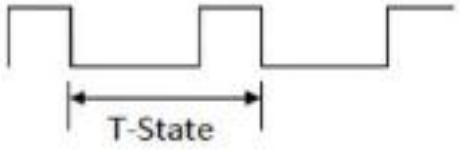
System Bus Timing

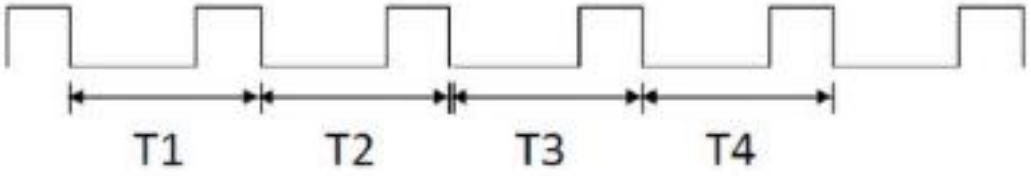
System Timing Diagrams

- T-State:
  - One clock period is referred to as a T-State

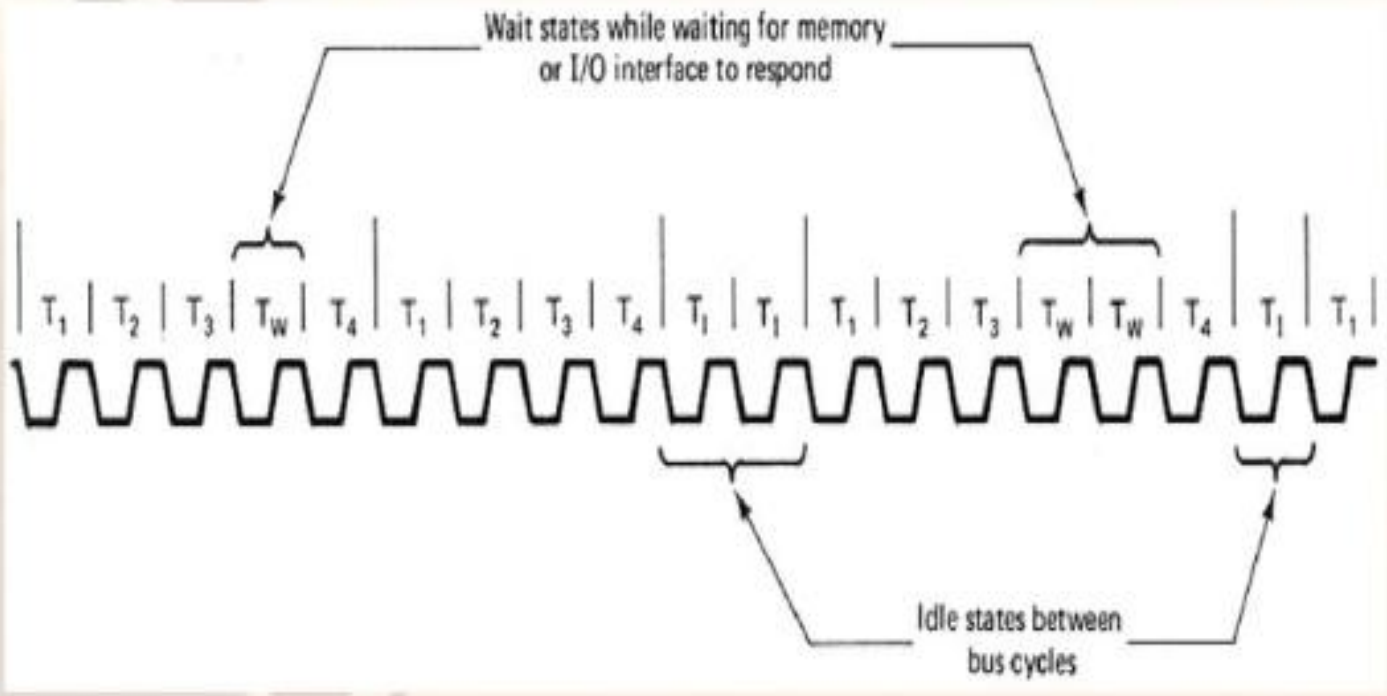


  - An operation takes an integer number of T-States

- CPU Bus Cycle:
  - A bus cycle consists of 4 or more T-States

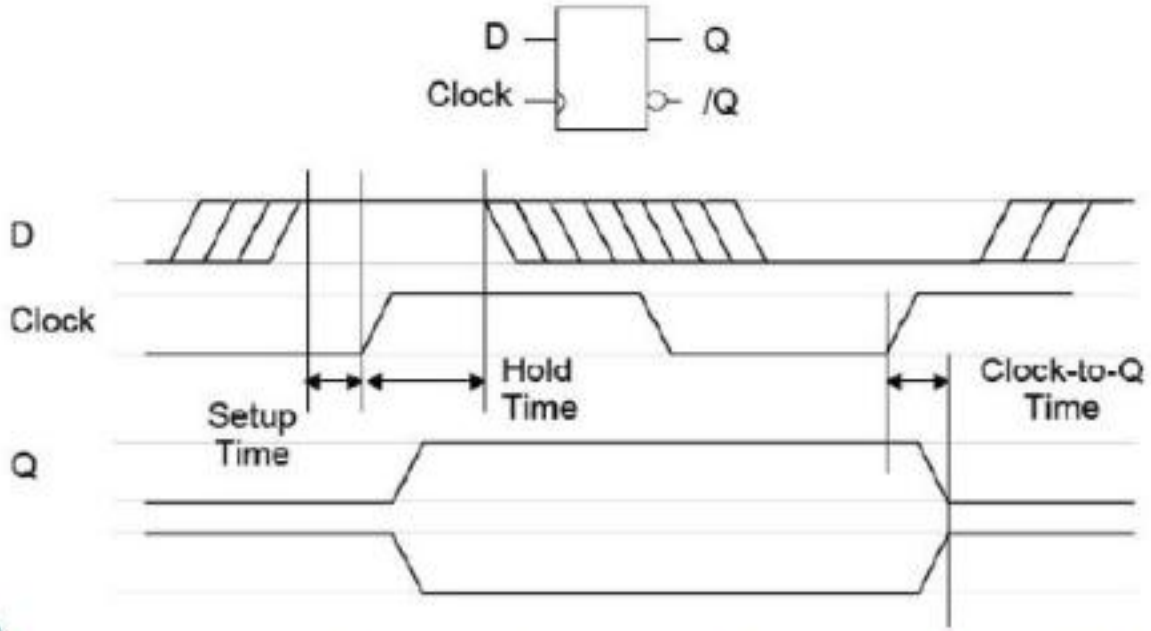


Typical sequence of bus cycles



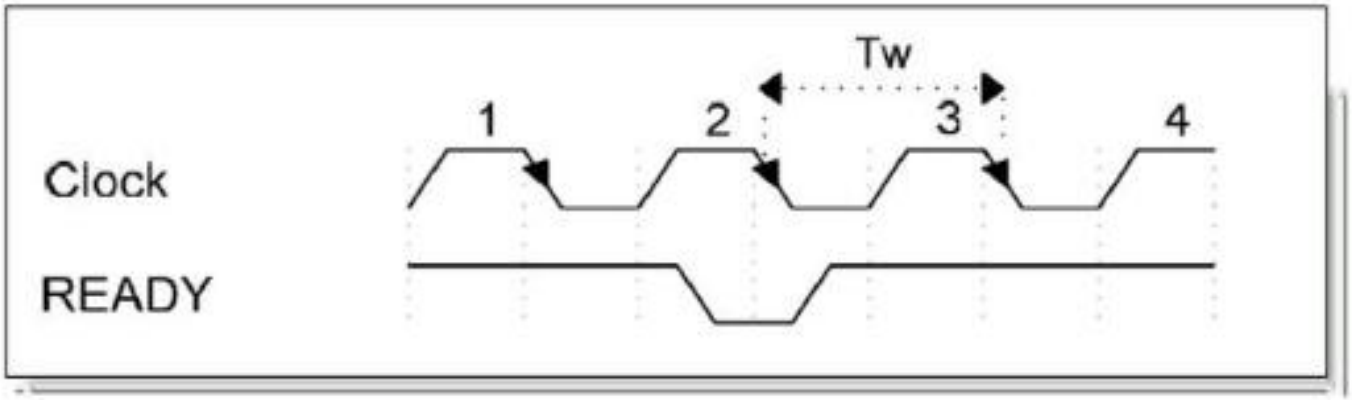
Length of a bus cycle = 4 clock cycles + wait states

Setup & Hold Time



- Setup time – The time before the rising edge of the clock, while the data must be valid and constant
- Hold time – The time after the rising edge of the clock during which the data must remain valid and constant

WAIT State



- A wait state (T_w) is an extra clocking period, inserted between T2 and T3, to lengthen the bus cycle, allowing slower memory and I/O components to respond.

- The READY input is sampled at the end of T2, and again, if necessary in the middle of Tw. If READY is '0' then a Tw is inserted.

Bus Timing

During T 1 :

- *The address is placed on the Address/Data bus.*
- *Control signals M/ IO , ALE and DT/ R specify memory or I/O, latch the address onto the address bus and set the direction of data transfer on data bus.*

During T 2 :

- *8086 issues the RD or WR signal, DEN , and, for a write, the data.*
  - DEN *enables the memory or I/O device to receive the data for writes and the 8086 to receive the data for reads.*

During T 3 :

- *This cycle is provided to allow memory to access data.*
- *READY is sampled at the end of T 2 .*
  - *If low, T 3 becomes a wait state.*
  - *Otherwise, the data bus is sampled at the end of T 3 .*

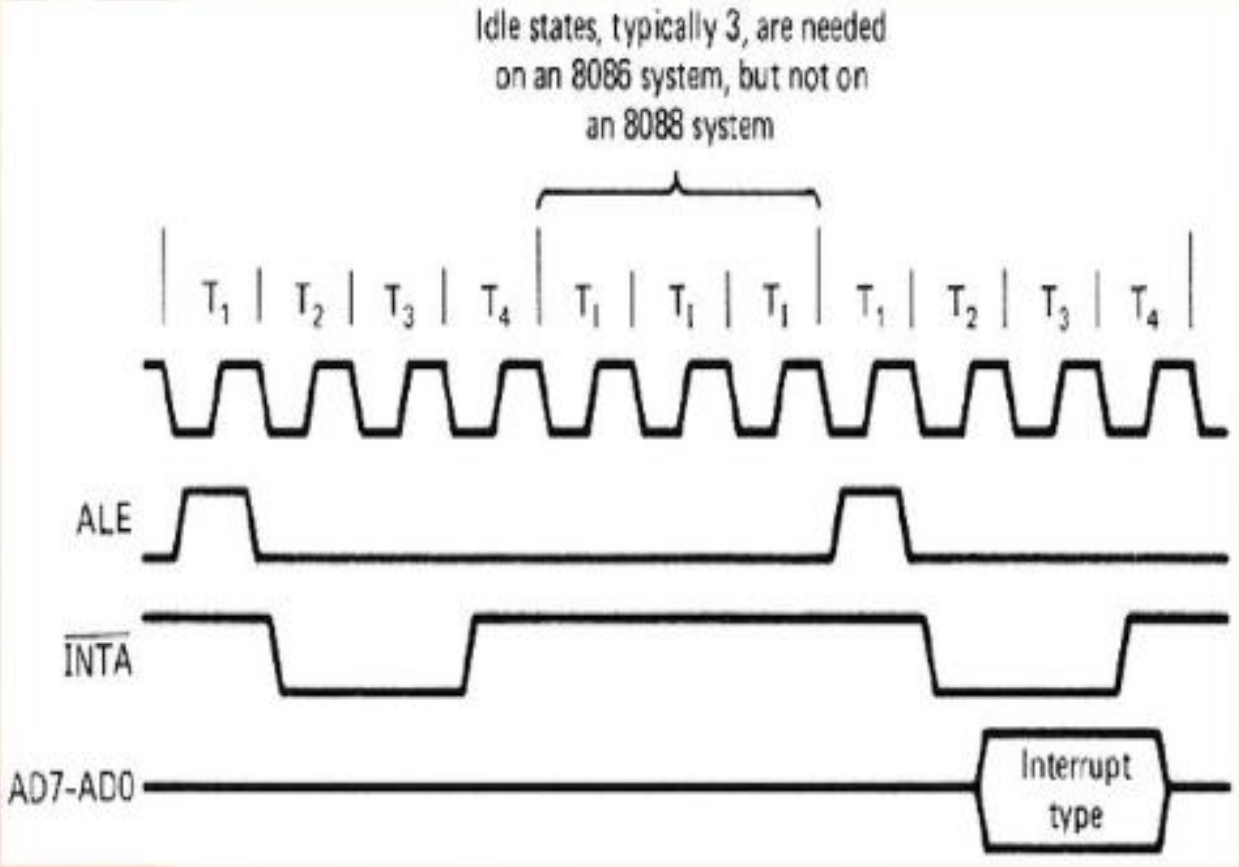
During T 4 :

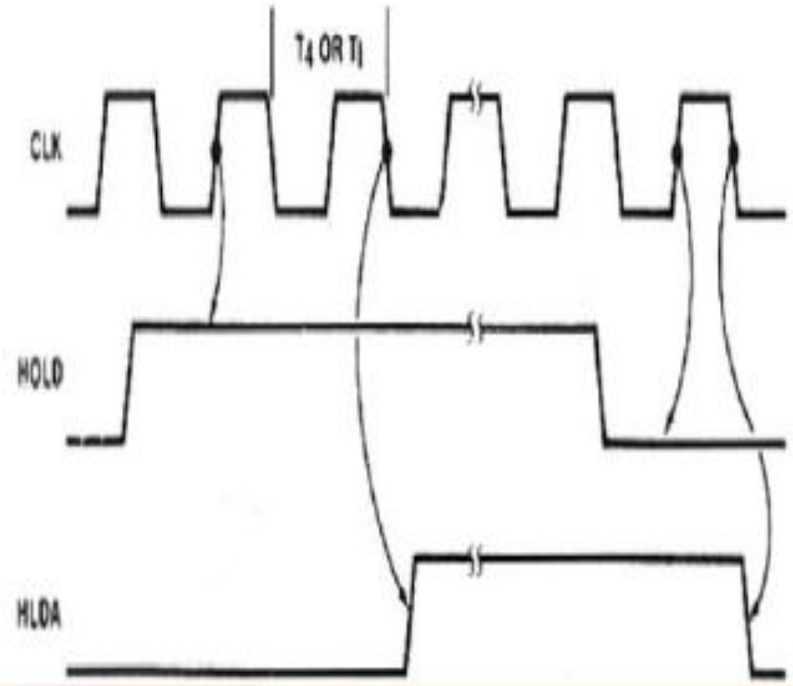
- *All bus signals are deactivated, in preparation for next bus cycle.*
- *Data is sampled for reads, writes occur for writes.*

INTA CYCLE – MINIMUM MODE

- If an interrupt request has been recognized during previous cycle & an instruction has been completed, then a –ve pulse is applied to INTA during current bus cycle & next cycle
- Pulses extend from T2 to T4
- The I/F accepting the ACK,puts INTERRUPT TYPE on AD7-AD0 which will be available from T2 to T4

INTA cycle


Idle states, typically 3, are needed
on an 8086 system, but not on
an 8088 system
T1
T2
T3
T4
TI
TI
TI
T1
T2
T3
T4
ALE
INTA
AD7-AD0
Interrupt
type



The HOLD pin is tested at the leading edge of each clock pulse.

If a HOLD signal is received by the processor before T4 or during a T1 state, then the CPU activates HLDA and the succeeding bus cycle will be given to the requesting master until that master drops its request.

Maximum mode – bus grant

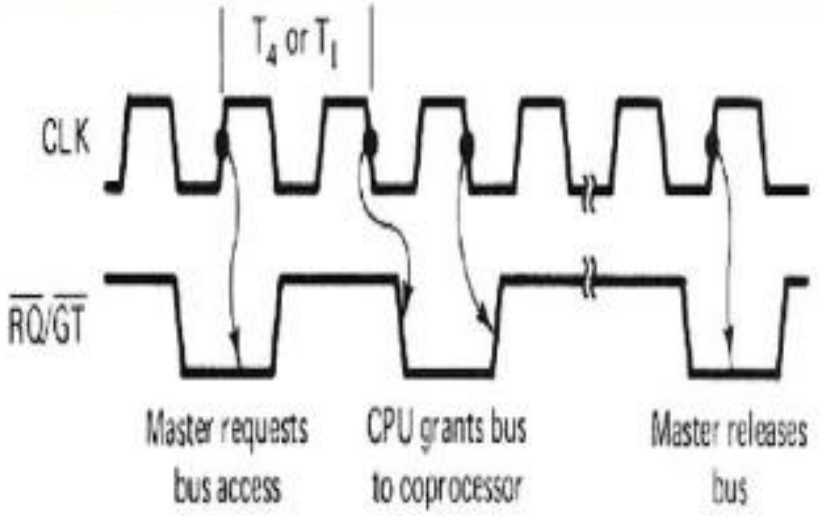


Figure 8-16 Timing for maximum mode bus requests and grants. (Reprinted by permission of Intel Corporation. Copyright 1979.)

RQ/GT is done in sequence of 3 pulses
RQ/GT are examined at the rising edge of each clock pulse, if a request is detected ,µP will apply a grant pulse to RQ/GT Immediately following next T4 or T1

RQ/GT0 has higher priority

Bus grant – maximum mode

- REQUESTING MASTER gets the bus control for one or more bus cycles.It sends release pulse to µP over the same line as it had sent request

I/OPROGRAMMING

- I/O programming discuss the ways in which information can be transferred between input-output devices or mass storage devices and the CPU or memory. The three modes of transfer of device data, commands and status are,

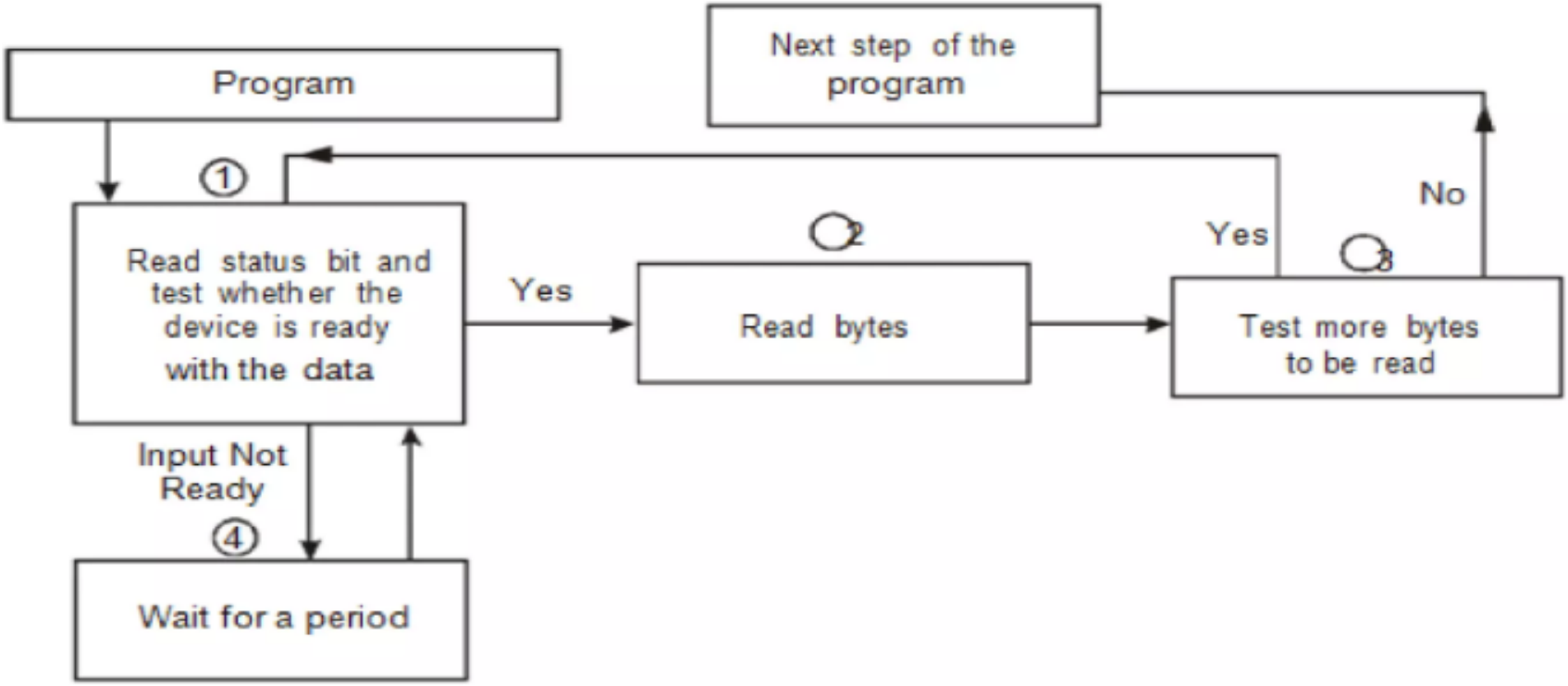
1. Programmed I/O
2. Interrupt driven I/O
3. DMA Transfer

- ***Programmed I/O*** **:** The program determines which interfaces need servicing bit testing the ready bits in their status registers. Programmed testing of ready bits or signals is known as *polling*.
- ***Interrupt driven I/O:*** An external interrupt is sent to the CPU from the interface when the interface has data to input or is ready to accept data and the I/O operation is performed by an interrupt routine.
- **DMA transfer:** The interface requests the use of the bus by sending a signal through the control line and makes the necessary transfer without the help of the CPU.

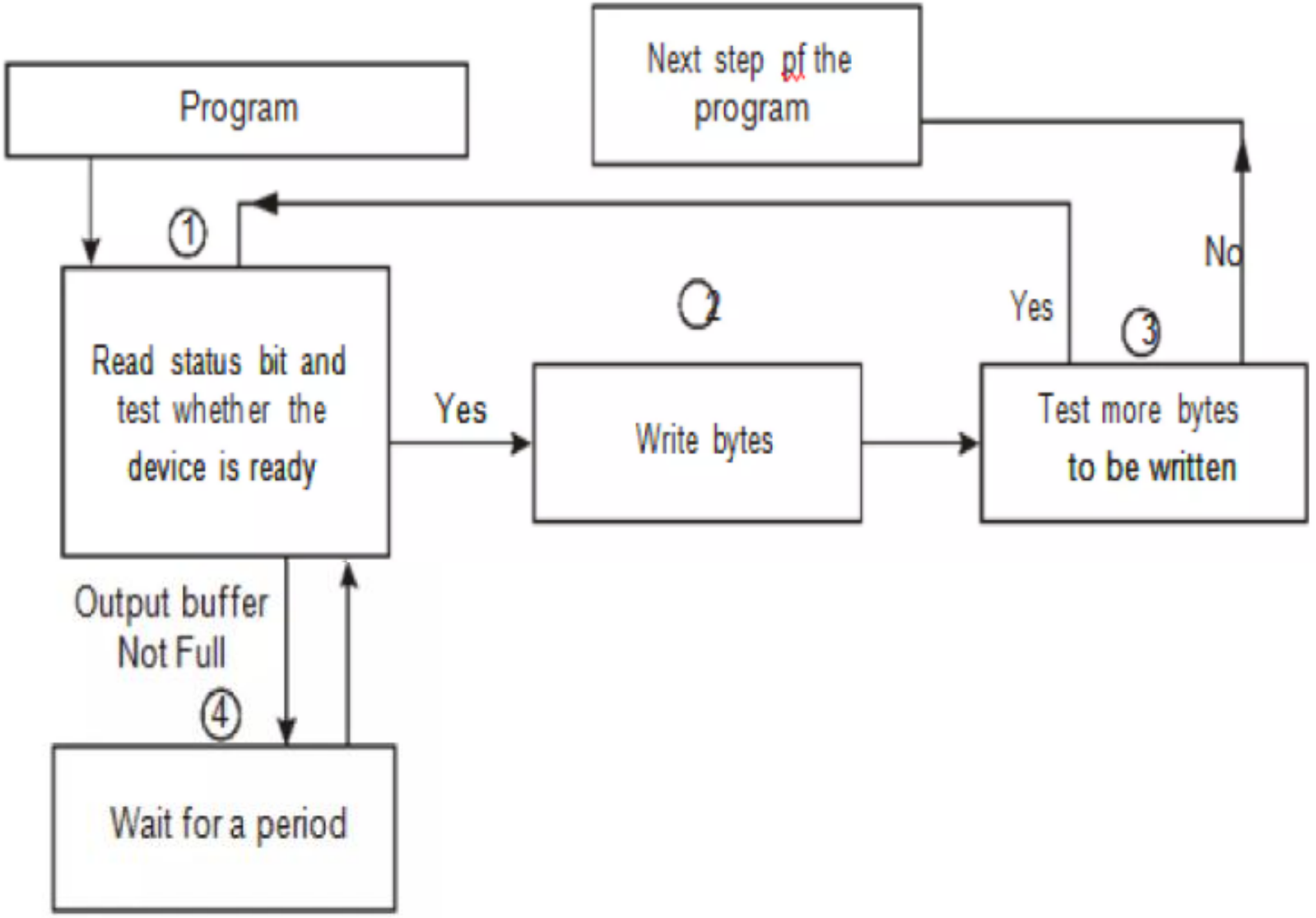
PROGRAMMED I /O

Programmed I/O consists of continually examining the status of an interface and performing an I/O operation with the interface when its status indicates that it has data to be input or its data-out buffer register is ready to receive data from the CPU.

Read input in programmed I/O mode



Output write in programmed I/O mode



Interrupt driven I /O

- There are several ways of combining with interrupt I/O, some involving only software, some only hardware, and some a combination of the two. They are,
  i) Polling
  ii) Daisy chaining
  iii) Interrupt priority management hardware

Polling

Polling is the most common and simplest method of I/O control. It requires no special hardware and all I/O transfers are controlled by the CPU program. Polling is a synchronous mechanism, by which devices are serviced in sequential order.

The polling technique has the following limitations:

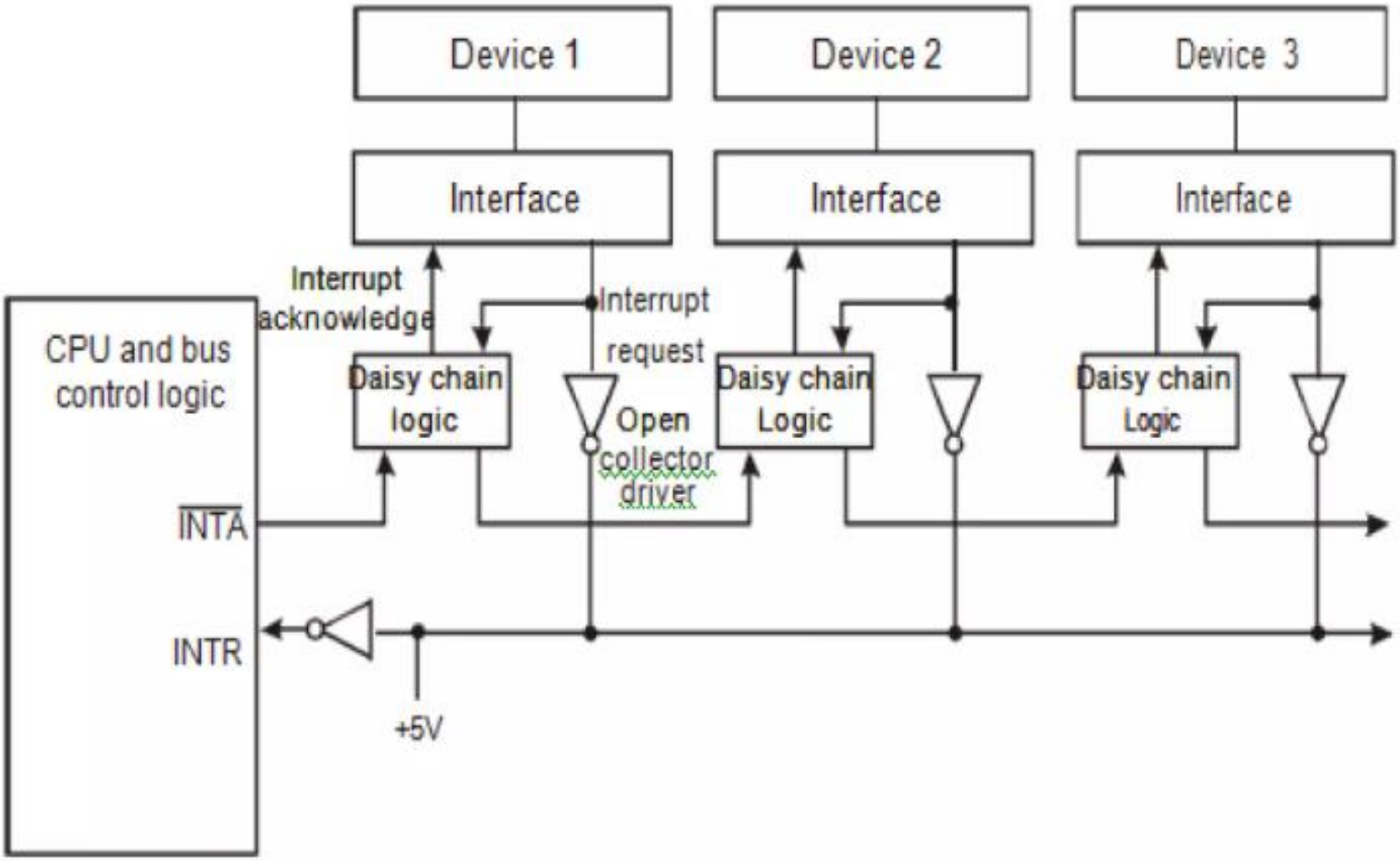
- It is wasteful of the processors time, as it needlessly checks the status of all devices all the time.
- It is inherently slow
- When fast devices are connected to a system, polling may simply not be fast enough to satisfy the minimum service requirements

Daisy chaining

- It is a simple hardware means of attaining a priority scheme. It consists of associating a logic circuit with each interface and passing the interrupt acknowledge signal through these circuits as shown in Fig.
- The priority of an interface is determined by its position on the daisy chain. The closer it is to the CPU the higher its priority.
- This is significantly faster than a pure software approach. A daisy chain is used to identify the device requesting service.

- Because more than one device can assert the shared interrupt line simultaneously, some method must be employed to ensure device priority.
- This is done using the interrupt acknowledge signal generated by the processor in response to an interrupt request.
- Each device is connected to the same interrupt request line, but the interrupt acknowledge line is passed through each device, from the highest priority device first, to the lowest priority device last.
- After preserving the required registers, the microprocessor generates an interrupt acknowledge signal. This is gated through each device.
- If device 1 generated the interrupt, it will place its identification signal on the data bus, which is read by the processor, and used to generate the address of the interrupt-service routine.
- If device 1 did not request the servicing, it will pass the interrupt acknowledge signal on to the next device in the chain. Device 2 follows the same procedure, and so on.

Daisy chaining



Interrupt priority management hardware

- A more flexible hardware priority arrangement can be held by designing a programmable interrupt priority management circuit and including it in the bus control logic.
- This is the fastest system. The duty is placed on the requesting device to request the interrupt, and identify itself. The identity could be a branching address for the desired interrupt-handling routine.

- If the device just supplies an identification number, this can be used in conjunction with a lookup table to determine the address of the required service routine. Response time is best when the device requesting service also supplies a branching address.

Direct Memory Access Block Transfer

- A DMA controller allows devices to transfer data to or from the system's memory without the intervention of the processor.
- During any given bus cycle, one of the system components connected to the system bus is given control of the bus.
- Taking control of the bus for a bus cycle is called cycle stealing.