# BIBLIOGRAPHY


Knuth, D.E. The Art of Computer Programming vol.3, Sorting and Searching. Newyork: Addison-Wesley Publishing Co., 1973.

Newyork, IBM Programming Publications Department. "DOS/VS Access Method Service User's Guide" Bangkok: IBM, 1977. (Mimeographed)

____ . "OS and DOS/VS Assembler Language" Bangkok: IBM, 1973. (Mimeographed)

____ . "DOS/VS Data Management Guide" Bangkok: IBM, 1977. (Mimeographed)

Sorenson, Tremblay. An Introduction to Data Structures with Applications. Newyork: McGraw-Hill Book Co., 1976.

Sweden, IBM Nordic Laboratory. "DOS/VS Sort/Merge Logic" Bangkok: IBM, 1973. (Mimeographed)

____ , "DOS/VS Sort/Merge Installation Reference Manual" Bangkok: IBM, 1973. (Mimeographed)

____ . "DOS/VS System Management Guide" Bangkok: IBM, 1973. (Mimeographed)

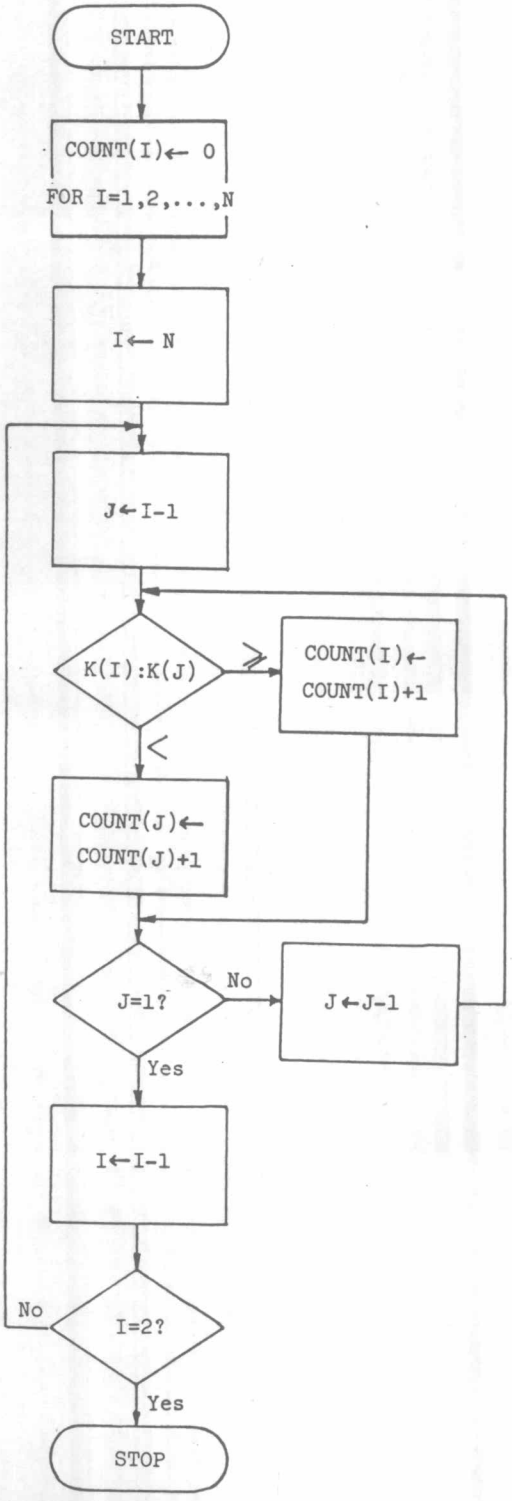
____ . "DOS/VS Sort/Merge Programmer's Guide" Bangkok: IBM, 1973. (Mimeographed)

____ . "DOS/VS System Generation" Bangkok: IBM, 1973. (Mimeographed)

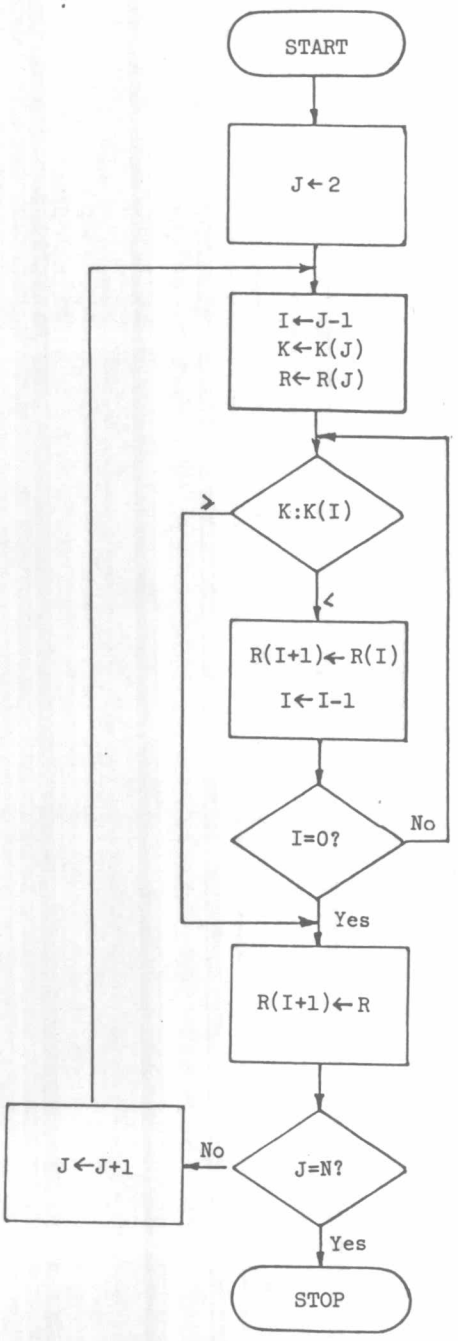
____ . "DOS/VS Supervisor and I/O Macros" Bangkok: IBM, 1973. (Mimeographed)

ภาคผนวก ก.

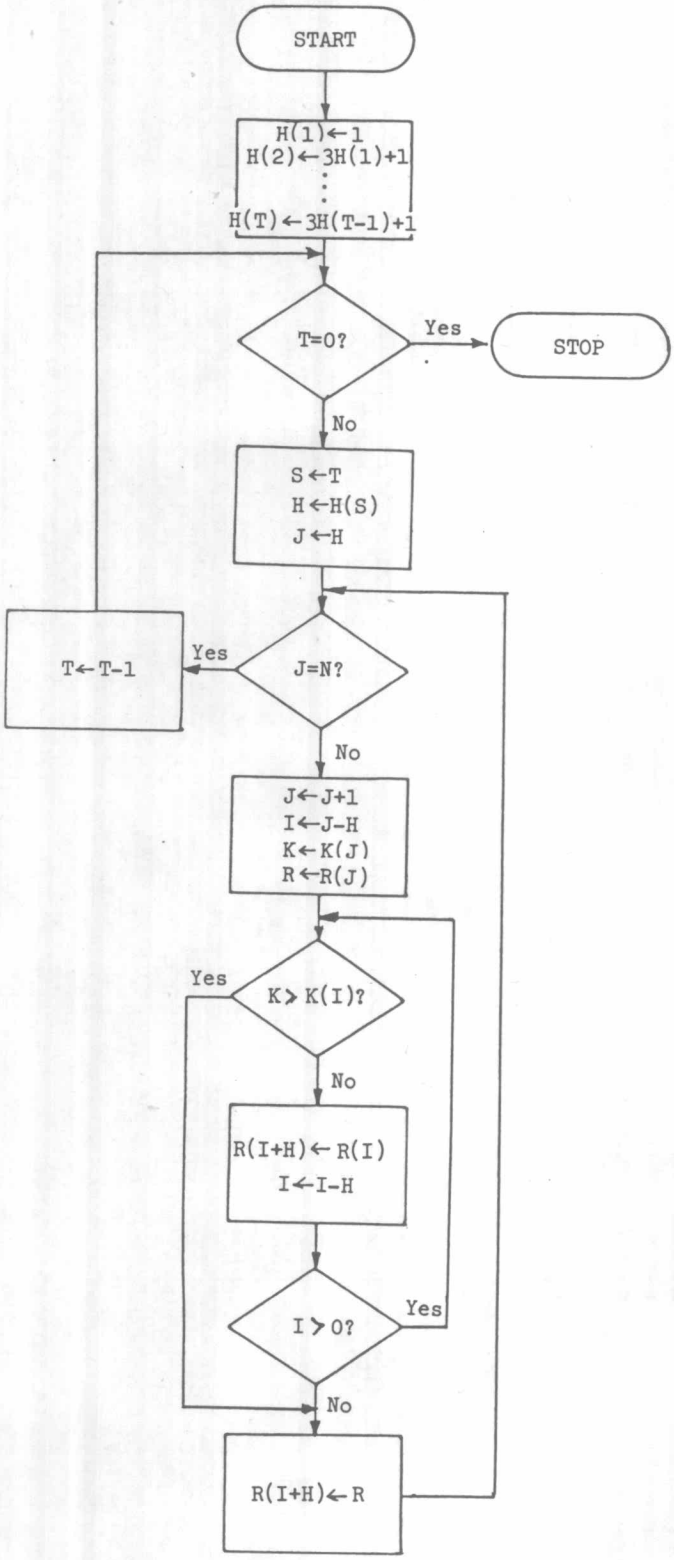
ผังงานแสดงวิธีการเรียงลำดับข้อมูล



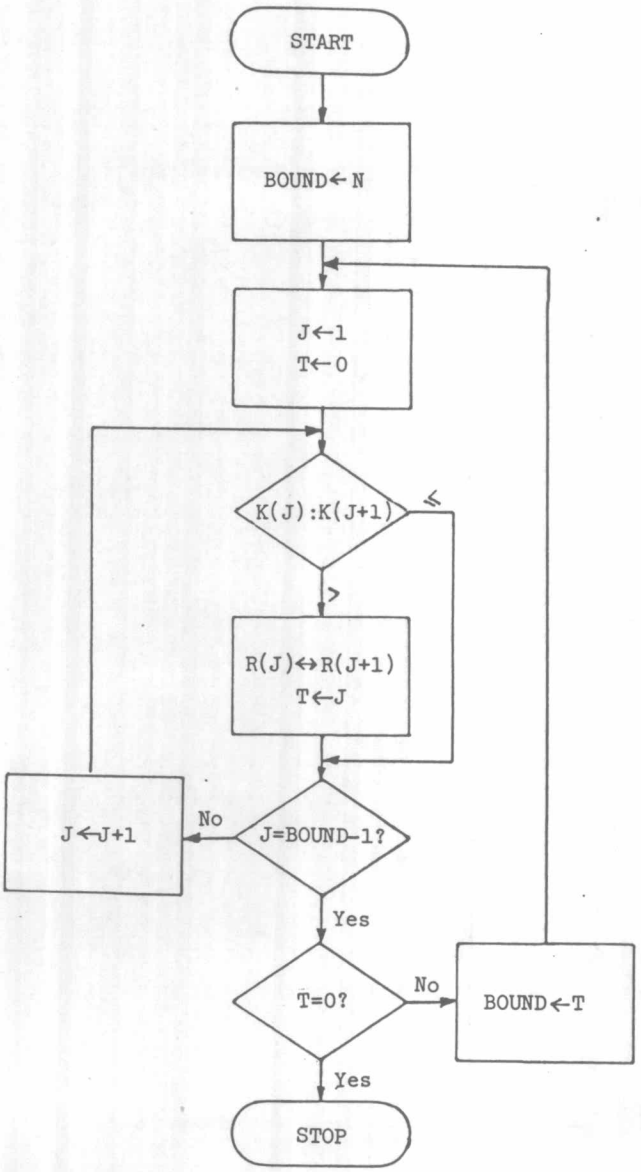

รูปที่ ก.1 ผังงานแสดงวิธีการเรียงลำดับข้อมูลแบบเปรียบเทียบโดยการนับ



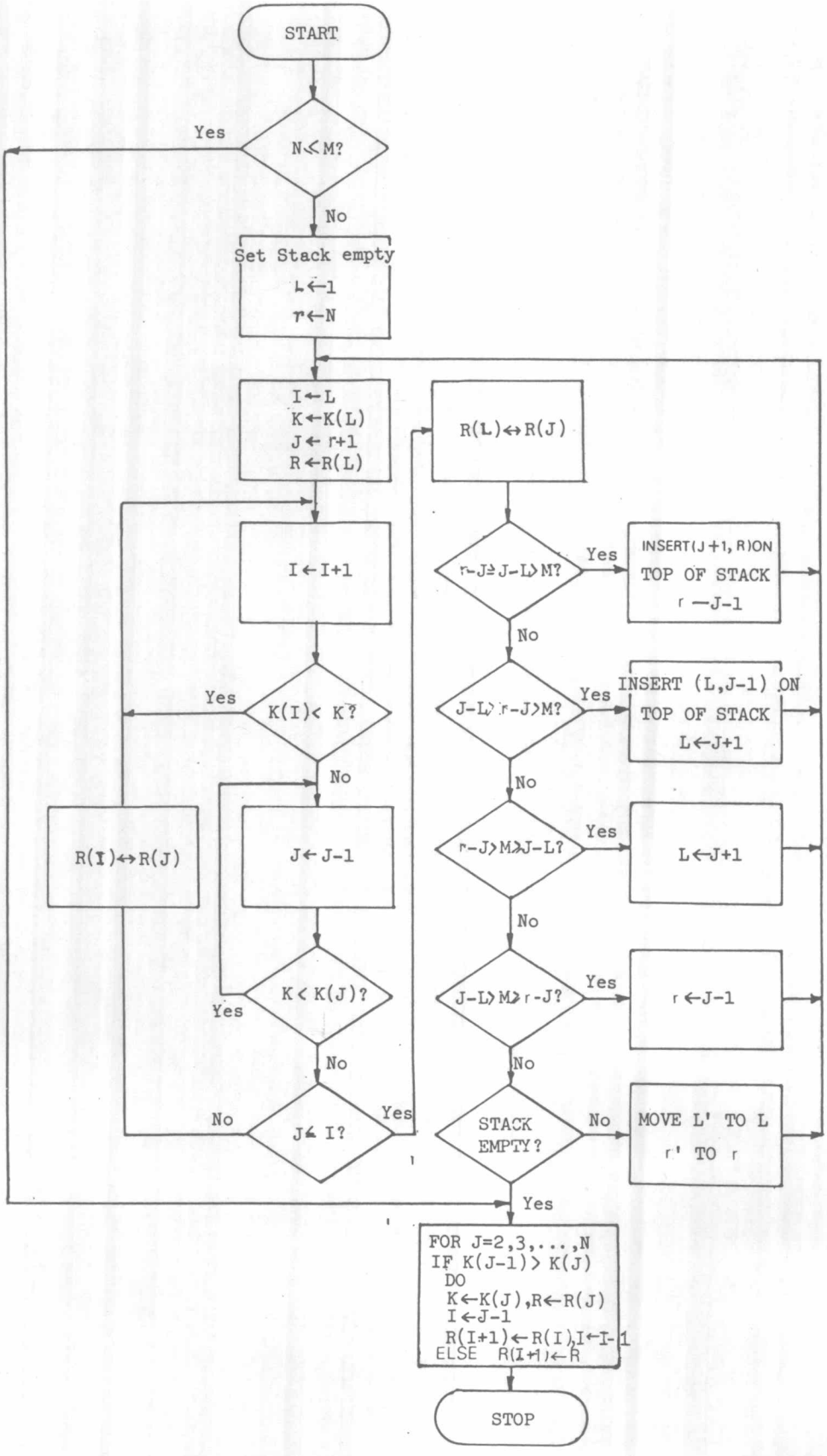

รูปที่ ก.2 ผังงานแสดงวิธีการเรียงลำดับข้อมูลแบบแทรก



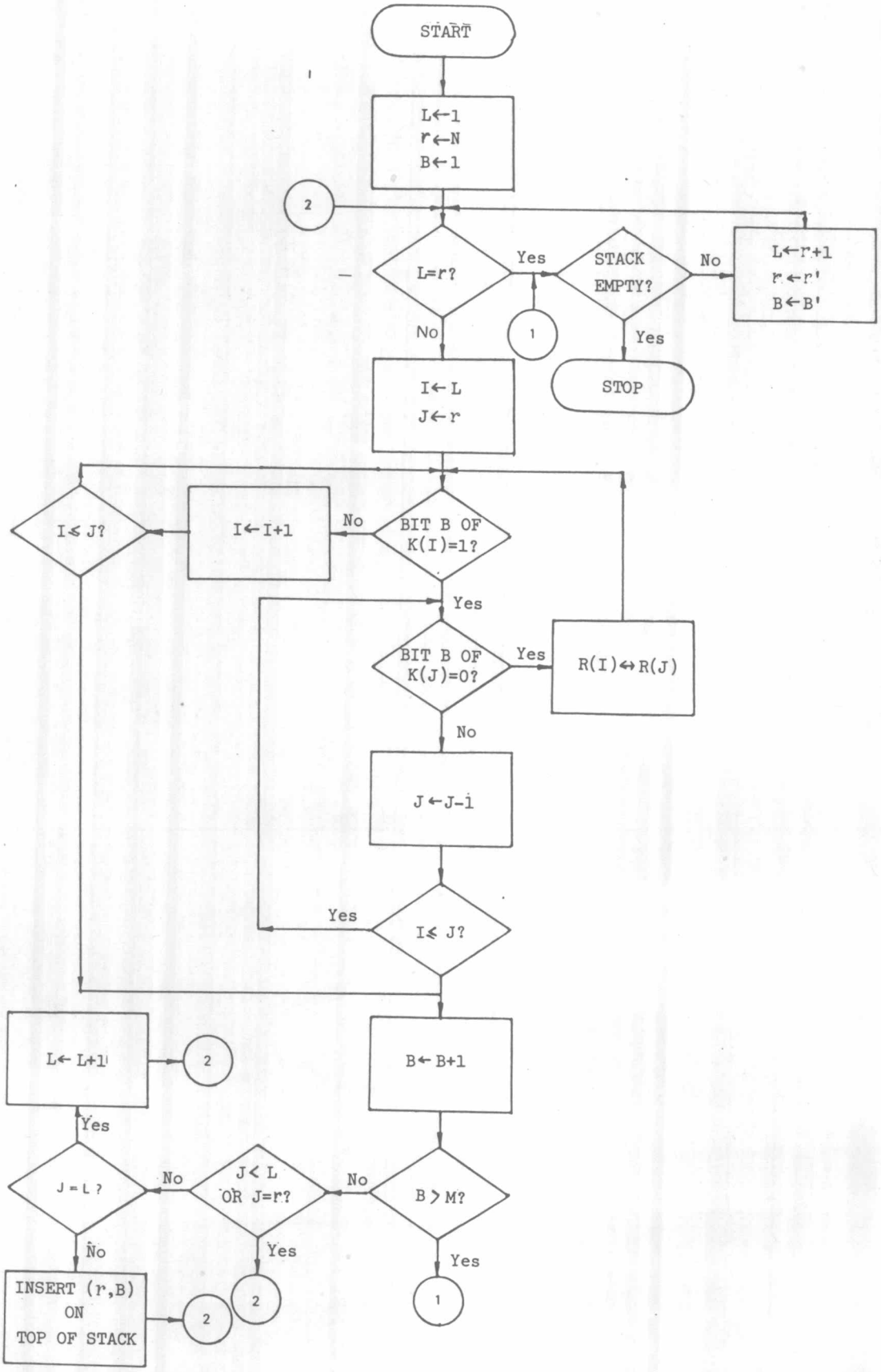

รูปที่ ก. 3 ผังงานแสดงวิธีการเรียงลำดับข้อมูลแบบเชลล์



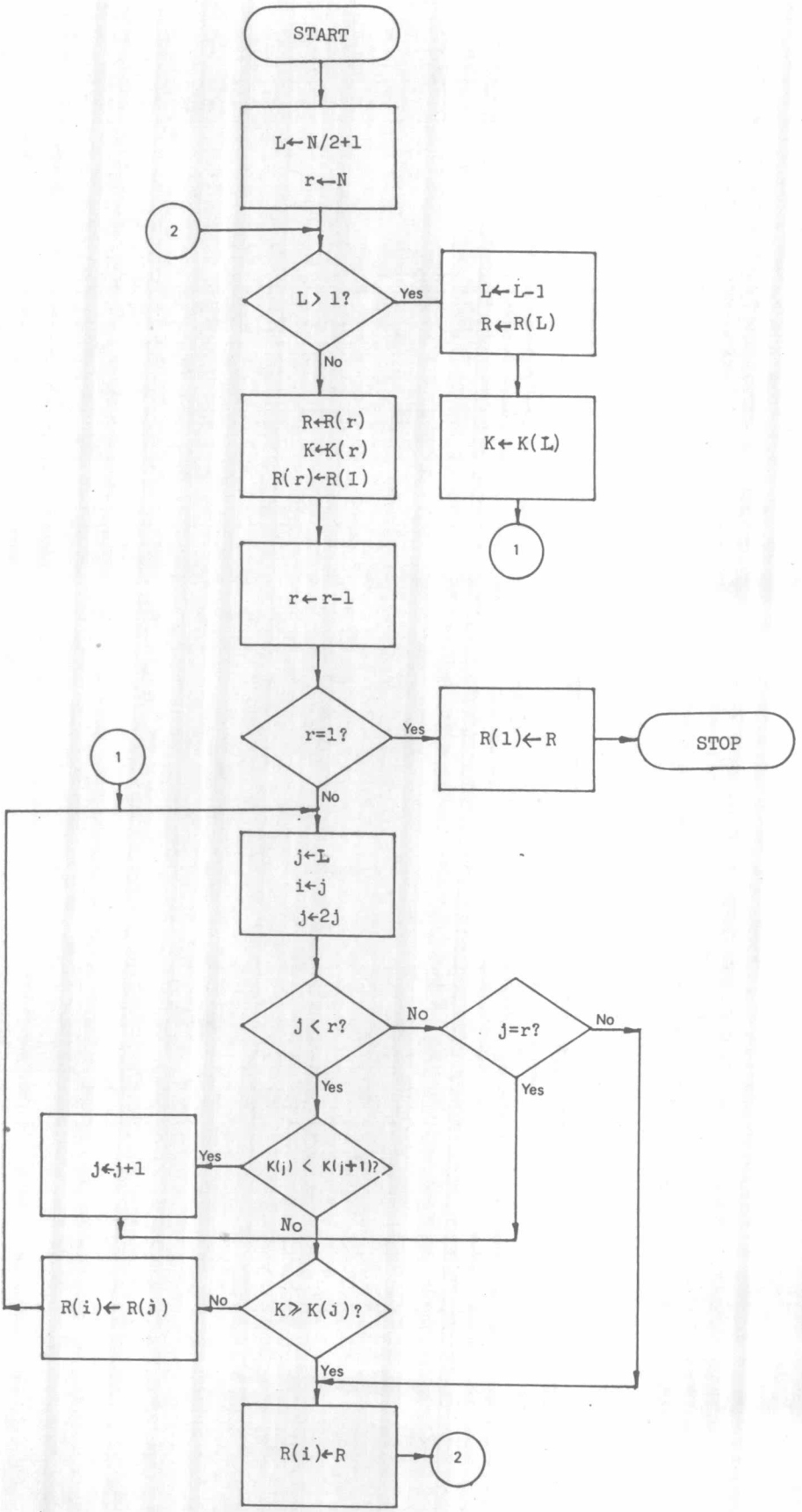

รูปที่ ก. 4 ผังงานแสดงวิธีการเรียงลำดับข้อมูลแบบลอยตัว



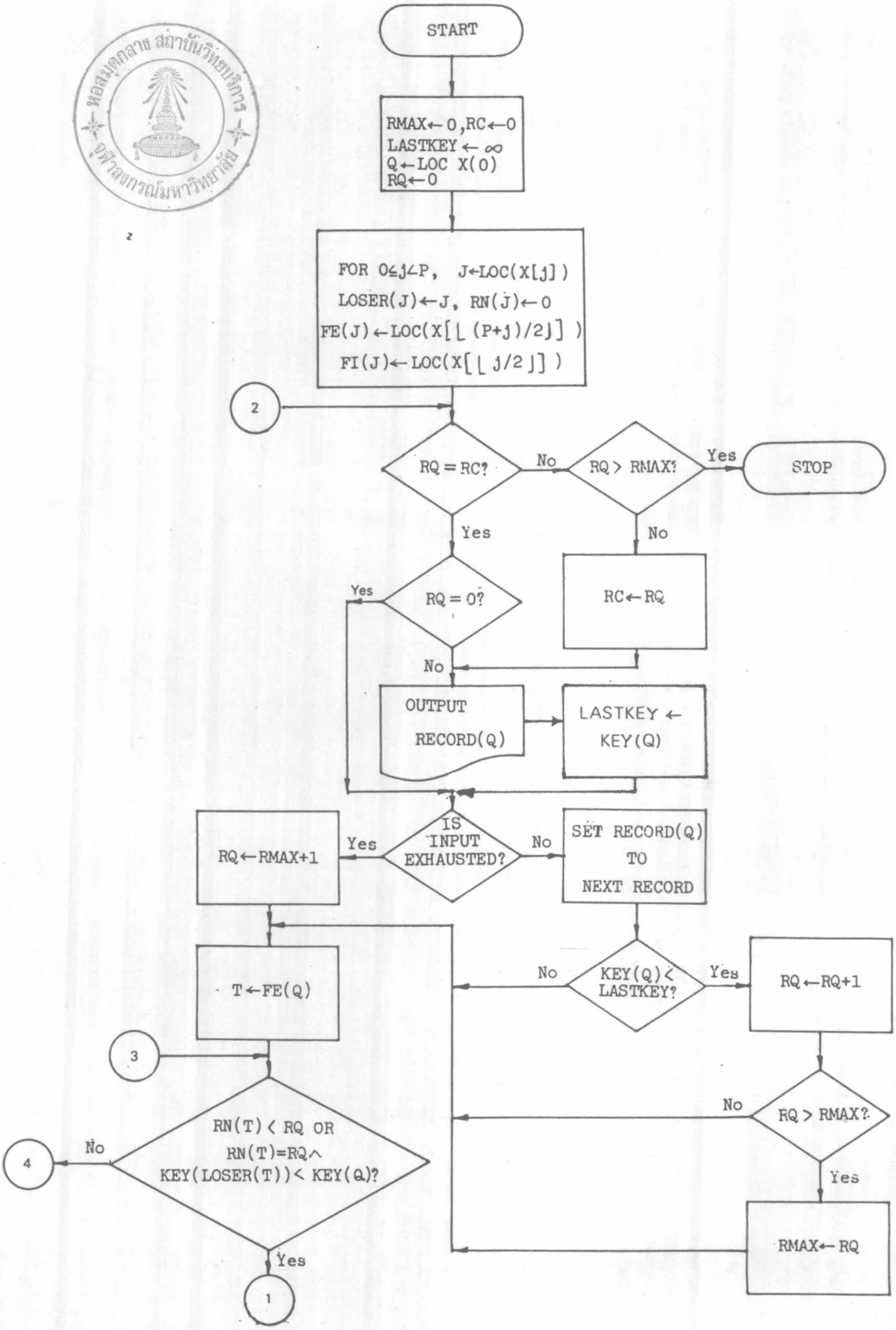

รูปที่ ก. 5 ผังงานแสดงวิธีการเรียงลำดับข้อมูลแบบควิคซอท



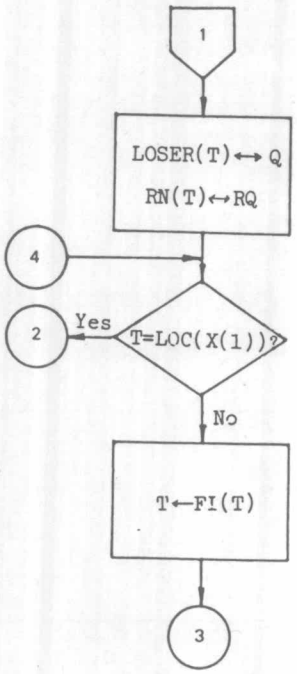

รูปที่ ก. 6 ผังงานแสดงวิธีการเรียงลำดับข้อมูลแบบเรดิกซ์-เอกเชนจ์




รูปที่ ก. 7 ผังงานแสดงวิธีการเรียงลำดับข้อมูลแบบฮีพซอท




รูปที่ ก. 8 ผังงานแสดงวิธีการเรียงลำดับข้อมูลแบบเลือกแทนที่




รูปที่ ก. 9 ผังงานแสดงวิธีการเรียงลำดับข้อมูลแบบเลือกแทนที่

ภาคผนวก ข.

คำสั่งควบคุมการเรียงลำดับที่ใช้ในการทดสอบ

1. คำสั่ง SORT เป็นคำสั่งให้เรียงลำดับ ที่ใช้ในการทดสอบคือ

SORT FIELDS=(8,4,BI,A,1,7,BI,a),SIZE=1000,WORK=1

หมายความว่า ให้เรียงลำดับเขตข้อมูล คอลัมน์ที่ 8 ยาว 4 ไบท์ และเขตข้อมูลคอลัมน์ที่ 1 ยาว 7 ไบท์ จากน้อยไปมากทั้งสองเขตข้อมูล โดยมีเขตข้อมูลแรกเป็นคีย์หลัก จำนวนระเบียนทั้งหมดมีประมาณ 1000 ระเบียน และจำนวนแฟ้มข้อมูลที่เป็นที่เก็บข้อมูลชั่วคราวมี 1 แฟ้มข้อมูล

2. คำสั่ง RECORD เป็นคำสั่งที่อธิบายเกี่ยวกับระเบียน ที่ใช้ในการทดสอบคือ

RECORD LENGTH=(1025,,),TYPE=F

หมายความว่า ระเบียนมีความยาวคงที่ระเบียนละ 1025 ไบท์ และไม่มีการทำให้ความยาวของระเบียนเปลี่ยนแปลง

3. คำสั่ง INPFIL เป็นคำสั่งอธิบายเกี่ยวกับแฟ้มข้อมูลนำเข้า ที่ใช้ในการทดสอบคือ

INPFIL BLKSIZE=5125,BYPASS,CLOSE=RWD

หมายความว่า แฟ้มข้อมูลนำเข้าเก็บข้อมูลเป็นบลอค แต่ละบลอคมีขนาด 5125 ไบท์ ในกรณีที่พบระเบียนที่มีความยาวไม่เท่ากับ 1025 ไบท์ จะไม่นำระเบียนนั้นมาเรียงลำดับและเมืออ่านข้อมูลจบแล้วให้หมุนเทปกลับ

4. คำสั่ง OUTFIL เป็นคำอธิบายเกี่ยวกับแฟ้มข้อมูลผลลัพธ์ ที่ใช้ในการทดสอบคือ

```
OUTFIL  BLKSIZE =5125
```

หมายความว่า ให้เก็บข้อมูลในแฟ้มข้อมูลผลลัพธ์เป็นบลอค บลอคละ 5125 ไบท์

5. คำสั่ง OPTION เป็นการเลือกกำหนดให้โปรแกรมสำเร็จรูปทำงานบางอย่างที่ผู้ใช้งานต้องการ ที่ใช้ในการทดสอบคือ

```
OPTION LABEL=(U,U,S),PRINT=ALL,ROUTE=LST,NODUMP,DIAG
```

หมายความว่าเลเบลของแฟ้มข้อมูลนำเข้าและแฟ้มข้อมูลผลลัพธ์เป็นประเภทไม่มีเลเบล ส่วนของแฟ้มข้อมูลที่เป็นที่เก็บข้อมูลชั่วคราวจะเป็นประเภทมาตรฐาน การพิมพ์ข้อความเกี่ยวกับการทำงานของโปรแกรมสำเร็จรูปให้พิมพ์ทางกระดาษทั้งข้อความที่เป็นความผิดพลาดธรรมดา และความผิดพลาดที่อาจทำให้โปรแกรมสำเร็จรูปหยุดทำงาน และเมื่อเกิดความผิดพลาดจากการเรียงลำดับไม่ต้อง DUMP ข้อมูลออกจากหน่วยความจำ

ภาคผนวก ค.

รายละเอียดของข้อมูลที่เก็บอยู่ใน PPI

บริเวณ PPI เป็นบริเวณที่เก็บข้อมูลที่สร้างจากเฟสต่าง ๆ เพื่อทำให้เฟสเหล่านั้นสามารถทำงานติดต่อกันได้ บริเวณ PPI จะถูกสร้างตั้งแต่เริ่มการทำงานของเฟส 0 จนกระทั่งจบการทำงานของโปรแกรม DSECT สำหรับบริเวณนี้คือ ILSSPPI

PPI จะอยู่ที่ตำแหน่งต่ำสุดในหน่วยความจำ ที่ใช้เป็นที่ทำงานของโปรแกรมเรียงลำดับนี้

ทุก ๆ โมดุลของโปรแกรมจะใช้รีจีสเตอร์ 13 เป็นเบสรีจีสเตอร์ของ PPI การอ้างถึงเขตข้อมูลใน PPI ทำได้โดยการบวกค่าดีสเพลสเมนท์กับข้อมูลในเบสรีจีสเตอร์ ซึ่งจะได้แสดงค่าดีสเพลสเมนท์ทั้งในรูปของเลขฐาน 10 และเลขฐาน 16 พร้อมทั้งคำอธิบายเขตข้อมูลต่าง ๆ

บริเวณข้อมูล ILSSPPI

ขนาด $694_{10}$ ไบท์

หน้าที่ เก็บข่าวสารระหว่างเฟส

| ดีสเพลสเมนท์ ฐาน 10 | ดีสเพลสเมนท์ ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | อธิบาย |
|---|---|---|---|---|
| 0 | 0 | 72 | PPISVARE | บริเวณเก็บค่ารีจีสเตอร์ของระบบ |
| 72 | 48 | 1 | PPIINBPT | จำนวนฟิสิคอลบลอค (Physical block) ที่เขียนใน 1 แทรค ของ SORTIN (สำหรับจานแม่เหล็ก ถ้าเป็นการรวมแฟ้มข้อมูลอย่างเดียว จะเป็นจำนวนฟิสิคอลบลอคซึ่งอยู่ในแทรคของข้อมูลที่จะนำมารวมกัน) |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | อธิบาย |
|---|---|---|---|---|
| | | | | ค่าของ PPIINBPT นี้จะถูกกำหนดโดยโมดุลที่คำนวณค่า B/G |
| 73 | 40 | 1 | PPIOBPT | จำนวนฟิสิคอลบลอคที่จะเขียนลงบน SORTOUT 1 แทรค ซึ่งสร้างโดยโมดุลคำนวณค่า B/G |
| 74 | 4A | 2 | PPIWK | ถ้าเป็นการเรียงลำดับโดยใช้เทปแม่เหล็ก จะเป็นจำนวนหน่วยของที่เก็บข้อมูลชั่วคราว (ฟิสิคอล) ถ้าเป็นการเรียงสำดับโดยใช้จานแม่เหล็ก ก็คือค่าของ M + 1 ถ้าเป็นการรวมแฟ้มข้อมูลอย่างเดียว ก็คือจำนวนแฟ้มข้อมูลนำเข้า + 1 สร้างขึ้นโดยโมดุล RCB ซึ่งเป็นโมดุลที่ประมวลผลคำสั่ง SORT/MERGE |
| 76 | 4C | 3 | PPINBUF | จำนวนบัฟเฟอร์ทั้งหมด ไบท์แรกแสดงจำนวนสำหรับเฟส 1 ไบท์ที่ 2 แสดงจำนวนสำหรับเฟสที่ 2 และไบท์ที่ 3 สำหรับเฟสที่ 3 สร้างโดยโมดุลที่คำนวณค่า B/G |
| 79 | 4F | 1 | PPINUMCF | จำนวนเขตข้อมูลที่ต้องการเรียงลำดับ สร้างโดยโมดุล RCB |
| 80 | 50 | 40 | PPISTAR | เป็นตารางตำแหน่งเริ่มต้นของแฟ้มข้อมูลนำเข้าและแฟ้มข้อมูลผลลัพธ์ในจานแม่เหล็ก คือ OUT 1 , IN 1,...IN 9,โดยจะเก็บดังต่อไปนี้ |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| | | (ไบท์ 0) | XXXX.... | บอกประเภทของอุปกรณ์ |
| | | | 0000.... | ประเภท 2311 I/O |
| | | | 0001.... | ประเภท 2314 I/O |
| | | | 0010.... | ประเภท 3330 I/O |
| | | | 0100.... | ประเภท 3340 I/O |
| | | | 1110.... | ประเภทแฟ้มข้อมูลแบบ VSAM |
| | | | 1111.... | ประเภทแฟ้มข้อมูลอยู่ในเทป |
| | | | ....XXXX | (ตำแหน่งของจานแม่เหล็ก, |
| | | (ไบท์ 1-3) | | อยู่ในรูป CCHH หรือเป็น 1 หมด ถ้าแฟ้มข้อมูล อยู่ในเทปหรือจัดอยู่ในรูป VSAM) |
| 120 | 78 | 36 | PPISTARW | ตารางตำแหน่งเริ่มต้นของแฟ้มข้อมูลที่เป็นที่เก็บข้อมูลชั่วคราว จะมีทั้งหมด 9 ที่ ให้เก็บตำแหน่งผู้ใช้งานสามารถระบุได้ 8 แห่ง โปรแกรมจะแบ่งเนื้อที่ออกเป็น 2 ส่วน และเนื้อที่แห่งที่ 9 จะถูกสร้างขึ้น ถ้าจุดแบ่งครึ่งไปตกอยู่ในอีกบริเวณหนึ่ง สร้างโดยโมดุล RCC |
| | | (ไบท์ 120-127) | PPINFIB | ค่าระดับ FIBONACCI ใหม่ (มี 9 ตัว) สร้างโดยโมดุล RAA หรือ RAB |
| 138 | 8A | (ไบท์ 138-141) | PPIAWLNK | ข่าวสารที่เชื่อมกันสำหรับ ALTWK |
| | | (ไบท์ 0) | | เป็นหมายเลขของเทปที่ใช้เป็นที่เก็บข้อมูลชั่วคราว สร้างโดยโมดุล RCC |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| | | (ไบท์ 1) | | การเพิ่มค่าของที่เก็บข้อมูลชั่วคราวที่เป็นเทป |
| | | | | แม่เหล็ก ที่ติดต่อกับ ALTWK |
| | | | | สร้างโดยโมดุลที่ทำหน้าที่เขียนในเฟส 1 และ |
| | | | | โมดุลที่ทำหน้าที่อ่านหรือเขียนในเฟส 2 |
| | | (ไบท์ 2) | | สวิชที่ใช้บอกว่าจบแฟ้มข้อมูล (EOF) ที่ระดับนั้น ๆ |
| | | | | เปิดสวิชโดยโมดุลที่ทำหน้าที่อ่านในเฟส 2 |
| | | | | และปิดโดยโมดุล RAC หรือ RAD |
| | | (ไบท์ 3) | | รหัสของยูนิตของ EOV ที่จบก่อน |
| | | | | สร้างโดยโมดุลที่ทำหน้าที่เขียนในเฟส 1, 2 |
| 156 | 9C | 48 | PPICFLDS | ตารางของเขตข้อมูลที่จะเรียงลำดับ ตัวหนึ่งจะมี |
| | | | | 4 ไบท์ สำหรับแต่ละเขตข้อมูล |
| | | (ไบท์ 0 - 1) | | บอกตำแหน่งว่าเขตข้อมูลที่จะนำมาเรียงลำดับ |
| | | | | เป็นตำแหน่งที่เท่าใด |
| | | (ไบท์ 2) | | ความยาวของเขตข้อมูล |
| | | (ไบท์ 3) | | รูปแบบของเขตข้อมูลที่จะนำมาเรียงลำดับ |
| | | | | สร้างโดยโมดุล RCB |
| 160 | A0 | 4 | PPISEQCT | นับจำนวนของ ซีเควันซ์ ที่เกิดขึ้น |
| | | | | สร้างโดยโมดุลที่ทำหน้าที่ควบคุมการทำงาน |
| | | | | (Algorithm) |
| 164 | A4 | 40 | PPIENDAR | ตารางของตำแหน่งจบบริเวณที่เก็บข้อมูลใน |
| | | | | จานแม่เหล็ก สำหรับ SORTOUT หรือ |
| | | | | SORTIN 1-9 อยู่ในแบบ CCHH |
| | | | | สร้างโดยโมดุล RCC |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| 196 | C4 | 44 | | ตารางสำหรับการทำ CHKPT ในเฟส 3 ของการเรียงลำดับโดยใช้เทปแม่เหล็ก |
| 204 | CC | 36 | PPIENDW | ตารางของตำแหน่งจบแฟ้มข้อมูลชั่วคราวในจานแม่เหล็ก มี 9 ตัว อยู่ในแบบ CCHH |
| 204 | CC | 4 | PPIOFIB | ค่าระดับ FIBONACCI เก่า (นั่นคือจำนวนสตริงที่มีอยู่จริงบนเทปแต่ละม้วน) สร้างโดยโมดุล RAA หรือ RAB |
| 222 | DE | 2 | PPIDFIBL | ค่าที่เอามาใส่เพื่อให้เกิดระดับของ FIBONACCI สร้างโดยโมดุล RAA หรือ RAB |
| 240 | FO | 18 | PPITPTBL | ตารางของเทปแม่เหล็ก จะใช้ตัวละ 2 ไบท์ สำหรับยูนิตของเทปแม่เหล็กแต่ละม้วน จัดอยู่ในรูป |
| | | ไบท์ 0 | 1..0 0000 | เป็นข้อมูลนำเข้า (0 = ข้อมูลผลลัพธ์) |
| | | | .1.0 0000 | ยูนิตของเทปนั้นต้องเปิดแฟ้มข้อมูล (0 = ไม่ต้องเปิด) |
| | | | ..10 0000 | ยูนิตนั้นมีข้อมูลเต็ม |
| | | ไบท์ 1 | | การเพิ่มค่าของบลอคควบคุมการทำงาน สร้างโดยโมดุล RAA หรือ RAB |
| 240 | FO | 4 | PPISTAR2 | ถ้าเป็นเฟส 0 จะหมายถึงจำนวนแทรคในบริเวณที่เก็บข้อมูลชั่วคราว สร้างโดยโมดุลที่คำนวณค่า B/G |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| | | ไบท์ 0-1 | | ในเฟส 1-3 เป็นค่าออฟเซทของครึ่งที่สอง ของที่เก็บข้อมูลชั่วคราว |
| | | ไบท์ 2-3 | | ในเฟส 1-3 เป็น format ของแทรคสุดท้าย (Logical Track) ในเฟส 1 สร้างโดยโมดุล RAB |
| 244 | F4 | 2 | PPIKEYLN | ค่าของ KEYLEN ที่กำหนดในคำสั่ง OPTION สร้างโดยโมดุล RCM |
| 246 | F6 | 2 | PPIBPTRK | จำนวนฟิลิคอลบลอคใน 1 แทรค สำหรับแฟ้มข้อมูล ที่ใช้เป็นที่เก็บข้อมูลชั่วคราว สร้างโดยโมดุลที่คำนวณค่า B/G |
| 248 | F8 | 2 | PPIBPT2 | จำนวนฟิลิคอลบลอคใน 1 แทรค สำหรับอุปกรณ์นำข้อมูลเข้า 2314 เพื่อรวมแฟ้มข้อมูล สร้างโดยโมดุลคำนวณค่า B/G |
| 250 | FA | 2 | PPINEXT | จำนวนบริเวณที่ใช้เป็นที่เก็บแฟ้มข้อมูลชั่วคราว สร้างโดยโมดุล RCC แต่อาจสร้างขึ้นใหม่ ในเฟส 1 โดยโมดุล RAB ถ้าพบว่าจุดแบ่งครึ่งไปตกอยู่ในบริเวณใดบริเวณหนึ่งที่ใช้เป็นที่เก็บข้อมูลชั่วคราว |
| 250 | FA | 2 | PPIBPT3 | จำนวนฟิลิคอลบลอคใน 1 แทรค สำหรับอุปกรณ์นำข้อมูลเข้า 3330 เพื่อรวมแฟ้มข้อมูล |
| 252 | FC | 6 | PPIGVAL | จำนวนไบท์ต่อ 1 สตริง (ถ้าเรียงลำดับโดยใช้จานแม่เหล็ก) สร้างโดยโมดุล RCM หรือโมดุลที่คำนวณค่า B/G และจะถูกแก้ไขในการทำงานแต่ละรอบของเฟส 2 โดยโมดุล RAC หรือ RAD |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเซตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| 258 | 102 | 18 | PPIDFIB | ค่าคงที่ที่ใส่เข้ามาเพื่อให้ได้ค่าระดับของ FIBONACCI สร้างโดยโมดุล RAA หรือ RAB |
| 258 | 102 | 18 | PPIUNIT | ชื่อตำแหน่งแต่ละหน่วยของจานแม่เหล็กที่ใช้เป็นบริเวณที่เก็บข้อมูลชั่วคราว บริเวณหนึ่งจะใช้เนื้อที่ 2 ไบท์ สร้างโดยโมดุล RCC |
| 276 | 114 | 6 | PPISW1 | สวิช |
| | | ไบท์ 0 | 1... .... | ข้อมูลมีความยาวไม่คงที่ (0 = ความยาวคงที่) |
| | | | .1.. .... | ใช้เทปเป็นที่เก็บข้อมูลชั่วคราว |
| | | | ..X. .... | ที่เก็บข้อมูลชั่วคราวใช้อุปกรณ์ 2311 |
| | | | ...1 .... | ที่เก็บข้อมูลชั่วคราว ใช้อุปกรณ์ 2314 |
| | | | .... 1... | มีที่เก็บข้อมูลชั่วคราว บางหน่วยเป็นเทป 7 แทรค (0 = ทั้งหมดใช้เทป 9 แทรค) |
| | | | .... .1.. | กำลังประมวลผลเฟส 1 |
| | | | .... ..1. | กำลังประมวลผลเฟส 2 |
| | | | .... ...1 | กำลังประมวลผลเฟส 3 |
| | | ไบท์ 1 | 1... .... | ใช้สำหรับรวมแฟ้มข้อมูลอย่างเดียว |
| | | | .1.. .... | ต้องดึงเซตข้อมูลที่ใช้ในการเรียงลำดับ |
| | | | ..1. .... | เรียงลำดับจากมากไปน้อย (0 = เรียงลำดับจากน้อยไปมาก) |
| | | | ...1 .... | ลำดับของข้อมูลในเฟส คือ เรียงจากมากไปน้อย (0 = จากน้อยไปมาก) |

| | | |
|---|---|---|
| | .... 1... | กำหนดขนาดของแฟ้มข้อมูลให้ (ผู้ใช้งานไม่กำหนด) |
| | .... .1.. | โปรแกรมสำเร็จรูปถูกเรียกใช้จากโปรแกรมภาษาอื่น |
| | .... ..1. | ข้อมูลที่นำเข้า จะนำเข้าโดย E15 (กำหนด E15 ในคำสั่ง INPFIL) |
| | .... ...1 | ข้อมูลที่จะนำออก จะทำโดย E35 (กำหนด E35 ในคำสั่ง OUTFIL) |
| ไบท์ 2 | 1... .... | มีการออกจากโปรแกรมสำเร็จรูปไปทำ ณ จุดออก |
| | .XX. .... | เลือกพิมพ์ข้อความ<br>00 ไม่มีการพิมพ์<br>01 พิมพ์เฉพาะในกรณีที่เกิดความผิดพลาดอย่างมาก<br>10 พิมพ์ข้อความทั้งหมด |
| | ...X X... | เป็นการกำหนดเกี่ยวกับการทำคำสั่ง ADDROUT<br>00 ไม่มีการกำหนด ADDROUT<br>01 เอาค่าตำแหน่ง<br>10 เอาค่าตำแหน่งและเขตข้อมูลที่ใช้เรียงลำดับ |
| | .... .1.. | ตรวจสอบผลลัพธ์ที่จานแม่เหล็ก (กำหนด VERIFY) |
| | .... ..1. | มีการกำหนดคำสั่ง CALAREA |
| | .... ...1 | เลือกใช้คำสั่ง BYPASS |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| | | ไบท์ 3 | 1... .... | ข้อมูลที่นำเข้าอยู่ในรูป ASCII (0 = EBCDIC) |
| | | | .1.. .... | ข้อมูลอยู่ในรูป ASCII มีความยาวไม่คงที่ |
| | | | ..1. .... | กำหนดให้มีการทำ CHKPT |
| | | | ...1 .... | ต้องการให้มีการจัดรูปแบบของข้อมูล |
| | | | | บนที่เก็บข้อมูลชั่วคราว |
| | | | .... 1... | ครึ่งที่สองของที่เก็บข้อมูลชั่วคราว ที่เป็นจาน |
| | | | | แม่เหล็กกำลังใช้เขียนข้อมูลอยู่ (0 = ครึ่งแรก) |
| | | | .... .1.. | จำนวนหน่วยของที่เก็บข้อมูลชั่วคราวถูกเพิ่มอีก1 |
| | | | .... ..1. | ซีเลคเตอร์แชนแนล 2 ตัว (0 = ซีเลคเตอร์ |
| | | | | 1 ตัว หรือเป็นมัลติเพลคเซอร์) |
| | | | .... ...1 | อุปกรณ์ที่เป็น SYSLOG เป็นเครื่องพิมพ์ |
| | | | | (0 = คอนโซล) |
| | | ไบท์ 4 | XX.. .... | อุปกรณ์ SYSLST |
| | | | | 00 พิมพ์ผลลัพธ์ทางเครื่องพิมพ์ |
| | | | | 01 คอนโซล |
| | | | | 10 เทปแม่เหล็ก |
| | | | | 11 จานแม่เหล็ก |
| | | | ..1. .... | ข้อมูลนำเข้าจะเก็บที่เดียวกับข้อมูลผลลัพธ์ |
| | | | ...1 .... | ข้อมูลผลลัพธ์เก็บที่เดียวกับที่เก็บข้อมูลชั่วคราว |
| | | | .... 1... | ข้อมูลนำเข้าเก็บที่เดียวกับที่เก็บข้อมูลชั่วคราว |
| | | | .... .1.. | จบแฟ้มข้อมูลในแฟ้มข้อมูลนำเข้าแฟ้มแรกจริง |
| | | | .... ..1. | มีการเลือกใช้ ALTWK |
| | | | .... ...1 | จบแฟ้มข้อมูลโดยผู้ใช้งาน (E15) |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| | | ไบท์ 5 | XX.. .... | อุปกรณ์ที่ใช้ในการทำคำสั่ง CHKPT |
| | | | | 00 คือ 2400 |
| | | | | 01 คือ 2311 |
| | | | | 11 คือ 2314 |
| | | | ..1. .... | ใช้การเรียงลำดับแบบ Split-Cylinder |
| | | | ...1 .... | ใช้อุปกรณ์ 3330 เป็นอุปกรณ์ที่ใช้ในการทำ CHKPT |
| | | | .... 1.... | มีบัฟเฟอร์ของผลลัพธ์ 2 บัฟเฟอร์ในเฟส 3 (0 = มีบัฟเฟอร์เดียว) |
| | | | .... .1.. | ใช้โปรแกรมสำเร็จรูปใน FOREGROUND (0 = BACKGROUND) |
| | | | .... ..1. | โปรแกรมสำเร็จรูปหยุดการทำงาน เนื่องจากผู้ใช้งาน ณ จุด E15 E25 หรือ E35 |
| | | | .... ...1 | กำหนดให้มีการทำ RESTART ของ CHKPT |
| 282 | 11 | 2 | PPIRCDL1 | ค่า $l_1$ จากคำสั่งควบคุม RECORD (หรือใส่ให้เอง) |
| 284 | 11 | 2 | PPIRCDL2 | ค่า $l_2$ จากคำสั่งควบคุม RECORD (หรือใส่ให้เอง) |
| 286 | 11 | 2 | PPIRCDL3 | ค่า $l_3$ จากคำสั่งควบคุม RECORD (หรือใส่ให้เอง) |
| 288 | 120 | 2 | PPIRCDL4 | ค่า $l_4$ จากคำสั่งควบคุม RECORD (หรือใส่ให้เอง) |
| 290 | 122 | 2 | PPIRCDL5 | ค่า $l_5$ จากคำสั่งควบคุม RECORD (หรือใส่ให้เอง) |

ดีสเพ่ลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| 292 | 124 | 4 | PPICKTAB | ตำแหน่งของตารางการทำ CHKPTในเทป แม่เหล็ก สร้างโดยโมดุลที่ทำหน้าที่อ่านหรือเขียนข้อมูล |
| 296 | 128 | 4 | PPISRTG | ค่าG จำนวนของระเบียนใน RSA สร้างโดยโมดุลคำนวณค่า B/G |
| 300 | 12 | 4 | PPINMAX | จำนวนของระเบียนที่โปรแกรมสำเร็จรูป จะจัดเรียงให้ได้ สร้างโดยโมดุลคำนวณค่า B/G |
| 304 | 130 | 4 | PPIRMAX | เทปแม่เหล็ก: คือจำนวนความจุของข้อมูล เป็นไบท์ที่พอเหมาะที่จะรวมกันเป็น 1 บลอค ซึ่งเหมาะกับความยาวของเทป จานแม่เหล็ก: คือขนาดของพื้นที่หลัก (M * GVAL) สร้างโดยโมดุลคำนวณค่า B/G |
| 308 | 134 | 4 | PPITAVLC | ขนาดของหน่วยความจำ ที่จะนำมาใช้ในการ เรียงลำดับได้ สร้างโดยโมดุล RCM หรือโมดุล RCC |
| 312 | 138 | 4 | PPISPGN1 | ตำแหน่งของไบท์ต่อไปในบริเวณที่เก็บข้อมูลชั่วคราว สร้างโดยโมดุลที่กำหนดการทำงานโมดุลแรก ในแต่ละเฟส ตั้งแต่เฟส 1-3 |
| 316 | 13 | 2 | PPIMRGOR | ค่า M จะกำหนดให้เป็นค่าสูงสุดโดยโมดุลที่คำนวณค่า B/G และสามารถเปลี่ยนแปลงได้ |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| 318 | 13E | 2 | PPINBLK | ขนาดของบลอคของข้อมูลนำเข้า<br>สร้างโดยโมดุล RCK |
| 320 | 140 | 2 | PPIOBLK | ขนาดของบลอคของข้อมูลที่เป็นผลลัพธ์<br>สร้างโดยโมดุล RCK |
| 322 | 142 | 2 | PPISBLK | ค่าขนาดของบลอคของที่เก็บข้อมูลชั่วคราว<br>สร้างโดยโมดุลคำนวณค่า B/G |
| 324 | 144 | 4 | PPIINTAB | ตำแหน่งของตารางบัฟเฟอร์ของข้อมูลนำเข้า<br>สร้างโดยโมดุลที่มีหน้าที่จัดเตรียมเนื้อที่ในหน่วยความจำ |
| 328 | 148 | 4 | PPIOUT1 | ตำแหน่งบัฟเฟอร์ผลลัพธ์บัฟเฟอร์แรก<br>ในเฟส 3 จะชี้ไปยังตำแหน่งของบัฟเฟอร์ตัวที่เพิ่งจะใส่ข้อมูลลงไป<br>สร้างโดยโมดุลที่ทำหน้าที่จัดเตรียมเนื้อที่ในหน่วยความจำ จะเพิ่มค่าอีก 4 ไบท์<br>ถ้าระเบียนมีความยาวไม่คงที่ |
| 332 | 14C | 4 | PPIOUT2 | ตำแหน่งของบัฟเฟอร์ผลลัพธ์บัฟเฟอร์ที่ 2 |
| 336 | 150 | 4 | PPIBINSZ | ในเฟส 1 คือค่าของเนื้อที่แต่ละส่วนใน RSA<br>สร้างโดยโมดุลซึ่งทำหน้าที่คำนวณขนาดของ Bin. |
| 336 | 150 | 4 | PPIOUTW | ในเฟส 3 บริเวณใช้สำหรับเก็บตำแหน่งเริ่มต้นของบัฟเฟอร์ผลลัพธ์ที่เกิดขึ้นขณะนั้น (ติดต่อกับโมดุลที่ทำหน้าที่เขียน) |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| 340 | 154 | 4 | PPIAVBIN | ตำแหน่งของบริเวณที่เก็บข้อมูลต่อไป สร้างโดยโมดุลที่ทำหน้าที่นำข้อมูลเข้าหรือออกจากบลอค |
| 340 | 154 | 4 | PPIDBTAB | ตำแหน่งตารางที่เก็บตำแหน่งบัฟเฟอร์ของระเบียนนำเข้า ซึ่งเพิ่งจะนำออกจากบลอค สร้างโดยโมดุลที่ทำหน้าที่อ่านข้อมูลใน เฟส 2 หรือ 3 |
| 344 | 158 | 4 | PPINBIN | จำนวนของบริเวณที่เก็บข้อมูล (Bin) ทั้งหมด สร้างโดยโมดุลที่ทำหน้าที่นำข้อมูลเข้าหรือออกจากบลอคใน เฟส 1 |
| 344 | 158 | 4 | PPISEQRD | ตำแหน่งระเบียนในบริเวณที่เก็บผลลัพธ์ (เอาไว้สำหรับตรวจสอบลำดับ) สร้างโดยโมดุลที่ทำหน้าที่บลอค |
| 348 | 15C | 4 | PPIMDEX1 | ข่าวสารเกี่ยวกับส่วนโปรแกรมของผู้ใช้งาน สำหรับ เฟส 1 |
| | ไบท์ 0 | | 1... .... | โปรแกรมสำเร็จรูปกำหนดตำแหน่งให้กับส่วนโปรแกรมของผู้ใช้งาน (0 = ผู้ใช้งานกำหนดให้) |
| | | | .1.. .... | ออก ณ จุดออก E11 |
| | | | ..1. .... | ออก ณ จุดออก E15 |
| | | | ...1 .... | ออก ณ จุดออก E17 |
| | | | .... 1... | ออก ณ จุดออก E18 |
| | | | .... .XXX | ไม่ได้ใช้งาน |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| | ไบท์ 1-3 | | | ตำแหน่งโปรแกรมของผู้ใช้งาน สร้างโดยโมดูล RCJ ตำแหน่งเหล่านี้อาจสร้างโดยโมดูล RCC ถ้าเป็นการรีโลเคท (relocate) |
| 352 | 160 | 4 | PPIMDEX2 | ข่าวสารเกี่ยวกับโปรแกรมของผู้ใช้งานสำหรับเฟส 2 |
| | ไบท์ 0 | | 1... .... | โปรแกรมสำเร็จรูปกำหนดตำแหน่งให้โปรแกรมของผู้ใช้งาน (0 = ผู้ใช้งานกำหนดให้) |
| | | | .1.. .... | ออก ณ จุดออก E21 |
| | | | ..1. .... | ออก ณ จุดออก E25 |
| | | | ...1 .... | ออก ณ จุดออก E27 |
| | | | .... XXXX | ไม่ได้ใช้งาน |
| | ไบท์ 1-3 | | | ตำแหน่งโปรแกรมของผู้ใช้งาน เช่นเดียวกับ PPIMDEX1. |
| 356 | 164 | 4 | PPIMDEX3 | ข่าวสารเกี่ยวกับโปรแกรมของผู้ใช้งานสำหรับเฟส 3 |
| | ไบท์ 0 | | 1... .... | โปรแกรมสำเร็จรูปกำหนดตำแหน่งโปรแกรมของผู้ใช้งานให้ (0 = ผู้ใช้งานกำหนดเอง) |
| | | | .1.. .... | ออก ณ จุดออก E31 |
| | | | ..1. .... | ออก ณ จุดออก E35 |
| | | | ...1 .... | ออก ณ จุดออก E37 |
| | | | .... 1... | ออก ณ จุดออก E38 |
| | | | .... .1.. | ออก ณ จุดออก E39 |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| | | | .... ..1. | ออก ณ จุดออก E32 |
| | | | .... ...X | ไม่ได้ใช้งาน |
| | ไบท์ 1-3 | | | เป็นตำแหน่งโปรแกรมของผู้ใช้งาน เช่นเดียวกับ PPIMDEX1 |
| 360 | 168 | 4 | PPIAXERT | ตำแหน่งของส่วนโปรแกรมที่ทำหน้าที่ดึงข้อมูล สร้างโดยโมดุลที่คำนวณค่าขนาดของ Bin และถูกสร้างขึ้นใหม่อีกโดยโมดุล RTMG เมื่อส่วนโปรแกรมนั้นถูกนำไปเก็บไว้ที่ตอนต้นในหน่วยความจำ |
| 364 | 16C | 4 | PPIEXTSZ | ขนาดของส่วนโปรแกรมที่ทำหน้าที่ดึงข้อมูล สร้างโดยโมดุลที่ทำหน้าที่คำนวณขนาดของ Bin |
| 368 | 170 | 2 | PPILEXFD | ความยาวของเขตข้อมูล ที่ใช้ในการเรียงลำดับทั้งหมดที่ดึงออกมา ค่านี้เอาไว้ใช้สำหรับคำสั่งที่ใช้เปรียบเทียบ สร้างโดยโมดุลที่คำนวณค่าของ Bin |
| 370 | 172 | 2 | PPILEXFF | ค่าของ PPILEXFD ที่ปัดขึ้นให้มีค่าเป็น FULLWORD สร้างโดยโมดุลที่คำนวณค่าของ Bin |
| 372 | 174 | 4 | PPISEQCK | บริเวณที่ใช้ในการติดต่อสำหรับการตรวจสอบลำดับ สร้างโดยโมดุลที่ทำหน้าที่บลอคในเฟส 3 |
| 376 | 178 | 4 | PPIFILSZ | ขนาดของแฟ้มข้อมูล ตามที่กำหนด สร้างโดยโมดุล RCB |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเซตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| 380 | 17C | 4 | PPIEXSZ1 | ในเฟส 0 คือขนาดโปรแกรมของผู้ใช้งาน ถ้าผู้ใช้งานกำหนดให้ในคำสั่งควบคุม MODS. |
| 380 | 17C | 4 | PPIRCDCT | จำนวนของระเบียนในแฟ้มข้อมูลนำเข้า สร้างโดยโมดุลสุดท้ายของแต่ละเฟส มีค่าเท่ากับ PPICOUNT + PPIDELCT - PPIINSCT |
| 380 | 180 | 4 | PPIEXSZ2 | ในเฟส 0 คือขนาดโปรแกรมของผู้ใช้งาน ถ้าผู้ใช้งานกำหนดให้ สร้างโดยโมดุล RCB |
| 384 | 180 | 4 | PPICOUNT | คือจำนวนของระเบียนที่เข้าประมวลผลตลอดการทำงานของเฟส สร้างโดยโมดุลที่ทำหน้าที่ที่นำข้อมูลเข้า หรือออกจากบลอค และถูกสร้างให้มีค่าเป็น 0 เมื่อจบการทำงานของเฟส |
| 388 | 184 | 4 | PPIEXSZ3 | ในเฟส 0 คือขนาดโปรแกรมของผู้ใช้งานในเฟส 3 ถ้าผู้ใช้งานกำหนดให้ สร้างโดยโมดุล RCB |
| 388 | 184 | 4 | PPIINSCT | จำนวนระเบียนที่เพิ่มเข้ามา สร้างโดยโมดุลที่ทำหน้าที่นำข้อมูลเข้าหรือออกจากบลอค |
| 392 | 188 | 4 | PPIDELCT | จำนวนของระเบียนที่นำออก สร้างโดยโมดุลที่ทำหน้าที่นำข้อมูลเข้าหรือออกจากบลอค |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| 396 | 18C | 4 | PPISBLCT | ตำแหน่งของตาราง ที่ทำหน้าที่นับค่าบลอค สร้างโดยโมดุลที่ทำหน้าที่นำข้อมูลเข้าหรือออกจากบลอค |
| 400 | 190 | 4 | PPISTDCB | ตำแหน่งของตารางที่ใช้เก็บตำแหน่งของบลอค ควบคุมการทำงาน |
| 404 | 194 | 32 | PPIPDWA | บริเวณที่สร้างขึ้นของโมดุล ที่ทำหน้าที่รวมแฟ้มข้อมูล สำหรับเริ่มสร้างและเรียงลำดับระเบียน M ระเบียน<br>ในเฟส 0 หรือ 1 จะใช้เป็นบริเวณที่เก็บข้อมูลชั่วคราว<br>ในเฟส 2 จะใช้เป็นบริเวณที่เก็บข้อมูลชั่วคราวของทุกโมดุล นอกจากโมดุลที่ใช้ในการอ่านและโมดุลที่ทำหน้าที่รวมแฟ้มข้อมูล<br>ในเฟส 3 ใช้เป็นบริเวณที่เก็บข้อมูลชั่วคราว นอกจากโมดุลที่ทำหน้าที่เลือกส่วนโปรแกรม ซึ่งทำหน้าที่อ่านข้อมูลและโมดุลซึ่งทำหน้าที่นำโมดุลอื่นเข้ามาในหน่วยของความจำ และโมดุลที่ทำหน้าที่รวมแฟ้มข้อมูล จะประกอบด้วยค่าดีสเพลสเมนท์ของโมดุลที่ทำหน้าที่นำโมดุลอื่นเข้ามาในหน่วยความจำ |

ดีสเพลสเมนท์ คือ ระยะที่โมดุลอยู่ห่างจากตำแหน่งเริ่มต้นของ PPI ดังนี้คือ

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเซตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| 404 | 194 | 2 | | ดีสเพลสเมนท์ของโมดุลที่นำโมดุลอื่นเข้ามาในเฟส 2 |
| 406 | 196 | 2 | | ดีสเพลสเมนท์ของโมดุลที่นำโมดุลอื่นเข้ามาในเฟส 3 |
| 434 | 1B2 | 2 | | ดีสเพลสเมนท์ของโมดุลที่นำโมดุลอื่นเข้ามาในเฟส 1 |
| 436 | 1B4 | 32 | | รายการของโมดุลที่ต้องติดต่อ แต่ละตัวในรายการนี้ก็คือ ตำแหน่งของโมดุลในหน่วยความจำหลัก ตำแหน่งเหล่านี้จะถูกสร้างโดยโมดุลที่ทำหน้าที่นำโมดุลอื่นเข้ามาสำหรับแต่ละเฟสของเฟส 1 - 3 |
| 436 | 1B4 | 4 | PPISSC | โมดุลที่ทำหน้าที่ควบคุมการทำงานของเฟส |
| 440 | 1B8 | 4 | PPIALG | โมดุลกำหนดขั้นตอนการทำงาน |
| 444 | 1BC | 4 | PPIDEB | โมดุลที่ทำหน้าที่นำข้อมูลออกจากบลอค |
| 448 | 1CC | 4 | PPINET | โมดุลที่ทำหน้าที่เรียงลำดับข้อมูลหรือรวมข้อมูล |
| 452 | 1C4 | 4 | PPIBLK | โมดุลที่ทำหน้าที่บลอค |
| 456 | 1C8 | 4 | PPIWRT | โมดุลที่ทำหน้าที่เขียน |
| 460 | 1CC | 4 | PPIRD | โมดุลที่ทำหน้าที่อ่าน |
| 464 | 1D0 | 4 | PPIMOVE | โมดุลที่ทำหน้าที่เคลื่อนข้อมูล |
| 456 | 1C8 | 12 | | รายการของโมดุลที่ต้องติดต่อใน เฟส 0 |
| 456 | 1C8 | 4 | PPIDICT | ตำแหน่งของคำสั่งควบคุมที่ได้ทำให้สั้นลงเรียบร้อยแล้ว สร้างโดยโมดุล RCA |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเซตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| 460 | 1CC | 4 | PPISCAN | ตำแหน่งของโมดุล RCL<br>สร้างโดยโมดุล RCA |
| 464 | 1D0 | 4 | PPIPRINT | ตำแหน่งโมดุลที่ทำหน้าที่พิมพ์ของเฟส 0<br>สร้างโดยโมดุล RTMG |
| 468 | 1D4 | 2 | PPIDEPHO | ในเฟส 1 คือชื่อตำแหน่งของผลลัพธ์<br>ในเฟส 2 คือจำนวนยูนิตของเทปผลลัพธ์ที่เพิ่มขึ้น<br>ในเทปจะสร้างโดยโมดุลที่กำหนดขั้นตอนการทำงาน<br>ในเฟส 1 หรือ 2<br>ในเฟส 3 จะบ่งประเภทของข้อมูลที่นำเข้า<br>(สำหรับจานแม่เหล็ก) |
| | ไบท์ 0 | | | 0 = ข้อมูลนำเข้าอยู่ที่ครึ่งแรกของเนื้อที่<br>1 = ข้อมูลนำเข้าอยู่ที่ครึ่งที่สองของเนื้อที่ |
| | ไบท์ 1 | | | 0 = สำหรับข้อมูลที่เป็น Interleave<br>1 = สำหรับข้อมูลที่เป็น Sequential |
| 470 | 1D6 | 3 | PPIPRTS1 | ใช้บ่งข้อความที่จะพิมพ์ออกให้ผู้ใช้งานทราบ<br>สำหรับเฟส 1 - 3 |
| | | | | ข้อความหมายเลข สร้างโดย |
| | ไบท์ 0 | | 1... .... | 7901A โมดุลกำหนดขั้นตอนการทำงาน |
| | | | .1.. .... | 7902A โมดุลที่ทำงานตอนจบเฟส |
| | | | ..1. .... | 7903I โมดุลกำหนดขั้นตอนการทำงาน |
| | | | ...1 .... | 7904A โมดุลที่ทำหน้าที่อ่านหรือเขียนข้อมูล |
| | | | .... 1... | 7905I โมดุลที่ทำงานตอนจบเฟส |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย | |
|---|---|---|---|---|---|
| | | | | ข้อความหมายเลข | สร้างโดย |
| | | | .... .1.. | 7906I | โมดุลที่ทำงานตอนจบเฟส |
| | | | .... ..1. | 7908A | โมดุลซึ่งทำหน้าที่เลือกส่วนโปรแกรมที่จะทำงานเกี่ยวกับบลอคควบคุมการทำงาน |
| | | | .... ...1 | 7904A | โมดุลที่ทำหน้าที่อ่านหรือเขียน (สำหรับ CCB) |
| | | ไบท์ 1 | 1... .... | 7101I | โมดุลที่ทำงานตอนจบเฟส |
| | | | .1.. .... | 7n59A | โมดุลที่ทำงานตอนจบเฟส |
| | | | ..1. .... | 7909I | โมดุลที่ทำงานตอนจบเฟส |
| | | | ...1 .... | 7201I | โมดุลที่ทำงานตอนจบเฟส |
| | | | .... 1... | 7302I | EOJ |
| | | | .... .1.. | 7907I | โมดุลที่ทำหน้าที่เรียงลำดับหรือรวมแฟ้มข้อมูล |
| | | | .... ..1. | 7910I | EOJ |
| | | | .... ...X | ไม่ได้ใช้งาน | |
| | | ไบท์ 2 | 1... .... | 7917I | VSAM I/O* |
| | | | .1.. .... | 7918I | VSAM I/O* |
| | | | ..1. .... | 7919I | |
| | | | ...1 .... | 7920I | VSAM I/O* |
| | | | .... 1... | 7921I | VSAM I/O, RSN* |
| | | | .... .1.. | 7n85I | VSAM I/O,ASL, RSN* |

*จะยังไม่ใช้งานจนกว่าจะนำโปรแกรมของผู้ใช้งานที่เขียนให้เข้ามาทำงานในเฟส 1 เข้ามาในหน่วยความจำ

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| | | | | ข้อความหมายเลข สร้างโดย |
| | | | .... ..XX | ไม่ได้ใช้งาน |
| 472 | 1D8 | 8 | PPINMS1 | ชื่อของส่วนโปรแกรมของผู้ใช้งานในเฟส 1 สร้างโดยโมดุล RCJ |
| 480 | 1E0 | 8 | PPINMS2 | ชื่อส่วนโปรแกรมของผู้ใช้งานในเฟส 2 สร้างโดยโมดุล RCJ หลังจากนี้บริเวณนี้จะใช้สำหรับ VSAM เกี่ยวกับข่าวสารความผิดพลาดอันเกิดจากการ CLOSE ในเฟส 1 เพื่อส่งให้โมดุลที่ทำหน้าที่ในการพิมพ์ข้อความให้ผู้ใช้งานทราบ |
| 488 | 1E8 | 8 | PPINMS3 | ชื่อส่วนโปรแกรมของผู้ใช้งานในเฟส 3 สร้างโดยโมดุล RCJ บริเวณนี้จะใช้เก็บชื่อของแฟ้มข้อมูล ถ้าเกิดความผิดพลาดเนื่องจาก I/O หรือการเปิดแฟ้มข้อมูล VSAM หรือชื่อซองจุดออก |
| 496 | 1F0 | 4 | PPIATTCH | ตำแหน่งคอนดิชันโคด (Condition Code) ของโปรแกรมที่เป็นภาษา COBOL ตำแหน่งสร้างโดยโมดุล RTMG รหัสสร้างโดยโมดุล RCN |
| 500 | 1F4 | 2 | PPIK1 | ค่าคงที่ที่ใช้เสมอ |
| 502 | 1F6 | 2 | PPIK2 | |
| 504 | 1F8 | 2 | PPIK4 | |
| 506 | 1FA | 2 | PPIK8 | |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| 508 | 1FC | 11 | PPILBPSN | การกำหนดเกี่ยวกับลักษณะของเทปของผู้ใช้งาน แต่ละไบท์จะแสดงข้อมูลของ SORTOUT SORTIN 1-9 SORTWK ในรูปแบบต่อไปนี้คือ |
| | ไบท์ 1 | | 1... .... | ไม่ต้องหมุนเทปกลับก่อน OPEN (0 = หมุนกลับ) |
| | | | .1.. .... | CLOSE = UNLD |
| | | | ..1. .... | ไม่ต้องหมุนเทปกลับเมื่อ CLOSE (0 = หมุนกลับ) |
| | | | ...1 .... | เลเบลของเทปไม่มาตรฐาน |
| | | | .... 1... | ไม่มีเลเบล |
| | | | .... .1.. | ไม่มีเทปมาร์ค (Tape mark) |
| | | | .... ..XX | ไม่ได้ใช้งาน |
| | | | | สร้างโดยโมดุล RCM |
| 519 | 207 | 9 | PPIVOLS | จำนวน Volume ของที่เก็บข้อมูลนำเข้า (1 ไบท์ สำหรับแต่ละ SORTIN 1-9) |
| | | | | สร้างโดยโมดุล RCK |
| 528 | 210 | 1 | PPININS | จำนวนของแฟ้มข้อมูลนำเข้า (Logical) |
| | | | | สร้างโดยโมดุล RCB |
| 529 | 211 | 2 | PPISYS | สวิช |
| | ไบท์ 0 | | 1... .... | PPIMULT จำนวนพาร์ทิชั่นเสมือนที่ใช้งานอยู่มีมากกว่า 1 พาร์ทิชั่น |
| | | | .1.. .... | PPIINWK แฟ้มข้อมูลนำเข้ากับที่เก็บข้อมูลชั่วคราวอยู่ต่างแชนแนลกัน |
| | | | ..1. .... | PPIOUTWK แฟ้มข้อมูลผลลัพธ์กับที่เก็บข้อมูลชั่วคราวอยู่ต่างแชนแนลกัน |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| | | | ...1 .... | PPIM115 CPU แบบ 115 |
| | | | .... 1... | PPIM125 CPU แบบ 125 |
| | | | .... .1.. | PPIM135 CPU แบบ 135 |
| | | | .... ..1. | PPIM145 CPU แบบ 145 หรือใหญ่กว่า |
| | | | .... ...1 | PPIDOPT ให้พิมพ์ข้อความ (เลือกใช้ DIAG ในคำสั่ง OPTION) |
| | ไบท์ 1 | | XXXX X... | ไม่ได้ใช้งาน |
| | | | .... .1.. | PPICHKL5 ค่า $1_5$ โปรแกรมสำเร็จรูปกำหนดให้เอง |
| | | | .... ..1. | PPIWLRB ความยาวของระเบียนหรือบลอคผิดจากที่กำหนด |
| | | | .... ...1 | PPICHKL4 ค่า $1_4$ โปรแกรมสำเร็จรูปกำหนดให้เอง |
| 531 | 213 | 1 | | ไม่ได้ใช้งาน |
| 532 | 214 | 2 | PPIDELPS | ตำแหน่งของ DELBLANK สร้างโดยโมดุล RCI |
| 534 | 216 | 1 | PPIDELCH | ตัวอักษรของ DELBLANK สร้างโดยโมดุล RCI |
| 535 | 217 | 3 | PPIDIAG | โมดุลสุดท้ายที่นำเข้ามาในหน่วยความจำ (ตัวอักษร 3 ตัว) สร้างโดยโมดุล RTMG (ในเฟส 0) โมดุล RSD (เฟส 1) โมดุล RSI (เฟส 2) หรือโมดุล RSM (เฟส 3) |
| 538 | 21A | 2 | | ไม่ได้ใช้งาน |
| 540 | 21C | 4 | PPITIME | ค่าของเวลา สร้างโดยโมดุล RTMG RMC RSE RSJ RSM RSN |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| 544 | 220 | 4 | PPIBUFOF | ข่าวสารของบัฟเฟอร์ออฟเซท (Buffer offset) |
| | ไบท์ 0-1 | | | ของข้อมูลนำเข้า |
| | ไบท์ 2-3 | | | ของข้อมูลผลลัพธ์ |
| 548 | 224 | 40 | PPIPATCH | เก็บไว้ใช้งานต่าง ๆ |
| 588 | 24C | 1 | PPISW3 | สวิช |
| | | | 1... .... | PPISKSDS ผลลัพธ์เป็น VSAM KSDS |
| | | | .1.. .... | PPIESDS ผลลัพธ์เป็น VSAM ESDS |
| | | | 11.. .... | PPISVSOP ผลลัพธ์เป็น VSAM (KSDS หรือ ESDS) |
| | | | ..X. .... | ไม่ได้ใช้งาน |
| | | | ...1 .... | PPISVSIN ข้อมูลนำเข้าเป็น VSAM |
| | | | .... 1... | PPISOUT1 OUTREC ในเฟส 1 |
| | | | .... .1.. | PPISOUT3 OUTREC ในเฟส 3 |
| | | | .... ..1. | PPISSUM มีการใช้คำสั่งควบคุม SUM |
| | | | .... ...1 | PPISINOM มีการใช้คำสั่งควบคุม INCLUDE/OMIT |
| 589 | 24D | 1 | PPIVSNR | จำนวนหน่วยของแฟ้มข้อมูลนำเข้าแบบ VSAM |
| 590 | 24E | 2 | PPIVSCBL | จำนวนความยาวทั้งหมดของบลอคควบคุมของ VSAM (ACB + LXLST + RPL) ในเฟส 1 |
| 590 | 24E | 2 | PPIRPLDP | คือค่าดีสเพลสเมนท์ของ RPL จาก ACB ในเฟส 3 |
| 592 | 250 | 2 | PPIVSAR | ตำแหน่งของที่เก็บข้อมูลชั่วคราวของ VSAM และใช้สำหรับเก็บข่าวสารเมื่อเกิดความผิดเกี่ยวกับบลอคควบคุมของ VSAM |
| 596 | 254 | 4 | PPIALTTR | ตำแหน่งของตารางที่แปลคำสั่ง ALTSEQ |
| 600 | 258 | 4 | PPISUM | ตำแหน่งของโมดูลที่ทำคำสั่ง SUM |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | อธิบาย |
|---|---|---|---|---|
| 604 | 25C | 4 | PPIINCL | ตำแหน่งของโมดุลที่ทำคำสั่ง INCLUDE/OMIT |
| 608 | 260 | 4 | PPIOUTRC | ตำแหน่งตารางของ OUTREC |
| 612 | 264 | 20 | PPISYSNO | ชื่อหน่วยของแฟ้มข้อมูลที่ผู้ใช้งานบ่งเอาไว้ จะเรียงตามลำดับ คือ SORTOUT SORTIN 1-9 SORTWK 1-10 โดยใช้ 1 ไบท์สำหรับแฟ้มข้อมูลแต่ละแฟ้มดังนี้ คือ |
| | ไบท์ 0 | | PPISRTO | เป็น SYSTEM NUMBER ของ SORTOUT |
| | ไบท์ 1-9 | | PPISRT1 | เป็น SYSTEM NUMBER ของ SORTIN |
| | ไบท์ 10-20 | | PPISRTW | เป็น SYSTEM NUMBER ของ SORTWK |
| 632 | 270 | 4 | PPIADRCF | ตำแหน่งบริเวณที่เก็บข้อมูล ที่จะนำมาเรียงลำดับ เมื่อมีการกำหนด ADDROUT และมีการดึงเขตข้อมูลมารวมกัน |
| 636 | 27C | 2 | PPISW2 | สวิช |
| | ไบท์ 0 | | 1... .... | ใช้ 3330 เป็นอุปกรณ์ที่เก็บข้อมูลชั่วคราว |
| | | | .X.. .... | ไม่ได้ใช้งาน |
| | | | ..1. .... | ใช้ 3400 เป็นอุปกรณ์ที่เก็บข้อมูลชั่วคราว |
| | | | ...X .... | ไม่ได้ใช้งาน |
| | | | .... 1... | ใช้ 3340 เป็นอุปกรณ์ที่เก็บข้อมูลชั่วคราว |
| | | | .... .1.. | ใช้ 3340 เป็นอุปกรณ์ที่จะใช้ทำ CHKPT |
| | | | .... ..X. | ไม่ได้ใช้งาน |
| | | | .... ...1 | NODUMP |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| | ไบท์ 1 | | 1... .... | ROUTE = LOG |
| | | | .1.. .... | ROUTE = LST |
| | | | ..1. .... | ERASE |
| | | | ...1 .... | DUMP |
| | | | .... 1... | ไม่ได้ใส่ OPERAND ของ WORK |
| | | | .... .1.. | SYSLST = SYSLOG |
| | | | .... ..1. | SYSLST = UA |
| | | | .... ...1 | แฟ้มข้อมูลของผลลัพธ์มีความยาวไม่คงที่ และไม่บลอค (RCK) |
| 638 | 27E | 2 | PPICYLZ | จำนวนของแทรคใน 1 CYLINDER สำหรับที่เก็บข้อมูลชั่วคราว ที่เป็นจานแม่เหล็ก และจะต้องเป็นจำนวนคู่เสมอเมื่อใช้แบบแยก CYLINDER สร้างโดยโมดูลที่คำนวณค่า B/G |
| 640 | 280 | 1 | PPISW4 | สวิช |
| | | | 1... .... | PPICCWT สามารถแปล CCW ได้ |
| | | | .1.. .... | PPIVRTM ใช้พาร์ทิชั่นเสมือน |
| | | | ..1. .... | PPICCWT1 เฟส 1 ใช้ตำแหน่งของ I/O เป็นตำแหน่งจริง |
| | | | ...1 .... | PPICCWT2 เฟส 2 ใช้ตำแหน่งของ I/O เป็นตำแหน่งจริง |
| | | | .... 1... | PPICCWT3 เฟส 3 ใช้ตำแหน่งของ I/O เป็นตำแหน่งจริง |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| | | | .... .1.. | PPIVIRT มีการกำหนด VIRT เป็นพารามิเตอร์ของพารามิเตอร์ STORAGE |
| | | | .... ..1. | PPISTRG มีการให้ค่าพารามิเตอร์ STORAGE |
| | | | .... ...1 | PPISIZX กำหนดค่าพารามิเตอร์ SIZE ในคำสั่ง EXEC |
| 641 | 281 | 1 | PPISW5 | สวิช |
| | | | 1... .... | PPISTOLI ยอมทำงานเมื่อการเปิดแฟ้มข้อมูลนำเข้า ไม่ถูก (VSAM) |
| | | | .1.. .... | PPISTOLO ยอมทำงานเมื่อการเปิดแฟ้มข้อมูลผลลัพธ์ ไม่ถูก (VSAM) |
| | | | ..1. .... | PPISRUSE กำหนด REUSE |
| | | | ...1 .... | PPISRRDS กำหนด RRDS |
| | | | .... 1... | PPISRPSI RPS บนอุปกรณ์นำข้อมูลเข้า |
| | | | .... .1.. | PPISRPSW RPS บนอุปกรณ์ที่เก็บข้อมูลชั่วคราว |
| | | | .... ..1. | PPISRPSO RPS บนอุปกรณ์ผลลัพธ์ |
| | | | .... ...X | ไม่ได้ใช้งาน |
| 642 | 282 | 2 | | ไม่ได้ใช้งาน |
| 644 | 284 | 3 | PPISPGN2 | ตำแหน่งของไบท์ต่อไปในบริเวณที่เก็บข้อมูลชั่วคราวของเฟสที่เปลี่ยนแปลงไม่ได้ |
| 648 | 288 | 4 | PPIIDAL | ตัวชี้ชี้ไปยังรายการของ IDAL |
| 652 | 28 | 4 | PPIWIDAL | ตำแหน่งของรายการที่เก็บค่าตัวชี้ของ IDAL ที่ชี้ไปที่บัฟเฟอร์ ที่เพิ่งจะนำข้อมูลออกจากบลอค |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| 656 | 290 | 4 | PPIFIDAL | ตำแหน่งของรายการที่เก็บค่าตัวชี้ของ IDAL โดยเป็นไปตาม Floating Buffer |
| 660 | 294 | 4 | PPIDADDR | ตัวชี้ชี้ไปยังตารางตำแหน่งของจานแม่เหล็ก |
| 664 | 298 | 4 | PPIACTTR | ตำแหน่งของตารางแปลรหัส ASCII |
| 668 | 296 | 4 | PPITAVLR | ขนาดของพาร์ทิชั่นจริงที่เชื่อมต่อพาร์ทิชั่นเสมือน |
| 672 | 2A0 | 2 | PPILSUM | ขนาดความยาวของส่วนโปรแกรมที่ทำหน้าที่เกี่ยวกับคำสั่ง SUM |
| 674 | 2A2 | 2 | PPILINCL | ขนาดความยาวของส่วนโปรแกรมที่ทำหน้าที่เกี่ยวกับคำสั่ง INCLUDE/OMIT |
| 676 | 2A4 | 2 | PPILOTRC | ความยาวของตาราง OUTREC |
| 678 | 2A6 | 6 | | ข่าวสารของ ALTSEQ |
| 678 | 2A6 | 2 | PPINAQ | จำนวนครั้งของการเปลี่ยนแปลง ที่ระบุในพารามิเตอร์ CODE ของคำสั่ง ALTSEQ |
| 680 | 2A8 | 4 | PPIAQ | ตารางของคำสั่ง ALTSEQ ที่ได้ตัดให้สั้นแล้วในดิกชันนารี |
| 678 | 2A6 | 10 | | เป็นข่าวสารของ IDAL (ในเฟส 2,3) |
| 678 | 2A6 | 2 | PPIIDLLN | ความยาวของ IDAL |
| 680 | 2A8 | 4 | PPIIDLO1 | ตำแหน่ง IDAL ของบัฟเฟอร์ผลลัพธ์ที่ 1 |
| 684 | 2AC | 4 | PPIIDLO2 | ตำแหน่ง IDAL ของบัฟเฟอร์ผลลัพธ์บัฟเฟอร์ที่ 2 |
| 684 | 2AC | 4 | PPIEDICT | ในเฟส 0 คือตำแหน่งของไบท์ต่อไปที่สามารถใช้งานได้<br>ในเฟส 2 คือตำแหน่งสุดท้ายของบริเวณที่เปลี่ยนแปลงไม่ได้ (Fixable Area) ในหน่วยความจำ |

ดีสเพลสเมนท์

| ฐาน 10 | ฐาน 16 | ขนาด | ชื่อเขตข้อมูล | คำอธิบาย |
|---|---|---|---|---|
| | | | | สร้างโดยโมดุล ASG |
| 688 | 2B0 | 1 | PPILABAD | ตำแหน่งของเลเบล ใน SORTOUTVTOC |
| | | | | สร้างโดยโมดุล RCC |
| 694 | 2B6 | 1 | PPIKF | คงที่ |
| 695 | 2B7 | 1 | | ไม่ใช้งาน |
| 696 | 2B8 | 4 | PPISECTV | ตัวชี้ที่ชี้ไปยังตารางที่เก็บค่าของเซ็คเตอร์ของ RPS |
| 700 | 2BC | 4 | PPINNMS | ตารางชื่อของแฟ้มข้อมูลนำเข้า |
| 704 | 2C0 | 8 | PPIOUTNM | ชื่อแฟ้มข้อมูลผลลัพธ์ที่โปรแกรมสำเร็จรูปกำหนดให้ |

ภาคผนวก ง.

การทำงานของโมดุลที่สำคัญ

| โมดุล | เริ่มทำงาน | หน้าที่ | เรียกโมดุล | จบการทำงาน |
|---|---|---|---|---|
| RTMG | รับการควบคุมจาก โปรแกรมระบบหรือจาก โปรแกรมภาษาชั้นสูง | 1.ตรวจสอบการ เรียกใช้โปรแกรม สำเร็จรูป<br>2.เก็บค่ารีจีสเตอร์ ทั้งหมด<br>3.อ่านและพิมพ์ข่าว สารของเฟส 0 | 1.RCL<br>2.RCM<br>3.RCB<br>4.RCI<br>5.RCJ<br>6.RCK<br>7.RCO<br>8.RCQ<br>9.RCP<br>10.RCS<br>11.RCD<br>12.RCC<br>13.RCE หรือ RCF หรือ RCH<br>14.RCG | โดยปกติจะส่งการควบคุม ให้เฟส 1 นอกจากมีการ กำหนดว่าเป็น CALAREA จะส่งการทำงานกลับไป ยังโปรแกรมระบบ หรือ โปรแกรมผู้ใช้งาน |
| RCA | โดย RCL | 1.อ่านคำสั่งควบคุม จาก SYSIPT ทำให้สิ้นลงและ นำไปเก็บใน พจนานุกรม | 1.RCM | ส่งการควบคุมไปให้ RTMG |

| โมดุล | เริ่มทำงาน | หน้าที่ | เรียกโมดุล | จบการทำงาน |
|---|---|---|---|---|
| | | 2.พิมพ์คำสั่งควบคุมถ้าใช้งานกำหนด | | |
| RCD | โดย RTMG | 1.ตัดสินว่าจำเป็นจะต้องดึงเขตข้อมูลหรือไม่<br>2.คำนวณส่วนโปรแกรมที่ใช้ในการดึงเขตข้อมูล<br>3.คำนวณขนาดของที่เก็บข้อมูลในหน่วยความจำให้เหมาะสมกับความยาวของระเบียน | - | ส่งการควบคุมกลับไปให้ RTMG |
| RCC | โดย RTMG | 1.ตรวจสอบเนื้อที่ในหน่วยความจำและตรวจสอบอุปกรณ์<br>2.เปิดแฟ้มข้อมูล SORTWK ของจานแม่เหล็ก | 1.RCN | ส่งการควบคุมกลับไปให้ RTMG |
| RCEหรือ RCFหรือ RCH | โดย RTMG | 1.คำนวณค่า B (Blocking factor) | - | ส่งการควบคุมกลับไปให้ RTMG |

| โมดุล | เริ่มทำงาน | หน้าที่ | เรียกโมดุล | จบการทำงาน |
|---|---|---|---|---|
| RPA หรือ RPB | โดยโมดุล RBA หรือ RBB หรือRBC | เขียนข้อมูลจากบัฟเฟอร์ ผลลัพธ์ลงบนแฟ้มข้อมูลชั่วคราวโดยใช้แมคโคร EXCP และ WAIT | โมดุล RAA หรือ RAB | ส่งการควบคุมไปให้โมดุลRBA หรือ RBB หรือ RBC |
| RSE | โดยโมดุล RAA หรือ RAB | คำนวณจำนวนระเบียนทั้งหมดที่ได้จากการอ่านเพิ่มและลด | โมดุล RMC | 1.ส่งการควบคุมไปให้โมดุล AAC หรือ AAD<br>2.ถ้าไม่ต้องใช้เฟส2 ส่งไปให้โมดุล RSJ |
| RAC หรือ RAD | โดยโมดุล RSI | กระจายสตริงบนที่เก็บข้อมูลชั่วคราว | 1.RSG หรือ RSH ถ้ากำหนดCHKPT | 1.ครั้งแรกส่งการควบคุมไปให้ RGD หรือ RGE<br>2.ครั้งต่อมาส่งการควบคุมให้โมดุลที่เรียก<br>3.เมื่อจบการทำงานส่งการควบคุมให้โมดุล RSJ |
| RSI | โดยโมดุล AAB หรือ AAD | นำโมดุลต่าง ๆ ในเฟส 2 เข้ามาในหน่วยความจำ | ทุกโมดุลที่ใช้ในการทำงานของเฟส2 | ส่งการควบคุมไปให้ RAC หรือ RAD |

| โมดุล | เริ่มทำงาน | หน้าที่ | เรียกโมดุล | จบการทำงาน |
|---|---|---|---|---|
| | | 2.ขนาดหน่วยความจำภายใน<br>3.ค่าเมิร์จออร์เดอร์<br>4.จำนวนบัฟเฟอร์<br>5.ค่าระเบียนสูงสุดที่เรียงลำดับได้ | | |
| RCG | โดย RTMG | สร้างรหัสสำหรับส่วนโปรแกรมที่เหมาะสมที่ใช้ดึงเขตข้อมูล | - | ส่งการควบคุมกลับไปให้ RTMG |
| RAAหรือ RAB | 1.โดยโมดุล RPA หรือ RPB<br>2.โดยโมดุล ROA | ตัดสินว่าจะบันทึกสตริงลงบน SORTWK ใด | - | 1.ส่งการควบคุมไปให้ RPA หรือ RPB<br>2.RSE |
| RSD | โดยโมดุล AAA หรือ AAB | นำส่วนโปรแกรมอื่นเข้ามาในหน่วยความจำ | โมดุลที่กำหนดการทำงานของทุกโมดุลที่ใช้ในการทำงานของเฟส 1 | โมดุล ROA |
| ASA | โดยโมดุล RSD | สร้างเนื้อที่ในหน่วยความจำ เช่น บัฟเฟอร์ข้อมูลนำเข้าและผลลัพธ์ | - | ส่งการควบคุมไปให้ RSD |
| ROA | โดยโมดุล RSD | เรียงลำดับข้อมูลโดยการสร้างสตริง | 1.RDA หรือ RDB หรือ RDC | ส่งการควบคุมไปให้ RAA หรือ RAB |

| โมดุล | เริ่มทำงาน | หน้าที่ | เรียกโมดุล | จบการทำงาน |
|---|---|---|---|---|
| ASF หรือ ASG หรือ ASH | โดยโมดุล RSI | จัดเตรียมเนื้อที่ในหน่วยความจำ เพื่อการทำงานของเฟส 2 | - | ส่งการควบคุมไปให้โมดุล RSI |
| ROB หรือ ROD หรือ ROF | 1.โดยโมดุล RGD หรือ RGE<br>2.โดยโมดุล RBF หรือ RBH หรือ RBV | รวมแฟ้มข้อมูล | เรียกส่วนโปรแกรมที่ทำงานเกี่ยวกับการดึงเขตข้อมูลและคำสั่ง SUM | ส่งการควบคุมไปให้โมดุล RBF หรือRBH หรือ RBV |
| RBF หรือ RBH หรือ RBV | โดยโมดุล ROB หรือ ROD หรือ ROF | นำข้อมูลออกจากบลอคหรือใส่ข้อมูลในบลอค | 1.โมดุล RGD หรือ RGE<br>2.โมดุล RPC หรือ RPD | ส่งการควบคุมไปให้โมดุล ROB หรือ ROD หรือ ROF |
| RGD หรือ RGE | 1.โดยโมดุล RAC หรือ RAD<br>2.โดยโมดุล RBF หรือ RBH หรือ RBV | อ่านข้อมูลจาก SORTWK โดยใช้แมคโคร EXCP และ WAIT | - | 1.ส่งการควบคุมไปให้โมดุล RBF หรือ RBH หรือ RBV<br>2.โมดุล ROB หรือ ROD หรือ ROF |
| RPCหรือ RPD | โดยโมดุล RBF หรือ RBH หรือ RBV | บันทึกข้อมูลลงบนที่เก็บข้อมูลชั่วคราว โดยใช้แมคโคร EXCP และ WAIT | 1.RPC เรียกโมดุล RAC<br>2.RPD เรียกโมดุล RAD | 1.โดยปกติส่งการควบคุมไปให้โมดุล RBF หรือ RBH หรือ RBV<br>2.เมื่อจบสตริงส่งการควบคุมไปให้RAC หรือ RAD |

| โมดุล | เริ่มทำงาน | หน้าที่ | เรียกโมดุล | จบการทำงาน |
|---|---|---|---|---|
| RSG หรือ RSH | โดยโมดุล RAC หรือ RAD | สร้างตารางสำหรับการทำ CHKPT | - | ส่งการควบคุมไปให้ RAC หรือ RAD |
| RSP | โดยโมดุล RAC หรือ RAD | เริ่มการทำงานใหม่เมื่อจบแฟ้มข้อมูลนำเข้าแฟ้มใดแฟ้มหนึ่ง | - | ส่งการควบคุมไปให้ RAC หรือ RAD |
| RSQ | โดยโมดุล RAC หรือ RAD | เริ่มการทำงานของรอบใหม่ | - | ส่งการควบคุมไปให้โมดุล RAC หรือRAD |
| RSJ | โดยโมดุล RAC หรือ RAD | คำนวณจำนวนระเบียนทั้งหมดที่อ่านเข้ามาเพิ่มและลด ตลอดการทำงานของเฟส2 พิมพ์ข้อความ | โมดุล RMC | ส่งการควบคุมไปให้ RSM |
| RSM | โดยโมดุล RSJ | นำส่วนโปรแกรมต่าง ๆ ในเฟส 3 เข้ามาในหน่วยความจำ | เรียกโมดุลที่จำเป็นในเฟส 3 ทุกโมดุล | ส่งการควบคุมไปให้ ASP |
| ASK หรือ ASL หรือ ASP | โดยโมดุล RSM | เตรียมเนื้อที่ในหน่วยความจำ | - | ส่งการควบคุมไปให้ RSM |
| | โดยโมดุล RSM | เริ่มการเรียงลำดับข้อมูล | - | ส่งการควบคุมไปให้ ROC หรือ ROE หรือ ROG |

| โมดุล | เริ่มทำงาน | หน้าที่ | เรียกโมดุล | จบการทำงาน |
|---|---|---|---|---|
| ROC หรือ ROE หรือ ROG | 1.โดยโมดุล ASP<br>2.โดยโมดุล RBGหรือ RBI หรือ RBJ | รวมแฟ้มข้อมูล | เรียกส่วนโปรแกรมที่ทำงานเกี่ยวกับคำสั่ง SUM หรือส่วนโปรแกรมที่ทำหน้าที่ดึงเขตข้อมูล | ส่งการควบคุมไปให้ RBG หรือ RBI หรือ RBJ |
| RBG หรือ RBI หรือ RBJ | โดยโมดุล ROC หรือ ROE หรือ ROG | นำข้อมูลเข้าหรือออกจากบลอค | 1.โมดุล RGF หรือ RGG หรือ RGH หรือ RGI<br>2.โมดุล RPE หรือ RPF หรือ RPG<br>3.ส่วนโปรแกรมที่ทำคำสั่ง OUTREC | ส่งการควบคุมไปให้ ROC หรือ ROE หรือ ROG |
| RGF หรือ RGG หรือ RGH หรือ RGI | โดยโมดุล RBG หรือ RBI หรือ RBJ | อ่านข้อมูลจาก SORTWK โดยใช้แมคโคร EXCP และ WAIT ถ้าเป็น VSAM อ่านโดยใช้แมคโคร GET | 1.ส่วนโปรแกรม E31 ถ้ามีหลาย volume และเลเบลไม่มาตรฐาน<br>2.ส่วนโปรแกรม E38 ถ้าระเบียนมีความยาวไม่เท่ากับที่กำหนด | ส่งการควบคุมไปให้ RBG หรือ RBI หรือ RBJ |

| โมดุล | เริ่มทำงาน | หน้าที่ | เรียกโมดุล | จบการทำงาน |
|---|---|---|---|---|
| | | | 2. RBA หรือ RBB หรือ RBC<br>3. ส่วนโปรแกรมที่ทำหน้าที่ดึงเขตข้อมูล | |
| RDAหรือ RDB หรือ RDC | โดยโมดุล ROA | นำข้อมูลออกจากบลอค | 1.RGA หรือ RGB หรือ RGI<br>2.RBDถ้าระเบียนยาวกว่า 256 ไบท์<br>3.ส่วนโปรแกรมที่ทำคำสั่ง INCLUDE/OMIT | ส่งการควบคุมไปให้ ROA |
| RGA หรือ RGB หรือ RGI | โดยโมดุล RDA หรือ RDB หรือ RDC | อ่านข้อมูลโดยคำสั่ง EXCP และ WAIT ถ้าเป็น VSAM อ่านโดยคำสั่งGET | เรียกโมดุลที่ทำงานเกี่ยวกับ E18 ถ้าพบว่าการอ่านเกิดความผิดพลาด | 1.ถ้าการทำงานปกติส่งการควบคุมกลับไปให้ RBA หรือ RBB หรือ RBC<br>2.เมื่อเกิดความผิดพลาดส่งให้ E18 หรือ RAA หรือ RAB |
| RBA หรือ RBB หรือ RBC | โดยโมดุล ROA | บลอคข้อมูลโดยเคลื่อนวินเนอร์จาก RSA ไปเก็บในบัฟเฟอร์ผลลัพธ์ | 1.RBD ถ้าระเบียนยาวกว่า 256ไบท์<br>2.RPA หรือ RPB | ส่งการควบคุมไปให้ ROA |

| โมดุล | เริ่มทำงาน | หน้าที่ | เรียกโมดุล | จบการทำงาน |
|---|---|---|---|---|
| RPE หรือ RPF หรือ RPG | โดยโมดุล RBG หรือ RBI หรือ RBJ | บันทึกข้อมูลลงบนแฟ้มข้อมูลผลลัพธ์ | ส่วนโปรแกรม E39 หรือรับพาสเวิร์ดของ VSAM | ส่งการควบคุมไปให้ RBG หรือ RBIหรือ RBJ |
| RSN | 1.โดยโมดุล RBG หรือ RBJหรือRBI<br>2.โดยโมดุล AAA หรือ AAB ใน เฟส 1 | 1.คำนวณจำนวนระเบียนทั้งหมดที่เข้ามาประมวลผลในเฟส 3<br>2.เลิกการทำงานโดยแมคโคร EOJ | 1.RMC และ RMD<br>2.RSR | ส่งการควบคุมไปให้โปรแกรมระบบหรือโปรแกรมผู้ใช้งาน |
| RMC และ RMD | โดยโมดุลใดโมดุลหนึ่งในเฟส3 | พิมพ์ข้อความให้ผู้ใช้งาน | - | ส่งการควบคุมไปให้โมดุลที่เรียกใช้ |
| RSR | โดยโมดุล RSN | ลบข้อมูลในกรณีที่ผู้ใช้งานกำหนด | - | ส่งการควบคุมไปให้ RSN |

ภาคผนวก จ.

วิธีการเรียงลำดับที่ใช้ในโปรแกรมสำเร็จรูป

การเรียงลำดับในโปรแกรมสำเร็จรูปสามารถเรียงลำดับได้ 12 เขตข้อมูล ซึ่งแต่ละเขตข้อมูลนี้จะต้องมีตำแหน่งตรงกันในแต่ละระเบียน เขตข้อมูลที่จะเรียงลำดับเป็นเขตข้อมูลแรกเรียกว่าเขตข้อมูลหลัก ส่วนเขตข้อมูลอื่นจะเรียงลำดับก่อนหลังตามความสำคัญ

โปรแกรมการเรียงลำดับจะถือว่าระเบียนนำเข้าอยู่กันแบบสุ่ม นอกเสียจากว่าจะมีการกำหนด PRESEQ สำหรับการเรียงลำดับที่ใช้จานแม่เหล็กเป็นที่เก็บข้อมูลชั่วคราว ซึ่งในกรณีโปรแกรมจะถือเสมือนว่า ระเบียนต่าง ๆ ใกล้จะอยู่ในลำดับที่ต้องการแล้ว ในการเรียงลำดับโปรแกรมสำเร็จรูปจะใช้

1. วิธีการเรียงลำดับเพียงวิธีเดียว คือแบบเลือกแทนที่
2. เลือกวิธีการกระจายสตริงวิธีใดวิธีหนึ่งใน 3 วิธี คือ
   - 2.1 แบบ Polyphase
   - 2.2 แบบ Interleave
   - 2.3 แบบ Split Cylinder

   การจะใช้วิธีใดขึ้นอยู่กับประเภทของบริเวณที่ใช้เป็นที่เก็บข้อมูลชั่วคราว และจำนวนบริเวณที่กำหนดให้โปรแกรมใช้
3. การรวมแฟ้มข้อมูลเพียงวิธีเดียว คือแบบไบนารี่

<u>วิธีการเรียงลำดับ</u>

โปรแกรมสำเร็จรูปจะใช้วิธีการเรียงลำดับแบบเลือกแทนที่ โดยวิธีการดังนี้คือ

1. อ่านข้อมูลที่จะนำมาเรียงลำดับจากแฟ้มข้อมูล นำเข้า 1 บลอค ไปเก็บในบัฟเฟอร์ข้อมูลนำเข้า

2. เคลื่อนระเบียน 1 ระเบียนจากบัฟเฟอร์ข้อมูล นำเข้าไปเก็บไว้ที่บริเวณที่เก็บระเบียน (RSA หรือ Record Storage Area) เมื่อระเบียนในบัฟเฟอร์ข้อมูลนำเข้าหมด ก็จะนำบลอคใหม่จากแฟ้มข้อมูลนำเข้า เข้ามาใหม่อีก

3. ใช้การเปรียบเทียบโดยการสร้างทรี เพื่อจะหาว่าระเบียนใด จะเป็นระเบียนต่อไปที่จะนำไปใส่ในสตริง ระเบียนที่ถูกเลือกจะเรียกว่า วินเนอร์ และระเบียนอื่นเรียกว่า ลูสเซอร์ วินเนอร์จะเป็นตัวที่มีค่าน้อยที่สุด ถ้าเป็นการเรียงลำดับจากน้อยไปมาก และเป็นตัวที่มีค่ามากที่สุด ถ้าเป็นการเรียงลำดับจากมากไปน้อย

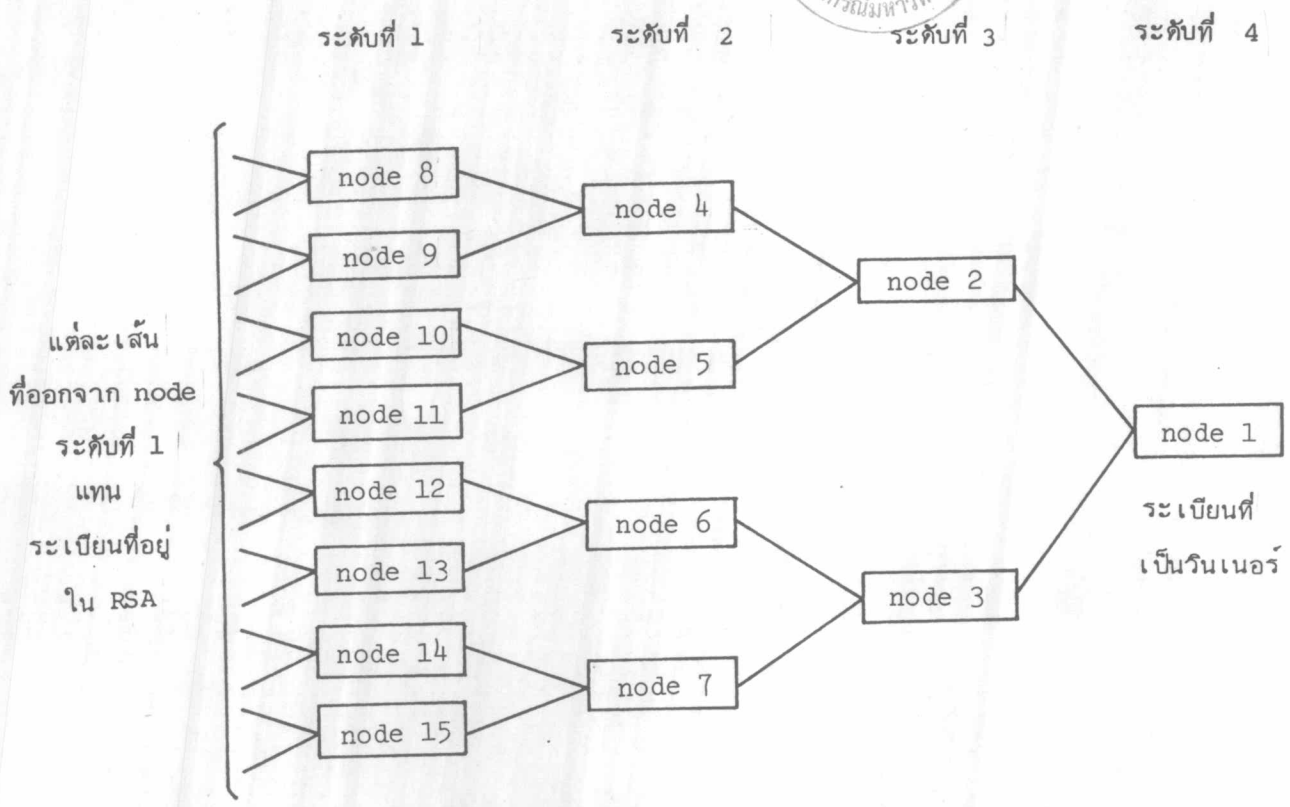
ถ้าใช้เทปเป็นที่เก็บข้อมูลชั่วคราว บางสตริงจะเรียงลำดับแบบตรงข้ามกับลำดับที่ต้องการ เพื่อประโยชน์ในการอ่านเทปถอยหลัง

4. เคลื่อนระเบียนใหม่จากบัฟเฟอร์ข้อมูลนำเข้า เข้ามาใน RSA และเปรียบเทียบกับวินเนอร์ที่หาได้ ถ้าระเบียนที่เข้ามาใหม่ไม่ได้อยู่ในสตริงเดียวกับวินเนอร์ตัวที่หาได้ ระเบียนนั้นก็จะใช้ Flag เพื่อบ่งว่าจะใช้เปรียบเทียบตอนสร้างสตริงต่อไป

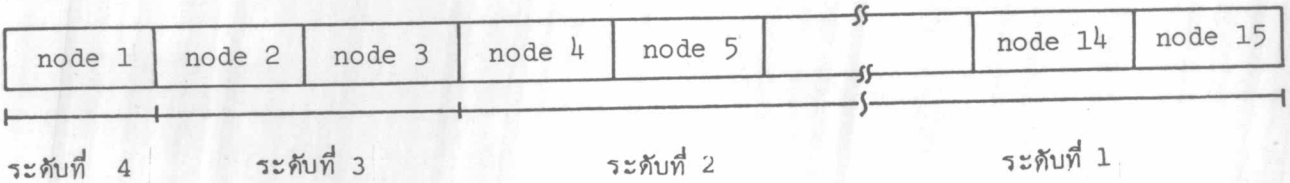
5. เคลื่อนระเบียนที่เป็นวินเนอร์ไปเก็บในบัฟเฟอร์ผลลัพธ์ เมื่อบัฟเฟอร์ผลลัพธ์เต็มก็จะนำไปเก็บในอุปกรณ์ที่ใช้เป็นที่เก็บข้อมูลชั่วคราว วิธีการนี้จะทำต่อไปเรื่อย ๆ จนกระทั่งหมดระเบียนในแฟ้มข้อมูลนำเข้า

โครงสร้างของทรีที่ใช้ในการเรียงลำดับแบบเลือกแทนที่นี้ จะอยู่ในหน่วยความจำ และเรียกว่า Node ซึ่งแต่ละ Node จะเก็บข่าวสารเกี่ยวกับการเปรียบเทียบระเบียน 2 ระเบียนใน RSA Node ต่าง ๆ จะถูกแบ่งเป็นกลุ่มตามระดับ (ดังรูปที่ จ.1)

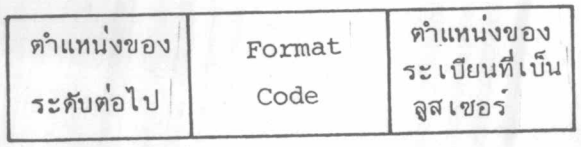
ระเบียนที่จะนำมาเรียงลำดับนั้น จะใช้การเปรียบเทียบทีละคู่ก่อน แล้วเชื่อมแต่ละคู่ที่มาเปรียบเทียบกันนั้น ด้วยตำแหน่งของ Node ในระดับที่ 1 คือระเบียนแต่ละคู่จะเปรียบเทียบกันเพื่อหาวินเนอร์ เมื่อได้วินเนอร์แล้ว ตำแหน่งของลูสเซอร์ จะถูกเก็บไว้ใน Node ที่ระดับที่ 1 แล้ววินเนอร์ในระดับที่ 1 จะถูกนำไปเปรียบเทียบกับวินเนอร์ในระดับที่ 2 ต่อไป




แผนผังของโครงสร้างทรีแบบ Conceptual




แผนผังแสดงโครงสร้างของทรีในหน่วยความจำ




โครงสร้างของ node (3 เวิร์ด)

รูปที่ จ. 1 แสดงโครงสร้างของการจัดทรี

การเปรียบเทียบในระดับที่ 2 นี้ ก็จะได้ระเบียนที่เป็นลูสเซอร์และตำแหน่งของลูสเซอร์ก็จะถูกเก็บเอาไว้ที่ Node ในระดับที่ 2 และวินเนอร์ในระดับที่ 2 ก็จะถูกนำไปเปรียบเทียบกับวินเนอร์ในระดับที่ 3 ต่อไป การเปรียบเทียบจะทำต่อไปเรื่อย ๆ จนกระทั่งถึงระดับสุดท้าย ก็จะได้วินเนอร์เพียงตัวเดียว

ในบริเวณ RSA นี้ จะเตรียมที่เพิ่มจากจำนวนระเบียนที่จะต้องเรียงลำดับในทรีอีก 1 ที่ เนื่องจากว่าหลังจากที่เลือกวินเนอร์ตัวสุดท้ายได้แล้ว วินเนอร์ตัวนี้ยังคงเก็บอยู่ใน RSA เพื่อคอยเปรียบเทียบกับระเบียนที่จะอ่านเข้ามาใหม่ว่า ระเบียนที่อ่านเข้ามาใหม่นั้น จะอยู่ในสตริงเดียวกันหรือไม่ ถ้าระเบียนที่เข้ามาใหม่อยู่ในสตริงเดียวกัน ก็จะเปรียบเทียบด้วยวิธีการเดิม แต่ถ้าระเบียนใหม่ไม่ได้อยู่ในสตริงเดียวกัน ตำแหน่งของระเบียนใหม่ ก็จะใช้ Flag เพื่อเก็บไว้เปรียบเทียบกับสตริงตัวต่อไป ระเบียนที่คงอยู่ในทรี ก็จะเปรียบเทียบกันเฉพาะระเบียนที่ไม่มี Flag เท่านั้น ระเบียนที่ถูก Flag จะถือว่าเป็นลูสเซอร์สำหรับทุก Node ในทรี

ระเบียนใหม่ก็จะถูกเปรียบเทียบกับลูสเซอร์ในระดับที่ 1 โดยหาตำแหน่งของลูสเซอร์ตัวนี้จาก Node ในระดับที่ 1 การเปรียบเทียบครั้งนี้ก็จะทำให้ได้วินเนอร์และลูสเซอร์ เก็บตำแหน่งของลูสเซอร์ใน Node และเปรียบเทียบวินเนอร์กับลูสเซอร์ในระดับที่ 2

การเปรียบเทียบจะทำไปจนกระทั่งได้วินเนอร์ 1 ตัว ซึ่งในขณะนี้วินเนอร์ตัวก่อนถูกเคลื่อนไปเก็บในบัฟเฟอร์ผลลัพธ์แล้ว และตำแหน่งที่เคยอยู่ก็ว่างลง ดังนั้นจึงอ่านระเบียนใหม่เข้ามา และวิธีการก็จะทำดังกล่าวมาแล้วอีก

ในเฟส 0 จะมีส่วนของโปรแกรมซึ่งทำหน้าที่ในการคำนวณหาค่าความยาวสูงสุดของสตริง ที่เหมาะสมสำหรับการใช้จานแม่เหล็กเป็นที่เก็บข้อมูลชั่วคราว และในเฟส 1 เมื่อเริ่มจัดทรี สตริงจะจบลงเมื่อมีความยาวเท่ากับความยาวสูงสุดแล้ว แต่ถ้าใช้เทปเป็นที่เก็บข้อมูลชั่วคราว สตริงจะมีความยาวไม่จำกัด จนกระทั่งหมดระเบียนที่เหมาะที่จะอยู่ในสตริงเดียวกัน เมื่อสตริงหนึ่งจบก็เริ่มเรียงลำดับสตริงใหม่ต่อไป จนกระทั่งหมดข้อมูลในแฟ้มข้อมูลนำเข้า

รูปที่ จ.2 เป็นตัวอย่างการเรียงลำดับแบบเลือกแทนที่ ซึ่งมีขั้นตอนดังนี้คือ

1. อ่านข้อมูลนำเข้า 1 บล็อคจากแฟ้มข้อมูลนำเข้า

2. เคลื่อนระเบียน (8) จากบัฟเฟอร์ข้อมูลนำเข้าไปใส่ในบริเวณ RSA ซึ่งในขณะนี้ไม่มีระเบียนที่จะเปรียบเทียบด้วย นำตำแหน่งของระเบียน (8) ไปเก็บใน node ในทรี และเคลื่อนระเบียนต่อไป (9) มาเก็บใน RSA

3. เปรียบเทียบ (8) กับ (9) สมมุติต้องการเรียบลำดับจากน้อยไปมาก ดังนั้น (8) คือ วินเนอร์ในระดับที่ 1

4. เคลื่อนระเบียนต่อไป (13) เข้ามาในบริเวณ RSA ไม่มีระเบียนใดใน node ที่จะนำมาเปรียบเทียบกับ (13) ดังนั้นจึงนำตำแหน่งของ (13) ไปเก็บที่ node ในทรี และเคลื่อนระเบียนต่อไป (5) เข้ามาในบริเวณ RSA

5. เปรียบเทียบ (13) กับ (5) จะได้ว่า (5) ก็คือวินเนอร์ในระดับที่ 1 เปรียบเทียบวินเนอร์ในระดับที่ 1 คือ (5) กับ (8) จะได้ว่า (5) คือวินเนอร์ของระดับที่ 2 และขณะนี้โครงสร้างของทรีมีข้อมูลอยู่เต็มแล้ว

6. เคลื่อนระเบียนต่อไป (4) เข้ามาในบริเวณ RSA ระเบียนนี้จะทำให้ที่ว่างใน RSA เต็มพอดี เปรียบเทียบ (4) กับ (5) ระเบียน (4) จะถูก Flag (4*) เนื่องจากว่า (4) มีค่าน้อยกว่า (5) ดังนั้นจึงไม่เหมาะที่จะอยู่สตริงเดียวกับ (5)

7. เปรียบเทียบ (4*) กับ (13) จะได้ว่า (13) จะเป็นวินเนอร์ในระดับที่ 1 เพราะระเบียน (4) ถูก Flag นำระเบียน (13) ไปเปรียบเทียบกับระเบียน (8) จะได้ว่า (8) เป็นวินเนอร์ในระดับที่ 2

8. เคลื่อนวินเนอร์ตัวที่แล้ว คือระเบียน (5) ไปเก็บในบัฟเฟอร์ผลลัพธ์

9. เปลี่ยนตำแหน่งของ (4*) ไปแทนตำแหน่งของ (5) ในทรี

10. เคลื่อนระเบียนต่อไป (7) จาก RSA เข้าไปเก็บในบริเวณที่ระเบียน (5) เคยอยู่ นำระเบียน (7) เปรียบเทียบกับวินเนอร์ คือ (8) ระเบียน (7) จะถูก Flag

11. เปรียบเทียบ (7*) กับ (9) ระเบียน (9) จะเป็นวินเนอร์ในระดับที่ 1 เนื่องจากระเบียน (7*) ถูก Flag ไว้ เปรียบเทียบ (9) กับ (13) จะได้ (9) เป็นวินเนอร์ในระดับที่ 2

12. เคลื่อนวินเนอร์ตัวที่แล้ว (8) ไปเก็บในบัฟเฟอร์ผลลัพธ์

13. เคลื่อนระเบียนต่อไป (6) จากบัฟเฟอร์ผลลัพธ์เข้ามาเก็บในบริเวณ RSA โดยเก็บแทนที่ระเบียน (8) เปรียบเทียบ (6) กับ (9) ระเบียน (6) จะถูก Flag เพราะมีค่าน้อยกว่า (9)

14. เปรียบเทียบ (6*) กับ (7*) จะได้ (6*) เป็นวินเนอร์ในระดับที่ 1 เปรียบเทียบ (6*) กับ (13) จะได้ (13) เป็นวินเนอร์ในระดับที่ 2

15. เคลื่อนวินเนอร์ตัวที่แล้ว (9) ไปเก็บในบัฟเฟอร์ผลลัพธ์

16. เคลื่อนระเบียนต่อไป (10) จากบัฟเฟอร์ข้อมูลนำเข้ามาเก็บใน RSA โดยเก็บแทนที่ (9) เปรียบเทียบ (10) กับ (13) ระเบียน (10) จะถูก Flag

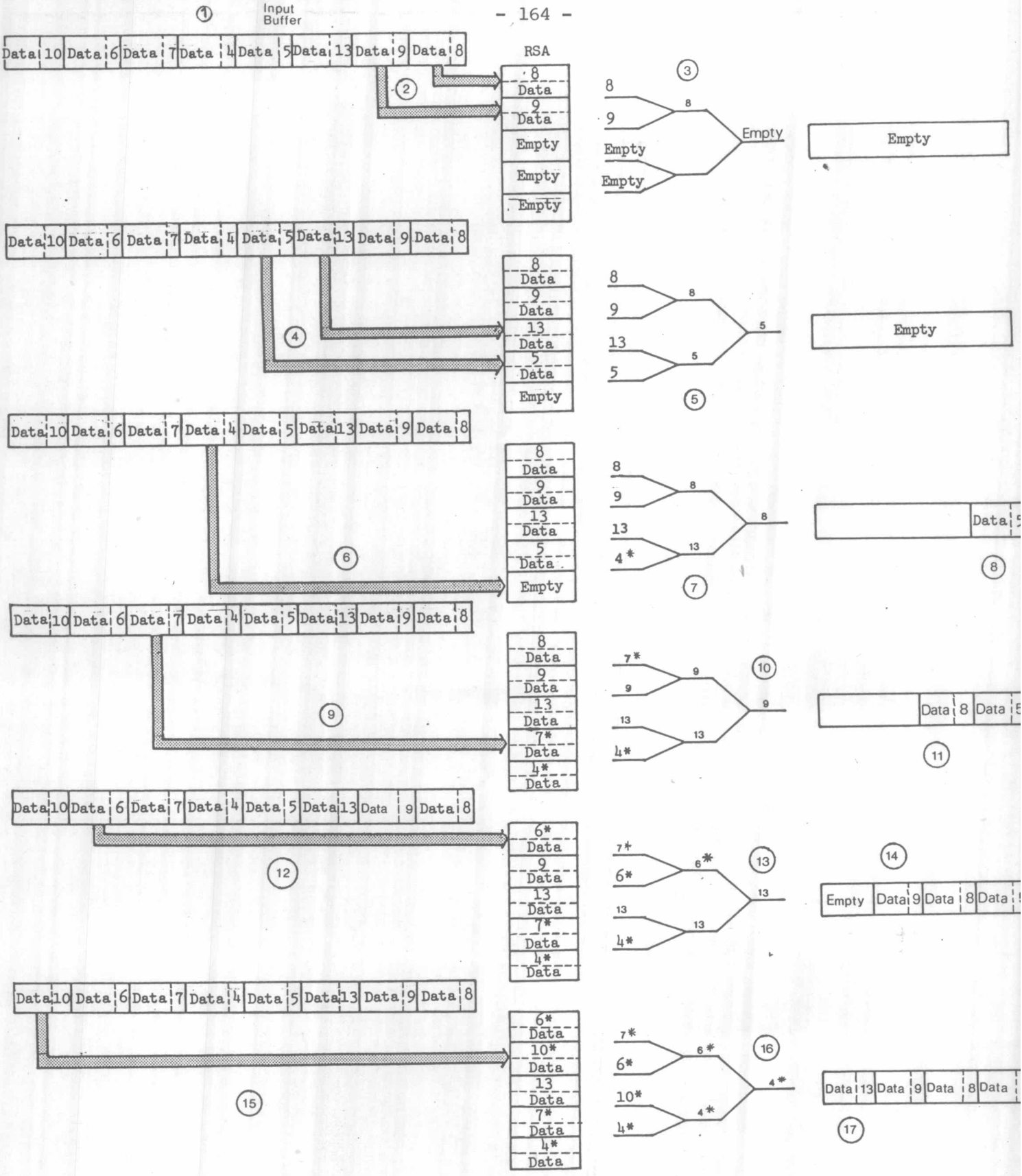
17. เปรียบเทียบ (10*) กับ (4*) จะได้ (4*) เป็นวินเนอร์ในระดับที่ 1 เปรียบเทียบ (6*) กับ (4*) เป็นวินเนอร์ในระดับที่ 2

18. เคลื่อนวินเนอร์ตัวที่แล้ว (13) เข้าไปเก็บในบัฟเฟอร์ผลลัพธ์ เนื่องจาก (4*) ถูก Flag ดังนั้นจะต้องมีการเริ่มสตริงใหม่

เมื่อบัฟเฟอร์ของผลลัพธ์เต็ม ข้อมูลที่อยู่ในบัฟเฟอร์ก็จะถูกบันทึกลงไปเก็บไว้ในที่เก็บข้อมูลชั่วคราว ซึ่งเป็นเทปแม่เหล็กหรือจานแม่เหล็ก โดยที่วิธีการนำไปเก็บนี้ จะเลือกใช้วิธีการกระจายหนึ่งใน 3 วิธีที่จะกล่าวต่อไป

วิธีการกระจายสตริง

โปรแกรมสำเร็จรูปจะมีวิธีการในการกระจายสตริง 3 แบบ คือ แบบโพลีเฟส จะใช้วิธีการนี้เมื่ออุปกรณ์ที่ใช้เป็นที่เก็บข้อมูลชั่วคราวเป็นเทปแม่เหล็ก ส่วนอีก 2 วิธี คือแบบ Interleave และ Split Cylinder นั้น จะใช้จานแม่เหล็กเป็นที่เก็บข้อมูลชั่วคราว




รูปที่ จ.2 ตัวอย่างการเรียงลำดับแบบเลือกแทนที่

วิธีการกระจายแบบโพลีเฟส

ในการกระจายแบบนี้ เทป 1 ม้วนจะถูกเก็บเอาไว้เพื่อใช้ในการรวมข้อมูล สตริงแรกจะบันทึกลงในเทปม้วนแรกในลำดับที่ต้องการ (คือจากน้อยไปมากหรือจากมากไปน้อย) ส่วนสตริงต่อไปที่จะบันทึกลงบนเทปม้วนต่อไป จะเขียนในลำดับตรงข้าม และสตริงที่ 2 ของเทปแต่ละม้วนจะบันทึกในลำดับตรงข้ามกับสตริงแรก เมื่อบันทึกสตริงต่าง ๆ ลงบนเทปซึ่งเป็นที่เก็บข้อมูลชั่วคราวเรียบร้อยแล้ว สตริงจะอยู่สลับลำดับกัน

จำนวนสตริงที่จะบันทึกลงบนเทปแต่ละม้วน จะบันทึกตามหลักการกระจายจำนวนแบบ Fibonacci ในรูป จ.3 จะแสดงให้เห็นจำนวนสตริง ที่กระจายบนที่เก็บข้อมูลชั่วคราวจำนวนต่าง ๆ กัน

| ระดับ | 2 ม้วน | | 3 ม้วน | | | 4 ม้วน | | | | 5 ม้วน | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 2. | 1 | 2 | 1 | 1 | 2 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 |
| 3. | 3 | 2 | 3 | 2 | 4 | 3 | 2 | 4 | 4 | 3 | 2 | 4 | 4 | 4 |
| 4. | 5 | 8 | 7 | 6 | 4 | 7 | 6 | 4 | 4 | 7 | 6 | 4 | 8 | 8 |
| 5. | 13 | 8 | 13 | 24 | 20 | 15 | 14 | 12 | 8 | 15 | 14 | 12 | 8 | 16 |
| 6. | 21 | 34 | 37 | 24 | 48 | 29 | 56 | 52 | 44 | 31 | 30 | 28 | 24 | 16 |
| 7. | 55 | 34 | 81 | 68 | 44 | 85 | 56 | 108 | 100 | 61 | 120 | 116 | 108 | 92 |
| 8. | 89 | 144 | 149 | 274 | 230 | 193 | 164 | 108 | 208 | 181 | 120 | 230 | 228 | 212 |

รูปที่ จ.3 แสดงจำนวนสตริงบนที่เก็บข้อมูลชั่วคราวต่างจำนวนกัน

เพื่อให้เห็นวิธีการดำเนินการเพื่อให้ได้สตริงตามจำนวนที่ต้องการ จะขอยกตัวอย่างการกระจายสตริงในกรณีที่มีเทปเป็นที่เก็บข้อมูลชั่วคราว 4 ม้วน ดังนี้คือ (ดูรูป ค.4 ประกอบ)

1. ในตอนเริ่มต้นสตริง 1 สตริง จะถูกบันทึกลงบนเทปแต่ละม้วนทั้ง 3 ม้วน ส่วนเทปม้วนที่ 4 นั้น จะเก็บไว้ใช้ตอนรวมข้อมูล

2. ในระหว่างรอบที่ 2 และรอบต่อ ๆ ไปนั้น จะมีเทป 1 ใน 3 ม้วนที่จะไม่มีการบันทึกสตริง จะเรียกเทปม้วนนี้ว่าอยู่ในภาวะ static สำหรับรอบนั้นเทปที่จะอยู่ในภาวะนี้จะสลับกันทีละม้วน เริ่มจากเทปม้วนแรกที่ได้รับสตริง

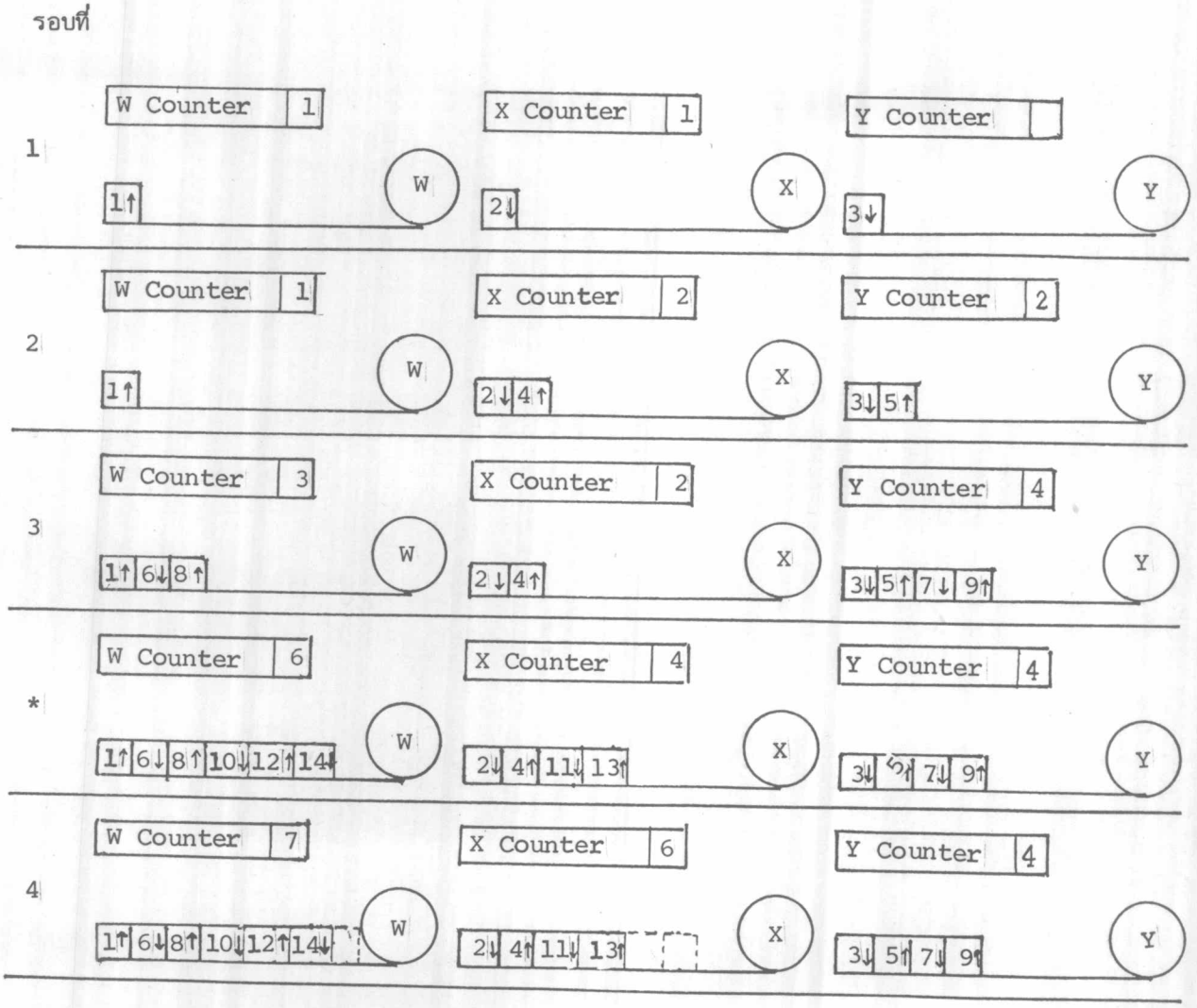
3. จำนวนสตริงที่จะบันทึกบนเทปม้วนอื่นที่ active ในระหว่างรอบที่มีจำนวนสตริงบนเทปถูกต้อง จะมีค่าเท่ากับจำนวนสตริงในเทปม้วนที่ static

4. หลังจากรอบที่ 1 แล้ว รอบใดจะเรียกว่ามีจำนวนสตริงไม่ถูกต้อง ก็ต่อเมื่อในรอบนั้น จำนวนสตริงในเทปแต่ละม้วนเป็นเลขคี่หมด การทำให้จำนวนที่อยู่ในรอบ (*) เป็นไปตามจำนวนที่ควรจะเป็นแบบ Fibonacci จะใช้วิธีการคือ เพิ่มจำนวนสตริงบนเทป 2 ม้วน ซึ่งมีจำนวนสตริงน้อยที่สุด โดยจำนวนสตริงที่เพิ่มมีค่าเท่ากับจำนวนสตริงทั้งหมดบนเทปม้วนที่ 3 (ม้วนที่มีค่ามากที่สุด)

5. จำนวนสตริงในรอบอาจไม่ถูกต้องได้ในกรณีที่เทปม้วนที่มีการเขียนสตริงเป็นม้วนแรกมีจำนวนสตริงเป็นเลขคู่ คือรอบที่ (**) ในกรณีนี้ค่าที่กำหนดในเทปจะถูกปรับเพื่อให้เทปนั้นมีสตริงเป็นเลขคี่ (ทาง logical เท่านั้น แต่ไม่มีการเคลื่อนสตริงระหว่างเทป)

สตริงจะถูกเขียนลงบนเทปที่เป็นที่เก็บข้อมูลชั่วคราวจนกระทั่งหมดสตริง เมื่อไม่มีสตริงจะกระจายต่อไป แต่จำนวนสตริงในแต่ละรอบยังไม่ครบ จะมีการเติมสตริงที่เป็น Dummy ลงไป Dummy สตริงจะเป็นแต่ตัวเลขที่บวกเข้ากับจำนวนสตริงในเทปแต่ละม้วน แต่เมื่อถึงการรวมแฟ้มข้อมูลแล้ว Counters ที่นับจำนวนสตริงจะปรับค่าเพื่อให้รวมสตริงเฉพาะสตริงจริงเท่านั้น

ในรูป จ. 4 จะแสดงให้เห็นการเรียงลำดับที่มีจำนวนสตริงทั้งหมด 14 สตริง กระจายในเทป 3 ม้วน ลูกศรจะแสดงลำดับของสตริง จำนวนสตริงที่มีอยู่ในเทปแต่ละม้วน จะแสดงทางด้านบนคือที่เก็บค่า COUNTER เมื่อจบรอบที่ 4 จำนวนสตริงที่ต้องปรากฏบนเทปแต่ละม้วนจะต้องเป็น 7 6 4 แต่มีจำนวนสตริงจริงเพียง 14 สตริง ดังนั้น จึงต้องเติม Dummy 3 สตริง




รูป จ. 4 แสดงการกระจายสตริง

เมื่อถึงตอนรวมแฟ้มข้อมูล สตริงที่เป็น Dummy จะมีการรวมเช่นเดียวกับสตริงจริง คือ ถ้าเป็นการรวมสตริงที่เป็น Dummy ทั้ง 3 สตริง ก็จะได้สตริงที่เป็น Dummy 1 สตริง แต่ถ้ารวมกับสตริงที่เป็นสตริงจริงแล้ว สตริงที่เป็น Dummy ก็จะหมดไป

ในเฟสที่ 2 จะเป็นการรวมสตริงและได้สตริงที่ยาวกว่าเดิม ต่อจากนั้นนำสตริงที่รวมได้ไปใส่ในเทปม้วนที่ว่าง จนกว่าสตริงจะหมดจากเทปม้วนใดม้วนหนึ่ง ซึ่งหลังจากนั้นเทปม้วนนี้ก็เป็นเทปว่าง และสามารถรับข้อมูลได้อีก ซึ่งรวมทั้งข้อมูลจากเทปที่ใช้เป็นเทปผลลัพธ์เมื่อครั้งที่แล้ว การทำงานจะทำต่อไปเช่นนี้จนกระทั่งมีเทปม้วนหนึ่งเป็นเทปว่าง และเทปม้วนอื่นมีสตริงอยู่เพียงสตริงเดียว

ต่อจากนั้นจะเป็นการทำงานของเฟสที่ 3 ซึ่งจะรวมสตริงในเทปแต่ละม้วน และนำไปใส่ในที่เก็บผลลัพธ์ที่ผู้ใช้งานต้องการ

วิธีกระจายแบบ INTERLEAVE

เป็นวิธีการกระจายสตริงอีกแบบหนึ่ง ซึ่งจะใช้สำหรับบริเวณที่เก็บข้อมูลชั่วคราว ซึ่งอยู่ในอุปกรณ์ที่เข้าถึงข้อมูลได้โดยตรง ในเฟส 1 ซึ่งเป็นการเรียงลำดับภายในหน่วยความจำจะเขียนสตริงลงบนเนื้อที่ครึ่งหนึ่งของบริเวณ ที่ใช้เป็นที่เก็บข้อมูลชั่วคราว ส่วนอีกครึ่งหนึ่งจะเก็บเอาไว้ใช้ในระหว่างการทำงานของเฟส 2 สตริงจะกระจายลงบนบริเวณที่เก็บข้อมูลชั่วคราวในลักษณะที่จะทำให้มีการเคลื่อนหัวอ่านของจานแม่เหล็กน้อยที่สุด เมื่อถึงการทำงานของเฟส 2

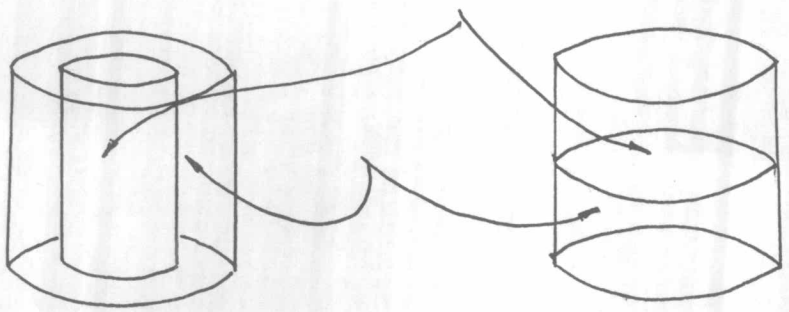
ในการเรียงลำดับแบบเลือกแทนที่จะสร้างสตริงที่มีความยาวคงที่ สตริงเหล่านี้โดยปกติจะไม่ได้บันทึกในแทรคที่ต่อเนื่องกันไป แต่จะบันทึกสลับกันกับแทรคที่เตรียมเอาไว้ให้สำหรับสตริงที่จะตามมา ค่าของ INTERLEAVE FACTOR มีค่าเท่ากับค่าของ MERGE ORDER นั่นคือสตริงจะถูกบันทึกทุกแทรคที่ M ของอุปกรณ์ที่สามารถเข้าถึงข้อมูลได้โดยตรง สตริงจะถูกบันทึกต่อเนื่องกันเมื่อบริเวณที่เก็บข้อมูลชั่วคราวมีเหลือน้อยเกินไป ไม่พอที่จะทำวิธีดังกล่าวได้

การจัดรูปแบบ (Format) ของแทรคจะจัดเมื่อจะมีการบันทึกข้อมูลครั้งแรก

ในระหว่างการทำงานในรอบแรกของเฟสที่ 2 สตริงจะถูกนำมารวมกันทำให้ได้สตริงที่ยาวกว่าเดิม และจะถูกบันทึกลงในอีกครึ่งหนึ่งของบริเวณที่เก็บข้อมูลชั่วคราวโดยวิธีการเดิม ถ้ามีเนื้อที่ว่างพอ การทำงานจะทำด้วยวิธีดังกล่าวโดยการสลับบริเวณของแต่ละครึ่งของเนื้อที่จนกระทั่งจำนวนสตริงมีค่าน้อยกว่า หรือเท่ากับค่า MERGE ORDER จึงส่งการทำงานไปให้เฟส 3 เพื่อรวมสตริงให้เป็นสตริงเดียว และบันทึกลงในอุปกรณ์ที่ผู้ใช้งานเลือก

วิธีกระจายแบบ SPLIT CYLINDER

วิธีนี้จะใช้เมื่อมีบริเวณที่ใช้เป็นที่เก็บข้อมูลชั่วคราวเพียงบริเวณเดียว การจัดรูปแบบของแทรคจะจัดเมื่อมีการบันทึกข้อมูลครั้งแรก การกระจายจะเป็นแบบเดียวกับวิธีที่แล้ว แต่เนื่องจากการแบ่งบริเวณที่ใช้เป็นที่เก็บข้อมูลชั่วคราวจะแบ่งในแนวนอน ดังนั้น SEEK TIME ในการอ่านจากครึ่งหนึ่งของเนื้อที่ แล้วไปบันทึกในอีกครึ่งของเนื้อที่ จึงลดลง



รูป จ.5 เปรียบเทียบการกระจายแบบ INTERLEAVE และ SPLIT CYLINDER

ภาคผนวก ฉ.

การอ่านผลลัพธ์จาก PDAID

ผลลัพธ์ที่ได้จาก PDAID จะแบ่งออกเป็น 2 กลุ่มเหมือนกัน คือทางด้านซ้ายและขวา ในการอ่านผลลัพธ์จาก PDAID จะอ่านจากด้านซ้ายไปด้านขวาตามลำดับ โดยเริ่มจากโมดุล SORT เป็นต้นไป ในแต่ละกลุ่มประกอบด้วยเขตข้อมูลดังนี้คือ

1. ชื่อโมดุล 8 ตัวอักษร
2. ค่าของ SVC (Supervisor Call)
3. ตำแหน่งที่เรียกโมดุลนั้นเข้ามาทำงาน
4. พาร์ทิชั่นที่โปรแกรมทำงาน ในที่นี้คือ F3
5. ตำแหน่งที่โมดุลนั้นถูกนำเข้ามาเก็บในหน่วยความจำ
6. ตำแหน่งที่โมดุลนั้นเริ่มทำงาน

ภาคผนวก ช.

หน้า

โปรแกรมย่อย SORTC



```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                    DOS/VS ASSEMBLER REL 34.0 02.37 82-04-11
                                    1 ************************************************************************
                                    2 **  MAIN PROGRAM TO SORT DATA IN CARD                                  **
                                    3 ************************************************************************


000000                              5 SORTC    CSECT
                                    6          PRINT ON,NOGEN
000000 05A0                         7          BALR  10,0
                          00002     8          USING *,10
000002 90ED A01E          00020     9          STM   14,13,S1
000006 58F0 A056          00068    10          L     15,=A(READC)
00000A 05EF                        11          BALR  14,15
00000C 58F0 A06A          0006C    12          L     15,=A(SORT)
000010 05EF                        13          BALR  14,15
000012 58F0 A06E          00070    14          L     15,=A(READD)
000016 05EF                        15          BALR  14,15
000018 98ED A01E          00020    16          LM    14,13,S1
00001C 07FE                        17          BR    14
                                   18          END
000020                             21 S1       DS    9D
000068                             22          LTORG
000068 00000000                    23                =A(READC)
00006C 00000000                    24                =A(SORT)
000070 00000000                    25                =A(READD)
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                    DOS/VS ASSEMBLER REL 34.0 02.37 82-04-11
                                    27 ********************************************************************************
                                    28 **  THIS CONTROL SECTION IS CREATED TO READ DATA FROM CARDS AND      **
                                    29 **  WRITE THEM INTO DISK , AFTER THAT  READ THEM BACK FROM DISK AND  **
                                    30 **  PRINT ON HARD COPY FOR CHECKING .                               **
                                    31 ********************************************************************************


000000                              33 READC    CSECT
                                    34 * 8316921 PAVADEE VARAPUNYAVATANA
000000 05C0                         35          BALR  12,0
                          00002     36          USING *,12
000002 90ED C98E          00990     37          STM   14,13,SV1
                                    38          PRINT ON,NOGEN
                                    39          OPEN  CARD,SORTIN1  OPEN FILES.
                                    48          CNTRL PRINT,SK,1    SKIP 1 PAGE
                          00028     54 XX       EQU   *
                                    55 MOVEDATA GET   CARD,DATA     GET DATA FROM CARD.
                                    61          PUT   SORTIN1,DATA  WRITE DATA TO DISK.
000048 47F0 C026        00028       67          B     XX            BRANCH TO GET DATA  IF DATA IS NOT AT END
                                    68 EOFRTN   CLOSE CARD,SORTIN1  CLOSE FILES.
                                    77          OPEN  PRINT,DISKWR  OPEN FILES.
                                    85          CNTRL PRINT,SK,1    SKIP 1 PAGE.
000080 D283 C28D C28C 028F 028E     92          MVC   PR,PR-1
000086 D21E C2C2 C826 02C4 0828     93          MVC   PR+5(31),=C'*** I/P DATA BEFORE SORTING ***'
                                    94          PUT   PRINT,PR
                                   100          CNTRL PRINT,SP,3
                                   106 GETDISK  GET   DISKWR,DATA   READ DATA FROM DISK.
0000BA D283 C28D C28C 028F 028E    112 WRDK     MVC   PR,PR-1       CLEAR PRINT AREA.
0000C0 D24F C2C7 C26C 02C9 026E    113          MVC   PR+10(30),DATA  MOVE DATA TO PRINT AREA.
                                   114          CNTRL PRINT,SP,1    SPACE 1 LINE.
                                   120          PUT   PRINT,PR      PRINT DATA FROM PRINT AREA TO HARD COPY.
0000E4 47F0 C0A8        000AA      126          B     GETDISK       BRANCH TO GET DATA FROM DISK AGAIN.
                                   127 EOFDISK  CLOSE PRINT,DISKWR  CLOSE FILES.
0000EA 98ED C98E          00990    136          LM    14,13,SV1
0000EE 07FE                        137          BR    14            BRANCH BACK TO CALLING PROGRAM.
                                   138          END
                                   141          FILE  CARD
                                   295          FILE  PRINT
00026F                             446 DATA     DS    CL80          BUFFER OF CARD
0002BE 40                          447          DS    C' '
0002BF                             448 PR       DS    CL132         BUFFER OF PRINT
000343                             449 DBUFF    DS    808C          BUFFER OF SORTIN1
00066B                             450 DB1      DS    800C          BUFFER OF DISKWR
000990                             451 SV1      DS    9D            SAVEAREA FOR RETURNING TO FORTRAN
0009D8                             452          DS    0D
                                   453 SORTIN1  DTFSD BLKSIZE=808,                                          *
                                                      IOAREA1=DBUFF,                                        *
                                                      DEVADDR=SYS002,                                       *
                                                      RECFORM=FIXBLK,                                       *
                                                      DEVICE=3340,                                          *
                                                      RECSIZE=80,
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                    DOS/VS ASSEMBLER REL 34.0 02.37 82-04-11
                                                TYPEFLE=OUTPUT,                                          *
                                                WORKA=YES
000A78                                   504            DS    0D
                                         505 DISKWR     DTFSD BLKSIZE=800,IOAREA1=DB1,                   *
                                                      EOFADDR=EOFDISK,                                   *
                                                      RECFORM=FIXBLK,                                    *
                                                      DEVADDR=SYS005,                                    *
                                                      DEVICE=3340,RECSIZE=80,                            *
                                                      TYPEFLE=INPUT,WORKA=YES
000B00                                   551            LTORG
000B00 5B5BC2D6D7C5D540                  552                   =C'$$BOPEN '
000B08 5B5BC2C3D3D6E2C5                  553                   =C'$$BCLOSE'
000B10 00000190                          554                   =A(PRINT)
000B14 00000108                          555                   =A(CARD)
000B18 0000026E                          556                   =A(DATA)
000B1C 000009D8                          557                   =A(SORTIN1)
000B20 000002BE                          558                   =A(PR)
000B24 00000A78                          559                   =A(DISKWR)
000B28 5C5C5C40C961D740                  560                   =C'*** I/P DATA BEFORE SORTING ***'
```

```
                                        562 ********************************************************************************
                                        563 **  THIS CONTROL SECTION IS CREATED TO LOAD SORT PROGRAM UTILITY     **
                                        564 **  FROM CORE IMAGE LIBRARY TO 'LOADLOC' LOCATION IN CONTROL SECTION **
                                        565 **  'LOADLOC' IN THE ADJACENT PHASE. AFTER THAT, THE ROUTINE WILL    **
                                        566 **  BRANCH TO EXECUTE IT AND GET THE SORTING RESULT BACK TO THIS     **
                                        567 **  CONTROL SECTION. FOR SORT PROGRAM VALIDITY CHECKING, RETURN      **
                                        568 **  CODE WILL BE CHECKED AND RESULT OF SORTING WILL BE PRINT OUT ON  **
                                        569 **  HARD COPY FOR CHECKING.                                          **
                                        570 ********************************************************************************

000000                                  572 SORT     CSECT
                                        573          ENTRY PARAM,JCL,RCD,INPFL,SM,OUTFL,OPT,MODS,ALTSQ
                                        574          ENTRY OUTRC,INCOM,DATA
000000 05C0                             575          BALR  12,0
                                00002   576          USING *,12,11
000002 41BC OFFF                00FFF   577          LA    11,4095(12)
000006 90EC C6D6                006D8   578          STM   14,12,SV2      SAVE CONTENT FOR USE
00000A 50D0 C716                00718   579          ST    13,SV2+64      SAVE CONTENT OF REG13
                                        580          PDUMP SV2,DATA1
000018 4580 C0D6                000D8   585          BAL   8,GJCS
                                        586          PDUMP PARAM,SAVAREA
000026 5840 C35E                00360   591          L     4,=V(LOADLOC)  LOAD ADDR. FOR SORT ROUTINE
                                        592          LOAD  SORT,(4)       LOAD SORT FROM CIL.
000032 18F1                             598          LR    15,1           LOAD ADDR. OF ENTRY POINT
000034 4110 C7F6                007F8   599          LA    1,PARAM        LOAD ADDR. OF PARAMETER LIST ARRAY
000038 41D0 CED6                00ED8   600          LA    13,SAVAREA     LOAD ADDR. OF SAVEAREA
00003C 05EF                             601          BALR  14,15          BRANCH TO SORT
00003E D501 CF1E C37E 00F20 00380       602          CLC   RETURN(2),=H'0' COMPARE RETURNING CODE
000044 4770 C082        00084           603          BNE   SORTERR        IF ERROR GO TO SORTERR
                                        604          OPEN  SPRINT
000056 D283 C76F C76E 00771 00770       612          MVC   SPR,SPR-1      IF NOT,CLEAR PRINT
00005C D21A C779 C3BA 00778 003BC       613          MVC   SPR+10(27),=C'*** SORTING IS COMPLETE ***'
                                        614          CNTRL SPRINT,SK,1
                                        620          PUT   SPRINT,SPR
000080 47F0 CORE        000C0           626          B     ENDOJ
                                        627 SORTERR  OPEN  SPRINT
000092 D283 C76F C76E 00771 00770       635          MVC   SPR,SPR-1
000098 D21E C779 C3D5 00778 003D7       636          MVC   SPR+10(31),=C'*** SORTING IS NOT COMPLETE ***'
                                        637          CNTRL SPRINT,SK,1
                                        643          PUT   SPRINT,SPR
0000BC 47F0 CORE        000C0           649          B     ERRPT
                                00OC0   650 ENDOJ    EOJ   *
                                000C0   651 ERRPT    EQU   *
                                        652          CLOSE SPRINT
0000CE 98EC C6D6                006D8   660          LM    14,12,SV2      RETURN CONTENT OF ALL REGS.
0000D2 58D0 C716                00718   661          L     13,SV2+64
0000D6 07FE                             662          BR    14             RETURN TO FORTRAN
                                000D8   663 GJCS     EQU   *
0000D8 D2EF C467 C466 00469 00468       664          MVC   BUFINP,BUFINP-1
0000DE D2EF C557 C467 00559 00469       665          MVC   BUFINPX,BUFINP
```

```
0000E4 4140 C7F6          007F8   665          LA    4,PARAM
                                  667          OPEN  CD
                                  675 READ     GET   CD,DATAIN
000106 D504 C417 C3F4 00419 003F6 681          CLC   DATAIN+1(5),=C'**JCS'
00010C 4780 C0F4          000F6   682          BE    READ
000110 D503 C417 C372 00419 00374 683          CLC   DATAIN+1(4),=C'SORT'
000116 4780 C18A          0018C   684          BE    RSORT
00011A D505 C417 C380 00419 00382 685          CLC   DATAIN+1(6),=C'RECORD'
000120 4780 C19A          0019C   686          BE    RRECORD
000124 D505 C417 C386 00419 00388 687          CLC   DATAIN+1(6),=C'INPFIL'
00012A 4780 C1AA          001AC   688          BE    RINPFL
00012E D505 C417 C38C 00419 0038E 689          CLC   DATAIN+1(6),=C'OUTFIL'
000134 4780 C1BA          001BC   690          BE    ROUTFIL
000138 D505 C417 C392 00419 00394 691          CLC   DATAIN+1(6),=C'OPTION'
00013E 4780 C1CA          001CC   692          BE    ROPTION
000142 D505 C417 C398 00419 0039A 693          CLC   DATAIN+1(6),=C'ALTSEQ'
000148 4780 C1F0          001F2   694          BE    RALTSEQ
00014C D505 C417 C39E 00419 003A0 695          CLC   DATAIN+1(6),=C'OUTREC'
000152 4780 C200          00202   696          BE    ROUTREC
000156 D502 C417 C3F9 00419 003FB 697          CLC   DATAIN+1(3),=C'SUM'
00015C 4780 C210          00212   698          BE    RSUM
000160 D506 C417 C3FC 00419 003FE 699          CLC   DATAIN+1(7),=C'INCLUDE'
000166 4780 C220          00222   700          BE    RINCLUDE
00016A D503 C417 C376 00419 00378 701          CLC   DATAIN+1(4),=C'OMIT'
000170 4780 C220          00222   702          BE    ROMIT
000174 D503 C417 C37A 00419 0037C 703          CLC   DATAIN+1(4),=C'MODS'
00017A 4780 C1E0          001E2   704          BE    RMODS
00017E D504 C417 C403 00419 00405 705          CLC   DATAIN+1(5),=C'**END'
000184 4780 C2DA          002DC   706          BE    RET
000188 47F0 C232          00234   707          B     ERROR
00018C 4150 C82E                  00830 708 RSORT    LA    5,JCL
000190 4590 C260                  00262 709          BAL   9,CONT
000194 5050 4000                  00000 710          ST    5,0(0,4)
000198 47F0 C2CE          002D0   711          B     MOVE
00019C 4150 C8CE                  008D0 712 RRECORD  LA    5,RCD
0001A0 4590 C260                  00262 713          BAL   9,CONT
0001A4 5050 4004                  00004 714          ST    5,4(0,4)
0001A8 47F0 C2CE          002D0   715          B     MOVE
0001AC 4150 C96E                  00970 716 RINPFL   LA    5,INPFL
0001B0 4590 C260                  00262 717          BAL   9,CONT
0001B4 5050 4008                  00008 718          ST    5,8(0,4)
0001B8 47F0 C2CE          002D0   719          B     MOVE
0001BC 4150 CA0E                  00A10 720 ROUTFIL  LA    5,OUTFL
0001C0 4590 C260                  00262 721          BAL   9,CONT
0001C4 5050 400C                  0000C 722          ST    5,12(0,4)
0001C8 47F0 C2CE          002D0   723          B     MOVE
0001CC 4150 CA8E                  00A80 724 ROPTION  LA    5,OPT
0001D0 4590 C260                  00262 725          BAL   9,CONT
0001D4 5050 4010                  00010 726          ST    5,16(0,4)
0001D8 D2EF 5000 C557 00000 00559 727          MVC   0(240,5),BUFINPX
0001DE 47F0 C0F4          000F6   728          B     READ
```

```
LOC   OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
0001F2 4150 CB9E           00BA0   729 RMODS    LA    5,MODS
0001E6 4590 C260           00262   730          BAL   9,CONT
0001EA 5050 4014           00014   731          ST    5,20(0,4)
0001EE 47F0 C2CE     00200         732          B     MOVE
0001F2 4150 CC3E           00C40   733 RALTSEQ  LA    5,ALTSQ
0001F6 4590 C260           00252   734          BAL   9,CONT
0001FA 5050 4028           00028   735          ST    5,40(0,4)
0001FE 47F0 C2CE     00200         736          B     MOVE
000202 4150 CCDE           00CE0   737 ROUTREC  LA    5,OUTRC
000206 4590 C260           00262   738          BAL   9,CONT
00020A 5050 402C           0002C   739          ST    5,44(0,4)
00020E 47F0 C2CE     00200         740          B     MOVE
000212 4150 CD7E           00080   741 RSUM     LA     5,SM
000216 4590 C260           00262   742          BAL   9,CONT
00021A 5050 4030           00030   743          ST    5,48(0,4)
00021E 47F0 C2CE     00200         744          B     MOVE
                           00222   745 RINCLUDE EQU   *
000222 4150 CE1E           00E20   746 ROMIT    LA    5,INCOM
000226 4590 C260           00262   747          BAL   9,CONT
00022A 5050 4034           00034   748          ST    5,52(0,4)
00022E 47F0 C2CE     00200         749          B     MOVE
                                   750 ERROR    OPEN  SPRINT
000242 D203 C76F C76E 00771 00770  758          MVC   SPR,SPR-1
000248 D20A C779 C408 0077B 0040A  759          MVC   SPR+10(11),=C'INVALID JCS'
                                   760          PUT   SPRINT,SPR
00025E 47F0 C0BE     000C0         766          B     ENDOJ
000262 955C C45D     0045F         767 CONT     CLI   DATAIN+71,C'*'
000266 4780 C27C     0027E         768          BE    CONTI
00026A D245 C467 C417 00469 00419  769          MVC   BUFINP(70),DATAIN+1
000270 D2EF C557 C467 00559 00469  770          MVC   BUFINPX,BUFINP
000276 D2EF C467 C466 00469 00468  771          MVC   BUFINP,BUFINP-1
00027C 07F9                        772          BR    9
00027E 4160 C467           00469   773 CONTI    LA    6,BUFINP
000282 4170 C417           00419   774          LA    7,DATAIN+1
000286 D245 6000 7000 00000 00000  775 CONTN    MVC   0(70,6),0(7)
00028C 4166 0046           00046   776          LA    6,70(6)
                           00290   777 NN       EQU   *
                                   778          GET   CD,DATAIN
0002A0 4170 C416           00418   784          LA    7,DATAIN
0002A4 D54F C45D 005C 0045F 0005C  785          CLC   DATAIN+71,C'*'
0002AA 4780 C2C0     002C2         786          BE    CONTN1
0002AE D246 6000 C416 00000 00418  787          MVC   0(71,6),DATAIN
0002B4 D2EF C557 C467 00559 00469  788          MVC   BUFINPX,BUFINP
0002BA D2EF C467 C466 00469 00468  789          MVC   BUFINP,BUFINP-1
0002C0 07F9                        790          BR    9
0002C2 D246 6000 7000 00000 00000  791 CONTN1   MVC   0(71,6),0(7)
0002C8 4166 0047           00047   792          LA    6,71(6)
0002CC 47F0 C28E     00290         793          B     NN
0002D0 D29F 5000 C557 00000 00559  794 MOVE     MVC   0(160,5),BUFINPX
0002D6 47F0 C0F4     000F6         795          B     READ
                                   796 RET    CLOSE CD
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                    DOS/VS ASSEMBLER REL 34.0 02.37 82-04-11

0002EA 07F8                              804          BR    8
                                         805 EOF      CLOSE CD
                                         813          OPEN  SPRINT
00030A D283 C76F C76E 00771 00770        821          MVC   SPR,SPR-1
000310 D215 C779 C3A4 0077B 003A6        822          MVC   SPR+10(22),=C'END OF JCS IS REQUIRED'
                                         823          PUT   SPRINT,SPR
000326 47F0 C0BE      000C0              829      3     ENDJ
000330                                   830          LTORG
000330 5858C2D7C4E4D4D7                  831                =CL8'$$BPDUMP'
000338 000006D800000720                  832                =A(SV2,DATA1)
000340 000007F800000ED8                  833                =A(PARAM,SAVAREA)
000348 E2D6D9E340404040                  834                =C_8'SORT'
000350 5858C2D6D7C5D540                  835                =C'$$BOPEN '
000358 5858C2C3D3D6E2C5                  836                =C'$$BCLOSE'
000360 00000000                          837                =V(_DADLOC)
000364 00000F28                          838                =A(SPRINT)
000368 00000771                          839                =A(SPR)
00036C 00000650                          840                =A(CD)
000370 00000418                          841                =A(DATAIN)
000374 E2D6D9E3                          842                =C'SORT'
000378 D6D4C9E3                          843                =C'OMIT'
00037C D4D6C4E2                          844                =C'MODS'
000380 0000                              845                =H'0'
000382 D9C5C3D6D9C4                      846                =C'RECORD'
000388 C9D5D7C6C9D3                      847                =C'INPFIL'
00038E D6E4E3C6C9D3                      848                =C'OUTFIL'
000394 D6D7E3C9D6D5                      849                =C'OPTION'
00039A C1D3E3E2C5D8                      850                =C'ALTSEQ'
0003A0 D6E4E3D9C5C3                      851                =C'OUTREC'
0003A6 C5D5C440D6C640D1                  852                =C'END OF JCS IS REQUIRED'
0003BC 5C5C5C40E2D6D9E3                  853                =C'*** SORTING IS COMPLETE ***'
0003D7 5C5C5C40E2D6D9E3                  854                =C'*** SORTING IS NOT COMPLETE ***'
0003F6 5C5CD1C3E2                        855                =C'**JCS'
0003FB E2E4D4                            856                =C'SUM'
0003FE C9D5C3D3E4C4C5                    857                =C'INCLUDE'
000405 5C5CC5D5C4                        858                =C'**END'
00040A C9D5E5C1D3C9C4C0                  859                =C'INVALID JCS'
                                         860          EDJ
000418                                   863 DATAIN   DS    CL80
000468 40                                864          DC    C' '
000469                                   865 BUFINP   DS    CL240
000559                                   866 BUFINPX  DS    CL240
                                         867 CD       DTFCD DEVADDR=SYSIPT,IOAREA1=INCARD,BLKSIZE=80,DEVICE=3505,    *
                                                            EOFADDR=EOF,ERROPT=SKIP,MODNAME=LOGICCD,WORKA=YES
000688                                   890 INCARD   DS    CL80
                                         891 LOGICCD  CDMOD DEVICE=3505,WORKA=YES
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                    DOS/VS ASSEMBLER REL 34.0 02.37 82-04-11

0006D8                            996 SV2      DS    9D
000720                            997 DATA1    DS    CL80
000770 40                         998          DC    C' '
000771                            999 SPR      DS    CL132
0007F8                           1000          DS    0D
0007F8 00000000                  1001 PARAM    DC    A(0)          ADDR. OF SORT CARD IMAGE
0007FC 00000000                  1002          DC    A(0)          ADDR. OF RECORD CARD IMAGE
000800 00000000                  1003          DC    A(0)          ADDR. OF INPFIL CARD IMAGE
000804 00000000                  1004          DC    A(0)          ADDR. OF OUTFIL CARD IMAGE
000808 00000000                  1005          DC    A(0)          ADDR. OF OPTION CARD IMAGE
00080C 00000000                  1006          DC    A(0)          ADDR. OF MODS CARD IMAGE
000810 00000000                  1007          DC    A(0)          ADDR. OF E11 CARD IMAGE
000814 00000000                  1008          DC    A(0)          ADDR. OF E21 CARD IMAGE
000818 00000000                  1009          DC    A(0)          ADDR. OF E31 CARD IMAGE
00081C 00000F20                  1010          DC    A(RETURN)     ADDR. OF RETURNING CODE
000820 00000000                  1011          DC    A(0)          ADDR. OF ALTSEQ CARD IMAGE
000824 00000000                  1012          DC    A(0)          ADDR. OF OUTREC CARD IMAGE
000828 00000000                  1013          DC    A(0)          ADDR. OF SUM CARD IMAGE
00082C 00000000                  1014          DC    A(0)          ADDR. OF INCLUDE/OMIT CARD IMAGE
000830                           1015 JCL      DS    CL150         SORT JCS
0008D0                           1016 RCD      DS    CL160         RECORD JCS
000970                           1017 INPFL    DS    CL150         INPFIL JCS
000A10                           1018 OUTFL    DS    CL150         OUTFIL JCS
000AB0                           1019 OPT      DS    CL240         OPTION JCS
000BA0                           1020 MODS     DS    CL160
000C40                           1021 ALTSQ    DS    CL150         ALTSEQ JCS
000CE0                           1022 OUTRC    DS    CL160         OUTREC JCS
000D80                           1023 SM       DS    CL160         SUM JCS
000E20                           1024 INCOM    DS    CL180         INCLUDE/OMIT JCS
000ED8                           1025 SAVAREA  DS    9D
000F20 0000                      1026 RETURN   DC    H'0'
000F28                           1027          DS    0D
                                 1028 SPRINT   DTFPR DEVADDR=SYSLST,                                    *
                                                   IOAREA1=SPR,                                        *
                                                   BLKSIZE=132,                                        *
                                                   CONTROL=YES,                                        *
                                                   DEVICE=3203,                                        *
                                                   MODNAME=LOGICPRT,                                   *
                                                   PRINTOV=YES,                                        *
                                                   WORKA=YES
                                 1049 LOGICPRT PRMOD CONTROL=YES,                                      *
                                                   DEVICE=3203,PRINTOV=YES,                            *
                                                   WORKA=YES
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                    DOS/VS ASSEMBLER REL 34.0 02.37 82-04-11

                                   1155 ********************************************************************************
                                   1166 **  SUBPROGRAM TO READ DATA FROM DISK(SORTED DATA) TO HARD COPY      **
                                   1167 ********************************************************************************


000000                             1169 READD    CSECT
                                   1170          PRINT ON,NOGEN
000000 05B0                        1171          BALR  11,0
                           00002   1172          USING *,11
000002 90ED B0B6           00088   1173          STM   14,13,S2
                                   1174          OPEN  DPRINT
                                   1182          OPEN  DISKWR1
000026 D283 B12F B12F 00131 00130  1190          MVC   DPR,DPR-1
00002C D21D B134 B57E 00136 00580  1191          MVC   DPR+5(30),=C'*** O/P DATA AFTER SORTING ***'
                                   1192          CNTRL DPRINT,SK,1
                                   1198          PUT   DPRINT,DPR
                                   1204          CNTRL DPRINT,SP,3
                                   1210 GETA     GET   DISKWR1,DATA1
00006E D283 B12F B12F 00131 00130  1216 WRDKTEST MVC   DPR,DPR-1
000074 D24F B139 C71E 00138 00720  1217          MVC   DPR+10(30),DATA1
                                   1218          CNTRL DPRINT,SP,1
                                   1224          PUT   DPRINT,DPR
000098 47F0 B05C        0005E      1230          B     GETA
                                   1231 EOFD     CLOSE DPRINT,DISKWR1
0000AE 98ED B0B6           00088   1240          LM    14,13,S2
0000B2 07FE                        1241          BR    14
                                   1242          EJJ
000088                             1245 S2       DS    9D
                                   1246 DPRINT   DTFPR DEVADDR=SYSLST,IOAREA1=DPR,BLKSIZE=132,CONTROL=YES,      *
                                                       DEVICE=3203,MODNAME=LOGICPR,PRINTOV=YES,WORKA=YES
                                   1267 LOGICPR  PRMOD CONTROL=YES,DEVICE=3203,PRINTOV=YES,WORKA=YES
000130 40                          1382          DC    C' '
000131                             1383 DPR      DS    CL132
                                   1384 DISKWR1  DTFSD BLKSIZE=800,                                              *
                                                       EOFADDR=EOFD,                                             *
                                                       IOAREA1=SBUFF,                                            *
                                                       DEVADDR=SYS006,DEVICE=3340,                               *
                                                       RECFORM=FIXBLK,                                           *
                                                       RECSIZE=80,TYPEFLE=INPUT,WORKA=YES
000240                             1430 SBUFF    DS    CL800
000560                             1431          LTORG
000560 5B5BC2D6D7C5D540            1432                =C'$$BOPEN '
000568 5B5BC2C3D3D6E2C5            1433                =C'$$BCLOSE'
000570 00000100                    1434                =A(DPRINT)
000574 00000131                    1435                =A(DPR)
000578 00000188                    1436                =A(DISKWR1)
00057C 00000720                    1437                =A(DATA1)
000580 5C5C5C4006610740            1438                =C'*** O/P DATA AFTER SORTING ***'
                           00000   1439          END   SORTC
```



```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                    DOS/VS ASSEMBLER REL 34.0 07.51 82-04-11

000688                             467 SV2      DS    9D
0006F0                             468 DATA1    DS    CL90
000740 40                          469          DC    C' '
000741                             470 SPR      DS    CL132
0007C8                             471          DS    0D
0007C8 00000000                    472 PARAM    DC    A(0)          ADDR. OF SORT CARD IMAGE
0007CC 00000000                    473          DC    A(0)          ADDR. OF RECORD CARD IMAGE
0007D0 00000000                    474          DC    A(0)          ADDR. OF INPFIL CARD IMAGE
0007D4 00000000                    475          DC    A(0)          ADDR. OF OUTFIL CARD IMAGE
0007D8 00000000                    476          DC    A(0)          ADDR. OF OPTION CARD IMAGE
0007DC 00000000                    477          DC    A(0)          ADDR. OF MODS CARD IMAGE
0007E0 00000000                    478          DC    A(0)          ADDR. OF E11 CARD IMAGE
0007E4 00000000                    479          DC    A(0)          ADDR. OF E21 CARD IMAGE
0007E8 00000000                    480          DC    A(0)          ADDR. OF E31 CARD IMAGE
0007EC 00000EF0                    481          DC    A(RETURN)     ADDR. OF RETURNING CODE
0007F0 00000000                    482          DC    A(0)          ADDR. OF ALTSEQ CARD IMAGE
0007F4 00000000                    483          DC    A(0)          ADDR. OF OUTREC CARD IMAGE
0007F8 00000000                    484          DC    A(0)          ADDR. OF SUM CARD IMAGE
0007FC 00000000                    485          DC    A(0)          ADDR. OF INCLUDE/OMIT CARD IMAGE
000800                             486 JCL      DS    CL150         SORT JCS
000898                             487 RCD      DS    CL160         RECORD JCS
000940                             488 INPFL    DS    CL150         INPFIL JCS
0009E0                             489 OUTFL    DS    CL150         OUTFIL JCS
000A80                             490 OPT      DS    CL240         OPTION JCS
000B70                             491 MODS     DS    CL150
000C10                             492 ALTSQ    DS    CL150         ALTSEQ JCS
000CB0                             493 OUTRC    DS    CL160         OUTREC JCS
000D50                             494 SM       DS    CL160         SUM JCS
000DF0                             495 INCOM    DS    CL180         INCLUDE/OMIT JCS
000EA8                             496 SAVAREA  DS    9D
000EF0 0000                        497 RETURN   DC    H'0'
000EF8                             498          DS    0D
                                   499 SPRINT   DTFPR DEVADDR=SYS010,                                    *
                                                      IOAREA1=SPR,                                       *
                                                      BLKSIZE=132,                                       *
                                                      CONTROL=YES,                                       *
                                                      DEVICE=3203,                                       *
                                                      MODNAME=LOGICPRT,                                  *
                                                      PRINTOV=YES,                                       *
                                                      WORKA=YES
                                   520 LOGICPRT PRMOD CONTROL=YES,                                       *
                                                      DEVICE=3203,PRINTOV=YES,                           *
                                                      WORKA=YES
                          00000    635          END   SORTT
```

```
                                1 ********************************************************************************
                                2 **  THIS CONTROL SECTION IS CREATED TO DEFINE AREA FOR SORT PROGRAM  **
                                3 **  UTILITY EXECUTION AREA.                                         **
                                4 ********************************************************************************

                                5 LOADLOC  CSECT
000000                          7          DS    20CL1024
000000                          8          END   LOADLOC
                     00000
```

โปรแกรมย่อย SORTT



```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                    DOS/VS ASSEMBLER REL 34.0 07.51 82-04-11
                                    1 ********************************************************************************
                                    2 **  MAIN PROGRAM TO SORT DATA IN TAPE                                         **
                                    3 ********************************************************************************

000000                              5 SORTT   CSECT
                                    6         PRINT ON,NOGEN
000000 05A0                         7         BALR  10,0
                          00002     8         USING *,10
000002 90ED A016          00018     9         STM   14,13,S1
000006 58F0 A05E          00060     10        L     15,=A(SORT)
00000A 05EF                         11        BALR  14,15
00000C 98ED A016          00018     12        LM    14,13,S1
000010 07FE                         13        BR    14
                                    14        EOJ
000018                              17 S1     DS    9D
000060                              18        LTORG
000060 00000000                     19                =A(SORT)
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                    DOS/VS ASSEMBLER REL 34.0 07.51 82-04-11
                                   21 ********************************************************************************
                                   22 **  THIS CONTROL SECTION IS CREATED TO LOAD SORT PROGRAM UTILITY     **
                                   23 **  FROM CORE IMAGE LIBRARY TO 'LOADLOC' LOCATION IN CONTROL SECTION **
                                   24 **  'LOADLOC' IN THE ADJACENT PHASE. AFTER THAT, THE ROUTINE WILL    **
                                   25 **  BRANCH TO EXECUTE IT AND GET THE SORTING RESULT BACK TO THIS     **
                                   26 **  CONTROL SECTION. FOR SORT PROGRAM VALIDITY CHECKING, RETURN      **
                                   27 **  CODE WILL BE CHECKED AND RESULT OF SORTING WILL BE PRINT OUT ON  **
                                   28 **  HARD COPY FOR CHECKING.                                          **
                                   29 ********************************************************************************

000000                             31 SORT     CSECT
000000 05C0                        32          BALR  12,0
                         00002     33          USING *,12,11
000002 41BC 0FFF                   34          LA    11,4095(12)
000006 90EC C6A6                   35          STM   14,12,SV2      SAVE CONTENT FOR USE
00000A 50D0 C6E6                   36          ST    13,SV2+64      SAVE CONTENT OF REG13
00000E 4580 C0C2                   37          BAL   8,GOCS
000012 5840 C32E                   38          L     4,=V(LOADLOC)  LOAD ADDR. FOR SORT ROUTINE
                                   39          LOAD  SORT,(4)       LOAD SORT FROM CIL.
00001E 18F1                        45          LR    15,1           LOAD ADDR. OF ENTRY POINT
000020 4110 C7C6                   46          LA    1,PARA4        LOAD ADDR. OF PARAMETER LIST ARRAY
000024 41D0 CEA6                   47          LA    13,SAVAREA     LOAD ADDR. OF SAVEAREA
000028 05EF                        48          BALR  14,15          BRANCH TO SORT
00002A D501 CEEE C34E 00FF0 00350  49          CLC   RETURN(2),=H'0' COMPARE RETURNING CODE
000030 4770 C06E       00070       50          BNE   SORTERR        IF ERROR GO TO SORTERR
                                   51          OPEN  SPRINT
000042 D283 C73E C73E 00741 00740  59          MVC   SPR,SPR-1      IF NOT,CLEAR PRINT
000048 D21A C749 C38A 0074B 0038C  60          MVC   SPR+10(27),=C'*** SORTING IS COMPLETE ***'
                                   61          CNTRL SPRINT,SK,1
                                   67          PUT   SPRINT,SPR
00006C 47F0 C0AA       000AC       73          B     ENDOJ
                                   74 SORTERR  OPEN  SPRINT
00007E D283 C73E C73E 00741 00740  82          MVC   SPR,SPR-1
000084 D21E C749 C3A5 0074B 003A7  83          MVC   SPR+10(31),=C'*** SORTING IS NOT COMPLETE ***'
                                   84          CNTRL SPRINT,SK,1
                                   90          PUT   SPRINT,SPR
0000A8 47F0 C0AA       000AC       96          B     ERRPT
                         000AC     97 ENDOJ    EQU   *
                         000AC     98 ERRPT    EQU   *
                                   99          CLOSE SPRINT
0000BA 98EC C6A6                  107          LM    14,12,SV2      RETURN CONTENT OF ALL REGS.
0000BE 58D0 C6E6                  108          L     13,SV2+64
0000C2 07FE                       109          BR    14             RETURN TO FORTRAN
                         000C4    110 GOCS     EQU   *
0000C4 D2FF C437 C436 00439 00438 111          MVC   BUFINP,BUFINP-1
0000CA D2FF C527 C437 00529 00439 112          MVC   BUFINPX,BUFINP
0000D0 4140 C7C6                  113          LA    4,PARA4
                                  114          OPEN  CD
                                  122 READ     GET   CD,DATAIN
0000F2 D5D4 C3E7 C3C4 003E9 003C6 129          CLC   DATAIN+1(5),=C'**JCS'
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
000DE8 4780 C0E0       000E2        129            BE    READ
000DEC D503 C3E7 C342 003E9 00344   130            CLC   DATAIN+1(4),=C'SORT'
000102 4780 C176       00178        131            BE    RSORT
000106 D505 C3E7 C350 003E9 00352   132            CLC   DATAIN+1(6),=C'RECORD'
00010C 4780 C186       00188        133            BE    RRECORD
000110 D505 C3E7 C356 003E9 00358   134            CLC   DATAIN+1(6),=C'INPFIL'
000116 4780 C196       00198        135            BE    RINPFL
00011A D505 C3E7 C35C 003E9 0035E   136            CLC   DATAIN+1(6),=C'OUTFIL'
000120 4780 C1A6       001A8        137            BE    ROUTFIL
000124 D505 C3E7 C362 003E9 00364   138            CLC   DATAIN+1(6),=C'OPTION'
00012A 4780 C1B6       001B8        139            BE    ROPTION
00012E D505 C3E7 C368 003E9 0036A   140            CLC   DATAIN+1(6),=C'ALTSEQ'
000134 4780 C1DC       001DE        141            BE    RALTSEQ
000138 D505 C3E7 C36E 003E9 00370   142            CLC   DATAIN+1(6),=C'OUTREC'
00013E 4780 C1EC       001EE        143            BE    ROUTREC
000142 D502 C3E7 C3C9 003E9 003CB   144            CLC   DATAIN+1(3),=C'SUM'
000148 4780 C1FC       001FE        145            BE    RSUM
00014C D506 C3E7 C3CC 003E9 003CE   146            CLC   DATAIN+1(7),=C'INCLUDE'
000152 4780 C20C       0020E        147            BE    RINCLUDE
000156 D503 C3E7 C346 003E9 00348   148            CLC   DATAIN+1(4),=C'OMIT'
00015C 4780 C20C       0020E        149            BE    ROMIT
000160 D503 C3E7 C34A 003E9 0034C   150            CLC   DATAIN+1(4),=C'MODS'
000166 4780 C1CC       001CE        151            BE    RMODS
00016A D504 C3E7 C3D3 003E9 003D5   152            CLC   DATAIN+1(5),=C'**END'
000170 4780 C2C6       002C8        153            BE    RET
000174 47F0 C21E       00220        154            B     ERROR
000178 4150 C7FE             00800  155 RSORT      LA    5,SC
00017C 4590 C24C             0024E  156            BAL   9,CONT
000180 5050 4000             00000  157            ST    5,0(0,4)
000184 47F0 C2BA       002BC        158            B     MOVE
000188 4150 C89E             00BA0  159 RRECORD    LA    5,RCD
00018C 4590 C24C             0024E  160            BAL   9,CONT
000190 5050 4004             00004  161            ST    5,4(0,4)
000194 47F0 C2BA       002BC        162            B     MOVE
000198 4150 C93E             00940  163 RINPFL     LA    5,INPFL
00019C 4590 C24C             0024E  164            BAL   9,CONT
0001A0 5050 4008             00008  165            ST    5,8(0,4)
0001A4 47F0 C2BA       002BC        166            B     MOVE
0001A8 4150 C9DE             009E0  167 ROUTFIL    LA    5,OUTFL
0001AC 4590 C24C             0024E  168            BAL   9,CONT
0001B0 5050 400C             0000C  169            ST    5,12(0,4)
0001B4 47F0 C2BA       002BC        170            B     MOVE
0001B8 4150 CA7E             00480  171 ROPTION    LA    5,OPT
0001BC 4590 C24C             0024E  172            BAL   9,CONT
0001C0 5050 4010             00010  173            ST    5,16(0,4)
0001C4 D2EF 5000 C527 00000 00529   174            MVC   0(240,5),BUFINPX
0001CA 47F0 C0E0       000E2        175            B     READ
0001CE 4150 CB6E             00870  176 RMODS     LA    5,MODS
0001D2 4590 C24C             0024E  177            BAL   9,CONT
0001D6 5050 4014             00014  178            ST    5,20(0,4)
0001DA 47F0 C2BA       002BC        179            B     MOVE
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT
0001DE 4150 CC0E             00C10   180 RALTSEQ  LA    5,A_TSQ
0001E2 4590 C24C             0024E   181          BAL   9,CONT
0001E6 5050 4028             00028   182          ST    5,40(0,4)
0001EA 47F0 C2BA       002BC         183          B     MOVE
0001EE 4150 CC8E             00C80   184 ROUTREC  LA    5,OUTRC
0001F2 4590 C24C             0024E   185          BAL   9,CONT
0001F6 5050 402C             0002C   186          ST    5,44(0,4)
0001FA 47F0 C2BA       002BC         187          B     MOVE
0001FE 4150 CD4E             00D50   188 RSUM     LA     5,SM
000202 4590 C24C             0024E   189          BAL   9,CONT
000206 5050 4030             00030   190          ST    5,48(0,4)
00020A 47F0 C2BA       002BC         191          B     MOVE
                             0020E   192 RINCLUDE EQU   *
00020E 4150 CDFE             00DF0   193 ROMIT    LA    5,INCOM
000212 4590 C24C             0024E   194          BAL   9,CONT
000216 5050 4034             00034   195          ST    5,52(0,4)
00021A 47F0 C2BA       002BC         196          B     MOVE
                                     197 ERROR    OPEN  SPRINT
00022E D283 C73E C73E 00741 00740    205          MVC   SPR,SPR-1
000234 D20F C749 C3D8 00748 003DA    206          MVC   SPR+10(11),=C'INVALID JCS'
                                     207          PUT   SPRINT,SPR
00024A 47F0 C0AA             000AC   213          B     ENDJJ
00024E 955C C42D             0042F   214 CONT     CLI   DATAIN+71,C'*'
000252 4780 C268             0026A   215          BE    CONTI
000256 D245 C437 C3E7 00439 003E9    216          MVC   BUFINP(70),DATAIN+1
00025C D2FF C527 C437 00529 00439    217          MVC   BUFINPX,BUFINP
000262 D2FF C437 C436 00439 00438    218          MVC   BUFINP,BUFINP-1
000268 07F9                          219          BR    9
00026A 4160 C437             00439   220 CONTI    LA    5,BUFINP
00026E 4170 C3E7             003E9   221          LA    7,DATAIN+1
000272 D245 6000 7000 00000 00000    222 CONTN    MVC   0(70,6),0(7)
000278 4166 0046             00046   223          LA    6,70(6)
                             0027C   224 NN       EQU   *
                                     225          GET   CD,DATAIN
00028C 4170 C3E6             003E8   231          LA    7,DATAIN
000290 D54F C42D 005C 0042F 0005C    232          CLC   DATAIN+71,C'*'
000296 4780 C2AC       002AE         233          BE    CONTN1
00029A D246 6000 C3E6 00000 003E8    234          MVC   0(71,6),DATAIN
0002A0 D2FF C527 C437 00529 00439    235          MVC   BUFINPX,BUFINP
0002A6 D2FF C437 C436 00439 00438    236          MVC   BUFINP,BUFINP-1
0002AC 07F9                          237          BR    9
0002AE D246 6000 7000 00000 00000    238 CONTN1   MVC   0(71,6),0(7)
0002B4 4166 0047             00047   239          LA    6,71(6)
0002B8 47F0 C27A             0027C   240          B     NN
0002BC D29F 5000 C527 00000 00529    241 MOVE     MVC   0(160,5),BUFINPX
0002C2 47F0 C0F0             000E2   242          B     READ
                                     243 RET    CLOSE CD
0002D6 07F8                          251          BR    8
                                     252 EOF      CLOSE CD
                                     260          OPEN  SPRINT
0002E6 D283 C73E C73E 00741 00740    269          MVC   SPR,SPR-1
```

```
LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                    DOS/VS ASSEMBLER REL 34.0 07.51 82-04-11

0002EC 0215 C749 C374 00748 00376   269          MVC   SPR+10(22),=C'END OF JCS IS REQUIRED'
                                    270          PUT   SPRINT,SPR
000312 47F0 C0AA      000AC         276      B     ENDJJ
000318                              277          LTORG
000318 E2D6D9E340404040             278                =CL8'SORT'
000320 5B5BC2D6D7C5D540             279                =C'$$BOPEN '
000328 5B5BC2C3D3D6E2C5             280                =C'$$BCLOSE'
000330 00000000                     281                =V(LOADLOC)
000334 00000EF8                     282                =A(SPRINT)
000338 00000741                     283                =A(SPR)
00033C 00000620                     284                =A(CD)
000340 000003E8                     285                =A(DATAIN)
000344 E2D6D9E3                     286                =C'SORT'
000348 D6D4C9E3                     287                =C'OMIT'
00034C D4D5C4E2                     288                =C'MODS'
000350 0000                         289                =H'0'
000352 D9C5C3D6D9C4                 290                =C'RECORD'
000358 C9D5D7C6C9D3                 291                =C'INFILE'
00035E D6E4E3C6C9D3                 292                =C'OUTFIL'
000364 D6D7E3C9D6D5                 293                =C'OPTION'
00036A C1D3E3E2C5D8                 294                =C'ALTSEQ'
000370 D6E4E3D9C5C3                 295                =C'OUTREC'
000376 5B5BC440D6C640D1             296                =C'END OF JCS IS REQUIRED'
000386 5C5C5C40E2D6D9E3             297                =C'*** SORTING IS COMPLETE ***'
000397 5C5C5C40E2D6D9E3             298                =C'*** SORTING IS NOT COMPLETE ***'
0003A6 5C5CD1C3E2                   299                =C'**JCS'
0003AB E2E4D4                       300                =C'SUM'
0003AE C9D5C3D3E4C4C5               301                =C'INCLUDE'
0003B5 5C5C5D5C4                    302                =C'**END'
0003BA C9D5E5C1D3C9C440             303                =C'INVALID JCS'
                                    304          EJJ
0003E8                              307 DATAIN   DS    CL80
000438 40                           308          DC    C' '
000439                              309 BUFINP   DS    CL240
000529                              310 BUFINPX  DS    CL240
                                    311 CD       DTFCD DEVADDR=SYS009,IOAREA1=INCARD,BLKSIZE=80,DEVICE=3505,    *
                                                       EOFADDR=EOF,ERROPT=SKIP,MODNAME=LOGICCD,WORKA=YES
000658                              334 INCARD   DS    CL80
                                    335 LOGICCD  COMOD DEVICE=3505,WORKA=YES
```

# ประวัติผู้เขียน

นางสาวภาวดี วรปัญญาวัฒนา เกิดวันที่ 17 พฤษภาคม พ.ศ. 2500 ที่จังหวัดกรุงเทพมหานคร จบการศึกษาชั้นมัธยมศึกษาตอนต้น จากโรงเรียนสายปัญญาในปีการศึกษา 2516 ชั้นมัธยมศึกษาตอนปลายสายวิทยาศาสตร์ จากโรงเรียนเบญจมราชาลัย ในปีการศึกษา 2518 และการศึกษาในระดับปริญญาตรีจากคณะครุศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2522