

Техническое описание
и
инструкция по эксплуатации
платы ЛА-ТМР

Москва 2001

ТЕХНИЧЕСКОЕ ОПИСАНИЕ И ИНСТРУКЦИЯ ПО ЭКСПЛУАТАЦИИ
ПЛАТЫ
СЧЕТЧИКОВ-ТАЙМЕРОВ И ВВОДА/ВЫВОДА ЦИФРОВОЙ
ИНФОРМАЦИИ
В ПЭВМ ТИПА ИВМ РС/ХТ/АТ.

- * шесть 16 разрядных программируемых счетчиков-таймеров;
- * высокостабильный кварцевый генератор частоты 10 МГц;
- * 16 ТТЛ совместимых цифровых линии: по 2 байта 8 - выход
8 - вход
параллельная передача данных
- * два независимых прерывания для IRQ ИВМ РС.

Содержание.

	Стр.
1. АОЗТ "Руднев-Шилаев".	
1.1 Центр АЦП. -----	3
1.2 Продукция центра. -----	4
2. Плата таймеров ЛА-ТМР.	
Назначение и состав. -----	6
2.1 Функциональная схема. -----	7
Схема расположения переключателей. -----	8
2.1.1 Цифровой порт. -----	9
2.1.2 Схема шестиканальных таймеров. -----	9
Режимы счета -----	10
Работа со счетчиками -----	12
2.2 Выбор базового адреса. -----	13
2.3 Выбор режимов работы платы. -----	14
2.4 Описание входных разъемов. -----	15
3. Программирование платы ЛА-ТМР.	
3.1 Описание регистров ввода-вывода.	
Выбор конфигурации. -----	16
3.2 Пример использования. -----	20
4. Комплект поставки. -----	22
5. Подготовка к работе и уход за платой ЛА-ТМР. Требования по технике безопасности. -----	22
6. Описание программы TMR.EXE, входящей в комплект поставки. -----	22
Приложение. Прерывания. -----	25
7. Гарантийные обязательства. -----	32

2. Плата таймеров ЛА-ТМР. Назначение и состав.

Плата ЛА-ТМР - это таймеры и цифровые порты для IBM PC/XT/AT и совместимых с IBM компьютеров. Каналы счетчиков-таймеров платы имеют несколько режимов работы, включающих подсчет событий, генерацию импульсов, выработку сигналов прямоугольной формы и измерение частоты. Счетчики-таймеры могут быть проконтролированы программно по уровню или по фронту сигнала. ЛА-ТМР имеет два независимых входа прерывания по фронту сигнала, либо от внешнего воздействия, либо по команде каналов счетчиков-таймеров. Цифровой порт может быть использован для управления устройствами выработки импульсов, получения информации от удаленных устройств при лабораторных исследованиях, калибровке радиоэлектронной продукции, промышленном процессе мониторинга и контроля за объектами.

2.1 Функциональная схема. Схема расположения переключателей.

Плата ЛА-ТМР состоит из следующих функциональных узлов.

Схема счетчика-таймера. Включает два независимых трехканальных таймера, схему синхронизации, высокостабильный кварцевый генератор (10 МГц) и делитель частоты на 10. Входы CLK и GATE обоих таймеров через сопротивление присоединены к шине +5В, так что в отсутствие этих сигналов CLK и GATE таймеров будут иметь уровень логической "1". Входы обоих счетчиков таймеров CLK0 присоединены к схеме синхронизации. На них может быть подана либо частота $F=10$ МГц, либо $F/10=1$ МГц со схемы синхронизации. Выход третьего канала OUT2 второго таймера (ТМР-1) через инвертор выведен на внешний разъем (OUT2). На внешний разъем выведена тактовая частота, которая может быть выбрана схемой синхронизации: F или F/10 - сигнал F ext. на внешнем разъеме.

Цифровой порт. Состоит из двух независимых 8 битных портов вывода - РА и ввода - РВ. Порт ввода со стробированием (импульс STROBE на внешнем разъеме). Для непрерывного ввода сигнал STROBE необходимо заземлить.

Схема обработки прерываний. Содержит два независимых устройства обработки прерываний для таймера - вход INT1 (разъем таймера) и для цифрового порта - вход INTO (разъем цифрового порта).

Расположение переключателей.

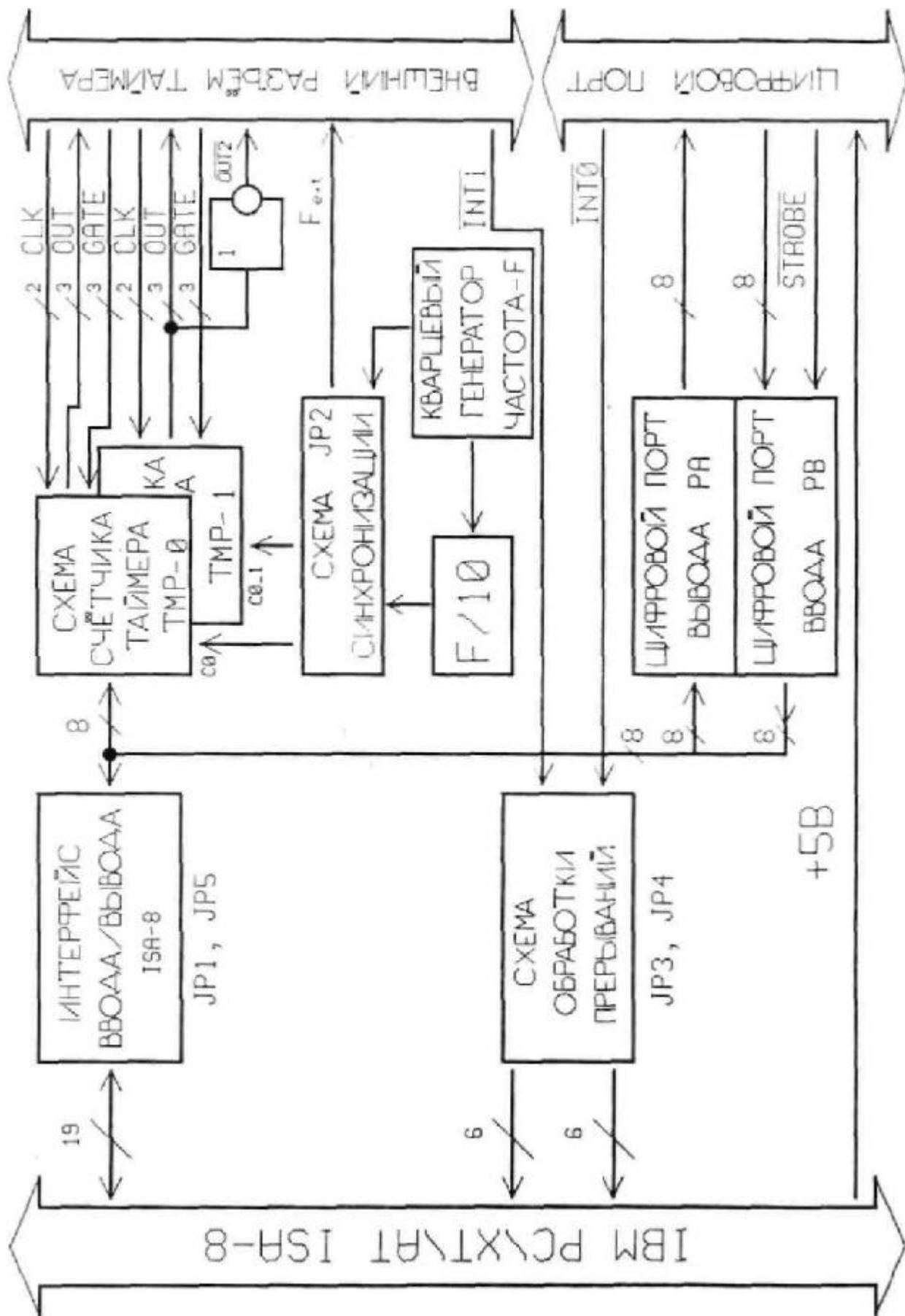
На рис 2.1.1 стр. 8 приведена схема расположения переключателей (упрощенная монтажная схема).

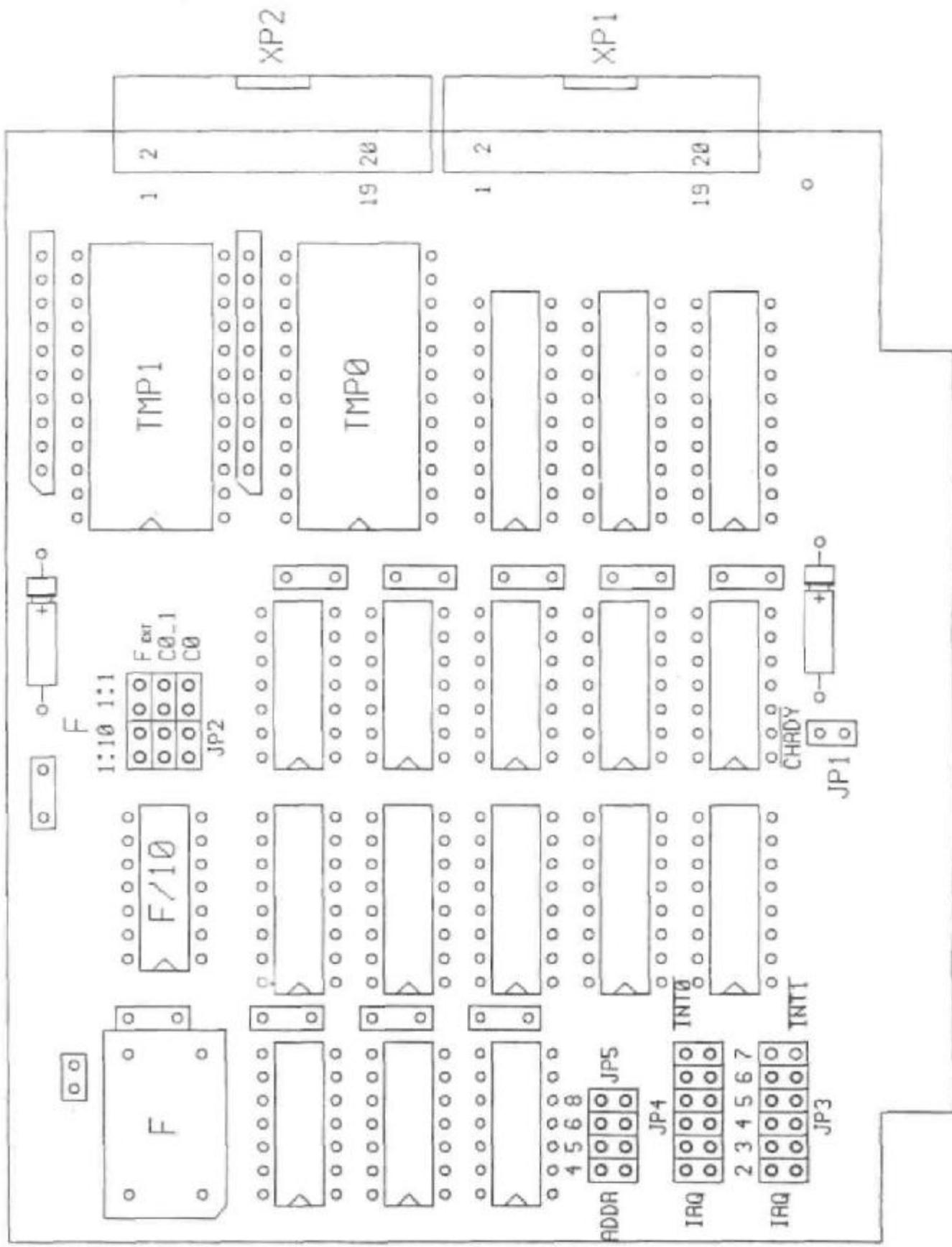
JP1 - переключатель скорости обмена данных (для ПЭВМ с тактовой частотой более 8 МГц должен быть замкнут);

JP2 - переключатель схемы синхронизации - выбор частоты синхронизации $F=10$ МГц или $F/10=1$ МГц для входов CLK0 обоих таймеров и для внешнего использования F_{ext} .

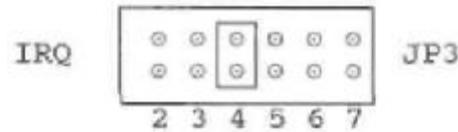


частота синхронизации $F=1$ МГц, частота синхронизации $F=10$ МГц. Причем для CO, CO_1 и F_{ext} может быть выбрана в качестве входной, как $F=10$ МГц, так и $F=1$ МГц, независимо друг от друга.

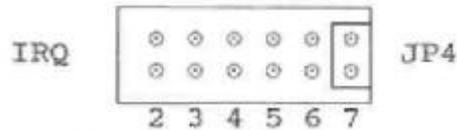




JP3 - переключатель, выбирающий сигнал разрешения прерывания для таймера. Для примера JP3 показан в положении IRQ4.



JP4 - переключатель, выбирающий сигнал разрешения прерывания для цифрового порта. Для примера JP4 показан в положении IRQ7.



JP5 - переключатель, выбирающий базовый адрес платы (используется шестнадцатиричная система для номера). См. п. 2.2 Выбор базового адреса.

2.1.1 Цифровой порт.

Плата ЛА-ТМР содержит два независимых цифровых порта - РА вывод данных (8 цифровых линий), РВ ввод данных (8 цифровых линий), независимо друг от друга выведенных на внешний разъем. Для цифрового порта на внешнем разъеме имеется сигнал STROBE, см. функц. схему ЛА-ТМР. Стробированная запись в порт происходит уровнем логического "0". Данные хранятся до следующего строба. При заземленном входе STROBE, порт РВ работает в обычном режиме ввода - данные на выходе меняются синхронно с данными на входе.

На внешний разъем цифрового порта выведен сигнал управления прерыванием INTO, информацию о приходе которого можно считать из статусного регистра. Запись происходит по перепаду из "0" в "1". Этот сигнал подан на схему обработки прерываний, с которой он через переключатель JP4 может быть подключен к одному из прерываний IRQ2..IRQ7.

2.1.2 Схема шестиканальных таймеров.

Плата ЛА-ТМР содержит два трехканальных программируемых таймера P8254 для формирования сигналов с различными временными параметрами. По регистрам он идентичен микросхеме P8253 (отечественный аналог КР580ВИ53). В состав таймера входят: буфер шины данных, схема управления вводом-выводом и три независимых канала, каждый из которых содержит регистр режима, схему управления каналом, буфер и 16-разрядный счетчик.

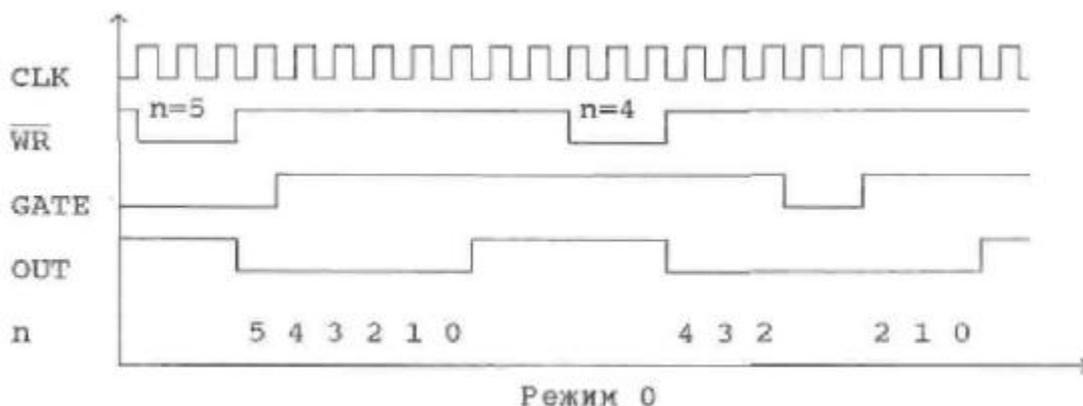
Программируемый таймер содержит три независимых 16 разрядных канала с общей схемой управления. Каждый канал может работать в шести режимах. Программирование режимов работы каналов осуществляется индивидуально и в произвольном порядке путем ввода управляющих слов в контрольные регистры, а в счетчики - запрограммированного числа байтов. Управляющее слово определяет режим работы канала, тип счета (двоичный **или** двоично-десятичный), формат чисел (одно- или двухбайтовый). Обмен информацией с IBM PC осуществляется по 8 разрядному двунаправленному каналу данных. Максимальное значение счета: в двоичном коде 2^{16} , в двоично-десятичном коде 10^4 .

Режимы счета.

Программирование канала осуществляется путем ввода управляющих слов в регистр режима каналов и начального значения в его счетчики. Каждый канал имеет управляющий вход GATE и выход OUT и может работать в одном из следующих шести режимов.

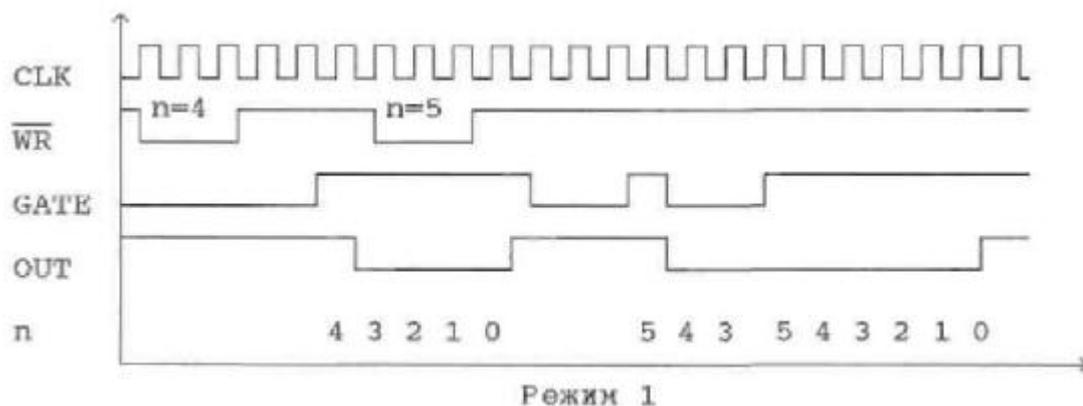
Режим 0 (прерывание терминального счета).

После записи управляющего слова в регистр режима канала на выходе OUT устанавливается напряжение низкого уровня; загрузка счетчика не изменяет это состояние. Затем начинается декремент счетчика (последовательное вычитание из числа, занесенного в него, единицы). В момент, когда счетчик обнулится, на выходе OUT устанавливается напряжение высокого уровня и сохраняется до загрузки счетчика новым значением. Счет возможен только при наличии сигнала высокого уровня на входе GATE. Низкий уровень этого сигнала или ниспадающий фронт запрещают счет. Перезагрузка счетчика во время счета приводит к следующему: загрузка младшего байта останавливает текущий счет, загрузка старшего байта запускает новый цикл счета. Минимально допустимое значение счетчика равно 2.



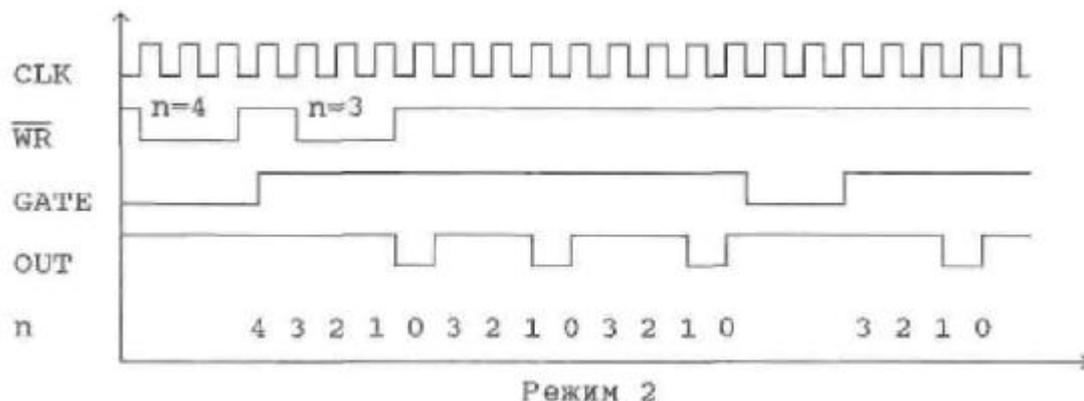
Режим 1 (ждущий мультивибратор).

На выходе OUT формируется отрицательный импульс длительностью $t=n \cdot T$, где n - число, загруженное в счетчик, T - период тактовых импульсов. Низкий уровень на выходе OUT устанавливается со следующего такта после подачи на вход GATE сигнала высокого уровня. Загрузка в счетчик нового числа не влияет на длительность текущего импульса, а учитывается при следующем запуске. Перезапуск счетчика производится нарастающим фронтом входа GATE (без перезагрузки счетчика). Минимальное допустимое $n=1$.



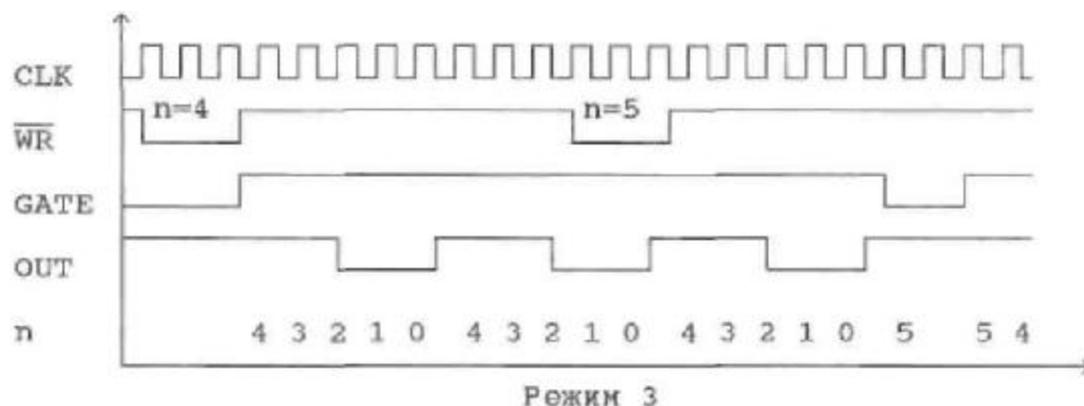
Режим 2 (генератор частоты).

Каждый раз после достижения счетчиком нуля на выходе OUT появляется отрицательный импульс с длительностью один такт. Перегрузка счетчика сказывается только после перезапуска счетчика. При исчезновении сигнала высокого уровня на входе GATE прекращается счет и на выход OUT подается напряжение высокого уровня. Перезапуск счетчика происходит при наличии на входе GATE сигнала высокого уровня.



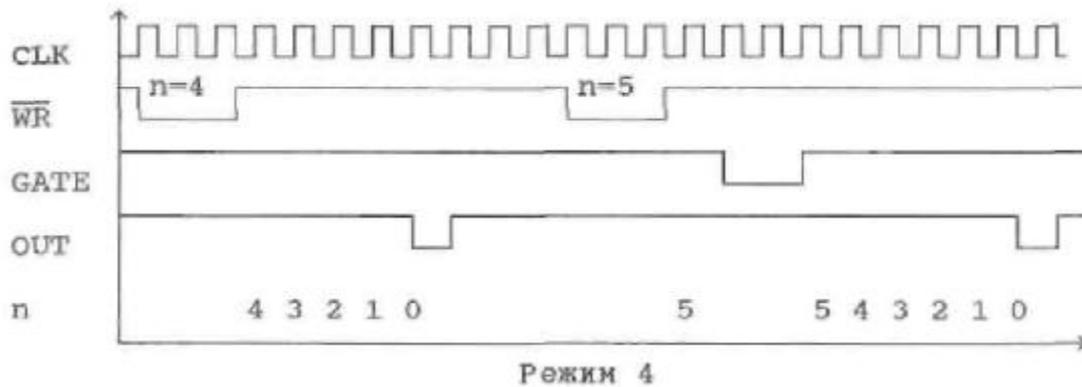
Режим 3 (генератор меандра).

Аналогичен режиму 2, но положительный уровень выходного сигнала занимает первый полупериод, а отрицательный - второй полупериод. Точнее, если n (начальное значение счетчика) четно, то длительность положительного и отрицательного полупериодов равна $n \cdot T/2$; если же n нечетно - то $(n+1) \cdot T/2$ и $(n-1) \cdot T/2$ соответственно. Низкий уровень сигнала на входе GATE запрещает счет, на выходе OUT устанавливается сигнал высокого уровня. Высокий уровень GATE разрешает счет, а нарастание его запускает счетчик начального состояния. Отметим, что $n=3$ в этом режиме недопустимо.



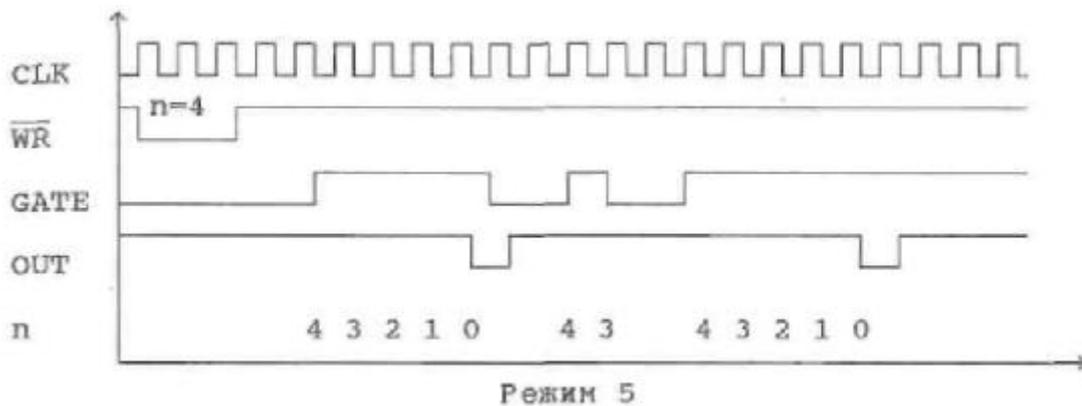
Режим 4 (счетчик событий).

По окончании отсчета числа, загруженного в счетчик, на выходе OUT формируется отрицательный импульс длительностью один такт. Запись в счетчик во время счета младшего байта не влияет на текущий счет, а запись старшего байта перезапускает счетчик. Низкий уровень входа GATE запрещает счет, высокий - разрешает. Минимальное допустимое значение счетчика равно 1.



Режим 5 (счетчик событий с автозагрузкой).

Отличие от режима 4 состоит в том, что каждое нарастание сигнала на входе GATE перезапускает счетчик. Перегрузка счетчика не влияет на текущий цикл, однако следующий цикл определяется вновь занесенным числом.



Работа со счетчиками.

Операции чтения/записи.

Для каждого из трех счетчиков в контрольном байте должны быть соответствующим образом заданы:

- тип операции чтения/записи;
- режим счета;
- используемый код счетчика (BCD).

Контрольный байт должен быть сформирован и записан до того, как будет производиться запись значения счетчика. Поскольку контрольный регистр и все три регистра счетчиков имеют отдельные адреса и каждый контрольный байт сам выбирает счетчик, которому он предназначен (посредством SC1 и SCO), то нет никаких специальных ограничений на последовательность программирования каналов таймера. Может быть использована любая последовательность программирования, предусмотренная соглашением Intel 8254. Существуют три типа операций над счетчиком: прочитать/загрузить LSB (less significant byte = наименее значимый байт = младший байт), прочитать/загрузить MSB (most significant byte = наиболее значимый байт = старший байт), прочитать/загрузить LSB, затем MSB. При программировании необходимо следить за парностью операций чтения/записи и порядком следования байтов.

Чтение счетчика/счетчиков.

У таймера 8254 предусмотрена специальная команда Read-Back ("прочитать назад"), позволяющая проверить для любого выбранного канала таймера:

- значение счетчика;
- запрограммированный режим;
- текущее состояние выхода OUT;
- текущее состояние флага конца счета NC (Null Count)

Эта команда записывается в контрольный регистр таймера, ее формат описан в разделе регистры таймера.

Команда удобна для синхронного контроля за состоянием счетчиков, поскольку позволяет произвести одновременное защелкивание текущих значений счетчика выбранных каналов. Если после послыки в порт BASE+7 двоичного кода 1101 1110 (CNT=0 - защелкнуть текущие значения, C2=C1=C0=1 - выбрать все три канала) прочитать значения всех трех счетчиков, то это будет эквивалентно трем отдельным операциям чтения счетчиков "на лету" (см. ниже), НО произведенным синхронно.

При использовании данной команды для чтения статусных байтов, последние также подвергаются процедуре одновременного "защелкивания", и затем могут быть по очереди прочитаны из регистров счетчиков таймера BASE+4/5/6 для одного и BASE+8/9/A для второго таймера.

Чтение значения отдельного счетчика.

Существует два способа чтения текущего значения счетчика канала.

1. Чтение с остановом счетчика.

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

2 Чтение "на лету".

Для считывания счетчика без остановки процесса счета используется послыка в порт BASE+7 (BASE+B) управляющего слова в режиме "защелкивания". Это управляющее слово фиксирует текущее значение счетчика и Вы можете считать его младший байт, а затем старший байт.

2.2 Выбор базового адреса.

Плата имеет базовые адреса, которые выбираются переключателем JP5 (его расположение показано на рис.2.1.2) стр. 8. Так как плата ЛА-ТМР использует двенадцать адресов, то адресные линии А0, А1, А2 и А3 используются самим портом ввода/вывода, А9=1; А7=0 - жестко сконфигурированы.

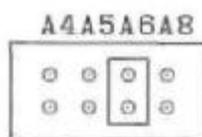


При этом, если переключатель замкнут - соответствующая ей адресная линия имеет уровень логического "0", при разомкнутой переключатель - уровень логической "1". В таблице приведены возможные комбинации базовых адресов:

Базовый адрес	A4	A5	A6	A8
200 (200-20B)	..0	..0	..0	..0
210 (210-21B)	..1	..0	..0	..0
220 (220-22B)	..0	..1	..0	..0
230 (230-23B)	..1	..1	..0	..0
240 (240-24B)	..0	..0	..1	..0
250 (250-25B)	..1	..0	..1	..0
260 (260-26B)	..0	..1	..1	..0
270 (270-27B)	..1	..1	..1	..0
300 (300-30B)	..0	..0	..0	..1
310 (310-31B)	..1	..0	..0	..1
320 (320-32B)	..0	..1	..0	..1
330 (330-33B)	..1	..1	..0	..1
340 (340-34B)	..0	..0	..1	..1
350 (350-35B)	..1	..0	..1	..1
360 (360-36B)	..0	..1	..1	..1
370 (370-37B)	..1	..1	..1	..1

Все базовые адреса приведены в шестнадцатиричной системе счисления или в гексакодах (Hex).

В качестве примера приведем рисунок для установленного базового адреса 330 (Hex):



Базовый адрес платы необходимо устанавливать так, чтобы она не занимала адреса портов уже вставленных плат в компьютер и не возникало конфликтов с другими устройствами. Установка адреса ЛА-ТМР - на производстве - 330 (hex).

2.3 Выбор режимов работы платы.

2.3.1 Выбор прерывания IRQ (переключатель JP3 и JP4).

Программируемый контроллер прерываний (ПКП, Programmable Interrupt Controller, PIC) реализует векторную систему прерываний IBM-совместимого компьютера. Микросхема 8259А фирмы Intel (советский аналог КР580ВН59), а также ее модификации 8259А-2 и 8259А-8, поддерживает 8 уровней прерываний от восьми различных устройств.

Основные функции контроллера:

- фиксация запросов прерывания от восьми внешних источников;
- программное маскирование поступающих запросов;
- присвоение фиксированных или циклически изменяемых приоритетов входам контроллера, на которые поступают запросы;
- инициализация вызова процедуры обработки поступившего аппаратного прерывания.

Количество обслуживаемых внешних источников прерываний может быть увеличено путем каскадирования нескольких контроллеров.

CLK1; CLK2 и CLK1_1; CLK2_1 - входы сигналов синхронизации для каналов TMP0 и TMP1 соответственно, (CLK0 и CLK0_1 таймеров могут быть присоединены к схеме синхронизации (F= 10 МГц или F= 1 МГц, переключатель JP2) , а на внешний разъем не выведены),
 GATE0..2 и GATE0_1..2_1 - управляющие входы для каналов таймеров TMP0 и TMP1 соответственно,
 OUT0..2 и OUT0_1..2_1 - выходы каналов таймеров TMP0 и TMP1 соответственно,
 INT1 - прерывание для таймера,
 F_{ext}- выход схемы синхронизации.

Программирование платы ЛА-ТМР.

3.1 Описание регистров ввода-вывода. Выбор конфигурации.

Управление платой ЛА-ТМР производится посредством чтения/записи внутренних регистров платы. Все регистры ЛА-ТМР имеют 8-разрядный интерфейс (чтение производится инструкцией IN AL,DX; запись - инструкцией OUT DX,AL).

Регистры счетчиков таймера Intel 8254 являются 16-разрядными, остальные регистры ЛА-ТМР имеют 8-разрядный формат. Доступ к регистрам ЛА-ТМР осуществляется через 12 портов ввода/вывода, занимающих в адресном пространстве ПЭВМ 12 последовательных адресов (начиная с базового адреса BASE).

В первой колонке указано смещение выбираемого регистра относительно базового адреса (В).

	Чтение	Запись
В+0	цифровой порт ввода РВ	цифровой порт вывода РА
В+1	управляющий регистр прерывания	управляющий регистр прерывания
В+2	статусный регистр прерывания	сброс прерывания 1 канала
В+3	не используется	сброс прерывания 2 канала
В+4	чтение 0 канала счетчика/таймера TMP0	0 канал счетчика/таймера TMP0
В+5	чтение 1 канала счетчика/таймера TMP0	1 канал счетчика/таймера TMP0
В+6	чтение 2 канала счетчика/таймера TMP0	2 канал счетчика/таймера TMP0
В+7	контрольный регистр счетчика/таймера TMP0	контрольный регистр счетчика/таймера TMP0
В+8	чтение 0 канала счетчика/таймера TMP1	0 канал счетчика/таймера TMP1
В+9	чтение 1 канала счетчика/таймера TMP1	1 канал счетчика/таймера TMP1
В+А	чтение 2 канала счетчика/таймера TMP1	2 канал счетчика/таймера TMP1
В+В	контрольный регистр счетчика/таймера TMP1	контрольный регистр счетчика/таймера TMP1

Примечание: 4 адреса со смещениями (относительно BASE) С, D, Е, F являются

резервными.

Цифровой порт ввода и вывода.

Как уже упоминалось ранее цифровой порт состоит из двух физически независимых устройств. Порт PA - вывод данных (восемь цифровых линий, выведенных на внешний разъем) и порт PB - ввод данных (восемь цифровых линий, выведенных на внешний разъем, независимо от PA) . В целом порт PA и PB занимают 16 линий на внешнем разъеме XP1.

Цифровой порт вывода данных PA может быть только записан и использует адрес BASE+0.

Формат расположения битов данных регистра вывода:

BASE+0	D7	D6	D5	D4	D3	D2	D1	DO
только запись	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0

Цифровой порт ввода данных PB может быть только прочитан и использует адрес BASE+0.

Формат расположения битов данных регистра ввода:

BASE+1	D7	D6	D5	D4	D3	D2	D1	DO
только чтение	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0

PA7..PA0 - двоичный цифровой код. PA7 - старший бит, PA0 - младший бит (PB7..PB0 - соответственно).

Контрольный регистр.

ЛА-TMP содержит контрольный регистр, управляющий разрешением запроса прерывания на шину ПЭВМ IBM PC и имеет адрес BASE+1. Контрольный регистр может быть как записан, так и прочитан.

Формат данных:

BASE+1	D7	D6	D5	D4	D3	D2	D1	D0
запись	X	X	X	X	X	X	EN1	EN0

EN0 = 0 - запрещает выставление прерывания от контакта внешнего разъема цифрового порта (сигнал INTO);

EN0 = 1 - разрешает.

EN1 = 0 - запрещает выставление прерывания от контакта таймерного разъема (сигнал INT1);

EIRQ1 = 1 - разрешает.

Статусный регистр.

ЛА-TMP имеет статусный регистр для оперативного определения режима работы платы и для ее тестирования. Статусный регистр может быть только прочитан. Адрес BASE+2.

Формат:

BASE+2	D7	D6	D5	D4	D3	D2	D1	D0
только чтение	X	X	X	X	X	X	IRQ1	IRQ0

IRQ1=1 показывает, что есть запрос на прерывание от внешнего события, пришедшего от контакта разъема таймера,

IRQ0=1 показывает, что есть запрос на прерывание от внешнего события, пришедшего от контакта разъема цифрового порта.

Регистры программируемого таймера/счетчика.

LA-TMP имеет два таймера/счетчика, каждый из которых состоит из 3 независимых шестнадцатиразрядных каналов счетчика и занимают, соответственно, по четыре адреса BASE+4, BASE+5, BASE+6, BASE+7 ДЛЯ TMP0 И BASE+8, BASE+9, BASE+A, BASE+B ДЛЯ TMP1, используемых для программирования таймера.

BASE+4 и +8	счетчик 0	Чтение/запись
BASE+5 и +9	счетчик 1	Чтение/запись
BASE+6 и +A	счетчик 2	Чтение/запись
BASE+7 и +B	контрольный регистр	Запись

Внутренняя структура таймера/счетчика является 16 разрядной, а внешний интерфейс 8 разрядный.

Ниже приведено краткое описание регистров таймера Intel 8254 и их формата.

Формат данных контрольного регистра:

BASE+7	D7	D6	D5	D4	D3	D2	D1	DO
только запись	SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC1 и SC0 - выбирает счетчик:

SC1	SC0	
0	0	Счетчик 0
0	1	Счетчик 1
1	0	Счетчик 2
1	1	Команда чтения статуса

RW1 и RW0 - выбор операции чтения/записи:

RW1	RW0	Операция
0	0	Защелкивание счетчика
0	1	Чтение/запись младшего байта
1	0	Чтение/запись старшего байта
1	1	Чтение/запись сначала младшего, затем старшего байта

M2, M1, M0 - выбор режима работы счетчика:

M2	M1	M0	Режим
0	0	0	0 прерывания терминального счета
0	0	1	1 ждущий мультивибратор
X	1	0	2 генератор частоты (импульсный)
X	1	1	3 генератора меандра
1	0	0	4 счетчик событий
1	0	1	5 счетчик событий с внешней загрузкой

BCD - выбор способа кодирования счетчика:

BCD	Тип кода счетчика
0	двоичный
1	двоично-кодированный десятичный

Если установлен двоичный (0), то может быть счет любого числа в диапазоне от 0 до 65535, если двоично-десятичный - от 0 до 9999.

Если SC1 и SCO установлены в 1, происходит операция считывания статусного слова. Формат данных контрольного регистра в этом случае становится следующим:

BASE+7/B	D7	D6	D5	D4	D3	D2	D1	DO
только запись	1	1	CNT	STA	C2	C1	CO	X

CNT=0 - считывание текущего значения счета выбранного счетчика, STA=0 - считывание текущего режима

счетчика, C2, C1, CO - выбор счетчика для операции считывания статусного слова:

C2 = 1	выбрать	счетчик	2
C1=1	выбрать	счетчик	1
C0=1	выбрать	счетчик	0

Если SC1 и SCO установлены в 1, а STA - 0, происходит считывание статусного слова, выбранного счетчика C2, C1, CO. Формат данных статусного слова:

BASE+4/5/6 +8/9/A	D7	D6	D5	D4	D3	D2	D1	D0
чтение	OUT	NC	RW1	RW0	M2	M1	M0	BCD

OUT - текущее состояние выхода выбранного канала.

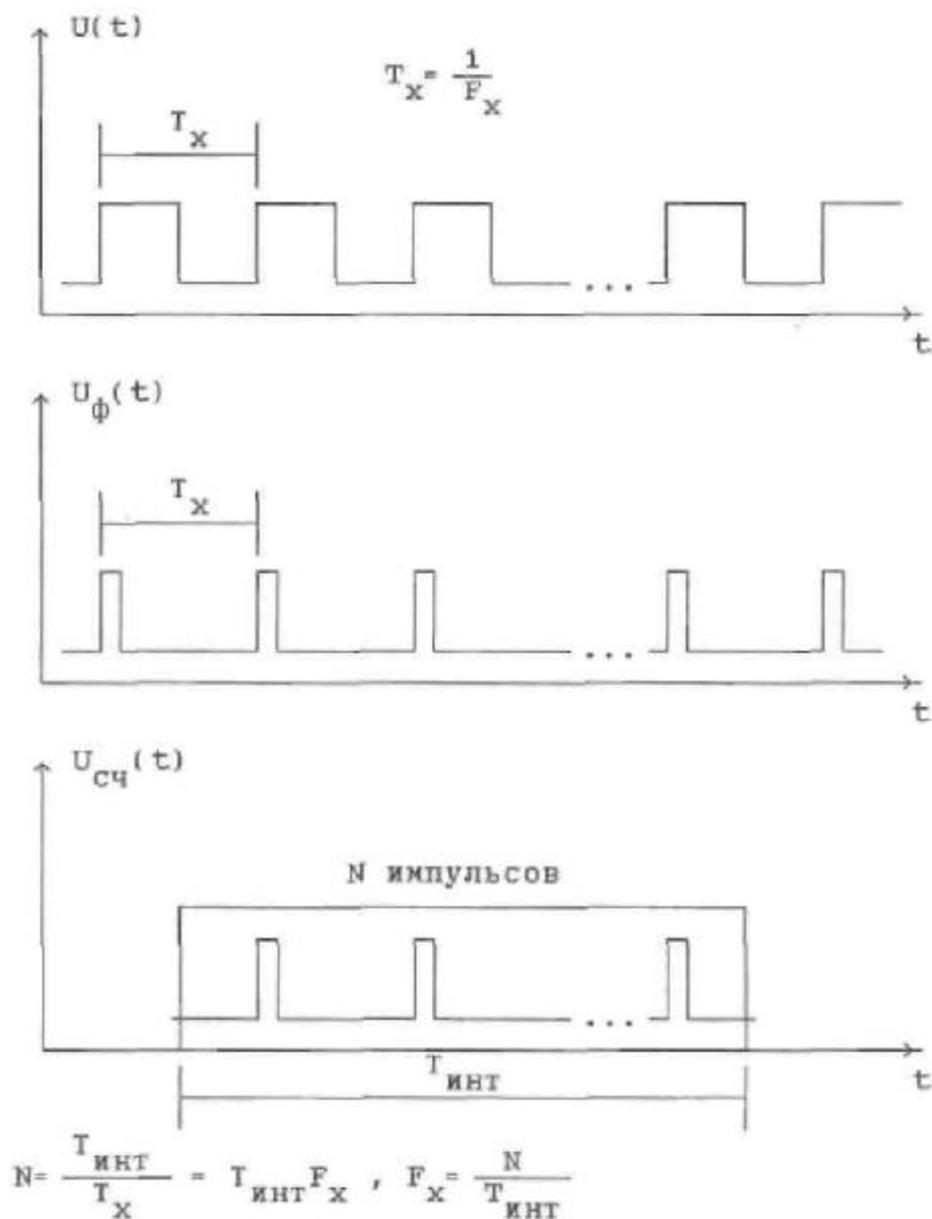
N0=0, если последнее записанное 16-разрядное слово для счета уже перезагружено в считывающий элемент после выполнения прошлого счета.

Для более детального изучения программирования таймера/счетчика можно обратиться к Справочнику "Микропроцессоры и микропроцессорные комплекты интегральных микросхем", том I, Москва, "Радио и связь" 1988 г.

3.2 Пример использования платы ЛА-ТМР.

Цифровой метод измерения частоты. Метод заключается в подсчете числа периодов исследуемого колебания за определенный интервал времени. Различают два метода: частотомер и периодомер.

3.2.1 Метод частотомера заключается в том, что за известный интервал времени $T_{\text{инт}}$ подсчитывается число периодов N колебания $U(t)$ неизвестной измеряемой частоты F_x .

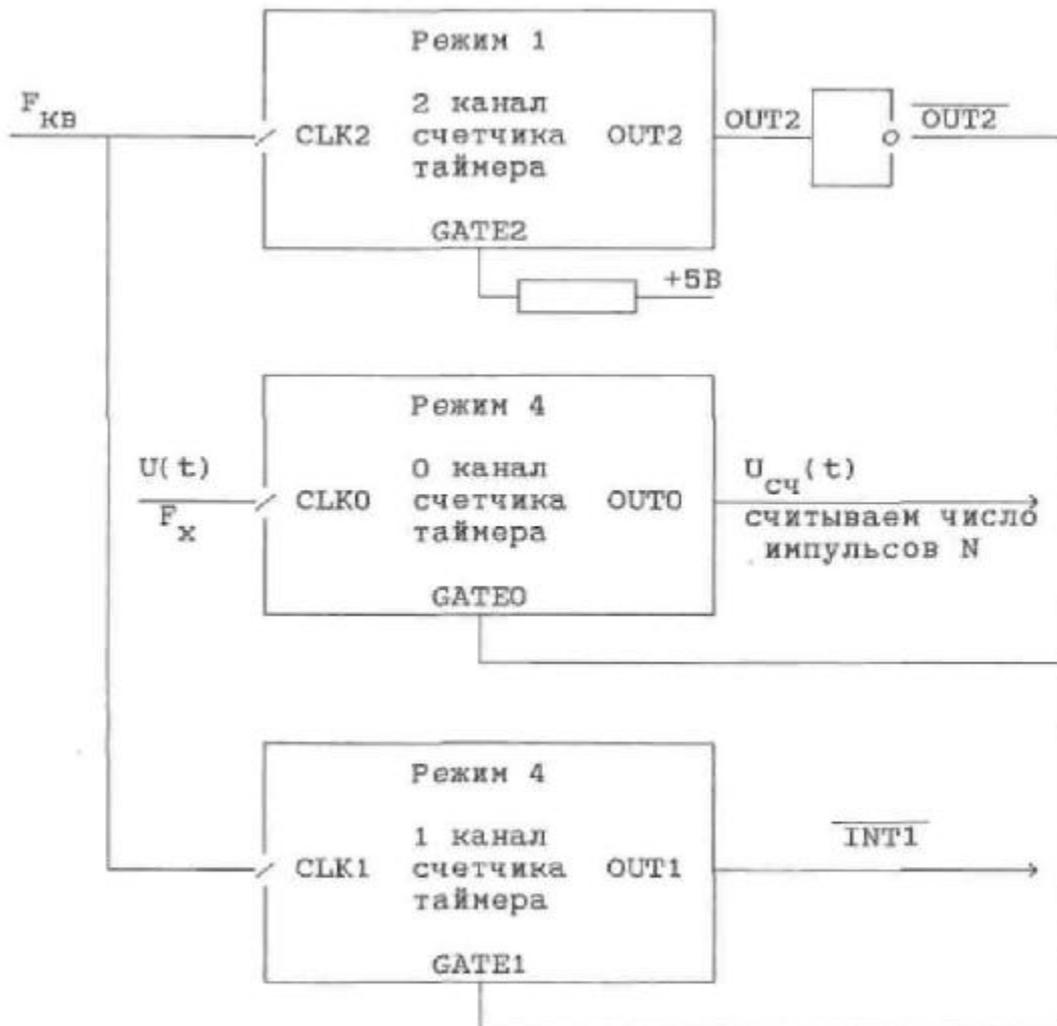


Структура подобного частотомера аналогична структуре АЦП с промежуточным преобразованием в интервал.

Один из возможных вариантов соединения счетчиков-таймеров для этого режима приведен на функциональной схеме ниже:

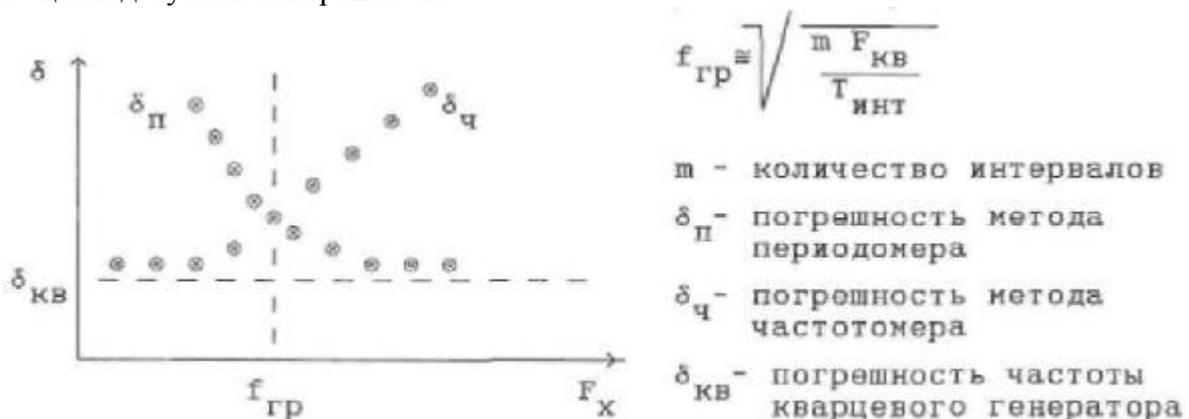
$F_{\text{кв}}$ - кварцевая частота схемы синхронизации

$F=10$ МГц или $F/10=1$ МГц



3.2.2 Метод периодомера заключается в том, что счетные импульсы подсчитываются за время, равное одному или нескольким периодам измеряемой частоты.

Необходимость применения каждого метода необходимо решать в конкретных условиях из оценки допустимой погрешности.



4. Комплект поставки.

В комплект поставки платы ЛА-ТМР входит:

- плата ЛА-ТМР с шиной ISA-8 (для IBM PC\XT\AT), ... 1 шт.
- ответная часть внешних разъемов Б20, 2 шт.
- техническое описание и инструкция по эксплуатации
с гарантийными обязательствами, 1 шт.
- дискетта с программным обеспечением 1 шт.
- упаковочная тара для транспортировки 1 шт.

АО "Руднев-Шиляев" оставляет за собой право усовершенствования платы и поставляемого программного обеспечения без оповещения и согласования с пользователем.

5. Подготовка к работе и уход за платой ЛА-ТМР. Требования по технике безопасности.

Плата ЛА-ТМР содержит лишь цепи безопасного сверхнизкого напряжения и, согласно ГОСТ 25861-83 (СТ СЭВ 3743-82) п.2.1.2 примечание, не требует специальной защиты персонала от случайного соприкосновения со своими вторичными цепями.

Она рассчитана для установки в один из слотов IBM PC, от которого получает питание.

Потребление (по шине питания IBM PC):
+ 5В ... не более 350 мА

Перед установкой платы необходимо выключить Ваш компьютер и все периферийные устройства (такие как принтер и монитор, например), определить местоположение каждой платы в вашем IBM PC и освободить место для платы ЛА-ТМР. Перед каждой ее установкой необходимо протереть разъем, вставляемый в слот IBM PC, слегка увлажненной спиртом хлопчатобумажной тканью. Расход

спирта на каждую операцию - 0,05 см . После установки в компьютер, ЛА-ТМР закрепляется винтом за верхнюю часть крепежно-установочного кронштейна в PC.

К внешним разъемам присоединить кабели (их цоколевка показана на рисунке в п.2.4), соединяющие плату с периферийными устройствами и кабели всех периферийных устройств.

Все используемые в конфигурации аналоговые и цифровые каналы должны иметь общее с IBM PC заземление!

На этом аппаратная часть установки платы ЛА-ТМР завершена. Если плата ЛА-ТМР длительное время не используется, желательно ее вынимать из компьютера и хранить в отапливаемых помещениях при температуре около 20 град. С.

6. Описание программы TMR.EXE, входящей в комплект поставки.

С помощью этой программы можно запрограммировать режимы каналов таймеров платы ЛА-ТМР, задать коэффициенты деления счетчиков. Формат обращения к программе такой:

tmr.exe /base330 /n=0 /mode=2 /cnt=100

/base330 - задание базового адреса,

/n=0 - задание канала счетчика-таймера P8254 (n=0..2),

/mode=2 - задание режима работы таймера, mode=0..5,

/cnt=100 - коэффициент деления, исходная тактовая частота -10 МГц или 1 МГц, если включен делитель на 10.

Удобнее пользоваться программой с помощью *.bat файла. Он может быть следующего вида:

```
tmr.exe /base330 /n=0 /mode=2 /cnt=%1
tmr.exe /base3 3 0 /n=1 /mode=3 /cnt=%1
tmr.exe /base3 3 0 /n=2 /mode=5 /cnt=%1
tmr.exe /base3 34 /n=0 /mode=2 /cnt=%1
mr.exe /base334 /n=1 /mode=3 /cnt=%1
tmr.exe /base334 /n=2 /mode=5 /cnt=%1
```

Первые три строки - программирование первого таймера (его трех каналов) $p=0\dots p=2$, вторые три строки - программирование второго таймера (при этом программа будет "воспринимать" программирование второго таймера как первого, но с другим базовым адресом). Указание /cnt=%1 приведет к запросу с клавиатуры коэффициента деления счетчика. Исходный текст tmr на языке C приведен ниже.

```
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
#include <string.h>
#define rlow ((char)rate)
#define rhig ((char)(rate>>8))
#define outB  outportb
#define inpB   inportb
#define TIMER_0      (Base+4)
#define TIMER_1      (Base+5)
#define TIMER_2      (Base+6)
#define TIMER_CTRL   (Base+7)

int Base=0x330;

void error(char * msg)
{
    printf("%s\n",msg);
//    if (buf0) tfree(buf0);
    exit(1);
}

int argum(char* argi, const char* cmp) // 0 if no equal
{
    int len=strlen(cmp);
    if (len>strlen(argi)) return 0;
    return !memcmp(argi,cmp, len);
}

char recognized[10]={0,0,0,0,0,0,0,0,0,0}; int getarg(int Nargs,
char** arg, const char* cmp, char* format, void* val)
{
    for(int i=1; i<Nargs ; i++) {
        if (recognized[i]) continue; if
        (argum(arg[i],cmp))
            { int k=sscanf(arg[i]+strlen(cmp),format,val);
            recognized[i]=k;
        }
    }
}
```

```

        return k;
    } } return 0;
}
int norecog(int Narg)
{
    for(int i=1; i<Narg; i++) {
        if (recognized[i]) continue; return i;
    }
    return 0;
}
void main (int Narg, char *arg[])
{
    int unsigned rate, mode, No, ok, err; // Base
    ok=getarg(Narg,arg, "/base=", "%x", &Base);
    if (!ok)
        ok=getarg(Narg,arg, "/base", "%x", &Base);
    if (!ok)
{ Base=0x300; printf("Base: Invalid. Setting to
default=%x", Base); }; // frequency
    ok=getarg(Narg,arg, "/cnt=", "%d", &rate);
    if (!ok)
        ok=getarg(Narg,arg, "/cnt", "%d", &rate);
    if (!ok) error("Counter: Invalid or absent."); // mode
    ok=getarg(Narg,arg, "/mode=", "%d", &mode);
    if (!ok)
        ok=getarg(Narg,arg, "/mode", "%d", &mode);
    if (!ok mode>5) error("Mode: Invalid or absent.");
// N
    ok=getarg(Narg,arg, "/n=", "%d", &No); if (!ok)
    ok=getarg(Narg,arg, "/n", "%d", &No); if (!ok No>2)
    error("Channel Number:
                                Invalid or absent.");
    err=norecog(Narg); if (err)
    {
        printf("\n*** %s ***", arg[err]);
        error("Unrecognized parameter.");
    }
// timers programming
    outB(TIMER_CTRL,
                                N Mode
                                (No<<6) | (mod<<1) | 0x30); // 00 11 010 0
    outB(TIMER_0+No, rlow);
    outB(TIMER_0+No, rhig); }

```

Приложение. Прерывания.

Средства генерации и обработки прерываний являются важной составной частью любой вычислительной системы, в том числе и персональной ЭВМ. Механизм прерываний обеспечивает эффективное взаимодействие устройств ввода/вывода с микропроцессором. Существуют аппаратные и программные прерывания. Аппаратные прерывания генерируются аппаратурой, либо с системной платы, либо с платы расширения, вставляемой в один из слотов ЭВМ. В данном случае мы будем преимущественно рассматривать аппаратные прерывания, вырабатываемые платой ввода/вывода LA-TMP.

Поддержка прерываний со стороны процессора 80x86.

Аппаратные прерывания не координируются с работой программного обеспечения. Идеология обработки прерываний схемотехнически заложена в устройство процессоров Intel 80x86. Один из выводов микросхемы процессора - INTR (INInterrupt Request) - специально предназначен для этой цели. Сигнал INTR позволяет внешним устройствам прерывать работу процессора по выполнению текущей программы и переключать его на работу по выполнению другой программы, называемой процедурой обработки прерывания. Сигнал INTR называют сигналом запроса на прерывание (маскируемым). Процессору можно запретить обработку прерываний по этому входу, если обнулить специальный флаг разрешения прерываний в регистре флагов процессора командой Ассемблера CLI (CLear Interrupt enable flag). Вход INTR опрашивается процессором в начале каждого процессорного такта и должен быть равен логической единице в течение, как минимум, двух процессорных тактов. Только в этом случае процессор, закончив выполнение текущей команды, перейдет к обработке поступившего прерывания.

Когда процессор приступает к выполнению процедуры обработки прерывания, он прежде всего выполняет два цикла шины, которые называются циклами подтверждения прерываний. Во время второго цикла процессор по шине данных считывает номер прерывания. По этому номеру в таблице векторов прерываний процессор определяет адрес начала процедуры обработки поступившего прерывания. Сигнал INTR поступает на процессор от выхода INT первого контроллера прерываний 8259.

Контроллер прерываний 8259.

Для управления аппаратными прерываниями в IBM PC используется микросхема программируемого контроллера прерываний Intel 8259.

Назначение и функционирование.

Программируемый контроллер прерываний (ПКП, Programmable Interrupt Controller, PIC) реализует векторную систему прерываний IBM-совместимого компьютера. Микросхема 8259А фирмы Intel (советский аналог КР580ВН59), а также ее модификации 8259А-2 и 8259А-8, поддерживает 8 уровней прерываний от восьми различных устройств. Основные функции контроллера:

- фиксация запросов на прерывания от 8 внешних источников;
- программное маскирование поступающих запросов;
- присвоение фиксированных или циклически изменяемых приоритетов входам контроллера, на которые поступают запросы;
- инициализация вызова процедуры обработки поступившего аппаратного прерывания.

Регистр запросов на прерывания (Interrupt Request Register, IRR) обслуживается через входы IR0 - IR7 контроллера. Сигнал на одном из входов IR0 - IR7 - это запрос на прерывание соответствующего уровня (0 - 7). В соответствии с сигналом запроса на прерывание схемой управления устанавливается соответствующий бит в регистре IRR.

Регистр состояния (регистр обрабатываемых запросов, In-Service Register, ISR) описывает в битах 0 - 7 прерывания каких уровней (0-7) в данный момент обрабатываются.

Регистр маскирования запросов на прерывания (Interrupt Mask Register, IMR) описывает, прерывания каких уровней в настоящий момент замаскированы. Единичное значение бита в IMR указывает на то, что прерывание соответствующего уровня при появлении запроса в IRR блокируется.

Схема обработки приоритетов (шифратор приоритетов, Priority Resolver) определяет, прерывание какого уровня в данный момент является наиболее приоритетным для выполнения.

Схема управления ПКП формирует сигнал запроса на прерывание, поступающий на вход INT (запрос на прерывание) микропроцессора. Если флаг IF регистра флагов процессора равен 1 (прерывания разрешены), процессор отвечает сигналом по линии INTA (подтверждение прерывания), после чего сбрасывается в 0 разряд IRR и устанавливается в 1 разряд ISR, соответствующие уровню обрабатываемого прерывания. После получения второго сигнала подтверждения от процессора по линии INTA, ПКП передает на шину данных 8-битовый номер прерывания. Данная последовательность работы схемы управления выполняется при подключении ПКП к системе с микропроцессорами 8088/8086.

Схема каскадирования отвечает за работу каскада из нескольких контроллеров. При подключении к ведущему контроллеру выход INT каждого ведомого подключается к одному из входов IR0 - IR7 ведущего. Далее этот сигнал передается ведущим на вход INT процессора. Когда процессор возвращает сигнал INTA, ведущий контроллер не только устанавливает бит в ISR и сбрасывает бит в IRR, но и выдает на свои выходы CAS0 - CAS2 номер уровня прерывания, к которому подключен ведомый, пославший запрос на прерывание. Сигналы по линии CAS0 - CAS2 принимаются всеми ведомыми, однако обрабатываются только тем, который подключен к линии IR с соответствующим номером.

Стандартное использование в IBM PC XT/AT.

Количество обслуживаемых внешних источников прерываний может быть увеличено путем каскадирования нескольких контроллеров. В IBM PC XT применяется один контроллер 8259, в PC AT - два, причем второй включен в режиме каскадирования запросов через линию IR2 первого контроллера. Таким образом, на системной плате PC AT поддерживается 16 уровней запроса прерываний IRQ (не путать с IR, обозначающим вывод микросхемы 8259), причем IRQ0-IRQ7 соответствуют выводам IR0-IR7 первого контроллера, а IRQ8-IRQ15 - выводам IR0-IR7 второго контроллера. В режиме фиксированных приоритетов (см. ниже), который является основным для IBM PC AT (устанавливается процедурой POST BIOS), приоритеты запросов IRQ расположены в следующем порядке (по убывающей): 0, 1, 2, 8, 9, 10, 11, 12, 13, 14, 15, 3, 4, 5, 6, 7.

Разные запросы IRQ используются разными стандартными устройствами (таймер, клавиатура, контроллер НГМД, и т.д.), поэтому фирмой IBM установлен стандарт использования запросов наиболее распространенными устройствами. Он различен для PC XT и PC AT. В следующей таблице в порядке убывания приоритетов приведено распределение запросов IRQ между устройствами ПЭВМ. Следует соблюдать осторожность при выборе уровня запросов для ЛА-TMP, чтобы выбранный IRQ и написанное пользователем программное обеспечение не приводило к взаимодействию с системными устройствами или к блокированию их работы. Лучше выбирать малоиспользуемый (или резервный) канал (2, 4, 7 для XT; 2, 3, 7, 9, 12, 15 для AT).

XT:		AT:	
0	Таймер	0	Таймер
1	Клавиатура	1	Клавиатура
2	Канал ввода/вывода	2	Канал ввода/вывода
		8	Часы реального времени
		9	Программно переводится в IRQ2
		10	Резерв
		11	Резерв
		12	Резерв
		13	Математический сопроцессор
		14	Контроллер жесткого диска
		15	Резерв
3	COM1	3	COM2
4	COM2	4	COM1
5	Жесткий диск	5	LPT2
6	Гибкий диск	6	Контроллер дисководов
7	LPT1	7	LPT1

Для вывода информации в ПКП используются 2 порта ввода-вывода из адресного пространства IBM PC AT. Порт с четным адресом (обычно это порт 20h) и порт с нечетным адресом (обычно 21h). Для PC AT, который оснащен двумя контроллерами ПКП, порты первого контроллера имеют те же адреса (20h и 21h), порты второго контроллера расположены по адресам A0h и A1h.

Режимы работы программируемого контроллера прерываний.

1. Режим фиксированных приоритетов (Fixed Priority, Fully Nested Mode). В этом режиме контроллер находится сразу после инициализации. Запросы прерываний имеют жесткие приоритеты от 0 до 7 (0 - высший) и обрабатываются в соответствии с приоритетами. Прерывание с меньшим приоритетом никогда не будет обработано, если в процессе обработки прерываний с более высокими приоритетами постоянно возникают запросы на эти прерывания.

2. Автоматический сдвиг приоритетов (Automatic Rotation). В этом режиме дается возможность обработать прерывания всех уровней без их дискриминации. Например, после обработки прерывания уровня 4 ему автоматически присваивается низший приоритет, при этом приоритеты для всех остальных уровней циклически сдвигаются и прерывания уровня 5 будут иметь в данной ситуации высший приоритет и, следовательно, возможность быть обработанными.

3. Программно-управляемый сдвиг приоритетов (Specific Rotation). Программист может сам передать команду циклического сдвига приоритетов ПКП, задав соответствующее управляющее слово. В команде задается номер уровня, которому

требуется присвоить максимальный приоритет. После выполнения такой команды

устройство работает так же, как и в режиме фиксированных приоритетов, с учетом их сдвига. Приоритеты сдвигаются циклически, таким образом если максимальный приоритет был назначен уровню

3. то уровень 2 получит минимальный и будет обрабатываться последним.

4. Автоматическое завершение обработки прерывания (Automatic End Of Interrupt, АЕОІ). В обычном режиме работы процедура обработки аппаратного прерывания должна перед своим завершением очистить свой бит в ISR специальной командой, иначе новые прерывания не будут обрабатываться ПКП. В режиме АЕОІ нужный бит в ISR автоматически сбрасывается в тот момент, когда начинается обработка прерывания нужной процедурой обработки и от нее не требуется издавать команду завершения обработки прерывания (ЕОІ). Сложность работы в данном режиме обуславливается тем, что все процедуры обработки аппаратных прерываний должны быть повторно входимыми, т. к. за время их работы могут повторно возникнуть прерывания того же уровня.

5. Режим специальной маски (Special Mask Mode). Данный режим позволяет отменить приоритетное упорядочение обработки запросов и обрабатывать их по мере поступления. После отмены режима специальной маски предшествующий порядок приоритетов уровней сохраняется.

6. Режим опроса (Polling Mode). В этом режиме аппаратные прерывания не происходят автоматически. Появление запросов на прерывание должно определяться считыванием ІRR. Данный режим позволяет так же получить от ПКП информацию о наличии запросов на прерывания и, если запросы имеются, номер уровня с максимальным приоритетом, по которому есть запрос.

Программирование ПКП.

Ниже описывается управление ПКП в ПЭВМ на базе микропроцессоров Intel 8086/8088/80286. Возможности, позволяющие включить ПКП в ПЭВМ, использующие другие процессоры, опущены.

Для вывода информации в ПКП используются 2 порта ввода-вывода из адресного пространства ІВМ РС АТ. Порт с четным адресом (обычно это порт 20п) и порт с нечетным адресом (обычно 21п). Для РС АТ, который оснащен двумя контроллерами ПКП, порты первого контроллера имеют те же адреса (20п и 21п), порты второго контроллера расположены по адресам 70h и 71п. После процедуры инициализации ПКП готов к работе в заданном режиме. Для изменения режимов работы, задаваемых при инициализации, требуется переинициализировать его заново.

В процессе работы с ПКП Вы можете без переинициализации:

- маскировать и размаскировать аппаратные прерывания;
- изменять приоритеты уровней;
- задавать команду завершения обработки аппаратного прерывания;
- устанавливать и сбрасывать режим специальной маски;
- переводить контроллер в режим опроса и считывать состояние регистров ІSR и ІRR; для этого Вам потребуется вывести в порты ПКП одно из трех слов рабочих приказов ОСW1 - ОСW3.

Далее везде формат записываемых данных будет иметь дополнительную сигнатуру А0, для индикации четного (А0=0) или нечетного (А0=1) адреса порта, в который производится запись. Формат первого слова рабочих приказов ОСW1:

А0	7	6	5	4	3	2	1	0
1	М7	М6	М5	М4	М3	М2	М1	М0

Единичное значение одного из битов M0 - M7 означает, что прерывания соответствующего уровня (IR0 - IR7) маскируются и не будут обрабатываться контроллером.

Второе слово рабочих приказов (OCW2) предназначено для вывода команды завершения обработки аппаратного прерывания (EOI), циклического сдвига и явного изменения приоритетов уровней. Назначение битов OCW2 следующее:

A0	7	6	5	4	3	2	1	0
0	R	SL	EOI	0	0	L2	L1	L0

L2, L1, L0: Эти 3 бита определяют номер уровня прерывания, если он требуется в команде.

EOI: =1 - команда завершения обработки аппаратного прерывания R, SL: 0 0

Используется вместе с EOI=1 0 1

Специфицированный EOI (сбрасывает бит, определяемый полями L0 -L2 в ISR)

1 0 Циклический сдвиг приоритетов влево на одну позицию 1 1 Назначение низшего приоритета уровню, определяемому полями L0 - L2.

Как уже говорилось, процедура обработки аппаратного прерывания должна перед своим завершением очистить свой бит в ISR выводом команды завершения обработки прерывания (End Of Interrupt, EOI) .

Существует два варианта команды EOI: обычный и специфицированный EOI. Обычный EOI очищает бит в ISR, соответствующий прерыванию с максимальным приоритетом. Специфицированный EOI (R=0, SL=1, EOI=1, L0 - L2 равно номеру уровня прерывания) очищает в ISR бит, соответствующий прерыванию с номером, указанным в L0 - L2 независимо от его приоритета. Команды с битом R=1 позволяют изменить приоритеты уровней. Циклический сдвиг приоритетов сдвигает приоритеты влево на единицу, при этом, если после обычного распределения приоритетов, издать команду циклического сдвига, уровень 0 получит низший приоритет, уровень 1 - наивысший, уровень 2 - следующий за ним и т.д. Команда явного назначения низшего приоритета одному из уровней изменяет приоритеты остальных уровней циклически. Таким образом, если Вы зададите низший приоритет уровню 5, то уровень 6 получит наивысший. Третье слово рабочих приказов OCW3 позволяет установить и отменить режим специальной маски, перевести контроллер в режим опроса и прочитать содержимое IRR и ISR. Назначение битов OCW3 приведено на рисунке:

A0	7	6	5	4	3	2	1	0
0	0	ESMM	SMM	0	1	P	RR	RIS

ESMM, SMM:

1 0 - Отменить режим специальной маски

1 1 - Установить режим специальной маски Единичное значение бита P (бит опроса, Polling Bit) переводит контроллер в режим опроса. Если после этого считать данные из порта с четным адресом, в регистр AL загрузится байт следующего содержания:

7	6	5	4	3	2	1	0
I	0	0	0	0	L2	L1	L0

Если I=1, значит имеются запросы на прерывания и тогда L0 - L2 - это номер уровня с наивысшим приоритетом, по которому имеется запрос на прерывание.

Если P=0, Вы можете считать информацию из ISR или IRR. Для этого необходимо издать команду чтения ISR или IRR

P	PR	RIS:	
0	1	0	Чтение IRR
0	1	1	Чтение ISR

и затем считать значение из порта с нечетным адресом. Единичное значение бита ESMM позволяет в зависимости от значения бита SMM установить или отменить режим специальной маски.

Управление прерываниями в IBM PC.

Процедуры обработки прерываний (ПОП) это готовые подпрограммы, на выполнение которых процессор переходит при обработке запросов IRQ. Термин прерывание отражает тот факт, что если прерывания не запрещены (установлен Interrupt enable Flag) и конкретное аппаратное прерывание не замаскировано (соответствующий бит в IMR контроллера 8259 равен нулю) то процессор приостановит выполнение своего текущего задания, к которому вернется лишь выполнив код ПОП. После этого прерванная (фоновая) программа продолжит свою работу "как ни в чем не бывало": она ничего не узнает о том, что процессор отвлекался. Это не так, если пользователем предусмотрен обмен информацией между прерываемой программой и ПОП. Такой обмен можно организовать через общие переменные. При написании программ для ЛА-ТМР это может быть очень полезно, например, когда ПОП занимается записью данных в циклический буфер, а фоновая программа производит обработку данных из этого буфера. Такой режим ПОП полностью аналогичен процедуре обработки прерываний от клавиатуры, которая заносит коды нажимаемых клавиш в циклический буфер области данных BIOS. Любая пользовательская программа, использующая ввод с клавиатуры, либо напрямую, либо опосредованно (через DOS) обменивается с ПОП информацией об адресах головы и хвоста используемой части буфера (между головой и хвостом находятся данные, которые уже введены ПОПой, но еще не извлечены пользователем для обработки). ПОП "наращивает" данные со стороны головы (увеличивая при этом указатель головы), пользователь "откусывает" их от хвоста (увеличивая при этом указатель хвоста). Так как буфер циклический, то пользователь должен вовремя остановиться, чтобы не пытаться извлекать данные из опустошенного буфера (указатели головы и хвоста совпадают), а ПОП должен прекратить ввод кодов с клавиатуры при переполнении буфера, когда голова начинает упираться в хвост (чтобы не затереть уже введенные, но еще не обработанные данные), выдав при этом сообщение об ошибке или как-то иначе, попросив пользователя прекратить бесплодные попытки ввести что-то с клавиатуры. При переполнении буфера BIOS обычно начинает "пищать" динамиком. Таким образом организованное взаимодействие позволяет полностью разделить программную поддержку аппаратуры (ПОП) и математическую обработку данных, не только облегчив при этом структурирование программы, но и обеспечив асинхронность обработки данных.

Для того, чтобы иметь возможность вернуться точно в нужное место прерванной (фоновой) программы, адрес этого места (CS:IP) запоминается процессором на стеке, вместе с регистром флагов. Затем в CS:IP загружается адрес ПОП и ей передается управление. ПОП иногда называют драйвером прерывания. Она должна завершаться инструкцией IRET (возврат из прерывания), которая завершает процесс, начатый прерыванием, возвращая старые значения CS:IP и регистра флагов, тем самым давая фоновой программе возможность продолжить свое выполнение из прежнего

состояния. Помимо этого, процедура обработки аппаратного прерывания должна восстановить перед возвратом управления содержимое всех использовавшихся ею регистров, тем самым не оставляя следов взаимодействия с процессором.

Адреса программ прерываний называют векторами. Каждый вектор имеет длину четыре байта. В первом слове хранится значение IP, а во втором - CS. Младшие 1024 байт памяти содержат вектора прерываний, таким образом имеется место для 256 векторов. Вместе взятые они называются таблицей векторов. Вектор для прерывания 0 начинается с ячейки 0000:0000, прерывания 1-с адреса 0000:0004, 2 - с 0000:0008, и т.д. Например, если в таблице по адресу 0000:0084 (прерывание 21H -функции DOS) содержится значение 420002D8, то адрес ПОП 21H будет равен (CS:IP) D802:0042. Запросы на прерывания DRQ0-DRQ7 соответствуют векторам прерываний от 8H до 0FH; для PC AT запросы на прерывания 8-15 обслуживаются векторами от 70H до 77H.