Prova pratica di Calcolatori Elettronici

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

14 settembre 2022

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

```
struct st1 { char vc[4]; }; struct st2 { int vd[4]; };
class cl {
        st1 c1; st1 c2;
        long v[4];
public:
        cl(char c, st2& s);
        void elab1(st1 s1, st2 s2);
        void stampa()
        {
                for (int i=0; i < 4; i++) cout << c1.vc[i] << ' '; cout << "\n";
                for (int i=0; i < 4; i++) cout << c2.vc[i] << ' '; cout << "\n";
                for (int i=0; i < 4; i++) cout << v[i] << ' '; cout << "\n\n";
        }
};
```

   Realizzare in Assembler GCC le funzioni membro seguenti.

```
cl::cl(char c, st2& s2) {
        for (int i = 0; i < 4; i++) {
                c1.vc[i] = c2.vc[i] = c;
                v[i] = s2.vd[i] - c2.vc[i];
        }
}
void cl::elab1(st1 s1, st2 s2) {
        cl cla('z', s2);
        for (int i = 0; i < 4; i++) {
                if (c2.vc[i] < s1.vc[i]) {
                        c1.vc[i] = cla.c2.vc[i];
                        v[i] = cla.v[i];
                }
        }
}
```

2. Colleghiamo al sistema delle periferiche PCI di tipo `ce`, con vendorID `0xedce` e deviceID `0x1234`. Ogni periferica `ce` usa 32 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 che operano in bus mastering, sono in grado di generare richieste di interruzione, e sono anche in grado di eseguire autonomamente la traduzione da indirizzi

virtuali a fisici. Ciascuna periferica contiene un certo numero di canali, numerati da 0 a `MAX_CHAN` $-$ 1, ciascuno in grado di operare independentemente dagli altri. Questo permette di avere più trasferimenti attivi contemporaneamente. Per attivare un trasferimento in bus mastering su un canale è necessario scrivere l'indirizzo di destinazione nei registri BMPTR_HIGH e BMPTR_LOW, la quantità di byte da trasferire nel registro BMLEN, e l'indirizzo della radice del TRIE da usare per la traduzione degli indirizzi nel registro BMCR3; a questo punto si può scrivere il numero del canale nel registro CHN, avviando coì l'operazione.

La periferica possiede un registro STS, di sola lettura, con un bit per ogni canale (il bit meno significativo corrisponde al canale 0). Ciascun bit vale i-esimo vale 1 o 0 a seconda se il corrispondente canale è attivo o disattivo. La periferica invia una richiesta di interruzione quando un qualunque bit passa da 1 a 0 (vale a dire, il corrispondente trasferimento è stato completato), ma solo se non c'è già una richiesta precedente in attesa di risposta. La lettura di STS funge da risposta alle richieste di interruzione.

I registri accessibili al programmatore sono i seguenti:

1. **CHN** (indirizzo b, 4 byte, lettura/scrittura): scrivendo i in questo registro si attiva il canale i-esimo, con i valori correnti di BMPTR e BMLEN; la periferica ignora la richiesta se il canale era già attivo, o se i non corrisponde a nessun canale;
2. **STS** (indirizzo $b + 4$, 4 byte, lettura): registro di stato (vedere sopra);
3. **BMCR3** (indirizzo $b + 8$, 4 byte, lettura/scrittura): indirizzo della radice del TRIE da usare per la traduzione degli indirizzi interessati dal trasferimento;
4. **BMPTR_HIGH** (indirizzo $b + 12$, 4 byte, lettura/scrittura): parte alta dell'indirizzo di destinazione del trasferimento;
5. **BMPTR_LOW** (indirizzo $b + 16$, 4 byte, lettura/scrittura): parte bassa dell'indirizzo di destinazione del trasferimento;
6. **BMLEN** (indirizzo $b + 20$, 4 byte, lettura/scrittura): numero di byte da trasferire;

La periferica usa il contenuto di BMPTR_HIGH, BMPTR_LOW, BMCR3 e BMLEN solo quando si scrive in CHN il numero di un canale non precedentemente attivo.

Vogliamo fornire all'utente una primitiva

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

Il parametro `id` identifica una delle periferiche `ce` installate. La primitiva permette di leggere da tale periferica una sequenza di `quanti` byte dal primo canale (in ordine di identificatore) non attualmente già attivo. Se tutti i canali sono attivi la primitiva attende che se ne liberi uno. I byte letti saranno scritti a partire dall'indirizzo `buf`. È un errore se il buffer non è accessibile in scrittura da livello utente o se la periferica richiesta non esiste. La primitiva abortisce il processo in caso di errore.

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