INTRODUZIONE AL BUS IIC

È uno standard da tempo impiegato in molti apparati consumer e anche in alcuni progetti proposti nei mesi passati. Ci siamo però accorti che non tutti conoscono i fondamenti su cui si basa questo sistema di trasmissione dati

di Andrea Sbrana - 1ª parte

Il bus IIC è stato ideato dalla Philips con lo scopo di far dialogare più chip e periferiche con il minor numero di fili possibile, ma al tempo stesso con la massima affidabilità.

I primi esperimenti furono eseguiti con dialoghi tra prodotti per la cucina, come forno a microonde, frigorifero, lavastoviglie, poi, vista la piena funzionalità ed affidabilità, questo bus fu impiegato anche per far dialogare chip all'interno di una stessa scheda, come per esempio il telaio di un televisore.

Oggi, infatti, una delle più comuni apparecchiature consumer che fa uso del bus IIC è proprio il televisore: ci sono collegati i controller di gestione, il VFO, la memoria EEPROM, il ricevitore del telecomando, il decoder del televideo e quello del PAL, i sincronizzatori video, i PLL, i processori audio, ecc.

Ma l'IIC bus viene anche impiegato con successo nei telefoni cellulari, nelle centraline auto, nelle schede di comunicazione, nei controller industriali e, comunque, dove non sia necessaria un'elevata velocità, perché, come vedremo, la massima velocità standard raggiungibile è di 100 kHz, anche se poi ci sono periferiche dette FAST che possono anche raggiungere i 400 kHz ed alcune anche 1 MHz.

Lo standard IIC bus è nato infatti per rispondere a poche ma importanti regole: numero ridotto di componenti sulla scheda, linee di connessione (quindi costi) ridotte al minimo, velocità non eccessiva.

Per realizzare sistemi con queste caratteristiche, era necessario ideare una struttura a bus di tipo seriale. Per struttura a bus, si intende una architettura in cui tutti i chip del circuito sono collegati per mezzo di linee comuni, mentre per "seriale" si intende che l'informazione passa serialmente, ovvero su di una sola linea un dato dopo l'altro.

Per facilitare la comprensione, pensiamo allo schema di un computer attuale: il processore è connesso (vedi Figura 1) alle periferiche con un bus PARALLELO, ovvero sia i dati che gli indirizzi vengono passati uno per ogni filo, quindi il dato D0 passerà sempre per il filo D0, il dato D1 per il filo D1 e così via.

Chiaramente, in questo modo la velocità è la massima possibile, perché in un solo colpo di clock viene passato sia un indirizzo completo (esempio 32 bit), sia un dato (altri 32 bit) che altri segnali di controllo, ma ovviamente per ogni linea, sullo stampato dovrà corrispondere una pista fisica.

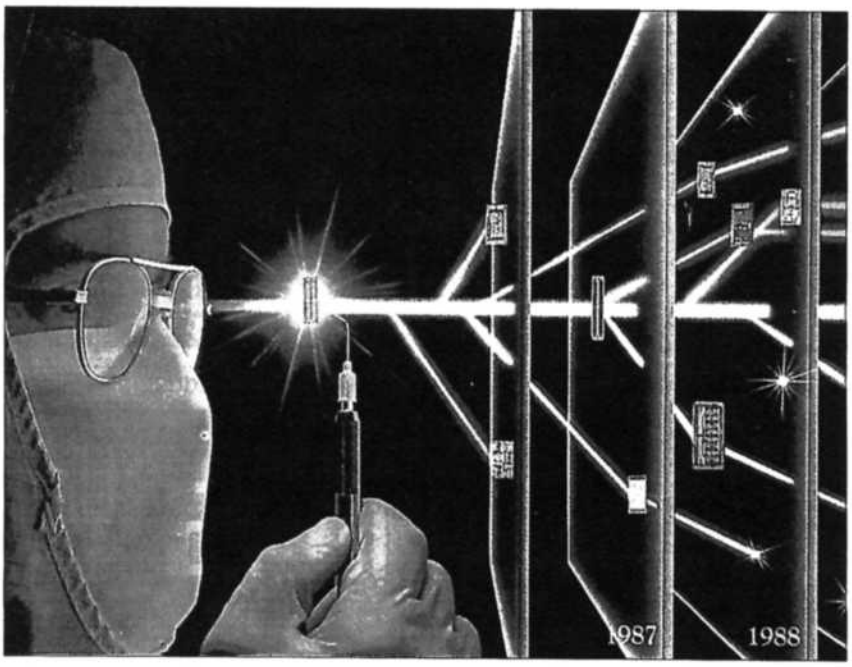
In Figura 2 possiamo vedere la struttura di un bus IIC, ovvero con due soli fili. Questa volta, per far passare un ipotetico indirizzo di 32 bit e un dato di altri 32 ci vorranno almeno 64 colpi di clock, uno per ogni bit, più quelli per i segnali di controllo. Capite quindi la differenza tra un collegamento di tipo PARALLELO ed un altro di tipo SERIALE. Poiché nella maggior parte delle applicazioni consumer non è rilevante la velocità di esecuzione, i costruttori preferiscono realizzare piste con pochi collegamenti a scapito della velocità.

Analisi dei collegamenti

Il bus IIC supporta sia processori che periferiche di tipo NMOS, CMOS, IIL. Sono necessari, ed al tempo stesso sufficienti, due soli fili di collegamento: SDA (Serial DAta) e SCL (Serial CLock) che si incaricano rispettivamente di portare i dati e di sincronizzare i vari dialoghi.

Ogni chip IIC viene univocamente determinato da un indirizzo e può sia ricevere che trasmettere informazioni con un protocollo che poi vedremo. Chiaramente ci saranno anche periferiche che saranno solo abilitate alla sola ricezione, come un controller per display LCD, altre abilitate alla sola trasmissione, come interfacce per tastiere, altre infine abilitate ad entrambe le funzioni, come per esempio le memorie.

Vedremo poi che ci sono anche periferiche che possono modificarsi da master in slave (dopo chiariremo anche questo concetto) e viceversa, come per esempio i microcontroller. Il bus IIC infatti è stato concepito come MULTI-MASTER-BUS,



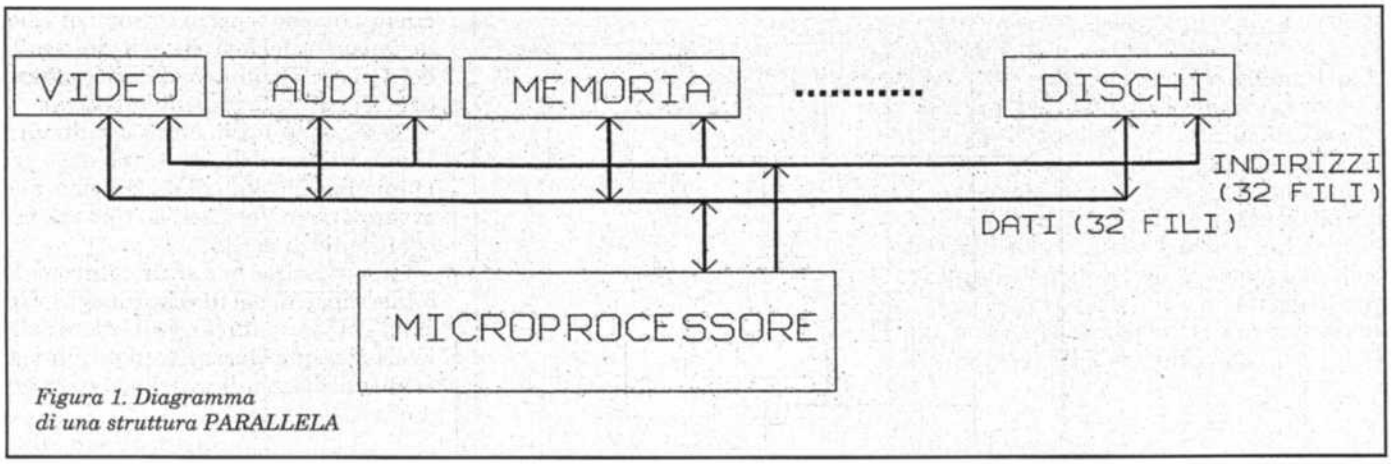


Figura 1. Diagramma di una struttura PARALLELA

ovvero che permette la coesistenza di più master, ovviamente in attività uno per volta.

A grandi linee, il dialogo tra due chip A e B, potrebbe svolgersi nel seguente modo: se A vuole inviare un'informazione a B, allora:

- A inizia il dialogo (se possibile)
- A (in questo caso MASTER) invia l'indirizzo di B (SLAVE)
- A invia il dato (o i dati) a B
- A termina il dialogo

Se, invece, A vuole ricevere informazioni da B allora:

- A inizia il dialogo (se possibile)
- A invia l'indirizzo di B
- A attende il dato da B
- A termina il dialogo

Abbiamo specificato che A inizia il dialogo solo se è possibile, ovvero se il bus è ritenuto LIBERO, e non impegnato da altre comunicazioni.

La sincronizzazione del dialogo avviene, negli esempi precedenti, sempre da A. Per prevenire collisioni sul bus, è prevista una procedura di arbitrazione del bus.

Terminologia del bus IIC

TRANSMITTER
Il chip che invia dati sul bus
RECEIVER
Il chip (o i chip) che riceve dati dal bus
MASTER
Il chip che inizia un dialogo, genera il clock di riferimento e poi termina il dialogo
SLAVE
Il chip indirizzato dal master
MULTI-MASTER
Quando sono attivi contemporaneamente più chip alla volta
ARBITRATION
Meccanismo che garantisce la correttezza dei dialoghi
SYNCHRONIZATION
Procedura di sincronizzazione tra due o più chip

Caratteristiche generali

In Figura 3 possiamo vedere come ogni singola periferica venga connessa al bus IIC ed in particolare anche l'equivalente elettrico sui due segnali SDA e SCL.

Quest'ultimi sono entrambi bidirezionali e connessi al positivo per mezzo di una resistenza di pull-up.

Quando il bus è libero, entrambe le linee sono mantenute ad alto livello logico tramite queste resistenze di pull-up. Il numero di periferiche che è possibile connettere al bus è quasi illimitato, poiché vengono offerti "ripetitori" di segnali per aumentare le distanze tra i chip.

L'alimentazione dei chip sotto bus IIC è rigorosamente di 5 volt: lo standard impiegato da tutti i chip delle ultime generazioni, ma ci sono periferiche che riescono a funzionare anche sotto i 3 volt.

I valori delle due resistenze di pull-up, non devono essere casuali, ma vanno calcolati in base a specifiche scelte

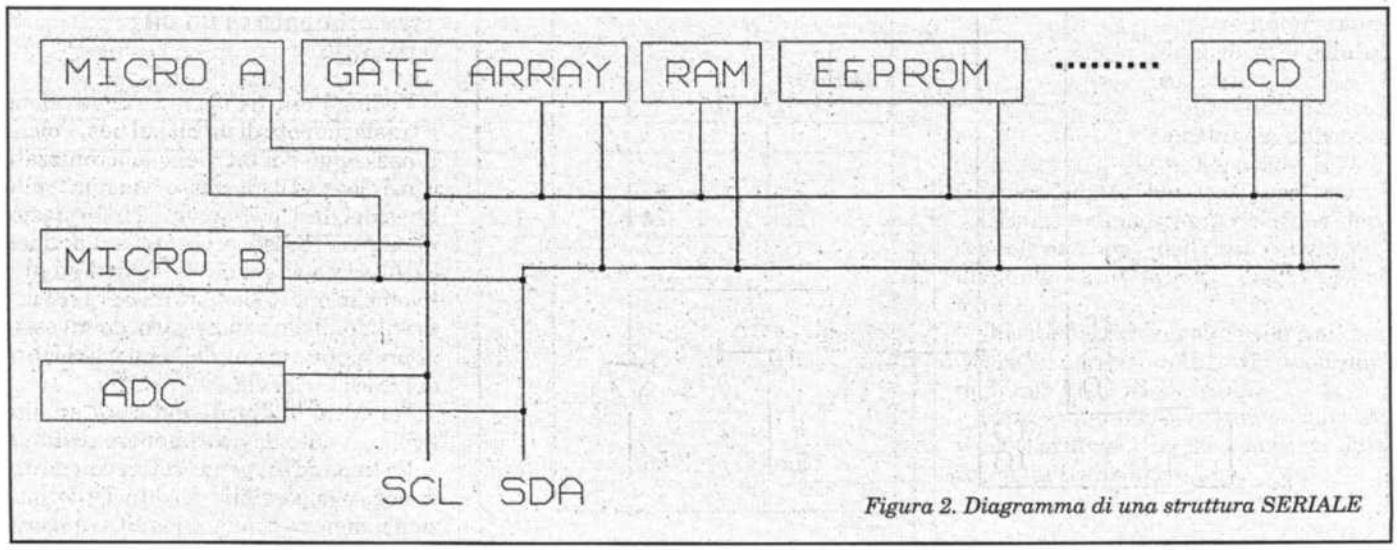


Figura 2. Diagramma di una struttura SERIALE

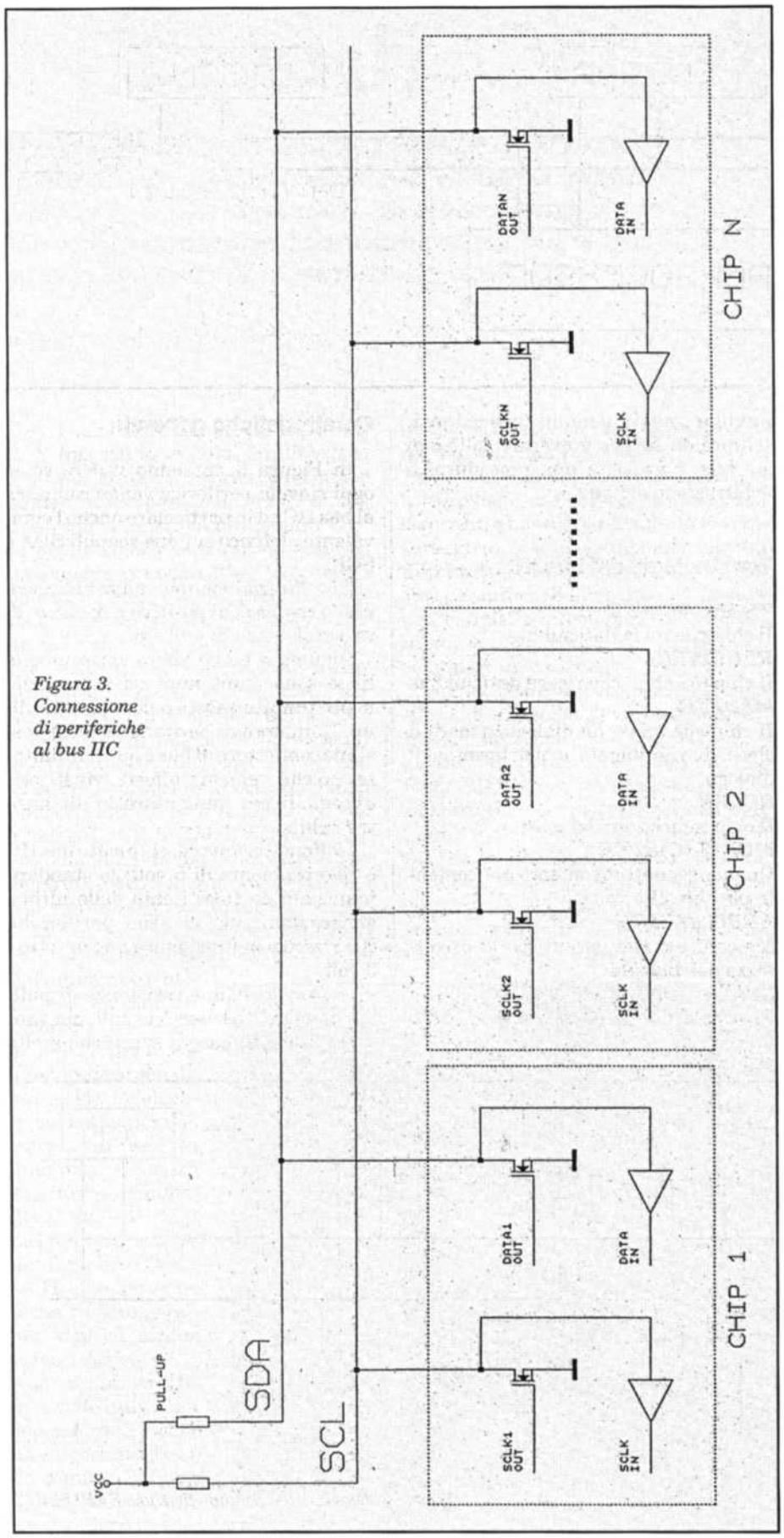


Figura 3. Connessione di periferiche al bus IIC

circuitali come tensione di alimentazione, capacità del bus (che non deve eccedere i 400 pF), numero di chip connessi (che identifica la corrente relativa).

La tensione di alimentazione limita il valore minimo delle due resistenze per la minima corrente di sink dichiarata, ovvero 3 mA a Vol-max, cioè 0,4 volt per ogni stadio di uscita.

La capacità del bus è calcolata come la totale capacità del filo che collega i vari chip. Tale capacità limita il valore delle resistenze per garantire dei fronti di salita e di discesa di massimo 1 microsecondo.

Condizione di Start e di Stop

Con condizione di start e stop viene inteso rispettivamente il protocollo di inizio di un dialogo e quello di terminazione. In Figura 4 possiamo vedere entrambe queste condizioni.

La condizione di START può ovviamente avvenire solo quando il bus è libero, ovvero solo quando sia la linea SDA che quella SCL si trovano ad alto livello.

Allora abbiamo una condizione di START se, mantenendo la linea SCL ad alto livello, si porta la linea SDA a basso livello. Successivamente sarà possibile il dialogo sincronizzandoci con il clock e con modalità che adesso vedremo.

La condizione di STOP, invece, si potrà verificare solo al termine di un dialogo, quindi viene identificata quando, con la linea SCL ad alto livello, la linea SDA viene portata da basso ad alto livello. Vedremo che non è possibile avere questa condizione durante il normale dialogo.

Trasferimento di un bit e dialogo

Vediamo ora in Figura 5 come avviene il trasferimento di un bit sul bus. Poiché il passaggio del bit viene sincronizzato con il clock ed il bit stesso "viaggia" sulla linea dei dati, ovviamente l'informazione dovrà essere presente sulla linea SDA prima che il clock si porti ad alto livello. In quel momento il dato è ritenuto valido. Il dato successivo, dovrà essere presentato prima della nuova risalita del clock e così via.

Per tutta la durata del clock ad alto livello, il dato dovrà rimanere costante sulla linea SDA, pena l'interruzione del dialogo o la possibile perdita d'informazione, mentre con la linea SCL a basso

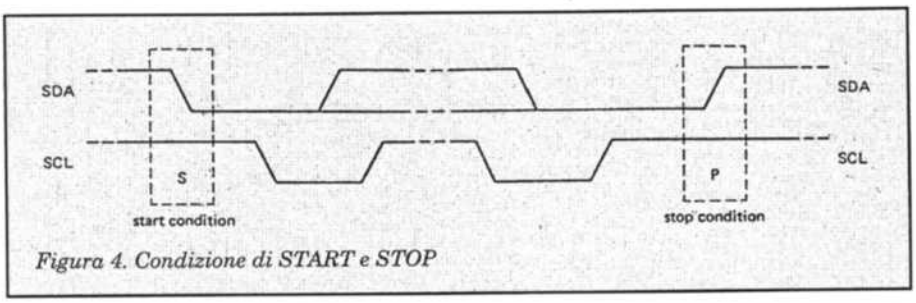


Figura 4. Condizione di START e STOP

livello è possibile anche mettere le periferiche in three-state.

In Figura 6 troviamo un esempio di dialogo che avviene sul bus.

Ogni byte inviato sulla linea SDA, sia esso di dato, di indirizzo o di controllo, deve necessariamente essere di 8 bit.

Al contrario, il numero di byte che possono transitare durante un dialogo può variare a discrezione del master (quindi in definitiva del programmatore del controller che svolge tale funzione).

Ognuno di questi byte però, deve essere seguito da un bit detto di acknowledge, ovvero di riconoscimento.

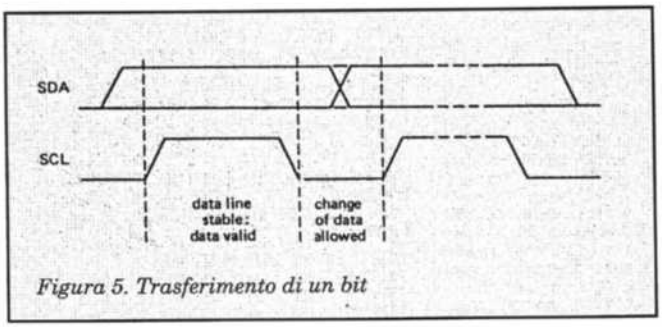


Figura 5. Trasferimento di un bit

Vedremo che alcune periferiche non rispettano però completamente questa direttiva. Il byte viene inviato con il MSB (Major Significant Bit = bit di peso maggiore) per primo.

Quindi, se per esempio volessimo inviare il numero 156 decimale, che in binario viene espresso come 10011100, dovremmo prima inviare "1", poi "0", poi "0", poi "1", poi "1", poi "1", poi "0" ed infine "0".

Ci sono casi in cui il ricevitore, dopo la ricezione di un byte, non possa ricevere altri byte per un certo tempo perché impegnato in attività proprie, come per esempio la memorizzazione di un dato in una memoria di tipo EEPROM che richiede circa 10 mS di tempo. In queste situazioni il ricevitore stesso tiene bassa la linea SCL fino a quando non sarà ancora disponibile a ricevere ancora dati (come il DTR nella RS-232).

Acknowledge

Il trasferimento dei dati con l'acknowledge è obbligatorio sul bus IIC. Il clock relativo al bit di acknowledge viene generato dal master che, al tempo stesso rilascia la linea SDA (livello alto) per verificare se l'acknowledge viene prodotto dallo slave indirizzato. Il chip slave infatti, in corrispondenza dell'impulso di clock relativo all'acknowledge, deve portare a basso livello la linea SDA e tenerla così per tutta la durata dell'impulso stesso (vedi Figura 7).

Generalmente, un ricevitore che è stato indirizzato dal master, è obbligato a generare un acknowledge dopo ogni byte ricevuto). Quando ciò non accade, ad esempio perché lo slave è impegnato, lo stesso slave deve tenere alta la linea SDA.

In questo modo, per recuperare del tempo prezioso, il master può decidere di inviare una condizione di STOP ed interrompere così il collegamento in modo da dedicarsi ad altri compiti.

Allo stesso modo, il master deve abortire il dialogo se lo slave, dopo aver inviato l'acknowledge, si blocca per vari motivi.

Un master, mentre è in ricezione, deve segnalare l'invio dell'ultimo byte non inviando l'acknowledge dopo l'ultimo byte.

Questa pratica è molto utile per "scaricare" in una sola volta tutto il contenuto di una memoria seriale.

Il prossimo mese vedremo come avviene l'arbitrazione ed i formati delle trame in transito sul bus.

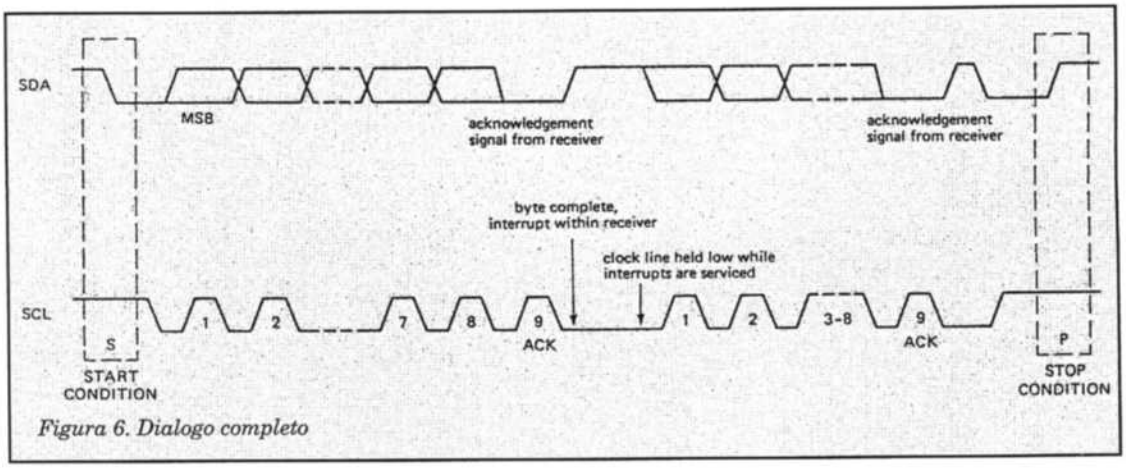


Figura 6. Dialogo completo

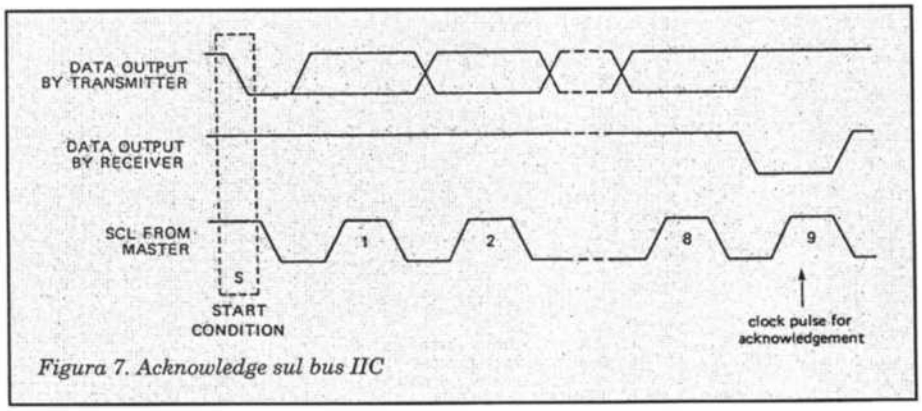


Figura 7. Acknowledge sul bus IIC

continua