intel®

APPLICATION NOTE

AP-156

October 1986

Designing Modules for iPDS™ and iUP Systems

DALE OLLILA
DSHO TECHNICAL PUBLICATIONS

Order Number: 230682-001

INTRODUCTION

The Intel Personal Development System (iPDS™) is a new development tool concept. It provides a subset of the capability of an Intellec® Series II/III development system, in a portable, and less expensive package. One of the features offered by the iPDS system is the expansion capability designed into the product. The basic iPDS system can be expanded to include a parallel processor, a wide range of serial (RS232C interface) and parallel (Centronics interface) devices, numerous MULTIMODULE™ (iSBX™ interface) devices, additional flexible disk drives, and a growing line of plug-in emulator and PROM programming modules.

The plug-in modules for the iPDS system communicate over an interface referred to as the Intel Personal Development Expansion bus (iPDS bus). The iPDX bus is also used in another Intel product, the iUP-200/201 Universal Programmer (iUP). There are some differences in iPDX bus implementation between the iUP and iPDS systems, but the basic interface is the same. Intel PROM programming modules can be used in either system.

THE iPDX BUS

The iPDX bus is a byte-wide, parallel interface between a plug-in module and the iPDS or the iUP system. The iPDX bus allows a variety of plug-in modules to be added to the iPDS system. (The iUP system normally is used with PROM programming modules.) Some of the possible types of plug-in modules are:

- PROM programming modules
- Emulator (EMV) modules for various microprocessor or microcontroller families
- Test instrumentation modules (e.g., logic or signature analyzers)
- Analog interface modules (e.g., analog/digital or digital/analog converters)
- Serial communication modules (e.g., modem or cassette controller modules)
- Parallel communication modules (e.g., direct interface to other CPU buses)
- Program storage modules (e.g., modules storing alternate operating systems, diagnostic programs, or games)

Intel Corporation produces plug-in modules that allow PROM programming and emulation for a variety of Intel chips. The special needs of individual users may not be satisfied by the plug-in modules that are available. This application note presents the specifications and design criteria for user-designed plug-in modules using the iPDX bus. User-designed plug-in modules can expand the usefulness of the iPDS system in the design lab, on the production floor, and in field applications.

iPDX Bus Features

The iPDX bus's capabilities are nearly equal to the capabilities of the iSBX™ bus. In some respects the iPDX bus is more powerful than the iSBX bus, due to the variable and switched supply voltages included on the bus. The features of the iPDX bus are:

- The controlling (iPDS or iUP) system supplies +5VDC and ground to the iPDX bus.
- The controlling (iPDS or iUP) system supplies switched voltages of +5.7VDC, −12VDC, and +8VDC to +27VDC to the plug-in modules. In addition, the iUP system controls a variable switched voltage (+8VDC to +15VDC) and the iPDS system controls a +12VDC switched voltage to the plug-in modules. The switched voltages are turned on and off under program control.
- A number of options are available for controlling iPDX bus transactions. These options include:

  1) Using iPPS software to supervise the uploading and execution of firmware from the plug-in module.

  2) Using a user-written driver program to supervise the uploading and execution of firmware from the plug-in module.

  3) Using a user-written driver program to control all iPDX bus activity.

  4) Using a user-written monitor program to allow control of iPDX bus activity from the system console.
- The plug-in modules that interface with the iPDX bus enable easy and fast changes of entire I/O subsystems.
- A prototyping tool (product code iPDS-PROTO) allows users to quickly design and build custom plug-in modules.
- The resources of a powerful, general-purpose development system (the iPDS system) are available to plug-in modules that use the iPDX bus.

Advantages and Limitations of iPDX Bus Implementation

The system (iUP or iPDS) that the iPDX bus is implemented on offers advantages for and imposes limitations on plug-in module use. The user's design requirements may dictate that the plug-in module be used with only one of the available systems. Plug-in modules that are universal must be designed to avoid the limitations of both systems.

iUP/iPDX BUS ADVANTAGES AND LIMITATIONS

Plug-in modules used with the iUP system are normally restricted to PROM-type programming functions. Table 1 lists the advantages and limitations of the iUP/iPDX Bus.

iPDS™/iPDX BUS ADVANTAGES AND LIMITATIONS

Plug-in modules used with the iPDS system can make use of all the features listed in the iPDX Bus Features section on page 4. The limitations for an iPDS/iPDX bus plug-in module are in the amount of power available from some of the voltage supply lines. Table 2 lists the advantages and limitations of the iPDS/iPDX Bus.

iPDX Bus Functional Description

The iPDX bus is an extension to the CPU bus of the iUP or iPDS system. The iPDX bus is active in the I/O address range 10H–1FH of the controlling CPU. Figure 1 is a functional block diagram of the iPDX bus as implemented on the iUP system. Figure 2 is a functional block diagram of the iPDX bus as implemented on the iPDS system.

iUP/iPDX BUS IMPLEMENTATION

The iPDX bus is the only I/O interface for the iUP-200/201 Universal Programmer, other than the serial interface of the iUP system. The iUP system normally performs one function, the programming of PROM-type devices. Intel PROM-type devices include EPROMs, E²PROMs, and the EPROM portion

Table 1. iUP/iPDX Bus Implementations

Advantages	Limitations
The iUP system provides ample power for programming any type of PROM device. Two variable supply voltages are available for plug-in module use. The I/O space of the iUP system is mostly unused, so operation in unused I/O space is possible.	Direct control of CPU operation is only possible using uploaded plug-in module firmware. The V_{CC} line supplies a maximum of 1.0A to the plug-in module .

Table 2. iPDS™/iPDX Bus Implementations

Advantages	Limitations
The resources of the iPDS system (RAM, console, mass storage, etc.) are available to the plug-in. The user has the option of using iPPS software or user-written programs to control the plug-in module. Any PROM programming module that works with the iPDS system and iPPS software also works with the iUP system. The V_{CC} supply line can handle up to 2.5A draw. This draw is adequate for most user applications.	Only one of the variable supply voltages (+VHSW) is available on the iPDS bus. The other variable line (+VLSW) has as fixed output of +12VDC. Power supplied to the iPDX bus is not adequate for gang programming modules.

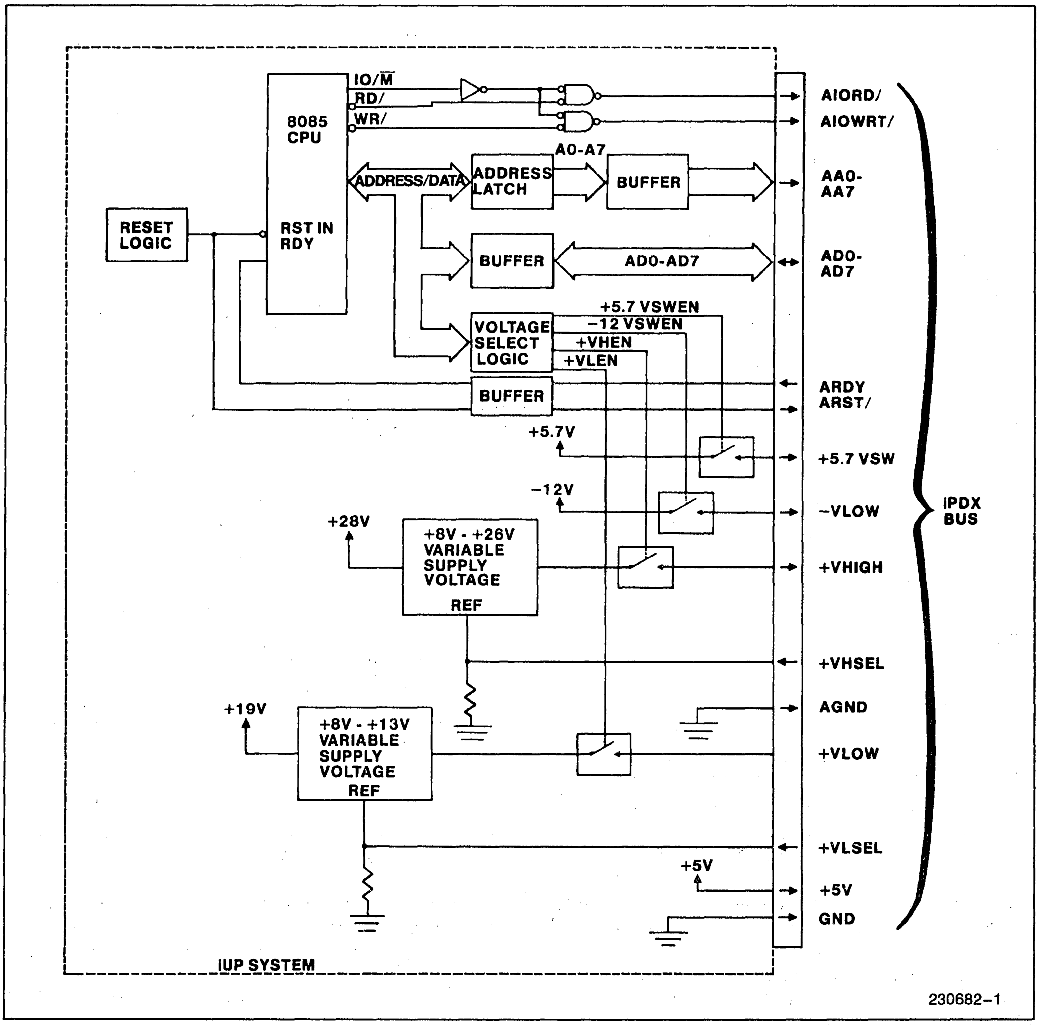


Figure 1. iUP/iPDX Bus, Functional Block Diagram

of various microcontrollers. The iUP system can program non-Intel PROM-type devices, but in most cases a personality plug-in module for the non-Intel device must be designed by the user. Note, however, that the Intel iUP-Fast 27/K PROM programming module (with firmware change) can program any 28-pin JEDEC device.

The iPDX bus implementation on the iUP system is optimized for maximum programming power capabilities. Each of the switched voltage supply lines from the iUP system provides at least twice the power of the corresponding line from an iPDS system. Refer to the Power Specifications (page 10) section for specific power capabilities.

The switched voltage lines are turned on and off under program control by the controlling CPU. The switched voltages are:

- +5.7VSW
- +VHIGH
- +VLOW
- −VLOW

Two of the switched voltages (+VHIGH and +VLOW) are variable. The +VLOW line provides +8V to +15V at 700 mA as determined by a precision resistance on the +VLSEL line. The +VHIGH line provides +8V to +27V at 300 mA as determined by a precision resistance on the +VHSEL line.

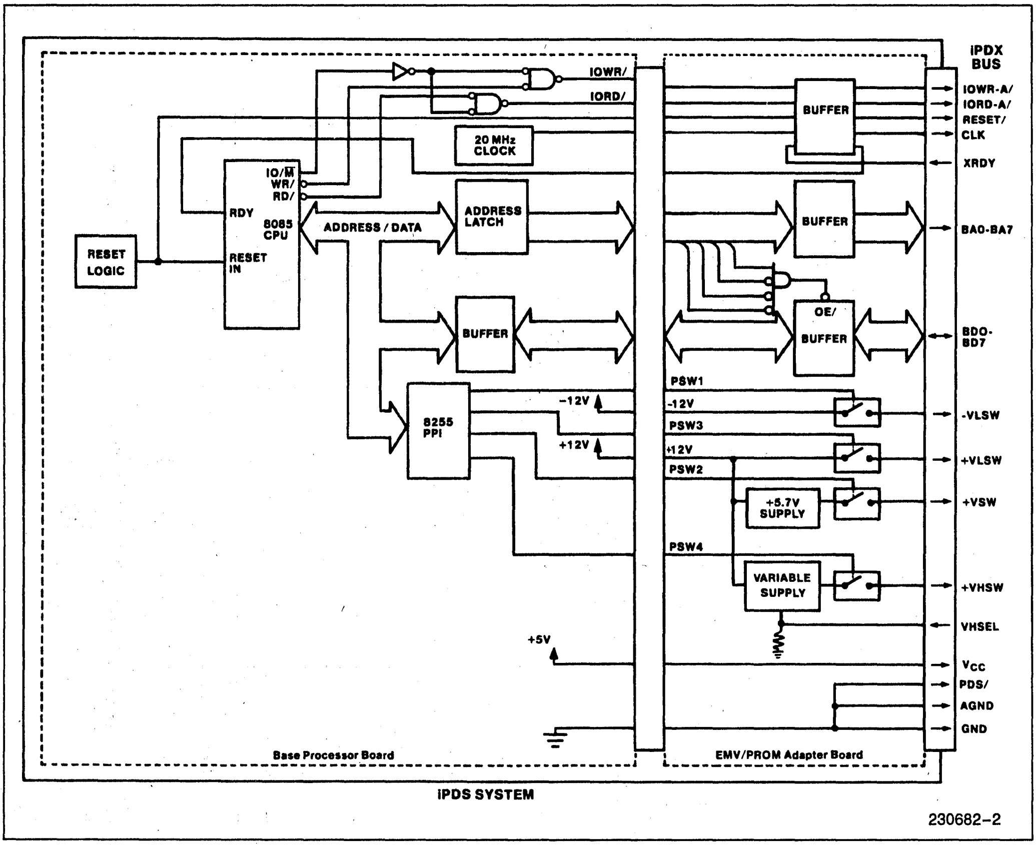


Figure 2. IPDS™/iPDX Bus, Functional Block Diagram

Refer to the Power Considerations (page 14) section for details on the control of the variable supply voltages.

The iUP/IPDX bus implementation provides not only program control of the switched voltage lines. It also allows monitoring of the on/off condition of these lines. The I/O ports used to control and monitor the switched voltages are discussed in the Switched Voltage Programming section (page 22).

Buffered data (AD0–AD7) is placed on the iPDX bus each time address line 4 (A4) is '1' during I/O accesses by the controlling CPU. This ensures that the data lines will be active for I/O addresses of 10H to 1FH. It also places data on the bus for addresses of 3XH, 5XH, 7XH, 9XH, BXH, DXH and FXH. The iPPS software only uses I/O addresses of 1XH when initially contacting the plug-in module, so there is no problem with this I/O addressing.

The address, read, write, reset and ready lines feed directly from the iUP system to the plug-in module on the iPDX bus. Figure 1 is a functional block diagram of the iUP system that shows the iPDX signals, their direction of flow, and the controlling circuitry in the iUP system. Refer to other sections of this application note for specific details on iUP/iPDX bus implementation.

iPDS™/IPDX BUS IMPLEMENTATION

The iPDS system implementation of the iPDX bus is a powerful, general-purpose interface to plug-in modules. The iPDS interface has less power handling capabilities than the iUP interface, but it has additional system resources.

The iPDX/iPDX bus interface uses a separate board in the iPDS system. The iPDS-140 option for the iPDS system is an interface between the iPDX bus and the base processor board of the iPDS system. The iPDS-140 option buffers all address, data, and control signals that go to the iPDX bus. The top address nibble is decoded on the iPDS-140 option to enable data transfers during reads or writes to I/O addresses 10H to 1FH.

The switched voltages for the iPDX bus are developed on the iPDS-140 option. The iPDS-140 option uses +12VDC and −12VDC from the iPDS system to generate the switched voltages. Refer to the Power Specifications and Power Considerations sections (pages 10 and 14) for details on the power available for the iPDX bus.

The switched voltages are under program control of the CPU in the iPDS system. These control signals are sent through an 8255 PPI chip to the iPDS-140 option. Refer to the Programming Switched Voltages section (page 22) for details on switched voltage control.

The V_{CC} (+5VDC) and ground lines from the base processor board are fed directly to the iPDX bus. The PDS/ and AGND lines of the iPDX bus are connected to the ground line within the iPDS-140 option. The PDS/ line is used by PROM programming plug-in modules to indicate the controlling system to iPPS software. All PROM programming plug-in modules feed the PDS/ line (J1−20) back so iPPS software can read its '1' or '0' status. Refer to the iPPS Software Protocol section (page 14) for details on the module status byte.

The address, read, write, reset, clock, and ready lines are buffered on the iPDS-140 option, but they are not modified by the iPDS system. Figure 2 is a functional block diagram of the iPDS system that shows the iPDX signals, their direction of flow, and the controlling circuitry in the iPDS system. Refer to other sections of this application note for specific details on iPDS/iPDX bus implementation.

iPDX BUS SPECIFICATIONS

The specifications for the iPDX bus are divided into four categories:

- Signal listings and descriptions.
- Detailed power (DC) specifications.
- Detailed timing (AC) specifications.
- Outline drawings and detailed mechanical specifications.

iPDX Bus Signal Descriptions

Table 3 presents the pinout of the iPDX bus and gives the associated signal names for both the iPDS and iUP systems.

Table 4 lists the signal names (iPDS and iUP systems) of the iPDX bus and gives a short description of each group of signals.

Table 3. iPDX Bus Pinout

Pin	iPDS Mnemonic	iUP Mnemonic	Input/ Output	Pin	iPDS™ Mnemonic	iUP Mnemonic	Input Output
1	GND	GND	O	22	GND	GND	O
2	GND	GND	O	23	Reserved	Reserved	N/A
3	BA0	AA0	O	24	BD0	AD0	I/O
4	BA1	AA1	O	25	BD1	AD1	I/O
5	BA2	AA2	O	26	BD2	AD2	I/O
6	BA3	AA3	O	27	BD3	AD3	I/O
7	BA4	AA4	O	28	BD4	AD4	I/O
8	BA5	AA5	O	29	BD5	AD5	I/O
9	BA6	AA6	O	30	BD6	AD6	I/O
10	BA7	AA7	O	31	BD7	AD7	I/O
11	V_{CC}	+5V	O	32	Reserved	Reserved	N/A
12	V_{CC}	+5V	O	33	+VHSW	+VHIGH	O
13	+VSW	+5.7VSW	O	34	+VLSW	+VLOW	O
14	+VSW	+5.7VSW	O	35	Reserved	Reserved	N/A
15	CLK	Not Used	O	36	−VLSW	−VLOW	O
16	IOWR-A/	AIOWRT/	O	37	AGND	AGND	O
17	IORD-A/	AIORD/	O	38	+VHSEL	+VHSEL	I
18	RESET/	ARST/	O	39	Not Used	+VLSEL	I
19	XRDY	ARDY	I	40	GND	GND	O
20	PDS/	PDS/	O(iPDS)	41	GND	GND	O
21	GND	GND	O				

Table 4. iPDX Bus Signal Descriptions

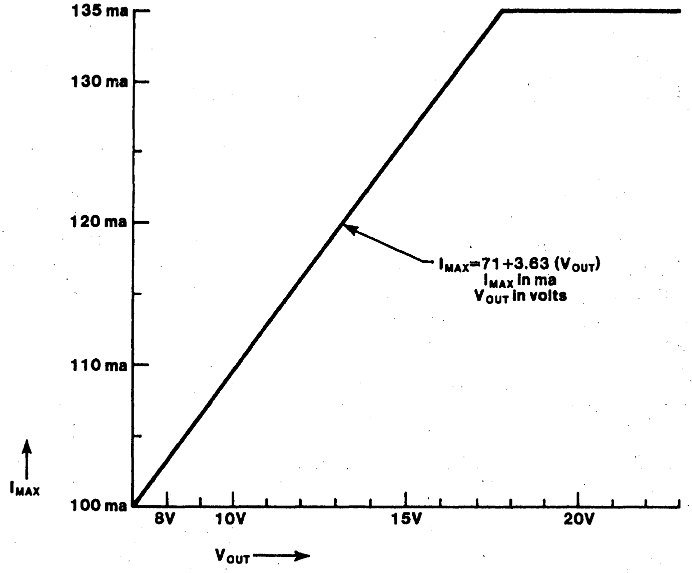
Signal Name(s)		Description
iPDS™	iUP	
GND	GND	Reference potential for all signals and supply voltages.
AGND	AGND	Analog ground. Reference potential for the programmable high voltage signal (+VHSW or +VHIGH).
BA0–BA7	AA0–AA7	Address lines from the iPDS system or the iUP system that define the I/O register to be accessed
BD0–BD7	AD0–AD7	Bi-directional, parallel data lines between the plug-in module, and the iPDS or the iUP system.
V_{CC}	+5V	Supply voltage for plug-in module circuitry.
CLK	Not Used	Clock signal (20 MHz) from the iPDS system.
IOWR-A/	AIOWRT/	I/O write signal from the iPDS or the iUP system. An active low indicates that output data from the iPDS or the iUP system is on the data lines. Data is sampled on the trailing edge of this signal.
IORD-A/	AIORD/	I/O read signal from the iPDS or the iUP system. An active low indicates that input data from the plug-in module should be placed on the data lines. Data is sampled on the trailing edge of this signal.
RESET/	ARST/	Reset signal from the iPDS or the iUP system.
XRDY	ARDY	Asynchronous ready signal from the plug-in module. An active high indicates that the plug-in module has accepted write data from, or presented valid read data to, the iPDS or the iUP system. A low level causes the iPDS or the iUP system to enter a wait state after either the IORD-A/ (AIORD/) or IOWR-A/ (AIOWRT/) line is activated.
PDS/	Not connected	A ground from the iPDS system. This signal is sampled by iPPS software and indicates that a PROM programming module is installed in an iPDS system.
+VSW	+5.7VSW	Switched +5.7VDC that can be turned on or off by the iPDS or the iUP system under program control.
+VHSW	+VHIGH	Switched +8VDC to +26VDC that can be turned on or off by the iPDS or the iUP system under program control. The actual voltage is determined by the +VHSEL signal from the plug-in module.
+VLSW	+VLOW	Switched +8VDC to +13VDC that can be turned on or off by the iPDS or the iUP system under program control For the iUP system the actual voltage is determined by the +VLSEL signal from the plug-in module. The iPDS system outputs only a fixed voltage of +12VDC on the +VLSW line.
−VLSW	−VLOW	Switched −12VDC that can be turned on or off by the iPDS or the iUP system under program control.
+VHSEL	+VHSEL	High plus programming voltage select (iPDS and iUP systems). A precision resistance in the plug-in module determines the voltage on the +VHSW (+VHIGH) line.
Not Used	+VLSEL	Low plus programming voltage select (iUP system only). A precision resistance in the plug-in module determines the voltage on the +VLOW line.

Power Specifications

The +5VDC line is always active on the iPDX bus. This line normally powers plug-in module circuitry. Switched voltages are also available to power plug-in module circuitry. The user must first set up appropriate driver routines and programming voltages before the switched voltage lines become active.

Table 5 lists the supply signals available at the iPDX bus and the specifications for each signal.

Figure 3 shows the power available on the iPDS +VHSW signal line for the programmable voltages. The other power supply signals give rated power over their full range.



230682-3

Figure 3. Power Available (iPDS™ +VHSW Signal)

Table 5. iPDX Bus Power Specifications

Signal Name		Supply Voltage and Tolerance		Maximum Current		Notes
iPDS™	iUP	iPDS	iUP	iPDS	iUP	
V_{CC}	+5V	+5VDC ±2.5%		2.5 amps	1.0 amps	
+VSW	+5.7VSW	+5.7VDC ±50 mv		250 mA	1.5 amps	1
+VHSW	+VHIGH	+8VDC to +27VDC ±2%		135 mA	300 mA	1, 2
+VLSW	+VLOW	+12VDC ±1.0V	+8VDC to +15VDC ±2%	200 mA	700 mA	1, 3
−VLSW	−VLOW	−12VDC ±0.5V		50 mA	100 mA	1

NOTES:

1. This voltage is switched and is under program control of the iPDS or the iUP system.
2. The voltage is controlled by the +VHSEL signal. Figure 3 shows the derating required for each selected voltage of +VHSW.
3. The voltage is controlled by the +VLSEL signal (iUP system only).

Electrical (DC) Specifications

The signal names for the iPDX bus indicate whether or not the signals are active high or active low. If the name ends with a slash (/), the signal is active low. If the name has no slash following it, the signal is active high. Table 6 shows the electrical specifications for the iPDX bus.

The electrical characteristics for the iPDX bus signals are shown in Table 7. The voltage and current specifications refer to the TTL high or TTL low state of the iPDX bus signal. The signal type (input or output) is the signal direction when viewed from the iPDS or the iUP system side of the iPDX bus. Positive currents are defined as currents entering the interface.

Timing (AC) Specifications

Figure 4 shows the timing specifications for the iPDX bus. Table 8 lists definitions of the timing parameters used for the iPDX bus. Refer to the *MCS®-80/85 Family User's Manual* or the *8085A-2* data sheet for specific details on the timing specifications for the iPDX bus.

The +VHSEL/+VLSEL signals, the data bus signals, and the ready (XRDY or ARDY) signal originate in the plug-in module. The voltage select and data bus signals have straightforward timing requirements, but the timing requirements for the ready signal need explanation.

CAUTION

230682-4

> Whenever the ready signal (XRDY or ARDY) goes low, the CPU generates wait-states until the ready signal returns high. The ready signal should not be driven low for more than a few bus cycles unless complete suspension of all CPU bus activity is allowable in the user's application.

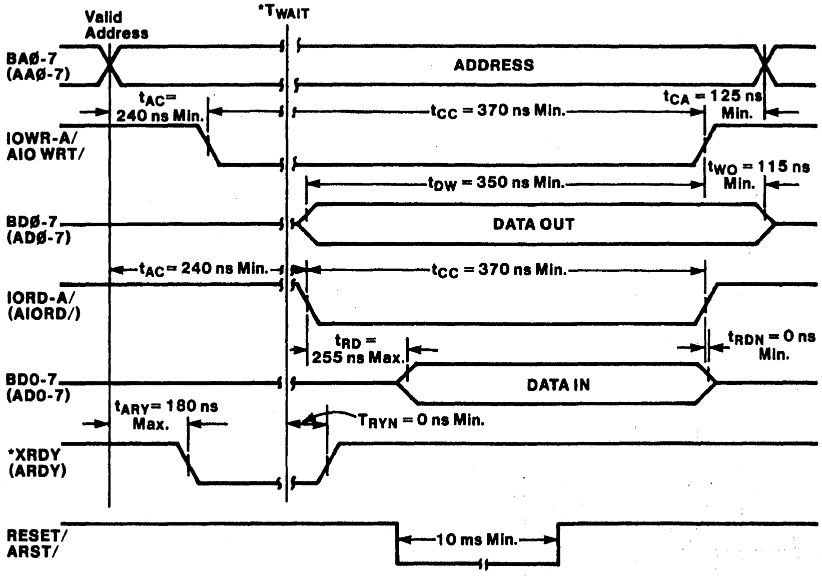
The ready signal is normally high for all read/write transfers over the iPDX bus. The ready signal can be driven low to insert one or more wait-states in the CPU bus cycle, in cases where the plug-in module uses slow memory devices or slow peripheral devices.

Table 6. iPDX Bus Electrical Specifications

Active State	Logical State	Electrical Signal Level	At Receiver	At Driver
LOW	0	H = TTL High State	5.25V ≥ H ≥ 2.0V	5.25V ≥ H ≥ 2.4V
	1	L = TTL Low State	0.8V ≥ L ≥ −0.5V	0.5V ≥ L ≥ 0.0V
HIGH	0	L = TTL Low State	0.8V ≥ L ≥ −0.5V	0.5V ≥ L ≥ 0.0V
	1	H = TTL High State	5.25V ≥ H ≥ 2.0V	5.25V ≥ H ≥ 2.4V

Table 7. Electrical Characteristics of iPDX Bus Signals

Signal Type	I_{OL} Max	I_{IL} Max	I_{OH} Max	I_{IH} Max	V_{OL} Max	V_{IL} Max	V_{OH} Min	V_{IH} Min
All Outputs	24 mA		−5 mA		0.5V		2.4V	
Inputs (except RDY signal)		−12.8 mA		50 μA		0.8V		2.0V
ARDY (input)		−4 mA		50 μA		0.8V		2.0V



*One or more wait states (T_{WAIT}) are inserted in the CPU bus cycle after the ready signal (XRDY or ARDY) goes low.

230682-5

Figure 4. iPDS Bus Read/Write Timing

Table 8. iPDX Timing Definitions

Symbol	Description
t_{AC}	The time between valid address (A0–A7) and the leading edge of the control signal.
t_{ARY}	The time between valid address (A0–A7) and the trailing edge of the ready signal.
t_{CA}	The time between the trailing edge of the control signal and the end of valid address.
t_{CC}	The width of the control signal.
t_{DW}	The time between the start of valid data (D0–D7) and the trailing edge fo the write control signal.
t_{RD}	The time between the leading edge of the read control signal and the start of valid data (D0–D7).
t_{RDH}	The time between the trailing edge of the read control signal and the end of valid data (D0–D7).
t_{RYH}	The time between the end of T_{WAIT} and the leading edge of the ready signal.
t_{WD}	The time between the trialing edge of the write control signal and the end of valid data (D0–D7).

Mechanical Specifications

The mechanical specifications define the connector requirements and the outline and mounting dimensions for plug-in modules using the iPDX bus. Figure 5 is an outline drawing of a plug-in module for the iPDX bus. All plug-in modules for the iPDX bus must comply with the dimensions specified in Figure 5.

HARDWARE DESIGN CONSIDERATIONS

Plug-in modules designed around the iPDX bus must follow certain design rules. These design rules are:

- The first four inches (measured from the connector end) of the plug-in module must meet the mechanical and outline specifications shown in Figure 5.

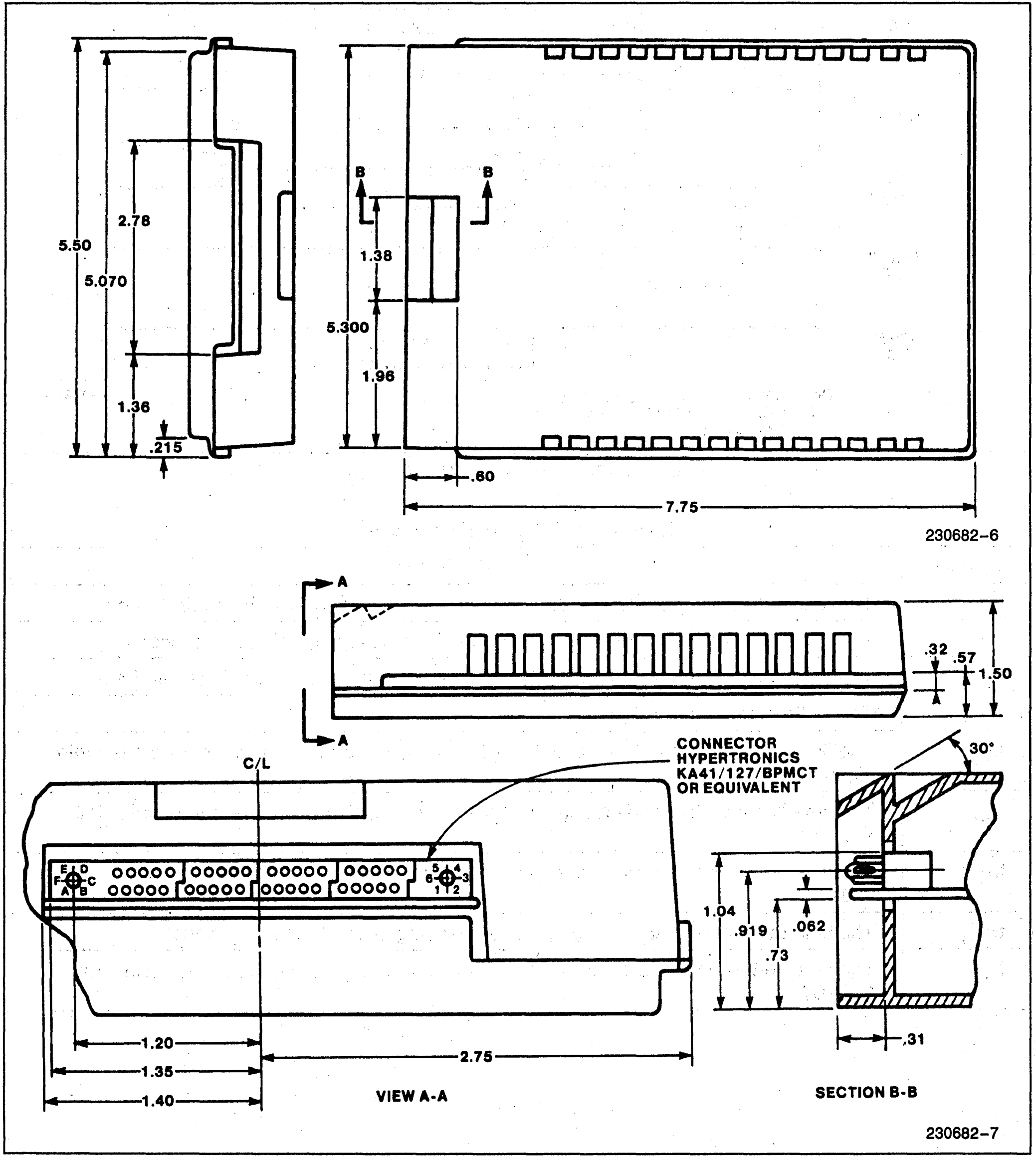


Figure 5. Plug-In Module Mechanical Specifications

- The maximum V_{CC} (+5VDC) current available is 2.5 amps for iPDS plug-in modules or 1.0 amps for iUP plug-in modules.
- Switched voltages of +5.7VDC, +8VDC to +27VDC, +12VDC and −12VDC are available to circuitry on a plug-in module under program control Table 5 lists power specifications for the iPDX bus.
- If a programmed voltage (positive only) is required by the plug-in module, an appropriate precision resistor must be installed in the plug-in module.
- All signals (except +VHSEL and +VLSEL) returned by the plug-in module must be TTL levels.
- Provisions must be made to sample the PDS/ signal on PROM programming plug-in modules that use iPPS software while connected to an iPDS system. (The PDS/signal is low when the module is connected to the iPDS system and floating when connected to the iUP system. Firmware can use the signal 1) to specify whether a power supply status port is available, 2) to specify whether E3H (iPDS) or 03H (iUP) is the correct port for turning on power supplies, and 3) to compensate for differences in timing between the two systems.)
- Direct memory access (DMA) transactions are not supported on the iPDX bus.

Mechanical Considerations

Plug-in modules for the iPDX bus must have an enclosure that meets the mechanical specifications shown in Figure 5 for the first four inches (measured from the connector end) of the module. Intel has developed a prototyping kit (product code iPDS-PROTO) to simplify the mechanical and hardware portions of the design. This prototyping kit consists of the plug-in module enclosure, a prototyping board, iPDX bus connector, a hardware kit, isolation capacitors, and wire-wrap pins. The iPDS-PROTO kit can accept up to 30 ICs and associated discrete components in the available board space. If a plug-in module designed around the iPDX bus goes to a production phase, use of the module tooling can be licensed through Intel.

Power Considerations

The maximum power dissipation for an iPDS plug-in module is 20.5 watts with a maximum draw of 12.5 watts from the V_{CC} line. The maximum power dissipation for an iUP plug-in module is 32.5 watts with a maximum draw of 5.13 watts from the V_{CC} line and 8.625 watts from the +5.7 VSW line. A maximum of 7.5 watts can be dissipated within a plastic plug-in module (more power can be dissipated at the PROM socket).

V_{CC} (+5VDC) is the only voltage present at all times on the iPDX bus. If the plug-in module circuitry requires other voltage levels for operation, the switched voltages must be turned on first by software. The Programming Considerations section shows the iPDX bus set-up requirements for turning on/off each of the switched voltage signals.

The variable switched voltages (+VHSW on an iPDS plug-in module, and +VHIGH and +VLOW on an iUP plug-in module) use one or more precision resistors on the plug-in module to determine their line voltage. The precision resistor on the plug-in module must be connected between the AGND line and the +VHSEL line of the iPDX bus. Plug-in modules for an iUP system can also program the +VLOW line by connecting a precision resistor between the AGND line and the +VLSEL line of the iPDX bus. Figure 6 shows a chart and two equations that indicate the precision resistor values corresponding to programmable voltages. Figure 7 shows three kinds of circuits that allow the plug-in module to select more than one programming voltage level.

PROGRAMMING CONSIDERATIONS

PROM programming modules are normally controlled by iPPS software residing in either the iPDS or the iUP system. User-designed plug-in modules (other than programming plug-in modules) are controlled by user-supplied driver programs. The iPPS Software Protocol section explains the iPPS-iPDX bus interface. The Switched Voltage Programming section gives programming requirements for accessing switched voltages in the iPDS/iPDX bus interface. The User-Written iPDX Bus Drivers section presents the programming requirements for user-supplied driver programs.

iPPS Software Protocol

PROM programming plug-in modules that run under control of iPPS software must contain firmware. The firmware in the PROM programming module is a program that has routines for programming the device(s) that the plug-in module is designed to program. This firmware is uploaded into RAM in the controlling (iPDS or iUP) system the first time the TYPE command in the iPPS command language is executed. After the module firmware is uploaded, the iPDS or the iUP system controls the programming operation. The iPPS software communicates with the plug-in module over 7 of the 16 I/O ports allocated for iPDX bus communication. Table 9 lists the I/O port assignments recognized by iPPS software.

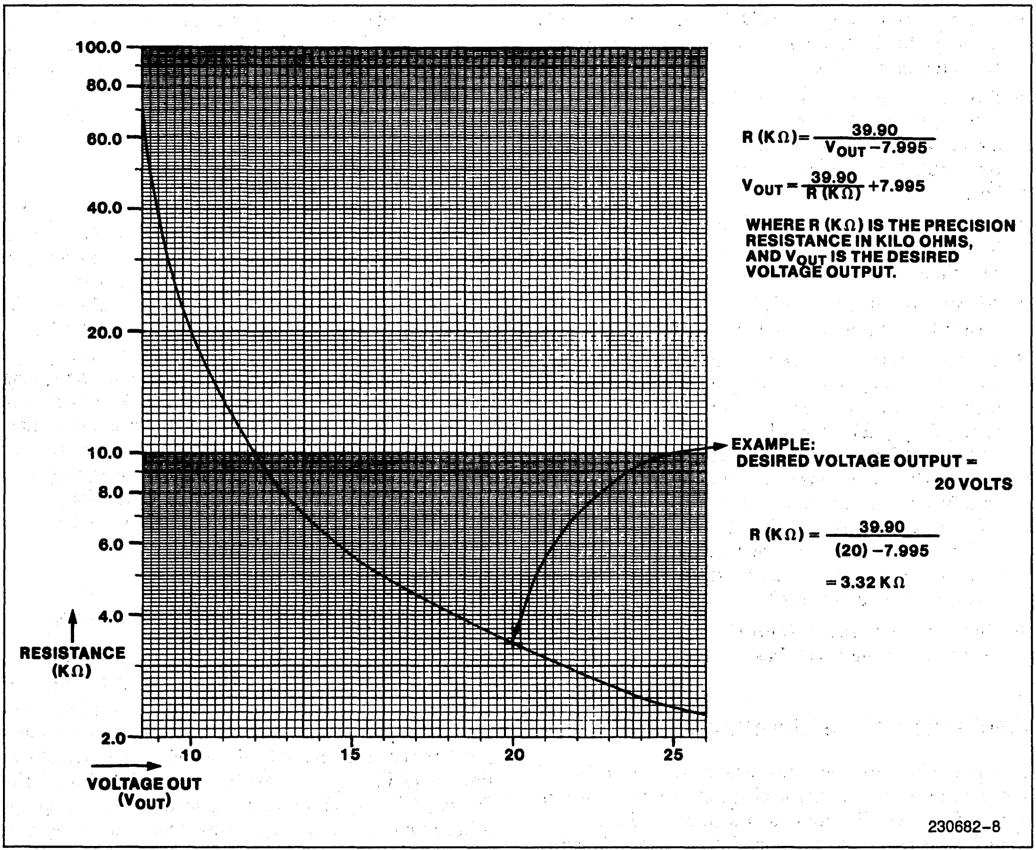


Figure 6. Programmable Voltage Resistor Values

The control words, corresponding to an I/O write to port addresses 10H, 11H and 12H, control various functions on the plug-in module. These functions may include voltage select and routing for the target PROM socket, the programming pulse, or chip selects, and set/clear the upload flag. The bit definitions for the control words are shown in Figure 8.

The status word, corresponding to an I/O read of port address 10H, contains information about the current state of monitored functions on the plug-in module. The bit definitions for the status word are shown in Figure 9.

The plug-in module firmware is read when the iPPS TYPE command is first executed. The iPPS software uploads plug-in module firmware by writing the plug-in module PROM location to I/O ports 13H (A0–A7) and 14H (A8–A15), respectively, and then reading the data at I/O port 11H. The plug-in module firmware uploads to absolute address 7020H in the iPDS or iUP system. After the plug-in module firmware is uploaded to the iPDS or the iUP system, the upload flag (bit 1 of control word 0) is set by the controlling system. Setting the upload flag causes bit 1 of the status word to indicate that additional firmware uploads are not required.

PLUG-IN MODULE FIRMWARE

The firmware (for plug-in modules running under control of iPPS software) controls all plug-in module operation, except the firmware upload operation itself. This firmware must be written in 8085 code and formatted as shown in Table 10.

The first two bytes of plug-in module firmware must contain the total number of bytes to be uploaded (including the two length bytes and the two check-sum bytes). The third byte must contain the number of different devices the plug-in module can read or program.

The plug-in module firmware is divided into segments and a segment is required for each PROM type that the module can program. Each segment contains a descriptor (first 14 bytes) and a code section.

Descriptor Section

The first two descriptor bytes contain the address of the next segment of firmware. The last segment of the firmware must contain the address of the first segment. If there is only one segment, the segment must reference itself.

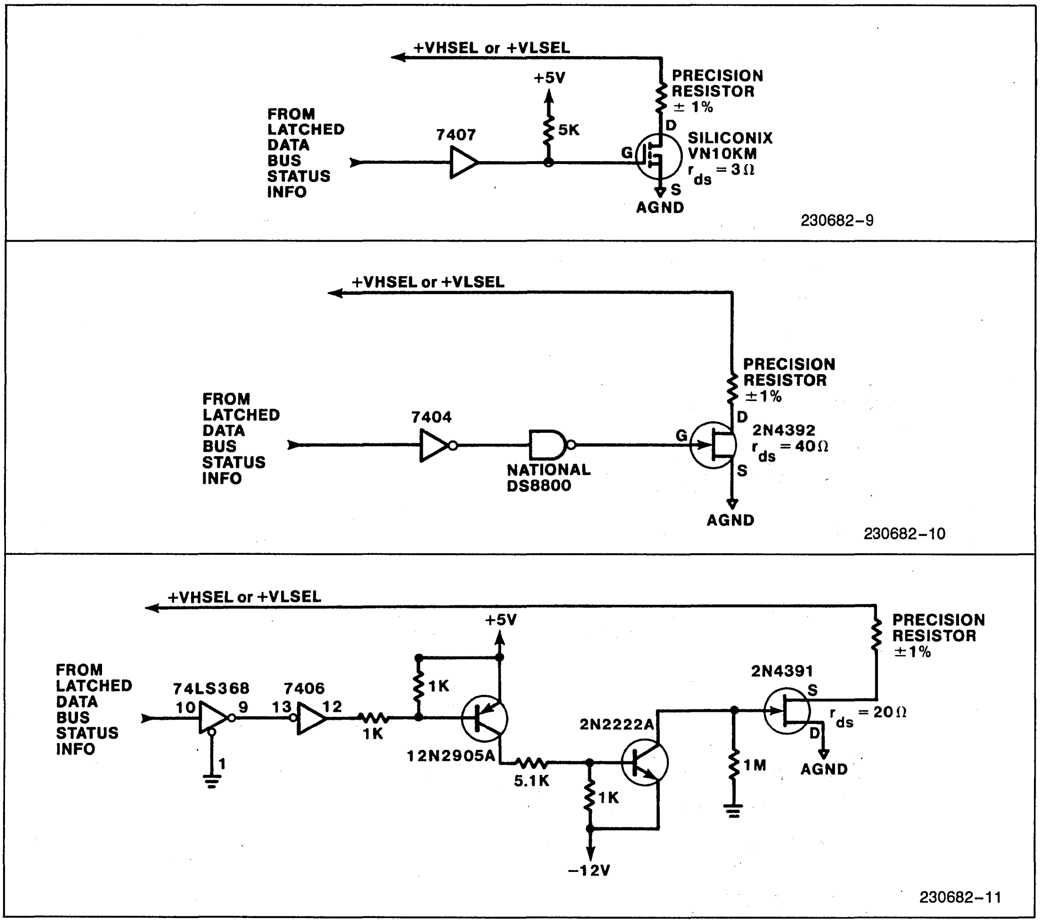


Figure 7. Three Precision Resistor Switching Circuits

Table 9. I/O Port Assignments Used by iPPS Software

I/O Port Address	I/O Write Active	I/O Read Active
10H	Write control word 0	Read module status
11H	Write control word 1	Read personality PROM data
12H	Write control word 2	Available
13H	Write address (A0–A7)	Available
14H	Write address (A8–A15)	Available
15H	Write address (A16–A19)	Available
16H	Write data (D0–D7)	Read device data

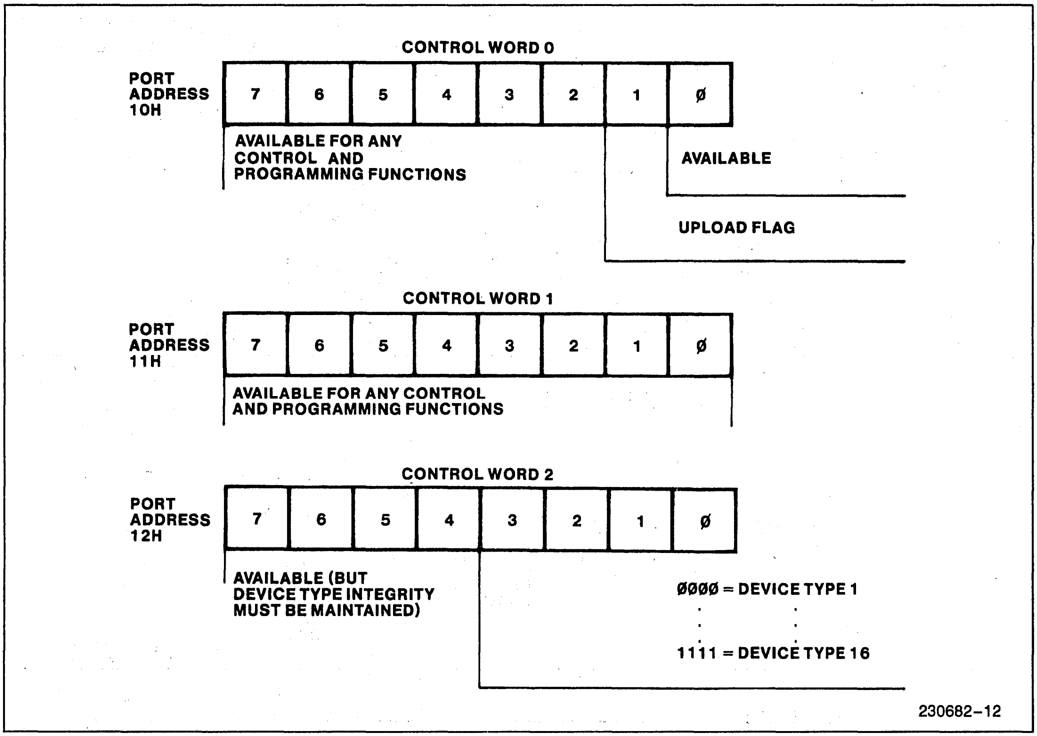


Figure 8. iPPS Control Word Bit Definitions

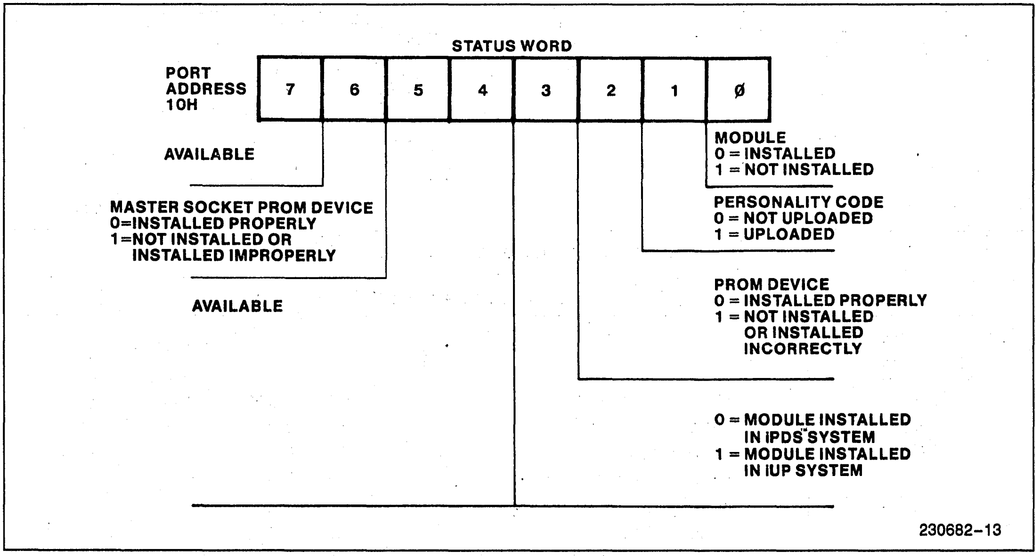


Figure 9. iPPS Status Word Bit Definitions

Table 10. Plug-In Module Firmware Format

Personality Prom Address			Contents
SEGMENT	DESCRIPTOR	0	8 LSBS of the length of the personality PROM.
		1	8 MSBS of the length of the personality PROM.
		2	Number of types the module can program.
		3	8 LSBS of the address of the next segment in the table (U).
		4	8 MSBS of the address of the next segment in the table (U).
		5	1st ASCII character of PROM type.
		6	2nd ASCII character of PROM type.
		7	3rd ASCII character of PROM type.
		8	4th ASCII character of PROM type.
		9	5th ASCII character of PROM type.
		10	6th ASCII character of PROM type.
		11	7th ASCII character of PROM type.
		12	8th ASCII character of PROM type.
		13	8 LSBS of PROM address range.
		14	8 MOBS of PROM address range.
		15	8 MSBS of PROM address range.
		16	Bits 0–5 indicate PROM word length. Bit 6 indicates the blank state of the PROM. Bit 7 is not used
	CODE	17	Jump to blankcheck routine (V).
		18	8 LSB of address of blankcheck routine.
		19	8 MSB of address of blankcheck routine.
		20	Jump to program routine (W).
		21	8 LSB of address of program routine.
		22	8 MSB of address of program routine.
		23	Jump to overlay check routine (X).
		24	8 LSB of address of overlay check routine.
		25	8 MSB of address of overlay check routine.
		26	Jump to reverse socket routine (Y).
		27	8 LSB of address of reverse socket routine.
		28	8 MSB of address of reverse socket routine.
		29	Jump to read routine (Z).
		30	8 LSB of address of read routine.
		31	8 MSB of address of read routine.
		V	Start blankcheck code.
		V+N	"RETURN"
		W	Start code for program routine.
		W+N	"RETURN"
		X	Start code for overlay check.
		X+N	"RETURN"
		Y	Start code for reverse socket routine.
		Y+N	"RETURN"
		Z	Start code for read routine.
		Z+N	"RETURN"
Next segment (U)			
Next two locations after last byte of last segment			Checksum (LSB) Checksum (MSB)

The next eight descriptor bytes contain the ASCII code for the device being programmed. Spaces (ASCII code 20H) must be used to fill any unused bytes of this ASCII code.

The remaining four descriptor bytes contain specific PROM device information, with the first three bytes holding the available PROM address range and the final byte holding PROM data information. Bits 0–5 of the PROM data information byte contain the word length (binary equivalent in bits) of the selected PROM. Bit 6 of the PROM data information byte indicates the unprogrammed state of each PROM bit (i.e., a 0 in the bit 6 location means a device bit is unprogrammed in the high state and programmed in the low state). Bit 7 of the PROM data information byte is not used.

Code and Checksum Sections

The code section is subdivided into a jump op code section followed by blankcheck, program, overlay check, reverse socket detect, and read routines.

The jump op code section contains the jump op codes and addresses of each programming routine for the device covered in this segment. The programming routines referenced in this section include read, blankcheck, program, overlay check, reverse socket detect, and read. The referenced routines may actually reside in other segments.

The blankcheck, program, overlay check, reverse socket detect, and read programming routines must be in 8085 code. These routines are hardware specific instructions for checking and programming the device. The following subsections describe relevant details of these routines and provide other information needed to develop module firmware.

The final two bytes of firmware following the last segment contain the checksum for the plug-in module firmware chip. The checksum is the 2's complement of the sum of the previous bytes in the plug-in module firmware chip.

Memory Variable and Stack Locations—Memory locations 6000H to 60FFH are reserved for variables and stack. Please note that this leaves space for a very small stack. The following is a list of variables that the user needs to know to interface to iPPS software.

6000H Lowest address for 80 bytes of input buffer.

6050H Lowest address for 80 bytes of output buffer; space is also used for variables when PROMs greater than 32K bytes are edited.

601AH Used to indicate on-line (00H) or off-line (01H) operation.

60A2H Used to pass the current status of the iUP programmer to the iPPS software.

60B4H-60B5H Both 60B4H and 60B5H are general purpose locations for passing information. See information in this section on creating firmware for displaying messages on the host.

60B6H Used to indicate when powering down has finished, i.e., when an operation has been completed. The module firmware should set this location to 01H when power is turned on. This location is reset to 00H when the power is shut off. This information is needed by the iPDS system, since the iPDS system does not have a status port (such as 02H in the iUP programmer) to indicate whether power is on or off.

60B7H For passing an address between module and iPPS software: contains LSB of address.

60B8H For passing an address between module and iPPS software: contains MOB of address.

60B9H For passing an address between module and iPPS software: contains HOB of address.

60BAH Contains data to be programmed from the iUP programmer to PROM.

60BBH Contains data read from PROM to iUP programmer.

60CCH Indicates operation in process. Used in off-line keyboard interrupts. See keyboard interrupt routine below.

60CFH Used for the lock function. The iPPS software sets this location to 00H before calling the reverse socket check. The module firmware sets this location to FFH if a lock function is available or leaves it at 00H to indicate that no function is available. (This ensures backwards compatibility with older modules.) The iPPS software then sets this location to 01 before calling the programming routine. This value indicates to the module that lock (rather than programming) is requested. (If programming is requested, the value is 00H.)

60D0H Used in the lock function. The module firmware uses this location to indicate which parameter is being passed. On modules that just lock (like 8751AH), the lock sequence will never go above 1.

On authenticated PROMs, the sequence numbers may be greater than 1. This allows the module, iPPS software, or user to edit the parameters. The parameters should be stored in a buffer and this location is used to index the buffer. If the user responds NO to the EXECUTE query, module firmware should reset this location to the beginning (0). The buffer values (instead of the PROM's actual values) are then sent back. These locations are programmed only when the user responds YES to the EXECUTE query. Module firmware should be set to 0 when finished.

60D2H Indicates a PROM that is greater than 32K bytes has been edited. (00H = NO; 01H = YES).

60D3H Indicates whether the module should be using the programming socket. There is a bug in the initialization of this flag, so until iPPS-PDS software and the iUP programmer firmware are upgraded, the module firmware needs to set this location as follows:

(1) For PROMs less than 32K bytes, set to 00.

(2) For all devices when on-line, set to 00.

This covers the two conditions in which the master socket will never be accessed.

60FFH Top of the stack.

Parameters for Major Subroutines—Unless otherwise noted, the module returns results using the following codes:

00H means "pass".
12H means "power supply failure".
07H means "abort".

Information on Code Section Routines—The following paragraphs provide information on routines included in the code section of the PROM programming firmware. Note that the meaning of "iUP programmer" in these paragraphs depends on the system being considered. "iUP programmer" can mean either iUP-200A/201A firmware or iPPS-PDS software.

Blank Check Routine—The iUP programmer passes no parameters to the module. The module firmware checks the entire PROM and passes back results in the B register. (Fail = 05H.) If the PROM fails the blank check test, the actual value of the PROM is passed back in 60BBH and the location in 60B7H, 60B8H, and 60B9H. In the off-line mode, any undefined value in B defaults to abort.

Program Routine—The iUP programmer sends the location to be programmed in 60B7H, 60B8H, and 60B9H, and sends the data to be programmed in 50BAH. It also resets 60CFH to 00H. The module returns results in the A register. (Fail = 01.) In the off-line mode, any undefined results default to abort. If the programming failed, the actual value of the PROM is passed back in 60BBH and the location in 60B7H, 60B8H, and 60B9H. The off-line error message will show the address of the failure and user data XOR PROM data. In the on-line mode, the host console will the show failure address, user data, and PROM data.

Overlay Check Routine—The iUP programmer passes no parameters. The iPPS software does not use the overlay check routine; it does its own overlay check on the portion of PROM to be programmed.

In the off-line mode, data the user wants to program is in memory starting at 8000H, and the entire PROM is checked with results sent back in the B register. (Fail = 01.) The module firmware may also send back 03H in the B register to indicate that the iUP programmer should perform the overlay check (on edited PROMs greater than 32K, the iUP programmer automatically performs the overlay check). Any undefined result defaults to abort.

The iUP programmer uses the following algorithms to determine whether the new user data can be programmed over a nonblank PROM location:

1. For PROMs with FFH as a blank state:

   IF [(user data AND PROM data) XOR user data = 0] THEN overlay is possible

2. For PROMs with 00H as a blank state:

   IF [(user data XOR PROM data) AND PROM data = 0] THEN overlay is possible

Reverse Socket Check Routine—The iUP programmer indicates in 60D3H which socket to check and initializes 60CFH to 00H. The module sends back results in the A register. (Fail = 04H.) In the off-line mode, the iUP programmer only recognizes pass, abort, and will default to fail for any other unrecognized result. On chips which support the lock function, 60CFH is set to FFH; on old modules or for chips that do not support the lock function, 60CFH is left at 00H. Addition of other initialization tests can be accomplished by adding these tests to the module reverse socket code. Then, if an error occurs, the module can send a specific error message and abort.

Read Routine—The iUP programmer passes the location to be read in 69B7H, 60B8H, and 60B9H; a code for the (master or program) socket that is to be read from is passed in SKTFLG. The module passes the data read in 60BBH and the result in the A register.

NOTE:

There is no failed status, only pass, abort, or power supply failure, In the off-line mode, any undefined result defaults to power supply failure.

Lock Routine—The iUP programmer checks module installation, sets location 60CFH to OOH, and performs the reverse socket test. If 60CFH still equals OOH after the reverse socket check, then the lock function is not available for that module and/or chip. If, however, 60CFH equals 01H after a reverse socket check, then the lock function is available; 60CFH will remain at 01H until the command is finished.

Next (with 60CFH = 01 and 60D0H = 00H), the iUP programmer calls the program subroutine. The module firmware can then communicate with the user by returning (in the A register) one of the values shown in Table 11. When needed, the HL register pair points to the text to be displayed (where the first byte of the message is the length of the message). Handshaking will continue until the result returned is OOH or one of the aborts occurs. (During this process, data sent by the user is contained in location 60BAH and data from the PROM or buffer is contained in 60BBH.) If data values are required, the module stores these values in a buffer (in the module firmware) using 60D0H as an index. No programming or locking is performed until the user has answered YES to the EXECUTE query. At this point, interrupts are disallowed.

Table 11. A-Register Results

Value	Meaning
OOH	Pass/done and 60D0H = 00H
02H	Continue and send message pointed to by HL registers
04H	Send execute query to user
07H	Abort (with message)
09H	Lock not available/illegal operation
0AH	Failed; send "PROM BLANK" message
0BH	Failed; send "LOCK FAILED" message
0CH	Failed; send "LOCK FAILED AT" message
0DH	Illegal parameter value
12H	Power supply failure
17H	Abort (without abort message)

Verify—On-line verification is performed by iPPS software using reads. Upon failure, the addresses, user data, and PROM data are displayed. Off-line verification is done by the iUP programmer firmware. Upon failure, the address and user data or PROM data are displayed. The user then has the option of pressing the VERIFY key again to continue verification or pressing the CLEAR key to abort.

Editing PROMS Larger than 32K Bytes—In the off-line mode, editing of PROMs greater than 32K requires a master socket and some special considerations. The iUP programmer has only 32K of image RAM; so, on PROMs greater than 32K, the iUP programmer expects a master PROM in the master socket. The iUP programmer uses this master PROM as the source for programming and overlay checks. (Note that for PROMs larger than 32K bytes, pressing the ROM-to-RAM key does not load data into the URAM. Thus, in using this method of expanding the editing features of the iUP programmer, it is no longer possible to load a 27512 into URAM and then copy URAM to a 27256.)

When the user wishes to edit (off-line) a PROM greater than 32K, data to be edited is copied in 1K blocks to the URAM. (Each 1K block copied always starts on a 1K boundary.) Up to thirty-one 1K blocks can be copied and edited; the last 1K of URAM is not available because this space is needed to manage the editing.

Power-Down Sequence—For current modules, there is an assumption that the module does not need to know when the iPPS software is going to shut off the power supplies; so, the module firmware cannot find this out. For modules that require a certain power-down sequence, there are two possibilities.

- Plan the module to correspond to the iPPS software power-down sequence:
  1. Port 11H is set to 0.
  2. 60B6H is set to 0.
  3. All bits in port 10H are set to 0 except bit 1 (the upload flag), which is not modified.
  4. All power supplies are shut off.
- Module firmware shuts off selected controls in the appropriate order until there is no danger when the iPPS software decides to shut off power supplies. The one check that may be needed is an off-line check. When off-line, the module always checks, reads, or programs the entire PROM—so that if the module is off-line and at the last address, then the iUP programmer will be powering down.

Creating Firmware for Displaying Messages on the Host—To send messages to be displayed by the host, use the following algorithm.

```
Check locations 60A1H to determine whether the host is the iUP programmer or iPDS system.
If host is the iUP programmer
  Call 7006H to blank the display
  Set HL to 6050H (output buffer)
  Insert a carriage return as the first character
  Fill in the message in the output buffer
  Increase the byte count of the message by 1 (for the carriage return) and place the count in the B register
  Set HL = 0
  Call 7003
If the host is on-line (i.e., if the host is the iPDS system)
  Set 60B4H = 21H to indicate message to iPPS software
  Fill in the message starting at 6054H (output buffer plus 3)
  Insert a carriage return and linefeed at the end of message
  Set B register = message length plus 6
  Call 7000H
```

(7000H and 7003H are actually jump tables to the real address. The jump tables are generated by iPPS software so that updates to iPPS software will be backwards compatible.)

Power Supply Status—There is no status register (02H) to read to tell whether the power supplies have been turned on in the iPDS. Thus, module firmware must monitor 60B6H, if the host is an iPDS. 60B6H is set to 0 upon initialization and when power supplies are turned off. The module firmware must set it to 1 when the power supplies are turned on and set it to 0 when the power supplies are turned off.

WAIT Routine Difference—The 250 microsecond WAIT routines in the iUP programmer and iPDS firmware are inaccurate for short periods of time and do not match each other exactly. (These routines were not revised to ensure backwards compatibility.) For precise timing, the user should write a loop taking into account the differences between the iUP programmer and iPDS clocks.

Use of the E Register—The E register is reserved for use in keyboard interrupts. The module may use the E register if interrupts are first disabled and a known value is restored before re-enabling interrupts. This use of the E register will cause no key presses to be serviced. It is much safer to leave the E register alone.

Keyboard Interrupt Logic—The keyboard interrupt logic is as follows.

```
Save PSW and HL
Save the character read in 60C1H
If the iUP programmer is on-line
  then if key pressed is the on-line key
    then E register = 81H
    else ignore key pressed
  else if key pressed is clear display
    then E register = 88H
   if 60CCH <> 0 /*if operation is process*/
   E register = 80H            /*value key press*/
  Restore PSW and HL
  Return
```

Switched Voltage Programming

There are four switched voltages on the iPDS bus that are turned on or off under program control. The iPDS and iUP systems use different I/O addresses for programming the switched voltages. Under iPPS software, the plug-in module firmware controls the switched voltages. Under user-prepared driver software, separate commands must be included to turn on or off the required switched voltages.

iUP SWITCHED VOLTAGE PROGRAMMING

The iUP system switches the +5.7 VSW, +VLOW, +VHIGH, and −12 VSW supply lines on and off under program control. The controlling program must write twice to I/O port 03H to set/clear and then clock (high to low transition) the switched voltage flip-flops. The first write to I/O port 03H must have bit 0 (clock) high and bits 1 through 4 set for the desired program voltages. The second write to I/O port 03H keeps bits 1 through 4 at the desired program voltage level while bit 0 goes low. The on/off status of each switched voltage line can be checked by reading I/O port 02H. The iUP system turns off a switched voltage supply line whenever an overcurrent condition is sensed on that line. Figure 10 contains switched voltage control and status bit definitions for the iUP system.

iPDS™ SWITCHED VOLTAGE PROGRAMMING

The iPDS system switches the +5.7 VSW, +VLOW, +VHIGH, and −12 VSW supply lines on and off under program control. The controlling program (either iPPS software or a user-written driver program) must write to I/O port E3H in order to turn on/off the required switched voltages. Figure 11 shows the bit definitions for programming the iPDS switched voltage lines.

User-Written IPDX™ Bus Drivers

User-written iPDX bus driver programs normally access plug-in modules designed for use with the iPDS system. A user-designed iPDX bus plug-in module can address a wide range of applications. The iPDX bus driver program for a user-designed plug-in module can range from simple (e.g., using a single I/O port to upload PROM data to the iPDS system), to complex (e.g., using nearly all the I/O ports to control a high-level instrumentation function).

The I/O ports available to the iPDX bus occupy addresses 10H through 1FH in the iPDS I/O space. Since both an I/O read and an I/O write are associated with each I/O address, the user has 32 I/O ports available for each driver program. Figure 12 is a blank chart that can be used to assign I/O addresses for a specific user driver program. Keep this chart for reference while writing the driver program.

The driver program must be written in 8085 code. Use no more than byte-wide transfers of address, data, and control information. The plug-in module can operate on information of virtually any bit length. The 8-bit width of the iPDX data bus imposes a byte-wide only requirement on all information transfers over the iPDX bus.

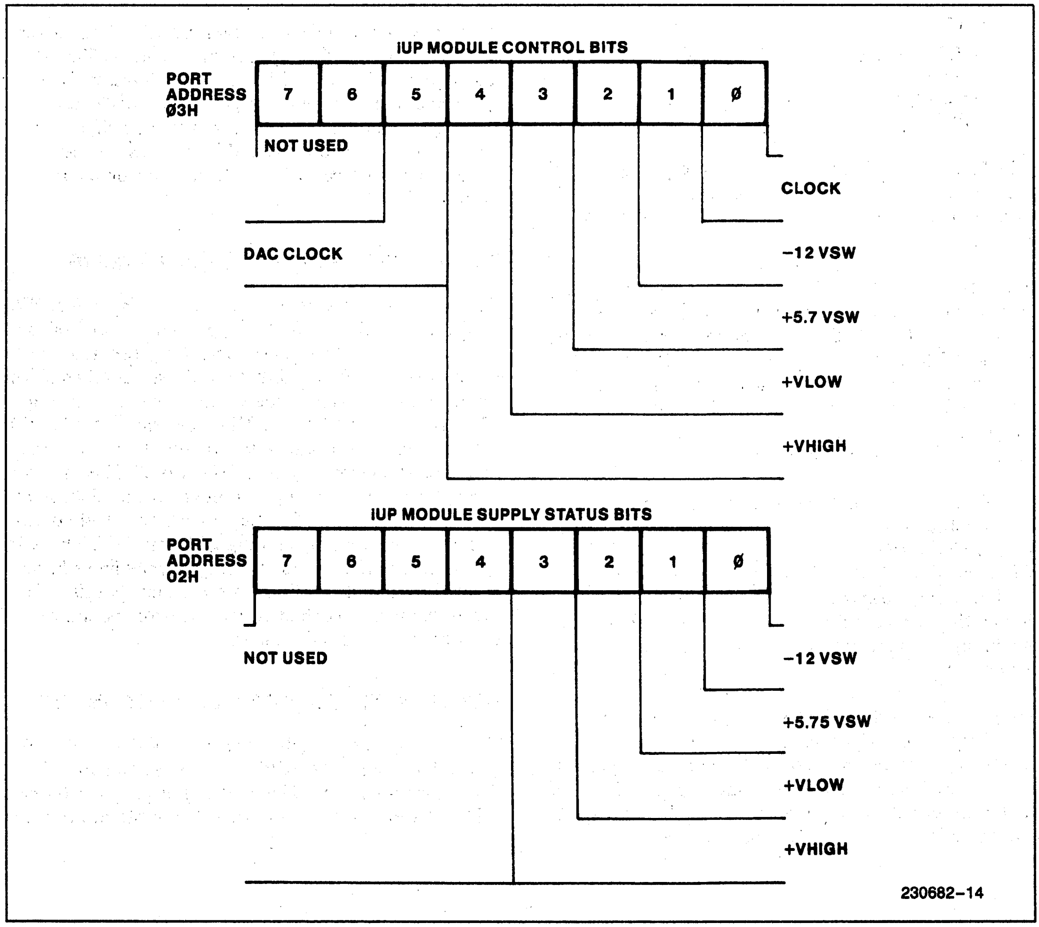
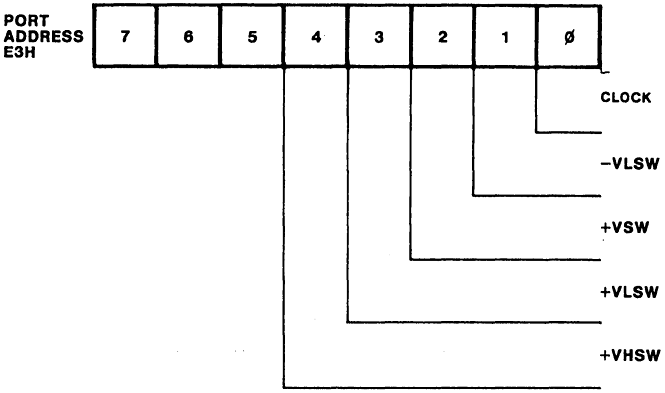


Figure 10. IUP Switched Voltage Control and Status Bit Definitions



230682-15

Figure 11. iPDS™ Switched Voltage Control Bit Definitions

I/O Port Address	I/O Write Active	I/O Read Active
10H		
11H		
12H		
13H		
14H		
15H		
16H		
17H		
18H		
19H		
1AH		
1BH		
1CH		
1DH		
1EH		
1FH		

Figure 12. Chart of iPDX Bus I/O Address Assignments