บทที่ 4

การประเมินผลและตัวอย่างการประยุกต์ใช้งาน

ในระบบไมโครโปรเซสเซอร์แบบกระจายที่สร้างขึ้น ไมโครโปรเซสเซอร์แต่ละตัวสามารถทำงานได้อัตราเดียวกับไมโครโปรเซสเซอร์ทั่วไป ยกเว้นในกรณีที่มีการสื่อสารกับไมโครโปรเซสเซอร์ตัวอื่นผ่านหน่วยความจำร่วม ความเร็วในการทำงานจะลดลง เนื่องจากบางครั้งไมโครโปรเซสเซอร์ต้องหยุดรออยู่จนกว่าจะถึงลำดับที่มีสิทธิ์ใช้หน่วยความจำร่วม ดังนั้นในบทนี้จะกล่าวถึงการวิเคราะห์เวลาที่เพิ่มขึ้นในการ access หน่วยความจำร่วม และการทดลองเพื่อหาค่าความเร็วในการทำงานที่ลดลงเมื่อใช้หน่วยความจำร่วม แล้วนำมาสรุปเป็นคุณสมบัติของระบบที่ออกแบบ พร้อมกับยกตัวอย่างการประยุกต์ใช้งานเพื่อเป็นแนวทางในการนำไปใช้งานอย่างจริงจังต่อไปในอนาคต

4.1 การวิเคราะห์ Access Time ในการใช้หน่วยความจำร่วม

กำหนดให้ Access Time เป็นเวลานับตั้งแต่ไมโครโปรเซสเซอร์เริ่มส่งสัญญาณ REQUEST มายังวงจร ARBITER เพื่อขอใช้หน่วยความจำร่วม จนกระทั่งการใช้หน่วยความจำร่วมเสร็จสิ้นลงหรือเวลาที่ไมโครโปรเซสเซอร์ใช้ในแมชินไซเคิลการเขียนหรืออ่านหน่วยความจำในสภาวะเอ็กซ์คิว จะได้ว่า

$$\text{Access Time} = TQ + TM + TP \qquad (4.1)$$

โดยที่

TQ = Queueing Time คือเวลาที่ไมโครโปรเซสเซอร์ต้องรอนับจากเริ่มส่งสัญญาณ REQUEST จนกระทั่งได้รับสัญญาณ GRANT TQ สำหรับระบบที่มีไมโครโปรเซสเซอร์ 4 ตัว หาได้จากตาราง 4.1

TM = Memory Setting Time คือเวลาที่ใช้ทำงานภายในหน่วยความจำ RAM เพื่อเตรียมรับหรือส่งข้อมูลจากแอดเดรสที่กำหนด เท่ากับ 1 WAIT STATE ที่ต้องเพิ่มขึ้น

ตาราง 4.1 QUEUEING TIME ในกรณีต่าง ๆ

SCANNING SIGNAL	uP 1	uP 2	uP 3	uP 4
S1	0	Tw1	Tw1 + Tw2	Tw1+Tw2+Tw3
S2	Tw2+Tw3+Tw4	0	Tw2	Tw2 + Tw3
S3	Tw3 + Tw4	Tw3+Tw4+Tw1	0	Tw3
S4	Tw4	Tw4 + Tw1	Tw4+Tw1+Tw2	0

เมื่อ Twi = จำนวน Wait state ที่เกิดขึ้นจากไมโครโปรเซสเซอร์ i

= 1 clock ถ้าไมโครโปรเซสเซอร์ i ไม่ต้องการใช้หน่วยความจำร่วม

หรือ = 3 clock ถ้าไมโครโปรเซสเซอร์ i ต้องการใช้หน่วยความจำร่วม

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

TP = Processor Cycle Time คือเวลาที่ไมโครโปรเซสเซอร์ต้องใช้ในการทำคำสั่ง อ่านหรือเขียนข้อมูลในหน่วยความจำ ซึ่งในกรณี Z-80 จะมี 3 STATE คือ T1, T2 และ T3

พิจารณาสองกรณีต่อไปนี้

กรณีดีที่สุด เกิดขึ้นในกรณีที่ไมโครโปรเซสเซอร์ส่งสัญญาณ REQUEST ไปยัง CONTROLLER แล้วทันใดนั้นก็ได้รับสัญญาณ SCANNING จึงได้รับสัญญาณ GRANT ทันที โดยไม่มี Queueing Time (รูป 4.1) จากสมการ 4.1 จะได้

Access Time = 0 + 1 + 3 = 4 clock period (4.2)

ที่สัญญาณ clock 2 MHz จะได้ Access Time = 2 microsec.

กรณีเลวที่สุด เกิดขึ้นเมื่อไมโครโปรเซสเซอร์ต้องการใช้หน่วยความจำร่วมอีกครั้งหนึ่ง หลังจากเพิ่งเสร็จสิ้นการใช้หน่วยความจำร่วม และไมโครโปรเซสเซอร์ตัวอื่น ๆ ก็ต้องการใช้หน่วยความจำ (รูป 4.2) จากสมการ 4.1 จะได้

Access Time = (3X3) + 1 + 3 = 13 clock period (4.3)

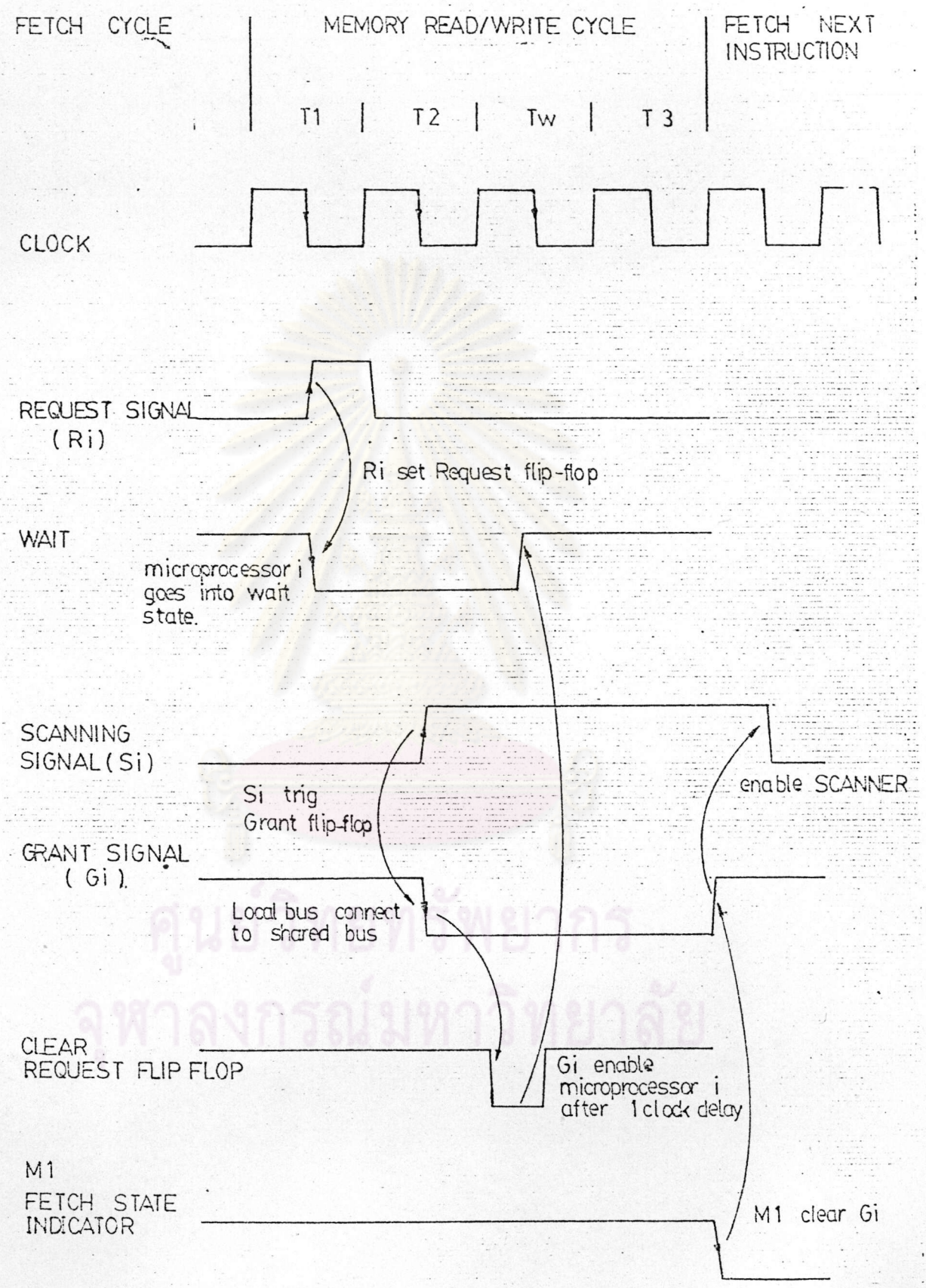
ที่สัญญาณ clock 2 MHz จะได้ Access Time = 6.5 microsec.

จะเห็นว่า Access Time ในการใช้หน่วยความจำร่วม มีค่าระหว่าง 4 ถึง 13 clock period ในขณะที่ Access Time ในการใช้หน่วยความจำส่วนตัวของไมโครโปรเซสเซอร์แต่ละตัวเท่ากับ 3 clock period นั่นคือ Access Time ในการใช้หน่วยความจำร่วมมากกว่าปกติ 0.3 ถึง 3.3 เท่า ตาราง 4.2 แสดงให้เห็นค่า Access Time ที่มากที่สุดในกรณีที่ระบบมีไมโครโปรเซสเซอร์ตั้งแต่ 2 ถึง 4 ตัว ใช้หน่วยความจำร่วมและ Access Time ที่เพิ่มขึ้นจากเวลาปกติ แสดงไว้ในรูป 4.3

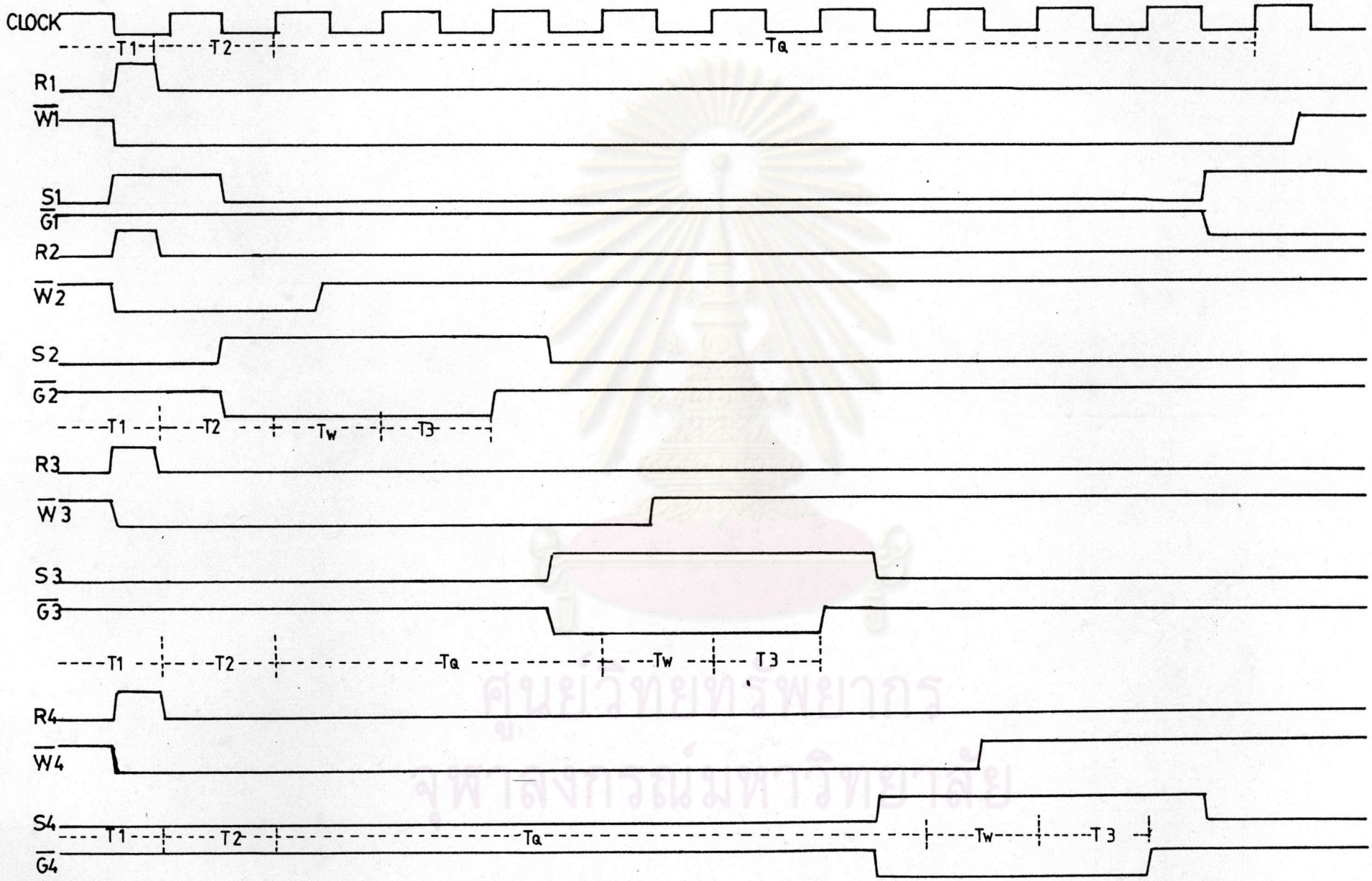
จากตาราง 4.2 ทำให้สามารถพยากรณ์อัตราการรับส่งข้อมูลสูงสุดผ่านหน่วยความจำ ระหว่างไมโครโปรเซสเซอร์ 2 ตัวในระบบที่มีไมโครโปรเซสเซอร์เพียง 4 ตัว ได้ดังนี้

ไมโครโปรเซสเซอร์ใช้เวลาเฟทช์คำสั่ง 4 clock หรือ 2 microsec.

Maximum Access Time ในการใช้ Shared Memory 6.5 microsec.

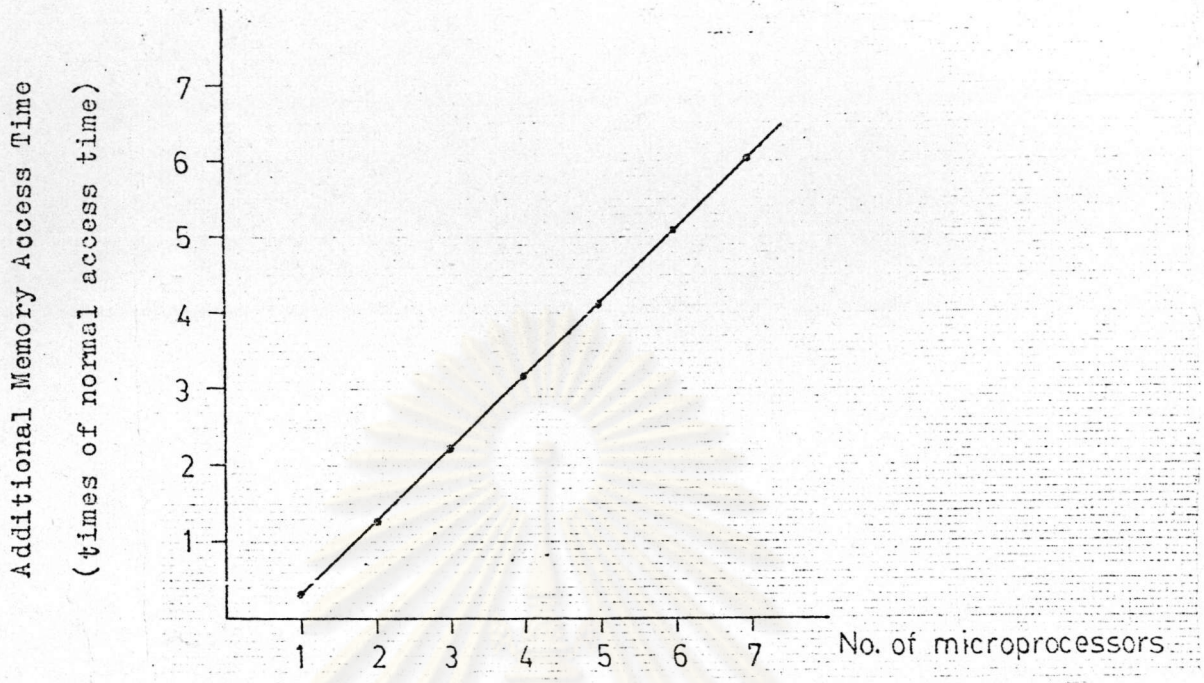


รูป 4.1 การ ACCESS หน่วยความจำร่วมกรณีดีที่สุด

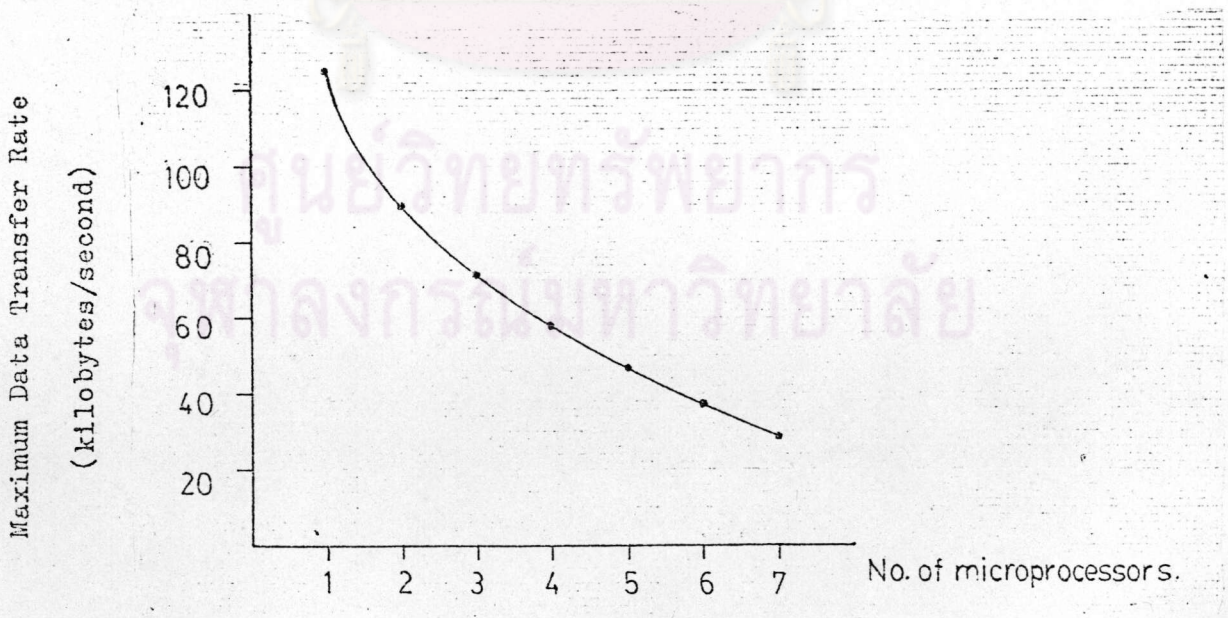


รูป 4.2 การ Access หน่วยความจำร่วมกรณีเลวที่สุด

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูป 4.3 ACCESS TIME ที่เพิ่มขึ้นเมื่อใช้หน่วยความจำร่วมในระบบไมโครโปรเซสเซอร์แบบกระจาย



รูป 4.4 อัตราการรับส่งข้อมูลสูงสุดระหว่างไมโครโปรเซสเซอร์ที่ใช้หน่วยความจำร่วมในระบบไมโครโปรเซสเซอร์แบบกระจาย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตาราง 4.2 แสดงค่า Access Time ที่มากที่สุดในการใช้หน่วยความจำร่วม

จำนวนไมโครฯ ที่ใช้หน่วยความจำร่วม	Access Time สูงสุด (clock)	Access Time สูงสุด 2 MHz (microsec)	Access Time ที่เพิ่มขึ้น (จำนวนเท่าของปกติ)
1	4	2.0	0.33
2	7	3.5	1.34
3	10	5.0	2.34
4	13	6.5	3.34

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ดังนั้นในการส่งข้อมูลไปยังหน่วยความจำร่วมใช้เวลา 8.5 microsec/byte.
ในทำนองเดียวกันในการรับข้อมูลจากหน่วยความจำร่วมใช้เวลา 8.5 microsec/byte.

นั้นคือ ในการรับส่งข้อมูล 1 ไบท์ ต้องใช้เวลา 17 microsec.
อัตราการรับส่งข้อมูลที่สูงที่สุดในการใช้หน่วยความจำร่วม เท่ากับ 58 kilobyte/sec.

อัตราการรับส่งข้อมูลผ่านหน่วยความจำร่วมลดลงตามจำนวนไมโครโปรเซสเซอร์ที่ใช้หน่วยความจำร่วมที่เพิ่มขึ้น ดังรูป 4.4 ซึ่งจะเห็นว่าอัตราการรับส่งข้อมูลจะลดลงต่ำกว่า 30 kilobyte/sec. เมื่อจำนวนไมโครโปรเซสเซอร์ที่ใช้หน่วยความจำร่วมมีตั้งแต่ 7 ตัวขึ้นไป อัตราการรับส่งข้อมูล 30 kilobyte/sec. นี้ใกล้เคียงกับการรับส่งข้อมูลระหว่างไมโครโปรเซสเซอร์ผ่าน I/O Port ซึ่งมีค่าประมาณ 23 kilobyte/sec. (19)ระบบที่ออกแบบนี้จึงใช้ได้กับระบบที่มีไมโครโปรเซสเซอร์ไม่เกิน 6 ตัว

4.2 การทดลองเพื่อหาเวลาทำงานที่เพิ่มขึ้นเมื่อไมโครโปรเซสเซอร์ใช้หน่วยความจำร่วม

การทดลองนี้กระทำขึ้นเพื่อหาความสัมพันธ์ระหว่างเวลาทำงานที่เพิ่มขึ้นกับจำนวนไมโครโปรเซสเซอร์ที่ใช้หน่วยความจำร่วม เพื่อกำหนดจำนวนไมโครโปรเซสเซอร์สูงสุดที่สามารถใช้หน่วยความจำร่วมได้

การทดลองทำโดยให้ไมโครโปรเซสเซอร์ตัวที่ 1 วนเวียนอ่านข้อมูลจากหน่วยความจำร่วมพร้อมกับบันทึกจำนวนไบท์ที่อ่านได้ไว้ภายในช่วงเวลาที่กำหนด ในขณะที่ไมโครโปรเซสเซอร์ตัวที่ 2 และ 3 วนเวียนอ่านข้อมูลจากหน่วยความจำร่วมไปพร้อมกันด้วย การจับเวลาทำโดยใช้ Z-80 CTC (Counter Timer Chip) ควบคุมโดยไมโครโปรเซสเซอร์ตัวที่ 1 จากจำนวนไบท์ที่อ่านได้ทำให้คำนวณเวลาที่ไมโครโปรเซสเซอร์ใช้ในการอ่านข้อมูล 1 ไบท์ ในกรณีต่าง ๆ กันคือ เมื่อไมโครโปรเซสเซอร์ตัวที่ 1 ใช้หน่วยความจำส่วนตัว เมื่อไมโครโปรเซสเซอร์ตัวที่ 1 เพียงตัวเดียวใช้หน่วยความจำร่วม เมื่อไมโครโปรเซสเซอร์ตัวที่ 1 และ 2 ใช้หน่วยความจำร่วมพร้อมกัน และเมื่อไมโครโปรเซสเซอร์ทั้งสามตัวใช้หน่วยความจำร่วมพร้อมกัน แล้วนำมาเปรียบเทียบจะได้ความสัมพันธ์ระหว่างเวลาทำงานที่เพิ่มขึ้นกับจำนวนไมโครโปรเซสเซอร์ที่ใช้หน่วยความจำร่วม แผนผังการทำงานและโปรแกรมสำหรับไมโครโปรเซสเซอร์แต่ละตัว

แสดงไว้ดังรูป 4.5 และ 4.6 ตามลำดับ

4.2.1 ขั้นตอนการทดลอง

1. Load โปรแกรมในรูป 4.6 ก ลงในไมโครโปรเซสเซอร์ตัวที่ 1 และรูป 4.6 ข ลงในไมโครโปรเซสเซอร์ตัวที่ 2 และ 3 ตามลำดับ

2. ให้ไมโครโปรเซสเซอร์ตัวที่ 1 วนเวียนอ่านข้อมูลจากหน่วยความจำส่วนตัว โดยเปลี่ยน SHARED ในโปรแกรมรูป 4.6 ก เป็น 1E01 บันทึกจำนวนไบท์ที่อ่านได้จาก NOBYTE.

3. ให้ไมโครโปรเซสเซอร์ตัวที่ 1 เพียงตัวเดียว วนเวียนอ่านข้อมูลจากหน่วยความจำร่วม บันทึกจำนวนไบท์ที่อ่านได้จาก NOBYTE.

4. ให้ไมโครโปรเซสเซอร์ตัวที่ 1 และ 2 วนเวียนอ่านข้อมูลจากหน่วยความจำร่วมพร้อมกัน แล้วบันทึกจำนวนไบท์ที่อ่านได้จาก NOBYTE.

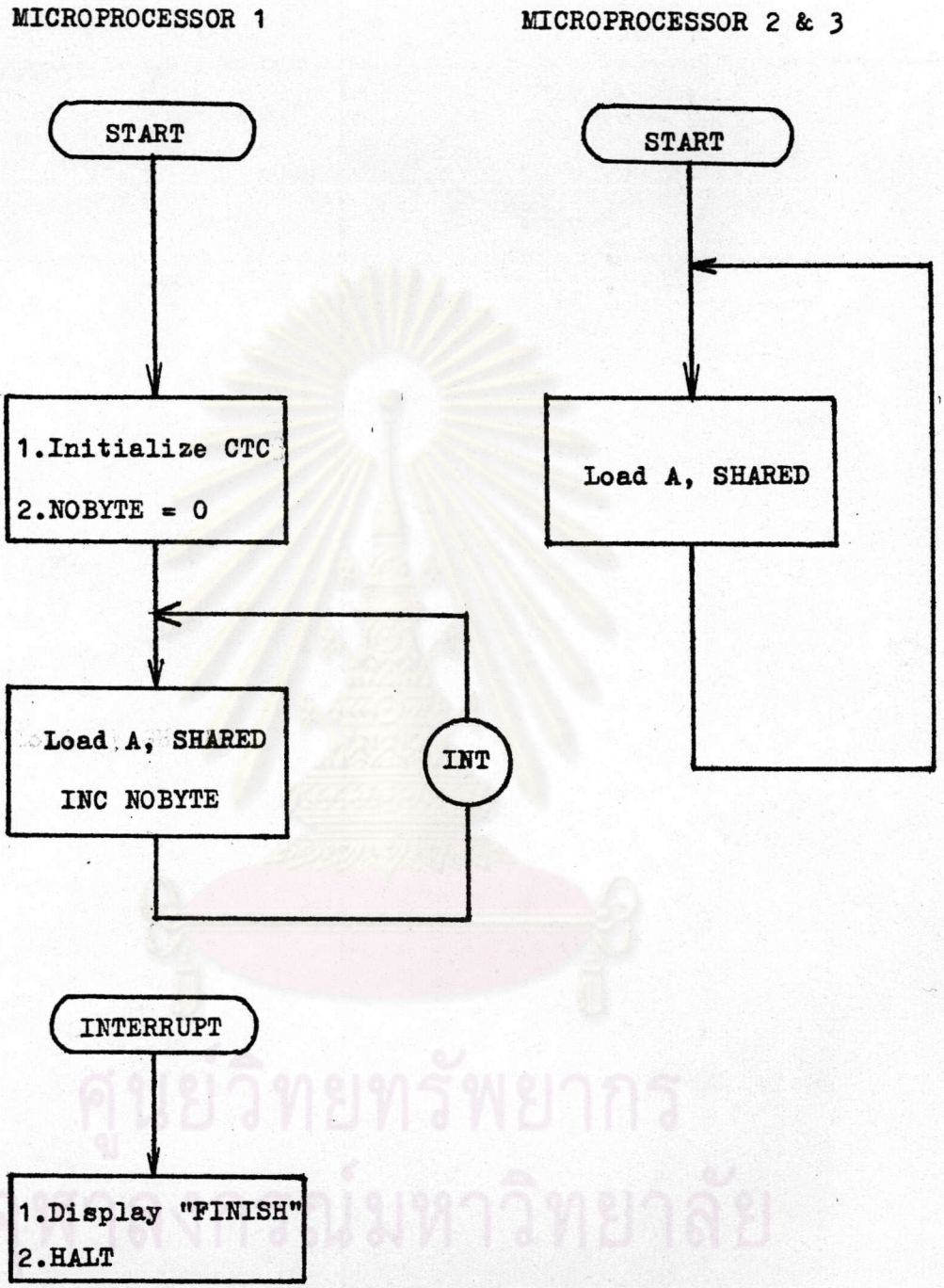
5. ให้ไมโครโปรเซสเซอร์ตัวที่ 1, 2 และ 3 วนเวียนอ่านข้อมูลจากหน่วยความจำร่วมพร้อมกัน แล้วบันทึกจำนวนไบท์ที่อ่านได้จาก NOBYTE.

6. ทำการทดลองข้อ 2-5 ซ้ำอีก โดยเปลี่ยนแปลงช่วงเวลาให้ยาวขึ้น ซึ่งทำได้โดยเปลี่ยน NUM ในโปรแกรมรูป 4.6 ก เป็นค่าต่าง ๆ กัน จนครบ 16 ค่า ดังตาราง 4.3

4.2.2 ผลการทดลอง

ผลการทดลองตามข้อ 4.2.1 แสดงไว้ในตาราง 4.3 ซึ่งนำมาคำนวณเวลาโดยเฉลี่ยที่ไมโครโปรเซสเซอร์ใช้ในการอ่านข้อมูลจากหน่วยความจำร่วม หนึ่งไบท์ ได้ตามตาราง 4.4 และนำค่าเฉลี่ยมาเปรียบเทียบเวลาทำงานที่เพิ่มขึ้น แล้วสรุปเป็นความสัมพันธ์ระหว่างเวลาทำงานที่เพิ่มขึ้นกับจำนวนไมโครโปรเซสเซอร์ที่ใช้หน่วยความจำร่วม ดังรูป 4.7

จากรูป 4.7 จะเห็นว่าการใช้หน่วยความจำร่วม ทำให้ไมโครโปรเซสเซอร์ทำงานช้าลงประมาณ 9% เป็นอย่างน้อย และการเพิ่มจำนวนไมโครโปรเซสเซอร์ที่ใช้หน่วยความจำร่วมจาก 1 ตัว ไปจนถึง 4 ตัว เวลาทำงานที่เพิ่มขึ้นจะแตกต่างกัน



รูป 4.5 แผนผังการทำงานของไมโครโปรเซสเซอร์ในการทดลอง

```
                ;           MICROPROCESSOR 1 PROGRAM

1800            CTCO        EQU 40 H
      (05FE)    SCAN        EQU 05FE H
      (1808)    NUM         EQU 1808 H
      (5001)    SHARED      EQU 5001 H
      (181A)    NOBYTE      EQU 181A H
1800  3E18                  LD A, 18          ; Loading the interrupt
1802  ED47                  LD I, A           ; register
1804  3EB5                  LD A, B5          ; loading the channel
1806  D340                  OUT(CTCO), A      ; control
1808  3EXX                  LD A, NUM         ; loading the constant
180A  D340                  OUT(CTCO), A      ; register for Time period
180C  3EA8                  LD A, 0A8 H       ; loading the interrupt
180E  D340                  OUT(CTCO), A      ; vector register
1810  ED5E                  IM 2              ; set interrupt mode 2
1812  PB                    EI
1813  210000                LD HL, 0000       ; set NOBYTE to zero
1816  220019                LD(NOBYTE), HL
1819  3A0150      LOOP      LD A, (SHARED)    ; read data from shared memory
181C  23                    INC HL            ; increment NOBYTE
181D  220019                LD(NOBYTE), HL
1820  18F7                  JR LOOP           ; loop until time cut
18A8  18B0                  ORG 18A8 H
                            DEFW INTERRUPT
                INTERRUPT   EQU 18B0 H
18B0                        ORG 18B0 H
18B0  DD21E018  INTERRUPT   LD IX, BUF        ; point to data buffer
18B4  CDFE05                CALL SCAN         ; display FINISH and then
18B7  76                    HALT              ; halt
18E0                        ORG 18E0 H
18E0  37        BUF         DEFB 37 H
18E1  A6                    DEFB A6 H
18E2  30                    DEFB 30 H
18E3  23                    DEFB 23 H
18E4  30                    DEFB 30 H
18E5  0F                    DEFB 0F H
```

ก) โปรแกรมของไมโครโปรเซสเซอร์ตัวที่ 1

```
                            MICROPROCESSOR 2 PROGRAM

8000                        ORG 8000 H
      (5001)    SHARE       EQU 5001 H
8000  3A0150                LD A,(SHARE )     ; read data from share memory
8003  18FB                  JR LOOP           ; again and again
```

ข) โปรแกรมของไมโครโปรเซสเซอร์ตัวที่ 2 และ 3

รูป 4.6 โปรแกรมสำหรับไมโครโปรเซสเซอร์ 1 (ก) และไมโคร-โปรเซสเซอร์ 2 และ 3 (ข) ในการทดลอง

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตาราง 4.3 ผลการทดลองวัดจำนวนไบท์ที่ไมโครโปรเซสเซอร์อ่านได้ในกรณีต่าง ๆ

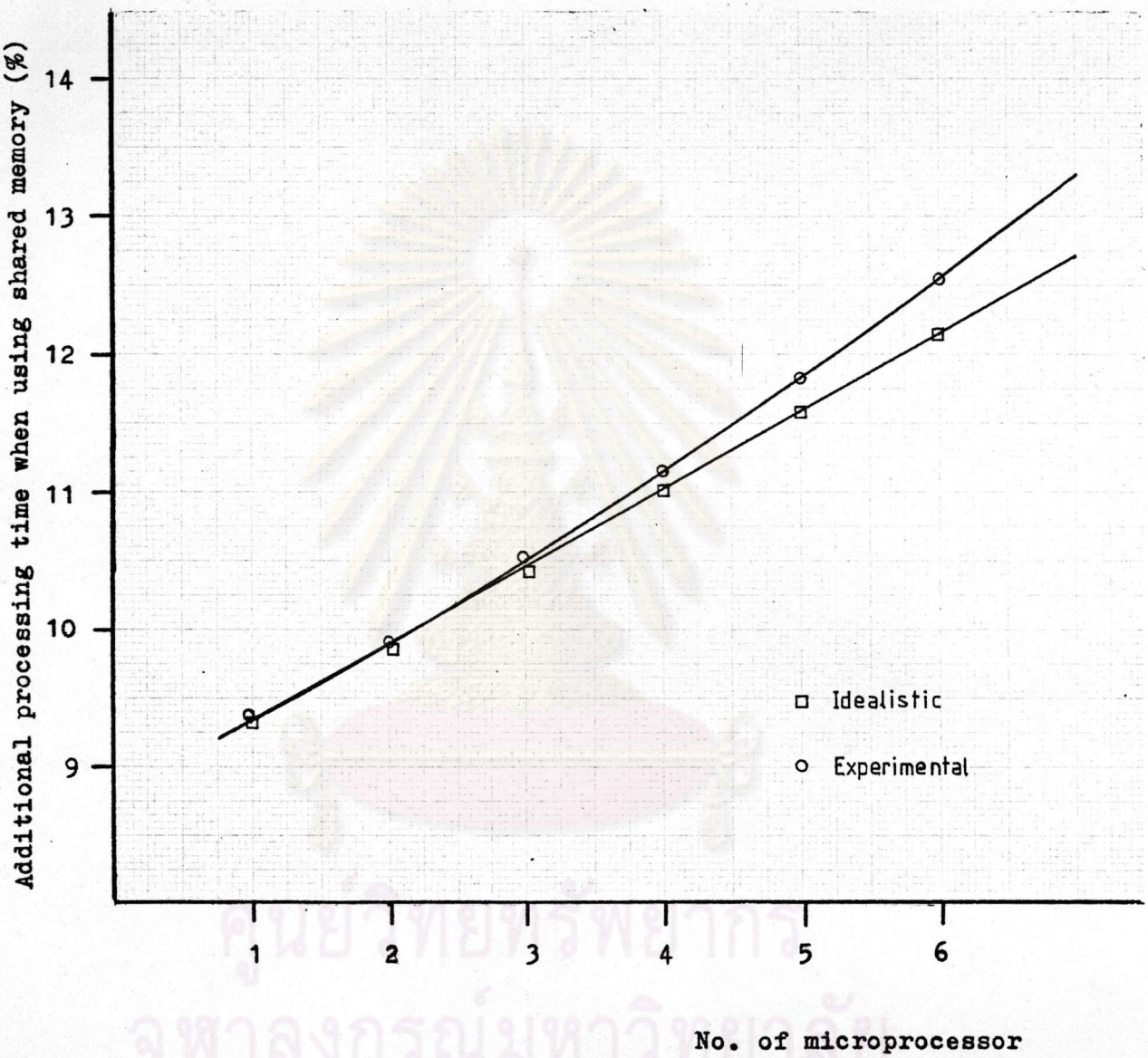
	เวลา[1] (microsec)	NUM (HEX)		จำนวนไบท์ที่อ่านได้เมื่อ		
			ใช้หน่วยความจำส่วนตัว	ไมโครฯ 1 ตัว ใช้หน่วยความจำร่วม	ไมโครฯ 2 ตัว ใช้หน่วยความจำร่วม	ไมโครฯ 3 ตัว ใช้หน่วยความจำร่วม
1	36513	FF	1388	1268	1264	1254
2	34366	F0	1306	1193	1189	1181
3	32074	E0	1219	1114	1110	1102
4	29784	D0	1132	1034	1031	1023
5	27492	C0	1045	954	951	944
6	25201	B0	958	875	872	866
7	22910	A0	870	795	792	787
8	20619	90	783	716	713	708
9	18328	80	696	636	634	629
10	16037	70	609	556	554	550
11	13746	60	522	477	475	472
12	11455	50	435	397	396	393
13	9164	40	348	318	316	314
14	6873	30	260	238	237	235
15	4582	20	173	158	158	157
16	2291	10	86	79	78	78

1. เวลาคำนวณจากสมการ $T = \frac{1}{1.7878} \times 256 \times NUM = 143.19\ NUM$ microsec.

ตาราง 4.4 เวลาที่ไมโครโปรเซสเซอร์ใช้ในการอ่านข้อมูลจากหน่วยความจำร่วม กรณีต่าง ๆ (microsec)

	ไมโครฯ ใช้ใช้หน่วยความจำส่วนตัว	ไมโครฯ 1 ตัว ใช้หน่วยความจำร่วม	ไมโครฯ 2 ตัว ใช้หน่วยความจำร่วม	ไมโครฯ 3 ตัว ใช้หน่วยความจำร่วม
1	26.306	28.796	28.887	29.117
2	26.314	28.806	28.903	29.099
3	26.312	28.792	28.895	29.105
4	26.311	28.804	28.888	29.114
5	26.308	28.818	28.909	29.123
6	26.306	28.801	28.900	29.110
7	26.333	28.818	28.927	29.110
8	26.333	28.797	28.919	29.123
9	26.333	28.818	28.909	29.138
10	26.333	28.844	28.948	29.158
11	26.333	28.818	28.939	29.123
12	26.333	28.854	28.928	29.148
13	26.333	28.818	29.000	29.185
14	26.435	28.878	29.000	29.147
15	26.486	29.000	29.000	29.185
16	26.639	29.000	29.371	29.372
ค่าเฉลี่ย	26.359	28.841	28.958	29.153
ส่วนเบี่ยงเบน	0.086	0.064	0.113	0.068
เวลาทำงานที่เพิ่มขึ้น	0%	9.42%	9.86%	10.60%

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูป 4.7 ความสัมพันธ์ระหว่างเวลาทำงานที่เพิ่มขึ้นกับจำนวนไมโคร-โปรเซสเซอร์ที่ใช้หน่วยความจำร่วม

ตาราง 4.5 เปรียบเทียบการใช้หน่วยความจำร่วมวิธีต่าง ๆ

วิธี	จำนวนไมโครฯ ที่ใช้หน่วยความจำร่วม	อัตราการรับ-ส่ง* ข้อมูล	จำนวน device ที่เพิ่มขึ้น	ความยุ่งยากในการสร้างและใช้งาน
DMA[1]	ไม่จำกัด	มากกว่า 1 Mbyte	1-2	มาก
TWO-PORT RAM[2]	2 ตัว	142.8 Kbyte	1-2	มาก
PETRIU (16)[3]	4 ตัว	86 Kbyte	5	มาก
POLCZYNSKI (15)[4]	8 ตัว	125 Kbyte**	4	น้อย
LOEWER (14)[5]	2 ตัว	71 Kbyte	9	ปานกลาง
HOJBERG (17)[6]	6 ตัว	80 Kbyte	7	มาก
วิทยานิพนธ์นี้[7]	6 ตัว	58 Kbyte	4	น้อย

1. ต้องใช้ DMA controller สำหรับไมโครฯ แต่ละแบบ
2. ใช้ได้กับการสื่อสารข้อมูลจำนวนไม่มาก และราคาแพง
3. ขั้นตอนในการสร้างยุ่งยากมากและใช้กับ priority พิเศษไม่ได้
4. ใช้ได้เฉพาะไมโครฯ ที่จังหวะเฟทช์และเอ็กซ์คิวยาวเท่ากัน
5. ใช้ได้กับไมโครฯ Z-80 และขยายระบบให้ใหญ่ขึ้นไม่ได้
6. ต้องใช้สัญญาณ clock ความถี่สูง (มากกว่า 10 MHz) 2 สัญญาณ
7. ใช้ได้กับไมโครฯ ทุกชนิดที่มีสภาวะ WAIT และสัญญาณบอกสภาวะเฟทช์

* สำหรับระบบที่มีไมโครโปรเซสเซอร์ Z-80 ไม่เกิน 4 ตัว ทำงานที่ 2 MHz

** ไมโครโปรเซสเซอร์ 1802 ที่ 2 MHz

ไม่เกิน 2% แต่อัตราการเปลี่ยนแปลงจะเพิ่มสูงขึ้นเมื่อจำนวนไมโครโปรเซสเซอร์ที่ใช้หน่วยความจำร่วมมีมากขึ้น เช่นการเพิ่มไมโครโปรเซสเซอร์จาก 5 ตัว เป็น 6 ตัว การเปลี่ยนแปลงของเวลาทำงานที่เพิ่มขึ้น จะเท่ากับ 40% ของการเปลี่ยนแปลงเมื่อเพิ่มไมโครโปรเซสเซอร์จาก 1 เป็น 4 ตัว ดังนั้น ระบบที่ออกแบบนี้ควรใช้งานกับไมโครโปรเซสเซอร์ไม่เกิน 4 ตัว ซึ่งเวลาการใช้หน่วยความจำร่วมจะแตกต่างกันไม่เกิน 2% แต่ถ้าจำเป็นอาจเพิ่มจำนวนได้ถึง 6 ตัว โดยยอมให้เวลาการใช้หน่วยความจำร่วมต่างกันไม่เกิน 4%

คุณสมบัติที่สำคัญของวงจร ARBITER ที่ออกแบบ เปรียบเทียบกับวิธีการอื่น ๆ ที่แสดงไว้ในตาราง 4.5 จะเห็นว่าวิธีการที่เสนอในวิทยานิพนธ์นี้ เหมาะสมที่สุดที่จะใช้ในระบบไมโครโปรเซสเซอร์แบบกระจาย เนื่องจากใช้ได้กับไมโครโปรเซสเซอร์หลายแบบ มีความยุ่งยากในการสร้างและใช้งานน้อยมาก โดยสามารถรวมกลุ่มไมโครโปรเซสเซอร์ได้สูงสุดถึง 6 ตัว และอัตราการรับส่งข้อมูลต่ำกว่าวิธีอื่นที่ยุ่งยากและมีข้อจำกัดเพียงไม่เกิน 30%

4.3 ตัวอย่างการประยุกต์ใช้งาน

เพื่อเป็นแนวทางในการนำระบบไมโครโปรเซสเซอร์แบบกระจายที่สร้างขึ้นไปใช้งาน ขอยกตัวอย่างระบบเก็บรวบรวมข้อมูล (Data acquisition system) ที่มีลักษณะดังนี้

1. ใช้ไมโครคอมพิวเตอร์ Apple II เป็นโฮสคอมพิวเตอร์

2. ใช้ระบบไมโครโปรเซสเซอร์แบบกระจายที่ใช้หน่วยความจำร่วม เป็นระบบสื่อสารฟรอนด์เอนด์ (Front-end communication system) สำหรับโฮสคอมพิวเตอร์

3. เก็บข้อมูลจากรีโมทเทอร์มินัล (Remote Terminal) 32 ตัว โดยที่รีโมทเทอร์มินัลแต่ละตัวจะเก็บข้อมูลจากวงจร Analog to Digital Converter ขนาด 8 บิท จำนวน 32 วงจร ระบบนี้จึงเก็บข้อมูลได้รวมทั้งสิ้น 1024 จุด

จะเห็นว่าระบบเก็บรวบรวมข้อมูลที่ยกขึ้นเป็นตัวอย่างนี้มีขนาดใหญ่ การสร้างระบบทั้งหมดขึ้นจริงต้องใช้เวลาและเงินทุนจำนวนมาก รวมทั้งอยู่นอกเหนือวัตถุประสงค์

และขอบเขตของการวิจัยนี้ ดังนั้นจึงได้จำลองการทำงานบางส่วนขึ้นแทนระบบจริง ทั้งนี้โดยมีจุดประสงค์เพื่อยกตัวอย่างการใช้งานระบบไมโครโปรเซสเซอร์แบบกระจายที่สร้างขึ้น โดยเน้นการสื่อสารข้อมูลระหว่างไมโครโปรเซสเซอร์ผ่านหน่วยความจำร่วมเป็นสำคัญ และจะขอกล่าวถึงรายละเอียดการทำงานของระบบเก็บรวบรวมข้อมูลที่ยกเป็นตัวอย่างก่อน แล้วจึงกล่าวถึงระบบที่จำลองขึ้นต่อไป

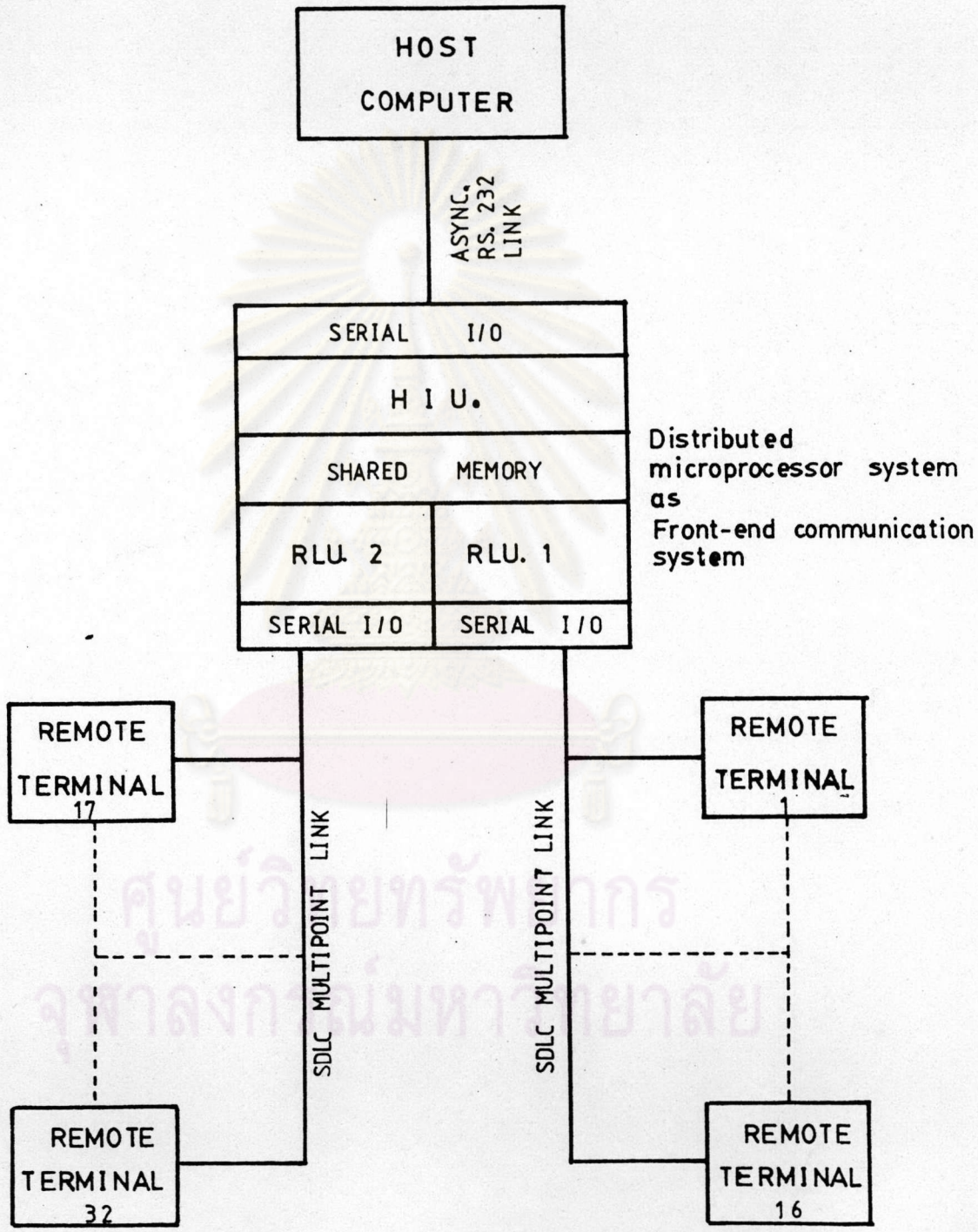
4.3.1 โครงสร้างของระบบเก็บข้อมูลตัวอย่าง

ระบบนี้มีส่วนประกอบหลัก 3 ส่วนคือ โฮสคอมพิวเตอร์, ระบบสื่อสารฟรอนท์เอนด์ และรีโมทเทอร์มินัล ดังรูป 4.8 โดยแต่ละส่วนมีรายละเอียดการทำงานดังนี้

4.3.1.1 โฮสคอมพิวเตอร์

มีหน้าที่ติดต่อกับผู้ใช้ โดยผู้ใช้จะป้อนข้อมูลหมายเลขของรีโมทเทอร์มินัลที่ต้องการข้อมูล และไบท์คำสั่งที่ต้องการส่งไปยังรีโมทเทอร์มินัลตัวนั้นให้กับโฮสคอมพิวเตอร์ ซึ่งโฮสคอมพิวเตอร์จะส่งไบท์คำสั่งไปยังรีโมทเทอร์มินัล พร้อมกับรับข้อมูลจากรีโมทเทอร์มินัลนั้นมาแสดงบนจอภาพ โดยมีระบบสื่อสารฟรอนท์เอนด์เป็นตัวกลาง ไบท์คำสั่งที่ส่งไปจะมีค่าระหว่าง 0 ถึง 128 ซึ่งรีโมทเทอร์มินัลใช้ถอดรหัสตีความเป็นคำสั่งให้ปฏิบัติงานต่าง ๆ รายละเอียดของไบท์คำสั่งจะไม่กล่าวถึงในที่นี้ ผู้ใช้อาจสั่งให้โฮสคอมพิวเตอร์นำข้อมูลจากรีโมทเทอร์มินัลมาคำนวณค่าต่าง ๆ นอกเหนือไปจากการแสดงผลเพียงอย่างเดียวก็ได้ตามแต่ความประสงค์ของผู้ใช้

ไมโครคอมพิวเตอร์ Apple II ที่ใช้เป็นโฮสคอมพิวเตอร์ จะต้องมีแผ่นวงจร Serial Interface Card ทำหน้าที่รับส่งข้อมูลกับระบบสื่อสารฟรอนท์เอนด์ร่วมอยู่ด้วย โดยรายละเอียดการรับส่งข้อมูลระหว่างโฮสคอมพิวเตอร์กับระบบสื่อสารฟรอนท์เอนด์จะกล่าวในภายหลัง ส่วนข้อมูลจากรีโมทเทอร์มินัลนั้นจะเก็บไว้ที่หน่วยความจำแอดเดรส $C000 ถึง $C020 ตัวอย่างการรับคำสั่งจากผู้ใช้และการแสดงข้อมูลบนจอภาพ แสดงไว้ในรูป 4.9 และ 4.10 ตามลำดับ



รูป 4.8 ระบบเก็บรวบรวมข้อมูลที่ยกเป็นตัวอย่าง

```
TERMINAL NUMBER # ?
2

COMMAND CODE?
12

IN PROCESS.
```

รูป 4.9 ตัวอย่างการรับคำสั่งจากผู้ใช้ของโฮสคอมพิวเตอร์

REMOTE TERMINAL 2

SENSOR#	VALUE	SENSOR#	VALUE
1	128	17	42
2	256	18	9
3	34	19	8
4	22	20	17
5	10	21	59
6	14	22	18
7	16	23	136
8	18	24	244
9	0	25	194
10	9	26	43
11	27	27	71
12	56	28	53
13	14	29	19
14	4	30	4
15	122	31	12
16	254	32	10

รูป 4.10 ตัวอย่างการแสดงข้อมูลบนจอภาพของโฮสคอมพิวเตอร์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.3.1.2 ระบบสื่อสารฟรอนท์เอนด์

เป็นระบบไมโครโปรเซสเซอร์แบบกระจายที่ประกอบด้วยไมโครโปรเซสเซอร์ 3 ตัว ตัวหนึ่งจัดการสื่อสารกับโฮสคอมพิวเตอร์ เรียกว่า Host Interface Unit (HIU.) อีกสองตัวที่เหลือจัดการสื่อสารกับรีโมทเทอร์มินัล เรียกว่า Remote Link Unit (RLU.) โดยที่ RLU. แต่ละตัวติดต่อกับรีโมทเทอร์มินัล 16 ตัว กล่าวคือ RLU.1 ติดต่อกับรีโมทเทอร์มินัล 1-16 และ RLU.2 ติดต่อกับรีโมทเทอร์มินัล 17-32 การสื่อสารระหว่าง HIU. กับ RLU. กระทำผ่านหน่วยความจำร่วมและการทำงานของไมโครโปรเซสเซอร์ทั้งสามสรุปได้ดังนี้

4.3.1.2.1 Remote Link Unit (RLU.)

RLU. แต่ละตัวจัดการสื่อสารกับรีโมทเทอร์มินัล 16 ตัว โดยมีการรับส่งข้อมูลแบบอนุกรมซิงโครนัสในลักษณะ half duplex-multipoint ใช้ SDLC protocol มี RLU. เป็น primary station และรีโมทเทอร์มินัลเป็น secondary station จึงสรุปหน้าที่ต่าง ๆ ของ RLU. ได้ดังนี้

ก. Line control ดำเนินการรับและส่งข้อมูลระหว่างรีโมทเทอร์มินัลกับระบบสื่อสารฟรอนท์เอนด์ โดยใช้ SDLC protocol ดังรูป 4.11

OPENING FLAG (F)	ADDRESS FIELD (A)	CONTROL FIELD (C)	INFORMATION FIELD (I)	FRAME CHECK SEQUENCE (FCS)	CLOSING FLAG (F)
01111110	8 BITS	8 BITS	VARIABLE LENGTH (ONLY IN I FRAMES)	16 BITS	01111110

รูป 4.11 SDLC FRAME FORMAT

ในการส่งข้อมูลด้วย SDLC protocol แต่ละ frame จะประกอบด้วย FLAG FIELD 8 บิทแรกถัดมาเป็นแอดเดรสของรีโมทเทอร์มินัลขนาด 8 บิท มีค่าตั้งแต่ 1 ถึง 32 8 บิทถัดไปเป็น CONTROL FIELD ซึ่งแบ่งเป็น 2 แบบคือ information frame ใช้แสดงการส่งข้อมูล และ supervisory frame ใช้แสดงการส่งคำสั่ง ถัดไปเป็นส่วนของข้อมูล ในกรณีที่รีโมทเทอร์มินัลส่งข้อมูลมายัง RLU. ส่วนนี้จะยาว 256 บิท เป็นข้อมูลจำนวน 32 ไบท์เรียงกันตามลำดับ แต่ถ้าเป็นคำสั่งจาก RLU. ไปยังรีโมท-

เทอร์มินัลส่วนนี้จะเป็นไบท์คำสั่งยาวเพียง 8 บิท สองส่วนสุดท้ายของ frame จะเป็น FRAME CHECK SEQUENCE และ CLOSING FLAG มีไว้ใช้ตรวจสอบความถูกต้องของข้อมูลที่ได้รับ

ช. Polling RLU. จะทำหน้าที่จัดลำดับให้รีโมทเทอร์มินัลแต่ละตัวผลัดกันส่งข้อมูลผ่านสายส่งชุดเดียวกัน เพื่อป้องกันการสับสนของข้อมูล ซึ่งรีโมทเทอร์มินัลจะส่งข้อมูลได้ต่อเมื่อได้รับการโพลลิ่ง (polling) จาก RLU. เท่านั้น ลักษณะการโพลลิ่งจะเป็นแบบ sequencial กล่าวคือ RLU. จะสอบถามรีโมทเทอร์มินัลตั้งแต่ตัวแรกจนถึงตัวที่สิบหกตามลำดับ แล้ววนกลับมาถามตัวแรกใหม่อีกครั้ง และในการโพลลิ่งแต่ละครั้ง ถ้ารีโมทเทอร์มินัลมีข้อมูลพร้อมที่จะส่ง จะส่งข้อมูลนั้นตอบมาทันที แต่ถ้ายังมีข้อมูลไม่พร้อมรีโมทเทอร์มินัลจะตอบปฏิเสธการส่งข้อมูล ซึ่ง RLU. หลังจากได้รับข้อมูลหรือคำปฏิเสธแล้วจะโพลลิ่งตัวถัดไปทันที ดังรูป 4.12

เวลาที่ RLU. ใช้ในการโพลลิ่งรีโมทเทอร์มินัลแต่ละตัว ในกรณีที่รีโมท-เทอร์มินัลมีข้อมูลพร้อมที่จะส่ง จะเท่ากับผลรวมของตัวประกอบต่าง ๆ ดังนี้

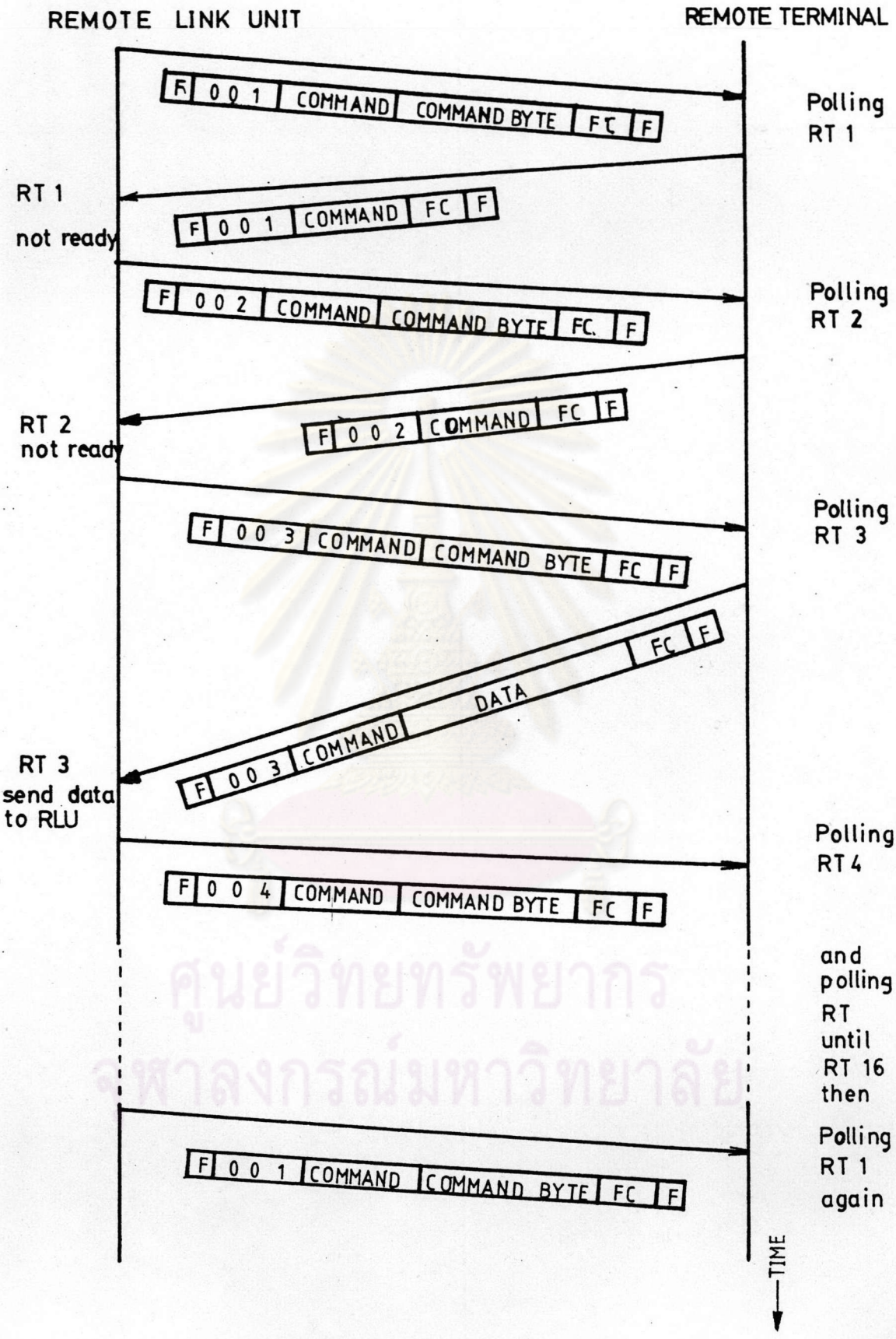
1. เวลาที่ RLU. ใช้ส่ง polling command ไปยังรีโมทเทอร์มินัล เท่ากับเวลาที่ใช้ส่งข้อมูลจำนวน 56 บิท ตาม format ในรูป 4.13 ดังนั้น ถ้าส่งข้อมูลด้วยอัตรา 1200 Baud ต้องใช้เวลาส่งคำสั่ง 0.048 วินาที

2. เวลาที่รีโมทเทอร์มินัลใช้ส่งข้อมูลจำนวน 32 ไบท์ กลับมายัง RLU. เท่ากับเวลาที่ใช้ในการส่งข้อมูลจำนวน 304 บิท ตาม format ในรูป 4.14 ดังนั้น ถ้าส่งข้อมูลด้วยอัตรา 1200 Baud ต้องใช้เวลาส่งข้อมูล 0.253 วินาที

3. เวลาที่ RLU. ใช้ทำโปรแกรมสร้าง polling command และ SDLC protocol ถ้าโปรแกรมนี้ยาวไม่เกิน 1000 คำสั่ง และไมโครโปรเซสเซอร์ทำงานที่ clock 2.5 MHz ต้องใช้เวลาทำโปรแกรม 0.006 วินาที

4. เวลาที่รีโมทเทอร์มินัลใช้ตีความคำสั่งและสร้าง SDLC protocol เพื่อส่งข้อมูลจำนวน 32 ไบท์ กลับมาให้ RLU. กำหนดให้ใช้เวลาเท่ากับเวลาที่ RLU. ใช้ในการทำโปรแกรมคือ 0.006 วินาที

5. เวลาที่สัญญาณใช้เดินทางระหว่าง RLU. กับรีโมทเทอร์มินัล ตามปกติสัญญาณเดินทางได้ 1 กิโลเมตร ภายในเวลาประมาณ 9.3 microsec (20) ดังนั้น



รูป 4.12 การโพลลิ่งรีโมทเทอร์มินัลต่าง ๆ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ถ้ารีโมทเทอร์มินัลกระจายอยู่ในพื้นที่ไม่เกิน 1 กิโลเมตร จากระบบสื่อสารฟรอนท์เอนด์ เวลาที่สัญญาณใช้เดินทางจึงน้อยมากเมื่อเทียบกับเวลาที่ใช้ในข้อ 1-4 จึงไม่นำมาคิด

เมื่อรวมเวลาที่คำนวณได้ตามข้อ 1-5 จะได้เวลาที่ RLU. ใช้ในการเก็บข้อมูลจากรีโมทเทอร์มินัลแต่ละตัวเท่ากับ 0.313 วินาที ดังนั้น ในการเก็บข้อมูลจากรีโมทเทอร์มินัลทั้ง 16 ตัว แต่ละรอบ RLU. ต้องใช้เวลา 5.0 วินาที

FLAG 8 BIT	ADDRESS 8 BIT	CONTROL 8 BIT	COMMAND BYTE	CRC FRAME CHECK 16 BIT	FLAG 8 BIT

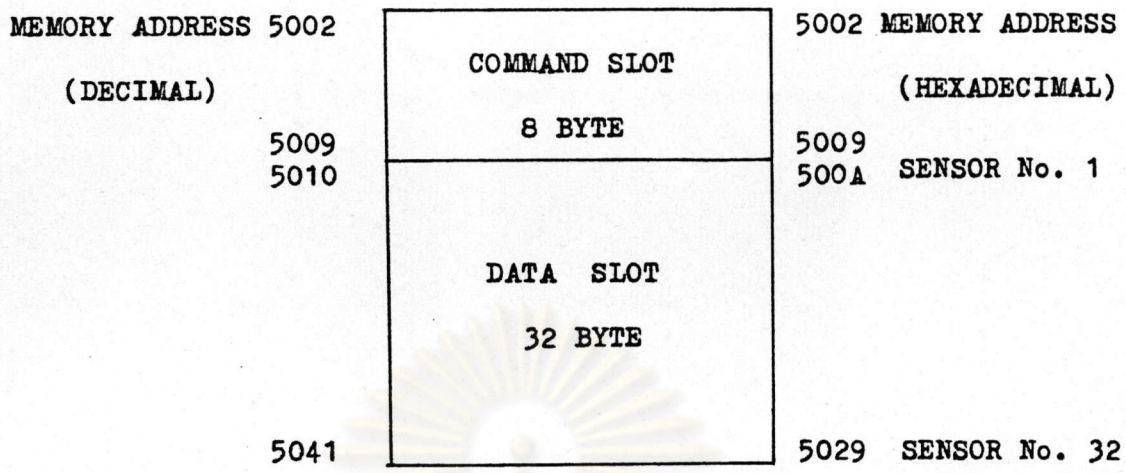
รูป 4.13 RLU. POLLING FRAME

FLAG 8 BIT	ADDRESS 8 BIT	CONTROL 8 BIT	DATA 32 BYTE(256 BIT)	CRC FRAME CHECK 16 BIT	FLAG 8 BIT

รูป 4.14 REMOTE TERMINAL DATA FRAME

ค. HIU. communication การสื่อสารข้อมูลระหว่าง RLU. และ HIU. กระทำผ่านหน่วยความจำร่วมอย่างอะซิงโครนัส กล่าวคือ ต่างเขียนและอ่านข้อมูลจากหน่วยความจำร่วมโดยอิสระไม่ขึ้นต่อกัน เปรียบเทียบได้กับการติดประกาศ และการอ่านประกาศบนแผ่นป้าย ที่ผู้ติดประกาศและผู้อ่านประกาศไม่จำเป็นต้องทำงานอย่างสัมพันธ์กัน

พื้นที่ในหน่วยความจำร่วมขนาด 2K ไบท์ เริ่มต้นตั้งแต่แอดเดรส 5000 จะแบ่งเป็น 32 ส่วน แต่ละส่วนจัดสรรให้กับรีโมทเทอร์มินัลแต่ละตัว และแบ่งออกเป็น 2 ช่อง คือ ช่องข้อมูล (data slot) และช่องคำสั่ง (command slot) ดังตัวอย่างในรูป 4.15



รูป 4.15 พื้นที่ในหน่วยความจำร่วมที่จัดสรรให้รีโมทเทอร์มินัล 1

พื้นที่หน่วยความจำร่วมที่จัดสรรให้รีโมทเทอร์มินัลแต่ละตัวมีขนาด 40 ไบท์ โดยเป็นช่องคำสั่ง 8 ไบท์ ซึ่งในที่นี้ใช้งานเพียงไบท์เดียว ส่วนที่เหลือสำรองไว้เพิ่มจำนวนคำสั่งให้มากขึ้นในอนาคต และช่องข้อมูล 32 ไบท์ ใช้บรรจุข้อมูลที่ส่งมาจากรีโมทเทอร์มินัลต่าง ๆ เพื่อให้ HIU. เลือกส่งไปยังโฮสคอมพิวเตอร์อีกต่อหนึ่ง การจัดสรรพื้นที่ในหน่วยความจำร่วมสำหรับ RLU.1 และ 2 แสดงไว้ในตาราง 4.6 และ 4.7 ตามลำดับ

เมื่อใดก็ตามที่ RLU. ได้รับข้อมูลจากรีโมทเทอร์มินัล และตรวจสอบความถูกต้องของข้อมูลแล้ว RLU. จะนำข้อมูลนั้นไปเรียงใส่ไว้ในช่องข้อมูลของรีโมทเทอร์มินัลนั้น ในทำนองเดียวกันในการโพลลิ่งทุกครั้ง RLU. จะอ่านข้อมูลจากช่องคำสั่งส่งไปยังรีโมทเทอร์มินัลที่เป็นเจ้าของช่องคำสั่งนั้น

หน้าที่ทั้งสามของ RLU. สรุปเขียนเป็นแผนผังการทำงานได้ดังรูป 4.16

4.3.1.2.2 Host Interface Unit (HIU.)

เป็นไมโครโปรเซสเซอร์ที่จัดการสื่อสารระหว่างระบบสื่อสารฟรอนท์เอนด์กับโฮสคอมพิวเตอร์ โดย HIU. จะรับไบท์คำสั่งมาใส่ไว้ในช่องคำสั่งของรีโมทเทอร์มินัลที่ผู้ใช้เลือกและส่งข้อมูลจากช่องข้อมูลกลับไปให้โฮสคอมพิวเตอร์ตามความประสงค์ของผู้ใช้ หน้าที่หลักของโฮสคอมพิวเตอร์จึงมีเพียง 2 ประการคือ

ตาราง 4.6 การจัดสรรพื้นที่ในหน่วยความจำร่วมสำหรับรีโมทเทอร์มินัล 1 ถึง 16 ซึ่งติดต่อกับ RLU. 1

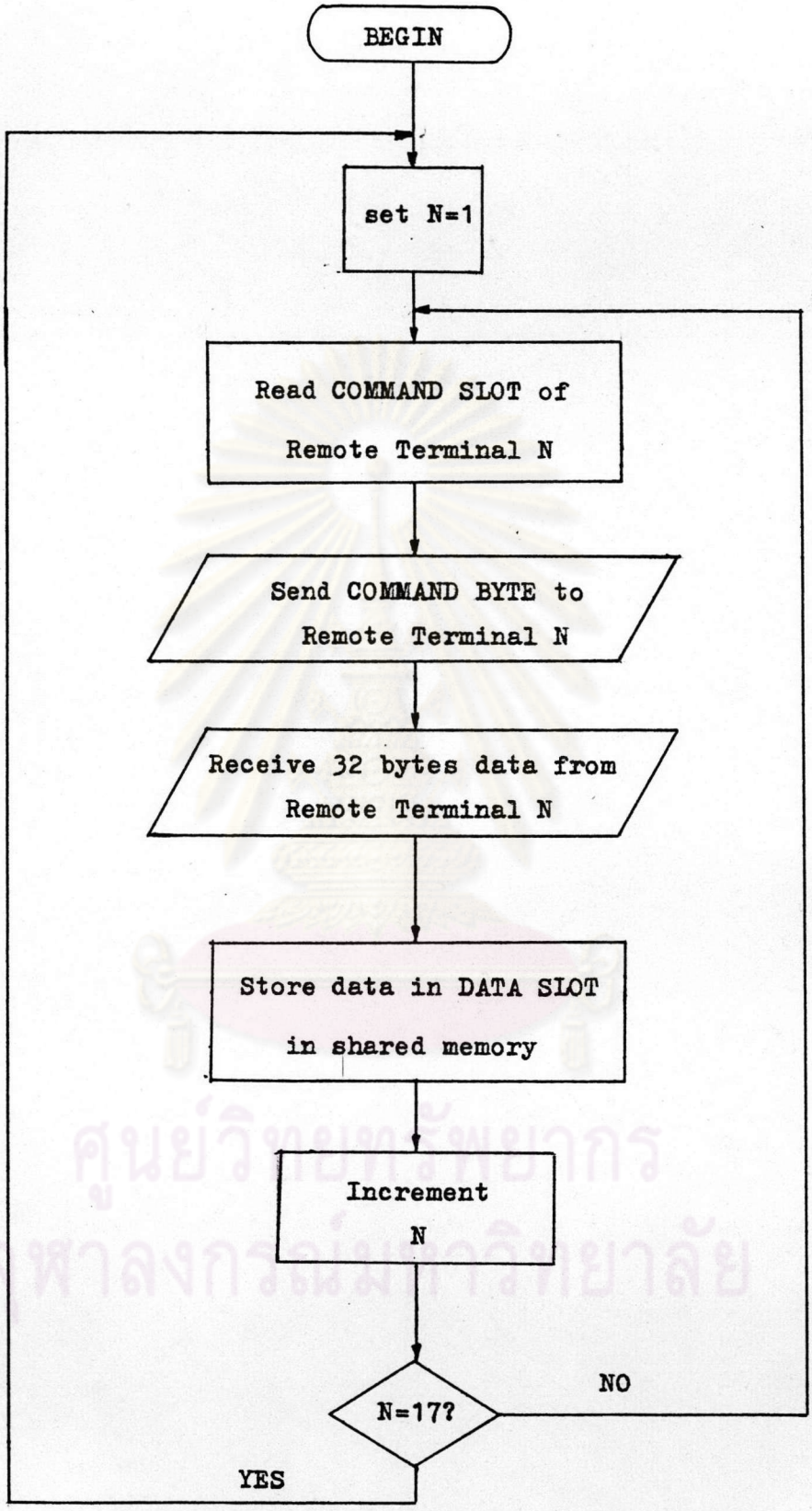
REMOTE TERMINAL	COMMAND SLOT BEGINNING ADDRESS	COMMAND SLOT TERMINATING ADDRESS	DATA SLOT BEGIN SENSOR# 1	DATA SLOT END SENSOR# 32
1	5002	5009	5010	5041
2	5042	5049	5050	5081
3	5082	5089	5090	5121
4	5122	5129	5130	5161
5	5162	5169	5170	5201
6	5202	5209	5210	5241
7	5242	5249	5250	5281
8	5282	5289	5290	5321
9	5322	5329	5330	5361
10	5362	5369	5370	5401
11	5402	5409	5410	5441
12	5442	5449	5450	5481
13	5482	5489	5490	5521
14	5522	5529	5530	5561
15	5562	5569	5570	5601
16	5602	5609	5610	5641

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตาราง 4.7 การจัดสรรพื้นที่ในหน่วยความจำร่วมสำหรับรีโมทเทอร์มินัล 17 ถึง 32 ซึ่งติดต่อกับ RLU. 2

REMOTE TERMINAL	COMMAND SLOT BEGINNING ADDRESS	COMMAND SLOT TERMINATING ADDRESS	DATA SLOT BEGIN SENSOR # 1	DATA SLOT END SENSOR # 32
17	5642	5649	5650	5681
18	5682	5689	5690	5721
19	5722	5729	5730	5761
20	5762	5769	5770	5801
21	5802	5809	5810	5841
22	5842	5849	5850	5881
23	5882	5889	5890	5921
24	5922	5929	5930	5961
25	5962	5969	5970	6001
26	6002	6009	6010	6041
27	6042	6049	6050	6081
28	6082	6089	6090	6121
29	6122	6129	6130	6161
30	6162	6169	6170	6201
31	6202	6209	6210	6241
32	6242	6249	6250	6281

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูป 4.16 แผนผังการทำงานของ RLU.

1. Host Computer Communication การสื่อสารข้อมูลระหว่างโฮสคอมพิวเตอร์และ HIU. เป็นแบบอนุกรมอซิงโครนัส RS232-half duplex ด้วยอัตรา 1200 Baud โดยโฮสคอมพิวเตอร์เป็นผู้เริ่มต้นการติดต่อเสมอ กล่าวคือ HIU. จะรอคำสั่งจากโฮสคอมพิวเตอร์ แล้วจึงส่งข้อมูลที่โฮสคอมพิวเตอร์ต้องการกลับไป ข้อมูลที่ส่งมาจากโฮสคอมพิวเตอร์ประกอบด้วย หมายเลขรีโมทเทอร์มินัล และไบท์คำสั่งซึ่ง HIU. เมื่อได้รับข้อมูลนี้แล้ว จะนำไบท์คำสั่งไปใส่ไว้ในช่องคำสั่งของรีโมทเทอร์มินัลที่ผู้ใช้เลือกพร้อมกับส่งข้อมูลจากช่องข้อมูลกลับไปให้โฮสคอมพิวเตอร์ แล้วกลับมารอรับคำสั่งใหม่อีกครั้ง

Format ที่ใช้ในการรับส่งข้อมูลระหว่างโฮสคอมพิวเตอร์และ HIU. แสดงไว้ในรูป 4.17

START BIT		ASCII-CODED DATA	STOP BIT
1	1	7	2

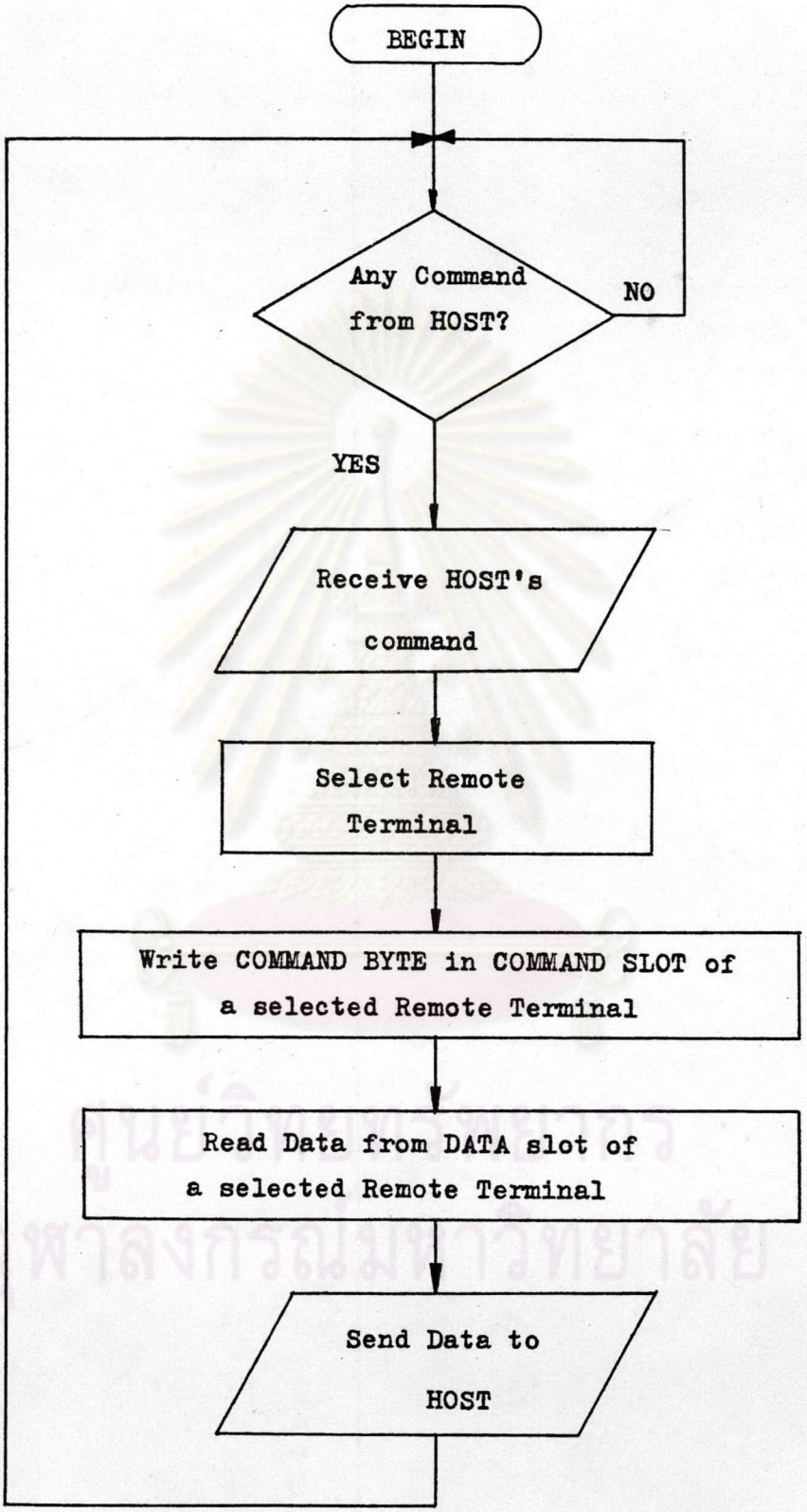
รูป 4.17 แบบฟอร์มการสื่อสารข้อมูลระหว่าง HIU. กับโฮสคอมพิวเตอร์

2. RLU. Communication การสื่อสารกับ RLU. กระทำผ่านหน่วยความจำร่วมในลักษณะเดียวกับที่ได้อธิบายแล้วในหัวข้อ 4.3.1.2.1 ค HIU. Communication แต่มีทิศทางตรงข้ามกล่าวคือ HIU. จะส่งข้อมูลให้ RLU. ในช่องคำสั่ง และรับข้อมูลจาก RLU. จากช่องข้อมูล ในขณะที่ RLU. รับข้อมูลจาก HIU. ผ่านช่องคำสั่ง และส่งข้อมูลไป HIU. ผ่านช่องข้อมูล ในลักษณะนี้จะเห็นได้ว่าจะมีข้อมูลชุดล่าสุดจากรีโมทเทอร์มินัลพร้อมที่จะให้ HIU. ส่งไปยังโฮสคอมพิวเตอร์ในทันทีที่ได้รับคำสั่ง

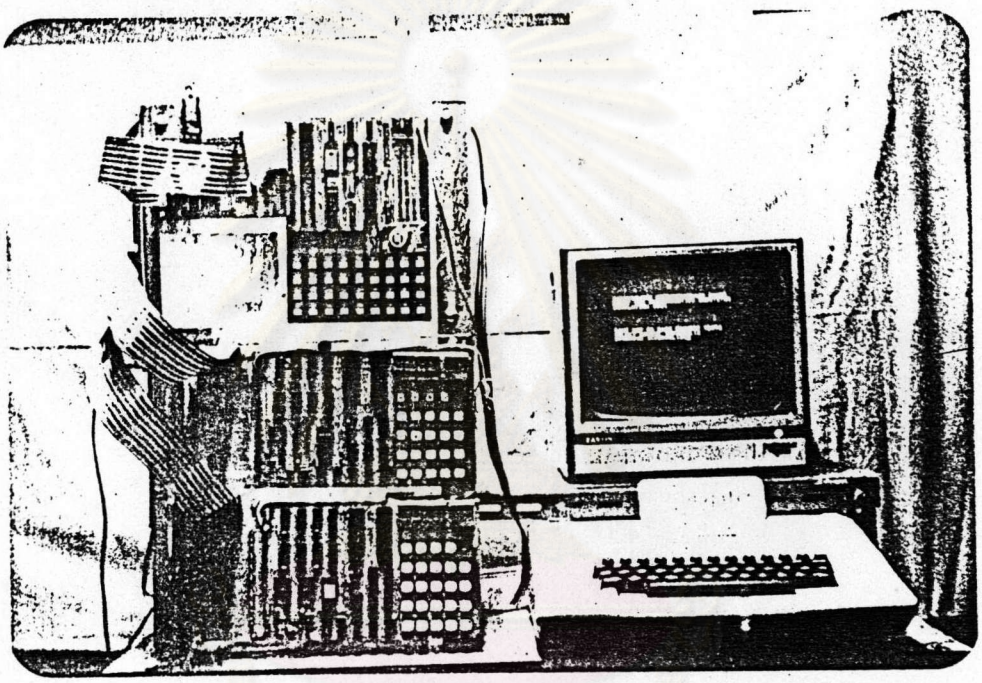
แผนผังการทำงานของ HIU. แสดงไว้ในรูป 4.18

4.3.2 การจำลองระบบเก็บรวบรวมข้อมูลตัวอย่าง

ระบบเก็บรวบรวมข้อมูลที่ยกเป็นตัวอย่างในหัวข้อ 4.3.1 ได้จำลองขึ้นดังรูป 4.19 โดยมีส่วนประกอบดังนี้



รูป 4.18 แผนผังการทำงานของ HIU.



รูป 4.19 ระบบเก็บรวบรวมข้อมูลจำลองที่สร้างขึ้น

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.3.2.1 โฮสคอมพิวเตอร์ ใช้ไมโครคอมพิวเตอร์ Apple II โดยมี Serial Interface Card ติดตั้งไว้ที่ slot 1 และทำหน้าที่ต่าง ๆ เช่นเดียวกับโฮสคอมพิวเตอร์ที่กล่าวไว้ในหัวข้อ 4.3.1.1 ทุกประการ โปรแกรมของโฮสคอมพิวเตอร์แสดงไว้ในภาคผนวก ก.

4.3.2.2 ระบบสื่อสารฟรอนท์เอนด์จำลอง ใช้ระบบไมโครโปรเซสเซอร์แบบกระจายที่สร้างขึ้น โดยแบ่งการทำงานเป็น 2 ส่วนคือ

ก. Host Interface Unit (HIU.) เป็นไมโครคอมพิวเตอร์บอร์ดเดียว (single board microcomputer) รุ่น MPF-1 โดยมีวงจรสื่อสารข้อมูลแบบอนุกรมเพิ่มเติมขึ้น รายละเอียดของวงจรสื่อสารข้อมูลแบบอนุกรมแสดงไว้ในภาคผนวก ง. HIU. จำลองทำหน้าที่เช่นเดียวกับ HIU. ที่กล่าวในหัวข้อ 4.3.1.2.2 ทุกประการ และโปรแกรมของ HIU. แสดงไว้ในภาคผนวก ช.

ข. Remote Link Unit and Remote Terminal Simulator เป็นไมโครคอมพิวเตอร์บอร์ดเดียวรุ่น JZ-80 2 ตัว ตัวหนึ่งจำลองการทำงานของ RLU. 1 และรีโมทเทอร์มินัล 1-16 อีกตัวหนึ่งจำลองการทำงานของ RLU. 2 และรีโมทเทอร์มินัล 17-32

ในการจำลองกำหนดว่า

1. รีโมทเทอร์มินัลทุกตัวมีข้อมูลพร้อมที่จะส่งตลอดเวลา
2. การรับส่งข้อมูลระหว่าง RLU. กับรีโมทเทอร์มินัลทุกครั้งไม่มีความผิดพลาดเกิดขึ้น

จากข้อกำหนดนี้สรุปได้ว่า ในเวลาทุก ๆ 5 วินาที RLU. จะเขียนข้อมูลชุดใหม่ลงในช่องข้อมูลของหน่วยความจำร่วม ดังนั้น ในระบบจำลองนี้จึงแทนหน้าที่การ polling และ line control ของ RLU. ด้วยการวนเวียนสร้างตัวเลขอย่างสุ่ม ครั้งละ 32 ไบท์ แล้วนำไปใส่ในช่องข้อมูลของรีโมทเทอร์มินัลแต่ละตัวจนครบ 32 ตัว ภายในเวลาเท่ากับเวลาที่ใช้ในการโพลลิ่งและรับข้อมูลจากรีโมทเทอร์มินัล ข้อมูลที่ปรากฏในช่องข้อมูลจึงมีการเปลี่ยนแปลงในอัตราใกล้เคียงที่คำนวณจากระบบจริง ส่วนหน้าที่อื่น ๆ ของ RLU. คงเป็นไปตามที่กล่าวไว้ในหัวข้อ 4.3.1.2.1 ทุกประการ

การทดสอบการทำงานของระบบเก็บข้อมูลจำลองที่สร้างขึ้น ปรากฏว่าระบบทำงานได้อย่างถูกต้อง โดยมีข้อมูลชุดใหม่มาแทนที่ชุดเก่าในช่องข้อมูลในหน่วยความจำร่วมทุก 3.9 วินาที และสามารถส่งข้อมูลที่สร้างขึ้นโดย RLU. จำลองผ่าน HIU. ไปแสดงผลบนจอภาพของโฮสคอมพิวเตอร์ได้อย่างถูกต้อง

การใช้ระบบไมโครโปรเซสเซอร์แบบกระจายทำหน้าที่ระบบฟรอนท์เอนด์นับเป็นพัฒนาการอีกก้าวหนึ่งนับจากการใช้วงจร hard-wire และไมโครโปรเซสเซอร์ทำหน้าที่ระบบฟรอนท์เอนด์ของโฮสคอมพิวเตอร์ในระบบเก็บรวบรวมข้อมูลขนาดใหญ่ โดยมีวัตถุประสงค์เพื่อลดภาวะการสื่อสารของโฮสคอมพิวเตอร์ให้น้อยลง การใช้ระบบไมโครโปรเซสเซอร์แบบกระจายที่ออกแบบในวิทยานิพนธ์นี้ ทำหน้าที่ดังกล่าว จะเพิ่มขีดความสามารถของระบบให้สูงขึ้น พร้อมกับลดความยุ่งยากของซอฟท์แวร์ลงโดยค่าใช้จ่ายด้านฮาร์ดแวร์เพิ่มขึ้นเพียงเล็กน้อย ยกตัวอย่างเช่น ถ้าใช้ไมโครโปรเซสเซอร์ตัวเดียวทำงานนี้ เวลาในการเก็บรวบรวมข้อมูลจะเพิ่มขึ้นเป็นสองเท่าและซอฟท์แวร์จะยาวกว่าไม่น้อยกว่าสองเท่า ซึ่งหมายถึงการพัฒนาซอฟท์แวร์ย่อมยุ่งยากมากขึ้นด้วย ข้อดีอีกประการหนึ่งคือ การเพิ่มหน้าที่ต่าง ๆ ให้กับระบบฟรอนท์เอนด์ที่ใช้ระบบไมโครโปรเซสเซอร์แบบกระจายทำได้ง่ายกว่า เพราะในระบบที่ออกแบบไว้ ใช้ความสามารถเพียงบางส่วนของไมโครโปรเซสเซอร์แต่ละตัวเท่านั้น จึงเป็นการสมควรที่จะพัฒนาระบบไมโครโปรเซสเซอร์แบบกระจายที่ออกแบบนี้ไปใช้งานอย่างจริงจังต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย