SMS

SPRINT^{plus}

PLD and PROM Programmer

OPERATION MANUAL

Version 2.2

10 Nov 1986

Sprint Hardware, Software and Manuals

SMS
Microcomputer systems
Im Morgental 13
8994 Hergatz-Schwarzenberg
West Germany
(49) 07522-4460

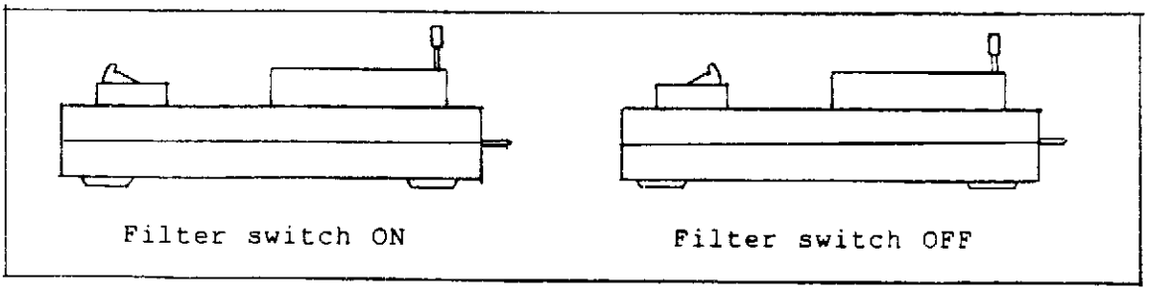
SPRINT Table of contents

```
    Sprint PAL programmer  Version  2.1

a = assemble PLD device
b = blank check device
c = change device type
e = edit pattern in memory
i = read Jedec file from disk to memory
l = list contents of memory
p = program device
q = exit to DOS
r = read device
s = program security fuse
t = test device using Jedec vectors
v = compare device to memory
w = write data to Jedec file
x = convert 24 pin PAL to PLDC20G10

Enter command
```

```
Cypress   22V10

Checksum

67E0

Filter
1 2 3 4 5
on
off
```

```
     Sprint   PROM programmer  Version  2.1

a = set address range
b = blank check device
c = change device type
e = edit memory
f = fill memory
i = read data from disk to memory
l = list memory
p = program device
q = exit to DOS
r = read device into memory
v = verify device
w = write data to disk file

Enter command
```

```
Cypress 7C261/3/4

Checksum

0

Filter
1 2 3 4 5
on
off
```



Sprint overview

Sprint is designed to be a convenient programming tool for EPROMs, EPLDs and EEPROMs using popular IBM and compatible desktop workstations. Sprint is very user-friendly, no special knowledge of programmable devices is required for operation of Sprint. All actions are menu driven, written in clear text. Sprint^{plus} adds several additional software features to the original Sprint product, including a Macro PLD assembler, partial programming of PROMs, and support for many new devices, including 512K bit EPROMs, Bipolar PALs, and additional Cypress CMOS PROMs.

Sprint consists of three parts,

1) An IBM plug-in board containing the pin-driver electronics that provide the signals for programming the devices,
2) A floppy disk containing the software program which controls the timing sequences and voltages,
3) An external socket adapter with a zero-insertion 28 pin socket.

The Sprint board may be left permanently installed in the user's IBM compatible PC/XT/AT computer, it does not interfere with normal operation. The section on installation helps the user to install the Sprint hardware in any IBM PC or compatible computer.

One key advantage of Sprint is the ability to add new devices and features (like Sprint^{plus}) without modifying the hardware. Updates are released by SMS on floppy disk. The purchaser has the right to one free-of-charge update to the Sprint software. A certificate is included which must be presented to SMS together with the original floppy disk. Please write to us at SMS if you would like to have any devices added to the list of supported chips. We will need detailed programming instructions and samples of the devices for implementation.

Starting Sprint

Once your computer is powered-on and running, insert your copy of the Sprint disk and simply type 'SMS'. The Sprint master menu will appear. Enter a '1' for PALs, and EPLDs. Enter a '2' for EPROMs and PROMs. (You may also start the programs without the master menu by typing 'PAL' or 'PROM'.) The entire software is menu driven and prompts the user in clear instructions for action. Input from the user may be in upper or lower case letters.

Next the user is requested to identify the device type to be programmed. Once this menu operation is complete, the main menu will be displayed. The commands available on the main menu are explained in the rest of this document.

To start a command, simply enter the letter shown on the menu, a single keystroke is all that is required. Commands which program devices require a confirming 'y' to avoid accidental programming. Before returning to the main menu, if data has been displayed on the screen as a result of a command, the user will be prompted to enter a 'return' after reading the message on the screen.

Scroll Control — More?

All commands support controlled scrolling of the screen. after 20 lines of text are displayed, the question 'more?' is displayed. Answering '.' or 'n' will terminate the command, any other key will allow an additional 20 lines of text to be displayed.

Sprint PROM functions

A Set address range

In order to partially program devices, the A command can be used to set the operating range. The user will be prompted for the start and end addresses of the range to be programmed. During blank check, program and verify, only the range selected will be addressed. This allows PROMs to be combined together, for example. See the Read command for more detail.

If a range is set, a box will appear on the screen to display the currently active range.

The checksum display on the screen refers to the entire device, not the range selected.

B Blank check

This command verifies that the device is empty. The device in the socket is read and the contents compared to the expected 'erased' condition. The device type selected must match the physical device inserted in the socket. Some devices require special blank check sequences, Sprint automatically uses these sequences if required.

C Change device type

The active device type can be changed using this command. The user is prompted by a menu with the list of currently available devices. After selection, SPRINT returns to the main menu. Illegal selections are ignored. The size of the memory buffer is automatically set to the device size selected. The checksum of memory is calculated and displayed on the main menu screen. For large devices, like the 27512, this can take a few seconds.

Some PROM devices have, in addition to the data array, additional reset and enable functions. Sprint automatically adjusts the total memory size to include these extra functions.

NOTE

Be sure to set the 'Filter' switches on the socket pod to match the diagram shown on the screen. Failure to do so may result in damaged devices.

EXAMPLE: Changing Device Type

Entering 'c' from the main menu of 'PROM' results in the following screen:

```
SPRINT    Device type selection, enter vendor:

      A = AMD
      B = Cypress
      C = Fujitsu
      D = Hitachi
      . . . .       and many more
```

entering a 'B' results in the following second menu:

```
                Device menu for Cypress

       A = 7C225
       B = 7C235
       C = 7C245
       D = 7C261/3/4
       E = 7C281/2
       . . . .      and many more
```

from which the letter for the required device type is entered.

To allow conversions from one device type to another, memory contents are not erased when device types are changed. The device type selected is shown on the main menu screen.

NOTE

Be sure to set the 'Filter' switches on the socket pod to match the diagram shown on the screen. Failure to do so may result in damaged devices.

D DOS command

This allows PCDOS commands to be executed without leaving the Sprint environment. The most common use is for directory listings, or changing paths, but any normal DOS command can be entered, in the same manner as in DOS itself. For example:

```
DIR *.JED
```

will list all the files with the extension or file type ".JED" (short for JEDEC).

E Edit memory contents

Contents of the buffer memory can be examined, displayed and then corrected with this Screen oriented editor. After typing 'e', the user will be prompted to enter the address to be edited. The user should enter the HEX address, followed by a return. The screen will be filled with 256 bytes around the desired location, and the cursor will be positioned to that location. All 8 cursor control keys may be used to point to any location in the memory. With the ' key, the TEXT mode is entered. This allows the entry of ASCII text data directly. ESC returns to the cursor mode.

Data may be entered while in cursor mode by simply typing in the hex data. The ASCII equivalent will be displayed at the end of the line.

The ESC key will cause a new checksum to be calculated, and the main menu to be displayed. For very large devices, this can cause some seconds of delay. The correct checksum is displayed on the main menu screen.

F Fill memory

A range of buffer memory can be initialized to any value with this command. After 'f' is entered, the user will be prompted for the hex start address, end address, and one byte of data. The addresses and data are separated by commas. After return, the memory will be filled with the range of memory with the data provided.

I Input disk data into memory

Loads the data from the file selected into the buffer memory. After 'i' is entered, the system prompts the user for a filename. When prompted, a filename in normal PCDOS filename format should be entered, with an optional drive, path and extension. For example the sample 2764 binary file on the root directory of drive b will be:

```
B:..\T2764.COM
```

if no file extension is provided, the default '.COM' will be used. The default drive and directory is the one is use when PROM started, or the one selected by the 'D' (DOS) command.

Data will be read from the disk until the buffer is full, or end of the file is reached. The number of bytes read from the file are displayed on the screen. For registered proms, the total length includes the data plus reset and enable bytes.

After reading, a new checksum value will the calculated. For very large devices, this can cause some seconds of delay. The correct checksum is displayed on the main menu screen.

Four kinds of file types are supported,

a) Binary format

If this format is selected, the data in the file will be read until either the end of file is reached, or the size of the prom is reached. With the binary format, Sprint allows the user to selectively read either the Even, Odd or All bytes of disk data. A non-zero start address in the data file may be specified, the default is zero. Finally a destination address of the data in the buffer memory can be selected. The default for the destination address is the beginning of the programming range (see Set Range command).

b) HEX space

This is a string of bytes in hex format, separated by a space character. Returns and line feeds are ignored. An example of the format is shown below, it is also on the distribution floppy disk. (file "DEMO.HEX")

```
STX  12 23 34 45 56 67 78 89 90 01 12 23 34  45
56 67 78 89 12 23 34 45 56 67 78 89 90 ETX
```

c) Intel HEX and Motorola S record formats.

These formats are supported according to each manufacturers data. A starting address of the data in the file may be selected. Sprint automatically determines if the HEX file is SPACE, Intel or Motorola.

L List memory contents

Displays in tabular form the contents of the buffer memory. When 'l' is entered, the user is prompted for the starting and ending addresses. Memory is displayed in HEX, 16 bytes per line, from the start to the end, with a pause after each block of 20 lines on the screen. The ASCII equivalent is also shown at the end of each line. A comma separates the starting and ending addresses.

O Display Vpp voltage

This command will display on the screen, the Vpp voltage Sprint will use based on the device selected. It can be used to compare the voltage to be used with the manufacturers data sheet.

P Program device

When 'p' is entered, the device in the socket will be programmed using the data stored in the memory. The device type selected by the user must match the physical device in the socket. The user is prompted to confirm the action with a 'y' key, any other key will abort the command.

The device will first be checked for erasure (Blank check). If the device is not erased, Sprint will test if the pattern in memory can be overprogrammed on top of the existing pattern in the device. If so, the user will be prompted to 'skip blank check'. Entering 'y' will allow Sprint to program on top of the existing pattern.

Then the device will be programmed using the algorithm defined by the manufacturer. Intelligent programming is used to keep the programming time to a minimum. The intelligent programming algorithm used is automatically determined by the vendor/device type chosen.

Extra PROM features like reset and synchronous enable locations are automatically programmed in devices which have them. See the appendix for details on these features.

After programming the device will be verified to confirm proper programming. When all is OK, the message PASS appears on the screen.

If there are any errors, they will be displayed on the screen, and the message 'FAIL' will appear.

Q Quit

Returns the user to the operating system.

R Read device to memory

Reads the device in the socket pod into the memory buffer when 'r' is entered. Note that after data has been read into memory, the device type may be changed without altering any data stored.

When registered PROMs are selected, the reset byte and programmable synchronous enables (if any) are automatically read into the buffer memory for correct duplication or storage. Any PROM device type having the same pinout as the selected device can be read. For example, Bipolar proms can be read into memory by selecting the CMOS device with the same pinout.

After reading, a new checksum value will the calculated. For very large devices, this can cause some seconds of delay. The correct checksum is displayed on the main menu screen.

With the A command, a starting address range larger than the physical device may be entered. This allows the user to read in several smaller EPROMs, at successively higher addresses, in order to combine these smaller EPROMs into one larger EPROM. The working memory is 32768 bytes in size. An EPROM in the socket can be read into any portion of this working memory. When a larger device is selected with the Change command, this working memory is not cleared. To combine 4 2764's into a 27256, read the 4 2764's into addresses 0-1fff, 2000-3fff, 4000-5fff, and 6000-7fff. Change to a 27256, then program.

V Verify device

Compares the device to the data in memory. Any differences are displayed in clear text, with device data and address. All features of the device are verified, including extra reset and enable bits and architecture bits.

W Write buffer data to disk

Writes the data in the buffer onto the disk using the filename supplied. After entering 'w', the user will be prompted to enter a PCDOS style filename. A drive label, path or extension may optionally be entered. The default extension for PROMs is '.COM'. The format used is binary.

Sprint PLD functions

A Assemble

The command calls the Sprint PLD Macro assembler. The result of the Macro assembler will be stored on disk, with the same filename as the source file, but with the extension '.JED'. After assembly is complete, the system will return to the Sprint main menu, the user should confirm the correct device type is selected, and read in the file from the disk.

B Blank check

This command verifies that the device is empty. The device in the socket is read and the contents compared to the expected 'erased' condition. The device type selected must match the physical device inserted in the socket. Reading and checking PLDs involves the use of voltages higher than 5v. If the device type displayed on the main menu is not the same as in the socket, damage may result.

C Change device type

The active device type can be changed using this command. The user is prompted by a menu with the list of currently available devices. After selection, SPRINT returns to the main menu. Illegal selections are ignored. The size of the memory buffer is automatically set to the device size selected. The checksum of memory is calculated and displayed on the main menu screen.

PLDs have different internal sizes. Only the valid fuse locations are enabled when a PLD type is selected. The internal size is compatible with the Jedec data files generated by PLD development software.

NOTE

Be sure to set the 'Filter' switches on the socket pod to match the diagram shown on the screen. Failure to do so may result in damaged devices.

EXAMPLE: Changing Device Type

Entering 'c' from the main menu of 'PAL' results in the following screen:

```
SPRINT    Device type selection, enter vendor:

      A = AMD
      B = Cypress
      C = MMI
      D = National Semiconductor
      E = Texas Instruments
```

entering a 'B' results in the following second menu:

```
                Device menu for Cypress

      A = PALC16L8
      B = PALC16R4
      C = PALC16R6
      D = PALC16R8
      E = PALC22V10
      . . . .      and many more
```

from which the letter for the required device type is entered.

To allow conversions from one device type to another, memory contents are not erased when compatible device types are changed. The device type selected is shown on the main menu screen. If when changing device types the internal size of the fuse arrays change, the memory will be reset to 0 to avoid incorrect conversions.

NOTE

Be sure to set the 'Filter' switches on the socket pod to match the diagram shown on the screen. Failure to do so may result in damaged devices.

E Edit memory contents

Contents of the buffer memory can be examined, displayed and then corrected with this command. The user will be asked for the decimal product term and input term numbers to be examined. Once this has been entered, followed by a return, the location will be displayed. The program then waits for user commands as follows:

CR	display next fuse
Backspace	display previous fuse
.	exit to main program
X	stores a 0, meaning 'connect' at the current point
-	stores a 1, meaning 'disconnect' at the current point
_	set entire product term to no connect (on)
*	set entire product term to all connected (off)

The checksum of memory is calculated and displayed on the main menu screen.

I Input disk data into memory

Loads the data from the file selected into the buffer memory. After 'i' is entered, the system prompts the user for a file name.

When prompted, a file in normal PCDOS filename format should be entered, with an optional drive, directory path and extension. For example a 22V10 binary file on drive b will be:

B:PAL22V10.JED

if no file extension is provided, the default '.JED' will be used.

Sprint expects standard JEDEC format information. If the data file contains test vectors, then these will be stored in a separate buffer for use by the 't' command. Sprint will check if the number of fuses in the JEDEC file match the size of the device selected.

If the JEDEC file contains the command Q1, then the security fuse will automatically be set after programming.

The checksum will be calculated and displayed on the main menu screen.

L List memory contents

After entering 'l', the user is prompted to enter the start and ending product term for the display. The screen will show 'X' for a connection and '-' for an open 'fuse'. Lines containing all 'X' are not displayed, since such product terms have no logical function. Many PLDs have unused portions of the fuse map, these unused portions are disabled in memory and are not displayed. The 'X' corresponds to a '0' in the Jedec file. The '-' corresponds to a '1'.

If a hard copy of the fuse map is required, the user should type the 'CNTL' and the 'Prt Sc' keys simultaneously before typing return, after entering the range of product terms. This will turn on the printer. The 'CNTL' - 'Prt Sc' keystroke should be repeated after the 'ready' prompt in order to turn off the printer.

P Program device

When 'p' is entered, the device in the socket will be programmed using the data stored in the memory. The device type selected by the user must match the physical device in the socket. The user is prompted to confirm the action with a 'y' key, any other key will abort the command.

The device will first be checked for erasure (Blank check). If the device is not erased, Sprint will test if the pattern in memory can be overprogrammed on top of the existing pattern in the device. If so, the user will be prompted to 'skip blank check'. Entering 'y' will over-program the device.

Then the device will be programmed using the algorithm defined by the manufacturer. Intelligent programming is used to keep the programming time to a minimum. The intelligent programming algorithm used is automatically determined by the vendor/device type chosen.

After programming the device will be verified to confirm proper programming.

After verification, if the security fuse command was included in the JEDEC file, the security fuse of the device will be set. If test vectors were loaded, they will then be used to exercise the device.

Q Quit

Returns the user to the operating system.

R Read device to memory

Reads the device in the socket pod into the memory buffer when 'r' is entered. Memory is first cleared, then the data is read-in. Note that after data has been read into memory, the device type may be changed to a compatible device without altering any data stored.

If the device is empty, Sprint will report a warning on the screen. A device may appear blank if its security fuse has been set.

S Set security protection fuse

PLDs support an option to program one last fuse after other fuses have been programmed. This last fuse prohibits reading (and verification) of the device. This means that the pattern set into the fuses of the device are no longer readable by those who should not have access to the pattern. Before the security fuse is programmed, the user is prompted to confirm the action with a 'y', any other key will abort the command. This must be the last step in the programming sequence since the 'P' command expects to be able to verify (read) a device after programming.

If the JEDEC file contains the security fuse command, then this function will automatically be performed after programming.

T Test PLD using test vectors

If a JEDEC file with test vectors has been read into memory, then the device in the socket can be tested with this command for conformance to these test sequences. Entering the 't' command will begin this test. All inputs will be set, followed by any clock pulses. Then the outputs are compared to the expected value. Any errors are reported with the vector number and pin.

Inputs as 'X' are set to '1' for better floating input simulation, high impedance outputs are tested for incorrect '0' conditions, but not for low impedance '1' conditions. Test simulation can also be used to verify a device after the security fuse has been programmed. Errors are displayed in clear text on the screen.

V Verify device

Compares the device to the data in memory. Any differences are displayed in clear text, with device data and address.

W Write buffer data to disk

Writes the data in the buffer onto the disk using the filename supplied. After entering 'w', the user will be prompted to enter a PCDOS style filename. A drive label, directory path or extension may optionally be entered. The default extension for PLDs is '.JED'. The format used is the JEDEC standard.

X Translate 24 pin PAL to 20G10

This special command converts the pattern in memory from a Bipolar 24 pin PAL into a Cypress CMOS PLDC20G10. All conversions are done automatically, with the original 80 x 40 fuse map repositioned into the 90 x 44 fuse map of the PLDC20G10. Sprint will also add to the original fuse map, the special pre-programmed output controls to match the function of the original device, then all the fuse locations will be translated, and unused inputs will be dis-connected. Sprint will use the currently selected device type as input, and will automatically select the Cypress PLDC20G10 when complete.

Original fuse data may be read into memory either from a device or disk file prior to translation.

Devices supported are:

12L10	20L10
14L8	20L8
16L6	20R4
18L4	20R6
20L2	20R8

PLDASM

Overview

The Sprint PLD assembler is designed to convert an input file, created by any editor, into a standard JEDEC file for use by the SPRINT EPLD/EPROM programmer. The JEDEC file can also be used with other types of programmers.

Input consists of pin definitions, followed by equations for the function to be performed by the PLD. Sprint PLDASM supports a flexible Macro feature which reduces the amount of data that has to be typed-in for the PLD equations. This Macro function also makes the input file easier to read and understand, for better documentation of the application.

The input file is entered using an editor on the IBM PC, if the editor inserts control characters for right justification or other purposes, the non-document mode (as in Wordstar) should be selected.

The completed file is assembled into a JEDEC file using PLDASM. Errors in the input file are identified and reported in clear text. The line number/character number where the error was detected is displayed on the screen for use in correcting the error.

The PLDASM automatically performs 2 passes in the assembly process. The first pass sets the macrocell output stage in high-density PLDs, the second pass fills in the equations. Device macrocells are not to be confused with the Macro function of the PLDASM.

An example of the use of PLDASM shows the operation. From the main menu of Sprint, PLD version, typing an 'a' will load the PLDASM and prompt as below:

Enter name of file :

PLDASM assumes a file type of '.PLD', so after the prompt, only the name need be entered:

Enter name of file :EXAMPLE

If the file EXAMPLE.PLD is on the default drive, it will be used in the assembly, and a JEDEC file with the filename EXAMPLE.JED will be created on the same drive. It's very simple. If a different extension or drive is required, then the expanded entry would be as shown here:

Enter name of file :B:/PLDASM/EXAMPLE

The resulting file will appear on drive B:, directory PLDASM as EXAMPLE.JED. The result file always has the file extension '.JED'.

Sprint PLDASM syntax

An input file for the Sprint PLDASM consists of the following parts:

Comments (Documentation)

Device selection

Pin definition

Macro definition

Equations

End

All input can be either upper or lower case. Sprint PLDASM ignores the case of the text.

Comments

Comments are noted by the character '. All input after the ', until the end of the line are ignored. Comments can be placed anywhere in the input file. All text preceding the word 'DEVICE' is ignored.

Label Definition

A label in the PLDASM consists of any string of up to 10 ASCII alphabetic or numeric characters. The first character must not be a 1 or a 0. A '/' may precede the label to indicate active low functions. Some words are reserved, see the list below.

The labels '1' and '0' are reserved words and indicate fixed logical 1 and 0 conditions respectively. VCC and GND are not allowed as labels.

Device definition

The word DEVICE, in column 1 of the input file, is the logical start of the data to be assembled by PLDASM. the device type to be assembled must follow the word DEVICE, there must be a space between DEVICE and the type selected. Currently supported devices are listed in the appendix.

Pin definition

After the device definition, and before the keyword 'START' (see below), the pins and Macros are defined. Typically the pins are defined first, but Macros and pin definitions may be mixed.

The pin definition consists of a label followed by a pin number. There may be an optional equal ('=') sign between the label and the pin number.

The pin number is the decimal physical pin number. One definition per line is allowed.

MACRO definition

A MACRO definition consists of the word MACRO followed by a label (the MACRO name) followed by a string of characters up to 180 characters long (end-of-line characters are ignored). The terminator is the character ';'. A common error is to omit this terminating character. The ';' character is not included into the Macro.

Later, in the equations, the Macro name can be inserted into the text where the string is to be substituted. In the equations, the Macro name is preceded by a '&'. An example is shown below:

```
....
MACRO  test1   /a13*/a12*/a11* a10* a9;
....

START
....
output = &test1* a1;
```

The AMD PALASM example of a Barrel Shifter is included on the floppy diskette to show the use and advantage of Macros.

Macros cannot be inverted, so no '/' may precede the label.

Equations

The keyword 'START' in column 1 signifies the end of the pin and MACRO definitions. The following text until the word 'END' is found in column 1 is assembled into the PLD device type chosen. Illegal combinations such as using an dedicated input pin for output are identified and reported with line and character number.

Each line in the equation section consists of the following parts:

output pin	label
enable (optional)	label.ena
function or '+'	macrocell function + indicates OR'd terms
input equations	set of 'AND' equations
;	end of equation for this pin

OUTPUT PIN

The '/' character before a label signifies inversion. PLDASM enforces the concept that the pin definition label and the label for the output pin in the equations, must have the same inversion status. Use a '/' in the FUNCTION field to indicate that the result of the equation is active low. When converting from a MMI/AMD PALASM file, simply move the '/' from the column one to before the = or :=.

ENABLE

If the word '.ENA' follows the output pin name, then this line will be used for the output enable of the pin. If an enable is used, then the output pin name must appear once with and once without the word '.ENA'. The only function allowed on a enable line is '='. The string of input equations must end in ';'.

If no enable is included, and the device supports an output enable on the pin, then PLDASM will automatically set the output enable to a fixed '1' (enabled) condition.

FUNCTION

The function can be one of the following:

```
  =   combinational active high
 /=   combinational active low
 :=   registered active high
/:=   registered active low
```

Not all devices support all these possible functions. Some are fixed with regards to output inversion and output registers. Consult the device data sheet for allowed functions.

INPUT EQUATIONS

After the function, the input equations for each output stage are listed. The equations consist of groups of AND and OR terms. Each AND term consists of LABELs, and MACROs separated by '*' and terminated by either a ';' or '+'. The ';' terminator denotes the end of the equations for this pin. The '+' terminator separates AND terms, it indicates that these AND terms are OR'd together.

The LABEL is a standard PLDASM label, with an optional '/' in front to indicate inversion. MACROs can be used to replace any combination of LABEL, '/', '*', '+' or ';'. MACROs are indicated by a '&' in front of the MACRO name defined above. Macros cannot be inverted, so no '/' may precede the MACRO name.

End of file

The keyword END indicates the end of the file. If the program terminates without the message 'Assembly Complete', then the keyword END has likely been omitted. Successful completion will result in a JEDEC file being written to the disk.

Reserved Words

Some words are reserved for use by the PLDASM, these must not be used as labels. The words are:

```
DEVICE   LABEL   ARESET   SPRESET   ENA
END      MACRO   START    SECURITY
```

SPRINT PLDASM EXAMPLE

The attached example shows a file using the PLDASM and MACROs. The file is from the AMD data book on PALs.

First the header text, all data up to the reserved word "DEVICE" in column one. This explains who wrote the file, and some background information as documentation for the reader only.

Then the DEVICE is defined, here it is a PAL16R4. The JEDEC file created for a device is the same for all manufacturers of the same device, so no vendor need be specified.

Then the pins, and MACROs are defined. Any sequence may be used, but typically the numeric order is the easiest to read.

The keyword "START" indicates the beginning of the equations. The output pin label starts in column 1. It has the same polarity (a / in front) as the definition above.

The output pin /ZERO has been added as an example of the use of the output enable. When the inputs S0, S1, and S2 are all 0, the output will be enabled. It will be high if any of the data bits (D0 to D7) are 1, otherwise it will be low. If the inputs S0, S1, S2 are not all 0, then the output will be high impedance.

The keyword "END" indicates the end of the file. After this the PLDASM will write the translated JEDEC file to the disk. The data can then be read by SPRINT by an input command with the same filename as the original '.PLD' file.

The resulting JEDEC file is shown below.

```
                                                PAL DESIGN SPECIFICATION
PAT001                                          KEVIN M. OW-WING 6-22-85
4-BIT SLICE FOR AN 8 BIT BARREL SHIFTER
ADVANCED MICRO DEVICES
Modified by SMS to show the output enable function.
device 16r4;

CK      1
D7      2
D6      3
D5      4
D4      5
D3      6
D2      7
D1      8
D0      9
GND     10
/E      11
/ZERO   12
S0      13
Q0      14
Q1      15
Q2      16
Q3      17
S1      18
S2      19
VCC     20

macro   f0   /S2*/S1*/S0;
macro   f1   /S2*/S1* S0;
macro   f2   /S2* S1*/S0;
macro   f3   /S2* S1* S0;
macro   f4    S2*/S1*/S0;
macro   f5    S2*/S1* S0;
macro   f6    S2* S1*/S0;
macro   f7    S2* S1* S0;

start

Q3 /:= &f0*/D3 +  &f1*/D2 + &f2*/D1 + &f3*/D0 +
       &f4*/D7 +  &f5*/D6 + &f6*/D5 + &f7*/D4;

Q2 /:= &f0*/D2 +  &f1*/D1 + &f2*/D0 + &f3*/D7 +
       &f4*/D6 +  &f5*/D5 + &f6*/D4 + &f7*/D3;

Q1 /:= &f0*/D1 +  &f1*/D0 + &f2*/D7 + &f3*/D6 +
       &f4*/D5 +  &f5*/D4 + &f6*/D3 + &f7*/D2;

Q0 /:= &f0*/D0 +  &f1*/D7 + &f2*/D6 + &f3*/D5 +
       &f4*/D4 +  &f5*/D3 + &f6*/D2 + &f7*/D1;

/ZERO.ENA   = /S0*/S1*/S2;
/ZERO       = /D0*/D1*/D2*/D3*/D4*/D5*/D6*/D7;
end
```

EXAMPLE JEDEC FILE
BARREL SHIFTER

```
*
F0*
QF2048*
L512 11101110111111111101111111101111*
L544 11101110111111111111101111011111*
L576 11101101111111111111111110101111*
L608 11101101111111111111111111011011*
L640 10011110111111111111111111101111*
L672 11011010111111111111111111011111*
L704 11011101101111111111111111101111*
L736 11011101111110111111111111011111*
L768 11101110111111111111101111101111*
L800 11101110111111111111111110011111*
L832 11101101111111111111111111101011*
L864 10101101111111111111111111011111*
L896 11011010111111111111111111101111*
L928 11011110101111111111111111011111*
L960 11011101111110111111111111101111*
L992 11011101111111111101111111011111*
L1024 11101110111111111111111111101111*
L1056 11101110111111111111111111011011*
L1088 10101101111111111111111111101111*
L1120 11101001111111111111111111011111*
L1152 11011110101111111111111111101111*
L1184 11011110111110111111111111011111*
L1216 11011101111111111101111111101111*
L1248 11011101111111111111101111011111*
L1280 11101110111111111111111111101011*
L1312 10101110111111111111111111011111*
L1344 11101001111111111111111111101111*
L1376 11101101101111111111111111011111*
L1408 11011110111110111111111111101111*
L1440 11011110111111111101111111011111*
L1472 11011101111111111111101111101111*
L1504 11011101111111111111111110011111*
L1792 11101110111111111111111111101111*
L1824 10111011101110111011101110111011*
C76c0*
0000
```

SPRINT PLDASM Program Operation

On starting PLDASM, the filenames with the type '.PLD' on the current default directory will be displayed. Enter the filename (the file type is not required) then return. If the message "File Not Found' is displayed, please check the filename and path and try again.

If the filename is ok, then the processing of PASS1, PASS2 and the writing of the JEDEC file to the disk will be reported on the screen.

If any errors are detected, they will be reported with the line number and character number of the position in the file where the error was detected. In some cases this may be a few bytes after the actual error location. Note that the character position counts TAbs as one character. The error types are listed below:

Error Message	Explanation
Device not supported	See list of currently supported devices.
Macro definition error	Macro too long, or too many Macros.
Reserved word	Some words are reserved for use by the PLDASM, see the reserved word list. In any case a reserved word was found in an unexpected location.
Too many/few pins	Attempt was made to define pins not on the physical device.
Missing terminator -> ;	A ';' is required at end of the line.
Device too small	The user attempted to use more OR terms than in the physical device
Label not defined	Check spelling, labels need to be listed in the definition portion of the file.
Improper use of pin	Attempt was made to use a pin in a manner not supported by the physical device. For example an active high output on a PAL16L8.
Use /= or /:= for output inversion	See section on OUTPUT PIN
Output enable error	Attempt was made to use an output enable in a device which does not have it.

Device types

In this version of PLDASM, the following part types are supported.

Device	Vendor	Name	Special Features (Keywords)
PAL16L8	any	16L8	
PAL16R4	any	16R4	
PAL16R6	any	16R6	
PAL16R8	any	16R8	
PAL22V10	AMD	22V10	ARESET, SPRESET
PALC22V10	Cypress	22V10	ARESET, SPRESET
PLDC20G10	Cypress	20G10	
8C322	Cypress	8C322 or 22V10	ARESET, SPRESET
PAL10H8	any	10H8	
PAL10L8	any	10L8	
PAL12H6	any	12H6	
PAL12L6	any	12L6	
PAL12L10	any	12L10	
PAL14H4	any	14H4	
PAL14L4	any	14L4	
PAL14L8	any	14L8	
PAL16C1	any	16C1	
PAL16H2	any	16H2	
PAL16L2	any	16L2	
PAL16L6	any	16L6	
PAL18L4	any	18L4	
PAL20L2	any	20L2	
PAL20L8	any	20L8	
PAL20L10	any	20L10	
PAL20R4	any	20R4	
PAL20R6	any	20R6	
PAL20R8	any	20R8	

Entering just 'return', without a filename will display a list of devices currently supported by PLDASM.

NOTE:

When using the 22V10 or 8C322, the ARESET and SPRESET terms are used as a predefined output pin label. Simply add an equation as shown in the example below, to use these terms:

```
ARESET  =  label1 * label2;
```

Installation

Hardware

Sprint can be installed in any long slot of the IBM PC/XT/AT or compatible. Sprint operation is independent of CPU clock speed. Simply open the computer, and plug the card into a spare slot. Sprint uses the port addresses 300-31F Hex. If there is a conflict with any other card in your system, the Sprint address can be changed using the program NEWADD. See the NEWADD discription.

If your system is a 'Turbo' XT, with an 8 Mhz modified clock rate, the I/O cycles may not meet the IBM specification. Switch the computer to 4.77 MHz mode when using SPRINT.

After the chassis is closed, the socket pod can be plugged into the 40 pin connector on the back panel. The flat ribbon cable connector to the socket adapter is keyed and can only be inserted one way, it is long enough to allow the adapter to be placed on top of the system next to the display.

After installation, boot the system with your usual PCDOS boot diskette or hard disk. Insert the Sprint disk and run the the program 'TEST'. For installation checkout, the command '1' requires no extra equipment for operation, and it does a test of the board. The message 'no errors detected' should appear on the screen. If this test is ok, run the test 7, and check the results for no errors. Then exit the test program by typing Q. If there are any errors, check first the positions of the filter switches on the socket adapter, then check if any other boards in the system conflict with the address 300-31f Hex used by Sprint as shipped from the factory.

Changing Port Addresses

To change the port address, jumpers 1-4 can be cut or strapped as shown in the table below. The jumpers are located between the top and middle rows of chips, near the metal plate at the end of the card, The bits that can be controlled are:

address	jumpers inserted
220	4,1,2
240	4,1 ,3
260	4,1
280	4 ,2,3
2A0	4 ,2
2C0	4 ,3
300	1,2,3
340	1 ,3

Now run NEWADD to change to SPRINT software to match the new address.

Software

The Sprint distribution diskette contains the following files:

```
 Volume in drive A is SPRINT
 Directory of  A:\

SMS      EXE     12265   6-05-86   1:41p
PAL      EXE     30334  12-03-85   9:22a
PROM     EXE     19418  11-17-85   3:47p
TEST     EXE     10670  11-17-85   8:05p
PLDASM   EXE     19341   6-04-86   3:35p
NEWADD   EXE     13233   9-20-86   2:33p
SHIFTER  PAL      1280  11-02-85   5:27p
SHIFTER  PLD      1280  11-02-85   5:27p
SHIFTER  JED      2328  11-02-85   5:38p
T27128   COM     16384   6-04-86   2:41p
```

For best results copy all programs to either your hard disk or another floppy disk. Keep the Sprint master diskette in a safe place. (The Sprint diskette is not copy protected).

The file SHIFTER.PAL is an example of a PAL16R4 application in MMI PALASM format. The file SHIFTER.PLD is the same file in Sprint Macro PLDASM format. The file SHIFTER.JED is the resulting JEDEC file.

The file T2764.COM is an example of the binary format used by Sprint PROM.

Dates and lengths of the files will change as new versions are distributed.

NEWADD

NEWADD is a program which automatically changes the port addresses in SPRINT software to match a new address for the hardware. The hardware address should only be changed if there is a conflict with other plug-in cards. The usual starting address is 300 HEX. Other addresses usable are 220, 240, 260, 280, 2A0, 2C0, and 340.

The programs TEST, PAL, PROM, and SMS need to be modified for a new address. As always you should keep the original SMS disk in a safe place, run 'NEWADD' on your copy of the SMS disk. NEWADD will ask for the new HEX address of the card, and will check to see if the card is installed at that address. The program will then automatically change the port addresses in these 4 programs, and store the programs back onto the disk.

After conversion, just run the program TEST, and perform tests 1 and 7 as a quick check of the changes.

Calibration

The distribution diskette contains a test program for calibration of Sprint. The menu driven package instructs the user in all actions to test and adjust the card. Note that calibration should be performed each 6 months.

A good voltmeter is required for calibration (commands 2, 3). An oscilloscope is needed for timer testing and output driver testing (4, 5, 6).

The commands of the test program are:

1 = readback all inputs	self test
2 = set Vcc voltage	allows adjustment of operating Voltage
3 = set Vpp voltage	allows adjustment of Program Voltage
4 = pulse all outputs	test Vpp drivers
5 = test Vcc drivers	test Vcc drivers
6 = test delay	verify on-board timer function
7 = test relay	verify socket pod ground relay function
q = exit to DOS	

To use the test program, boot the system with the distribution diskette and start the program 'TEST'.

Technical description

Size IBM PC standard full length card. Uses port addresses 300-31F hex. Base address switchable to any address between 200 and 3EC HEX.

Power

5V	1.0 amp	
+12V	1.0 amp (peak)	.4 amp average
-12V	.05 amp	

Socket pod This is the external socket for connection to the device to be programmed or read. It provides a 28 pin .3 and .6 inch wide socket for compatibility with a wide range of devices. Five 'Filter' switches are located on the pod for bypass capacitors according to manufacturers published programming specifications:

1 = Vcc filter capacitor for 24 pin parts
2 = Vcc filter capacitor for 20 pin parts
3 = Vpp filter capacitor for socket pin 1
4 = Vpp filter capacitor for socket pin 22
5 = current limiter for socket pin 20

Ground A relay can select device ground as socket pin 14, or pin 8 and 21. The relay is driven by the bit 0 line. This relay is included on socket pods which have no display LEDs.

Timer On board real time clock. Pulse widths are programmable from 100 us to 25.5 ms in 100 us steps.

Vpp Programmable from 5 to 25.6 volts in 100 mv steps. DC-DC converter on-board operates from +12V supply. Can be switched to any device pin.

Vcc Programmable from 0 to 9.0 volts in 50 mv steps. Drawn from +12V supply. Switched to pins 24, 26 and 28 of the socket for 20, 24 and 28 pin devices.

Pin drivers 28 pins can each be programmed to be:

Output:

Logic 0.	Output from a 7407 driver.
Logic 1.	5 Volts via a 10K resister.
Vpp	50 ma per pin, 350 ma total.

Input:
74HCT input current limited with a 10k resister.

Memory 256k bytes of total memory is sufficient to operate Sprint.

List of PROMs

This list is updated regularly. Additional devices are added as they become available and the programming algorithm has been proven. This list represents the devices supported be version 2.1 of the PROM software.

Device	Vendor	Technology	Size	Filter*	Select	Notes
27S25	AMD	BIPOLAR	512x8 R	1,5	B,A	3
27S35	AMD	BIPOLAR	1024x8 R	1,5	B,B	1,3
27S45	AMD	BIPOLAR	2048x8 R	1,5	B,C	1,2,3
27S181	AMD	BIPOLAR	1024x8	1,5	B,D	3
27S281	AMD	BIPOLAR	1024x8	1,5	B,D	3
27S191	AMD	BIPOLAR	2048x8	1,5	B,E	3
27S291	AMD	BIPOLAR	2048x8	1,5	B,E	3
7C225	Cypress	CMOS	512x8 R	1	B,A	-
7C235	Cypress	CMOS	1024x8 R	1	B,B	1
7C245	Cypress	CMOS	2048x8 R	1	B,C	1,2
7C261/3/4	Cypress	CMOS	8192x8	5	B,F	-
7C269	Cypress	CMOS	8192x8 R	5	B,G	1,4
7C281/82	Cypress	CMOS	1024x8	1	B,D	-
7C291/92	Cypress	CMOS	2048x8	1	B,E	-
6381	MMI	BIPOLAR	1024x8	1,5	B,D	3
63S1681	MMI	BIPOLAR	2048x8	1,5	B,E	3
63RA481	MMI	BIPOLAR	512x8 R	1,5	B,A	3
63RS881	MMI	BIPOLAR	1024x8 R	1,5	B,B	3
63RA1681	MMI	BIPOLAR	2048x8 R	1,5	B,C	3
63RS1681	MMI	BIPOLAR	2048x8 R	1,5	B,C	3
2716	All	NMOS	2048x8	2,5	*,*	
2732	All	NMOS	4096x8	2,4,5	*,*	
2732A	All	NMOS	4096x8	2,4,5	*,*	
27C32A	All	CMOS	4096x8	2,4,5	*,*	
2764	All	NMOS	8192x8	3,5	*,*	
2764A	All	NMOS	8192x8	3,5	*,*	
27C64	All	CMOS	8192x8	3,5	*,*	
27128	All	NMOS	16384x8	3,5	*,*	
27128A	All	NMOS	16384x8	3,5	*,*	
27C128	All	CMOS	16284x8	3,5	*,*	
27256	All	NMOS	32768x8	3,5	*,*	
27256	All	NMOS	32768x8	3,5	*,*	
27256	All	NMOS	32768x8	3,5	*,*	
27C256	All	CMOS	32768x8	3,5	*,*	
27512	All	NMOS	65536x8	5	*,*	

, Sprint provides on screen a list of 13 vendors of NMOS and CMOS standard EPROMs. By choosing the correct vendor and part type, the correct programming voltage and sequences will be used. Intelligent programming is used on all devices except the 2716 and 2732. The correct programming voltage is selected for each vendor/part type.

* For reference only, correct filter switch positions are always displayed on screen.

PROM NOTES:

1. Registered PROMs which have a programmable initialization byte have this byte stored after the last data byte. The location in memory of this byte is:

Device size	reset byte location (HEX)
1024	400
2048	800
8192	2000

Sprint automatically programs this data into the PROM during the programming cycle. The byte can be read in from a device or from a disk data file. When reading a device, Sprint automatically determines the reset byte value.

2. Proms which have additional programmable features have the data stored in a control byte after the reset byte value.

Device size	Control byte location (HEX)
1024	401
2048	801
8192	2001

Sprint automatically programs this data into the PROM during the programming cycle. The byte can be read in from a device or from a disk data file. When reading a device, Sprint automatically determines the control byte value.

3. This device can only be read by Sprint. It cannot be programmed. Contents can be written into Cypress devices.

4. The Cypress 7C269 has programmable enable and programmable sync bits. The memory location 2001 Hex is the control byte location containing the instructions for Sprint for programming these extra features. The data stored in this location has the following meaning, a value of:

0 = async enable
1 = sync enable
2 = async initialize
3 = sync initialize

List of PLDs

This list is updated regularly. Additional devices are added as they become available and the programming algorithm has been proven. This list represents the devices supported be version 2.1 of the PAL software.

Device	Vendor	Technology	Size	Filter*	Select*	Notes
PAL16R8	AMD	BIPOLAR	64x32	2,5	A,D	1
PAL16R6	AMD	BIPOLAR	64x32	2,5	A,C	1
PAL16R4	AMD	BIPOLAR	64x32	2,5	A,B	1
PAL16L8	AMD	BIPOLAR	64x32	2,5	A,A	1
PAL22V10	AMD	BIPOLAR	132x44	1,5	A,E	1
PALC16L8	Cypress	CMOS	64x32	2,5	B,A	-
PALC16R8	Cypress	CMOS	64x32	2,5	B,D	-
PALC16R6	Cypress	CMOS	64x32	2,5	B,C	-
PALC16R4	Cypress	CMOS	64x32	2,5	B,B	-
PALC22V10	Cypress	CMOS	132x44	1,5	B,E	-
PLDC20G10	Cypress	CMOS	90x44	1,5	B,F	-
PAL10H8	MMI/NS	BIPOLAR	64x32	5	#,A	-
PAL10L8	MMI/NS	BIPOLAR	64x32	5	#,B	-
PAL12H6	MMI/NS	BIPOLAR	64x32	5	#,C	-
PAL12L6	MMI/NS	BIPOLAR	64x32	5	#,D	-
PAL12L10	MMI/NS	BIPOLAR	80x40	5	#,E	-
PAL14H4	MMI/NS	BIPOLAR	64x32	5	#,F	-
PAL14L4	MMI/NS	BIPOLAR	64x32	5	#,G	-
PAL14L8	MMI/NS	BIPOLAR	80x40	5	#,H	-
PAL16C1	MMI/NS	BIPOLAR	64x32	5	#,I	-
PAL16H2	MMI/NS	BIPOLAR	64x32	5	#,J	-
PAL16L2	MMI/NS	BIPOLAR	64x32	5	#,K	-
PAL16L6	MMI/NS	BIPOLAR	64x32	5	#,L	-
PAL16L8	MMI/NS/TI	BIPOLAR	64x32	5	#,M	-
PAL16R4	MMI/NS/TI	BIPOLAR	64x32	5	#,N	-
PAL16R6	MMI/NS/TI	BIPOLAR	64x32	5	#,O	-
PAL16R8	MMI/NS/TI	BIPOLAR	64x32	5	#,P	-
PAL18L4	MMI/NS	BIPOLAR	80x40	5	#,Q	-
PAL20L2	MMI/NS	BIPOLAR	80x40	5	#,R	-
PAL20L8	MMI/NS/TI	BIPOLAR	80x40	5	#,S	-
PAL20L10	MMI/NS	BIPOLAR	80x40	5	#,T	-
PAL20R4	MMI/NS/TI	BIPOLAR	80x40	5	#,U	-
PAL20R6	MMI/NS/TI	BIPOLAR	80x40	5	#,V	-
PAL20R8	MMI/NS/TI	BIPOLAR	80x40	5	#,W	-

PLD NOTES:

* For reference only, Vendor and device names together with the select codes are displayed on the screen during the change device sequence. Filter switch positions which must be 'on' are always displayed on screen.

\# This code depends on the vendor being used. The Select device menu displays this information. For reference, C is MMI, D is National Semi., E is Texas Instruments.

1. This device can only be blank checked, read, verified or tested by Sprint. It cannot be programmed. This device is included to allow automatic conversion to CMOS PALs and PLDs.

Changes to Version 2.2

SMS has added to V2.2 the following features:

1) The PROM editor is now a full screen-oriented editor. Cursor control keys are used to point to data to be changed, and any changed data is displayed immediately on the screen. Also displayed on the screen are ASCII equivalents of the data. The editor offers HEX data and TEXT modes of operation.

2) The PROM List memory command is now screen oriented.

3) Intel HEX and Motorola S record formats are now included in the Input command.

4) Additional PROM devices (TI and VTI) are included.

SMS has corrected the following key items in V2.2:

1) Bipolar PAL programming is now supported on machines with clock speeds higher than 4.77 MHz.

2) Testing of PAL devices using JEDEC files containing X (don't care) conditions, will have the pins set to X either at hi-Z or 0 depending whether or not the pin is an output. This occurs fully automatically.

3) PLDASM now includes the PAL20L10. The device command may now be on line one.

4) Cypress programming yields for the 7c292 will be increased by the use of new timing parameters.

Please Note:

Most Asian TURBO IBM-PCs (8 MHz) do not meet the 650 ns specification for I/O cycles as defined by IBM. These computers will not work with SPRINT unless switched to 4.77 MHz mode. A minimum of 410 ns is required for proper operation.