# LAP = Liotro Advanced Processor

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

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

## ASSEMBLY PRIMITIVO

R0 .... Rn = REGISTRI A k BIT

```
Ri = cost                MOV  Ri, cost
Ri = Rj op cost          op   Ri, Rj, cost
Ri = Rj op Rk            op   Ri, Rj, Rk
```

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

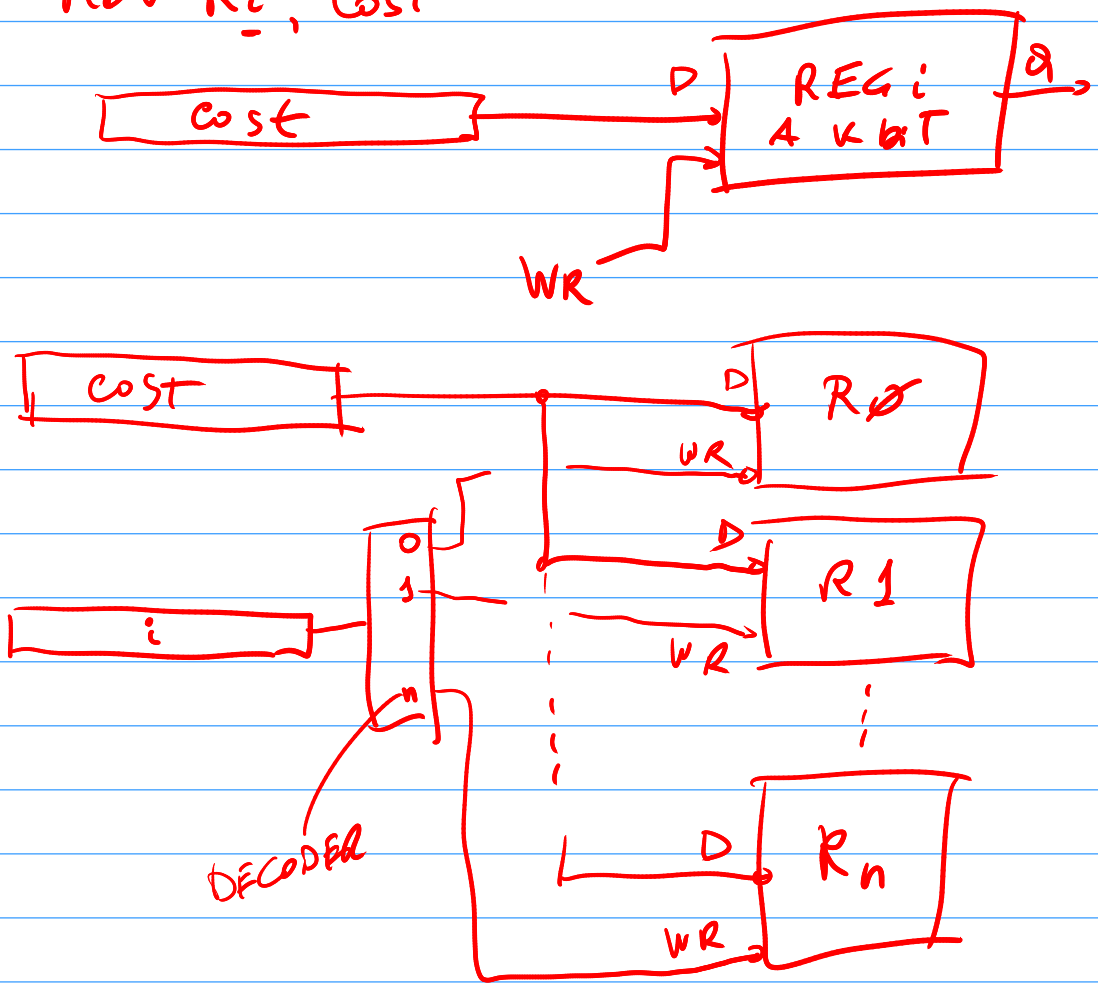
R0 = R3

MOV R0, R3

# ASSEGNAZIONE

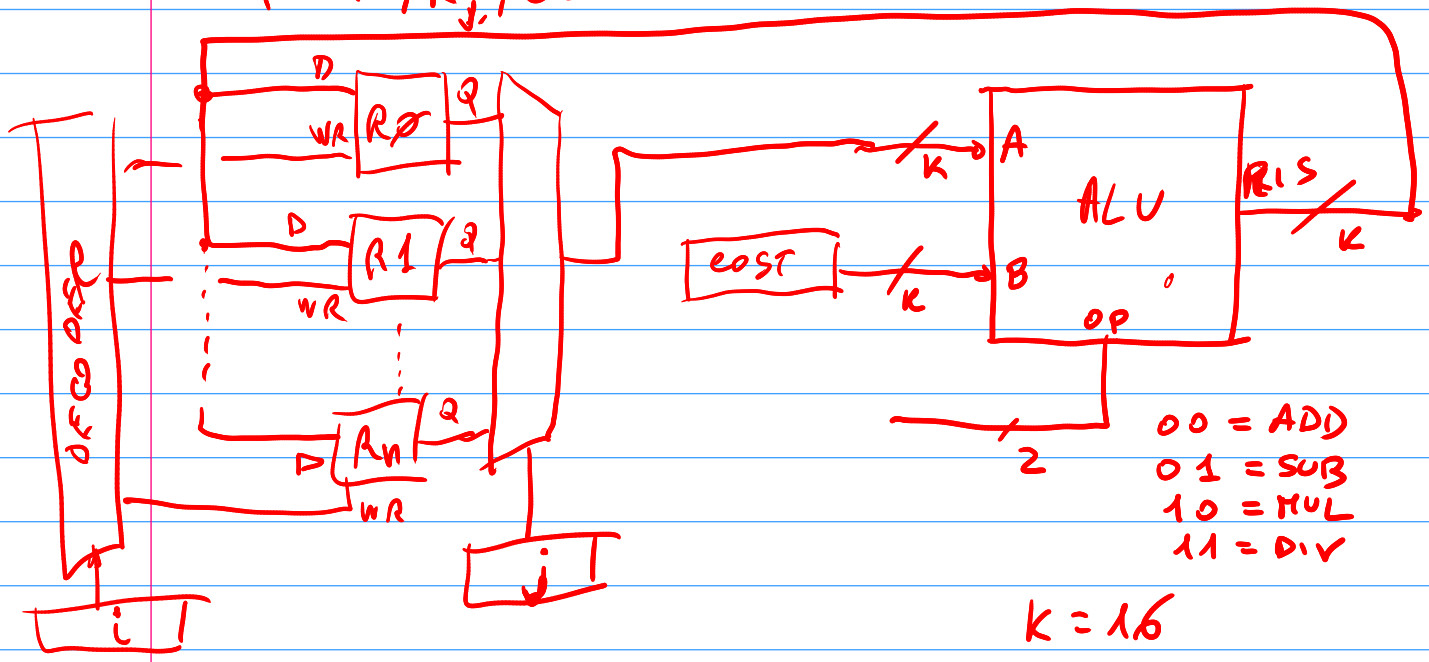
$R_i = cost$

MOV $R_i$, cost



# OPERAZIONE

op $R_i, R_j$, cost ← $R_i = R_j$ op cost




K = 16

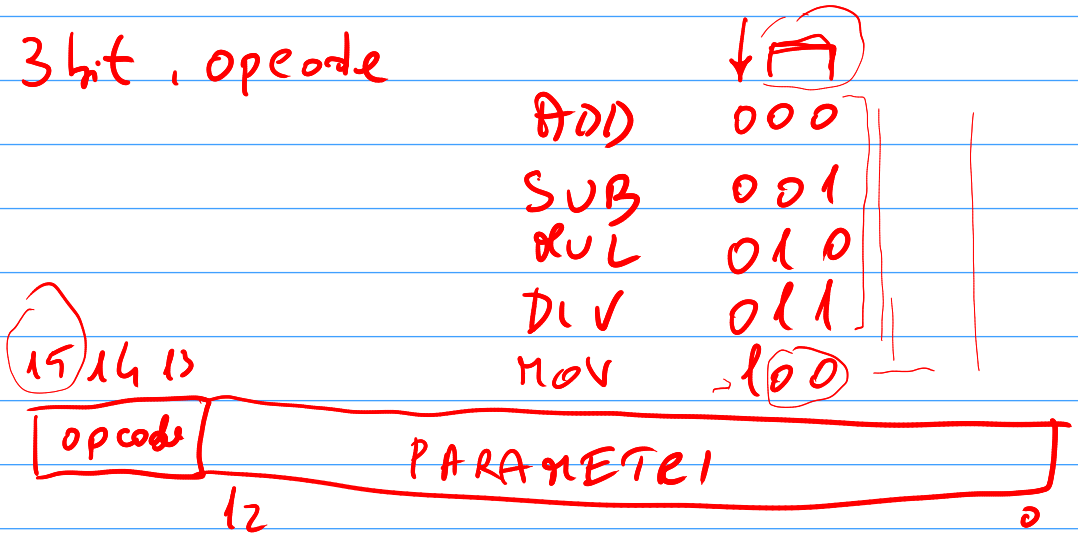
R0 ... R7

op $R_i, R_j, R_K$

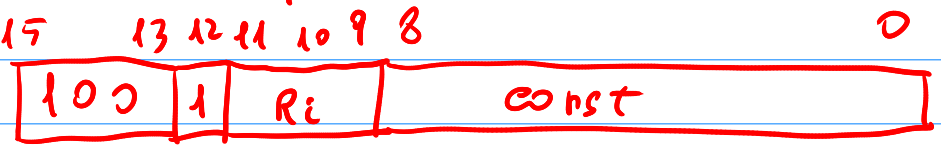
$R_i = R_j$ op $R_K$

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

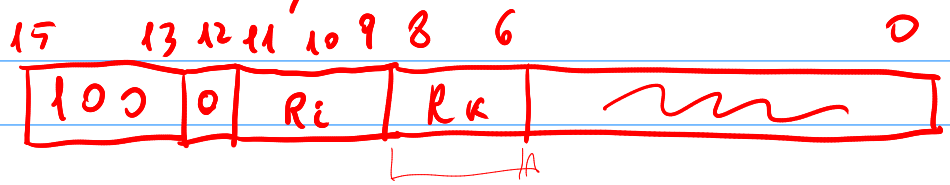
3 bit, opcode




MOV Ri, const

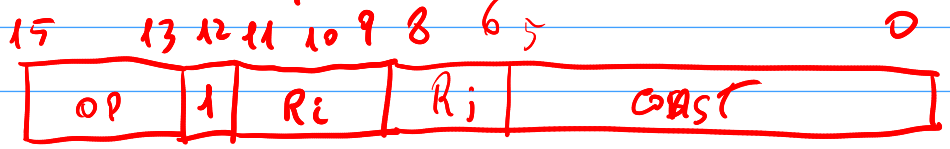



MOV Ri, Rk

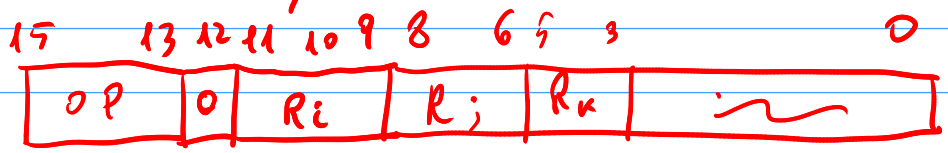


op = ADD, SUB, MUL, DIV

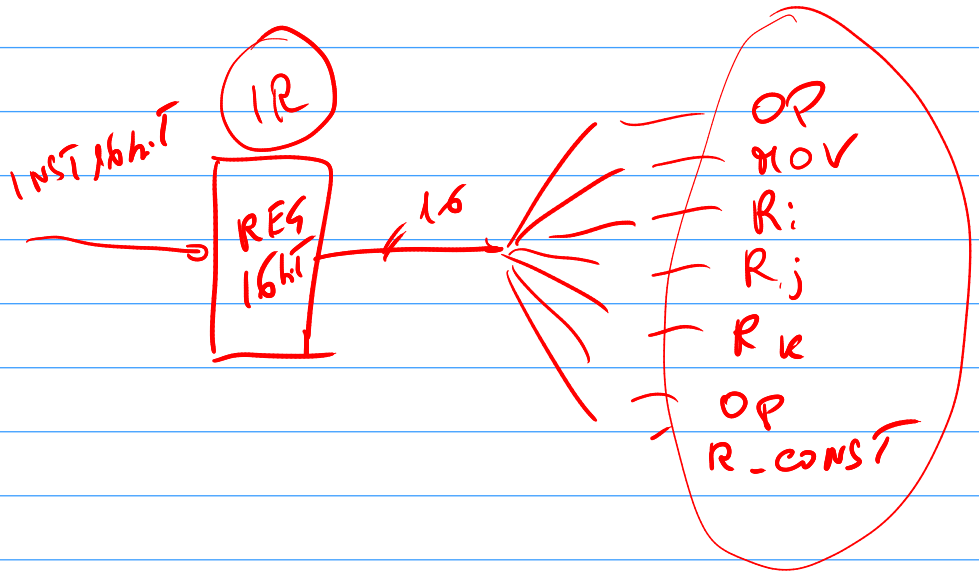
OP $R_i, R_j$, const




OP $R_i, R_j, R_k$ $R_i := R_j$ op $R_k$

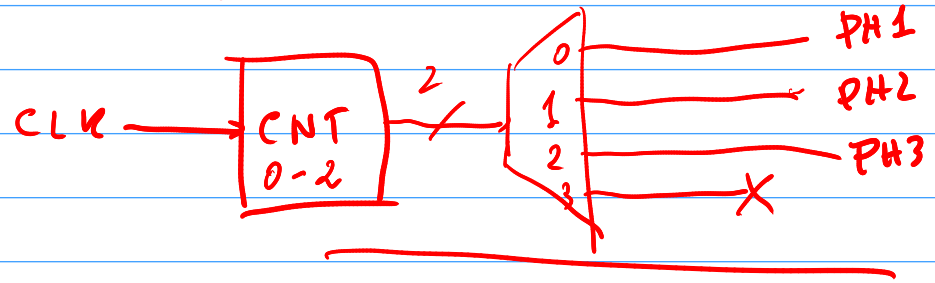


| | BIN | HEX |
|---|---|---|
| MOV R0, 3 | 1001 0000 0000 0011 , | 9003 |
| MOV R1, 7 | 1001 0010 0000 0111 , | 9207 |
| MUL R1, R1, 4 | 0101 0010 0100 0100 , | 5244 |
| ADD R0, R0, R1 | 0000 0000 0000 1000 , | 0008 |

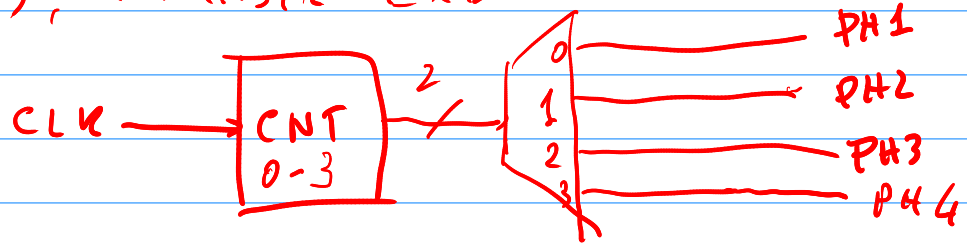


R7 → Program Counter

CNT=0 , 1. INSTR_REG ← [R7] ; INSTRUCTION FETCH
CNT=1 , 2. INSTR EXE
CNT=2 , 3. R7 ← R7 + 1 , ADD R7,R7,#1

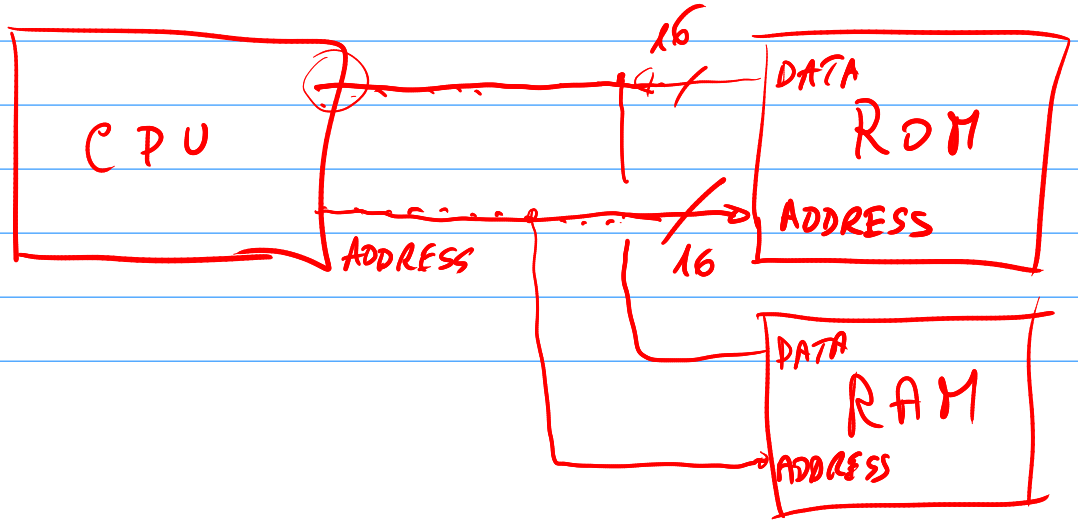


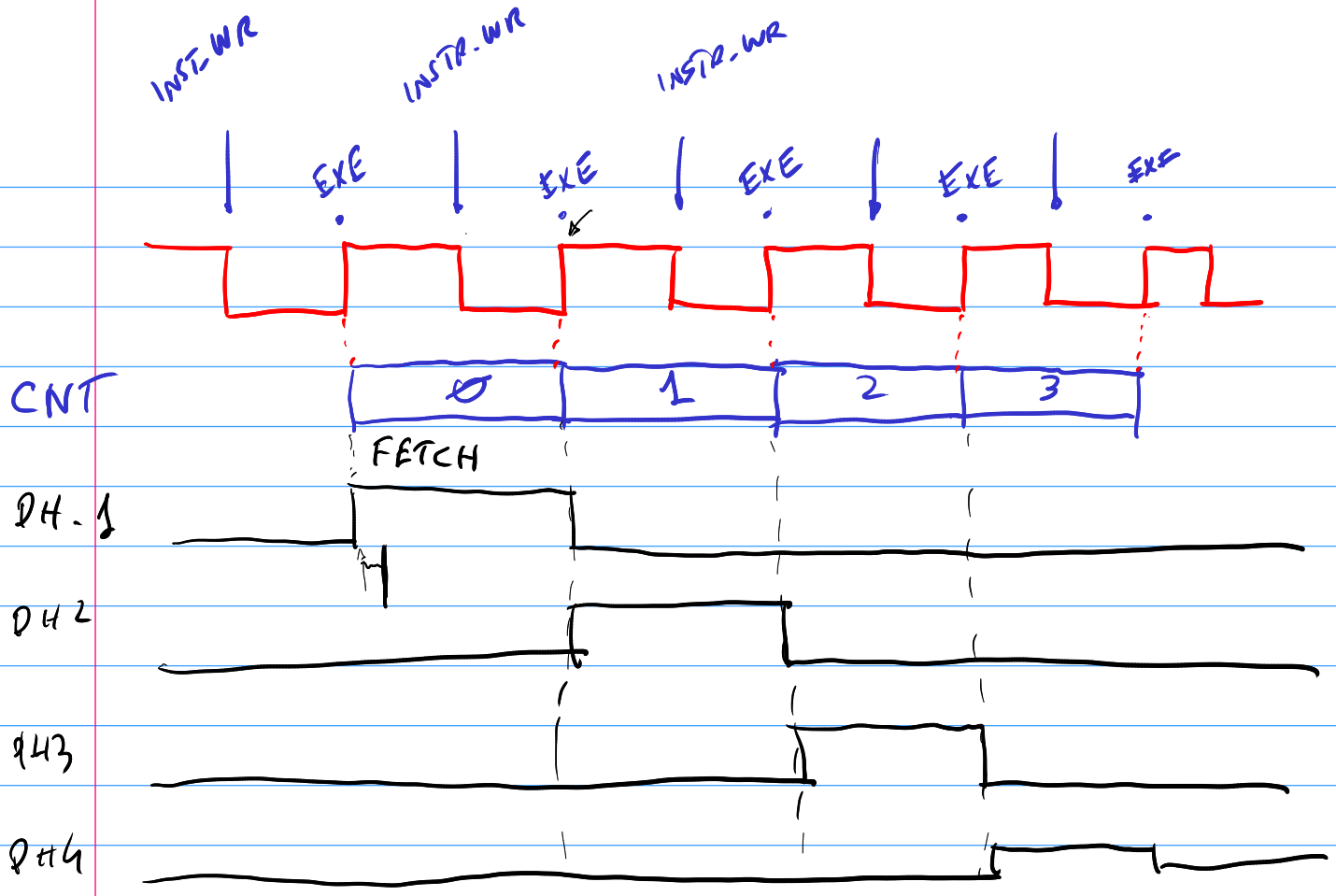

CNT=0 , 1. INSTR_REG ← [R7] ; INSTRUCTION FETCH
CNT=1 , 2. INSTR EXE
CNT=2 , 3. INSTR_REG ← ADD R7,R7,#1
CNT=3 , 4. INSTR EXE




PHASE_1 → CNT=0

INSTR_REG ← [R7] ; INSTRUCTION FETCH




INSTR.WR
INSTR.WR
INSTR.WR
EXE
EXE
EXE
EXE
EXE
CNT
0
1
2
3
FETCH
QH.1
QH2
QH3
QH4