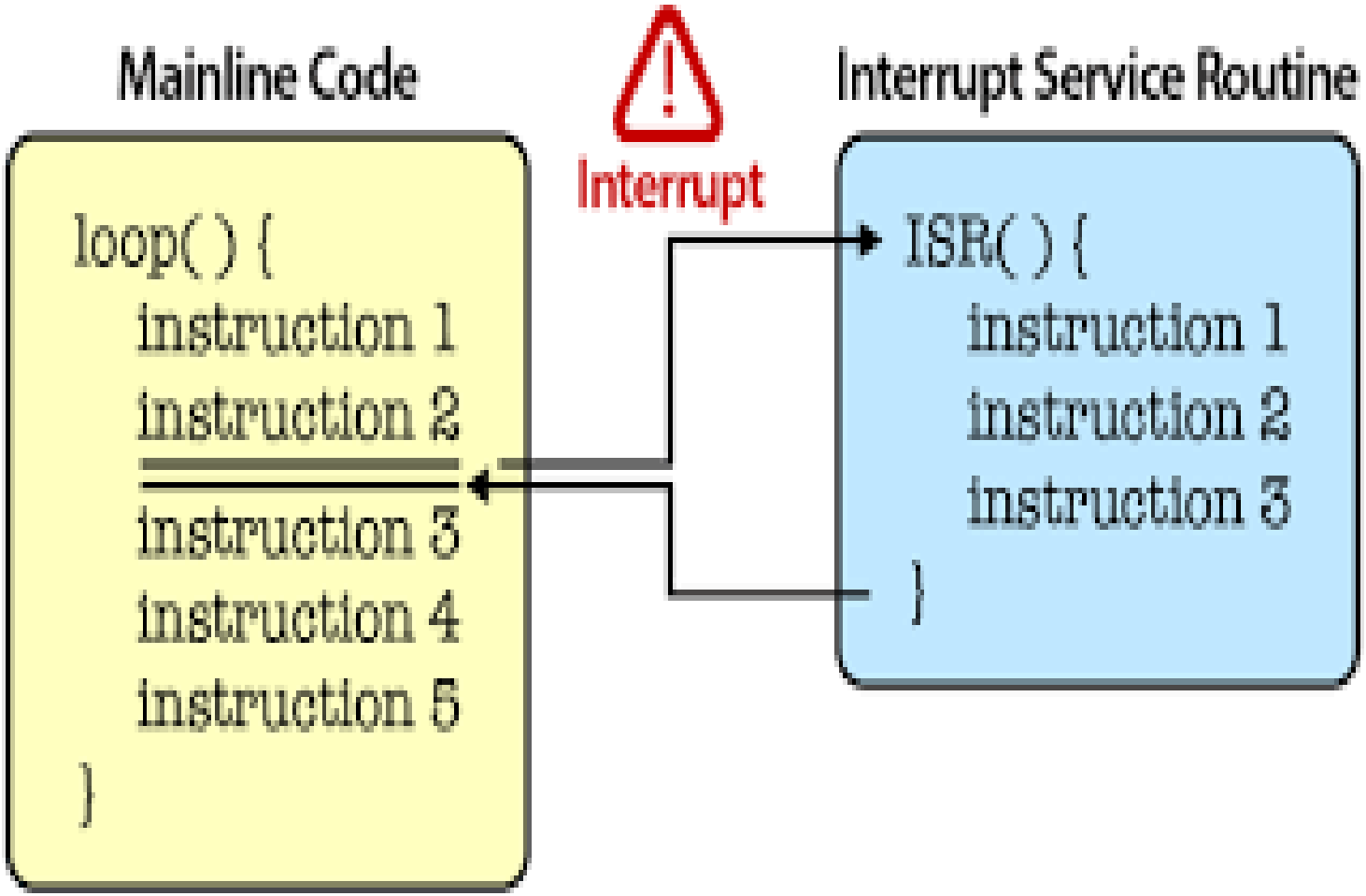
Interrupts

- Interrupt is a signal to the processor that some important event has occurred.
- If we press a key , an interrupt goes to the processor and then the processor reads the key pressed.
- Corresponding to each interrupt, there will be an **Interrupt Service Routine (ISR)** that will be executed. But, before executing the ISR, the processor has to temporarily halt the work it is doing.
- So it saves the contents of the registers by pushing the register values and stack pointer onto the stack.
- Then the processor loads the interrupt vector i.e the address at which the ISR is lying, into the Program Counter.

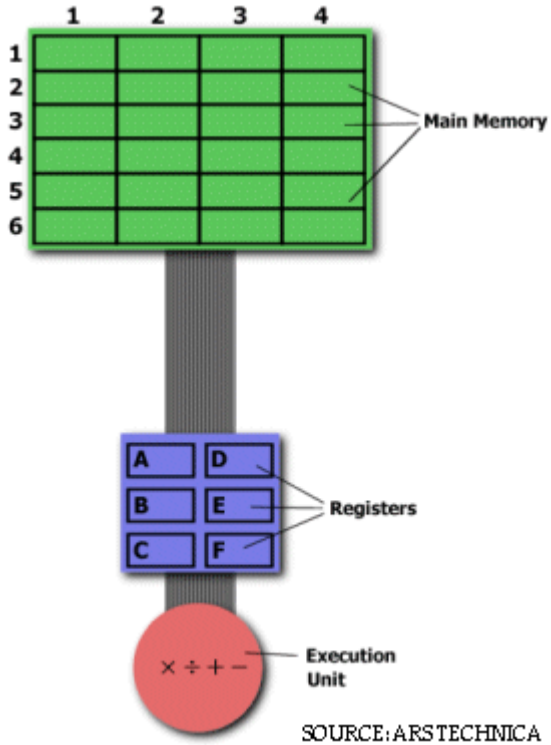
Interrupts


Mainline Code
Interrupt
Interrupt Service Routine
loop( ) {
instruction 1
instruction 2
instruction 3
instruction 4
instruction 5
}
ISR( ) {
instruction 1
instruction 2
instruction 3
}

- A Programmable Interrupt Controller (PIC) interfaces with the CPU to handle external devices and decide which interrupt has to be processed. An interrupt table contains the details of various interrupts such as interrupt number (IRQ), interrupt vector, priority of interrupt, frequency with which interrupt is likely to occur.
- Interrupt that has to be processed immediately is called Non-Maskable Interrupt (NMI)

CISC

- CISC stands for Complex Instruction Set Computer (CISC).
- CISC is characterized by its large instruction set.
- A large number of instructions are available to program the processor. So, the number of instructions required to do job is very less and hence less memory is required.
- The aim of the CISC processors is to reduce the software complexity by increasing the complexity of the processor.
- Example are Intel x86 family and Motoralla 68000 series processors

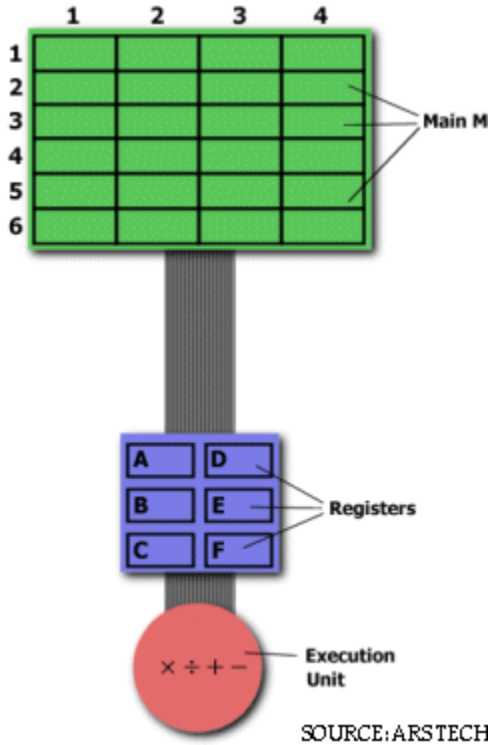


- The primary goal of CISC architecture is to complete a task in as few lines.
- This is achieved by building processor hardware that is capable of understanding and executing a series of operations.
- For this particular task, a CISC processor would come prepared with a specific instruction (we'll call it "MULT").
- When executed, this instruction loads the two values into separate registers, multiplies the operands in the execution unit, and then stores the product in the appropriate register.
- Thus, the entire task of multiplying two numbers can be completed with one instruction:

MULT 2:3, 5:2

MULT 2:3, 5:2

For instance, if we let "a" represent the value of 2:3 and "b" represent the value of 5:2, then this command is identical to the C statement "a = a * b."



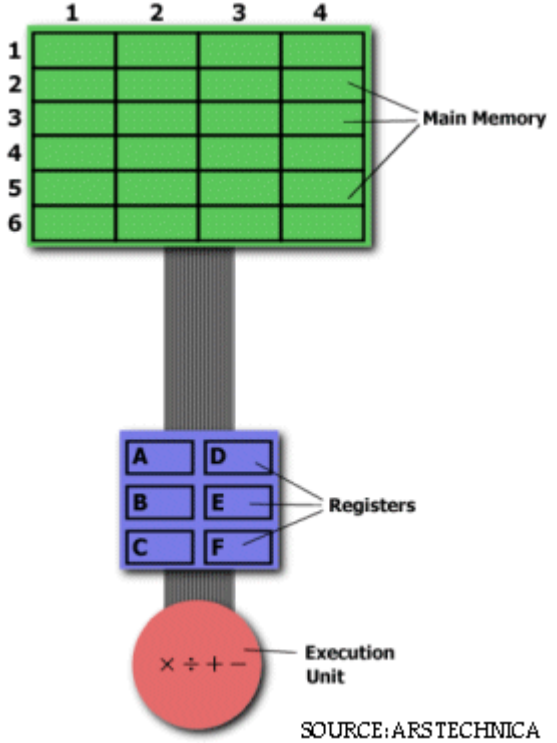
One of the primary advantages of this system is that the compiler has to do very little work to translate a high-level language statement into assembly. Because the length of the code is relatively short, very little RAM is required to store instructions. The emphasis is put on building complex instructions directly into the hardware.

RISC

- RISC stands for Reduced Instruction Set Computer is characterized by its limited number of instructions.
- A complex instruction is obtained as a sequence of simple instructions.
- So in this processor, software is complex but the processor architecture is simple.
- Large number of registers are required in RISC processors, which are small size and consume less power.
- RISC processors are in the pipelined instructions.
- In this case, one instruction is being executed, second instruction is decoded and the third instruction is fetched to faster execution of the program.
- Embedded systems generally use RISC processors. Example are ARM, ATMEL, AVR,MIPS, Microchip PIC, Power PC and Sun SPARC.

- RISC processors only use simple instructions that can be executed within one clock cycle.
- Thus, the "MULT" command described above could be divided into three separate commands:
  - "LOAD," which moves data from the memory bank to a register,
  - "PROD," which finds the product of two operands located within the registers, and
  - "STORE," which moves data from a register to the memory banks. In order to perform the exact series of steps described in the CISC approach, a programmer would need to code four lines of assembly:

```
LOAD A, 2:3
LOAD B, 5:2
PROD A, B
STORE 2:3, A
```



- At first, this may seem like a much less efficient way of completing the operation. Because there are more lines of code, more RAM is needed to store the assembly level instructions. The compiler(in case of High level Language) must also perform more work to convert a high-level language statement into code of this form.

- However, the RISC strategy also brings some very important advantages. Because each instruction requires only one clock cycle to execute, the entire program will execute in approximately the same amount of time as the multi-cycle "MULT" command.
- These RISC "reduced instructions" require less transistors of hardware space than the complex instructions, leaving more room for general purpose registers.
- Because all of the instructions execute in a uniform amount of time (i.e. one clock), pipelining is possible.

Analyze processor

Clock speed

Length of the registers

Width of data bus and address bus

Number of registers

Internal RAM

Internal ROM

On-chip peripherals such as timer, UART, ADC, DAC

Interrupt lines

Number of programmable I/O lines

Memory

- Memory is an important part of processor based embedded systems.
- If the processors contain built in memory then this memory is called **on-chip memory**.
- If the memory is connected with processor externally than this is called **off-chip memory**.

- The memory is divided into two categories:
  - Program memory
  - Data memory
- Program memory stores the firmware permanently.
- Data Memory contents are erased when power is switched off.
- These both can be internal or external to the processor.

Program Storage Memory (ROM)

- The program memory or code storage memory of an embedded system stores the program instructions.
- The code memory retains its contents even after the power to it is turned off. It is generally known as non-volatile storage memory.
- Depending on the fabrication, erasing and programming techniques they are classified into several types.

Program Storage Memory (ROM)

- **Masked ROM (MROM) :**
  - Masked ROM is a one-time programmable device.
  - The device is factory programmed by masking and metallization process at the end of production itself, according to the data provided by the end user.
  - The primary advantage of this is low cost for high volume production.
  - Masked ROM is a good for storing the embedded firmware for low cost embedded devices.
  - It is not possible to modify the device firmware if firmware is to be upgraded.
  - And also since MROM is permanent in bit storage, it is not possible to alter the bit information.

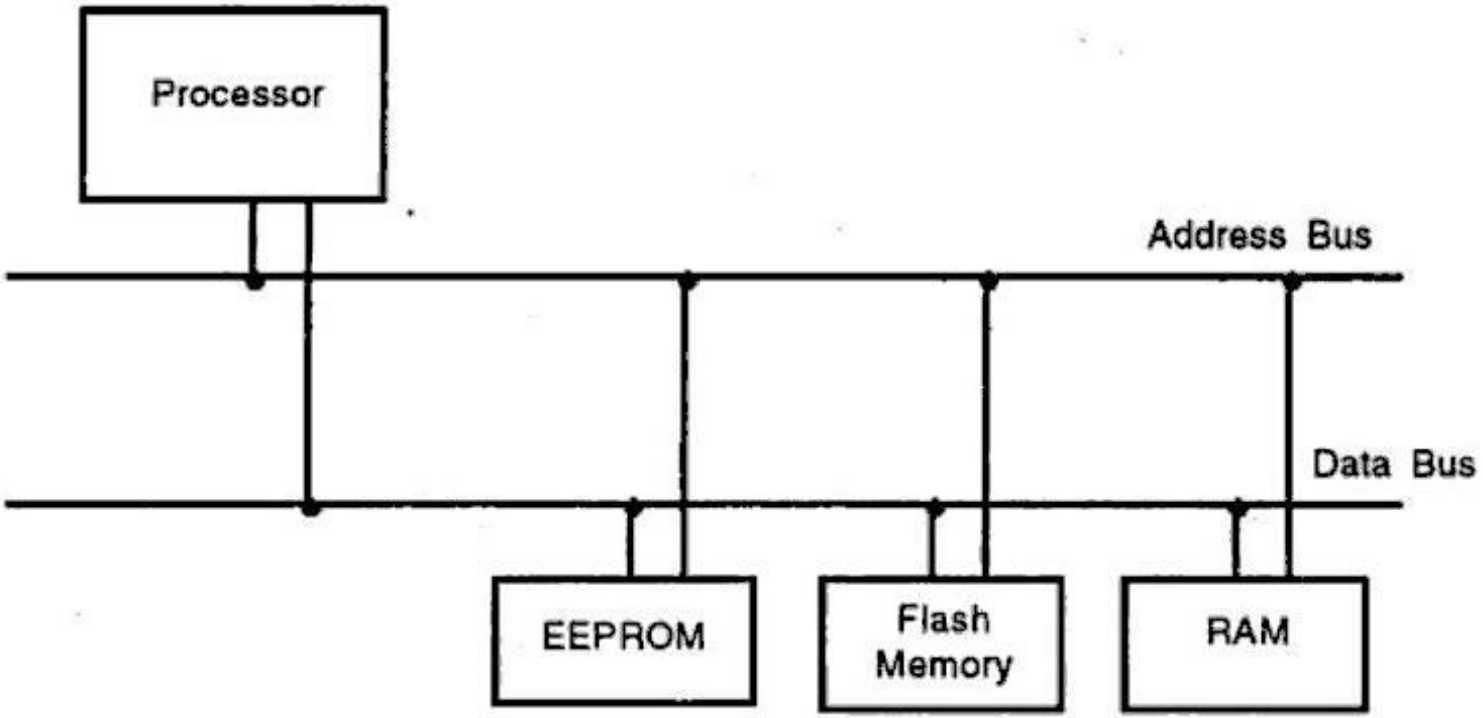
- **Programmable Read Only Memory (PROM)** :
  - It is also called **One Time Programmable Memory (OTP):**
  - It is not pre-programmed by the manufacturer.
  - The end user is responsible for programming these devices.
  - It is a low cost solution for commercial production and is **not able to reprogrammed**.
  - It is not useful and worth for the development purpose as need to change the code continuously.

- **Erasable Programmable Read Only Memory (EPROM):**.
  - During the development phase the code is subject continuous changes and using an OTP each time to load the code is not economical. So EPROM gives the flexibility to re-program the same chip
  - EPROM Programmer is need to store the information.
  - To erase the content, it is necessary to expose in ultraviolet rays for fixed duration.
  - Even though the EPROM chip is flexible in terms of re-programmability, it needs to be taken out of the circuit board and put in a UV eraser device for 20-30 minutes. So it is a tedious and time-consuming process.

- **Electrically Erasable Programmable Read Only Memory(EEPROM):**
  - The information contained in the EEPROM memory can be altered by using electrical signals at the register/Byte level.
  - It is erasable and reprogrammed **in-circuit.**
  - These chips include a chip erase mode and in this mode they can be erased in a few milliseconds.
  - ATMEL's AT250 (1KB) , AT25160 (2 KB) AT25320 (4 KB), AT 25640 (8 KB) are the EEPROMs.

- **FLASH:**
  - Flash is the ROM technology and is the most popular ROM technology used in today's embedded designs.
  - FLASH memory is a variation of EEPROM technology. It combines the re-programmability of EEPROM and the high capacity of standard ROMs.
  - The memory is divided into sectors or blocks. Typical sector size is 256 bytes or 16 KB. Each sector is erasable unit.
  - Flash memory is nowadays extensively used in embedded systems for storing the firmware.
  - The main attraction of Flash data can be accessed through file names rather than through memory addresses.
  - AMD's AM29F010 is Flash device of 128 KB capacity.

Memory in Embedded System



Read-Write Memory/Random Access Memory (RAM)

- RAM is the **data memory** or working memory of the processor.
- Processor can read from it and write to it. RAM is volatile, meaning when the power is turned off, all the contents are destroyed.
- RAM is direct access memory, meaning we can access the desired memory location directly without the need for traveling through the entire memory locations to reach the desired memory position (i.e random access memory location)
- .
- Ram generally falls into three categories:
  - Static RAM (SRAM)
  - Dynamic RAM (DRAM)
  - Non-volatile Memory (NVRAM)

Memory Cell Operation

- The basic element of a **semiconductor memory** is the memory cell.
- Although a variety of electronic technologies are used, all semiconductor memory cells share certain properties:
  - They exhibit two stable states, which can be used to represent binary 1 and 0.
  - They are capable of being written into (at least once), to set the state.
  - They are capable of being read to sense the state.

Memory Cell Operation

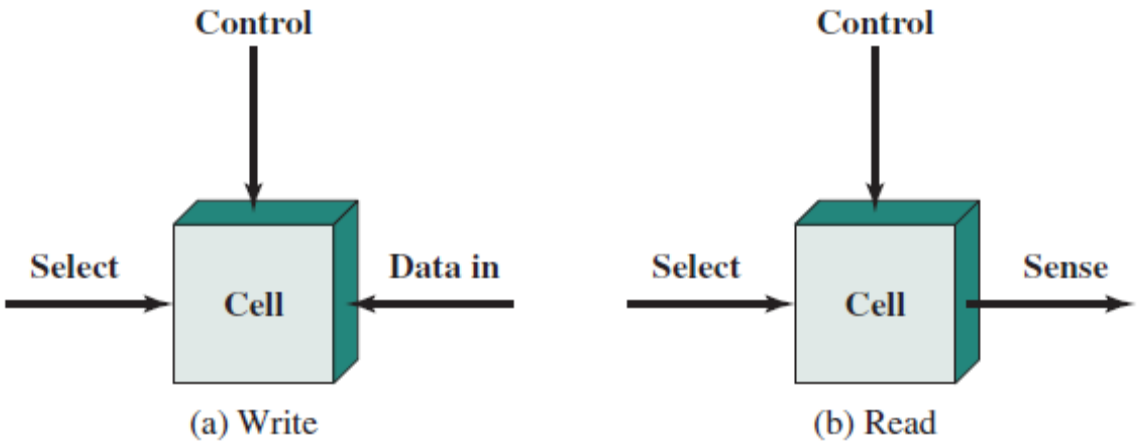
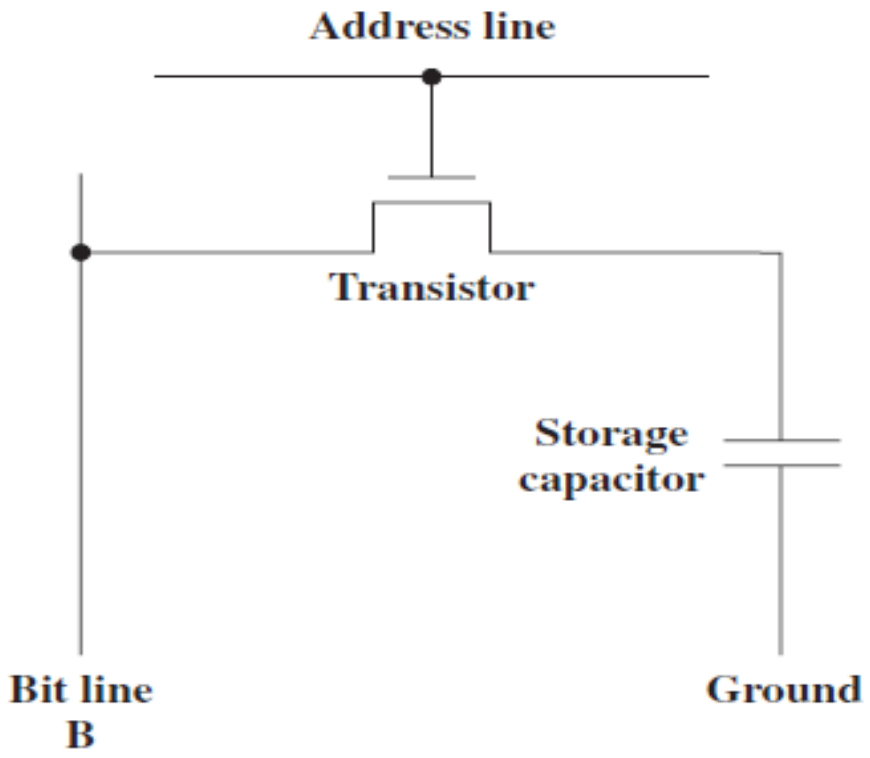


Figure 5.1 Memory Cell Operation

Memory Cell Operation

- Most commonly, the cell has three functional terminals capable of carrying an electrical signal.
- The select terminal, as the name suggests, selects a memory cell for a read or write operation.
- The control terminal indicates read or write operation.
- For writing, the other terminal provides an electrical signal that sets the state of the cell to 1 or 0.
- For reading, that terminal is used for output of the cell's state.

DRAM



Bi

(a) Dynamic RAM (DRAM) cell

DRAM

- A **dynamic RAM (DRAM)** is 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.
- Because capacitors have a natural tendency to discharge, dynamic RAMs require periodic charge refreshing to maintain data storage.

DRAM

- Figure(DRAM) shows the structure for an individual cell that stores one bit.
- The address line is activated when the bit value from this cell is to be read or written.
- The transistor acts as 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.

DRAM

- For the write operation, a voltage signal is applied to the bit line; a high voltage represents 1, and a low voltage represents 0.
- A signal is then applied to the address line, allowing a charge to be transferred to the capacitor.

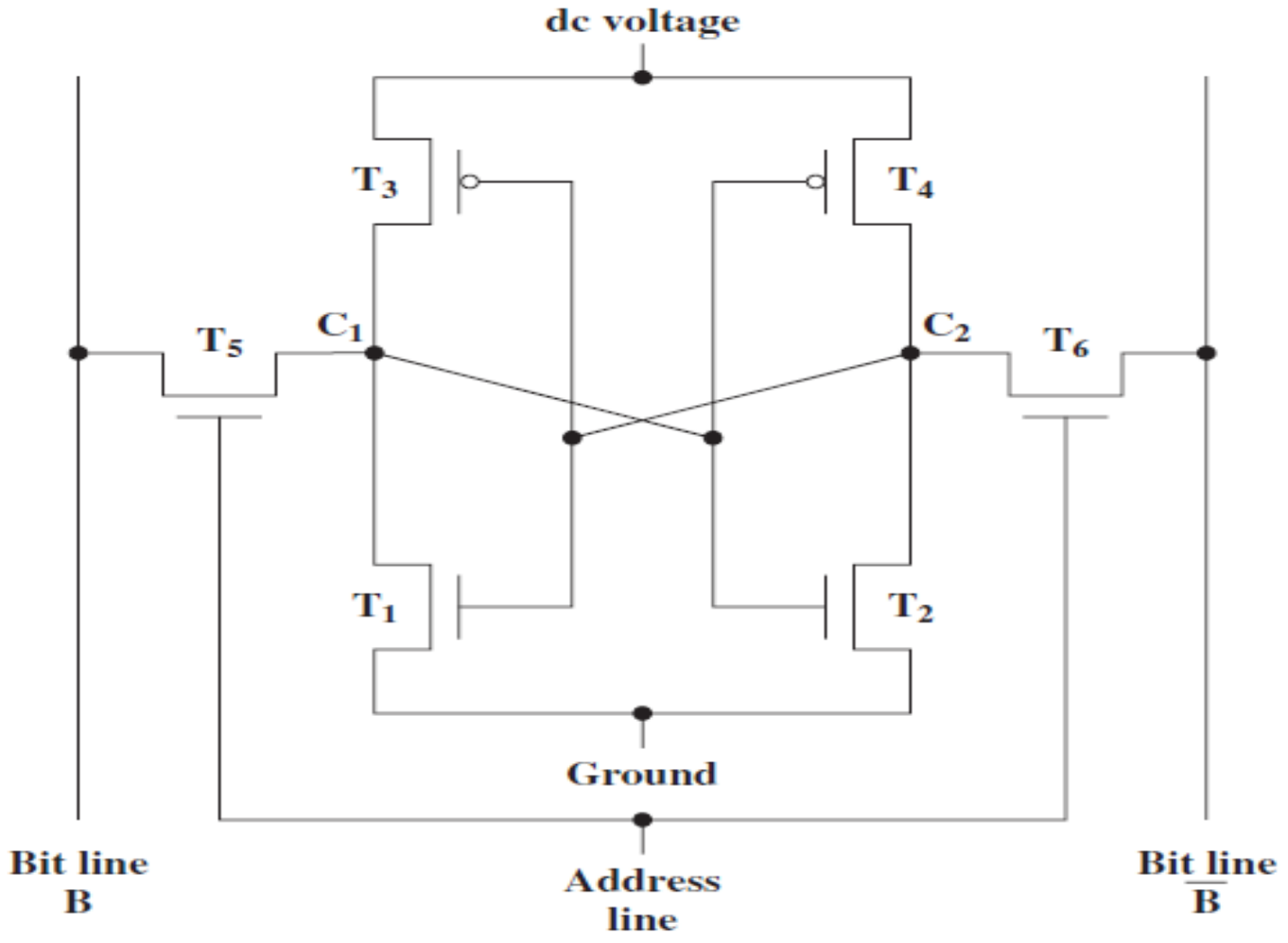
DRAM

- For the read operation, when the address line is selected, the transistor turns on and the charge stored on the capacitor is fed out onto a bit line and to a sense amplifier.
- The sense amplifier compares the capacitor voltage to a reference value and determines if the cell contains a logic 1 or a logic 0.

DRAM

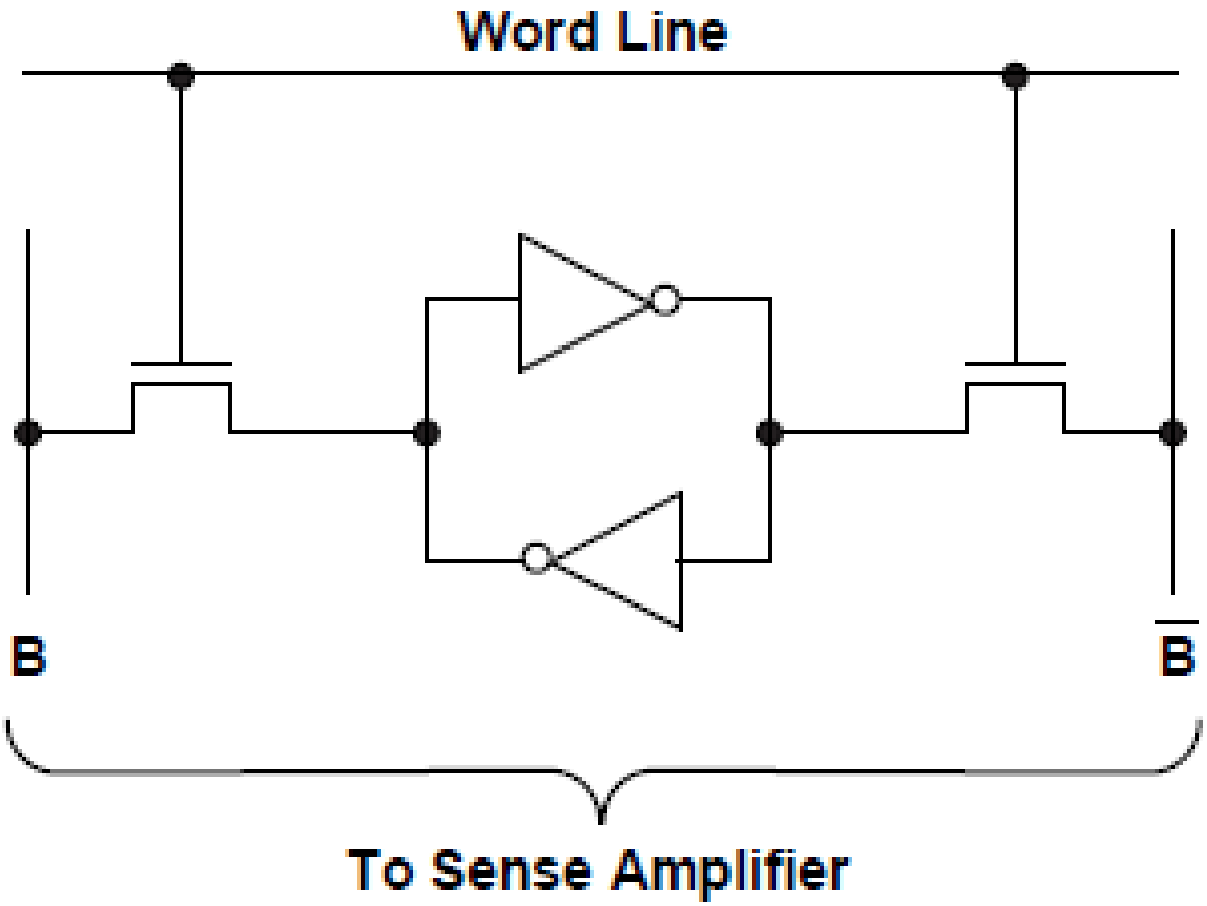
- Although the DRAM cell is used to store a single bit (0 or 1), it is essentially an analog device.
- The capacitor can store any charge value within a range; a threshold value determines whether the charge is interpreted as 1 or 0.

SRAM



(b) Static RAM (SRAM) cell

SRAM



Source: ICE, "Memory 1997"

SRAM

- SRAM is a digital device and stores binary values using traditional flip-flop logic-gate to form stable state configurations.
- A static RAM will hold its data as long as power is supplied to it.
- SRAM figure(b) shows structure for an individual cell.

SRAM

- Four transistors (T1, T2, T3, T4) are cross connected in an arrangement that produces a stable logic state.
- In logic state 1, point C1 is high and point C2 is low; in this state, T1 and T4 are off and T2 and T3 are on.
- In logic state 0, point C1 is low and point C2 is high; in this state, T1 and T4 are on and T2 and T3 are off.
- Both states are stable as long as the direct current (dc) voltage is applied.
- Unlike the DRAM, no refresh is needed to retain data.

SRAM

- As in the DRAM, the SRAM address line(word line) is used to open or close a switch.
- The address line controls two transistors (T5 and T6). When a signal is applied to this line, the two transistors are switched on, allowing a read or write operation.
- For a write operation, the desired bit value is applied to line B, while its complement is applied to line B.
- This forces the four transistors (T1, T2, T3, T4) into the proper state.
- For a read operation, the bit value is read from line B.

DRAM VS SRAM

SRAM Cell	DRAM Cell
Made up of 6 CMOS transistors (MOSFET)	Made up of a MOSFET and a capacitor
Does not require refreshing	Requires refreshing
More expensive	Less expensive
Fast in operation.	Slow in operation due to refresh requirements.

- **NVRAM:**
  - Non-volatile RAM is a random access memory with battery backup.
  - It contains static RAM based memory and a battery for providing supply to the memory in the absence of external power supply.
  - The memory and battery are packed together in a single package.
  - The life span of NVRAM is expected to be around 10 years.