

# IL BUS I<sup>2</sup>C

In molti progetti d'elettronica, anche riferiti ad ambiti tra loro diversi (industriale, telecomunicazioni, etc.), spesso è rintracciabile una situazione ricorrente in cui un dispositivo intelligente, il più delle volte un microcontrollore, si trova a doversi interfacciare con svariate periferiche quali memorie, convertitori A/D, real time clock, sensori, etc. Proprio per la frequenza di questa situazione progettuale, nei primi anni ottanta la Philips sviluppò l'I<sup>2</sup>C (I<sup>2</sup>C o IIC sono l'acronimo di Inter-Integrated Circuit), un bus di comunicazione di tipo seriale che semplificasse, in modo flessibile ed economico, l'interfacciamento tra varie tipologie di integrati. Infatti, con questo bus non è necessario progettare nessun'interfaccia, poiché i dispositivi I<sup>2</sup>C compatibili, di qualunque natura essi siano, hanno sul loro stesso chip tutta la circuiteria necessaria per essere direttamente inseriti sulla linea di comunicazione, inoltre per la trasmissione e la ricezione sono necessarie due sole linee. Il numero di dispositivi collegabili è molto elevato e virtualmente dipende solo dalla capacità complessiva del bus. In commercio sono numerosi gli integrati che presentano questo tipo d'interfaccia: solo la Philips, già nel 1995, ne dichiarava più di 150.

Prima di addentrarci nella descrizione dettagliata del bus identifichiamo le tipologie di interlocutori che vi possono essere collegati.

Il bus I<sup>2</sup>C è bidirezionale, pertanto vi saranno presenti dispositivi in grado di trasmettere, di ricevere o di fare entrambe le cose (non contemporaneamente) secondo le loro esigenze di funzionamento.

È definito **trasmettitore** un dispositivo che invia dati sul bus; di rimando è definito **ricevitore** un dispositivo che riceve dati dal bus.

Oltre che in base alla direzione in cui transitano i dati, i dispositivi possono essere classificati in base alla loro capacità di richiedere o meno una comunicazione con un altro dispositivo.

In particolare si definisce **master** un dispositivo in grado di richiedere una comunicazione, generare il necessario segnale di clock e chiudere la comunicazione; si definisce, invece, **slave** il dispositivo indirizzato da un master.

Ad esempio il sensore di temperatura LM75 è uno slave in quanto non può richiedere nessuna comunicazione, ma deve essere interpellato da un altro dispositivo che provveda a fornirgli il segnale di clock.

Invece può comportarsi da trasmettitore, inviando al master il valore della temperatura misurata, o da ricevitore, leggendo le impostazioni provenienti dal master.

Un microcontrollore può fungere da master o, se interrogato da un altro microcontrollore, da slave.

Detto questo, possiamo definire l'I<sup>2</sup>C come un **bus bidirezionale di tipo seriale, multi-master** (in cui cioè più dispositivi possono assumere il controllo del bus) **con velocità di comunicazione che raggiunge i 100 Kbit/s nella modalità normale (standard mode), i 400 Kbit/s nella modalità veloce (fast mode) e i 3.4 Mbit/s nella modalità ad alta velocità (High Speed Mode I<sup>2</sup>C v.2.0)** (fig. 1).



Figura 1 – Esempio di bus I<sup>2</sup>C

La trasmissione dei dati avviene, come accennato, per mezzo di due sole linee denominate rispettivamente **SDA** (Serial Data) e **SCL** (Serial Clock).

Sulla prima viaggiano effettivamente i bit d'informazione che trasmettitore e ricevitore si scambiano; sulla seconda linea viaggia il segnale di clock, generato sempre ed esclusivamente da un master (sia esso il trasmettitore o il ricevitore), il cui scopo è quello di sincronizzare i dispositivi stabilendo la validità e il significato dei bit presenti sulla linea dati.

Se sul bus sono presenti più master ciascuno dovrà generare il proprio segnale di clock.

Naturalmente, per evitare situazioni di conflitto in cui più master tentano di assumere contemporaneamente il controllo del bus, è stato previsto un meccanismo di arbitraggio: le uscite SCL e SDA dei dispositivi I<sup>2</sup>C sono di tipo open-drain o open-collector (fig. 2).

Questo, se da una parte richiede che le due linee siano collegate all'alimentazione positiva tramite due resistenze di

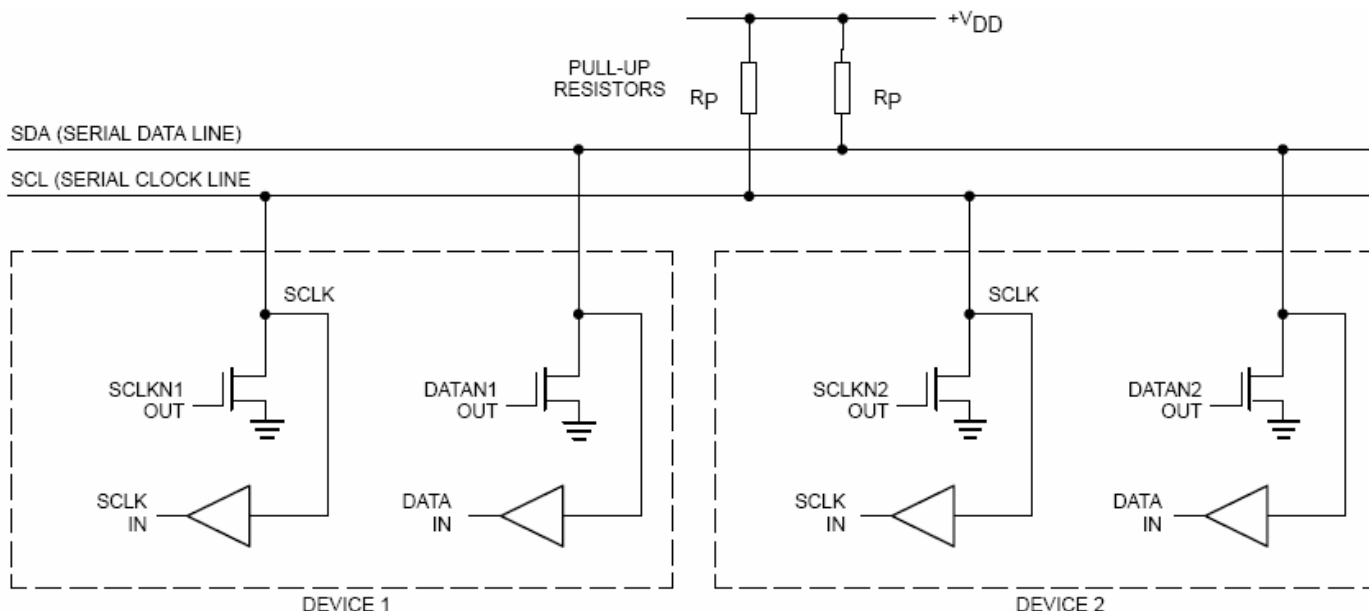


Figura 2 – Connessione dei dispositivi I<sup>2</sup>C sul bus

pull-up, dall'altra fa sì che tutti i dispositivi collegati al bus formino un *and cablato*, ossia diano luogo ad una situazione in cui le due linee rimangono alte se e solo se tutte le uscite sono alte, viceversa è sufficiente che una sola uscita vada bassa perché la linea corrispondente si porti nello stato logico basso.

In questa maniera se un master sta trasmettendo un livello alto mentre un altro invia un livello basso, esso riuscirà a rilevare il conflitto in quanto il livello sulla linea non corrisponde a quello da lui inviato disattivando la propria uscita, mentre il bus risulta automaticamente assegnato al dispositivo che trasmette un livello basso.

Con le due resistenze di pull-up si esaurisce l'hardware da montare per utilizzare il bus I<sup>2</sup>C.

Naturalmente la loro scelta non è casuale, ma dipende da considerazioni sulla velocità di trasmissione (legata alla capacità complessiva della linea) e sui margini di rumore, come vedremo più avanti.

Analizziamo ora il protocollo di comunicazione, ossia l'insieme di regole che i dispositivi devono seguire per comunicare con successo.

Abbiamo detto che sulla linea SDA viaggiano i dati, mentre sulla linea SCL viaggia il clock generato dal master di turno. I dati sulla linea SDA sono considerati validi in corrispondenza dei livelli alti del clock.

Ciò vuol dire che il dispositivo che sta trasmettendo può cambiare i bit sulla linea dati esclusivamente in corrispondenza dei livelli bassi del clock (fig. 3).

A questa regola esistono solo due eccezioni utilizzate per indicare due situazioni particolari: l'inizio di una comunicazione da parte di un master e la corrispondente chiusura.

Quando un master intende intraprendere una comunicazione deve generare una transizione da alto a basso sulla linea dati in corrispondenza di un livello alto del clock; tale condizione è chiamata condizione di START.

Quando il master intende chiudere la comunicazione, deve generare una transizione da basso ad alto su SDA, in corrispondenza di un livello alto su SCL; tale condizione viene detta di STOP.

L'informazione che deve essere scambiata nel corso della comunicazione, è racchiusa tra una condizione di START e una condizione di STOP (fig. 4).

Nulla vieta, comunque, che un master, terminata la comunicazione con un certo slave, ne intraprenda una nuova con un altro generando una ulteriore condizione di START e producendo una unica condizione finale di STOP. L'unità d'informazione che può essere trasmessa è il byte (8 bit); non esiste un limite teorico al numero di byte che può essere trasmesso nel corso di un singolo trasferimento.

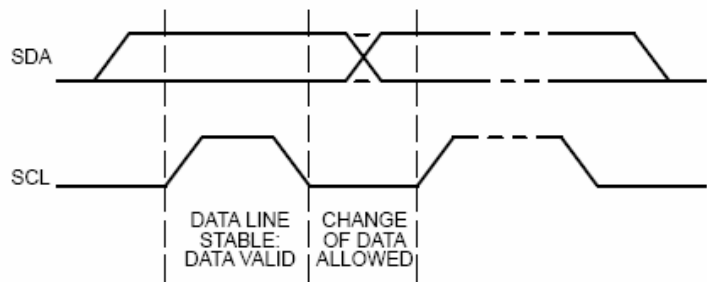


Figura 3 – Trasferimento dei bit

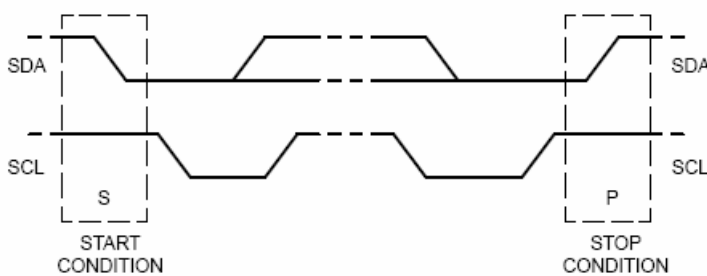


Figura 4 – Condizioni di START e di STOP

Nell'invio un byte viene posto *per primo il bit più significativo*.

In seguito all'invio di un byte il trasmettitore rilascia temporaneamente la linea dati (ovvero la pone nello stato alto) per consentire al ricevitore di porvi un bit di acknowledge che testimoni l'avvenuta ricezione.

Pertanto una comunicazione elementare richiede nove colpi di clock: otto per il byte e uno per l'acknowledge (fig. 5).

Il primo byte inviato successivamente alla condizione di START, contiene l'indirizzo dello slave con il quale si desidera stabilire la comunicazione e l'indicazione sul verso in cui avverrà la comunicazione stessa.

In particolare i primi sette bit codificano l'indirizzo dello slave e l'ottavo (R/W), se alto, indica una richiesta di dati, se basso indica una trasmissione (fig. 6).

Quando viene ricevuto un indirizzo ciascuno slave lo confronta con il proprio.

Se viene trovata una corrispondenza, lo slave interessato risponde con un acknowledge e si pone a disposizione del master a seconda dello stato del bit R/W.

Nella figura 7 è riportato il diagramma temporale di un trasferimento dati completo

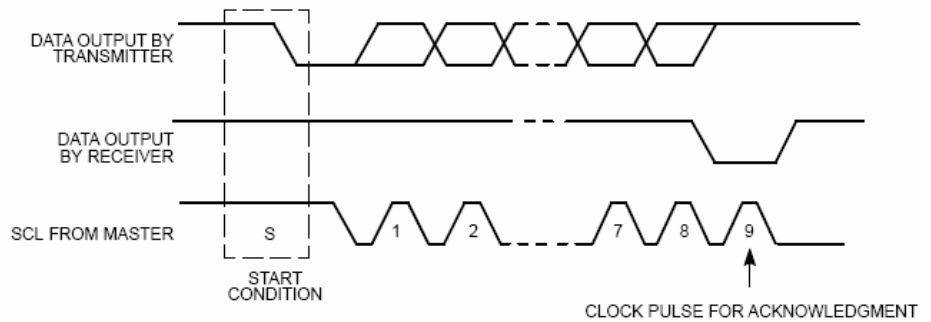


Figura 5 – Invio del primo byte e bit di acknowledge



Figura 6 – Struttura del primo byte

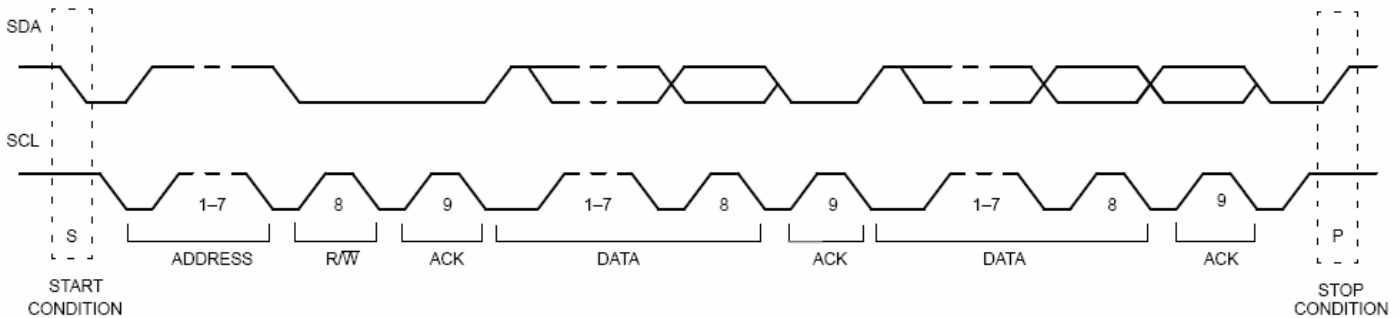


Figura 7 – Esempio di trasferimento dati

A questo punto occorre spendere qualche parola sul modo in cui vengono assegnati gli indirizzi ai vari dispositivi.

L'I<sup>2</sup>C, nella sua versione classica, destina, come detto, 7 bit all'identificazione di ciascuno slave; questo comporta che siano indirizzabili  $2^7=128$  dispositivi diversi (in realtà 112 perché alcuni indirizzi sono riservati).

Sfruttare appieno questo spazio d'indirizzamento comporta che sul package di ciascun integrato I<sup>2</sup>C compatibile, vi siano 7 pin destinati esclusivamente alla sua identificazione.

Spesso questo è inaccettabile per ragioni di spazio e, al fine di ridurre le dimensioni del contenitore, si adotta di solito la seguente soluzione: sul chip i sette bit d'indirizzo vengono suddivisi in due parti di cui una viene fissata definitivamente dal costruttore e non è resa accessibile, l'altra rimane invece programmabile da parte dell'utente.

Ad esempio nel caso di un LM75 sono accessibili solo i tre bit meno significativi  $A_2$ ,  $A_1$  e  $A_0$ , mentre i rimanenti quattro sono collegati all'alimentazione positiva o negativa all'interno del chip stesso, per cui l'indirizzo dello slave è del tipo riportato in figura 8.

È evidente, allora, come il numero di bit programmabili, a disposizione per un integrato di un certo tipo, stabilisca il numero massimo di dispositivi di quel tipo collegabili al bus.

Nel nostro esempio saranno collegabili solo otto LM75.

Esiste anche una versione estesa dell'I<sup>2</sup>C in cui il formato di indirizzamento degli slave è esteso da 7 a 10 bit per venire incontro all'esigenza di disporre di un maggiore numero di indirizzi.

Per conservare la compatibilità con il formato a 7 bit, la soluzione adottata prevede l'invio di un primo byte (quello immediatamente successivo alla condizione di START) contenente cinque bit fissi nella configurazione 11110, i primi due

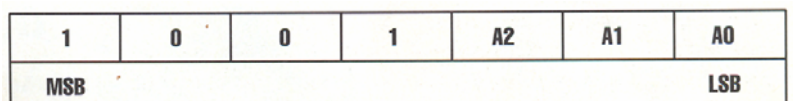


Figura 8 – Esempio d'indirizzo di uno slave LM75

bit dell'indirizzo e il bit di R/W, ed un secondo byte contenente i restanti otto bit dell'indirizzo.

Ciascun byte è seguito dall'invio di un bit di acknowledge da parte dello slave.

Quando viene inviato un indirizzo a 10 bit ciascuno slave confronta i primi due bit con i propri: probabilmente più dispositivi troveranno una corrispondenza e risponderanno con un primo acknowledge A1.

In seguito, questi stessi, confronteranno il secondo byte d'indirizzo con il proprio, e solo uno risponderà con un secondo acknowledge A2 (fig. 9).

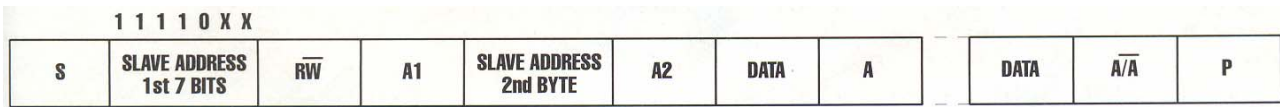


Figura 9 – Indirizzamento a 10 bit

Questo slave rimarrà indirizzato dal master fintantoché non troverà sul bus una condizione di STOP o una nuova condizione di START con un differente indirizzo di slave.

Nella comunicazione tra un master e uno slave, possono insorgere due condizioni potenzialmente dannose per il corretto scambio di dati:

- a livello di byte, lo slave è in grado di ricevere i bytes provenienti dal master alla velocità da questo impostata, ma necessita di un certo tempo per gestire quanto ricevuto oppure per preparare i bytes da trasmettere al master, tempo che non viene concesso dal master che continua a comandare la linea SCL alla sua frequenza;
- a livello di bit, lo slave potrebbe non essere sufficientemente veloce nelle sue operazioni interne per mantenere la sincronizzazione con il master.

A questi problemi di **sincronizzazione** pone rimedio l'and cablato.

Viene sfruttato dallo slave che, in condizioni critiche, può mantenere bloccata a livello basso la linea SCL forzando il master in uno stato di attesa, in una sorta di procedura di handshake: il master, infatti, non può procedere all'emissione di un livello alto sulla linea SCL finché questa viene tenuta a '0' da un altro dispositivo.

Pertanto nel primo caso lo slave mantiene a livello basso SCL finché non ha correttamente riconosciuto il byte arrivato o prodotto il byte da trasmettere.

Nel secondo caso lo slave mantiene a livello basso SCL, estendendone il semiperiodo, e in definitiva rallentando il clock del bus: la velocità del master è così adattata alla velocità interna dello slave più lento.

La figura 10 mostra un esempio di sincronizzazione del clock, dove si suppone che CLK1 sia il clock interno generato dal master e CLK2 il segnale presente sulla linea di clock interna allo slave.

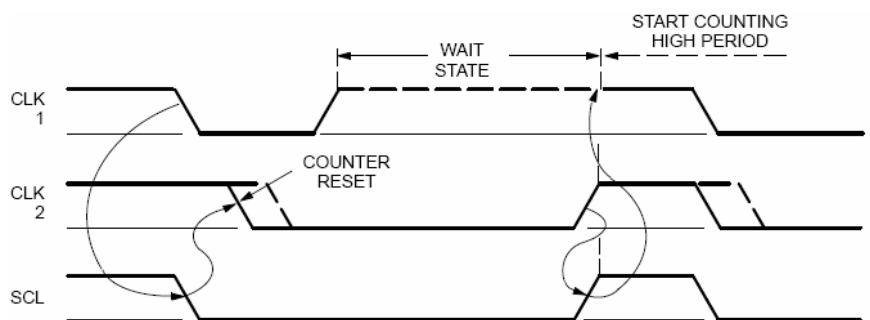


Figura 10 – Sincronizzazione del clock

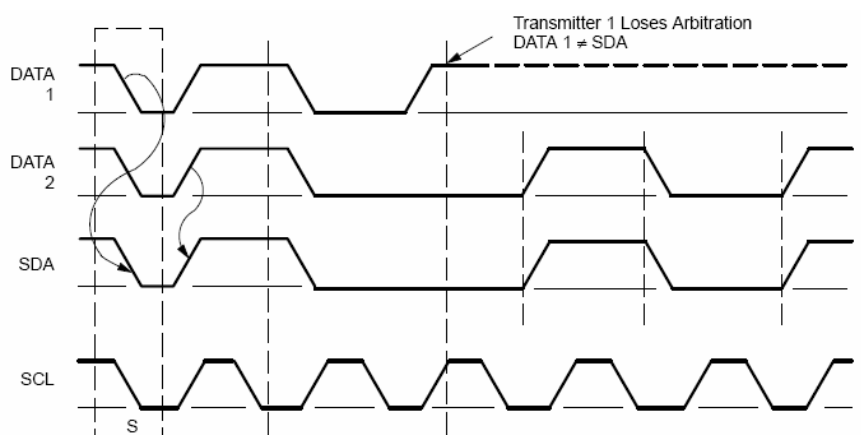
Un master può iniziare una comunicazione solamente se il bus è libero; due o più master potrebbero però generare la condizione di START "in contemporanea".

In tal caso ha inizio un processo, denominato **arbitraggio**, che opera sulla linea SDA quando SCL si trova a livello alto. Il processo, che sfrutta l'and cablato, consiste nel paragonare ciò che si trasmette con quello che effettivamente si trova sulla linea SDA.

Quando due dispositivi trasmettono due livelli differenti, quello prevalente risulta essere il livello basso.

Se un master trasmette su SDA un livello '0' e un altro trasmette su SDA un livello '1', quest'ultimo disabilita il suo stato d'uscita, poiché sulla linea vede un valore diverso rispetto a quello che tenta di trasmettere.

La figura 11 illustra un esempio d'arbitraggio fra due elementi.



4 Figura 11 – Procedura di arbitraggio tra due master ELT - 06/01/2008

Veniamo alle caratteristiche elettriche del bus e dei dispositivi I<sup>2</sup>C.

Nella tabella successiva sono riportate le specifiche elettriche che devono essere rispettate affinché gli stadi di I/O dei dispositivi I<sup>2</sup>C possano connettersi correttamente alle linee SDA e SCL.

PARAMETER	SYMBOL	STANDARD-MODE DEVICES		FAST-MODE DEVICES		UNIT
		Min.	Max.	Min.	Max.	
LOW level input voltage: fixed input levels V <sub>DD</sub> -related input levels	V <sub>IL</sub>	-0.5 -0.5	1.5 0.3V <sub>DD</sub>	-0.5 -0.5	1.5 0.3V <sub>DD</sub>	V
HIGH level input voltage: fixed input levels V <sub>DD</sub> -related input levels	V <sub>IH</sub>	3.0 0.7V <sub>DD</sub>	*1) *1)	3.0 0.7V <sub>DD</sub>	*1) *1)	V
Hysteresis of Schmitt trigger inputs: fixed input levels V <sub>DD</sub> -related input levels	V <sub>hys</sub>	n/a n/a	n/a n/a	0.2 0.05V <sub>DD</sub>	- -	V
Pulse width of spikes which must be suppressed by the input filter	t <sub>SP</sub>	n/a	n/a	0	50	ns
LOW level output voltage (open drain or open collector): at 3 mA sink current at 6 mA sink current	V <sub>OL1</sub> V <sub>OL2</sub>	0 n/a	0.4 n/a	0 0	0.4 0.6	V
Output fall time from V <sub>IHmin</sub> to V <sub>ILmax</sub> with a bus capacitance from 10 pF to 400 pF: with up to 3 mA sink current at V <sub>OL1</sub> with up to 6 mA sink current at V <sub>OL2</sub>	t <sub>of</sub>	- n/a	250 <sup>3)</sup> n/a	20 + 0.1C <sub>b</sub> <sup>2)</sup> 20 + 0.1C <sub>b</sub> <sup>2)</sup>	250 250 <sup>3)</sup>	ns
Input current each I/O pin with an input voltage between 0.4 V and 0.9V <sub>DDmax</sub>	I <sub>i</sub>	-10	10	-10 <sup>4)</sup>	10 <sup>4)</sup>	μA
Capacitance for each I/O pin	C <sub>i</sub>	-	10	-	10	pF

n/a = not applicable

1. Maximum V<sub>IH</sub> = V<sub>DDmax</sub> + 0.5 V

2. C<sub>b</sub> = capacitance of one bus line in pF.

3. The maximum t<sub>f</sub> for the SDA and SCL bus lines quoted in Table 4 (300 ns) is longer than the specified maximum t<sub>of</sub> for the output stages (250 ns). This allows series protection resistors (R<sub>s</sub>) to be connected between the SDA/SCL pins and the SDA/SCL bus lines as shown in Figure 37 without exceeding the maximum specified t<sub>f</sub>.

4. I/O pins of fast-mode devices must not obstruct the SDA and SCL lines if V<sub>DD</sub> is switched off.

A causa della varietà delle tecnologie con cui sono realizzati i componenti interfacciabili al bus I<sup>2</sup>C (CMOS, NMOS, BiCMOS e bipolare), come si può notare dalla tabella i livelli di tensione in ingresso corrispondenti allo '0' logico e all' '1' logico possono essere fissi oppure dipendenti dal valore di V<sub>DD</sub>.

I dispositivi con livelli d'ingresso fissati a 1.5 V (V<sub>ILMAX</sub>) e a 3.0 V (V<sub>IHMIN</sub>) possono mantenere la propria tensione di alimentazione, mentre i resistori di pull-up devono essere collegati a +5V ± 10% (fig. 12).

I dispositivi con livelli d'ingresso dipendenti dal valore di V<sub>DD</sub> devono avere una linea comune di alimentazione alla quale si collegano i resistori di pull-up (fig. 13).

Quando sono presenti dispositivi con livelli d'ingresso sia fissi sia dipendenti da V<sub>DD</sub>, questi ultimi devono essere connessi a un'unica alimentazione +5V ± 10%, come pure i resistori di pull-up (fig. 14).

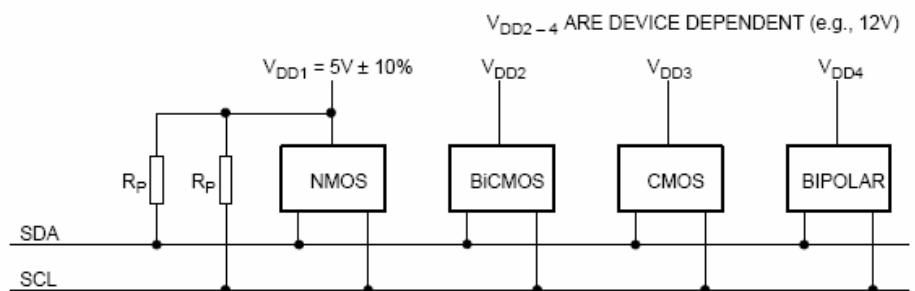


Figura 12 – Dispositivi con livelli d'ingresso fissi collegati al bus I<sup>2</sup>C

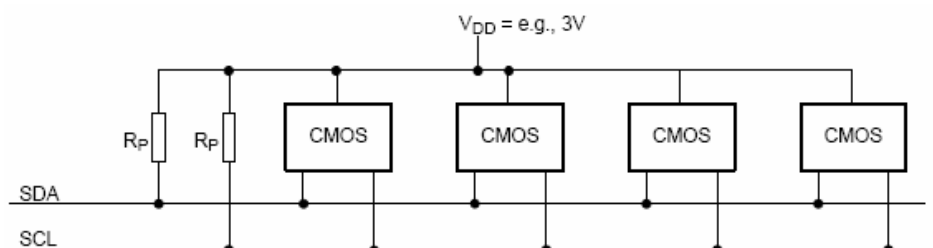


Figura 13 – Dispositivi con livello di ingresso dipendenti da V<sub>DD</sub> collegati al bus I<sup>2</sup>C



Come mostrato in figura 15, dei resistori  $R_S$  di valore (di norma) non superiore ai 300  $\Omega$ , possono essere utilizzati come protezione da spike eventualmente indotti sulle linee SDA ed SCL.

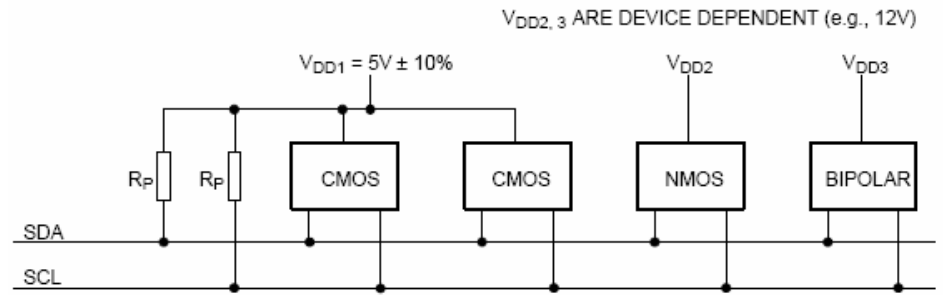


Figura 14 – Dispositivi con livelli di ingresso dipendenti da  $V_{DD}$  (alimentazione  $V_{DD1}$ ) e dispositivi con livello di ingresso fissi  $V_{DD}$  (alimentazione  $V_{DD2}$ ,  $V_{DD3}$ ) collegati al bus I<sup>2</sup>C

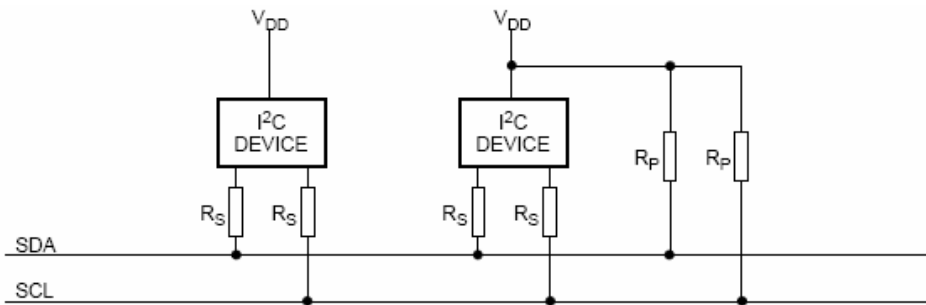


Figura 15 - Resistori serie ( $R_S$ ) per la protezione da spikes

Per i livelli logici sono previsti dei margini di rumore pari a  $0.1 V_{DD}$  per il livello basso e  $0.2 V_{DD}$  per il livello alto.

Un aspetto da considerare è la scelta del valore per le due resistenze di pull-up  $R_P$ : esso dipende dalla tensione di alimentazione, dalla capacità del bus e dal numero dei dispositivi connessi. Poiché si richiede che la massima tensione d'uscita ancora accettabile come uno zero logico dagli altri dispositivi sia  $V_{OL,MAX} = 0.4V$  con una corrente massima  $I_{MAX} = 3 mA$  (fig. 16) risulterà un limite inferiore per il valore di  $R_P$  pari a

$$R_{P \min} = \frac{V_{DD} - 0.4}{0.003}$$

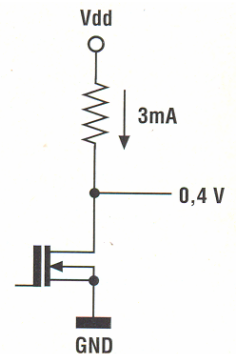


Figura 16 – Valore minimo della tensione di uscita dei driver I<sup>2</sup>C

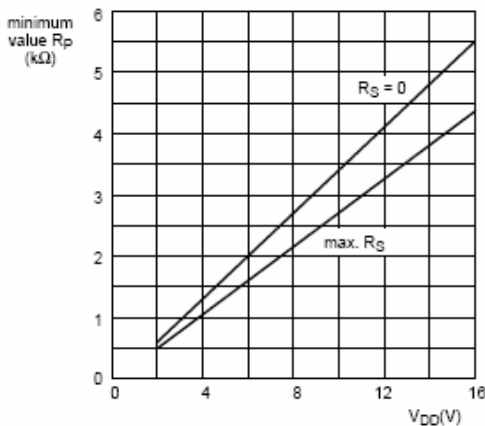


Figura 17 – Andamento del valore minimo della resistenza di pull-up in funzione della tensione di alimentazione

Ad esempio, se  $V_{DD} = 5V$ , le resistenze di pull-up non devono avere valore inferiore a 1.5 k $\Omega$ .

Il valore minimo delle resistenze dipende dalla tensione di alimentazione e varia con esso in modo lineare, come in fig. 17.

D'altra parte, la capacità del bus (ossia la capacità parassita che si distribuisce sui due fili) sommata a quella degli stadi d'ingresso dei dispositivi collegati, limita il massimo valore accettabile per  $R_P$  (fig. 18): ciò per evitare che i fronti di salita dei segnali sul bus presentino

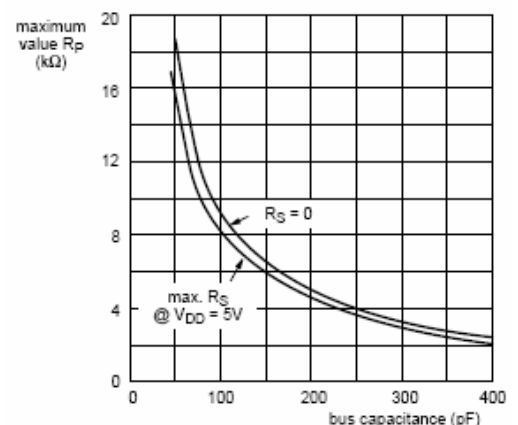


Figura 18 – Andamento del valore massimo della resistenza di pull-up in funzione della capacità elettrica del bus

pendenze poco ripide e non rispondenti allo standard.

Da un punto di vista pratico è possibile considerare 4.7 k $\Omega$  un valore adeguato per una gran parte di applicazioni.

Infatti, ipotizzando una tensione d'alimentazione di 5 V, con tale valore di resistenza è possibile pilotare fino a 200 pF, ossia più di 20 integrati.

Come appena introdotto, un parametro importante (riportato nella tabella successiva) è il valore di capacità massima

che ciascuna linea del bus può venire ad assumere in conseguenza al collegamento su di essa degli stadi di uscita di circuiti integrati I<sup>2</sup>C: le specifiche Philips fissano tale valore a 400 pF.

Considerando che per ciascun dispositivo viene stabilito un valore di capacità di ingresso non superiore ai 10 pF, risulta che il numero massimo di componenti collegabili sul bus I<sup>2</sup>C è superiore a 40.

PARAMETER	SYMBOL	STANDARD-MODE I <sup>2</sup> C-BUS		FAST-MODE I <sup>2</sup> C-BUS		UNIT
		Min.	Max.	Min.	Max.	
SCL clock frequency	$f_{SCL}$	0	100	0	400	kHz
Bus free time between a STOP and START condition	$t_{BUF}$	4.7	–	1.3	–	$\mu$ s
Hold time (repeated) START condition. After this period, the first clock pulse is generated	$t_{HD:STA}$	4.0	–	0.6	–	$\mu$ s
LOW period of the SCL clock	$t_{LOW}$	4.7	–	1.3	–	$\mu$ s
HIGH period of the SCL clock	$t_{HIGH}$	4.0	–	0.6	–	$\mu$ s
Set-up time for a repeated START condition	$t_{SU:STA}$	4.7	–	0.6	–	$\mu$ s
Data hold time: for CBUS compatible masters (see NOTE, Section 9.1.3) for I <sup>2</sup> C-bus devices	$t_{HD:DAT}$	5.0 0 <sup>1)</sup>	– –	– 0 <sup>1)</sup>	– 0.9 <sup>2)</sup>	$\mu$ s $\mu$ s
Data set-up time	$t_{SU:DAT}$	250	–	100 <sup>3)</sup>	–	ns
Rise time of both SDA and SCL signals	$t_r$	–	1000	$20 + 0.1C_b$ <sup>4)</sup>	300	ns
Fall time of both SDA and SCL signals	$t_f$	–	300	$20 + 0.1C_b$ <sup>4)</sup>	300	ns
Set-up time for STOP condition	$t_{SU:STO}$	4.0	–	0.6	–	$\mu$ s
Capacitive load for each bus line	$C_b$	–	400	–	400	pF

All values referred to  $V_{IHmin}$  and  $V_{ILmax}$  levels (see Table 3).

1. A device must internally provide a hold time of at least 300 ns for the SDA signal (referred to the  $V_{IHmin}$  of the SCL signal) in order to bridge the undefined region of the falling edge of SCL.
2. The maximum  $t_{HD:DAT}$  has only to be met if the device does not stretch the LOW period ( $t_{LOW}$ ) of the SCL signal.
3. A fast-mode I<sup>2</sup>C-bus device can be used in a standard-mode I<sup>2</sup>C-bus system, but the requirement  $t_{SU:DAT} \geq 250$  ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCL signal. If such a device does stretch the LOW period of the SCL signal, it must output the next data bit to the SDA line  $t_{rmax} + t_{SU:DAT} = 1000 + 250 = 1250$  ns (according to the standard-mode I<sup>2</sup>C-bus specification) before the SCL line is released.
4.  $C_b$  = total capacitance of one bus line in pF.

Per ultimo, in figura 19 sono rappresentate le temporizzazioni relative a SDA e SCL in riferimento ai valori dei parametri indicati nella tabella precedente.

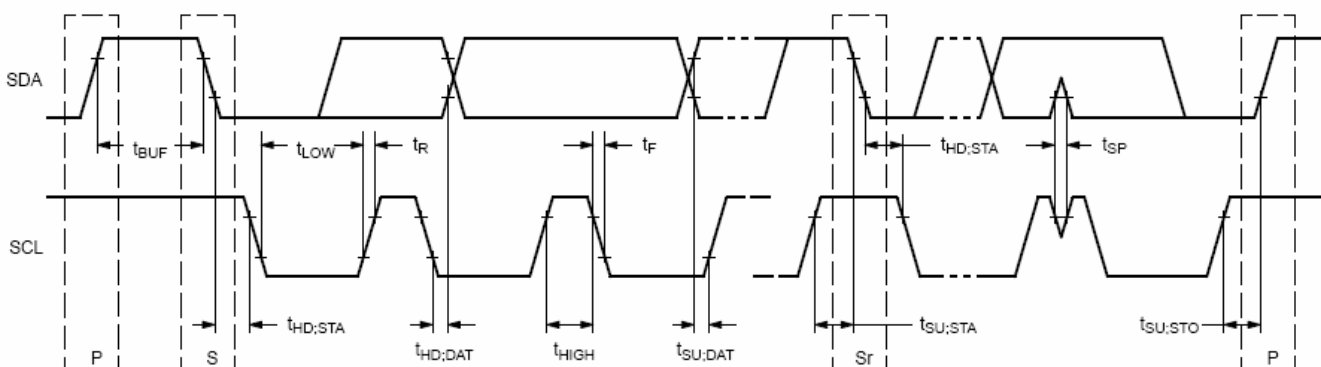


Figura 19 – Definizione delle temporizzazioni sul bus I<sup>2</sup>C