

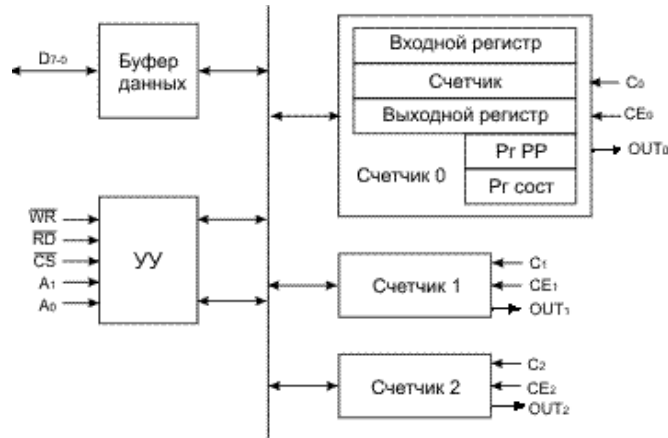
Программируемый интервальный таймер

Микросхема 8253/8254 Intel. Российский аналог: К580ВИ53, К1810ВИ54.

Программируемый таймер предназначен для получения программно управляемых временных задержек и генерации времязадающих функций. Таймер позволяет повысить эффективность программирования процессов управления и синхронизации внешних устройств. Программируемые интервальные таймеры позволяют формировать заданные интервалы времени, независимые от быстродействия вычислительной системы, и являются одним из средств обеспечения работы ПЭВМ в реальном масштабе времени и выполняют:

- измерение временных задержек между событиями,
- подсчет числа внешних событий, прерывание процессора по достижении их запрограммированного числа,
- управление процедурой включения/выключения мотора устройств,
- программирование скоростей передачи данных между,
- разделение времени между программами.

Структурная схема



Таймер состоит из буфера данных, устройства управления и аппаратуры каналов (трех независимых 16-битных счетчиков).

Буфер данных – 8-битный, двунаправленный, z-состоянием. Предназначен для сопряжения внутренней шины данных таймера с системной шиной ЭВМ. Данные принимаются или передаются через буфер данных при выполнении команд чтения или записи.

Устройство управления принимает входные сигналы с системой шины и генерирует управляющие сигналы для других функциональных блоков системы таймера.

Назначение выводов таймера:

C – входной сигнал счетных импульсов для каждого счетчика;

CE – входной сигнал разрешения счета для каждого счетчика;

OUT - однобитный выходной сигнал для каждого счетчика;

RD - управляющий сигнал чтения;

WR - управляющий сигнал записи;

CS - выбор кристалла;

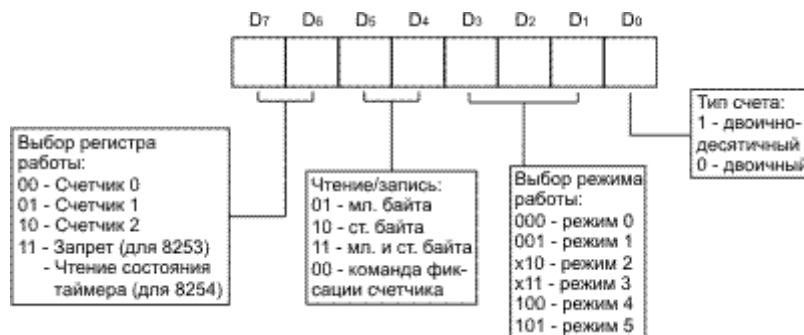
A0, A1 - выбор счетчиков или управляющего слова для чтения или записи:

A1	A0	Регистр
0	0	Счетчик 0
0	1	Счетчик 1
1	0	Счетчик 2
1	1	Регистр режима работы

Счетчики вычитающие – под воздействием счетных импульсов состояние счетчика уменьшается на 1. Они могут считать в двоичном коде: ffffh-0000h или в двоично-десятичном коде: 9999h – 0000h.

Управляющее слово режима работы

Инициализация и управление работой счетчиков осуществляется с помощью управляющего слова режима работы.



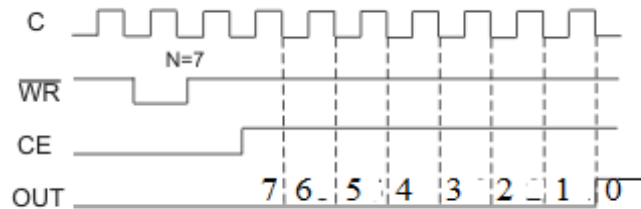
Функции сигнала разрешения счета

Режим работы	Состояние СЕ		
	"0" или спад	Фронт	"1"
0. Программируемая задержка	Запрещает счет	-	Разрешает счет
1. Программируемый одновибратор	-	Запускает счет сначала. Устанавливает в след. такте OUT=0	-
2. Делитель частоты	Запрещает счет. Немедленно устанавливает OUT=1	Запускает счет сначала	Разрешает счет
3. Генератор меандра	Запрещает счет. Немедленно устанавливает OUT=1	Запускает счет сначала	Разрешает счет
4. Одиночный программно-формируемый строб	Запрещает счет	-	Разрешает счет
5. Одиночный аппаратно формируемый строб	-	Запускает счет сначала	-

Режим 0 - программируемая задержка

Сигнал $OUT_{исх} = 0$ (исходный сигнал).

После подачи сигнала разрешения счета через интервал времени $N \cdot T$, где T – период следования входных сигналов, а N – число загруженное в счетчик, формируется $OUT = 1$



Единичное значение сигнала OUT сохраняется до загрузки управляющего слова режима работы или до загрузки значения в счетчик.

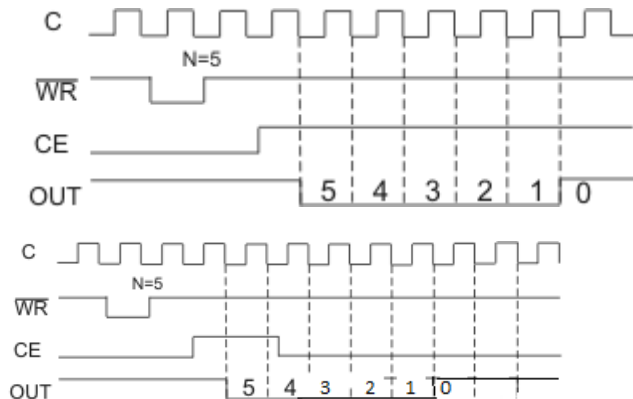
Этот режим не перезагружается. При записи значения в счетчик оно сразу попадает в счетчик, не фиксируясь во входном регистре.

Снятие сигнала CE (сигнала разрешения счета) приостанавливает счет.

Загрузка нового значения (N) во время счета приостанавливает счет на время загрузки, а затем счет начинается с нового значения N .

Режим 1 - Программируемый одновибратор

Выходной сигнал в исходном состоянии равен 1 ($OUT_{исх} = 1$). После подачи фронта сигнала разрешения счета формируется выход $OUT = 0$ длительностью $N \cdot T$. После этого выходной сигнал вновь возвращается в 1 ($OUT = 1$)



Режим 1 перезагружаемый.

Значение N записывается и хранится во входном регистре счетчика. Каждый фронт сигнала CE запускает счет сначала.

Загрузка нового значения в счетчик на текущий цикл счета не влияет, оно будет использоваться в следующем цикле счета (по входу следующего фронта CE).

Режим 2 - Делитель частоты

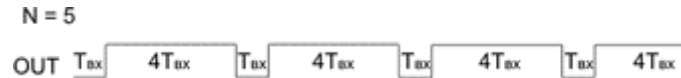
Выходная частота: $f_{\text{ВЫХ}} = f_{\text{ВХ}}/N$.

Период выходного сигнала: $T_{\text{ВЫХ}} = N \cdot T_{\text{ВХ}}$.

OUT = 1, для $T_1 = (N-1) \cdot T_{\text{ВХ}}$.

OUT = 0, для $T_0 = T_{\text{ВХ}}$.

$T_0 + T_1 = T_{\text{ВЫХ}}$.



Режим 3 - Генератор меандра

Выходная частота: $f_{\text{ВЫХ}} = f_{\text{ВХ}}/N$.

Период выходного сигнала: $T_{\text{ВЫХ}} = N \cdot T_{\text{ВХ}}$.

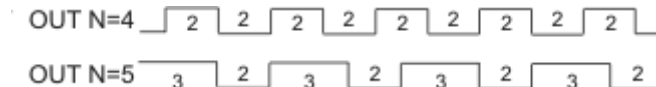
Если N - четное, то

$$T_0 = T_1 = (N \cdot T_{\text{ВЫХ}})/2.$$

Если N - нечетное, то

$$T_0 = ((N-1) \cdot T_{\text{ВЫХ}})/2,$$

$$T_1 = ((N+1) \cdot T_{\text{ВЫХ}})/2.$$

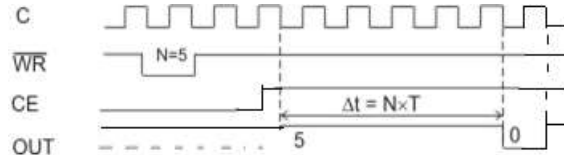


Режим 4 - Одиночный программно формируемый строб (импульс)

Сигнал $OUT_{исх} = 1$ (исходный сигнал).

После подачи сигнала разрешения счета $CE = 1$ через интервал длительностью $\Delta t = N \times T$ формируется импульс $OUT = 0$ длительностью T .

После $OUT = 1$.



Этот режим очень похож на режим 0 (не по выходному сигналу, а по режиму счета). Загрузка начального значения осуществляется сразу в счетчик (как и в режиме 0). Влияние сигнала CE и перезагрузки счетчика в режиме счета такой же, как в режиме 0.

Режим 5 - Одиночный аппаратно формируемый строб (импульс)

Сигнал OUT формируется так же, как в режиме 4. Влияние сигнала режима счета и перезагрузка счета в режиме счета действует как в режиме 1.

Режим перезагружаемый.

Программирование таймера

Каждый из трех счетчиков таймера программируется независимо друг от друга (но можно подать выходной сигнал одного счетчика на вход другого).

При программировании в счетчик заносят управляющее слово режима работы. Начальное значение может задаваться одним байтом (младшим или старшим) или двумя (сначала младший, затем старший).

```
    cw    equ 00110000b
    ...
    mov    al, cw          ; номер счетчика в управляющем слове
    out    cwReg, al
    mov    cx, N
    out    CNT0, al        ; по адресу счетчика 0 выдаем младший байт
    xchg    al, ah
    out    CNT0, al        ; по адресу счетчика 0 выдаем старший байт
```

Чтение содержимого счетчика

1 способ. Приостановка работы счетчика

Для обеспечения стабильных показаний необходимо приостановить работу канала либо подачей сигнала низкого уровня на вход CE (кроме режима 1), либо блокированием тактовых импульсов на вход C.

Пример: Работа счетчика приостанавливается и читается содержимое счетчика одной или двумя командами ввода.

```
in      al, CNT0
xchg    ah, al
in      al, CNT0
xchg    ah, al
```

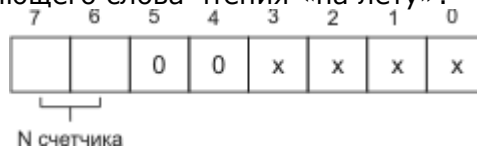
т.е. в AH будет слово, которое соответствует содержимому счетчика.

Но приостановить работу счетчика не всегда возможно.

2 способ. Чтение «на лету»

По адресу регистра управляющего слова выдается специальная команда чтения на лету, по которой заданный счетчик защелкивается в выходном регистре и выходное значение регистра не меняется до операции чтения или перепрограммирования счетчика.

Формат управляющего слова чтения «на лету»:



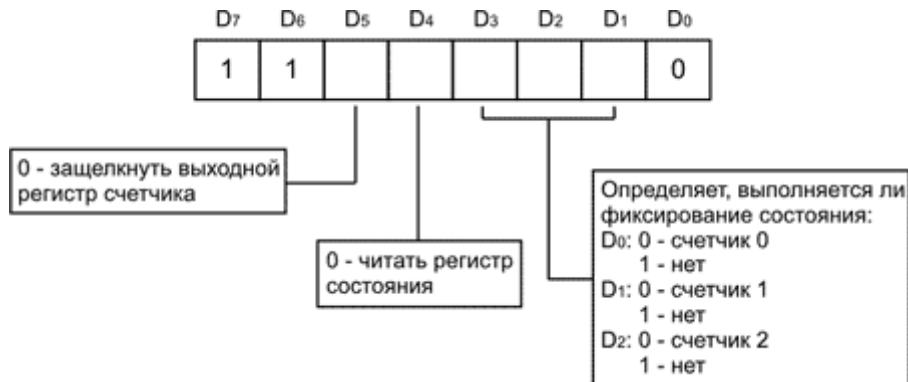
```
cwrд    equ 0h
...
mov     al, cwrд
out     CWreg, al
in      al, CNT0
xchg    ah, al
in      al, CNT0
xchg    ah, al
```

3 способ. Чтение состояния таймера

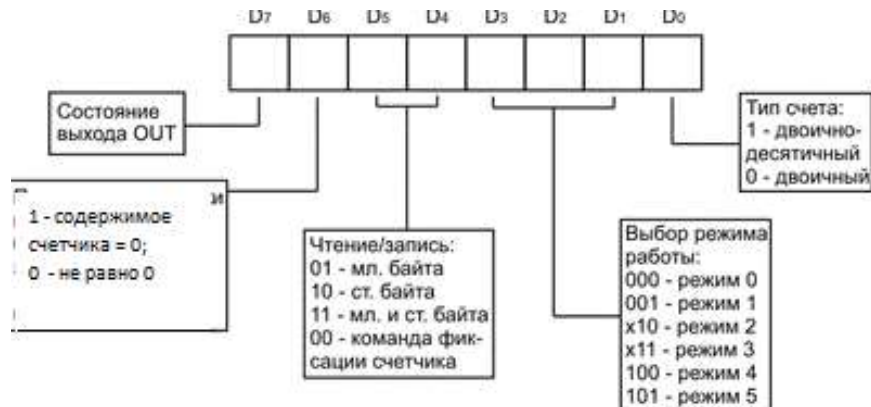
Только для схемы 8254.

Управляющее слово действует сразу на 3 счетчика.

Формат управляющего слова чтения состояния следующий:



Формат регистра состояния



После выполнения команды чтение состояния таймера для заданных счетчиков:

1. Запоминается состояние

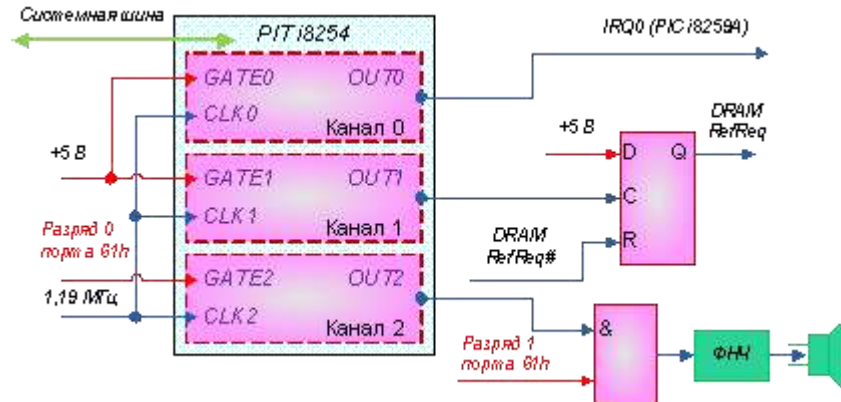
и / или

2. Защелкивается содержимое выходных регистров.

Эти регистры остаются неизменными до выполнения операции чтения или перепрограммирования счетчика.

```
cwrstatus      equ 11001100b; счетчик0 зафиксировать
...
mov     al, cwrstatus
out     CWreg, al
in  al, CNT0      ; это регистр состояния
                ; нужно сохранить
in      al, CNT0; младший байт
xchg    ah, al
in      al, CNT0
xchg    ah, al; теперь будет старший байт
```

Назначение счетчиков таймера в IBM PC



В IBM PC счетчики таймера имеют следующее назначение:

Счетчик	Назначение	Режим, начальное значение
0	Прерывание по таймеру (IRQ ₀)	3, счетчик=0 (65536)
1	Запрос для канала 0 ПДП (регенерация памяти)	2, счетчик=18
2	Генератор звука динамика	3

В IBM PC таймер имеет базовый адрес 40h и следующие программируемые регистры:

Адрес	Операция	Назначение
40h	Запись	Загрузка счетчика Сч0
	Чтение	Чтение счетчика Сч0
41h	Запись	Загрузка счетчика Сч1
	Чтение	Чтение счетчика Сч1
42h	Запись	Загрузка счетчика Сч2
	Чтение	Чтение счетчика Сч2
43h	Запись	Запись управляющего слова в регистр режима работ

Тактовая частота каждого канала равна 1,19318 МГц, т. е. каждый такт имеет длительность 0,84 мсек. Вход СЕ счетчиков 0 и 1 всегда имеют высокий уровень, поэтому счет в этих счетчиках разрешен всегда. Вход СЕ счетчика 2 управляется битом 0 порта 61h.

При начальной загрузке BIOS инициализирует счетчик 0 для работы в режиме 3 с начальным значением 0 (т. е. 65536 декрементов на цикл счета). Поэтому частота прерывания по таймеру равна $1,19 \text{ МГц} / 65536 = 18,2 \text{ Гц}$ и прерывание IRQ0, связанное с вектором Int 8, происходит 18,2 раз в секунду, т. е. каждые 55 мс.

Можно перепрограммировать канал 0, но тогда

- BIOS не сможет отслеживать текущее время и дату;
- нарушится работа с гибкими дисками, т. к. включение и выключение их двигателей отсчитывается по текущему времени.

Счетчик 1 работает в режиме 2 с начальным значением 18, поэтому регенерация памяти происходит каждые 18 мс. Перепрограммировать его нельзя, т. к. это приведет к потере данных в ОЗУ.

```
CW0 equ 00110110b
N0 equ 0
CW1 equ 01010100b
N1 equ 18
CW2 equ 10xx0110b
N2 equ xx
```

Динамик подключается к счетчику 2. Счетчик 2 настраивается на режим 3. Загружается старший байт или младший байт в зависимости от того, сигнал какой частоты хотим получить. Значение счетчика 2 вычисляется по формуле $N = 1193181/f$ (1193181 – входная тактовая частота таймера в Гц, f – требуемая частота звука).

Пример. Написать фрагмент программы включения динамика ПК на частоте ноты ре 1-ой октавы (587,3 Гц).

$f_{вх} = 1,193180$ МГц, $T = 0,84$ мс.

$N = f_{вх}/f_{вых} = 2031$

```
mov al, 10110110b
out 43h, al
mov ax, 2013
out 42h, al; младший байт
mov al, ah; старший байт
out 42h, al
in al, 61h
or al, 11b
out 61h, al
```