Computer Organization and Architecture

Memory Unit-Introduction

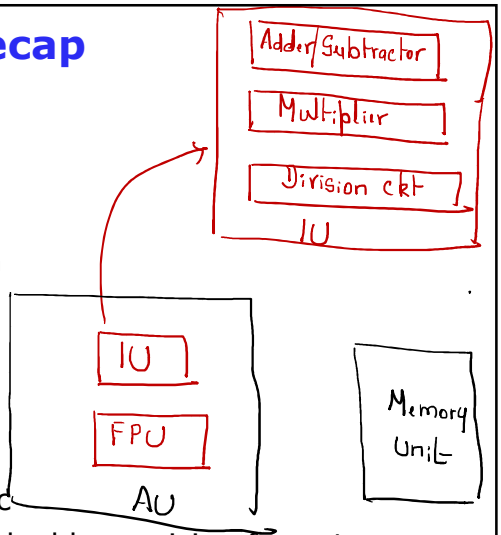
Veena Thenkanidiyoor
National Institute of Technology Goa

1

1

Recap

- Arithmetic unit design
- Integer arithmetic
  - Number representation
    - Signed
    - Unsigned
  - Adder/Subtractor
  - Multipliers
  - Division circuit
- Floating point Arithmetic
  - IEEE 754 single and double precision formats
  - Arithmetic operations



2

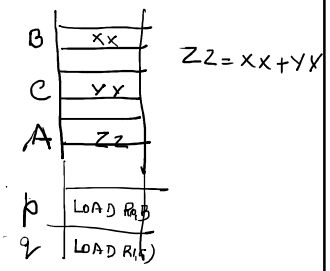
2

Computer Memory

- Program and data they operate on are held in the memory of the computer
  - Number & character operands, as well as instructions are stored in the memory of the computer

Assignment statement: **A = B + C**

```
LOAD   R0, B
LOAD   R1, C
ADD    R2, R0, R1
STORE  A, R2
```



- Execution speed of programs
  - Also dependent on the speed with which instructions and data can be transferred between the processor and memory

3

3

Computer Memory

- Desirable to have **large memory**
  - To execute large programs
  - To handle huge amount of data
- Ideally to have memory that is **fast, large, and inexpensive**
  - Unfortunately not possible to meet this requirement
  - Increased speed and size will increase the cost too
    - **Clever solutions to address this**

4

4

Computer Memory

- Maximum size
  - Determined by the addressing scheme
    - 16-bit computer --- generates 16-bit addresses
      - Addressing up to 2^{16}=64K memory locations

- Modern computers are byte addressable

1 bit → cell
1 Byte
1 word → Collection of bits Bytes

32 bit Add
Word Length = 32 bit (Word) Higher order 30 bits, Lower order 2 bits → byte.

- Memory is designed to store and retrieve data in word-length quantities
  - Word: number of bits actually stored or retrieved in one memory access

5

5

Byte Addressable Memory

Word Address	Byte Address			
0	3	2	1	0
1	7	6	5	4
2	11	10	9	8
	⋮			
$2^{k/4}-1$	2^k-4	2^k-3	2^k-2	2^k-1

Little Endian

Word Address	Byte Address			
0	0	1	2	3
1	4	5	6	7
2	8	9	10	11
	⋮			
$2^{k/4}-1$	2^k-1	2^k-2	2^k-3	2^k-4

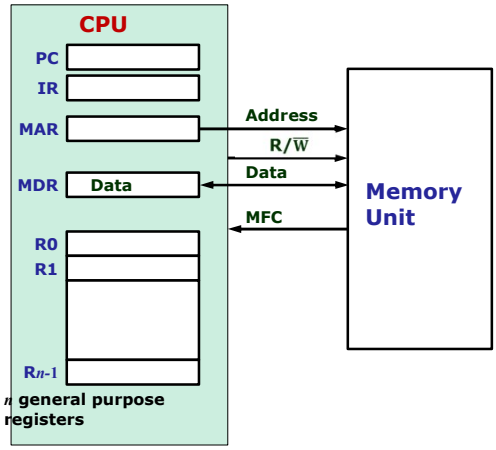
Big Endian

Example: Intel Processors **Motorola Processors**

6

6

Memory Read and Write Operation

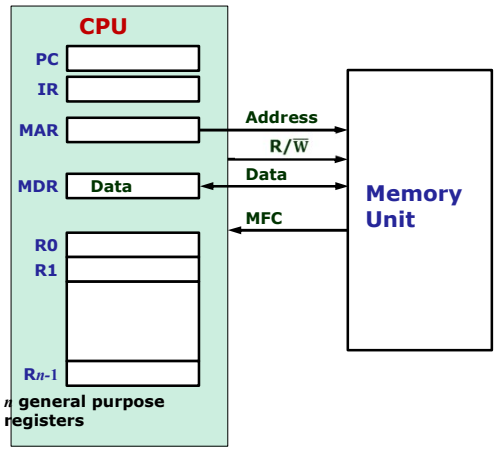


- **Read**
  - Processor loads the address of memory location into MAR
  - Set the R/$\overline{W}$ line to 1
  - Memory responds by placing the data from address location onto data line
  - Confirm the action by asserting MFC (memory function complete) signal
  - Upon receiving MFC signal, processor loads the data on data line into MDR

7

7

Memory Read and Write Operation



- **Write**
  - Processor loads the address of memory location into MAR
  - Processor loads data into MDR
  - Set the R/$\overline{W}$ line to 0 to indicate write operation
  - Processor places the data in MDR onto data line
  - Data on data line is written into memory location
  - Memory confirms the action by asserting MFC signal

8

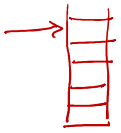
8

Computer Memory

- CPU executes the instructions for which the instructions and operands have to come from memory unit
- Operations which involve memory:
  - Instruction fetch
    - Memory read
  - Memory operand fetch and store
    - Memory read
    - Memory write
- Instructions involving memory access:
  - LOAD and STORE instructions
- Block transfer
  - If read or write operations involve consecutive address locations in the main memory
  - Only address sent is the starting address

LOAD R0, B
LOAD R1, C
ADD R2, R1, R0
STORE A, R2

ADDM A, B, C



9

9

Memory Performance Parameters

- **Access Time**:
  - Time interval between initiation of one operation and completion of that operation
  - Example: Time between assertion of Read signal and MFC signal
- **Memory Cycle Time**:
  - Minimum time delay between the initiation of two successive memory operations
  - Time delay between start of a read/write operation to start of next memory operation
- Memory cycle time is usually slightly larger than access time

10

10

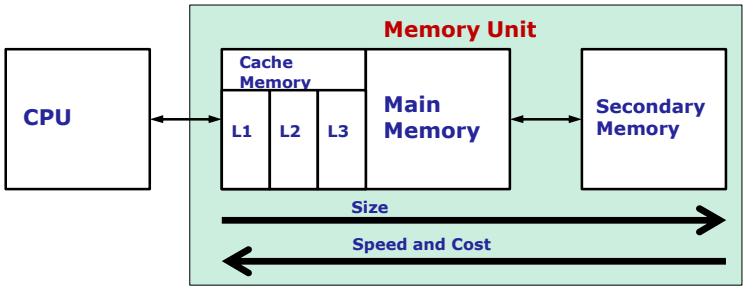
Memory Latency and Memory Organization

- **Latency**: Time to access the first of the sequence of memory words
- What is involved in determining the latency of the memory operation?
  - Processor issues the logical address to memory unit
  - The logical address need to be converted into physical address
- Memory unit is called random access memory (RAM)
  - Any location can be accessed for read/write operation independent of the location's address
- Memory unit is organised in hierarchical manner

11

11

Memory Hierarchy



- Processor processes instructions and data faster than it can be fetched from memory unit
- Memory access time is the bottleneck
- One way to reduce memory access time is to use faster memory
  - A small and faster memory bridge the gap between processor and main memory
- **Virtual memory**

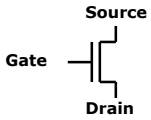
12

12

Semiconductor Memories

1 bit → Memory cell

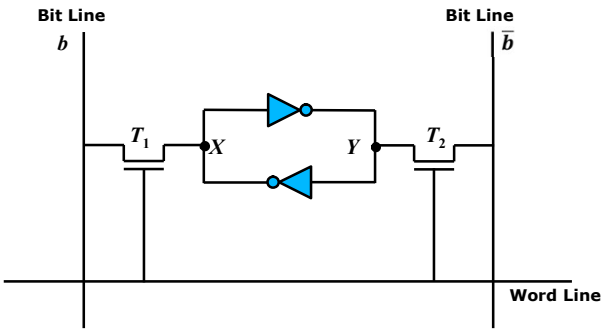
- Two basic ways of designing memory
  - Static RAM (SRAM)
  - Dynamic RAM (DRAM)
- **Static RAM**:
  - Built using metal-oxide semiconductor (MOS) transistors
  - MOS transistors acts as switch
    - +5 v (when Gate input is 1): Transistor conducts: ON state
    - 0 v (when Gate input is 0): Transistor does not conducts: OFF state



13

13

Static RAM Cell

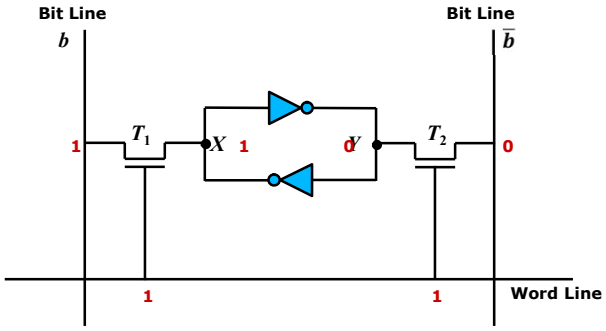


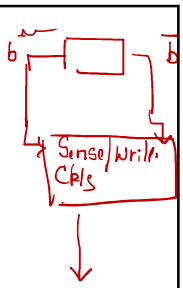
- Two inverters are cross connected to form latch
- Inverters are connected to 2 transistors which act as switches
- Switches are opened or closed under the control of word line
- This circuit retain the state (bit) as long as power is applied (Static Memory)

14

14

Static RAM Cell - Read



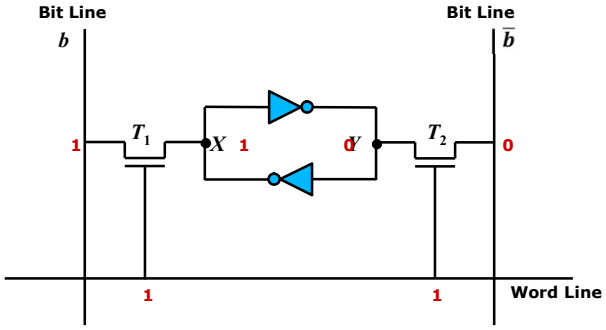


- Two inverters are cross connected to form latch
- Inverters are connected to 2 transistors which act as switches
- Switches are opened or closed under the control of word line
- This circuit retain the state (bit) as long as power is applied (Static Memory)

15

15

Static RAM Cell - Write

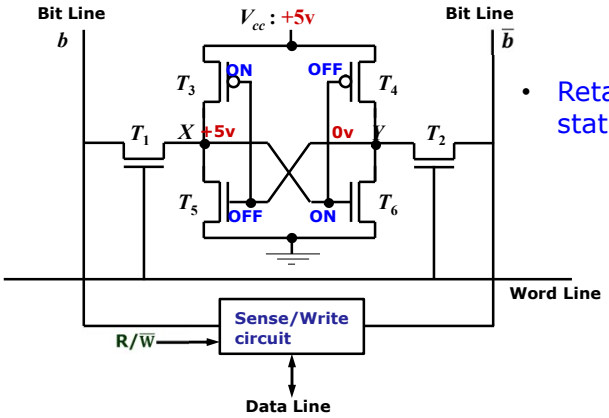


- Two inverters are cross connected to form latch
- Inverters are connected to 2 transistors which act as switches
- Switches are opened or closed under the control of word line
- This circuit retain the state (bit) as long as power is applied (Static Memory)

16

16

CMOS Static RAM Cell



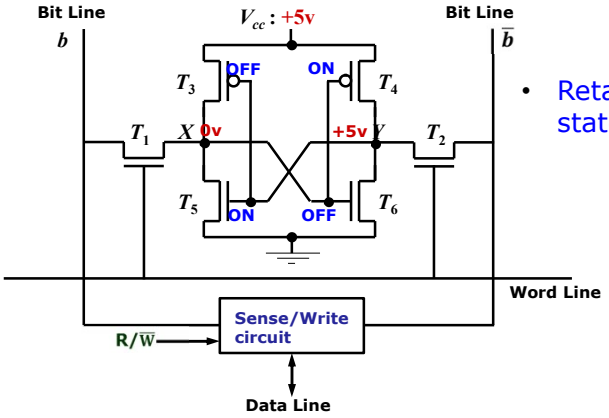
- Retaining the state 1

- This circuit retain the state (bit) as long as power is applied (Static Memory)
- Continuous power is needed for a cell to retain the state

17

17

CMOS Static RAM Cell



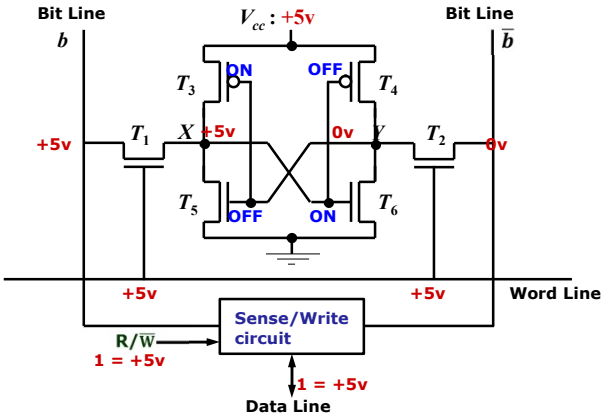
- Retaining the state 0

- This circuit retain the state (bit) as long as power is applied (Static Memory)
- Continuous power is needed for a cell to retain the state

18

18

CMOS Static RAM Cell - Read

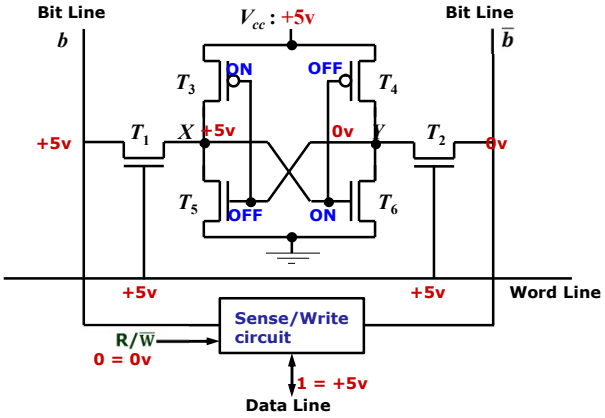


- This circuit retain the state (bit) as long as power is applied (Static Memory)
- Continuous power is needed for a cell to retain the state

19

19

CMOS Static RAM Cell - Write

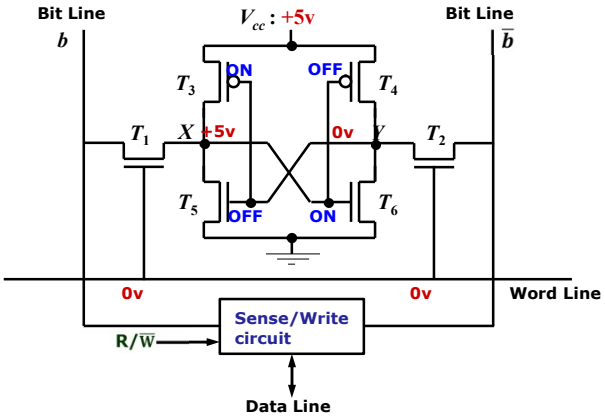


- This circuit retain the state (bit) as long as power is applied (Static Memory)
- Continuous power is needed for a cell to retain the state

20

20

CMOS Static RAM Cell



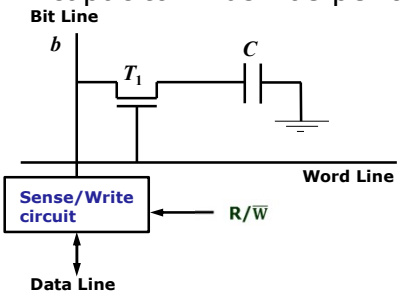
- Volatile
- Low power consumption-current flows in the cell only when the cell is accessed
- Access time is less i.e. faster memory
- Uses 6 transistors: Costly
- Used in applications where speed is critical concern: Cache

21

21

Dynamic RAM (DRAM) Cell

- Less expensive and simpler cell
- Information is stored in the form of a charge on a capacitor (C)
  - Charge in capacitor is stored only for short time
    - Tens of a millisecond
  - However, a cell is required to store information for a much longer time
- To retain information for longer time, content of capacitor mush be periodically refreshed



- Low speed as refresh needed
- Only 1 transistor is used
- Used to build main memory

22

22

Reference

- **Carl Hamacher, Zvonko Vranesic and Safwat Zaky, "Computer Organization", 5th Edition, Tata McGraw Hill, 2002**

23

23

Thank You

24

24