

# Interfacciamento dei processori

## Seconda Parte: Perché proteggere le linee d'uscita dei computer

### PREMESSA

Nella prima parte di questa serie ci siamo occupati delle interfacce d'uscita per un dato a 8 bit, e abbiamo concluso che qualunque *dispositivo programmabile* dispone *intrinsecamente* di almeno **almeno una via** in grado di assicurare questo servizio (*porte d'uscita* in un *microcontrollore* o la *porta parallela* nel *personal computer*).

Se con esse si desidera pilotare una **periferica** va subito sottolineato che **si potrebbe fare a meno** dello strato di **interfaccia**, **essendo però consapevoli** che le uscite di una *logica di controllo* saranno inevitabilmente **sottoposte a stress**, soprattutto se chiamate ad **erogare** corrente!!

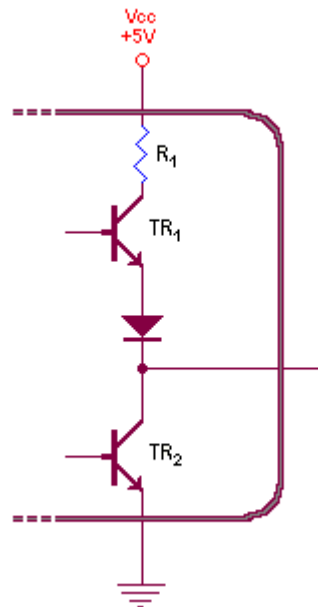
### CARATTERISTICHE DELLE LINEE D'USCITA DI UN COMPUTER

Sebbene questo argomento possa essere sviluppato in termini didattici più profondi sento di poter rendere l'idea anche senza spingere a fondo l'acceleratore...

La domanda (tra quelle *scontate ma non troppo*) è: Qual è la vera ragione per cui le *linee d'uscita* delle *porte* di un *computer* sono così *delicate*?

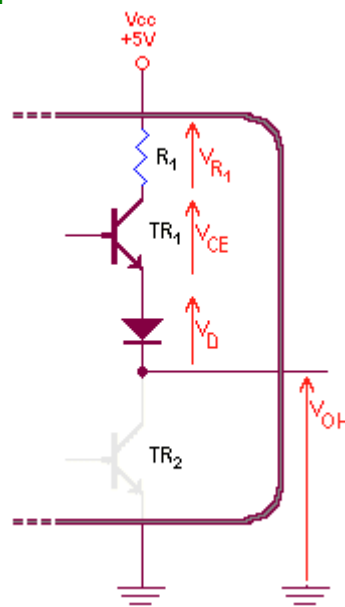
### Gli standard TTL

Con riferimento alla **logica TTL** (**T**ransistor **T**ransistor **L**ogic), utilizzata molto frequentemente per queste nostre *uscite*, i **limiti funzionali** delle rispettive caratteristiche sono a tutti noti; anche senza entrare nei dettagli, possiamo cercare di analizzarli, a partire dalla struttura **a totem** di ciascuna di esse.



Osservando la parte circuitale finale dei **dispositivi TTL** sono evidenti i due transistor *uno sopra l'altro*, come le figure di un *idolo pellerossa*, e per questo noti come **stadio d'uscita totem-pole**; questa configurazione è stata studiata per garantire una *bassa impedenza d'uscita* con entrambi i possibili livelli di tensione su essa presenti.

Quando la nostra *porta logica* (o una *linea d'uscita*) **TTL** è *a livello alto* (**1 logico**) il valore della sua *tensione d'uscita* ( $V_{OH}$ ) dipende dalla *corrente* che è costretta ad *erogare verso massa*, attraverso l'eventuale carico; in queste condizioni il transistor **TR<sub>2</sub>** è interdetto (cioè si comporta come un *circuito aperto*) mentre il transistor **TR<sub>1</sub>** è in normale conduzione.

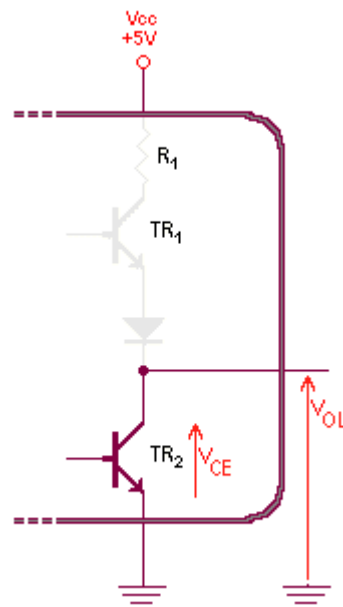


Nel funzionamento tipico (cioè quando l'*uscita TTL* è chiamata a pilotare gli *ingressi TTL* di oggetti simili a se stessa) è **obbligatorio** assicurare che la  $V_{OH}$  non scenda sotto i  $2,4V$ , il *valore limite* minimo standard perché l'uscita venga ritenuta un **1 logico**; questa condizione è rispettata fino ad un carico massimo di  $0,4mA$ , più che sufficiente per pilotare *una ventina* di *ingressi TTL<sub>LS</sub>* (condizione nota come *fan-out*).

Questo non significa che la corrente chiesta all'uscita non possa essere maggiore, ma **bisogna essere consapevoli** che un'*uscita TTL non è adatta* (e **non è nata**) per **erogare corrente**; si vede chiaramente che maggiore è la corrente richiesta dal carico minore sarà anche la  $V_{OH}$  disponibile, a causa della maggiore *caduta di tensione interna*, rispettivamente  $V_{R1}$  sul resistore  $R_1$  e  $V_{CE1}$  sul transistor  $TR_1$  (essendo pressoché inalterata quella  $V_D$  sul diodo, sempre intorno agli  $0,8V$ ).

Senza contare che la maggiore *caduta interna* tende inesorabilmente a **produrre calore**, stressando inutilmente il dispositivo di controllo.

Il discorso è completamente diverso se la *porta logica* (o *linea d'uscita*) **TTL** è *a livello basso* (**0 logico**); in queste condizioni il transistor  $TR_1$  è interdetto (cioè si comporta come un *circuito aperto*) mentre il transistor  $TR_2$  è in **forte conduzione** (o, come si suol dire, *in saturazione*).



In questa situazione il valore della sua *tensione d'uscita*  $V_{OL}$  è ora **comunque quasi nullo** (coincide con la tensione  $V_{CE}$  di saturazione di  $TR_2$ ,

mai superiore ai  $0,3V$ ) ed è sostanzialmente indipendentemente dalla *corrente* che è chiamata ad *assorbire* (per altro fino  $8mA$ , secondo le specifiche dello standard  $TTL_{LS}$ ): una **situazione ideale** per ogni tipo di applicazione!!

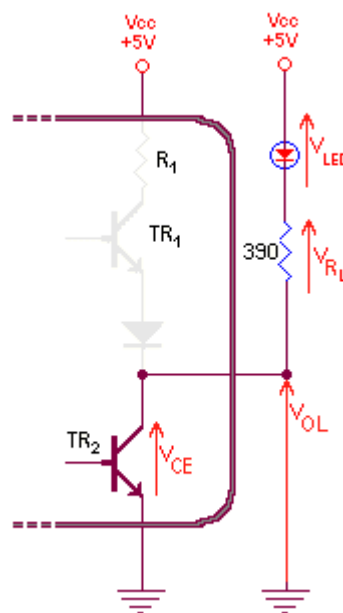
### Allora... Come ci si deve comportare?

Le considerazioni appena descritte vengono spesso *ignorate* (se non *disattese*...) dai *progettisti improvvisati*, soddisfatti più dall'esito finale positivo che dalla sostanza delle cose. Ma a me piace lasciare un *messaggio positivo* abituando i miei lettori a *subire il meno possibile*, cercando di suggerire un **metodo** che consenta loro di **capire** il perché delle cose e di **prendere posizione** di fronte alle possibili alternative.

Supponiamo di dover servire l'**accensione di un Led** collegato ad una *linea* di una *porta d'uscita*; ecco una buona occasione per chiarire i concetti legati a questo tipo di problema!!

Sebbene non esista una legge che ne caldeggi l'utilizzo, ritengo che il modo migliore per collegare una **periferica** ad una *linea d'uscita* sia quello (come amo dire..) "*in bevuta*", cioè quello di farla lavorare in *logica negativa*.

Per un **Led Rosso** la mia scelta è, dunque, certamente la seguente:



La *logica negativa* si qualifica nel fatto che *chi produce effetto* è il livello **0**: si vede *ad occhio* che il **Led si accenderà** se l'uscita della porta è a **0 logico**, situazione tipica di un transistor ( $TR_2$ ) *in saturazione*, che forza la *tensione d'uscita*  $V_{OL}$  verosimilmente a **0,1V** o **0,2V** (praticamente *una massa*); mentre non avrà sufficiente *differenza di potenziale* ai suoi capi se l'uscita della porta è **alta (1 logico)**, rimanendo **spento**.

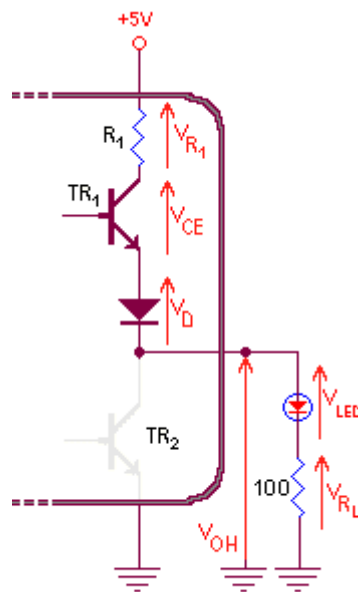
Il **vantaggio** di questa soluzione è quello di far lavorare la *porta di controllo* nel modo ad essa più naturale (essendo più adatta ad *assorbire corrente*) senza sottoporla ad alcun surriscaldamento; l'unico componente da scegliere è il **resistore**: il valore della sua *resistenza* dovrà assicurare una corrente nei limiti dello standard, cioè al massimo di **8mA**.

Il conto è presto fatto:  $[(V_{CC} - V_{LED} - V_{CE})/I_{LED}] = [(5V - 1.8V - 0.2V)/8mA] = 3V/8mA = 375 \text{ ohm}$ , portato al valore normalizzato di **390 ohm**.

Per completezza: la *caduta di tensione* ai capi di un **diode LED** dipende dalla sua corrente di lavoro ma anche dal suo *colore*; escludendo i *Led Flash* di ultima generazione (*bianchi* o *blu*, che assorbono fino a **40mA** con cadute tipiche di **3V**) nessuno dei rimanenti tipi potrà accendersi con tensioni applicate minori di **1,6V** (la cosiddetta *tensione di soglia*, molto maggiore di quella di un **diode normale al silicio**, pari a **0,6V**) mentre la tensione di lavoro, con una corrente tipica di **10mA**, può arrivare a **1,8V** per il **Rosso**, **1,9V** per il **Giallo**, **2,0V** per il **Verde** e per l'**Arancio**.

Un ulteriore **vantaggio** sta nella possibilità di impiegare una *tensione di alimentazione esterna*  $V_{DD}$  diversa da  $V_{CC}=5V$ ; in questo caso è fondamentale collegare *insieme* le masse dei 2 alimentatori e ricalcolare il valore della *resistenza* del **resistore** (per esempio con  $V_{DD}=12V$  esso passerà a **1250 ohm**, normalizzato a **1,2 Kohm**).

Molto frequentemente si da, invece, per scontata la soluzione alternativa:



**Di sicuro** la **corrente** che la **porta d'uscita** è costretta ad **erogare** è di gran lunga maggiore a quella per la quale è stata progettata (**0,4mA**), anche se questo, nei limiti della decenza, **non le nuocerà** più di tanto, se non per il fatto di contribuire a **riscaldare l'ambiente** per effetto dell'aumentata sua **caduta interna**. Volendo mantenere le ipotesi precedenti (cioè **accendere un Led Rosso**) risulta difficile anche il progetto del valore della **resistenza del resistore**: il valore della **caduta di tensione interna**,  $V_{R1} + V_{CE1} + V_D$ , è valutato (in condizioni standard di erogazione massima pari a **0,4mA**) in **2,6V** (dovendone lasciare in uscita  $V_{OH} = 2,4V$ , a partire dai **5V**) e sarà di certo maggiore (anche se difficilmente prevedibile) se la serie **R<sub>1</sub>, TR<sub>1</sub>, diodo**, sarà percorsa dagli **8mA** desiderati per accendere il **Led** in condizioni analoghe a quelle di prima..

**In conclusione** il **resistore** potrebbe anche non essere necessario, dato che la **tensione residua in uscita** si avvicinerà a quella del **diode LED Rosso** in zona di funzionamento (circa **1,8V**) ma, per buona etica, non faremo mancare un minimo di sicurezza, fissando la sua **resistenza** a **100 ohm**, probabilmente anche troppo elevata.

Nella sua **illogicità** la scelta di costringere la porta ad erogare corrente ha un suo **vantaggio**: non richiede alimentazione esterna e, probabilmente, è questa la ragione della sua diffusione; se non è possibile accedere alla **V<sub>CC</sub>**

della *porta di controllo* (come succede per esempio con la *porta parallela* di un PC) non rimane che tentare questa via, *illogica*, come detto, ma tutto sommato non *impraticabile*, se perseguita con consapevolezza.

**Comunque**, nel caso di una **periferica progettata seriamente**, il problema non si pone: sarà essa stessa a fornire lo strato di bufferizzazione (spesso con un semplice transistor) tale da non **caricare** la logica *a monte*.

### Electronica e ... filosofia.

Vale la pena ricordare una delle *mie* regole fondamentali: il *progetto elettronico* **non è una scienza esatta**!!

La conoscenza della *teoria* e del *funzionamento* dei componenti (come resistori, condensatori, diodi, transistor, operazionali, ecc..) è **sempre cosa buona e giusta**, ma è altrettanto certo che il progetto troppo preciso è intrinsecamente incongruente!

Basti pensare alla **tolleranza** delle caratteristiche di un *componente passivo* (che va dal **5%** tipico per la *resistenza* di un **resistore**, al **10%**, **20%** tipico per la *capacità* di un **buon condensatore**, al **50%** (e più) per la *capacità* di un **condensatore elettrolitico**; per non parlare delle imprecisioni dei *componenti attivi*..

Per un **transistor** la cosiddetta **dispersione sui parametri** è un dato di fatto; è molto educativo scorrere (almeno una volta) il *data sheet* di uno di questi oggetti. Il costruttore stesso si limita a fornire solo i valori massimi delle *correnti* e delle *cadute di tensione* relative ad un prefissato punto di lavoro; e anche per il più importante parametro di progetto (il *guadagno di corrente in continua*,  $h_{FE}$ ) viene fornito sempre un *valore massimo* e un *valore minimo*, di solito per una determinata corrente d'uscita). Dunque, anche **transistor con la stessa sigla** presentano caratteristiche notevolmente diverse tra loro.

Inutile dunque perdere tempo in *conti esatti alla terza cifra*: conviene invece imparare a progettare associando ad ogni parametro un valore *approssimato ma verosimile*; ma di questo parleremo la prossima puntata... Intanto puoi guardare e stamparti questo *data sheet*.

<http://www.ortodoxism.ro/datasheets/SGSThompsonMicroelectronics/mXyzzwx.pdf>

## COME PROTEGGERE LE LINEE D'USCITA DI UN

E' ora di riprendere il filo del discorso; ora conosciamo la fragile natura delle nostre **porte di controllo** e possiamo pensare di collegare loro le **periferiche** che ci servono, valutando con cognizione di causa **se** e **come** dotarle di **interfaccia**.

### Proteggere le linee d'uscita del computer

Abbiamo capito che, talvolta, il compito dell'**interfaccia** è quello di **organizzare le informazioni**, in modo **intelligente** e **specializzato**, al fine di assicurare la **disponibilità** di un **protocollo**, cioè un **insieme di regole** da condividere tra computer e periferica nello scambio dei dati e dei comandi; a questo tipo appartengono l'**interfaccia centronics** (tipica del controllo delle **stampanti parallele** ma **riciclabile** anche per altri dispositivi **paralleli**) e l'**interfaccia RS232** (tipica del controllo di dispositivi **seriali**).

Molto spesso, però, la presenza di uno **strato** di **interfacciamento** è semplicemente legata alla necessità di garantire alle eventuali **periferiche** **poste a valle** i livelli di **tensione** e **corrente** adatti al loro controllo, per evitare che esse **carichino** la sorgente del **segnale di controllo**, di norma disponibile su uno dei **pin** delle medesime porte, **poste a monte**.

Anche il concetto di **carico** (come quello di **buffer**, discusso la volta scorsa) è frutto di un modo **gergale** di trattare le **cose elettroniche** e può essere correttamente utilizzato per definire almeno un paio di **punti di vista**, per altro **equivalenti** tra loro; per **fare la mentalità** è utile sapere che, per **carico**, si può intendere:

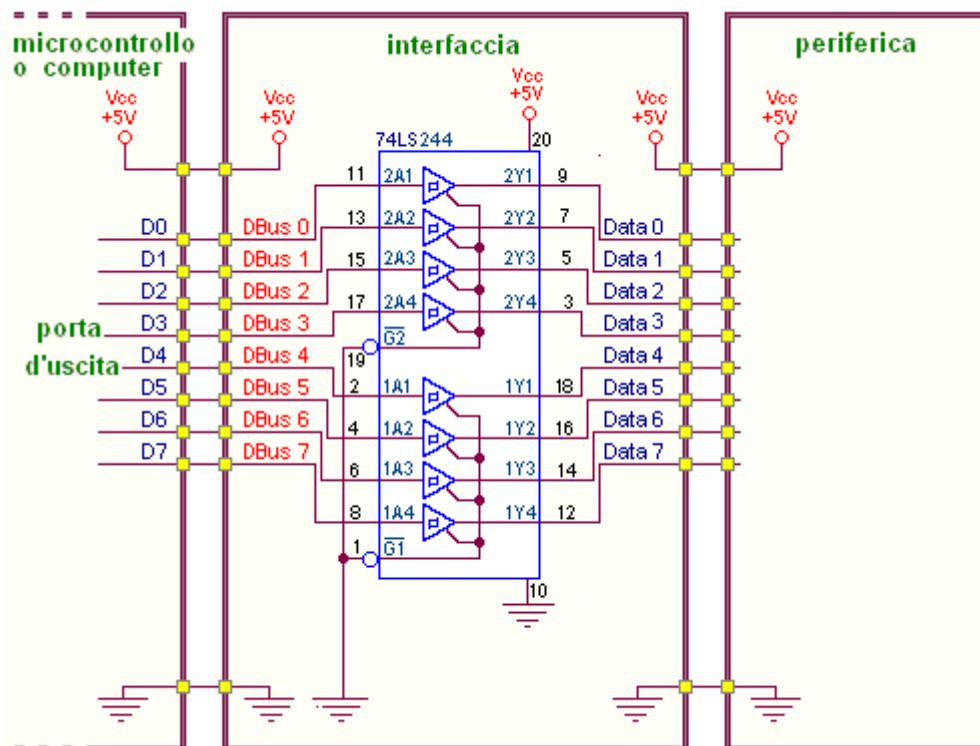
- un **oggetto che assorbe corrente** da una **linea d'uscita** sulla quale è applicata una **differenza di potenziale** rispetto a massa
- ma anche il **valore della corrente assorbita** dalla linea

E' utile tener presente un'altra delle **mie** regole fondamentali: "**non esiste corrente senza tensione** ma **può esistere tensione senza corrente**": sembra un'ovvietà ma è un sacrosanto principio al quale si pensa raramente.. Una batteria di pile mantiene per lungo tempo inalterata la sua **tensione nominale** se rimane inutilizzata sul tavolo (quindi **senza erogare corrente**), ma comincerà a scaricarsi tanto più velocemente quanto maggiore sarà il **carico** (cioè la **corrente**) ad essa richiesta.



In conclusione, **caricare** una linea significa dunque, indifferentemente, **applicarle un oggetto** (di solito un **resistore** di resistenza diversa da infinito) o **prelevarle una data corrente**. Una cosa è sicura: il **carico** di quella linea deve **rimanere nei limiti** previsti dalla **logica** che la governa, altrimenti si rischia di **vanificarne l'effetto** o, nella peggiore delle evenienze, di **bruciarla**, con danno spesso irreversibile su tutto il sistema di controllo.

Da questo punto di vista sia le **porte a 8 bit**, presenti (di norma) in ogni **microcontrollore**, che la **porta parallela** di un **personal computer** sono **decisamente fragili** ed è sempre saggio tutelarne l'incolumità almeno con la presenza, **in cascata**, di un **buffer di corrente** come il **74LS244** (con uscite **non invertenti**) o il **74LS240** (con uscite **invertenti**) oppure con l'aiuto di economici **transistori**, molto adatti allo scopo essendo **intrinsecamente** amplificatori di corrente.



Lo schema mette in evidenza la struttura di questo componente, contenente **2 gruppi** di **4 amplificatori di corrente non invertenti**, ciascuno controllato separatamente da un suo piedino di abilitazione, **G1** e **G2**, entrambi **attivi bassi**:

- quando il segnale ad essi applicato è attivo (cioè a massa, **0**) il livello logico presente in ingresso a ciascun amplificatore (o **1** o **0**) è copiato sulla sua uscita
- quando il segnale ad essi applicato non è attivo (cioè a **1**, oppure con abilitazione *scollegata*) tutte le 4 uscite del rispettivo gruppo di amplificatori passano in **alta impedenza (Hi-Z)**, il *terzo stato* del quale abbiamo già parlato nella puntata precedente, che esprime la condizione che, di fatto, **scollega praticamente** la periferica a valle, **anche se fisicamente** la cosa **non si vede!**

E' importante capire **cosa significa** mettere un'uscita in **alta impedenza**; supponiamo di applicare una **tensione** ad un **resistore**:

- la corrente che lo attraversa è soggetta alla Legge di Ohm per cui, a parità di tensione applicata, più grande è la sua *resistenza* e minore è la corrente che passa.
- se la sua *resistenza* elettrica è *infinita* la corrente è inevitabilmente *nulla*, cioè un **resistore** con questa caratteristica si comporta come un **circuito aperto**, come se esso stesso fosse *sparito*, lasciando *il vuoto* tra i punti ai quali era collegato..

Tecnicamente si dice che le uscite di questo componente hanno **3 stati logici**: i consueti **1** e **0** e il *terzo stato*, detto **Hi-Z**, appunto **alta impedenza**, condizione che, di fatto, **scollega praticamente** l'oggetto ad esse collegato, **anche se fisicamente** la cosa **non si vede!**

**In conclusione:** lo scopo dell'eventuale impiego di questo componente è quello garantire sulle sue uscite il **medesimo livello logico** applicato ai suoi ingressi dalla **logica di controllo**, ma con una **corrente maggiore** rispetto a quella che la **logica di controllo** stessa è in grado di fornire; la possibilità di fruire del *terzo stato* (**Hi-Z**) è, in questo caso, controproducente per cui entrambi i suoi piedini di abilitazione, **G1** e **G2**, saranno **attivati**, cioè fissati a **massa**.

Dalla tabella (che raccoglie le principali **caratteristiche elettriche** del componente) si può notare la disponibilità (eccezionale in ambito **TTL**) di **15 mA erogati** (con uscita a "1") e di **24 mA assorbiti** (con uscita a "0")

| Caratteristiche Elettriche  | Valori   |
|---|--|
| potenza dissipata   | <b>135 mW</b> (massima con uscita a "1", assorbe <b>27 mA</b> a <b>5V</b> )        |
|   | <b>230 mW</b> (massima con uscita a "0", assorbe <b>46 mA</b> a <b>5V</b> )        |
|   | <b>270 mW</b> (massima con uscita <b>Hi-Z</b> , assorbe <b>54 mA</b> a <b>5V</b> ) |
| corrente erogata <b>tipica</b> in uscita                          | <b>15 mA</b> (tipica buffer con uscita a "1")                                      |
| corrente assorbita <b>tipica</b> in uscita                        | <b>24 mA</b> (tipica buffer con uscita a "0")                                      |
| tempo di propagazione <b>massimo</b> da ingresso a uscita         | <b>18 ns</b> (fronte di salita) e <b>18 ns</b> (fronte di discesa)                 |
| tempo di abilitazione <b>massimo</b> da <b>enable</b> a uscita    | <b>23 ns</b> (verso uscita a "1") e <b>30 ns</b> (verso uscita a "0", 5pF)         |
| tempo di disabilitazione <b>massimo</b> da <b>enable</b> a uscita | <b>25 ns</b> (da uscita a "1") e <b>18 ns</b> (da uscita a "0", 5pF)               |
| [ i tempi sono stati rilevati con carico di 667ohm/45 pF ]        |  |

Si tratta di una tipica **interfaccia di bufferizzazione**, in grado di assicurare alla **periferica controllata** (collegata *a valle*, per esempio una *batteria di led*) la corrente ad essa necessaria (a parità di tensione) senza **caricare troppo** (cioè *assorbire troppa corrente*) la **logica controllante** (la porta del computer *a monte*).

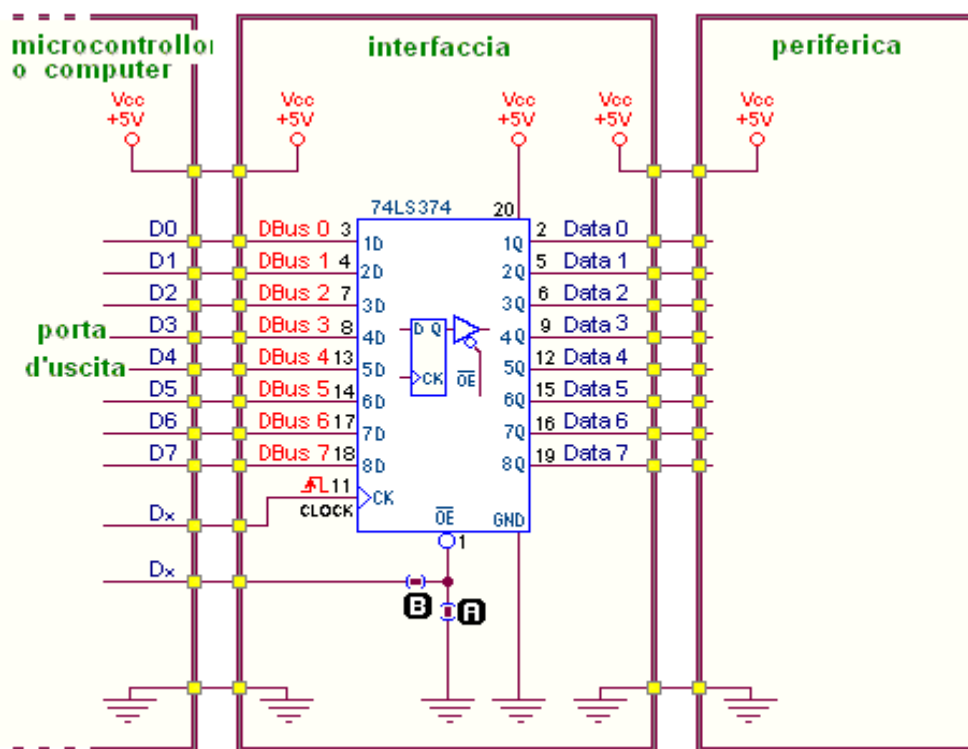
**Nota:** a questo punto della conversazione mi piace aggiungere una di quelle considerazioni che rimangono latenti, **scontate ma non troppo**: Come fa un **buffer** (come il **74LS244**, ma anche un semplice **transistor**) ad **amplificare** la corrente?

Non è una magia né una sua particolare abilità; esso si limita ad **amministrare la corrente altrui**: sollecitato da una **corrente decisamente piccola** è in grado di trasferire sul **carico** una corrente **centinaia di volte più grande**, semplicemente prendendola dov'è: dall'alimentatore!

## Proteggere le linee d'uscita con un "valore aggiunto"

La volta scorsa abbiamo presa in considerazione anche l'opportunità di delegare la *funzione di interfaccia* a particolari **memorie esterne a 8 bit**, molto utili anche per il fatto di offrire *contemporaneamente uscite bufferizzate* con caratteristiche identiche a quelle assicurate dai citati **74LS244**, appena analizzati. Ricordiamo che si tratta di *batterie* di otto **flip-flop in parallelo**: i **74LS374** (8 flip-flop **D-Type**) e i **74LS373** (8 flip-flop **D-Latch**).

Questa soluzione è proposta qui di seguito e costituirà il *circuito base* per interfacce con numero di uscite grande a piacere, comunque multiplo di 8, oggetto delle prossime puntate.



Lo schema mette in evidenza due cose importanti:

- poiché si tratta di una *memoria esterna aggiunta* (non strettamente necessaria, per la verità, essendo una memoria *anche* la porta che la controlla) deve disporre di una *linea di sincronismo (Clock)* necessaria alla memorizzazione del dato, da fornire sul suo **pin 11** e da realizzare con l'ausilio della *linea di uscita* di una *ulteriore porta* della *logica di controllo*
- la presenza (nascosta) di **amplificatori di corrente** su ciascuna delle 8 uscite è evidenziata dal piedino di abilitazione **OE**, *attivo basso*

A questo proposito lo schema propone **due soluzioni** (mi raccomando !!) **alternative**:

- la cosa più ovvia (e più comune..) è collegare l'**Out Enable** *direttamente a massa* (saldando **A**): in questo modo alla **periferica** si assicura il **medesimo livello logico** presente (internamente) sulle uscite dei *flip-flop D-Type*, ma con una **corrente maggiore** rispetto a quella che esse sarebbero in grado di fornire, tipica della *logica TTL*
- **oppure** (saldando **B**) è possibile lasciare la possibilità di porre le 8 uscite in *alta impedenza*; naturalmente si tratta di una soluzione sofisticata (del tutto particolare, non strettamente necessaria e controproducente nell'uso normale) che tra l'altro richiede una *linea di controllo in più*, disponibile solo richiedendo al computer una *porta ulteriore*, sulla quale intervenire *da software*

Nella prossima puntata entreremo nel merito del progetto di alcune **periferiche d'uscita**, tra le più utili, decidendo il modo migliore per **interfacciarle** alla *logica di controllo*.

---

Questo Tutorial è **del tutto originale**, creato e pensato per gli amici di **Grix** e articolato in numerose *puntate*...

Ogni suo **testo, immagine, schema, progetto** è protetto dalla normativa sul **diritto d'autore** [[http://www.siae.it/Faq\\_siae.asp](http://www.siae.it/Faq_siae.asp)]; la **riproduzione a fini commerciali**, totale o parziale, è quindi **vietata** in qualunque forma, su qualsiasi supporto e con qualunque mezzo.

L'autore è invece orgoglioso di offrire gratuitamente il suo lavoro e la sua esperienza a chiunque desidera arricchire le proprie conoscenze, autorizzando la **stampa** di quest'opera per un uso **personale e non commerciale** e l'eventuale utilizzo dei contenuti in ambiti esclusivamente amatoriali o didattici con la *speranza* che ne venga almeno citata la fonte.

Puoi **scaricare qui la versione PDF** della **SECONDA PARTE**

Alcuni argomenti **citati** in questa parte sono **disponibili** nella *monografia "Data Sheet"* della sezione "**Dentro il Computer**" del sito



una ricca raccolta di dispositivi e progetti pensata per aiutarti a comprendere i segreti del tuo computer [sia *Personal* che *microcontrollore*]

In particolare sono disponibili le descrizioni dettagliate dei *buffer tre-state* **74LS240** e **74LS244**

[http://www.giobe2000.it/HW/DataSheet/pag/74240\\_1.htm](http://www.giobe2000.it/HW/DataSheet/pag/74240_1.htm)

[http://www.giobe2000.it/HW/DataSheet/pag/74244\\_1.htm](http://www.giobe2000.it/HW/DataSheet/pag/74244_1.htm)

per i quali sono disponibili anche i relativi **Manuali stampabili Gratuiti** (4 pagine, 208 kBytes)

<http://www.giobe2000.it/Manuali/DataSheet.htm>