

Moduli intercomunicanti

di Anna Pugliese

In questo numero di «Appunti di Informatica» cercheremo di dare un'occhiata alle comunicazioni. La cosa in sé non è delle più semplici in quanto sotto la voce comunicazioni possono trovarsi la maggior parte dei concetti dell'informatica di base; d'altro canto è sicuramente interessante se si pensa al numero di comunicazioni che sono necessarie per realizzare funzionalità anche abbastanza semplici. Le comunicazioni insomma, stanno alla base del funzionamento dei computer. Esse si dividono in particolari classi di comunicazioni, in base al modello utilizzato ed al tipo di oggetti che le usano. È evidente l'impossibilità di farne una trattazione esauriente, per cui lo scopo che ci prefiggiamo è quello di una introduttiva presentazione generale seguita dalla dettagliata illustrazione di una particolare forma di comunicazione: quella tra unità di elaborazione

I livelli di comunicazione

La parola comunicazione sta ad indicare un processo di trasferimento di informazione fra due o più entità.

Determinare quali siano le entità comunicanti è necessario per comprendere il tipo di comunicazione con cui si ha a che fare. Per chiarire questo concetto, può essere utile rifarsi alla più naturale delle comunicazioni, quella fra persone.

È noto come due persone possano scambiarsi informazioni utilizzando svariate forme di comunicazione: il linguaggio, i gesti, gli sguardi ed altro; per fare un esempio strano due persone possono scambiarsi informazione persino con il silenzio.

Supponiamo di voler chiedere a qualcuno che ore sono. Possiamo anzitutto scegliere la forma di comunicazione, decidendo ad esempio di pronunciare la frase «che ore sono?» oppure (... non troppo fine) fare un breve fischio per attirare su di noi lo sguardo del nostro interlocutore e subito dopo battere l'indice della mano destra due volte sulla parte superiore del polso sinistro.

Ma qualunque sia la forma di comunicazione scelta, noi la stiamo trattando come comunicazione fra due persone; in realtà essa, per essere realizzata, necessita di una serie di scambi informativi di più basso livello, vale a dire di

comunicazioni tra moduli quali il cervello, con la bocca, la lingua e le corde vocali, per la forma di comunicazione vocale, mentre per quella gestuale saranno interessati soprattutto gli arti superiori.

È inutile spingersi oltre nell'osservazione di questo straordinario uniprocessor sul quale gira un sistema operativo distribuito (vedi capacità di lavorare in équipe) e multiprogrammato in time-sharing, che è l'uomo, sul cui funzionamento sono necessarie infinite astrazioni per poter ottenere schemi finiti. Facciamo però tesoro di ciò che abbiamo capito: due entità A e B necessitano, per comunicare tra loro, di un protocollo di comunicazione di livello $L(A)=L(B)$ il cui funzionamento è realizzato mediante protocolli di più basso livello.

Il livello di comunicazione è dunque determinato dal tipo di entità che lo usa; due entità sono dette dello stesso livello se hanno uguale livello di comunicazione, e solo in questo caso possono servirsi del corrispondente protocollo.

I modelli di comunicazione

Stabilito che sotto la voce «comunicazione» sono compresi tanti livelli di comunicazione, vediamo ora come una ulteriore classificazione (trasversale alla precedente) possa essere fatta in base ad una caratteristica del modo in cui lo

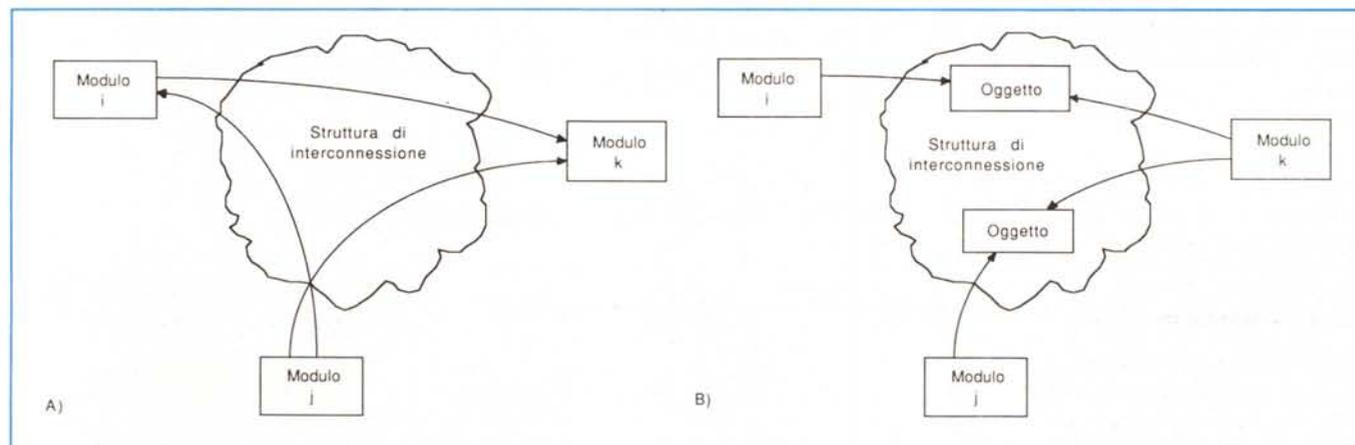


Figura 1 - La figura 1a mostra lo schema di comunicazione con modello ad ambiente locale, in cui la struttura di interconnessione è costituita da linee di comunicazione. In figura 1b, la struttura di interconnessione, che si riferisce al modello ad ambiente globale, è basata su procedure comuni la cui esecuzione, modificando il valore di un certo oggetto, riflette l'informazione associata a quel valore.

scambio informativo avviene. Come al solito, il paragone con il reale è tanto utile quanto poco significativo.

Pensiamo alle solite due persone che vogliono comunicare.

È certamente degna di considerazione la differenza esistente tra una comunicazione diretta (ad esempio quella vocale o gestuale) in cui lo scambio informativo avviene tra il mittente ed il destinatario senza l'intervento di terzi, ed una comunicazione indiretta (ad esempio «ti lascio un messaggio sulla scrivania») in cui si sfrutta uno strumento di comunicazione estraneo ai due interlocutori (un bel foglietto di carta!). Bene, questa diffe-

trattazione ed iniziare a restringerla. Facciamo allora una puntualizzazione: consideriamo entità interlocutori, a qualsiasi livello appartengono purché uguali tra loro, qualunque oggetto possa essere considerato un «modulo» in base alla seguente definizione:

«un MODULO è un'entità attiva ed autonoma, internamente sequenziale, e definita indipendentemente».

Così, se prendiamo in considerazione il livello del sistema operativo, avremo che i moduli saranno nient'altro che i PROCESSI e le comunicazioni fra processi saranno implementate tramite CANALI (ambiente locale) oppure tramite

MONITOR (ambiente globale).

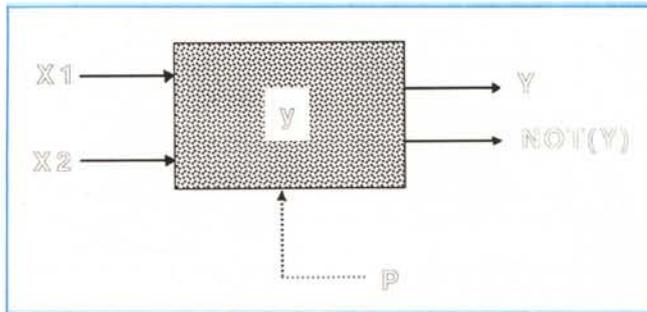
Il livello che vogliamo prendere in considerazione è quello del firmware, in cui un modulo corrisponde ad un'unità di elaborazione.

Il flip-flop

Un'unità di elaborazione si serve, per comunicare, di un'interfaccia costituita essenzialmente da un registro, due indicatori di sincronizzazione ed opportuni collegamenti. Elemento base per la realizzazione di questa interfaccia è il cosiddetto elemento di memoria o «flip-flop». In figura 2 è mostrato lo schema generale di un flip-flop. I suoi morsetti sono costituiti da un segnale P, mediante il quale giungono al flip-flop gli impulsi generati da un clock, e da quattro segnali a livelli: due ingressi, x1 e x2 (alcuni tipi ne hanno uno solo), e due uscite Y e NOT(Y) (che chiaramente hanno sempre valori opposti). Questi elementi sono casi particolari di reti sequenziali, e perciò sono caratterizzati da uno stato interno che indicheremo con il simbolo «y».

Le caratteristiche funzionali essenziali di un elemento di memoria M, sono le seguenti:

Figura 2 - Flip-flop.



renza riflette, nel campo dell'informatica, i due fondamentali modelli di comunicazione possibili: l'ambiente locale e quello globale. Per modello di comunicazione all'ambiente locale, si intende un principio di funzionamento dello scambio informativo, basato sulla pura e semplice trasmissione del «valore» dell'informazione. Quando lo scambio avviene sfruttando oggetti che sono estranei sia al mittente che al destinatario, e quindi pubblici, si parla invece di modello ad ambiente globale. Detto in parole semplici: si parla di ambiente globale, tutte le volte che la comunicazione avviene sfruttando memoria comune ai due interlocutori (vedi figura 1).

Il punto

Prima di scendere nei dettagli di un particolare protocollo di comunicazione, facciamo il punto della situazione. Oltre che suddivise in livelli e modelli, le comunicazioni possono essere classificate in base a molti altri criteri, ottenendo una vastissima gamma di tipi di comunicazione in cui la sincronia o asincronia e la simmetria o asimmetria delle stesse rappresentano due ulteriori aspetti, di primaria importanza. Da questo momento in poi, però, le dimensioni dello spazio a nostra disposizione ci consigliano di smettere di ampliare la

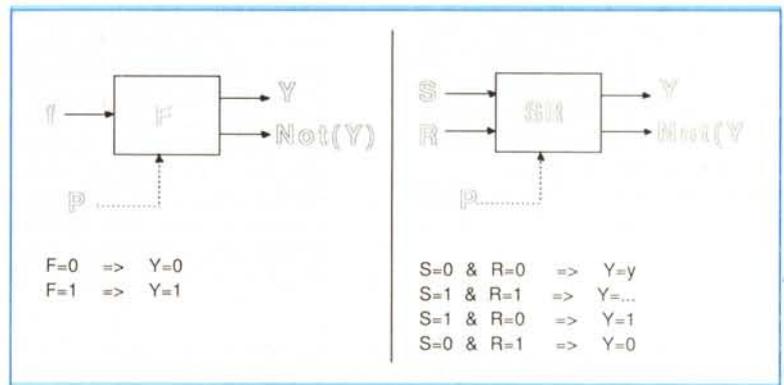


Figura 3 - Schema dei flip-flop F (elemento di registro) ed SR (Set Reset).

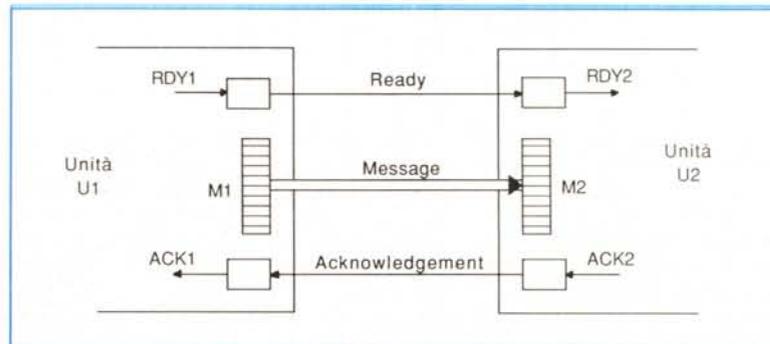


Figura 4 - I collegamenti dedicati e l'interfaccia di comunicazione sincrona tra due unità di elaborazione: U1 unità mittente ed U2 unità destinataria.

- 1) gli ingressi e le uscite di M sono segnali a livelli detti variabili di eccitazione di M;
- 2) quando l'impulso di sincronizzazione non è applicato al terminale P (cioè P = 0), M non cambia il suo contenuto;
- 3) il valore dell'uscita Y di M coincide con il contenuto «0» o «1» di M.

A questo punto per definire un particolare tipo di flip-flop basta specificare come il suo stato interno varia in dipendenza dei valori applicati ai morsetti d'ingresso. Fra i diversi tipi di elementi di memoria ne descriveremo due: il flip-flop «F» e quello «SR». La figura 3 mostra lo schema di questi due elementi. Il flip-flop «F» è il più importante essendo l'elemento base dei registri di memoria; esso funziona in modo che lo stato interno y, e quindi l'uscita Y assuma il valore dell'ingresso F ogni volta che l'impulso è applicato a P. Il flip-flop «SR» la cui sigla sta per «Set Reset», serve invece a memorizzare «1» in y se S è R uguale a «1», mentre y assume «0» se è a valere «1». Ma smettiamola di perderci dietro queste definizioni e vediamo di usare i flip-flop parlando finalmente di interfacce di comunicazione.

```

SEND (U2,msg) ::=
begin
M1:=msg;
RDY1:=1;
wait until ACK1=1;
RDY1=0;
wait until ACK1=0;
end

RECEIVE (U1,vtg) ::=
begin
wait until RDY2=1;
vtg:=M2;
ACK2=1;
wait until RDY2=0;
ACK2=0;
end
    
```

Figura 5

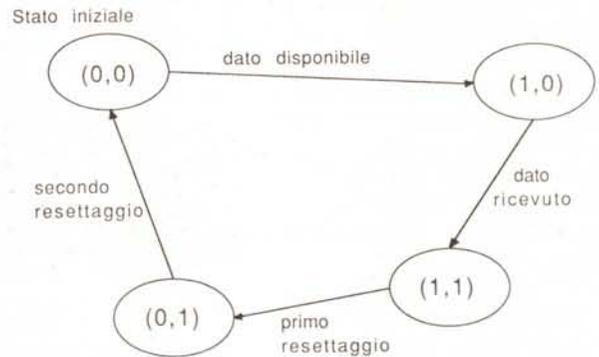


Figura 6
Il grafo in figura riporta le transizioni di stato dei segnali di Ready e di Acknowledgement (i cui valori sono rispettivamente il primo ed il secondo elemento della coppia) durante una comunicazione simmetrica sincrona.

Comunicazione simmetrica, sincrona a rendez-vous stretto

Teniamo sott'occhio la figura 4. Essa mostra il supporto fisico necessario affinché l'unità U1 possa inviare messaggi all'unità U2. Per permettere l'invio di messaggi da U2 ad U1 occorre aver un ulteriore supporto sistemato nel verso opposto. Il tipo di comunicazione che stiamo esaminando è detto simmetrico perché ha un solo mittente ed un solo destinatario, sincrono perché come vedremo le due unità dovranno sincronizzarsi per scambiarsi il messaggio, ed utilizza un protocollo di sincronizzazione detto «a rendez-vous stretto» che in parole povere può essere definito: di tipo breve. Il supporto fisico consiste di quattro indicatori di sincronizzazione, RDY1, RDY2, ACK1, ACK2, che sono realizzati mediante flip-flop di tipo «F».

L'unità U1, che con riguardo all'interfaccia considerata è quella mittente, può assegnare il valore «0» o «1» a RDY1 che è l'ingresso di un flip-flop; il valore assegnato all'ingresso, in seguito al successivo impulso verrà memorizzato come stato interno del flip-flop stesso, e prodotto in uscita; questa uscita è collegata fisicamente all'unità U2 dove costituisce l'ingresso di un altro flip-flop la cui uscita è rappresentata dal morsetto RDY2, che U2 può leggere. Analogo è il discorso per ACK1 ed ACK2 anche se

nel verso opposto. Per quanto riguarda M1 ed M2, essi sono due registri, composti quindi da 8 o 16 o quanti se ne vuole flip-flop «F» allacciati. U1 può scrivere in M1 qualsiasi valore, mentre U2 può spostare il valore di M2 in qualsiasi altro registro e da quest'ultimo, o direttamente da M2, leggerlo.

A questo punto possiamo passare al vero e proprio algoritmo di comunicazione che è riportato in figura 5. L'algoritmo, scritto in un linguaggio che ha la sola pretesa di essere sufficientemente espressivo, dovrà essere implementato mediante microprogrammazione nelle due unità di elaborazione (la procedura SEND in U1 e quella di RECEIVE in U2). Le due procedure sono scritte in modo da realizzare la sincronizzazione necessaria tra le due unità, facendo attendere la prima fino a quando anche la seconda è giunta ad eseguire la procedura corrispondente. U2 comincia la ricezione attendendo che il valore di RDY2 venga posto ad 1, la qual cosa esprime la disponibilità del messaggio in M2 essendo l'istruzione RDY1:=1 eseguita solo dopo che U1 ha posto il messaggio in M1. U2 allora può trasferire il valore di M2 nel registro targa che vuole e che abbiamo indicato con VTG, ponendo in seguito a tale operazione l'indicatore ACK2 al valore 1 per segnalare ad U1 l'avvenuta ricezione.

Notare che ACK è un'abbreviazione per «Acknowledgment», che vuol dire, tra l'altro, «ringraziamento». Ora manca solo la fase di resettaggio degli indicatori, che avviene prima in U1, per RDY1, e poi in U2, per ACK2. A questo punto la sincronizzazione tra le due unità è terminata e gli indicatori sono pronti per eventuali successive comunicazioni. La figura 6 contiene il grafo di stato dei segnali RDY ed ACK. Il tipo di comunicazione visto, costituisce un caso particolare fra unità di elaborazione. Utilizzando un supporto fisico diverso (come ad esempio flip-flop di tipo «contatore binario», registri di risposta, bus e corrispondenti arbitri etc.) è possibile implementare, mediante opportuni microprogrammi, altri algoritmi di comunicazione, tutti simili, in essenza, a quello di figura 5. La scelta del tipo di comunicazione più adatto è subordinata alla funzionalità che si intende svolgere mediante la comunicazione stessa. Avremo così che in un multiprocessor, le comunicazioni fra i processori saranno solitamente di tipo asimmetrico sincrono, mentre quelle tra il processore ed un driver di I/O, realizzate come unità a sé stante, saranno piuttosto di tipo simmetrico asincrono, basate sull'utilizzazione di un buffer nell'unità destinataria che permette al processore di svincolarsi dal sincronizzarsi con unità notoriamente più lente. **MC**