2.3 INTRODUCTION TO MULTIPROGRAMMING

8086 and 8088 can be configured in two modes of operation, the minimum mode and the maximum mode. The minimum mode is used for a small system with a single processor, a system in which the 8086/8088 generates all the necessary bus control signals directly (thereby minimizing the required bus control logic). The maximum mode is for medium-size to large systems, which often include two or more processors

MULTIPROCESSOR SYSTEMS

Multiprocessor Systems refer to the use of multiple processors that execute instructions simultaneously and communicate using mailboxes and semaphores

Maximum mode of 8086 is designed to implement 3 basic multiprocessor configurations:

1. Coprocessor(8087)
2. Closely coupled (dedicated I/O processor:8089)
3. Loosely coupled (Multibus)

Coprocessors and closely coupled configurations are similar - both the CPU and the external processor share:

- Memory
- I/O system
- Bus & bus control logic Clock generator

COPROCESSOR CONFIGURATION

WAIT instruction allows the processor to synchronize itself with external hardware, eg., waiting for 8087 math co-processor. When the CPU executes WAIT state, TEST input is asserted (low), the waiting state is completed and execution will resume.

ESC instruction:

ESC opcode, operand,

opcode: Immediate value recognizable to a coprocessor as an instruction opcode

Operand:Name of a register or a memory address (in any mode)

When the CPU executes the ESC instruction, the processor accesses the memory operand by placing the address on the address bus.If a coprocessor is configured to share the system bus, it will recognize the ESCinstruction and therefore will get the opcode and the operand

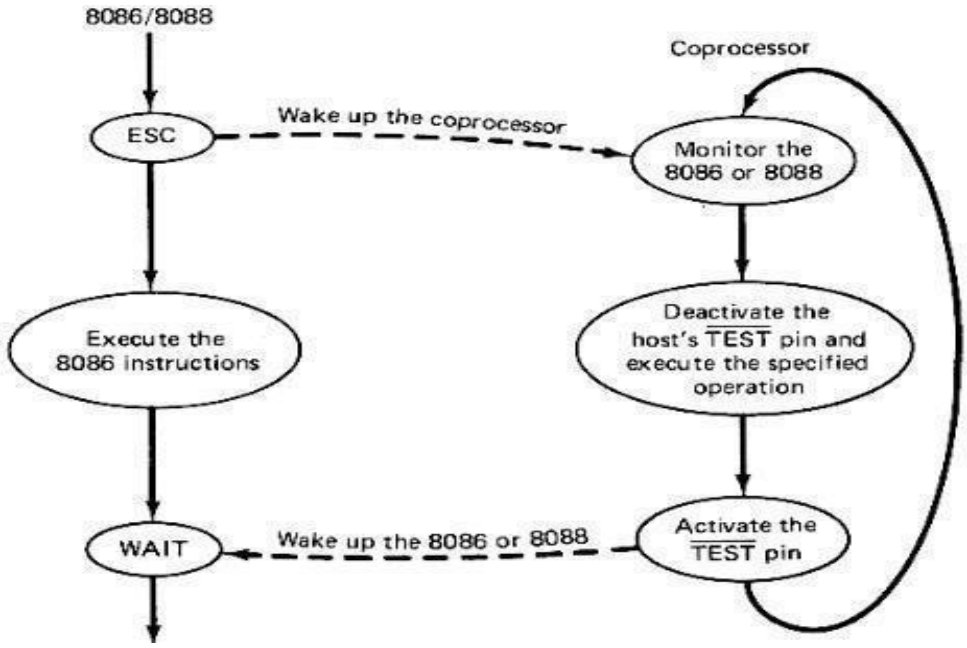

Figure 2.3.1 Synchronisation between the 8086 and its coprocesssor

[Source: "Microcomputer Systems: The 8086 / 8088 Family - Architecture, Programming and Design" by Yu-Cheng Liu, Glenn A.Gibson]

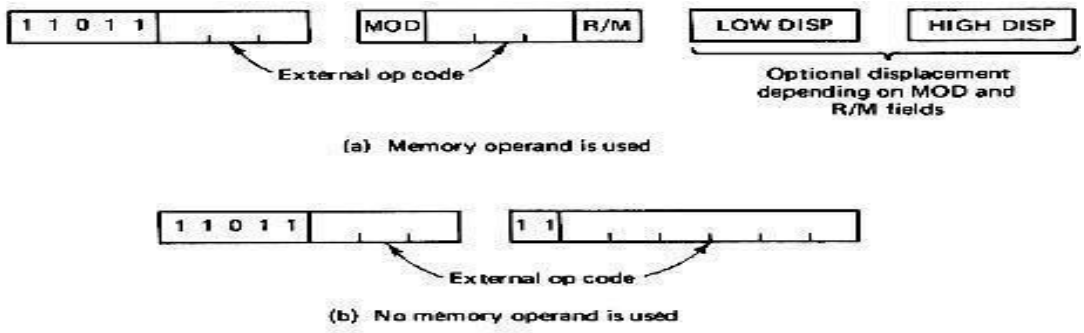

Figure 2.3.2 Machine code formats for the ESC instruction

[Source: "Microcomputer Systems: The 8086 / 8088 Family - Architecture, Programming and Design" by Yu-Cheng Liu, Glenn A.Gibson]

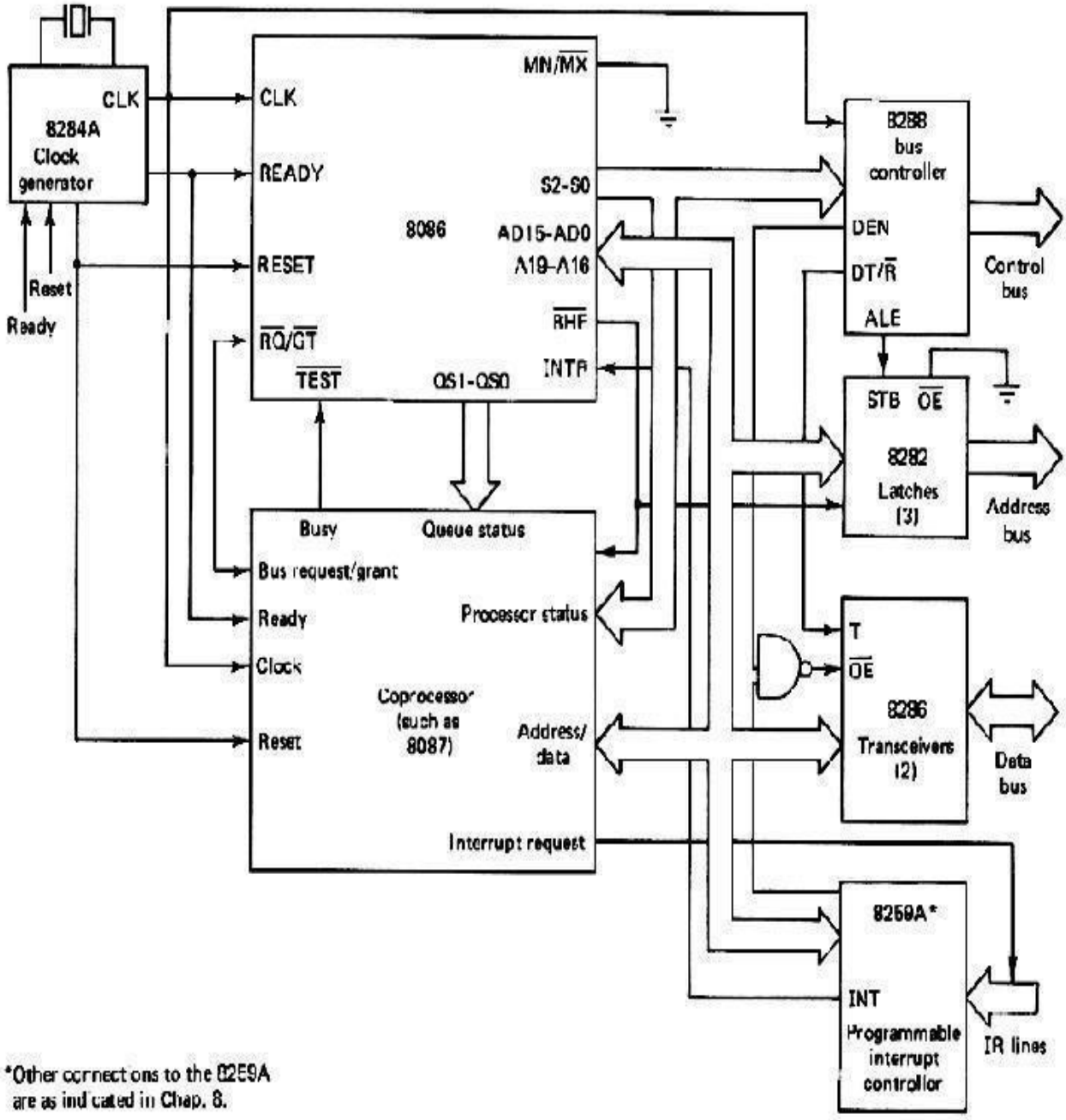

Figure 2.3.3 Coprocessor Configuration

[Source: "Microcomputer Systems: The 8086 / 8088 Family - Architecture, Programming and Design" by Yu-Cheng Liu, Glenn A.Gibson]

CLOSELY COUPLED CONFIGURATION

Closely Coupled processor may take control of the bus independently - 8089 sharesCPU's clock and bus control logic.

- communication with host CPU is by way of shared memory
- host sets up a message (command) in memory
- independent processor interrupts host on completion
- Two 8086's cannot be closely coupled
- Coprocessor cannot take control of the bus, it does everything through the CPU
  -

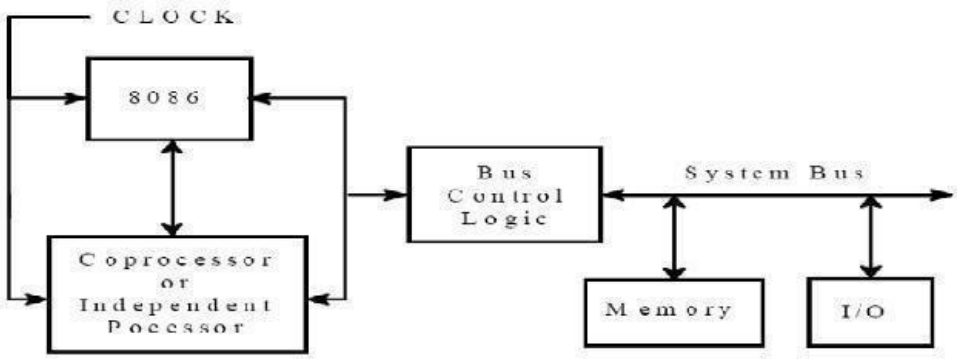

Figure 2.3.4 closely coupled configuration

[Source: "Microcomputer Systems: The 8086 / 8088 Family - Architecture, Programming and Design" by Yu-Cheng Liu, Glenn A.Gibson]

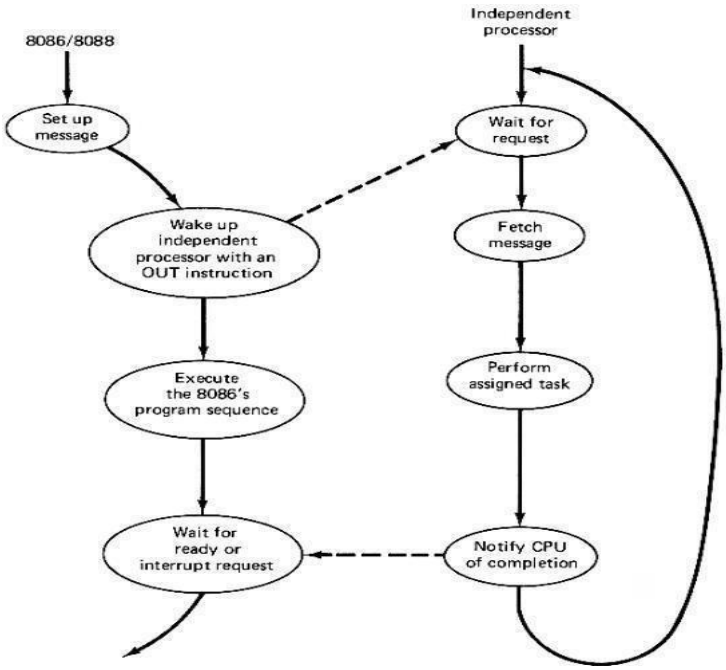

Figure 2.3.5 Interprocessor communication through shared memory

[Source: "Microcomputer Systems: The 8086 / 8088 Family - Architecture, Programming and Design" by Yu-Cheng Liu, Glenn A.Gibson]

LOOSELY COUPLED CONFIGURATIONS:

A loosely coupled configuration provides the following advantages:

1. High system throughput can be achieved by having more than oneCPU.
2. The system can be expanded in a modular form. Each bus master module is an independent unit and normally resides on a separate PC board. Therefore, a bus master module can be added or removed without affecting the other modules in the system.
3. A failure in one module normally does not cause a breakdown of the entiresystem and the faulty module can be easily detected and replaced.
4. Each bus master may have a local bus to access dedicated memory or I/O devices so that a greater degree of parallel processing can be achieved. More than one bus master module may have access to the shared system bus

Extra bus control logic must be provided to resolve the bus arbitration problem.The extra logic is called bus access logic and it is its responsibility to make sure that only one bus master at a time has control of the bus.

Simultaneous bus requests are resolved on a **priority basis:** There are three schemes forestablishing priority as shown in Figure 2.3.6

1. Daisy chaining.
2. Polling.
3. Independent requesting

DAISY CHAINING:

- Need a bus controller to monitor bus busy and bus request signals
- Sends a bus grant to a Master and each Master either keeps the service or passes it on
- Controller synchronizes the clocks
- Master releases the Bus Busy signal when finished

POLLING:

- Controller sends address of device to grant bus access
- Can use priority resolution here:

- Highest priority is granted first, if it does not respond, then a lower priority is granted, and so on until someone accepts
- (ie: one request line, 3-bit grant line).

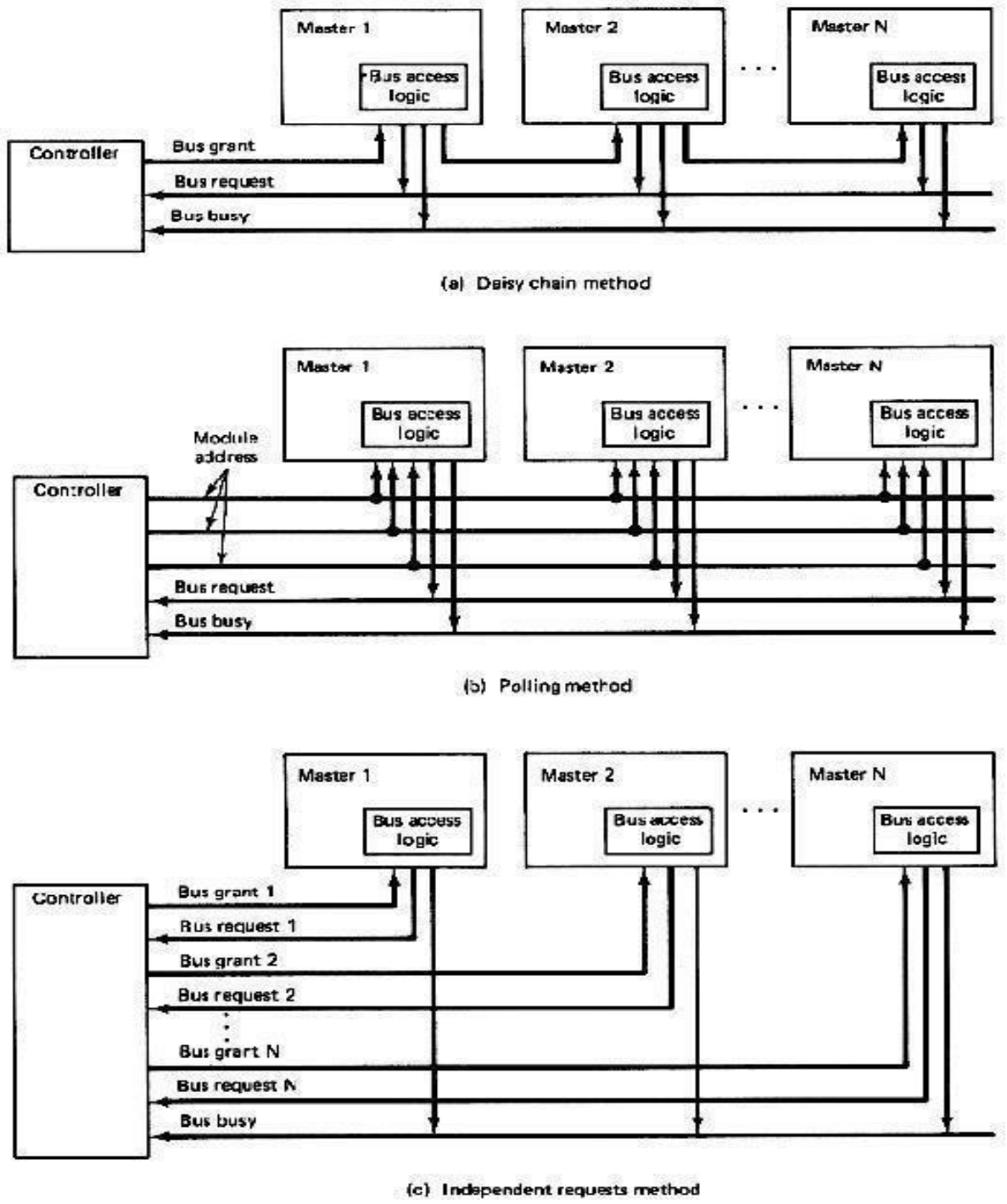

Figure 2.3.6 Bus allocation schemes

[Source: "Microcomputer Systems: The 8086 / 8088 Family - Architecture, Programming and Design" by Yu-Cheng Liu, Glenn A.Gibson]

INDEPENDENT:

- Each master has a request and grant line
- Now just a question of priority
- Could have fixed priority, rotating priority, etc.
- usually fixed because memory is desired to be the highest priority
- Synchronization of the clocks must be performed once a Master is recognized
- Master will receive a common clock from one side and pass it to the controller which will derive aclock for transfer
- Can accurately predict calculations (since memory is always the highest priority)