Nome: Benfenati Domenico

Prova CSD
9 Maggio 2022

Traccia

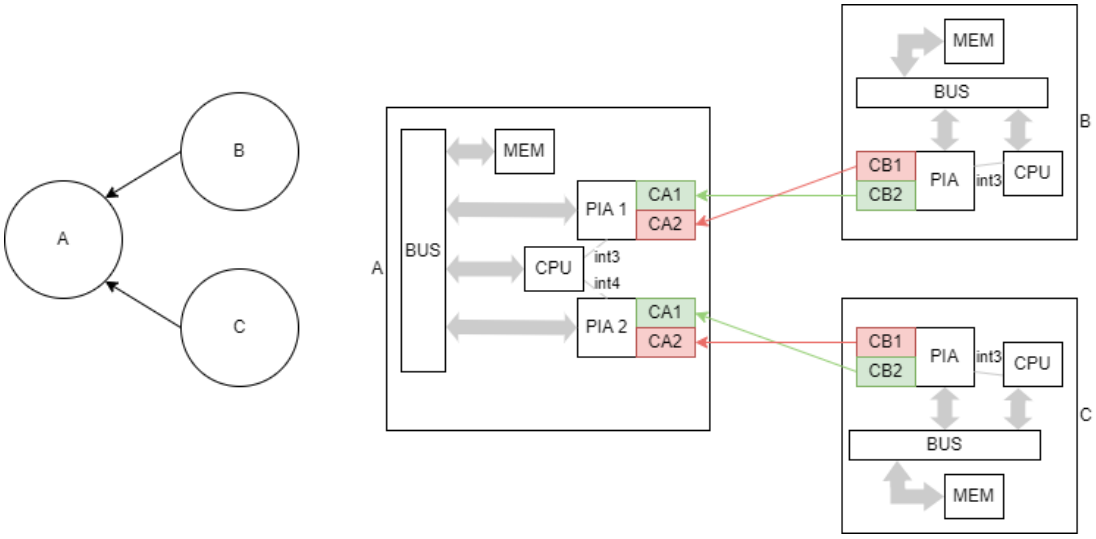
Un sistema è composto da tre unità, A, B e C, tra loro collegate mediante due periferiche parallele che interconnettono A con B e A con C rispettivamente. Il sistema opera in due fasi successive, come descritto di seguito:

- Fase 1: inizialmente A riceve da B un messaggio di N caratteri con N non definito inizialmente, il cui primo carattere è proprio uguale ad N. La ricezione da B deve essere gestita con il meccanismo delle interruzioni. In questa fase si assume che C non possa interrompere A.
- Fase 2: successivamente dopo aver completato la ricezione del primo messaggio da B, A riceve altri T, con T non noto, messaggi di lunghezza N, ricevendo un carattere da B e uno da C in modo alternato, iniziando da B.

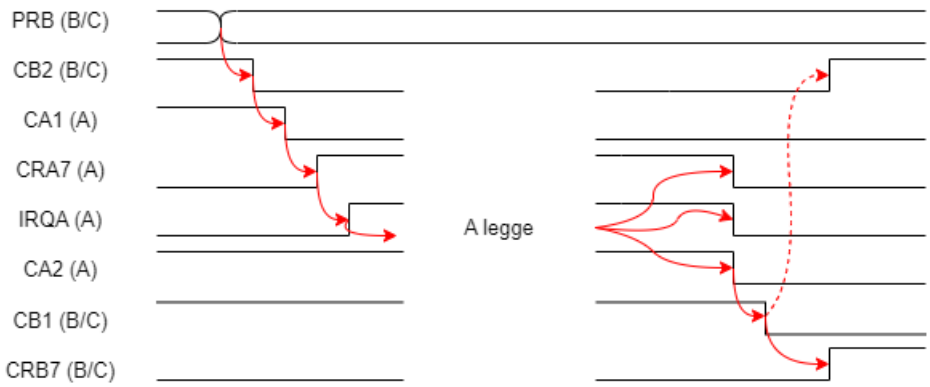
Per la Fase 2 è possibile considerare due diverse ipotesi di funzionamento:

- **ipotesi semplificativa**: B e C si coordinano esternamente, e B ha una priorità superiore rispetto a C. Vengono prima inviati tutti i messaggi da B e poi tutti i messaggi da C.
- **ipotesi standard**: non è possibile stabilire a priori l'ordine di arrivo dei caratteri da B e C; A deve gestire in SW l'alternanza.

Architettura



Protocolli



Mappa della memoria

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(fase1){
11         ricevi Carattere da PIA1;
12         if (primo = true){
13             salva N;
14             primo = false;      //primo verifica che sia il primo carattere
15         }
16         num_car ++ ;    //num_car e' una variabile comune che conta i caratteri ricevuti
               da B e da C insieme
17     }
18     else{
19         if (attiva_C = false){
20             ricevi Carattere da PIA1;
21             num_car ++ ;
22             attiva_C = true;
23             if (num_car == N){
24                 num_car = 0;
25                 num_mess ++ ;   //num_mess tiene traccia dei T messaggi ricevuti
26                 if(num_mess == T){
27                     disattiva PIA1;
28                     disattiva PIA2;
29                 }
30                 if(CRA7C == 1){
```

```
                    ricevi Carattere da PIA2;
                    num_car ++ ;
                }
            }
        }
    }
}

ISR_C(){
    if(!fase1){
        if(attiva_C = true){
            ricevi Carattere da PIA2;
            num_car ++ ;
            attiva_C = false;
            if (num_car == N){
                num_car = 0;
                num_mess ++;
                if(num_mess == T){
                    disattiva PIA1;
                    disattiva PIA2;
                }
                if(CRA7B == 1){
                    ricevi Carattere da PIA1;
                    num_car ++ ;
                }
            }
        }
    }
}
```

Programma assembly