

## Esercizio DMA 1

Supponiamo di voler interfacciare la periferica Transmitter del calcolatore simulato da SPIM mediante la tecnica DMA. Lo scopo è di trasferire un certo numero di caratteri dalla memoria alla Console. Quindi dobbiamo realizzare, mediante DMA, un trasferimento di uscita, da memoria a periferica.

Ricordiamo che il simulatore SPIM non gestisce la simulazione di trasferimenti con DMA. Supporremo quindi, senza poter simulare davvero il tutto, che nella periferica ci siano le seguenti differenze rispetto a quanto visto fino ad ora. Riassumiamo quanto visto fino ad ora: per la periferica c'è un registro di stato, i cui ultimi 2 bit sono utilizzati per indicare 1) lo stato di ready / non ready. 2) lo stato in cui la periferica ha interrupt abilitato / interrupt disabilitato; inoltre c'è un registro dati. Infine l'indirizzo dei 2 registri è noto (nel caso di SPIM si ipotizza che il registro di stato del Transmitter si trovi mappato a 0xFFFF0008 e, ad esempio, il registro dati alla locazione dell'indirizzo di stato + 4. Ipotizziamo che l'indirizzo del registro di stato in questo esercizio sia 0x80000010. Ci saranno quindi le seguenti differenze, per gestire il trasferimento con tecnica DMA:

- nella periferica esiste anche un registro DMA\_MemoryAddress, che rappresenta la posizione in memoria dove mettere / prendere i dati (questo è il registro chiamato "registro Dove" nelle lezioni), si supponga questo registro periferica mappato all'indirizzo del registro di stato della periferica + 8;
- nella periferica esiste anche un registro DMA\_Length, che rappresenta il numero di dati da trasferire (questo è il registro chiamato "registro Quanti" nelle lezioni), si supponga questo registro periferica mappato all'indirizzo del registro di stato della periferica + 12;
- nel registro di stato della periferica, oltre ai 2 bit già visti, si hanno anche i seguenti bit:
  - DMA\_Direction # 3<sup>rd</sup> least significant bit, direzione del trasferimento: 1 = IN / 0 = OUT;
  - DMA\_Start # 4<sup>th</sup> least significant bit, partenza del trasferimento: 1 = trasferimento attivo / 0 = trasferimento disattivato;
  - DMA\_concluded # 5<sup>th</sup> least significant bit, 1 = trasferimento concluso / 0 = trasferimento non concluso;
  - DMA\_Sizeofdatum # 6<sup>th</sup> least significant bit, 0 = dato da 1 byte, 1 = dato da 4 byte.

NB

- La scrittura del valore "1" nel bit DMA\_Start avvia il trasferimento con DMA con i valori correnti di DMA\_MemoryAddress e DMA\_Length! Quindi questo bit va scritto per ultimo, dopo aver predisposto il trasferimento ovvero dopo aver predisposto i valori corretti nei registri periferica DMA\_MemoryAddress e DMA\_Length.
- La logica presente nella periferica, quando DMA\_Length diventa = 0, attiva una richiesta di interruzione; il gestore delle eccezioni nell'andare a cercare la causa dell'interruzione tra le varie periferiche, per questa periferica verificherà se sia = 1 il 5<sup>th</sup> least significant bit del registro di stato periferica, non il 1<sup>st</sup> least significant bit come invece farebbe per un trasferimento a controllo di programma.

Completare il file EsDMA.asm in modo da utilizzare il DMA per trasferire i dati (ATTENZIONE: non è ovviamente possibile provare il file su SPIM perché la tecnica DMA non è implementata).

## Esercizio DMA 2

Nell'esercizio precedente, per gestire correttamente il DMA, sarebbe necessario modificare anche un altro file. Specificare quale file bisognerebbe modificare, in che modo, e perché sia necessario modificarlo.

---

---

---

---

---

---

---

---

---

---

---

---

Quali sono in generale i vantaggi e gli svantaggi di utilizzare il DMA?

---

---

---

---

---

---

---

---

---

---

---

---