บทที่ 7

การจัดพิมพ์เอกสาร

เมื่อได้เขียนโปรแกรมไปแล้ว ก็มีความจำเป็นที่จะต้องจัดพิมพ์โปรแกรมเหล่านั้นออกมาเป็นเอกสารประกอบ ซึ่งอาจจะนำไปใช้ตรวจสอบการทำงาน ใช้อ้างอิง ใช้สื่อสารกับบุคคลอื่น ฯลฯ ก็แล้วแต่ผู้ใช้ ทั้งนี้เอกสารที่จัดพิมพ์ออกมาจะต้องชัดเจน ถูกต้องครบถ้วน โดยเฉพาะการพิมพ์ชาร์ต ซึ่งเป็นรูปภาพควรจะจัดพิมพ์ให้มีลักษณะใกล้เคียงกับภาพที่ปรากฏบนจอภาพให้มากที่สุด เอกสารต่างๆที่จะจัดพิมพ์ ได้แก่

1. ชาร์ตที่ผู้ใช้เขียนขึ้น : เนื่องจากชาร์ตมีขนาดใหญ่ จึงจัดพิมพ์แยกออกเป็น 4 หน้า

2. โปรแกรมภาษานีมอนิคที่แปลได้ : เป็นผลลัพธ์จากการแปลชาร์ต ในการพิมพ์ได้จัดรูปแบบให้สามารถลดจำนวนหน้าลงได้ 4 เท่า

3. สมการแสดงความสัมพันธ์หรือสมการบูลลีน : แสดงความสัมพันธ์ของสเต็ปและทรานสิชั่นต่างๆ สำหรับนำไปประยุกต์ใช้งานกับฮาร์ดแวร์ประเภทอื่นๆ

4. ตารางแสดงสเต็ปและทรานสิชั่นภายในชาร์ต : แสดงรายละเอียดต่างๆของทุกๆสเต็ปและทรานสิชั่น

5. ตารางแสดงการนำรีเลย์ช่วยมาใช้งาน : แสดงการเข้าคู่ของรีเลย์ช่วยกับ I/O

7.1 การจัดการกับเครื่องพิมพ์

ก่อนที่จะออกแบบโปรแกรมจะต้องทำความเข้าใจกับการนำเครื่องพิมพ์มาใช้งานก่อน เนื่องจากเป็นอุปกรณ์ที่จะใช้พิมพ์เอกสารต่างๆ ในงานวิจัยนี้ได้เลือกใช้เครื่องพิมพ์ของ Epson รุ่น LQ1050 , LQ550[1] และ HP Laserjet III[2,3] เพราะมีอยู่ภายในห้องปฏิบัติการ การศึกษาเครื่องพิมพ์ทั้งสองยี่ห้อจะกระทำไปพร้อมๆกัน

การจัดกระบวนการตั้งต้น

จากการศึกษาคู่มือของเครื่องพิมพ์เหล่านั้นพบว่าสามารถส่งคำสั่งไปควบคุมการทำงานได้โดยตรง โดยชุดคำสั่งของ LQ1050 และ LQ550 เป็นชุดคำสั่งเดียวกัน ทั้งชุดคำสั่งของ EPSON (LQ) และ HP ต่างก็จะต้องขึ้นต้นด้วยรหัส

---

1. Epson. LQ-850 and LQ-1050 24 pins dot matrix printers. Reference guide. Nagono, Japan : 1989. pp. 49-112.

2. Hewlett Packard. Laser Jet III printer technical reference manual. U.S.A. : 1990. pp. 6-1 - 6-27.

3. Hewlett Packard. LaserJet III printer user manual. West Germany : 1990. pp. B-1 - B-13.

"Escape (#27)" ด้วยกันทั้งนั้น และเรียกชุดคำสั่งเหล่านี้ว่า "Escape command" คำสั่งต่างๆที่จะนำมาใช้ ได้แก่

1. คำสั่งจัดกระบวนการตั้งต้น (Initialize) : จัดกระบวนการตั้งต้นของเครื่องพิมพ์ใหม่

LQ : Esc, @

HP : Esc, E

2. คำสั่งจัดระยะห่างระหว่างบรรทัด (Line feed)

LQ : Esc, +, n (n/360 inch line feed)

HP : Esc, &, l, n, c (n/48 inch VMI (Vertical Movement Index))

3. คำสั่งเลือกใช้ตัวอักษรกราฟฟิก : เพื่อใช้พิมพ์ชาร์ต

LQ : Esc, t, n (n=1)

HP : Esc, (, 1, 0, U (เลือกใช้ชุดตัวอักษร PC-8 ซึ่งมีตัวอักษรกราฟฟิก)

4. คำสั่งเลือกประเภทของตัวอักษร

LQ : Esc, k, n (n=1, San serif)

HP : Esc, (, s, 4, 1, 4, 8, T (Universe Typeface)

5. คำสั่งเลือกใช้กระดาษขนาด A4

HP : Esc, &, l, 2, 6, A

6. คำสั่ง Form feed : มีได้อยู่ใน Escape command เพราะเป็นมาตรฐานของเครื่องพิมพ์

LQ, HP : #12

สถานะของเครื่องพิมพ์

ในการส่งข้อมูลไปพิมพ์ยังเครื่องพิมพ์ จะเกิดการสื่อสารระหว่างไมโครคอมพิวเตอร์กับเครื่องพิมพ์ ดังนั้นจะต้องเข้าใจสถานะต่างๆของเครื่องพิมพ์ให้ชัดเจน เพื่อข้อมูลที่ส่งไปจะได้ไม่สูญหาย ตกหล่น ในโปรแกรม BIOS มีฟังก์ชั่นหนึ่งซึ่งจะช่วยตรวจสอบสถานะของเครื่องพิมพ์ให้ ได้แก่การเรียกใช้ Interrupt 17 ฟังก์ชั่นที่ 2 ซึ่งจะต้องตั้งค่าพารามิเตอร์ก่อนดังนี้ Register AH = 02H (Read printer status)

Register DX = ตำแหน่งของเครื่องพิมพ์ 0000 = LPT1

0001 = LPT2

จากนั้นเรียกใช้ Interrupt 17 ก็จะได้รับสถานะของเครื่องพิมพ์กลับมา ซึ่งอยู่ใน รีจิสเตอร์ AH สามารถตีความหมายได้ดังนี้

บิตที่ 0 : Time out, หมดเวลาการรอคอย

บิตที่ 3 : I/O error เกิดปัญหาในการสื่อสาร

บิตที่ 4 : Selected, เครื่องพิมพ์นี้ถูกเลือก

บิตที่ 5 : No paper, ไม่มีกระดาษ

บิตที่ 6 : Acknowledge, ได้รับข้อมูลแล้ว

บิตที่ 7 : Ready, พร้อมที่จะรับข้อมูลต่อไป

ในที่นี้จะให้ความสนใจใน 4 บิต ได้แก่ บิตที่ 0, 3, 5, 7 จากการทดสอบให้เครื่องพิมพ์อยู่ในสถานะต่างๆ และทำการอ่านสถานะต่างๆขึ้นมา สามารถสรุปได้เป็นตารางดังนี้

บิต				สถานะ
7	5	3	0	
1	0	1	0	ปิดเครื่องพิมพ์
0	1	1	0	เปิด, ไม่ on line, ไม่มีกระดาษ
0	0	1	0	เปิด, แต่ไม่ on line, มีกระดาษ
1	0	0	0	เปิด, on line, มีกระดาษ

ตารางที่ 7.1 : สถานะของเครื่องพิมพ์

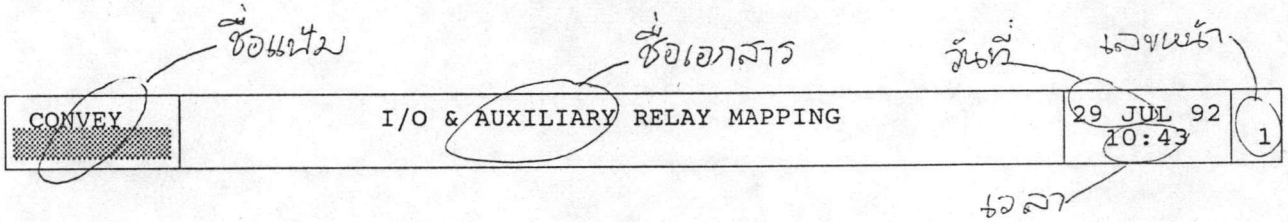
ได้เขียนโปรแกรมย่อยขึ้นมา 2 โปรแกรม ได้แก่ ReadPrinterStatus และ PrinterHandle โดยโปรแกรมย่อย ReadPrinterStatus จะเรียกใช้ Interrupt 17 เพื่อตรวจสอบสถานะของเครื่องพิมพ์ และจะวิเคราะห์สถานะของบิตที่ 0, 3, 5, 7 เพื่อรายงานกลับไปให้โปรแกรม PrinterHandle ซึ่งเรียกใช้โปรแกรมนี้ Printerhandle จะวิเคราะห์สถานะของเครื่องพิมพ์อีกครั้งหนึ่งตามตารางที่ 7.1 พร้อมกับรายงานให้ผู้ใช้ทราบหากเกิดปัญหาในการพิมพ์

```
Procedure ReadPrinterStatus(Var Busy,NoPaper,IOerror,TimeOut : boolean);
Var Reg         : Registers;
    Result,Code : Byte;
Begin
        Reg.AH:=02;         {Read status}
        Reg.DX:=0000;       {From PRN (LPT1)}
        Intr($17,Reg);      {BIOS Printer service}
        Code:=Reg.AH;

        Result:=Code and 128;                   {Test bit 7}
        If Result=0   then   Busy:=true;
        If Result=128 then   Busy:=false;

        Result:=Code and 32;                    {Test bit 5}
        If Result=0  then    NoPaper:=false;
        If Result=32 then    NoPaper:=true;

        Result:=Code and 8;                     {Test bit 3}
        If Result=0  then    IOerror:=false;
        If Result=8  then    IOerror:=true;

        Result:=Code and 1;                     {Test bit 0}
        If Result=0  then    TimeOut:=false;
        If Result=1  then    TimeOut:=true;
End;
```

รูปที่ 7.1 : โปรแกรมย่อย ReadPrinterStatus

7.2 รูปแบบของเอกสาร

เพื่อให้เกิดความเป็นระเบียบในกลุ่มเอกสารจึงได้ออกแบบให้ทุกๆเอกสารจะต้องพิมพ์ส่วนหัวของเอกสารตามรูปแบบที่ได้กำหนดร่วมกันไว้ ดังนี้



รูปที่ 7.2 : ส่วนหัวของเอกสาร

ช่องที่ 1 : คือ ชื่อแฟ้มข้อมูลของชาร์ตที่จัดทำเอกสารอยู่

ช่องที่ 2 : ชื่อของเอกสาร ได้แก่

- FUNCTION CHART
- GENERATED MNEMONIC
- EQUATION OF RELATION
- STEP & TRANSITION LIST
- I/O & AUXILIARY RELAY MAPPING

ช่องที่ 3 : วันที่ และเวลาขณะที่พิมพ์เอกสาร

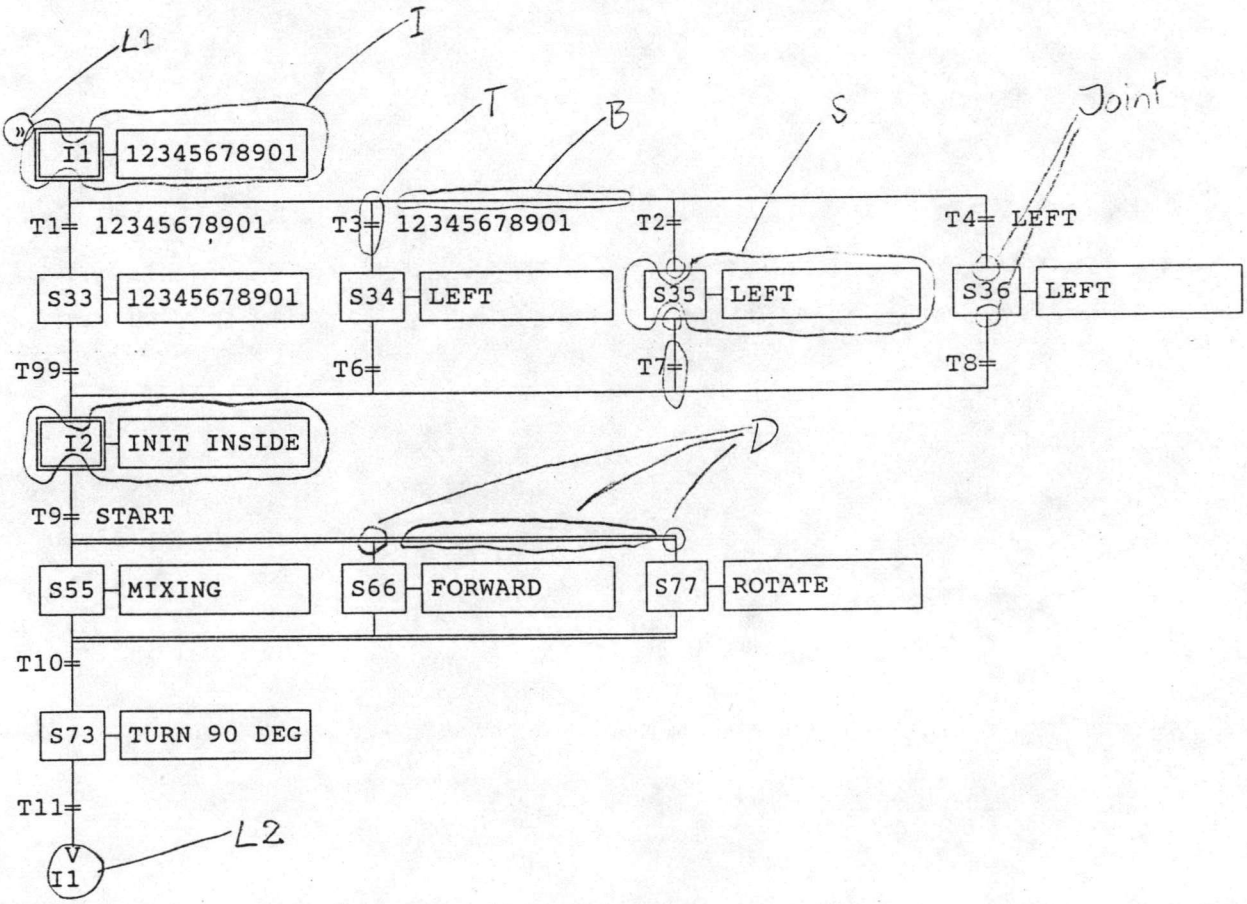
ช่องที่ 4 : หมายเลขหน้า

ได้เขียนโปรแกรมย่อย DateTime เพื่อทำหน้าที่หาค่าวันและเวลาขณะนั้น โดยจะเรียกใช้ DOS Function หมายเลข 2A เพื่อถามวันที่ และเรียกใช้ฟังก์ชั่นหมายเลข 2C เพื่อถามเวลา โปรแกรมย่อย PrintHeader จะจัดการพิมพ์ส่วนหัวของเอกสารทุกฉบับ

7.3 การพิมพ์ภาพชาร์ต

7.3.1 เทคนิคการพิมพ์

เนื่องจากชาร์ตเป็นภาพวาด จึงมี 2 แนวทางที่จะพิมพ์ชาร์ตได้ คือ พิมพ์ทีละจุดหรือใช้ตัวอักษรกราฟฟิก แบบหลังดูจะง่ายกว่าแต่จะต้องพิจารณาก่อนว่ามีตัวอักษรกราฟฟิกที่จะนำมาใช้แทนภาพต่างๆได้ครบหรือไม่ จึงนำภาพตัวอย่างชาร์ตที่ประกอบด้วยองค์ประกอบต่างๆ ครบถ้วนมาพิจารณาเทียบกับตัวอักษรกราฟฟิกต่างๆ



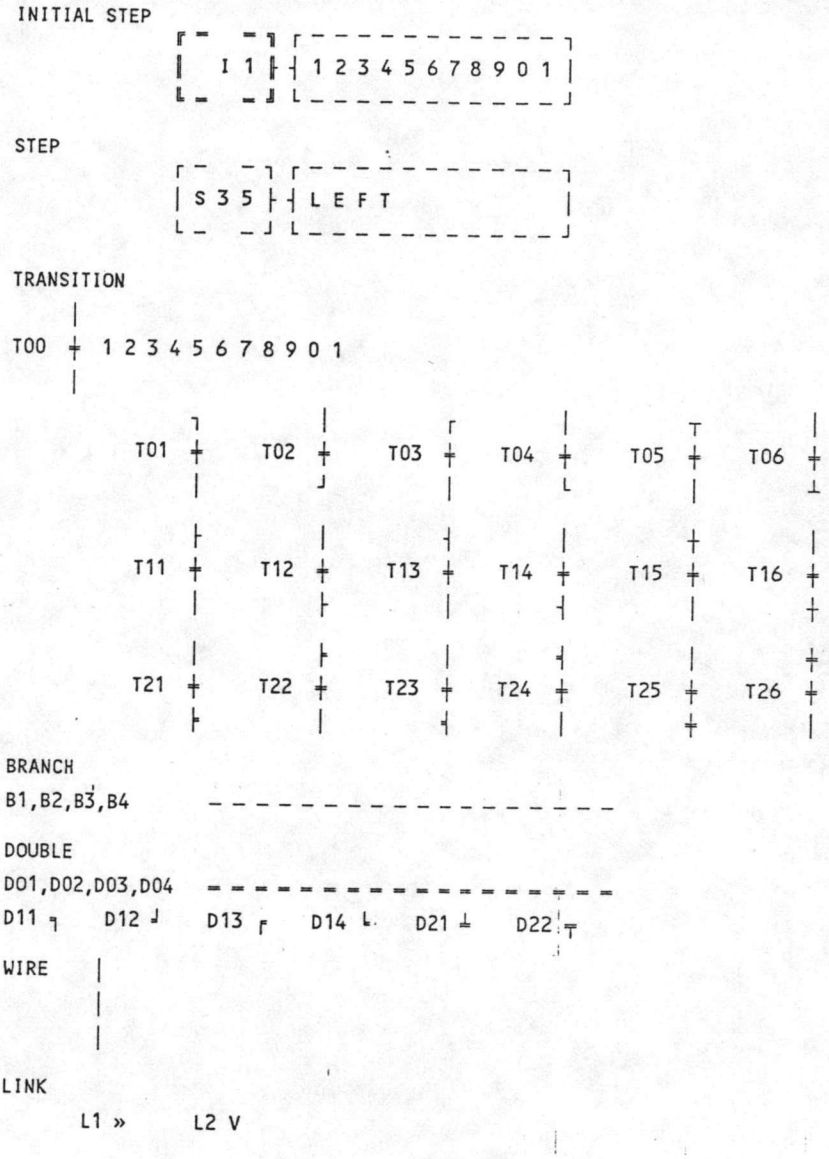
รูปที่ 7.3 : วิเคราะห์ภาพชาร์ตเพื่อการพิมพ์เอกสาร

DEC	HEX	CHAR	DEC	HEX	CHAR
160	A0	á	192	C0	└
161	A1	í	193	C1	┴
162	A2	ó	194	C2	┬
163	A3	ú	195	C3	├
164	A4	ñ	196	C4	─
165	A5	Ñ	197	C5	┼
166	A6	ª	198	C6	╞
167	A7	º	199	C7	╟
168	A8	¿	200	C8	╚
169	A9	⌐	201	C9	╔
170	AA	¬	202	CA	╩
171	AB	½	203	CB	╦
172	AC	¼	204	CC	╠
173	AD	¡	205	CD	═
174	AE	«	206	CE	╬
175	AF	»	207	CF	╧
176	B0	░	208	D0	╨
177	B1	▒	209	D1	╤
178	B2	▓	210	D2	╥
179	B3	│	211	D3	╙
180	B4	┤	212	D4	╘
181	B5	╡	213	D5	╒
182	B6	╢	214	D6	╓
183	B7	╖	215	D7	╫
184	B8	╕	216	D8	╪
185	B9	╣	217	D9	┘
186	BA	║	218	DA	┌
187	BB	╗	219	DB	▉
188	BC	╝	220	DC	▄
189	BD	╜	221	DD	▌
190	BE	╛	222	DE	▐
191	BF	┐	223	DF	▀

รูปที่ 7.4 : ตัวอักษรกราฟฟิกที่นำมาใช้ได้

7.3.2 <u>วิเคราะห์ภาพองค์ประกอบ</u>

นำภาพแต่ละองค์ประกอบจากรูปที่ 7.3 มาวิเคราะห์แยกให้เป็นการนำตัวอักษรกราฟฟิกมาใช้งาน ได้ดังต่อไปนี้

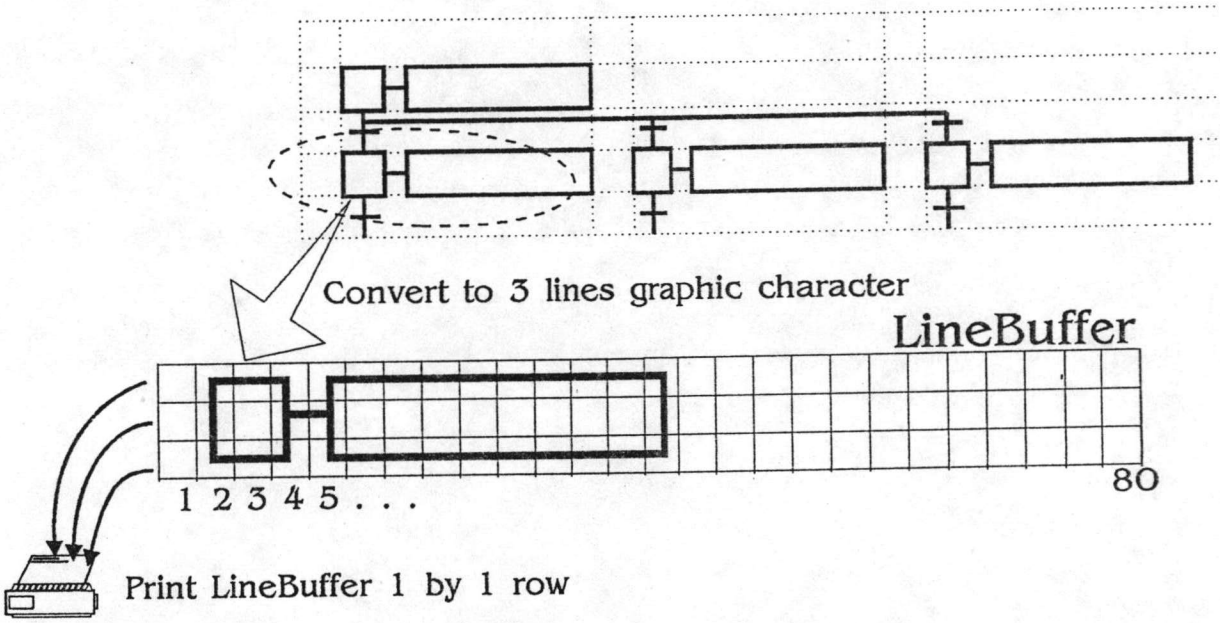


รูปที่ 7.5 : ผลจากการวิเคราะห์

จะเห็นว่าทุกๆองค์ประกอบ สามารถแทนได้ด้วยตัวอักษร ดังนั้นในการพิมพ์จึงเลือกใช้วิธีแปลงภาพซาร์ตให้เป็นตัวอักษรกราฟฟิกก่อนแล้วค่อยส่งพิมพ์ ซึ่งก็คือการพิมพ์แฟ้มข้อมูลประเภท Text file นั่นเอง จึงทำให้การพิมพ์ซาร์ตพิมพ์ได้รวดเร็ว

7.3.3 การแปลงภาพชาร์ตโดยใช้ตัวอักษรกราฟฟิก

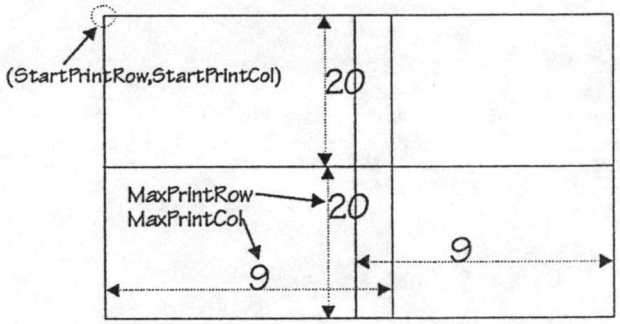
ในการพิมพ์จะอ่านชาร์ตขึ้นมา 1 แถว และแปลงไปเป็นตัวอักษรกราฟฟิก จากรูปที่ 7.5 จะเห็นว่ารูปชาร์ต 1 แถว จะแปลงได้เป็นตัวอักษรกราฟฟิก 3 บรรทัด ในการแปลงจึงได้เตรียมตัวแปรไว้ 1 ตัว (LineBuffer) ซึ่งมีโครงสร้างเป็นแถวลำดับ (Array) ขนาด 3 แถว x 80 คอลัมน์ แต่ละแถวของตัวแปรนี้จะรับค่าในการแปลงของตัวอักษรกราฟฟิกแต่ละบรรทัด และมี 80 คอลัมน์ ซึ่งเท่ากับความกว้างของหน้ากระดาษนั่นเอง และเมื่อแปลงชาร์ตเสร็จ 1 แถว ก็จะส่งตัวแปรนี้ไปพิมพ์ ทำเช่นนี้สลับกันจนหมดชาร์ต โปรแกรมย่อยที่ทำหน้าที่แปลงชาร์ต 1 แถว ไปเป็นตัวอักษรกราฟฟิก 3 บรรทัด และจัดเก็บในตัวแปร LineBuffer คือโปรแกรมย่อย AssemblyLineBuffer



รูปที่ 7.6 : การแปลงชาร์ตไปเป็นตัวอักษรกราฟฟิก

7.3.4 แบ่งชาร์ตพิมพ์ 4 หน้า

เนื่องจากชาร์ตมีขนาดใหญ่เกินกว่าที่จะจัดพิมพ์ลงใน 1 หน้าได้ และเพื่อไม่ให้ภาพชาร์ตที่พิมพ์เล็กเกินไป จึงจัดพิมพ์ชาร์ตเป็น 4 หน้ากระดาษ A4 โดยแบ่งชาร์ตเป็น 4 ส่วน ส่วนละ 20 แถว และ 9 คอลัมน์ (คอลัมน์ที่ 9 ใช้ร่วมกัน) ดังนั้นในการแปลงชาร์ต 1 แถว ก็จะแปลงเพียง 9 คอลัมน์ท่านั้น ในการควบคุมขอบเขตการแปลงชาร์ตไปพิมพ์ให้ได้ 1 หน้า จะมีตัวแปรควบคุมขอบเขตดังแสดงในรูปที่ 7.7



รูปที่ 7.7 : แบ่งชาร์ตเป็น 4 ส่วน

ลำดับการพิมพ์จะไล่จากบนซ้าย ไปบนขวา ไปล่างซ้าย และไปล่างขวา เมื่อจะพิมพ์ส่วนใด ตัวแปร StartPrintRow และ StartPrintCol ก็จะไปชี้ที่มุมซ้ายของส่วนนั้น เพื่อจะได้เริ่มแปลงจากตำแหน่งที่ถูกต้องและแปลงต่อไปเป็นจำนวน MaxPrintCol ต่อ 1 แถว โดยแปลงทั้งหมด MaxPrintRow แถว การกำหนดเลขหน้าของ ชาร์ตแต่ละส่วนจะใช้วิธีแสดงส่วนที่พิมพ์โดยตรง ดังนี้

Upper Left

Upper Right

Lower Left

Lower Right

รูปที่ 7.8 : เลขหน้าของชาร์ต

เนื่องจากในการวาดชาร์ต จะเริ่มวาดจากซ้ายบนก่อน ในบางครั้งอาจจะใช้พื้นที่เฉพาะ ในส่วนซ้ายบนหรือส่วนซ้ายบนกับส่วนซ้ายล่าง หรือส่วนซ้ายบนกับส่วนขวาบน ดังนั้นหากจัดพิมพ์ชาร์ตไปทุกส่วนก็จะมีบางหน้าที่ว่างเปล่า จึงได้เขียนโปรแกรมย่อย FindPrintFrame เพื่อหาขอบล่างและขอบขวาของภาพ เพื่อในการพิมพ์จะได้พิมพ์เฉพาะส่วนที่มีรูปเท่านั้น ตัวอย่างชาร์ตที่พิมพ์แสดงในภาคผนวก ง

7.4 การพิมพ์โปรแกรมนีมอนิคที่แปลได้

7.4.1 รูปแบบ

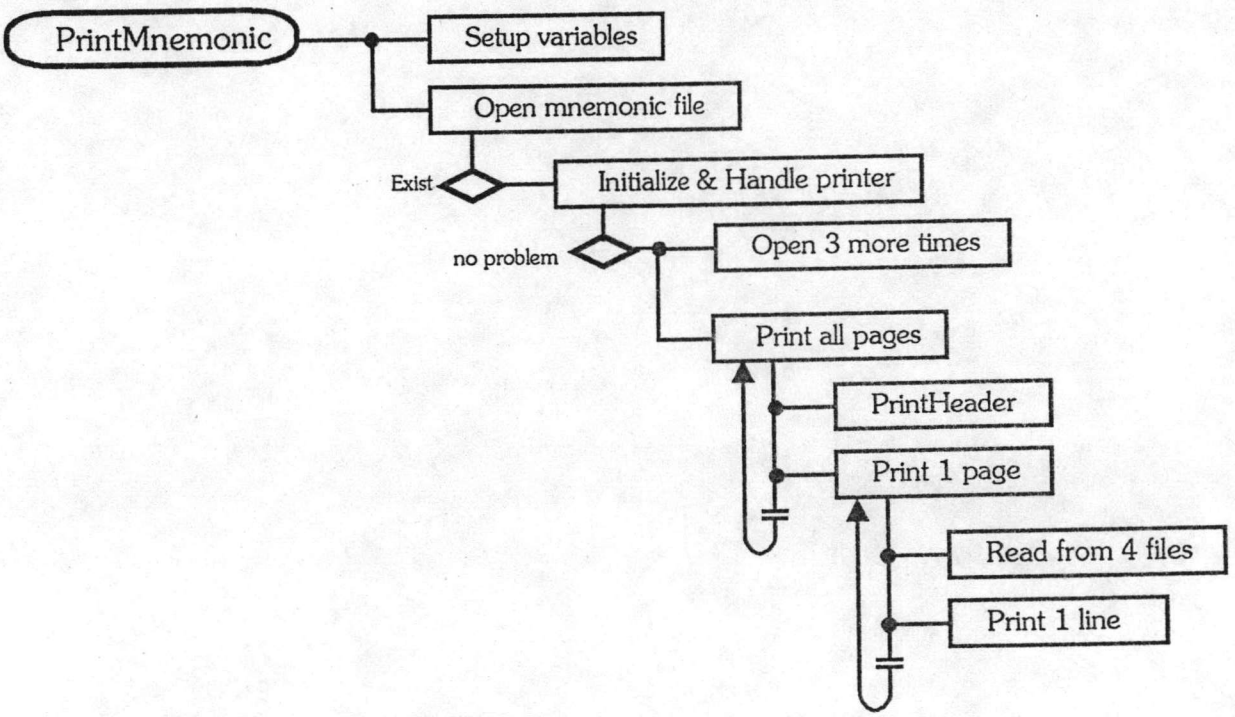
เนื่องจากโปรแกรมภาษานีมอนิคเป็นโปรแกรมที่ใช้เนื้อที่ในแนวคอลัมน์น้อย แต่ใช้จำนวนแถวมาก หากพิมพ์ออกไปโดยตรง นอกจากจะใช้จำนวนหน้ามากแล้ว ในการอ่านโปรแกรมจะต้องเสียเวลาพลิกกลับไปมาค่อนข้าง

มากด้วยจึงออกแบบให้ใน 1 หน้ากระดาษพิมพ์โปรแกรมนีมอนิคจำนวน 4 คอลัมน์ คอลัมน์ละ 60 บรรทัด โดยพิมพ์จากบนลงล่างและจากซ้ายไปขวา

ส่วนหัวของแต่ละหน้าจะใช้รูปแบบร่วมที่ได้ออกแบบไว้ในรูปที่ 7.2 โดยใช้ชื่อเอกสารว่า "GENERATED MNEMONIC"

7.4.2 โปรแกรมที่ออกแบบ

เทคนิคที่ใช้ในการพิมพ์โปรแกรมนีมอนิคพร้อมกัน 4 คอลัมน์ ก็คือการเปิดแฟ้มข้อมูลเดียวกัน 4 ครั้ง ซึ่งจะเสมือนการใช้งานแฟ้มข้อมูล 4 แฟ้มพร้อมกัน จากนั้นจัดตัวชี้ตำแหน่งของข้อมูล 4 ตัว ชี้ข้อมูลตำแหน่งต่างๆที่สอดคล้องกันในแต่ละแฟ้ม และอ่านข้อมูลจากแฟ้มทั้ง 4 ณ ตำแหน่งตัวชี้ขึ้นมาพิมพ์ ทั้งนี้หากตัวชี้ข้อมูลของแฟ้มใดชี้ที่จุดสิ้นสุดของข้อมูลของแฟ้มนั้น จะหยุดพิมพ์ข้อมูลจากแฟ้มนั้นทันที และจะพิมพ์เอกสารหน้านั้นเป็นหน้าสุดท้าย โปรแกรมย่อย PrintMnemonic จะทำหน้าที่นี้ ตัวอย่างการพิมพ์ดูได้จากภาคผนวก ง



รูปที่ 7.9 : DSD ของโปรแกรมย่อย PrintMnemonic

7.5 การพิมพ์สมการแสดงความสัมพันธ์

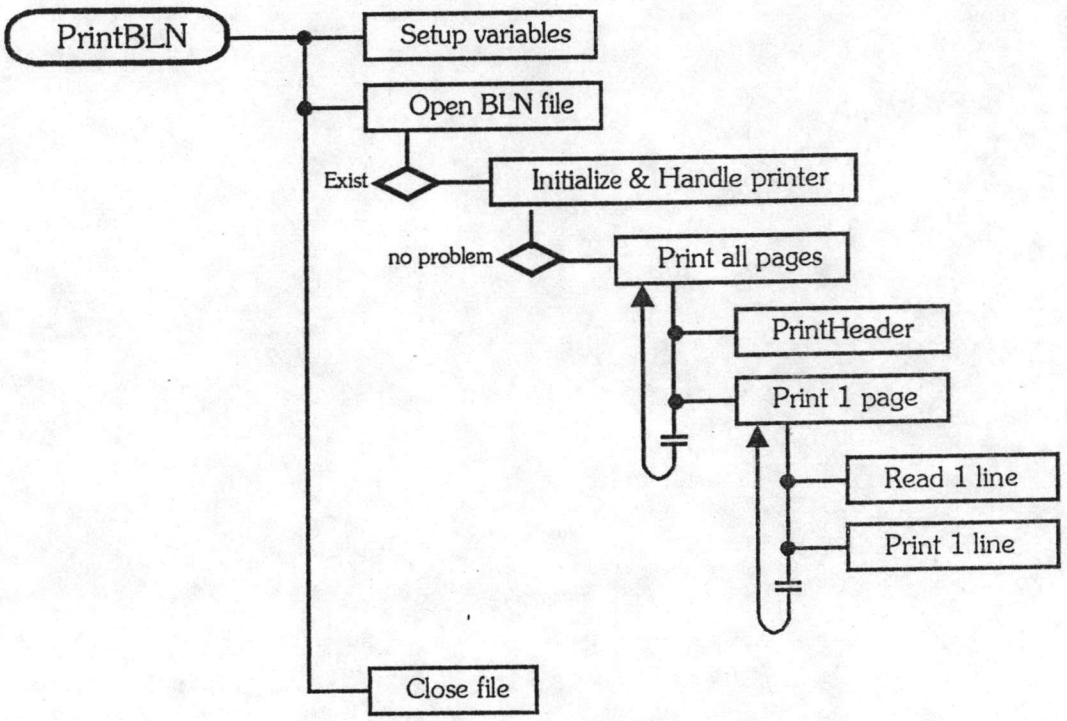
7.5.1 รูปแบบ

รูปแบบการพิมพ์จะพิมพ์บรรทัดละ 1 สมการ และพิมพ์บรรทัดเว้นบรรทัด เพื่อให้อ่านได้ไม่ยาก ใน 1

30 บรรทัด ส่วนหัวของแต่ละหน้าจะใช้รูปแบบตามรูปที่ 7.2 และใช้ชื่อเอกสารว่า "EQUATION OF RELATION"

7.5.2 โปรแกรมที่ออกแบบ

ในการพิมพ์สมการบูลลีนนี้ จะนำข้อมูลมาจากแฟ้มข้อมูลที่มีนามสกุล "BLN" โปรแกรมย่อยที่ควบคุมการพิมพ์ ได้แก่ PrintFileBLN ซึ่งมีการทำงานดังแผนภาพในรูป 7.10



รูปที่ 7.10 : DSD ของโปรแกรมย่อย PrintFileBLN

7.6 การพิมพ์ตารางแสดงสเต็ปและทรานสิชั่นภายในชาร์ต

7.6.1 รูปแบบ

การพิมพ์ตารางนี้ จะวิเคราะห์จากภาพชาร์ตว่าประกอบด้วยสเต็ปเริ่มต้น สเต็ปและทรานสิชั่นใดบ้าง จากนั้นจะจัดกลุ่มขององค์ประกอบเหล่านั้นและพิมพ์รายละเอียดต่างๆลงในตาราง ซึ่งมี 6 คอลัมน์ โดยมีรายละเอียดดังนี้

คอลัมน์ที่ 1 : ชื่อประเภทขององค์ประกอบจะแสดงเพียงครั้งเดียวที่บรรทัดเริ่มต้นขององค์ประกอบนั้น

คอลัมน์ที่ 2 : ชื่อหมายเลขขององค์ประกอบนั้น

คอลัมน์ที่ 3 : รายละเอียดหรือหมายเหตุขององค์ประกอบนั้นที่ผู้ใช้ได้ป้อนไว้

คอลัมน์ที่ 4 : I/O, หน้าสัมผัสอินพุตหรือเอาต์พุตที่ผู้ใช้กำหนด

คอลัมน์ที่ 5 : หมายเลขรีเลย์ช่วยที่ระบบกำหนดให้

คอลัมน์ที่ 6 : แสดงประเภทของอินพุตหรือเอาต์พุตนั้น ได้แก่

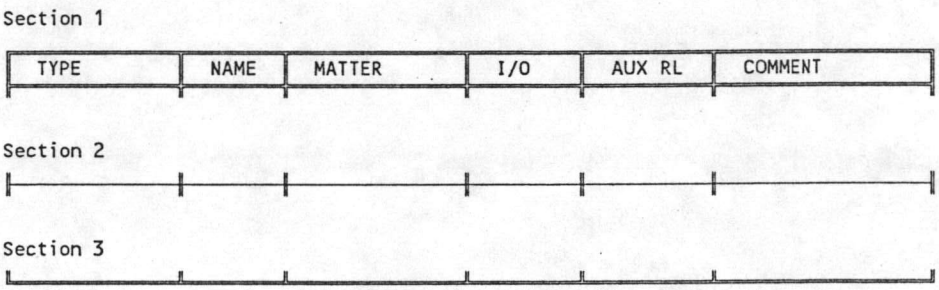
- เอาต์พุตปกติ : ไม่แสดงข้อความใด
- เอาต์พุตคงค่าไว้ : HSET
- เอาต์พุตที่ถูกรีเซ็ท : HRST
- ค่าคงที่ของตัวตั้งเวลา : ตัวเลข 4 หลัก (ฐาน 10)
- ค่าคงที่ของตัวนับ : ตัวเลข 4 หลัก (ฐาน 10)
- ตัวนับที่ถูกรีเซ็ท : CRST
- อินพุตปกติ : ไม่แสดงข้อความใด
- นำนิเสธของอินพุตไปใช้ : I

TYPE	NAME	MATTER	I/O	AUX RL	COMMENT
INITIAL	I1	CONVEYORS		3200	
	I2	MAIN MOTORS		3202	
	I3	LINE 1		3203	
	I4	STOP MOTOR1		3204	
	I5	LINE 2		3510	
	I6	STOP MOTOR2		3511	
	I7	LINE 3		3512	
	I8	STOP MOTOR3		3513	
STEP	S1	START MAIN	2000	3205	
	S11	MOTOR 6 ON	0206	3211	HSET
	S21	MOTOR 6 OFF	0206	3212	HRST
	S32	MOTOR 1 ON	0201	3213	HSET
	S33	P1 ON	0211	3214	HSET
	S41			3215	

รูปที่ 7.11 : ตัวอย่างตารางแสดงสเต็ปและทรานสิชั่น

7.6.2 โปรแกรมที่ออกแบบ

เริ่มต้นของหน้าจะพิมพ์หัวของเอกสารก่อนตามรูปแบบในรูปที่ 7.2 และใช้ชื่อเอกสารว่า "STEP & TRANSITION LIST" โดยเรียกใช้โปรแกรมย่อย PrintHeader ภาพตารางสามารถแยกแบ่งได้ดังนี้



รูปที่ 7.12 : แบ่งตารางเป็นส่วนๆ

ส่วนแรกเป็นหัวของตารางจะพิมพ์เพียงครั้งเดียวใน 1 หน้า เขียนโปรแกรมย่อย SendTheader เพื่อเก็บภาพส่วนหัวไว้ และสั่งพิมพ์ทุกๆครั้งที่ขึ้นหน้าใหม่ จากนั้นเป็นส่วนของเนื้อความที่จะพิมพ์ จัดการด้วยโปรแกรมย่อย BodyPrint ซึ่งจะต้องระบุว่าจะให้พิมพ์เนื้อความขององค์ประกอบใด ซึ่งโปรแกรมก็จะไปค้นหาองค์ประกอบนั้นทั้งหมดจากชาร์ต มาพิมพ์ลงตรงส่วนนี้ เมื่อพิมพ์เสร็จจะขีดเส้นใต้ (ส่วนที่ 2 ในรูปที่ 7.12) หรืออาจจะปิดท้ายตาราง (ส่วนที่ 3 ในรูปที่ 7.12) ทั้งนี้ขึ้นอยู่กับว่ามีองค์ประกอบใดที่จะต้องพิมพ์ต่อไปหรือไม่

7.7 การพิมพ์ตารางแสดงการนำรีเลย์ช่วยมาใช้งาน

7.7.1 รูปแบบ

ตารางนี้จะแสดงให้ทราบว่า ระบบได้ใช้รีเลย์ช่วยหมายเลขใดบ้าง และใช้คู่กับอินพุตหรือเอาต์พุตใด ซึ่งจะช่วยในการอ่านโปรแกรมนีมอนิค สามารถทำความเข้าใจได้เร็วขึ้น ตารางที่พิมพ์จะมี 2 คอลัมน์ คอลัมน์แรกแสดงหมายเลขรีเลย์ช่วย คอลัมน์ที่ 2 แสดงหมายเลขของอินพุตหรือเอาต์พุตที่จับคู่กับรีเลย์ช่วยตัวนั้น หากในสเต็ปหรือทรานสิชั่นใดมิได้ใช้ I/O (Input/Output) ในช่องช้างๆของรีเลย์ช่วยประจำสเต็ปหรือทรานสิชั่นนั้น ก็จะเว้นว่างไว้ ใน 1 หน้าจะพิมพ์พร้อมๆกัน 4 ตาราง เพื่อประหยัดจำนวนหน้ากระดาษ

AUX	I/O
3200	
3204	
3208	2002
3212	0206
3300	2002
3301	0005
3304	

AUX	I/O
3201	0000
3205	2000
3209	2003
3213	0201
6113	1
6113	1
3305	T011

AUX	I/O
3202	
3206	2000
3210	2001
3214	0211
6113	1
3302	T000
3306	

AUX	I/O
3203	
3207	2008
3211	0206
3215	
6113	1
3303	T004
3307	T000

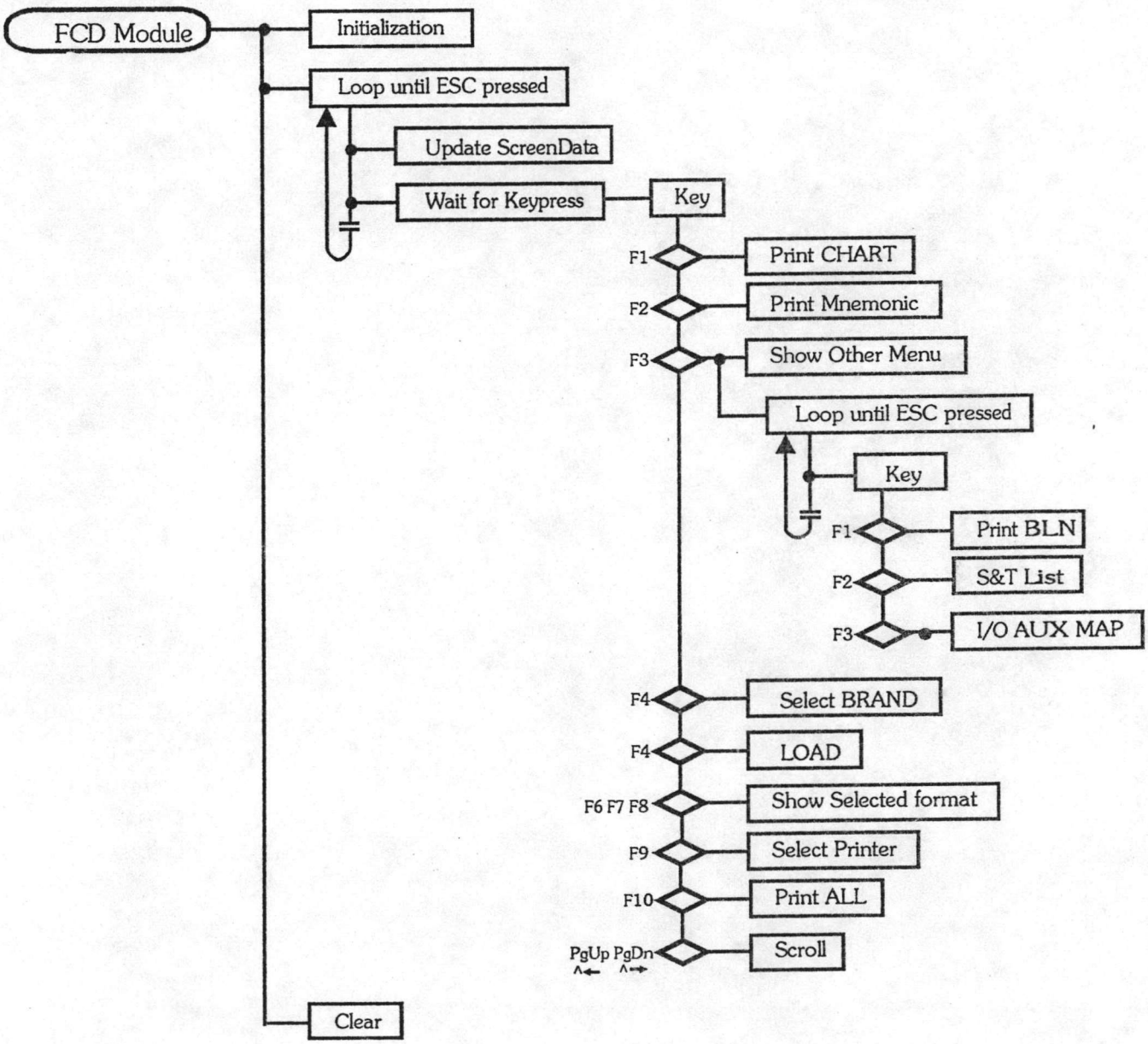
รูปที่ 7.13 : ตารางแสดงการนำรีเลย์ช่วยมาใช้งาน

7.7.2 โปรแกรมที่ออกแบบ

เช่นเดียวกับเอกสารอื่น ในส่วนหัวของเอกสารจะพิมพ์ตามรูปแบบที่กำหนด โดยเรียกใช้โปรแกรมย่อย PrintHeader ชื่อเอกสาร คือ "I/O & AUXILIARY RELAY MAPPING" ภาพตารางจะแบ่งเป็น 3 ส่วน ได้แก่ ส่วนหัว, ส่วนเนื้อความ และส่วนปิดท้าย โปรแกรมย่อย PrintIOAUX จะจัดการพิมพ์ตารางนี้ โดยภายในมีโปรแกรมย่อย IOAUXheader เพื่อจัดการการพิมพ์ส่วนหัวของตาราง จากนั้นจะไปค้นหาสเต็ปและทรานสิชั่นทุกๆตัว เพื่อนำค่ารีเลย์ช่วย และอินพุตหรือเอาต์พุตมาพิมพ์ลงตาราง

7.8 โปรแกรมหลัก

โปรแกรมที่ทำหน้าที่จัดการงานในโมดูลการพิมพ์นี้ (Function Chart Document, FCD) มีชื่อว่า "FCPRINT" มีขั้นตอนการทำงานดังแสดงในรูปที่ 7.14 ซึ่งงานหลักที่ทำ ก็คือ การให้บริการต่อผู้ใช้ในการจัดพิมพ์เอกสารต่างๆ โดยแสดงเป็นเมนูให้ผู้ใช้เลือก และจัดการเรียกโปรแกรมย่อยที่จัดการงานในแต่ละส่วนมารับหน้าที่ไปอีกต่อหนึ่ง



รูปที่ 7.14 : DSD ของโมดูล FCD