intel

# The 8086 Family User's Manual

October 1979

Additional copies of this manual or other Intel literature may be obtained from:

Literature Department
Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

The information in this document is subject to change without notice.

Intel Corporation makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Intel Corporation assumes no responsibility for any errors that may appear in this document. Intel Corporation makes no commitment to update nor to keep current the information contained in this document.

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.

Intel software products are copyrighted by and shall remain the property of Intel Corporation. Use, duplication or disclosure is subject to restrictions stated in Intel's software license, or as defined in ASPR 7-104.9(a)(9).

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of Intel Corporation.

The following are trademarks of Intel Corporation and may be used only to describe Intel products:

| | | |
|---|---|---|
| i | iSBC | Multimodule |
| ICE | Library Manager | PROMPT |
| iCS | MCS | Promware |
| Insite | Megachassis | RMX |
| Intel | Micromap | UPI |
| Intelevision | Multibus | µScope |
| Intellec | | |

and the combination of ICE, iCS, iSBC, MCS, or RMX and a numerical suffix.

# Table of Contents

## CHAPTER 3 THE 8089 INPUT/OUTPUT PROCESSOR

## CHAPTER 4 HARDWARE REFERENCE INFORMATION

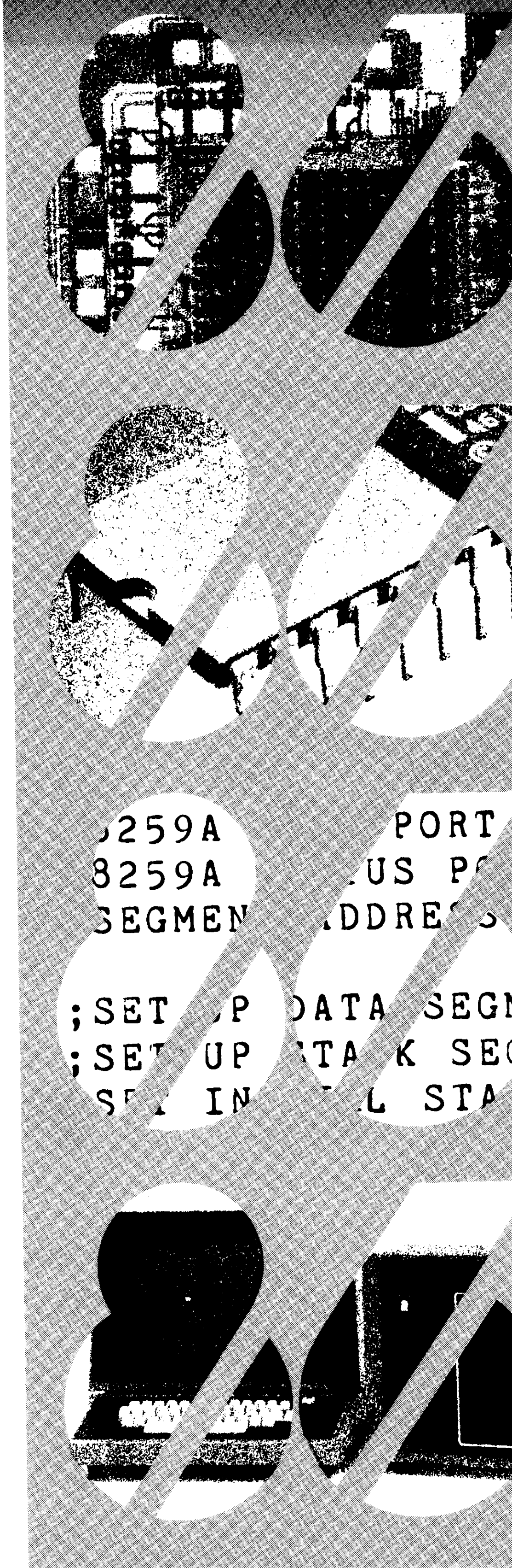


## APPENDIX A APPLICATION NOTES



## APPENDIX B DEVICE SPECIFICATIONS

# Chapter 1
# Introduction



# CHAPTER 1
# INTRODUCTION

This publication describes the Intel® 8086 family of microcomputing components, concentrating on the 8086, 8088 and 8089 microprocessors. It is written for hardware and software engineers and technicians who understand microcomputer operating principles. The manual is intended to introduce the product line and to serve as a reference during system design and implementation.

Recognizing that successful microcomputer-based products are judicious blends of hardware and software, the *User's Manual* addresses both subjects, although at different levels of detail. This publication is the definitive source for information describing the 8086 family components. Software topics, such as programming languages, utilities and examples, are given moderately detailed, but by no means complete, coverage. Additional references, available from Intel's Literature Department, are cited in the programming sections.

## 1.1 Manual Organization

The manual contains four chapters and three appendices. The remainder of this chapter describes the architecture of the 8086 family, and subsequent chapters cover the individual components in detail.

Chapter 2 describes the 8086 and 8088 Central Processing Units, and Chapter 3 covers the 8089 Input/Output Processor. These two chapters are identically organized and focus on providing a *functional* description of the 8086, 8088 and 8089, plus related Intel hardware and software products. Hardware reference information—electrical characteristics, timing and physical interfacing considerations—for all three processors is concentrated in Chapter 4.

Appendix A is a collection of 8086 family application notes; these provide design and debugging examples. Appendix B contains complete data sheets for all the 8086 family components and system development aids; summary data sheets covering compatible components from other Intel product lines are also reproduced in Appendix B.

## 1.2 8086 Family Architecture

Considered individually, the 8086, 8088 and 8089 are advanced third-generation microprocessors. Moreover, these processors are elements of a larger design, that of the 8086 family. This systems architecture specifies how the processors and other components relate to each other, and is the key to the exceptional versatility of these products.

The components in the 8086 family have been designed to operate together in diverse combinations within the systematic framework of the overall family architecture. In this way a single family of components can be used to solve a wide array of microcomputing problems. A component mix can be tailored to fit the performance needs of an application precisely, without having to pay for unneeded capabilities that may be bundled into more monolithic, CPU-centered architectures. Using the same family of components across multiple systems limits the learning curve problem and builds on past experience. Finally, the modular structure of the family architecture provides an orderly way for systems to grow and change.

The 8086 family architecture is characterized by three major principles:

1. System functions are distributed among specialized components.

2. Multiprocessing capabilities are inherent in the hardware.

3. A hierarchical bus organization provides for the complex data flows required by high-performance systems without burdening simpler systems with unneeded capabilities.

### Functional Distribution

Table 1-1 lists the components that constitute the 8086 microprocessor family. All components are contained in standard dual in-line packages and require single +5V power sources.

Table 1-1. 8086 Component Family

| Microprocessor | | Technology | Pins | Description |
|---|---|---|---|---|
| 8086 | Central Processing Unit (CPU) | HMOS | 40 | 8/16 bit general-purpose microprocessor; 16-bit external data path. |
| 8088 | Central Processing Unit (CPU) | HMOS | 40 | 8/16 bit general-purpose microprocessor; 8-bit external data path. |
| 8089 | Input/Output Processor (IOP) | HMOS | 40 | 8/16 bit microprocessor optimized for high-speed I/O operations; 8-bit and 16-bit external data paths. |

| Support Component | | Technology | Pins | Function |
|---|---|---|---|---|
| 8259A | Programmable Interrupt Controller (PIC) | NMOS | 28 | Identifies highest-priority interrupt request. |
| 8282<br>8283 | Octal Latch<br>Octal Latch (Inverting) | Bipolar | 20 | Demultiplexes and increases drive of address bus. |
| 8284 | Clock Generator and Driver | Bipolar | 18 | Provides time base. |
| 8286<br>8287 | Octal Bus Transceiver<br>Octal Bus Transceiver (Inverting) | Bipolar | 20 | Increases drive on data bus. |
| 8288 | Bus Controller | Bipolar | 20 | Generates bus command signals. |
| 8289 | Bus Arbiter | Bipolar | 20 | Controls access of microprocessors to multimaster system bus. |

## Microprocessors

At the core of the product line are three microprocessors that share these characteristics:

- Standard operating speed is 5 MHz (200 ns cycle time); a selected 8 MHz version of the 8086 CPU is also available.
- Chips are housed in reliable 40-pin packages.
- Processors operate on both 8- and 16-bit data types; internal data paths are at least 16 bits wide.
- Up to 1 megabyte of memory can be addressed, along with a separate 64k byte I/O space.
- The address/data and status interfaces of the processors are compatible (the address and data buses are time-multiplexed at the processor, i.e., an address transmission is followed by a data transmission over a subset of the same physical lines).

The 8086 and 8088 are third-generation central processing units (CPUs) that differ primarily in their external data paths. The 8088 transfers data between itself and other system components 8 bits at a time. The 8086 can transfer either 8 or 16 bits in one bus cycle and is therefore capable of greater throughput. Both processors have two operating modes, selectable by a strapping pin. In minimum mode, the CPUs emit the bus control signals needed by memory and I/O peripheral components. In maximum mode, an 8288 Bus Controller assumes responsibility for controlling devices attached to the system bus. CPU pins no longer needed for bus control are then redefined to provide signals that support multiprocessing systems.

The 8089 Input/Output Processor (IOP) is an independent microprocessor whose design has been optimized for transferring data. The 8089 typically runs under the direction of a CPU, but it executes a separate instruction stream and can operate in parallel with other system processors. The IOP contains two independent I/O channels that combine attributes of both CPUs and advanced DMA (direct memory access) controllers. The channels can execute programs and perform programmed I/O operations similar to CPUs. They may also transfer data by DMA, at rates up to 1.25 megabytes per second (5 MHz version). The channels can support mixes of 8- and 16-bit I/O devices and memory. Combining speed with programmable intelligence, the 8089 can assume the bulk of I/O processing overhead and thereby free a CPU to perform other tasks.

### Interrupt Controller

The 8259A Programmable Interrupt Controller (PIC) is a new, 8086 family-compatible version of the familiar 8259 that has been enhanced to operate with the advanced interrupt facilities of the 8086 and 8088 CPUs. The 8259A accepts interrupt requests from up to eight sources; up to 64 sources may be accommodated by "cascading" additional 8259As. Each interrupt source is assigned a priority number that typically reflects its "criticality" in the system. The 8259A has several built-in, priority-resolving mechanisms that are selectable by software commands from the CPU. These modes operate somewhat differently, but in general the 8259A continuously identifies the highest-priority active interrupt request and generates an interrupt request to the CPU if this request has higher priority than the request currently being processed. When the CPU recognizes the interrupt request, the 8259A transfers a code to the CPU that identifies the interrupt source.

### Bus Interface Components

Components may be selected from this modular group to implement different system bus configurations. Except for the 8284, all components are optional; their inclusion in a system is based on the needs of the application. All of the bus interface components are implemented using bipolar technology to provide high-quality, high-drive signals and very fast internal switching.

The 8284 Clock Generator and Driver provides the time base for the 8086 family microprocessors. It divides the frequency signal from an external crystal or TTL signal by three and outputs the 5 MHz or 8 MHz processor clock signal. It also provides the microprocessors with reset and ready signals.

8282 or 8283 Octal Latches may be added to a system to demultiplex the combined address/data bus generated by the 8086 family microprocessors. A demultiplexed bus provides separate stable address and data lines required by many peripheral components. Two latches demultiplex 16 bits of the bus to provide an address space of up to 64k bytes, while three latches generate the full 20-bit (megabyte) address space. The latches also provide the high drive on the address lines needed in larger systems.

8286 and 8287 Octal Bus Transceivers are used to provide more drive on data lines than the processors themselves are capable of providing. One or two transceivers may be used depending on the width of the data bus (8 or 16 bits).

The 8288 Bus Controller decodes status signals output by an 8089, or a maximum mode 8086 or 8088. When these signals indicate that the processor is to run a bus cycle, the 8288 issues a bus command that identifies the bus cycle as memory read, memory write, I/O read, I/O write, etc. It also provides a signal that strobes the address into 8282/83 latches. The 8288 provides the drive levels needed for the bus control lines in medium to large systems.

The 8289 Bus Arbiter controls the access of a processor to a multimaster system bus. A multimaster bus is a path to system resources (typically memory) that is shared by two or more microprocessors (masters). Arbiters for each master may use one of several priority-resolving techniques to ensure that only one master drives the shared bus.

## Multiprocessing

Employing multiple processors in medium to large systems offers several significant advantages over the centralized approach that relies on a single CPU and extremely fast memory:

- system tasks may be allocated to special-purpose processors whose designs are optimized to perform certain types of tasks simply and efficiently;

- very high levels of performance can be attained when multiple processors can execute simultaneously (parallel processing);
- robustness can be improved by isolating system functions so that a failure or error in one part of the system has a limited effect on the rest of the system;
- the natural partitioning of the system promotes parallel development of subsystems, breaks the application into smaller, more manageable tasks, and helps isolate the effects of system modifications.

The 8086 family architecture is explicitly designed to simplify the development of multiple processor systems by providing facilities for coordinating the interaction of the processors.

The architecture supports two types of processors: independent processors and coprocessors. An independent processor is one that executes its own instruction stream. The 8086, 8088 and 8089 are examples of independent processors. An 8086 or 8088 typically executes a program in response to an interrupt. The 8089 starts its channels in response to an interrupt-like signal called a channel attention; this signal is typically issued by a CPU.

The 8086 architecture also supports a second type of processor, called a coprocessor. Coprocessor "hooks" have been designed into the 8086 and 8088 so that this type of processor can be accommodated in the future. A coprocessor differs from an independent processor in that it obtains its instructions from another processor, called a host. The coprocessor monitors instructions fetched by the host and recognizes certain of these as its own and executes them. A coprocessor, in effect, extends the instruction set of its host processor.

The 8086 family architecture provides built-in solutions to two classic multiprocessing coordination problems: bus arbitration and mutual exclusion. Bus arbitration may be performed by the bus request/grant logic contained in each of the processors, by 8289 Bus Arbiters, or by a combination of the two when processors have access to multiple shared buses. In all cases, the arbitration mechanism operates invisibly to software.

For mutual exclusion, each processor has a LOCK (bus lock) signal which a program may activate to prevent other processors from obtaining a shared system bus. The 8089 may lock the bus during a DMA transfer to ensure that both the transfer completes in the shortest possible time and that another processor does not access the target of the transfer (e.g., a buffer) while it is being updated. Each of the processors has an instruction that examines and updates a memory byte with the bus locked. This instruction can be used to implement a semaphore mechanism for controlling the access of multiple processors to shared resources. (A semaphore is a variable that indicates whether a resource, such as a buffer or a pointer, is "available" or "in use"; section 2.5 discusses semaphores in more detail).

## Bus Organization

Figure 1-1 summarizes the 8086 family bus structure. There are two different types of buses: system and local. Both buses may be shared by multiple processors, i.e., both are multimaster buses. Microprocessors are always connected to a local bus, and memory and I/O components usually reside on a system bus. The 8086 family bus interface components link a local bus to a system bus.

### Local Bus

The local bus is optimized for use by the 8086 family microprocessors. Since standard memory and I/O components are not attached to the local bus, information can be multiplexed and encoded to make very efficient use of processor pins (certain MCS-85™ peripheral components can be directly connected to the local bus). This allows several pins to be dedicated to coordinating the activity of multiple processors sharing the local bus. Multiple processors connected to the same local bus are said to be local to each other; processors on different local buses are said to be remote to each other, or configured remotely. Both independent processors and coprocessors may share a local bus; on-chip arbitration logic determines which processor drives the bus. Because the processors on the local bus share the same bus interface components, the local configuration of multiple processors provides a compact and inexpensive multiprocessing system.

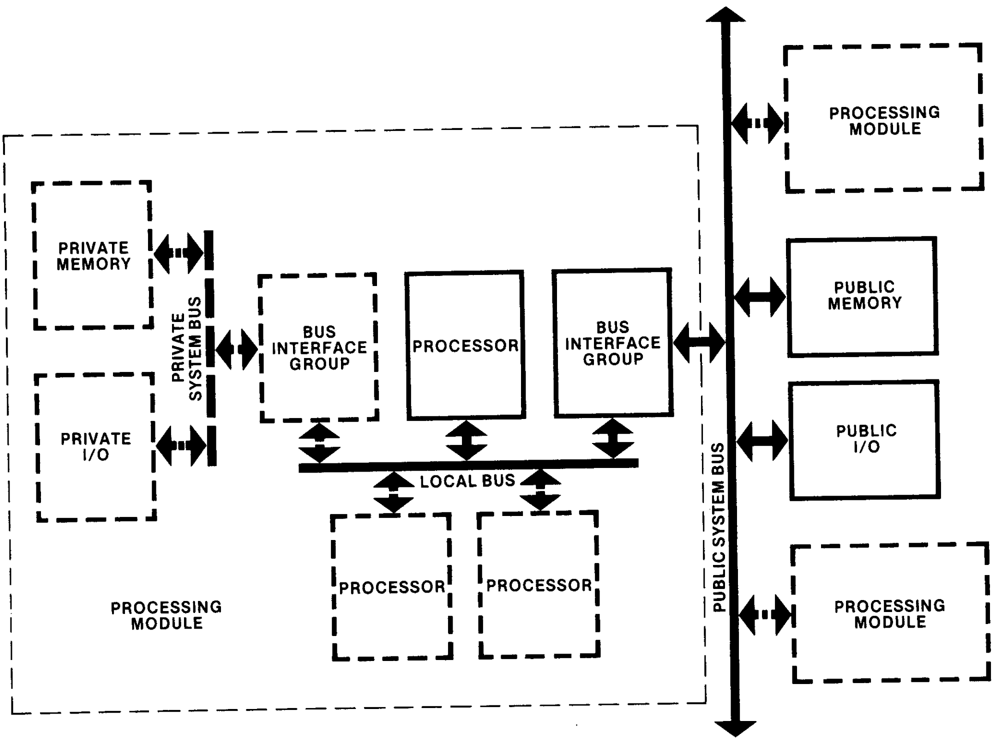



Figure 1-1. Generalized 8086 Family Bus Structure

## System Bus

A full implementation of an 8086 system bus consists of the following five sets of signals:

1. address bus,
2. data bus,
3. control lines,
4. interrupt lines, and
5. arbitration lines.

These signals are designed to meet the needs of standard memory and I/O devices; the address and data buses are demultiplexed and traditional control signals (memory read/write, I/O read/write, etc.) are provided on the system bus.

The system bus design is modular and subsets may be implemented according to the needs of the application. For example, the arbitration lines are not needed in single-processor systems or in multiple-processor systems that perform arbitration at the local-bus level.

A group of bus interface components transforms the signals of a local bus into a system bus. The number of bus interface components required to generate a system bus depends on the size and complexity of the system; reduced application needs translate directly into reduced component counts. These main variables determine the configuration of a bus interface group: address space size (number of latches), data bus width (number of transceivers), and arbitration needs (presence of a bus arbiter).

The 8086 family system bus is functionally and electrically compatible with the Multibus™ multimaster system bus used in Intel's iSBC™ line of single board computing products. This compatability gives system designers access to a wide variety of computer, memory, communications and other modules that may be incorporated into products, used for evaluation or for test vehicles.

## Processing Modules

The processor(s) and bus interface group(s) that are connected by a local bus constitute a processing module. A simple processing module could consist of a single CPU and one bus interface group. A more complex module would contain multiple processors, such as two IOPs, or a CPU and one or two IOPs. One bus interface group typically links the processors in the module to a public system bus. If there are multiple processing modules in the system, all memory or I/O connected to the public bus is accessible to all processing modules on the public bus. 8289 Bus Arbiters in each processing module control the access of the modules to the public bus and hence to the public memory and I/O.

A second bus interface group may be connected to a processing module's local bus, generating a second bus. This bus can provide the processing module with a private address space that is not accessible to other processing modules. Distributing memory and I/O resources in this manner can improve system robustness by isolating the effects of failures. It can also increase system throughput dramatically. If processor programs and local data are placed in private memory, contention for use of the public system bus can be held to a minimum to ensure that shared resources are quickly available when they are needed. In addition, processors in separate modules can simultaneously fetch instructions from private memory spaces to allow multiple system tasks to proceed in parallel.

## Bus Implementation Examples

This section summarizes the 8086 family bus organization by showing how components from the family can be combined to implement diverse bus configurations. The first two examples illustrate special cases that extend the applicability of the 8086 family to smaller systems. The remaining examples add and recombine the same basic components to form progressively more complex bus configurations. Note that these examples are intended to be illustrative rather than exhaustive; many different combinations of components can be tailored to fit the needs of individual applications.

In its minimum mode configuration, the 8088 time-multiplexes its 8-bit data bus with the lower eight bits of its 20-bit address bus (figure 1-2). This multiplexed address/data bus, and the bus control signals emitted by the 8088, are directly compatible with the multiplexed bus components of Intel's 8085 family. These peripherals contain on-chip logic that demultiplexes a combined address/data bus. In addition, many of these devices are multifunctional, combining, for example, RAM, I/O ports and a timer on a single chip. By using these components, it is possible to build small (as few as four chips) economical systems that are nonetheless capable of performing significant computing tasks.

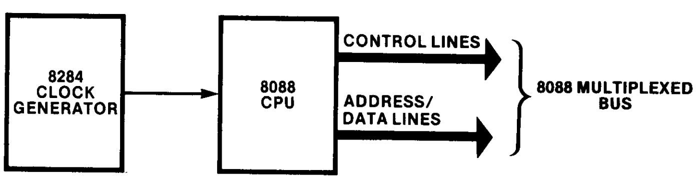



Figure 1-2. 8088 Multiplexed Bus

Combining 8282/83 latches with a minimum mode 8086 or 8088 produces a minimum mode system bus (figure 1-3). Two latches provide an address space of up to 64k bytes; adding a third latch provides access to the full megabyte of memory. An 8288 Bus Controller is not required for this implementation as the CPUs themselves emit the bus control signals when they are configured in the minimum mode. This demultiplexed bus structure is compatible with the wide array of memory and I/O components that have been developed for the industry-standard 8080A CPU. Eight-bit peripherals may be connected to both the upper and lower halves of the 8086's 16-bit data bus. 8286/87 transceivers may be added to provide additional drive on the data lines, where required. Including an 8259A gives the CPU the ability to respond to multiple interrupt sources without polling. The minimum mode system bus configuration is well-suited to a variety of systems whose computational requirements can be met by a single 8086 or 8088 CPU.

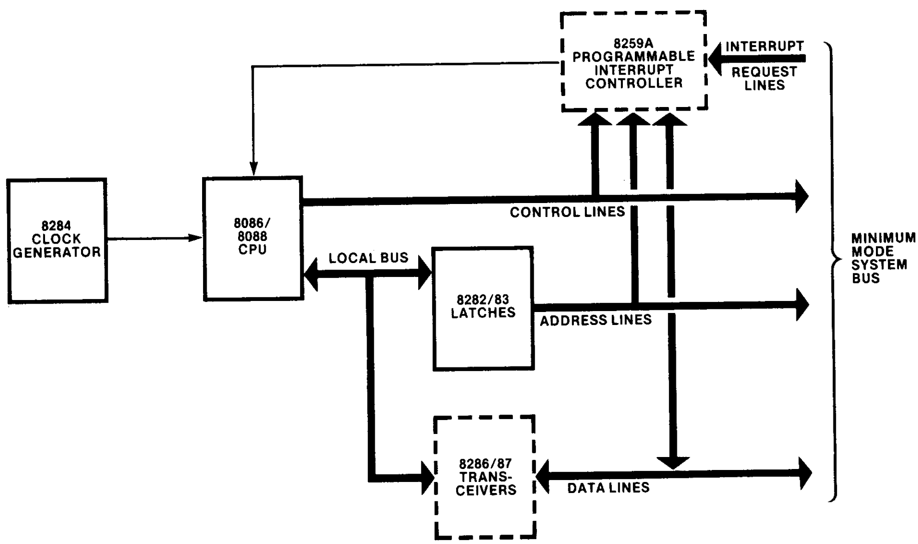



Figure 1-3. Minimum Mode System Bus

When an 8086 or 8088 is configured in maximum mode and an 8288 is added to control the system bus, one or two 8089s may be directly connected to the CPU (figure 1-4). The processors all share the same latches, transceivers, clock and bus controller, via the local bus. Arbitration logic built into the 8086, 8088 and 8089 coordinates use of the local bus, and thus of the system bus. This bus configuration enables the powerful I/O handling capabilities of the 8089 to be incorporated into systems of moderate size and cost.

The 8289 enables high-performance systems to be designed as a series of independent processing modules whose activities are coordinated via a shared system bus. Figure 1-5 shows the multimaster system bus interface; this bus structure is electrically compatible with the Multibus™ architecture used in Intel iSBC™ single-board computing systems.

Several different combinations of processors may be attached to the local bus of a multimaster computing module:

- a single 8086 or 8088
- a single 8089
- two 8089s
- an 8086 or 8088 and one 8089
- an 8086 or 8088 and two 8089s

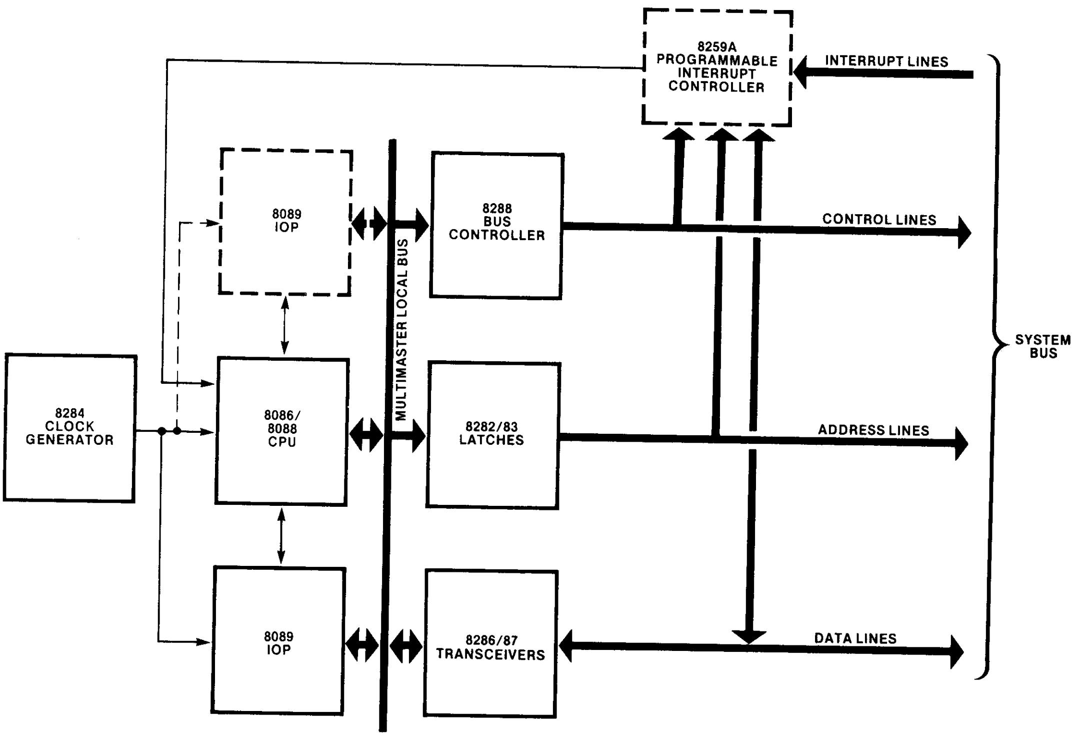



Figure 1-4. Multimaster Local Bus

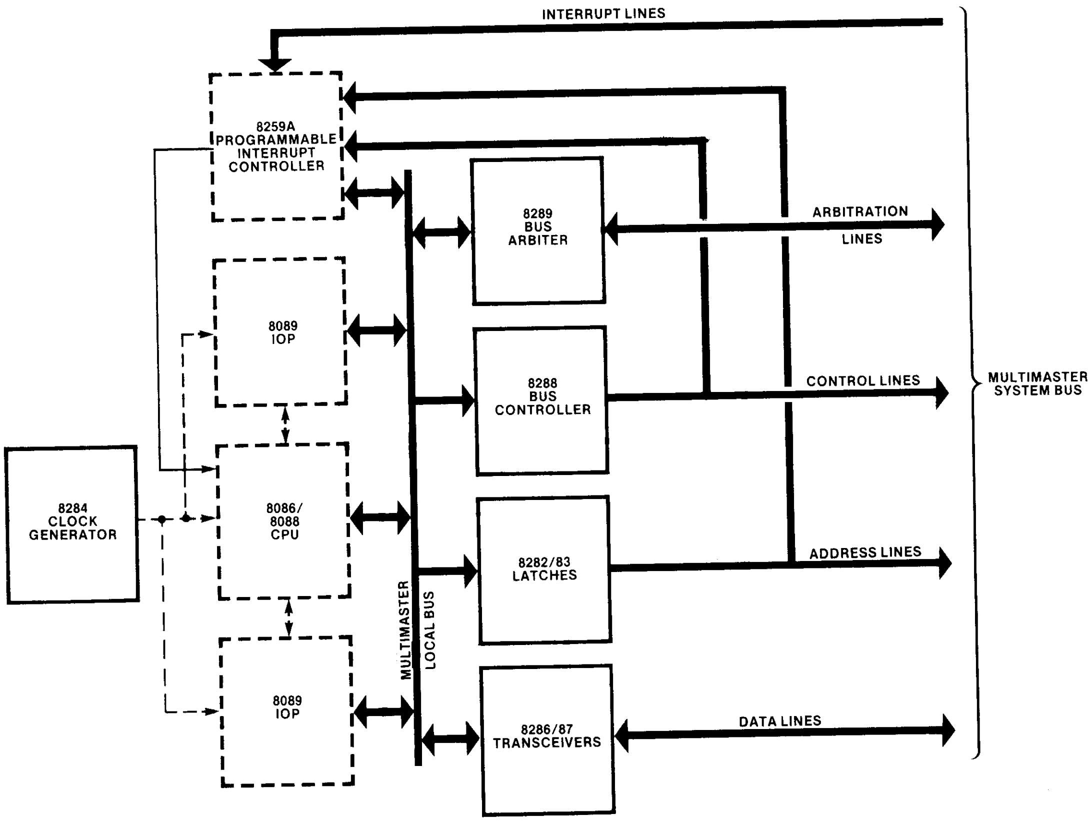



**Figure 1-5. Basic Multimaster Processing Module**

All of the processors on the local bus obtain access to the system bus through a single set of interface components.

One or two 8089s in a multimaster processing module may be configured with a private I/O bus as shown in figure 1-6. In this configuration, memory access commands are directed to the public multimaster system bus, while I/O commands use the private I/O bus. Memory, containing the 8089's programs, as well as I/O devices, may be connected to the private I/O bus. Taking this approach can greatly reduce the 8089's use of the system bus as most memory and I/O accesses can be made to the private address space. The system bus is thus made available for use by other processors, and the 8089 can execute in parallel with other processors for extended periods. A limited private I/O bus may be implemented using the 8-bit multiplexed peripherals of the 8085 family, eliminating the latches and transceivers shown in figure 1-6.

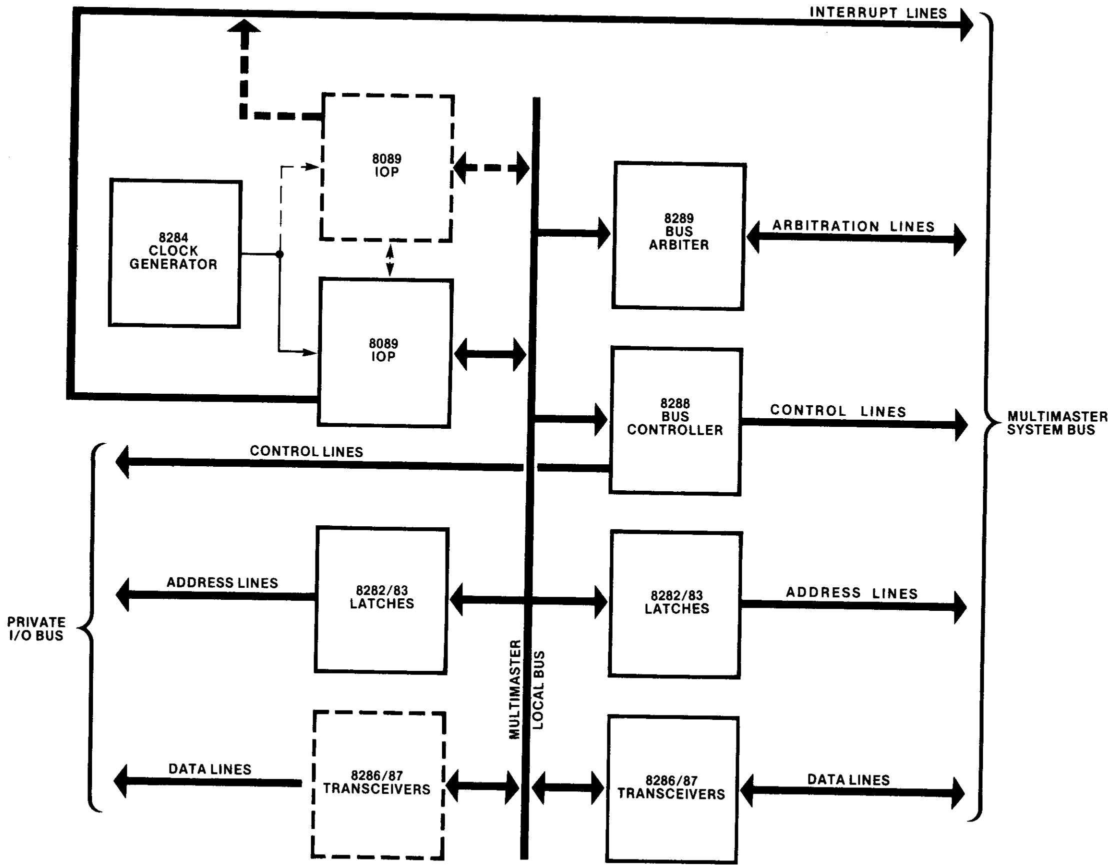



Figure 1-6. Private I/O Bus

Adding a second 8288 to the local bus allows an 8086 or 8088 in a processing module to divide its address space into system and resident sections (figure 1-7). A PROM or decoder is used to direct an address reference to the system bus or to the resident bus. The resident bus allows the CPU to run out of its own address space to minimize its use of the system bus. Since no other processors can access the private memory on the CPU's resident bus, operating system code and data in this space is protected from errors in other processor programs. If a second 8289 is added to a resident bus module, the resident bus becomes a second multimaster system bus.

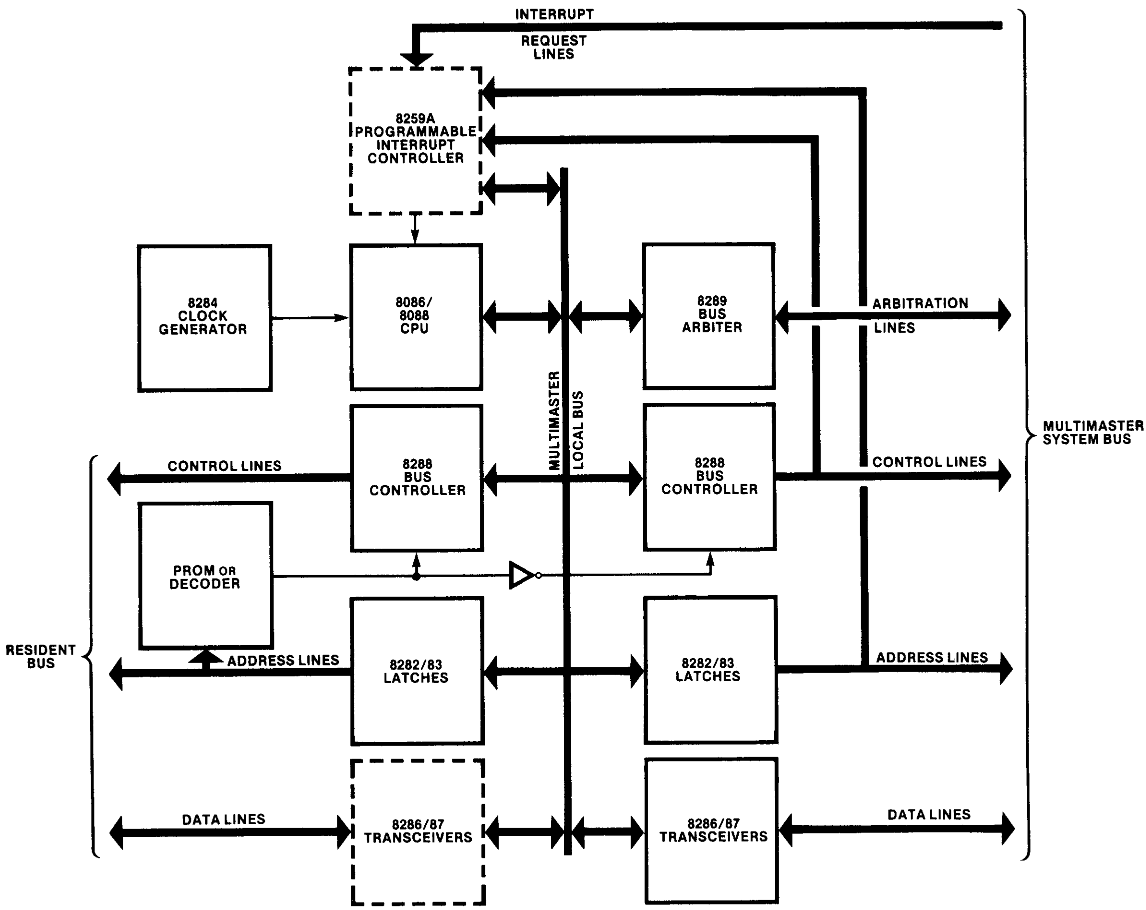



**Figure 1-7. Resident Bus**

As an alternative to the resident bus, a private read-only memory space can be implemented using the RD (read) signal provided by the CPUs in lieu of an 8288 Bus Controller.

Multiprocessing systems of widely varying complexity can be constructed from multimaster processing modules. Each module can be designed and implemented separately and can be optimized to perform a given task. The modules can communicate with each other by means of interrupts and messages placed in system memory. Additional functions can be added to a system by incorporating the new functions into modules and connecting the modules to the system bus.

Figure 1-8 illustrates a hypothetical system in which nine processors are distributed among five multimaster processing modules. (For clarity, bus interface components are not shown in figure 1-8.) A supervisor module controls the system, primarily responding to interrupts and dispatching other modules to perform tasks. The supervisor CPU, like the other processors in the system, executes code from private memory that is inaccessible to other modules. System memory, which is accessible to all the processors, is used only for messages, common buffers, etc. This helps to "protect" the processors from each other and to keep system bus contention at a minimum. The database module is responsible for maintaining all system files. Each of the three graphics modules supports a graphics CRT terminal. An 8089 in each module performs data transfers and CRT refresh and calls upon an 8088 for intensive computational routines.

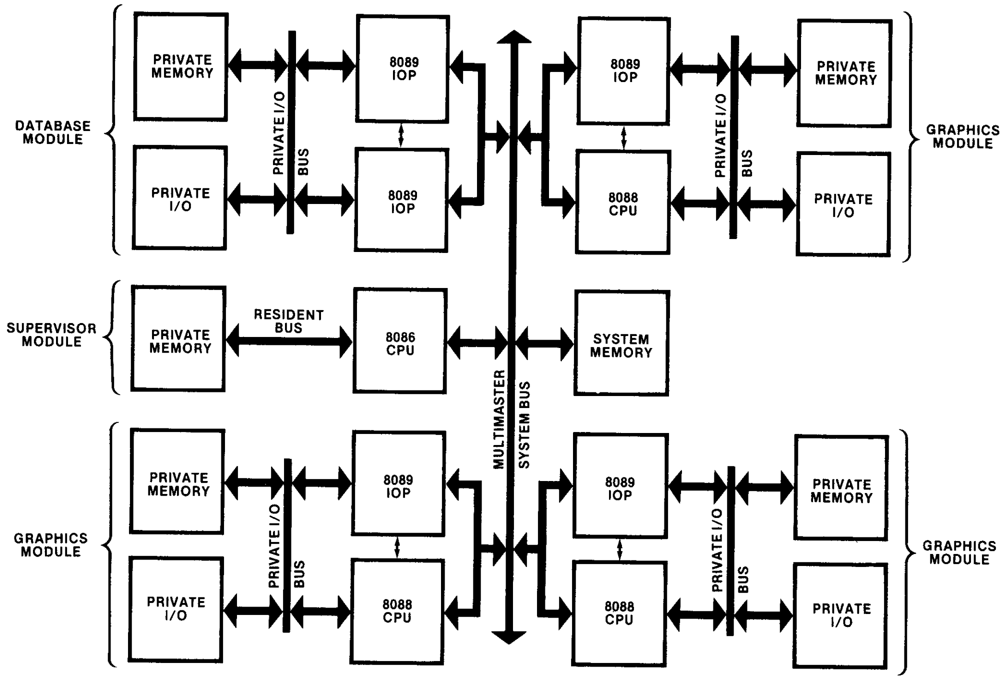



Figure 1-8. Multimaster Design Example

## 1.3 Development Aids

Intel provides the sophisticated tools needed for timely and economical development of products based on the 8086 family. The 8086 family system development environment is focused on the Intellec® Series II Microcomputer Development System (figure 1-9). The Intellec system is a multiple-microprocessor system that runs ISIS-II, a disk-based operating system that has been proven in thousands of installations. The Intellec has built-in interfaces for a printer, a PROM programmer and a paper tape reader/punch. This same hardware and operating system may be used to develop systems based on other Intel microprocessor families such as the 8085 and the 8048.

Three language translators support 8086 family programming. PL/M-86 is a high-level language for the 8086 and 8088 that supports structured programming techniques. It is upward-compatible with PL/M-80, the most widely used high-level microprocessor language. ASM-86 may be used to write assembly language programs for the 8086 and the 8088 CPUs and gives the programmer access to the full power of these CPUs. 8089 programs are written in ASM-89, the 8089 assembly language.

The language translators produce compatible outputs that can be manipulated by the software development utilities. LINK-86, for example, can combine programs written in ASM-86 with PL/M-86 programs. LIB-86 allows related programs to be stored in libraries to simplify storage and retrival. LOC-86 assigns absolute memory addresses to programs. OH-86 changes the format of an executable program for PROM programming or for loading into the RAM of a test vehicle.

The UPP-301 Universal PROM Programmer can burn programs into any of Intel's PROM memories; the UPP plugs into the Intellec® system and allows program data to be manipulated from the console before it is programmed into the PROM.

The SDK-86 is an (minimum mode) 8086-based prototyping and evaluation kit. It includes the CPU, RAM, I/O ports and a breadboard area for interfacing customer circuits. A ROM-based monitor program is supplied with the kit. Monitor commands may be entered from an on-board keypad or from a terminal; the monitor returns results to the SDK-86's on-board LED display or to a terminal. Monitor commands allow programs to be entered, run, stopped, and single-stepped; memory contents can be altered as well as displayed. The SDK-C86 Software and Cable Interface connects an SDK-86 to an Intellec® system. The software supplied with the cable enables programs to be transferred between the development system and the SDK-86 to allow users to develop programs using the text editor, translators and utilities of the Intellec system and then download the program to the SDK-86 for execution.

The iSBC 86/12™ board is a high-performance single board computer based on a maximum mode 8086 CPU. The board contains 32k of dual-port RAM that is accessible to the CPU via the on-board bus and to other processors via the built-in Multibus™ interface. The board also has an asynchronous serial port, parallel ports with sockets for drivers and terminators, two timers and sockets for 16k of ROM.

An iSBC 86/12™ can be linked to an Intellec® system using the iSBC 957™ Intellec-iSBC 86/12 Interface and Execution Package. The package includes a ROM-based monitor for the iSBC 86/12 board, software for the Intellec system and cabling to connect the two. The package supports data transfers between Intellec diskettes and iSBC 86/12 memory, full speed execution of customer programs on the iSBC 86/12 board, breakpoints, single-stepping, and data moves, replacements, searches and compares. All commands are entered from the Intellec console.

The ICE-86™ module is an in-circuit emulator for the 8086 microprocessor. A 40-pin probe replaces the 8086 in the system under test. This probe is connected to ICE-86 circuit boards that in turn plug into the Intellec® chassis. The ICE-86 module emulates the 8086 in the system under test in response to commands entered through the Intellec console. These commands allow the user to debug the system by setting breakpoints, tracing the flow of execution, single-stepping, examining and altering memory and I/O, etc. All references to program variables and labels are symbolic (i.e., their PL/M-86 or ASM-86 names). Software testing can also map "system under test" memory into the Intellec memory to permit software testing to begin before prototype hardware has been developed.

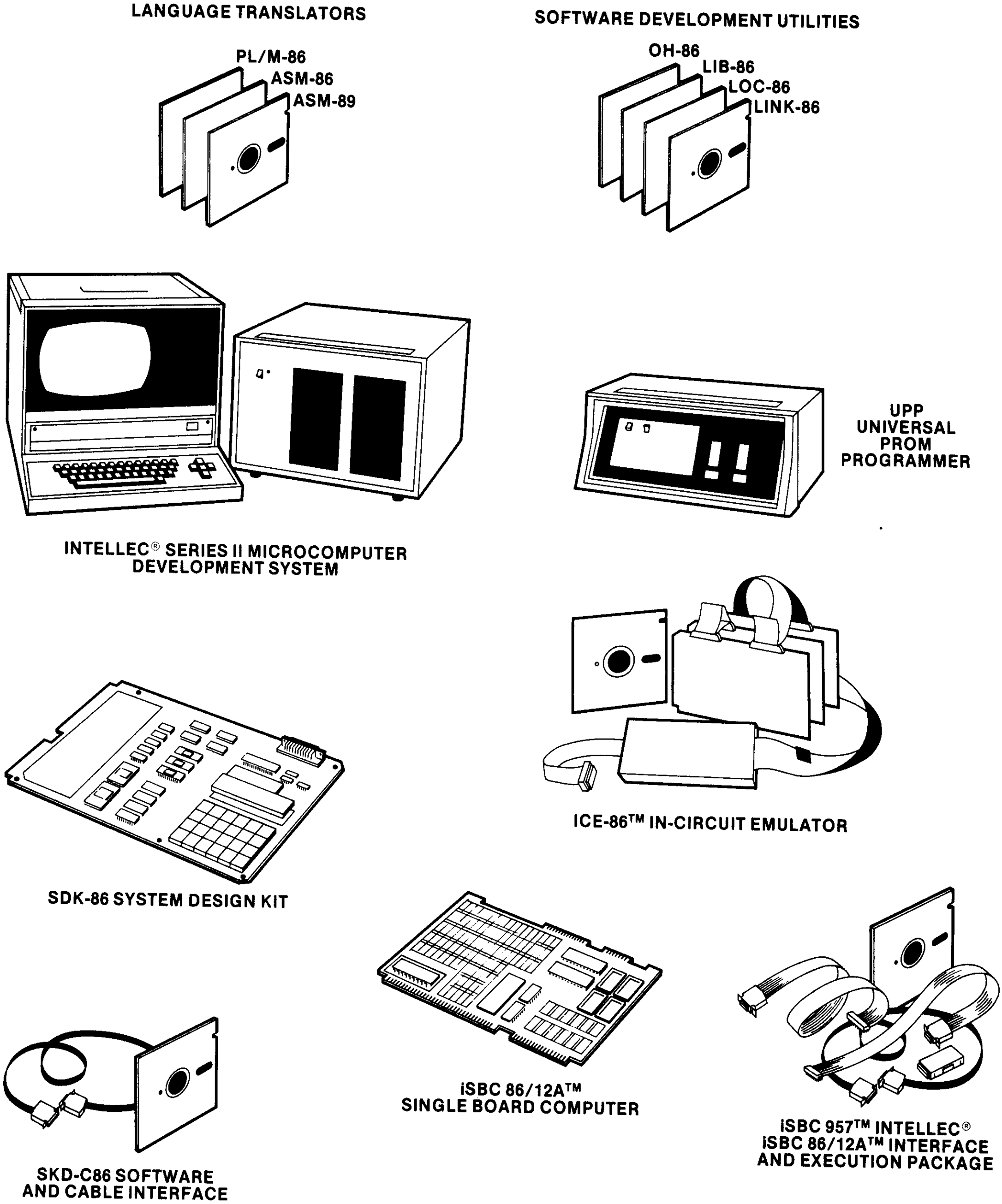



Figure 1-9. 8086 Family Development Aids