตัวควบคุมการเข้าถึงหน่วยความจำโดยตรง จะบรรจุตัวสร้างตำแหน่ง (address generator) ของตัวเอง, รีจิสเตอร์ต้นกำเนิดและจุดหมาย และตัวนับการส่งย้ายข้อมูล (transfer counter) บัซของตำแหน่งและบัซของข้อมูลของการเข้าถึงหน่วยความจำหน้าที่พิเศษ จะลดขนาดความขัดแย้งระหว่างหน่วยประมวลผลกลาง และตัวควบคุมการเข้าถึงหน่วยความจำให้น้อยลง การทำงานของการเข้าถึงหน่วยความจำประกอบด้วยบัสล็อก หรือเวิร์ดเคียว ส่งย้ายไปยังหรือไปจากหน่วยความจำ

## 2.2 โครงสร้างของ TMS320C3x ดีเอสเค (DSP Starter Kit : DSK)

### 2.2.1 โครงสร้างของดีเอสเค

- TMS320C31 เป็นดีเอสพีอิงครรชนี่
- คำสั่งมีเวลาคาบรอบ (cycle time) 40 นาโนวินาที, 50 ล้านคำสั่งอิงครรชนี่ต่อวินาที (MFLOPS), 25 ล้านคำสั่งต่อวินาที (MIPS)
- อินเทอร์เฟซพอร์ตเครื่องพิมพ์แบบขนาน (parallel printer port interface) ที่เป็นมาตรฐานหรือพัฒนาแล้ว เพื่อต่อกับเครื่องคอมพิวเตอร์แม่ (host PC<sup>TW</sup>) และTMS320C31 ติดต่อกับโปรแกรมของเครื่องคอมพิวเตอร์ได้
- รับข้อมูลอนาลอกโดยผ่านวงจรเชื่อมต่อสัญญาณอนาลอก (AIC) TLC32040
  - อัตราตัวแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล (ADC) และตัวแปลงสัญญาณสัญญาณดิจิทัลเป็นสัญญาณอนาลอก (DAC) สามารถเปลี่ยนแปลงพิสัยแบบพลวัตได้ 14 บิตใน 20000 ตัวอย่างต่อวินาที
  - มีฟิลเตอร์สร้างเอาท์พุทใหม่ และสามารถบายพาส (bypass) ได้, ตัวเก็บประจุแบบสวิตช์ (switched-capacitor) ป้องกันสัญญาณรบกวน เป็นอินพุตฟิลเตอร์
- ใช้คอนเนคเตอร์ปลั๊กแบบ RCA มาตรฐาน สำหรับอนาลอกอินพุตและเอาท์พุต
- คอนเนคเตอร์อีมูเลเตอร์ XDS510
- คอนเนคเตอร์เพิ่มเติมที่กำหนดขา TMS320C31 ทั้งหมดสำหรับใช้กับบอร์ดดีเอสเคตัวลูก

### 2.2.2 ภาพรวมของ DSK

ส่วนประกอบพื้นฐานของดีเอสเคมีดังนี้

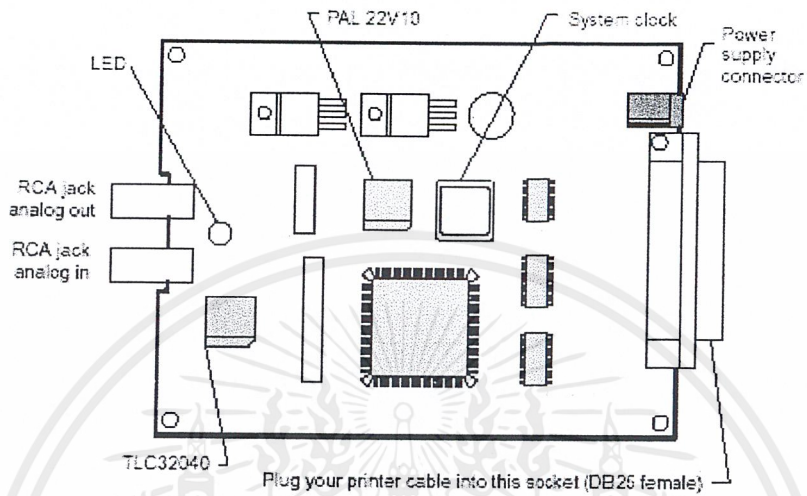
- DSP รุ่น TMS320C31
- วงจรเชื่อมต่อสัญญาณอนาลอกTLC32040
- คอนเนคเตอร์เพิ่มเติม
- สัญญาณนาฬิกาของระบบ (System clock)
- อินเทอร์เฟซพอร์ตของเครื่องพิมพ์แบบขนาน
- ไฟแสดงผลสามสี (Tri-color LED)

ส่วนประกอบพื้นฐานของดีเอสเคแสดงดังรูปที่ 2-8

เส้นทางสัญญาณทั้งหมดของ TMS320C3x จะไปยังคอนเนคเตอร์เพิ่มเติม โดยคอนเนคเตอร์เพิ่มเติมจะรวมทั้ง เฮดเดอร์ (header) 32 ขา 4 ตัว , บล็อกจัมเปอร์ (jumper block) 11 ขา 1 ตัว และเฮดเดอร์ XDS510 12 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรเชื่อมต่อสัญญาณแบบอนาลอก TLC32040 ต่อกับพอร์ตอนุกรมของ TMS320C3x  
ใช้ คอนเนคเตอร์ RCA 2 ตัวเป็นอนาลอกอินพุตและเอาต์พุตของบอร์ด



รูปที่ 2-8 แสดงส่วนประกอบพื้นฐานของดีเอสเค

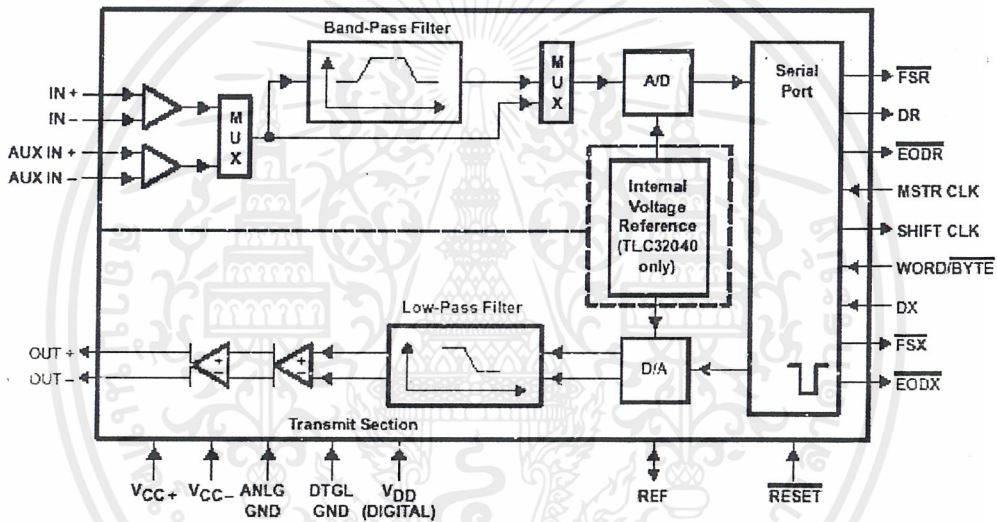
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3 วงจรเชื่อมต่อสัญญาณอนาล็อก

การใช้ดีเอสพีในการประยุกต์ใช้งานนั้นจะต้องมีส่วนประกอบที่สำคัญคือ ตัวที่เชื่อมต่อสัญญาณอนาล็อกเพื่อจะติดต่อกับสัญญาณอินพุตอนาล็อกและสัญญาณเอาต์พุตอนาล็อก ด้านอินพุตจะมีฟิวเตอร์ที่มีความถี่มากกว่า 1/2 เท่าของความถี่ซิกตัวอย่าง (Sampling frequency) ซึ่งตามทฤษฎีซิกคาคตัวอย่าง ความถี่ที่ใช้ซิกคตัวอย่าง ต้องมีค่าน้อยกว่า 2 เท่าของความถี่สูงสุดในสัญญาณ

$$F_s > 2f$$

วงจรเชื่อมต่อสัญญาณอนาล็อก (Analog interface circuit ; AIC) ของ DSK คือ TLC32040 ซึ่งจะเชื่อมต่อกับพอร์ตอนุกรมของ TMS320C31 มีลักษณะดังรูปที่ 2-9



รูปที่ 2-9 แสดงบล็อกไดอะแกรมของวงจรเชื่อมต่อสัญญาณอนาล็อก

ลักษณะ และส่วนประกอบที่สำคัญดังนี้ คือ

- มี 2 อินพุต (IN+ IN-, AUX IN+ AUX IN-)

IN สามารถต่อใช้งานได้โดยใช้งานผ่านตัวคอนเนคเตอร์ RCA บน DSK บอร์ด

AUX สามารถต่อใช้งานผ่านขา 3 จาก JP3

การเลือกใช้งานระหว่าง IN, AUX สามารถเลือกใช้ได้จากบิตควบคุมที่จะกล่าวถึงต่อไป

- มี 1 เอาต์พุต (OUT+, OUT-)

OUT สามารถต่อใช้งานได้โดยใช้งานผ่านตัวคอนเนคเตอร์ RCA บน DSK บอร์ดเช่นเดียวกับ

IN

- ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล และตัวแปลงสัญญาณดิจิตอลเป็นอนาล็อก ขนาด 14 บิต

- ฟิลเตอร์อินพุตและฟิลเตอร์เอาต์พุต (Switched-capacity antialiasing input filter & reconstruction output filter)

ฟิลเตอร์อินพุตเป็นฟิลเตอร์ที่ทำให้ช่วงความถี่ผ่าน (bandpass filter) ที่จะทำให้ความถี่ผ่านในช่วง  $300 \text{ Hz} < f < 3.4 \text{ kHz}$  สามารถเลือกใช้งานหรือไม่ก็ได้โดยผ่านการควบคุมด้วยบิตควบคุม นอกจากนี้ฟิลเตอร์อินพุตนี้ยังสามารถปรับความถี่สำหรับความถี่ cut off หรือ bandwidth ที่ได้ออกแบบเอาไว้ โดยจะขึ้นอยู่กับความถี่ซิกตัวอย่าง

ฟิลเตอร์เอาต์พุต เป็นฟิลเตอร์แบบความถี่ต่ำผ่าน (low-pass filter) โดยจะยอมให้มีความถี่ต่ำกว่า  $3.4 \text{ kHz}$  ผ่าน โดยไม่สามารถปรับค่าต่าง ๆ ได้

- ใช้เทคโนโลยี CMOS

ความถี่ซิกตัวอย่างสามารถโปรแกรมได้จนถึง  $20 \text{ kHz}$

- ความถี่ Master Clock เป็นตัวที่กำหนดสัญญาณทุกอย่างในวงจรเชื่อมต่อสัญญาณอนาล็อก Master clock จะได้รับสัญญาณมาจากตัวจับเวลา 0 ของ TMS320C31

- Shift Clock เป็นสัญญาณที่ได้มาจากการหาร Master clock ด้วย 4 ใช้สำหรับเป็น สัญญาณนาฬิกาของการส่งข้อมูลแบบอนุกรมของ วงจรเชื่อมต่อสัญญาณอนาล็อก และ TMS320C31

- ขา DR เป็นขาที่ใช้ในการส่งข้อมูลจาก วงจรเชื่อมต่อสัญญาณอนาล็อก ไปที่พอร์ตอนุกรมของ TMS320C31 โดยการส่งจะส่งเป็นจังหวะเดียวกับความถี่ Shift clock

- ขา DX เป็นขาที่ใช้ในการรับข้อมูลจากพอร์ตอนุกรมของ TMS320C31 ไปที่ วงจรเชื่อมต่อสัญญาณอนาล็อก โดยการส่งจะส่งเป็นจังหวะเดียวกับความถี่ Shift clock

- ขา FSR สัญญาณของ FSR จะเป็นสถานะต่ำระหว่างการส่งข้อมูล เมื่อ FSR เป็นสถานะต่ำ TMS320C31 จะเริ่มการรับบิตจาก วงจรเชื่อมต่อสัญญาณอนาล็อก ผ่าน DR ของ วงจรเชื่อมต่อสัญญาณอนาล็อก บิต MSB ของ DR จะถูกส่งการที่ FSR จะเป็นสถานะต่ำ FSR จะไม่เกิดขึ้นหลังจากการติดต่อแบบ Secondary

- ขา FSX สัญญาณของ FSX จะเป็นต่ำ เมื่อ พอร์ตอนุกรมของ TMS320C31 เริ่มการส่งบิตไปยัง วงจรเชื่อมต่อสัญญาณอนาล็อก ผ่าน DX ของ วงจรเชื่อมต่อสัญญาณอนาล็อก ในการติดต่อทั้งหมด FSX จะมีสถานะต่ำระหว่างการส่ง

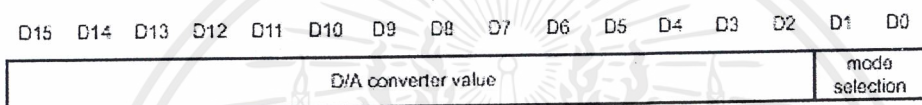
- รีเซต ฟังก์ชันรีเซตใช้สำหรับการกำหนดค่าเริ่มต้นให้กับ TA, TA', TB, RA, RA', RB และ รีจิสเตอร์ควบคุม ฟังก์ชันรีเซตนี้จะกำหนดค่าเริ่มต้นของการติดต่อแบบอนุกรมระหว่าง วงจรเชื่อมต่อสัญญาณอนาล็อก และ DSP หลังจากมี negative-going pulse มาที่ขา RESET รีจิสเตอร์ของ วงจรเชื่อมต่อสัญญาณอนาล็อก จะถูกกำหนดค่าทำให้สร้างอัตราการแปลงข้อมูลเท่ากับ  $8 \text{ kHz}$  และมี master clock เท่ากับ  $5.184 \text{ MHz}$  รีเซตฟังก์ชันรีเซตใช้สำหรับการกำหนดค่าเริ่มต้นให้กับ TA, TA',

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TB, RA, RA', RB และ รีจิสเตอร์ควบคุม ฟังก์ชันรีเซตนี้จะกำหนดค่าเริ่มต้นของการติดต่อแบบอนุกรมระหว่าง วงจรเชื่อมต่อสัญญาณอนาล็อก และ DSP หลังจากมี negative-going pulse มาที่ขา RESET รีจิสเตอร์ของ วงจรเชื่อมต่อสัญญาณอนาล็อก จะถูกกำหนดค่าทำให้สร้างอัตราการแปลงข้อมูลเท่ากับ 8 kHz และมี master clock เท่ากับ 5.184 MHz

### การใช้งาน และควบคุม วงจรเชื่อมต่อสัญญาณอนาล็อก

การส่งข้อมูลเกิดขึ้นผ่าน DR และ DX ซึ่งเป็นรีจิสเตอร์อนุกรมของ วงจรเชื่อมต่อสัญญาณอนาล็อก ซึ่งการส่งข้อมูลแต่ละครั้งจะมีรูปแบบการส่งดังรูปที่ 2-10 คือ



รูปที่ 2-10 รูปแบบของบิตในการส่งข้อมูลของ DR or DX

LSBs	Mode
00	Tx Counter A --- TA, Rx Counter A --- RA Tx Counter B --- TB, Rx Counter B --- RB
01	Tx Counter A --- TA + TA', Rx Counter A --- RA + RA' Tx Counter B --- TB, Rx Counter B --- RB
10	Tx Counter A --- TA - TA', Rx Counter A --- RA + RA' Tx Counter B --- TB, Rx Counter B --- RB
11	Tx Counter A --- TA, Rx Counter A --- RA Tx Counter B --- TB, Rx Counter B --- RB

รูปที่ 2-11 แสดงบิตควบคุม D0, D1 ของการส่งข้อมูลในแต่ละแบบ

จากรูปที่ 2-11 บิต D0 และ D1 ใช้สำหรับควบคุมฟังก์ชันการทำงานของ การสื่อสาร โดยถ้า D1, D0 มีค่าเท่ากับ 0 (โหมดที่ 1) จะเป็นการส่งแบบปกติ แต่ถ้า มีค่าเท่ากับ 1 ทั้งสองตัว (โหมดที่ 4) ทำให้เกิดการร้องขอการส่งแบบ secondary ซึ่งทำให้เกิดการส่งแบบ secondary ตามมาหลังจาก shift clock 4 ลูก ซึ่งการส่งแบบ secondary เป็นการส่งคำสั่งควบคุมเพื่อกำหนดค่าเริ่มต้น และควบคุม วงจรเชื่อมต่อสัญญาณอนาล็อก โดยจะส่งหนึ่งครั้ง และจะกลับไปเป็นการส่งแบบปกติตามเดิม ส่วนในกรณีที่มีตัวใดตัวหนึ่งเป็น 0 และอีกตัวเป็น 1 จะทำให้เกิดการเปลี่ยนความถี่การซักรค่าตัวอย่างใน

คาบถัดไป ส่วนการส่งในโหมด 2 และ 3 ใช้เพื่อเป็นการเพิ่ม หรือลดความถี่ซีกค่าตัวอย่างโดยการลด หรือเพิ่มคาบการซีกตัวอย่าง ซึ่งจะใช้อย่างมากในการประยุกต์ใช้กับโมเด็ม

การส่งแบบ secondary จะส่ง รีจิสเตอร์ A ซึ่งประกอบด้วย TA, RA , รีจิสเตอร์ B ซึ่งประกอบด้วย TB, RB รีจิสเตอร์ A' ซึ่งประกอบด้วย TA', RA' และ บิตควบคุม ซึ่งจะควบคุมการทำงานของ วงจรเชื่อมต่อสัญญาณอนาล็อก ค่า TA, TB สามารถนำมาใช้ดังนี้

- ออกแบบค่าความถี่ของฟิวเตอร์อินพุต (Switched-capacitor filter frequency) ได้ โดยคิดจาก

$$\text{SCF Clock Frequency} = \text{Master Clock Frequency} / (2 \times \text{TA})$$

เพื่อให้ฟิวเตอร์อินพุตมีการทำงานตามคุณลักษณะของฟังก์ชันถ่ายโอน จะต้องมีความถี่ Switched-capacitor clock เท่ากับ 288 kHz ซึ่งจะทำให้มี bandwidth หรือความถี่ cut-off เท่ากับ 3600 Hz

- ออกแบบค่าความถี่การซีกค่าตัวอย่าง (Sampling Frequency)

$$\text{Sampling Frequency} = \text{Master Clock Frequency} / (2 \times \text{TA} \times \text{TB})$$

ส่วนค่า Shift Clock หาได้จาก

$$\text{Shift Clock} = \text{Master Clock} / 4$$

สำหรับค่า TA และ TB จะใช้ในส่วนของการแปลงสัญญาณอนาล็อกเป็นดิจิทัล ส่วน RA และ RB จะใช้ในการทำงานเดียวกันในส่วนของการแปลงสัญญาณดิจิทัลเป็นอนาล็อก รูปแบบการส่งของ Secondary communication คือ

ตารางที่ 2-6 แสดงรูปแบบการส่งแบบ secondary

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
รีจิสเตอร์ A	X	X						X	X						0	0
รีจิสเตอร์ A'		X						X							0	1
รีจิสเตอร์ B		X						X							1	0
รีจิสเตอร์ควบคุม	X	X	X	X	X	X	X	X	D7	D6	D5	D4	D3	D2	1	1

บิตควบคุมมีดังนี้ คือ

D2 = 0/1 ใช้/ไม่ใช้ ฟิวเตอร์อินพุต

D3 = 0/1 ใช้/ไม่ใช้ ฟังก์ชันวนรูป

D4 = 0/1 ใช้/ไม่ใช้ AUX IN+ และ AUX IN-

D5 = 0/1 ซิงโครไนส์ หรือ อะซิงโครไนส์ การส่ง และรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

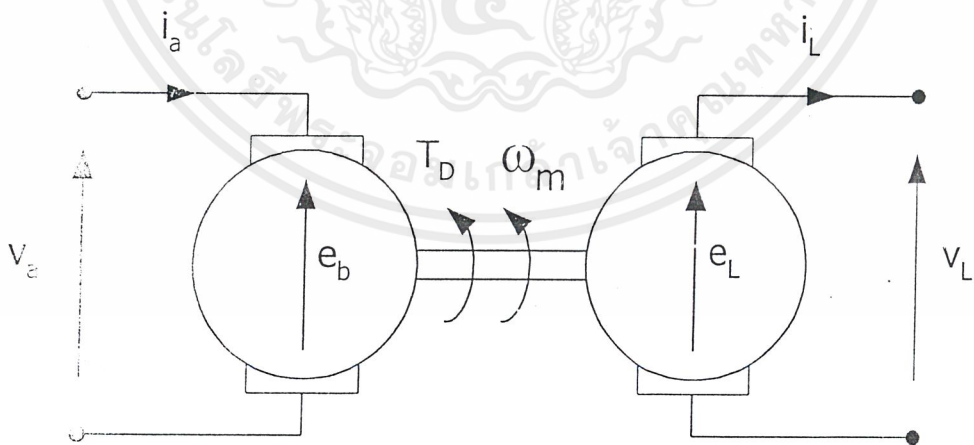
D6, D7 = 0/1 บิตควบคุมอัตราขยาย ซึ่งจะมีค่าดังรูปที่ 2-12

INPUT CONFIGURATIONS	CONTROL REGISTER BITS		ANALOG INPUT†	A/D CONVERSION RESULT
	d6	d7		
Single-ended configuration Analog input = IN+ – ANLG GND = AUX IN+ – ANLG GND	1	1	±3 V	Half scale
	0	0		
	1	0	±3 V	Full scale
	0	1	±1.5 V	Full scale

รูปที่ 2-12 แสดงบิตควบคุมการขยาย

### 2.4 มอเตอร์กระแสตรง

มอเตอร์กระแสตรงประเภท Permanent Magnet การหมุนของมอเตอร์กระแสตรงเกิดจากการดึงดูดกันของสนามแม่เหล็กสองสนามแม่เหล็ก คือ สนามแม่เหล็กที่เกิดจากการ Permanent Magnet  $k\Phi$  (ที่ Stator) กับสนามแม่เหล็กที่เกิดจากการให้กระแสไฟฟ้า  $I_a$  ไหลเข้าที่อาร์มเจอร์ที่โรเตอร์ สามารถเขียนสมการที่เกี่ยวข้องได้กับรูปที่ 2-13 ดังนี้



รูปที่ 2-13 แสดงรูปตัวแปรต่างๆ ของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการของแรงดัน

$$v_a = i_a R_a + L_a \frac{di_a}{dt} + e_b \quad (2.1)$$

$R_a$  : ความต้านทานภายในของอาร์เมเจอร์ ( $\Omega$ )

$L_a$  : ความเหนี่ยวนำภายในของอาร์เมเจอร์ (H)

$e_b$  : แรงเคลื่อนย้อนกลับ (back emf. :V)

สมการของแรงเคลื่อนย้อนกลับ

$$e_b = k\Phi\omega \quad (2.2)$$

$\omega$  : ความเร็วรอบ (Rad/Sec.)

$\Phi$  : สนามแม่เหล็กจาก Permanent Magnet

$k$  : ค่าคงที่

สมการของแรงบิดที่มอเตอร์สร้างขึ้น

$$T_D = T_L + J \frac{d\omega}{dt} + B\omega \quad | \quad B \approx 0$$

$$T_D = T_L + J \frac{d\omega}{dt} \quad (2.3)$$

$$T_D = k\Phi i_a \quad (2.4)$$

$T_L$  : Load torque (N-m)

$J$  : Moment of Inertia ( $\text{Kg-m}^2$ )

สมการที่ (2.1) (2.2) (2.3) (2.4) นำมาแปลงลาปลาซ ได้เป็น

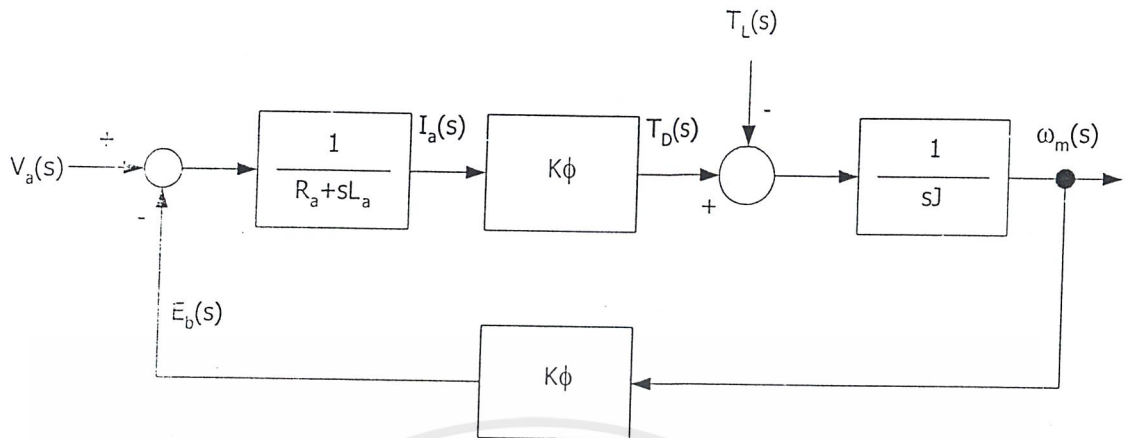
$$V_a(s) = R_a I_a(s) + sL_a I_a(s) + E_b(s) \quad (2.5)$$

$$E_b(s) = k\Phi\omega(s) \quad (2.6)$$

$$T_D(s) = T_L(s) + sJ\omega(s) \quad (2.7)$$

$$T_D = k\Phi I_a(s) \quad (2.8)$$

จากสมการ (2.5) (2.6) (2.7) (2.8) สามารถเขียน บล็อกไดอะแกรม ของมอเตอร์ได้ดังรูปที่



รูปที่ 2-14 แสดงบล็อกไดอะแกรมของมอเตอร์

จากรูปที่ 2-14 จะถูกนำไปใช้ในการออกแบบระบบควบคุมต่อไป แต่ก่อนที่จะออกแบบได้ ต้องรู้พารามิเตอร์ของมอเตอร์ก่อน คือ  $R_a$ ,  $L_a$ ,  $k\Phi$  และ  $J$  ถ้าพิจารณาความสัมพันธ์ระหว่าง  $W(s)$  กับ  $V_a(s)$  โดยไม่พิจารณา  $T_L(s)$

$$\frac{w(s)}{V_a(s)} = \frac{k\Phi}{(R_a + sL_a)sJ + (k\Phi)^2} \quad (2.9)$$

$$\frac{w(s)}{V_a(s)} = \frac{1}{\left( s^2 \frac{L_a J}{k\Phi^2} + s \frac{R_a J}{k\Phi^2} + 1 \right) k\Phi} \quad (2.10)$$

กำหนด 
$$\tau_m = \frac{R_a J}{(k\Phi)^2} \quad (2.11)$$

$$\tau_e = \frac{L_a}{R_a} \quad (2.12)$$

สมการ (2-10) สามารถจัดรูปใหม่ได้คือ

$$\frac{w(s)}{V_a(s)} = \frac{1}{(s^2 \tau_m \tau_e + s\tau_m + 1)k\Phi} \quad (2.13)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 ตัวควบคุมแบบพีไอดี (PID Controller)

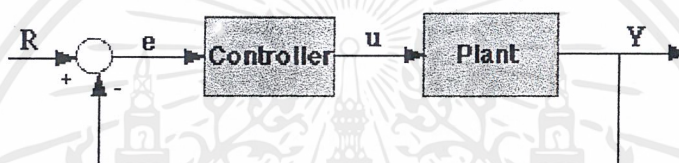
ฟังก์ชันถ่ายโอนของตัวควบคุมแบบพีไอดีจะมีลักษณะดังนี้

$$K_p + \frac{K_i}{s} + K_D s = \frac{K_D s^2 + K_p s + K_i}{s}$$

$K_p$  = ค่าขยายตามอัตราส่วน

$K_i$  = ค่าขยายอินทิกรัล

$K_d$  = ค่าขยายเดอริเวทีฟ (Derivative)



รูปที่ 2-15 แสดง ระบบวงปิดที่ใช้ตัวควบคุม

จากรูปที่ 2-15  $e$  แสดงถึงค่าความคลาดเคลื่อน ซึ่งเป็นผลต่างของค่าอินพุต ( $R$ ) และค่าเอาต์พุต ( $Y$ ) ค่าความคลาดเคลื่อนนี้จะถูกส่งไปที่ตัวควบคุมแบบพีไอดี และตัวควบคุมจะคำนวณสัญญาณความคลาดเคลื่อน และส่งสัญญาณ  $u$  เพื่อไปควบคุมระบบ โดย  $u$  จะมีค่าดังนี้

$$u = K_p e + K_i \int e dt + K_D \frac{de}{dt}$$

เมื่อได้เอาต์พุตใหม่ เอาต์พุตจะถูกส่งกลับไปคำนวณค่าความคลาดเคลื่อนอีกครั้ง และตัวควบคุมก็จะหาสัญญาณควบคุมใหม่ เป็นเช่นนี้ตลอดไป

**ลักษณะของตัวควบคุมแบบพี ไอ และแบบดี**

ตัวควบคุมแบบพีจะมีผลต่อการลดค่า rise time แต่จะไม่กำจัดค่าความคลาดเคลื่อนที่สถานะสมดุล (steady-state error) ตัวควบคุมแบบไอ จะกำจัดค่าความคลาดเคลื่อนที่สถานะสมดุล แต่อาจทำให้ผลตอบสนองในช่วง transient แ่่งลง ส่วนตัวควบคุมแบบดี จะเพิ่มเสถียรภาพของระบบ ลด overshoot และทำให้ผลตอบสนองในช่วง transient ดีขึ้น ผลของตัวควบคุมทั้ง 3 แบบรวมอยู่ในตารางที่ 2-7 ดังนี้

ตารางที่ 2-7 แสดงผลตอบสนองต่อตัวควบคุมแบบพี ไอ และดี

ตัวควบคุม	Rise time	Overshoot	Settling time	Steady-state error
แบบพี	ลดลง	เพิ่มขึ้น	เปลี่ยนแปลงเล็กน้อย	ลดลง
แบบไอ	ลดลง	เพิ่มขึ้น	เพิ่มขึ้น	กำจัด
แบบดี	เปลี่ยนแปลงเล็กน้อย	ลดลง	ลดลง	เปลี่ยนแปลงเล็กน้อย

**หลักการแปลงตัวควบคุมพีไอดีแบบต่อเนื่อง เป็นพีไอดีแบบดิสครีต (Discrete)**

หลักการของตัวควบคุม พีไอดีแบบต่อเนื่องขยายออกไปใช้กับแบบดิสครีตได้ ก่อนอื่นเราจะหาพัลซ์ฟังก์ชันถ่ายโอนของตัวควบคุม พีไอดี ดิสครีต ส่วนค่าขยายตามอัตราส่วนยังคงใช้  $K_p$  ส่วนค่าขยายอินทิกรัลใช้การประมาณแบบ trapezoidal ให้  $u(t)$  เป็นอินทิกรัลของ  $e(t)$  ดังนั้นค่าของอินทิกรัลที่  $t=(k+1)T$  คือ

$$u[(k+1)T] = u(kT) + \int_{kT}^{(k+1)T} e(\tau)d\tau$$

$$= u(kT) + T/2 \{e[(k+1)T] + e(kT)\}$$

ดังนั้น

$$zU(z) = U(z) + T/2 [z E(z) + E(z)]$$

$$U(z)/E(z) = T/2 [(z+1)/(z-1)]$$

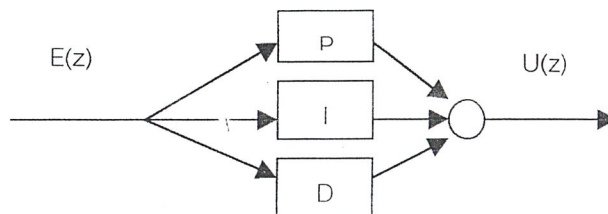
ส่วนค่าขยายประมาณดังนี้ หากค่าขยายของ  $e(t)$  ที่  $t = kt$  คือ  $u(kT)$  ดังนั้น

$$u(kT) = \frac{e(kT) - e[(k-1)T]}{T}$$

แปลง  $z$  ได้

$$U(z)/E(z) = (z-1)/Tz$$

ดังนั้น จากรูปที่ 2-16 ฟังก์ชันถ่ายโอนของ พีไอดี ที่เป็นดิจิทัลคือ



รูปที่ 2-16 แสดงบล็อกไดอะแกรมของพีไอดี

$$U_P(z) = K_p e(k)$$

$$U_I(z) = K_i \frac{T}{2} \left( \frac{1+z^{-1}}{1-z^{-1}} \right) e(z)$$

$$U_D(z) = K_d \frac{(z-1)}{Tz} e(z)$$

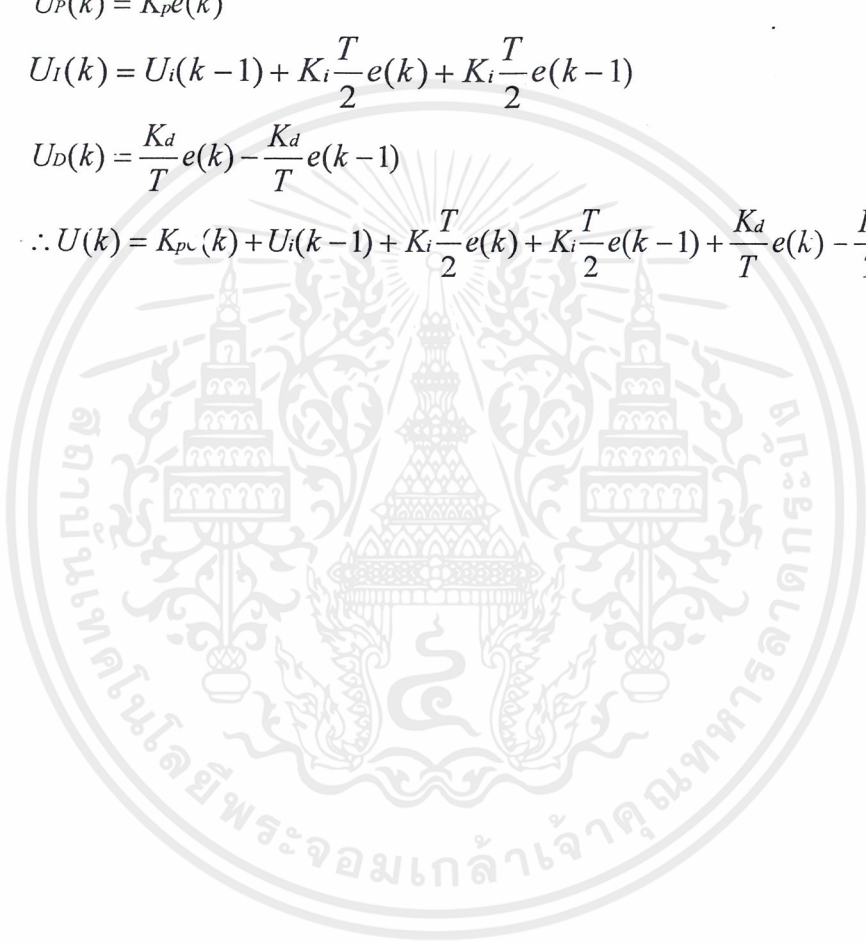
แก้เป็นสมการผลต่างได้

$$U_P(k) = K_p e(k)$$

$$U_I(k) = U_I(k-1) + K_i \frac{T}{2} e(k) + K_i \frac{T}{2} e(k-1)$$

$$U_D(k) = \frac{K_d}{T} e(k) - \frac{K_d}{T} e(k-1)$$

$$\therefore U(k) = K_p e(k) + U_I(k-1) + K_i \frac{T}{2} e(k) + K_i \frac{T}{2} e(k-1) + \frac{K_d}{T} e(k) - \frac{K_d}{T} e(k-1)$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

#### การออกแบบการทดลอง

##### 3.1 การทดลองการเชื่อมต่อกับอุปกรณ์เชื่อมต่อสัญญาณอนาล็อก

การเขียน โปรแกรมจะมีขั้นตอนต่าง ๆ ดังต่อไปนี้

1. กำหนดค่าความถี่การซีกตัวอย่าง ก่อนที่จะทำการกำหนดค่าเริ่มต้นของวงจรเชื่อมต่อสัญญาณอนาล็อก จะต้องทำการกำหนดค่าความถี่การซีกตัวอย่างก่อน ซึ่งการใช้อุปกรณ์เชื่อมต่อสัญญาณอนาล็อกเบอร์ TLC32040 นี้จะสามารถแปลงค่าได้จนถึง 20 kHz การกำหนดความถี่การซีกตัวอย่างสามารถกำหนดได้จากรีจิสเตอร์ A และ รีจิสเตอร์ B ในส่วนการส่ง และรับ ค่าเหล่านี้จะถูกนำไปใส่ไว้ในตัวนับ (counter) ที่เหมาะสมเมื่อ ตัวนับ นั้นมีค่าเป็น 0 Tx ตัวนับ A และ B จะเป็นตัวกำหนดอัตราการแปลงค่าในด้านการแปลงสัญญาณอนาล็อกเป็นดิจิตอล ส่วน Rx ตัวนับ A และ B เป็นตัวกำหนดอัตราการแปลงค่าในด้านการแปลงสัญญาณดิจิตอลเป็นอนาล็อก ในโปรแกรมตัวอย่าง (โปรแกรมตัวอย่างจะอยู่ในบทที่ 4 โปรแกรมแรก) จะใช้ค่าความถี่ซีกตัวอย่าง เท่ากับ 8 kHz ความถี่ Master clock จากตัวจับเวลา 0 ของ TMS320C31 เท่ากับ 6.25 MHz จะได้ค่าต่าง ๆ ดังนี้

$$TA = MCLK / (2 \times SCF) = 6.25 \text{ MHz} / (2 \times 288 \text{ kHz}) = 10.85 \quad (11 = 01011b)$$

$$TB = MCLK / (2 \times TA \times Fs) = 6.25 \text{ MHz} / (2 \times 11 \times 8000) = 35.51 \quad (36 = 100100b)$$

จะได้

$$\text{รีจิสเตอร์ A} = 0001011000101100b = 162Ch$$

$$\text{รีจิสเตอร์ B} = 0100100010010010b = 4892h$$

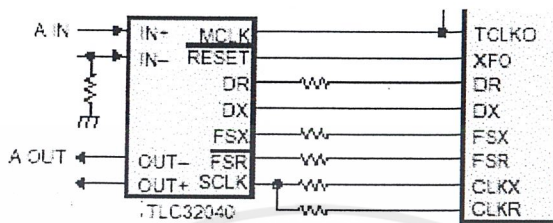
นอกจากจะกำหนดค่า รีจิสเตอร์ A และ B แล้ว ยังต้องกำหนดค่า รีจิสเตอร์ควบคุมด้วย ซึ่งจะกำหนดตามบิตต่าง ๆ ที่ได้กล่าวไปในบททฤษฎีแล้ว

2. กำหนดค่าเริ่มต้นของตัวจับเวลา 0 เนื่องจากอุปกรณ์เชื่อมต่อสัญญาณอนาล็อกมีการเชื่อมต่อกับตัวจับเวลา 0 เพื่อนำสัญญาณที่ขา TCLK0 จากตัวจับเวลา 0 มาเป็น Master clock (ขา MCLK) ของวงจรเชื่อมต่อสัญญาณอนาล็อก ตัวจับเวลา 0 จะส่งสัญญาณที่ขา TCLK0 ทุกครั้งที่ รีจิสเตอร์ timer ตัวนับ มีค่าเท่ากับ รีจิสเตอร์ timer period โดย Master Clock ของวงจรเชื่อมต่อสัญญาณอนาล็อก จะมีค่าตั้งแต่ 75 kHz จนถึง 10.368 MHz สัญญาณ Master clock ที่ใช้ใน โปรแกรมตัวอย่างจะใช้เท่ากับ 6.25 MHz จึงกำหนดค่าของ timer period register 0 เท่ากับ 1 และกำหนดให้ตัวจับเวลา 0 ทำงานใน Clock mode ซึ่งจะมีเอาต์พุตออกมาเป็นคลื่นสี่เหลี่ยมที่มีความถี่เท่ากับ 1/2 เท่าของ pulse mode โดยจะกำหนดค่าในรีจิสเตอร์ต่าง ๆ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ timer period 0                      ตำแหน่ง 808028h กำหนดค่า 0x1  
 รีจิสเตอร์ global control timer 0            ตำแหน่ง 808020h กำหนดค่า 0x3C1

**3. ทำการรีเซตวงจรเชื่อมต่อสัญญาณนาฬิกา**



รูปที่ 3.1 แสดงการเชื่อมต่อระหว่างอุปกรณ์เชื่อมต่อสัญญาณนาฬิกา และ TMS320C31

จากรูป ขา RESET ของอุปกรณ์เชื่อมต่อสัญญาณนาฬิกา เชื่อมต่อกับขา XFO ของ TMS320C31 ดังนั้นจึงสามารถรีเซตอุปกรณ์เชื่อมต่อสัญญาณนาฬิกาได้โดยกำหนดค่าในรีจิสเตอร์ IOF คือ กำหนด 2h ให้ IOF จะเป็นการรีเซต

**4. ค่าเริ่มต้นของพอร์ตอนุกรม** เนื่องจากวงจรเชื่อมต่อสัญญาณนาฬิกาจะส่งข้อมูลเข้า และออก จึงต้องกำหนดให้ขา DX, FSX, CLKX และ DR, FSR, CLKR ทั้งหมดเป็นขาสัญญาณในการส่ง และรับข้อมูลของพอร์ตอนุกรมตามลำดับ โดยที่ CLKX และ CLKR จะรับสัญญาณมาจาก Shift clock ของวงจรเชื่อมต่อสัญญาณนาฬิกา กำหนดให้สามารถทำการอินเทอร์รัพท์ได้ทั้งทางรับ และทางส่ง และกำหนดให้ข้อมูลในรีจิสเตอร์ส่งข้อมูลของพอร์ตอนุกรมเป็น 0

ค่าต่าง ๆ ที่กำหนดให้กับรีจิสเตอร์ของพอร์ตอนุกรมใน โปรแกรมตัวอย่างมีดังนี้

- รีจิสเตอร์ FSX/FX/CLKX                      ตำแหน่ง 808042h            กำหนดค่า 0x131
- รีจิสเตอร์ FSR/FR/CLKR                    ตำแหน่ง 808043h            กำหนดค่า 0x131
- รีจิสเตอร์ global control ของพอร์ตอนุกรม    ตำแหน่ง 808010h            กำหนดค่า 0x0E970300
- รีจิสเตอร์ส่งข้อมูลของพอร์ตอนุกรม        ตำแหน่ง 808048h            กำหนดค่า 0

การกำหนดค่ารีจิสเตอร์ global control ของพอร์ตอนุกรมทำให้พอร์ตอนุกรมมีลักษณะดังนี้

- กำหนดให้ FSX เป็นอินพุต
- กำหนดให้สัญญาณทั้งทางรับ และส่งแปรค่าได้
- ส่ง และรับข้อมูล 16 บิต
- กำหนดให้มีการอินเทอร์รัพท์ได้ทั้งการส่ง และรับ

5. ทำการเอาวงจรเชื่อมต่อสัญญาณอนาลอกออกจากสถานะรีเซต โดยการกำหนดค่า 6h ให้ IOF จะเป็นการทำให้อุปกรณ์เชื่อมต่อสัญญาณอนาลอกออกมาจากสถานะรีเซต

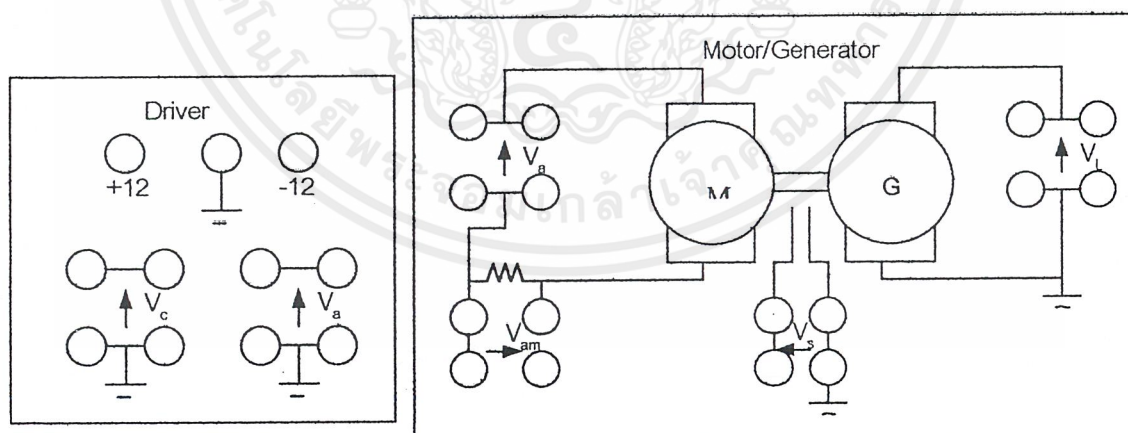
### 3 กำหนดค่าเริ่มต้นของวงจรเชื่อมต่อสัญญาณอนาลอก

การส่งข้อมูลในรีจิสเตอร์ควบคุมต่าง ๆ จะส่งแบบ secondary ซึ่งโดยปกติการส่งข้อมูลของตัวเชื่อมต่ออนาลอกจะส่งข้อมูลจริงในบิตที่ 2-15 และใช้บิตที่ 0 และ 1 ในการควบคุม ถ้าบิตควบคุมเป็น 1 ทั้ง 2 ตัวจะร้องขอทำการส่งแบบ secondary ซึ่งจะเกิดขึ้นหลังจากการส่ง shift clock 4 ลูก เมื่อเกิดการส่งแบบนี้ บิต 0 และ 1 ก็จะใช้ในการตรวจว่า ข้อมูลที่ส่งมาเป็นข้อมูลประเภทใด (รีจิสเตอร์ A, A', B หรือ รีจิสเตอร์ควบคุม) หลังจากส่งข้อมูลแบบ secondary 1 ข้อมูลแล้ว การส่งจะกลับไปส่งข้อมูลแบบปกติอีกครั้ง ดังนั้นในโปรแกรมตัวอย่าง จึงมีการวนลูการส่งข้อมูลประเภทนี้อยู่ถึง 4 ครั้ง เพื่อให้ได้การกำหนดค่าเริ่มต้นตัวเชื่อมต่ออนาลอกตามต้องการ

### 3.2 การออกแบบวงจรในการควบคุมมอเตอร์แบบป้อนกลับ

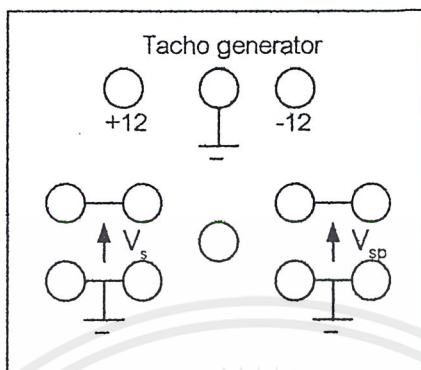
ชุดทดลองมอเตอร์ ประกอบด้วย

1. ชุดวงจรขับมอเตอร์ เป็นวงจรที่ใช้ไฟเลี้ยง  $\pm 12$  โวลต์ และใช้  $V_c$  เป็นส่วนที่รับโวลต์คำสั่งจากเอาต์พุตของวงจรตัวคูณ
2. ชุดมอเตอร์/เจนเนอเรเตอร์ ประกอบด้วยมอเตอร์มีโพลเป็นเจนเนอเรเตอร์ เมื่อมอเตอร์หมุนจะทำให้เจนเนอเรเตอร์สร้างแรงดันไฟฟ้า



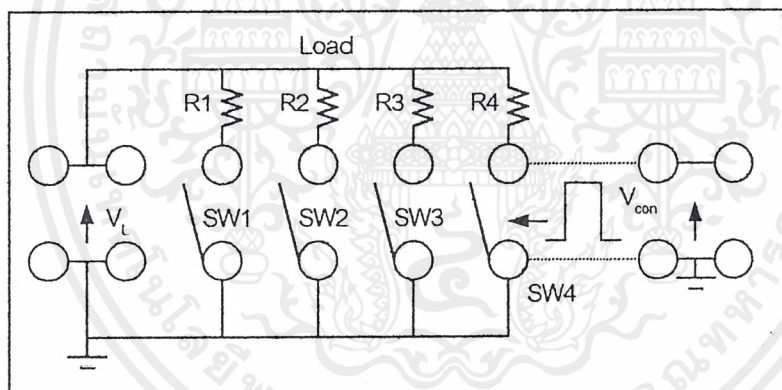
รูปที่ 3-2 แสดงชุดวงจรขับมอเตอร์ และชุดมอเตอร์/เจนเนอเรเตอร์

3. ชุดแทคโคเจเนอเรเตอร์ (Tacho Generator) ในการทดลองจะใช้  $V_{sp}$  เป็นแรงดันป้อนกลับในการป้อนกลับ เพื่อใช้ในการคำนวณค่าความผิดพลาด



รูปที่ 3-3 แสดงชุดทดลองของแทคโคเจเนอเรเตอร์

4. ชุดโหลด ประกอบด้วยชุดตัวต้านทานซึ่งเมื่อผลักสวิตช์จะทำให้โวลท์ของ  $V_L$  ลดลง ทำให้มอเตอร์หมุนช้าลงด้วย

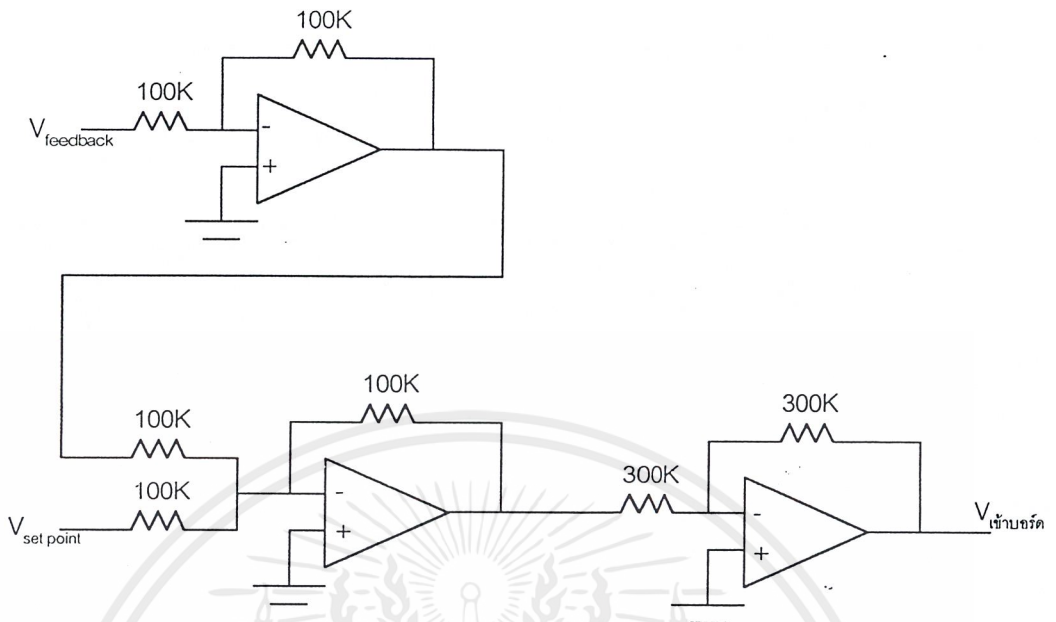


รูปที่ 3-4 แสดงชุดโหลด

วงจรที่ใช้ในการทดลอง ประกอบด้วย

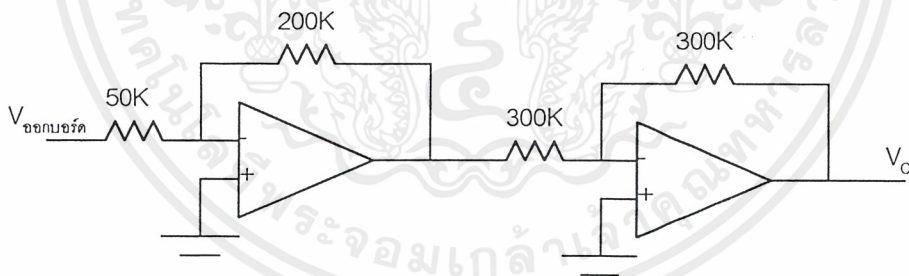
1. วงจรบวก ใช้ในการหาค่าผิดพลาดระหว่างเซ็ทพอยต์กับแรงดันป้อนกลับจากชุดแทคโคเจเนอเรเตอร์ โดยใช้อปแอมป์เบอร์ 74LS741 แต่เนื่องจากในการทดลองต้องการการป้อนกลับแบบลบ จึงใช้อินเวอร์ตติ้งแอมป์ในการกลับเฟสของแรงดันป้อนกลับ จากวงจรในรูปที่ 3-5 จะได้

$$V_{\text{เข้าบอร์ด}} = V_{\text{set point}} - V_{\text{feedback}}$$



รูปที่ 3-5 แสดงวงจรในการป้อนกลับ และวงจรบวกเพื่อหาค่าผิดพลาด  $e(k)$

2. วงจรขยาย เพื่อใช้ในการขยายสัญญาณเอาต์พุตที่ออกจากบอร์ด เนื่องจากบอร์ดให้ช่วงความดันที่น้อย จึงทำการคูณสัญญาณที่ออกมาจากบอร์ดด้วย 2 ดังรูปที่ 3-6เป็นการใช้อินเวอร์ตติ้งแอมป์ จึงต้องกลับเฟสอีกครั้ง



รูปที่ 3-6 แสดงวงจรคูณสัญญาณที่ออกจากบอร์ด

จากวงจรดังรูปที่ 3-6 จะได้ฟังก์ชันถ่ายโอนเป็น

$$V_c = (200/50) V_{\text{ออกบอร์ด}}$$

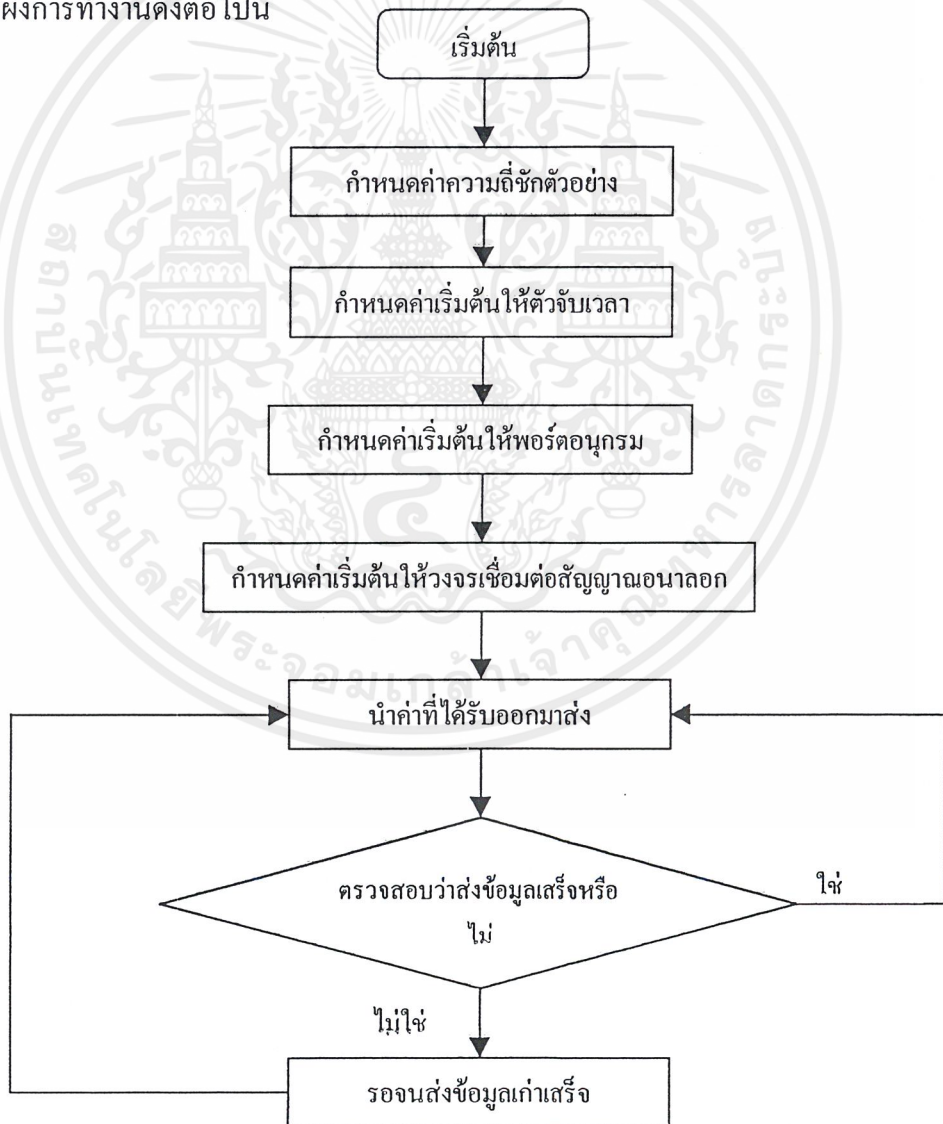
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการทดลอง

#### 4.1 ผลการทดลองการใช้วงจรเชื่อมต้อสัญญาณอนาล็อก

4.1.1 การทดลองรับค่าเข้าบอร์ดดีเอสเคและส่งค่าเดิมออกมาจากบอร์ดดีเอสเค โปรแกรมนี้จะรับค่าอินพุตที่คอนเนคเตอร์ RCA อินพุต และส่งค่าเดิมออกไปที่คอนเนคเตอร์ RCA เอาท์พุต ซึ่งสามารถที่จะใส่อินพุต เป็นสัญญาณไซน์ สี่เหลี่ยม หรือสามเหลี่ยมก็ได้ แต่ควรจะมีค่าความถี่อยู่ระหว่าง 500 Hz จนถึง 3 kHz ซึ่งเอาท์พุตออกมาจะมีค่าตามอินพุต แต่จะมีความล่าช้าของสัญญาณอยู่บ้าง มีผังการทำงานดังต่อไปนี้



รูปที่ 4-1 แสดงผังการทำงานของโปรแกรมรับค่า และส่งค่าเดิมออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----;
โปรแกรมที่ 1 เป็น โปรแกรมรับอินพุต และส่งเอาต์พุตออกเป็นค่าเดิม
;-----;

        .start      ".text",0x809900      ;จุดเริ่มต้นสำหรับ โปรแกรม
        .start      ".data",0x809C00     ;จุดเริ่มต้นสำหรับข้อมูล
        .data

AICSEC   .word      0162C,1h,4892h,63h ;Fs = 8 kHz filter
PBASE   .word      808000h
SETSP   .word      0E970300h
ATABLE  .word      AICSEC
;-----;
;-----;
;-----;
        .entry      BEGIN                ;เริ่มโปรแกรม
        .text
;-----;
;-----;
        กำหนดค่าเริ่มต้นของ โปรแกรม
;-----;
;-----;
BEGIN   LDI         @PBASE,AR0            ;AR0 -> 808000h
        LDI         1,R0                 ;กำหนดค่า timer period = 1 ทำให้
        STI         R0,*+AR0(28h)       ;TCLK0=6.25 MHz
        LDI         03C1h,R0            ;รีจิสเตอร์ timer global = 03C1h
        STI         R0,*+AR0(20h)
        LDI         62h,IOF              ;รีเซตอุปกรณ์เชื่อมต่ออนุกรม
        LDI         @ATABLE,AR1         ;AR1 -> ค่าเริ่มต้นในของอุปกรณ์เชื่อมต่อ
        RPTS        99                   ;อนุกรม
        NOP
        LDI         131h,R0              ;กำหนดรีจิสเตอร์ควบคุมการรับและส่ง
        STI         R0,*+AR0(42h)       ;FSX/DX/CLKX=เป็นขาพอร์ตอนุกรม
        STI         R0,*+AR0(43h)       ;FSR/DR/CLKR=เป็นขาพอร์ตอนุกรม

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LDI      @SETSP,R0
STI      R0,*+AR0(40h)      ;กำหนดค่าให้กับ serial global control reg
LDI      0,R0
STI      R0,*+AR0(48h)      ;เคลียร์รีจิสเตอร์ส่งข้อมูล
OR       06h,IOF            ;ทำให้อุปกรณ์เชื่อมต่ออนุกรมออกจากรีเซต
LDI      03h,RC             ;ส่งข้อมูล secondary 4 ครั้ง
RPTB    SECEND              ;จนถึง SECOND
CALL     TWAIT              ;รอนกว่าจะส่งข้อมูลใหม่ได้
LDI      03h,R0             ;ร้องขอการส่งแบบ secondary
STI      R0,*+AR0(48h)
CALL     TWAIT              ;รอนกว่าจะส่งข้อมูลใหม่ได้
LDI      *AR1++(1),R0       ;ส่งข้อมูลเพื่อกำหนดค่าเริ่มต้น
SECEND   STI      R0,*+AR0(48h)
;-----เริ่มต้น โปรแกรมหลัก-----
;รับค่าเข้ามาที่รีจิสเตอร์ R6 ส่งออกไปที่รีจิสเตอร์ R7
;-----
LOOP     CALL     TWAIT      ;รอนกว่าจะส่งค่าได้
         CALL     AICIO_I    ;ส่งค่าออกไป
         LDI      R6,R7
         BR       LOOP
         .end
;-----
;-----รอให้ส่งข้อมูลจนเสร็จ-----
;เป็นโปรแกรมที่ตรวจสอบว่าข้อมูลได้ส่งออกไปจากพอร์ตอนุกรมของอุปกรณ์เชื่อมต่อแบบ
;อนุกรมแล้วหรือไม่ เพื่อที่จะได้ส่งข้อมูลถัดไป
;-----
TWAIT   PUSH     AR0
         PUSH     R0
         LDI      @PBASE,AR0
TWI     LDI      *+AR0(40h),R0      ;R0=ข้อมูลใน serial global control reg

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AND    02h,R0    ;ตรวจสอบดูว่าบัพเฟอร์ของรีจิสเตอร์ส่ง
                    ;ว่างหรือไม่ พร้อมทั้งจะส่งข้อมูลหรือไม่
BZ     TW1       ;วนลูปจนกว่าจะพร้อม
POP    R0
POP    AR0
RETS

```

```

;-----
;-----การส่งค่าออกไปที่อุปกรณ์เชื่อมต่อข้อมูล-----
;ส่งค่าออกไป โดยนำค่าที่ส่งออกไปใส่บิต R6
;-----

```

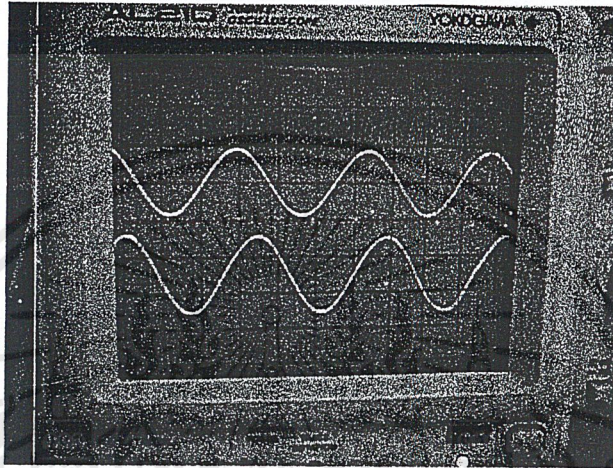
```

AICIO_I  LDI    R7,R6
          LSH    2,R6    ;สองบิตสุดท้ายต้องเป็น 0 (ส่งข้อมูลปกติ)
IO        PUSH  AR0
          LDI    @PBASE,AR0
          STI    R6,*+AR0(48h) ;ส่งข้อมูลใน R6 ออกไป
          POP    AR0
          RETS

```

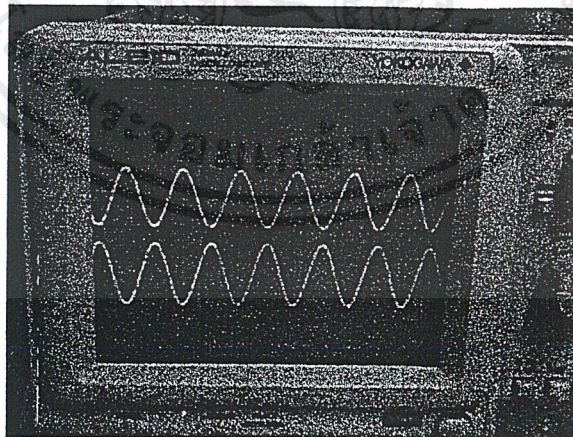
4.1.2 การทดลองผลของความถี่สัญญาณอินพุตที่มีผลต่อการล่าช้าของสัญญาณ โดยในการทดลองจะทำการรับสัญญาณขาขึ้นที่ความถี่ต่างกันผ่านทางวงจรเชื่อมต่อสัญญาณอนาล็อก แล้วส่งออกจากรุ่นนั้นหาการล่าช้าระหว่างสัญญาณที่เป็นอินพุตกับเอาต์พุต ซึ่งผลการทดลองได้ดังรูป

4.1.2.1 การทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 300 Hz ใช้ความถี่การซัดตัวอย่างเป็น 8kHz ขนาดเวลาต่อช่องเท่ากับ 1 ms สัญญาณด้านบนเป็นอินพุต สัญญาณด้านล่างเป็นเอาต์พุต



รูปที่ 4.2 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 300 Hz

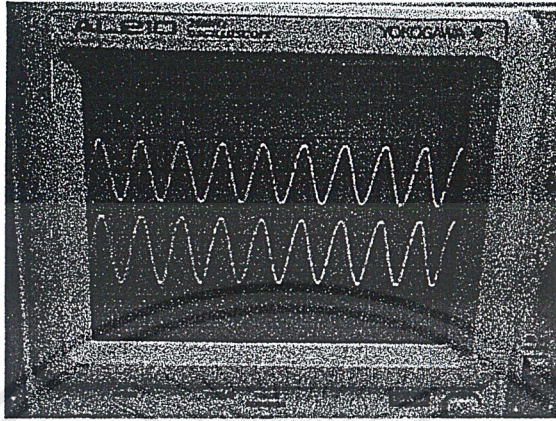
4.1.2.2 การทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 600 Hz ใช้ความถี่การซัดตัวอย่างเป็น 8kHz ขนาดเวลาต่อช่องเท่ากับ 1 ms สัญญาณด้านบนเป็นอินพุต สัญญาณด้านล่างเป็นเอาต์พุต



รูปที่ 4.3 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 600 Hz

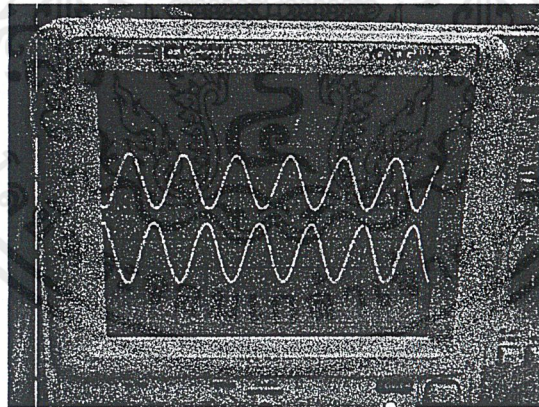
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.3 การทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 900 Hz ใช้ความถี่การซัดตัวอย่างเป็น 8kHz ขนาดเวลาต่อช่องเท่ากับ 1 ms สัญญาณด้านบนเป็นอินพุต สัญญาณด้านล่างเป็นเอาต์พุต



รูปที่ 4.4 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 900 Hz

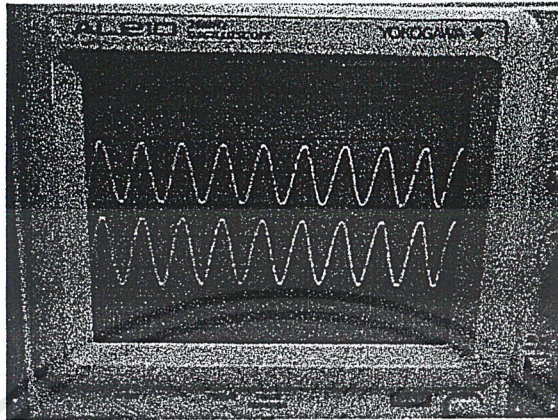
4.1.2.4 การทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 1200 Hz ใช้ความถี่การซัดตัวอย่างเป็น 8kHz ขนาดเวลาต่อช่องเท่ากับ 0.5 ms สัญญาณด้านบนเป็นอินพุต สัญญาณด้านล่างเป็นเอาต์พุต



รูปที่ 4.5 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 1200 Hz

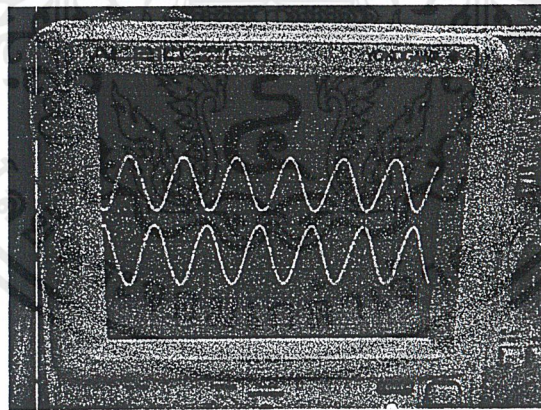
4.1.2.5 การทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 1500 Hz ใช้ความถี่การซัดตัวอย่างเป็น 8kHz ขนาดเวลาต่อช่องเท่ากับ 0.5 ms สัญญาณด้านบนเป็นอินพุต สัญญาณด้านล่างเป็นเอาต์พุต

4.1.2.3 การทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 900 Hz ใช้ความถี่การซัดตัวอย่างเป็น 8kHz ขนาดเวลาต่อช่องเท่ากับ 1 ms สัญญาณด้านบนเป็นอินพุต สัญญาณด้านล่างเป็นเอาต์พุต



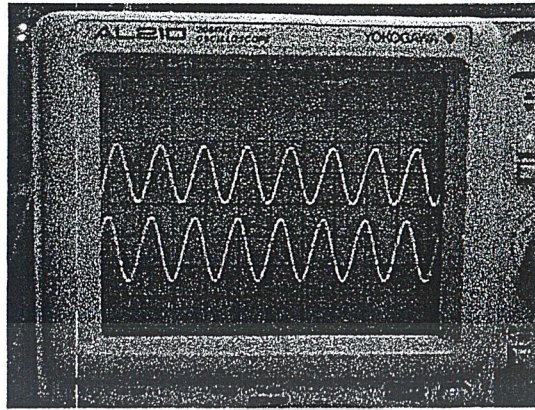
รูปที่ 4.4 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 900 Hz

4.1.2.4 การทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 1200 Hz ใช้ความถี่การซัดตัวอย่างเป็น 8kHz ขนาดเวลาต่อช่องเท่ากับ 0.5 ms สัญญาณด้านบนเป็นอินพุต สัญญาณด้านล่างเป็นเอาต์พุต



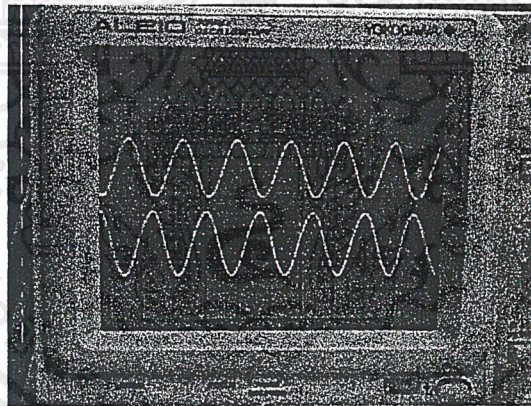
รูปที่ 4.5 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 1200 Hz

4.1.2.5 การทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 1500 Hz ใช้ความถี่การซัดตัวอย่างเป็น 8kHz ขนาดเวลาต่อช่องเท่ากับ 0.5 ms สัญญาณด้านบนเป็นอินพุต สัญญาณด้านล่างเป็นเอาต์พุต



รูปที่ 4.6 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 1500 Hz

4.1.2.6 การทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 1200 Hz ใช้ความถี่การชักตัวอย่างเป็น 20kHz ขนาดเวลาต่อช่องเท่ากับ 0.5 ms สัญญาณด้านบนเป็นอินพุต สัญญาณด้านล่างเป็นเอาต์พุต



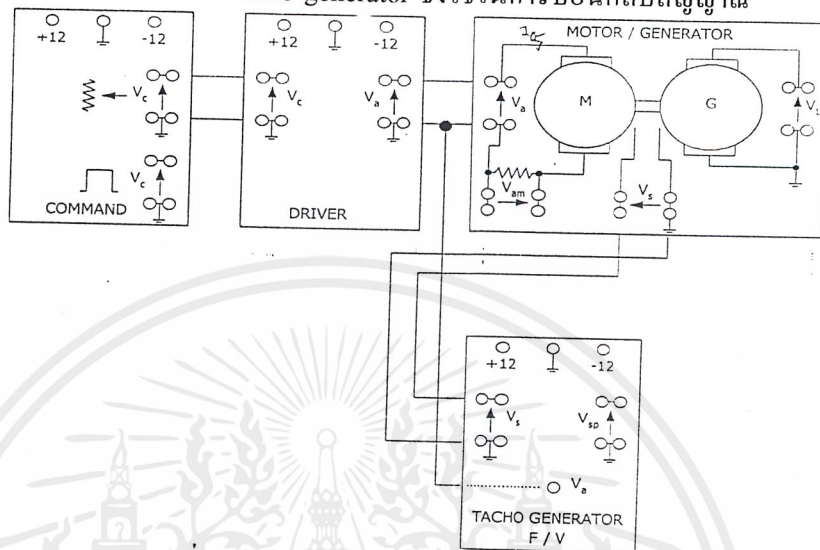
รูปที่ 4.7 แสดงผลการทดลองเมื่อความถี่ของสัญญาณอินพุตเป็น 1200 Hz

จากผลการทดลองจะเห็นได้ว่า ความถี่ของสัญญาณอินพุตมีผลต่อการล่าช้าระหว่างสัญญาณอินพุตกับสัญญาณเอาต์พุต โดยเมื่อความถี่มากขึ้นจะทำให้เกิดการล่าช้ามากขึ้นจนกระทั่งถึงค่าความถี่ค่าหนึ่งจะทำให้เกิดการล่าช้าจนครบลูก เช่น ที่ความถี่ 900 Hz เป็นต้น

จากการทดลองที่ 4.1.2.4 และ การทดลองที่ 4.1.2.6 ซึ่งเป็นการทดลองโดยใช้ความถี่ของสัญญาณอินพุตเท่ากันคือ 1200 Hz แต่ใช้ค่าความถี่ชักตัวอย่างไม่เท่ากัน จากรูปที่ 4.5 และรูปที่ 4.7 จะเห็นได้ว่าค่าความถี่ชักตัวอย่างไม่มีผลต่อการล่าช้าของสัญญาณ

### 4.2 การทดลองหาค่าพารามิเตอร์ของมอเตอร์กระแสตรง

การทดลองหาค่าพารามิเตอร์มอเตอร์ ทำโดยทดลองกับกล่องมอเตอร์ที่ประกอบด้วย วงจรขับมอเตอร์ มอเตอร์กระแสตรง และ Tacho-generator ซึ่งใช้ในการป้อนกลับสัญญาณ



รูปที่ 4-3 แสดงบล็อกทดลองมอเตอร์

#### 4.2.1 การทดลองวัดค่า $R_a$

ทำการทดลองโดยการถือค่อมอเตอร์

1. ปรับ  $V_c$  เป็น 0.5 โวลต์ วัดค่า  $V_o$  และ  $V_{am}$  บันทึกค่า  $V_o$ ,  $V_{am}$  และ  $I_{am}$  ( $I_{am} = V_{am} / 0.5$ ) และ  $R_a$  ( $R_a = V_a / I_{am}$ ) ลงในตาราง
2. ทำการทดลองตามข้อ 1 โดยเพิ่มค่าครั้งละ 0.5 V จนกระทั่งถึง 3 V
3. คำนวณค่า  $R_a$  เฉลี่ย

ตารางที่ 4-1 แสดงผลการหาค่า R

Vc (V)	Va (V)	Vam (V)	Iam (A)	Ra
0.5	0	0	0	0
1	0	0	0	0
1.5	1.28	0.04	0.08	16
2	1.77	0.04	0.08	22.125
2.5	2.28	0.04	0.08	28.5
3	2.77	0.05	0.1	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$R_a \text{ เฉลี่ย} = (\sum R_a) / \text{จำนวนครั้งที่วัด}$$

$$= 23.58 \Omega$$

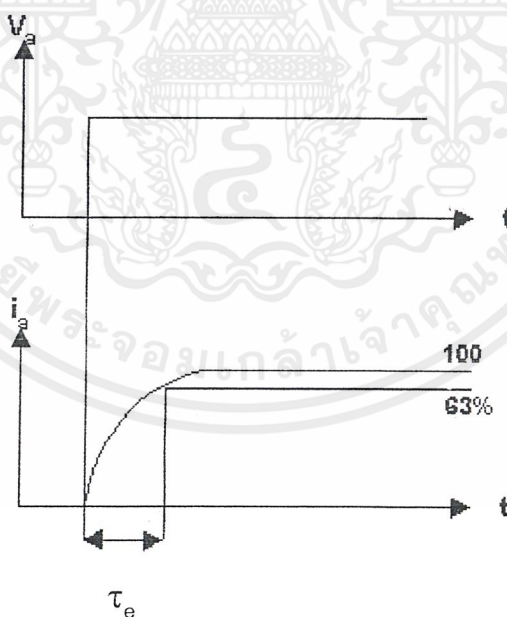
#### 4.2.2 การทดลองการวัดค่า $L_a$

จากรูปที่ 4-3 ทำให้  $e_b = 0$  โดยการล๊อคมอเตอร์

วิธีการทดลอง

1. ปรับสัญญาณ  $V_c$  เป็นชนิดพัลส์ ขนาด 0-3 โวลต์ ที่ความถี่ 50 เฮิรซ์
2. ปรับออสซิลอสโคปวัดสัญญาณ  $V_a$  และ  $I_a$  (ได้จากการวัดสัญญาณ  $V_{am}, I_{am} = V_{am}/R_{am}$ ) เปรียบเทียบกับเวลา บันทึกลงในรูปที่ 4-4
3. จากรูปสัญญาณในข้อ 2 วัดค่า  $\tau_c$
4. คำนวณค่า  $L_a = (\tau_c)(R_a)$

จากรูป 4-4 จะได้  $L_a = 0.003 \text{ H}$



รูปที่ 4-4 แสดงความสัมพันธ์ระหว่าง  $V_a, I_a$  ในการหาค่า  $\tau_c$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.2.3 การทดลองการคำนวณค่า $K\phi$

วิธีการทดลอง

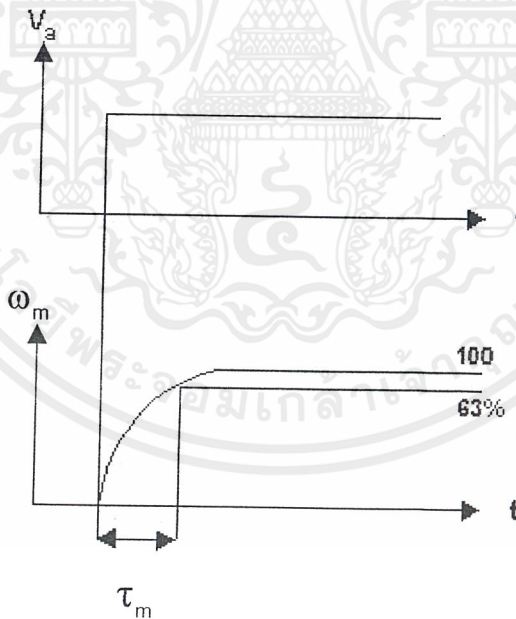
1. คำนวณค่า  $K\phi$  และบันทึกค่า  $I_{a,w}$  และ  $K\phi$  แต่ละค่าลงในตาราง
2. คำนวณค่า  $K\phi$  เฉลี่ย

จากการทดลองหาค่า  $K\phi$  เฉลี่ยได้เท่ากับ 0.01528

### 4.2.4 การทดลองการวัดค่า $\tau_m$

วิธีการทดลอง

1. ใช้วงจรตามรูปโดยไม่ต้องถือคอมเตอร์
2. ปรับสัญญาณ  $V_c$  เป็นชนิดพัลส์ ขนาด 0-3 โวลต์ ที่ความถี่ 10 เฮิรซ์
3. ใช้ออสซิลอสโคปรวัดสัญญาณ  $V_a$  และ  $V_{sp}$  กับเวลา บันทึกลงในรูปที่ 4-5
4. จากรูปสัญญาณในข้อ 3 วัดค่า  $\tau_m$



รูปที่ 4-4 แสดงความสัมพันธ์ระหว่าง  $V_a, I_a$  ในการหาค่า  $\tau_m$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**การหาฟังก์ชันถ่ายโอนของมอเตอร์กระแสตรง**

จากบล็อกของมอเตอร์ในรูปที่ 4-3 เมื่อหาค่าพารามิเตอร์ต่าง ๆ ได้ตามข้างต้นก็สามารถหาฟังก์ชันถ่ายโอนของมอเตอร์กระแสตรงที่ใช้ในการทดลองได้ตามสมการที่ 2-13 ในบทที่ 2

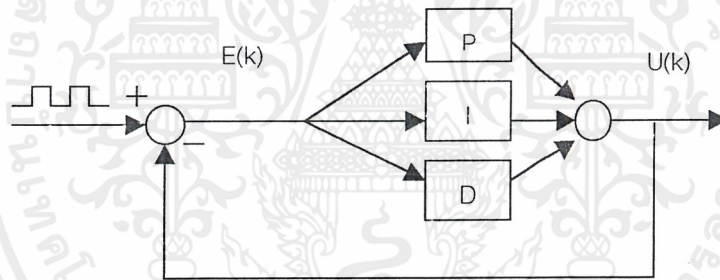
$$\frac{w(s)}{Va(s)} = \frac{1}{(s^2\tau_m\tau_e + s\tau_m + 1)k\Phi}$$

จะได้

$$\frac{w(s)}{Va(s)} = \frac{1}{3.735 \times 10^{-6} s^2 + 3.3616 \times 10^{-3} s + 0.01528} \quad (4-1)$$

**4.3 การทดลองเขียนโปรแกรมควบคุมแบบพี ไอ และดี**

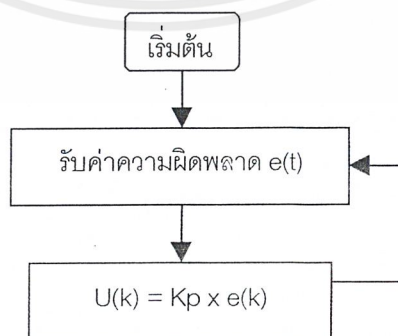
ในการทดลองการเขียนโปรแกรมพี ไอ และดี ได้ทำการทดลองเขียนกับสัญญาณสี่เหลี่ยมที่ได้มาจากอุปกรณ์กำเนิดสัญญาณ (function generator) โดยจะเทียบความถูกต้องของสัญญาณกับโปรแกรมจำลอง (MATLAB)



รูปที่ 4-5 แสดงบล็อกโคโตะแกรมในการทดลองเขียนตัวควบคุมแบบพี ไอ และดี

**4.3.1 การเขียนโปรแกรมควบคุมแบบพี**

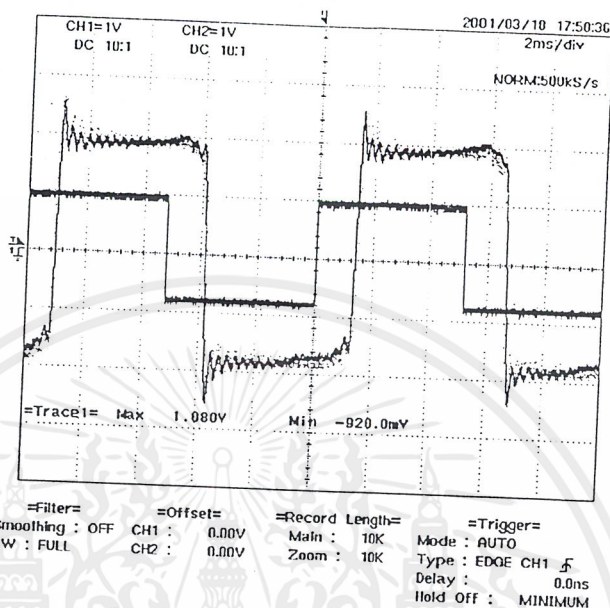
การเขียน โปรแกรมควบคุมแบบพี มีผังงานดังนี้



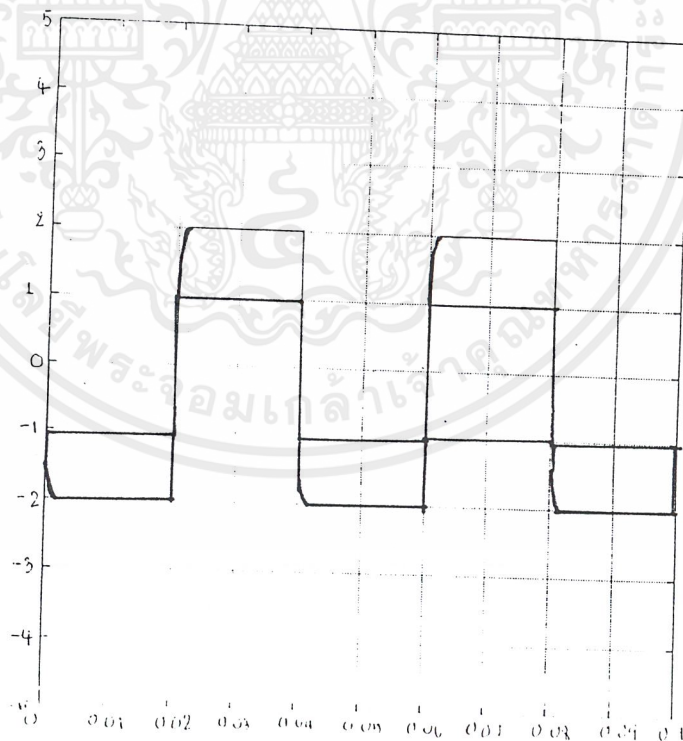
รูปที่ 4-6 แสดงผังงานในการเขียน โปรแกรมตัวควบคุมแบบพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโปรแกรมพีไอดีในภาคผนวก เมื่อกำหนดค่า  $K_p = 2$  ค่า  $K_i$  และ  $K_d = 0$  จะได้กราฟ ดังรูปที่ 4-7 เมื่อเทียบกับกราฟที่ได้มาจากการทำการจำลอง โปรแกรม MATLAB ในรูป 4-8



รูปที่ 4-7 แสดงผลของตัวควบคุมแบบพีจากการทดลองจริง เมื่อให้ค่า  $K_p$  เท่ากับ 2

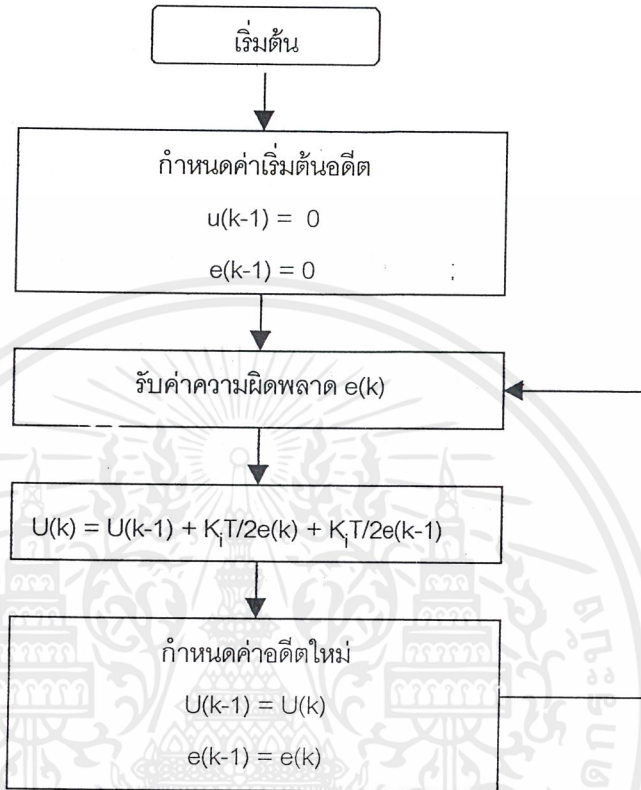


รูปที่ 4-8 แสดงผลของตัวควบคุมแบบพีจากโปรแกรม MATLAB เมื่อให้ค่า  $K_p$  เท่ากับ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

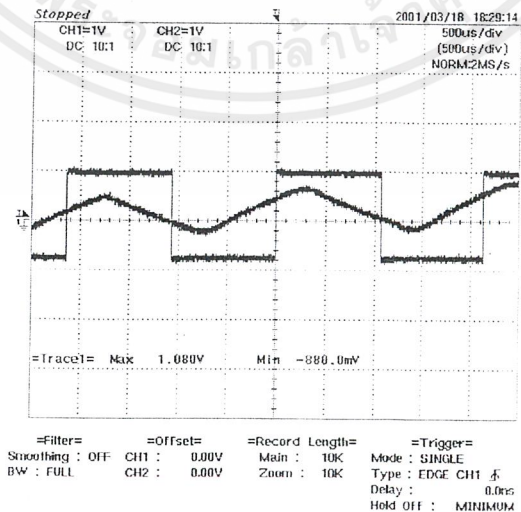
### 4.3.1 การเขียนโปรแกรมควบคุมแบบไอ

การเขียน โปรแกรมควบคุมแบบ ไอ มีผังงานดังนี้



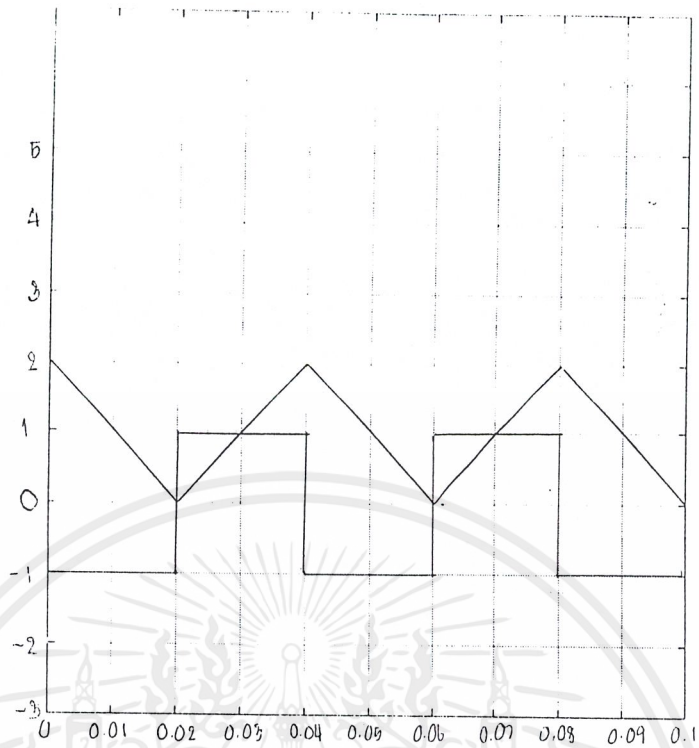
รูปที่ 4-9 แสดงผังงานในการเขียนโปรแกรมตัวควบคุมแบบไอ

จากโปรแกรมพีไอดีในภาคผนวก เมื่อกำหนดค่า  $K_i = 1000$  ค่า  $K_p$  และ  $K_d = 0$  จะได้กราฟ ดังรูปที่ 4-10 เมื่อเทียบกับกราฟที่ได้มาจากการทำการจำลองโปรแกรม MATLAB ในรูป 4-11



รูปที่ 4-10 แสดงผลของตัวควบคุมแบบไอจากการทดลองจริง เมื่อให้ค่า  $K_i$  เท่ากับ 1000

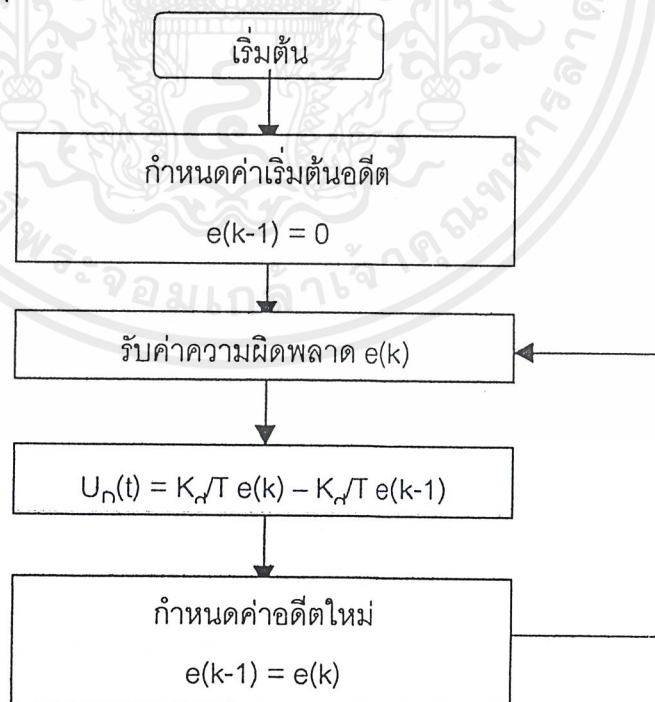
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-17 แสดงผลของตัวควบคุมแบบไอจากโปรแกรม MATLAB เมื่อให้ค่า  $K_i$  เท่ากับ 1000

#### 4.3.3 การเขียนโปรแกรมควบคุมแบบดี

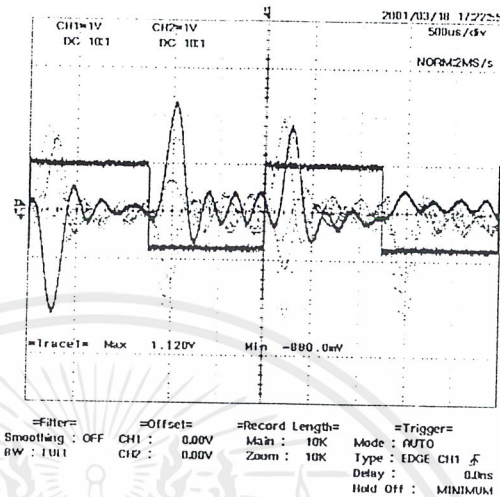
การเขียน โปรแกรมควบคุมแบบดี มีผังงานดังนี้



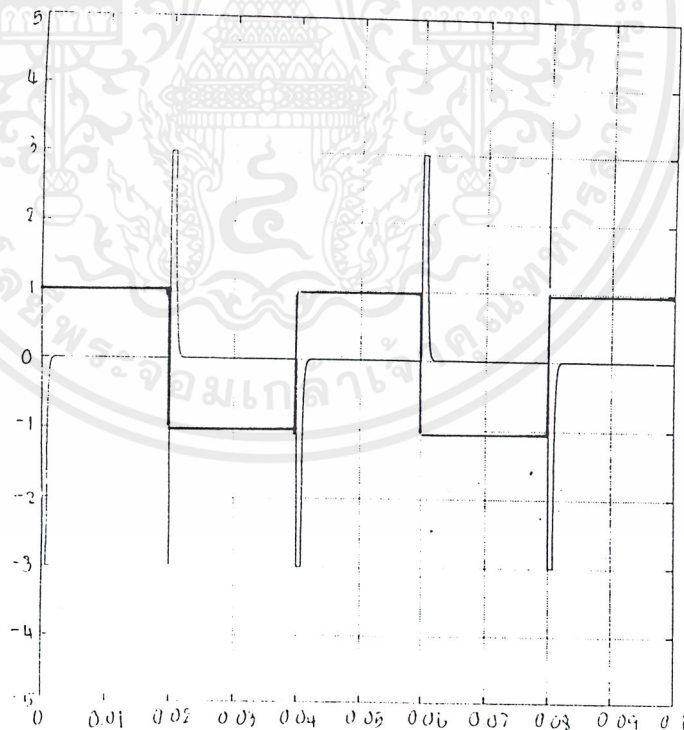
รูปที่ 4-18 แสดงผังงานในการเขียนโปรแกรมตัวควบคุมแบบดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโปรแกรมพีไอดีในภาคผนวก เมื่อกำหนดค่า  $K_d = 0.0125$  ค่า  $K_p$  และ  $K_i = 0$  จะได้กราฟ ดังรูปที่ 4-13 เมื่อเทียบกับกราฟที่ได้มาจากการทำการจำลองโปรแกรม MATLAB ในรูป 4-14



รูปที่ 4-13 แสดงผลของตัวควบคุมแบบดิจิตอลจาก โปรแกรม MATLAB เมื่อให้ค่า  $K_d$  เท่ากับ 0.0125

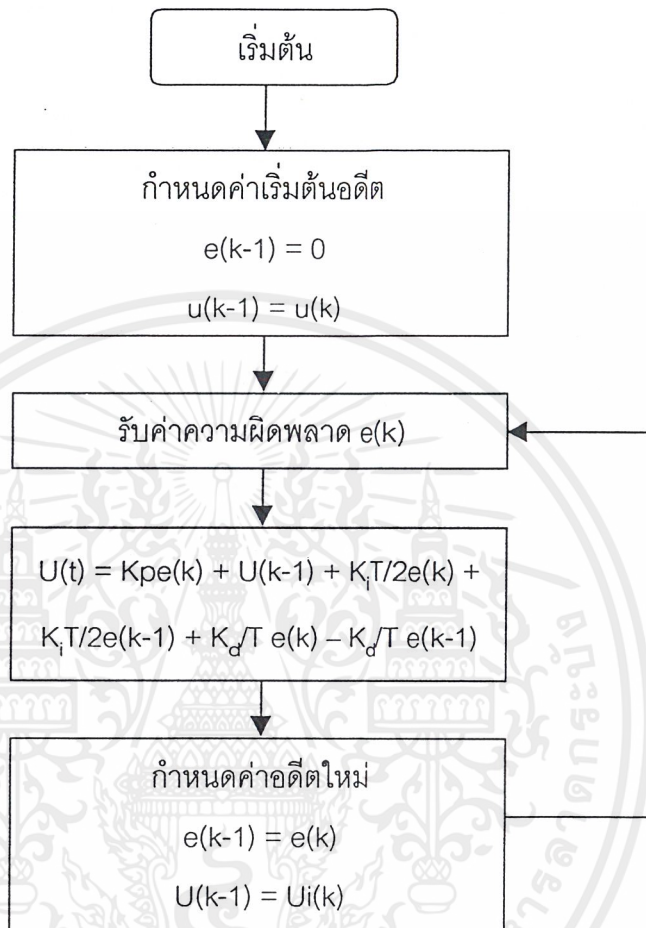


รูปที่ 4-8 แสดงผลของตัวควบคุมแบบดิจิตอลจาก โปรแกรม MATLAB เมื่อให้ค่า  $K_d$  เท่ากับ 0.0125

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

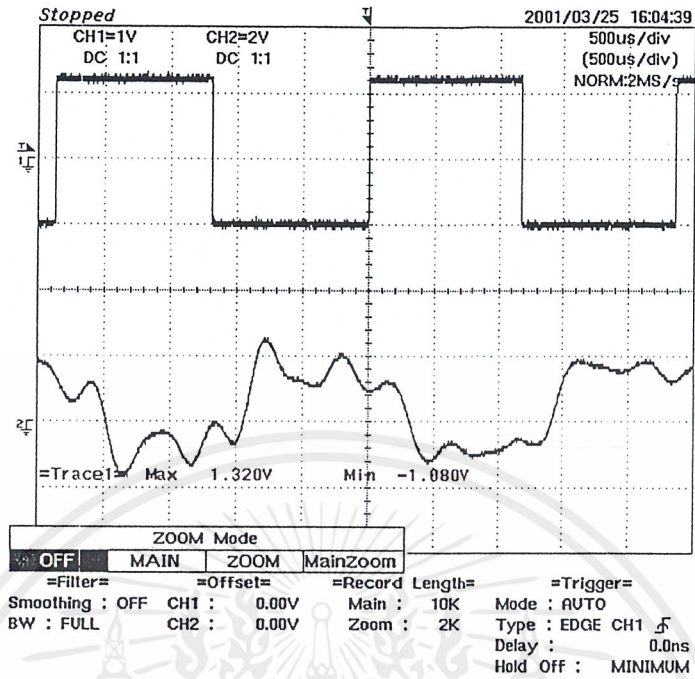
#### 4.3.4 การเขียนโปรแกรมควบคุมแบบพีไอดี

การเขียน โปรแกรมควบคุมแบบพีไอดี มีผังงานดังนี้

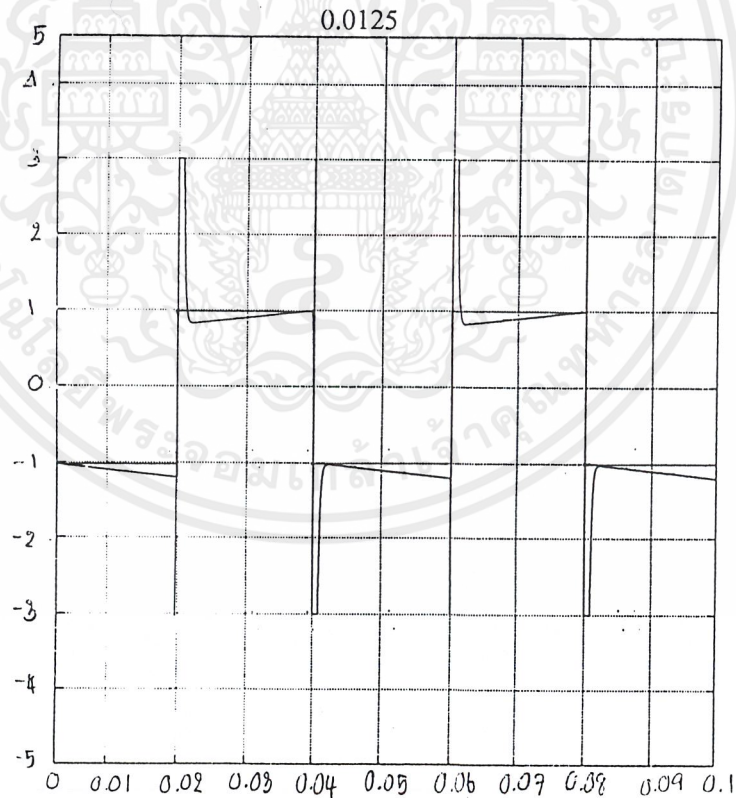


รูปที่ 4-21 แสดงผังงานในการเขียน โปรแกรมควบคุมแบบพีไอดี

จากโปรแกรมพีไอดีในภาคผนวก เมื่อกำหนดค่า  $K_i = 1$   $K_d = 0.0125$   $K_p = 10$  จะได้กราฟดังรูปที่ 4-22 เมื่อเทียบกับกราฟที่ได้มาจากการทำการจำลองโปรแกรม MATLAB ในรูป 4-23



รูปที่ 4-22 แสดงผลของการทดลองการเขียนโปรแกรมแบบพีไอดี เมื่อใช้ค่า  $K_p = 1$   $K_i = 10$   $K_d =$



รูปที่ 4-23 แสดงผลของตัวควบคุมแบบพีไอดีจากโปรแกรม MATLAB เมื่อให้ค่า  $K_p = 1$   $K_i = 10$

$$K_d = 0.0125$$

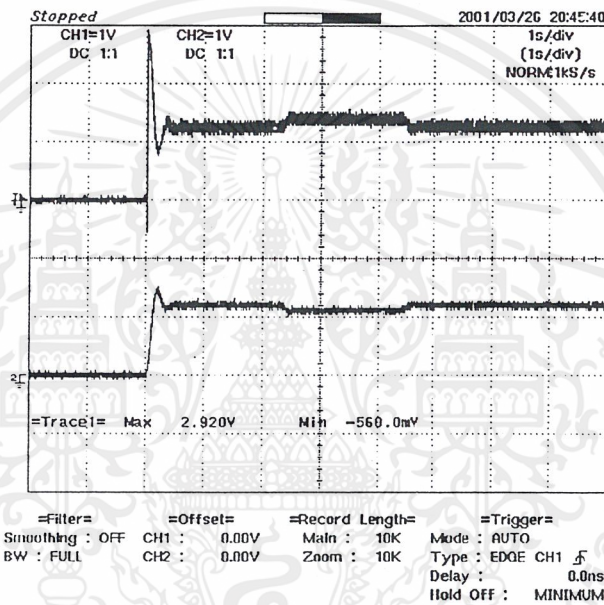
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4 ผลการทดลองในการควบคุมแบบมอเตอร์

กราฟที่ได้จากการทดลอง โดยการเขียน โปรแกรมที่แสดงในภาคผนวก รูปบนจะเป็น สัญญาณควบคุมที่ออกจากบอร์ด ส่วนรูปล่างจะเป็นสัญญาณเอาต์พุตที่ได้จากการทดลอง เมื่อแปร ค่า  $K_p$ ,  $K_i$  และ  $K_d$  ต่าง ๆ กัน และมีแรงดันเซตพอยต์เท่ากับ 2 V

##### 4.4.1 ผลการควบคุมความเร็วมอเตอร์ด้วยตัวควบคุมแบบพี

ผลการควบคุมแบบพีดังรูปที่ 4-24 ค่า  $K_p$  เท่ากับ 3 จะสังเกตจากรูป เมื่อมีการเปิดสวิตซ์ โหลด ความเร็วรอบ (อยู่ในรูปของสัญญาณแรงดัน) จะเปลี่ยนเพียงเล็กน้อย ซึ่งเป็นผลจากการใช้ตัว ควบคุมแบบพี นอกจากนี้ จากรูปจะมี Steady-state error



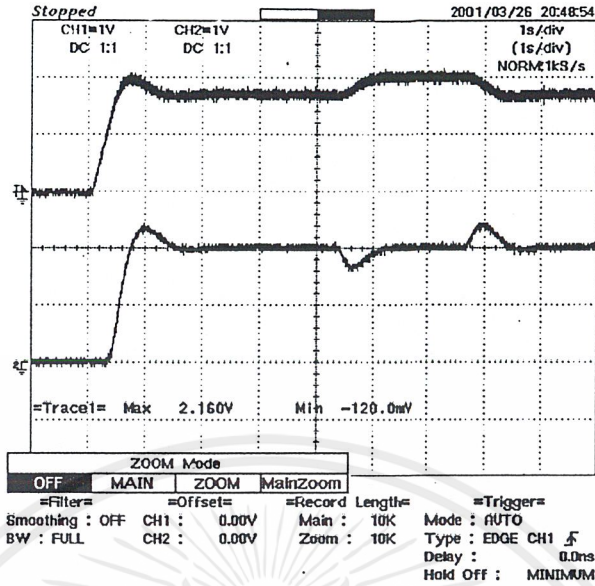
รูปที่ 4-24 แสดงผลการควบคุมความเร็วมอเตอร์ด้วยตัวควบคุมแบบพี

กราฟรูปบนแสดงสัญญาณควบคุมที่ออกจากบอร์ด

กราฟรูปล่างแสดงสัญญาณผลเอาต์พุตของมอเตอร์

##### 4.4.2 ผลของการควบคุมความเร็วมอเตอร์ด้วยตัวควบคุมแบบไอ

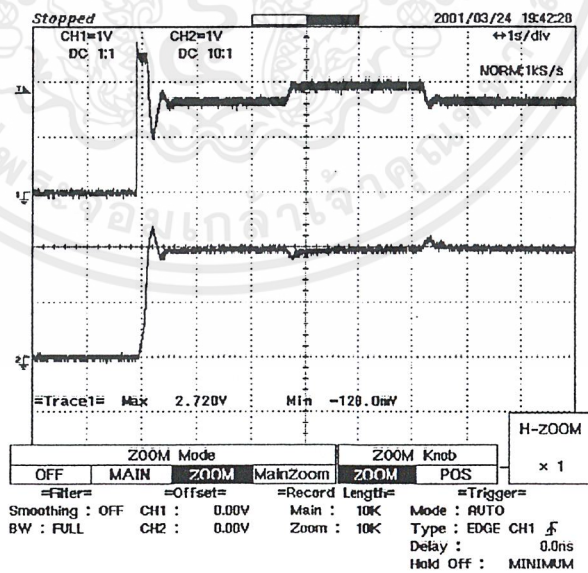
ผลการควบคุมแบบไอเป็นดังรูปที่ 4-25 ค่า  $K_i$  เท่ากับ 2 เมื่อมีการเปิดสวิตซ์โหลด ความเร็ว รอบจะเปลี่ยนเล็กน้อยเพียงช่วงเวลาสั้น ๆ แล้วจะกลับมาเป็นความเร็วรอบที่เท่าเดิม และเมื่อใช้ตัว ควบคุมแบบพี จะเห็นว่าไม่มี Steady-state error (ดังกราฟ เซตพอยต์เท่ากับ 2) ซึ่งเป็นผลจากการ ควบคุมแบบไอ นอกจากนี้ผลจะมี Overshoot



รูปที่ 4-25 แสดงผลการควบคุมความเร็วมอเตอร์ด้วยตัวควบคุมแบบไออ กราฟรูปบนแสดงสัญญาณควบคุมที่ออกจากบอร์ด กราฟรูปล่างแสดงสัญญาณผลเอาที่พุดของมอเตอร์

4.4.3 ผลของการควบคุมความเร็วมอเตอร์ด้วยตัวควบคุมแบบพีไอ

ผลการควบคุมแบบพีไอเป็นดังรูปที่ 4-26 ค่า  $K_p = 3$   $K_i = 5$  เมื่อมีการเปิดสวิตซ์โหลด ความเร็วรอบจะเปลี่ยนแปลงเล็กน้อยเพียงช่วงเวลาสั้น ๆ แล้วจะกลับมาเป็นความเร็วรอบที่เท่าเดิม และ



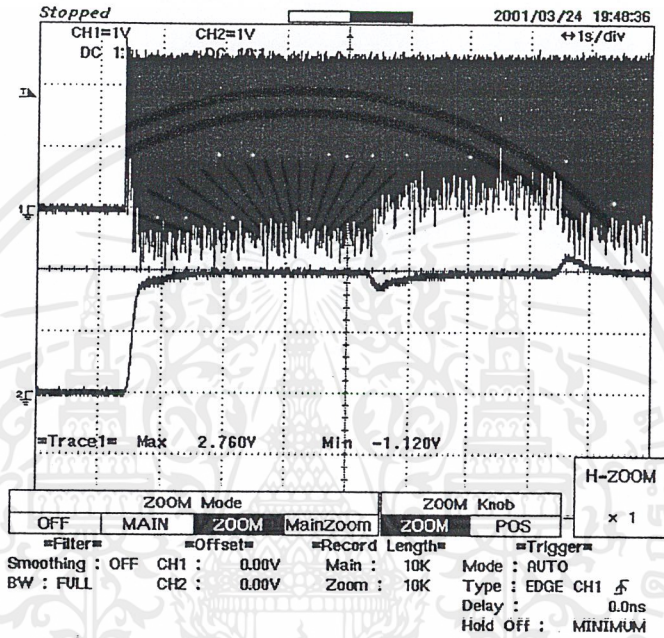
รูปที่ 4-26 แสดงผลการควบคุมความเร็วมอเตอร์เมื่อใช้ตัวควบคุมแบบพีไอ กราฟรูปบนแสดงสัญญาณควบคุมที่ออกจากบอร์ด กราฟรูปล่างแสดงสัญญาณผลเอาที่พุดของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมี rise time ลดลงเมื่อใช้ตัวควบคุมแบบพีไอ เมื่อเทียบกับตัวควบคุมแบบไอดี

#### 4.4.4 การควบคุมความเร็วมอเตอร์ด้วยตัวควบคุมแบบพีดี

ผลการควบคุมแบบพีดีเป็นดังรูปที่ 4-27 ค่า  $K_p = 3$  และ  $K_d = 0.02$  เมื่อมีการสวิตช์โหลด ความเร็วรอบจะเปลี่ยนแปลงเล็กน้อย จะสังเกตจากกราฟ จะเห็นว่า overshoot ลดลง เมื่อเทียบกับการใช้  $K_p$  อย่างเดียว ซึ่งเป็นผลจากการใช้ตัวควบคุมแบบพีดี



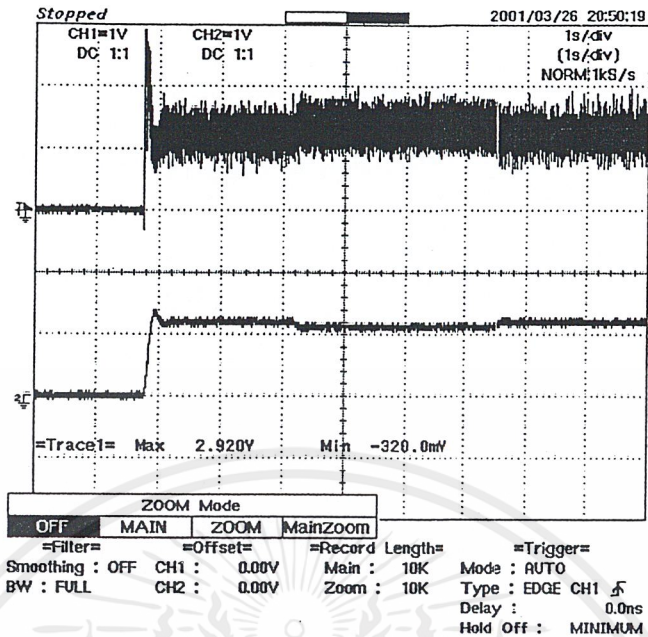
รูปที่ 4-27 แสดงผลการควบคุมความเร็วมอเตอร์แบบพีดี

กราฟรูปบนแสดงสัญญาณควบคุมที่ออกจากบอร์ด

กราฟรูปล่างแสดงสัญญาณผลเอาต์พุตของมอเตอร์

#### 4.4.5 การควบคุมความเร็วมอเตอร์ด้วยตัวควบคุมแบบพีไอดี

ผลการควบคุมแบบพีไอดีเป็นดังรูปที่ 4-28 ค่า  $K_p = 3$   $K_i = 5$   $K_d = 0.02$  เมื่อมีการเปิดสวิตช์โหลด ความเร็วรอบจะเปลี่ยนแปลงเล็กน้อยเพียงช่วงเวลานั้น ๆ แล้วจะกลับมาเป็นความเร็วรอบที่เท่าเดิม และ ผลของตัวควบคุมแบบพีไอ จะช่วยลด Overshoot ที่เกิดจากตัวควบคุมแบบไอดี



รูปที่ 4-28 แสดงผลของการควบคุมความเร็วมอเตอร์แบบพีไอดี

กราฟด้านบนแสดงสัญญาณควบคุมที่ออกจากบอร์ด

กราฟรูปล่างแสดงสัญญาณผลเอาต์พุตของมอเตอร์

ตารางที่ 4-2 แสดงความเร็วมอเตอร์จากการควบคุมแบบต่างๆ เมื่อมีการเปิดสวิตช์โหลด

ตัวควบคุม	ไม่เปิดสวิตช์	เปิดสวิตช์ 1	เปิดสวิตช์ 1,2	เปิดสวิตช์ 1,2,3
$K_p = 3$	884 รอบ/นาที	835 รอบ/นาที	814 รอบ/นาที	810 รอบ/นาที
$K_i = 2$	1440 รอบ/นาที	1440 รอบ/นาที	1443 รอบ/นาที	1442 รอบ/นาที
$K_p = 3, K_i = 5$	1441 รอบ/นาที	1439 รอบ/นาที	1437 รอบ/นาที	1439 รอบ/นาที
$K_p = 3, K_d = 0.02$	904 รอบ/นาที	848 รอบ/นาที	828 รอบ/นาที	817 รอบ/นาที
$K_p = 3, K_i = 5, K_d = 0.02$	1444 รอบ/นาที	1439 รอบ/นาที	1442 รอบ/นาที	1440 รอบ/นาที

## บทที่ 5

### บทวิจารณ์และสรุป

#### ผลการทดลอง

1. โปรแกรมที่ 1 เป็น โปรแกรมรูป เป็นการทดลองเพื่อดูการรับส่งข้อมูลผ่านทางพอร์ตอนุกรม จากการทดลองพบว่า มีการล่าช้าหลังของสัญญาณและมีปัญหาเวลาที่กับระบบที่มีการป้อนกลับ
2. การทดสอบผลของเวลาที่ผลต่อการล่าช้าหลังของสัญญาณ (Lag) จากการทดลองพบว่า เมื่อความถี่ของสัญญาณอินพุตมากขึ้น จะทำให้มีการล่าช้าหลังกันมากขึ้น นอกจากนี้ความถี่ในการซั๊กตัวอย่างยังไม่ีผลกับการล่าช้าหลังของสัญญาณ
3. การทดลองการหาค่าพารามิเตอร์ เพื่อหาฟังก์ชันถ่ายโอนของระบบมอเตอร์ จะพบว่าอาจมีการคลาดเคลื่อนเนื่องจากการอ่านค่าของความเร็วรอบจากเครื่องวัดความเร็วรอบ
4. การทดลองเขียน โปรแกรมควบคุมพีไอดี จากการทดลองพบว่าลักษณะของกราฟที่ได้จะมีแตกต่างจากที่ได้จากการจำลองใน โปรแกรมแมทแล็บ (MATLAB)บ้าง เนื่องจากแบบจำลองเป็นอุดมคติ และมีการล่าช้าหลังของสัญญาณที่ออกจากบอร์ดดีเอสเค

#### ปัญหาที่พบ

1. มีการล่าช้าหลังของสัญญาณ อันเนื่องมาจากบอร์ดที่ใช้ในการศึกษาเป็นบอร์ดที่มีการมอนิเตอร์ ทำให้โปรแกรมมีการทำงานช้ากว่าอินพุตที่เข้ามา จึงเกิดปัญหาการล่าช้าหลังขึ้น
2. เนื่องจากการรับส่งข้อมูลผ่านทางพอร์ตอนุกรมจะผ่านทางวงจรเชื่อมต่อกับสัญญาณอนาล็อก ซึ่งมีช่วงการใช้งานเพียง +3 โวลต์ ถึง -3 โวลต์ ทำให้ในการนำไปใช้งานค่อนข้างจำกัด

#### แนวทางการแก้ไขปัญหา

สำหรับปัญหาเรื่องช่วงการใช้งาน ทางผู้จัดทำได้ทำการขยายช่วงการใช้งาน โดยใช้วงจรขยายช่วย คือหลังจากที่คำนวณค่าของเอาต์พุตที่จะส่งออกมาได้แล้ว เราจะทำการหารค่าเอาต์พุตก่อนจะส่งออก เพื่อให้บอร์ดสามารถส่งออกมาได้กว้างขึ้น จากนั้นนำค่าเอาต์พุตไปผ่านวงจรขยายเพื่อคืนค่ากลับมา

ภาคผนวก

;โปรแกรมการเขียนตัวควบคุมแบบพีไอดี เพื่อนำมาควบคุมมอเตอร์

```

.start ".text",0x809A00      ;starting address for text
.start ".data",0x809B00     ;start address for data
.data                        ;data section
AICSEC .word 0A14h,1h,3E7Eh,63h ;Fs = 8 kHz
PBASE .word 808000h         ;peripheral base address
SETSP .word 0E970300h      ;serial port set-up data
ATABLE .word AICSEC        ;SP0 AIC init table address
.text                        ;text section
.entry BEGIN                ;start of code
dbase .word 809800h
ddbase .word 809900h
Kp .float 4
Ki .float 2                 ;Ki = 1000
Kd .float 0                 ;Kd = 0.0125
T .float 0.000125          ;sampling time 0.000125
F .float 8000              ;sampling frequency=1/T 8000
G .float 0.5
G1 .float 0.5
ZERO .float 0
MAX .float 8000            ;=+3V
MAXI .float 8000
MIN .float -8000          ;=-3V
MINI .float -8000
ZMAX .float 16383
.text
.entry BEGIN
BEGIN ldp AICSEC
AICSET LDI @PBASE,AR0     ;AR0 -> 808000h

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LDI      1,R0
STI      R0,*+AR0(28h)      ;timer period reg
LDI      03C1h,R0          ;init timer global register
STI      R0,*+AR0(20h)      ;reset timer
LDI      62h,IOF           ;AIC reset = 0
LDI      @ATABLE,AR1       ;AR1 -> AIC init data
RPTS     99                ;repeat next instr 100 times
NOP
LDI      131h,R0           ;X & R port control register data
STI      R0,*+AR0(42h)      ;FSX/DX/CLKX
STI      R0,*+AR0(43h)      ;FSR/DR/CLKR
LDI      @SETSP,R0
STI      R0,*+AR0(40h)      ;FSX
LDI      0,R0              ;R0 = 0
STI      R0,*+AR0(48h)      ;clear serial port XMIT register
OR       06h,IOF           ;bring AIC out of reset
LDI      03h,RC            ;RC=3 to transmit 4 values
RPTB     SECEND
CALL     TWAIT             ;wait for data transmit
LDI      03h,R0            ;value for secondary XMIT
STI      R0,*+AR0(48h)      ;secondary XMIT request to AIC
CALL     TWAIT             ;wait for data transmit
LDI      *AR1++(1),R0       ;AR1 -> next AIC init data
SECEND   STI      R0,*+AR0(48h) ;DTR = current AIC data
;-----

```

```

ldi      @dbase,AR1
ldi      @ddbbase,AR2
initial  ldf      0,R0          ;at 809801h=ui(k-1)=0
stf      R0,*+AR1(IR1)        ;at 809802h=c(k-1)=0
addi     1,IR1
cmpi     3,IR1
bn       initial

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        ldi        1,IR1
initial1  ldf        @ZERO,R0
        stf        R0,*+AR2(IR1)
        addi       1,IR1
        cmpi       4,IR1
        bn         initial1
;-----
        ;ldf        0,R7
        call       AICIO_P
PID      float    R6,R6          ;R6=e(k)
;-----
;for p->up(k)=Kp*e(k)
;-----
        ldf        @ZERO,R1
        cmpf       @Kp,R1
        bz         P0
        ldf        @Kp,R1
        mpyf3      R1,R6,R2      ;R2=Kp*e(k)
        ldf        @G1,R0
        mpyf       R0,R2          ;R2=up(k)*0.5
        b         PNZ
P0       ldf        @ZERO,R2
PNZ      stf        R2,*+AR2(1)   ;at 809901=Kp*e(k)
;-----+
;for i->ui(k)=ui(k-1)+Ki*t/2*e(k)+Ki*t/2*e(k-1)
;-----
        ldf        @ZERO,R1
        cmpf       @Ki,R1
        bz         PI0
        ldf        *+AR1(1),R1    ;R1=ui(k-1)
        ldf        @Ki,R2          ;R2=Ki

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้เพื่อการศึกษาเท่านั้น;R3=Tอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ldf      @G,R4      ;R4=0.5
mpyf     R2,R3      ;R3=Ki*T
mpyf     R3,R4      ;R4=0.5Ki*T
ldf      *+AR1(2),R5 ;R5=e(k-1)
addf3    R6,R5,R2   ;R2=e(k)+e(k-1)
mpyf     R2,R4      ;R4=(0.5Ki*T)(e(k)+e(k-1))
addf     R1,R4      ;R4=ui(k)
ldf      @MIN,R1    ;R1=8192
ch       ldf      @ZERO,R1
        cmpf     R1,R4      ;if R4<R1 then work at IN
        bn      IN
IP       ldf      @MAX,R1      ;R1=8191
        cmpf     R4,R1      ;if R1<ui(k) then ui(k)=8000
        bn      IMAX      ;else work at PINZ
        b       PINZ
IMAX     ldf      @MAXI,R4
        b       PINZ
IN       ldf      @MIN,R1
        cmpf     R1,R4
        bn      IMIN      ;else
        b       PINZ
IMIN     ldf      @MINI,R4      ;if error below limit(-8192)
        b       PINZ      ;then ui(k)=-8000
PIO      ldf      @ZERO,R4
PINZ     stf      R4,*+AR2(2)      ;at 809902=ui(k)
        ldf      @G1,R0
        mpyf     R0,R2      ;R2=up(k)*0.5
        stf      R4,*+AR1(1)      ;up(k-1) = up(k)

```

```

;-----
;for d->ud(k)=Kd/Te(k)-Kd/Te(k-1)
;-----

```

```
ldf      @ZERO,R1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    cmpf      @Kd,R1
    bz        PD0
    ldf       @Kd,R1          ;R1=Kd
    ldf       @F,R2          ;R2=1/T
    ldf       *+AR1(2),R3    ;R3=e(k-1)
    subf3     R3,R6,R4       ;R4=e(k)-e(k-1)
    mpyf      R1,R2          ;R2=Kd/T
    mpyf      R2,R4          ;R4=Kd/T[e(k)-e(k-1)]
    b        PDNZ
PD0    ldf     @ZERO,R4
PDNZ   stf     R4,*+AR2(3)    ;at 809903 =Kd/T[e(k)-e(k-1)]
;-----
    ldf       *+AR2(1),R1    ;R1=up(k)
    ldf       *+AR2(2),R2    ;R2=ui(k)
    ldf       *+AR2(3),R3    ;R3=ud(k)
    addf      R1,R2
    addf3     R2,R3,R7       ;R7=u(k)
CHECK  LDF    @ZERO,R1      ;limit input,output
        CMPF  R1,R7
        BN   ERRN
ERRP   LDF    @MAX,R1
        CMPF  R7,R1
        BN   LIMAX
        B    SEND
LIMAX  LDF    @MAXI,R7      ;limit input not to over +3V
        B    SEND
ERRN   LDF    @MIN,R1
        CMPF  R1,R7
        BN   LIMIN
        B    SEND
LIMIN  LDF    @MINI,R7     ;limit output not to below -3V
        B    SEND

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SEND      stf      R6,*+AR1(2)      ;e(k-1)=e(k)
          fix      R7,R7
          call     AICIO_P
          b        PID
;-----
TWAIT     PUSH     AR0
          PUSH     R0
          LDI      @PBASE,AR0
TW1       LDI      *+AR0(40h),R0
          AND      02h,R0
          BZ       TW1
          POP      R0
          POP      AR0
          RETS
;-----
AICIO_I   LDI      R7,R6      ;copy output to modify for AIC
          LSH      2,R6
IO        PUSH     AR0      ;save AR0
          LDI      @PBASE,AR0 ;AR0 -> 0808000h
          STI      R6,*+AR0(48h) ;DTR = next data for AIC D/A
          LDI      *+AR0(4Ch),R6 ;R6 = DRR data from AIC A/D
          LSH      16,R6     ;left shift for sign extension
          ASH      -18,R6    ;right shift keeping sign
          POP      AR0      ;restore AR0
          RETS           ;return from subroutine
;-----
AICIO_P   CALL     TWAIT     ;wait for data to be transferred
          CALL     AICIO_I   ;call AIC transfer routine

```

## กิตติกรรมประกาศ

ขอกราบขอบพระคุณ

- อาจารย์สว่าง เดิศดิรสุนทร ที่รับเป็นอาจารย์ที่ปรึกษาและคอยให้คำแนะนำในการทำงาน
- อาจารย์ถาวร เบญจนราษฎร์ ที่ให้คำแนะนำและข้อคิดในการเขียน โปรแกรมพีไอดี
- อาจารย์นนทวัฒน์ จุลเดชะ ที่ให้ความกรุณาชี้แนวทางในการหาค่าพารามิเตอร์ของมอเตอร์
- อาจารย์ชนินทร์ บุญดิษฐานุสรณ์ และขอขอบคุณคุณ
- คุณวุฒิพงษ์ แห่งเนคเทค ที่ให้คำแนะนำเกี่ยวกับบอร์ดมากขึ้น

นอกจากนี้ขอขอบคุณบิดามารดาที่เป็นกำลังใจให้เสมอมา ขอขอบคุณเพื่อนๆทุกคนที่ช่วยเหลือไม่ว่าจะเป็นกิจ ที่อนุเคราะห์ที่ทำงานและที่พิมพ์งาน, บอมบ์ที่ช่วยดูแลเรื่องวงจรอิเล็กทรอนิกส์ และ พี่สไตร์

คณะผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## หนังสืออ้างอิง

1. Texas Instrument, "TMS320c3x s users guide " ,Printing Company ,Oct 1994
2. Texas Instrument, "TMS320C3X DSP Starter Kit User's Guide", Printing Company ,Oct 1994
3. Texas Instrument, "TLC32040,TLC32041C,TLC32041I Analog Interface Circuit", Printing Company ,1995
4. Rulph Chassaing, "Digital Signal Processing Laboratory Experiment Using C and TMS320C31 DSK",
5. Katsuhiko Ogata, "Modern Control Engineering", second edition,Prentice-Hall Inc.,1990
6. รศ.วิพันธ์ ปรีชาพานิชย์, "การวิเคราะห์ระบบควบคุมเวลาติดสคริต", พิมพ์ครั้งที่ 1, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2537

