

Elaborato ASM

Architettura degli elaboratori

Nicola Serlonghi

VR409046



A.A. 2016/2017

Università degli Studi di Verona

Indice

Descrizione generale del progetto	3
Variabili e registri utilizzati	4
Funzioni implementate e modalità di passaggio e restituzione dei valori	4
Pseudo codice e diagramma di flusso	5

Descrizione generale del progetto

Il programma da noi implementato permette il monitoraggio in maniera ottimizzata di un impianto chimico industriale mediante l'utilizzo di Assembly. Esso riceve come input il pH di una soluzione contenuta in un serbatoio, e fornisce in uscita lo stato della soluzione (compatibilmente con delle soglie pre-impostate) in termini acido (A), basico (B) e neutro (N). Il sistema deve sempre portare la soluzione allo stato neutro e se essa lo è già deve semplicemente rimanere in quello stato. Se invece abbiamo una sostanza che è acida o basica da più di 5 cicli di clock provvederemo al sesto ciclo ad aprire la relativa valvola, BS se ci troviamo di fronte ad una soluzione acida o AS se siamo allo stato basico, per riportare la soluzione allo stato neutro.

Il programma è stato realizzato con una funzione Assembly che gestisce il tutto in modo da ottimizzare il più possibile rispetto ad un codice C e far rendere quindi la macchina che lo esegue al massimo delle sue capacità.

INPUTS

Vengono letti da un file che deve contenere su ogni riga i seguenti valori nel medesimo ordine: INIT, RESET, PH

Di seguito viene descritto ogni valore:

- INIT [1]: valore binario, quando vale 1 il sistema è acceso; quando vale 0 il sistema è spento e deve restituire 0.
- RESET [1]: quando posto a 1 il controllore deve essere resettato, ovvero tutte le uscite devono essere poste a 0 e il sistema riparte.
- PH [3]: valore del pH misurato dal rilevatore. Il range di misura è compreso tra 0 e 14 con risoluzione di 0,1. Il valore è espresso in decime di pH e sempre riportato in 3 cifre.

OUTPUTS

Vengono scritti su un file ed ogni riga contiene i seguenti valori nel medesimo ordine: ST, NCK, VLV

Di seguito viene descritto ogni valore:

- ST [1]: indica in quale stato si trova la soluzione al momento corrente (acida - A, basica - B, neutra - N).
- NCK [2]: indica il numero di cicli di clock trascorsi nello stato corrente.
- VLV [2]: indica quale valvola aprire per riportare la soluzione allo stato neutro nel caso in cui la soluzione si trovi da più di 5 cicli di clock in stato acido (BS) o basico (AS).

Consideriamo invece come valori di soglia:

- $\text{pH} < 6.0$: Acido
- $6.0 \leq \text{pH} \leq 8.0$: Neutro
- $\text{pH} > 8.0$: Basico

Variabili e Registri utilizzati

Nel nostro programma abbiamo deciso di cercare di non andare ad utilizzare alcuna variabile in modo da renderlo il più performante possibile. Siamo riusciti nel nostro intento e siamo andati ad utilizzare solamente i 4 registri `eax`, `ebx`, `ecx`, `edx` nel seguente modo:

- `eax`: questo registro contiene l'indirizzo di memoria della prima posizione dell'array di `char` che rappresenta il buffer di input. All'interno del programma viene poi incrementato in modo da poter accedere ad ogni posizione di esso e leggere tutti i dati forniti in input correttamente.
- `ebx`: questo registro contiene l'indirizzo di memoria della prima posizione dell'array di `char` che rappresenta il buffer di output. All'interno del programma viene poi incrementato in modo da poter accedere ad ogni posizione di esso ed inserire al suo interno tutti i dati da fornire in output correttamente.
- `ecx`: questo registro contiene il valore del NCK a cui siamo arrivati in quel momento, viene quindi incrementato o azzerato in base agli input che riceve la macchina.
- `edx`: questo registro viene utilizzato per memorizzare lo stato in cui era la macchina nel ciclo di clock precedente e poter dunque confrontarlo con lo stato attuale e poter aprire le valvole (AS o BS) in maniera corretta.

Funzioni implementate e modalità di passaggio e restituzione dei valori

Nel nostro programma abbiamo utilizzato solamente una funzione Assembly che va a svolgere tutte le operazioni necessarie affinché esso funzioni in maniera corretta e rispetti le specifiche date. La funzione da noi creata è di tipo `void`, non restituisce quindi alcun valore, ma vengono tutti passati per riferimento. Essa a dunque due parametri, `bufferin` e `bufferout_asm`, che sono due puntatori di tipo `char` che puntano rispettivamente al primo valore dell'array `bufferin` e al primo valore dell'array `bufferout_asm`. Il primo viene quindi utilizzato per caricare in ingresso i valori necessari mentre il secondo per fornire in output i valori calcolati.

Diagramma di flusso e pseudo codice

Quando la funzione viene lanciata per prima cosa andiamo a salvare i valori già presenti nei registri ebp, eax, ebx, ecx ed edx nello stack dopo di che procediamo pulendo i registri che andremo ad utilizzare in seguito. Ora possiamo iniziare a lavorare sul file di input e lo facciamo controllando se il file è vuoto o meno. Se il controllo ha esito positivo usciamo dalla funzione senza fare nulla, se invece ha esito negativo entriamo nel ciclo e cominciamo a svolgere le seguenti operazioni per ogni riga presente nel file:

- Andiamo a controllare se la macchina è accesa, se lo è passiamo al punto successivo, se non lo è stampiamo in output -,- -,- - e procediamo alla riga successiva.
- Procediamo ora ad incrementare il contatore NCK d'una unità usando le dovute precauzioni per i casi limite che si possono andare a creare.
- Ora andremo a stabilire tramite gli opportuni controlli sui valori relativi al pH attuale se la soluzione si trova nello stato basico, acido e neutro e procederemo a svolgere le opportune operazioni in base allo stato trovato.
- Per i casi acido e basico come prima cosa viene confrontato se lo stato attuale è uguale a quello precedente, in caso positivo procediamo ad azzerare l'NCK e poi si procede come in caso negativo a controllare se sono già stati raggiunti i 5 cicli di clock ($NCK \geq 5$). Se sono stati raggiunti si stampa lo stato del sistema aprendo l'opportuna valvola, mentre se non sono stati raggiunti si stampa lo stato del sistema con le valvole chiuse.
- In caso la sostanza sia neutra si svolgono tutte le operazioni indicate al punto precedente tranne che per il controllo del raggiungimento dei 5 cicli di clock che qui non è necessario.
- Una volta fatto tutto ciò vengono incrementati i registri necessari per ciclare in maniera corretta e andiamo a controllare se siamo arrivati a fine file, se non lo siamo torniamo a svolgere tutte le operazioni da primo punto sulla riga successiva. Se invece ci troviamo alla fine del file, ripristiniamo il contenuto di tutti i registri salvati precedentemente nello stack e la nostra funzione termina.

Alla pagina successiva è presente il diagramma di flusso della funzione appena descritta.

