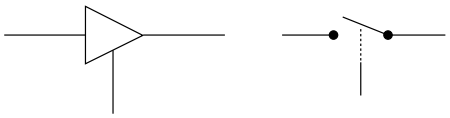
Registri e circuiti combinatori vengono spesso raggruppati condividendo un certo insieme di connessioni (fili). Questo insieme di connessioni costituisce quello che è noto come ***bus***. Un bus nel quale diversi componenti hanno facoltà di scrivere e di leggere è solitamente un «bus dati», perché serve allo scambio di informazioni tra questi componenti; tuttavia, va osservato che solo un componente alla volta può scrivere, mentre non ci sono limitazioni per la lettura concorrenziale.

Bus con il buffer a tre stati

La realizzazione di un bus dati può avvenire con l'ausilio di un multiplatore o di un «buffer a tre stati» (*tri-state buffer*), ovvero un componente che funziona come porta non invertente, ma con un ingresso addizionale, con il quale è possibile abilitare il funzionamento in qualità di porta non invertente (*buffer*), oppure si può isolare completamente l'uscita. Nel disegno seguente si vede a confronto il simbolo del buffer a tre stati con quello che farebbe un interruttore tradizionale:



Il buffer a tre stati, oppure il multiplatore, servono quindi per controllare la scrittura nel bus; tra le due opzioni, la scelta del buffer a tre stati è quella più economica e più semplice, perché il multiplatore comporta l'uso di molte porte logiche e, di conseguenza, introduce un ritardo di propagazione maggiore.

Nelle figure successive si vede l'adattamento di un registro semplice, con buffer a tre stati, per il collegamento a un bus; poi si vede come si possono collegare registri o altri componenti così predisposti, distinguendo tra un bus dati e un bus di controllo.

Figura u103.2. Registro semplice adattato per connettersi con un bus dati.

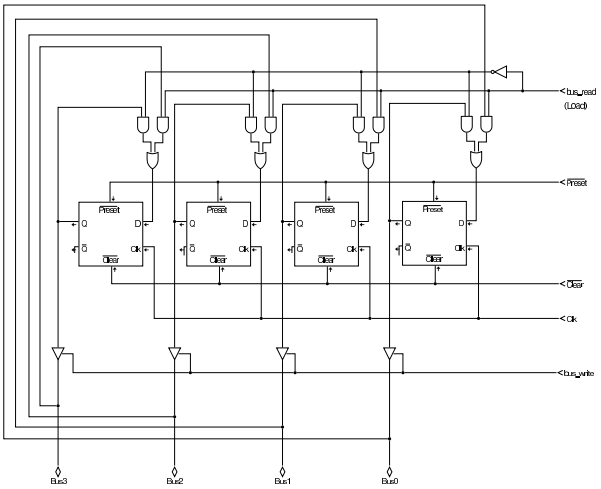


Figura u103.3. Schemi più semplici dell'adattamento di un registro. L'ingresso ***bus_read***, o ***br***, richiede al registro il caricamento del valore che si può leggere dal bus, mentre l'ingresso ***bus_write***, o ***bw***, concede al registro di immettere i propri dati nel bus.

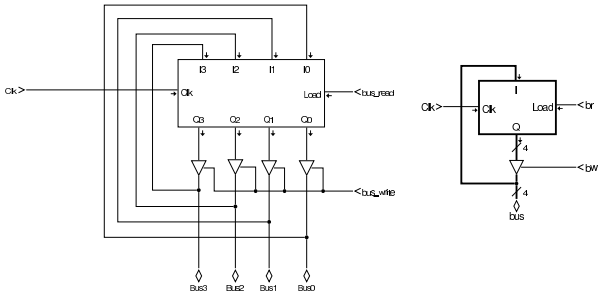
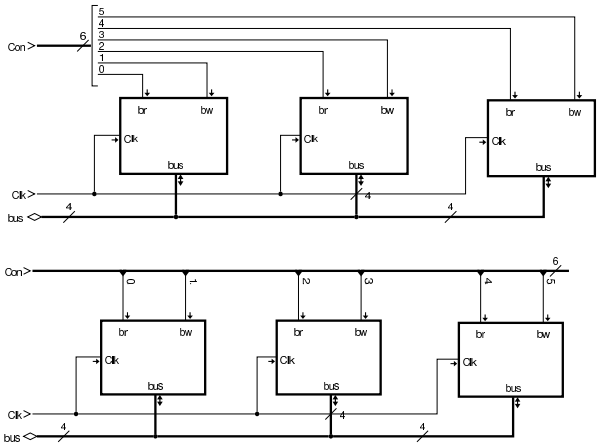


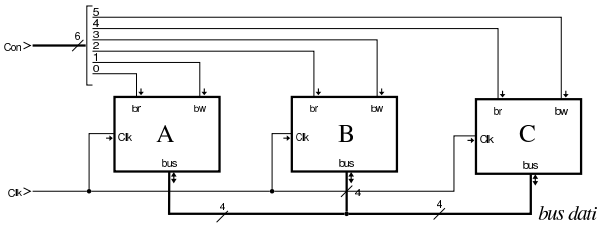
Figura u103.4. Collegamento di più componenti in un bus: gli ingressi che abilitano la lettura dal bus o la scrittura nel bus, assieme ad altri ingressi di controllo eventuali, si collegano a un bus di controllo secondario. Nel secondo disegno si vede un modo alternativo di rappresentare il collegamento al bus di controllo, prelevando un filo alla volta.



Unità di controllo del bus

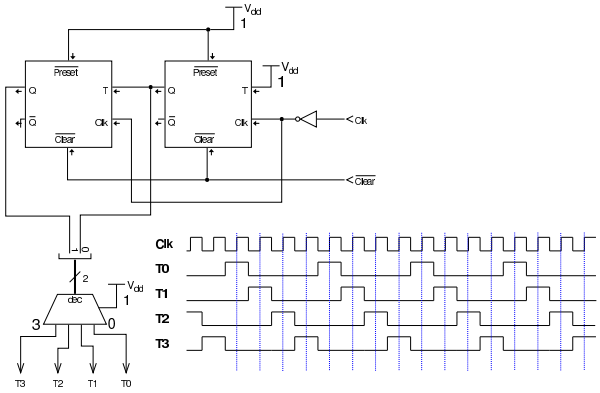
Per dirigere correttamente l'utilizzo di un bus dati, è necessario un sistema di controllo, attraverso il quale scandire le fasi di ogni procedimento che si intende applicare. Per comprendere il meccanismo si riprende lo schema già apparso in cui tre componenti sono connessi su un bus dati e sono controllati da un bus di controllo; su questi componenti si ipotizzano soltanto operazioni di trasferimento delle informazioni.

Figura u103.5. Esempio di riferimento con tre componenti che condividono un bus dati.



Per scandire le operazioni da svolgere, l'unità di controllo ha bisogno di un contatore. Nella realizzazione circuitale di un'unità di controllo si fa riferimento normalmente a un contatore a scorrimento, il quale può essere realizzato come si vede nella figura successiva. Ciò che è importante e che si evidenzia nel tracciato che appare nella figura, è che le uscite risultino attive a cavallo dell'impulso di clock usato per pilotare i componenti del bus. Per ottenere questo risultato, l'impulso di clock che viene usato dal contatore viene invertito.

Figura u103.6. Contatore a scorrimento, usato per l'unità di controllo del bus di esempio: il segnale di clock viene invertito, in modo tale che le uscite T_n siano attive a cavallo del margine positivo che viene usato dai componenti del bus dati. Video: *http://www.youtube.com/watch?v=vXco4E4SNTo* .



L'unità di controllo, realizzata attraverso circuiti logici, deve mettere assieme un decodificatore della funzione da applicare con le uscite del contatore a scorrimento. Per esempio, seguendo lo schema della figura successiva, si vede che la funzione f_1, con la quale si vuole copiare il valore del registro ***A*** nel registro ***B*** e poi dal registro ***B*** nel registro ***C***, il decodificatore attiva la seconda linea a partire dall'alto (etichettata con f_1); quindi, attraverso il collegamento di porte AND, si fa in modo di attivare le uscite ***Aw*** e ***Br*** nel momento in cui il contatore a scorrimento ha l'uscita T_0 attiva; quindi, analogamente, si fa in modo di attivare le uscite ***Bw*** e ***Cr*** nel momento in cui il contatore a scorrimento ha l'uscita T_1 attiva. ***Aw***, ***Br***, e ***Bw*** e ***Cr***, attivate secondo la scansione descritta, vanno ad attivare gli ingressi di lettura-scrittura dei registri rispettivi, consentendo il trasferimento di dati previsto. La realizzazione dell'unità di controllo della figura successiva è estremamente semplificata e il tempo T_2 non viene nemmeno utilizzato, comportando quindi un'attesa inutile in quel momento; tuttavia, il tempo T_3 serve invece per bloccare il segnale di clock nell'unità di controllo, la quale richiede di essere azzerata per poter recepire un nuovo comando dall'ingresso ***f*** .

Figura u103.7. Unità di controllo: sono previste solo quattro funzioni molto semplici, per le quali bastano solo due tempi; pertanto il tempo T_2 rimane inutilizzato e il tempo T_3 serve a bloccare l'unità di controllo fino a quando l'ingresso ***Run*** viene azzerato e poi riattivato. Video: *http://www.youtube.com/watch?v=r-DZgjy-ya0* .

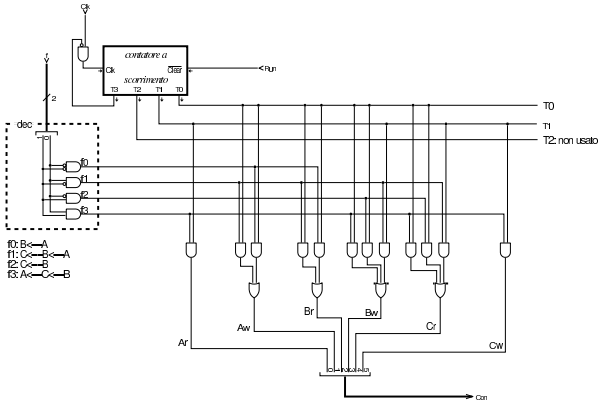
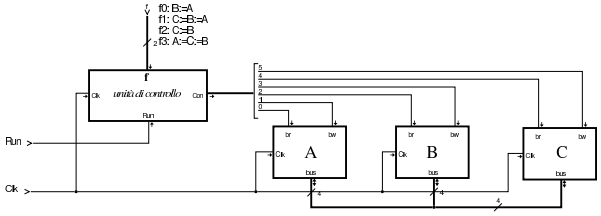


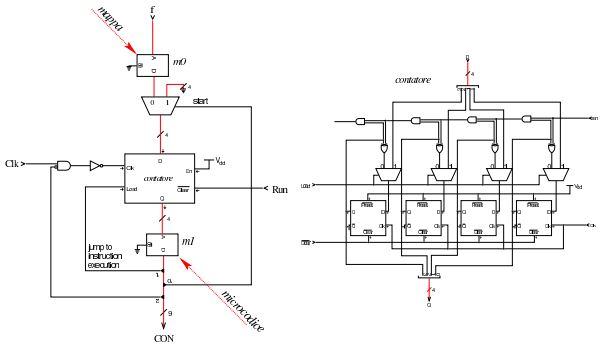
Figura u103.8. Unità di controllo connessa ai componenti che deve pilotare. Video: *http://www.youtube.com/watch?v=kpET2kEcUIo* .



Microcodice

La realizzazione di un'unità di controllo di un bus dati, sotto forma di circuito logico, può risultare in un lavoro estremamente complesso. Per questa ragione, si utilizza solitamente una memoria ROM (in sola lettura), suddivisa in due parti: ***mappa*** e ***microcodice***. Le due parti della memoria ROM vanno viste come due tabelle: la prima traduce la funzione desiderata in un indirizzo che punta alla seconda, dove inizia il codice che descrive i vari passaggi da eseguire. Si osservi la figura successiva, dove a sinistra c'è lo schema dell'unità di controllo e a destra c'è lo schema del contatore utilizzato all'interno della stessa.

Figura u103.9. Unità di controllo realizzata attraverso memorie ROM: a sinistra lo schema a blocchi dell'unità, a destra la scomposizione del contatore a quattro bit (un semplice contatore incrementante, con possibilità di caricare un valore, basato su flip-flop D). Video: *http://www.youtube.com/watch?v=mBGhNP3Uujs* .



Le due memorie ROM sono denominate, rispettivamente, ***m0*** ed ***m1***. Nella figura successiva si vede il contenuto delle due memorie, rappresentato in forma tabellare, mettendo in corrispondenza l'indirizzo in ingresso della memoria (ingresso ***A***) e il contenuto che viene rappresentato nell'uscita (***D***). La memoria ***m0*** di questi esempi utilizza solo due bit per gli indirizzi, i quali corrispondono alla funzione richiesta all'unità di controllo; in uscita, la stessa memoria, produce un valore a quattro bit che rappresenta, a sua volta, un indirizzo per la memoria ***m1***.

Figura u103.10. Contenuto e collegamento tra le memorie ROM ***m0*** e ***m1*** che compongono il microcodice.

m_0

	indirizzo	contenuto	
f_0	0	0001	(1)
f_1	1	0011	(3)
f_2	2	0110	(6)
f_3	3	1000	(8)

m_1

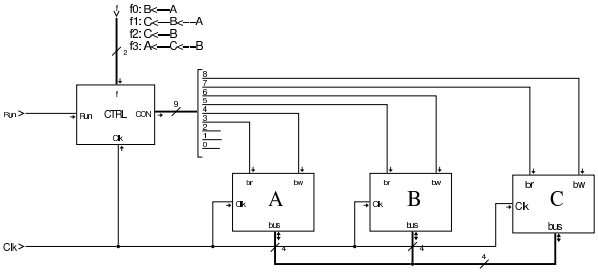
indirizzo	contenuto	
0	000000010	*load counter*
1	000110000	*B <-- A*
2	000000111	*load zero, stop clock*
3	000110000	*B <-- A*
4	011000000	*C <-- B*
5	000000111	*load zero, stop clock*
6	011000000	*C <-- B*
7	000000111	*load zero, stop clock*
8	011000000	*C <-- B*
9	100001000	*A <-- C*
10	000000111	*load zero, stop clock*

L'uscita della memoria ***m0*** viene inviata a un contatore; l'uscita del contatore serve ad accedere alla memoria ***m1***; l'uscita della memoria ***m1*** è ciò che serve per controllare il bus, con l'aggiunta di qualche linea per controllare la stessa unità. Per comprendere cosa succede, vengono scanditi i vari passaggi nell'elenco successivo, ipotizzando che sia richiesta la funzione f_1.

1. All'avvio l'ingresso ***Run*** è a zero, cosa che comporta l'azzeramento del contatore. Così facendo, dalla memoria ***m1*** viene selezionato l'indirizzo zero, dal quale si ottiene il valore 000000010_2. Il bit attivo, di questo valore, viene usato per richiedere al contatore il caricamento dell'indirizzo che riceve in ingresso, proveniente dalla memoria ***m0***, la quale lo produce in base alla funzione richiesta all'unità di controllo.
2. Attivando l'ingresso ***Run***, appena si presenta una **variazione negativa** del valore di ***Clk*** (l'ingresso ***Clk*** è invertito per questo) il contatore carica l'indirizzo proveniente dalla memoria ***m0*** (0011_2) e lo riproduce nell'ingresso della memoria ***m1***, dalla quale si ottiene il valore 000110000_2, corrispondenti alla richiesta di trasferire il contenuto di ***A*** in ***B***.
3. Si presenta un'altra variazione negativa del valore di ***Clk*** e il contatore viene incrementato di un'unità, selezionando da ***m1*** l'indirizzo 0100_2, con il quale la memoria ***m1*** produce il valore 011000000_2, corrispondenti alla richiesta di trasferire il contenuto di ***B*** in ***C***.
4. Si presenta un'altra variazione negativa del valore di ***Clk*** e il contatore viene incrementato di un'unità, selezionando da ***m1*** l'indirizzo 0101_2, con il quale la memoria ***m1*** produce il valore 000000111_2, corrispondenti alla richiesta di: fornire in ingresso al contatore il valore zero, caricare il contatore (con il valore zero), bloccare l'ingresso ***Clk***.

A questo punto il ciclo di esecuzione di una funzione è terminato e, per eseguirne un'altra, dopo aver fornito il valore corrispondente alla nuova funzione occorre disattivare e riattivare l'ingresso ***Run***.

Figura u103.11. Collegamento dell'unità di controllo ai componenti del bus dati: le prime tre linee del bus di controllo sono utilizzate dall'unità stessa e non servono a pilotare i moduli del bus dati.



Ciò che si scrive nelle memorie ROM che compongono l'unità di controllo può essere prodotto con strumenti che ne consentono una rappresentazione simbolica. Tkgate dispone di un compilatore che produce i file-immagine del microcodice, della mappa che consente di raggiungere le istruzioni nel microcodice attraverso la specificazione dei codici operativi, ed eventualmente anche il macrocodice, a partire da un sorgente unico. Il listato successivo mostra un sorgente compatibile con l'esempio di questa sezione.

Listato u103.12. Microcodice e codice operativo scritto per Tkgate 2.

```
map       bank[1:0]  m0;
microcode bank[8:0]  m1;
//-------------------------------------------------------
field ctrl_start[0];    // parte dall'indirizzo 0
field ctrl_load[1];     // carica l'indirizzo nel contatore.
field stop[2];          // stop clock.
field a_br[3];          // A <-- bus
field a_bw[4];          // A --> bus
```

```
field b_br[5];          // B <-- bus
field b_bw[6];          // B --> bus
field c_br[7];          // C <-- bus
field c_bw[8];          // C --> bus
//-------------------------------------------------------
op f0 {
  map f0: 0;
  +0[7:0]=0;
};
op f1 {
  map f1: 1;
  +0[7:0]=1;
};
op f2 {
  map f2: 2;
  +0[7:0]=2;
};
op f3 {
  map f3: 3;
  +0[7:0]=3;
};
//-------------------------------------------------------
begin microcode @ 0
load:
    ctrl_load;                  // CNT <-- TBL[f]

f0:
    b_br a_bw;                  // B <-- A
    ctrl_start ctrl_load stop;  // CNT <-- 0

f1:
    b_br a_bw;                  // B <-- A
    c_br b_bw;                  // C <-- B
    ctrl_start ctrl_load stop;  // CNT <-- 0

f2:
    c_br b_bw;                  // C <-- B
    ctrl_start ctrl_load stop;  // CNT <-- 0

f3:
    c_br b_bw;                  // C <-- B
    a_br c_bw;                  // A <-- C
    ctrl_start ctrl_load stop;  // CNT <-- 0

end
```

Il contenuto della memoria che rappresenta la «mappa» è organizzato in «codici operativi». Per la precisione, il codice operativo è l'indirizzo della mappa in cui si trova, a sua volta, l'indirizzo della memoria contenente il codice operativo, da cui inizia l'esecuzione di una certa funzione. Per esempio, il codice operativo della funzione f_2 è 2, come si vede nel listato di Tkgate:

```
op f2 {
  map f2: 2;
  +0[7:0]=2;
};
```

Viene riproposta la figura che mette a confronto le due memorie, ***m0*** e ***m1***, con l'aggiunta di altri dettagli, a completamento dell'argomento.

Figura u103.14. Mappa e microcodice.