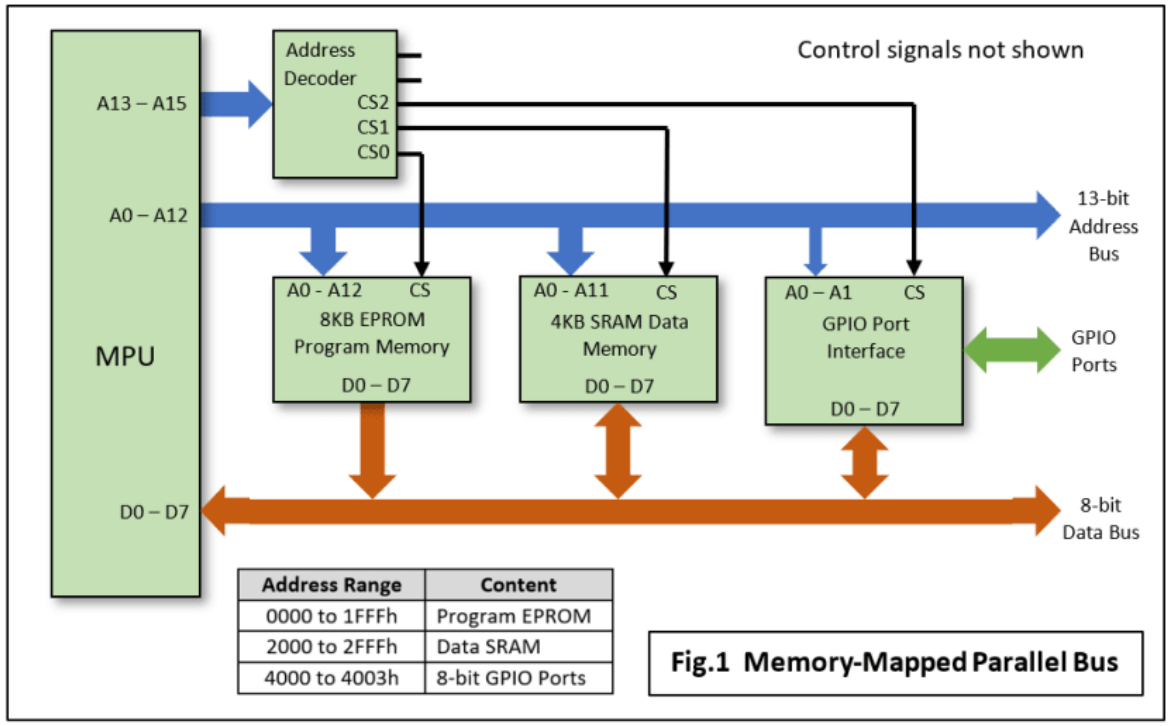
Communication BUS Concept

Communication BUS Concept - Terminology

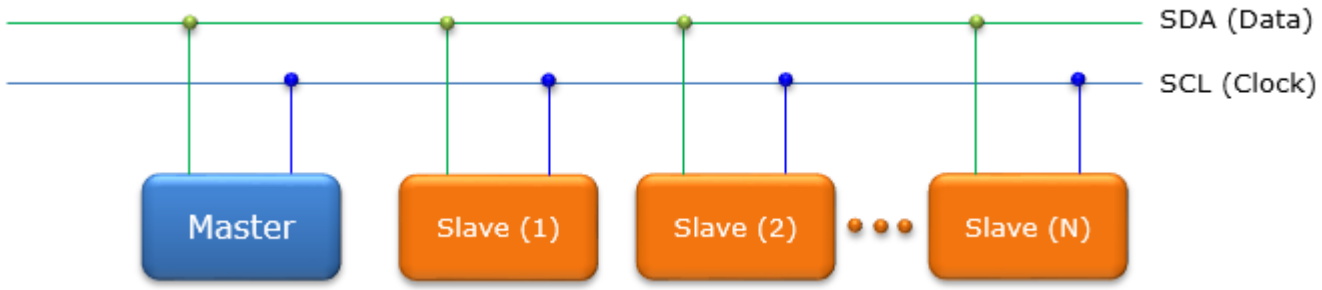
- A bus typically consists of three types of signal lines:
  - Address
    - Carry address of destination for which transfer is initiated
    - Can be shared or separate for read, write data
  - Data
    - Carry information between source and destination components
    - Can be shared or separate for read, write data
    - Choice of data width critical for application performance
  - Control
    - Requests and acknowledgements
    - Specify more information about type of data transfer

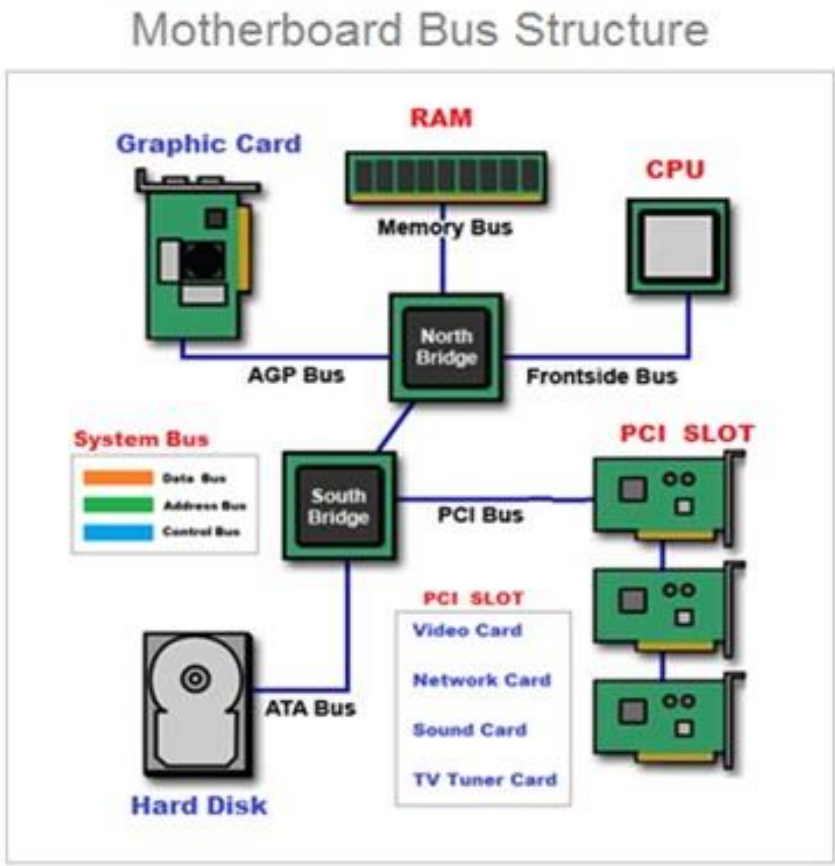
Communication BUS Concept - Terminology



Address Range	Content
0000 to 1FFFh	Program EPROM
2000 to 2FFFh	Data SRAM
4000 to 4003h	8-bit GPIO Ports

Fig.1 Memory-Mapped Parallel Bus



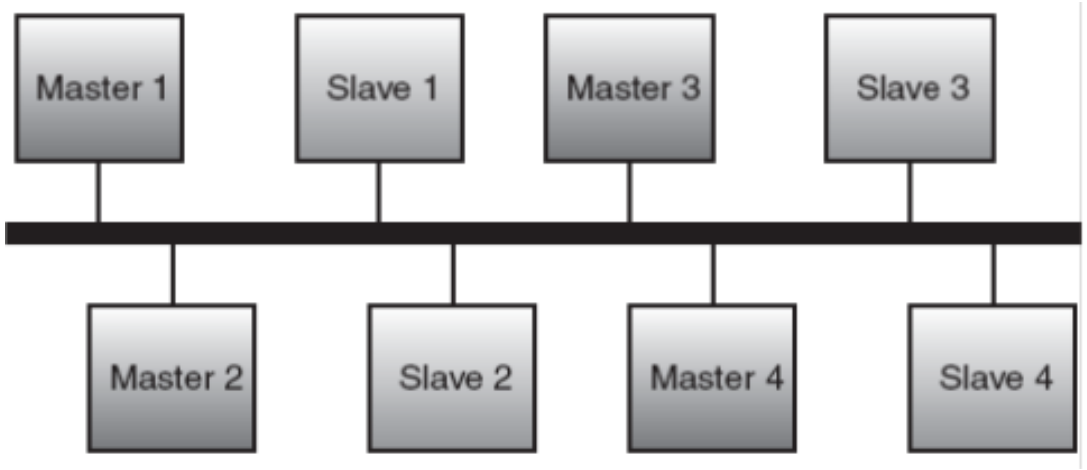


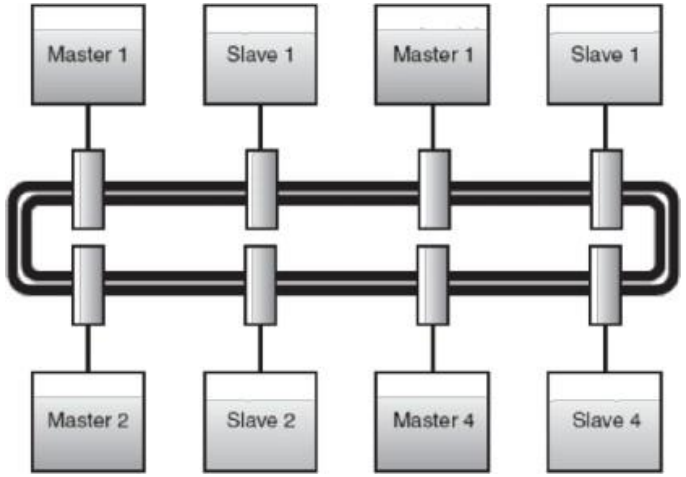
Communication BUS Concept - Terminology

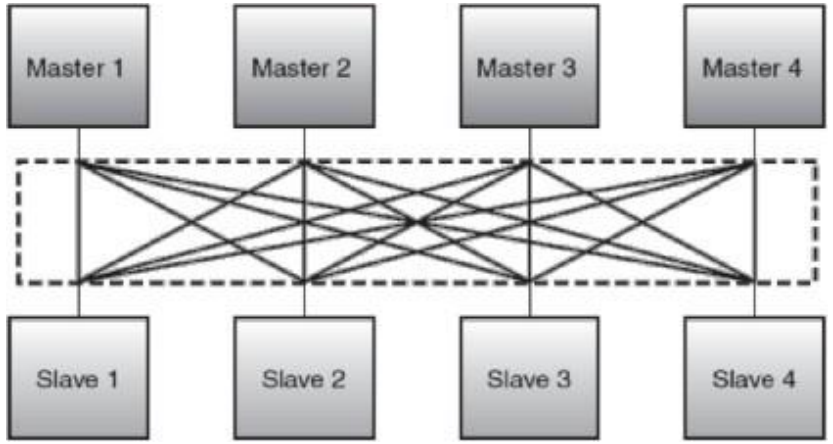
- Master (or Initiator)
  - IP component that initiates a read or write data transfer
- Slave (or Target)
  - IP component that does not initiate transfers and only responds to incoming transfer requests
- Arbiter
  - Controls access to the shared bus
  - Uses an arbitration scheme to select master to grant access to bus
- Decoder
  - Determines which component a transfer is intended for
- Bridge
  - Connects two buses
  - Acts as slave on one side and master on the other

BUS Topologies

- Shared Bus
  - Simple traditional structure
- Hierarchical Shared Bus
  - Improves system throughput
  - Multiple simultaneous transfers
- Full crossbar bus (Mesh)
- Ring bus

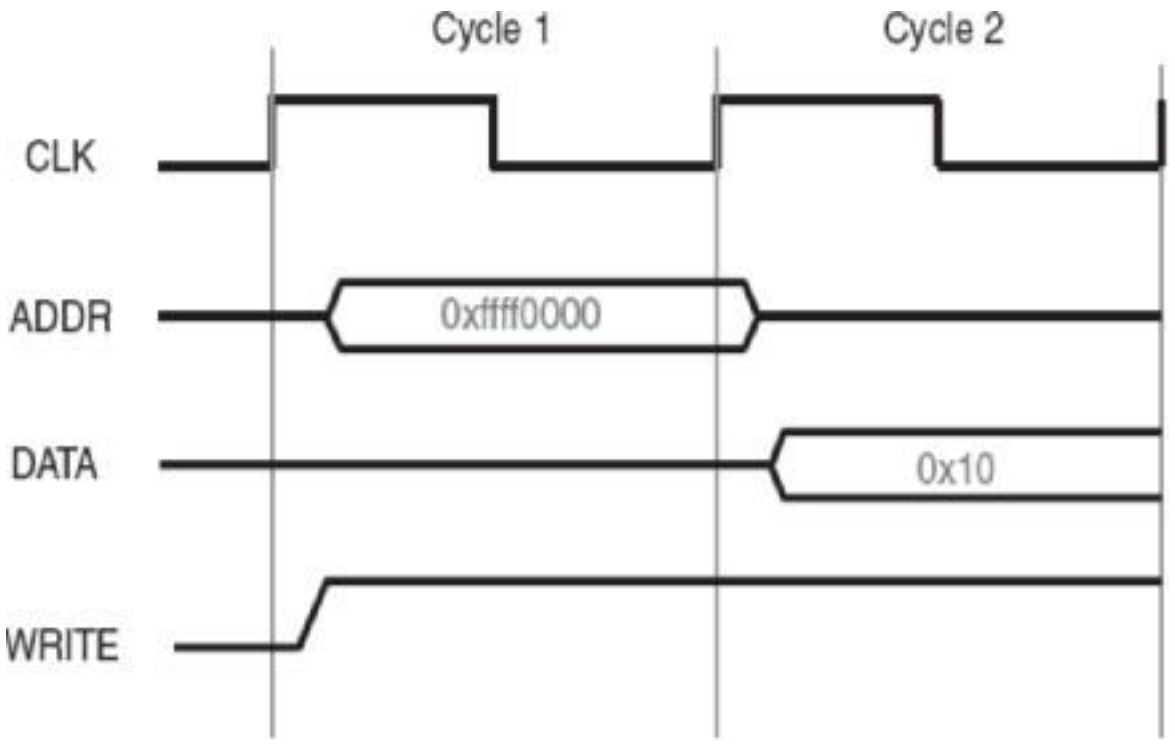






BUS Clocking

- Synchronous Bus
  - Includes a clock in control lines
  - Fixed protocol for communication that is relative to clock
  - Involves very little logic and can run very fast
  - Requires frequency converters across frequency domains

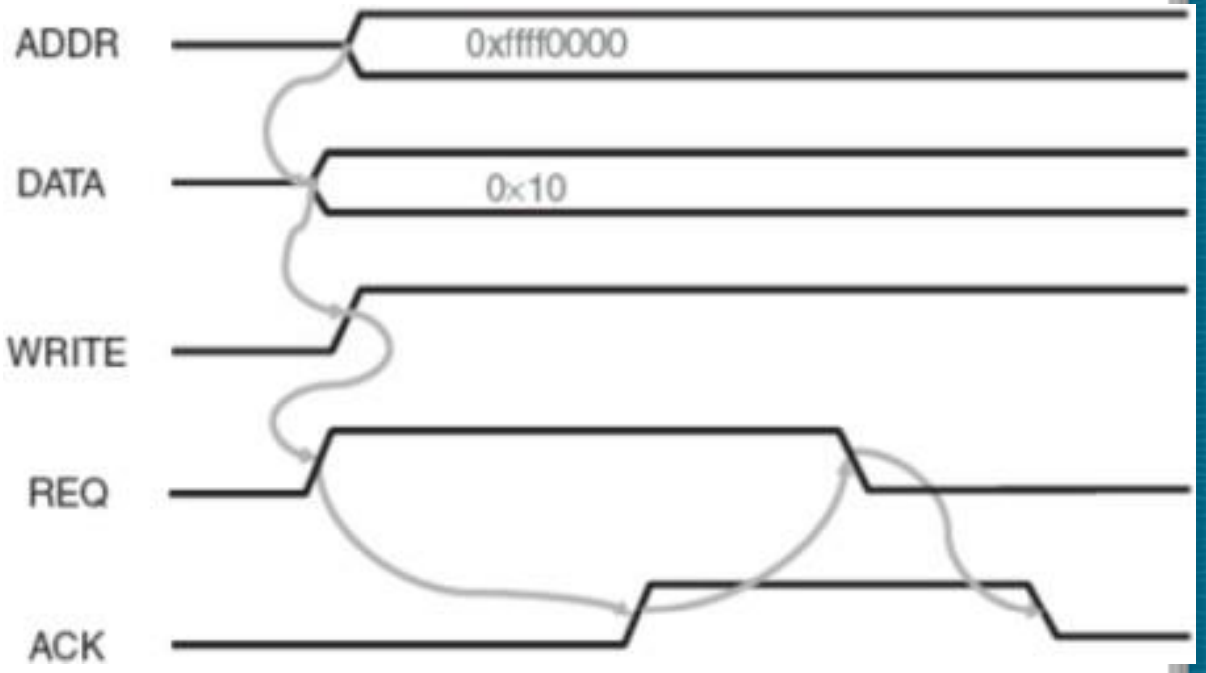


Synchronous Communication

- Advantages
  - It helps you to transfer a large amount of data.
  - It offers real-time communication between connected devices.
  - Each byte is transmitted without a gap between the next byte.
  - It also reduces time time timing errors.
- Disadvantages
  - The accuracy of the received data depends on the receiver's ability to count the received bits accurately.
  - The transmitter and receiver need to operate simultaneously with the same clock frequency.

BUS Clocking

- Asynchronous Bus
  - Not clocked
  - Requires a handshaking protocol
    - performance not as good as that of synchronous bus
    - No need for frequency converters, but does need extra lines
  - Does not suffer from clock signal related issues like clock skew

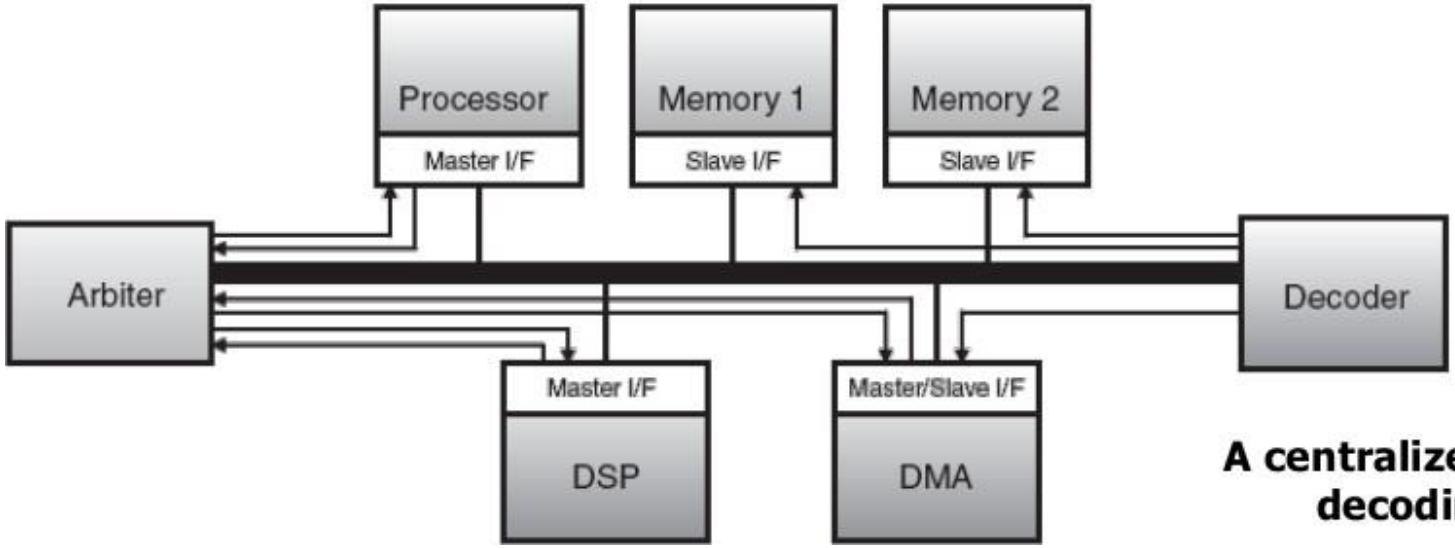


Asynchronous communication

- Advantages
  - This is a highly flexible method of data transmission.
  - Synchronization between the receiver and transmitter is unnecessary.
  - It helps you to transmit signals from the sources which have different bit rates.
  - The Transmission can resume as soon as the data byte transmission is available.
  - This mode of Transmission is easy for implementation.
- Disadvantages
  - In Asynchronous Transmission, additional bits called start and stop bits are required to be used.
  - The timing error may take place as it is difficult to determine synchronicity.
  - It has a slower transmission rate.
  - May create false recognition of these bits because of noise on the channel.

BUS Decoding and Arbitration

- A bus implementation includes logic for both decoding and arbitration :
  - Decoding – determining the target for any transfer initiated by the master
  - Arbitration – deciding which master can use the shared bus if more than one master requested bus access simultaneously

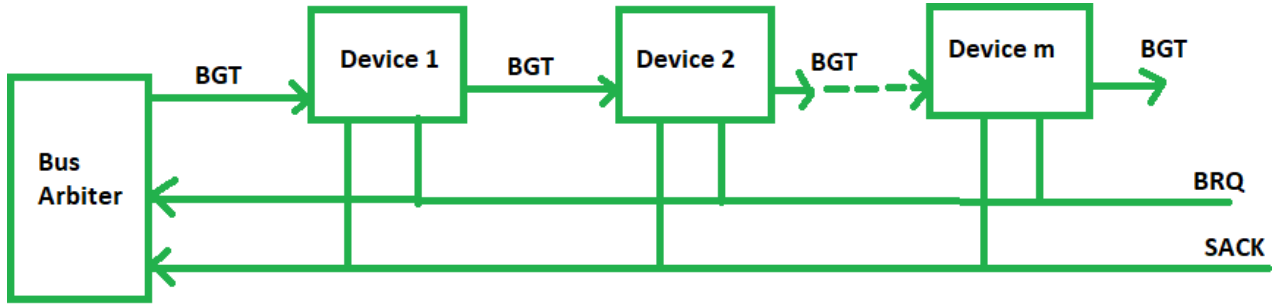


A centralized arbitration / decoding scheme

BUS Arbitration

- Bus arbitration is the process by which the next device to become the bus master is selected and bus mastership is transferred to it. The selection of bus master is usually done on the priority basis. There are two approaches to bus arbitration: Centralized and
- distributed.
  - Centralized Arbitration
    - In centralized bus arbitration, a single bus arbiter performs the required arbitration. The bus arbiter may be the processor or a separate controller connected to the bus.
    - There are three different arbitration schemes that use the centralized bus arbitration approach. There schemes are: (a.) Daisy chaining (b.) Polling method (c.) Independent request

BUS Arbitration

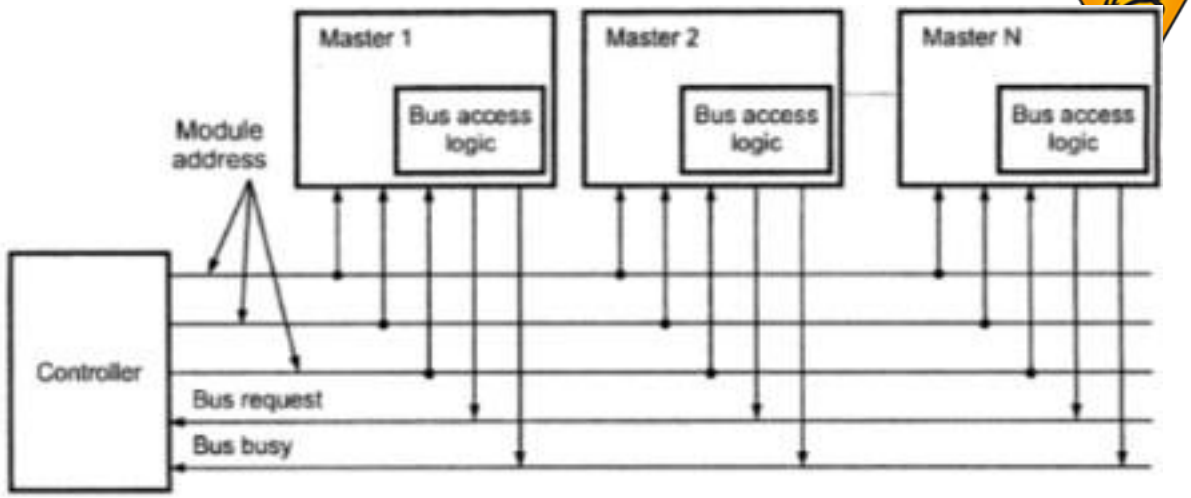


Daisy chained bus arbitration

- Daisy chaining
  - Advantages
    - Simplicity and Scalability.

    The user can add more devices anywhere along the chain, up to a certain maximum value.
  - Disadvantages
    - The value of priority assigned to a device is depends on the position of master bus.
    - Propagation delay is arises in this method.
    - If one device fails then entire system will stop working.

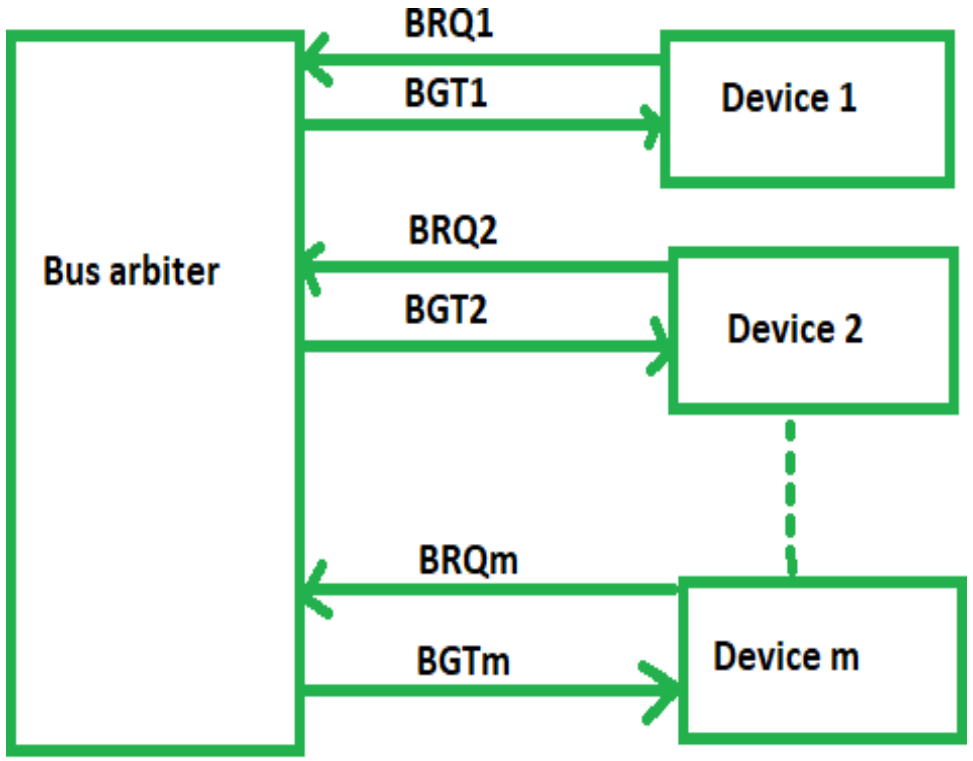
BUS Arbitration



- **Polling**
  - Advantages
    - This method does not favor any particular device and processor
    - The method is also quite simple.
    - If one device fails then entire system will not stop working.
  - Disadvantages
    - Adding bus masters is difficult as increases the number of address lines of the circuit.

BUS Arbitration

- **Fixed priority or Independent Request method –**
  - Advantages
    - Generates fastest response
  - Disadvantages
    - High hardware cost
    - Large Number of lines required



Fixed priority bus arbitration method

BUS Arbitration

- **Distributed Arbitration**
  - In distributed arbitration, all devices participate in the selection of the next bus master.
  - In this scheme each device on the bus is assigned a4-bit identification number.
  - The number of devices connected on the bus when one or more devices request for the control of bus, they assert the start-arbitration signal and place their 4-bit ID numbers on arbitration lines, ARB0 through ARB3.

BUS Arbitration

- **Distributed Arbitration cont...**
  - These four arbitration lines are all open-collector. Therefore, more than one device can place their 4-bit ID number to indicate that they need to control of bus. If one device puts 1 on the bus line and another device puts 0 on the same bus line, the bus line status will be 0. Device reads the status of all lines through inverters buffers so device reads bus status 0as logic 1. Scheme the device having highest ID number has highest priority.
    When two or more devices place their ID number on bus lines then it is necessary to identify the highest ID number on bus lines then it is necessary to identify the highest ID number from the status of bus line. Consider that two devices A and B, having ID number 1 and 6,
  - respectively are requesting the

BUS Arbitration

- **Distributed Arbitration cont...**
  - Device A puts the bit pattern 0001, and device B puts the bit pattern 0110. With this combination the status of bus-line will be 1000; however because of inverter buffers code seen by both devices is 0111.
    - Each device compares the code formed on the arbitration line to its own ID, starting from the most significant bit. If it finds the difference at any bit position, it disables its drives at that bit position and for all lower-order bits.
    - It does so by placing a 0 at the input of their drive. In our example, device detects a different on line ARB2 and hence it disables its drives on line ARB2, ARB1 and ARB0. This causes the code on the arbitration lines to change to 0110. This means that device B has won the race.

BUS
Arbitration

- **Distributed Arbitration cont...**
  - The decentralized arbitration offers high reliability because operation of the bus is not dependent on any single device.

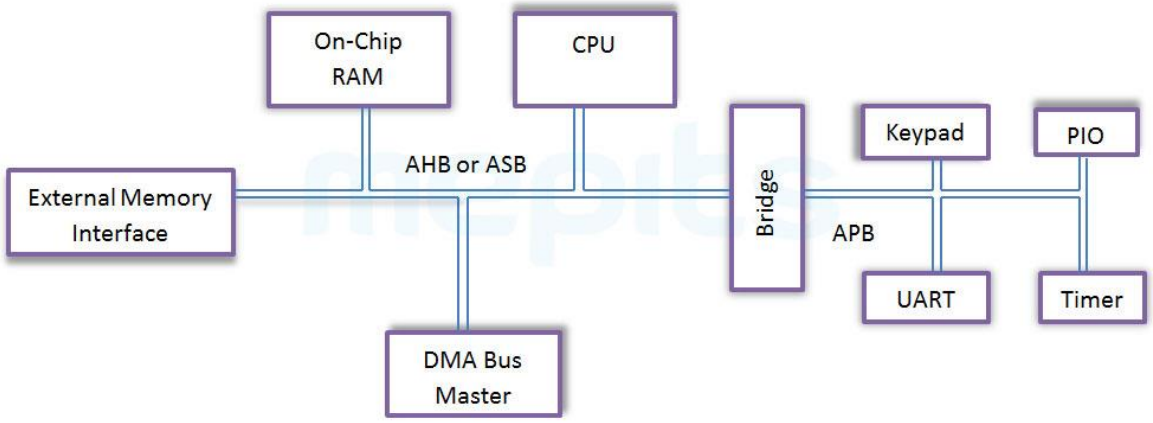
BUS Standards

- Bus standards are useful for defining a specific interface and data transfer protocol

- In reality, several competing standards for SoC design:
  -AMBA (ARM) - Advanced Microcontroller Bus Architecture
  -CoreConnect (IBM)
  -Wishbone (OpenCores)
  -Avalon (Altera)
  -Quick Path (Intel)
  -HyperTransport (AMD)
  -STBus (STMicroelectronics)

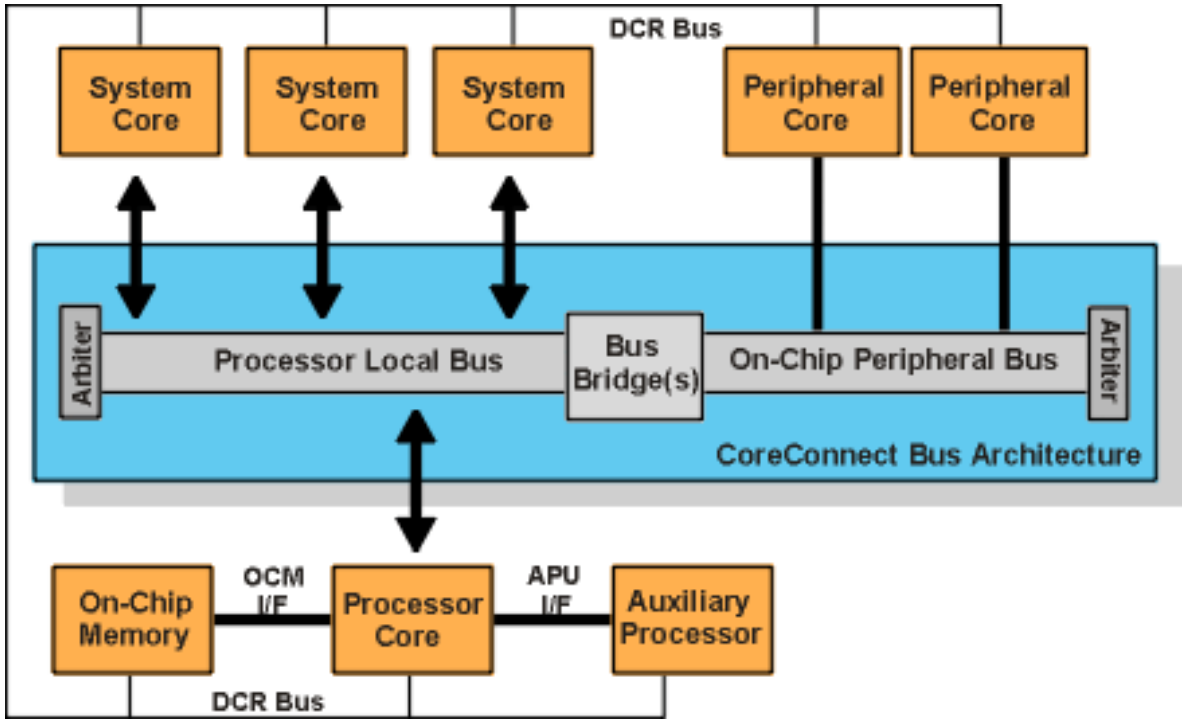
Advanced Microcontroller Bus Architecture

- is an open-standard, on-chip interconnect specification for the connection and management of functional blocks in system-on-a-chip (SoC) designs
- It facilitates development of multi-processor designs with large numbers of controllers and components with a bus architecture.
- AMBA was introduced by ARM in 1996.
- The first AMBA buses were the Advanced System Bus (ASB) and the Advanced Peripheral Bus (APB).



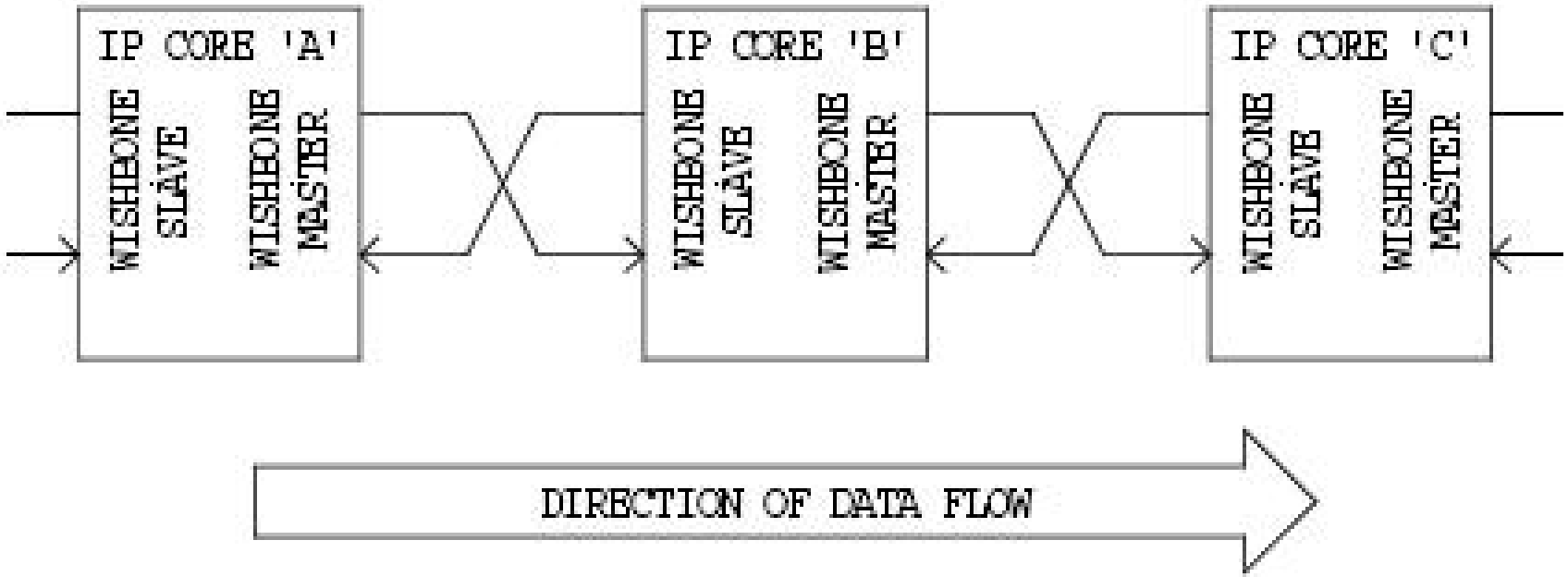
CoreConnect

- CoreConnect is a microprocessor bus-architecture from IBM for system-on-a-chip (SoC) designs
- It was designed to ease the integration and reuse of processor, system, and peripheral cores



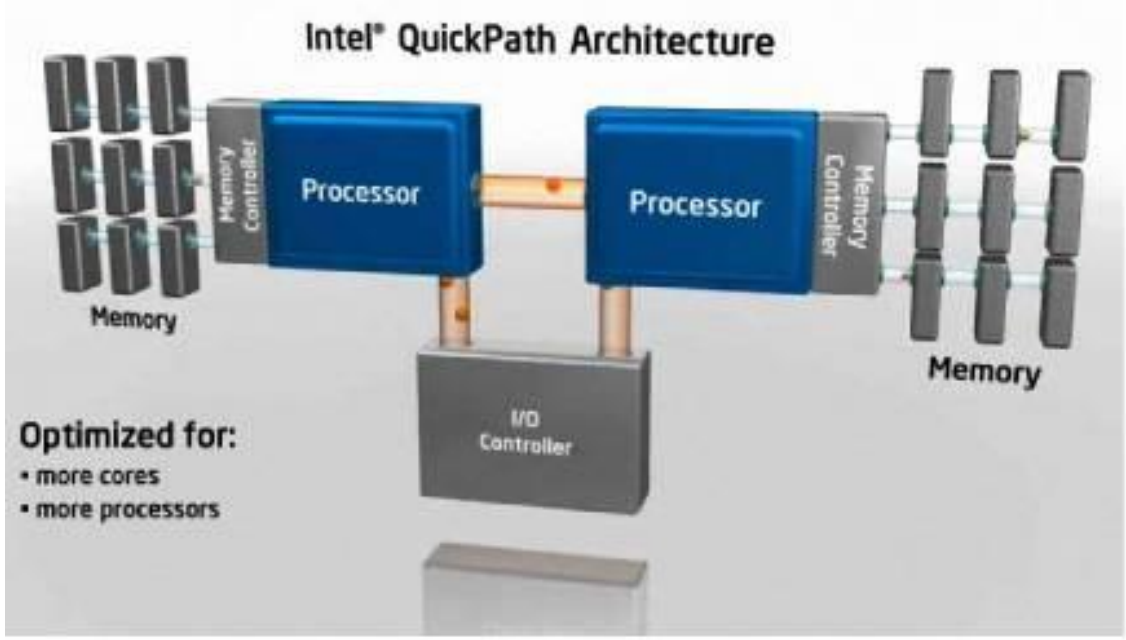
Wishbone (OpenCores)

- The Wishbone Bus is an open source hardware computer bus intended to let the parts of an integrated circuit communicate with each other.



Quick Path (Intel)

- is a point-to-point processor interconnect developed by Intel which replaced the front-side bus (FSB) in Xeon, Itanium, and certain desktop platforms starting in 2008



Hyper Transport (AMD)

- formerly known as **Lightning Data Transport** (**LDT**), is a technology for interconnection of computer processors.
- It is a bidirectional serial/parallel high-bandwidth, low-latency point-to-point link that was introduced on April 2, 2001
- System bus architecture of AMD central processing units (CPUs) from Athlon 64 through AMD FX