

295
14/11/23

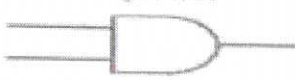





12
Nov-23

ANSWER KEY SET 1		QID : 1510230047		
DIGITAL ELECTRONICS AND MICROPROCESSORS 4032				
I	PART A	Split up	Sub total	total
1	1'S complement 0101001 2's complement 1+ 0101010	1 1	2	10
2	Theorem-1: Complement of a sum = product of complements $\overline{A+B} = \overline{A} \cdot \overline{B}$ Theorem-2: Complement of product = sum of the complements $\overline{A \cdot B} = \overline{A} + \overline{B}$	1 1	2	
3	i. Counters ii. registers iii. frequency divider iv. data transfer	0.5 x 4	2	
4	i. binary weighted resistor type ii. R-2R ladder type network	2 x 1	2	
5	Zero flag, sign flag, carry flag, parity flag, auxiliary carry flag (any 2)	2 x 1	2	

II	PART B	Split up	Sub total	total
1		3	6	
2		3	6	

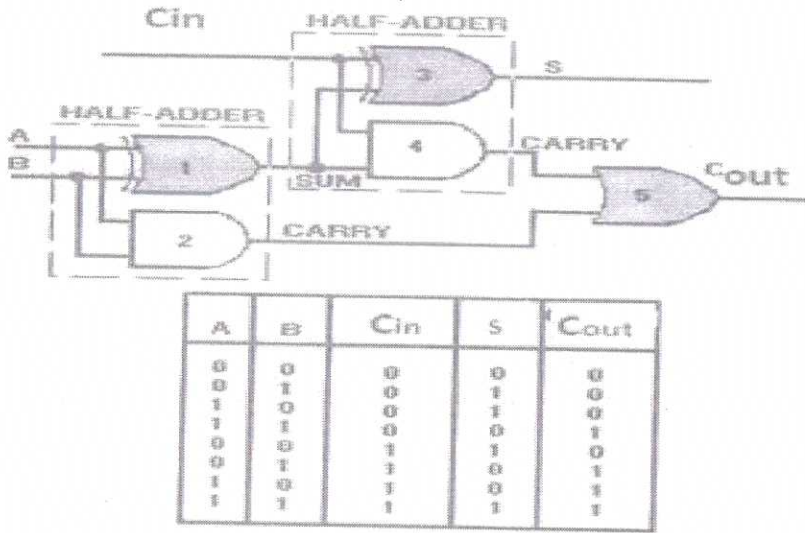
3	<p>Select inputs S_1 S_0</p>	6	6																						
4	<p>In JK F/F, when $J = K = 1$ and $Q = 0$, the output will change from 0 to 1 after the time interval Δt. The output will oscillate back and forth between 0 and 1 in the duration t_p of clock pulse, so at the end of clock pulse the value of Q is ambiguous. This is called race around condition.</p>	4	6																						
5	<table border="1"> <thead> <tr> <th>Sl. no.</th> <th>Asynchronous counter</th> <th>Synchronous counter</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>All the flip-flops are not clocked simultaneously.</td> <td>All the flip-flops are clocked simultaneously.</td> </tr> <tr> <td>2</td> <td>Output of first flip-flop drives the clock for second flip-flop, the output of second drives the third and so on.</td> <td>there is no interconnection between output of one flip-flop and clock of next flip-flop.</td> </tr> <tr> <td>3</td> <td>The settling time is cumulative sum of individual flip-flops.</td> <td>The settling time is equal to highest settling time of all flip-flops.</td> </tr> <tr> <td>4</td> <td>Asynchronous counter is known as serial counter</td> <td>Synchronous counter is known as parallel counter.</td> </tr> <tr> <td>5</td> <td>Its design and implementation is very simple.</td> <td>Design and implementation becomes tedious and complex as the number of states increases.</td> </tr> <tr> <td>6</td> <td>slow in speed</td> <td>faster in speed</td> </tr> </tbody> </table>	Sl. no.	Asynchronous counter	Synchronous counter	1	All the flip-flops are not clocked simultaneously.	All the flip-flops are clocked simultaneously.	2	Output of first flip-flop drives the clock for second flip-flop, the output of second drives the third and so on.	there is no interconnection between output of one flip-flop and clock of next flip-flop.	3	The settling time is cumulative sum of individual flip-flops.	The settling time is equal to highest settling time of all flip-flops.	4	Asynchronous counter is known as serial counter	Synchronous counter is known as parallel counter.	5	Its design and implementation is very simple.	Design and implementation becomes tedious and complex as the number of states increases.	6	slow in speed	faster in speed	6x1'	6	
Sl. no.	Asynchronous counter	Synchronous counter																							
1	All the flip-flops are not clocked simultaneously.	All the flip-flops are clocked simultaneously.																							
2	Output of first flip-flop drives the clock for second flip-flop, the output of second drives the third and so on.	there is no interconnection between output of one flip-flop and clock of next flip-flop.																							
3	The settling time is cumulative sum of individual flip-flops.	The settling time is equal to highest settling time of all flip-flops.																							
4	Asynchronous counter is known as serial counter	Synchronous counter is known as parallel counter.																							
5	Its design and implementation is very simple.	Design and implementation becomes tedious and complex as the number of states increases.																							
6	slow in speed	faster in speed																							

6	<p>TRAP: It is a non-maskable interrupt, having the highest priority among all interrupts. By default, it is enabled until it gets acknowledged. In case of failure, it executes as ISR and sends the data to backup memory.</p> <p>RST7.5: It is a maskable interrupt, having the second highest priority among all interrupts. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to 003CH address.</p> <p>RST 6.5: It is a maskable interrupt, having the third highest priority among all interrupts. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to 0034H address.</p> <p>RST 5.5: It is a maskable interrupt. When this interrupt is executed, the processor saves the content of the PC register into the stack and branches to 002CH address.</p> <p>INTR: It is a maskable interrupt, having the lowest priority among all interrupts. It can be disabled by resetting the microprocessor. The microprocessor checks the status of INTR signal during the execution of each instruction. When the INTR signal is high, then the microprocessor completes its current instruction and sends active low interrupt acknowledge signal.</p>	1.5 1 1 1 1.5	6	
7	<ol style="list-style-type: none"> 1. It is an 8-bit microprocessor i.e. it can accept, process, or provide 8-bit data simultaneously. 2. It operates on a single +5V power supply connected at Vcc; power supply ground is connected to Vss. 3. 8085 microprocessor requires two-phase, 50% duty cycle, TTL clock. These clock signals are generated by an internal clock generator. 4. The maximum clock frequency of 8085 microprocessors is 3MHz whereas minimum clock frequency is 500 KHz. 5. 8085 microprocessor provides 16 address lines, therefore it can access $2^{16} = 64K$ bytes of memory. 6. It provides 8 bit I/O addresses to access (2^8) 256 I/O ports. 7. In 8085, the lower 8-bit address bus (A0 -A7) and data bus (D0 -D7) are Multiplexed to reduce number of external pins. But due to this, external hardware (latch) is required to separate address lines and data lines. 8. 8085 microprocessor has five hardware interrupts: TRAP, RST 5.5, RST 6.5, RST 7.5, INTR. 9. The hardware interrupt capability of 8085 microprocessor can be increased by providing external hardware.. 10. It supports 74 instructions. 	Any 6 6 x 1	6	

PART C				Split up	Sub total	total																																																																															
III a	<p>Gate Name Symbol Notation Truth table</p> <p>AND  $F = A \cdot B$ or $F = AB$ <table border="1" data-bbox="965 347 1141 448"> <tr><th>A</th><th>B</th><th>AB</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table></p> <p>OR  $F = A + B$ <table border="1" data-bbox="965 459 1141 560"> <tr><th>A</th><th>B</th><th>A+B</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table></p> <p>NOT  $F = \bar{A}$ or $F = A'$ <table border="1" data-bbox="973 571 1133 638"> <tr><th>A</th><th>F</th></tr> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </table></p> <p>NAND  $F = \overline{(A \cdot B)}$ <table border="1" data-bbox="957 660 1133 772"> <tr><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table></p> <p>NOR  $F = \overline{(A + B)}$ <table border="1" data-bbox="965 772 1133 884"> <tr><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table></p> <p>XOR  $F = \bar{A} \oplus B$ $F = \bar{A}B + A\bar{B}$ <table border="1" data-bbox="965 884 1133 996"> <tr><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </table></p>	A	B	AB	0	0	0	0	1	0	1	0	0	1	1	1	A	B	A+B	0	0	0	0	1	1	1	0	1	1	1	1	A	F	0	1	1	0	A	B	F	0	0	1	0	1	1	1	0	1	1	1	0	A	B	F	0	0	1	0	1	0	1	0	0	1	1	0	A	B	F	0	0	0	0	1	1	1	0	1	1	1	0	6 x 1	6	
A	B	AB																																																																																			
0	0	0																																																																																			
0	1	0																																																																																			
1	0	0																																																																																			
1	1	1																																																																																			
A	B	A+B																																																																																			
0	0	0																																																																																			
0	1	1																																																																																			
1	0	1																																																																																			
1	1	1																																																																																			
A	F																																																																																				
0	1																																																																																				
1	0																																																																																				
A	B	F																																																																																			
0	0	1																																																																																			
0	1	1																																																																																			
1	0	1																																																																																			
1	1	0																																																																																			
A	B	F																																																																																			
0	0	1																																																																																			
0	1	0																																																																																			
1	0	0																																																																																			
1	1	0																																																																																			
A	B	F																																																																																			
0	0	0																																																																																			
0	1	1																																																																																			
1	0	1																																																																																			
1	1	0																																																																																			
III b	<p>$(8645)_{10} \rightarrow \begin{array}{r} 8 \overline{) 86} \\ \underline{80} \\ 6 \\ \underline{64} \\ 2 \end{array}$ $0.4518 : 3.60 \rightarrow 3$ $0.60 \times 8 = 4.80 \rightarrow 4$ $0.80 \times 8 = 6.40 \rightarrow 6$ $0.40 \times 8 = 3.20 \rightarrow 3$ ✓</p> <p>$(86.45)_{10} = (126.3463)_{8}$ $[2 \times 3 = 6 \text{ carries}]$</p> <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <p>$(28.A2)_{16}$ $= 2 \times 16^1 + 8 \times 16^0 + A \times 16^{-1} + 2 \times 16^{-2}$ $= 2 \times 16 + 8 \times 1 + 10 \times \frac{1}{16} + 2 \times \frac{1}{256}$ $= (40.632)_{16}$</p> </div> <p>$(61.625) = 111101.101$</p>	3 3 3	9																																																																																		

IV a	<p>(i) Propagation delay: The time required for the output of a digital circuit to change states after a change at one or more of its inputs. The speed of a digital circuit is specified in terms of the propagation delay time.</p> <p>(ii) Power dissipation: It is specified in terms of power consumption per gate and is the product of supply voltage and supply current.</p> <p>(ii) Speed- power product: The speed of a logic circuit can be increased, ie. The propagation delay can be reduced, at the expense of power dissipation.</p> <p>(iv) Fan out: <i>Fan-out (output load factor)</i> is the maximum number of inputs that can be driven by a logic gate.</p>	2 2 2 2	8	
IV b	<p>ADVANTAGES:</p> <ul style="list-style-type: none"> ➤ TTL family is the fastest saturating logic family. ➤ It has good noise immunity. ➤ TTL gates are compatible with other logic families. ➤ Good fan-out; TTL gates can drive up to 10 gates ➤ TTL gates exhibit low output impedance for high/low states. <p>DIS ADVANTAGES:</p> <ul style="list-style-type: none"> ➤ TTL gates cannot be used in applications where large noise voltages exist. ➤ Power dissipation of TTL gates is much higher than that of MOS gates. ➤ Cost of TTL gates is higher. <p>CMOS ADVANTAGES</p> <ol style="list-style-type: none"> 1. Extremely large fan out 2. Lowest power dissipation 3. Very high noise immunity and noise margin 4. Lowest propagation delay 5. Single power supply is required <p>CMOS disadvantages</p> <ol style="list-style-type: none"> 1. Increase in cost in design 2. Packing density is less 	3 2 2 2	7	

Va



$S = A \oplus B \oplus Cin$
 $Cout = (A \oplus B)Cin + AB$

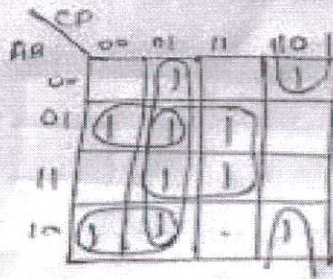
4

7

3

Vb

$f(A,B,C,D) = \sum m(1,2,4,5,7,8,9,10,13,15)$



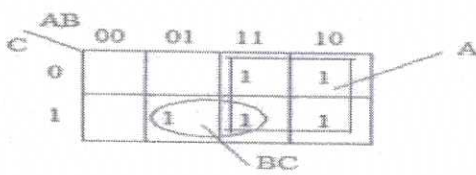
$f(A,B,C,D) = BD + \bar{C}D + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{B}C\bar{D}$

4

8

4

Vi
a



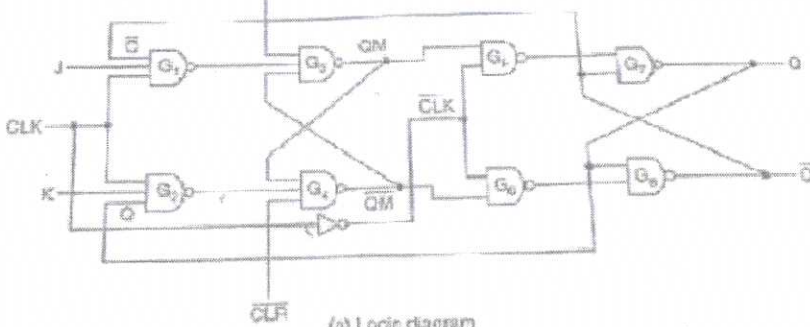
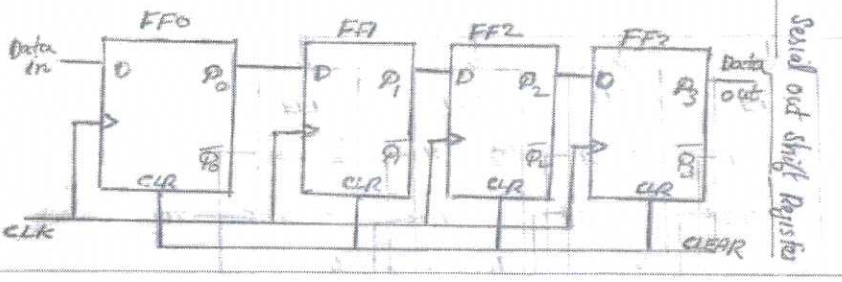
$Y = A + BC$

$f(ABC) = ABC + AB\bar{C} + A\bar{B}C + \bar{A}BC + A\bar{B}\bar{C}$
 $= AB(C + \bar{C}) + A\bar{B}(C + \bar{C}) + \bar{A}BC$ [C + C̄ = 1]
 $= AB.1 + A\bar{B}.1 + \bar{A}BC$ [A.1 = A]
 $= A(B + \bar{B}) + \bar{A}BC$
 $= A.1 + \bar{A}BC = A + \bar{A}BC$
 $= (A + \bar{A})(A + BC)$ [Distributive law, A + (BC) = (A+B)(A+C)]
 $= A + BC$

4

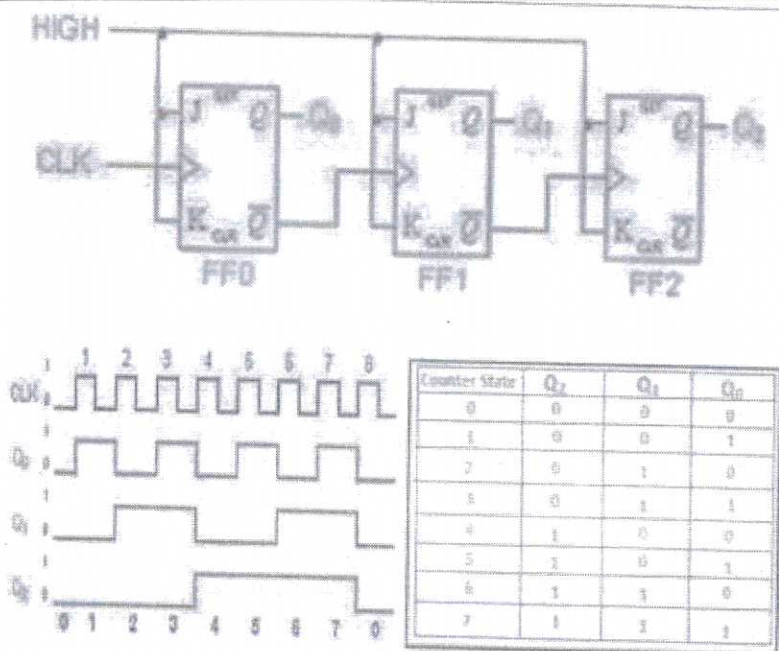
8

4

<p>Vi b</p>	<p>Here one perform as master and the other slave and overall circuit is known as master slave. when ever the clock is high the master is active. according to the state of input the out put of master is set or reset at this state the slave is inactive and output remains in previous state. when ever the input clock is low the slave is active. the final output is the output of the slave</p>  <p>(a) Logic diagram</p> <table border="1" data-bbox="510 884 933 1097"> <thead> <tr> <th colspan="3">Inputs</th> <th>Output</th> <th>Comments</th> </tr> <tr> <th>J</th> <th>K</th> <th>C</th> <th>Q</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> <td>Q₀</td> <td>No change</td> </tr> <tr> <td>0</td> <td>1</td> <td></td> <td>0</td> <td>RESET</td> </tr> <tr> <td>1</td> <td>0</td> <td></td> <td>1</td> <td>SET</td> </tr> <tr> <td>1</td> <td>1</td> <td></td> <td>\bar{Q}_0</td> <td>Toggle</td> </tr> </tbody> </table>	Inputs			Output	Comments	J	K	C	Q		0	0		Q ₀	No change	0	1		0	RESET	1	0		1	SET	1	1		\bar{Q}_0	Toggle	<p>2 3 2</p>	<p>7</p>	
Inputs			Output	Comments																														
J	K	C	Q																															
0	0		Q ₀	No change																														
0	1		0	RESET																														
1	0		1	SET																														
1	1		\bar{Q}_0	Toggle																														
<p>Vii a</p>	 <p>Serial in Serial out Shift Register</p>	<p>4 3</p>	<p>7</p>																															

7

VII
b



Three flip flops are used to count eight values. high value is given to j and k so that flip flop will toggle at the application of each clock pulse. output of flip flop is connected to clock of next. Q0 is LSB Q2 is MSB

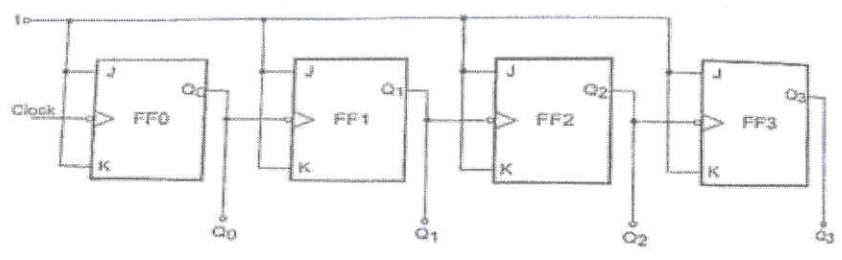
3

8

3

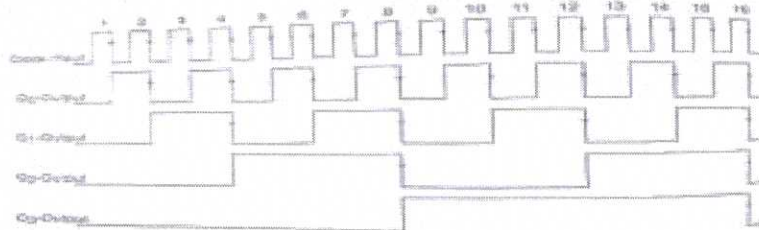
2

VIII
a



The logic diagram for a mod-16 asynchronous counter

9



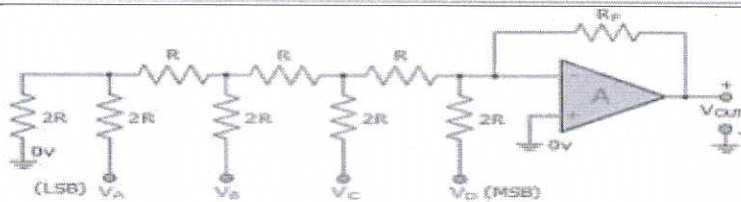
The timing diagram for a mod-16 asynchronous counter

After clock pulse	Q3	Q2	Q1	Q0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Counting sequence diagram

The J and K FFs are connected to HIGH so that all the FFs are in toggle mode. The external clock is connected to FF0 only whereas the clocks for other FFs are the Q outputs of the preceding FFs. So FF1 toggles for each clock pulses and provides LSB of the counting sequences. The FF1 toggles the next pulse after FF0 attains HIGH. Like that the FF2 and FF3 toggle only after their preceding FFs attains HIGH outputs.

VIII
b



As its name implies, the "ladder" description comes from the ladder-like configuration of the resistors used within the network. A R-2R resistive ladder network provides a simple means of converting digital voltage signals into an equivalent analogue output. Input voltages are applied to the ladder network at various points along its length and the more input points the better the resolution of the R-2R ladder. The output signal as a result of all these input voltage points is taken from the end of the ladder which is used to drive the inverting input of an operational amplifier

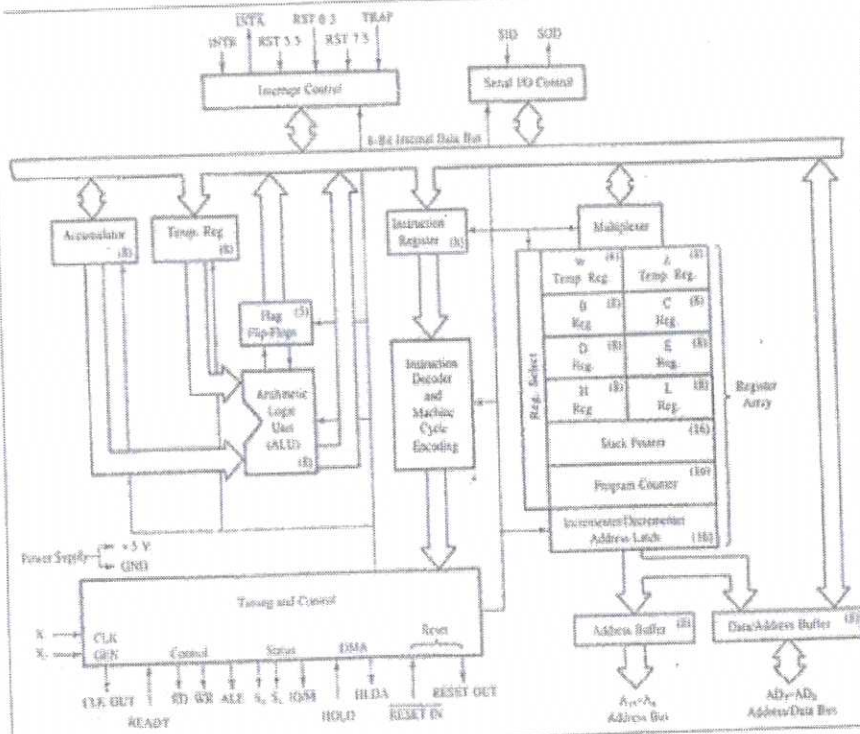
3

6

3

IX a	<p>The way of specifying data to be operated by an instruction is called addressing mode.</p> <p>Types of addressing modes –</p> <p>Immediate addressing mode: In immediate addressing mode the source operand is always data. Here data to be used is immediately given in the instruction itself. If the data is 8-bit, then the instruction will be of 2 bytes, if the data is of 16-bit then the instruction will be of 3 bytes. Eg:- MVI B,05 (move the data 05 immediately to register B)</p> <p>Register addressing mode: here data/operand to be operated is in the general purpose register. therefore the operation is performed within various registers. Eg:- MOV A,B (move the content of register B to register A)</p> <p>Direct Addressing Mode: In direct addressing mode, the data to be operated is available inside a memory location and that memory location is directly specified as an operand. The operand is directly available in the instruction itself. Eg:- LDA F500 (load the content of memory location to the accumulator)</p> <p>Indirect addressing mode: data to be operated is available inside a memory location and that memory location is indirectly specified by a register pair. Eg:- EDAX B (move content of B-C register to accumulator)</p> <p>Implied/Implicit/Inherent Addressing Mode: In implied/implicit addressing mode the operand is hidden and the data to be operated is available in the instruction itself. Eg:- CMA (finds and stores the 1's complement of the contents of accumulator, A in A) RRC (rotate accumulator A right by one bit)</p>	Any 3	6	3 x 2
---------	--	----------	---	----------

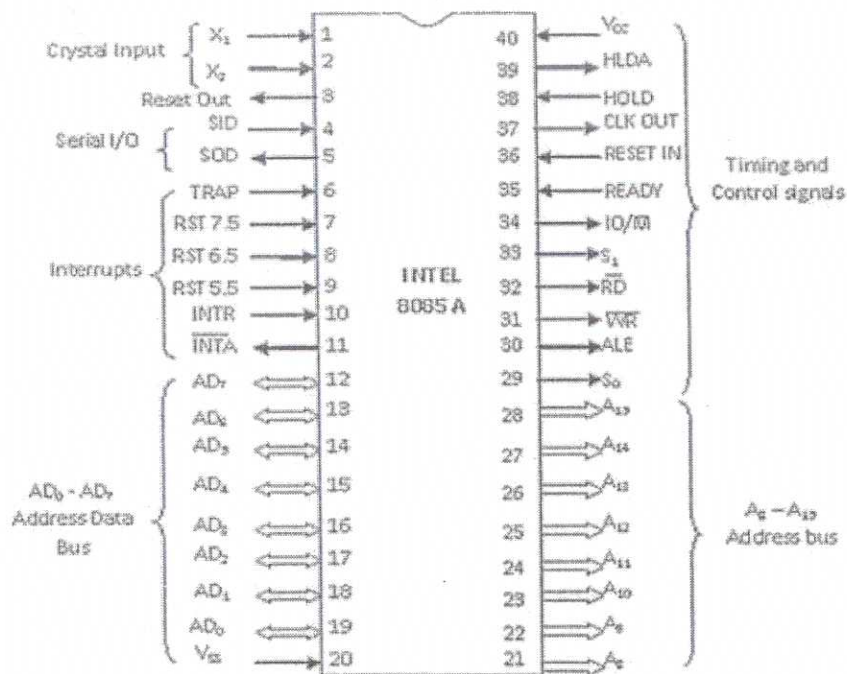
IX
b



9

9

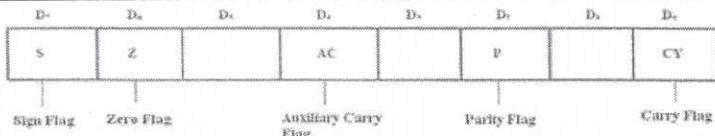
X a



Pin diagram of 8085 microprocessor

9

9

	<p>8085 Pin description</p> <ul style="list-style-type: none"> Higher Order Address pins- $A_{15} - A_8$ <ul style="list-style-type: none"> The address bus has 8 signal lines $A_8 - A_{15}$ which are unidirectional. Lower Order Address/ Data Pins- AD_7-AD_0 <ul style="list-style-type: none"> These are time multiplexed pins and are de-multiplexed using the pin ALE So, the bits $AD_0 - AD_7$ are bi-directional and serve as $A_0 - A_7$ and $D_0 - D_7$ at the same time. <ul style="list-style-type: none"> During the execution of the instruction, these lines carry the address bits during the early part, then during the late parts of the execution, they carry the 8 data bits. In order to separate the address from the data, we can use a latch to save the value before the function of the bits changes. Control Pins - RD, WR <ul style="list-style-type: none"> These are active low Read & Write pins Status Pins - ALE, IO/M (active low), S_1, S_0 <ul style="list-style-type: none"> ALE (Address Latch Enable)-Used to de-multiplex AD_7-AD_0 IO/M - Used to select I/O or Memory operation S_1, S_0 - Denote the status of data on data bus Interrupt Pins - TRAP, RST7.5, RST 6.5, RST 5.5, INTR, INTA <ul style="list-style-type: none"> These are hardware interrupts used to initiate an interrupt service routine stored at predefined locations of the system memory. Serial I/O pins - SID (Serial Input Data), SOD (Serial Output Data) <ul style="list-style-type: none"> These pins are used to interface 8085 with a serial device. Clock Pins- $X_1, X_2, CLK(OUT)$ <ul style="list-style-type: none"> X_1, X_2 - These are clock input pins. A crystal is connected between these pins such that $f_{crystal} = 2f_{8085}$ where $f_{crystal}$ = crystal frequency & f_{8085} = operating frequency of 8085 CLK(OUT) - This is an auxiliary clock output source 		
X b	 <ul style="list-style-type: none"> Zero flag (Z): If a result of an instruction has a value zero, this flag is set. Otherwise it is reset Sign flag (S): If the MSB of the result of an operation has a value 1, this flag is set, otherwise reset Carry flag (C): After an operation if a carry is generated a carry flag will set Auxiliary carry flag (AC): During the arithmetic operation, if a carry is moved from the 3rd bit to 4th bit, then this flag is set, otherwise it is reset Parity flag (P): The flag is set only when the result of operation contains even number of ones. If it is odd, the flag will reset. 	6	6

Handwritten signature and notes:
 12/05/23
 Lect in EEE
 CPE Panchur.