เอกสารอ้างอิง

1. AT&T, "ISS'87 AT&T Technical Papers," Phoenix, Arizona, pp. 1-126, March 15, 1987
2. CCITT, "Integrated Services Digital Network(ISDN):Recommendations of The Series I," CCITT Red Book, Vol. III, Fascicles III.5, Int. Telecomm. Union, Geneva, Switzerland, 1985,
3. Special Issue on Integrated Services Digital Network ISDN, SIEMENS Telecom Report, Vol. 8, April 1985
4. Special Issue on Integrated Services Digital Network: Recommendation and Field Trials-I, IEEE. Journal on Selected Areas in Communications, Vol. SAC-4, No. 3, May 1986
5. Special Issue on ISDN,NEC. Res. and Develop.,1987
6. Lyhne P., D. Gneiting, "ISDN-A Basic for Enhanced and New Services for the Next Decades," Proceeding of ISS'87, pp. 875-879, Phoenix, Arizona USA, March 15-20, 1987.
7. Aldershlan Hrair, "ISDN Standards Evolution," AT&T Technical Journal, Vol. 65, Issue 1, Jan.-Feb. 1986.
8. Rosenbrock Karl Hienz, "ISDN - a Logical Evolution of the Digital Telephone Network," 1986 Yearbook of the Deutsche Bundespost, West Germany, 1986.
9. Special Issue on Integrated Services Digital Network - II, IEEE. Journal on Selected Areas in Communications, Vol. SAC-4, No. 8, November 1986
10. Northern Telecom, "ISDN Basic Rate Access : User-Network Interface Specification, Canada 1986
11. Adolphs A.,P. Wagner, T. Alvestad, "Subsets Terminals and Terminal Adapters for the Public ISDN," Electrical Communication, Vol 61,No. 1, pp. 72-80, 1987.
12. Ishizaki Y, et al., "ISDN Terminal," Nec. Res. & Develop. Special Issue on "ISDN"," pp. 19-37, 1987

13. Geiger G., L. lerach, "ISDN-Oriented Modular VLSI Chip Set for Central-Office and PABX Applications," IEEE. Journal on Selected Areas in Communication," Vol. SAC-4, No. 8, pp. 1268-1274, November 1986.

14. Fischer W.,E.H. Goeldner, "Performance of the ISDN User-Network Interface for Signalling and Packetized User-Data Transfer ," Proceeding of ISS'87, pp. 286-290, Phoenix, Arizona USA , March 15-20, 1987.

15. Eqqebrecht Lewis C, Interface to the IBM Personal Computer, Howard W.Sams & Son CO., Inc., Indianapolis, 1983.

16.Intel, The MCS 80/85 Family User's Manual, Intel Corporation, Santa Clara, California, 1983.

17. Leventhal Lance A., 8080A-8085 Asssembly Language Programming, McGraw-Hill Book Co., Osborne/McGraw-Hill, Berkeley, California, 1986.

18. Biggerstaff Ted J., System Software Tools, Prentice-Hall Int. Inc. , Englewood Cliffs, New Jersey, 1986.

19. Duncan Ray, Advanced DOS, June 1986.

20. Borland, Turbo C:Reference and User's Guide, Borland International Inc., Scotts Valley, California, 1987.

21. Lapeyre J.De., L. Vrijens, M. Boisseau, "An Experiment to Evaluate the CCITT Recommendation(I420) on ISDN Access," Proceeding of ISS'87, pp. 870-874, Phoenix, Arizona USA, March 15-20, 1987.

22. Klunker Jurgen,Dr., "Development and Operational of an ISDN Exchange," German Technology Symposium in Thailand,Bangkok," Thailand, November 9-13, 1987.

23. Blackshaw R.E., "Integrated Service Digital Network," Open System Data Transfer, pp. 1-12, February 1984,

24. Treves S.R., "Preparing for the Integrated Services Digital Net work," Communication International, pp. 32-38, July 1984,

25. Stallings William, "Tutorial :Integrated Services Digital Network (ISDN), IEEE. Computer Society Press, Washington D.C.,1985

26. Waber Kurt W., "Considerations on Customer Access to the ISDN," IEEE. Trans. on Comm., Vol. COM-30, No. 9, pp. 2131-2136, September 1982,

27. Proceeding on ISS'87,Phoenix,Arizona,March 1987.

28. Griffiths John M., "ISDN Network Terminating Equipment," IEEE. Trans. on Communication, Vol.COM-30, No. 9, pp. 2137-2142, September 1982.

29. Willems Mark H.G., "Design of a PC Interface Card for Office Communication Using the ISDN Basic Access," IEEE. Journal on Selected Areas in Communication, Vol. 7, No. 2, pp.285-302 , February 1989.

30. Newell J.A., L.D.Landy, J.A.Miller, "The Role of Customer Premises Equipment in ISDN," AT&T Technical Journal, Vol.65,Issue 1 , Jan.-Feb. 1986.

31. Blaise, Turbo C Tools Functions Support for Turbo C, Blaise Computing Inc., Berkeley, California, 1987

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

ไอซีทำหน้าที่แปลงข้อมูลแบบขนานเป็นแบบอนุกรม

ในการแปลงข้อมูลแบบขนานจากบัสของ IBM PC ไปเป็นข้อมูลแบบอนุกรมที่อัตราเร็ว 64 kbps เพื่อที่จะส่งออกไปในช่องสัญญาณ B ของข่าย ISDN นั้นการเลือกใช้อุปกรณ์ที่เหมาะสมจะสามารถลดขั้นตอนในการจัดการกับข้อมูลที่จะทำการแปลงได้มาก โดยเฉพาะถ้าต้องการให้ข้อมูลที่ทำการส่งไปในช่องสัญญาณ B สามารถตรวจสอบความถูกต้องได้ ในบทนี้จะกล่าวถึงไอซี(IC) เบอร์ SAB82520(HSCC) ซึ่งนำมาใช้ในการแปลงข้อมูลและสามารถตรวจสอบความถูกต้องของข้อมูลที่ได้รับได้

ก.1 ส่วนประกอบของ HSCC

HSCC เป็นอุปกรณ์ที่มีความสามารถสูงในการจัดการกับข้อมูลที่จะส่งออกและข้อมูลที่ได้รับโดยการทำงานต่าง ๆ ของไอซีตัวนี้กับ CPU ที่ควบคุมจะติดต่อกันโดยผ่านทางสัญญาณขัดจังหวะ และสาเหตุการเกิดสัญญาณขัดจังหวะ CPU ก็จะสามารถทราบได้โดยการอ่านค่าในรีจิสเตอร์ (Register) ภายใน HSCC และเช่นเดียวกัน CPU ก็สามารถสั่งงาน HSCC ได้โดยการเขียนคำสั่งลงในรีจิสเตอร์ต่างๆ ภายใน HSCC

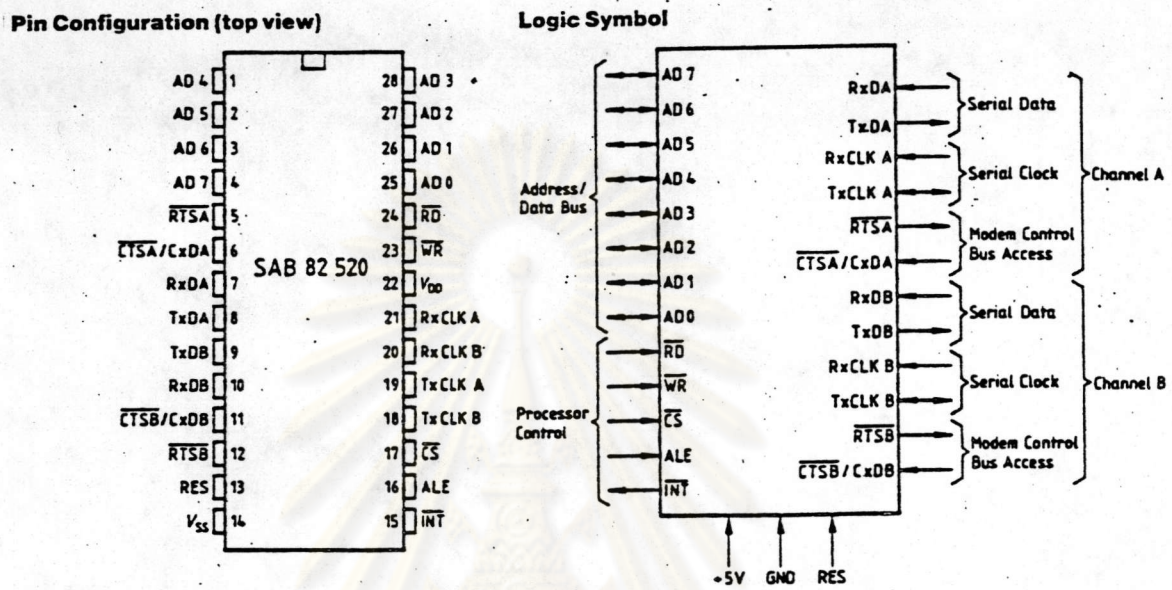
ส่วนประกอบภายในของ HSCC แสดงเป็นแผนภาพกรอบ ดังรูปที่ ก.1

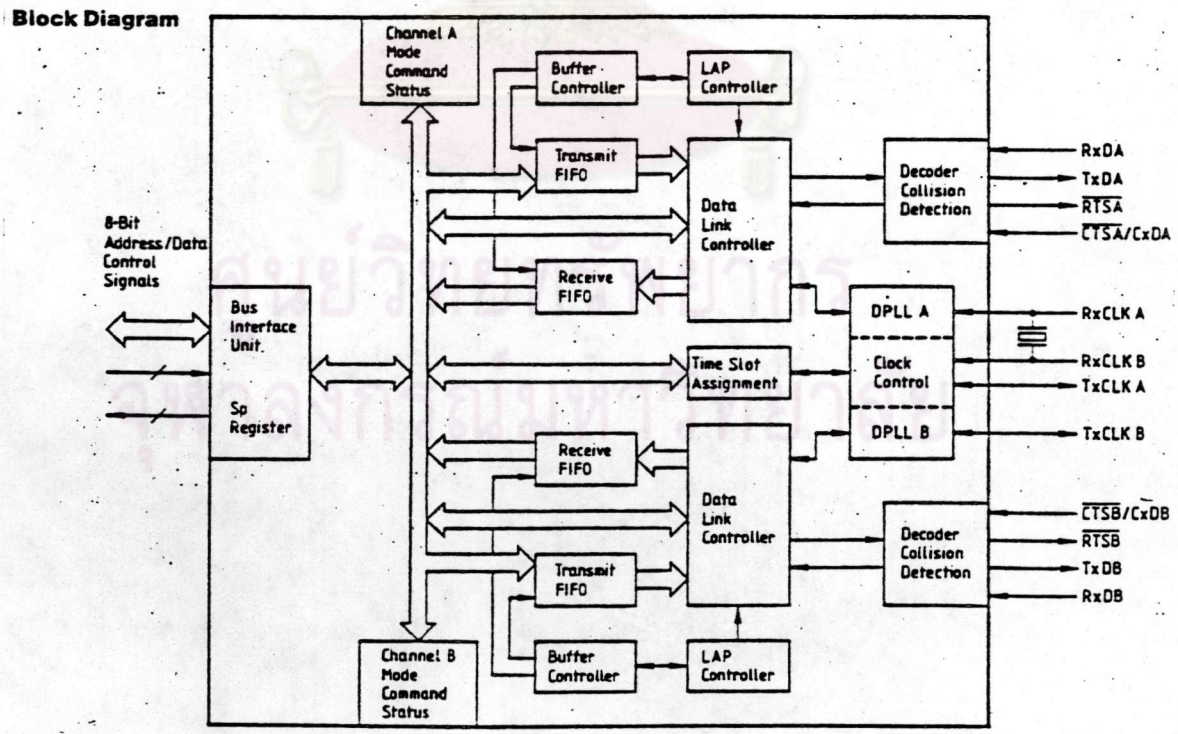
ก.2 การทำงานของ HSCC

ซีพียู (CPU) จะกำหนดโมดและควบคุมการทำงานของ HSCC โดยการเขียนและอ่านค่าในรีจิสเตอร์ภายในของ HSCC การติดต่อระหว่าง CPU กับ HSCC จะกระทำผ่านทางสัญญาณขัดจังหวะในกรณีที่มีเหตุการณ์สำคัญเกิดขึ้นภายใน HSCC โดย CPU จะทราบสาเหตุของการเกิดสัญญาณขัดจังหวะได้โดยการอ่านค่าใน Interrupt Status Register(ISTA) และ Extended Interrupt Register(EXIR) และสำหรับเหตุการณ์ที่เกิดขึ้นภายใน HSCC แต่ไม่สร้างสัญญาณขัดจังหวะสามารถอ่านได้จาก Status Register(STAR) โดย CPU สามารถสั่งงานและตอบรับการได้รับสัญญาณขัดจังหวะจาก HSCC โดยการเขียนคำสั่งใน Command Register (CMDR)

ก.3 การกำหนดโมดการทำงาน

การทำงานของ HSCC จะสามารถกำหนดได้เป็น 2 ส่วน ส่วนแรกคือกำหนดลักษณะการใช้งานสัญญาณนาฬิกาและรูปแบบ (Configuration) ของการเชื่อมต่อแบบอนุกรม (Serial Interface) โดยการเขียนข้อมูลลงใน Common Configuration Register(CCR)และส่วน



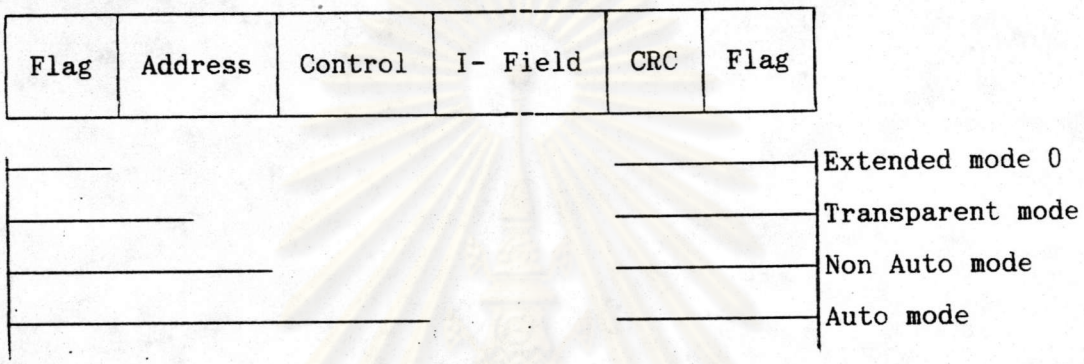


รูปที่ ก.1 แผนภาพกรอบแสดงส่วนประกอบของ HSCC

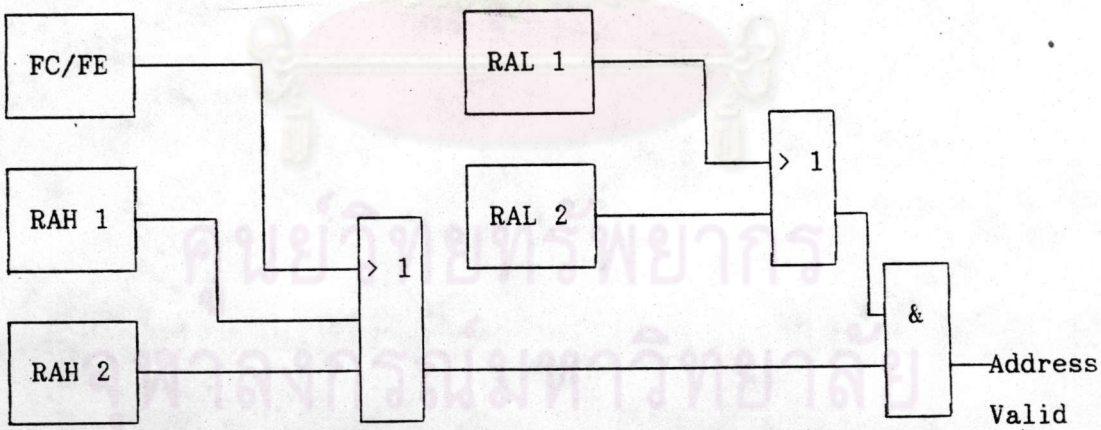
ที่สองคือกำหนดโมดการทำงานของ HDLC Controller ซึ่งทำได้โดยการเขียนข้อมูลใน Mode Register (MODE)

ก.3.1 โมดการทำงานของ HDLC Controller

สามารถกำหนดโมดการทำงานได้ 5 แบบ ซึ่งแต่ละแบบจะมีวิธีการจัดการกับเฟรมข้อมูลแบบ HDLC ต่างกันดังแสดงในรูปที่ ก.2



ก. แสดงการจัดการกับเฟรม HDLC ภายใน HSCC



ข. แสดงการเปรียบเทียบแอดเดรสที่ได้รับ

รูปที่ ก.2 แสดงการจัดการกับเฟรม HDLC ในโมดต่าง ๆ

ก.3.1.1 Auto Mode ในโมดนี้ HSCC จะจัดการกับเฟรมข้อมูลชนิด S และ I โดยอัตโนมัติ โดยมีข้อจำกัดคือสามารถมีเฟรมที่รอการตอบรับจากฝ่ายตรงข้ามได้เพียง 1 เฟรม

เท่านั้น และข้อมูลที่ได้รับมาในส่วนข่าวสาร HSCC จะเก็บไว้ชั่วคราว และจะแจ้งให้ CPU ทราบโดยการส่งสัญญาณขัดจังหวะออกไป ส่วนเฟรมข้อมูลชนิด U และเฟรมที่มีค่า LAP ต่างออกไป HSCC จะส่งไปให้ CPU โดยตรง

การจัดการกับเฟรมข้อมูลที่ได้รับ HSCC จะตรวจสอบค่าแอดเดรสของเฟรมที่ได้รับ โดยการนำเอาแอดเดรสไบต์แรกเปรียบเทียบกับค่า FE_H และ/หรือ FC_H และค่าที่เขียนลงในรีจิสเตอร์ RAH1 และ RAH2 แอดเดรสไบต์ที่สองกับค่าในรีจิสเตอร์ RAL1 และ RAL2 เฟรมที่ HSCC จะรับได้ต้องมีแอดเดรสตรงกับค่าในรีจิสเตอร์ใดรีจิสเตอร์หนึ่งสำหรับแอดเดรสทั้ง 2 ไบต์ และ HSCC ที่ได้กำหนดโมดการทำงานแบบ Auto จะจัดการกับเฟรมข้อมูลโดยอัตโนมัติเฉพาะกับเฟรมที่มีแอดเดรสตรงกับค่าในรีจิสเตอร์ RAH1 และ RAL1 เท่านั้น เฟรมนอกจากนี้ HSCC จะส่งต่อไปให้ CPU จัดการโดยตรง

และข้อมูลบิต 1 ในรีจิสเตอร์ RAH1 HSCC จะใช้สำหรับการตรวจสอบว่าเฟรมที่ได้รับนั้นเป็นเฟรมคำสั่งหรือเฟรมคำตอบ

ก.3.1.2 Non-Auto Mode ในโมดนี้ HSCC จะส่งเฟรมข้อมูลทุกเฟรมที่ได้รับการตรวจสอบแอดเดรสทั้ง 2 ไบต์แล้วว่าถูกต้องไปให้ CPU โดยตรง โดยข้อมูลที่ได้รับในส่วนข่าวสาร HSCC จะเก็บไว้ชั่วคราวในรีจิสเตอร์ภายใน และค่าในส่วนแอดเดรสและควบคุมของเฟรมที่ได้รับ CPU จะสามารถอ่านได้จากรีจิสเตอร์ภายใน HSCC

ก.3.1.3 Transparent Mode ในโมดนี้ HSCC จะตรวจสอบเฉพาะแอดเดรสไบต์แรกเท่านั้น ส่วนแอดเดรสไบต์ที่สอง CPU จะสามารถอ่านได้จากรีจิสเตอร์ RAL1 ข้อมูลส่วนควบคุมจากรีจิสเตอร์ RHCR และข้อมูลอื่นๆ ที่เกี่ยวข้องกับเฟรมที่ได้รับจากรีจิสเตอร์ RSTA ในขณะที่ข้อมูลในส่วนข่าวสารจะเก็บไว้ชั่วคราวในรีจิสเตอร์ RFIFO

ก.3.1.4 Extended Transparent Mode 0 ในโมดนี้ HSCC จะเก็บเฟรมข้อมูลที่ได้รับมาทั้งหมด ยกเว้นแฟล็กและส่วนตรวจสอบความถูกต้อง (CRC) ไว้ในรีจิสเตอร์ RFIFO และนอกจากนั้นข้อมูลไบต์แรกถัดจากแฟล็กจะเก็บไว้ในรีจิสเตอร์ RAL1 ไบต์ที่สองในรีจิสเตอร์ RHCR และข้อมูลที่เกี่ยวข้องอื่นๆ ในรีจิสเตอร์ RSTA

ก.3.1.5 Extended Transparent Mode 1 การกำหนดให้ HSCC ทำงานในโมดนี้จะแยกเป็น 2 กรณีคือ กรณีแรกให้วงจรควบคุมภายใน HSCC ที่ทำหน้าที่รับเฟรม HDLC ไม่ทำงาน (Deactivate) กรณีนี้เฟรมข้อมูลที่ HSCC ได้รับ CPU จะสามารถอ่านได้จากรีจิสเตอร์ RAL1 ทุกๆ 8 คาบสัญญาณนาฬิกา กรณีที่สองให้วงจรรับเฟรม HDLC ทำงานในกรณีนี้

ข้อมูลไบต์แรกของเฟรมที่ได้รับ HSCC จะนำไปเปรียบเทียบกับค่า FE_H, FC_H และค่าที่เขียนไว้ในรีจิสเตอร์ RAH1 และ RAH2 ในกรณีที่ข้อมูลที่ได้รับนั้นตรงกับค่าใดค่าหนึ่ง ข้อมูลที่เหลือทั้งหมด HSCC จะเก็บไว้ในรีจิสเตอร์ RFIFO

ก.4 การกำหนดโมดการทำงานของวงจรสร้างสัญญาณนาฬิกา

ภายใน HSCC จะมีวงจร Oscillator (OSC) วงจร Baud Rate Generator (BRG) และวงจร Digital Phase Lock Loop (DPLL) ทำให้ HSCC สามารถสร้างสัญญาณนาฬิกาได้จากภายในหรือรับสัญญาณนาฬิกาจากภายนอก การเลือกสัญญาณนาฬิกาสำหรับการทำงานของ HSCC นั้น ทำได้โดยการเขียนข้อมูลลงในรีจิสเตอร์ CCR Timing Control Register (TCR) และ Baud Rate Generator Register (BGR) โดยสามารถเลือกโมดการทำงานของวงจรสร้างสัญญาณนาฬิกาได้ 7 โมด ดังแสดงในตารางที่ ก.1 และแสดงที่มาของสัญญาณนาฬิการได้ในรูปที่ ก.3

ตารางที่ ก.1 Clock Modes and Clock Sources

Clock	TCR		Common Clock Sources		Channel A Clock Sources		Channel B Clock Sources		Common Sources		Clock Output
Mode	TSS	TIO	BRG	DPLLA/B	REC	TRM	REC	TRM	R STROBE	X STROBE	TxCLKA/B
0	0	0	–	–	RxCLKA	TxCLKA	RxCLKB	TxCLKB	–	–	–
1	0	0	–	–	RxCLKA	RxCLKA	RxCLKB	RxCLKB	TxCLKA	TxCLKB	–
2	0	0	RxCLKA	BRG	DPLLA	TxCLKA	DPLLB	TxCLKB	–	–	–
2	1	1	RxCLKA	BRG	DPLLA	BRG:16	DPLLB	BRG:16	–	–	BRG:16
3	0	1	RxCLKA	BRG	DPLLA	DPLLA	DPLLB	DPLLB	–	–	DPLLA/B
4	0	1	–	–	OSC	OSC	OSC	OSC	–	–	OSC
5	0	0	–	–	RxCLKA	RxCLKA	RxCLKB	RxCLKB	TSAR	TSAX	1)
6	0	0	OSC	BRG	DPLLA	TxCLKA	DPLLB	TxCLKB	–	–	–
6	1	1	OSC	BRG	DPLLA	BRG:16	DPLLB	BRG:16	–	–	BRG:16
7	0	1	OSC	BRG	DPLLA	DPLLA	DPLLB	DPLLB	–	–	DPLLA/B

1) TxCLKA is used for synchronization, TxCLKB supplies a tristate control signal (cf. 4.1.2.6)

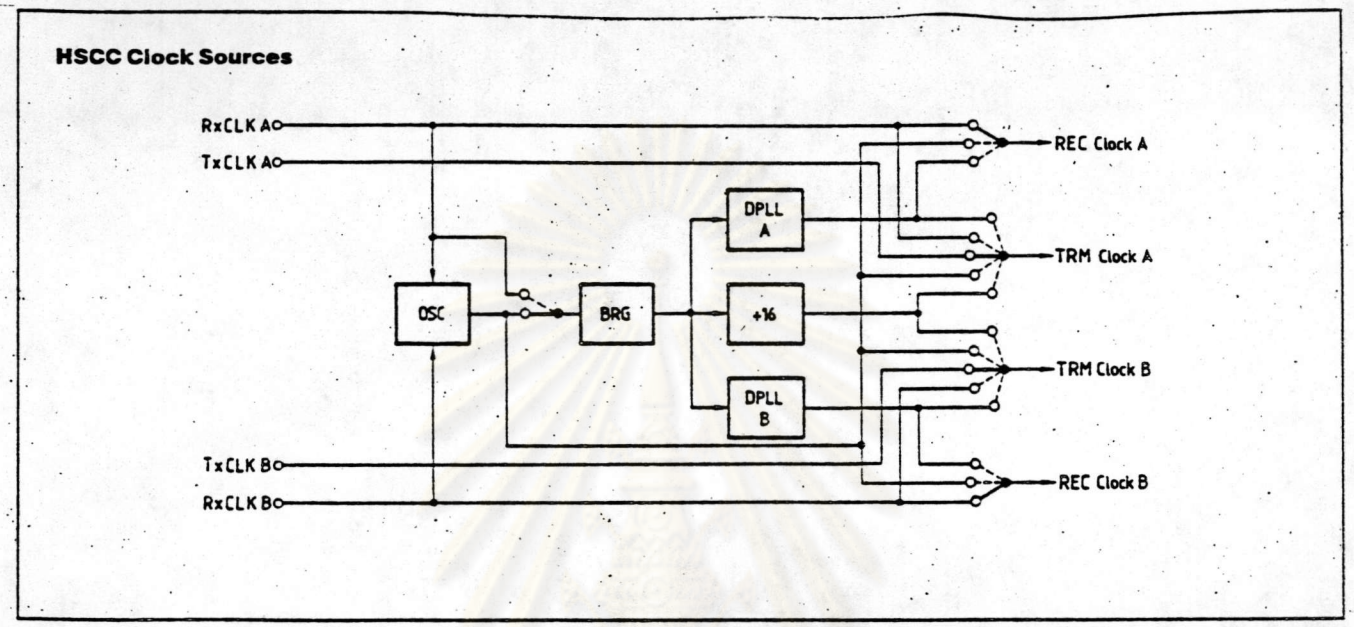
สำหรับการทำงานในโมด 1 นั้น สามารถนำไปใช้ในกรณีของการส่งข้อมูลที่อาศัยวิธีการมัลติเพล็กซ์เชิงเวลา (Time Division Multiplex) ได้ และโมด 5 สามารถนำไปใช้ในระบบที่มีการส่งข้อมูลแบบกำหนดช่องเวลา (Time Slot) ในระบบ PCM ได้

ก.5. การกำหนดรูปแบบการใช้งานการรับส่งข้อมูลแบบอนุกรม

การรับส่งข้อมูลแบบอนุกรม HSCC จะรับส่งผ่านทางขา TxDA/TxDB และ RxDA/RxDB ซึ่งอาจจะเลือกใช้ควบคู่กับสัญญาณควบคุมอื่นๆ คือ RTS และ CTS ได้ รวมทั้งยังสามารถ

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

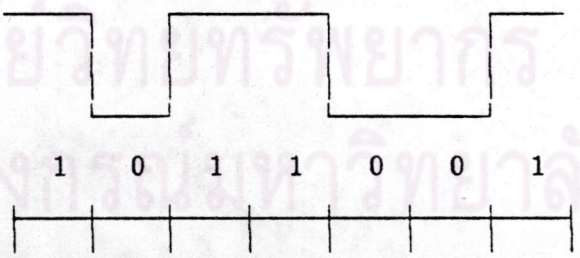
เลือกข้อมูลที่จะให้ HSCC ส่งในระหว่างที่อยู่ในสถานะว่าง (Idle State) ได้ โดยการเขียนข้อมูลลงในรีจิสเตอร์ CCR และการกำหนดสามารถเลือกได้ 4 แบบ ดังนี้



รูปที่ ก.3 ที่มาของสัญญาณนาฬิกาของ HSCC

ก.5.1 แบบจุดต่อจุดข้อมูลเข้ารหัสแบบ NRZ (Point-to-Point NRZ Encoding)

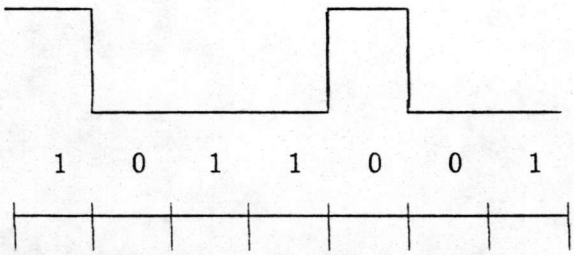
การกำหนดแบบนี้ HSCC จะรับส่งข้อมูลที่อยู่ในรูปแบบ NRZ ดังแสดงในรูปที่ ก.4



รูปที่ ก.4 การเข้ารหัสแบบ NRZ

ก.5.2 แบบจุดต่อจุดข้อมูลเข้ารหัสแบบ NRZI(NRZI Encoding)

การกำหนดแบบนี้ HSCC จะรับส่งข้อมูลที่อยู่ในรูปแบบ NRZI ดังแสดงในรูปที่ ก.5



รูปที่ ก.5 การเข้ารหัสแบบ NRZI

ก.5.3 แบบบัสแบบที่ 1 (Bus Configuration, Timing Mode 1)

การกำหนดแบบนี้ข้อมูลที่รับส่งจะอยู่ในรูปแบบ NRZ โดยมีขาสัญญาณ CxD ในการตรวจสอบการชนกันของข้อมูลที่ส่งออก โดยจะทำการตรวจสอบหลังจากที่ได้ส่งข้อมูลออกไปแล้วเป็นเวลาครึ่งหนึ่งของ 1 คาบสัญญาณนาฬิกา

ก.5.4 แบบบัสแบบที่ 2 (Bus Configuration, Timing Mode 2)

การทำงานจะเหมือนกับแบบที่ ก.5.3 ยกเว้นช่วงเวลาการตรวจสอบการชนกันของข้อมูลที่ส่งออกไป จะเป็น 1 คาบสัญญาณนาฬิกา

ก.6 การรับส่งข้อมูลระหว่าง CPU กับ HSCC

ก.6.1 CPU รับข้อมูลจาก HSCC

ข้อมูลที่ HSCC ได้รับมาจะถูกเก็บไว้ในรีจิสเตอร์ RFIFO ซึ่งมีขนาด 64 ไบต์ เมื่อ HSCC ได้รับข้อมูลครบ 32 ไบต์ หรือข้อมูลส่วนสุดท้ายของเฟรมหรือเฟรมที่มีความยาวน้อยกว่า 32 ไบต์ จะส่งสัญญาณขัดจังหวะออกไปเพื่อให้ CPU มาทำการอ่านข้อมูลออกไปก่อนที่ HSCC จะทำการรับข้อมูลใหม่หรือก่อนที่จะมีข้อมูลใหม่ส่งมาถึง และเมื่อ CPU อ่านข้อมูลเรียบร้อยแล้วจะต้องแจ้งให้ HSCC ทราบโดยการเขียนคำสั่งลงในรีจิสเตอร์ CMDR และในกรณีที่ข้อมูลที่ HSCC ได้รับเป็นส่วนสุดท้ายของเฟรมที่ส่งมา หรือเป็นข้อมูลของเฟรมที่ขนาดสั้นกว่า 32 ไบต์ HSCC จะจัดการส่งข้อมูลที่เกี่ยวข้องกับข้อมูลที่ได้รับนั้นไปในรีจิสเตอร์ RSTA RFBC และ RHCR ให้ CPU ด้วย ดังนี้

ก. ผลการตรวจสอบแอดเดรส
ข. ข้อมูลในส่วนควบคุมของเฟรมที่ได้รับ
ค. ประเภทของเฟรมที่ได้รับ (เฟรมคำสั่งหรือเฟรมคำตอบ)
ง. ผลการตรวจสอบความถูกต้องของข้อมูล
จ. มีข้อมูลเก็บไว้ในรีจิสเตอร์ RFIFO หรือไม่
ฉ. เฟรมที่ได้รับมีการสั่งยกเลิก (Abort) หรือไม่
ช. ระหว่างที่รับเฟรมนี้เกิดเหตุการณ์ Frame Overflow หรือไม่
ซ. ความยาวของข้อมูลที่ได้รับในรีจิสเตอร์ RFIFO

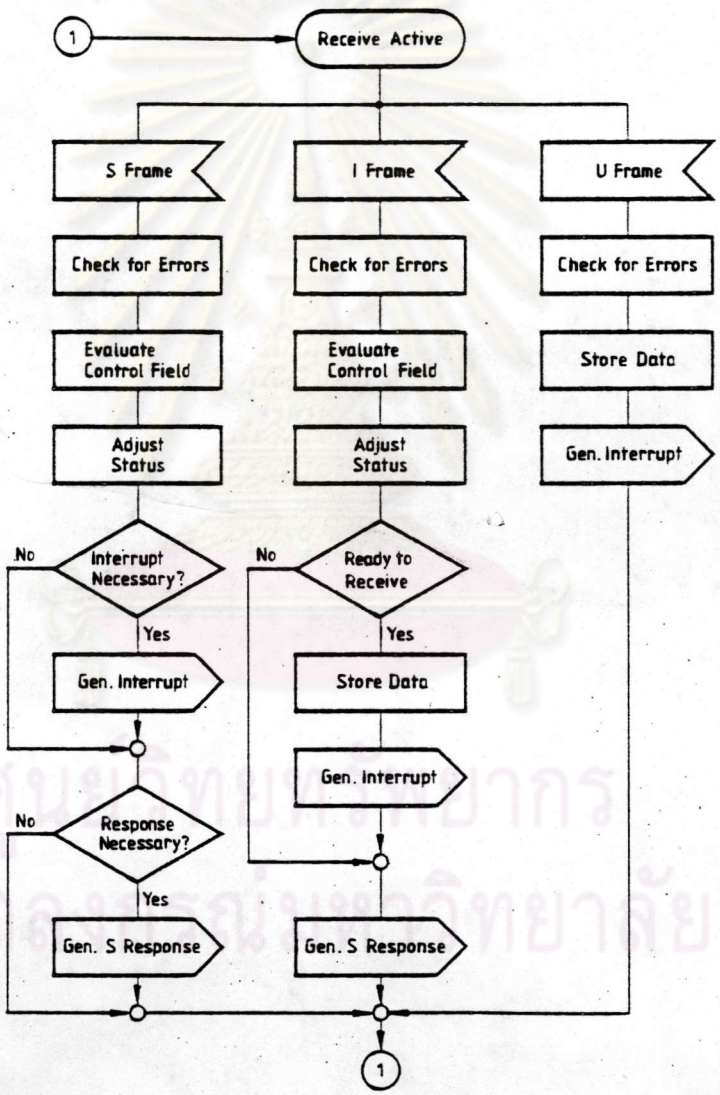
ก.6.2 CPU ส่งข้อมูลให้ HSCC เมื่อ CPU ต้องการให้ HSCC ส่งข้อมูลออกไป CPU ต้องเขียนข้อมูลที่ต้องการส่งลงในรีจิสเตอร์ XFIFO ซึ่งมีขนาด 2X32 ไบต์ โดยเมื่อเขียนครบ 32 ไบต์ หรือเมื่อจบข้อมูลที่ต้องการเขียนที่มีความยาวน้อยกว่า 32 ไบต์ CPU ต้องเขียนคำสั่งลงในรีจิสเตอร์ CMDR เพื่อให้ HSCC จัดการส่งข้อมูลนั้นออกไป ในการสั่งให้ HSCC ส่งข้อมูลออกไปนั้น จะส่งได้ 2 วิธีคือ วิธีแรกให้ HSCC ส่งข้อมูลในแบบ I-Frame ซึ่งในกรณีนี้ส่วนแอดเดรสและควบคุม HSCC จะจัดการให้โดยอัตโนมัติ วิธีที่สองคือ ส่งแบบ Transparent frame (T-frame) วิธีนี้ CPU จะต้องเขียนข้อมูลส่วนแอดเดรสและควบคุมลงในรีจิสเตอร์ XFIFO ด้วยและถ้าหลังจากที่ CPU เขียนข้อมูลให้ HSCC เรียบร้อยและเขียนคำสั่งลงในรีจิสเตอร์ CMDR โดยระบุว่าข้อมูลที่เขียนลงในรีจิสเตอร์ XFIFO นั้นเป็นการจบข้อมูลที่ต้องการส่งสำหรับเฟรมนั้นแล้ว HSCC จะเติมส่วนตรวจสอบความถูกต้องและแฟล็กปิดให้โดยอัตโนมัติ ส่วนในกรณีที่ยังไม่หมดเฟรม HSCC จะส่งสัญญาณขัดจังหวะเพื่อขอข้อมูลส่วนต่อไปจาก CPU

ก.7 การทำงานในระดับโปรโตคอลชั้นที่ 2

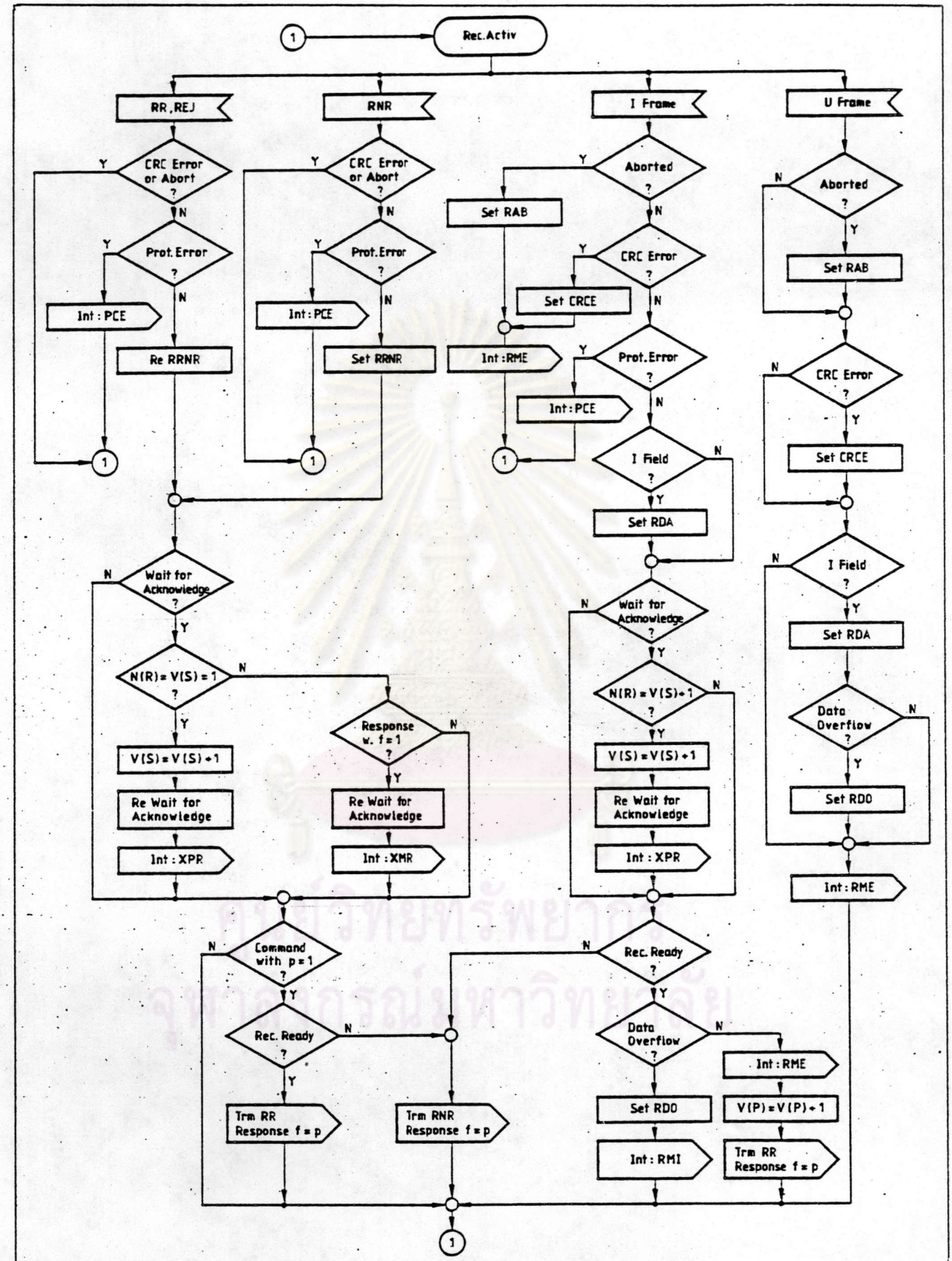
นอกเหนือจากการตรวจสอบแอดเดรสของเฟรมที่ได้รับแล้ว HSCC ยังสามารถจัดการกับเฟรมข้อมูลชนิด S และ I ได้โดยอัตโนมัติโดยไม่จำเป็นต้องแจ้งให้ CPU ทราบในกรณีที่กำหนดโมดการทำงานแบบ Auto ส่วนเฟรมชนิด U HSCC จะส่งให้ CPU โดยตรง

ก.7.1 การรับเฟรมข้อมูล กรณีที่กำหนดโมดการทำงานแบบ Auto HSCC จะจัดการกับเฟรมชนิด S โดยไม่แจ้งให้ CPU ทราบ ยกเว้นกรณีที่เฟรมที่ได้รับนั้นไม่ถูกต้อง และสำหรับเฟรมชนิด I HSCC จะตรวจสอบความถูกต้องของเฟรมที่ได้รับ และแจ้งให้ CPU ทราบก็ต่อเมื่อเฟรมที่ได้รับนั้นถูกต้องและจะเก็บข้อมูลที่ได้รับมาไว้ในรีจิสเตอร์ RFIFO ส่วนเฟรมชนิด U HSCC จะเก็บไว้ในรีจิสเตอร์ RFIFO และส่งให้ CPU โดยตรงต่อไป ขั้นตอนการรับเฟรมข้อมูลแสดงไว้ในรูปที่ ก.6

ก.7.2 การส่งเฟรมข้อมูล ในกรณีที่กำหนดการทำงานในโมด Auto HSCC จะจัดการส่งเฟรมชนิด S โดยอัตโนมัติ การส่งเฟรมโดย HSCC จะทำได้ 2 วิธีคือส่งแบบ I-Frame และส่งแบบ T-Frame การส่งเฟรมแบบ I-Frame นั้น HSCC จะเป็นตัวที่ทำหน้าที่ในการตรวจสอบว่าเฟรมที่ได้ส่งไปแล้วนั้นได้รับคำตอบหรือไม่ โดยมีขั้นตอนการตรวจสอบแสดงไว้ในรูปที่ ก.7 ซึ่งผลจากการตรวจสอบ HSCC จะแจ้งให้ CPU ทราบ โดยการส่งสัญญาณขัดจังหวะโดยผลการตรวจสอบจะเป็นได้ 3 แบบคือ เฟรมที่ได้ส่งไปแล้วนั้นผู้รับได้รับถูกต้อง หรือต้องส่งเฟรมนั้นใหม่ หรือไม่ได้รับคำตอบจากฝ่ายตรงข้ามภายในเวลาที่กำหนด การส่งเฟรมแบบนี้ต้องกำหนดให้ Timer ทำงานในโมด Internal ส่วนการส่งเฟรมแบบ T-Frame นั้นสามารถทำได้กับการทำงานในทุกโมด และหลังจากที่ได้ส่งเฟรมข้อมูลออกไปแล้ว HSCC จะยอมให้มีการเขียนข้อมูลลงในรีจิสเตอร์ XFIFO ทันที และ HSCC จะแจ้งให้ CPU ทราบโดยการส่งสัญญาณขัดจังหวะ และถ้าต้องการจับ



รูปที่ ก.6 ขั้นตอนการจัดการกับเฟรมที่ได้รับโดย HSCC ในโมด Auto



รูปที่ ก.7 ขบวนการตรวจสอบการได้รับเฟรมข้อมูล

เวลาในการส่งเฟรมแบบนี้ สามารถทำได้โดยกำหนดให้ Timer ทำงานในโมด External

ก.8 การทำงานของวงจร Baud Rate Generator (BRG)

วงจร BRG ภายใน HSCC จะทำหน้าที่ปรับแต่งสัญญาณนาฬิกาความถี่สูงที่ได้จาก OSC (Oscillator) ภายนอกให้อยู่ในระดับที่จะใช้เป็นความถี่ที่ DPLL ต้องการโดยค่าที่ใช้ในการปรับความถี่นั้น สามารถกำหนดได้ โดยการเขียนข้อมูลลงในรีจิสเตอร์ BGR และ TCR

ก.9 การทำงานในโมดทดสอบ (Test Mode)

HSCC สามารถทำงานในโมดทดสอบได้โดยการเขียนข้อมูลในบิต TLP ลงในรีจิสเตอร์ MODE การทำงานในโมดนี้ HSCC จะทำการต่อขารับและส่งข้อมูลเข้าด้วยกัน ซึ่งจะทำให้สามารถทำการทดสอบช่องสัญญาณที่รับส่งเฟรมแบบ HDLC ของ HSCC ได้

ตารางที่ ก.2 แอดเดรสของรีจิสเตอร์ทำหน้าที่ HDLC Controller

AddressH		Read	Write
Channel B	Channel A		
40 - 5F	00 - 1F	RFIFO	XFIFO
30,B1	20,A1	ISTA	MASK
31,B0	21,A0	STAR	CMDR
32,B3	22,A3	MODE	MODE
33,B2	23,A2	TIMR	TIMR
34,B5	24,A5	EXIR	XAD1
35,B4	25,A4	RFBC	XAD2
36,B7	26,A7		RAH1
37,B6	27,A6	RSTA	RAH2
38,B9	28,A9	RAL1	RAL1
39,B8	29,A8	RHCR	RAL2

ก.10 รีจิสเตอร์ต่างๆ ภายใน HSCC

การทำงานต่างๆ ของ HSCC จะสามารถสั่งงานได้จาก CPU โดยการเขียนข้อมูลลงในรีจิสเตอร์ที่เกี่ยวข้อง ซึ่งภายใน HSCC จะประกอบด้วยรีจิสเตอร์ต่าง ๆ ส่วนหนึ่งและอีกส่วนหนึ่ง คือ FIFO ซึ่งแยกเป็น RFIFO และ XFIFO โดยการเขียนอ่าน FIFO สามารถทำได้ที่ตำแหน่งแอดเดรส 00_H-$1F_H$ และ 40-$5F_H$ รีจิสเตอร์ภายในของ HSCC จะสามารถแยกเป็นประเภทใหญ่ๆ ได้ 2 ประเภท โดยมีรายละเอียดของรีจิสเตอร์แต่ละตัว ดังนี้

ก.10.1 รีจิสเตอร์ที่ทำหน้าที่ HDLC Controller

ก.10.1.1 ตำแหน่งของรีจิสเตอร์ต่างๆ ดังแสดงในตารางที่ ก.2

ก.10.1.2 รีจิสเตอร์ควบคุม (Control Register)

ก.10.1.2.1 Interrupt Status Register (ISTA) อ่านแอดเดรส 20/30 มีค่าภายหลังการ Reset คือ 00_H

7 0

RME	RPF	RSC	XPR	TIN	ICA	EXA	EXB

RME - Receive Message End; เป็นการบอกว่ามีข้อมูลที่เป็นส่วนสุดท้ายของเฟรมที่ได้รับหรือเฟรมที่มีขนาดสั้นกว่า 32 ไบต์เก็บใน RFIFO

RPF - Receive Pool Full; HSCC ได้รับข้อมูลจำนวน 32 ไบต์ เก็บอยู่ในรีจิสเตอร์ RFIFO

RSC - Receive Status Change; ในการทำงานในโมด Auto เป็นการบอกว่า ผ่ายตรงข้ามมีสถานะเปลี่ยนไประหว่าง Receive Ready และ Receive not Ready และค่าสถานะปัจจุบัน สามารถอ่านได้จากบิต RRNR ในรีจิสเตอร์ STAR

XPR - Transmit Pool Ready; XFIFO สามารถรับข้อมูลเพิ่มได้อีก

TIN - Timer Interrupt;

ICA - Interrupt of Channel A; มีสัญญาณขัดจังหวะเกิดขึ้นจากช่อง A ของ HSCC

EXA - Extended Interrupt of Channel A; มีเหตุการณ์เกิดขึ้นในช่อง A ของ HSCC สาเหตุการเกิดสัญญาณขัดจังหวะ อ่านได้จาก EXIR

EXB - Extended Interrupt of Channel B;สำหรับช่อง B ของ HSCC

ภายหลังจาก CPU ได้อ่านรีจิสเตอร์ที่เกี่ยวข้องแล้ว ข้อมูลในบิต ICA,EXA และ EXB จะถูกลบทิ้ง และบิตที่เหลือทุกบิตจะถูกลบทิ้งเมื่อ CPU อ่านรีจิสเตอร์ ISTA บิต ICA,EXA และ EXB ใช้งานเฉพาะในรีจิสเตอร์ ISTA ของช่อง B เท่านั้น

ก.10.1.2.2 Mask Register(MASK) เขียนแอดเดรส 20/30 มีค่าภายหลังการ Reset คือ 00_H

รีจิสเตอร์นี้ใช้สำหรับการเลือกที่จะไม่ให้ HSCC สร้างสัญญาณขัดจังหวะ เมื่อเกิดเหตุการณ์ดังกล่าวขึ้น และในกรณีที่เลือก Masked บิต EXA หรือ EXB เมื่อเกิดเหตุการณ์ที่กำหนดในรีจิสเตอร์ EXIR ขึ้น HSCC จะไม่สร้างสัญญาณขัดจังหวะส่งให้ CPU แต่จะมีการ Set บิตนี้ในรีจิสเตอร์ ISTA

ก.10.1.2.3 Extened Interrupt Register (EXIR) อ่านแอดเดรส24/34 ค่าภายหลังการ Reset เป็น 00_H

7							0
XMR	XDU	PCE	RFO	CSC	0	0	0

XMR - Transmit Message Repeat; ในโมด Auto ต้องส่งเฟรมข้อมูลออกไปใหม่ เนื่องจากไม่ได้รับเฟรมตอบรับ หรือผู้รับไม่พร้อมที่จะรับ หรือเกิดการชนกันของข้อมูลหลังจากส่งข้อมูลไบต์ที่ 32 แล้ว

XDU - Transmit Data Underrun; สำหรับแสดงว่าข้อมูลที่ส่งออกไปถูกบิดท้ายด้วยข้อมูล Idle เนื่องจากภายในรีจิสเตอร์ XFIFO ไม่มีข้อมูล

PCE - Protocol error; ใช้ในกรณีการทำงานในโมด Auto เป็นการแสดงว่า HSCC ได้รับเฟรมที่ไม่ถูกต้อง

RFO - Receive Frame Overflow; ใช้แสดงเหตุการณ์ที่มีเฟรมข้อมูลส่งมาที่ HSCC แต่ HSCC ไม่สามารถรับได้เพราะบัฟเฟอร์ภายในไม่ว่าง

CSC - Clear-to-send change; ถ้าบิต CIE ในรีจิสเตอร์ TCR เป็น "1" บิตนี้จะใช้แสดงว่าสัญญาณมีการเปลี่ยนระดับที่ขา CTS ของ HSCC ส่วนค่าจริงจะสามารถอ่านได้จากรีจิสเตอร์ STAR

ก.10.1.2.4 Status Register(STAR) อ่านแอดเดรส 21/31H ค่าภายหลังการ Reset คือ 48H

7							0
XDOV	XFW	XRNR	RRNR	MBR	CEC	CTS	0

XDOV - Transmit Data Overflow; แสดงว่ามีการเขียนข้อมูลลงในรีจิสเตอร์ XFIFO มากกว่า 32 ไบต์

XFW - XFIFO write enable; แสดงว่าสามารถเขียนข้อมูลลงในรีจิสเตอร์ XFIFO ได้

XRNR - Transmit RNR; แสดงสถานะของ HSCC ในโมด Auto

RRNR - Receive RNR ; แสดงสถานะของฝ่ายตรงข้าม

MBR - Message Buffer ready; แสดงว่ารีจิสเตอร์ RFIFO สามารถรับข้อมูลได้อย่างน้อย 1 เฟรม

CTS - Clear to send; ถ้าบิต CIE ในรีจิสเตอร์ TCR เป็น "1" บิตนี้จะใช้แสดงสถานะตรงข้ามของระดับสัญญาณที่ขา CTS

CEC - Command Execution Control; บิตนี้จะเป็น "1" ขณะที่ HSCC กำลังทำงานตามคำสั่งที่เขียนลงในรีจิสเตอร์ CMDR

ก.10.1.2.5 Command Register(CMDR)เขียนแอดเดรส 21/31H ค่าภายหลังการ Reset คือ 00H

7							0
RMC	RHR	RNR	STI	XTF	XIF	XME	XRES

RMC - Receive Message Complete; เป็นคำสั่งที่ใช้ตอบการได้รับสัญญาณขัดจังหวะ เนื่องจากเหตุการณ์ที่แสดงด้วยบิต RPF หรือ RME ทำให้ HSCC สามารถลบข้อมูลในรีจิสเตอร์ RFIFO ทิ้งได้

RHR - Reset HDLC Receiver; สั่งให้ HSCC ทำการ Reset วงจรรับเฟรม HDLC มีผลทำให้ข้อมูลที่อยู่ในบัฟเฟอร์ของ HSCC ถูกลบทิ้ง และถ้าทำงานในโมด Auto ค่าตัวแปรที่นับจำนวนการรับและส่งเฟรมจะถูก

Reset ด้วย

RNR - Receive not ready; กำหนดสถานะของ HSCC เป็น Receive not ready ("1") หรือ Receive ready ("0") ในโมด Auto

STI - Start timer; ให้ timer ภายในเริ่มทำงาน

XTF - Transmit transparent frame; สั่งให้ HSCC ส่งข้อมูลออกไปแบบ T-frame

XIF - Transmit I frame; สั่งให้ HSCC ส่งข้อมูลออกไปแบบ I-frame

XME - Transmit message end; บอกให้ HSCC ทราบว่าข้อมูลที่อยู่ในขณะนี้เป็นข้อมูลส่วนสุดท้ายของเฟรมที่จะทำการส่ง

XRES - Transmitter reset; กำหนดให้ HSCC ทำการ Reset วงจรส่งเฟรม HDLC ในกรณีนี้ข้อมูลในรีจิสเตอร์ XFIFO จะถูกลบทิ้ง และ HSCC จะส่งข้อมูล Idle ออกไป

ในการเขียนคำสั่งลงในรีจิสเตอร์ CMDR นั้น HSCC จะใช้เวลาประมาณ 2.5 คาบสัญญาณนาฬิกาในการทำงานตามคำสั่ง และในขณะนี้บิต CEC จะเป็น "1" และจะไม่สามารถเขียนข้อมูลลงในรีจิสเตอร์ CMDR ได้

ก.10.1.2.6 Mode Register(MODE) อ่านเขียนแอดเดรส $22/32_H$ ค่าภายหลังการ Reset เป็น 00_H

7							0
MDS1	MDS0	ADM	TMD	RAC	RTS	TRS	TLP

MDS1-MDS0 - Mode select; กำหนดโมดการทำงานของ HDLC Controller

00 - Auto mode

01 - Non-auto mode

10 - Transparent mode

11 - Extended transparent mode

ADM - Address mode; สำหรับเลือกจำนวนไบต์ที่ใช้เป็นส่วนของแอดเดรส โดยเป็นแบบ 1 ไบต์ ("0") หรือแบบ 2 ไบต์ ("1") และถ้ากำหนดการทำงานในโมด Extended transparent จะใช้แยกระหว่าง Extended transparent mode 0 และ mode 1

RAC - Receive active; กำหนดให้วงจรรับเฟรม HDLC ทำงาน ("1") หรือไม่ทำงาน ("0")

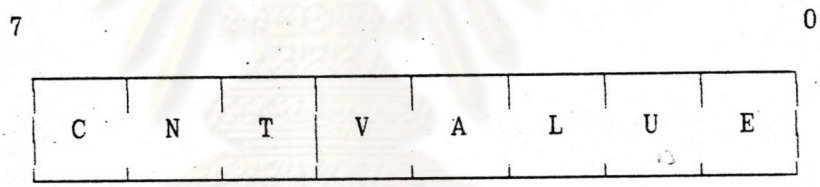
TMD - Timer mode;กำหนดการทำงานของ Software Timer ภายในของ HSCC ระหว่างโมด Internal ("1") และ External ("0")

RTS - Request to send; กำหนดการสั่งงานขา RTS ระหว่างให้ขา RTS อยู่ในสถานะ Active ตลอด("1") หรือให้ควบคุมโดย HSCC ("0")

TRS - Timer Resolution select; ค่าความละเอียดของหน่วยเวลาที่ใช้ใน Software timer (factor k) ระหว่าง 2^{15}("0") และ 2^{9} ("1") คาบสัญญาณนาฬิกา

TLP - Test loop; สำหรับการทดสอบตัวเองของช่องสัญญาณแบบ HDLC ภายใน HSCC

ก.10.1.2.7 Timer Register(TIMR) อ่านเขียนแอดเดรส 23/33H



VALUE - ใช้สำหรับกำหนดช่วงเวลา t_1 โดยจะคำนวณได้จาก

$$t_1 = k^* (value+1)*TCP$$

โดยที่ k สามารถเลือกได้ระหว่าง 2^{15} กับ 2^{9}

TCP คือ คาบสัญญาณนาฬิกาที่ใช้ส่งข้อมูล

CNT - Count; การใช้งานค่านี้จะขึ้นอยู่กับการกำหนดการทำงานของ Software timer ภายใน HSCC

สำหรับ Internal Timer mode;จะระบุจำนวนเฟรมชนิด S ทั้งหมดที่สามารถส่งออกไปได้ในกรณีที่เฟรมชนิด I ที่ได้ส่งออกไปแล้วไม่ได้รับคำตอบ โดยการส่ง HSCC จะจัดการโดยอัตโนมัติเมื่อหมดเวลา t_1 และถ้ากำหนดให้ CNT = 7 เป็นการบอกว่า HSCC สามารถส่ง S เฟรมใดๆสั่งชนิด S ได้ไม่จำกัด

สำหรับ External Timer mode; ค่าใน CNT บวกกับ VALUE จะบอกคาบเวลา t_2 ที่ HSCC จะส่งสัญญาณขัดจังหวะออกมาเมื่อหมดเวลาการนับของ Software Timer โดยคำนวณได้จาก

$$t_2 = 32 * k * CNT * TCP + t_1$$

ถ้า CNT = 7, สัญญาณขัดจังหวะนี้จะถูกส่งออกมาทุกครั้งที่หมดเวลา t_1

การหยุดการทำงานของ Timer นี้จะทำได้ตลอดเวลา โดยการเขียนข้อมูลลงในรีจิสเตอร์ TIMR

ก.10.1.2.8 Receive Frame Byte Counter(RFBC) อ่านแอดเดรส $25/35_H$ ค่าภายหลังการ Reset คือ 00_H

7

RDC7							RDC0

RDC7-RDC0 - Receive data count;บอกจำนวนไบต์ข้อมูลของเฟรมที่ได้รับ โดย RDC4-RDC0 จะบอกจำนวนไบต์ข้อมูลของข้อมูลที่ได้รับอยู่ในรีจิสเตอร์ RFIFO และถ้าเฟรมข้อมูลที่ได้รับยาวมากกว่า 223 ไบต์ บิต RDC7-RDC5 ก็จะยังคงมีค่าเป็น 7 ตลอดเวลา

ก.10.1.2.9 Receive Status Register(RSTA) อ่านแอดเดรส $27/37_H$

7 0

RDA	RDO	CRC	RAB	HA1	HA0	C/R	LA

RDA - Data Received; บอกว่ามีข้อมูลเก็บอยู่ในรีจิสเตอร์ RFIFO

RDO - Receive Data Overflow; เกิดเหตุการณ์ Data Overflow เนื่องจากเฟรมที่ HSCC กำลังรับอยู่นี้

CRC - CRC Compare; ผลการตรวจสอบค่า CRC ถูกต้อง

RAB - Receive Message Aborted; ข้อมูลที่ได้รับถูกยกเลิกโดยผู้ส่ง

HA1,HA0 - High Byte Address Compare; ผลการตรวจสอบแอดเดรสไบต์แรกของเฟรมที่ได้รับ

10 - RAH1

00 - RAH2

01 - FE_H หรือ FC_H

C/R - Command-Response; ค่าของบิต C/R ของเฟรมที่ได้รับ

LA - Low Byte Address Compare; ผลการตรวจสอบแอดเดรสไบต์ที่สองของเฟรมที่ได้รับ

1 - RAL1

0 - RAL2

ก.10.1.3 <u>รีจิสเตอร์เก็บข้อมูล</u> (Data Register)

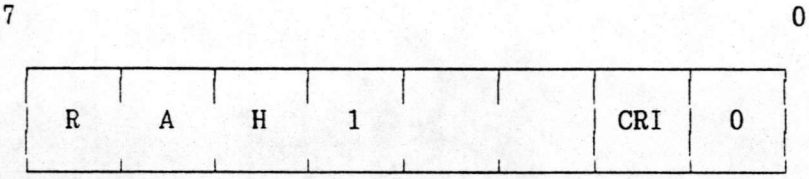
ก.10.1.3.1 Receive FIFO (RFIFO) อ่านแอดเดรส $0/40_H$ ข้อมูลที่ HSCC ได้รับสามารถอ่านได้จากรีจิสเตอร์ RFIFO หลังจาก CPU ได้รับสัญญาณขัดจังหวะ RME หรือ RPF

ก.10.1.3.2 Transmit FIFO(XFIFO) เขียนแอดเดรส $0/40_H$ CPU สามารถส่งข้อมูลให้ HSCC ส่งออกไปได้โดยเขียนลงในรีจิสเตอร์นี้ หลังจากได้รับสัญญาณขัดจังหวะ XPR

ก.10.1.3.3 Transmit Address 1 (XAD1) เขียนแอดเดรส $24/34_H$ ในการทำงานโมด Auto แอดเดรสไบต์แรกที่ HSCC จะนำไปใช้ต้องเขียนลงที่รีจิสเตอร์นี้

ก.10.1.3.4 Transmit Address 2 (XAD2) เขียนแอดเดรส $25/35_H$ ในการทำงานโมด Auto แอดเดรสไบต์ที่สองของเฟรม HDLC HSCC จะอ่านจากรีจิสเตอร์นี้

ก.10.1.3.5 Receive Address Byte High Register(RAH1) เขียนแอดเดรส $26/36_H$

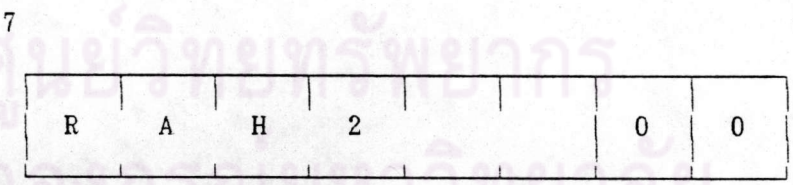


RAH1 - ค่าของแอดเดรสไบต์แรกของเฟรม HDLC ค่าที่หนึ่ง

CRI - Command-response bit interpretation; ในการทำงานโมด Auto ใช้ในการแปลความหมายของบิต C/R ของเฟรมข้อมูลที่ได้รับดังแสดง

	CRI = 1 C/R Value	CRI = 0 C/R Value
Command receive	0	1
Response receive	1	0
Command transmit	1	0
Response transmit	0	1

ก.10.1.3.6 Receive Address Byte High Register(RAH2) เขียนแอดเดรส $27/37_H$



RAH2 - ค่าของแอดเดรสไบต์แรกค่าที่สอง

ก.10.1.3.7 Receive Address Byte Low Register(RAL1) อ่านเขียนแอดเดรส $28/38_H$ คือ ค่าของแอดเดรสไบต์ที่สอง ค่าแรกที่ HSCC จะใช้ในการรับเฟรม HDLC โดยรีจิสเตอร์นี้จะเขียนได้ในโมดการทำงานแบบ Auto และ Non-auto และสามารถอ่านได้ในโมดอื่น

ก.10.1.3.8 Receive Address Byte Low Register(RAL2) เขียนแอดเดรส 29/39H คือค่าของแอดเดรสไบต์ที่สอง ค่าที่สองที่ HSCC จะใช้ในการรับเฟรม HDLC ในการทำงานในโมด Auto และ Non-auto

ก.10.1.3.9 Receive HDLC Control Register(RHCR) อ่านแอดเดรส 29/39H เก็บค่าในส่วนควบคุมของเฟรม HDLC ที่ HSCC ได้รับ

ก.10.2 รีจิสเตอร์ร่วม (Common Register)

ก.10.2.1 ตำแหน่งของรีจิสเตอร์ต่างๆ แสดงในตารางที่ ก.3

ตารางที่ ก.3 แสดงแอดเดรสของรีจิสเตอร์ร่วม

AddressH	Read	Write
2B,AA		BGR
2C,AD	TCR	TCR
2D,AC		TSAR
2E,AB		TSAX
2F,AE	CCR	CCR

ก.10.2.2 ความหมายของแต่ละรีจิสเตอร์

ก.10.2.2.1 Common Configuration Register (CCR) อ่านแอดเดรส 2FH ค่าภายหลังการ Reset คือ 00H

7							
PU	SC1	SC0	ODS	LSS	CM2	CM1	CM0

PU - Power up; ใช้สำหรับสั่งงาน HSCC ระหว่างโมด Power-up และ Power-down

SC1,SC0 - Serial part configuration

00 - จุดต่อจุด เข้ารหัสแบบ NRZ

10 - จุดต่อจุด เข้ารหัสแบบ NRZI

01 - แบบบัสแบบที่ 1 เข้ารหัสแบบ NRZ

11 - แบบบัสแบบที่ 2 เข้ารหัสแบบ NRZ

ODS - Output driver select;กำหนดลักษณะการต่อของขา TxD ระหว่างแบบ push-pull ("1") หรือ Open drain ("0")

LSS - Line sync select; เลือกสัญญาณที่จะให้ HSCC ส่งออกไป เมื่ออยู่ในสถานะว่างโดยจะส่งแฟล็ก ("1") หรือสัญญาณ Idle ("0")

CM2-0 - Clocking mode; เลือกการทำงานของวงจรสร้างสัญญาณนาฬิกา

ก.10.2.2.2 Time Slot Assignment Register Receive (TSAR) เขียนแอดเดรส $2D_H$

7

	T	S	N	R		TSS	TCS0

TSNR - Time-slot number receive;

TSS - Time-slot select; เลือกจำนวนช่องเวลาใน 1 เฟรมระหว่าง 32 ช่อง ("1") หรือ 64 ช่อง ("0")

TCS0 - Clock shift 0;ใช้ในการกำหนดจำนวนคาบสัญญาณนาฬิกาที่จะใช้ในการส่งข้อมูลออก บิต 0

ก.10.2.2.3 Time-Slot Assignment Register Transmit (TSAX) เขียนแอดเดรส $2E_H$

7

	T	S	N	X		TCS2	TCS1

TSNX - Time-slot number transmit

TCS2,TCS1 - Clock shift; ใช้ในการกำหนดจำนวนคาบสัญญาณนาฬิกาที่จะใช้ในการส่งข้อมูลออก บิต 1 และ 2

ก.10.2.2.4 Timing Control Register (TCR) อ่านเขียนแอดเดรส $2C_H$ ค่าภายหลังการ Reset คือ 00_H ผลที่เกิดจากการเขียนข้อมูลลงในรีจิสเตอร์นี้จะขึ้นอยู่กับโมดของสัญญาณนาฬิกาที่เลือกใช้ ดังนี้

Clock mode 5

7							0
RCS2	RCS1	RCS0	CCS	TIO	CIE	0	0

Clock mode 2,6

7							0
BR9	BR8	BDF	TSS	TIO	CIE	0	0

Clock mode 3,4,7

7							0
BR9	BR8	BDF	0	TIO	CIE	0	0

Clock mode 0,1

7							0
0	0	0	0	0	CIE	0	0

RCS2-0 - Receive clock shift;

CCS - Channel capacity select; เลือกอัตราเร็วที่ใช้ส่งข้อมูลระหว่าง 56 kbit ("1") หรือ 64 kbit ("0")

BR9-8 - Band rate บิตที่ 9 และ 8

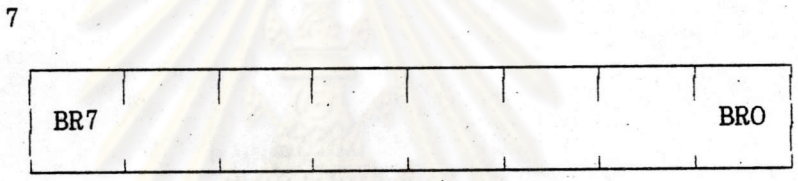
BDF - Baud rate division factor; ใช้ในการกำหนดตัวหารสำหรับ Baud Rate Generator ระหว่าง 1 ("0") หรือได้จากค่า BR9-0 ("1")

TSS - Transmit clock source select; เลือกต้นกำเนิดของสัญญาณนาฬิกาที่ใช้ในการส่งข้อมูลระหว่าง TxCLK ("0") หรือสัญญาณที่ได้จาก BRG หารด้วย 16 ("1")

TIO - Transmit clock input/output; กำหนดขา TxCLK เป็นขาสัญญาณเข้า ("0") หรือสัญญาณออก ("1")

CIE - Clear to send interrupt enable; ให้ HSCC สร้างสัญญาณขัดจังหวะส่งให้ CPU ("1") หรือไม่ ("0") ในกรณีที่ขา CTS active

ก.10.2.2.5 Baud Rate Generator Register (BGR) เขียนแอดเดรส $2B_H$



BR7-0 - Baud rate บิต 7-0; ใช้ร่วมกับบิต BR8-9 ในรีจิสเตอร์ TCR ในการคำนวณหาตัวหารสำหรับ BRG โดยค่า BR9-0 จะใช้ในการคำนวณ ดังนี้

$k = 2*(N+1)$

N ได้จาก BR9-0

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

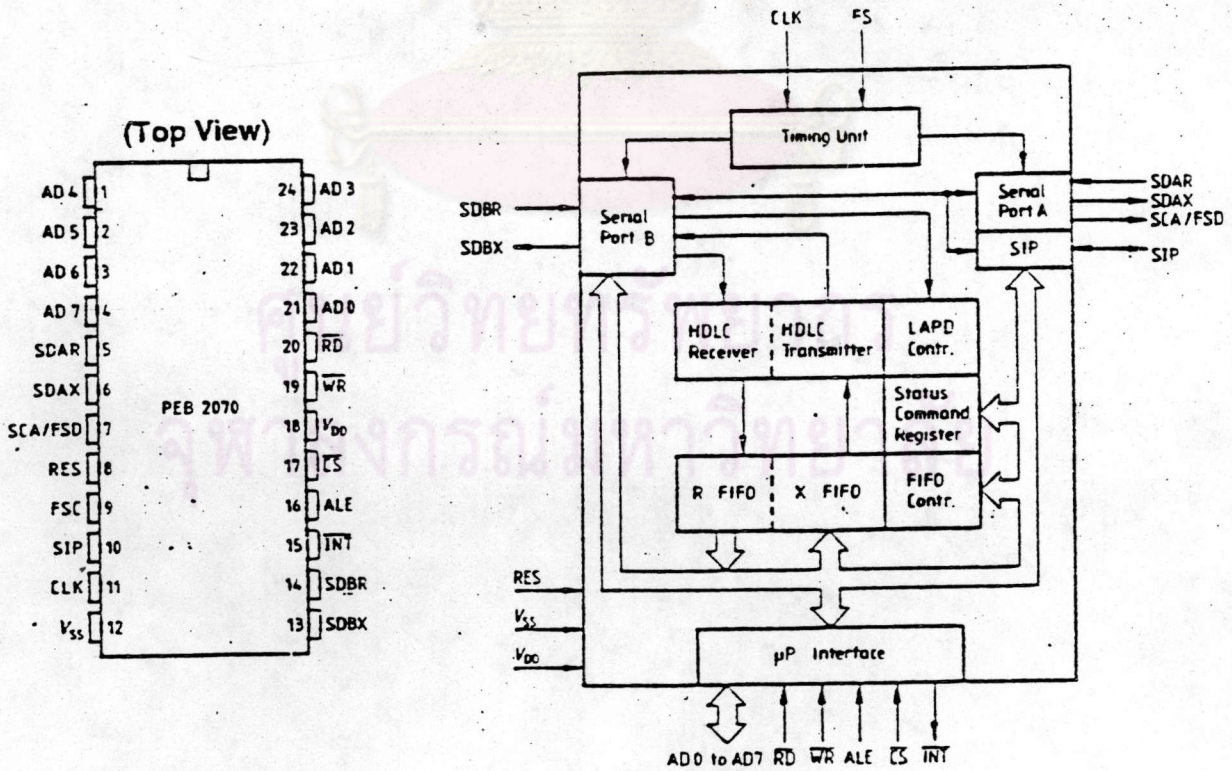
ภาคผนวก ข.

ไอซีทำหน้าที่ ISDN Controller

ไอซีที่เลือกมาใช้สำหรับทำหน้าที่ตามโปรโตคอล LAPD ในข่าย ISDN นั้นคือไอซีเบอร์ PEB 2070 หรือ ISDN Communications Controller (ICC) โดย ICC ตัวนี้จะทำหน้าที่ควบคุมการติดต่อในระดับโปรโตคอลชั้นที่ 2 ระหว่างอุปกรณ์ปลายทางของผู้เช่ากับชุมสายของข่าย ISDN รวมทั้งทำหน้าที่สร้างเฟรมข้อมูลตามแบบ ISDN คือ 2B+D และการเลือกใช้ช่องสัญญาณ B ในการส่งข้อมูลด้วย และเนื่องจากไอซี ICC นี้มีรายละเอียดที่คล้ายคลึงกับไอซี HSCC ดังนั้นในส่วนที่เหมือนกันจะไม่ยกมากล่าวอีกแต่จะใช้วิธีอ้างถึงในการกล่าวถึงรายละเอียดของไอซี ICC นี้

ข.1 ส่วนประกอบของ ICC

ส่วนประกอบของ ICC แสดงเป็นแผนภาพกรอบดังในรูปที่ ข.1



รูปที่ ข.1 แผนภาพกรอบแสดงส่วนประกอบของ ICC

ข.2 วงจรสำหรับเชื่อมโยงกับอุปกรณ์อื่นของ ICC

ไอซี ICC นี้มีวงจรสำหรับเชื่อมโยงกับอุปกรณ์อื่น ๆ 3 ส่วนด้วยกันคือ

- พอร์ทอนุกรม B (SPb) ซึ่งใช้งานเป็นวงจรเชื่อมโยงแบบ IOM หรือ HDLC
- พอร์ทอนุกรม A และพอร์ทสำหรับเชื่อมโยงกับอุปกรณ์ที่ให้ข้อมูลออกแบบ SLD (SIP) สำหรับเชื่อมโยงกับแหล่งกำเนิดและแหล่งรับข้อมูลที่รับส่งในช่องสัญญาณ B
- พอร์ทแบบขนานสำหรับต่อกับโปรเซสเซอร์

ข.2.1 พอร์ทอนุกรม B (Serial Port B:SPb)

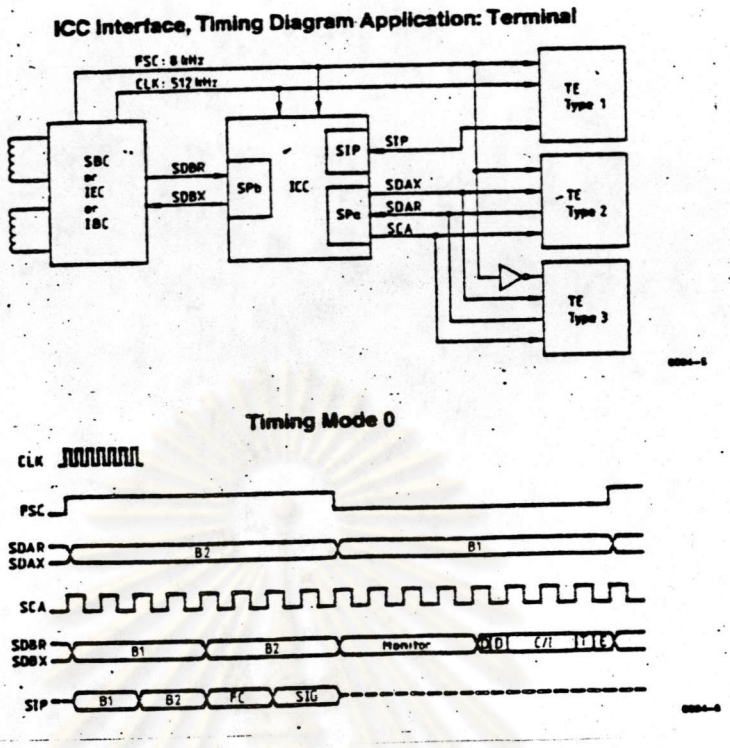
การใช้งานพอร์ทอนุกรม B สามารถกำหนดได้โดยการเขียนข้อมูลลงในรีจิสเตอร์ MODE ซึ่งจะสามารถเลือกใช้ได้ 2 แบบ คือการเชื่อมโยงแบบ IOM และแบบ HDLC ในกรณีของการใช้งานแบบ IOM นั้น ICC จะต่ออยู่กับอุปกรณ์ที่ทำหน้าที่ของโปรโตคอลชั้นที่ 1 ซึ่งการต่อจะสามารถเลือกได้ว่าจะต่อแบบจุดต่อจุด (Point-to-point) หรือจุดต่อหลายจุด (Point-to-Multipoint)

ข.2.1.1 โครงสร้างการเชื่อมโยงแบบ IOM การส่งข้อมูลโดยผ่านวงจรเชื่อมโยงแบบ IOM นั้น ข้อมูลที่ส่งจะส่งในลักษณะ 4 ไบต์ภายในระยะเวลา 125 ไมโครวินาที โดย 2 ไบต์แรกจะเป็นข้อมูลอัตราเร็ว 64 kbps สำหรับช่องสัญญาณชนิด B ไบต์ที่สามจะใช้สำหรับส่งข้อมูลระหว่าง ICC กับอุปกรณ์ทำหน้าที่โปรโตคอลชั้นที่ 1 ไบต์ที่ 4 ประกอบด้วยข้อมูลที่ใช้ควบคุมอุปกรณ์ทำหน้าที่โปรโตคอลชั้นที่ 1 ในการใช้งานวงจรจากเทอร์มินัลของผู้เช่าไปยังชุมสายและใช้ในการกำหนดลูบทดสอบ ข้อมูล 2 บิตภายในไบต์ที่ 4 ใช้ส่งข้อมูลอัตราเร็ว 16 kbps สำหรับช่องสัญญาณ D ดังแสดงในรูปที่ ข.2 และ ข.3

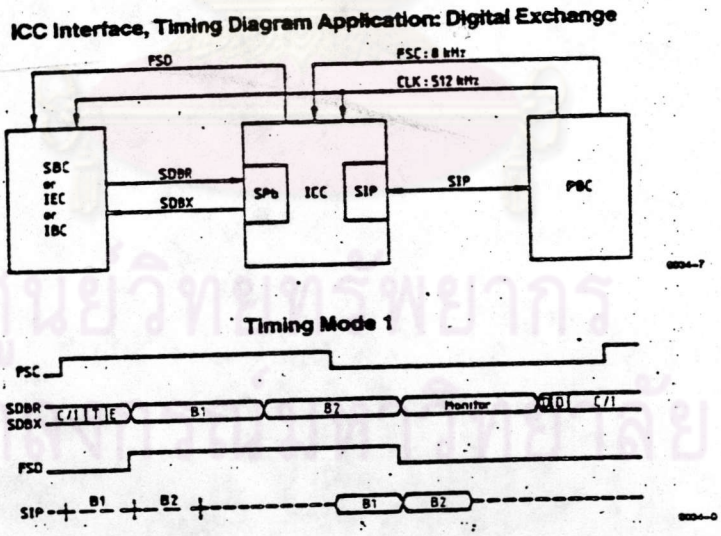
ดังนั้น เมื่อรวมข้อมูลทั้งหมดแล้วที่วงจรเชื่อมโยงแบบ IOM นี้ จะมีอัตราเร็วข้อมูล 256 kbps

ข.2.1.2 สัญญาณของวงจรเชื่อมโยงแบบ IOM วงจรเชื่อมโยงแบบ IOM ประกอบด้วยสายสัญญาณ 4 เส้นคือ สำหรับรับและส่งข้อมูลทิศทางละเส้น ผ่านขา SDBR และ SDBX ของ ICC และสัญญาณนาฬิกา 2 เส้น ทำหน้าที่สำหรับรับส่งข้อมูลและการ Synchronize เฟรมที่ทำการส่งผ่านทางขา CLK และ FSC ตามลำดับ สัญญาณนาฬิกาทั้ง 2 เส้นนี้ ICC จะได้รับมาจากอุปกรณ์ทำหน้าที่โปรโตคอลชั้นที่ 1 การใช้งานวงจรเชื่อมโยงแบบ IOM สามารถใช้ได้ 2 แบบคือแบบที่ 1 ใช้ภายในเทอร์มินัล และแบบที่ 2 ใช้ภายในชุมสาย

ข.2.1.3 การ Activate วงจรเชื่อมโยง IOM เมื่อ ICC อยู่ในสถานะว่าง ขาที่ใช้สำหรับรับส่งข้อมูลจะอยู่ที่ระดับ High ("1") และถ้าใช้งาน ICC ในเทอร์มินัลสัญญาณนาฬิกาที่ขา CLK และ FSC จะถูก Disable ในขณะที่ถ้าใช้ที่ชุมสายสัญญาณนาฬิกาทั้ง 2 เส้นจะ



รูปที่ ข.2 โครงสร้างวงจรเชื่อมโยงแบบ IOM เมื่อใช้ในเทอร์มินัล



รูปที่ ข.3 โครงสร้างวงจรเชื่อมโยงแบบ IOM เมื่อใช้ในชุมสาย

ยังคงมีอยู่ เมื่อใช้ ICC ในเทอร์มินัลวงจรเชื่อมโยง IOM จะถูกสั่งให้อยู่ในสถานะว่างโดยอุปกรณ์ทำหน้าที่โปรโตคอลชั้นที่ 1 ทุกครั้งที่อุปกรณ์ตัวนี้ถูกสั่ง Deactivate การสั่งให้วงจร IOM กลับมาทำงานอีกครั้งหนึ่งนั้นทำได้โดย อุปกรณ์ทำหน้าที่โปรโตคอลชั้นที่ 1 หรือสั่งโดยการเขียนข้อมูลลงในรีจิสเตอร์ของ ICC สั่งให้ ICC ส่งข้อมูลระดับ LOW ("0") ออกไปที่ขา SDBX

ข.2.2 พอร์ท SLD (SLD Interface Port:SIP)

พอร์ท SLD นี้จะทำหน้าที่รับส่งข้อมูลตามมาตรฐานการเชื่อมโยงแบบ SLD ซึ่งเป็นการรับส่งข้อมูล 2 ทิศทางที่ความถี่ 512 kHz บนสายสัญญาณเส้นเดียว เมื่อใช้งาน ICC ภายในชุมสาย ตัว ICC นี้จะต่อกับ PEB 2050 ผ่านทางพอร์ท SLD และ ICC จะรับสัญญาณนาฬิกาจาก PEB 2050 นี้ และถ้าใช้งาน ICC ภายในเทอร์มินัล พอร์ท SLD จะทำให้ ICC สามารถต่อกับ PEB 2060 หรือไอซีอื่น ๆ ที่ใช้พอร์ทแบบ SLD ได้ กรณีนี้สัญญาณนาฬิกา ICC จะได้รับจากอุปกรณ์ทำหน้าที่โปรโตคอลชั้นที่ 1

ข.2.3 พอร์ทอนุกรม A (Serial Port A: SPa)

พอร์ทนี้ใช้รับส่งข้อมูลแบบอนุกรมสำหรับอุปกรณ์ TE ซึ่งจะส่งข้อมูลสำหรับช่องสัญญาณชนิด B ทั้ง 2 ช่อง โดยมีอัตราเร็ว 128 Kbps และนอกจากขาสำหรับรับส่งข้อมูลคือ SDAR และ SDAX แล้ว อุปกรณ์ภายใน TE จะรับสัญญาณนาฬิกาจาก ICC และอุปกรณ์ที่ต่ออยู่กับพอร์ท A นี้สามารถรับสัญญาณนาฬิกาความถี่ 8 KHz จากอุปกรณ์ทำหน้าที่โปรโตคอลชั้นที่ 1 ที่มีขนาด Duty Cycle 50% เพื่อทำการกำหนดการรับส่งข้อมูลจาก Codec หรือ USART ได้ ดังแสดงในในรูปที่ ข.2

ข.2.4 วงจรเชื่อมโยงกับไมโครโปรเซสเซอร์ (uP Interface)

วงจรเชื่อมโยง uP ประกอบไปด้วยวงจรทำหน้าที่รับส่งข้อมูลกับบัสข้อมูล รีจิสเตอร์ และวงจรตรรกสำหรับควบคุมการทำงานของบัส โดยอาศัยพอร์ทนี้ ICC จะสามารถต่อโดยตรงกับ CPU ที่มีลักษณะของบัสข้อมูลแบบมัลติเพลกซ์ได้ โดยการอ้างตำแหน่งของรีจิสเตอร์ภายใน ICC ทำได้ในขณะที่ขาสัญญาณ ALE เป็น "1" และโดยอาศัยขาสัญญาณ READ และ WRITE ทำให้ CPU สามารถทำการอ่านและเขียนข้อมูลภายในรีจิสเตอร์ของ ICC ได้ ซึ่งการทำงานทุกอย่างของ ICC จะสามารถควบคุมได้โดยการเขียนข้อมูลลงในรีจิสเตอร์คำสั่งของ ICC และในขณะที่ ICC กำลังทำงาน CPU สามารถทราบสถานะของ ICC ได้โดยการอ่านค่าจากรีจิสเตอร์แสดงสถานะ และ CPU สามารถกำหนดโมดการทำงานของ ICC ได้โดยผ่านทางพอร์ทนี้เช่นเดียวกัน

ข.3 การประมวลข้อมูลบนวงจรเชื่อมโยง IOM ของ ICC (Processing of the IOM Interface Data in the ICC

ข.3.1 ช่องสัญญาณ B (B-channels) ข้อมูลอัตราเร็ว 64 kbps จะมีการรับส่งโดยตรงระหว่างพอร์ท SPb กับพอร์ท SLD หรือพอร์ท SPb กับพอร์ท SPa โดยการเลือกใช้ช่องสัญญาณ B1 หรือ B2 นั้น ทำได้โดยการเขียนข้อมูลลงในรีจิสเตอร์ SPCR

ข.3.2 ช่องสัญญาณ D (D-Channel) ภายในวงจรควบคุม HDLC ข้อมูลที่ได้รับมาในช่องสัญญาณ D จะถูกดึงออกจากส่วน I-field ของเฟรมของโปรโตคอลชั้นที่ 2 และ ICC จะเก็บไว้ในรีจิสเตอร์ RFIFO เพื่อรอให้ CPU มาทำการอ่านออกไป ส่วนในด้านการส่งนั้น CPU จะต้องทำการเขียนข้อมูลที่จะส่งออกลงไปในรีจิสเตอร์ XFIFO ส่วนข้อมูลในส่วนแอดเดรสและควบคุมของเฟรมชั้นที่ 2 จะทำโดย ICC เองหรือเขียนลงในรีจิสเตอร์ XFIFO โดย CPU

ข.3.3 การควบคุมวงจรชั้นที่ 1 (Commands and Indications to Control Layer-1) ข้อมูลที่จะใช้สำหรับการ Activate/Deactivate วงจรในชั้นที่ 1 นั้นจะสามารถส่งได้ทั้งจากเทอร์มินัลหรือชุมสาย โดย CPU ที่ต่ออยู่กับ ICC นั้นทำการเขียนข้อมูลจำนวน 4 บิต ลงในรีจิสเตอร์เฉพาะ จากนั้น ICC จะทำการส่งคำสั่งที่เขียนลงไปนั้นไปยังอุปกรณ์ชั้นที่ 1 ผ่านทางวงจรเชื่อมโยง IOM จนกว่า CPU จะเปลี่ยนคำสั่งในรีจิสเตอร์ดังกล่าว

ในระหว่างที่ทำขบวนการนี้อยู่นั้น จะมีการแลกเปลี่ยนสัญญาณมาตรฐานที่เรียกว่า Info ต่าง ๆ บนคู่สายที่จุดอ้างอิง S หรือ U และจะสิ้นสุดขบวนการก็ต่อเมื่ออุปกรณ์ชั้นที่ 1 ได้รับ Info ที่ต้องการ และ ICC จะทราบการสิ้นสุดขบวนการเมื่อได้รับข้อมูลจำนวน 4 บิต ที่ส่งมาจากอุปกรณ์ชั้นที่ 1 และเมื่อ ICC ได้รับข้อมูล 4 บิตนี้และได้ทำการตรวจสอบแล้วว่าไม่มีการเปลี่ยนแปลงในช่วงเวลาที่ ICC ทำการอ่าน 2 ครั้งสุดท้าย ICC จะสร้างสัญญาณขัดจังหวะแจ้งให้ CPU ทราบต่อไป

ข.3.4 ช่องสัญญาณสำหรับการตรวจสอบ (Monitor channel) เมื่อนำเอา ICC ไปใช้งานในเทอร์มินัลที่ต่ออยู่บนบัสที่จุดอ้างอิง S นั้น ICC จะสามารถทำการส่งข้อมูลออกไปบนคู่สายที่จุดอ้างอิง S ได้ก็ต่อเมื่อคู่สายว่างในขณะนั้น ซึ่ง ICC จะทราบโดยอุปกรณ์ชั้นที่ 1 แจ้งให้ทราบผ่านทางช่องสัญญาณสำหรับการตรวจสอบนี้ โดยถ้าข้อมูลบิตที่ใช้เป็น "0" แสดงว่าบัสว่าง ICC จะสามารถส่งข้อมูลออกไปได้ และถ้าเป็น "1" แสดงว่ากำลังมีการใช้งานบัสอยู่ และ ICC จะต้องรอจนกว่าบัสจะว่างจึงจะสามารถส่งข้อมูลออกไปได้

ข.4 ฟังก์ชั่นต่าง ๆ ของ ICC (Functions)

การกำหนดโมดการทำงานและการควบคุมลำดับขั้นตอนการทำงานต่าง ๆ ของ ICC นั้น สามารถทำได้โดยการอ่านและเขียนรีจิสเตอร์ภายในของ ICC ซึ่งฟังก์ชั่นต่าง ๆ ที่ ICC สามารถทำได้ จะประกอบไปด้วย

- การรับส่งข้อมูลระหว่าง ICC กับ CPU
- ควบคุมฟังก์ชั่นต่าง ๆ ของโปรโตคอลชั้นที่ 2
- กำหนดการเลือกใช้ช่องสัญญาณ B ในการส่งข้อมูล
- เข้าไปใช้งานช่องสัญญาณ 64 kbit

- ควบคุมการทำงานของอุปกรณ์ชั้นที่ 1
- กำหนดลูปทดสอบ

ในกรณีที่เกิดเหตุการณ์บางเหตุการณ์ขึ้นภายใน ICC ICC จะแจ้งให้ CPU ทราบโดยการส่งสัญญาณขัดจังหวะออกไป ซึ่ง CPU จะทราบสาเหตุของสัญญาณขัดจังหวะได้จากการอ่านค่าในรีจิสเตอร์ ISTA หรือ EXIR และในขณะใด ๆ CPU สามารถตรวจสอบสถานะของ ICC หรือของฝ่ายตรงข้ามได้โดยการอ่านรีจิสเตอร์ STAR และเมื่อใดก็ตามที่ CPU ได้รับสัญญาณขัดจังหวะ CPU จะต้องตอบรับการได้รับสัญญาณขัดจังหวะนั้น โดยการเขียนข้อมูลลงในรีจิสเตอร์ CMDR

ข.4.1 <u>การกำหนดโมดการทำงาน</u>

การกำหนดโมดการทำงานของวงจรควบคุม HDLC Timer และพอร์ทอนุกรม B สามารถทำได้โดยการเขียนข้อมูลลงในรีจิสเตอร์ MODE

ข.4.1.1 <u>โมดการทำงานของวงจรควบคุม HDLC</u> กำหนดได้ 5 โมดเช่นเดียวกับ HSCC ในภาคผนวก ก ยกเว้นใน ICC จะใช้รีจิสเตอร์ SAP1 SAP2 TEI1 และ TEI2 แทน RAH1 RAH2 RAL1 และ RAL2 ใน HSCC ตามลำดับ

ข.4.1.2 <u>โมดการทำงานของพอร์ทอนุกรม B</u> วงจรควบคุม HDLC ภายใน ICC ทำการรับและส่งข้อมูลผ่านทางพอร์ทอนุกรม B (SPb) ซึ่งพอร์ทนี้สามารถกำหนดให้มีรูปแบบการทำงานในลักษณะของการเชื่อมโยงแบบ IOM หรือ HDLC ได้ ในกรณีที่ใช้การเชื่อมโยงแบบ IOM นั้น ICC จะทำงานที่ความถี่ 512 kHz โดยจะส่งข้อมูลออกไปที่พอร์ทนี้ด้วยความถี่ 256 kHz โดยมีอัตราเร็วข้อมูลเป็น 16 kbps ในกรณีที่ใช้การเชื่อมโยงแบบ HDLC ความถี่ที่ใช้ในการส่งข้อมูลกับที่ ICC ใช้ในการทำงานจะเท่ากัน ส่วนอัตราเร็วข้อมูลนั้นจะขึ้นอยู่กับสัญญาณที่ได้รับที่ขา FSC

ข.4.2 <u>การรับส่งข้อมูลระหว่าง CPU และ ICC</u>

การรับส่งข้อมูลระหว่าง CPU กับ ICC นั้นทำได้วิธีเดียวกับ HSCC ในภาคผนวก ก

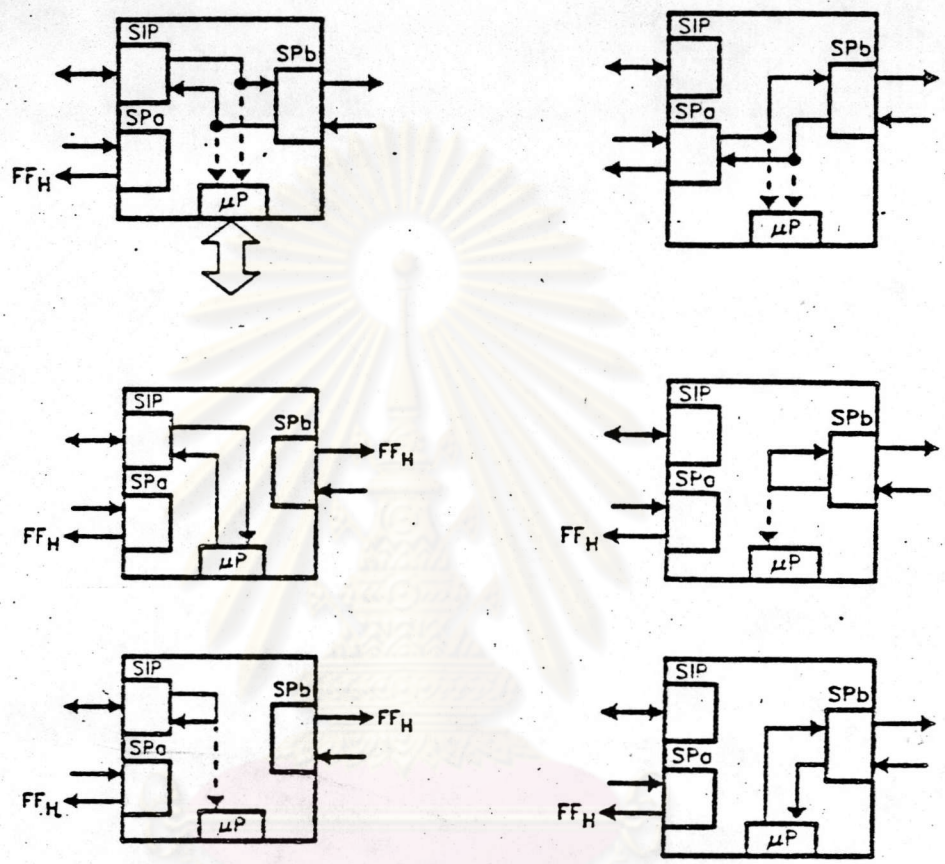
ข.4.3 <u>การทำงานในระดับโปรโตคอลชั้นที่ 2</u>

การทำงานในระดับโปรโตคอลชั้นที่ 2 ทำได้วิธีเดียวกับ HSCC ในภาคผนวก ก

ข.4.4 <u>การสวิตช์ช่องสัญญาณ B</u>

ICC สามารถสวิตช์ช่องสัญญาณ B ทั้ง 2 ช่องได้อย่างอิสระ โดยอาจจะเลือกส่งข้อมูลจากพอร์ท SLD และ/หรือพอร์ท SPa กับพอร์ท SPb โดยการเขียนข้อมูลลงในรีจิสเตอร์

SPCR โดยสามารถเลือกสวิตช์ช่องสัญญาณ B ได้ 4 แบบ ดังนี้



รูปที่ ข.4 การสวิตช์ช่องสัญญาณ B

ข.4.4.1 <u>สวิตช์พอร์ท SPb กับพอร์ท SLD</u> ดังแสดงในรูปที่ ข.4 การสวิตช์แบบนี้ ข้อมูลในช่องสัญญาณ B จากพอร์ท SLD จะถูกส่งตรงออกไปที่พอร์ท SPb ส่วนที่พอร์ท SPa ICC จะส่งข้อมูลออกเป็นค่า FF_H CPU สามารถทำการตรวจสอบข้อมูลที่กำลังรับส่งในช่องสัญญาณ Bได้

ข.4.4.2 <u>สวิตช์พอร์ท SPb กับพอร์ท SPa</u> ดังแสดงในรูปที่ ข.4 การสวิตช์แบบนี้ข้อมูลในช่องสัญญาณ B จากพอร์ท SPa จะถูกส่งตรงออกไปที่พอร์ท SPb ส่วนที่พอร์ท SLD ICC จะส่งข้อมูลเป็นค่า FF_H หรือค่าใด ๆ ออกไปขึ้นอยู่กับการเลือกใช้ช่องสัญญาณ B ของพอร์ท SPa CPU สามารถทำการตรวจสอบข้อมูลที่กำลังรับส่งในช่องสัญญาณ B ได้

ข.4.4.3 <u>ช่องสัญญาณ B ถูก disable</u> ดังแสดงในรูปที่ ข.4 การสวิตช์แบบนี้ ICC จะส่งข้อมูลค่า FF_H ออกทั้งที่พอร์ท SPa และ SPb และข้อมูลในช่องสัญญาณ B จะถูก

สะท้อนกลับที่พอร์ท SLD CPU สามารถตรวจสอบข้อมูลซึ่งส่งในช่องสัญญาณ B และถูกลูปอยู่ที่พอร์ท SLD ได้

ข.4.4.4 กำหนดลูปที่พอร์ท SPb ดังแสดงในรูปที่ ข.4 ICC จะส่งข้อมูลค่า FF_H ออกไปที่พอร์ท SPa ส่วนพอร์ท SLD นั้น ICC จะส่งข้อมูลค่า FF_H หรือค่าใด ๆ ขึ้นอยู่กับการเลือกใช้ช่องสัญญาณของพอร์ท SLD นั้น และข้อมูลในช่องสัญญาณ B จะถูกสะท้อนกลับที่พอร์ท SPa ซึ่งทำให้ CPU สามารถทำการตรวจสอบข้อมูลที่ทำการส่งในช่องสัญญาณ B ได้

ข.4.5 การเข้าไปใช้งานช่องสัญญาณอัตราเร็ว 64 kbit

CPU สามารถเข้าไปใช้งานช่องสัญญาณอัตราเร็ว 64 kbit ได้โดยการอ่านรีจิสเตอร์ BCR1 BCR2 หรืออ่านเขียนรีจิสเตอร์ BCX1 BCX2 และข้อมูลที่เกี่ยวข้องกับการใช้งานพอร์ท SLD คือ SIP Feature control นั้น สามารถเขียนและอ่านได้โดยใช้รีจิสเตอร์ SFCR และการ Synchronize กับเฟรมขนาดความยาว 125 us เพื่ออ่านและเขียนข้อมูลในช่องสัญญาณอัตราเร็ว 64 kbit นั้น ทำได้โดยการเขียนข้อมูลลงในรีจิสเตอร์ STCR การกำหนดการ Synchronize กับเฟรมขนาด 125 us นั้นทำได้ 2 แบบคือ แบบที่ 1 Synchronous transfer 1 การกำหนดแบบนี้สัญญาณขัดจังหวะ SIN ICC จะสร้างขึ้นที่จุดเริ่มต้นของเฟรม ส่วนแบบ Synchronous transfer 0 นั้น ICC จะสร้างที่จุดกึ่งกลางของเฟรม และเมื่อ ICC สร้างสัญญาณขัดจังหวะ SIN แล้ว CPU จะต้องตอบรับสัญญาณขัดจังหวะนี้ภายในเวลา 60 us โดยการเขียนข้อมูลลงในรีจิสเตอร์ STCR ถ้าการตอบรับโดย CPU ทำไม่ทันในเวลา 60 us ICC จะสร้างสัญญาณขัดจังหวะ SOV และจะหยุดขบวนการ

ข.4.5.1 การตรวจสอบช่องสัญญาณ B ที่พอร์ท SIP ทำได้โดยการกำหนดค่าในรีจิสเตอร์ดังนี้

รีจิสเตอร์ SPCR กำหนดให้ช่องสัญญาณ B ถูก disable

รีจิสเตอร์ SPCR กำหนด transfer 1

ในกรณีนี้ข้อมูลที่ส่งในพอร์ท SLD สามารถอ่านได้จากรีจิสเตอร์ BCX1 หรือ BCX2 สำหรับช่องสัญญาณ B1 หรือ B2 ตามลำดับ

ข.4.5.2 กำหนดลูปช่องสัญญาณ B ที่พอร์ท SLD ทำได้โดยการกำหนดค่าในรีจิสเตอร์ ดังนี้

รีจิสเตอร์ SPCR กำหนดให้ช่องสัญญาณ B ถูก disable

รีจิสเตอร์ STCR กำหนด transfer 1

หลังจากที่ CPU ได้รับสัญญาณขัดจังหวะแล้วจะสามารถตรวจสอบว่าข้อมูลในพอร์ท SLD พร้อมที่จะให้ CPU ทำการอ่านหรือไม่จากบิต BVS ในรีจิสเตอร์ STAR หลังจากนั้น CPU จะ

สามารถอ่านข้อมูลที่ ICC ได้รับมาจากพอร์ท SLD จากรีจิสเตอร์ BCX1 หรือ BCX2 และหลังจากนั้นข้อมูลที่ ICC จะส่งออกที่พอร์ท SLD จะสามารถเขียนลงในรีจิสเตอร์ BCX1 หรือ BCX2

ข.4.5.3 การตรวจสอบช่องสัญญาณ B ซึ่งเชื่อมต่อระหว่างพอร์ท SLD กับพอร์ท SPb ทำได้โดยการกำหนดค่าในรีจิสเตอร์ ดังนี้

รีจิสเตอร์ SPCR กำหนดให้สวิตช์ช่องสัญญาณ B สำหรับพอร์ท SLD

รีจิสเตอร์ STCR กำหนด transfer 0

ข้อมูลที่ได้รับจากพอร์ท SPb สามารถอ่านได้จากรีจิสเตอร์ BCX1 ข้อมูลที่ได้รับจากพอร์ท SLD จากรีจิสเตอร์ BCR1 หรือ BCR2 และก่อนที่จะทำการอ่านข้อมูลที่ได้จากพอร์ท SLD จะต้องตรวจสอบบิต BVS ในรีจิสเตอร์ STAR ก่อน

ข.4.5.4 การตรวจสอบช่องสัญญาณ B ซึ่งเชื่อมต่อระหว่างพอร์ท SPa กับพอร์ท SPb ทำได้โดยการกำหนดค่าในรีจิสเตอร์ ดังนี้

รีจิสเตอร์ SPCR กำหนดให้สวิตช์ช่องสัญญาณ B สำหรับพอร์ท SPa

รีจิสเตอร์ STCR กำหนด transfer 0

ข้อมูลที่ได้รับจากพอร์ท SPb อ่านได้จากรีจิสเตอร์ BCX1 หรือ BCX2 และข้อมูลที่ได้จากพอร์ท SPa อ่านได้จากรีจิสเตอร์ BCR1 หรือ BCR2

ข.4.5.5 การตรวจสอบช่องสัญญาณ B ที่พอร์ท SPb ทำได้โดยการกำหนดค่าในรีจิสเตอร์ ดังนี้

รีจิสเตอร์ SPCR กำหนดลูปที่พอร์ท SPb

รีจิสเตอร์ STCR กำหนด tranfer 0

ข้อมูลที่ได้รับที่พอร์ท SPb สามารถอ่านได้จากรีจิสเตอร์ BCX1 หรือ BCX2

ข.4.5.6 กำหนดลูปช่องสัญญาณ B ที่พอร์ท SPb ทำได้โดยการกำหนดค่าในรีจิสเตอร์ ดังนี้

รีจิสเตอร์ SPCR กำหนดให้ลูปที่พอร์ท SPb

รีจิสเตอร์ STCR กำหนด transfer 0

ข้อมูลที่ได้รับที่พอร์ท SPb สามารถอ่านได้จากรีจิสเตอร์ BCX1 หรือ BCX2 และข้อมูลที่จะส่งออกไปสามารถเขียนลงในรีจิสเตอร์ BCX1 หรือ BCX2 หลังจากนั้น

ข.4.6 การควบคุมอุปกรณ์ทำหน้าที่โปรโตคอลชั้นที่ 1

การแลกเปลี่ยนข้อมูลต่าง ๆ ระหว่าง ICC กับอุปกรณ์ที่ทำหน้าที่โปรโตคอลชั้นที่ 1 (เช่น IEC,SBC เป็นต้น) จะทำผ่านช่องสัญญาณสำหรับการตรวจสอบ (Monitor channel)

และช่องสัญญาณ C/I (ซึ่งประกอบด้วยข้อมูล 4 บิต ภายในช่องสัญญาณขนาด 64 kbps)

ข.4.6.1 ช่องสัญญาณสำหรับการตรวจสอบ (Monitor Channel)

ข.4.6.1.1 โมดการส่ง ใช้สำหรับสั่งงานอุปกรณ์ชั้นที่ 1 โดยการเขียนข้อมูลลงในรีจิสเตอร์ MONR ซึ่ง ICC จะส่งออกไปในช่องสัญญาณสำหรับการตรวจสอบ และหลังจากส่งข้อมูลในรีจิสเตอร์ MONR แล้ว ICC จะส่งคำสั่ง NOP (FO_H-FF_H) ต่อไปโดยอัตโนมัติ

ข.4.6.1.2 โมดการรับ การใช้งานช่องสัญญาณสำหรับการตรวจสอบในโมดการรับสามารถใช้งานได้ 2 แบบ ขึ้นอยู่กับการเขียนข้อมูลลงในรีจิสเตอร์ MODE ดังนี้

แบบที่ 1 ส่งข้อมูลควบคุมโดย ICC จะใช้ข้อมูลบิตที่ 3 ในการควบคุมการทำงานของวงจรสำหรับส่ง HDLC การใช้งานแบบนี้จะใช้ในการควบคุมการใช้ช่องสัญญาณ D เมื่อต่ออุปกรณ์ TE บนบัส

แบบที่ 2 ส่งข้อมูลเมื่อข้อมูลบิต E ในช่องสัญญาณ B* เป็น "0" ICC จะทำการเก็บข้อมูลไบต์ต่อไปที่ได้รับจากช่องสัญญาณนี้ไว้ในรีจิสเตอร์ MONR และ ICC จะสร้างสัญญาณขัดจังหวะ MOR ส่งให้ CPU ต่อไป การใช้งานแบบนี้จะใช้ในการอ่านข้อมูลจาก IEC

ข.4.6.2 บัส TIC ในกรณีที่ใช้งานพอร์ท SPb แบบ IOM จะสามารถต่อ ICC เข้ากับอุปกรณ์ชั้นที่ 1 ได้มากถึง 8 ตัว ในลักษณะของการต่อแบบบัส

ข.4.6.2.1 การต่อแบบบัส เมื่อต่อ ICC เข้ากับอุปกรณ์ชั้นที่ 1 ในแบบบัสการต่อจะอยู่ในลักษณะจุดต่อหลายจุด โดยขาส่งข้อมูลออกของ ICC จะต่อแบบ Wired-or กับอุปกรณ์ชั้นที่ 1

ข.4.6.2.2 การเข้าไปใช้งานบัส การเข้าไปใช้งานบัสทำได้โดยอาศัยช่องสัญญาณสำหรับการตรวจสอบ ซึ่งอยู่ภายในช่องสัญญาณ B* และข้อมูลที่รับส่งในช่องสัญญาณ B1 หรือ B2 จะต่อในลักษณะ Wired-or การเข้าไปใช้ช่องสัญญาณ B* สามารถทำได้ ดังนี้

- เข้าไปใช้ช่องสัญญาณ D ทำได้โดย ICC ส่งคำขอเพื่อเข้าใช้งาน
- เข้าไปใช้ช่องสัญญาณ C/I ทำได้โดย CPU ส่งคำขอไปยัง ICC

ข.4.6.3 ช่องสัญญาณ C/I (C/I Channels)

ข.4.6.3.1 โมดการส่ง ข้อมูลที่ CPU เขียนลงในรีจิสเตอร์ CIXR จะถูก ICC ส่งออกไปในช่องสัญญาณ C/I ไปยังอุปกรณ์ชั้นที่ 1 อย่างต่อเนื่องโดยขณะนี้บัส TIC จะ

ต้องอยู่ในสถานะไม่ว่าง การขอเข้าไปใช้งานบัส TIC จะสามารถทำได้โดยการเขียนข้อมูลลงในบิต TBC ในรีจิสเตอร์ CIXR เช่นเดียวกัน

ข.4.6.3.2 โมดการรับ ICC จะได้รับข้อมูลที่ส่งมาในช่องสัญญาณ C/I ตลอดเวลาและเมื่อข้อมูลที่ได้รับนั้นมีการเปลี่ยนแปลง ICC จะเก็บข้อมูลนี้ไว้ในรีจิสเตอร์ CIRR และสถานะของบัส TIC CPU จะทราบได้โดยการอ่านบิต TBA ในรีจิสเตอร์ CIRR เช่นกัน โดย CPU จะรับทราบการเปลี่ยนแปลงของข้อมูลนี้โดยสัญญาณขัดจังหวะ CIC ซึ่งส่งมาจาก ICC

ข.4.6.4 <u>การ Activate/Deactivate</u> เมื่อ ICC อยู่ในสถานะ Deactivate สัญญาณนาฬิกาที่ ICC เคยได้รับจะถูก disable และขาส่งข้อมูลออก SDBX จะอยู่ที่ระดับ "1" และ ICC จะถูก Activate ได้โดยอุปกรณ์ชั้นที่ 1 ส่งสัญญาณนาฬิกาและข้อมูลแจ้งการขอ Activate ให้ ICC ในช่องสัญญาณ C/I พร้อมกับที่ ICC จะสร้างสัญญาณขัดจังหวะ CIC ส่งให้ CPU และถ้าจะให้ ICC ได้รับการ Activate จาก CPU ก็ทำได้โดยเขียนข้อมูลลงในรีจิสเตอร์ SPCR สั่งให้ ICC ส่งข้อมูลระดับ "0" ออกที่ขา SDBX ซึ่งเมื่ออุปกรณ์ชั้นที่ 1 ได้รับสัญญาณที่ส่งออกจากขา SDBX ของ ICC ก็จะส่งสัญญาณนาฬิกาพร้อมกับส่งข้อมูลแจ้งการ Activate มาให้ ICC ทราบ

นอกจากนั้นวงจรเชื่อมโยง IOM สามารถสั่งให้ Deactivate ได้จาก CPU โดยเขียนคำสั่งลงในรีจิสเตอร์ CIXR และส่งต่อออกไปให้อุปกรณ์ชั้นที่ 1 และ CPU จะรับทราบการ Deactivate ของ ICC โดยสัญญาณขัดจังหวะเมื่อ ICC ได้รับคำตอบและสัญญาณนาฬิกาถูก disable

ข.4.7 <u>โมดทดสอบ</u>

ICC สามารถทำการทดสอบตัวเองได้โดยการเขียนข้อมูลลงในบิต TLP ในรีจิสเตอร์ SPCR ซึ่งเมื่ออยู่ในโมดนี้ขาสัญญาณเข้าและออกของพอร์ท SPb จะถูกต่อเข้าด้วยกัน โดยในระหว่างอยู่ในโมดนี้อัตราเร็วการทำงานของ ICC จะเพิ่มขึ้น 64 เท่า

ข.5. <u>โมดการโปรแกรม ICC</u>

ตัวแปรต่าง ๆ ที่ ICC ใช้ในการทำงานรวมทั้งข้อมูลที่ทำการรับส่งระหว่าง CPU กับ ICC สามารถกำหนดได้โดยการเขียนและอ่านข้อมูลในรีจิสเตอร์ต่าง ๆ ของ ICC ซึ่งแยกได้เป็น 3 กลุ่ม คือ

- รีจิสเตอร์สำหรับการรับส่งข้อมูลระหว่าง ICC กับ CPU
- รีจิสเตอร์สำหรับควบคุมวงจรควบคุม HDLC
- รีจิสเตอร์สำหรับควบคุมวงจรรับส่งแบบอนุกรม

ข.5.1 รีจิสเตอร์สำหรับควบคุมวงจรควบคุม HDLC

ข.5.1.1 ตำแหน่งแอดเดรสของรีจิสเตอร์ต่าง ๆ เป็นดังตารางที่ ข.1

ตารางที่ ข.1 แอดเดรสของรีจิสเตอร์

Address$_H$	Read	Write
00-1F	RFIFO	XFIFO
20	ISTA	MASK
21	STAR	CMDR
22	MODE	MODE
23	TIMR	TIMR
24	EXIR	XAD1
25	RFBC	XAD2
26		SAP1
27	RSTA	SAP2
28	TEI1	TEI1
29	RHCR	TEI2

ข.5.1.2 รีจิสเตอร์ควบคุม สำหรับความหมายของบิตต่าง ๆ ที่ไม่มีรายละเอียดสามารถพลิกดูได้ในภาคผนวก ก ที่มีชื่อเดียวกัน

ข.5.1.2.1 Interrupt Status Register(ISTA) อ่านแอดเดรส 20_H ค่าภายหลังการ Reset คือ 00_H

7							0
RME	RPF	RSC	XPR	TIN	CIC	SIN	EXI

CIC - C/I Code Change; ICC จะส่งสัญญาณขัดจังหวะนี้ให้ CPU เมื่อ ICC ได้รับทราบการเปลี่ยนแปลงของข้อมูลในช่องสัญญาณ C/I

SIN - Synchronous transfer interrupt; เกิดขึ้นเมื่อ ICC Synchronize กับเฟรมขนาด 125 us ตามที่ได้กำหนดในรีจิสเตอร์ STCR

ข.5.1.2.2 Mask Register(MASK) เขียนแอดเดรส 20_H ค่าภายหลังการ Reset คือ 00_H

ข.5.1.2.3 Extended Interrupt Rigister(EXIR) อ่านแอดเดรส 24_H ค่าภายหลังการ Reset คือ 00_H

7							0
XMR	XDU	PCE	RFO	SOV	MOR	0	0

SOV - Synchronous transfer overflow; CPU ไม่สามารถตอบสนองการเกิดสัญญาขัดจังหวะ SIN ได้ในเวลาที่กำหนด (โดยการเขียนข้อมูลลงในรีจิสเตอร์ STCR)

MOR - Monitor byte receive ICC ได้รับข้อมูลที่ส่งมาในช่องสัญญาณสำหรับการตรวจสอบและได้เก็บไว้ในรีจิสเตอร์ MONR

ข.5.1.2.4 Status Register(STAR) อ่านแอดเดรส 21_H ค่าภายหลังการ Reset คือ 48_H

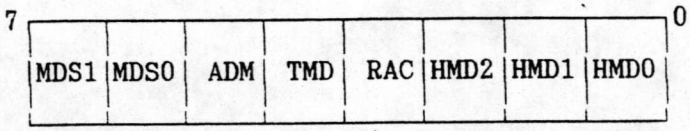
7							0
XDOV	XFW	XRNR	RRNR	MBR	0	BVS	0

BVS - B channel valid on SIP; CPU สามารถเข้าไปใช้งานช่องสัญญาณ B ที่พอร์ท SLD ได้

ข.5.1.2.5 Command Register (CMDR) เขียนแอดเดรส 21_H ค่าภายหลังการ Reset คือ 00_H

7							0
RMC	RHR	RNR	STI	XTF	XIF	XME	XRES

ข.5.1.2.6 Mode Register (MODE) อ่านเขียนแอดเดรส 22_H ค่าภายหลังการ Reset คือ 00_H



HDM2-0 HDLC port mode; กำหนดโมดการทำงานของพอร์ท B ดังนี้

000 - เชื่อมโยงแบบ IOM ในลักษณะจุดต่อจุดหรือใช้ใน NT ในลักษณะแบบจุดต่อหลายจุด กรณีนี้จะไม่มีการใช้งานช่องสัญญาณสำหรับการตรวจสอบ

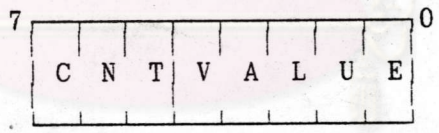
001 - เชื่อมโยงแบบ IOM ใช้ที่อุปกรณ์เทอร์มินัลที่ต่ออยู่บนบัส S ในแบบจุดต่อหลายจุด กรณีนี้ช่องสัญญาณสำหรับการตรวจสอบจะใช้สำหรับควบคุมการเข้าไปใช้งานช่องสัญญาณ D

101 - เชื่อมโยงแบบ HDLC สัญญาณ Strobe จะ Active เมื่ออยู่ที่ระดับ "0" สำหรับช่องสัญญาณ B1

110 - เชื่อมโยงแบบ HDLC สัญญาณ Strobe จะ Active เมื่ออยู่ที่ระดับ "1" สำหรับช่องสัญญาณ B2

111 - เชื่อมโยงแบบ HDLC สัญญาณนาฬิกาสำหรับการรับส่งข้อมูลและการทำงานของไอซีจะเท่ากัน

ข.5.1.2.7 Timer Register (TIMR) อ่านเขียนแอดเดรส 23_H



CNT - Count ความหมายของค่านี้ขึ้นอยู่กับการกำหนดโมดการทำงานให้ ICC

VALUE - กำหนดคาบเวลาโดยจะคำนวณได้จาก

$$t1 = (VALUE + 1) \times 2^{15} \times 1/cp$$

ถ้า CP = 512 KHz; t1 = (VALUE + 1) x 64 ms

สำหรับการทำงานของ timer ในโมด External; ค่า CNT บวกกับ VALUE ใช้ในการคำนวณหาคาบเวลา t2 สำหรับ ICC ที่จะสร้างสัญญาณขัดจังหวะ TIN ให้ CPU เมื่อหมดคาบเวลา t1 การคำนวณได้จาก

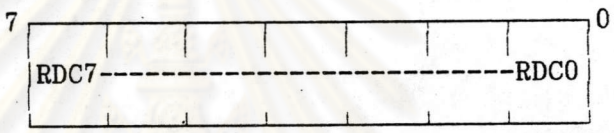
$$t2 = CNT \times 2^{20} \times 1/CP + t1$$

ถ้า CP = 512 KHz, t2 = CNT x 2.048 + t1 ms

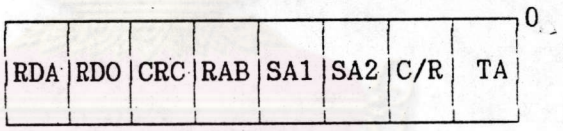
และถ้า CNT = 7 ICC จะสร้างสัญญาณขัดจังหวะ TIN ส่งให้ CPU ทุกครั้งที่หมดคาบเวลา t1

สำหรับการทำงานของ Timer ในโมด Internal; CNT จะแสดงจำนวนเฟรมชนิด S ที่ ICC จะสามารถส่งออกไปได้ในขณะที่รอคำตอบจากฝ่ายตรงข้ามเมื่อหมดคาบเวลา t1 โดยค่ามากที่สุดคือ 6

ข.5.1.2.8 Receive Frame Byte Counter (RFBC) อ่านแอดเดรส 25_H ค่าภายหลังการ Reset คือ 00_H



ข.5.1.2.9 Receive Status Register (RSTA) อ่านแอดเดรส 27_H



SA1 - 0 - SAPI address : บอกค่า SAPI ของเฟรมที่ได้รับ ดังนี้

10 - ค่าในรีจิสเตอร์ SAPI1

00 - ค่าในรีจิสเตอร์ SAPI2

01 - ค่าคงที่ FE_H หรือ FC_H

TA - TEI address 0 คือค่าในรีจิสเตอร์ TEI2 และ 1 คือค่าใน TEI1

ข.5.1.3 <u>รีจิสเตอร์ข้อมูล</u>

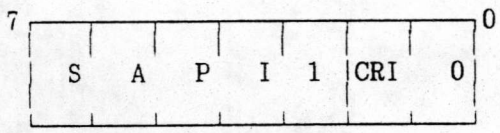
ข.5.1.3.1 Receive FIFO (RFIFO) อ่านแอดเดรส 00_H

ข.5.1.3.2 Transmit FIFO (XFIFO) เขียนแอดเดรส 00_H

ข.5.1.3.3 Transmit Address 1 (XAD1) เขียนแอดเดรส 24_H ค่าที่เขียนในรีจิสเตอร์นี้คือค่า SAPI ที่จะให้ ICC ทำการส่งในโมด Auto

ข.5.1.3.4 Transmit Address 2 (XAD2) เขียนแอดเดรส 25_H ค่าที่เขียนในรีจิสเตอร์นี้คือค่า TEI ที่จะให้ ICC ทำการส่งในโมด Auto

ข.5.1.3.5 SAPI Register 1 (SAP1) เขียนแอดเดรส 26H

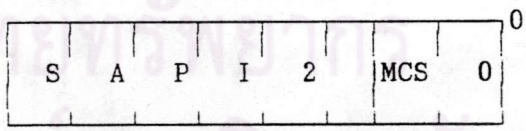


SAPI1 - ค่าแรกของค่า SAPI ที่จะใช้ในการรับเฟรมข้อมูล

CRI - Command -response interpretation; ในการทำงานในโมด Auto ค่าในบิต C/R ICC จะทำการกำหนด ดังนี้

	CRI = 1 C/R Value	CRI = 0 C/R Value
Commands rec.	0	1
Responses rec.	1	0
Commands trm.	1	0
Responses trm.	0	1

ข.5.1.3.6 SAPI Register 2 (SAP2) เขียนแอดเดรส 27H



SAPI2 - ค่า SAPI ค่าที่สองที่จะใช้ในการรับเฟรมข้อมูล

MCS - Module count select; ใช้ในการแยกระหว่างใช้ modulo 8 ("0") และ Modulo 128 ("1") สำหรับการรับส่งเฟรมข้อมูล และใช้ได้เฉพาะในโมด Auto เท่านั้น

ข.5.1.3.7 TEI Register 1(TEI1) อ่านเขียนรีจิสเตอร์ 28H ค่าในรีจิสเตอร์ TEI1 นี้ในโมด Auto และ Non-outo จะใช้สำหรับเป็นค่าที่จะใช้ในการรับเฟรมข้อมูล ส่วนในโมด Transparent และ Extended Transparent นั้น จะเป็น

ค่าของข้อมูลไบต์แรกถัดจากแฟล็ก

ข.5.1.3.8 TEI Register 2(TEI2) เขียนแอดเดรส 29H เป็นค่า TEI ค่าที่สองที่จะใช้ในการรับเฟรมข้อมูล

ข.5.1.3.9 Receive HDLC Control Register (RHCR) อ่านรีจิสเตอร์ 29H เป็นค่าที่ ICC ได้จากส่วนควบคุมของเฟรม HDLC หรือค่าในไบต์ที่สองถัดจากแฟล็กในโมด Extended Transparent

ข.5.2 รีจิสเตอร์สำหรับวงจรเชื่อมโยงแบบอนุกรม

ข.5.2.1 แอดเดรสของรีจิสเตอร์แสดงในตารางที่ ข.2

ตารางที่ ข.2 แสดงแอดเดรสของรีจิสเตอร์

AddressH	Read	Write
30	SPCR	SPCR
31	CIRR	CIXR
32	MONR	MONR
33	SSGR	SSGX
34	SFCR	SFCR
35	BCX1	BCX1
36	BCX2	BCX2
37	BCR1	BCR2
38	BCR2	BCR2

ข.5.2.2 รีจิสเตอร์ควบคุม

ข.5.2.2.1 Serial Port Control Register(SPCR) อ่านเขียนแอดเดรส 30H ค่าภายหลังการ Reset คือ 00H

7							0
BPZ	SAC	SPM	TLP	B1C1	B1C0	B2C1	B2C0

BPZ - Set B port to zero; สั่งให้ ICC ส่งข้อมูลค่า "0" ออกที่ขา SDBX

SAC - SIP port activated; กำหนดให้พอร์ท SLD อยู่ในสถานะ High impedance ("0") หรือ Activate ("1")

SPM - Serial port timing mode; เลือกโมดการใช้งานของ ICC ระหว่างโมด 0 และโมด 1

TPL - Test loop; สั่งให้ ICC ต่อขาสัญญาณเข้าและออกเข้าด้วยกัน กรณีนี้ คาบเวลา t1 และ t2 จะลดลงด้วยอัตราส่วน 64 เท่า

B1C1, B1C0 กำหนดการใช้ช่องสัญญาณ B1

B2C1, B2C0 กำหนดการใช้ช่องสัญญาณ B2

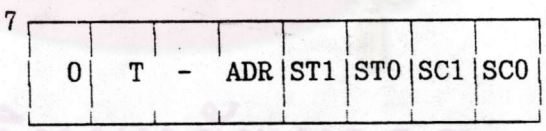
00 - disable ช่องสัญญาณ B และช่องสัญญาณ B สะท้อนกลับที่พอร์ท SLD

01 - สวิตช์ระหว่างพอร์ท SLD กับพอร์ท SPb

10 - สวิตช์ระหว่างพอร์ท SPa กับพอร์ท SPb

11 - ช่องสัญญาณ B สะท้อนกลับที่พอร์ท SPb หรือสวิตช์พอร์ท SPb ให้ CPU

ข.5.2.2.2 Synchronous Transfer Control Register (STCR) เขียนแอดเดรส 27_H ค่าภายหลังการ Reset คือ 00_H



T-ADR - TIC Bus address; แอดเดรสของอุปกรณ์แต่ละตัวที่ต่ออยู่บนบัส

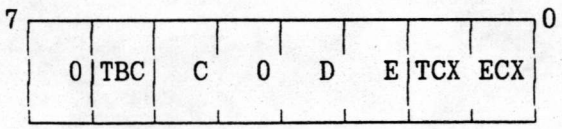
ST1 - Synchronous transfer 1; จะเกิดสัญญาณขัดจังหวะที่จุดเริ่มต้นเฟรม

ST0 - Synchronous transfer 0; จะเกิดสัญญาณขัดจังหวะที่จุดกึ่งกลางเฟรม

SC1 - Synchronous transfer 1 completed; เป็นการแจ้งการได้รับสัญญาณขัดจังหวะ SIN ซึ่งต้องตอบก่อนที่จะถึงจุดกึ่งกลางเฟรมมิฉะนั้นจะทำให้เกิดการ Overflow ได้

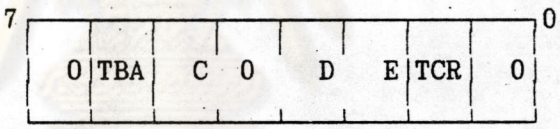
SC0 - Synchronous transfer 0 completed; เป็นการแจ้งการได้รับสัญญาณขัดจังหวะ SIN ซึ่งจะต้องตอบก่อนที่จะถึงจุดเริ่มต้นของเฟรมถัดไป มิฉะนั้นจะทำให้เกิดการ Overflow ได้

ข.5.2.2.3 Command/Indicate Transmit Register (CIXR) เขียนแอดเดรส 31_H ค่าภายหลังการ RESET คือ $3F_H$



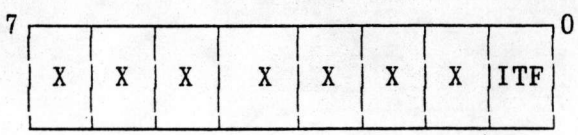
TBC - TIC - Bus control; บัสถูกใช้งานอยู่
CODE- ข้อมูลที่ ICC ส่งออกไปที่พอร์ท SPb ในช่องสัญญาณ C/I
TCX - T channel transmit; ข้อมูลที่ส่งออกที่พอร์ท SPb ในช่องสัญญาณ T
ECX - E channel transmit; ข้อมูลที่ส่งออกที่พอร์ท SPb ในช่องสัญญาณ E

ข.5.2.2.4 Command/Indicate Receive Registor (CIRR) อ่านแอดเดรส 31_H ค่าภายหลังการ Reset คือ $7C_H$



TBA - TIC - bus access; ICC กำลังใช้งานบัส TIC (TBA = 0)
CODE - ICC ได้รับข้อมูลในช่องสัญญาณ C/I จากพอร์ท SPb
TCR - T channel receive; ICC ได้รับข้อมูลในช่องสัญญาณ T จากพอร์ท SPb

ข.5.2.2.5 Serial Port Feature Register (SPFR) เขียนแอดเดรส 38_H ค่าภายหลังการ Reset คือ 00_H



ITF - Interframe time fill; กำหนดข้อมูลที่จะส่งโดยวงจรเครื่องส่ง HDLC ระหว่างส่งเป็น "1" (ITF = 0) และส่งข้อมูลแฟล็ก (ITF = 1)

ข.5.2.3 <u>รีจิสเตอร์ข้อมูล</u>

ข.5.2.3.1 Monitor Register(MONR) อ่านเขียนแอดเดรส 32_H ข้อมูลที่ CPU เขียนลงในรีจิสเตอร์นี้ ICC จะส่งออกไปในช่องสัญญาณสำหรับการตรวจสอบที่พอร์ท SPb

ข.5.2.3.2 B channel Transmit 1/2 (BCX1/2) อ่านเขียนแอดเดรส 35/36H เขียนและอ่านข้อมูลในบัฟเฟอร์ของช่องสัญญาณ B ในโมดการส่ง

ข.5.2.3.3 B channel Receive Register 1/2(BCR1/2) อ่านแอดเดรส $37/38_H$ อ่านข้อมูลในบัฟเฟอร์ของช่องสัญญาณ B ในโมดการรับ

ข.5.2.3.4 SIP Signalling Register Transmit (SSGX) เขียนแอดเดรส 33_H ข้อมูลที่ CPU เขียนลงในรีจิสเตอร์นี้ ICC จะส่งออกในไบต์ Signalling ที่พอร์ท SLD

ข.5.2.3.5 SIP Signalling Register Receive (SSGR) อ่านแอดเดรส 33_H ข้อมูลในไบต์ Signalling ที่ ICC ได้รับจากพอร์ท SLD CPU สามารถอ่านได้จากรีจิสเตอร์นี้

ข.5.2.3.6 SIP Feature Control Register(SFCR) อ่านเขียนแอดเดรส 34_H ข้อมูลที่ CPU เขียนลงในรีจิสเตอร์ ICC จะส่งออกไปที่พอร์ท SLD ในช่องสัญญาณ FC และข้อมูลที่ ICC ได้รับจากไบต์ FC CPU สามารถอ่านได้จากพอร์ทนี้เช่นกัน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

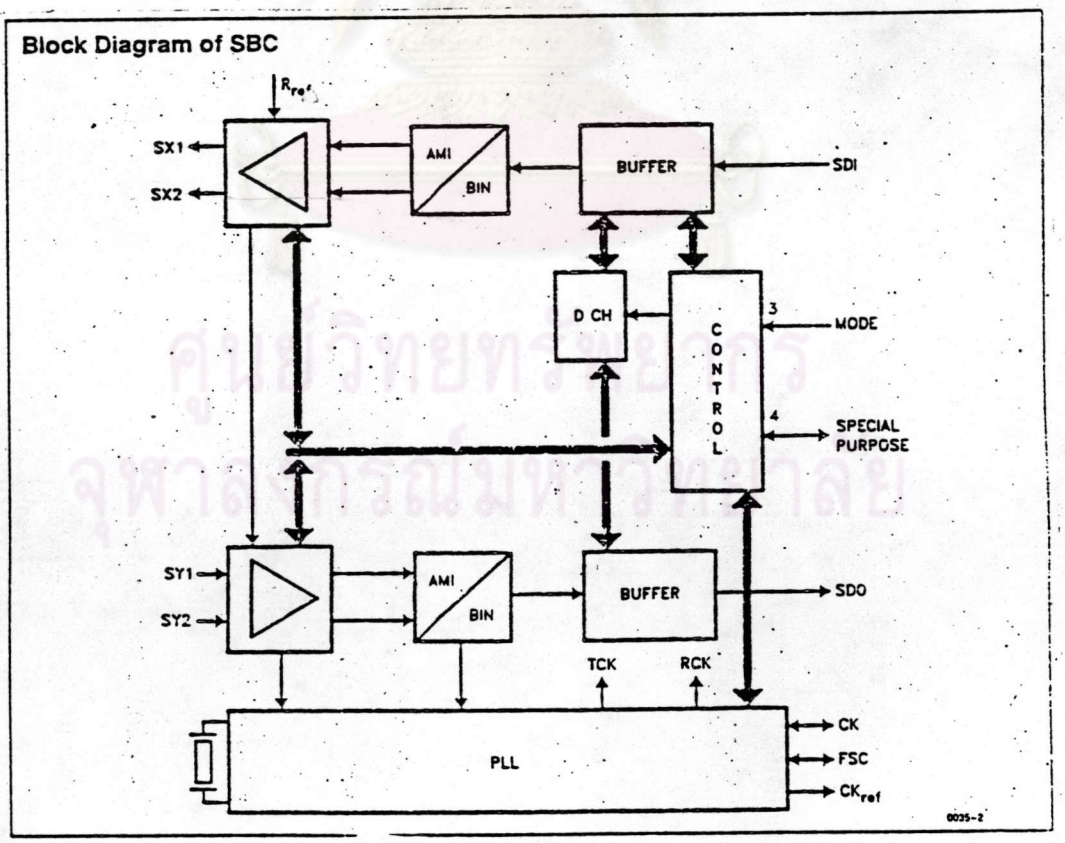
ภาคผนวก ค.

ไอซีทำหน้าที่เชื่อมโยงที่จุดอ้างอิง S

การเชื่อมโยงของอุปกรณ์เทอร์มินัลกับบัสที่จุดอ้างอิง S นั้น การทำงานของอุปกรณ์เทอร์มินัลจะต้องเป็นไปตามมาตรฐานที่กำหนดโดย CCITT I.430 ซึ่งครอบคลุมตั้งแต่ลักษณะการเชื่อมต่อ สัญญาณที่รับส่งผ่านจุดอ้างอิง S และส่วนประกอบของเฟรมข้อมูลที่ส่งออกไป ในการพัฒนาแผงควบคุมวงจรนี้ได้เลือกใช้ไอซีเบอร์ PEB2080 (SBC) ทำหน้าที่เป็นอุปกรณ์เชื่อมโยงที่จุดอ้างอิง S ซึ่งจะทำหน้าที่ตามโปรโตคอลชั้นที่ 1 ของข่าย ISDN

ค.1 ส่วนประกอบภายในของ SBC

ไอซีที่เลือกใช้นี้สามารถทำขบวนการ Activate/Deactivate ตามข้อกำหนดของ CCITT โดยอาศัย State Control Block ภายในของ SBC เองโดยไม่ต้องอาศัย CPU แต่อย่างใด ส่วนประกอบภายในของ SBC แสดงได้เป็นแผนภาพกรอบดังแสดงในรูปที่ ค.1



รูปที่ ค.1 แสดงแผนภาพกรอบของ SBC

ค.2 การเชื่อมโยง

SBC จะทำหน้าที่เชื่อมโยงระหว่างข้อมูลที่ได้รับที่จุดอ้างอิง S กับวงจรเชื่อมโยง IOM ภายในอุปกรณ์เทอร์มินัลหรือชุมสาย การใช้งาน SBC สามารถกำหนดได้โดยอาศัยสัญญาณที่ขาของ SBC แบบต่าง ๆ ดังแสดงในตารางที่ ค.1

ตารางที่ ค.1 แสดงฟังก์ชันของขาสัญญาณเข้าและออกของ SBC ที่โมดต่าง ๆ

Application	Operation Mode	M2	M1	M0	DCL	FSC	CP	X3	X2	X1	X0
TE	Inverted Mode	0	0	0	o: 512 kHz*	o: 8 kHz*	o: 1536 kHz*	i: $\overline{\text{ENCK}}$	o: 2560 kHz	o: 3840 kHz	o: RDY
TE	Inverted Mode	0	0	1	o: 512 kHz*	o: 8 kHz*	o: 1536 kHz*	i: $\overline{\text{ENCK}}$	o: 1280 kHz	o: 3840 kHz	o: RDY
TE	Normal Mode	0	1	0	o: 512 kHz*	o: 8 kHz*	o: 1536 kHz*	i: $\overline{\text{ENCK}}$	o: ECHO	o: 3840 kHz	i: CON
TMD	MUX Mode	0	1	1	i: 4096 kHz	i: 8 kHz	o: 512 kHz*	i: 1	i: TS2	i: TS1	i: TS0
TMD	Normal Mode	0	1	1	i: 512 kHz	i: 8 kHz	o: 512 kHz*	i: 0	—	—	i: CON
NT	Normal Mode	1	1	1	i: 512 kHz	i: 8 kHz	i: TM2	i: BUS	i: SSP	i: DEX	i/o: DE
SLMD	MUX Mode	1	0	0	i: 4096 kHz	i: 8 kHz		i: BUS	i: TS2	i: TS1	i: TS0
SLMD	Normal Mode	1	1	0	i: 512 kHz	i: 8 kHz		i: BUS	—	o: 7680 kHz	i: o

* Synchronized to S i: Input o: Output

Output Stages

Application	Module Interface Operating Mode	M2	M1	M0	DCL	FSC	CP	X2	X1	X0	SD0
TE	512 kHz Inverted	0	0	0	Push/Pull	Push/Pull	Push/Pull	Push/Pull	Push/Pull	Push/Pull	Push/Pull
TE	512 kHz Inverted	0	0	1	Push/Pull	Push/Pull	Push/Pull	Push/Pull	Push/Pull	Push/Pull	Push/Pull
TE	IOM Interface	0	1	0	Push/Pull	Push/Pull	Push/Pull	Push/Pull	Push/Pull		Push/Pull
TMD	4096 kHz Inverted	0	1	1			Push/Pull				Open Drain
TMD	IOM Interface	0	1	1			Push/Pull				Push/Pull
NT	IOM Interface	1	1	1						Open Drain*	Open Drain*
SLMD	4096 kHz Inverted	1	0	0							Open Drain
SLMD	IOM Interface	1	1	0							Push-Pull

* With Integral Pull-Up Resistor

ค.2.1 วงจรเชื่อมโยงกับจุดอ้างอิง S การเชื่อมโยงกับจุดอ้างอิง S ของ SBC จะเป็นไปตามมาตรฐาน CCITT I.430 โดยวงจรส่งออกผ่านทางขา SX1 และ SX2 อาจจะต่อกับหม้อแปลงซึ่งอัตราส่วนของหม้อแปลง (n = 2:1) สามารถกำหนดได้โดยตัวต้านทานที่ขา Rref และวงจรรับข้อมูลเข้าผ่านทางขา SY1 และ SY2 ก็จะต่อจากหม้อแปลง (n = 2:1) เช่นเดียวกัน โดยที่ขา SY1 ต้องต่อตัวเก็บประจุขนาด 10 nF ด้วย

ค.2.2 วงจรเชื่อมโยง IOM วงจรเชื่อมโยง IOM ของ SBC ประกอบด้วยสัญญาณ 4 เส้นด้วยกันคือสัญญาณนาฬิกา DCL FSC และสัญญาณข้อมูลเข้าและออกที่อัตราเร็ว 256 kbps ภายในเฟรมข้อมูลที่ส่งผ่านวงจรเชื่อมโยงนี้ ประกอบด้วยข้อมูลในช่องสัญญาณ B1 และ B2 ช่องสัญญาณสำหรับการตรวจสอบและช่องสัญญาณ B* ซึ่งจะมีข้อมูลที่เป็นของช่องสัญญาณ C/I จำนวน 4 บิต ใช้สำหรับให้ SBC สามารถติดต่อกับ CPU โดยผ่าน ICC ได้

การใช้งานสัญญาณนาฬิกา DCL และ FSC นั้น ถ้าใช้ภายในอุปกรณ์ TE จะเป็นขาสัญญาณออก และใช้ในอุปกรณ์ TMD NT และ SMLD จะเป็นขาสัญญาณเข้า

ค.2.3 <u>การต่อสัญญาณเส้นอื่น ๆ</u> SBC นั้น มีขาสัญญาณเข้าและออกบางสัญญาณที่มีความหมายขึ้นอยู่กับการกำหนดโมดการทำงานของ SBC ดังได้แสดงไว้ในตารางที่ ค.1 โดยจะมีความหมายดังนี้

ENCK : Asynchronous wake Line ใช้สำหรับขอให้ SBC ส่งสัญญาณนาฬิกาให้กับอุปกรณ์ชั้นที่ 2 ที่ต่ออยู่ เช่นเดียวกับการส่งสัญญาณระดับ "0" เข้าที่ขา SDI

BUS : S Bus Configuration (Point-to-point หรือ Bus)

ECHO : Synchronous to SDO คือบิต Echo ซึ่งอยู่ในเฟรมที่จุดอ้างอิง S บิตที่ 24 และ 25

RDY : สัญญาณชิกแนลลิงสำหรับแจ้งสถานะของช่องสัญญาณ Dให้อุปกรณ์ชั้นที่ 2 ทราบ

CON : สำหรับเป็นการบอกว่าอุปกรณ์ที่ใช้ SBC ตัวนี้ต่ออยู่กับจุดอ้างอิง S ในลักษณะการต่อแบบบัส (จุดต่อหลายจุด)

TM2 :

DEX :

DE :

TS (2:0) : Time Slot Setting: Time Slot = 4TS2+2TS1+TS0

SSP : สัญญาณพัลซ์ที่ส่งออกที่ความถี่ต่าง ๆ เพื่อการทดสอบ

7680 kHz :

3840 kHz :

2560 kHz :

1280 kHz :

1536 kHz : สัญญาณนาฬิกานี้จะ Synchronize กับสัญญาณนาฬิกาที่จุดอ้างอิง S

512 kHz :

ค.3 <u>ฟังก์ชั่นในการส่ง</u>

SBC จะทำการส่งข้อมูลระหว่างวงจรเชื่อมโยง IOM กับวงจรเชื่อมโยงที่ต่อออกภายนอกโดยสัญญาณนาฬิกาที่ใช้ในการส่งข้อมูลถ้าใช้ SBC ใน NT หรือ SLMD จะได้รับมาจากวงจร

เชื่อมโยง IOM และถ้าใช้ใน TE หรือ TMD สัญญาณนาฬิกาที่ใช้จะรับมาจากวงจรเชื่อมโยงที่จุดอ้างอิง S และข้อมูลที่ส่งในช่องสัญญาณ B ที่จุดอ้างอิง S SBC จะส่งโดยไม่สนใจข้อมูล ขณะที่ SBC อยู่ในสถานะ Active (ถ้าใช้ใน TE หรือ TMD ข้อมูลจะส่งออกได้เมื่อมีการส่งคำสั่ง AR) ส่วนการส่งข้อมูลที่วงจรเชื่อมโยง IOM นั้น SBC จะส่งตลอดเวลาโดยไม่สนใจข้อมูลที่ทำการส่ง

ค.3.1 <u>การใช้งาน SBC ใน TE</u> SBC จะอาศัย PLL ภายในสร้างสัญญาณนาฬิกาความถี่ 192 kHz และ 512 kHz จากสัญญาณข้อมูลอัตราเร็ว 192 kbps ที่ได้รับมาจากจุดอ้างอิง S และจะส่งข้อมูลออกไปด้วยอัตราเร็ว 192 kbps โดยทิ้งระยะห่างกับเฟรมข้อมูลที่ได้รับ 2 บิตและขณะเดียวกัน SBC ก็จะส่งสัญญาณนาฬิกาความถี่ 512 และ 8 kHz ออกไปที่ขา DCL และ FSC ตามลำดับ

ค.4 <u>ฟังก์ชั่นการควบคุมและขั้นตอนการทำงานต่าง ๆ</u>

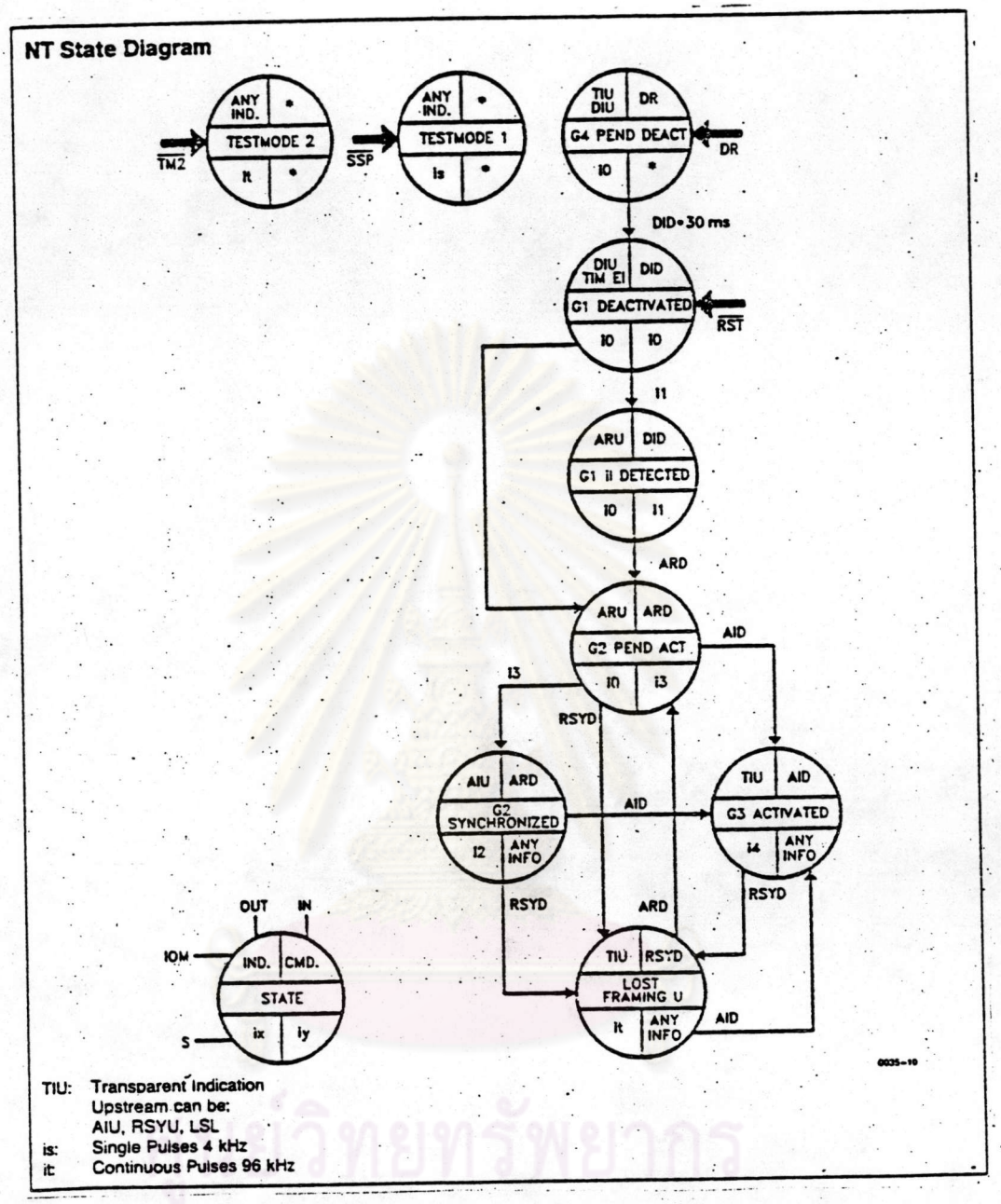
วงจรควบคุมภายใน SBC จะเป็นส่วนที่ทำขบวนการ Activate/Deactivate และฟังก์ชั่นการทดสอบต่าง ๆ ซึ่งจะเป็นไปตามมาตรฐาน CCITT I.430 โดยการติดต่อสื่อสารจะทำในช่องสัญญาณ C/I ของวงจรเชื่อมโยง IOM และผ่านทางวงจรเชื่อมโยงที่จุดอ้างอิง S

ค.4.1 <u>การใช้งาน SBC ใน TE และ TMD</u>

ค.4.1.1 <u>ขบวนการ Activate/Deactivate</u> การควบคุมการทำขบวนการ Activate/Deactivate ของ SBC ที่จุดอ้างอิง S จะเป็นไปตามมาตรฐาน CCITT I.430 โดยสามารถแสดงสถานะ และคำสั่งที่เกี่ยวข้องได้ในรูปที่ ค.2 และตารางที่ ค.2 ตามลำดับ

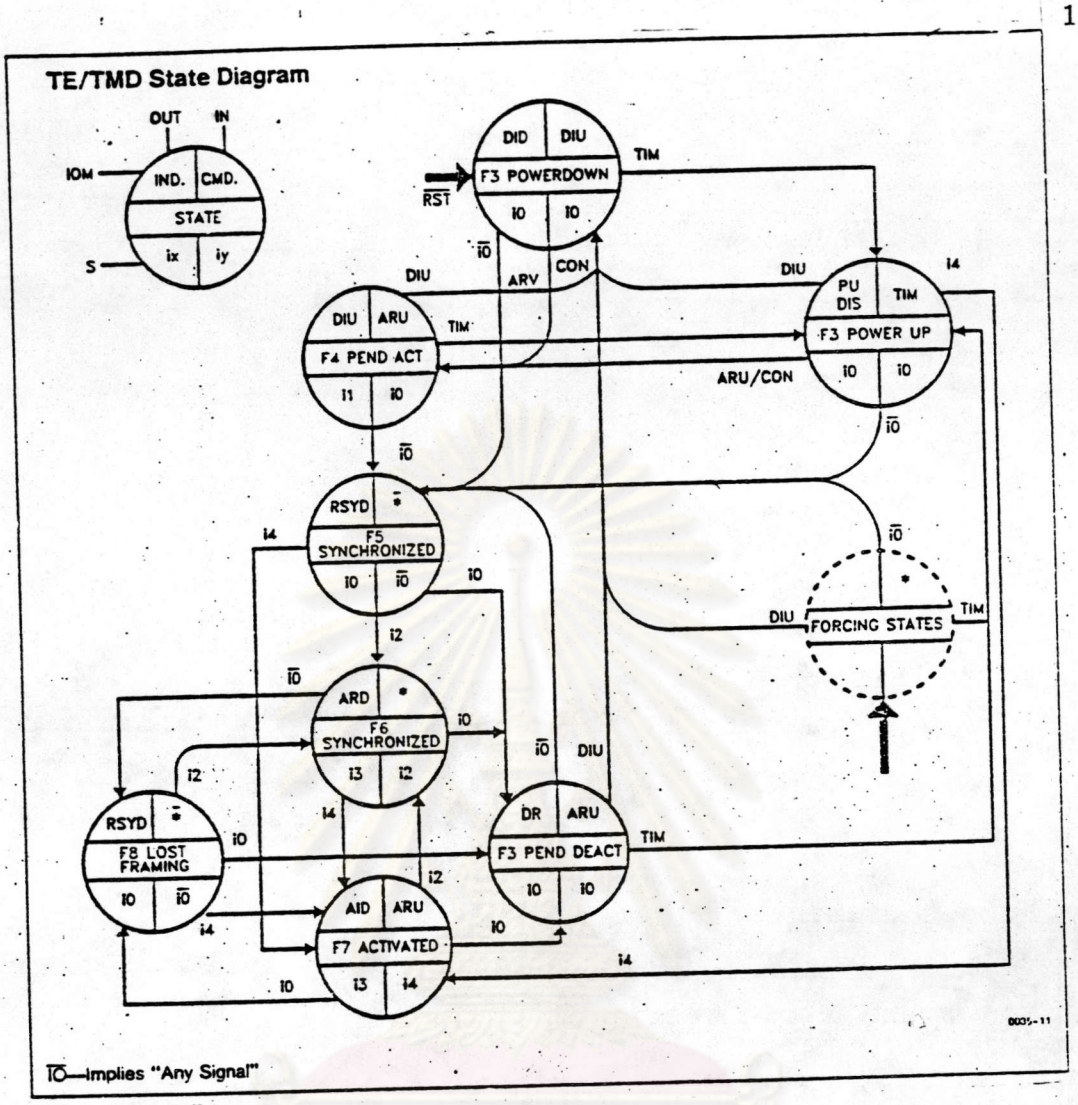
สำหรับการใช้งานใน TE แล้วสัญญาณนาฬิกาที่ SBC เป็นผู้สร้างนั้นสามารถสั่งได้โดยอาศัยคำสั่งจาก CPU ผ่านอุปกรณ์ชั้นที่ 2 เมื่อ SBC อยู่ในสถานะ F3 และได้รับคำสั่ง Power Down SBC จะหยุดส่งสัญญาณนาฬิกาออกไปหลังจากที่เวลาผ่านไปช่วงหนึ่ง และ SBC จะสร้างสัญญานาฬิกาอีกครั้งเมื่อ SBC ได้รับสัญญาณระดับ "0" ที่ขา SDI หรือ ขา ENCK ที่ขา X3

ค.4.1.2 <u>การทดสอบ Loop3</u> การทดสอบ Loop3 ทำได้โดยคำสั่ง Activate Request Loop และเมื่อสามารถ Synchronize ลูปได้แล้ว SBC จะส่งข่าวสาร Test indication(TI) ออกไป ลูปที่ทำการทดสอบนี้จะคงอยู่จนกว่าจะมีการสั่งเปลี่ยนแปลงผ่านทางคำสั่งในช่องสัญญาณ C/I เมื่ออยู่ในสถานะนี้ SBC จะไม่ส่งสัญญาณใด ๆ ออกที่วงจรเชื่อมโยงที่จุดอ้างอิง S แสดงได้ในรูปที่ ค.3



รูปที่ ค.2 แสดงไดอะแกรมสถานะของขบวนการ Activate/Deactivate

ค.4.1.3 การควบคุมการเข้าใช้ช่องสัญญาณ การควบคุมการเข้าใช้ช่องสัญญาณจะเป็นการป้องกันการรบกวนซึ่งกันและกันในการใช้ช่องสัญญาณ D ของอุปกรณ์ TE ที่ต่ออยู่บนบัสเดียวกัน เมื่อ TE ส่งข้อมูลไปในบิต D และเมื่อ NT หรือ SLMD ซึ่งอยู่ที่ปลายทางอีกข้างหนึ่งของบัสได้รับจะส่งกลับมาให้ TE เป็นข้อมูลถัดไปของบิต E TE ที่เป็นผู้ส่งข้อมูลจะทราบว่าข้อมูลที่ส่งไปแล้วนั้น NT หรือ SLMD ได้รับหรือไม่โดยการเอาข้อมูลในบิต E ที่ได้รับเปรียบเทียบกับข้อมูลที่ได้ส่งไปแล้วในบิต D ถ้าปรากฏว่าข้อมูลบิต E ไม่เท่ากับข้อมูลบิต D แสดงว่าเทอร์มินัลตัวนั้นไม่สามารถเข้าใช้ช่องสัญญาณ D ได้ จะต้องส่งข้อมูลที่มีค่า "1" ออกไป



รูปที่ ค.2 แสดงไดอะแกรมสถานะของขบวนการ Activate/Deactivate (ต่อ)

และจะสามารถขอเข้าใช้ช่องสัญญาณ D ได้ใหม่ก็ต่อเมื่อได้รับข้อมูลค่า "1" ติดต่อกันเป็นจำนวน X บิต ซึ่งในเวลานั้นเทอร์มินัลตัวนี้จะสามารถขอเข้าใช้ช่องสัญญาณ D ได้ใหม่ โดยการเริ่มส่งแฟล็ก HDLC(01111110) ค่า X จะมีค่าระหว่าง 8 หรือ 10 ขึ้นอยู่กับชั้นของลำดับความสำคัญที่เลือกใช้ และเมื่อเทอร์มินัลใด ๆ ก็ตามสามารถส่งข้อมูลออกไปได้ทั้งเฟรม ค่า X สำหรับเทอร์มินัลนั้นก็จะเพิ่มขึ้นอีก 1 และเมื่อเทอร์มินัลได้รับข้อมูลค่า "1" ในบิต E ติดต่อกันจำนวน 9 หรือ 10 บิต ค่า X สำหรับเทอร์มินัลนั้นก็จะถูก Reset ให้กลับไปเป็นค่าเดิมอีกครั้งหนึ่ง

การทำงานตามฟังก์ชันนี้ จะเป็นการทำงานร่วมกันระหว่าง SBC กับ ICC โดย SBC จะทำหน้าที่ตรวจสอบข้อมูลในบิต E และแจ้งสถานะของช่องสัญญาณ D ในขณะนั้นให้ ICC ทราบการติดต่อระหว่าง SBC กับ ICC นี้มีตัวแปรที่เกี่ยวข้องคือ

P : Priority (8 หรือ 10) ส่งไปพร้อมกับคำสั่ง ARP

C : จำนวนบิต E ที่มีค่าเป็น "1" ติดต่อกัน

ตารางที่ ค.2 แสดงคำสั่งที่ใช้ในขบวนการ Activate/Deactivate

NT Instructions and Messages

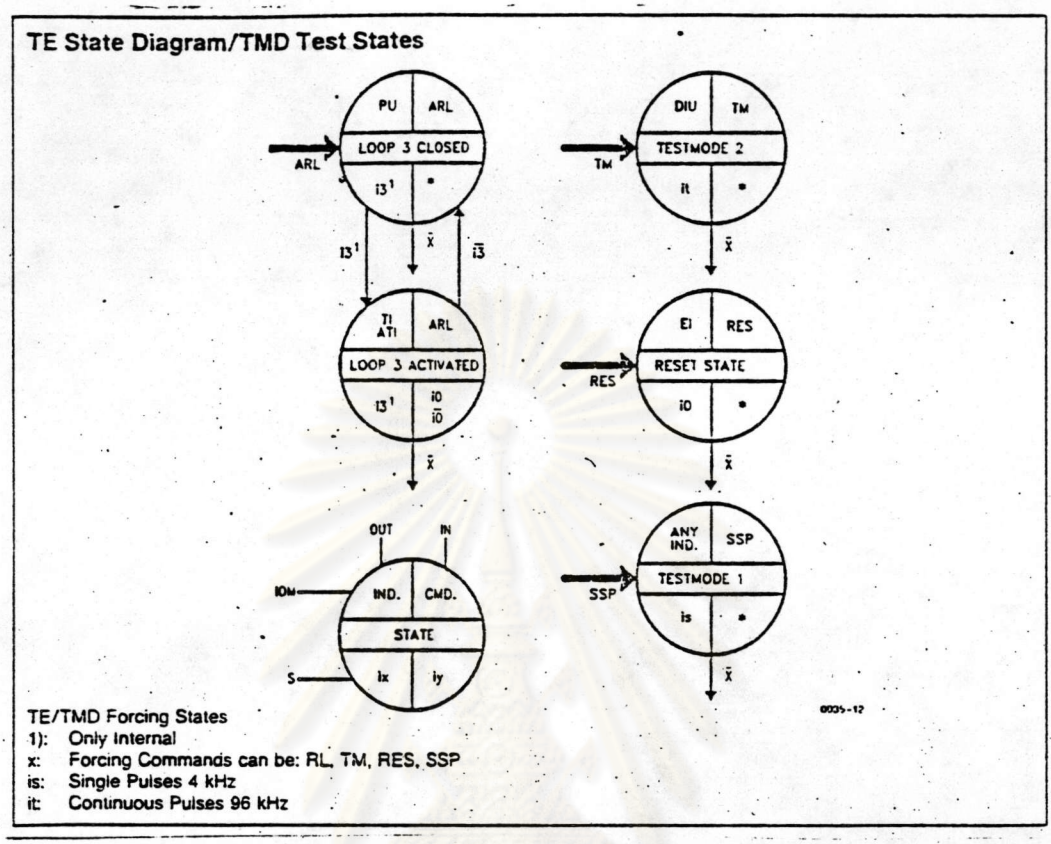
Instructions Downstream	Abb	Code	Remarks
Deactivate Request	DR	000x	Deactivation Instruction*
Send Single Pulses	SSp	0010	Transmission of Single Pulses*
Resynchronization	RSYD	0100	Transmission of Continuous Pulses
Activate Request	ARN	1000	Transmission of Info 2
Activate Request Loop	ARL	1010	Transmission of Info 2, Switch Test Loop 2
Activate Indication	AI	1100	Transmission of Info 4
Activate Indication Loop	AIL	1110	Transmission of Info 4, Switch Test Loop 2
Messages Upstream			
Timing	RIM	0000	SBC Requires Clocks
Lost Signal Level	LSL	0001	No Received Level
Resynchronizing	RSYU	0100	Receiver Not Synchronous
Error Indication	EI	0110	$\overline{RST}$ and $\overline{TM2}$
Active Request	ARU	1000	Info 1 Received
Active Indication	AIU	1100	Receiver Synchronous
Deactivate Indication	DIU	1111	Time TS 6 or TS 7 after Deactivation Instruction is Finished

*Unconditional Instructions

TE/TMD Instructions and Messages

Instruction Upstream	Abb	Code	Remarks
Timing	TIM	0000	Module Interface is Clocked
Reset	RS	0001	Reset (x)
Send Single Pulses	SSP	0010	Transmission of Single Pulses (x)
Test Mode	TM	0100	Transmission of Continuous Pulses
Activate Request P8	AR 8	1000	Activation Instruction, D Channel Priority is 8
Activate Request P 10	AR 10	1001	Activation Instruction, D Channel Priority is 10
Activate Request Loop	ARL	1010	Activation of Test Loop 3 (x)
Deactivate Indication	DIU	1111	Module Interface can be Shut Down
Messages Downstream			
Power Up	PU	0111	Module Interface is Clocked
Deactivate Request	DR	0000	Deactivation Request Form S
Slip Detected	SD	0010	Wander is Greater than 18 μs Peak-Peak
Disconnected	DIS	0011	PIN CON is at Ground
Error Indication	EI	0110	$\overline{RST}$ or Reset
Resynchronizing	RSYD	0100	Level Received, Not Synchronous
Activate Request	ARD	1000	Info 2 Received
Test Indication	TI	1010	Test Loop or Test Mode
Awake Test Indication	ATI	1011	Level Received During Test Loop
Activate Indication P8	AI 8	1100	Info 4 Received, D Channel Priority is 8 or 9
Activate Indication P10	AI 10	1101	Info 4 Received, D Channel Priority is 10 or 11
Deactivate Indication	DID	1111	Clocks are Switched Off, Quiescent State

(x): Unconditional Instructions



รูปที่ ค.3 แสดงไดอะแกรมสถานะสำหรับการทดสอบ

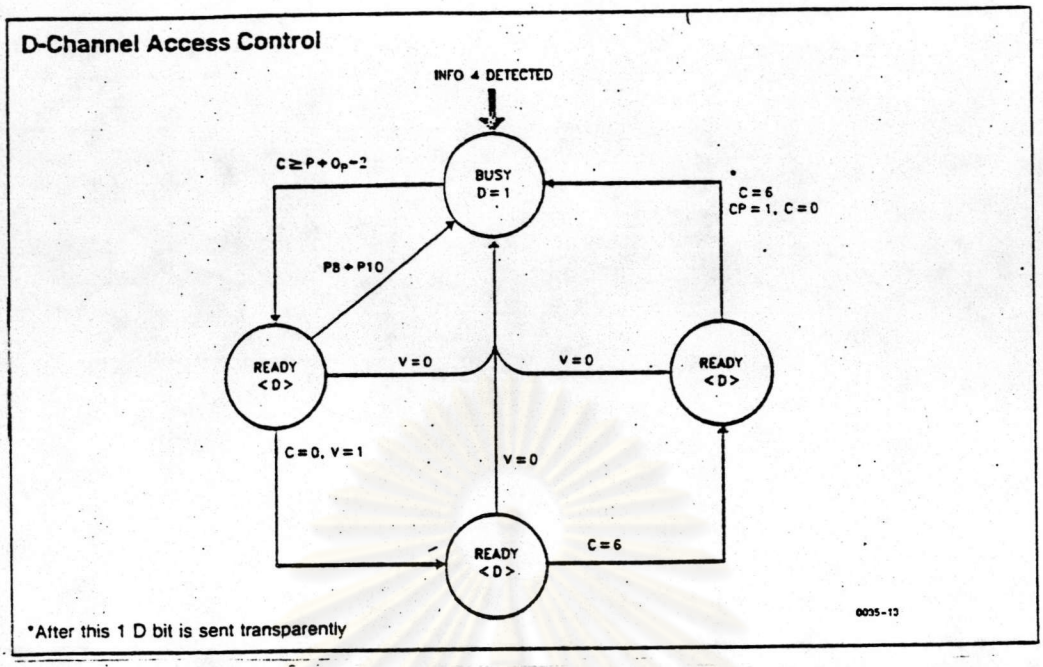
V : ผลการเปรียบเทียบระหว่างบิต D กับบิต E

O_8 : Priority ลดลงเป็น Priority 8

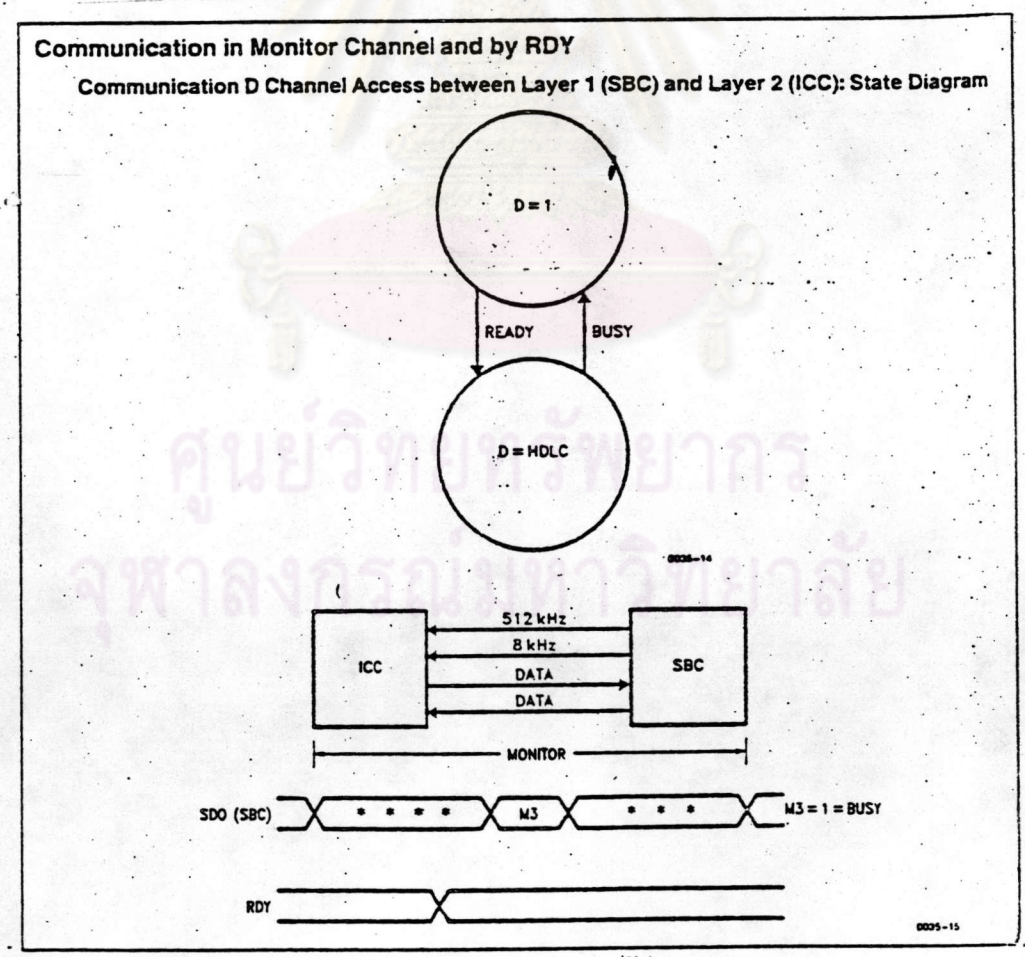
O_{10}: Priority ลดลงเป็น Priority 10

ซึ่งสามารถอธิบายได้เป็นไดอะแกรมดังแสดงในรูปที่ ค.4 และเมื่อติดต่อผ่านทาง ICC สามารถทำได้ดังแสดงในรูปที่ ค.5

ค.4.2.4 สถานะการ Reset และทดสอบ เมื่อได้รับการ Reset SBC จะเข้าสู่สถานะ F3 (Deactivated) ซึ่งในสถานะนี้ SBC จะไม่สร้างสัญญาณนาฬิการออกไปและ ถ้าการ Reset ทำพร้อม ๆ กับที่สัญญาณที่ขา ENCK เป็น "0" SBC จะสร้างสัญญาณนาฬิกาและส่งออกไปที่ขา DCL และ FSC ขณะเดียวกันก็จะส่งข่าวสาร ErrorIndication(EI) ไปในช่องสัญญาณ C/I ด้วย และ SBC จะสามารถเข้าสู่สถานะ F3 นี้ได้เช่นกันโดยใช้คำสั่งReset ผ่านทางช่องสัญญาณ C/I



รูปที่ ค.4 แสดงไดอะแกรมการเข้าใช้ช่องสัญญาณ D



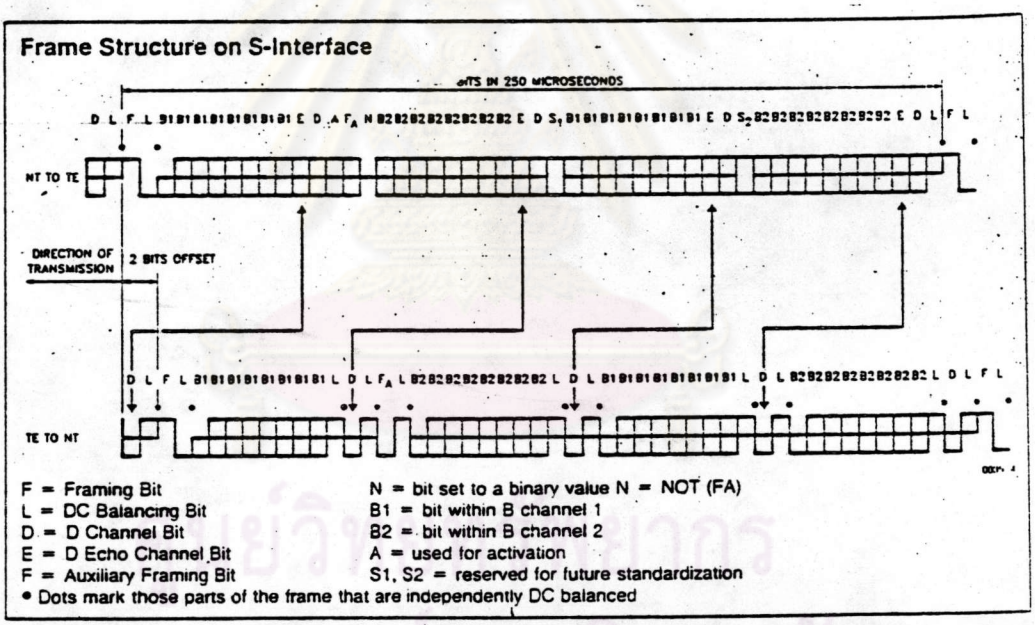
รูปที่ ค.5 แสดงไดอะแกรมการเข้าใช้ช่องสัญญาณ D ผ่านทาง ICC

ถ้าส่งคำสั่ง Test Mode ออกไป SBC จะส่งสัญญาณความถี่ 96 kHz ออกไปทางวงจรเชื่อมโยงจุดอ้างอิง S อย่างต่อเนื่อง

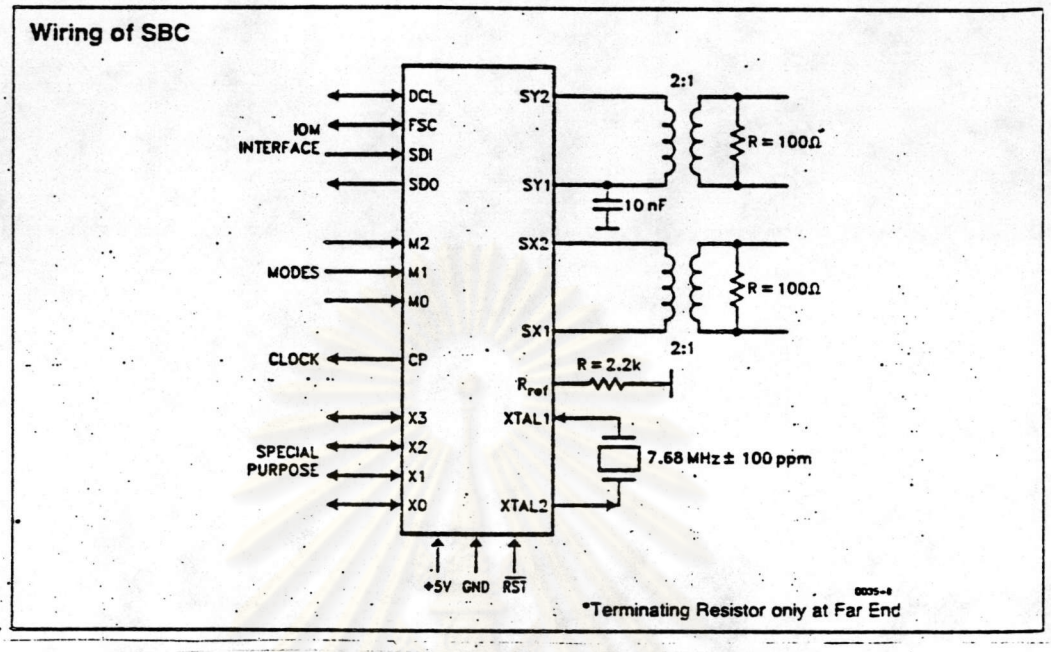
ถ้าส่งคำสั่ง Send Single Pulse ออกไป SBC จะส่งสัญญาณพัลซ์เดี่ยวขนาดความถี่ 4 kHz ออกไป

ค.5 <u>การเชื่อมโยงที่จุดอ้างอิง S (S-interface)</u>

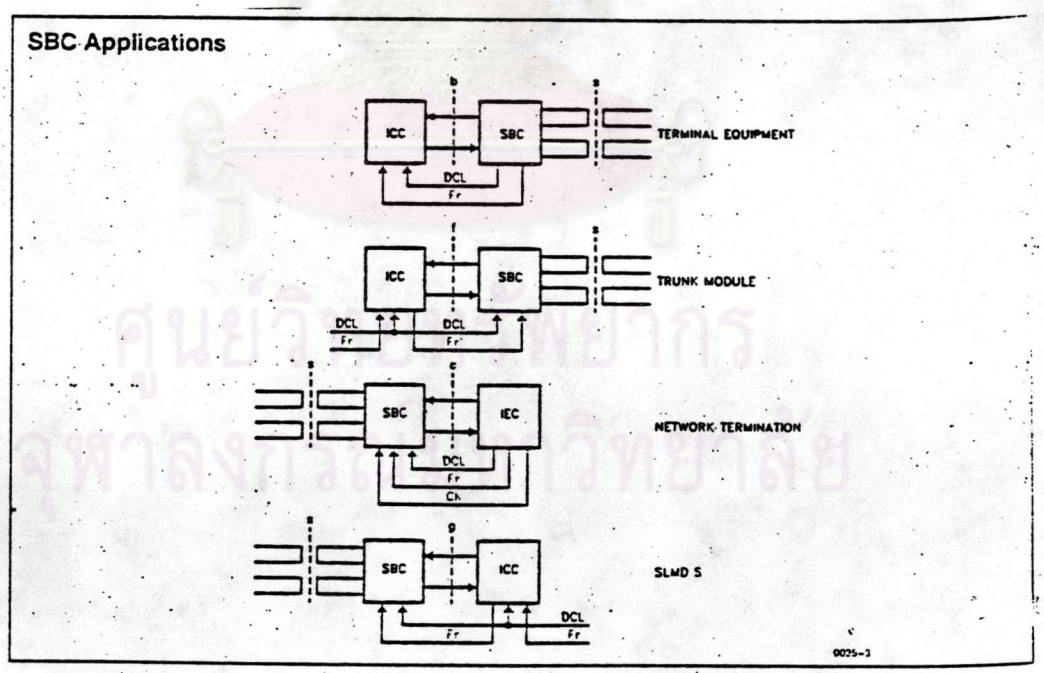
ข้อมูลที่ SBC ทำการรับส่งกับจุดอ้างอิง S จะเป็นไปตามมาตรฐานที่กำหนดโดย CCITT คือเป็นเฟรมข้อมูลขนาด 48 บิตภายในคาบเวลา 250 ไมโครวินาทีดังแสดงในรูปที่ ค.6 และการใช้งาน SBC ที่จุดอ้างอิง S สามารถนำไปใช้ได้ดังแสดงในรูปที่ ค.7 และ ค.8



<u>รูปที่ ค.6 แสดงเฟรมข้อมูลที่ส่งออกที่จุดอ้างอิง S</u>



รูปที่ ค.7 แสดงการต่อ SBC สำหรับใช้งานที่จุดอ้างอิง S



รูปที่ ค.8 แสดงการใช้งาน SBC ในอุปกรณ์ต่าง ๆ

ภาคผนวก ง.

โครงสร้างของซอฟต์แวร์

ง.1 ลำดับการให้บริการสัญญาณขัดจังหวะจาก HSCC

การจัดลำดับการตรวจสอบสาเหตุการเกิดสัญญาณขัดจังหวะสำหรับ HSCC ได้จัดไว้ตามลำดับดังนี้

1. Receive Pool Full(RPF)
2. Receive Message End(RME)
3. Receive Frame Overflow(RFO)
4. Timer Interrupt(TIN)
5. CTS Transition(CSC)
6. Receive Status Change(RSC)
7. Protocol Error(PCE)
8. Transmit Data Underrun(XDU)
9. Transmit Pool Ready(XPR)
10. Transmit Message Repeat(XMR)

ง. ลำดับการให้บริการสัญญาณขัดจังหวะจาก ICC

การจัดลำดับการตรวจสอบสาเหตุการเกิดสัญญาณขัดจังหวะสำหรับ ICC ได้จัดไว้ตามลำดับดังนี้

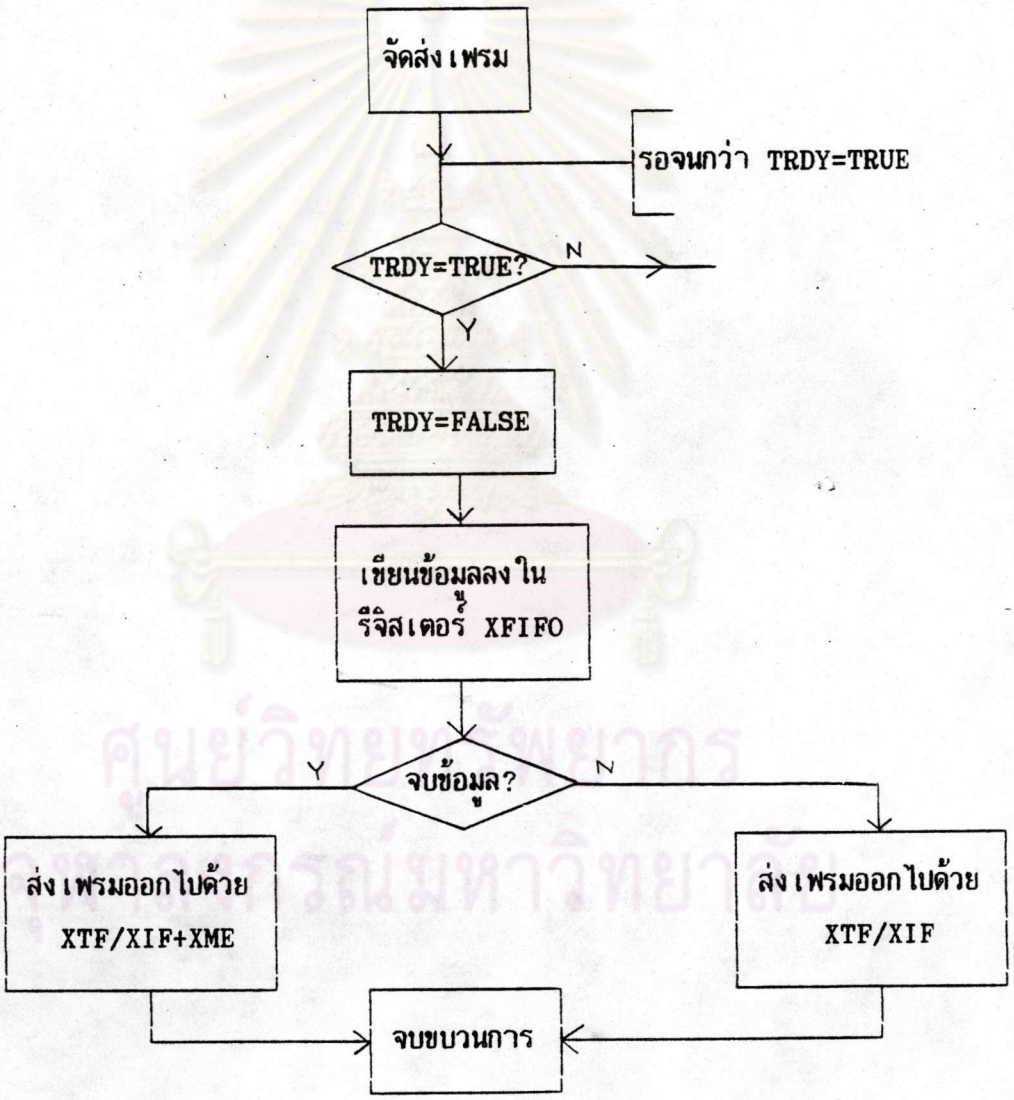
1. Command/Indicate Code Change(CIC)
2. Receive Pool Full(RPF)
3. Receive Message End(RME)
4. Receive Frame Overflow(RFO)
5. Timer Interrupt(TIN)
6. Receive Status Change(RSC)
7. Protocol Error(PCE)
8. Monitor Byte Receive(MOR)
9. Synchronous Transfer Interrupt(SIN)
10. Synchronous Transfer Overflow(SOV)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

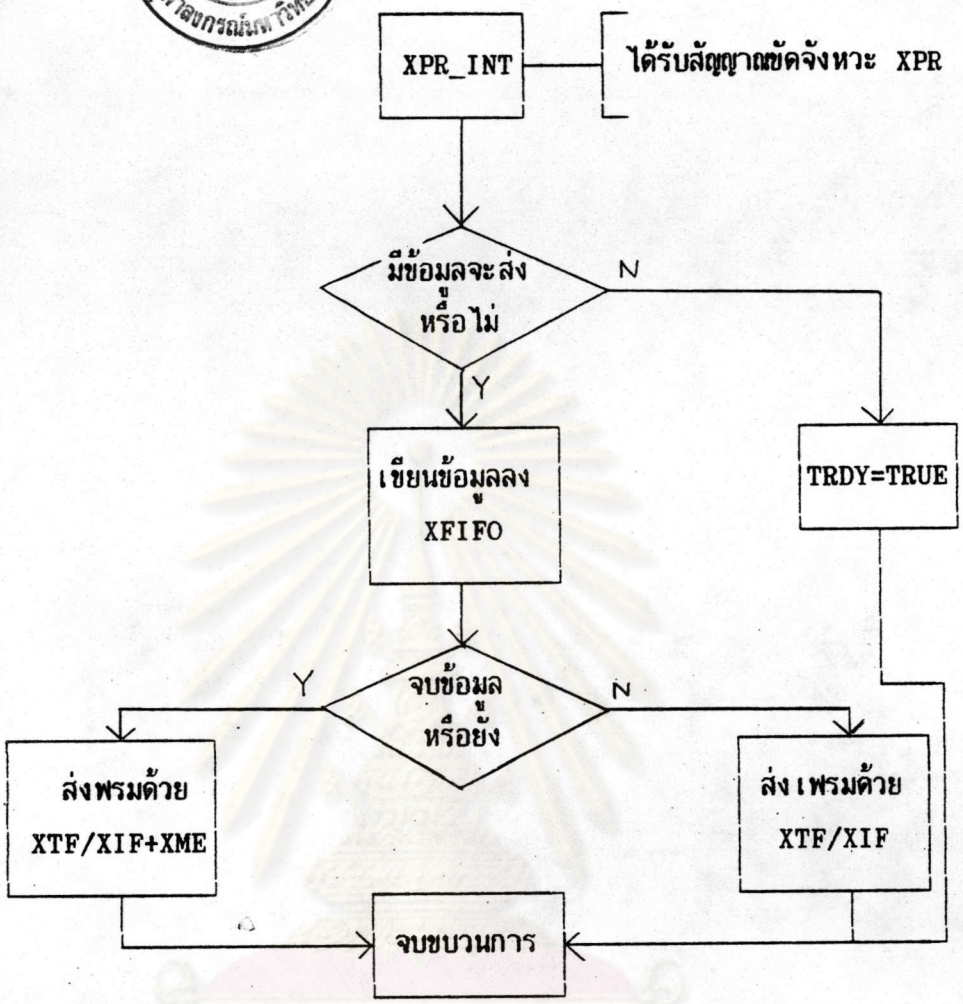
11. Transmit Data Underrun(XDU)
12. Transmit Message Repeat(XMR)
13. Transmit Pool Ready(XPR)

ง.3 ผังงานของโปรแกรมการส่งและรับข้อมูล

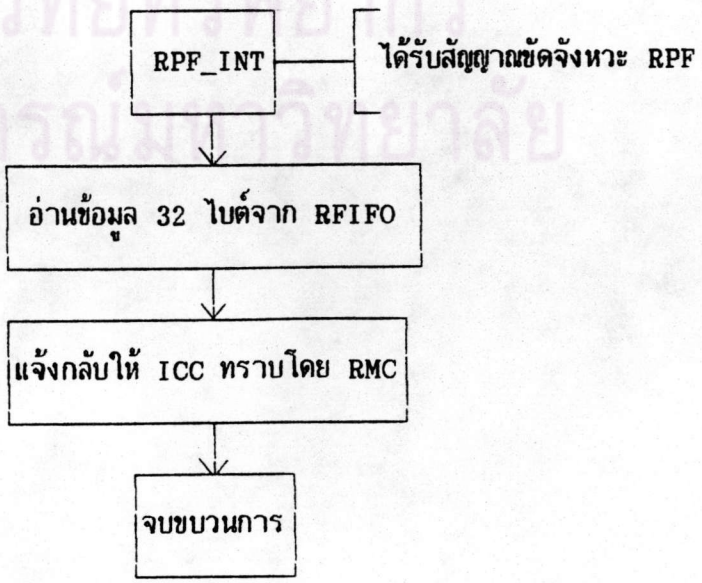
ง.3.1 โปรแกรมด้านส่ง

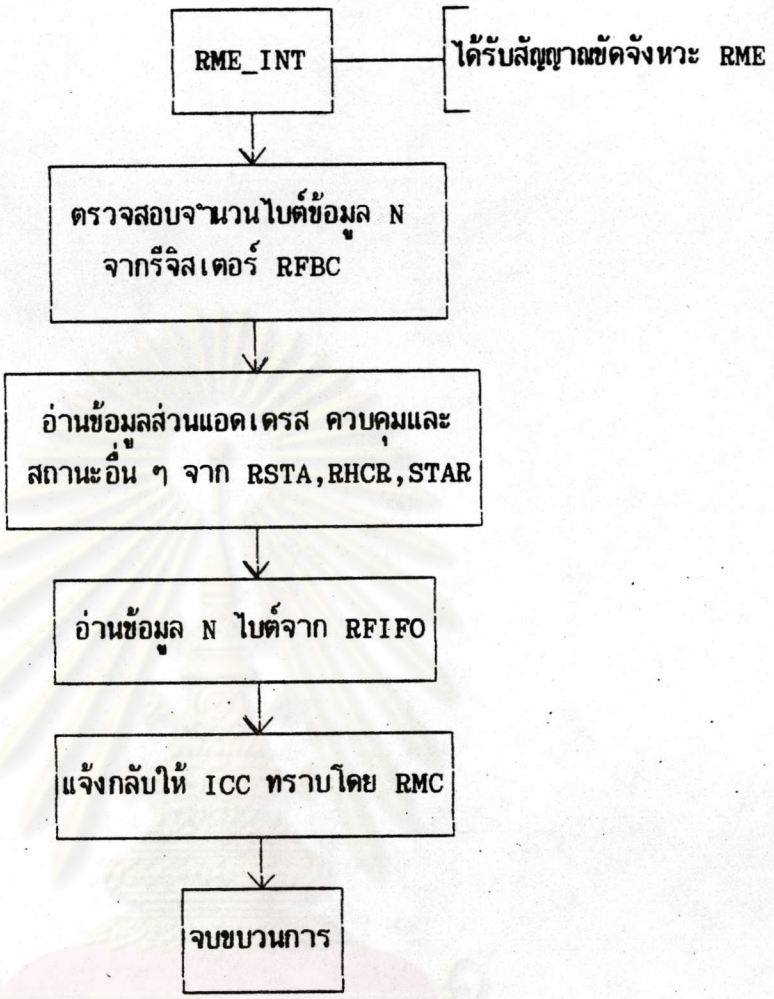


TRDY เป็นตัวแปรสำหรับบอกสถานะของบัฟเฟอร์ภายใน ICC ว่าพร้อมที่จะรับข้อมูลหรือไม่ โดยการเปลี่ยนแปลงสถานะได้จากการตรวจสอบบิต XFW ในรีจิสเตอร์ STAR หรือเมื่อได้รับสัญญาณขัดจังหวะ XPR



ง.3.2 <u>โปรแกรมด้านรับ</u>





หมายเหตุ สำหรับสัญญาณการส่งเฟรมและการได้รับข้อมูลโดย HSCC จะมีขั้นตอนเหมือนกันกับวิธีการของ ICC ยกเว้นขั้นตอนก่อนและหลังขบวนการเหล่านี้เท่านั้นที่ต่างกัน

ง.4 ฟังก์ชันต่างๆ ของซอฟต์แวร์

ซอฟต์แวร์ที่ได้พัฒนาขึ้นมานั้นได้กระทำในลักษณะของฟังก์ชันย่อย ๆ ที่ทำงานเป็นอิสระจากกัน ทำให้สามารถเปลี่ยนแปลงขั้นตอนการทำงานภายในของแต่ละฟังก์ชันได้ง่าย และสำหรับซอฟต์แวร์ที่ทำงานบนแผงวงจรนั้น จะประกอบไปด้วยฟังก์ชันย่อย ๆ ต่อไปนี้

ฟังก์ชัน	หน้าที่
BDATAI	- แจ้งการได้รับข้อมูลจากช่องสัญญาณ B
BDATAR	- ร้องขอให้มีการส่งข้อมูลออกไปในช่องสัญญาณ B
BXMTED	- แจ้งการจบข้อมูลที่จะส่งออกไปในช่องสัญญาณ B
CH_CMDR	- ตรวจสอบการทำงานของรีจิสเตอร์ CMDR ของ HSCC
CHADDR	- ตรวจสอบข้อมูลส่วนแอดเดรสของเฟรมที่ได้รับ

พังก์ชัน	หน้าที่
CHCTRL	- ตรวจสอบข้อมูลส่วนควบคุมของเฟรมที่ได้รับ
CIC_INT	- เปลี่ยนแปลงสถานะของตัวแปรเก็บสถานะของวงจรชั้นที่ 1 เมื่อเกิดการเปลี่ยนแปลง
CKL2ST	- ตรวจสอบและแก้ไขค่าของตัวแปรแสดงสถานะของวงจรข้อมูล
CLREXP	- ลบล้างตัวแปรเก็บสถานะความผิดปกติต่าง ๆ และกำหนดตัวแปรของระบบสำหรับวงจรข้อมูลนั้นๆ ให้เป็นค่าเริ่มต้นคือ '0'
CLRFRC	- ลบล้างตัวแปรเก็บสถานะของการเกิดเหตุการณ์ Frame rejection
DISIQ	- ยกเลิกข่าวสารที่รอการส่งออกไปด้วยเฟรม I
DLDATAI	- แจ้งการได้รับเฟรม I
DLDATAR	- ร้องขอให้มีการส่งข่าวสารออกไปโดยเฟรม I
DLESTI,DLESTC	- แจ้งผลการทำขบวนการเชื่อมต่อวงจร้อมูล
DLESTR	- ร้องขอให้ทำขบวนการเชื่อมต่อวงจรข้อมูล
DLRELI,DLRELC	- แจ้งผลการทำขบวนการปลดวงจร้อมูล
DLRELR	- ร้องขอให้ทำขบวนการปลดวงจรข้อมูล
DLUIRQ	- ร้องขอให้มีการส่งข่าวสารโดยเฟรม UI
DXMTED	- แจ้งการจบข้อมูลที่จะส่งออกไปในช่องสัญญาณ D
ENQRSP	- ตอบสนองการสอบถามสถานะจากคู่สนทนา
ESDLK	- ส่งเฟรม SABM เพื่อทำขบวนการเชื่อมต่อวงจรข้อมูล
FRTYPE	- จำแนกชนิดของเฟรมระหว่างเฟรม I และเฟรม T
FRVALID	- ตรวจสอบความถูกต้องของเฟรมที่ได้รับ
GETLST	- อ่านข้อมูลสำหรับเฟรม I ที่ได้ส่งออกไปแล้วมาใหม่เพื่อจัดการส่งออกไปใหม่
HQUP	- เตรียมข้อมูลเพื่อส่งออกไปโดย HSCC
HSCCINIT	- กำหนดการเริ่มต้นทำงานให้กับ HSCC
ICCINIT	- กำหนดการเริ่มต้นทำงานให้กับ ICC
IFRQUP	- เตรียมข้อมูลเพื่อส่งเฟรม I ออกไป
IVRETX.	- ส่งเฟรม I ที่ได้ส่งออกไปแล้วใหม่
L1FAIL	- แจ้งผลความล้มเหลวที่เกิดขึ้นจากการทำขบวนการ Activate
L3INIT	- เก็บสถานะเมื่อมีการสั่งงานชั้นที่ 2 โดยชั้นที่ 3
MDLERR	- แจ้งผลความผิดพลาดในระดับชั้นที่ 2 ให้ Management entity ทราบ

พังก์ชัน	หน้าที่
NRERR	- ฟื้นฟูความผิดพลาดที่เกิดขึ้นเนื่องจากเฟรมที่ได้รับมีค่า N(R) ไม่ถูกต้อง
PCRXEDB	- รับทราบการอ่านข้อมูลออกจากหน่วยความจำสำหรับช่องสัญญาณ B โดย PC เรียบร้อยแล้ว
PCRXEDD	- รับทราบการอ่านข้อมูลออกจากหน่วยความจำสำหรับช่องสัญญาณ D โดย PC เรียบร้อยแล้ว
PUTHQ	- อ่านข้อมูลที่จะส่งออกไปในช่องสัญญาณ B จากหน่วยความจำบนแผงวงจรและบันทึกเก็บไว้ในรีจิสเตอร์ XFIFO ของ HSCC
PUTIQ	- อ่านข้อมูลที่จะส่งออกไปในส่วนข่าวสารของเฟรม I จากหน่วยความจำภายในโปรแกรมและบันทึกเก็บไว้ในรีจิสเตอร์ XFIFO ของ ICC
RD_BLK	- อ่านข้อมูลจากรีจิสเตอร์ภายในของ HSCC หรือ ICC มาเก็บไว้ในหน่วยความจำภายในโปรแกรม
REQ_ACT	- Activate วงจรในระดับโปรโตคอลชั้นที่ 1
REQFRIC	- โปรแกรมหลักของการให้บริการ ICC
REQFRPC	- โปรแกรมหลักของการให้บริการ PC
RQFRHSC	- โปรแกรมหลักของการให้บริการ HSCC
RSPDM	- โต้ตอบการได้รับเฟรม DM
RSPDSC	- โต้ตอบการได้รับเฟรม DISC
RSPFRMR	- โต้ตอบการได้รับเฟรม FRMR
RSPIFR	- โต้ตอบการได้รับเฟรม I
RSPRXFR	- ตรวจสอบชนิดของเฟรมที่ได้รับและสั่งงานการโต้ตอบเฟรมชนิดนั้น ๆ
RSPSABM	- โต้ตอบการได้รับเฟรม SABM
RSPUA	- โต้ตอบการได้รับเฟรม UA
RSPUI	- โต้ตอบการได้รับเฟรม UI
RXBPC	- รับข้อมูลจาก PC เพื่อจะส่งออกในช่องสัญญาณ B
RXDPC	- รับข้อมูลจาก PC เพื่อจะส่งออกในช่องสัญญาณ D
SDFRMR	- ส่งเฟรม FRMR
SDLST4	- เปลี่ยนค่าตัวแปรเก็บสถานะของวงจรข้อมูลเป็นสถานะ 4
SDLST5	- เปลี่ยนค่าตัวแปรเก็บสถานะของวงจรข้อมูลเป็นสถานะ 5
SDLST6	- เปลี่ยนค่าตัวแปรเก็บสถานะของวงจรข้อมูลเป็นสถานะ 6

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

พังก์ชัน	หน้าที่
SDLST7	- เปลี่ยนค่าตัวแปรเก็บสถานะของวงจรข้อมูลเป็นสถานะ 7
SDLST8	- เปลี่ยนค่าตัวแปรเก็บสถานะของวงจรข้อมูลเป็นสถานะ 8
SENDFR	- ส่งเฟรม LAPD ที่เป็นเฟรมคำสั่ง
SENDIFR	- ส่งเฟรม I ออกไป
SEV_8088	- โปรแกรมใช้ประจำบริการสำหรับสัญญาณขัดจังหวะที่เกิดจาก PC
SEV_HSC	- โปรแกรมใช้ประจำบริการสำหรับสัญญาณขัดจังหวะที่เกิดจาก HSCC และทำหน้าที่เรียกใช้รูทีนย่อยอื่นๆ ที่เกี่ยวข้องต่อไป ซึ่งจะประกอบไปด้วยพังก์ชันต่อไปนี้คือ
	HRPF_INT เมื่อเกิดเหตุการณ์ RPF
	HRME_INT เมื่อเกิดเหตุการณ์ RME
	HRFO_INT เมื่อเกิดเหตุการณ์ RFO
	HTIN_INT เมื่อเกิดเหตุการณ์ TIN
	HRSC_INT เมื่อเกิดเหตุการณ์ RSC
	HPCE_INT เมื่อเกิดเหตุการณ์ PCE
	HXDU_INT เมื่อเกิดเหตุการณ์ XDU
	HXMR_INT เมื่อเกิดเหตุการณ์ XMR
	HXPR_INT เมื่อเกิดเหตุการณ์ XPR
SEV_ICC	- โปรแกรมใช้ประจำบริการสำหรับสัญญาณขัดจังหวะที่เกิดจาก ICC และทำหน้าที่เรียกใช้รูทีนย่อยอื่น ๆ ที่เกี่ยวข้องต่อไป ซึ่งจะประกอบไปด้วยพังก์ชันต่อไปนี้คือ
	IRPF_INT เมื่อเกิดเหตุการณ์ RPF
	IRME_INT เมื่อเกิดเหตุการณ์ RME
	IRFO_INT เมื่อเกิดเหตุการณ์ RFO
	ITIN_INT เมื่อเกิดเหตุการณ์ TIN
	IRSC_INT เมื่อเกิดเหตุการณ์ RSC
	IPCE_INT เมื่อเกิดเหตุการณ์ PCE
	IXDU_INT เมื่อเกิดเหตุการณ์ XDU
	IXMR_INT เมื่อเกิดเหตุการณ์ XMR
	IXPR_INT เมื่อเกิดเหตุการณ์ XPR
	MOR_INT เมื่อเกิดเหตุการณ์ MOR
	SIN_INT เมื่อเกิดเหตุการณ์ SIN
	SOV_INT เมื่อเกิดเหตุการณ์ SOV

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

พังก์ชัน	หน้าที่
STOPTMR	- สั่งให้ตัวจับเวลาหยุดทำงาน
STTIMR	- สั่งงานให้ตัวจับเวลาภายใน ICC ซึ่งทำหน้าที่เป็นตัวจับเวลา T200 เมื่อทำงานในโมด External เริ่มจับเวลา
SWOFF	- สั่งให้ ICC และ HSCC กลับไปอยู่ในโมด Power down
SWON	- สั่งงานพังก์ชันกำหนดการเริ่มต้นทำงานให้กับ ICC และ HSCC
T200EXP	- สั่งงานแผงวงจรเมื่อหมดเวลา T200
TXENQ	- สอบถามสถานะของคู่สนทนาโดยอาศัยเฟรม RNR และ RR
UIFRQUP	- เตรียมข้อมูลเพื่อส่งเฟรม UI ออกไป
WR_BLK	- บันทึกข้อมูลจากหน่วยความจำภายในโปรแกรมลงในรีจิสเตอร์ของ HSCC หรือ ICC
WRADDR	- เตรียมข้อมูลส่วนแอดเดรสสำหรับเฟรมที่จะส่งออกไป
WRCTRL	- เตรียมข้อมูลส่วนควบคุมสำหรับเฟรมที่จะส่งออกไป
XMIT	- สั่งงานให้ ICC จัดการส่งเฟรมที่ได้สร้างเสร็จเรียบร้อยแล้วออกไป

และสำหรับโปรแกรมที่ทำงานบน PC นั้นจะประกอบไปด้วยพังก์ชันต่าง ๆ ดังต่อไปนี้

พังก์ชัน	หน้าที่
call_exit()	- ทำหน้าที่หยุดการทำงานของโปรแกรมบน PC
clear_io_port()	- กำหนดค่าเริ่มต้นให้พอร์ท DPIO จาก 8088
create_file()	- เปิดแฟ้มข้อมูลเพื่อเขียนข้อมูลที่ได้รับจากช่องสัญญาณ B
directory()	- อ่านรายชื่อแฟ้มข้อมูลบนแผ่นจานแม่เหล็กและนำมาแสดงผลที่จอแสดงผล
file_sent()	- แจ้งให้ผู้ใช้และแผงวงจรทราบเมื่อได้ส่งข้อมูลจากแฟ้มข้อมูลจบเรียบร้อย
file_transfer()	- ส่งข้อมูลจากแฟ้มข้อมูลของ PC ไปในช่องสัญญาณ B
inddisc()	- แจ้งให้ผู้ใช้ทราบเมื่อได้รับแจ้งผลการทำขบวนการปลดวงจรข้อมูล
indlink()	- แจ้งให้ผู้ใช้ทราบเมื่อได้รับแจ้งผลการทำขบวนการเชื่อมต่อวงจรข้อมูล

ฟังก์ชัน	หน้าที่
interrupt isdn_service()	- ตรวจสอบสถานะที่ทำให้เกิดสัญญาณขัดจังหวะจากแผงวงจร และจัดการเก็บสถานะตัวแปรที่เกี่ยวข้องเพื่อการสั่งงานต่อไป
key_service()	- ตรวจสอบคำสั่งที่ได้รับจากคีย์บอร์ด
main()	- โปรแกรมหลักของโปรแกรมที่ทำงานบน PC
menu()	- แสดงรายการหลักของโปรแกรมบน PC
msgsnt()	- แจ้งให้ผู้ใช้และแผงวงจรทราบเมื่อได้ส่งข่าวสารจบ
open_file()	- เปิดแฟ้มข้อมูลที่จะส่งออกไปโดยช่องสัญญาณ B
read_file()	- อ่านข้อมูลจากแฟ้มข้อมูลเพื่อส่งออกในช่องสัญญาณ B
rec_power_off()	- แจ้งให้ผู้ใช้ทราบเมื่อได้รับแจ้งผลการกลับเข้าสู่โมด Power down
rec_power_on()	- แจ้งให้ผู้ใช้ทราบเมื่อได้รับแจ้งผลการเริ่มต้นการทำงานของแผงวงจร
receiveb()	- ได้รับข้อมูลจากช่องสัญญาณ B
recmsg()	- แสดงข่าวสารที่ได้รับจากช่องสัญญาณ Dให้ผู้ใช้รับทราบ
req_power_off()	- สั่งให้แผงวงจรเข้าสู่การทำงานในโมด Power down
req_power_on()	- สั่งให้แผงวงจรเข้าสู่การทำงานในโมด Power up
reqdisc()	- สั่งให้แผงวงจรเริ่มทำขบวนการปลดวงจรข้อมูล
reqlink()	- สั่งให้แผงวงจรเริ่มทำขบวนการเชื่อมต่อวงจรข้อมูล
reset_sys_card()	- สั่งให้แผงวงจรเริ่มต้นการทำงาน
screen_panel()	- แสดงจอภาพที่บ่งบอกว่าโปรแกรมบน PC กำลังทำงาน
signalling()	- ส่งข้อมูลให้แผงวงจรเพื่อส่งออกไปในช่องสัญญาณ D
vect_install()	- ติดตั้งเวกเตอร์สำหรับสัญญาณขัดจังหวะที่ใช้
write_file()	- บันทึกข้อมูลที่ได้รับจากช่องสัญญาณ B

U8'
U17
U1
U31
U42
U39
U58
U36
U54
U25
U43
U32
U2
U3
U4
U53
U59
U60
U7
U8
U18
U51
U29
U28
U44
U35
U33
U34
U55
U45
DIP-1
U6
U7
U19
U21
U20
U24
U22
U23
U52
U13
U12
U11
U10
U9
U46
U41
U38
U37
U57
U14
U56
U25
U30
U27
U26
U50
U15
U49
U16

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียน

นายบัญญัติ พัดทอง เกิดเมื่อวันที่ 16 พฤศจิกายน 2504 ที่จังหวัดบุรีรัมย์ ได้รับปริญญาวิศวกรรมศาสตร์บัณฑิต สาขาวิศวกรรมไฟฟ้า จากมหาวิทยาลัยขอนแก่น เมื่อปี 2525 และเข้าศึกษาต่อปริญญามหาบัณฑิต ที่ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัยในปี พ.ศ. 2528 ในระหว่างศึกษาได้ทำงานในตำแหน่งวิศวกรระดับห้า สังกัดแผนกอุปกรณ์คอมพิวเตอร์สื่อสาร กองสื่อสารข้อมูลคอมพิวเตอร์ ฝ่ายวางแผนระบบไฟฟ้า การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย