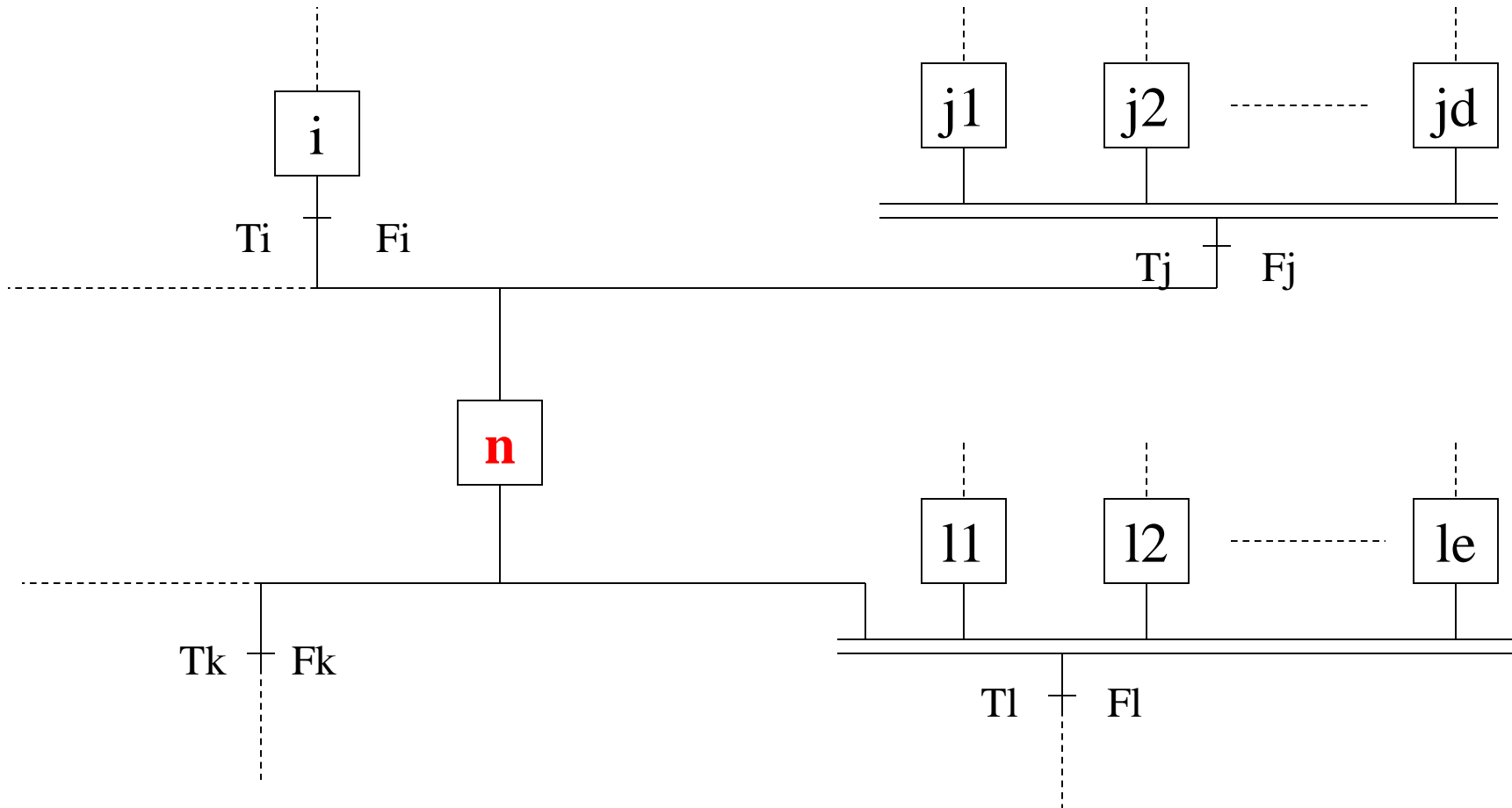


TRADUZIONE DEGLI SFC IN LADDER LOGIC

Equazioni Booleane Equivalenti
Algoritmi di Evoluzione
Traduzione in Ladder Logic

REGOLE DI EVOLUZIONE

◇ Data una generica fase n di un SFC



EQUAZIONI BOOLEANE EQUIVALENTI

- ◇ Equazione dello stato:

$$X_n = \sum_i (X_i \cdot F_i) + \sum_j \left[F_j \cdot \prod_{h=1}^{d_j} X_{jh} \right]$$

- ◇ Equazione di disattivazione

$$X_n = X_n \cdot \text{NOT} \left\{ \sum_k F_k + \sum_l \left[F_l \cdot \prod_{h=1}^e X_{lh} \right] \right\}$$

- ◇ Equazioni di uscita

$$A = X_n$$

$$B = X_n \cdot c$$

- ◇ **Ma le transizioni si perdono in questa rappresentazione ...**

ALGORITMI DI EVOLUZIONE

⇒ Algoritmi senza ricerca di stabilità

✓ **Quattro** passi ad ogni ciclo:

✓ **Lettura ingressi**

✓ **Transizioni superabili**

✓ **Determinazione nuovo stato**

✓ **Aggiornamento uscite**

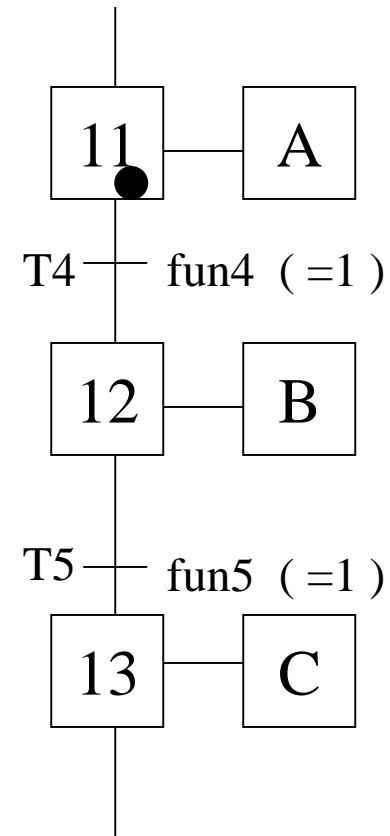
✓ **Ma le uscite vengono aggiornate ad ogni scansione dell'SFC: questo potrebbe essere sbagliato!**



ALGORITMI DI EVOLUZIONE

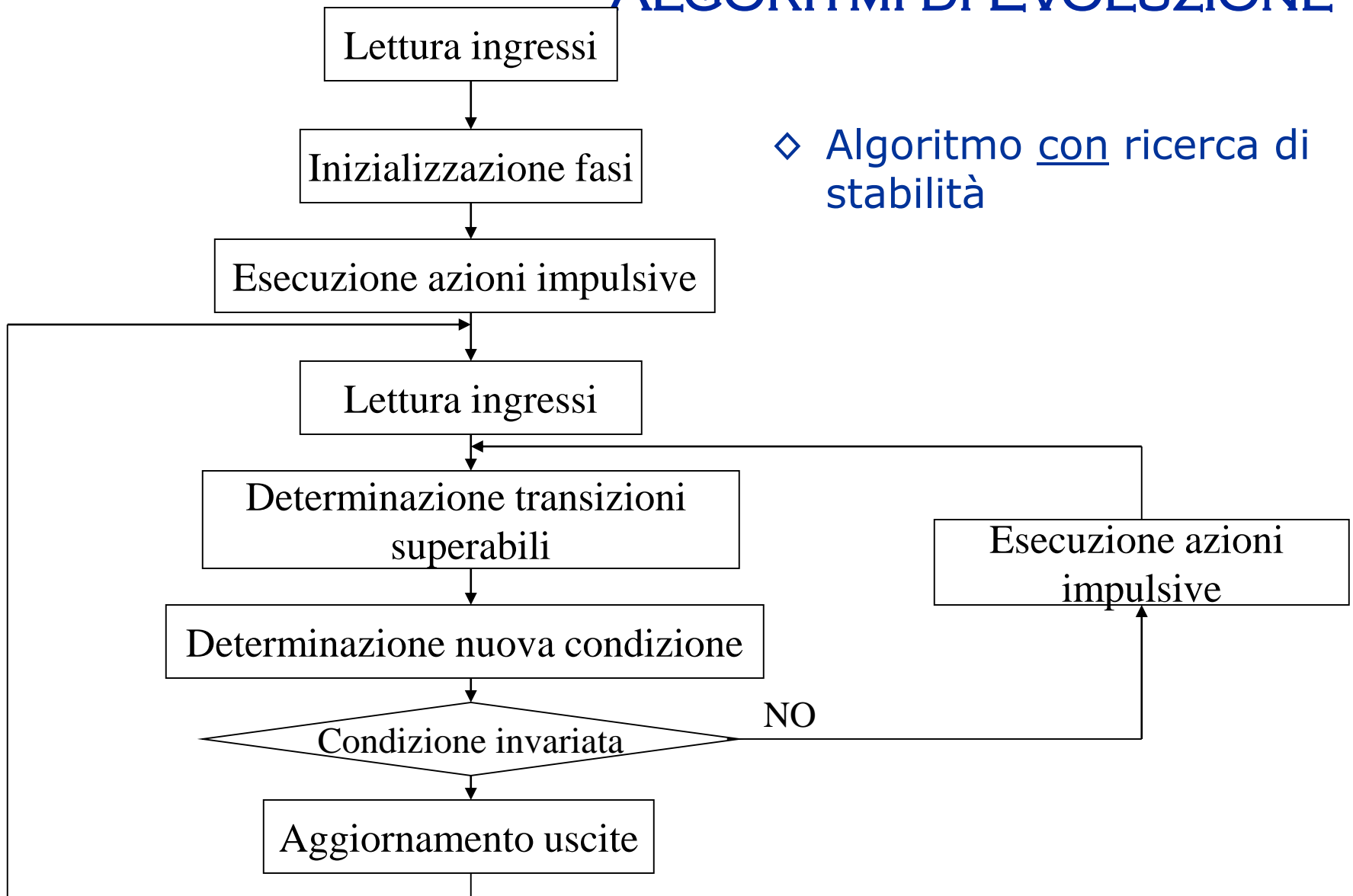
◇ Cosa succede per una fase non stabile?

- ◇ In questo caso l'azione continua B non deve essere accesa nemmeno per una scansione
- ◇ Ma se B fosse impulsiva allora deve essere generato l'impulso



ALGORITMI DI EVOLUZIONE

◇ Algoritmo con ricerca di stabilità

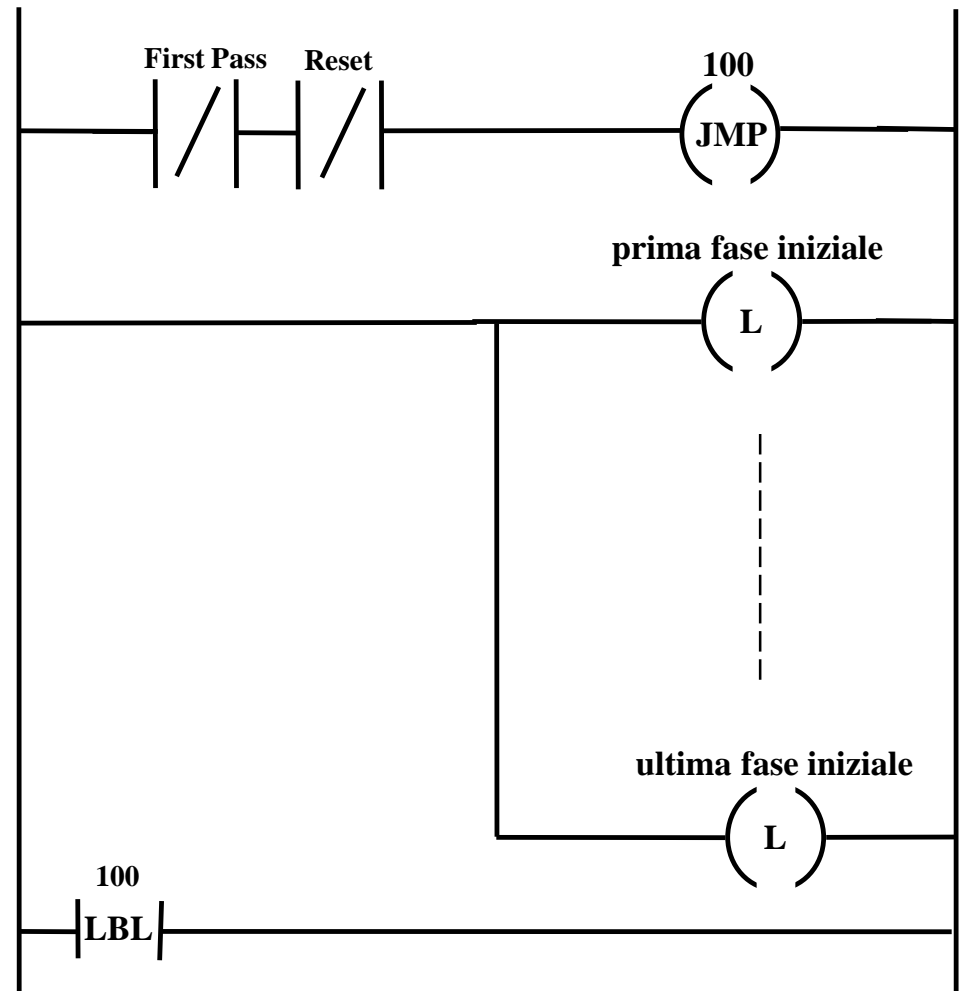


TRADUZIONE IN LADDER LOGIC

- ◇ Ad ogni **fase** si associa un bit di memoria (**marker di fase**)
- ◇ Ad ogni **transizione** si associa un bit di memoria (**marker della transizione**)
- ◇ Quattro sezioni:
 - ◇ Sezione di **inizializzazione**
 - ◇ Sezione di **esecuzione delle azioni**
 - ◇ Sezione di **valutazione delle transizioni**
 - ◇ Sezione di **aggiornamento della condizione**

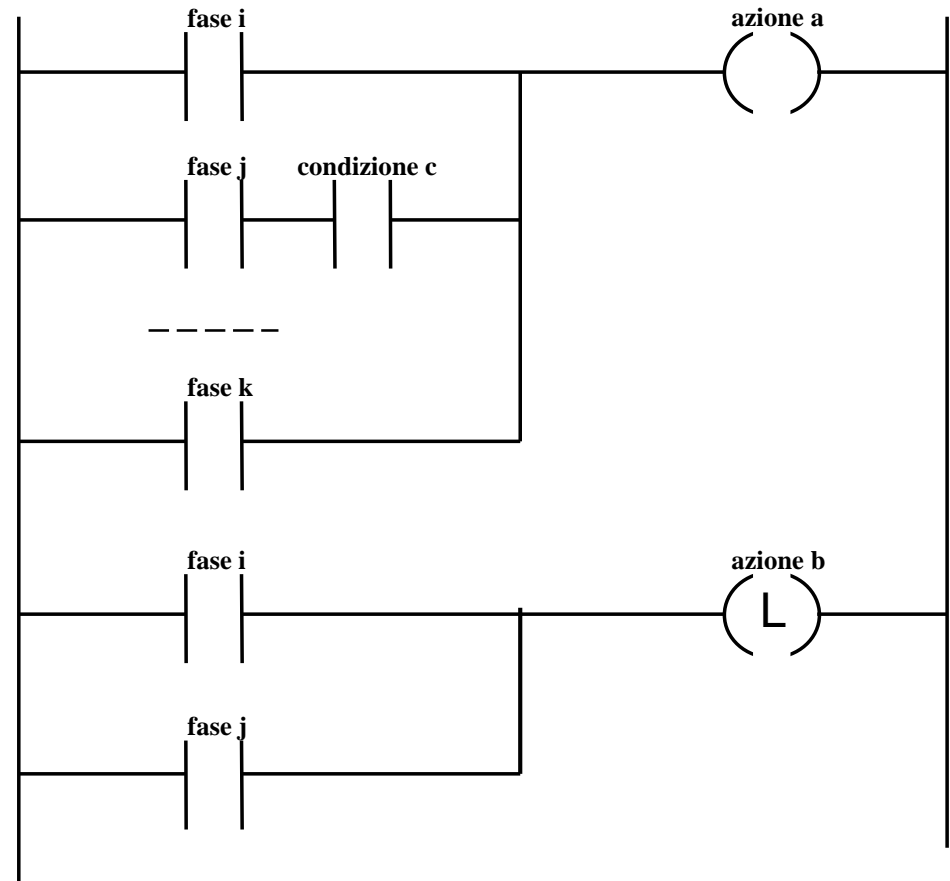
INIZIALIZZAZIONE

- ◇ Eseguita una sola volta
inizializza ad **1** i marker
delle fasi iniziali



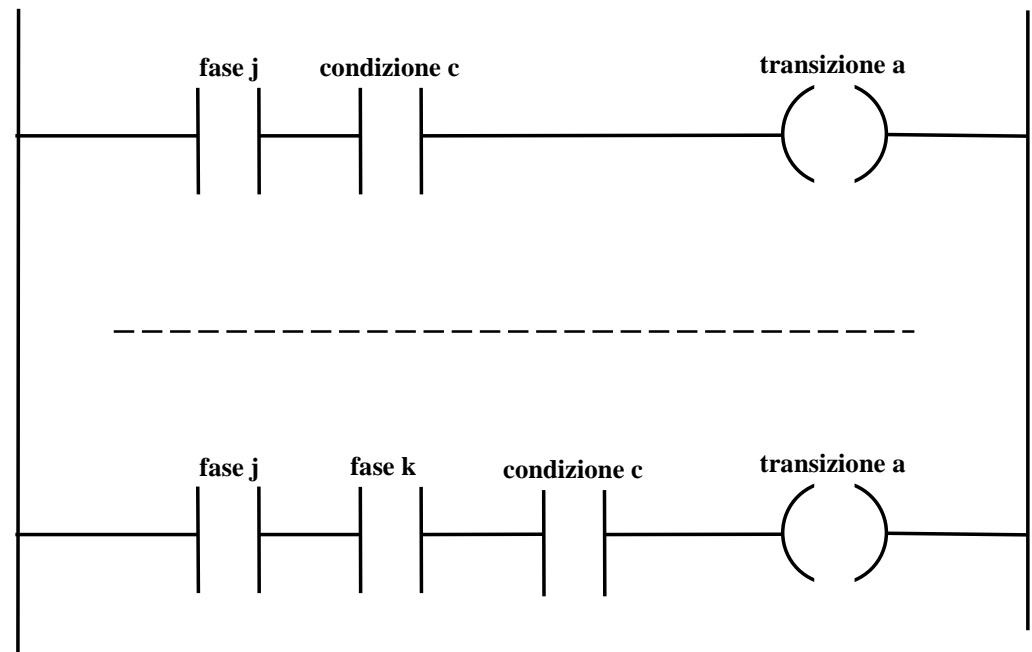
ESECUZIONE DELLE AZIONI

- ◇ Per ogni azione **continua** si prevede un rung di abilitazione su cui si trovano in OR tutte le fasi che implicano quella azione eventualmente in AND con delle condizioni
- ◇ Per le azioni **memorizzate** si userà LATCH/UNLATCH bobina
- ◇ Per le azioni **impulsive** si dovrà codificare l'LCV che produce l'opportuno impulso



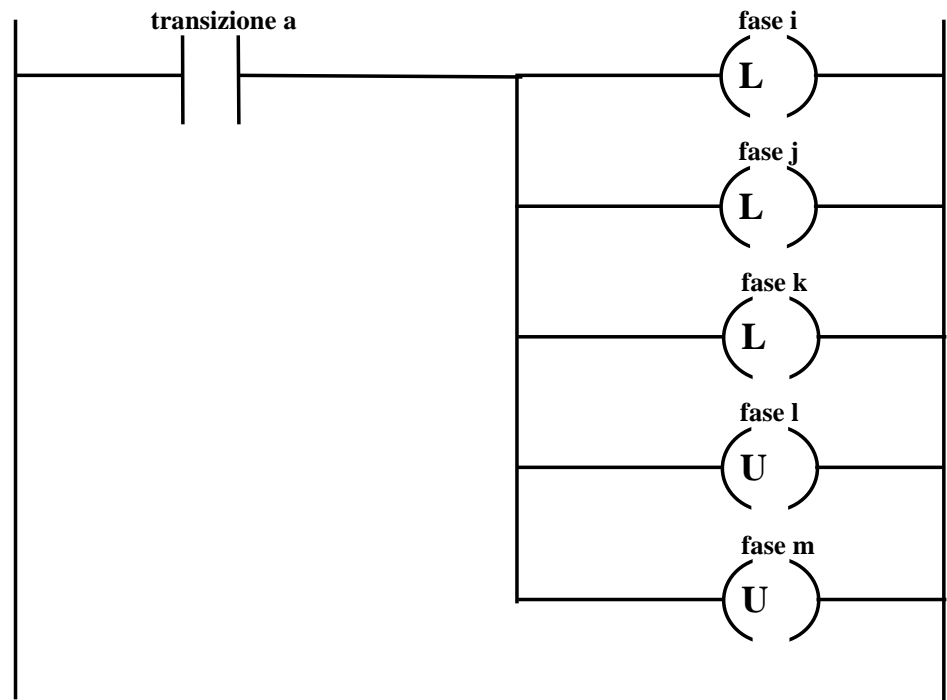
VALUTAZIONE DELLE TRANSIZIONI

- ◇ Il marker di ogni transizione è **attivato** se la transizione è superabile
- ◇ Si ha un rung per ogni transizione



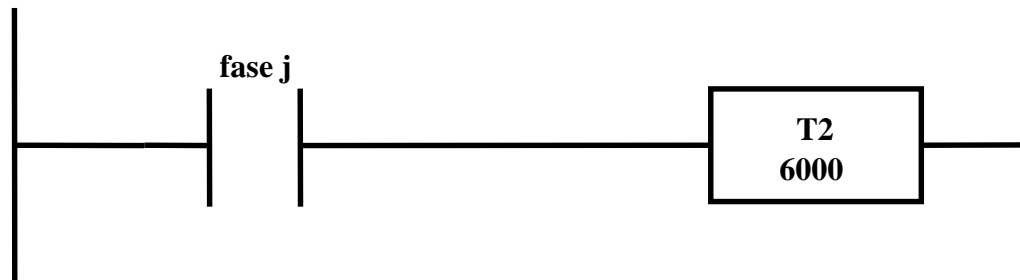
AGGIORNAMENTO CONDIZIONE

- ◇ In corrispondenza delle transizioni attivate **spegne** i marker delle fasi a monte ed **accende** quelli delle fasi a valle



TRADUZIONE DELLE VARIABILI TEMPORALI

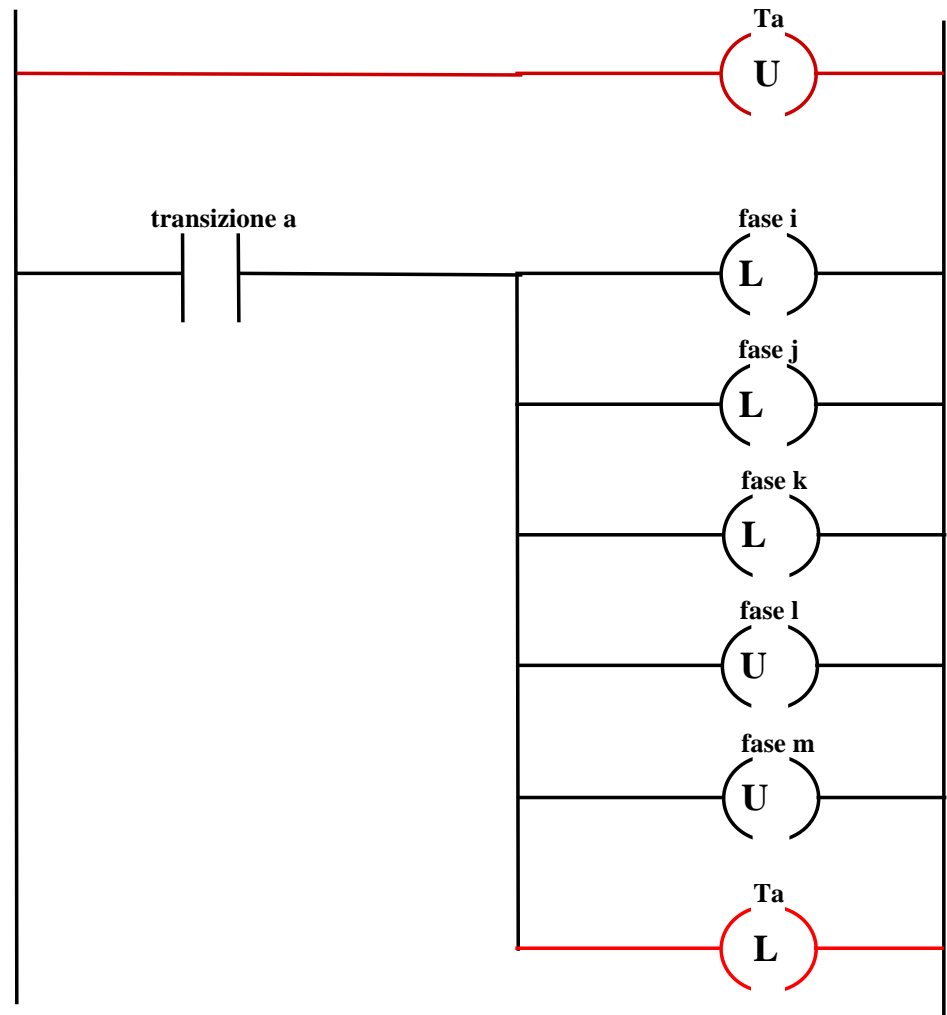
- ◇ Le variabili temporali devono essere realizzate con dei temporizzatori senza ritenzione attivati dall'accensione dei marker di fase
- ◇ Questi temporizzatori possono essere messi in una sezione a parte



- ◇ $t/Xj/6\text{sec} \equiv T2$

ALGORITMO CON RICERCA DI STABILITÀ: AGGIORNAMENTO CONDIZIONE

- ◇ 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



ALGORITMO CON RICERCA DI STABILITÀ: ESECUZIONE

- ◇ Solo se T_a non è abilitato aggiorniamo le azioni continue
- ◇ Le azioni impulsive le eseguiamo sempre

