

Compito 18 gennaio 2007

Note Title

30/10/2008

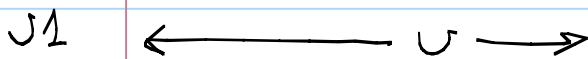
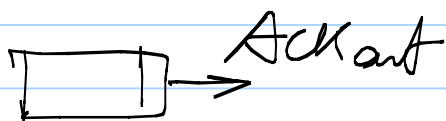
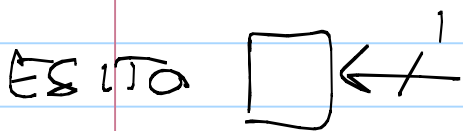
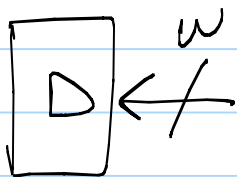
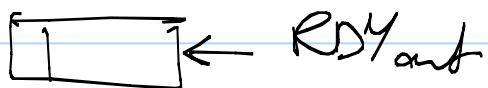
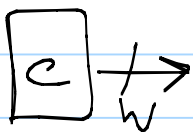
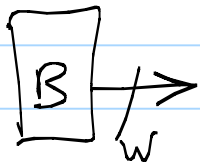
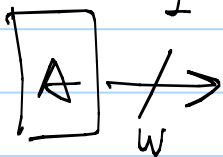
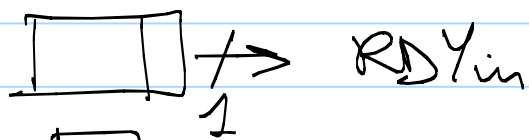
1° esercizio

Soluzione 1

Ipotesi

- i bit aw delle due sono memorizzati nei registri aw_1, \dots, aw_3 da 1 bit, le cui uscite sono variabili di condizionamento per le Porte Centrali
- $U1$ manda comunque tutte le triple (N) a U
- $U2$ si prepara a ricevere il risultato quando manda $\langle A_N, B_N, C_N \rangle$ (U non deve fare Ack prima di spedire i risultati)
- Utilizziamo 3 AUV per calcolare $(A+B) + (C+temp)$ in un ciclo solo

Interfaccia vs. U1



0. ($RDY_{in} = \phi$) map, ϕ

(= 1) 0 \rightarrow ($Rtemp, OW_1, OW_2, OW_3$)

A \rightarrow Ntemp, reset RDY_{in} , set ACK_{in}

0 \rightarrow Etemp, 1

1. ($RDY_{in}, \sigma(OW_1, OW_2, OW_3), zero(Ntemp) = 0$) map, 1

(= 100) Ntemp - 1 \rightarrow Ntemp, (A+B) + (C + Rtemp)
 \rightarrow Rtemp, reset RDY_{in} , set ACK_{in} , 1

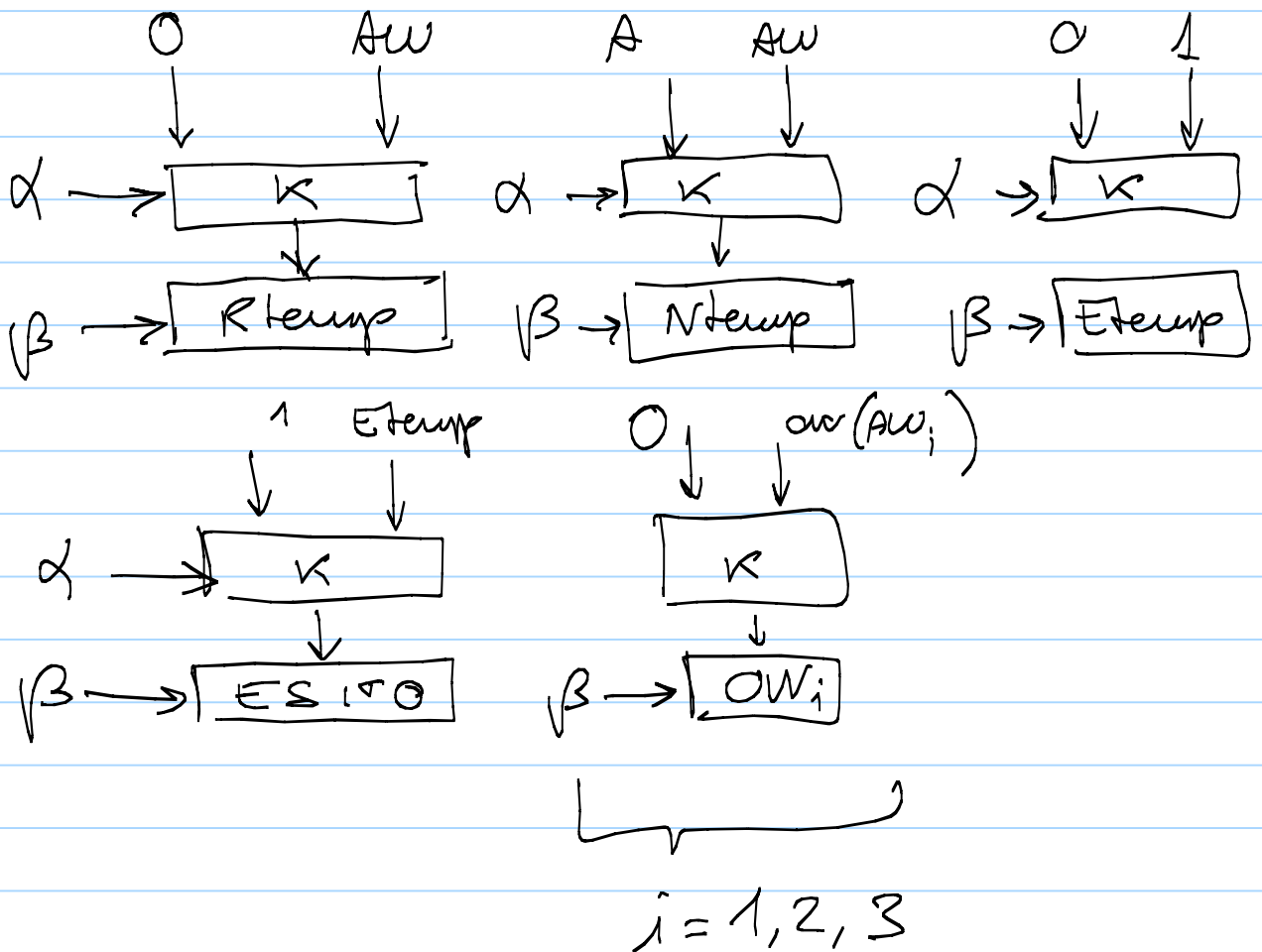
(= 110) 1 \rightarrow Etemp, Ntemp - 1 \rightarrow Ntemp,
(A+B) + (C + Rtemp) \rightarrow Rtemp
reset ACK_{in} , set RDY_{in} , 1

(= 101) Etemp \rightarrow ESTO, Rtemp \rightarrow D,
reset ACK_{out} , set RDY_{out} , ϕ

(= 111) 1 \rightarrow ESTO, Rtemp \rightarrow D,
reset ACK_{out} , set RDY_{out} , ϕ

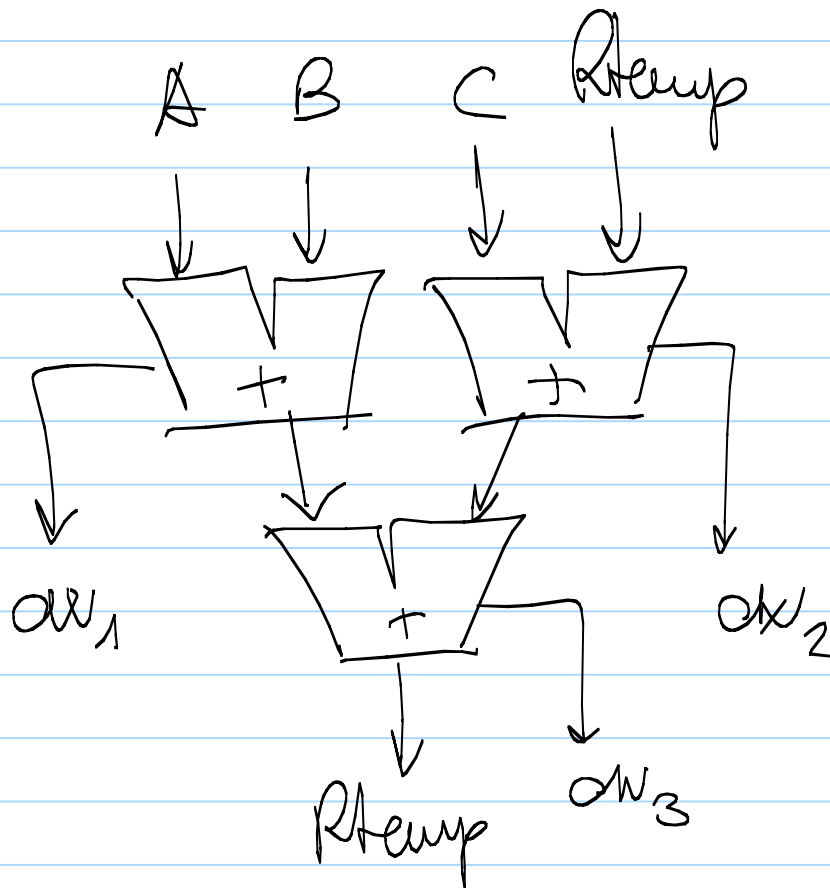
Class dei regjstri

(individuo fuor i possibil ingressi)



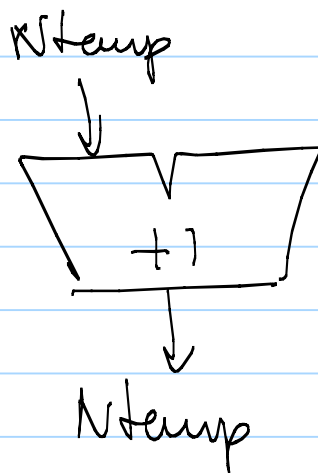
Operazioni

Ci serve solo
due "operazioni": l'aggiornamento dello
stato temporaneo!



che serve anche per avere l'ingresso dei
registri w_1

e la funzione che calcola il
decremento della variabile
d'iterazione !



che rimane fare con una quarta
AVV

Valutazione del ciclo di clock

$T_{\omega_{PO}}$: uscite di registri (RDY_{in}) $0T_p$
 fusione di registri (or (ow_1, ow_2, ow_3)) $1T_p$
 zero ($Ntemp$)
 not (or ($Ntemp_0 \dots Ntemp_{s1}$)) $2T_p$

$2T_p$

$T_{\sigma_{PO}}$ per il calcolo di ow_i ← $\begin{matrix} \text{cambio} \\ \text{lungo} \\ \text{in } \sigma_{PO} \end{matrix}$

$5T_p$ + $5T_p$ + $2T_p$
 $AW_1 \parallel AW_2$ AW_3 $K \text{ in } di$

$12T_p$

PC

2 stati

3 round di condizionamento

Labello dello stato con max
16 righe

In realtà:

max 7 fasi in 2 μ -istruzioni

ogni fase ha 4 istruzioni

(3 round di condizionamento +
1 round di stato)

e quindi

2 livelli di logica

$\times T_{Opc}$ e $\times T_{Wpc}$

$$T_{Opc} = T_{Wpc} = 2T_p$$

$$\begin{aligned}
 \mathcal{L} &= T_{\omega_{PO}} + \overset{2tp}{T_{\sigma_{PC}}} + \overset{2tp}{T_{\omega_{PC}}} + \overset{12tp}{T_{\sigma_{PB}}} \\
 &\quad \underset{\substack{1tp \\ \delta_s}}{\max} \left\{ T_{\sigma_{PC}}, \left\{ T_{\omega_{PC}} + T_{\sigma_{PB}} \right\} \right\} + \\
 &\quad \text{17tp}
 \end{aligned}$$

Tempo medio di esecuzione

$$T = (N+2) \mathcal{L}$$

1 per ϕ . 1 per l'ultimo $(N+1)$ esec.
 mu-istruzione 1

$$= 17(N+2)tp$$

Comments

→ memorizzare $0W$; in registri non è strettamente necessario

→ Richiede un ciclo in + per testare l'ultima overflow

→ uso di $3 AW \times (A+B+C+Temp)$
comperta un ritardo per ad
dennere $2(T_{AW})$ nello Q_{PO}

Varianti

Tener conto di ACKout

(non dare per scontato che U1 sia pronto a ricevere il risultato quando arriva l'ultima triple)

Nella 1. devo testare anche ACKout

1. (RDY_{in}, 01(OW₁, OW₂, OW₃), zero(Ntemp), ACK_{out}

= 0 - - -) map, 1

(= 100 -)

(= 110 -)

} come nel caso precedente

(= 1-10)

map, 1 // aspetto ACK

(= 1011)

Etemp → ESTO, Rtemp → D,
reset ACK_{out}, set RDY_{out}, ∅

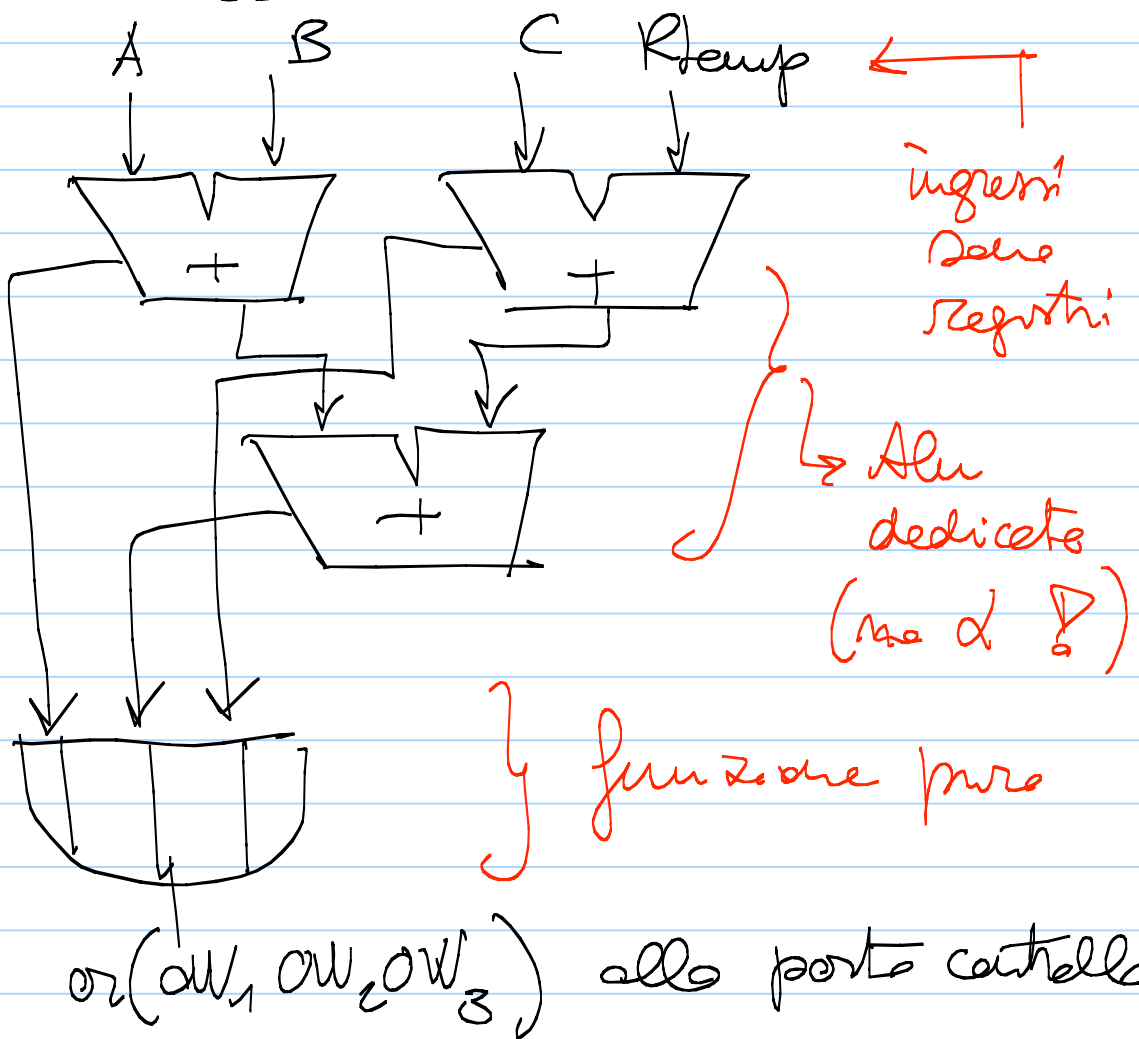
(= 1111)

1 → ESTO, Rtemp → D,
reset ACK_{out}, set RDY_{out}, ∅

Soluzione 2

Dal momento che gli OW_i non dipendono da α e β , evitiamo di utilizzare registri e li testano direttamente.

Vuol dire che in W_{PO} dobbiamo produrre



Di conseguenza nel μ -codice, per testare questa condizione scriveremo

1. $(\dots or(ar(A+B), ar(c+Rtemp), ar(A+B+c+Rtemp)))$
↳ la abbreviamo con $cond(A, B, c, Rtemp)$

A questo punto il μ -programma può essere scritto in modo da controllare l'esito degli ar nello stesso fase dove viene accumulata la relativa "somma" in Rtemp

ATTENZIONE !

però !

avzo μ -18 istruzioni tipo

(... COND(A, B, C, Rtemp) ...)

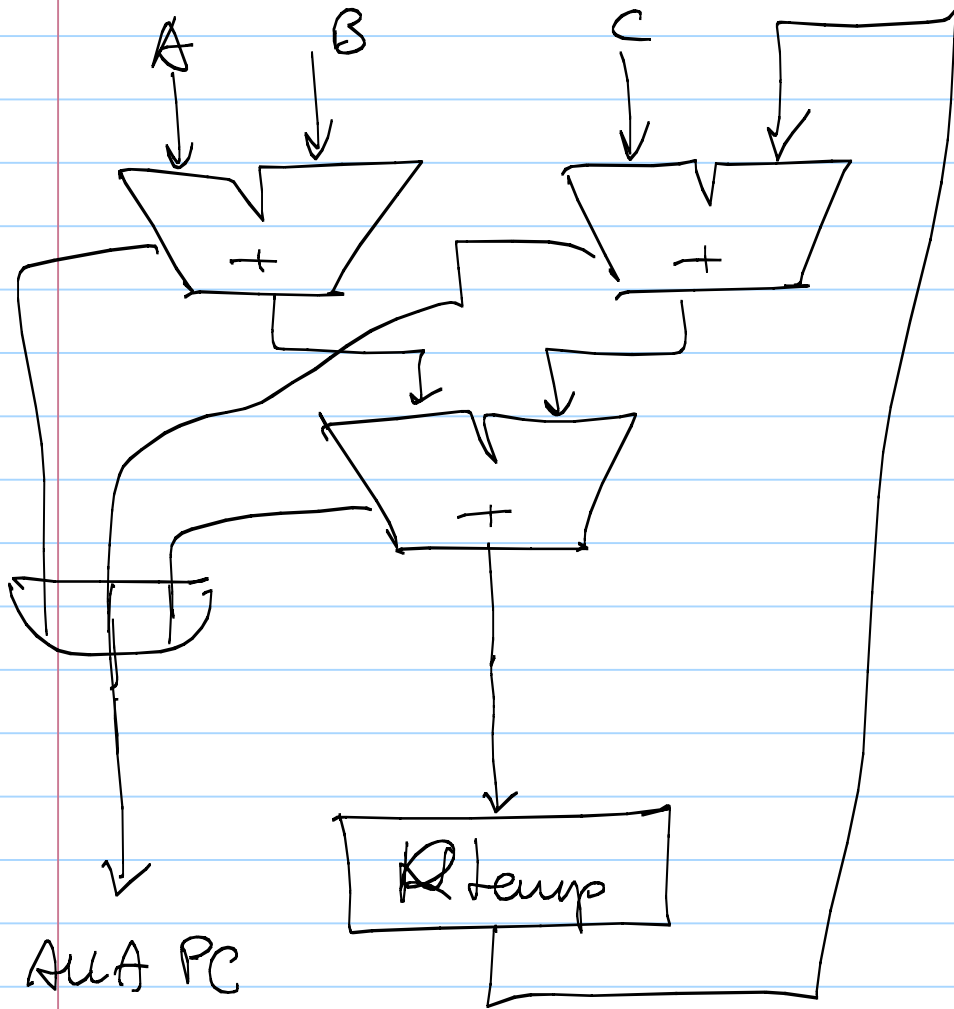
... , (A+B) + (C + Rtemp) \rightarrow Rtemp, ...

In questo caso, però non dobbiamo pensare che servono 6 ALU, 3 per calcolare la condizione e 3 per calcolare la somma.

Quando sono stabili i flag ALU, sono anche stabili le uscite dei circuiti delle ALU.

Tali risultati rimangono validi fino al prossimo ciclo di clock.

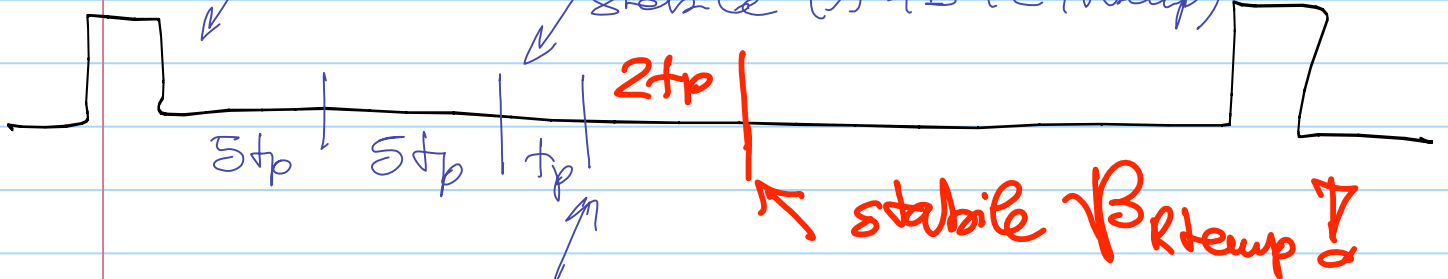
utilizzando solo 3 ALU



iniziale
scrittura
numero
state
RO

A, B, C, Rtemp: uscite stabili

stabile $(A+B+C+Rtemp)$



stabile $\forall Rtemp$!

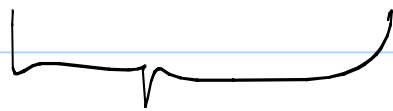
stabile $COND(A, B, C, Rtemp)$

quindi \times questa particolare
fase:

$$T_{\omega_{ps}} = 11 \Delta p$$

$$T_{\omega_{pc}} = 2 \Delta p$$

$$T_{\sigma_{ps}} = \cancel{\Delta p}$$



naturalmente questa vale se
nella fase l_0 ($A+B+\dots$) $\rightarrow R_{temp}$
fase l_0 solo $\mu-q_p$.

Se l_0 $\mu-q_p$ fase

$(A+B+C+R_{temp}) \rightarrow R_{temp}$, $N_{temp} - 1 \rightarrow N_{temp}$

allora T_{Op} sarebbe dato

$$\text{da } T_{Op} = Z_{Op} \begin{cases} Z_{Op} & \text{K su} \\ & \text{ingreso} \\ S_{Op} & \text{ALU} \end{cases}$$

(confronto domanda 2 dello
secondo esercizio)

Dunque in questo caso il

μ -programma in μop
avrebbe come segue:

\emptyset . ($RDY_{in} = 0$) Map, \emptyset

(= 1) $0 \rightarrow Rtemp$, $A \rightarrow Ntemp$, reset RDY_{in}
set ACK_{in} , 1

1. (RDY_{in} , $COND(A, B, C, Rtemp)$, zero ($Ntemp$) = 0 --)
Map, 1

(= 100) $Ntemp - 1 \rightarrow Ntemp$, reset RDY_{in}
 $A + B + C + Rtemp \rightarrow Rtemp$,
set ACK_{in} , 1

(= 110) $1 \rightarrow Etemp$, $Ntemp - 1 \rightarrow Ntemp$
 $A + B + C + Rtemp \rightarrow Rtemp$
set ACK_{in} , reset RDY_{in} , 1

(= 0-1) $Etemp \rightarrow ESTO$, $Rtemp \rightarrow D$
reset ACK_{out} , set RDY_{out} , \emptyset

ultima iterazione già effettuata, devo solo
comunicare cosa ho calcolato ...

Calcolo del ciclo di clock

$$T_{W_{PO}} = 11t_p \quad (2ALU + OR)$$

$$T_{O_{PA}} = 7t_p \quad (1ALU + K * Ntemp \dots)$$

il calcolo di $A+B+C+Ktemp$
è sempre effettuato dopo
aver testato la condizione
quindi è già compreso
nell' $11t_p$ di $T_{W_{PO}}$

$$T_{O_{PC}} = T_{W_{PC}} = 2t_p$$

(stesse considerazioni
della soluzione 1)

$$C = 11t_p + (2t_p + 7t_p) + t_p = 21t_p$$

$$T = 21(N+2)t_p$$

Confronto delle 2 soluzioni!

La seconda soluzione impiega + tempo della prima:

il tempo preso per il calcolo di $A+B+C+R_{temp}$ "copriva"

il tempo per il calcolo di $N_{temp}-1$ che adesso invece è progetto riciclamento ($A+B+C+R_{temp}$ in T_{op} e $N_{temp}-1$ in T_{op})

a fronte di questo, il numero di μ -istruzioni da eseguire non cambia

⇒ la seconda sol è meno veloce