



Scuola Politecnica e delle Scienze di Base  
Corso di Laurea Magistrale in **Ingegneria Informatica**

## *Elaborato finale per il corso di Calcolatori Elettronici II*

Prof.ssa Valentina Casola

A.A. 2019/2020

Studenti:

Della Torca Salvatore

Matr. M63001011

# Indice

<b>1</b>	<b>Specifiche di progetto</b>	<b>1</b>
<b>2</b>	<b>Soluzione</b>	<b>2</b>
2.1	Architettura del sistema . . . . .	2
2.2	Protocolli . . . . .	3
2.3	Mappa della memoria . . . . .	5
2.4	Implementazione . . . . .	7
2.4.1	Documentazione del codice . . . . .	7
2.4.2	Codice Assembly . . . . .	10
2.4.3	Configurazione in Asim . . . . .	10

# Capitolo 1

## Specifiche di progetto

Un sistema S1 è dotato di una periferica parallela P1 operante secondo il meccanismo delle interruzioni. Il sistema è programmato per ricevere  $N$  vettori di 5 interi ciascuno, espressi su 1 byte, secondo la seguente modalità: ogni 5 interi  $xi_1...xi_5$  ricevuti sulla periferica P1, il sistema effettua una verifica sul vettore ricevuto tramite una funzione di libreria VERIFICA, che prende in ingresso l'indirizzo del vettore, la sua dimensione e il valore  $VAL=100$  e restituisce un valore pari a 1 se la somma degli elementi del vettore è maggiore di VAL.

Nel caso in cui il valore restituito sia pari a 1, il vettore di 5 interi  $[xi_1, xi_2, xi_3, xi_4, xi_5]$  viene trasferito in un'apposita area di memoria utilizzando un DMA. In caso contrario, il vettore non viene copiato.

*(Nota: i vettori di 5 interi acquisiti nel tempo devono essere memorizzati in aree contigue).*

Una volta ricevuti gli  $N$  vettori in maniera corretta, le interruzioni sulle periferiche vengono disabilitate dal programma principale, che quindi deve essere in grado di rilevare la fine della trasmissione.

In particolare, è stata fatta la scelta di porre  $N=1$ , ovvero di trasmettere un solo vettore di 5 numeri.

# Capitolo 2

## Soluzione

### 2.1 Architettura del sistema

Da un punto di vista architetturale i due sistemi  $S1$  e  $S2$  sono costituiti da una memoria, un processore e una periferica parallela PIA. In più il sistema  $S1$  che riceve presenta anche un DMA che ha il compito di trasferire il vettore di 5 interi in un'apposita area di memoria nel caso in cui vada a buon fine la funzione di verifica.

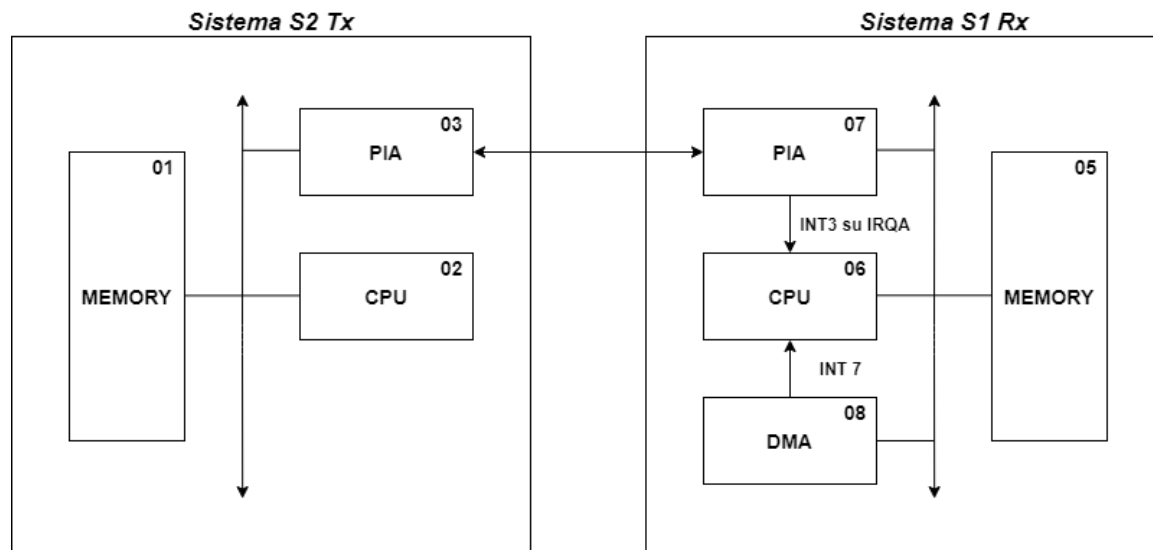


Fig. 2.1: Architettura del sistema

Il programma è stato implementato mediante l'utilizzo di 2 interruzioni, entrambe localizzate nel sistema  $S1$ :

- Interruzione in ricezione della PIA del sistema  $S1$ , sulla linea **INT3**, che è responsabile di ricevere e memorizzare i valori che sono stati inviati dalla PIA del sistema  $S2$ ;

- Interruzione del DMA, sulla linea INT7, che è stata implementata per permettere al programma principale di disabilitare le interruzioni.

## 2.2 Protocolli

Le seguenti immagini mostrano come sono legate le PIA tra di loro e il protocollo di comunicazione.

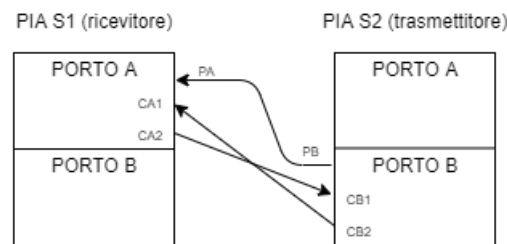


Fig. 2.2: Collegamento tra le PIA

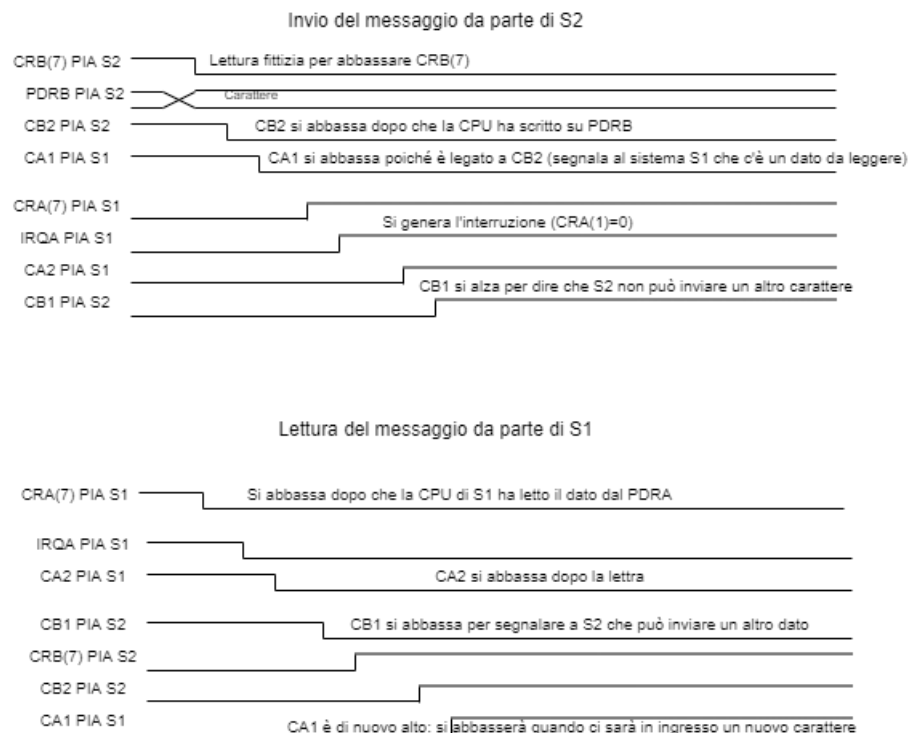


Fig. 2.3: Protocollo di comunicazione tra le PIA

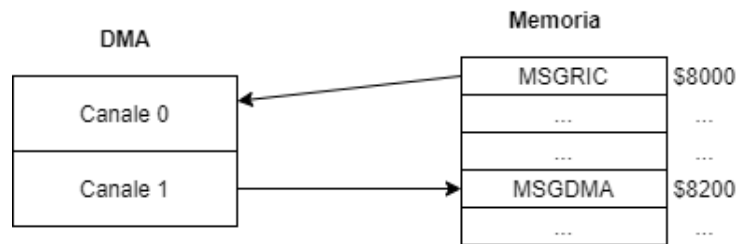


Fig. 2.4: Comunicazione DMA-memoria

Il DMA si occupa invece di prendere il messaggio depositato dalla PIA all'indirizzo 8000 e di trasferirlo all'indirizzo 8200. Nel registro `caddr0` del canale 0 carico quindi l'indirizzo 8000, mentre in `caddr1` del canale 1 carico l'indirizzo 8200. Una volta configurati i registri di modo per entrambi i canali per settare la comunicazione memoria interfaccia, si invia la richiesta di bus al processore.

Una volta che la richiesta viene accettata, il DMA inizia il trasferimento in modalità block: prende un byte per volta dall'indirizzo 8000 e lo memorizza nel canale 0, lo passa poi al registro dato del canale 1, e questo poi trasferisce il byte all'indirizzo 82000. Per ogni byte trasferito si decrementa il contatore di caratteri da trasferire: quando questo contatore diventa 0, allora si attiva l'interruzione legata a EOP (INT7) per segnalare alla CPU che il trasferimento è finito e quindi ritorna a lui il controllo del bus indirizzi.

## 2.3 Mappa della memoria

	MEMORIA S1 RX			MEMORIA S2 TX	
\$8000	00	00	\$8000	30	28
\$8002	00	00	\$8002	08	00
\$8004	00	05	\$8004	15	90
\$8006	00	100	\$8006	05	
\$8008	00	00	\$8008		
\$8010	00		\$8010		
\$8200	00	00	\$8200	MAIN	
\$8202	00	00	\$8202		
\$8204	00		\$8204		
\$8500	MAIN		\$8500		
\$9000	INT3		\$9000		
\$9200	INT7		\$9200		
\$9500	VERIFICA		\$9500		

Fig. 2.5: Mappa della memoria dei sistemi S1 e S2

	REGISTRI S1			REGISTRI S2	
\$2004	PDRA	CRA	\$2004	PDRA	CRA
\$2006	PDRB	CRB	\$2006	PDRB	CRB
\$2008			\$2008		
\$2010	CADDR0		\$2010		
\$2012	CADDR1	CCOUNT1	\$2012		
\$2014		nbyte	\$2014		
\$2016			\$2016		
\$2018	CNTRL		\$2018		
\$2020		MODE	\$2020		
\$2022		RESET	\$2022		
\$2024			\$2024		

Fig. 2.6: Mappatura dei registri delle periferiche

Dato che il processore M68000 utilizza la tecnica degli *autovettori*, le ISR delle periferiche si trovano in una memoria ROM di 256 locazioni in una posizione che dipende dalla linea di interruzione alla quale è collegata la periferica associata.

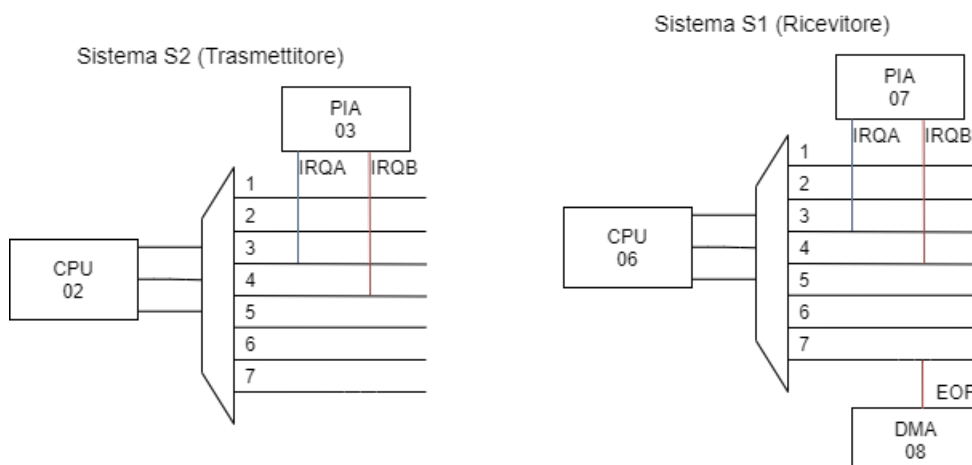


Fig. 2.7: Collegamenti alle linee di interruzione

Poiché la ISR della PIA del sistema S1 è collegata alla linea INT3, e poiché nella ROM i 7 autovettori si trovano a partire dall'indice 24 fino all'indice 31, la ISR della PIA si troverà all'indice  $24 + 3 = 27$ , il cui indirizzo è pari a  $27 \cdot 4 = 108 \rightarrow 6C$ .

Lo stesso discorso vale anche per la ISR del DMA che è mappata sulla linea INT7. Per il sistema S2 non è stato rappresentato il vettore delle eccezioni in quanto tale sistema (il trasmettitore) non scatena alcuna interruzione.

\$	INDICE	ROM
00	0	
01	1	
6C	27	\$9000
70	28	
74	29	
7C	31	\$9200
FF	255	

Fig. 2.8: Memoria dei vettori ISR per sistema S1



## 2.4 Implementazione

### 2.4.1 Documentazione del codice

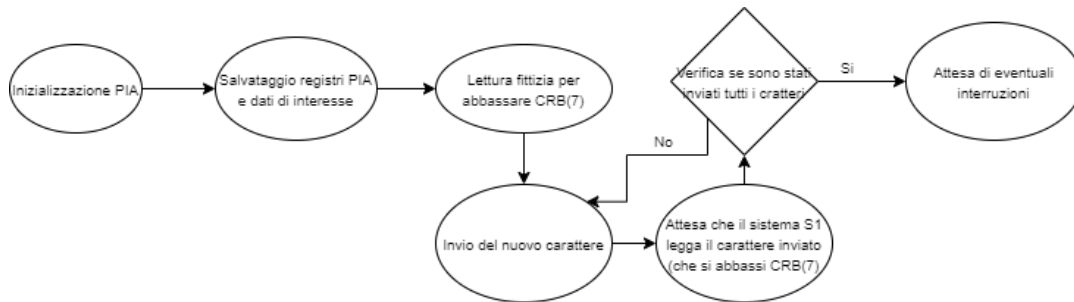


Fig. 2.9: Diagramma di flusso Main S2

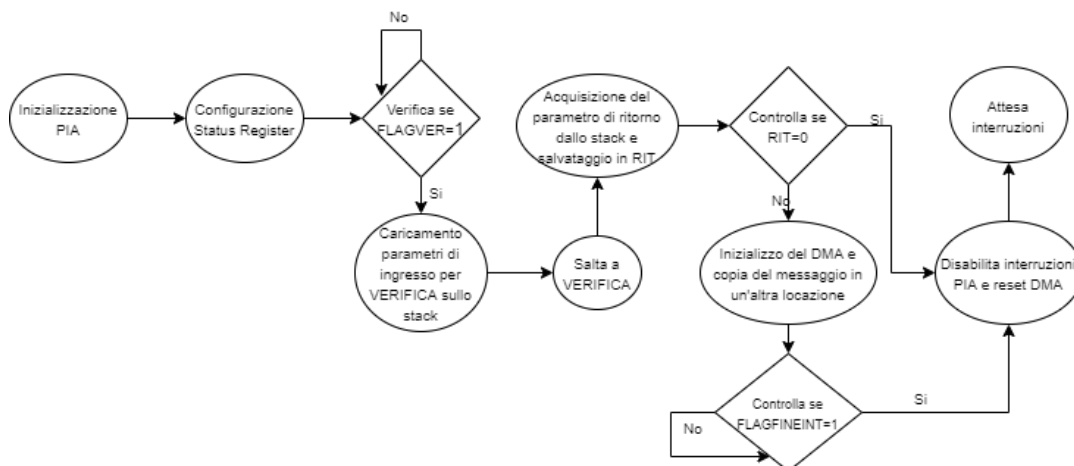


Fig. 2.10: Diagramma di flusso Main S1

FLAGVER è il flag che verrà messo ad 1 dall'interruzione INT3 della PIA, per segnalare al programma principale che il vettore di numeri è stato completamente ricevuto e quindi si può chiamare la funzione VERIFICA. Se tale funzione restituisce 0, allora il DMA non lavora proprio e si passa alla disabilitazione delle interruzioni.

Se invece il DMA effettua il trasferimento, allora prima di passare alla disabilitazione delle interruzioni si controlla il flag FLAGFINEINT, che verrà messo ad 1 dalla ISR del DMA per segnalare che il trasferimento memoria-memoria è stato completato. Tale flag lo setta questa ISR poiché è stato scelto di trasmettere un solo vettore.

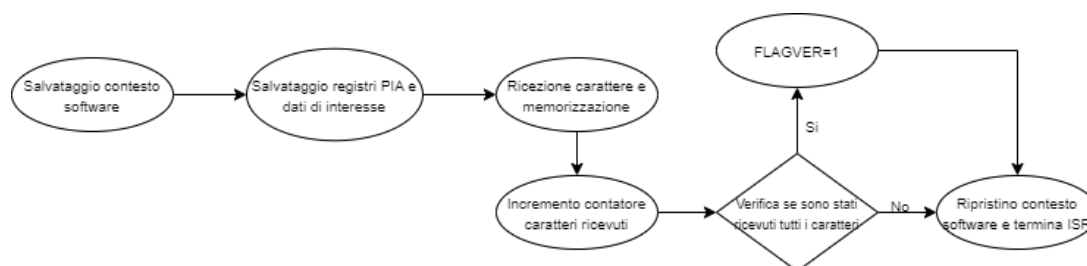


Fig. 2.11: Diagramma di flusso INT3 (ISR della PIA) di S1



Fig. 2.12: Diagramma di flusso INT7 (ISR del DMA) di S1

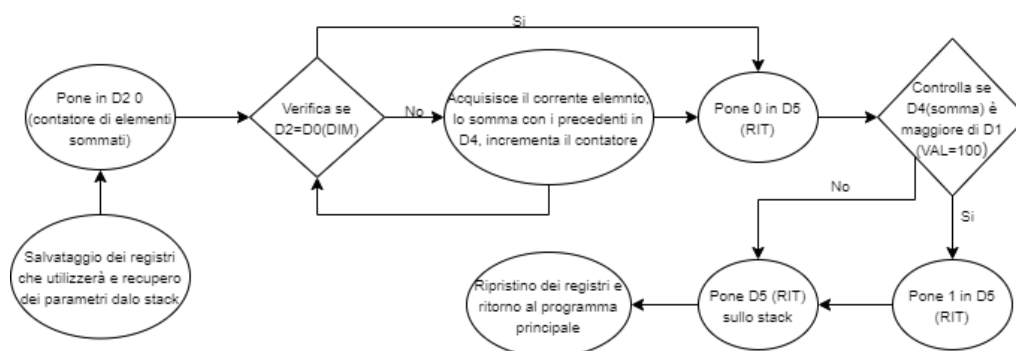


Fig. 2.13: Diagramma di flusso funzione VERIFICA

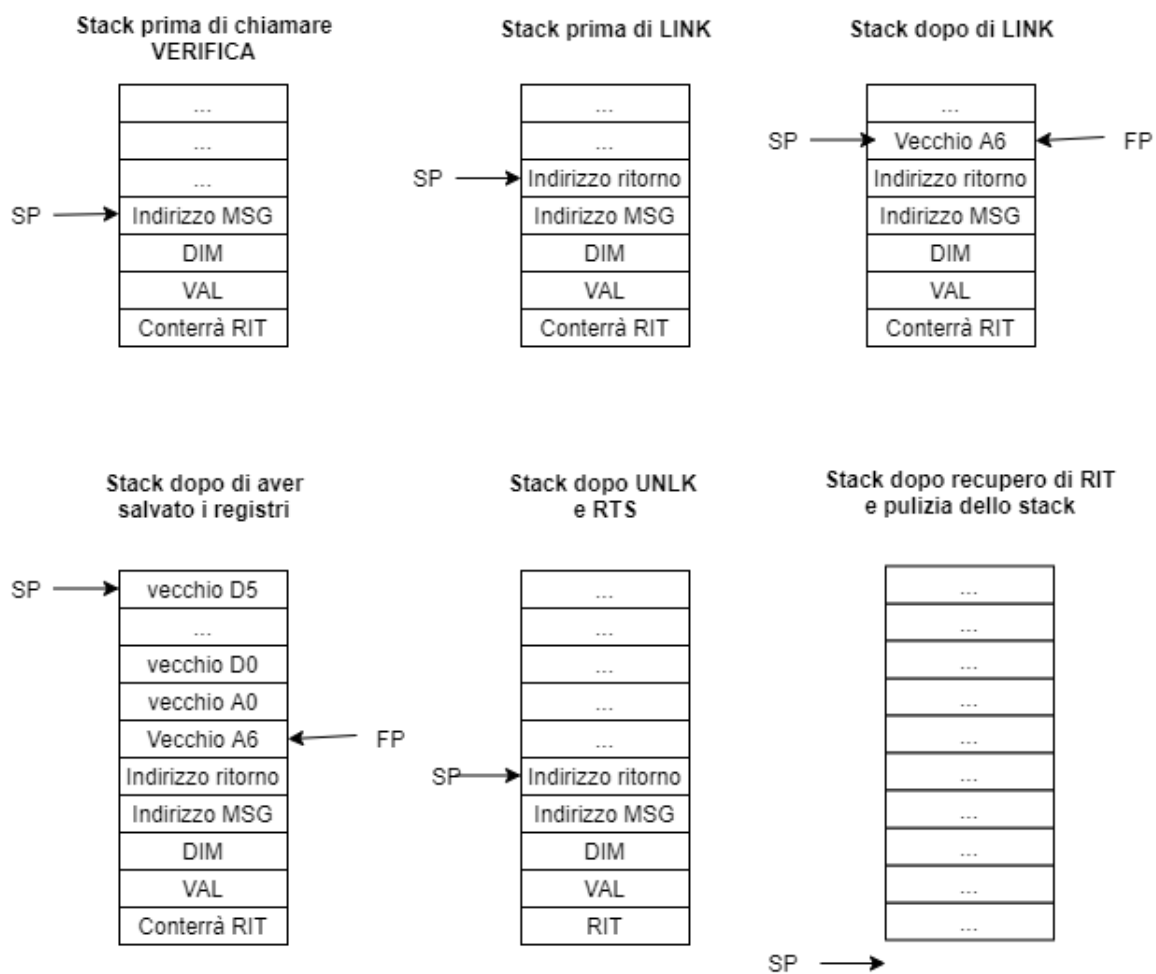


Fig. 2.14: Evoluzione dello stack

## 2.4.2 Codice Assembly

Per il codice Assembly si rimanda ai file .a68, presenti nello zip, dotati di opportuni commenti.

## 2.4.3 Configurazione in Asim

```

Configuration name: config.cfg

CHIP Name: MEMORY
Type: MMU/BUS.  Identif: 01.      BUS: 0000.
Address 1: 00008000.      Address 2: 00000000.
Com1: 0000. Com2: 0010. Com3: 0008. Com4: 0000.

CHIP Name: M68000
Type: CPU.      Identif: 02.      BUS: 0001.
Address 1: 00009000.      Address 2: 00009200.
Com1: 0000. Com2: 0000. Com3: 0000. Com4: 0000.

CHIP Name: M6821PIA
Type: Device.   Identif: 03.      BUS: 0001.
Address 1: 00002004.      Address 2: 00002007.
Com1: 0002. Com2: 0003. Com3: 0004. Com4: 0207.

CHIP Name: MEMORY
Type: MMU/BUS.  Identif: 05.      BUS: 0000.
Address 1: 00008000.      Address 2: 00000000.
Com1: 0000. Com2: 0010. Com3: 0008. Com4: 0000.

CHIP Name: M68000
Type: CPU.      Identif: 06.      BUS: 0005.
Address 1: 0000A000.      Address 2: 0000A200.
Com1: 0000. Com2: 0000. Com3: 0000. Com4: 0000.

CHIP Name: M6821PIA
Type: Device.   Identif: 07.      BUS: 0005.
Address 1: 00002004.      Address 2: 00002007.
Com1: 0006. Com2: 0003. Com3: 0004. Com4: 0203.

CHIP Name: I8237DMA
Type: Device.   Identif: 08.      BUS: 0005.
Address 1: 00002010.      Address 2: 0000201F.
Com1: 0006. Com2: 0007. Com3: 0006. Com4: 0000.
  
```

Fig. 2.15: File di configurazione ASIM