1. **Особенности архитектуры микроконтроллера MSP430;**

− 16-тиразрядная RISC архитектура;

− ортогональная архитектура, при которой каждая команда пригодна для каждого режима адресации;

− 27 основных команд + 24 эмулированных;

− 7 согласованных способов адресации;

− полный доступ ко всем регистрам, включая счетчик команд, регистр состояния, и указатель стека;

− генератор шести основных констант;

− однотактные регистровые операции;

− большой размер регистрового файла, уменьшающий количество обращений к памяти (16 регистров = 4 специальных + 12 РОН);

− 20-битная шина адреса, 16-битная шина данных;

− Фон-Неймановская адресная шина общей памяти и шина данных памяти;

− пересылки память-память без промежуточного сохранения в регистре.

1. **Физические принципы работы портов;**

Микроконтроллер имеет 63 линии (ножки, вывода, пина) ввода/вывода, объединенные в 8-миразрядные порты ввода/вывода P1, P2, …, P8, PJ (для P8 программно доступно 3 бита, для PJ – 4 бита). Каждый из 63 пинов могут работать в режиме цифрового ввода/вывода, порты P1 – P7 могут быть сконфигурированы в режиме входа/выхода внутренней периферии микроконтроллера, пины порта PJ – могут быть сконфигурированы в качестве линий JTAG-интерфейса.

− независимые индивидуально программируемые входы/выходы;

− любые комбинации входа или выхода;

− индивидуально конфигурируемые прерывания от Р1 и Р2;

− раздельные регистры данных для входов и выходов.

1. **Типы регистров, доступных портам, и их назначение;**

− PxIN – каждый бит регистра отражает величину входного сигнала на соответствующей ножке (пине) ввода/вывода, когда она сконфигурирована на функцию ввода/вывода;

− PxOUT – каждый бит в регистре содержит значение, которое будет выведено на соответствующую ножку ввода/вывода, сконфигурированную на функцию ввода/вывода и имеющую направление на вывод;

− PxDIR – каждый бит в регистре позволяет независимо от выбранной для этой ножки функции (ввод/вывод или периферия) выбрать направление соответствующей ножки: 0 – вход, 1 – выход;

− PxREN – каждый бит в регистре включает или отключает подтягивающий резистор соответствующего ножки ввода / вывода.;

− PxDS – каждый бит в регистре осуществляет выбор допустимой выходной силы тока для соответствующей ножки: 0 – пониженная (по умолчанию), 1 – полная;

− PxSEL – каждый бит определяет, как будет использована ножка – в качестве порта ввода/вывода или в качестве функции периферийного модуля: 0 – I/O, 1 – периферия;

− PxIV – генерирует значение для изменения счетчика команд, соответствующее прерыванию с максимальным приоритетом;

− PxIE – каждый бит в регистре разрешает прерывания по соответствующей линии ввода/вывода;

− PxIES – каждый бит в регистре определяет направление перепада для генерации запроса на прерывание: 0 – по фронту, 1 – по спаду;

− PxIFG – каждый бит в регистре соответствует флагу запроса на прерывание по соответствующей линии ввода/вывода

1. **Общая теория по прерываниям (прерывание, вектор прерывания, таблица векторов прерываний, типы прерываний).**

Различают следующие виды прерываний:

− системные немаскируемые (сигнал RST/NMI в режиме NMI, сигнал от сторожевого таймера, сбой тактового генератора, ошибка доступа Flashпамяти);

− пользовательские немаскируемые (сбой напряжения питания от PMM, доступ к несуществующей памяти, события буфера JTAG);

− маскируемые прерывания (генерируются периферийными модулями).

Микроконтроллер имеет следующую структуру прерываний:

− векторные прерывания без необходимости опроса;

− векторы прерываний расположены вниз от адреса 0FFFEh.

Для обслуживания прерываний используются подпрограммы обработки прерываний, начальные адреса которых сведены в таблицу векторов прерываний. Таблица векторов прерываний содержат 64 вектора. В каждом векторе должен быть записан 16-битных адрес обработчика прерываний, на который будет передано управление в случае возникновения прерывания.

При одновременном возникновении нескольких прерываний управление передается по вектору, имеющему более высокий приоритет. В MSP микроконтроллерах вектора с большим адресом имеют более высокий приоритет.

Кратко рассмотрим, как происходит порядок обработки прерывания. Задержка от возникновения запроса на прерывание до начала выполнения обработчика составляет 6 циклов. При возникновении прерываний выполняются следующие действия:

1. Завершается выполнение текущей инструкции.

2. Счетчик команд PC, указывающий на следующую инструкцию, сохраняется в стеке;

3. Регистр состояния SR сохраняется в стеке.

4. Выбирается прерывание с наивысшим приоритетом.

5. Сбрасывается флаг запроса на прерывания, если данному вектору соответствует единственному источник. Если источников несколько, флаг завтра на прерывание необходимо сбрасывать программно.

6. Все биты регистра состояния SR кроме SCG0 сбрасываются в 0; т.к. бит GIE = 0 маскируемые прерывания запрещаются.

7. В счетчик команд PC загружается содержимое выбранного вектора прерывания.

Возврат из прерывания выполняется командой RETI, которая выполняется за 5 циклов и загружает из стека SR, PC.

1. **Особенности работы с кнопками и диодами.**

Пользователю программно доступны две кнопки S1 и S2, P1.7 и P2.2 соответственно. Также программно доступны 8 светодиодов, три из которых (LED1 – LED3) размещены рядом с кнопками и подключены соответственно к выводам P1.0, P8.1, P8.2. Еще 5 светодиодов (LED4 – LED8) размещаются в блоке сенсорных кнопок и подключены к выводам P1.1 – P1.5 соответственно.