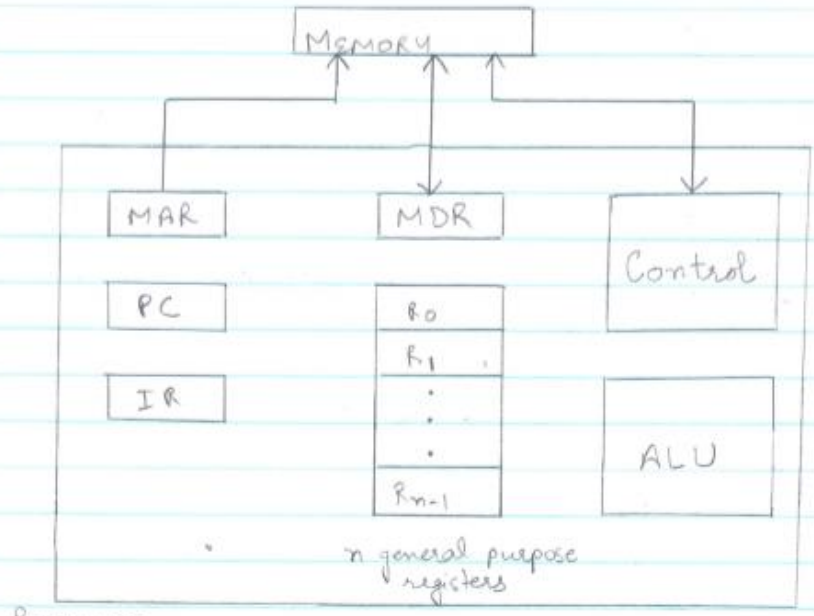
VTU Computer Organization -Solution (18cs34)

Ans 1 Q1 Connection between memory and processor (operating steps) with diagram?

Now, how contents are transferred between memory and processor?
Address of memory location to be accessed is sent to memory unit. Control signals are issued, and data is transferred do or from memory.



Connections b/w Processor & the memory

Memory :- Stores data and instructions.

- Instruction register (IR) - Holds instructions that is currently being executed. Its output is available to the control circuits which generate the timing signals that control various processing elements involved in executing the instruction

- Program counter (PC) - contains memory address of next instruction to be fetched and executed.

- Memory address register (MAR) - holds address of location to be accessed

- Memory data register (MDR) - contains data to be written into or read out of the addressed location.

Operating steps

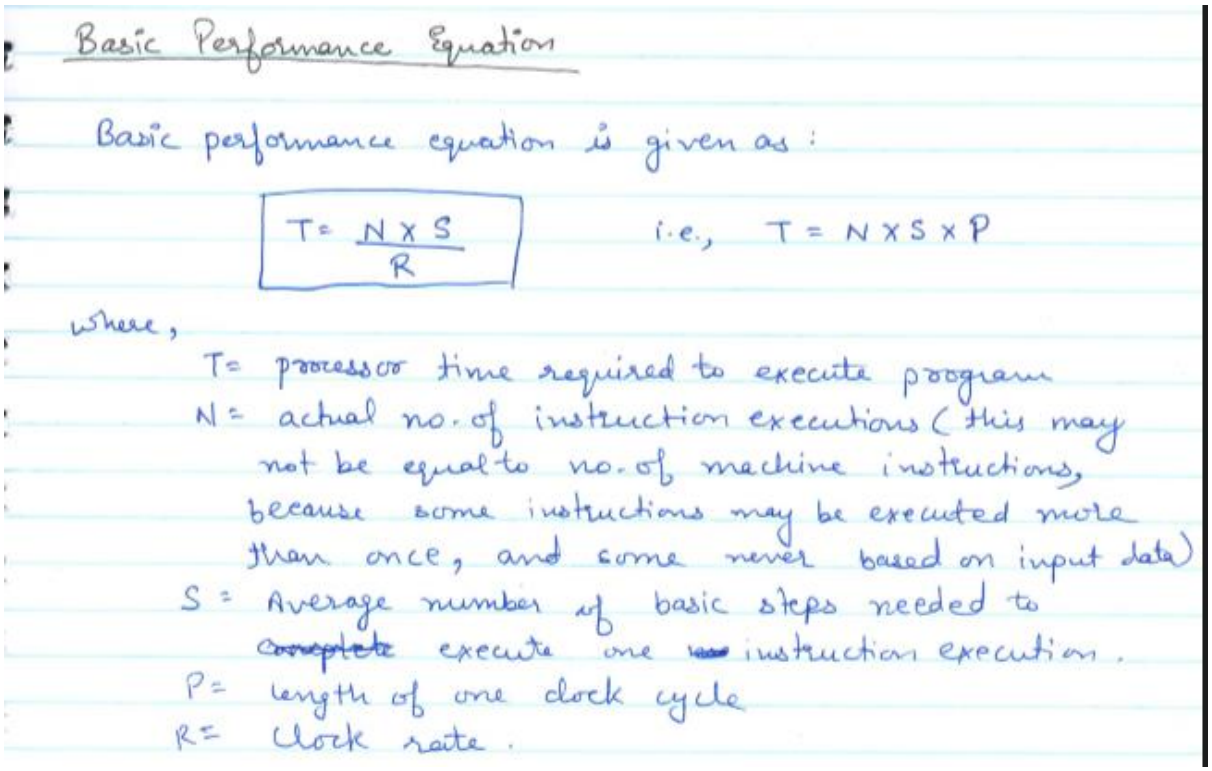
* Programs (list of instructions) reside in memory (usually ~~stored there wrong~~ get there through input unit).

1. PC is set to point to first instruction of program.
2. This PC contents is transferred to MAR and Read control signal is sent to memory.
3. After time required to access the memory elapses, addressed word (1st instruction in this case) is read out from memory and loaded in MDR.
4. MDR contents are transferred to IR.
5. If instruction involves an operation by ALU:

get operands from memory or general purpose register.
If operand resides in memory, its address is sent to MAR. Read cycle is initialized. Operand comes to MDR. It is sent to ALU. Similarly, more operands are sent to ALU (if required).
ALU performs operation and sends result to MDR.
The address of location where result is to be stored is sent to MAR, and Write cycle is initiated.
⑥ PC is incremented to point to next instruction.

NOTE: If a source of destination is a register (R), MAR, MDR steps are not required as registers are directly accessible to ALU as both reside inside processor. MAR and MDR are required only if we want to access main memory for read or write operation.

Q2 Basic performance equation and SPEC rating



Performance Measurement

Therefore, now a days, computer performance is measured using benchmark programs. Standardized programs are used for better comparisons.

The performance measure is the time taken by computer to execute a given benchmark.

A non profit organization called System Performance Evaluation Corporation (SPEC) selects and publishes ~~representative~~ representative application programs for different application domains, together with test results for many commercially available computers.

The programs selected range from game playing, compiler and database applications to numerically intensive programs in astrophysics and quantum chemistry.

In each case, the program is compiled for the computer under test, and running time on real computer is measured. Simulation is not allowed. The same program is also compiled and run on one computer selected as a reference.

$$\text{SPEC rating} = \frac{\text{Running time on the reference computer}}{\text{Running time on the computer under test}}$$

The test is repeated for all the programs in SPEC suite, and geometric mean of results is computed.

Let SPEC_i be the rating for program i in the suite.

Overall SPEC rating is given by:

$$\text{SPEC rating} = \left(\prod_{i=1}^{n} \text{SPEC}_i \right)^{\frac{1}{n}}$$

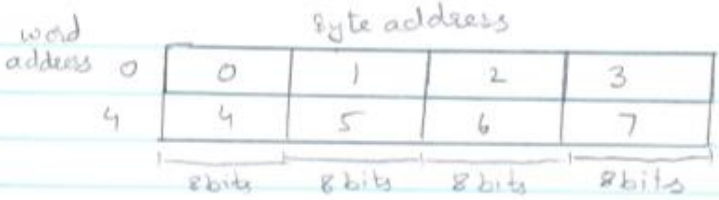
* Geometric mean :- n^{th} root of product of n values

where n is no. of programs in the suite.

Q3 Byte addressability (Big-endian and Little-endian assignments with diagram)

Byte Addressability

Successive addresses refer to successive byte locations in the memory. The term byte-addressability memory is used for this assignment.
Byte locations have addresses 0, 1, 2, ...
1 byte = 8 bits.
If word length of machine is 32-bits, successive words are located at addresses 0, 4, 8, ... each word consisting four bytes (32 bits)



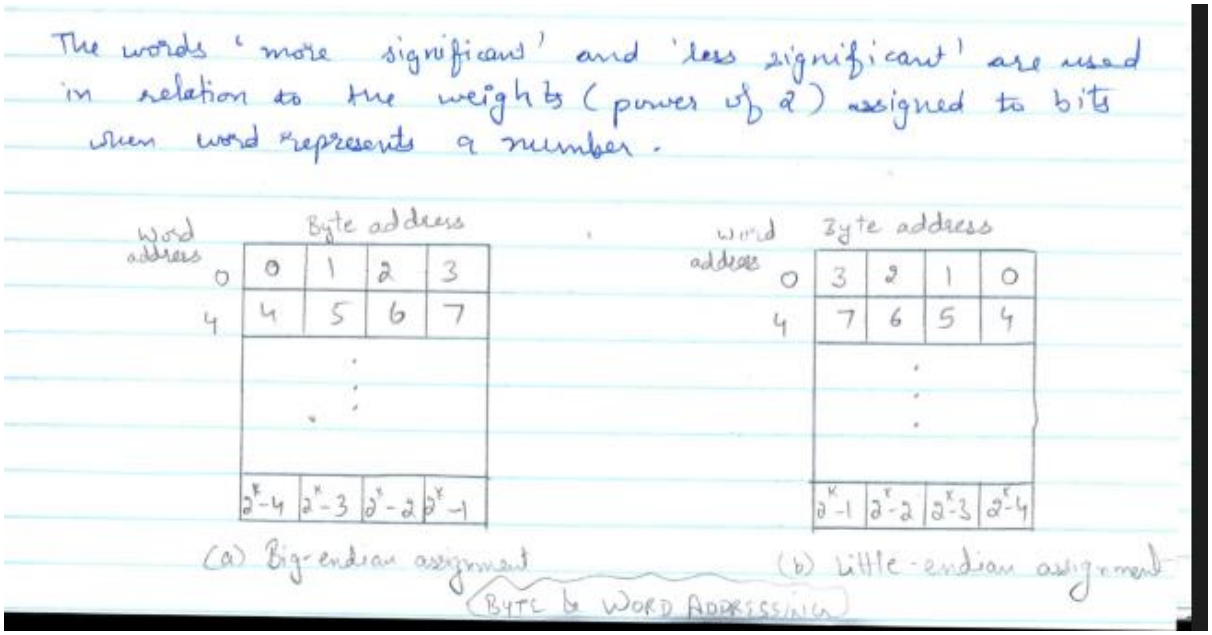
Big-endian and little-endian assignments.

These 2 methods are used for byte addressing. Any one method is selected out of these.

Big-endian assignment - Lower byte addresses are used for more significant bytes (leftmost bytes) of the word.

Little-endian assignment - Lower byte addresses are used for the less significant bytes (rightmost bytes) of the word.

The words 'more significant' and 'less significant' are used in relation to the weights (power of 2) assigned to bits when word represents a number.



Words are said to be aligned in memory if they begin at a byte address that is a multiple of the number of bytes in a word.

Q4 Instruction types (one-address, two-address, three-address instructions)

Basic Instruction Types

$C \leftarrow [A] + [B]$

Add contents of A & B, place sum in C. A, B contents are unchanged

Consider three-address instruction

Operation Source1, Source2, Destination

Add A, B, C

Two-address instruction

Operation Source, Destination

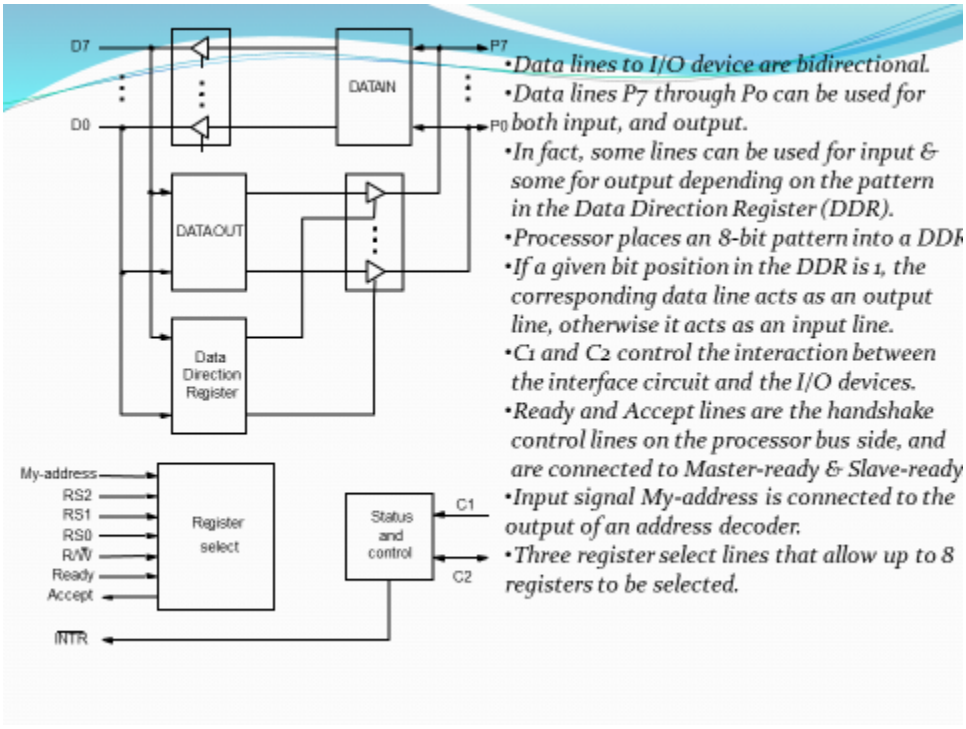
Move B, C
Add A, C

It may happen that 2-address instruction doesnot fit in one word for usual word length & address size. In that case, we may adopt one-address instruction. Operation Source/Destination.
A processor register, usually called the accumulator, may be used for this purpose. This may be used as a register to temporarily hold values.

Load	A	Copy A contents to accumulator
Add	B	Add B contents to accumulator
Store	C	Store accumulator contents to C.

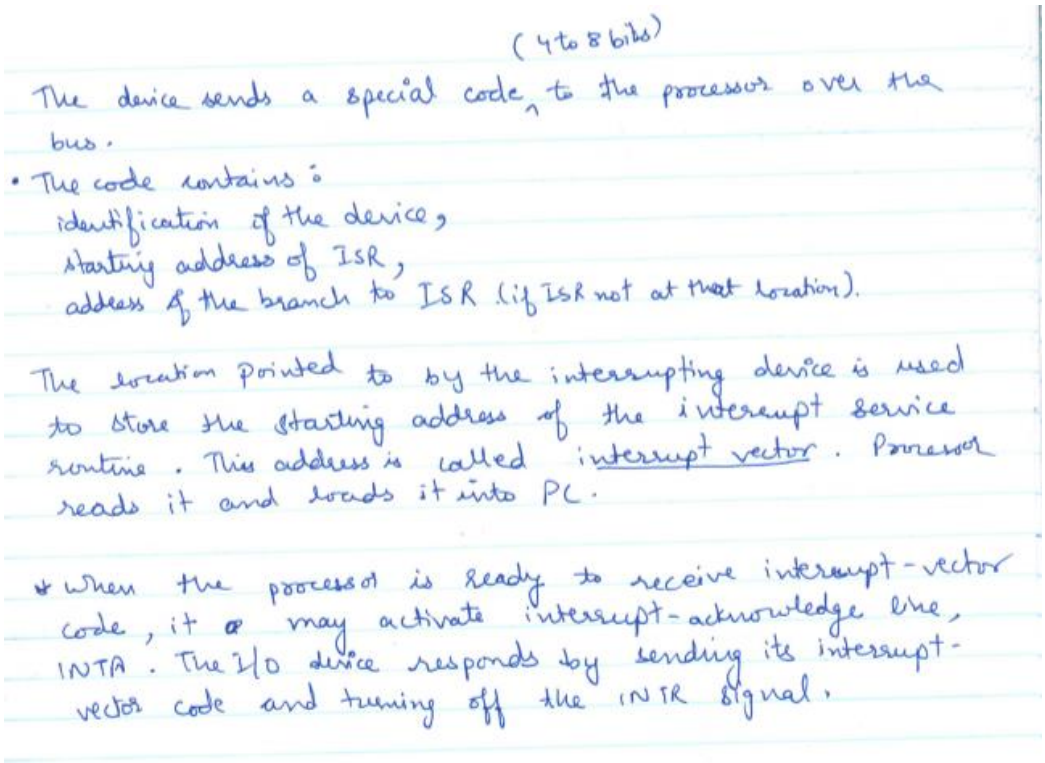
MODULE 2

1. With a neat diagram, explain general 8-bit parallel interface.



- *Data lines to I/O device are bidirectional.*
- *Data lines P7 through Po can be used for both input, and output.*
- *In fact, some lines can be used for input & some for output depending on the pattern in the Data Direction Register (DDR).*
- *Processor places an 8-bit pattern into a DDR*
- *If a given bit position in the DDR is 1, the corresponding data line acts as an output line, otherwise it acts as an input line.*
- *C1 and C2 control the interaction between the interface circuit and the I/O devices.*
- *Ready and Accept lines are the handshake control lines on the processor bus side, and are connected to Master-ready & Slave-ready*
- *Input signal My-address is connected to the output of an address decoder.*
- *Three register select lines that allow up to 8 registers to be selected.*

Ans Vectored Interrupts



(4 to 8 bits)

The device sends a special code, to the processor over the bus.

- The code contains :
  identification of the device,
  starting address of ISR,
  address of the branch to ISR (if ISR not at that location).

The location pointed to by the interrupting device is used to store the starting address of the interrupt service routine. This address is called interrupt vector. Processor reads it and loads it into PC.

* When the processor is ready to receive interrupt-vector code, it may activate interrupt-acknowledge line, INTA. The I/O device responds by sending its interrupt-vector code and turning off the INTR signal.

Module 3

Q1 Internal organization of RAM/memory chip/128bit memory chip?

- Memory-cells are organized in the form of array (Figure 8.2).
- Each cell is capable of storing 1-bit of information.
- Each row of cells forms a memory-word.
- All cells of a row are connected to a common line called as **Word-Line**.
- The cells in each column are connected to **Sense/Write** circuit by 2-bit-lines.
- The Sense/Write circuits are connected to data-input or output lines of the chip.
- During a write-operation, the sense/write circuit
  → receive input information &
  → store input info in the cells of the selected word.

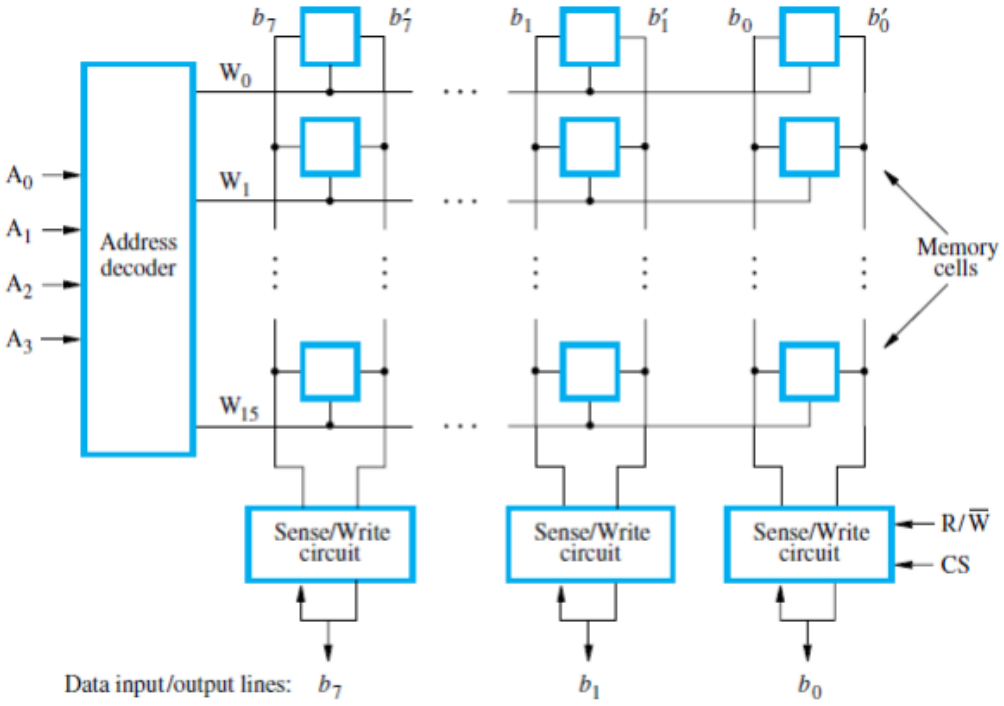


Figure 8.2 Organization of bit cells in a memory chip.

- The data-input and data-output of each Sense/Write circuit are connected to a single bidirectional data-line.
- Data-line can be connected to a data-bus of the computer.
- Following 2 control lines are also used:
  - **1) R/W'** → Specifies the required operation.
  - **2) CS'** → Chip Select input selects a given chip in the multi-chip memory-system.

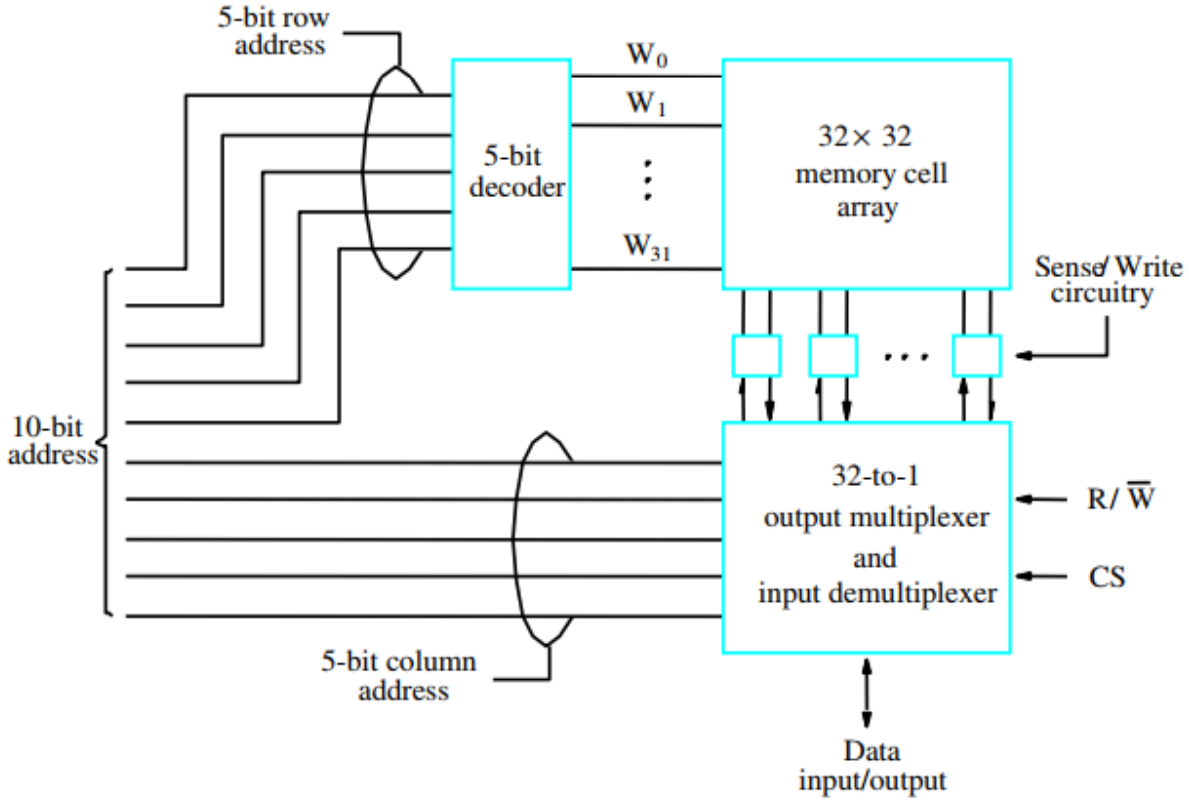


Figure 5.3. Organization of a 1K × 1 memory chip.

10- bit address line is needed, but there is only one data line resulting in 15 external connections. 10-bit address is divided into two groups of 5 bits each to form the row and the column addresses for the cell array. A row address selects a row of 32 cells, all of which are accessed in parallel. However, according to the column address, only one of these cells is connected to the external data line by output multiplexer and input demultiplexer.

Q3 Asynchronous and Synchronous DRAM

Asynchronous DRAM or 2M*8 Asynchronous DRAM

- Less expensive RAMs can be implemented if simple cells are used.
- Such cells cannot retain their state indefinitely. Hence they are called **Dynamic RAM (DRAM).**
- The information stored in a dynamic memory-cell in the form of a charge on a capacitor.
- This charge can be maintained only for tens of milliseconds.
- The contents must be periodically refreshed by restoring this capacitor charge to its full value.

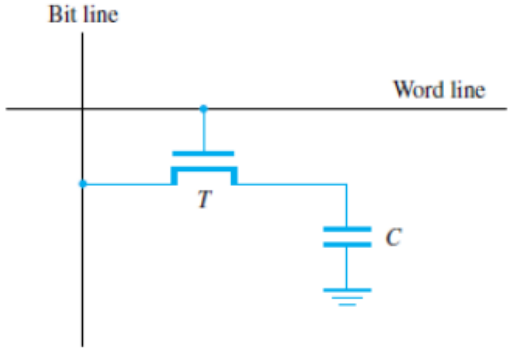


Figure 8.6 A single-transistor dynamic memory cell.

- In order to store information in the cell, the transistor T is turned 'ON' (Figure 8.6).
- The appropriate voltage is applied to the bit-line which charges the capacitor.
- After the transistor is turned off, the capacitor begins to discharge.
- Hence, info. stored in cell can be retrieved correctly before threshold value of capacitor drops down.
- During a read-operation,
  - → transistor is turned 'ON'
  - → a sense amplifier detects whether the charge on the capacitor is above the threshold value.
    - ➢ If (charge on capacitor) > (threshold value) → Bit-line will have logic value '1'.
    - ➢ If (charge on capacitor) < (threshold value) → Bit-line will set to logic value '0'.

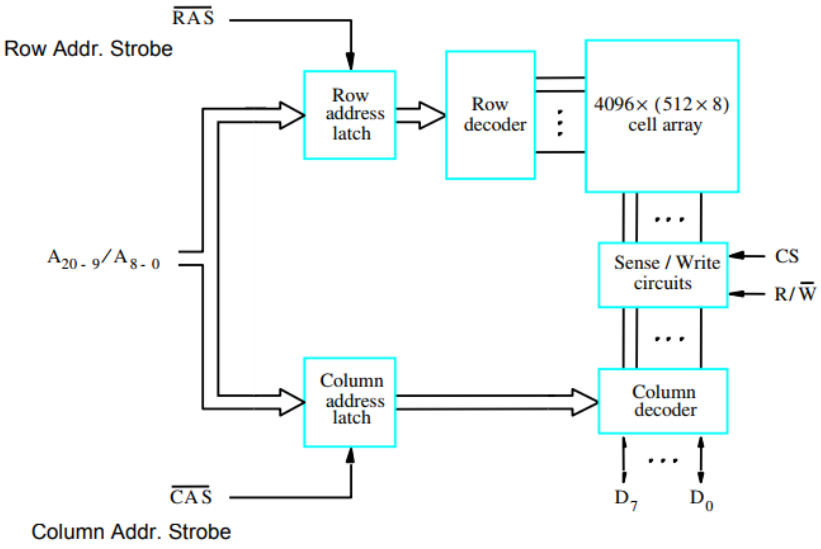


Figure 5.7. Internal organization of a 2M × 8 dynamic memory chip.

- During Read/Write-operation,
  - → row-address is applied first.
  - → row-address is loaded into row-latch in response to a signal pulse on **RAS'** input of chip. (RAS = Row-address Strobe CAS = Column-address Strobe)
- When a Read-operation is initiated, all cells on the selected row are read and refreshed.
- Shortly after the row-address is loaded, the column-address is

- 21 bit address is needed to access a byte in the memory. 21 bit is divided as follows:
  1) 12 address bits are needed to select a row.
     i.e. $A_{8-0} \rightarrow$ specifies row-address of a byte.
  2) 9 bits are needed to specify a group of 8 bits in the selected row.
     i.e. $A_{20-9} \rightarrow$ specifies column-address of a byte.

FAST PAGE MODE:

When DRAM in the above diagram is accessed, the contents of all 4096 cells in the selected row are sensed, but only 8 bits are placed on the data lines D_{7-0}, as selected by A_{8-0}. Fast page mode makes it possible to access the other bytes in the same row without having to reselect the row.
A latch is added at the output of the sense amplifier in each column.

Q6 Explain three types of mapping functions for cache memory.

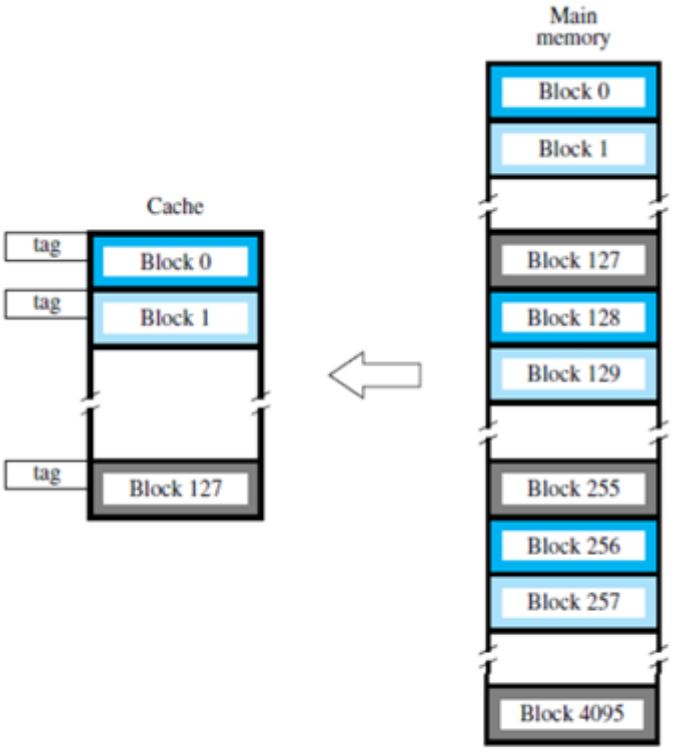


Figure 8.16 Direct-mapped cache.

DIRECT MAPPING

This technique is easy to implement but not very flexible.

Block j of the main memory maps onto j modulo 128 of the cache. For example, whenever **one of** the main memory blocks 0, 128, 256, ... Is loaded in the cache, it is stored in cache block 0 . Main memory blocks 1, 129, 257, ... are stored in cache block 1 (one at a time), and so on. Contention may occur for a single cache block required by multiple memory blocks. E.g when for program execution both memory block 1 and 129 are required but cache block 1 can only store one memory block. To resolve this, new blocks are allowed to overwrite the currently resident block.

From example,

4096 memory blocks need to be mapped to 128 cache blocks. i.e, each cache block identified 32 memory blocks(4096/128).

Main memory address is divided into three parts:

Tag (5 bits): identify which memory block (out of 32 in this case) is currently resident in the cache

Block (7 bits): cache block position where the new memory block must be stored

Word (4 bits): selects one of the words of the memory block (out of 16 words per block in this case)

ASSOCIATIVE MAPPING

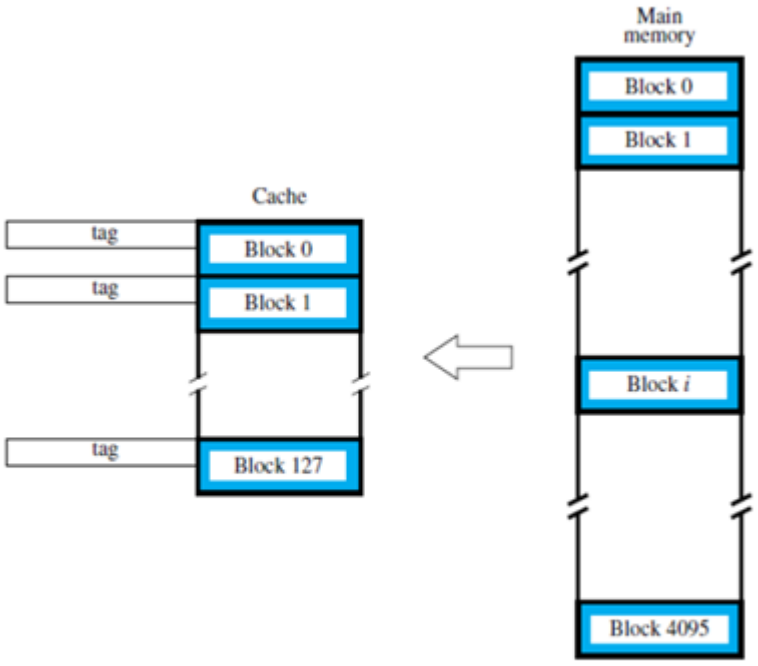


Figure 8.17 Associative-mapped cache.

- It is more flexible than direct mapping technique but more expensive. Main memory block can be placed into any cache block position.
- Memory address is divided into two fields:
  - Low order 4 bits identify the memory word within a block.
  - High order 12 bits or tag bits identify a memory block when residing in the cache.
- Flexible, and uses cache space efficiently.
- Replacement algorithms can be used to replace an existing block in the cache when the cache is full.
- Cost is higher than direct-mapped cache because of the need to search all 128 patterns to determine whether a given block is in the cache.

SET-ASSOCIATIVE MAPPING

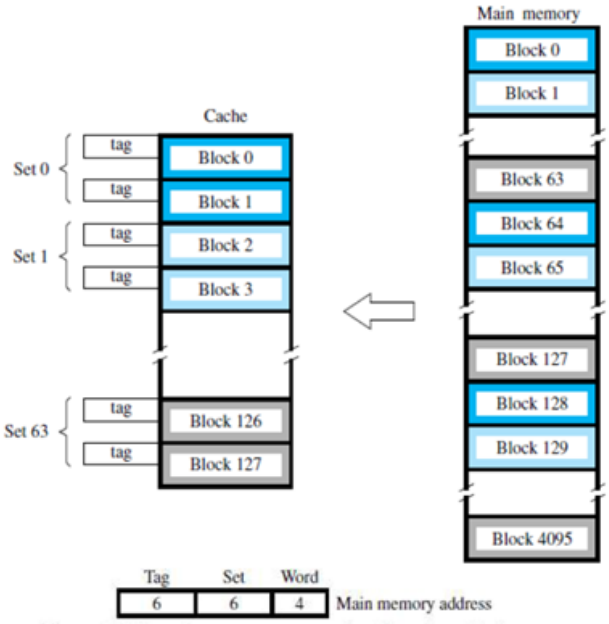


Figure 8.18 Set-associative-mapped cache with two blocks per set.

It is a **combination** of direct mapping and associative mapping techniques.
Blocks of the cache are grouped into **sets**, and the mapping allows a block of the main memory to reside in **any block of a specific set**.
Contention problem of direct mapping is eased by having a few choices for block placement. Hardware cost is reduced by decreasing the associative search.

Module 4

With a figure, explain circuit arrangement for binary division.

• An n-bit positive-divisor is loaded into register M. An n-bit positive-dividend is loaded into register

Q at the start of the operation. Register A is set to 0

• After division operation, the n-bit quotient is in register Q, and the remainder is in register A.

• Procedure:

step 1:

Do the following n times

i) If the sign of A is 0, shift A and Q left one bit position and subtract M from

A; otherwise, shift A and Q left and add M to A .

ii) Now, if the sign of A is 0, set q0 to 1; otherwise set q0 to 0.

Step 2:

If the sign of A is 1, add M to A (restore).

bit carry look-ahead adder

CARRY-LOOKAHEAD ADDITIONS

• The logic expression for si(sum) and ci+1(carry-out) of stage i are

si= xi+yi+ci ------(1)

ci+1=xiyi+xici+yici ------(2)

• Factoring (2) into

ci+1=xiyi+(xi+yi)ci

we can write ci+1=Gi+ PiCi where Gi=xiyi and Pi=xi+yi

• The expressions Gi and Pi are called generate and propagate functions .

• If Gi=1, then ci+1=1, independent of the input carry ci. This occurs when both xi and yi are 1.

Propagate function means that an input-carry will produce an output-carry when either xi=1 or yi=1.

• All Gi and Pi functions can be formed independently and in parallel in one logic-gate delay.

• Expanding ci terms of i-1 subscripted variables and substituting into the ci+1 expression, we obtain

ci+1= Gi+PiGi-1+PiPi-1Gi-2. . . . . .+P1G0+PiPi-1 . . . P0c0

• Conclusion: Delay through the adder is 3 gate delays for all carry-bits &

4 gate delays for all sum-bits.

• Consider the design of a 4-bit adder. The carries can be implemented as

c1=G0+P0c0

c2=G1+P1G0+P1P0c0

c3=G2+P2G1+P2P1G0+P2P1P0c0

c4=G3+P3G2+P3P2G1+P3P2P1G0+P3P2P1P0c0

• The carries are implemented in the block labeled carry-look ahead logic. An adder implemented in

this form is called a carry-look ahead adder.

• Limitation: If we try to extend the carry-look ahead adder for longer operands, we run into a

problem of gate fan-in constraints.

Module 5

.Give the control sequence for execution of complete instruction ADD (R3), R1.

Add (R3), R1

Step	Action
1	PC_{out} , MAR_{in} , Read, Select4,Add, Z_{in}
2	Z_{out} , PC_{in} , Y_{in} , WMF C
3	MDR_{out} , IR_{in}
4	$R3_{out}$, MAR_{in} , Read
5	$R1_{out}$, Y_{in} , WMF C
6	MDR_{out} , SelectY, Add, Z_{in}
7	Z_{out} , $R1_{in}$, End

• Instruction execution proceeds as follows:

Step1--> The instruction-fetch operation is initiated by
→ loading contents of PC into MAR &
→ sending a Read request to memory.
The Select signal is set to Select4, which causes the Mux to select constant 4. This value is added to operand at input B (PC's content), and the result is stored in Z.

Step2--> Updated value in Z is moved to PC. This completes the PC increment operation and PC will now point to next instruction.

Step3--> Fetched instruction is moved into MDR and then to IR.
The step 1 through 3 constitutes the **Fetch Phase**.
At the beginning of step 4, the instruction decoder interprets the contents of the IR. This enables the control circuitry to activate the control-signals for steps 4 through 7.
The step 4 through 7 constitutes the **Execution Phase**.

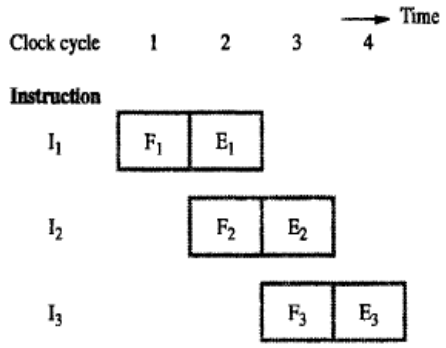
Step4--> Contents of R3 are loaded into MAR & a memory read signal is issued.

Step5--> Contents of R1 are transferred to Y to prepare for addition.

Step6--> When Read operation is completed, memory-operand is available in MDR, and the addition is performed.

Step7--> Sum is stored in Z, then transferred to R1.The End signal causes a new instruction fetch cycle to begin by returning to step1.

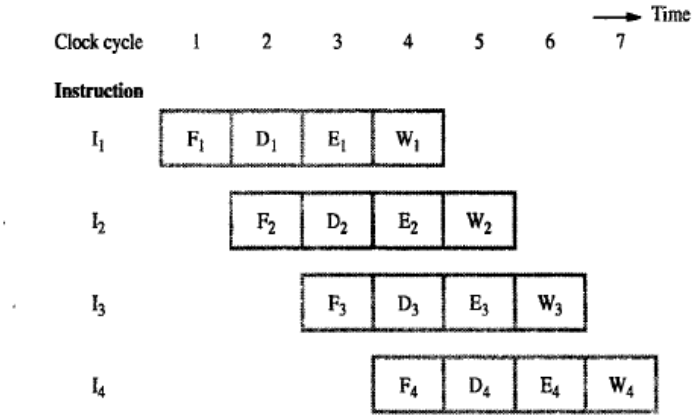
3 Explain the differences between Hardwired and Micro-programmed control.



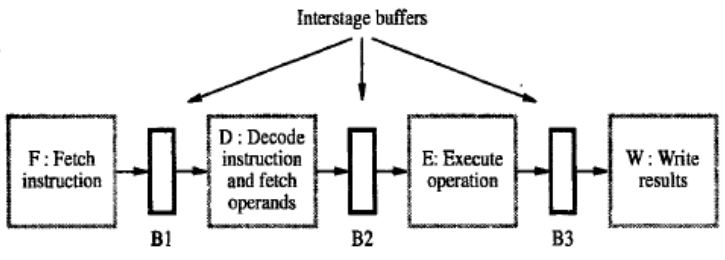
(c) Pipelined execution

Figure 8.1 Basic idea of instruction pipelining.

The computer is controlled by a clock whose period is such that the fetch and execute steps of any instruction can each be completed in one clock cycle. Operation of the computer proceeds as in Figure 8.1c. In the first clock cycle, the fetch unit fetches an instruction I_1 (step F_1) and stores it in buffer B1 at the end of the clock cycle. In the second clock cycle, the instruction fetch unit proceeds with the fetch operation for instruction I_2 (step F_2). Meanwhile, the execution unit performs the operation specified by instruction I_1, which is available to it in buffer B1 (step E_1). By the end of the



(a) Instruction execution divided into four steps



(b) Hardware organization

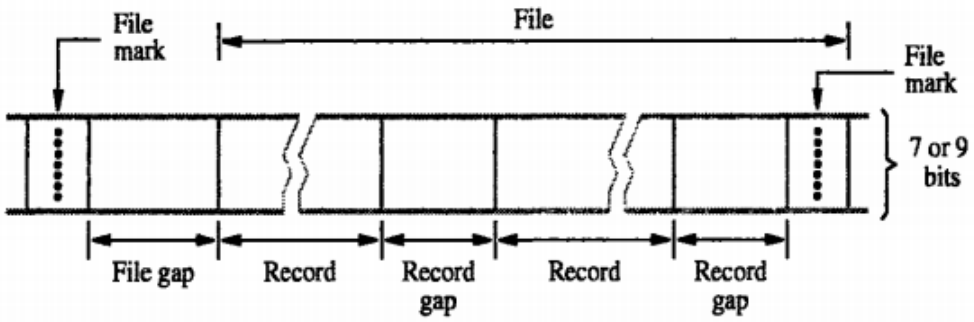


Figure 5.33 Organization of data on magnetic tape.