## Chapter 7

## Software for controlling the pulse programmer and the pulse shaping unit

In this chapter, programs on the pulse programmer (PPM) and the pulse shaping unit (PSP) will be described. They are written with the 8048 assembly language. The main functions of PPM and PSP consist of the same GPIB controllers and the same system controllers. The GPIB controller communicates the data between IBM PC and itself (PPM or PSP). The system controller controls the main circuit to the write state or to the execute state. The flowchart of the main program of PPM is the same as one of PSP, but their details have some difference.

Since these programs are written in 8048 assembly language, so assignment the initial parameters are necessary. These assignments depend upon hardware of PPM and PSP. These programs perform these assignments. In gpib bus, computer will send commands to PPM or PSP. If command is gpwrt command, the specified PPM or PSP will ready to receive data. Total commands of the gpib card on the IBM PC computer are 16 commands. But PPM or PSP can receive only 8 commands, such as gptrg, gpsclr, gpaclr, gpsloc, gpaloc, gpifc, gpren and gpwrt (see appendix A for these commands). The structure flowchart of PPM or PSP for gpib communication is shown in fig.7-1.

From fig.7-1, when PPM or PSP is turned on, it will start from top of flowchart until receiving commands. It will wait for receiving the commands. When the commands arrive, it will be sent to, the MEAN procedure and then executing these commands, except gpwrt command - it require the more data. So it will wait for receiving the data until receiving the ending code. The ending code is 3 bytes of zero value for PPM or 2 bytes of 255 values for PSP. All data will be stored on system memory. After receiving the ending code, the second LED on the front panel key will be lighted and the system will return to wait for receiving the next command. If the next command is gptrg command, then pulse sequences for PPM or pulse shapes for PSP are released from the output of the system. The detail for PPM and PSP will be given in the later sections.

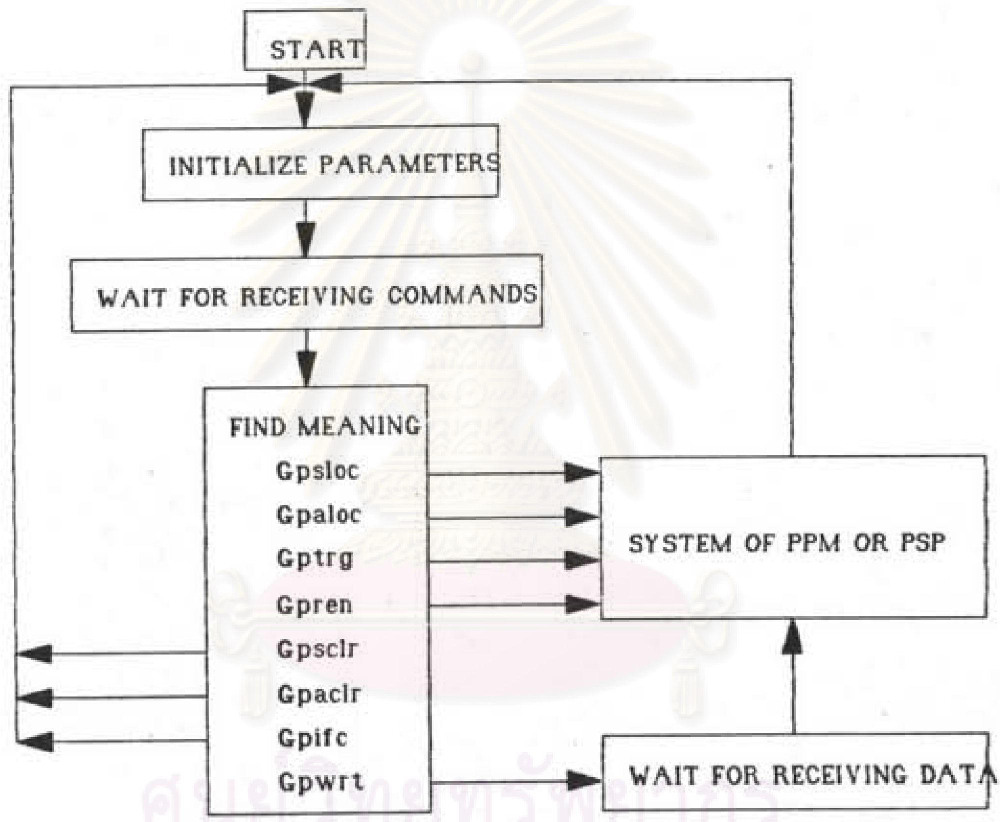


Fig.7-1 The roughly flowchart of PPM and PSP for gpib system.



**Software for controlling pulse programmer**

Before description the main program of PPM, it will agree to the following notation of PPM.

**PPM notation**

R2 : Keep last command

R3 : Keep order of data (OOD) = (0..2)

R4 : Update port2 of 8035

R5 : Update portA of 8255

R6 : Update portB of 8255

R7 : Update portC of 8255

R48($30) : Keep single/rerun (SRR)

R49($31) : Keep state of specific local mode (SLM) = (0..1)

R50($32) : Keep state of listen (SOL) = (0..1)

R51($33) : Keep no. of command

R55($37) : Keep WRT flag = (0..1)

R56($38) : Keep LOC flag = (0..1)

R57($39) : Keep CLR flag = (0..1)

R58($3A) : Keep TRG flag = (0..1)

R59($3B) : Keep UNL flag = (0..1)

R60($3C) : Keep UNT flag = (0..1)

R61($3D) : Keep state of end of data (EOD) = (0..1)

R62($3E) : Keep state of last three bytes zero (LTBZ) = (0..1)

R63($3F) : Keep address of GPIB

F0 : Using in COMPAR procedure

F1 : Keep state of end of identity (EOI)



| Bit no. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | EOD | $\overline{OE2}$ | START | $\overline{W}$/R | $\overline{OE1}$ | $\overline{LE3}$ | $\overline{LE2}$ | $\overline{LE1}$ |
| Active | ‾\|_\|‾ | LOW | ‾\|_\|‾ | LO/HI | LOW | LOW | LOW | LOW |

Table 7-1 Bits of port B (R6) are used to control the main circuit of PPM.

| Bit no. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | $\overline{\text{GPDIR}}$ | $\overline{\text{SRQ}}$ | $\overline{\text{GPEN}}$ | CPEN | $\overline{\text{LED1}}$ | REN | $\overline{\text{RES}}$ | INC |
| Active | LOW | LOW | LOW | HIGH | LOW | HIGH | LOW | ¯\|_\|¯ |

Note : SRQ is set to high.

Table 7-2 Bits of port C (R7) are used to control the main circuit of PPM.

| Bit no. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | NDAC | REN | NRFD | EOI | - | - | - | - |
| Active | LOW | LOW | LOW | LOW | - | - | - | - |

Table 7-3 Bits of port 2 are used to control the GPIB system.

In this section, the detail of PPM will be given. The above-mentioned initial parameters should be known before description the next detail. PPM and PSP have the same main flowcharts as shown Fig.7-2. But GDATA procedure of PPM and some initial parameters are different from ones of PSP.

From Fig.7-2, the initial parameters are assigned in INIT procedure when PPM is turned on. In MAIN procedure, NDAC and NRFD are assigned to be low and high, respectively for acknowledgment computer (controller). In the other word, to tell the computer that PPM ready to receive any commands. The program flow to GREN procedure. It will check the gpren command until any commands (ATN=0, DAV=0) or data (ATN=1, DAV=0) arrive. If any commands arrive, it will flow to GDAVCL, SNRFDC, GCOM, MEAN, SDAC, GDAVH and CEOI procedures and return to MAIN procedure again. This process will get the command to find meaning, execute it immediately and handshake with computer also (see flowchart of MEAN procedure in fig.7-3). If any data arrive, it will flow to GDAVC, SNRFD, SEOI, GDATA, SDAC, GDAVH, CEOI and CEOD procedures and return to MAIN procedure again. This process will get the data to store in system memory (in the write state) and handshake with computer also. It will occur after receiving the gpwrt command and the specified address from the computer is the same as one is assigned on the bottom dip switch of PPM. If SOL=0 then it will not flow to SEOI and GDATA procedures. If the data is the ending code (2 bytes of zero values), the GDATA procedure will assign EOD=1 (see flowchart of the GDATA procedure in Fig.7-4). It will flow to SDAC, GDAVH, CEOI, CEOD and SREAD procedures and return to MAIN procedure, after passing the GDATA procedure. Since this program is very complicated, it is not easy to describe and understand. So reader should see the circuits, the flowcharts and the source program which is shown in appendix D and read appendix A also.

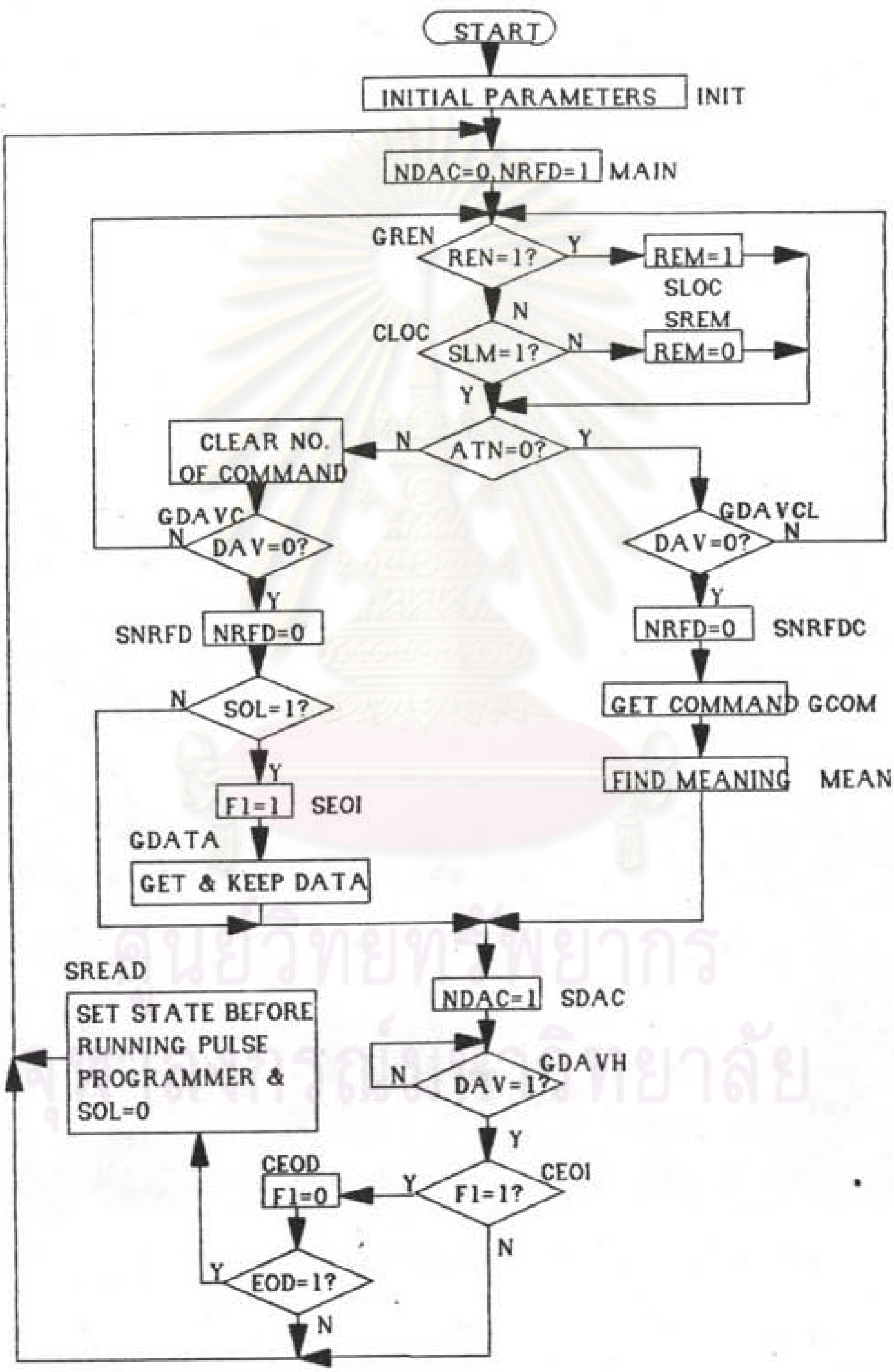


Fig.7-2 The main flowchart of PPM and PSP.

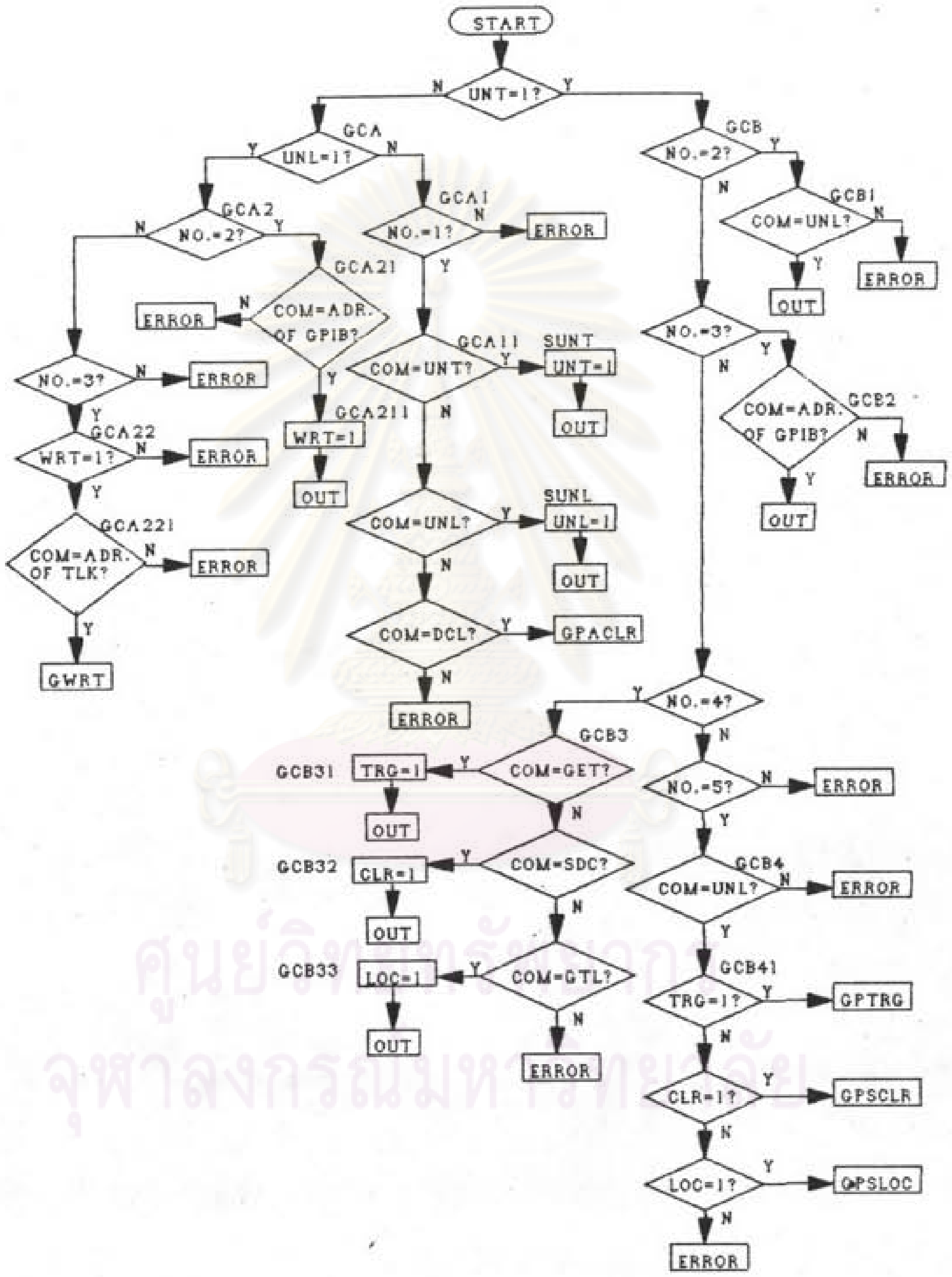


Fig.7-3 The flowchart of procedure MEAN of PPM and PSP.

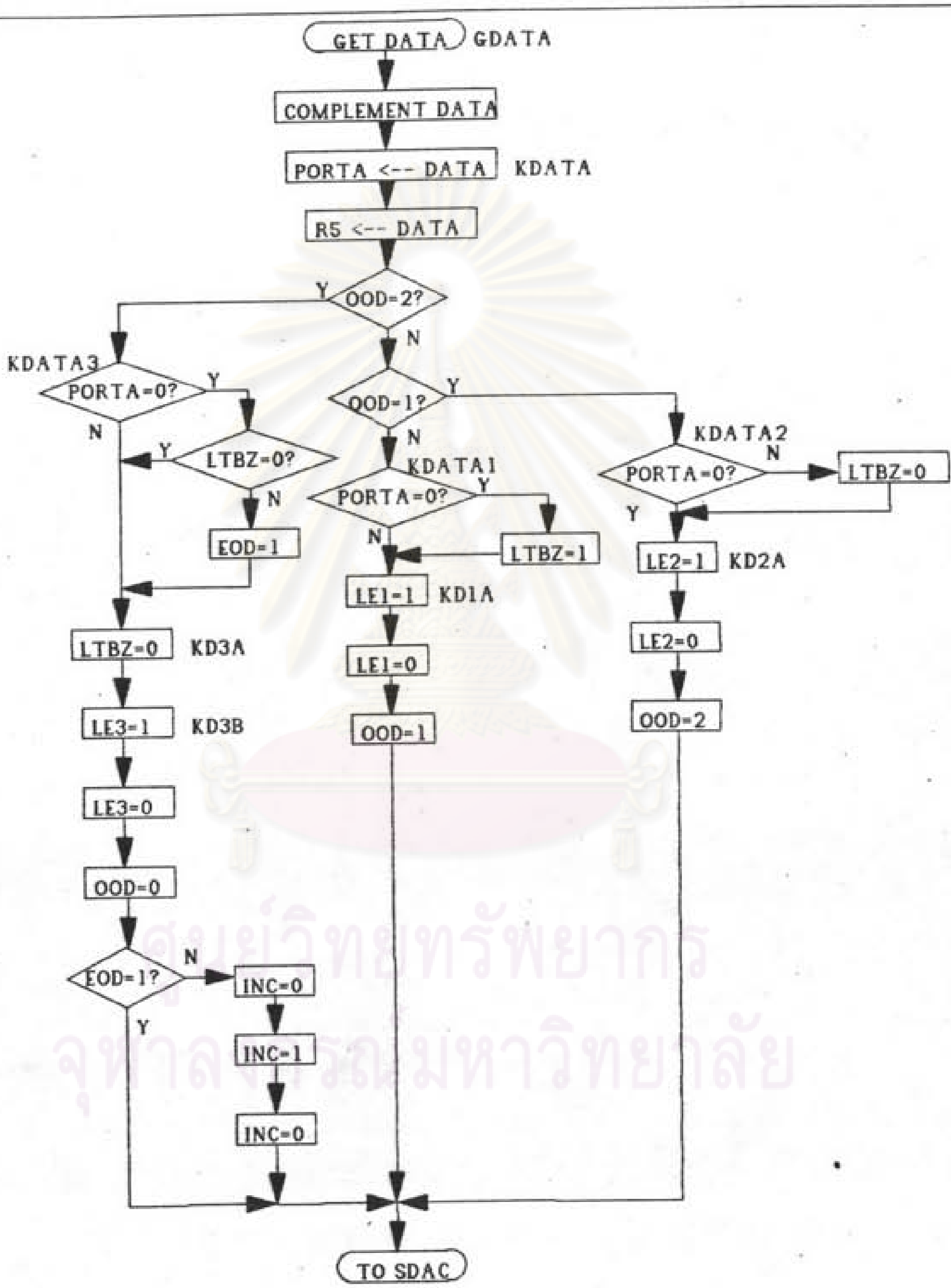


Fig.7-4 The flowchart of procedure GDATA for PPM.

## Software for controlling pulse shaping unit

The notation of PSP is the same as notation of PPM, but the bits patterns of port B (R6) and port C (R7) of PSP are different from ones of PPM which are shown in Table 7-4 and Table 7-5, respectively.

| Bit no. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | EOD | EN | $\overline{OE2}$ | $\overline{W}$/R | OC | $\overline{LE}$ | $\overline{OE1}$ | START |
| Active | ‾\|_\|‾ | HIGH | LOW | LO/HI | HIGH | LOW | LOW | ‾\|_\|‾ |

Table 7-4 Bits of port B (R6) are used to control the main circuit of PSP.

| Bit no. | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| Name | $\overline{GPDIR}$ | $\overline{SRQ}$ | $\overline{GPEN}$ | $\overline{LED3}$ | $\overline{WREN}$ | $\overline{LED1}$ | INC | $\overline{RES}$ |
| Active | LOW | LOW | LOW | LOW | LOW | LOW | _\|‾\|_ | LOW |

Note : SRQ is set to high.

Table 7-5 Bits of port C (R7) are used to control the main circuit of PSP.

The main flowchart of PSP is the same as one of PPM as shown in Fig.7-2. The flowchart of MEAN procedure of PSP is also the same as one of PPM as shown in Fig.7-3. But the flowchart of the GDATA procedure of PSP is different from one of PPM, as shown in Fig.7-5.

This GDATA procedure remains to assign EOD=1 for receiving the ending code, 2 bytes of 255 values for PSP. Since the data of PSP and PPM are different, so the GDATA procedure are also different. The source program of PSP is shown in appendix E.

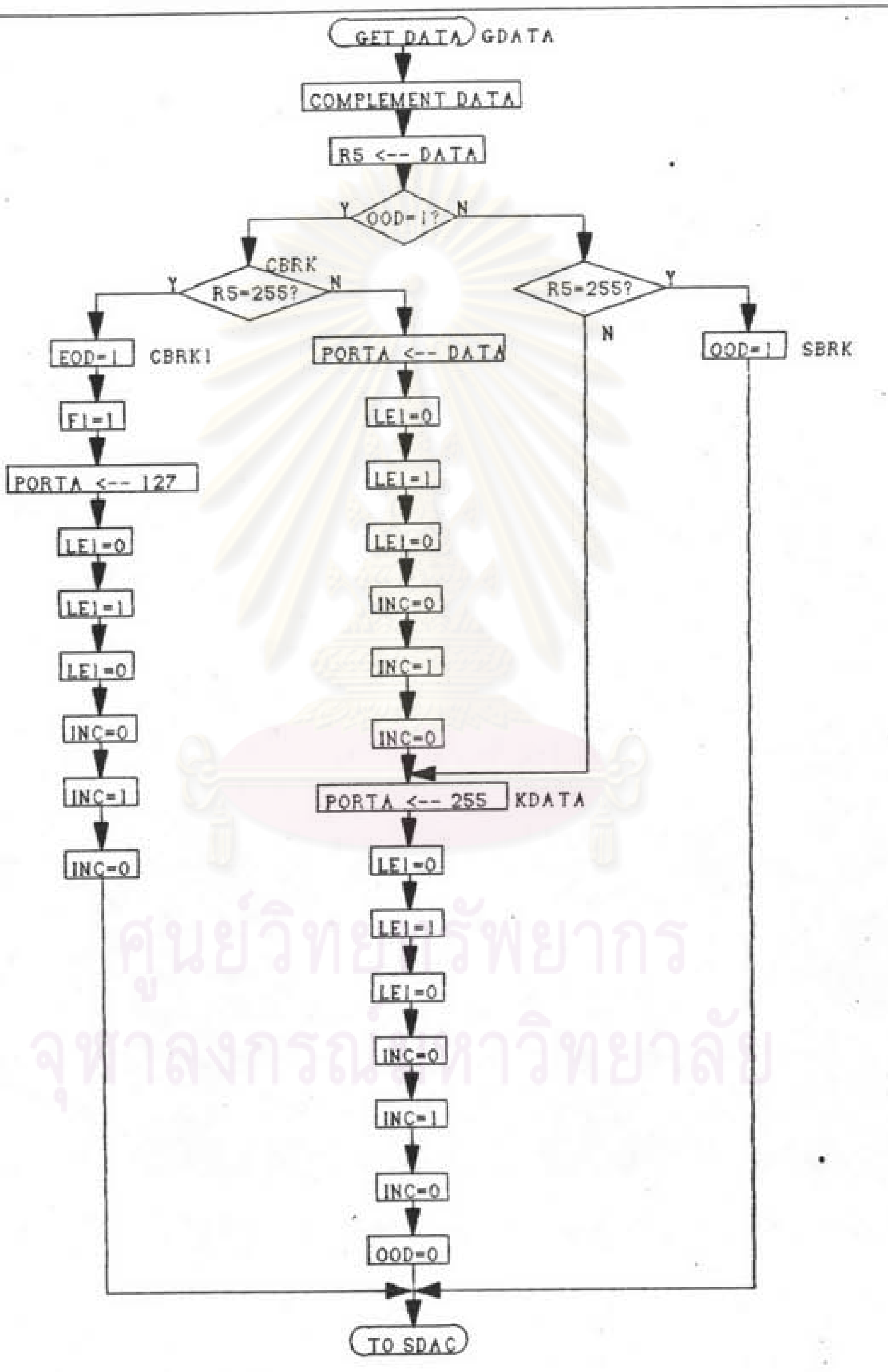


Fig.7-5 The flowchart of procedure GDATA for PSP.