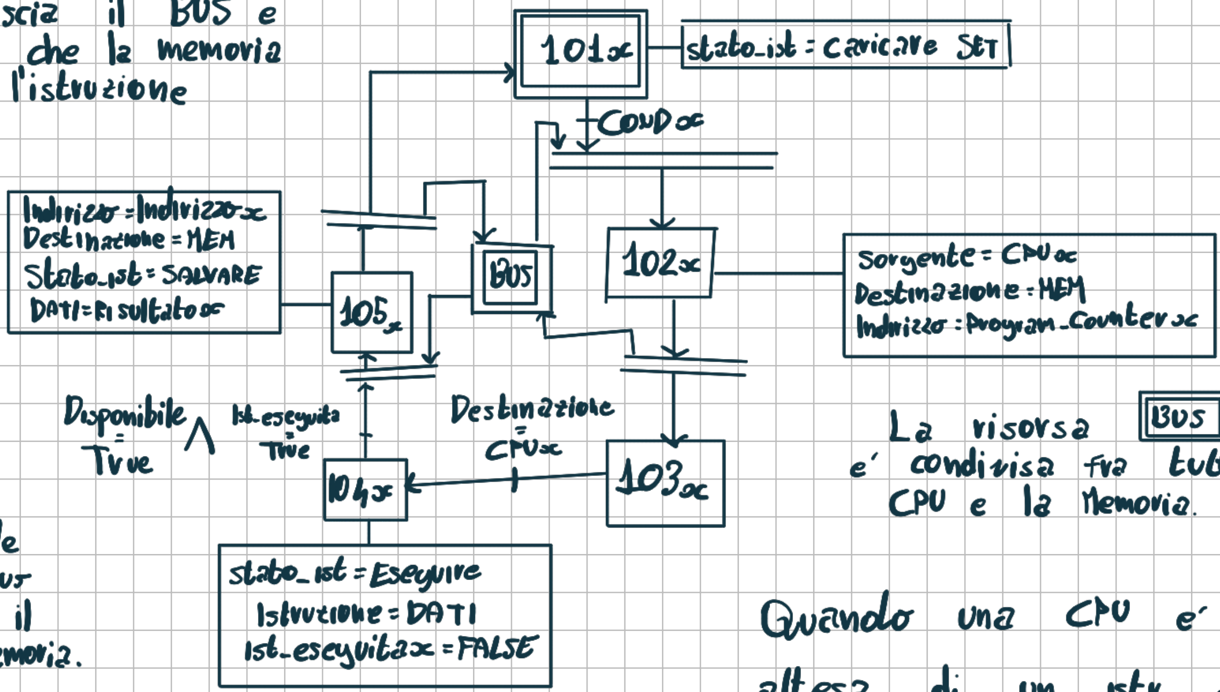
Es 1) Quando la CPU pone sorgente, destinazione ed indirizzo deve accertarsi che il bus sia libero. Il bus può essere acceduto da più sorgenti solo in lettura

102: accedo al BUS e scrivo
103: si rilascia il BUS e si attende che la memoria restituisca l'istruzione

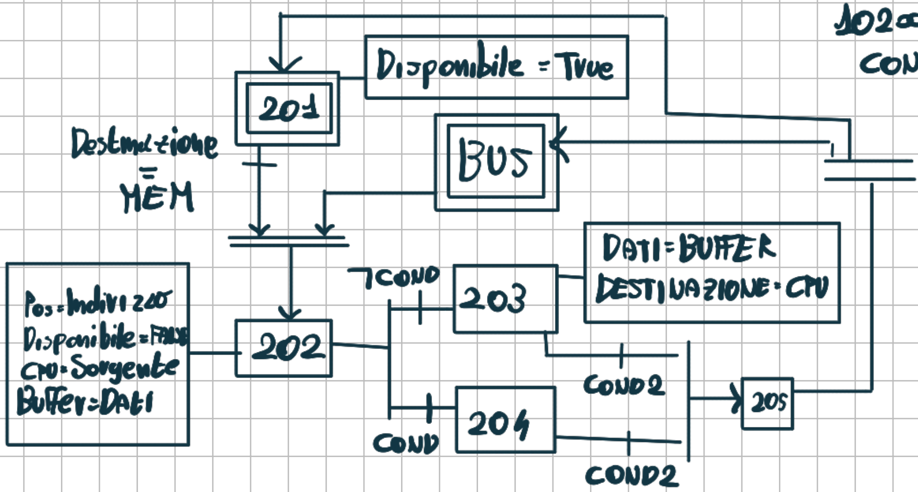


105x = Riprende accesso al bus per mandare il ris. alla memoria.

La risorsa BUS è condivisa fra tutte le CPU e la Memoria.

Quando una CPU è in attesa di un istr. dalla memoria (stato 103) nessuna CPU deve scrivere sul BUS, per questo fra 101x e 102x ci sta la condizione COND x

$COND\,x$ = Destinazione $\neq$ MEM $\wedge$ $\forall y \neq x$, 103_y = FALSE

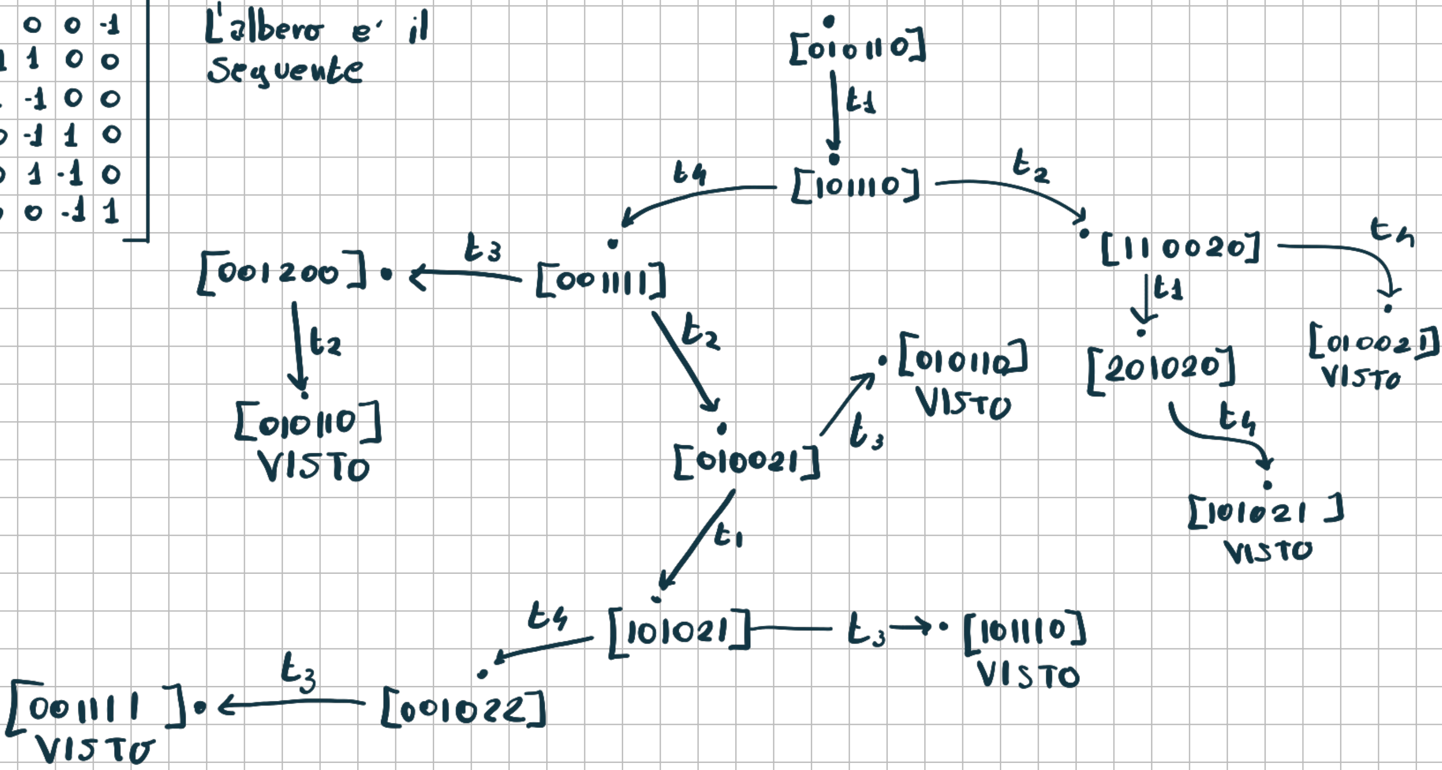


COND1 = Posizione > SLACK
COND2 = Lettura o Scrittura terminata

Es 2) La matrice di incidenza e'

$$C = \begin{bmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

L'albero e' il seguente



Dall'albero si nota chiaramente che la rete e' reversibile, viva e limitata, in particolare, non ci sono MAI piu' di 2 token per posto. Calcolo ora i P-invarianti studiando $Ker(C^T)$

$$C^T = \begin{bmatrix} 1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{cases} \gamma_1 - \gamma_2 + \gamma_3 = 0 \\ \gamma_2 - \gamma_3 - \gamma_4 + \gamma_5 = 0 \\ \gamma_4 - \gamma_5 - \gamma_6 = 0 \\ -\gamma_1 + \gamma_6 = 0 \end{cases} \Rightarrow \begin{cases} \gamma_2 = \gamma_1 + \gamma_3 \\ \gamma_4 = \gamma_1 + \gamma_5 \\ \gamma_1 = \gamma_6 \end{cases} \Rightarrow Ker(C^T) = \left\{ [a, a+b, b, a+c, c, a]^T \; a,b,c \in \mathbb{R} \right\}$$

$\Rightarrow$ i P-invarianti sono:

$$\left.\begin{aligned} \gamma_1 &= [0\,1\,1\,0\,0\,0]^T \\ \gamma_2 &= [0\,0\,0\,1\,1\,0]^T \\ \gamma_3 &= [1\,1\,0\,1\,0\,1]^T \end{aligned}\right\}$$

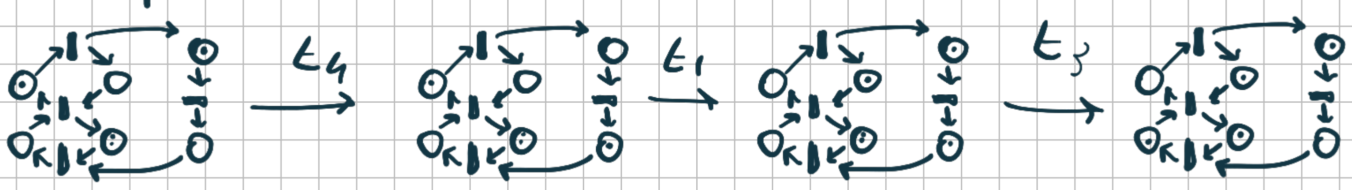
L'unione dei supporti copre P: $\|\gamma_1\| \cup \|\gamma_2\| \cup \|\gamma_3\| = P$ Quindi la rete e' conservativa. Scrivo le 3 equazioni di invarianza, che sono: $\gamma_i^T \cdot x = \gamma_i^T \cdot x_0$

$$\gamma_1^T \cdot x = x(p_2) + x(p_3) = [011000] \cdot [010110]^T = 1$$

$$\gamma_2^T \cdot x = x(p_4) + x(p_5) = [000110] \cdot [010110]^T = 2$$

$$\gamma_3^T x = x(p_1) + x(p_2) + x(p_4) + x(p_6) = [110101] \cdot [010110]^T = 2$$

Come si puo' notare dall'albero di raggiungibilita', la sequenza $t_4 \to t_1 \to t_3$ porta da $[110020]^T$ a $[101110]^T$.



Adesso studio Ker(C) per i T-invarianti.

$$C = \begin{bmatrix} 1 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \Rightarrow \begin{cases} \eta_1 = \eta_4 \\ \eta_1 = \eta_2 \\ \eta_2 = \eta_3 \\ \eta_3 = \eta_4 \end{cases} \Rightarrow Ker(C) = \{[x\ x\ x\ x]^T\ x \in \mathbb{R}\} \Rightarrow \text{il T-invariante è } \eta = [1\ 1\ 1\ 1]^T$$

⇒ Se le transizioni scattano un UGUALE numero di volte la rete torna nello stato originale. Sequenze ammissibili a partire da x_0 sono: $t_1 t_4 t_3 t_2$ e $t_1 t_4 t_2 t_3$

Es 3) Il sistema implementa delle RW-lock