

Week #6:

Qno.1

Write short notes on:  
Wire, bus, port and timing diagram for read and write  
protocols.

Wire:

A wire is a flexible metallic conductor, especially one made of copper, usually insulated and used to carry electric current in a circuit.

Port:

A port is a conducting device like metal, on the periphery of a processor, through which a signal is input to or output from the processor.

A port may refer to single wire or to a set of wires with a single function.

Bus:

The term bus refers to a set of wires with a single function within a communication. The term bus can also refer to the entire collection of wires used for the communication.

Read and write protocol

For read

The CPU must send the memory address.

The read line must be enabled.

The process must wait till the memory is ready.

Then accept the bits in the data lines.

GOOD MORNING  
PAGE NO.: \_\_\_\_\_  
DATE: \_\_\_\_\_

GOOD MORNING  
PAGE NO.: \_\_\_\_\_  
DATE: \_\_\_\_\_

rd/wr

enable

addr

data

tsetup

tread

read protocol

Fig: Timing diagram for read protocol.

For write

- o The CPU must send the memory address.
- o The write line must be enabled.
- o The processor sends the data over the data lines.
- o The processor must wait till the memory is ready.

rd/wr

enable

addr

data

tsetup

twrite

Fig: Timing diagram for write protocol.

Qno.2

Explain strobe and Handshake protocol with necessary block diagram and time diagram. Why compromise of strobe and handshake is needed? Explain with necessary diagram and protocol diagram or timing diagram.



Ans: Strobe protocol

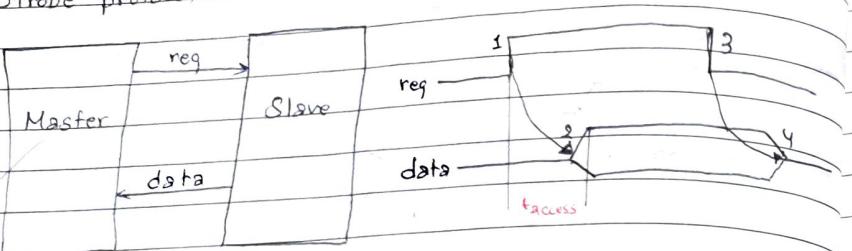


Fig: Strobe protocol.

1. Master asserts req to receive data.
2. Slave puts data on bus within time taccess.
3. Master receives data and deasserts req.
4. Slave ready for next request.

Handshake protocol:

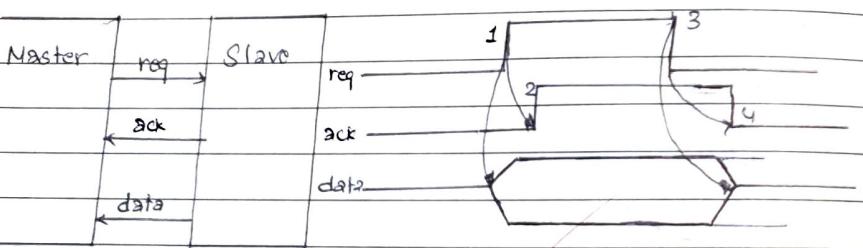


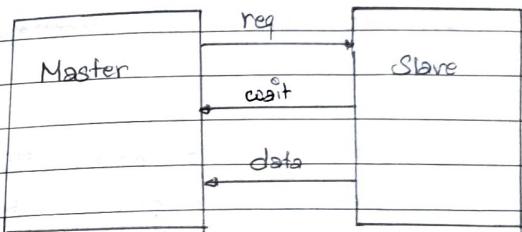
Fig: Handshake protocol.

Master asserts req to receive data.  
 Slave asserts ack to receive data.  
 Slave puts data on bus and asserts ack.  
 Master receives data and deasserts req.  
 Slave ready for next request.

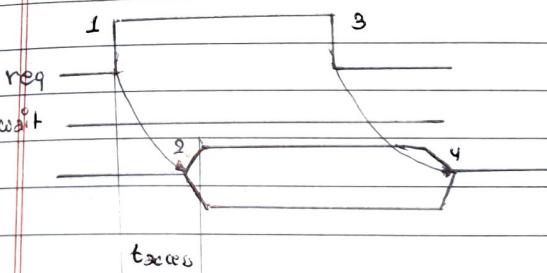
Strobe & Handshake combined.

It is a compromise of both handshake and strobe protocol. A compromise is required to achieve both the speed of a strobe protocol and the varying response time tolerance of a handshake protocol.

Block diagram & Timing diagram.

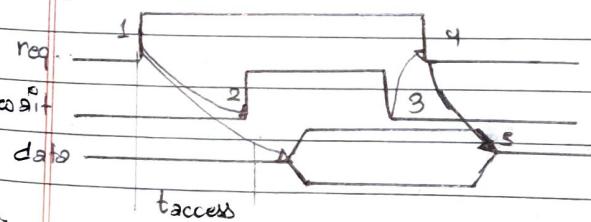


Fast-response case:



1. Master asserts req to receive data.
2. Slave puts data on bus within time taccess. (Wait line is unused)
3. Master receives data and deasserts req.
4. Slave ready for next request.

Slow-response case:



1. Master asserts req to receive data.
2. Slave can't put data within access, asserts wait ack.
3. Slave puts data on bus and deasserts wait.
4. Master receives data and deasserts req.
5. Slave Slave ready for next request.

Ques 3

Write short notes on following topics with necessary block diagram:

#### a. Port based I/O:

In port based I/O, also known as parallel I/O, a port can be directly read and written by processor instructions just like any other register in the microprocessor. The port is usually connected to a dedicated register.

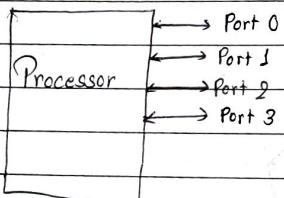


Fig: Port based I/O

#### b. Bus based I/O:

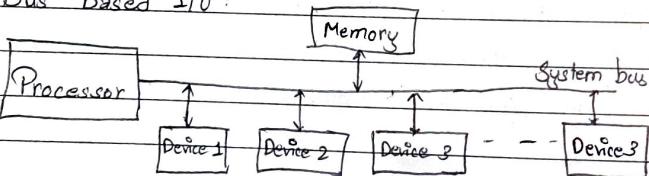


Fig: Bus based I/O.

In bus-based I/O, the microprocessor has a set of address, data and control ports for I/O addressing which forms a single bus.

The communication protocol is built in the processor. We normally consider the access to the peripheral as I/O but don't consider the access to memory as I/O, since the memory is considered more as a part of the processor.

#### c. Extended port based parallel I/O:

When a processor may require more ports than that are available, in such case, a parallel I/O can again be used to interface with another port based parallel I/O. Thus, we have extended the number of available port. Using such peripheral in this manner is often called as extended parallel I/O.

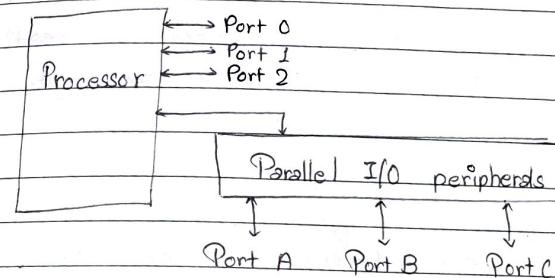


Fig: Extended port based parallel I/O.

#### d. Extended bus based I/O:

When a processor require more ports then, a parallel I/O peripheral is used to increase the number of ports along with the system bus. Using such a peripheral in this manner is often called as extended bus based I/O.

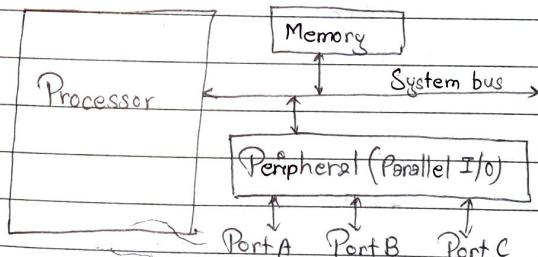


Fig: Extended bus based I/O.

### e) Memory mapped I/O:

In memory mapped I/O, I/O devices are mapped to 16 bit address. The peripherals occupy specific address in the existing address space.

### f) Standard I/O:

Standard I/O is also known as I/O mapped I/O. In this process, the I/O devices are mapped to 8-bit address. i.e. the processor selects the signals from those devices which are using 8-bit address lines from processor.

If the processor uses standard I/O, the processor requires special instruction for reading and writing peripheral. These instructions are often called IN & OUT.

Qno.4

Explain fixed instruction interrupt, vectored interrupt, interrupt address table and differentiate them. State some interrupt issues.

Ans: Fixed interrupt:  
In fixed interrupt, the address to which the microprocessor jumps on an interrupt is built into the microprocessor. So it is fixed and cannot be changed.

The assembly programmer either puts the ISR at that address or jump to the real ISR if there is not enough bytes available.

### Vectored interrupt:

In vectored interrupt, peripheral must provide the address to the processor. In this method, along with INT pin, INTA is also required to acknowledge that

the interrupt has been detected and the peripheral can provide the address of relevant ISR using system bus. The peripheral provides the address through the address bus which is read by microprocessor.

### Interrupt address table:

Interrupt address table is used as compromise between the fixed and vectored interrupt method. A table with ISR address which is stored in memory of the processor. A peripheral device provides a number corresponding to an entry in a table rather than providing the ISR address. And the processor reads this entry number from the bus and reads the corresponding table entry to obtain the ISR address.

### Some interrupt issues:

External interrupt may be maskable or non-maskable. In maskable interrupt, the programmer may force the microprocessor to ignore the interrupt pin either by executing a specific instruction to disable the interrupt or by setting bits in an interrupt configuration register. It is important when critical works need to be executed first.

In non-maskable interrupt, the interrupt cannot be disabled by the programmer. It requires a distinct pin and used for very drastic situation such as power failure. In such condition, a non-maskable interrupt can cause a jump to a subroutine that store the critical data in non-volatile memory before the power is completely gone.

Another issue regarding the interrupt is jump ISR in which microprocessor

content or partial state before jumping to ISR. Some processor saves PC, registers which consumes many cycle, however must not modify the register if its content is not saved.

Qno.5

Why direct memory access (DMA) controller is needed? Describe DMA controller with block diagram and summarize its operation.

Ans:

DMA is specially used to transfer the data between the memory and peripherals. The peripheral request the service from the DMA control. The DMA controller request the processor to use its service from the system bus prior to data transfer. Data transfer takes place without the involvement of the processor and processor can continue its regular task unless it requires the system bus.

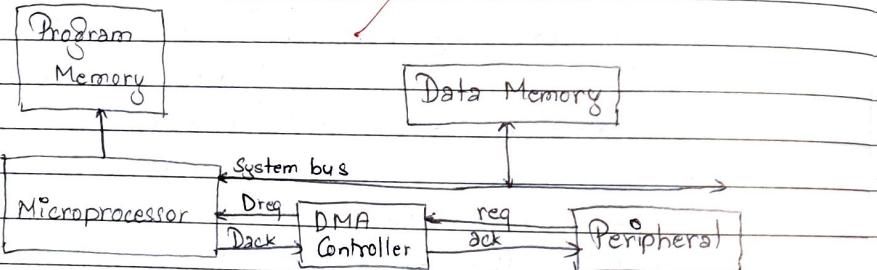


Fig: DMA controller

Operation:

The flow of action for the transfer of data between peripheral and memory using DMA can be summarized as:

Initially processor is busy executing its main program.

- o After peripheral has data within its register, it asserts request line for service from DMA.
- o DMA asserts request signal to request the system bus from processor.
- o Processor release the system bus after getting the request from DMA and acknowledge about it to DMA.
- o DMA asserts acknowledge signal to peripherals, and starts transfer of data as requested.
- o After the completion of transfer, all control lines are deasserted and processor regains the control of system bus.

Qno.6

What is arbitration? Describe priority arbitration and its type.

Ans:

Arbitration is the mechanism of making priority for services to the peripheral devices in case of request from many devices.

Priority Arbitration:

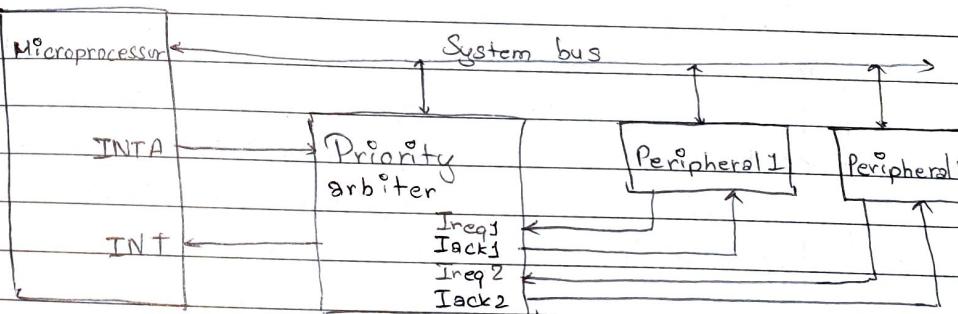


Fig: Arbitration using priority arbiter.

Priority arbiter is a single purpose processor which is used to arbitrate among various request from the peripherals. Each of the peripherals which are connected to the arbiter can make request.

service. Using certain priority mechanism, arbiter selects a peripheral to permit the requested service. The figure above shows the priority arbiter connected with peripherals which use vectored interrupt to request service and processor provide services to the peripheral. Arbiter is connected to the system bus for configuration only. The configurations may include setting priorities of the peripheral.

#### Types of arbitration

- Fixed priority

Each peripheral is assigned a unique rank. If two peripherals simultaneously request for the service then, the arbiter chooses the one with the higher rank.

Such method is efficient when there is a clear distinction in priority among peripherals. But it can cause high-ranked peripherals to get much more service than other.

- Rotating priority or round-robin priority:

In round robin, each peripheral gets almost equal time for service from the arbiter. In this method, the priority of peripheral changes based on history of servicing of those peripherals. So, arbiter gets more complex.

This priority method is efficient when there is not much difference in priority among peripherals.

#### Qn. 7

Describe Daisy chain arbitration and network oriented arbitration with block diagram and operations.

#### Daisy chain arbitration

In daisy chain arbitration, peripherals are connected to each other in daisy-chain like manner.

The arbitration is built within the peripherals with each having request and acknowledge signals as shown in figure below. The request and acknowledge signal flow through the peripherals; peripheral request flows downstream to processor and processor ack acknowledge flows upstream to requesting peripherals. The peripheral which is connected near to the processor gets the highest priority while peripheral at the end of chain gets lowest one.

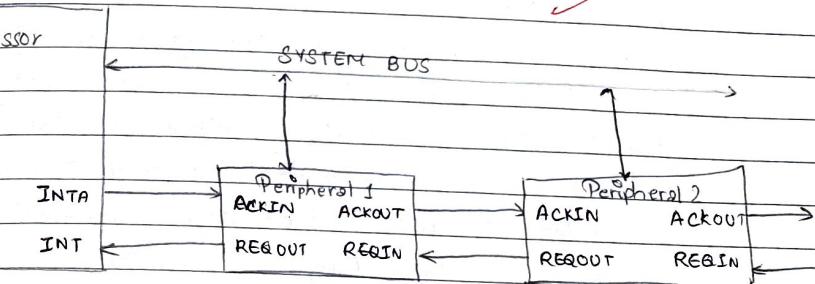


Fig: Daisy chain configuration.

#### Operation:

Suppose peripheral 2 requires service from the processor then, the operations can be summarized as:

- Microprocessor is busy in executing its own task.
- The request signal from peripheral 2 is send to processor through the peripheral 1 and interrupt pin is asserted.
- Processor stops its current work, stores its state, and asserts acknowledge strobe signal.
- The acknowledge signal reaches to peripheral 2 through peripheral 1. Since, the request is not generated by peripheral 1, it passes the signal to peripheral 2.
- Peripheral 2 puts its interrupt address vector on the system bus.

- Microprocessor read the ISR address from address bus and jump into that address to execute the ISR.
- After the execution of ISR, processor retrieves its state and resumes its operation.

### Network oriented arbitration:

A network oriented arbitration is done for multiple microprocessor communicating via a shared bus called as network. An arbitration is built into the bus protocol, since the bus serves as only connection among the microprocessor.

However, multiple bus processor may try to access the bus simultaneously resulting in data collision. The protocol must be designed in such a way that the contending processor don't start sending the data at the same time. Also, some statistical method can be used so as to make chances of data collision very rare. Some protocol uses efficient address encoding schemes in which higher priority address will overwrite the lower priority one.

### Ques 8

Write short notes on:

#### b) Daisy chain aware peripherals:

Generally, peripherals have acknowledge input and request output lines but daisy chain aware peripherals must have additional acknowledge output and request input lines. However if the peripherals do not contain acknowledge output and request input lines then they will not be daisy chain aware peripherals.

But they can be made daisy chain aware by certain logic whose complexity may increase based on complexity of system.

### Ques 9

Why multilevel bus architecture exist? Describe the two level and three level bus system with necessary bus diagram.

Multilevel bus architecture are implemented in the system to improve overall performance of the system.

#### Two level bus system.

Two level bus system consist of a high speed processor, local bus, a low speed peripheral bus and a bridge to connect two buses.

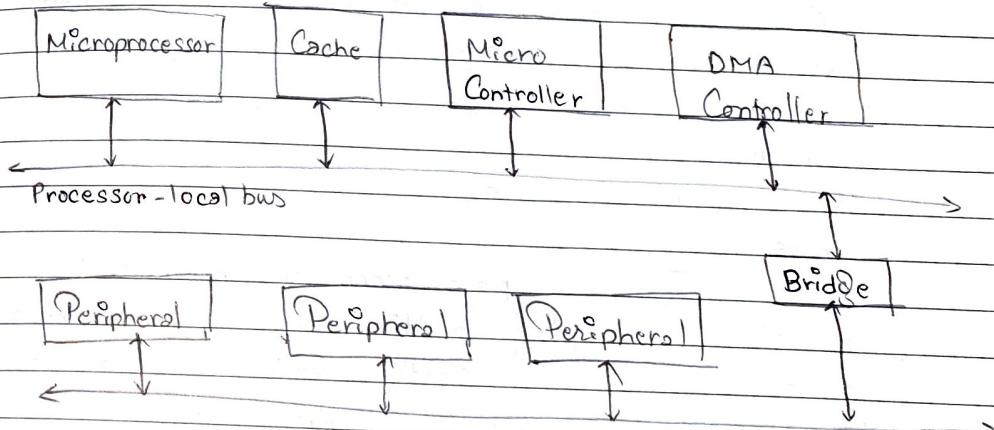


Fig: Two level bus architecture.

The processor-local bus connects very high speed devices such as microprocessor, cache, memory controller, etc. These buses are wide, as wide as memory word.

The peripheral bus connects those peripherals which do not have access to processor local bus. It emphasizes on portability, low power or low gate count. So, the interface of communication is comparatively efficient.

Bridge is a single purpose processor that connects the two buses of the system and also converts communication on one bus to communication on another bus.

### Three level Bus Hierarchy:

It consists of processor local bus, system bus and peripheral bus. A local bus connects the processor to a cache and may support one or more local devices. The system bus acting as high-speed bus, off loads much of the traffic from the processor local bus. And the peripheral bus is used to connect various peripherals in the system.

Qno. 10

Write short notes on:

#### a) Parallel communication.

It takes place when the physical layer is capable of carrying multiple bits of data at a time from one device to another. Each wires carries a single bit of data. The bus is composed of multiple data lines along with control and power lines.

##### Advantages:

- High data throughput
- Less complexity

##### Disadvantages:

- Long parallel wires can result in Ferranti effect. According to this effect, there is a voltage build up due to capacitance and voltage at receiving end is more than that of sending end.

- Small variation in length and cause misalignments as the bit at receiving ends will react at different time.

#### b) Serial communication:

In this communication, a physical layer carries one bit of data at a time. A word of data is transmitted one bit at a time in a single data wires along with control and power line running from one device to another.

##### Advantages:

- It doesn't exhibit Ferranti effect and data misalignment.
- Significant reduction in the size, complexity of the connector and the associated cost.
- Throughput can be better for two distant devices as compared to that of parallel.

##### Disadvantages:

- Interfacing logic and communication protocol is more complex. i.e. On the sending side, a transmitter must decompose data words into bits and on the receiving side, the receiver must compose bits into words.
- For the short distance, its throughput is very less.

#### c) Wireless communication:

Wireless communication eliminates the need for devices to be physically connected in order to communicate.

- The physical layer used in wireless communication is either infrared or radio frequency channel.

#### d) Radio frequency:

It uses the electromagnetic wave frequencies in the radio spectrum. For such communication an analog

circuitry as well as antenna is required at both communicating devices.

Advantages:

- The advantage of using RF is that the line of sight is not necessary and thus longer distance communication is possible.
- The range of communication depends on the transmission power used by transmitter.

### e) Infrared

Infrared uses electromagnetic wave frequencies that are just below the visible light spectrum, thus undetectable by human eyes. These waves can be generated by infrared diode and detected by using an infrared transistor.

Advantages:

- It is relatively cheap to build transmitters and receivers.

Disadvantages:

- Since line of sight is used between the transmitter and receiver. So, range of communication is very restricted.

Qno 11

Describe Inter-IC/ I<sub>2</sub>C/ T<sup>2</sup>C protocol and explain read and write operation with necessary block diagram and timing diagram:

Ans 1  
T<sup>2</sup>C is a serial protocol for two-wire interface to connect low speed device like microcontroller, EEPROM, A/D & D/A converter, I/O interface and other similar peripherals in embedded system.

This protocol enables peripheral ICs (in electric)

and system to communicate with each other using simple communication hardware. The data transfer rates of I<sup>2</sup>C is upto 100k bits/sec and 7 bit addressing are possible.

Since 7 bit addressing are used, the total 128 devices can communicate with single shared I<sup>2</sup>C bus.

I<sup>2</sup>C consist of two wires i.e. a data wire called serial data line (SDA) and a clock wire called Serial-clock-line (SCL).

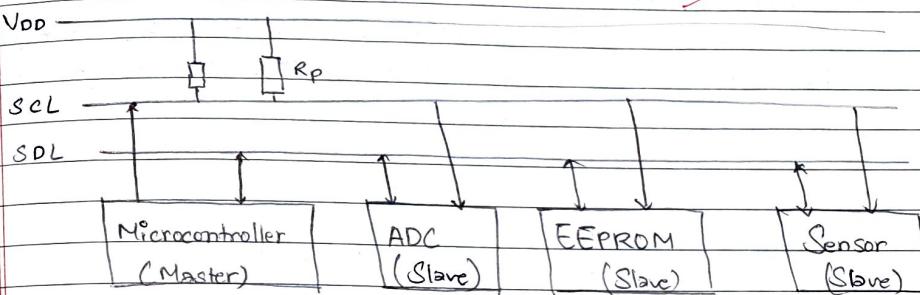


Fig: I<sup>2</sup>C bus structure.

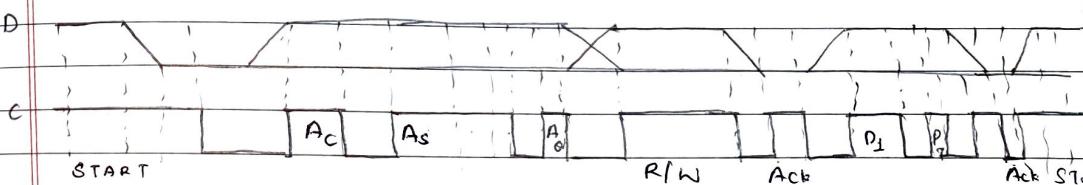


Fig. Timing diagram of typical read/write cycle.  
Operation:

- The master initiates the transfer with a start condition. A start condition is represented by high to low transition of SDA line while the SCL is held high.
- The address of the device to which the data is to be written is sent with most significant bit

- down to the least significant bit.
- For write operation, the master sends a zero after sending the address and the slave acknowledge the transmission by holding the SDA line low during first Ack clock cycle.
- Next the master transmit a byte of data with most significant bit as the first bit.
- The slave acknowledge the reception of data by holding SDA line low during the second Ack clock cycle.
- Finally, master terminates the transfer by generating a stop condition. Stop condition is represented by a low to high transition of SDA line while the SCL is held high.

~~stop condition~~  
stop condition