Lezione 8 - Esempi di dispositivi di I/O: il TIMER

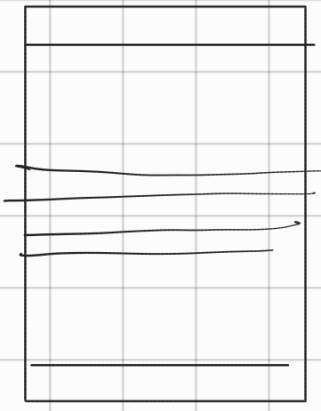
- Come riferimento consideriamo il timer Intel-8254
- A cosa può servire: un Timer
  - Generare ritardi programmabili in maniera accurata evento con ritardo ben definito
  - Real time clock riferimento temporale, tempo reale
  - Conteggio eventi che si verificano in un dato intervallo di tempo
  - Generatore di frequenza programmabile es. Genera forma d'onda onda quadra, o per audio
  - Generatore di onda quadra programmabile
  - Generatore di "impulso digitale" es. per interrupt
  - Moltiplicatore binario di frequenza moltiplicano in freq. una forma d'onda
  - Generatore di forma d'onda complessa
  - Controllo di motori

Timer circuito indispensabile

- È un buon esempio per mostrare la complessità interna di un'interfaccia di I/O
- Datasheet reperibili sul sito del corso
  https://web.archive.org/web/20150603122044/https://download.intel.com/design/archives/periphrl/docs/23124406.pdf

Nei chip, anche dei registri interni, dovrò indirizzare quei registri

Come collego Timer al processore? (1) Devo riuscire dal processore ad indirizzare i registri interni del TIMER (ne Ho 4)



Spazio di indirizzamento

Scelgo 0x40 .. 0x43

L'idea è far comparire quei registri nello spazio di indirizzamento

? Qual è l'hardware che mi permette di far apparire questi registri nel chip in modo tale che possa vederli dal PROCESSORE?

RICONOSCITORE DI INDIRIZZO!

Dato l'indirizzo su 64 bit, genero il segnale di chip select e i 2 bit di indirizz. interno del chip, come uscita del blocco

collegamento FATTO

- D7 - D0 collegati a Data della CPU, sono gli 8 bit meno significativi
- write e read (negati) collegati al R/W della CPU

=> Voglio ora generare indirizzo 0x40 ... 0x43

Come genero chip select? Porta NAND che riconosce in ingresso ognuna delle 16 (4x4) cifre esadecimale (Scarto 2 bit meno significativi)

uscita 1 quando presento indirizzo 00.. 40 (anche 41..43), non ho meno i bit meno significativi

→ uscita bassa, ovvero NOT

Per generare i 2 più significativi, li collego ad A1 A0 per generare i 4 indirizzi 40, 41, 42, 43, riconosce i 4 indirizzi

uso i 2 bit meno significativi, li collego ad A0 A1 per 40, 41, 42, 43

Riconoscitore: filtra indirizzo e mappa ognuno di questi chip nello spazio di memoria

spazio memoria: insieme degli indirizzi che un processore può generare, e insieme memoria fisica

PROCESS. RISC V

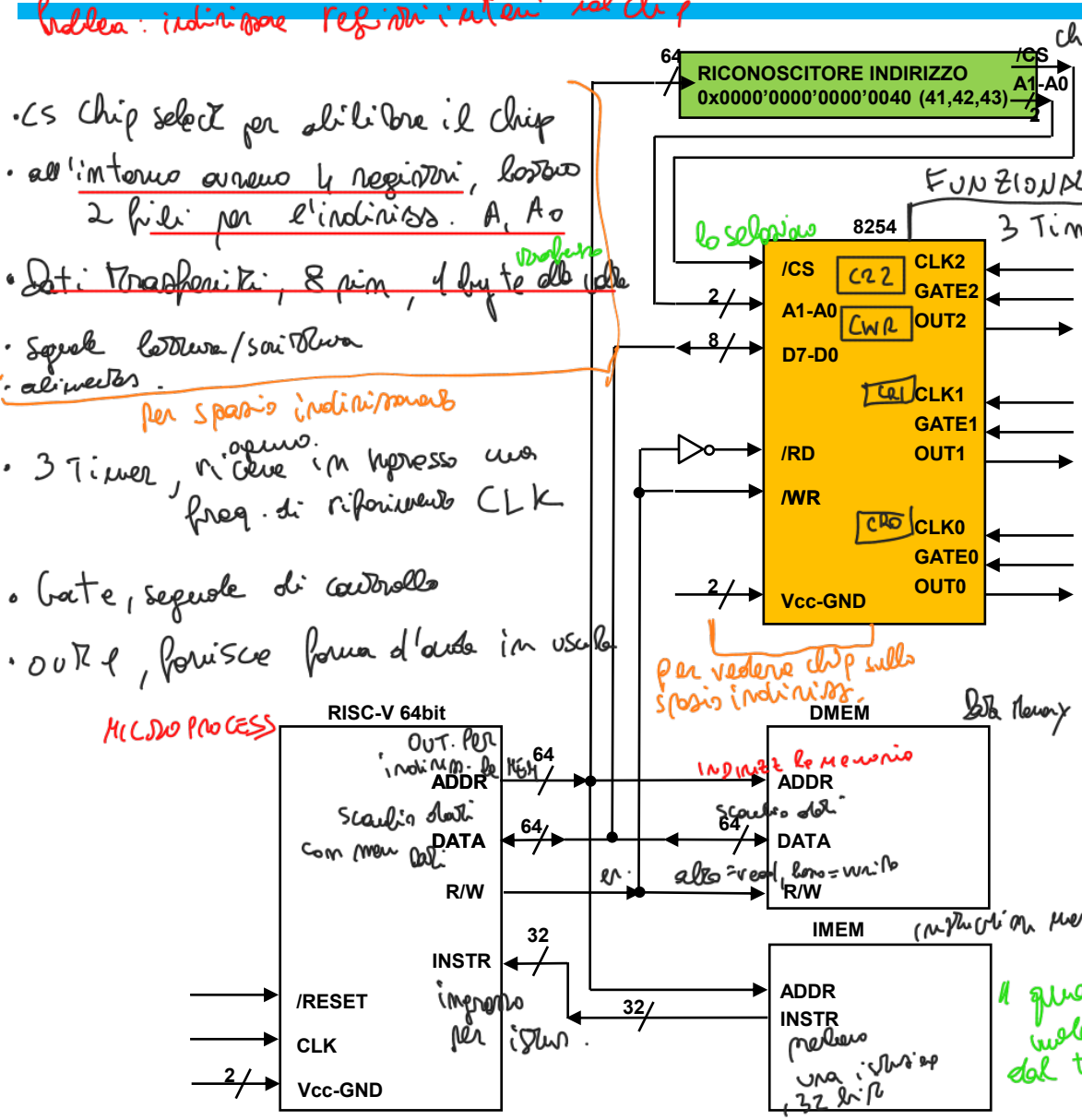
- insieme di fili (64) per gli indirizzi ADDRESS, 64 fili con cui indirizzo le memorie

con cui controllo Data Memory

- Ho collegato ad esso 2 memorie, Data Memory, instruction memory
- Segnale Read/Write
- Segnale di ingresso a 32 bit con cui prelevo le istruzioni
- Segnale clock, direttore., reset

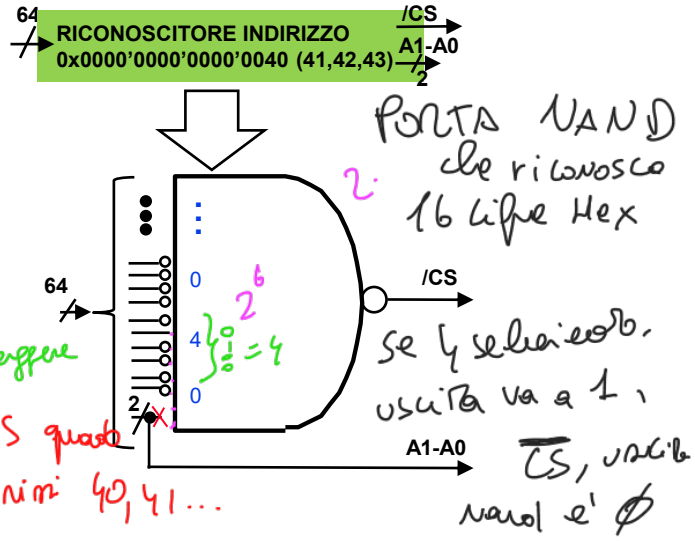
RICONOSCITORE INDIRIZZI MAPPA REGISTRI DISPOSITIVI INTERNO SPAZIO DI MEM

Timer Intel-8254: Visione elettrica del chip e di insieme



- Segnali di pilotaggio, DIGITALI
- **/CS**: Chip-Select per abilitarlo
- **A1-A0**: indirizzamento 2 bit
- **D7-D0**: scambio dati su 8 bit
- **/RD, /WR**: Read, Write
- **Vcc-GND**: alimentazione

- **CLKj**: clock del timer j
- **GATEj**: trigger hardware o congelamento conteggio
- **OUTj**: forma d'onda prodotta dal timer j



Timer Intel-8254: Visione Funzionale

- **Visione Funzionale del chip:**
  - **3 contatori indipendenti a 16-bit**
  - **6 modalità di conteggio (detti 'modi operativi')**

MODO	FUNZIONE	Condizione iniziale	Timing generato
0	Interrupt al termine del conteggio	OUT=X,GATE=1	OUT(N·T_C)=0,OUT(∞)=1
1	Programmable one shot	OUT=1,GATE=0	GATE=1→OUT(N·T_C)=0, OUT(∞)=1
2	Rate generator	OUT=X,GATE=1	OUT((N-1)·T_C)=1,OUT(1·T_C)=0,...
3	Square wave rate generator	OUT=X,GATE=1	OUT(~N/2·T_C)=1, OUT(~N/2·T_C)=0,...
4	Software triggered strobe	OUT=X,GATE=1	OUT(N·T_C)=1,OUT(1·T_C)=0,OUT(∞)=1
5	Hardware triggered strobe	OUT=1,GATE=0	GATE=1→OUT(N·T_C)=1,OUT(1·T_C)=0,OUT(∞)=1

GENERAZIONE DI UN GRADINO (verso uno)

GENERAZIONE DI ONDA PERIODICA

GENERAZIONE DI UN IMPULSO (verso zero)

Timer Intel-8254 (2)

FUNZIONAMENTO Registri: internamente ho contatori a 16 bit, però li vedo esternamente come registri da 8 bit) posso scrivere solo 8 bit alle volte

- **Visione logica del chip (tutti registri a 8 bit):**

Data Register

A1-A0	/RD	/WR			
00	1	0	0x40	CR0	Counter Register 0 Scrivendo in questo registro fisso la Time Constant 0
01	1	0	0x41	CR1	Counter Register 1 Scrivendo in questo registro fisso la Time Constant 1
10	1	0	0x42	CR2	Counter Register 2 Scrivendo in questo registro fisso la Time Constant 2

Definisco la N, 16 BIT

DR — Data register scambio dati

Selez. 2 contatore, leggo

Control Register

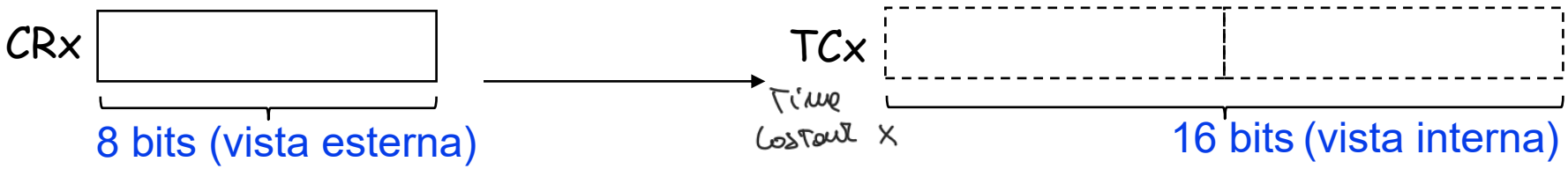
11	1	0	0x43	CWR	Control Word Register Scrivendo in questo registro fisso il funzionamento di un dato contatore oppure dò comandi particolari

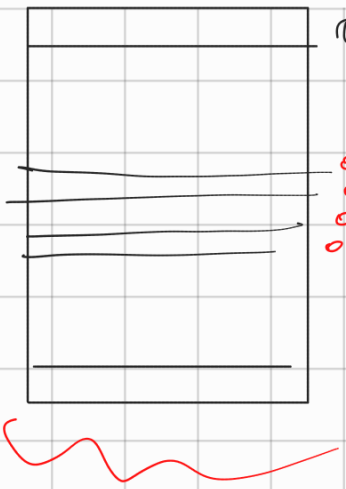
CR — control register per controllo del chip

Control register

Nota: la Time Constant è a 16 bit. CRj funziona come un "bus" a 8 bit verso un registro interno a 16 bit che contiene tale Time Constant

vista esterna a 8 BIT, internamente registri a 16 BIT





16 BIT
N

decido una N e voglio scriverla in un registro. N a 16 bit

N = costante di tempo

CWR = control word register

8 bit = 1 Byte

decido una N, voglio scriverla in uno dei registri, dove N è da 16 bit, ma i registri CRX sono da 8 bit, bisogni di un registro interno da 16 bit

0X43 CWR Registro di controllo da 8 bit qui imposto il modo di funzionamento

Data Register

- In un chip di I/O ho sempre registri DR con cui scambio dati, e insieme di registri che uso per effettuare il controllo del chip. (control registers, CR)

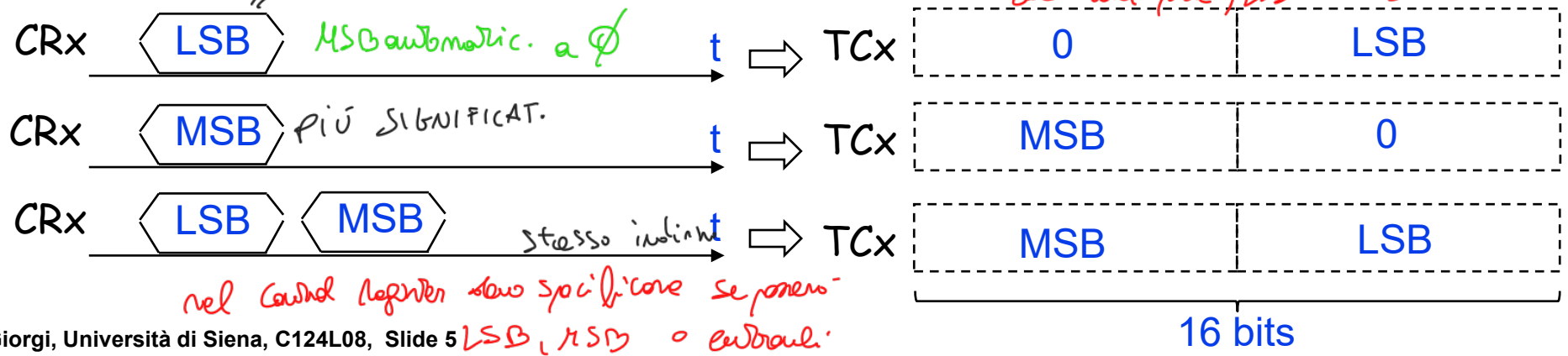
- Passaggio dati specificato attraverso la modalità di lettura/scrittura

Suddivido 16 bit in 2 gruppi da 8

- Nei DISPOSITIVI I/O avrò sempre Data Registers con cui scambio dati, e altri per controllarli (CONTROL REGISTERS)

Modalità di lettura/scrittura

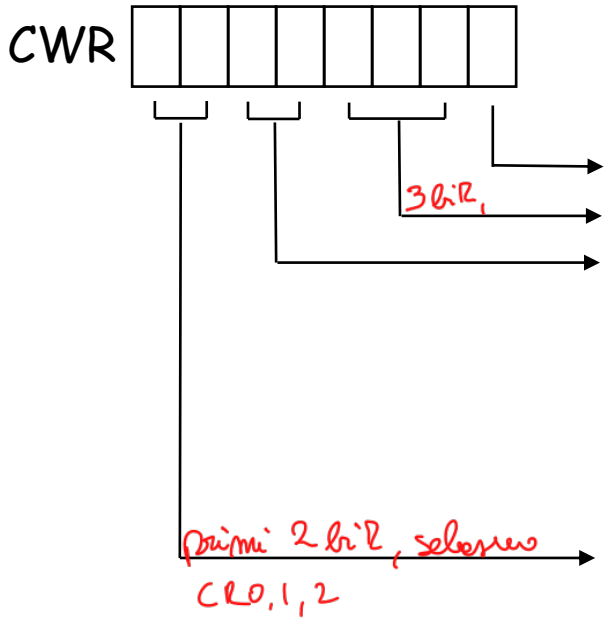
- Poiché nei dispositivi di I/O si tende ad utilizzare un numero limitato di indirizzi (nel nostro caso 4) si usa un registro per dare i comandi (CWR) e gli altri registri per leggere o scrivere dati (CR0,CR1,CR2)
- Dato che i contatori sono **internamente** a 16 bit e posso scrivere solo 8 bit (1 byte) per volta: in quale ordine vengono inviati tali byte?
  - LSB+MSB: invio prima il byte meno significativo (Least Significant Byte o LSB) e poi su quello più significativo (Most Significant Byte o MSB)
- posso risparmiare un invio se uno dei due byte è zero
  - LSB: invio solo il byte meno significativo (es. 0x00A4 → invio solo 0xA4)
  - MSB: invio solo il byte più significativo (es. 0x B500 → invio solo 0xB5)



Programmazione del timer

CON GATE ABILITO CONTEGGIO

1) Scrivere in CWR per selezionare uno dei contatori, modo, ordine LSB/MSB (nell'indirizzo 43)
2) Scrivere in CRj la costante di tempo (poi scrivo la costante di Tempo)
3) Abilitare il conteggio con un segnale alto sul pin GATE (così l'uscita OUT varia)
4) L'uscita OUT produrrà un segnale di uscita alla fine del (oppure durante il) conteggio a seconda del modo prescelto



Cosa scrivo in CWR

Es. CWR ← 10110110 (2, 3)
significa che il timer #2 deve effettuare un conteggio binario in modalità #3, le scritture/letture seguono lo schema LSB+MSB

Lettura del conteggio (1)

- **Ci sono tre maniere per leggere il valore di un contatore**
  1) **lettura semplice**
  2) **Counter Latch Command**
  3) **Read-Back Command**

1) La lettura semplice consiste nel leggere il valore di CRj previo bloccaggio del conteggio tramite il segnale GATEj

- **Se non si blocca il conteggio il valore letto può essere inconsistente**

2) Counter-Latch Command

CWR	c	c	0	0	-	-	-	-

cc: contatore 0, 1, 2 — 00: invio comando di memorizzazione conteggio

- (cc=0, 1 o 2 indica il contatore, '-' indica non specificato)
- Lettura di 1 o 2 byte da CRcc (a seconda dello schema di lettura/scrittura precedentemente impostato)

3) Read-Back Command

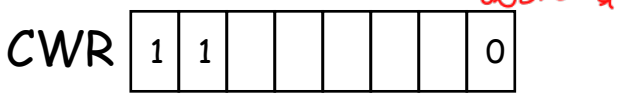
CWR	1	1						0

- V. slide successiva

Lettura del conteggio (2)

- **Read-Back Command**

  1) Si realizza in questo modo: scrittura in CWR delle seguenti info



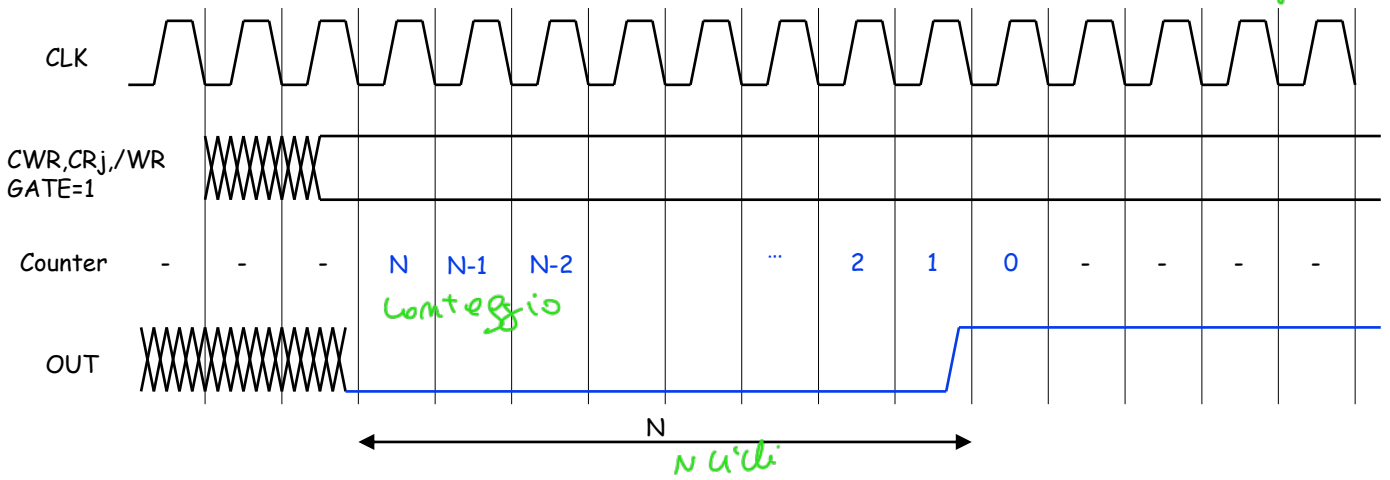
CNT0 – se 1 seleziona il contatore 0
CNT1 – se 1 seleziona il contatore 2
/STATUS – se 0 nel latch di uscita va la parola di comando
ultima data a quel contatore
/COUNT – se 0 nel latch di uscita va il conteggio

2) Successiva lettura di CRx

- **Vantaggi:**
  - **Consente di leggere i conteggi di più contatori simultaneamente (caso di bit /COUNT attivo)**
  - **Consente di leggere come sono stati programmati i 3 contatori (caso di bit /STATUS attivo)**
    - **In particolare, i 6 bit meno significativi danno i valori precedentemente scritti in CWR per quel contatore**

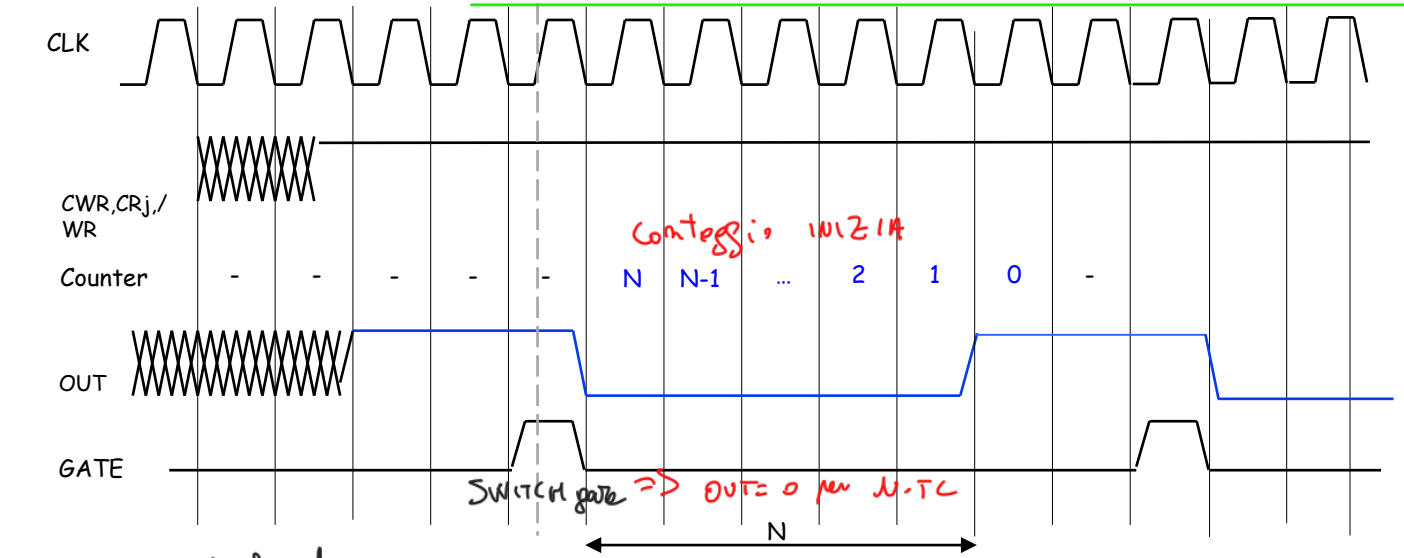
Modi per generare un gradino dopo il conteggio

- **Modo 0: conteggio abilitato dalla scrittura di CR**



Nota: usato ad esempio per generare un interrupt alla fine di un conteggio

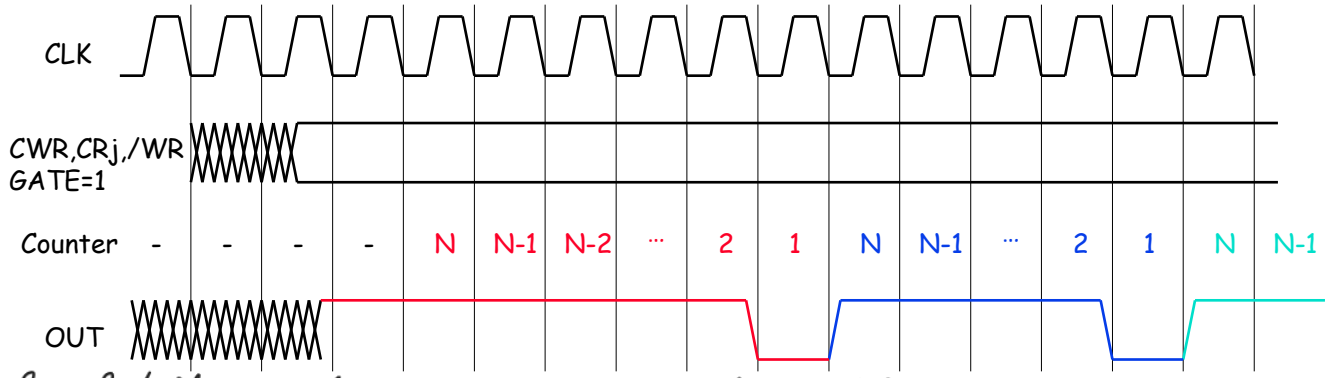
- **Modo 1: one shot (conteggio abilitato dal segnale hardware GATE)**



Nota: mentre OUT è agganciato ai fronti in discesa del clock (CLK), il segnale GATE viene viceversa campionato sui fronti in salita del clock. (per approfondimenti vedere manuale del chip 8254)

Modi per generare un'onda periodica

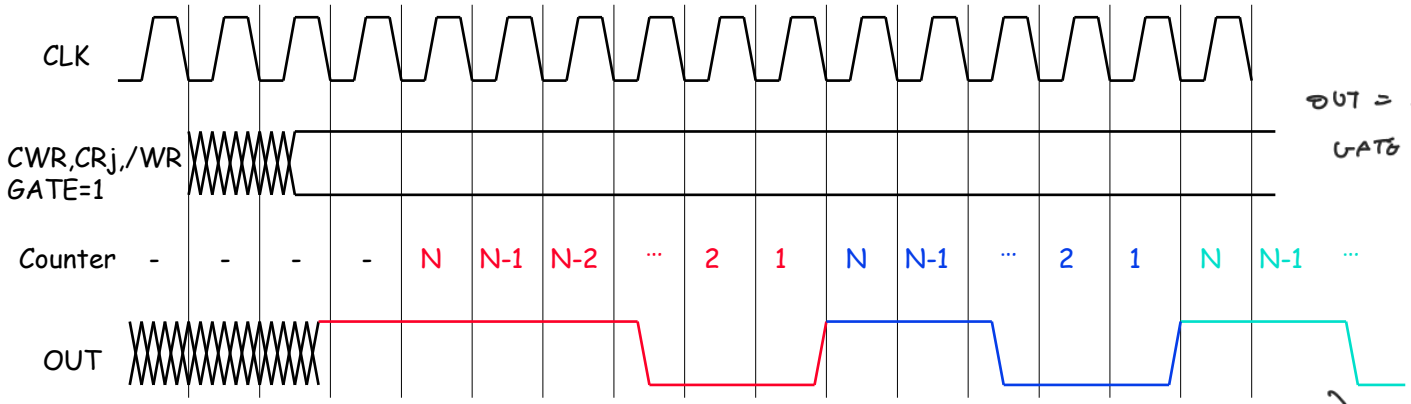
- **Modo 2: rate generator (divisore di frequenza)**
  → genera un onda quadra con duty-cycle pari a (N-1)/N



Nota1: Se f è la frequenza del segnale di clock (CLK), la forma d'onda d'uscita (OUT) avrà frequenza f/N

Nota2: L'uscita è inizialmente alta e diventa bassa, per la durata di un ciclo, quando il contatore raggiunge il valore 1

- **Modo 3: square wave mode (generatore d'onda quadra)**

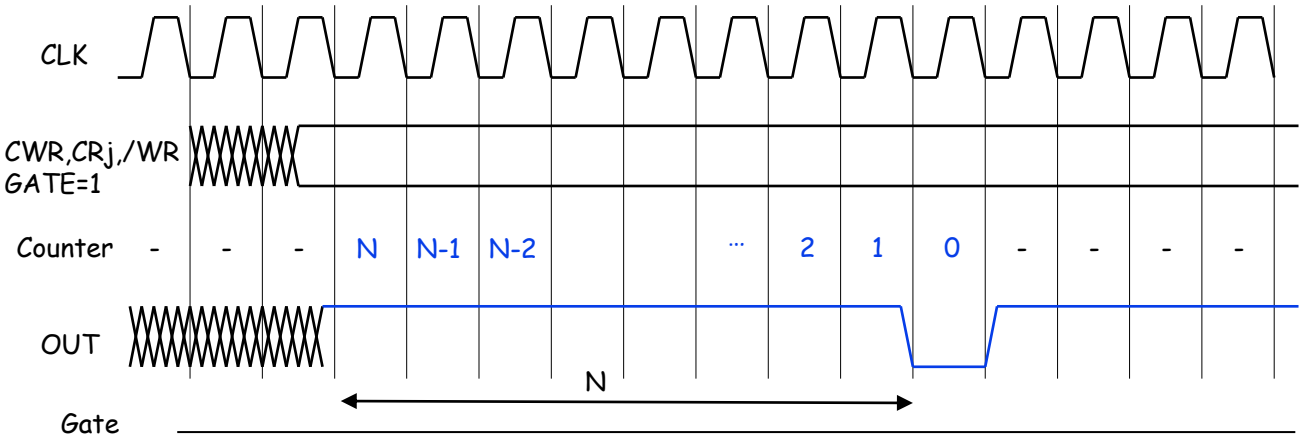


- **OUT rimane alto nei periodi N, N-1, … , (N+1)/2**
- **OUT rimane basso nei periodi N/2, (N-1)/2, … , 2, 1**

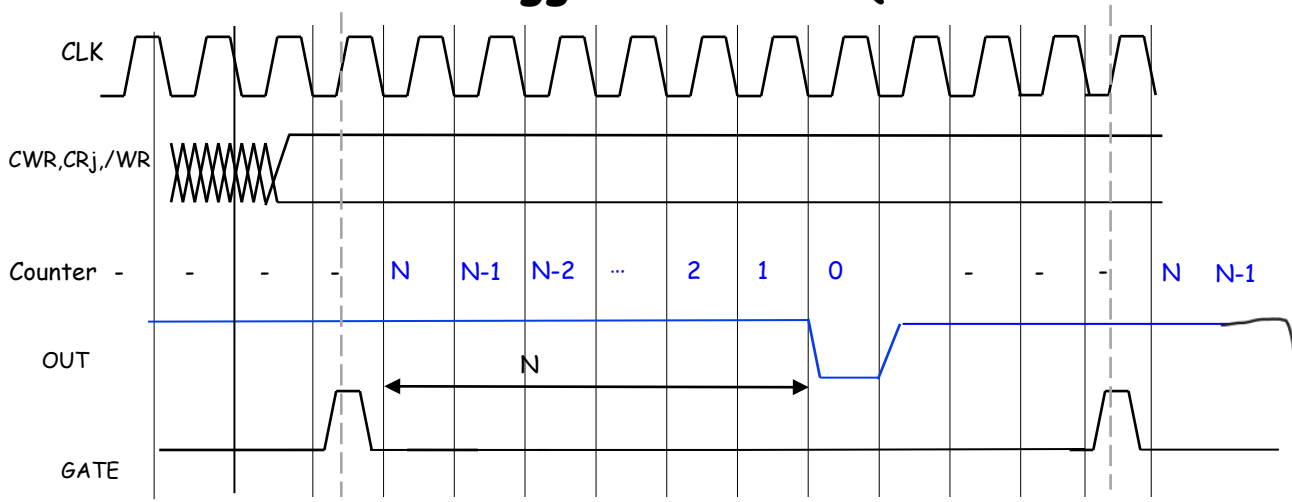
→ Il duty cycle è ½ se N è pari, altrimenti è ((N+1)/2)/N

Modi per generare un impulso a fine conteggio

- **Modo 4: software triggered strobe (abilitato alla scrittura di CR)**



- **Modo 5: Hardware triggered strobe (abilitato ad hardware da GATE)**



Nota: mentre OUT è agganciato ai fronti in discesa del clock (CLK), il segnale GATE viene viceversa campionato sui fronti in salita del clock. (per approfondimenti vedere manuale del chip 8254)

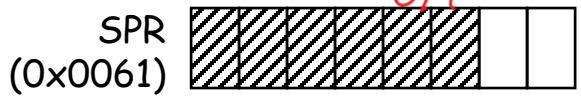
Riepilogo risorse 8254 nei Personal Computer

Chip name	Register name		I/O port	IRQ
Intel 8254	CR0	Counter Register 0	0x0040	IRQ0
Intel 8254	CR1	Counter Register 0	0x0041	-
Intel 8254	CR2	Counter Register 0	0x0042	-
Intel 8254	CWR	Control Word Register	0x0043	-

Glue logic	SPR	Speaker Register	0x0061	

- **CLK0=CLK1=CLK2 = 1.19318MHZ**
- **TIMER-0: timer di sistema**
  - **GATE0 sempre attivo**
  - **OUT0 genera IRQ0**
- **TIMER-1: timer per il rinfresco della memoria**
  - **GATE1 sempre attivo**
  - **OUT1 genera impulsi con un periodo di 15ms per attivare il rinfresco della memoria**
- **TIMER-2: generatore di beep (v. slide successiva)**
  - **GATE2 controllabile tramite bit0 di SPR,Speaker Register (porta di I/O 0x0061)**
  - **OUT2 in AND col bit1 di SPR, l'uscita va poi sul beeper**

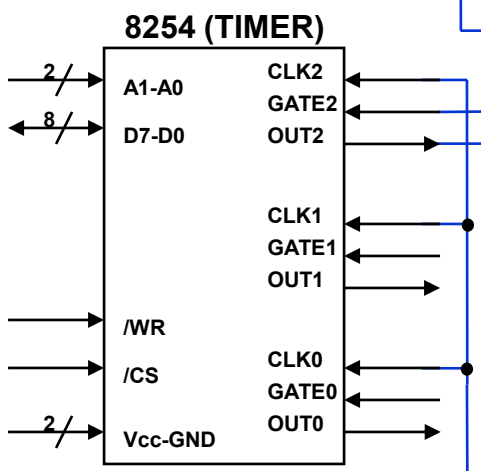
Speaker Register e generazione di suoni sullo speaker

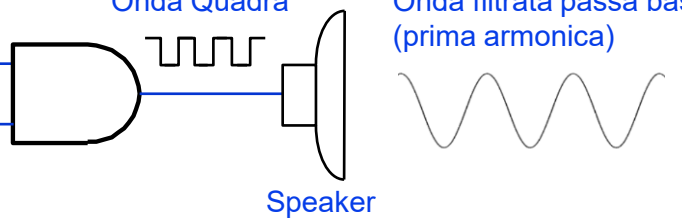


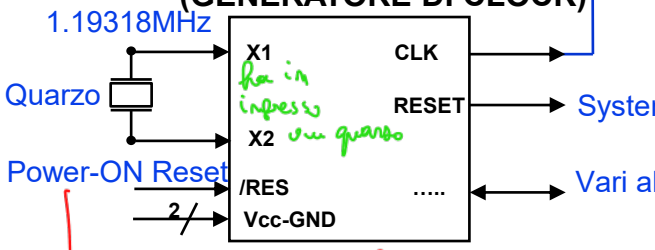
- **Realizzato in "logica sparsa"**

Abilita Gate2 (Abilitazione generale del timer2)

Controlla lo speaker







Esempio di programmazione del Timer 8254 in C

- **Generare una interruzione dopo Δt=50ms**
  - **Fc=1.193$\overline{18}$ MHz** (Tc=1/1.193$\overline{18}$·10^6 ≈ 838ns)
  - **N=Fc·Δt=1.193$\overline{18}$·10^6·50·10^-3 → N ≈ 59659**

```
int init_8254 (void)
{
   int cr_l, cr_h;

   outp(0x0043, 0x30);  /* contatore 0, costante di tempo a 16 bit,
                           modo 0, conteggio binario */

   cr_l = 59659 & 0x000000FF;
   cr_h = (59659 >> 8) & 0x000000FF;
   outp(0x0040, cr_l);
   outp(0x0040, cr_h);
}
```

→V. SLIDE 7