



รายละเอียดทั่วไปของเครื่องนี้แอด -2200/200

เพื่อให้เข้าใจระบบการแปลโปรแกรมภาษาฟอร์แทรนของเครื่องนี้แอด -2200/200 ได้ดียิ่งขึ้น จึงขอกล่าวำถึงลักษณะแฟ้มข้อมูล (FILE) ของเทปตัวแปลโปรแกรม (BINARY RUN TAPE) ลักษณะเรคคอร์ดของออบเจคโปรแกรม ลักษณะของเทปบูทสแตรปรูทีน (TAPE BOOTSTRAP ROUTINE) และโหลดเดอรัมมิเตอร์ (LOADER MONITOR) โดยสังเขปเสียก่อนที่จะเริ่มกล่าวถึงระบบการแปลโปรแกรมภาษาฟอร์แทรนของเครื่องนี้แอด -2200/200 อย่างละเอียดต่อไป

2.1 ลักษณะแฟ้มข้อมูลของเทปตัวแปลโปรแกรม

เทปตัวแปลโปรแกรมเป็นแฟ้มข้อมูลประเภทหนึ่งเรคคอร์ดต่อบล็อก และมีขนาดของข้อมูลไม่แน่นอน (VARIABLE UNBLOCK RECORD) ขนาดของข้อมูลที่มากที่สุดจะไม่เกิน 250 ตัวอักษรต่อบล็อก . เนื่องจากขีดจำกัดดังกล่าวนี้ จึงใช้ตัวอักษรตัวแรกเป็นตัวบอกลักษณะของเรคคอร์ด เรียกว่าแบนเนอร์คาร์แรกเตอร์ (BANNER CHARACTER) ซึ่งมีความหมายดังนี้ :-

- (22)₈ เรคคอร์ดแรกของเทปบูทสแตรปรูทีน
- (42)₈ เรคคอร์ดต่อ ๆ มาของเทปบูทสแตรปรูทีน
- (41)₈ ทูติยภูมิเรคคอร์ด (NON - HEADER RECORD) และไม่ใช่เรคคอร์ดสุดท้าย
- (44)₈ ทูติยภูมิเรคคอร์ดและเป็นเรคคอร์ดสุดท้าย
- (50)₈ ปฐมภูมิเรคคอร์ด (SEGMENT HEADER RECORD) และไม่ใช่เรคคอร์ดสุดท้าย
- (54)₈ ปฐมภูมิเรคคอร์ดและเป็นเรคคอร์ดสุดท้าย

เรคคอร์ดในเทปตัวแปลโปรแกรม สามารถแบ่งออกเป็น 6 ประเภทใหญ่ ๆ ดังนี้ :-

(รูปที่ 2.1)

2.1.1 เรคคอร์ดแรกของแฟ้มข้อมูล (THE HEADER LABEL RECORD)

2.1.2 เรคคอร์ดแรกของเทปบูทสทราปรูทีน (THE FIRST RECORD OF TAPE BOOTSTRAP ROUTINE)

2.1.3 เรคคอร์ดต่อมาของเทปบูทสทราปรูทีน (THE TAPE BOOTSTRAP ROUTINE RECORDS)

2.1.4 ออบเจคโปรแกรมเรคคอร์ด (OBJECT PROGRAM RECORDS)

2.1.5 เรคคอร์ดสุดท้ายของแฟ้มข้อมูล (THE TRAILER LABEL RECORD)

2.1.6 เรคคอร์ดสุดท้ายของทุกแฟ้มข้อมูล (THE END - OF - RESERVED - INFORMATION RECORDS)

2.2 ลักษณะเรคคอร์ดของออบเจคโปรแกรม ¹

ออบเจคโปรแกรมที่อยู่ในเทปตัวแปลโปรแกรม ประกอบด้วย 2 ส่วนใหญ่ ๆ ดังนี้ :-

2.2.1 ส่วนหัว (IDENTIFICATION AND CONTROL FIELD)

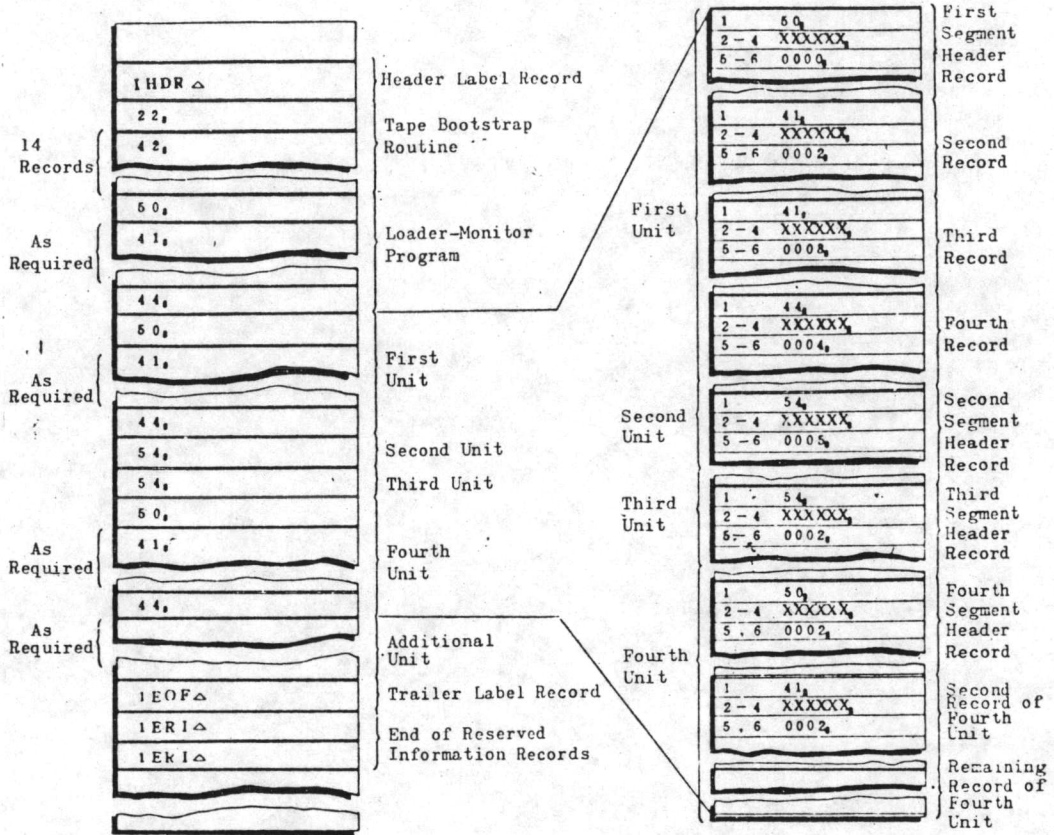
ส่วนนี้จะบอกลักษณะโดยทั่ว ๆ ไปของออบเจคโปรแกรมเรคคอร์ด ซึ่งมี 2 ประเภท คือ :-

2.2.1.1 ส่วนหัวปฐมภูมิ (SEGMENT HEADER RECORD) (รูปที่ 2.2) จะเป็นเรคคอร์ดแรกของโปรแกรมยูนิต ซึ่งส่วนหัวนี้将有ความยาว 24 ตัวอักษร และแบนเนอร์คาร์แรคเตอร์จะมีค่า $(50)_8$ หรือ $(54)_8$

2.2.1.2 ส่วนหัวทุติยภูมิ (NON - HEADER RECORD) (รูปที่ 2.3) จะเป็นเรคคอร์ดต่อ ๆ ไปของโปรแกรมยูนิต ซึ่งส่วนหัวนี้将有ความยาว 7 ตัวอักษร และแบนเนอร์คาร์แรคเตอร์จะมีค่า $(41)_8$ หรือ $(44)_8$

2.2.2 ส่วนข้อมูล (DATA FIELD)

ส่วนนี้จะอยู่ถัดจากส่วนหัว ประกอบด้วยกลุ่มข้อมูลซึ่งมีตัวคุม (DATA FIELD CONTROL CHARACTER) นำหน้าอยู่หนึ่งตัว (ตารางที่ 2.1) และตามด้วยกลุ่มตัวอักษร (CHARACTER STRING) โหลดเดอรัมอนิเตอร์จะใช้ตัวคุมนี้ในการแปลงโค้ดของออบเจคโปรแกรม (OBJECT PROGRAM) ไปเป็นโค้ดทางเครื่องคอมพิวเตอร์ (MACHINE CODE) เพื่อเก็บไว้ในส่วนความจำ (MAIN MEMORY)



BINARY RUN TAPE

ASSIGNMENT OF RECORD SEQUENCE NUMBERS FOR FOUR BRT UNITS SHOWN, ASSUMING FIRST UNIT HAS FOUR RECORDS.

รูปที่ 2.1 ลักษณะของ เทปตัวแปลโปรแกรม

Control character			Meaning
No.	Octal	Binary	
1	01 to 17	00nnnn	Interpret the nnnn digits as a binary number. Move following nnnn chars. to successive locations, placing leftmost char. in location specified by current setting of distribution counter (in X6), clear punctuation in locations into which the chars. are moved. Advance distribution counter by nnnn.
2	21 to 37	0lnnnn	Perform same function as control char. no. 1, and set word mark in leftmost char. location loaded.
3	41 to 57	10nnnn	Perform same functions as control char. no. 1, and set an item mark in leftmost char. location loaded.
4	60	110000	Place following 3 chars. into distribution counter. (The next string will be loaded with its leftmost char. at this address.)
5	61	110001	Terminate loading. Interpret the following 3 chars. as normal starting location for unit just loaded.
6	62	110010	Clear area of memory, using following 7 chars. to identify area to be cleared and char. with which to clear it. (Chars. 1 thru 3 are interpreted as lowest address of the area to be cleared; chars. 4 thru 6 are interpreted as highest address; and char. 7 is transferred to every location in the cleared areas with punctuation marks cleared.)
7	63	110011	Set an item mark in location whose address is one less than current setting of dis. counter.
8	64	110100	Set an item mark in the location whose address is 1 less than current value of dis. counter.
9	77	111111	Read the next record.

ตารางที่ 2.1 ตัวควบคุมในส่วนข้อมูลในเรคคอร์ดของออบเจคโปรแกรม

Data Field Control Characters.

2.3 เทปบูทสแตรปรูทีน (TAPE BOOTSTRAP ROUTINE)

เครื่องคอมพิวเตอร์นี้แอด -2200/200 มีลักษณะการออกแบบทางฮาร์ดแวร์ (HARDWARE DESIGN) ที่สำคัญดังนี้ :-

2.3.1 สามารถอ่านข้อมูลเข้ามาในส่วนของความจำได้ทีละเรคคอร์ด โดยกดปุ่มบูทสแตรปบนแผงควบคุม

2.3.2 คำสั่ง SET WORD - MARK SW (22)₈ สามารถทำงานได้โดยตัวเองไม่ต้องมีเวอร์คมาร์ค โดยปกติตำแหน่งเริ่มต้นของแต่ละคำสั่ง จะต้องมีการเวอร์คมาร์ค และสิ้นสุดคำสั่งหนึ่ง ๆ เมื่อพบอีกหนึ่งเวอร์คมาร์ค

เทปบูทสแตรปรูทีน เขียนขึ้นมาเพื่อเชื่อมความสัมพันธ์ระหว่างระบบฮาร์ดแวร์ (HARDWARE) กับระบบซอฟต์แวร์ (SOFTWARE) ในขณะเริ่มต้น กล่าวคือ เมื่อเริ่มต้นขึ้นมาเครื่องคอมพิวเตอร์ยังไม่สามารถทำงานได้ เพราะในส่วนของความจำยังไม่มีคำสั่งใด ๆ อยู่เลย เรคคอร์ดแรกของบูทสแตรปรูทีน จะถูกอ่านเข้าไปโดยการกดปุ่มบูทสแตรปบนแผงควบคุม เครื่องคอมพิวเตอร์จะทำงานตามคำสั่งที่เข้ามา ซึ่งก็คืออ่านเรคคอร์ดต่อมาอีก 14 เรคคอร์ดของบูทสแตรปรูทีน เพื่อที่จะดึงโหลดเดอรัมมิเตอร์เข้ามาไว้ในส่วนของความจำ หลังจากนั้นเครื่องคอมพิวเตอร์ก็จะทำงานต่อไปอย่างต่อเนื่อง

2.4 โหลดเดอรัมมิเตอร์ (LOADER MONITOR)¹

เป็นโปรแกรมควบคุมที่มีความสำคัญ และมีหน้าที่การทำงานที่สลับซับซ้อนมาก ลักษณะโดยสังเขปมีดังนี้ :-

2.4.1 ลักษณะโครงสร้าง

2.4.1.1 พื้นที่สื่อสาร (COMMUNICATION AREA) เป็นส่วนที่ใช้ติดต่อกับโปรแกรมอื่น ๆ

2.4.1.2 ส่วนที่เป็นคำสั่ง (LOADER INSTRUCTION AREA) เป็นคำสั่งต่าง ๆ ของโหลดเดอรัมมิเตอร์

2.4.1.3 ส่วนที่เป็นค่าคงที่และพื้นที่ใช้งาน (CONSTANTS AND WORKING AREA) เป็นค่าคงที่ต่าง ๆ และพื้นที่สำหรับเก็บข้อมูลที่อ่านจากเทปแม่เหล็ก

2.4.2 หน้าที

2.4.2.1 การหาโปรแกรมที่ต้องการ (SEARCHING) โดยสามารถทำได้ทั้งหาไปข้างหน้า (FORWARD SEARCH) และหาถอยหลัง (BACKWARD SEARCH) ลักษณะการหาโปรแกรมที่ต้องการขึ้นอยู่กับค่าของพารามิเตอร์ที่กำหนดในพื้นที่สื่อสาร ซึ่งสามารถแบ่งออกเป็น 5 ลักษณะ ดังนี้ :-

- ก. หาโดยชื่อโปรแกรมและเซกเมนต์ (SEARCH BY PROGRAM AND SEGMENT NAME)
- ข. หาโดยชื่อเซกเมนต์ ภายในโปรแกรมที่กำหนด (SEARCH BY SEGMENT NAME WITHIN A GIVEN PROGRAM)
- ค. หาโดยวิสัยปิลิตี้และตำแหน่งสัมพันธ์ (SEARCH BY VISIBILITY AND RELATION POSITION)
- ง. หาโดยชื่อโปรแกรม เซกเมนต์และวิสัยปิลิตี้ (SEARCH BY PROGRAM NAME, SEGMENT NAME AND VISIBILITY)
- จ. หาโดยชื่อเซกเมนต์และวิสัยปิลิตี้ ภายในโปรแกรมที่กำหนด (SEARCH BY SEGMENT NAME & VISIBILITY WITHIN A GIVEN PROGRAM)

2.4.2.2 การดึงโปรแกรมเข้ามาในส่วนของความจำ (LOADING) จะแปลงโค้ดของออบเจกต์โปรแกรมให้เป็นโค้ดของเครื่องคอมพิวเตอร์ แล้วนำไปไว้ในส่วนของความจำตามตำแหน่งที่กำหนดไว้

2.4.2.3 การเริ่มต้น (STARTING) เมื่อโปรแกรมยูนิตที่ต้องการได้เข้ามาอยู่ในส่วนของความจำเรียบร้อยแล้ว โหลดเคอร์มอมิเตอร์ จะส่งการควบคุมไปให้ตามค่าของพารามิเตอร์ที่กำหนดไว้ในพื้นที่สื่อสาร ซึ่งมีอยู่ 3 ลักษณะดังนี้ :-

- ก. การเริ่มต้นแบบธรรมดา (NORMAL START MODE) จะส่งการควบคุมไปให้โปรแกรมยูนิตที่เข้ามาตามตำแหน่งที่กำหนดไว้ใน NORMAL STARTING LOCATION
- ข. การเริ่มต้นแบบพิเศษ (SPECIAL START MODE) จะส่งการควบคุมไปให้โปรแกรมที่อยู่ในส่วนของความจำเรียบร้อยแล้ว ตามตำแหน่งที่

กำหนดไว้ใน SPECIAL STARTING LOCATION

- ค. การเริ่มต้นแบบกลับคืน (RETURN START MODE) จะส่งการควบคุมกลับไปให้โปรแกรม ก่อนที่จะมาโหลดเดอรัมมิเตอร์ ตามตำแหน่งที่ถูกเก็บไว้เมื่อเริ่มเข้ามาในโหลดเดอรัมมิเตอร์

2.5 ระบบการแปลโปรแกรมภาษาฟอร์แทรน

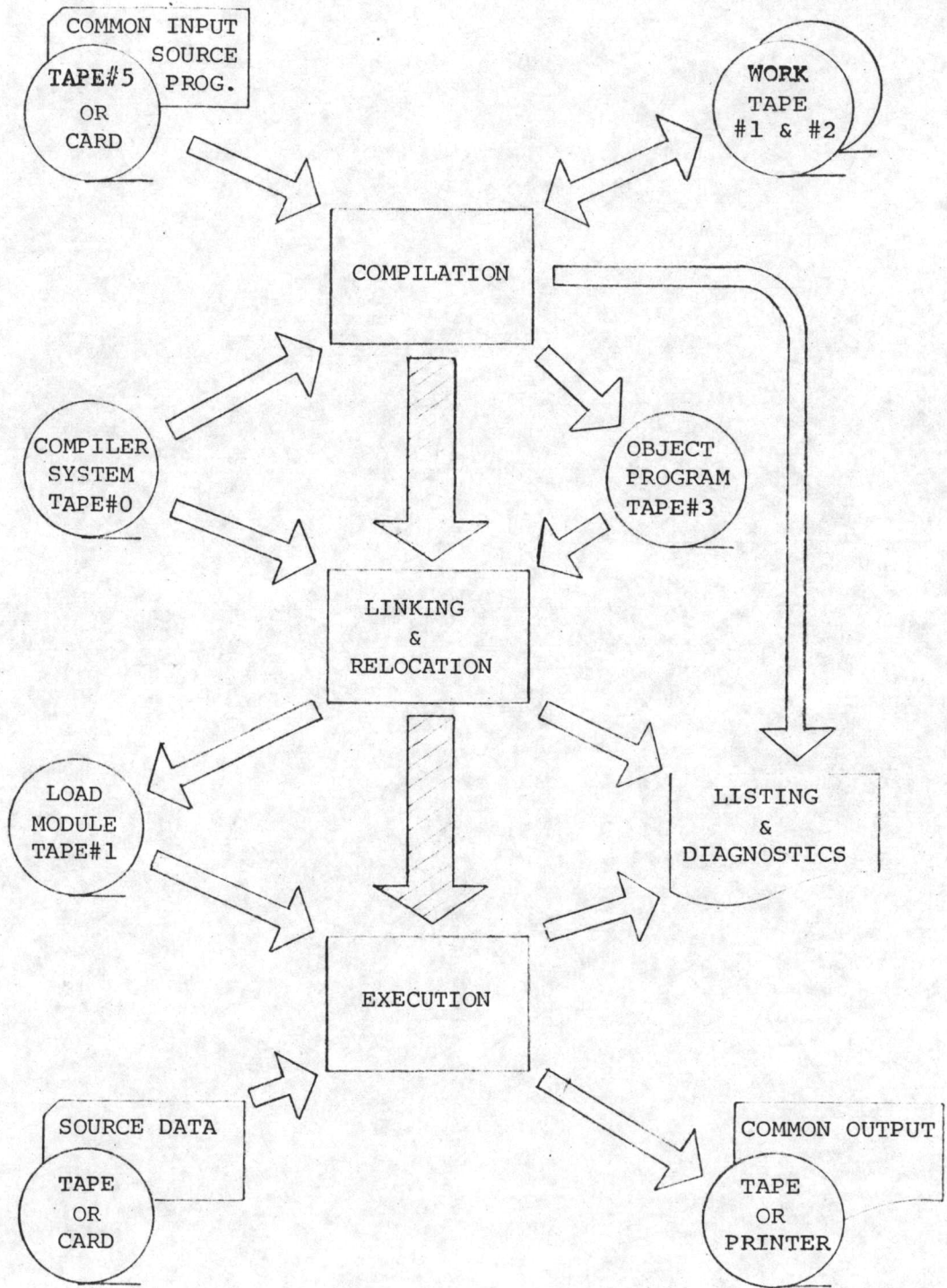
การแปลโปรแกรมภาษาฟอร์แทรนของเครื่องนี้แอดค -2200/200 ในระบบโหลดแอนด์โก สามารถแบ่งออกเป็น 3 ขั้นตอนใหญ่ ๆ ดังนี้ :- (รูปที่ 2.4)

2.5.1 การแปล (COMPILATION) จะแปลซอร์สโปรแกรม (SOURCE PROGRAM) ภาษาฟอร์แทรนไปเป็นรีโลเซตเทเบิ้ลออบเจคโปรแกรม (RELOCATABLE OBJECT PROGRAM) ออบเจคโปรแกรม ที่ได้นี้ยังไม่สามารถทำงานได้ เนื่องจากมีบางส่วนที่ไม่สามารถแปลได้ ได้แก่ ส่วนอ้างอิงภายนอกโปรแกรม (EXTERNAL REFERENCE)

ขั้นตอนในการแปลนี้ ยังแบ่งออกเป็นขั้นตอนย่อยอีก ² เช่น การจัดกลุ่มของคำในแต่ละกระทรวงความ (LEXICAL ANALYSIS PHASE) การตรวจสอบไวยากรณ์ของแต่ละกระทรวงความ (SYNTAX ANALYSIS PHASE) การแปลงโค้ด (INTERPRETATION PHASE) การเตรียมเนื้อที่ในส่วนความจำ (STORAGE ASSIGNMENT) เป็นต้น

2.5.2 การเชื่อมและรีโลเซต (LINKING AND RELOCATION) แปลส่วนอ้างอิงภายนอกโปรแกรมของแต่ละรูทีน นำรูทีนต่าง ๆ มาเชื่อมต่อกัน พร้อมกับจัดตำแหน่ง (ADDRESS) ต่าง ๆ ให้ถูกต้องเหมาะสม โปรแกรมใหญ่ที่ได้นี้สามารถทำงานได้ เรียกว่า โหลดโมดูล (LOAD MODULE)

2.5.3 การทำงานของโปรแกรม (EXECUTION) โหลดเดอรัมมิเตอร์จะดึงโหลดโมดูลเข้ามาไว้ในส่วนความจำ จากนั้นโปรแกรมจะเริ่มทำงานตามคำสั่งต่าง ๆ ที่เขียนไว้ จนกระทั่งจบโปรแกรม เป็นการสิ้นสุดการแปลโปรแกรมภาษาฟอร์แทรนหนึ่งโปรแกรม



รูปที่ 2.4 แผนภาพแสดงระบบการแปลโปรแกรมภาษาฟอร์แทรน