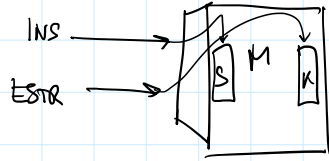
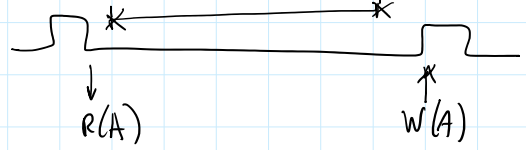


A

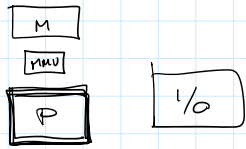
$A \rightarrow B, B \rightarrow A$



$\emptyset. (R_1 R_2 R_3 \text{ OR } (NE) NE_0 \leq$
 $(1 \ 1) \quad \quad \quad 1.$

Gtp

$1. \left(\begin{matrix} \text{seq}_0 (IN_1 - IN_2) = 0 \\ = 1 \end{matrix} \right) \quad \begin{matrix} IN_1 \rightarrow n[INS] \\ IN_2 \rightarrow \dots \end{matrix}$

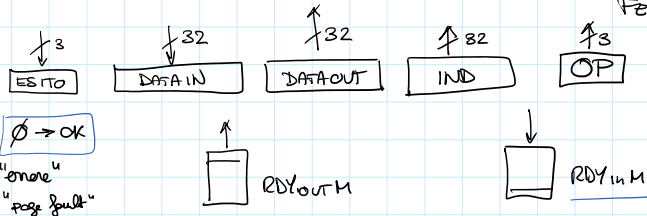


while (true) {
 prendo di un istruzione da M[ic]
 decodifico
 eseguo
 halt-interruzione. *interfaccia apposita*

interfaccia standard con M

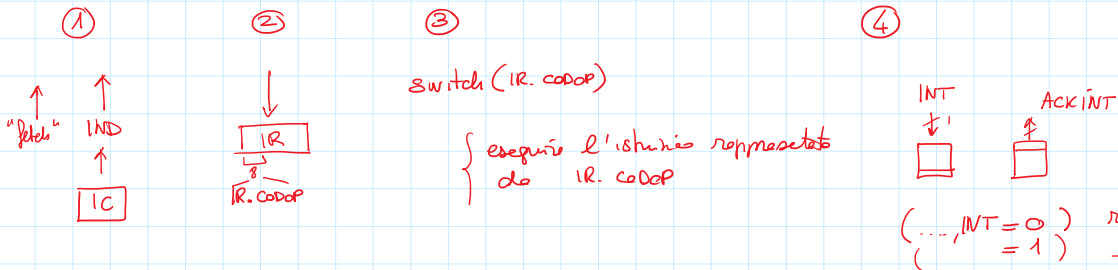
3 operazioni

LOAD } D-Risc
 STORE } dell'interprete FW
 FETCH



"read" → OP
 "load" → OP
 "write" → OP
 "store" → OP
 "fetch" → OP

STRUTTURA dell'interprete FW D-RISC



chp. ic → IND, "fetch" → OP, set RDYoutM, chl.

chl. (RDYinM, or(ESITO) = 0-) map, chl. ^{interp}
 (=10) DATAin → IR, reset RDYinM, ? →

codop			
-------	--	--	--

 (=11) map, halt-eccezioni,

interp. (IR.codop = "ADD") REG[IR.Ra] + REG[IR.Rb] → REG[IR.Rc],

8	6	6	6
codop	Ra	Rb	Rc

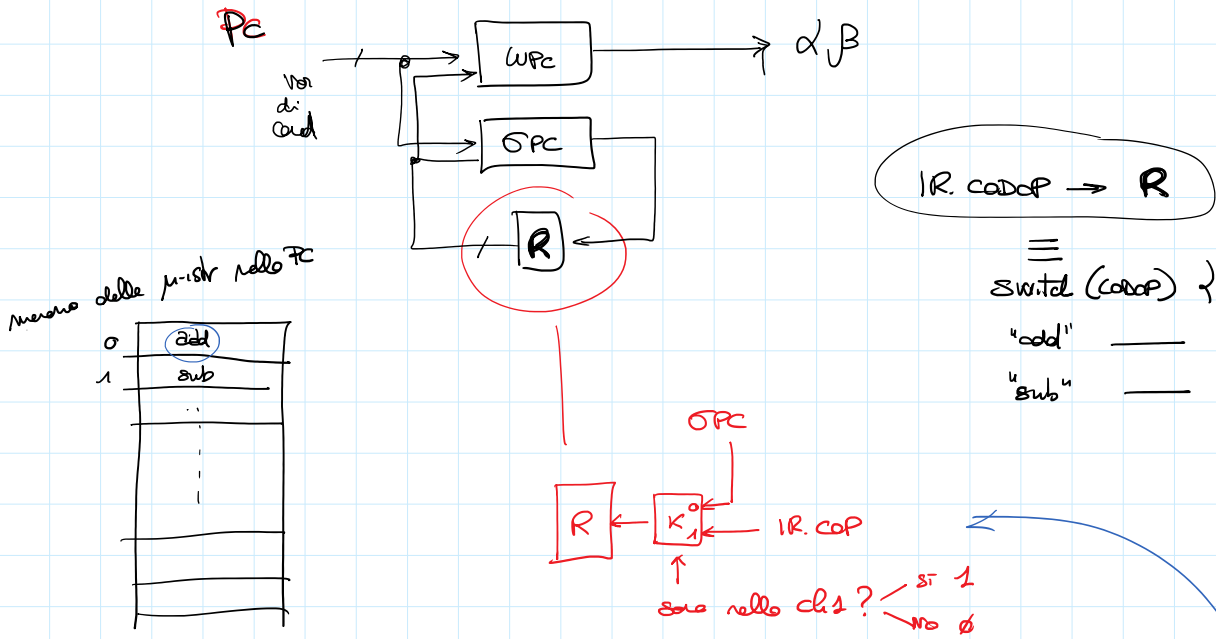
(IR.codop, INT = "add" ∅) REG[IR.Ra] + REG[IR.Rb] → REG[IR.Rc],
 IC+1 → IC, chl ∅
 (= "add" 1) REG[IR.Ra] + REG[IR.Rb] → REG[IR.Rc],
 IC+1 → IC, "halt-interruzione"

(IR.codop, INT = "sub" ∅) REG[] - REG[] → REG[]
 = "sub" 1

(= "load" ∅) ? → IND "load" → M } M-ist

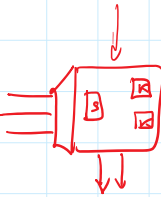
(= "load" -) REG[IR.Ra] + REG[IR.Rb] → IND, REG[] ← DATAin } M-ist
 "load" → OP, set RDYoutM, load

(= "goto" ∅) REG[IR.Ra] → ic, chl ∅
 (= "goto" 1) " " " " , halt-interruzione



ch ϕ . IC → IND, "faldi" → OP, set RDYoutM, ch1

ch1. (RDYinM, or(ESITO) = 0 →) nop, ch ϕ
 (= 11) reset RDYin, halt-ec
 (= 10) reset RDYin, DATAIN → IR, DATAIN.COP → R



add ϕ . (INT = ϕ) REG[IR.Ra] + REG[IR.Rb] → REG[IR.Rc], IC+1 → IC, ch ϕ
 (= 1) " " " " " " , halt-int

load ϕ . REG[IR.Ra] + REG[IR.Rb] → IND, "load" → OP, set RDYoutM, load1

load1. (RDYinM, or(ESITO), INT = 0 →) nop, load ϕ
 (= 11 -) reset RDYinM, halt-ec
 (= 1 - 1) reset RDYinM, DATAIN → REG[IR.Rc], IC+1 → IC, halt-intens.
 (= 100) " " " " " " ch ϕ

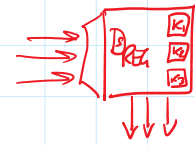
IF < ϕ . segue (REG[IR.Ra] - REG[IR.Rb]) → F, IF < 1

IF < 1. (F, INT) = ϕ IC+1 → IC, ch ϕ
 (= 01) IC+1 → IC, halt-int
 (= 10) IC + IR.offset → IC, ch ϕ
 (= 11) " " , halt-intens.

STORE ϕ . $REG[IR.R_a] + REG[IR.R_b] \rightarrow IND$, $REG[IR.R_c] \rightarrow DATAOUT$,
 "store" \rightarrow OP, set RDY_{outM} , store 1

store 1.

$(RDY_{inM}, OP(ESTD), INT \neq 0 \dots) NOP$
 $(= 100) IC+1 \rightarrow IC, cl \phi$



3 counters!

store ϕ
 store 1
 store

$REG[IR.R_a] + REG[IR.R_b] \rightarrow IND$, store 1

$REG[IR.R_c] \rightarrow DATAOUT$, "store" \rightarrow OP, set RDY_{outM} , store 2