

Лабораторная работа №5 Обработка прерываний

Цель работы:

1. Изучить особенности системы прерываний микроконтроллеров;
2. Разработать программу-обработчик прерывания.

Краткие теоретические сведения

Прерывание - процесс переключения ЦП с одной программы на другую по внешнему сигналу с сохранением информации для последующего возобновления прерванной программы.

Основная цель введения прерываний – реализация асинхронного режима работы и распараллеливания работы отдельных устройств вычислительного комплекса. Механизм прерываний реализуется аппаратно-программными средствами.

Этапы:

- ПУ формирует запрос прерывания.
- Запросы могут поступать асинхронно.
- Запросы сохраняются в специальном регистре.
- Состояние регистра анализируется программно или аппаратно.
- Либо информация о месте возникновения запроса передаётся в виде адреса ячейки памяти (векторная система прерываний).
- Вызывается программа-обработчик прерывания.

Каждое событие, требующее прерывания, сопровождается сигналом прерывания, оповещающим об этом вычислительную машину, и называемым запросом прерывания.

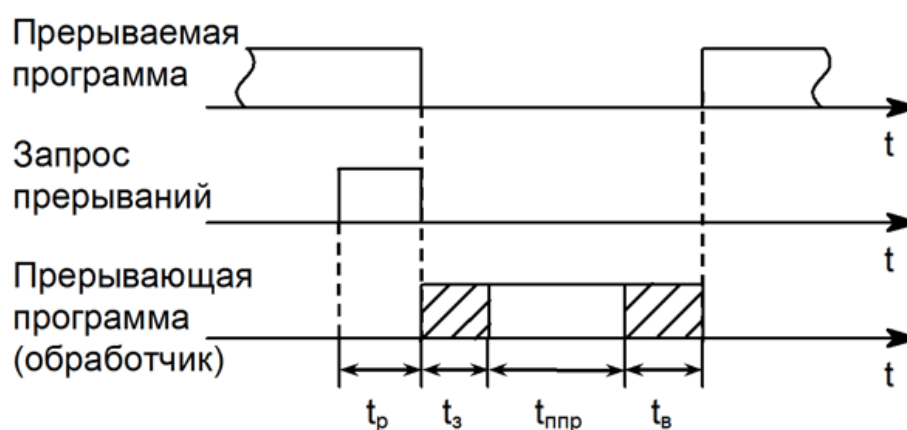
Состояние программы представляет собой совокупность состояний всех запоминающих элементов в соответствующий момент времени (например, после выполнения последней команды). При возникновении прерывания микроконтроллер сохраняет в стеке содержимое счетчика команд и загружает в него адрес соответствующего вектора прерывания. Последней командой подпрограммы обработки прерывания должна быть команда, которая осуществляет возврат в основную программу и восстановление предварительно сохраненного счетчика команд. Во время выполнения обработчика прерывания некоторая информация может подвергнуться изменению. Поэтому при переходе к обработчику прерывания необходимо сохранить элементы, подвергающиеся изменению. Набор таких элементов представляет собой вектор состояния программы. При этом другая информация о состоянии ячеек памяти не существенна или может быть восстановлена программным путем.

Вектор начального состояния содержит всю необходимую информацию для начального запуска программы. Во многих случаях вектор начального состояния содержит только один элемент – начальный адрес запускаемой программы.

Вектор прерывания является вектором начального состояния прерывающей программы (обработчика) и содержит всю необходимую информацию для перехода к обработчику, в том числе его начальный адрес. Каждому типу

прерываний соответствует свой вектор прерывания, который инициализирует выполнение соответствующего обработчика. Обычно векторы прерывания хранятся в специально выделенных фиксированных ячейках памяти с короткими адресами, представляющих собой таблицу векторов прерываний. Для перехода к соответствующей прерывающей программе процессор должен располагать вектором прерывания и адресом этого вектора. По этому адресу, как правило, находится команда безусловного перехода к подпрограмме обработки прерывания.

Как правило, управление запоминанием и возвратом возложено на обработчик прерывания. В этом случае обработчик состоит из трех частей – подготовительной (пролог) и заключительной (эпилог), обеспечивающих переключение программ, и собственно прерывающей программы, выполняющей затребованные запросом операции. Время реакции определяется как временной интервал от момента поступления запроса прерывания до начала выполнения прерывающей программы.



- t_p – время реакции системы на прерывание;
- t_3 – время запоминания состояния прерываемой программы;
- $t_{ппр}$ – время собственно прерывающей программы;
- $t_в$ – время восстановления состояния прерванной программы

При наличии нескольких источников запросов должен быть установлен определенный порядок обслуживания поступающих запросов, называемый приоритетными соотношениями или дисциплиной обслуживания. Совокупность всех возможных типов прерывания процессора представляет собой систему прерывания микроконтроллера. Дисциплина обслуживания определяет, какой из нескольких запросов, поступивших одновременно, подлежит обработке в первую очередь, и имеет ли право данный запрос прерывать тот или иной обработчик прерывания.

В случае если во время обработки прерывания поступает запрос на прерывание с более высоким уровнем приоритета, управление передается обработчику прерывания более высокого приоритета, при этом работа обработчика прерывания с более низким уровнем приоритета приостанавливается. Возникает вложенность прерываний. Максимальное число

программ, которые могут приостанавливать друг друга называется глубиной прерываний.

Ниже приведён алгоритм выполнения команды процессором, поддерживающим ввод-вывод с прерыванием.



Обработка прерываний в ADuC842

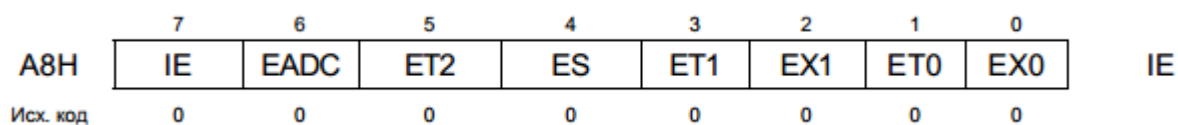
Система прерываний микроконвертера ADuC842 включает восемь источников прерываний, которые обслуживаются в соответствии с двухуровневой системой приоритетов. Источники прерываний, порядок их опроса при одинаковом установленном уровне приоритета, адреса векторов (таблицы прерываний) приведены в следующей таблице:

Прерывание	Источник прерывания	Адреса вектора	Уровень приоритета
PSMI	Монитор напряжения питания	43H	1 (наивысший)
WDS	Сторожевой таймер	5BH	2
IE0	Внешнее прерывание INTO	03H	2
ADCI	Конец преобразования АЦП	33H	3
TFO	Переполнение Таймера 0	0BH	4
IE1	Внешнее прерывание INT1	13H	5
TF1	Переполнение Таймера 1	1BH	6
I2CI/ISPI	Прерывание синхронного последовательного интерфейса	3BH	7
RI/TI	Прерывание асинхронного интерфейса UART	23H	8
TF2/EXF2	Прерывание от Таймера 2	2BH	9
Ti1	Прерывание счетчика времени TIC	53H	11 (нижайший)

Для того, чтобы прерывания могли быть обслужены, необходимо выполнить следующие подготовительные действия:

- Расположить процедуру обслуживания прерывания по адресу, указанному в таблице переходов (соответствующего прерывания).
- Установить в регистре IE бит разрешения всех прерываний (EA) в «1»;
- Установить в регистре IE или IEIP2 бит разрешения индивидуального прерывания в «1».

Для разрешения и установки приоритетов прерываний используются регистры IE, IP и IEIP2. В регистрах IE и IP могут адресоваться отдельные биты, а регистр IE2 адресуется только целиком как байт. Регистр IE содержит общий и индивидуальные бит разрешения всех источников прерываний. Для разрешения прерываний соответствующие биты должны быть установлены в «1».



Бит	Мнемоника	Описание
IE.7	IE	Общий бит разрешения прерываний. Должен быть установлен для обслуживания любого источника прерывания. Если EA=0, все прерывания запрещены.
IE.6	EADC	Бит разрешения прерывания АЦП.
IE.5	ET2	Бит разрешения прерывания по переполнению Таймера 2.
IE.4	ES	Бит разрешения прерывания от последовательного порта UART.
IE.3	ET1	Бит разрешения прерывания по переполнению Таймера 1.
IE.2	EX1	Бит разрешения внешнего прерывания INT1.
IE.1	ETO	Бит разрешения прерывания по Переполнению Таймера 0.
IE.0	EXO	Бит разрешения внешнего прерывания INTO EXO.

Идентификация прерываний в Keil C51:

Номер прерывания	Адрес обработчика	Номер прерывания	Адрес обработчика
0	0x003	10	0x053
1	0x00B	11	0x05B
2	0x013	12	0x063
3	0x01B	13	0x06B
4	0x023	14	0x073
5	0x02B	15	0x07B
6	0x033	16	0x083
7	0x03B	17	0x08B
8	0x043	18	0x093
9	0x04B	19	0x09B

Пример обработки прерывания от UART:

```
void int_uart (void) interrupt 4 // адрес обработчика - 0x023
{
    P0=0x15;
```

```

}
int main(){
    EA=1;    //разрешаем прерывания
    ES=1;    //разрешаем прерывание от UART'a
    ...
    while (1)
    {
        P0=i;
        t1();
        i*=2;
        if (i == 16) i = 1;
    }
}

```

Порядок действий процессора:

```

t0ISR:
    ORG 0Bh
    PUSH PSW
    PUSH ACC
    . . .
    POP ACC
    POP PSW
    RETI

```

Задание на лабораторную работу:

Разработать программу, обрабатывающую прерывание с кнопки. Например, при нажатии останавливаются бегущие огни на светодиодах.

Содержание отчёта:

1. Цель работы;
2. Таблица векторов прерываний микроконтроллера ADuC842;
3. Листинг разработанной программы;
4. Выводы по проделанной работе.