

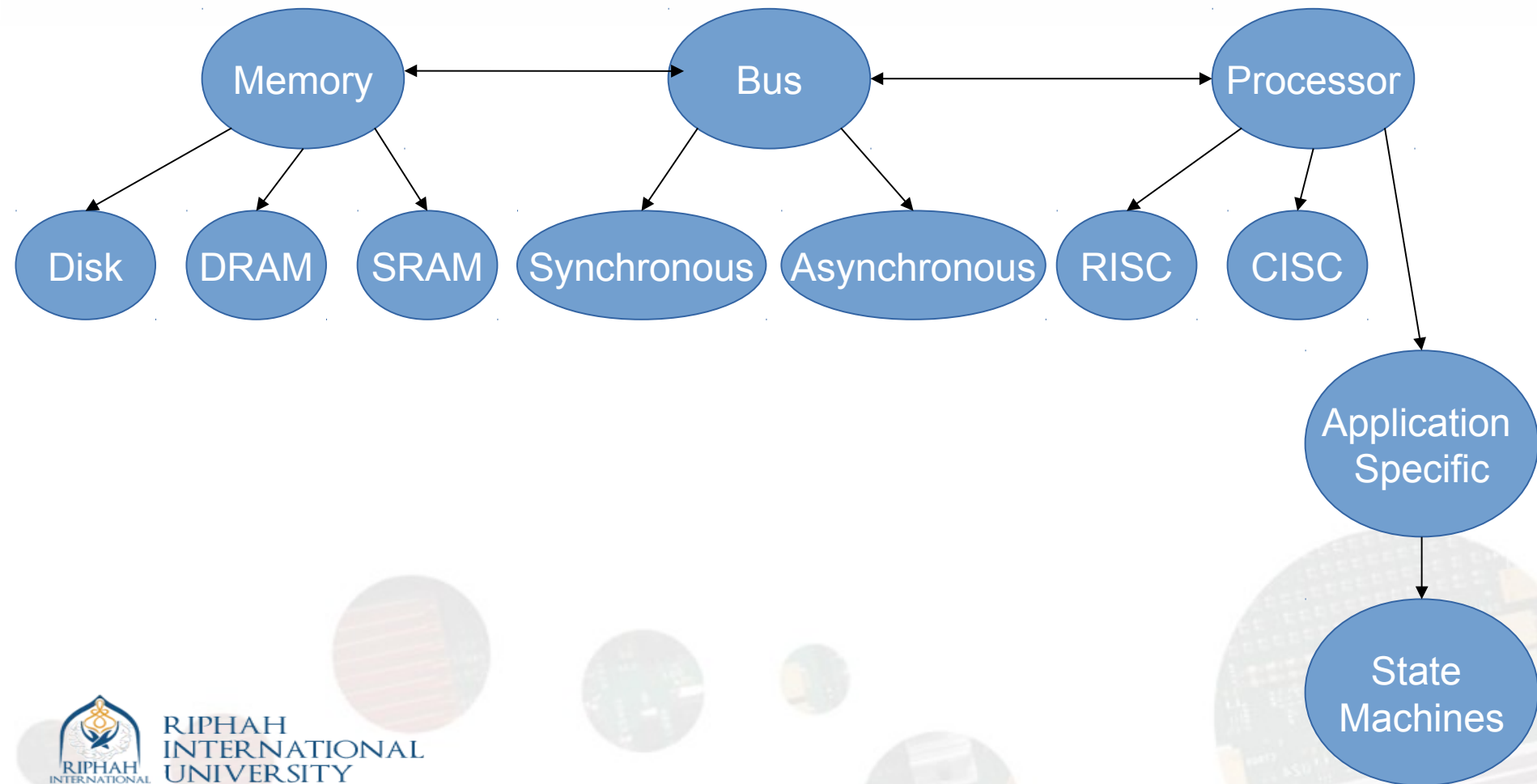
Memory & Processor Bus

Tassadaq Hussain

Riphah International University
Islamabad Pakistan

Microsoft Barcelona Supercomputing Center
Universitat Politècnica de Catalunya
Barcelona, Spain

Embedded System Components



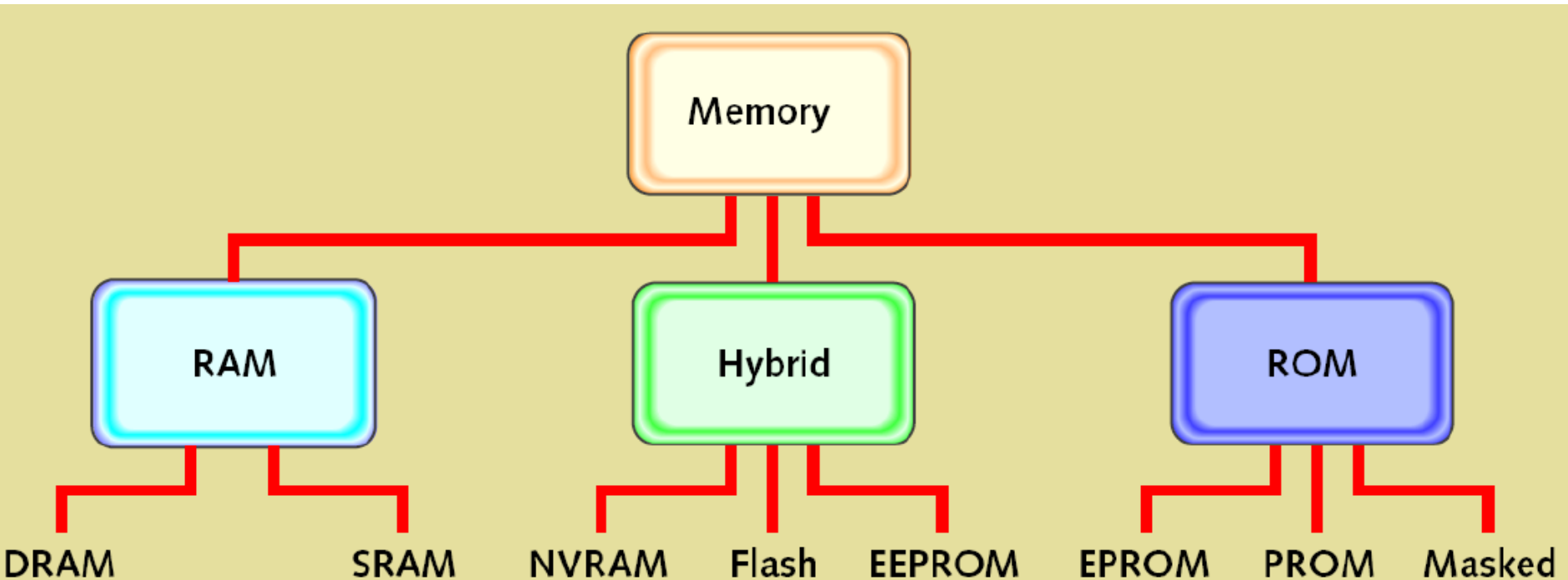
Contents

- Memory types
- Bus Systems

Types of Memories

RAM = Random Access Memory

ROM = Read Only Memory



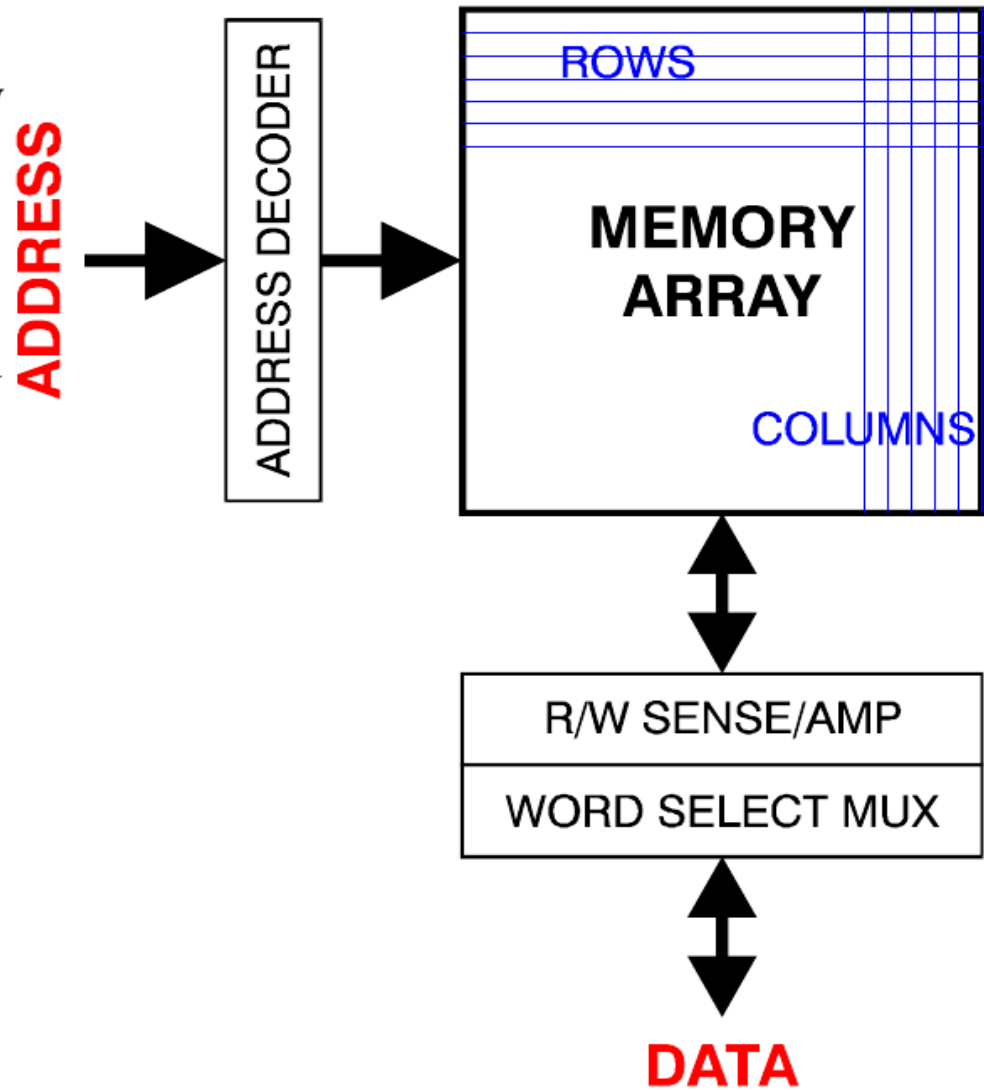
Memory Characteristics

Type	Volatile?	Writeable?	Erase Size	Erase Cycles	Cost/byte	Speed
SRAM	Yes	Yes	Byte	Unlimited	* Expensive	Fast
DRAM	Yes	Yes	Byte	Unlimited	Moderate	Moderate
Masked ROM	No	No	n/a	n/a	Inexpensive	Fast
PROM	No	Once, with a programmer	n/a	n/a	Moderate	Fast
EPROM	No	Yes, with a programmer	Entire chip	Limited (see specs)	Moderate	Fast
EEPROM	No	Yes	Byte	Limited (see specs)	* Expensive	Fast to read, slow to write
Flash	No	Yes	Sector	Limited (see specs)	Moderate	Fast to read, slow to write
NVRAM	No	Yes	Byte	Unlimited	* Expensive	Fast

Memory Hardware Management

◆ 2-D array composed of identical memory cells

- Address decoder selects one row
- Sense amps detect and amplify memory cell value
- Word select takes a subset of columns that have the byte/word of interest (mux = multiplexor)

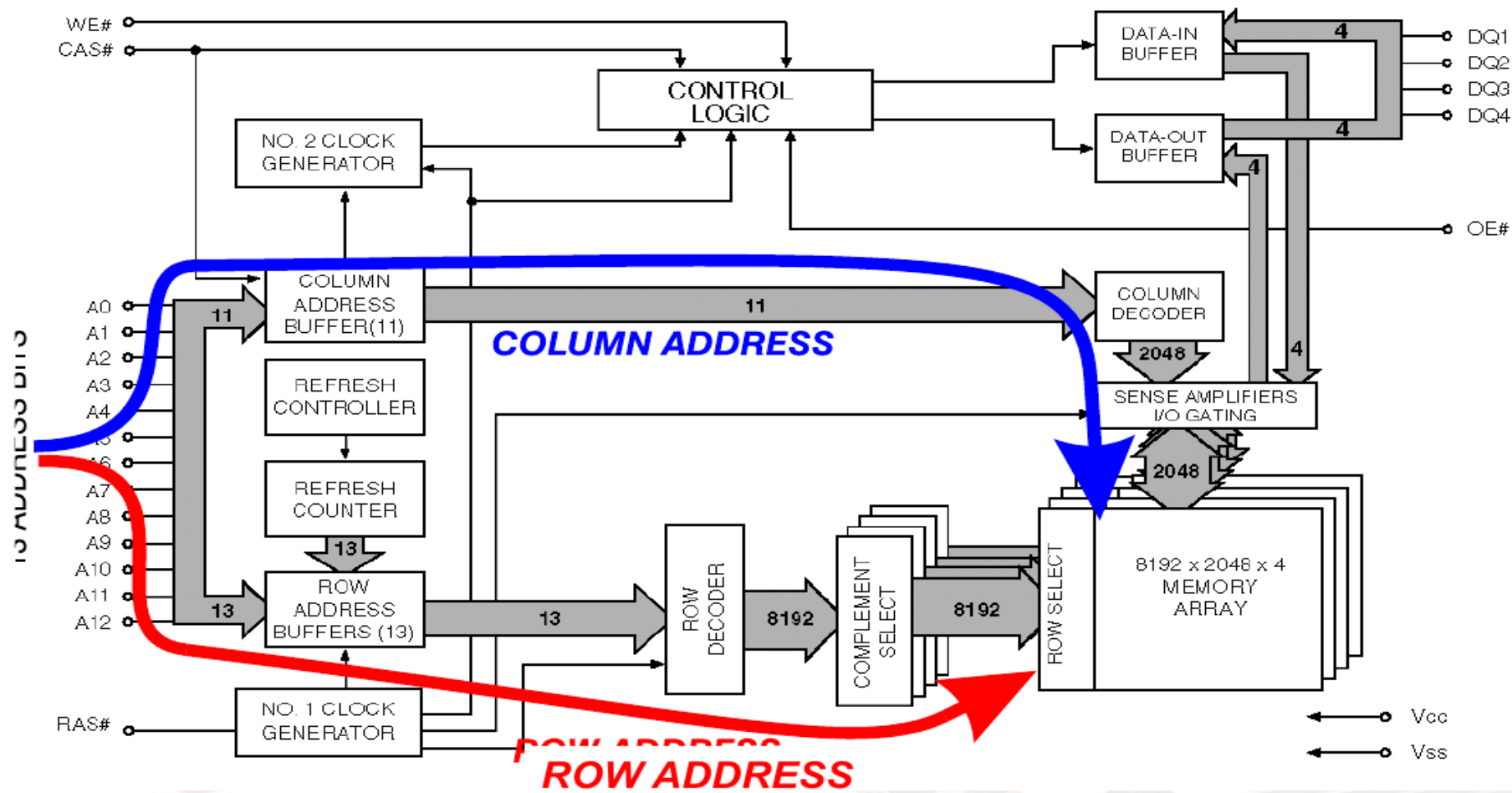


◆ Memory cell construction varies

- Speed vs. density
- Volatile vs. non-volatile

DRAM Internal Organization

FUNCTIONAL BLOCK DIAGRAM
MT4LC16M4A7 (13 row addresses)



Multiplexed Addresses

◆ SRAM chips have a pin for every address line

- Gives fast access, which is what SRAM is all about
- For example, 64K bit x 1 chip has 16 address lines
- For example, 256K bit x 8 (2 Mbit chip) has 18 address pins; 8 data pins

◆ DRAMS split the address in half (multiplex high and low bits)

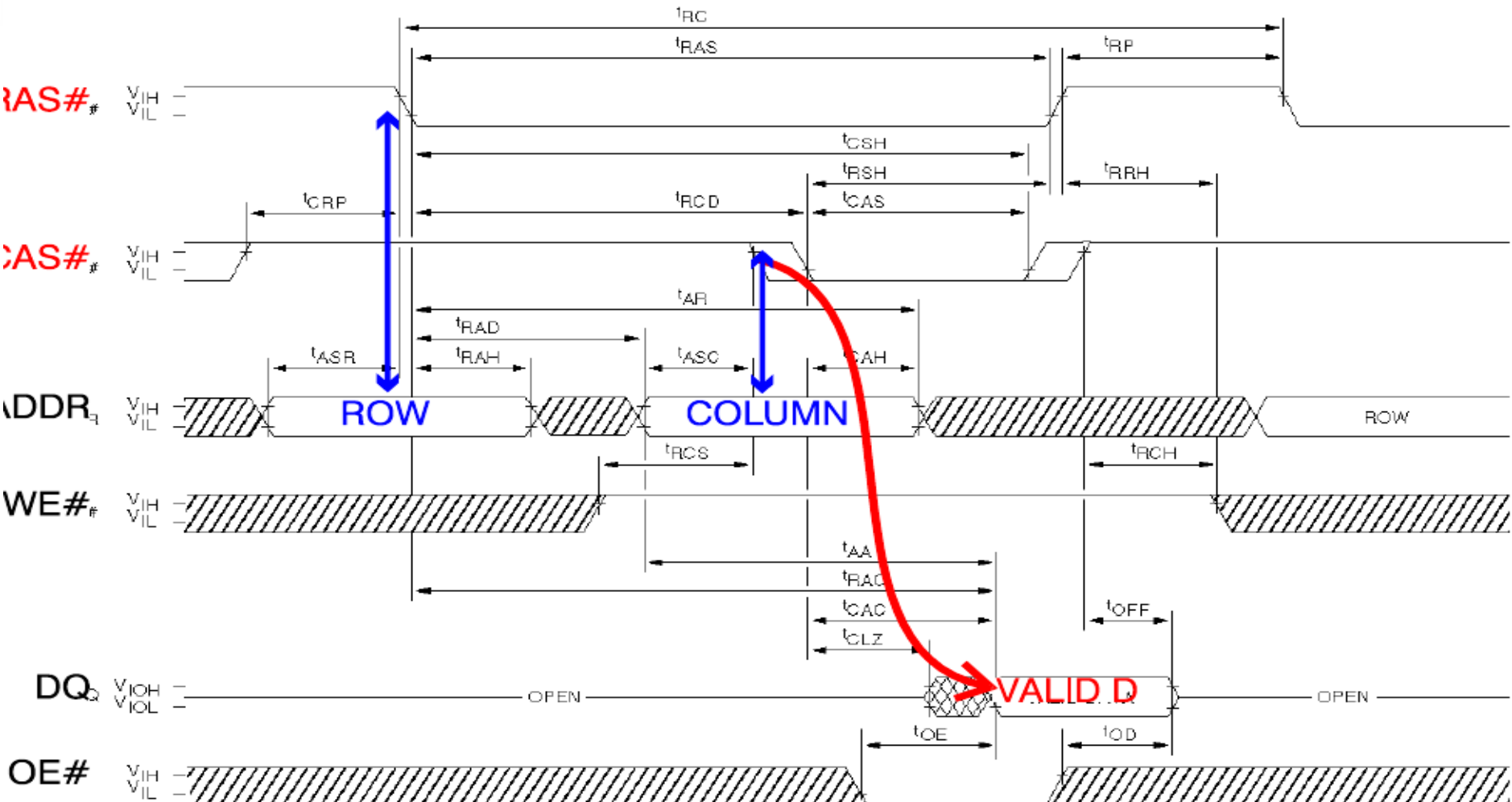
- The top 8 bits were the row address
- Then bottom 8 bits selected one column (the column address)
- This organization reduces the DRAM pin count – same pins for both Row & Col
 - 8 address bits can be sent at a time, in sequence
 - Only 8 pins and two strobe signals
 - vs. 16 pins and a strobe signal
 - Also ties in with the internal memory organization

Address



DRAM : Read Cycle

READ CYCLE



Non-Volatile RAM Technologies

- ◆ **Sometimes memory has to survive a power outage**
 - On desktop machines this is (mostly) done by hard disk
 - Many embedded systems don't have magnetic storage (cost, reliability, size)
- ◆ **Battery backed SRAM (fairly rare now that EEPROM is cheap)**
 - Mold a battery right into the SRAM plastic chip case
 - Just as fast & versatile as SRAM
 - Typically retains data for 4-7 years (usually limited by battery shelf life)
 - Cost includes both SRAM and a dedicated battery
- ◆ **FRAM**
 - Relatively new technology – in the marketplace, but not mainstream (yet)
 - Ferroelectric RAM
 - Unlimited read/write cycles
 - Intended as non-volatile drop-in replacement for SRAM (still expensive)

Read Only Memory

- ◆ **Masked ROM – pattern of bits built permanently into silicon**
 - Historically the most dense (least expensive) NV memory
 - BUT – need to change masks to change memory pattern (\$\$\$\$, lead time)
 - Every change means building completely new chips!
 - It also means throw the old chips away ... they can't be changed
- ◆ **Masked ROM seldom used in low-end embedded systems**
 - Too expensive to make new chips every time a change is needed
 - Takes too long (multiple weeks) to get the new chips

PROM Types

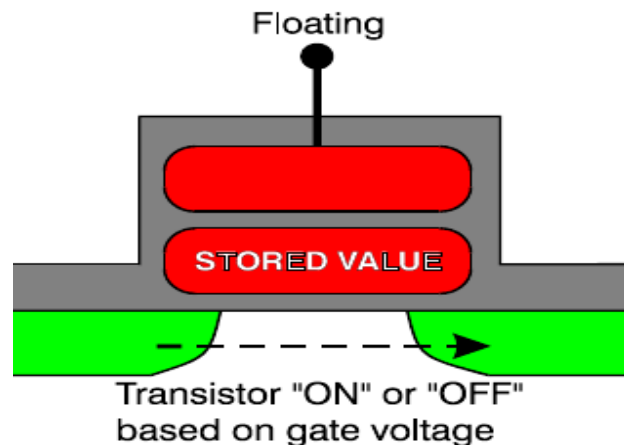
- ◆ **PROM: Programmable Read-Only Memory**
 - Generic term for non-volatile memory that can be modified
- ◆ **OTPROM – “One Time” PROM**
 - Can only be programmed a single time (think “blowing fuses” to set bit values)
 - Holds data values indefinitely
- ◆ **EPROM – “Eraseable” PROM**
 - Entire chip erased at once using UV light through a window on chip
 - Mostly obsolete and replaced by flash memory
- ◆ **EEPROM – “Electrically Eraseable” PROM**
 - Erasure can be accomplished in-circuit under software control
 - Same general operation as flash memory EXCEPT...
 - ...EEPROM can be erased/rewritten a byte at a time
 - Often have both flash (for bulk storage) and EEPROM (for byte-accessible writes) in same system

Flash Memory

Flash Memory Operation

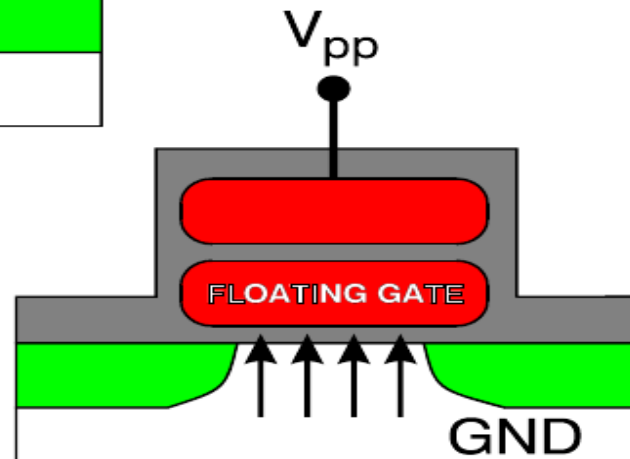
◆ Flash memory (and EEPROM, etc.) hold data on a floating transistor gate

- Gate voltage turns transistor on or off for reading data
 - Usually, erasure results in all “1” values
- Erase/program cycles wear out the gate
 - E.g., max 100K cycles for NOR flash
 - E.g., max 1M cycles for NAND flash
- Data retention can be 100 years+
- Cheaper than EEPROM; not byte modifiable

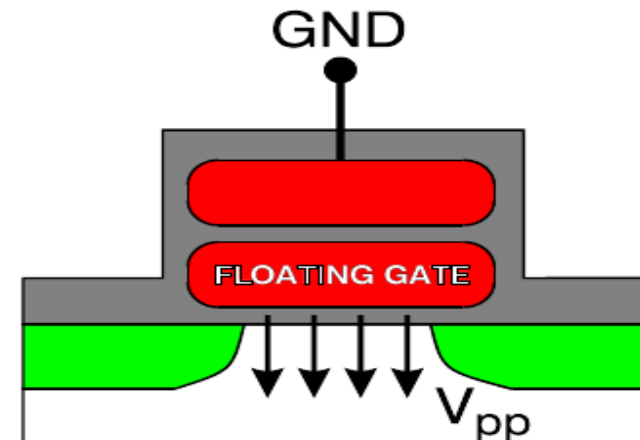


Operate

**NAND Flash
PROM Operation**



Program



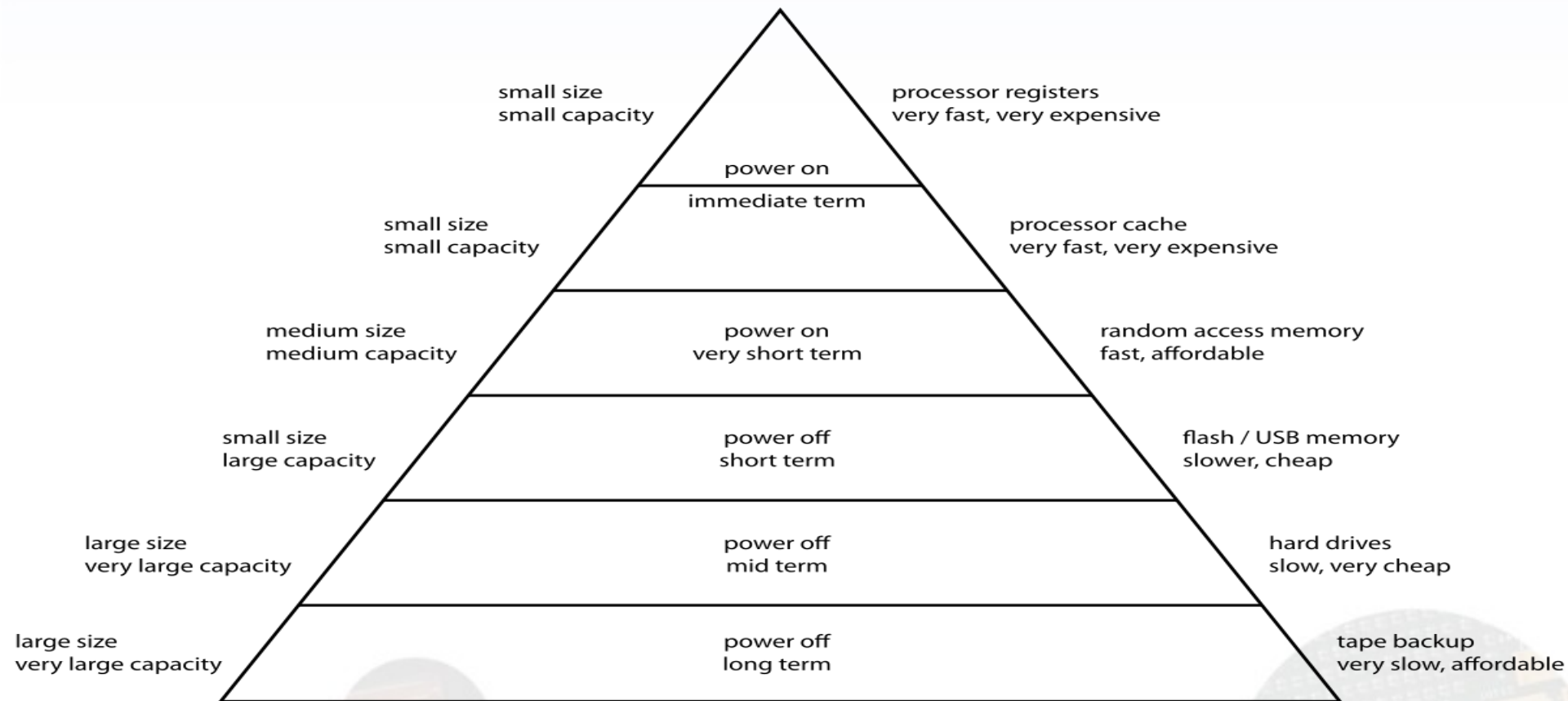
Erase

Memory Hierarchy

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

Memory Hierarchy - Diagram

Computer Memory Hierarchy



Characteristics

- Location
- Capacity
- Unit of transfer
- Access method
- Performance
- Physical type
- Physical characteristics
- Organisation

Location

- CPU
- Internal
- External

Capacity

- Word size
 - The natural unit of organisation
- Number of words
 - or Bytes

Unit of Transfer

- Internal
 - Usually governed by data bus width
- External
 - Usually a block which is much larger than a word
- Addressable unit
 - Smallest location which can be uniquely addressed

Access Methods (1)

- Sequential
 - Start at the beginning and read through in order
 - Access time depends on location of data and previous location
 - e.g. tape
- Direct
 - Individual blocks have unique address
 - Access is by jumping to vicinity plus sequential search
 - Access time depends on location and previous location
 - e.g. disk

Access Methods (2)

- Random
 - Individual addresses identify locations exactly
 - Access time is independent of location or previous access
 - e.g. RAM
- Associative
 - Data is located by a mechanism based on placement
 - Access time is independent of location or previous access
 - e.g. cache

Performance

- Access time
 - Time between presenting the address and getting the valid data
- Transfer Rate
 - Rate at which data can be moved

Physical Types

- Semiconductor
 - RAM
- Magnetic
 - Disk & Tape
- Optical
 - CD & DVD
- Others
 - Bubble
 - Hologram

Physical Characteristics

- Switching
- Decay
- Volatility
- Erasable
- Power consumption

Organisation

- Physical arrangement of bits into words
- Not always obvious
- e.g. interleaved

The Bottom Line

- How much?
 - Capacity
- How fast?
 - Time is money
- How expensive?

Storage Hierarchy List

- Registers
- L1 Cache
- L2 Cache
- Main memory
- Disk cache

So you want fast?

- It is possible to build a computer which uses only static RAM (see later)
- This would be very fast
- This would need no cache
 - How can you cache cache?
- This would cost a very large amount

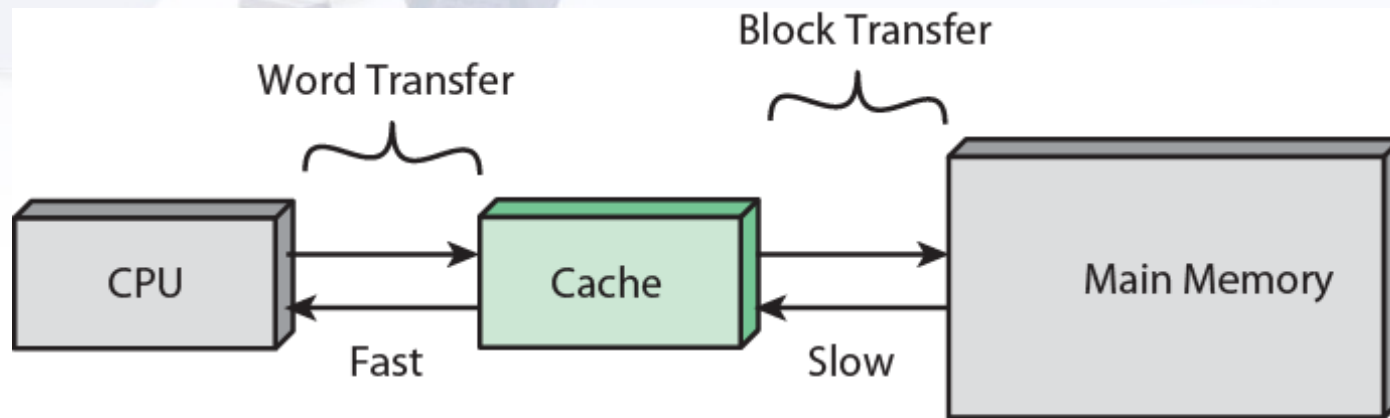
Local Memory System

- Cache
- Scratchpad

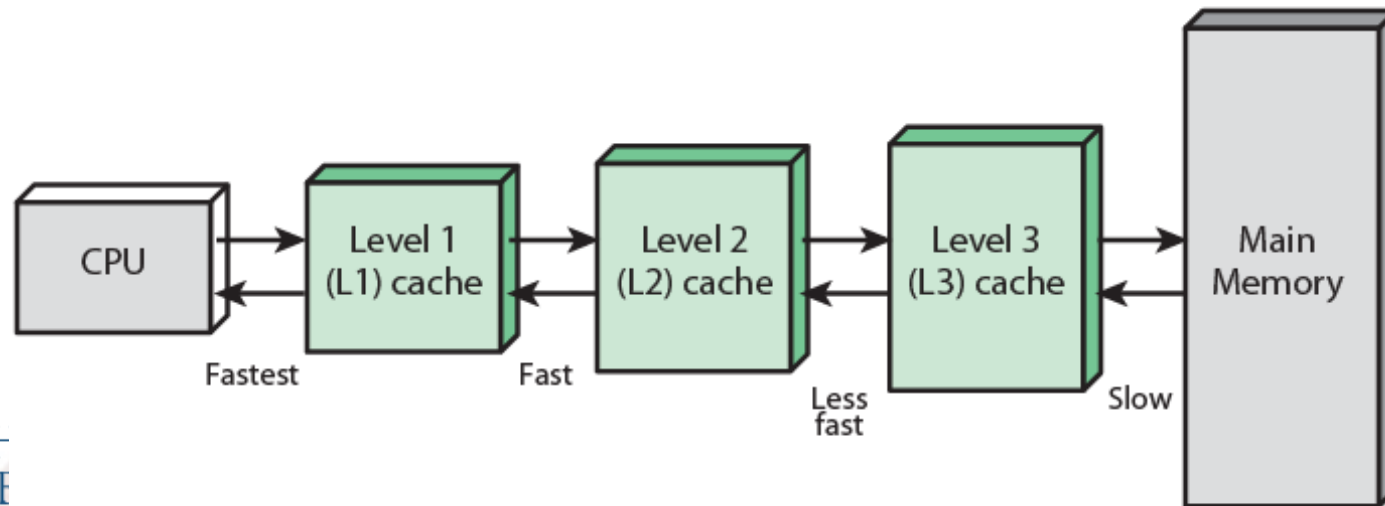
Cache

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

Cache and Main Memory

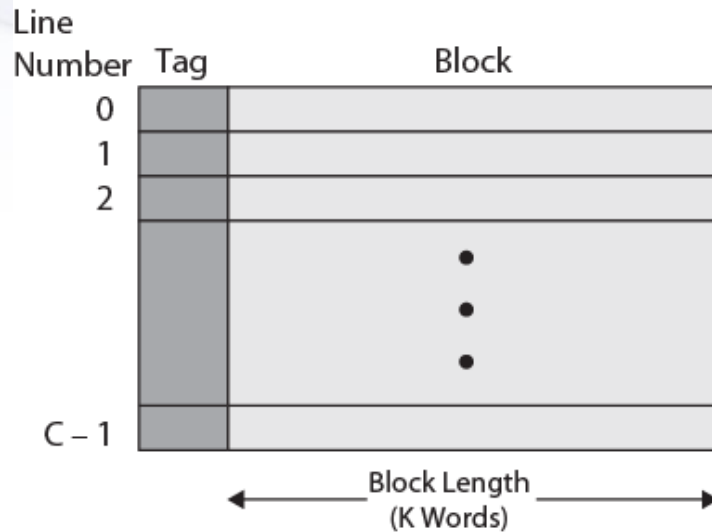


(a) Single cache

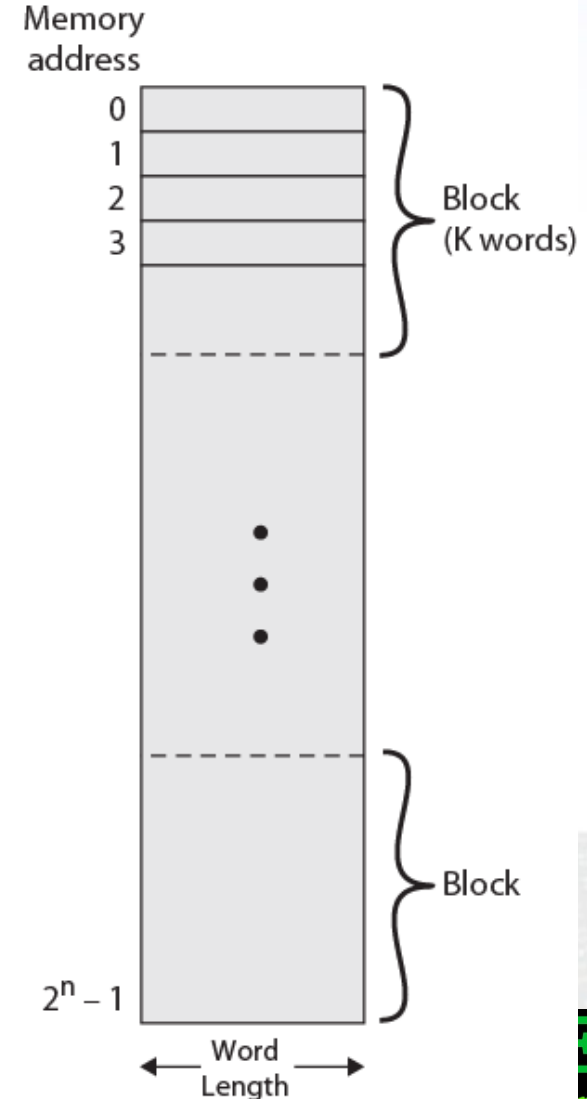


(b) Three-level cache organization

Cache/Main Memory Structure



(a) Cache



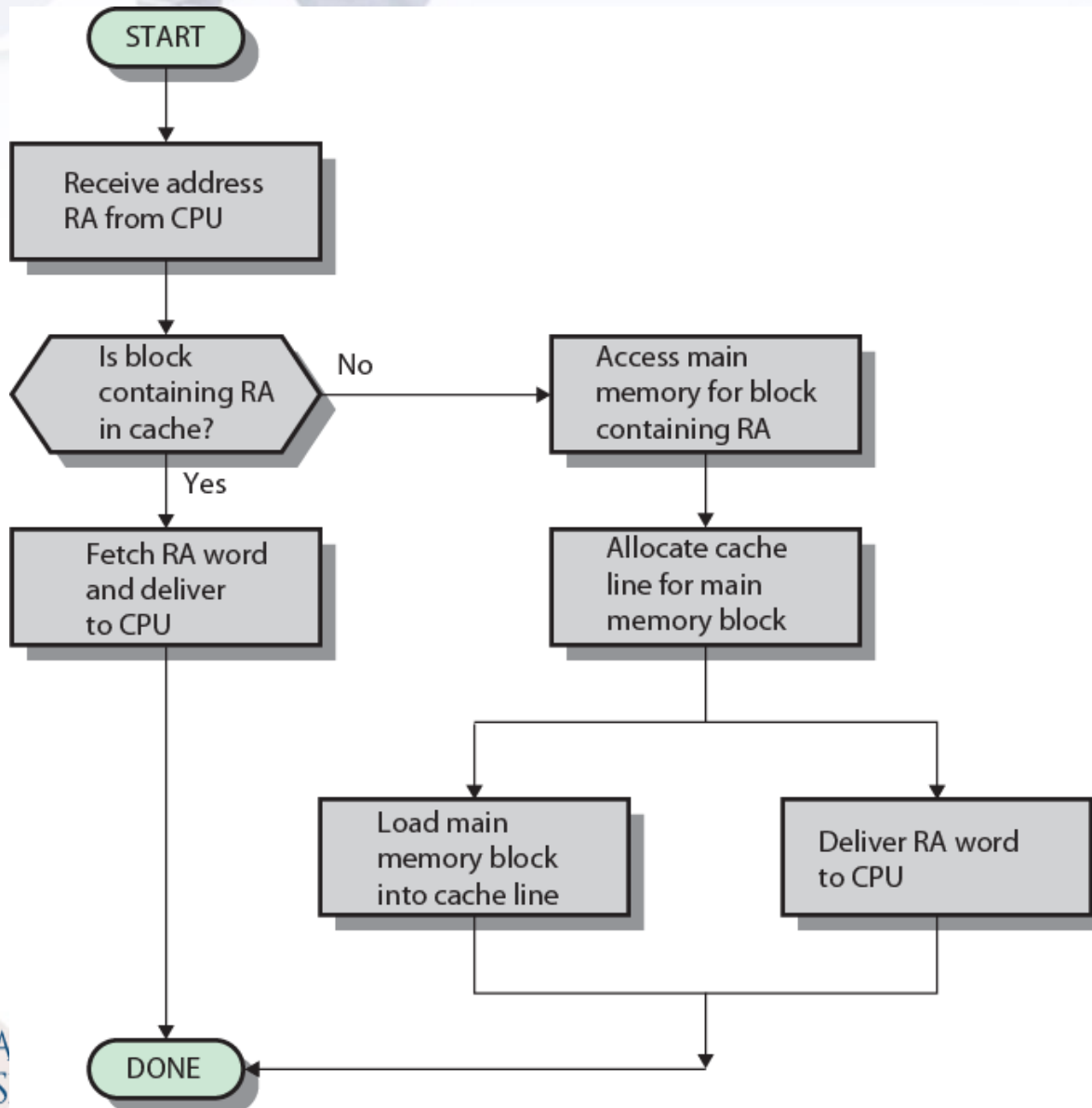
(b) Main memory

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

slot

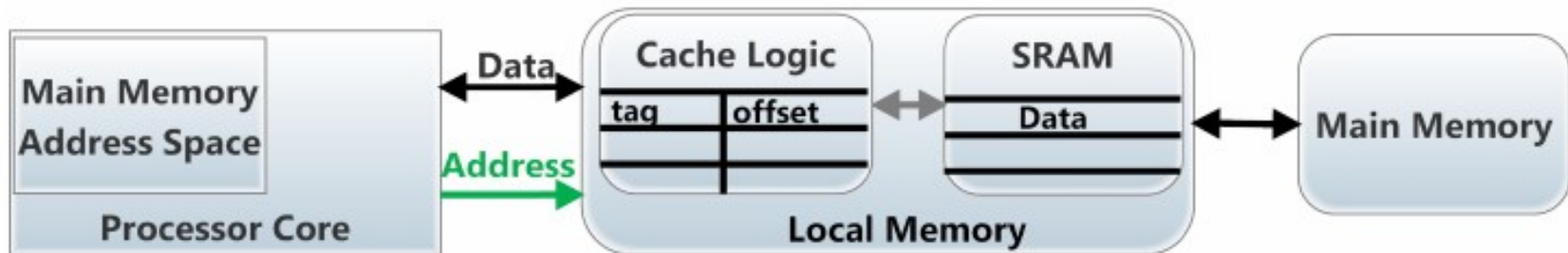
Cache Read Operation - Flowchart



Cache Design

- Addressing
- Size
- Mapping Function
- Replacement Algorithm
- Write Policy
- Block Size
- Number of Caches

A Conventional Memory System Architecture



Cache

- Caches are present in most memory systems.
- The Cache dynamically stores a subset of the frequently used data. Thus, the timing of a load or store operation depends on the relationship between its effective address and the effective addresses of earlier operations.

- Conventional Cache used byte addressable memory.
- 2^b = size of a Cache
- 2^B = size of Main Memory
- Addr = Address of Main Memory
- CL = Data Transfer from/to the memory
- NCL = Number of Cache lines (CLs)
- CLS = Cache Line Size

Direct-Mapped Cache

- Direct Mapped cache is an array of fixed size blocks.
- Each block holds consecutive bytes of main memory data.

Fully associative cache

- A fully associative cache.
- A fully associative cache permits data to be stored in any cache block, instead of forcing each memory address into one particular block. — When data is fetched from memory, it can be placed in any unused block of the cache.

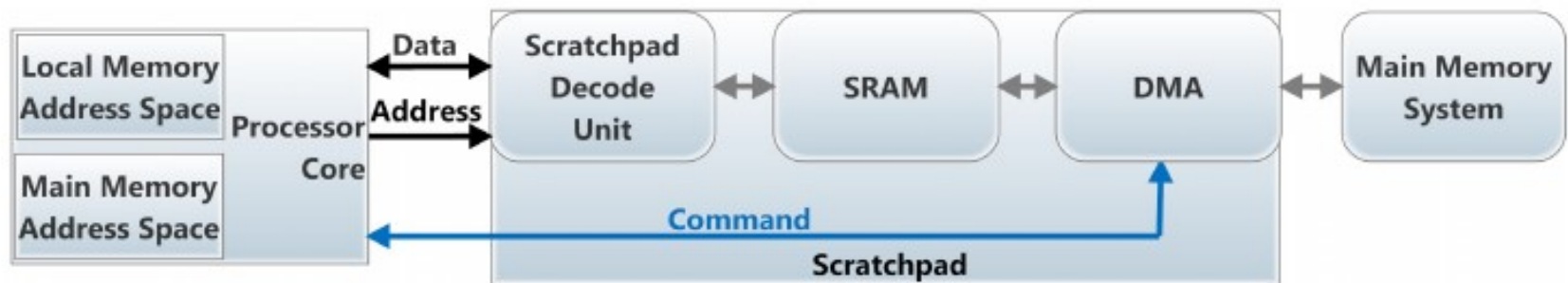
Assignment

- You have 16 bit of address bus
- The maximum size of main memory 64K byte (byte addressable)
- Design a cache having $NCL = 8$ and $CLS = 4K$
 - Find out required memory for cache

Set Associative Cache

- A set-associative scheme is a hybrid between a fully associative cache, and direct mapped cache.
- It's considered a reasonable compromise between the complex hardware needed for fully associative caches (which requires parallel searches of all slots), and the simplistic direct-mapped scheme, which may cause collisions of addresses to the same slot (similar to collisions in a hash table).

Scratchpad

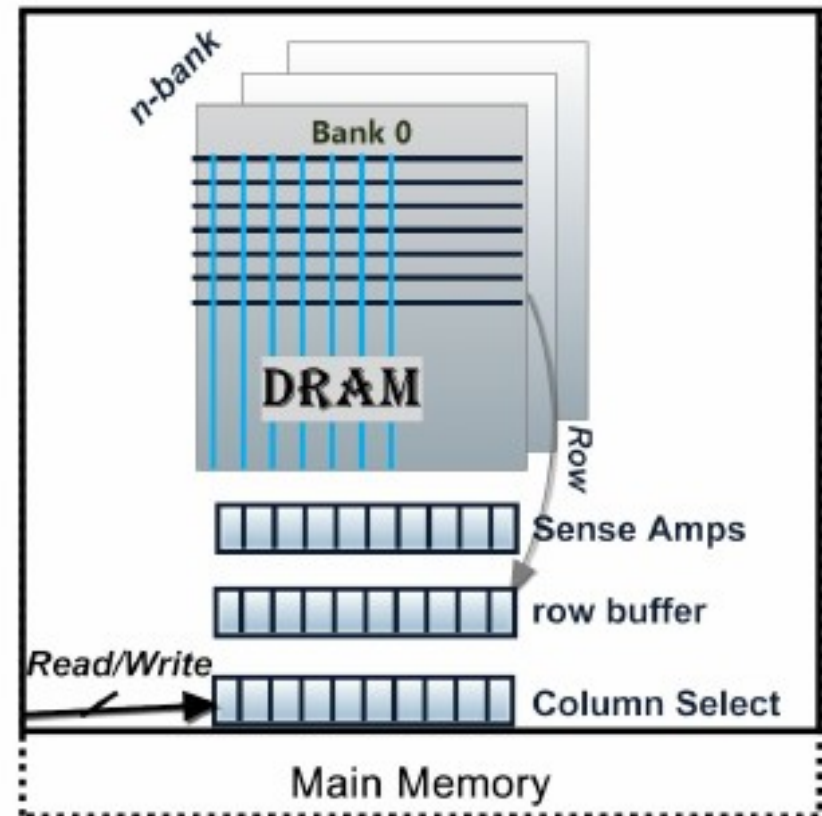


Scratchpad

- The Scratchpad is a fast directly addressed software managed SRAM memory.
- The Scratchpad has better real-time guarantees than caches and by its significantly lower overheads it is better in access time, energy consumption and area.
- Instead of using traditional load/store instructions the scratchpad uses direct memory-memory operations using DMA.
- The Scratchpad memory access uses source and destination address registers, each of which holds a starting address of the memory.

Main Memory System

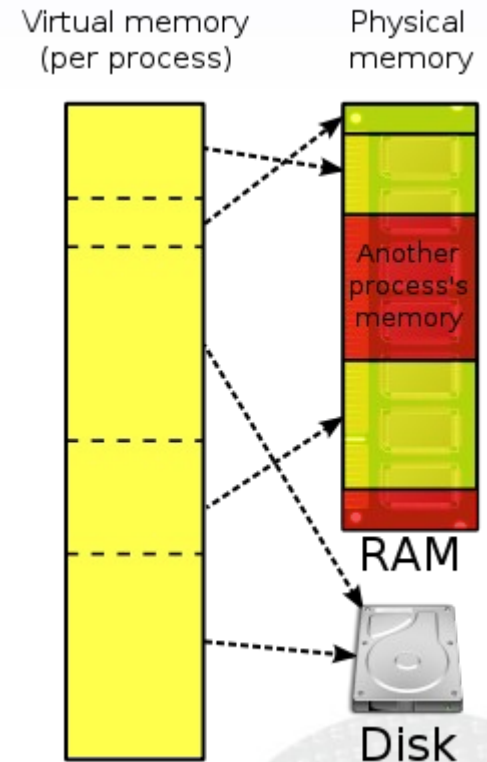
- DRAM is combination Row x Column
- Row Address
- Column Address



Virtual Memory

- Virtual memory is a memory management capability of an OS that uses hardware and software to allow a computer to compensate for physical memory shortages by temporarily transferring data from random access memory (RAM) to disk storage

- The operating system, using a combination of hardware and software, maps memory addresses used by a program, called virtual addresses, into physical addresses in computer memory.
- Main storage, as seen by a process or task, appears as a contiguous address space or collection of contiguous segments.



Memory Bus

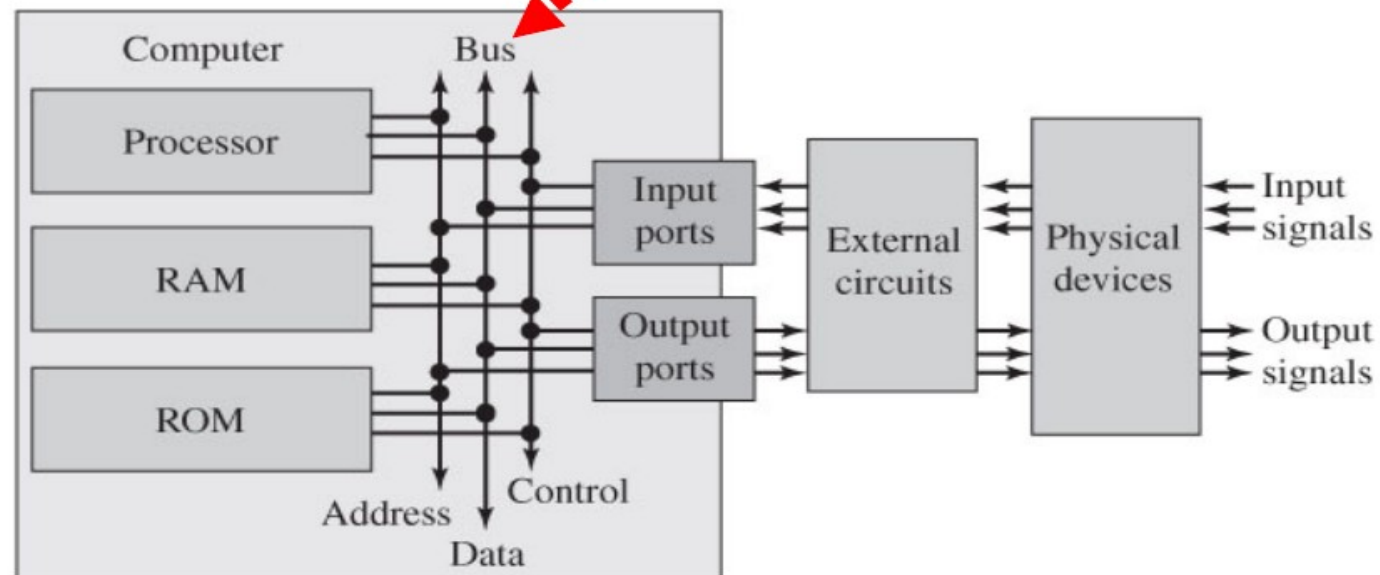
◆ Used to transfer data to and from processor

- Various types of memory
- I/O data as well
- Carries: address, data and control signals

**“Memory” Bus
also does I/O**

Figure 1.1
The basic components of a computer system include processor, memory, and I/O.

[Valvano]



Memory Accessing

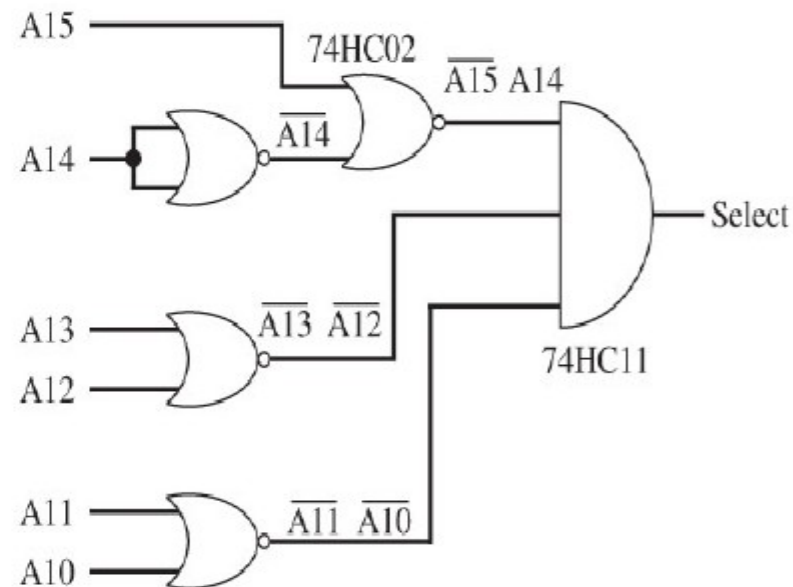
◆ Every device on bus must recognize its own address

- Must decide which of multiple memory chips to activate
- Each I/O port must decide if it is being addressed
- High bits of address decoded to “select” device; low bits used within device

◆ “Memory Mapped” I/O

- I/O devices and memory share same address space (e.g., Freescale)
- Alternative: separate memory and I/O control lines (e.g., Intel)
- What address does this decode?

Figure 9.7
An address decoder identifies on which cycles to activate.



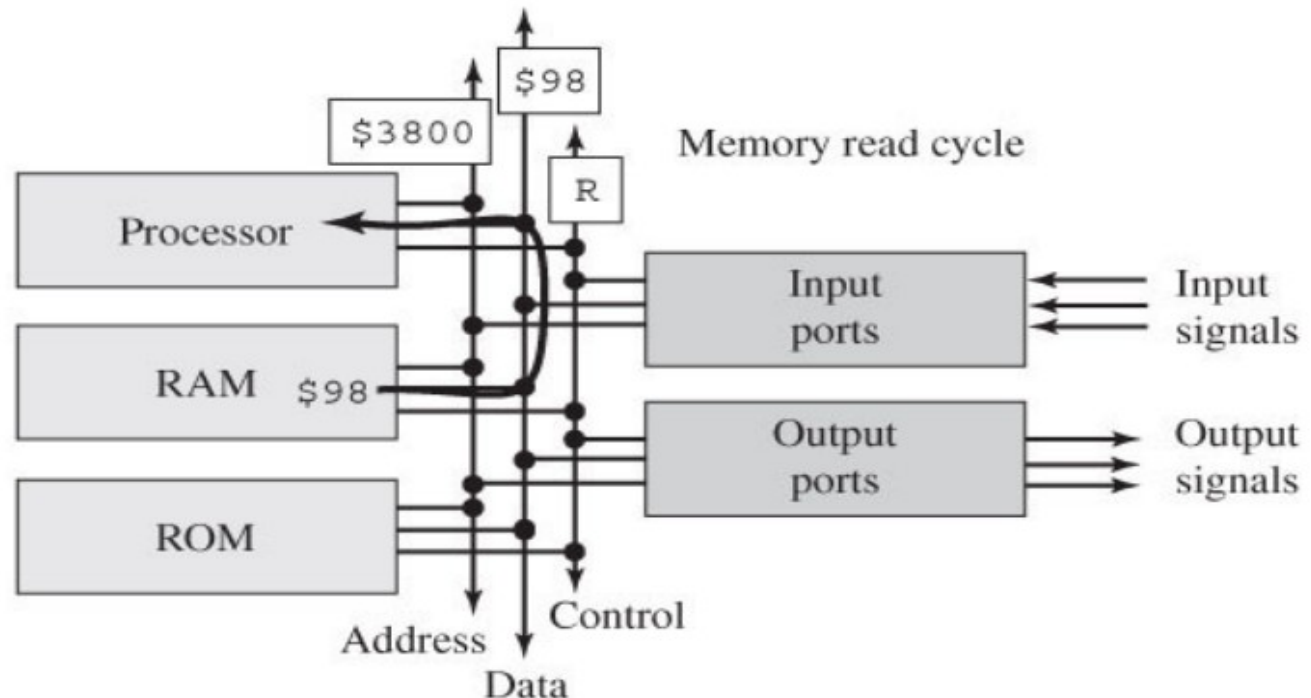
Bus Interconnections

◆ Processor bus (“memory bus”) connects CPU to memory and I/O

- Data lines – actually transfers data
- Address lines – feed memory address and I/O port number
- Control lines – provides timing and control signals to direct transfers
- Sometimes these lines are shared to reduce hardware costs

A memory read cycle copies data from RAM, ROM, or an input device into the processor.

[Valvano]



Bus typical parameters

◆ Clock

- System clock so other devices don't have to have their own oscillators
- Drives bus timing for synchronous transfers

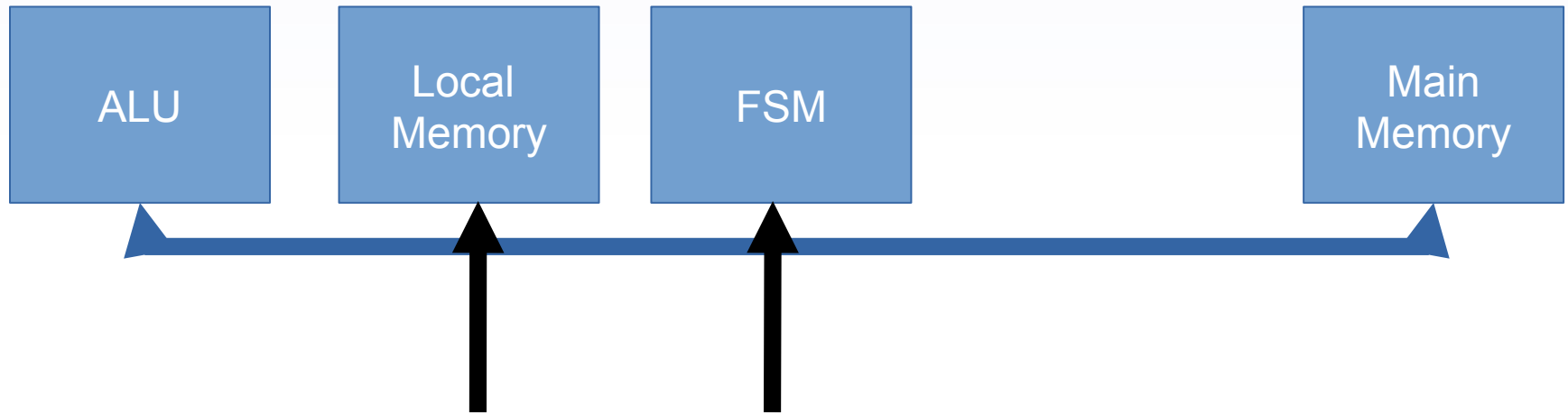
◆ Address & Data

- Used for memory R/W, I/O, and DMA
- Sometimes multiplexed, sometimes separate
- Sometimes address is multiplexed (high/low) to make DRAM interface simpler

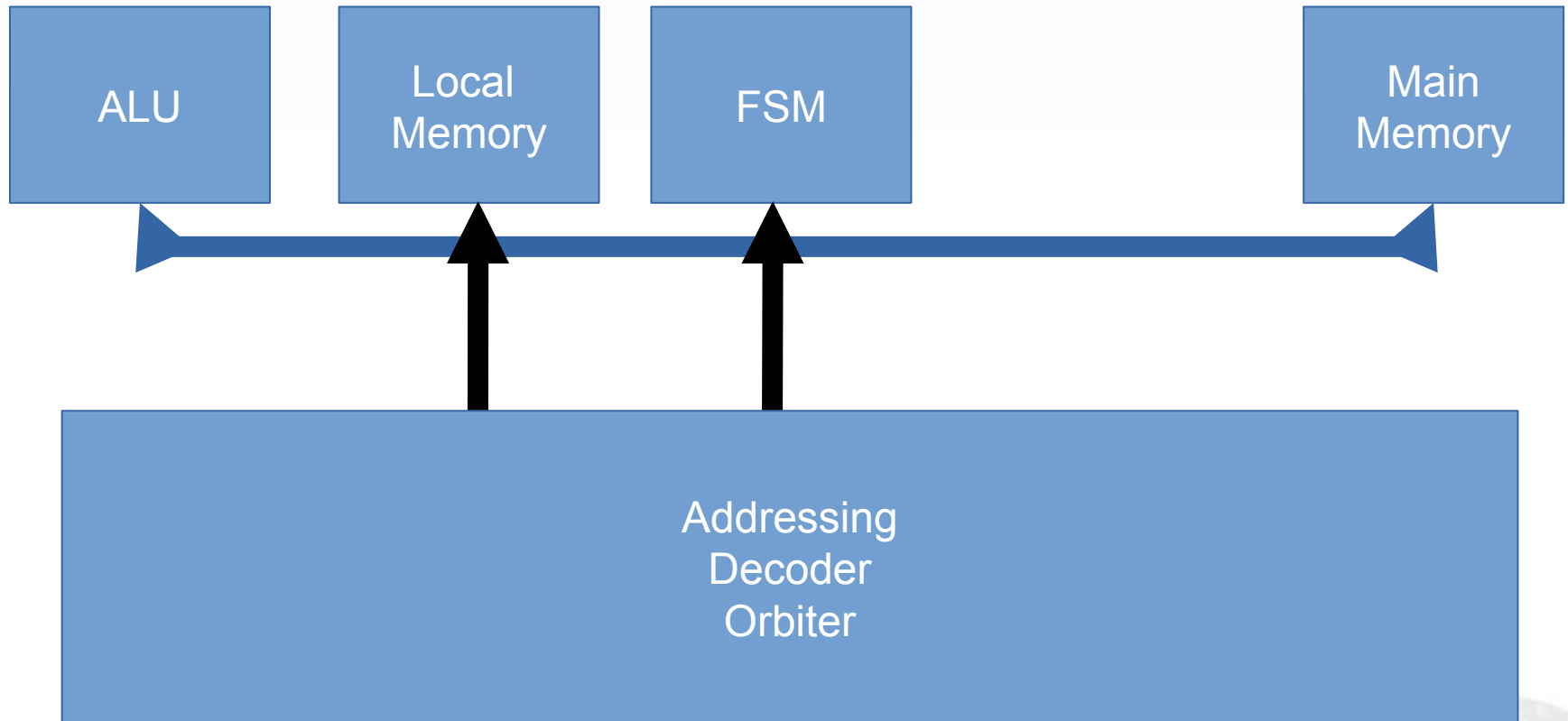
◆ Control signals

- Read/write – which way is data moving?
- Memory vs. I/O – if they are separate address spaces (Intel, not Freescale)
- Byte vs. word – is it a whole word, or just a byte?
- Device controls – interrupt request/grant; DMA request/grant; etc.

Multi System Architecture



Multi System Architecture



Address Space

An address space defines a range of discrete addresses, each address space of which may correspond to different digital systems, such as peripheral device, disks, memories or other logical or physical entity.

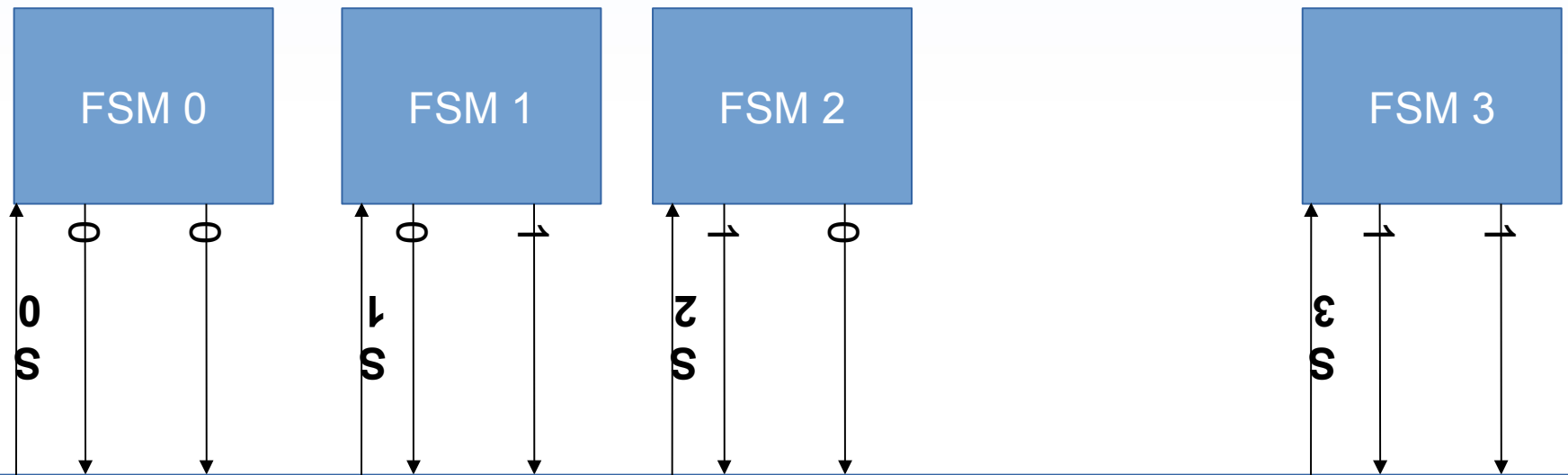
Example

A system has 8 bit address bus.

How many components can be accessed by the system.

$$2^{\text{address bus}} = 256$$

Address Bus = 2 bit



Addressing
Decoder
If address = 00
FSM = Enable
Else.....

Address Mapping

For 8 bit address bus we have Address range
from

$$0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 = 0x00 = 0$$

$$1\ 1\ 1\ 1\ 1\ 1\ 1\ 1 = 0xFF = 256$$

Address Allocation

- Depending upon requirement we allocate addresses.
- Suppose we have an I/O interface which controls 8 LEDs, 64 Byte Memory, Stepper Motor having 4 inputs and 64 Byte Main Memory.
- How many addresses are required to these components?

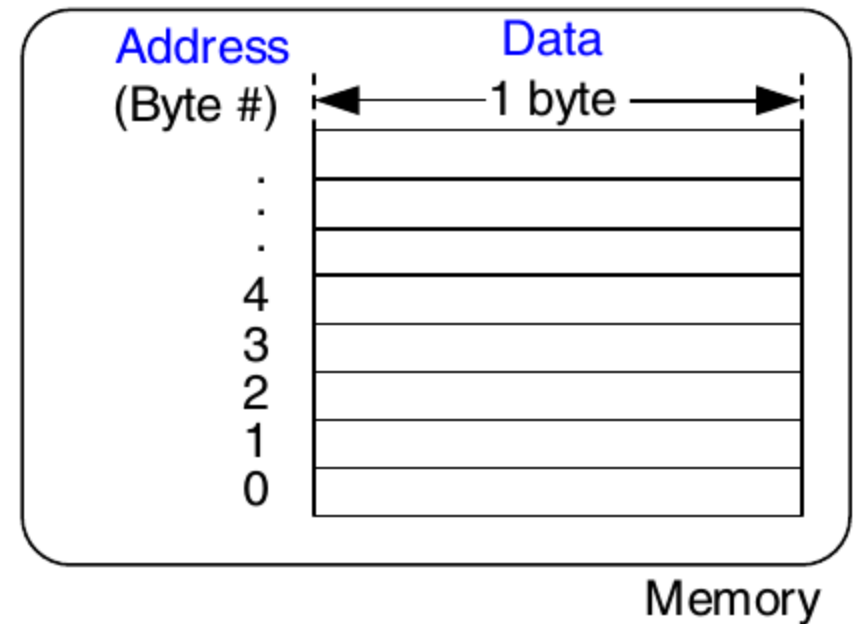
8 Bit LED => 8 Addresses => 0000000 – 0000 0111

Local Memory => 64 Addresses => 0000000 – 0011 1111

Stepper => 4 Addresses => 0000000 – 0000 0011

Main Memory => 64 Addresses => 0000000 – 0011 1111

Memory



Arbitration

An arbiter is a device used in a multi-system to decide which core/system is allowed to control the data transfer.

