4. External Microprocessor Interface System Bus. Bus operations.
5. Memory Interface. Organization of input-output. Types of system cycles.

Bus operation

The 8086 has a combined address and data bus commonly referred to as a time multiplexed bus. This technique provides the most efficient use of pins on the processor while permitting the use of a standard 40-lead package. This "local bus" can be buffered directly and used throughout the system with address latching provided on memory and I/O modules. In addition, the bus can also be demultiplexed at the processor with a single set of address latches if a standard non-multiplexed bus is desired for the system.

Each processor bus cycle consists of at least four CLK cycles. These are referred to as T_1, T_2, T_3 and T_4 (see Figure 5). The address is emitted from the processor during T_1 and data transfer occurs on the bus during T_3 and T_4. T_2 is used primarily for changing the direction of the bus during read operations. In the event that a "NOT READY" indication is given by the addressed device, "Wait" states (T_W) are inserted between T_3 and T_4. Each inserted "Wait" state is of the same duration as a CLK cycle. Periods

MINIMUM AND MAXIMUM MODES

The requirements for supporting minimum and maximum 8086 systems are sufficiently different that they cannot be done efficiently with 40 uniquely defined pins. Consequently, the 8086 is equipped with a strap pin (MN/$\overline{MX}$) which defines the system configuration. The definition of a certain subset of the pins changes dependent on the condition of the strap pin. When MN/$\overline{MX}$ pin is strapped to GND, the 8086 treats pins 24 through 31 in maximum mode. An 8288 bus controller interprets status information coded into $\overline{S_0}$, $\overline{S_2}$, $\overline{S_2}$ to generate bus timing and control signals compatible with the MULTIBUS architecture. When the MN/$\overline{MX}$ pin is strapped to V_{CC}, the 8086 generates bus control signals itself on pins 24 through 31, as shown in parentheses in Figure 2. Examples of minimum mode and maximum mode systems are shown in Figure 4.

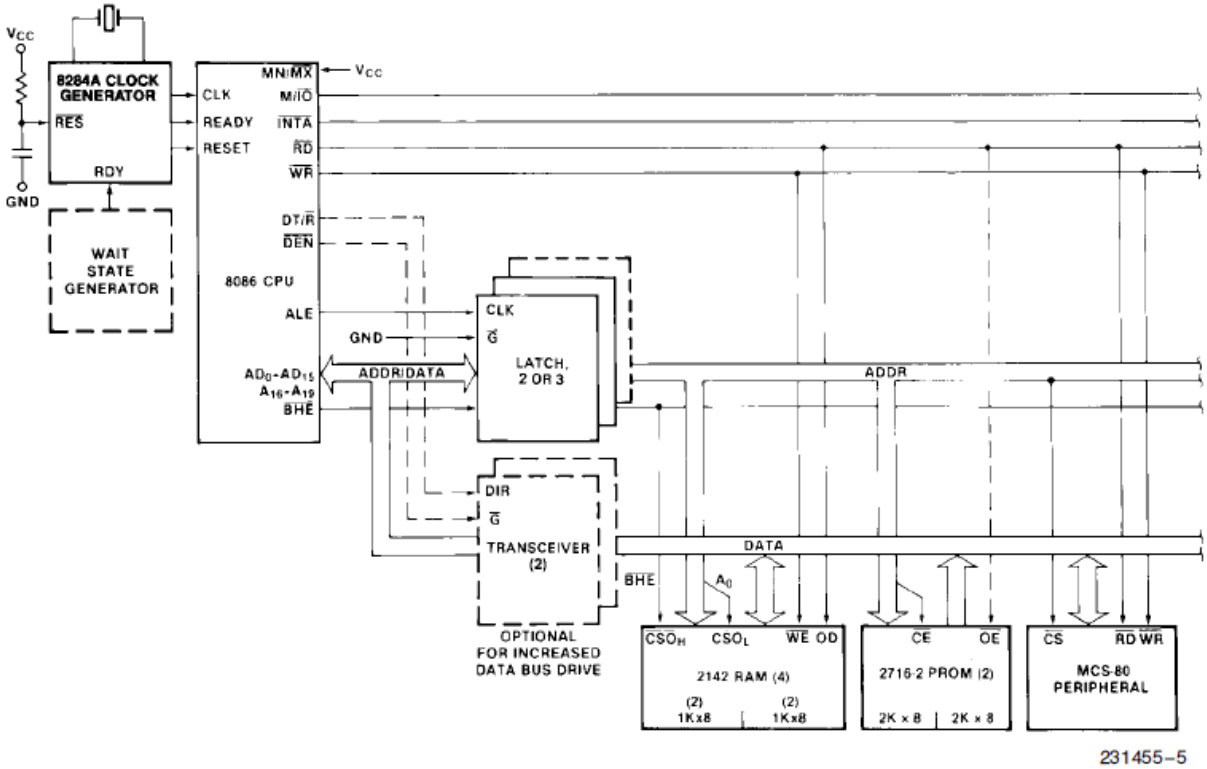

Figure 4a. Minimum Mode 8086 Typical Configuration

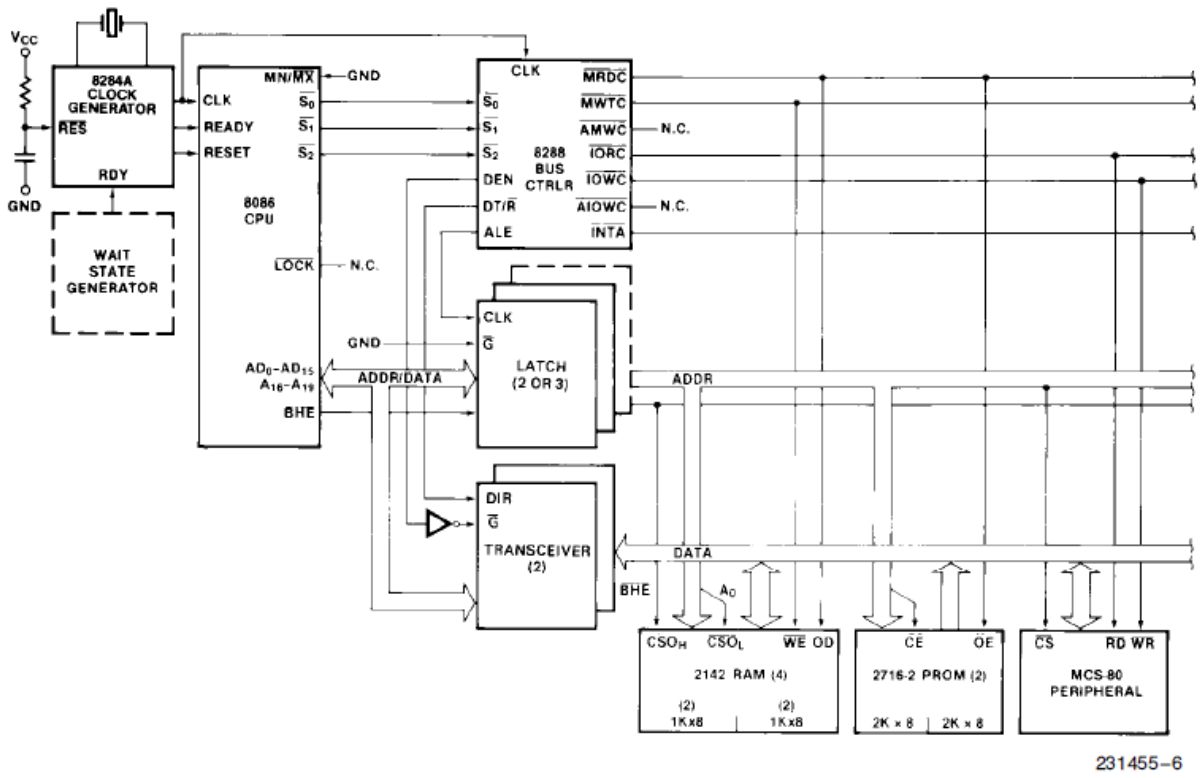

Figure 4b. Maximum Mode 8086 Typical Configuration

can occur between 8086 bus cycles. These are referred to as "Idle" states (T_I) or inactive CLK cycles. The processor uses these cycles for internal housekeeping.

During T_1 of any bus cycle the ALE (Address Latch Enable) signal is emitted (by either the processor or the 8288 bus controller, depending on the MN/$\overline{MX}$ strap). At the trailing edge of this pulse, a valid address and certain status information for the cycle may be latched.

Status bits $\overline{S_0}$, $\overline{S_1}$, and $\overline{S_2}$ are used, in maximum mode, by the bus controller to identify the type of bus transaction according to the following table:

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Characteristics
0 (LOW)	0	0	Interrupt Acknowledge
0	0	1	Read I/O
0	1	0	Write I/O
0	1	1	Halt
1 (HIGH)	0	0	Instruction Fetch
1	0	1	Read Data from Memory
1	1	0	Write Data to Memory
1	1	1	Passive (no bus cycle)

Status bits S_3 through S_7 are multiplexed with high-order address bits and the $\overline{BHE}$ signal, and are therefore valid during T_2 through T_4. S_3 and S_4 indicate which segment register (see Instruction Set description) was used for this bus cycle in forming the address, according to the following table:

S_4	S_3	Characteristics
0 (LOW)	0	Alternate Data (extra segment)
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data

S_5 is a reflection of the PSW interrupt enable bit. $S_6 = 0$ and S_7 is a spare status bit.

I/O ADDRESSING

In the 8086, I/O operations can address up to a maximum of 64K I/O byte registers or 32K I/O word registers. The I/O address appears in the same format as the memory address on bus lines A_{15}–A_0. The address lines A_{19}–A_{16} are zero in I/O operations. The variable I/O instructions which use register DX as a pointer have full address capability while the direct I/O instructions directly address one or two of the 256 I/O byte locations in page 0 of the I/O address space.

I/O ports are addressed in the same manner as memory locations. Even addressed bytes are transferred on the D_7–D_0 bus lines and odd addressed bytes on D_{15}–D_8. Care must be taken to assure that each register within an 8-bit peripheral located on the lower portion of the bus be addressed as even.

External Interface

PROCESSOR RESET AND INITIALIZATION

Processor initialization or start up is accomplished with activation (HIGH) of the RESET pin. The 8086 RESET is required to be HIGH for greater than 4 CLK cycles. The 8086 will terminate operations on the high-going edge of RESET and will remain dormant as long as RESET is HIGH. The low-going transition of RESET triggers an internal reset sequence for approximately 10 CLK cycles. After this interval the 8086 operates normally beginning with the instruction in absolute location FFFF0H (see Figure 3b). The details of this operation are specified in the Instruction Set description of the MCS-86 Family User's Manual. The RESET input is internally synchronized to the processor clock. At initialization the HIGH-to-LOW transition of RESET must occur no sooner than 50 μs after power-up, to allow complete initialization of the 8086.

NMI asserted prior to the 2nd clock after the end of RESET will not be honored. If NMI is asserted after that point and during the internal reset sequence, the processor may execute one instruction before responding to the interrupt. A hold request active immediately after RESET will be honored before the first instruction fetch.

All 3-state outputs float to 3-state OFF during RESET. Status is active in the idle state for the first clock after RESET becomes active and then floats to 3-state OFF. ALE and HLDA are driven low.

EXTERNAL SYNCHRONIZATION VIA TEST

As an alternative to the interrupts and general I/O capabilities, the 8086 provides a single software-testable input known as the $\overline{TEST}$ signal. At any time the program may execute a WAIT instruction. If at that time the $\overline{TEST}$ signal is inactive (HIGH), program execution becomes suspended while the processor waits for $\overline{TEST}$ to become active. It must remain active for at least 5 CLK cycles. The WAIT instruction is re-executed repeatedly until that time. This activity does not consume bus cycles. The processor remains in an idle state while waiting. All 8086 drivers go to 3-state OFF if bus "Hold" is entered. If interrupts are enabled, they may occur while the processor is waiting. When this occurs the processor fetches the WAIT instruction one extra time, processes the interrupt, and then re-fetches and re-executes the WAIT instruction upon returning from the interrupt.

Basic System Timing

Typical system configurations for the processor operating in minimum mode and in maximum mode are shown in Figures 4a and 4b, respectively. In minimum mode, the MN/$\overline{MX}$ pin is strapped to V_{CC} and the processor emits bus control signals in a manner similar to the 8085. In maximum mode, the MN/$\overline{MX}$ pin is strapped to V_{SS} and the processor emits coded status information which the 8288 bus controller uses to generate MULTIBUS compatible bus control signals. Figure 5 illustrates the signal timing relationships.

SYSTEM TIMING—MINIMUM SYSTEM

The read cycle begins in T_1 with the assertion of the Address Latch Enable (ALE) signal. The trailing (low-going) edge of this signal is used to latch the address information, which is valid on the local bus at this time, into the address latch. The $\overline{BHE}$ and A_0 signals address the low, high, or both bytes. From T_1 to T_4 the M/$\overline{IO}$ signal indicates a memory or I/O operation. At T_2 the address is removed from the local bus and the bus goes to a high impedance state. The read control signal is also asserted at T_2. The read ($\overline{RD}$) signal causes the addressed device to enable its data bus drivers to the local bus. Some time later valid data will be available on the bus and the addressed device will drive the READY line HIGH. When the processor returns the read signal to a HIGH level, the addressed device will again 3-state its bus drivers. If a transceiver is required to buffer the 8086 local bus, signals DT/$\overline{R}$ and $\overline{DEN}$ are provided by the 8086.

A write cycle also begins with the assertion of ALE and the emission of the address. The M/$\overline{IO}$ signal is again asserted to indicate a memory or I/O write operation. In the T_2 immediately following the address emission the processor emits the data to be written into the addressed location. This data remains valid until the middle of T_4. During T_2, T_3, and T_W the processor asserts the write control signal. The write ($\overline{WR}$) signal becomes active at the beginning of T_2 as opposed to the read which is delayed somewhat into T_2 to provide time for the bus to float.

The $\overline{BHE}$ and A_0 signals are used to select the proper byte(s) of the memory/IO word to be read or written according to the following table:

$\overline{BHE}$	A0	Characteristics
0	0	Whole word
0	1	Upper byte from/to odd address
1	0	Lower byte from/to even address
1	1	None

I/O ports are addressed in the same manner as memory location. Even addressed bytes are transferred on the D_7-D_0 bus lines and odd addressed bytes on $D_{15}-D_8$.

The basic difference between the interrupt acknowledge cycle and a read cycle is that the interrupt acknowledge signal ($\overline{INTA}$) is asserted in place of the read ($\overline{RD}$) signal and the address bus is floated. (See Figure 6.) In the second of two successive INTA cycles, a byte of information is read from bus lines D_7-D_0 as supplied by the inerrupt system logic (i.e., 8259A Priority Interrupt Controller). This byte identifies the source (type) of the interrupt. It is multiplied by four and used as a pointer into an interrupt vector lookup table, as described earlier.

BUS TIMING—MEDIUM SIZE SYSTEMS

For medium size systems the MN/$\overline{MX}$ pin is connected to V_{SS} and the 8288 Bus Controller is added to the system as well as a latch for latching the system address, and a transceiver to allow for bus loading greater than the 8086 is capable of handling. Signals ALE, DEN, and DT/$\overline{R}$ are generated by the 8288 instead of the processor in this configuration although their timing remains relatively the same. The 8086 status outputs ($\overline{S_2}$, $\overline{S_1}$, and $\overline{S_0}$) provide type-of-cycle information and become 8288 inputs. This bus cycle information specifies read (code, data, or I/O), write (data or I/O), interrupt acknowledge, or software halt. The 8288 thus issues control signals specifying memory read or write, I/O read or write, or interrupt acknowledge. The 8288 provides two types of write strobes, normal and advanced, to be applied as required. The normal write strobes have data valid at the leading edge of write. The advanced write strobes have the same timing as read strobes, and hence data isn't valid at the leading edge of write. The transceiver receives the usual DIR and $\overline{G}$ inputs from the 8288's DT/$\overline{R}$ and DEN.

The pointer into the interrupt vector table, which is passed during the second INTA cycle, can derive from an 8259A located on either the local bus or the system bus. If the master 8259A Priority Interrupt Controller is positioned on the local bus, a TTL gate is required to disable the transceiver when reading from the master 8259A during the interrupt acknowledge sequence and software "poll".

Below Fig. 5 shows bus timing waveform in maximum mode for 8086 system.

NOTES:
1. All signals switch between V_{OH} and V_{OL} unless otherwise specified.
2. RDY is sampled near the end of T_2, T_3, T_W to determine if T_W machines states are to be inserted.
3. Cascade address is valid between first and second INTA cycle.
4. Two INTA cycles run back-to-back. The 8086 LOCAL ADDR/DATA BUS is floating during both INTA cycles. Control for pointer address is shown for second INTA cycle.
5. Signals at 8284A or 8288 are shown for reference only.
6. The issuance of the 8288 command and control signals ($\overline{MRDC}$, $\overline{MWTC}$, $\overline{AMWC}$, $\overline{IORC}$, $\overline{IOWC}$, $\overline{AIOWC}$, $\overline{INTA}$ and DEN) lags the active high 8288 CEN.
7. All timing measurements are made at 1.5V unless otherwise noted.
8. Status inactive in state just prior to T_4.

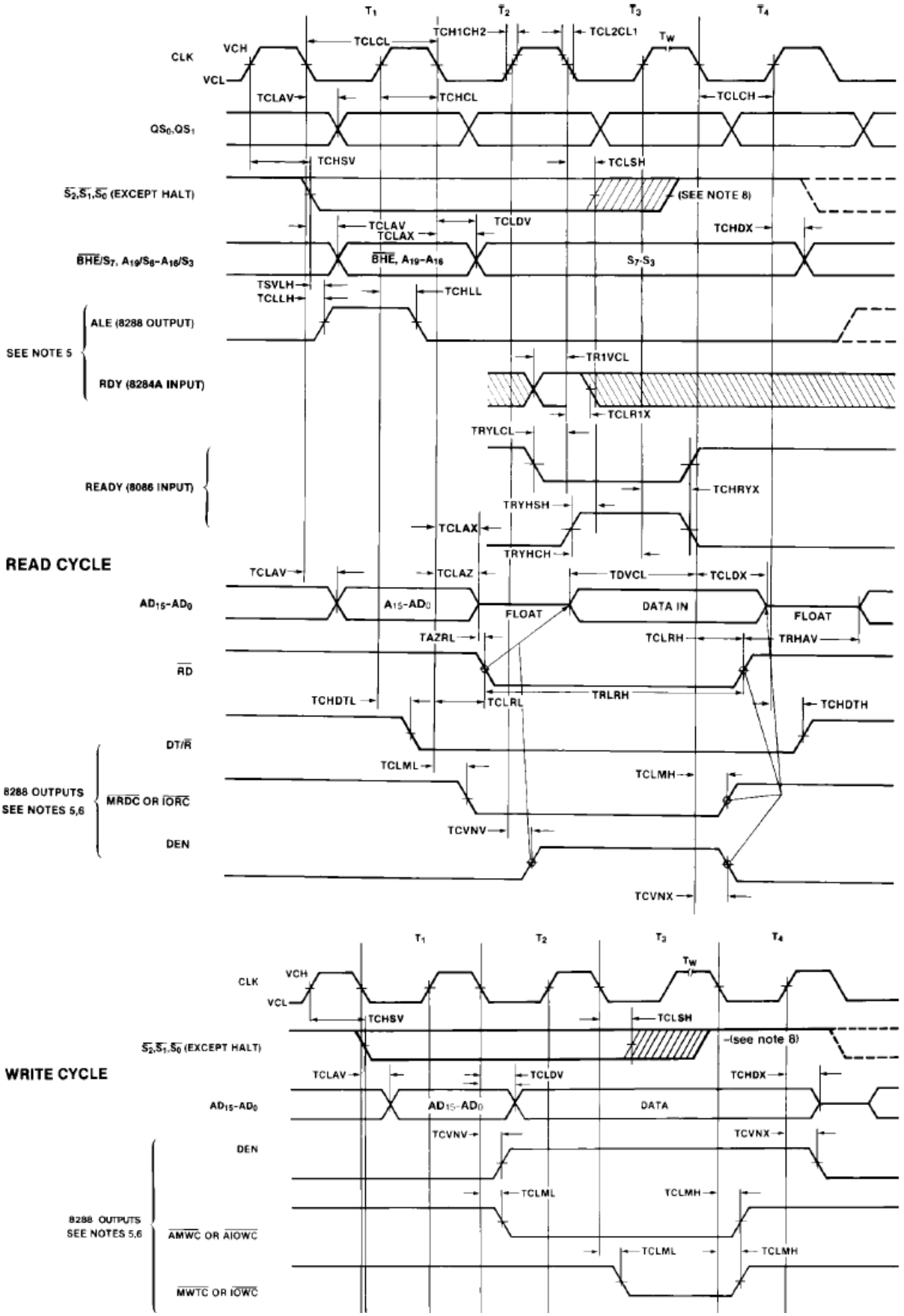

Fig. 5 Maximum Mode Waveform

I8086 Microprocessor system bus

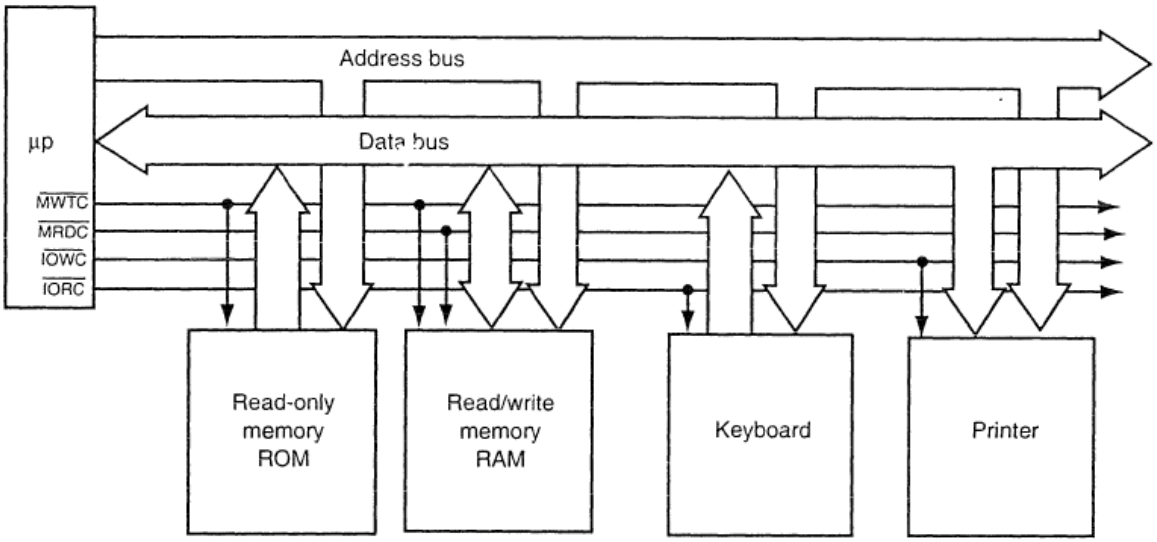

Fig. 6 Block-diagram of a typical Microprocessor system based on I8086

Buses. A **bus** is a common group of wires that interconnect components in a computer system. The buses that interconnect the sections of a computer system transfer address, data, and control information between the microprocessor and its memory and I/O systems. In the microprocessor-based computer system, three buses exist for this transfer of information: address, data, and control. Figure 1–8 shows how these buses interconnect various system components such as the microprocessor, read/write memory (RAM), read-only memory (ROM), and a few I/O devices.

The address bus requests a memory location from the memory or an I/O location from the I/O devices. If I/O is addressed, the address bus contains a 16-bit I/O address from 0000H–FFFFH. The 16-bit I/O address or port number selects one of 64K different I/O devices. If memory is addressed, the address bus contains a memory address. The memory address varies in width with the different versions of the microprocessor. The 8086 and 8088 address 1M byte of memory using a 20-bit address that selects locations 00000H–FFFFFH. The 80286 and 80386SX address 16M bytes of memory using 24b address.

The data bus transfers information between the microprocessor and its memory and I/O address space. Data transfers vary in size from 8-bits wide to 64-bits wide in various members of the Intel microprocessor family. The 8088 contains an 8-bit data bus that transfers 8-bits of data at a time. The 8086, 80286, 80386SL, 80386SX, and 80386EX transfer 16-bits of data through their data buses; the 80386DX, 80486SX, and 80486DX transfer 32-bits of data; and finally, the Pentium and Pentium Pro transfer 64-bits of data. The advantage of a wider data bus is speed in applications that use wide data. For example, if a 32-bit number is stored in memory, it takes the 8088 microprocessor four transfer operations to complete because its data bus is only 8-bits wide.

The control bus contains lines that select the memory or I/O and cause them to perform a read or a write operation. In most computer systems, there are four control bus connections: $\overline{\text{MRDC}}$ **(memory read control),** $\overline{\text{MWTC}}$ **(memory write control),** $\overline{\text{IORC}}$ **(I/O read control),** and $\overline{\text{IOWC}}$ **(I/O write control).** Note that the over-bar indicates that the control signal is active-low—that is, it is active when a logic zero appears on the control line. For example, if $\overline{\text{IOWC}} = 0$, the microprocessor is writing data from the data bus to an I/O device whose address appears on the address bus.

The microprocessor reads the contents of a memory location by sending the memory an address through the address bus. Next, it sends the memory read control signal ($\overline{\text{MRDC}}$) to cause memory to read data. Finally the data read from the memory are passed to the microprocessor through the data bus. Whenever a memory write, I/O write, or I/O read occurs, the same sequence

ensues except that different control signals are issued and the data flow out of the microprocessor through its data bus for a write operation.

Basic 8088/80188 Memory Interface

Both the 8088 and 80188 microprocessors have an 8-bit data bus, which makes it ideal to connect to the common 8-bit memory devices available today. The 8-bit memory size makes the 8088 and especially the 80188 ideal as a simple controller. For the 8088/80188 to function correctly with the memory, however, the memory system must decode the address to select a memory component, and it must use the RD, WR, and IO/M control signals provided by the 8088/80188 to control the memory system.

The minimum mode configuration is used in this section and is essentially the same as the maximum mode system for memory interface. The main difference is that, in maximum mode, IO/$\overline{M}$ is combined with $\overline{RD}$ to generate an $\overline{MRDC}$ signal, and IO/$\overline{M}$ is combined with $\overline{WR}$ to generate an $\overline{MWTC}$ signal. These maximum mode control signals are developed inside the 8288 bus controller. In the minimum mode, the memory sees the 8088 or the 80188 as a device with 20 address connections (A_{19}–A_0), 8 data bus connections(AD_7–AD_0) and the control signals IO/$\overline{M}$, $\overline{RD}$, and $\overline{WR}$.

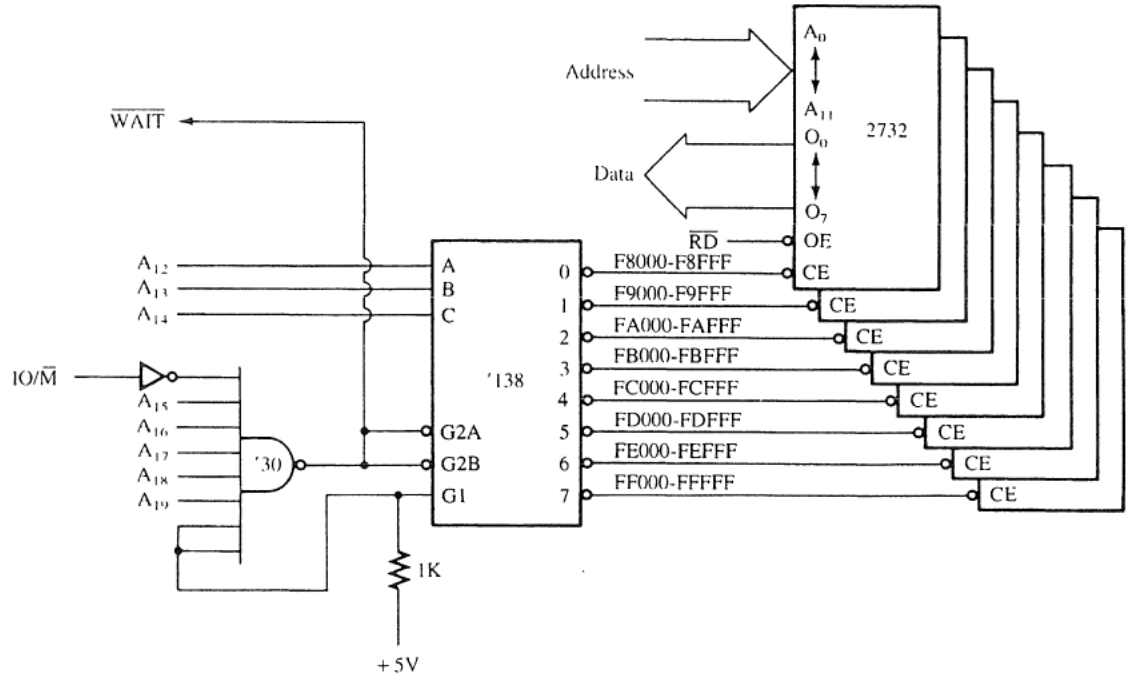

FIGURE 9–19 Eight 2732 EPROMs interfaced to the 8088 microprocessor. Note that the output of the NAND gate is used to cause a wait state whenever this section of the memory is selected.

Figure 9–19 illustrates an 8088 microprocessor connected to eight 2732 EPROMs, 4K × 8 memory devices that are in very common use today. The 2732 has one more address input (A_{11}) than the 2716, and twice the memory. The device in this illustration decodes eight 4K × 8 blocks of memory, for a total of 32K × 8 bits of the physical address space for the 8088.

The decoder (74LS138) is connected a little differently than might be expected because the slower version of this type of EPROM has a memory access time of 450 ns. Recall from Chapter 8 that when the 8088 is operated with a 5 MHz clock, it allows 460 ns for the memory to access data. Because of the decoder's added time delay (12 ns), it is impossible for this memory to function within 460 ns. In order to correct this problem, we must add a NAND gate to generate a signal to enable the decoder and a signal for the wait state generator, covered in Chapter 8. (Note that the 80188 can internally insert from 0–15 wait states without any additional external hardware, so it does not require this NAND gate.) With a wait state inserted every time this section of the memory is accessed, the 8088 will allow 660 ns for the EPROM to access data. Recall that an extra wait state adds 200 ns (1 clock) to the access time. The 660 ns is ample time for a 450 ns memory component to access data, even with the delays introduced by the decoder and any buffers added to the data bus.

Notice that the decoder is selected for a memory address range that begins at location F8000H and continues through location FFFFFH–the upper 32K bytes of memory. This section of memory is an EPROM because FFFF0H is where the 8088 starts to execute instructions after a hardware reset. We often call location FFFF0H the **cold-start** location. The software stored in this section of memory would contain a JMP instruction at location FFFF0H that jumps to location F8000H so that the remainder of the program can execute.

Interfacing RAM to the 8088. RAM is a little easier to interface than EPROM because most RAM memory components do not require wait states. An ideal section of the memory for the RAM is the very bottom, which contains vectors for interrupts. Interrupt vectors (discussed in more detail in Chapter 11) are often modified by software packages, so it is rather important to encode this section of the memory with RAM.

In Figure 9–20, sixteen 62256 32K × 8 static RAMS are interfaced to the 8088, beginning at memory location 00000H. This circuit board uses two decoders to select the sixteen different RAM memory components and a third to select the other decoders for the appropriate memory sections. Sixteen 32K RAMs fill memory from location 00000H through location 7FFFFH, for 512K bytes of memory.

The first decoder (U4) in this circuit selects the other two decoders. An address beginning with 00 selects decoder U3, and an address that begins with 01 selects decoder U9. Notice that extra pins remain at the output of decoder U4 for future expansion. These allow more 256K × 8 blocks of RAM, for a total of 1M × 8, simply by adding the RAM and the additional secondary decoders.

Also notice from the circuit in Figure 9–20 that all the address inputs to this section of memory are buffered, as are the data bus connections and control signals $\overline{RD}$ and $\overline{WR}$. Buffering is important when many devices appear on a single board or in a single system. Suppose that three other boards like this are plugged into a system. Without the buffers on each board, the load on the system address, data, and control buses would be enough to prevent proper operation. (Excessive loading causes the logic 0 output to rise above the 0.8 V maximum allowed in a system.) Buffers are normally used if the memory will contain additions at some future date. If the memory will never grow, then buffers may not be needed.

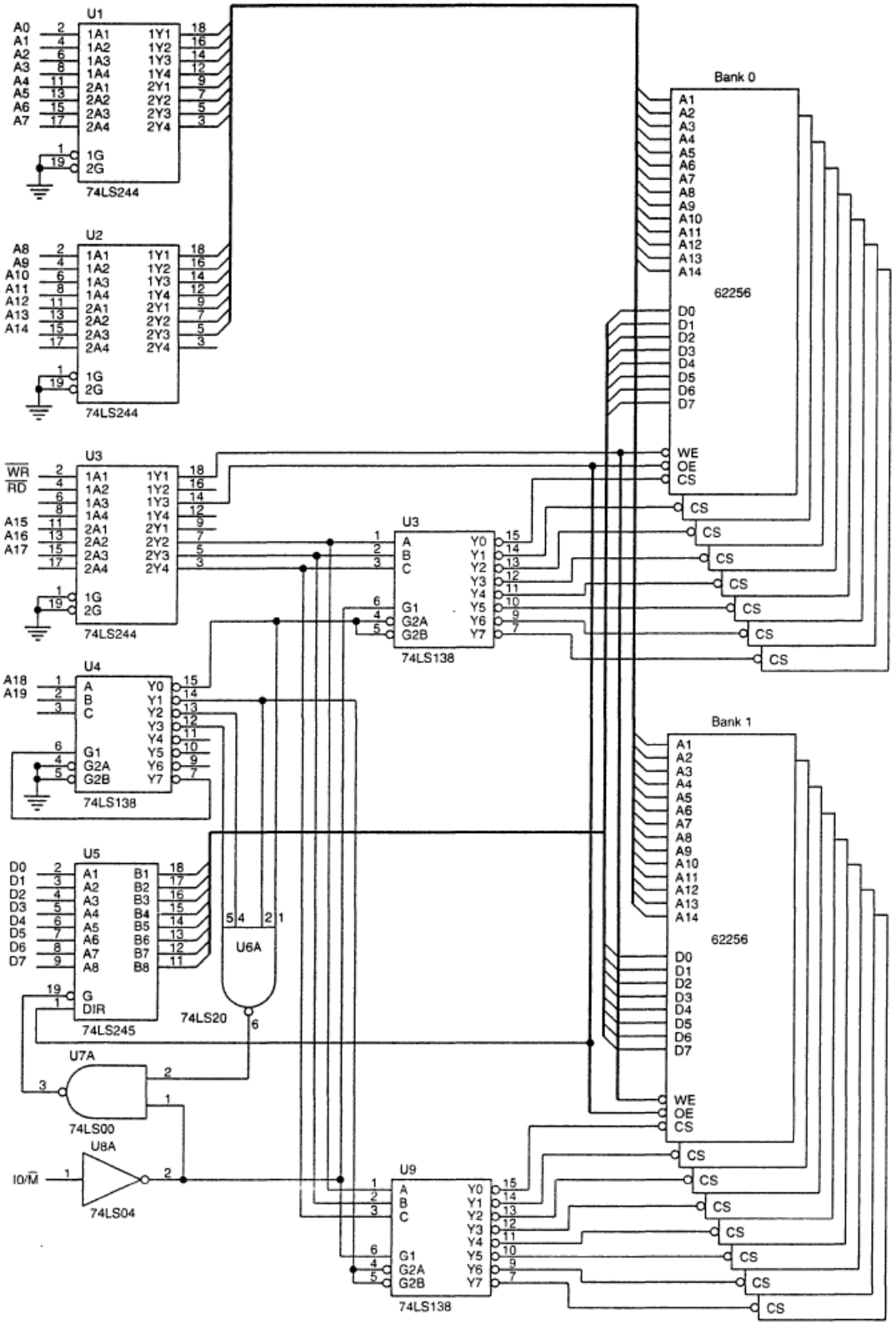

FIGURE 9–20 A 512K byte static memory system using 16 62255 SRAMS

Interfacing Flash Memory

Flash memory (EEPROM) is becoming commonplace for storing setup information on video cards as well as for storing the system BIOS in the personal computer. Flash memory is also found in many other applications to store information that is only changed occasionally.

The only difference between a flash memory device and SRAM is that the flash memory device requires a 12 V programming voltage to erase and write new data. The 12 V can either be available at the power supply, or a 5 V-to-12 V converter designed for use with flash memory can be obtained.

Figure 9–21 illustrates a 28F400 Intel flash memory device interfaced to the 8088 microprocessor. The 28F400 can be used as either a 512K × 8 memory device or as a 256K × 16 memory device. Because it is interfaced to the 8088, its configuration is 512K × 8. Notice that

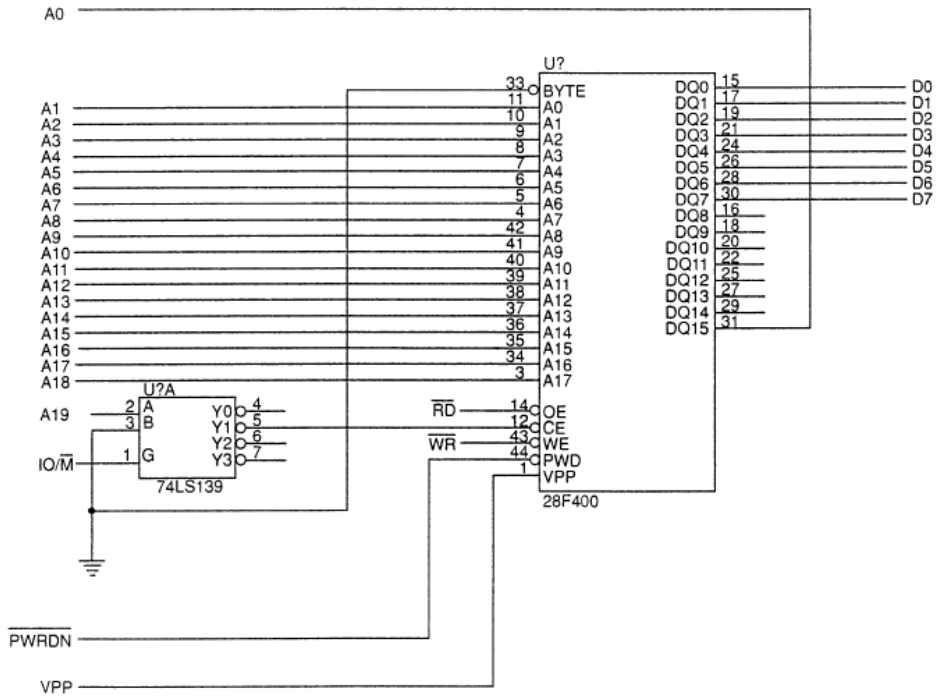

FIGURE 9–21 The 28F400 flash memory device interfaced to the 8088 microprocessor

the control connections on this device are identical to that of an SRAM–$\overline{CE}$, $\overline{OE}$, and $\overline{WE}$. The only new pins are V_{PP}, which is connected to 12 V for erase and programming; $\overline{PWD}$, which selects the power down mode when a logic 0 and is also used for programming; and $\overline{BYTE}$, which selects byte (0) or word (1) operation. Note that the pin DQ15 functions as the least-significant address input when operated in the byte mode. Another difference is the amount of time required to accomplish a write operation. The SRAM can perform a write operation in as little as 10 ns, but the flash memory requires approximately 0.4 seconds to erase a byte. The topic of programming the flash memory device is covered in Chapter 10, along with I/O devices. The flash memory device has some internal register and is programmed using I/O techniques not yet explained. This chapter concentrates on its interface to the microprocessor.

Notice in Figure 9–21 that the decoder chosen is the 74LS139, because only a simple decoder is needed for a flash memory device this large. The decoder uses address connection A_{19} and IO/$\overline{M}$ as inputs. The A_{15} signal selects the flash memory for locations 80000H through FFFFFH, and IO/$\overline{M}$ enables the decoder.