**Лекція №3**

# 3. Програмований інтервальний таймер I8253

# 3.1 Принцип роботи таймера

ВІС програмованого таймера I8253 призначена для організації роботи мікропроцесорних систем у режимі реального часу і дозволяєформувати сигнали з різними тимчасовими і частотними характеристиками.

Мiкросхема являэ собою однокристальний трьохканальний програмуэмий пристрiй,призначений дляотримання програмно-керованих часових затримок i виконання заданих часових функцiй в мiкропроцесорних системах. ВIС мiстить три канала. Вкожному каналi э регiстр керуючого слова,16-розрядний програ-муэмий лiчильник , працюючий удвійовому або двійково-десятковому кодi (реверсивний), а також двобайтний буферний регiстр, в який по спеціальній командi переписуються константи лiчильника.

Програмування кожного каналу таймера зводиться до слідуючихоперацій:

-запис у регістріндивідуальногокеруючого слова керування каналом;

-запис у 16-розрядний програмуємий лічильник необхідної константи перерахунку;

Запис індивідуального слова керування каналу здійснюється по єдиній для всіх каналів адресі (А0=1, А1=1) регістру керуючого слова.

Вказівка конкретного каналу, до якого відноситься керуюче слово, містится у самому керуючому слові.

Адресація регістрів таймера

|  |  |  |
| --- | --- | --- |
| А0 | А1 | Найменування |
| 0  0  1  1 | 0  1  0  1 | Лічильник каналу 0  Лічильник каналу 1  Лічильник каналу 2  Регістркеруючого слова |

# 

# 3.2 Блок схема таймера

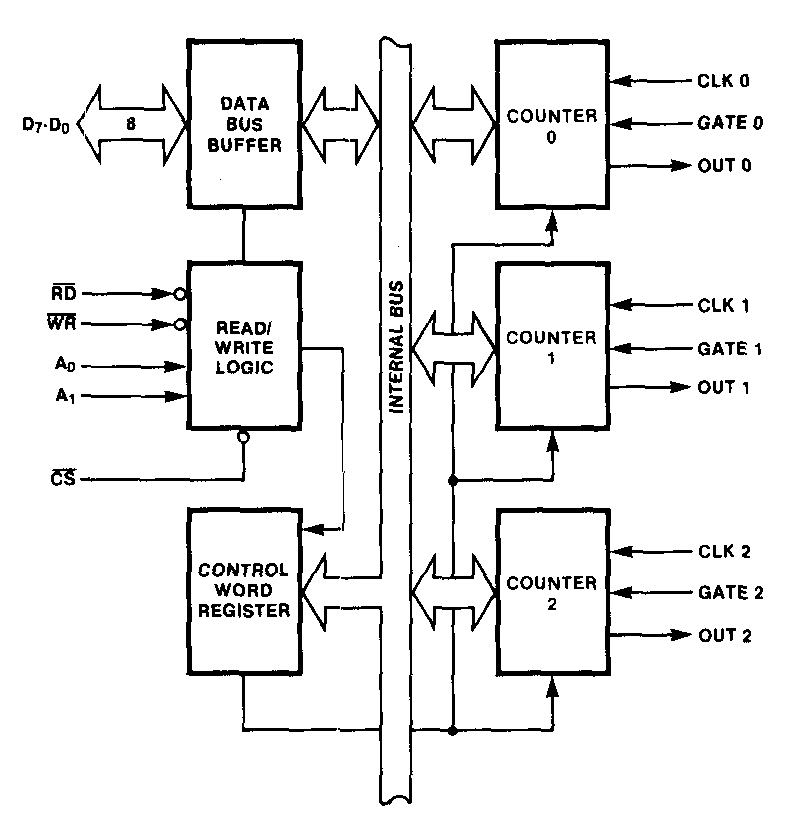
Спрощена структурна схема ПТ приведена на рис.3.1. До складу ВІС входять:

буфер даних*(BD),* призначений для обмінуданими і керуючими словами між МП і ПТ;

схема керування читанням-записом *(RWCU) ,*що забезпечує виконання операцій введення-виводу інформації в ПТ;

регістр керуючого слова *(RGR),* призначений для запису керуючих слів, що задають режими роботи лічильників;

лічильник каналів(СТ0-СТ2).



*DBB -* буфер даних

*RWL -* схема керування читанням-записом

*CWR -* регістр керуючого слова

*СТ0-СТ2 –* лічильник каналів

D7-D0- лінії шини даних

А0-А1- лінії адреси

/RD, /WR- сигнали читання-запису

/CS- сигнал вибору мікросхеми

CLK0, CLK1, CLK2-лічильні входи

OUT0, OUT1, OUT2-виходи лічильників

GATE0, GATE1, GATE2-входи дозволу-заборони ліку та апаратного запуску рахування

Рисунок 3.1-Спрощена структурна схема таймеру

Підключення ПТ до шин мікропроцесора показане на рисунку 3.2.



ША

**ШД**

**ШК**

Рисунок 3.2-Підключення до шин системи

Умовне графічне зображення І8253 представлене на рис. 3.3.



Рисунок 3.3-Графічне зображення таймера

**3.3 Принцип роботи**

ВIС І8253/54 мiстить 3 16-розрядних двійково-десяткових регістрів з попереднiм заповненням .Функціональна конфiгурація його входiв i виходiв встановлюється за допомогою засобiв програмного забеспечення.

Лiчильники повністю незалежнi. Кожен може мати свiй режим роботи i тип лiчби(двійковий або двійково-десятковий). Завантаження лічильників початковим значенням лічби визначається програмно.

Зчитування змiсту кожного лічильника здiйснюється за допомогою команди керування.Таймер також містить додаткове обладнання i команду, якi дозволяють читати змiст лiчильника "на льоту", не заважаючи його роботi.

Програмування таймера вiдбувається за допомогою команди керування. Кожен лічильник індивідуально налагоджується відповідно змiсту регістра керуючого слова.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **SC1**  00-канал 0  01-канал 1  10-канал 2  11-код командиRBC  00-код командиCLC  01-читання\запис мл.байту к.  10-читання\запис ст.байту к.  11-читання\запис мл. а потім ст. байту к.  000-режим 0  001-режим 1  010-режим 2  011-режим 3  100-режим 4  101-режим 5  0-двійковий  1-дв.\десятко-вий рахунок | **SC0** | **RW1** | **RW0** | **M2** | **M1** | **M0** | **BCD** |

CLK - тактовi входи лiчильника;

GATE - входи блокуванняроботи;

OUT - виходилiчильника;

RD/ - читаннялiчильника;

WR/ - запискомандиабоданих;

Рисунок 3.4 –Формат команди керування

**3.4Режимироботи таймера**

Таймер має шість режимів роботи:

* + режим «нуль» - формування сигналу по закінченню інтервалу часу;
  + режим «1» формування імпульсу заданої тривалості;
  + режим «2» генератор імпульсів потрібної частоти;
  + режим «3» програмований генератор прямокутних імпульсів;
  + режим «4» формування імпульсу по закінченню заданого часу;
  + режим «5» лічильник подій.

Режими передбачають зміну виходу таймера OUT з низького рівня OUT=0 до високого OUT=1 після закінчення лічби ( досягнення значення 0 константи перерахунку).

**Режим 0. Формування сигналу по закінченню заданого інтервалу часу**

Після запису за допомогою сигналу /WR керуючого байта СWR, який переводить відповідний канал таймеру у режим 0 вихід таймера OUT встановлюється значення логічної «1»(рис .1.6 а).

Після запису константи перерахунку ( наприклад ,N=4), канал N починає роботу (рис 1.6,б) під час якої константа перерахунку зменшується по тактам вхідних імпульсів CLK, вихід OUT встановлюється в 0.

Коли константа перерахунку досягне нульового значення , вихід таймера встановлюється у високий рівень OUT=1 (рис.1.6,б).

Високий рівень входу GATE=1 дозволяє роботу лічильника, а GATE=0 – призупиняє його роботу. Таким чином, час видачі сигналу по закінченню лічби залежить від введеної константи перерахунку і може збільшуватись на тривалість ∆t за допомогою сигнала керуванняGATE (рис.1.6,в).

З рисунку видно, що в режимі 0 сигнал

GATE 1 –дозволяєлічбу.

Перехід GATE з 1 в 0 -призупиняє лічбу

Перехід GATE 0 в 1 -дозволяє продовження рахунку.

**Режим 1. Формування iмпульсу заданої тривалості (строб)**

Після запису за допомогою сигналу /WR керуючого байта СWR , який переводить відповідний канал таймера у режим 1 . Вихід таймера OUT встановлюється в 1 (рис.1.7а).

Після завантаження константи перерахунку (N=4) таймер починає роботу і формується інтервал часу ∆t строб в залежності від константи перерахунку, а вихід таймера встановлюється у низький рівень OUT=0 (рис.1.7,б).

Високий рівень входу GATE=1, а також перехід його в низький рівень і його утримання, дозволяє роботу лічильника, а перехід сигналу GATE з низького рівня у високий, призводить до перезавантаження константи перерахунку лічильника на початкове значення , що дозволяє змінювати інтервал стробу (рис.1.7,в).

З рисунку видно, що в режимі 1 сигнал

GATE 1 –дозволяє лічбу

Перехід GATE з 1 в 0 -дозволяє лічбу

Перехід GATE з 0 в 1 -перезавантаження константи перерахунку.

**Режим 2. Програмований генератор імпульсів потрібної частоти**

Після запису за допомогою сигналу /WR керуючого байта СWR , який переводить відповідний канал таймера у режим 2 . Вихід таймера OUT встановлюється в 1 (рис.1.8а).

Після завантаження константи перерахунку (N=4) таймер починає роботу.

Коли константа перерахунку досягне значення 1, вихід таймера встановлюється у низький рівень OUT=0 і тримається у цьому стані доки константа перерахунку досягне нульового значення, після чого вихід таймера приймає значення 1 (OUT=1) і відбувається апаратне перезавантаження константи перерахунку на початкове значення N=4 (рис.1.8,б).

Високий рівень GATE =1 , а також перехід сигналу GATE з високого рівня в низький і його утримання , дозволяє роботу таймера, а перехід сигналу GATE з низького рівня у високий, призводить до перезавантаження константи перерахунку лічильника на початкове значення, що дозволяє змінювати частоту імпульсів(рис.1.8,в).

З рисунку видно, що в режимі 2 сигнал

GATE 1 –дозволяє лічбу

GATE 0 –зупиняє лічбу

Перехід GATE з 0 в 1 -перезавантаження константи перерахунку.

**Режим 3. Програмований генератор прямокутних імпульсів**

Після запису керуючого байта СWR , відповідний канал таймера налаштовується на режим 3 . Вихід таймера OUT встановлюється в 1 (рис.1.9а).

Після завантаження константи перерахунку таймер починає роботу і рівень логічної одиниці на виході таймера OUT=1 буде триматися до половини значення константи перерахунку N/2 (якщо N парне), після цього встановлюється логічний 0 , який утримується до досягнення нульового значення константи перерахункуN(рис.1.9,б), після чого відбувається автоматичне перезавантаження константи перерахунку на початкове значення і таймер починає свою роботу спочатку. Таким чином на виході таймера формується сигнал прямокутної форми, в якому тривалість логічного нуля дорівнює тривалості логічної одиниці (меандр,якщо Nпарне). Якщо число N непарне то на виході таймера логічна одиниця буде протягом (n+1)/2 і логічний нуль протягом (n-1)/2.

Високий рівень GATE=1 дозволяє роботу таймера ,а низький рівень GATE=0 призупиняє, а перехід сигналу GATE з 0 в 1 перезавантажує константу перерахунку на початкове значення і дозволяє роботу таймера (рис.1.9,в).

З рисунку видно, що в режимі 3 сигнал

GATE 1 –дозволяє лічбу

GATE 0 –призупиняє лічбу

ПерехідGATE з 0 в 1 -перезавантаження константи перерахунку.

**Режим 4. Формування iмпульсу по закiнченню заданого часу**

Після запису за допомогою сигналу /WR керуючого байта СWR , який переводить відповідний канал таймера у режим 4. Вихід таймера OUT встановлюється в 1 (рис.1.10а).

Після запису константи перерахунку (N=4) таймер починає роботу.

При досягненні константи N=0 стан виходу таймера OUT встановлюється в 0 на один період CLK , після чого OUT знову встановлюється в 1 ( тобто через період часу (n+1) Tсlk) OUT=0 на період одного такту (рис.1.10,б).

Високий рівень входу GATE=1 дозволяє роботу лічильника, GATE=0 – призупиняє його роботу,а перехід сигналу GATE з низького рівня у високий, призводить до продовження рахунку(рис.1.10,в).

З рисунку видно, що в режимі 4 сигнал

GATE 1 –дозволяє лічбу

Перехід GATE з 1 в 0 -призупиняє лічбу

Перехід GATE 0 в 1 -дозволяє продовження лічби.

**Режим 5. Лiчильник подiй**

Після запису за допомогою сигналу /WR керуючого байта СWR , який переводить відповідний канал таймера у режим 5. Вихід таймера OUT встановлюється в 1 (рис.1.11а).

Після запису константи перерахунку (N=4) таймер починає роботу.

При досягненні константи N=0 стан виходу таймера OUT встановлюється в 0 на один період CLK, після чого OUT знову встановлюється в 1 ( тобто через період часу (n+1) Tсlk) OUT=0 на період одного такту (рис.1.11,б).

Високий рівень входу GATE=1, а також перехід в 0 і його утримання, дозволяє роботу лічильника, а перехід сигналу GATE з низького рівня у високий, призводить до перезавантаження константи перерахунку лічильника на початкове значення , що дозволяє змінювати час видачі імпульсу (рис.1.11,в).

З рисунку видно, що в режимі 5 сигнал

GATE 1 –дозволяє лічбу

Перехід GATE з 1 в 0 -дозволяє лічбу

Перехід GATE з 0 в 1 -перезавантаження константи перерахунку.

Керуючі слова можуть бути записані в ПТ у довільному порядку. У будь-який наступний момент часу записуютьсяпочатковікодилічильниківу відповідностізізначеннямирозрядів*D5, D4* керуючихслів.

Виходячи з особливості впливу сигналу GATE на роботу лічильника каналу інтервального таймера, режими його роботи можна поділити на дві групи:

* + з призупинкою (паузою) по сигналу GATE (режим 0,3,4);
  + з перезавантаженням константи по сигналу GATE (режим 1,2,5).



# 3.6 Програмування таймера

Програмування таймера досить гнучке, не має значення послiдовнiсть завантаження керуючих слов режиму окремих лiчильникiв, тобто не обов'яз-ково повинно бути першим керуюче слово лiчильника 0, останнiм-2. Регiстр керуючого слова режиму кожного лiчильника має свою адресу i може завантажуватись незалежно.Однак завантаження змiсту лiчильника повинно вiдповiдати послiдовностi,запрограмованiй в керуючомусловi режиму (-молодший а потім старший байт ).

При програмуванні таймера програміст вирішує наступні задачі:

1. Ініціалізація каналів інтервального таймера.

Здійснюється за допомогою :

* + Запису керуючого слова CWR,
  + Запису константи перерахунку.

2.-Читання регістру стану інтервального таймера, або константи перерахунку RBC ( ReadBackCommand).

Здійснюється за допомогою:

- запису керуючого байту RBC,

- читання регістру стану ініціалізації таймеру RST,

- читання константи перерахунку

а) з зупинкою таймера

б) без зупинки таймера.

Керуюче слово задає один з шести режимів роботи, тип ліку (двійковий чи двійково-десятковий), порядок завантаження і розмірність (один чи два байти) константи. Воно завантажується в регістри RSW каналів таймера, зберігається до слідуючого перепрограмування. Після цього завантажуються константи, як іприймають значення:

* + двійковий – 0-65536
  + двійково-десятковий – 0-9999

Константи перерахунку завантажуються у таймер при А0, А1 ( 00, 01, 10 ).

Існує 2 варіанти завантаження: спочатку всі слова керування, а потім константи, чи слово керування і константу для кожного каналу послідовно.

**3.7Читання змісту лічильника**

Якщо потрiбно знати значення лiчби у процесi роботи (особливо це використовується при визначеннi кiлькостi подiй), то можливо використати два засоби читання, якi не пошкоджують iнформацiю:

1. Використовуючи команду вводу з обраного лiчильника (READ), керуючи входами А0-А1 програмiст може вибирати лiчильник значення якого потрiбно знати.

При читаннi значення лiчильника треба заборонити роботу лiчильника за допомогою команди GATE або шляхом заборони тактових iмпульсiв. При читаннi змісту лiчильника першим читається молодший байт LSB, другим- старший байт MSB. Слiд пам'ятати, що процедура читання повинна бути обов'язково доведена до кiнця. Якщо запрограмовано читати два байти, то вони повиннi бути прочитанi до слiдуючої команди завантаження.

2) Другий засiб – читання пiд час лiчби по команіCLC (на льоту), без зупинки. Для реалiзацii цього засобу таймер має додатковий регiстр,звернення до якого здiйснюється за допомогою команди вивiд. При цьому для читання змiсту лiчильника "на льоту" в регiстр керуючого слова режиму завантажується спецiальний код, пiд дiєю якого змiст лiчильника без впливу на його роботу фiксується у додатковому регiстрi. Пiсля цього використовується звичайна операцiя ввод(RD) для читання змiсту регiстра.



Рисунок 3.5–Формат команди СLC

Приклад програмування

Таймеру відповідають чотири порти введення/виводу з наступними адресами наприклад:

40h - канал 0;

41h - канал 1;

42h - канал 2;

43h - керуючийрегістр.

MVI A, 00110000B ;формуваннякеруючого слова лічильника 0

OUT PORT\_RUS ;запис 2-х байт конст., режим 0, рахунок двійковий-десятковий

MVI A, DATA\_LOW;завантаження молодшого байта кон. У лічильник 0

OUT PORT\_0 ; при А0,А1=00

MVI A, DATA\_HIGH ;завантаження старшого байта кон. У лічильник 0

OUT PORT\_0 ; при А0, А1=00

MVI A, 00000000B ;формування керуючого слова для читання

OUT PORT\_RUS ; на льоту слова стану і завантаження в таймер

IN PORT\_0 ;читання молодшого байта лічильника 0

IN PORT\_0 ;читання старшого байта лічильника 0

Контрольні питання та завдання

1.Архітектура Програмованого Таймера (ПТ) І8253. Призначення, структурна схема, призначення основних блоків, вхідних/вихідних сигналів.

2.Система мікрокоманд, формати команд, формат слова-стану. Режими роботи ВІС І8253 (програмований таймер). Підключення до шин МП.

3.Поясніть адресацію регістрів таймера.

4.Намалюйте схему таймера та поясніть його роботу.

5. На лабораторній розробіть фрагменти програм ініціалізації таймера , в різних режимах.