



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

Elaborato finale per il Corso di Calcolatori Elettronici 2

Prof. Valentina Casola

A. A. 2019/2020

Studente:

Salvatore Della Torca

Matr. M63001011

Indice

1	Specifiche di progetto	1
2	Soluzione	2
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	9
2.4.3	Configurazione in Asim	9

Capitolo 1

Specifiche di progetto

Un sistema S è dotato di una periferica seriale e di una periferica parallela. Il sistema trasmette un messaggio di N caratteri sulla periferica seriale e ottiene l'eco del messaggio sulla periferica parallela.

Lo studente può operare in una tra le seguenti tre ipotesi (di differente complessità):

1. Il messaggio è trasmesso interamente dalla seriale e successivamente è ricevuto mediante il meccanismo delle interruzioni sulla periferica parallela. Non può essere inviato un altro messaggio sulla seriale se non è stato ricevuto prima l'eco sulla parallela.
2. Il messaggio è trasmesso sulla seriale ma nel frattempo sulla parallela può iniziare l'eco. In questo caso non è possibile inviare un messaggio sulla seriale se non si è ricevuto prima totalmente il messaggio sulla parallela.
3. Non viene fatta alcuna ipotesi di correlazione tra le attività della periferica seriale e la parallela.

Si illustrino:

- l'architettura complessiva del sistema;
- il collegamento tra i dispositivi;
- i protocolli;
- il software e la memoria con riferimento a dati e programmi in essa allocati.

Ho scelto di lavorare nell'ipotesi 2.

Capitolo 2

Soluzione

2.1 Architettura del sistema

Il sistema è fatto da 2 dispositivi che comunicano tra di loro, S1 e S2, che sono identici: entrambi hanno la propria CPU, la propria memoria, una periferica parallela (PIA M6821) e una seriale (USART Intel8251).

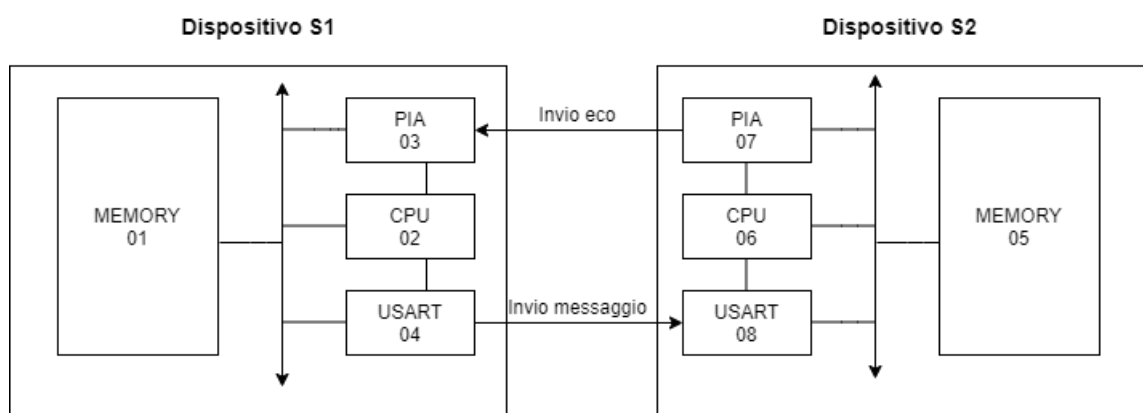


Fig. 2.1: Architettura complessiva

- In S1 ci sarà un'interruzione generata dalla USART verso la CPU per la trasmissione: essa però non verrà usata in quanto ho deciso di trasmettere nel main. Inoltre ci sarà un'interruzione scatenata dalla PIA sul porto A per ricevere l'eco scatenata dal S2.
- In S2 ci sarà invece solo un'interruzione, quella scatenata dall'USART, per ricevere i singoli caratteri inviati dal sistema S1: tale ISR si occupa poi di inviare con la PIA (porto B) l'eco del carattere appena ricevuto.

2.2 Protocolli

La seguente immagine mostra il protocollo della USART: in entrambi i sistemi *DSR* viene alzato da configurazione; l'USART Z1 poi sul fronte di discesa di *TxRDY* shifta il dato da inviare di una posizione verso sinistra, poiché è pronto ad inviare. Il sistema S2 invece si ritrova il dato in *DATAIN* eseguendo lo shift verso destra di una posizione: una volta letto il dato viene abbassato *RxRDY* che, essendo legato al *TxRDY* dell'altro sistema, segnala a S1 che può inviare un nuovo carattere.

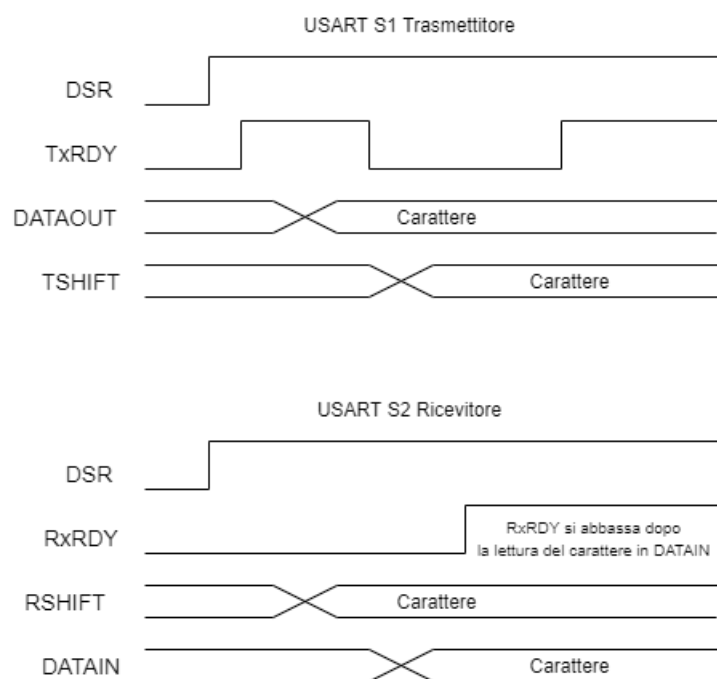


Fig. 2.2: Protocollo di comunicazione tra le USART

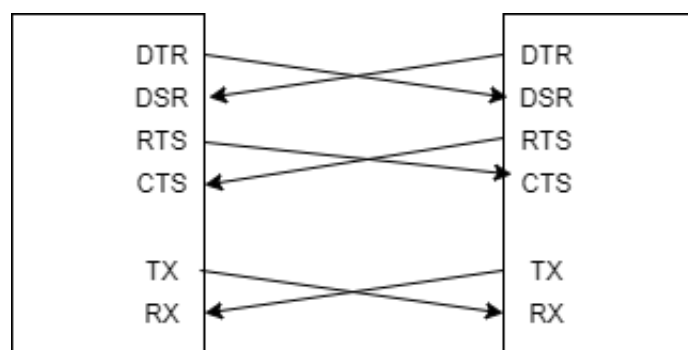


Fig. 2.3: Come sono legate le USART

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

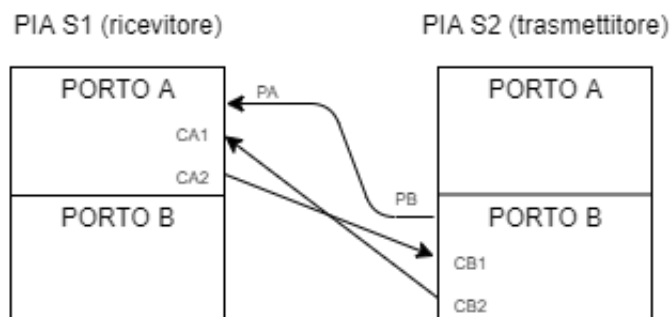


Fig. 2.4: Come sono legate le PIA

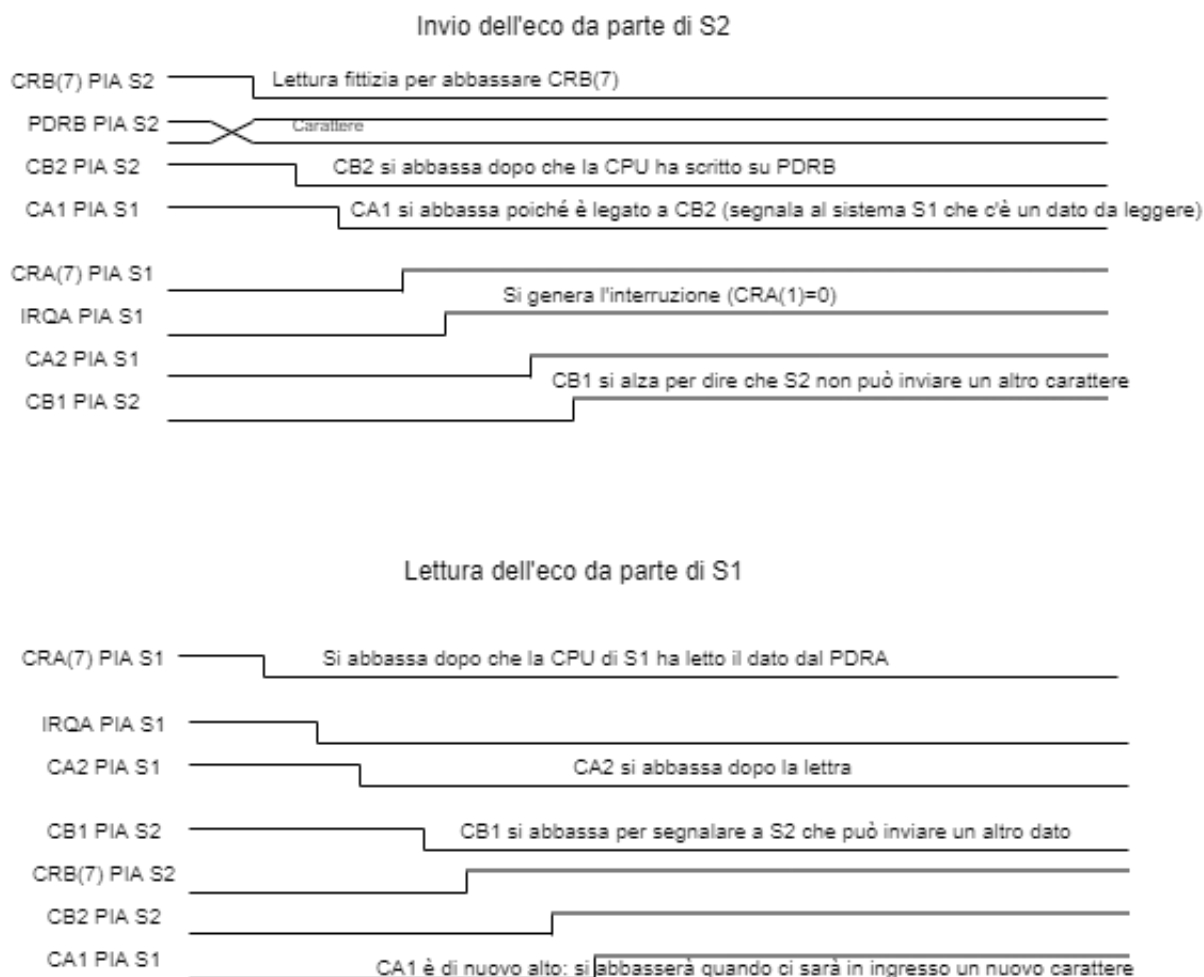


Fig. 2.5: Protocollo di comunicazione tra le PIA

2.3 Mappa della memoria

Mappa memoria sistema S1					Mappa memoria sistema S2	
\$8000	1B	03	0E	07	\$8000	
\$8004	09	0B	1E	15	\$8004	
\$8006	08	01	00		\$8006	08 00
*					*	
*					*	
\$8200	MAIN				\$8200	MAIN
*					*	
*					*	
\$8700	RICEZIONE				\$8700	INTERRUPT
*					*	
\$8800	TRUSART				*	
*					*	

Fig. 2.6: Mappe delle memorie di S1 e S2

Mappatura dispositivi del sistema S1			Mappatura dispositivi del sistema S2		
\$2002	USARTD	} USART 04	\$2002	USARTD	} USART 08
\$2003	USARTCTR		\$2003	USARTCTR	
\$2004	PDRA	} PIA 03	\$2004	PDRA	} PIA 04
\$2005	CRA		\$2005	CRA	
\$2006	PDRB		\$2006	PDRB	
\$2007	CRB		\$2007	CRB	

Fig. 2.7: Mappatura delle periferiche in memoria

Vediamo adesso come le periferiche sono legate alle linee di interruzione della CPU.

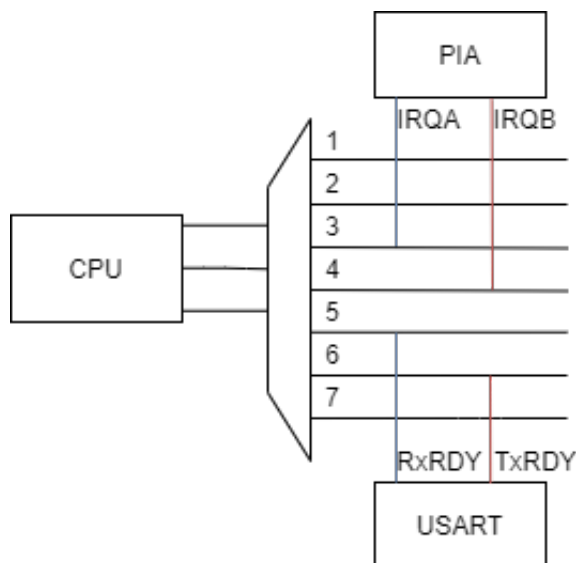


Fig. 2.8: Collegamento tra CPU e periferiche

In entrambe i sistemi le periferiche sono legate alla CPU nello stesso modo, ciò che cambia sono le ISR: nel sistema S1 abbiamo una ISR per la PIA porto A e una vuota per la trasmissione della USART; nel sistema S2 abbiamo solo la ISR per la ricezione della USART. Tale ISR saranno inserite in apposite locazioni del vettore delle interruzioni che si trova nella ROM.

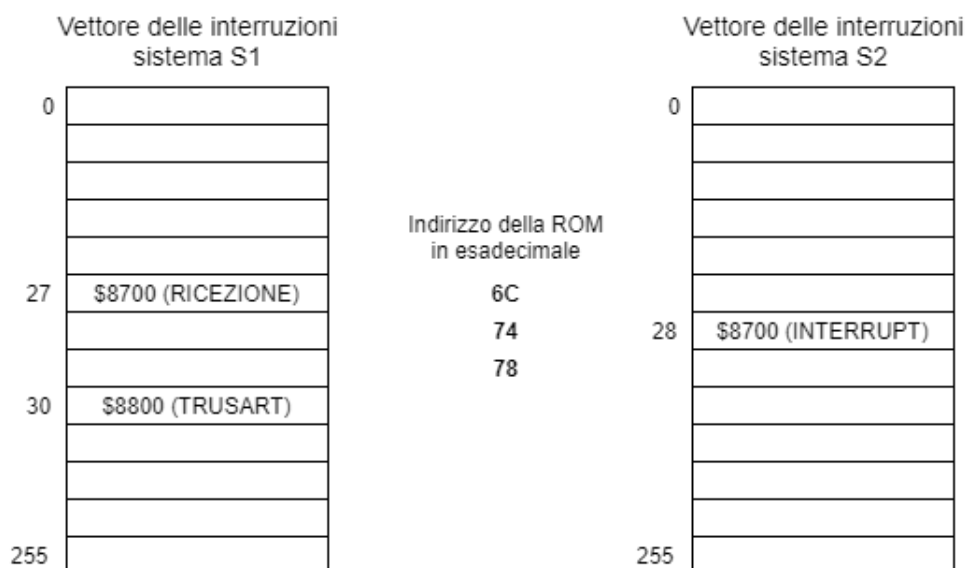


Fig. 2.9: Vettori delle interruzioni

Il vettore delle interruzioni è di 256 locazioni: dalla locazione 24 a 31 ci sono i vettori per i dispositivi gestiti con tecnica autovettorizzata, che contengono l'indirizzo della RAM in

cui si trova la ISR per gestire l'interruzione. L'indirizzo reale della ROM in cui inserire l'indirizzo si ottiene facendo $(24 + X)4$, dove X è la linea d'interruzione a cui è legato il dispositivo da gestire.

2.4 Implementazione

2.4.1 Documentazione del codice

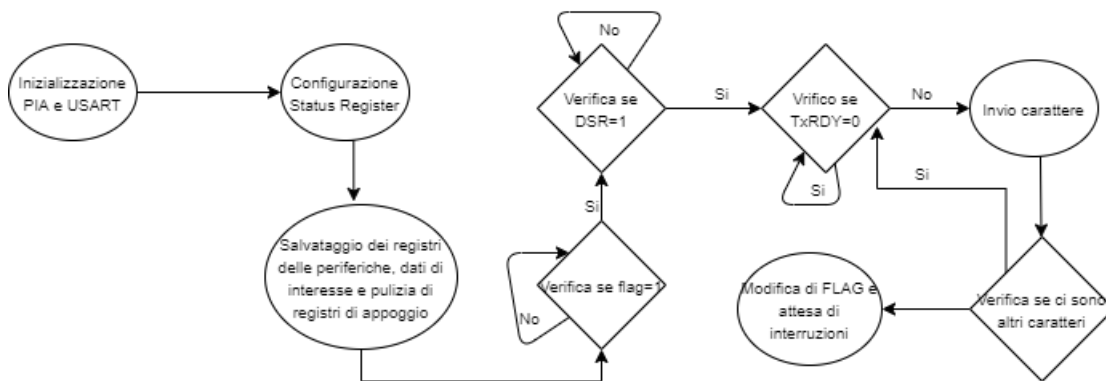


Fig. 2.10: Diagramma di flusso Main S1

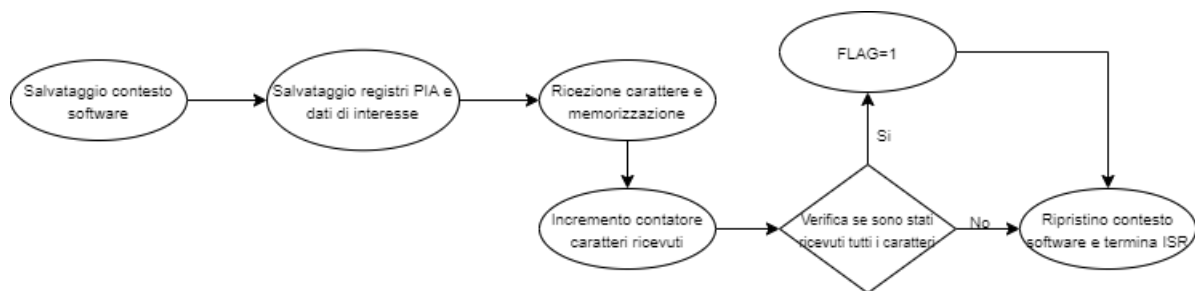


Fig. 2.11: Diagramma di flusso di RICEZIONE di S1

Vorrei centrare l'attenzione su *FLAG*: all'inizio viene posto ad 1 per inviare il messaggio e, una volta inviati tutti i caratteri del messaggio, viene posto a 0. Sarà rimesso ad 1 dalla ISR quando sono stati ricevuti tutti i caratteri dell'eco da S2. Ciò serve a rispettare la richiesta: "In questo caso non è possibile inviare un messaggio sulla seriale se non si è ricevuto prima totalmente il messaggio sulla parallela".



Fig. 2.12: Diagramma di flusso Main S2

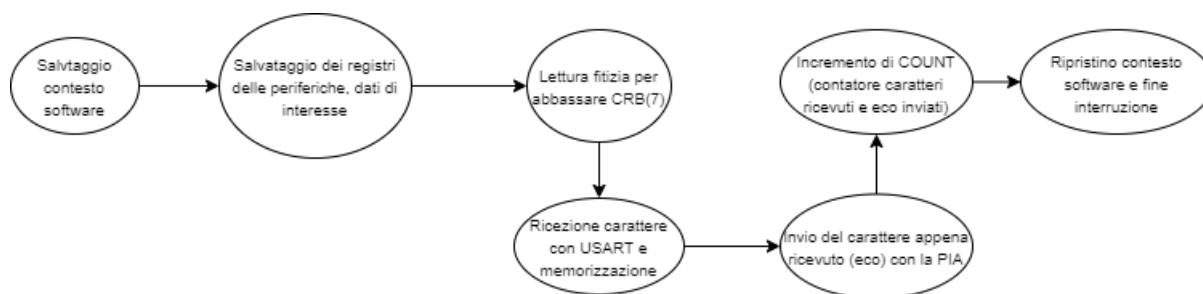


Fig. 2.13: Diagramma di flusso INTERRUPT di S2

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

```

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: I8251USART
Type: Device.    Identif: 04.      BUS: 0001.
Address 1: 00002002.      Address 2: 00002003.
Com1: 0002. Com2: 0005. Com3: 0006. Com4: 0008.

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: 00009000.      Address 2: 00009200.
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: I8251USART
Type: Device.    Identif: 08.      BUS: 0005.
Address 1: 00002002.      Address 2: 00002003.
Com1: 0006. Com2: 0005. Com3: 0006. Com4: 0004.

```

Fig. 2.14: File di configurazione Asim