

# Esercizi 04

## Input / Output

### Chapter 10

1) Un dispositivo di conversione analogico-digitale ADC, gestito ad interrupt, acquisisca dati alla frequenza di 2 KHz. La durata della routine di servizio delle sue interruzioni sia  $S_{ADC} = 200 \mu s$ .

Si indichi la percentuale  $P$  di tempo di CPU impegnata per il trasferimento dei dati (si trascuri la predisposizione iniziale del dispositivo).

**Soluzione:**

La frequenza di 2 KHz corrisponde ad un periodo  $T_{ADC} = 500 \mu s$ .

Ogni  $500 \mu s$  il processore viene interrotto per gestire il trasferimento di un dato dal dispositivo.

Questa gestione ha una durata di  $200 \mu s$

Pertanto, la frazione di tempo in cui il processore è impegnato per il trasferimento dei dati è:

$$200 \mu s / 500 \mu s = 0.4.$$

In percentuale:  $P = 40\%$

2) Un dispositivo di conversione analogico-digitale ADC, gestito ad interrupt, acquisisce dati alla frequenza di 2 KHz. La durata della routine di servizio delle sue interruzioni è  $S_{ADC} = 200 \mu s$ .

Si supponga che, nel sistema descritto, esista anche un altro dispositivo  $D$  che genera richieste di interruzione più prioritarie di quelle di ADC, per trasferire un dato ogni  $T_D = 2 \text{ ms}$ .

Si indichi la massima durata  $S_D$  della routine di servizio del dispositivo  $D$ , onde evitare perdita di dati

**Soluzione:**

Il dispositivo  $D$  è il più prioritario e, quindi, la sua routine di servizio viene attivata, senza alcun ritardo, non appena esso genera una richiesta di interruzione.

Pertanto, non vi può essere alcuna perdita dei dati trasferiti da  $D$ , almeno fin tanto che  $S_D$  non supera il periodo di 2 ms di generazione delle sue richieste di interruzione.

Per quanto riguarda il dispositivo meno prioritario ADC, la routine di servizio delle sue interruzioni subisce un ritardo pari a  $S_D$  ogni volta che viene interrotta da  $D$ .

Confrontando il periodo delle interruzioni di  $D$  ( $T_D = 2 \text{ ms}$ ) con la durata della routine di servizio di ADC ( $S_{ADC} = 200 \mu s$ ), si constata che, durante l'esecuzione di quest'ultima, può verificarsi al più una sola interruzione da parte di  $D$ .

Questa interruzione, se si verifica (caso peggiore), introduce un ritardo pari a  $S_D$  nel completamento della routine di servizio di ADC.

Questo ritardo comporta perdita di dati di ADC se la sua routine di servizio non termina entro i 500  $\mu\text{s}$  del suo periodo. Deve quindi essere:

$$S_{\text{ADC.max}} = S_{\text{ADC}} + S_{\text{D}} = 200 \mu\text{s} + S_{\text{D}} \leq 500 \mu\text{s}. \text{ Cio\`e } S_{\text{D}} \leq 300 \mu\text{s}$$

**3) Un dispositivo d'ingresso IN1, gestito ad *interrupt*, fornisce dati ad intervalli regolari di durata 1 ms.**

**Calcolare la durata della routine di servizio delle interruzioni di IN1 affinché la percentuale di tempo di processore (trascurando la predisposizione iniziale del dispositivo) impegnata per il trasferimento dei dati sia pari al 20%.**

**Soluzione:**

Ogni 1 ms (1000  $\mu\text{s}$ ), il processore è impegnato, per un tempo pari a  $T_r$   $\mu\text{s}$ , ad eseguire la routine di servizio delle interruzioni .

La percentuale di tempo di processore impegnata per questa attività è:

$$100 \cdot (T_r \mu\text{s}) / (1000 \mu\text{s})$$

Affinché questa percentuale sia pari al 20%, deve essere:

$$100 \cdot T_r / 1000 = 20. \text{ Da cui } T_r = 200 \mu\text{s}$$

**4) Un sistema con ciclo di bus della durata di 30 ns ha un dispositivo, gestito in DMA, che trasferisce un dato ogni 1  $\mu\text{s}$ .**

**Quale è la percentuale minima e massima di tempo di CPU che viene perso per il trasferimento dei dati?**

**Soluzione:**

L'occupazione minima di tempo di CPU si ha assumendo l'assenza totale di conflitto tra CPU e DMA per tutte le richieste di DMA:  $\text{Min} = 0 \%$

L'occupazione massima si ha supponendo un conflitto ad ogni richiesta del DMA ed è pari a 30 ns ogni 1  $\mu\text{s}$ .

$$\text{In percentuale \u00e9 } 100 \cdot 30 \cdot 10^{-9} / 10^{-6} = 3\% \quad \text{Max} = 3 \%$$

$$100 \times ( \text{misura piccola} / \text{misura grande} )$$

**5) Si consideri un sistema in cui l'accesso ai word in memoria avviene in un ciclo di 200 ns. Un convertitore A/D, controllato in DMA, trasferisca blocchi da  $2^{11}$  word e non vi siano altri dispositivi di I/O.**

**Si desidera che il rallentamento subito dal processore durante il trasferimento di uno di questi blocchi di dati non superi il 2%.**

**Si calcoli quale deve essere la frequenza massima  $f_C$  del *clock* che sincronizza l'acquisizione e il trasferimento dei dati del convertitore A/D.**

### **Soluzione**

Il massimo ritardo (caso peggiore) si ha quando il conflitto descritto si verifica ad ogni dato trasferito in DMA.

Per contenere questo ritardo entro il limite del 2% è necessario che il periodo  $p_C$  del clock che sincronizza il trasferimento dei singoli dati sia tale da non sottrarre al processore più di due cicli del bus di memoria ogni 100 cicli, ovvero non più di un ciclo ogni 50.

50 cicli corrispondono ad un periodo  $p_C = 50 \cdot 200 \text{ ns} = 10 \text{ } \mu\text{s}$

**6) Un dispositivo, gestito in DMA, trasferisce un dato ogni 100  $\mu\text{s}$ ; il rallentamento massimo subito dalla CPU a causa di questo trasferimento è pari allo 0,3%.**

**Si calcoli la durata  $T_b$  (in ns) del ciclo di bus.**

### **Soluzione:**

Per un rallentamento del 0.3% la CPU subisce tre cycle stealing ogni 1000 cicli. Questo significa che in 1000 cicli il dispositivo ha trasferito 3 dati (e ho avuto 1 collisione per dato).

Per trasferire 3 dati ci metto  $100 \mu\text{s} \times 3 = 300 \text{ } \mu\text{s}$ , e in questo tempo faccio 1000 cicli; quindi, la CPU ha un periodo di  $300 \text{ } \mu\text{s} / 1000 = 0.3 \text{ } \mu\text{s} = 300 \text{ ns}$

**7) Due dispositivi di conversione analogico-digitale, ADC1 e ADC2, acquisiscono campioni dello stesso segnale, con frequenze di campionamento diverse, ADC1 via DMA, ADC2 gestito tramite le interruzioni.**

**A causa delle caratteristiche del segnale (di durata complessiva pari a 1 s), si debbano acquisire i campioni tra l'istante 0 e l'istante 0.5 s (escluso) con ADC1 alla frequenza di 100 KHz, e quelli dall'istante 0.5s all'istante 1s (escluso) con ADC2 alla frequenza di 200 Hz. Il primo campione viene acquisito da ADC1 all'istante 0 e quello di ADC2 all'istante 0.5s.**

**Il numero di campioni acquisiti dai due convertitori A/D è, quindi:  $N_1 = 50000$  acquisiti da ADC1 e  $N_2 = 100$  acquisiti da ADC2**

**Supponendo che il ciclo di bus duri 200 ns, la RSI di ADC2 20  $\mu\text{s}$ , quella del DMA controller di ADC1 50  $\mu\text{s}$  e trascurando le predisposizioni iniziali dei dispositivi, si indichi il tempo di CPU complessivamente impegnato per l'acquisizione dei 50100 campioni.**

### **Soluzione:**

Trascurando le predisposizioni iniziali dei dispositivi, il tempo  $T_{1_{DMA}}$  sottratto alla CPU per l'acquisizione dei primi 50000 campioni via DMA, nel caso peggiore in cui il processore subisce un ritardo di un ciclo del bus per ciascuno dei dati trasferiti, è:

$$T_{1_{DMA}} = 50000 * 200 \text{ ns} = 10000 \text{ } \mu\text{s}$$

A questo tempo va aggiunta la durata dell'esecuzione della RSI del DMA controller, attivata dopo il trasferimento dell'ultimo dato:

$$T_{1_{DMA.RSI}} = 50 \text{ } \mu\text{s}.$$

Per l'acquisizione dei successivi 100 campioni, gestita tramite le interruzioni, va conteggiato il tempo impiegato dal processore per l'esecuzione della relativa RSI, che viene attivata per ciascuno dei dati trasferiti:

$$T_{2_{RSI}} = 100 * 20 \text{ } \mu\text{s} = 2000 \text{ } \mu\text{s}$$

Il tempo  $C$  di CPU complessivamente impegnato per l'acquisizione dei 50100 campioni è la somma di questi tempi:

$$C = T_{1_{DMA}} + T_{1_{DMA.RSI}} + T_{2_{RSI}} = (10000 + 50 + 2000) \text{ } \mu\text{s} = 12050 \text{ } \mu\text{s}$$

**8) Un disco rigido è organizzato in settori da 512 byte, 32 settori per traccia, 64 tracce per faccia.**

**Un giro completo del disco viene compiuto in 512 ms.**

**Il trasferimento del contenuto di un settore da disco a memoria o viceversa avviene in DMA, un word alla volta. Durante il trasferimento dei 128 word di un settore, il processore può subire un rallentamento dovuto ai conflitti con il DMA controller nell'accesso al bus.**

**Si vuole contenere questo rallentamento entro un valore massimo dello 0.4%.**

**Si calcoli quale dovrebbe essere la durata  $T_b$  massima dei cicli del *bus* che assicura un rallentamento del processore non superiore a questo valore.**

**Soluzione:**

In 512 ms viene effettuato un giro completo del disco, durante il quale vengono letti/scritti/trasferiti i dati di 32 settori, ciascuno contenente  $512/4 = 128$  word (dati).

Vengono quindi trasferiti  $128 * 32 = 2^{12} = 4096$  dati in 512 ms, ovvero un dato ogni  $P_d = 512/4096 \text{ ms} = 125 \text{ } \mu\text{s}$

Se  $T_b$  è la durata di un ciclo del bus, il rallentamento massimo subito dal processore, dovuto ai conflitti con il DMA controller per l'accesso al bus (*cycle stealing*), in percentuale è:

$$100 * T_b / P_d$$

Si vuole contenere questo rallentamento entro un valore dello 0.4%; pertanto deve essere

$$100 * T_b / P_d \leq 0.4. \text{ Cioè:}$$

$$100 * T_b / (125 * 10^{-6}) \leq 0.4, \text{ ovvero } T_b \leq 0.4 * 125 * 10^{-6} / 100 = 500 * 10^{-9}$$

$$T_b \leq 500 \text{ ns}$$

**9) Un disco con 8 superfici, 110 tracce/superficie, 64 settori/traccia, 512 byte/settore, ruota alla velocità di 480 rpm (rivoluzioni / minuto).**

**Un processore legge N settori consecutivi (trasferiti da disco alla massima velocità) gestendo il trasferimento con una interruzione per ogni byte trasferito.**

**Supponendo che il servizio di ciascuna interruzione duri 2.5  $\mu$ s, si calcoli la percentuale PI di tempo di CPU dedicato alla lettura dei byte di un settore.**

**Se invece il trasferimento dei dati venisse effettuato in DMA, con una interruzione dopo il trasferimento dell'ultimo dato di ciascun settore, ipotizzando che il servizio di tale interruzione duri 2.5  $\mu$ s e che il ciclo di bus abbia una durata di 500 ns, si calcoli la percentuale PD<sub>MAX</sub> di tempo massimo di CPU dedicato alla lettura degli N settori.**

**Soluzione:**

Alla velocità di 480 rpm, un giro completo del disco viene effettuato in 60/480 secondi, ovvero in 125 ms. In questi 125 ms vengono letti/scritti/trasferiti i dati di 64 settori, ciascuno contenente 512 byte (dati).

Vengono quindi trasferiti  $512 \cdot 64 = 32768$  dati in 125 ms, ovvero un dato ogni  $P_d = 125 / 32768$  ms = 3.815  $\mu$ s

Se il trasferimento dei dati viene effettuato tramite le interruzioni, per ciascun dato viene eseguita la RSI che ha una durata di 2.5  $\mu$ s

La percentuale di tempo di CPU dedicato alla lettura dei byte di un settore è quindi  $100 \cdot 2.5 \mu\text{s} / 3.815 \mu\text{s} = 65.53\%$

Se invece il trasferimento dei dati viene effettuato in DMA, la percentuale di occupazione del processore nel caso peggiore è quella che si ottiene quando ogni richiesta di DMA confligge con un accesso di CPU, procurando un ritardo pari alla durata di un ciclo del bus (500 ns).

In questo caso va considerato anche l'impegno del processore per l'esecuzione della RSI (2.5  $\mu$ s) dopo il trasferimento dell'ultimo dato di ciascun settore.

Il tempo di processore impegnato per trasferire i 512 dati di ciascun settore è:

$$TD_{MAX} = 512 \cdot 500 \text{ ns} + 2.5 \mu\text{s} = 258.5 \mu\text{s}$$

Nei 125 ms in cui il disco effettua un giro completo vengono letti 64 settori; pertanto, la durata del trasferimento dei dati di uno dei 64 settori è  $125 / 64 = 1.953$  ms.

la percentuale PD<sub>MAX</sub> di tempo massimo di CPU dedicato alla lettura degli ciascuno degli N settori è:  $PD_{MAX} = 100 \cdot 258.5 \mu\text{s} / 1.953 \text{ ms} = 13.23 \%$

**10) In un sistema dotato di due dispositivi (D1 e D2) che trasferiscono dati con modalità DMA, ciascuno alla velocità di 10 Kbyte/s, posto che un ciclo di memoria abbia la durata di 100 ns, si indichi il tempo massimo T che può intercorrere tra l'istante di invio del segnale *Bus Request*, con cui il DMA controller chiede di utilizzare un ciclo del bus di memoria per operare il trasferimento del dato richiesto (*DMA REQ*) da uno dei due dispositivi e l'istante**

**di inizio del ciclo di memoria assegnato al dispositivo. (Si suppone che non vi siano altri dispositivi, oltre a D1 e D2, che operino in DMA).**

**Soluzione:**

Il tempo massimo  $T$  richiesto si verifica nel caso peggiore, cioè quando entrambi i dispositivi generano la richiesta DMA REQ immediatamente dopo l'inizio di un ciclo del bus. In questa circostanza, il processore concede l'uso del primo ciclo del bus disponibile, che inizia dopo 100 ns, a uno dei due dispositivi (D2 nella figura) (si ricorda che il processore considera le proprie richieste di uso di cicli del bus meno prioritarie di quelle dei DMA controller).

L'altro dispositivo (D1) si trova nel caso peggiore e potrà utilizzare il ciclo successivo, con un ritardo  $T = 200$  ns.

## **Esercizi 04**

### **Pipeline**

### **Chapter 11**

**1) Un processore, con la pipeline disattivata, per svolgere 100 operazioni impiega un tempo  $T_1$  di 1us. Lo stesso processore, con la pipeline attiva, per svolgere le stesse 100 operazioni impiega un tempo  $T_2$  di 109 ns.**

**Quale è il fattore di speedup?**

**Quanti stadi avrà la pipeline?**

Soluzione:

Il fattore di speedup è  $T_1/T_2 = 1000 / 109 =$  circa 10

Il tempo senza pipeline è  $100 * 10$  stadi  $* 1$ ns a stadio = 1000 ns

Il tempo con pipeline è  $10 * 1$ ns +  $1$ ns  $* (100-1) = 109$  ns

**2) Si calcoli il valore minimo di numero  $N$  di istruzioni di un programma affinché in un processore con una pipeline a 5 fasi, il fattore di speedup risulti essere almeno 3.**

Soluzione:

Senza pipeline l'esecuzione sarebbe lunga  $N * 5$  stadi  $* T$

Con la pipeline l'esecuzione sarebbe  $5 * T + (N-1) * T = (5 + N - 1) * T$

Lo speedup è  $N * 5 / (N + 5 - 1) = N * 5 / (N + 4)$  e deve essere 3

Quindi  $5n = 3n + 12$ , quindi  $5n - 3n = 12$ , quindi  $2n = 12$ , quindi  $n = 6$

3) Supponendo di disporre di due processori compatibili (eseguono con lo stesso significato il medesimo codice macchina), tali che la durata dell'esecuzione di un'intera istruzione sia la stessa ma dotati di una pipeline rispettivamente di  $s_1=5$  e  $s_2=8$  stadi, si calcoli quanto vale il rapporto  $R = T_1/T_2$  tra le durate dell'esecuzione della medesima successione di 100 istruzioni, partendo da *pipeline* vuoti e in condizioni ideali (senza *hazard*).

**Soluzione:**

$$T_1 = (5 + 100 - 1) * t = \text{con } t \text{ che vale } z/5$$

$$T_2 = (8 + 100 - 1) * t = \text{con } t \text{ che vale } z/8$$

$$T_1/T_2 = 104 z/5 / 107 z/8 = z104/5 * 8/z107 = 104*8 / 5*107 = 1.5555$$

4) Un processore è dotato di una pipeline a  $k=6$  stadi da 15 ns ciascuno. Si è valutato che mediamente una istruzione su 20 è soggetta ad un inceppamento della pipeline (dovuto a *data hazard* o ad altra causa) che allunga l'esecuzione di quella istruzione di un tempo equivalente a 4 stadi. In queste ipotesi calcolare il fattore di *speedup*  $S_k (= T_1/T_k)$ , trascurando il transitorio iniziale.

**Soluzione:**

$$\text{Per eseguire 20 istruzioni } T_1 \text{ è } 20 * 6 * 15\text{ns} = 1800\text{ns}$$

$$\text{Con la pipeline ho, a regime, 1 istruzione ogni } 15\text{ns, quindi } 20 * 15\text{ns} = 300\text{ns}$$

Però su 20 operazioni ho un inceppamento, con un ritardo di  $4 * 15\text{ns} = 60\text{ns}$ , per un totale di 360ns

$$\text{Lo speedup } T_1 / T_k \text{ è } 1800 / 360 = 5$$

5) Un processore è dotato di una pipeline a  $k=4$  stadi e di una memoria *cache*. Mediamente il numero di accessi in memoria per ogni istruzione è pari a 1,3 e l'*hit rate* della *cache* è pari al 91%. Si è rilevato che ogni *miss* nella *cache* provoca nella pipeline un inceppamento (ritardo) di 2 stadi. In queste ipotesi, calcolare il fattore di *speedup*  $S_{K1} = T_1/T_k$ , trascurando il transitorio iniziale.

Si supponga inoltre che, a causa dei vari *hazard* la pipeline subisca mediamente un ulteriore inceppamento di 1 stadio ogni 20 istruzioni. In queste ipotesi, calcolare il fattore di *speedup*  $S_{K2} = T_1/T_k$ , sempre trascurando il transitorio iniziale.

**Soluzione:**

Il tempo  $T_1$  richiesto per eseguire  $N$  istruzioni senza pipeline deve tener conto del fatto che il 9% degli  $1.3*N$  accessi alla memoria comporta un ritardo pari a  $2*Ts$ ; allora è

$$T_1 = N*k*Ts + (0.09 * 1.3)*N*2*Ts = N*Ts*(4 + 0.09 * 1.3 * 2) = 4.234 * N * Ts$$

Con una pipeline di k stadi, a regime e senza inceppamenti, viene completata un'istruzione ogni  $T_s$ ; il tempo richiesto per eseguire N istruzioni con pipeline è  $N * T_s$  più i ritardi dovuti ai miss nella cache (che si verificano nel 9% degli  $1.3 * N$  accessi alla memoria che hanno luogo nell'esecuzione delle N istruzioni); allora è

$$T_k = N * T_s + (0.09 * 1.3) * N * 2 * T_s = N * T_s * (1 + 0.09 * 1.3 * 2) = 1.234 * N * T_s$$

$$S_{k1} = T_1 / T_k = (4.234 * N * T_s) / (1.234 * N * T_s) = 3.43118$$

Diversamente dai miss nella cache, che introducono un ritardo sia in presenza, sia in assenza di una pipeline, gli ulteriori ritardi dovuti ad altri hazard (dipendenza tra i dati, salti condizionati) si verificano solo nel caso in cui vi sia una pipeline.

Pertanto, il valore di  $T_1$  rimane quello calcolato prima, tenendo conto solo dei ritardi dovuti ai cache miss:  $T_1 = 4.234 * N * T_s$ ;

mentre il valore di  $T_k$  aumenta per i ritardi dovuti agli altri hazard:

$$T_k = 1.234 * N * T_s + (N/20) * T_s = (1.234 + 0.05) * N * T_s$$

$$S_{k2} = T_1 / T_k = (4.234 * N * T_s) / (1.284 * N * T_s) = 3.297508$$