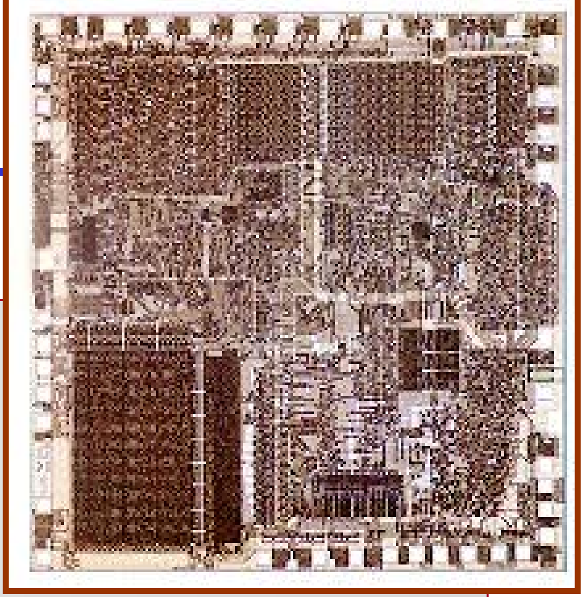
Week 7

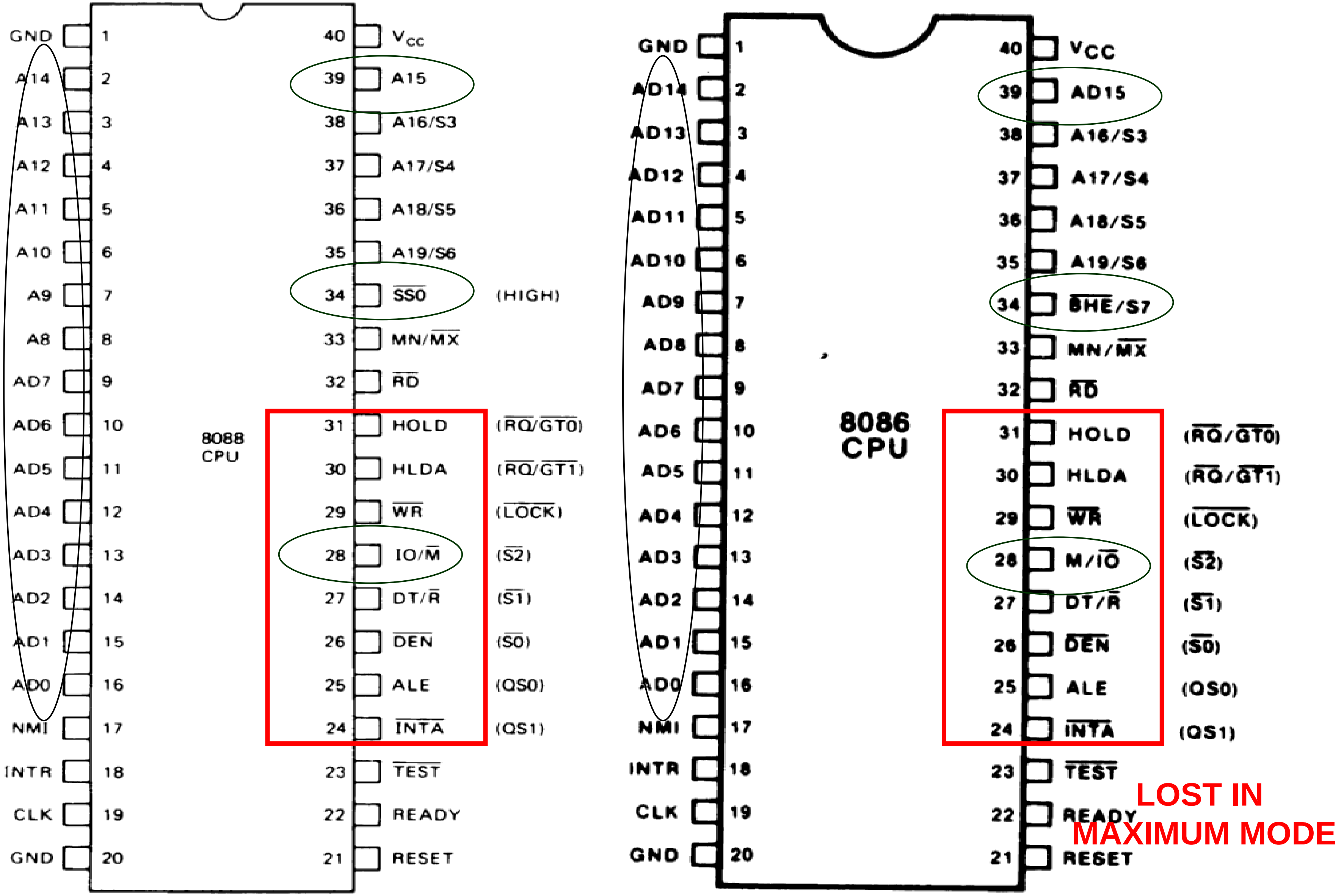
The 8088 and 8086 Microprocessors

8086 and 8088 Microprocessors



- 8086 announced in 1978; 8086 is a 16 bit microprocessor with a 16 bit data bus
- 8088 announced in 1979; 8088 is a 16 bit microprocessor with an 8 bit data bus
- Both manufactured using High-performance Metal Oxide Semiconductor (HMOS) technology
- Both contain about 29000 transistors
- Both are packaged in 40 pin dual-in-line package (DIP)
- Address lines A0-A7 and Data lines D0-D7 are multiplexed in 8088. These lines are labelled as AD0-AD7.
  - By multiplexed we mean that the same pysical pin carries an address bit at one time and the data bit another time
- Address lines A0-A15 and Data lines D0-D15 are multiplexed in 8086. These lines are labelled as AD0-AD15.

8088 and 8086 Microprocessors



LOST IN MAXIMUM MODE

Minimum-mode and Maximum-mode Systems

- 8088 and 8086 microprocessors can be configured to work in either of the two modes: the minimum mode and the maximum mode

✓ Minimum mode:

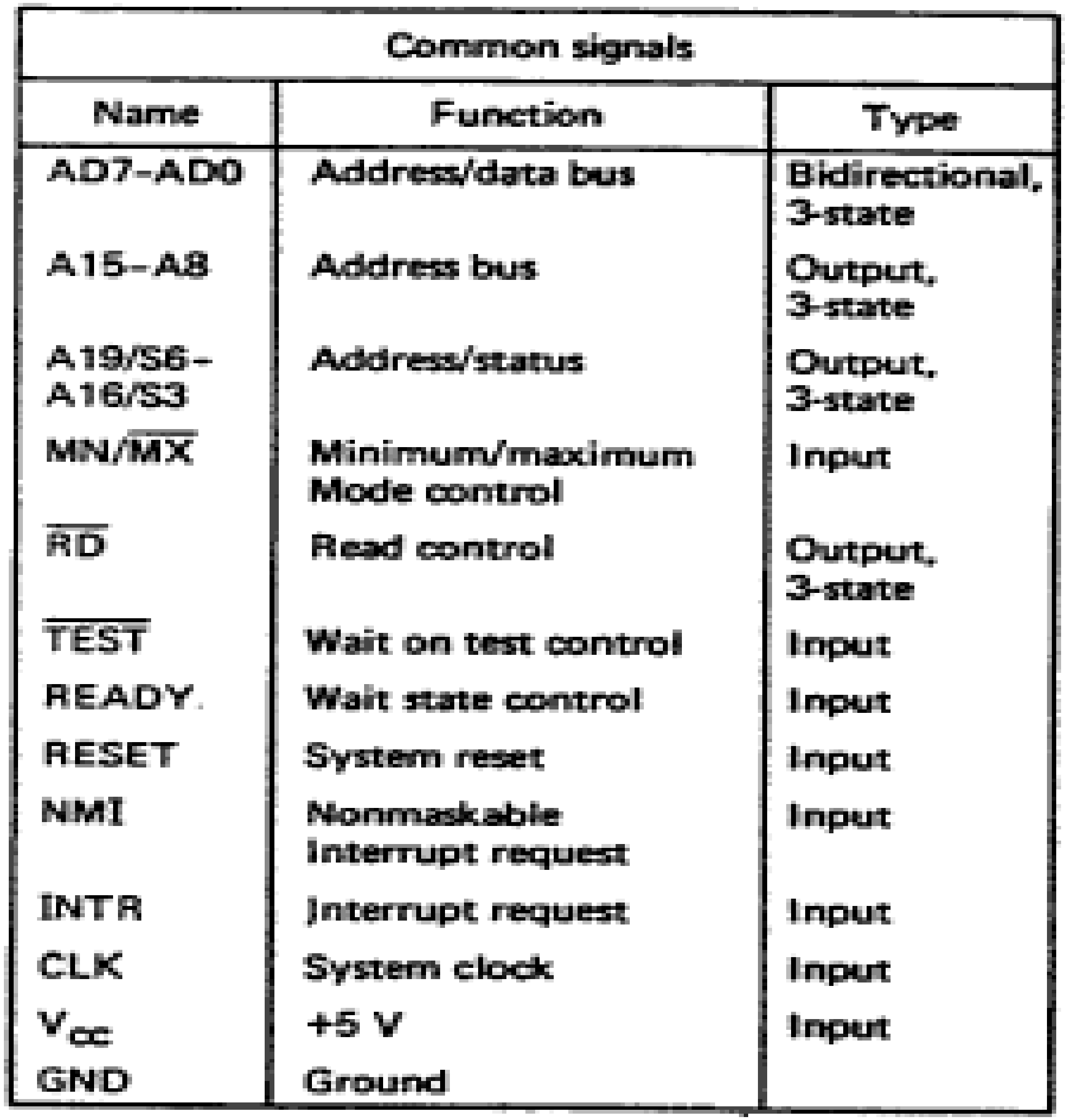
- ➢ Pull MN/$\overline{\text{MX}}$ to logic 1
- ➢ Typically smaller systems and contains a single microprocessor
- ➢ Cheaper since all control signals for memory and I/O are generated by the microprocessor.

✓ Maximum mode

- ➢ Pull MN/$\overline{\textbf{MX}}$ logic 0
- ➢ Larger systems with more than one processor (*designed to be used when a coprocessor (8087) exists in the system)*

PINs on microprocessor

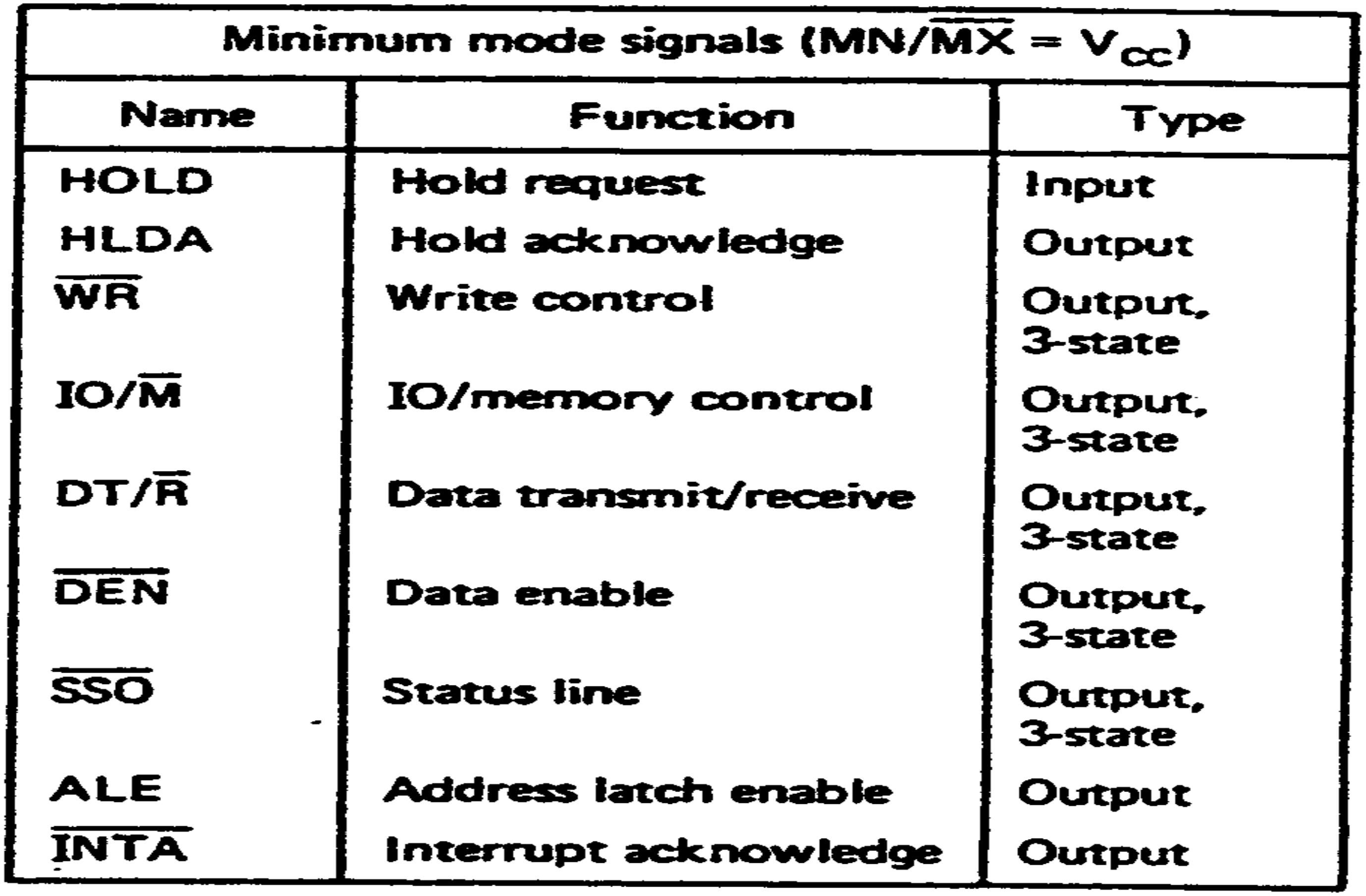
Signals common to both minimum and maximum systems→



Common signals		
Name	Function	Type
AD7–AD0	Address/data bus	Bidirectional, 3-state
A15–A8	Address bus	Output, 3-state
A19/S6– A16/S3	Address/status	Output, 3-state
MN/$\overline{MX}$	Minimum/maximum Mode control	Input
$\overline{RD}$	Read control	Output, 3-state
$\overline{TEST}$	Wait on test control	Input
READY	Wait state control	Input
RESET	System reset	Input
NMI	Nonmaskable Interrupt request	Input
INTR	Interrupt request	Input
CLK	System clock	Input
V_{cc}	+5 V	Input
GND	Ground	

PINs on microprocessor

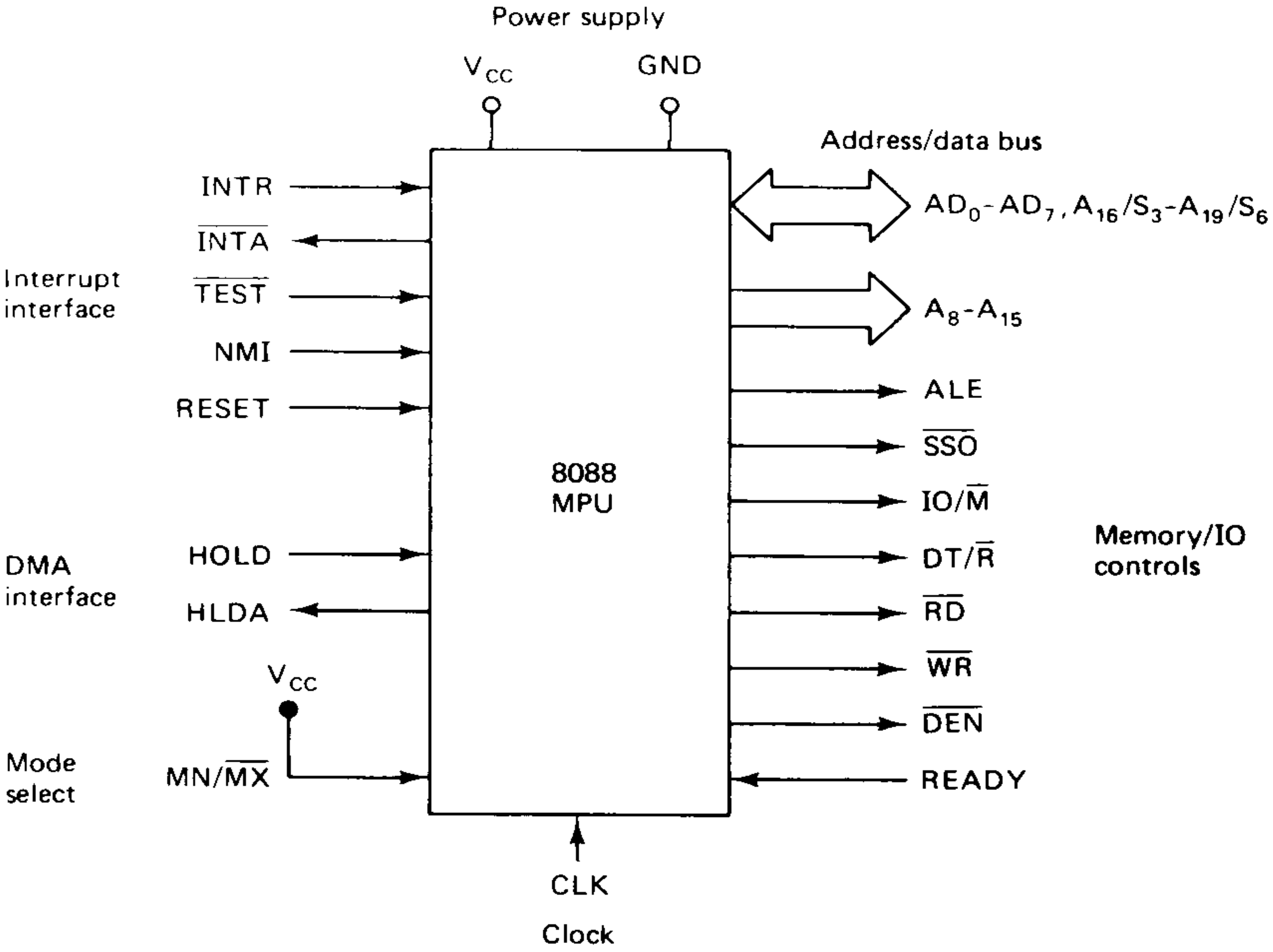
Minimum mode unique signals



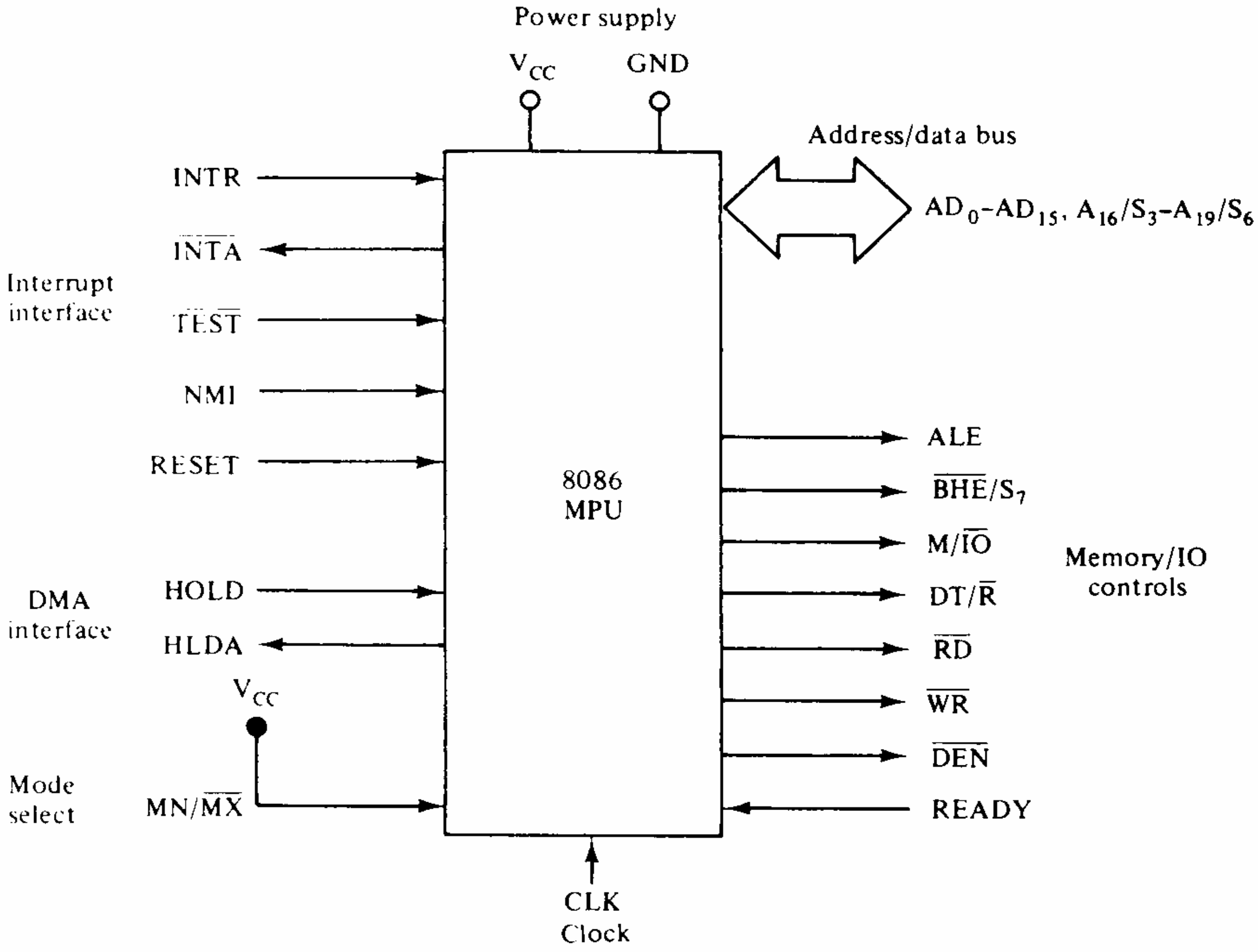
Minimum mode signals (MN/$\overline{MX}$ = V_{CC})

Name	Function	Type
HOLD	Hold request	Input
HLDA	Hold acknowledge	Output
$\overline{WR}$	Write control	Output, 3-state
IO/$\overline{M}$	IO/memory control	Output, 3-state
DT/$\overline{R}$	Data transmit/receive	Output, 3-state
$\overline{DEN}$	Data enable	Output, 3-state
$\overline{SSO}$	Status line	Output, 3-state
ALE	Address latch enable	Output
$\overline{INTA}$	Interrupt acknowledge	Output

8088 Minimum-mode block diagram



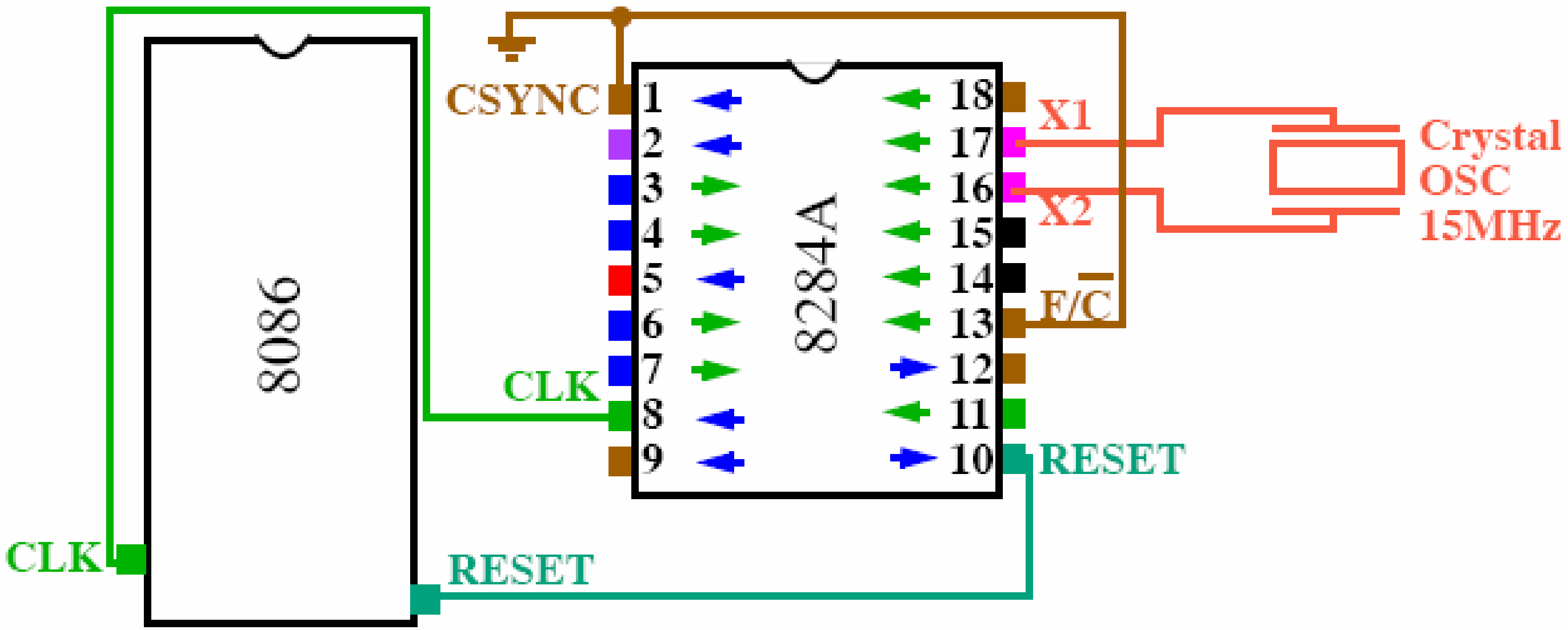
8086 Minimum-mode block diagram



8284 clock generator

- Clock generation
- RESET synchronization
- READY synchronization
- Peripheral clock signal

Connection of the 8284 and the 8086.



Minimum Mode Interface

- Address/Data bus: 20 bits/8 bits (AD0-AD7) multiplexed for 8088
- Address/Data bus: 20 bits/16 bits (AD0-AD15) multiplexed for 8086
- Status signals: A_{16}-A_{19} multiplexed with status signals S_3-S_6 respectively
  - **S3 and S4 together** form a 2 bit binary code that identifies which of the internal segment registers were used to generate the physical address that was output on the address bus during the current bus cycle.
  - **S5** is the logic level of the **IF**,
  - **S6 is always logic 0**. S0,S1,S2 used in maxmode (later)

S4	S3	Address status
0	0	Alternate(relative to ES segment)
0	1	Stack (relative to SS Segment)
1	0	Code/None (relative to CS segment or a default zero)
1	1	Data (relative to DS segment)

Minimum Mode Interface

- Control Signals: (8088)
  - ✓ **Address Latch Enable (ALE)** is a pulse to logic 1 that signals external circuitry when a valid address is on the bus. This address can be latched in external circuitry on the 1-to-0 edge of the pulse at ALE.
  - ✓ **IO/$\overline{\text{M}}$ line:** memory or I/O transfer is selected (complement for 8086)
  - ✓ **$\overline{\text{DT}}$/R line**: direction of data is selected
  - ✓ **$\overline{\text{RD}}$ line: =**0 when a read cycle is in progress
  - ✓ **$\overline{\text{WR}}$ line:** =0 when a write cyle is in progress
  - ✓ **$\overline{\text{DEN}}$ line: (Data enable)** Enables the external devices to supply data to the processor. Used when sharing memory with another processor.

Control Signals: (8088)

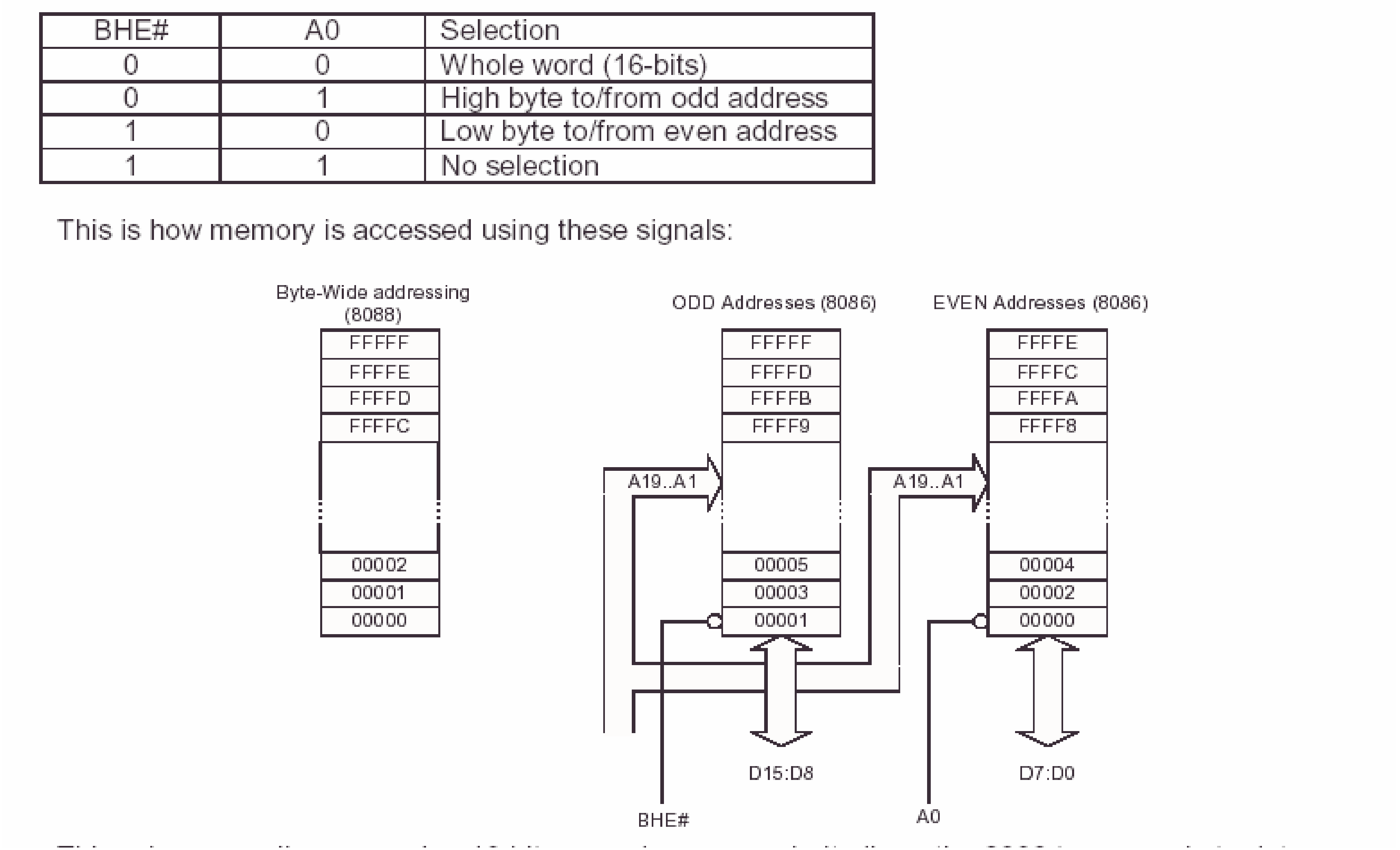
SSO (System Status Output) line **(only for 8088)**

=1 when data is read from memory and =0 when code is read from memory

IO/M	DT/R	SS0	FUNCTION
0	0	0	Interrupt Acknowledge
0	0	1	Memory Read
0	1	0	Memory Write
0	1	1	Halt
1	0	0	Opcode Fetch
1	0	1	I/O Read
1	1	0	I/O Write
1	1	1	Passive

Minimum Mode Interface

✓**BHE (Bank High Enable) line** (8086 only) :=0 for most significant byte of data and also carries S_7=1



BHE#	A0	Selection
0	0	Whole word (16-bits)
0	1	High byte to/from odd address
1	0	Low byte to/from even address
1	1	No selection

This is how memory is accessed using these signals:

Data Transfer

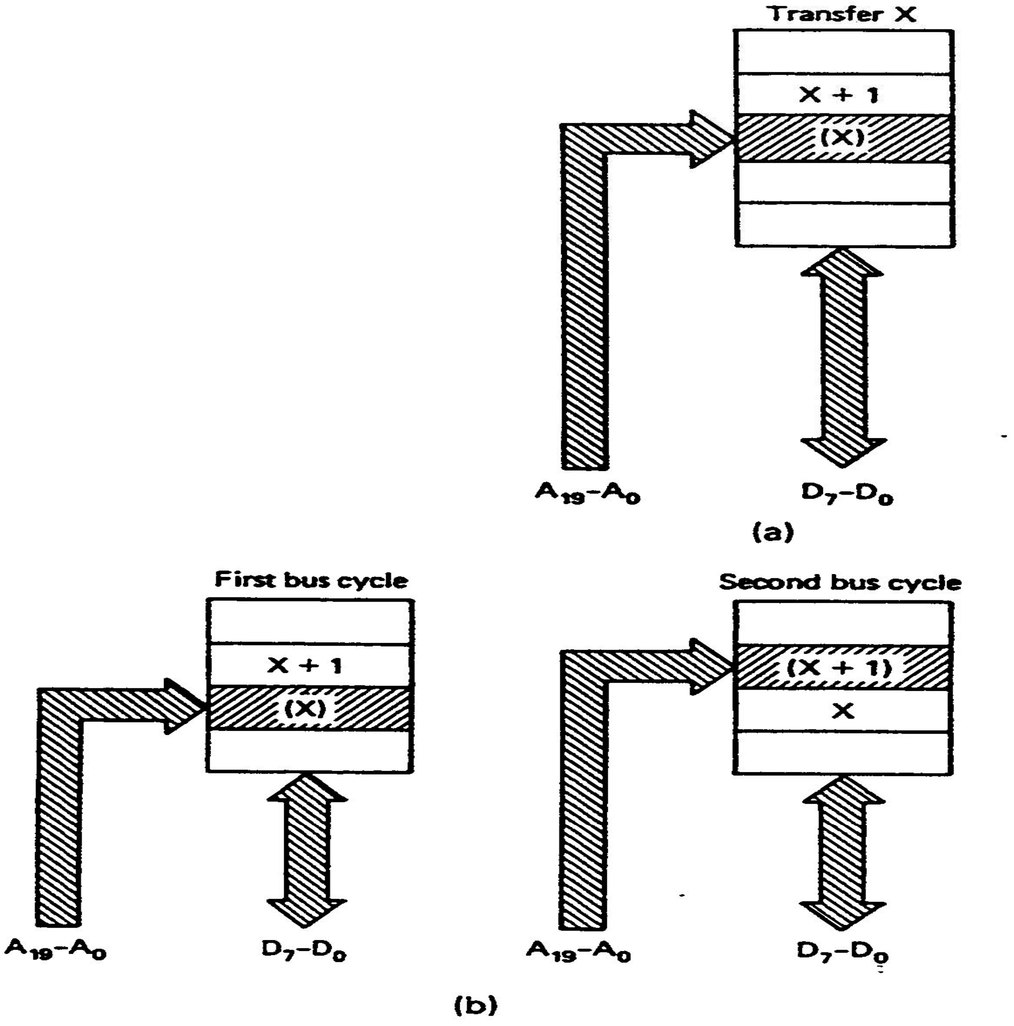
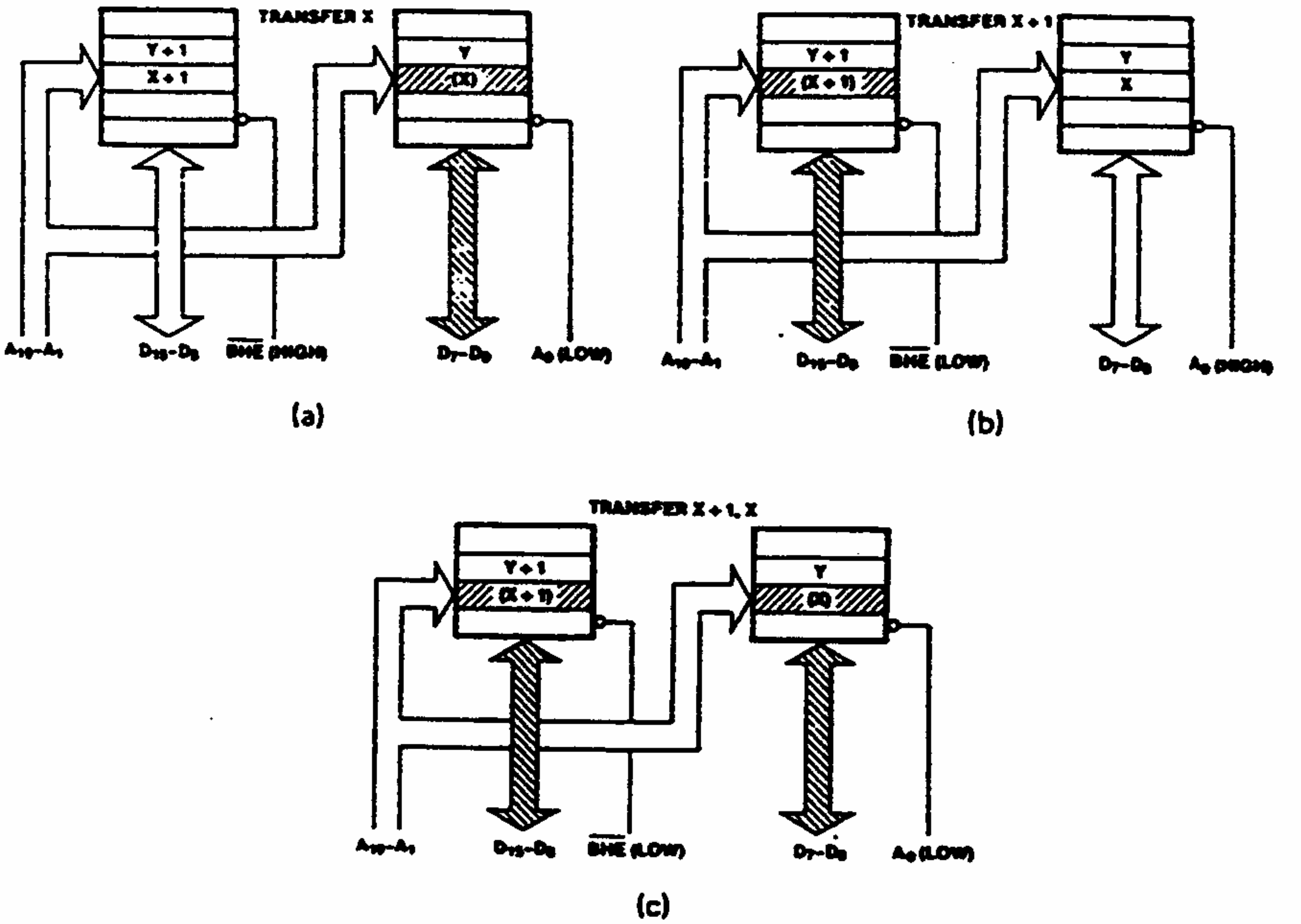


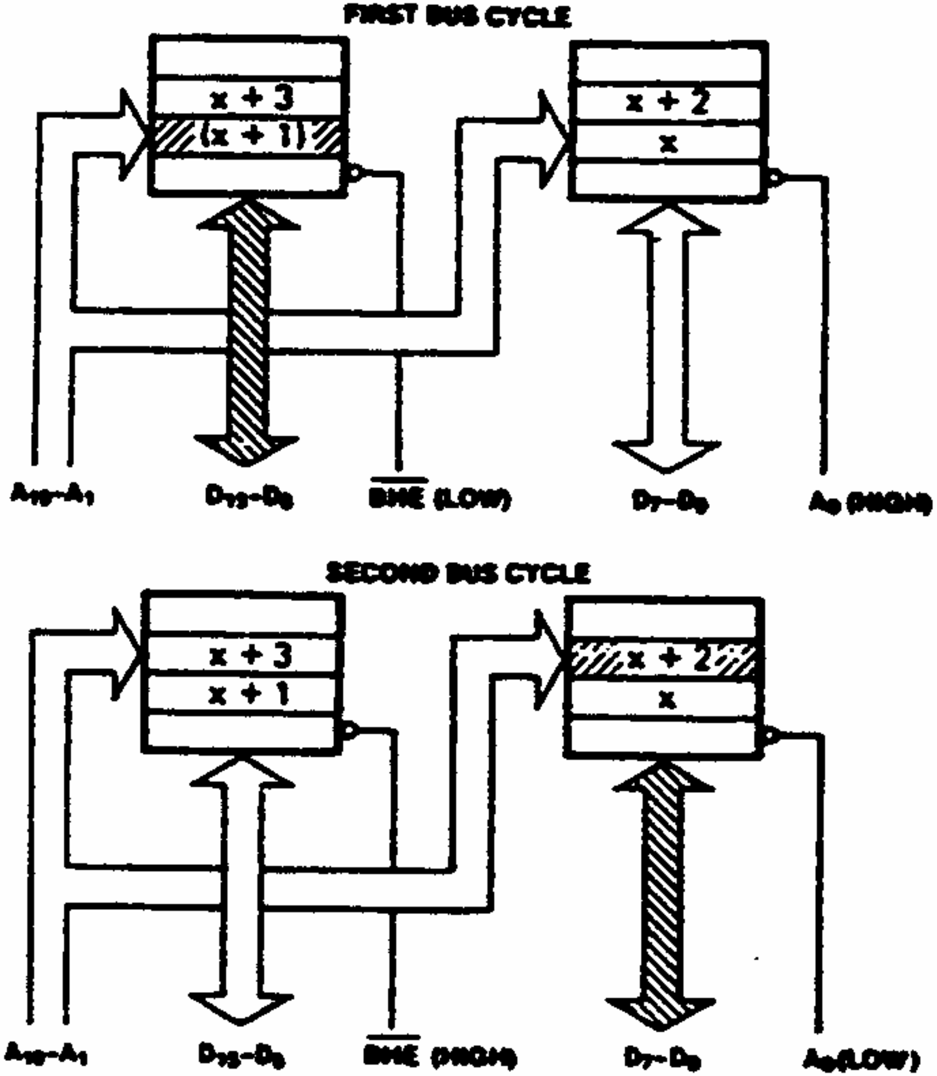
Figure 8–16 (a) Byte tmasfer by the 8088. (b) Word transfer by the 8088.


TRANSFER X
Y + 1
X + 1
Y
(X)
A_{19}–A_1
D_{15}–D_8
$\overline{BHE}$ (HIGH)
D_7–D_0
A_0 (LOW)
TRANSFER X + 1
Y + 1
(X + 1)
Y
X
A_{19}–A_1
D_{15}–D_8
$\overline{BHE}$ (LOW)
D_7–D_0
A_0 (HIGH)
TRANSFER X + 1, X
Y + 1
(X + 1)
Y
(X)
A_{19}–A_1
D_{15}–D_8
$\overline{BHE}$ (LOW)
D_7–D_0
A_0 (LOW)

(a)

(b)

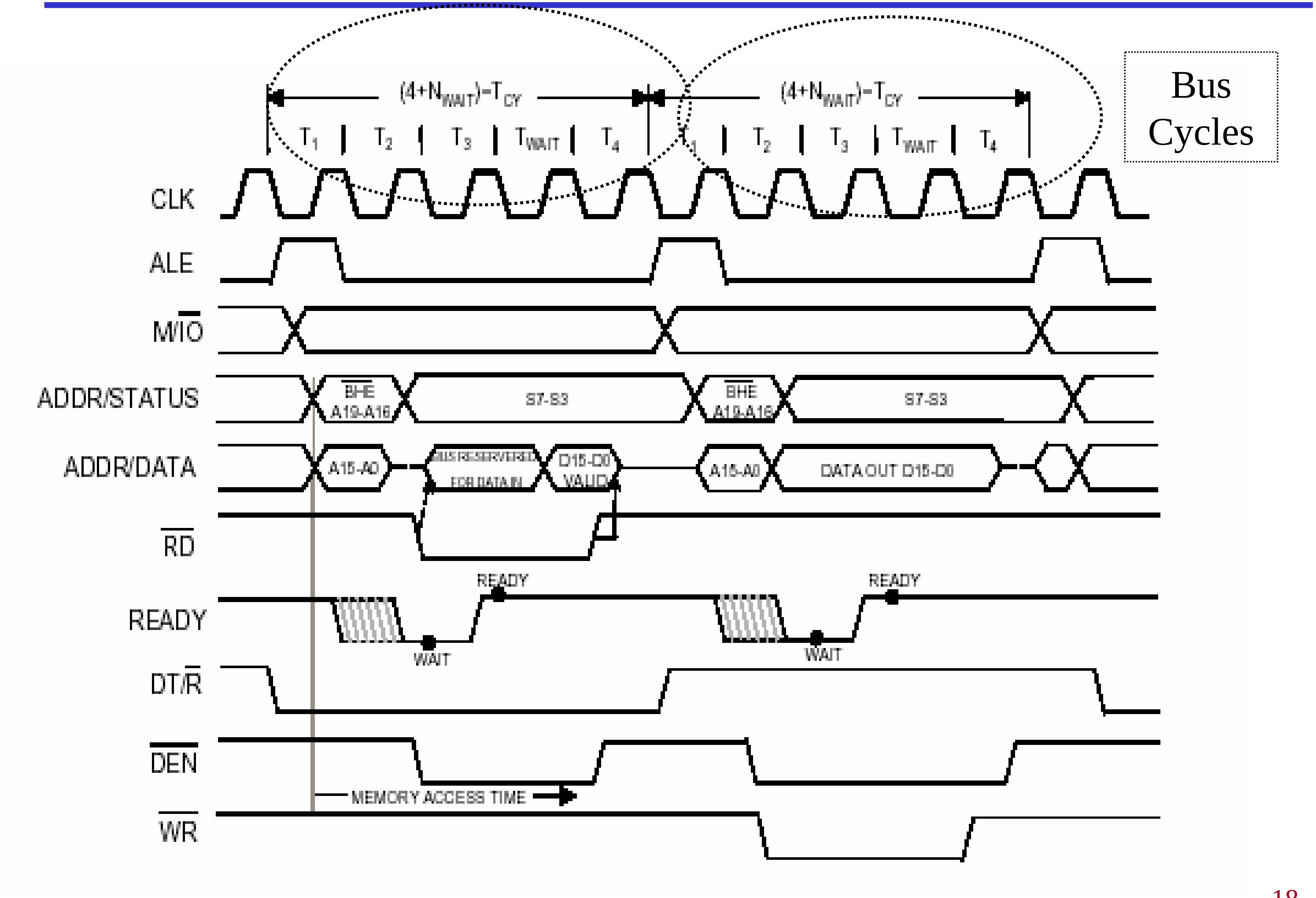
(c)


FIRST BUS CYCLE
x + 3
(x + 1)
x + 2
x
A19-A1
D15-D8
$\overline{BHE}$ (LOW)
D7-D0
A0 (HIGH)
SECOND BUS CYCLE
x + 3
x + 1
x + 2
x
A19-A1
D15-D8
$\overline{BHE}$ (HIGH)
D7-D0
A0(LOW)

Bus Cycle and Time States

- A bus cycle (machine cycle) defines the basic operation that a microprocessor performs to communicate with external devices
- Examples of bus cycles are memory read, memory write, input/output read and input/output write.
- A bus cycle corresponds to a sequence of events that starts with an address being output on the system bus followed by a read or write data transfer
- During these operations, a series of control signals are also produced by the MPU to control the direction and timing of the bus.
- Each bus cycle consists of at least four clock periods: T1, T2, T3, and T4.
- These clock periods are also called the T-States

Bus Cycle and Time States



Bus Cycles

Bus Cycle and Time States

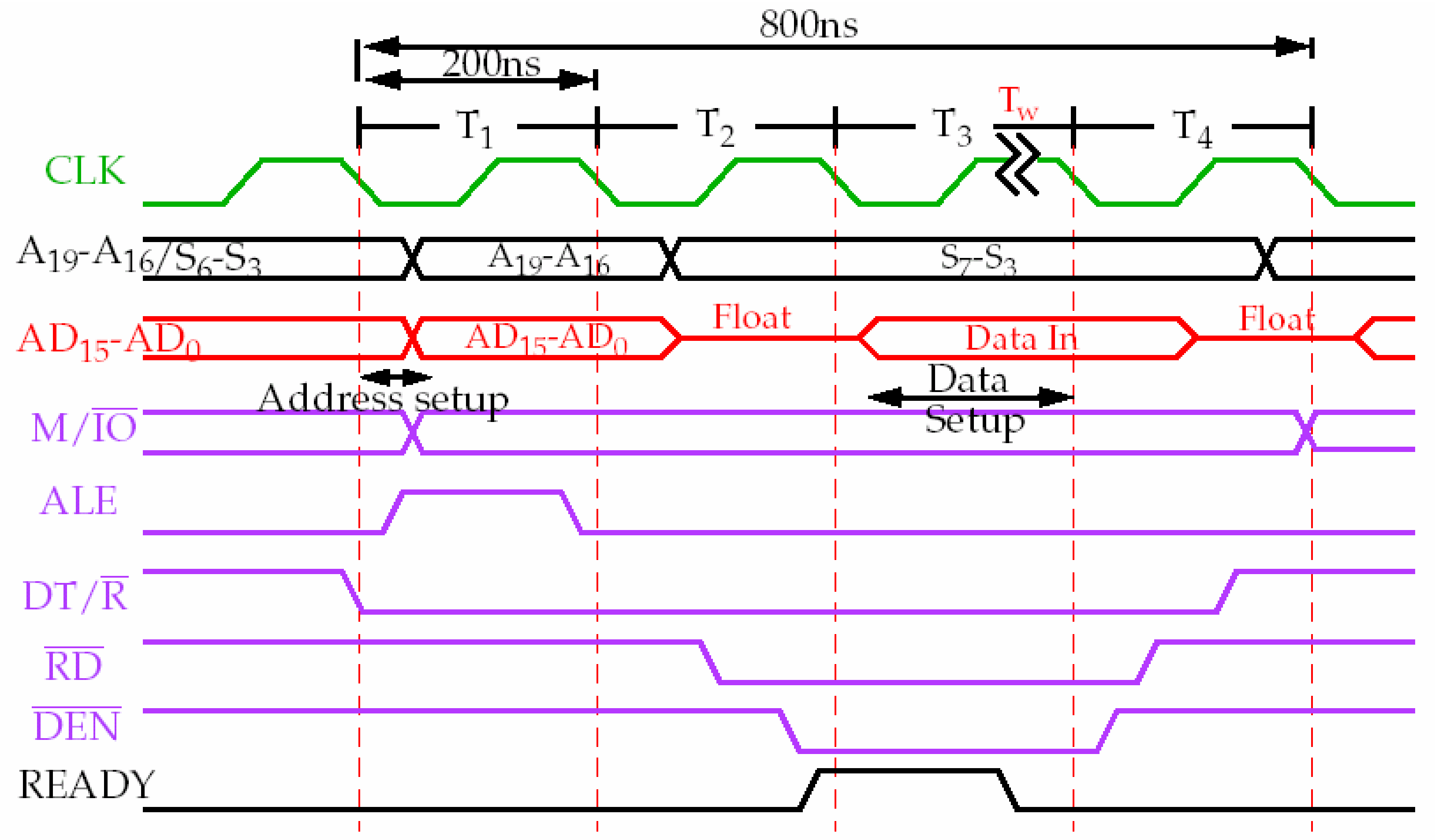
T1 - start of bus cycle. Actions include setting control signals (or S0-S2 status lines) to give the required values for ALE, DTR IO/M putting a valid address onto the address bus.

T2 - **the RD or WR control signals are issued, DEN is asserted and in the case of a write, data is put onto the data bus. The DEN turns on the data bus buffers to connect the CPU to the external data bus. The READY input to the CPU is sampled at the end of T2 and if READY is low, a wait state T_W (one or more) is inserted before T3 begins.**

T3 - this clock period is provided to allow memory to access the data. If the bus cycle is a read cycle, the data bus is sampled at the end of T3.

T4 - **all bus signals are deactivated in preparation for the next clock cycle. The 8088 also finishes sampling the data (in a read cycle) in this period. For the write cycle, the trailing edge of the WR signal transfers data to the memory or I/O, which activates and write when WR returns to logic 1 level.**

Read Cycle of the 8088

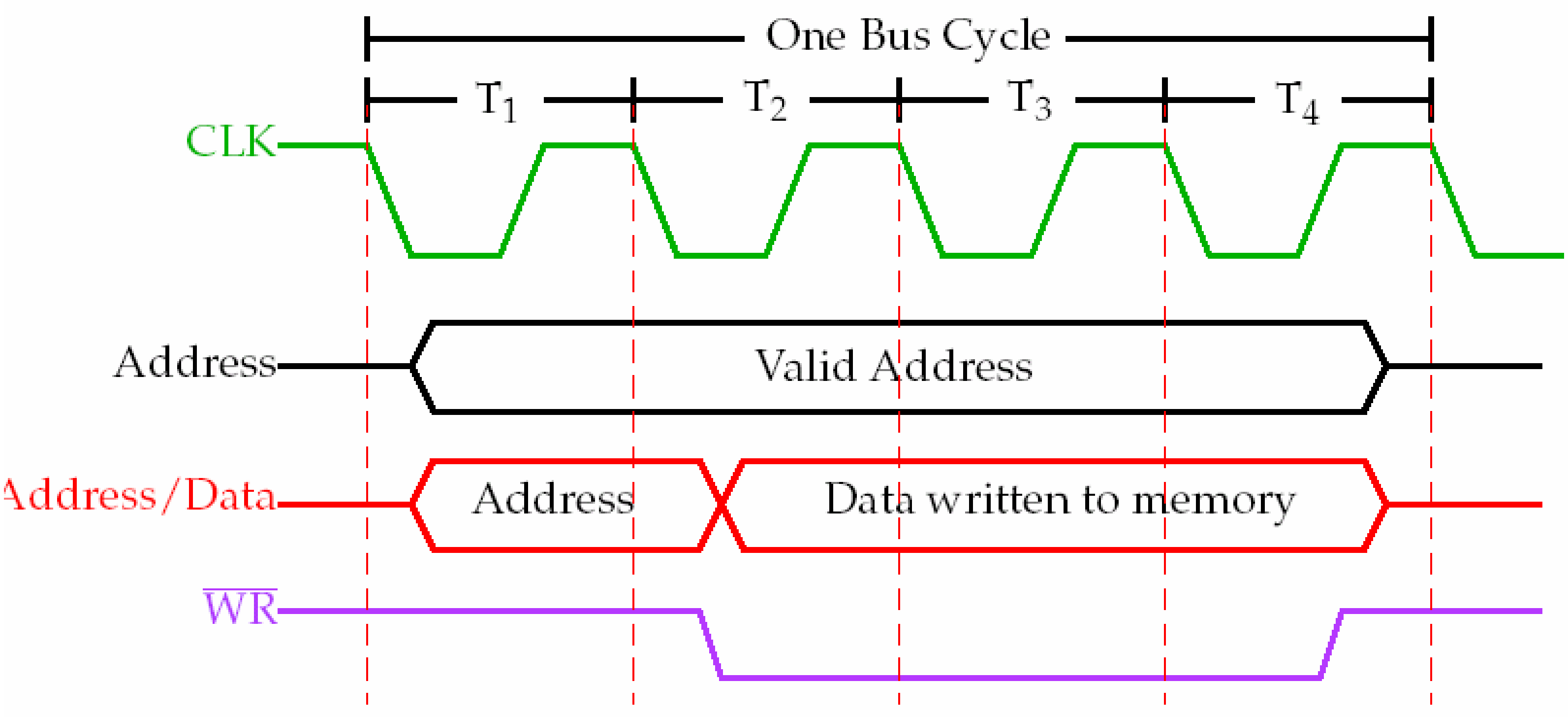


Bus Timing for a Read Operation

Read Cycle

- Each BUS CYCLE (machine cycle) on the 8086 equals **four** system clocking periods (T states).
- The clock rate is **5MHz**, therefore one Bus Cycle is *800ns*.
- Memory specs (memory access time) must match constraints of system timing.
- For example, bus timing for a read operation shows almost *600ns* are needed to read data.
- However, memory must access faster due to setup times, e.g.
- Address setup and data setup.
- This subtracts off about *150ns*.
- Therefore, memory must access in at least *450ns* minus another *30-40ns* guard band for buffers and decoders.
- *420ns* DRAM required for the 8086.

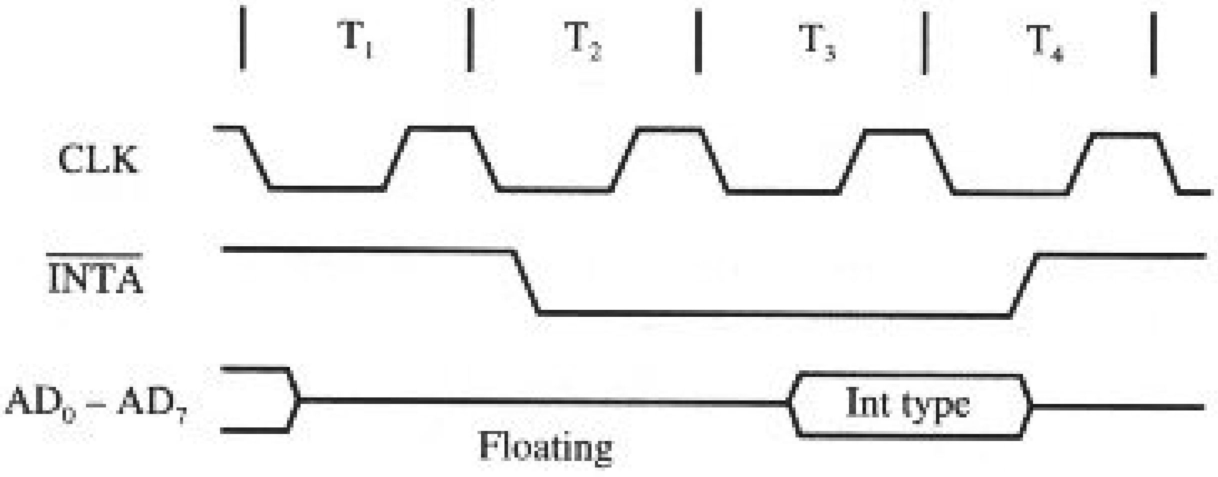
Write Cycle in 8088/8086 Minmode



Simplified 8086 Write Bus Cycle

Minimum Mode Interface

- Interrupt signals:
  - **INTR (Interrupt request)** :=1 shows there is a service request, sampled at the final clock cycle of each instruction acquisition cycle.
  - **INTA** : @T1 tri-states the Address Bus. Processor responds with two pulses going to 0 when it services the interrupt and waits for the interrupt service number after the second pulse.



(a) Minmode

Interrupt Signals cont’d

Interrupt	Logic	Disabled by SW?	Priority
NMI	Rising Edge	No	High
INTR	High	Yes	Low

–**NMI (Nonmaskable interrupt) :** A leading edge transition causes the processor go to the interrupt routine after the current instruction is executed.

–**$\overline{\text{TEST}}$**: Processor suspends operation when=1. Resumes operation when=0. Used to syncronize the processor to external events. (All 8087-capable compilers and assemblers automatically generate a **WAIT** instruction before each coprocessor instruction. The **WAIT** instruction tests the CPU's **TEST** pin and suspends execution until its input becomes "**LOW**". When **TEST=0, WAIT** instruction is like **NOP**.

In all 8086/8087 systems, the 8086 /TEST pin is connected to the 8087 BUSY pin. As long as the EU executes a coprocessor instruction, it forces its BUSY pin "HIGH"; thus, the WAIT opcode preceding the coprocessor instruction stops the CPU until any still-executing coprocessor instruction has finished)

–**RESET** : =0. Need at least 4 clock cycles. Issuing reset causes the processor to fetch the first instruction from the memory FFFF:0000h.

Minimum Mode Interface

- DMA (Direct Memory Access) Interface Signals:
  - HOLD: External device puts logic level 1 to HOLD input to take control of the bus for DMA request. (sampled at every rising edge of the CLK)
  - HLDA (Hold acknowledge) : Processor responds by putting logic level 1 to HLDA. (at the end of T4)
  - In this state; Address and Data lines, SSO, IO/$\overline{M}$, DT/$\overline{R}$, $\overline{RD}$, $\overline{WR}$, $\overline{DEN}$ signals are all put to high-Z state

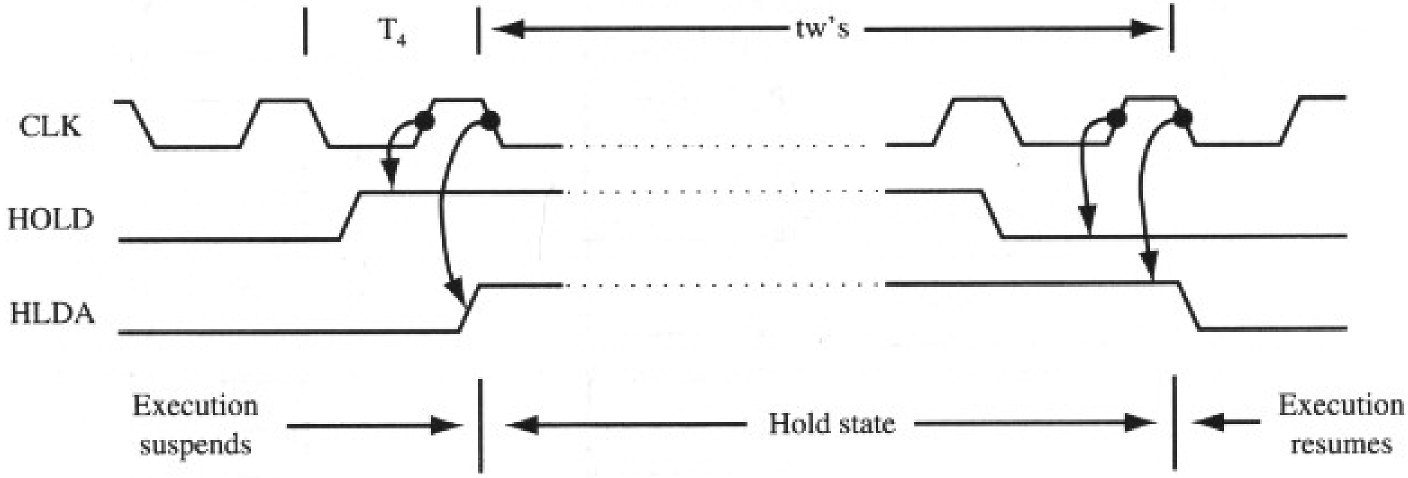


FIGURE 10.14 HOLD timing

Minimum Mode Interface

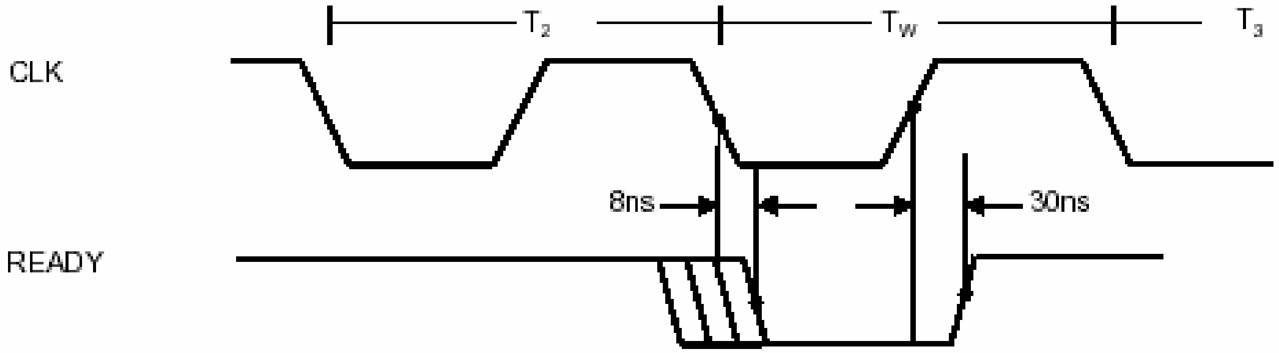
- READY Control line:
  - –can be used to insert wait states into the bus cycle so that it is extended by a number of clock periods.

✓If the access time for a memory device is longer than the memory access time calculated, need to give extra clock periods, wait state Tw, for memory.

✓The READY input is sampled at the end of T2 and again, if applicable, in the middle of Tw. If READY is a logic 0 on 1-to-0 clock transition, then Tw is inserted between T2 and T3. And will check for logic 1 on 0-to-1 clock transition in the middle of Tw to see if it shall go back T3.

✓During the wait state, signals on the buses remain the same as they were at the start of the WAIT state.

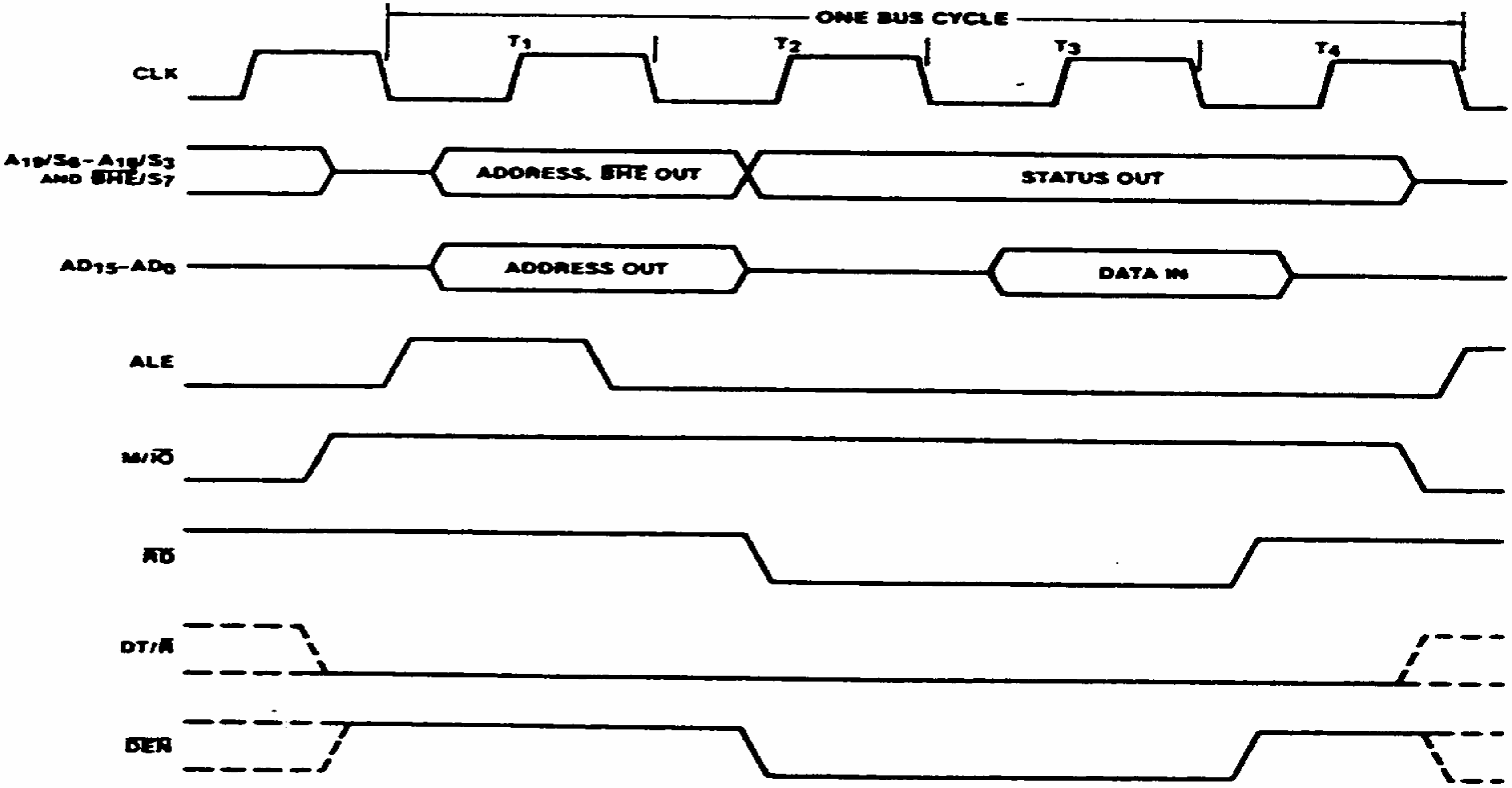
✓By having the WAIT state, slow memory and devices has at least one more cycle (200ns for 5 MHz 8088) to get its data output.



(a) 8088/86 READY Input Timing

Read Cycle of the 8086 - minimum mode

- ➢BHE is output along with the address during T1
- ➢Data can be read during T3 over all 16 data bus lines
- ➢M/IO replaces IO/M
- ➢SSO status signal is not produced



(a)

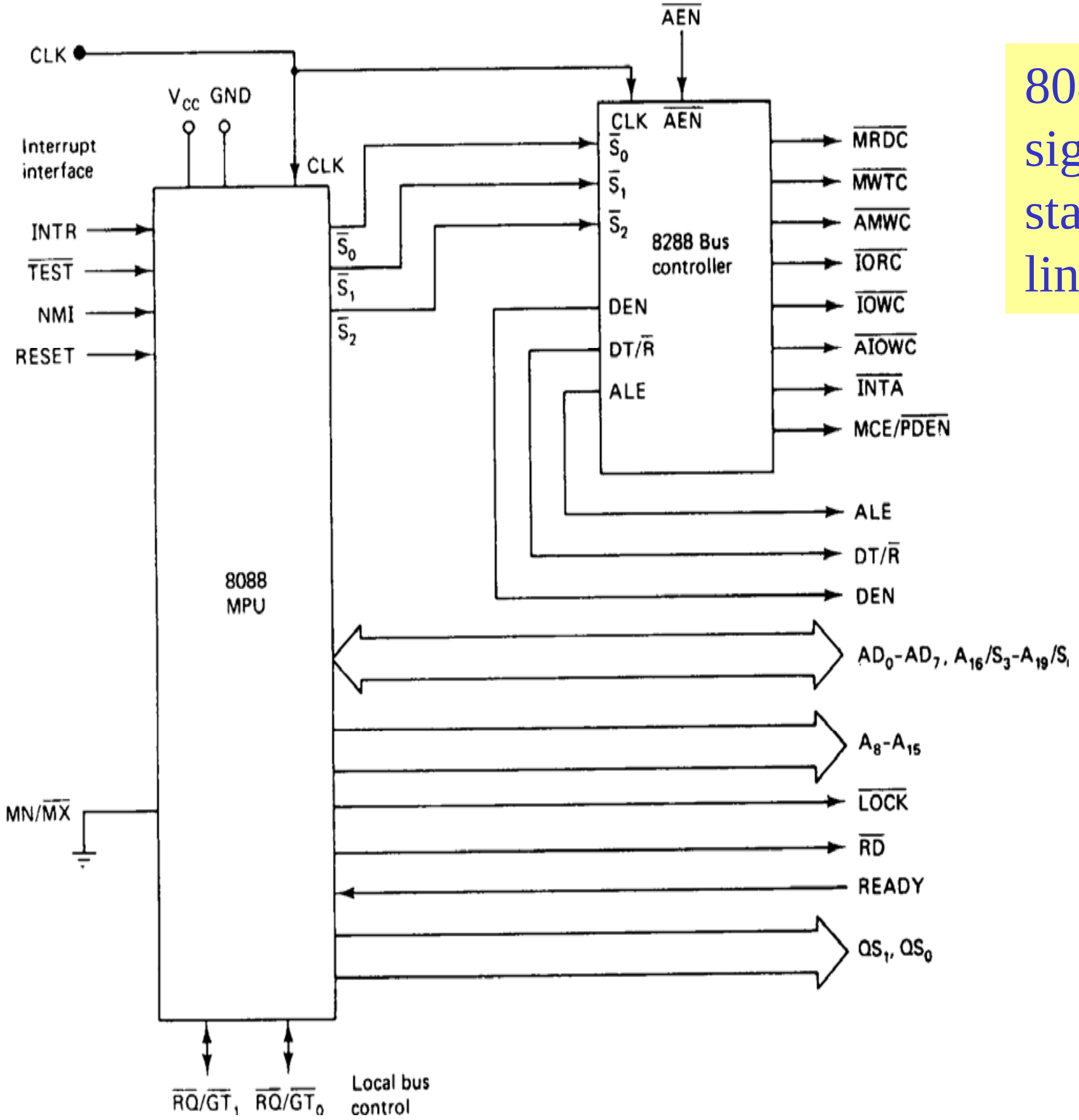
Maximum Mode Interface

- Used in a multiprocessor environment
- 8288 Bus Controller is used for bus control
- WR¯,IO/M¯,DT/R¯,DEN¯,ALE, INTA¯ signals are not readily available
- Instead following generated from S0,S1 and S2:
  - **MRDC¯** (memory read command)
  - **MWRT¯** (memory write command)
  - **AMWC¯** (*advanced* memory write command)
  - **IORC¯** (I/O read command)
  - **IOWC¯** (I/O write command)
  - **AIOWC¯** (*Advanced* I/O write command)
  - **INTA¯** (interrupt acknowledge)

Maximum Mode Interface

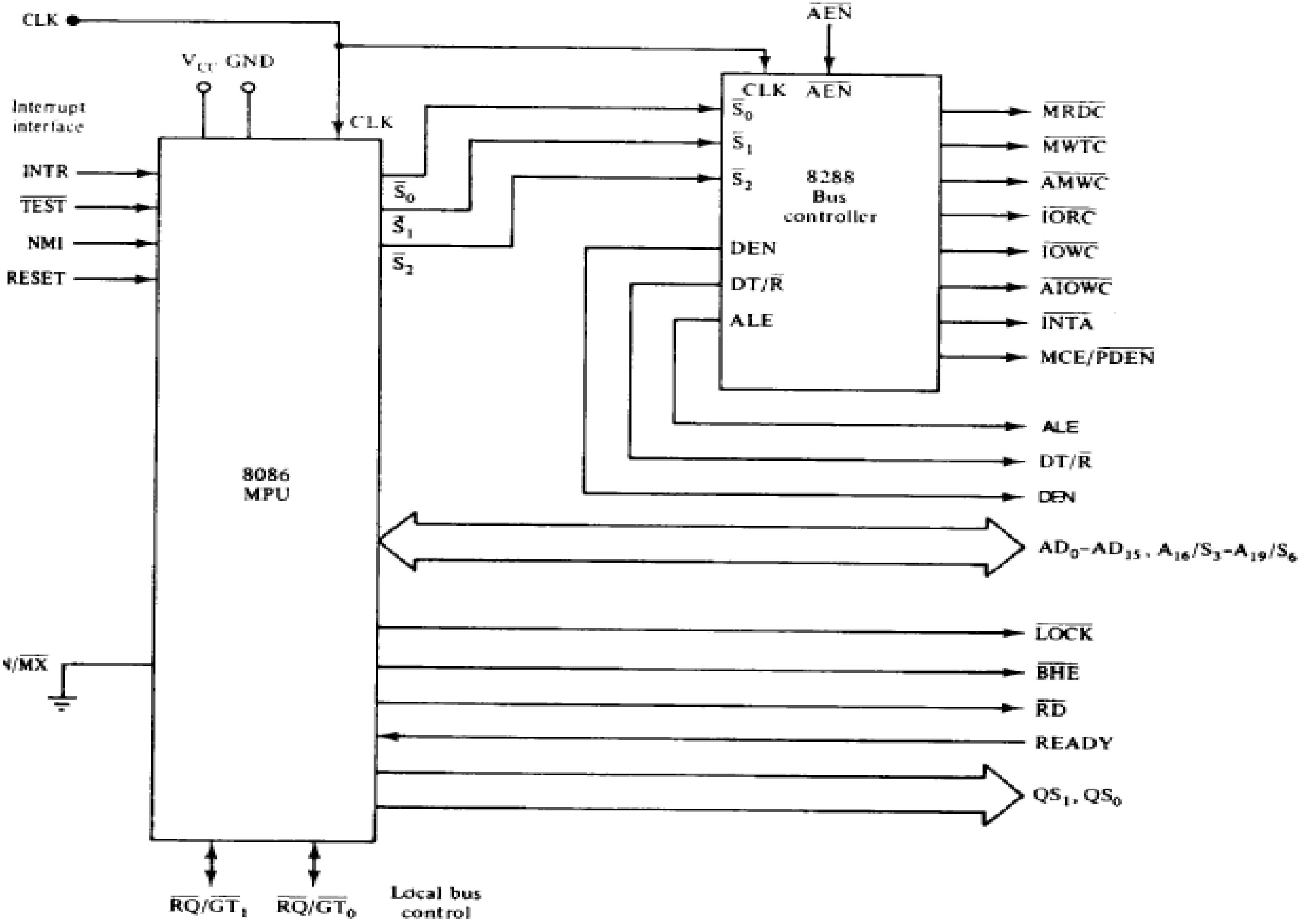
- **DEN, DT/R¯** and **ALE** signals are the same as minimum-mode systems
- **LOCK¯:** when =0, prevents other processors from using the bus
- **QS$_0$ and QS$_1$ (queue status signals)** : informs about the status of the queue
- **RQ¯/GT ¯$_0$** and **RQ¯/GT ¯$_1$** are used instead of HOLD and HLDA lines in a multiprocessor environment as request/grant lines. Bidirectional!

Maximum-mode interface circuit diagram (8088)



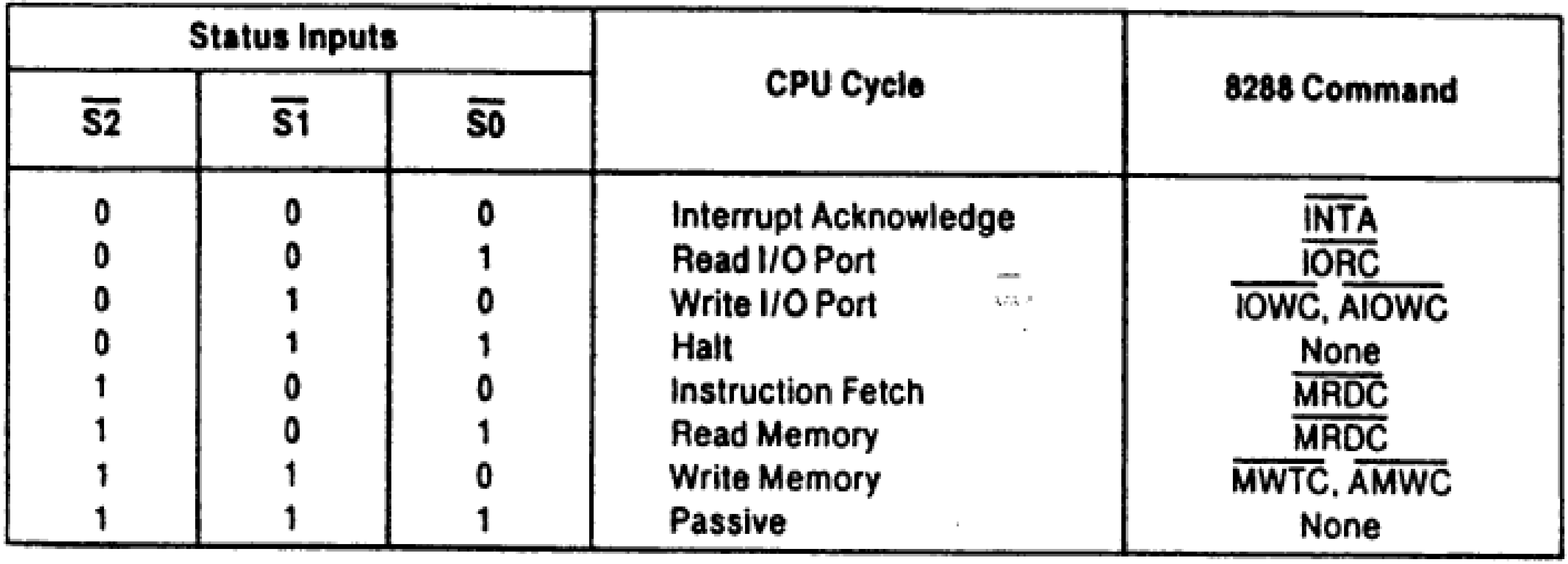
8088 does not provide all the signals, instead it outputs a status code on three signal lines S0, S1, and S2

Maximum-mode interface circuit diagram (8086)



Status Bits

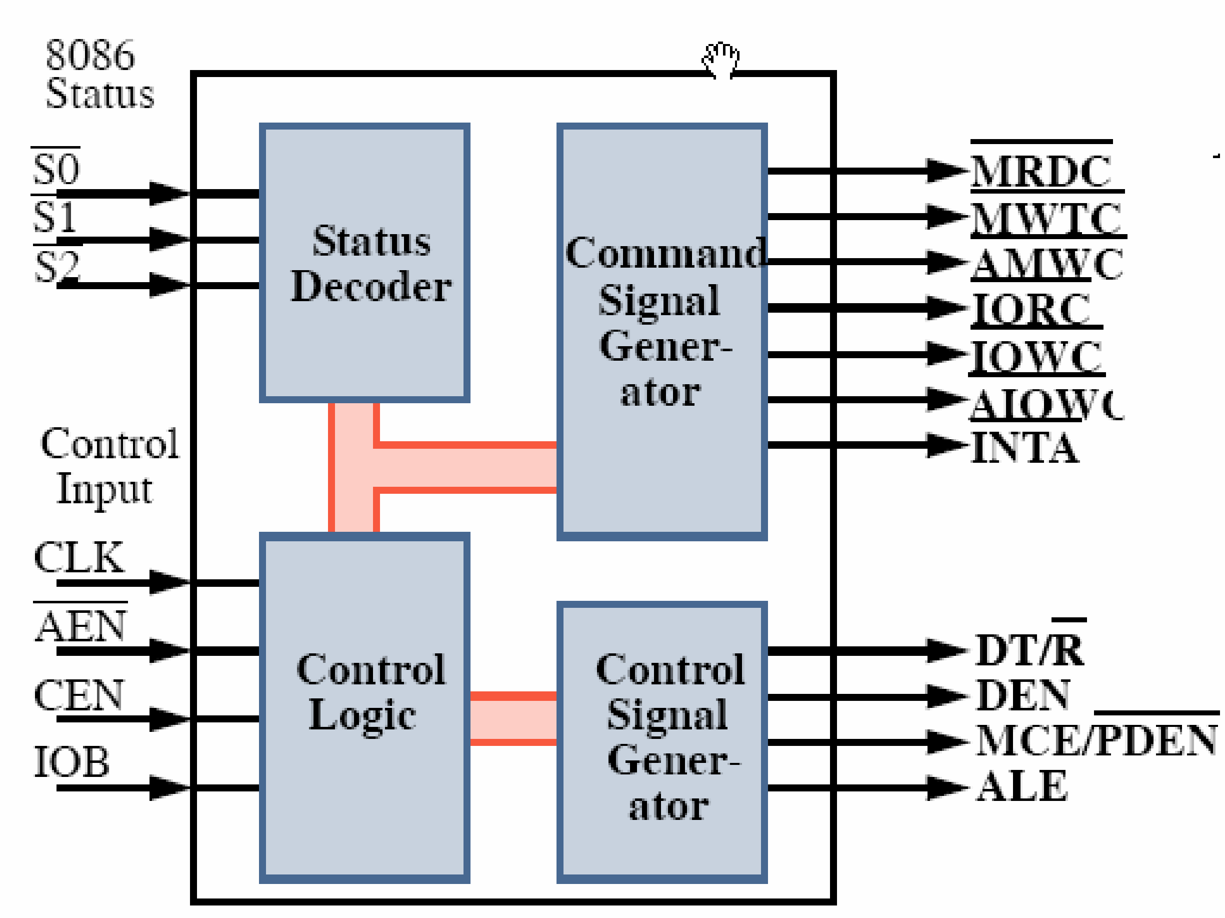
- Indicate the function of the current bus cycle.
- They are normally decoded by the 8288 bus controller



Status Inputs			CPU Cycle	8288 Command
$\overline{S2}$	$\overline{S1}$	$\overline{S0}$		
0	0	0	Interrupt Acknowledge	$\overline{INTA}$
0	0	1	Read I/O Port	$\overline{IORC}$
0	1	0	Write I/O Port	$\overline{IOWC}$, $\overline{AIOWC}$
0	1	1	Halt	None
1	0	0	Instruction Fetch	$\overline{MRDC}$
1	0	1	Read Memory	$\overline{MRDC}$
1	1	0	Write Memory	$\overline{MWTC}$, $\overline{AMWC}$
1	1	1	Passive	None

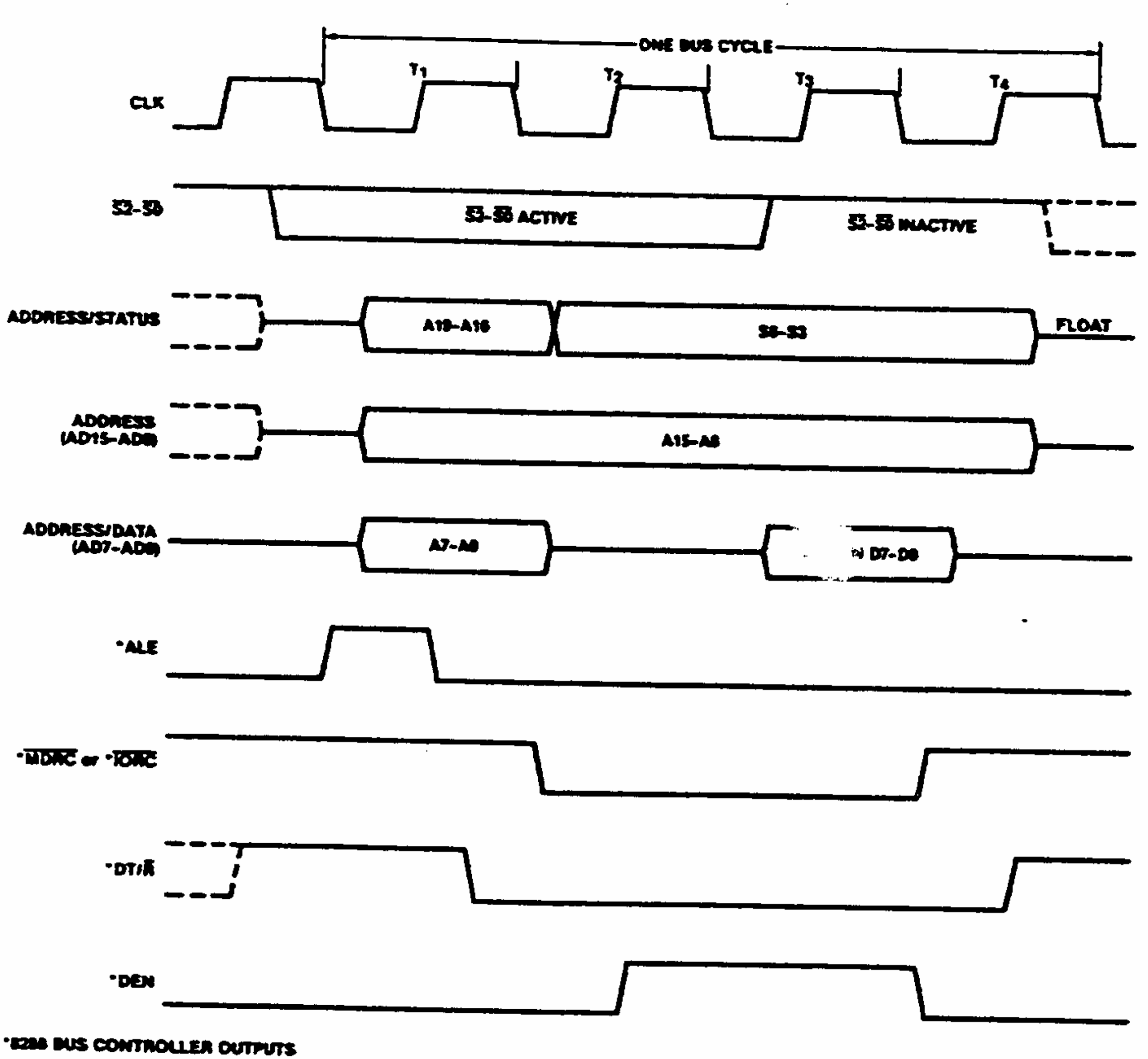
- The signals shown above are produced by 8288 depending on the state of S_0, S_1 and S_2.

Maximum Mode Interface



Block Diagram of the 8288

Read Cycle of the 8086 - maximum mode



*8288 BUS CONTROLLER OUTPUTS

(b)

Maxmode operation

$\overline{RQ}/\overline{GT}_0$ and $\overline{RQ}/\overline{GT}_1$:are request grant signals used by other devices called bus masters to take over the 8088’s system bus to read or write into the memory (such as 8087 coprocessor)

Sequence of operations:

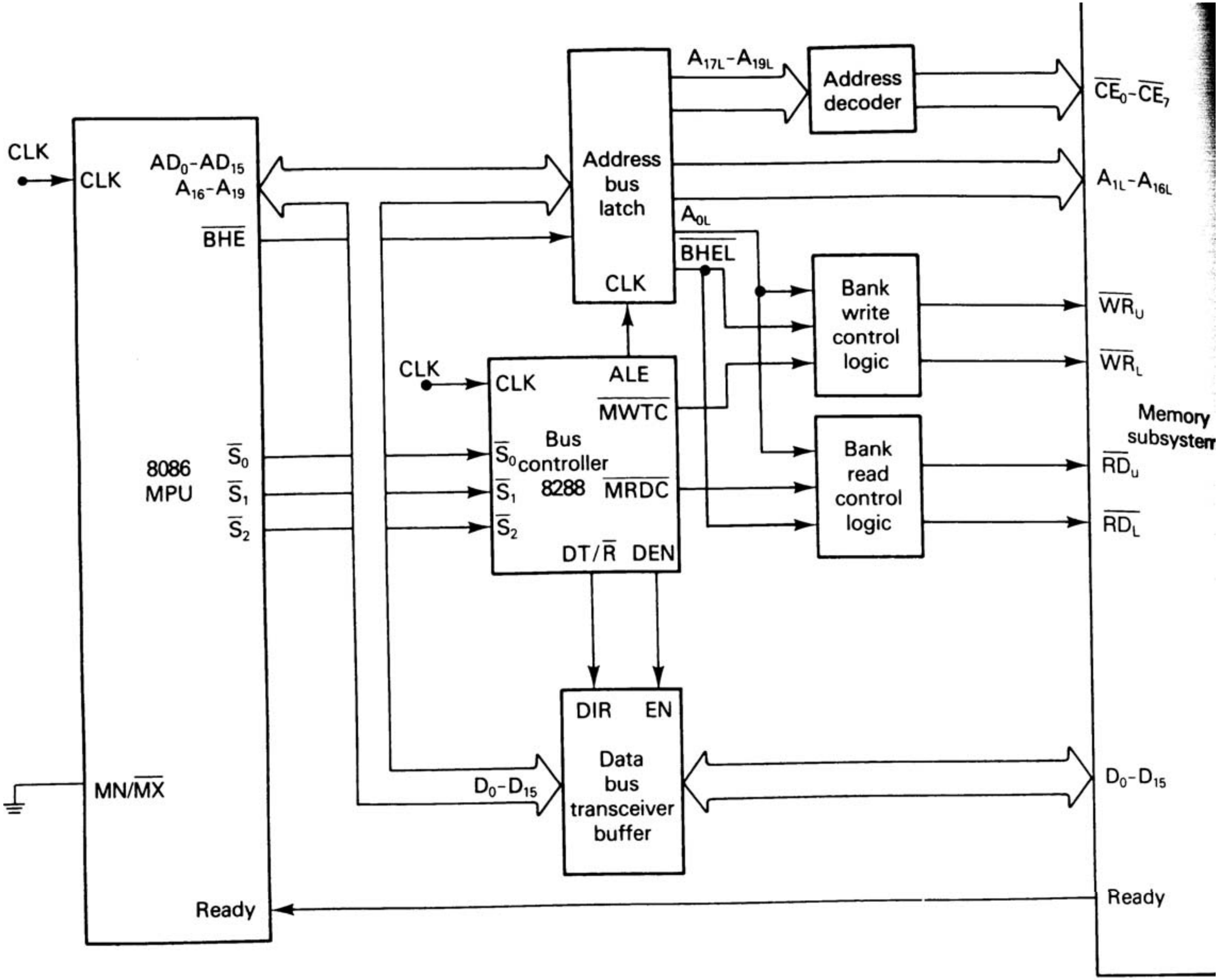
1. **When another device decides to take over the system bus, it will pull $\overline{RQ}/\overline{GT}_0$ low for one clock period.**
2. **When the 8088 ready to release the system bus, it will use $\overline{RQ}/\overline{GT}_0$ as an output to inform the new bus master. A second low level one period pulse does this.**
3. **The 8088 enters a Hold ACK state until the new bus master is ready to give the bus back by sending a third low level one period pulse as an input.**

Maxmode operation

QS0 and QS1 (queue status) pins provide information on the 8088 internal instruction queue. The information is used by the 8087 coprocessor. The queue status bits indicate the contents of the internal instruction queue according to the following table:

QS1	QS0	instruction queue contents
0	0	No operation (queue is idle)
0	1	First byte of an opcode
1	0	Queue is empty
1	1	Subsequent byte of an opcode

Memory Interface - maxmode



8088-8087 connection

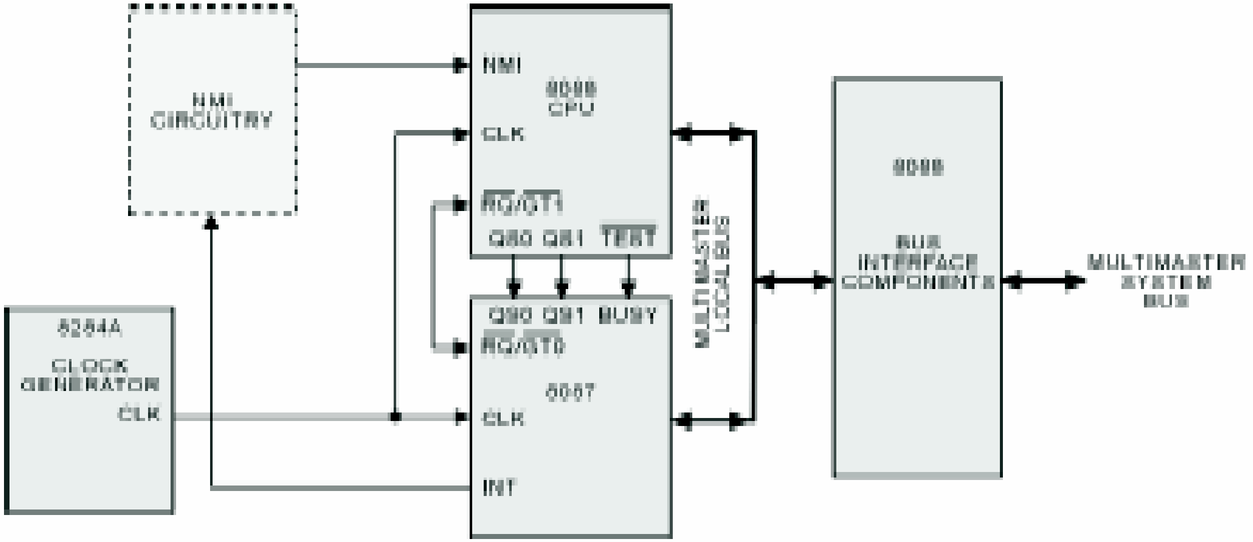


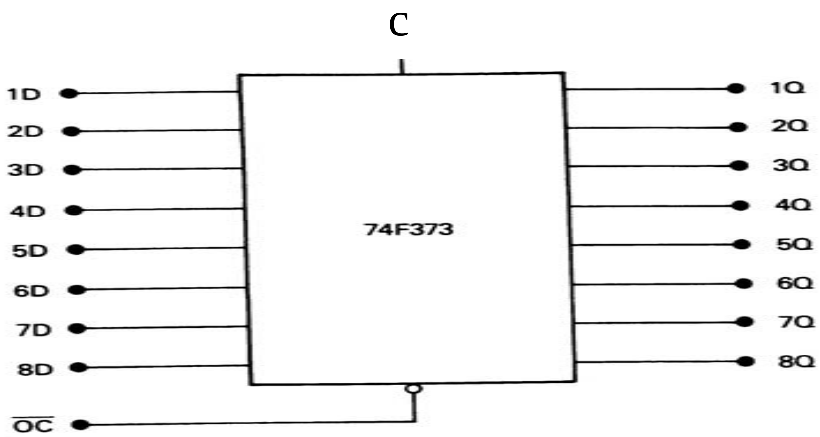
Figure 3: 8088/8087 Connection

✓The 8087 updates its internal instruction queue by monitoring the 8088's QS0-QS1 queue status lines.

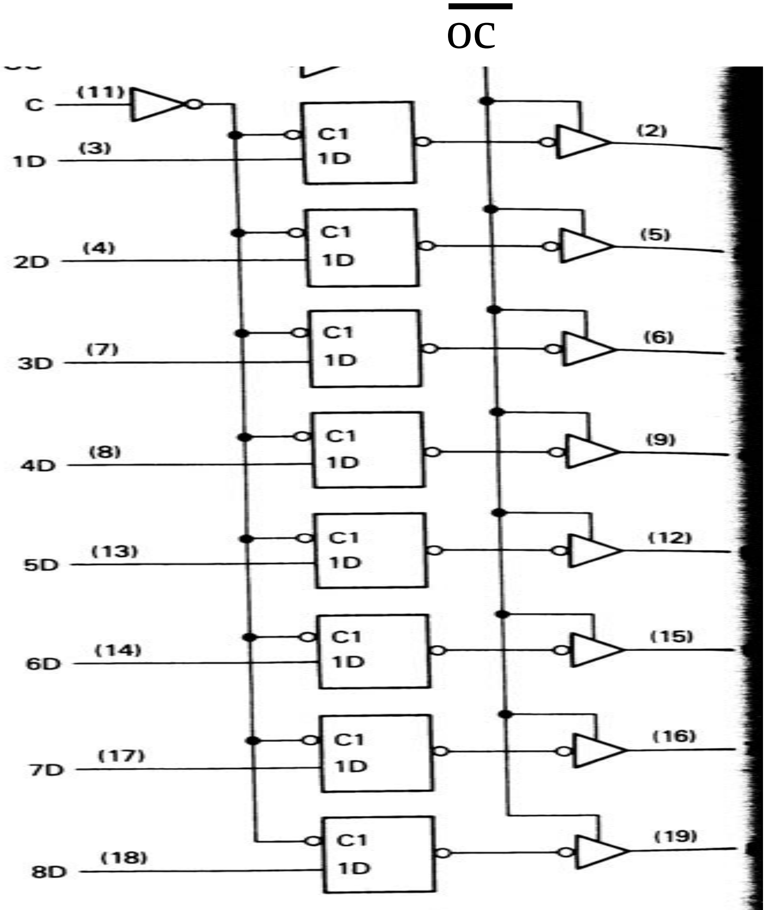
✓In this way, the 8087 maintains an instruction queue that is identical to the one in the 8088.

✓When the 8087 assumes control of the system's buses, it controls the 8288 bus controller through its own S0–S2 status lines in the same manner as the 8088 does during normal operations.

Address Bus Latches and Buffers



(a)

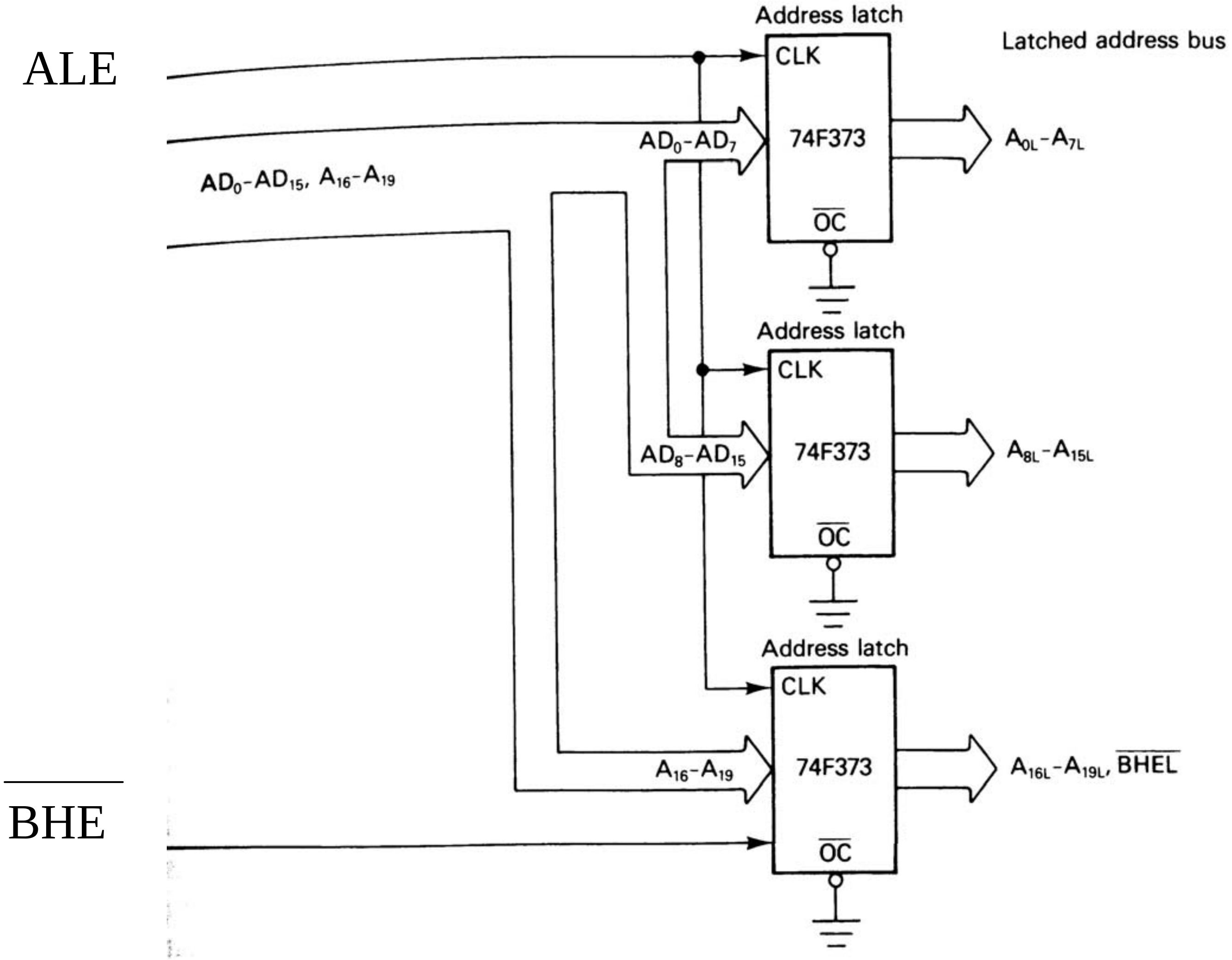


(b)

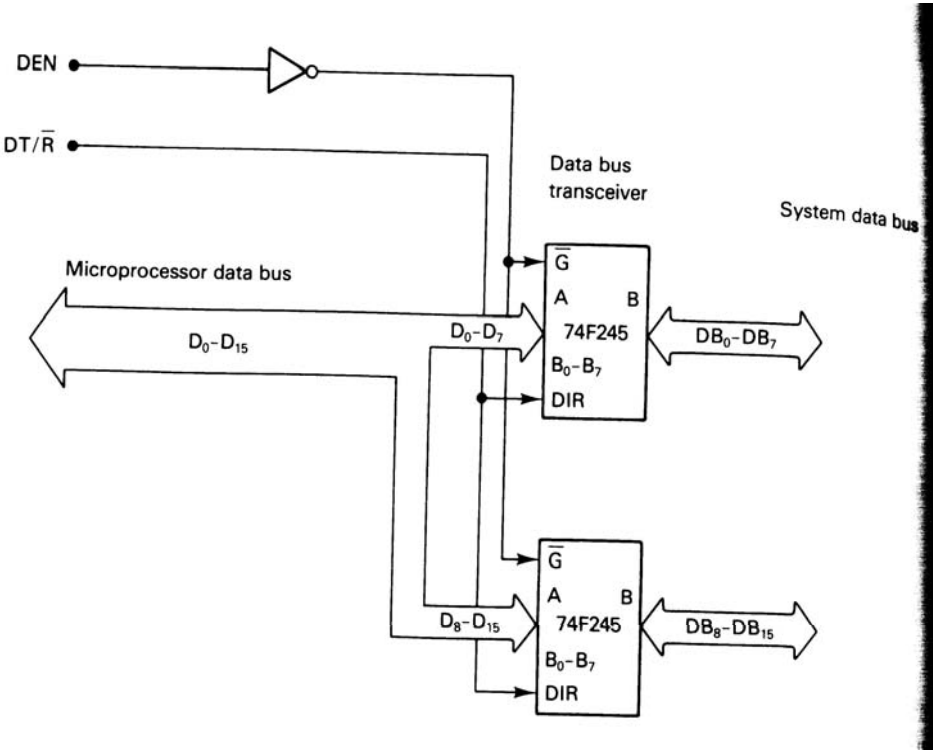
Inputs			Output
$\overline{OC}$	Enable C	D	Q
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

(c)

Address Latch Circuit



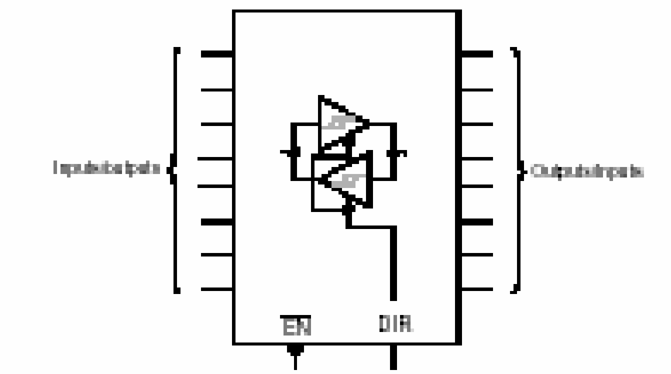
Data Bus Transceiver Circuit



GBAR ENABLE	DIR	OPERATION
L	L	B data to A
L	H	A data to B
H	x	Isolation

Buffered Systems

- Buffering (boosting) of the control, data, and address busses to provide sufficiently strong signals to drive various IC chips
  - When a pulse leaves an IC chip it can lose some of its strength depending on how far away the receiving IC is located
  - Plus the more pins a signal is connected to (i.e., fanout) the stronger the signal must be to drive them all which requires bus buffering
  - bus buffering = boosting the signals travelling on the busses
  - unidirectional bus 74LS244
  - bidirectional bus 74LS245



8088 System

9–3 BUS BUFFERING AND LATCHING 315

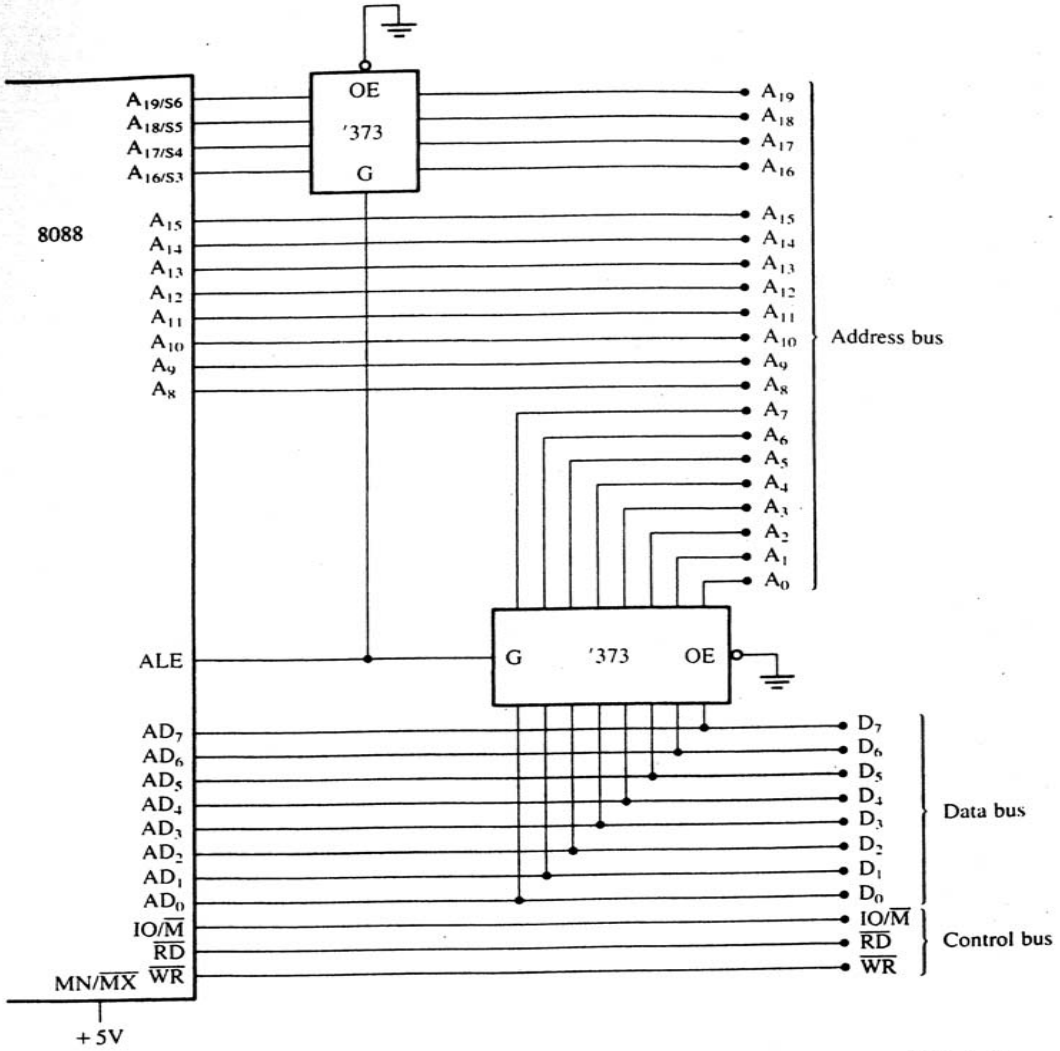
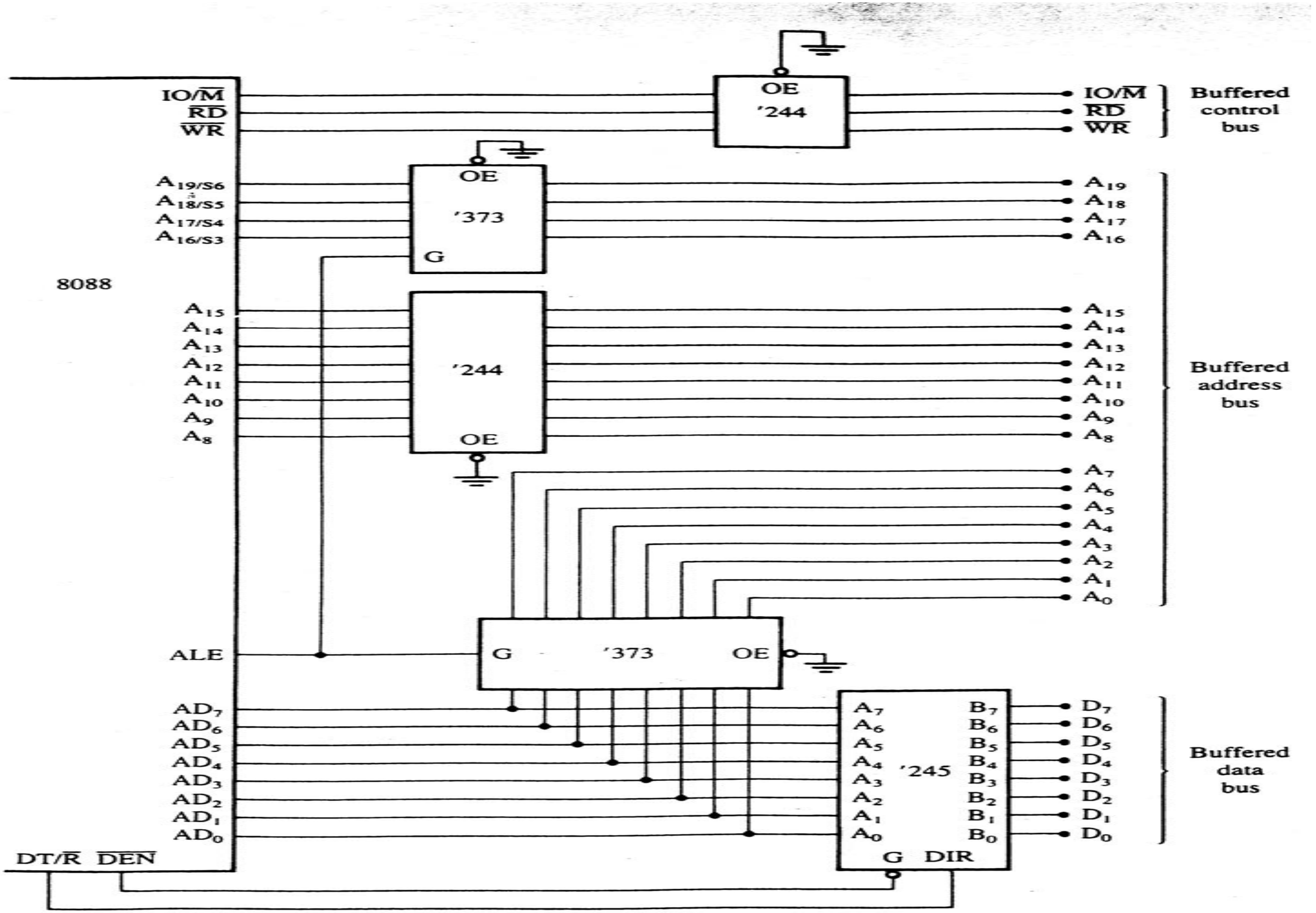
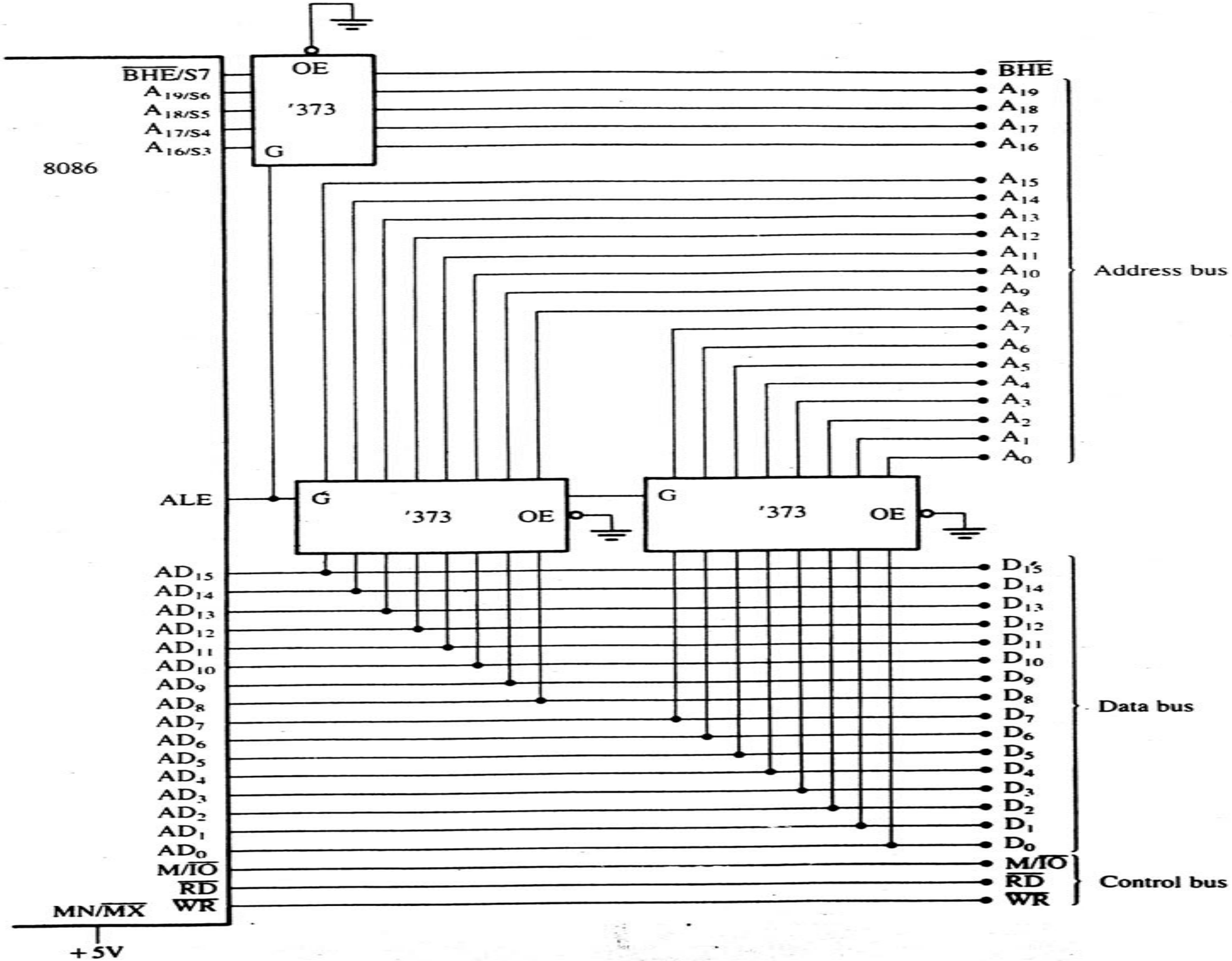


FIGURE 9–5 The 8088 microprocessor shown with a demultiplexed address bus. This is the model used to build many 8088-based systems.

Fully buffered 8088

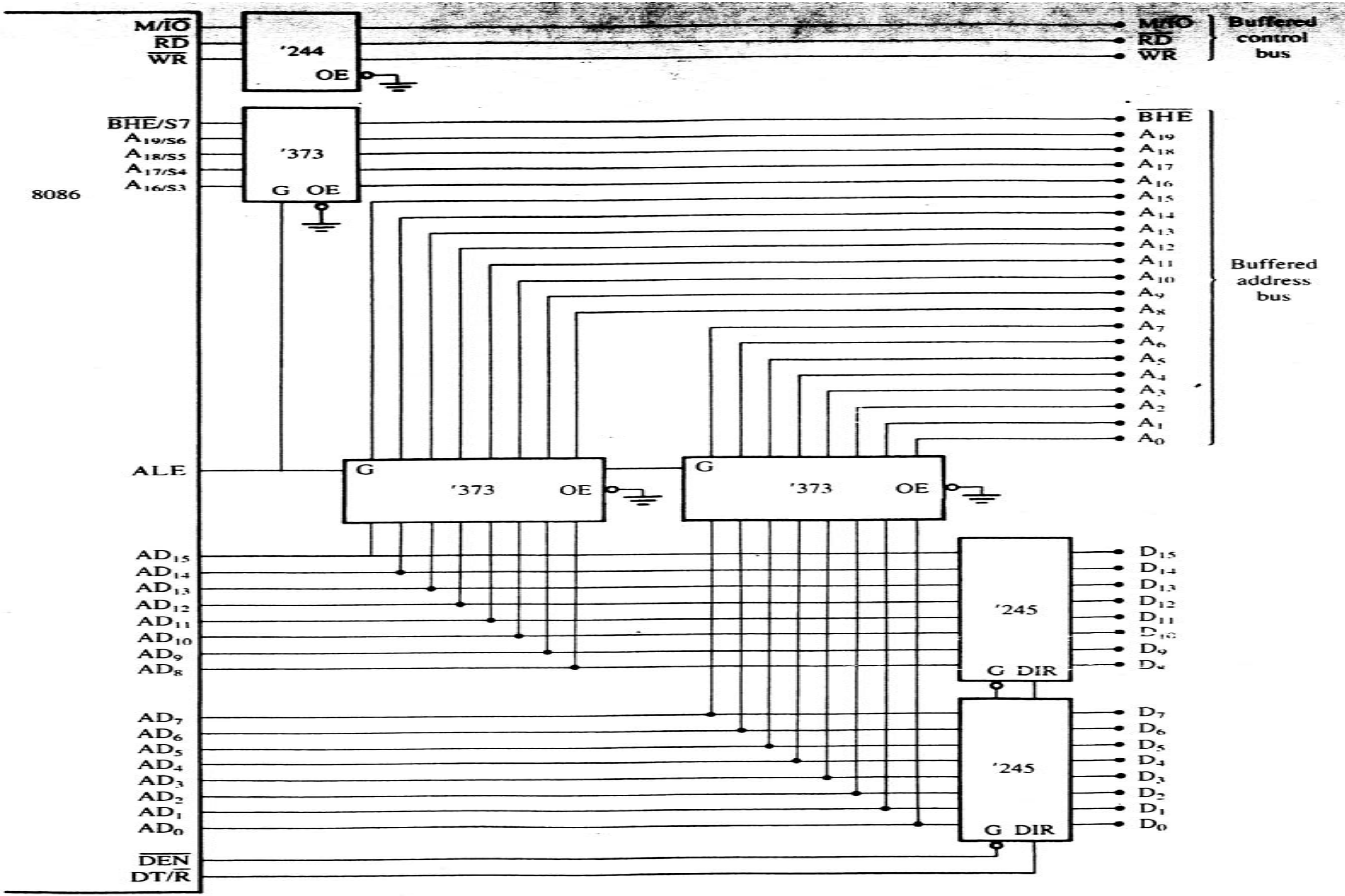


8086 System

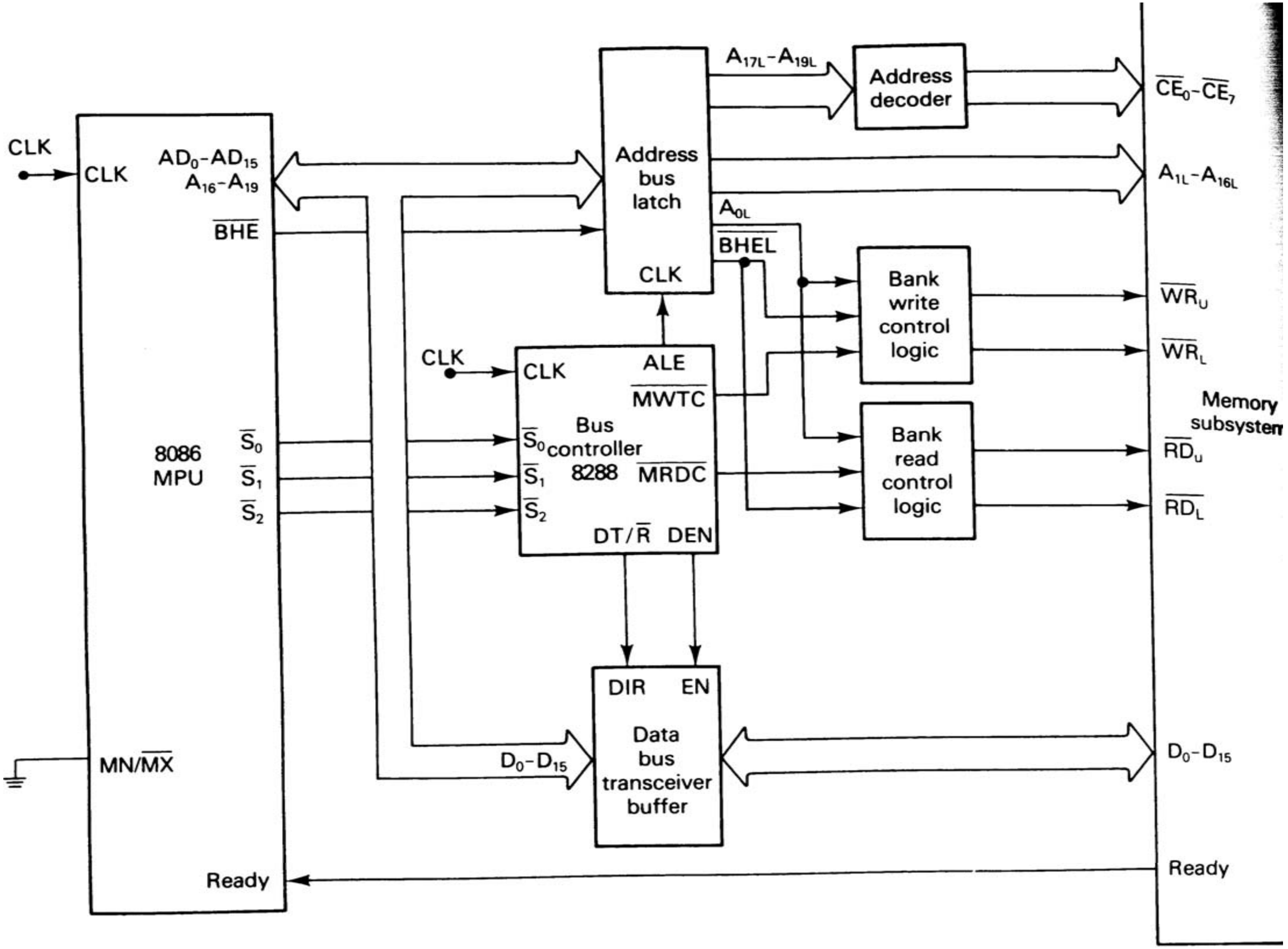


8086
BHE/S7
A19/S6
A18/S5
A17/S4
A16/S3
OE
'373
G
ALE
'373
OE
G
'373
OE
AD15
AD14
AD13
AD12
AD11
AD10
AD9
AD8
AD7
AD6
AD5
AD4
AD3
AD2
AD1
AD0
M/IO
RD
WR
MN/MX
+5V
BHE
A19
A18
A17
A16
A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
Address bus
D15
D14
D13
D12
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
Data bus
M/IO
RD
WR
Control bus

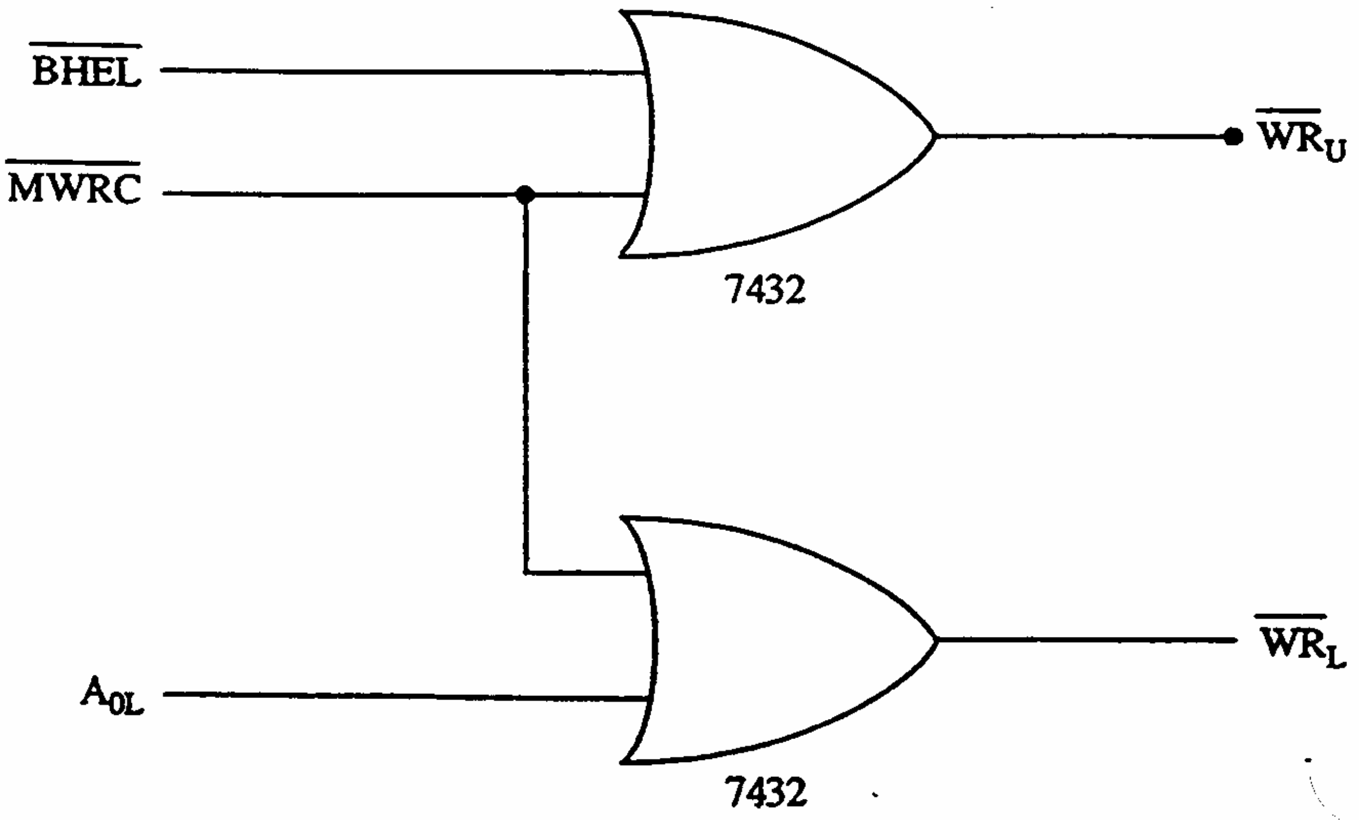
Fully Buffered 8086



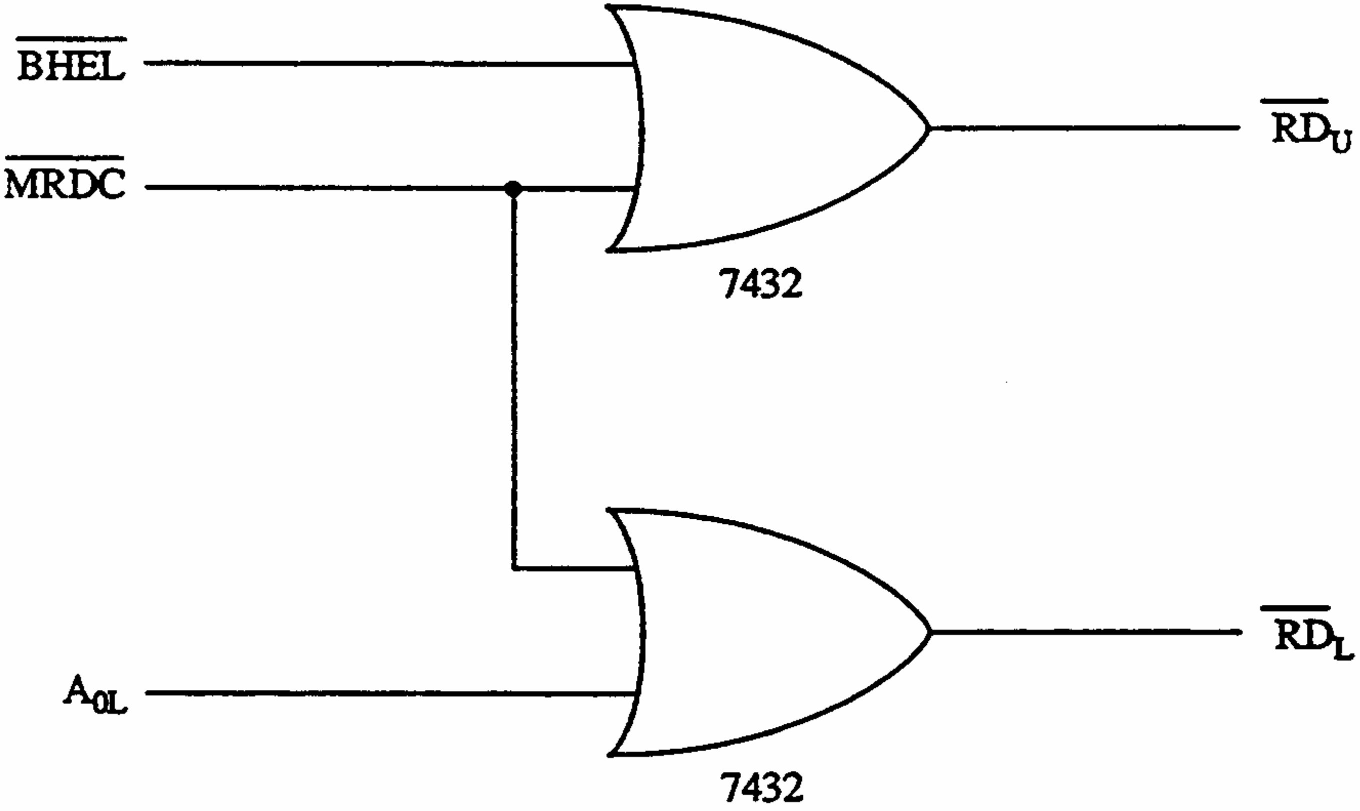
Memory Interface



Bank Write Control Logic



Bank Read Control Logic


$\overline{BHEL}$
$\overline{MRDC}$
A_{0L}
7432
7432
$\overline{RD_U}$
$\overline{RD_L}$