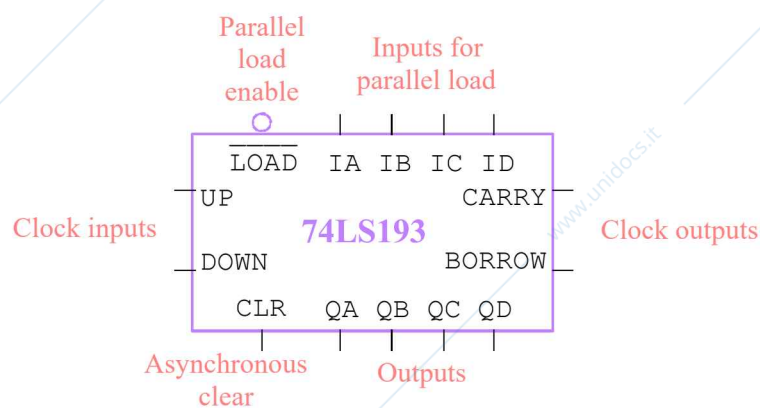


## Svolgimento

### 1 a)

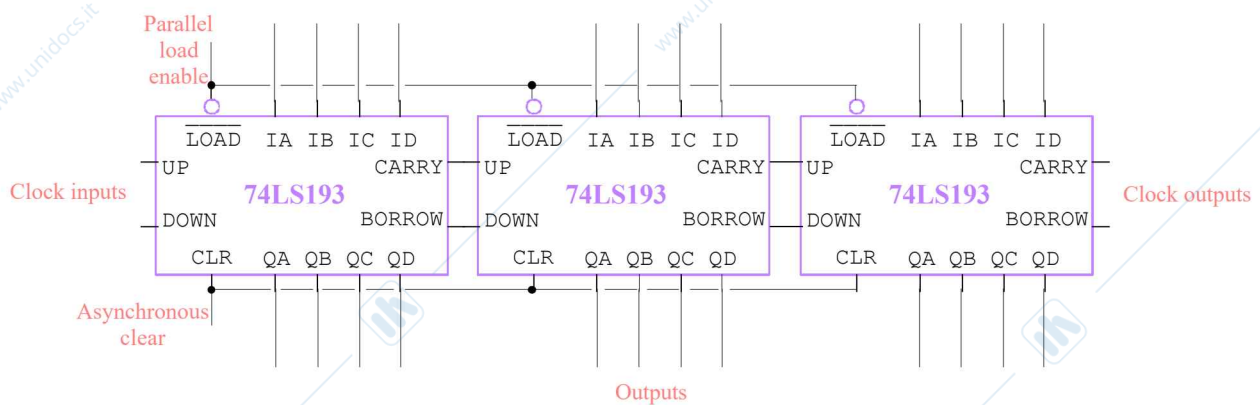
Il circuito realizza un generatore di otto forme d'onda scandibili singolarmente ad una frequenza di clock impostabile a piacere. I dati numerici sono contenuti in una memoria RAM da  $32k\text{byte}$ . Ogni forma d'onda occuperà un banco di  $32k/8=4k\text{byte}$  all'interno della RAM; quindi, dei  $\log_2(32 \times 1024)=15$  bit di indirizzo, 12 verranno utilizzati per scandire il banco contenente la forma d'onda, mentre gli altri 3 serviranno per scegliere uno degli otto banchi disponibili.

Per la scansione sia dei blocchi che dei singoli byte di dato, contenuti nel blocco in esame, si utilizzerà un contatore LS193 dotato di ingressi indipendenti per il conteggio in avanti / indietro (Up/Down). In questo modo sarà facile potere scandire le forme d'onda in entrambi i sensi, per indirizzi crescenti o decrescenti, in base alla necessità. La figura seguente mostra gli ingressi e le uscite di tale integrato.



Le uscite sono quattro ed è permesso il caricamento parallelo dei quattro flip/flop contenuti al suo interno, abilitando a livello *low* l'ingresso LOAD. Nonostante nell'applicazione in esame questa funzione non sarà utilizzata, è interessante sapere che è comunque a disposizione a costo zero. Un master clear asincrono permette di azzerare tutti i flip/flop e ripartire da 0000.

Il segnale di clock non giunge su un unico piedino, bensì ne sono presenti due: a seconda di quale dei due è attivato dal clock, il contatore provvederà ad effettuare il conteggio in avanti (0000@0001@0010@0011@...) o all'indietro (0000@1111@1110@1101@...). Sono presenti anche due uscite che rappresentano rispettivamente l'impulso di CARRY (quando il conteggio passa da 1111@0000) e di BORROW (passando da 0000@1111). Grazie a questi due piedini è molto facile porre in cascata più contatori, in modo da aumentare il numero delle uscite e, quindi, la capacità di conteggio. La figura seguente mostra la modalità di impiego in cascata.



Il contatore LS193 differisce dal fratello LS192 semplicemente per il tipo di conteggio: mentre il primo è un contatore binario (cioè conta da 0000@1111, da  $0_H$  a  $F_H$ ), il secondo è BCD (ossia conta da 0000@0101, da  $0_H$  a  $9_H$ ). Per l'applicazione in esame, naturalmente, è necessario usare contatori binari, altrimenti si perderebbero delle combinazioni e, in definitiva, non verrebbero lette delle locazioni di memoria.

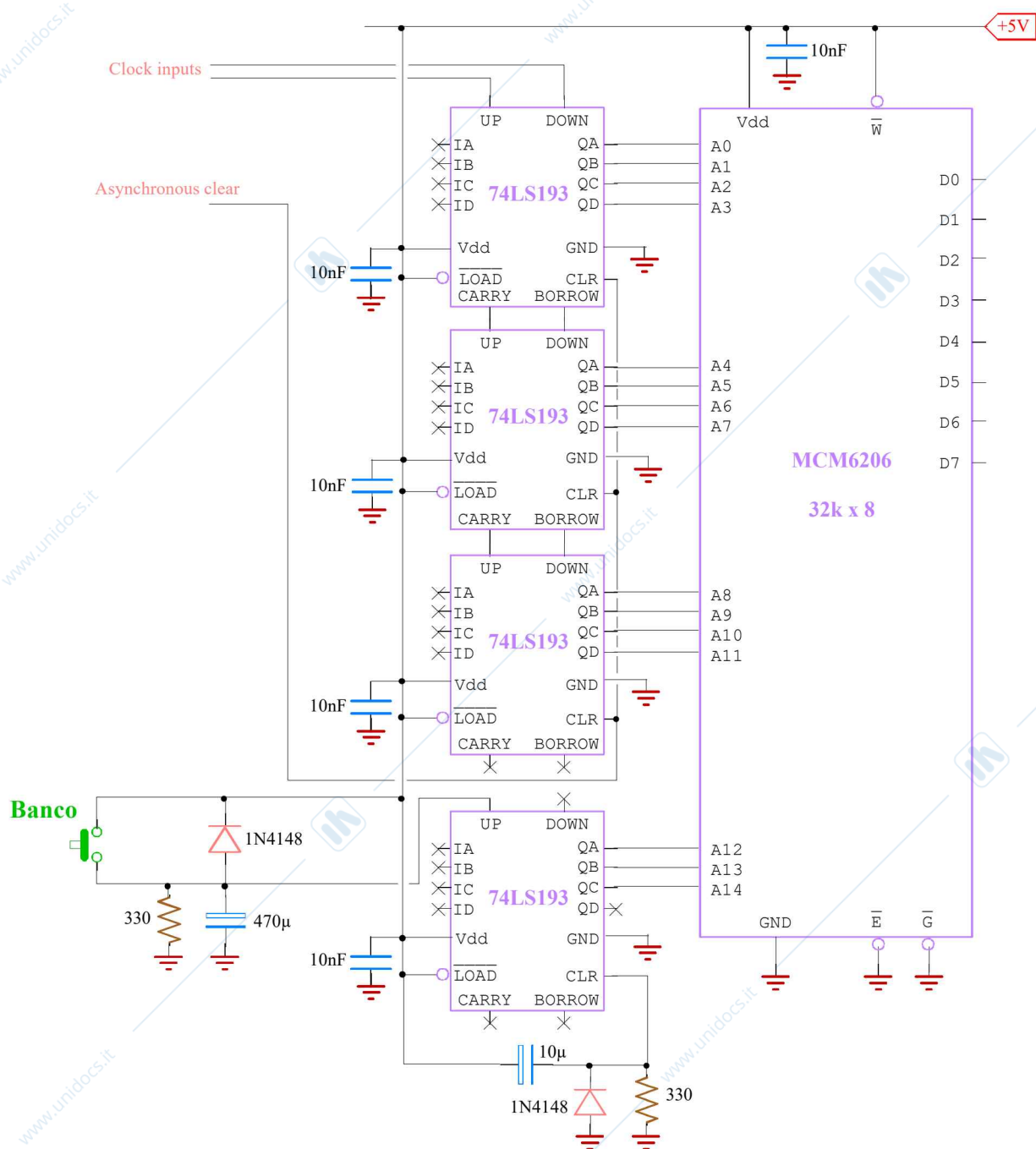
Per il blocco di *Scansione*, avendo bisogno di 12 bit per gli indirizzi, saranno sufficienti tre contatori connessi in cascata (indirizzi da  $000_H$  a  $FFF_H$ ). Mentre per il blocco di *Selezione* basterà un solo contatore, di cui si utilizzeranno solamente 3 delle quattro uscite. In tutti e due i casi non verranno utilizzati gli ingressi di caricamento parallelo e, quindi, il piedino LOAD dovrà essere sempre tenuto disabilitato. Volendo già prevedere un reset comune a tutti i contatori del blocco *Scansione*, tutti i piedini CLR saranno connessi insieme.

La figura della pagina seguente mostra il progetto circuitale dei due blocchi di *Scansione* e di *Selezione* della memoria. Naturalmente ora si sono collegati anche i piedini di alimentazione, Vdd e GND, di tutti i contatori ed anche della memoria, che erano stato finora trascurati. In particolare sono stati introdotti i condensatori di tank ai capi di ogni condensatore, in modo da ridurre gli spike iniettati sull'alimentazione a causa di rapidi fronti di assorbimento. Sono stati anche connessi i piedini di abilitazione della RAM: i vari chip enable, E e G, a livello attivo *low*, mentre il piedino di write, W, è disabilitato a livello *high*, poiché si vuole solo leggere dalla RAM.

La rete presente sul piedino di CLEAR del contatore di banco permette di dare un impulso *high* all'accensione del circuito, in modo da resettare il contatore e partire dal banco  $0_H$ ; nel giro di qualche decina di microsecondi il condensatore si carica e lascia libero il contatore. Il diodo serve per scaricare velocemente il condensatore all'atto dello spegnimento del circuito (quando viene tolta dell'alimentazione Vdd) per impedire che il piedino di CLR si porti a tensioni negative.

Per quanto riguarda il pulsante che comanda il contatore di selezione **Banco**, ma anche per tutti gli altri che si andranno ad introdurre nel circuito, è necessario prevedere gli opportuni accorgimenti per limitare il problema dei rimbalzi dei contatti, dei disturbi sull'alimentazione e, in generale, per limitare i consumi. A tale scopo si è aggiunto un semplice filtro RC di debouncing, con costante di tempo di qualche centinaio di millisecondi, con diodo per lo scaricamento rapido del condensatore.

È importante osservare che i contatori appartengono alla famiglia TTL e, quindi, hanno tutti i problemi tipici, soprattutto dal punto di vista delle correnti di ingresso che si devono considerare in fase di dimensionamento delle resistenze di pull down esterne. Anche se non si tratta della famiglia TTL standard 74xx, bensì di quella Low-power Schottky 74LSxx, le correnti di ingresso quotate sui data sheet sono ancora elevate; soprattutto per ciò che riguarda la corrente di ingresso a livello low nel caso peggiore può valere  $1.6mA$ . Quindi non è possibile utilizzare resistenze di pull-down di  $10k\Omega$ , bensì si deve ricorrere a valori molto più bassi, ad esempio  $330\Omega$  che determinano una tensione massima di  $0.5V$ , ancora accettabile come livello *low*. Tale scelta impone, purtroppo, un elevato consumo di corrente in fase di pressione del pulsante.

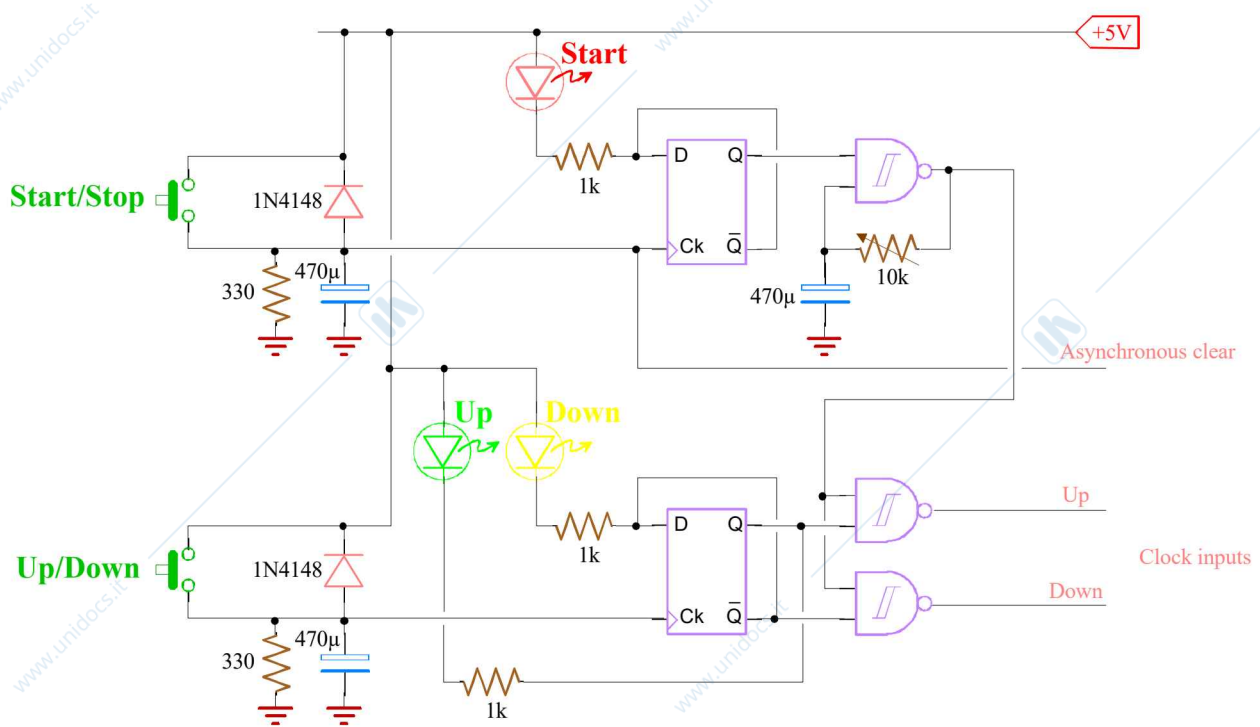


## 1 b)

Innanzitutto è necessario creare un semplice generatore di frequenza di clock. Un requisito interessante è la facile variabilità della frequenza, ad esempio tramite un semplice potenziometro. E' possibile utilizzare un semplice ring-oscillator con degli inverter ed una rete di carica RC: Il circuito minimo utilizza una unica porta che, per migliorare le prestazioni di scatto e la stabilità in frequenza, è necessario che sia di tipo con ingresso a Trigger di Schmidt.

Sia per il comando di **Start/Stop** che per quello di **Up/Down**, si vogliono utilizzare dei pulsanti. Si vuole che ad ogni pressione del pulsante la funzione associata ad esso cambi ciclicamente. E' conveniente quindi utilizzare due semplici flip/flop di tipo toggle. Il primo flip/flop, comandato dal pulsante di **Start/Stop**, dovrà abilitare o meno la generazione degli impulsi di clock: per risparmiare corrente, si decide di disabilitare l'oscillatore. Il secondo, comandato dal pulsante **Up/Down** devierà il segnale di clock verso i piedini Up o Down del primo contatore LS193.

Inoltre, si decide di fare in modo che alla pressione del tasto di Start la forma d'onda riparta dall'inizio, ossia dalla posizione di memoria 000<sub>H</sub>.



La figura soprastante riporta lo schema elettrico del blocco *Up/Down Clock*. Sono stati aggiunti anche tre LED per indicare il funzionamento (Start) e la direzione (Up e Down) di scansione del banco di memoria.

Nonostante non sia necessario usare due NAND a trigger di Schmidt per le uscite dei clock, si è preferito usare queste, poiché ne serve comunque una per la generazione del clock, ed in un integrato ne sono contenute quattro.

Prelevando il segnale di reset direttamente dal pulsante di Start/Stop si ha un effetto che, forse, potrebbe risultare indesiderato. I contatori si azzerano tutte le volte che viene premuto il pulsante, per avviare (Start) il conteggio, il quale può effettivamente iniziare solo a pulsante rilasciato. Purtroppo alla successiva pressione del pulsante (Stop) non solo viene fermata la scansione, ma si resettano i contatori. A causa di questo fatto non è possibile “congelare” la lettura all’istante in cui si era giunti. Come esercizio aggiuntivo il Lettore può provare a modificare semplicemente il circuito. E siccome l’appetito vien mangiando (assieme ad altre leggiadre sfiziosità di varia natura, NdA) si consiglia anche di fare in modo che i contatori si azzerino anche quando viene premuto il pulsante di Banco, in modo da iniziare sempre dall’indirizzo  $000_H$ .

## 1 c)

La massima frequenza di scansione dei blocchi è limitata da vari fattori: la velocità di accesso alla RAM (il suo tempo di lettura); il tempo necessario all’elettronica a valle della memoria (ad esempio un DAC) per elaborarne i dati; la massima frequenza di clock dei contatori; il generatore di clock.

Dai data sheet si ricava che i contatori LS193 possono raggiungere una frequenza massima di clock di  $33MHz$ , anche se è consigliato lavorare al massimo a  $25MHz$ . Per la RAM MCM6206, il tempo di accesso tra l’applicazione di un indirizzo ed il dato valido in uscita ( $t_{AVQV}$ ) è di  $30ns_{max}$ , corrispondente ad una frequenza di lavoro limite di  $33MHz$ . Tenendo conto che, però, un DAC a valle della RAM può richiedere un tempo di conversione anche di  $1\mu s$ , conviene limitare la frequenza di funzionamento.

Anche nel caso di frequenza elevata a  $25MHz$ , il tempo richiesto per scandire completamente la forma d’onda memorizzata in un banco è di  $1/25M \cdot 4096 \gg 164\mu s$ . Supponendo di avere memorizzato nel banco solamente un periodo della forma d’onda, la cadenza di generazione della forma d’onda sarà di  $6.1kHz$ . In realtà, poiché la frequenza di campionamento  $f_s$  è, appunto,  $25MHz$ , si potrebbero

campionare più periodi all'interno del banco. Il limite massimo è imposto dal teorema del campionamento: la massima frequenza ricostruibile dall'elettronica a valle è al massimo pari a  $f_s/2$ , ossia  $12.5MHz$ . Volendo limitarsi a solo  $1MHz$ , ossia un segnale di periodo  $1\mu s$ , è possibile memorizzare nel banco un numero di periodi pari a  $164\mu s/1\mu s=164$ .

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari

www.unidocs.it - Appunti e dispense per superare i tuoi esami universitari