

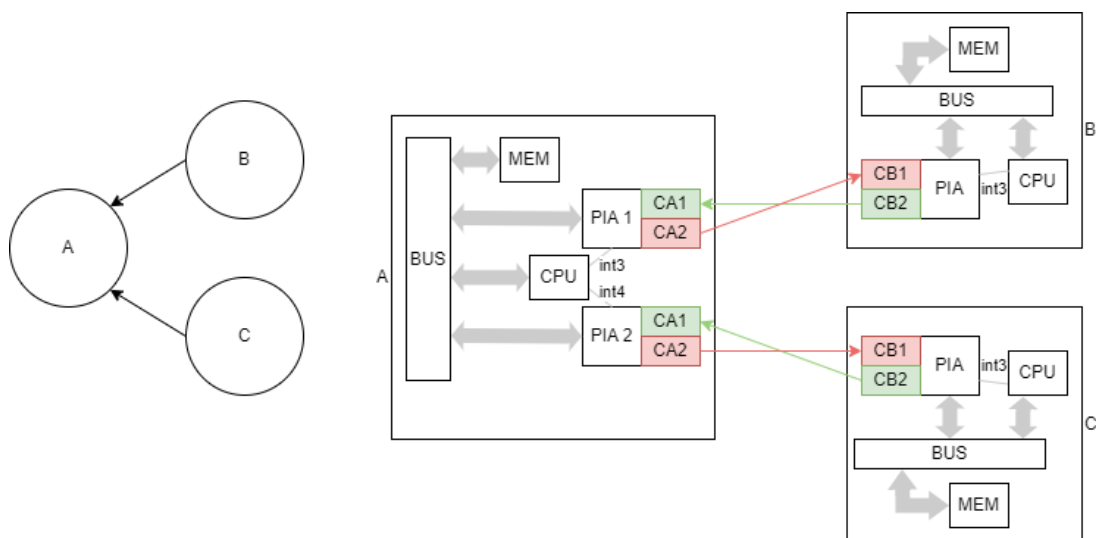


Traccia

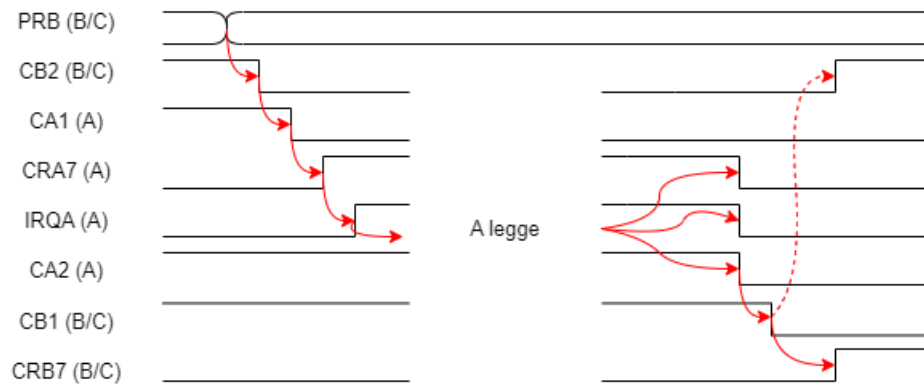
Un sistema è composto da tre unità, A, B e C, tra loro collegate mediante due periferiche parallele che interconnettono A con B e A con C rispettivamente. Il sistema opera come segue:

A riceve e memorizza M messaggi da N caratteri ciascuno, che possono arrivare indifferentemente da B o da C. Cominciati a ricevere caratteri da un nodo, A continuerà la ricezione da tale nodo. Se nel frattempo arrivassero interruzioni da parte dell'altro nodo, A le terrà in "sospeso" e procederà a servirla dopo aver completato la ricezione del messaggio corrente.

Architettura



Protocolli



Mappa della memoria

MEMORIA RAM	
\$8000	LOCK B_ATTIVO
\$8002	COUNT_CAR COUNT_MESS
\$8004	MESSAGE MESSAGE
\$8006	MESSAGE MESSAGE
...	
\$8100	MAIN
...	
\$8500	ISR_B
...	
\$8700	ISR_C
...	

MEMORIA ROM	
\$6C	00 00 85 00
\$70	00 00 87 00

Pseudocodice

```

1  MAIN(){
2      inizializza PIA 1;
3      inizializza PIA 2;
4      Enable stato utente;
5      Enable interruzioni;
6      while(true);           //attendo l'interruzione di uno dei due sistemi
7  }
8
9  ISR_B(){
10     if(tas lock){
11         if(B_attivo = true){
12             leggi da PIA1;
13             count_car ++ ;
14             if(count_car == N){

```

```

15         count_car = 0;
16         count_mess ++ ;
17         B_attivo = false;
18         if(count_mess == M){
19             disattiva PIA1;
20         }
21         if(CRA7(C) == 1){
22             leggi da PIA2;
23             count_car ++ ;
24         }
25     }
26 }
27 }
28 return from ISR;
29
30 ISR_C(){
31     if(tas lock){
32         if(B_attivo = false){
33             leggi da PIA2;
34             count_car ++;
35             if(count_car == N){
36                 count_car = 0;
37                 count_mess ++ ;
38                 B_attivo = true;
39                 if(count_mess == M){
40                     disattiva PIA2;
41                 }
42                 if(CRA7(B) ==1){
43                     leggi da PIA1;
44                     count_car ++ ;
45                 }
46             }
47         }
48     }
49     return from ISR;
50 }

```

Programma assembly

1		ORG	\$8000
2	N	EQU	4
3	M	EQU	10
4			
5	LOCK	DC.B	0
6	B_ATTIVO	DC.B	0
7	COUNT_CAR	DC.B	0
8	COUNT_MESS	DC.B	0
9	MESSAGGE	DS.B	N
10			
11		ORG	\$8100
12	PIA1D	EQU	\$2004
13	PIA1C	EQU	\$2005

```

14 PIA2D      EQU      $2006
15 PIA2C      EQU      $2007
16
17 MAIN       JSR       SETUP_PIA
18             MOVE.W   SR,D0
19             ANDI.W   #$D8FF,D0
20             MOVE.W   D0,SR
21 LOOP       END       LOOP
22
23 SETUP_PIA  MOVE.B    #0,PIA1C
24             MOVE.B    #$00,PIA1D
25             MOVE.B    #%00100101,PIA1C
26             MOVE.B    #0,PIA2C
27             MOVE.B    #$00,PIA2D
28             MOVE.B    #%00100101,PIA2C
29             RTS
30
31             ORG       $8500
32 ISR_B      MOVEA.L   A0,-(A7)
33             MOVEA.L   A1,-(A7)
34             MOVEA.L   A2,-(A7)
35             MOVEA.L   A3,-(A7)
36             MOVE.L    D0,-(A7)
37             MOVEA.L   A4,-(A7)
38
39             MOVE.L    #PIA1D,A0
40             MOVE.L    #PIA1C,A1
41             MOVE.L    #PIA2D,A2
42             MOVE.L    #PIA2C,A3
43             MOVE.L    #MESSAGE,A4
44
45             TAS       LOCK
46             MOVE.B    B_ATTIVO,D0
47             CMP.B     #0,D0
48 * se B_attivo=0 e' il turno di B
49             BNE       FINE
50             MOVE.B    COUNT_CAR,D0
51             MOVE.B    (A0),(A4,D0)
52             ADD.B     #1,D0
53             CMP.B     #N,D0
54             BNE       FINE
55             MOVE.B    #0,COUNT_CAR
56             MOVE.B    #1,B_ATTIVO
57             MOVE.B    COUNT_MESS,D0
58             ADD.B     #1,D0
59             CMP.B     #M,D0
60             BNE       LEGGI_C
61             MOVE.B    #$00,(A1)
62 * controllo se C e' sospeso
63 LEGGI_C    MOVE.B     (A3),D0
64             ANDI.B     #$80,D0
65             CMP.B     #$80,D0
66             BNE       FINE
67 * leggo il carattere da C

```

```

68             MOVE.B  COUNT_CAR,D0
69             MOVE.B  (A2),(A4,D0)
70             ADD.B   #1,COUNT_CAR
71 * ripristino il contesto
72 FINE         MOVEA.L (A7)+,A4
73             MOVE.L  (A7)+,D0
74             MOVEA.L (A7)+,A3
75             MOVEA.L (A7)+,A2
76             MOVEA.L (A7)+,A1
77             MOVEA.L (A7)+,A0
78             RTE
79
80
81             ORG      $8700
82 ISR_C        MOVEA.L A0,-(A7)
83             MOVEA.L A1,-(A7)
84             MOVEA.L A2,-(A7)
85             MOVEA.L A3,-(A7)
86             MOVE.L  D0,-(A7)
87             MOVEA.L A4,-(A7)
88
89             MOVE.L  #PIA1D,A0
90             MOVE.L  #PIA1C,A1
91             MOVE.L  #PIA2D,A2
92             MOVE.L  #PIA2C,A3
93             MOVE.L  #MESSAGE,A4
94
95             TAS     LOCK
96             MOVE.B  B_ATTIVO,D0
97             CMP.B   #1,D0
98 * se B_attivo=1 e' il turno di C
99             BNE     FINE
100            MOVE.B  COUNT_CAR,D0
101            MOVE.B  (A2),(A4,D0)
102            ADD.B   #1,D0
103            CMP.B   #N,D0
104            BNE     FINE
105            MOVE.B  #0,COUNT_CAR
106            MOVE.B  #0,B_ATTIVO
107            MOVE.B  COUNT_MESS,D0
108            ADD.B   #1,D0
109            CMP.B   #M,D0
110            BNE     LEGGI_C
111            MOVE.B  #$00,(A1)
112 * controllo se B e' sospeso
113 LEGGI_C      MOVE.B  (A1),D0
114            ANDI.B  #$80,D0
115            CMP.B   #$80,D0
116            BNE     FINE
117 * leggo il carattere da B
118            MOVE.B  COUNT_CAR,D0
119            MOVE.B  (A0),(A4,D0)
120            ADD.B   #1,COUNT_CAR
121 * ripristino il contesto

```

```
122 FINE      MOVEA.L (A7)+,A4
123          MOVE.L (A7)+,D0
124          MOVEA.L (A7)+,A3
125          MOVEA.L (A7)+,A2
126          MOVEA.L (A7)+,A1
127          MOVEA.L (A7)+,A0
128          RTE
129
130          END      MAIN
```
