8051 AND ADVANCED PROCESSOR ARCHITECTURES – Lesson-7: REAL WORLD INTERFACING – Part 2

4. Addresses of Ports and Devices in Real World Interfacing

Device Control Register, Status Register, Receive Buffer, Transmit Buffer

- Each I/O device is at a distinct address or set of addresses
- Each device has three sets of registers —data buffer register(s), control register(s) and status register

Device Addresses

- Device control and status addresses and port address remains constant and are not re-locatable in a program as the glue circuit (hardware) to accesses these is fixed during the circuit design.
- There can be common addresses for input and output buffers, for example SBUF in 8051

The processor, memory, devices Glue Circuit

- The processor, memory and devices are interfaced (glued) together using a programmable circuit like GAL or FPGA. The circuit consists of the address decoders as per the memory and device addresses allocated and the needed latches multiplexers/ demultiplexers

Device Addresses

- There may be common addresses for control and status bits
- There can be a control bits, which changes the function of a register at a device address

Example

- S*erial line device* addresses of device registers
- Fixed by its hardware configuration of UART port interface circuit in a of a system employing 80x86 processor. .
- 0x2F8 to 0x2FE at COM2 COM1 in IBM PC

Feature of UART serial line device in PC

- Two I/O data buffer registers (one for receiving and other for transmitting) at a common address, 0x2F8
- Data of two bytes of *Divisor Latch* are at the distinct addresses, 0x2F8 (LSB) and 0x2F9 (MSB)
- *Three Control Registers of the device are at three distinct addresses* 0x2F90x2FA, 0x2FB and 0x2FC.

Feature of UART serial line device in PC

- Three Status Registers of the device are at three distinct addresses 0x2FA, 0x2FD and 0x2FE

Device Addresses

- Processor accesses device registers and buffer registers from allocated addresses for the Ports and Devices

5. Memory Mapped IO to ports and device

Memory mapped IO or device Access

- Processor access to device is as if to a memory address

Interfacing Processor with Memory Mapped IO

- No separate I/O address space exists for the ports and devices.
- Instructions as well as control signals for operations on bytes at the memory, IO port and device addresses are same.
- No separate input-output and memory load-store instructions.
- Arithmetic, logical and bit manipulation instructions that are available for data in memory, are also available for the IO operations.

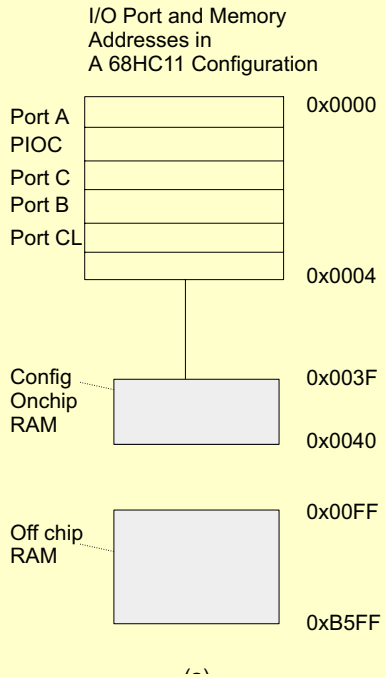
Interfacing Processor with Memory Mapped IO

- Enables direct manipulation of the data taken from the IO port or device.
- Directly manipulate the data stored at the IO port or device.
- All the arithmetic, logical and bit manipulation instructions that are available for data in memory, can be done using an accumulator or any other register or any other memory address, where the IO port byte is transferred after or during or before the arithmetic or logical operation.

Memory mapped IOs Example

- 8051 microcontroller devices have the addresses for processor-accesses that are not distinct from the memory and are accessed with same set of instructions and control signals RD and WR

Processor and memory organisation with I/O devices memory assignments in the 68HC11 (having memory mapped IO architecture Memory and Port addresses in 68HC11



6. IO address Mapped IO port or Device Access

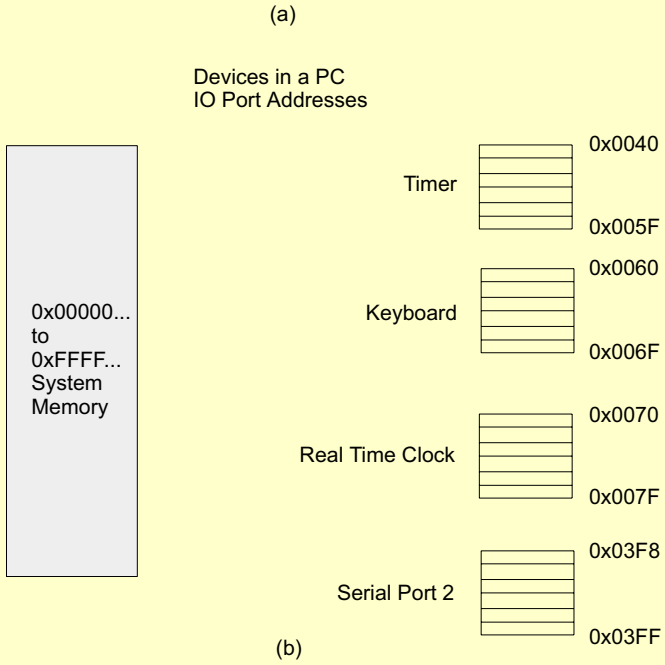
IO mapped IO

- Processor access to device is by distinct instruction and control signals
- Memory address is accessed by Load and Store instructions and IO device address is accessed by distinct set of instructions *OUT* and *IN* and distinct set of control signals (IOWR and IORD

IO mapped IOs Example

- 80x86 processor accesses the external devices using the addresses in space, which is distinct from the memory

Device Addresses in 80x86 based PC



Features of IO addresses mapped IOs

- Separate I/O address space than for the ports and devices.
- Instructions and control signals for operations on bytes at the memory and IO port and devices are distinct.
- Advantage of simplicity.
- IO devices and port addresses are interfaced independently of memory without considering the memory addresses that are assigned for software and data.

Features of IO addresses mapped IOs

- Processor separate *input-output* (for read and write) instructions and memory load-store (for read and write) instructions.
- All the arithmetic, logical and bit manipulation instructions that are available for data in memory, can be done using an accumulator, where the IO port byte is transferred before an arithmetic or logical operation

7. Interrupts and IOs

Interrupt driven IO

- Processor access to device is by executing an ISR on a device-interrupt, for example, interrupt on timer overflow, keyboard data ready or transmit-data buffer empty

Interrupt driven IO

- Used when the processor needs to perform a prolong data transfer operations using a I/O device and wants to be able to do other work while waiting for the transfer operations to complete.

Interrupt driven IO

- IO device function slow as compared to processor.
- Interrupt driven IO can be used in those cases.
- *Interrupt* is the mechanism used by most processors to handle *asynchronous* type of events

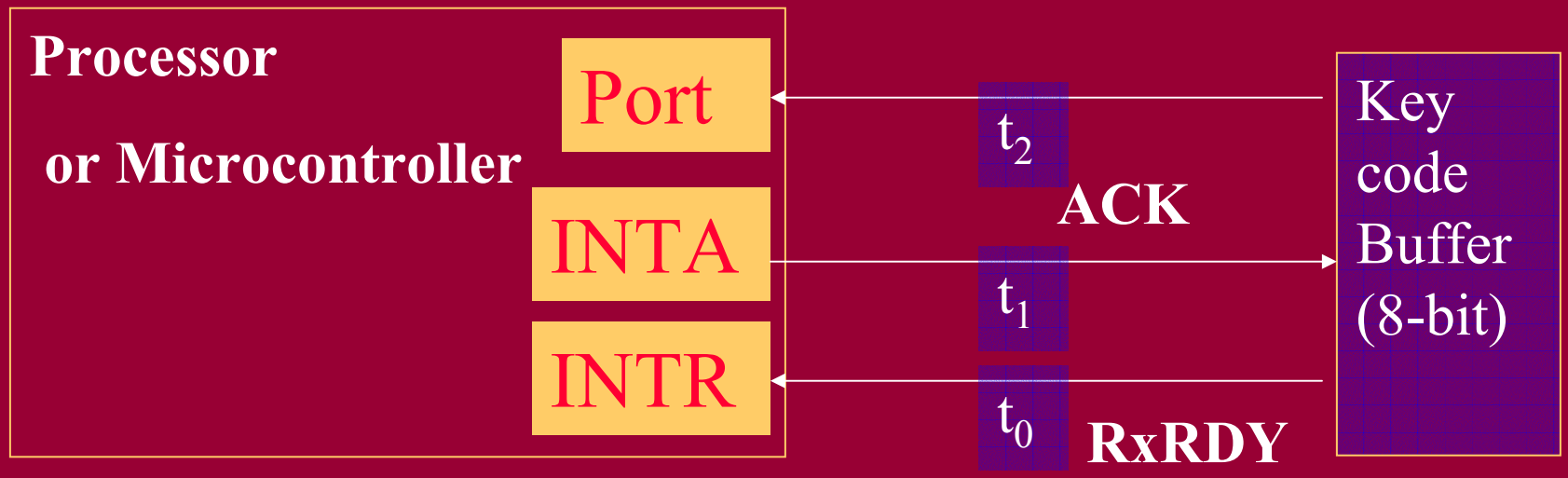
Interrupt driven IO

- Interrupt allows a device to request that the processor to stop what it is currently doing and execute software called interrupt service routine to process the device's request, much like a procedure call. Here the call is initiated by an event at the external or internal device rather than by the program instruction running on the processor

Keyboard **Example**

- Takes about 10 ms to send the code for the key and maximum 10 keys can be pressed in 1 s
- When does a key input event occurs is not fixed.
- Intervals between two events of successive key inputs are not fixed.
- Interrupt driven mode, when a key is pressed, an interrupt signal RxRDY (receiver data ready) to the processing unit causes the execution of a service routine and the service routine program reads the byte for code.

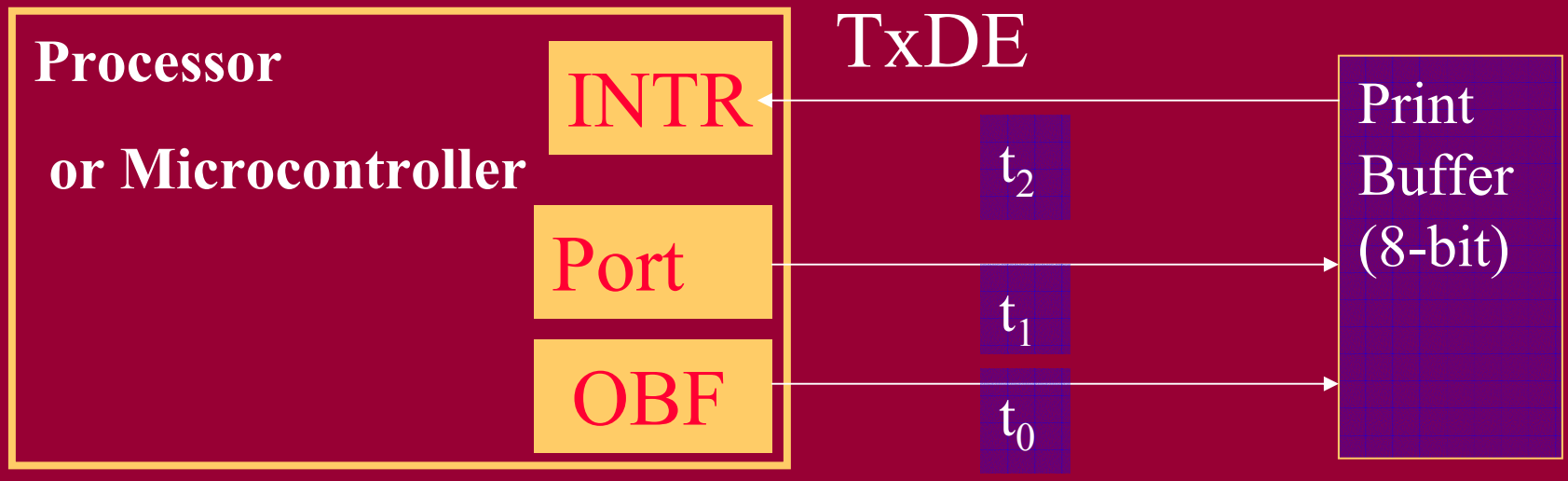
Keyboard interrupt



Printer example

- Maximum 300 characters can print in 1 s,
- 0.3 ms to print the code sent at the output by a port.
- When does a print operation is complete for a character is not fixed.
- Intervals between two events of successive print of the characters are not fixed.
- Interrupt driven mode, when a print action completes, an interrupt signal TxDE (transmission data empty) to the printer processing-unit (print controller) will cause the execution of a service routine and the service routine will send another byte as output.

Printer interrupt



Summary

We learnt

- Addresses of Ports and Devices in Real World Interfacing
- Processor accesses device registers and buffer registers from allocated addresses for the Ports and Devices
- Memory mapped IO— Processor access to device is as if to a memory address
- IO mapped IO — Processor access to device is as if to a memory address
- Interrupt driven IO — Processor access to device is by executing an ISR on a device-interrupt, for example, interrupt on timer overflow, keyboard data ready or transmit-data buffer empty

End of Lesson 7 of Chapter 2