สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมเครื่องใช้ไฟฟ้าด้วยแรบบิทไมโครโปรเซสเซอร์ผ่านเครือข่ายอินเตอร์เน็ต

Electric Appliance Controller by Using Rabbit Microprocessor via Internet

T062179

โดย

นายพงษ์พันธ์ บัวชุม

นายสิปปวิชญ์ ชุมทอง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....

เลขทะเบียน 62119

วัน,เดือน,ปี 31 ก.ค. 2549

ข.....

ง.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRIC APPLIANCE CONTROLLER BY USING RABBIT MICROPROCESSOR VIA INTERNET

BY

MR. PONGPUN BUACHUM

MR. SIPPAWIT CHUMTHONG

A THESIS SUBMITED IN PARTIAL FULFILLMENT OF

THE REQUIREMENT FOR THE DEGREE OF

BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2004

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ELECTRIC APPLIANCE CONTROLLER BY USING RABBIT MICROPROCESSOR VIA INTERNET
ชื่อนักศึกษา	นาย พงษ์พันธ์ บัวชุม รหัสนักศึกษา 45015804
	นาย สิปปวิชญ์ ชุมทอง รหัสนักศึกษา 45015824
อาจารย์ที่ปรึกษา	อาจารย์พิชญ สุพรรณกูล
อาจารย์ที่ปรึกษาร่วม	ผศ.มนต์ชัย แช่มช้อย
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต
	สาขาวิชา วิศวกรรมสารสนเทศ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2547

ปริญญานี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเรียบร้อยแล้ว

.....

(อาจารย์พิชญ สุพรรณกูล)

อาจารย์ที่ปรึกษา

.....

(ผศ.มนต์ชัย แช่มช้อย)

อาจารย์ที่ปรึกษาร่วม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	การควบคุมเครื่องใช้ไฟฟ้าด้วยแรบบิทไมโครโปรเซสเซอร์ผ่านเครือข่ายอินเตอร์เน็ต
ชื่อนักศึกษา	นายพงษ์พันธ์ บัวชุม รหัสนักศึกษา 45015804
	นายสิปปวิชญ์ ชุมทอง รหัสนักศึกษา 45015824
อาจารย์ที่ปรึกษา	อาจารย์พิชญ สุพรรณกูล
อาจารย์ที่ปรึกษาร่วม	ผศ.มนต์ชัย แช่มช้อย
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต
	สาขาวิชา วิศวกรรมสารสนเทศ
ภาควิชา	วิศวกรรมสารสนเทศ
ปีการศึกษา	2547

บทคัดย่อ

ด้วยเทคโนโลยีที่ทันสมัยในปัจจุบัน ทำให้เราสามารถควบคุมเครื่องใช้ไฟฟ้าภายในบ้านผ่านระบบเครือข่ายอินเตอร์เน็ตได้ แต่ภายในบ้านจะต้องอาศัยเครื่องคอมพิวเตอร์เป็นเซิร์ฟเวอร์สำหรับการควบคุม โครงงานนี้จึงปรับปรุงระบบการควบคุม โดยการประยุกต์ใช้งานไมโครโปรเซสเซอร์ที่มีชื่อว่า "แรบบิท" ซึ่งเป็นไมโครโปรเซสเซอร์ที่สามารถเป็นเซิร์ฟเวอร์ได้ด้วยตัวเองและมีการแจ้งสถานะการทำงานให้ผู้ใช้ได้รับทราบ จึงทำให้สามารถตรวจสอบสถานะเครื่องใช้ ไฟฟ้าได้ โครงงานนี้จึงนำเสนอระบบควบคุมเครื่องใช้ไฟฟ้าผ่านระบบเครือข่ายอินเตอร์เน็ตและควบคุมผ่านชุดทดลอง Rabbit โดยใช้ภาษา Dynamic C เป็นภาษาในการพัฒนาโปรแกรมการควบคุม ซึ่งจะทำให้ลดงบประมาณในการใช้ชุดทดลอง Rabbit มาควบคุมแทนคอมพิวเตอร์

โครงงานนี้สามารถควบคุมแสงสว่างภายในบริเวณบ้านในแต่ละจุดได้ และสามารถควบคุมการทำงานของเครื่องใช้ไฟฟ้าอื่นๆได้ เช่น โทรทัศน์ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Electric Appliance Controller by Using Rabbit Microprocessor via Internet		
Student	Mr. Pongpun Buachum	No.	45015804
	Mr. Sippawit Chumthong	No.	45015824

Advisor	Mr.Pichaya Supanakoon
Co-Advisor	Asst.Prof.Monchai Chamchoy
Graduate Level	Bachelor Degree of information Engineering
Department	Information Engineering
Academic year	2004

ABSTRACT

We can control the electric home appliance function by internet network as the advanced technology at present. The computer works as server for controlling the function in home. This project was improved control system by micro-processor application of Rabbit which is micro-processor and server in the same time. Moreover the user can know the working situation because of Rabbit. In this way, the user can check the electric home appliance situation for all the time. This project presented as electric home appliance system that uses the development program language "Dynamic-C" for controlling Rabbit and links to internet network too. That can decrease the budget of Rabbit by using for appliance function control instead of computer. Then we can also control the lights in each point and the other electric home appliance function such as TV.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่อง "การควบคุมเครื่องใช้ไฟฟ้าด้วยแรบบิทไมโครโปรเซสเซอร์ผ่านเครือข่ายอินเตอร์เน็ต (Electric Appliance Controller by Using Rabbit Microprocessor Via Internet)" เป็นโครงงานที่จัดทำขึ้นโดยได้รับการสนับสนุนจาก อาจารย์พิชญ สุพรรณกูล อาจารย์ที่ปรึกษาและผศ.มนต์ชัย แช่มช้อย อาจารย์ที่ปรึกษาร่วม ซึ่งได้ให้คำปรึกษา ข้อชี้แนะ และความช่วยเหลือในหลายสิ่งหลายอย่างจนกระทั่งลุล่วงไปได้ด้วยดี ขอกราบขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้

นายพงษ์พันธ์ บัวชุม
นายสิปปวิชญ์ ชุมทอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูปภาพ	ฉ
สารบัญตาราง	ช
บทที่ 1 บทนำ	1
1.1 แนวความคิดและที่มาของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตงาน	1
1.4 ผลที่คาดว่าจะได้รับ	2
บทที่ 2 ชุดพัฒนา RCM2200 (Rabbit Core Module 2200)	3
2.1 ลักษณะทั่วไปของ RCM2200	3
2.2 Ports	8
2.2.1 พอร์ทอนุกรม (Serial Ports)	8
2.2.2 พอร์ทขนาน (Parallel Ports)	16
2.2.3 พอร์ทลูก (Slave Port)	24
2.3 Processor register	26
2.4 Memory Mapping and Interface	28
2.4.1 Memory Mapping Unit	31
2.4.2 Memory Interface Unit	33
2.4.3 Memory Bank Control Register	33
2.4.4 Allocation of Extended Code and Data	35
2.5 I/O Bank Control Register	35
2.6 Timers	37
2.6.1 Timer A	39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.6.2 Timer B	41
บทที่ 3 ระบบเครือข่าย และ ความปลอดภัย	44
3.1 ประเภทของระบบเครือข่าย	44
3.2 องค์ประกอบของระบบเครือข่าย	45
3.3 แบบของโปรโตคอล	45
3.4 LAN กับ Ethernet	46
3.5 TCP/IP กับเครือข่ายอินเตอร์เน็ต	46
3.6 IP Address	48
3.7 Domain Name System (DNS)	49
3.8 SLIP และ PPP	50
3.9 จุดเด่นของโพรโตคอล TCP/IP	51
3.10 จุดอ่อนของ IP	51
บทที่ 4 การออกแบบระบบ	53
บทที่ 5 การทดลอง	58
บทที่ 6 สรุปและวิจารณ์ผลการทดลอง	60
บรรณานุกรม	62
ภาคผนวก ก. Package Schematic and Pinout	63
ภาคผนวก ข. Rabbit Pin Descriptions	64
ภาคผนวก ค. Rabbit Internal I/O Registers	67
ภาคผนวก ง. วงจรควบคุม	72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 ลักษณะของ Rabbit Core Module ของชุดพัฒนา RCM2200	3
รูปที่ 2.2 บล็อกไดอะแกรมของ RCM2200	4
รูปที่ 2.3 ลักษณะพอร์ตใช้งานของ RCM 2200	5
รูปที่ 2.4 RCM2200 I/O Pinouts	5
รูปที่ 2.5 Block Diagram of Rabbit Serial Ports	8
รูปที่ 2.6 Functional Block Diagram of a Serial Port	9
รูปที่ 2.7 Generation of Serial Port Interrupts	11
รูปที่ 2.8 Serial Port Synchronization	13
รูปที่ 2.9 Full-Duplex Clocked Serial Timing Diagram with Internal Clock	15
รูปที่ 2.10 Synchronous Serial Data Transmit Timing with External Clock	16
รูปที่ 2.11 Synchronous Serial Data Receive Timing with External Clock	16
รูปที่ 2.12 Parallel Port D Block Diagram	19
รูปที่ 2.13 Parallel Port E Block Diagram	22
รูปที่ 2.14 เส้นทางข้อมูลของพอร์ทลูก	24
รูปที่ 2.15 Slave port	25
รูปที่ 2.16 Rabbit รีจิสเตอร์	27
รูปที่ 2.17 ส่วนประกอบการจัดตำแหน่งของหน่วยความจำ	28
รูปที่ 2.18 ตัวอย่างการทำงานของ Memory Mapping	29
รูปที่ 2.19 memory interface unit	30
รูปที่ 2.20 Overview of Rabbit Memory Mapping	31
รูปที่ 2.21 Memory Segments	32
รูปที่ 2.22 Typical Memory Mapping and Memory Usage	35
รูปที่ 2.23 External I/O Bus Cycles	36
รูปที่ 2.24 Block Diagram of Timers A and B	38
รูปที่ 2.25 Reload Register Operation	39
รูปที่ 3.1 การต่อเน็ตเวิร์คแบบ chain	46
รูปที่ 3.2 TCP/IP ตัดข้อมูลออกเป็น packet	47
รูปที่ 3.3 การรับส่งข้อมูลผ่าน gateway	48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

		หน้า
รูปที่ 4.1 ส่วนที่เชื่อมต่อกับอุปกรณ์ไฟฟ้ากับ rabbit		53
รูปที่ 4.2 การเชื่อมต่อระหว่าง Rabbit กับ Work Station		54
รูปที่ 4.3 Flowchart การทำงาน		55
รูปที่ 4.3 การเข้าใช้งานระบบควบคุมอุปกรณ์ไฟฟ้า		55
รูปที่ 4.4 หน้า web page ของการควบคุมเครื่องใช้ไฟฟ้า		56
รูปที่ 4.5 วงจรควบคุม		57
รูปที่ 5.1 การป้อนรหัสเพื่อเข้าใช้งาน		58
รูปที่ 5.2 หน้า web page ของการควบคุมเครื่องใช้ไฟฟ้า		58
รูปที่ 5.3 การควบคุมเครื่องใช้ไฟฟ้าและการแจ้งสถานะการทำงาน		59
ภาคผนวก	รูปที่ 1 Package Outline and Pin Assignments	63
	รูปที่ 2 ลายทองแดงด้านหน้า	72
	รูปที่ 3 ลายทองแดงด้านหลัง	72
	รูปที่ 4 ตำแหน่งการวางอุปกรณ์	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 หน้าที่ของขาต่างๆของ RCM2200	6
ตารางที่ 2.2 Serial Port Registers	10
ตารางที่ 2.3 Serial Port Status Registers	10
ตารางที่ 2.4 Serial Port Control Registers	10
ตารางที่ 2.5 Peripheral Device Address and Interrupt Vectors	12
ตารางที่ 2.6 Synchronous Serial Port Signals	13
ตารางที่ 2.7 Parallel Port B Data Register PBDR	17
ตารางที่ 2.8 Parallel Port C Data Register and Function Register	18
ตารางที่ 2.9 Parallel Port D Registers	20
ตารางที่ 2.10 Parallel Port D Control Register	21
ตารางที่ 2.11 Parallel Port E Registers	22
ตารางที่ 2.12 Parallel Port E Control Register	23
ตารางที่ 2.13 Segment Registers	32
ตารางที่ 2.14 Segment Size Register	33
ตารางที่ 2.15 Memory Bank Control Register x	33
ตารางที่ 2.16 MMU Instruction/Data Register	34
ตารางที่ 2.17 I/O Bank Control Reg	36
ตารางที่ 2.18 External I/O Register Address Range and Pin Mapping	37
ตารางที่ 2.19 Timer A I/O Registers	40
ตารางที่ 2.20 Timer A Control and Status Register	41
ตารางที่ 2.21 Timer A Control Register	41
ตารางที่ 2.22 Timer B Registers	42
ตารางที่ 2.23 Timer B Control and Status Register	42
ตารางที่ 2.24 Timer B Control Register (TBCR)	43
ตารางที่ 2.25 Timer B MSB x Register	43
ภาคผนวก ตารางที่ 1 Rabbit Pin Descriptions	64
ตารางที่ 2 Rabbit Internal I/O Registers	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

อินเตอร์เน็ตเป็นสิ่งแพร่หลายในยุคปัจจุบัน การบริการต่างๆก็จะให้บริการทางอินเตอร์เน็ตกันมากขึ้น เช่น การโอนเงิน จ่ายค่าโทรศัพท์ ซื้อของ ฯลฯ เมื่อก่อนต้องไปโอนเงินถึงที่ธนาคารจึงจะโอนเงินได้ แต่ในปัจจุบัน คุณสามารถโอนเงินได้โดยการเข้าสู่ระบบอินเตอร์เน็ต โดยเข้ายังเว็บไซต์ของธนาคารก็สามารถโอนเงินได้ ไม่ต้องรอคิวให้เสียเวลา และอินเตอร์เน็ตในปัจจุบันบรรจุไปด้วยเทคโนโลยีสมัยใหม่อีกมากมาย อาจพูดได้ว่า "เราจะทำอะไรก็สามารถทำได้โดยทางอินเตอร์เน็ต"

1.1 แนวความคิดและที่มาของปัญหา

เนื่องจากปัจจุบันมีการรณรงค์การประหยัดพลังงานไฟฟ้า และเทคโนโลยีก็ทันสมัยมากขึ้นแล้วเราก็มักจะลืมปิดเครื่องใช้ไฟฟ้าตอนออกไปข้างนอกบ้าน ทำให้สิ้นเปลืองพลังงานไฟฟ้าโดยใช่เหตุ จึงมีความคิดที่จะควบคุมการทำงานเครื่องใช้ไฟฟ้าแต่ละอย่างที่สามารถควบคุมได้และที่มักจะเปิดลืมทิ้งไว้ เช่น ระบบแสงสว่างภายในบ้าน โทรทัศน์ พัดลม ฯลฯ และทราบว่าชุดพัฒนา Rabbit สามารถต่อเข้ากับระบบเน็ตเวิร์คได้ จึงใช้ชุดทดลองนี้เป็นเซิร์ฟเวอร์สำหรับการควบคุมการทำงานของเครื่องใช้ไฟฟ้าแทนการใช้คอมพิวเตอร์เป็นเซิร์ฟเวอร์ ซึ่งจะช่วยลดงบประมาณลงได้

1.2 วัตถุประสงค์

1. เพื่อลดงบประมาณในการใช้ชุดทดลอง Rabbit มาควบคุมแทนคอมพิวเตอร์
2. เพื่อความสะดวกในการตรวจสอบการทำงานของเครื่องใช้ไฟฟ้า
3. เพื่อนำเทคโนโลยีมาประยุกต์ใช้งานให้เข้ากับยุคปัจจุบัน
4. เพื่อเป็นแนวทางที่จะนำ Rabbit ไปประยุกต์ใช้ในงานอื่นต่อไป

1.3 ขอบเขตของโครงงาน

1. มีระบบควบคุมการเปิด-ปิดแสงสว่างภายในบริเวณบ้าน
2. มีระบบควบคุมการเปิด-ปิดเครื่องใช้ไฟฟ้าอื่นเช่น พัดลมโทรทัศน์ ฯลฯ
3. มีการแจ้งสถานะการทำงานผ่านหน้าจอคอมพิวเตอร์ที่ควบคุมการทำงาน
4. มีการควบคุมผ่านระบบอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ผลที่คาดว่าจะได้รับ

1. ลดงบประมาณโดยนำชุดทดลอง Rabbit มาใช้ควบคุมแทนคอมพิวเตอร์
2. มีความสะดวกในการตรวจสอบการทำงานของเครื่องใช้ไฟฟ้า
3. นำเทคโนโลยีมาประยุกต์ใช้งานได้จริง
4. เพื่อการวางแผนของธุรกิจเกี่ยวกับอินเตอร์เน็ตในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ชุดพัฒนา RCM2200 (Rabbit Core Module 2200)

Rabbit ถูกออกแบบโดยบริษัท Z-World ซึ่งเป็นบริษัทผู้ผลิต single board คอมพิวเตอร์ราคาต่ำมาเป็นเวลายาวนาน ผลิตภัณฑ์ของ Z-World ถูกสนับสนุนโดยระบบการพัฒนาภาษา C (Dynamic C)

Rabbit ใช้งานง่าย มีบัสขนาด 8 บิต เช่นเดียวกับ Z80 และสามารถใช้ชุดคำสั่งของ Z80 กับ Rabbit ได้ Rabbit สามารถต่อเป็นเซิร์ฟเวอร์ได้ด้วยตัวมันเอง โดยไม่ต้องใช้ร่วมกับเครื่องคอมพิวเตอร์

2.1 ลักษณะทั่วไปของ RCM2200

รูปที่ 2.1 ลักษณะของ Rabbit Core Module ของชุดพัฒนา RCM2200

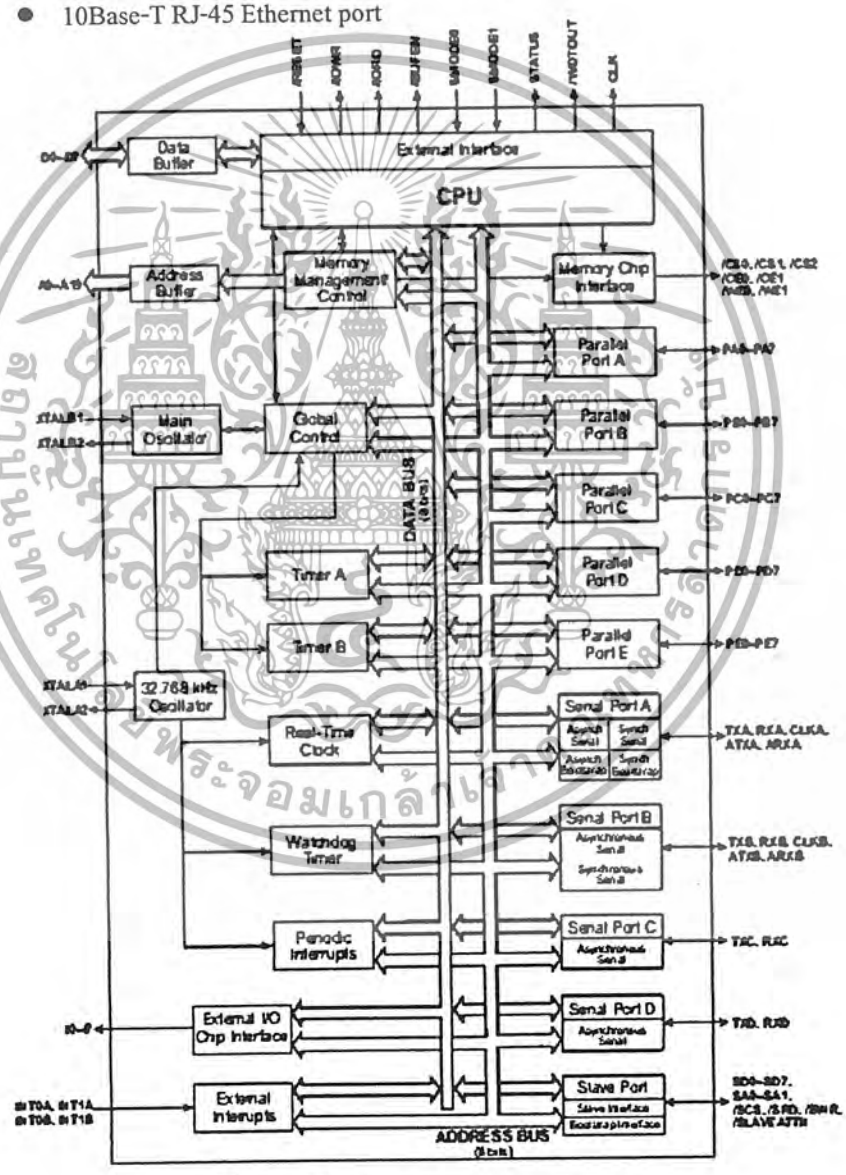
ชุดพัฒนา RCM2200 ประกอบไปด้วย Rabbit Core Module ดังแสดงในรูปที่ 2.1 และ Prototyping Board ซึ่งเราสามารถเขียนโปรแกรมทดสอบ RCM2200 รวมไปถึงการออกแบบแอพพลิเคชั่น

ชุดพัฒนานี้เป็นชุดพัฒนาสำหรับ TCP/IP เป็นของบริษัท Rabbit Semiconductor ซึ่งชุดพัฒนา RCM2200 ใช้โปรเซสเซอร์ของบริษัทตัวเอง โดยรุ่น RCM2200 มีคุณสมบัติดังนี้

- มีขนาดเล็ก : 1.60" × 2.30" × 0.86" (41 mm × 58 mm × 22 mm)
- ใช้ Microprocessor: Rabbit 2000 ทำงานที่ 22.1 MHz
- I/O ขนาน 26 line, กำหนดให้เป็น input หรือ output ได้ 16 line, 7 fixed inputs, 3 fixed outputs

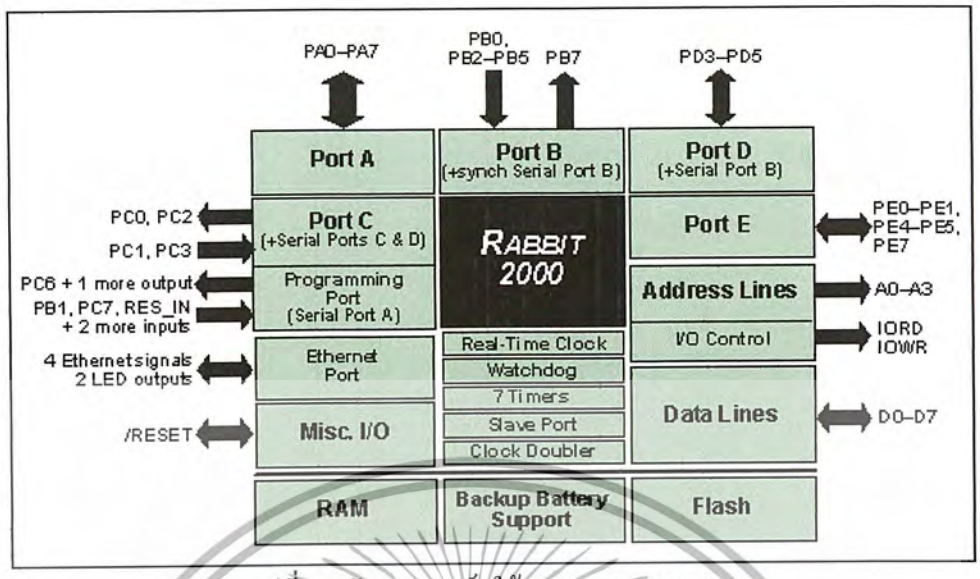
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 8 data lines (D0-D7)
- 4 address lines (A0-A3)
- Memory I/0 read, write
- External reset input
- 8-bit timers 5 ตัว และ 10-bit timers 2 ตัว
- 256K-512K flash memory, 128K-512K SRAM
- Real-time clock
- 10Base-T RJ-45 Ethernet port



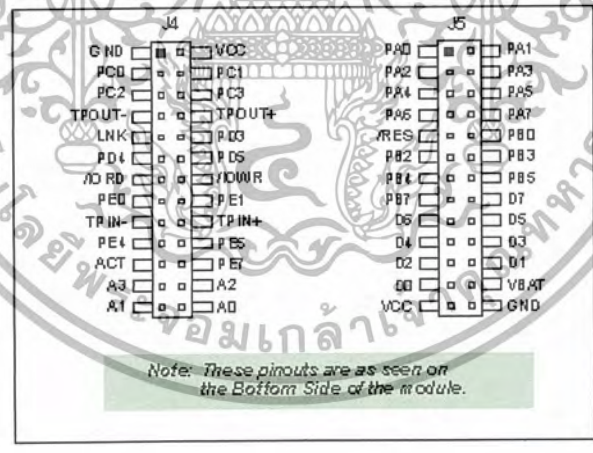
รูปที่ 2.2 บล็อกไดอะแกรมของ RCM2200

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ลักษณะพอร์ตใช้งานของ RCM 2200

RCM2200 มี 26 อินพุท/เอาท์พุท แบ่งออกเป็น ขนาด 8 บิต จำนวน 5 พอร์ต บน J4 และ J5 โดยขาสัญญาเป็นได้ทั้งอินพุทและเอาท์พุท คือ PA0-PA7, PD3-PD5, PE0-PE1, PE4, PE5 และ PE7 ดังแสดงในรูปที่ 2.2 และรูปที่ 2.3 ส่วนในรูปที่ 2.4 เป็นขาที่ใช้ต่ออุปกรณ์เพื่อใช้ในการรับและส่งข้อมูลภายนอก หน้าที่ของขาแต่ละขาแสดงไว้ในตารางที่ 2.1



รูปที่ 2.4 RCM2200 I/O Pinouts

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 หน้าที่ของขาต่างๆของ RCM2200

Pin		Pin Name	Default Use	Alternate Use	Notes
Header J4	1	GND			
	2	VCC			
	3	PC0	Output	TXD	
	4	PC1	Input	RXD	
	5	PC2	Output	TXC	
	6	PC3	Input	RXC	
	7	TPOUT-			Ethernet transmit port
	8	TPOUT+			
	9	LNK			Ethernet link (LNK) LED indicator
	10	PD3	Bitwise or parallel programmable I/O		
	11	PD4		ATXB output	
	12	PD5		ARXB input	
	13	/IORD	Input (I/O read strobe)		
	14	/IOWR	Output (I/O write strobe)		
	15	PE0	Bitwise or parallel programmable I/O	I0 control or INT0A input	
	16	PE1		I1 control or INT1A input	
	17	TPIN-			Ethernet receive port
	18	TPIN+			
	19	PE4	Bitwise or parallel programmable I/O	I4 control or INT0B input	
	20	PE5		I5 control or INT1B input	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 หน้าที่ของขาต่างๆของ RCM2200 (ต่อ)

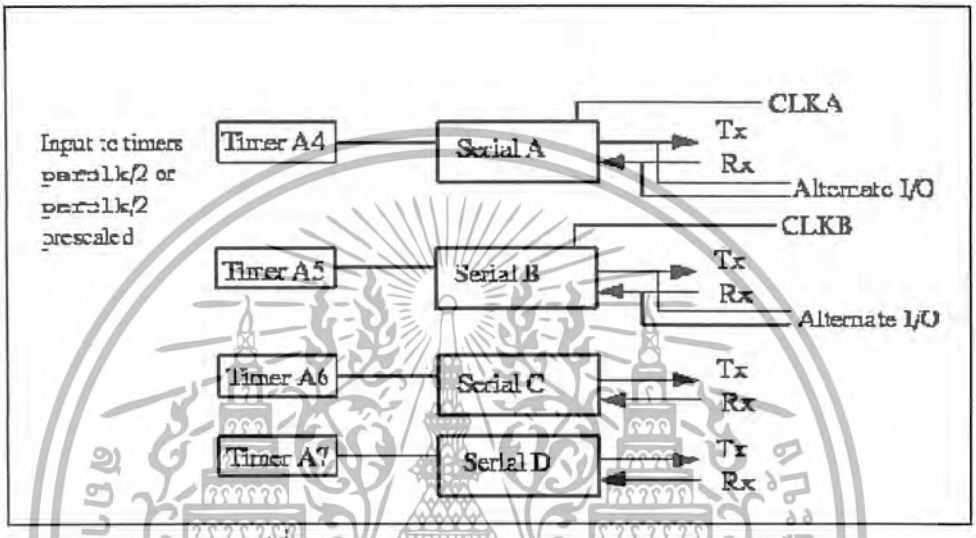
Pin		Pin Name	Default Use	Alternate Use	Notes
Header J4	21	ACT			Ethernet active (ACT)
	22	PE7	Bitwise or parallel programmable I/O	I7 control or slave port chip select	
	23-26	A[3:0]			Rabbit 2000 address bus
Header J5	1-8	PA[0:7]	Bytewide programmable parallel I/O	Slave port data bus SD0-SD7	
	9	/RESET	Reset output	Reset input	This weak output can be driven externally
	10	PB0	Input	Serial port clock CLKB input or output	
	11	PB2	Input	Slave port write /SWR	
	12	PB3	Input	Slave port read /SRD	
	13	PB4	Input	SA0	Slave port address lines
	14	PB5	Input	SA1	
	15	PB7	Output	Slave port attention line /SLAVEATTN	
	16-23	D[7:0]	Input/Output		Rabbit 2000 data bus
	24	VBAT	3 V battery input		
	25	VCC			
	26	GND			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Ports

2.2.1 พอร์ทอนุกรม (Serial Ports)

Rabbit 2000 มี 4 พอร์ทอนุกรม คือ A, B, C และ D พอร์ทแสดงในรูปที่ 2.5 ทั้งหมดสามารถทำอะซิงโครนัส ติดต่อโดยมีอัตราการส่งสูง พอร์ท A และ B สามารถถูกเปลี่ยนไปเป็นพอร์ทสัญญาณนาฬิกาได้และสามารถเปลี่ยนสลับกันของขา I/O พอร์ต A มีความสามารถพิเศษของการเป็น cold boot ของระบบไมโครโปรเซสเซอร์

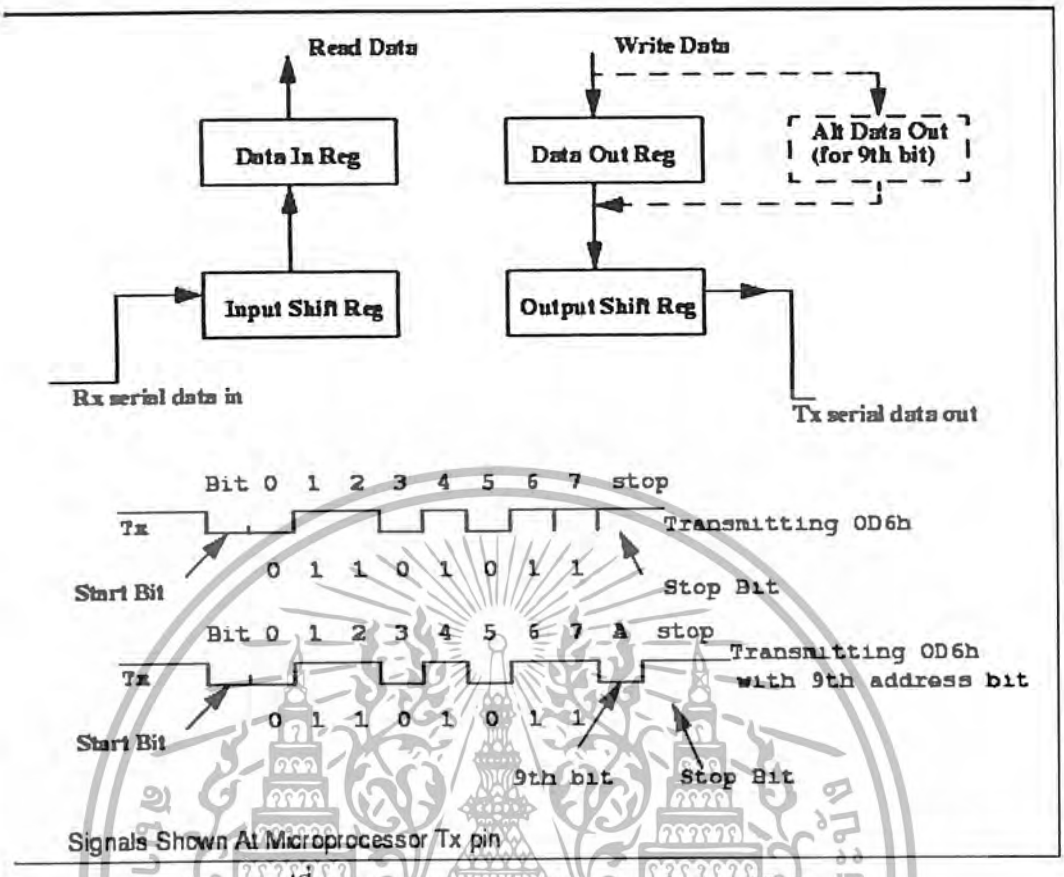


รูปที่ 2.5 Block Diagram of Rabbit Serial Ports

2.2.1.1 Register Layout Serial Port

ในรูปแสดงให้เห็นถึงฟังก์ชั่นบล็อกไดอะแกรมของพอร์ทอนุกรม โดยแต่ละพอร์ทอนุกรมจะมีดาต้ารีจิสเตอร์ การเขียนลงไปในดาต้ารีจิสเตอร์จะเริ่มทำการส่ง Least significant bit (LSB) โดยปกติจะเป็นการเริ่มส่งครั้งแรก เป็นเรื่องจริงสำหรับการติดต่อทั้ง 2 อย่างคือ อะซิงโครนัสและซิงโครนัส ถ้าเขียนถูกแสดงให้เห็นว่าจะทำให้แอดเดรสของดาต้ารีจิสเตอร์มีการสลับ บิตแอดเดรสพิเศษ หรือบิตที่ 9 (บิตที่ 8 ถ้าข้อมูลมี 7 บิต) ถูกส่ง เมื่อบิตข้อมูลได้ถูกรับ บิตแอดเดรสถูกอ่านจากดาต้ารีจิสเตอร์ รีจิสเตอร์คอนโทรลถูกใช้เพื่อส่งและรับตัวแปร Status รีจิสเตอร์อาจจะถูกเพื่อตรวจบนการคำนวณของพอร์ทอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 Functional Block Diagram of a Serial Port

สัญญาณนาฬิกาอินพุทที่เข้ามายังพอร์ทอนุกรมแต่ละตัวจะมี 6 ครั้ง อัตราในการส่งในโหมดอะซิงโครนัส และ 2 ครั้ง อัตราการส่งสำหรับโหมด clock อนุกรม ในขณะที่สัญญาณนาฬิกาภายในถูกใช้ Timers A4-A7 ให้สัญญาณนาฬิกาสำหรับ Serial Ports A-D เวลาสามารถแบ่งความถี่โดยแต่ละเลขมาจาก 1-256 มีความถี่อินพุทเพื่อ Timers สามารถถูกเลือกในทางต่างๆ หนึ่งในนั้นคือสัญญาณนาฬิการอบๆ โดยตัวเลือกและเลือกความถี่ผลึกสำหรับศูนย์กลางออสซิลเลเตอร์ โดยส่วนใหญ่จะใช้กระบวนการส่งข้อมูล ซึ่งจะมากที่สุดใช้ปกติอัตราสามารถถูกรับลงสู่เป็นจำนวนเกือบ 2400 bps ที่ความถี่สัญญาณนาฬิกา Rabbit สูงกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 Serial Port Registers

Register	Address xx = 00, 01, 10, 11 for A, B, C, D	Mnemonic x = A, B, C, D
Data Register	11xx0000	SxDR
Alternate Data Register to Send 9th (8th) Address Bit.	11xx0001	SxAR
Status Register (read, write to clear transmit IRQ)	11xx0011	SxSR
Control Register (write only)	11xx0100	SxCR

ตารางที่ 2.3 Serial Port Status Registers (adr = 11xx0011, xx = A,B,C,D)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1,0
Receiver ready (there is a byte in the receive data register)	9th bit received	Receive buffer overrun	0	Transmitter data register is full	Transmitter is sending a byte	0,0

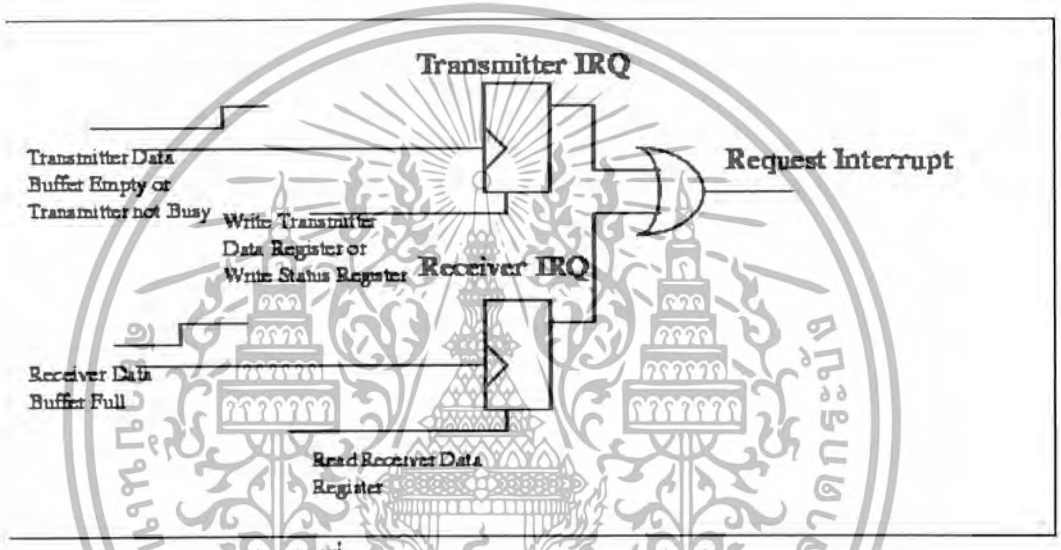
ตารางที่ 2.4 Serial Port Control Registers (adr = 11xx0100, xx = A,B,C,D)

Bit 7,6	Bit 5,4	Bit 3,2	Bit 1,0
00—no op 01—receive 1 byte clocked mode (A,B) 10—send one byte clocked mode (A,B) 11—reserved for future use	00—use port C for serial input 01—use port D for serial input 1x—disable receiver input	00—async mode, 8 bits 01—async mode 7 bits 10—clocked mode external clock (A,B) 11—clocked mode internal clock (A,B)	00—no interrupt 01— priority 1 interrupt 10—priority 2 11—priority 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.2 Serial Port Interrupt

โดยทั่วไปอินเตอร์รัพเว็กเตอร์จะถูกใช้สำหรับการรับส่งอินเตอร์รัพ มีการต้องการให้อินเตอร์รัพแยกออกมาจาก Flip-flop สำหรับเครื่องรับส่ง ถ้า flip-flop เหล่านี้ถูกตั้งขึ้น อินเตอร์รัพถูกต้องการ flip-flop ถูกตั้งโดยขอบขาขึ้นเท่านั้น flip-flop ถูกลบโดยพัลส์ที่ถูกสร้างขึ้นโดย I/O อ่านหรือเขียนซึ่งแสดงในรูป ในขณะที่การอินเตอร์รัพได้ถูกร้องขอ อินเตอร์รัพจะย้ายทันทีเมื่อมีอันดับก่อนหลังและชุดคำสั่งประมวลผลสำเร็จ อินเตอร์รัพจะหายไปถ้ามีความต้องการของ flip-flop ได้ถูกลบไปก่อนที่อินเตอร์รัพเข้ามาแทนที่ ถ้า flip-flop ไม่ถูกลบไปในอินเตอร์รัพ อินเตอร์รัพอื่นๆ จะเข้ามาเมื่อมีการเรียงลำดับที่ต่ำลงมา



รูปที่ 2.7 Generation of Serial Port Interrupts

ความต้องการอินเตอร์รัพรับ Flip-flop ถูกตั้งหลังจากบิตหยุดมีการทำงาน nominally ½ ของทางผ่านบิตหยุด บิตข้อมูลได้ถูกส่งบนสัญญาณนาฬิกาเดียวกันมาจากตัวรับ shift register ไปยังตัวรับของดาต้ารีจิสเตอร์

ความต้องการอินเตอร์รัพรับ Flip-flop ถูกตั้งบนขอบขึ้นของบิตเริ่มต้นสำหรับดาต้ารีจิสเตอร์ว่างและที่ขอบที่ตกลงของบิตหยุดสำหรับ shift register ที่ว่าง ถ้าไม่ใช่ดาต้ารีจิสเตอร์บนขอบที่เราตกลงไว้นี้ของบิตหยุด ผู้ส่งต้องทำงานด้วย ผู้ส่งไม่สามารถทำงานได้เท่านั้นถ้าดาต้ารีจิสเตอร์ว่างตำแหน่งของบิตหยุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

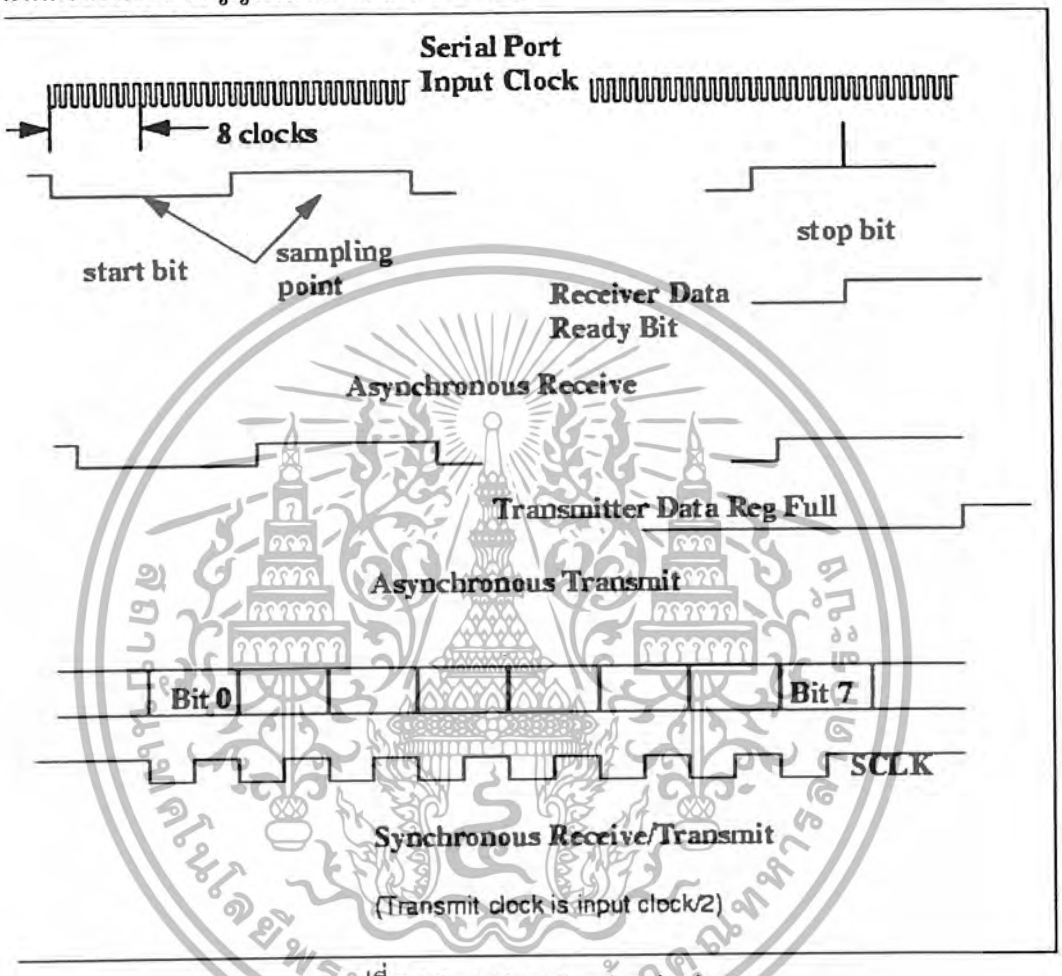
ตารางที่ 2.5 Peripheral Device Address and Interrupt Vectors

On-Chip Peripheral	I/O Address Range	ISR Starting Address
System Management (periodic interrupt)	0xh	{IIR, 00h}
Memory Management	1xh	No interrupts
Slave Port	2xh	{IIR,80h}
Parallel Port A	3xh	No interrupts
Parallel Port B	4xh	No interrupts
Parallel Port C	5xh	No interrupts
Parallel Port D	6xh	No interrupts
Parallel Port E	7xh	No interrupts
External I/O Control	8xh	No interrupts
External Interrupts	9xh	INT0 - {EIR, 00h} INT1 - {EIR, 10h}
Timer A	Axh	{IIR, A0h}
Timer B	Bxh	{IIR, B0h}
Serial Port A	Cxh	{IIR, C0h}
Serial Port B	Dxh	{IIR, D0h}
Serial Port C	Exh	{IIR, E0h}
Serial Port D	Fxh	{IIR, F0h}
RST 10 instruction	n/a	{IIR, 20h}
RST 18 instruction	n/a	{IIR, 30h}
RST 20 instruction	n/a	{IIR, 40h}
RST 28 instruction	n/a	{IIR, 50h}
RST 38 instruction	n/a	{IIR, 70h}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.3 Clocked Serial Ports

พอร์ท A และ B สามารถอยู่ในโหมด clock ได้โดยที่ data line และ clock line จะถูกขับดังแสดงในรูป ข้อมูลและสัญญาณนากาจะถูกเตรียม 8 บิต burst พอร์ทอนุกรมสามารถสร้างสัญญาณนาฬิกาได้ หรือจัดสัญญาณนาฬิกาเตรียมไว้ภายนอก



รูปที่ 2.8 Serial Port Synchronization

ตารางที่ 2.6 Synchronous Serial Port Signals

Rabbit Signal Names	Pin Function
CLKA or CLKB	Serial Clock
TxA or TxB on Parallel Port C ATxA or ATxB on Parallel Port D	Data Transmit
RxA or RxB on Parallel Port C ARxA or ARxB on Parallel Port D	Data Receive

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการส่งโดยใช้โหมดสัญญาณนาฬิกาภายใน ผู้ใช้ต้องโหลดจากดาต้ารีจิสเตอร์และเก็บโค๊ดในการส่ง ในขณะที่ Shift register จบการส่งสัญลักษณ์ ถ้าอื่นๆ ดาต้ารีจิสเตอร์จะถูกโหลดเข้าไป shift register และถูกส่งโดย 8-clock burst หนึ่งตัวอักษรทำงานอยู่ในกระบวนการประมวลผลของการส่ง ในขณะที่ตัวอื่นๆ รออยู่ในดาต้ารีจิสเตอร์เก็บค่าแบบ tagged กับโค๊ตในการส่ง โค๊ดในการส่งจะมีประสิทธิภาพในการเก็บข้อมูลชั่วคราวเป็น 2 เท่า

ในการรับตัวอักษรที่อยู่ในโหมดสัญญาณนาฬิกาภายใน รีจิสเตอร์รับตัวต่อไปควรหยุดการทำงาน ผู้ใช้เก็บโค๊ดตัวรับในรีจิสเตอร์ควบคุม การเกิดของ 8 สัญญาณนาฬิกาจะมีการสร้างและผู้ส่งจำเป็นต้องตรวจสอบสัญญาณนาฬิกาและเอาท์พุทตัวต่อไป ไปยังสายข้อมูลบนขอบขาลงของแต่ละสัญญาณนาฬิกา ตัวรับจะเป็นตัวอย่างของข้อมูลที่เป็นขอบขาขึ้นของแต่ละสัญญาณนาฬิกา โหมดในการรับไม่สามารถทำการเพิ่มที่เก็บข้อมูลเป็น 2 เท่าได้ ในขณะที่ใช้สัญญาณนาฬิกาภายในรีจิสเตอร์ตัวต่อไปไม่สามารถทำงานได้ก่อนที่จะรับตัวอักษรอื่นสามารถทำการเริ่มต้นได้ แต่อย่างไรก็ตาม การร้องขออินเตอร์รัพและตัวอักษรอื่นมีการเข้ามาเมื่ออยู่ขอบขาขึ้นของพัลส์ของสัญญาณนาฬิกาตัวต่อไป ถ้าโค๊ดตัวรับจะถูกเก็บก่อนที่ตำแหน่งของขอบขาลงตัวต่อไป ตัวรับตัวอื่นๆ จะเริ่มโดยไม่มีการหยุดสัญญาณนาฬิกา อินเตอร์รัพจะยอมใน ½ ของสัญญาณนาฬิกา

ในการส่งในโหมดสัญญาณนาฬิกาภายนอก โดยที่ผู้ใช้สามารถโหลดดาต้ารีจิสเตอร์และเก็บโค๊ดการส่ง ในขณะที่รีจิสเตอร์ตัวต่อไปไม่ทำงานและตัวรับจะเตรียมสัญญาณนาฬิกา ดาต้าบิตจะถูกส่งไปยังรีจิสเตอร์ตัวต่อไปและเลื่อนออก โดยการส่งจะส่งไปยังรีจิสเตอร์ตัวต่อไป ไบต์ไม่สามารถโหลดเข้าไปในรีจิสเตอร์ตัวส่ง และโค๊ดการส่งใหม่จะถูกเก็บ

การรับไบต์ในโหมดสัญญาณนาฬิกาภายนอก โดยผู้ใช้จะตั้งค่ารับโค๊ดสำหรับค่าแรกของไบต์และเก็บโค๊ดที่ได้รับมาสำหรับไบต์ต่อไป หลังจากแต่ละไบต์จะถูกลบออกจากดาต้ารีจิสเตอร์ตั้งแต่โค๊ดตัวรับจะถูกเก็บหลังจากตัวส่งส่งไปเก็บในไบต์ต่อไป เครื่องรับต้องบริการอินเตอร์รัพภายใน ½ สัญญาณนาฬิกา โดยอัตราในการส่ง เพื่อรักษาเครื่องส่งกำลังความเร็วเต็มที่ นี่ไม่ได้ฝึกถ้าไม่ใช่การควบคุมการไหลถูกทำหรือผู้ส่งใส่ช่องว่างระหว่างสัญญาณนาฬิกานั้น

โดยปกติเพื่อที่จะดำเนินการคมนาคมความเร็วสูงต่อไป การจัดการที่ดีที่สุดจะสำหรับเครื่องรับเพื่อเตรียมสัญญาณนาฬิกา เมื่อเครื่องรับเตรียมแล้ว ผู้ส่งสามารถเก็บขึ้นเพราะว่ามันสามารถบีบขนาดและมีเวลาตัวอักษรเต็มเพื่อตอบอินเตอร์รัพว่างเปล่าดาต้ารีจิสเตอร์ผู้ส่งเสมอ เครื่องรับจะตอบอินเตอร์รัพว่าถูกกำหนดบนขอบขาขึ้นของสัญญาณนาฬิกาสุท้าย ถ้าอินเตอร์รัพสามารถถูกบริการภายใน ½ ของสัญญาณนาฬิกา จะถูกบริการไม่มีค้างในอัตราข้อมูล ถ้ามันใช้เครื่องรับนานกว่าเพื่อตอบ แล้วจะเป็นช่องว่างระหว่างไบต์ ความยาวซึ่งยึดหลักบนอินเตอร์รัพตัวอย่างเช่น ถ้าอัตราในการส่งคือ 400,000 bps แล้วสามารถถูกส่งได้สูงถึง 50,000 ไบต์ต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

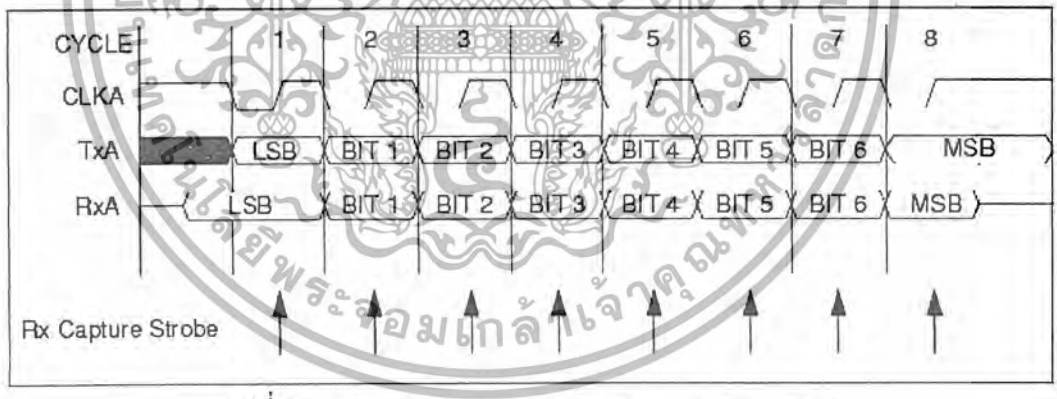
หรือ 1 ไบต์ ในทุก 20 ไมโครวินาที ถ้าไม่มีข้อมูลผู้ส่งสามารถตอบอินเตอร์รัพภายใน 20 ไมโครวินาที เครื่องรับสามารถตอบอินเตอร์รัพของมันภายใน ½ ของสัญญาณนาฬิกา หรือ 1.25 ไมโครวินาที ถ้ามันสามารถตอบภายใน 3/2 ของสัญญาณนาฬิกา หรือ 2.75 ไมโครวินาที อัตราข้อมูลจะช้าถึง 44,444 ไบต์ต่อวินาที ถ้ามันสามารถตอบใน 2.5 ของสัญญาณนาฬิกา หรือ 6.25 ไมโครวินาที อัตราช้าถึง 40,000 ไบต์ต่อวินาที ถ้ามันสามารถตอบใน 3.5 ของสัญญาณนาฬิกา หรือ 8.75 ไมโครวินาที อัตราช้าถึง 36,363 ไบต์ต่อวินาที

ถ้ามีการสื่อสารแบบ 2 ทาง สัญญาณนาฬิกาสามารถถูกย้อนไปมาเพื่อให้เครื่องรับเตรียมสัญญาณนาฬิกา นี่เป็นการยุ่งยากมากขึ้น เครื่องรับไม่สามารถเริ่มข้อความ ถ้าความพยายามเครื่องรับเพื่อรับตัวอักษรและผู้ส่งไม่มีกำลังส่ง บิตสุดท้ายที่ส่งจะถูกรับสำหรับ 8 บิตทั้งหมด

2.2.1.4 Clocked Serial Timing

2.2.1.4.1 Clocked Serial Timing with Internal Clock

สำหรับการติดต่อสื่อสารแบบอะซิงโครนัสแบบอนุกรม สัญญาณนาฬิกาแบบอนุกรมสามารถสร้างได้จาก Rabbit หรืออุปกรณ์ภายนอก แผงผังเวลาดังรูป จะแสดงให้เป็นการสื่อสารแบบ 2 ทาง สัญญาณนาฬิกาแบบอนุกรมถูกสร้างภายในโดย Rabbit โหมดสัญญาณนาฬิกาที่สามารถเข้ากันได้กับ SPI อื่นๆ ที่สนับสนุนโดย Rabbit 2000

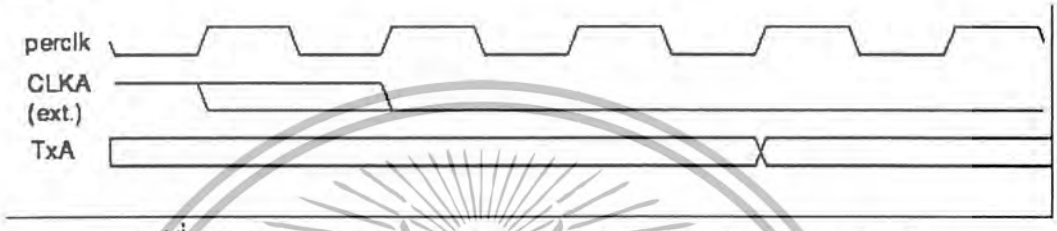


รูปที่ 2.9 Full-Duplex Clocked Serial Timing Diagram with Internal Clock

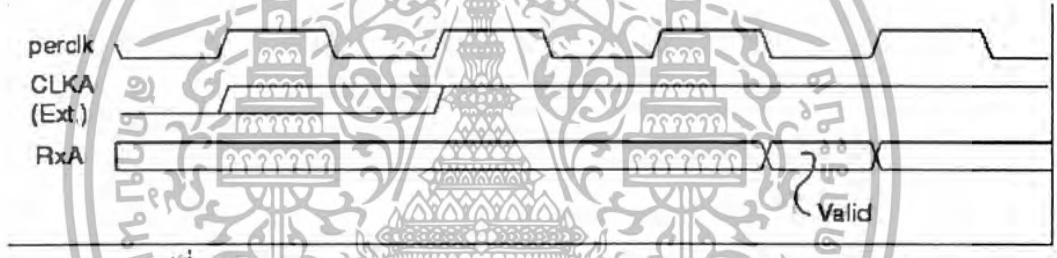
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.4.2 Clocked Serial Timing with External Clock

ในระบบที่ซึ่งสัญญาณนาฬิกาแบบอนุกรมถูกสร้างขึ้นโดยอุปกรณ์ภายนอก สัญญาณนาฬิกาจะถูกส่งไปพร้อมกับสัญญาณเสริมของอุปกรณ์ภายใน (perclk) ก่อนที่ข้อมูลสามารถจะถูกส่งหรือรับโดย Rabbit การถือหลักเมื่อสัญญาณนาฬิกาแบบอนุกรมภายนอกถูกสร้างขึ้น ในความสัมพันธ์เพื่อ perclk มันอาจจะใช้ที่ใดๆ จาก 2 ถึง 3 รอบสัญญาณนาฬิกาสำหรับสัญญาณนาฬิกาภายนอกที่จะถูกเกิดขึ้นพร้อมกันกับสัญญาณนาฬิกาภายในก่อนข้อมูลใดๆ จะสามารถถูกโยกย้าย



รูปที่ 2.10 Synchronous Serial Data Transmit Timing with External Clock



รูปที่ 2.11 Synchronous Serial Data Receive Timing with External Clock

2.2.2 พอร์ทขนาน (Parallel Ports)

Rabbit 2000 มีพอร์ทขนานอยู่ 5 พอร์ท โดยแต่ละพอร์ทมีขนาด 8 บิต ประกอบด้วยพอร์ท A, B, C, D และ E โดยแต่ละขาจะใช้สำหรับพอร์ทขนานซึ่งใช้กับฟังก์ชั่นต่างๆ โดยมีคุณสมบัติของแต่ละพอร์ทดังนี้

- พอร์ท A แชร์กับเชื่อมต่อพอร์ทข้อมูลของ slave port
- พอร์ท B แชร์กับเส้นทางควบคุมสำหรับ slave port และเป็น clock I/O สำหรับ clock ของพอร์ทอนุกรม A และ B
- พอร์ท C แชร์กับพอร์ทข้อมูล I/O แบบอนุกรม
- พอร์ท D มีขนาด 4 บิต แชร์กับสลับขา I/O สำหรับพอร์ทอนุกรม A และ B 4 บิต ไม่สามารถแชร์ได้ โดยที่พอร์ท D สามารถแก้ไขให้มีการออกของข้อมูล พอร์ท D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ยังมี output preload registers ซึ่งสามารถสร้าง clock ไปยังเอาท์พุตรีจิสเตอร์ อยู่ใน timer control สามารถสร้างพัลส์ได้

- พอร์ท E ทุกบิตของพอร์ท E สามารถปรับแก้เป็น I/O Strobe 4 บิตของพอร์ท E สามารถใช้กับอินเตอร์รัพข้อมูลภายนอก 1 บิตของพอร์ท E สามารถแชร์ชิพ slave port ที่เลือกเอาไว้ พอร์ท E มี output preload register อยู่ภายใต้การควบคุมของ timer control สำหรับสร้างพัลส์

2.2.2.1 Parallel Ports A

พอร์ทขนาน A เป็นพอร์ทเดียวที่มีรีจิสเตอร์ทั้งการอ่านและเขียน รีจิสเตอร์นี้ไม่ควรถูกใช้ถ้าหาก slave port มีการทำงาน เพราะว่า รีจิสเตอร์คอนโทรล slave port จะใช้โดยไม่คำนึงถึงพอร์ท A เมื่อพอร์ททำงานเป็นอินพุท จะเก็บข้อมูลไว้ที่ตำแหน่ง 080h ใน SPCR (Slave Port Control Register) เมื่อพอร์ททำงานเป็นเอาท์พุท จะเก็บข้อมูลที่ตำแหน่ง 084h ใน SPCR พอร์ทขนาน A จะเป็นอินพุทหลังจากการ reset

ในขณะที่พอร์ทอ่านค่าที่อ่านได้บนขาที่ค่าเท่ากับ "1" จะมีค่าเป็น high และ "0" จะมีค่าเป็น low สิ่งนี้สามารถแตกต่างจากค่าที่เก็บไว้ในรีจิสเตอร์ได้เพื่อสถานะที่แตกต่างโดยแรงไฟฟ้าภายนอก

2.2.2.2 Parallel Ports B

พอร์ทขนาน B จะประกอบไปด้วย 6 อินพุทและ 2 เอาท์พุทดังตารางที่แสดง

ตารางที่ 2.7 Parallel Port B Data Register PBDR (adr = 040h)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Read	Echo drive	Echo drive	PB5 in	PB4 in	PB3 in	PB2 in	PB1 in	PB0 in
Write	PB7	PB6	x	x	x	x	x	x

เมื่อ Slave port ทำงาน พอร์ทขนาน PB2-PB7 จะเป็นตัวกำหนดค่าฟังก์ชั่นต่างๆให้ slave port แต่อย่างไรก็ตาม เป็นไปได้ที่จะใช้ PB0-PB5 ในการอ่านข้อมูล ในขณะที่ PB2-PB7 ถูกใช้งานสำหรับ slave port อยู่ หรือ PB6 และ PB7 สามารถอ่านสัญญาณที่ขับเคลื่อนมาได้

62119
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PB0 จะทำหน้าที่เป็น clock สำหรับพอร์ทอนุกรม B และ PB1 จะทำหน้าที่เป็น clock ของพอร์ทอนุกรม A

2.2.2.3 Parallel Ports C

พอร์ทขนาน C จะประกอบไปด้วย 4 อินพุท และ 4 เอาท์พุท ที่เป็นพอร์ทเลขคู่คือ PC0, PC2, PC4, PC6 จะเป็นเอาท์พุท และ PC1, PC3, PC5, PC7 จะเป็นอินพุท ในขณะที่ดาต้ารีจิสเตอร์อ่าน บิตที่ 1,3,5,7 จะส่งข้อมูลไปยังที่ขา โดยที่บิต 0,2,4,6 จะส่งค่าของสัญญาณเพื่อที่จะทำให้เป็นเอาท์พุทบัฟเฟอร์ โดยแต่ละดาต้ารีจิสเตอร์ของพอร์ทขนาน C จะถูกขับไปยังขาหรือพอร์ทใดพอร์ทหนึ่งของพอร์ทอนุกรม ค่าบิตใน PCFR (Port C Function Register) ถ้าดาต้ารีจิสเตอร์หรือพอร์ทที่มีการส่งจะส่งไปยังขานั้น

ตารางที่ 2.8 Parallel Port C Data Register and Function Register

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCDR (r) adr = 050h	PC7 in	Echo drive	PC5 in	Echo drive	PC3 in	Echo drive	PC1 in	Echo drive
PCDR (w) adr = 050h	x	PC6	x	PC4	x	PC2	x	PC0
PCFR (w) adr = 055h	x	Drive TXA	x	Drive TXB	x	Drive TXC	x	Drive TXD

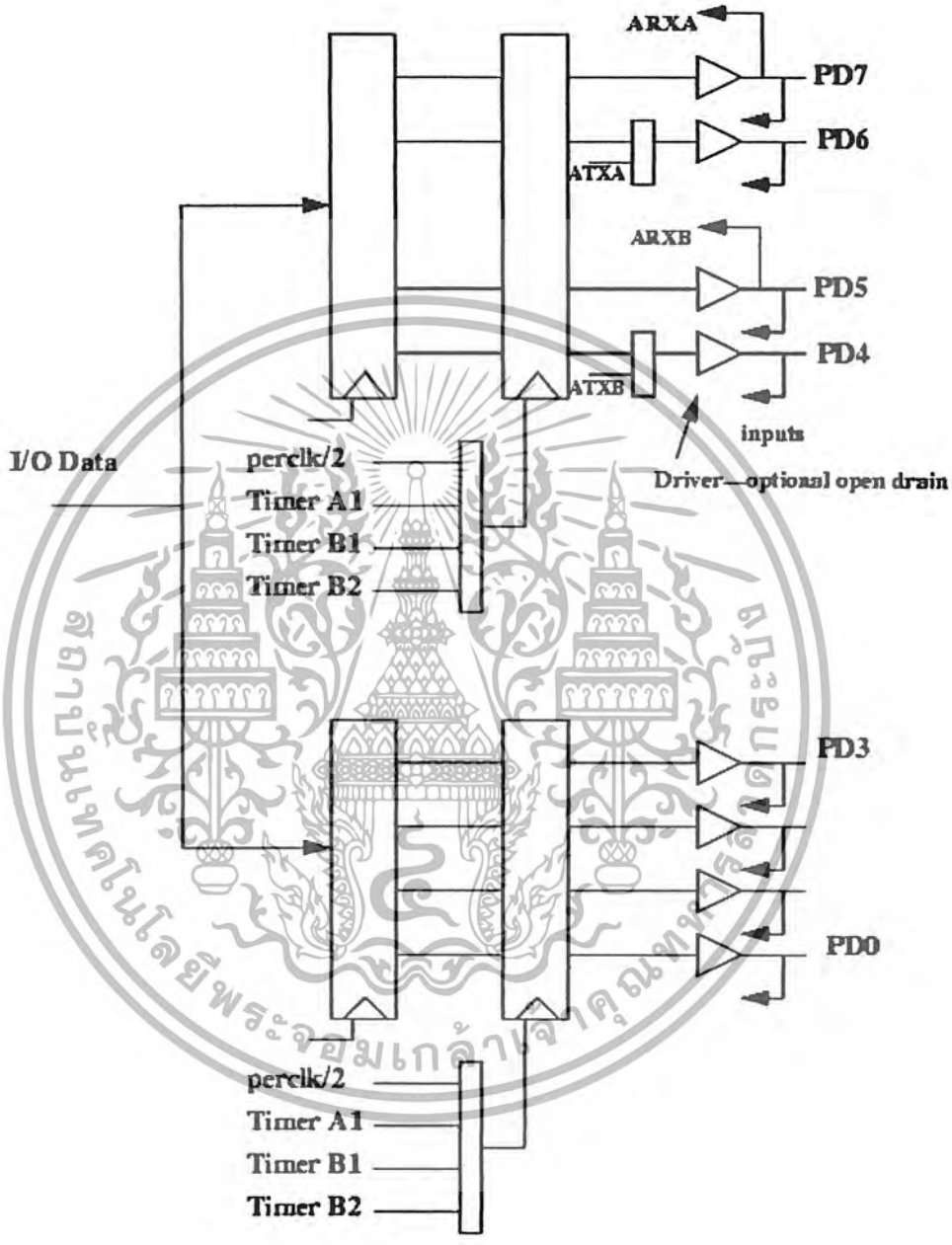
พอร์ทขนาน C จะแชร์โดยขาเดียวกับอนุกรมพอร์ท A-D โดยที่อินพุทของบนพอร์ทขนานสามารถปรับเป็นพอร์ทอินพุทแบบอนุกรม

2.2.2.4 Parallel Ports D

พอร์ทขนาน D จะมี 8 ขาซึ่งสามารถโปรแกรมได้ โดยสามารถแบ่งเป็นอินพุทและเอาท์พุทในขณะที่โปรแกรมอยู่นั้น ขาสามารถแบ่งได้โดยการปิดพอร์ทเอาท์พุทหรือเอาท์พุทแบบปกติ ขาของพอร์ทเอาท์พุทสามารถกำหนดตำแหน่งโดยบิตถ้ามีการออกแบบ พอร์ท D บิตที่ 4 และ 5 สามารถเลือกใช้สำหรับพอร์ทอนุกรม B และบิตที่ 6 และ 7 สามารถเลือกใช้สำหรับพอร์ทอนุกรม A การ reset ดาต้าไดเร็กรีจิสเตอร์ จะเป็น 0 รวมทั้งขาอินพุทด้วย บิตที่อยู่ในรีจิสเตอร์จะปรับค่าเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0 เพื่อทำให้แน่ใจว่าข้อมูลเข้าไปในรีจิสเตอร์ คือ สิ่งที่นำออกเมื่อโหลด รีจิสเตอร์อื่นๆ ทั้งหมดที่เกี่ยวข้องกับพอร์ทจะไม่ถูกทำการ reset



รูปที่ 2.12 Parallel Port D Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.9 Parallel Port D Registers

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PDDR (R/W) adr = 060h	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
PDDCR (W) adr = 066h	out = open drain	out = open drain	out = open drain	out = open drain	out = open drain	out = open drain	out = open drain	out = open drain
PDFR (W) adr = 065h	x	alt TXA	x	alt TXB	x	x	x	x
PDDDR (W) adr = 067h	dir =out	dir =out	dir =out	dir =out	dir =out	dir =out	dir =out	dir =out
PDB0R (W) adr = 068h	x	x	x	x	x	x	x	PD0
PDB1R (W) adr = 069h	x	x	x	x	x	x	PD1	x
PDB2R (W) adr = 06Ah	x	x	x	x	x	PD2	x	x
PDB3R (W) adr = 06Bh	x	x	x	x	PD3	x	x	x
PDB4R (W) adr = 06Ch	x	x	x	PD4	x	x	x	x
PDB5R (W) adr = 06Dh	x	x	PD5	x	x	x	x	x
PDB6R (W) adr = 06Eh	x	PD6	x	x	x	x	x	x
PDB7R (W) adr = 06Fh	PD7	x	x	x	x	x	x	x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.10 Parallel Port D Control Register (adr = 064h)

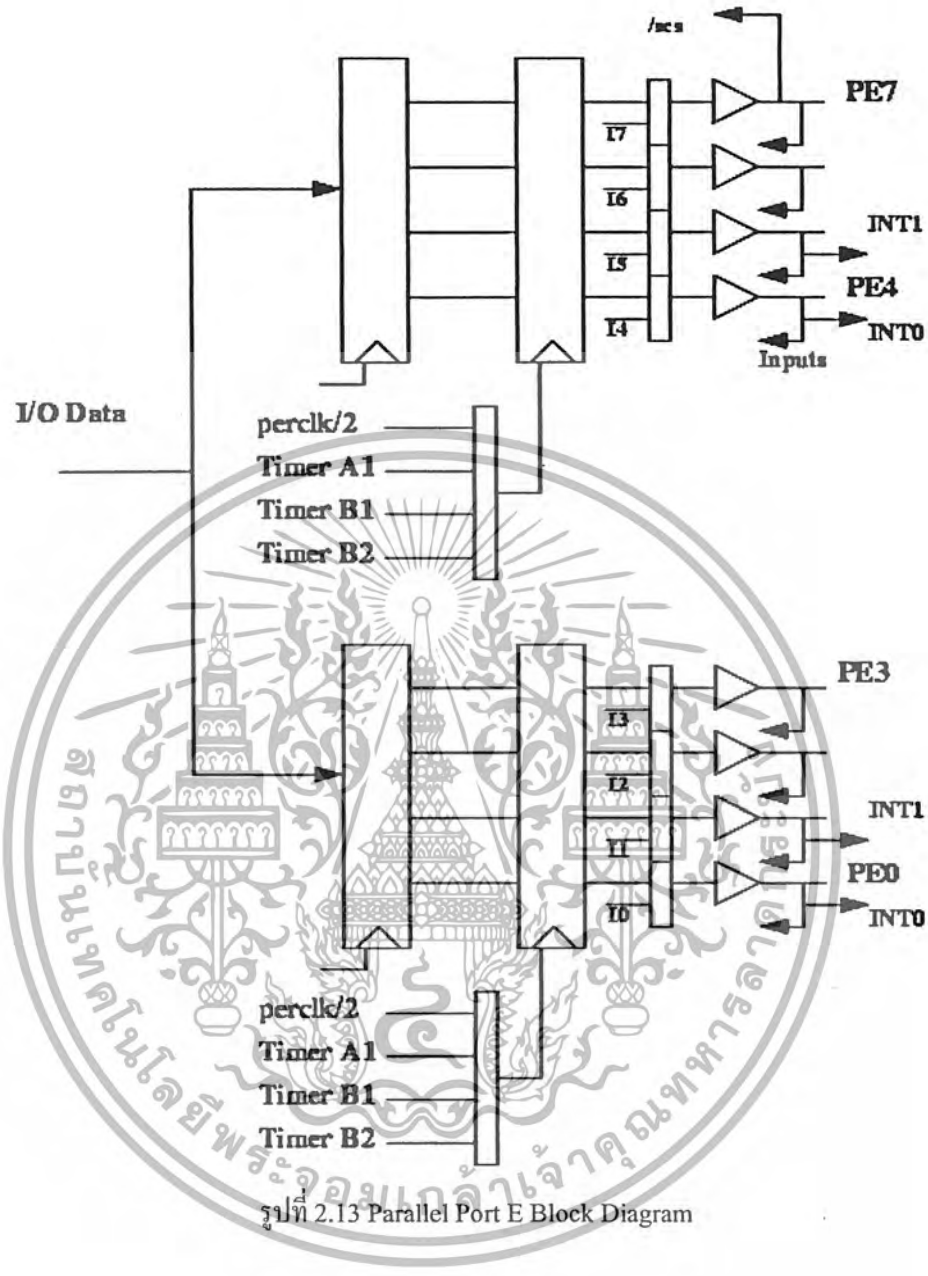
Bits 7, 6	Bits 5, 4	Bits 3, 2	Bits 1, 0
x	00—clock upper nibble on pclk/2 01—clock on timer A1 10—clock on timer B1 11—clock on timer B2	x	00—clock lower nibble on pclk/2 01—clock on timer A1 10—clock on timer B1 11—clock on timer B2

- PDDR—Parallel port D data register. Read/Write.
- PDDDR—Parallel port D data direction register A "1" ทำให้ขาเป็นเอาท์พุทและสามารถเขียนได้อย่างเดียวเท่านั้น
- PDDCR—Parallel port D drive control register A "1" ทำให้ขานั้นสามารถส่งข้อมูลได้ แต่สามารถเขียนได้อย่างเดียว
- PDFR—Parallel port D function control register พอร์ทนี้สามารถเป็นพอร์ทเอาท์พุทโดยมีตำแหน่งที่ 4 และ 6 เป็นพอร์ทอนุกรม เขียนได้เท่านั้น
- PDBxR—เป็นพอร์ทขนาด 8 บิต ใช้ในการตั้งเป็นเอาท์พุทพอร์ทเฉพาะ
- PDCR—Parallel port D control register เป็นรีจิสเตอร์ที่ถูกใช้สำหรับการควบคุมสัญญาณนาฬิกาของส่วนบนและส่วนล่างแบ่งออกเป็นส่วนๆ ของเอาท์พุทรีจิสเตอร์ตัวสุดท้าย มีการ reset ค่าที่บิต 0, 1, 4, และ 5 จะมีค่าเป็น 0

2.2.2.5 Parallel Ports E

พอร์ทขนาน E จะมีขา I/O ทั้งหมด 8 ขา และสามารถเขียนโปรแกรมและแบ่งเป็นขาอินพุทและเอาท์พุท โดยที่ PE7 ถูกใช้เมื่อพอร์ทถูกเลือกในขณะที่ slave port กำลังทำงานอยู่โดยแต่ละพอร์ทเอาท์พุทของพอร์ท E สามารถปรับเปลี่ยนเป็น I/O Strobes 4 ในพอร์ท E จะถูกใช้เป็นการควบคุม เป็นสิ่งที่ทำได้เพื่อการสร้าง precise timing pulse

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 Parallel Port E Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.11 Parallel Port E Registers

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PEDR (R/W) adr = 070h	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
PEFR (W) adr = 075h	alt /I7	alt /I6	alt /I5	alt /I4	alt /I3	alt /I2	alt /I1	alt /I0
PEDDR (W) adr = 077h	dir = out	dir = out	dir = out	dir = out	dir = out	dir = out	dir = out	dir = out
PEB0R (W) adr = 078h	x	x	x	x	x	x	x	PE0
PEB1R (W) adr = 079h	x	x	x	x	x	x	PE1	x
PEB2R (W) adr = 07Ah	x	x	x	x	x	PE2	x	x
PEB3R (W) adr = 07Bh	x	x	x	x	PE3	x	x	x
PEB4R (W) adr = 07Ch	x	x	x	PE4	x	x	x	x
PEB5R (W) adr = 07Dh	x	x	PE5	x	x	x	x	x
PEB6R (W) adr = 07Eh	x	PE6	x	x	x	x	x	x
PEB7R (W) adr = 07Fh	PE7	x	x	x	x	x	x	x

ตารางที่ 2.12 Parallel Port E Control Register (adr = 074h)

Bits 7, 6	Bits 5, 4	Bits 3, 2	Bits 1, 0
x	00—clock upper nibble on pclk/2 01—clock on timer A1 10—clock on timer B1 11—clock on timer B2	x	00—clock lower nibble on pclk/2 01—clock on timer A1 10—clock on timer B1 11—clock on timer B2

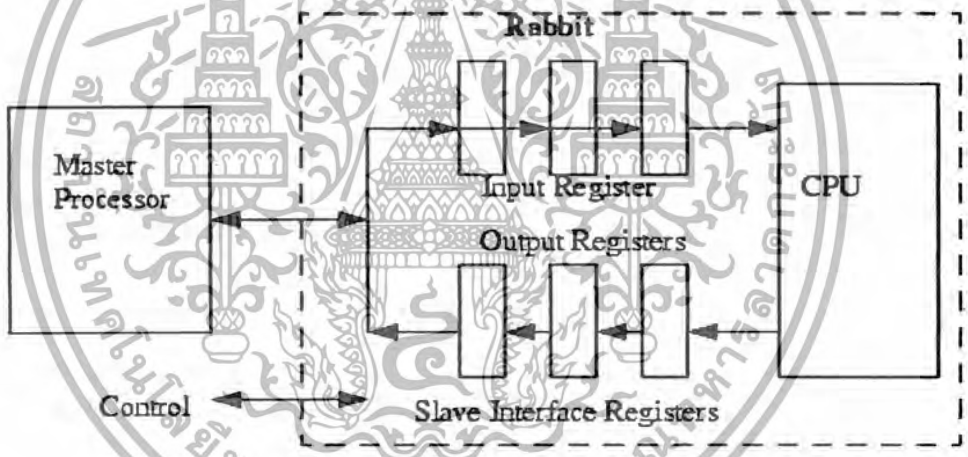
- PEDR—Port E data register อ่านค่าต่างๆ ที่ขา เขียนที่พอร์ท E preload register
- PEDDR—Port E data direction register เป็น "1" ทำให้มีการติดต่อที่ขาคล้ายกับเอาท์พุทรีจิสเตอร์มีค่าเท่ากับ “0” หลังจากการ reset
- PEFR—Port E function register. เป็น "1" ทำให้เอาท์พุทออกมามีลักษณะคล้ายกับ I/O strobe โดยธรรมชาติของ I/O strobe ที่ถูกควบคุมโดย I/O bank control registers (IBxCR) เส้นทางของข้อมูลจะถูกตั้งค่าเพื่อเป็นเอาท์พุทของ I/O strobe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PEBxR—เป็นรีจิสเตอร์เฉพาะเมื่อตั้งค่าเฉพาะบิตเอาท์พุท
- PECR—Parallel port E control register เป็นรีจิสเตอร์ที่ใช้ควบคุมการทำงานของสัญญาณนาฬิกาของส่วนบนและส่วนล่างแบ่งออกเป็นส่วนๆ ของเอาท์พุท รีจิสเตอร์ตัวสุท้าย มีการ reset บิตที่ 0, 1, 4, และ 5 ให้เป็น "0"

2.2.3 พอร์ทลูก (Slave Port)

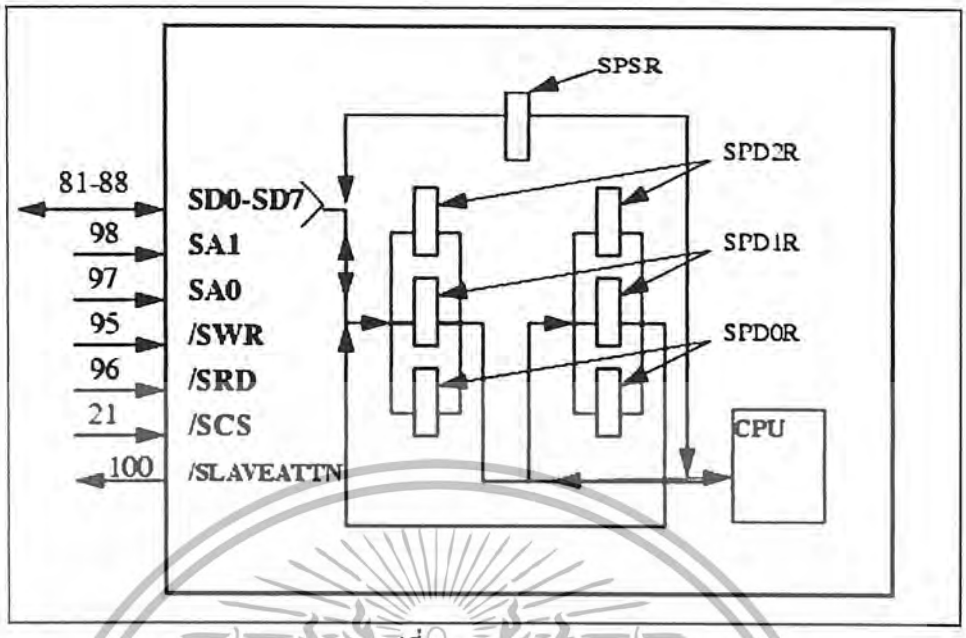
พอร์ทลูกได้ถูกออกแบบให้มาใช้กับโปรเซสเซอร์ตัวอื่นๆ หรือไม่ก็ใช้กับ Rabbit ตัวอื่นๆ มีพอร์ทสามารถแบ่งได้ด้วย พอร์ทขนาน A และพอร์ทข้อมูล bidirectional โดยที่ตัวแม่อ่านรีจิสเตอร์ทั้ง 3 รีจิสเตอร์ก่อนแล้วค่อยเลือกส้นทางได้มาจากรีจิสเตอร์แอดเดรสและอ่านได้มาจาก strobe เป็นสาเหตุ ที่ทำให้รีจิสเตอร์สามารถมีเอาท์พุทด้วยพอร์ทได้ด้วยที่ริจิสเตอร์ตัวเดิมสามารถเขียนเหมือนกับรีจิสเตอร์ I/O โดย Rabbit ตัวลูกที่เพิ่มอีก 3 เส้นทางจะเอามาทำเป็นเส้นทางย้อนกลับ



รูปที่ 2.14 เส้นทางข้อมูลของพอร์ทลูก

Rabbit ลูก สามารถอ่านรีจิสเตอร์ได้ด้วยรีจิสเตอร์ที่เหมือนกัน เหมือนกันกับ I/O รีจิสเตอร์ใขขณะที่ข้อมูลเข้ามาเขียนลงรีจิสเตอร์อีกตัวหนึ่ง ในขณะที่ตัวแม่สามารถให้ตัวลูกอ่านข้อมูลที่เข้าไป ส่วนสายอื่นๆ จะให้ตัวแม่ จะบอกการไหลของข้อมูลและยังไม่ได้อ่านจากตัวแม่ พอร์ทลูกสามารถส่งย้อนสัญญาณไปยังตัวแม่เพื่อที่จะจัดรูปแบบ เพื่อใช้ในการติดต่อสื่อสารในพอร์ทลุก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 Slave port

Slave processor มีการเชื่อมต่อกับดาต้าบัสของตัวแม่ การติดต่อสื่อสารระหว่างตัวแม่กับตัวลูก สามารถอาศัยรีจิสเตอร์ 3 ตัวบนอุปกรณ์บน Rabbit ในแต่ละเส้นทางของการติดต่อสื่อสารสำหรับผลลัพธ์ทั้งหมด 6 ดาต้ารีจิสเตอร์ดังภาพรวมพอร์ทลูกที่ซึ่งถูกอ่านโดย slave หรือ master โดย 2 เส้นทางของ slave address ได้ถูกใช้โดย master เป็นตัวเลือกรีจิสเตอร์ ให้อ่านหรือเขียนรีจิสเตอร์นำเอาข้อมูลจาก master ไปยัง slave รีจิสเตอร์นั้นทำงานสวนกลับกันทางเดิม ปรากฏว่ามี read-write register ทั้ง 3 ตัวอยู่ภายใต้การควบคุม ดังนั้น master สามารถจัดการเขียนส่งไปยัง status register ซึ่งถูกเอาไปใช้เช่นเดียวกับอุปกรณ์สัญญาณต่างๆ และเป็นไปไม่ได้ที่สามารถเขียนส่งไปยัง status register โดยที่รีจิสเตอร์ทั้ง 3 ตัวนี้สามารถเขียนเช่นเดียวกับอ่าน register ส่ง ไปยัง slave Rabbit master เป็นตัวจัดการความสามารถของ strobe จนถึงอ่าน tree read data register และ status register ซึ่งรีจิสเตอร์เป็นรีจิสเตอร์ที่เขียนส่งไปยัง Rabbit

รีจิสเตอร์ตัวแรก หรือรีจิสเตอร์ทั้ง 3 คู่มีความพิเศษในการเขียน สามารถ interrupt โปรเซสเซอร์ตัวอื่นๆ ได้ในจุดเชื่อมต่อของการติดต่อสื่อสารของ master-slave สายเอาท์พุทที่มาจากตัวลูกจะรักษาสถานะไว้ ในขณะที่ตัวลูกเขียนส่งไปยังรีจิสเตอร์ลูก 0 เส้นทางนี้อาจถูก interrupt โดยตัวแม่ได้ ระบบวงจรภายในซึ่งอยู่ในตัวลูกสามารถถูกกำหนดค่า interrupt ของตัวลูกในขณะที่ตัวแม่เขียนส่งไปยังรีจิสเตอร์ตัวลูกที่ 0

รีจิสเตอร์แสดงสถานะซึ่งมีค่าในทั้ง 2 ฝั่ง ได้ถูกเก็บไว้ในรีจิสเตอร์ทุกๆตัว และมีการรายงาน ถ้าความสามารถของ interrupt ได้ถูกร้องขอจากข้างใดข้างหนึ่ง ซึ่งรีจิสเตอร์แสดงสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เก็บค่า track ว่า "full-empty" เป็นสถานะของแต่ละรีจิสเตอร์ รีจิสเตอร์ถูกพิจารณาในขณะที่ข้างหนึ่งของการเชื่อมต่อ มันกลายเป็นสถานะว่าง ถ้าฝั่งอื่นอ่านมัน เส้นทางนี้สามารถที่ทดสอบได้ ถ้าเส้นทางอื่นมีการปรับเปลี่ยนรีจิสเตอร์ หรือฝั่งอื่นมีพื้นที่ในการเก็บข้อมูลลงรีจิสเตอร์

การเชื่อมการติดต่อของ Master-slave สามารถทำให้เป็นความจริง "Set and forget" อาศัยข้อตกลงที่ใช้ในการติดต่อสื่อสาร โดยแต่ละฝั่งสามารถทำให้เกิดการสั่งการ หรือการร้องขอโดยการเก็บข้อมูลในรีจิสเตอร์ และแล้วไปเกี่ยวกับการทำงานในขณะที่ทางด้านอื่นสามารถดูแลเกี่ยวกับการร้องขอ ตามตาราง ส่วนด้านอื่นสามารถเตือนโดย interrupt สามารถเก็บไว้ได้ในขณะที่ข้อมูลถูกเก็บไว้ในรีจิสเตอร์ 0

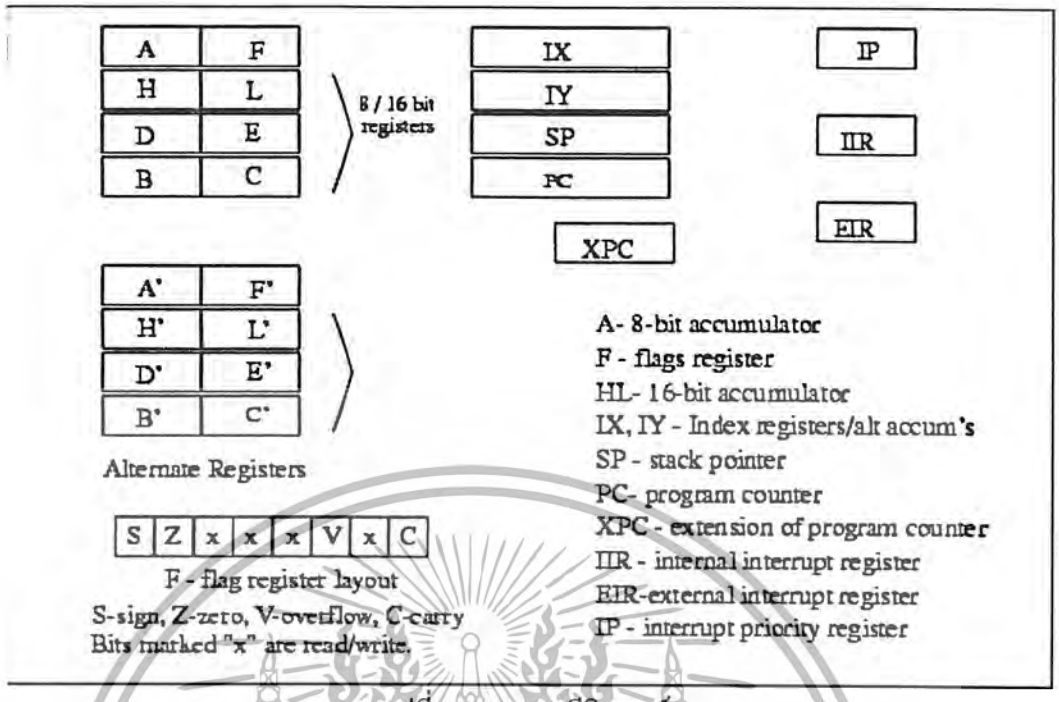
รีจิสเตอร์ทั้งสามตัว ถูกตรวจสอบโดยแต่ละด้าน สำหรับแต่ละด้านทางการติดต่อรีจิสเตอร์ตัวแรก รีจิสเตอร์ 0 มีฟังก์ชั่นพิเศษเพราะว่า interrupt สามารถสร้างขึ้นโดยการเขียนของรีจิสเตอร์ โดยมีสาเหตุมาจาก interrupt ย้ายไปฝั่งอื่น ถ้า interrupt สามารถทำได้ มีโปรโตคอลชนิดหนึ่งที่สามารถเก็บข้อมูลในรีจิเตอร์ 1,2 และขั้นสุดท้ายของการเก็บไว้ที่รีจิสเตอร์ 0 แล้ว 24 บิตของข้อมูลมีค่าส่งไปยังเส้นทางของ interrupt บนฝั่งตรงข้ามของการติดต่อ

ข้อมูลที่มีขนาดใหญ่จะถูกส่งข้ามการติดต่อนั้นต้องอาศัย Interrupt สำหรับการส่งทีละไบต์ เหมือนกับส่งไปยังพอร์ทอนุกรม หรือ UART ในกรณีการส่งแบบ Full-duplex สามารถส่งไปต่างๆ เหมือนกับ สามารถทำอะไรก็ได้ด้วย UART overhead สำหรับการส่งของ interrupt อาจจะมีการส่งประมาณ 100 clock ต่อไบต์ สมมติคำสั่ง 20 คำสั่งบนเส้นทางของ interrupt หลายวิธีที่สร้างความพอใจโดยส่งข้อมูล 2 ไบต์ ในแต่ละ interrupt ซึ่งจะมีค่าใกล้เคียงกับครึ่งหนึ่งของ overhead ถ้าพบอยู่ระหว่างกระบวนการ ข้อมูลสามารถส่งด้วยความเร็วประมาณ 25 clock ต่อไบต์ โดยแต่ละจุดรีจิสเตอร์แสดงสถานะกำลังรอสำหรับฝั่งอื่นๆ ที่อ่าน/เขียน ข้อมูลรีจิสเตอร์แล้วเขียน/อ่านรอบโดยฝั่งอื่น

2.3 Processor register

รีจิสเตอร์ของ Rabbit มีลักษณะใกล้เคียงกับ Z18 หรือ Z80 ซึ่งมีรีจิสเตอร์แบบใหม่เพิ่มขึ้นมาคือ XPC และ IP โดยที่มีรีจิสเตอร์ EIR ซึ่งยังคงเหมือนกับ Z180 ซึ่งถูกใช้บอกข้อมูลเกี่ยวกับเวกเตอร์ของอินเตอร์รัพสำหรับในการอินเตอร์รัพ ส่วนรีจิสเตอร์ IIR เกิดขึ้นที่ตำแหน่งเดียวกับ logical ในชุดคำสั่ง เหมือนกับ Z80 รีจิสเตอร์แต่มีบางฟังก์ชั่นชี้ไปยังตารางของอินเตอร์รัพเวกเตอร์สำหรับในการสร้างอินเตอร์รัพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 Rabbit รีจิสเตอร์

ตัวประมวลผลของ Rabbit มีตัวประมวลผล 2 ตัว คือ รีจิสเตอร์ A จะเก็บไว้ในการคำนวณ 8 บิต ในการจัดการ 8 บิต เช่น and หรือ or รีจิสเตอร์ HL ซึ่งมีขนาด 16 บิต จะถูกเก็บไว้สำหรับการประมวลผล 16 บิต สำหรับการทำงานหลายอย่าง IX และ IY สามารถทำงานแทนสำหรับ HL ได้

รีจิสเตอร์สามารถทำให้ F ซึ่งเป็นค่ารีจิสเตอร์หรือสถานะของรีจิสเตอร์จะมีค่าซึ่งเตรียมข้อมูลเกี่ยวกับการทำงานครั้งสุดท้ายค่ารีจิสเตอร์ไม่สามารถเข้าถึงได้ หน่วยความจำได้โดยตรง แต่งตั้งให้ใช้คำสั่ง POP AF และ PUSH AF โดยปกติแล้วค่าจะถูกตรวจสอบโดยคำสั่ง Jump Flag ซึ่งแสดงการทำงานด้านคณิตศาสตร์ และการทำงานด้านลอจิก ซึ่งรวมไปถึงกฎของแต่ละคำสั่งมี 4 บิตที่ไม่ได้ใช้ สำหรับอ่าน/เขียนค่าในรีจิสเตอร์ซึ่งจะเก็บค่าต่างๆ และเส้นทางของคำสั่ง PUSH AF และ POP AF แต่บิตควรใช้อย่างระมัดระวังตั้งแต่ Rabbit สามารถใช้บิตพวกนี้ในเรื่องอื่นๆด้วย

รีจิสเตอร์ IX, IY และ HL สามารถเก็บไว้ใน index ซึ่งเก็บค่าพอยเตอร์ที่อยู่ของหน่วยความจำสำหรับข้อมูลสามารถส่งข้อมูลและเก็บข้อมูล จนกระทั่ง Rabbit มีแอดเดรสจำนวนหลาย MB หรือมากพอสำหรับการเก็บข้อมูลอินเด็กรีจิสเตอร์สามารถเข้าถึงแอดเดรสได้โดยตรงประมาณ 64 k ของหน่วยความจำ ขอบเขตตำแหน่งจะมีผลมาจากอุปกรณ์การแมพพิ่งของหน่วยความจำและคำสั่งพิเศษบางคำสั่งสำหรับบางโปรแกรมใช้หน่วยความจำประมาณ 64 k ซึ่งเพียงพอต่อการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

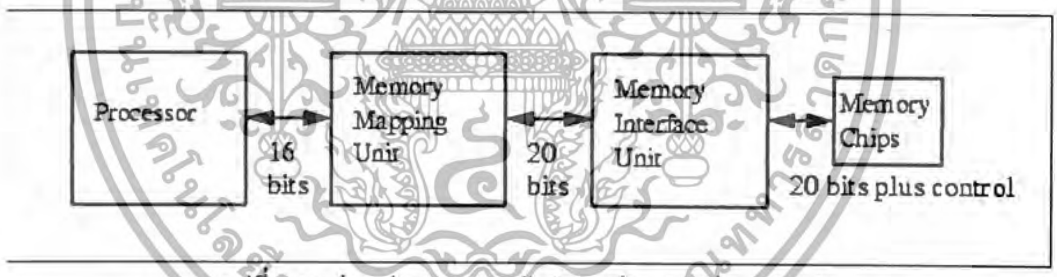
โดยรีจิสเตอร์ SP จะเป็นตัวชี้ stack ซึ่งใช้สำหรับการหาเส้นทางและการเชื่อมต่อการอินเตอร์รัพเช่นเดียวกับการเก็บข้อมูลทั่วไป

ข้อได้เปรียบของ Rabbit คือการมีรีจิสเตอร์ให้เลือกเป็นชุดคำสั่งพิเศษ 2 ชุดที่หน้าที่พิเศษในการสลับของรีจิสเตอร์ที่ถูกเลือก กับรีจิสเตอร์ทั่วไป การติดต่อระหว่างที่รีจิสเตอร์ถูกเลือกในสมัย Z80 นั้นเป็นไปได้ยากเพราะว่า คำสั่งการแลกเปลี่ยนมีไว้สำหรับค่ากลางที่เกี่ยวกับการติดต่อสื่อสารระหว่างรีจิสเตอร์ทั่วไปกับรีจิสเตอร์ที่ถูกเลือกเอาไว้โดยมีประสิทธิภาพมากขึ้นโดยการกำหนดตัวเลขให้แก่รีจิสเตอร์แต่ละตัวทำให้ง่ายต่อการเขียนโปรแกรม

2.4 Memory Mapping and Interface

ชุดคำสั่งของ Rabbit มีการติดต่อกับที่ว่างของข้อมูลโดยตรงซึ่งหมายความว่าที่อยู่ของคำสั่งจะมีความยาว 16 บิตและรีจิสเตอร์อาจจะใช้ในการชี้ตำแหน่งของข้อมูล, โปรแกรมเคาเตอร์และ SP ซึ่งมีความยาว 16 บิต

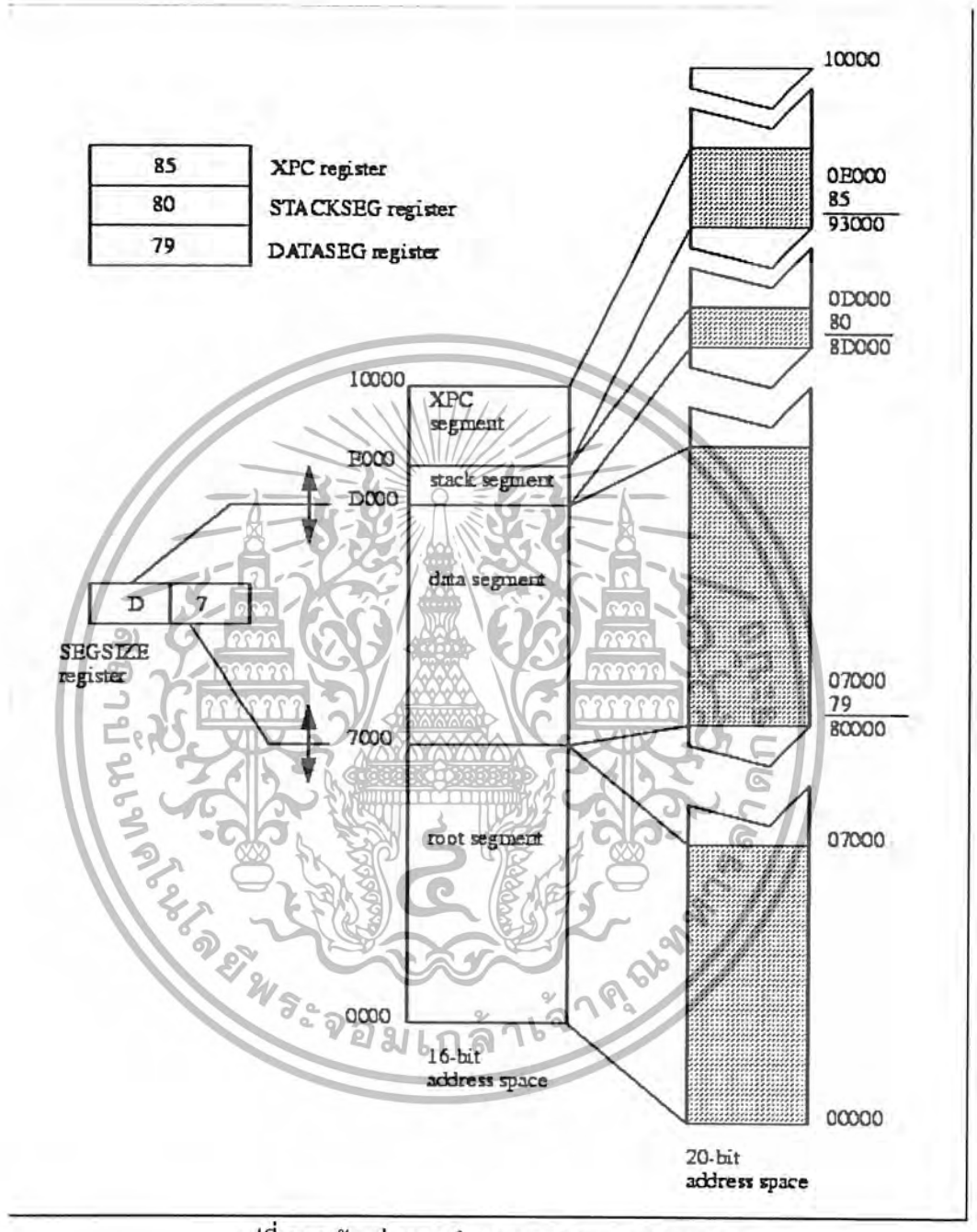
เพราะชุดคำสั่งของ Rabbit ใช้แอดเดรสที่มีความยาว 16 บิต ชุดคำสั่งที่สั้นกว่าและสามารถประมวลผลได้เร็วกว่า โดยที่ Rabbit มีการวางตำแหน่งของหน่วยความจำที่คล้าย Z180 แต่ประสิทธิภาพมากกว่า โดยในรูปที่ 2.11 จะแสดงความสัมพันธ์ระหว่างส่วนต่างๆ



รูปที่ 2.17 ส่วนประกอบการจัดตำแหน่งของหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถแบ่งหน่วยความจำออกเป็นส่วนๆ ได้ดังรูปที่ 2.18



รูปที่ 2.18 ตัวอย่างการทำงานของ Memory Mapping

Root segment เป็นการกำหนดตำแหน่งพื้นฐานของหน่วยความจำแบบแฟลชและบรรจุโค๊ดเริ่มต้น เช่นเดียวกับโค๊ดที่เกิดการเก็บข้อมูล

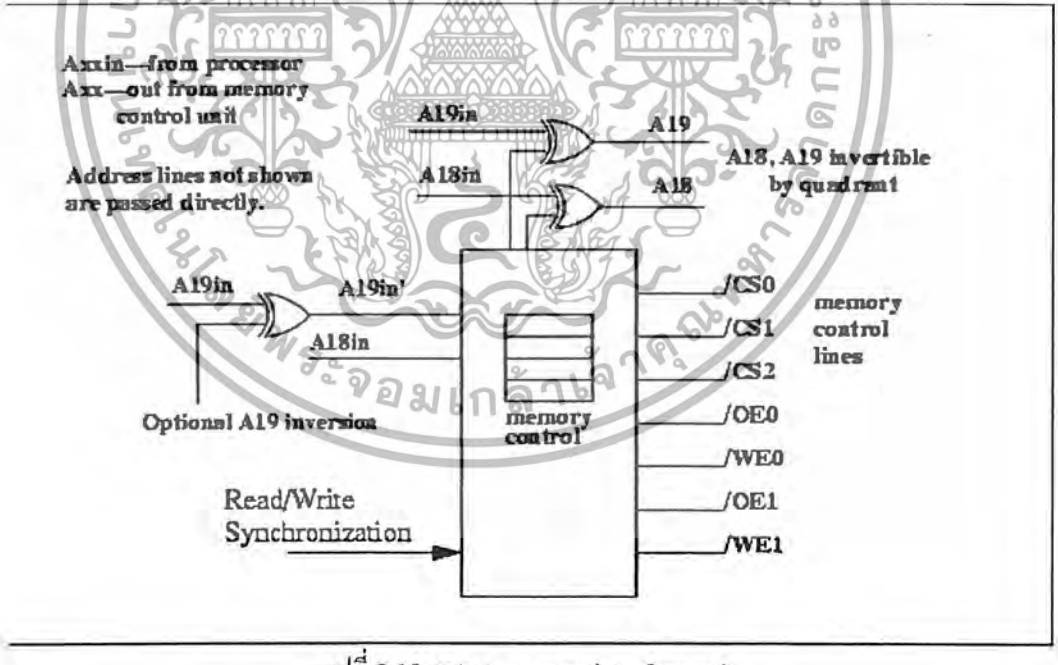
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data segment ใช้ในการเก็บหลายรูปแบบ โดยใช้ผลรวมของวิธีการต่างๆ สำหรับการตั้งค่าบนหน่วยความจำ

Stack memory เป็นส่วนที่เก็บข้อมูลของระบบ โดยแต่ละสะแตกจะมีขนาด 4 K

XPC Segment โดยปกติใช้เป็นการประมวลผลของโค๊ด ซึ่งไม่ได้เก็บข้อมูลใน root segment หรือ data segment ชุดคำสั่งพิเศษ จะได้รับการสนับสนุนจากการประมวลผลซึ่งได้เห็นใน XPC segment

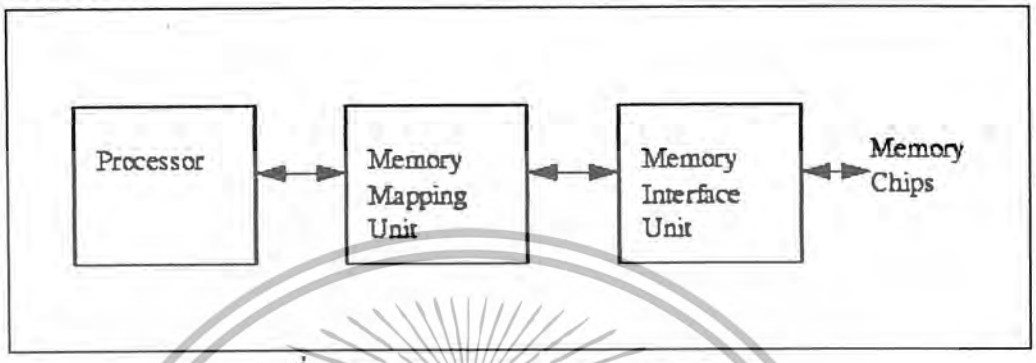
The memory interface unit จะได้รับข้อมูลขนาด 20 บิตซึ่งถูกสร้างจาก MMU มีเงื่อนไขในการเปลี่ยนที่แถว A16, A18, A19 แถวตำแหน่งอื่นๆ สามารถเข้าได้เลย memory interface unit จะควบคุมสัญญาณสำหรับข้อมูลภายนอก การเชื่อมต่อกับข้อมูลภายนอกโดยเลือก (/CS0, /CS1, /CS2), ข้อมูลที่ถูกส่งออก (/OE0, /OE1) และสามารถเขียนที่ (/WE, /WE1) แต่ละสัญญาณจะตอบสนองไปยัง แถวที่ควบคุมสามารถพบในหน่วยความจำแบบ static ซึ่งสร้างมาจากหน่วยความจำที่ควบคุมสัญญาณ 20 บิต ตำแหน่งจะถูกแบ่งเป็น ¼ ของ 256 K *bank control register* สำหรับแต่ละตัวเพื่อที่จะใช้ในการตัดสินใจว่าจะเลือกใช้ตัวไหน และจะยอมให้เอาท์พุทออกตัวไหนและจะให้ตัวไหนเขียนไปบนหน่วยความจำ



รูปที่ 2.19 แสดง memory interface unit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.8 เป็นการแสดงให้เห็นถึงการทำงานของหน่วยความจำ Rabbit ซึ่ง Memory Mapping Unit (MMU) ได้รับข้อมูลที่มีขนาด 16 บิต และแปลงข้อมูลเหล่านี้เป็นข้อมูลขนาด 20 บิต โดยที่ Mapping Interface จะรับข้อมูลขนาด 20 บิต และสร้างสัญญาณควบคุมโดยตรงไปยังชิพของหน่วยความจำ



รูปที่ 2.20 Overview of Rabbit Memory Mapping

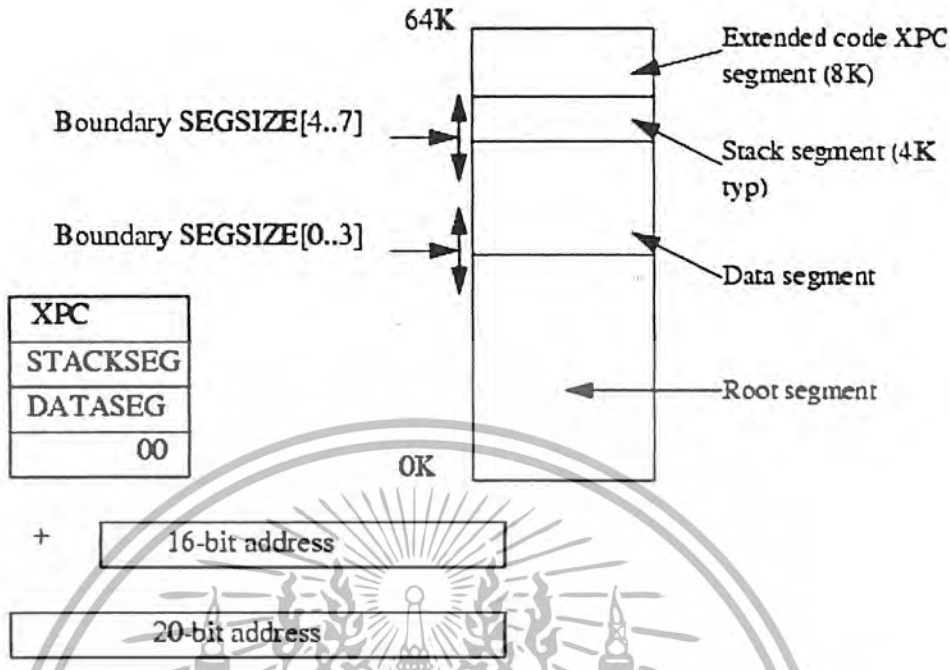
2.4.1 Memory Mapping Unit

โดยที่ข้อมูลขนาด 64 K มีขนาด 16 บิตเข้ามาโดยคำสั่งของ Processor ซึ่งจะถูกแบ่งออกเป็นส่วน โดยแต่ละส่วนจะมีขนาด 4 K ยกเว้นส่วนที่เป็น code ส่วนนี้จะมีการปรับขนาดและบางส่วนมีการเพิ่มขึ้นจากขนาดที่เป็นศูนย์ และบางส่วนได้ลบออกจากแอดเดรส

ข้อมูลแบ่งออกเป็น 4 ส่วน โดย Segment size register (SEGSIZE) กำหนดขนาดในไดอะแกรม ส่วนของการขยายรหัสมักจะเกิดขึ้นที่แอดเดรส 0E000h-0FFFFh โดยส่วนของสแตกมีการขยายมาจากแอดเดรสเฉพาะ โดยมาจากค่า 4 บิตบนของ SEGSIZE ไปยัง 0DFFFh ถ้าค่า 4 บิตบนของ SEGSIZE มากกว่าหรือเท่ากับ 0Eh ส่วนของสแตกจะหายไป ถ้าบิตเหล่านั้นถูกกำหนดให้มีค่าเท่ากับ 0 ข้อมูล 2 ส่วนที่อยู่ข้างล่างส่วนของสแตกก็จะหายไปด้วย

4 บิตล่างของ SEGSIZE ได้กำหนดขอบเขตล่างที่แสดงในรูป ถ้ามีขนาดเท่ากับขอบหรือมากกว่า 0Eh ส่วนของข้อมูลจะหายไป ถ้าส่วนของข้อมูลเริ่มต้นที่ 0 ส่วนของ code ก็จะหายไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 Memory Segments

MMU (Memory Management Unit) จะรับข้อมูลที่มีขนาด 16 บิตจาก Processor และเปลี่ยนเป็น 20 บิต โดยมีการทำงานดังนี้

1. นำข้อมูลที่มีขนาด 16 บิต โดยตรวจสอบ 4 บิตบนของแอดเดรส โดยทุกข้อมูลมีความเป็นไปได้ที่จะต้องเลือก 1 ใน 4 ส่วนนี้
2. โดยแต่ละส่วนจะเป็นรีจิสเตอร์ขนาด 8 บิตซึ่งจะนำมารวมเป็น 4 บิตบนของข้อมูล 16 บิตเพื่อที่จะทำให้ข้อมูลมีขนาด 20 บิต

ตารางที่ 2.13 Segment Registers

Segment Register	Function
XPC	Locates extended code segment in physical memory. Read and written by processor instructions: ld a,xpc, ld xpc,a, lcall, lret, ljp
STACKSEG = 11h	Locates stack segment in physical memory.
DATASEG = 12h	Locates data segment in physical memory.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.14 Segment Size Register

	Bits 7..4	Bits 3..0
SEGSIZE = 13h	Boundary address stack segment.	Boundary address data segment.

2.4.2 Memory Interface Unit

ข้อมูลขนาด 20 บิตจะถูกสร้างโดย MMU ส่งไปยัง Memory Interface Unit ซึ่งจะแยกเขียนควบคุมรีจิสเตอร์สำหรับ 256 K ซึ่งเป็นควอแรนท์แรกของ 1 M ของหน่วยความจำทางกายภาพ การควบคุมรีจิสเตอร์จะมีการระบุการร้องขอเข้าไปในหน่วยความจำที่ติดต่อกับ Rabbit โดยมีชิพ 3 แบบที่เลือกเส้นทางของเอาท์พุท (/CS0, /CS1, /CS2) สามารถเลือกใช้ชิพโดยเลือกจาก 3 ชนิดนี้ มีการกำหนดการเลือกชิพจะถูกเลือกสำหรับให้ข้อมูลผ่านไปยังควอแรนท์ ซึ่งชิพตัวเดียวสามารถส่งได้มากกว่า 1 ควอแรนท์

2.4.3 Memory Bank Control Register

ในตารางจะอธิบายถึงการทำงานของรีจิสเตอร์ที่คอยควบคุม Bank โดนเป็นรีจิสเตอร์ที่สามารถเขียนได้อย่างเดียว โดยแต่ละตัวรีจิสเตอร์จะควบคุม 1 ควอแรนท์ใน 1 M ของหน่วยความจำ

ตารางที่ 2.15 Memory Bank Control Register x (MBxCR=14h+x)

Bits 7,6	Bit 5	Bit 4	Bit 3	Bit 2	Bits 1,0
00--4 wait states 01--2 wait states 10--1 wait states 11--0 wait states	1--Invert address A19	1--Invert address A18	1--Write-protect memory this quadrant	0--use /OE0, /WE0 1--use /OE1, /WE1	00--use /CS0 01--use /CS1 1x--use /CS2

- บิตที่ 7,6 เป็นเลขที่บอกสถานะซึ่งจะใช้ในการเข้าถึงควอแรนท์ โดยมีสถานะการรอ การอ่าน ต้องการ 2 ช่วงเวลา และการเขียนต้องการ 3 ช่วงเวลา โดยสถานะการรอนี้จะถูกใช้ในการเข้าถึงข้อมูล ไม่ใช่สำหรับหน่วยความจำที่มาขากชุดคำสั่งได้ถูกวิเคราะห์มา
- บิตที่ 5,4 เป็นบิตที่ยอมให้แอดเดรสมีการสลับเปลี่ยนแปลง ซึ่งจะเกิดหลังจากมีการทำลอจิก โดยเลือก Bank รีจิสเตอร์ ดังนั้นการตั้งค่าที่บรรทัดนั้นจะไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลกระทบภายนอกกับควอแรนท์ซึ่งรีจิสเตอร์ควบคุมหน่วยความจำ Bank ควบคุมอยู่

- บิตที่ 3 มีการเขียนพัลส์เพื่อให้หน่วยความจำเข้าถึงควอแรนท์ ใช้ประโยชน์ในการป้องกันข้อมูลแบบแฟลชจากพัลส์ที่เขียนผิด
- บิตที่ 2 เลือกชุด /OEx และ /WEx เพื่อการเข้าถึงควอแรนท์
- บิตที่ 1,0 เป็นการตัดสินในการเลือกชิพจาก 3 ชิพ ซึ่งจะนำไปใช้ในการเข้าถึงควอแรนท์ โดยทุกๆ บิตของรีจิสเตอร์จะถูกทำให้เป็น 0 เมื่อมีการ reset

2.4.3.1 Optional A16, A19 Inversions by Segment (/CS1 Enable)

ในทางกลับกันของ A16 หรือ A19 ได้ถูกควบคุมให้มีการเขียนหรืออ่านรีจิสเตอร์ MMIDR ซึ่งถูกใช้ในการส่งค่ากลับของส่วนที่เป็น root และส่วนที่เป็นข้อมูล โดยการสลับค่าที่ในขณะที่ส่วนนั้นได้ถูกเข้าถึง

ข้อกำหนดต่างๆ ของ /CS1 มีค่าที่ใช้ได้สำหรับระบบที่ถูกดึงผ่านเวลาการเข้าของแบตเตอรี่สำรองของแรม โดยที่ทำให้ /CS1 ใช้ได้ การหน่วงเวลาของสวิทช์มีผลทำให้ /CS1 จะมีค่าสูงขึ้นเมื่อกำลังจะเลิกการใช้งาน โดยวิธีการบายพาสเป็นการเพิ่มพลังงานในการใช้ตั้งแต่แรมซึ่งสามารถใช้และควบคุมการเข้าควบคุมโดย /OE1

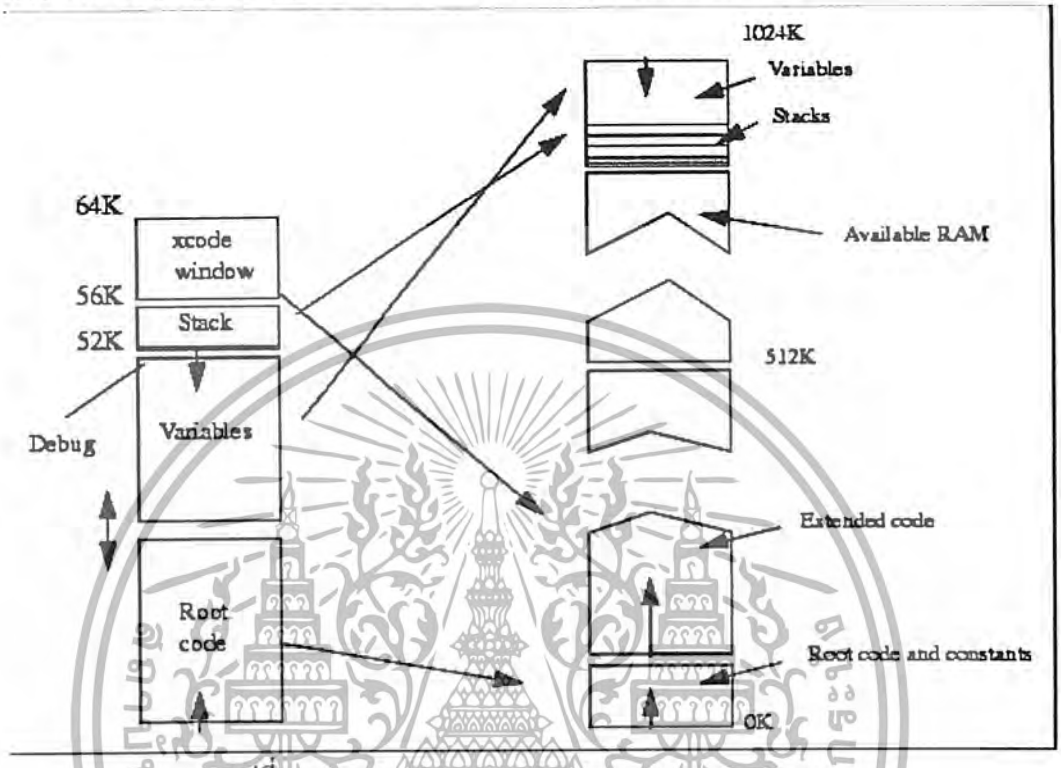
ตารางที่ 2.16 MMU Instruction/Data Register (MMIDR = 010h)

Bits 7,6,5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
000	1--force /CS1 to be always enabled	These bits must always be 0			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 Allocation of Extended Code and Data

Dynamic C จะประมวลผล code ส่งไปยังที่ว่างของ root code หรือที่ว่างของ extended code โดยที่ root code จะเริ่มทำงานที่ตำแหน่งที่ต่ำและประมวลผลต่อมา



รูปที่ 2.22 Typical Memory Mapping and Memory Usage

การกำหนดตำแหน่งของส่วนขยายโค๊ดจะเริ่มเหนือ root code และข้อมูล การเลือกตำแหน่งโดยปกติจะเริ่มและจบที่หน่วยความจำแบบแฟลช

ข้อมูลต่างถูกกำหนดจากการทำงานของแรมส่งกลับไปยังหน่วยความจำ โดยการติดตั้งเริ่มจากตำแหน่งที่ 52 K ใน 64 K แล้วทำต่อไปเรื่อยๆ 52 K จะเริ่มการแชร์การ root code และข้อมูล และได้ถูกกำหนดขึ้นมาจาก 0

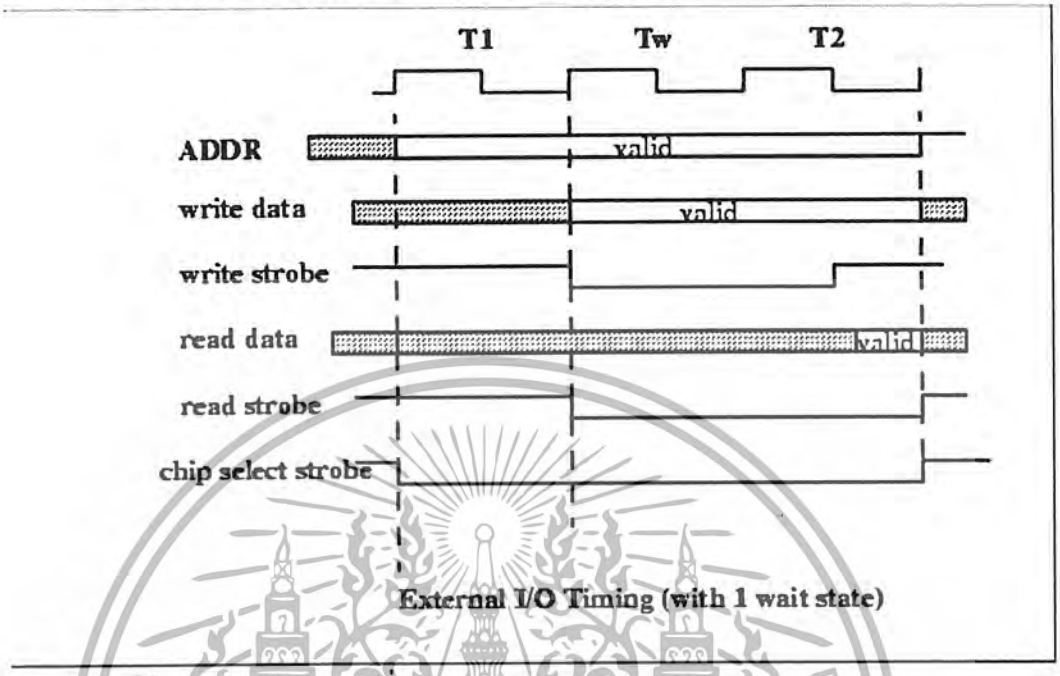
Dynamic C มีการรวมกันระหว่าง extended และข้อมูลที่คงที่ ซึ่งจะอยู่ในส่วนของแฟลช

2.5 I/O Bank Control Register

ขาของพอร์ท E สามารถแบ่งเป็น I/O strobe โดยแต่ละ 8 strobe จะมีรีจิสเตอร์ควบคุมโดยที่ควบคุม strobe และระยะเวลาคอยโดยใส่ค่าที่ขาลงไปในบัส I/O Cycle การเขียนจะถูกปิดบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ strobe อื่น ชนิดของ strobe สามารถดูได้ที่รูป แต่ละ 8 I/O strobe จะทำงานสำหรับตำแหน่งที่เกิดขึ้น 1/8 ของ 64 K พื้นที่ว่างของแอดเดรส I/O ที่อยู่ภายนอก



รูปที่ 2.23 External I/O Bus Cycles

ตารางที่ 2.17 I/O Bank Control Reg (adr IBxCR = 08xh)

Bits 7,6	Bits 5,4	Bit 3	Bits 2-0
Wait state code 11-1 10-3 01-7 00-15	/IX strobe type 00--chip select 01--read strobe 10--write strobe 11--or of read and write strobe	1--permit write 0--inhibit write	Ignored

8 I/O Bank Control Registers ได้กำหนดจำนวนของสถานะการรอ I/O ที่ประยุกต์การเข้าถึง I/O ภายนอกในพื้นที่ที่ถูกควบคุมจากแต่ละรีจิสเตอร์แม้ว่าเกี่ยวกับ strobe จะไม่สามารถใช้ได้

คอนโทรลเหนือการสร้างของสถานะการรออิสระของไม่ว่าหรือไม่เกี่ยวข้อง strobe ในพอร์ท E ทำงานอยู่ 2 บิตบนของแต่ละรีจิสเตอร์ได้กำหนดจำนวนของสถานะของสถานะรอ มี 4 อย่างให้เลือกคือ 1,2,7 และ 15 ในการ reset บิตที่ถูกเคลียร์ทำให้ได้ 15 สถานะการรอ มีอย่างน้อย 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นสถานะการรอ I/O และค่าที่ต่ำที่สุดของ I/O ภายนอกวัฏจักรอ่านใช้เวลา 3 สัญญาณนาฬิกา ห้ามการเขียนฟังก์ชั่นมีผลต่อพอร์ท E ที่เขียน strobe และสัญญาณ /IOWR

การควบคุมบิตจะมีผลกระทบกับช่องว่าง I/O ภายใน ซึ่งไม่มีรีจิสเตอร์ที่เกี่ยวข้องกับการอ่านหรือเขียน การเข้าถึง I/O ภายในหรือเขียนจะมีระยะเวลาเท่ากับ 2 สัญญาณนาฬิกา

I/O strobe ส่วนใหญ่ทำการเชื่อมต่อกับอุปกรณ์ภายนอกได้ง่าย มีการคืนค่า 5 บิตบนในแต่ละรีจิสเตอร์ ถูกเคลียร์ไป PPE(Parallel Port E) จะไม่เป็นเอาท์พุทส่งสัญญาณ ถ้าไม่ใช่บิตของรีจิสเตอร์ data-direction ถูกกำหนดสำหรับออกแบบตำแหน่งเอาท์พุท นอกจากนี้รีจิสเตอร์ฟังก์ชั่นพอร์ท E ต้องตั้งค่าที่ "1" สำหรับแต่ละตำแหน่ง

แต่ละรีจิสเตอร์ I/O Bank ได้ถูกเลือกโดย 3 บิตสำคัญมากที่สุดอยู่ I/O ขนาด 16 บิต ตารางจะแสดงความสัมพันธ์ระหว่างรีจิสเตอร์ควบคุม I/O และที่ว่างที่คล้ายคลึงกัน ซึ่งมีขนาด 64 K

ตารางที่ 2.18 External I/O Register Address Range and Pin Mapping

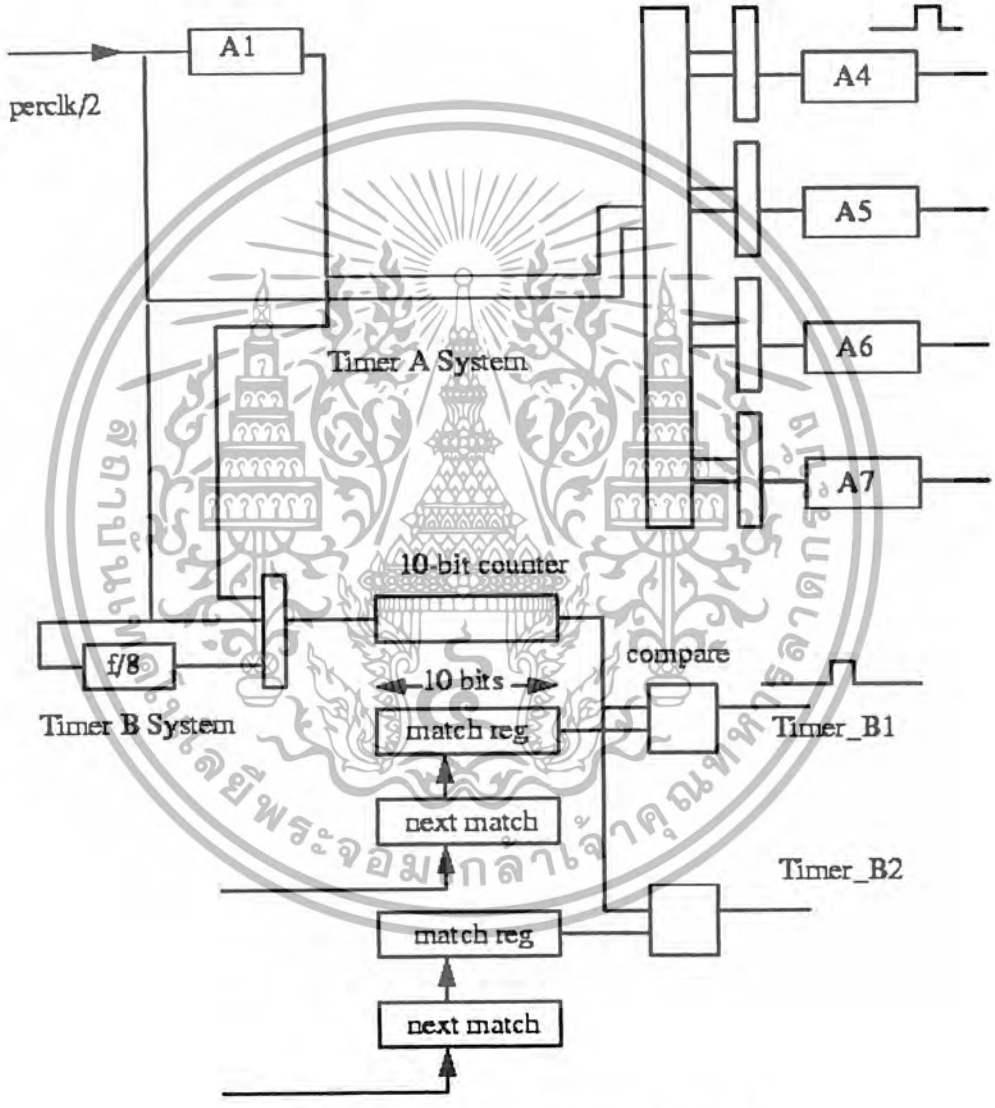
Control Register	Port E Pin	I/O Address A[15:13]	I/O Address Range
IB0CR	PE0	000	0x0000-0x1FFF
IB1CR	PE1	001	0x2000-0x3FFF
IB2CR	PE2	010	0x4000-0x5FFF
IB3CR	PE3	011	0x6000-0x7FFF
IB4CR	PE4	100	0x8000-0x9FFF
IB5CR	PE5	101	0xA000-0xBFFF
IB6CR	PE6	110	0xC000-0xDFFF
IB7CR	PE7	111	0xE000-0xFFFF

2.6 Timers

Rabbit มีระบบ Timer หลายระบบโดยมีสัญญาณอินเตอร์รัพคาบ ซึ่งมาจาก 32.768 MHz ออสซิลเลเตอร์ หารด้วย 6 จะมีอินเตอร์รัพทุก 448 ไมโครวินาที ถ้าเป็นไปได้ มีจุดมุ่งหมายตามความต้องการของสัญญาณอินเตอร์รัพ Timer A ประกอบด้วย 10 countdown ขนาด 8 บิต และสามารถรีโหลดรีจิสเตอร์ซึ่งสามารถแคสเคสขึ้น 2 ขั้น แต่ละครั้งรีจิสเตอร์ countdown สามารถตั้งค่าโดยแบ่งตามจำนวนเลขซึ่งอยู่ระหว่าง 1-256 เอาท์พุทจาก 6 ค่าไทม์เมอร์ซึ่งเตรียมค่าไว้ อัตราการส่งสัญญาณนาฬิกาสำหรับรีจิสเตอร์พอร์ทอนุกรมจำนวนหนึ่งจากรีจิสเตอร์ทั้งหมด เป็นสาเหตุที่ทำให้เกิดการอินเตอร์รัพและสัญญาณนาฬิกาไทม์เมอร์ชิงโครไนส์ พอร์ทเอาท์พุทขนาน Timer B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประกอบด้วย 10 บิตเคาเตอร์ ซึ่งไม่สามารถเขียนได้แต่อ่านได้ มี 2 รีจิสเตอร์โดยที่ ขนาด 10 บิต และเปรียบเทียบ ถ้า match รีจิสเตอร์สามารถเข้ากันเคาเตอร์และพัลส์เป็นค่าเอาท์พุทที่ออกมา ดังนั้นไทม์เมอร์สามารถถูกโปรแกรมไปยังเอาท์พุทที่เป็นพัลส์เพื่อจะตัดสินใจนับในอนาคต พัลส์ถูกสร้างมาจากสัญญาณนาฬิกาของไทม์เมอร์ซิงโครไนส์ รีจิสเตอร์เอาท์พุทพอร์ทขนาน เหมือนกับสาเหตุที่มาจากอินเตอร์รัพซึ่ง Timer B สะดวกในการสร้างและมีความเที่ยงตรงในเรื่องของเวลา เพราะว่าได้ถูกโปรแกรมส่วนควบคุมเอาไว้

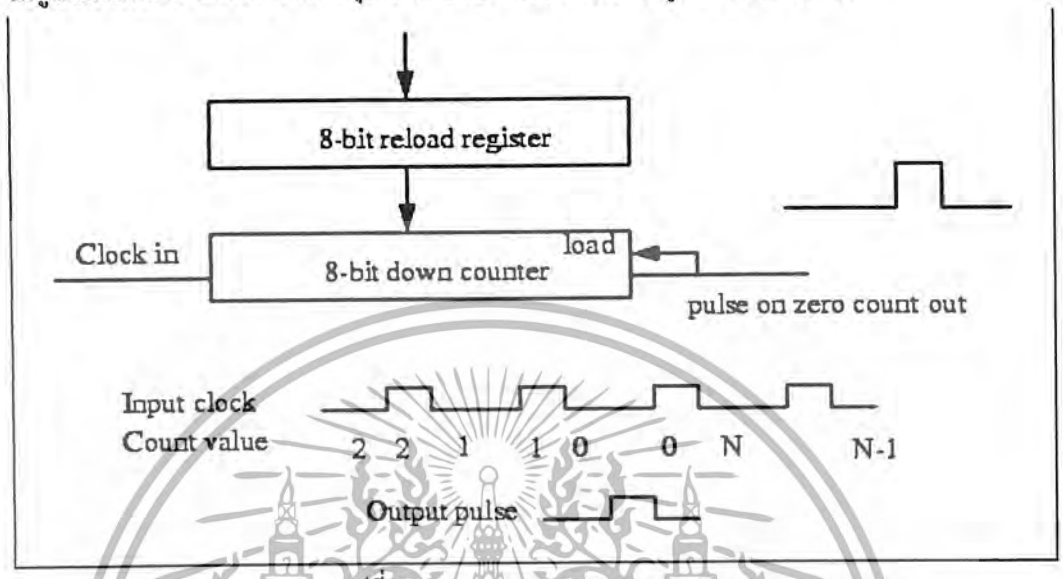


รูปที่ 2.24 Block Diagram of Timers A and B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 Timer A

Timer A ประกอบด้วย A1, A4, A5, A6 และ A7 มีขนาด 8 บิต รีจิสเตอร์นับถอยหลังแสดงในรูป มีรีโหลดรีจิสเตอร์ ซึ่งบรรจุเลขในระยะ 0-255 ตัวนับจะถูกหารโดย (N+1)



รูปที่ 2.25 Reload Register Operation

ระบบการทำงานของ Timers สามารถถูกควบคุมโดยสัญญาณนาฬิกาของอุปกรณ์เสริมหรือสัญญาณอุปกรณ์เสริมหาร 2 สัญญาณนาฬิกาโดยปกติจะสัญญาณนาฬิกาของหน่วยประมวลผลโดยหารด้วย 8 เอาท์พุทพัลส์จะเป็น 1 สัญญาณนาฬิกา สัญญาณนาฬิกาของตัวนับสามารถแทนที่บนขอบขาล่างของพัลส์ ในขณะที่ตัวนับจะมีค่าเท่ากับ "0" รีโหลดรีจิสเตอร์จะถูกโหลดพัลส์อินพุทต่อไปเพื่อแทนที่ของตัวนับที่จะดำเนินต่อไป รีโหลรีจิสเตอร์อาจจะถูกโหลดที่ระยะเวลาหนึ่งตั้งแต่สัญญาณนาฬิกาของอุปกรณ์เสริมถูกโหลดใหม่ synchronous กับสัญญาณนาฬิกาของตัวประมวลผล

Timer A4, A5, A6 และ A7 จะเตรียมความเร็วในการส่งสัญญาณนาฬิกาสำหรับพอร์ทอนุกรม พอร์ท A, B, C และ D ยกเว้นสำหรับถ้ามีค่าความเร็วต่ำ สัญญาณนาฬิกา A1 ไม่ต้องการที่ถูกใช้เป็นที่วัดอินพุทของสัญญาณนาฬิกาสำหรับ A4-A7 เช่น ถ้าสัญญาณนาฬิกาของระบบเท่ากับ 11.0592 MHz และ A4 หารด้วย 144 อะซิงโครนัสความเร็วในการส่งของ 2400 bps (โดยสมมุติ Timer ถูกสัญญาณนาฬิกา โดยนาฬิกาอุปกรณ์เสริมที่หารโดย 2) สัญญาณนาฬิกาอินพุทเพื่อพอร์ทอนุกรมมีค่า 8 หรือ 16 ครั้ง อัตราการส่งสำหรับอะซิงโครนัส อัตราความเร็วในการส่งเท่ากับ 11.0592 MHz สัญญาณนาฬิกาเท่ากับ 11,059,200/8 = 1,382,400

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ 5 countdown registers ใน timer A เงื่อนไขการนับสถานีปลายทางถูกรายงานในรีจิสเตอร์สถานะและถูกโปรแกรมในการสร้างอินเตอร์รัพ มีหนึ่งเป็นอินเตอร์รัพเว็กเตอร์ สำหรับ Timer A และลำดับก่อนหลังอินเตอร์รัพทั่วไป รีจิสเตอร์สถานะทั่วไป(TACSR) มีบิตสำหรับแต่ละ Timer จะแสดงถ้าเอาท์พุทพัลส์สำหรับ Timer นำออกมาตั้งแต่การอ่านครั้งสุดท้ายของรีจิสเตอร์สถานะ ในขณะที่รีจิสเตอร์สถานะทำการอ่าน บิตเหล่านั้นจะถูกลบ ไม่มีบิตที่เสียไป สิ่งใดสิ่งหนึ่งจะถูกอ่านโดยรีจิสเตอร์สถานะ อ่านหรือติดตั้งหลังจากที่รีจิสเตอร์สถานะอ่านเสร็จสมบูรณ์ ถ้าบิตบนและอินเตอร์รัพที่คล้ายกันทำงาน แต่อย่างไรก็ตามอินเตอร์รัพที่กระจายออกมาไม่ถูกต้องสำหรับแต่ละบิตกับอินเตอร์รัพที่เกิดขึ้น ถ้าบิตอ่านในสถานะรีจิสเตอร์ มันจะลบและไม่มีอินเตอร์รัพเพื่อบิตมาร้องขอ มันเป็นไปได้ที่บิตจะเป็นสาเหตุทำให้เกิดการอินเตอร์รัพ บิตที่เพิ่มเติมหนึ่งหรือมากกว่าจะถูกตั้งก่อนรีจิสเตอร์สถานะถูกอ่าน หลังบิตเหล่านี้ถูกลบ พวกมันไม่สามารถเป็นสาเหตุให้อินเตอร์รัพ บิตถ้ามีบนและอินเตอร์รัพที่คล้ายคลึงถูกทำให้ใช้ได้อินเตอร์รัพจะใช้สถานที่ทันทีที่ลำดับก่อนหลังยอม อย่างไรก็ตามถ้าบิตบนและอินเตอร์รัพที่คล้ายถูก reset

2.6.1.1 Timer A I/O Registers

ตารางที่ 2.19 Timer A I/O Registers

Register Name	Register Mnemonic	I/O address (hex)	R/W
Timer A Control/Status Register	TACSR	A0	R/W
Timer A Control Register	TACR	A4	W
Timer A1 Time Constant 1 Register	TAT1R	A3	W
Timer A4 Time Constant 4 Register	TAT4R	A9	W
Timer A5 Time Constant 5 Register	TAT5R	AB	W
Timer A6 Time Constant 6 Register	TAT6R	AD	W
Timer A7 Time Constant 7 Register	TAT7R	AF	W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.20 Timer A Control and Status Register (adr = 0A0h)

	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Read	A7 count done	A6 count done	A5 count done	A4 count done	0	0	A1 count done	This bit is write only.
Write	A7 interrupt enable	A6 interrupt enable	A5 interrupt enable	A4 interrupt enable	x	x	A1 interrupt enable	1--enable Timer A

บิตที่ 1, 4-7 ใช้สำหรับอ่าน/เขียน ตัวนับปลายทางจะถึง Timer A1 และ A4-A7 การอ่านที่รีจิสเตอร์สถานะจะทำการลบบิตเหล่านี้ (1 และ 4-7) การเขียนถึงบิตเหล่านั้นเป็นอินเตอร์รัพสำหรับ Timers ที่คล้ายกัน

บิต 0 ใช้สำหรับเขียน ตั้งค่า "1" เพื่อทำให้สัญญาณนาฬิกา (perclk/2) ใช้ได้สำหรับ Timer A ตั้งค่า "0" เพื่อยกเลิกสัญญาณนาฬิกา บิตที่ 1, 4-7 ถูกเขียนเพื่อทำให้เกิดอินเตอร์รัพใช้ได้เป็นเวลาคล้ายคลึงกัน

ตารางที่ 2.21 Timer A Control Register (adr = 0A4h)

Bit 7 A7	Bit 6 A6	Bit 5 A5	Bit 4 A4	Bits 3, 2	Bits 1, 0
Source A7 0-pclk/2 1-A1	Source A6 0-pclk/2 1-A1	Source A5 0-pclk/2 1-A1	Source A4 0-pclk/2 1-A1	not used ignored	00—Interrupt disabled 01—Enable priority 1 interrupt 10—Enable priority 2 interrupt 11—Enable priority 3 interrupt

2.6.2 Timer B

Timer B สามารถถูกขับโดยตรงโดย perclk/2, perclk/16 หรือเอาท์พุทของ Timer A Timer B มีการทำงาน 10 บิต ตัวนับจะเปรียบเทียบระหว่าง 2 รีจิสเตอร์ B1 match register และ B2 match register ในขณะเปลี่ยนแปลงตัวนับเพื่อค่าเท่ากับรีจิสเตอร์ พัลส์ภายในกับความยาว 1 สัญญาณนาฬิกาอุปกรณ์เสริมเกิดขึ้น พัลส์สามารถถูกใช้เพื่อสาเหตุอินเตอร์รัพ และ/หรือสัญญาณนาฬิกาเอาท์พุทรีจิสเตอร์ของพอร์ทขนาน D และ E

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

match register ถูกโหลดจาก match preload register ถูกเขียนโดยคำสั่ง I/O ดาต้าไบต์ใน match preload register ในขณะที่พัลส์มีรีจิสเตอร์ในการสร้าง

ถ้า match register ต้องการเปลี่ยนความต้องการไบต์สำคัญมากที่สุดเพื่อถูกเปลี่ยนครั้งแรก

ตารางที่ 2.22 Timer B Registers

Register Name	Register Mnemonic	I/O Address (hex)	R/W	On Reset To
Timer B Control/Status Register	TBCSR	B0	R/W	xxxxx000
Timer B Control Register	TBCR	B1	W	xxxxxx00
Timer B MSB 1 Reg	TBM1R	B2		x
Timer B LSB 1 Reg	TBL1R	B3	W	x
Timer B MSB 2 Reg	TBM2R	B4	W	x
Timer B LSB 2 Reg	TBL2R	B5	W	x
Timer B Count MSB Reg	TBCMR	BE	R	x
Timer B Count LSB Reg	TBCLR	BF	R	x

ตารางที่ 2.23 Timer B Control and Status Register (TBCSR) (adr = 0B0h)

Bits 7:3	Bit 2	Bit 1	Bit 0
Not used	1—A match with match register 2 was detected. This bit is cleared when this register is read; setting this bit to 1 enables the interrupt.	1—A match with match register 1 was detected. This bit is cleared when this register is read; setting this bit to 1 enables the interrupt.	1—Enable the main clock for this timer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.24 Timer B Control Register (TBCR)

Bits 7:4	Bits 3:2	Bits 1:0
Not used	00—Counter clocked by perclk/2 01—Counter clocked by output of timer A1 1x—Timer clocked by perclk/2 divided by 8	00—Interrupt disabled xx—Interrupt priority xx enabled.

ตารางที่ 2.25 Timer B MSB x Register (TBM1R/TBM2R = 0B2h/0B4h)

Bits 7:6	Bits 5:0
Two most significant bits of timer match preload register.	Not used.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ระบบเครือข่าย และ ความปลอดภัย

3.1 ประเภทของระบบเครือข่าย

ระบบเครือข่ายคอมพิวเตอร์ อาจนิยามได้ 2 ประเภทคือ

1. เครือข่ายทางกายภาพ (Physical Networks) หมายถึงสายและอุปกรณ์ที่ใช้ในกาเชื่อมต่อคอมพิวเตอร์ในระบบเครือข่าย อันได้แก่
   - Ethernet Wiring ซึ่งมีการเชื่อมต่อได้หลายแบบ เช่น thick coaxial cable (10BASE5) แบบ thin coaxial cable (10BASE2) และแบบ twisted pair (10BASE-T) หรือที่มักเรียกกันว่า UTP
   - สายใยแก้วนำแสง Optical Fiber (FDDI)
   - สายโทรศัพท์ทั้งแบบ Analog และ ISDN
   - สายเคเบิลใต้มหาสมุทร

   ซึ่ง Physical Networks ยังรวมไปถึงการเชื่อมต่อแบบที่มองไม่เห็นด้วยอีกด้วย เช่น
   - สัญญาณไมโครเวฟ
   - สัญญาณดาวเทียม
   - ระบบโทรศัพท์เคลื่อนที่
2. เครือข่ายเชิงตรรกะ (Logical Networks) เป็นเครือข่ายที่เกิดจากการสร้างความสัมพันธ์ระหว่างเครื่องคอมพิวเตอร์ที่อยู่บนเครือข่ายทางกายภาพ โดยความสัมพันธ์นั้นหมายถึง การทำงานร่วมกันอย่างใดอย่างหนึ่ง การมีจุดสนใจร่วมกัน การใช้ข้อมูลร่วมกัน หรือกิจกรรมใดๆ ที่กำหนดให้มนุษย์มีส่วนร่วม (ไม่จำเป็นว่าจะต้องเป็นคอมพิวเตอร์) เช่น
   - Internet
   - School Net
   - GINET (Government Information Networks)
   - UNINET (University Networks)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 องค์ประกอบของระบบเครือข่าย

จะต้องมี 3 ประการนี้จึงจะเป็นเครือข่ายคอมพิวเตอร์ได้

1. เครื่องคอมพิวเตอร์ที่อยู่บนระบบเครือข่าย
   - เครื่องคอมพิวเตอร์ PC / Macintosh
   - เครื่องคอมพิวเตอร์เวิร์คสเตชัน
2. Physical Media หรือสื่อเชื่อมต่อทางกายภาพอันได้แก่ สาย (Cable) และ Hub หรืออุปกรณ์เชื่อมต่อต่างๆ
3. ระเบียบพิธีการติดต่อสื่อสาร (Protocol) ก็คือระเบียบหรือข้อตกลง (rules) ที่ตั้งขึ้น เพื่อทำให้ผู้ที่จะสื่อสารกันเข้าใจกันและกัน ตัวอย่างเช่นสัญญาณธงที่ทหารเรือใช้สื่อสารกันเป็นต้น

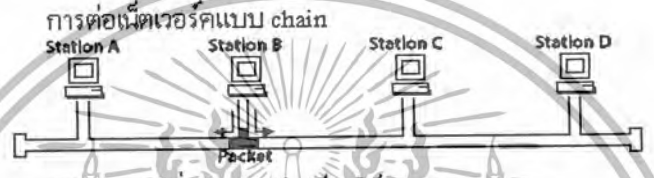
3.3 แบบของโปรโตคอล

1. Protocol ที่ใช้กับ Physical Networks ใช้ควบคุมสายเชื่อมต่อ บางทีเรียก Data Link เช่น
   - Ethernet Protocol ใช้ควบคุมการเชื่อมต่อด้วย Ethernet Wiring
   - FDDI ควบคุมการเชื่อมต่อแบบใยแก้วนำแสง
   - SLIP/PPP ควบคุมการเชื่อมต่อด้วยสายโทรศัพท์
2. Protocol ที่ใช้กับ Logical Networks ใช้ควบคุมการสื่อสารบน Logical Networks เช่น
   - TCP/IP ใช้กับการสื่อสารบนระบบอินเตอร์เน็ต/อินทราเน็ต/เอ็กซ์ทราเน็ต
   - IPX/SPX ใช้กับการสื่อสารบนเครือข่าย Novell Netware ของเครื่อง PC
   - NETBEUI ใช้กับการสื่อสารของเครือข่าย Microsoft Network
   - AppleTalk ใช้กับการสื่อสารระหว่างเครื่อง Apple Macintosh

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 LAN กับ Ethernet

รูปแบบของ LAN ในโลกนี้พอจะแบ่งออกได้เป็นระบบใหญ่ๆ ได้ 2 ระบบคือ Ethernet กับ Token Ring ระบบ Ethernet นั้นนิยมกันมากกว่าระบบ Token Ring มากๆ โดยระบบ Ethernet นั้นสามารถต่อได้ในแบบ Chain เรียกตามภาษาเทคนิคว่าแบบ 10Base-2 และ 10Base-5 กับแบบรูปดาว Star เรียกตามภาษาเทคนิคว่า 10Base-T เจ้าเลข 10 ข้างหน้านี้หมายความว่ามันสามารถส่งข้อมูลด้วยความเร็ว 10 Mbps (10 Megabits per second) เลข 2 และ 5 ข้างท้ายบอกว่าสายที่ใช้ต่อ LAN ในวงเดียวกันยาวได้อย่างมาก 200 (จริงๆแล้ว 180) และ 500 เมตรตามลำดับ ส่วนตัว T แปลว่าสายที่ใช้เป็นแบบ Twisted Pair ซึ่งยาวได้ไม่เกิน 100 เมตร



รูปที่ 3.1 การต่อเน็ตเวิร์คแบบ Chain

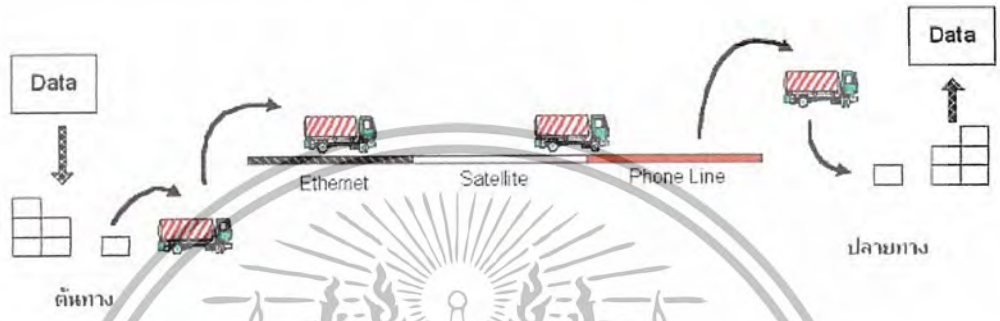
Token Ring คือ LAN ที่มีลักษณะต่อกันเป็นวงและมีตัวนำสารที่เรียกว่า Token วิ่งวนรอบๆ ไม่ค่อยมีใครเขาใช้กัน เพราะมีราคาแพงและต่อยาก เครื่องคอมพิวเตอร์ที่มาต่อกันในระบบ Ethernet นั้นจะต้องมี Ethernet Card ทำหน้าที่รับส่งสัญญาณในเครือข่าย ซึ่งแต่ละ Card จะมีหมายเลข Ethernet Address เฉพาะตัว ระบบ Ethernet นี้เมื่อต่อกับ Internet แล้วเวลามีใครนอกเครือข่าย ติดต่อมาหาเครื่องใดเครื่องหนึ่งในเครือข่าย Internet Address นั้น จะต้องถูกแปลงมาให้เป็น Ethernet Address ก่อนเพื่อการอ้างถึงให้ถูกว่าเป็นเครื่องใด โดยข้อมูลนั้นพร้อม Ethernet Address ปะไว้ที่หัวจะไหลผ่านเข้าไปในเครื่องคอมพิวเตอร์ทุกเครื่อง เครื่องใดก็ตามเมื่อทราบว่าข้อมูลนั้นเป็นของตัวก็จะหยิบออกไปเอง จากการที่ข้อมูลนี้ไหลผ่าน Network Card ของทุกเครื่อง ทำให้สามารถดังฟังข้อมูลได้ใน LAN เดียวกัน สามารถอ่าน e-mail หรือ อ่านหน้าจอใครก็ได้บน LAN เดียวกัน

3.5 TCP/IP กับเครือข่ายอินเตอร์เน็ต

เครื่องคอมพิวเตอร์บนเครือข่ายอินเตอร์เน็ต สื่อสารระหว่างกันโดยใช้ Transmission Control Protocol (TCP) และ Internet Protocol (IP) รวมเรียกว่า TCP/IP ข้อมูลที่ส่งจะถูกตัดออกเป็นส่วนๆ เรียก packet แล้วจ่าหน้าไปยังผู้รับด้วยการกำหนด IP Address เช่น สมมติเราส่ง e-mail ไปหาใครสักคน e-mail ของเราจะถูกตัดออกเป็น packet ขนาดเล็กๆ หลายๆ อัน ซึ่งแต่ละอันจะจ่าหน้าถึงผู้รับเดียวกัน packets พวกนี้ก็จะวิ่งไปรวมกับ packets ของคนอื่นๆ ด้วย ทำให้ในสายของข้อมูล packets ของเราอาจจะไม่ได้เรียงติดกัน packets พวกนี้จะวิ่งผ่าน ชุมทาง (gateway) ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

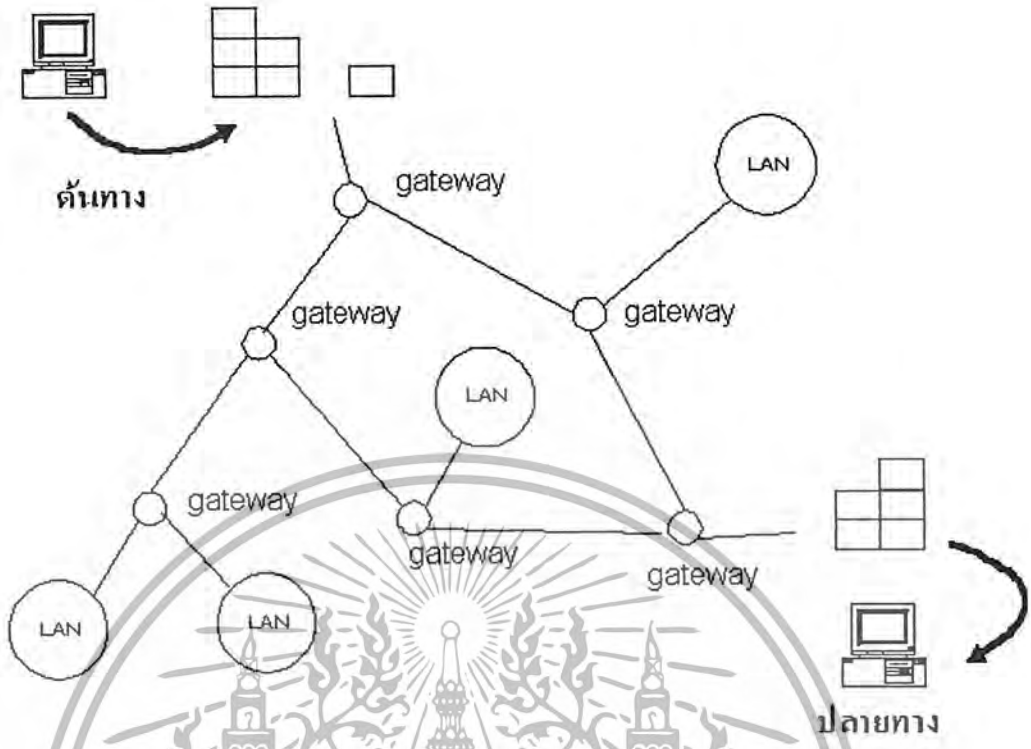
โดยตัว gateway (อาจเรียก router) จะอ่านที่อยู่ที่จ่าหน้า แล้วจะบอกทิศทางที่ไปของแต่ละ packet ว่าจะวิ่งไปในทิศทางไหน packet ก็จะวิ่งไปตามทิศทางนั้น เมื่อไปถึง gateway ใหม่ก็จะถูกกำหนดเส้นทางให้วิ่งไปยัง gateway ใหม่ที่อยู่ถัดไป จนกว่าจะถึงเครื่องปลายทาง เช่นเราติดต่อกับเครื่องในอเมริกา อาจจะต้องผ่าน gateway ถึง 10 แห่ง เมื่อ packet วิ่งมาถึงปลายทางแล้ว เครื่องปลายทางก็จะเอา packets เหล่านั้นมาเก็บสะสมจนกว่าจะครบ จึงจะต่อกลับคืนให้เป็น e-mail



รูปที่ 3.2 TCP/IP ตัดข้อมูลออกเป็น packet

การที่ข้อมูลมีลักษณะเป็น packet ทำให้ในสายสื่อสารสามารถที่จะ ขนส่งข้อมูลโดยไม่ต้องจอง (occupies) สายไว้สายจึงสามารถใช้ร่วมกันกับข้อมูลที่ส่งจากเครื่องอื่นได้ ต่างจากโทรศัพท์ที่ขณะใช้งาน จะไม่มีใครใช้สายได้ ดังตัวอย่างในรูปข้างล่างนี้ เครื่องคอมพิวเตอร์ A และ C สื่อสารกันด้วย packet สีดำ ซึ่งใช้สายร่วมกับเครื่องคอมพิวเตอร์อื่น ๆ ซึ่ง packet ดังกล่าวอาจจะเป็นสัญญาณเสียง (เช่น Internet Phone) ซึ่งเมื่อ packet เดินทางมาถึงก็จะถูกจับมารวมกันให้เป็นเสียงของการพูดคุย ไม่เหมือนโทรศัพท์แบบปรกติ ที่ขณะใช้งานสาย จะไม่สามารถนำไปทำงานอื่น ๆ ได้อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 การรับส่งข้อมูลผ่าน Gateway

3.6 IP Address

เครื่องคอมพิวเตอร์ที่ต่ออยู่บนอินเตอร์เน็ตก็เปรียบคล้ายๆ กับเครื่องโทรศัพท์ที่มีเบอร์เฉพาะตัว ซึ่งก็จะมีเพียงเบอร์เดียวในโลก เช่นเครื่อง Einstein ซึ่งเป็น Internet Server ของภาควิชาฟิสิกส์ มี IP Address เป็น 202.28.156.98 ตัวเลขที่เป็น IP Address เป็นตัวเลขขนาด 32 บิต แบ่งออกเป็น 4 ชุดๆ ละ 8 บิต ดังนั้นตัวเลข 1 ชุดที่เราเห็นคั่นด้วยจุดนั้น จึงแทนได้ด้วยตัวเลขจาก 0 ถึง 255

ตัวเลข 4 ชุดนี้จะถูกแบ่งออกเป็น 2 ส่วน คือ network number และ ส่วนของ host number โดยขนาดของแต่ละส่วนจะใหญ่หรือเล็กขึ้นอยู่กับว่าเครื่องคอมพิวเตอร์นั้นอยู่ในเน็ตเวิร์ค class ใด ซึ่ง class ของเน็ตเวิร์คแบ่งออกเป็น 4 classes ดังนี้

1. **Class A** เป็นเน็ตเวิร์คขนาดใหญ่ มี network number ตั้งแต่ 1.0.0.0 ถึง 127.0.0.0 นั่นคือใน class นี้นั้น จะมีส่วนของ host number ถึง 24 บิตซึ่งอนุญาตให้มีจำนวนเครื่องได้ 1.6 ล้านเครื่องใน 1 เน็ตเวิร์ค ซึ่งจะมีเน็ตเวิร์คขนาดใหญ่แบบนี้ได้เพียง 127 เน็ตเวิร์คเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. **Class B** เป็นเน็ตเวิร์คขนาดกลาง มี network number ตั้งแต่ 128.0.0.0 ถึง 191.255.0.0 นั่นคือใน class นี้มีส่วนของ network number 16 บิต และส่วนของ host number ได้ 16 บิต ทำให้มีจำนวนของเน็ตเวิร์คได้ถึง 16320 เน็ตเวิร์ค และ 65024 hosts
3. **Class C** เป็นเน็ตเวิร์คขนาดเล็ก มี network number ตั้งแต่ 192.0.0.0 ถึง 223.255.255.0 นั่นคือใน Class นี้มีส่วนของ network number 24 บิต และ ส่วนของ host number 8 บิต ทำให้มีจำนวนของเน็ตเวิร์คได้ถึง 2 ล้านเน็ตเวิร์คและมีจำนวน host ในแต่ละเน็ตเวิร์คเท่ากับ 254 hosts
4. **Class D** เป็นส่วนที่เก็บรักษาไว้สำหรับใช้ในอนาคต มี IP Address ตั้งแต่ 224.0.0.0 ถึง 254.0.0.0

3.7 Domain Name System (DNS)

เบอร์ IP Address เป็นตัวเลขที่ใช้ไม่ค่อยสะดวกและจำยาก ด้วยเหตุนี้จึงมีการคิดระบบตั้งชื่อแบบที่เป็นตัวอักษร ให้มีความหมายเพื่อการจดจำได้ง่ายกว่ามาก เวลาเราอ้างถึงเครื่องใดบนอินเตอร์เน็ต เราก็จะใช้ชื่อ DNS เช่น www.nectec.or.th แต่ในการใช้งานจริงนั้นเครื่องคอมพิวเตอร์ที่เราใช้อยู่ เมื่อรับคำสั่งจากเราแล้ว เค้าจะขอ (request) เครื่องคอมพิวเตอร์ที่ทำหน้าที่บริการบอกเลขหมาย IP Address (ทำหน้าที่คล้ายสมุดโทรศัพท์ Yellow Pages) ซึ่งเรียกกันว่าเป็น DNS Server หรือ Name Server ตัว Name Server เมื่อได้รับ request ก็จะตอบเลขหมาย IP Address กลับมาให้ เช่น สำหรับ www.nectec.or.th นั้นจะตอบกลับมาเป็น 164.115.115.9 จากนั้นเครื่องคอมพิวเตอร์ของเราจึงจะเริ่มทำการติดต่อ กับคอมพิวเตอร์เป้าหมาย ซึ่งมันก็จะผ่านกระบวนการแบบที่กล่าวไปข้างต้น คือแบ่งข้อมูลออกเป็น packet จ่าหัวด้วย IP จากนั้นส่ง packet ไปซึ่งก็จะวิ่งผ่าน gateway ต่างๆ มากมายไปยังเป้าหมาย

บางทีเราจะพบกรณีที่คอมพิวเตอร์ที่เป็น Name Server นั้นไม่ทำงาน เราจะไม่สามารถติดต่อเครื่องอื่นบนอินเตอร์เน็ตได้อีกต่อไปโดยใช้ชื่อ DNS หากเราทราบ IP Address เราสามารถใช้ IP Address ได้ตรงๆ ทำให้เราไม่จำเป็นต้องพึ่งสมุดโทรศัพท์ของ Name Server ด้วยเหตุนี้เราจึงทำการเก็บชื่อและ IP Address ไว้ในสมุดโทรศัพท์ส่วนตัวประจำเครื่อง เช่นบนระบบยูนิกซ์มีไฟล์ /etc/hosts เอาไว้เก็บชื่อ DNS ที่ใช้บ่อยๆ

ระบบการตั้งชื่อ DNS นั้นคล้ายกับระบบไปรษณีย์ โดยมีประเทศอยู่หลังสุด เช่น .th คือประเทศไทย .de คือประเทศเยอรมัน .uk คือ ประเทศสหราชอาณาจักร แต่สำหรับสหรัฐอเมริกานั้นยกเว้น จากนั้นจะแบ่งเครือข่ายออกเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- .edu หรือ .ac เครือข่ายมหาวิทยาลัย หรือ สถาบันการศึกษา
- .com หรือ .co เครือข่ายบริษัท ห้างร้าน
- .mil เครือข่ายทางการทหาร
- .org หรือ .or เครือข่ายองค์การที่ไม่หวังผลกำไร (พรรคการเมืองไทยก็ใช้ระบบนี้)
- .gov หรือ .go เครือข่ายหน่วยงานของรัฐบาล
- .net หรือ .net เครือข่ายของผู้ดูแลเน็ตเวิร์ค หรือ เจ้าของเน็ตเวิร์ค

3.8 SLIP และ PPP

เราคงเคยได้ยินคำ 2 คำนี้มาบ้าง โดยเฉพาะเมื่อต้องการต่ออินเตอร์เน็ตผ่าน MODEM และสายโทรศัพท์ โดยปรกติแล้ว TCP/IP นั้นเปรียบเสมือนรถบรรทุกสินค้าที่ใช้ขนข้อมูลในรูป packet ไปยังที่ต่างๆ ซึ่งรถบรรทุกนั้นสามารถวิ่งบนถนนทั้งแบบลาดยาง แบบคอนกรีต หรือแบบลูกรัง ในสภาวะที่เป็น LAN นั้น TCP/IP วิ่งอยู่บน Frame ที่เป็น Ethernet ซึ่งมารองพื้น เปรียบเสมือนถนนให้รถบรรทุกวิ่ง

บนสายโทรศัพท์ก็เช่นเดียวกัน ต้องมีระบบที่มารองพื้นเพื่อให้ TCP/IP สามารถวิ่งได้ ตัว Protocol ที่ใช้ก็จะมีความแตกต่างจาก Ethernet เพราะคนละ medium กัน ซึ่งก็มีอยู่ 2 แบบ คือ Serial Line Internet Protocol (SLIP) และ Point to Point Protocol (PPP) ตัวโปรโตคอลทั้ง 2 นี่คล้ายกัน เพียงแต่ SLIP เกิดจากการทดลองแล้วพัฒนามาเป็น PPP ซึ่งมีมาตรฐานกว่า

การต่อ Windows 95 เข้ากับอินเตอร์เน็ตนี่ สามารถต่อได้ทั้งเข้ากับ LAN โดยใช้ Network Card และต่อแบบ Dial-Up ซึ่ง Windows 95 ก็มี PPP ให้ใช้อยู่แล้ว หรือจะต่อทั้ง 2 อย่างในขณะเดียวกันก็ได้ ซึ่งเครื่องสามารถแยกแยะได้ว่าหากเราใช้ LAN ก็จะให้ packet เดินผ่าน Network Card แต่หากติดต่อข้างนอก ก็จะผ่าน MODEM แทน ซึ่งผมก็ใช้งานแบบนี้อยู่ ค่อนข้างดีไม่มีปัญหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 จุดเด่นของโพรโตคอล TCP/IP

1. สามารถนำส่งข้อมูลไปถึงจุดหมายได้แม้เส้นทางบางที่เสียหาย : เป็นจุดประสงค์หลักที่ช่วยให้ทนต่อความล้มเหลว โดยหากระหว่างการสื่อสารข้อมูลและมีเส้นทางใดเสียหายหรือล้มเหลว IP เน็ตเวิร์กจะปรับใช้เส้นทางอื่นที่ทดแทนได้เพื่อนำส่งข้อมูลให้ไปถึงปลายทางอย่างอัตโนมัติ ผู้ส่งและผู้รับข้อมูลไม่จำเป็นต้องรับรู้หรือปรับตัวแต่ประการใด

2. ไม่ขึ้นกับแพลตฟอร์มใด ๆ : ไม่ว่าเครือข่ายนั้นเป็นเครือข่ายท้องถิ่นหรือเครื่อข่ายระหว่างภูมิภาค เป็นไฟล์/พรินต์เซิร์ฟเวอร์หรือไคลเอ็นต์/เซิร์ฟเวอร์ เป็นระบบปฏิบัติการใด เน็ตเวิร์กอินเทอร์เฟซเป็นแบบใดก็ตาม ในมุมมองของโพรโตคอล TCP/IP ก็คือ IP เน็ตเวิร์ก

3.10 จุดอ่อนของ IP

1. รับส่งโดยไม่มีการรักษาความปลอดภัยเนื้อข้อมูล : การรับส่งข้อมูลด้วย IP แพ็กเก็ตไม่มีทั้งการเข้ารหัสข้อมูลและป้องกันการปลอมแปลงใด ๆ การไม่เข้ารหัสข้อมูลอาจทำให้ผู้ไม่ประสงค์ดีระหว่างเส้นทางที่ IP แพ็กเก็ตผ่านดักลอบดูเนื้อข้อมูลอย่างง่ายดาย แม้ว่าเราอาจสามารถบังคับเส้นทางของ IP แพ็กเก็ตได้ก็ไม่อาจมั่นใจได้ว่าระหว่างทางมีการดักลอบดูหรือไม่

ในเรื่องปัญหาการปลอมแปลงแบ่งออกเป็นสองกรณีคือ การปลอมแปลงหรือดัดแปลงเนื้อข้อมูล และการปลอมแปลงส่วนหัวของ IP แพ็กเก็ต ทั้งสองกรณีให้ผลเหมือนกันคือผู้รับได้ข้อมูลที่ผิดจากความเป็นจริง ทว่าจุดประสงค์ต่างกัน หากเป็นกรณีแรกนั้น ผู้ไม่หวังดีต้องการหลอกหรือกลั่นแกล้งให้ได้ข้อมูลผิด ๆ หากเป็นกรณีหลัง ผู้ไม่หวังดีต้องการแอบอ้างว่าข้อมูลนั้นมากจากแหล่งที่ผู้รับไว้ใจหรือแหล่งอื่นที่กลายเป็นเหยื่อของการแอบอ้างโดยไม่รู้ตัว

2. รับส่งโดยไม่คำนึงถึงคุณภาพการให้บริการ : การรับส่งต่อ IP แพ็กเก็ตระหว่างเครือข่ายย่อยไปเป็นทอดนั้นใช้หลักการใครมาก่อนได้ก่อน ฉะนั้นจึงคาดเดาไม่ได้ว่าข้อมูลที่นำส่งไปจะไปถึงปลายทางเมื่อใด แม้ว่า IP เน็ตเวิร์กใช้หลักการเลือกเส้นทางที่เหมาะสมที่สุดในขณะนั้นก็ตาม หากแต่ความเหมาะสมนั้นผู้ส่งและผู้รับไม่อาจคาดการณ์หรือมีส่วนร่วมตัดสินใจได้เลยว่าจะช้าเร็วหรือมีโอกาสที่ข้อมูลผิดพลาดมากน้อยเพียงไร

เมื่อมีปัญหาย่อมมีทางแก้ไข สำหรับเรื่องการรับส่งโดยไม่มีการรักษาความปลอดภัยเนื้อข้อมูลนั้น องค์กรกลางของอินเทอร์เน็ตได้ออกมาตรฐานที่ช่วยแก้ไขปัญหานี้คือ IPSec โดยมีทั้งการเข้าและถอดรหัสเนื้อข้อมูลในระดับ IP แพ็กเก็ตการตรวจสอบความถูกต้องเนื้อข้อมูลและการพิสูจน์ตนของ IP แพ็กเก็ตเพื่อป้องกันการปลอมแปลง

ส่วนเรื่องการรับส่งโดยไม่คำนึงถึงคุณภาพการให้บริการ ตามจริงแล้วมาตรฐานแต่ต้นของ IP เน็ตเวิร์กมีการจัดลำดับความสำคัญของ IP แพ็กเก็ต โดยเป็นแฟล็ก Type of Service สองกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มแรกเป็นเลขลำดับความสำคัญยิ่งมีค่ามากยิ่งสำคัญมาก (ใช้แนวคิดไพรออริตี้คิว) กลุ่มที่สองเป็นประเภทของงานที่ใช้ IP แพ็กเก็ตนั้นมีสามประเภทคือ งานที่มีความล่าช้าสูง งานที่ต้องการทรูพุตสูงและงานที่ต้องการความน่าเชื่อถือสูง แต่กลับไม่มีใครใส่ใจแฟล็กลำดับความสำคัญเพื่อพิจารณาส่งต่อ IP แพ็กเก็ต หาทางแก้ไขมีสองแนวทางคือให้เป็นหน้าที่ของระดับดาต้าลิงก์ (ระดับล่างของระดับเน็ตเวิร์ก) หรือส่งเสริมให้การส่งต่อ IP แพ็กเก็ตพิจารณาแฟล็กลำดับความสำคัญดังกล่าว

เหตุที่อินเทอร์เน็ตใช้รูปแบบของโพรโตคอล IP ก็เพราะว่า ณ เวลาที่มีการออกแบบนั้นระบบเครือข่ายยังมีความเร็วต่ำ และมีสัญญาณรบกวนมาก อีกทั้งการใช้งานในขณะนั้นก็มีเพียงอีเมล์ FTP, Usenet News และการใช้งานออนไลน์แบบเช่น Telnet ซึ่ง ล้วนแล้วแต่อยู่ในรูปแบบของตัวษรเท่านั้น (Text Mode) ดังนั้นเพียงเท่านี้ ก็เพียงพอกับการทำงาน ต่อมาเมื่ออินเทอร์เน็ตได้ขยายการเชื่อมโยงออกไปทั่วโลก ข้อจำกัดที่สำคัญที่เกิดขึ้นจากการออกแบบในยุคเริ่มต้น ก็ก่อปัญหาขึ้น

เมื่อมีการออกแบบและสร้างไฮเปอร์เท็กซ์ (HTML) ขึ้น ทำให้มีเครือข่าย WWW ที่มีการประยุกต์รูปภาพ เสียง และวีดีโอ บนเครือข่าย ทำให้การใช้งานบนระบบต้องการความเร็วและแบนด์วิดธ์มีความต้องการเพิ่มสูงขึ้น รวมถึงความต้องการในเรื่องคุณภาพ หรือ Qos (Quality of Service) ซึ่งหากต้องการส่งข้อมูลด้านเสียง หรือ ภาพ สำหรับโทรทัศน์วิทยุ หรือ โทรศัพท์แล้ว ก็ต้องการส่งข้อมูลด้านเสียงหรือภาพ สำหรับโทรทัศน์ วิทยุ หรือโทรศัพท์แล้ว ก็ต้องการคุณภาพความคมชัด และชัดเจนในระดับที่ยอมรับได้ ไม่ติดขัด มีการนำเอาเครือข่ายอินเทอร์เน็ตมาประยุกต์ใช้งานต่าง ๆ อีกมากมาย ตั้งแต่ระบบ Online ระบบ E-Commerce ระบบการศึกษาเช่น Vitual ClassRoom หรืองานสอนทางไกล การทำ Video Conference หรือ การประชุมทางไกล ซึ่งการประยุกต์เหล่านี้ ล้วนต้องการความเร็วในการรองรับ และ Qos ที่สูงมาก ๆ โดยเฉพาะในเรื่องความเร็วที่จะต้องเพิ่มจาก 10 เป็น 100 เป็น 1,000 ระดับกิกะบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การออกแบบระบบ

ในบทนี้จะกล่าวถึงการทำงานของ Rabbit และการออกแบบในการนำไปประยุกต์ใช้ให้ทันกับสมัยเทคโนโลยีในปัจจุบันให้เหมาะกับการใช้งาน โดยสามารถที่จะนำไปประยุกต์ใช้งานในชีวิตประจำวันได้จริงโดยจะนำ Rabbit มาเป็นอุปกรณ์ในการควบคุมเครื่องใช้ไฟฟ้าภายในบ้านโดยผ่านระบบอินเตอร์เน็ต จะมีการแจ้งสถานะให้ผู้ใช้ได้ทราบว่าขณะนี้ สถานะของอุปกรณ์ไฟฟ้าที่เราควบคุมนั้นอยู่ใน สถานะปิด หรือ เปิดการใช้งานอยู่ ซึ่งจะทำให้เราสามารถควบคุม ได้โดยผ่านระบบอินเตอร์เน็ต ได้ซึ่งจะทำให้สะดวก และง่ายต่อการตรวจสอบซึ่งจะทำให้เราสามารถประหยัดไฟฟ้าได้ โดยไม่จำเป็น โดยจะมีหลักการทำงานโดยคร่าวๆ ได้ดังนี้

ในส่วนการทำงานของ Rabbit ในการใช้ควบคุม จะแบ่งการทำงานออกเป็น 2 ส่วน

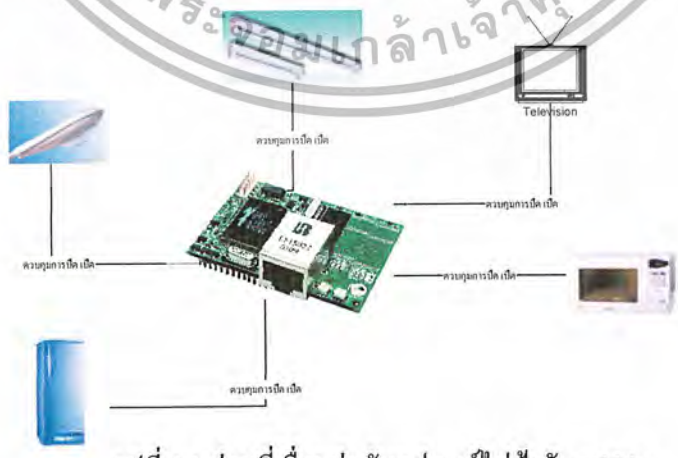
1. ส่วนที่เชื่อมต่อกับอุปกรณ์ไฟฟ้า
2. ส่วนที่เชื่อมต่อกับระบบอินเตอร์เน็ต

ซึ่งจะมีรายละเอียดดังต่อไปนี้

1. ส่วนที่เชื่อมต่อกับอุปกรณ์ไฟฟ้า

ซึ่งในส่วนนี้คือส่วนที่ Rabbit เชื่อมต่อกับอุปกรณ์ฟ้า หรือ โหลด โดยให้ Rabbit เป็นตัวควบคุมการทำงานของอุปกรณ์ไฟฟ้าโดยป้อนคำสั่งผ่านมาทางระบบอินเตอร์เน็ต

- ควบคุมระบบการปิด / เปิด ของแสงสว่าง
- รับข้อมูลสวิทช์เพื่อส่งข้อมูลกับไปยัง ผู้ใช้เพื่อแจ้งสถานะ
- ควบคุมการปิด / เปิด เครื่องบันทึกวีดีโอ

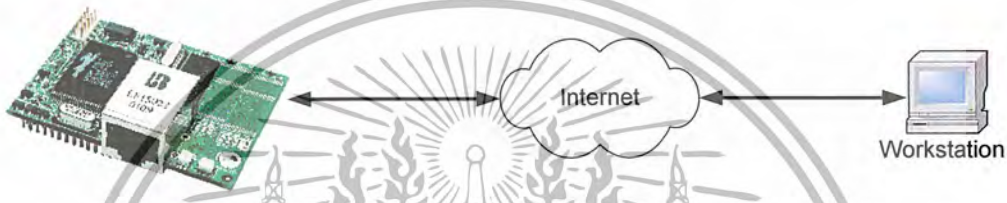


รูปที่ 4.1 ส่วนที่เชื่อมต่อกับอุปกรณ์ไฟฟ้ากับ rabbit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนที่เชื่อมต่ออินเตอร์เน็ต

ส่วนที่ Rabbit นั้นเชื่อมต่อกับระบบอินเตอร์เน็ตเพื่อที่จะรับส่งข้อมูลในการควบคุมผ่านอินเตอร์เน็ต ซึ่งในการส่งข้อมูลจาก Rabbit ไปยังเครื่อง Workstation (รูปที่ 4.2) จะเป็นการส่งข้อมูลแบบ Real Time โดยให้ Rabbit เป็น Server ขนาดเล็ก ซึ่ง Rabbit จะมีหน่วยความจำและความสามารถเพียงพอในการส่งข้อมูล ซึ่งจะทำให้ไม่จำเป็นต้องส่งข้อมูลตลอดเวลา คือ ส่งแค่ครั้งเดียว ก็จะทำงานไปตลอดจนกว่าจะมีการ สั่งยกเลิกหรือ มีการรีเซ็ต โดยในการส่งข้อมูลจะกำหนดไอพีทั้งทางด้านรับและด้านส่ง เพื่อง่ายในการส่งที่ถูกต้อง



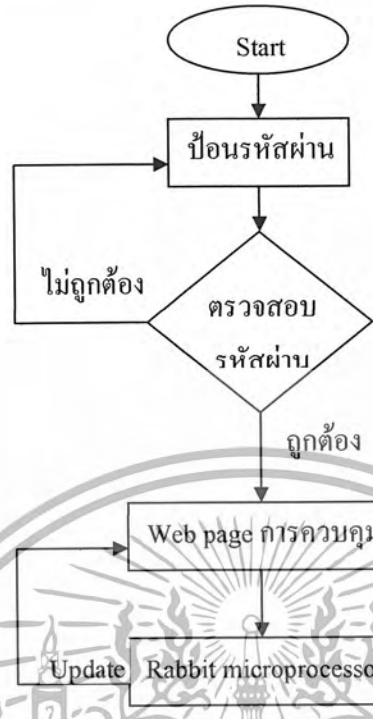
รูปที่ 4.2 การเชื่อมต่อระหว่าง Rabbit กับ Work Station

การทำงานของ Rabbit

ในการทำงานจะใช้ Rabbit เป็น Server ในการประมวลผลทั้งหมดของโปรแกรม ซึ่งในการทำงานในการประมวลผล Rabbit จำทำหน้าที่สองอย่างพร้อมกัน 2 ส่วนในเวลาเดียวกัน คือ

1. **เป็นตัวประมวลผล** ซึ่ง Rabbit จะคอยเป็นตัวรับค่าจากอุปกรณ์เพื่อที่จะนำมาประมวลผลสำหรับการประมวลผลนั้นจะทำทำตามลำดับการโปรแกรมการทำงาน
2. **เป็นเซิร์ฟเวอร์** ซึ่งจะเป็นโปรแกรมที่ควบคุมอุปกรณ์โดยจะแสดงข้อมูลโดยผ่านอินเตอร์เน็ต สำหรับในการเข้าไปใช้งานโปรแกรมเพื่อทำการควบคุมอุปกรณ์ ผู้ใช้ต้องป้อนรหัสและ Password เพื่อป้องกันการเข้าไปใช้ของบุคคลอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

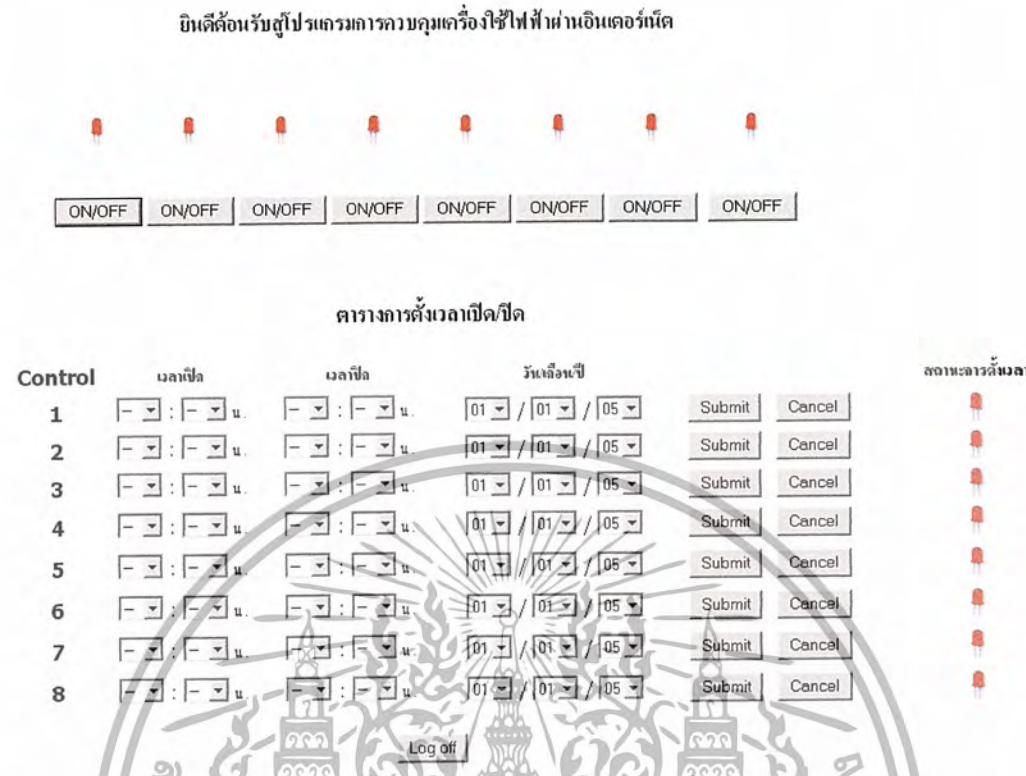


รูปที่ 4.3 Flowchart การทำงาน

รูปที่ 4.3 เป็น flowchart ของการทำงานร่วมกันระหว่างเว็บไซต์ควบคุมกับแรบบิทไมโครโปรเซสเซอร์ ซึ่งจะเริ่มจากกานป้อนรหัสผ่านเพื่อเข้าสู่การควบคุม ดังแสดงในรูปที่ 4.4 จากนั้นระบบจะทำการตรวจสอบรหัสว่าถูกต้องหรือไม่ ถ้าไม่ถูกต้องจะต้องทำการป้อนรหัสใหม่ และถ้าถูกต้อง จะเข้าสู่หน้า web page การควบคุมการทำงานของเครื่องใช้ไฟฟ้า ดังรูปที่ 4.5 สำหรับการควบคุม จะมีการ update ข้อมูลอยู่ตลอดเวลาระหว่างเว็บไซต์ควบคุมกับแรบบิทไมโครโปรเซสเซอร์

รูปที่ 4.4 การเข้าใช้งานระบบควบคุมอุปกรณ์ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

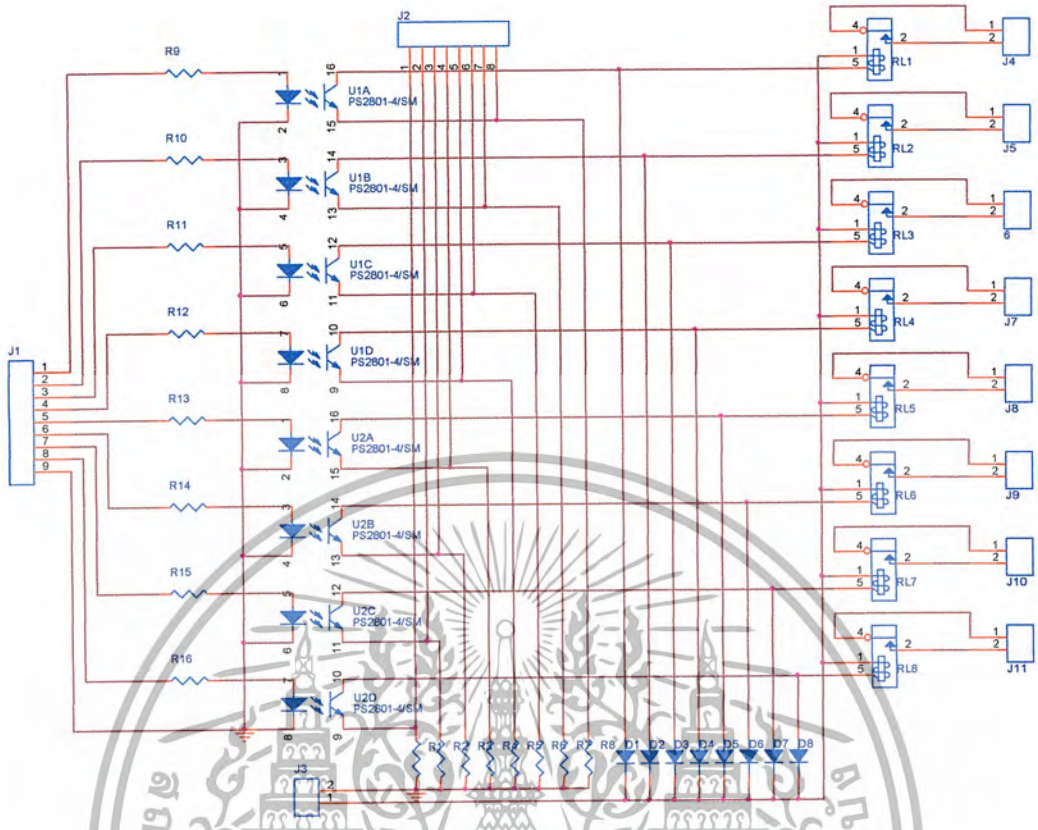


รูปที่ 4.5 หน้า web page ของการควบคุมเครื่องใช้ไฟฟ้า

การควบคุมเครื่องใช้ไฟฟ้าในแต่ละเครื่อง สามารถควบคุมการเปิด/ปิดโดยการกดปุ่มการทำงาน ON/OFF หรือสามารถตั้งเวลาการเปิด/ปิดได้ ซึ่งสามารถตั้งเวลาการเปิด/ปิดได้ตามเวลาที่ต้องการ โดยมีสัญลักษณ์ของ LED เป็นตัวแสดงสถานะการทำงานของเครื่องใช้ไฟฟ้าในแต่ละเครื่อง ซึ่ง LED สีแดง จะแสดงถึงสถานะการปิดการทำงาน (OFF) ของเครื่องใช้ไฟฟ้า และ LED สีเขียว จะเป็นสถานะการเปิดการทำงาน (ON) ของเครื่องใช้ไฟฟ้า

หลังจากทำการควบคุมเสร็จแล้ว เมื่อต้องการออกจากระบบ จะต้องทำการ Log off ออกจากระบบทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 วงจรควบคุม

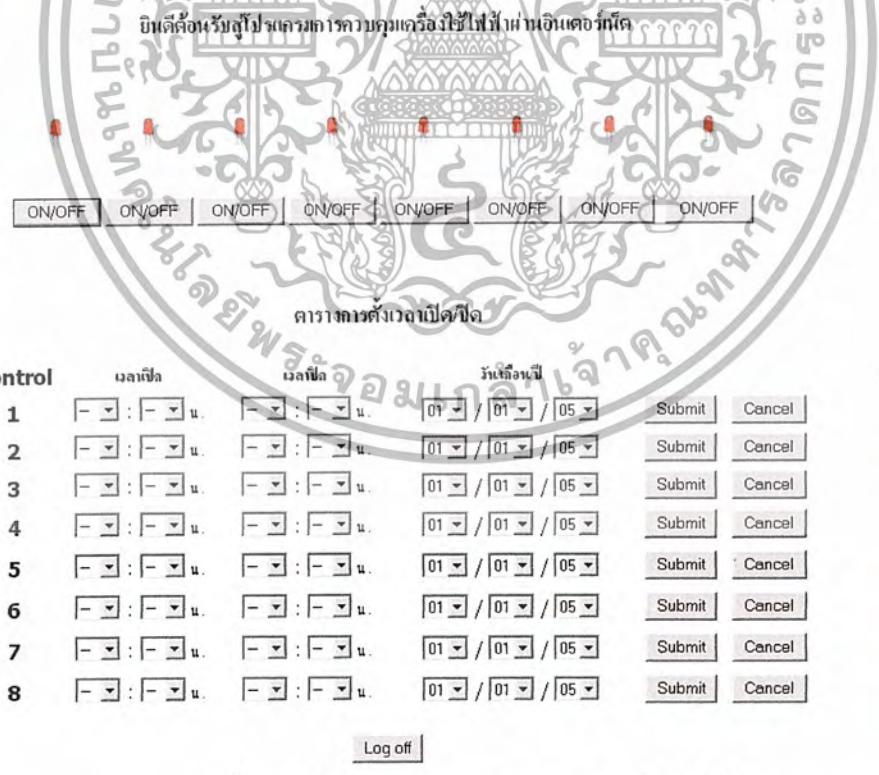
สำหรับวงจรควบคุมในรูปที่ 4.5 เป็นวงจรควบคุมการเปิด/ปิดเครื่องใช้ไฟฟ้าโดยใช้รีเลย์เป็นตัวควบคุม วงจรควบคุมจะรับข้อมูลจากแรบบิทไมโครโปรเซสเซอร์ เพื่อควบคุมการทำงานของวงจรรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ผลการทดลอง

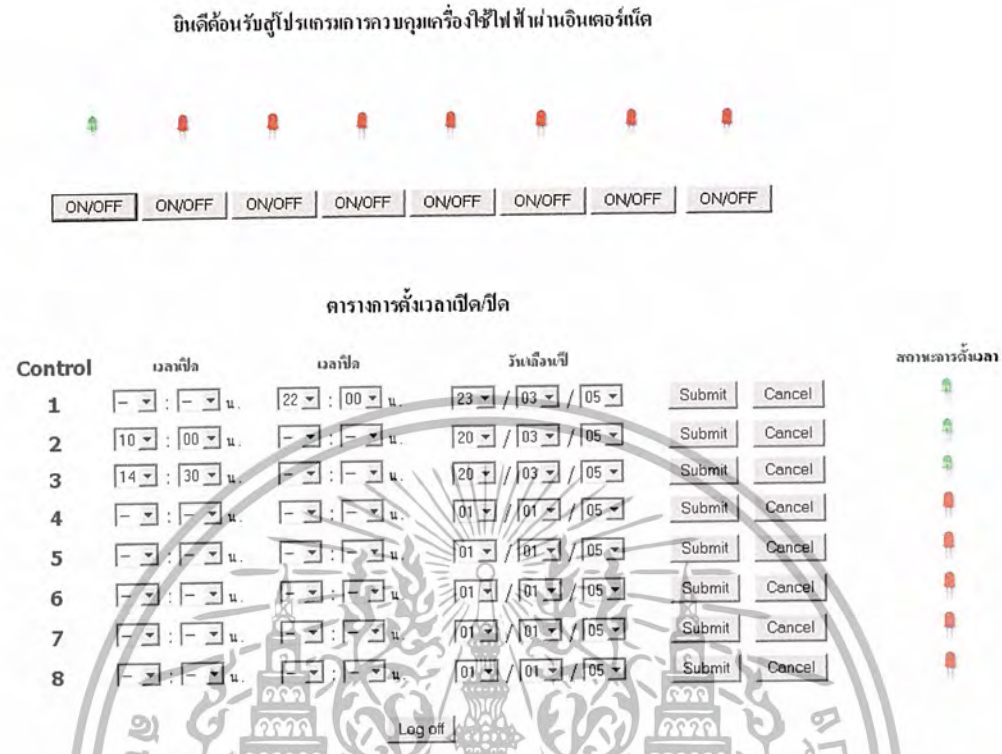
สำหรับผลการทดลองที่ได้ เป็นการควบคุมเครื่องใช้ไฟฟ้าผ่านระบบอินเตอร์เน็ต โดยในการใช้งานต้องมีรหัสผ่านเพื่อเข้าไปใช้งานเพื่อป้องกันบุคคลอื่นเข้าไปใช้งานการควบคุมครื่องใช้ไฟฟ้าภายในบ้านของเราได้ ดังแสดงในรูปที่ 5.1

รูปที่ 5.1 การป้อนรหัสเพื่อเข้าใช้งาน



รูปที่ 5.2 หน้า web page ของการควบคุมเครื่องใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 การควบคุมเครื่องใช้ไฟฟ้าและการแจ้งสถานะการทำงาน

จากรูปที่ 5.2 เป็นหน้า web page สำหรับการควบคุมการทำงานของเครื่องใช้ไฟฟ้าในแต่ละเครื่อง ซึ่งสามารถควบคุมการเปิด/ปิดโดยการกดปุ่มการทำงาน ON/OFF หรือสามารถตั้งเวลาการเปิด/ปิดได้ ซึ่งตั้งเวลาการเปิด/ปิดได้ตามเวลาที่ต้องการ โดยมีสัญลักษณ์ของ LED เป็นตัวแสดงสถานะการทำงานของเครื่องใช้ไฟฟ้าในแต่ละเครื่องและสถานะของการตั้งเวลาการเปิด/ปิด ซึ่ง LED สีแดง จะแสดงถึงสถานะการปิดการทำงาน (OFF) ของเครื่องใช้ไฟฟ้า และ LED สีเขียว จะเป็นสถานะการเปิดการทำงาน (ON) ของเครื่องใช้ไฟฟ้า

ส่วนในรูปที่ 5.3 เป็นการควบคุมเครื่องใช้ไฟฟ้า ที่มีการตั้งเวลาเปิด/ปิดล่วงหน้า สำหรับการตั้งเวลาการเปิด/ปิดเครื่องใช้ไฟฟ้านั้น LED สีแดง จะบอกถึงสถานะไม่ตั้งเวลา และเมื่อมีการตั้งเวลาการเปิด/ปิดเครื่องใช้ไฟฟ้า LED จะกระพริบสีแดงสีเขียวสลับกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปและวิจารณ์ผลการทดลอง

สำหรับการทดลองของโครงงานนี้สามารถควบคุมเครื่องใช้ไฟฟ้าผ่านอินเตอร์เน็ตได้โดยใช้ Rabbit 2200 เป็นตัวควบคุมการทำงานของโปรแกรมโดยการพัฒนามาจาก ไดนามิกซี เป็นตัวพัฒนาโปรแกรม ที่ใช้ในการติดต่อกับส่วนต่างๆ ของโปรแกรม

ซึ่งในปัจจุบันเทคโนโลยีได้เข้ามามีบทบาทในชีวิตประจำวันของเรามากขึ้นทุกวัน ดังนั้น ในฐานนะที่เราเป็นคนที่อยู่ในสังคมปัจจุบันเราน่าจะตามเทคโนโลยีให้ทัน และแนวโน้มในอนาคต เราสามารถที่จะพัฒนาและสร้างโปรแกรมที่ มี Rabbit เป็นตัวควบคุมได้มาก เพราะ Rabbit ได้รองรับเทคโนโลยีทางด้าน อินเตอร์เน็ต และที่สำคัญ สามารถนำไปพัฒนาได้หลายรูปแบบ

6.1 ปัญหาในการทดลอง

สำหรับปัญหาที่เจอในการทดลองมีทั้งในส่วนของซอร์ฟแวร์ และฮาร์ดแวร์

ปัญหาทางด้านซอร์ฟแวร์

1. โปรแกรมที่ใช้พัฒนาโปรแกรมยังไม่ค่อยแพร่หลายเท่าที่ควร
2. หาคู่มือที่เป็นภาษาไทยได้ยากหรือ แทบจะไม่มี
3. สามารถที่จะพัฒนาได้หลายภาษาแต่ไม่ทราบว่าตัวโปรแกรมไหน ง่าย และ ควบคุมได้ดีที่สุด
4. ไม่มีประสบการณ์ในการในด้านการเขียน โปรแกรมควบคุม Rabbit

ปัญหาทางด้านฮาร์ดแวร์

1. Rabbit ค่อนข้างราคาสูงและหาซื้อได้ยาก
2. ยังไม่เป็นที่แพร่หลายเท่าที่ควร
3. อุปกรณ์ที่ใช้ในส่วนที่ติดต่อกับ Rabbit บางตัวหาได้ยาก
4. ไม่มีประสบการณ์ด้านฮาร์ดแวร์

6.2 แนวทางการแก้ไข

6.2.1 ศึกษาโปรแกรมให้มากขึ้น

6.2.2 ศึกษาโปรแกรมตัวอย่างจากเว็บที่มีตัวอย่าง

6.2.3 ปรึกษาอาจารย์ที่ปรึกษาเมื่อไม่เข้าใจ

6.2.4 ทุ่มเทกับการทำงานให้มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 แนวทางการพัฒนา

6.3.1 ศึกษางานอุปกรณ์ต่างๆ ให้มากขึ้นเพื่อที่จะนำไปพัฒนาเพิ่มขึ้น

6.3.2 ศึกษาระบบโครงสร้างของ Rabbit เพื่อนำมาพัฒนาร่วมกับไมโครโปรเซสเซอร์ตัวอื่นเพื่อที่จะให้มีประสิทธิภาพมากขึ้น

6.3.3 สามารถพัฒนามาเป็นธุรกิจได้ถ้าโครงงานมีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. http://www.rabbitsemiconductor.it/
2. http://www.rabbitsemiconductor.com/
3. http://www.synes.co.th/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.

Package Schematic and Pinout

รูปที่ 1 Package Outline and Pin Assignments

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

Rabbit Pin Descriptions

ตารางที่ 1 Rabbit Pin Descriptions

Pin Group	Pin Name	Direction	Pin Numbers
Hardware	CLK	Output	1
	/RESET	Input	37
	XTALA1	Input	40
	XTALA2	Output	41
	XTALB1	Input	90
	XTALB2	Output	91
CPU Buses	A0-A19	Output	7, 17-20, 61-68,
	D0-D7	Bidirectional	9-16
Status/Control	/WDTOUT	Output	34
Status	STATUS	Output	38
Status	SMODE1	Input	35-36 (1:0)
Chip Selects	/CS0	Output	8
	/CS1	Output	5
	/CS2	Output	4
Output Enables	/OE0	Output	6
	/OE1	Output	76
Write Enables	/WE0	Output	69
	/WE1	Output	80
I/O Control	/BUFEN	Output	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1 Rabbit Pin Descriptions (ต่อ)

Pin Group	Pin Name	Direction	Pin Numbers
I/O Read Strobe	/IORD	Output	32
I/O Write Strobe	/IOWR	Output	31
I/O Port A	PA0-PA7	Input/	81-88
I/O Port B	PB0-PB7	6 In/2 Out	93-100
I/O Port C	PC0-PC7	4 In/4 Out	51. 54-60
I/O Port D	PD0-PD7	Input/	43-50
I/O Port E	PE7-PE0	Input/	21-26. 29. 30
Power	VBAT		42
	VDD		3. 28. 53. 78. 92
	VSS		2. 27. 39. 52. 77.
Serial Ports	CLKA	Input/	94
	CLKB	Input/	93
	RXA. TXA.	RX--input	51. 54-60
	ARXA. ATXA.	RX--input	43-46
Slave Port	SD0-SD7	Bidirectional	81-88
	/SLAVEATTN	Output	100
	/SRD	Input	96
	/SWR	Input	95
	SA0. SA1	Input	97.98
	/SCS	Input	21
I/O Strobes	/I0./I1.	Outputs	21-26. 29. 30
External Interrupt	INT0A. INT0B	Inputs	24. 30
External Interrupt	INT1A. INT1B	Inputs	23. 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

Rabbit Internal I/O Registers

ตารางที่ 2 Rabbit Internal I/O Registers

Address	Reset Value	Functionality
GCSR=00h	11000000	Global Control Status Register. Control of clocks, periodic interrupts, and monitoring of watchdog.
RTCCR=01h	00000000	Real-Time Clock Control Register.
RTC0R=02h	xxxxxxxx	Real-Time Clock Byte 0 Register.
RTC1R=03h	xxxxxxxx	Real-Time Clock Byte 1 Register.
RTC2R=04h	xxxxxxxx	Real-Time Clock Byte 2 Register.
RTC3R=05h	xxxxxxxx	Real-Time Clock Byte 3 Register.
RTC4R=06h	xxxxxxxx	Real-Time Clock Byte 4 Register.
RTC5R=07h	xxxxxxxx	Real-Time Clock Byte 5 Register.
WDTCR=08h	00000000	Watchdog Timer Control Register.
WDTTR=09h	00000000	Watchdog Timer Test Register.
GOCR=0Eh	00000x00	Global Output Control Register.
GCDR=0Fh	xxxxx000	Global Clock Doubler Register.
MMIDR=10h	xxx00000	Memory Management I and D Space Register. Controls I & D space enable and battery switchover support for /CS1.
XPC	00000000	Not an I/O register, but initialized to zero by reset.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2 Rabbit Internal I/O Registers (ต่อ)

Address	Reset Value	Functionality
STACKSEG=11h (Z180 CBR)	00000000	Stack segment memory pointer. Locates stack segment in physical memory.
DATASEG=12h (Z180 BBR)	00000000	Data segment memory pointer. Locates data segment in physical memory.
SEGSIZE=13h (Z180 CBAR)	11111111	Specifies start of data segment and start of stack segment in 64K memory space.
MB0CR=14h	00000000	Memory Bank 0 Control Register. Controls mapping of first memory quadrant 256K to physical memory chips.
MB1CR=15h	xxxxxxxx	Memory Bank 1 Control Register. Controls mapping of second memory quadrant to physical memory chips.
MB2CR=16h	xxxxxxxx	Memory Bank 2 Control Register. Controls mapping of third memory quadrant to physical memory chips.
MB3CR=17h	xxxxxxxx	Memory Bank 3 Control Register. Controls mapping of fourth memory quadrant to physical memory chips.
SPD0R=20h	xxxxxxxx	Slave Port Register 0. Separate registers for read and write used for slave port communication.
SPD1R=21h	xxxxxxxx	Slave port register 1.
SPD2R=22h	xxxxxxxx	Slave port register 2.
SPSR=023h	00000000	Slave port status register.
SPCR=24h	000x0000	Slave port control register.
PADR=30h	xxxxxxxx	Parallel port A data register. R/W.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2 Rabbit Internal I/O Registers (ต่อ)

Address	Reset Value	Functionality
PBDR=40h	00xxxxxx	Parallel port B data register. R/W.
PCDR=50h	x0x0x0x0	Parallel port C data register.
PCFR=55h	x0x0x0x0	Port C function register.
PDDR=60h	xxxxxxxx	Parallel port D data register. R/W.
PDCR=64h	xx00xx00	Port D control register
PDFR=65h	xxxxxxxx	Port D function register.
PDDCR=66h	xxxxxxxx	Port D drive control register.
PDDDR=67h	00000000	Port D data direction register.
PDB0R=68h	xxxxxxxx	Port D bit 0 register. W
PDB1R=69h	xxxxxxxx	Bit 1.
PDB2R=6Ah	xxxxxxxx	Bit 2.
PDB3R=6Bh	xxxxxxxx	Bit 3.
PDB4R=6Ch	xxxxxxxx	Bit 4.
PDB5R=6Dh	xxxxxxxx	Bit 5.
PDB6R=6Eh	xxxxxxxx	Bit 6.
PDB7R=6Fh	xxxxxxxx	Bit 7.
PEDR=70h	xxxxxxxx	Parallel port E data register. R/W.
PECR=74h	xx00xx00	Port E control register.
PEFR=75h	xxxxxxx	Port E function register.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2 Rabbit Internal I/O Registers (ต่อ)

Address	Reset Value	Functionality
PEDDR=77h	0000000	Port E data direction register.
PEB0R=78h	xxxxxxx	Port E bit 0 register. W
PEB1R=79h	xxxxxxx	Bit 1.
PEB2R=7Ah	xxxxxxx	Bit 2.
PEB3R=7Bh	xxxxxxx	Bit 3.
PEB4R=7Ch	xxxxxxx	Bit 4.
PEB5R=7Dh	xxxxxxx	Bit 5.
PEB6R=7Eh	xxxxxxx	Bit 6.
PEB7R=7FH	xxxxxxx	Bit 7
IB0CR=80h	00000xxx	External I/O control bank 0
IB1CR=81h	00000xxx	External I/O control bank 1
IB2CR=82h	00000xxx	External I/O control bank 2
IB3CR=83h	00000xxx	External I/O control bank 3
IB4CR=84h	00000xxx	External I/O control bank 4
IB5CR=85h	00000xxx	External I/O control bank 5
IB6CR=86h	00000xxx	External I/O control bank 6
IB7CR=87h	00000xxx	External I/O control bank 7
I0CR=98h	xx000000	External interrupt 0 control register.
I1CR=99h	xx000000	External interrupt 1 control register.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2 Rabbit Internal I/O Registers (ต่อ)

Address	Reset Value	Functionality
TACSR=0A0h	0000xx00	Timer A Control/Status Register
TACR=0A4h	xxxxxxxx	Timer A Control Register
TAT1R=0A3h	0000xx00	Timer A1 Time Constant 1 Register
TAT4R=0A9h	xxxxxxxx	Timer A4 Time Constant 4 Register
TAT5R=0ABh	xxxxxxxx	Timer A5 Time Constant 5 Register
TAT6R=0ADh	xxxxxxxx	Timer A6 Time Constant 6 Register
TAT7R=0AFh	xxxxxxxx	Timer A7 Time Constant 7 Register
TBCSR=0B0h	xxxxx000	Timer B Control/Status Register
TBCR=0B1h	xxxx0000	Timer B Control Register
TBM1R=0B2h	xxxxxxxx	Timer B MSB 1 Reg
TBL1R=0B3h	xxxxxxxx	Timer B LSB 1 Reg
TBM2R=0B4h	xxxxxxxx	Timer B MSB 2 Reg
TBL2R=0B5h	xxxxxxxx	Timer B LSB 2 Reg
TBCMR=0BEh	xxxxxxxx	Timer B Count MSB Reg
TBCLR=0BFh	xxxxxxxx	Timer B Count LSB Reg
SADR=0C0h	xxxxxxxx	Serial port A data register receive/send.
SAAR=0C1h	xxxxxxxx	Serial port A alternate data register (transmit 9th bit)
SASR=0C3h	0xx00000	Serial port A status register.
SACR=0C4h	xx000000	Serial port A control register.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

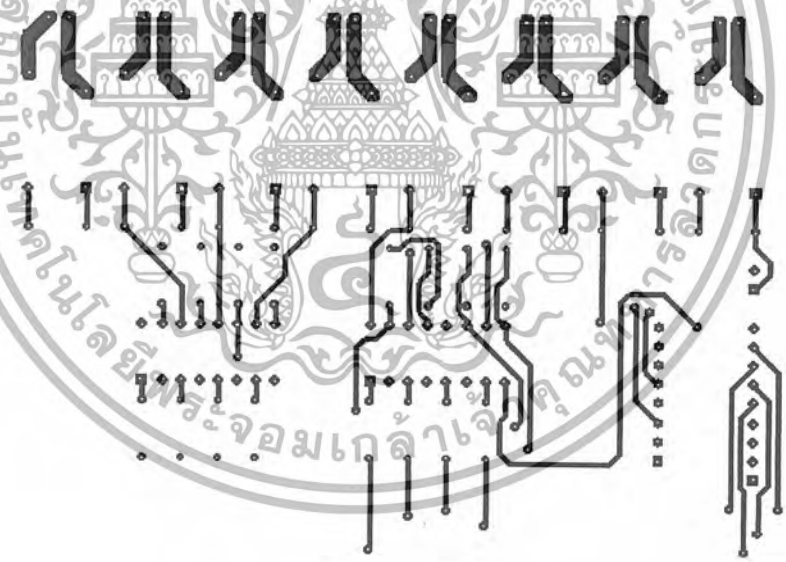
ตารางที่ 2 Rabbit Internal I/O Registers (ต่อ)

Address	Reset Value	Functionality
SBDR=0D0h	xxxxxxxx	Serial port B data register receive/send.
SBAR=0D1h	xxxxxxxx	Serial port B alternate data register (transmit 9th bit)
SBSR=0D3h	0xx00000	Serial port B status register.
SBCR=0D4h	xx000000	Serial port B control register.
SCDR=0E0h	xxxxxxxx	Serial port C data register receive/send.
SCAR=0E1h	xxxxxxxx	Serial port C alternate data register (transmit 9th bit)
SCSR=0E3h	0xx00000	Serial port C status register.
SCCR=0E4h	xx00x000	Serial port C control register.
SDDR=0F0h	xxxxxxxx	Serial port D data register receive/send.
SDAR=0F1h	xxxxxxxx	Serial port D alternate data register (transmit 9th bit)
SDSR=0F3h	0xx00000	Serial port D status register.
SDCR=0F4h	xx00x000	Serial port D control register.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

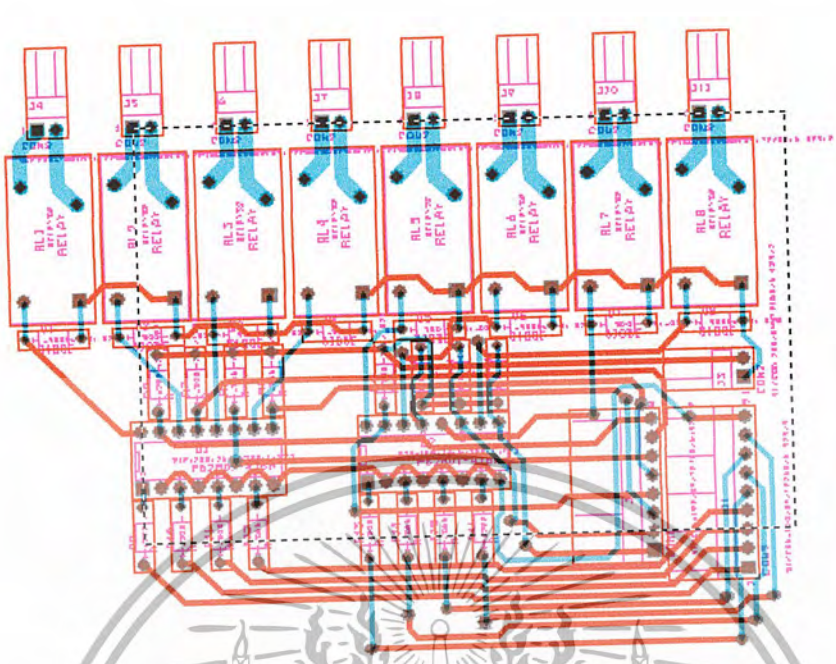
ภาคผนวก ง.
วงจรควบคุม

รูปที่ 2 ลายทองแดงด้านหน้า



รูปที่ 3 ลายทองแดงด้านหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4 ตำแหน่งการวางอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้