signetics

# MOS MICROPROCESSOR

SUPPORT SOFTWARE
FOR USE WITH THE
NCSS TIMESHARING
SYSTEM ......SP52

# SUPPORT SOFTWARE FOR USE WITH THE NCSS TIMESHARING SYSTEM | SP52

2650 MICROPROCESSOR

APPLICATIONS MEMO

## 1. INTRODUCTION

A series of programs is described that provide the microprocessor application's design engineer with on-line support for the development of programs to be run on the Signetics 2650 microprocessor. These programs include a cross-assembler, a cross-simulator, and two utility programs that convert the object file produced by the assembler into either one of two tape formats – one suitable for loading into the 2650 microprocessor and the other suitable for burning PROMs. The programs are accessed through a comminications terminal connected to a National CSS Data Center via standard telephone lines.

The first few sections describe the available programs and provide detailed instructions for using them. All available usage options are included as reference information. A final section, called "Operating Instructions," provides the user with step-by-step procedures for generating, editing, assembling, punching, and simulating Signetics 2650 programs. These procedures explain some of the more commonly used features of both the NCSS and the Signetics facilities and demonstrate how to use them.

## 2. USAGE OVERVIEW

The user creates the source file for his assembly language program by using the EDIT facility available on the NCSS system, or he may have his program punched onto cards and read into the system at a NCSS Data Center. Once the source file resides on the system, the user executes the assembler, which translates symbolic source statements into machine language instructions, and generates both an assembled listing of the source file and an object file. If the assembler reports any errors in the source file, the EDIT facility may again be invoked to correct the source file. The corrected source file is then resubmitted to the assembler. Once the assembler reports no errors, the user may input the object file to the simulator which then simulates execution of the program. The simulator provides the following capabilities:

1) Establishes initial program conditions.
2) Monitors execution sequences.
3) Modifies the program until it operates as desired.

Once the program operates correctly, the user may repeat the entire cycle: correct his source file; reassemble; and test the new program using the simulator. When the program is fully tested and debugged, it may be punched onto tape.

## 3. EXECUTING 2650 SUPPORT PROGRAMS

### A. GENERAL

To execute any of the 2650 support programs, the following command must be entered:

```
ATTACH P2650
```

This causes the P2650 "PROTECT" Exec to execute. It prints:

```
P2650 Attached as XXX, (Y) RUN? >
P2650 - Version "No." - "Date"
Run on "DATE"
ENTER COMMAND (e.g., HELP) >
```

At this point the user may enter any one of the following commands:

| | |
|---|---|
| HELP | Print Command List |
| HELP 'NAME' | Print Command in Detail |
| QUIT | Exit P2650 (Return to VP/CSS) |
| NEW | Print New Features |
| PIPHASM | Assemble 2650 Program |
| PIPSIM | Simulate 2650 Program |
| PIPHTAP | Punch PIPBUG Tape |
| PIPSTAP | Punch PROM Burning Tape |

No other CSS command may be executed while under control of the P2650 "PROTECT" Exec; e.g., you cannot edit your file until you exit P2650 by typing "QUIT":

```
ENTER COMMAND > QUIT
```

### B. HELP – AN ON-LINE INFORMATION RESOURCE FACILITY

To determine what commands are currently available on P2650, type:

```
HELP
```

To obtain information on how to enter any command except HELP or QUIT, type HELP followed by the name of the desired command; e.g.,

```
HELP PIPHASM
```

A description of the command and its format will be printed.

## 4. PROGRAM DESCRIPTIONS

### A. PIPHASM – SIGNETICS 2650 PIP ASSEMBLER

PIPHASM supports the 2650 assembler languages as specified in the basic manual set (2650 BM 1000). It outputs a hexadecimal object module in a format acceptable to the two tape-punching programs, PIPHTAP and PIPSTAP, and to the simulator, PIPSIM.

Following is the format of the command for executing the assembler:

PIPHASM SOURCE (DISPLAY) (WIDTH)*

where

PIPHASM causes the assembler to execute.

SOURCE is the name of the user's source file. This file has a type of "SYSIN".

DISPLAY is an optional parameter specifying that the listing is to be printed either on the user's console (CON) or on the off-line printer (PTR). If this parameter is missing, CON is assumed.

WIDTH is an optional parameter specifying the line width of the user's console in characters per line—either 80 characters (1) or 120 characters (0). If no parameter is specified, 120 characters per line is assumed. This parameter may be specified only if CON has been specified by DISPLAY.

The object file produced by the assembler will have the same file name as the input file with ".OBJ" concatenated at the end; it will have a filetype of "DATA".

## B. SIGNETICS 2650 SIMULATOR

The 2650 simulator, a program written in FORTRAN IV, simulates the execution of a program without using the 2650 processor. The simulator executes a 2650 program by maintaining its own internal FORTRAN storage registers to describe the program, the microprocessor registers, the ROM/RAM memory configuration, and the input data to be read dynamically from I/O devices. The user may request traces of the processor status, dumps of the memory contents, and program timing statistics. Multiple simulations of the same program with different parameters may be executed during one simulation run.

The simulator requires as input both the program object module produced by the 2650 assembler and a file of user commands. It produces a listing of user commands, executes the program, and prints ("displays") both static and dynamic information as requested by the user commands. The user may direct the input of the simulator either to a terminal or to a line printer.

PIPSIM SOURCE COMMAND (DISPLAY)

where

PIPSIM causes the simulator to execute.

*Parenthesis indicate an optional parameter with a default value.

SOURCE is the name of the source file originally submitted to the assembler. The simulator concatenates .OBJ onto the name of the source file and uses the designator, SOURCE.O, to find the file containing the object module of the program to be executed. File names are limited to eight characters. This object module is ordinarily produced by the assembler and has a filetype of "DATA."

COMMAND is the name of a file containing the user's commands. This file has a filetype of "DATA."

DISPLAY is an optional parameter specifying the destination of all printed output either to the user's console (CON) or to the off-line printer (PRT). If no parameter is specified, the user's console is assumed.

## C. PAPER TAPE UTILITIES

### *1) PIPHTAP*

PIPHTAP punches the "hex" object file onto tape in a format acceptable as input to the 2650 Prototyping Card (2650 PC 1000). See Signetics Applications Memo SS51 for the tape format specification.

The command format for PIPHTAP is:

PIPHTAP SOURCE

where

SOURCE is the name of the source file originally submitted to the assembler.

When "EXECUTION:" is printed, turn the punch on.

### *2) PIPSTAP*

PIPSTAP punches the "hex" object file onto tape in a form suitable for burning PROMs in SMS format. PIPSTAP uses the same command format as PIPHTAP; i.e.,

PIPSTAP SOURCE

where

SOURCE is the name of the source file originally submitted to the assembler.

PIPSTAP responds with a request for the following information:

- The name of your object file.
- The value (two hexadecimal digits) representing the unburned state of your PROM.
- The byte size (four decimal digits) of the PROMs to be burned.
- Up to eight pairs of START/END addresses (four hexadecimal digits). Each address pair identifies an area of code in the object module.

NOTE: All numbers entered *must* contain leading zeros; e.g., when entering the size of a PROM as 256, you must enter 0256.

A START address larger than 7FFF, e.g., 8000, terminates the input mode. Once the input mode is terminated, the punch must be turned on. PIPSTAP punches and prints a record for each PROM specified.

START/END addresses are rounded down/up to the limits of the affected PROM. Thus if:

```
          INITIAL PROM VALUE  = FF,
          PROM SIZE           = 0256,
          START ADDR          = 0040,
     and
          END ADDR            = 0240,
```

PIPSTAP punches three records: 0000 – 00FF, 0100 – 01FF, and 0200 – 02FF. Each of the records is preceded by its initial address (0000, 0100, and 0200). This initial address is punched into the tape so that it is visible. This enables the tape to be separated into individual strips for each PROM. The areas 0000 - 003F and 0241 - 02FF are filled with FFs.

Each record is punched in exactly the order that its START/END address was entered so that multiple records may be punched for the same PROM. When PIPSTAP stops punching, turn the punch off.

## 5. OPERATING INSTRUCTIONS FOR USING THE NCSS TIMESHARING SERVICE

### A. GENERAL

1. The computer requests the user to type information by printing a > character at the start of a line.

2. The user terminates each line typed with a carriage return.

3. The user deletes (tells the computer to ignore) characters that were erroneously typed by typing the @ character. The computer deletes one preceding character for each @ character typed; e.g., the message LANE@@@INE corrects the word LANE to LINE. The [ character deletes all characters previously typed on the line.

4. In all of the following examples, lines typed by the user are underlined to distinguish them from lines printed by the computer.

### B. LOGGING IN TO CSS

1. Set the terminal to "LINE" mode.

2. Select the half-duplex mode using the HALF/FULL duplex switch on your terminal (not required on some terminals).

3. Dial the NCSS-supplied telephone number.

4. When you hear a high-pitched tone (indicating that you have established communication with the computer), place the telephone receiver in the modem coupler.

5. Log on by typing an 'S' or a 'O' followed by a carriage return; i.e.,

| | |
|---|---|
| S carriage return | (when using a 10 cps terminal) |
| O carriage return | (when using a 30 cps terminal) |

In response, the system types

```
CSS ONLINE – XXXX
```

to signal that you have reached an NCSS monitor. XXXX is the name of the NCSS system with which you have established a connection. The system also types the prompt character >, indicating that it is ready to accept additional input from your terminal. In response, you should type:

```
>L WEST XXXXXX
```

where XXXXXX is your user ID number.

The system will respond with

```
PASSWORD
XXXXXXXX
```

providing a blocked-out area in which you enter your password. Type the password on top of the blocked-out area and press the carriage return.

When the system responds with

```
A/C INFO:
```

press the carriage return. (You may optionally enter some accounting information if you desire.).

*Messages from the NCSS system are printed here.*

CSS.211 data

time>

### C. USING THE EDITOR TO CREATE A NEW SOURCE FILE AND/OR TO EDIT AN EXISTING SOURCE FILE

*1) Creating a New Program Source File*

a. On NCSS every file has a file name (FN) and a file type (FT). A file name is the unique name to be assigned to your program. Assign your program a file name of 1-to-4 alphanumeric characters beginning with an alphabetic character. The file type of your source program is "SYSIN." The object file created by the assembler is your unique file name plus the .OBJ appendage. The file type of object files is "DATA."

b. The timesharing computer stores all source and object files on disk. The user may obtain a directory of the files stored in his user area by typing the letter, L, e.g.,

```
time > L
FILENAME    FILETYPE   MODE   ITEMS
PROG        SYSIN      P      40
PROG.OBJ    DATA       P      5
```

c. To create a new program source file, the user calls the editor program with an indication of the file name and file type to be created. The editor recognizes that the file name specified is not in the directory and creates a new file.

```
time > E filename SYSIN
NEW FILE.
INPUT:
```

*Type your program here. If you make mistakes, use the @ key or finish typing your program and make corrections as specified in step d below. Type an additional carriage return after the last line to exit the new file input mode. The system responds with:*

```
EDIT:
```

d. If you wish to edit your program (i.e., correct any typing errors or omissions), proceed to step 2b below. If you do not wish to edit your program at this time, type "FILE" to exit the editor.

*2) Editing a Program Source File*

a. The edit mode can be entered directly when the editor is called by specifying a filename-filetype already on disk; e.g., if PROG SYSIN already exists on disk, enter:

```
time > E PROG SYSIN
EDIT:
```

*Enter edit commands.*

b. The system's editing capabilities are based on the pointer concept; i.e., any line in a file can be located by an imaginary pointer. This pointer can be moved up or down, positioned at the beginning or end of the file, or positioned at a specific line. The position of the pointer determines where the next edit request takes place. The position of the pointer is referred to as the "current line."

c. Following is a list of some of the most frequently used editing commands: *(NOTE: Whenever "n" is indicated in a command, it represents a decimal number. If "n" is left off the command, the number 1 is assumed.)*

| | |
|---|---|
| >T | Moves the pointer to the first line of the file. |
| >DO n | Moves the pointer down n lines and prints the new current line. |
| >UP n | Moves the pointer up n lines and prints the new current line. |
| > L/string/ | Moves the pointer to the next line which contains the character string specified between the slash delimiters. It then prints that line. It does not search the current line for the string. If the character string contains a / , then some other character, such as the $, may be used as the delimiter. |
| >P n | Print n lines starting with the current line. Also move the pointer to the last line printed. If n = 1 or is absent, the current line is printed and the pointer is not moved. |
| >DE n | Delete n lines starting with the current line. |
| >R text | Replace the entire line following the pointer with the text on the R line. The text is separated from the R by only 1 blank. Any additional spaces are considered part of the text. |
| >C /string 1/ string 2/ | Replace character string 1 in the current line with character string 2. If the / character appears in either of the strings, use some other character, such as the $, as the string delimiter. |
| >I<br>INPUT:<br>> . . . . .<br>><br>EDIT: | An I followed by a carriage return puts the editor into input mode. This request is issued to insert lines after the current line. After the "INPUT:" message is printed, the user types one or more lines to be inserted into the program. The last line typed should be followed by two carriage returns to return to EDIT mode. The pointer is moved to point to the last line inserted. |

d. Error Messages

Editor error messages are as follows:

| | |
|---|---|
| ? | Invalid edit request. |

| | |
|---|---|
| EOF: | The end of file is reached by an edit request. The request is terminated, and the pointer is positioned after the last line of the file. |
| TRUNCATED | The following line was truncated as shown. Only 72 character lines are permitted. |

e. Exiting the Editor

To exit the editor and save your new file, type:

```
>FILE
```

To exit the editor without changing your original file, type:

```
>QUIT
```

**D. ASSEMBLING THE PROGRAM SOURCE FILE TO CREATE A HEXADECIMAL FORMAT OBJECT FILE FOR PROGRAM SIMULATION AND FOR PROGRAM DEBUGGING ON THE PROTOTYPING SYSTEM**

```
time > ATTACH P2650
P2650 ATTACHED AS 192, (T)
P2650 - Version 2.0 - 1/5/76
RUN ON 'DATE'
P2650 COMMAND (e.g., HELP) > PIPHASM filename
P2650 ASSEMBLER . . .
RUN . . . (YES OR NO)? > YES
EXECUTION:
```

*(Your assembly listing will be printed here. Be patient—there may be a short delay before printing starts.)*

```
TOTAL ASSEMBLER ERRORS = X
```

**E. LOGGING OFF NCSS**

Exit P2650, log off the NCSS system, and review your program for logical and syntactical errors.

```
ENTER COMMAND > QUIT
time > LOGOUT
XXXX VPU'S,XX CONNECT HRS,XX I/O
LOGGED OFF AT time ON date
```

**F. CHECKING OUT YOUR PROGRAM USING THE SIMULATOR**

1. Log on the system as described in step B.

2. Using the editor program, create a file containing the simulator commands. This file is of type DATA. The file name may be the same as the source file name but with a .TST appendage:

```
time > E filename.TST DATA
NEW FILE
INPUT:
```

*NOTE: The directions for using the editor described in steps C.1 and C.2 apply here also.*

Enter commands here.

```
EDIT:
    >FILE
```

3. Request a simulator run.

```
time > ATTACH P2650
P2650 ATTACHED AS 192, (T)
P2650 - Version 2.0 - 1/5/76
RUN ON 'DATE'
P2650 COMMAND; e.g., 'HELP' > PIPSIM filename
  filename.TST
P2650 SIMULATOR . . .
RUN . . . (YES OR NO) ? > YES
EXECUTION:
```

*The simulator listing is printed here.*

**G. LOGGING OFF**

Exit P2650, log off the NCSS system, and review the simulator listing to determine program correctness.

```
P2650 COMMAND > QUIT
time > LOGOUT
XXXX VPU'S XX CONNECT HRS, XX I/O
LOGGED OFF AT time ON date.
```

**H. PUNCHING A PAPER TAPE FOR DEBUGGING ON THE PROTOTYPE CARD SYSTEM**

Check to ensure that the punch is off. After the "EXECUTION:" message is printed by the computer, turn the punch on. Turn the punch off after it stops punching.

```
P2650 COMMAND > PIPHTAP filename
```

*(NOTE: Do not use the .OBJ extension on the filename. The punch program assumes this is the .OBJ file and automatically adds this extension.)*

```
EXECUTION:
```

*.OBJ file will be listed here.*

```
P2650 COMMAND > QUIT
```

Log off the system as in step G above.

**I. PUNCHING A PAPER TAPE FOR BURNING PROMS**

Check to see that the punch is off, and log into the system using the procedures outlined in step B.

Execute PIPSTAP:

```
P2650 COMMAND (e.g., HELP) > PIPSTAP filename
P2650 PIPSTAP . . .
RUN . . . (YES OR NO)? > YES
```

PIPSTAP responds with a request for the unburned state of your PROM. Since PIPSTAP punches data into each location of the PROM, if your object module does not fill the entire PROM, PIPSTAP requires a value that can be used for the other locations. This value must be entered as two hexadecimal digits:

```
    INITIAL PROM VALUE? 00
```

PIPSTAP then asks for the size (in bytes) of your PROM, which must be entered in four decimal digits. The maximum allowable size is 1024.

```
    PROM SIZE? 0256
```

PIPSTAP requests both a START and an END address for the code to be punched. Use four hexadecimal digits for each address as shown below. Don't forget the leading zeros.

```
    START ADDR? 0000
    END ADDR?   000A
```

PIPSTAP will request up to eight pairs of START/END addresses. Enter a number larger than 7FFF, e.g., 8000, when you have completely described the object module:

```
    START ADDR? 8000
```

When you press

Carriage Return

PIPSTAP punches 50 frames of leader followed by the PROM record specified by your START and END addresses. The START address of your PROM, 0000, is punched into the tape so that it is visible.

When punching is complete, turn the punch off and log off the system.

## 6. REFERENCE DOCUMENTS

For additional information, consult the following manuals:

- Signetics 2650 Microprocessor Manual (2650BM 1000)
- VP/CSS Reference Manual (Form 106-3, available from NCSS)
- VP/CSS Edit Command (Form 108-4, available from NCSS)

## NCSS SALES OFFICES

**ATLANTA**
250 Piedmont Ave., N.E.
Suite 1004
Atlanta, GA 30312
(404) 659-1600

**BETHLEHEM**
Pentamation Enterprises
800 Ostrum Street
Bethlehem, PA 18015
(215) 691-3616

**CAMBRIDGE**
1033 Massachusetts Ave.
Cambridge, Mass. 02138
(617) 868-2950

**CHICAGO**
625 N. Michigan Ave.
Chicago, Ill. 60611
(312) 751-2200

**CLEVELAND**
1100 Superior Ave.
Cleveland, Ohio 44114
(216) 771-5550

**DETROIT**
23777 Greenfield Rd.
Southfield, Mich. 48075
(313) 559-7766

**ELIZABETH**
27 Prince Street
Elizabeth, N.J. 07208
(201) 965-2250

**HARTFORD**
630 Oakwood Avenue
West Hartford, Conn. 06110
(203) 247-9607

**HOUSTON**
4600 Post Oak Place Dr.
Houston, Texas 77027
(713) 621-9231

**LOS ANGELES**
1888 Century Park East
Los Angeles, Ca 90067
(213) 277-7511

**NEWPORT BEACH**
3919 Westerly Place
Suite 109
Newport Beach, CA 92660
(714) 833-8370

**NEW YORK**
485 Madison Avenue
New York, N.Y. 10022
(212) 688-7930

**PHILADELPHIA**
Three Penn Center
Philadelphia, PA 19102
(212) 665-1566

**PHOENIX**
2613 North 3rd Street
Phoenix, Ariz. 85004
(602) 948-5921

**PORTLAND**
510 S.E. Morrison
Portland, Ore 97214
(503) 233-2541

**ROCHESTER**
Scientific Calculations
110 Allens Creek Road
Rochester, N.Y. 14618
(716) 286-9653

**SAN DIEGO**
6363 Alvarado Court
San Diego, CA 92120
(714) 286-9635

**SAN FRANCISCO**
One California St.
San Francisco, CA 94111
(415) 989-3930

**STAMFORD**
2777 Summer Street
Stamford, Conn. 06905
(203) 327-9100

**SUNNYVALE**
430 So. Pastoria Ave.
Sunnyvale, CA 94086
(408) 739-6271

**WASHINGTON, D.C.**
1501 Wilson Blvd., #501
Arlington, VA 22209
(703) 524-8460

**Corporate Headquarters**
**NORWALK**
300 Westport Avenue
Norwalk, Conn. 06851
(203) 853-7200

signetics
a subsidiary of **U.S. Philips Corporation**

Signetics Corporation
811 East Arques Avenue
Sunnyvale, California 94086
Telephone 408/739-7700