Sistemi ad Eventi Discreti

a.a. 2016 - 2017

Prof. Luca Ferrarini

Sequential function chart

Tapparella automatica (1)

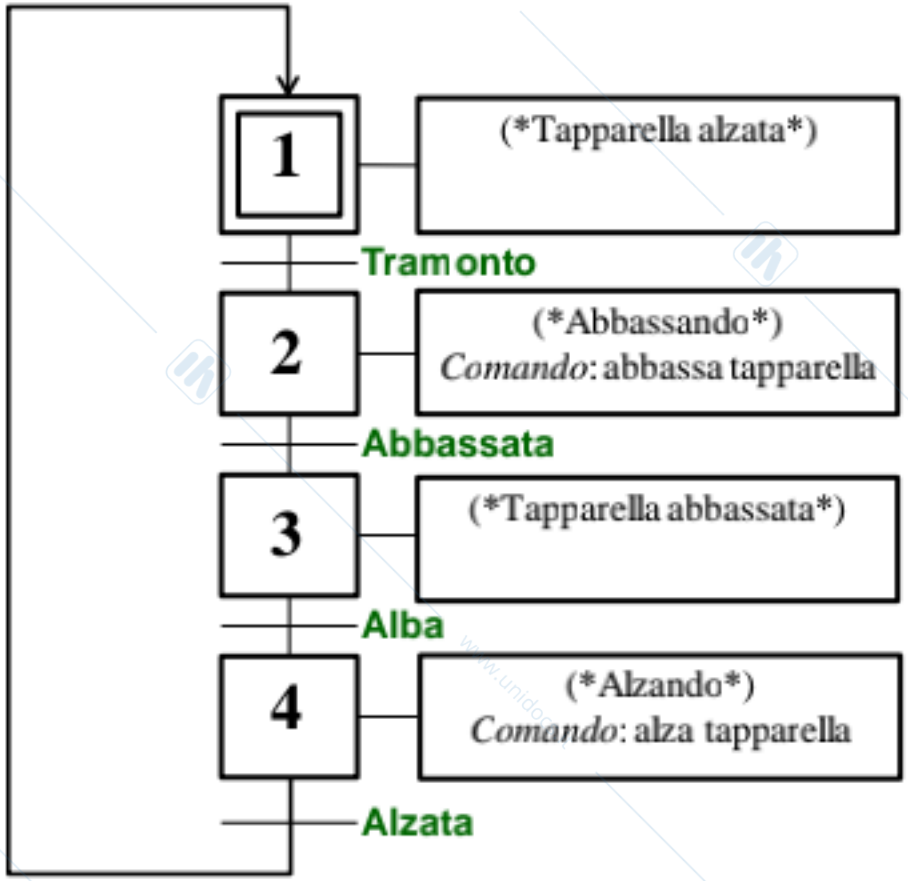
Esempio:

Vogliamo descrivere il funzionamento di un sistema di tapparelle automatiche. Le tapparelle devono scendere quando il sole tramonta e devono risalire quando il sole sorge.

Componenti:

- Motore per alzare ed abbassare le tapparelle
- Sensore di «tapparella alzata»
- Sensore di «tapparella abbassata»
- Sensore di luce esterna

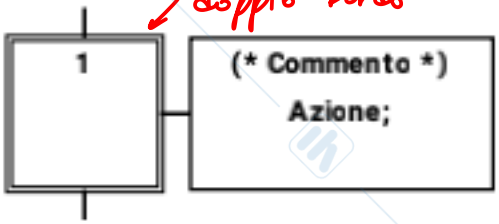
Tapparella automatica (2)



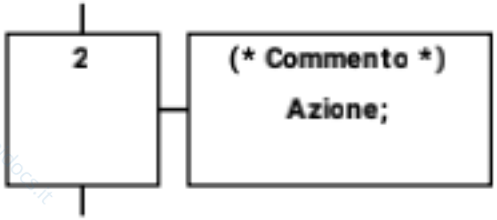
Elementi di base dell'SFC (1)

PASSI

- Passo iniziale (con azioni associate)



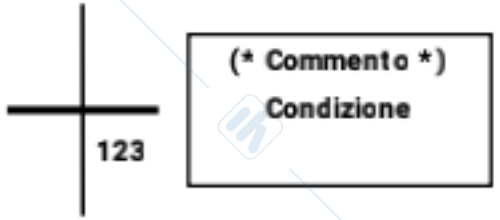
- Passo (con azioni associate)



Elementi di base dell'SFC (2)

TRANSIZIONI

Transizione (con condizioni associate)



Sono istantanea

REGOLE DI COMPOSIZIONE

- I collegamenti sono sempre orientati dall'alto verso il basso
- È possibile solo saltare da una transizione ad un passo (e da un passo ad una transizione)

REGOLE DI EVOLUZIONE

- **Situazione iniziale**: caratterizzata dai passi iniziali, che per definizione sono attivi all'inizio dell'esecuzione del programma SFC
- **Abilitazione di una transizione**: una transizione è detta abilitata quando tutti i passi precedenti connessi alla corrispondente transizione sono attivi
- **Attivazione di una transizione**: una transizione è detta attiva quando è abilitata e la condizione associata alla transizione vale TRUE
- **Modifica dello stato dei passi**: l'attivazione di una transizione, simultaneamente, porta allo stato attivo i passi immediatamente successivi ed allo stato inattivo i passi immediatamente precedenti
- **Attivazione simultanea di transizioni**: se più transizioni risultano attive nello stesso istante allora vengono tutte superate contemporaneamente

Gestione delle eccezioni:

- **Attivazione e disattivazione simultanea di un passo**: se, durante un'operazione, un passo viene contemporaneamente attivato e disattivato, la priorità viene data all'attivazione *(importante per le azioni PULSE)*

Elementi di base dell'SFC (5)

AZIONI

es. tapparella → erano tutti passi con azioni N → azione attiva solo quando entri nel passo

Azioni semplici:

Se non si mette niente è implicito che sia di tipo N

N	Non-Store	L'azione termina quando il passo diventa inattivo
S	Set (Stored)	L'azione continua anche quando il passo diventa inattivo e termina quando l'azione viene resettata
R	Reset	Termina un'azione attivata con i qualificatori S, SD, SL o DS
D	Time Delayed	Un timer viene settato quando il passo diventa attivo; se il passo è ancora attivo dopo l'azzeramento del timer, l'azione comincia e termina quando il passo si disattiva → appena entri in un passo inizia un timer e appena scade il timer l'azione comincia (mi ritarda l'azione) → Se esco prima del timer l'azione non inizia proprio
L	Time Limited	L'azione comincia quando il passo diventa attivo e continua finché il passo diventa inattivo o trascorre un certo intervallo di tempo
P	Pulse	L'azione comincia quando il passo diventa attivo e viene eseguita una sola volta
SD	Stored and time Delayed	L'azione comincia dopo un ritardo anche se il passo diventa inattivo e continua finché non resettata
DS	Delayed and Stored	Un timer viene settato quando il passo diventa attivo; se il passo è ancora attivo dopo l'azzeramento del timer, l'azione comincia e continua finché non resettata
SL	Stored and time Limited	L'azione comincia quando il passo diventa attivo e continua finché non viene resettata o non trascorre un certo intervallo di tempo

bisogna saperle

derivate di quelle essenziali

es. D5sec. → ritardo con timer di 5 secondi

Elementi di base dell'SFC (6)

AZIONI

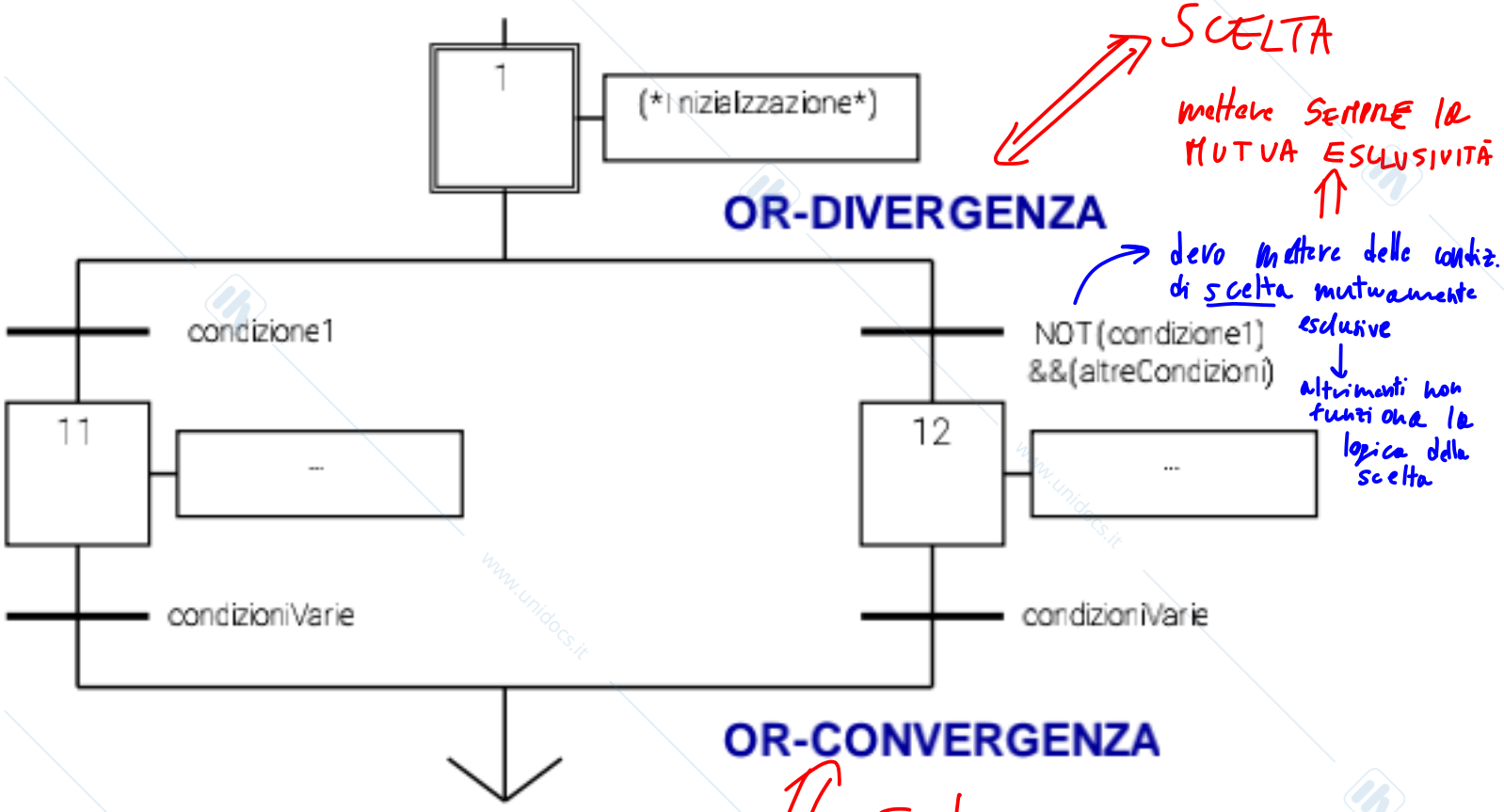
non le faremo

Azioni complesse:

- Le azioni complesse rappresentano altri programmi;
- La normativa IEC 61131-3 non specifica esattamente la sintassi per l'invocazione dei programmi all'interno dei passi;
- Ogni costruttore segue una propria convenzione.

Elementi di base dell'SFC (7)

STRUTTURE CLASSICHE



SCELTA

mettere SEMPRE la MUTUA ESCLUSIVITÀ

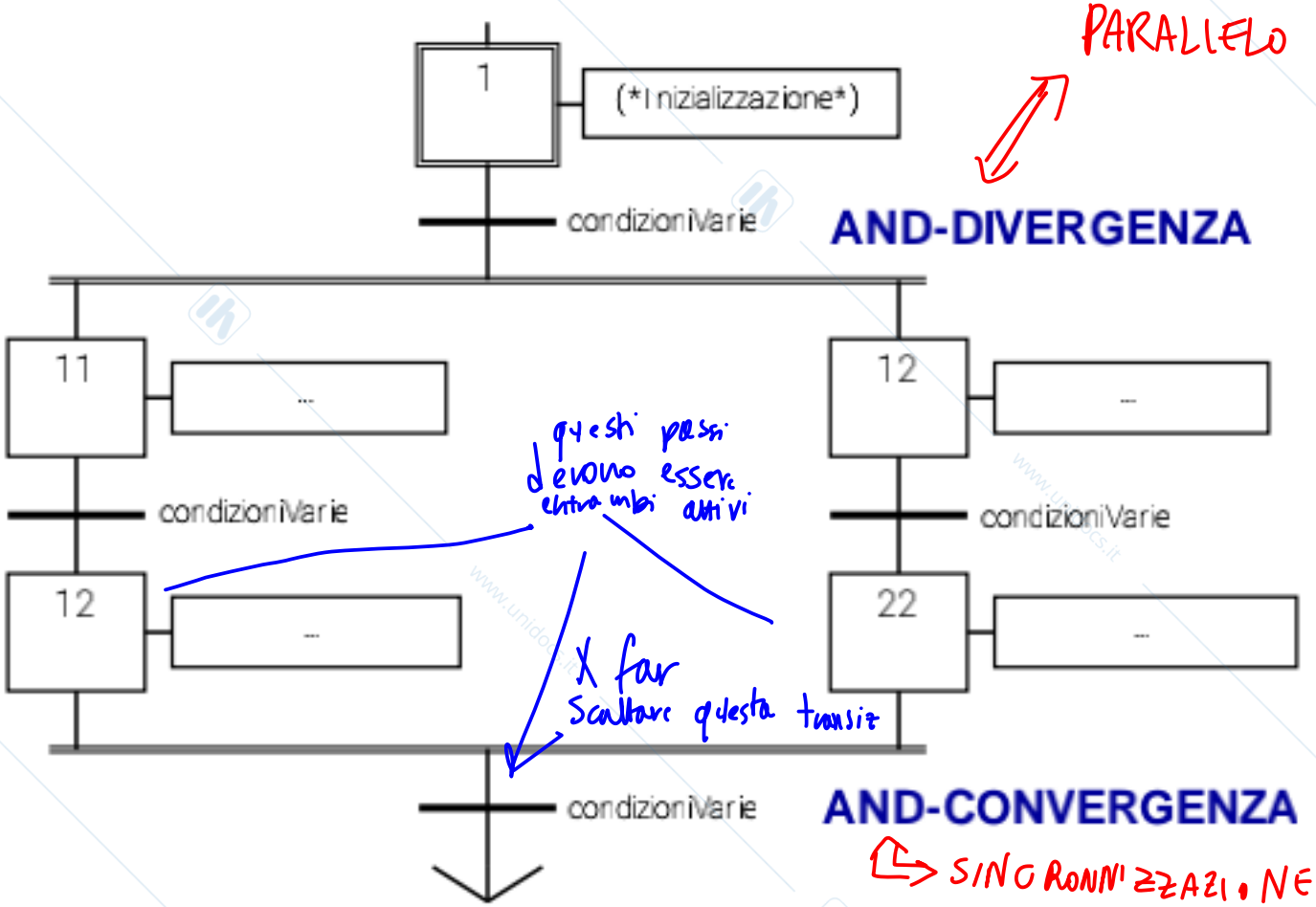
devo mettere delle condiz. di scelta mutuamente esclusive

altrimenti non funziona la logica della scelta

FINE SCELTA

Elementi di base dell'SFC (8)

STRUTTURE CLASSICHE



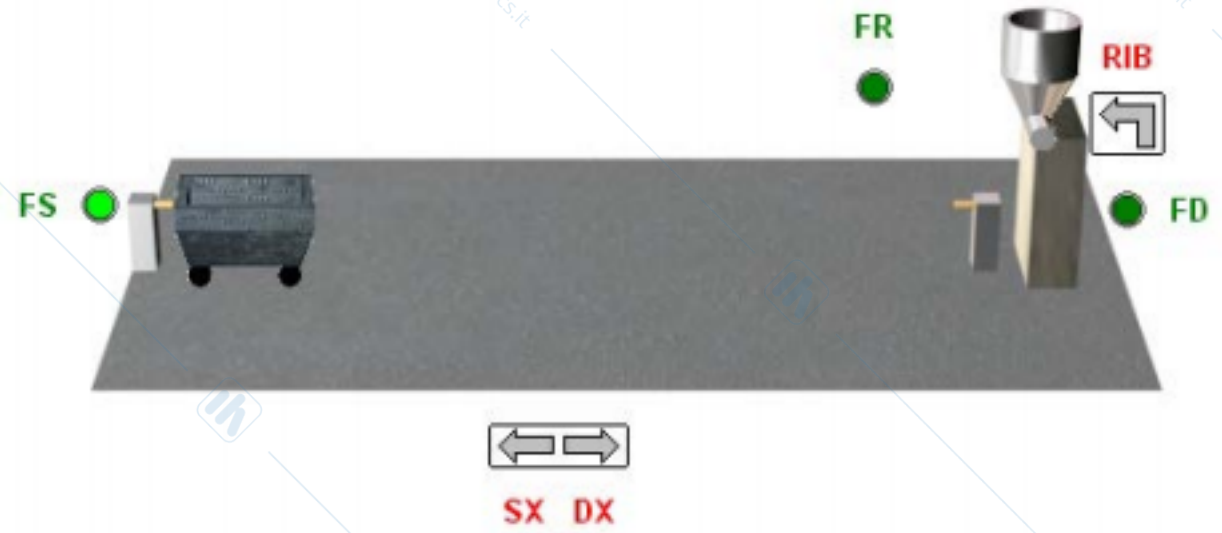
PARALLELO

questi passi devono essere entrambi attivi

X far Scattare questa transiz

SINCRONIZZAZIONE

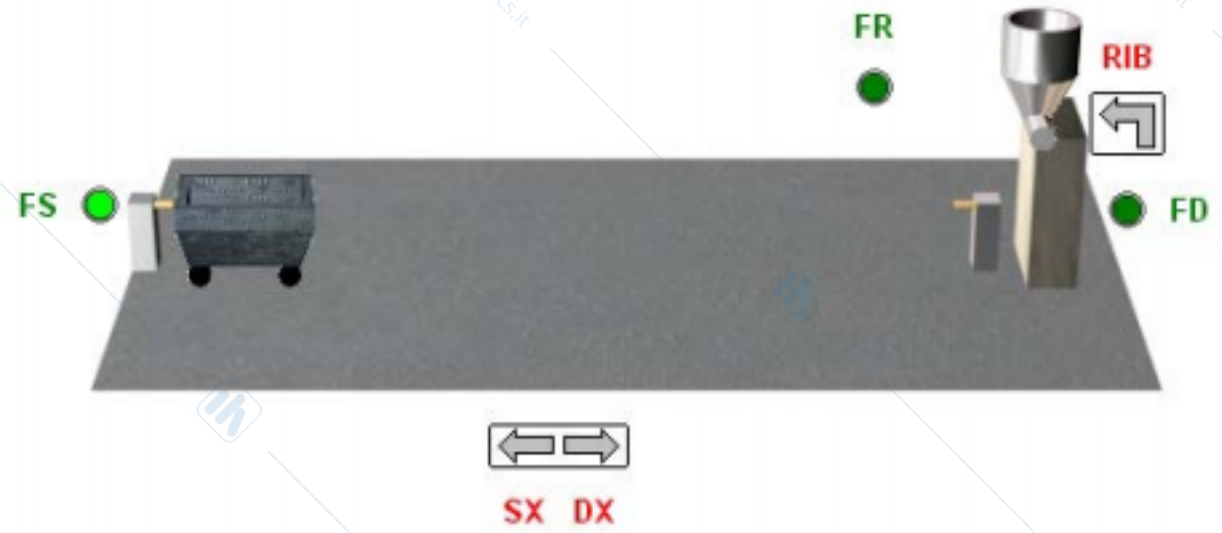
Il carrello (1)



in verde SENSORI
in rosso ATTUATORI

- Ogni volta che l'operatore aziona il pulsante di START, bisogna portare il carrello a destra.
- Quando il carrello è arrivato a destra, bisogna caricare il carrello ribaltando il serbatoio. (RIB)
- Alla fine del caricamento il carrello deve essere riportato a sinistra. (FR)
- Qualcun altro si occuperà di svuotare il carrello e di riempire nuovamente il serbatoio.

Il carrello (2)



Sensori (misure)

- **START**: inizia la sequenza
- **FS**: fine corsa sinistro
- **FD**: fine corsa destro
- **FR**: fine riempimento

Attuatori (comandi)

- **SX**: vai a sinistra
- **DX**: vai a destra
- **RIB**: ribalta serbatoio

Il carrello (3)

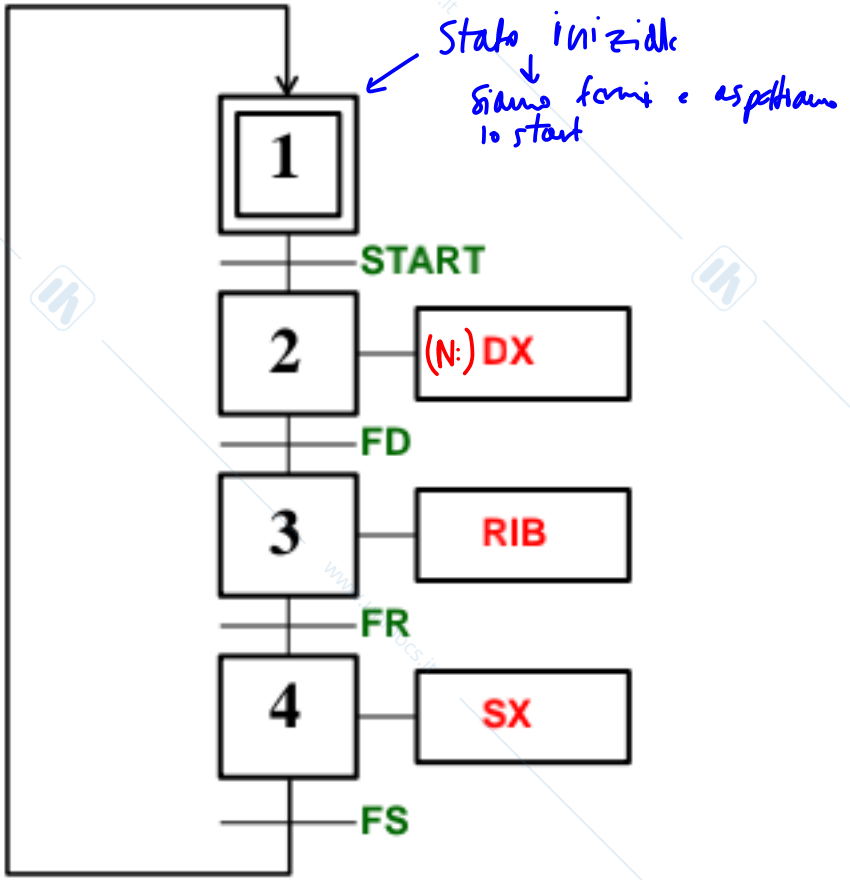
Attuatori (comandi)

- **SX**: vai a sinistra
- **DX**: vai a destra
- **RIB**: ribalta serbatoio

Sensori (misure) /input

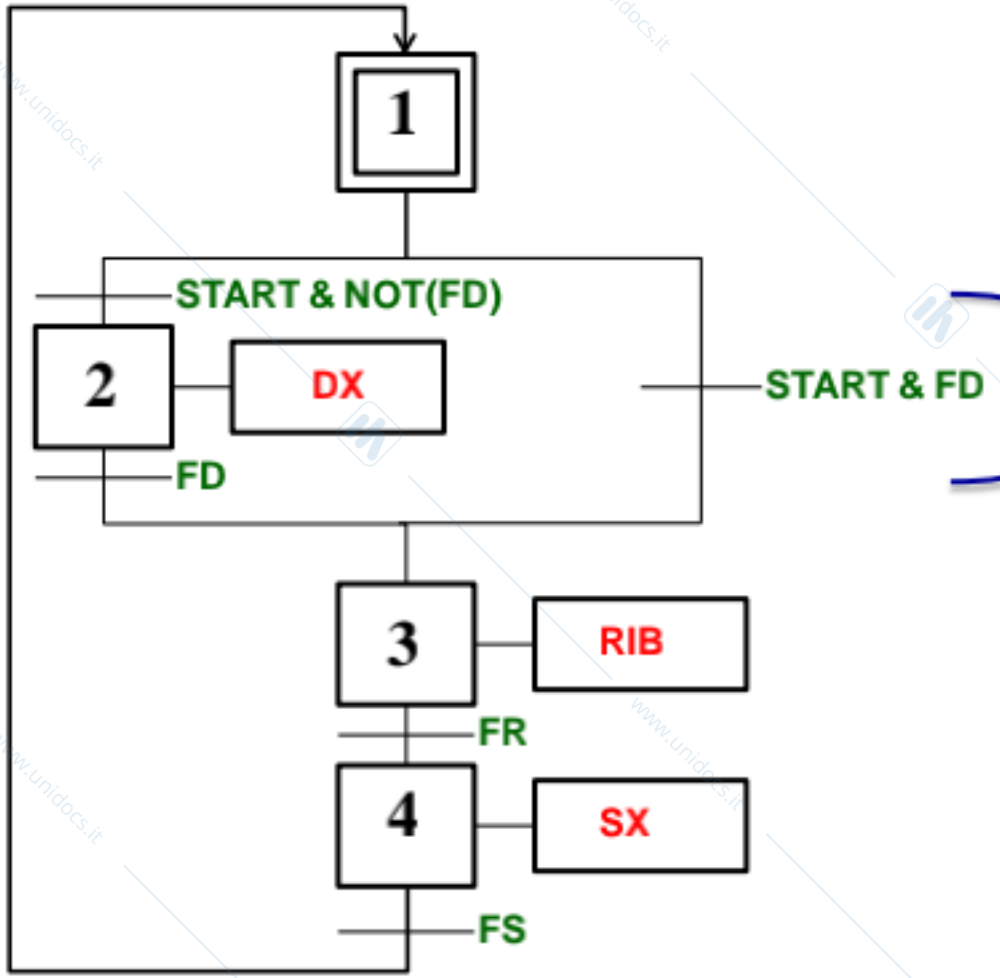
- **START**: inizia la sequenza
- **FS**: fine corsa sinistro
- **FD**: fine corsa destro
- **FR**: fine riempimento

In generale vale che:
INPUT/ SENSORE ≡ TRANSIZ.
ATTUATORE ≡ PASSO



Il carrello (4)

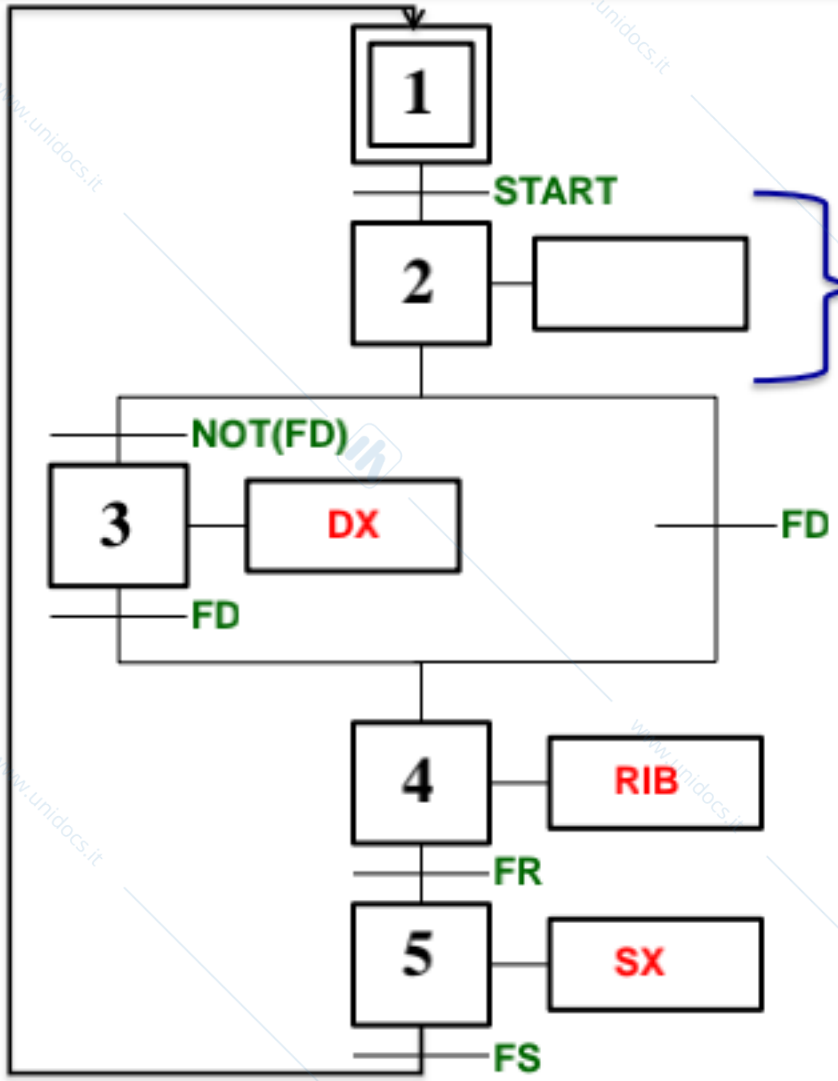
Sarebbe il caso di alt/disalt. simultanea di un passo
↳ se fosse già a destra il passo 2 sarebbe alt./disalt. simultanea
con FD pari a true



E se il carrello fosse già a destra?

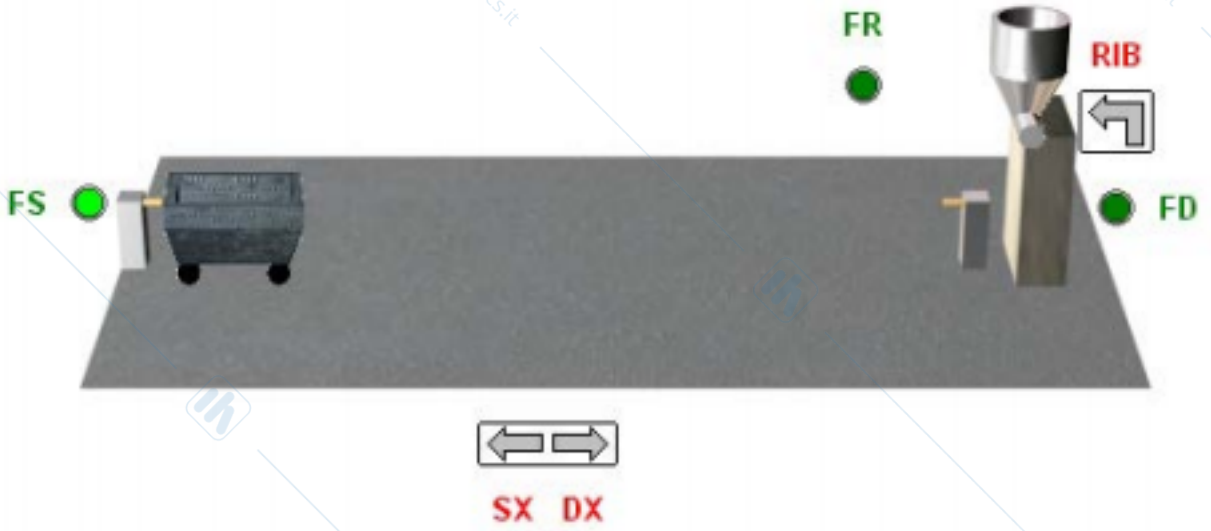
OR-DIVERGENZA

Il carrello (5)



Alternativa:
Per definire delle azioni di inizializzazione da eseguire ad ogni START?

Il carrello (6)

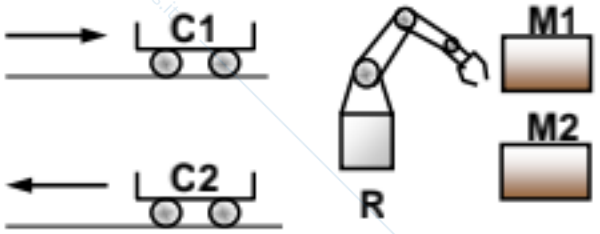


Sensori (misure)		Attuatori (comandi)
- FS: fine corsa sinistro	⟷	- SX: vai a sinistra
- FD: fine corsa destro	⟷	- DX: vai a destra
- FR: fine riempimento	⟷	- RIB: ribalta serbatoio
- START: inizia la sequenza		

perché devo avere sempre un feedback (sensore) della fine dell'azione che ho iniziato

Ad ogni attuatore corrisponde un sensore… perché?

Stazione di lavoro (1)



- Il robot R preleva il prodotto dal carrello C1 e lo deposita sulla macchina M1
- La macchina M1 effettua la lavorazione
- Il robot R trasferisce il prodotto dalla macchina M1 alla macchina M2
- La macchina M2 effettua la lavorazione
- Il robot R trasferisce il prodotto finito sul carrello C2

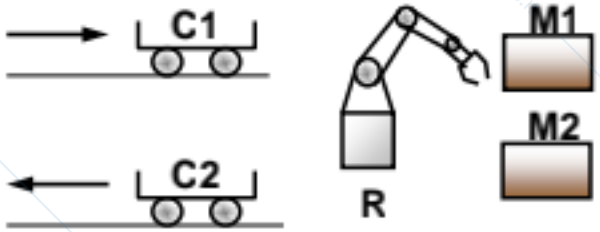
Comandi:

- **CM1:** carica M1
- **SM2:** scarica M2
- **TM2:** trasferisci da M1 a M2
- **LM1:** lavorazione M1
- **LM2:** lavorazione M2

Sensori (misure):

- **FCM1:** fine caricamento M1
- **FLM1:** fine lavorazione M1
- **FTM2:** fine trasferimento da M1 a M2
- **FLM2:** fine lavorazione M2
- **FSM2:** fine scaricamento M2
- **C1:** C1 presente
- **C2:** C2 presente

Stazione di lavoro (2)

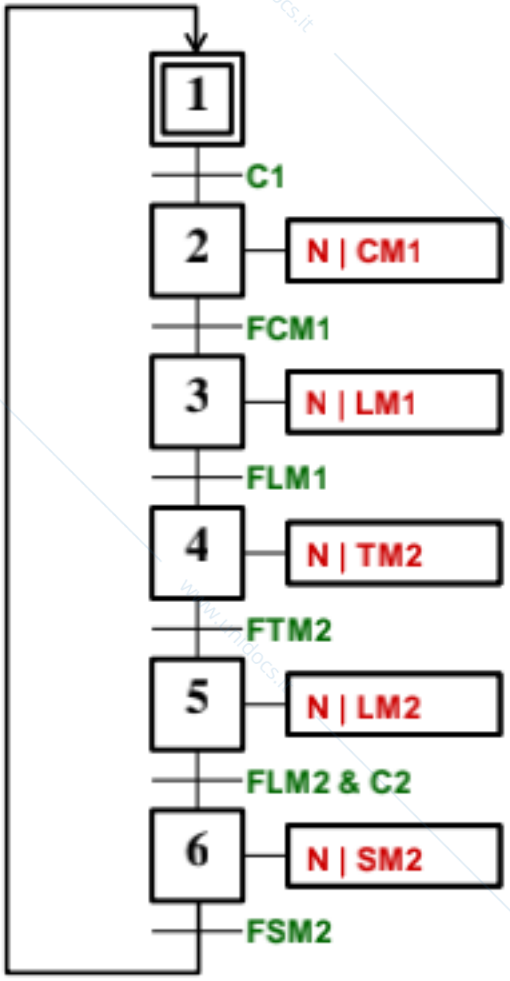


Comandi:

- **CM1:** carica M1
- **SM2:** scarica M2
- **TM2:** trasferisci da M1 a M2
- **LM1:** lavorazione M1
- **LM2:** lavorazione M2

Sensori (misure):

- **C1:** C1 presente
- **C2:** C2 presente
- **FCM1:** fine caricamento M1
- **FLM1:** fine lavorazione M1
- **FTM2:** fine trasferimento da M1 a M2
- **FLM2:** fine lavorazione M2
- **FSM2:** fine scaricamento M2

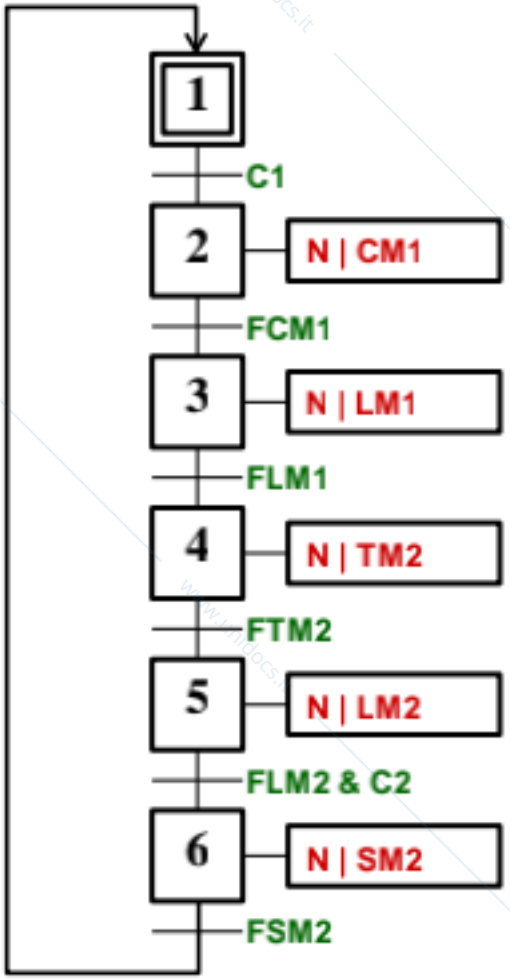


Stazione di lavoro (3)

Siamo sicuri di questa soluzione?

È tutta una questione di interpretazione del testo!

Interpretazione SEQUENZIALE



INTERPRETAZIONE «A RISORSE»

L'interpretazione a risorse permette i parallelismi necessari ad ottimizzare l'impianto.

Di base, una risorsa può essere:

- Disponibile: generalmente, basta uno stato
- Occupata: generalmente, specificato in diversi stati

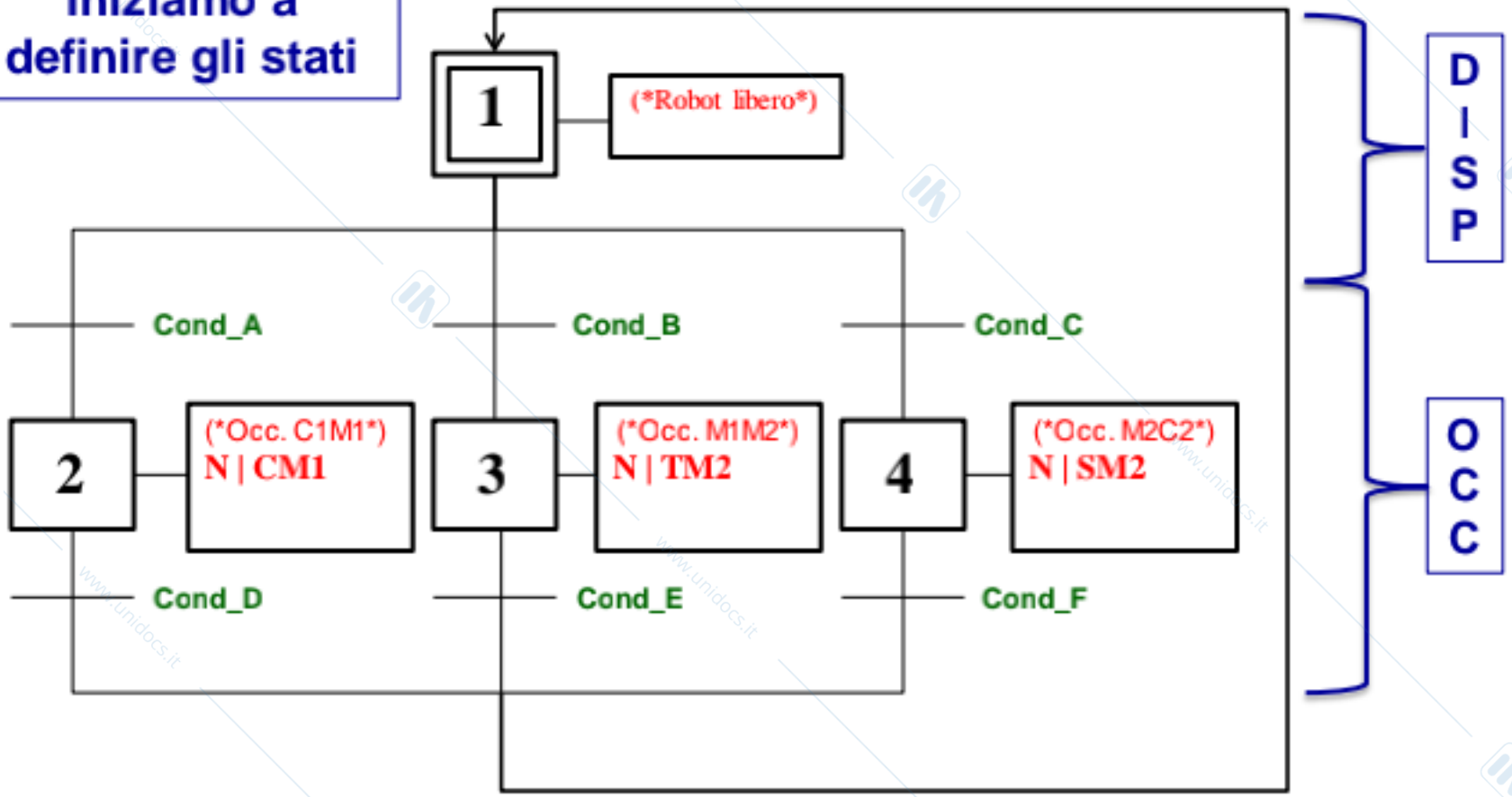
Risorse dell'impianto:

- Robot
- Macchina M1
- Macchina M2

Stazione di lavoro (4)

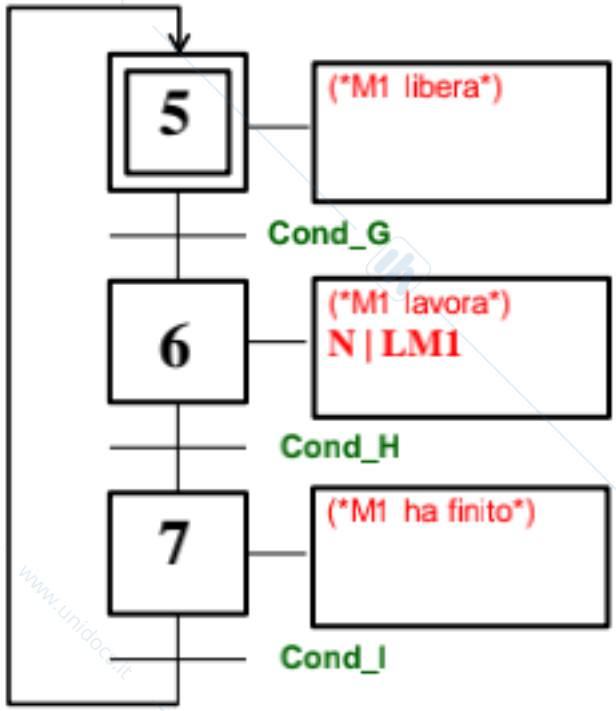
Robot R:

Iniziamo a definire gli stati

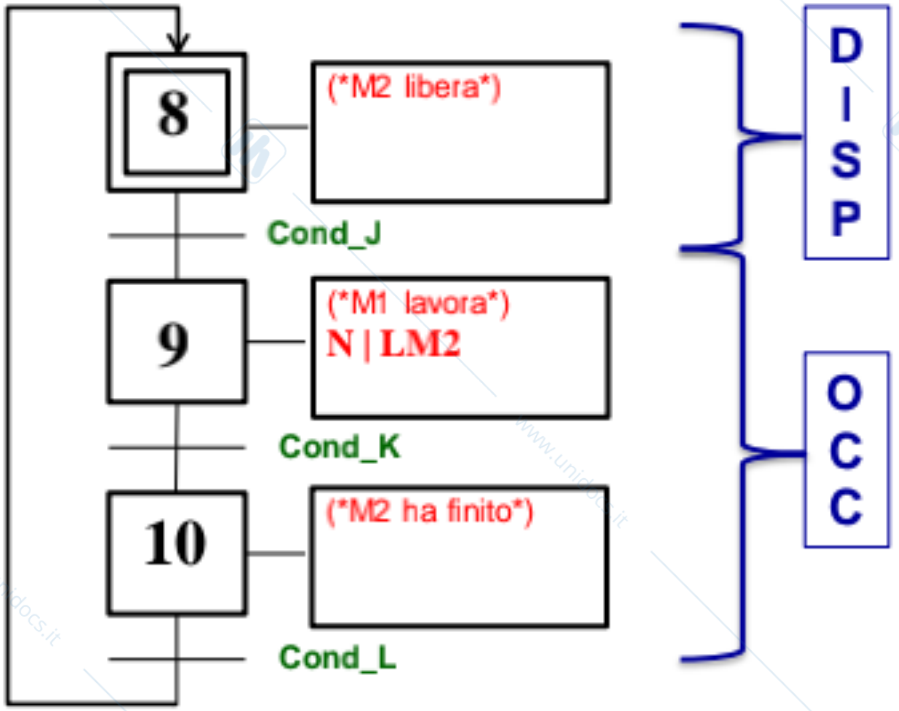


Stazione di lavoro (5)

Macchina M1:

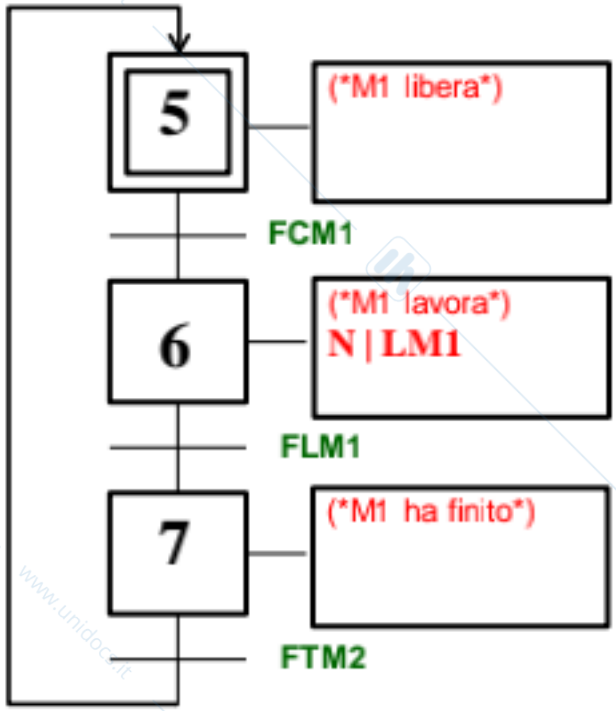


Macchina M2:

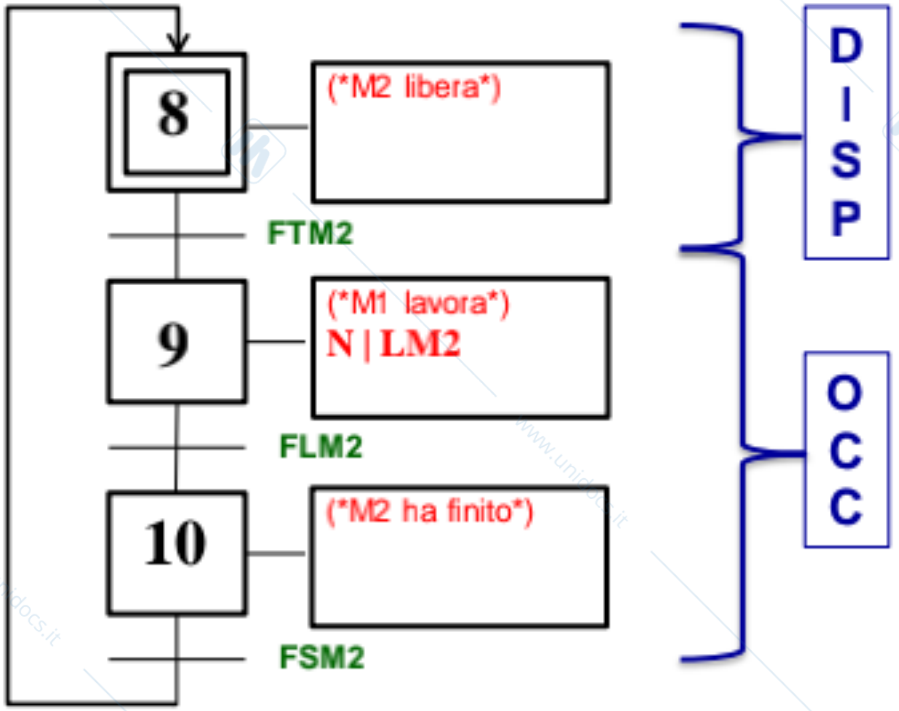


Stazione di lavoro (6)

Macchina M1:



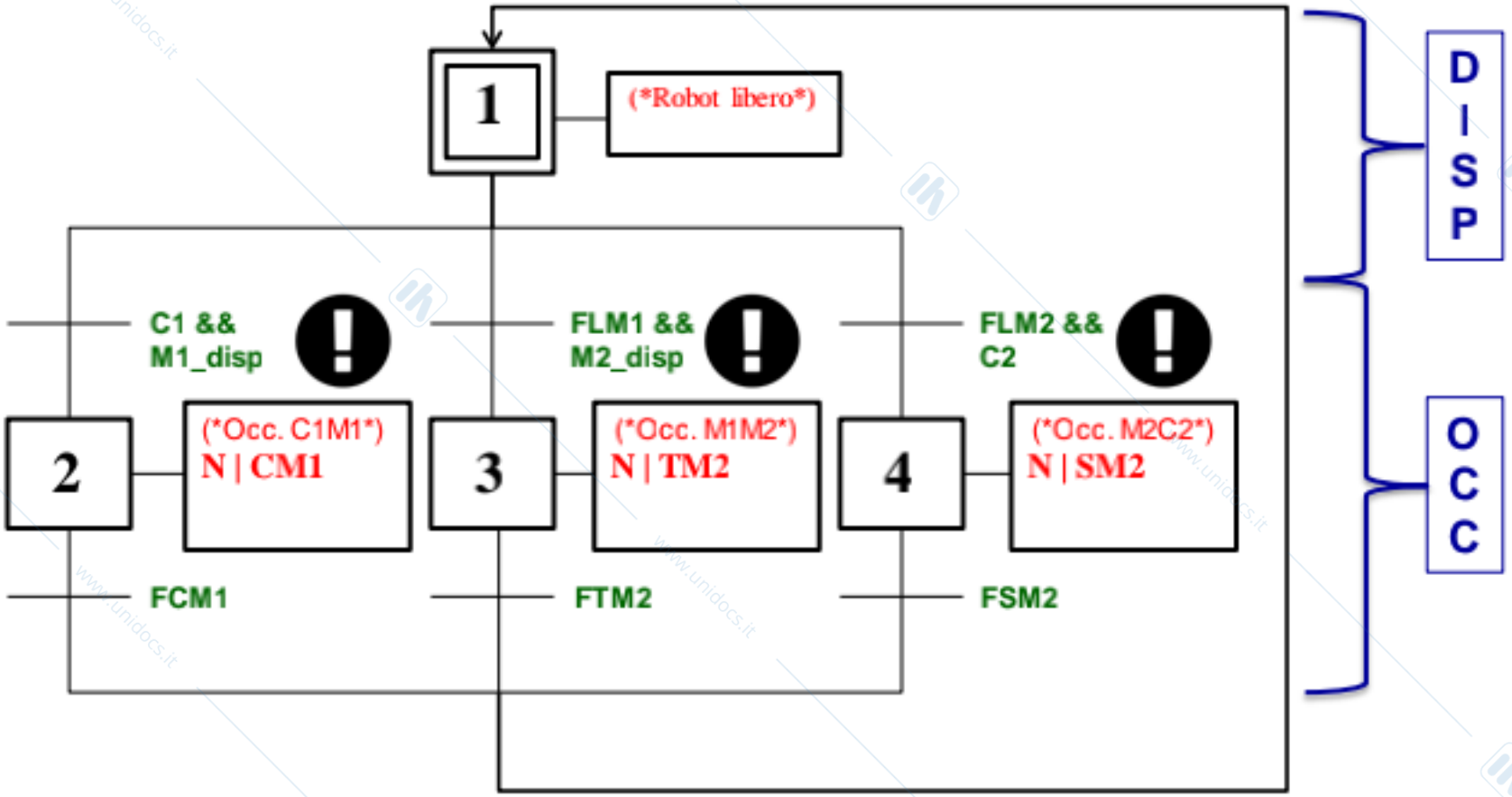
Macchina M2:



Ora definiamo le condizioni delle transizioni

Stazione di lavoro (7)

Robot R:



Stazione di lavoro (8)

Abbiamo 3 problemi diversi sulle transizioni evidenziate:

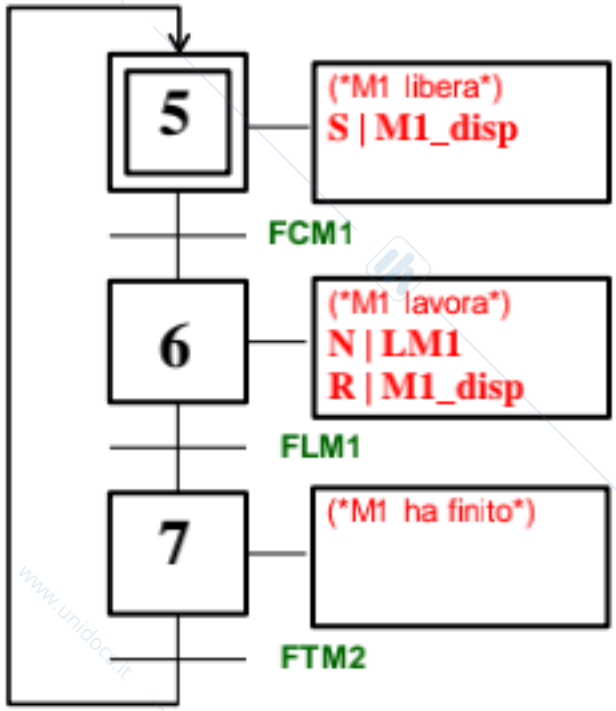
- 1) Non abbiamo dei sensori di M1_disp o M2_disp
- 2) …
- 3) …

1) Non abbiamo dei sensori di M1_disp o M2_disp

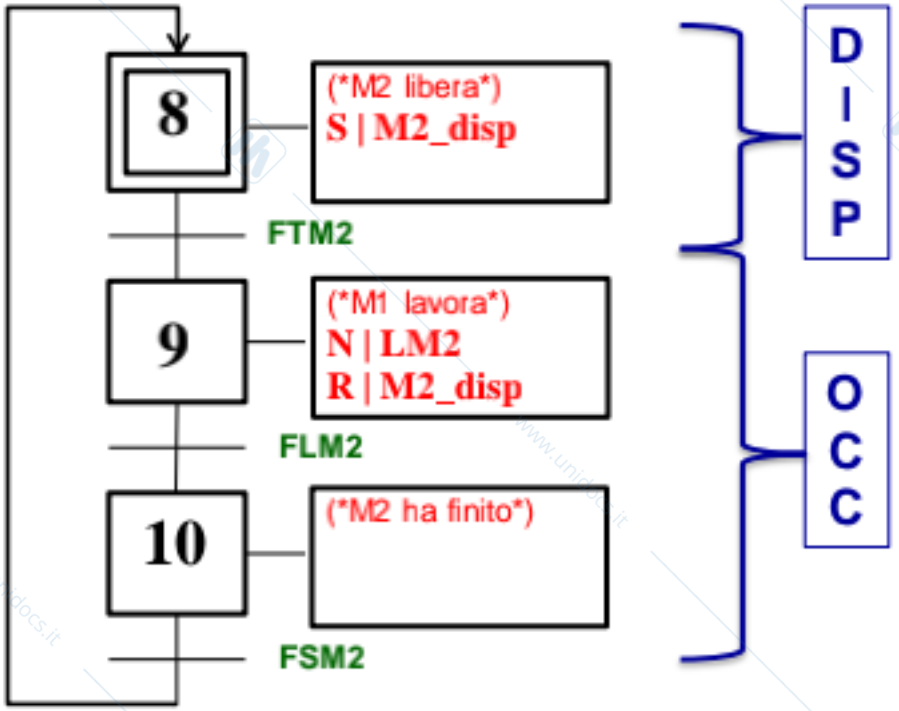
→ Definiamo delle **memorie** booleane chiamate «M1_disp» ed «M2_disp»: la loro gestione avverrà nei modelli di M1 ed M2.

Stazione di lavoro (9)

Macchina M1:



Macchina M2:



Stazione di lavoro (10)

Abbiamo 3 problemi diversi sulle transizioni evidenziate:

- 1) Non abbiamo dei sensori di M1_disp o M2_disp
- 2) E se FLM1 o FLM2 avvenissero mentre R è occupato?
- 3) …

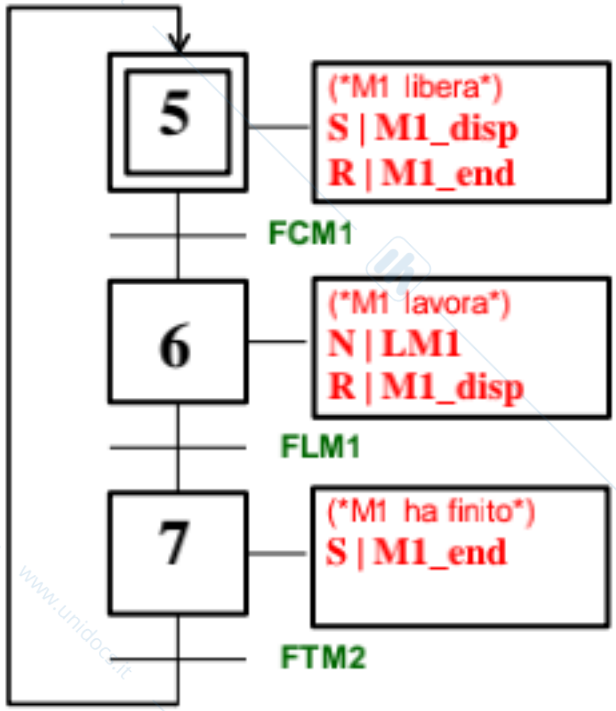
2) E se FLM1 o FLM2 avvenissero mentre R è occupato?

Se i segnali «FLM1» ed «FLM2» sono di tipo impulsivo bisogna accertarsi che tutti e 3 i sistemi (R, M1 ed M2) siano sempre pronti a riceverlo!

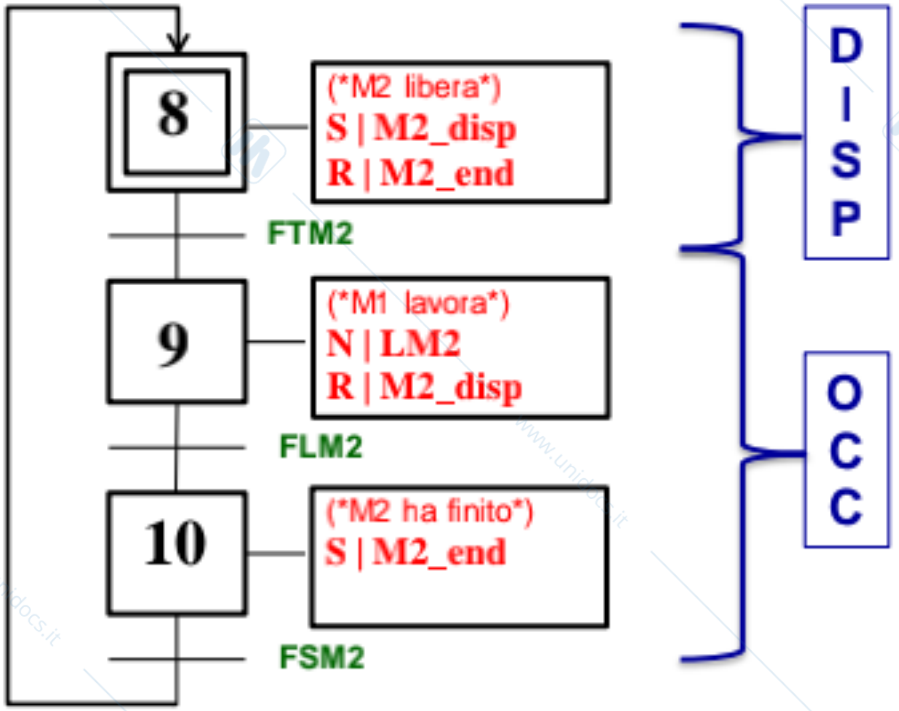
→ Definiamo altre **memorie** booleane chiamate «M1_end» ed «M2_end»: la loro gestione avverrà nei modelli di M1 ed M2.

Stazione di lavoro (11)

Macchina M1:

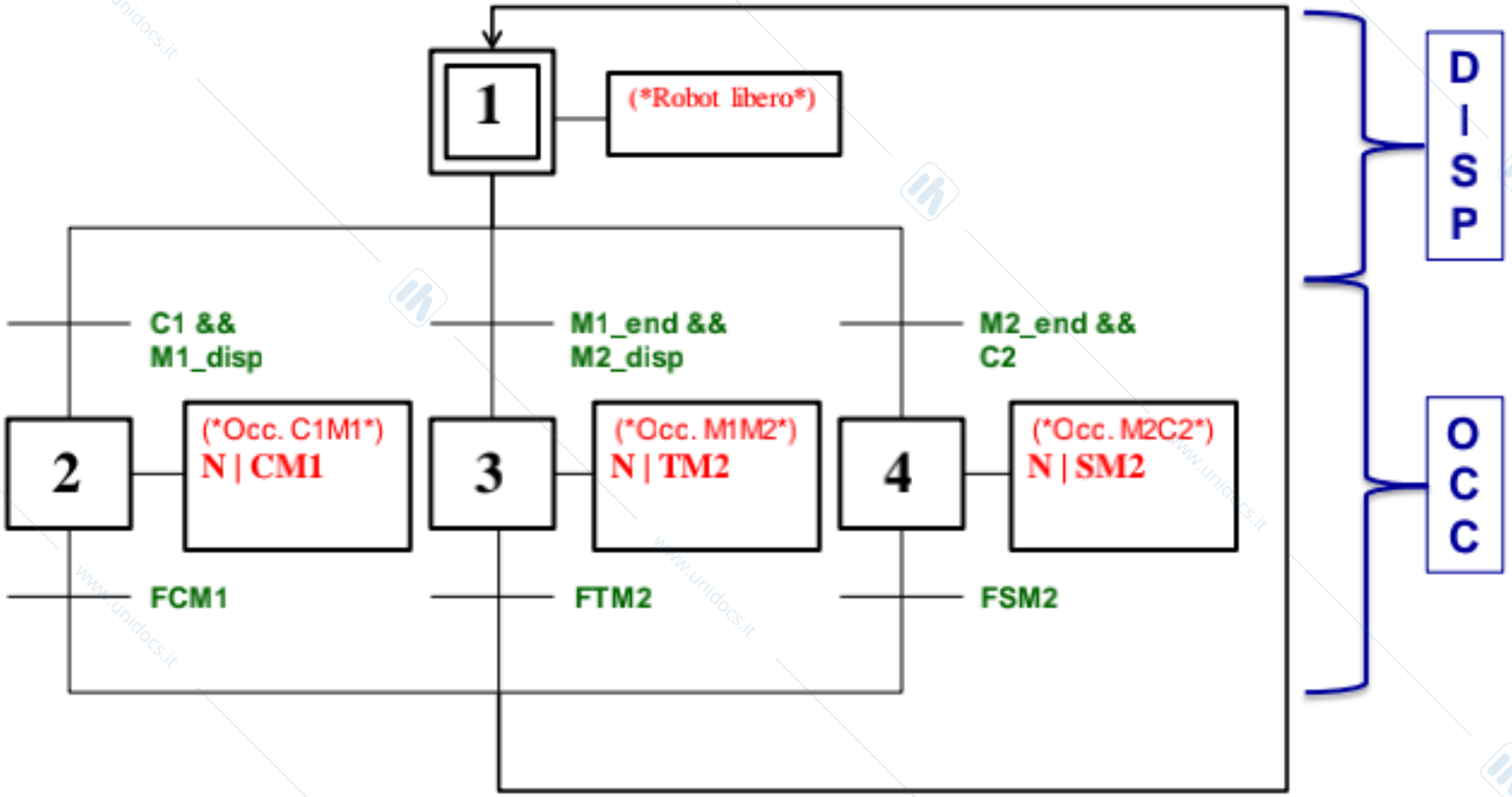


Macchina M2:



Stazione di lavoro (12)

Robot R:



Stazione di lavoro (13)

Abbiamo 3 problemi diversi sulle transizioni evidenziate:

1) Non abbiamo dei sensori di M1_disp o M2_disp

2) E se FLM1 o FLM2 avvenissero mentre R è occupato?

3) …

Soluzione alternativa ai primi 2 punti:

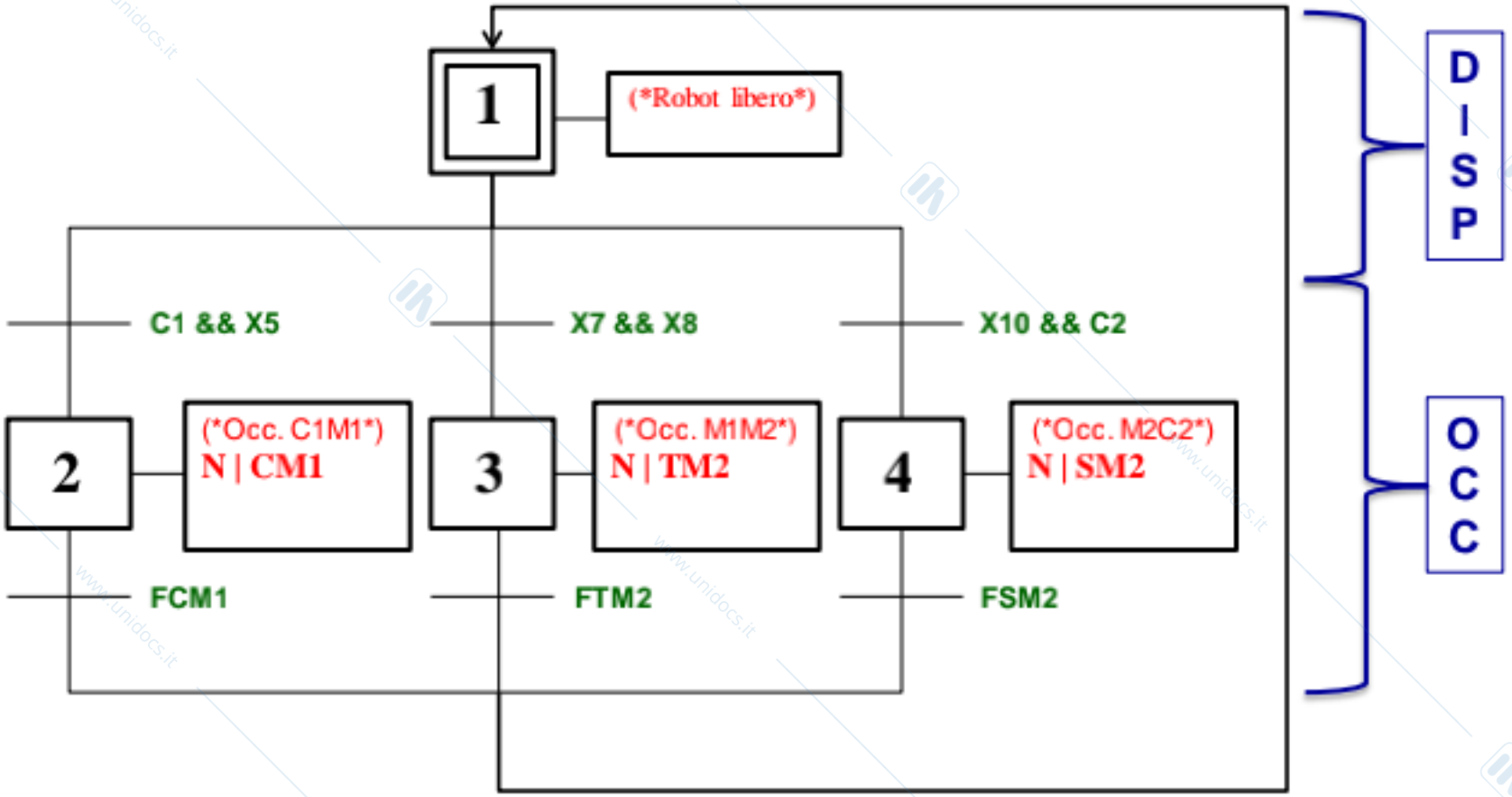
→ Definire le condizioni di transizione direttamente sugli stati!

PROS:
- Si evita di definire le memorie
- Le stesse memorie possono essere usate in diversi punti

CONS:
- Poco leggibile e poco modulare

Stazione di lavoro (14)

Robot R:



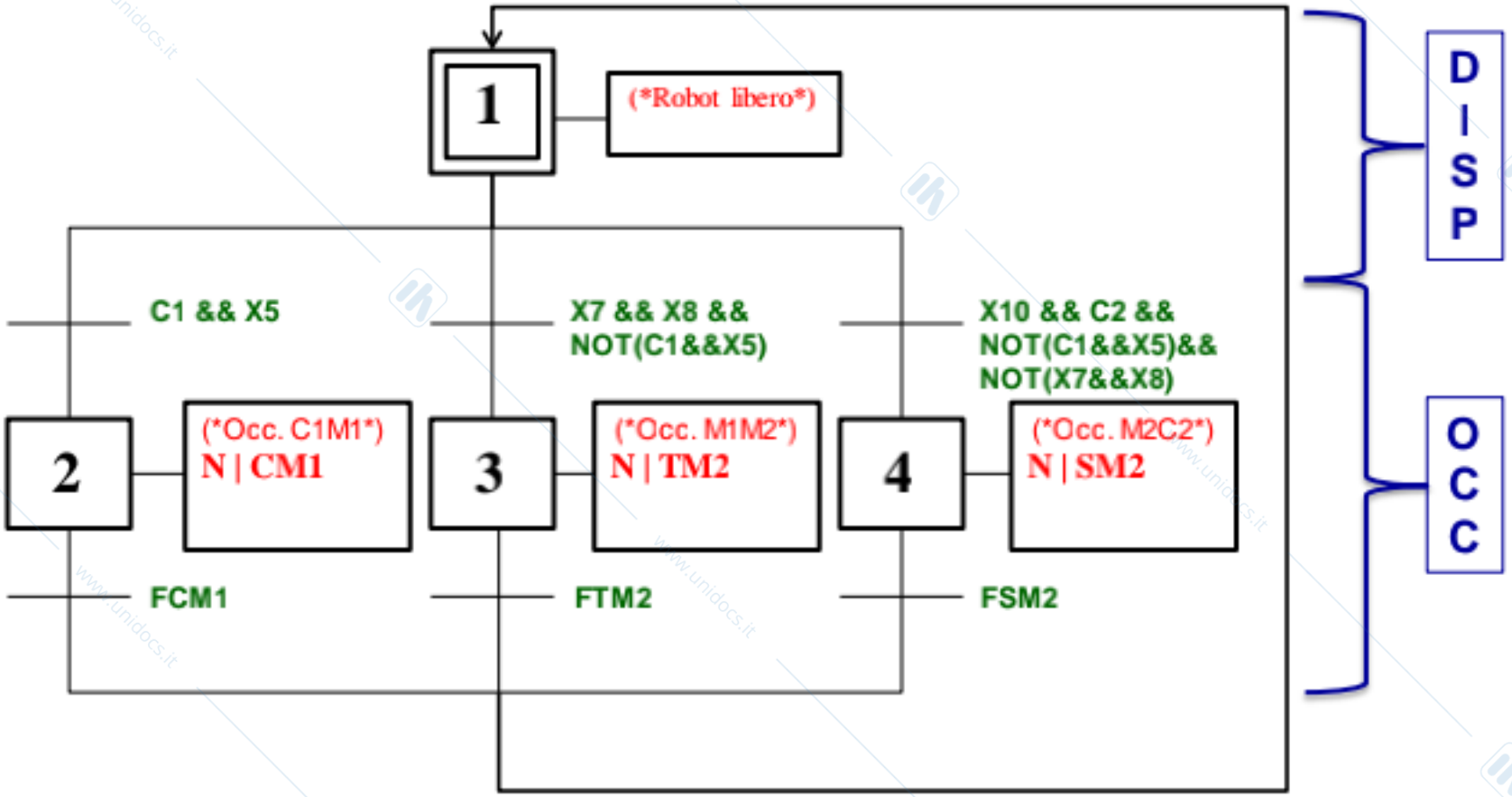
Stazione di lavoro (15)

Abbiamo 3 problemi diversi sulle transizioni evidenziate:

- 1) Non abbiamo dei sensori di M1_disp o M2_disp
- 2) E se FLM1 o FLM2 avvenissero mentre R è occupato?
- 3) OR-divergenza → condizioni mutualmente esclusive!

Stazione di lavoro (16)

Robot R:



Stazione di lavoro (17)

Commenti sull'interpretazione a risorse:

- Modularità del modello: facilità nel passare da 1 ad *n* macchine.
- Facilità di rappresentazione: pensando alle macchine in termini di disponibile / occupata non bisogna pensare a tutti i parallelismi che possono venire fuori (MA BISOGNA CONTROLLARLI!).
- Facilità di modifica: per rappresentare un'altra lavorazione, basta aggiungere i relativi stati «occupati» per ogni risorsa coinvolta.