

# Domanda 1

Note Title

22/06/2007

Assumiamo che l'interfaccia fra E ed W  
contenga i registri:

A (32 bit)

C (8 bit)

$RDY_{in}$  1 bit : transizione di livello

$ACK_{in}$  1 bit : transizione di livello

e che l'interfaccia con le unità  $W_i$   
contenga i registri:

$OUT_i$  (32 bit)

$ACK_i$  1 bit

$RDY_i$  1 bit

}  $\forall i \in [0, 7]$

Assumiamo inoltre di avere a disposizione

1) un OR dei bit  $ACK_0 \dots ACK_7$

(singolo polo OR, ritardo  $1t_p$ )

2) una rete combinatoria con ingressi

$ACK_0 \dots ACK_7$

e uscita

$w$  ( $\log_2(8)$  bit)

che indica il numero  $k$  di una unità  $W_k$  pronta ad accettare un valore da  $E$  (quindi tale  $x$  cui  $ACK_k = 1$ )

(Questo sarà una rete a due livelli di logica, quindi con ritardo pari a  $2t_p$ )

3) di una rete che, utilizzando 3 ALU intercannere ed alberi ed un Comparatore, calcola

$$CHECKSUM(A) == C$$

(Ritardo  $\frac{2 * 5t_p}{ALU} + \frac{2t_p}{COMPARATORE}$ )

Con queste assunzioni, il microprogramma dell'unità  $E$  può essere scritto come segue (minimizzare il numero di stati dello PC significa ridurre il numero delle misistazioni):

$\emptyset$ . ( $RDY_{in}$ , OR( $ACK_0 \dots ACK_7$ ),  $CHECKSUM(A,C)=0$ )  
 $map, \emptyset$  } ancora non ho ricevuto

(= 100)  $map, \emptyset$  } ricevuto, nessun  $w$ , pronto

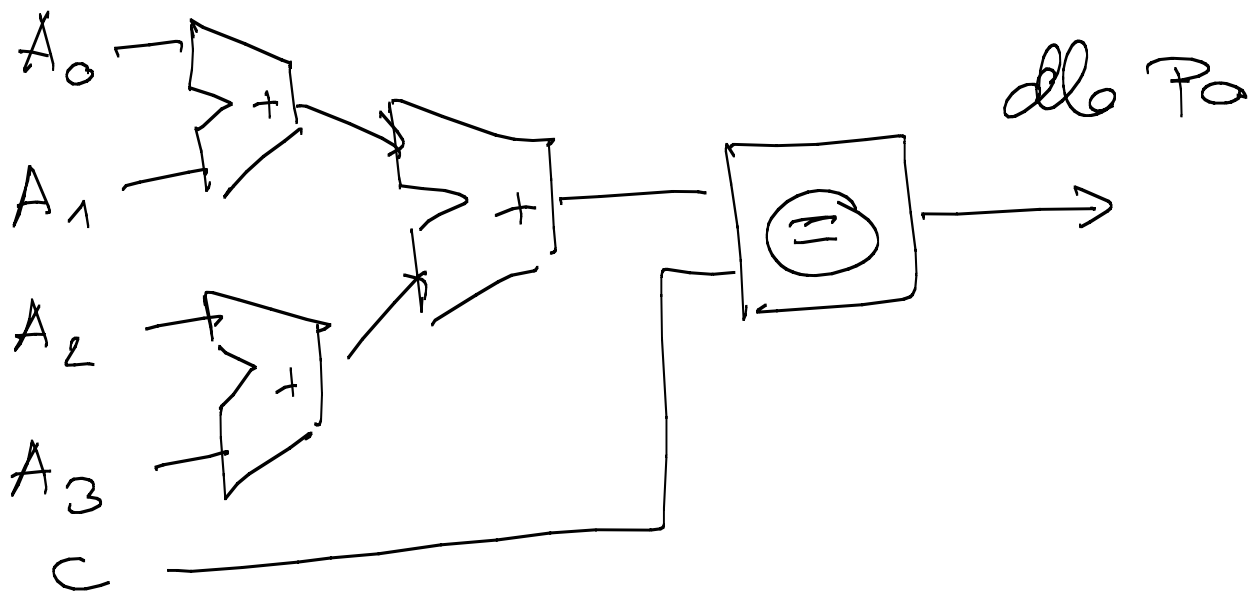
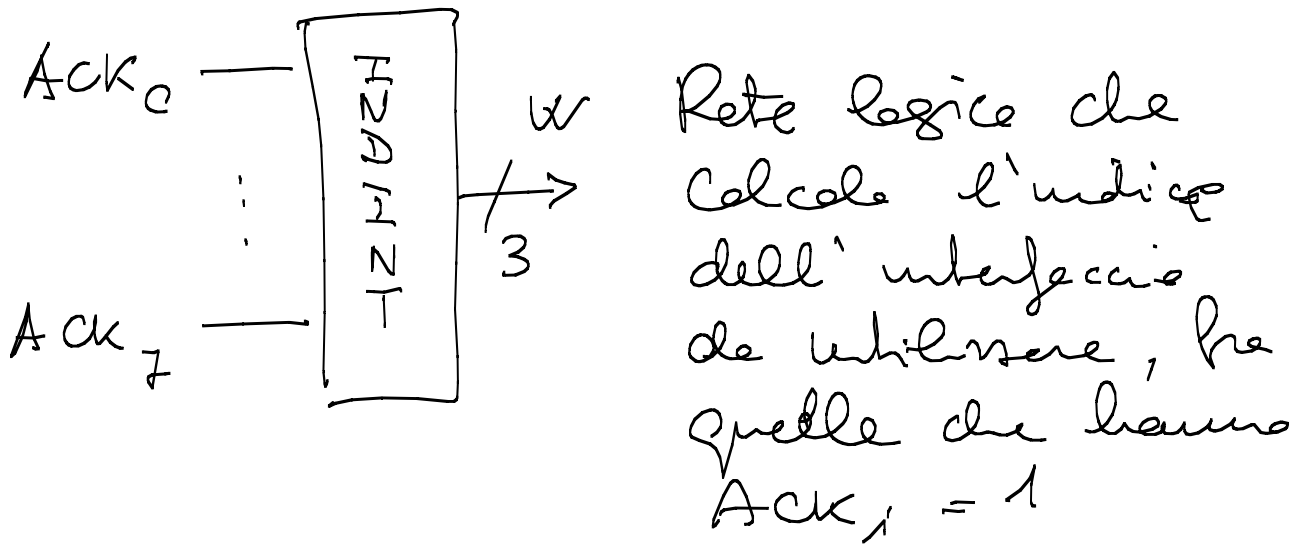
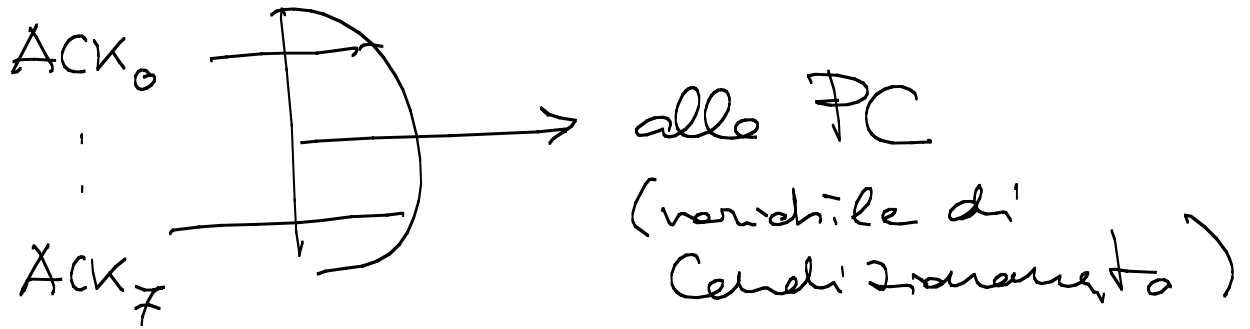
(= 110)  $map, \emptyset$  } ricevuto,  $w$ , pronto  
 CHECKSUM errato

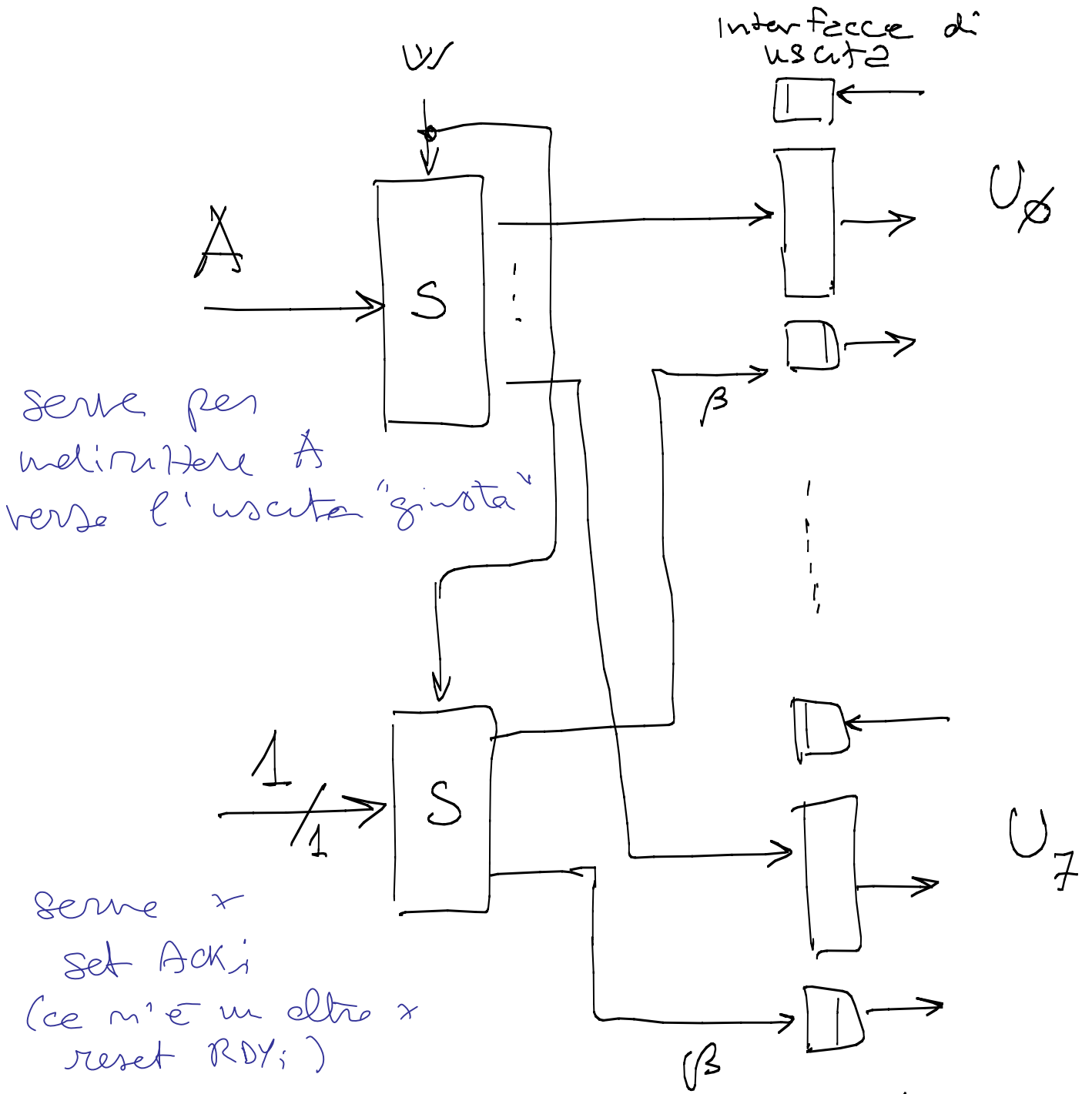
(= 111)  $A \rightarrow OUT[w]$ , set  $RDY[w]$   
 reset  $RDY_{in}$ , set  $ACK_{in}$ , reset  $ACK[w]$ ,  $\emptyset$

caso "normale":

ricevuto,  $w[w]$  pronto,  
 checksum corretto

# Parte Operativa





serve x  
set ACK<sub>i</sub>  
(ce n'è un altro x  
reset RDY<sub>i</sub>)

viene scritto il registro dell'interfaccia  
di uscita identificata dalle uscite  
della rete INDINT

tempo di ciclo

$$T_{\omega_{po}} = \max \left\{ t_p, \left( 5t_p + 5t_p + 2t_p \right) \right\}$$

or controllo checksum

$$= 12t_p$$

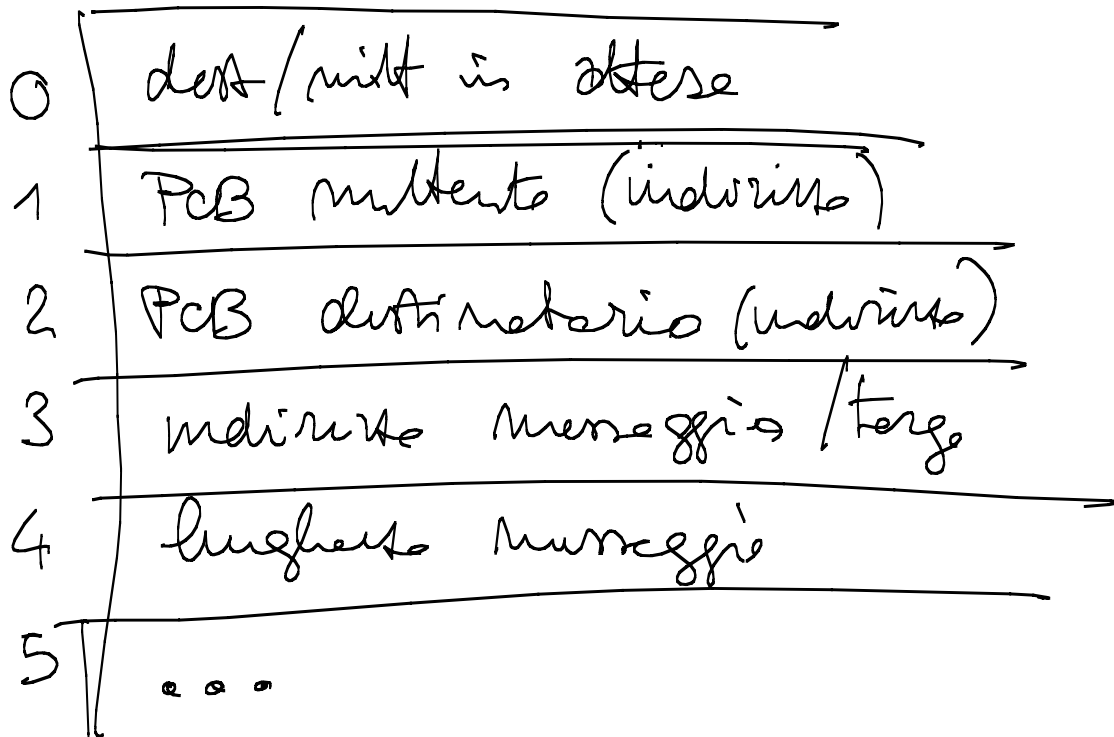
$$\begin{array}{l} T_{\sigma_{pc}} = \emptyset \\ T_{\omega_{pc}} = 2t_p \end{array} \left. \vphantom{\begin{array}{l} T_{\sigma_{pc}} \\ T_{\omega_{pc}} \end{array}} \right\} \begin{array}{l} \text{1 sola misurazione} \\ \text{quindi ha un} \\ \text{unico stato interno} \end{array}$$

$$T_{\sigma_{pa}} = \emptyset \left. \vphantom{T_{\sigma_{pa}}} \right\} \begin{array}{l} \text{solo trasferimenti} \\ \text{fra registri} \end{array}$$

$$\begin{aligned} \tau &= T_{\omega_{pa}} + \max \left\{ T_{\omega_{pc}} + T_{\sigma_{pa}}, T_{\sigma_{pc}} \right\} \\ &+ \delta = 12t_p + 2t_p + t_p = \\ &= 15t_p \end{aligned}$$

## Domanda 2

Struttura dati canale



pseudo codice

```
if (destinatario in attesa) {  
  copie msg nelle var target  
  sveglia destinatario  
  cancello destinatario in attesa  
}  
else {  
  memorizzo indirizzo msg nel canale  
  " lunghezza " " " "  
  setto mittente in attesa  
  dico al schedulatore  
}
```



precone in esecuzione  
va in attesa

primo precone dello  
lista pronti va  
in esecuzione



Send :	LOAD R <sub>canale</sub> , R <sub>i</sub> , R <sub>destatere</sub>
Controllo destinatarie	IF = R <sub>destatere</sub> , R <sub>vero</sub> , GO
mi metto in attesa →	STORE R <sub>canale</sub> , R <sub>0</sub> , R <sub>vero</sub>
memoria dati x copia	ADD R <sub>0</sub> , #4, R <sub>i</sub>
memoria dati x copia	STORE R <sub>canale</sub> , R <sub>i</sub> , R <sub>memeggi</sub>
memoria dati x copia	INC R <sub>i</sub>
memoria dati x copia	STORE R <sub>canale</sub> , R <sub>i</sub> , R <sub>memeggi</sub>
mi deschedolo	MOV R <sub>myPCB</sub> , R <sub>PCB<sub>is</sub></sub>
mi deschedolo	"salvataggio dati nel PCB"
mi deschedolo	GOTO scheduler

] de qui  
non torn

GO : STORE Rcanale, Ri, R falso

base  
inizializzato  
per l'orizzonte

ADD R0, #3, Ri

LOAD Rcanale, Ri, R l'orizzonte

LOOP :

LOAD Rmessaggio, R l'indirizzo messaggio, R l'indirizzo

STORE R l'orizzonte, R l'indirizzo messaggio, R l'indirizzo

code  
di copia

DEC R l'indirizzo messaggio

IF  $\geq \emptyset$  R l'indirizzo messaggio, LOOP

sveglio  
dest.

ADD R  $\emptyset$ , #2, Ri

LOAD Rcanale, Ri, R l'indirizzo - svegliopel

CALL Sveglia precare, R ret1

ritorno  
delle  
send

GOTO R ret

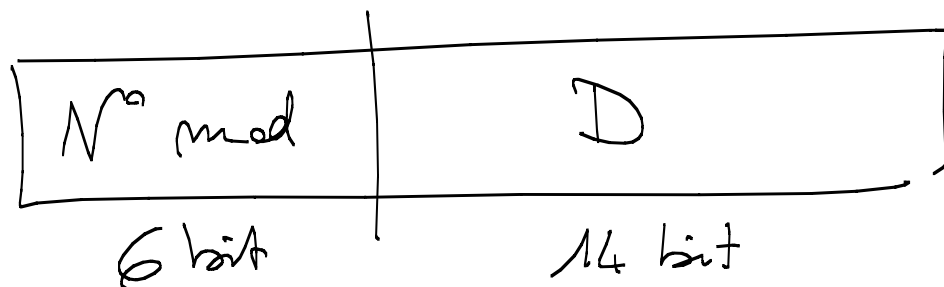
## Domande 3

1) La dimensione dello pagine di memoria virtuale è completamente indipendente dell'organizzazione della cache, quindi non posso dedurre alcunché

2) Il processore può inviare la richiesta (1 $\mu$ s) quindi deve attendere: 3 $\mu$ s del tempo di accesso della cache e 1 $\mu$ s per la misurazione che controlla l'errore. Quindi può avere un nuovo dato ogni 5 $\mu$ s. Pertanto la banda massima è  $B = \frac{1}{5\mu}$

3) Occorrono 64 moduli da  
16 K per fare 1 M.

Assumendo che l'indirizzo sia  
interpretato come



il contatore che seleziona lo  
però fra le 64 "lettre"  
di ciascuno dei moduli potrà  
essere realizzato con 3 livelli  
di logica (vedi dispense)

Pertanto  $t_a = 5t_p + 3t_p = 8t_p$   
modulo  
di memoria K