Computer Function Interconnection Memory

120CT Advanced Computer Architecture

Dr Dianabasi Nkantah
ab0480@coventry.ac.uk

Today.....

- Components of a computer
- Computer Function
  - Instruction Fetch and Execute
  - Interrupts
  - I/O Function
- Interconnection Structures
  - Bus Interconnection
    - Bus Structure
    - Multiple-Bus Hierarchies
  - Point-to-Point Interconnect
  - Peripheral Component Interconnect (PCI)
  - PCI Express (PCIe)
- Computer Memory
  - RAM
  - ROM
  - Memory Hierarchy
  - Cache

Computer Components

- Contemporary computer designs are based on concepts developed by John von Neumann at the Institute for Advanced Studies, Princeton
- Referred to as the *von Neumann architecture* and is based on three key concepts:
  1. Data and instructions are stored in a single read-write memory
  2. The contents of this memory are addressable by location, without regard to the type of data contained there
  3. Execution occurs in a sequential fashion (unless explicitly modified) from one instruction to the next

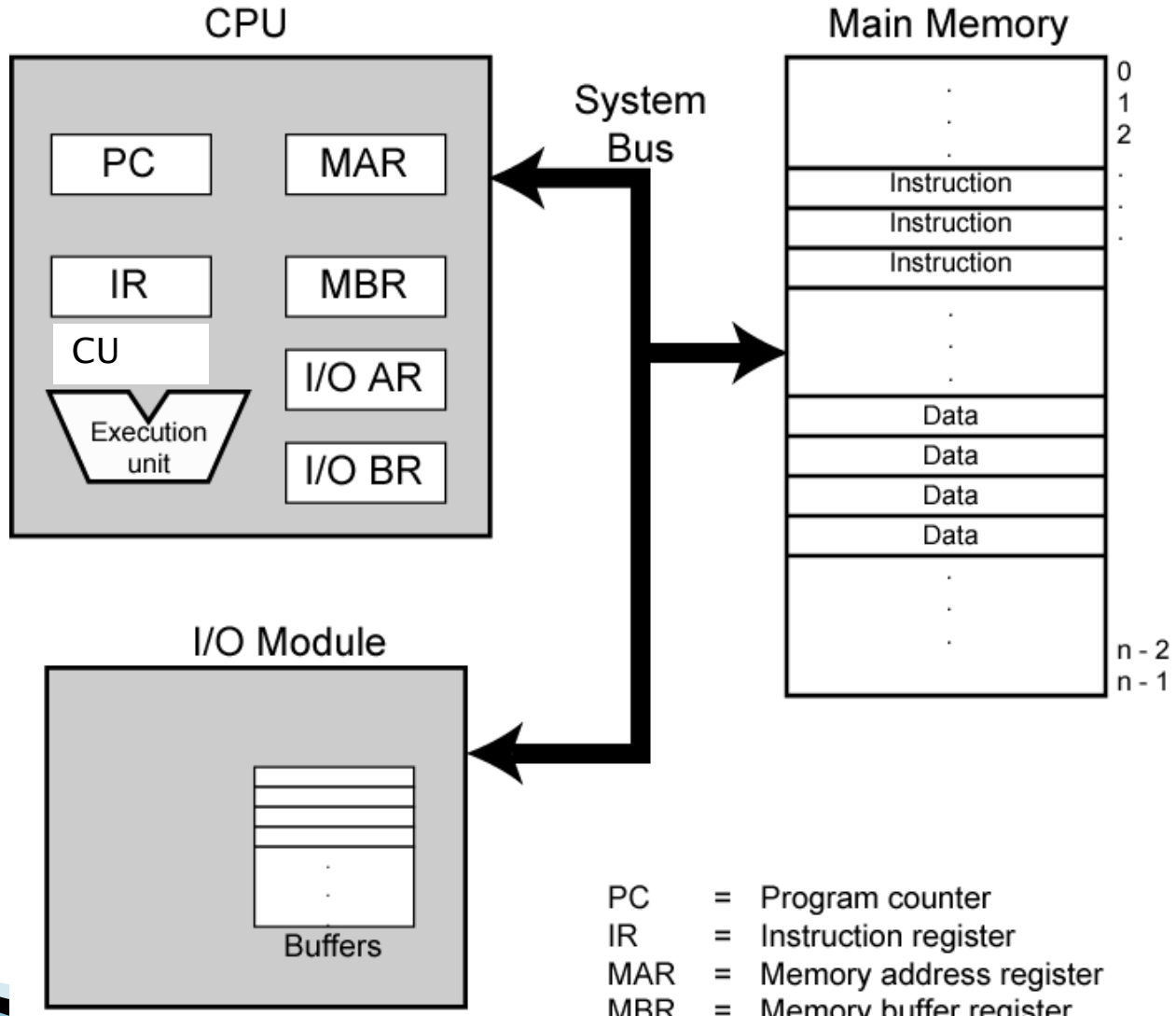
- *Hardwired program*
  - The result of the process of connecting the various components in the desired configuration

- An alternative to the *von Neumann architecture* is the *Harvard architecture*

Components

- The Control Unit and the Arithmetic and Logic Unit constitute the **Central Processing Unit**
- Data and instructions need to get into the system and results need to get out
  - **Input/output**
- Temporary storage of code and results is needed
  - **Main memory**

Computer Components: Top Level View

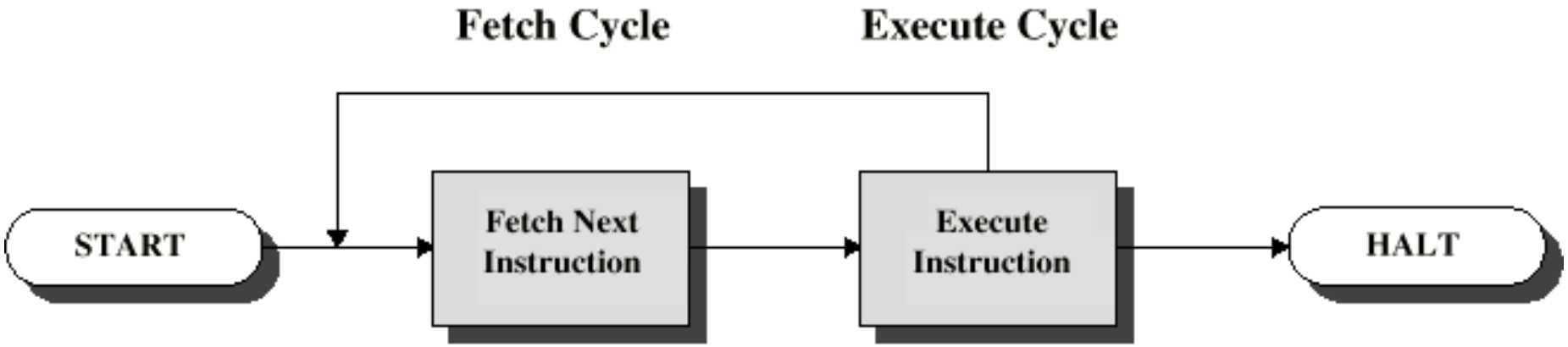


PC = Program counter
IR = Instruction register
MAR = Memory address register
MBR = Memory buffer register
I/O AR = Input/output address register
I/O BR = Input/output buffer register

Stallings, W., 2013

Basic Instruction Cycle

- Two steps:
  - Fetch
  - Execute



Fetch Cycle

- Program Counter (PC) holds address of next instruction to fetch
- Processor fetches instruction from memory location pointed to by PC
- Increment PC
  - Unless told otherwise
- Instruction loaded into Instruction Register (IR)
- Processor interprets instruction and performs required actions

Execute Cycle

- Processor-memory Transfer
  - Data transfer between CPU and main memory
- Processor-I/O Transfer
  - Data transfer between CPU and I/O module
- Data processing
  - Some arithmetic or logical operation on data
- Control
  - Alteration of sequence of operations
    - e.g. jump
- Combination of above

Interrupts

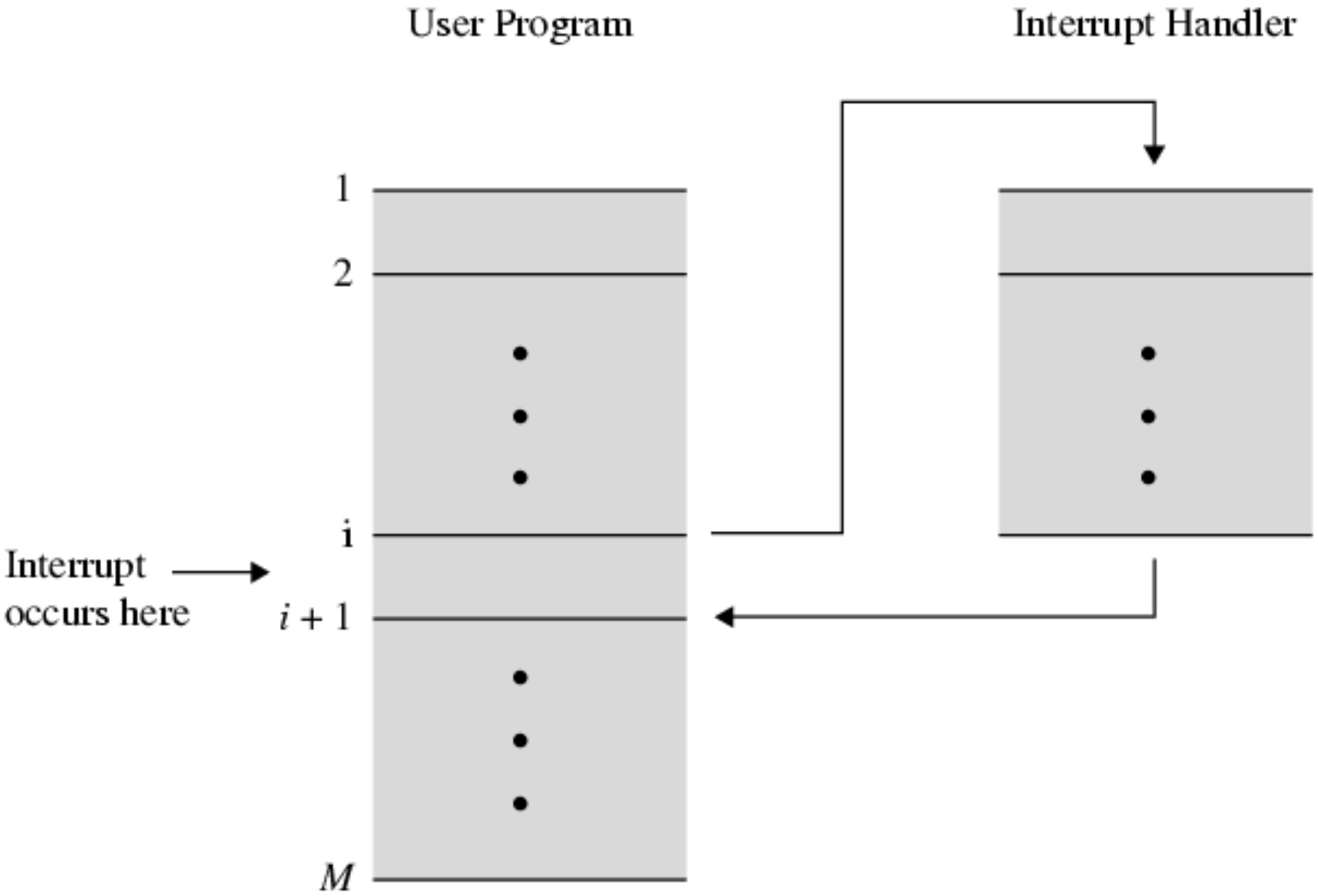
- Mechanism by which other modules (e.g. I/O) may interrupt normal sequence of processing

- Types of Interrupts:
  - **Program**
    - e.g. overflow, division by zero
  - **Timer**
    - Generated by internal processor timer
    - Used in pre-emptive multi-tasking
  - **I/O**
    - from I/O controller
  - **Hardware failure**
    - e.g. memory parity error

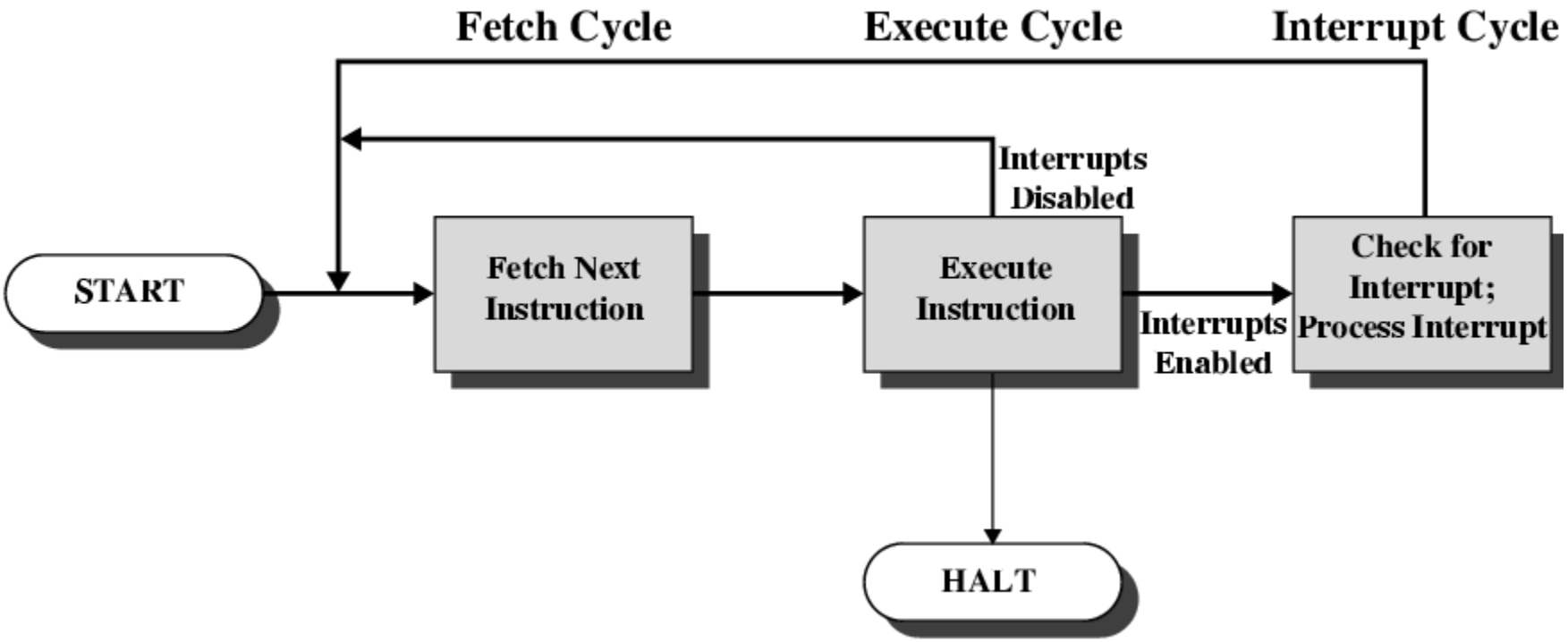
Interrupt Cycle

- Added to instruction cycle
- Processor checks for interrupt
  - Indicated by an interrupt signal
- If no interrupt, fetch next instruction
- If interrupt pending:
  - Suspend execution of current program
  - Save context
  - Set PC to start address of interrupt handler routine
  - Process interrupt
  - Restore context and continue interrupted program

Transfer of Control via Interrupts



Instruction Cycle with Interrupts

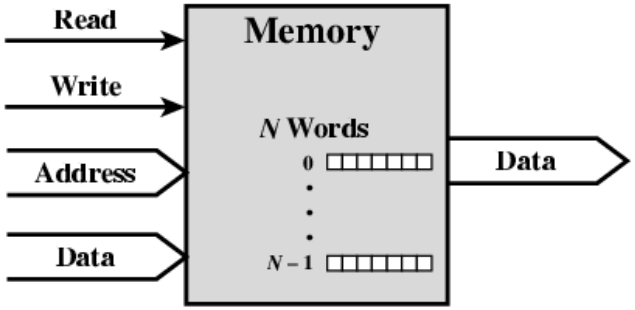


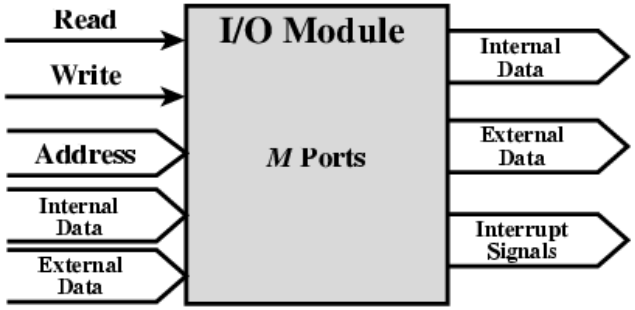
Multiple Interrupts

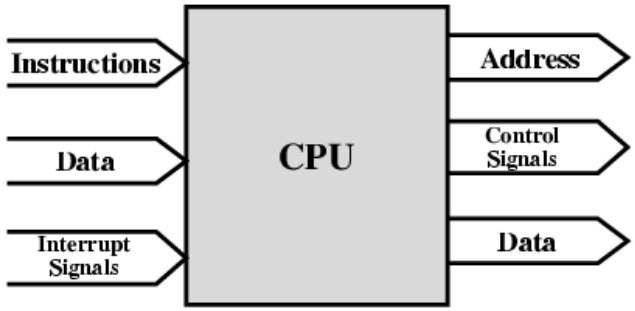
- **Disable interrupts**
  - Processor will ignore further interrupts whilst processing one interrupt
  - Interrupts remain pending and are checked after first interrupt has been processed
  - Interrupts handled in sequence as they occur

- **Define priorities**
  - Low priority interrupts can be interrupted by higher priority interrupts
  - When higher priority interrupt has been processed, processor returns to previous interrupt

Computer Interconnection







Memory Connection

- Receives and sends data
- Receives addresses (of locations)
- Receives control signals
  - Read
  - Write
  - Timing

Input/Output Connection(1)

- Similar to memory from computer's viewpoint
- Output
  - Receive data from computer
  - Send data to peripheral
- Input
  - Receive data from peripheral
  - Send data to computer

Input/Output Connection(2)

- Receive control signals from computer
- Send control signals to peripherals
  - e.g. spin disk
- Receive addresses from computer
  - e.g. port number to identify peripheral
- Send interrupt signals (control)

CPU Connection

- Reads instruction and data
- Writes out data (after processing)
- Sends control signals to other units
- Receives (& acts on) interrupts

Buses

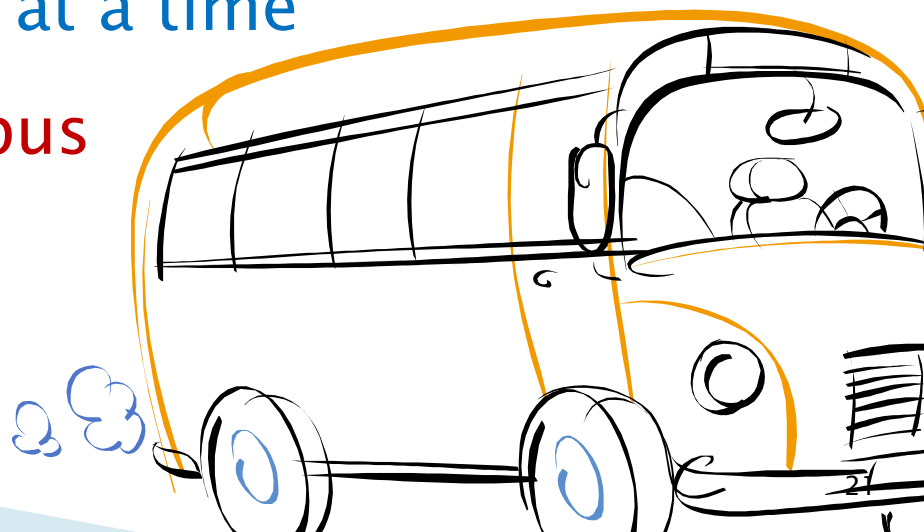
- There are a number of possible interconnection systems
- Single and multiple BUS structures have been the most common for decades
  - e.g. Control/Address/Data bus (PC)
  - e.g. Unibus (DEC-PDP)

What is a Bus?

- A communication pathway connecting two or more devices
- Usually broadcast
- Often grouped
  - A number of channels in one bus
    - e.g. 32 bit data bus is 32 separate single bit channels

Data Bus

- Data lines that provide a path for moving data among system modules
- May consist of 32, 64, 128, or more separate lines
- The number of lines is referred to as the *width* of the data bus
- The number of lines determines how many bits can be transferred at a time
- The width of the data bus is a key factor in determining overall system performance



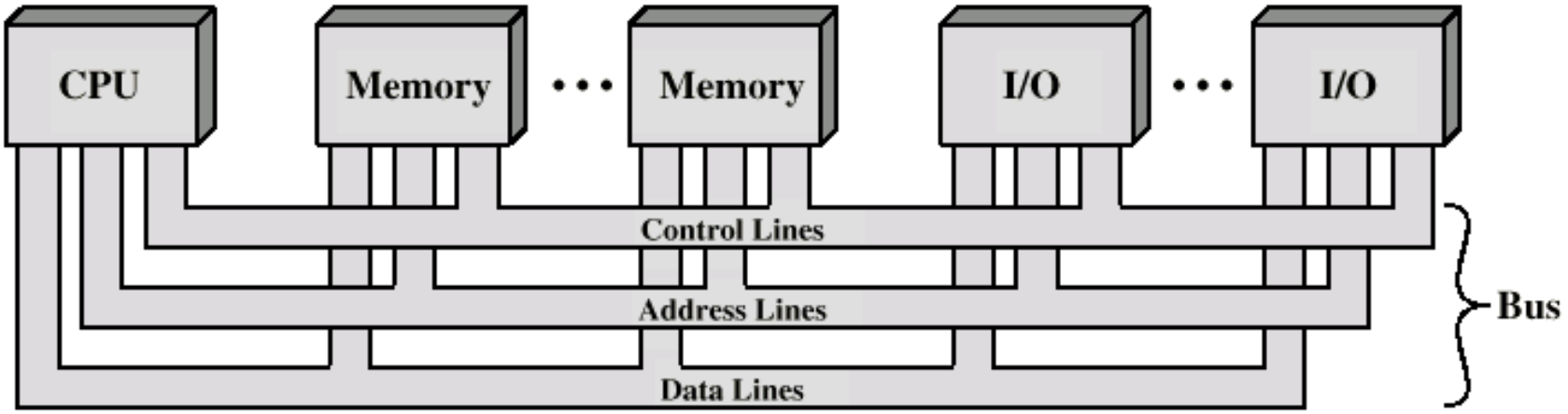
Address bus

- Identify the source or destination of data
  - e.g. CPU needs to read an instruction (data) from a given location in memory
- Bus width determines maximum memory capacity of system
  - e.g. 8080 has 16 bit address bus giving 64k address space

Control Bus

- Control and timing information
  - Memory read/write signal
  - Interrupt request
  - Clock signals

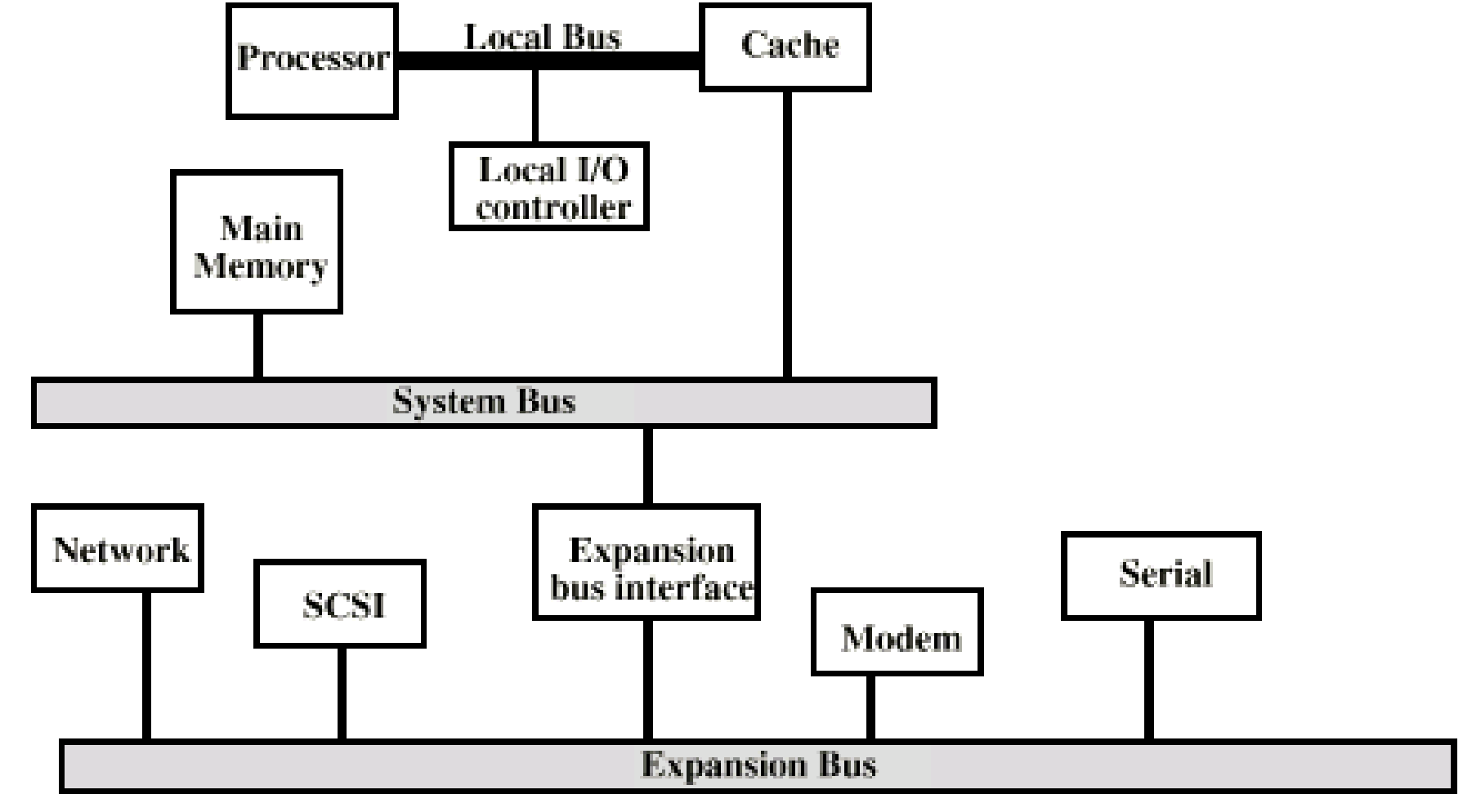
Bus Interconnection Scheme



Single Bus Problems

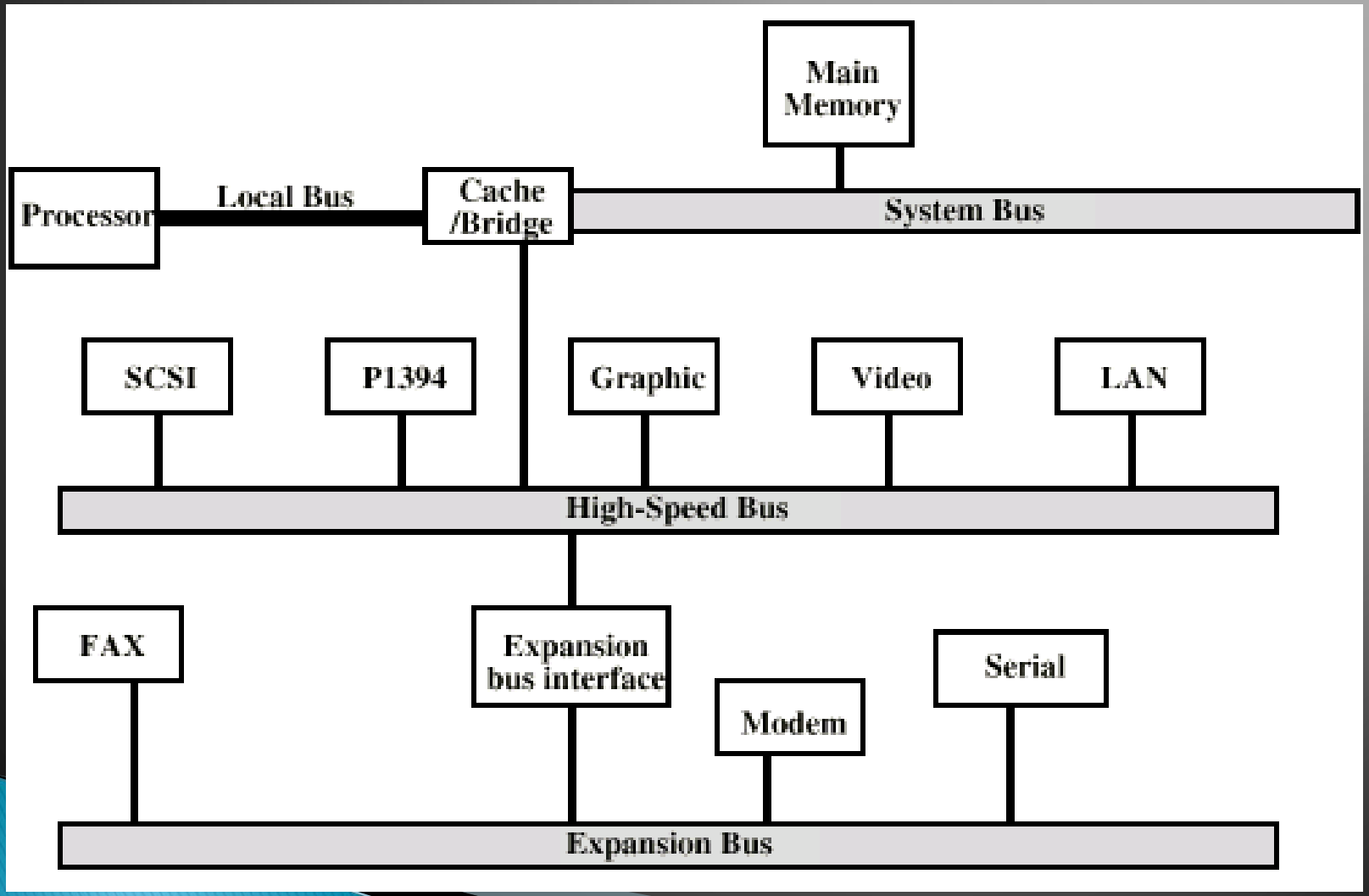
- Lots of devices on one bus leads to:
  - Propagation delays
    - Long data paths mean that co-ordination of bus use can adversely affect performance
    - If aggregate data transfer approaches bus capacity
- Most systems use multiple buses to overcome these problems

Traditional Bus Architecture (with cache)



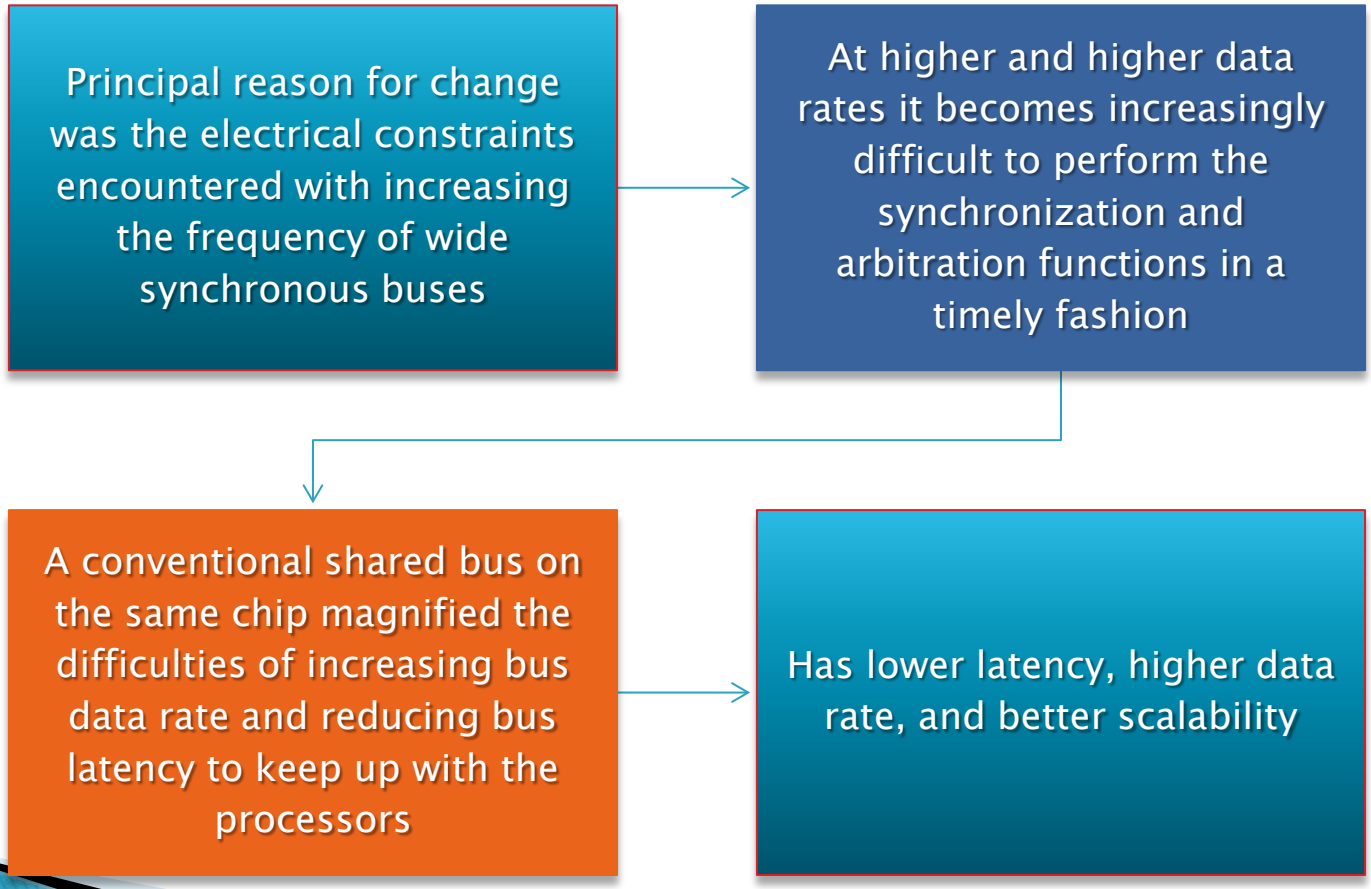
Stallings, W., 2013

High Performance Bus



Stallings, W., 2013

Point-to-Point Interconnect



PCI and PCIe

- **Peripheral Component Interconnect (PCI)**
  - A popular high bandwidth, processor independent bus that can function as a mezzanine or peripheral bus
  - Delivers better system performance for high speed I/O subsystems
- **PCI Express (PCIe)**
  - Point-to-point interconnect scheme intended to replace bus-based schemes such as PCI
  - Key requirement is high capacity to support the needs of higher data rate I/O devices, such as Gigabit Ethernet
  - Another requirement deals with the need to support time dependent data streams

Memory: Key Characteristics of Computer Memory Systems

Location Internal (e.g. processor registers, cache, main memory) External (e.g. optical disks, magnetic disks, tapes) **Capacity** Number of words Number of bytes **Unit of Transfer** Word Block **Access Method** Sequential Direct Random Associative	**Performance** Access time Cycle time Transfer rate **Physical Type** Semiconductor Magnetic Optical Magneto-optical **Physical Characteristics** Volatile/nonvolatile Erasable/nonerasable **Organization** Memory modules

Key Characteristics of Computer Memory Systems

Capacity and Performance:

The two most important characteristics of memory

Three performance parameters are used:

Access time (latency)

- For random-access memory it is the time it takes to perform a read or write operation
- For non-random-access memory it is the time it takes to position the read-write mechanism at the desired location

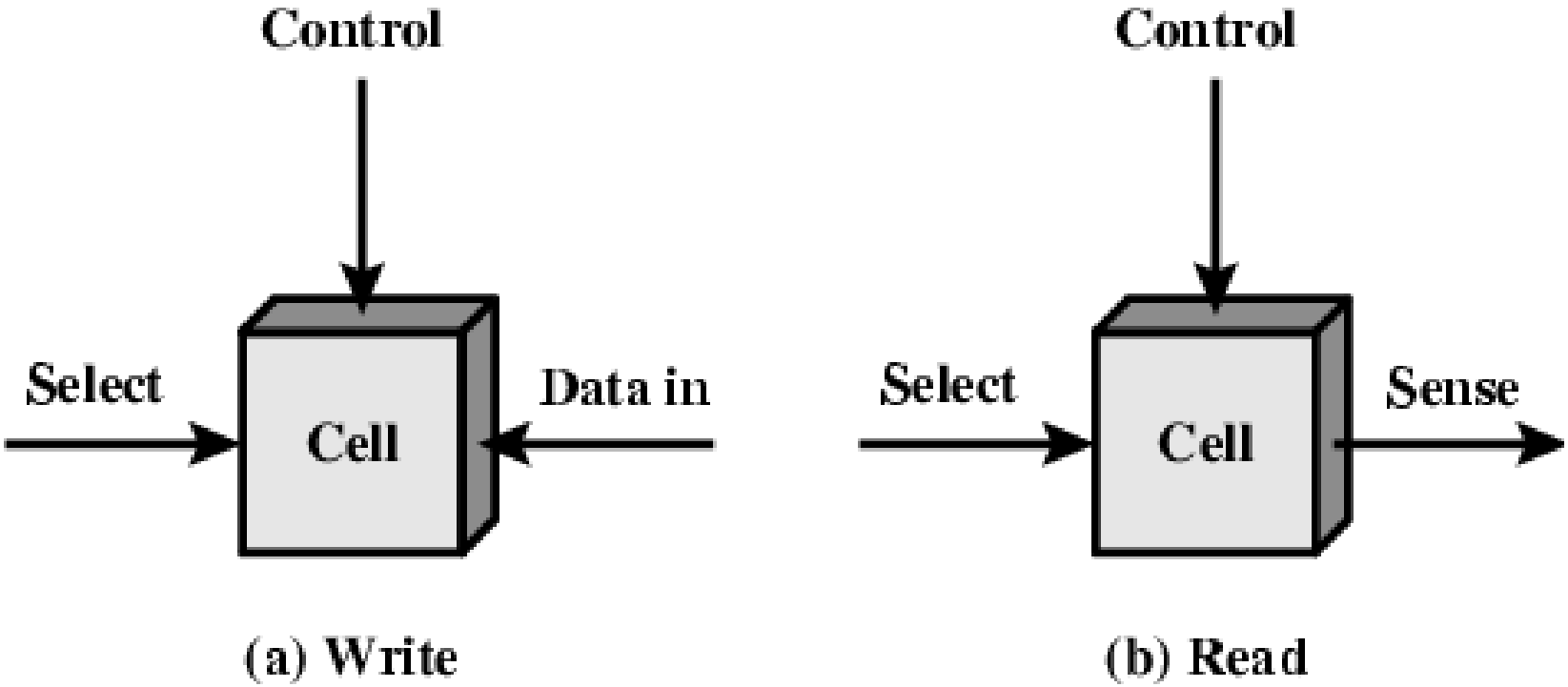
Memory cycle time

- Access time plus any additional time required before second access can commence
- Additional time may be required for transients to die out on signal lines or to regenerate data if they are read destructively
- Concerned with the system bus, not the processor

Transfer rate

- The rate at which data can be transferred into or out of a memory unit
- For random-access memory it is equal to 1/(cycle time)

Memory Cell Operation



Semiconductor Memory Types

<table>
<tr><th>Memory Type</th><th>Category</th><th>Erasure</th><th>Write Mechanism</th><th>Volatility</th></tr>
<tr><td>Random-access memory (RAM)</td><td>Read-write memory</td><td>Electrically, byte-level</td><td>Electrically</td><td>Volatile</td></tr>
<tr><td>Read-only memory (ROM)</td><td rowspan="2">Read-only memory</td><td rowspan="2">Not possible</td><td>Masks</td><td rowspan="5">Nonvolatile</td></tr>
<tr><td>Programmable ROM (PROM)</td><td rowspan="4">Electrically</td></tr>
<tr><td>Erasable PROM (EPROM)</td><td rowspan="3">Read-mostly memory</td><td>UV light, chip-level</td></tr>
<tr><td>Electrically Erasable PROM (EEPROM)</td><td>Electrically, byte-level</td></tr>
<tr><td>Flash memory</td><td>Electrically, block-level</td></tr>
</table>

Semiconductor Memory

- **RAM**
  - Misnamed as all semiconductor memory is random access
  - Read/Write
  - Volatile
  - Temporary storage
  - Static or dynamic

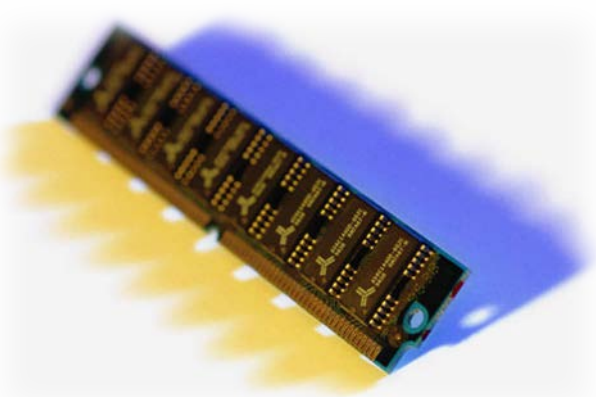
Dynamic RAM (DRAM)

- RAM technology is divided into two technologies:
  - Dynamic RAM (DRAM)
  - Static RAM (SRAM)

- **DRAM**
  - Made with cells that store data as charge on capacitors
  - Presence or absence of charge in a capacitor is interpreted as a binary 1 or 0
    - This depends on the threshold voltage of the transistor connected to the capacitor
  - Requires periodic charge refreshing to maintain data storage
  - The term *dynamic* refers to tendency of the stored charge to leak away, even with power continuously applied

Static RAM (SRAM)

- Digital device that uses the same logic elements used in the processor
- Binary values are stored using traditional flip-flop logic gate configurations
- Will hold its data as long as power is supplied to it



Static RAM

- Bits stored as on/off switches
- No charges to leak
- No refreshing needed when powered
- More complex construction
- Larger per bit
- More expensive
- Does not need refresh circuits
- Faster
- Used for Cache
- Digital
  - Uses flip-flops

SRAM versus DRAM

- **Both volatile**
  - Power must be continuously supplied to the memory to preserve the bit values

- **Dynamic cell**

DRAM

  - Simpler to build, smaller
  - More dense (smaller cells = more cells per unit area)
  - Less expensive
  - Requires the supporting refresh circuitry
  - Tend to be favored for large memory requirements
  - Used for main memory

- **Static**

SRAM

  - Faster
  - Used for cache memory (both on and off chip)

Read Only Memory (ROM)

- Contains a permanent pattern of data that cannot be changed or added to
- No power source is required to maintain the bit values in memory
- Data or program is permanently in main memory and never needs to be loaded from a secondary storage device
- Data is actually wired into the chip as part of the fabrication process
  - Disadvantages of this:
    - No room for error, if one bit is wrong the whole batch of ROMs must be thrown out
    - Data insertion step includes a relatively large fixed cost

Read Only Memory (ROM) – Potential Applications

- Permanent storage
  - Nonvolatile
- Microprogramming
- Library subroutines
- Systems programs (BIOS)
- Function tables

Types of ROM

- Written during manufacture
  - Very expensive for small runs
- **Programmable (once)**
  - PROM
    - Needs special equipment to program
- **Read "mostly"**
  - Erasable Programmable (EPROM)
    - Erased by UV
  - Electrically Erasable (EEPROM)
    - Takes much longer to write than read
  - Flash memory
    - Erase whole memory electrically

Programmable ROM (PROM)

- Less expensive alternative
- Nonvolatile and may be written into only once
- Writing process is performed electrically and may be performed by supplier or customer at a time later than the original chip fabrication
- Special equipment is required for the writing process
- Provides flexibility and convenience
- Attractive for high volume production runs

Read-Mostly Memory

EPROM

- Erasable programmable read-only memory
- Erasure process can be performed repeatedly
- More expensive than PROM but it has the advantage of the multiple update capability

EEPROM

- Electrically erasable programmable read-only memory
- Can be written into at any time without erasing prior contents
- Combines the advantage of non-volatility with the flexibility of being updatable in place
- More expensive than EPROM

Flash Memory

- Intermediate between EPROM and EEPROM in both cost and functionality
- Uses an electrical erasing technology, does not provide byte-level erasure
- Microchip is organised so that a section of memory cells are erased in a single action or "flash"

Synchronous DRAM (SDRAM)

One of the most widely used forms of DRAM

Exchanges data with the processor synchronised to an external clock signal and running at the full speed of the processor/memory bus without imposing wait states

With synchronous access the DRAM moves data in and out under control of the system clock

- The processor or other master issues the instruction and address information which is latched by the DRAM
- The DRAM then responds after a set number of clock cycles
- Meanwhile the master can safely do other tasks while the SDRAM is processing

Double Data Rate SDRAM (DDR SDRAM)

- SDRAM can only send data once per bus clock cycle
- Double-data-rate SDRAM can send data twice per clock cycle, once on the rising edge of the clock pulse and once on the falling edge
- Developed by the JEDEC Solid State Technology Association (Electronic Industries Alliance's semiconductor-engineering-standardisation body)

Memory Hierarchy

- Design constraints on a computer's memory can be summed up by three questions:
  - How much, how fast, how expensive
- There is a trade-off among capacity, access time, and cost
  - Faster access time, greater cost per bit
  - Greater capacity, smaller cost per bit
  - Greater capacity, slower access time
- The way out of the memory dilemma is not to rely on a single memory component or technology, but to employ a memory hierarchy

Memory Hierarchy

- Registers
  - In CPU
- Internal or Main memory
  - May include one or more levels of cache
  - “RAM”
- External memory
  - Backing store

Memory Hierarchy

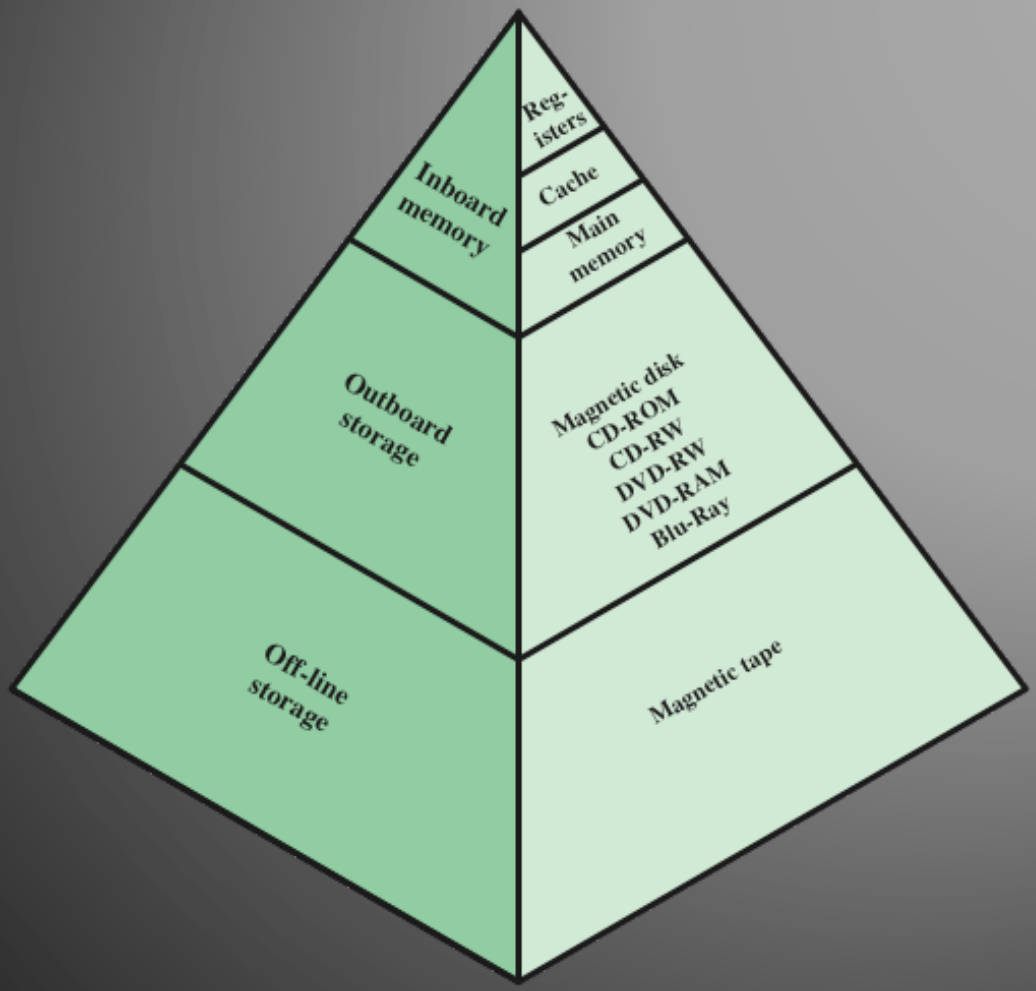


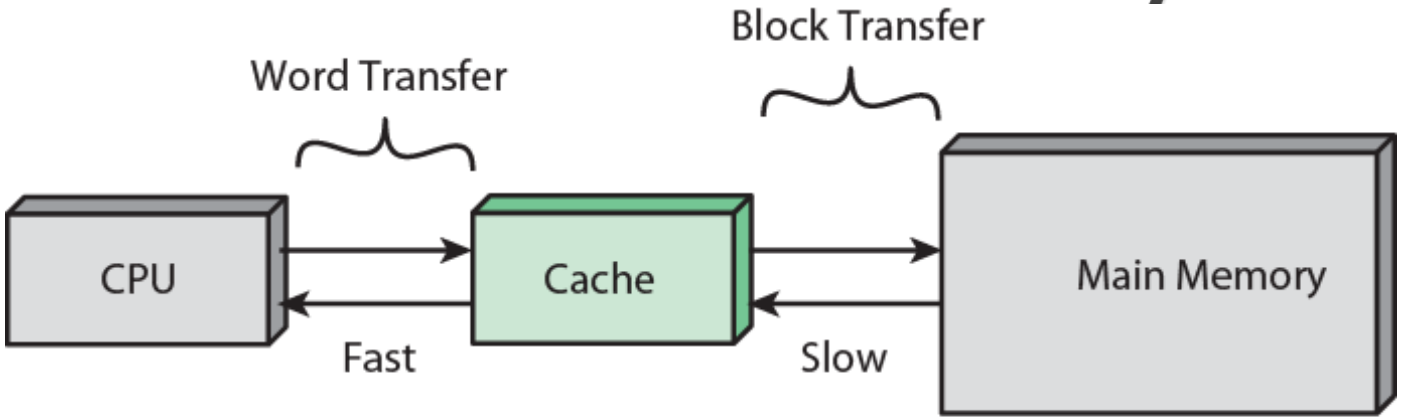
Figure 4.1 The Memory Hierarchy

Stallings, W., 2013

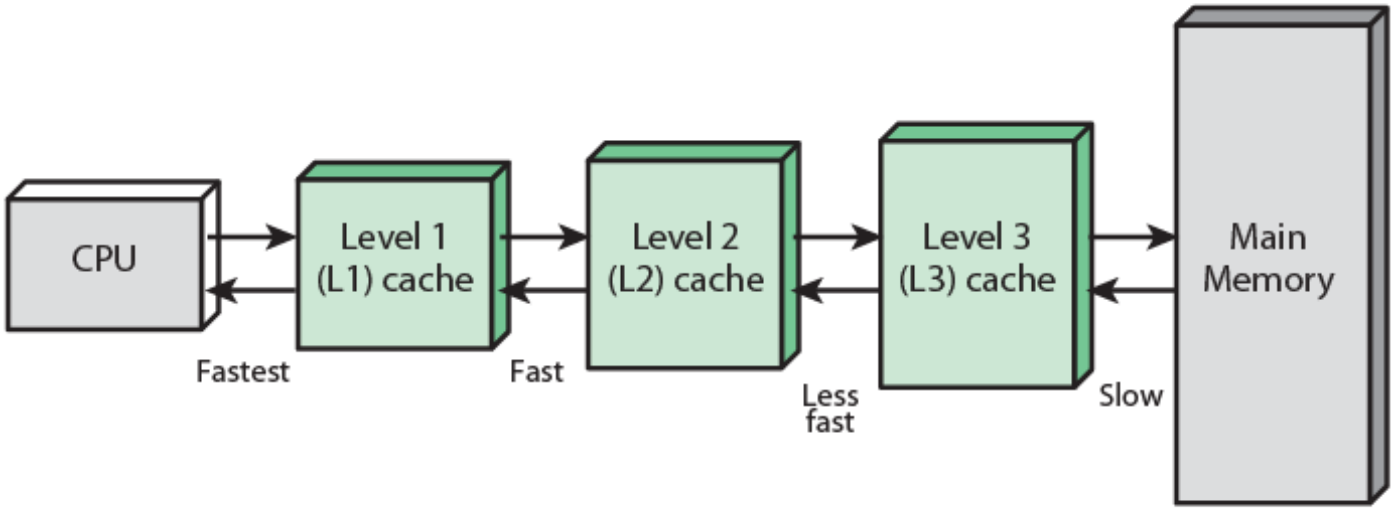
Cache

- Small amount of fast memory
- Sits between normal main memory and CPU
- May be located on CPU chip or module

Cache and Main Memory

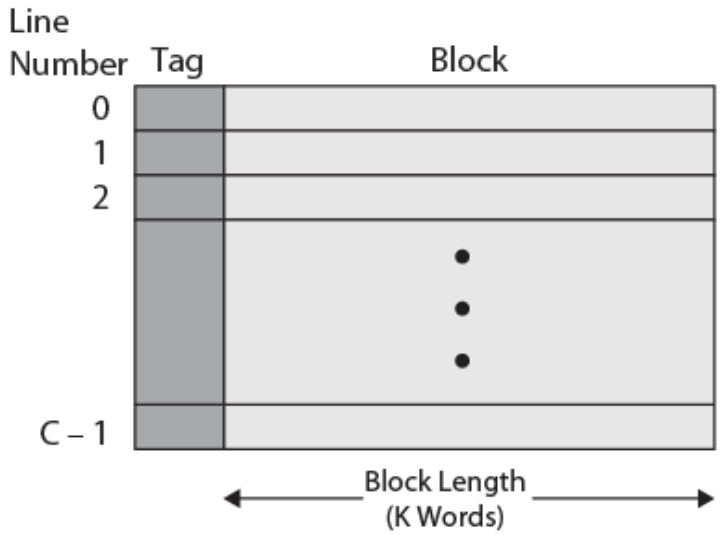


(a) Single cache

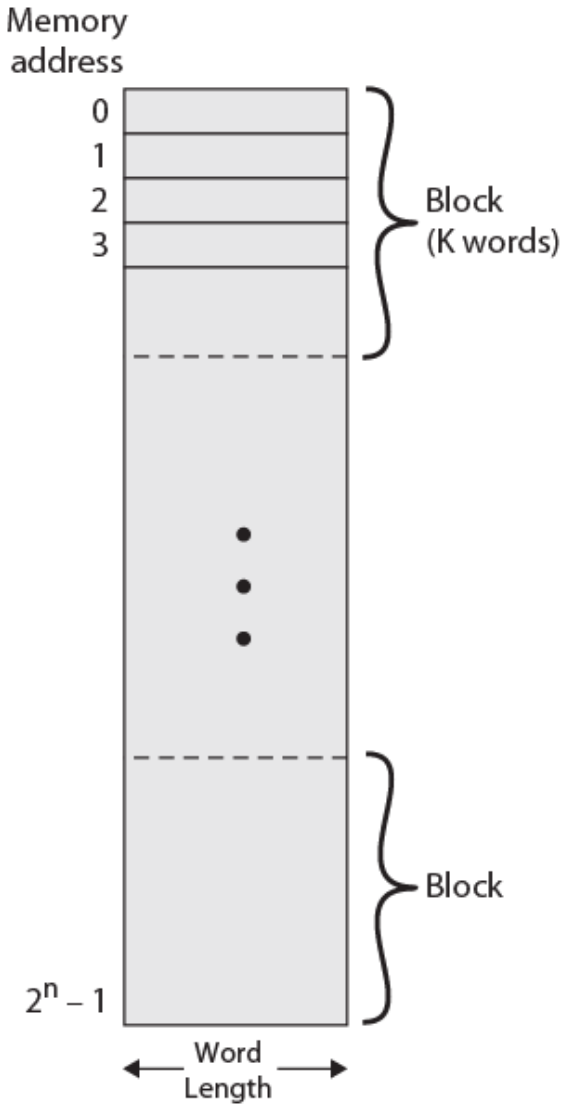


(b) Three-level cache organization

Cache/Main Memory Structure



(a) Cache



(b) Main memory

Cache operation – overview

- CPU requests contents of memory location
- Check cache for this data
- If present, get from cache (fast)
- If not present, read required block from main memory to cache
- Then deliver from cache to CPU
- Cache includes tags to identify which block of main memory is in each cache slot

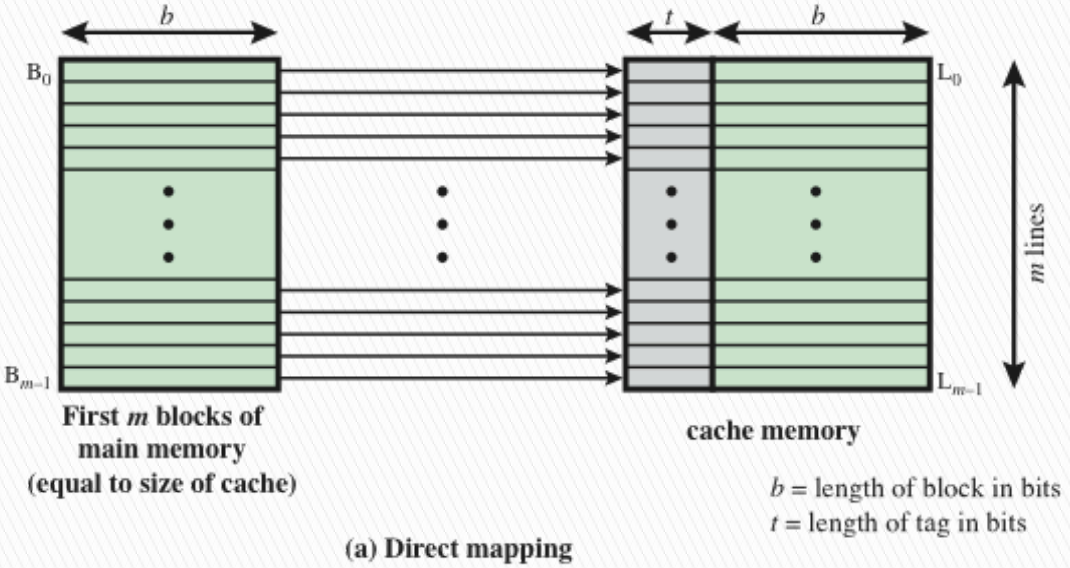
Mapping Function

- Direct Mapping
- Associative Mapping
- Set Associative mapping

Direct Mapping

- Each block of main memory maps to only one cache line
  - i.e. if a block is in cache, it must be in one specific place

Direct Mapping



Associative Mapping

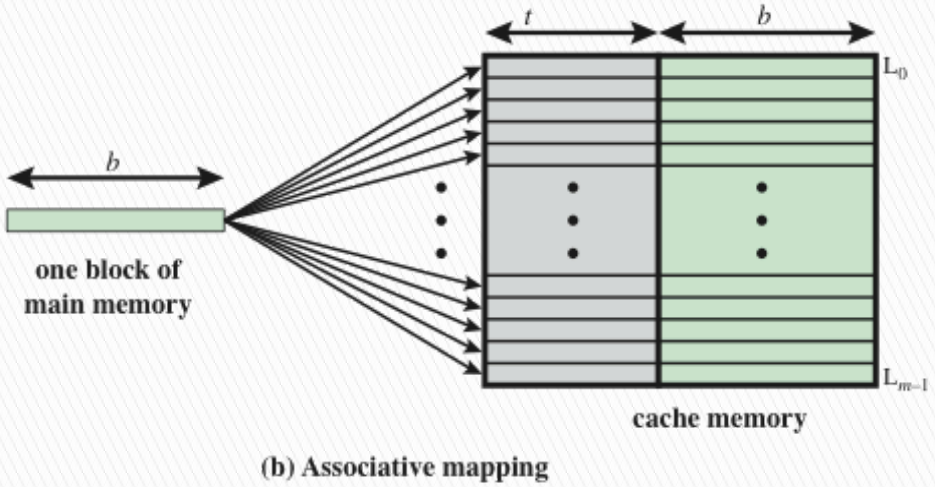


Figure 4.8 Mapping From Main Memory to Cache: Direct and Associative

Direct Mapping pros & cons

- Simple
- Inexpensive

- Fixed location for given block
  - If a program accesses 2 blocks that map to the same line repeatedly, cache misses are very high

Victim Cache

- Lower miss penalty
  - Originally proposed as an approach to reduce the conflict misses of direct mapped caches without affecting its fast access time
- Remember what was discarded
  - Already fetched
  - Use again with little penalty
- Fully associative
- 4 to 16 cache lines
- Between direct mapped L1 cache and next memory level

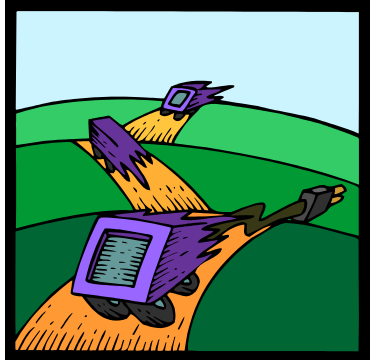
Associative Mapping

- A main memory block can load into any line of cache
- Memory address is interpreted as tag and word
- Tag uniquely identifies block of memory
- Every line's tag is examined for a match
- Cache searching gets expensive

Set Associative Mapping

- Compromise that exhibits the strengths of both the direct and associative approaches while reducing their disadvantages
- Cache is divided into a number of sets
- Each set contains a number of lines
- A given block maps to any line in a given set
  - e.g. Block B can be in any line of set I
- E.g. 2 lines per set
  - 2 way associative mapping
  - A given block can be in one of 2 lines in only one set

Replacement Algorithms



- Once the cache has been filled, when a new block is brought into the cache, one of the existing blocks must be replaced
- For direct mapping there is only one possible line for any particular block and no choice is possible
- For the associative and set-associative techniques a replacement algorithm is needed
- To achieve high speed, an algorithm must be implemented in hardware

The four most common replacement algorithms are:

- **Least recently used (LRU)**
  - Most effective
  - Replace that block in the set that has been in the cache longest with no reference to it
  - Because of its simplicity of implementation, LRU is the most popular replacement algorithm
- **First-in-first-out (FIFO)**
  - Replace that block in the set that has been in the cache longest
  - Easily implemented as a round-robin or circular buffer technique
- **Least frequently used (LFU)**
  - Replace that block in the set that has experienced the fewest references
  - Could be implemented by associating a counter with each line
- **Random**

Unified Versus Split Caches

- It has become common to split cache:
  - One dedicated to instructions
  - One dedicated to data
  - Both exist at the same level, typically as two L1 caches
- Advantages of unified cache:
  - Higher hit rate
    - Balances load of instruction and data fetches automatically
    - Only one cache needs to be designed and implemented
- Trend is toward split caches at the L1 and unified caches for higher levels
- Advantages of split cache:
  - Eliminates cache contention between instruction fetch/decode unit and execution unit
    - Important in pipelining

External Memory

- Magnetic Disk
  - Circular platter constructed of nonmagnetic material, called substrate, coated with a magnetizable material.
- Solid State Devices
  - Memory device made with solid state components that can be used as replacement to hard disk drive (HDD)
- Optical Memory
- Magnetic tape
  - Use the same reading and recording techniques as disk systems
  - Medium is flexible polyester tape coated with magnetizable material