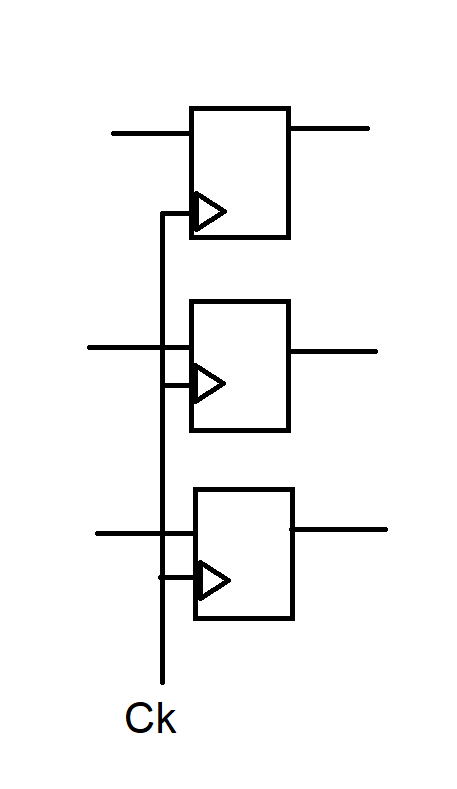
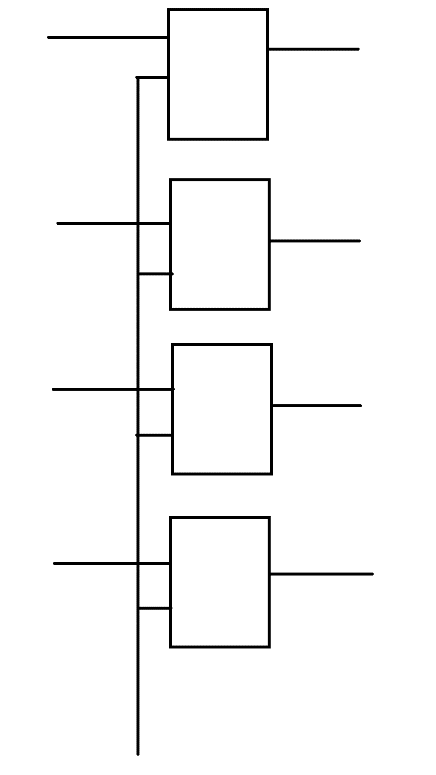
Abbiamo visto i flip flop di tipo set-reset, i flip flop d (normali), i flip flop d (master-slave), i flip flop di tipo T e i flip flop di tipo J-K (che è il flip flop generale che permette di realizzare qualunque tipo di dispositivo.

D sta per Data, T sta per Toggle (cioè commutare).

Una situazione in cui si possono usare i flip flop sono i registri di tipo d: questi sono una sequenza di flip-flop (di tipo d master slave). Essi permettono di memorizzare una sequenza di bit

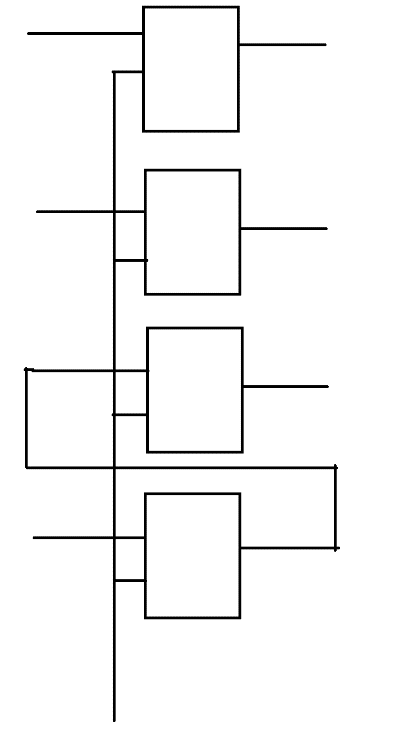
Questo meccanismo permette di memorizzare 3 bit. Esso è controllato da un unico clock ed è edge triggered dalle salite. Questo vuol dire che quando clock passa da 0 a 1 i valori in ingresso vengono memorizzati nelle uscite. I registri sono i componenti fondamentali del nostro processori, poiché permettono di conservare dati per potervi fare operazioni.

Passiamo ora all’esempio (leggermente più complicato) di un registro contatore. I registri contatori possono essere realizzati utilizzando dei flip-flop di tipo T.

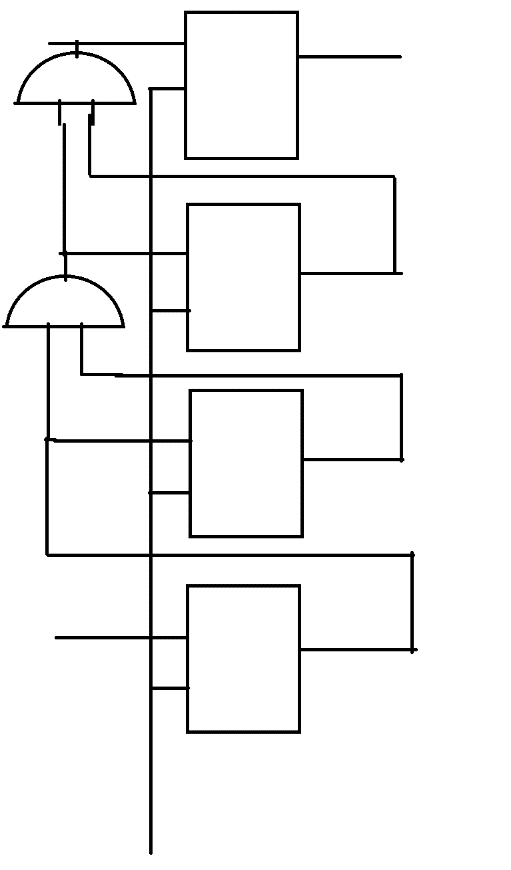


Immaginiamo di inizializzare tutti i bit in uscita a 0 e di volerlo far contare fino a 15. Cioè ogni volta che cambia il valore del clock il valore immagazzinato aumenti di 1.

Per la cifra meno significativa è facile (poiché il suo valore cambia a ogni clock). Per la seconda cifra basta invece inviare all’ingresso T l’output del flip-flop della prima cifra. In questo modo dopo il primo impulso di clock alla seconda cifra ci sarà in ingresso T, e al secondo ciclo questo avrà in uscita 1.

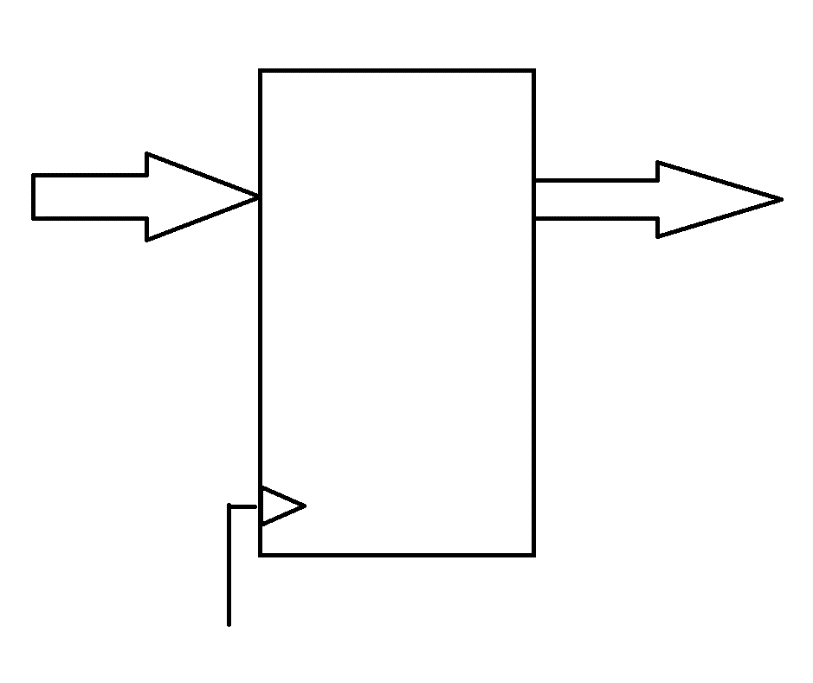


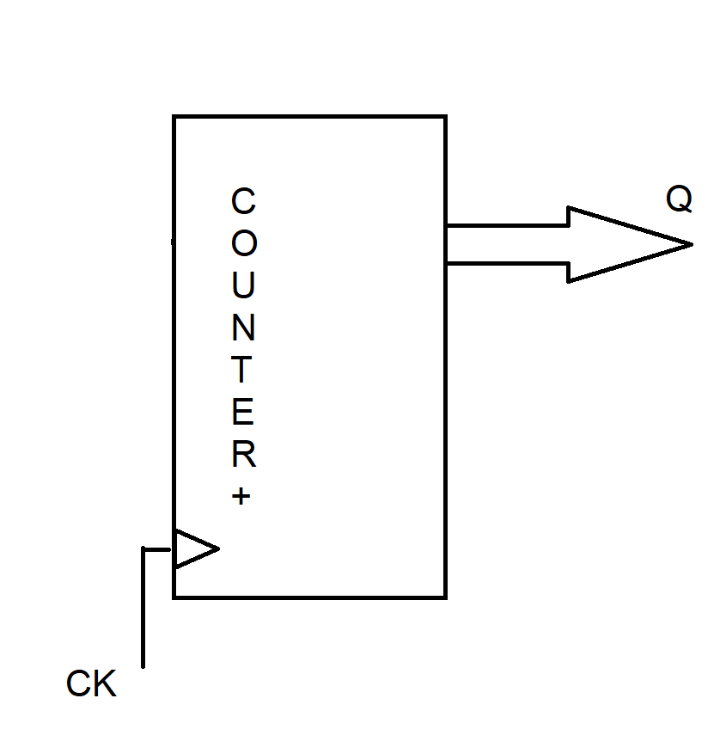
T2 deve essere valutato invece come un AND tra Q0 e Q1 e andando avanti, l’ingresso di T3 deve essere valutato come un AND tra T2 e Q2. Un eventuale T4 sarà = a un AND tra T3 e Q3 e così via… Questo permette la realizzazione di un registro a numero arbitrario di bit.



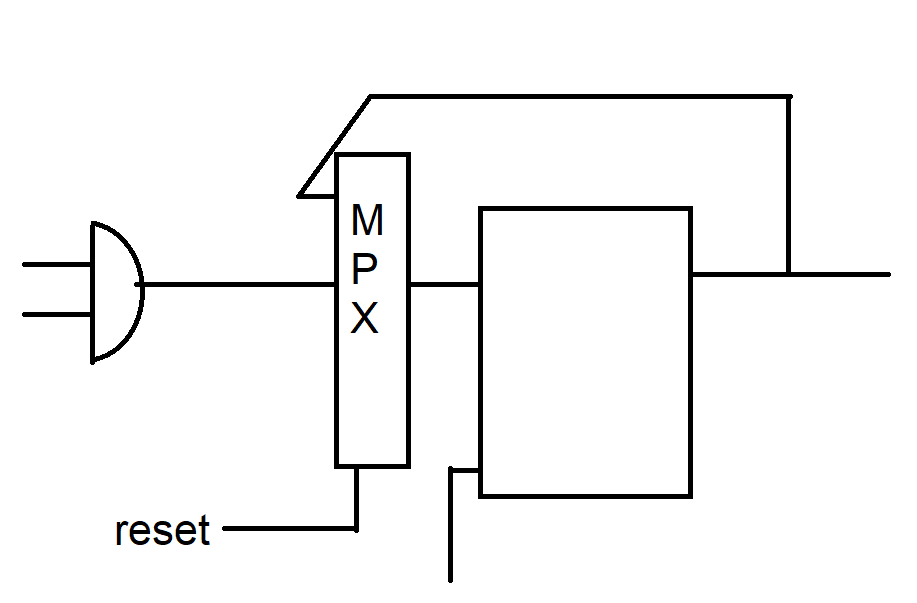
Volendo è possibile anche realizzare un registro che conta all’indietro. Ciò viene reso abbastanza facile è che oltre alle uscite Q questi flip flop hanno anche un’uscita -Q.

I registri possono essere anche visualizzati come dei moduli a loro volta utilizzabili. Se volessimo rappresentare in forma compatta un registro di tipo D potremmo farlo così:

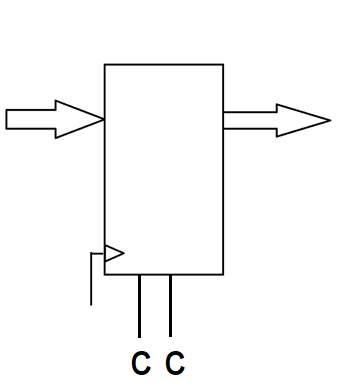




Potrei però aggiungere altri ingressi che mi permettano di avere un comportamento diverso. Potrei ad esempio aggiungere un ingresso reset che mi ri-azzeri il contatore (che se vale 1 quando arriva l’impulso di clock imposti tutti i Q a 0). Come si potrebbe fare? Beh tenendo conto che all’interno ci sono dei Flip-Flop di tipo T, per azzerarli il modo è prendere Q e ri-inviarla all’ingresso T: Se Q vale 0, riportandolo su T e inviando un impulso di clock l’uscita non cambia, mentre un 1 sull’ingresso T vuol dire “cambia” e quindi il valore passa da 1 a 0 al successivo impulso di clock. Per avere entrambe le funzionalità (contare in avanti e ri-azzerare) basta aggiungere un multiplexer.



Aggiungendo le variabili di controllo al multiplexer è possibile far funzionare il dispositivo in più modi differenti.

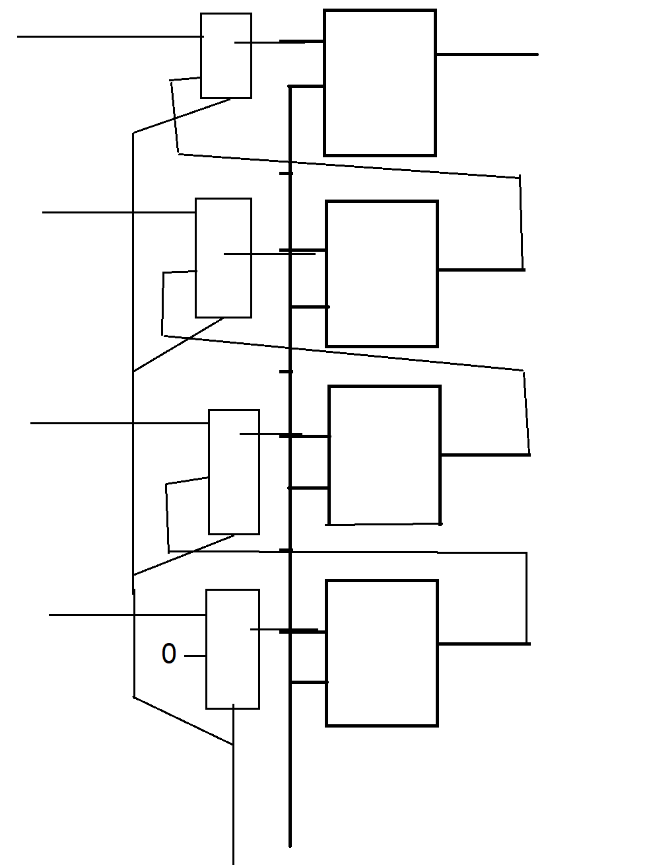


Le varie funzionalità potrebbero essere: conta in avanti, conta all’indietro, non fare niente e resetta (?)

Passiamo adesso al Registro a Scorrimento.

Prendiamo dei Flip-Flop di tipo D e mettiamo su tutti gli ingressi un multiplexer a 2 ingressi, con tutti i segnali di controllo collegati allo stesso C0.

Questo fa sì che tutti i bit vengano spostati di 1 (passano da q0 a Q1, da Q1 a Q2): ciò permette di trasportare l’informazione su un solo filo di comunicazione (poiché vengono inviati in ordine).



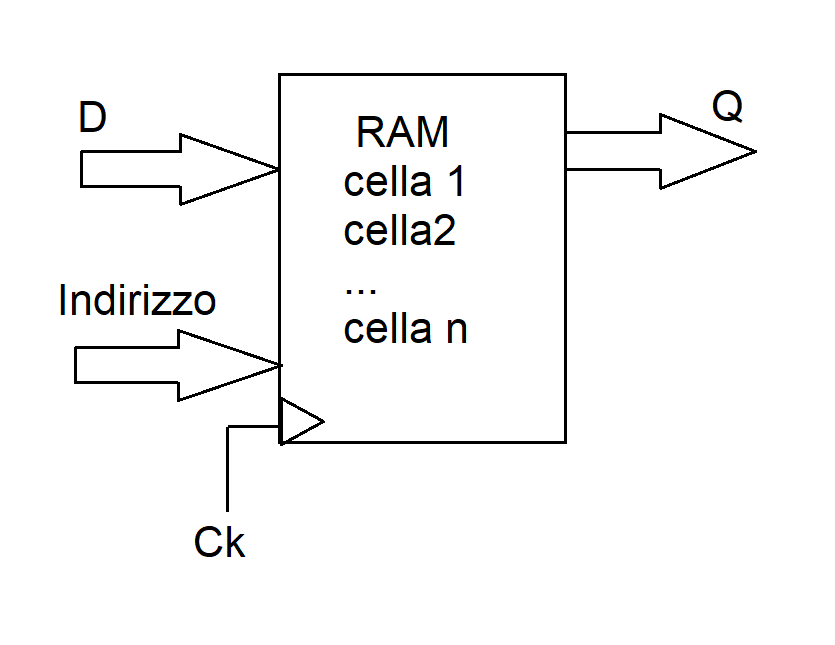
I registri a scorrimento erano in effetti un metodo di trasmissione a distanza di informazioni (questo perché tutti i bit erano inviati con frequenza uguale agli impulsi di clock).

Un altro utilizzo per i registri a scorrimento è quello di moltiplicare per 2 un valore ( se facciamo scorrere di una posizione abbiamo moltiplicato per 2, se scorriamo di 2 abbiamo moltiplicato per 8 ecc) quindi un registro a scorrimento può essere parte di un circuito moltiplicatore. Il vantaggio è quello di avere un solo dispositivo hardware che sia modulare, lo svantaggio è che al crescere della potenza di 2 per cui moltiplicare aumentano gli impulsi di clock necessari (notare che analogamente se facciamo scorrere all’indietro otteniamo una divisione per 2).

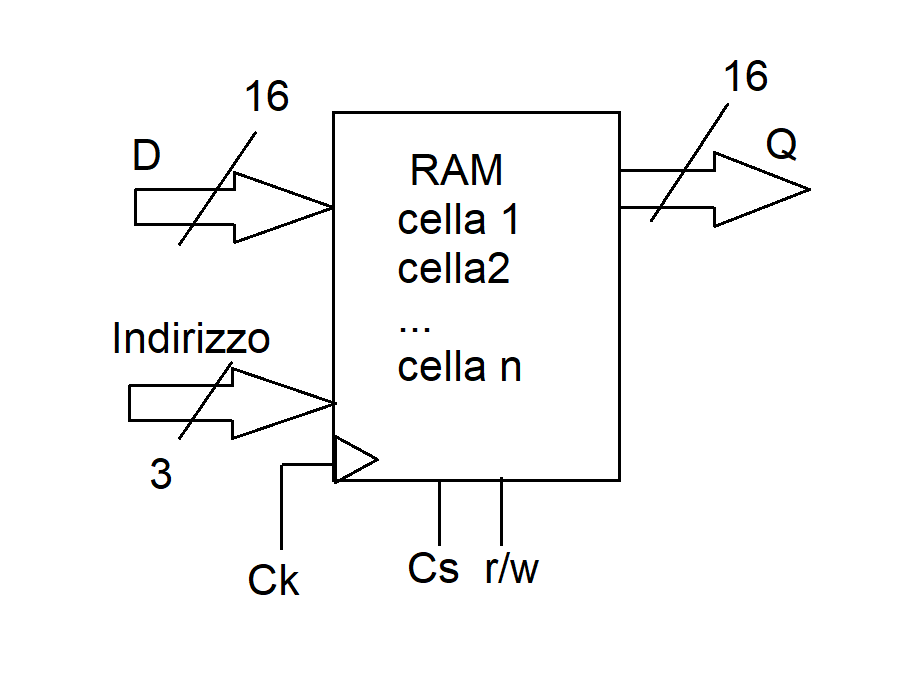
E adesso veniamo al succo. Ecco una struttura di UNA MEMORIA RAM STATICA.

Abbiamo detto che un singolo flip-flop è in grado di memorizzare un bit. Se al clock non arrivano altri impulsi rimane memorizzato il segnale inviato precedentemente. Il primo passo per costruire una memoria è quello di costruire un registro replicando il flip-flop. A questo punto il registro può memorizzare K bit (per esempio con 16 flip flop possiamo memorizzare un INT).

Il passo successivo è quello di usare tanti registri per memorizzare tanti numeri diversi. Questi registri li possiamo “numerare” e possiamo chiamare il numero assegnato al registro “INDIRIZZO”. Possiamo quindi provare a usare tale indirizzo per accedere in lettura o in scrittura a tali registri.

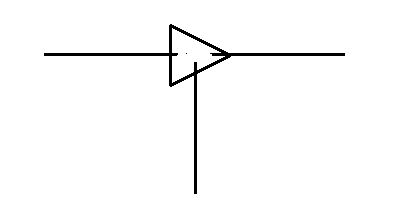


Per poter accedere in scrittura, dobbiamo innanzitutto inviare l’indirizzo della cella a cui vogliamo accedere ( se abbiamo ad esempio 8 celle di memoria l’indirizzo sarà codificato su 3 bit) e il controllo che indica la scrittura, dopodiché deve arrivare il nuovo valore che deve sovrascrivere quello precedente



Per quanto riguarda la lettura, si svolge analogamente ma anziché inviare un valore particolare, dopo che sono arrivati controllo e indirizzo da Q escono i (16?) bit che erano memorizzati nel registro scelto.

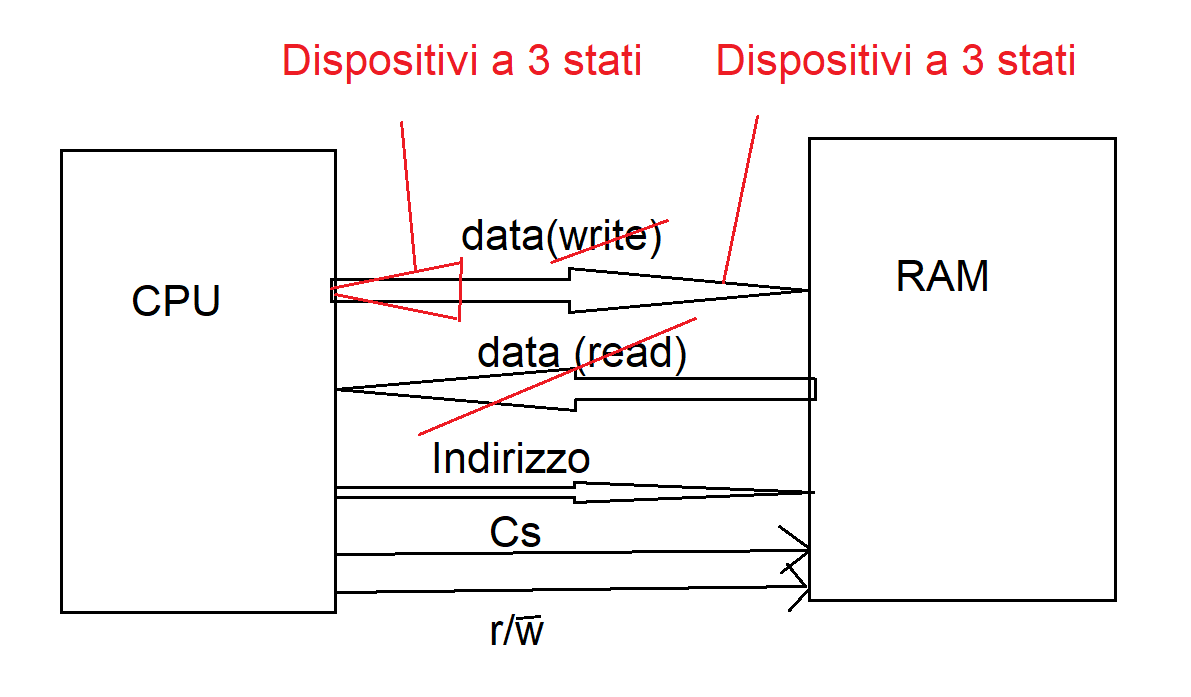
C’è poi un terzo controllo possibile, quello di Nessuna-Operazione (poiché ci sono 3 valori di controllo possibili servono 2 bit di controllo, uno chiamato Cs, che fintanto che è = 0 indica l’operazione di N-O, e l’altro è chiamato r/w che, quando Cs = 1 se vale 1 esegue la lettura, mentre se vale 0 esegue la scrittura -> Non si possono fare lettura e scrittura in una sola volta). È importante marcare che lettura e scrittura sono mutualmente esclusive perché ciò ci permette di risparmiare spazio: infatti una delle limitazioni sono il numero di fili in comunicazione con l’esterno. Se teniamo conto che scrittura e lettura non avvengono contemporaneamente si può economizzare usando solo K (in questo caso 16) fili che funzionino sia come uscite che come entrate. Ciò da un punto di vista realizzativo è abbastanza complicato, ma si è trovata una soluzione usando dei dispositivi chiamati “Dispositivi a tre stati”.



Hanno un ingresso, un uscita e un ingresso di controllo. Quando il valore di controllo è 0 è come se l’uscita fosse staccata (non si propaga il valore in ingresso, ma 0). Quando il valore di controllo è 1 si propaga il valore di ingresso in uscita. Esso, quindi, è come un dispositivo che rappresenta la funzione identità ( invia 1 solo se entrambi i tipi di ingresso valgono 1). Ciò permette di inviare più uscite su uno stesso filo, a patto che il segnale di controllo 1 arrivi sempre al massimo su un solo dispositivo.

Questo permette di collegare i fili in uscita da una ram con i fili in ingresso, a patto che da qualche parte due gruppi di dispositivi a 3 stati impediscano che entrambi i gruppi di fili siano “attivi” contemporaneamente. A fungere da variabile di controllo per i dispositivi a 3 stati sono le stesse variabili di controllo Cs e r/-w (r/w negato) (che passano attraverso un AND). Se Cs vale 1 e r/w vale 1 vengono abilitati i fili in uscita (-> lettura) e disabilitati quelli in entrata, Se Cs vale 1 e r/w vale 0 (e quindi not r/w vale 1) vengono abilitati i fili in entrata (-> scrittura) e disabilitati quelli in uscita.

In sostanza abbiamo un bit (Chip Select) che codifica se sul dispositivo vanne fatte operazioni o meno, un altro bit che decide se l’operazione da fare è in lettura o scrittura. Poi per risparmiare sul numero di fili (perché sono la parte che costa di più) si usando i Dispositivi a 3 stati. Infatti, ci sono già n fili in ingresso che codificano l’indirizzo (dove il numero di celle è 2^n) e per evitare di aggiungerne altri 2\*K (ingresso/uscita) se ne aggiungono solo K (la cui funzione è alternata, appunto, dai dispositivi a 3 stati -> infatti normalmente, la condivisione di più uscite può essere un problema, perché potrebbe bruciarsi il filo o danneggiarsi i dispositivi). Il collegamento è in quel senso bidirezionale.



La RAM funziona come dispositivo Slave e la CPU come dispositivo Master, essendo la CPU a inviare i segnali di controllo. C’è inoltre da dire che CPU e RAM condividono un filo in ingresso di clock.