1 Basics of Computer Design

1.1 COMPUTER ARCHITECTURE

- It deals with the system attributes which are visible to the programmer, and these attributes have a direct impact on the execution of the program.
- Few examples of such attributes are I/O mechanism, bits required to represent data types, instruction sets, etc.

Computer organisation:

- In computer organisation we deal with the operational unit of the hardware system and its interconnections
- Examples of organisational attributes are hardware details that are hidden from the programmer.

Von–Neumann architecture:

- It is based on the concept of the same memory holding both data and instructions.
- In this architecture, we have common addresses and data buses for both CPU and Memory.
- John Von Neumann proposed this architecture in 1945.
- In this architecture, single instruction takes 2 clock cycles.
- In this architecture, CPU cannot access instructions and read/write at the same interval of time.
- This architecture is less expensive.
- It is used for personal computers and small-scale computers.

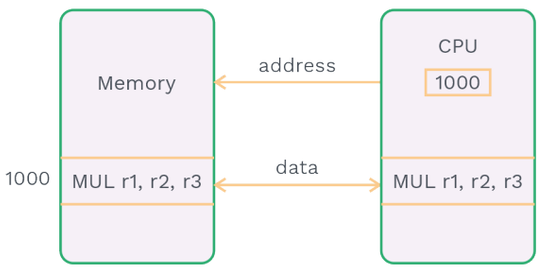


Fig. 1.1 Von–Neumann Architecture

Harvard architecture:

- It is based on the concept of distinct memory models for instructions and data.
- It has two sets of address/data buses shared by CPU and memory.
- It allows two simultaneous memory fetches.
- It allows CPU to fetch the instruction, and it can perform both read and write operations at the same time.
- This architecture is more expensive than Von Neumann's architecture.
- It is used in microcontrollers and digital signal processing (DSP) devices

such as sonar, radar, etc.

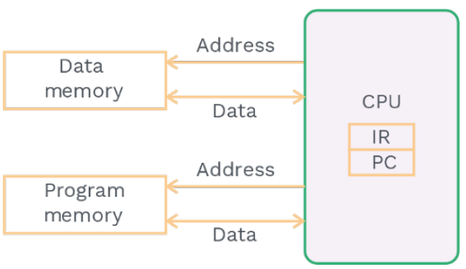


Fig. 1.2 Harvard Architecture

PRACTICE QUESTIONS

Q1 **Among the given options, choose the correct one :**

I) Von neumann architecture shares common memory for instructions and data.

II) Harvard architecture has separate physical memories for instructions and data.

a) Only I is true **b) Only II is true**

c) Both I and II are true **d) Neither I nor II is true**

Sol: **c)** Harvard architecture has two different sets of memory, one set for storing the data elements and another set for storing the code. While Von–Neumann architecture shares common memory for data and instruction.

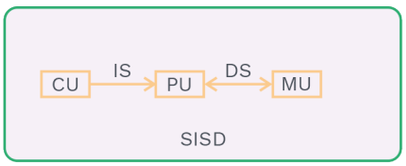
Flynn's classification:

- It is a way of organising multiple processors.
- It was first introduced by Flynn and is the most common approach for categorizing the systems with parallel processing capabilities.
- This classification has four categories of computer systems.

- Single Instruction, Single Data (SISD) stream
- Single Instruction, Multiple Data (SIMD) stream
- Multiple Instruction, Single Data (MISD) Stream
- Multiple Instruction, Multiple Data (MIMD) stream

1) Single Instruction, Single Data (SISD) stream:

- In SISD, one instruction runs on one processor on the data stored on single memory.
- Uniprocessor comes into this category.



CU : Control unit
IS : Instruction stream
PU : Processing unit
DS : Data stream
MU : Memory unit

Fig. 1.3 SISD Stream

- Machine instructions are executed in sequential order.
- Systems based on this architecture are known as sequential computers.
- Most of the conventional computers are based on this model.
- Instructions and the data must be stored in primary memory for processing.
- The rate of transferring information internally determines the speed of processing in SISD.

2) Single Instruction, Multiple Data (SIMD) stream:

- In this architecture, a single machine instruction controls simultaneous execution of a number of processing elements on lock step basis.
- Each processing element has an associated data memory for the execution of instruction on different sets of data by different processors.
- Examples of this model are vectors and arrays.

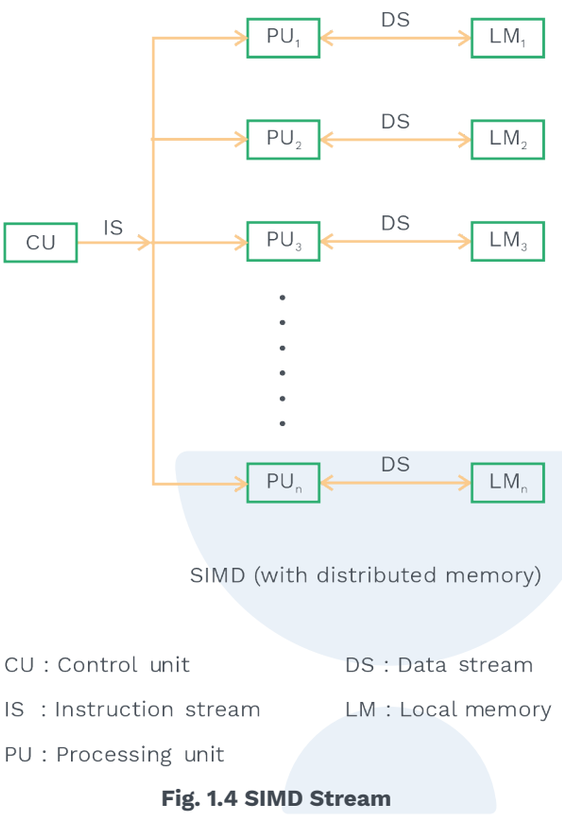


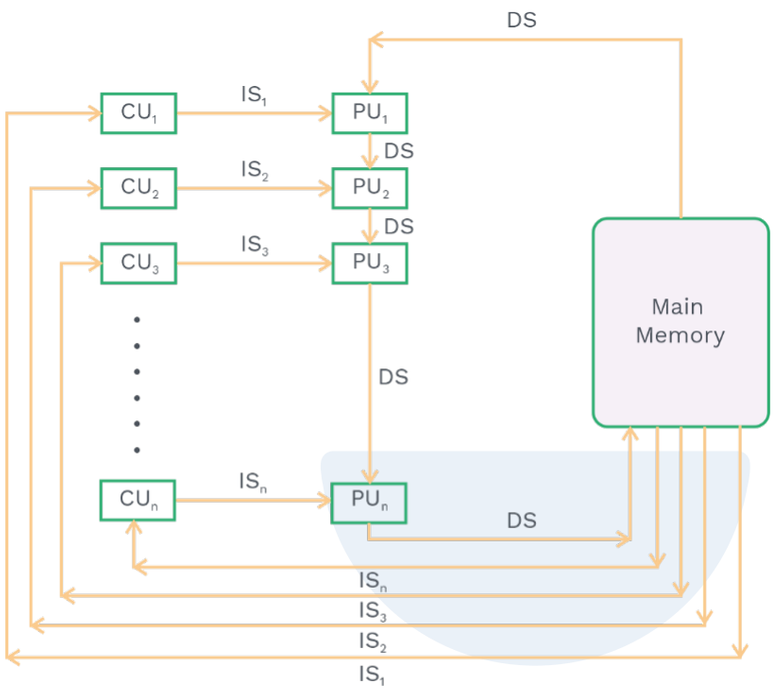
Fig. 1.4 SIMD Stream

- Scientific computing is based on SIMD architecture.
- Cray's vector processing machine is based on this architecture.
- Example of SIMD:

If we want to subtract 10 from each element of array A, it can be done by SIMD model.

3) Multiple Instruction, Single Data (MISD) stream:

- In this architecture, a sequence of data is transmitted to a set of processors, in which every processor executes a different instruction sequence.
- This architecture is not commercially implemented.



MISD architecture

CU : Control unit PU : Processing unit

IS : Instruction set DS : Data set

Fig. 1.5 MISD Stream

- Machines built using the MISD architecture are not very useful in most of the applications.
- Example of MISD is:
  A = tan(a),
  B = sec(a), a: Variable (Single data stream)
  C = cosec(a)
- Here data stream is the same, but the instruction stream is different, i.e. the CPU will first calculate the value of tan(a), second the value of sec(a) and third the value of cosec(a).

4) Multiple Instruction, Multiple Data (MIMD) stream:

- In this architecture, different instructions of different data set execute in different processors.

- Different processors take programs and data from common shared memory, and processors communicate with each other through that memory.
- Example of a shared memory MIMD system is a symmetric multiprocessor (SMP).
- In symmetric multiprocessors, multiple processors are capable of sharing single memory with the help of buses.

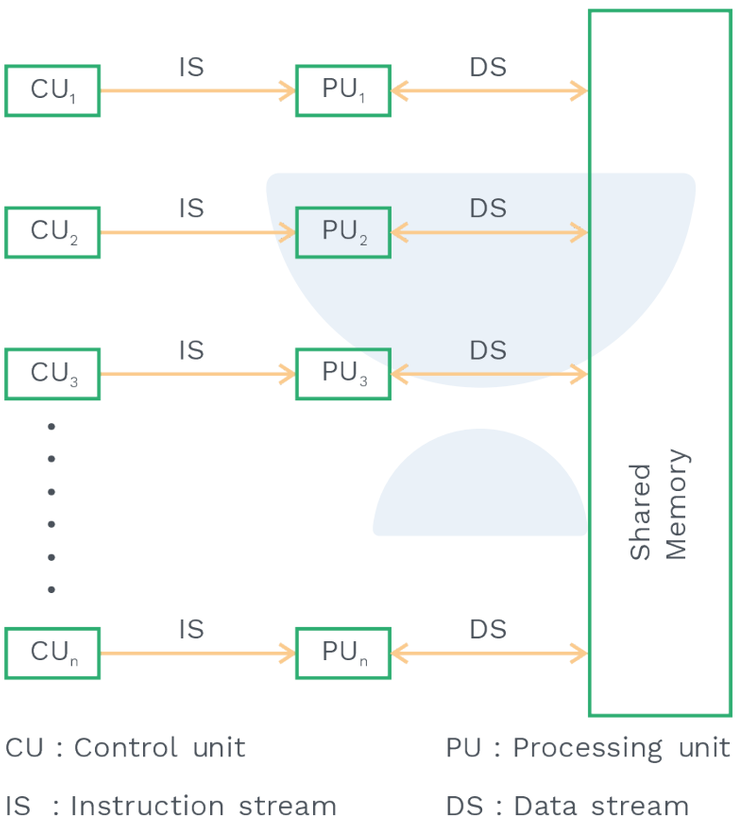
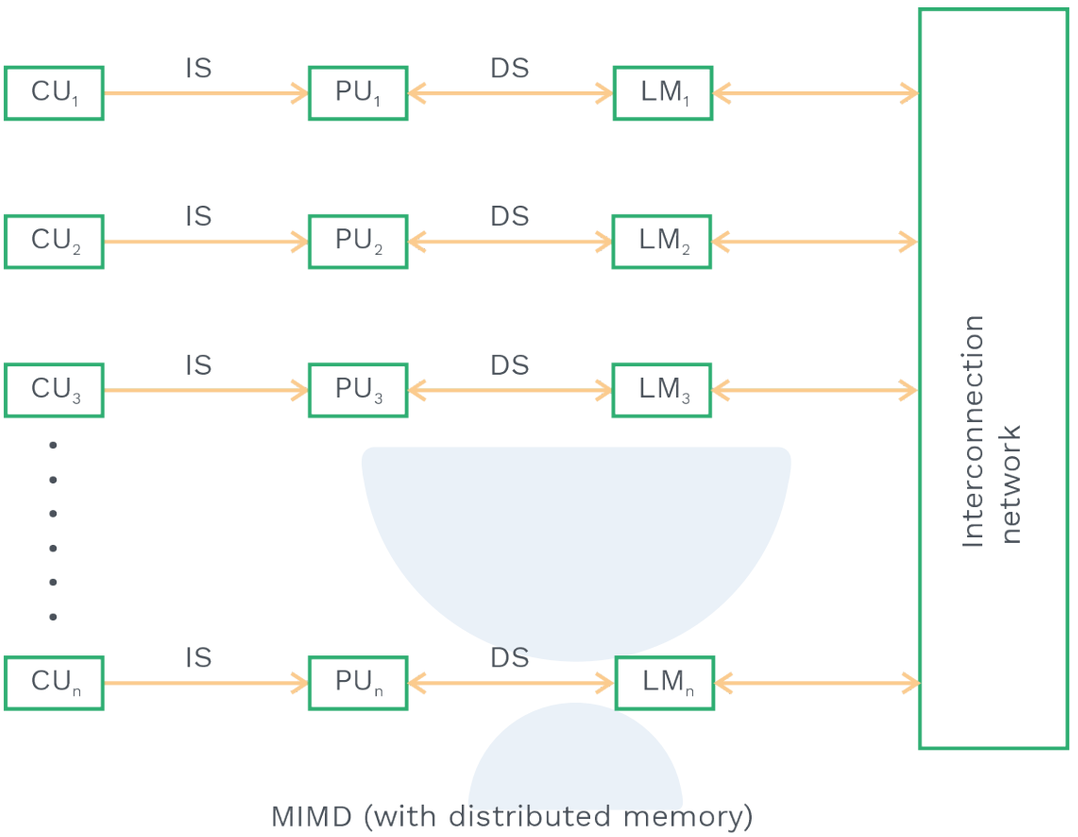


Fig. 1.6 MIMD Stream



MIMD (with distributed memory)

LM : Local memory
CU : Control unit
PU : Processing unit
IS : Instruction set
DS : Data set

Fig. 1.7 MIMD Stream with Distributed Memory

- A more recent development in shared memory MIMD is non–uniform memory access (NUMA) organisation.
- A collection of independent uniprocessors or SMPs maybe interconnected to form a cluster.

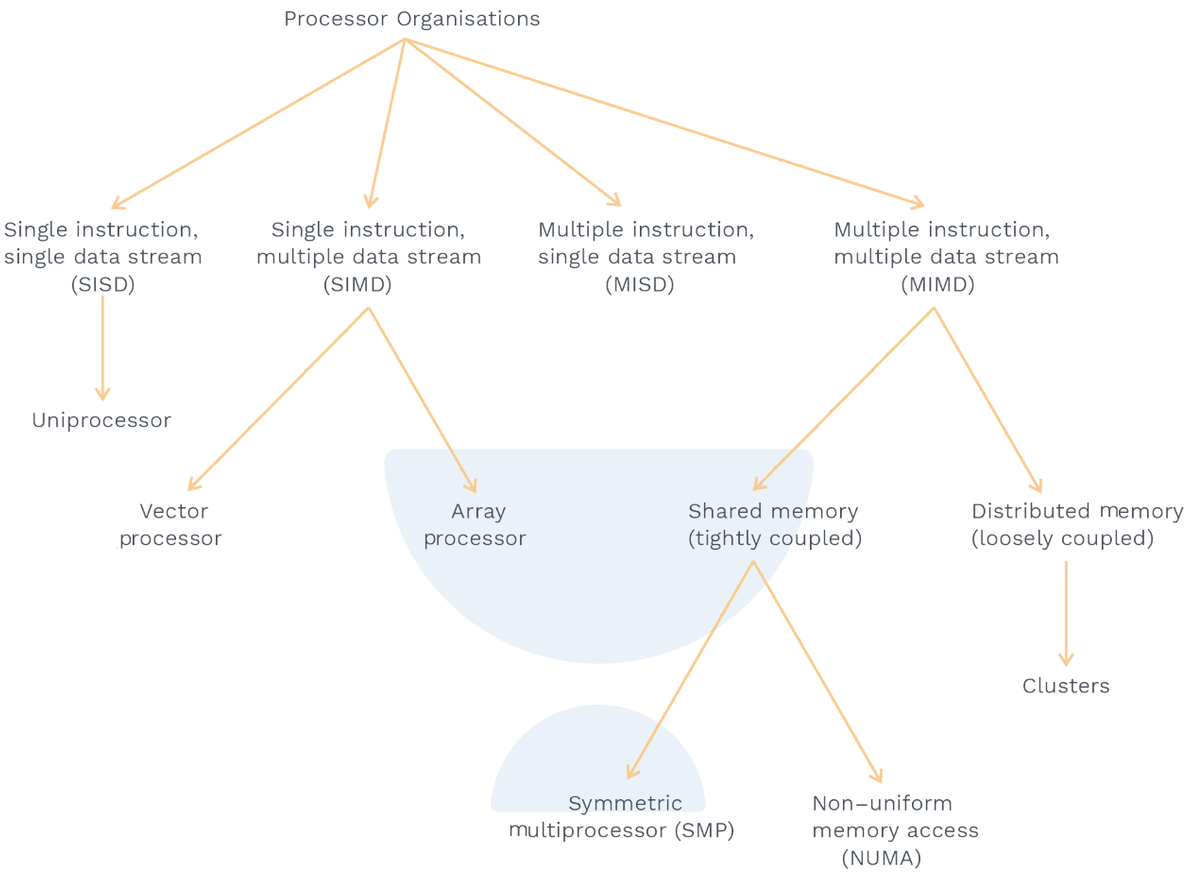


Fig. 1.8 A Taxonomy of Parallel Processor Architectures.

Symmetric multiprocessors:

- With the increase in demand for performance-enhancing and with the dropping in cost of microprocessors, designers have introduced the concept of an SMP organisation.
- Characteristics of an SMP (Symmetric Multiprocessor) are detailed below.
- There are two or more similar processors of comparable capability.
- Memory access time for all the processors will be the same as they are sharing the main memory , as well as input/output, are interconnected.
- All the processors share access to input/output devices through the same channel or through different channels, which provides paths to the same devices.
- All processors can perform the same function (so the term symmetric is used).

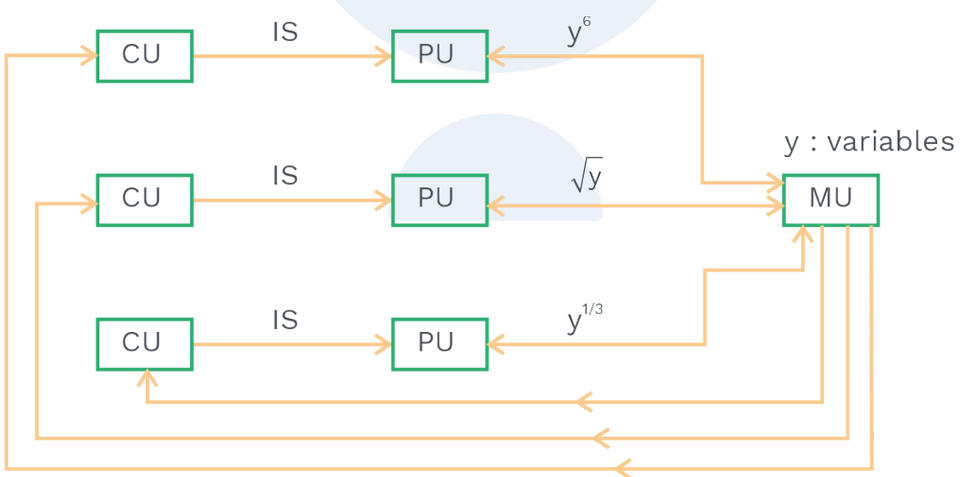
PRACTICE QUESTIONS

Q2 **Suppose we want to compute the value of $y^6, \sqrt{y}, y^{1/3}$ using the given value of y. Which of the following among Flynn's CPU classifications will be best suited for the above calculation?**

a) (SISD) stream.
b) (SIMD) stream.
c) (MISD) stream.
d) (MIMD) stream.

Sol: **c)**

MISD is useful in such operations where different computation is done on the same data set.



CU : Control unit IS : Instruction stream

PU : Processor unit MU : Memory unit

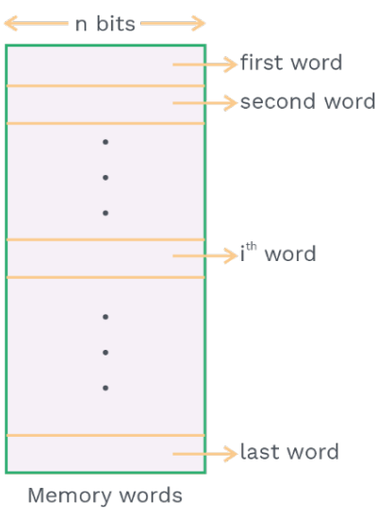
Q3 **The vector and arrays belong to which class of Flynn's classification.**
a) (SIMD) stream.
b) (SISD) stream.
c) (MISD) stream.
d) (MIMD) stream.

Sol: **a)**

The vector and arrays belong to SIMD (Single instruction and multiple data streams).
For example: If we want to decrement all the elements of an array by some constant value, it is usually performed by SIMD type of classification.

Memory locations and addresses:

- The number operands, character operands and instructions are stored in the memory.
- In memory, there are many storage cells, each of which stores the information in binary form.
- A single bit may not be sufficient to provide ample information. So, data is rarely represented by a single bit.
- A pattern of bits is capable of providing valuable information.
- Group of n bits accessed is referred to as word, and n is known as the word length.

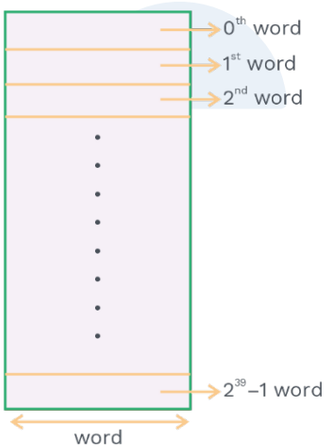


- Most modern computers have word lengths ranging from 16 to 64 bits.
- Word length of 32 bits shows that a single word can store a 32–bit 2's complement number, each occupying 8 bits.
- Size of machine instruction will be equal to one or more words.
- For accessing the information from memory(word or byte), we require unique addresses for each location.
- It is most widely used that numbers from 0 to 2^K–1 are used for representing K values.
- 2^k address space means that there are 2^k address cells, and each cell is represented with k bits number.

PRACTICE QUESTIONS

Q4 **If there are 512 G locations (words) in the memory, then find the number of bits to represent each word uniquely.**

Sol: Total number of locations (words) = 512 G



$1G = 2^{30}$

$512\ G = 512 \times 2^{30} = 2^9 \times 2^{30} = 2^{39}$

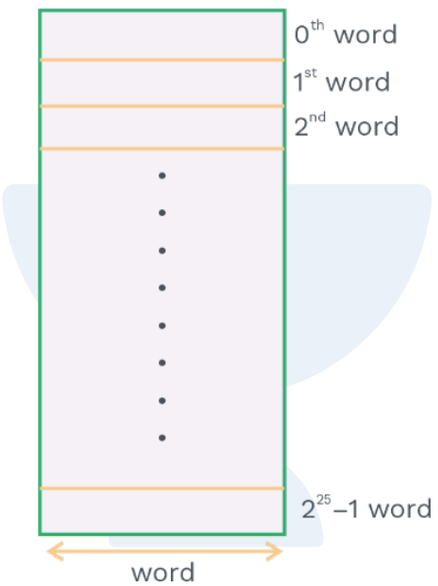
The number of bits required to represent each word uniquely:

$= \lceil \log_2 2^{39} \rceil = 39$ bits.

Q5 **If there are 25 bits available to represent each word of memory, then how many maximum number of locations (words) can be addressed uniquely?**

Sol: n = number of bits = 25

Maximum number of words that can be addressed uniquely



$= 2^{25}$ words $= 2^5 \times 2^{20}$ words

$= 32 \times 1M$ words $= 32$ M words

Q6 **If memory is byte addressable (1 word = 1 Byte) and the total size of memory is 128 KB. Calculate the number of bits required to represent each word and each byte uniquely.**

Sol: Total memory size = 128KB

Number of bits required to represent each byte = $\lceil \log_2 2^7 \times 2^{10} \rceil = \lceil \log_2 2^{17} \rceil$

= 17 bits

Since 1 word = 1 Byte

Number of bits required to represent each word = 17 bits

Q7 **If the size of main memory is 512 GB. The size of the word is 16 bits. Find the number of bits required to represent each bit uniquely if the memory is word addressable.**

Sol: Size of word = 16 bits

8 bits = 1 Byte

16 bits = 2 Byte

1 word = 2 Byte

Total number of words = $\frac{512\,GB}{2B}$ = 256 G words

1 word = 16 bits

256 G words = 256 G × 16 bits = $2^8 \times 2^{30} \times 2^4 = 2^{42}$ bits

So, number of bits required to represent each bit uniquely inside memory

= $\lceil \log_2 2^{42} \rceil$ = 42 bits

Q8 **Memory size = 128 Gb**
word size = 16 bits
Give the memory capacity in words, bytes and bits and also give the number of bits required for addressing, respectively.

Sol:

Memory capacity	Addressability
Bit = 128 Gb	$2^7 \times 2^{30} = 2^{37}$ $\boxed{n = 37}$
Byte = 16 GB	$2^4 \times 2^{30} = 2^{34}$ $\boxed{n = 34}$
Word = 8 GW	$2^3 \times 2^{30} = 2^{33}$ $\boxed{n = 33}$

n = number of bits required for addressing respectively.

RAM chips and organization:

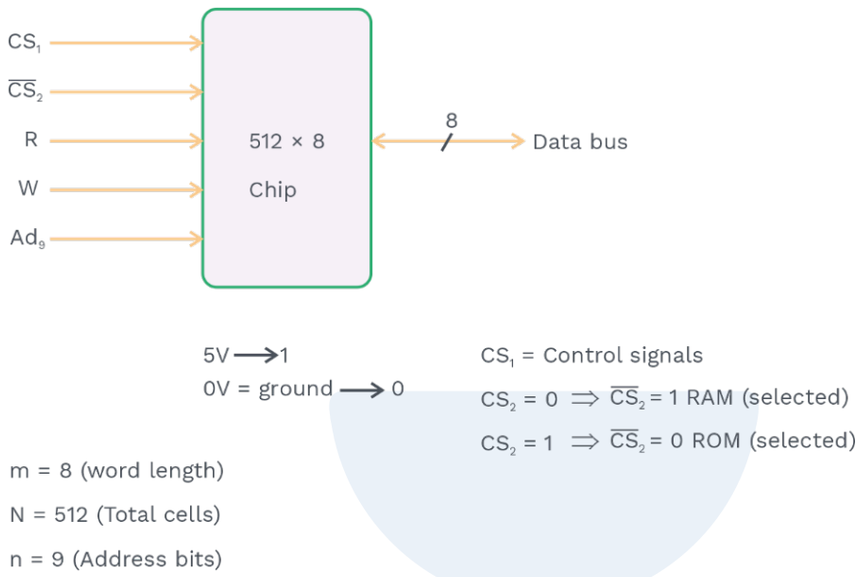


Fig. 1.9

Building 1024 B using 256 B chips:

$$\frac{1024\,B}{256\,B} = 4$$

N = 1024 W (1W = 1B)

n = $\lceil \log_2 1024 \rceil$ = 10-bit address

00 – select first RAM chip
01 – select second RAM chip
10 – select third RAM chip
11 – select fourth RAM chip
R = Read signal (it is active high)
W = Write signal (it is active high)
Ad = Address (8 bits) = Each chip has 256 words.
CS_1 = Chip select, it will be done by decoder as shown above. It is active high.
$\overline{CS_2}$ = It is active low and is connected with G (ground), which is always 0. Hence 0 input is given to CS_2, so $\overline{CS_2}$ =1. Hence all the chips will be working as RAM and not ROM.

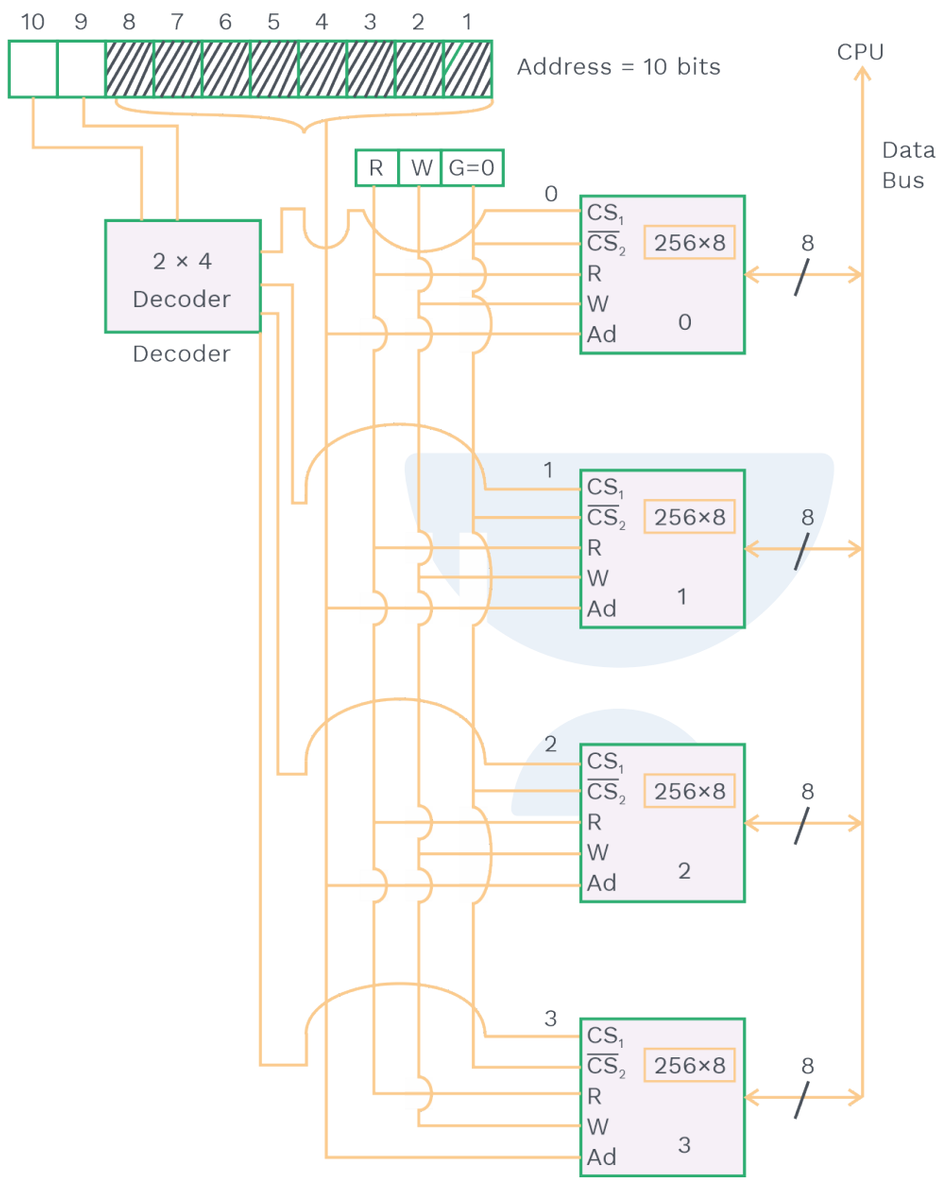


Fig. 1.10

For example, Address=1000101011, R=1, W=0, G=0.
It means CPU wants to read the data from memory. If the two MSB bits are 10, they will select the 2nd RAM chip. G = 0, which means the chip will act like RAM. 8 LSB bits will signify the word number (43rd word from 2nd chip). 43rd word (8 bit) will be kept on the data bus, and CPU will read it.

PRACTICE QUESTIONS

Q9 **How many 256 MB RAM chips are needed to build 2GB RAM?**

Sol: Number of RAM chips required $= \frac{2\,GB}{256\,MB} = \frac{2 \times 2^{30}}{2^8 \times 2^{20}} = \frac{2^{31}}{2^{28}}$

$= 2^3 = \boxed{8\,chips}$

Q10 **How many 128 KB RAM chips are required to build 256 MB of RAM?**

Sol: Total capacity RAM = 256 MB

Size of RAM chip available = 128 KB

Number of RAM chips required

$= \frac{256\,MB}{128\,KB} = \frac{2^8 \times 2^{20}}{2^7 \times 2^{10}}$

$= 2^{11}$

= 2048 chips

Building larger RAMs using smaller RAM chips:

⇒ There are broadly two types of strategies:

1) Vertical increasing (constant word size):

64 MB RAM using 128 × 8 chips

Each word = 8 bits

(Here 1W = 8 bits = 1 Byte)

Number of chips required

$= \frac{64\,MB}{128\,W} = \frac{64\,MW}{128\,W} = \frac{2^{26}}{2^7} = 2^{19}$

$= 2^{19}$ chips

= 512 K chips

Decoder size = $\boxed{19 \times 2^{19}}$

Address = 19 + 7 = 26 bit

19 → To decoder

7 → To each chip from 2^7

PRACTICE QUESTIONS

Q11 **How many 64K × 8 RAM chips are required to build 4 MB of RAM chip? Also, calculate the size of the decoder required. Assume memory is Byte addressable.**

Sol: Number of chips required

$$= \frac{\text{Total Required Size of RAM}}{\text{Available RAM Size}}$$

$$= \frac{4\,\text{MB}}{64\,\text{K} \times 8} \quad \text{(Here, we assume 1 word = 1 Byte)}$$

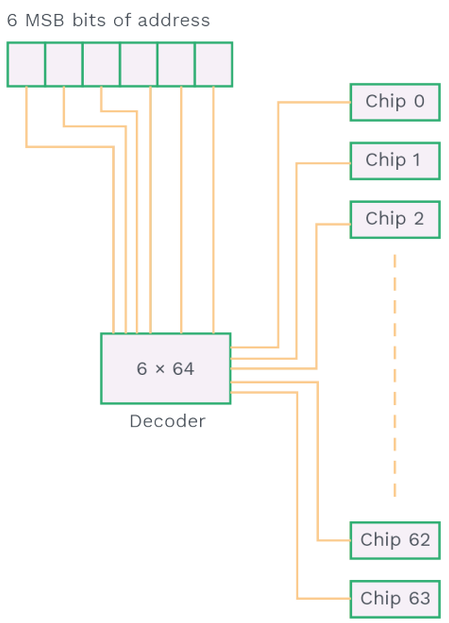
$$= \frac{4\,\text{MB}}{64\,\text{KW}} = \frac{4\,\text{MW}}{64\,\text{KW}} = \frac{2^2 \times 2^{20}}{2^6 \times 2^{10}}$$

$$= = \frac{2^{22}}{2^{16}} = 2^6 = 64\,\text{chips are used}$$

Here we are using the concept of vertical increasing, i.e., word size will remain constant.

Hence $6 \times 2^6 =$ $\boxed{6 \times 64}$ size decoder will be required.

* At a single time, only 1 chip will be activated by the decoder, the rest all will remain deactivated.



Each chip has an enable line which activates the chip. We need to select one of 64 chips so a $\boxed{6 \times 64}$ size decoder will be required for this purpose.

Q12 **Calculate the size of the decoder and the number of chips required to build a RAM of 32 Gb from 256 K × 8 RAM chip. Also, give the diagrammatic representation of the decoder and RAM chips. Assume Byte addressable memory.**

Sol: Given, the size of RAM chips = 256 K × 8

Required size of larger RAM chip = 32 Gb

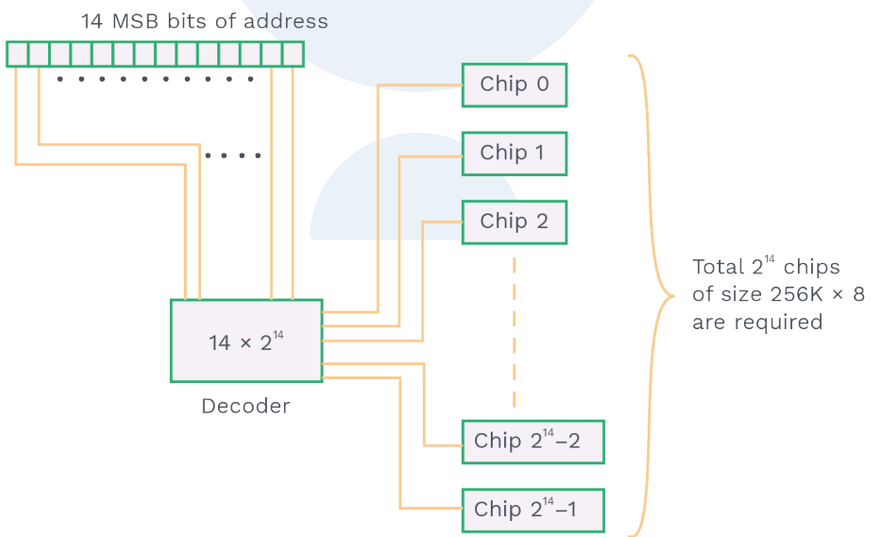
Number of smaller chips required

$$= \frac{32\,\text{Gb}}{256\,\text{K} \times 8} = \frac{2^5 \times 2^{30}}{2^8 \times 2^{10} \times 2^3} = \frac{2^{35}}{2^{21}}$$

$= 2^{14}$ chips

Here, again we are using the concept of vertical increasing, i.e., word size will remain constant.

So $\boxed{14 \times 2^{14}}$ size decoder will be required.



- Each chip has an enable line which activates the chip. We need to select one of 2^{14} chips. Hence a decoder of size $\boxed{14 \times 2^{14}}$ will be required for this purpose.
- As we are using a decoder, at a single time, only 1 chip will be activated by the decoder, the rest will remain deactivated.

2) Horizontal Increasing (change in word size):

- Here, the word size will be changed

For example: we want to build a 256 B RAM using 256 × 1 RAM.

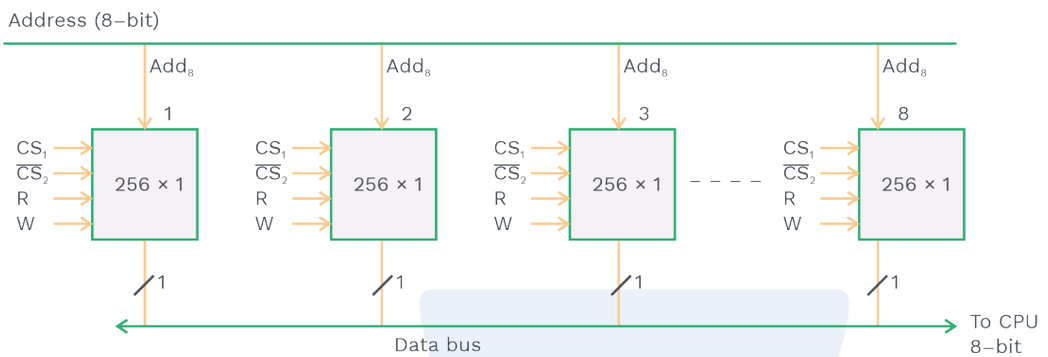


Fig. 1.11

- Here, all the chips are activated simultaneously.
- We are assuming that our required chip has a word size of 1 byte and the given chip has a word size of 1 bit.

Required chip size = 256 B (1W = 1B)
Given chip size = 256 × 1 (1W = 1bit)
Number of chips (given) required

$$= \frac{256\,B}{256 \times 1} = 8\,\text{chips}$$

- Each smaller chip has 256 memory locations of 1 bit each. But we are required to give 1 Byte of data (8 bits) to the CPU through the data bus.
- Each chip will give 1 bit each, so we have to concatenate all the bits from 8 chips, i.e., 1 bit each from 8 chips and concatenated 8 bits (1 Byte) will be given to the data bus.
- All 8 chips will be activated simultaneously to serve the purpose of sending 8 bits (1 Byte).
- As there are 256 locations in each smaller chip, an address line of 8 bits is required and sent to each smaller chip.
- Each chip has CS_1, $\overline{CS_2}$, R(Read), W(Write) pins which do their functions the same as in vertical increasing case.

PRACTICE QUESTIONS

Q13 **We are given a 128 × 1 RAM chip. Designers want to build a bigger RAM chip of size 128 B. How many RAM chips of size 128 × 1 will be required to get the larger RAM chip? Also, give a diagrammatical representation of arrangements of chips.**

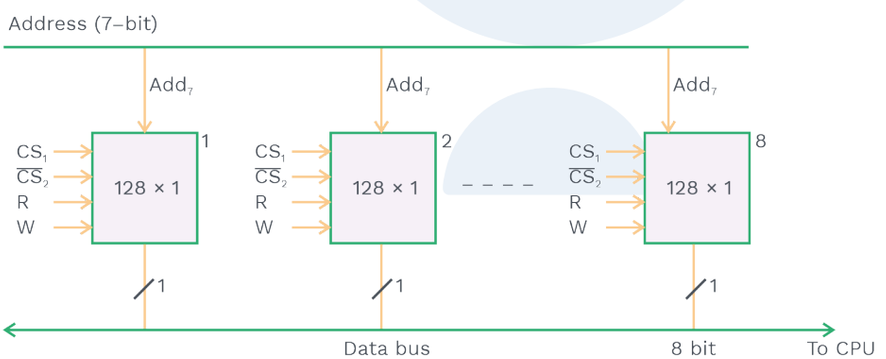
Sol: Required RAM size = 128 B

Given RAM size = 128 × 1

In given RAM size, 1 word = 1 Bit

In required RAM chip, 1 word = 8 Bit

$$\text{Number of chips required} = \frac{128\,\text{B}}{128 \times 1} = 8 \text{ chips}$$



- Here again, we are using horizontal increasing, where word size will be changed.
- All 8 bits (1 bit from each chip) will be merged together and sent to the data bus.
- Each smaller chip has 128 locations (cells), so a 7-bit address is required to uniquely address each location of the respective chips.
- Address pin Add_7 will be activated simultaneously for all the chips so that each chip can give 1 bit each.

3) Vertical + Horizontal increasing:

- In the real world, we end up using vertical + horizontal increasing to serve the purpose of practical needs.

For example: We are required to build 64 KB RAM using 512 × 1 RAM chips.

- Here 1 word = 1 Byte (in our required chip)

Number of smaller chips required

$$= \frac{64\,KB}{512 \times 1} = \frac{2^6 \times 2^{10} \times 2^3}{2^9}$$

$$= \frac{2^{19}}{2^9} = 2^{10} = 1024 \text{ chips}$$

- Total, we need 1024 chips, but here these chips will be arranged both in a horizontal and vertical manner.
- Given smaller chip gives only 1 bit, but we are required 8 bits (1 Byte) in larger RAM chip, so here horizontal increasing is required.

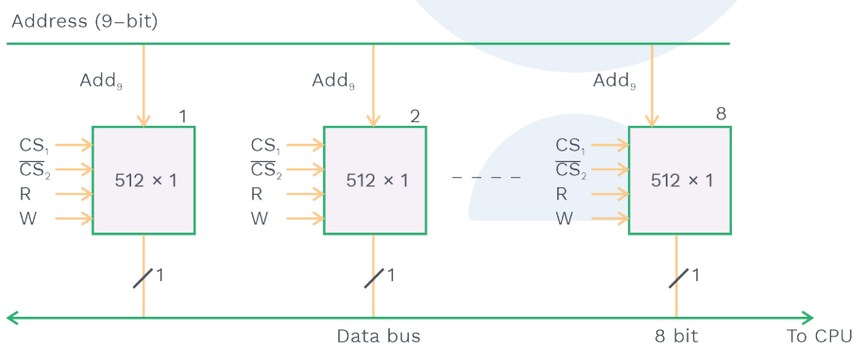


Fig. 1.12 Horizontal Part

- Now, this is only 1 chunk of horizontal arrangement of chips; it will give 8 bits in total by merging, as we have seen earlier in the horizontal case.
- Total, we require 1024 (2^{10}) chips, but out of these, 8 chips per horizontal block are required to get the desired word length.
- $\frac{1024}{8} = \frac{2^{10}}{2^3} = \boxed{2^7}$, this will be the total number of horizontal block sets which will be arranged vertically.

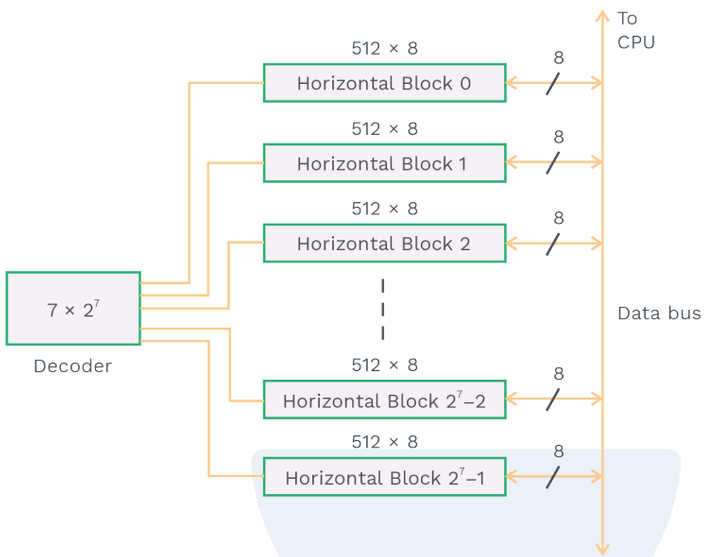


Fig. 1.13 Vertical Arrangement of Horizontal Blocks

- Here, the address will be divided into 2 parts, the first one is the decoder and the second one is the address bits for each horizontal chunk of chips.

$$\underbrace{7 \text{ bits}}_{\text{To decoder}} + \underbrace{9 \text{ bits}}_{\text{To Horizontal Block}} = 16 \text{ bits (Address)}$$

- In total, we have 16 bits, which will give a total RAM size of 2^{16} = 64 KB, where 1 word = 1 Byte.
- As there are a total of 2^7 horizontal blocks, we need to select one of 2^7 blocks; it will be done by the decoder of size $\boxed{7 \times 2^7}$.
- After selecting one of 2^7 blocks, the horizontal arrangement comes into play; a 9 bit address will be given to each chip inside the selected horizontal block.
- As there are 512 locations, each of size 1 bit, one location of each chip will be selected, so total 8 bits will be put on the data bus.
- In this way, this hybridisation (vertical + horizontal) concept will be implemented, which is used in most practical applications.

PRACTICE QUESTIONS

Q14 **We are given 64 K × 1 chips of RAM. Our aim is to build a larger RAM chip of size 512 KB. In larger RAM chip 1 word = 1 Byte. Calculate the total number of chips required and the size of the decoder. Also, give the diagrammatic arrangements of chips.**

Sol: Here 1 word = 1 Byte (in a larger RAM chip)

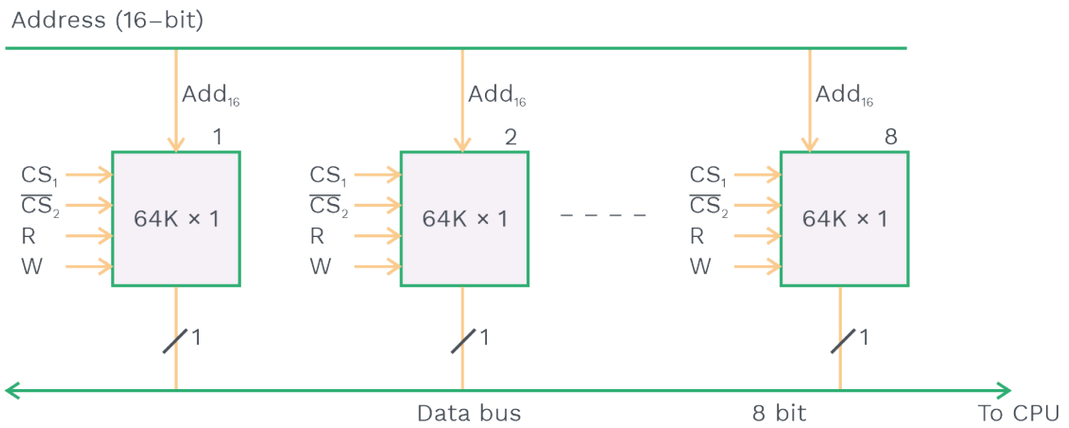
Number of smaller chips required

$$= \frac{512\,KB}{64\,K \times 1}$$

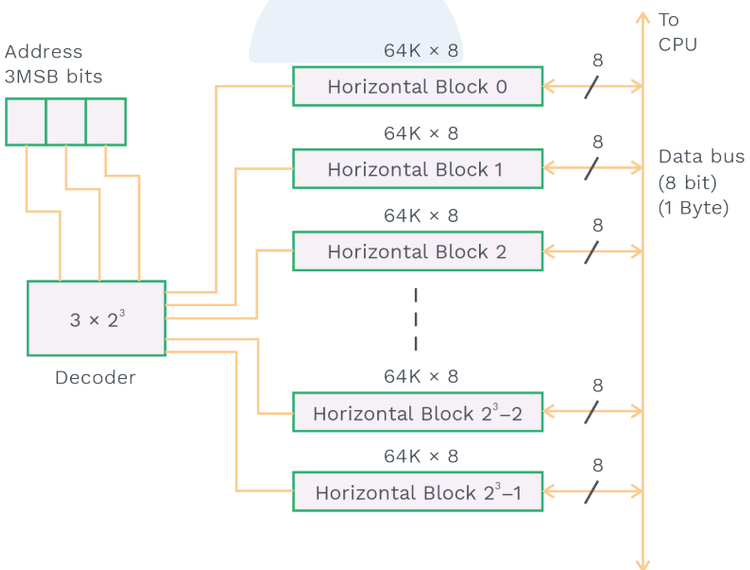
$$= \frac{2^9 \times 2^{10} \times 2^3}{2^6 \times 2^{10}}$$

$$= \frac{2^{22}}{2^{16}} = 2^6 = 64 \text{ chips}$$

- Total, we need 64 chips, but these chips will be arranged both in the horizontal and vertical ways.
- Smaller chips give only 1 bit, but we are required to get 8 bits (1 Byte) in our target chip. So, both horizontal and vertical increasing is required.
- In total, we require 64 (2^6) chips, but out of these 64 chips, 8 chips will be required per horizontal block.
- $\frac{64}{8} = 8 = \boxed{2^3}$ horizontal blocks will be placed vertically, one after the other and each horizontal block will contain 8 chips.
- As there are 2^3 horizontal blocks, we require a decoder of size $\boxed{3 \times 8}$ to select one of these horizontal blocks.
- Address will be divided into 2 parts; the first part goes to the decoder and the second part goes to each smaller chip inside the horizontal block.
- 3 bits = To decoder
  16 bits = To each chip in the horizontal block.



- This is only 1 chunk (Block) of horizontal arrangement of chips; it will give 8 bits (1 Byte) in total by concatenating each bit.
- There is a total of 2^3 horizontal blocks; out of these, we need to select one; this task will be accomplished by the decoder of size $\boxed{3 \times 2^3}$.
- After selection of one of 2^3 blocks, horizontally arranged chips will be given an address of 16 bits (to each chip).



Diagrammatical representation
of vertical + horizontal scheme

- Total, we have 19 bits, so a total of 2^{19} = 512 KB memory will be available.

Previous Years' Question

How many 32 K × 1 RAM chips are needed to provide a memory capacity of 256 K bytes?

a) 8 **b)** 32 **c)** 64 **d)** 128

Sol: c) **(GATE CS-2009)**

Previous Years' Question

A RAM chip has a capacity of 1024 words of eight bits each (1K × 8). The number of 2 × 4 decoders with enable input line needed to construct a 16K × 16 RAM from 1K × 8 RAM is

a) 4 **b) 5** **c)** 6 **d)** 7

Sol: b) **(GATE CSE-2013)**

Big Endian and Little Endian:

- Byte addresses can be allocated in two different ways across the words.
- The first one is the big endian in which the most significant byte(leftmost side) of the word is stored at lower byte address.
- Another one is the little endian, which has opposite ordering. In this the least significant byte(rightmost side)of the word is stored at lower byte address.
- The bit present in the position of the highest and lowest powers of 2 are used as the most significant and the least significant bits, respectively.
- Given below is the diagrammatic representation of big–endian.

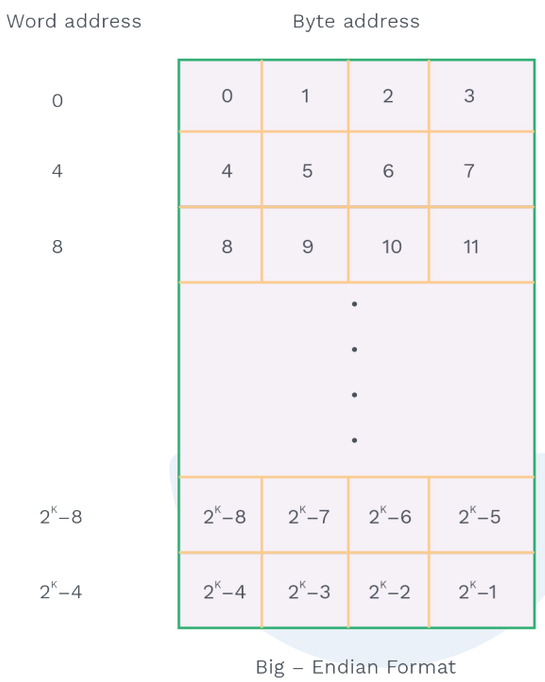


Fig. 1.14

- Both the methods little endian and big endian is used in many commercial machines.
- The byte addresses 0,4,8,... are considered as the address of the words in succession. These byte addresses are used when a specific word is read or written from memory.
- Given below is a signed integer.

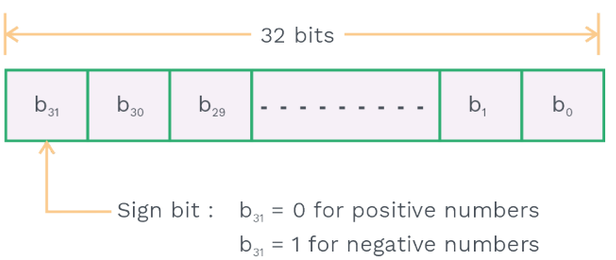


Fig. 1.15

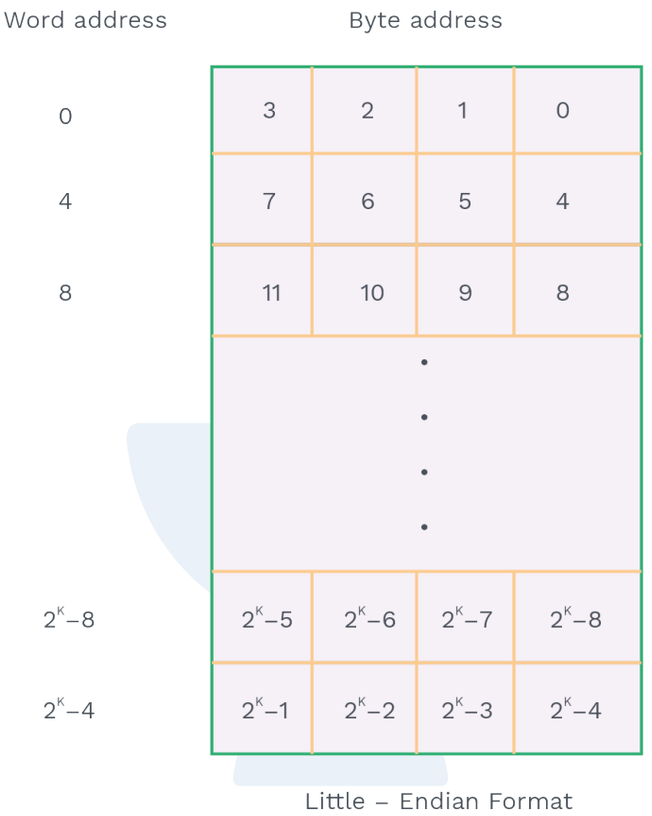


Fig. 1.16

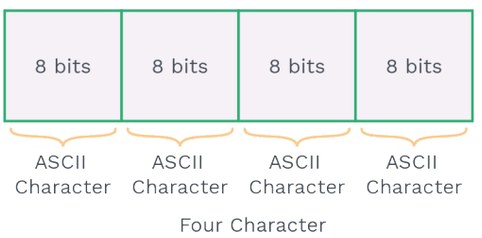


Fig. 1.17

- This is the example of encoded information in a 32-bit word.

PRACTICE QUESTIONS

Q15 **A CPU write the 'register content' into the memory in a big–endian mode. When the same content is read back from memory into a register, the CPU reads it in little–endian mode.**
Now choose the correct statement considering the above scenario.

a) The new content in the register will be different from what was written.
b) The contents will be the same
c) The content may be the same or different
d) Endianness need not be the same while reading and writing the contents into/from memory.

Sol: **a)**

In little–endian representation, the last byte of the binary number of multiple bytes data types is stored first.
In Big–endian representation, the first byte of the binary number of multiple data types is stored first.
Hence the new content in the register will be different from what was written.

Q16 **We are given two bytes of data as 10101100 00110011. How it will be stored in little–endian format?**

a)

0×100	0×101
00110011	10101100

b)

0×100	0×101
10101100	00110011

c)

0×100	0×101
10100011	11000011

d)

0×100	0×101
11000011	10100011

Sol: **a)**

In little–endian format, last byte of binary representation is stored first.

0×100	0×101
00110011	10101100

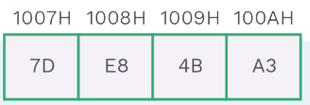
Q17 **A hexadecimal number A34BE87D is stored from memory address 1007H in little endian format. What will be the byte stored in memory location 1007H, 1008H, 1009H and 100AH?**

a) A3, 4B, E8, 7D **b) D7, 8E, B4, 3A**

c) A3, 4B, D7, 8E **d) 7D, E8, 4B, A3**

Sol: **d)** As we know in little–endian representation, the last byte of the binary number of multiple byte data type is stored first.

So hexadecimal number A34BE87D will be stored as:



Computer registers:

- Registers are required to store the operands, the result of the intermediate computations and also the addresses of the operands.
- The data register (DR) has the operand read from memory.
- The accumulator (AC) register is a general-purpose register.
- After fetching an instruction from memory, it is stored in the instruction register (IR).
- For storing the temporary data during the execution of an instruction, we use temporary registers(TR).

Special registers Symbol	Number of bits	Register name	Function
DR	16	Data register	Holds memory operand
AR	12	Address register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction register	Holds current instruction code
PC	12	Program counter	Holds address of next instruction

TR	16	Temporary register	Holds temporary data
INPR	8	Input register	Holds input character
OUTR	8	Output register	holds output character

List of registers for the basic computer:

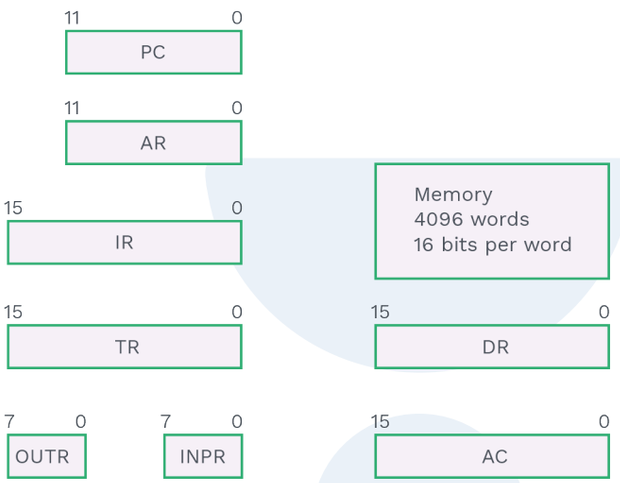


Fig. 1.18 Basic Computer Registers and Memory

- As there are 4096 words in the memory, therefore, the memory address register(AR) has 12 bits.
- Program counter is the register that holds the address of the instruction that is to be executed next.
- The PC goes through a sequence of counts, and it reads the instruction from memory sequentially.
- Instructions are read and executed sequentially unless any branch instruction comes across.
- A branch instruction jumps to a non-consecutive instruction in the program.
- The target address is held by the program counter (PC).
- There are mainly two registers : Input registers hold 8-bit data supplied by the input device. Output registers hold 8-bit data to output to an output device.

PRACTICE QUESTIONS

Q18 **Which of the following register is connected to system bus?**

a) PC (Program Counter) **b) MAR (Memory Address Register)**

c) IR (Instruction Register) **d) Accumulator**

Sol: **b)** MAR is connected to the system bus to access the memory. As we can see in the diagram given below.

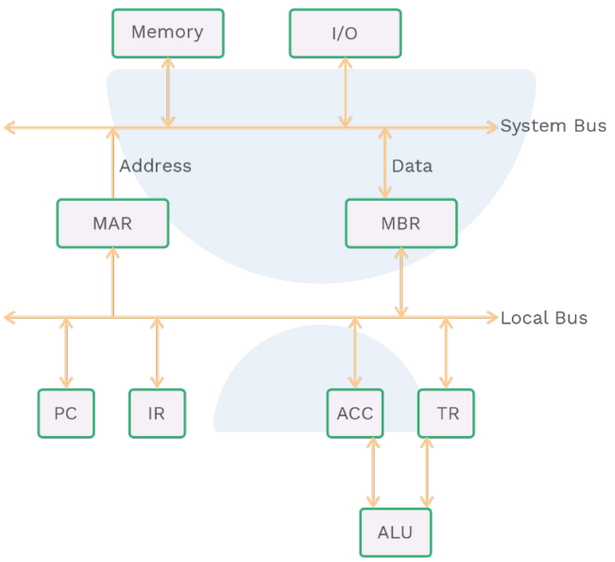


Fig. 1.20

Q19 **Which register always points to the next instruction to be executed?**

a) PC (Program Counter) **b) MAR (Memory Address Register)**

c) MDR (Memory Data Register) **d) IR (Instruction Register)**

Sol: **a)**

Program counter is the register that contains the address of the instruction which is to be executed next.

Q20 **Which register contains the most recently fetched instruction?**
a) MDR (Memory Data Register) **b) IR (Instruction Register)**
c) TR (Temporary Register) **d) MAR (Memory Address Register)**

Sol: **b)**

IR (Instruction Register) contains the instruction which is most recently fetched.

Q21 **Which register contains the address of the main memory location from where instruction has to be fetched?**
a) PC (Program Counter) **b) MAR (Memory Address Register)**
c) MDR (Memory Data Register) **d) IR (Instruction Register)**

Sol: **b)**

MAR signifies memory address register. It holds the address of the memory data to be fetched. It is connected to the address bus.

Q22 **Which register contains the contents found at the address held in the MAR?**
a) MDR **b) TR** **c) PC** **d) MAR**

Sol: **a)**

Memory Data Register (MDR) holds the contents found at the address held in the MAR. MDR content is directly connected to the data bus.

Common bus system:

- A basic computer design comprises a memory, a control unit and 8 special registers.
- The common bus acts as the connecting link among these modules. It is responsible for the transmission of information among these components.
- The connection of the registers and memory in a basic computer to a common bus system is explained in the diagram on the next page.
- The outputs of 7 registers and the memory are connected to the common bus.
- To select the specific output for bus lines at any time is determined from binary value of the selection variables S_0, S_1 and S_2.
- Each output refers to a number that represents a decimal number for given binary number.
- For example: DR holds S_2 S_1 S_0 = 011.
- The common bus lines are linked with the inputs of the memory and the registers.
- That register will only receive the data whose LD(Load) input enabled from the bus architecture in the next clock cycle.
- The bus content is transmitted to the memory when the Write Signal is set/Active.
- The 16-bit output content is placed from the memory to the bus when the Read Signal is set, and S_2 S_1 S_0=111.
- The registers-DR, IR, AC and TR are 16 bits each.
- The registers- PC and AR store 12-bit memory addresses.
- When the common bus caries memory addresses(12-bits), the initial four significant bits are set to 0's.
- After receiving the information from the bus, address register and the program counter transfers 12 least significant bits to register.

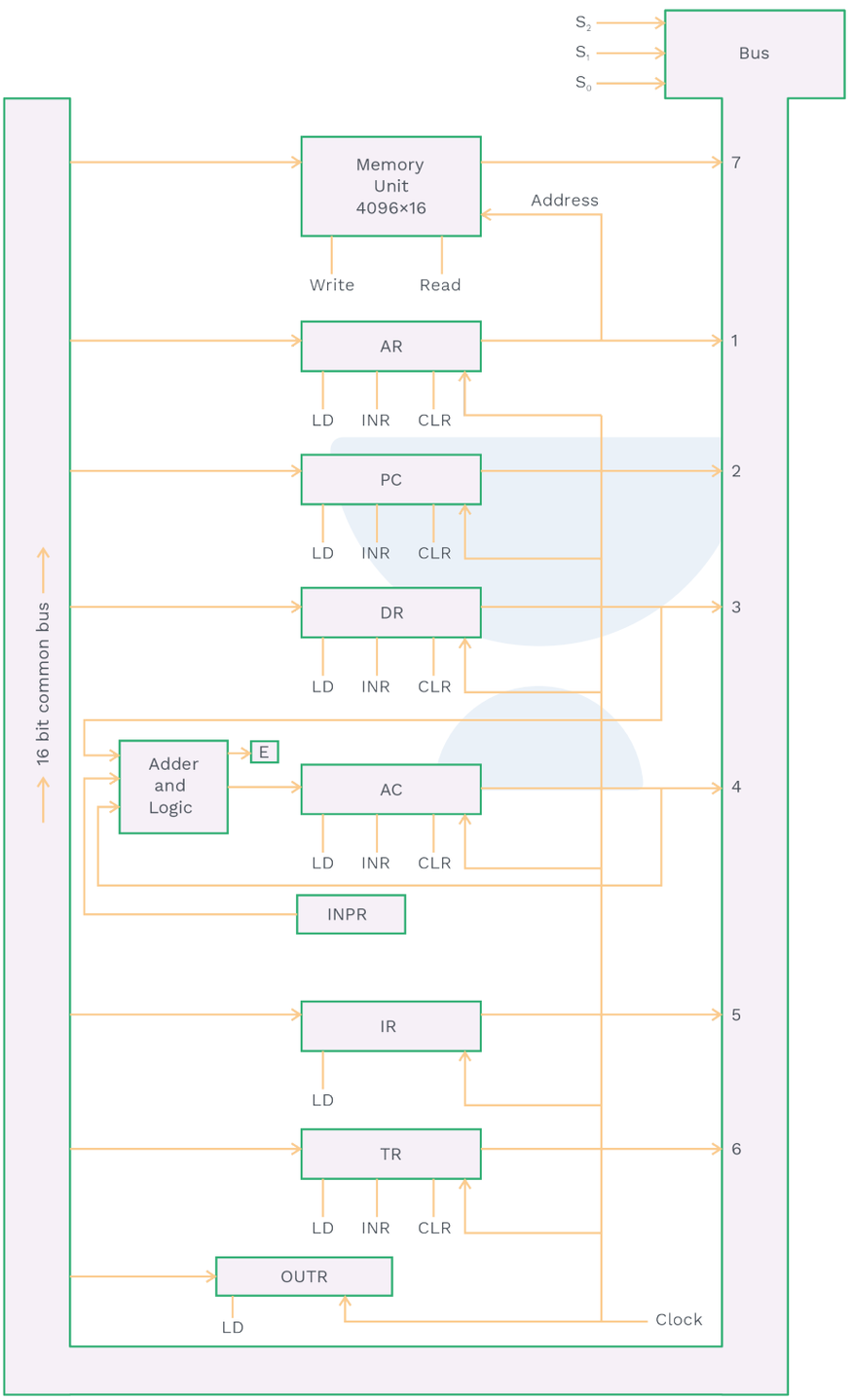


Fig. 1.20 Diagram of Basic Computer Registers Connected to a Common Bus

- INPR and OUTR refer to input and output registers.
- INPR is responsible for loading information to the bus whereas OUTR is responsible for taking data from bus.
- INPR takes data from bus and transfers it to the accumulator, and OUTR takes data from the accumulator and transfers it to other output devices.
- The bus lines are connected to the inputs of six registers and the memory.
- There are basic signals which are: LD (Load), INR (Increment) and CLR (Clear).

Chapter Summary

- Computer architecture refers to those attributes of a system which are visible to a programmer.
- Computer organization deals with the functionality of the hardware units and their connections.
- Types of Architecture
  1) **Von–Neumann architecture:**
     Based on the concept of the same memory, holding both data and instructions.
  2) **Harvard architecture:**
     Based on the concept of separate memories for data and instructions.
- Flynn's Classification
  1) Single Instruction, Single Data (SISD) stream
  2) Single Instruction, Multiple Data (SIMD) stream
  3) Multiple Instruction, Single Data (MISD) stream
  4) Multiple Instruction, Multiple Data (MIMD) stream

1) **Shared memory:**
   i) Symmetric multiprocessor (SMP)
   ii) Non–uniform memory access (NUMA)

2) **Distributed memory:**
   i) Clusters

- RAM chip organization:
  1) Vertical increasing (constant word size)
  2) Horizontal increasing (change in word size)
  3) Vertical + Horizontal increasing
- Memory locations and addresses

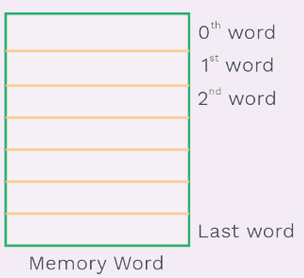


Fig. 1.21

- Byte ordering conventions:
  1) Big Endian
  2) Little Endian
- Computer Registers
  1) Data Register
  2) Address Register
  3) Accumulator
  4) Instruction Register
  5) Program Counter
  6) Temporary Register
  7) Input Register
  8) Output Register