

int a = 3 + 7 * 4; ←

COMPILATORE C PRIMITIVO

int var = cost;

int var1 = var2 op var3;
int var1 = var2 op cost;

int a = 3 + 7 * 4; ⇒
↓
int a = 7;
a = a * 4;
a = a + 3;

int a = 3;
int b = 7;
b = b * 4;
a = a + b;

ASSEMBLY PRIMITIVO

$R_0 \dots R_n = \text{REGISTER A K BIT}$

$R_i = \text{cost}$

$R_i = R_j \text{ op cost}$

$R_i = R_j \text{ op } R_k$

MOV R_i, cost

OP R_i, R_j, cost

OP R_i, R_j, R_k

op = +, -, *, /

ADD, SUB, MUL, DIV

int a = 3 + 7 * 4; =>

↓

int a = 7;

a = a * 4;

a = a + 3;

a → R0

MOV R0, 7

MUL R0, R0, 4

ADD R0, R0, 3

R0 = 0

ADD R0, R0, 7

MUL R0, R0, 4

ADD R0, R0, 3

int a = 3;

int b = 7;

b = b * 4;

a = a + b;

a → R0

b → R1

MOV R0, 3

MOV R1, 7

MUL R1, R1, 4

ADD R0, R0, R1

R0, R1 = 0

ADD R0, R0, 3

ADD R1, R1, 7

MUL R1, R1, 4

ADD R0, R0, R1

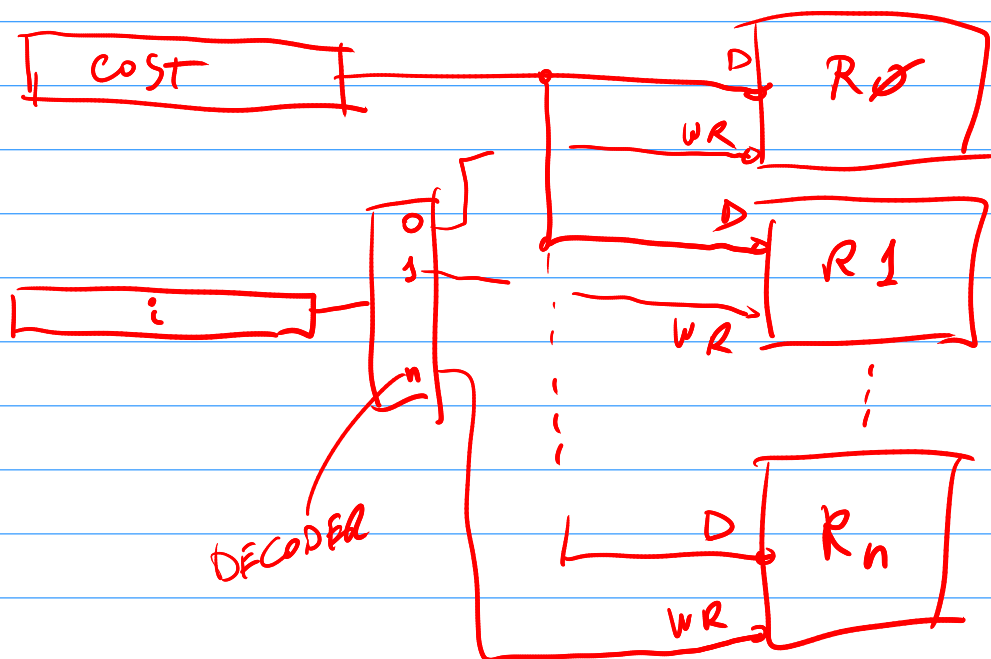
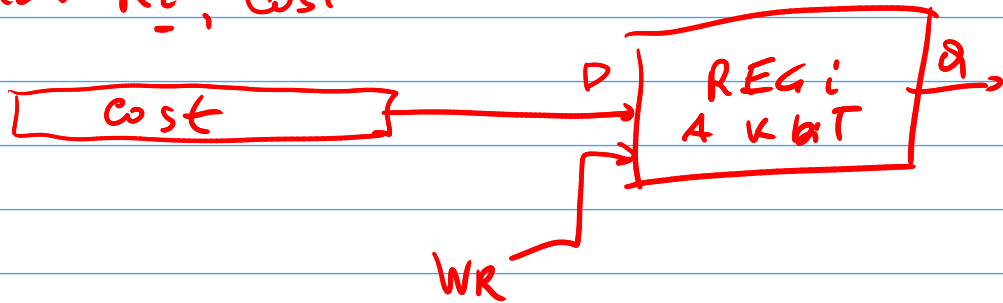
R0 = R3

MOV R0, R3

ASSEGNAZIONE

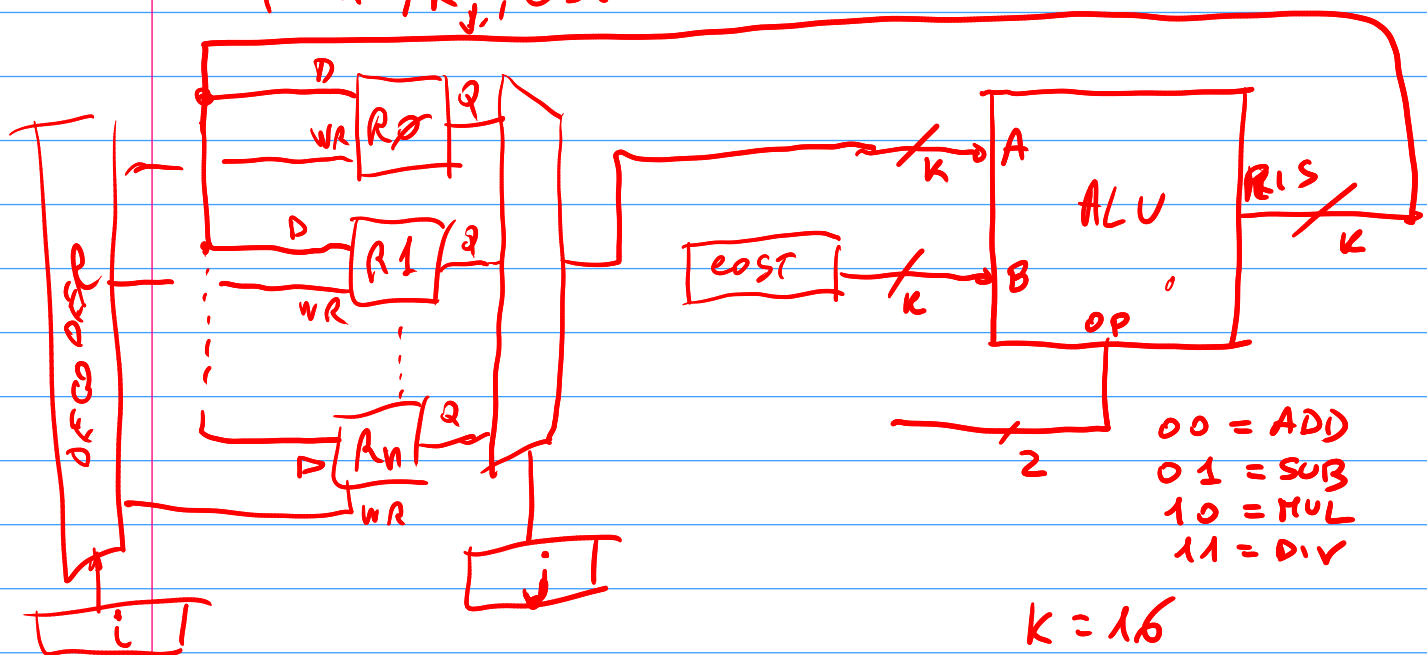
MOV R_i , cost

$R_i = \text{cost}$



OPERAZIONE

op $R_i, R_j, cost \leftarrow R_i = R_j$ op cost



$R_0 \dots R_7$

op R_i, R_j, R_k

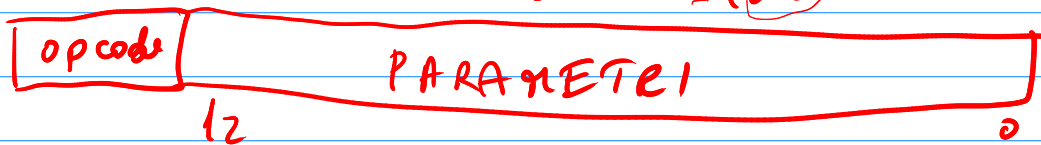
$R_i = R_j$ op R_k

MOV, ADD, SUB, MUL, DIV || 5 istruzioni

3 bit, opcode

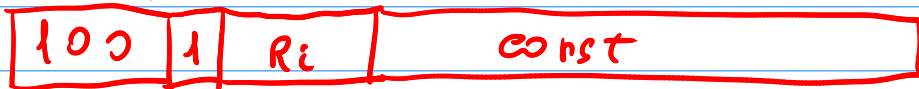
↓ 1
ADD 000
SUB 001
MUL 010
DIV 011
MOV 100

15 14 13



MOV R_i , const

15 13 12 11 10 9 8 0



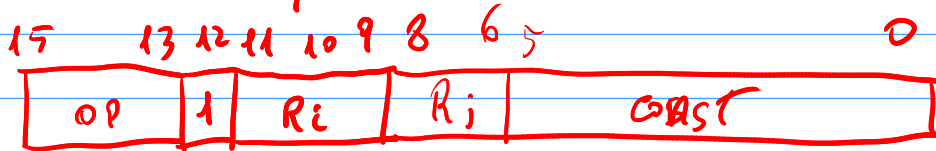
MOV R_i , R_k

15 13 12 11 10 9 8 6 0



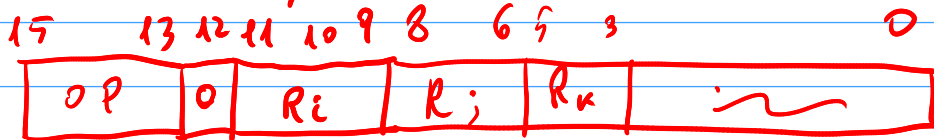
op = ADD, SUB, MUL, DIV

op R_i, R_j, const



op R_i, R_j, R_k

$R_i = R_j$ op R_k



	BIN	HEX
MOV R0, 3	1001 0000 0000 0011	9003
MOV R1, 7	1001 0010 0000 0111	9207
MUL R1, R1, 4	0101 0010 0100 0100	5244
ADD R0, R0, R1	0000 0000 0000 1000	0008