# **Лекция №3**

**3. Программируемый интервальный таймер I8253**

# 3.1 Принцип работы таймера

БИС программируемого таймера i8253 предназначена для организации работы микропроцессорных систем в режиме реального времени и позволяет формировать сигналы с различными временными и частотными характеристиками.

Микросхема являет собой однокристальный трехканальный программируемое устройство, предназначенное для получения программно-управляемых временных задержек и выполнения заданных временных функций в микропроцессорных системах.

БИС содержит три канала. В каждом канале есть регистр управляющего слова, 16-разрядный программируемый счетчик, работающий в двоичном или двоично-десятичном коде (реверсивный), а также двух байтный буферный регистр, в который по специальной команде переписываются константы счетчика.

Программирование каждого канала таймера сводится к следующим операций:

-запись в регистр индивидуального управляющего слова управления каналом;

-запись в 16-разрядный программируемый счетчик необходимой константы пересчета;

Управление периферийными устройствами часто требует от микропроцессорной системы точного задания временных интервалов между управляющими сигналами. Сделать это программно зачастую невозможно из-за сложного временного расписания обмена данными внутри микропроцессорной системы. К тому же вставка тактов ожидания в программу тормозит работу процессора. Такие задачи поручают программируемым таймерам (ПТ). С помощью ПТ процессор может формировать временные интервалы произвольной длительности, производить синхронизацию внешних устройств, организовывать счетчики событий, вести счет текущего времени и т. д.

Программируемый таймер КР580ВИ53 содержит три независимых счетчика с максимальной частотой счета 2 МГц. Каждый из трех счетчиков может быть запрограммирован на различные режимы работы и счета (двоичный или двоично-десятичный).

Таймер КР580ВИ53 может работать как:

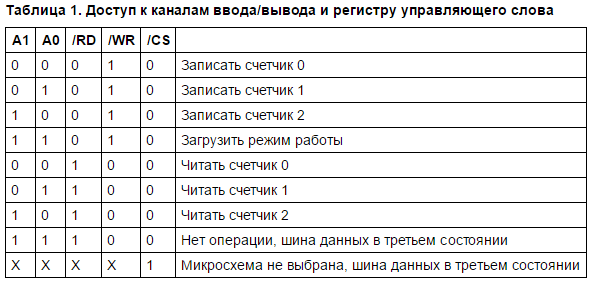
* программируемый тактовый генератор;
* счетчик событий;
* бинарный перемножитель;
* цифровой одновибратор;
* часы реального времени.

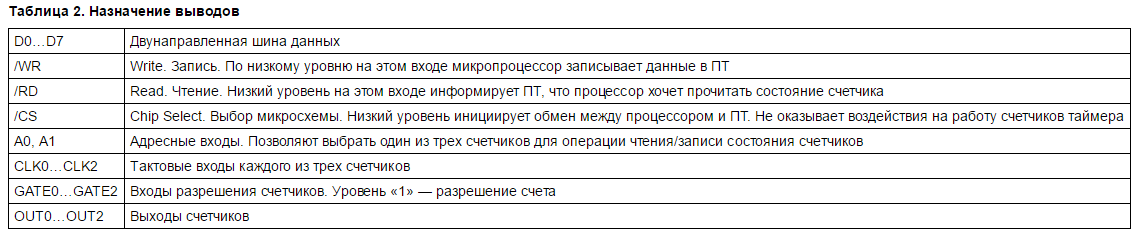
Запись индивидуального слова управления каналом осуществляется по единой для всех каналов адресу (А0 = 1, А1 = 1) регистра управляющего слова.

Указание конкретного канала, к которому относится управляющее слово, размещается в самом управляющему слове.

**Адресация регистров таймера**

|  |  |  |
| --- | --- | --- |
| А0 | А1 | Найменування |
| 0  0  1  1 | 0  1  0  1 | Счетчик канала 0  Счетчик канала 1  Счетчик канала 2  Регистр Управляющего слова |





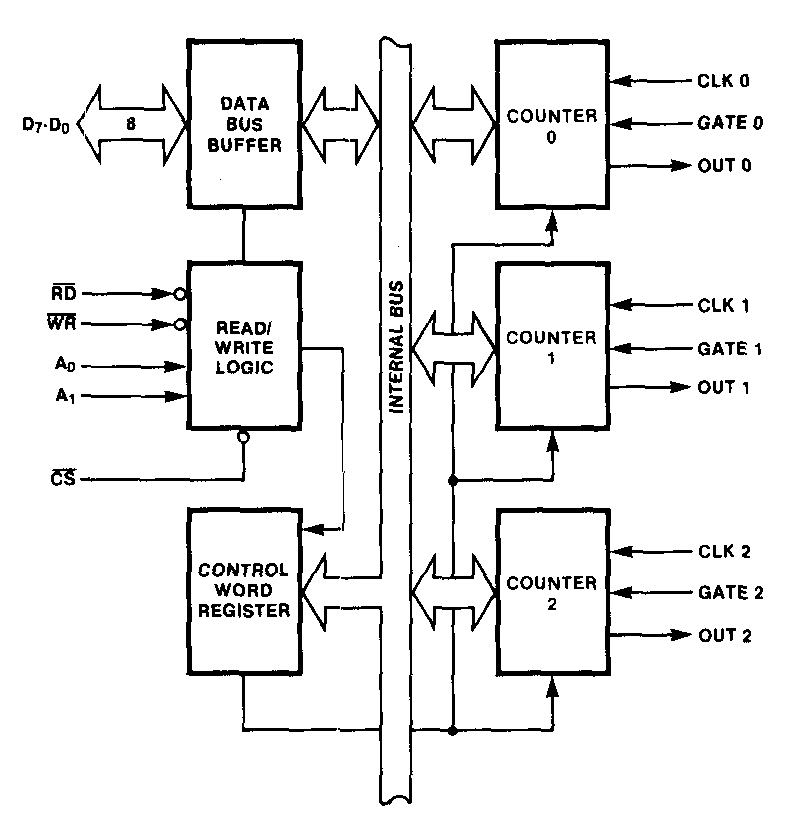
В состав ПТ входит буфер канала данных, логика чтения/записи, определяющая, к какому каналу обращается процессор, и трех независимых каналов. Каждый канал включает в себя 16-разрядный счетчик, регистр режима, схему управления и схему синхронизации.

В состав счетчика входят регистр хранения, буферный регистр и собственно счетчик. Регистр хранения содержит значение константы счета. В начале цикла работы канала константа счета из регистра хранения переписывается в счетчик, и затем по тактовым импульсам на входе CLK происходит декремент содержимого счетчика. Содержимое счетчика в любой момент времени может быть переписано в буферный регистр и прочитано процессором. В регистр режима записывается управляющее слово, определяющее режим работы канала. Схема управления синхронизирует работу отдельных схем в соответствии с запрограммированным режимом работы и работу канала с работой процессора. Схема синхронизации формирует серию внутренних импульсов с длительностью, определяемой внутренними времязадающими цепями, и периодом, равным периоду внешних тактовых импульсов.

# 3.2 Блок схема таймера

Упрощенная структурная схема ПО приведена на рис.3.1. В состав БИС входят:

* Буфер данных (BD), предназначенный для обмена данными и управляющими словами между МП и ПО;
* Схема управления чтением-записью (RWCU), что обеспечивает выполнение операций ввода-вывода информации в ПО;
* Регистр управляющего слова (RGR), предназначенный для записи управляющих слов, задающие режимы работы счетчиков;
* счетчик каналов (ctr0-ctr2).



**DBB - буфер данных**

**RWL - схема управления чтением-записью**

**CWR - регистр управляющего слова**

**Ст0-СТ2 - счетчик каналов**

**D7-D0- линии шины данных**

**А0-А1- линии адреса**

**/ RD, / WR- сигналы чтения-записи**

**/ CS- сигнал выбора микросхемы**

**CLK0, CLK1, CLK2-счетные входы**

**OUT0, OUT1, OUT2-выходы счетчиков**

**GATE0, GATE1, GATE2 входы разрешения-запрещения счета и аппаратного запуска счета**

Рисунок 3.1-Упрощенная структурная схема таймера

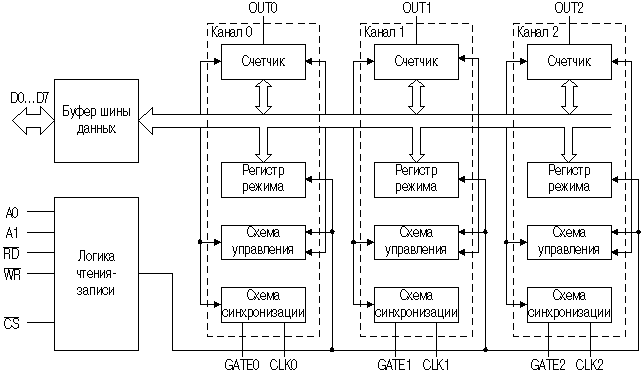
Подключение ПТ к шинам микропроцессора показано на рисунке 3.2.



Рисунок 3.2-подключение к шинам системы

Условное графическое изображение i8253 представлено на рис. 3.3.

Рис 3.3-Графическое изображение таймера



# 3.3 Принцип работы

БИС i8253 / 54 содержит 3, 16 разрядных двоично-десятичных регистра с предыдущим заполнением Функциональные конфигурация его входов и выходов устанавливается с помощью средств программного обеспечения.

Счетчики полностью независимые. Каждый может иметь свой режим работы и тип цифры (двоичный или двоично-десятичный). Загрузка счетчиков начальным значением счета определяется программным способом.

Считывания содержания каждого счетчика осуществляется с помощью команды управления. Таймер также содержит дополнительное оборудование и команду, которые позволяют читать содержание счетчика "на лету", не мешая его работе.

Программирование таймера происходит с помощью команды управления. Каждый счетчик индивидуально настраивается в соответствии содержания регистра управляющего слова.

00-канал 0

01-канал 1

10-канал 2

11-код команды RBC

00-код команды CLS

01-чтения \ зап мл.байта к.

10-чтения \ зап ст.байту к.

11-чтения \ зап мл. а затем ст. байта к.

000-режим 0

001-режим 1

010-режим 2

011-режим 3

100-режим 4

101-режим 5

0-двоичный

1ДФ. \ Десятичный счет

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **SC1** | **SC0** | **RW1** | **RW0** | **M2** | **M1** | **M0** | **BCD** |

Рисунок 3.4 -Формат команды управления

# 3.4 Режимы роботы таймера

Таймер имеет 6 режимов работы:

- Режим «0» - формирование сигнала по истечении интервала времени;

- Режим «1» формирования импульса заданной длительности;

- Режим «2» генератор импульсов нужной частоты;

- Режим «3» программируемый генератор прямоугольных импульсов;

- Режим «4» формирования импульса по окончании заданного времени;

- Режим «5» счетчик событий.

Режимы предусматривают изменение выхода таймера OUT с низкого уровня OUT = 0 до высокого OUT = 1 после окончания счета (достижения значение 0 константы пересчета).

**Режим 0. Формирование сигнала по истечению заданного интервала времени**

После записи с помощью сигнала / WR управляющего байта СWR, который переводит соответствующий канал таймера в режим 0 выход таймера OUT устанавливается значение логической «1» (рис .1.6 а).

После записи константы пересчета (например, N = 4), канал N начинает работу (рис 1.6, б) при которой константа пересчете уменьшается по тактам входных импульсов CLK, выход OUT устанавливается в 0.

Когда константа пересчете достигнет нулевого значения, выход таймера устанавливается в высокий уровень OUT = 1 (рис.1.6, б).

Высокий уровень входа GATE = 1 разрешает работу счетчика, а GATE = 0 - приостанавливает его работу. Таким образом, при выдаче сигнала по окончании счета зависит от введенной константы пересчета и может увеличиваться на продолжительность Δt с помощью сигнала керуванняGATE (рис.1.6, в).

Из рисунка видно, что в режиме 0 сигнал

GATE 1 -дозволяеличбу.

Переход GATE из 1 в 0 -призупиняе математике

Переход GATE 0 в 1 -позволяет продолжение счета.

**Режим 1. Формирование импульса заданной длительности (строб)**

После записи с помощью сигнала / WR управляющего байта СWR, который переводит соответствующий канал таймера в режим 1. Выход таймера OUT устанавливается в 1 (рис.1.7).

После загрузки константы пересчета (N = 4) таймер начинает работу и формируется интервал времени Δt строб в зависимости от константы пересчета, а выход таймера устанавливается на низком уровне OUT = 0 (рис.1.7, б).

Высокий уровень входа GATE = 1, а также переход его на низком уровне и его содержание, позволяет работу счетчика, а переход сигнала GATE с низкого уровня в высокий, приводит к перезагрузке константы пересчета счетчика на начальное значение, что позволяет изменять интервал строба (рис. 1.7, в).

Из рисунка видно, что в режиме 1 сигнал

GATE 1 -позволяет математике

Переход GATE из 1 в 0 -позволяет математике

Переход GATE с 0 в 1 -перезавантаження константы пересчета.

**Режим 2. Программируемый генератор импульсов нужной частоты**

После записи с помощью сигнала / WR управляющего байта СWR, который переводит соответствующий канал таймера в режим 2. Выход таймера OUT устанавливается в 1 (рис.1.8).

После загрузки константы пересчета (N = 4) таймер начинает работу.

Когда константа пересчете достигнет значения 1, выход таймера устанавливается на низком уровне OUT = 0 и держится в этом состоянии пока константа пересчете достигнет нулевого значения, после чего выход таймера принимает значение 1 (OUT = 1) и происходит аппаратное перезагрузки константы пересчете на исходное значение N = 4 (рис.1.8, б).

Высокий уровень GATE = 1, а также переход сигнала GATE с высокого уровня в низкий и его содержание, позволяет работу таймера, а переход сигнала GATE с низкого уровня в высокий, приводит к перезагрузке константы пересчета счетчика на начальное значение, что позволяет изменять частоту импульсов ( рис.1.8, в).

Из рисунка видно, что в режиме 2 сигнал

GATE 1 -позволяет математике

GATE 0 -зупиняе математике

Переход GATE с 0 в 1 -перезавантаження константы пересчета.

**Режим 3. Программируемый генератор прямоугольных импульсов**

После записи управляющего байта СWR, соответствующий канал таймера настраивается на режим 3. Выход таймера OUT устанавливается в 1 (рис.1.9).

После загрузки константы пересчета таймер начинает работу и уровень логической единицы на выходе таймера OUT = 1 будет держаться до половины значение константы пересчета N / 2 (если N четное), после этого устанавливается логический 0, который содержится до достижения нулевого значения константы перерахункуN (рис. 1.9, б), после чего происходит автоматическая перезагрузка константы пересчете на исходное значение и таймер начинает свою работу сначала. Таким образом, на выходе таймера формируется сигнал прямоугольной формы, в котором продолжительность логического нуля равна продолжительности логической единицы (меандр, если Nпарне). Если число N нечетное то на выходе таймера логическая единица будет в течение (n + 1) / 2 и логический ноль течение (n-1) / 2.

Высокий уровень GATE = 1 разрешает работу таймера, а низкий уровень GATE = 0 приостанавливает, а переход сигнала GATE с 0 в 1 перезагружает константу пересчете на исходное значение и позволяет работу таймера (рис.1.9, в).

Из рисунка видно, что в режиме 3 сигнал

GATE 1 -позволяет математике

GATE 0 -призупиняе математике

ПерехидGATE из 0 в 1 -перезавантаження константы пересчета.

**Режим 4. Формирование импульса по окончанию заданного времени**

После записи с помощью сигнала / WR управляющего байта СWR, который переводит соответствующий канал таймера в режим 4. Выход таймера OUT устанавливается в 1 (рис.1.10).

После записи константы пересчета (N = 4) таймер начинает работу.

При достижении константы N = 0 состояние выхода таймера OUT устанавливается в 0 на один период CLK, после чего OUT снова устанавливается в 1 (т.е. через период времени (n + 1) Tсlk) OUT = 0 на период одного такта (рис.1.10, б ).

Высокий уровень входа GATE = 1 разрешает работу счетчика, GATE = 0 - приостанавливает его работу, а переход сигнала GATE с низкого уровня в высокий, приводит к продолжению счета (рис.1.10, в).

Из рисунка видно, что в режиме 4 сигнал

GATE 1 -позволяет математике

Переход GATE из 1 в 0 -призупиняе математике

Переход GATE 0 в 1 -позволяет продолжение счета.

**Режим 5. Счетчик событий**

После записи с помощью сигнала / WR управляющего байта СWR, который переводит соответствующий канал таймера в режим 5. Выход таймера OUT устанавливается в 1 (рис.1.11).

После записи константы пересчета (N = 4) таймер начинает работу.

При достижении константы N = 0 состояние выхода таймера OUT устанавливается в 0 на один период CLK, после чего OUT снова устанавливается в 1 (т.е. через период времени (n + 1) Tсlk) OUT = 0 на период одного такта (рис.1.11, б ).

Высокий уровень входа GATE = 1, а также переход в 0 и его содержание, позволяет работу счетчика, а переход сигнала GATE с низкого уровня в высокий, приводит к перезагрузке константы пересчета счетчика на начальное значение, что позволяет изменять время выдачи импульса (рис.1.11 в).

Из рисунка видно, что в режиме 5 сигнал

GATE 1 -позволяет математике

Переход GATE из 1 в 0 -позволяет математике

Переход GATE с 0 в 1 -перезавантаження константы пересчета.

Управляющие слова могут быть записаны в ПО в произвольном порядке. В любой последующий момент времени записуютьсяпочатковикодиличильникиву видповидностизизначеннямирозрядивD5, D4 керуючихслив.

Исходя из особенности влияния сигнала GATE на работу счетчика канала интервального таймера, режимы его работы можно разделить на две группы:

- С приостановке (паузой) по сигналу GATE (режим 0,3,4)

- С перезагрузкой константы по сигналу GATE (режим 1,2,5).













# 3.6 Программирование таймера

Программирование таймера достаточно гибкое, не имеет значения последовательность загрузки управляющих слов режима отдельных счетчиков, то есть не обязательно должно быть первым управляющее слово счетчика 0, последнее-2. Регистр управляющего слова режима каждого счетчика имеет свой адрес и может загружаться независимо. Однако загрузка содержания счетчика должно соответствовать последовательности, запрограммированный в управляющем слове режима (младший а затем старший байт).

При программировании таймера программист решает следующие задачи:

1. Инициализация каналов интервального таймера.

Осуществляется с помощью:

- Записи управляющего слова CWR,

- Записи константы пересчета.

2.-Чтение регистра состояния интервального таймера, или константы пересчета RBC (ReadBackCommand).

Осуществляется с помощью:

- Записи управляющего байта RBC,

- Чтение регистра состояния инициализации таймера RST,

- Чтение константы пересчета

а) с остановкой таймера

б) без остановки таймера.

Управляющее слово задает один из шести режимов работы, тип счета (двоичный или двоично-десятичный), порядок загрузки и размерность (один или два байта) константы. Оно загружается в регистры RSW каналов таймера, сохраняется до следующего перепрограммирования. После этого загружаются константы, как и принимают значения:

- Двоичный - 0-65536

- Двоично-десятичный - 0-9999

Константы пересчета загружаются в таймер при А0, А1 (00, 01, 10).

Существует 2 варианта загрузки: сначала все слова управления, а затем константы, или слово управления и константу для каждого канала последовательно.

# 3.7 Чтение содержания счетчика

Если нужно знать значение счета в процессе работы (особенно это используется при определении количества событий), то возможно использовать два средства чтения, которые не повреждают информацию:

1) Используя команду ввода из выбранного счетчика (READ), управляя входами А0-А1 программист может выбирать счетчик, значение которого нужно знать. При чтении значения счетчика надо запретить работу счетчика с помощью команды GATE или путем запрета тактовых импульсов. При чтении содержания счетчика первым читается младший байт LSB, другим- старший байт MSB. Следует помнить, что процедура чтения должна быть обязательно доведено до конца. Если запрограммировано читать два байта, то они должны быть прочитанi к следуя команды загрузки.

2) Второй средство - чтение во время счета по команде CLC (на лету), без остановки. Для реальзации этого средства таймер имеет дополнительный регистр, обращение к которому осуществляется с помощью команды ВЫВОД. При этом для чтения содержания счетчика "на лету" в регистр управляющего слова режима загружается специальный код, под действием которого содержание счетчика без влияния на его работу фиксируется в дополнительном регистре. После этого используется обычная операция ввод (RD) для чтения содержания регистра.



Рисунок 3.5-формат команды СLC

пример программирования

Таймеру соответствуют четыре порта ввода / вывода со следующими адресами например

40h - канал 0;

41h - канал 1;

42h - канал 2;

43h - керуючийрегистр.

MVI A, 00110000B; формуваннякеруючого слова счетчика 0

OUT PORT\_RUS, запись 2-х байт кон., Режим 0, счет двоичный-десятичный

MVI A, DATA\_LOW; загрузки младшего байта кон. В счетчик 0

OUT PORT\_0; при А0, А1 = 00

MVI A, DATA\_HIGH; загрузки старшего байта кон. В счетчик 0

OUT PORT\_0; при А0, А1 = 00

MVI A, 00000000B формирование управляющего слова для чтения

OUT PORT\_RUS; на лету слова состояния и загрузки в таймер

IN PORT\_0, чтение младшего байта счетчика 0

IN PORT\_0, чтение старшего байта счетчика 0

**Контрольные вопросы и задачи**

1.Архитектура программируемого таймера (ПТ) И8253. Назначение, структурная схема, назначение основных блоков, входных / выходных сигналов.

2.Система микрокоманд, форматы команд, формат слова-состояния. Режимы работы БИС i8253 (программируемый таймер). Подключение к шинам МП.

3.Пояснить адресацию регистров таймера.

4. Нарисовать схему таймера и объясните его работу.

5. На лабораторной разработайте фрагменты программ инициализации таймера, в различных режимах.