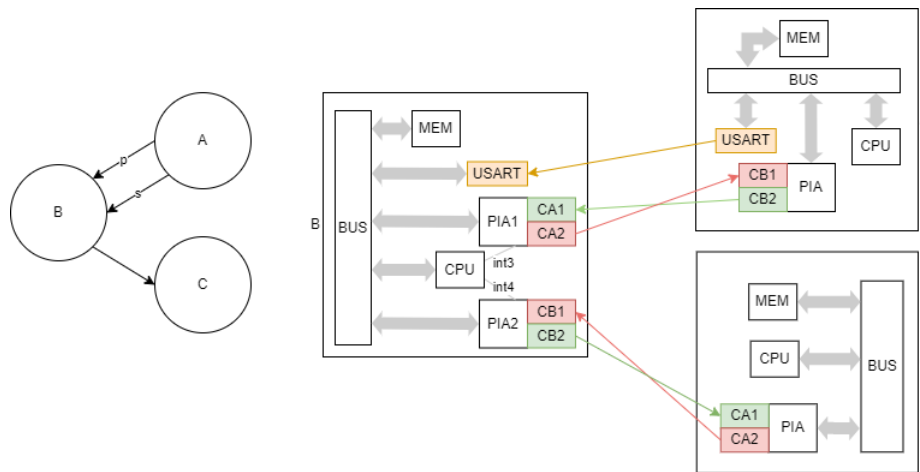




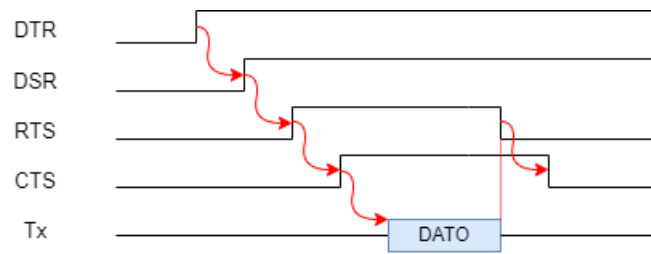
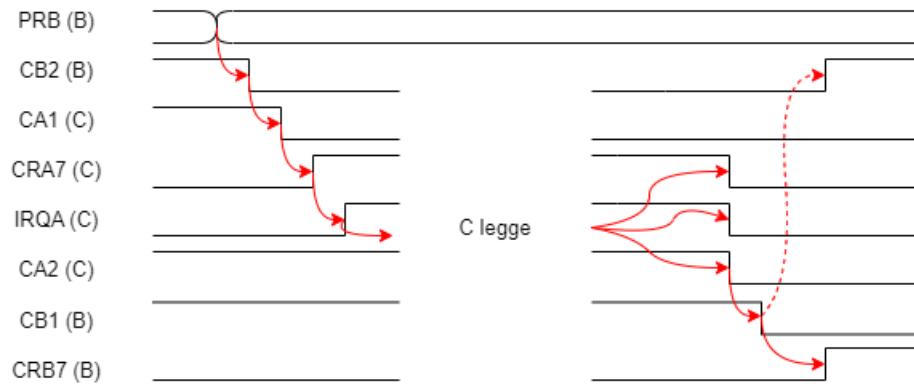
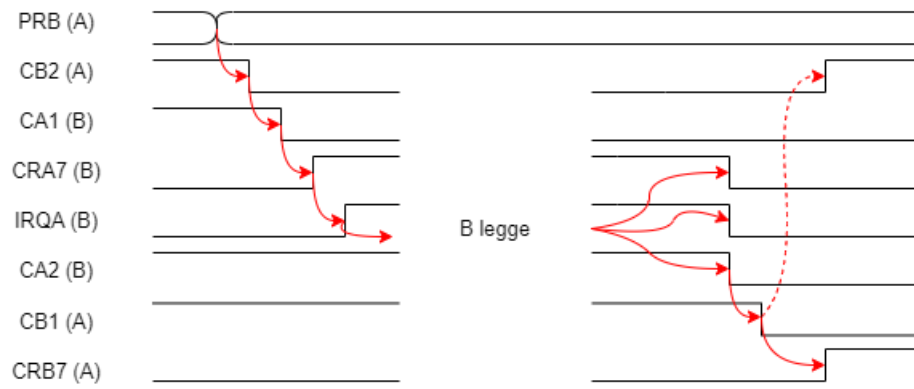
Traccia

Un sistema è composto da 3 unità: A, B e C. A e B sono collegate mediante due periferiche, una parallela e una seriale, mentre B e C tramite una parallela. Il sistema opera come segue:
A invia M messaggi da N caratteri ciascuno a B alternando le due periferiche. Per ogni messaggio ricevuto, B lo invia a C mediante la parallela.

Architettura



Protocolli



Mapa della memoria

MEMORIA RAM		MEMORIA ROM	
\$8000	COUNT_CAR	COUNT_MESS	\$6C 00 00 85 00
\$8002	CHECK	MESSAGE	\$70 00 00 87 00
\$8004	MESSAGE	MESSAGE	
\$8006	MESSAGE		
	...		
\$8100	MAIN		
	...		
\$8500	ISR_B		
	...		

Pseudocodice

Per tale esercizio si è scelto di gestire l'**ipotesi 3**, e quindi regolare lo scambio di messaggi gestendo gli eventuali conflitti tra le ISR dei nodi.

Si è scelto di risolvere tali situazioni mediante l'uso di appositi flag per la gestione delle fasi operative del nodo A. L'uso di tali flag in maniera opportuna rende inutile l'introduzione di un eventuale TAS.

```
1  MAIN(){
2      inizializza PIA 1 in ricezione;
3      inizializza PIA 2 in trasmissione;
4      configurazione USART;
5      Enable stato utente;
6      Enable interruzioni;
7      while(true);           //attendo l'interruzione
8  }
9
10 /*
11 FLAG UTILIZZATI:
12 - sndfromPIA: e' attivo se A ha inviato il carattere mediante la PIA
13 - sndfromUSART: e' attivo se A ha mandato il carattere mediante USART
14 - fwdMSG: e' attivo se devo inoltrare il messaggio a C
15 */
16 ISR_A_PIA(){
17     if(sndfromPIA = true AND fwdMSG = false){
18         leggi da PIA1;
19         count_car ++;
20         if(count_car == N){
21             count_car = 0;
22             count_mess ++ ;
23             fswMSG = true;
24             if(count_mess == M){
25                 disabilita PIA1;
26             }
27         }
28     }
29     return from ISR;
```

```

30 }
31
32 ISR_A_USART(){
33     if(sndfromUSART = true AND fwdMSG = false){
34         leggi da USART;
35         count_car ++;
36         if(count_car == N){
37             count_car = 0;
38             count_mess ++ ;
39             fswMSG = true;
40             if(count_mess == M){
41                 disabilita USART;
42             }
43         }
44     }
45     return from ISR;
46 }
47
48 ISR_C(){
49     if(fwdMSG == true){
50         Scrivi su PIA2;
51         count_car ++;
52         if(count_car == N){
53             count_car = 0;
54             if(sndfromPIA = true){
55                 sndfromPIA = false;
56                 sndfromUSART = true;
57                 controllo se USART ha interrotto;
58             }
59             else if(sndfromUSART = true){
60                 sndfromUSART = false;
61                 sndfromPIA = true;
62                 if(CRA7(A)==1){
63                     leggi da PIA1;
64                     count_car ++ ;
65                 }
66             }
67         }
68     }
69     return from ISR;
70 }
71
72 /* *****
73     VERSIONE CON TAS SU UNA VARIABILE
74     ***** */
75 MAIN(){
76     inizializza PIA 1 in ricezione;
77     inizializza PIA 2 in trasmissione;
78     configurazione USART;
79     Enable stato utente;
80     Enable interruzioni;
81     while(true);           //attendo l'interruzione
82 }
83

```

```

84  /*
85  FLAG UTILIZZATI:
86  - periferica: assume valori PIA(0) e USART(1) per identificare la periferica che sto
      usando
87  - inoltra: e' attivo se devo fare l'inoltro del messaggio a C
88  */
89  ISR_A_PIA(){
90      if(TAS(periferica)==0){
91          if(inoltra = false){
92              Leggi da PIA1;
93              count_car ++;
94              if(count_car == N){
95                  count_car = 0;
96                  count_mess ++;
97                  inoltra = true;
98                  if(count_mess == M){
99                      disattiva PIA1;
100                  }
101                  controlla se USART ha interrotto;
102              }
103          }
104          TAS_unlock;
105      }
106      return from ISR;
107  }
108
109  ISR_A_USART(){
110      if(TAS(periferica)==1){
111          STESSA COSA
112      }
113  }
114
115  ISR_C(){
116      VEDI SOLUZIONE 1;
117  }

```

Programma assembly

1		ORG	\$8000	
2	N	EQU	4	*dimensione di un messaggio
3	M	EQU	10	*numero totale di messaggi
4				
5	* FLAG -->	0=true	1=false	
6	COUNT_CAR	DC.B	0	*contatore per i caratteri
7	COUNT_MESS	DC.B	0	*contatore per i messaggi
8	SNDPIA	DC.B	0	*flag per la verifica della ricezione dalla PIA
9	SNDUSART	DC.B	1	*flag per la verifica della ricezione dall'USART
10	FWDMSG	DC.B	1	*flag per la fase di inoltro del messaggio a C
11	MESSAGGE	DS.B	N	*messaggio da N caratteri
12				
13		ORG	\$8200	
14	USARTD	EQU	\$2000	*registro dato dell'USART

```

15  USARTC      EQU      $2001      *registro modo-controllo dell'USART
16  PIA1D      EQU      $2004      *registro dato di PIA1
17  PIA1C      EQU      $2005      *registro controllo di PIA1
18  PIA2D      EQU      $2006      *registro dato di PIA2
19  PIA2C      EQU      $2007      *registro controllo di PIA2
20
21  MAIN        JSR      SETUP_PIA
22              JSR      SET_USART
23              MOVE.W   SR,D0      *salvo SR in D0 per modificarlo
24              ANDI.W   #$D8FF,D0  *imposto lo stato utente in D0
25              MOVE.W   D0,SR      *pongo il nuovo SR nel registro
26  LOOP        END      LOOP
27
28  SETUP_PIA   MOVE.B   #0,PIA1C    *ricezione con PIA1
29              MOVE.B   #$FF,PIA1D
30              MOVE.B   #00100101,PIA1C
31
32              MOVE.B   #0,PIA2C    *trasmissione con PIA2
33              MOVE.B   #$00,PIA2D
34              MOVE.B   #00100101,PIA2C
35              RTS
36
37  SET_USART   MOVE.B   #01011101,USARTC  *setto msg da 8 bit, comunicazione async
38              MOVE.B   #00110110,USARTC  *imposto l'USART in ricezione
39              RTS
40
41  ISR_A_PIA   ORG      $8500
42              MOVEA.L  A0,-(A7)
43              MOVEA.L  A1,-(A7)
44              MOVEA.L  A2,-(A7)
45              MOVE.L   D0,-(A7)
46              MOVE.L   D1,-(A7)
47              MOVEA.B  #PIA1C,A0
48              MOVEA.B  #PIA1D,A1
49              MOVEA.B  #MESSAGE,A2
50              MOVE.B   SNDPIA,D0
51              MOVE.B   FWDMSG,D1
52              CMP.B    #0,D0
53              BNE      FINE
54              CMP.B    #1,D1
55              BNE      FINE
56              MOVE.B   COUNT_CAR,D0
57              MOVE.B   (A1),(A2,D0)
58              ADD.B    #1,D0
59              MOVE.B   D0,COUNT_CAR
60              CMP.B    #N,D0
61              BNE      FINE
62              MOVE.B   #0,D0
63              MOVE.B   D0,COUNT_CAR
64              MOVE.B   COUNT_MESS,D1
65              ADD.B    #1,D1
66              MOVE.B   D1,COUNT_MESS
67              MOVE.B   FWDMSG,D0
68              MOVE.B   #0,D0

```

```

69      MOVE.B   DO,FWDMSG
70      CMP.B    #M,D1
71      BNE      FINE
72      MOVE.B   #$00,(A0)
73  FINE      MOVE.L   (A7)+,D1
74      MOVE.L   (A7)+,D0
75      MOVEA.L  (A7)+,A2
76      MOVEA.L  (A7)+,A1
77      MOVEA.L  (A7)+,A0
78      RTE
79
80      ORG      $8700
81  ISR_A_USART  MOVEA.L  A0,-(A7)
82      MOVEA.L  A1,-(A7)
83      MOVE.L   D0,-(A7)
84      MOVE.L   D1,-(A7)
85      MOVE.B   #USARTD,A0
86      MOVE.B   #MESSAGE,A1
87      MOVE.B   SNDUSART,D0
88      MOVE.B   FWDMSG,D1
89      CMP.B    #0,D0
90      BNE      FINE
91      CMP.B    #1,D1
92      BNE      FINE
93      MOVE.B   COUNT_CAR,D0
94      MOVE.B   (A0),(A1,D0)      *lettura da USART
95      ADD.B    #1,D0
96      MOVE.B   D0,COUNT_CAR
97      CMP.B    #N,D0
98      BNE      FINE
99      MOVE.B   #0,D0
100     MOVE.B   D0,COUNT_CAR
101     MOVE.B   COUNT_MESS,D0
102     ADD.B    #1,D0
103     MOVE.B   D0,COUNT_MESS
104     CMP.B    #M,D0
105     BNE      FINE
106     MOVE.B   #$00,USARTC
107  FINE      MOVE.L   (A7)+,D1
108      MOVE.L   (A7)+,D0
109      MOVEA.L  (A7)+,A1
110      MOVEA.L  (A7)+,A0
111
112     ORG      $8900
113  ISR_C      MOVEA.L  A0,-(A7)
114      MOVEA.L  A1,-(A7)
115      MOVEA.L  A2,-(A7)
116      MOVE.L   D0,-(A7)
117      MOVE.B   #PIA2D,A0
118      MOVE.B   #PIA2C,A2
119      MOVE.B   #MESSAGE,A1
120      MOVE.B   FWDMSG,D0
121      CMP.B    #0,D0
122      BNE      FINE

```

```

123      MOVE.B  (A1,DO), (A0)
124      MOVE.B  COUNT_CAR,DO
125      ADD.B   #1,DO
126      MOVE.B  DO,COUNT_CAR
127      CMP.B   #N,DO
128      BNE     FINE
129      MOVE.B  #0,DO
130      MOVE.B  DO,COUNT_CAR
131      MOVE.B  SNDPIA,DO
132      CMP.B   #0,DO
133      BNE     ELSEIF
134      MOVE.B  #1,DO
135      MOVE.B  D1,SNDPIA
136      MOVE.B  SNDUSART,DO
137      MOVE.B  #0,DO
138      MOVE.B  DO,SNDUSART
139      check se USART ha interrotto
140      JMP     FINE
141 ELSEIF  MOVE.B  SNDUSART,DO
142      CMP.B   #0,DO
143      BNE     FINE
144      MOVE.B  #1,DO
145      MOVE.B  DO,SNDUSART
146      MOVE.B  SNDPIA,DO
147      MOVE.B  #0,DO
148      MOVE.B  DO,SNDPIA
149      MOVE.B  (A2),DO
150      ANDI.B  #$80,DO
151      CMP.B   #$80,DO
152      BNE     FINE
153      MOVE.B  COUNT_CAR,DO
154      MOVE.B  (A0), (A1,DO)
155      ADD.B   #1,DO
156      MOVE.B  DO,COUNT_CAR
157 FINE    MOVEA.L (A7)+,DO
158      MOVEA.L (A7)+,A2
159      MOVEA.L (A7)+,A1
160      MOVE.L  (A7)+,A0
161
162      END     MAIN

```
