

Microprocessors is a programmable device.

Page:

Date: / /

Microprocessors :

Microprocessor is a computer processor that incorporates the function of a computer's central processing unit in a single integrated circuit (IC). Microprocessor is a controlling unit of a micro-computer, fabricated on a small chip capable of performing ALU (Arithmetic Logic Unit) operations and communicating with the other devices connected to it. Microprocessor consists of an ALU, registers array and a control unit. ALU performs arithmetical and logical operations on the data received from the memory or an input device. Registers array consists of registers identified by letters like B, C, D, E, H, L and accumulators (registers in which intermediate arithmetic and logic results are stored). The control unit controls the flow of data and instruction within the computer.

Block diagram of a microcomputer

INPUT device	Microprocessor (ALU + Registers array + control unit)	OUTPUT device
-----------------	---	------------------

Ultra White Plus Memory

Page :

Date: / /

\* Application areas of microprocessors.

Nowadays, a microprocessor based systems are used in instructions, automatic testing product, speed control, of motors, traffic light control, light control of furnaces etc. some of them are :-

1. Instrumentation :-

It is very useful in the field of instrumentation frequency counters, functions generators, frequency synthesizers, spectrum analyses and many other instruments are available, when microprocessors are used as controllers. It is also used in medical instrumentation.

2. Control :-

Microprocessors based controllers are available in home appliances, such as microwave oven, washing machine etc. Microprocessors are being used in controlling various parameters like speed, pressure, temperature etc. These are used with the help of suitable transducers.

Ultra White Plus

Page :	/	/
Date:	/	/

### 3. communication :

Microprocessors are being used in a wide range of communication equipments. In telephone industry, these are used in digital telephone sets. Telephone exchanges and modems etc. The use of microprocessors in television, satellite communication have made teleconference possible. Railway reservation and air reservation system also uses this technology. LAN and WAN are communication of vertical information through computer network.

### 4. consumer :

The use of microprocessors in toys, entertainment equipment and home application is making them more entertaining and full of features. The use of microprocessors is more widespread and popular. Now the microprocessors are used in :

1. Calculators
2. Accounting system
3. Games machine
4. complex industrial controllers
5. Traffic light control
6. Data acquisition system
7. Military application

*Ultra White Plus*

<p style="text-align: right;">Page : / /</p> <p>Differences between microprocessors and microcontrollers +</p>	
Microprocessors (MPU)	Microcontrollers (MCU)
1. Microprocessor is the core of computer system.	1. Microcontroller is simply a core of embedded system.
2. Microprocessor is not capable to be used in compact system.	2. Microcontroller has the potential to be used in compact system.
3. Microprocessors are the electronics that do the actual computation.	3. Microcontrollers include a microprocessor connected to a memory
4. It is a manager of the resources (I/O, memory) which lie outside of its architecture.	4. It have I/O, memory etc. built into it and specially designed for control applications.
5. It mainly used in personal computers.	5. It used mainly in washing machines MP3 - players.

Ultra White Plus

Page: / /  
Date: / /

- |                                  |   |
|----------------------------------|---|
| 6. It has less no. of registers. | 6. It has more no. of registers.        |
| 7. It is mainly used in PC.      | 7. It is used in working machines, MP3. |
| 8. Based on von Neumann model.   | 8. Based on Harvard model.              |

### Generation of Microprocessors

#### a. 1<sup>st</sup> generation:

This was the period during 1971 to 1973 of microprocessors history. In 1971, INTEL created the first microprocessor 4004 that would run at a clock speed of 108 kHz.

#### b. 2<sup>nd</sup> generation:

This was the period during 1978 to 1978 in which very efficient 8-bit microprocessors were implemented like Motorola 6800 and 6801, INTEL 8085 and Zilog - Z80, which were costly owing to their superfast speed.

**Ultra White Plus**

Page :	/ /
--------	-----

c. 3<sup>rd</sup> generation :

During this period 16 bit processor were created and designed using NMOS technology. From 1971 to 1980, INTEL 8086 | 80186 | 80286 and MOTOROLA 68000 and 68010 were developed and their speeds were 4 times better than 2<sup>nd</sup> generation.

d. 4<sup>th</sup> generation :

From 1981 to 1995 this generation developed 32 bit microprocessors by using CMOS fabrication. INTEL 80386 and MOTOROLA 68020 | 68 were the popular processors.

e. 5<sup>th</sup> generation :

From 1995 to until now this generation has been bringing out high performance and high speed microprocessors that make use of 64-bit processors. Such processors includes Pentium, Celeron, Dual and quad core processors.

**Ultra White Plus**

Page :  
Date: / /

\* Microprocessor system with its system architecture.

↳ System bus :

The 8085 MPU performs these operations using three set of communication lines called buses: the address bus, the data bus and the control bus. These buses are collectively called the system bus. The system bus is nothing but a group of wires to carry bits. All peripherals (and memory) share the same bus; however, the microprocessor communicates with only one peripheral at a time. The timing is provided by the control unit of the microprocessors.

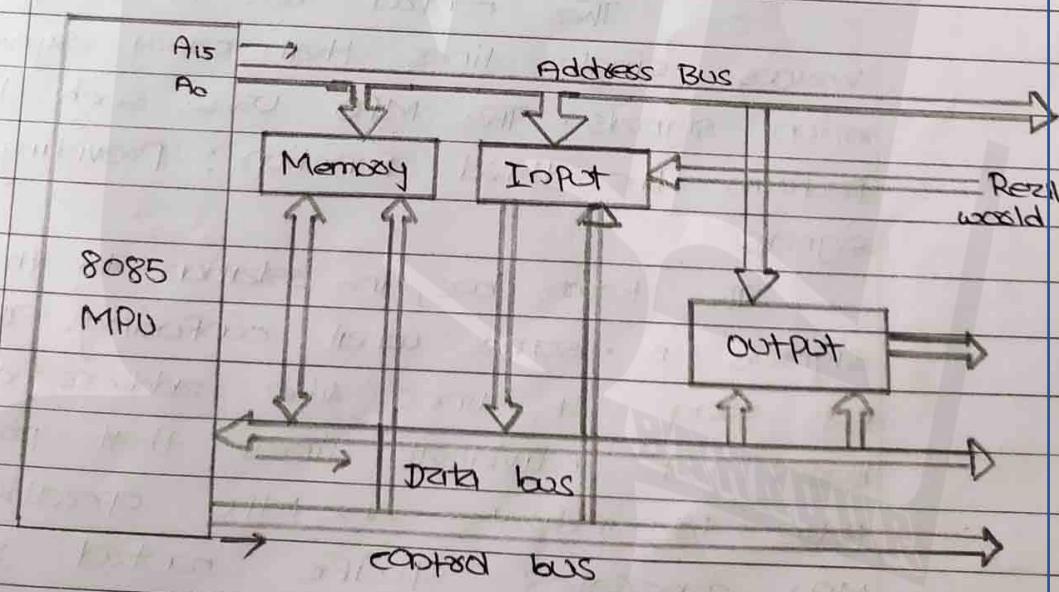


Fig : The 8085 Bus structure  
Ultra White Plus

Page :  
Date: / /

### 1. Address Bus :

The address bus is a group of 16 lines generally identified as  $X_0$  to  $X_{15}$ . The address bus is unidirectional. Bits flow in one direction from the MPU to peripheral devices. The MPU uses the address bus to perform the first function identifying a peripheral or a memory location.

In a computer system, each peripheral or memory location is identified by a binary number called an address, and the address bus is used to carry a 16-bit address.

### 2. Control bus :

The control bus is comprised of various single lines that carry synchronization signals. The MPU uses such lines to perform the third function: providing timing signals.

The term bus, in relation to the control signals is somewhat confusing. These are not group of lines like address or data buses, but individual lines that provide a pulse to indicate an MPU operation. The MPU generates specific control signals.

Ultra White Plus

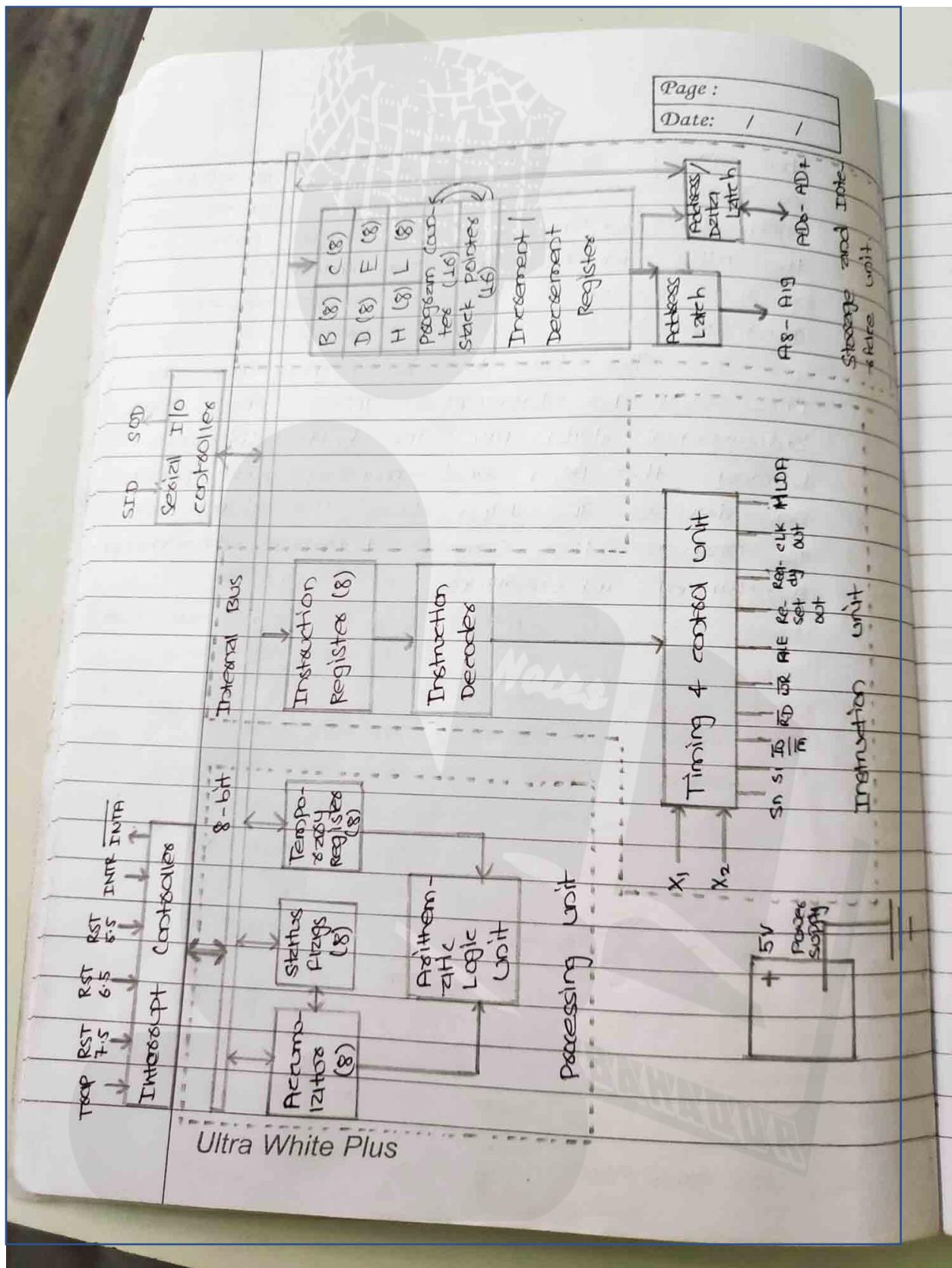
Page :  
Date: / /

for every operation (such as Memory Read or I/O write) it performs. These signals are used to identify a device type with which the MPU intends to communicate.

### 3. Data Bus :

The data bus is a group of eight lines used for data flow. These lines are bidirectional. Data flow is both directions between the MPU and memory and peripheral devices. The MPU uses the data bus to perform the second function transmission of binary information.

Ultra White Plus



Page : / /  
Date: / /

### Introduction to 8085

- ↳ Introduced in 1974
- ↳ It is 8-bit MP.
- ↳ It is a 40 pin dual-in line chip
- ↳ It uses a single +5V supply for its operations.
- ↳ Its clock speed is about 3MHz.

### Three units of 8085

1. Processing unit
2. Instruction unit
3. Storage and Interface unit

#### a. Processing unit

##### a. Arithmetic & Logic Unit (ALU)

- ↳ It performs various arithmetic and logic operations.
- ↳ The data is available in accumulators and temporary / general purpose registers.

##### \* Arithmetic operations :

- Addition, subtraction, Increment, Decrement etc.

##### \* Logic operations :

- AND, OR, X-OR, complement etc.

#### b. Accumulators

- ↳ It is the main registers of microprocessor.

*Ultra White Plus*

Page :	/
Date:	/ /

- ↳ It is also called register 'A'.
- ↳ It is an 8-bit register.
- ↳ It is used in the arithmetic and logic operations.
- ↳ It always contains one of the operation on which arithmetic / logic has to be performed.
- ↳ After the arithmetic / logic operation the contents of accumulator are replaced by the result.

### C. Status Flags

- ↳ Status flags are set of flip-flops which are used to check the status of accumulators after the operation is performed.

S	Z	X*	AC	X*	P	X*	CY
---	---	----	----	----	---	----	----

D7 D6 D5 D4 D3 D2 D1 D0

X\* = don't care condition

- S = sign flag
- Z = zero flag
- AC = auxiliary carry flag
- P = parity flag
- CY = carry flag

Ultra White Plus



1132

Page:	/
Date:	/ /

gk  
ions

re

o-

- Sign Flag (S)
  - ↳ It tells the sign of result stored in Accumulator after the operation is performed.
  - ↳ If result is -ve, sign flag is set (1)
  - ↳ If result is +ve, sign flag is reset (0)
- zero flag (Z)
  - ↳ It tells whether the result stored in Accumulator is zero or not after the operation is performed.
  - ↳ If the result is zero, zero flag is set (1).
  - ↳ If result is not zero, zero flag is reset (0)
- Auxiliary carry flag (AC)
  - ↳ It is used in BCD operations.
  - ↳ When there is carry in BCD addition, we add 0110 (6) to the circuit.
  - ↳ If there is carry in BCD addition, auxiliary carry is set (1).
  - ↳ If there is no carry, auxiliary carry is reset (0).
- Parity flag (P)
  - ↳ It tells the parity of data stored in Accumulator.
  - ↳ If Parity is even, parity flag is reset (0)

Ultra White Plus

Page : / /  
Date: / /

↳ If parity is odd, Parity flag is set (1).

d. Temporary Registers

- ↳ It is an 8-bit register.
- ↳ It is used to store temporary 8-bit operators from general purpose registers.
- ↳ It is also used to store intermediate results.

e. Program status word (PSW)

- ↳ The contents of Accumulator and Status Flag clubbed together is known as Program status word.
- ↳ It is the 16-bit word. Acc. (8) Status Flag (8)

2. Instruction units.

a. Instruction Registers.

- ↳ It is used to hold the current instruction which the microprocessor is about to execute.
- ↳ It is an 8-bit register.

b. Instruction Decoders

- ↳ It interprets the instruction stored in instruction registers.
- ↳ It generates various machine cycles depending upon the instruction.

*Ultra White Plus*

Page :	/ /
Date:	/ /

Q (a).

- ↳ The machine cycles are then given to the timing and control unit.
- ✓ c. Timing and control unit.
- ↳ It controls all the operations of microprocessor and peripheral devices.
- ↳ Depending upon the machine cycle received from Instruction Decoder, it generates 12 control signals

So and Si (status signals)

ALE (Address Latch Enable)

RD (Read, active low)

WR (Write, active low)

IO/M (Input-output | memory).

Ready

RESET IN

RESET OUT

CLK OUT

HOLD and HLDA

### 3. Storage and Interface Unit

#### a. General Purpose Registers

- ↳ There are 6 general purpose registers, namely B,C,D,E,H,L

Ultra White Plus

	<table border="1" style="float: right; margin-bottom: -20px;"> <tr><td>Page :</td></tr> <tr><td>Date: / /</td></tr> </table>	Page :	Date: / /
Page :			
Date: / /			
	<p>↳ Each of them is 8-bit registers.</p> <p>↳ They are used to hold data &amp; results.</p> <p>↳ To hold 16-bit data, combination of two 8-bit registers can be used, the combination known as Register Pairs.</p> <p>↳ The valid register pairs are : B-C, D-E, H-L</p>		
b.	<p>Stack pointers</p> <p>↳ It holds the address of top most item in the stack.</p> <p>↳ It is also 16-bit registers.</p> <p>↳ Any portion of memory can be used as stack.</p>		
c.	<p>Program Counter</p> <p>↳ It is used to hold the address of next instruction to be executed.</p> <p>↳ It is a 16-bit registers.</p> <p>↳ The microprocessor increments the value of program counter after the execution of the current instruction, so that, it always points to the next instruction.</p>		

Ultra White Plus

Page: / /  
Date: / /

- d. Increment / Decrement Registers
- ↳ This register is used to increment or decrement the value of stack pointer.
  - ↳ During push operations, the value of stack pointer is incremented.
  - ↳ During pop operations, the value of stack pointer is decremented.

e. Address Latch

- ↳ It is a group of 8 buffers.
- ↳ The upper - byte of 16-bit address is stored in this latch.
- ↳ And then it is made available to the peripheral devices.

f. Address | Data Latch

- ↳ The lower - byte of address and 8-bit of data are multiplexed.
- ↳ It holds either lower - byte of address or 8-bits of data.
- ↳ This is decided by ALE (Address Latch Enable).
- ↳ If  $ALE = 1$  then, it contains lower - byte of address.
- ↳ If  $ALE = 0$  then, it contains 8-bit data.

Ultra White Plus

Page : / /  
Date: / /

- ↳ Serial I/O controllers
- ↳ It is used to convert serial data into parallel and parallel data into serial.
- ↳ Microprocessors works with 8-bit parallel data.
- ↳ Serial I/O devices works with serial transfer of data.
- ↳ Therefore, this unit is the interface between micro-processors and serial I/O devices.

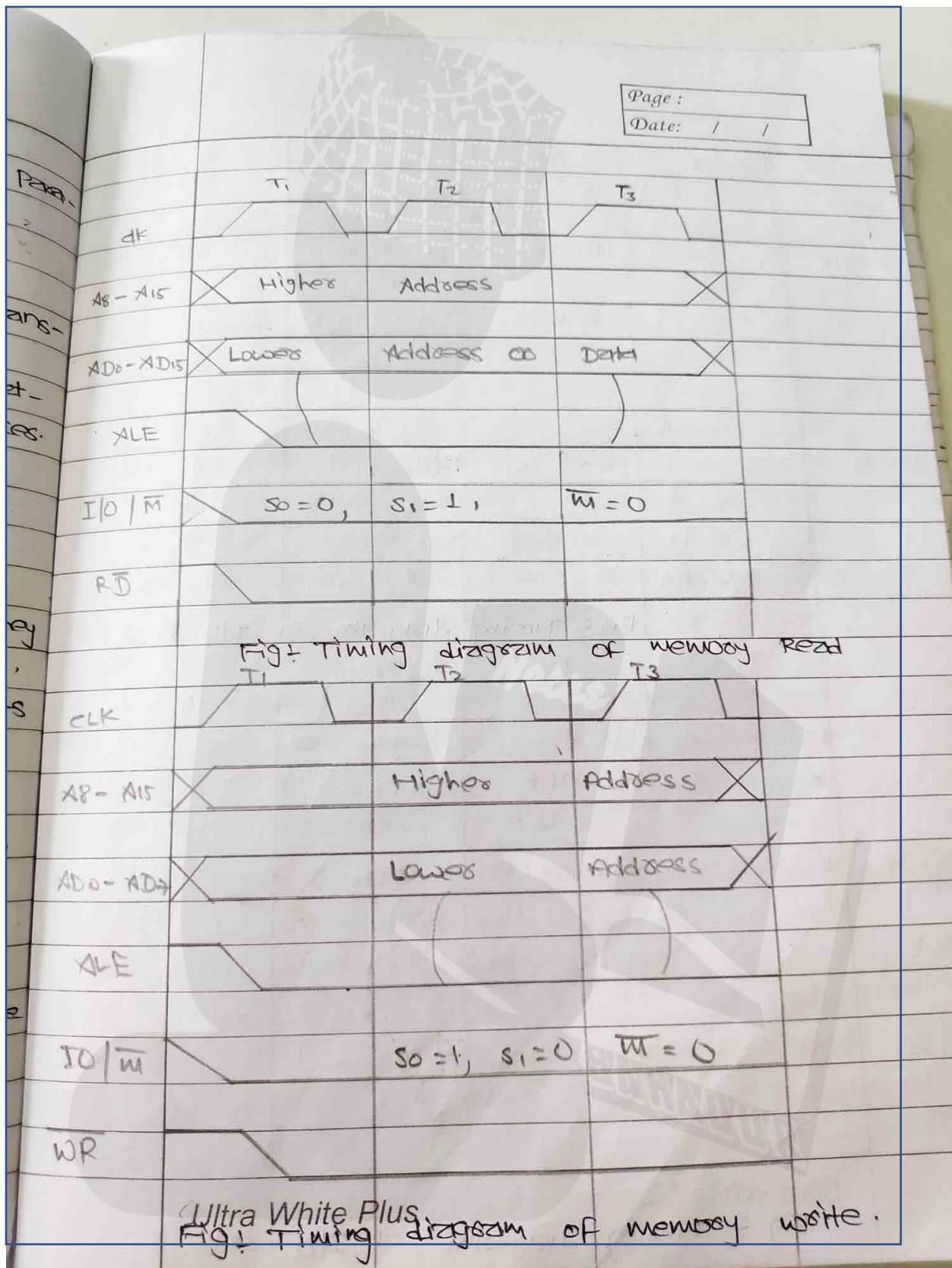
### Interrupt controllers

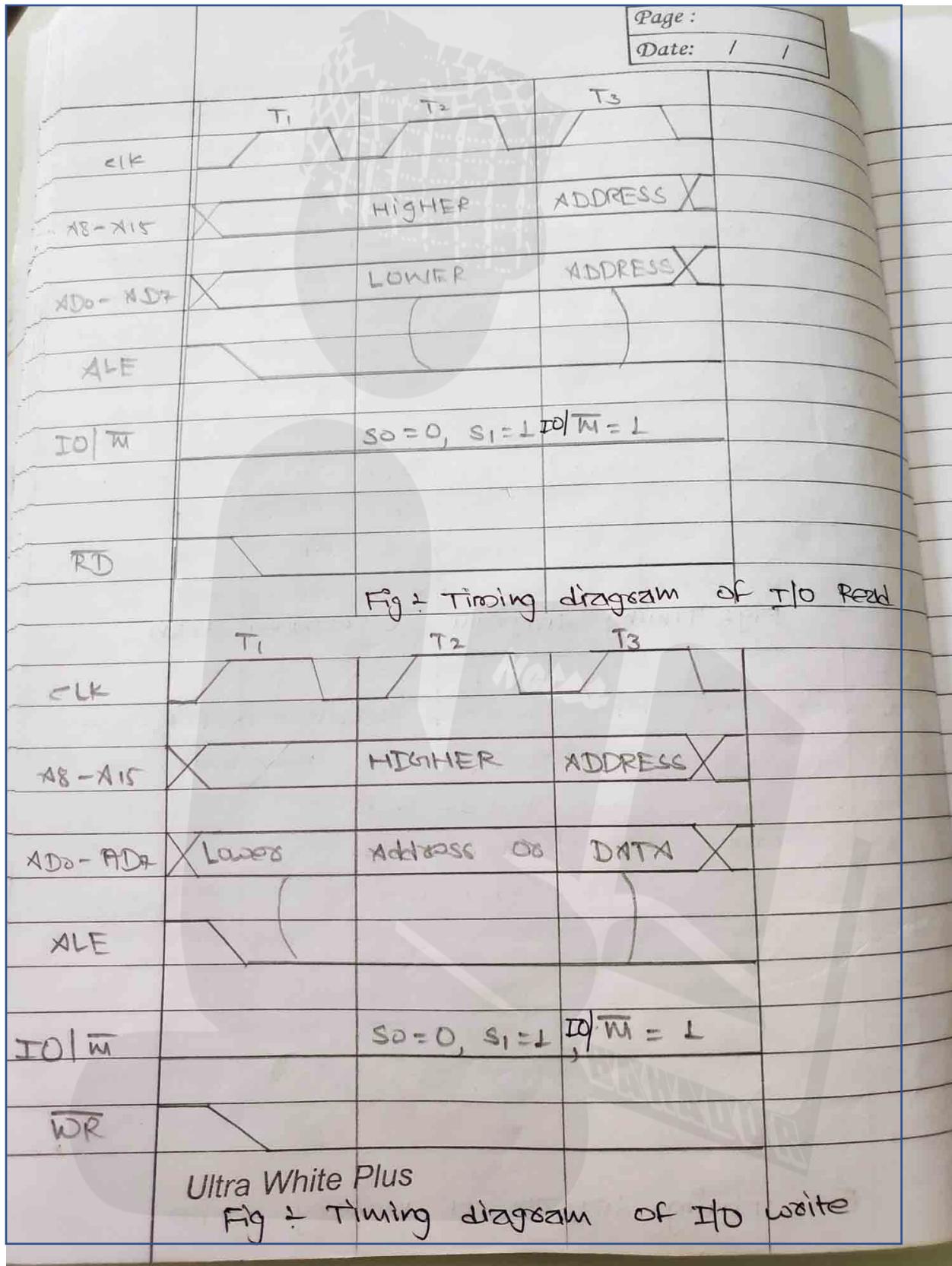
- ↳ It is used to handle the interrupts.
- ↳ There are 5 interrupt signals in 8085 microprocessor: T0, RST 4.5, RST 6.5, RST 5.5, INTR.
- ↳ Interrupt controller receives these interrupt according to their polarity and applies them to the microprocessors.
- ↳ There is one outgoing signal INTA which is called Interrupt Acknowledge.

### Power Supply

- ↳ This unit provides +5V power supply to the microprocessors.
- ↳ The microprocessors needs +5V power supply for its operation.

Ultra White Plus





Page : / /  
Date: / /

4. WAP for the division of 8-bit numbers.

INPUT data: FF | FF  
Memory Address: 2050

OUTPUT data: 01 | FE  
3051 | 305

Algorithm:

1. Start the program by loading HL pair registers with memory address.
2. Move data to B registers.
3. Load the second data into accumulator.
4. Compare two numbers to check carry.
5. Subtract two numbers.
6. Increment the value of carry.
7. Check whether the repeated subtraction is over.
8. Then store results (quotient and remainder) in given memory location.
9. Terminate program.

Ultra White Plus

Page :	
Date:	/ /
Program:	Mnemonics
Address	
2000	LXI H, 2050
2003	MOV B,M
2004	MVI C,00
2006	INX M
2007	MOV A,M
2008	CMP B
2009	JC 2011
200C	SUB B
200D	INR C
200E	JMP 2008
2011	STA 305B
2014	MOV ,A,C
2015	ST A 305L
2018	HL4
Explanation X, H, L, C, B registers are used for general purpose.	
1.	LXI H, 2050 will load HL Pairs registers with memory address of 2050.
2.	MOV B,M copies content of memory in register B.
3.	MVI C,00 assign 00 to C.
4.	MOV A,M copies content of memory into accumulator.
Ultra White Plus	

Page : / /  
Date: / /

5. INX H increment registers H to HL.
6. CMP B compares content of accumulator and register B.
7. JC 2011 jump to address 2011 if carry flag is set.
8. SUB B subtract content of accumulator with register B and store result in accumulator.
9. INR C increment register C.
10. JMP 2008 control will shift to memory address 2008.
11. STA 3050 stores renzhades at memory location 3050.
12. MOV A,C copies the content of registers into accumulator.
13. STX 3051 stores renzhades at memory location 3051.
14. HLT stops executing programs and halts any further execution.

### Q. Instruction for 8085 microprocessors.

- \* An instruction is command given by user to perform certain operation on data.
- \* An instruction is a binary pattern designed inside a micro-processors to perform certain task. The entire group of instruction is instruction set, determines what function of micro

Ultra White Plus

Page : / /

processes can perform.

- \* Each instruction has two parts : one is the task to be performed, called the operation code (op-code) and second is the data to be operated (operand).
- \* Operand - may include 8 bit (or 16 bit) data address an internal registers, memory location.

# Types of instruction.

on the basis of word / byte size.

```

graph TD
    A[ ] --> B[1 Byte]
    A --> C[2 Byte]
    A --> D[3 Byte]
  
```

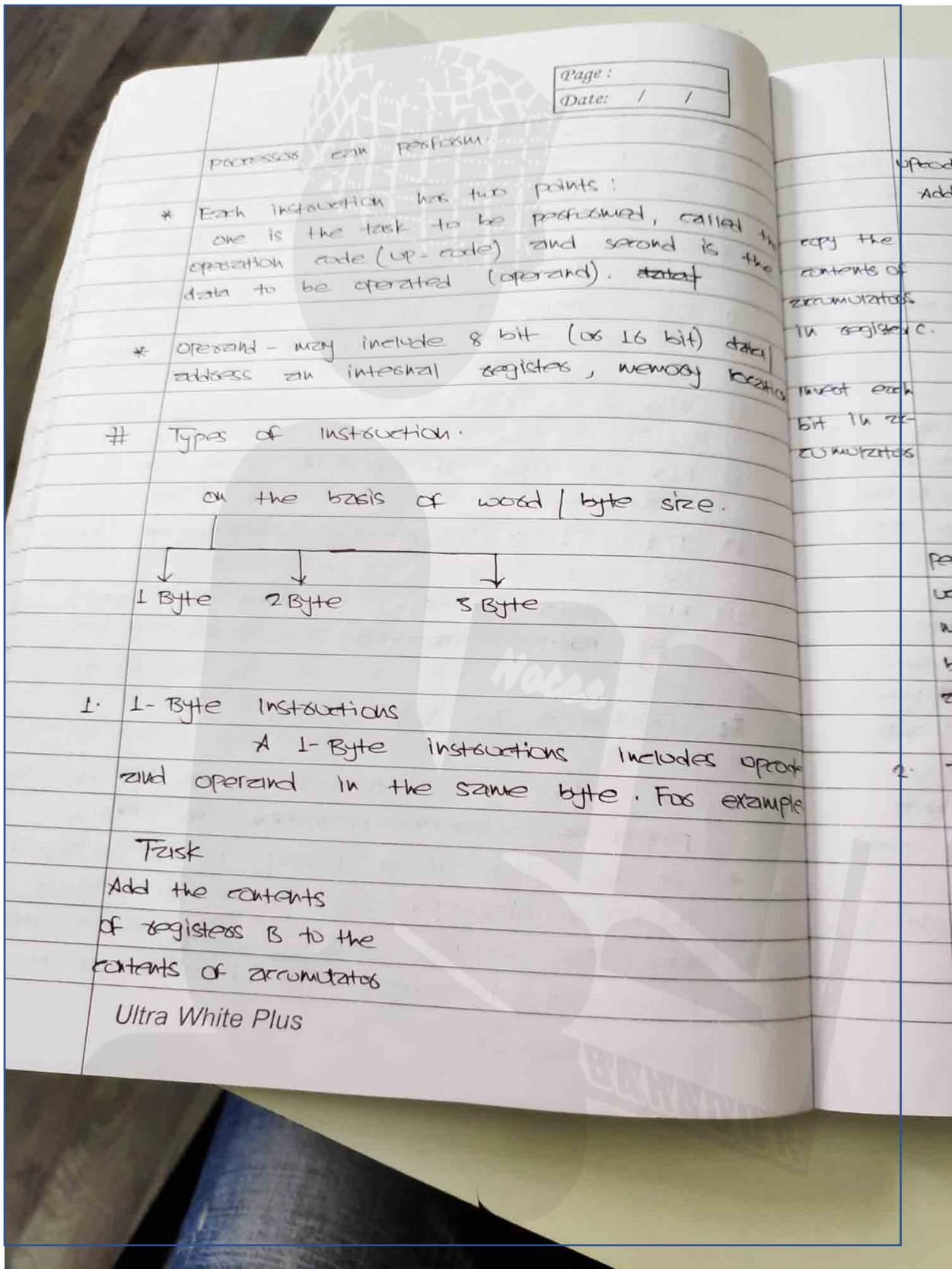
1. 1-Byte Instructions

A 1-Byte instruction includes opcode and operand in the same byte. For example

Task

Add the contents of registers B to the contents of accumulator

Ultra White Plus



opcode	operand	Binary code	Hex code
Add	B	1 000 000	80H

copy the contents of accumulators in register C.

next each bit in register C

MOV	C, A	0100 1111	4FH
-----	------	-----------	-----

These instruction are 1-byte instructions performing 3 different task in the first instruction, the operand B is specified and accumulators is assumed. In the 2nd instruction, both operand registers is specified. In 3rd, accumulators is assumed to be implicit operand.

## 2. Two - Byte instruction

In a 2-byte, the first byte specifies the operation code and 2nd byte specifies operand. For eg :

Ultra White Plus

Task	Opcode	Operand	Binary code	Hex code
work on 8-bit data byte in accumulators.	MVI	A, 32H	0011 1110 0011 0010	3E 32 1st Byte 2nd Byte

These instruction would require two memory locations each to store binary codes.

### 3. Three-Byte Instruction:

In a 3-byte instruction, the first byte specifies the opcode and following two bytes specify 16-bit address. The second byte should be low-order address and 3rd bytes is high order address.

Task	Opcode	Operand	Binary code	Hex code
Load contents of memory 2050 H into A.	LDA	2050 H	0011 1110 0101 0000 0010 0000	3A 50 20 1st Byte 2nd Byte 3rd Byte

These instruction would require three memory locations each to store binary codes.

The various ways of specifying data is addressing modes.

Ultra White Plus

Page : / /  
Date: / /

On the basis of function.

```

graph TD
    A[On the basis of function.] --> B[Data transfer]
    A --> C[Arithmetic]
    A --> D[Logical]
    A --> E[Branching]
    A --> F[Machine control]
  
```

1. Data transfer (copy) operations!

Data transfer is used for this copying functions. However, transfer is misleading, it creates the impression that the contents of a source are destroyed when in fact, the contents are retained without any modification. It copies a data from a location called a source to another, location, called a destination.

Types	Examples
* Between registers.	copy the contents of registers B to D.
* Specific data byte to a register or memory location.	Load register B with data byte 32 H.
* Between a memory location & a register.	From memory location 2000 H to register B.

Ultra White Plus

Page: / /  
Date: / /

\* Between an I/O device and the accumulators from a input keyboard to accumulators.

2. Arithmetic operations :  
 These instructions perform arithmetic operations such as addition, subtraction, increment, decrement.

- Addition :  
 Any 8-bit numbers, or the content of a registers, or the contents of memory location can be added to content of accumulators and stored in accumulators. No two other 8-bit registers can be added directly (ie, contents of registers B cannot be directly added to registers C)
- Subtraction :  
 Any 8-bit numbers, or the contents of registers or location of memory is subtracted. The subtraction is performed in 2's complement, result if -ve, are expressed in 2's complement. No two other registers can be subtracted directly.

Ultra White Plus

Page : / /  
Date: / /

- Increment / Decrement as 16-bit (as BC) :  
The 8-bit contents of 16-bit (as BC) of registers or memory location can be incremented / decremented by 1. They can be performed in anyone of the registers in a memory location.

### 3. Logical operations :

These instructions perform various logical operations with the content of accumulators.

- AND, OR, X-NOR :

Any 8-bit no. contents of registers, memory location can be ANDed, ORed, X-ORED with the accumulator's content and result are stored in accumulators.

- Rotate :

Each bit in accumulators can be shifted either left or right to next position.

- Compare :

Any 8-bit numbers, or register's contents, memory locations can be compared for equality, greater than, or less than with the contents of accumulators.

Ultra White Plus

Page :	
Date:	/ /

- complement :
 

The contents of accumulators can be complemented; all os are replaced by ls and all ls are replaced by os.
  
- 4. Branching operations :
 

The group of instructions alters the sequence of program execution either conditionally or unconditionally.
  
- Jump :
 

conditional jumps are an important aspect of the decision making process in programming. These instructions test for a certain condition (e.g zero or carry flag) and alter the program sequence when condition is met. In addition, the instruction set includes an instruction called unconditional jump.
  
- Call, Return and Restart :
 

These instructions change sequence of program either by calling a subroutine or returning from subroutine.

*Ultra White Plus*

*Page:* / /

## 5. Machine control operations:

These instructions control memory functions such as halt, interrupt or do nothing.

D. *How do we know that the other elements which are H, C, O, N, S, P, Cl, K, Ca, Fe, Mn, Cu, Zn, Mg, B, and so on, are also present in the plant? How do we know that the plant has absorbed all these elements from the soil?*

Ans. *When we eat plants, we eat all these elements. When we eat animals, we eat all these elements. So, we get all these elements from the food we eat.*

*Q. What are the different types of organic acids found in plants?*

Ans. *There are two types of organic acids found in plants:*

- 1. Carboxylic acids*
- 2. Phenolic acids*

*Q. What are the properties of carboxylic acids?*

Ans. *Properties of carboxylic acids:*

- 1. They are soluble in water.*
- 2. They are soluble in organic solvents like ethanol, ether, etc.*
- 3. They are acidic in nature.*
- 4. They are strong acids.*
- 5. They are weak acids.*

*Q. What are the properties of phenolic acids?*

Ans. *Properties of phenolic acids:*

- 1. They are soluble in water.*
- 2. They are soluble in organic solvents like ethanol, ether, etc.*
- 3. They are acidic in nature.*
- 4. They are strong acids.*
- 5. They are weak acids.*

Different between computers organization computers architecture.	
Computers Organization	Computers Architecture
1. It is concerned with the structure and behaviour of computer system as seen by the user.	1. It is concerned with the way hardware is put together.
2. It deals with the components of connection in a system.	2. It acts as an interface between hardware and software.
3. It tells us how exactly all the units in the system are connected.	3. It helps us to understand the functionalities of a system.
4. An organization is done on the basis of architecture.	4. While designing an architecture.

Ultra White Plus

Von Neumann Architec.	Harvard Architecture
Page : / /	Date: / /
1. It is the theoretical design based on the stored program concept.	1. It is a modern computer architecture based on the Harvard work's set by based computers model.
2. It uses same physical memory address for instructions & data.	2. It uses separate memory address for instructions and registers.
3. Process needs two cycles to execute an instruction.	3. Processors needs one cycle to complete an instruction.
4. Simplex control unit for design and development of memory is cheaper and faster.	4. Control unit for two buses is more complicated which is slower and faster.
Ultra White Plus	

Page :	
Date:	/ /

1. NAP to sum two 8 bit numbers.

- Assumption:
1. The starting address of the program is 2000.
  2. Memory address of first no. is 2050.
  3. Memory address of second no. is 2051.
  4. Memory address of result is 2052.

Now,

Input : 2050 : 08  
           : 2-51 : 04  
         Output : 2052 : 07

Algorithm.

1. Load the first number to accumulator through memory address 2050.
2. Move the content of accumulator to register B.
3. Load the second no. of accumulator through memory address 2051.
4. Add the content of accumulator and register B and result will be stored at accumulator.
5. store the result from accumulator to memory address 2052.

Ultra White Plus

6. Terminate the program.

Page: / /  
Date: / /

Memory Address	Mnemonics	Comment
2000	LDA 2050	A ← [2050]
2003	MOV B, A	B ← A
2004	LDA 2051	A ← [2051]
2007	ADD B	A ← A+B
2008	STA 2052	[2052] ← A
200B	HCI	Terminate

### Explanation

1. LDX 2050 : This instruction will load the numbers from memory to accumulators.
2. MOV B, A : This instruction on will move the content of a accumulator to registers B.
3. LDA 2051 : this instruction will load the numbers from memory to the accumulation.
4. ADD B : this instruction will sum the content of the accumulators with the content of the registers B.

Ultra White Plus

Page :  
Date: / /

5. STA 2052 : This instruction will store the content of the accumulators to the memory address 2052.

6. HLT: This instruction will terminate the program.

MVI → move immediately  
Directly give data to accumulators.

MVI A, 05H → data

MVI B, 04A

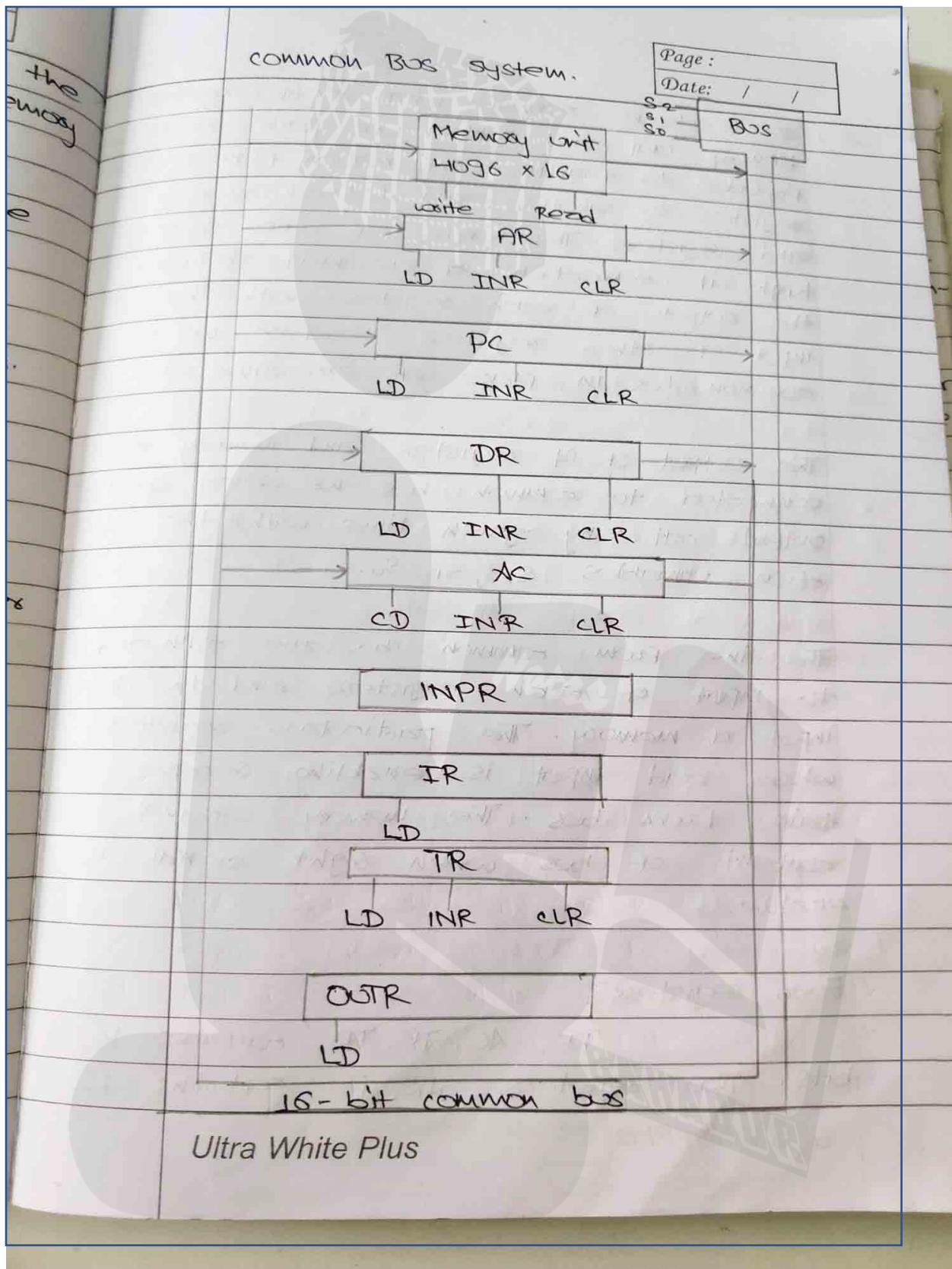
↳ Directly gives input to register B.

We cannot directly move data to register B. There is no such command.

LHDL → used to get local 16-bit data.

INR B → increase B by 1.

Ultra White Plus



The basic computer has eight registers. Memory unit and control unit paths must be provided to transfer information from one register to another and between memory and registers. The number of wires is very high if connections are made between the outputs of each registers with the input of other registers. Therefore, we use common bus in place of excessive wires.

The output of 4 registers and memory are connected to common bus. We select specific outputs at any given time with the selection variables  $S_2, S_1, S_0$ .

The lines from common bus are connected to input of each registers and the data input of memory. The particular register whose load input is enabling receives data from bus. The memory receives content of bus when write signal is enable.

Four registers!

DR, AC, IR, TR contains 16 bits. Two registers AR, PC contains 12 bits. Ultra White Plus

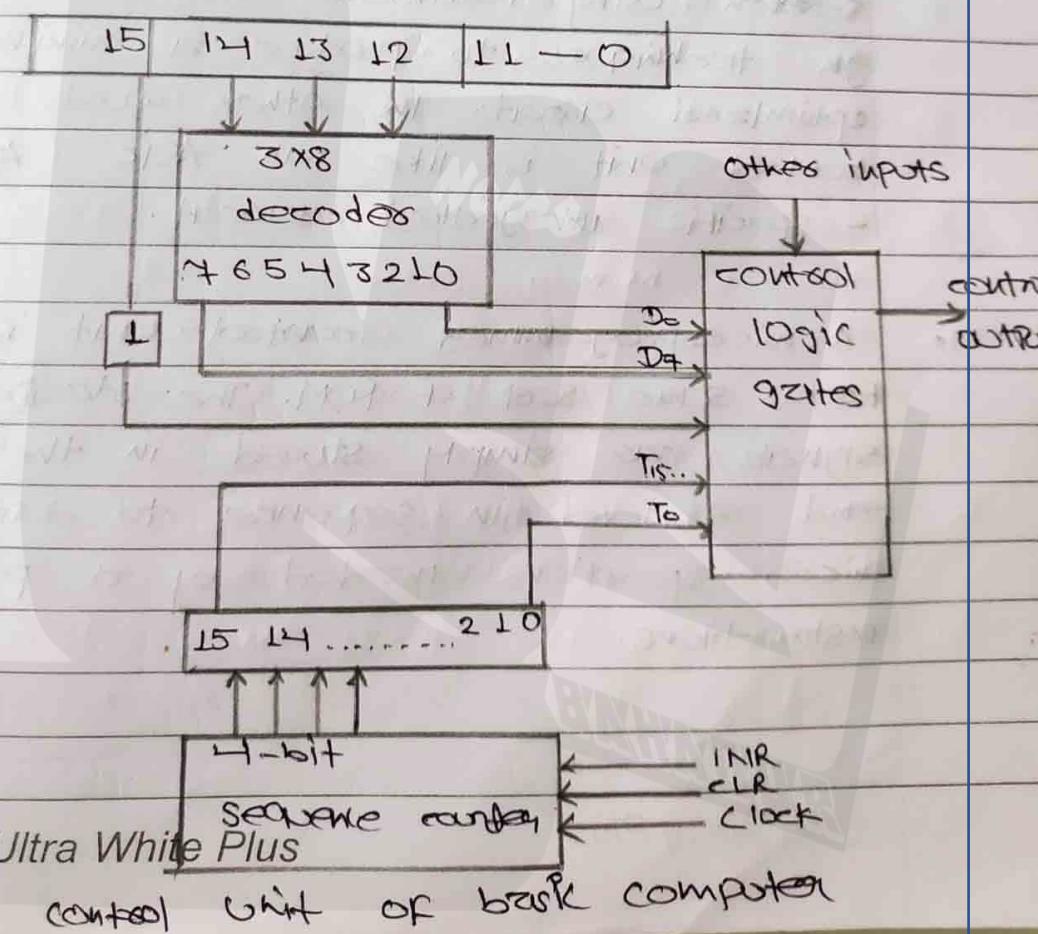
bits as they only hold memory address.

The input registers INPR and output registers OUTR have 8-bits.

INTR : INTR is connected to provide information to bus.

OUTR : OUTR only receive info from bus.

Timing and control



Page :	/	/
Date:		

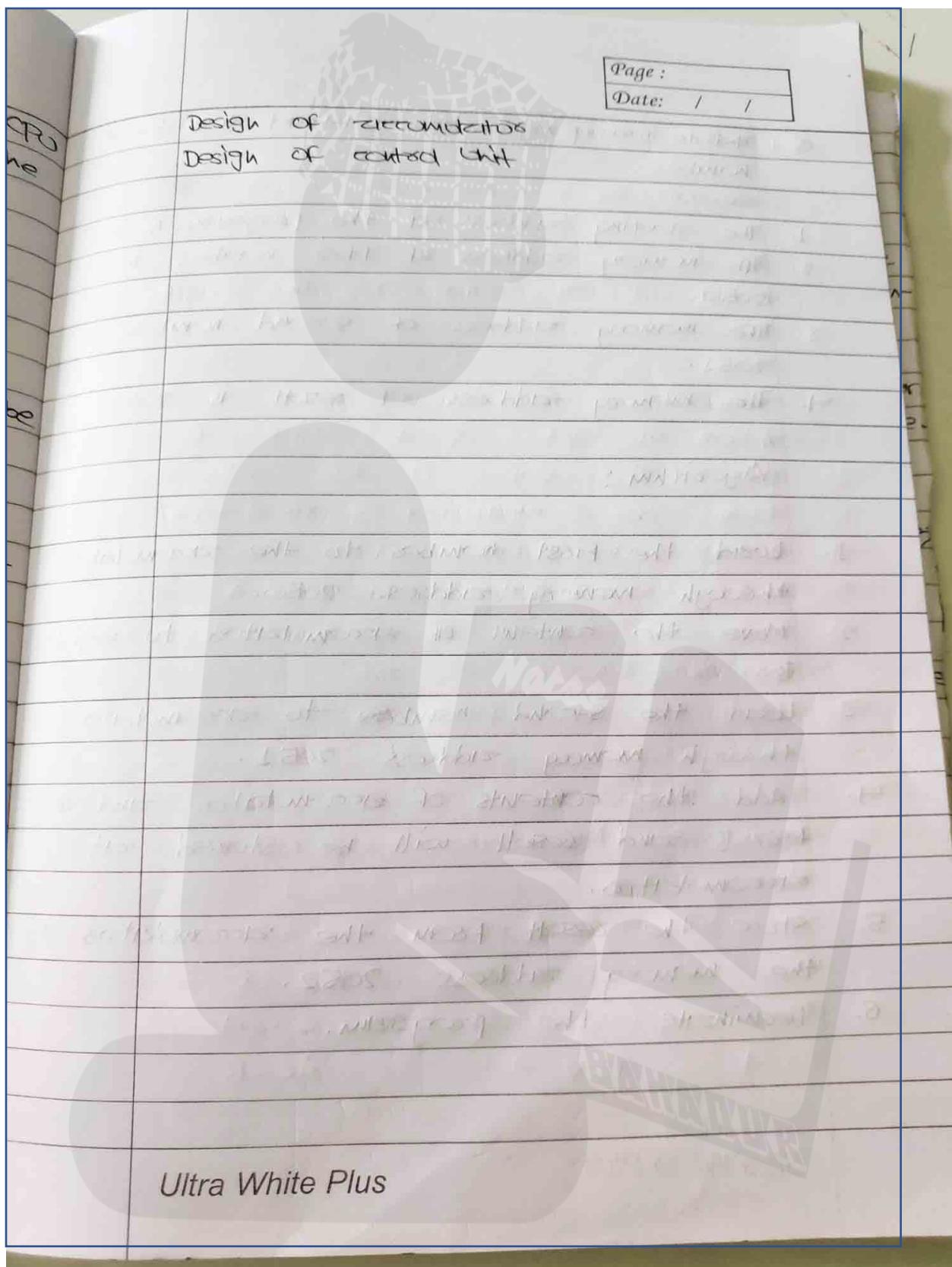
All sequential circuits in basic computer are driven by a master clock, with the expectation of INPR registers.

At each clock pulse, the clock unit sends control signals to control inputs of bus, the registers, and ALU.

control unit design and implementation can be done by two general methods.

- A hardware control unit is designed from scratch using traditional digital logic design techniques to produce a minimal, optimized circuit. In other words, the control unit is like an ASIC (Application-specific integrated circuit).
- A microprogrammed control unit is built from some sort of ROM. The desired control signals are simply stored in the ROM, and retrieved in sequence to drive the micro operations needed by a particular instruction.

Ultra White Plus



Page:	
Date:	/ /

Q.1. Write a program to subtract two 8-bit numbers.

1. The starting address of the program is 2000.
2. The memory address of first number is 2050.
3. The memory address of second number is 2051.
4. The memory address of result is 2052.

Algorithm :

1. Load the first numbers to the accumulator through memory address 2050.
2. Move the content of accumulator to register B.
3. Load the second numbers to accumulator through memory address 2051.
4. Add the contents of accumulator and register B and result will be stored at accumulator.
5. Store the result from the accumulator to the memory address 2052.
6. Terminate the program.

Ultra White Plus

Program :			Page : / /
Memory address	Mnemonics	comment	
2000	LDA 2050	AC - [2050]	
2003	MOV B, A	BL - A	
2004	LDA 2051	AC - [2051]	
2007	SUB B	AC - A+B	
2008	STA 2052	2052 <- A	
200B	HLT	Terminate	

Explanation :

1. LDA 2050 : This instruction will load the numbers from memory to the accumulator.
2. MOV B, A : This instruction will move the content of accumulator to the register B.
3. LDA 2051 : This instruction will load the numbers from memory to the accumulator.
4. SUB B : this instruction will store the content of the accumulator to the memory address 2052.
5. HLT : This instruction will terminate the program.

Ultra White Plus

Q. What to multiply two 8-bit numbers?

Sol:

Assumptions:

Assume that the first number stored at registers B, and second number is stored at registers C. And result may not have any carry.

Algorithm:

1. Assign the value of A to register B.
2. Assign the value of C to register C.
3. Move the content of B in A.
4. Rotate accumulator left without carry.
5. Rotate accumulator left without carry.
6. Store the content of accumulator at memory address 3050.
7. Half of the program.

Program:

Memory Address	Mnemonics	Comments
----------------	-----------	----------

2000	MVI B 05	B < -05
2002	MVI C 04	C < -04
2004	MOV A, B	A < -B

Ultra White Plus

		Page : / /
2005	RLC	Rotate the content of X without carry.
2006	RLC	Rotate the content of X without carry.
2007	STX 3050	3050 LDA
2008	HLT	End of the program.

**Explanation :**

1. MVI B 05 : assign the value 05 to B registers.
2. MVI B 04 : assign the value 04 to C registers.
3. MOV A,B : Move the content of registers B to registers A.
4. RLC : Rotate the content of accumulator left without carry.
5. ~~RLC~~ STX 3050 : Store the content of register X at memory location 3050.
6. HLT : stops the execution of the program.

Ultra White Plus

Date: / /

Q. What is the algorithm to multiply two 16-bit numbers?

Assumptions:

1. Starting address of program 2000.
2. Input memory location: 2050, 2051, 2052, 2053.
3. Output memory location: 2054, 2055, 2056, 2057.

Algorithm:

1. Load the first data in HL Pairs.
2. Move content of HL pairs to stack pointers.
3. Load the second data in HL pairs and move it to DF.
4. Make H register 2100H and L register 00H.
5. Add HL Pairs and stack pointers.
6. check for carry if carry increment it by 1 else move to next step.
7. Then move F to X and perform OR operation with accumulator and registers D.
8. The value of operation is zero, then stop the value else goto step 3.

Ultra White Plus

Program:	Mnemonics	Comments.
Memory address		
2000	LHLD 2050	Load H-L Pairs with address 2050.
2003	SPHL	Save it in stack Pointers.
2004	LHLD 2052	Load H-L Pairs with address 2052.
2007	XCHQ	Exchange HL and DE pairs content
2008	LXI H, 0000H	HL - 00H, LL - 00H
200B	LXI B, 0000H	BL - 00H, CL - 00H
200E	DAD SP	
200F	JN (2013)	Jump not carry
2012	INX B	Increment BC By 1.
2013	DCX D	Decrement DE by 1.
2014	MOV A, E	AL - E
2015	OR AP	OR the content of accumulators & D Register
2016	JNZ 200E	Jump not zero.
2019	SHLD 2054	LL - 2053, HL - 2054.
201C	MOV L, C	LC - C
201D	MOV H, B	BL - H
201E	SHLD 2056	LL - 2055, HL - 2056
2021 Ultra White Plus	HLT	Terminate the program.

Page / /  
Date / /

EXPLANATION:

Registers B, C, D, E, H, L and accumulators  
are also used for general purpose.

1. LHLD 2050 : Load HL pair with address 2050
2. SDHL : Give the content of HL in stack pointers.
3. LHLD 2052 : Load H-L pair with address 2052.
4. XCHG : Exchange the content of HL pair with DE.
5. LXI H, 000H : Make H as 00H and L as 00H.
6. LXI B, 000H : Make B as 00L and C as 00H.
7. DAD SP : ADD HL pair and stack pointers.
8. JNC 2013 : Jump to address 2013 if there will be no carry.
9. INX B : increments BC registers with L.
10. DCX D : decrements DE registers pairs by L.
11. MOV A,C : move the content of registers E to accumulators.
12. ORA 0 : OR the content of accumulators and D registers.
13. JNZ 200E : Jump to address 200E if there will be no 2050.

Ultra White Plus

- Page : / /  
Date: / /
- L4. SHLD 2054 : store the result to memory  
address 2054 and 2055 from  
HL Pairs registers.
- L5. MOV L,C : move the content of registers  
C to L.
- L6. MOV H,B : move the content of registers  
B to H.
- L7. SHLD 2056 : store the result to memory  
address 2056 and 2057 from  
the Pairs registers.
- L8. HLT : terminates the program.

Store program concept :

Any program which  
we can store if we want and it can  
and execute.

Computer Architecture :

Computer architecture  
refers to those attributes of a system  
that have a direct impact on logical  
execution of a program.

E.g:- Instruction set

2- The number of bits used to  
represent values digital type.

3- Input / output mechanism.

Ultra White Plus

## Date: / /

### 4- Memory addressing techniques.

Computer organization?

Computer organization refers to the operational unit and their interconnections that realize the architectural specifications. e.g. size: things transparent to the programmer.

E.g.-1: Interface between computer and peripherals.

E.g.-2: The memory technology being used.

### Memory Hierarchy

Capacity, cost and speed of different types of memory play a vital role while designing a memory system for computer.

- If the memory has larger capacity more application will get space to run smoothly.
- It's better to have fastest memory as far as possible to achieve a greater performance more over for the practical

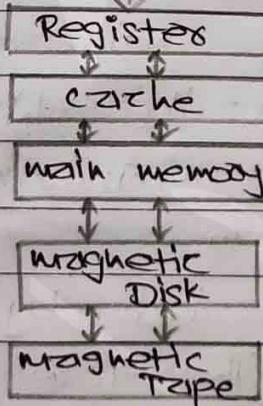
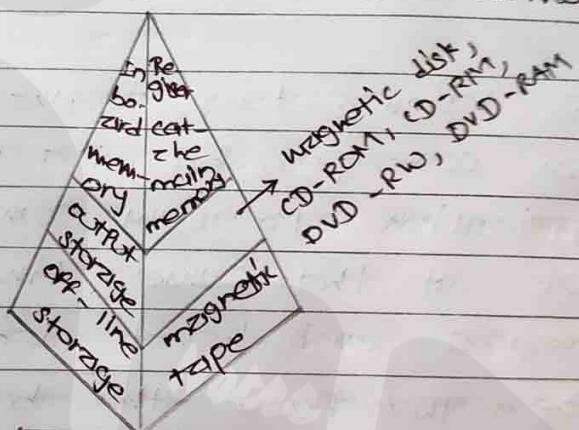
**Ultra White Plus**

- Date: / /
- \* Problem: What are the factors for memory hierarchy?
1. There is no relation between these three characteristics cost, capacity and access time. The cost increases with these conditions in same memory hierarchy because (i) If capacity increases access time increases that due to which cost per bit decreases (ii) If access time decreases (increasing bytes and due to which cost per bit increases).
  2. The designer takes to increase capacity because cost per bit decreases and the more application program can be accommodated but at the same time access time increases and hence decreases the performance. Therefore the best idea will be used memory hierarchy. Memory hierarchy is to obtain the highest possible access speed while minimizing the total cost of memory system but all accumulated information is needed by the CPU at the same time. Therefore it is more economical to used less cost storage device to serve as a backup for storing the

Ultra White Plus

Page :  
Date: / /

information that is not directly used by CPU. The memory unit that directly communicate with CPU is called main memory and devices that provides backup storage are called auxiliary memory. The memory hierarchy system consist of all storage devices employed in a computer system from the slow by high capacity auxiliary memory to a relatively faster to main memory and to an even smaller and faster cache memory.



### Memory Hierarchy

Ultra White Plus

Page :

Date: / /

## Registers :

Registers is the temporary storage location in the CPU that are designed for storing data temporarily during program execution. It quickly accepts, store and transfer data and instruction that are used immediately.

## Cache memory :

Cache memory is a high speed memory that resides between the CPU and the RAM in a computer. Cache memory stores data and instruction that the CPU is likely to need next. Moving data between RAM and CPU is one of the time consuming operation, simply because RAM is much more slower than the CPU. A partial solution to this problem is to include a cache memory in the CPU. When a program is running and the CPU needs to read a piece of data or program instructions from RAM, the CPU checks first to see whether the data is in cache memory. If the data is not there, the CPU reads the data from RAM into its

Ultra White Plus

Date: / /

registers, but it also loads a copy of the data into cache memory. The next time the CPU needs the data, it finds it in the cache memory and saves times having to load the data from RAM.

Since the late 1980's, most PCs had cache memory built-in to them, the CPU-resident cache is often called level -1 (L<sub>1</sub>) cache. To add even more speed to modern CPUs, an additional cache is added to the CPU's. This cache is called level -2 (L<sub>2</sub>) cache.

#### Features of Cache memory.

- Cache memory is faster than main memory.
- It consumes less access time as compared to main memory.
- It stores the program that can be executed.

Ultra White Plus

Date: / /

Main memory :-

The main memory in a computer is called Random Access Memory. It is also known as RAM. This is the part of the computer that stores operating system software. Software applications and other information for the central processing unit (CPU) to have fast and direct access when needed to perform tasks. It is called "Random access" because the CPU can go directly to any section of main memory; and does not have to go about the process in a sequential order.

RAM is one of the faster types of memory, and has the capacity to allow data to be read to be written. When the computer is shut down, all of the content held in RAM is lost. Main memory is available in two types: Dynamic Random Access Memory (DRAM) and static Random Access Memory (SRAM).

DRAM :- Dynamic Random Access Memory (DRAM) is the most common kind of main memory in a computer. DRAM is

Ultra White Plus

Page: / /  
Date: / /

constantly reading whatever information  
being held in memory.

- ii. SRAM + It is the second type of main memory in a computer. SRAM is random access memory (RAM) that retains data bits in its memory as long as power is being supplied.

### DRAM

1. Inexpensive.
2. Slower than SRAM.
3. Can store many bits per second.
4. Uses less power.
5. Generates less heat.
6. Used for main memory.
7. constructed of tiny capacitors that leak electricity.
8. Requires a recharge every few milliseconds to maintain its data.

### SRAM

1. Expensive.
2. Faster than DRAM.
3. cannot store many bits per second.
4. Uses more power.
5. Generates more heat.
6. used for main memory.
7. constructed of circuits similar to D flip-flop.
8. Holds its contents as long as power is available.

Ultra White Plus

## Unit - 3.

Page :

Date: / /

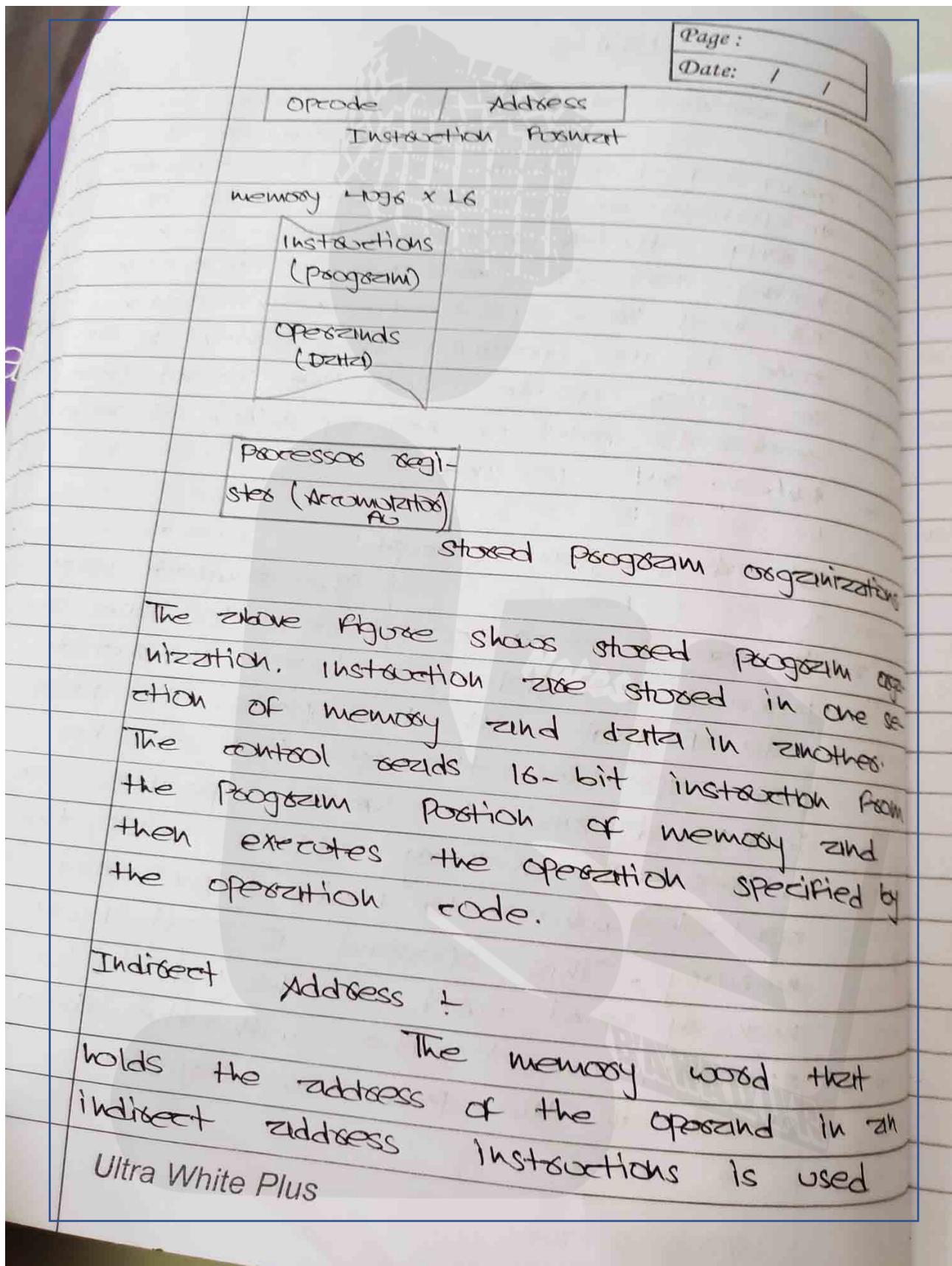
## Instruction code ?

An instruction code is a group of bits that instant the computer to perform a specific operations. It is usually divided into two parts each having its own particular interpretation. The most basic part of an instruction code is its operation part which is also called opcode. And the second part represents data or memory which is also known as operand.

## Stored program concept ?

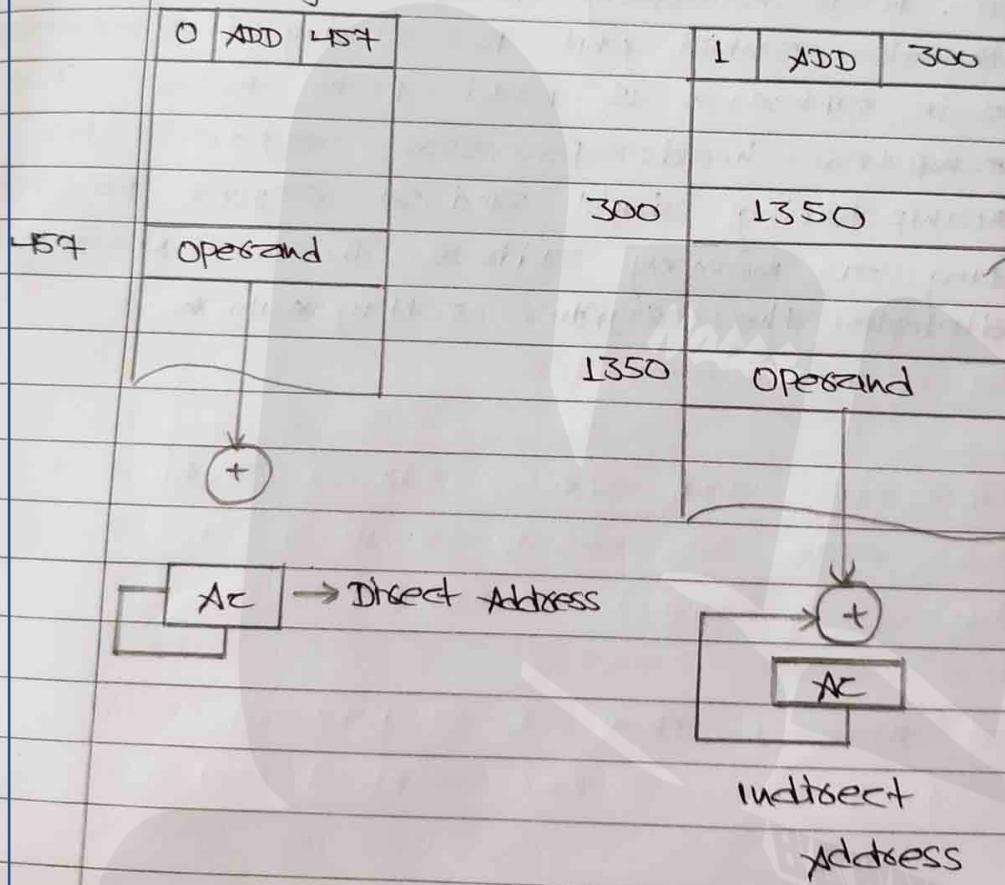
The simplest way to organize a computer is to have one processor, registers, and an instruction code format with two parts. The first part specifies the operation to be performed and the second specifies an address. The memory address tells the control where to find an operand in memory. This operand is read from memory and used as the data to be operated on together with the data stored in the processor registers.

Ultra White Plus



is a pointer to an address of data when the second part of the instruction code specifies an operand the instruction is said to have an immediate operand. In an indirect address, the second part of the instruction design at an address of memory word in which the address of the operand is found.

memory



Page :  
Date: / /

'D' represents direct addressing mode and 'I'  
represents indirect addressing mode.

Computer registers :

Computer instructions are normally stored in consecutive memory locations and are executed sequentially one at a time. The control sends an instruction from a specific address in memory and executes it. It is necessary to provide the registers in the control unit for storing the instruction code after it is read from memory. The computer needs processes, registers for manipulating data and 21 registers for holding a memory address. These requirements dictate the register configuration.

Computer Registers				Page : / /
* List of registers for the Basic Computer.				
Registers sym.	No. of bits	Registers name	Function.	
DR	16	Data Register	Holds memory operand.	
AR	12	Address reg.	Holds address for memory.	
AC	16	Accumulator	Processor register.	
IR	16	Instruction reg.	Holds instruction code.	
PC	12	Program counter.	Holds address of instruction.	
TR	16	Temporary Registers	Holds temporary data.	
INPR	8	Input Registers	Holds input characters.	
OUTR	8	Output registers	Holds output characters.	

Ultra White Plus

Page : / /  
Date: / /

- Control memory
1. The function of control unit in a digital computer is to initialize sequence of microoperations.
2. When the control signals are generated by hand size using conventional logic design techniques, the control unit is said to be hard wired.
3. Microprogramming is a second alternative to designing the control unit of a digital computers.
- i) The principle of microprogramming is an elegant and systematic method for controlling the microoperation sequences in a digital computer.
4. A control unit whose binary control variables are stored in memory is called microprogrammed control unit.
5. A memory that is a part of control unit is referred as a control memory.
- i. Each word in a control memory

Ultra White Plus

Page :	/	/
Date:	/	/

contains within it a micro instruction.

ii) A sequence of microinstructions constitute a micro-programmed.

iii) can be either read only memory or writable control memory (Dynamic micro-programming)

4. A computer that employs a micro-program control unit will have two separate memories. \* main memory and a control memory.

\* Microprogrammed :-  
Program stored in memory that generates all the control signals required to execute the instruction set correctly. It consists of micro instructions.

\* Micro-instruction :-  
It contains a control word and a sequencing word. Control word has all the control information required for 1 clock cycle and sequencing word has information needed to decide the next micro-instruction addresses.

Ultra White Plus

\* Microprogrammed sequences : The next address generator is sometimes called a microprogram sequence as it determines the address sequence that is read from control memory. The function of microprogrammed sequence is implemented by loading the control address registers by an address from control memory, specifying an external addresses and loading an initial address to start the control operations.

### Microprogrammed control unit

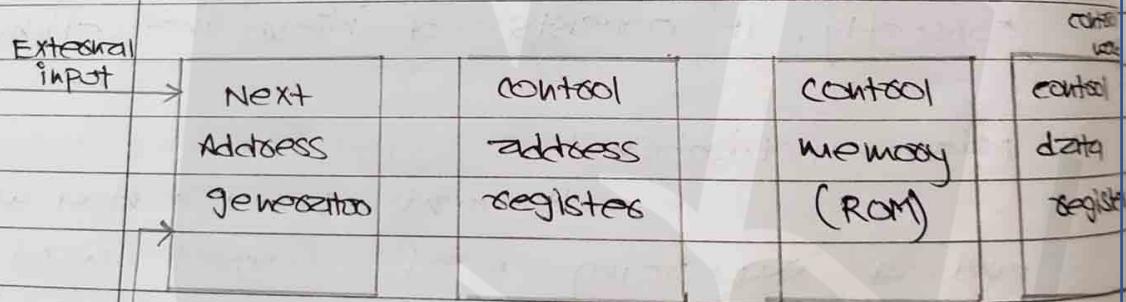


Fig 2: Microprogrammed control unit.

Ultra White Plus

## \* Addressing sequencing

Page : / /  
Date: / /

1. Microinstructions are stored in control memory in groups, with each group specifying a routine.
2. Each computer instruction has its own programmed routine in control memory to generate the micro-operations that execute the instruction.
3. To appreciate the address sequencing in a microprogrammed control unit.
  - i. An initial address is loaded into the control address registers when power is turned on in the computer.
  - ii. This address is usually the address of the first micro-instruction that activates the instruction phase routine.
  - iii. The control memory next must go through the routine that determines the effective address of the operand.

iv. The next step is to generate the micro operations that executes the instruction fetched from memory.

4. The transformation from the instruction code bits to an address in control memory where the routine is located is referred as a mapping process.

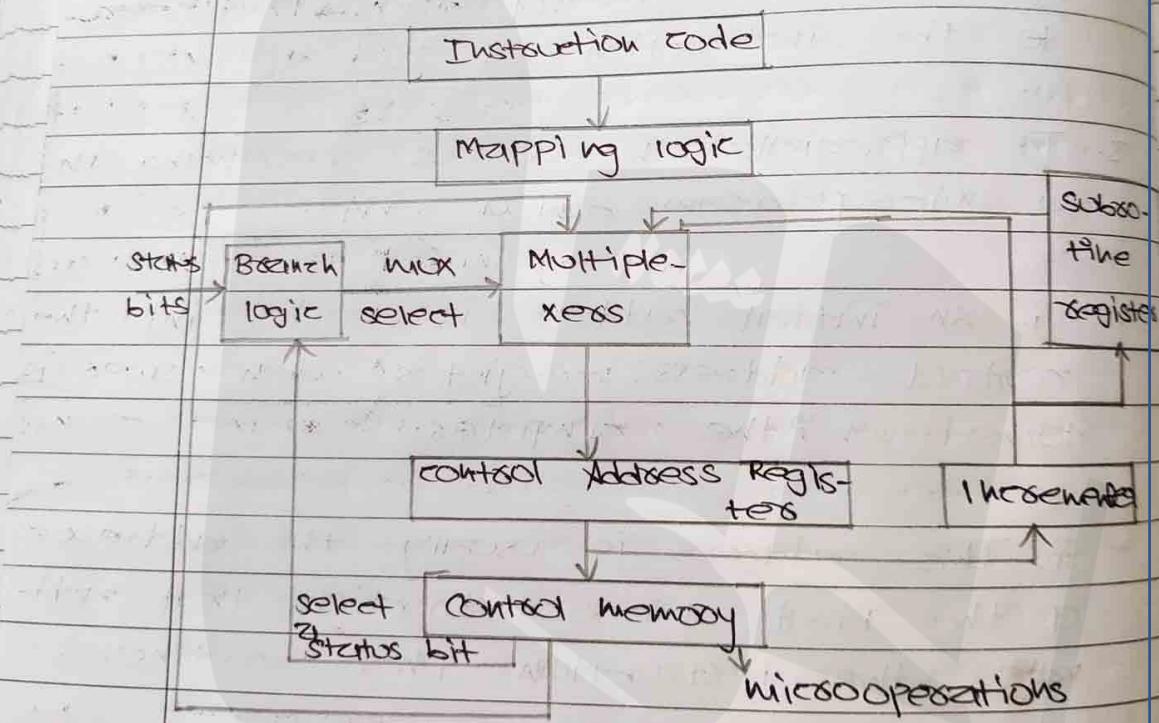


Fig : Selection of address from control memory.

<p style="text-align: right;">Page: / /</p> <p style="text-align: right;">Date: / /</p> <p>* conditional Branching</p>
<p>I. The status condition are special bits in the system that provides parameters information. Eg the carry bit, the sign bit, the move bits and input or output status.</p> <p>II. The status bit together with the field in the micro instruction that specifies a branch address control the conditional branch which is implemented in the branch logic.</p> <p>III. The branch logic hardware may be implemented by multiplexes.</p> <ul style="list-style-type: none"> <li>a. Branch to the indicated address if the condition is true.</li> <li>b. Otherwise the address registered is incremented.</li> </ul> <p>IV. An unconditional branch micro instruction can be implemented by loading the branch address from control memory into the control address register.</p>

Ultra White Plus

Page : / /  
Date: / /

V. If condition is true then branch (return from the next address field) of the current micro instruction) else fall through.

VI. condition to test overflow negative zero carry etc.

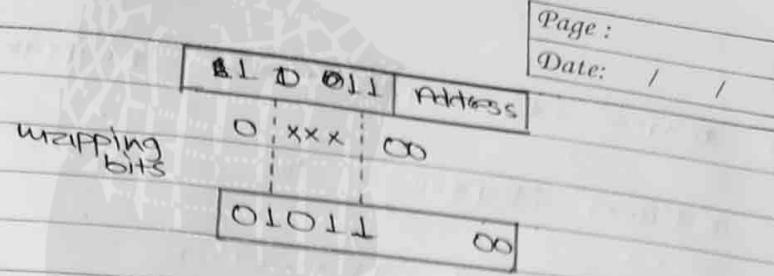
#### \* Mapping of instructions

1. \* Special type of branch exist when a micro instruction specifies a branch to the first word program routine for which instruction is located.

2. The status bits for this type of branch are the bits in the operation code part of the instruction.

3. One simple mapping process that converts the four bit operation code to a seven bit address for control memory.

*Ultra White Plus*



- a. Placing a zero in a MSB (most significant bit) of the address.
- b. Transferring the four operation code bits.
- c. Clearing the two least significant bit (LSB) of the control address registers.
- d. This provides for each computer instruction or microprogram routine with the capacity of four microinstructions.
  - i. If the routine needs more than four microinstruction it can use addresses 10000000 through 111111 if it uses fewer than four microinstructions, the unused memory location would be available for other routines.
5. One can extend this concept to a more general mapping rule by using a ROM or PLD (Programmable Logic

*Ultra White Plus*

Page : / /  
Date: / /

device) to specify the mapping function.

### Direct Mapping

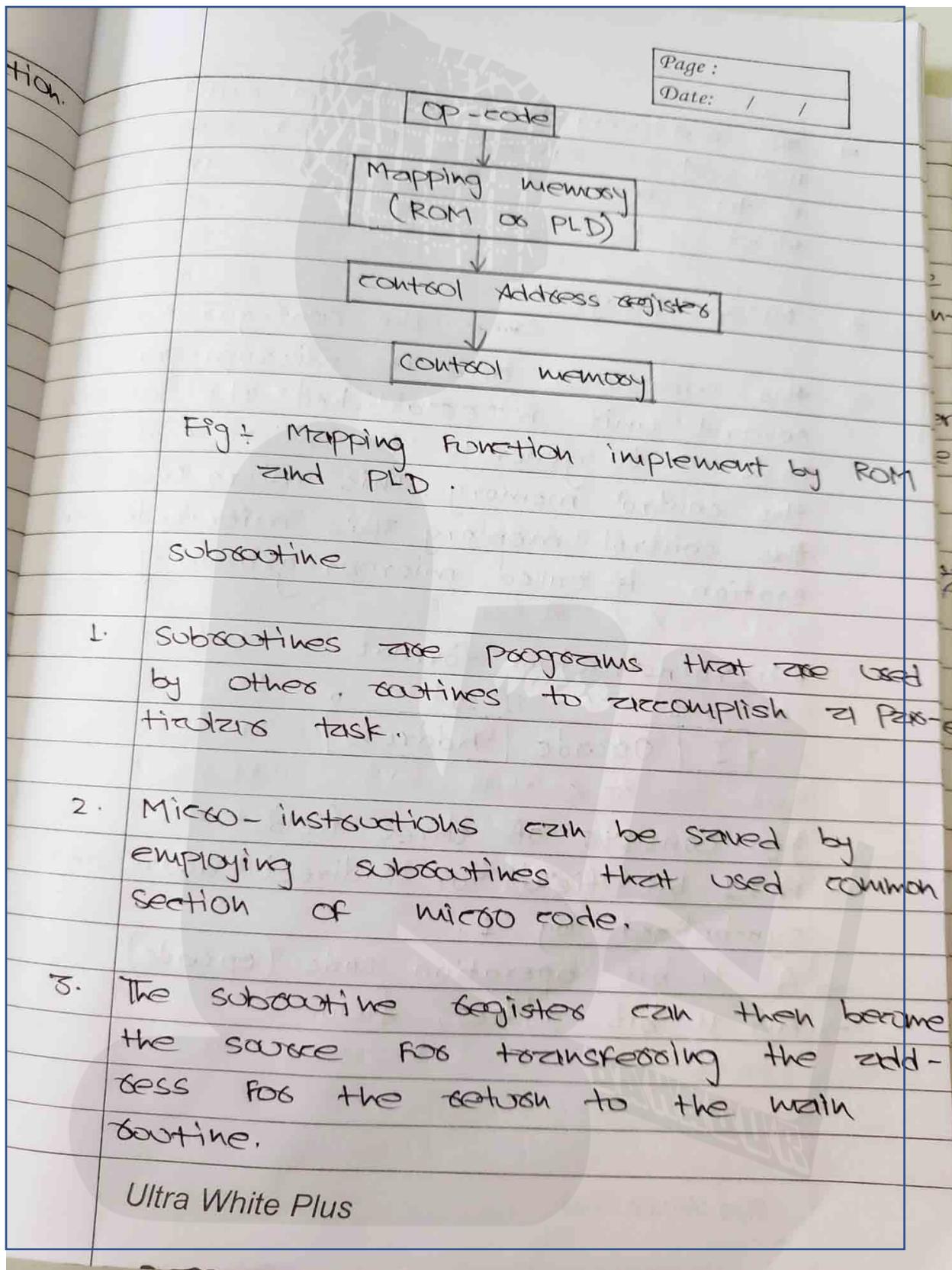
#### OP-codes of Instructions

		Address	
ADD	0000	→ 0000	ADD Routine
AND	0001	→ 0001	AND Routine
LDA	0010	→ 0010	LDA Routine
STA	0011	→ 0011	STA Routine
BUN	0100	→ 0100	BUN Routine
			control storage

Mapping Bits      10 | xxxx | 010

	Address	
→ 10	0000   010	ADD Routine
→ 10	0001   010	AND Routine
→ 10	0010   010	LDA Routine
→ 10	0011   010	STA Routine
→ 10	0100   010	BUN Routine

Ultra White Plus



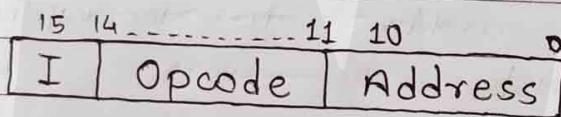
Page: / /  
Date: / /

4. The best way to structure a register file that stores addresses for sub-routines is to organize the registers in LIFO stack.

\* Micro-Program:

Once the configuration of the computer and its microprogram control unit is established the designer's task is to generate the micro code for the control memory. This micro code for the control memory. This microcode generation is called microprogramming.

Microinstruction format



It consists of three fields

- A 1-bit field for indirect addressing symbolized by I.
- A 4 bit operation code (opcode)
- An 11 bit address field.

Ultra White Plus

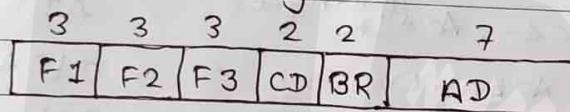
Page :  
Date: / /

Lists of four of the 16 possible memory reference instructions.

Symbol	Opcode	Description
ADD	0000	$AC \leftarrow AC + M(EA)$
BRANCH	0001	If ( $AC < 0$ ) then ( $PC \leftarrow EA$ )
STORE	0010	$M(EA) \leftarrow AC$
EXCHANGE	0011	$AC \leftarrow M(EA), M(EA) \leftarrow AC$

### Micro instruction Format

- The microinstruction format for the control memory.



$F_1, F_2, F_3$  : Operation Field

CD : Condition for branching

BR : Branch field

AD : Address field

### Micro operations

- i) The three bits in each field are encoded to specify 7 distinct micro operations

a. No more than 3 micro operations can

Ultra White Plus

Page :	/	/
Date:	/	/

- be chosen for a micro instructions.
- b. If fewer than 3 microoperations are used one or more of the fields will use the binary code 0000 for no operation.
  - ii. It is important to realize that 2 or more conflicting microoperations cannot be specified simultaneously.

FL	Microoperation	Symbol
000	None	NOP
001	$AC \leftarrow AC + DR$ <small>(Data Registers)</small>	ADD
010	$AC \leftarrow 0$	CLRAC
011	$AC \leftarrow AC + L$	INCAC
100	$AC \leftarrow DR$	DRTAC
101	$AR \leftarrow DR (0-10)$	DRTAR
110	$AR \leftarrow PC$	PCTAR
111	$M[AR] \leftarrow DR$	WRITE

Ultra White Plus

F <sub>2</sub>		Microoperation	Symbol
000	None		NOP
001	AC $\leftarrow$ AC - DR		SUB
010	AC $\leftarrow$ AC VDR		OR
011	AC $\leftarrow$ AC + DR		AND
100	DR $\leftarrow$ M[AR]		READ
101	DR $\leftarrow$ AC		ACTDR
110	DR $\leftarrow$ DR + L		INCDR
111	DR (0-10) $\leftarrow$ PC		PCTDR

F <sub>3</sub>		Microoperation	Symbol
000	None		NOP
001	AC $\leftarrow$ AC $\oplus$ DR		XOR
010	AC $\leftarrow$ $\overline{AC}$		COM
011	AC $\leftarrow$ SHL AC		SHL
100	AC $\leftarrow$ SHR AC		SHR
101	PC $\leftarrow$ PC + L		INCPC
110	PC $\leftarrow$ AR		ARIPC
111	Reserved		

Ultra White Plus

Page : / /  
Date: / /

CD	condition	symbol	comments
00	Always = 1	U	unconditional branch
01	DR (15)	I	Indirect address bit
10	AC (15)	S	sign bit of AC
11	AC = 0	Z	zero value in AC

BR	symbol	Function
00	JMP	CAR ← AD if condition = 1 CAR ← CAR + L if condition = 0
01	CALL	CAR ← AD, SBR ← CAR + L if condition = 1 CAR ← CAR + L if condition = 0
10	RET	CAR ← SBR (Return from subr- outine)
11	MAP	CAR(2-5) ← DR (11-14) CAR (0, 1, 6) ← 0

### Symbolic Micro-Instruction :

1. Symbols are used in micro-instruction as assembly language.

Ultra White Plus

Page :	
Date:	/ /

2. The simplest and most straightforward way to formulate an assembly language for a microprogram is to define symbols for each field of the micro-instruction and to give users the capability of defining their own symbolic addresses.
3. A symbolic micro-program can be translated into its binary equivalent by microprogramme assembler.

#### Sample Format

Five field : Label ; micro-ops ; CD ; BR ; AD

4. The label field may be empty or it may specify a symbolic address terminated with a colon.
5. The micro-operation field of one, two, or three symbols separated by commas.
6. The CD field uses one of the letters [U, I, S, Z] where U is unconditional branch, I indirect Address bit, S side of AC, Z zero value in AC.

Ultra White Plus

## BR - Byte registers

Page :  
Date: / /

4. The BR field contains one of the four symbols [Jmp, CALL, RET, MAP]
8. The AD field specifies a value for the address field of the micro-instructions with one of the [symbolic address, NEXT, empty]

## Binary Microprogram

1. The symbolic microprogram must be translated to binary either by means of an assembler program or by the user if the microprogram is simple.
2. The equivalent binary form of the microprogram is listed in table below!
3. Even though address three is not used, some binary value example all zeros must be specified for each word in control memory.
4. However, if some unforseen error occurs or if a noise signal sets CAR to the value of three, it will be wise to jump to address 64.

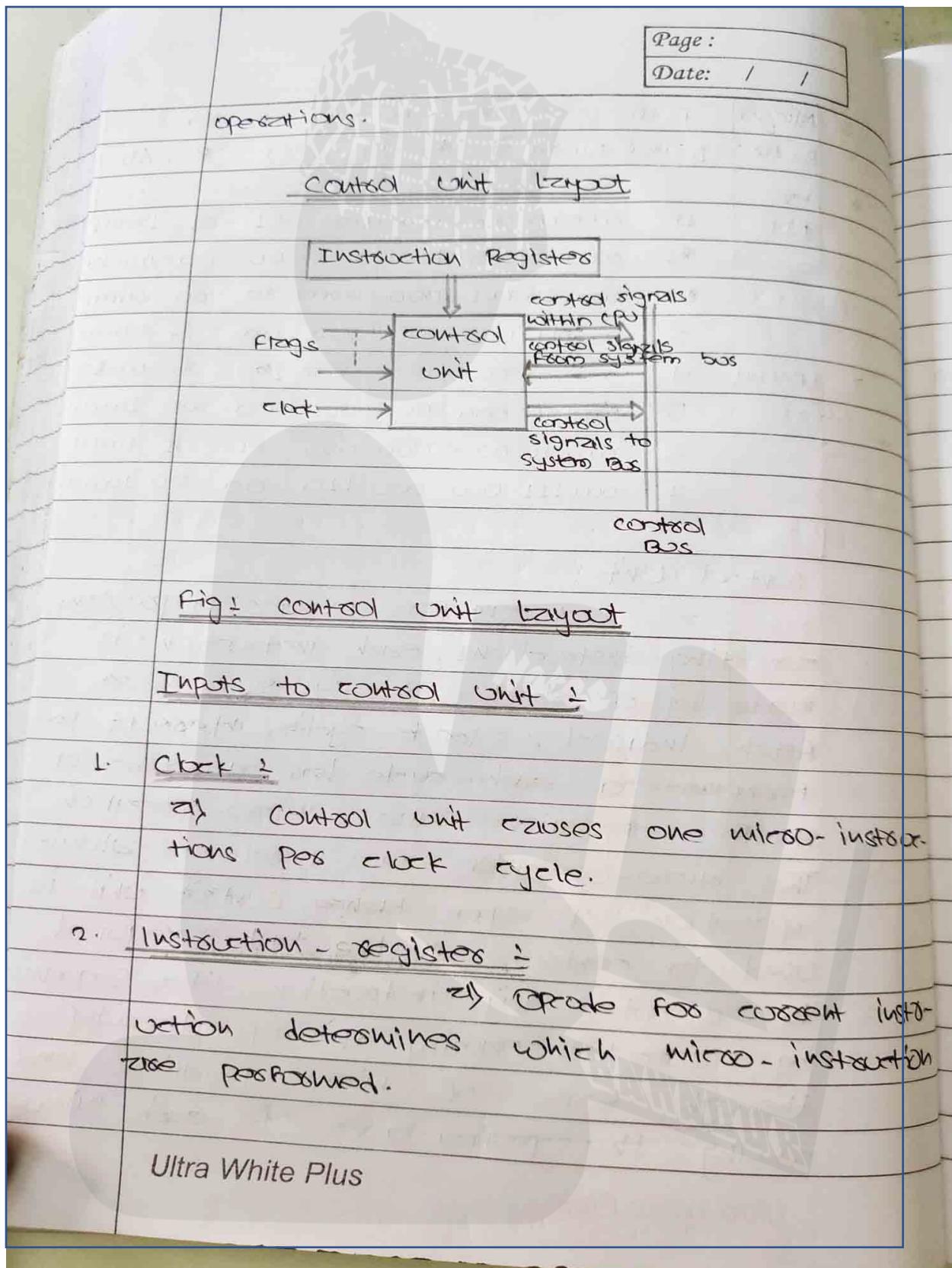
Ultra White Plus

Micro Routine	Address	Binary micro-instruction							
		Decimal	Binary	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	CD	BR	AD
ADD	0	00000000	000	000	000	0L	0L	1000011	
	1	00000000	000	100	000	00	00	0000000	
	2	00000000	000	100	0T00000	00	00	1000000	
	3	00000000	000	1T00000	000	00	00	1000000	
BRANCH	4	0000100000	000	000	000	100	00	0000000	
	5	0000101000	000	000	000	00	00	1000000	
	6	0000110000	000	000	000	0L	0L	1000011	
	7	0000111000	000	110	00	00	00	1000000	

control unit :

A computer executes a program consisting instructions, each instruction is made up of shorter sub-cycles such as fetch, Indirect, execute cycle, interrupt. Performance of each cycle has a number of shorter operation called micro-operations. Thus, micro-operations are functional atomic operation of CPU. Hence, control unit is used to control all the tasks required to execute any instruction like sequencing, execution, causing the CPU to step through series of micro-operations and causing the performance of each micro-

Ultra White Plus



Page :	
Date:	/ /

3. Flags :

- a) State of CPU.
- b) Results of previous operations

4. Control signal from Control Bus :

- a) Interrupts
- b) Acknowledgement

5. Control Unit Output :

- a) Within CPU
  - \* cause data movement
  - \* Activate specific ALU function
- b) Through control bus
  - \* To memory
  - \* To input / output modules

Design of control unit :

1. The bits of the micro-instruction are usually divided into fields with each field defining a distinct separate function.
2. The various fields encountered in instruction format provide;
3. Control bits to initiate micro-operation in the system.

Ultra White Plus

- Page : / /  
Date: / /
- b. Special bits to specify the way that next address is to be evaluated.
  - c. An addressing field for branching.
  - d. The number of control bits that initiate micro-operation can be reduced by grouping mutually exclusive variables into fields by encoding the 'k' bits in each field to provide  $2^k$  micro-operations.
  - e. Each field requires a decoder to produce the corresponding control signals.

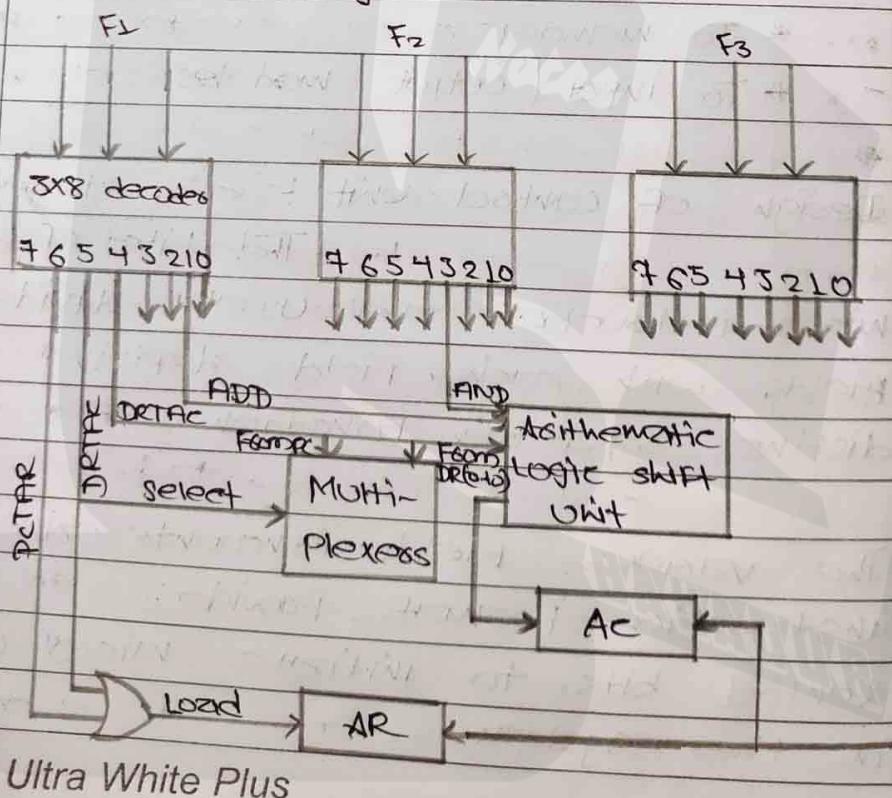
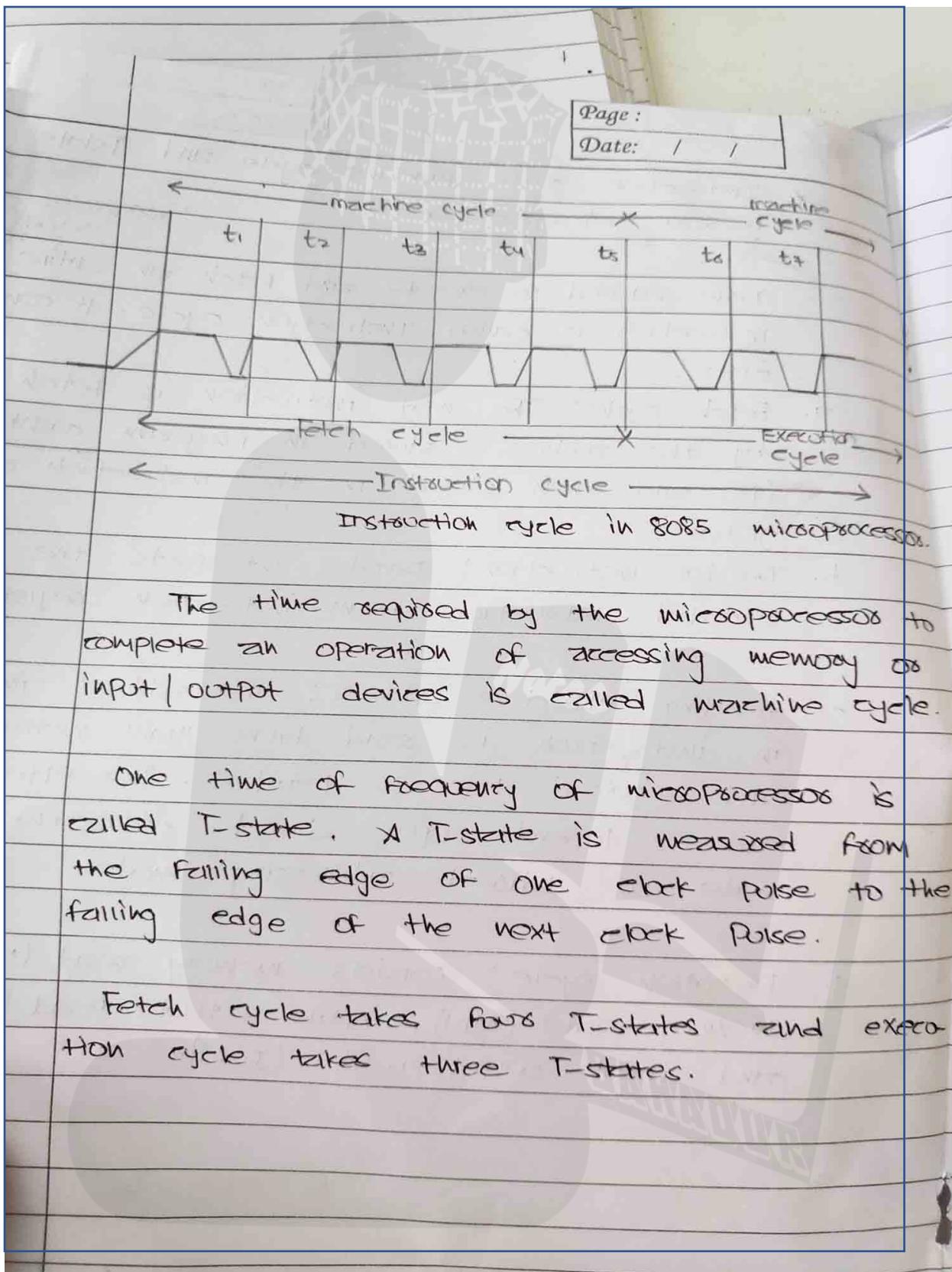


Fig 2. Decoding of control

\* Instructions cycle, machine cycle and T-state in 8085 microprocessors.

Time required to execute and fetch an entire instruction is called Instruction cycle. It consists:

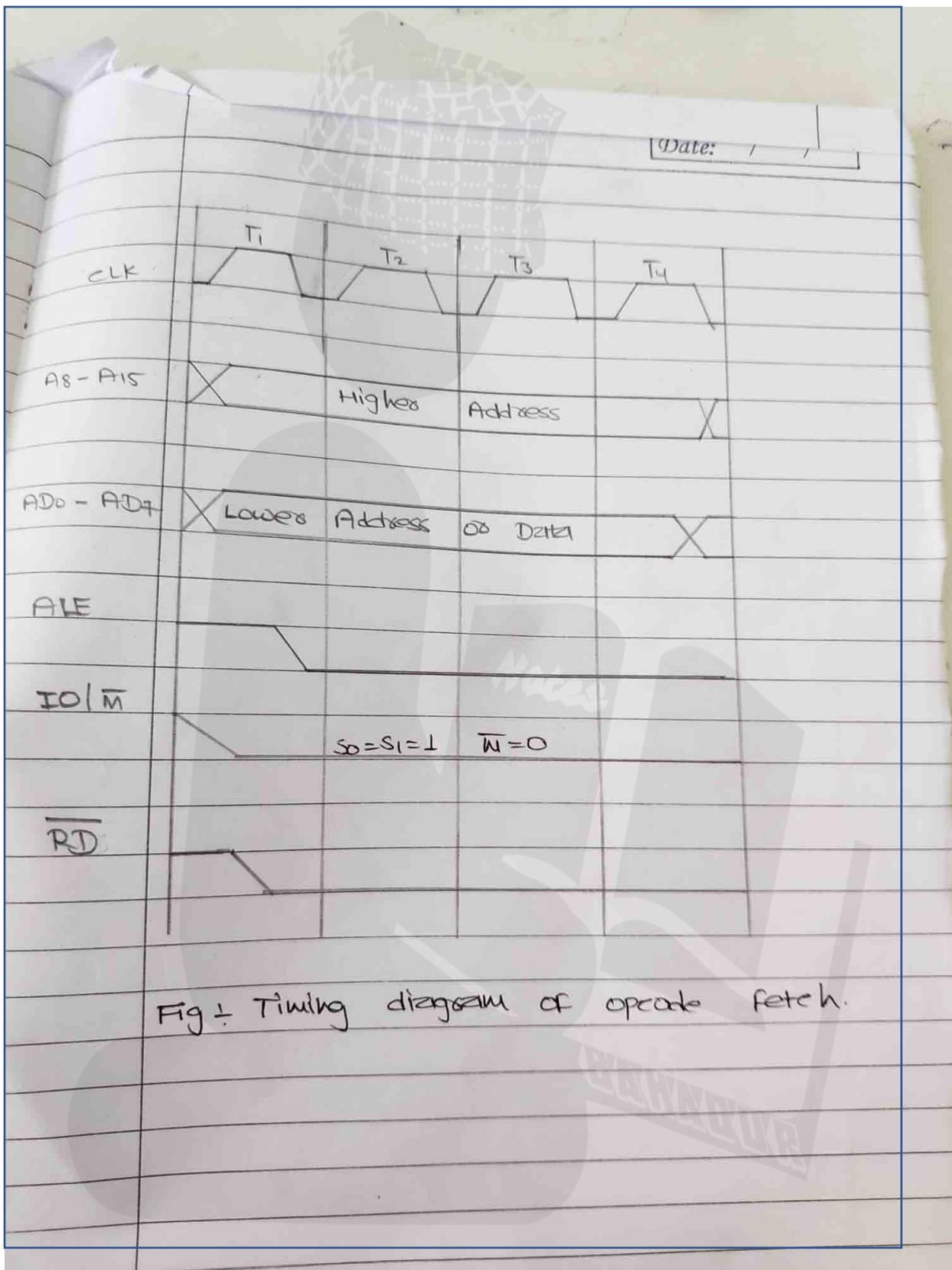
- Fetch cycle: The next instruction is fetched by the address stored in program counter (PC) and then stored in the instruction registers.
- Decode instruction: Decoder interprets the encoded instruction from instruction register.
- Reading effective address: The address given in instruction is read from main memory and required data is fetched. The effective address depends upon direct addressing mode or indirect addressing mode.
- Execution cycle: consists memory read (MR), memory write (MW), input output read (IOR) and input output write (IOW).



The time required by the microprocessor to complete an operation of accessing memory or input / output devices is called machine cycle.

One time of frequency of microprocessors is called T-state. A T-state is measured from the falling edge of one clock pulse to the falling edge of the next clock pulse.

Fetch cycle takes four T-states and execution cycle takes three T-states.



Page :	/
Date:	/ /

MVI B, 82H  
 MOV A,B  
 MOV C,X  
 MVI D, 34H  
 OUT PORT L  
 HLT

\* Map to ports & following

- load the no. 8BH in register D.
- Load the no. 6FH in register C.
- Increment the content of register C.
- Add the content of registers C and D.  
and display the sum at the output PORT L.  
SOPA

Ultra White Plus

Page :  
Date: / /

6. Recommendation (if requested) - NO new facts must be introduced here. On the basis of information presented in finding and conclusions, make some suggestion for next action. Remember that the writer of a report cannot make decision. He/ she can only suggest what action should be taken.
7. A closing section - A report should be signed and there should be a name and title shown at the foot, plus the date when the report was written.

Ultra White Plus

Page :  
Date: / /

A useful pattern for the sections is: to report at on / @ investigate on ... (subject) ... then as requested by ... (name and title) ... on ... (date) ...

3. Procedures - Give a brief description of the methods used to collect the information. Back-to-back interviews were held, visits made, questions issue use no. points is appropriate
4. Findings - This will be the longest section of the reports. Go through the procedure point by point and use no's and sub-headings for this section. On the each heading state what information was gathered at each stage
5. Conclusion :- No new facts must be introduced in this section. You must look at the finding and state the logical implications of them. What can you infer or conclude from the finding.

Ultra White Plus

Display gamma → OUT

Page :

Date: / /

\* Explain the different types of flags used in 8085?

D depending upon no. of bytes and no. of instruction operating.

The information may be presented in tables or graphics form and the writer must leave to produce clear conclusions and recommendation. All reports require the ability to record fax clearly as objectively, the ability to interpret information or make conclusion and the ability to present suggestion in which a situation may be improve.

The reports are often presented on the following prescribe series of headings:

1. Headings - There should be 2 headings in the report. "The name of the company" the report heading "report on BCA program of PKC".
2. Terms of references - Those section should state exactly while the report is being written.  
Why are you writing the report? What was requested? Who requested it? When were you ask to do it?

Ultra White Plus

MVI → data bytes  
 LDA → memory bytes  
 LXI →

Page :  
 Date: / /

Load the hexadecimal numbers 34H in register B and display the numbers at the output code table PORT 1.

SQ1

MVI B, 34H  
 MOV A,B  
 OUT PORT 1

Write instructions to load hexadecimal numbers 65H in register C and 92H in accumulator. Display the num. 65H at PORT 0 and 92H at PORT 1.

SQ1

MVI C, 65H → MOV A,C  
 MVI C, 92H → OUT PORT 0  
 MOV A,C  
 OUT PORT 1, 65H  
 OUT PORT 1, 92H  
 MVI A, 92H  
 OUT PORT 1.

Ultra White Plus