RAMA UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF ENGINEERING & TECHNOLOGY

CSPS-106 Computer Organization

Lecture-14

Mr. Dilip Kumar J Saini
Assistant Professor
Computer Science & Engineering

OUTLINE

- **REVERSE POLISH NOTATION**
- **INSTRUCTION FORMAT**
- **THREE, and TWO-ADDRESS INSTRUCTIONS**
- **ONE, and ZERO-ADDRESS INSTRUCTIONS**

REVERSE POLISH NOTATION

Arithmetic Expressions: A + B

A + B   Infix notation

+ A B   Prefix or Polish notation

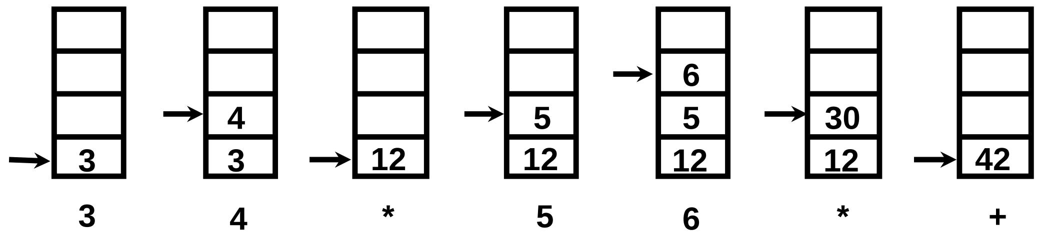
A B +   Postfix or reverse Polish notation

- The reverse Polish notation is very suitable for stack manipulation

Evaluation of Arithmetic Expressions

Any arithmetic expression can be expressed in parenthesis-free Polish notation, including reverse Polish notation

(3 * 4) + (5 * 6) ⇒ 3 4 * 5 6 * +



INSTRUCTION FORMAT

Instruction Fields

OP-code field - specifies the operation to be performed
Address field - designates memory address(s) or a processor register(s)
Mode field - specifies the way the operand or the effective address is determined

The number of address fields in the instruction format depends on the internal organization of CPU

- The three most common CPU organizations:

Single accumulator organization:

```
ADD   X            /* AC ← AC + M[X] */
```

General register organization:

```
ADD   R1, R2, R3   /* R1 ← R2 + R3 */
ADD   R1, R2       /* R1 ← R1 + R2 */
MOV   R1, R2       /* R1 ← R2 */
ADD   R1, X        /* R1 ← R1 + M[X] */
```

Stack organization:

```
PUSH  X            /* TOS ← M[X] */
ADD
```

THREE, and TWO-ADDRESS INSTRUCTIONS

Three-Address Instructions:

Program to evaluate X = (A + B) * (C + D) :

```
ADD    R1, A, B        /*  R1 ← M[A] + M[B]     */
ADD    R2, C, D        /*  R2 ← M[C] + M[D]     */
MUL    X, R1, R2       /*  M[X] ← R1 * R2       */
```

- Results in short programs
- Instruction becomes long (many bits)

Two-Address Instructions:

Program to evaluate X = (A + B) * (C + D) :

```
MOV    R1, A       /* R1 ← M[A]         */
ADD    R1, B       /* R1 ← R1 + M[B]    */
MOV    R2, C       /* R2 ← M[C]         */
ADD    R2, D       /* R2 ← R2 + M[D]    */
MUL    R1, R2      /* R1 ← R1 * R2      */
MOV    X, R1       /* M[X] ← R1         */
```

ONE, and ZERO-ADDRESS INSTRUCTIONS

One-Address Instructions:

- Use an implied AC register for all data manipulation
- Program to evaluate X = (A + B) * (C + D) :

```
LOAD    A    /*  AC ← M[A]          */
ADD     B    /*  AC ← AC + M[B]     */
STORE   T    /*  M[T] ← AC          */
LOAD    C    /*  AC ← M[C]          */
ADD     D    /*  AC ← AC + M[D]     */
MUL     T    /*  AC ← AC * M[T]     */
STORE   X    /*  M[X] ← AC          */
```

Zero-Address Instructions:

- Can be found in a stack-organized computer
- Program to evaluate X = (A + B) * (C + D) :

```
PUSH    A    /*  TOS ← A                    */
PUSH    B    /*  TOS ← B                    */
ADD          /*  TOS ← (A + B)              */
PUSH    C    /*  TOS ← C                    */
PUSH    D    /*  TOS ← D                    */
ADD          /*  TOS ← (C + D)              */
MUL          /*  TOS ← (C + D) * (A + B)    */
POP     X    /*  M[X] ← TOS                 */
```

MUTIPLE CHOICE QUESTIONS:

Sr no	Question	Option A	Option B	OptionC	OptionD
1	generation computers use assembly language:	first generation	second generation	third generation	fourth generation
2	Assembly language program iscalled:	Object program	Source program	Oriented program	All of these
3	By whom address of external function in the assembly source file supplied by when activated:	Assembler	Linker	Machine	Code
4	An -o option is usedfor:	Inputfile	Externalfile	Outputfile	None ofthese
5	The assemblertranslates ismorphically mapping from mnemonic in these statements to machineinstructions:	1:1	2:1	3:3	4:1

REFERENCES

- http://www.engppt.com/search/label/Computer%20Organization%20and%20Architecture
- http://www.engppt.com/search/label/Computer%20Architecture%20ppt