Computer Architecture

Course Code: CSE360

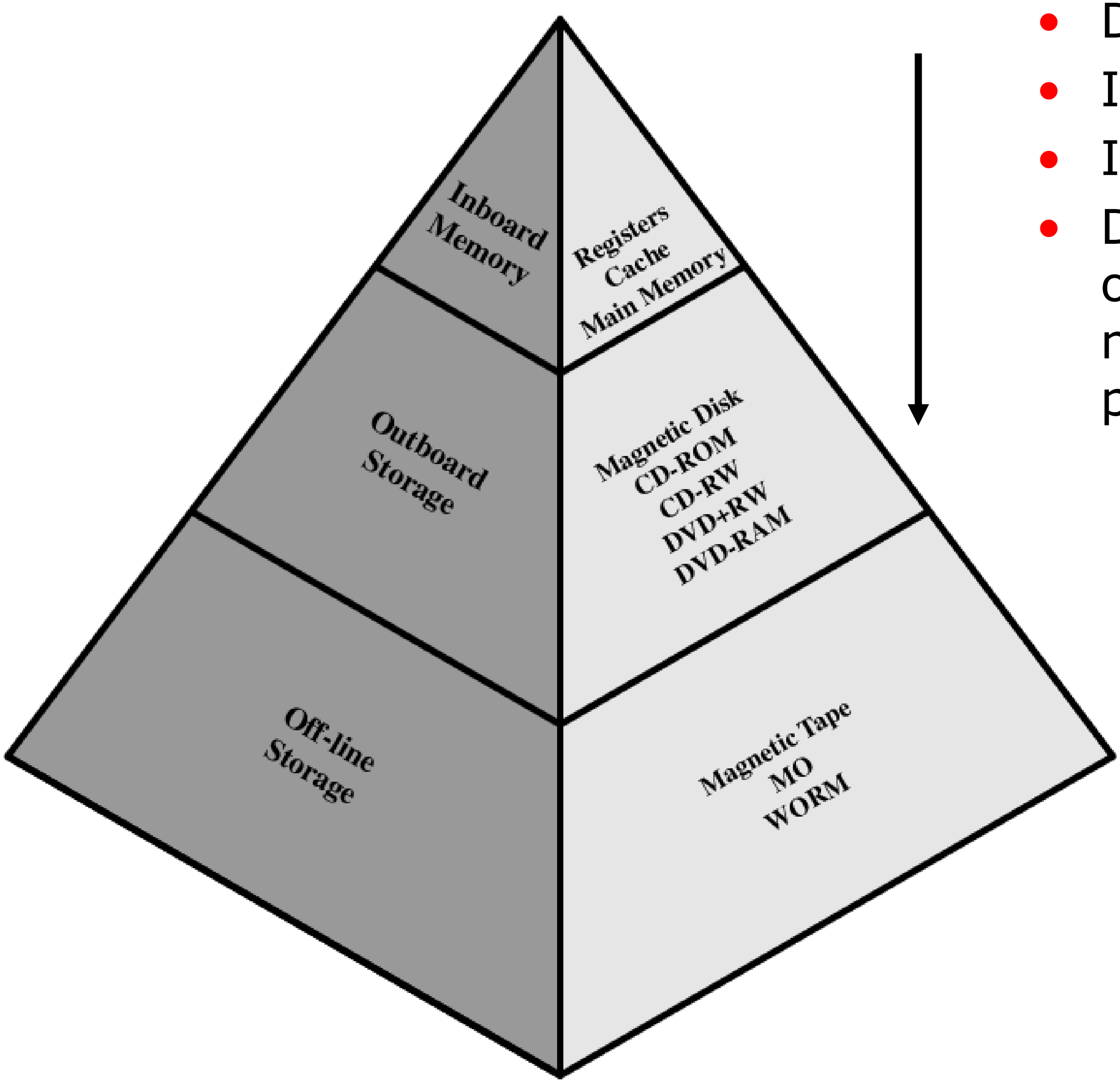
Lecture - 4

Cache Memory

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



- Decreasing cost per bit
- Increasing capacity
- Increasing access time
- Decreasing frequency of access of the memory by the processor

- Faster access time, greater cost per bit
- Greater capacity, smaller cost per bit
- Greater capacity, slower access time

Performance

- Access time
  - For random access memory, the time it takes to perform a read or write operation, i.e. time between presenting the address and getting the valid data
  - For non-random access memory, access time is the time it takes to position the read-write mechanism at the desired location
- Memory Cycle time
  - Consists of access time plus any additional time required before a second access can commence.
  - Time may be required for the memory to "recover" before next access
  - Cycle time = access time + recovery
- Transfer Rate
  - Rate at which data can be moved into or out of a memory unit.
  - For random access memory, it is equal to 1/(cycle time)
  - For non-random access memory, $T_N = T_A + \frac{N}{R}$

Performance

$$T_N = T_A + \frac{N}{R}$$

T_N = Average time to read or write N bits

T_A = Average access time

N = Number of bits

R = Transfer rate in bits per second (bps)

Physical Types

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

Physical Characteristics

- Decay, Volatility, Erasable, Power consumption
- In a volatile memory, information decays naturally or is lost when electrical power is switched off.
- In a non-volatile memory, information once recorded remains without deterioration until deliberately changed; no electric power is needed to retain the information.
- Magnetic-surface memories are non-volatile.
- Semiconductor memory may be either volatile or non-volatile.
- Non-erasable unit cannot be altered except by destroying the storage unit. Semiconductor memory of this type is known as read-only memory (ROM); must be also non-volatile.

Organisation

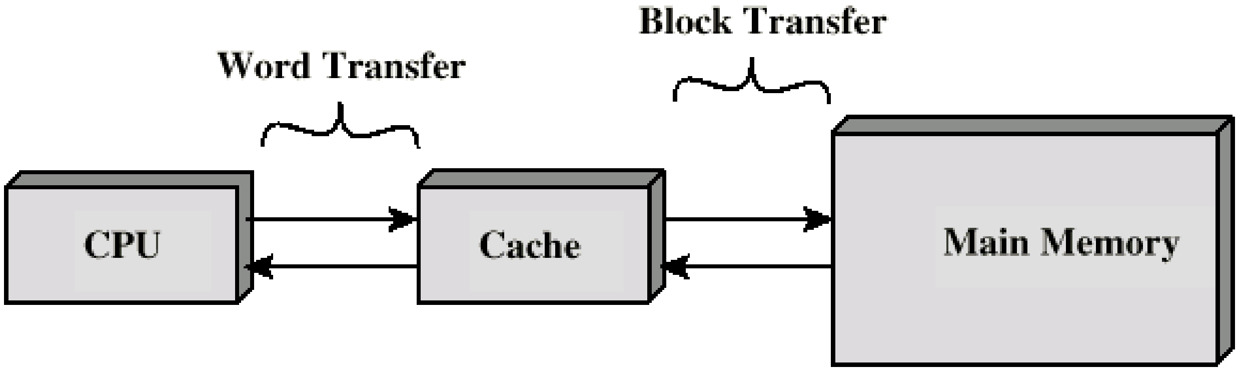
- For random-access memory, the organisation is a key design issue
- Organisation means physical arrangement of bits to form words
- The obvious arrangement is not always used

Hierarchy List

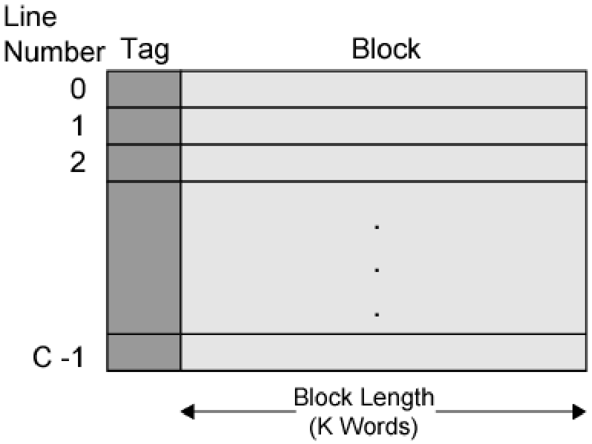
- Registers
- L1 Cache
- L2 Cache
- Main memory
- Disk cache
- Disk
- Optical
- Tape

Cache

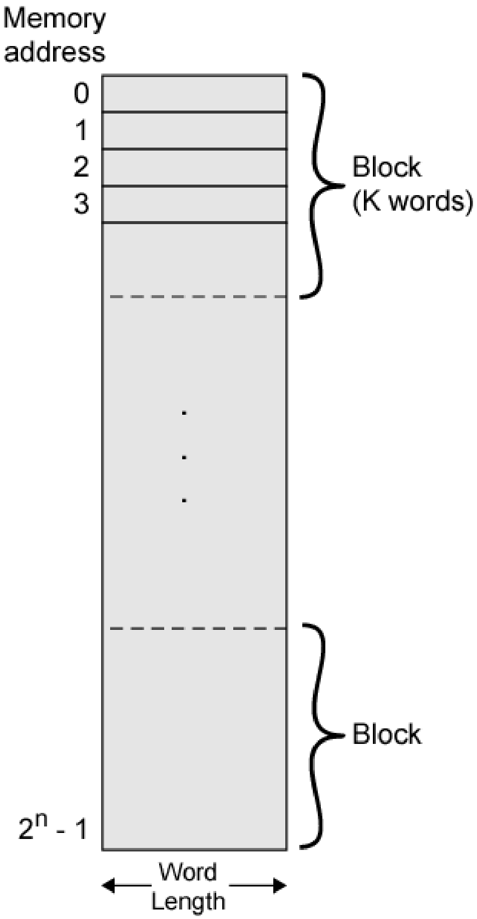
- Cache memory gives the fastest memory and at the same time provides a large memory size at the price of less expensive types of semiconductors memories.
- Sits between normal main memory (slower and relatively larger) and CPU
- May be located on CPU chip or module
- The cache contains a copy of portions of main memory.
- When the processor attempts to read a word of memory, a check is made to determine if the word is in the cache. If so, the word is delivered to the processor. If not, a block of main memory consisting of some fixed number of words is read into the cache and then it is delivered to processor.



Cache/Main Memory Structure



(a) Cache



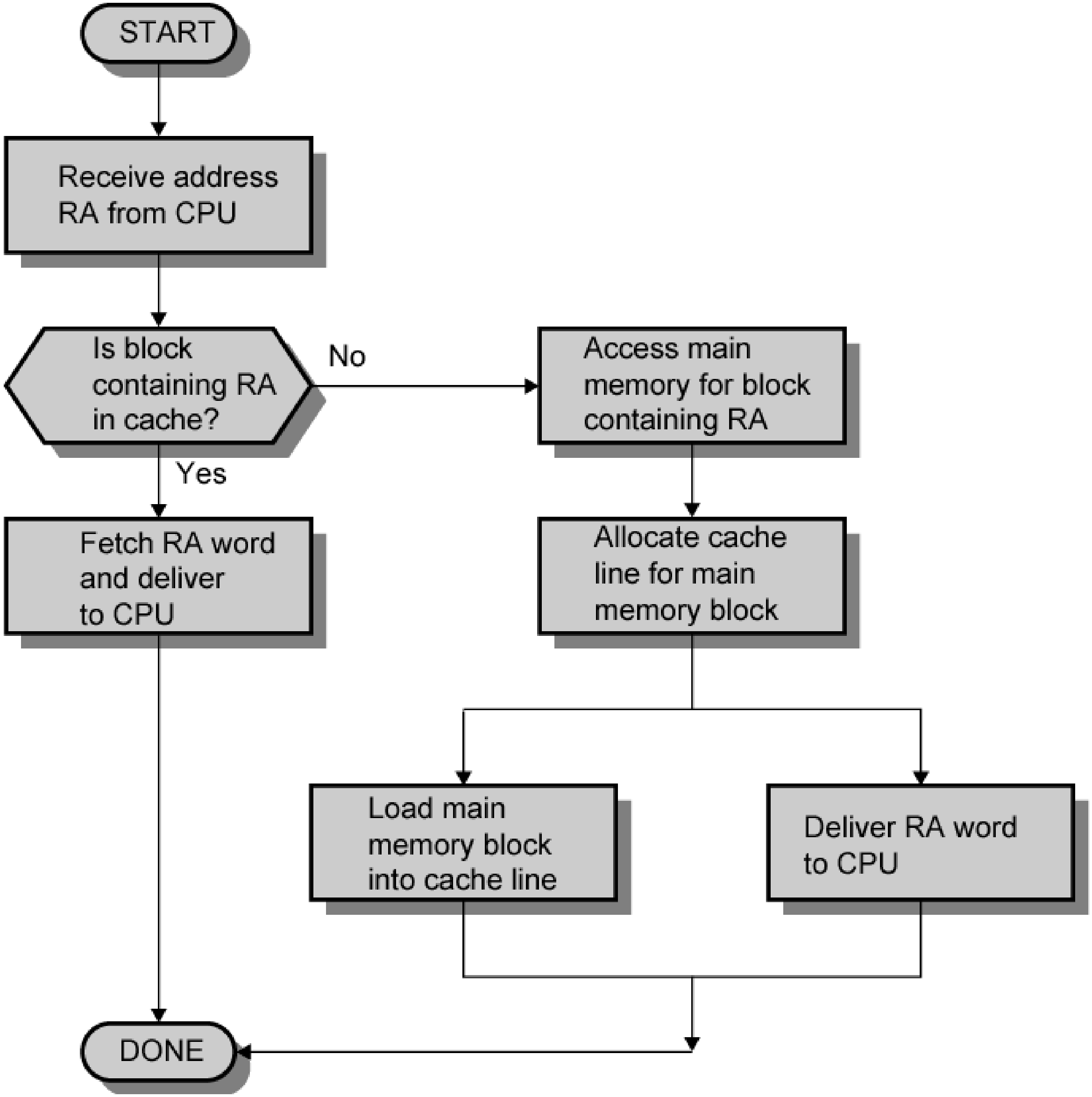
(b) Main memory

- Main memory consists of up to 2^n addressable words, with each word having a unique *n*-bit address.
- For mapping purposes, this memory is considered to consist of a number of fixed-length blocks of *K* words each. That is, there are $M=2^n/K$ blocks.
- The cache consists of *C* lines. Each line contains *K* words, plus a tag of a few bits; the number of words in the line is referred to as the line size.
- The number of lines in cache is considerably less than the number of main memory blocks (*C*<<*M*).
- If a word in a block of memory is read, that block is transferred to one of the lines of the cache.
- As there are more blocks than lines, an individual line cannot be uniquely and permanently dedicated to a particular block. Thus, each line includes a tag that identifies which particular block is currently being stored.

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

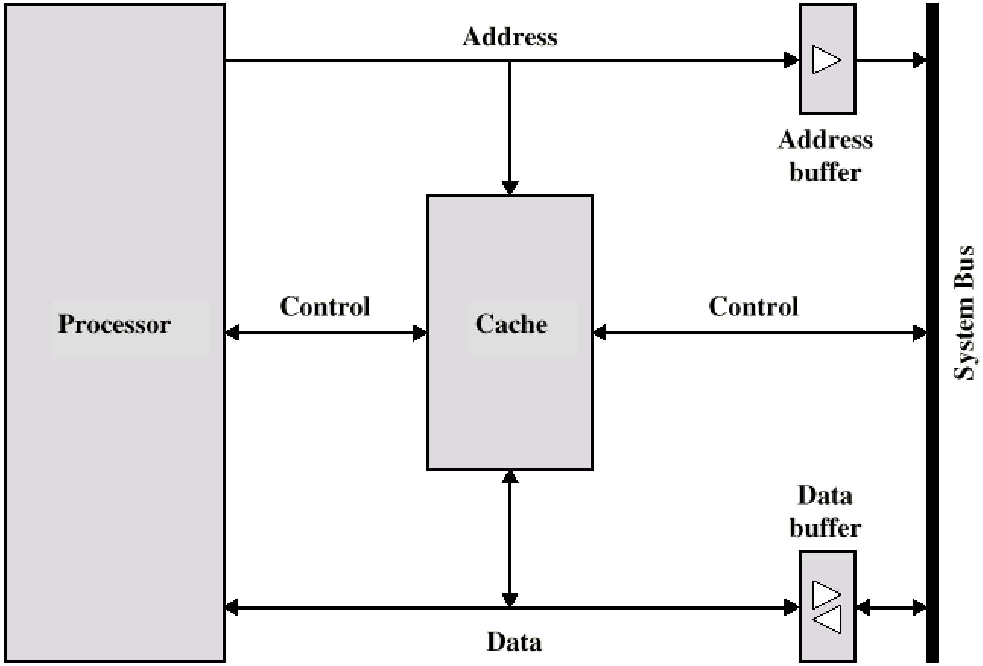
Cache Read Operation - Flowchart


START
Receive address RA from CPU
Is block containing RA in cache?
No
Access main memory for block containing RA
Yes
Fetch RA word and deliver to CPU
Allocate cache line for main memory block
Load main memory block into cache line
Deliver RA word to CPU
DONE

Size does matter

- Cost
  - More cache is expensive
- Speed
  - More cache is faster (up to a point)
  - Checking cache for data takes time

Typical Contemporary Cache Organization



- The cache connects to the processor via data, control, and address lines.
- The data and address lines also attach to data and address buffers which attach to a system bus from which main memory is reached.
- When a cache hit occurs, the data and address buffers are disabled and communication is only between processor and cache.
- When a cache miss occurs, the desired address is loaded onto the system bus and data are returned through the data buffer to both cache and processor.

Pentium 4 Cache Organization

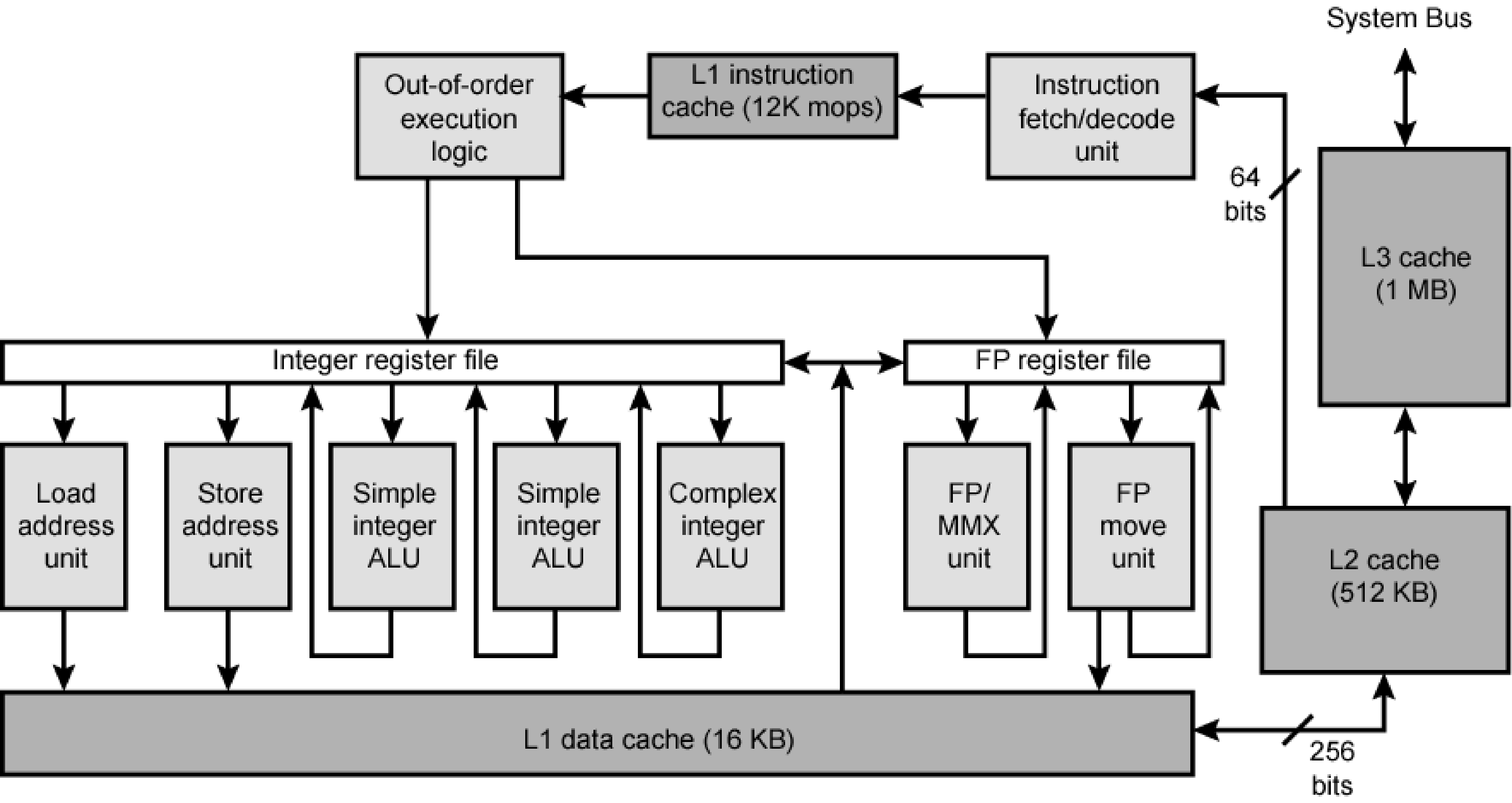
- 80386 – no on chip cache
- 80486 – includes single on-chip cache of 8 Kbytes, using 16 byte line size
- Pentium (all versions) – two on chip L1 caches
  - Data Cache (8KB) & instructions cache (8KB)
- Pentium III – L3 cache added off chip
- Pentium 4
  - L1 data cache is 8 Kbytes, line size of 64 bytes organization
  - L2 cache
    - Feeding both L1 caches
    - 128 bytes line size
  - L3 cache on chip

Pentium 4 Core Processor

- The processor core consists of four major components:
- Fetch/Decode Unit
  - Fetches program instructions from L2 cache
  - Decode into a series of micro-operations
  - Store micro-operations in L1 instruction cache
- Out of order execution logic
  - Schedules execution of micro-operations
  - Subject to data dependencies and resources availability
  - Thus, micro-operations may be scheduled in a different order than they fetched from instruction stream
  - This unit schedules speculative execution of micro-operations that may be required in future
- Execution units
  - Execute micro-operations
  - Fetching required data from L1 data cache
  - Results stored in registers temporarily
- Memory subsystem
  - This unit includes the L2 and L3 caches and the system bus which is used to access main memory when the L1 and L2 caches have a cache miss, and to access system I/O resources

- MOV AX, 5
- MOV BX,10
- ADD AX,BX

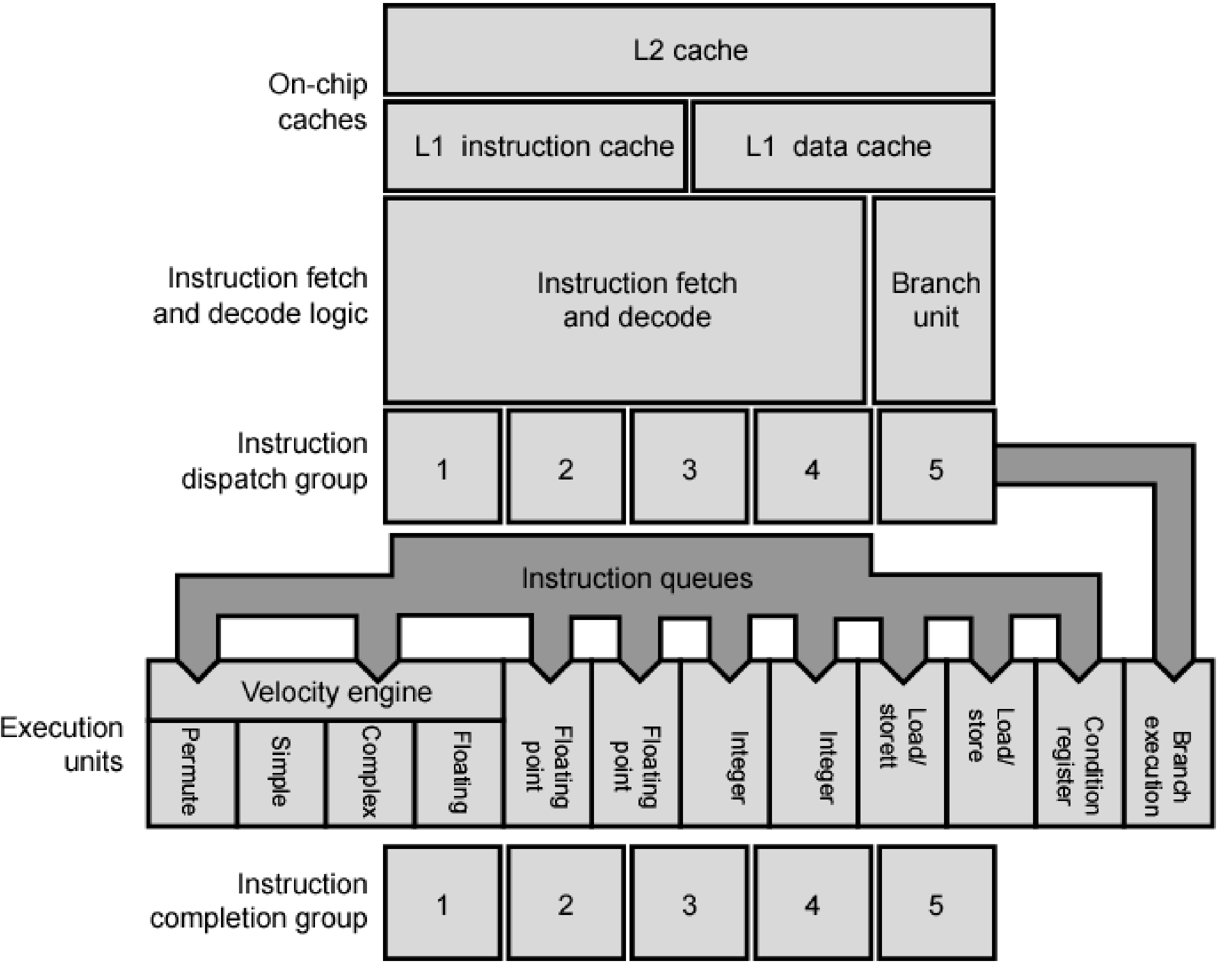
Pentium 4 Block Diagram



PowerPC Cache Organization

- 601 – single 32kb 8 way set associative
- 603 – 16kb (2 x 8kb)
- 604 – 32kb (2 X 16 KB)
- 620 – 64kb (2 X 32 KB)
- G3 & G4
  - 64kb L1 cache (2 X 32 KB)
  - 256k, 512k or 1M L2 cache
- G5
  - 32kB instruction cache
  - 64kB data cache

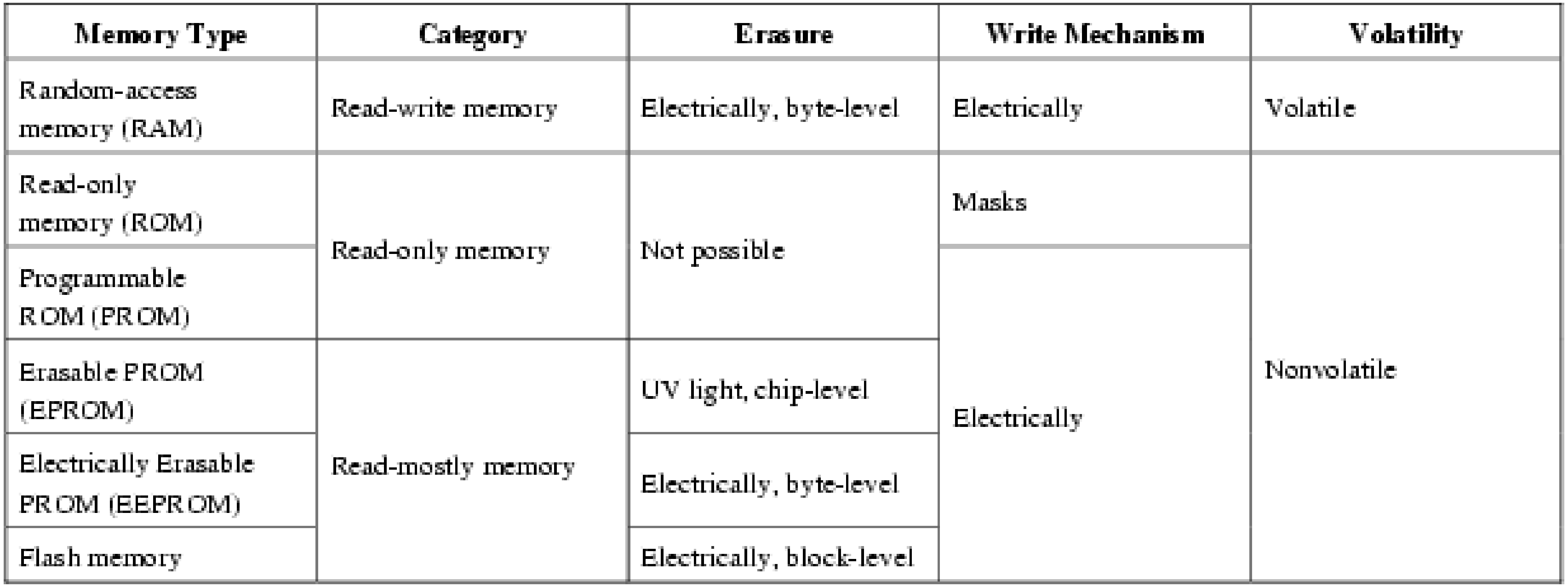
PowerPC G5 Block Diagram



- The core execution group include two arithmetic and logic units, which can execute in parallel, and two floating point units with their own registers and each with its own multiply, add, divide components.
- The instruction cache is read only, feeds into an instruction unit.
- (Will be discussed later in Chapter 14)

Internal Memory

Semiconductor Memory Types

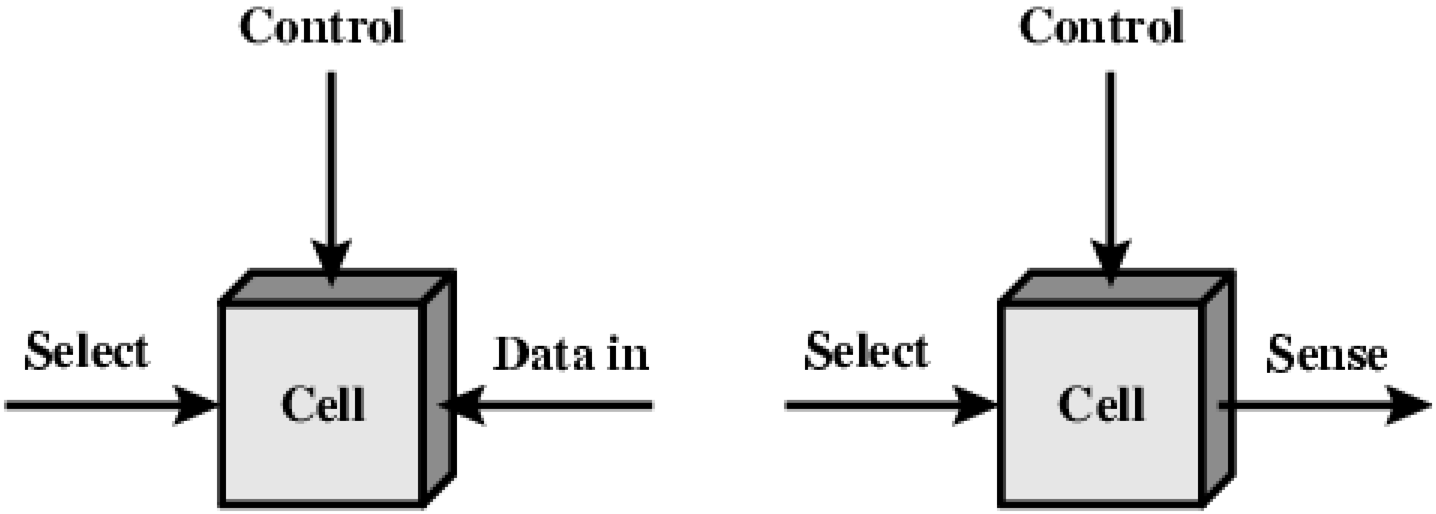


Memory Type	Category	Erasure	Write Mechanism	Volatility
Random-access memory (RAM)	Read-write memory	Electrically, byte-level	Electrically	Volatile
Read-only memory (ROM)	Read-only memory	Not possible	Masks	Nonvolatile
Programmable ROM (PROM)			Electrically	
Erasable PROM (EPROM)	Read-mostly memory	UV light, chip-level		
Electrically Erasable PROM (EEPROM)		Electrically, byte-level		
Flash memory		Electrically, block-level		

Semiconductor Memory

- RAM
  - **Misnamed** as all semiconductor memory is random access
  - **Read data** from the memory and to **write data** into the memory easily and rapidly(through **electrical signals**)
  - **Volatile** (if the power is interrupted, then the data are lost)
  - Can be used as temporary storage
  - Two traditional forms of RAM is Static or Dynamic (**SRAM** or **DRAM**)

Memory Cell Operation



(a) Write

(b) Read

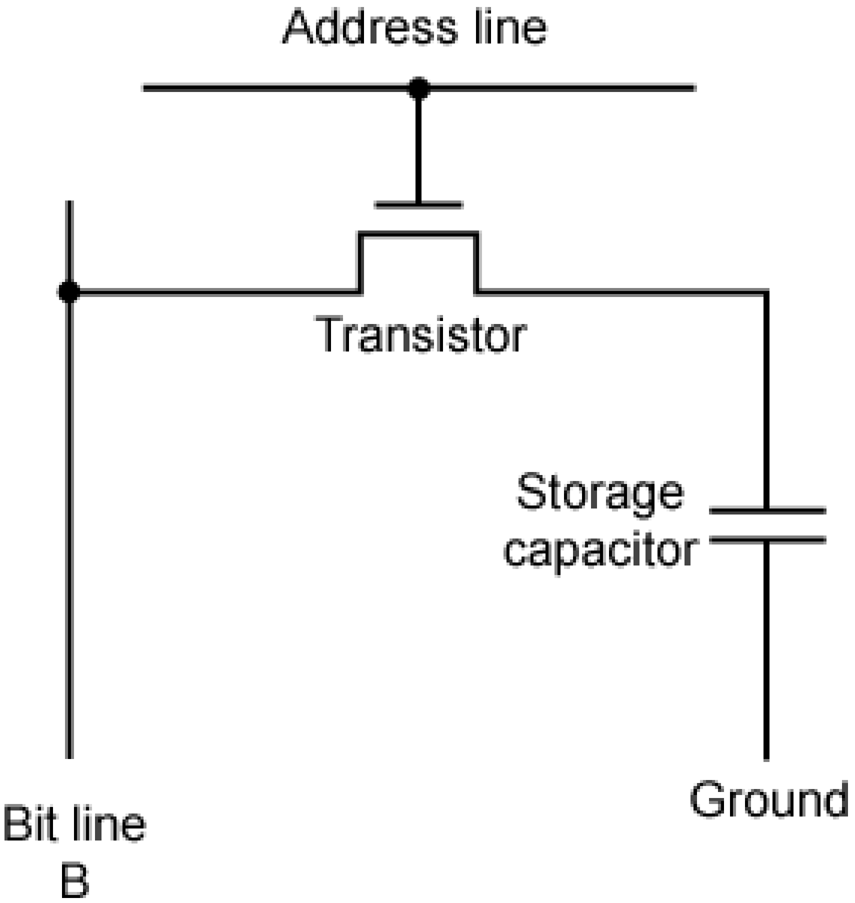
Three terminals carry an electrical signal:

- **Select terminal selects** a memory **cell for read and write** operation
- The **control terminal** indicates **read or write**
- For writing, the other terminal provides an electrical signal that **sets the state of the cell to 1 or 0.**
- For reading, the terminal is used for output of the cell’s state.

Dynamic RAM

- **Made with cells** that **store data as charge on capacitors** (the presence or absence of charge in a capacitor is interpreted **as** a binary **1 or 0**)
- **The term dynamic refers to** – the **tendency of the stored charge to leak away**
- **Capacitors have a natural tendency to discharge**, dynamic **RAMs need refreshing to maintain data storage**
- **Simpler construction**
- **Smaller per bit**
- **Less expensive**
- **Need refresh circuits**
- **Slower**
- **Main memory**
- Essentially **analogue** (although store a single bit **0 or 1**)
  - The capacitor can store any charge value within a range**; a threshold value determines whether the charge is 0 or 1.**

Dynamic RAM Structure & Operation

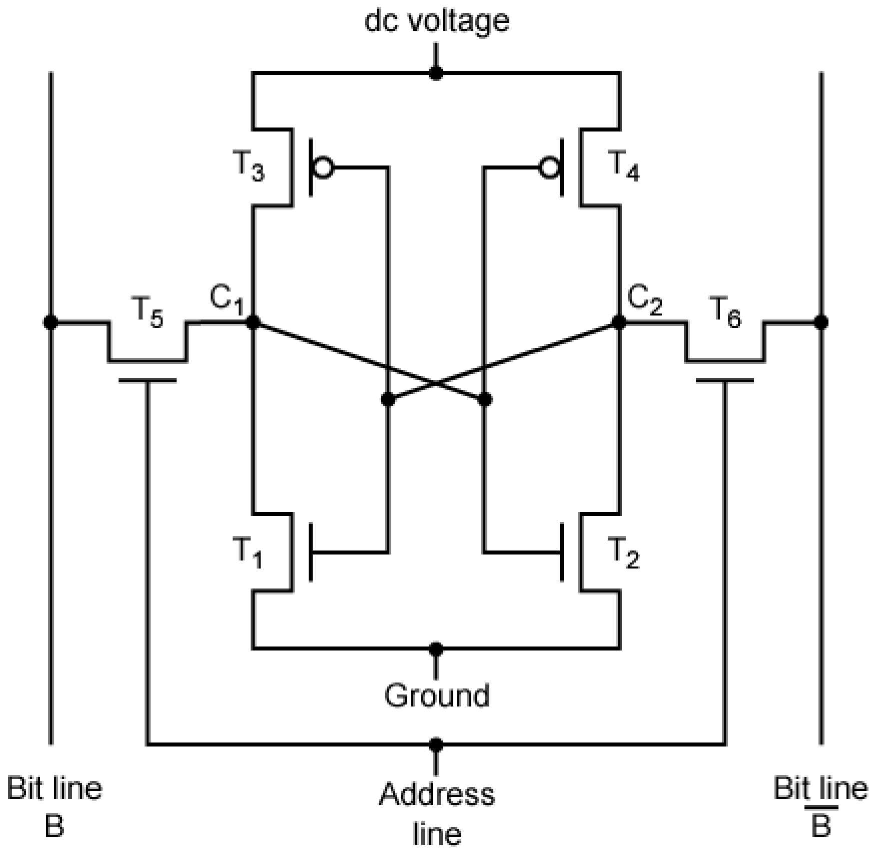


- Figure shows a DRAM structure of an individual **cell that stores one bit.**
- **Address line is activated** when **bit from this cell is to be read or written**
  - — Transistor acts a switch that is **closed** (allowing current to flow) if a voltage is applied to the address line and **open** (no current flows) if no voltage is present on the address line
- **Write**
  - — A voltage signal applied to bit line
    - – High voltage→1, low voltage→0
  - — Then signal is applied to address line
    - – Transfers charge to capacitor
- **Read**
  - — **Address line is selected**
    - – **transistor turns on**
  - — **Charge from capacitor is fed out via bit line and to sense amplifier**
  - — **The read out form the cell discharges the capacitor**, which must be **restored** to complete the operation

Static RAM

- **Bits stored as on/off switches**
- **Holds data as long as power is supplied to it**
- **No charges to leak**
- **No refreshing needed when powered**
- **More complex construction**
- **Larger per bit**
- **More expensive**
- **Does not need refresh circuits**
- **Faster**
- **Cache**
- Digital device using the same logic elements in the processor
  - Binary values are stored using **flip-flop logic gate** configuration.

Stating RAM Structure & Operation



- For a SRAM individual cell, **four transistors (T_1 ,T_2 ,T_3 ,T_4) are cross connected.**
- This arrangement gives stable logic state
- **State 1**
  - C_1 high, C_2 low
  - T_1 T_4 off, T_2 T_3 on
- **State 0**
  - C_2 high, C_1 low
  - T_2 T_3 off, T_1 T_4 on
- SRAM **address line is used to open or close a switch.**
- **Address line controls two transistors T_5 and T_6**
- When a signal is applied to this line, the two transistors are switch on, allowing read and write operation.
- **Write** – **apply the bit value to B & compliment to B**; this forces four transistors **(T_1 ,T_2 ,T_3 ,T_4) into proper state.**
- Read – **bit value is read from line B**

SRAM v DRAM

- Both volatile
  - Power needed to preserve data
- **Dynamic cell**
  - **Simpler, smaller** than a static memory cell
  - **More dense** (smaller cells=more cells per unit area)
  - **Less expensive**
  - **Needs refresh circuitry**
  - Tends to be favoured for **large memory** requirements
- **Static**
  - Faster
  - Cache

Read Only Memory (ROM)

- **Permanent storage** (cannot be changed)
  - **Nonvolatile** (no power source is required to maintain the bit values in memory)
  - While **it is possible to read a ROM**, **it is not possible to write** new data into it

➢ Important Application

- **Microprogramming (see later)**
- Library subroutines
- **Systems programs (BIOS)**
- Function tables

➢ Advantage of ROM

- **Data or program is permanently in main memory** and **need never be loaded from a secondary storage** device.

Types of ROM

- **Programmable read-only memory (PROM)**
  - Semiconductor memory whose contents may be set only once.
  - The writing process is performed electrically and performed by the user at a time of original chip fabrication
  - Needs special equipment to program
- Read "mostly" – read operations are far more frequent than write operations but for which non-volatile storage is required
  - **Erasable Programmable (EPROM)**
    - Read and write electrically, as with PROM
    - Before writing, all storage cells must be erased by ultraviolet (UV) radiation. [20 minutes take to erase]. After erasing you can write new program or data
    - EPROM is more expensive than PROM but it has advantage of multiple update capability.
  - **Electrically Erasable (EEPROM)**
    - Can be written into anytime without erasing whole contents; only byte or bytes addressed are updated
    - Takes much longer to write than read
    - EEPROM has advantage of non-volatility with flexibility of being updatable
    - More expensive than EPROM and also less dense (supporting fewer bits per cheap)
  - **Flash memory**
    - Erase whole memory or a section (block) electrically within one or a few seconds (in a single action or "**flash**"), which is much faster than EPROM

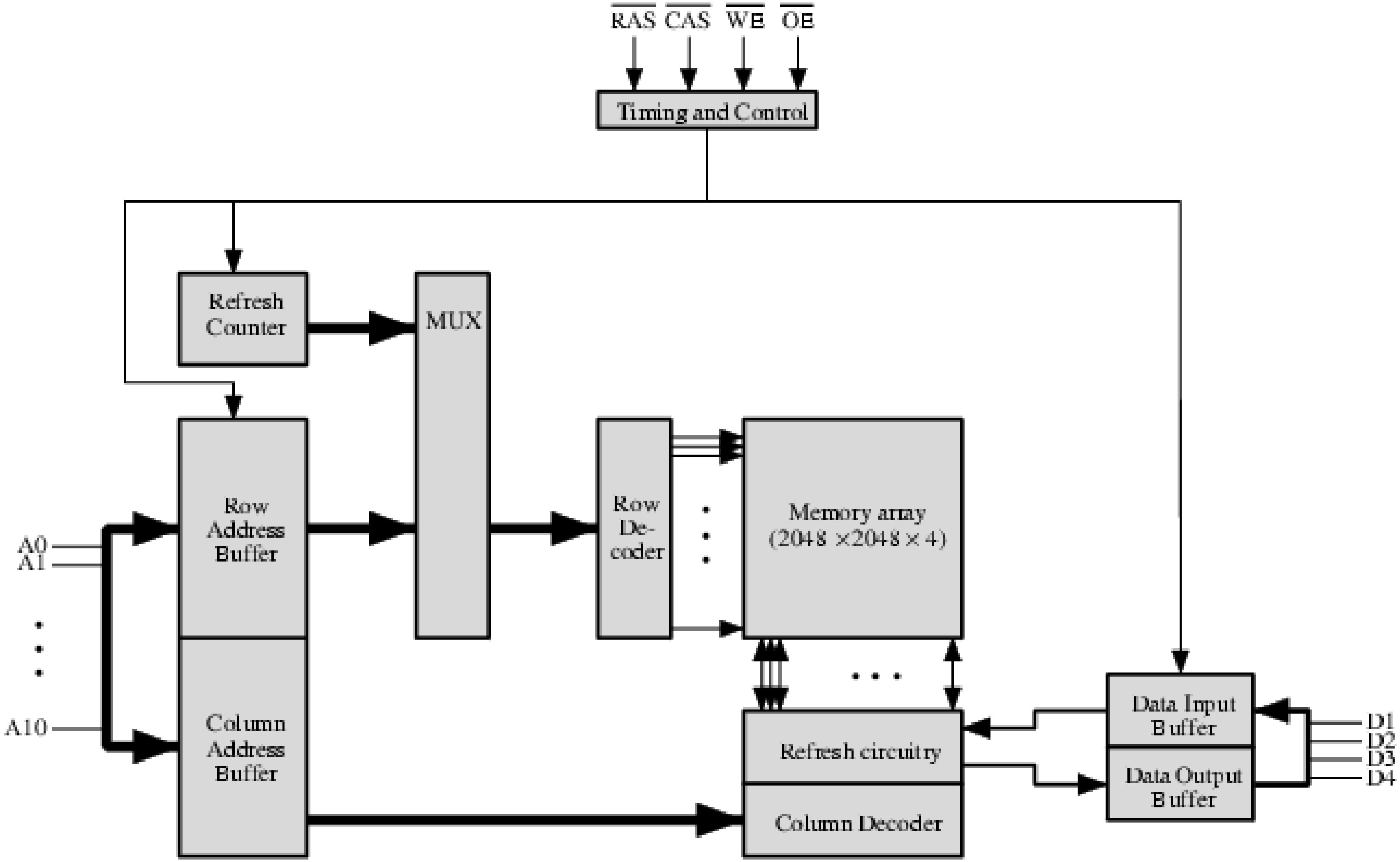
Organisation in detail

- A **16-Mbit (4Mx4) chip can be organised as 1M of 16 bit** words
- **4 bits are read and written at a time** (**four data lines are used for input and output of 4 bit**s to and **from a data buffer**).
- A **16Mbit chip can be organised as a 2048 x 2048 x 4bit** array
- **11 address lines are needed to select one of 2048 rows**. These 11 lines are fed into row decoder, which has 11 lines of input and 2048 lines of output. The logic of the **decoder activates one of the 2048 outputs depending on the bit pattern on the 11 input lines (2^{11}=2048)**
- **Additional 11 address lines select one of 2048 columns** of 4 bits per columns.
- On input (**write**), **the bit line is activated for a 1 or 0** according to the value of corresponding **data line**.
- On output (**read**), the value of each bit line is passed through a sense amplifier and **presented to the data lines**.
- The **row line selects which row of cells** is used for **reading and writing**
- Reduces number of address pins
  - **Multiplex row address and column address (first, 11 address signals are passed to define the row address** of the array, then **other 11 are presented for column address**)
  - Signals are accompanied by **RAS, CAS,WE, and OE.**
  - Adding one more pin doubles the number of rows and columns, so the size of memory grows by a factor of 4

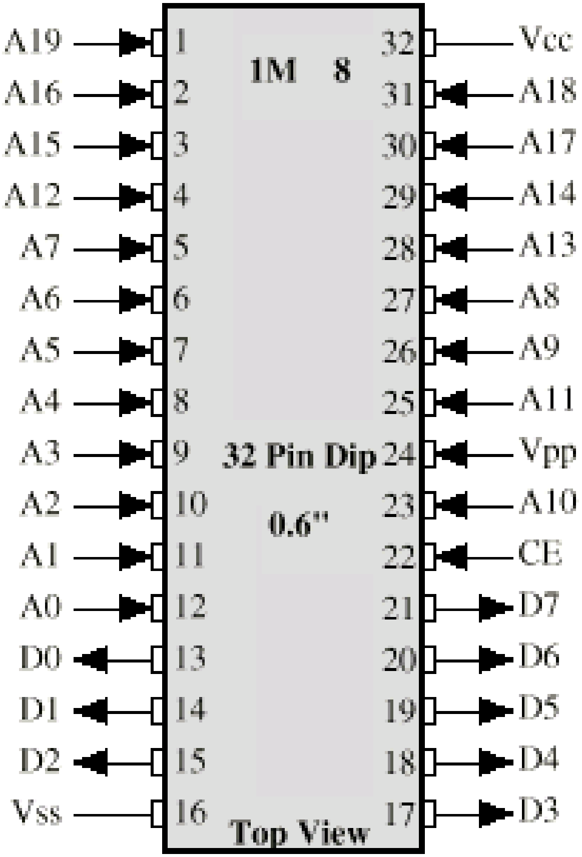
Refreshing

- **All DRAM requires refresh operation**
- For refreshing, to disable the DRAM chip while all data cells are refreshed.
- The refresh counter steps through all row values.
- For each row, output lines from the refresh counter are supplied to the row decoder and RAS line is activated
- **The data are read out and written back into the same location; this causes each cell in the raw to be refreshed**

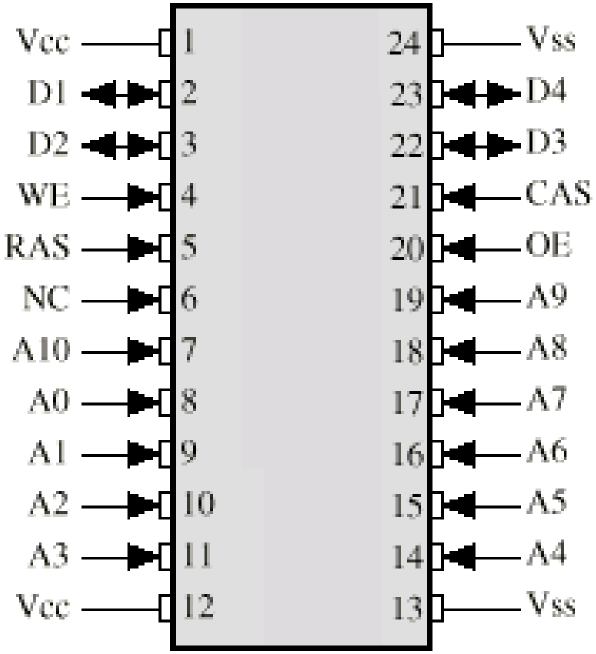
Typical 16 Mb DRAM (4M x 4)



Packaging



(a) 8 Mbit EPROM

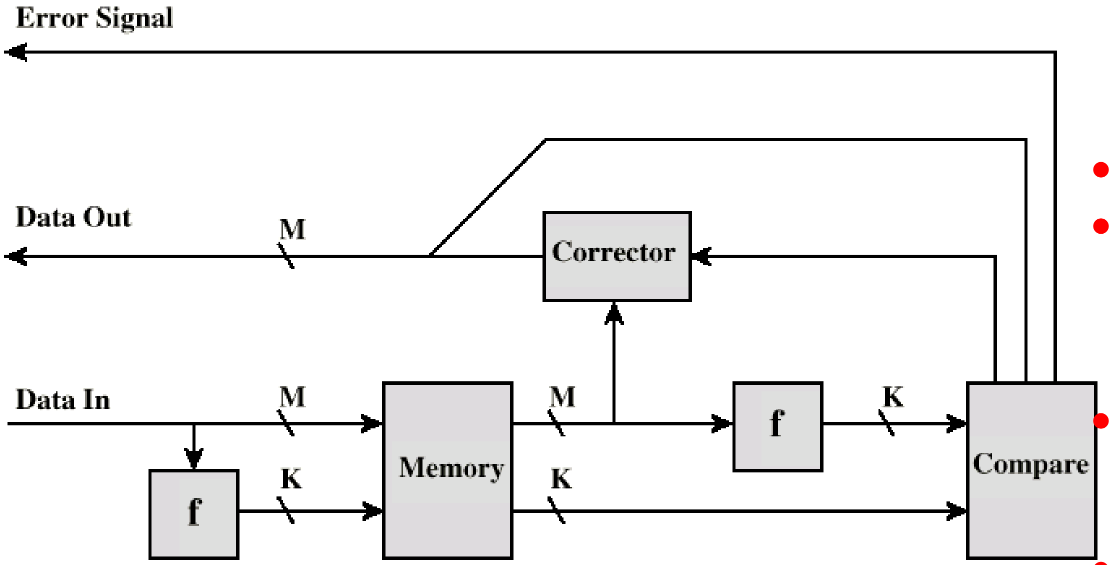


(b) 16 Mbit DRAM

Error Correction

- Hard Failure
  - Permanent defect
- Soft Error
  - Random, non-destructive
  - No permanent damage to memory
- Detected using Hamming error correcting code

Error Correcting Code Function



- When data are to be read into (stored) the memory, a function *f* is formed on the data to produce a code.
- Both code and data are stored
- Thus, if *M*-bit word of data is to be stored, and the code is of length *K* bits; actual size of the stored word is *M*+*K* bits
- When stored word is read out, the code is used to detect and possibly correct errors.
- A new set of *K* code bits is generated from *M* data bits and compared with the fetched code bits.
- This comparison yields three results:

  - No errors are detected and fetched data bits are sent out.

- An error is detected and possible to correct error. The data bits plus error correction bits are fed into a corrector, which produces M bits to be sent out.

- An error is detected, but it is not possible to correct it. This condition is reported.

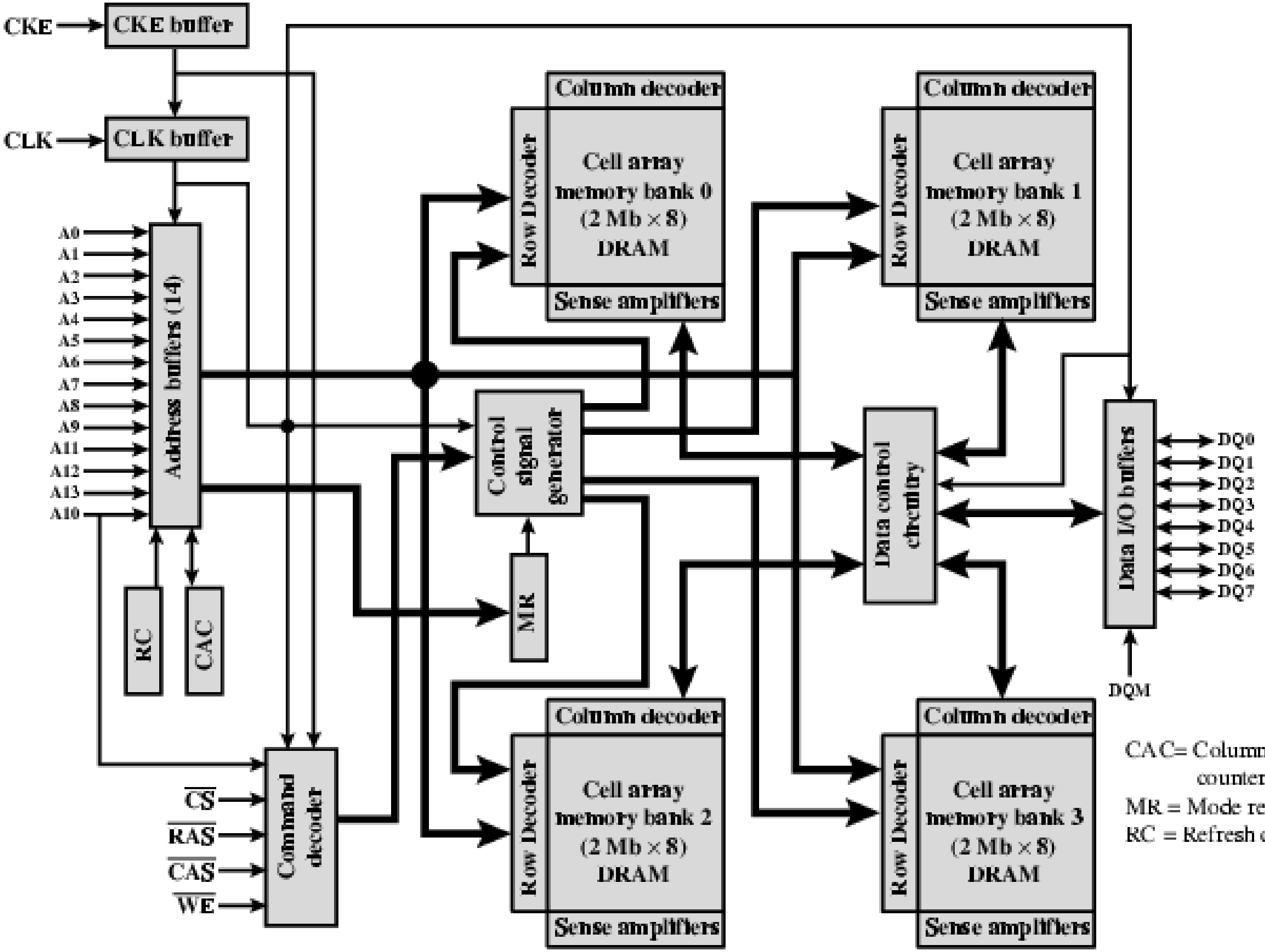
Advanced DRAM Organization

- **Basic DRAM same since first RAM chips**
- **Enhanced DRAM**
  - —**Contains small SRAM as well**
  - —**SRAM holds last line read (c.f. Cache!)**
- **Cache DRAM**
  - —**Larger SRAM** component
  - —**Use as cache or serial buffer**

Synchronous DRAM (SDRAM)

- Access is synchronized with an external clock
- Address is presented to RAM
- RAM finds data (CPU waits in conventional DRAM)
- Since SDRAM moves data in time with system clock, CPU knows when data will be ready
- CPU does not have to wait, it can do something else
- In burst mode, a series of data bits can be clocked out rapidly (on chip parallelism) after the first bit has been accessed.
- DDR-SDRAM sends data twice per clock cycle (leading & trailing edge)

SDRAM



CAC= Column address counter
MR = Mode register
RC = Refresh counter

DDR SDRAM

- **SDRAM can only send data once per clock**
- **Double-data-rate SDRAM can send data twice per clock cycle**
  - Rising edge and falling edge

Cache DRAM

- Mitsubishi
- Integrates small SRAM cache (16 kb) onto generic DRAM chip
- Used as true cache
  - 64-bit lines
  - Effective for ordinary random access
- To support serial access of block of data
  - E.g. refresh bit-mapped screen
    - CDRAM can prefetch data from DRAM into SRAM buffer
    - Subsequent accesses solely to SRAM