

Modalità di Traduzione Alternativa da SFC a
Ladder

Angelo Carusotti, Alessandro Proietto, Adriano Gizzi

7 dicembre 2007

Il Ladder Diagram (o linguaggio a contatti) è attualmente il più diffuso linguaggio di programmazione per PLC. I contatti possono essere associati agli ingressi digitali provenienti dal processo, o meglio al loro stato rappresentato in particolari bit della memoria, oppure a condizioni interne al dispositivo, sempre rappresentate da bit di memoria; la bobina può essere associata a un bit della memoria e, per suo tramite, comandare un'uscita digitale o variare una condizione interna.

Il linguaggio Ladder viene scritto con una semplice traduzione dal linguaggio SFC. Si è scelto di utilizzare una modalità di traduzione diversa da quella "Classica". Questa modalità verrà qui analizzata e confrontata con quella "Classica".

Modalità di Traduzione "Classica"	Modalità di Traduzione Utilizzata
Ad ogni Fase si associa un bit di memoria (Marker di Fase)	Ad ogni Fase si associa un bit di memoria (Marker di Fase)
Ad ogni transizione si associa un bit di memoria (Marker di Transizione)	Ad ogni transizione si associa un bit di memoria (Marker di Transizione)
Quattro Sezioni	Tre Sezioni
Inizializzazione	Inizializzazione
Valutazione Transizioni	Valutazione Transizioni e Aggiornamento Condizioni
Aggiornamento Condizioni	
Esecuzione Azioni	Esecuzione Azioni

Chiariamo meglio quanto esposto con un esempio:

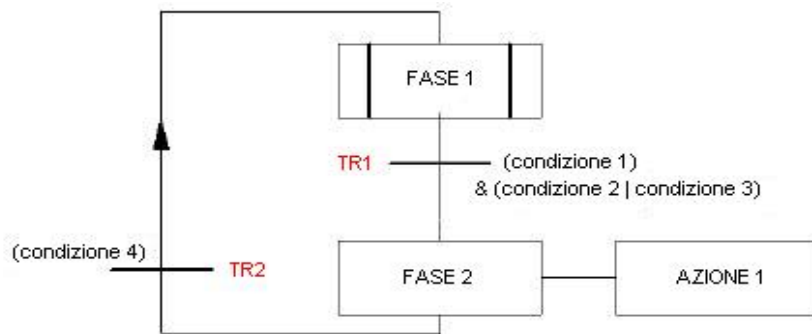


Figura 1: SFC a scopo illustrativo

Vediamo ora la traduzione in Ladder di questo SFC effettuata utilizzando prima una modalità e poi l'altra.

Modalità di Traduzione “Classica”



Figura 2: Inizializzazione

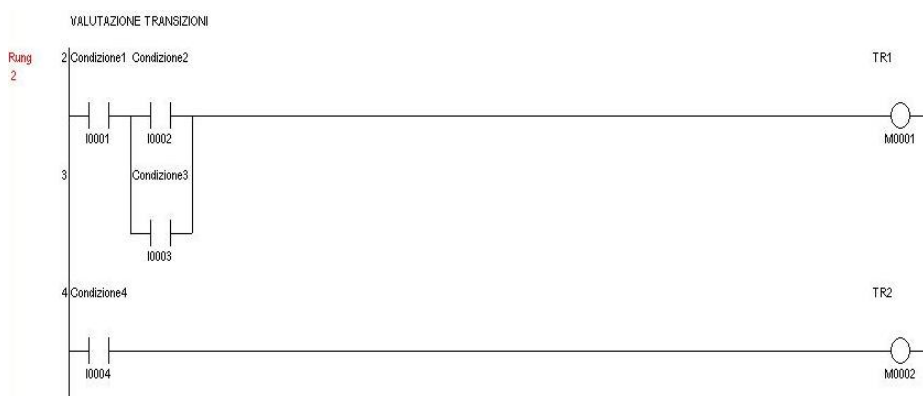


Figura 3: Valutazione Transizioni



Figura 4: Aggiornamento Condizioni



Figura 5: Azioni

Modalità di Traduzione Utilizzata



Figura 6: Inizializzazione

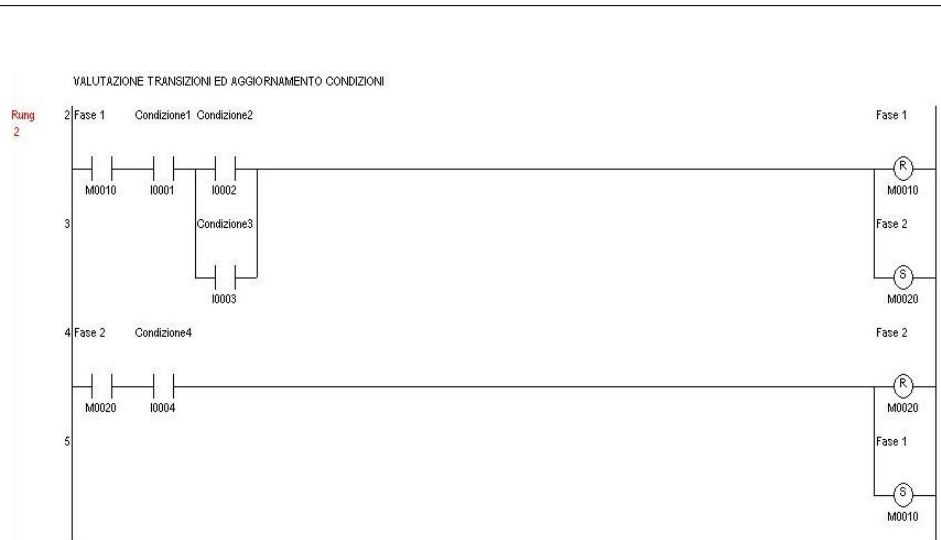


Figura 7: Valutazioni Transizioni ed Aggiornamento Condizioni



Figura 8: Azioni

Come si può facilmente vedere i vantaggi della seconda modalità sono molteplici:

- Minore Utilizzo di memoria
- Minor numero di Rung di codice da scrivere
- Minor numero di Rung da controllare durante il Debugging
- Possibilità di controllare una fase (o meglio il Marker di tale fase) e la transizione a Valle di essa (o meglio le condizioni che esprimono tale Transizione) direttamente su una stessa Rung, migliorando dunque ulteriormente la qualità del Debugging (non si deve saltare da una parte all'altra del programma).

ATTENZIONE: L'Algoritmo di traduzione presentato è un Algoritmo senza Ricerca di Stabilità. Ciò vuol dire che data una sua fase con relativa azione e due transizioni a monte e a valle di essa, se le condizioni di tali transizioni sono entrambe verificate l'azione relativa alla fase (sia l'azione impulsiva o meno) non sarà eseguita fintanto, che le due transizioni sia entrambe verificate.

N.B.

Anche l'Algoritmo di traduzione "Classico" è senza Ricerca di Stabilità.