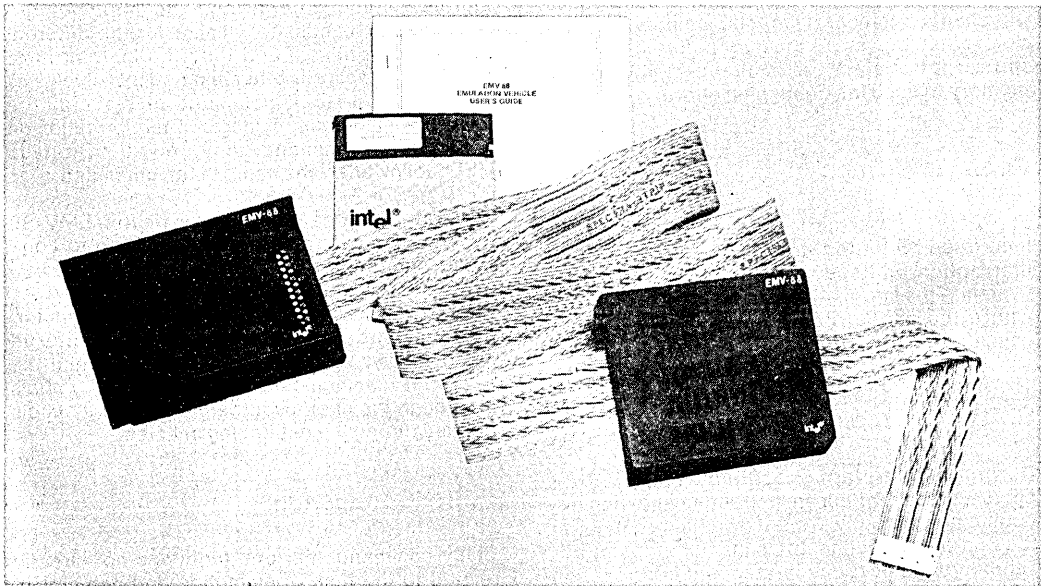




EMV-88 IAPX 8088 EMULATION VEHICLE

- 8 MHz in-circuit emulation
- Hosted on Intel's Personal Development System (iPDS™)
- Advanced easy-to-use features
 - Programmable function keys
 - Macros
 - Loop-control constructs
 - Instruction disassembly
- Help facility: EMV-88 command syntax reference at console
- Breakpoints
 - Three modes: execution, data access, or I/O access
 - One range breakpoint
 - Externally controlled breakpoints
 - Break-on-branch capability
- Full symbolic debugging
- 1 K byte real-time execution trace
- 4K bytes of on-board zero-wait-state mapped memory
- Software debugging with or without user system
- Includes Macro Assembler ASM 86/88, 8080/8085 to 8086/8088 assembly language source code conversion, LINK 86/88, and EMV-88 control software packages
- Fully supports 8088 RQ/QT, NMI, and Min and Max modes
- Supports PL/M 86/88

The EMV-88 system contains all the hardware, software, and documentation needed to interface to a user-designed iAPX-88 system and assists in the debugging and development of that system; in addition, the system can be used for testing in a manufacturing and/or service environment. The EMV-88 system consists of a buffer box and a controller that is hosted by an Intel Personal Development System (iPDS). The electrical and timing characteristics of the user's 8088 are emulated by the EMV-88 (see page 10 for timing comparisons). Software for the EMV-88 system allows the designer to emulate the user system's 8088 in real-time or single-step mode. Execution breakpoints stop emulation at user-specified conditions, and trace qualifiers control display of trace information. Program memory can be displayed and altered using ASM 86/87/88 mnemonics and symbolic references. Advanced emulator capabilities allow for programmable keys, command macros, and control constructs.



Intel Corporation Assumes No Responsibility for the Use of Any Circuitry Other Than Circuitry Embodied in an Intel Product. No Other Circuit Patent Licenses are Implied. Information Contained Herein Supercedes Previously Published Specifications On These Devices From Intel.

© INTEL CORPORATION, 1984

JUNE 1984
ORDER NUMBER:280021-001

FUNCTIONAL DESCRIPTION

The EMV-88 system provides fundamental capabilities for debugging an iAPX-88-based microsystem. These basic capabilities are described in the following sections.

Real-Time Breakpoint

The EMV-88 system allows a user system to execute user code at full clock speed (2 to 8 MHz) until a predefined breakpoint condition occurs. Breakpoints may be specified as a combination of four addresses or a combination of an address range and a single address. Breaks occur on execution addresses, read or write data addresses, read or write I/O port addresses, or on branch. Additionally, an externally supplied signal can cause a break. These break capabilities allow the user to stop the target system during the normal processing cycle and interrogate the state of the target system.

Real-Time Memory

The EMV-88 system supplies 4 Kbytes of high-speed RAM memory mappable on any even 4K-boundary within the 1 megabyte address space of the iAPX-88 microsystem. The RAM can be used to store the user program and make possible changes to user code. The memory can also be used as the user's memory before it exists in the target system, or in place of the user's memory to ease the debugging effort.

Real-Time Trace

The EMV-88 system maintains an active real-time trace buffer that tracks the last 1K byte of instruction addresses executed by the target system. This information can be used to discover where the user's program was before it broke emulation.

Software Break

During single-step execution, the EMV-88 system steps through an instruction and then executes a short software interrogation routine; at the end of the routine, the emulator stops or advances to the next single-step and interrogation cycle. This slow-down mode of emulation continues for a single instruction until a break condition is reached or a specified number of instruc-

tions has been executed. This type of emulation provides added execution control and microprocessor state information in exchange for real-time emulation.

Software Trace

Between single steps or after a real-time breakpoint, the EMV-88 system can automatically query the 8088 processor and optionally display up to four lines of information. This display can show execution address, disassembled code, current register values, or processor status information. Users can direct their display screens to present only desired information.

COMMANDS

The EMV-88 system has a friendly, easy-to-use human interface and commands that are well organized and easy-to-learn. Menu displays prompt and assist the user in learning the different commands. Figure 1 shows sample menu displays.

EMV-88 commands fall into four categories: utility commands, display/modify commands, emulation commands, and advanced commands. Once users understand the basic command categories, locating any command becomes simple. These categories, and many of the specific commands, are similar for the different emulation vehicles, and thus the learning time required to operate other emulation vehicles is reduced. The HELP command displays all the EMV-88 commands; if users want information on a particular command, they only need to type HELP followed by the name of the command.

Table 1 provides a summary of the EMV-88 system commands arranged according to command categories.

The EMV-88 system is a full symbolic emulator: all commands and displays can be entered symbolically. Thus the EMV-88 system and the user can communicate by referring to symbols defined in the user's source program or symbols defined during the debugging session.

Utility Commands

Utility commands perform functions not directly related to the task of emulation and debugging. These commands gain access to the IPDS system resources and display information about the emulator.

Display/Modify Commands

Display/modify commands change or display any register, port, or memory location addressable by the IAPX-88 target system. These commands provide access to specific areas of the processor or target system and thus minimize extraneous display information.

Emulation Commands

Commands that control program execution or initiate emulation fall into this category; for example, GO, BREAK, and DETRACE.

Advanced Commands

The advanced commands offer an easy way to increase debugging capability and create automated test sequences using the EMV-88. The advanced commands permit commands to be combined and saved as test routines. Tests can

be developed (with their own messages) that guide the user through a series of diagnostic and troubleshooting paths, making possible easy-to-use fault location down to the system or component level.

EMULATION MODES

The EMV-88 system offers three approaches to emulation: real-time emulation, single-step emulation, and branch-break emulation.

Programs with time-critical sections of code or critical interrupt routines can be emulated, traced, and debugged in real time. Real-time emulation supports specific execution breakpoints or range breakpoints. The real-time trace displays up to 1K byte of the most recently executed instruction addresses. The real-time emulation mode is entered by initiating emulation with the GO command. All break and trace commands that are associated with the GO command act in the EMV-88's real-time emulation mode.

```

HELP
HELP — Help is available for the following commands and definitions.
        Type "HELP <NAME>" where <name> is one of the following:
=====
UTILITY —   : BASE          EVALUATE  LOAD          SUFFIX
              : DEFINE         EXIT       REMOVE       SYMBOLS
              : DISABLE        HELP       RESET        TYPE
              : ENABLE         LIST       SAVE         WORKFILE
=====
DISPLAY —   : ASM           DREAL     POINTER      REGISTER
              : BYTE         DUMP      PORT         TREAL
              : DASH         MEMORY    REAL         WORD
=====
EMULATION — : BREAK         GO         PREVIOUS
              : DTRACE        MODE      STEP
=====
ADVANCED —  : COUNT         IF         MAP           WRITE
              : DIR          INCLUDE   PUT
              : FUNCTION     MACRO    REPEAT
=====
MISCELLANEOUS : ABOB7        DEMO      ENTERING     STRINGS
               : ABOB8        DISPLAY  EXPRESSIONS
               : CONTINUATION EDITING  NOTATION
=====
*

```

a) HELP Menu Display

Figure 1. Typical EMU-88 Menu Displays

```

*REG
-----
                                *REGISTER DISPLAY*

RAX=0031H      RAH=00H      RAL=31H
RBX=0000H      RBH=00H      RBL=00H
RCX=1234H      RCH=12H      RCL=34H
RDX=0000H      RDH=00H      RDL=00H

SP=0FFFFH     DP=1000H     SI=0000H     DI=0000H
CS=FFFFFH     DS=0000H     SS=0000H     ES=0000H

IP=0000H

RF=F002H      OF=0  DF=0  IFF=0  TF=0
               SF=0  ZF=0  AF=0  PF=0  CF=0
-----
*
    
```

b) REGISTER Display

```

*DTR
-----
                                *TRACE DISPLAY CONTROLS*
-----

TD= OFF        (instruction display, ON/OFF)
TR= OFF        (register display, ON/OFF)
TS= ON         (status display, ON/OFF)
-----

*DISPLAY START/STOP CONTROL*

TR0=00034H ON   TR1=00490H OFF   TR2= OFF   TR3=20035H ON
                ( enter both location and ON/OFF switch )

TV= OFF
                ( enter REGISTER VALUE SW )
-----
*
    
```

c) Menu Display for Setting Trace

Figure 1. Typical EMU-88 Menu Displays (continued)

Table 1. Summary of EMV-88 Commands and Command Categories

Commands	Description
<p>Utility Commands</p> <p>DEFINE DOMAIN ENABLE/DISABLE EVALUATE EXIT HELP INCLUDE LINE LIST LOAD MODULE REMOVE RESET SAVE SYMBOLS SUFFIX/BASE TYPE</p>	<p>Defines symbol or macro. Establishes default module. Control for expanded display. Evaluates any expression (numerical or logical). Terminates EMV-88 session. Displays command syntax. Loads a macro definition or a command file. Displays statement numbers and associated absolute addresses. Generates copy of emulation work session. Loads object file in mapped memory. Displays module names in EMV-88 module table. Deletes symbol or macro. Resets emulation processor. Saves memory to file. Displays symbols. Sets input and displays numeric base. Sets/displays data type to symbol name.</p>
<p>Emulation Commands</p> <p>BR BR0, 1, 2, 3 BRB BRR BV BC DTRACE GO MO PREVIOUS STEP TD TR TR0, 1, 2, 3 TS TV</p>	<p>Displays breakpoint menu. Breakpoint register for execution address. Breaks on branch. Breakpoint register for execution range. Breaks on value. Clears all breaks. Displays trace menu. Enters real-time emulation mode. Break qualifier. Displays execution trace. Enters slow-down emulation mode. Enables/disables display of code disassembly. Enables/disables display of registers. Enables/disables display by execution address. Enables/disables display of PSW. Enables/disables display by register value.</p>
<p>Display/Modify Commands</p> <p>ASM/DASM DUMP MEMORY PORT REGISTER BYTE WORD POINTER SINTEGER INTEGER REAL TREAL DREA</p>	<p>Changes/displays code memory in assembly language mnemonics. Displays memory as ASCII and hexadecimal. Displays menu for memory access. Changes/displays ports. Displays 8088 registers menu.</p> <p>Change/display memory.</p> <p>8087 commands.</p>

Table 1. Summary of EMV-88 Commands and Command Categories (continued)

Commands	Description
Advanced Commands	
DIR	Displays names of all available macros.
FUNCTION	Invokes macro assigned to function key.
MACRO	Displays macro text.
MAP	Sets/displays memory map.
PUT	Stores macro definitions.
WRITE	Evaluates and displays expressions and strings.
IF THEN	} Loop control constructs.
COUNT	
REPEAT	
WHILE	
UNTIL	

When full-speed emulation is not critical to the debugging effort, the EMV-88 system emulates one instruction, checks for a variety of break-point and trace-point conditions, displays trace information, and continues with another instruction. This slow-down mode of operation permits an enhanced non-real-time execution break and trace facility. The STEP command is used to enter the slow-down emulation mode.

A third mode, branch break, bridges the gap between real-time emulation and single-step emulation. During branch break, the EMV-88 system emulates in either real-time mode or single-step mode until any branch instruction is executed. After the branch has executed, the emulator breaks emulation. Thus this mode makes possible a fast and convenient mechanism for observing program flow.

INTENDED USE

The EMV-88 system is designed for debugging certain kinds of programs: it is particularly well suited to assist in debugging small- to medium-sized programs whose complexity is low to moderate in terms of interrupts, program nesting, and random execution flow. It is also designed for testing and troubleshooting in manufacturing and service environments.

Figure 2 shows the EMV-88 system and the IPDS in a debugging environment.

PHYSICAL DESCRIPTION

Hardware Components

The EMV-88 hardware consists of two parts: the controller and the buffer box. The controller contains all the logic to support mapped memory, break, trace, emulation, and communication with the host. The controller module is inserted in the side panel of the iPDS host development system and connects to the EMV/PROM programming adapter board, an option that must be present for EMV use. A five-foot cable connects the controller module to the buffer box. The buffer box buffers all iAPX-88 signals and contains the emulating 8088-2 processor and all logic to support both Min and Max modes of operation. The buffer box plugs into the user's 8088 socket by means of a short cable and supplies all iAPX-8088 signals to the target system.

Software Components

The EMV-88 software offers extensive emulation control of the target CPU. The software programs the controller, implements all emulator functions, displays information, and offers advanced features to further support debugging activities. Software is run on the iPDS host, and is packaged on a 5-1/4 inch diskette. An additional software diagnostic routine, included on the disk, thoroughly exercises the EMV-88 hardware.

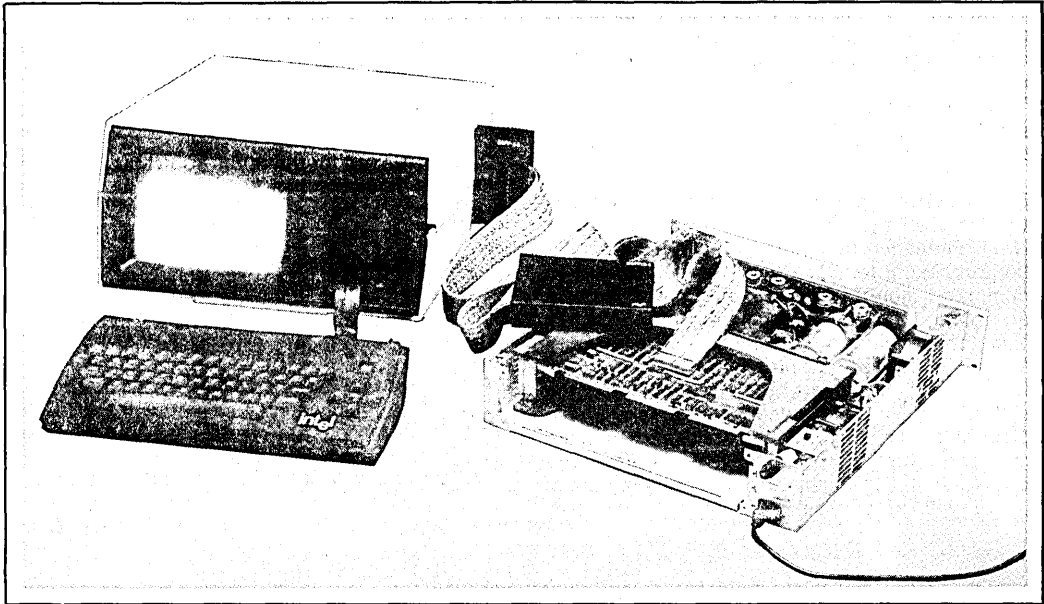


Figure 2. The EMV-88 System and the iPDS™ System in a Debugging Environment

The EMV-88 also includes the 8086/8088 Macro Assembler (ASM 86/88) and a software utility package. The macro assembler provides full macro capabilities, supports symbolic reference during code development and debugging, and supports modular code development and relocation features. The utility package includes the following software:

- CONV 86/88 to convert ASM 80/85 programs to ASM 86/88 programs
- LINK 86/88 to combine several ASM 86/88 object modules into a single object module
- LIB 86/88 to manage libraries of 8086/8088 object modules
- LOC 86/88 to locate relocatable modules to absolute executable addresses
- OH 86/88 to convert executable modules from object module format to hexadecimal format

Documentation

The EMV-88 system includes a documentation kit containing a comprehensive *EMV-88 Emulation Vehicle User's Guide* and an *EMV-88 Emulation Vehicle Pocket Reference*. In addition, extensive documentation is included for the Macro Assembler and the utilities software.

System Operation

A multitude of hardware and software interactions allow the EMV-88 to provide break, trace, map, and interrogation features. The iPDS host maintains symbol tables (from information passed to it from Intel language translators) and user symbols (defined during the debugging session). The symbol tables allow the EMV-88 to provide full symbolic debugging capabilities.

For the command monitor, the EMV-88 system requires 1K byte of user processor address space and six bytes of the user stack. The EMV-88 system supplies this 1K byte of memory and the necessary logic to map this memory anywhere in the user's decoded memory space on any 1K-byte boundary. During initialization, the EMV-88 system loads the command memory with the monitor software. This monitor routine has responsibility for interrogating the 8088 processor, changing any register or memory associated with the target system, and executing user code.

For example, during a memory display operation, the EMV-88 monitor causes the desired data in the memory to be passed to the iPDS host. The iPDS host takes the data, formats the data, associates symbolic information with the data, and displays the information for the user.

The address of the first byte of every executed target system instruction is stored in a 1K-byte circular trace buffer. During interrogation, this buffer can be read and displayed by the user. Trace information is passed back to the host where it is symbolically expanded, disassembled, and displayed.

Breakpoints are implemented by setting bits (associated with addresses) in a breakpoint RAM. Emulation begins by directing the emulation processor to execute user code rather than command-monitor code. Execution addresses are tracked through a queue emulator and, when a match between an execution address and a set bit in the breakpoint RAM is found, the EMV-88 system asserts the NMI line (after the execution of the instruction) to cause a break. Control is then passed to the software command monitor, which notifies the iPDS host of the break event. See Figure 3 for a block diagram of the EMV-88 hardware.

SPECIFICATIONS

EMV-88 Operating Requirements

The EMV-88 system operates with an iPDS system. The iPDS system must be equipped with the EMV/IUP adapter option (the iPDS-140).

EMV-88 System Specifications

Total access to 1M byte of IAPX-88 address space (except for 1K byte of EMV command memory)

Compatible with all Intel Series-II-based IAPX software

Can load a 4K-byte object file with symbols in less than one minute

Step mode operates at 5000:1 speed slow-down

Equipment Supplied

- EMV-88 emulator
- EMV-88 software and diagnostic diskette
- 8086/8088 Macro Assembler and utility package

Documentation Supplied

- EMV-88 Emulation Vehicle User's Guide*
- EMV-88 Emulation Vehicle Pocket Reference*

Documentation for the 8086/8088 Macro Assembler and utilities software

Emulation Clock Rate

- EMV-88-resident clock: 5 MHz
- User-supplied clock: 2 to 8 MHz

DC Characteristics

1. Output Low Voltage [$V_{OL}(\text{Max}) = 0.5 \text{ V}$]

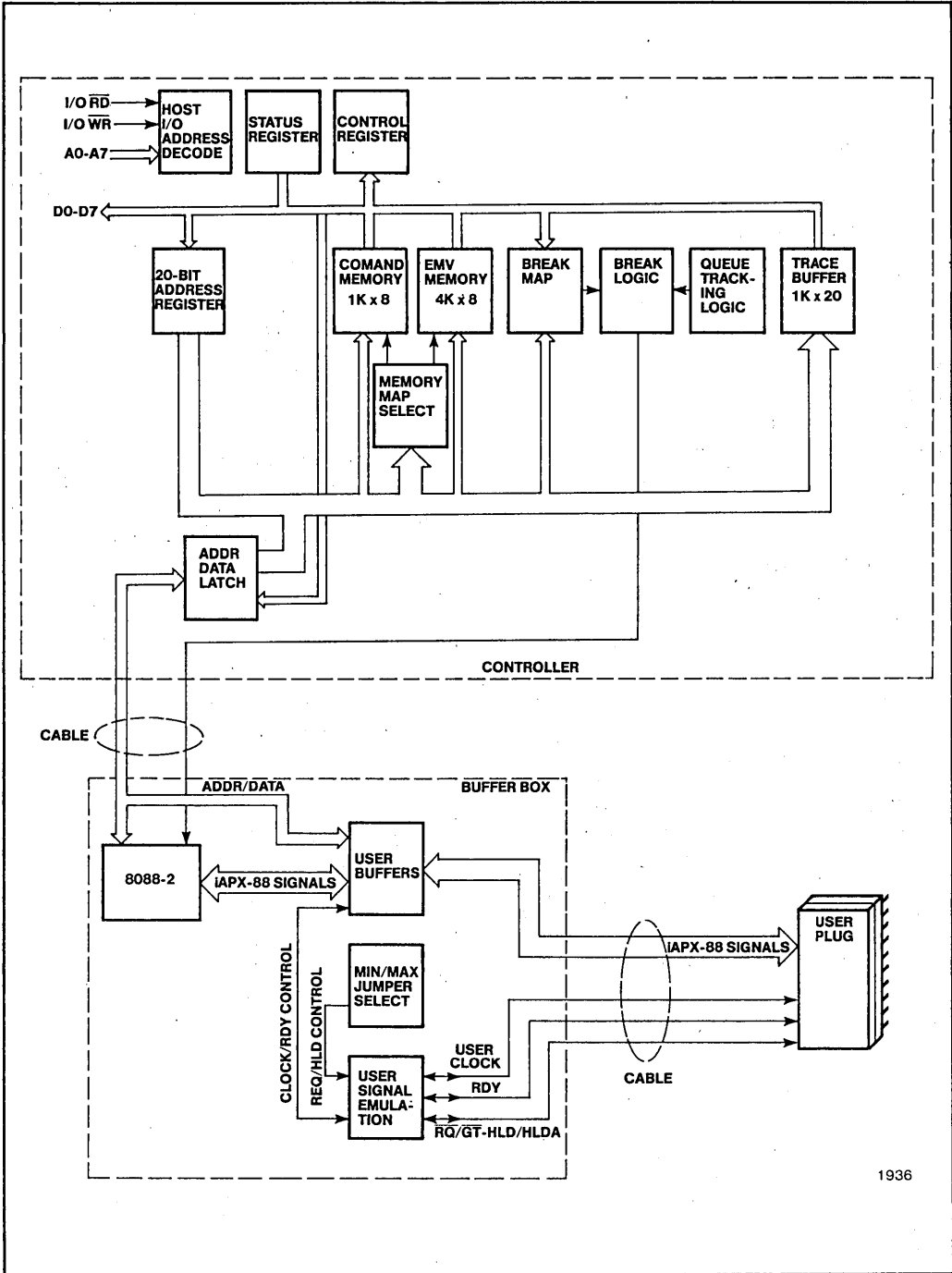
	$I_{OL}(\text{Min})$
AD0-AD7, A8-15	24 mA
A16/S3-A10/S6, \overline{RD} , \overline{LOCK} , QS0, QS1, $\overline{S0}$, $\overline{S1}$, $\overline{S2}$, \overline{WR} , IO/M, DT/ \overline{R} , \overline{DEN} , ALE, \overline{INTA}	24 mA
HLDA, $\overline{RQ}/\overline{GT}$	0.5 mA

2. Output High Voltage [$V_{OH}(\text{Min}) = 2.0 \text{ V}$]

	$I_{OH}(\text{Min})$
AD0-AD7, A8-15	-15 mA
A16/S3-A19/S6, $\overline{SS0}$, \overline{RD} , \overline{LOCK} , QS0, QS1, $\overline{S0}$, $\overline{S1}$, $\overline{S2}$, \overline{WR} , IO/M, DT/ \overline{R} , \overline{DEN} , ALE, \overline{INTA} , HLDA	-15 mA
$\overline{RQ}/\overline{GT}$	1.03 mA

3. Input Low Voltage [$V_{IL}(\text{Max}) = 0.8 \text{ V}$]

	$I_{IL}(\text{Max})$
AD0-AD7	-0.1 mA
NMI, CLK	-0.1 mA
READY	-0.4 mA
INTR, HOLD, \overline{TEST} , RESET	-2.0 mA
MN/ \overline{MX} (0.1 μf to GND)	-0.1 mA



1936

Figure 3. EMV-88 Block Diagram

4. Input High Voltage [$V_{IH}(\text{Min}) = 2.0 \text{ V}$]

	$I_{IH}(\text{Max})$
AD0-AD7	.02 mA
NMI, CLK	.02 mA
READY	.02 mA
INTR, HOLD, $\overline{\text{TEST}}$, RESET	0.4 mA
MN/ $\overline{\text{MX}}$.02 mA

Output Drive Capacity

The EMV-88 module has a greater output drive capacity than the 8088 chip, except for the RQ/GT output when $\overline{\text{GT}}$ is active.

EMV-88 User Cable

Capacitance: 22 pf/ft

Impedance: 105 ohms

Capacitive Loading

The EMV-88 module presents the user system with a maximum load of 34 pf and 0.8 mA.

All EMV-88 outputs are capable of driving a minimum of 15 pf and 20 mA while meeting all the probe's timing specifications. The EMV-88 module will drive larger capacitive loads but with possible performance degradation.

Timing Differences Between the 8088-2 Microprocessor and EMV-88 Emulation
MIN Mode

Symbol	Parameter	8088-2		EMV-88	
		Min ns	Max ns	Min ns	Max ns
TDVCL	Data in setup time	20		30	
TRYHCH	Ready setup time into 8088	68		86	
THVCH	HOLD setup time	20		7.5	
TINVCH	Setup time for recognition				
INTR		15		19	
NMI		15		78	
$\overline{\text{TEST}}$		15		77	
TCLAV	Address valid delay	10	60	24	74
TCHLAV	HLDA valid delay	10	100	18	118
TCHCTV	Control active delay	10	60	29	85
TCVCTV	Control active delay 1	10	70	22	80

MAX Mode

Symbol	Parameter	8088-2		EMV-88	
		Min ns	Max ns	Min ns	Max ns
TDVCL	Data in setup time	20		30	
TRYHCH	Ready setup time into 8088	68		86	
TINVCH	Setup time for recognition				
INTR		15	19		
NMI		15	78		
$\overline{\text{TEST}}$		15	77		
TCLAV	Address valid delay	10	60	24	74
TCLDV	Data valid delay	10	60	24	74
TCLRL	RD active delay	10	100	24	116
TGVCH	RQ/GT setup time	15		25	
TCHSV	Status active delay	10	60	24	70

Environmental Characteristics

Operating temperature: 50°-95° F (10°-35° C)

Operating humidity: 0-90% relative humidity,
non-condensingBuffer box: 7.25 in. x 1.5 in. x 5.75 in. (18.4
cm. x 3.8 cm. x 14.7 cm.)

Total weight: 2 lb. 6 oz. (1018 grams)

Physical CharacteristicsController: 7.75 in. x 1.5 in. x 5.75 in. (19.7
cm. x 3.8 cm. x 14.6 cm.)**Electrical Characteristics**Power required from the IPDS system: +5 V
±2.5% @ 2.5A (includes emulator requirements)

ORDERING INFORMATION

Part Number	Description
iPDS-EMV-88	Emulation vehicle for the iAPX-88 microsystem (includes diskette and documentation)