Instruction format

* computer perform task on basis of instruction Provided.

* Instruction in comp comprises of groups called fields.

* common fields:-

1) operation field
   - Specifies the operation to be performed like addition

2) Address field
   - which contain the location of oprand, that is register or memory location

3) Mode field
   - Specifies how oprand is to be founded.

* Instruction is of various length depending upon the no of addresses it contain.

* CPU organisation are of 3 types on the basis of no. of address field.

1) single accumulator org → involves acc
2) General register org → only registers for computation.
3) Stack org.

Scanned by TapScanner

* on the basis of no of address, instruction are classified as

i) Zero address (ADD (SP)+, (SP))

ii) one address ADD R_1 $AC \leftarrow AC + R_1$

iii) Two address ADD R_1, R_2 $R_1 \leftarrow R_1 + R_2$

iv) Three address ADD R_1, R_2, R_3 or ADD A, B, C

$M[A] = M[B] + M[C]$

Zero address

1) $$X = \frac{(A+B*C)}{(D-E*F+G*H)}$$

TOS → Top of stack

converting x to postfix expression:-

A B C * + D E F * - G H * + /

Zero address code

	cmnts
PUSH A	TOS ← A
PUSH B	TOS ← B
PUSH C	TOS ← C
~~Push~~ MUL	TOS ← B * C
ADD	TOS ← A + B * C
PUSH D	TOS ← D
PUSH E	TOS ← E
PUSH F	TOS ← F
MUL	TOS ← E * F

Scanned by TapScanner

SUB	TOS ← D-E*F
PUSH G	TOS ← G
PUSH H	TOS ← H
MUL	TOS ← G*H
ADD	TOS ← D-E*F + G*H
DIV	TOS ← A+B*C/D-E*F+G*H
POP X	

2) $X = (A+B) * (T+Q)$

Postfix

A B + T Q + *

Zero add code

PUSH A	TOS ← A
PUSH B	TOS ← B
ADD	TOS ← A+B
PUSH T	TOS ← T
PUSH Q	TOS ← Q
ADD	TOS ← T+Q
MUL	TOS ← (A+B)*(T+Q)
POP X	

Scanned by TapScanner

3) $X = \frac{A - B + C * (D * E - F)}{G + H * I}$

Post fix

A B - C D E * F - * + G H I * + * /

PUSH A	TOS ← A
PUSH B	TOS ← B
SUB	TOS ← A - B
PUSH C	TOS ← C
PUSH D	TOS ← D
PUSH E	TOS ← E
MUL	TOS ← D * E
PUSH F	TOS ← F
SUB	TOS ← D * E - F
MUL	TOS ← C * (D * E - F)
ADD	TOS ← A - B + C * (D * E - F)
PUSH G	TOS ← G
PUSH H	TOS ← H
PUSH I	TOS ← I
MUL	TOS ← H * I
ADD	TOS ← G + H * I
~~ADD~~	~~TOS ← A - B +~~
DIV	TOS ← A - B + C * (D * E - F) / G + H * I
POP X	

Scanned by TapScanner

One address

$$X = \frac{A + B * C}{D - E * F + G * H}$$

LOAD E	AC ← M[E]
MUL F	AC ← AC * M[F]
STORE T	M[T] ← E * F
LOAD D	AC ← M[D]
SUB T	AC ← D − E * F
STORE T	M[T] ← AC
LOAD G	AC ← M[G]
MUL H	AC ← AC * M[H]
ADD T	AC ← AC + M[T]
STORE T	M[T] ← D − E * F + G * H
LOAD B	AC ← M[B]
MUL C	AC ← AC * M[C]
ADD A	AC ← AC + M[A]
DIV T	AC ← AC / M[T]
STORE X	M[X] ← AC

Scanned by TapScanner

Two address

$$X = \frac{A+B*C}{D-E*F+G*H}$$

MOV R1, B	R1 ← M[B]
MUL R1, C	R1 ← ~~M[B]~~ R1 * M[C]
ADD R1, A	R1 ← R1 + M[A]
MOV R2, D	R2 ← M[D]
MOV R3, E	R3 ← M[E]
MUL R3, F	R3 ← R3 * M[F]
SUB R2, R3	R2 ← R2 - R3
MOV R3, G	R3 ← M[G]
MUL R3, H	R3 ← ~~R3~~ R3 * M[H]
ADD R2, R3	R2 ← R2 + R3
DIV R1, R2	R1 ← R1 / R2
MOV X, R1	M[X] ← R1

Scanned by TapScanner

Three address

$$X = \frac{A+B*C}{(D-E*F+G1*H)}$$

MUL R_1, B, C	$R_1 \leftarrow M[B] * M[C]$
ADD R_1, R_1, A	$R_1 \leftarrow R_1 + M[A]$
MUL R_2, E, F	$R_2 \leftarrow M[E] * M[F]$
SUB R_2, D, R_2	$R_2 \leftarrow M[D] - R_2$
MUL R_3, G, H	$R_3 \leftarrow M[G] * M[H]$
ADD R_2, R_2, R_3	$R_2 \leftarrow R_2 + R_3$
DIV X, R_1, R_2	$X \leftarrow R_1 / R_2$

Scanned by TapScanner