

Roll No: 35

PAGE NO.

DATE: / /

Name: Shreejan Pandit

Level: Bachelor

Program: BESE

Year: 2018 Semester: 4th

Class Roll No.: 35

Subject: Computer Organization and Architecture (COA)

Date: 2078-03-24

PAGE NO. _____
DATE: / /

Q1a

Ans

Memory based Addressing ModesRegister Based Addressing Mode

1) The operand is present in memory and its address is given in the instruction itself. This addressing mode is taking proper advantage of memory address.
e.g. Direct addressing mode.

An operand will be given in one of the register & register No. will be provided in the instruction. Without the register number present in instruction, operand is fetched ex. Register mode.

2) The memory address specified in instruction may give the address where the effective address is stored in memory.
ex! Indirect Addressing mode.

The register contains the address of the operand. The effective address can be derived from the content of the register specified in instruction.
Ex: Register indirect mode.

3) The content of base register is added to the address part of the instruction to obtain the effective Address.
eg. Base register addressing mode.

If we are having a table of data and our program needs to access all the values one by one we need something which decrements the program counter.
e.g. In Auto decrements mode.

4) The content of the index register is added to the address part i.e given in the instruction to obtain the effective address.
ex! Indexed Addressing mode

If we are having a table of data and our program needs to access all the values one by one we need something which increments PC.
ex! Auto increment mode

The different instructions formats are:

a) Zero address instruction:

A stack based computer doesn't use the address field in the instruction. To evaluate an expression, first it is converted to reverse polish notation i.e. Post fix notation.

Opcode

b) One address instruction:

This uses an implied accumulator register for data manipulation.

Opcode | Address

c) Two address instruction:

The result can be stored at different locations rather than just accumulator, but requires more memory number of bit to represent address.

Opcode | ADDR1 | ADDR2

d) Three address instruction:

This has one Opcode and three address field to specify a register or memory location.

Opcode | ADDR1 | ADDR2 | ADDR3

The different data types are:

a) Numeric data type:

Integer and real number

b) Non numeric data:

Character data, address data,
logical data -

Q2a

Ans

VHDL: Very high speed integrated ext Hardware description Language VHDL was developed to provide a standard for designing digital systems. It specifies a formal syntax. The designer creates a design file using the syntax just as a programmer writes a C-program. It is portable, device independent, simulation and system specification.

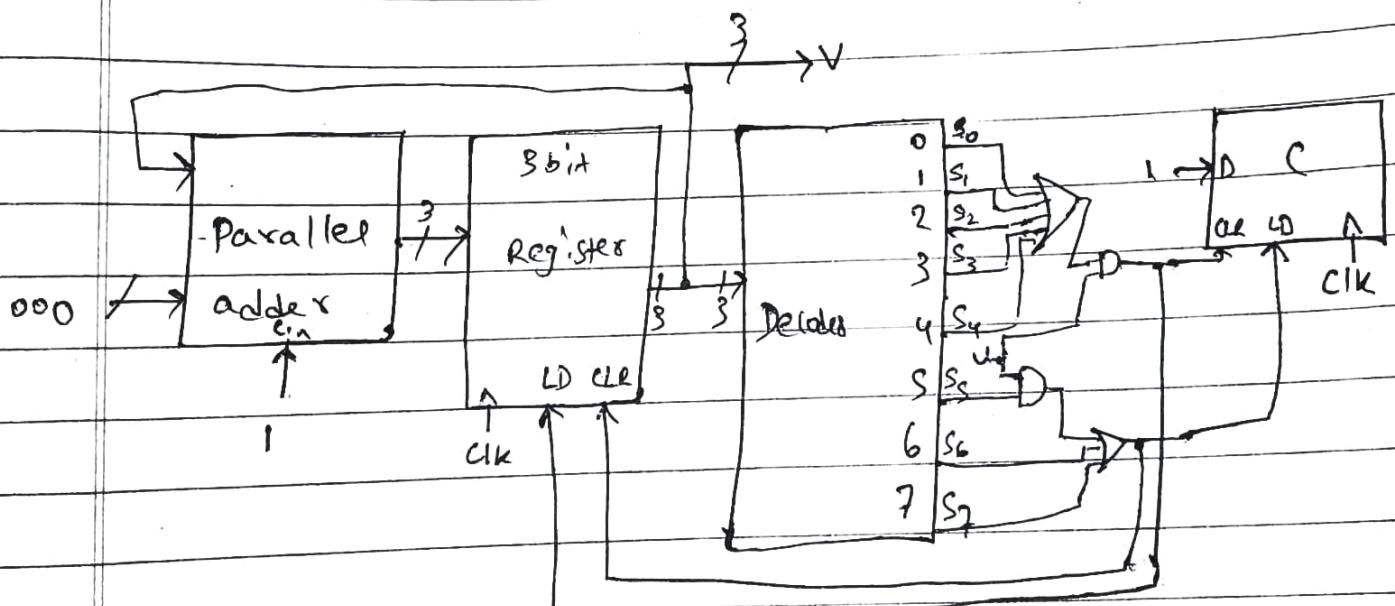


Fig: Implementation of RTL code for mod-6 counter using a Register

Q2b

COA reference to 8085

PAGE NO.
DATE: / /

- 1) The data bus is of 8 bits
- 2) The address bus is of 16 bits.
- 3) The input / output port address are of 8 bits.
- 4) The operating frequency is 3.2 MHz
- 5) It not have multiplication & division instruction.
- 6) It does not support instruction queue.
- 7) Memory space is not segmented.

COA reference to 8086

- 1) The data bus is of 16 bits
- 2) The address bus is of 20 bits.
- 3) The input / output port address are of 8 bits.
- 4) The operating frequency is 5 Hz, 8 MHz, 10 MHz,
- 5) It have multiplication and division instruction.
- 6) It support instruction queue.
- 7) Memory space is segmented.

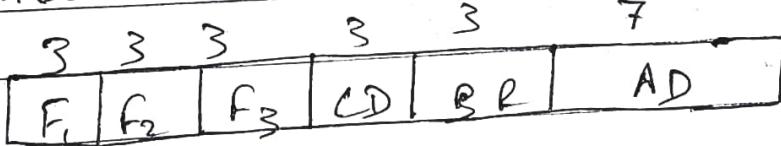
Ans

Q3 & Q4a

Micro instruction format:

The micro instruction format for the control memory is shown below. The 20 bits of the micro instruction are divided into four functional parts as follows:

- 1) The 3 Fields F_1, F_2, F_3 specify micro operations for complex.
- 2) The CD Field selects states bit conditions.
- 3) The BR Field specifies the type of branch to be used.
- 4) The AD Field contains a branch address. The address field is seven bit wide. Since the control memory has $128 = 2^7$ words.



F_1, F_2, F_3 : micro operation fields

CD: Condition for branching

BR: Branch field

AD: Address field.

Fig 4.5: Micro instruction format.

Address sequencing in micro programmed control unit:

Micro instructions are stored in control memory in groups, with each group specifying a routine. Each computer instruction has its own micro program routine to generate micro operation. The h/w that controls the address sequencing of the control memory must be capable of sequencing the micro instruction within a routine and be able to branch from one routine to another.

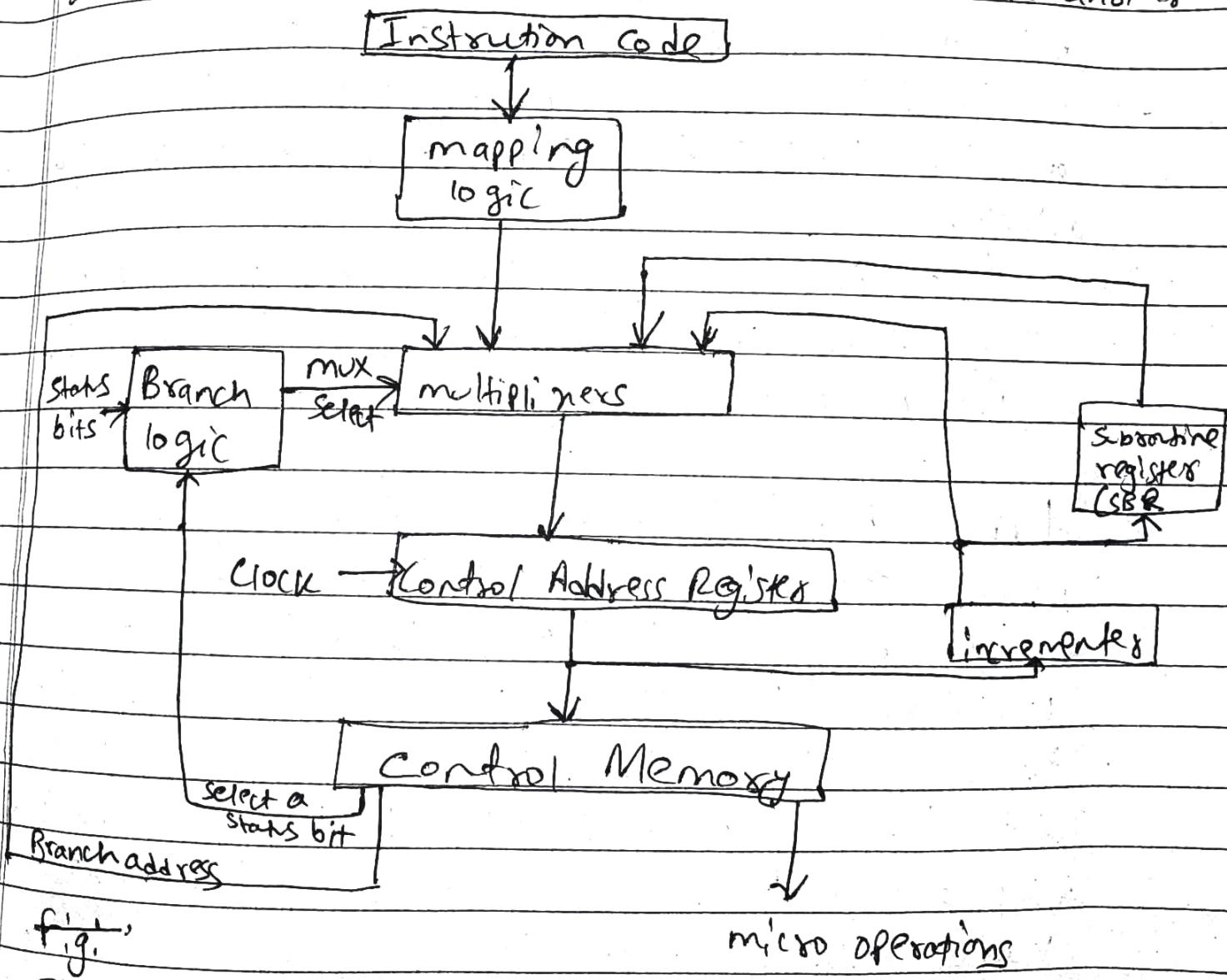


fig:

Fig: Shows a block diagram of control memory and the associated h/w for selecting next micro instruction address

Q5a

Ans Cache memory:

It is a type of fast, relatively small memory i.e. stored on computer h/w, commonly referred to cache, it's classed as RAM which computers microprocessors can access more quickly than regular RAM.

The purpose of cache memory is to store program instructions that are frequently used by SW during its general operations, this is why fast access is needed as it helps to keep the program running quickly.

→ The four most common replacement algorithm related to design issue of cache memory are:

- 1) FIFO: First in first out, FIFO replacement strategy fills the associative memory from its top location to its bottom location. When it copies data to its last location, the cache is full. It then goes back to the top location replacing its data with the next value to be stored.

2) LRU: Least Recently Used, The LRU strategy keeps track of the relative order in which each location is accessed and replaces the least recently used value with the new data. This requires a counter for each location in cache and generally is not used with associative cache. However, it's used frequently with set associative cache memory.

3) Random: As its name implies, the random strategy selects a location to use for the new data. In spite of lack of logic to its selection of location, this replacement method produces good performance, near to the FIFO strategy.

4) LFU: Replace the cache line that has experienced the fewest reference.

~~Ans~~ Q5b
DMA:

DMA ('Direct memory Access' transfers the block of data between the memory and peripheral devices of system, without the participation of the processor. The unit that controls the activity of accessing memory directly is called a DMA controller.

The DMA controller transfers data in three modes:

- **Burst mode:** Here, once the DMA controller gains the charge of the system bus, then it releases the system bus only after completion of data transfer. Till then the CPU has to wait for the system bus.
- **Cycle Stealing mode:** In this mode, the DMA controller forces the CPU to stop its operation & relinquish the control over the bus for a short term to DMA controller.
- **Transparent Mode:** Here the DMA controller takes the charge of system bus only if the processor doesn't require the system bus.

DMA is a mode of data transfer between memory and I/O devices. This happens without the involvement of the processor. We have two other methods of data transfer, programmed I/O and Interrupt-device I/O. These modes of data transfer are not useful for transferring a large block of data. But DMA controller completes the task at a faster rate and is also effective for transfer of large data block.

So, DMA input output transfer is chosen over other data transfer.

Ans

Q6a

Instruction Pipelining:

Pipeline processing can occur not only in the data stream but in the instruction stream as well. Most of the digital computers with complex instruction require instruction pipeline to carry out operation like fetch, decode and execute instruction. In general, the computer needs to process each instruction with the following sequence of steps:

- 1) Fetch instruction from memory
- 2) Decode the instruction
- 3) Calculate the effective address
- 4) Fetch the operands from memory.
- 5) Execute the instruction
- 6) Store the result in proper place.

Features of RISC architecture:

- Fixed length instruction
- Elimination leading & trailing instruction access memory.
- Fewer addressing mode
- Instruction Pipeline
- large no. of registers
- Hardwired control unit
- Delay loads & branches.

Features of CISC architecture:

- The length of code is short
- It requires very little RAM
- Less instruction is needed to write an application.
- It provides easier programming in assembly language.
- Instruction can be larger than a single word.

A1Q6b

Topology is arrangement with which computer system or network device are connected to each other. It may define both physical and logical aspect of the network.

A memory organization and multiprocessor uses multiple modalities to capture different types of DSM because IT technology is greatly advanced & lot's of information is shared via the internet. To improve the performance and efficiency of the that multiprocessor system and memory organization we can use different type of techniques that is based on the concept and implementation of h/w, s/w and hybrid DSM. This paper provides an almost exhaustive survey of the existing problem and solution in a uniform manner presenting their memory organization shared memory, distributed memory, distributed shared memory. Memory consistency model and s/w based DSM mechanism & issue of importance for various DSM system & approaches.

Q7a

An Wallace Tree Multiplier:

It is a hw implementation of a binary multiplier, a digital circuit that multiplies two integer. It uses a selection of full and half adders to sum partial products in stages until two numbers are left. If Wallace multipliers reduce as much as possible on each layer. It has three steps:

- 1) Multiply each bit of one of the arguments by each bit of other.
- 2) Reduce the number of partial products to two by layers of full and half adders.
- 3) Group the wires in two numbers, and add them with conv. conventional adder.

A3

Q7b

Pipeline Hazard:

Instruction pipeline improve the overall performance but it also introduces some problems / conflicts / hazard.

A hazard is a potential problem that can happen in a Pipeline processor. It refers to the possibility of erroneous computation. When CPU tries to simulate simultaneously execute multiple instruction which shows dependency.

Types of Pipeline Hazard:

a) Data Hazard:

It occurs when pipeline changes the order of read/write access to operand / data. It occurs due to data dependency.

b) Structural Hazard:

This is also known as resource conflict. It occurs when a part of processor bus is needed by 2 or more instruction at same time.

i) Branch Hazard:

It occurs when the process is tried to branch ie if a certain condition is true then jump to other part of instructions.

Q 4b

Ans

RTL code for Booth algorithm:

The algorithm is expressed as:

$$U = 0, Y_0 = 0$$

for, i=1 to n Do

If start of a string of 1's in
 $y \rightarrow$ If $y_{0-1} \& \& y_0 = 0$
 $\{ y_0 = 1 \}$

then $U = U - x = (U + x^{i+1})$

If end of a string of 1's in
 then $U = U + x^i$

$$\text{If } y_0 = 0, y_1 = 1$$

arithmetic shift right UV.

~~Arithmetic~~ ^{Circular} Shift right UV and
 copy y_0 to ~~y_0~~ y_1 .

→ Micro operation to booth's algorithm

Start

$$1: U = 0, Y_0 = 0, i \leftarrow n$$

$$2: U \leftarrow U + x^{i+1}$$

$$3: U \leftarrow U + x$$

$$4: i \leftarrow i - 1$$

$$5: \text{ashr}(UV), \text{cir}(Y), Y_1 \leftarrow y_0$$

2' 3': Goto 2

3: Finish \leftarrow 1

$(y_0 \oplus y_n)^2 : v \leftarrow v + (x \oplus y_0) + y_0$