Laboratorio di Architettura degli Elaboratori

A.A. 2017/18

Elaborato Assembly

Si ottimizzi un codice in linguaggio C che controlla un dispositivo per la gestione intelligente del consumo di energia elettrica all'interno di un sistema domotico mediante l'uso di Assembly inline. Il dispositivo riceve in ingresso lo stato acceso/spento di un numero finito di dispositivi di cui è noto il consumo istantaneo a priori, e fornisce in uscita la fascia di consumo ad ogni ciclo di clock. Qualora l'assorbimento istantaneo sia superiore al limite di 4.5kW per più di 5 cicli di clock consecutivi, il sistema deve disattivare l'interruttore generale. Al fine di prevenire questa situazione, il dispositivo può disattivare la lavatrice e la lavastoviglie (in questo ordine di priorità).

Inizialmente il sistema è sempre spento (**INT_GEN**=0) e gli interruttori della lavatrice (**INT_WM**) e della lavastoviglie (**INT_DW**) sono sempre non-armati (entrambi posti a 0). Fintanto che **INT_GEN**=0, il sistema rimane spento ed il dispositivo deve restituire 0 per tutti i bit di output. Dal momento in cui **RES_GEN** assume il valore 1: **INT_GEN**, **INT_DW** e **INT_WM** commutano a 1, il sistema si accende ed il dispositivo inizia a leggere lo stato acceso/spento di tutti i carichi (**LOAD**). Il dispositivo deve restituire in uscita la fascia di consumo istantaneo: **F1**<=1.5kW, 1.5kW<**F2**<=3kW, 3kW<**F3**<=4.5kW, **OL**>4.5kW. Nel caso in cui il sistema rimanga in overload (**OL**) per almeno 4 cicli di clock, il dispositivo commuta **INT_DW** a 0 e, fintanto che non viene riarmato (**RES_DW**=1), il carico relativo alla lavastoviglie deve essere ignorato; qualora ciò non sia sufficiente a uscire dallo stato **OL**, al ciclo di clock successivo (5° ciclo in **OL**) il dispositivo commuta **INT_WM** a 0 e, fintanto che non viene riarmato (**RES_WM**=1), il carico relativo alla lavatrice deve essere ignorato. Nel caso in cui il sistema permanga in **OL**, al successivo ciclo di clock (6° ciclo in **OL**) il dispositivo commuta **INT_GEN** a 0 ed il sistema si spegne.

Il programma deve essere lanciato da riga di comando con due stringhe come parametri, la prima stringa identifica il nome del file `.txt` da usare come input, la seconda quello da usare come output:

```
$ ./controller testin.txt testout.txt
```

Il programma deve leggere il contenuto di `testin.txt` contenente in ogni riga i seguenti valori:

```
RESET-LOAD
```

- **RESET** [3]: contiene la sequenza dei comandi di riarmo degli interruttori; nell'ordine **RES_GEN**, **RES_DW** e **RES_WM** (senza spazi)
- **LOAD** [10]: stato di accensione (1=ON, 0=OFF) dei carichi elettrici. Ogni carico ha un suo consumo istantaneo associato. Il carico complessivo del circuito è dato dalla somma di tutti i carichi accesi contemporaneamente. L'ordine ed il consumo di ogni carico è come da tabella:

Forno	Frigo	Aspira-polvere	Phon	Lava-stoviglie	Lava-trice	4xlamp 60W	4xlamp 100W	HI-FI	TV
2 kW	300 W	1200 W	1 kW	2 kW	1800 W	240 W	400 W	200 W	400 W

Il programma deve restituire i risultati del calcolo in `testout.txt` in cui ogni riga contiene:

```
INT-TH
```

- **INT** [3]: indica lo stato di attivazione (1=ON, 0=OFF) degli interruttori; in ordine **INT_GEN**, **INT_DW** e **INT_WM** (senza spazi)
- **TH** [2]: indica la fascia di consumo istantanea secondo la seguente codifica:
  - **F1** <= 1.5kW
  - 1.5kW < **F2** <= 3kW
  - 3kW < **F3** <= 4.5kW
  - **OL** > 4.5kW

Nel caso in cui la macchina sia spenta, il controllore deve restituire la stringa “000-00”.

Assieme al presente documento sono forniti:

- un file `controller.c` contenente il sorgente C che dovrà essere editato per inserire la parte assembly. **Il file può essere modificato solo nelle parti segnalate tramite commento!** Files contenenti modifiche esterne a queste sezioni saranno penalizzati e potranno comportare anche un voto insufficiente. È possibile utilizzare assembly inline o funzioni assembly richiamate da C, in quest’ultimo caso i files creati devono essere consegnati all’interno della cartella principale.
- un file `testin.txt` di esempio. In fase di valutazione sarà utilizzato un file di test diverso.
- un file `trueout.txt` da utilizzare per verifica del corretto funzionamento del programma. Sarà sufficiente usare il comando `diff testout.txt trueout.txt` per vedere eventuali differenze tra i due files (quello generato dal programma e quello corretto).

In fase di correzione, saranno valutati meglio progetti che ottimizzeranno meglio il codice, ovvero programmi che dimostreranno un minore tempo di esecuzione (a parità di hardware). Durante l’esame orale è possibile che venga richiesto di operare delle modifiche al codice sul momento.

Modalità di consegna:

Tutto il materiale va consegnato elettronicamente tramite procedura guidata sul sito Moodle del corso. Indicativamente 15 giorni prima della data di consegna sarà attivata una apposita sezione denominata "consegna_ASM_mmmaaaa" (mmm=mese, aaaa=anno); accedendo a quella pagina sarà possibile effettuare l'upload del materiale. La consegna del materiale comporta automaticamente l'iscrizione all'appello orale.

Il giorno seguente la data ultima di consegna (entro le ore 12.00) verrà pubblicato sul sito Moodle del corso il calendario provvisorio degli orali; in caso di impossibilità a presenziare alla discussione orale nell'orario assegnato è necessario comunicarlo al docente via email entro le successive 24 ore. Entro la sera del giorno seguente sarà pubblicato il calendario definitivo.

Materiale da consegnare:

Laboratorio di Architettura degli Elaboratori

A.A. 2017/18

Si richiede ad ogni gruppo di caricare un singolo archivio **.tgz** denominato `asm_cognome1_nome1_cognome2_nome2.tgz`[1] contenente tutti i files di seguito elencati senza sottocartelle:

1. Il file `controller.c` modificato, contenente la versione definitiva del progetto da usare per i test. Non rinominare il file fornito ma modificarlo opportunamente
2. Tutti gli altri files sorgenti necessari al progetto (ad esempio funzioni assembly)
3. Makefile per la compilazione (non fornito, ogni gruppo deve scrivere il proprio)
4. Un file `Relazione.pdf` con una relazione del progetto che affronti nel dettaglio almeno i seguenti punti:
    - le variabili utilizzate e il loro scopo;
    - le modalità di passaggio/restituzione dei valori delle funzioni create;
    - il diagramma di flusso o lo pseudo-codice ad alto livello del codice prodotto;
    - la descrizione delle scelte progettuali effettuate

Si ricorda che è possibile effettuare più sottomissioni, ma ogni nuova sottomissione cancella quella precedente. Ogni gruppo deve consegnare una sola volta il materiale, ovvero un solo membro del gruppo deve effettuare la sottomissione!

[1] Per generare correttamente un file filename.tgz in linux seguire la seguente procedura:

- posizionare tutti i files da comprimere in una cartella
- rinominare la cartella “filename”
- uscire dalla cartella e lanciare il comando `tar cvfz filename.tgz filename`