

COSTRUIAMOCI UN VERO MICROELABORATORE

HOME COMPUTER AMICO 2000

a cura della A.S.EL

Questo articolo sul microcomputer AMICO 2000 è la seconda parte di quello pubblicato su Sperimentare n° 6/1981 e completa la descrizione della scheda di I/O e del suo funzionamento. Sono qui inclusi utili esercizi per l'apprendimento delle funzioni svolte dai vari componenti. La A.S.EL. si scusa con i lettori del lungo intervallo dovuto a motivi di riorganizzazione interna, e ricorda di essere sempre a disposizione per ogni delucidazione sul sistema AMICO 2000.

In questa parte analizziamo un componente a 40 Pins, il 6522, denominato VIA (versatile interface adapter), che comprende le seguenti funzioni fondamentali:

- 2 porte di Ingresso/Uscita (I/O) parallelo
- 2 timers programmabili
- 1 registro a scorrimento (Shift Register) di Ingresso/Uscita seriale
- La logica di Interrupt e di scambio dati (Handshake)

La piastra A 2000/19 monta due di tali componenti, i cui pins vengono portati su

due connettori a 24 pins per permetterne il collegamento con il mondo esterno (vedere connessioni in fig. 9C 1ª parte).

REGISTRI INTERNI DELLA VIA

I registri della VIA sono 16; per indirizzarli si utilizzano 4 fili di indirizzo (RS0, RS1, RS2, RS3). In particolare si distinguono 4 tipi di registri fondamentali:

- 1 - Registri di I/O parallelo (registri 0-1-2-3-F)
- 2 - Registri di Timer (registri 4-5-6-7-8-9)

- 3 - Registro di Shift (registro A)
- 4 - Registri di controllo (registri B-C-D-E)

In fig. 1 riportiamo la mappa di memoria della VIA 6522.

Sezione di I/O parallelo (PIO)

La sezione PIO dispone di due porte bidirezionali da 8 bits ciascuna. Ogni porta è controllata da un registro di ingresso/uscita chiamato ORA (porta A) e ORB (porta B). Al registro ORA è associato il registro DDRA che determina la

INDIRIZZO		
0	ORB (PB0 ÷ PB7)	Registro B di I/O
1	ORA (PA0 ÷ PA7)	Registro A di I/O
2	DDRB	Registro B di direzione dati
3	DDRA	Registro A di direzione dati
4	T1L-L/T1C-L	Contatore 1 parte bassa
5	T1C-H	Contatore 1 parte alta
6	T1L-L	Memoria contatore 1 parte bassa
7	T1L-H	Memoria contatore 1 parte alta
8	T2L-L/T2C-L	Contatore e memoria 2 parte bassa
9	T2C-H	Contatore 2 parte alta
A	SR	Shift register
B	ACR	Registro ausiliario
C	PCR	Registro di periferica
D	IFR	Registro di interrupts pervenuti
E	IER	Registro di abilitazione interrupts
F	ORA	Registro A di I/O senza possibilità di Handshake

Fig. 1 Mappa di memoria della VIA 6522

direzione dei singoli bit del port A e a ORB il registro DDRB per i bits del port B.

Quando un bit del dato scritto nel DDR A o B è uguale a 1, il bit corrispondente del registro OR A o B è una uscita. Viceversa quando il bit è uno 0, il bit corrispondente è una entrata.

All'accensione della macchina, tutte le porte vengono messe in entrata e se successivamente una porta o parte di essa viene condizionata come uscita, i bit messi in Output sono automaticamente a 0.

Esiste un'asimmetria tra le porte A e B; infatti solo la porta A dispone anche del registro F. Tale registro permette di usare la porta A per normali operazioni di I/O senza l'intervento di eventuali segnali di controllo (Ready, Strobe ecc.) ottenibili invece con il registro I come si vedrà più avanti.

Ma torniamo all'uso dei registri ORA-B e DDRA-B.

Facciamo un esempio: sia DDRA=8A (10001010); avremo perciò che nel port A il 2°, il 4° e l'8° bit sono di uscita mentre i restanti sono di ingresso (vedere fig. 2). È chiaro che per usare le porte parallele bisogna prima caricare i registri DDR con il valore necessario e solo in seguito le porte parallele possono essere usate come locazioni di memoria.

Si voglia per esempio mettere il port A in uscita e il port B in ingresso: scriveremo perciò il seguente programma

```
LDA    ≠$FF
STA    DDRB
LDA    ≠00
STA    DDRA
```

Essendo l'indirizzo di base della VIA

numero 1 sulla scheda FA10, avremo:
 DDRA = indirizzo di base + 03 = FA13
 DDRB = indirizzo di base + 02 = FA12

Il programma scritto in linguaggio macchina sarà quindi:

```
A9    LDA    ≠$FF
FF
8D    STA    DDRA
13
FA
A9    LDA    ≠0
00
8D    STA    DDRB
12
FA
```

Ora il port A può essere usato in uscita come una qualsiasi locazione di memoria. Per esempio possiamo far uscire il dato 01 scrivendo:

```
LDA    =01
STA    ORA
```

Ed essendo ORA = indirizzo di base + 01, avremo:

```
A9    LDA    ≠01
01
8D    STA    ORA
11
FA
```

Per prendere confidenza con le operazioni di I/O parallelo, è consigliabile realizzare una basetta simile allo schema di fig. 3 ed eseguire gli esercizi seguenti:

Esercizio 1 - Connettere in parallelo il port A con il port B collegando rispettivamente fra loro i bits 0, 1, 2, 3, 4, 5, 6, 7 delle due porte; seguire scrupolosamente le indicazioni del connettore JA sullo schema di fig. 9C (1ª parte).

Mettere il port A in uscita e il port B in entrata usando la procedura già vista e cioè scrivendo 00 in FA12 e FF in FA13. A questo punto tutto ciò che viene scritto nel registro ORA (FA11) può essere riletto nel registro ORB (FA10) in quanto l'uscita A viene forzata con il contenuto di ORA e l'ingresso B, essendo collegato all'uscita A, ricarica nel registro ORB lo stesso dato.

Da notare che se rileggiamo ORA vi ritroveremo lo stesso dato che vi abbiamo scritto perchè il port A è completamente in uscita. Se il registro DDRA venisse definito con qualche bit in entrata, rileggendo ORA vi troveremo lo stato presente su quel bit (vedere esercizio 3).

Esercizio 2 - Collegare le basi dei transistor che pilotano i leds (vedere fig. 3) ai pins del port A seguendo il solito schema.

A questo punto potremmo far accendere i leds mettendo a livello logico 1 i bits corrispondenti del port A. Caricando di volta in volta nel registro ORA valori differenti, potremo far accendere a piacere tutti i leds che vogliamo e nella sequenza desiderata.

Come esercizio scrivere un programma che faccia accendere un led per volta facendo cambiare il led che si accende ad ogni pressione di un tasto della tastiera.

Esercizio 3 - Poniamo il port A parzialmente in ingresso e parzialmente in uscita e cioè con i 2 bit alti (PA6 e PA7) in uscita e gli altri 6 in ingresso; per fare ciò scriviamo quindi:

```
LDA    ≠$C0 (C0 = 1100 0000)
STA    DDRA
```

A questo punto possiamo eseguire una

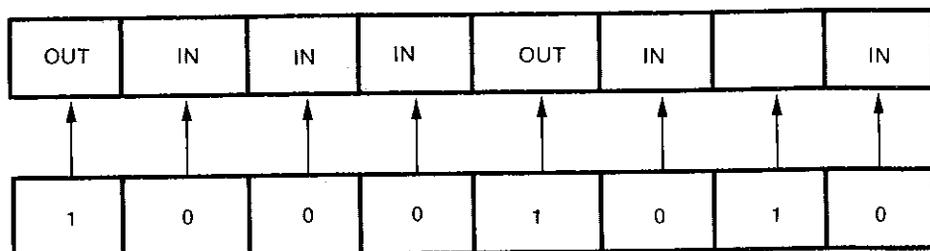


Fig. 2 - Corrispondenza dei bit fra registri OR e DDR

TABELLA 1			
PCR3	PCR2	PCR1	MODO DI FUNZIONAMENTO
∅	∅	∅	Interruzione sul fronte di discesa di CA2. Il bit di interruzione relativo a CA2 nel IFR viene posto a 1 da una transizione negativa su CA2. Il bit IFR0 viene riportato a 0 leggendo o scrivendo nel registro ORA, o scrivendo un 1 in IFR0.
∅	∅	1	Vale quanto detto per il modo 000 con la differenza che il bit IFR0 viene riportato a 0 solo scrivendoci un 1.
∅	1	∅	Interruzione sul fronte di salita di CA2. IFR0 viene riportato a ∅ leggendo o scrivendo il registro ORA.
∅	1	1	Vale quanto detto per il modo 010 con la differenza che IFR0 viene riportato a 0 solo scrivendoci un 1.
1	∅	∅	CA2 è usato in uscita. Il pin si mette a ∅ leggendo o scrivendo il registro ORA. Si riporta a 1 sulla transizione attiva di CA1.
1	∅	1	CA2 è usato come uscita impulsiva. Ad ogni lettura o scrittura di ORA viene generato un impulso negativo su CA2.
1	1	∅	CA2 è una uscita e si mantiene sempre bassa.
1	1	1	CA2 è una uscita e si mantiene sempre alta.

NOTA — Le stesse considerazioni di cui sopra, valgono per i bit PCR7, PCR6, PCR5, in ovvia relazione a CB2.

prova di uscita. Senza collegare nulla ai bits di ingresso del port A, scrivendo il dato 00 in ORA. Rileggendo il registro ORA leggeremo 3F. Questo perchè i 2 bits definiti come uscita sono regolarmente andati a 0 e vengono letti bassi mentre gli altri che sono ingressi vengono letti con ciò che è presente sui corrispondenti pins di ingresso è cioè a 1 in quanto nel nostro caso gli ingressi, che non sono collegati, vengono mantenuti a 1 dalle resistenze di PULL-UP.

Comunque il dato 0 che abbiamo scritto viene memorizzato dal registro di uscita e se cambiamo la direzione dei dati scrivendo FF nel DDRA, vedremo tutte le uscite andare a 0.

Esercizio 4- Poniamo il port A in ingresso e il port B in uscita. Colleghiamo 8 interruttori tra i fili del port A e la massa (GND) e gli 8 circuiti transistors-leds ai pins del port B. Il seguente programma

accenderà solo i leds connessi ai bits i cui corrispondenti del port A avranno gli interruttori chiusi. Analizzate il programma:

```

LDA    #$FF
STA    DDRB
LDA    #$00
STA    DDRA
LOOP   LDA    ORA
        EOR    #$FF
        STA    ORB
        JMP   LOOP
    
```

Perchè è stata introdotta l'istruzione EOR ≠\$FF?

Ad ogni porta di I/O sono associate due linee di controllo chiamate CA1, CA2 e CB1, CB2. Le linee CA1 e CB1 sono di solo ingresso mentre CA2 e CB2 possono essere sia di ingresso che di uscita. Scrivendo particolari bit nella VIA, la configurazione relativa ad ogni linea può essere programmata a piacere. Per esem-

pio la VIA può memorizzare un fronte di salita o di discesa su uno dei pins di controllo CA1 o CB1 ed in seguito avvertire il processor dell'accaduto.

Mediante ciò che si scrive nel PCR (registro di periferica) all'altezza dei bits 0 e 4 è possibile decidere se il fronte attivo deve essere di salita o di discesa; scrivendo uno 0 riveliamo il fronte di discesa, scrivendo un 1 si rivela il fronte di salita.

Il fig. 4 riportiamo in dettaglio i bits del registro PCR.

Il processor viene informato se sul pin selezionato è arrivato un fronte attivo con una semplice lettura dei bit 1 e 4 rispettivamente per CA1 e per CB1, nel registro IFR (registro degli Interrupts pervenuti).

Questo bit rimane a 0 fino a che non è stato ricevuto il segnale atteso, dopodichè si porta a 1. Dopo aver letto tale bit, è necessario resettarlo a 0 per permettere la ricezione di una successiva transizione.

Ciò può essere fatto o scrivendo un 1 nella opportuna posizione del registro IFR, o anche semplicemente leggendo o scrivendo il registro OR corrispondente. Esempio: una periferica parallela è collegata alla porta A e la stessa periferica fornisce un segnale di pronto. Quando questo segnale sale il dato è valido sul port A. (vedere fig. 5).

Il programma relativo è riportato in alto a sinistra alla pagina seguente.

MEMORIZZAZIONE DEGLI INGRESSI

Le porte A e B del 6522 possono avere differenti comportamenti a seconda se sono impiegate come ingressi o come uscite; le uscite sono evidentemente sempre memorizzate mentre gli ingressi non lo sono necessariamente.

Gli ingressi possono essere memorizzati o meno mediante il condizionamento dei bit 0 e 1 (rispettivamente port A e port B) di ACR (registro ausiliario figg. 6 e 7). Quando questo bit è a 0 non avviene la memorizzazione e cioè il processor legge in continuazione lo stato dei bit di ingresso e se questi cambiano, cambia anche la lettura. Il registro è cioè "trasparente".

Quando il bit è a1, avviene la memorizzazione, e cioè lo stato delle linee di ingresso viene trasferito all'interno della VIA sul fronte attivo di CA1 e CB1 (secondo la porta usata) e il processor lo legge stabilmente fino al successivo fronte sul piedino di controllo.

SEGNALI DI CONTROLLO IN USCITA

I pins CA2 e CB2 sono linee bidirezionali e perciò possono essere usate indifferentemente in ingresso oppure in uscita.

TABELLA 2 — Sezione dei metodi di funzionamento del Timer 1 in base alla programmazione dei bits di ACR.		
ACR7	ACR6	MODO DI OPERAZIONE
0	0	Genera un solo interrupt al termine del conteggio del valore caricato nel counter di T1. Il pin PB7 è disabilitato.
0	1	Genera interrupts continui ad intervalli dipendenti dal valore caricato in T1. PB7 è disabilitato.
1	0	Genera un interrupt come nel modo 00 ed inoltre un impulso di uscita sul pin PB7 ogni volta che viene caricato T1.
1	1	Genera interrupts continui come nel modo 01 ed inoltre un'onda quadra su PB7.

```

LDA    ≠ 0   Port A in ingresso
STA    DDRA
LDA    =1    CA1 rivela la transizione basso-alto
STA    PCR
ATTESA LDA    IFR    Leggo il registro IFR
AND    ≠02   CA1 ha ricevuto la transizione?
BEQ    ATTESA
LDA    ORA    Leggo il dato e riporto a 0 il bit del registro IFR
    
```

TIMER 2

INDIRIZZO	REGISTRO	SCRITTURA	LETTURA
08	T2L-L	Si memorizza la parte bassa di T2.	Si legge T2C-L. Si azzerà il bit di interrupt di T2.
09	T2C-H	Si scrive la parte alta di T2 e si trasferisce T2L-L — T2C-L. Trigger del counter. Si azzerà il bit di interrupt di T2.	Si legge T2C-H.

TABELLA 3 — TIMER 1

INDIRIZZO	REGISTRO	SCRITTURA	LETTURA
04	T1L-L	Si scrive in T1L/L Memoria di T1. Trigger del timer 1.	Si legge T1C-L. Si azzerà il bit di interrupt di T1.
05	T1C-H	Trasferimenti: T1L-H - T1C-H. T1L-L - T1C-L.	Si legge T1C-H.
06	T1L-L	Si azzerà il bit di interrupt di T1. Si scrive T1L-L. Memoria di T1	Si legge T1L-L.
07	T1L-H	Si scrive T1L-H. Si azzerà il bit di interrupt di T1.	Si legge T1L-H.

Mediante la programmazione dei bit 1,2,3 per CA2 e 5,6,7 per CB2 nel registro PCR, è possibile ottenere 8 diversi modi di impiego di cui 4 come ingressi e 4 come uscite (vedere tabella 1).

Con CA2 o CB2 in ingresso (bit 3 o 7 a 0) la natura del segnale letto può essere impulsiva oppure di stato. Il valore 0 nei bit 2 o 6 rispettivamente per A o B, rivela un impulso; viceversa il valore 1 rivela uno stato logico. È possibile inoltre scegliere tra stato logico positivo o negativo (vedere tabella 1).

Con CA2 o CB2 in uscita (bit 3 o 7 uguale a 1) è possibile generare un impulso della durata desiderata o anche della durata di un singolo ciclo macchina.

Le linee CA2 o CB2 possono comunque essere messe manualmente nello stato voluto (vedere tabella 1).

INTERRUZIONI

Le interruzioni vengono controllate da due registri, il registro degli interrupts pervenuti (IFR) e quello di abilitazione degli interrupts stessi (IER). Vedere fig. 8

Il registro IFR è di sola lettura; in esso si va a leggere se è arrivata una interruzione e da dove: linee di controllo CA1, CA2, CB1, CB2, shift register (SR), timers T1 e T2.

Il bit 7 di IFR e l'OR logico di tutti gli altri bits e cioè è a 1 se uno qualsiasi degli altri bits è a 1.

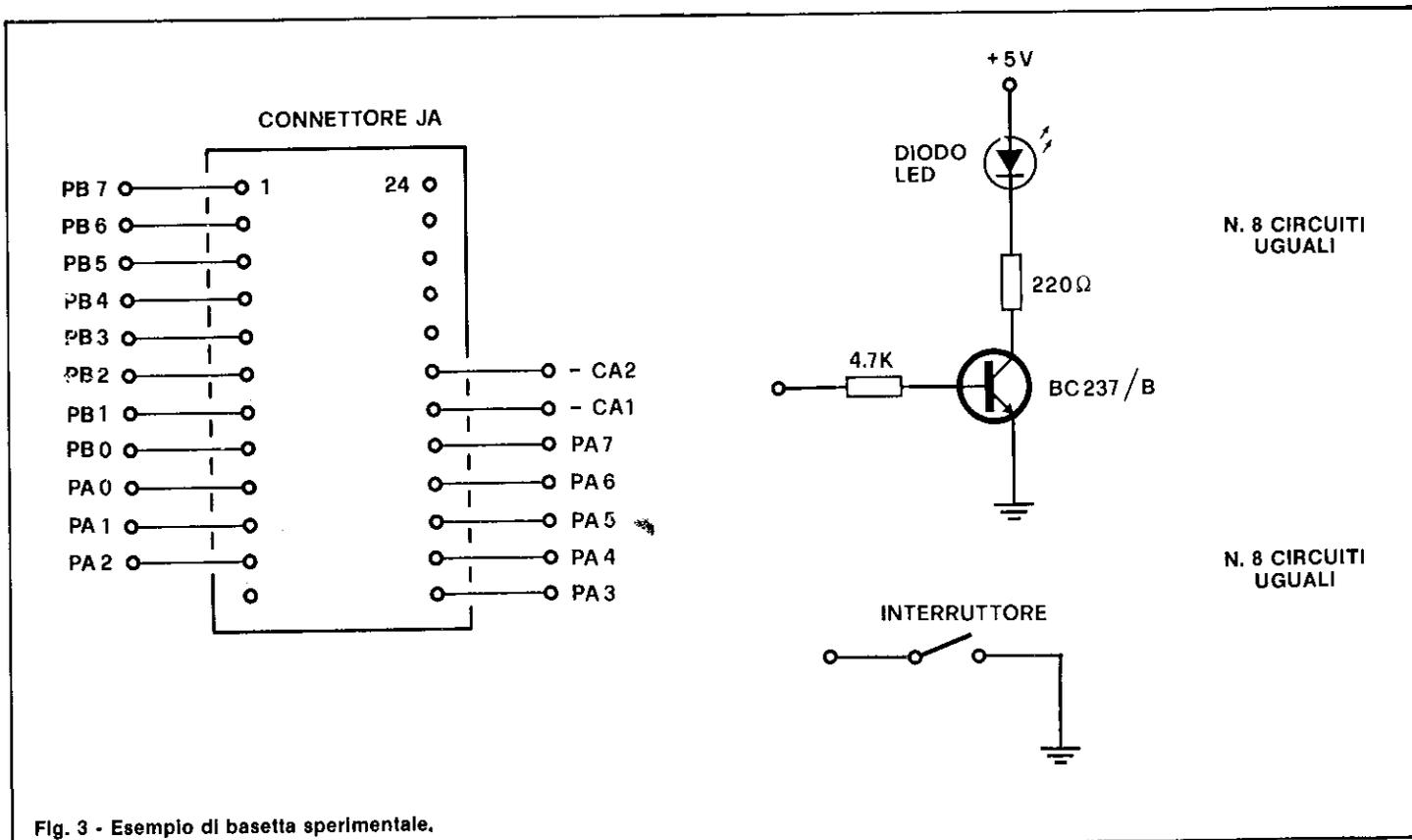


Fig. 3 - Esempio di basetta sperimentale.

Il registro IER è di scrittura e serve ad abilitare gli interrupts provenienti da qualsiasi sorgente.

Salvo il bit 7, ad ogni bit di IER, corrisponde un bit di IFR. Se un bit di IER è a 1, si abilita il passaggio dell'interrupt corrispondente in IFR.

Per mettere a 1 i bit di IER è necessario scrivere un 1 in posizione 7 e un 1 in tutti i bits che si vogliono abilitare; viceversa per mettere a 0 i bit di IER, bisogna scrivere uno 0 in posizione 7 e di nuovo un 1 in tutti i bit da mettere a 0.

Esempio: si vogliono abilitare gli interrupts relativi a CA1 e a CA2 e disabilitare tutti gli altri. Scriveremo perciò:

```
LDA  ≠$7C scriviamo 0111 1100 in
        IER disabilitando tutti
STA  IER      i bits dal 6 al 2.
LDA  ≠$83 scriviamo 1000 0011 in
        IER abilitando i bits
STA  IER      0 e 1
```

IDENTIFICAZIONE DEGLI INTERRUPTS

Nel caso in cui molti interrupts arrivino contemporaneamente, il programma dovrà leggere il contenuto di IFR per vedere quali interrupts sono pervenuti. L'ordine con cui vengono interrogati i singoli bits di IFR, determina la priorità secondo la quale vengono serviti gli interrupts.

Per esempio, se un interrupt su T1 ha la priorità più alta, il bit corrispondente deve essere interrogato per primo.

Il test del bit può essere fatto in più modi; i più usati sono l'AND≠Bit e lo SHIFT con test del Carry. Se per esempio vogliamo vedere se il bit 3 di IFR è = 1, possiamo scrivere:

```
LDA  IFR
AND  ≠$ 04
BEQ  NOUNO
```

Questa procedura può essere usata quando la priorità degli interrupt è casuale. Se invece usiamo lo SHIFT con test del Carry, otteniamo una priorità sequenziale (vedere colonna seguente) determinata dal senso dello SHIFT.

I TIMERS

La VIA 6522 contiene due timers che possono essere usati per realizzare delle temporizzazioni in ingresso, in uscita e all'interno di un programma. Quando essi sono usati in uscita, possono generare sia un singolo impulso che un treno di impulsi; se invece sono usati in ingresso, essi possono misurare la durata di un impulso o contare il numero degli impulsi ricevuti.

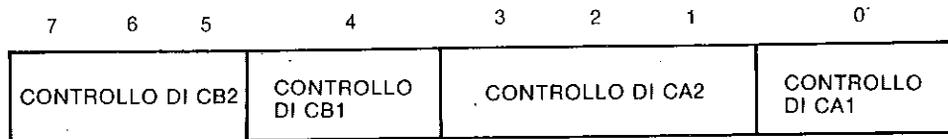


Fig. 4 - Registro di periferica (PCR).

Esempio:

```
CLC
LDA  IFR
LSR  ACC
BCS  UNO
CLC
LSR  ACC
BCS  DUE
.
.
.
Ecc.
```

oppure

```
CLC
LDA  IFR
ASL  ACC
BCS  UNO
CLC
ASL  DUE
.
.
.
Ecc.
```

Quando un timer genera o misura un solo impulso, funziona in modalità "impulsiva". In tale modo possono funziona-

re sia il Timer 1 che il Timer 2. Quando un timer genera o controlla un treno continuo di impulsi, si dice che funziona in modo "continuo". In tale modo può funzionare solo il Timer 1.

PARAGONE TIMER 1 - TIMER 2

Il Timer 2 può essere usato in input per contare gli impulsi che vengono applicati al pin PB6 della porta di ingresso B.

In tale caso bisogna preventivamente caricare il registro di conteggio con un valore conosciuto, ricordando che ad ogni impulso ricevuto, il contatore viene decrementato di 1.

Quando T2 è usato in uscita, esso può solo generare un ritardo all'interno del programma; può anche generare un singolo impulso solo se nel programma vengono inserite opportune istruzioni per mettere a 0 e a 1 un bit di un port. In ogni caso non può generare un treno di impulsi.

Il ritardo o la durata dell'impulso sono determinati dal valore che viene caricato nel counter, ricordando sempre che il conteggio avviene a decremento.

Il modo di funzionamento di T2 viene

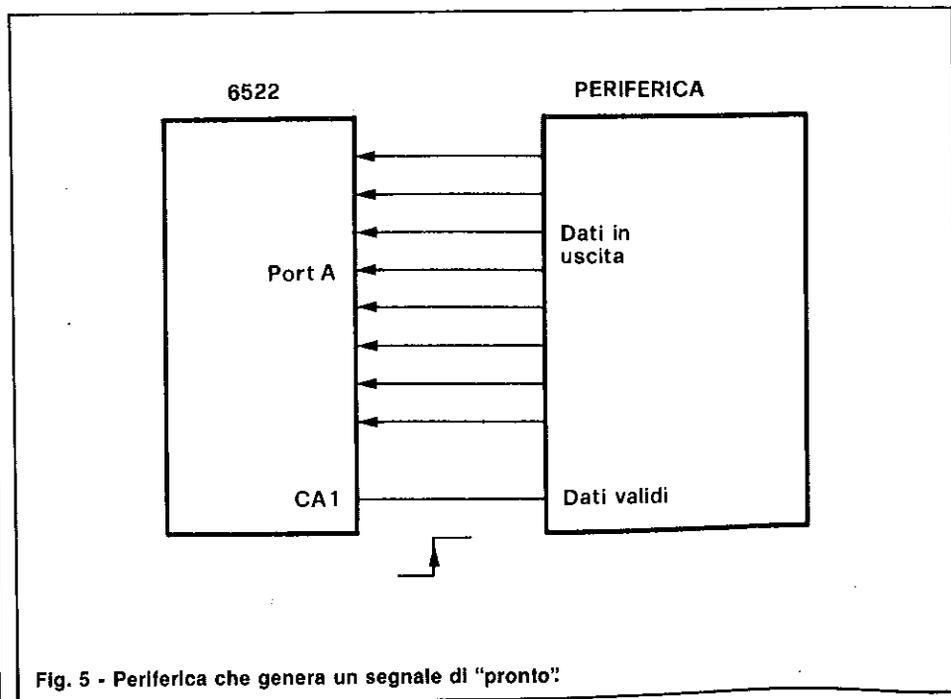


Fig. 5 - Periferica che genera un segnale di "pronto".

definito tramite il bit 5 del registro ACR. Uno 0 corrisponde al modo di uscita, mentre un 1 al conteggio di impulsi in ingresso.

Il Timer 1 è diverso dal Timer 2 e offre altre possibilità.

Esso ha 4 modi di funzionamento, riassunti in fig. 6, infatti può essere impiegato sia in modo "singolo impulso" che in modo "continuo". Oltre a ciò può essere abilitata o disabilitata una sua uscita diretta sul pin di uscita PB7 (vedere tabella 2).

Il Timer 1 nel modo "singolo impulso" funziona analogamente al T2 con la sola differenza che oltre alla funzione di ritardo del programma, è in grado di generare un singolo impulso direttamente sull'uscita PB7 (se questa è abilitata).

Nel modo continuo, T1, può generare su PB7 abilitato un'onda quadra in cui i tempi di ciclo sono programmabili caricando nel counter un valore adeguato. Si ricordi che il conteggio avviene sempre a decremento.

Inoltre è essenziale ricordare che ambedue i timers ad ogni fine conteggio, rendono un Interrupt che può essere ricevuto dalla CPU solo se il registro IER abilita tale segnale di interruzione. È implicito che T1 quando è usato in modo "continuo" genera interruzioni continue.

CARICAMENTO DEI COUNTER

Ogni Timer utilizza un counter da 16 bits, la cui parte bassa deve essere caricata per prima. Il successivo caricamento della parte alta del counter porta automaticamente a 0 il bit di interrupt corrispondente e fa partire il conteggio.

Al Timer 1 è anche associato un registro di memorizzazione da 16 bits. La presenza di tale registro permette di operare in modo "continuo". Infatti quando il contatore ha esaurito il conteggio, viene automaticamente ricaricato dal registro di memorizzazione e riparte per un nuovo conteggio.

Inoltre, il registro di memorizzazione, può essere letto e scritto in qualsiasi momento, senza influenzare il funzionamento del contatore principale; questa possibilità permette all'utente di costruire forme d'onda molto complesse.

Nella tabella 3 vengono riportati in dettaglio i registri coinvolti nel funzionamento dei due timers.

TEMPORIZZAZIONE REALE

Le forme d'onda generate dal Timer 1, vengono riportate in fig. 9. Si fa notare che la forma d'onda in uscita sul pin PB7, non è simmetrica, infatti il periodo basso

7	6
Ø - PB7 disabilitato 1 - PB7 abilitato	Ø - uscita "singolo impulso" 1 - uscita "continua"

Fig. 6 Significato dei bits di ACR per il funzionamento di T1.

TABELLA 4			
CONTROLLO DELLO SHIFT REGISTER			
ACR4	ACR3	ACR2	MODO DI FUNZIONAMENTO
Ø	Ø	Ø	Shift register disabilitato.
Ø	Ø	1	Shift register in ingresso sotto il controllo di T2.
Ø	1	Ø	Shift register in ingresso sotto il controllo della fase di macchina 02.
Ø	1	1	Shift register in ingresso sotto il controllo di clock esterno.
1	Ø	Ø	Shift register in uscita continua con velocità determinata da T2.
1	Ø	1	Shift register in uscita sotto il controllo di T2.
1	1	Ø	Shift register in uscita sotto il controllo della fase di macchina 02.
1	1	1	Shift register in uscita sotto il controllo di impulsi di clock esterni.

dura N+1,5 cicli di macchina (nell'Amico 2000 la 02 vale 1 microsecondo), mentre la parte alta N+2 cicli macchina.

Se per esempio vogliamo ottenere in uscita un'onda quadra avente una frequenza di 10 kHz dobbiamo caricare nel counter il valore HEX 30. Infatti 30 HEX = 48 Decimale e quindi 48+1,5+48+2 = 99,5 microsecondi che corrisponde al tempo di un ciclo completo. Essendo 10 kHz pari ad un tempo per ciclo di 100 microsecondi, abbiamo ottenuto una frequenza con ottima approssimazione.

IL REGISTRO A SCORRIMENTO (SHIFT REGISTER)

Il registro SR serve per la conversione seriale/parallela o viceversa. La velocità dello scorrimento può essere controllata

da tre diverse fonti: il Timer 2, la fase di macchina (02) oppure da un clock esterno.

La definizione di quale sorgente di temporizzazione debba essere impiegata, è affidata ai bits 2 e 3 del registro ACR, mentre il bit 4 stabilisce se lo SR è in ingresso o in uscita (vedere tabella 4).

Per usare lo SHIFT REGISTER in uscita si deve caricare il registro SR; con questa operazione inizia automaticamente lo scorrimento alla temporizzazione stabilita. Quando tutti e 8 i bits da trasmettere sono usciti, il flag di interrupt di SR (bit 2 del registro IFR) viene automaticamente posto a 1. Per proseguire occorre prima interrogare tale bit.

Nell'impiego come ricevitore seriale lo SHIFT REGISTER deve essere precaricato con un qualsiasi valore (per esempio 0) al fine di far partire lo scorrimento alla

7	6	5	4	3	2	1	0
CONTROLLO DI T1	CONTROLLO DI T2	CONTROLLO DELLO SHIFT REGISTER		PB LATCH	PA LATCH		

Nota - Latch = registro di memorizzazione

Fig. 7 - Registro di controllo (ACR).

7	6	5	4	3	2	1	0
IFR	T1	T2	CB1	CB2	SR	CA1	CA2
IER							

Fig. 8 - Registro di lettura e abilitazione degli interrupts.

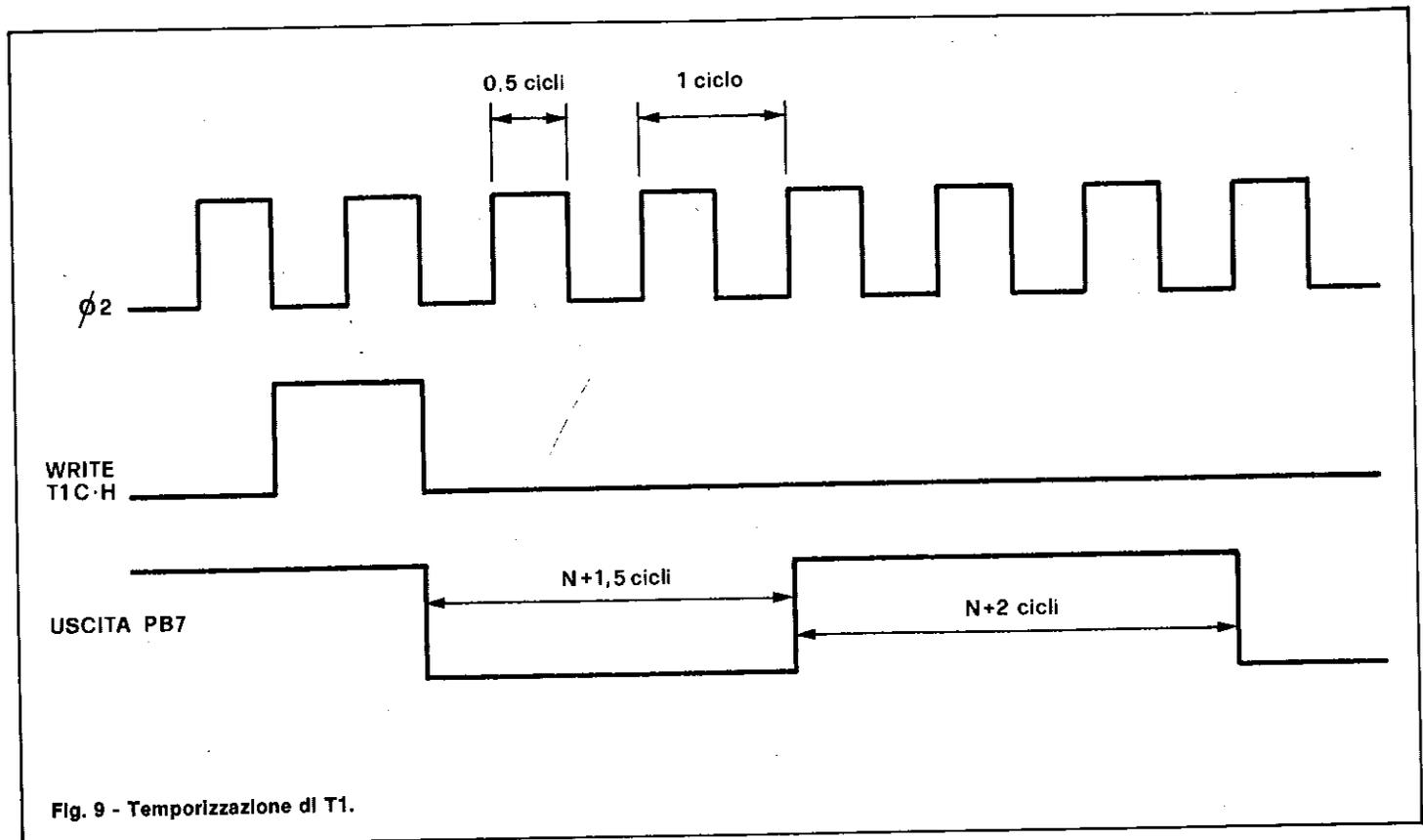


Fig. 9 - Temporizzazione di T1.

velocità determinata dalla fonte di temporizzazione scelta. Quando sono stati ricevuti 8 bits il flag di interrupt di SR viene messo a 1.

La disabilitazione dello SHIFT REGISTER si ottiene scrivendo nei bits 2,3,4 del registro ACR uno 0.

I dati sia in ingresso che in uscita corrono sulla linea CB2, mentre l'eventuale frequenza di clock esterno deve essere applicata a CB1.

Tenendo presente la tabella 4, analizziamo i vari modi di funzionamento dello SHIFT REGISTER:

MODO 000 - Shift register disabilitato. In queste condizioni la CPU può leggere o scrivere lo SR, ma l'operazione di scorrimento è disabilitata e il funzionamento di CB1 e CB2 è controllato dai bit relativi del registro PCR.

Il flag di interrupt nel registro IFR è sempre a 0.

MODO 001 - Shift register in ingresso sotto il controllo di T2.

In questa modalità la velocità di scorrimento è controllata dagli otto bits bassi di T2. Sul pin CB1 possono essere prelevati gli impulsi di shift per eventuale controllo di circuiti esterni. L'operazione di shift parte scrivendo o leggendo lo SR. Il primo bit ad entrare in CB2 deve essere il meno significativo e via via gli altri fino al più significativo. Dopo 8 bits, viene posto a 1 il flag di interrupt in IFR.

MODO 010 - Shift register in ingresso sotto il controllo della fase di macchina (02). In questo caso CB1 diventa un'uscita dalla quale si ricavano i segnali per sincronizzare la periferica in ingresso. Il Timer 2 diventa indipendente da SR. Lo scorrimento parte sempre leggendo o scrivendo lo SR.

MODO 011 - Shift register in ingresso sotto il controllo di un clock esterno. In questo modo CB1 diventa un ingresso in cui la periferica esterna fa entrare il clock di temporizzazione. Anche in questo modo dopo che sono stati accumulati 8 bits, il flag di interrupt in IFR viene messo a 1, però a differenza degli altri modi lo scorrimento continua. Leggendo o scrivendo lo SR si riporta a 0 il flag di interrupt ed il ciclo di lettura degli 8 bits può ricominciare.

MODO 100 - Shift register in uscita continua alla velocità determinata da T2. Questa modalità è molto simile alla 101. Lo SR non esaurisce l'operazione dopo 8 bits e il contenuto di SR continua ad uscire da CB2, in maniera circolare.

MODO 101 - Shift register in uscita sotto il controllo del Timer 2.

La velocità di uscita è controllata da T2 come nel caso precedente. Ad ogni lettura o scrittura di SR, vengono fatti uscire 8 bits da CB2, mentre su CB1 vengono generati 8 impulsi di sincronizzazione. Dopo 8 bits lo shift è disabilitato, il flag di

interrupt va a 1 e CB2 si posiziona come stabilito dal bit 5 del registro PCR. Se lo SR viene ricaricato prima che l'ultimo bit abbia terminato la sua temporizzazione, si può generare una uscita continua.

MODO 110 - Shift register in uscita sotto il controllo della fase di macchina (02). In questa condizione la velocità di scorrimento è condizionata dalla 02, mentre il Timer 2 rimane disponibile per l'utente.

MODO 111 - Shift register in uscita sotto il controllo di impulsi di clock esterno. In questa modalità lo scorrimento è controllato dagli impulsi applicati al pin CB1. Ogni 8 bit ricevuti, viene messo a 1 il flag di interrupt, ma non si disabilita lo scorrimento.

Ad ogni lettura o scrittura di SR, il bit di interrupt viene riportato a 0 e può ricominciare l'operazione di ingresso. ■

**nuovo
punto di vendita**

G.B.C.
italiana

**Via Matteotti, 18
S. VITTORE
OLONA**