8253

(Programmable interval timer)

M. Rebaudengo - M. Sonza Reorda - E. Sanchez

Politecnico di Torino
Dip. di Automatica e Informatica

Introduction

- The 8253 is a device that implements timing and counting tasks
- Initially implemented in a LSI DIP chip counting 24 pins
- It provides three programmable independent 16-bit counters
- It was replaced by the 8254 that implements the same functionalities plus some additional features.

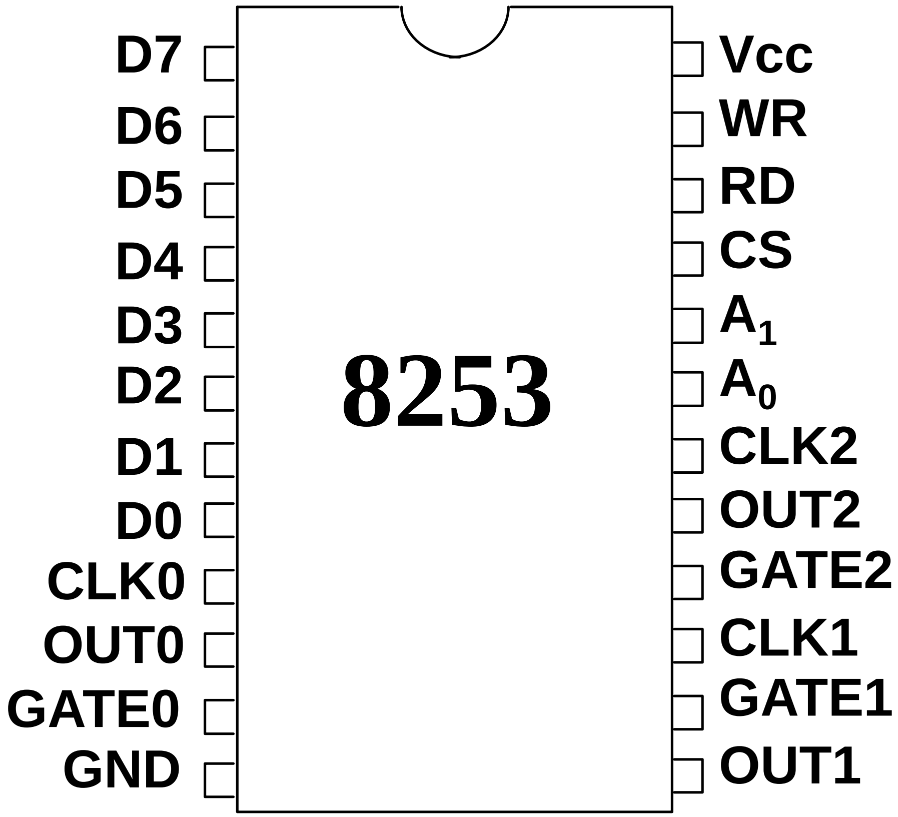
General usage

- Generation of accurate and programmable time delays
- Signal generator: square and digital one shot waves
- Event counter
- Interval counter
- Frequency divisor.

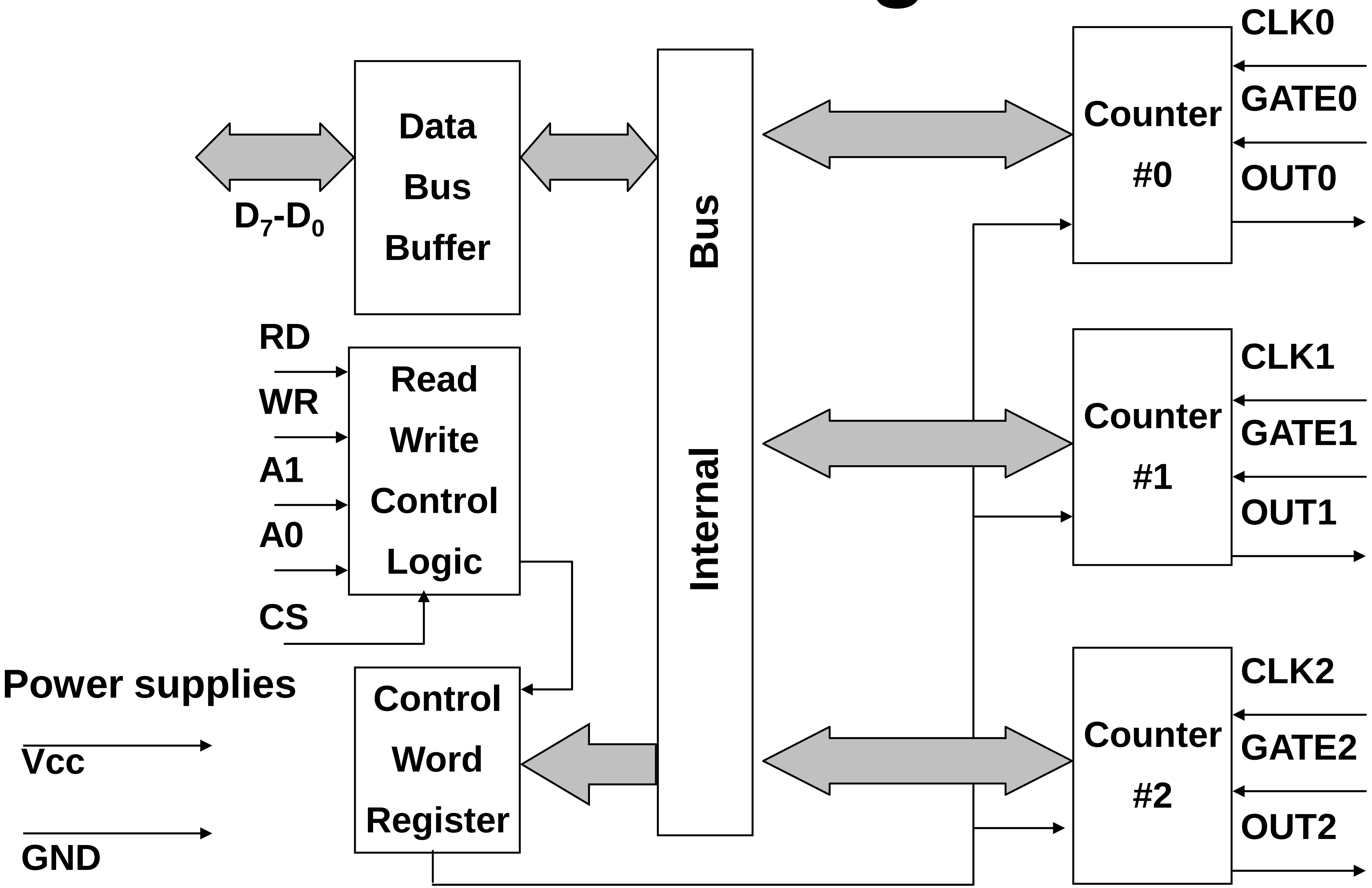
A typical example: pulse generator

- The 8253 is initially configured by the programmer
- One of the 8253 timers is loaded with a specific value
- The 8253 performs a decremental counting and when it reaches 0, an interruption is triggered.

The chip


D7
D6
D5
D4
D3
D2
D1
D0
CLK0
OUT0
GATE0
GND
8253
Vcc
WR
RD
CS
A_1
A_0
CLK2
OUT2
GATE2
CLK1
GATE1
OUT1

8253 block diagram



Data Bus Buffer

The Data Bus Buffer is an 8-bit bidirectional buffer that can assume the 3-state value, and it is the 8253 interface to the system Data Bus.

Device pins D_{7-0} are connected to the system Data Bus.

The Data Bus is used to:

- Program the 8253 by setting the operation mode of every timer/counter
- Upload the counters initial values
- Read the counters values

8253 counters

The 8253 includes 3 independent 16-bit counters

Any counter can be:

- externally loaded with a given value
- configured as a decremental counter using its own CLK signal
- used as a pure binary or a packed Binary Code Decimal (BCD) counter
- programmed in an independent way using a control register that choses one out of six operational modes.
- externally read.

8253 Interface

The 8253 can be seen as a set of 4 consecutive I/O ports, where:

- 3 correspond to the counters
- 1 is the device control register.

The 8253 is enabled by using the $\overline{CS}$.

The 8253 ports can be selected by using the signals A_0 and A_1.

Accessing to the device ports

A_0 and A_1 controls the access to the three 8253 ports as well as the Control Word Register (*CW*).

A_1	***A_0***	***Register***
0	**0**	**Counter #0**
0	**1**	**Counter #1**
1	**0**	**Counter #2**
1	**1**	**Control Register**

8253 Input/Output operations

A1	A0	RD	WR	CS	
0	0	0	1	0	Read counter #0
0	1	0	1	0	Read counter #1
1	0	0	1	0	Read counter #2
0	0	1	0	0	Load counter #0
0	1	1	0	0	Load counter #1
1	0	1	0	0	Load counter #2
1	1	1	0	0	Write Control register
X	X	X	X	1	Data Bus in 3-State
X	X	1	1	0	No-operation
1	1	0	1	0	Illegal condition
X	X	0	0	X	Illegal condition

Programming the 8253

It is possible to configure the 8253 via software.

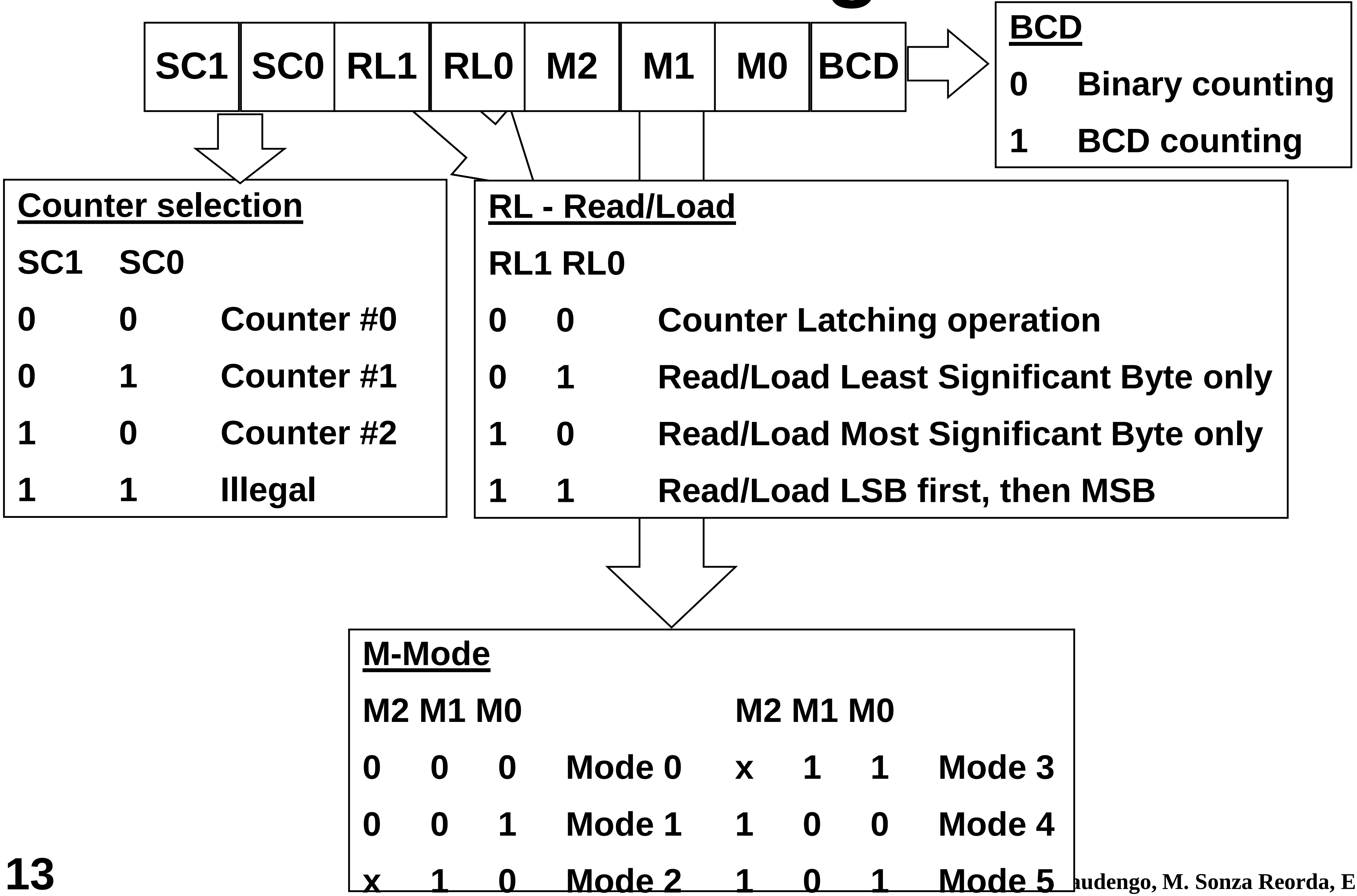
The CPU writes a set of control words to initialize the 8253 counters.

The functioning mode, the content and the counter output are undefined before the initialization.

For any counter, the control register determines:

- The functioning mode
- The counter loading mode
- The counter counting mode (binary or BCD).

Control register



SC1	SC0	RL1	RL0	M2	M1	M0	BCD

BCD

0	Binary counting
1	BCD counting

Counter selection

SC1	SC0	
0	0	Counter #0
0	1	Counter #1
1	0	Counter #2
1	1	Illegal

RL - Read/Load

RL1	RL0	
0	0	Counter Latching operation
0	1	Read/Load Least Significant Byte only
1	0	Read/Load Most Significant Byte only
1	1	Read/Load LSB first, then MSB

M-Mode

M2	M1	M0		M2	M1	M0	
0	0	0	Mode 0	x	1	1	Mode 3
0	0	1	Mode 1	1	0	0	Mode 4
x	1	0	Mode 2	1	0	1	Mode 5

Exercise

Write the sequence of instructions that initialize the three 8253 counters starting at 40H address in the following modes:

- **Counter 0: Binary counting, mode 0, initial value 1234H**
- **Counter 1: BCD counting, mode 2, initial value 100H**
- **Counter 2: Binary counting, mode 4, initial value 1FFFH.**

Solution

Control register address 43H.

The control words for the 3 8253 counters are:

```
CW0  EQU 00110000  ;  30h
CW1  EQU 01100101  ;  65h
CW2  EQU 10111000  ;  B8h

MOV  AL, CW0
OUT  43h, AL  ; programming the Counter #0
MOV  AL, CW1
OUT  43h, AL  ; programming the Counter #1
MOV  AL, CW2
OUT  43h, AL  ; programming the Counter #2
```

Solution
(cont)

```
MOV          AL, 34H
OUT          40h, AL ; Loading LSB counter 0
MOV          AL, 12h
OUT          40h, AL ; Loading MSB counter 0
MOV          AL, 01h
OUT          41h, AL ; Loading MSB counter 1
MOV          AL, 0FFh
OUT          42h, AL ; Loading LSB counter 2
MOV          AL, 1Fh
OUT          42h, AL ; Loading MSB counter 2
```

Reading the counters

In some applications, it is necessary to read the counter value while the counting process is in progress.

There are two possible reading modes:

- Disabling the counters
- On the fly.

Reading disabled counters

It is possible to normally access in reading mode to the 8253 counters by setting the address signals A_1-A_0.

In order to obtain a stable value, it is required to disable the involved counter via software, controlling the external GATE signal, or stopping the clock.

The counter value is provided in the following way:

- The first read operation loads in `AL` the least significant byte
- The second read operation loads in `AL` the most significant byte

Reading disabled counters (II)

It is absolutely required to perform the full reading procedure before trying to do new operations.

In particular, in the case the control word indicates that two different words should be read, two reading accesses should be performed before trying to access again to the same counter.

On the fly reading

The *on the fly* technique allows the user to do not stop the counter process while performing a reading access on the counter value.

In order to read a counter on the fly tis is required to:

- **Write the control register with a special command that makes the 8253 to save the current counter value in an internal register, saving a counter stable value**
- **Perform a complete reading access to the involved counter to read the previously saved counter.**

Control word command for on the fly reading

SC1	SC0	0	0	X	X	X	X

SC1	*SC0*	*Meaning*
0	0	Counter #0
0	1	Counter #1
1	0	Counter #2
1	1	Illegal

Exercise

Write a sequence of instructions for reading the 8253 counter 2 on the fly.

The obtained value should save in AX register.

Assume that the 8253 registers are allocating starting at address 40H.

```
MOV       AL, 80h
OUT       43h, AL
IN        AL, 42H
MOV       AH, AL
IN        AL, 42H
XCHG      AL, AH
```

8254

The 8254 (and the 82C54) replaced the 8253.

The 8254:

- **is compatible pin-by-pin with the 8253**
- **provides some additional functions**
- **is not 100% compatible at software level.**

Functioning modes

Any one of the three 8253 counters can be configured in one if the following operating modes:

- **mode 0: Interrupt on terminal count**
- **mode 1: Programmable One-shot**
- **mode 2: Rate generator**
- **mode 3: Square wave generator**
- **mode 4: S/W Triggered Strobe**
- **mode 5: H/W Triggered Strobe.**

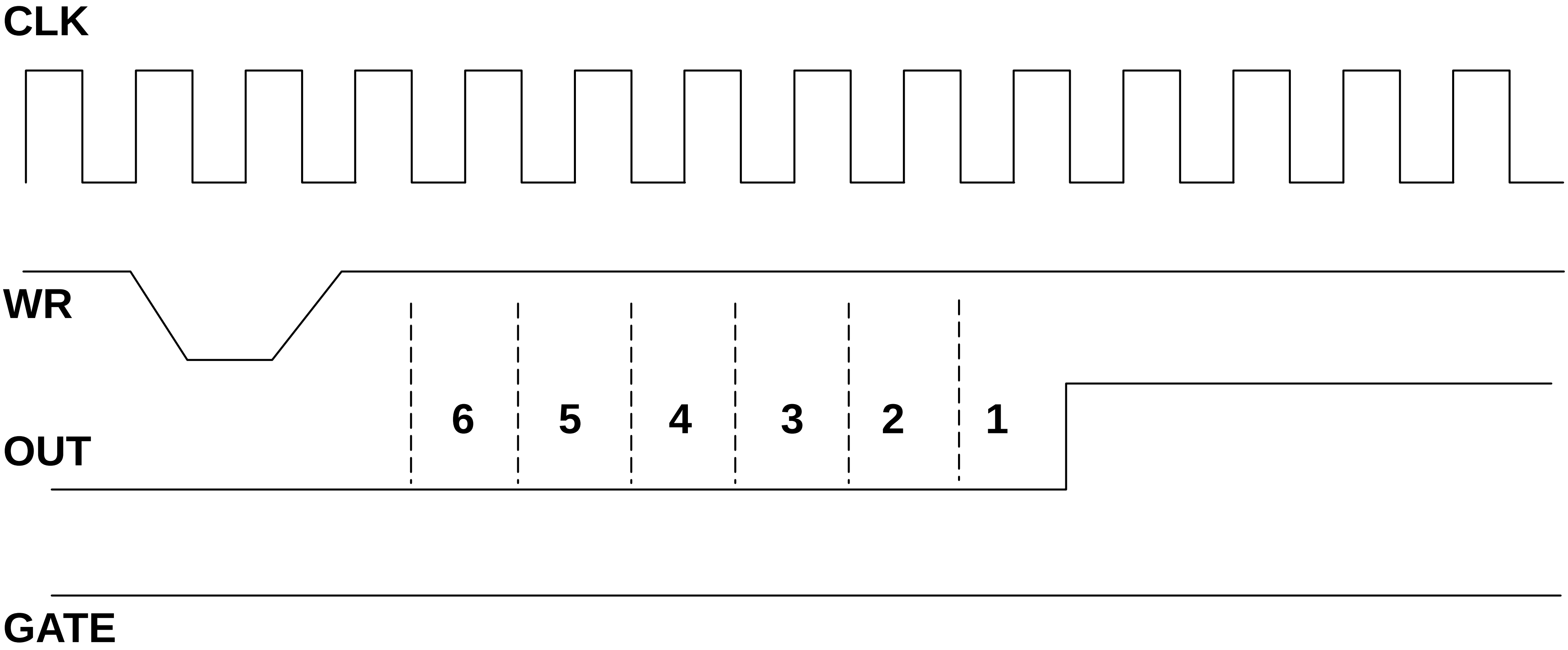
Mode 0
(*Interrupt on terminal count*)

The counter will count as soon as the counter is loaded.

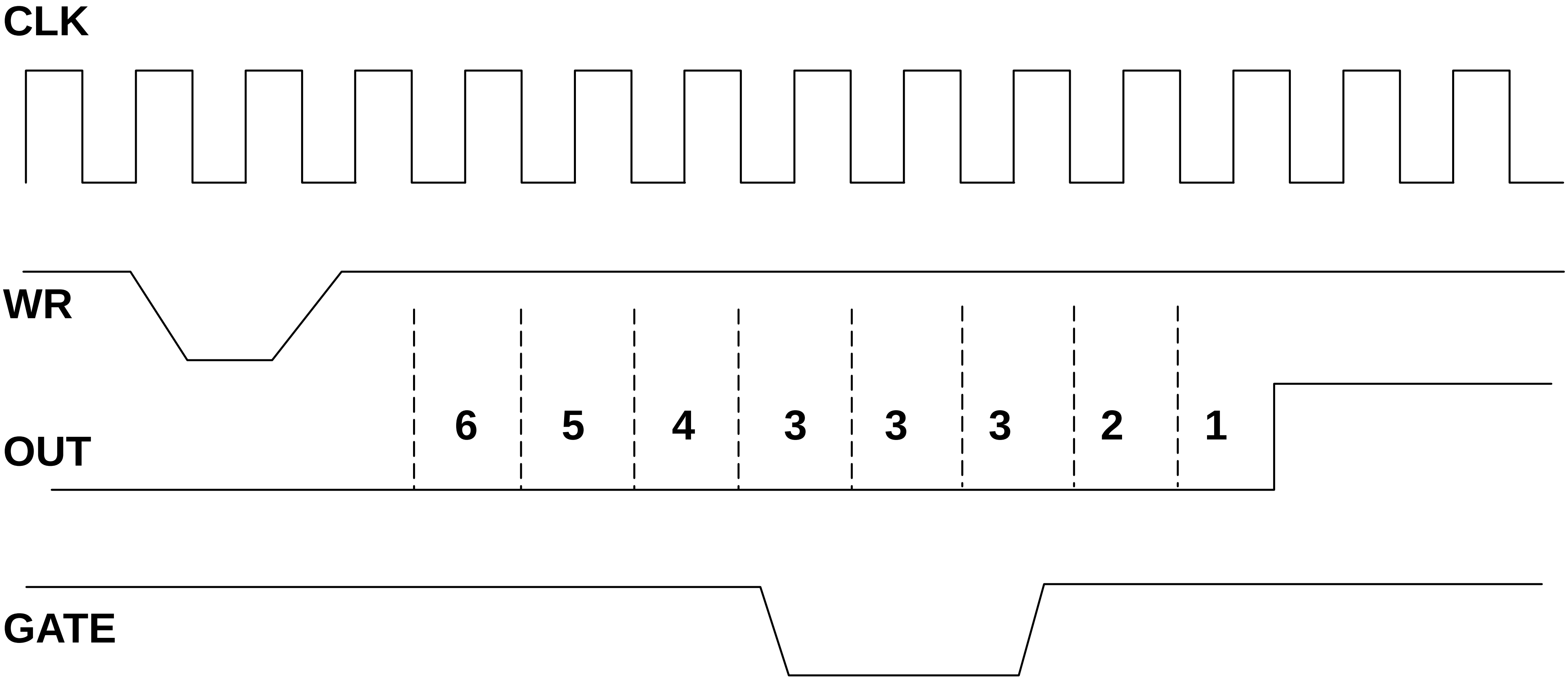
During the counting OUT is low; when the terminal count is reached it will go high until a new loading operation is performed.

GATE enables to count when it is high, and disables it when low.

Mode 0: example



Mode 0: Example (II)



Exercise

Configure the 8253 counter 1 in order to generate a 5ms delay. Assume that the system clock is 1MHz.

Solution

The clock period is 1us. Thus, for getting a 5ms delay it is necessary to count 5000 clock cycles.

```
MOV AL, 01100001B   ; counter 1, mode 0, BCD

OUT 43h, AL

MOV AL, 50h         ; most significant byte

OUT 41h, AL
```

Mode 1
(Programmable *One-Shot*)

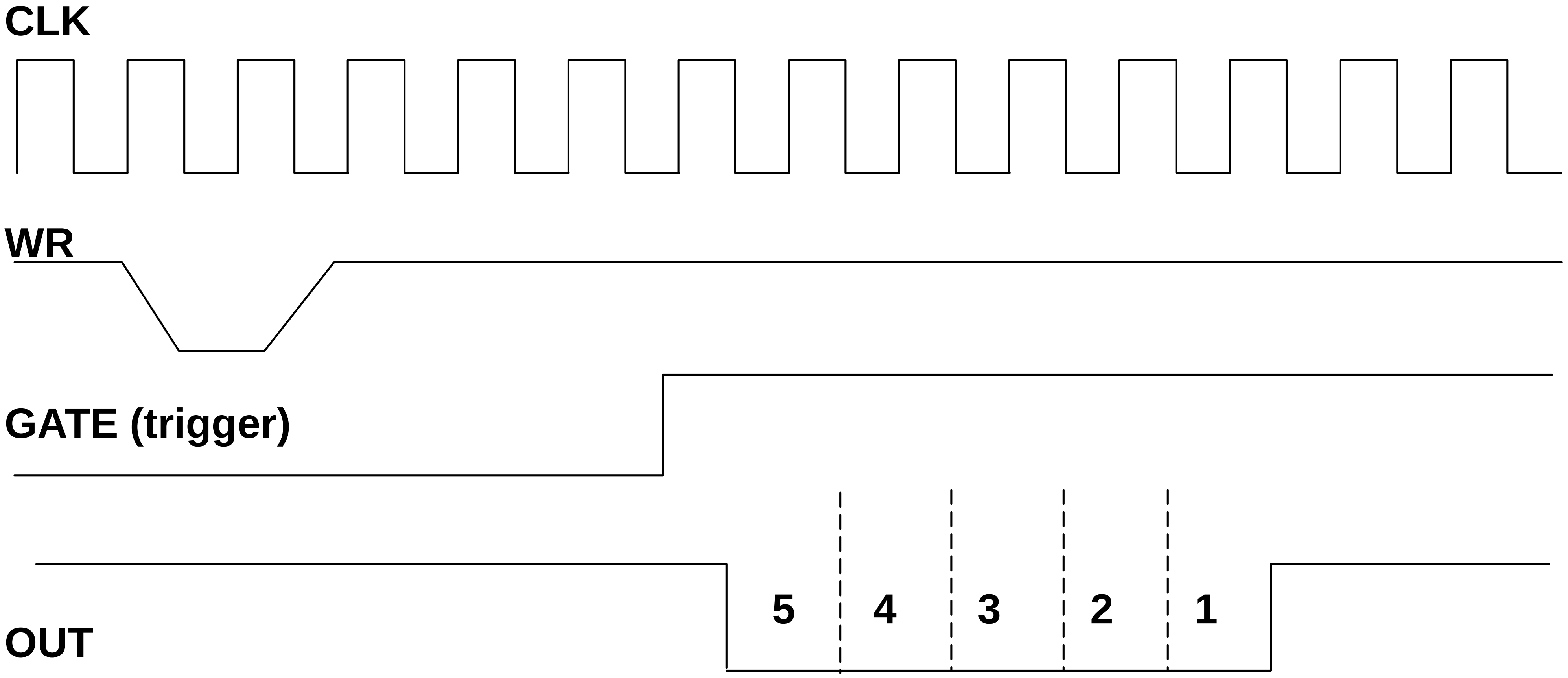
The count starts when a rising edge is detected in the signal GATE.

During the counting, OUT is low, otherwise it is high.

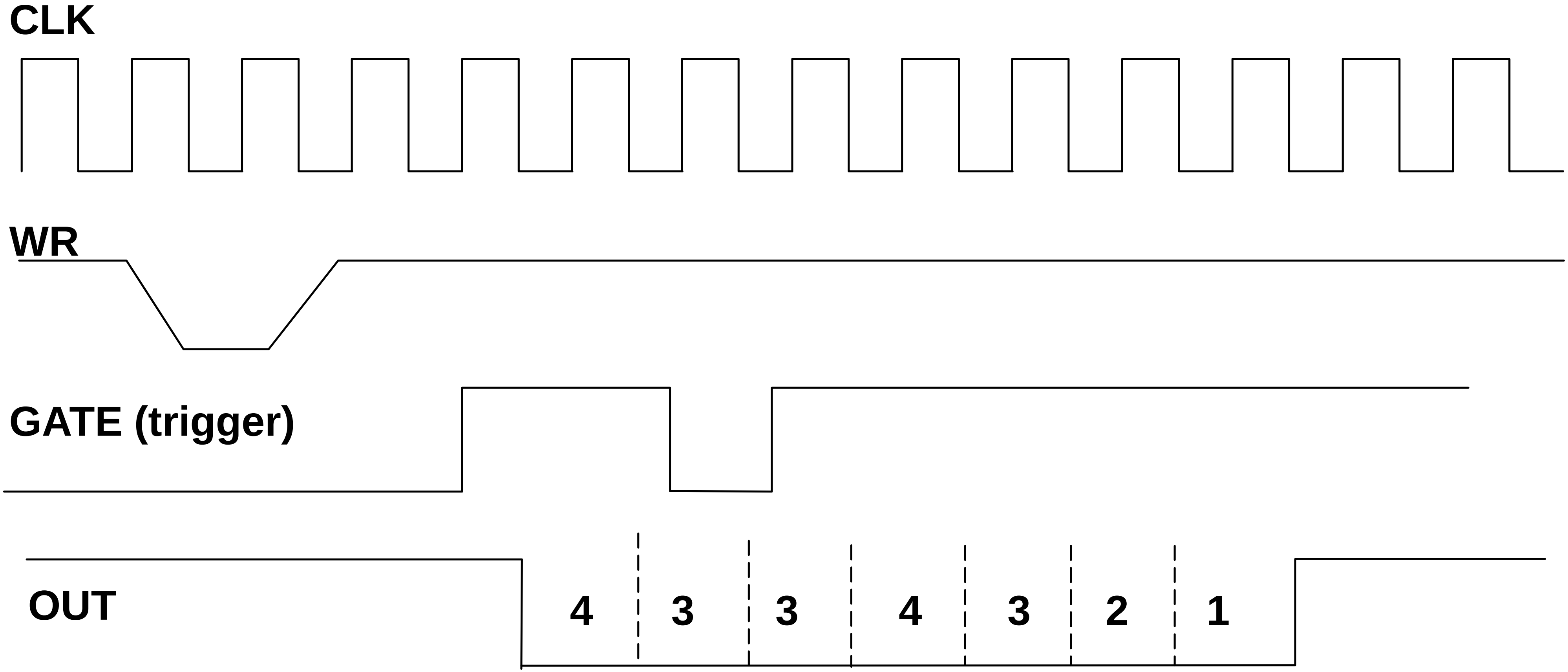
If a new value is loaded, it will not influence the counting process but the following one.

If a new rising edge is detected in GATE, the counting process restart.

Mode 1: example



Mode 1: example (II)



Mode 2
(*Rate generator*)

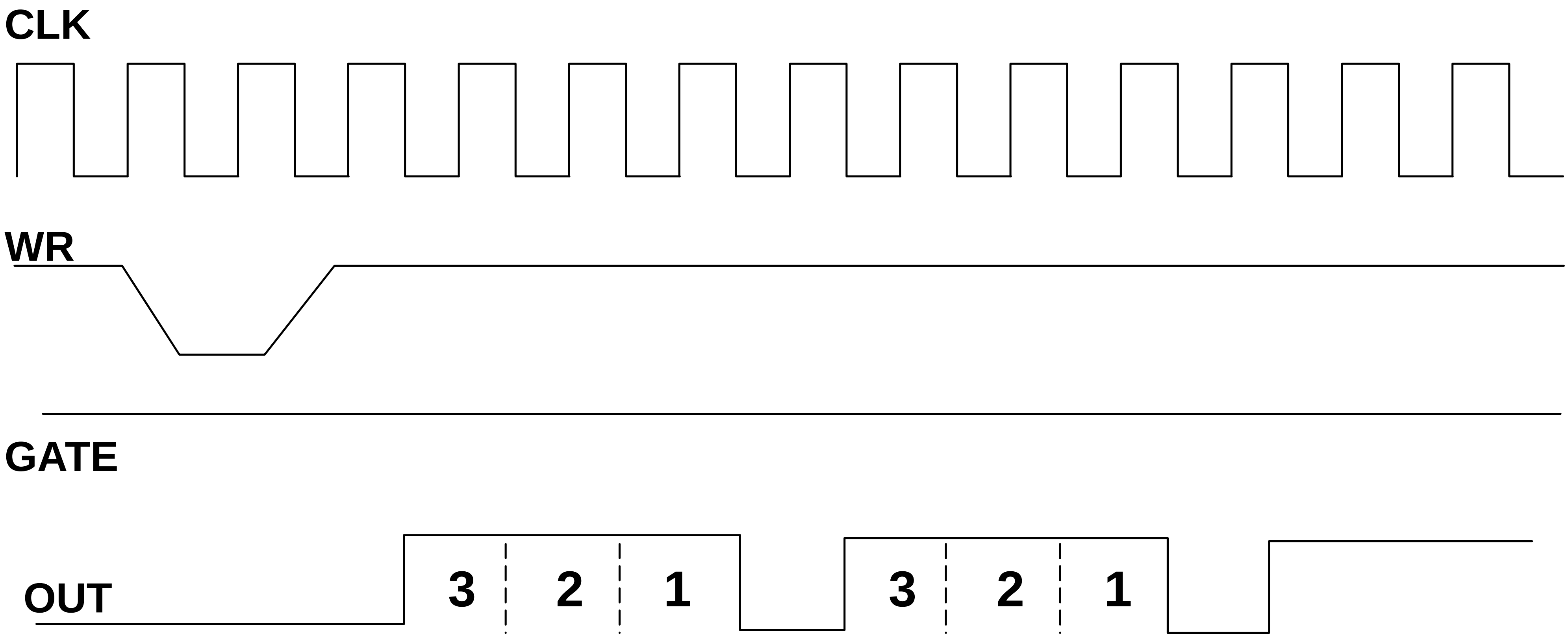
The counter works as a frequency divisor.

After *n* clock cycles in CLK, OUT goes low during a clock cycle.

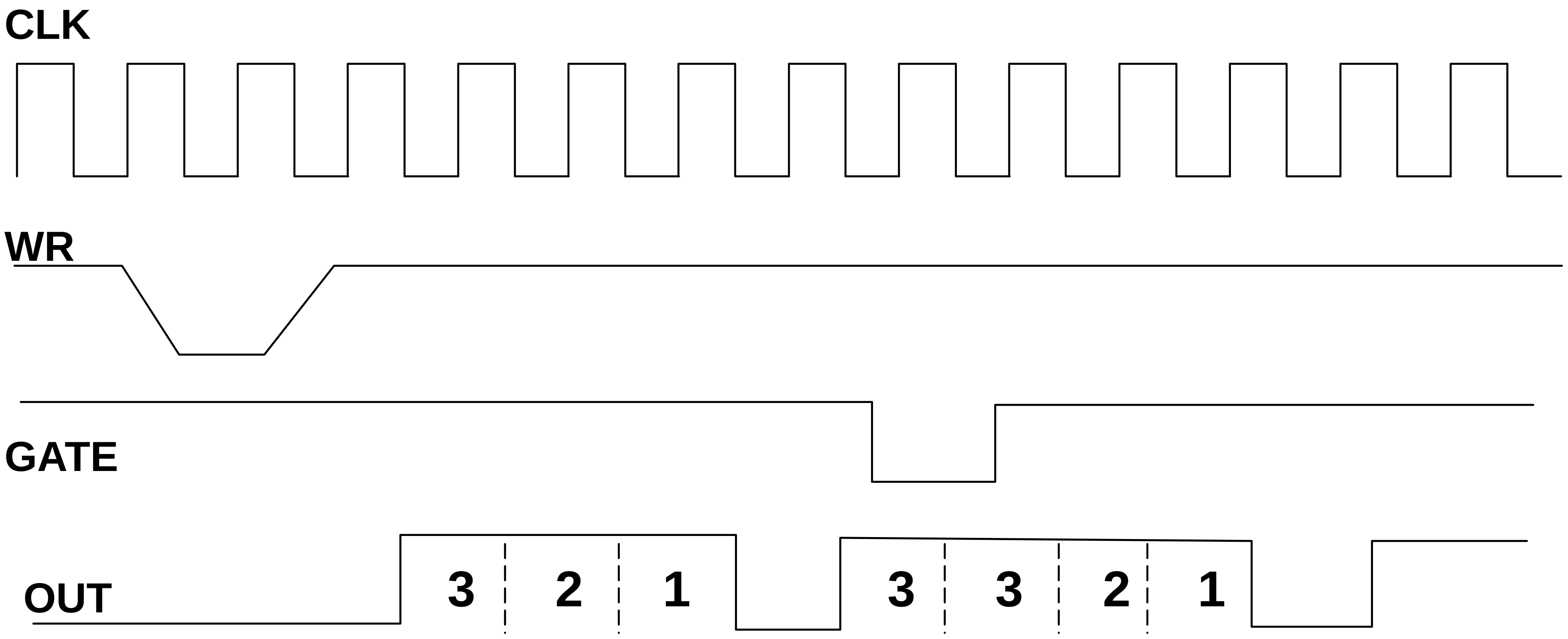
Counting starts when the counter is loaded; a new load influences the next counting process, not the current one.

If GATE is low, OUT is fixed to high, then the next rising edge restart the counting process.

Mode 2: example



Mode 2: example (II)



Exercise

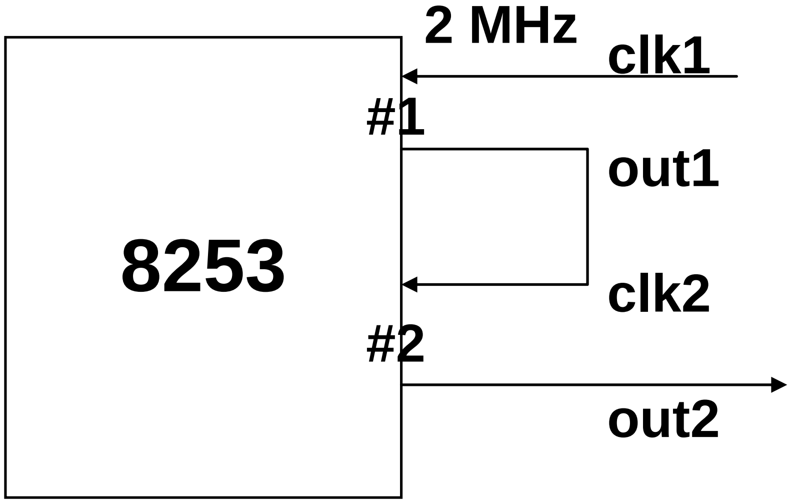
Generate a signal able to interrupt every 4s.

Assume that the system clock frequency is 2 MHz.

Solution:

The number of clock cycles to be count is 8.000.000. It is not possible to represent this number using 16 bits, then, it is necessary to concatenate two counters.

HWD diagram:



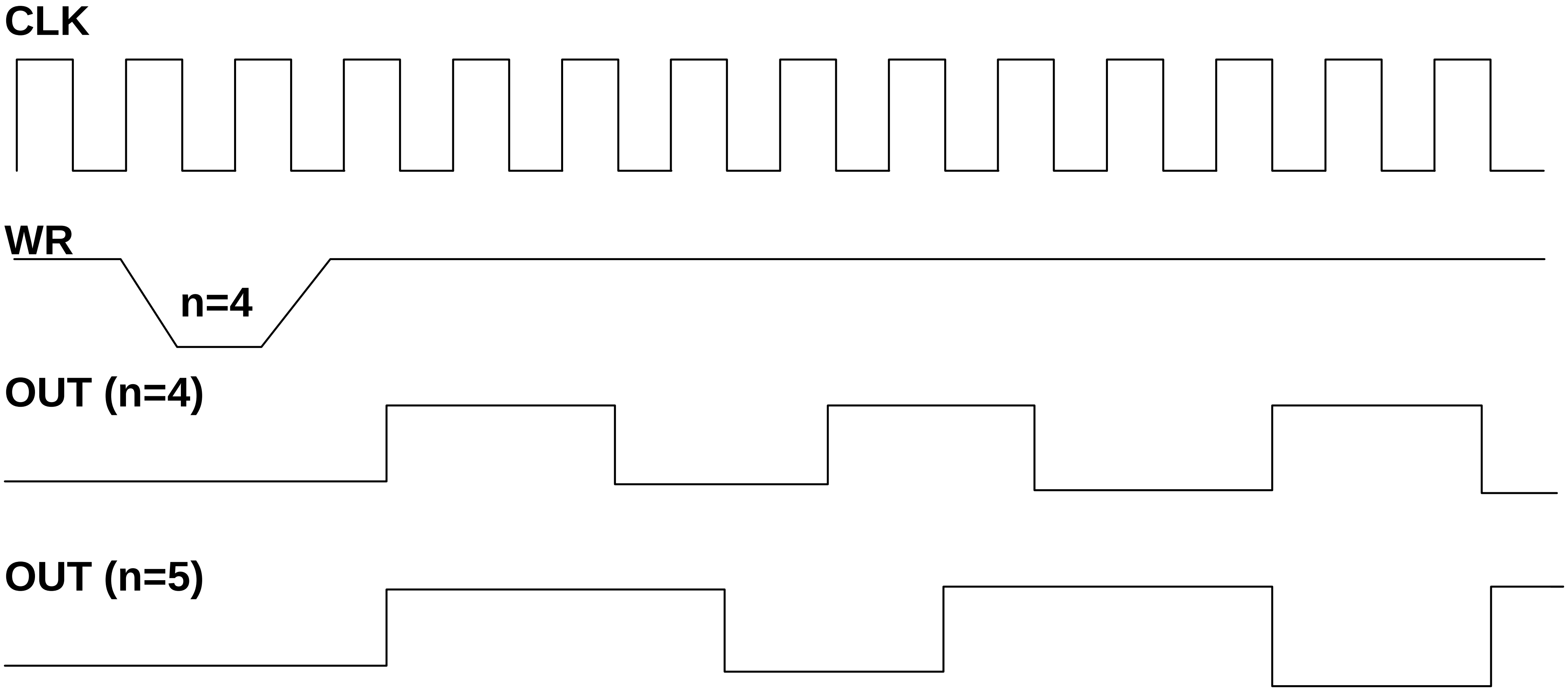
```
MOV       DX, 43h
MOV       AL, 74h   ; Counter 1, mode 2, binary
OUT       DX, AL
MOV       AL, 94h   ; Counter 2, mode 2, binary
OUT       DX, AL
MOV       DX, 41h
MOV       AX, 50000 ; 8,000,000 / 160
OUT       DX, AL    ; LSB counter 1
MOV       AL, AH
OUT       DX, AL    ; MSB counter 1
INC       DX
MOV       AL, 160
OUT       DX, AL    ; LSB counter 2
```

Mode 3
(*Square wave generator*)

It is similar to mode 2, but OUT signal is maintained high by a half of the counting period.

In the case, the counter constant *n* is odd, OUT is low for *(n-1)/2* and *high* for *(n+1)/2* clock cycles.

Mode 3: example



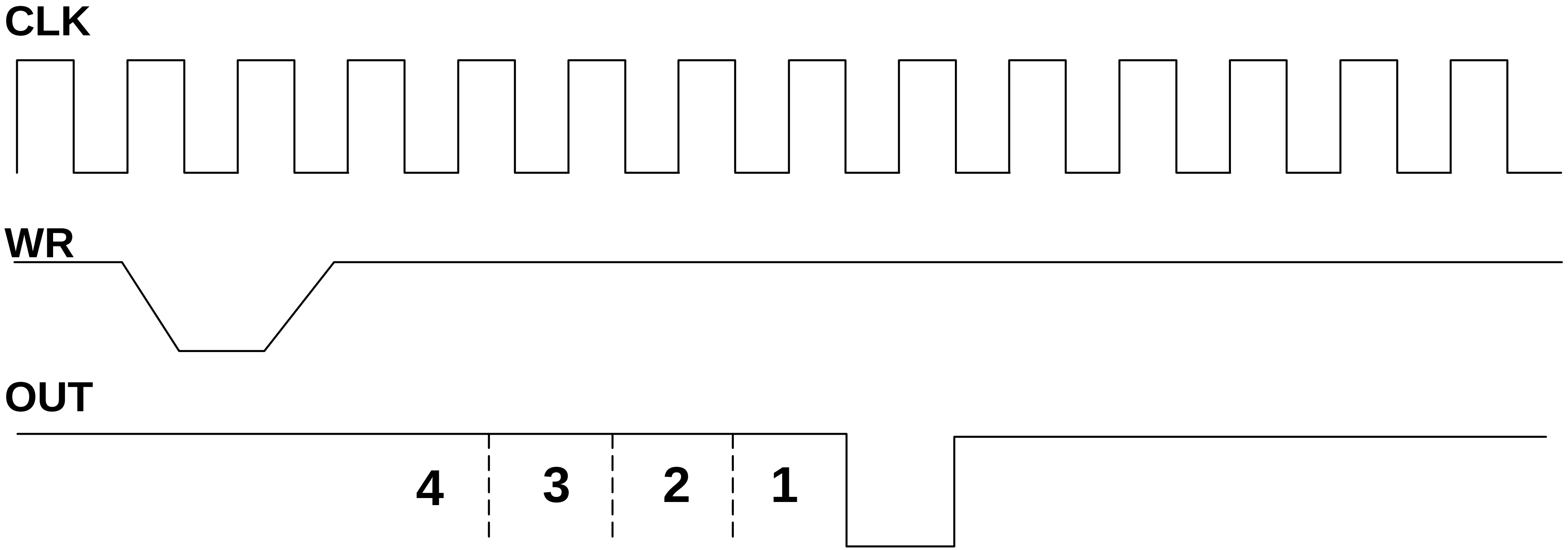
Mode 4
(*S/W Triggered Strobe*)

Once the counter is loaded, its value starts to decrement, keeping the OUT signal high. Once the termination value is reached, OUT goes to low during a clock cycle, and then returns high.

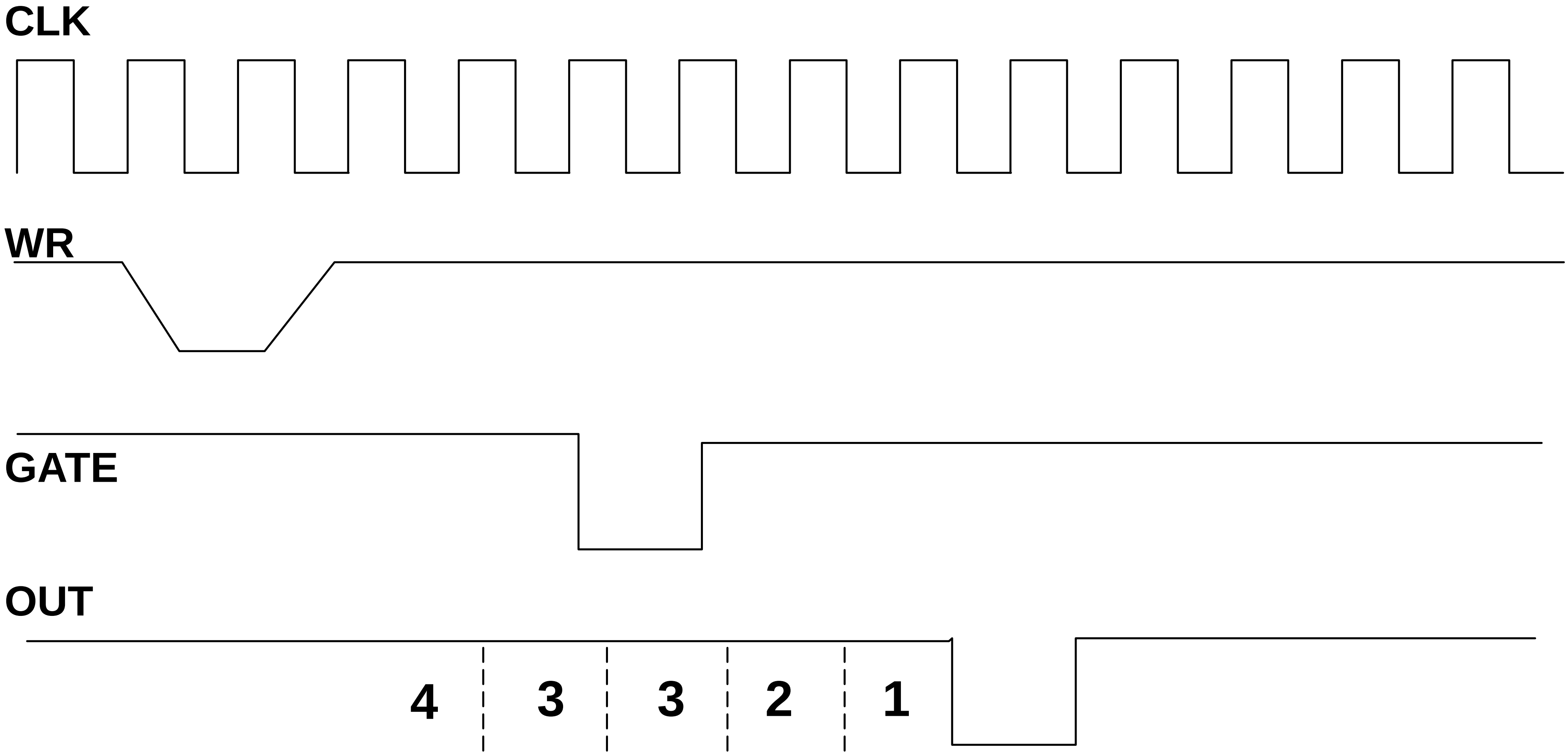
If a new value is loaded, the current counting process is not modified.

The counting process is paused when GATE is low, then continues the counting process when GATE is high.

Mode 4: example



Mode 4: example

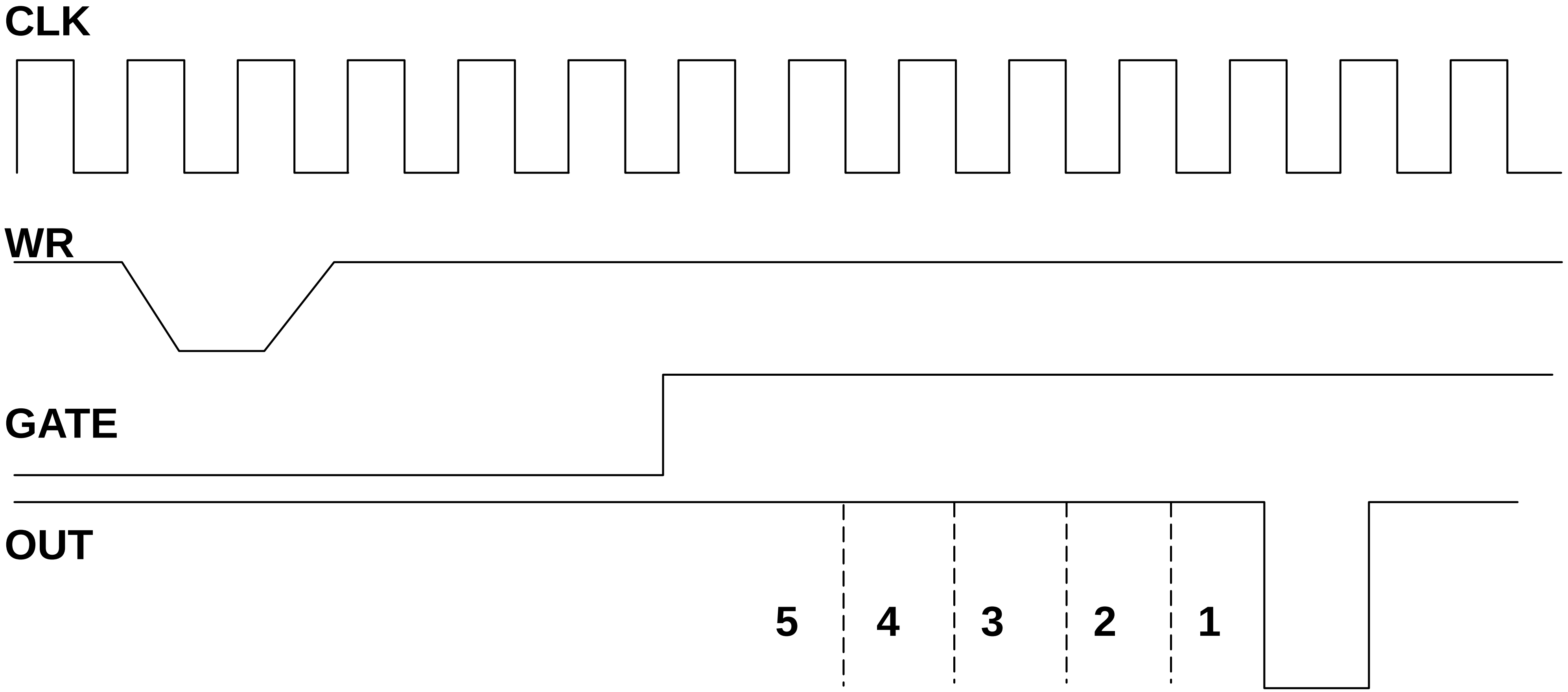


Mode 5
(*H/W Triggered Strobe*)

The counting process starts when a rising edge is detected in GATE. OUT goes low for a clock cycle when the termination value is reached.

Counter restarts when a rising edge is detected in GATE.

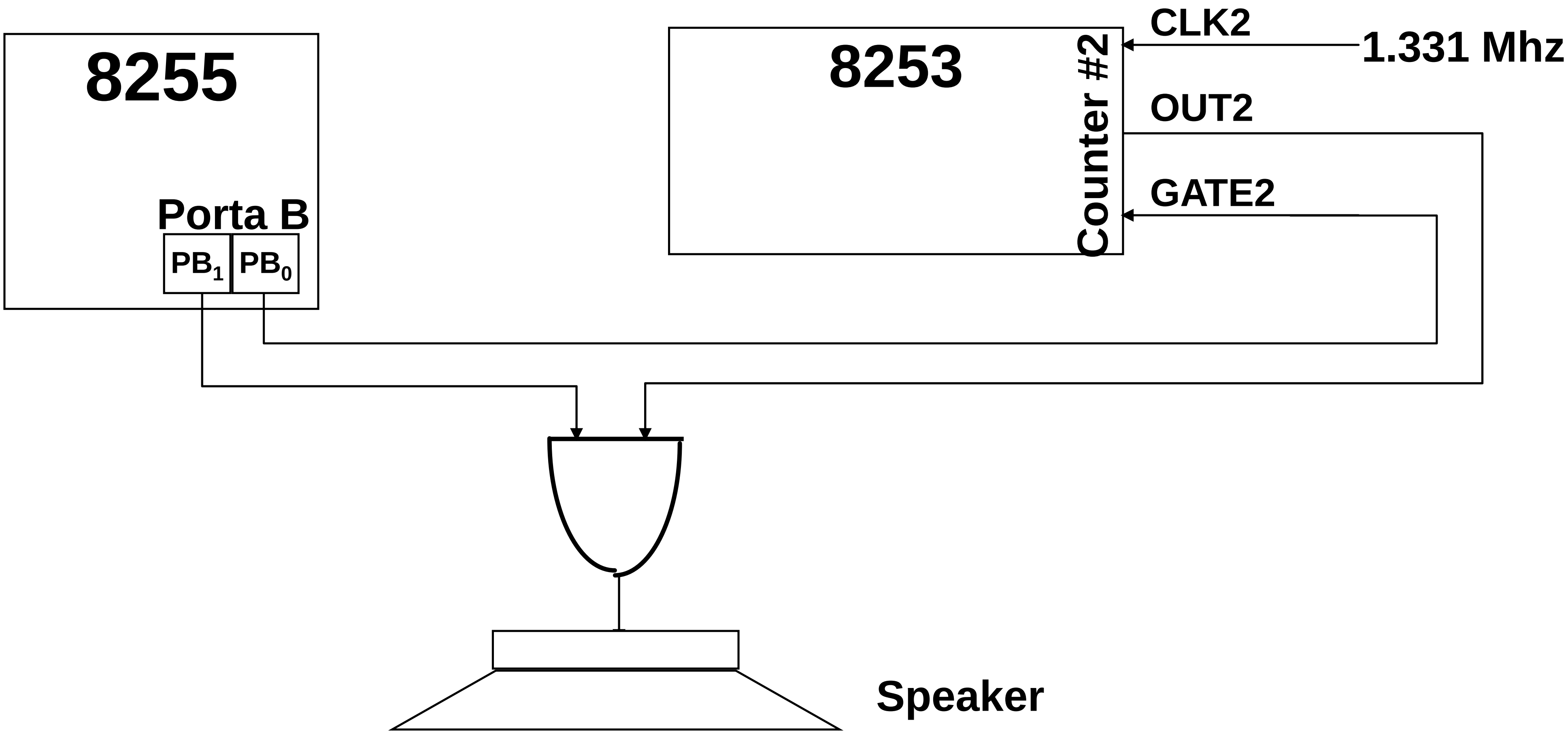
Mode 5: example



Example

In the following, it will be explained how to use the PC speaker via software.

Hardware diagram



Hardware diagram (II)

The bit 0 of the 8255 port B is connected to the 8253 GATE2, while the bit 1 in input is connected to an AND gate.

The second AND gate input is connected to the counter 2 OUT signal.

The AND output is connected to the speaker.

Addresses:

8255 (Port B) ⇒ 61h

8253 ⇒ 40h ÷43h.

Program code

The code procedure `beep` is explained. It receives in DI the frequency in Hz, and in BX the duration in hundredth of seconds of the desired sound.

```
PB_8255  EQU   61h  ; port B 8255
C2_8253  EQU   42h  ; cont. 2 8253
CW_8253  EQU   43h  ; Control Word 8253
.CODE
BEEP    PROC
     PUSH AX
     PUSH BX
     PUSH CX
     PUSH DX
     MOV  AL, 0B6h  ; counter 2, 2 byte, mode 3,
                    ; binary
     OUT  CW_8253, AL
     MOV  DX, 14h
     MOV  AX, 4F38H ; 144F38h = 1331000
                    ; = freq. clock
     DIV  DI
```

```
          OUT  C2_8253, AL      ; freq. (LSB)
          MOV  AL, AH
          OUT  C2_8253, AL      ; freq. (MSB)
          IN   AL, PB_8255
          MOV  AH, AL
          OR   AL, 03h
          OUT  PB_8255, AL      ; beeper enabled
l1:       MOV  CX, 2801
l2:       LOOP l2
          DEC  BX
          JNZ  L1
          MOV  AL, AH
          OUT  PB_8255,AL       ; beeper disabled
          POP  DX
          POP  CX
          POP  BX
          POP  AX
          RET
BEEP      ENDP
```

```
            MOV  CX, n
lab:        LOOP LAB
```

Which should be the value of *n* in order to have a 10ms cycle?

`LOOP` instruction duration:

17 clock cycles if the branch is taken;

5 clock cycles if branch is not taken.

`MOV` instruction duration:

4 clock cycles.

Then:

$$[17(n-1)+5+4]T = 0.01 \text{ sec}$$

***T* is the processor clock cycle period. If the clock frequency is 5MHz, n = 2801.**

Main program

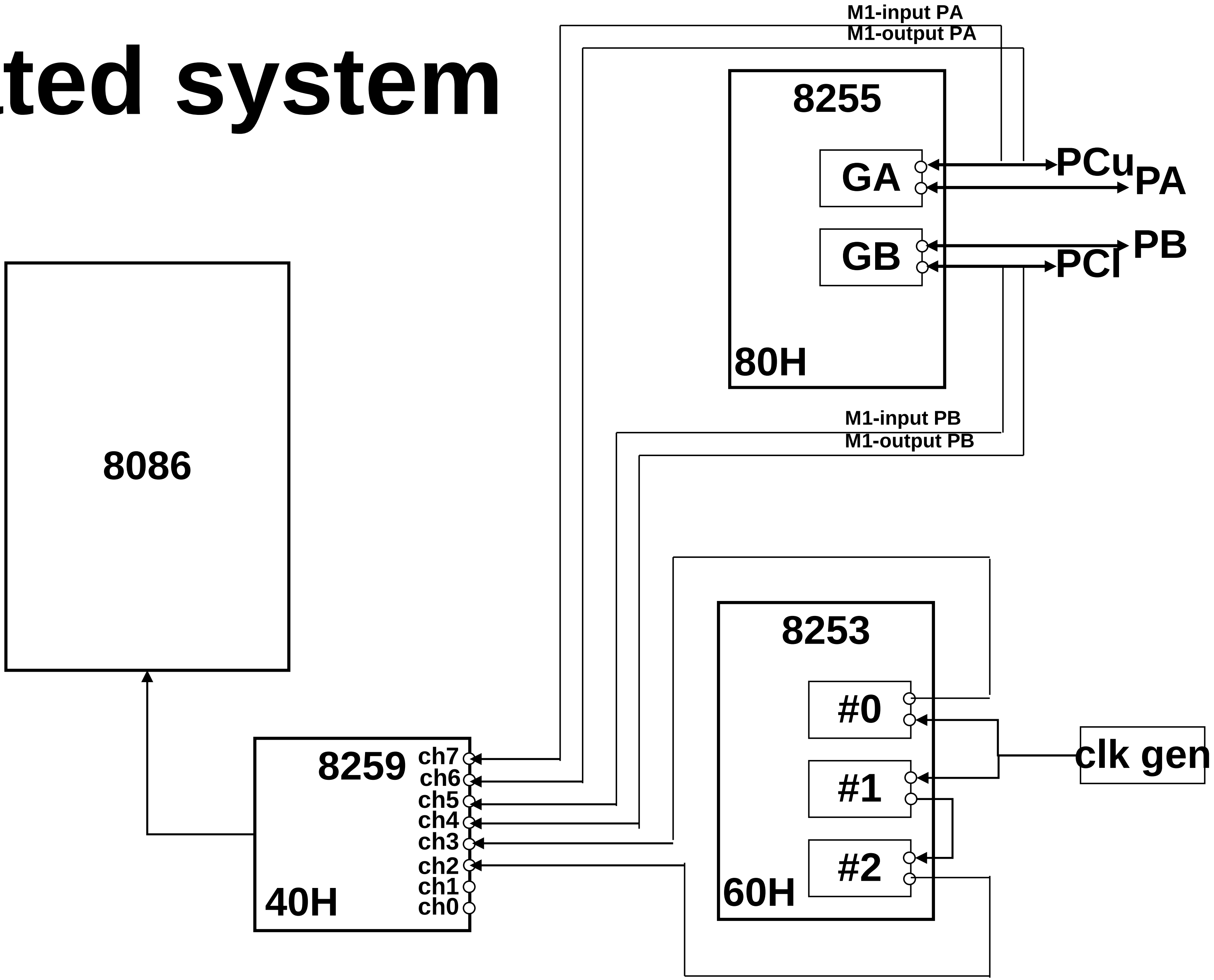
```
          ; it plays sounds in the 500-2000Hz range.
          ; any note last by 150ms.
          .STARTUP
          MOV  DI, 500
          MOV  BX, 15
again:    CALL BEEP
          ADD  DI, 20
          CMP  DI, 2000
          JB   again
          .EXIT
          END
```

Music notes

Notes frequency in the central octave in a piano.

DO	**261,7 Hz**
RE	**293,7 Hz**
MI	**329,6 Hz**
FA	**349,2 Hz**
SOL	**392,0 Hz**
LA	**440,0 Hz**
SI	**493,9 Hz**

Emulated system


M1-input PA
M1-output PA
8255
GA
GB
PCu
PA
PB
PCl
80H
M1-input PB
M1-output PB
8086
8253
#0
#1
#2
60H
clk gen
8259
ch7
ch6
ch5
ch4
ch3
ch2
ch1
ch0
40H