Il sequential functional chart

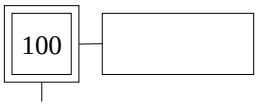
Il sequential functional chart o più brevemente SFC è un linguaggio di programmazione che consente di descrivere il funzionamento di macchine sequenziali. Nel 1975 in Francia fu istuita una commissione con lo scopo di formalizzare un nuovo linguaggio di programmazione che dovesse poi prendere il posto delle attuali tecniche descrittive per i processi di automazione. Il risultato fu il linguaggio GRAFCET (graphe de cordination etapes transitions) che nel 1987 entrò a far parte dello stanndard IEC 1131-3 con il nome di SFC. La maggior parte delle applicazioni industriali sono caratterizzate dall'esecuzione ciclica del software di controllo necessario a governare, in maniera real time, la macchina automatica (vale a dire il processo fisico e/o industriale più il controllore con la logica di controllo che agisce direttamente sugli attuatori). Il programma per il controllo del processo è inserito nei PLC industriali a cui giungono, tipicamente mediante bus, i segnali rilevati dai sensori dislocati sul campo e/o sul processo da controllare. Sempre dai PLC, inoltre, si snodano (ancora una volta mediante bus ma anche tramite singoli connettori se il processo da controllare non è assai complesso) numerose linee elettriche dirette verso gli attuatori.
L'esecuzione ciclica del software di controllo permette di trasferire sul processo, mediante gli attuatori, l'azione di controllo che scaturisce dall'elaborazione dei segnali ricevuti dai sensori di campo. Più in generale, l'esecuzione di ciascun ciclo di controllo prevede essenzialmente tre fasi: acquisizione dei segnali generati dai sensori; elaborazione dei segnali ricevuti tramite un algoritmo di controllo; attuazione dei segnali di controllo generati dall'algoritmo al passo precedente. Mentre nella maggior parte dei PLC industriali le fasi di acquisizione ed attuazione dei segnali sono eseguite in modo trasparente dal sistema operativo, la logica di controllo è descritta dal software. Sarà compito del programmatore rappresentare la logica di controllo (in questo note osserveremo l'SFC), provarla al simulatore e successivamente trasferirla nella memoria del PLC. In alcuni casi, qualora il PLC non conosca il linguaggio software adottato per modellare la logica di controllo, è necessaria un ulteriore fase di traduzione (ad esempio che traduca la logica di controllo scritta in SFC in LADDER, quest'ultimo è sicuramente riconosciuto dalla maggior parte dei PLC). La vera forza dell'SFC è la sua natura grafica di linguaggio di programmazione che lo rende semplice ed intuitivo. Tutti i sistemi a eventi discreti si prestano ad una descrizione mediante SFC (i sistemi a eventi discreti sono i sistemi il cui stato non varia con il passare del tempo ma solo all'occorrenza di particolari condizioni o eventi). Gli elementi base dell'SFC sono:

- la fase: identifica lo stato attuale della macchina a cui viene fatto corrispondere un azione di controllo (che in alcuni casi può anche essere nulla o vuota se la macchina non deve svolgere una precisa azione). Questo stato può essere

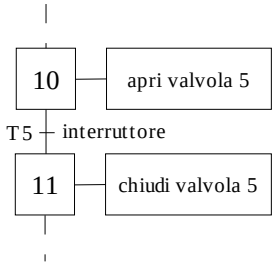
modificato solo se si verifica la transizione che sposta quindi la macchina in un nuovo stato, quello puntato dall'arco orientato;

- la transizione: rappresenta la condizione logica che assume l'espressione costruita mediante la combinazione di determinati segnali di sensori. Quest'ultimi se abilitati e/o disabilitati possono soddisfare l'intera condizione logica e permettere alla transizione di spostare l'attuale fase della macchina a quella immediatamente successiva;
- l'arco orientato: collega le fasi ed è interrotto dalle transizioni che determinano in questo modo la condizione da soddisfare affinchè si possono avere attivazioni e disattivazioni delle fasi. Quando una transizione può avvenire, la fase attuale è disattivata a scapito di quella puntata dall'arco orientato. In alcuni casi una freccia può indicare la direzione dell'arco, tuttavia se si adotta la convenzione di disporre il diagramma SFC in verticale assumendo tutte le transizioni valide dall'alto verso il basso non c'è possibilità di ambiguità e la suddetta freccia può dunque essere omessa;

Ogni fase che identifica uno stato della macchina ha un etichetta numerica che la identifica a cui viene associata una variabile booleana detta marker di fase. Quest'ultima assume valore logico 0 se la fase è inattiva e valore logico 1 se la fase è, invece, attiva. La fase iniziale di un SFC è identificata da un blocco (tipicamente un quadrato sufficientemente grande da contenerne l'etichetta di cui sopra) con doppia cornice. Se la fase dell'SFC è invece una comune fase dell'SFC il blocco che la rappresenta ha una singola cornice. L'azione di controllo associata alla fase, sia iniziale che comune, è interamente descritta in un rettangolo affiancato alla suddetta fase. Alcuni esempi:



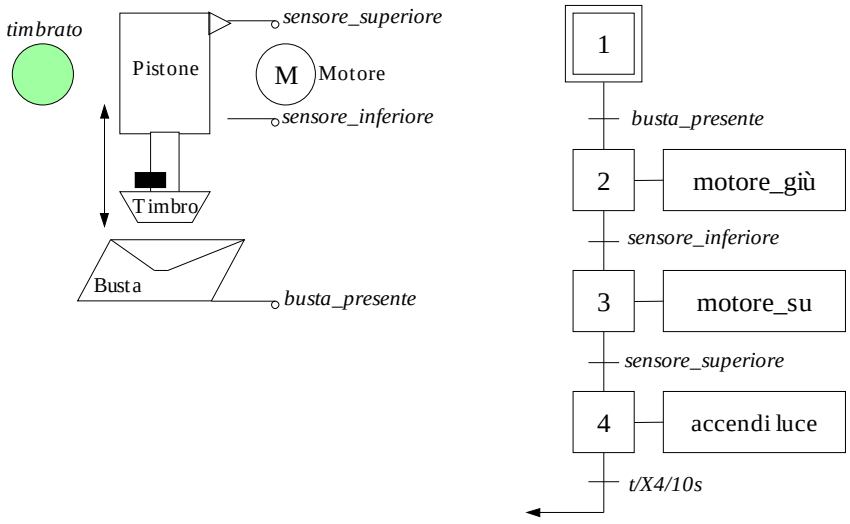
100 è la fase iniziale di un SFC a cui corrisponde un'azione nulla (la macchina che si trova in quello stato non svolge alcuna azione). *X100* è la variabile booleana assegnata allo stato 100. Quando la macchina è inizializzata (l'SFC quindi parte dallo stato iniziale) *X100=1,* non appena una transizione cambia lo stato della macchina *X100=0.* Ed ancora:



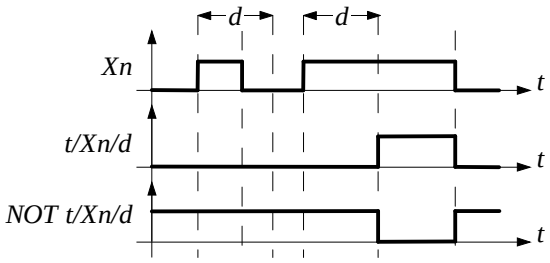
Nella fase (comune) 10 la macchina compie l'azione "apri la valvola 5" e rimane in questo stato (aprendo quindi sempre di più la valvola 5) finchè la condizione logica *interruttore!=0.* Non appena la condizione

diventa *interruttore=1* la transizione *T5* (anche le transizioni possono essere etichettate) permette alla macchina di portarsi nella fase 11 a cui corrisponde l'azione "chiudi la valvola 5".
Tra due fasi collegate da un arco deve esistere sempre una transizione e tra due transizioni deve esserci sempre una fase. Il progetto del software di controllo si ottiene isolando le azioni sequenziali che la macchina deve compiere. Tali azioni rappresenteranno gli stati della macchina. Ad ogni stato e per ogni azione possono poi essere considerati i soli segnali di ingresso e uscita necessari all'esecuzione di quella azione. Quindi si inseriscono le condizioni logiche (le transizioni) che collegano le varie fasi.

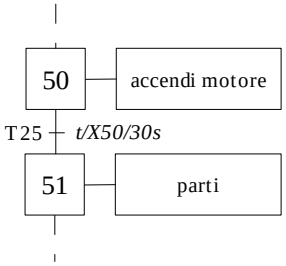
Esempio: timbratrice automatica
Descrizione del problema – La timbratrice automatica deve continuamente salire e scendere per effettuare timbri su delle buste che vengono poste sul supporto inferiore da un altra macchina. Quando il sistema di controllo della timbratrice automatica riceve il segnale di *busta presente* avvia il motore in direzione di discesa fino a che il *sensore inferiore* di fine corsa non segnale che il pistone di timbratura è arrivato in basso. A questo punto il motore deve invertire la sua corsa e far risalire il pistone finchè il *sensore superiore* del sistema di controllo non viene attivato. Fatto ciò la timbratice comunica all'esterno che la busta può essere rimossa dalla sede di lavorazione accendendo una luce per 10 secondi.



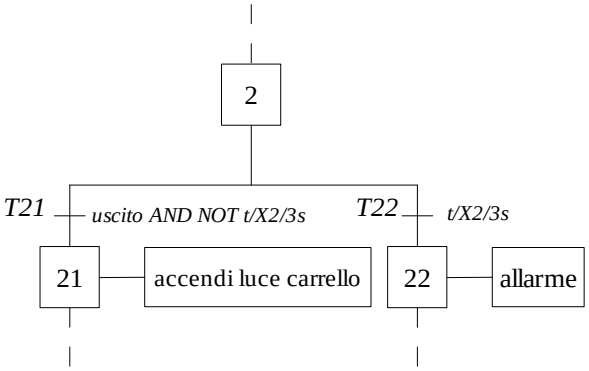
Questo esempio ci permette di introdurre il concetto di variabile temporale che nella sintassi dell'SFC si denota come *t/Xn/d*. Questa variabile assume valore logico 0 fintanto che la macchina non ha passato *d* secondi nella fase associata alla variabile booleana *Xn*. Quindi, nell'esempio della timbratrice automatica la macchina rimane bloccata nella fase 4 (a cui corrisponde l'azione "accendi luce timbrato") finchè non passano 10s. In presenza di una variabile temporale come condizione o parte di una condizione può essere utile capire come evolve la suddetta variabile osservandone il diagramma temporale. Eccone un esempio:



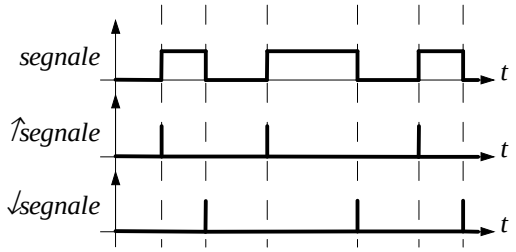
Un altro esempio di come utilizzare la variabile temporale è il seguente:



La variabile temporale permette di realizzare, inoltre, interessanti strutture come i watchdog timer. Il watchdog timer attiva un allarme se un evento non si verifica entro un certo tempo prestabilito. Eccone un esempio:

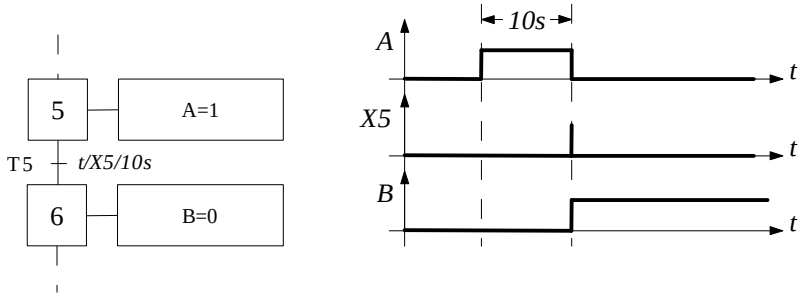


Nell'SFC in figura il sistema di automazione controlla l'estrazione del carrello di atterraggio per un aereo. Il sensore *uscito* indica la completa estrazione del carrello di atterragio, l'intera procedura per l'estrazione del carrello deve durare 3s, se ciò non si verifica la variabile temporale abilita la transizione *T22* e fa scattare la fase 22 a cui corrisponde come azione l'esecuzione di un segnale di allarme.
La variabile ed i segnali utili all'SFC possono avere una notazione classica, qualora la variabile segue perfettamente il segnale generato dal sensore di campo, ed una notazione orientata invece sui fronti di salita della variabile associata al segnale del sensore. La variabile ↑*segnale* reagisce ai fronti di salita della variabile *segnale*; essa assume valore logico alto solo nell'istante di tempo in cui *segnale* compie una transizione logica da 0 ad 1. La variabile ↓*segnale*, invece, reagisce sui fronti di discesa della variabile *segnale*:

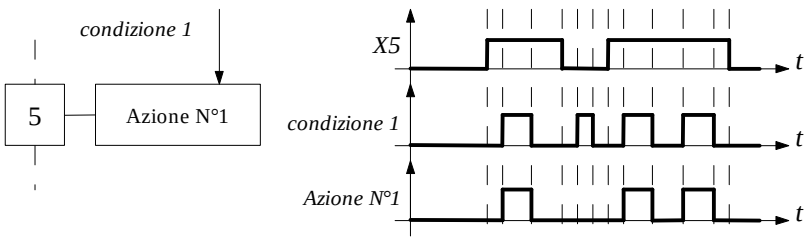


L’evoluzione di un automatismo descritto da uno schema SFC può essere visivamente interpretato e simulato come una rete di contenitori (le fasi e le azioni) collegati tra di loro attraverso le transizioni. Lo stato di attivazione di una fase è stabilito da un token immaginario; solo i possessori di un token sono attivi. Lo stesso token è quindi passato alla fasse successiva seguendo (oppure percorrendo) l’arco orientato solo se la condizione che abilita quell’arco è verificata. Ad una fase può essere associata una o più azioni (se più azioni sono associate ad una fase queste vengono elencate nell’unico blocchetto che quindi le raccoglie, eventualmente separate dal simbolo punto e virgola, oppure ogni azione associata alla fase è scritta in un blocchetto a parte e ciascun blocchetto descrittivo di un azione è quindi collegato alla fase. L’SFC mette a disposizione diverse tipologie di azioni:

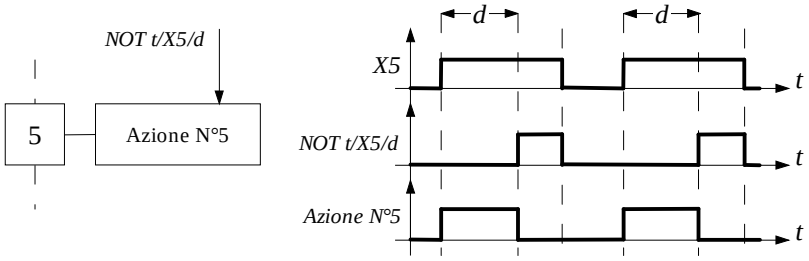
- azione continua: è la forma classica di azione, rappresenta l’esigenza di avere in uscita un segnale alto durante tutto il periodo di attivazione della fase. In altre parole il segnale di uscita è tenuto alto finchè la macchina è ferma nella fase ad essa associata. Non appena la macchina ha la possibilità di passare alla fase successiva il segnale di uscita assegnato alla fase viene abbassato.



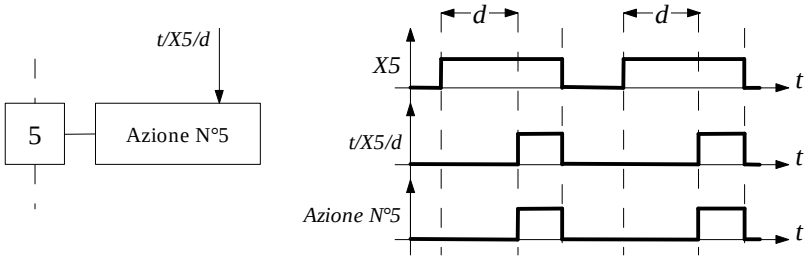
- azione condizionata: prevede una condizione aggiuntiva associata alla fase. Anche se la fase è attiva l’azione non sarà eseguita fintanto che la condizione ad essa associata non sarà anch’essa abilitata:



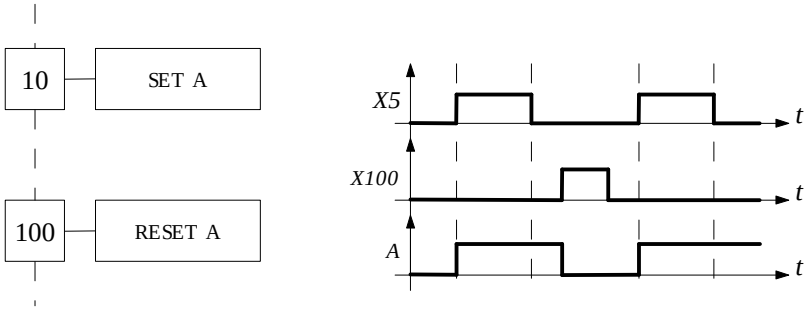
- azione limitata nel tempo: è ancora un'azione condizionata con la differenza che la condizione che abilita la fase è legata ad una variabile temporale negata che ne limita l'effetto nel tempo. L'azione viene eseguita finchè la variabile temporale (diventando abilitata) non ne determina la fine:



- azione ritardata nel tempo: è il caso duale all'azione limitata nel tempo. L'azione associata alla fase è eseguita finchè la variabile temporale, anch'essa associata alla fase, non ne decreta la fine:



- azione memorizzata: è utile a tenere settata una variabile booleana piuttosto che ripeterla ogni volta nelle fasi successive. Sono previste due azioni di memorizzazione: il set (S) imposta ad 1 una variabile booleana mentre il reset (R) imposta a 0 la stessa variabile booleana. Questi ordini, da considerarsi come azioni di tipo impulsivo, sono eseguiti quando le fasi a cui sono associati diventano attive. In tutte le altre fasi l'uscita conserva l'ultimo valore assegnatole:

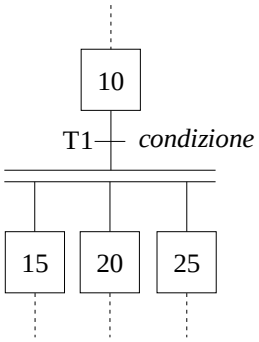


Infine, un azione associata ad una fase può essere implementata anche utilizzando i diversi linguaggi definiti dalla normativa IEC 1131-3 (appartengono ai linguaggi riconosciuti dalla suddetta normativa le istruzioni in instruction list (IL), quelle scritte in testo strutturato,

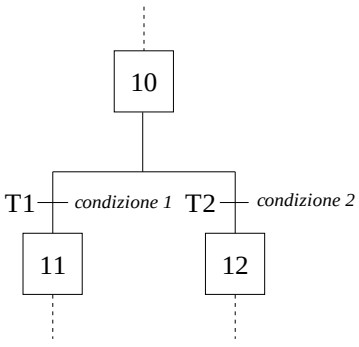
quelle espresse sottoforma di diagrammi LADDER (LD), quelle fatte da reti di blocchi funzionali (FB) e quelle rappresentate anche da altre SFC). Alcuni qualificatori per azioni sono:

Simbolo	Significato	Descrizione
N	non stored	L'azione termina quando la fase è disattivata.
R	reset	Azzera la variabile booleana a cui è associata.
S	set	Imposta la variabile booleana a cui è associata.
L	time limited	L'azione comincia non appena la fase è attivata e termina al trascorrere di un certo intervallo di tempo.
D	time delay	L'azione comincia solo dopo che la macchina occupa una fase da un certo intervallo di tempo.
P	pulse	L'azione è svolta quando la fase è attiva ed è eseguita una sola volta.

Nell'SFC sono poi presenti strutture di controllo classiche che consentono di modellare le diverse esigenze che si possono incontrare in fase di modellazione del software. Nella struttura del parallelismo una condizione può abilitare più fasi:

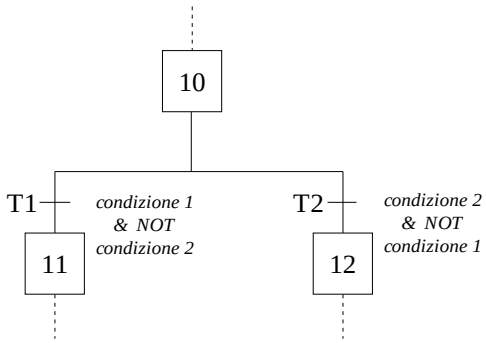


La seguente struttura, invece, realizza una scelta:

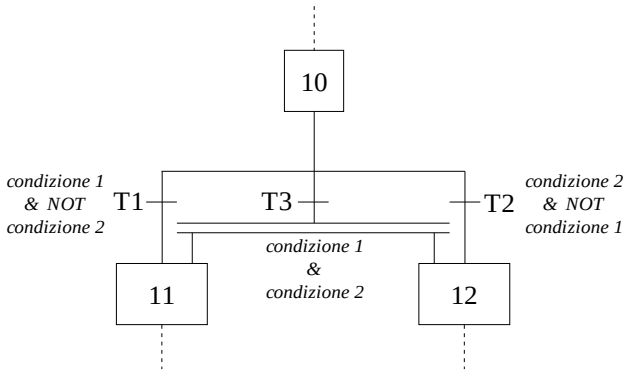


Tale struttura può tuttavia presentare delle ambiguità poichè se entrambe le condizioni sono abilitate è possibile far scattare entrambe le fasi 11 e 12. Ciò non è affatto un errore di sintassi ma chi ci dice che

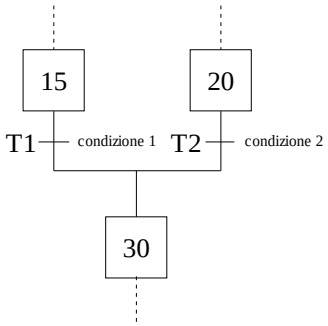
corrisponde a ciò che si intendeva modellare? Per questo motivo, se stiamo pensando di inserire nell'SFC una struttura di controllo capace di effettuare una scelta (in base alla condizione abilitata) dobbiamo essere certi che le condizioni T1 e T2 siano mutue ed esclusive. La mutua esclusione delle condizioni, allora, può essere garantita dalla natura delle variabili (magari perchè associate ad espressioni e/o ad eventi che mai si verificano in concomitanza) oppure se ciò non avviene essa deve essere costruita di proposito mediante una sintassi più precisa come quella proposta di seguito:



Nel caso in cui si vuole realmente avere la possibilità di abilitare entrambe le fasi è più opportuna una struttura di questo tipo:

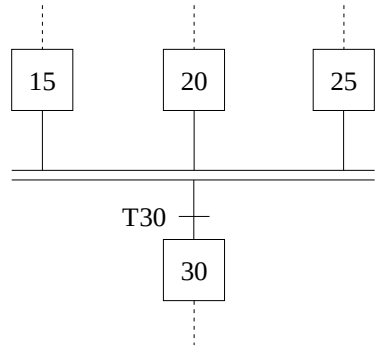


Si ha una struttura di convergenza quando due o più sequenze terminano nella stessa fase attraverso condizioni diverse:

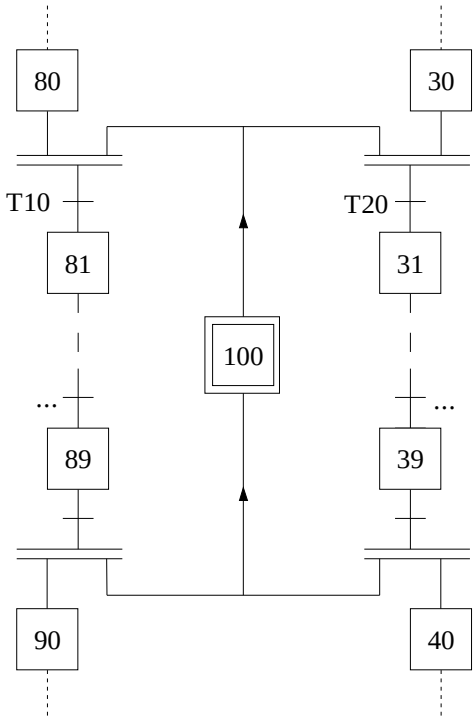


Le fasi 15 e 20 rappresentano le fasi di una sequenza finale. Entrambe le sequenze si ricongiungono in unica sequenza se è attiva una delle due condizioni *condizione 1* o *condizione 2*. Se la fase 15 è attiva e la

condizione espressa da T1 è abilitata, la fase 30 può essere innescata. Quest'ultima continua ad essere attivata anche se in seguito dovesse risultare abilitata la condizione espressa da T2 (gli eventi possono anche accadere in ordine inverso rispetto a quelli appena descritti). Un'altra struttura assai utile è quella che realizza la sincronizzazione:

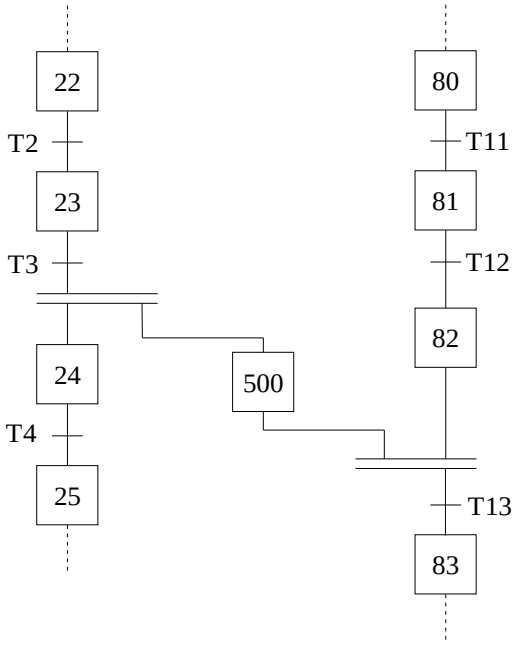


Qui la transizione T30 è valutata solo quando le fasi che partecipano alla sincronizzazione sono attivate (fasi 15, 20 e 25). In tal caso se la condizione T30 è abilitata, allora, la fase 30 può avere inizio. In alcune strutture si può avere l'esigenza di far accedere una parte dell'SFC ad una risorsa condivisa. Tale risorsa, essendo condivisa, è quindi accessibile anche da altri rami dell'SFC che vi accedono e necessita quindi di un semaforo che realizzi la mutua esclusione:



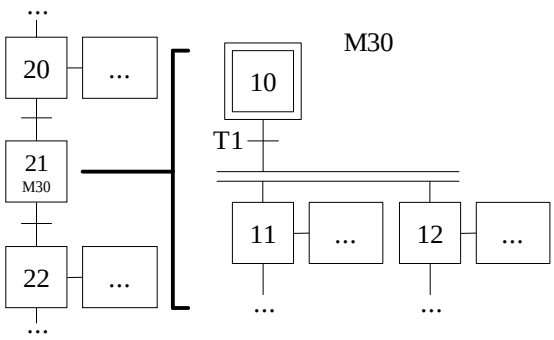
(notare che le condizioni T10 e T20 devono risultare mutuamente esclusive) Le fasi 80 e 30 sono le fasi intermedie di due cicli che accedono ad una risorsa condivisa, l'accesso è in questo schema SFC esclusivo per ogni ramo di SFC. La condizione T10 è valutata quando la fase 80 e la fase iniziale 100 sono attive (mediante struttura di

sincronizzazione). Anche la condizione T20 è valutata quando la fase 30 e la fase iniziale 100 sono attive. Le condizioni T10 e T20 sono mutue ed esclusive per cui può essere abilitata una oppure l'altra ma mai scattano in contemporanea. Se è abilitata per prima la fase 80 il ramo a sinistra può continuare della fase 81 ed accederà lungo quel ramo alla risorsa condivisa. Se nel frattempo è abilitata anche la fase 80 quest'ultima non accede alla porzione di SFC a destra (e quindi alla risorsa condivisa) poichè il token posseduto dalla fase 100 è al momento finito nel ramo sinistro dell'SFC. Solo quando il ramo sinistro dell'SFC esce dalla sezione critica e prosegue nella fase 90 il ramo destro dell'SFC (se ancora abilitato) può anch'esso accedere alla risorsa condivisa (dopo aver atteso). L'ultima struttura classica assai usata nell'SFC realizza la sincronizzazione locale fra due rami di SFC:

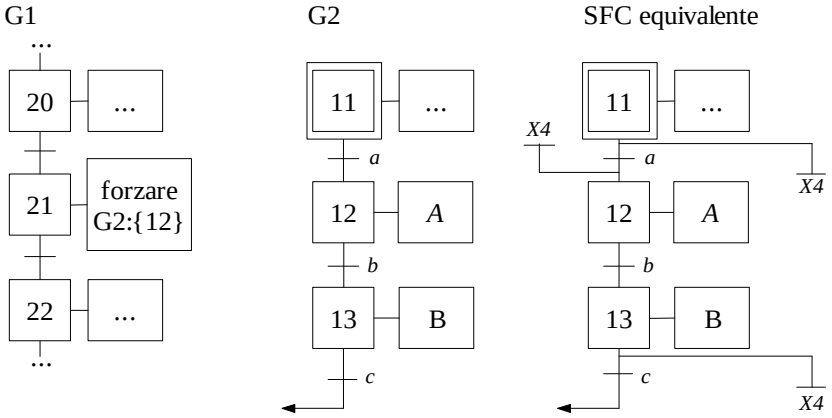


Quando il ramo destro dell'SFC in figura giunge nella fase 82 aspetta (sincronizzazione locale) che il ramo di sinistra superi la transizione T3 prima di andare oltre (T13 è infatti valutata solo quando le fasi 500 e 82 sono abilitate).

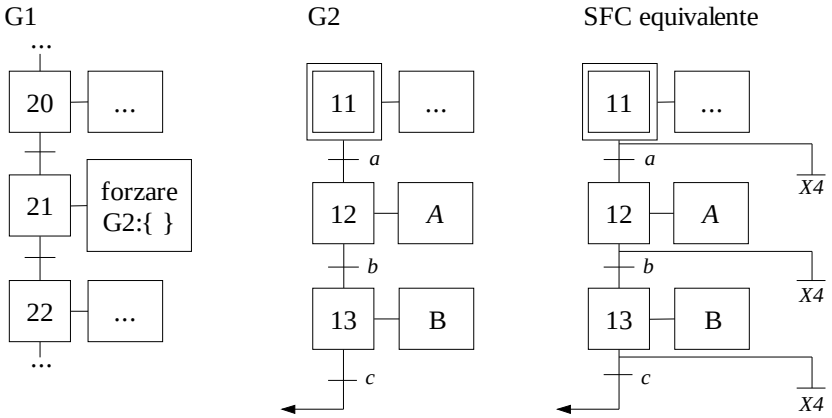
In alcuni casi, quando l'SFC assume proporzioni poco pratiche, è possibile avvalersi di mascrofasi. La macrofase sintetizza in un unica fase un intero SFC che è dunque un componente di quello principale. Per fare ciò si fa seguire al numero di fase il nome dell'SFC componente che può così essere osservato ed analizzato a parte:



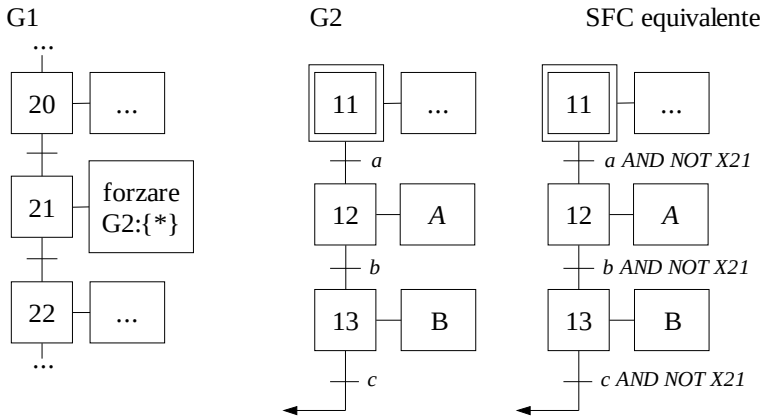
Oltre alle macrofasi la sintassi dell'SFC prevede anche le macroazioni. Le macroazioni impongono fra i vari SFC componenti un livello gerarchico utile poi al controllo complessivo del sistema di controllo. Alcune macroazioni che qui osserveremo sono: il forzamento, la sospensione ed il bloccaggio. Nel forzamento un SFC, ad esempio G1, forza la fase di un secondo SFC, ad esempio G2. La sintassi prevede l'aggiunta all'SFC G1 di una particolare fase:



Per ottenere il forzamento si può anche aggiungere una transizione superabile in uscita a tutte le fasi che non devono essere attive ed una transizione in ingresso alle fasi che, invece, devono esserle (vedi SFC equivalente).)
Nell'esempio, G1 forza G2 alla fase 12. Altra macroazione è la sospensione. La sintassi adoperata è simile al forzamento, in tal caso non viene indicata la fase da forzare che è così lasciata in bianco. La sospensione è un caso particolare di forzamento. Per ottenere la sospensione è sufficiente collocare una transizione superabile in uscita a tutte le fasi:

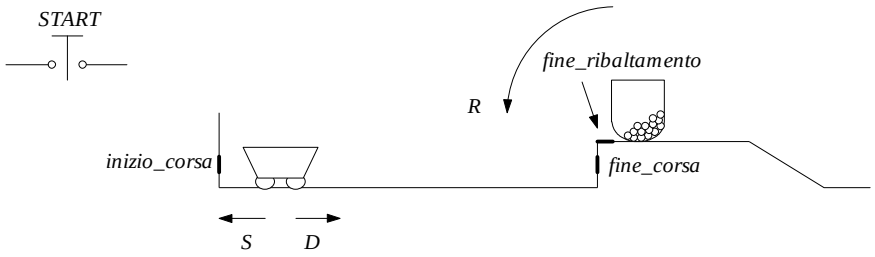


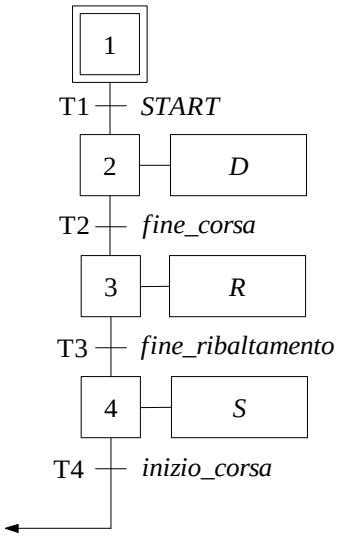
Un utile macroazione assai usata è la macroazione di bloccagio (nella maggior parte dei sistemi di controllo deve sempre essere disponibile un pulsante di emergenza per il bloccaggio dell'azione in corso). Il bloccaggio di un SFC da parte di un altro è una particolare azione di forzamento in cui la condizione è espressa dal simbolo *. E' ottenibile aggiungendo alle transizioni dell'SFC da controllare una condizione negata del segnale di emergenza.



Esempio: carrello trasportatore

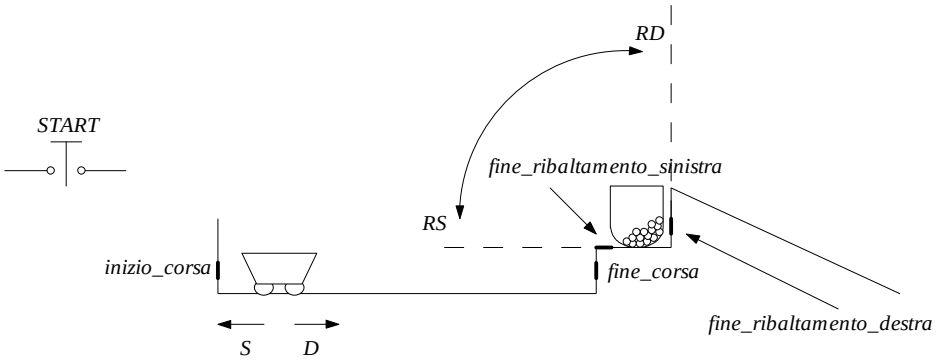
Descrizione – Quando richiesto da un operatore il carrello (azionato quindi da un pulsante di *start*) si sposta all'estrema destra di un binario finchè un sensore di *fine corsa* non ne individua la posizione, viene caricato mediante il ribaltamento di un serbatoio e si riporta all'estrema sinistra del binario. Nel sistema sono dunque presenti tre sensori: due rilevano la posizione di inizio e fine corsa mentre uno indica l'avvenuto ribaltamento del serbatoio. Il carrello può essere attuato da due segnali: il segnale *S* sposta il carrello verso sinistra; il segnale *D* sposta il carrello verso destra. Il serbatoio è invece ribaltato dal comando *R*.

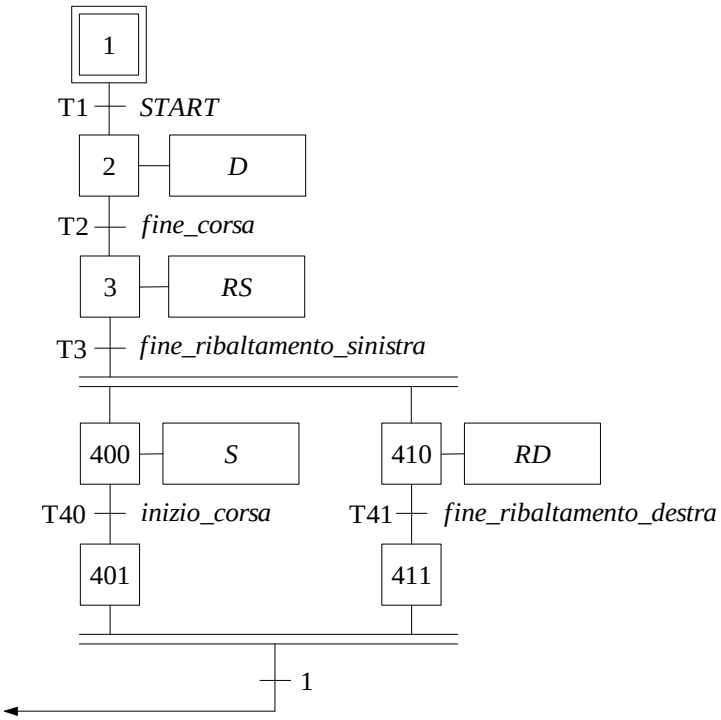




La descrizione del processo da controllare potrebbe lasciare sospesi dei dubbi sulle specifiche di progetto. Nella suddetta descrizione si assume che il serbatotio ritorni da solo nella posizione originaria. Se così non fosse occorre allora modificare leggermente l'SFC mostrato ed includere questa eventualità. Per fare ciò supponiamo, quindi, che il serbatotio risponda ai comandi: RS che ribalta il serbatotio a sinistra ed RD che ribalta il serbatotio a destra. Occorre poi prevedere l'aggiunta di un ulteriore sensore (oltre a quello già presenta di fine rotazione) che quindi chiameremo in tal caso: *fine ribaltamento sinistra* il sensore già presente e *fine ribaltamento destro* il sensore che segnala l'avvenuta rotazione verso destra. E' poi possibile seguire due approcci per il ribaltamento verso destra del serbatoio: in un approccio sequenziale possiamo pensare di azionare prima il comando per far tornare dietro il carrello (quindi verso sinistra) e poi ribaltare verso destra il serbatoio (dando quindi la priorità al carrello che una volta

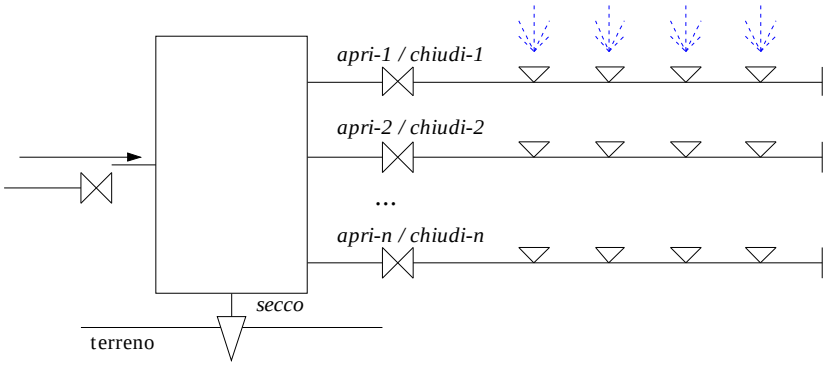
arrivato a fine corsa sarà svuotato da un operatore, è comunque possibile invertire le priorità); un secondo approccio è invece più interessante poichè prevede l'uso della struttura di parallelismo, in altre parole mentre il carrello è richiamato verso sinistra il serbatoio viene contemporaneamente ribaltato verso destra.

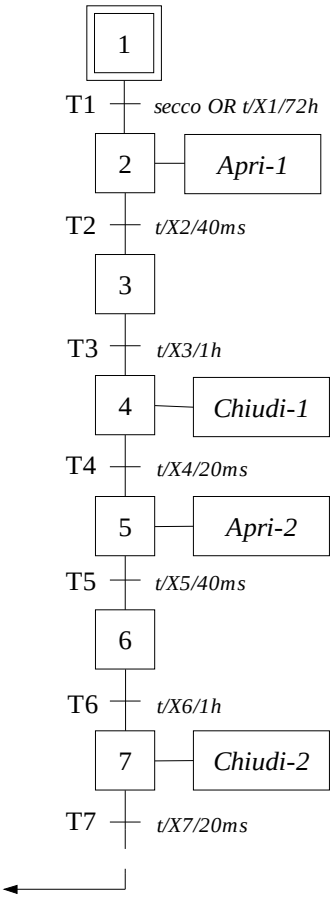




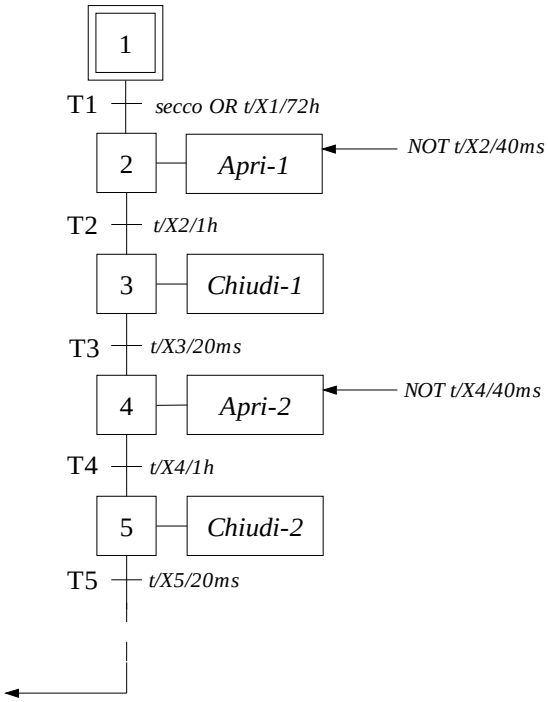
Esempio: il sistema di irrigazione

Descrizione – In un sistema di irrigazione l'avvio del ciclo di irrigazione è comandato da un sensore che genera un segnale *secco* quando l'aridità del terreno raggiunge una certa percentuale. Nonostante il suddetto segnale il ciclo di irrigazione può comunque avere inizio se sono passate 72 ore (3 giorni) consecutive senza alcun segnale di *secco*. Alla rete di irrigazione, inoltre, appartengono *n* rami e ciascuno di essi può essere abilitato da una valvola che ne comanda l'apertura e la chiusura. A causa della bassa pressione a disposizione i rami sono abilitati all'irrigazione, in sequenza, uno per volta. Ogni ramo eroga il terreno circostante per un ora prima di passare il turno al ramo successivo. L'i-esima elettrovalvola è comandata in apertura dal segnale *apri-i* che deve durare almeno 40ms ed è poi chiusa dal segnale *chiudi-i* che deve durare almeno 20ms.





Una diversa versione dell'SFC appena visto ci permette di avere una maggiore compattezza poichè fa uso di azioni limitate nel tempo e di meno fasi:

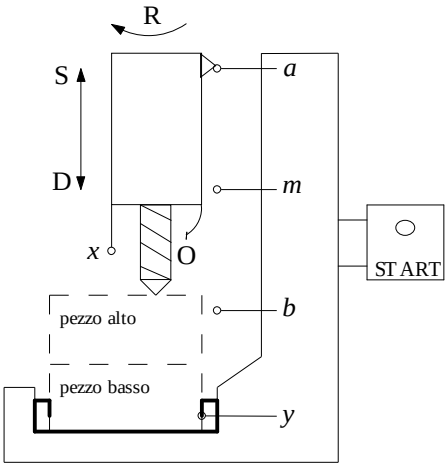


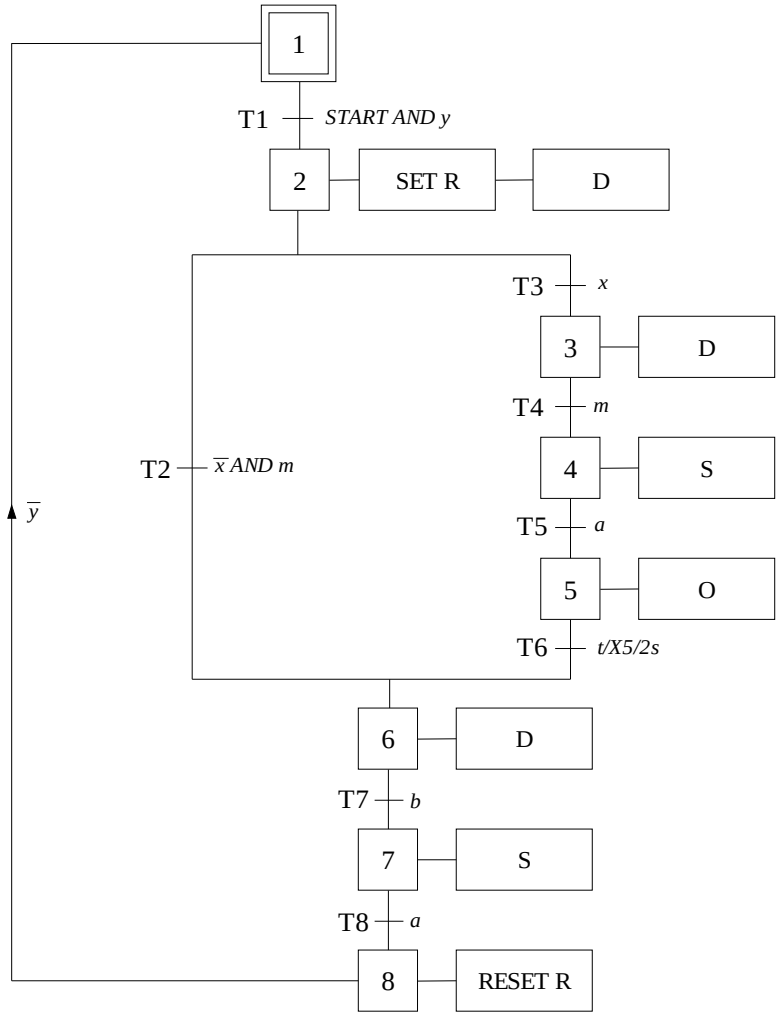
Esempio: il trapano automatico

Descrizione – Nella stazione di trapanatura automatica sono ammessi due tipologie di pezzi: pezzo alto e pezzo basso. Il contatto/sensore di altezza *x* segnale la tipologia del pezzo in concomitanza con un altro segnale, il segnale *m* di corsa media. La stazione, infatti, fa scorrere il trapano su un binario verticale. Su quest'ultimo sono distribuiti diversi sensori di posizione: *a* è un segnale di fine corsa; *b* è un segnale di inizio cora; *m* è un segnale di corsa media. Ai piedi della stazione automatica è poi possibile collocare uno dei due suddetti tipi di pezzi. Quando il pezzo è alla base della stazione un contatto/sensore *y* ne rileva la presenza. Le modalità di lavorazione sono diverse da pezzo a pezzo. Il pezzo più basso è forato in unica mandata; l'altro, invece, quello alto, è forato in due tempi (per dare la possibilità alla miccia di raffreddarsi). Quando il pezzo alto è collocato alla base del trapano automatico la foratura interessa la prima metà del pezzo, quindi la punta si solleva e viene spruzzato olio di raffreddamento sulla miccia. A questo punto la foratura del pezzo alto può essere completata. Il riconoscimento del tipo di pezzo avviene in questo modo:

- un pezzo viene considerato alto se il segnale *x* arriva prima del segnale *m*;
- un pezzo viene considerato basso se il segnale *m* arriva prima del segnale *x*;

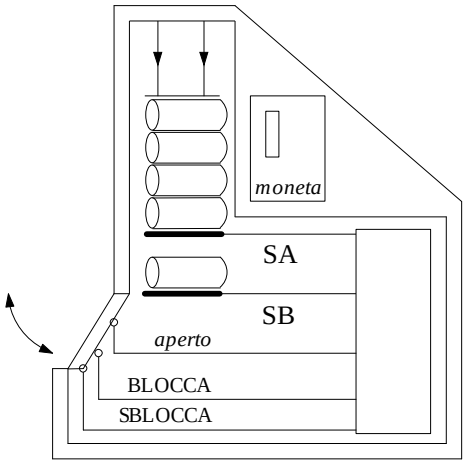
Il ciclo di lavorazione avviene se un operatore preme il tasto START e se uno dei due tipi di pezzi è nella sede di lavorazione. Per comandare gli attuatori di campo il sistema di controllo ha a disposizione i seguenti segnali: S comanda la salita del trapano, D comanda la discesa del trapano; R attiva la rotazione della miccia; O attiva la pompa per lubrificare e raffreddare la miccia.

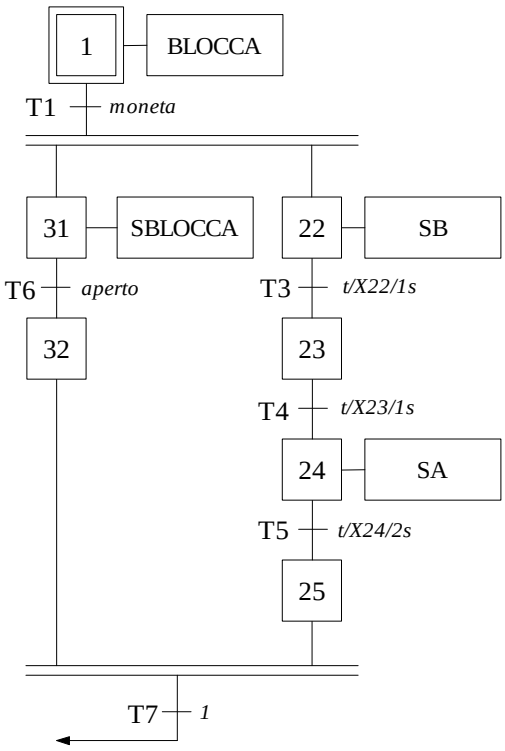



1
T1
START AND y
2
SET R
D
T3
x
3
D
T4
m
T2
x̄ AND m
4
S
ȳ
T5
a
5
O
T6
t/X5/2s
6
D
T7
b
7
S
T8
a
8
RESET R

Esempio: il distibutore automatico di bibite

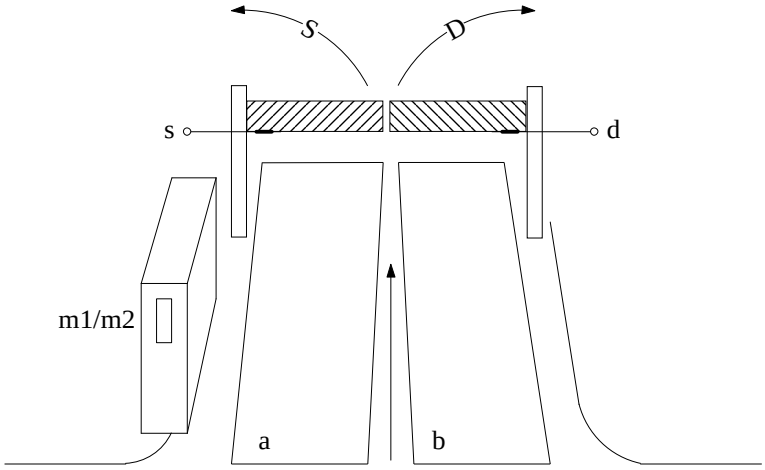
Descrizione – In un distributore automatico di bibite si hanno a disposizione i seguenti segnali di ingresso: *moneta* indica la presenza del quantitativo esatto di monete; *aperto* indica se il vano per prelevare la bibita è aperto. I segnali di controllo, invece, sono: BLOCCA e SBLOCCA che aiscono entrambi sulla porta del vano per prelevare la bibita; SA ed SB comandano due palette che servono a fare scendere le bibite una alla volta così come è rappresentato in figura:

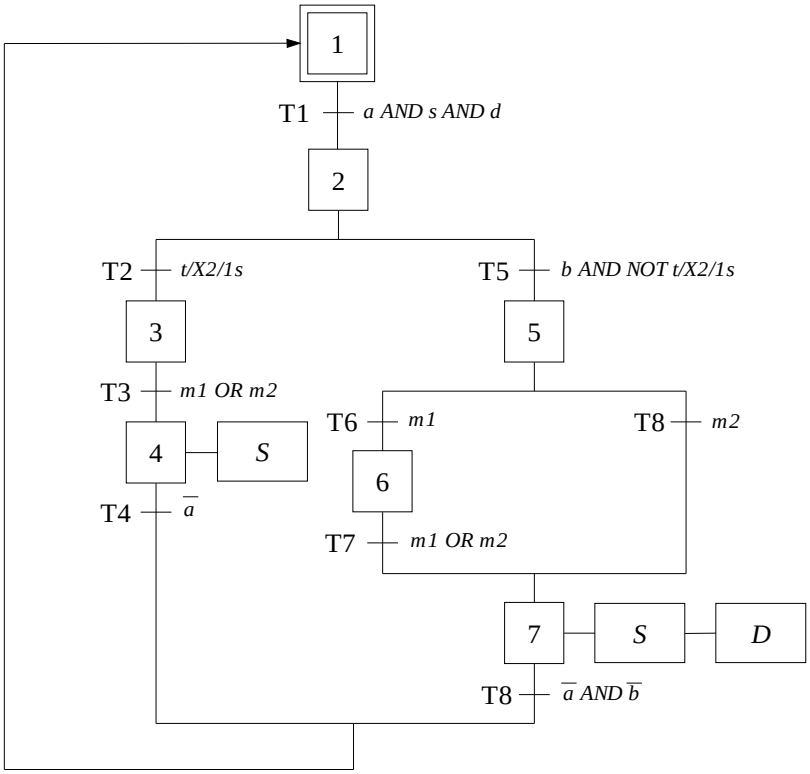




Esempio: il parcheggio automatico

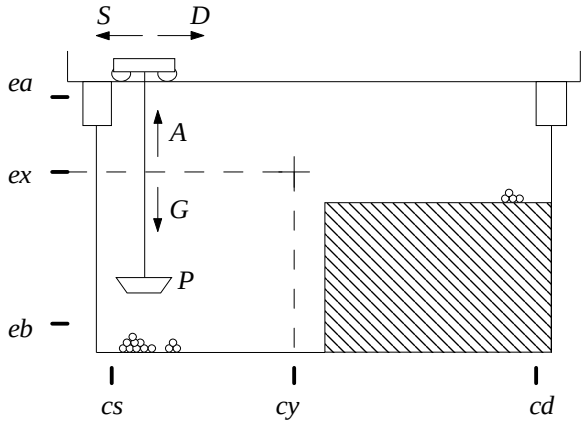
Descrizione – In un parcheggio a pagamento viene installato un sistema automatico per l'accesso. La barriera di ingresso al parcheggio si compone di due sbarre così comandate: in presenza di una macchina entrambe le sbarre vengono alzate; in presenza di una moto viene sollevata una sola sbarra. I comandi S e D alzano rispettivamente la sbarra sinistra e destra dell'ingresso, quando queste non sono comandate una molla di ritorno le riporta nelle posizione originaria. Alla sinistra dell'ingresso è presente inoltre una gettoneria, l'ingresso per le macchine è di 2€ (segnale *m2*) mentre per le moto è di 1€ (segnale *m1*). La presenza di una moto o di una macchina viene rilevata mediante due placche metalliche o celle di carico che generano poi i segnali *a* e *b* quando attivate. I comandi a disposizione del dispositivo automatico sono S per aprire la sbarra di sisnistra e D per aprire quella di destra. Non appena non viene più rilevata la presenza di una vettura le sbarre sono chiuse (i segnale *s* e *d* indicano la completa chiusura delle sbarre). Si assume che se un veicolo si appoggia prima sulla placca A deve poi impegnare la placca B entro 1s per essere considerato un auto, in caso contrario il dispositivo assumerà che il veicolo sulla placca A sia una moto.

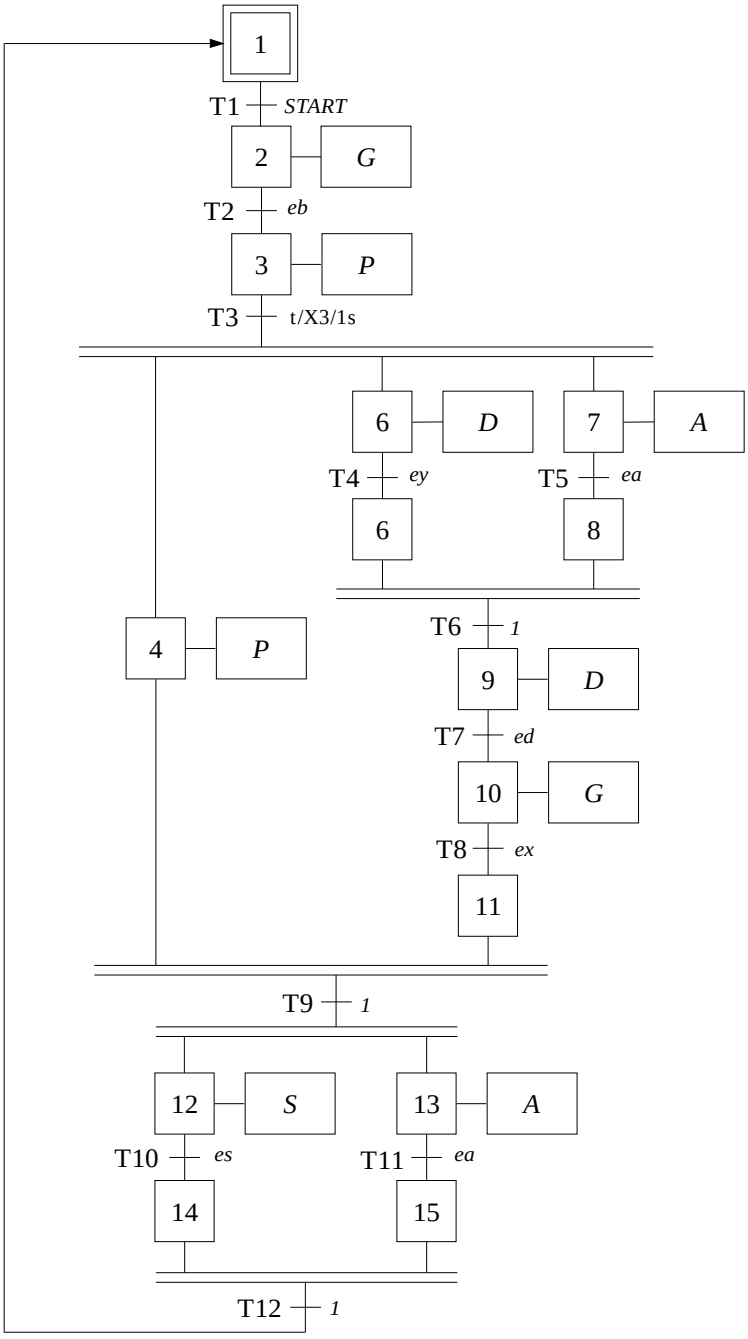



1
T1
a AND s AND d
2
T2
t/X2/1s
T5
b AND NOT t/X2/1s
3
5
T3
m1 OR m2
T6
m1
T8
m2
4
S
6
T4
$\overline{a}$
T7
m1 OR m2
7
S
D
T8
$\overline{a}$ AND $\overline{b}$

Esempio: l'elettrocalamita comandata

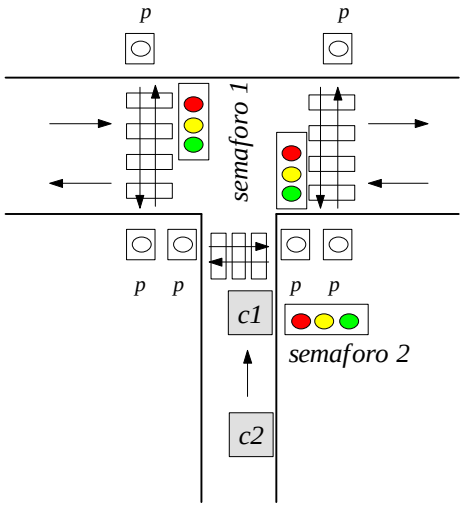
Descrizione – Nell'elettrocalamita comandata automatica si assiste ad una programmazione parallela delle fasi dovuta alla presenza di un ostacolo da evitare. L'elettrocalamita è azionata dal segnale *P* che attira a se i pezzi metallici situati in un intorno dell'area di lavoro. Il carrello puo spostare l'elettrocalamita (trascinandola con se) a sinistra ed a destra (segnali *S* e *D*). Inoltre, l'elettrocalamita può essere regolata in altezza, mediante una carrucola elettrica fissata al carrello, tramite i segnali *A* (per far salire l'elettrocalimita) e *G* (per far scendere l'elettrocalimita). I pezzi di metallo catturati dall'elettrocalamita in prossimità delle coordinate *(cs;eb)* devono essere rilasciati in *(cd;ex)*. Le coordinate dello spigolo da evitare in fase di azionamento del carrello *(cy;ex)* sono state già aumentate opportunamente per evitare collisioni tra quest'ultimo e l'elettrocalamita. Si assume che il carrello, quando azionato da un segnale di *START*, si trovi in *(cs;ea)*.

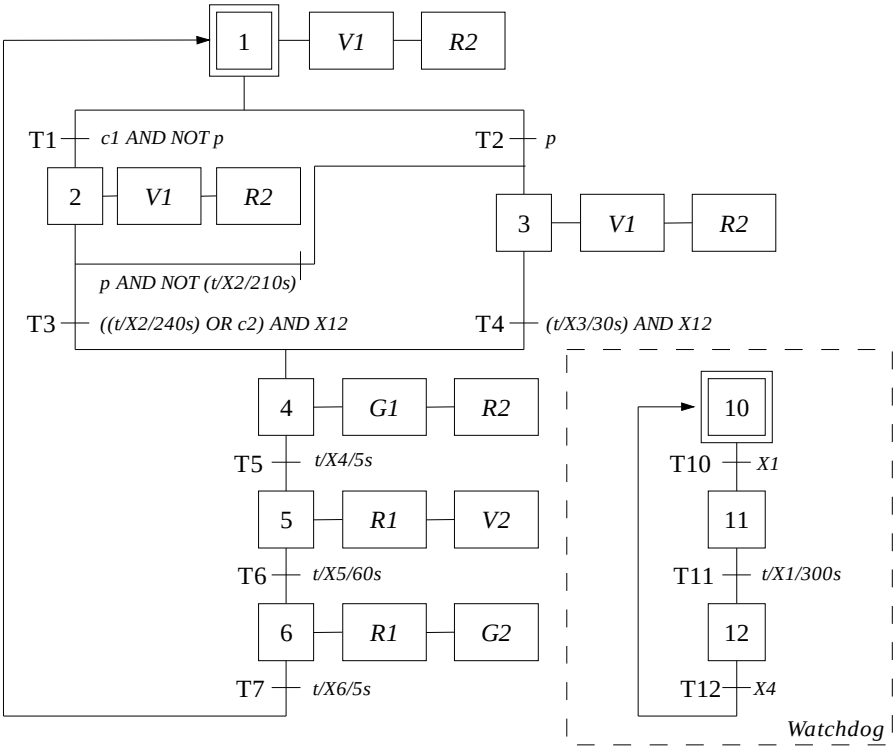



1
T1
START
2
G
T2
eb
3
P
T3
t/X3/1s
6
D
7
A
T4
ey
T5
ea
6
8
T6
1
4
P
9
D
T7
ed
10
G
T8
ex
11
T9
1
12
S
13
A
T10
es
T11
ea
14
15
T12
1

Esempio: il sistema semaforico

Descrizione – Si vuole controllare il flusso di traffico cittadino (vetture e pedoni) mediante un sistema semaforico intelligente. Il sistema è installato in prossimità di un incrocio fra una strada principale (ad elevata densità di flusso) ed una secondaria (con bassa densità di flusso). Il sistema deve inoltre considerare l'eventuale richiesta dei pedoni che intendono attraversare la strada. Sulla strada con maggior flusso di traffico il semaforo deve essere normalmente verde. Se tale luce rimane attiva per almeno 300 secondi il normale ciclo semaforico (luve verde, luce gialla per 5 secondi, luce rossa per 60 secondi) può essere attivato. La presenza di due sensori nell'asfalto (*c1* e *c2*) della strada con bassa densità di flusso permette di stabilire la presenza di una coda di auto in attesa. Per la strada secondaria il semaforo è normalmente rosso, se quindi *c1* e *c2* segnalano la presenza di vetture il ciclo semaforico deve essere attivato (luce verde per 300 secondi alla strada principale e luce verde alla strada secondaria solo dopo averne atteso il ciclo semaforico). Il ciclo si deve attivare anche solo quando il segnale *c1* indica la presenza di una sola auto, in tal caso la commutazione deve avvenire dopo 240 secondi. Ed inoltre, se è presente una richiesta di attraversamento dei pedoni (segnale *P*) il sistema semaforico deve commutare dopo 30 secondi e permettere a questi l'attraversamento.




1
V1
R2
T1
c1 AND NOT p
T2
p
2
V1
R2
3
V1
R2
p AND NOT (t/X2/210s)
T3
((t/X2/240s) OR c2) AND X12
T4
(t/X3/30s) AND X12
4
G1
R2
T5
t/X4/5s
5
R1
V2
T6
t/X5/60s
6
R1
G2
T7
t/X6/5s
10
T10
X1
11
T11
t/X1/300s
12
T12
X4
Watchdog