



POLITECNICO DI MILANO

V Facoltà di Ingegneria



 POLITECNICO DI MILANO
DEI

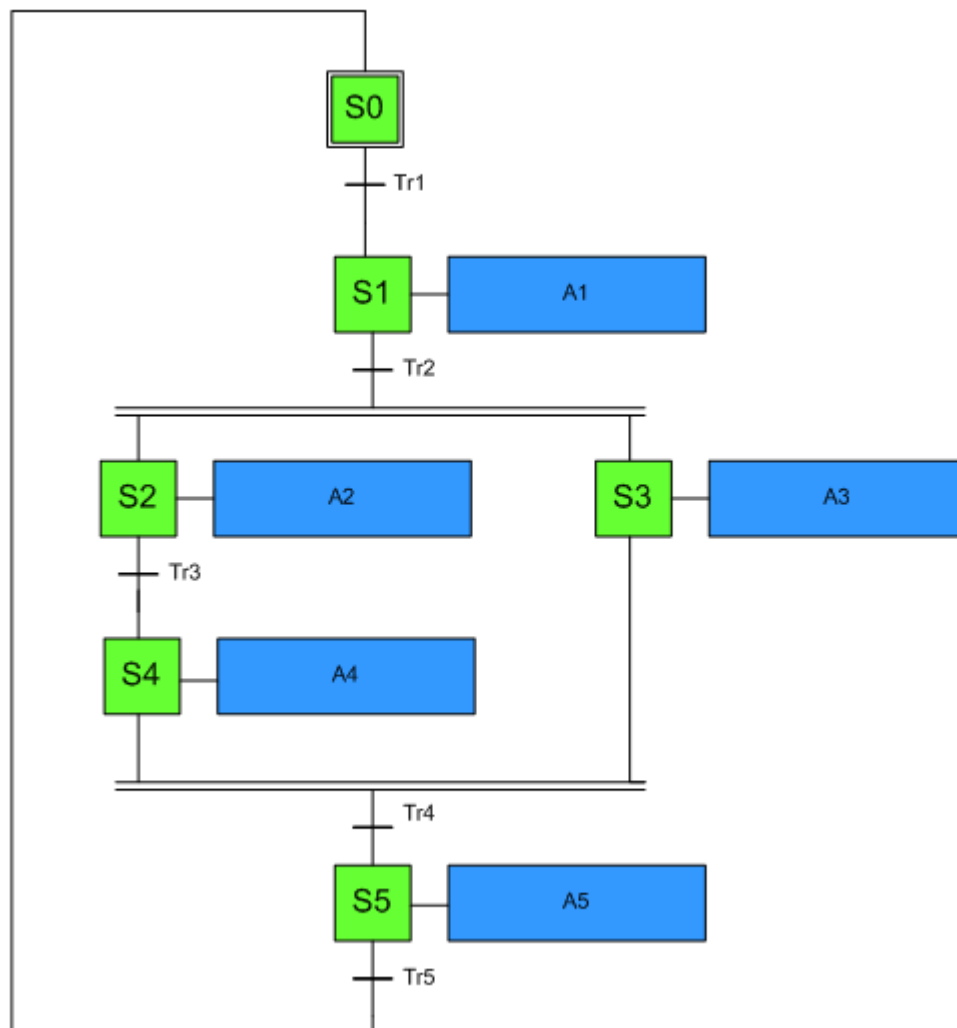
SFC

Sequential Function Chart

Prof. Luca Ferrarini



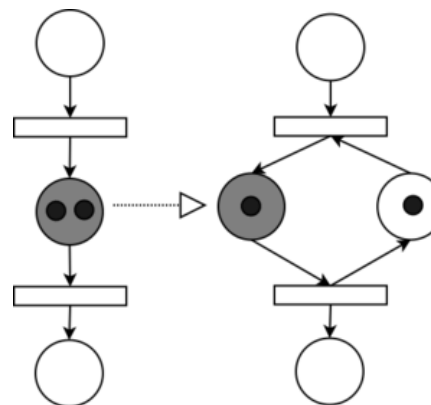
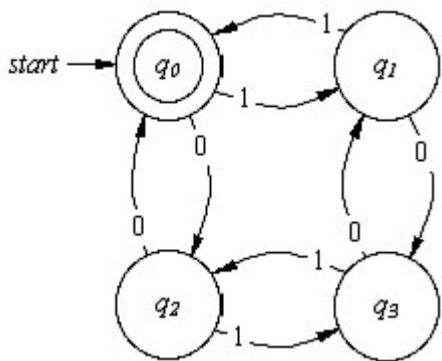
- Introduzione
- Elementi Base
- Operatori Base
- Regole di Evoluzione
- LD ed SFC





✓ Il Grafcet nasce dalle seguenti esigenze:

- Usare modelli formali per rappresentare funzioni di controllo logico per automazione industriale
- Standardizzare formalismi rappresentativi
- Arricchire gli automi (non hanno parallelismo)
- Semplificare le reti di Petri e renderle adatte ad applicazioni industriali
- Essere facilmente interpretabile dai dispositivi di controllo





- ✓ **Nel 1975 si costituisce una commissione in Francia per definire un nuovo formalismo**
 - Risultato: GRAFCET da “GRaph” (“grafo”) + AFCET (Association française de cybernétique économique et technique, che ha supportato i lavori)

- ✓ **Nel 1988 viene definito formalmente nello standard IEC (International Electrotechnical Commission)**
 - «Preparation of function charts for control systems», con il nome di «Sequential Function Chart»

- ✓ **Nel 1993 incluso nello standard IEC 1131-3 sulla programmazione dei controllori logici programmabili**



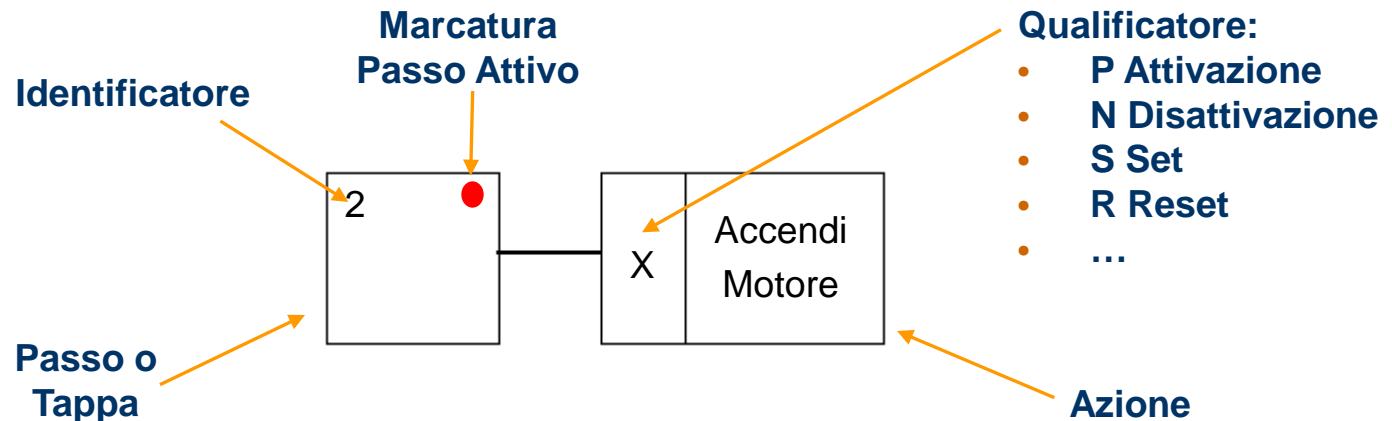
✓ Pochi e semplici concetti:

- **Passo**: stato di esecuzione della sequenza
- **Transizione**: passaggio tra un passo e l'altro durante l'esecuzione
- **Azione**: ogni transizione ha associata una o più azioni
- **Condizione logiche**: controllano lo scatto delle transizioni e quindi l'evoluzione dell'esecuzione (Vero = scatto, Falso = fermo)
- **Arco orientato**: identifica il passaggio tra un passo e l'altro tramite una transizione controllata da una condizione logica



✓ Passo (o tappa):

- Identificatore (solitamente un numero intero sequenziale)
- Può essere *attivo* o *inattivo*
- Rappresentazione grafica: quadrato (con bordo doppio se iniziale)
- Può avere una serie di “azioni” associate che vengono eseguite quando il passo è attivo, a ciascuna azione può quindi essere associato un qualificatore che specifica il momento o il periodo di esecuzione (N, S, P, ecc.)





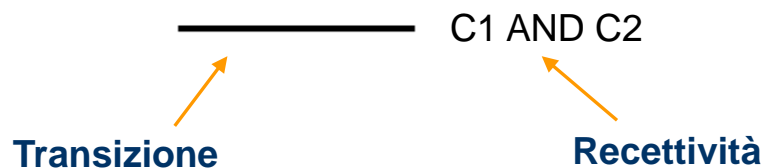
✓ Qualificatori:

N	Non-Stored	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
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
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
P	Pulse	L'azione comincia quando il passo diventa attivo/disattivo 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



✓ Transizione

- Identificatore
- Abilitata o disabilitata
- Può scattare
- Rappresentazione grafica: segmento orizzontale
- Associato a “recettività” (condizione logica su ingressi)



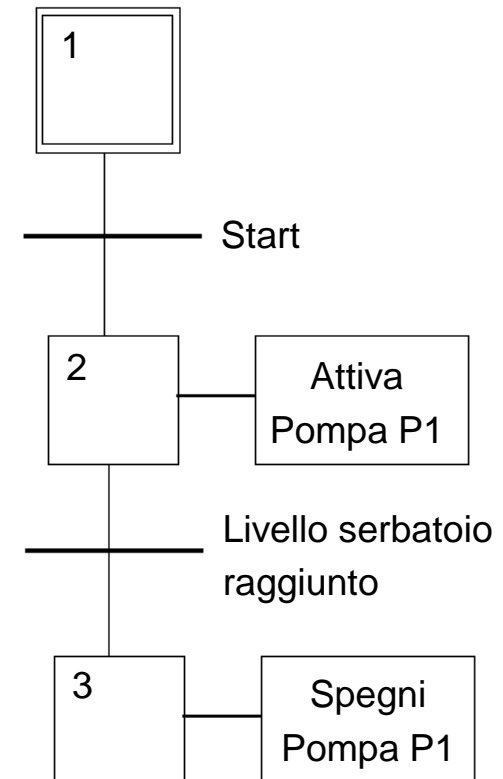


✓ Regola di connessione:

- Un passo può essere direttamente collegato solo con transizioni, mai con passi; analogamente, una transizione può essere direttamente collegata solo con passi, mai con transizioni

✓ Convenzione grafica:

- Archi con angoli retti
- Archi verso il basso: senza freccia
- Archi verso l'alto: con freccia





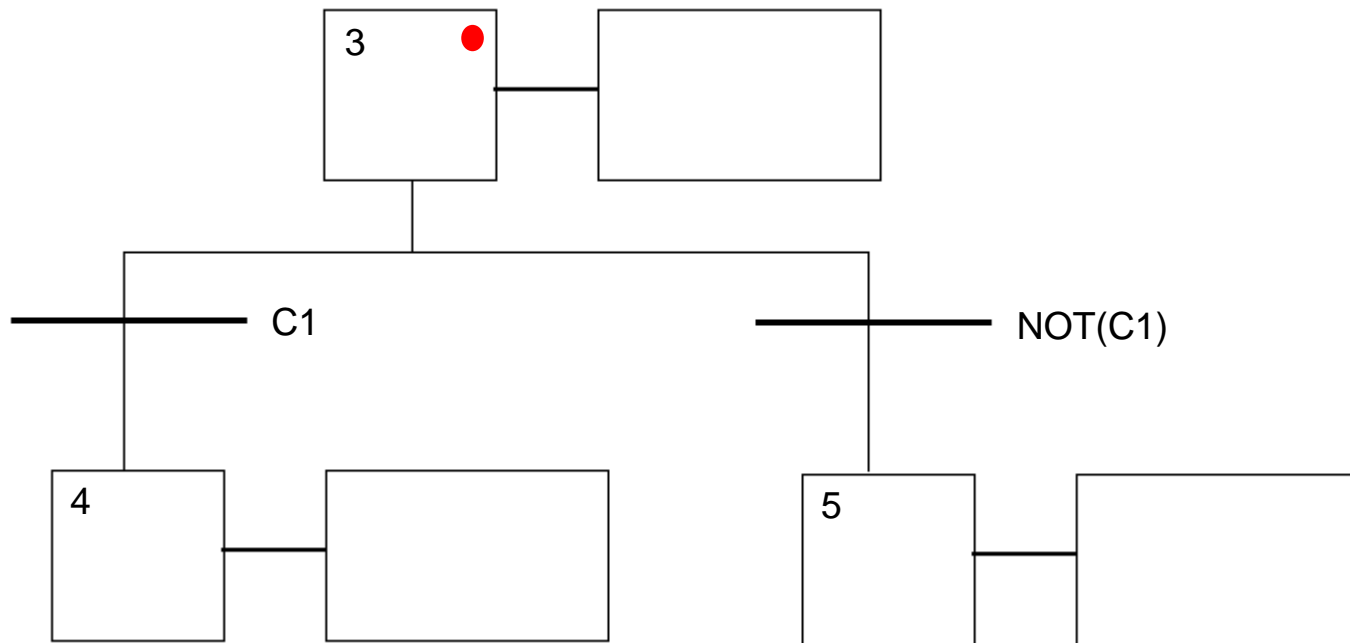
- ✓ **In Grafcet è possibile rappresentare due caratteristiche basilari per sistemi ad eventi:**
 - Scelta (operatori di tipo **OR**)
 - Parallelismo (operatori di tipo **AND**)

- ✓ **A tali caratteristiche corrispondono operatori grafici e semantici diversi, caratterizzati da un “inizio” (**divergenza**) e una “fine” (**convergenza**)**



✓ Inizio della scelta

- Due transizioni a valle di un singolo passo

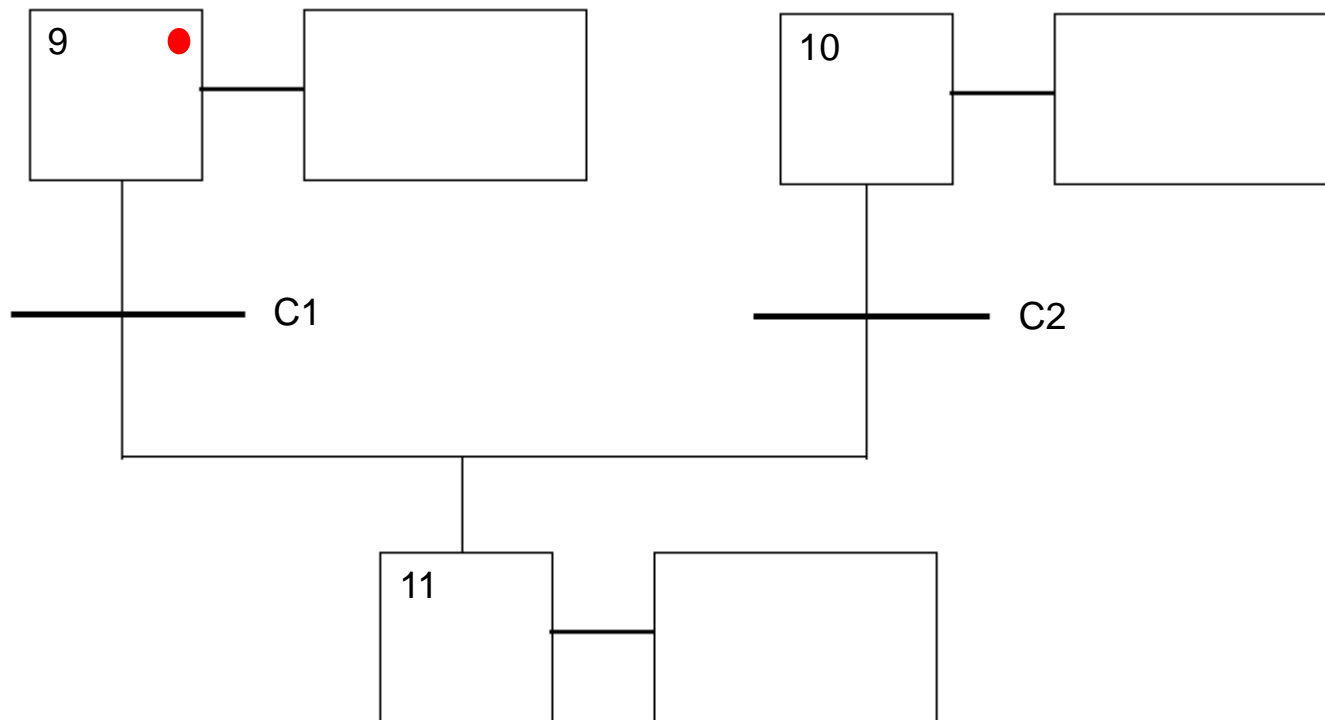


NB: e' consigliato (in alcuni ambienti di sviluppo obbligatorio) avere due **condizioni in mutua esclusione** tra loro per conoscere deterministicamente l'evoluzione della sequenza!!!!



✓ Fine della scelta

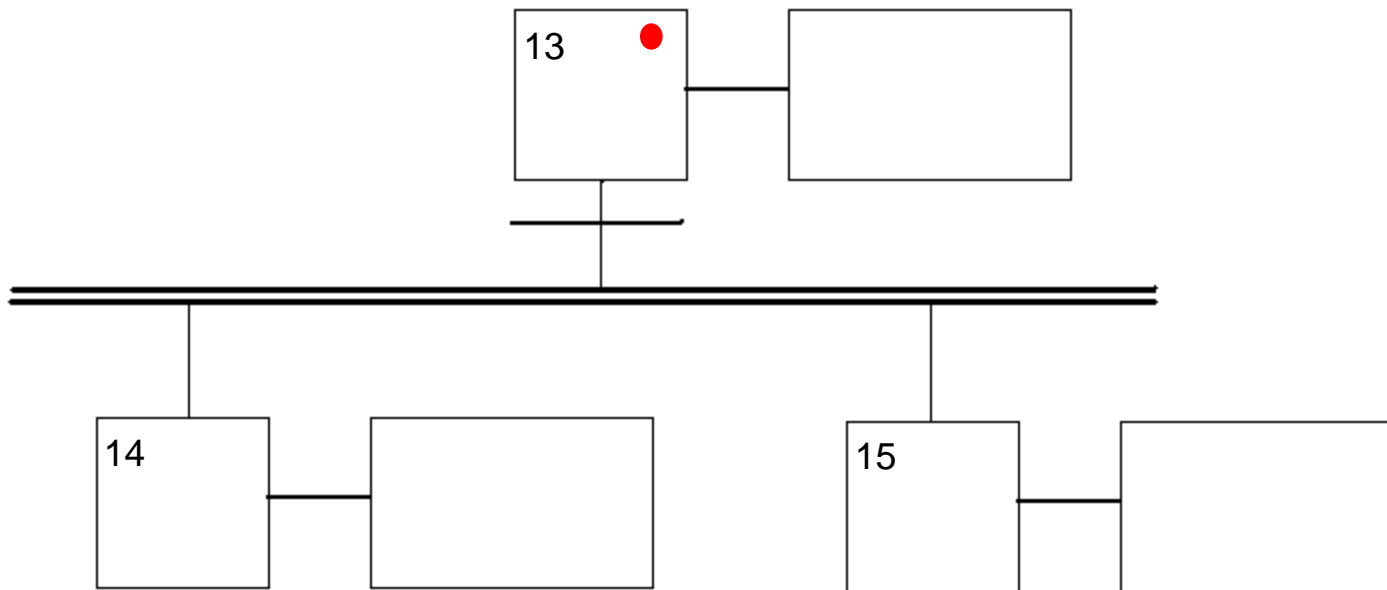
- Un passo con due transizioni a monte





✓ Inizio del parallelismo

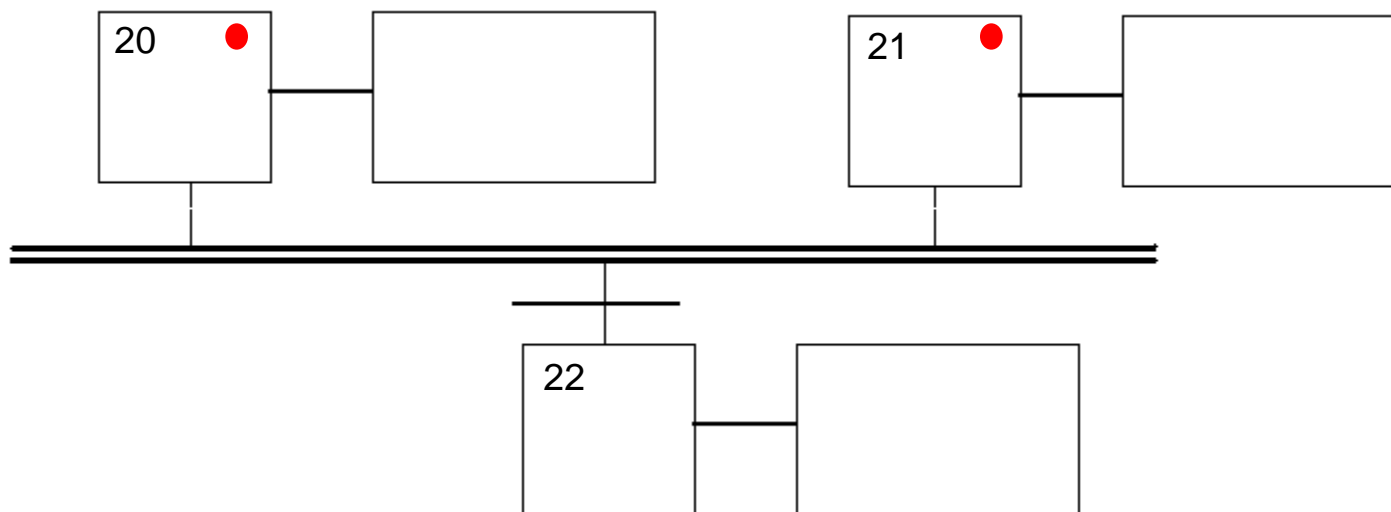
- Attivazione simultanea di più passi





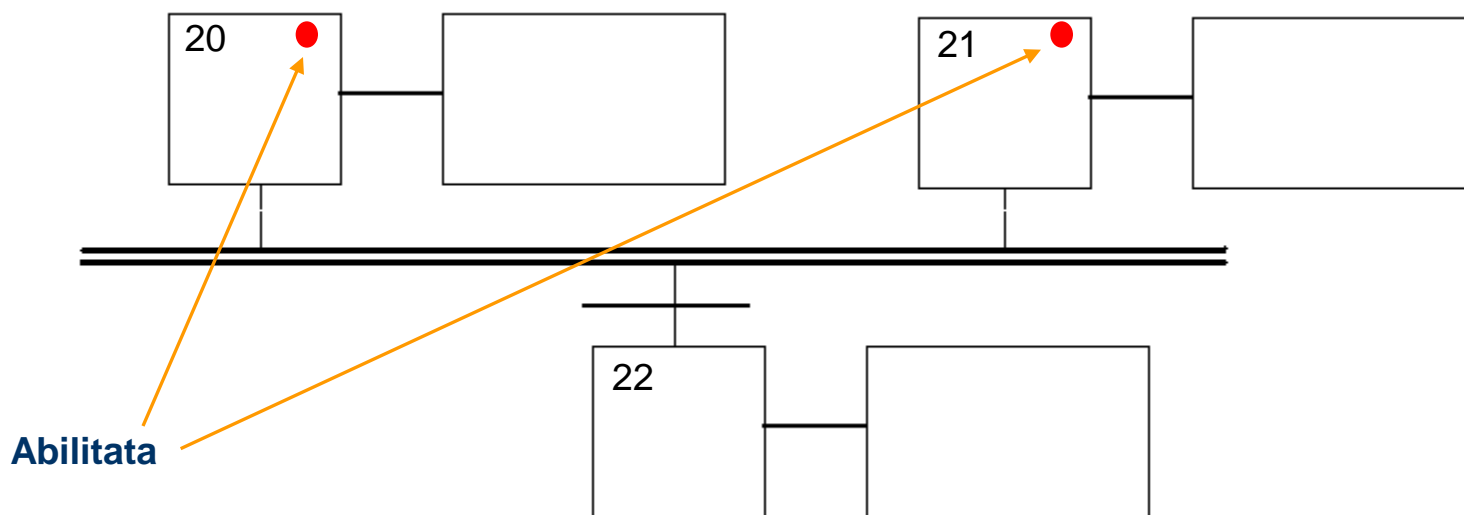
✓ Fine del parallelismo

- Disattivazione simultanea di più passi (sincronizzazione)





- ✓ Una transizione si dice **abilitata** se tutti i passi a monte sono attivi
- ✓ Una transizione si dice **superabile** se è abilitata e la sua recettività assume il valore logico vero (a volte si dice che la transizione può scattare)

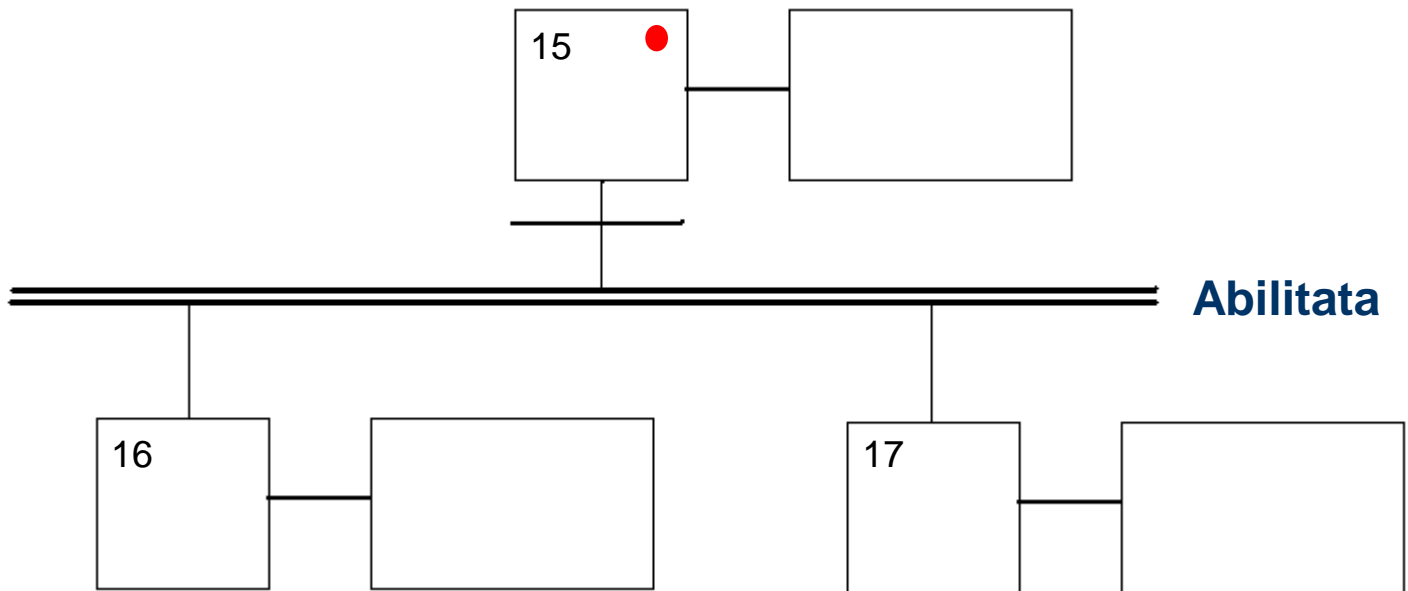




- ✓ *Regola 1* - **se una transizione è superabile, essa viene effettivamente superata**: tutti i passi a monte della transizione vengono disattivati, mentre tutti quelli a valle vengono abilitati
- ✓ *Regola 2* - tutte le transizioni superabili in un certo istante, vengono **superate contemporaneamente**

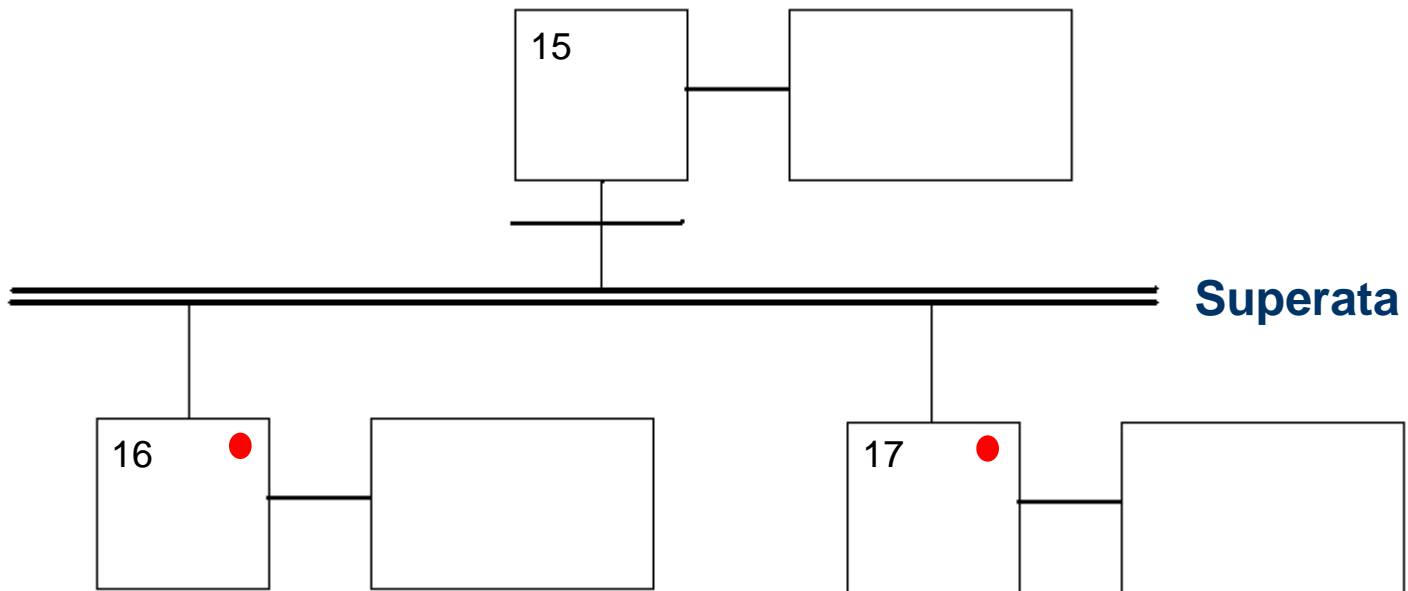


✓ Superamento di una transizione



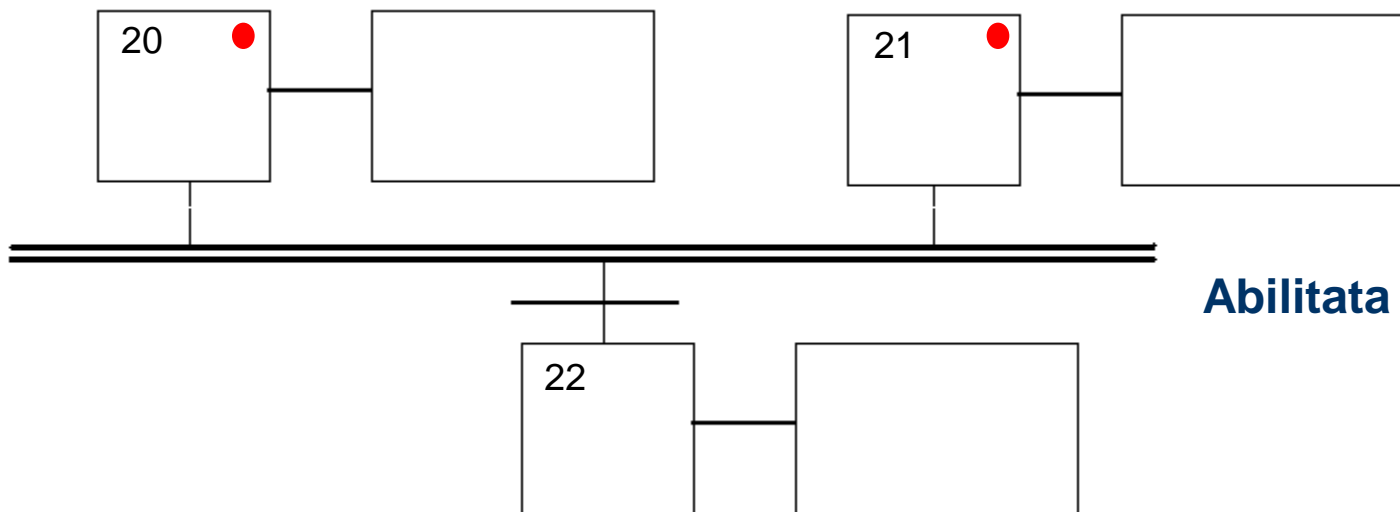


✓ Superamento di una transizione



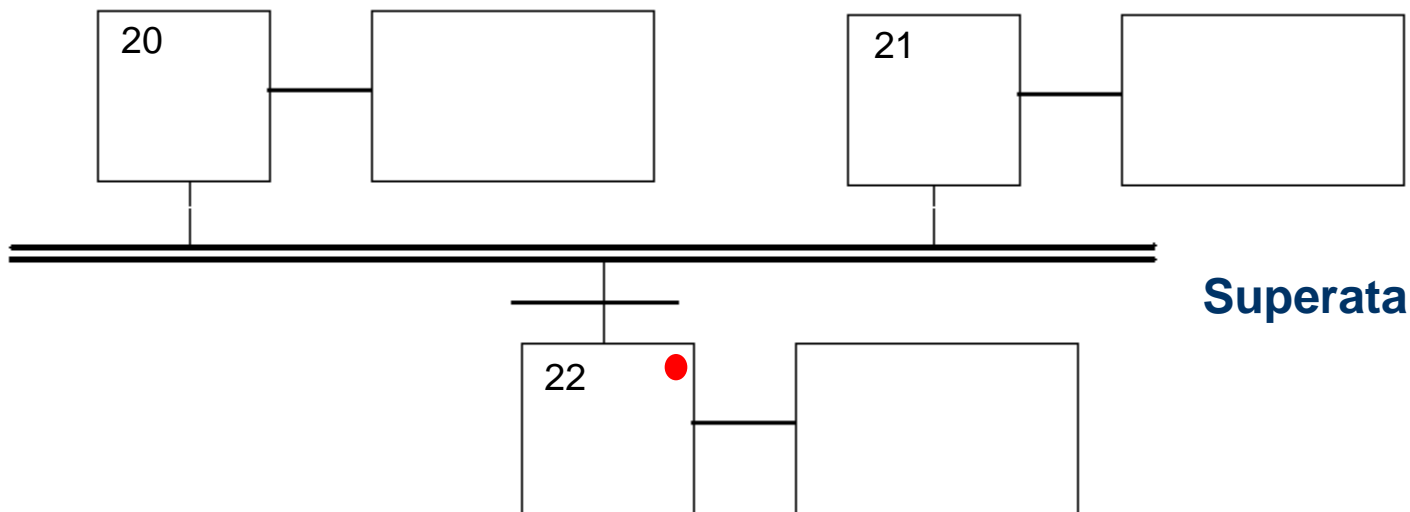


✓ Superamento di una transizione



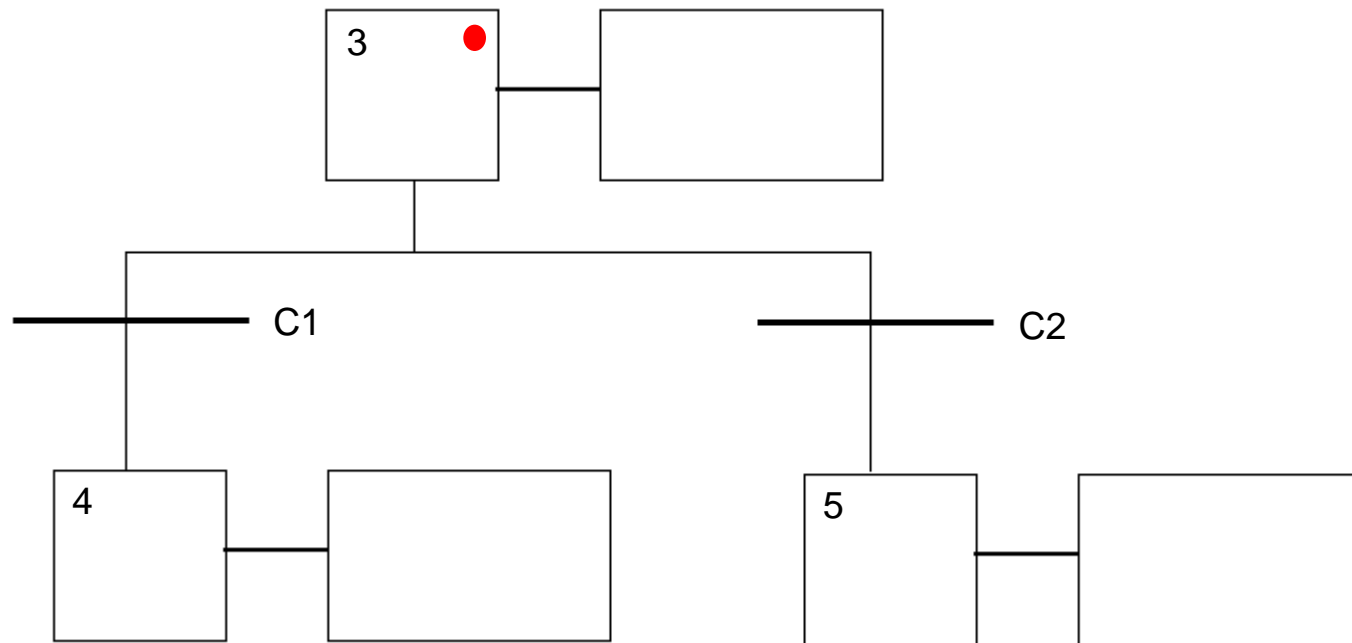


✓ Superamento di una transizione



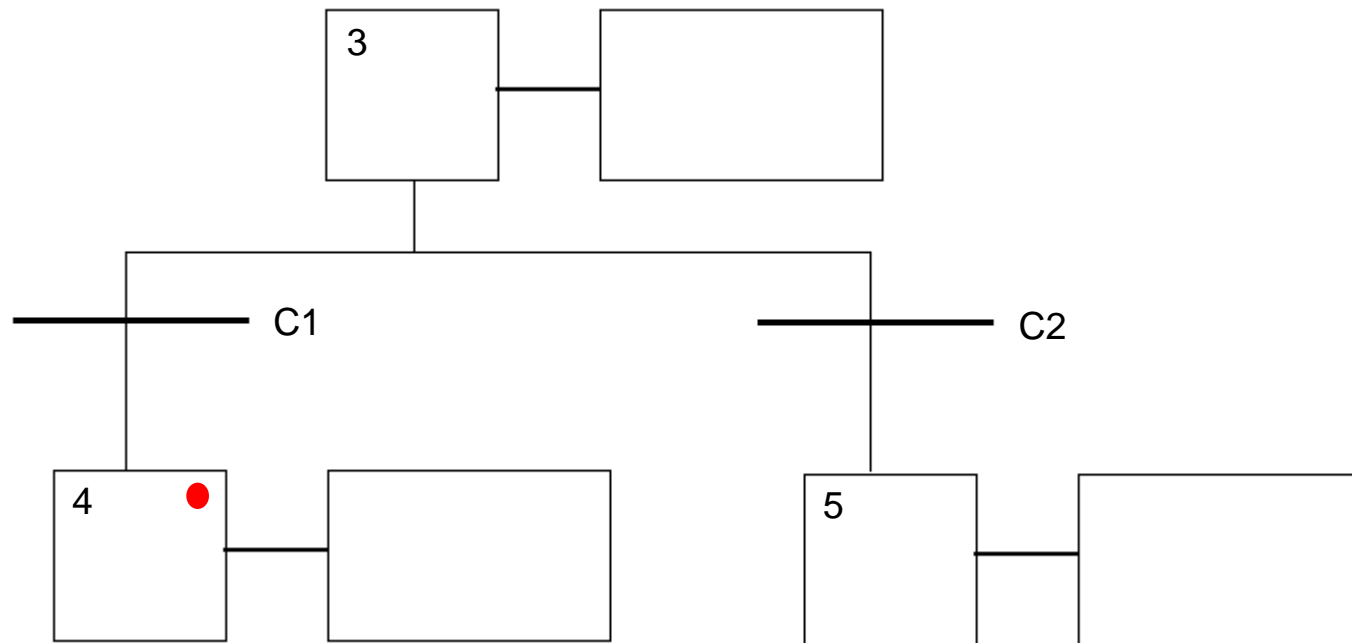
✓ Superamento di una transizione

- C1 vero, C2 falso



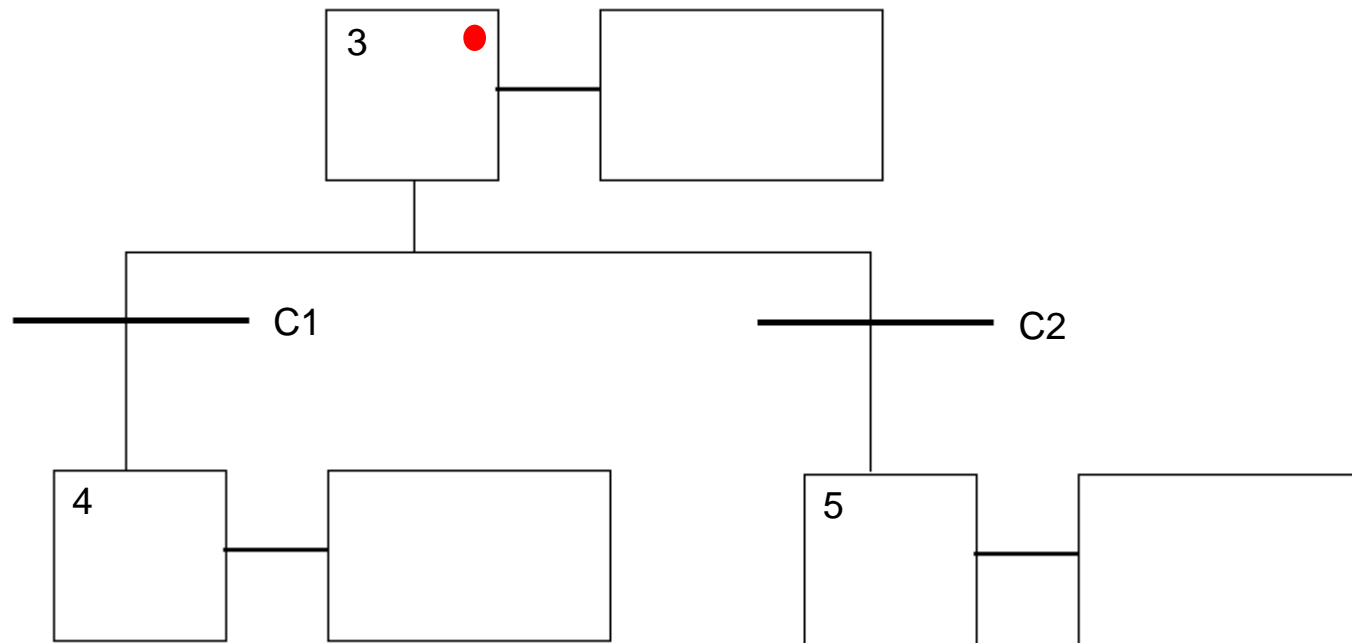
✓ Superamento di una transizione

- C1 vero, C2 falso



✓ Superamento di una transizione

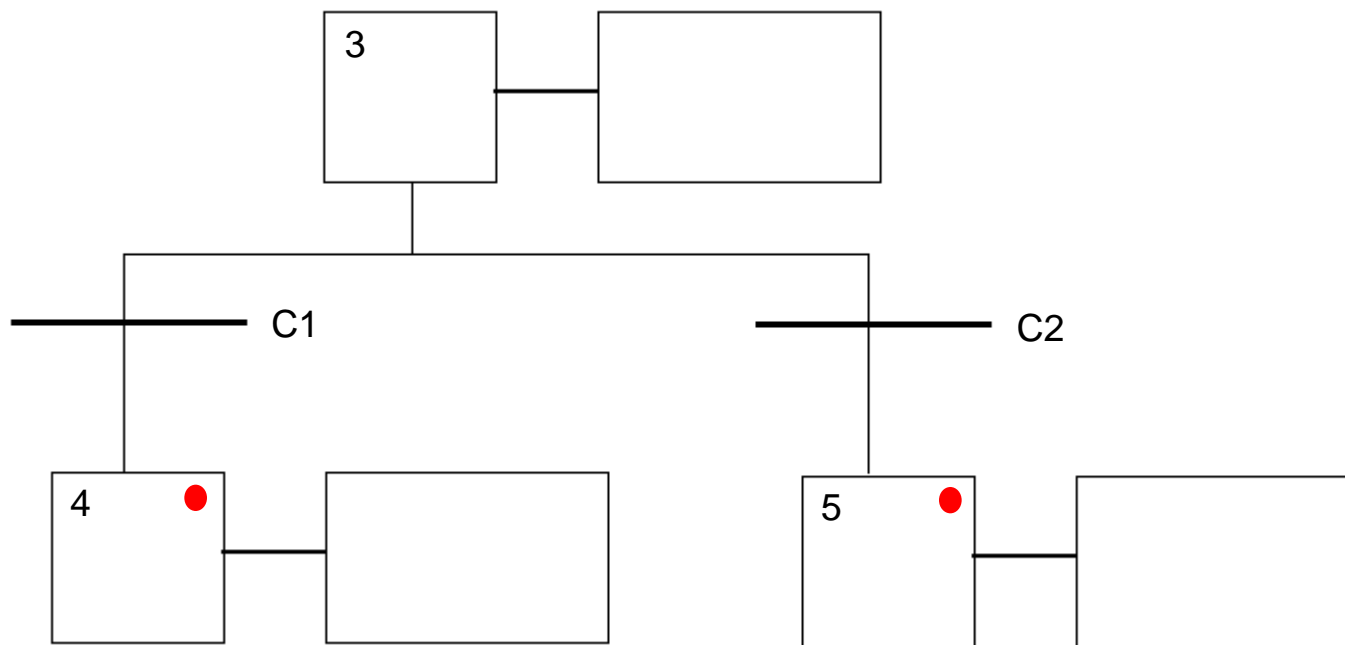
- C1 vero, C2 vero





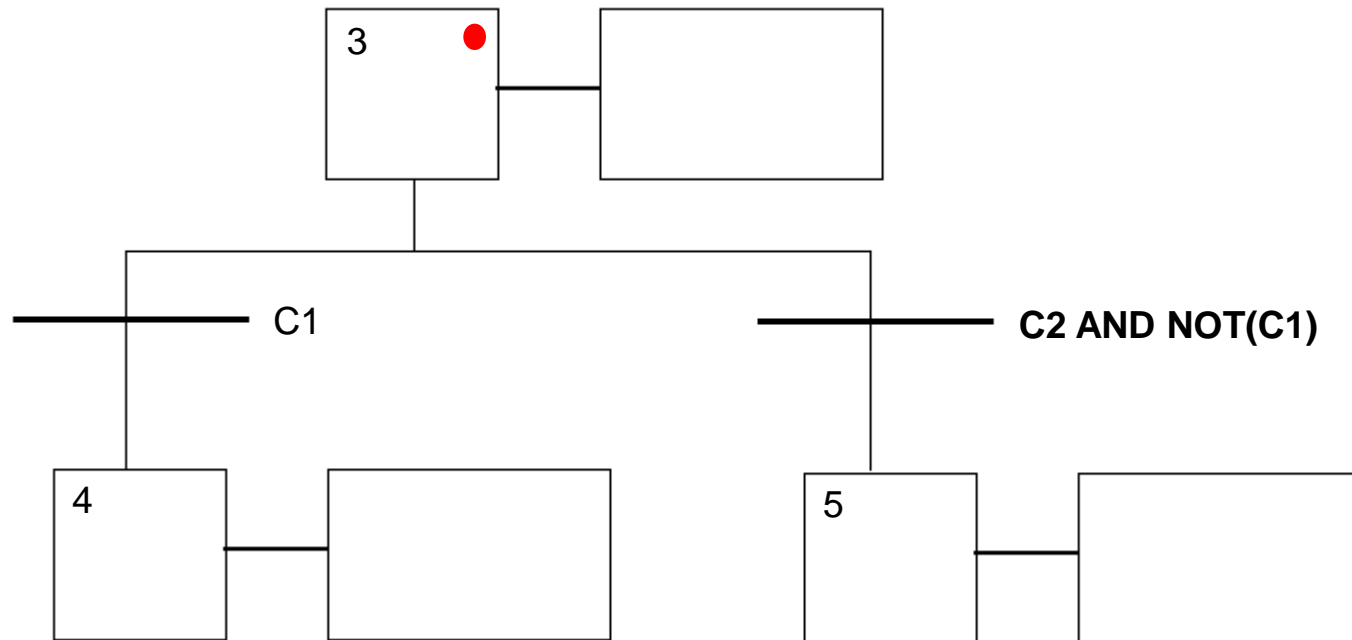
✓ Superamento di una transizione

- C1 vero, C2 vero



✓ Superamento di una transizione

- C1 vero, C2 vero → Situazione ambigua risolvibile con la mutua esclusione o con la priorità ad una delle due transizioni

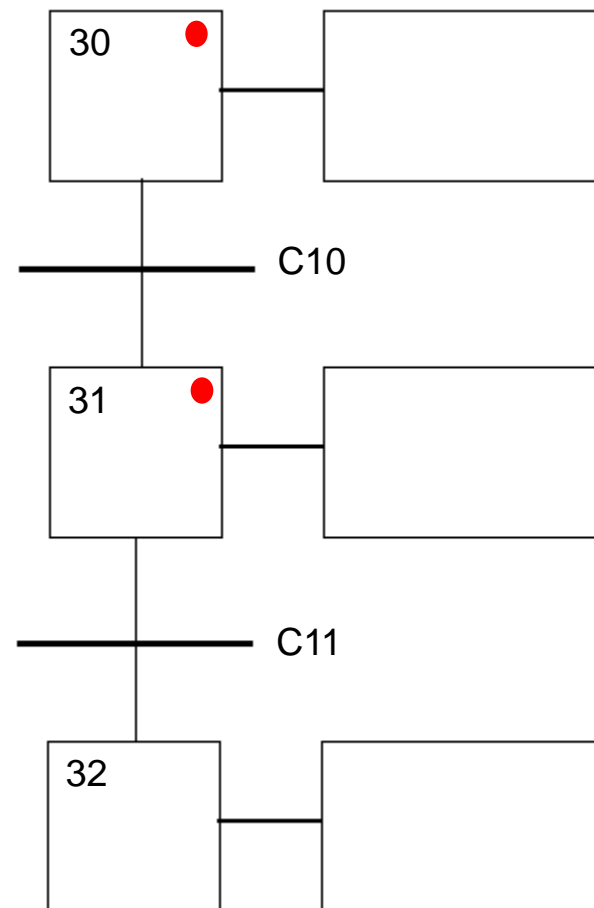




✓ Nota

- Nell'applicare le regole di evoluzione può accadere che un passo debba essere contemporaneamente disattivato e attivato
- Convenzionalmente, tale passo rimane attivo
- Tuttavia, tale situazione è da usare con cautela

C10 ed C11
entrambe vere:
che succede??

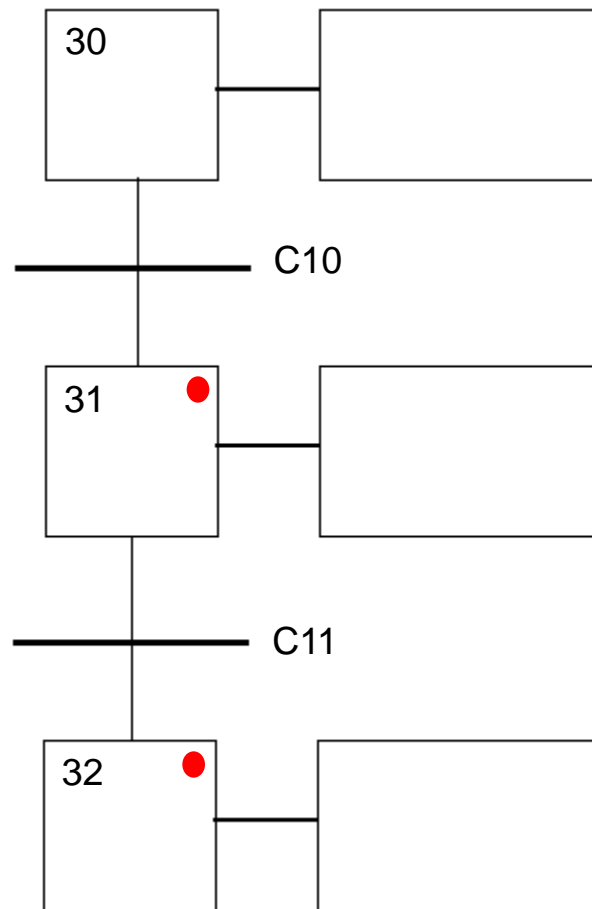




✓ Nota

- Nell'applicare le regole di evoluzione può accadere che un passo debba essere contemporaneamente disattivato e attivato
- Convenzionalmente, tale passo rimane attivo
- Tuttavia, tale situazione è da usare con cautela

C10 ed C11
entrambe vere:
che succede??





SFC

LD

Vantaggi

- Progettazione del controllo più agevole
- Permette anche una progettazione funzionale delle sole specifiche
- Il progetto è, per molti aspetti, anche documentazione
- Migliore la manutenzione di funzioni già sviluppate

Svantaggi

- Nonostante sia stato sviluppato più di 25 anni fa, e nonostante sia presente in normative industriali (IEC) da più di 10 anni, non è supportato da tutti i PLC

Vantaggi

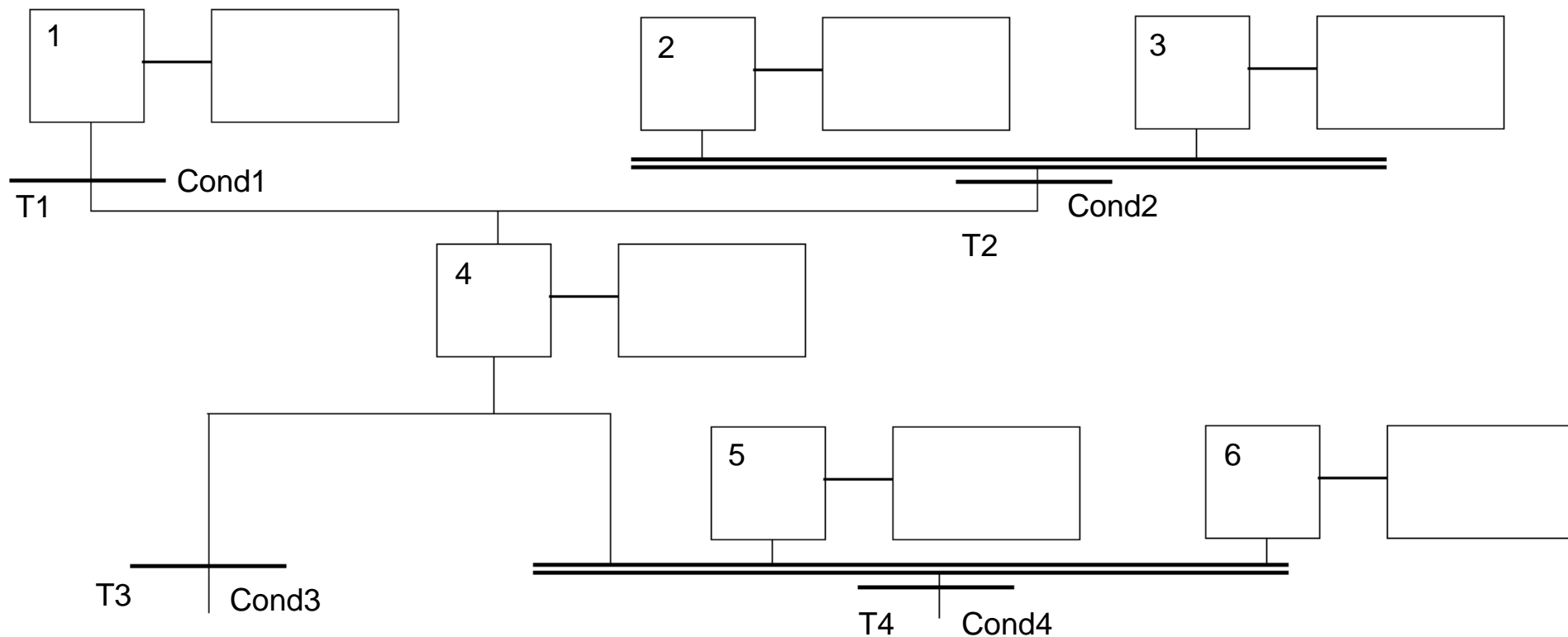
- Supportato da tutti i PLC

Svantaggi

- Molto di basso livello
- Poco leggibile e quindi manutenibile e riutilizzabile

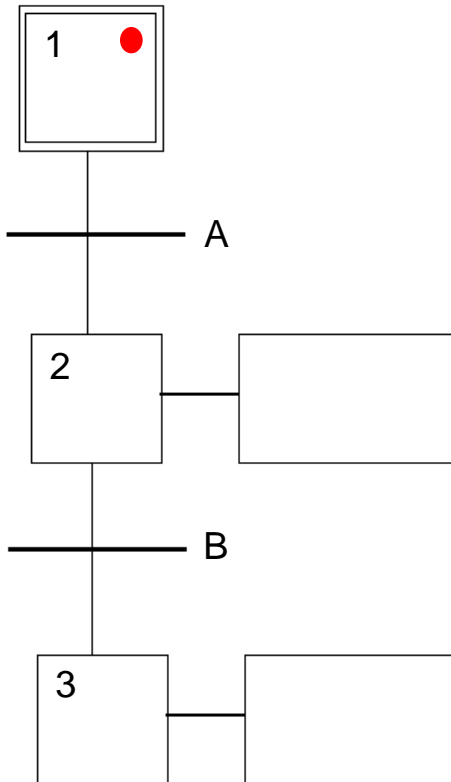


✓ Consideriamo il generico passo [4] di un SFC...



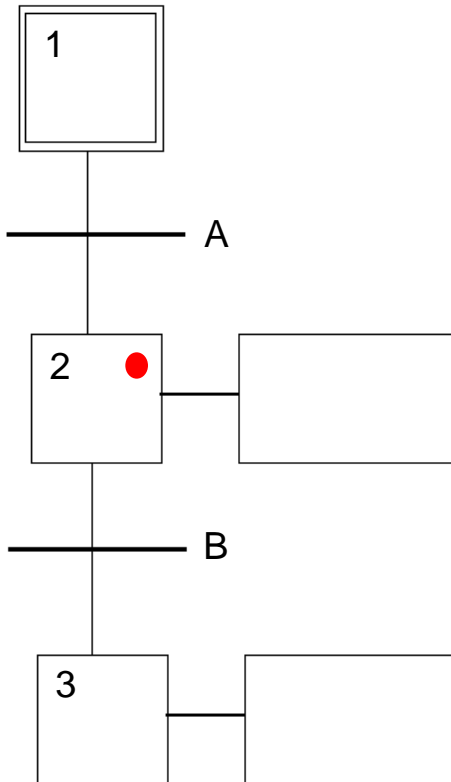


- ✓ La rappresentazione formale delle regole di evoluzione (o interpretazione) di uno schema SFC si chiama **algoritmo di evoluzione** (o di interpretazione)
- ✓ In letteratura esistono molte varianti possibili per gli algoritmi di evoluzione, proprio perché uno schema può essere interpretato in modi leggermente diversi



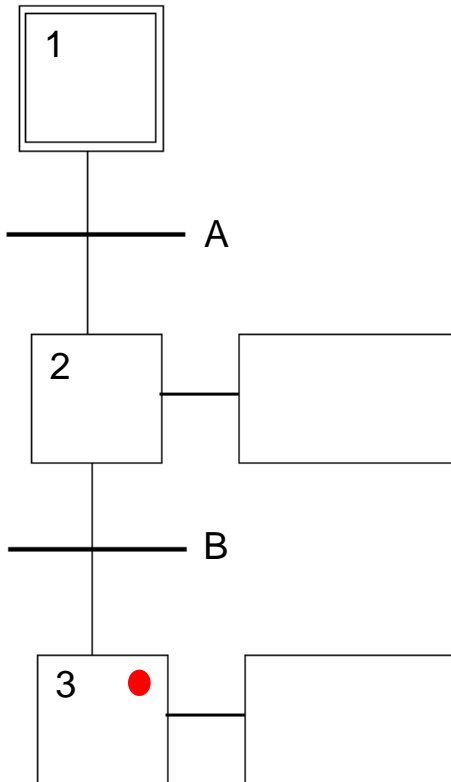
Ad esempio:

- ✓ Cosa succede se le condizioni A e B sono contemporaneamente verificate?



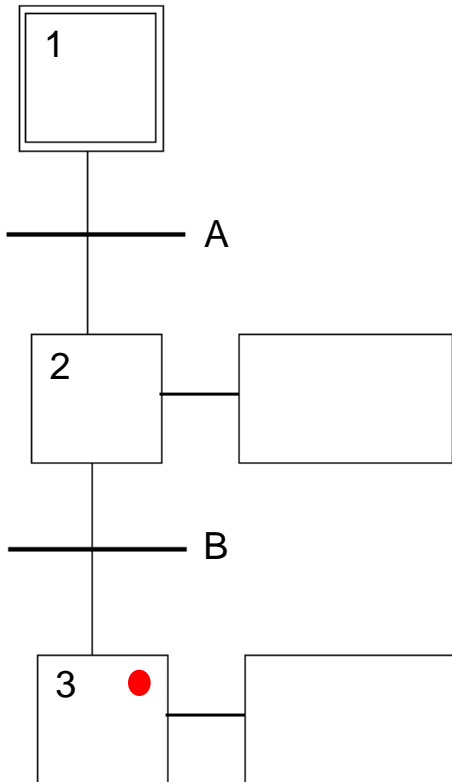
Ad esempio:

- ✓ Cosa succede se le condizioni A e B sono contemporaneamente verificate?
- ✓ Si attiverebbe il passo 2 e...



Ad esempio:

- ✓ Cosa succede se le condizioni A e B sono contemporaneamente verificate?
- ✓ Si attiverebbe il passo 2 e...
“immediatamente” dopo il passo 3
- ✓ Peccato che “immediatamente” non si possa ottenere con dispositivi reali e tale avverbio vada interpretato...



Ad esempio:

- ✓ Cosa succede se le condizioni A e B sono contemporaneamente verificate?
- ✓ Si attiverebbe il passo 2 e...
“immediatamente” dopo il passo 3
- ✓ Peccato che “immediatamente” non si possa ottenere con dispositivi reali e tale avverbio vada interpretato...
- ✓ Si attiva nello stesso ciclo? Nel ciclo successivo? Se nello stesso ciclo, eseguo o no le azioni contenute nel passo 2? C'è differenza tra azioni del passo 2 di tipo P (impulsivo) o N (non-stored, a livello)?



✓ Algoritmi di Evoluzione senza ricerca di stabilità

- Rappresentazione in forma algoritmica delle regole di evoluzione tali che in presenza di sequenze di transizioni con condizione logica vera, queste **vengono superate in cicli diversi**, e non nello stesso ciclo
- **Tutte le uscite associate ai passi intermedi vengono assegnate** (anche se il gettone di attivazione rimane nel passo un tempo piccolo a piacere)

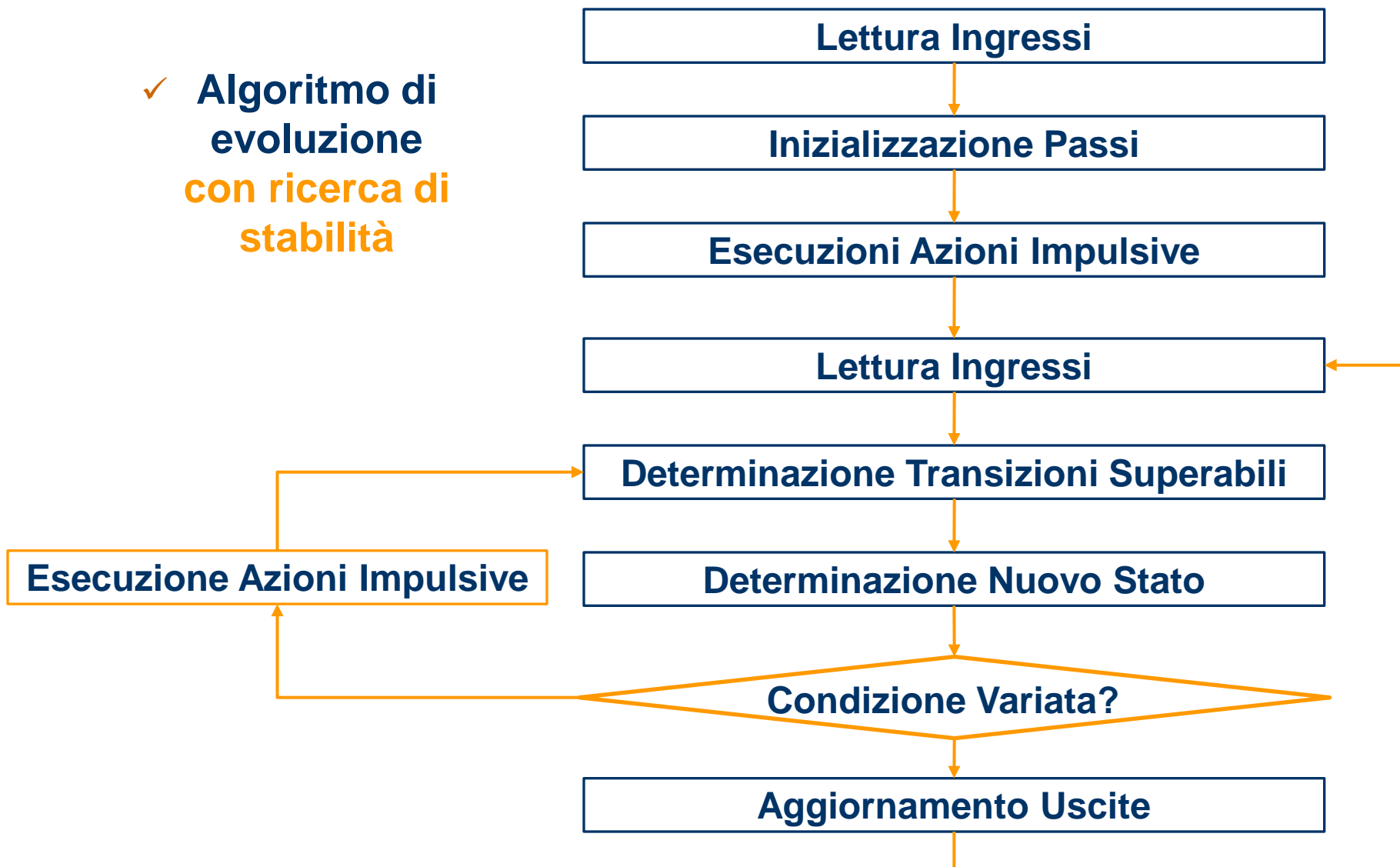
✓ Quattro passi ad ogni ciclo:

1. **Lettura Ingressi**
2. **Identificazione Transizioni superabili**
3. **Determinazione nuovo stato**
4. **Aggiornamento uscite**





✓ **Algoritmo di
evoluzione
con ricerca di
stabilità**

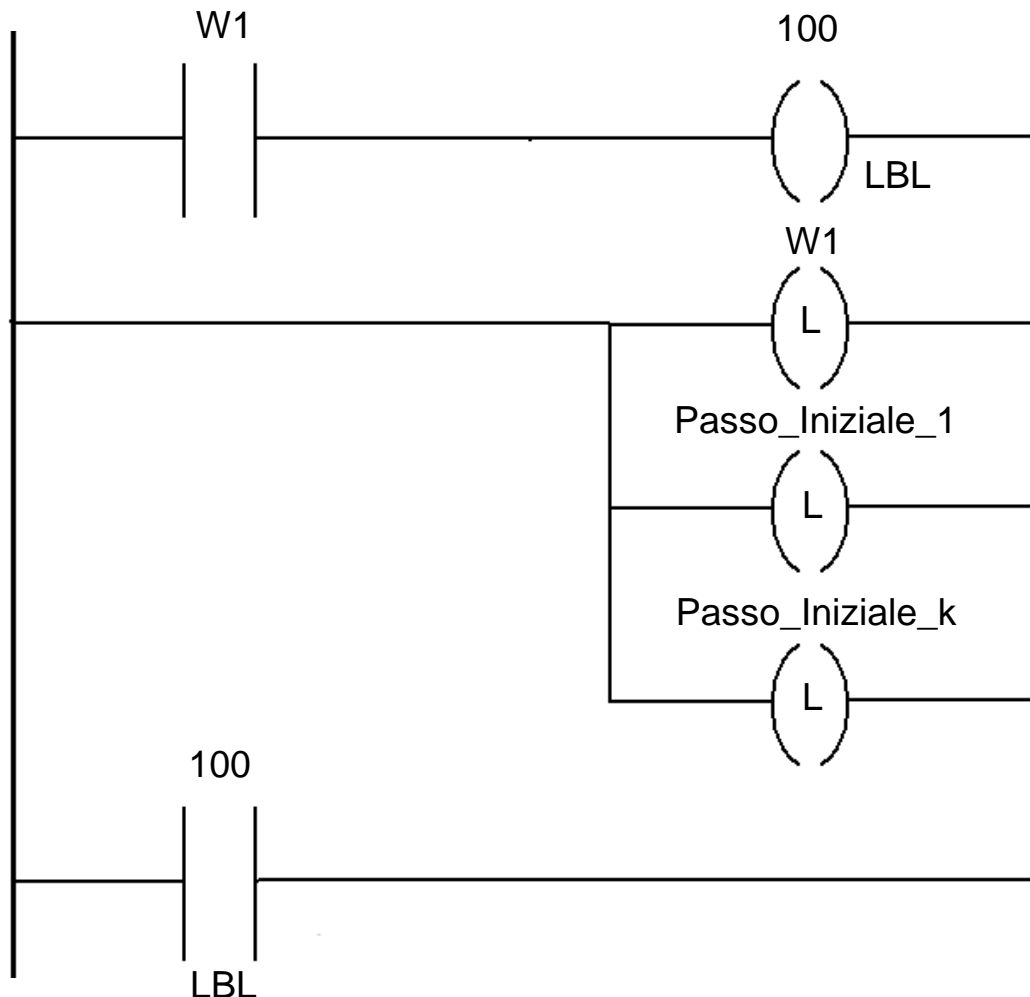




- ✓ **Implementazione in Ladder Diagram di uno schema SFC secondo l'algoritmo di evoluzione senza ricerca di stabilità**
- ✓ **Ad ogni passo si associa un bit di memoria (rappresenta lo stato del passo)**
- ✓ **Ad ogni transizione si associa un bit di memoria (rappresenta la condizione che una transizione è superabile in un certo stato)**
- ✓ **Scomposizione programma in quattro sezioni:**
 1. Sezione di inizializzazione
 2. Sezione di esecuzione delle azioni
 3. Sezione di valutazione delle transizioni
 4. Sezione di aggiornamento dello stato



- ✓ **Eseguita una sola volta: inizializza ad 1 gli stati dei passi iniziali**





- ✓ Per ogni azione di tipo N si prevede un piolo di abilitazione su cui si trovano in OR tutti i passi che prevedono quella azione; eventualmente, se un passo contiene un'azione condizionata da una condizione C1, lo stato di quel passo va in AND con C1
- ✓ Per le azioni memorizzate si usano bobine di tipo Latch o Unlatch
- ✓ Per le azioni impulsive si dovrà codificare in LD l'opportuno rivelatore di fronte



Traduzione in Ladder Diagram - Azioni



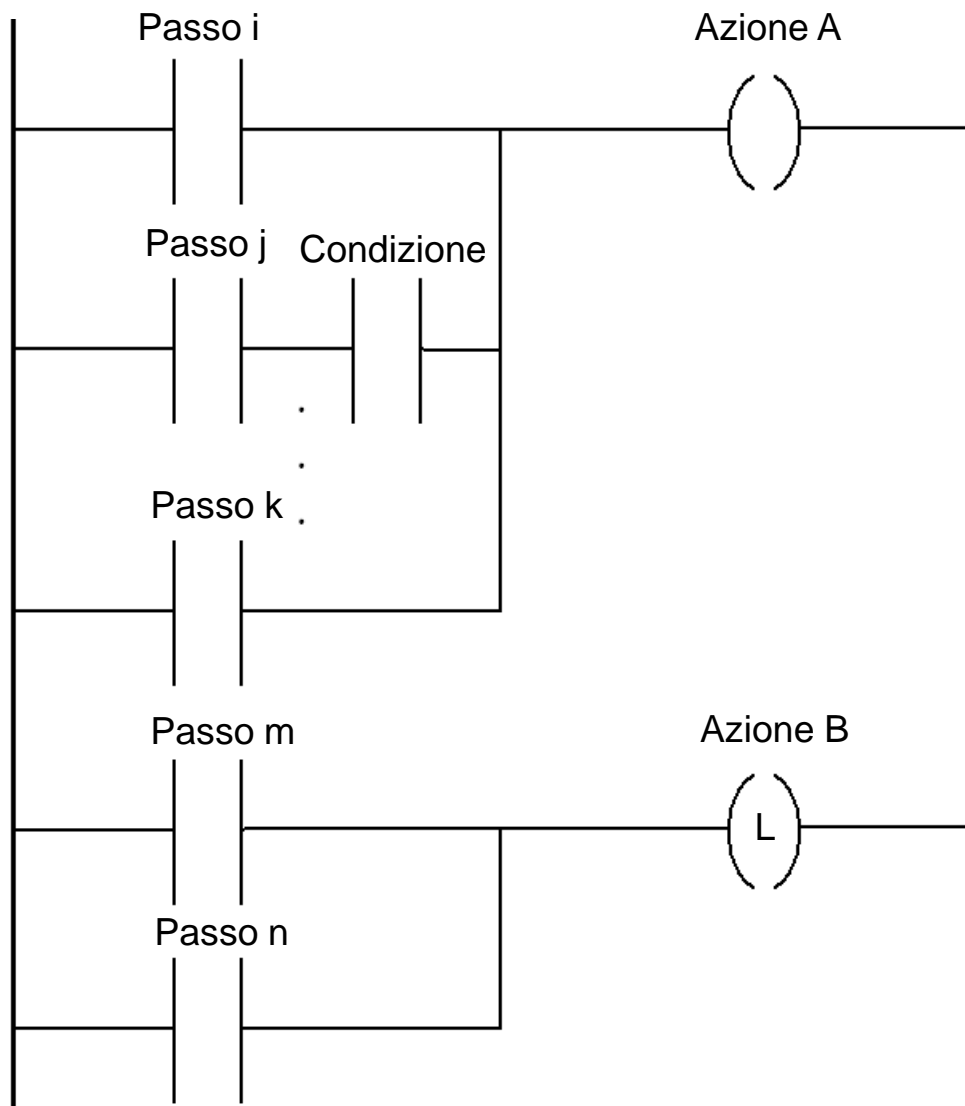
Introduzione

Elementi Base

Operatori Base

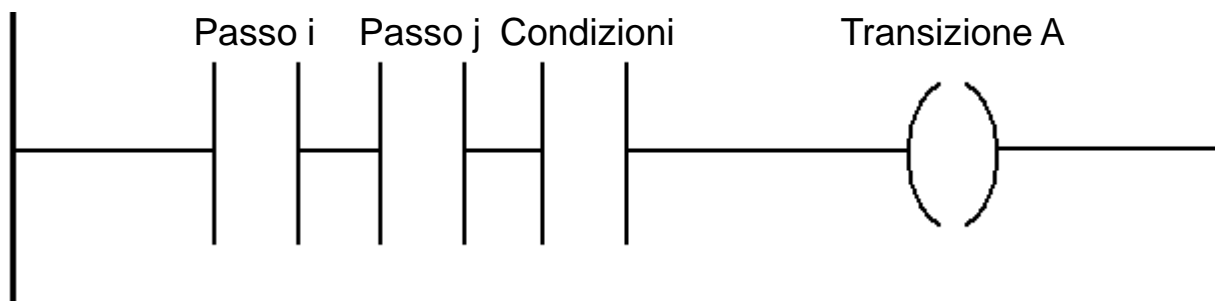
Regole di Evoluzione

LD ed SFC





- ✓ Il bit associato ad ogni transizione è attivato se la transizione è superabile
- ✓ Si ha un piolo per ogni transizione





Traduzione in Ladder Diagram – Aggiornamento Stato



Introduzione

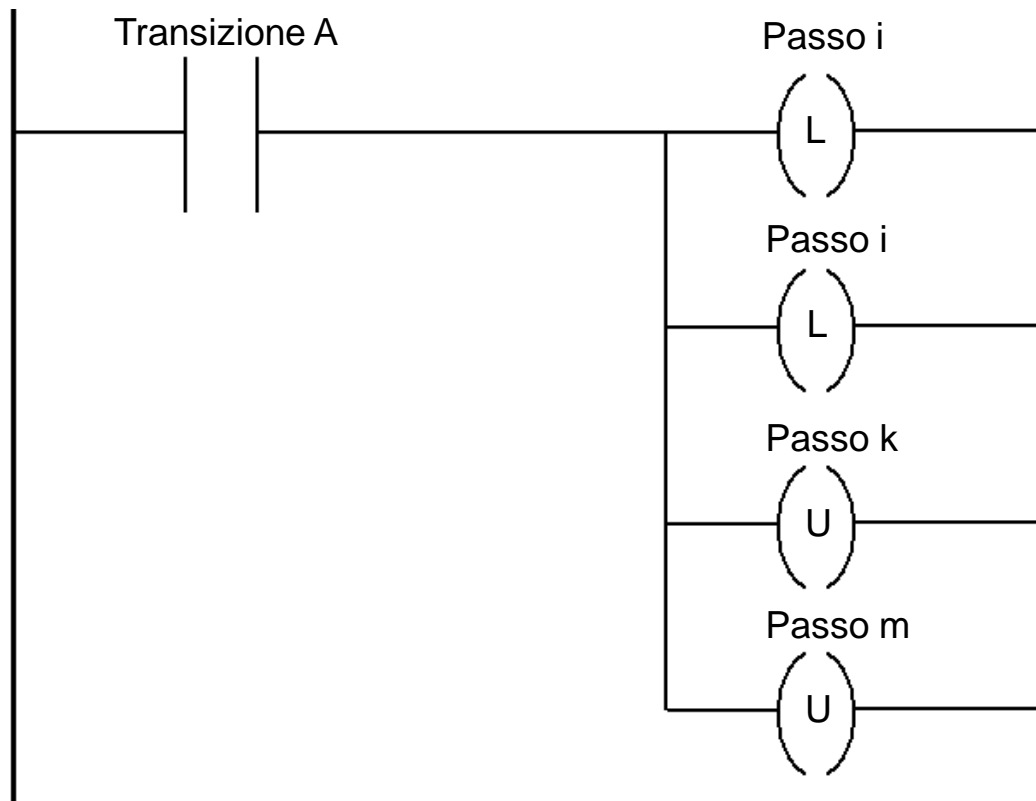
Elementi Base

Operatori Base

Regole di Evoluzione

LD ed SFC

- ✓ In corrispondenza delle transizioni superabili scrive 0 nei bit degli stati dei passi a monte e scrive 1 in quelli relativi ai passi a valle





Traduzione in Ladder Diagram – Variabili Temporal



Introduzione

Elementi Base

Operatori Base

Regole di Evoluzione

LD ed SFC

- ✓ Le variabili temporali devono essere realizzate con dei temporizzatori senza ritenzione attivati dai bit di stato dei passi.
- ✓ Questi temporizzatori possono essere messi in una sezione a parte





Traduzione in Ladder Diagram – cenno all'algoritmo con ricerca di stabilità



Introduzione

Elementi Base

Operatori Base

Regole di Evoluzione

LD ed SFC

- ✓ Associamo al bit Ta il significato di “almeno una transizione è stata abilitata durante la valutazione delle condizioni”
- ✓ Questo bit può essere attivato durante la sezione di aggiornamento della Condizione

