

b)

2) The addressing modes help us specify the way in which an operand's effective address is represented in any given instruction. The addressing modes describe an efficient and flexible way to define complex effective addresses.

In indexed addressing mode, the content of a given index register gets added to an instruction's address part so as to obtain the effective address. Here, the index register refers to the special CPU register that consists of an index value. It is pretty helpful whenever the instructions in a program access an array or larger range of memory addresses. It can be denoted as follows:

$X(R)$

The effective address here is denoted as follows:

$$EA = X + (R)$$

Using this mode, we get flexibility for specifying several locations of memory. However, it is very complex to implement the index addressing mode.

e.g.: Consider the instruction below:

load(R₂) JA

load(R₃), (R₂)

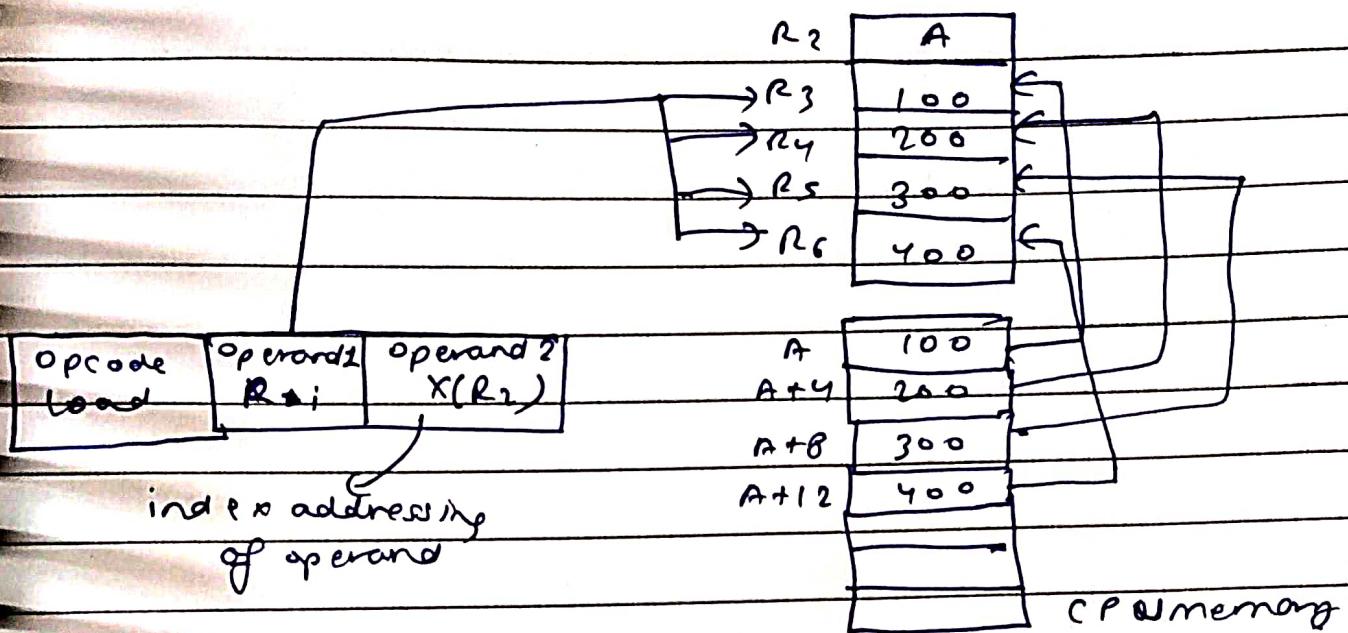
load(R₄), 4(R₂)

load R₅, 8(R₂)

load R₆, 12(R₂)

These instructions will load R₂, R₄, R₅, R₆ register along with the contents that are present at the successive memory addresses correspondingly from memory location A.

Date



Date

a) \Rightarrow if number of registers = 2^n , n = no. of selection
input lines

Here, number of registers = 16 = 2^4

Therefore, $n = 4$.

no. of selection inputs in each multiplexer = 4

Size of multiplexers = no. of registers \times

$$= 16 \times 7$$

$$\boxed{= 16 \times 1}$$

no. of multiplexers in 640 = no. of bits in each register

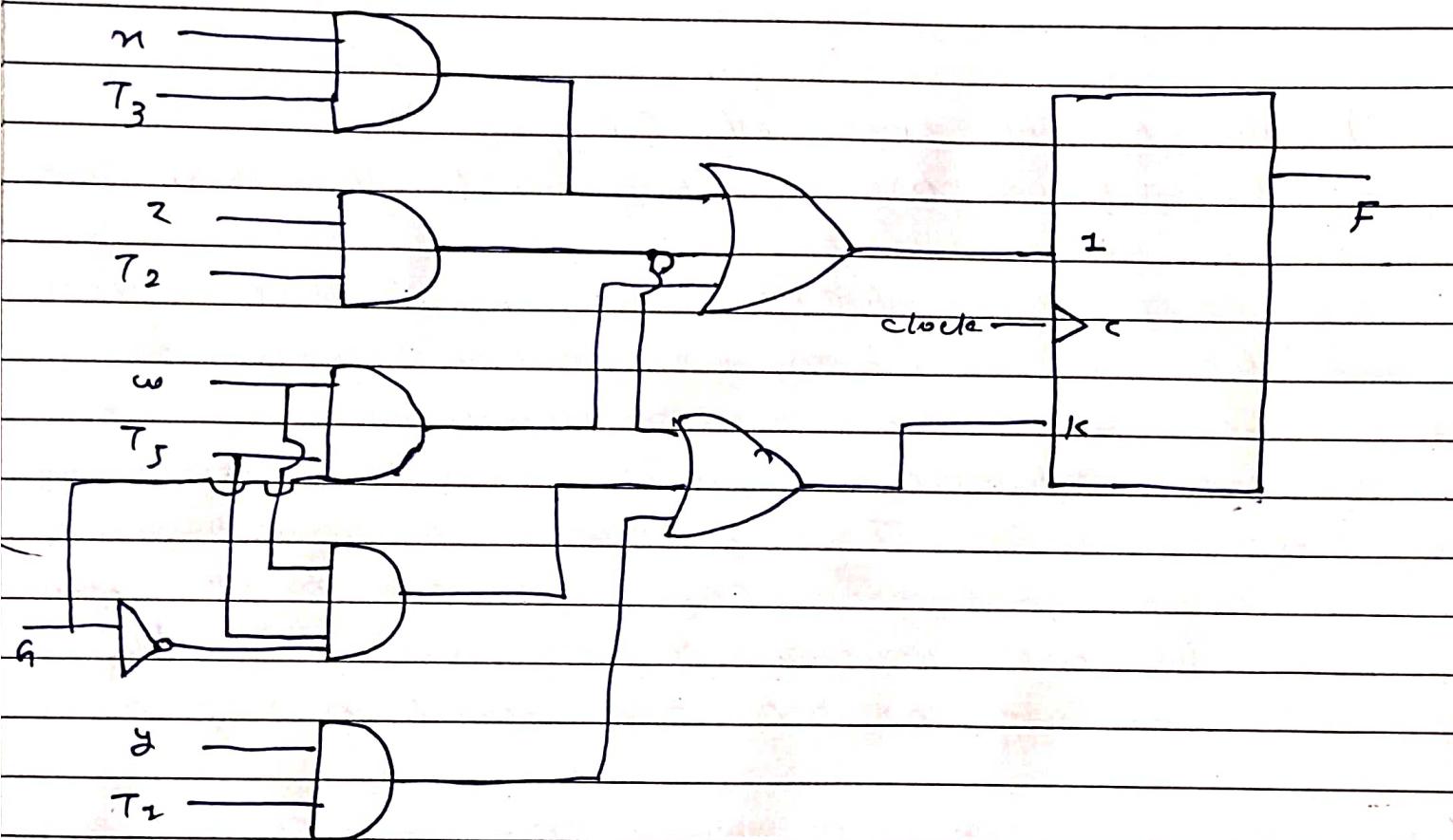
$$= 8$$

\therefore no. of multiplexers = 8.

d) S_01^2

$$\text{Preset } JF = xT_3 + zT_2 + wT_5 G$$

$$\text{Clear } KF = yT_2 + zT_2 + wT_5 G'$$



c)

FGI and FGO are corresponding I/O flags which are considered as control flip-flops. It is used to store the status of input devices such as keyboards, or other peripheral devices that are connected to the computer. When an input device is available the flag FGI is set to 1 and is cleared to 0 when information is accepted by computer. Initially the input flag FGI is cleared to 0. When key is struck in keyboard, 8 bit alphanumeric code is shifted to INPL and input flag FGI is set to 1. **Serial**

g)

⇒ A hardwired control unit is a structure of generating control signals using finite state machines (FSM) suitably. It is created as a sequential logic circuit. The final circuit is generated by physically linking different components like gates, flip flops etc.

A control unit whose binary control values are saved as words in memory is called a micro-programmed control unit. In these instructions, the control memory is considered to be ROM, where all the information is saved permanently. The memory address of the CU denotes the address of micro-instruction. Some of the differences between hardwired and micro-programmed CU are:

Hardwired CU

- it is not applicable to change the structure and instruction set, once it is developed.
- design of computer is complex.
- The architecture and instruction set are not specified.
- difficult to modify as the control signals that needs to be generated are hardwired.

microprogrammed CU

- it is applicable to make modifications by changing the microprogram saved in the control memory.
- design of computer is simplified.
- The architecture and instruction set are specified.
- easy to modify as the modification needs to be done only at the instruction level.

→ Hardwired control unit is faster when compared to microprogrammed CU as the required control signals are generated with the help of hardware.

→ This is slower than the other as micro instructions are used for generating signals here.

Hardware generates the control unit whose signals through a sequence of instructions is referred as a hardwired control unit. Hardwired control unit has a complex structure and is known to be faster. The hardware methods is fall into various categories. The process is generated into hardware signals either by hardwired Control units, whose on bits is typically generated the signals.

f)

→ In program controlled I/O, considerable overhead is incurred because several program instruction have to be executed for each data word transferred between the external devices and memory. Many high speed peripheral devices have a synchronous mode of operation that is data transfer are controlled by a clock of fixed frequency, independent of the CPU.

An I/O interface is bridge between processor and I/O devices - It controls the data exchange between external devices and main memory or external devices and processor registers. So an I/O interface offers an

Sandeep Shrestha
2K21/CO/427

Date

interface internal to computer which connects it to processor and main memory and an interface external to computer connecting it to external device or peripheral.

e) What is the role of IR and PC?

7 IR - Instruction Register and PC - Program Counter

- IR is the part of CPU's control unit that holds the instruction currently being executed or decoded. IR is used to store the instruction word. When the CPU fetches an instruction from memory, it is temporarily stored in the IR. The instruction is a binary word or code that defines the specific operation to be performed. The instruction word is also called OP code. The CPU decodes the instruction and then executes it.

- PC is a register in computer processor that contains the address (location) of instruction being executed at the current time. As each instruction gets fetched, the program counter increases its stored value by 1. It is a digital counter needed for faster execution of tasks as well as for tracking the current execution point.

i) What is the difference between direct and indirect address instruction? How many references to memory are needed for each type of instruction to bring an operand into a processor register?

→ In direct address mode, the address part of the instruction is equal to the effective address. The operand would reside in memory, and the address here is given directly by the instruction's address field. The other name for direct addressing mode is the absolute addressing mode. It is actually the simplest of all the addressing modes, but will provide very limited space for the address.

→ In indirect address mode, the address field of the instruction specifies the address of memory location that contains the effective address of operand.

direct addressing

- address field contains the effective address of operand.
- requires only one memory reference.
- has fast addressing.
- occupies smaller amount of space.
- no additional overhead is involved while searching for operand.
- address space is restricted.

indirect addressing

- address field contains the reference of effective address.
- requires 2 memory references.
- slower than direct addressing mode.
- occupies a large amount of space.
- additional overhead is required/involves while searching for operand.
- requires more number of memory references.

Date

- a direct address instruction needs 2 references to memory: 1) Read instruction, 2) Read operand
- an indirect address instruction needs 3 references to memory: 1) Read instruction, 2) Read effective address, 3) Read operand.

4) What is the basic advantage of using Interrupt-initiated data transfer over transfer under program control without an interrupt?

⇒ In an interrupt-initiated data transfer, the processor verifies the request and transfers control to ISR to perform the task before returning to the useful task, whereas without an interrupt, the processor must waste time performing all tasks.

for eg: - when a print command is given in an interrupt-initiated data transfer, it gives control to ISR and returns to the useful task, whereas without an interrupt, the processor must wait till the print document is transferred to printer.