Зміст

*Изм.*

*Лист*

*№ докум.*

*Підпис*

*Дата*

*Стор*

*1*

*ІАЛЦ.462637.002.ПЗ*

*Розроб.*

*Стешин Л.В,*

*Перевір.*

*Ткаченко В.В В.ВВ.В.*

*Реценз.*

*Н. Контр.*

*Затверд.*

Ткаченко В.В

*Пояснювальна записка*

*Стор.*

*Стор*

*4*

*НТУУ “КПІ” ФІОТ*

Вступ…………………………………………………………………………...2

1. Обзор однокристальних мікропроцесорів…………………………………….3
2. Мікропроцесорна система на базі РІС 16С84………………………………...8
   * + 1. 2.1. Розробка структурної схеми МПС………………………………………….8
   1. 2.2. Опис мікроконтроллера РІС 16С84………………………………………...12
      1. 2.2.1. Захальна характеристика………………………………………………..12
      2. 2.2.2. Структурна схема мікроконтроллера…………………………………..13
      3. 2.2.3. Система команд мікроконтроллера ……………………………………15
      4. 2.2.4. Типи адресації……………………………………………………………29
   2. 2.3. Підключениня пам’яті ………………………………………………………30
   3. 2.4. Контроллери ПП и ПДП…………………………………………………….35
   4. 2.5. Программований периферійний адаптер…………………………………..37
   5. 2.6. Підключення додаткових портів…………………………………………....39
3. Розробка программного забезпечення………………………………………...40
   1. 3.1. Структура программи ……………………………………………………….41
   2. 3.2. Процедура множення…………………………………………………….......44
   3. 3.3. Процедура ділення…………………………………………………………...46
   4. 3.4. Процедура обчислення квадратного кореня………………………………. 47
   5. 3.5. Алгоритм програми обчислення функції………………………………….. 49
   6. 3.6. Програма обчислення функції……………………………………………... 50
4. Розробка функціональної схеми ………………………………………………70
5. Розробка принципіальної схеми ………………………………………………74

Висновки……………………………………………………………………….75

Список використаної літератури……………………………………………..76

*записка*

Вступ

1. **Обзор однокристальных микропроцессоров**

Мікропроцесор як функціональний пристрій ЕОМ забезпечує ефективне автоматичне виконання операцій обробки цифрової інформації відповідно до заданого алгоритму. Для вирішення багатьох завдань в різних областях застосувань, мікропроцесор повинен володіти алгоритмічно повною системою команд (операцій).

При проектуванні МП вирішуються завдання визначення наборів команд, що виконуються програмним або апаратурним способом на основі заданої системи мікрокоманд. Апаратурна реалізація складних команд дає можливість збільшити швидкодію мікропроцесора, але вимагає значних апаратурних ресурсів кристала інтегральної схеми МП. Програмна реалізація складних команд дозволяє забезпечувати програмування складних завдань, змінювати кількість і особливості виконання складних команд. Проте швидкість виконання програмних команд нижча за швидкість виконання апаратурно-реалізованих команд.

Для опису МП як функціональних пристроїв необхідно охарактеризувати формат оброблюваних даних і команд, кількість, тип і гнучкість команд, методи адресації даних, число внутрішніх регістрів загального призначення і регістрів результату, можливості організації і адресації стека, параметри віртуальної пам'яті і інформаційну ємкість пам'яті, що прямо адресується. Велике значення мають засобу побудови системи переривань програм, побудови ефективних систем ввода—вывода даних і розвиненого інтерфейсу.

МП можуть бути реалізовані на різній фізичній основі: на електронній, оптоелектронній, оптичній, біологічній і навіть на пневматичній або гідравлічній.

За призначенням розрізняють універсальні і спеціалізовані мікропроцесори.

Універсальні МП призначені для вирішення широкого круга завдань. При цьому їх ефективна продуктивність слабо залежить від проблемної специфіки вирішуваних завдань В системі команд МП закладена алгоритмічна універсальність, що означає, що виконуваний машиною склад команд дозволяє отримати перетворення інформації відповідно до будь-якого заданого алгоритму.

До універсальних МП відносяться і секційні мікропроцесори, оскільки для них система команд може бути оптимізована в кожному приватному проекті створення секційного мікропроцесора.

Ця група МП найбільш численна, в неї входять такі комплекти як К580, Z80, Intel 80Ч86, К582, К587, К1804,к1810 і ін.

Спеціалізовані МП призначені для вирішення певного класу завдань, а іноді тільки для вирішення одного конкретного завдання. Їх істотними особливостями є простота управління, компактність апаратурних засобів, низька вартість і мала потужність споживання.

Спеціалізовані МП мають орієнтацію на прискорене виконання певних функцій, що дозволяє різко збільшити ефективну продуктивність при вирішенні тільки певних завдань.

Серед спеціалізованих мікропроцесорів можна виділити різні мікроконтролери, орієнтовані на виконання складних послідовностей логічних операцій; математичні МП, призначені для підвищення продуктивності при виконанні арифметичних операцій за рахунок, наприклад матричних методів їх виконання; МП для обробки даних в різних областях застосувань і так далі

За допомогою спеціалізованих МП можна ефективно вирішувати нові складні завдання паралельної обробки даних. Наприклад, вони дозволяють здійснити складнішу математичну обробку сигналів, чим широко використовувані методи кореляції, дають можливість в реальному масштабі часу знаходити відповідність для сигналів форми, що змінюється, шляхом порівняння їх з різними еталонними сигналами для ефективного виділення корисного сигналу на тлі шуму і так далі

По вигляду оброблюваних вхідних сигналів розрізняють цифрові і аналогові мікропроцесори.

Самі МП є цифровими пристроями обробки інформації.

Проте у ряді випадків вони можуть мати вбудовані аналого-цифрові і цифро-аналогові перетворювачі. Тому вхідні аналогові сигнали передаються в МП через перетворювач в цифровій формі, обробляються і після зворотного перетворення в аналогову форму поступають на вихід.

З архітектурної точки зору такими мікропроцесорами є аналогові функціональні перетворювачі сигналів. Вони виконують функції будь-якої аналогової схеми

(наприклад, проводять генерацію коливань, модуляцію, зсув, фільтрацію, кодування і декодування сигналів в реальному масштабі часу і т. д., замінюючи складні схеми, що складаються з операційних підсилювачів, котушок індуктивності, конденсаторів і так далі). При цьому застосування аналогового МП значно підвищує точність обробки аналогових сигналів і їх відтворюваність, а також розширює функціональні можливості за рахунок програмного “настроювання” цифрової частини мікропроцесора на різні алгоритми обробки сигналів.

Зазвичай у складі однокристальних аналогових МП є декілька каналів аналого-цифрового і цифро-аналогового перетворення. У аналоговому мікропроцесорі розрядність оброблюваних даних досягає 24 битий і більш. Велике значення приділяється збільшенню швидкості виконання арифметичних операцій.

Відмінна риса аналогових МП - це здібність до переробки великого об'єму числових даних, тобто до виконання операцій складання і множення з великою швидкістю, при необхідності навіть за рахунок відмови від операцій переривань і переходів. Аналоговий сигнал, перетворений в цифрову форму, обробляється в реальному масштабі часу і передається на вихід зазвичай в аналоговій формі через цифро-аналоговий перетворювач. При цьому згідно теоремі Котельникова частота квантування аналогового сигналу повинна удвічі перевищувати верхню частоту сигналу.

Одним з напрямів подальшого вдосконалення аналогових МП є підвищення їх універсальності і гнучкості. Тому разом з підвищенням швидкості обробки великого об'єму цифрових даних розвиватимуться засоби забезпечення розвинених обчислювальних процесів обробки цифрової інформації за рахунок реалізації апаратурних блоків переривання програм і програмних переходів.

По кількості виконуваних програм розрізняють одно- і багатопрограмні мікропроцесори.

У однопрограммных МП виконується тільки одна програма. Перехід до виконання іншої програми відбувається після завершення поточної програми.

У много- або мультипрограмних МП одночасно виконується декілька (звичайні декілька десятків) програм. Організація мультипрограмної роботи мікропроцесорних систем, що управляють, наприклад, дозволяє здійснити контроль за станом і управлінням великим числом джерел або приймачів інформації.

По числу БІС в мікропроцесорному комплекті розрізняють однокристальні, багатокристальні і багатокристальні секційні МП.

Процесори навіть найпростіших ЕОМ мають складну функціональну структуру, містять велику кількість електронних елементів і безліч розгалужених зв'язків. Реалізувати принципову схему звичайного процесора у вигляді однієї або декількох БІС практично неможливо із-за специфічних особливостей БІС (обмеженість кількості елементів, складність виконання розгалужених зв'язків, порівняльне невелике число виводів корпусу). Тому необхідно змінювати структуру процесора так, щоб повна принципова схема або її частини мали кількість елементів і зв'язків, сумісну з можливостями БІС. При цьому МП набувають внутрішньої магістральної структури, тобто в них до єдиної внутрішньої інформаційної магістралі підключаються всі основні функціональні блоки (арифметико-логічний, робочих регістрів, стека, переривань, інтерфейсу, управління і синхронізації і ін.).

2.Мікропроцесорна система на базі РІС 16С84

* + - 1. 2.1. Розробка структурної схеми МПС

МПС містить:

* однокристальний мікроконтролер РІС16С84;
* шина адресу и данных роз'єднана
* об'єм пам'яті – 1Кб (необхідно підключити 16 сторінок пам’яті;
* зовнішніх пристроїв, можен з яких має фіксований адрес в адресному просторі периферійних пристроїв і децентралізовані контролери приорітетных переривань і прямого доступу до пам’яті;
* програмований периферійний адаптер;

Ядром системи являеться мікроконтролер РІС16Х84. Він виконує основні арифметико-логічні перетворення цифрової інформації в системі. До нього підключається решта всіх пристроїв. Окрім арифметико-логічного пристрою, даний мікроконтролер містить пам'ять об'ємом 1кб, восьмирозрядний таймер-лічильник, чотири джерела прериванія:внешній (вхід Int,переполненіє таймера RTCC,прериваніє при зміні сигналів на лініях порту В, по завершенню запису даних в пам'ять EEPROM), чотири типи генераторів: RC генератор;обычный кварцевий резонатор; высокочастотный кварцевий резонатор; экономичный низькочастотний кристал; а також два порта: семирозрядний RB, п’ятирозрядний RA, за допомогою яких до нього підключається решта всіх блоків схеми.

У мікроконтролер записується програма, написана за допомогою спеціалізованої системи команд, що розглядається детально далі. На підставі цієї програми мікроконтролер проводить перетворення інформації, що поступає до нього через порти, і через них же видає необхідні сигнали. Для того, щоб змінити поведінку системи, необхідно перепрограмувати мікроконтролер.

Оскільки за технічним завданням шина адреси і даних роздільна, а для роботи з пам'яттю необхідно на ША подавати адресу, необхідний буферний регістр для зберігання адреси. На структурній схемі він позначений як регістр адреси (8-міразрядний.).

Пам'ять підключена до мікроконтролера сторінками по 128 байт. Для вибору конкретної сторінки з 16, номер сторінки подається з порту RA на дешифратор. 16 сигналів з дешифратора (1-16) подається на вхід CS модулів пам'яті даних. 16-а сторінка відсутня, оскільки вона повністю відведена під зовнішні пристрої, тому 16-й сигнал з дешифратора використовується як прапор того, що потрібний робота із зовнішніми пристроями. На сторінки пам'яті даних йде 7 молодших розрядів адреси (з регістра адреси), CS з дешифратора номера потрібної сторінки, сигнали R і W , що підсилюються підсилювачами, з відповідних виводів мікроконтролера і слово даних з шини. На виході отримуємо у разі операції читання слово з пам'яті за заданою адресою.

Адреси регістрів програмованого периферійного адаптера входять в загальний адресний простір пам'яті даних, розміщуються в 16-ій сторінці з пам'яті. Для вибору мікросхеми ППА через вхід CS побудований спеціальний селектор адреси. Він спрацьовує (тобто видає 1) тільки якщо вибрана 16-а сторінка пам'яті і в регістрі адреси (молодших 8 бітах) встановлене одне із значень 79h, 7Ah, 7Bh, 7Ch. Молодші 2 біта адреси додатково поступають на спеціальні входи мікросхеми для вибору регістра, з яким необхідно працювати. Сигнали R і W поступають з відповідних виходів мікроконтролера.

За технічним завданням в мікропроцесорній системі 60 зовнішніх пристрої. Їх регістри (стану і даних) входять в загальний адресний простір зовнішньої пам'яті даних. Для вибору кожного з пристроїв за допомогою входу CS побудований свій селектор адреси. На нього поступає 7 старших біт адреси з регicтра адреси і 16-й вихід дешифратора номера сторінки. Молодший біт адреси поступає на сам зовнішній пристрій для вибору регістра даних або регістра стану. Контроллери прямого доступу до пам'яті і пріоритетних переривань за технічним завданням децентралізованниє. Тобто в кожному пристрої є своя частина контроллерів. Відповідні входи/виходи організовуються в дейзі-ланцюжок. Таким чином, фіксуються пріоритети пристроїв - щоб поміняти пріоритети, прійдется змінювати фізичний порядок підключення.

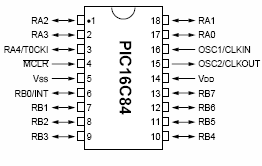
На вхід дейзи-ланцюжка контроллера пріоритетних переривань подається сигнал перевірки переривань. Якщо пристрою, до якого дійшов цей сигнал, потрібне переривання і воно готове і не замасковане, воно видає сигнал вимоги переривання на мікроконтролер і блокує розповсюдження сигналу далі. Інакше сигнал проходить по ланцюжку наступному пристрою. Якщо була вимога переривання, мікроконтролер видає сигнал підтвердження переривання. І в цьому випадку на шину адреси і даних видається вектор переривання, по якому знаходиться адреса обробника переривання в пам'яті програм.

Дейзі-ланцюжок контролера прямого доступу до пам'яті працює аналогічно ланцюжку пріоритетних переривань, за винятком того, що в ній відсутнє таке поняття як маска.

Структурна схема приведена на кресленні ІАЛЦ.463626.004 Е2.

**2.2. Опис мікроконтролера 1816ВЕ48**

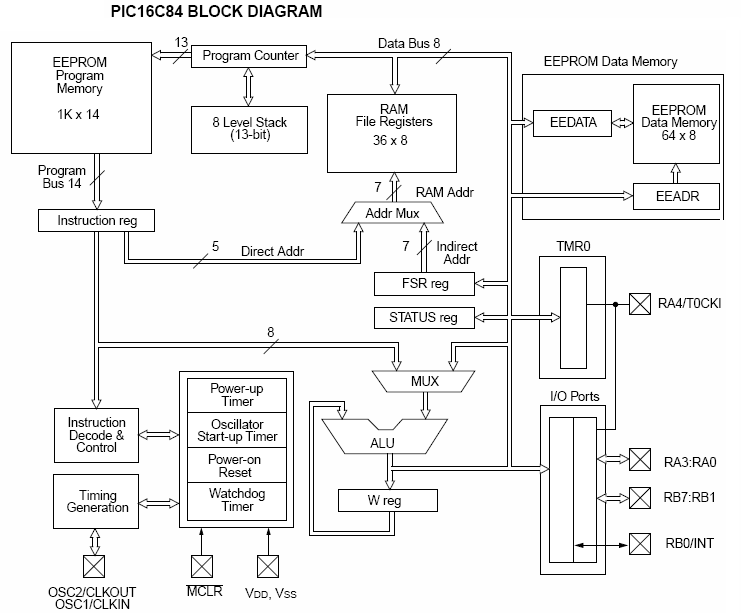
**2.2.1. Загальна характеристика**



*Умовне графічне позначення*

Призначення виводів

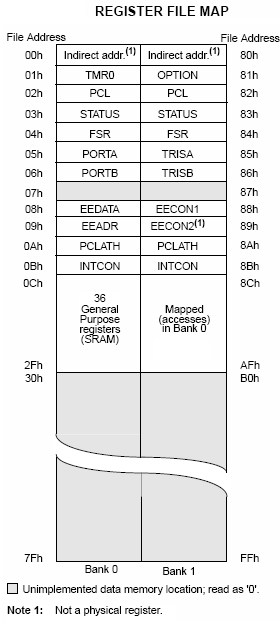
|  |  |  |
| --- | --- | --- |
| Позначення | Нормальний режим | Режим запису EEPROM |
| RA0 - RA3 | Двонаправленні линії ввода-виводу |  |
| RA4 / RTCC | Ножка порта вводу-виводу ичи вхід частоти для таймера/счетчика RTCC |  |
| RB0 / INT | Двонаправленна лінія вводу-виводу чи зовнішній вхід переривання |  |
| RB1 – RB5 | Двонаправленні ліні вводу-виводу |  |
| RB6 | Двонаправлена лінія вводу-виводу | Вхід тактової частоти для EEPROM |
| RB7 | Двонаправлена лінія вводу-виводу | Вхід/вихід EEPROM даних |
| MCLR / Vpp | Сброс контроллера | Сброс контроллера для EEPROM подати Vpp |
| OSC1 / CLKIN | Для підключення зовнішнього джерела синхронізації |  |
| OSC2 / CLKOUT | Для підключення зовнішнього джерела синхронізації чи вихід тактової частоти в режимі RC |  |
| Vdd | Напруга | Напруга |
| Vss | Заземлення | Заземлення |



**2.2.1.1 Огляд регістрів і ОЗУ.**

Область ОЗУ організована як 128 х 8. До осередків ОЗУ можна адресуватися прямо або побічно, через регістр покажчик FSR (04h). Це також відноситься і до EEPROM пам'яті даних-констант.

У регістрі статусу (03h) є біти вибору сторінок, які дозволяють звертатися до чотирьох сторінок майбутніх модифікацій цього кристала. Проте для Pic16c84 пам'ять даних існує тільки до адреси 02fh. Перші 12 адрес використовуються для розміщення регістрів спеціального призначення, допомагають контролювати і управляти контроллером. Регістри з адресами 0ch-2fh можуть бути використані, як регістри загального призначення, які є статичним ОЗУ. Деякі регістри спеціального призначення продубльовані на обох сторінках, а деякі розташовані на сторінці 1 окремо. Коли встановлена сторінка 1, те звернення до адрес 8ch-afh фактично адресує сторінку 0. До регістрів можна адресуватися прямо або побічно. У обох випадках можна адресувати до 512 регістрів.



**2.2.1.2 Адресація**

**Пряма адресація.**

Коли проводиться пряма 9-бітова адресація, молодші 7 біт беруться як пряма адреса з коди операції, а 2 біта покажчика сторінок Rp0, Rp1 з регістра статусу.

·**Непряма адресація.**

Будь-яка команда, яка використовує F0 (адреса 00) як регістра фактично звертається до покажчика, який зберігається в FSR (04h). Читання непрямим чином самого регістра f0 дасть результат 00h. Запис в регістр f0 непрямим чином виглядатиме як Nop, але біти статусу можуть бути змінені. Необхідний 9 - бітова адреса формується об'єднанням вмісту 8 - бітового регістра і біта IRP FSR з регістра статусу.

**2.2.1.3 Таймер/лічильник.**

Режим таймера вибирається шляхом скидання в нуль бита RTS, який знаходиться в регістрі OPTION. У режимі таймера RTCC буде інкрементіроваться від ВНУТРІШНЬОГО ДЖЕРЕЛА частоти кожен командний цикл (без переддільника). Після запису інформації в RTCC, інкрементірованіє його почнеться після двох командних циклів. Таке відбувається зі всіма командами, які проводять запис або чтение-модификацию-запись f1 (напр.MOVF f1, CLRF f1). Уникнути цього можна за допомогою запису в RTCC скоректованого значення. Якщо RTCC потрібно перевірити на рівність нулю без останову рахунку, слід використовувати інструкцію MOVF f1,w.

Режим лічильника вибирається шляхом установки в одиницю бита RTS, який находітсяв регістрі OPTION. У цьому режимі RTCC буде інкрементіроваться або позитивним, або негативним фронтом на ніжці Ra4/rtcc від ЗОВНІШНІХ подій. Напрям фронту визначається таким, що управляє бітом RTE в регістрі OPTION. При Rte=0 буде вибраний позитивний фронт.

Пределітель може бути використаний або спільно з RTCC, або з Watchdog таймером. Варіант підключення дільника контролюється бітом PSA в регістрі OPTION. При Psa=0 дільник буде приєднаний до RTCC. Вміст дільника програмі недоступно. Коефіцієнт ділення - програмується.

Переривання по RTCC виробляється тоді, коли відбувається переповнювання RTCC таймера/лічильника при переході від FFH до 00h. Тоді встановлюється біт запиту RTIF в регістрі Intcon<2>. Дане переривання можна замаскувати бітом RTIE в регістрі Intcon<5>. Біт запиту RTIF повинен бути скинутий програмно при обробці переривання. Переривання по RTCC не може вивести процесор з SLEEP тому, що таймер не функціонує в цьому режимі.

**2.2.1.4 Регістр статусу.**

Регістр (f3) містить арифметичні прапори АЛУ, стан контроллера при скиданні і біти вибору сторінок для пам'яті даних. (f3) доступний для будь-якої команди так само, як будь-який інший регістр. Проте, биті TO і PD встановлюються апаратний і не можуть бути записані в статус програмно. Це слід мати на увазі при виконанні команди з використанням регістра статусу. Наприклад, команда CLRF f3 обнулить всі біти, окрім бітів TO і PD, а потім встановить біт Z=1. Після виконання цієї команди регістр статусу може і не мати нульового значення (із-за бітів TO і PD) f3=000??100. Тому рекомендується для зміни регістра статусу використовувати тільки команди бітової установки BCF, BSF, MOVWF, які не змінюють решту біт статусу.

Розміщення прапорів в регитсре статусу наступне:

b7 b6 b5 b4 b3 b2 b1 b0

IRP Rp1 Rp0 TO PD Z DC C

C - Прапор перенесення/позики. Для команд ADDWF і SUBWF. Цей біт встановлюється, якщо в результаті операції з самого старшого розряду походить перенесення. Віднімання здійснюється шляхом збільшення додаткової коди другого операнда. При виконанні команд зрушення цей біт завжди завантажується з молодшого або старшого біта зрушуваного джерела.

DC - Прапор десяткового перенесення/позики. Для команд ADDWF і SUBWF. Цей біт встановлюється, якщо в результаті операції з четвертого розряду походить перенесення. Механізм установки десяткового біта перенесення "DC" той же самий, відрізняється тим, що відстежується перенесення з четвертого біта

Z - Прапор нульового результату

PD - Power Down ( режим зберігання даних)

TO - Time Out. Прапор спрацьовування Watchdog таймера

Rp1, Rp0 - Біти вибору сторінки пам'яті даних при прямій адресації

00= Сторінка 0 (00h-7fh)

01= Сторінка 1 (80h-ffh)

10= Сторінка 2 (100h-17fh)

11= Сторінка 3 (180h-1ffh)

На кожній сторінці розташоване 128 байт. У кристалі Pic16c84 використовується тільки Rp0. У цьому кристалі Rp1 може використовуватися просто як біт загального призначення читання/запису. Проте треба пам'ятати, що в подальших розробках він використовуватиметься.

IRP - Біт вибору сторінки пам'яті даних при непрямій адресації

0= Сторінки 0,1 (00h-ffh)

1= Сторінка 2,3 (100h-1ffh)

Цей біт в кристалі Pic16c84 використовувати не має сенсу. Тому його можна використовувати як біт загального призначення читання/запису. Проте, треба пам'ятати, що в подальших розробках він використовуватиметься.

**2.2.2 Довготривала Пам'ять даних EEPROM.**

Пам'ять даних EEPROM дозволяє прочитати і записати байт інформації. При записі байта автоматично стирається попереднє значення і записуються нові дані (стирання перед записом). Всі ці операції проводить вбудований автомат запису EEPROM. Вміст елементів цієї пам'яті зберігається при виключенні живлення.

Кристал Pic16c84 має пам'ять даних 64х8 EEPROM битий, яка дозволяє запис і читання під час нормальної роботи (у всьому діапазоні живлячої напруги). Ця пам'ять не належить області регістрів ОЗУ. Доступ до неї здійснюється через два регістри: EEDATA <08h>, який містить в собі восьмибітові дані для читання/запису і EEADR <09h>, який містить в собі адресу осередку до якої йде звернення. Додатково є два керівників регістра: Eecon1 <88h> і Eecon2 <89h> .

При прочитуванні даних з пам'яті EEPROM необхідно записати необхідну адресу в EEADR регістр і потім встановити біт RD Eecon1<0> в одиницю. Дані з'являться в наступному командному циклі в регістрі EEDATA і можуть бути прочитані. Дані в регістрі EEDATA защипуються.

BCF STATUS, Rp0 ; Bank 0

MOVLW Config\_addr ;

MOVWF EEADR ; Address to read

BSF STATUS, Rp0 ; Bank 1

BSF Eecon1, RD ; EE Read

BCF STATUS, Rp0 ; Bank 0

MOVF EEDATA, W ; W = EEDATA

При записі в пам'ять EEPROM, необхідно спочатку записати необхідну адресу в EEADR регістр і дані в EEDATA регістр. Потім виконати спеціальну послідовність команд, проїзводящуюю безпосередній запис:

BSF STATUS, Rp0 ;Bank 1

BCF INTCON, GIE; Disable Ints

BSF Eecon1, WREN; Enable Write

MOVLW 55h

MOVWF Eecon2 ;wr 55

MOVLW Aah

MOVWF Eecon2

BSF Eecon1,wr; встановити WR битий, почати запис

BSF INTCON, GIE; Enable Ints

Під час виконання цієї ділянки програми, всі переривання повинні бути заборонені для точного виконання тимчасової діаграми. Час запису - приблизно 10мс. Фактичний час запису змінюватиметься залежно від напруги, температури і індивідуальних властивостей кристала. В кінці запису битий WR автоматично обнуляється, а прапор завершення запису EEIF, він же запит на переривання встановлюється в 1.

Для запобігання випадковим записам в пам'ять даних передбачений спеціальний біт WREN в регістрі Eecon1. Рекомендується тримати біт WREN вимкненим, окрім тих випадків, коли потрібно відновити пам'ять даних. Більш того, кодові сегменти, які встановлюють біт WREN і ті, які виконують запис повинні зберігатися на різних адресах, щоб уникнути випадкового виконання їх обох при збої програми.

Регістри, що управляють, для EEPROM

EEDATA EEPROM регістр даних 08h

EEADR EEPROM регістр адреси 09h

Eecon1 EEPROM 1 регістр 88h, що управляє

Eecon2 EEPROM 2 регістр 89h, що управляє

**2.2.3. Переривання.**

Переривання в Pic16c84 можуть бути від чотирьох джерел:

- зовнішнє переривання з ніжки Rb0/int

- переривання від переповнювання лічильника/таймера RTCC

- переривання після закінчення запису даних в EEPROM

- переривання від зміни сигналів на ніжках порту Rb<7:4>.

Всі переривання мають один і той же вектор/адресу - 0004h. Проте, в регістрі переривань INTCON, що управляє, записується від якого саме джерела поступив запит переривання. Записується відповідним бітом-прапором. Таке переривання може бути замасковане індивідуально або загальним бітом. Єдиним виключенням є переривання по кінцю запису в EEPROM. Цей прапор знаходиться в іншому регістрі Eecon1.

Біт загального дозволу/заборони переривання GIE (INTCON <7>) вирішує (еслі=1) всі індивідуально незамасковані переривання або забороняє (еслі=0) їх. Кожне переривання окремо може бути додаткове разрешено/запрещено установкою/скиданням відповідного біта в регістрі INTCON.

Битий GIE обнуляється при скиданні. Коли починає оброблятися переривання, битий GIE обнуляється, щоб заборонити подальші переривання, адреса повернення посилається в стек, а в програмний лічильник завантажується адреса 0004h. Час реакції на переривання для зовнішніх подій, таких як переривання від ніжки INT або порту B, складає приблизно п'ять циклів. Це на один цикл менше, ніж для внутрішніх подій, таких як переривання по переповнюванню від таймера RTCC. Час реакції завжди однаковий.

У підпрограмі обробки переривання джерело переривання може бути визначений по відповідному біту в регістрі прапорів. Цей прапор-біт повинен бути програмно скинутий усередині підпрограми. Прапори запитів переривань не залежать від відповідних маскуючих бітів і біта загального маскування GIE. Команда повернення з переривання RETFIE завершує перериваючу підпрограму і встановлює біт GIE, щоб знову вирішити переривання.

Скидання запиту RTIF - справа програми обробки.

Скидання запиту RBIF - справа програми обробки.

Скидання запиту EEIF - справа програми обробки.

Регістр переривань і його біти, що управляє

GIE EEIE RTIE INTE RBIE RTIF INTF RBIF

{\ul\fs24\ulnone\lang1049 Адреса: 0bhЗначення при reset= 0000 000

RBIF -флаг переривання від зміни на порту RB.

Прапор встановлюється, коли сигнал на вході Rb<7:4> змінюється.

Прапор скидається програмним способом.

INTF - Прапор переривання INT.

Прапор встановлюється, коли на ніжці INT з'являється сигнал від

зовнішнього джерела переривання. Скидається програмним способом.

RTIF - Прапор переривання від переповнювання RTCC.

Прапор встановлюється, коли RTCC переповнюється.

Прапор скидається програмним способом.

RBIE - Біт дозволу/заборони RBIF переривання.

RBIE = 0 : забороняє RBIE переривання

RBIE = 1 : вирішує RBIE переривання

INTE | - Біт дозволу/заборони INT переривання.

INTE = 0 : забороняє INT переривання

INTE = 1 : вирішує INT переривання

RTIE Біт дозволу/заборони RTIF переривання.

RTIE = 0 : забороняє RTIE переривання

RTIE = 1 : вирішує RTIE переривання

EEIE - Біт дозволу/заборони переривання EEPROM запису.

EEIE = 0 : забороняє EEIF переривання

EEIE = 1 : вирішує EEIF переривання

GIE Біт дозволу/заборони всіх переривань.

GIE = 0 : забороняє переривання

GIE = 1 : вирішує переривання

Він скидається автоматично при наступних обставинах:

- по включенню живлення.

- по зовнішньому сигналу /mclr при нормальній роботі.

- по зовнішньому сигналу /mclr в режимі SLEEP.

- по закінченню затримки таймера WDT при нормальній роботі.

- по закінченню затримки таймера WDT в режимі SLEEP.

Порт введення-виводу.

Кристал має два порти: 5 битий порт RA і 8 битий порт RB з побітовим індивідуальним настроюванням на введення або на вивід.

Порт А - це порт шириною 5 битий, відповідні ніжки кристала Ra<4:0>. Лінії Ra<3:0> двонаправлені, а лінія Ra4 - вихід з відкритим стоком. Адреса регістра порту А - 05h. Що відноситься до порту А регістр TRISA, що управляє, розташований на першій сторінці регістрів за адресою 85h. Trisa<4:0> - це регістр шириною 5 битий. Якщо битий керівника TRISA регістра має значення одиниця, то відповідна лінія встановлюватиметься на введення. Нуль перемикає лінію на вивід і одночасно виводить на неї вміст відповідного регістра клямки.

Порт В - це двонаправлений порт, шириною у вісім битий (адреса регістра 06h). Що відноситься до порту В регістр TRISB, що управляє, розташований на першій сторінці регістрів за адресою 86h. Якщо битий керівника TRISB регістра має значення одиниця, то відповідна лінія встановлюватиметься на введення. Нуль перемикає лінію на вивід і одночасно виводить на неї вміст відповідного регістра клямки.

У кожної ніжки порту В є невелике активне навантаження (біля 100мка) на лінію живлення. Вона автоматично відключається, якщо ця ніжка запрограмована як вивід. Більш того, керівник битий RBPU Option<7> може відключити (Rbpu=1) всі навантаження. Скидання при включенні живлення також відключає всі навантаження.

Чотири лінії порту В (Rb<7:4>) мають здатність викликати переривання при зміні значення сигналу на будь-якій з них. Якщо ці лінії налаштовані на введення, то вони опитуються і защипуються в циклі читання Q1. Нова величина вхідного сигналу порівнюється із старою в кожному командному циклі. При неспівпаданні значення сигналу на ніжці і в клямці, генерується високий рівень. Виходи детекторів "неспівпадань" Rb4,rb5,rb6,rb7 об'єднуються по АБО і генерують переривання RBIF (що запам'ятовується в Intcon<0>). Будь-яка лінія, настроєна як вивід, не бере участь в цьому порівнянні. Переривання може вивести кристал з режиму SLEEP. У підпрограмі обробки переривання слід скинути запит переривання одним з наступних способів:

1) Заборонити переривання за допомогою обнулення бита RBIE Intcon<3>.

2) Прочитати порт В. Ето завершить стан порівняння.

3) Обнулити біт RBIF Intcon<0>.

Переривання по неспівпаданню і програмно встановлювані внутрішні активні навантаження на цих чотирьох лініях можуть забезпечити простій інтерфейс наприклад з клавіатурою, з виходом з режиму SLEEP по натисненню клавіш.

Ніжка Rb0 суміщена з входом зовнішнього переривання INT.

Деякі команди внутрішньо виконуються як чтеніє+запісь. Наприклад, команди BCF і BSF прочитують порт цілком, модифікують один біт і виводять результат назад. Тут необхідна обережність. Наприклад, команда BSF для біта 5 регістра f6 (порт В) спочатку вважає все вісім битий. Потім виконуються дії над бітом 5 і нове значення байта цілком записується у вихідні клямки. Якщо інший біт регістра f6 використовується як двонаправлене введення/вивід (скажемо біт 0) і в даний момент він визначений як вхідний, вхідний сигнал на цій ніжці буде лічений і записаний назад у вихідну клямку етой-же ніжки, затираючи її попередній стан. До тих пір, поки ця ніжка залишається в режимі введення, жодних проблем не виникає. Проте, якщо пізніше лінія 0 перемкнеться в режим виводу, її стан буде невизначеним.

На ніжку, що працює в режимі виводу, не повинні "навалюватися" зовнішні джерела струмів("монтажне І", "монтажне АБО"). Результуючі великі струми можуть пошкодити кристал.

Послідовне звернення до портів введення/виводу.

Запис в порт виводу відбувається в кінці командного циклу. Але при читанні, дані повинні бути стабільні на початку командного циклу. Будьте уважні в операціях читання, наступних відразу за записом в той же порт. Тут треба враховувати інерційність встановлення напруги на виводах. Може потрібно програмна затримка, щоб напруга на ніжці (залежить від навантаження) встигла стабілізуватися до початку виконання наступної команди читання.

**2.2.4. Система команд.**

Каждая команда PIC16C84 - это 14-битовое слово, которое разделено по смыслу на следующие части: - 1. код операции, -2. поле для одного и более операндов, которые могут участвовать или нет в этой команде. Система команд PIC16C84 включает в себя байт-ориентированные команды, бит-ориентированные, операции с константами и команды передачи управления.

Для байт-ориентированных команд "f" обозначает собой регистр, с которым производится действие; "d" - бит определяет, куда положить результат. Если "d" =0, то результат будет помещен в W регистр, при "d"=1 результат будет помещен в "f", упомянутом в команде.

Для бит-ориентированных команд "b" обозначает номер бита, участвующего в команде, а "f" -это регистр , в котором этот бит расположен.

Для команд передачи управления и операций с константами, "k" обозначает восьми или одиннадцати битную константу.

Все команды выполняются в течение одного командного цикла. В двух случаях исполнение команды занимает два командных цикла: проверка условия и переход, изменение программного счетчика как результат выполнения команды. Один командный цикл состоит из четырех периодов генератора. Таким образом, для генератора с частотой 4 МГц время исполнения командного цикла будет 1 мкс.

Байт-ориентированные команды.

Мнемокод Название команды

ADDWF f,d Сложение W с f

ANDWF f,d Логическое И W и f

CLRF f Сброс регистра f

CLRW - Сброс регистра W

COMF f,d Инверсия регистра f

DECF f,d Декремент регистра f

DECFSZ f,d Декремент f, пропустить команду, если 0

INCF f,d Инкремент регистра f

INCFSZ f,d Инкремент f, пропустить команду, если 0

IORWF f,d Логическое ИЛИ W и f

MOVF f,d Пересылка регистра f

MOVWF f Пересылка W в f

NOP - Холостая команда

RLF f,d Сдвиг f влево через перенос

RRF f,d Сдвиг f вправо через перенос

SUBWF f,d Вычитание W из f

SWAPF f,d Обмен местами тетрад в f

XORWF f,d Исключающее ИЛИ W и f

ADDLW k Сложение константы с W.

ANDLW k Логическое И константы и W

IORLW k Логическое ИЛИ константы и W

SUBLW k Вычитание W из константы.

MOVLW k Пересылка константы в W

XORLW k Исключающее ИЛИ константы и W

OPTION Загрузка W в OPTION регистр

TRIS f Загрузка TRIS регистра

Бит-ориентированные команды.

Мнемокод Название

BCF f,b Сброс бита в регистре f

BSF f,b Установка бита в регистре f

BTFSC f,b Пропустить команду,

если бит равен нулю

BTFSS f,b Пропустить команду,

если бит равен единице

Команды передачи управления.

Мнемокод Название

CALL k Вызов подпрограммы

CLRWDT Сброс Watchdog таймера

GOTO k Переход по адресу

RETLW k Возврат из подпрограммы с

загрузкой константы в W

RETFIE Возврат из прерывания.

RETURN Возврат из подпрограммы.

SLEEP Переход в режим SLEEP

1. **Розробка програмного забуїезпечення**