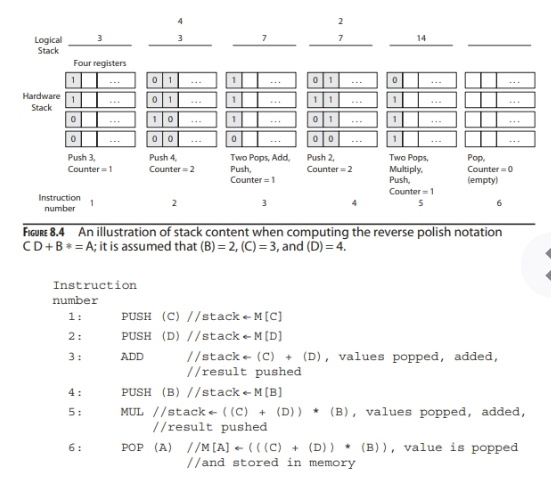
Stack ISA - Example of assembly program: A = B \* (C+D)



Acc ISA - Example of assembly program: A = B \* (C+D)

1. LD (C) // ACC ← M[C]
2. ADD (D) // ACC ← ACC + M[D]
3. MUL (B) // ACC ← ACC \* M[B]
4. ST(A) // M[A] ← ACC

CISC-ISA: Example of assembly program: A = B \* (C+D)

B = 4; C = 5; D = 10

The value in R1 after execution of instruction No. 1 is 5.

The value of R1 after execution of instruction No. 3 is 60

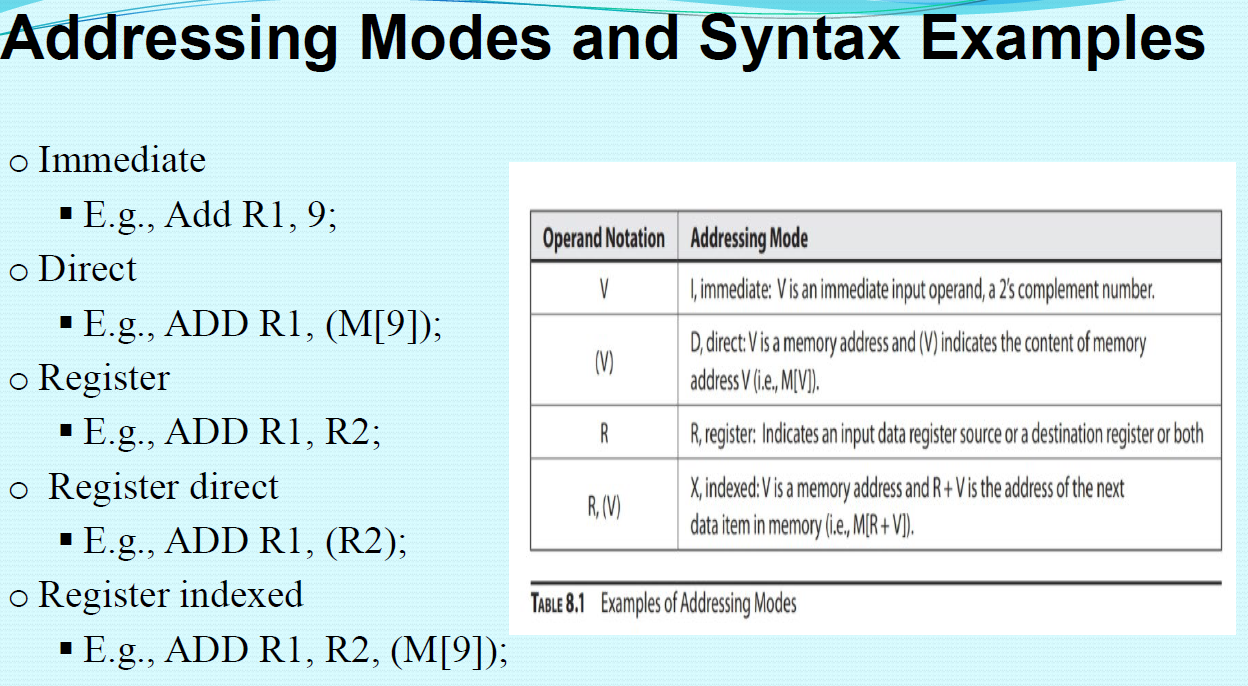
1. LD R1, (C) // R1 ←M[C]
2. ADD R1, (D) // R1 ← R1 + M[D]
3. MUL R1, (B) // R1 ← R1 \* M[B]
4. ST (A), R1 // M[A] ← R1

RISC-ISA: Example of assembly program: A = B \* (C + D)

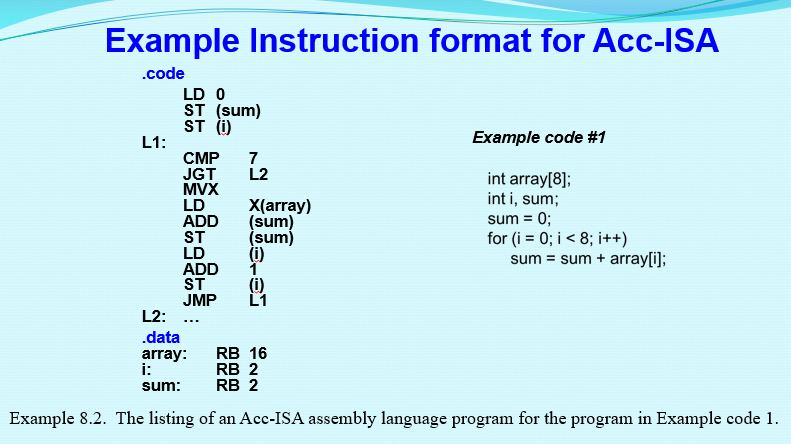
1. LD R1, (C ) // R1 ← M[C]
2. LD R2, (D ) // R2 ← M[D]
3. Add R3, R1, R2 // R3 ← R1 + R2
4. LD R4, (B) // R4 ← M[B]
5. MUL R5, R3, R4 // R5 ← R3 \* R4
6. ST (A), R5 // M[A] ← R5

Computation is performed by a RISC ISA. A = B \* (C + D). What is the value in R5 after the execution of code line # 6: (B = 5; C = 10; D = 15) ie: Code line # 6 has been completed. (20 pts)

R5 ?

* + 1. 
    2. Immediate E.g., Add R1, 9;
    3. //
    4. Direct E.g., ADD R1, (M[9]);
    5. Register E.g., ADD R1, R2;
    6. Register direct E.g., ADD R1, (R2);
    7. Register indexed E.g., ADD R1, R2, (M[9]);
    8. oRegister indexed E.g., ADD R1, R2, (M[9]);
    - What is the value in R1 ?
    1. RTN#1: R1 ← R1 +M[R2 + 9]

- Example 8.2 Assembly code listing of an Acc-ISA assembly language program for the high level (c) program in Example code 1



.code L1: The listing of an Acc-ISA assembly language program for the program in

.code //start program code

LD 0

ST

ST

L1:

CMP 7

JGT L2

MVX

LD X(array)

ADD (sum)

ST (sum) )

LD (i)

ADD 1

ST (i)

JMP L1 //loop back ( End of for loop)

Program level Translation:

Now, here is an example of a real C If-Then-Else:

**if**(x == 10)

{

x = 0;

}

**else**

{

x++;

}

Which gets translated into the following assembly/machine code:

X = 5; 0x 05 = 5 in decimal;

X = 0xA which is equal decimal 10.

Mov eax, $x

Cmp eax, 0x0A ; 0x0A = 10

Jne else

Mov eax, 0

Jmp end

Else:

Inc eax

End;

Mov $x, eax