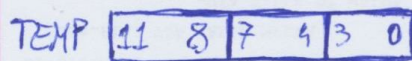
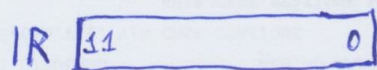
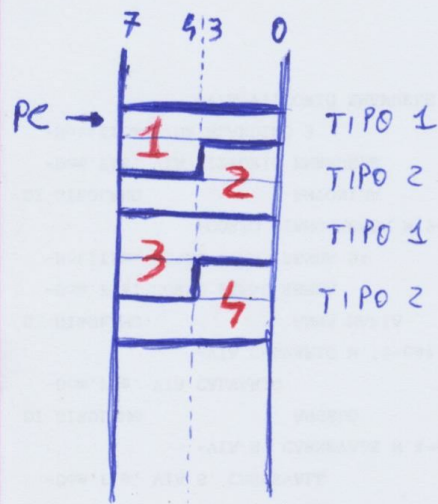


Sia dato un sistema con RAM di parole ad 8 bit e istruzioni di 12 bit. Progettare la fase di fetch considerando che tra una istruzione e l'altra non ci sono spazi liberi e che l'istruzione deve essere caricata nell'IR in un unico colpo di clock.



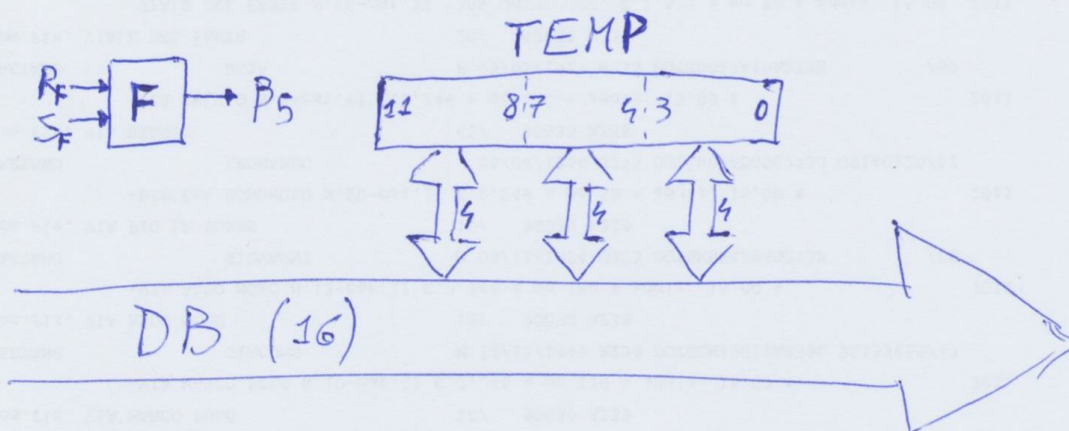
Il registro TEMP è necessario per assemblare i pezzi di istruzione. Solo quando tutta l'istruzione è contenuta in TEMP si può portare in IR.

Esistono 2 tipi di istruzioni che si ripetono alternativamente:

- Quelle che partono dal bit 7 (TIPO 1)
- Quelle che partono dal bit 3 (TIPO 2)

Si può usare un Flip-Flop RS che verrà settato prima a 0, poi a 1, poi a 0 e così via. Questo Flip-Flop serve a "ricordare" se l'istruzione da leggere è di Tipo 1 o di Tipo 2.

MODIFICHE ALL'ARCHITETTURA



RTL:

if ($F=0$) then // $B_5=0$, istruzione TIPO 1

$PC \rightarrow MAR, PC+1 \rightarrow PC;$

$M[MAR] \rightarrow MBR;$

$MBR \rightarrow TEMP_{11-4};$

$PC \rightarrow MAR;$

$M[MAR] \rightarrow MBR;$

$MBR_{7-4} \rightarrow TEMP_{3-0};$

$TEMP \rightarrow IR, 1 \rightarrow F; // S_F=1, R_F=0$

else // $B_5=1$, istruzione TIPO 2

$PC \rightarrow MAR, PC+1 \rightarrow PC;$

$M[MAR] \rightarrow MBR;$

$MBR_{3-0} \rightarrow TEMP_{11-8};$

$PC \rightarrow MAR, PC+1 \rightarrow PC;$

$M[MAR] \rightarrow MBR;$

$MBR \rightarrow TEMP_{7-0};$

$TEMP \rightarrow IR, 0 \rightarrow F; // S_F=0, R_F=1$

fi