KHUSHAL SACHDEVA
88044

Date
DELTA Pg No.

Assignment - 2

(1) $(46.5)_{10} = 32 + 8 + 4 + 2 + 0.5$

2^4 2^3 2^2 2^1 2^0 2^{-1}

2^5 | 2^4 2^3 2^2 2^1 | 2^0 2^{-1}

1 0 1 1 1 0 . 1

$(46.5)_{10} = (101110.1)_2$

MANTISSA : 0 10111010 00000000 (16 bits)

↓

Sign bit

Exponent : 00000110 (+6) (8-bits)

(2) a) 0001 0000 0010 0100

ADD $(024)_{16}$ ADD 024

↓ ↓

1 024 $\Rightarrow (1024)_{16}$

This is direct memory reference instruction which will perform ADD operation

ADD content of M[024] to AC $\Rightarrow$ AC + M[024]

b) 1011 0001 0010 0100 $= (B124)_{16}$

↓ ↓ ↓

I STA $(124)_{16}$ $\Rightarrow$ STA I 124

This is indirect memory reference instruction which will perform STA i.e. Store AC in M[124]

c) 0111 0000 0010 0000 $= (7020)_{16}$

↓ ↓

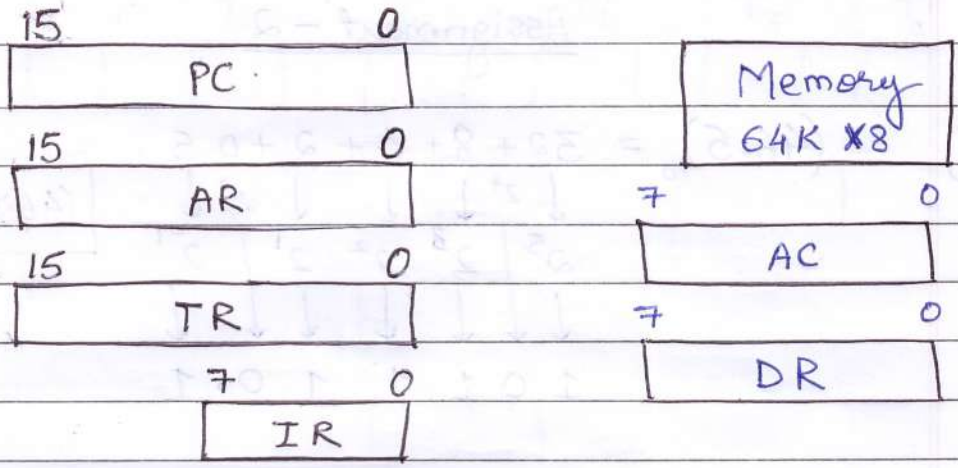
7 020

↓ ↓

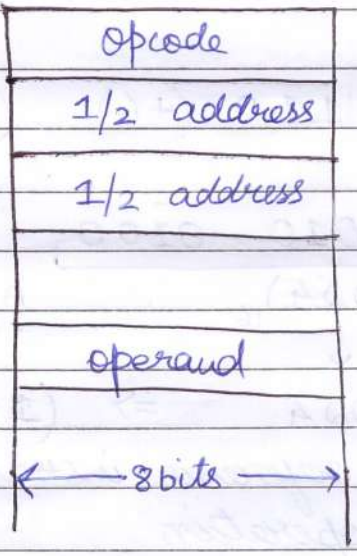
Register Increment AC (INC)

This is Register reference instruction which will increase AC (AC++)

(3) (a)



(b)



(c) T_0 : IR ← M[PC] , PC++

T_1 : AR(0-7) ← M[PC] , PC++

T_2 : AR(8-15) ← M[PC] , PC++

T_3 : DR ← M[AR]

(4) (a) Three address instructions : (Using General register computer)

MUL	R_1, E, F	$R_1 \leftarrow E * F$
SUB	R_1, D, R_1	$R_1 \leftarrow M[D] - R_1$
MUL	R_1, C, R_1	$R_1 \leftarrow M[C] * R_1$
MUL	R_2, A, B	$R_2 \leftarrow M[A] * M[B]$
ADD	X, R_1, R_2	$M[X] \leftarrow R_1 + R_2$

(b) Using a general register computer with two register instructions

MOV	R_1, F	$R_1 \leftarrow M[F]$
MUL	R_1, E	$R_1 \leftarrow R_1 * M[F]$
MOV	R_2, D	$R_2 \leftarrow M[D]$
SUB	R_2, R_1	$R_2 \leftarrow R_2 - R_1$
MUL	R_2, C	$R_2 \leftarrow R_2 * M[C]$
MOV	R_1, B	$R_1 \leftarrow M[B]$
MUL	R_1, A	$R_1 \leftarrow R_1 * M[B]$
ADD	R_1, R_2	$R_1 \leftarrow R_1 + R_2$
MOV	X, R_1	$M[X] \leftarrow R_1$

c) Using an accumulator type compiler with one address instructions :

LOAD	F	$AC \leftarrow M[F]$
MUL	E	$AC \leftarrow AC * M[E]$
STORE	T	$M[T] \leftarrow AC$
LOAD	D	$AC \leftarrow M[D]$
SUB	T	$AC \leftarrow AC - M[T]$
MUL	C	$AC \leftarrow AC * M[C]$
STORE	T	$M[T] \leftarrow AC$
LOAD	B	$AC \leftarrow M[B]$
MUL	A	$AC \leftarrow AC * M[A]$
ADD	T	$AC \leftarrow AC + M[T]$
STORE	X	$M[X] \leftarrow AC$

(d) Using a stack organized comp. with zero-address operation instructions

PUSH	E	TOS ← E
PUSH	F	TOS ← F
MUL		TOS ← E*F
PUSH	D	TOS ← D
SUB		TOS ← D − E*F
PUSH	C	TOS ← C
MUL		TOS ← C*(D − E*F)
PUSH	A	TOS ← A
PUSH	B	TOS ← B
MUL		TOS ← A*B
ADD		TOS ← A*B + C*(D − E*F)
POP	X	M[X] ← TOS

(5) a) 32 multiplexers, each of size 16 x 1
b) 4 inputs each, to select one of 16 registers
c) 4 to 16 line decoder
d) 32 + 32 + 1 = 65 data input lines
32 + 1 = 33 data output lines

(6) The address part of the indexed mode instruction must be set to zero.

(7) Z = effective address
a) Direct : Z = Y
b) Indirect : Z = M[Y]
c) Relative : Z = Y + W + 2
d) Indexed : Z = Y + X

Date
DELTA Pg No.

(8) a) There are $8 \times 8 = 64$ AND gates in each segment and an 8-bit adder

b) There are 7 segments in the pipeline

$K = 7$

c) Average time $= \frac{(K+n-1)t_p}{n} = \frac{(n+6)30}{n}$

for $n = 10$ $\quad t_{av} = \frac{16}{10} \times 30 = 48$ ns

for $n = 100$ $\quad t_{av} = \frac{106}{100} \times 30 = 31.8$ ns

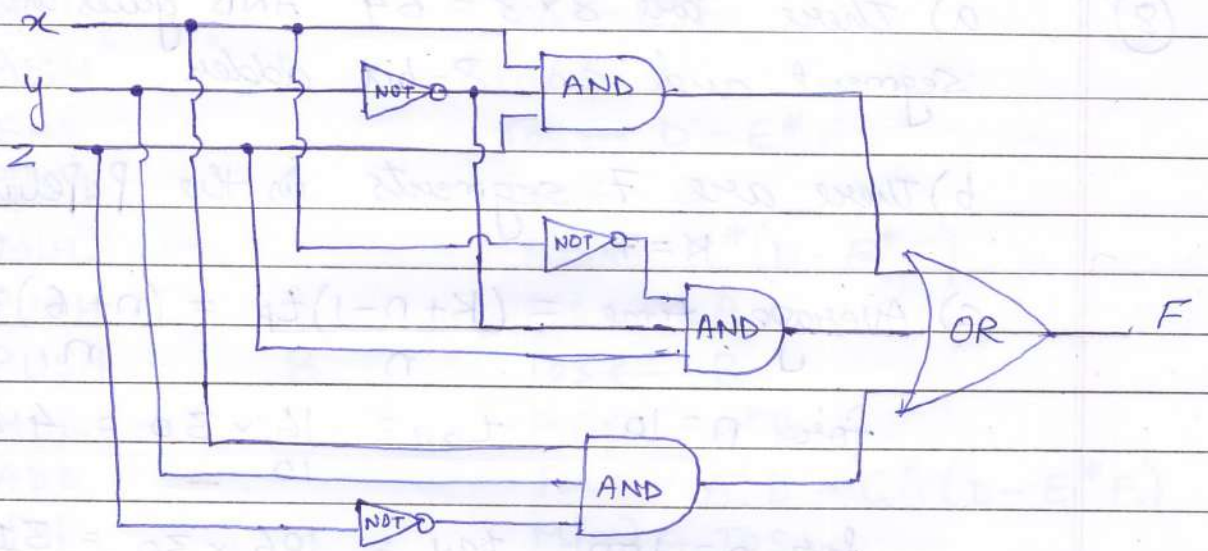
for $n \to \infty$ $\quad t_{av} = \frac{\infty}{\infty} \times 30 = 30$ ns

To increase the speed of multiplication, a carry-save adder is used to reduce the propagation time of the carries.

(9) a) Truth table $\quad F = xy'z + x'y'z + xyz$

x	y	z	F =
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

b)



c) $F = xy'z + x'y'z + xyz$

$= z[xy' + x'y' + xy]$

$= z[x(y+y') + x'y']$

$= z[x + x'y']$

$= z[x+y']$

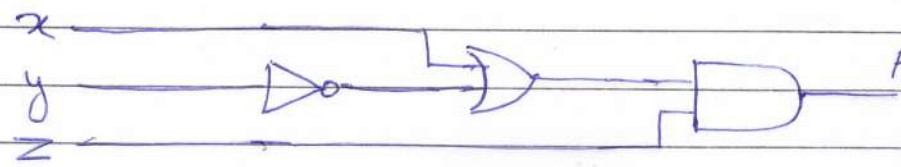
$[A + A'B = A + B]$

$\boxed{F = (x+y')z}$

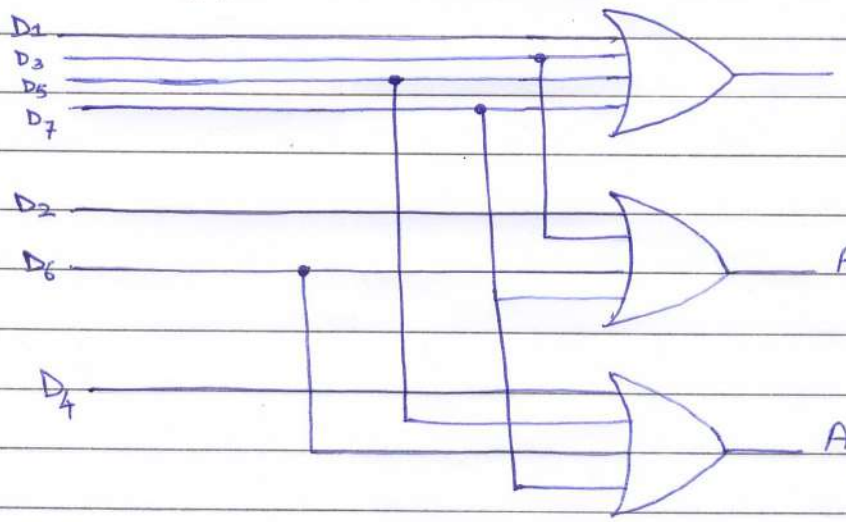
d)

x	y	z	$F = (x+y')z$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

e)



(10)



if all inputs equal to 0 or if only $D_0 = 1$: the outputs $A_2 A_1 A_0 = 000$

needs one more output to recognize that all zeroes input condition