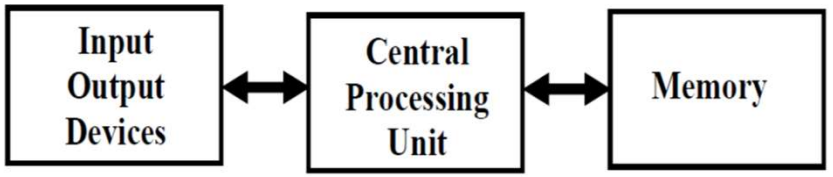
Memory Organization

Outline

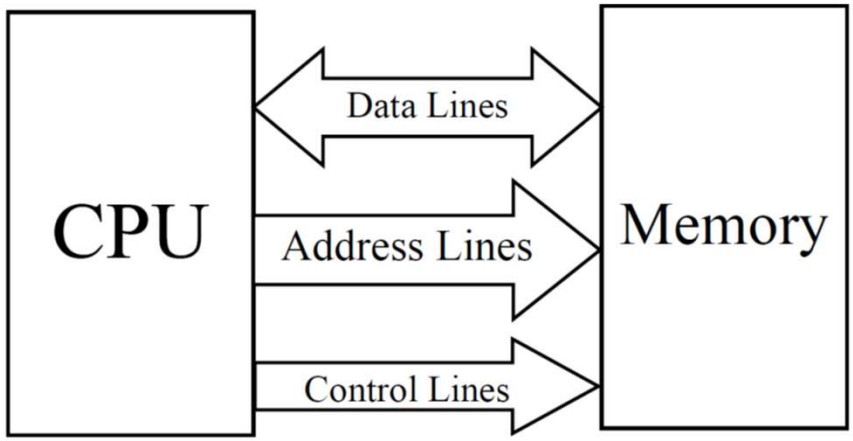
- Memory Organization
  - Memory
  - Memory Interfacing

Memory Architecture

- The Memory stores the instructions as well as data. No one can distinguish an instruction and data. The CPU has to be directed to the address of the instruction codes.
- The memory is connected to the CPU through the following lines
  - Address
  - Data
  - Control

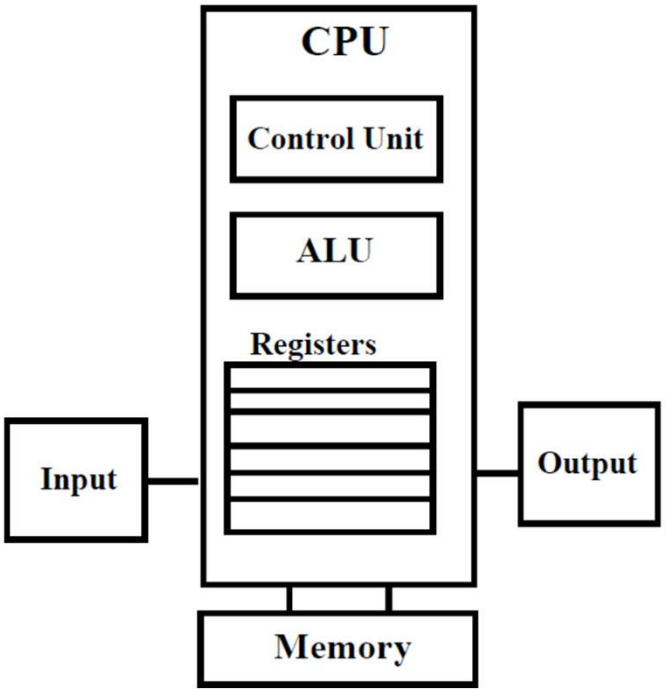


The Von Neumann Architecture

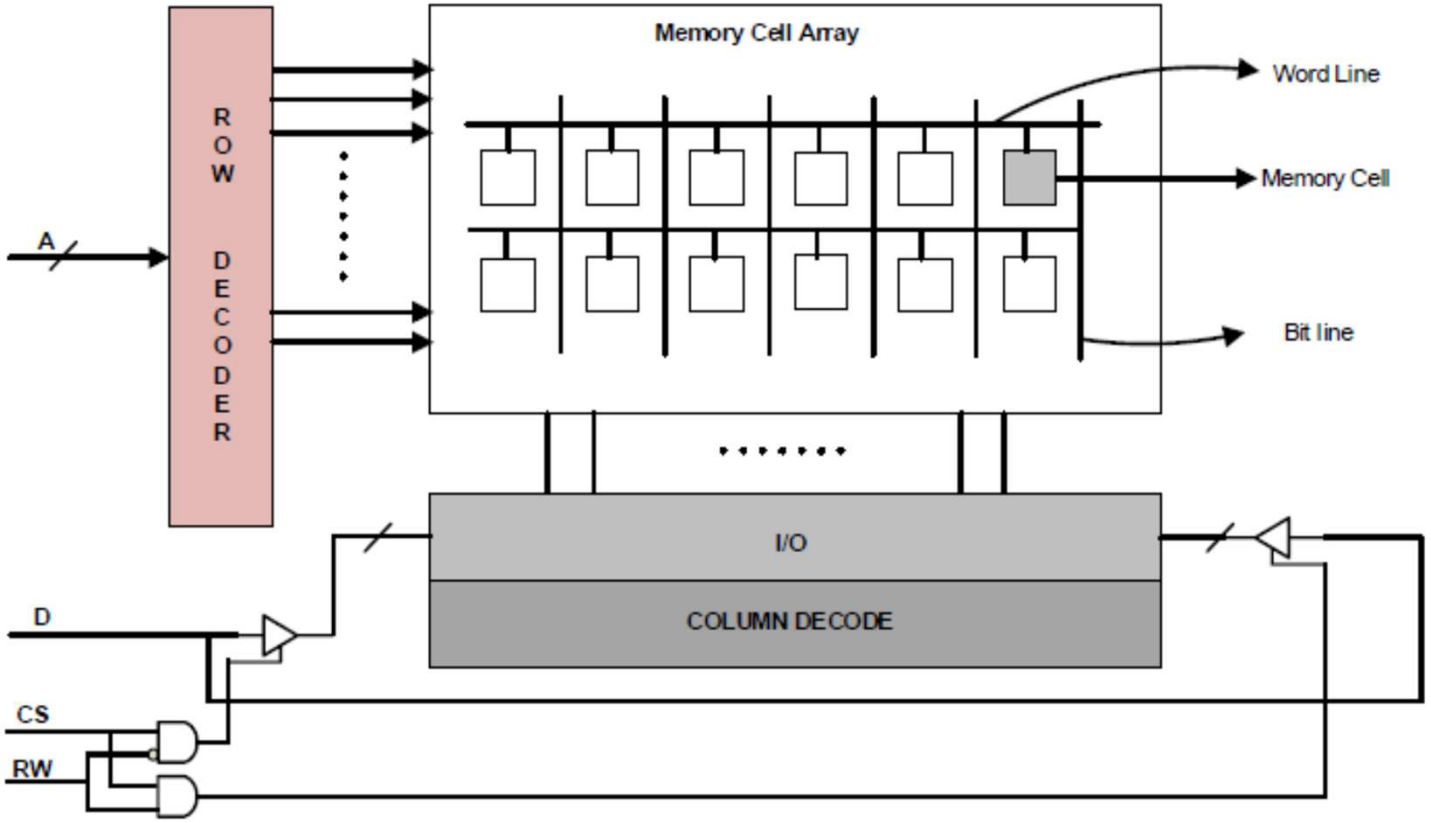


Memory Architecture

- In a memory read operation the CPU loads the address onto the address bus. The CPU then sends a read control signal. The data is stored in that location is transferred to the processor via the data lines. In the memory write operatior after the address is loaded the CPU sends the write control signal followed by the data to the requested memory location.
- The memory at the basic level can be classified as
  - Processor Memory (Register Array)
  - Internal on-chip Memory
  - Primary Memory
  - Cache Memory
  - Secondary Memory

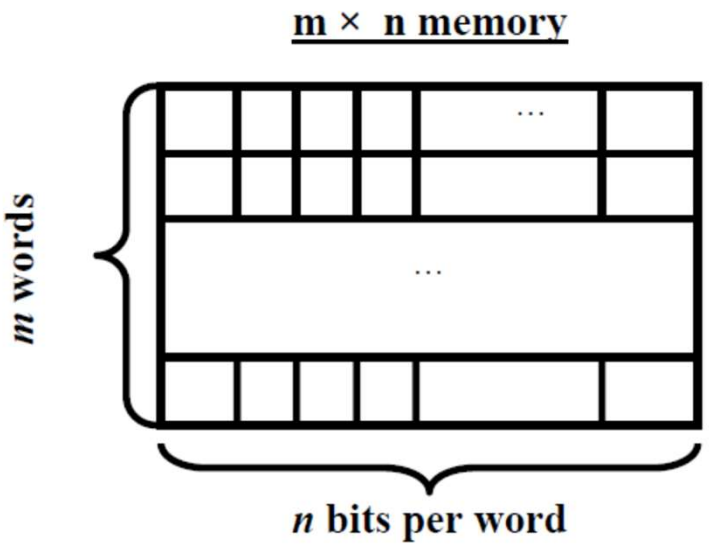


Memory Architecture



Data Storage

- An *m* word memory can store *m* x *n*: *m* words of *n* bits each. One word is located at one address therefore to address *m* words we need.
  - k = Log2(*m*) address input signals
  - or k number address lines can address $m = 2^k$ words
- **Example** 4,096 x 8 memory:
  - 32,768 bits
  - 12 address input signals
  - 8 input/output data signals



Memory Specification

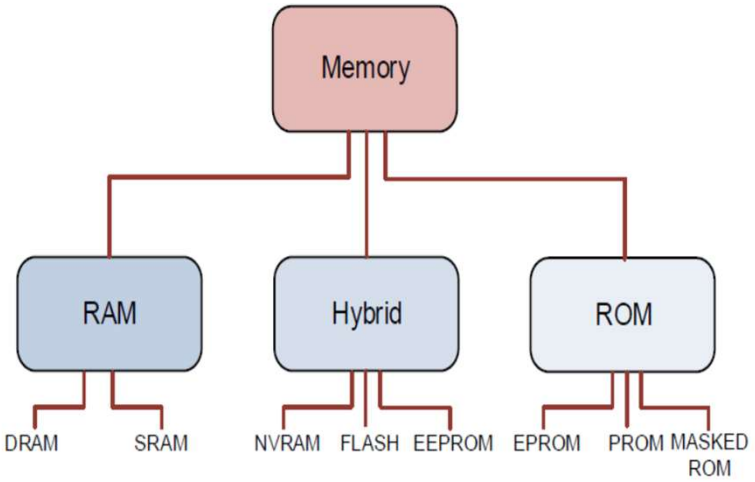
- There are two important specifications for the Memory as far as Real Time Embedded Systems are concerned.
  - Write Ability
  - Storage Performance
- **Write ability**
  - It is the manner and speed that a particular memory can be written
    - Ranges of write ability
      - High end
        - processor writes to memory simply and quickly e.g., RAM
      - Middle range
        - processor writes to memory, but slower e.g., FLASH, EEPROM (Electrically Erasable and Programmable Read Only Memory)
      - Lower range
        - special equipment, “programmer”, must be used to write to memory e.g., EPROM, OTP ROM (One Time Programmable Read Only Memory)
      - Low end
        - bits stored only during fabrication e.g., Mask-programmed ROM
  - In-system programmable memory
    - Can be written to by a processor in the embedded system using the memory
    - Memories in high end and middle range of write ability

Storage permanence

- It is the ability to hold the stored bits.
- Range of storage permanence
  - High end
    - essentially never loses bits e.g., mask-programmed ROM
  - Middle range
    - holds bits days, months, or years after memory's power source turned off e.g., NVRAM
  - Lower range
    - holds bits as long as power supplied to memory e.g., SRAM
  - Low end
    - begins to lose bits almost immediately after written e.g., DRAM

Memory Classification

- EEPROMs are electrically-erasable-and-programmable. Internally, they are similar to EPROMs, but the erase operation is accomplished electrically, rather than by exposure to ultraviolet light.
- Flash memory combines the best features of the memory devices. Flash memory devices are high density, low cost, nonvolatile, fast (to read, but not to write), and electrically reprogrammable. Thus Flash offers significant advantages and, as a direct result, the use of flash memory has increased dramatically in embedded systems.



- The third member of the hybrid memory class includes NVRAM (non-volatile RAM). Non-volatility is also a characteristic of the ROM. Logically an NVRAM is just an SRAM with a battery backup. When the power is turned on, the NVRAM operates just like any other SRAM. When the power is turned off, the NVRAM draws just enough power from the battery to retain its data. NVRAM is fairly common in embedded systems.

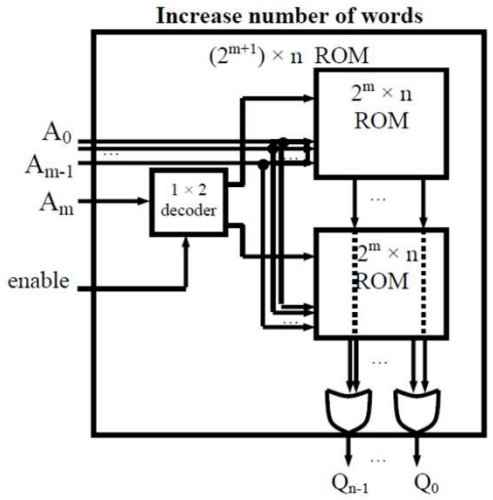
Common Memory Types

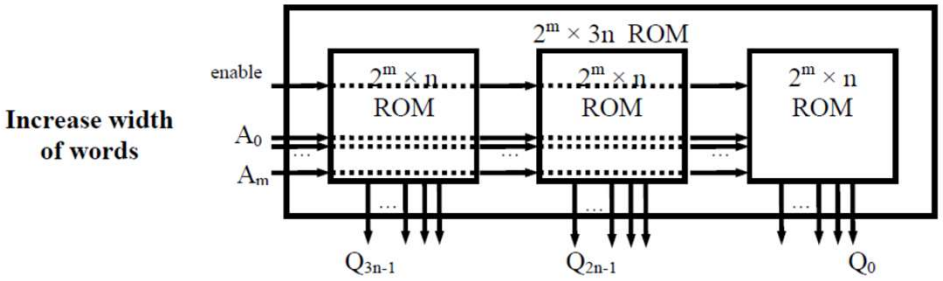
- Read Only Memory (ROM)
  - Mask-programmed ROM
  - OTP ROM: One-time programmable ROM
  - EPROM: Erasable programmable ROM
  - EEPROM
- Flash Memory
- RAM: “Random-access” memory
  - SRAM: Static RAM
    - Memory cell uses flip-flop to store bit
    - Requires 6 transistors
    - Holds data as long as power supplied
  - DRAM: Dynamic RAM
    - Memory cell uses MOS transistor and capacitor to store bit
    - More compact than SRAM
    - “Refresh” required due to capacitor leak
    - word’s cells refreshed when read
  - Typical refresh rate 15.625 microsec.
  - Slower to access than SRAM

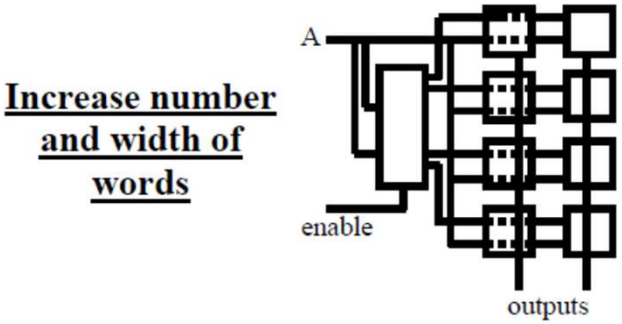
Composing memory

- Memory size needed often differs from size of readily available memories
- When available memory is larger, simply ignore unneeded high-order address bits and higher data lines
- When available memory is smaller, compose several smaller memories into one larger memory
  - Connect side-by-side to increase width of words
  - Connect top to bottom to increase number of words
  - added high-order address line selects smaller memory containing desired word using a decoder
  - Combine techniques to increase number and width of words

  The figures are given in next slides.

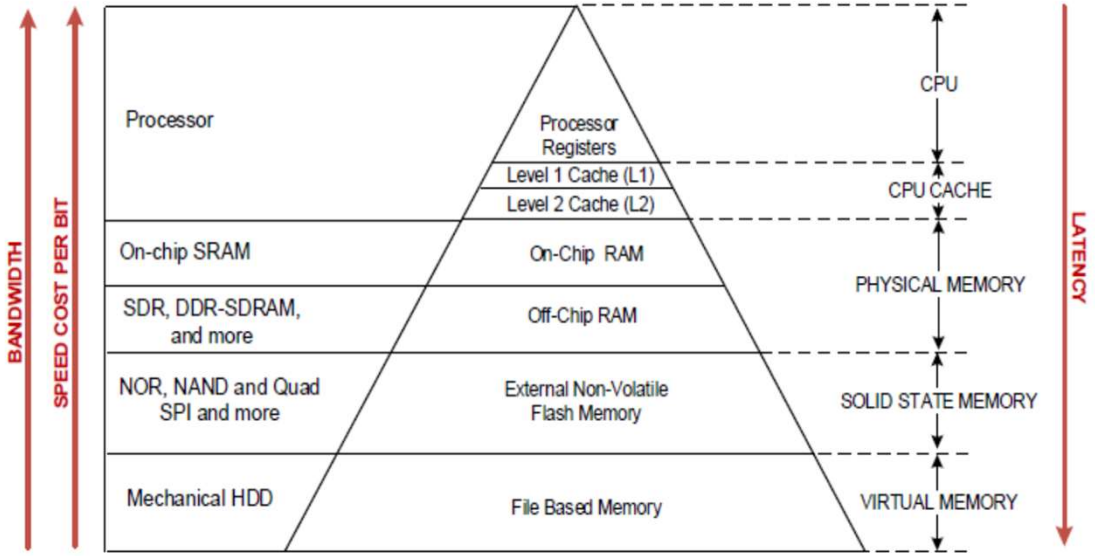

Increase number of words
$(2^{m+1}) \times n$ ROM
$2^m \times n$ ROM
A_0
A_{m-1}
A_m
1 × 2 decoder
enable
$2^m \times n$ ROM
Q_{n-1}
Q_0

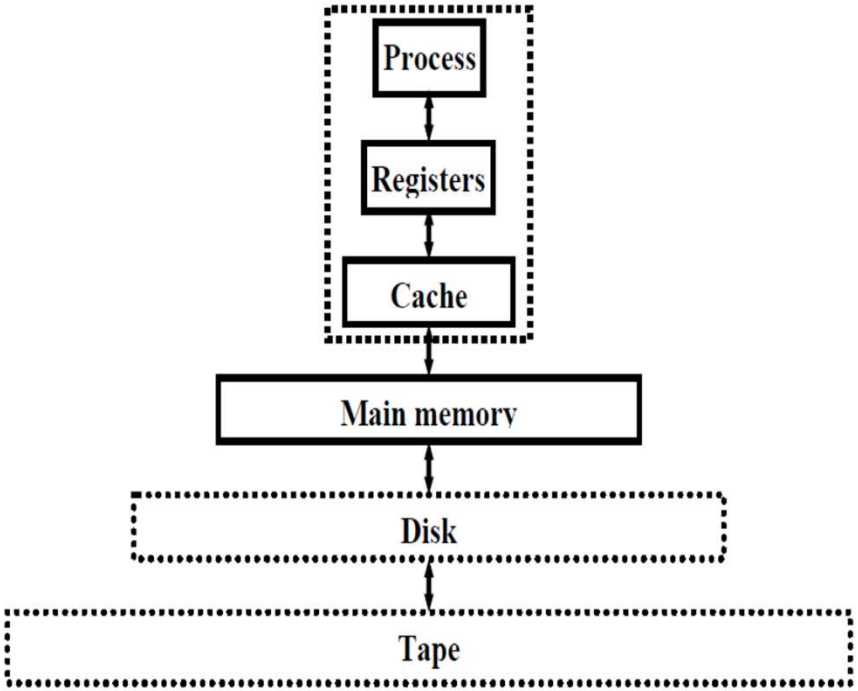

Increase width of words
$2^m \times 3n$ ROM
enable
$2^m \times n$ ROM
$2^m \times n$ ROM
$2^m \times n$ ROM
A_0
A_m
Q_{3n-1}
Q_{2n-1}
Q_0


Increase number and width of words
A
enable
outputs

Memory Hierarchy

- Main memory
  - Large, inexpensive, slow memory stores entire program and data
- Cache
  - Small, expensive, fast memory stores copy of likely accessed parts of larger memory
  - Can be multiple levels of cache





Cache Memory

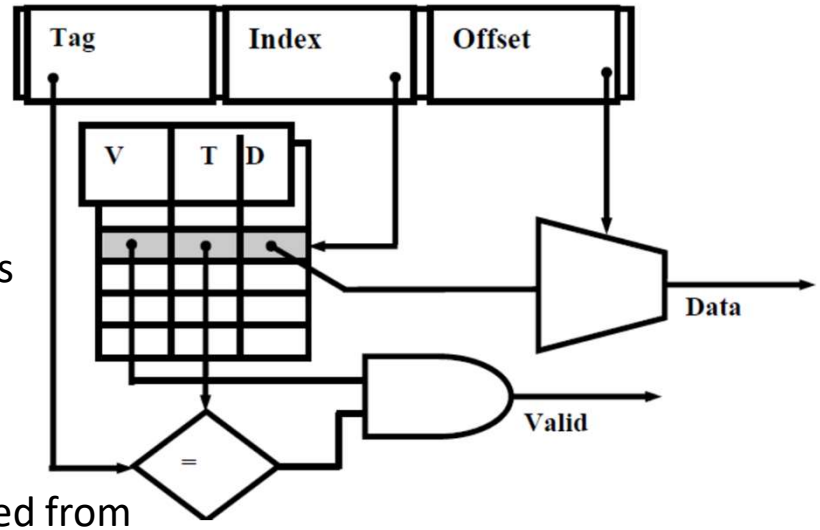
- Cache
  - Usually designed with SRAM & faster but more expensive than DRAM (a cycle vs. several cycle)
  - Usually on same chip as processor with space limited, so much smaller than off-chip main memory
- Cache operation
  - Request for main memory access (read or write) & First, check cache for copy cache hit e.g.- copy is in cache, quick access or cache miss - copy not in cache, read address and possibly its neighbors into cache
- Several cache design choices
  - cache mapping, replacement policies, and write techniques

Cache Memory – *Cache Mapping*

- It is necessary as there are far fewer number of available cache addresses than the memory
- Cache mapping used to assign main memory address to cache address and determine hit or miss
- Three basic techniques:
  - Direct mapping
  - Fully associative mapping
  - Set-associative mapping
- Caches partitioned into indivisible blocks or lines of adjacent memory addresses
  - usually 4 or 8 addresses per line

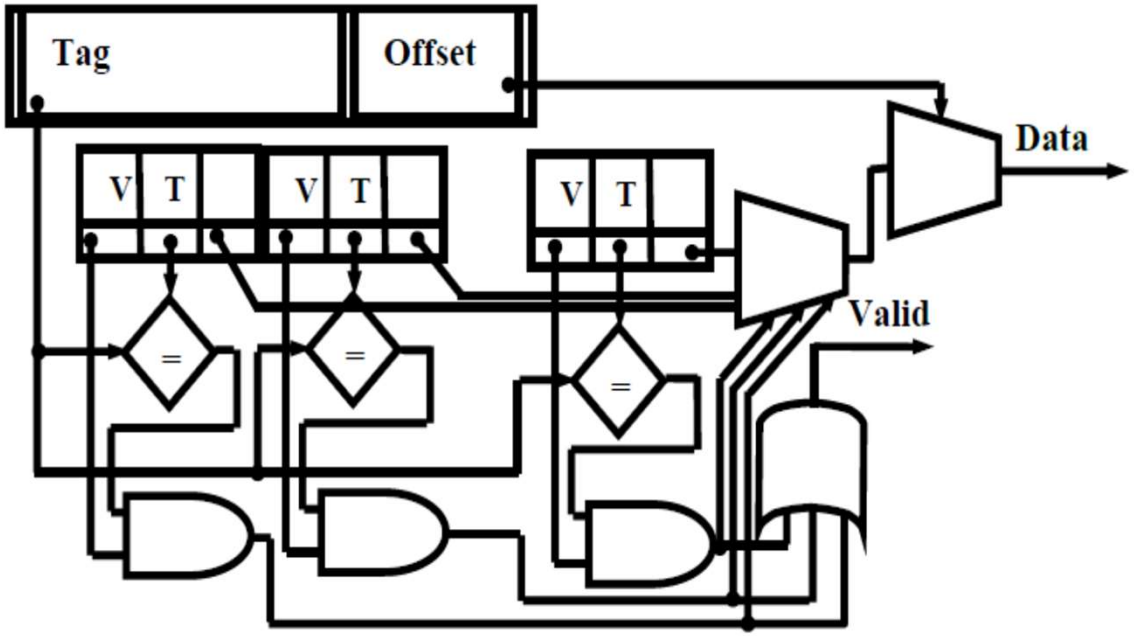
Direct Mapping

- Main memory address divided into 2 fields
  - Index which contains
    - cache address
    - number of bits determined by cache size
  - Tag
    - compared with tag stored in cache at address indicated by index
    - if tags match, check valid bit
  - Valid bit
    - indicates whether data in slot has been loaded from memory
  - Offset
    - used to find particular word in cache line



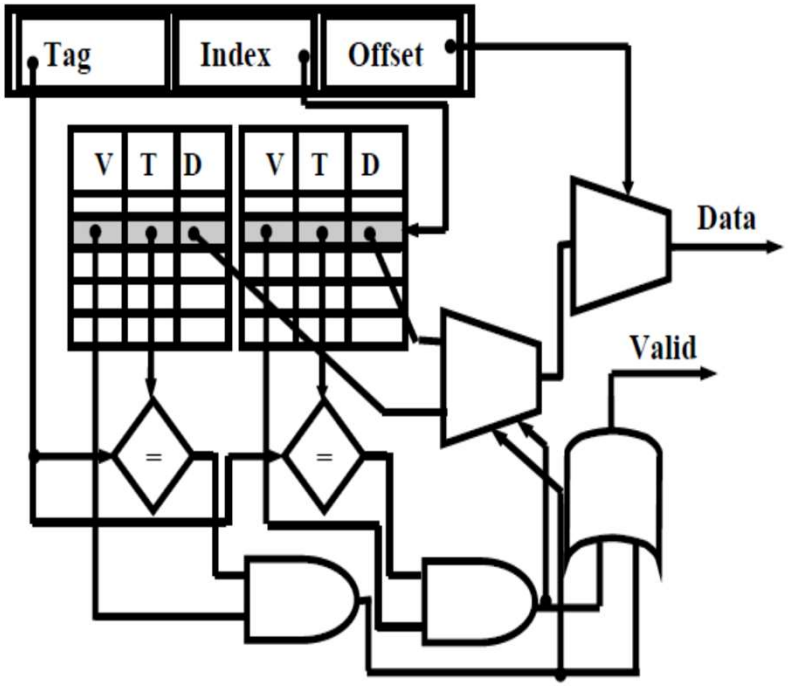
Fully Associative Mapping

- Complete main memory address stored in each cache address
- All addresses stored in cache simultaneously compared with desired address
- Valid bit and offset same as direct mapping



Set-Associative Mapping

- Compromise between direct mapping and fully associative mapping
- Index same as in direct mapping
- But, each cache address contains content and tags of 2 or more memory address locations
- Tags of that set simultaneously compared as in fully associative mapping
- Cache with set size N called N-way set-associative
  - 2-way, 4-way, 8-way are common



Cache replacement and writing

- **Cache-Replacement Policy**
  - Technique for choosing which block to replace
    - when fully associative cache is full or when set-associative cache's line is full
  - Direct mapped cache has no choice
  - Random
    - replace block chosen at random
  - LRU: least-recently used
    - replace block not accessed for longest time
  - FIFO: first-in-first-out
    - push block onto queue when accessed & choose block to replace by popping queue

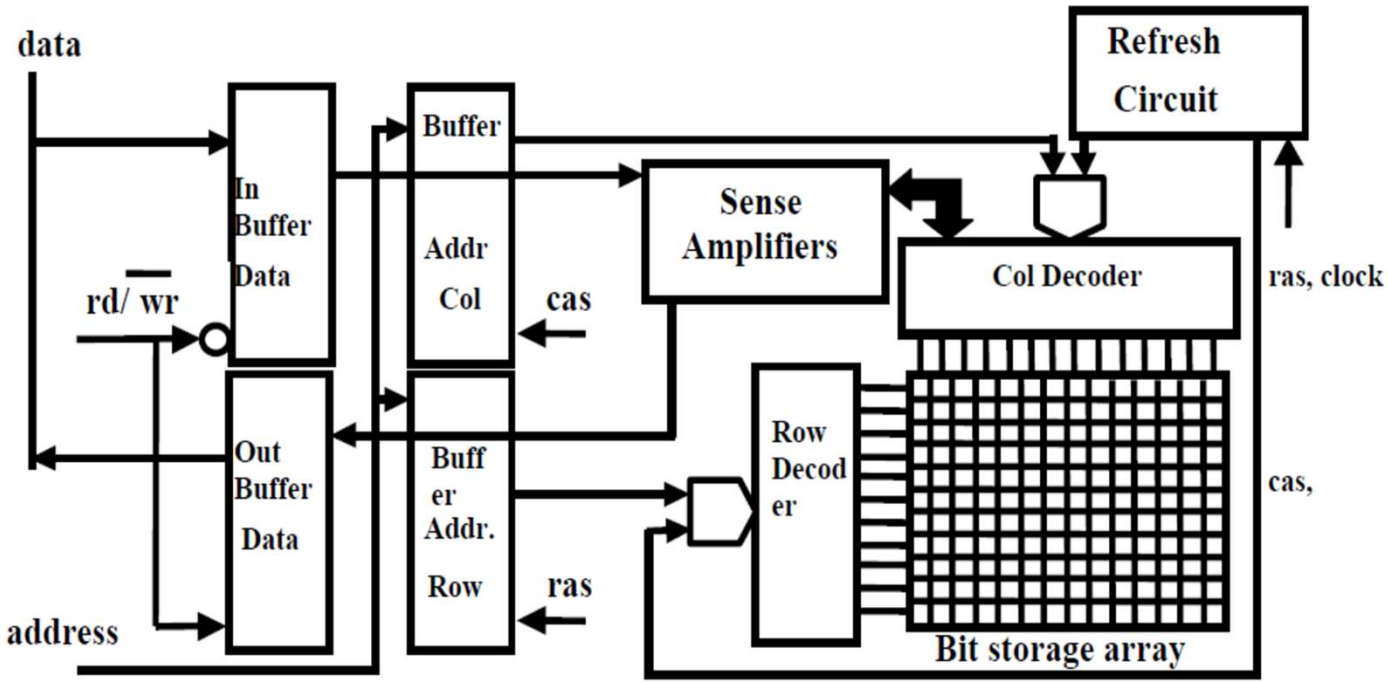
Cache replacement and writing

- **Cache Write Techniques**
  - When written, data cache must update main memory
  - Write-through
    - write to main memory whenever cache is written to & processor must wait for slower main memory write
    - easiest to implement & potential for unnecessary writes
  - Write-back
    - main memory only written when “dirty” block replaced & extra dirty bit for each block set when cache block written to
    - reduces number of slow main memory writes

DRAMs

- DRAMs commonly used as main memory in processor based embedded systems
  - high capacity, low cost
- Many variations of DRAMs proposed
  - need to keep pace with processor speeds
  - FPM DRAM: fast page mode DRAM
  - EDO DRAM: extended data out DRAM
  - SDRAM/ESDRAM: synchronous and enhanced synchronous DRAM
  - RDRAM: Rambus DRAM
- Address bus multiplexed between row and column components
- Row and column addresses are latched in, sequentially, by strobing *ras* (row address strobe) and *cas* (column address strobe) signals, respectively
- Refresh circuitry can be external or internal to DRAM device
  - strobes consecutive memory address periodically causing memory content to be refreshed
  - Refresh circuitry disabled during read or write operation

DRAM Circuit Diagram

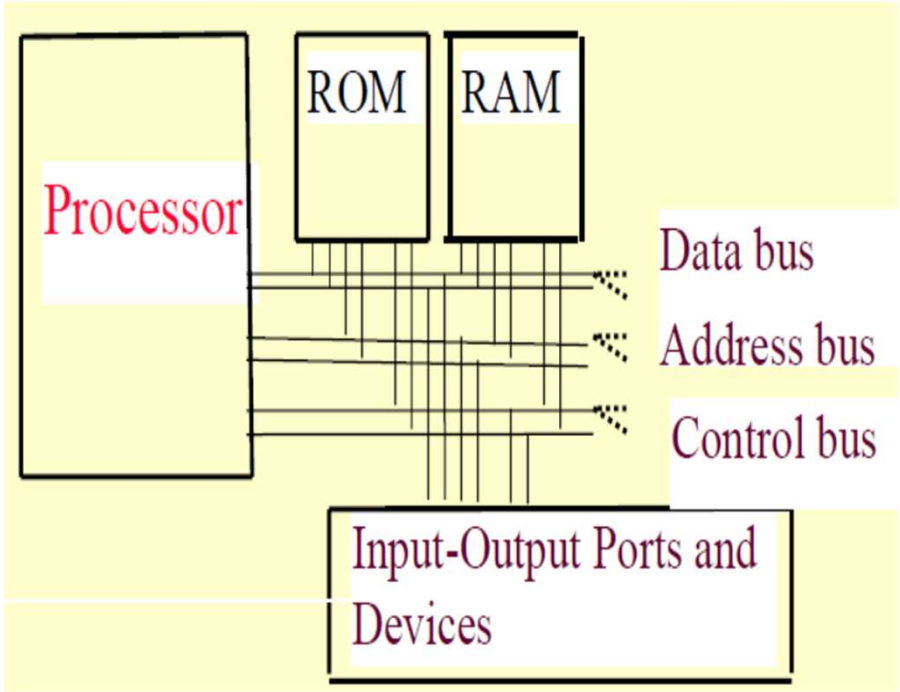


Interfacing Memory and Other components

- Interfacing of processor, memory and IO devices using memory system bus

Address Bus

- Processor issues the address of the instruction byte or word to memory system through the address bus.
- Processor execution unit, when required, issues the address of data (byte or word) to be read or written using the memory system through address bus.
- The address bus of 32-bits used to fetch the instruction or data from an address specified by 32-bit number.



Address Bus, data Bus, Control Bus

- **Data Bus**
- **Instruction fetch–** Processor issues the address of the instruction, it gets back the instruction through the data bus.
- **Data Read–** When it issues the address of The data, it loads the data through data bus.
- **Data Write–** When it issues the address of the data, it stores the data in the memory through the data bus. A data bus of 32-bits fetches, loads, or stores the instruction or data of 32-bits.

Address Bus, data Bus, Control Bus

Control Bus

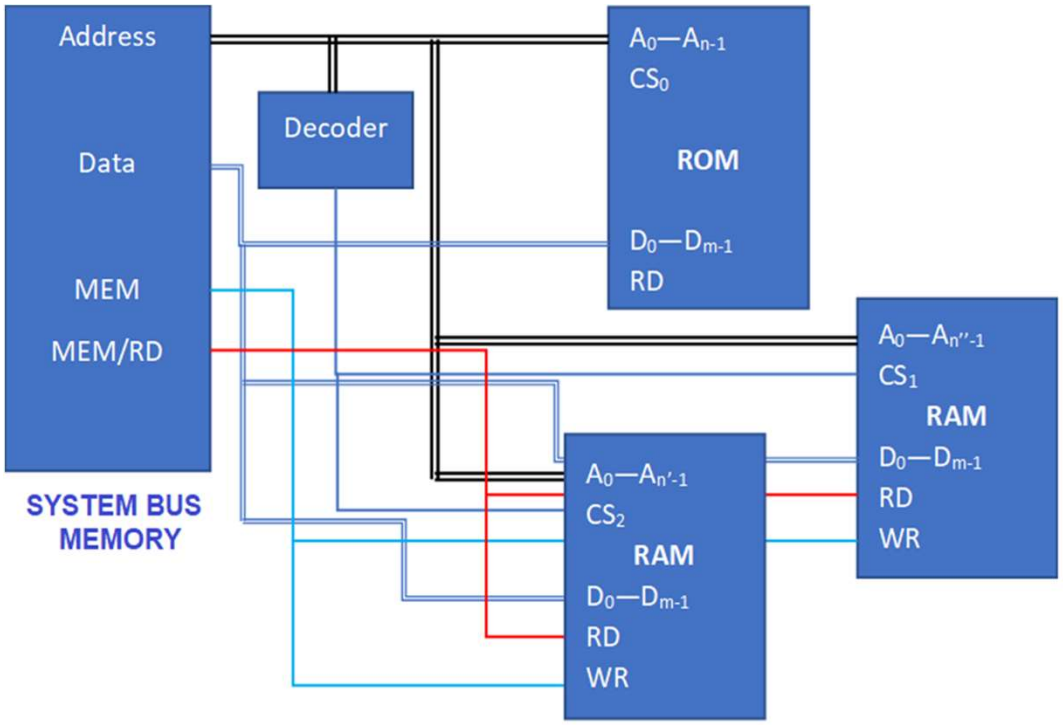
- Issues signals to control the timing of various actions during interconnection.
- Signals synchronize all the subsystems.
- address latch enable (ALE)[Address Strobe (AS) or address valid, (ADV)],
- memory ‘read’ (RD) or ‘write’ (WR) or IO ‘read’ (IORD) or ‘write,’(IOWR) or ‘data valid’(DAV)
- Other control signals as per the processor design.

Interrupts and DMA Control Signals

- Interrupt acknowledge (INTA) [on a request for drawing the processor attention to an event]
- INT (Interrupt) from external device interrupt to the system
- Hold acknowledge (HLDA) [on an external hold request for permitting use of the system buses]
- HOLD when external device sends a hold request for direct memory access (DMA).

Memory interfacing

- Memory interfacing can be done I two ways using
  - Harvard Architecture
  - Von Neuman Architecture
- The general interfacing of memory is shown in figure.



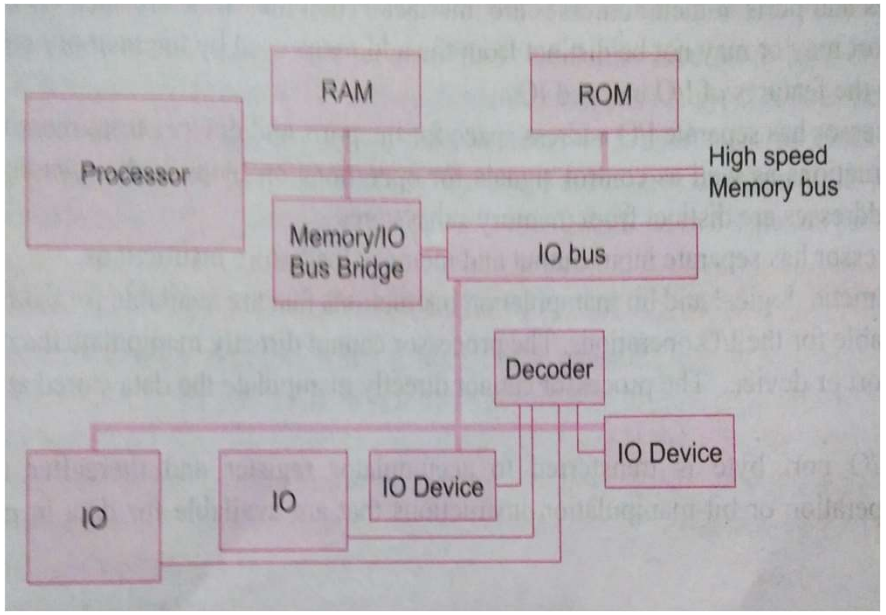
I/O Devices and Component Interfacing

Method 1: I/O devices or the components can be interfaced using ports which uses a decoder to connect to bus and control unit. The interfacing is done as per available control signal and timing diagram of system bus.

Method 2: other interfacing method is the use of I/O controller (called as bridge or switch) and it connects the processor and system memory on the system memory bus with I/O bus.

Memory mapped I/O operation

- Memory mapped I/o means the processor does not take I/O as separate addressed device but considers I/O operations like memory operations.
- Same address space is used for both memory and I/O. I
- It uses the same instruction and control signal to access memory and I/O.
- I/O devices addresses are distinct from memory subsystem address .
- The figure shows the memory mapped I/operations.
- The arithmetic, logic and bit manipulations are used for both I/O and Memory operations.



Memory mapped I/O operation

- For I/O mapped I/o, processor has separate instructions for I/O operations i.e. I/O devices has separate addresses and memory has separate addresses.
- It uses different control signals for I/O and memory operations.
- Thus the I/O read write cycles are different from memory read write cycles.

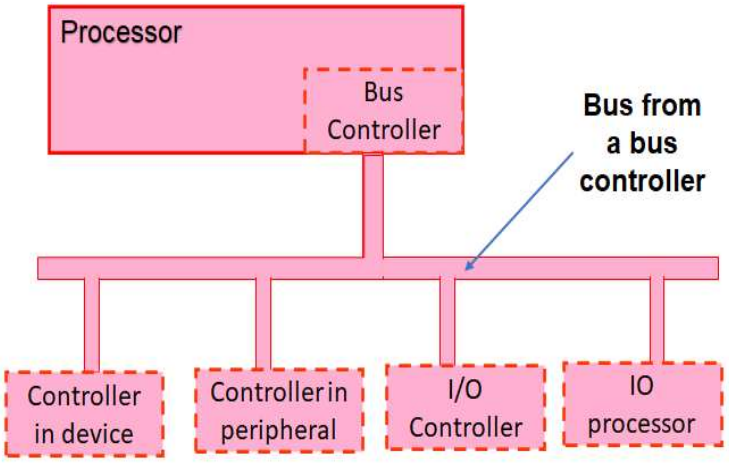
Bus Arbitrations

- When a number of devices or processors share a common bus, bus arbitration required.
- The bus arbitrator allows the bus to the device which wants to transmit while other devices are not granted bus.
- A controller or processor wait until the bus goes ideal.
- Figure shows how a common bus is shared by different devices.

- The bus arbiter can work in three ways to grant the bus to the devices which are
  - Daisy chain method
  - Independent bus request
  - Polling-bus request.

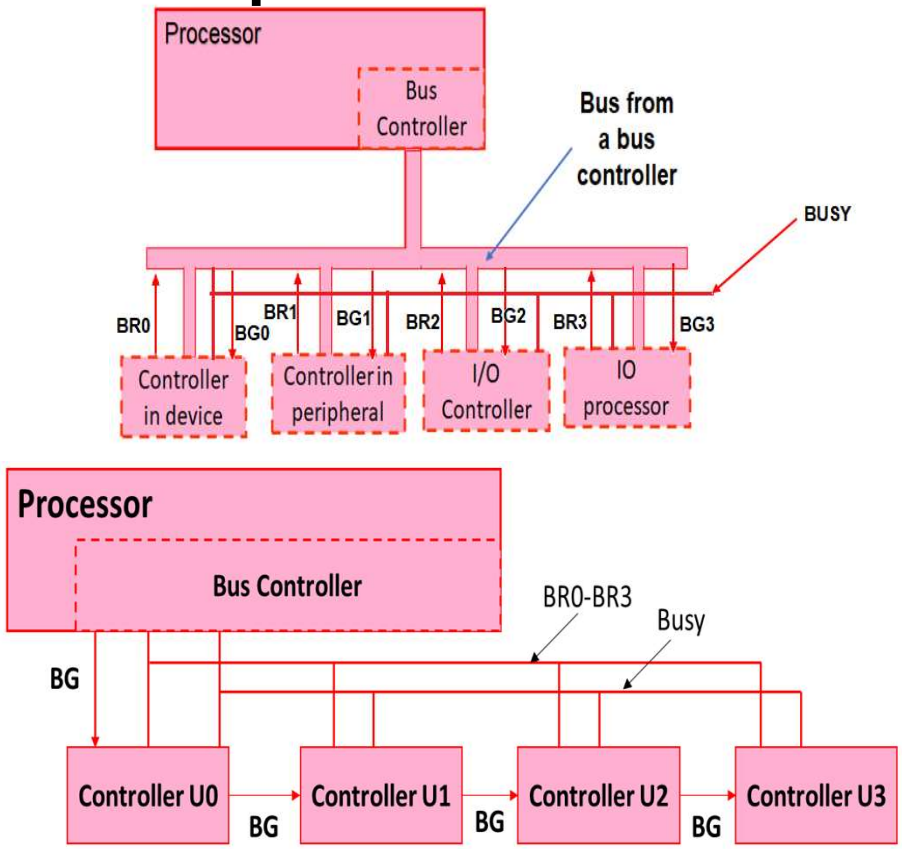
Bus Arbitration – daisy chain method

- Figure shows the daisy chain method of bus-arbitration.
- In this method the bus control passes from one bus master to next bus master and so on.
- Signal in the bus arbitration has a bus grant(BG) which functions like a token first passes to U0 device . If the U0 device does not want the bus, it then passes to U1 and so on.
- The device which wants the bus, raises a bus request (BR) and a busy signal is generated (BUSY) from the device which now acts as a bus master.
- When the bus master does not need the bus, it deactivates BR signal and controller then deactivates the BUSY signal.
- The controller then generates a BG signal which passes to other devices depend on the priority.



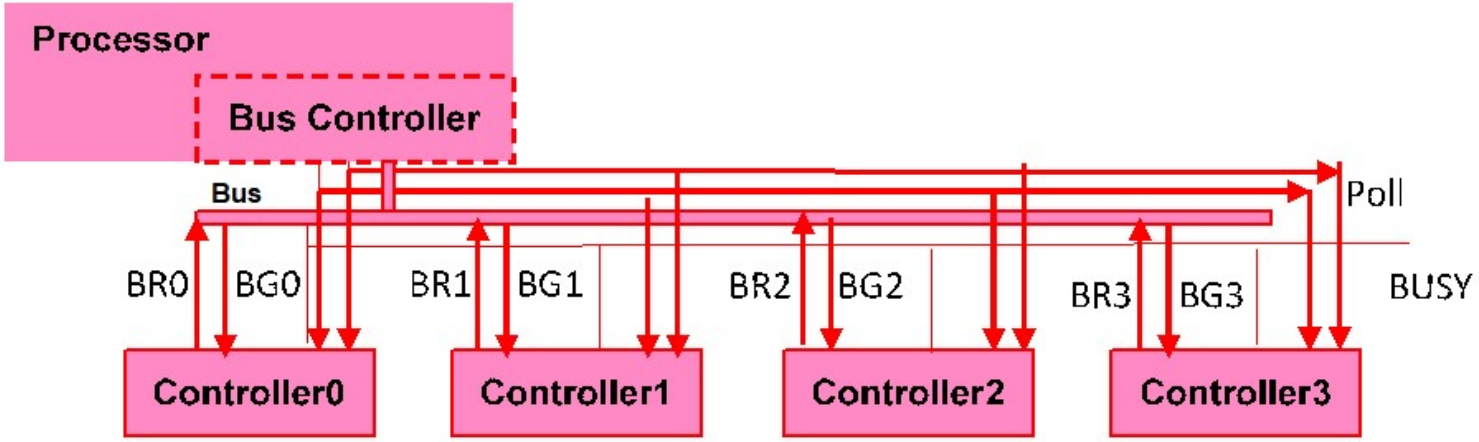
Bus Arbitration – Independent Bus Request method

- Figure shows the bus arbitration using independent bus request.
- The independent bus request and bus grant is the method used for centralised bus arbitration process.
- In this each device has separate Bus request signals BR0, BR1, BR2, BR3--------BRn also they have separate BUS grant signals BG0, BG1, BG2 - - - - - BG.
- When an i^{th} controller sends bus request, and upon receiving Bus grant, it uses the busy signal to indicate that bus is occupied by that controller.
- This BUSY signal is activated by the central controller and no other system raises BR by seeing the active Busy signal.
- In this method priority level of each controller on the devices can be changed dynamically.



Bus Arbitration- Polling Bus Request Method

- Figure shows the bus arbitration using polling bus request method.
- In this poll count is sent to main controller and is incremented. When a poll count= i^{th} poll count, it stops incrementing the poll count and the BR signal is received by the central controller activates the BG signal such that the ith device is now bus master. Simultaneously a BUSY signal is activated.
- When the BR signal is deactivated by the ith device, the busy signal is also removed and polling count starts by incrementing the poll counter value.
- It has the advantage that controller next to the current bus master gets the highest priority.



Embedded versus External Memory Devices

Features	Embedded Memory	External Memory
Type	1. Embedded RAM, ROM or flash memory in microcontrollers and VLSI Circuits 2. Embedded SRAM	1. External SRAM, DRAMS, RAMs, ROM and Flash ROM. 2. Large memory size
Features	1. Same masking steps as in other logic circuits in SRAM 2. DRAMs need a special complex design 3. Multiport features 4. Synchronous or asynchronous mode of operation	1. Needs external bus for interconnection 2. Drams needs refresher circuits 3. Multiport features 4. Synchronous or asynchronous mode of operation
Memory access time	Negligible small	Higher compared to embedded memory