Memory System Design

Geeth Karunarathne

Introduction

Real world issues of designing memory:

- cost
- speed
- size
- power consumption
- volatility
- etc.

The CPU–Main Memory Interface

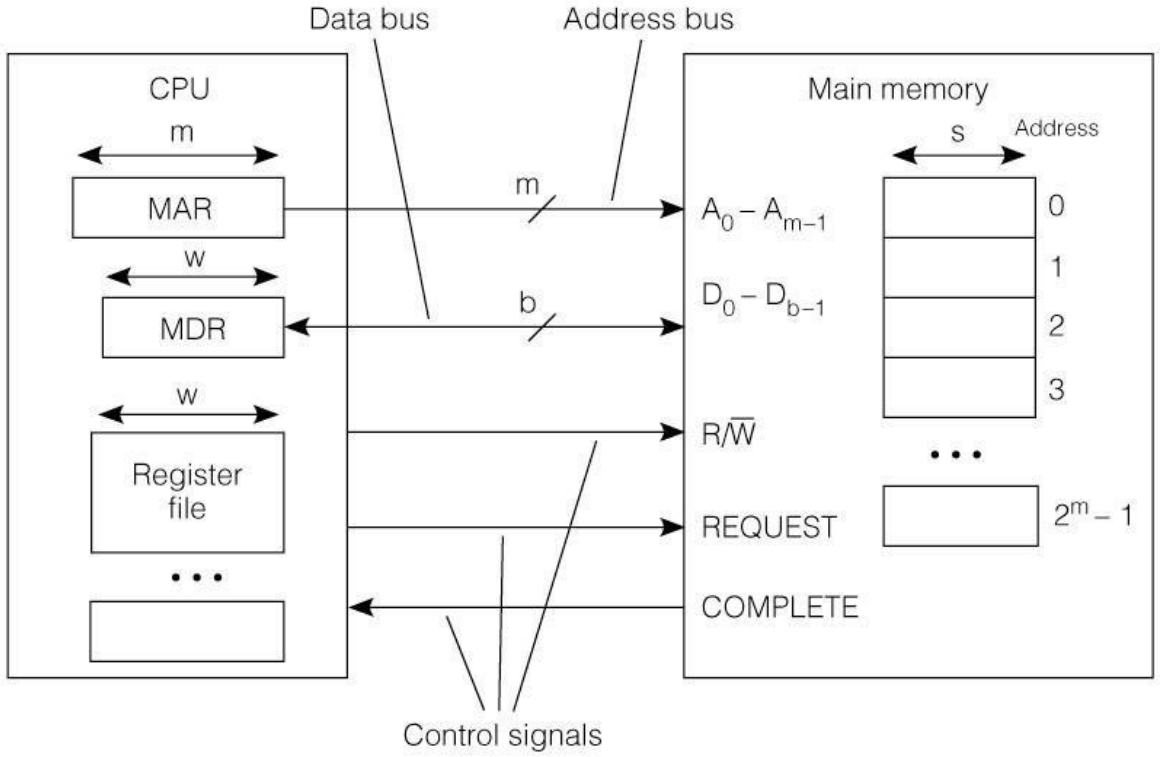
Sequence of events:

Read:

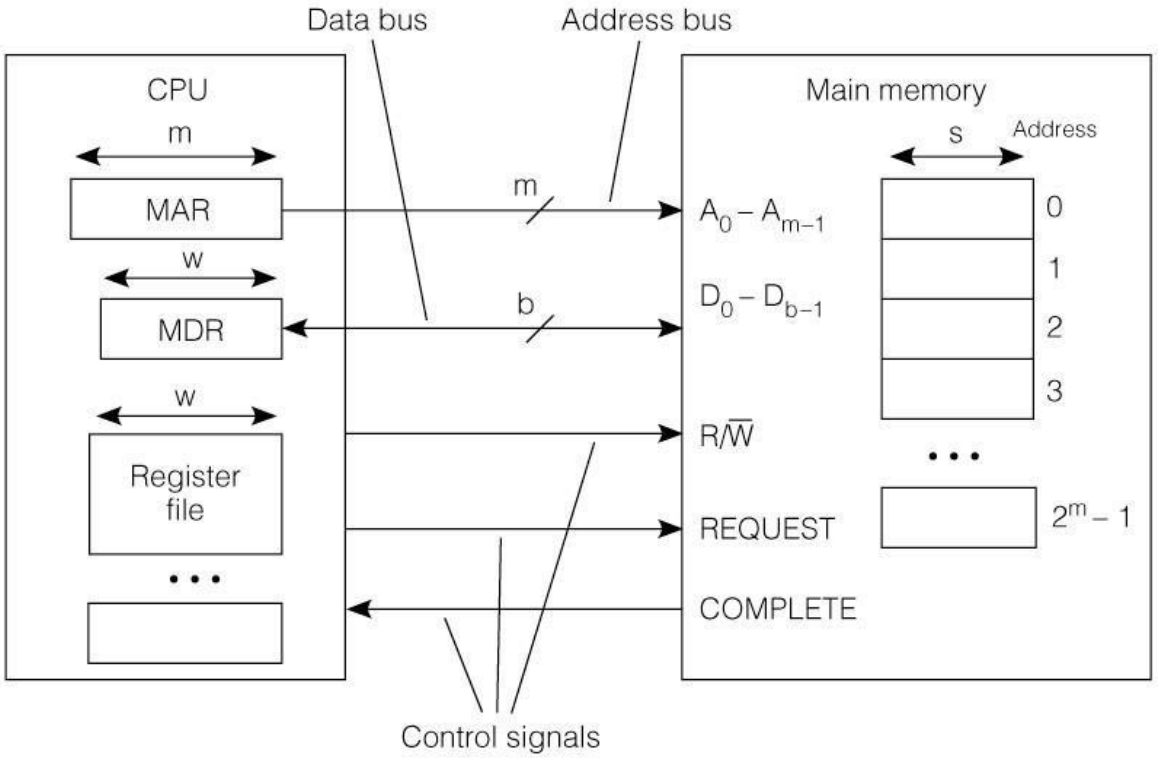
1. CPU loads MAR, issues Read, and REQUEST
2. Main Memory transmits words to MDR
3. Main Memory asserts COMPLETE.

Write:

1. CPU loads MAR and MDR, asserts Write, and REQUEST
2. Value in MDR is written into address in MAR.
3. Main Memory asserts COMPLETE.



The CPU–Main Memory Interface cont..



Additional points:

- if b<w, Main Memory must make w/b b-bit transfers.
- some CPUs allow reading and writing of word sizes <w.
  Example: Intel 8088: m=20, w=16,s=b=8.
  8- and 16-bit values can be read and written
- If memory is sufficiently fast, or if its response is predictable, then COMPLETE may be omitted.
- Some systems use separate R and W lines, and omit REQUEST

Some Memory Properties

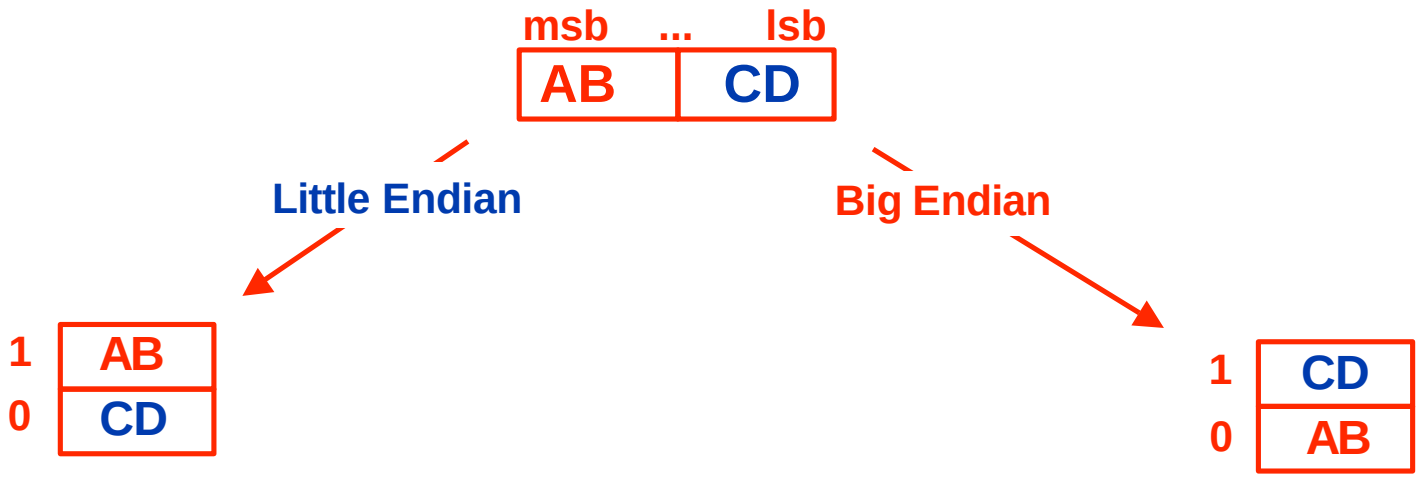
Symbol	Definition	Intel 8088	Intel 8086	IBM/Moto. 601
w	CPU Word Size	16bits	16bits	64 bits
m	Bits in a logical memory address	20 bits	20 bits	32 bits
s	Bits in smallest addressable unit	8	8	8
b	Data Bus size	8	16	64
2^m	Memory wd capacity, s-sized wds	2^{20}	2^{20}	2^{32}
2^mxs	Memory bit capacity	2^{20}x8	2^{20}x8	2^{32}x8

Big-Endian and Little-Endian Storage

When data types having a word size larger than the smallest addressable unit are stored in memory the question arises,

> "Is the least significant part of the word stored at the lowest address (***little Endian, little end first***) or–
>
> is the most significant part of the word stored at the lowest address (***big Endian, big end first***)"?

Example: The hexadecimal 16-bit number ABCDH, stored at address 0:



Memory Performance Parameters

Symbol	Definition	Units	Meaning
t_a	Access time	time	Time to access a memory word
t_c	Cycle time	time	Time from start of access to start of next access
k	Block size	words	Number of words per block
b	Bandwidth	words/time	Word transmission rate
t_l	Latency	time	Time to access first word of a sequence of words
$t_{bl} = t_l + k/b$	Block access time	time	Time to access an entire block of words

(Information is often stored and moved in blocks at the cache and disk level.)

The Memory Hierarchy

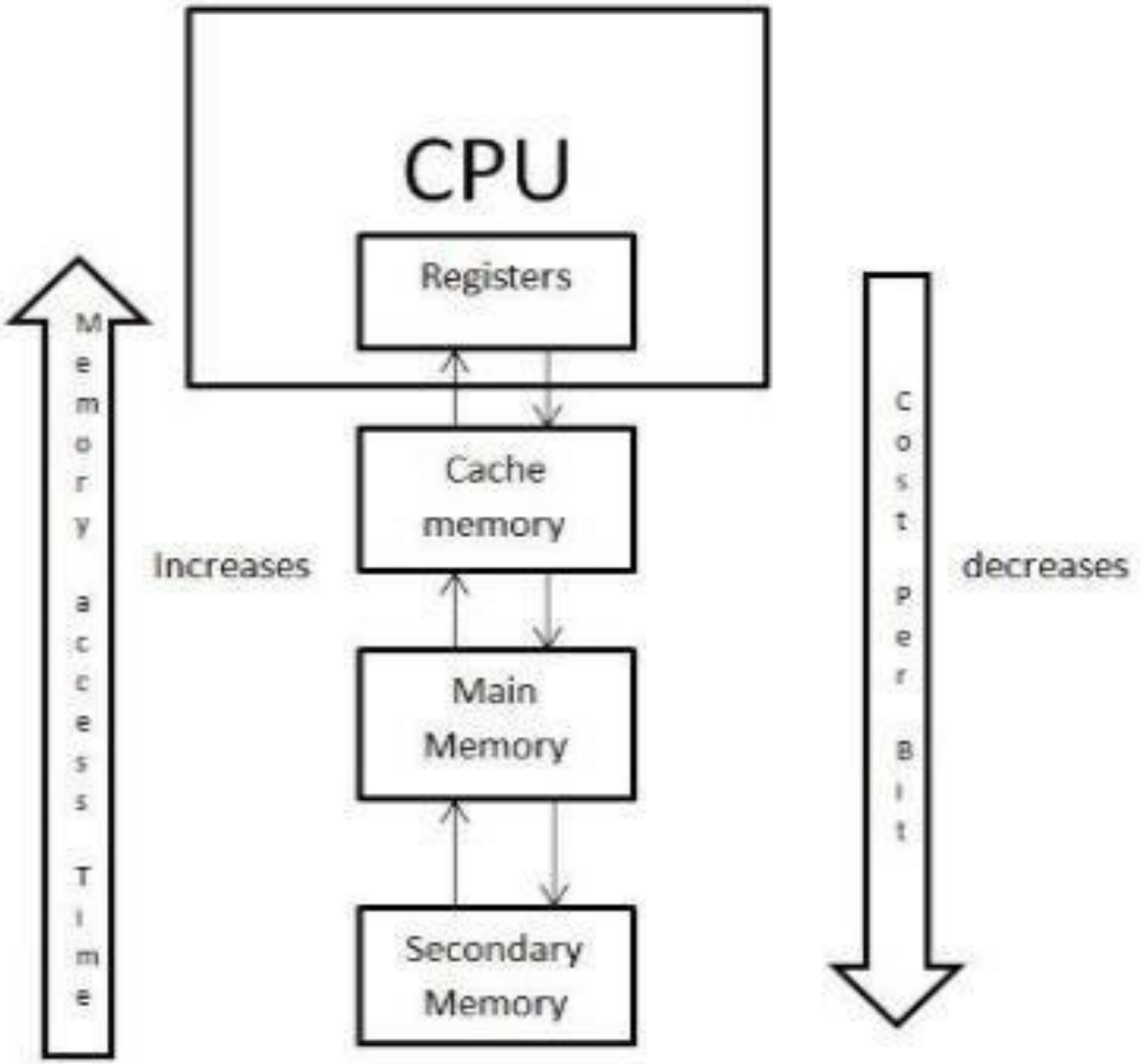


Figure 1

Semiconductor Memory Types

- Random Access Memory (RAM)
- Read Only Memory (ROM)
- Programmable Read Only Memory (PROM)
- Eraseable Programmable Read Only Memory (EPROM)
- Electronically Eraseable Programmable Read Only Memory (EEPROM)
- Flash Memory

Random Access Memory

- RAM allows reading and writing (electrically) of data at the byte level
- Two types
  - Static RAM
  - Dynamic RAM
- Volatile

Read Only Memory (ROM)

- Sometimes can be erased for reprogramming, but might have odd requirements such as UV light or erasure only at the block level
- Sometimes require special device to program, i.e., processor can only read, not write
- Types
  - EPROM
  - EEPROM
  - Custom Masked ROM
  - OTPROM
  - FLASH

ROM Uses

- Permanent storage – nonvolatile
- Microprogramming
- Library subroutines
- Systems programs (BIOS)
- Function tables
- Embedded system code

EPROM

Written to only with a programmer

Erased with ultraviolet light

Pros

- non-volatile storage without battery
- can write to it, but only with aid of programmer

Cons

- programmer requirements
- Expensive
- locations must be erased before writing

EEPROM

Written to with either programmer or the processor (electrically)
Erased with either a programmer or the processor (byte-by-byte electrically)

Pros

- non-volatile memory without batteries
- programmable a single-location at a time

Cons

- Expensive
- only smaller sizes available
- extremely slow write times (10 mS vs. 100 to 200 nS)

Custom masked ROM

You send the ROM manufacturer your data and they mask it directly to the ROM

Use only when you are selling large volume of a single product

Pros

- becomes cheaper to use for approximately more than 2000 parts
- components come from chip manufacturer already programmed and tested taking out a manufacturing step

Cons

- costs several thousand dollars for custom mask
- software changes are costly
- cannot be reprogrammed

OTPROM

- Uses fuses that are burned to disconnect a logic 1 and turn it to a logic 0
- Written to by you using a programmer similar to EPROM

•Once it's written to, the data is in there forever

Pros

cheaper than EPROM due to cheaper packaging

more packaging options than EPROM due to less constraints like erasure window

standard "off-the-shelf" component

cheaper than Custom masked ROM up to about 10,000 devices

Cons

to reprogram, have to throw out the chip - Should only be used for stable design

FLASH

- These memories are basically EEPROMs except that erasure occurs at the block level in order to speed up the write process
- Non-volatile

This makes FLASH work like a fast, solid state hard drive

Pros

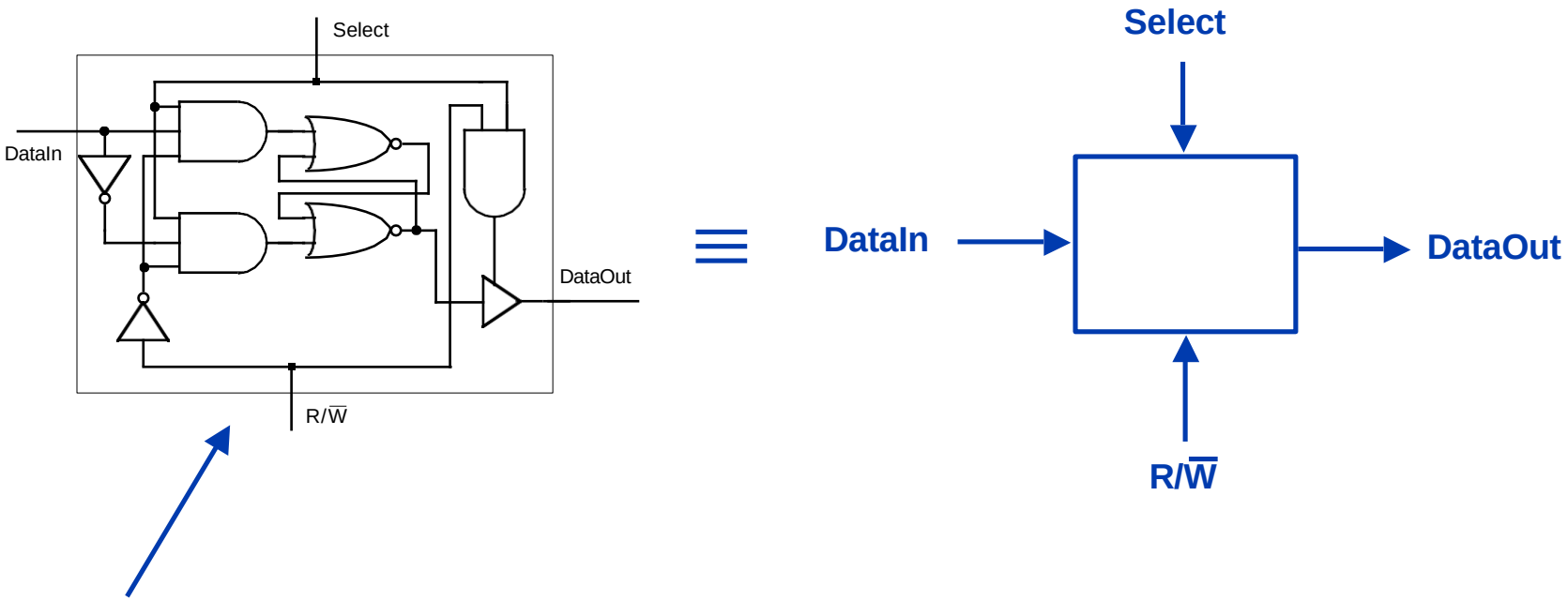
non-volatile
higher densities than both SRAM and DRAM

Cons

process of storing data is at a block level (and slower)
data cell must be erased before writing data to it

Memory Cells

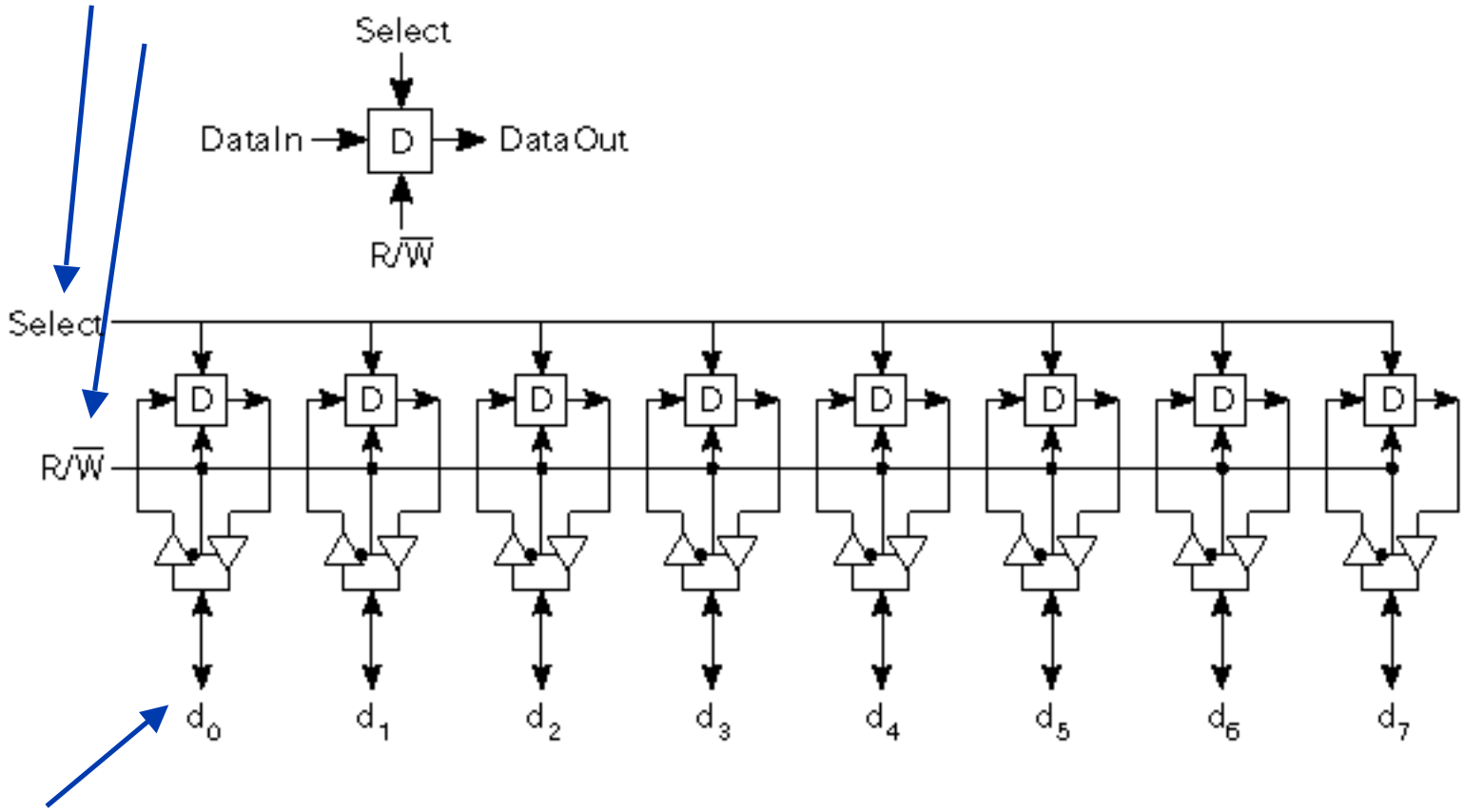
All RAM memory cells must provide these four functions: Select, DataIn, DataOut, and R/W



This “static” RAM cell is **unrealistic**

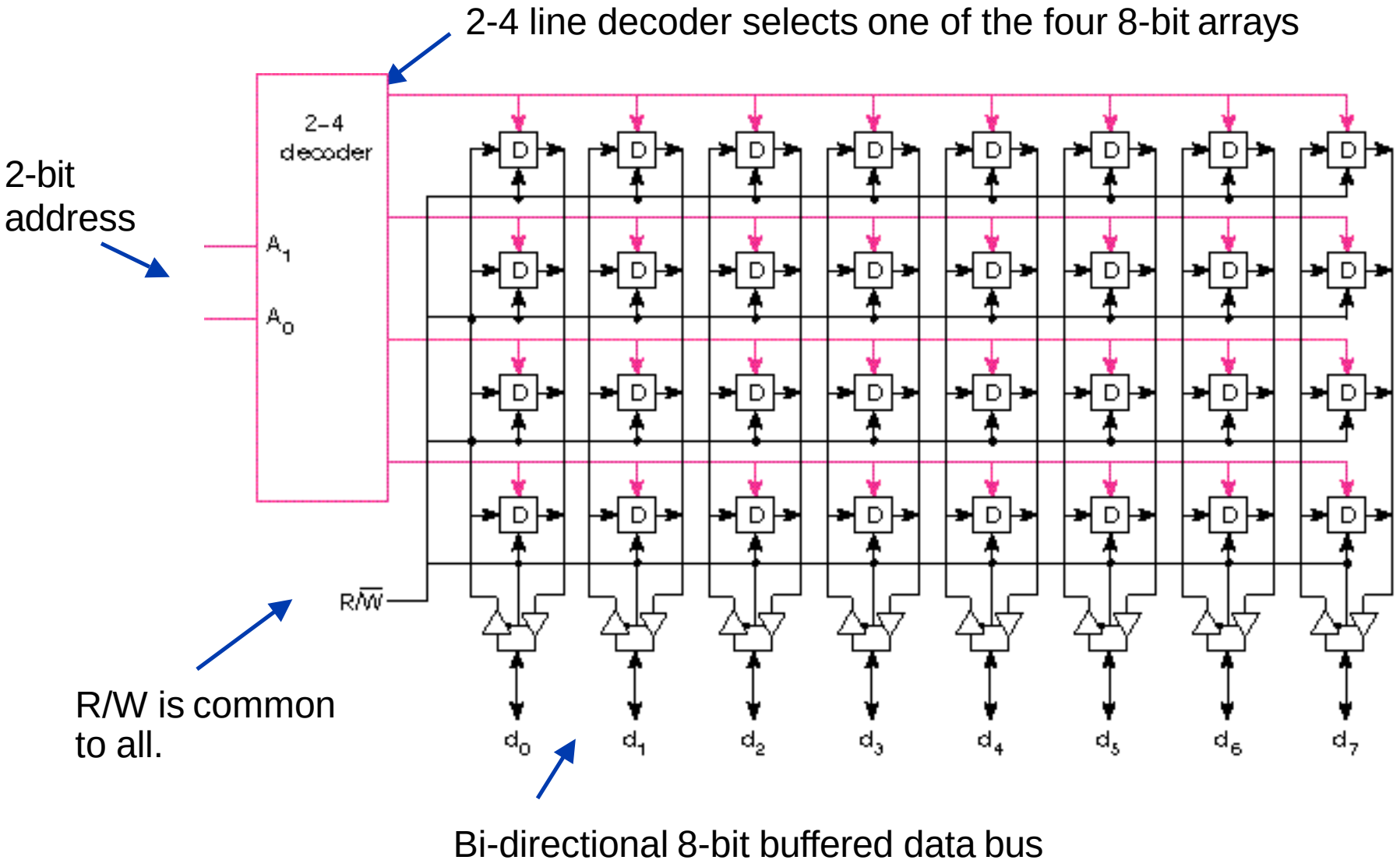
An 8-bit register as a 1D RAM array

The entire register is selected with one select line, and uses one R/W line



Data bus is bi-directional, and buffered

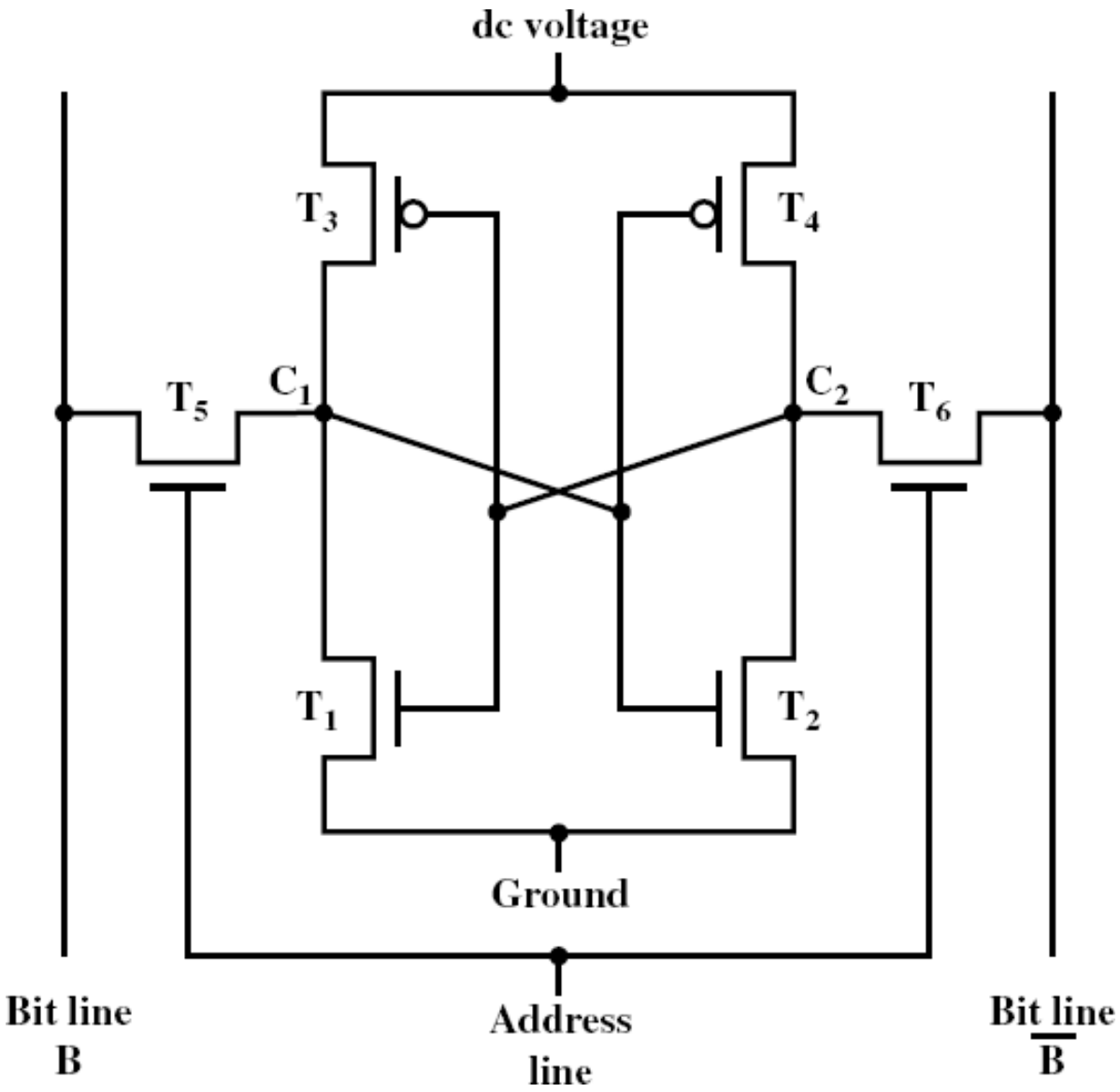
4x8 2D Memory Cell Array



Static RAM (SRAM)

- Essentially uses latches to store charge (transistor circuit)
- As long as power is present, transistors do not lose charge (no refresh)
- Very fast (no sense circuitry to drive nor charge depletion)
- Can be battery-backed – A small battery is piggy-backed to the RAM chip an allows data to remain even when power is removed (Not possible with DRAM)
- More complex construction
- Larger per bit
- More expensive
- Used for Cache RAM because of speed and no need for large volume or high density

SRAM Operation



SRAM Operation

Transistor arrangement gives stable logic state

State 1

- C1 high, C2 low
- T1 T4 off, T2 T3 on

State 0

- C2 high, C1 low
- T2 & T3 off, T1 & T4 on

Address line transistors

- T5 & T6 act as switches connecting cell

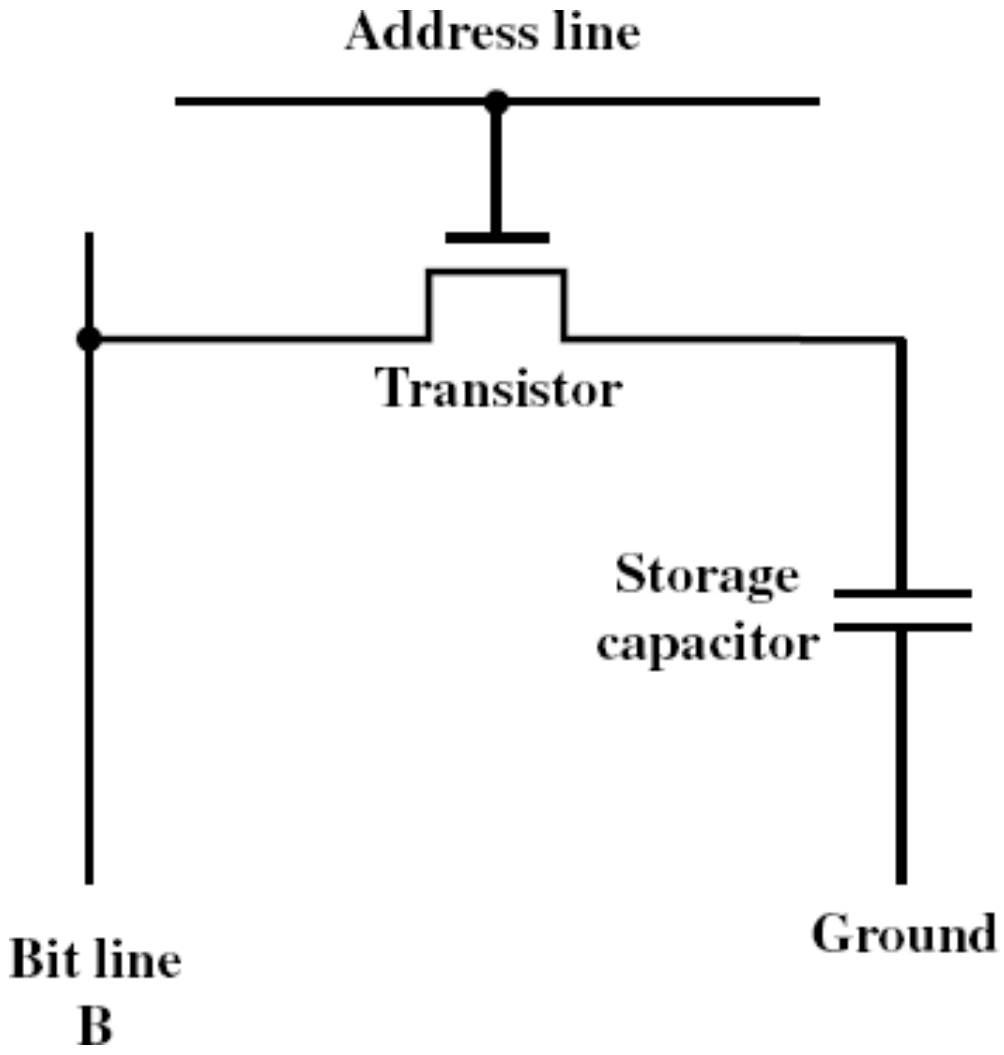
Write – apply value to B & compliment to B’

Read – value is on line B

Dynamic RAM (DRAM)

- Bits stored as charge in capacitors
- Simpler construction
- Smaller per bit
- Less expensive
- Slower than SRAM
- Typical application is main memory
- Essentially analogue -- level of charge determines value

DRAM Structure



DRAM Operation

Address line active when bit read or written

Logic ‘1’ closes transistor switch (i.e., current flows)

Write

- Voltage to bit line – High for 1 low for 0
- Signal address line – Controls transfer of charge to capacitor

Read

- Address line selected – transistor turns on
- Charge from capacitor fed via bit line to sense amplifier
- Compares with reference value to determine 0 or 1

SRAM vs. DRAM

Both volatile – Power needed to preserve data

DRAM

- Simpler to build, smaller
- More dense
- Less expensive
- Needs refresh
- Larger memory units

SRAM

- Faster
- Used for cache

Types of ROM

ROM Type	Cost	Programmability	Time to program	Time to erase
Mask pro-grammed	Very inexpensive	At the factory	Weeks (turn around)	N/A
PROM	Inexpensive	Once, by end user	Seconds	N/A
EPROM	Moderate	Many times	Seconds	20 minutes
Flash EPROM	Expensive	Many times	100 us.	1s, large block
EEPROM	Very expensive	Many times	100 us.	10 ms, byte

Cache

What is it?

A cache is a small amount of fast memory

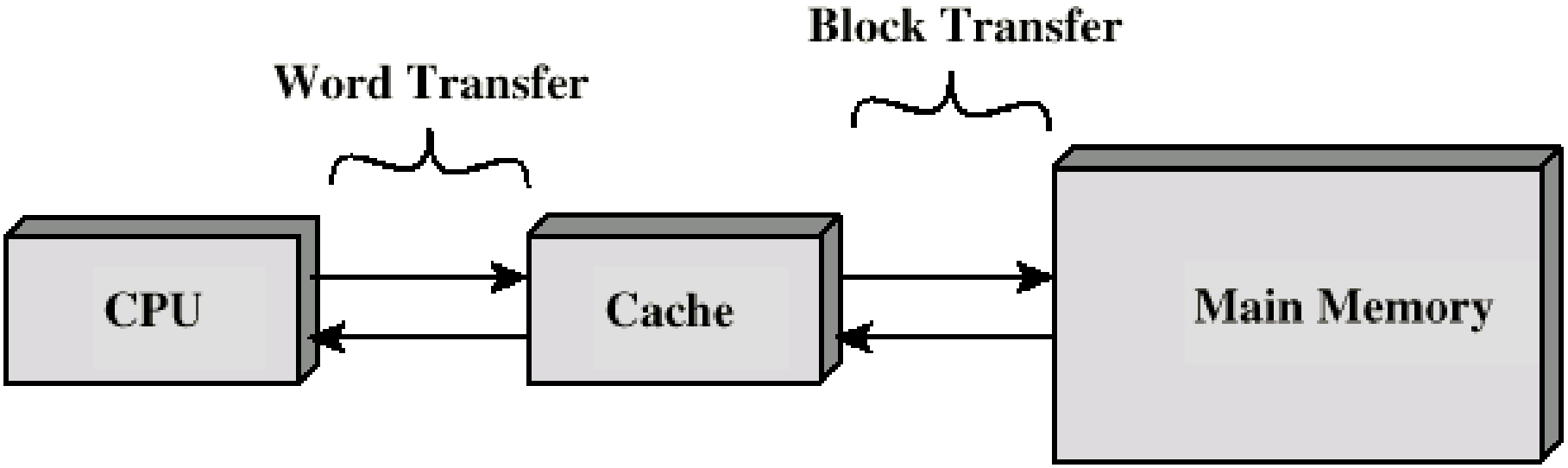
What makes small fast?

Simpler decoding logic

More expensive SRAM technology

Close proximity to processor – Cache sits between normal main memory and CPU or it may be located on CPU chip or module

Cache (continued)



Cache operation – overview

- CPU requests contents of memory location
- Check cache for this data
- If present, get from cache (fast)
- If not present, one of two things happens:
  - read required block from main memory to cache then deliver from cache to CPU (cache physically between CPU and bus)
  - read required block from main memory to cache and simultaneously deliver to CPU (CPU and cache both receive data from the same data bus buffer)

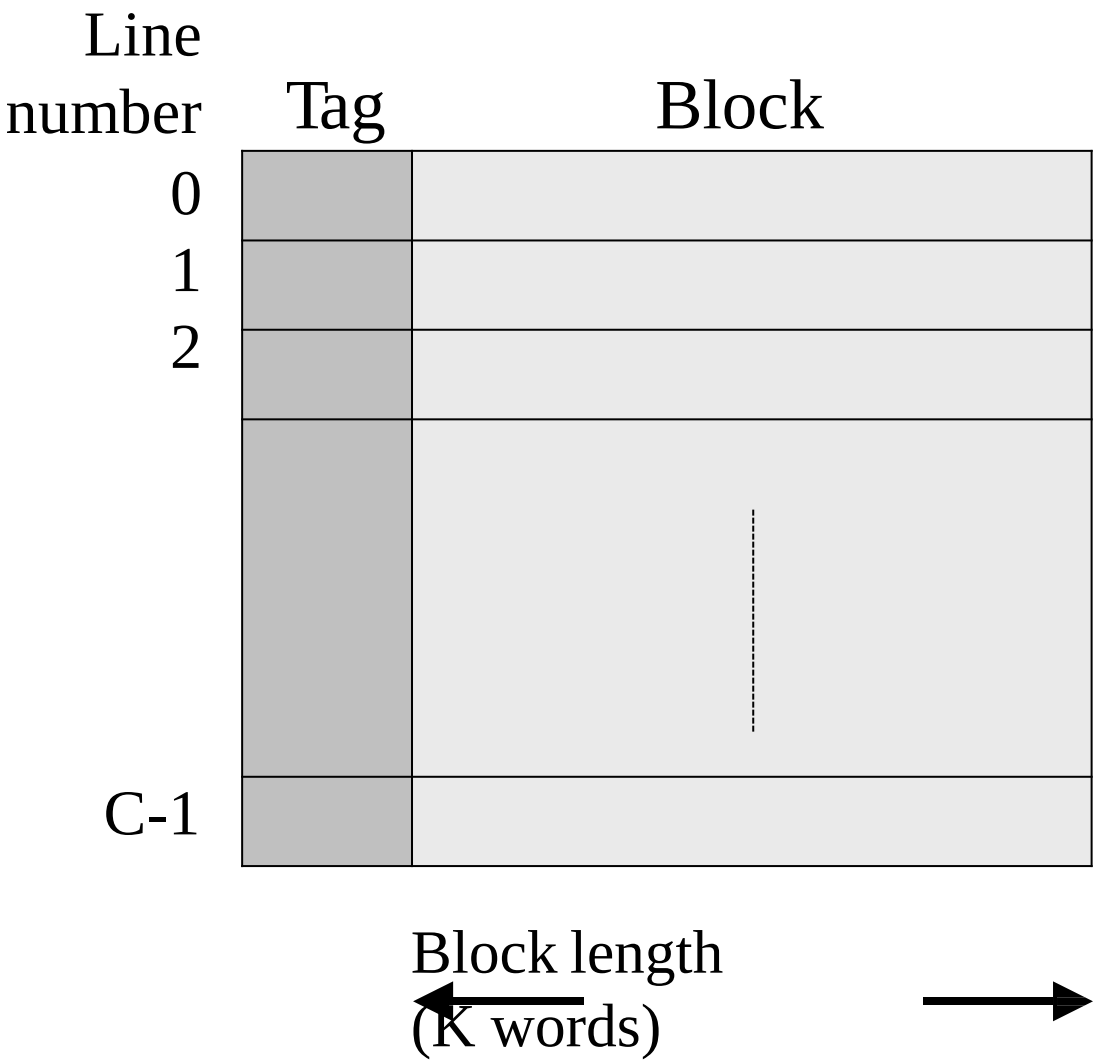
Principle of Locality

- Cache "misses" are unavoidable, i.e., every piece of data and code thing must be loaded at least once
- What does a processor do during a miss? It waits for the data to be loaded.
- Power consumption varies linearly with clock speed and the square of the voltage.
- Adjusting clock speed and voltage of processor has the potential to produce cubic (cubed root) power reductions.

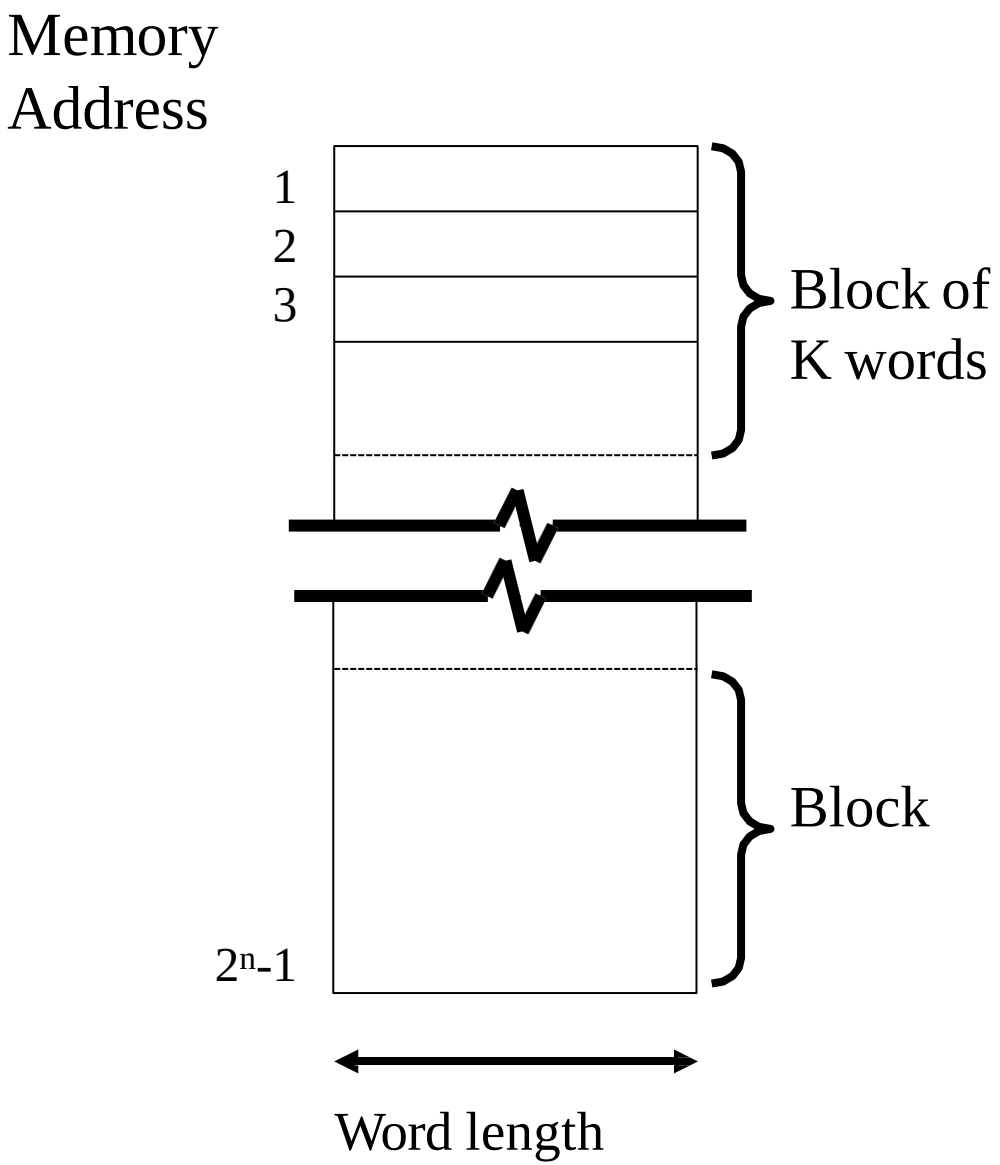
Cache Structure

- Cache includes tags to identify the address of the block of main memory contained in a line of the cache
- Each word in main memory has a unique n-bit address
- There are $M=2^n/K$ block of K words in main memory
- Cache contains C lines of K words each plus a tag uniquely identifying the block of K words

Cache Structure



Memory Divided into Blocks



Cache Design

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

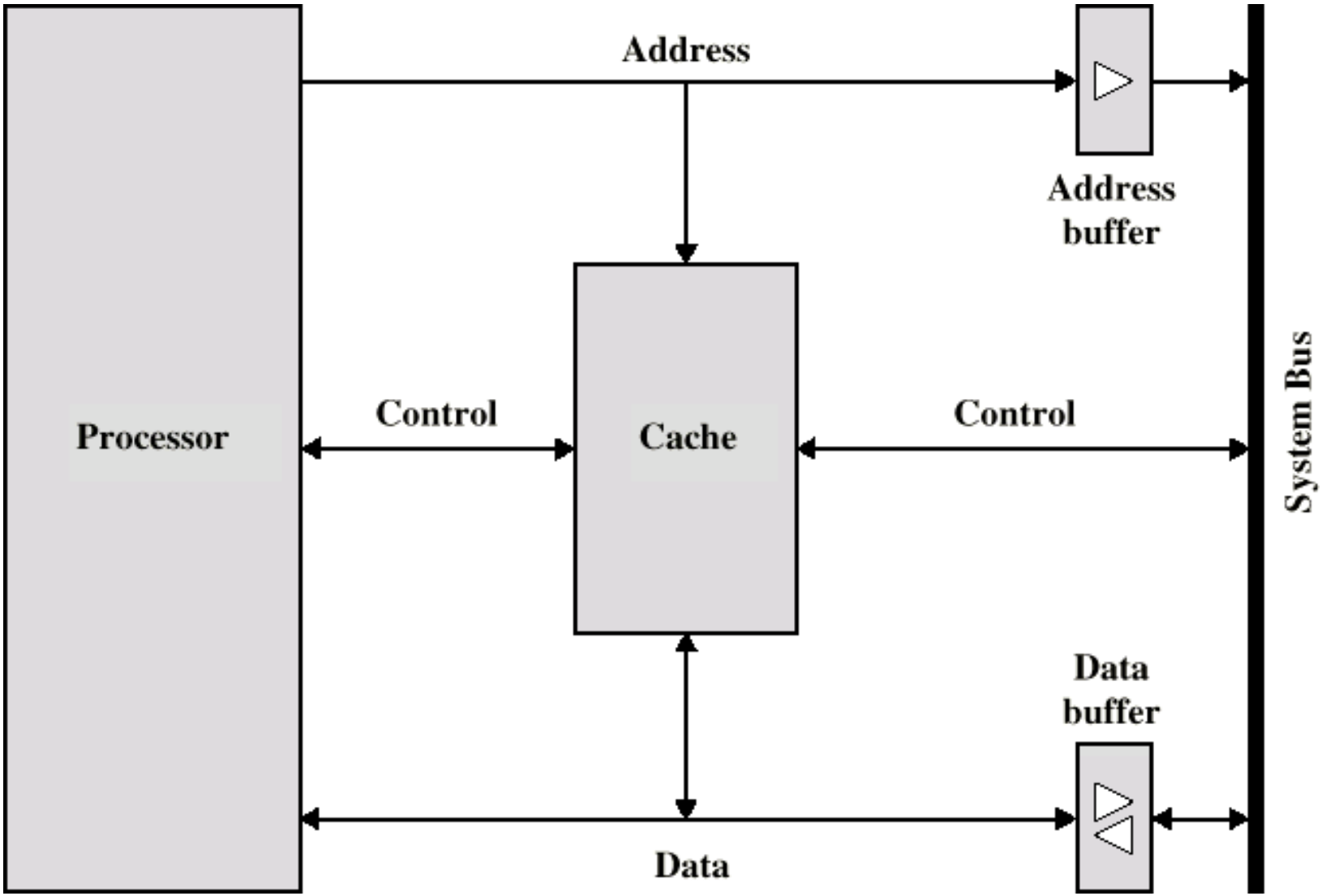
Cache size

Cost – More cache is expensive

Speed

- More cache is faster (up to a point)
- Larger decoding circuits slow up a cache
- Algorithm is needed for mapping main memory addresses to lines in the cache. This takes more time than just a direct RAM

Typical Cache Organization



Mapping Functions

- A mapping function is the method used to locate a memory address within a cache
- It is used when copying a block from main memory to the cache and it is used again when trying to retrieve data from the cache
- There are three kinds of mapping functions
  - Direct
  - Associative
  - Set Associative

Register File (RF)

- RF is an array of processor registers in CPU
- Highest level of the Memory Hierarchy
- Can be implemented using SRAM with multiple ports
- Two types of registers
  - User Visible Registers
  - Control and Status Registers

Types of Registers

- User Visible Registers
  - Address Register(AR)
  - Data Register(DR)
  - Temporary Register(TR)
  - General Purpose Registers (GPR)

- Control and Status Registers
  - Instruction Register(IR)
  - Program Counter(PC)
  - Memory Access Register (MAR)

Virtual Memory

- Virtual memory is a memory management technique where secondary memory can be used as a part of the main memory
- Virtual memory uses both hardware and software to compensate for physical memory shortages
- Temporarily transferring data from random access memory (RAM) to disk storage
- Mapping chunks of memory to disk files enables a computer to treat secondary memory as though it were main memory

Virtual Memory