

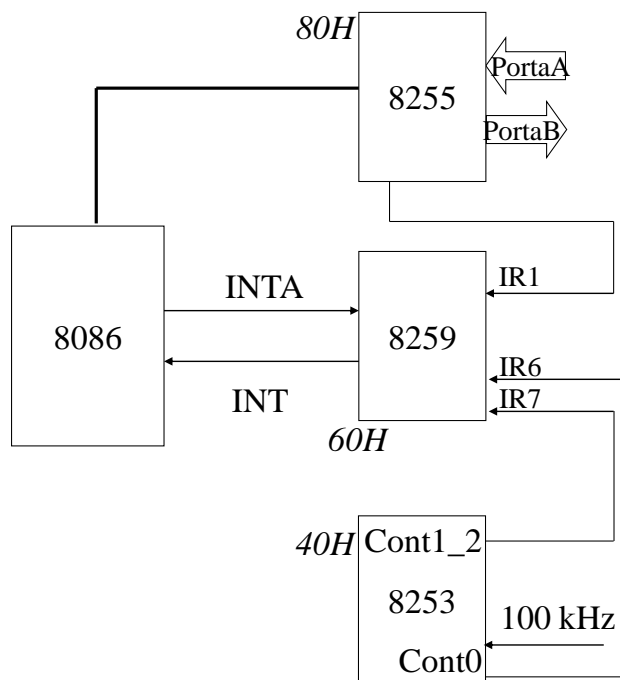
Architetture dei Sistemi di Elaborazione

Durata della prova 1 ora e 45 minuti,
è possibile usare libri e appunti

Esercizio 1 (vale 23 punti)

Si vuole programmare un sistema a processore basato su 8086 nel modo seguente:

1. Il sistema controlla *in polling* se sulla porta A dell'8255, programmata in modo 0, è stato inviato un nuovo dato su 8 bit (ogni cambio di valore corrisponda a un nuovo dato) e memorizza la sequenza in un vettore opportunamente dimensionato (vedi punto seguente)
2. L'acquisizione si interrompe quando sulla porta A dell'8255 è presente un valore già ricevuto; a seguito di questo evento
 - Parte un'attesa di 8 secondi al termine della quale
 - La sequenza ricevuta viene inviata tramite la porta B dell'8255, programmata in modo 1 output (tramite IR1).
3. Una volta svuotato il vettore, il sistema ricomincia a ricevere dati come da punto 1.



NOTA BENE: si tenga conto che nell'8253 il counter 0 è utilizzato singolarmente, mentre counter 1 e 2 sono collegati in cascata; il primo scatena una richiesta di interruzione su IR6, l'altro su IR7.

Si scrivano

- il programma principale
- le procedure di inizializzazione di 8255 e 8253
- le routine di gestione delle interruzioni.

Esercizio 2 (vale 5 punti)

Si consideri un sistema basato su processore ARM che includa un modulo periferico UART. La tabella che segue elenca i registri della porta UART e indica schematicamente il significato di ciascun bit.

Table 74: UART0 Register Map

Name	Description	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	Access	Reset Value*	Address
U0RBR	Receiver Buffer Register	MSB READ DATA LSB								RO	un-defined	0xE000C000 DLAB = 0
U0THR	Transmit Holding Register	MSB WRITE DATA LSB								WO	NA	0xE000C000 DLAB = 0
U0IER	Interrupt Enable Register	0	0	0	0	0	Enable Rx Line Status Interrupt	Enable THRE Interrupt	Enable Rx Data Available Interrupt	R/W	0	0xE000C004 DLAB = 0
U0IIR	Interrupt ID Register	FIFOs Enabled		0	0	IIR3	IIR2	IIR1	IIR0	RO	0x01	0xE000C008
U0FCR	FIFO Control Register	Rx Trigger		Reserved		-	Tx FIFO Reset	Rx FIFO Reset	FIFO Enable	WO	0	0xE000C008
U0LCR	Line Control Register	DLAB	Set Break	Stick Parity	Even Parity Select	Parity Enable	Number of Stop Bits	Word Length Select		R/W	0	0xE000C00C
U0LSR	Line Status Register	Rx FIFO Error	TEMT	THRE	BI	FE	PE	OE	DR	RO	0x60	0xE000C014
U0SCR	Scratch Pad Register	MSB LSB								R/W	0	0xE000C01C
U0DLL	Divisor Latch LSB	MSB LSB								R/W	0x01	0xE000C000 DLAB = 1
U0DLM	Divisor Latch MSB	MSB LSB								R/W	0	0xE000C004 DLAB = 1

Si commenti il seguente codice estratto da un programma completo, descrivendo nel modo più dettagliato possibile quali funzionalità della porta UART sono state usate e con quale scopo.

```

DLAB_BIT          EQU          0x00000080
U0LCR_val         EQU          0x00000003
U0IER_val         EQU          0x00000001
U0DLL_val         EQU          0x6A
U0DLM_val         EQU          0x18

LDR               R0, =U0LCR
LDR               R2, [R0]
TST               R2, #DLAB_BIT
ORREQ             R1, R2, #DLAB_BIT
STR               R1, [R0]
LDR               R0, =U0DLL
LDR               R1, =U0DLL_val
STR               R1, [R0]
LDR               R0, =U0DLM
LDR               R1, =U0DLM_val
STR               R1, [R0]
LDR               R0, =U0LCR
STR               R2, [R0]
LDR               R0, =U0LCR
LDR               R1, =U0LCR_val
STR               R1, [R0]
LDR               R0, =U0IER
LDR               R1, =U0IER_val
STR               R1, [R0]

```