

# Mikrocomputer-Technik

## MCT 49

### Teil 5: Timer

**Studiengang Technische Informatik (BA)**

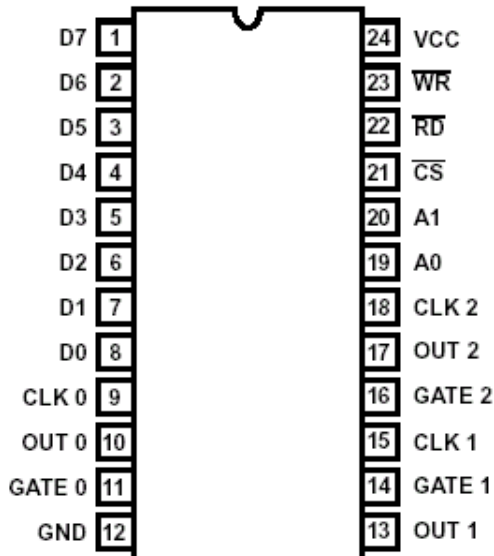
**Prof. Dr.-Ing. Alfred Rožek**

nur für Lehrzwecke  
Vervielfältigung nicht gestattet

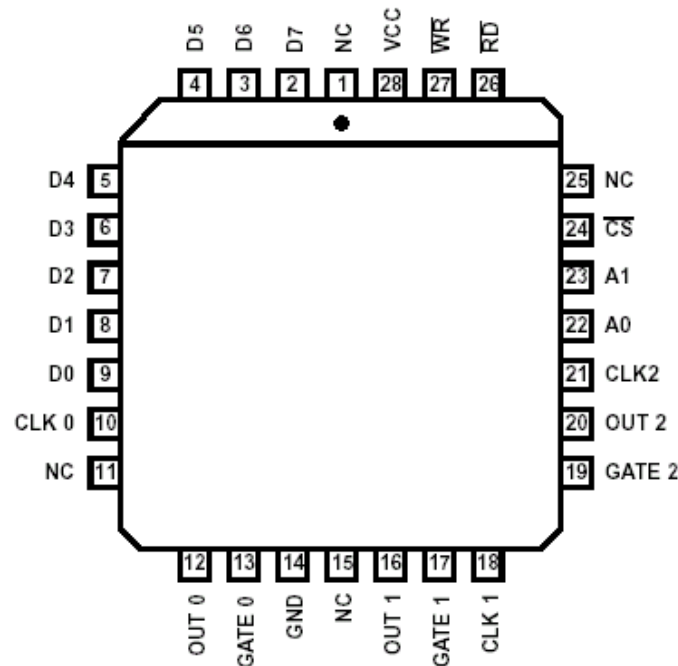
# Programmierbarer Intervall-Timer 8253/8254 (PIT)

- Anwendung
- Integration in ein Mikrocomputersystem
- Interne Struktur des Bausteins
- Programmierung
- Betriebsarten

82C54 (PDIP, Cerdip, SOIC)  
TOP VIEW



82C54 (PLCC/CLCC)  
TOP VIEW



- Drei voneinander unabhängig arbeitende 16-Bit-Zähler
- Taktfrequenz
  - bis 2 MHz beim 8253
  - und
  - bis zu 12 MHz beim 82C54
- per Software einstellbare Betriebsarten
- Binäres oder dezimales Zählen

Zur Nacharbeit empfohlen:  
[data\\_sheet\\_8254.pdf](#)

# Anwendung

---

- Steuerbarer Signalgenerator
- Ereigniszähler
- Binärer Frequenzmultiplizierer
- Echtzeituhr
- Digital einstellbare monostabile Kippstufe
- Flexibler Motor-Steuerbaustein

## **Zweck / Aufgabe:**

Programmgesteuerte Erzeugung präziser Verzögerungszeiten mit geringer Prozessorbelastung.  
(Anstatt Polling oder Software-Warteschleifen)

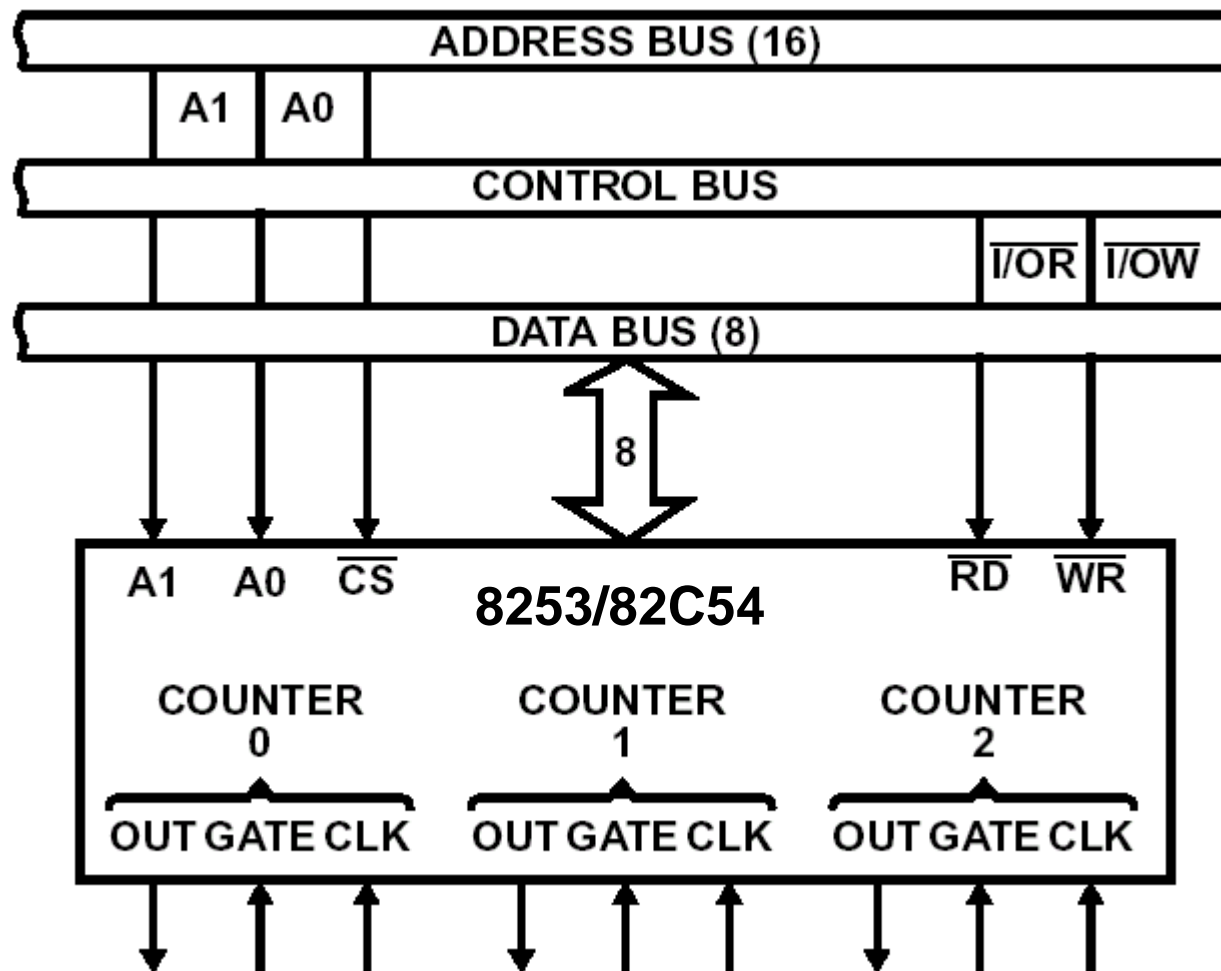
# Zähler 0, 1 und 2

Die drei Zähler sind funktionell vollkommen identisch. Jeder Zähler besteht aus einem vorbelegbaren 16-Bit-Abwärtszähler, der wahlweise für duales oder dezimales Zählen eingestellt werden kann. Die Funktion des Eingangs, Gates und Ausgangs wird durch die Auswahl des Steuerwortes im Steuerwort-Register bestimmt.

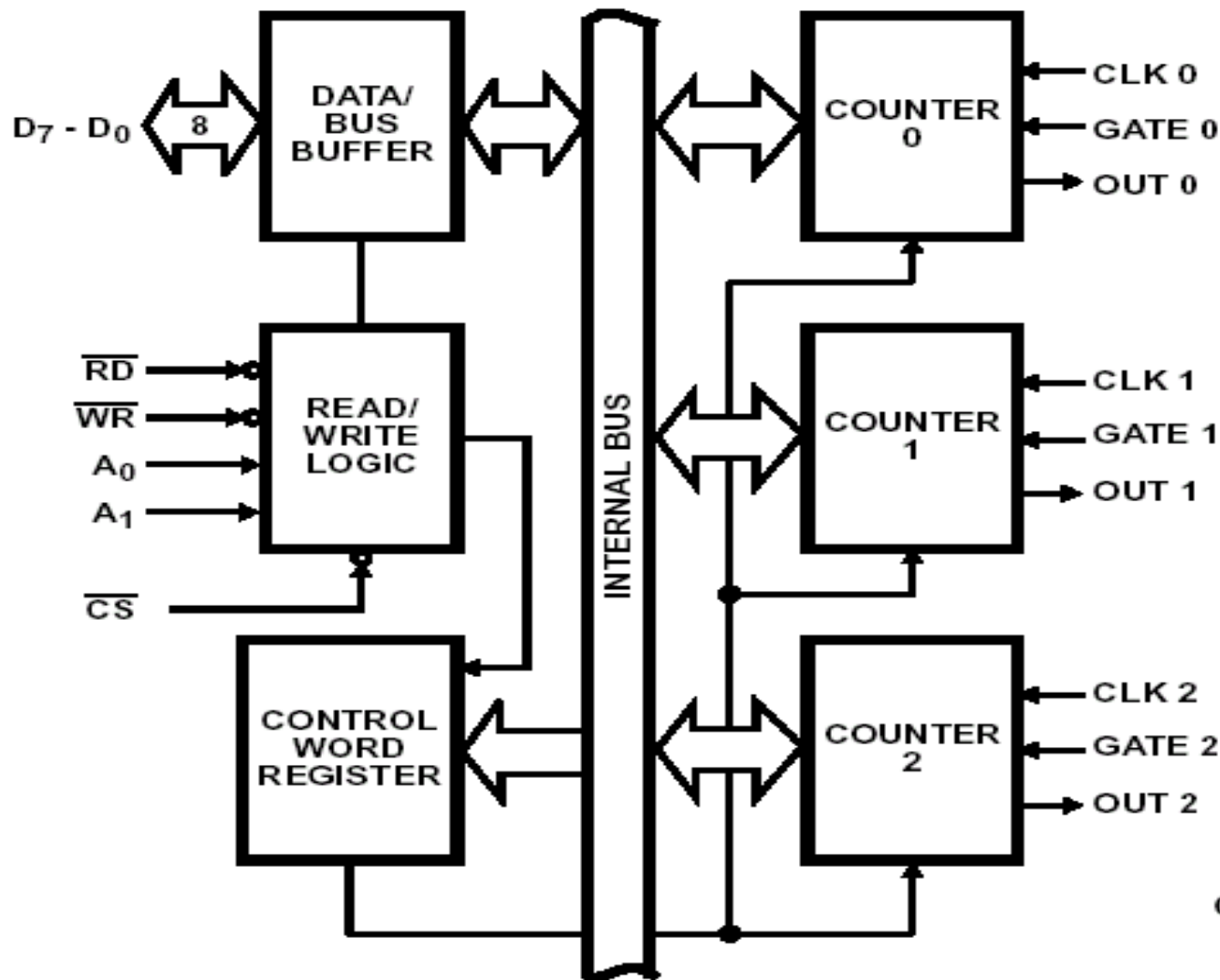
Die Zähler arbeiten voneinander unabhängig und können parallel in vollkommen verschiedenen Betriebsarten betrieben werden. Im Format des Steuerwortes sind spezielle Kennzeichen zum Laden des Zähleranfangswertes enthalten, so daß der Aufwand an System-Software möglichst gering bleibt.

Der Anwender kann bei Verwendung eines Zählers als Ereigniszähler, den Zählerinhalt mit einem einfachen LESE-Befehl auslesen. Für das Lesen der Zählerinhalte während des Zählens, besitzt der 8253/82C54 ein spezielles Steuerwort und eine Logik, so dass der Eingangstakt nicht gesperrt werden muss.

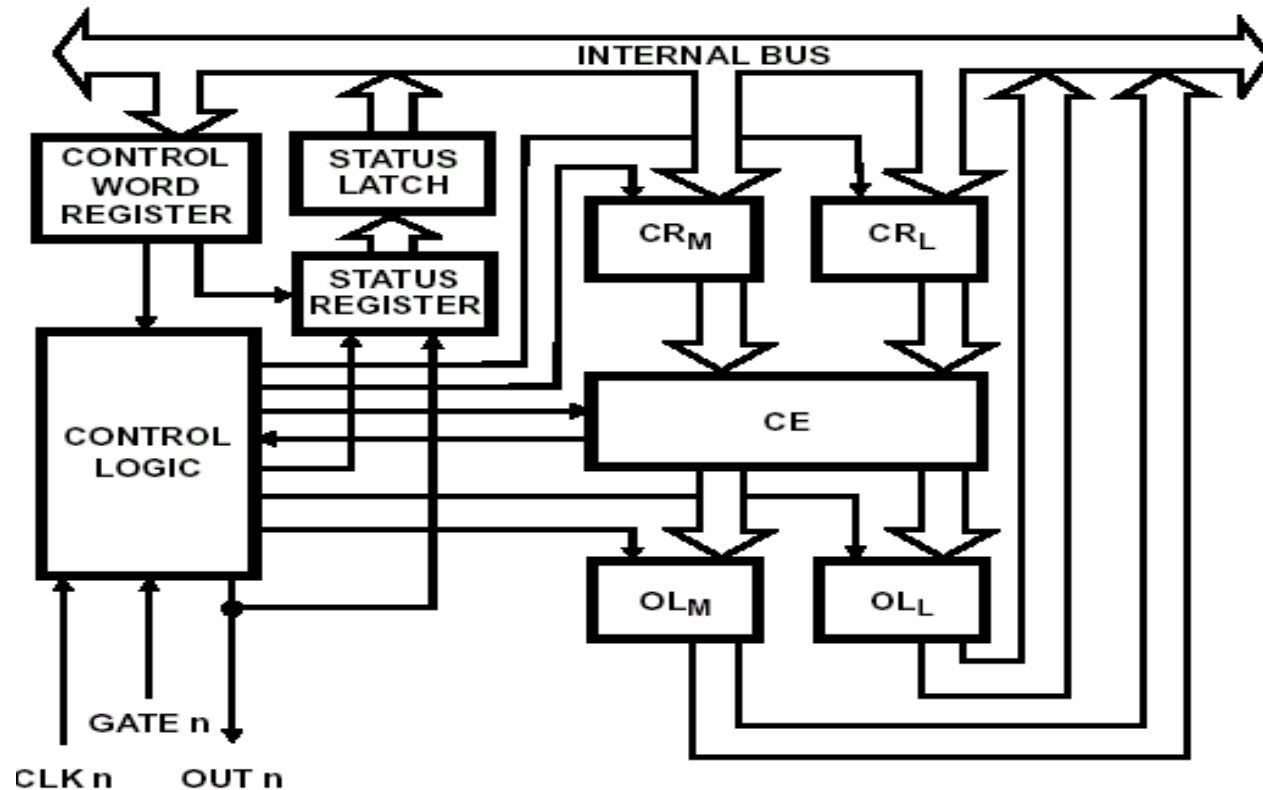
# Integration in ein Mikrocomputersystem



# Interne Struktur des 8253 / 8254



# Interner Aufbau eines Zählers im PIT



- ◆ CE = Counting Element (Zählelement)
- ◆  $OL_M$ ,  $OL_L$  = Output Latches (M=Most significant Byte, L = Least significant Byte)
- ◆  $CR_M$ ,  $CR_L$  = Count Register vom Anwender programmierbar
- ◆ 8254: Status-Latch für Read-Back-Befehl (Steuerwort lesen)

## Adressen des Bausteins im PC:

I/O	Register	Zugriff
40H	Zähler 0	Schreiben/Lesen
41H	Zähler 1	Schreiben/Lesen
42H	Zähler 2	Schreiben/Lesen
43H	Steuerregister	Nur Schreiben

## Initialisierung:

1. Steuerwort in das Steuerregister schreiben
2. Zähler mit einem Anfangswert laden  
(je nach Reload-Mode ein oder zwei Schreibbefehle!)

## Neuen Zählwert vorgeben:

Zähler mit Anfangszählwert entsprechend dem eingestellten Reload-Mode laden

(Das Steuerregister darf natürlich inzwischen nicht verändert worden sein)



# Aufbau des Steuerworts

7	6	5	4	3	2	1	0
<b>SC1</b>	<b>SC0</b>	<b>RL1</b>	<b>RL0</b>	<b>M2</b>	<b>M1</b>	<b>M0</b>	<b>BCD</b>

BCD	Zählformat
0	Binärzählen
1	In BCD zählen

SC1	SC0	Kanal
0	0	Zähler 0
0	1	Zähler 1
1	0	Zähler 2
1	1	8253: nicht erlaubt 8254: Read Back

RL1	RL0	Reload Operation
0	0	Zählerinhalt zwischenspeichern
0	1	Low-Byte Lesen/Schreiben
1	0	High-Byte Lesen/Schreiben
1	1	1. Low-Byte Lesen/Schreiben 2. High-Byte Lesen/Schreiben

M2	M1	M0	Mode/Betriebsart
0	0	0	Mode 0: Unterbrechungsanforderung bei Nulldurchgang
0	0	1	Mode 1: Monostabile Kippstufe
X	1	0	Mode 2: Taktgenerator mit Teiler N
X	1	1	Mode 3: Rechteckgenerator
1	0	0	Mode 4: Softwaregesteuertes Strobe
1	0	1	Mode 5: Hardwaregesteuertes Strobe

# Lese-Operationen<sub>1</sub>

Bei den meisten Zähleranwendungen wird es notwendig sein, den Zählerinhalt während des Zählens auszulesen und in Abhängigkeit von diesem Wert eine Entscheidung zu treffen.

Ereigniszähler sind die häufigsten Anwendungsfälle, die diese Möglichkeit ausnutzen.

Die Timer 8253/82C54 haben eine interne Logik, die es ermöglicht, den Zählerinhalt von jedem der drei Zähler auszulesen, ohne den Zählvorgang zu stören.

Es gibt drei Möglichkeiten, wie der Anwender den Zählerinhalt auslesen kann:

- 1. Leseoperation des Zählers** (zuvor muss der Zählvorgang gestoppt werden)
- 2. Counter-Latch Befehl**
- 3. Read-Back Befehl**

# Lese-Operationen<sub>2</sub>

## 1. Leseoperation des Zählers

Einfache Leseoperation des ausgewählten Zählers. Durch die Eingänge A0 und A1 wird der Zähler, dessen Inhalt ausgelesen werden soll ausgewählt. Die einzige Bedingung ist, um einen gültigen Zählerinhalt zu bekommen, dass der laufende Zählvorgang des ausgewählten Zählers durch den GATE-Eingang oder mit Hilfe einer externen Logik über den Takteingang unterbrochen wird.

Der Zählerinhalt wird wie folgt gelesen (bei RL1=1 und RL0=1):

- Mit dem ersten Lesebefehl an den ausgewählten Zähler wird das niederwertige Byte (LSB) des Zählerinhalts gelesen.
- Mit dem zweiten Lesebefehl das höherwertige Byte (MSB).

Wegen der internen Logik des 8253/82C54 ist es unbedingt notwendig, den Lesevorgang vollständig abzuschließen. Ist durch das Steuerwort festgelegt, dass 2 Bytes von einem Zähler gelesen werden sollen, müssen auch beide gelesen werden, bevor der Zähler wieder mit anderen Werten geladen werden kann.

# Lese-Operationen<sub>3</sub>

## 2. Counter-Latch Befehl

Die zweite Möglichkeit zur Bestimmung des Zählerinhalts ist das Lesen während des Zählens. Der Zählerbaustein hat eine spezielle interne Logik, die es dem Anwender ermöglicht, mit Hilfe eines einfachen Schreib-Befehls an die Adresse des Steuerwortregisters, den Inhalt jedes Zählers zu lesen, ohne dabei den Zählvorgang zu stören oder zu beeinflussen.

Will der Anwender den Zählerinhalt während des Zählens lesen, muss er das Steuerwort-Register mit einem besonderen Code laden, der veranlaßt, dass der aktuelle Zählerinhalt des ausgewählten Zählers in einen Zwischenspeicher übergeben wird. Der Anwender kann anschließend mit einer normalen Leseoperation an die Adresse des entsprechenden Zählers den Zählerinhalt aus dem Zwischenspeicher lesen.

**Steuerwort zur Übernahme des Zählerinhalts in einen Zwischenspeicher:**

7	6	5	4	3	2	1	0
SC1	SC0	0	0	X	X	X	X

SC1, SC0: Zählerauswahl (Select Counter)

D5, D4: kennzeichnet die Datenübergabe in einen Zwischenspeicher

# Lese-Operationen<sub>4</sub>

## 3. Read-Back Befehl (nur beim 82C54)

7	6	5	4	3	2	1	0
1	1	/ZW	/ST	Z2	Z1	Z0	0

### Anwendung:

- Modus eines Zählers ermitteln
- Stand eines Zählers ermitteln

### Steuerregister:

/ZW = 0: Zählerstand ermitteln

/ST = 0: Zählernodus ermitteln (d.h. Statuswort lesen)

Z2 bis Z0: Zählerauswahl (positive Logik)

- Zähler können einzeln oder
- gemeinsam angesprochen werden
- Es müssen dann auch alle Zähler gelesen werden. Wenn nicht, ist der betreffende Zähler blockiert

### Beispiel: Statuswort (Modus) des Zählers 0 ermitteln

```

MOV AL, 11100010B ;Read Back, Modus ermitteln, Zähler 0
OUT 43H,AL        ;Steuerwort schreiben
IN  AL, 40H        ;Modus von Zähler 0 lesen
  
```

# Programmierung der Zählregister (Byteweise)

## Byteweise lesen/schreiben:

Vorteil: Einfach, es wird nur ein OUT-Befehl benötigt.

Nachteil: Geringer Wertebereich (0 bis 256) oder geringe Auflösung (in 256er Schritten)

## 1. Nur Low-Byte lesen/schreiben

- beim Schreiben wird das High-Byte automatisch auf Null gesetzt
- Wertebereich: 0 bis FFH

Beispiel: Im Modus 3 soll Zähler 2 ein Rechtecksignal von 10 kHz ausgeben.

Zähleranfangswert =  $\text{CLK} / \text{Ausgabefrequenz} = 1193,18 \text{ kHz} / 10 \text{ kHz} = 119$  (gerundet)

```
MOV AL, 10010110B ;Zähler2, nur Low-Byte, Mode 3, Binär zählen
OUT 43H, AL        ;Steuerregister schreiben
MOV AL, 119        ;Zähleranfangswert
OUT 42H, AL        ;in Zählregister 2 schreiben
```

## 2. Nur High-Byte lesen/schreiben

- beim Schreiben wird das Low-Byte automatisch auf Null gesetzt
- Wertebereich: (0 bis FFH) \* 256 (gezählt wird also im 16-Bit-Format)

# Programmierung der Zählregister (Wortweise)

## Erst Low-Byte, dann High-Byte schreiben/lesen

### 1. Schreiben

Beispiel: Im Modus 3 soll Zähler 2 ein Rechtecksignal von 440 Hz ausgeben.

```
MOV AL, 10110110B ;Zähler2, Low+High, Mode 3, Binär zählen
OUT 43H, AL        ;Steuerregister schreiben
MOV AX, 1193180/440 ;Frequenz = 440 Hz
OUT 42H, AL        ;Zuerst Low-Byte,
MOV AL, AH         ;dann High-Byte
OUT 42H, AL        ;an den Zähler 2 ausgeben
```

### 2. Lesen

- a) Zählvorgang durch GATE = 0 unterbrechen und dann zwei Lesebefehle
- b) Mittels Reload-Befehl den Zählerwert in einen internen Zwischenspeicher übertragen und dann auslesen. Beispiel:

```
MOV AL, 10000000B ;Zähler2, Zählerinhalt zwischenspeichern
                ;(Bit 3 bis 0 werden vom Baustein nicht ausgewertet!)
OUT 43H, AL      ;Steuerregister schreiben
IN AL, 42H       ;Low-Byte lesen
MOV AH, AL
IN AL, 42H       ;High-Byte lesen
XCHG AL, AH      ;AX = Zählerinhalt
```

# 8253: Mode 0

## Interrupt bei Zählende<sub>1</sub>

### Unterbrechungsanforderung bei Zählernulldurchgang

Der Zählerausgang liegt nach dem Ausgeben des Steuerwortes auf L-Pegel.

Nachdem der Zähleranfangswert in den ausgewählten Zähler eingeschrieben wurde und der Zählvorgang begonnen hat, bleibt der Zählerausgang weiterhin auf L-Pegel. Bei Erreichen des Zählerendstandes (0) geht der Ausgang auf H-Pegel und bleibt dort solange, bis ein neuer Zähleranfangswert oder ein neues Steuerwort geladen wird. Der Zähler selbst wird auch nach Erreichen des Zählerendstandes weiter dekrementiert.

Wird während des Zählens der Zähler neu geladen, so erfolgt

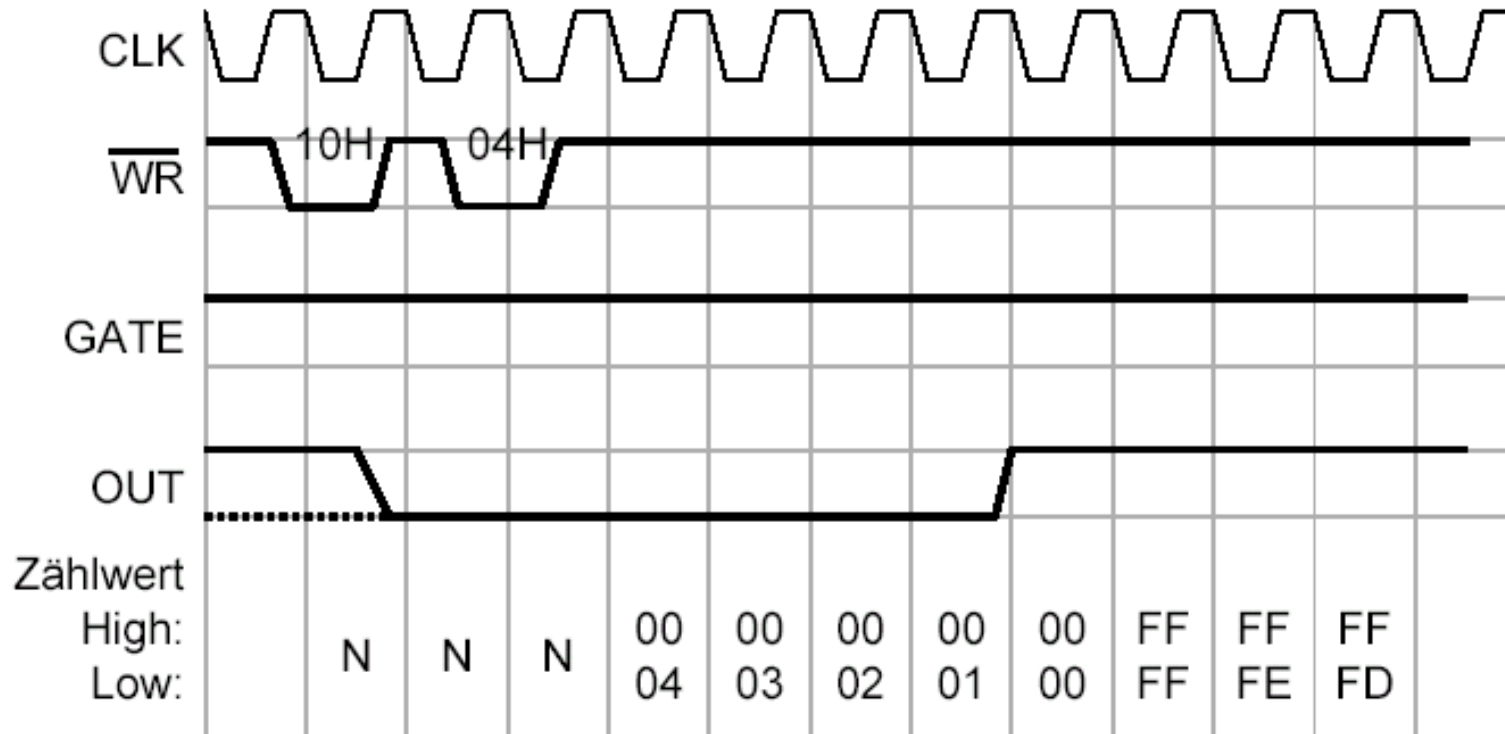
1. beim Laden des ersten Bytes der Abbruch des laufenden Zählvorgangs,
2. beim Laden des zweiten Bytes der Start mit dem neuen Anfangswert.

Durch einen L-Pegel am GATE-Eingang wird das Zählen gesperrt und durch einen H-Pegel wieder freigegeben.



# 8253: Mode 0

## Interrupt bei Zählende<sub>2</sub>



**Beispiel:** Zähler 0, Zählwert 4, Mode 0, nur Low-Wert genutzt, binär zählen

```
MOV AL, 00010000H    ;Zaehler 0, nur Low schr., Mode 0, binär zaehlen
OUT 43H,AL           ;Steuerwort in Steuerregister schreiben
MOV AL,4             ;Zaehlwert
OUT 40H,AL           ;in Zaehler Nr. 0 laden
```

# 8253: Mode 1

## Programmierbares Monoflop (One-Shot)<sub>1</sub>

---

### Programmierbare monostabile Kippstufe

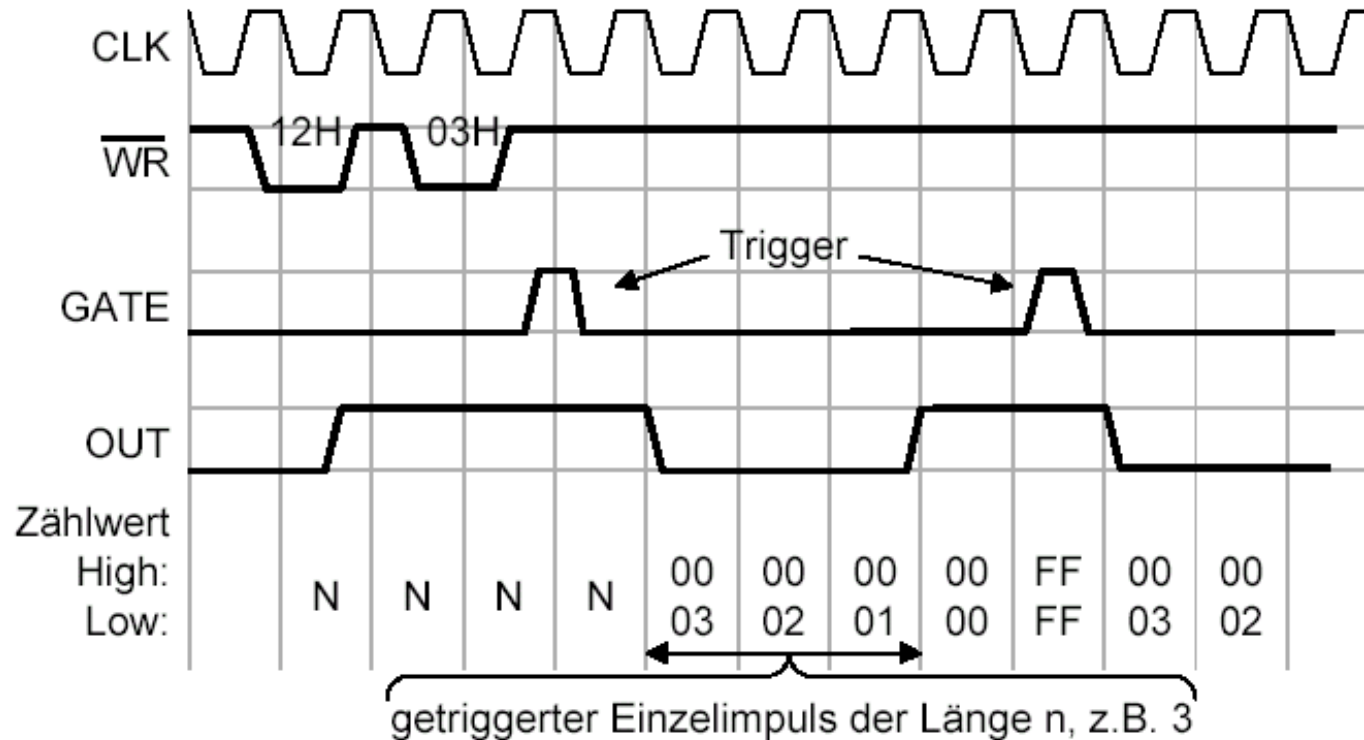
Der Zählerausgang geht nach einer positiven Signalfanke auf den GATE-Eingang und mit der nächsten Periode des Eingangstaktes auf L-Pegel.

Nach Erreichen des Endstandes (0) geht der Zählerausgang wieder auf H-Pegel über. Das Einschreiben eines neuen Zähleranfangswertes zu einem Zeitpunkt, bei dem der Ausgang auf L-Pegel liegt, hat keinen Einfluß auf den laufenden Zählvorgang. Der neue Anfangswert wird erst nach erneutem Triggern übernommen. Der aktuelle Zählerstand ist ohne weiteren Einfluß auf den Zählvorgang selbst jederzeit auslesbar (über einen Lesebefehl des Prozessors).

Die monostabile Kippstufe ist beliebig oft neu triggerbar, d.h., der Ausgang bleibt nach jeder ansteigenden Flanke des GATE-Eingangssignals auf L-Pegel bis der Zähler vom Anfangswert bis zum Endwert (0) gezählt hat.

# 8253: Mode 1

## Programmierbares Monoflop (One-Shot)<sub>2</sub>



**Beispiel:** Zähler 0, Zählwert 3, Mode 1, nur Low-Wert genutzt, binär zählen

```
MOV AL, 000100010H    ;Zaehler 0, nur Low schr., Mode 1, Bin. zaehlen
OUT 43H,AL             ;Steuerwort in Steuerregister schreiben
MOV AL,3               ;Zaehlwert
OUT 40H,AL             ;in Zaehler Nr. 0 laden
```

# 8253: Mode 2

## Ratengenerator<sub>1</sub>

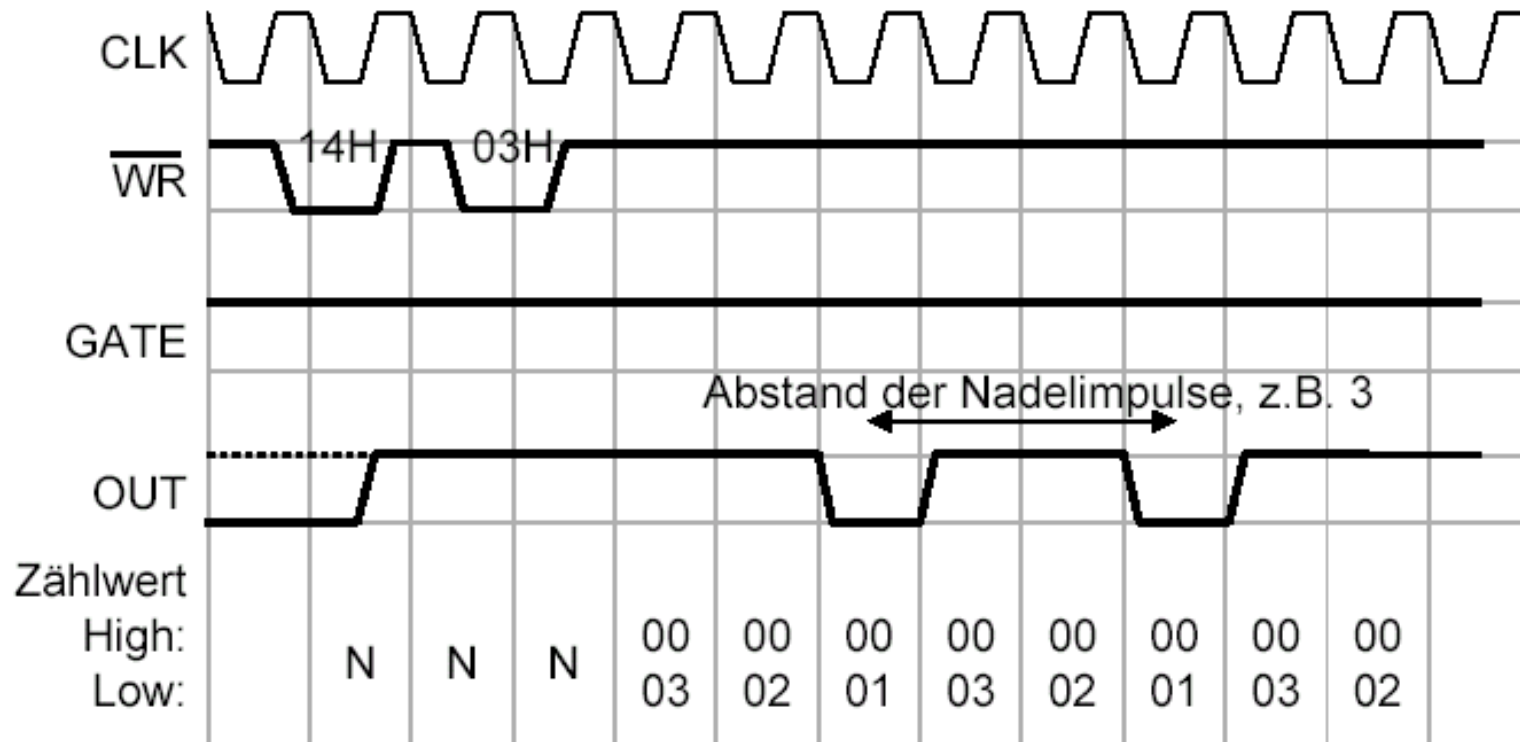
### Taktgenerator. Teiler durch N

Der Zählerausgang geht während eines Zählerdurchlaufs für eine Periode des Eingangstaktes auf L-Pegel. Die Anzahl der Taktperioden von einem Ausgangsimpuls zum nächsten ist gleich dem geladenen Zähleranfangswert. Wird zwischen zwei Ausgangsimpulsen der Zähler mit einem neuen Anfangswert geladen, wird erst die laufende Zählperiode beendet, bevor der Zähler mit dem neuen Wert startet.

Mit einem L-Pegel am GATE-Eingang wird am Zählerausgang ein H-Pegel erzeugt. Geht der GATE-Eingang wieder auf H-Pegel über, startet der Zähler erneut mit dem geladenen Anfangswert. Der GATE-Eingang ist daher zum Synchronisieren des Zählers geeignet.

Nachdem das Steuerwort für Betriebsart 2 eingeschrieben wurde, bleibt der Zählerausgang solange auf H-Pegel, bis der Zähleranfangswert vollständig geladen ist. Der Zähler ist also auch über die Software synchronisierbar.

# 8253: Mode 2 Ratengenerator<sub>2</sub>



**Beispiel:** Zähler 0, Zählwert 4, Mode 2, nur Low-Wert genutzt, binär zählen

```
MOV AL, 00010100H    ;Zaehler 0, nur Low schr., Mode 2, Bin. zaehlen
OUT 43H,AL            ;Steuerwort in Steuerregister schreiben
MOV AL,3              ;Zaehlwert
OUT 40H,AL            ;in Zaehler Nr. 0 laden
```

# 8253: Mode 3

## Rechteckgenerator<sub>1</sub>

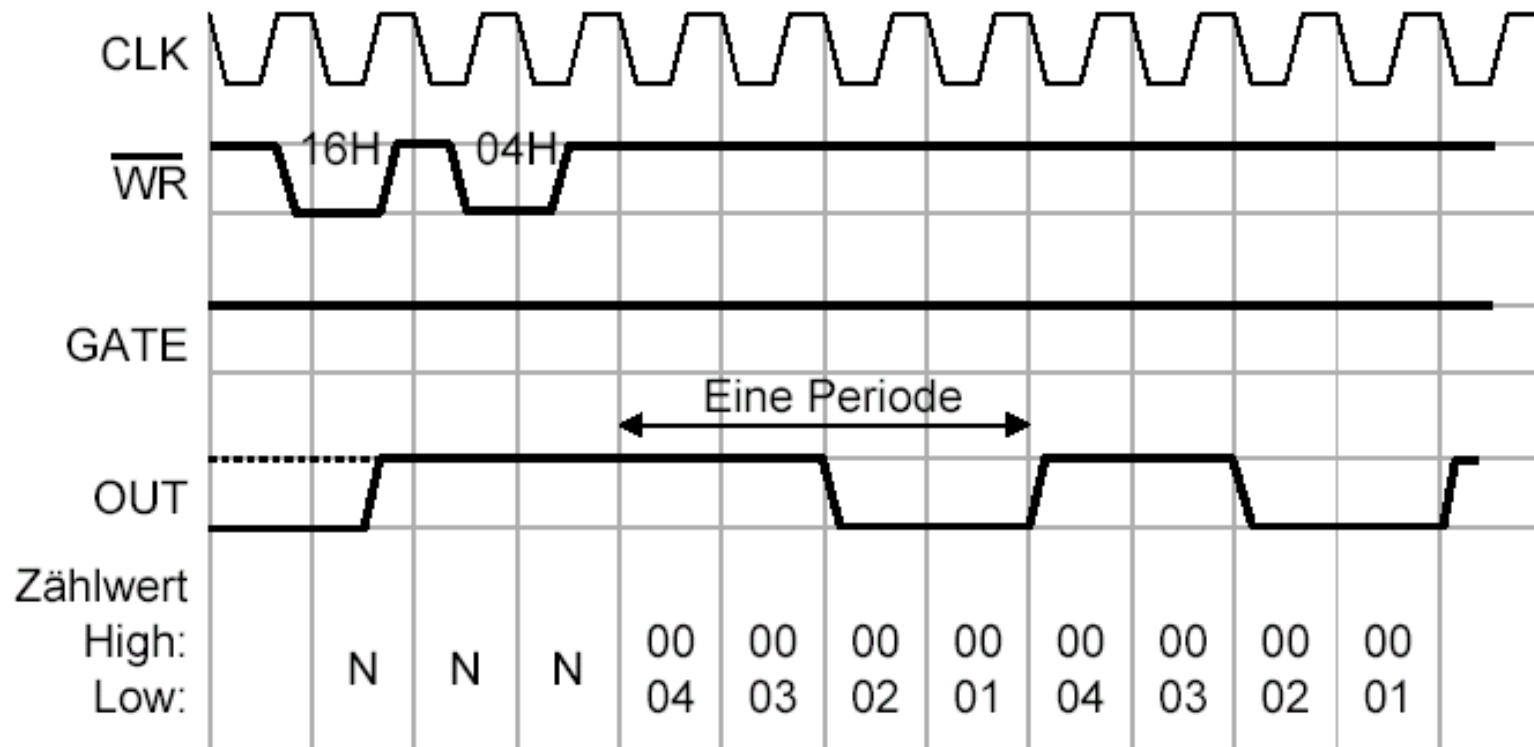
### Rechteckgenerator

Diese Betriebsart ist ähnlich wie Betriebsart 2 mit dem Unterschied, dass der Zählerausgang solange auf H-Pegel bleibt, bis der halbe Zähleranfangswert abgezählt wurde (gilt für geradzahlige Startwerte). Anschließend geht der Ausgang auf L-Pegel, bis die andere Hälfte abgezählt ist.

Bei ungeradzahligen Zähleranfangswerten  $N$  ist der Ausgang für  $(N + 1) / 2$  Eingangstakte auf H-Pegel und für  $(N - 1) / 2$  Takte auf L-Pegel.

Wird der Zähler während des Zählens mit einem neuen Anfangswert geladen, wird nach dem nächsten Wechsel des Ausgangssignals mit dem neuen Wert weitergearbeitet.

# 8253: Mode 3 Rechteckgenerator<sub>2</sub>



**Beispiel:** Zähler 0, Zählwert 4, Mode 3, nur Low-Wert genutzt, binär zählen

```
MOV AL, 00010110H    ;Zaehler 0, nur Low schr., Mode 3, Bin. zaehlen
OUT 43H,AL            ;Steuerwort in Steuerregister schreiben
MOV AL,4              ;Zaehlwert
OUT 40H,AL            ;in Zaehler Nr. 0 laden
```

# 8253: Mode 4

## Software getriggerte Impuls<sub>1</sub>

### Software gesteuertes Signal (Strobe)

Nach der Ausgabe des Steuerwortes für diese Betriebsart liegt der Zählerausgang auf H-Pegel. Nach dem Laden des Zähleranfangswertes startet der Zähler. Beim Erreichen des Endstandes 0 liegt am Ausgang für eine Taktperiode L-Pegel und anschließend dauerhaft wieder H-Pegel an.

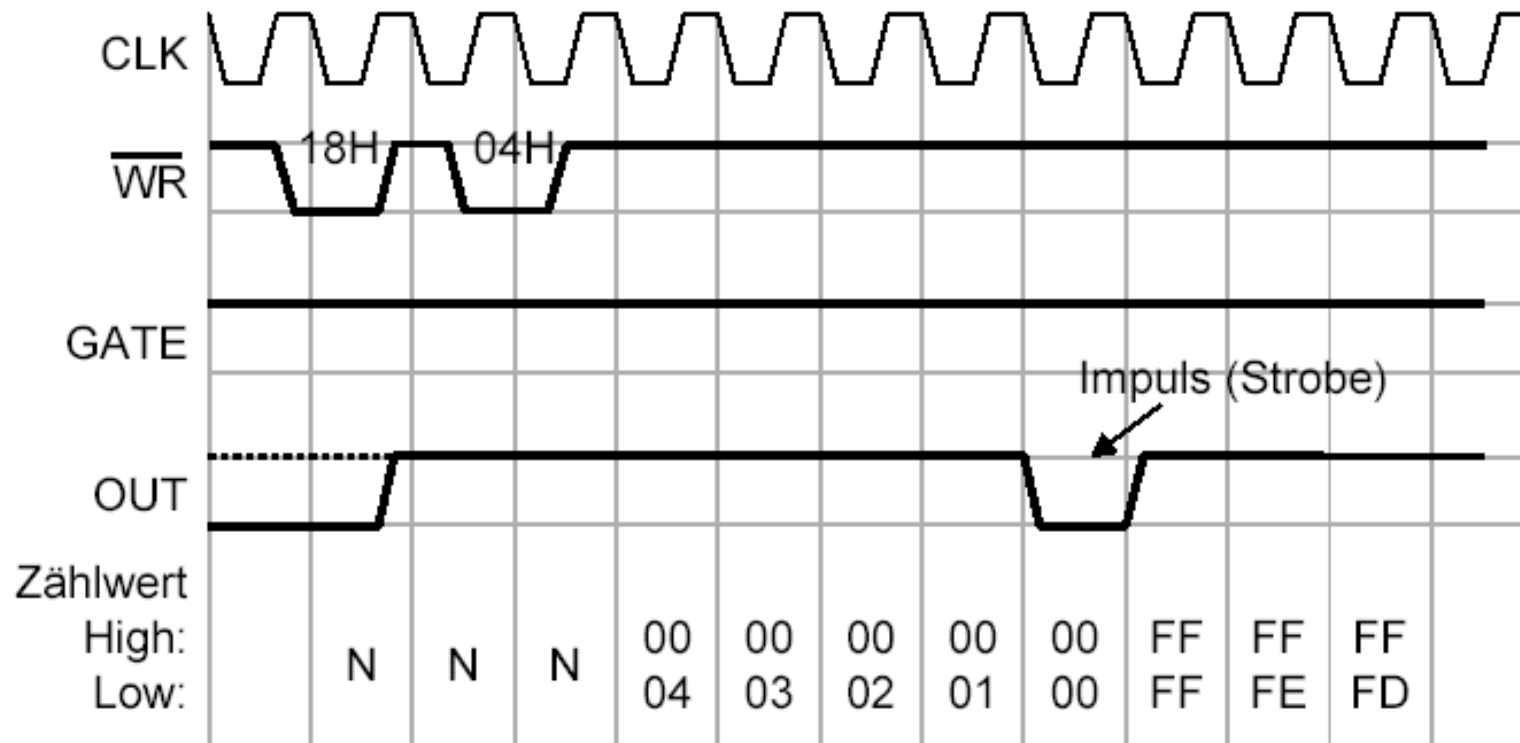
Wird während eines Zählerdurchlaufs ein neuer Zähleranfangswert eingegeben, zählt der Zähler sofort nach beendeter Eingabe mit dem neuen Wert weiter. Wird der Zähler nach Beendigung eines Zählerdurchlaufs neu geladen, startet er ebenfalls nach vollendeter Eingabe mit dem neuen Wert.

Mit einem L-Pegel am GATE-Eingang kann das Zählen unterbrochen werden.



# 8253: Mode 4

## Software getriggelter Impuls<sub>2</sub>



**Beispiel:** Zähler 0, Zählwert 4, Mode 4, nur Low-Wert genutzt, binär zählen

```
MOV AL, 00011000H    ;Zaehler 0, nur Low schr., Mode 4, Bin. zaehlen
OUT 43H,AL           ;Steuerwort in Steuerregister schreiben
MOV AL,4             ;Zaehlwert
OUT 40H,AL           ;in Zaehler Nr. 0 laden
```

# 8253: Mode 5

## Hardware getriggertter Impuls<sub>1</sub>

---

### Hardware gesteuertes Signal (Strobe)

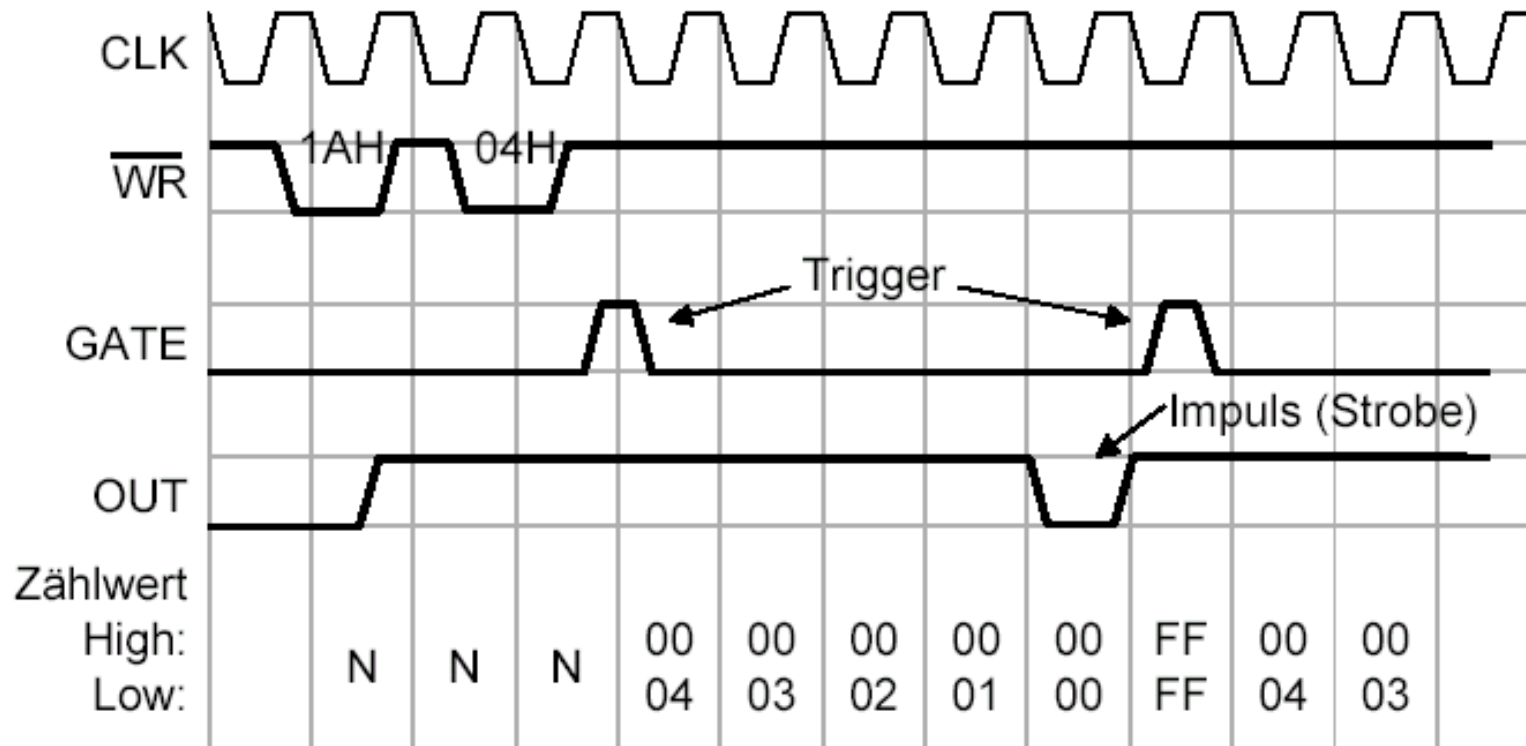
Der Zähler fängt mit der ansteigenden Flanke eines Triggersignals auf den GATE-Eingang zu zählen an und erzeugt am Ausgang, bei Erreichen des Endwertes (0), für die Länge einer Taktperiode L-Pegel.

Der Zähler ist beliebig oft neu triggerbar.

Der Ausgang wird dabei solange nicht auf L-Pegel gehen, bevor nicht der, mit der positiven Flanke des Triggersignals gestartete Zählerdurchlauf beendet ist.

# 8253: Mode 5

## Hardware getriggelter Impuls<sub>2</sub>



**Beispiel:** Zähler 0, Zählwert 4, Mode 5, nur Low-Wert genutzt, binär zählen

```
MOV AL, 00011010H    ;Zaehler 0, nur Low schr., Mode 4, Bin. zaehlen
OUT 43H,AL            ;Steuerwort in Steuerregister schreiben
MOV AL,4              ;Zaehlwert
OUT 40H,AL            ;in Zaehler Nr. 0 laden
```

# Funktionsübersicht GATE-Signal

Betriebsart	Gate-Signal		
	Low oder High -> Low	Low -> High	High
0	sperrt Zähler		gibt Zähler frei
1		1. startet Zähler 2. OUT nach 1 CLK auf Low-Pegel	
2	1. sperrt Zähler 2. OUT sofort auf High-Pegel	startet Zähler	gibt Zähler frei
3	1. sperrt Zähler 2. OUT sofort auf High	startet Zähler	gibt Zähler frei
4	sperrt Zähler		gibt Zähler frei
5		startet Zähler	