

Prova pratica di Calcolatori Elettronici

C.d.L. in Ingegneria Informatica, Ordinamento DM 270

20 luglio 2022

1. Siano date le seguenti dichiarazioni, contenute nel file `cc.h`:

```
struct st {
    long vv2[4];
    char vv1[4];
};
class cl {
    st s;
public:
    cl(char v[]);
    void elab1(st& ss, int d);
    void stampa()
    {
        for (int i = 0; i < 4; i++)
            cout << (int)s.vv1[i] << ' ';
        cout << '\t';
        for (int i = 0; i < 4; i++)
            cout << s.vv2[i] << ' ';
        cout << endl;
        cout << endl;
    }
};
```

Realizzare in Assembler GCC le funzioni membro seguenti.

```
cl::cl(char v[])
{
    for (int i = 0; i < 4; i++) {
        s.vv1[i] = s.vv2[i] = v[i];
    }
}
void cl::elab1(st& ss, int d)
{
    for (int i = 0; i < 4; i++) {
        if (d >= ss.vv2[i])
            s.vv1[i] -= ss.vv1[i];
        s.vv2[i] = i - d;
    }
}
```

2. Colleghiamo al sistema delle periferiche PCI di tipo **ce**, con vendorID **0xedce** e deviceID **0x1234**. Ogni periferica **ce** usa 8 byte nello spazio di I/O a partire dall'indirizzo base specificato nel registro di configurazione BAR0, sia *b*.

La periferiche **ce** sono periferiche di ingresso in grado di generare richieste di interruzione. Ciascuna periferica contiene un certo numero di canali, numerati a partire da 0, ciascuno in grado di operare indipendentemente dagli altri. Questo permette di avere più trasferimenti attivi contemporaneamente. Per attivare un canale è necessario settare il corrispondente bit nel registro di controllo (CTL). La periferica contiene un unico registro di ingresso, RBR, e un unico piedino per generare richieste di interruzione, condivisi tra tutti canali. Quando un canale attivo produce un nuovo valore e trova RBR libero, deposita il valore in RBR, scrive il numero del canale nel registro CHN e invia una richiesta di interruzione. Il registro RBR risulterà poi occupato fino a quando non verrà letto dal software.

Per conoscere il numero di canali che ciascuno dispositivo possiede si può leggere CHN quando CTL è 0. In ogni caso ciascuna periferica può avere al massimo **MAX_CHAN** canali.

I registri accessibili al programmatore sono i seguenti:

1. **CTL** (indirizzo *b*, 1 byte, lettura/scrittura): registro di controllo; il bit *i*-esimo permette di attivare (1) o disattivare (0) il canale *i*-esimo;
2. **CHN** (indirizzo *b* + 4, 1 byte, sola lettura): (CHannel Number) se la periferica ha inviato una richiesta di interruzione, il registro contiene il numero del canale che ha prodotto il valore contenuto in RBR; se non ci sono canali attivi, CHN contiene il numero di canali che la periferica possiede.
3. **RBR** (indirizzo *b* + 8, 1 byte, sola lettura): registro di ingresso;

L'interfaccia genera una interruzione se uno dei canali attivi ha depositato un nuovo valore in RBR. La lettura di RBR funge da risposta alla richiesta di interruzione: l'interfaccia non presenta nuovi valori in RBR e non cambia il contenuto di CHN fino a quando RBR non viene letto.

Vogliamo fornire all'utente una primitiva

```
void ceread(natl id, natl n, char *buf, natl quanti);
```

Il parametro *id* identifica una delle periferiche **ce** installate e il parametro *n* uno dei suoi canali. La primitiva permette di leggere una sequenza di *quanti* byte dal canale *n* della periferica *id*. Se il canale *n* è già occupato da un'altra richiesta, la primitiva attende che il canale si liberi. I byte letti saranno scritti a partire dall'indirizzo *buf*.

La primitiva abortisce il processo in caso di errore. È un errore tentare di accedere ad una periferica o a un canale che non esiste. Controllare eventuali problemi di Cavallo di Troia.

Modificare i file *io.s* e *io.cpp* in modo da realizzare la primitiva come descritto.