

Яндекс.Директ

Микроконтроллеры avr

Микроконтроллеры
на AliExpress! Отличный выбор!
Скидки. Быстрая доставка!

Спецпредложения Доставка
Оплата Товары из России

ru.aliexpress.com



Грузовые лифты - подъемники!

Мы завод! Работаем без
посредников! Подъемники от
59 000р! Узнать подробнее

Наша продукция О нас
Наши контакты

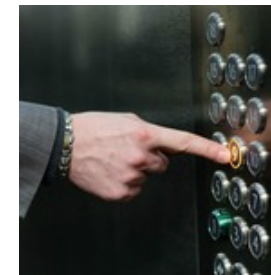
грузовой-лифт.рф
Адрес и телефон



Стартовые наборы Arduino от 1690р

Готовые стартовые наборы
Arduino с инструкцией на
русском языке. 30 уроков!

smartelements.ru
Адрес и телефон



Лифты под ключ в Краснодаре

Поставка, монтаж, сервис
лифтов от производителя.
Гарантия 24 мес! Звоните!

Модели Сертификаты
Оставить заявку Контакты
esd-lift.ru Адрес и телефон

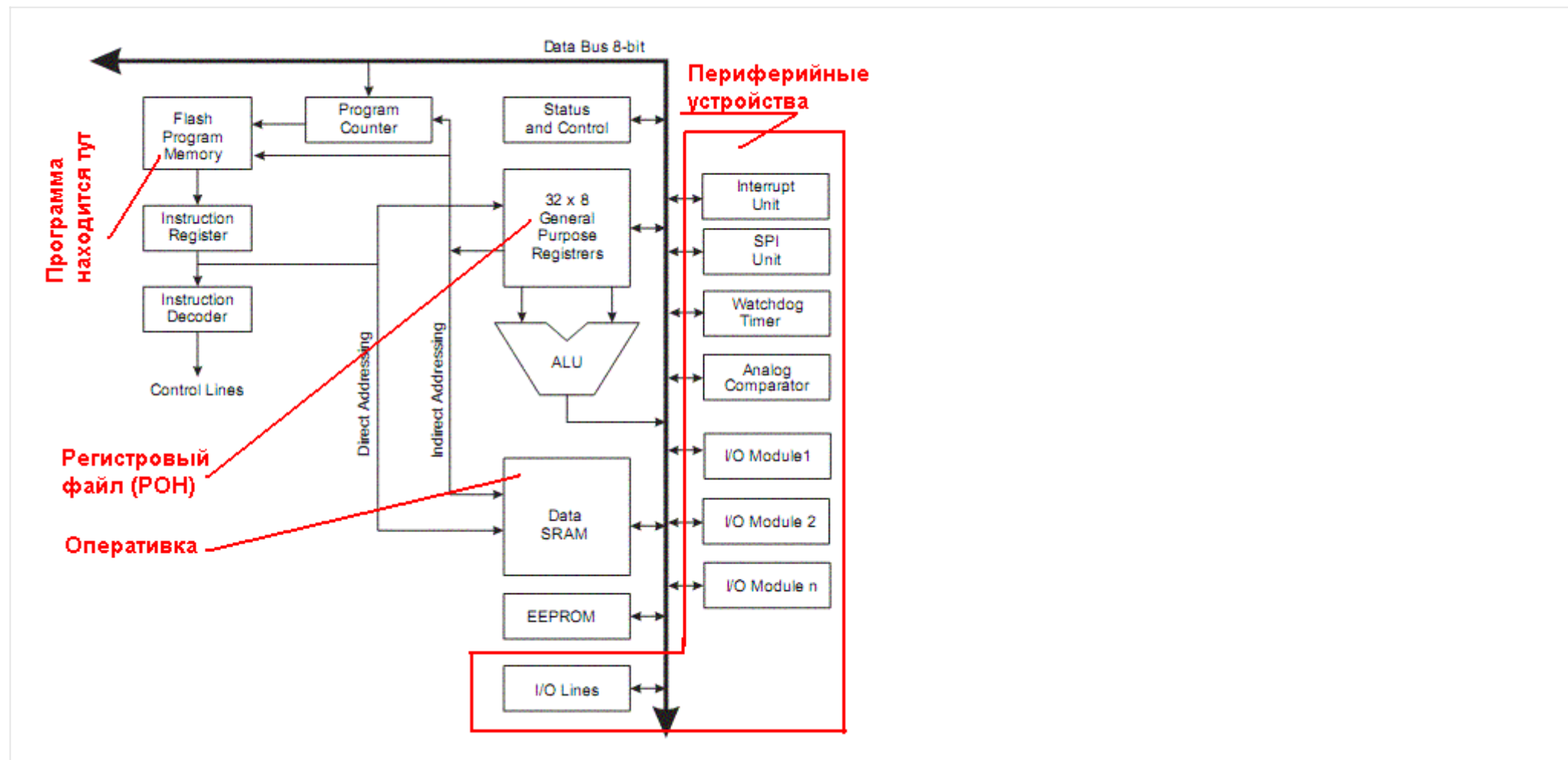
AVR. Учебный курс. Архитектура.

AVR. Учебный курс | 2 Июль 2008 | DI HALT | 98 Comments

Итак, камрад, прежде чем ты начнешь работать с контроллером, то неплохо бы тебе узнать что у него внутри.
Поэтому дам тебе краткий ликбез по **архитектуре контроллера AVR**.

Основой любого микроконтроллера является вычислительное ядро. Во всех моделях AVR оно практически одинаковое и это большой плюс. Именно единство архитектуры обеспечивает легкую переносимость кода.

Итак, что же у нас в основе микроконтроллера, взгляни на диаграмму:



Ядро состоит, в первую очередь, из памяти программ (Flash Programm Memory) и Арифметико-логического устройства (ALU), блока управления (на диаграмме не показан) и программного счетчик (Program Counter). Также есть тактовый генератор, задающий импульсы

относительно которых работают блоки микроконтроллера. Тактовый генератор можно сравнить с маятником и собачкой в будильнике: маятник туда сюда, собачка тикает по одному зубчику — шестеренки крутятся. Встала собачка — встал весь будильник.

При старте микроконтроллера значение программного счетчика равно 0000 — это адрес первой команды в нашей flash ROM. Микроконтроллер хватает оттуда два байта (код команды и ее аргументы) и отдает на выполнение в декодер команд (Instruction Decoder).

А дальнейшая судьба зависит от команды. Если это просто команда работы с какими-либо действиями, то они будут выполнены, а на следующем такте значение программного счетчика будет увеличено и из следующей пары ячеек памяти будут взяты еще два байта команды и также отправлены на выполнение.

Все интересней становится когда встречается команда перехода. В этом случае в Программный счетчик загружается адрес указанный в команде (абсолютный переход) или его значение увеличивается не на 1, а на столько сколько нужно и на следующем такте микроконтроллер возьмет команду уже с нового адреса.

Декодер команд загребаёт команду и скормливает ее логике блока управления, который уже пинает все остальные блоки, заставляя их делать нужные действия в нужном порядке.

Вся математика и обработка делается посредством ALU. Это, своего рода, калькулятор. Он может складывать, вычитать, сравнивать, сдвигать разными способами, иногда делить и умножать (это считается круто, встречается редко).

В качестве промежуточных операндов используются 32 ячейки — Оперативные регистры общего назначения РОН. Доступ к этим ячейкам самый быстрый, а число операций с их содержимым наиболее богатое. В ассемблере регистры эти называются просто R0,R1,R2 ... R31. Причем делятся они на три группы:

Младшие R0..R15

Обычные регистры общего назначения, но какие то ущербные. С ними не работают многие команды, например, такие как загрузка непосредственного числа. Т.е. нельзя, например, взять и присвоить регистру число. Зато можно скопировать число из любого другого регистра.

Старшие R16..R31

Полноценные регистры, работающие со всеми командами без исключения.

Индексные R26...R31

Шесть последних регистров из старшей группы особенные. В принципе, их можно юзать и как обычные регистры общего назначения. Но, кроме этого, они могут образовывать регистровые пары X(R26:R27), Y(R28,R29), Z(R30:R31) которые используются как указатели при работе с памятью.

ОЗУ

Кроме 32 регистров в микроконтроллере есть оперативная память. Правда не везде — в младших семействах AVR Tiny12 и Tiny11 оперативной памяти нет, так что приходится вертеться в 32 ячейках.

Оперативная память это несколько сотен ячеек памяти. От 64 байт до 4килобайт, в зависимости от модели. В этих ячейках могут храниться любые данные, а доступ к ним осуществляется через команды Load и Store.

То есть нельзя взять, например, и прибавить к ячейке в памяти, скажем, единицу. Нам сначала сделать операцию Load из ОЗУ в РОН, потом в регистре прибавить нашу единицу и операцией Store сохранить ее обратно в память. Только так.

EEPROM

Долговременная память. Память которая не пропадает после выключения питания. Если Flash может содержать только код и константы, а писать в нее при выполнении ничего нельзя (Это память Read Only), то в EEPROM можно сколько угодно писать и читать. Но в качестве оперативки ее особо не поюзаешь. Дело в том, что цикл записи в EEPROM длится очень долго — миллисекунды. Чтение тоже не ахти какое быстрое. Да и число циклов перезаписи всего 100 000, что не очень много в масштабах работы оперативной памяти. EEPROM используется для сохранения всяких настроек, предустановок, собранных данных и прочего барахла, что может потребоваться после включения питания и в основном на чтение. Эта память есть не во всех моделях AVR, но в подавляющем их большинстве.

Периферия

Периферия это внутренний фарш микроконтроллера. То что делает его таким универсальным. ALU, RAM, FLASH и Блок управления это как в компе Мать, Проц, Память, Винт — то без чего комп даже не запустится толком. То периферия это уже как сетевуха, видяха, звуковая

карта и прочие прикормы. Они могут быть разными, разной степени крутости и навороченности, а также комбинироваться в разном порядке.

Именно по наличию на кристалле той или иной периферии происходит выбор микроконтроллера под задачу.

Периферии всякой придумано великое множество, всего я наверное даже не опишу. Но дам основной набор присутствующий почти во всех AVR, а также в других современных контроллерах.

- Порты ввода вывода — то без чего невозможно взаимодействие контроллера с внешним миром. Именно порты обеспечивают то самое «ножкодрыгательство» управляющее другими элементами схемы. Захотели получить на выводе единичку, дали приказ соответствующему порту — получите, распишитесь. Захотели узнать какой там сигнал на входе? Спросили у соответствующего порта — получили. Почти все выводы микроконтроллера могут работать в режиме портов ввода-вывода.
- UART/USART приемопередатчик — последовательный порт. Работает по тому же асинхронному протоколу что и древние диаловые модемы. Старый как мир, надежный и простой как кувалда. Подходит для связи с компьютером и другими контроллерами.
- Таймеры/счетчики — задача таймеров отсчитывать тики. Сказал ему отсчитать 100 тактов процессора — он приступит и как досчитает подаст сигнал. Им же можно подсчитывать длительность входных сигналов, подсчитывать число входных импульсов. Да много чего умеет таймер, особенно в AVR. Подробное описание функций таймера занимает добрых три десятка страниц в даташите. При том, что таймеров самих существует несколько видов и фарш у них разный.
- АЦП — аналоговый вход. Есть не у всех микроконтроллеров, но вещь полезная. Позволяет взять и измерить аналоговый сигнал. АЦП это своеобразный вольтметр.
- I2C(TWI) интерфейс — последовательная шина IIC. Через нее осуществляется связь с другими устройствами. На IIC можно организовать своеобразную локальную сеть из микроконтроллеров в пределах одного устройства.
- SPI — еще один последовательный протокол, похожа на IIC, но не позволяет организовывать сети. Работает только в режиме Мастер-Ведомый. Зато ОЧЕНЬ быстрая.
- Аналоговый Компаратор — еще один аналоговый интерфейс. Но, в отличие от АЦП, он не измеряет, а сравнивает два аналоговых сигнала, выдавая результат $A > B$ или $A < B$ в двоичном виде.
- JTAG/DebugWire — средство отладки, позволяет заглянуть в мозги контроллера с помощью специального адаптера, например такого, какой встроен в мою демоплату [Pinboard](#). Иной раз без него как без рук.

- PWM — ШИМ генератор. Вообще это не отдельный блок, а дополнительная функция таймера, но тоже полезная. С помощью ШИМ генератора легко задать аналоговый сигнал. Например, менять яркость свечения светодиода или скорость вращения двигателя. Да мало ли куда его применить можно. Число каналов ШИМ разное от контроллера к контроллеру.

Еще бывают встроенные USB, Ethernet интерфейсы, часы реального времени, контроллеры ЖКИ дисплеев. Да чего там только нет, моделей микроконтроллеров столько, что задолбаешься только перечислять.

Взаимодействие ядра с периферией

Ядро одно на всех, периферия разная. Общение между ними происходит через память. Т.е. у периферии есть свои ячейки памяти — регистры периферии. У каждого периферийного устройства их не по одной штуки. В этих регистрах находятся биты конфигурации. В зависимости от того как эти биты выставлены в таком режиме и работает периферийное устройство. В эти же регистры нужно записывать данные которые мы хотим выдать, например, по последовательному порту, или считывать данные которые обработал АЦП. Для работы с периферией есть специальные команды IN и OUT для чтения из периферии в регистр РОН и записи из регистра РОН в периферию соответственно.

Поскольку ядро одинаковое, а периферия разная, то при переносе кода на другую модель микроконтроллера надо только подправить эти обращения так как название периферийных регистров от модели к модели может чуток отличаться. Например, если в контроллере один приемопередатчик UART то регистр прием данных зовется UDR, а если два, то у нас есть уже UDR0 и UDR1. Но, в целом, все прозрачно и логично. И, как правило, портирование кода с одного МК на другой, даже если он написан на ассемблере, не составляет большого труда. Особенно если он правильно написан.

Как узнать что есть в конкретном микроконтроллере?

Для этого на каждый МК есть даташит — техническая документация. И вот там, прямо на первой странице, написано что почему и как. Вот тебе пример, даташит на Мегу16 с моим закадровым переводом :) Жирным шрифтом помечены опции которые я гляжу в первую очередь, как наиболее интересные для меня, остальное, как правило, присутствует по дефолту.

Features (фичи!)

- High-performance, Low-power AVR® 8-bit Microcontroller
(понтная экономичная архитектура AVR)

- Advanced RISC Architecture

(просто офигенная вещь для рискованных чуваков!)

- 131 Powerful Instructions – Most Single-clock Cycle Execution

(131 крутая и быстрая команда!)

- 32 x 8 General Purpose Working Registers

(32 восьми разрядных регистра — те самые R0...R31)

- Fully Static Operation

(Полностью статические операции, т.е. тактовая частота может быть хоть 1 импульс в год)

- Up to 16 MIPS Throughput at 16 MHz

(скорость выполнения до 16 миллионов операций в секунду!)

- On-chip 2-cycle Multiplier

(а числа умеем множить за два такта! Это правда круто, народ!)

- High Endurance Non-volatile Memory segments

- **16K Bytes of In-System Self-programmable Flash program memory**

(памяти хватит набьдлокодить на 16кб кода)

- **512 Bytes EEPROM 8-bit**

(и нажрать на века 512 байт мусора в EEPROM)

- **1K Byte Internal SRAM**

(оперативки 1кб, кому там 2Гигабайт не хватает? Программировать не умеете! =) Тут и 64 байтов за глаза хватает. Помните Билла Гейтса и его «640кб хватит всем!» он знал о чем говорил :)

– Write/Erase Cycles: 10,000 Flash/100,000 EEPROM Microcontroller

(перешишивать флеш можно 10тыщ раз, еепром 100тыщ раз. Так что можешь не бояться экспериментировать)

– Data retention: 20 years at 85°C/100 years at 25°C(1)

(Если законсервируешь свой будильник на AVR, то твоих правнуков он еще и через 100 лет порадует)

– Optional Boot Code Section with Independent Lock Bits

In-System Programming by On-chip Boot Program

True Read-While-Write Operation

(поддержка бутлоадеров. Удобная вещь, позволяет прошиваться без программаторов)

– Programming Lock for Software Security In-System

(если жадный и умный, то можешь закрыть от посторонних прошивку и фиг кто выкрадет твои секреты)

• JTAG (IEEE std. 1149.1 Compliant) Interface

– Boundary-scan Capabilities According to the JTAG Standard Programmable

– Extensive On-chip Debug Support

– Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface Flash

(Отладочный интерфейс JTAG и его фишки)

• Peripheral Features

(А вот, собственно и периферия пошла)

– Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes

(два таймера 8ми разрядных, с кучей всяких режимов разных.

– One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode ATmega16

(один 16ти разрядный таймер счетчик, с кучей всяких примочек и фишек)

– Real Time Counter with Separate Oscillator

(таймер может тикать от отдельного генератора, удобно если хочешь сделать часы)

– **Four PWM Channels ATmega16L**

(Четыре ШИМ канала — на тех же таймерах)

– **8-channel, 10-bit ADC**

(восьмиканальный 10ти разрядный АЦП. Фичи его ниже)

8 Single-ended Channels

(можно замерять по очереди сразу 8 разных напряжений)

7 Differential Channels in TQFP Package Only

(7 дифференциальных каналов. Правда только в корпусе TQFP т.к. ног у него больше)

2 Differential Channels with Programmable Gain at 1x, 10x, or 200x

(два дифференциальных канала с программируемым усилением)

– **Byte-oriented Two-wire Serial Interface**

(Поддержка IIC с аппаратным кодированием байтов)

– Programmable Serial USART

(Последовательный интерфейс. Удобен для связи с компом)

– Master/Slave SPI Serial Interface

(SPI интерфейс, пригодится)

– Programmable Watchdog Timer with Separate On-chip Oscillator

(Спец таймер защиты от зависаний)

– On-chip Analog Comparator

(Тот самый компаратор)

• Special Microcontroller Features

(полезные свистоперделки)

– Power-on Reset and Programmable Brown-out Detection

(защита от косяков в работе при пониженном напряжении ака севшие батарейки)

– **Internal Calibrated RC Oscillator**

(А еще можно сэкономить 20рублей на покупке внешнего кварца. Он нафиг не нужен! :) И это круто!)

– External and Internal Interrupt Sources

(Есть внешние прерывания. Очень удобная вещь)

– Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby

(Дофига режимов энергосбережения)

• I/O and Packages

– **32 Programmable I/O Lines**

– **40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF**

(число полезных ножек, тем самых вводов выводов)

• Operating Voltages designs.

– **2.7 — 5.5V for ATmega16L**

– **4.5 — 5.5V for ATmega16**

(Питающие напряжения. Помните я говорил про низковольтные серии — вот они, во всей красе)

- Speed Grades

- 0 — 8 MHz for ATmega16L

- 0 — 16 MHz for ATmega16

(А это максимальные частоты для разных серий. Низковольтные лажают. Впрочем, они подвержены разгону)

- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L

- Active: 1.1 mA

- Idle Mode: 0.35 mA

- Power-down Mode: < 1 μ A (Потребляемая мощность в разных режимах. 1мА даже в активном режиме это фигня. В 10 раз меньше самого тухлого светодиода)

◀ AVR ◀ Микроконтроллер

98 thoughts on “AVR. Учебный курс. Архитектура.”

Beams

16 Июль 2008 в 20:18

Вопрос возник по этому абзацу:

Блок регистров (РОН), он же регистровый файл.

Если у нас 32 Регистра по 8 бит, то как регистровая пара образует 32 разряда?

Мои представления подсказывают что получается 16 разрядов, но я могу быть не прав.

Прошу внести ясность :)

★ DI HALT

16 Июль 2008 в 21:30

Точно бля, косяк. В запаре был. Щас поправлю. да адресация там 16ти разрядная.

Beams

16 Июль 2008 в 20:35

И в догонку...

В первом абзаце:

EEPROM это энергонезависимая память, что то вроде flash памяти, только допускает куда больше циклов перезаписи.

Дальше по тексту

EEPROM или энергонезависимая память...Используется, например, для складирования каких нибудь логов или другой какой собранной информации. Можно использовать как оперативку, но запись в нее осуществляется довольно медленно, а главное число циклов перезаписи порядка 100 000 раз, что не слишком много, если пользоваться ей десятки и сотни раз в секунду.

Хотя, с оперативной памятью, исходя из первой цитаты, дела обстоят ещё более плачевно, но её не смотря на это используют как оперативную память.

Думаю уместно привести сводную таблицу по циклам перезаписи скорости и объёму, но я могу ошибаться.

★ DI HALT

16 Июль 2008 в 21:32

Не вижу проблемы. В первом случае идет речь о EEPROM и во втором о EEPROM где противоречие?

Beams

16 Июль 2008 в 22:22

Описано что EEPROM допускает больше циклов перезаписи чем flash (ROM), при то что циклов перезаписи EEPROM (100 000) не позволяет использовать её в качестве ROM.

Beams

16 Июль 2008 в 22:28

Тоесть следуя аналогии можно предположить что Flash имеет менне 100 000 циклов перезаписи.

PS можно добавить тыкалку чтобы редактировать сообщение, а то я с первого раза не могу сформулировать, точнее могу но потом оказывается что сформулировал всё правильно, но не то что хотел сказать.

Beams

16 Июль 2008 в 22:30

А....вот ещё случайно нажал не туда :)

Тоесть следуя логике написанного можно предположить что Flash имеет менне 100 000 циклов перезаписи (которых ничтожно мало для ROM).

Вопрос зачем использовать Flash для ROM ?

★ DI HALT

16 Июль 2008 в 22:58

Flash разная бывает. Та что в контроллерх в качестве ROM перешивается всего несколько сот раз. Та что на флешках бытовых примерно 1000 циклов перезаписи.

Ты часом не путаешь RAM и ROM? RAM это обычно оперативка, память с произвольным доступом как на чтение так и на запись. ROM это память только на чтение. Точнее записать то ее можно (тот же флеш) но это требует другой энергетики и в обычно режиме контроллером не используется (точнее используется, но крайне редко, в случаях Boot loader загрузки)

Beams

16 Июль 2008 в 23:03

Конечно я про RAM...

В порыве страсти перепутал букву. Пересмотри пожалуйста всё с учётом того что я писал про RAM.

★ DI HALT

16 Июль 2008 в 23:09

Ну смотри. Конечно флеш в качестве оперативки юзать крайне неразумно. Сшаркивается быстро (хотя может щас придумали какую нибудь более износостойкую память), но если надо хранить какие нибудь переменные ,которые в принципе не нуждаются в сохранении, но которые меняются редко. Скажем раз в день. То их можно и в ЕЕПРОМ сунуть без особого вреда. Но эт на случай если основная ОЗУ забита уже донельзя. У меня на практике такая лажа была только пару раз.

Beams

16 Июль 2008 в 23:16

ок понял, я не прав, прав не я :)

Вот она коллизия в чистом виде (пить надо меньше, спать надо больше) интерпретатор поломался.

Я из опуса уяснил что flash используется как RAM, виноват.

Beams

16 Июль 2008 в 20:50

PS. О том как я сюда...

На сайте <http://www.dihalt.ru> есть ссылка , как я на <http://www.dihalt.ru> ...

Итог ответа Яндекса на Di Halt, почему такой запрос ...

Потому что работать на работе не интересно :)

Ну и ... Токс рекомендовал почитать на досуге, досуг появился.

Да и ... мелькать стало часто имя на страницах не без известного «бульварного чтива» (автор суждения строкою выше) имя, что ещё больше стимулировало интерес.

Почему интересуюсь...Автоматизатор всё таки, хоть и в процессе обучения...ещё один вечный студент. (это не уже не первое образование)

★ DI HALT

16 Июль 2008 в 21:51

Автоматчег? Гы, коллега! Я тоже автоматику заканчивал.

Beams

16 Июль 2008 в 22:35

Я на 4 курсе Курганского Государственного Университета :)

Только вот обманули меня немного.

Автоматизация технологических процессов и производств нефтяной и газовой промышленности, я как дурак ждал когда нам начнут про микроконтроллеры, а нам всё задвижки да клапаны...

Решил твёрдо освоить микроконтроллеры самостоятельно, и вот с этим желанием совпало рождение такого прекрасного проэкта.

PS

Только бы не загнулась всё на середине как у Калашникова...

★ **DI HALT**

16 Июль 2008 в 22:53

У нас контроллеры тоже слабо. Все больше на аналоговую автоматику упирали. И только на 5м курсе пошли PLC контроллеры и микропроцессорные системы.

Beams

16 Июль 2008 в 22:31

Дорогой Артемий поправьте пожалуйста строку

«Оповещать о новых комментариях по почте»

помоему она не согласована :)

★ **DI HALT**

16 Июль 2008 в 22:53

Да тут много ещё косяков. Так что править и править. Со временем поправим.

Beams

16 Июль 2008 в 22:42

А ещё я честно не понял зачем ссылка администрирование, да ещё и весь движок на ладони с блогом разработчиков (если пройти по ссылке)?...

Или я после регистрации стал богом :) или это не для меня регистрация была...

★ **DI HALT**

16 Июль 2008 в 22:52

Це не движок, а твоя персональная админка. Да это стандартная вордпрессовская шняга. Ты просто юзером зарегился. Можешь свой профайл ковырять, пероснальное отображение сайта себе настроить.

Beams

16 Июль 2008 в 22:55

Хорошо а мне не дают по ссылочкам тыкать говорят я не администратор :)

Beams

16 Июль 2008 в 23:05

Только не бейте за флуд :)

★ **DI HALT**

16 Июль 2008 в 23:09

Да какой нахрен флуд. Тут ещё никого нету, вот будут юзеров побольше, тогда да, буду за порядком следить. А щас это дружеская беседа.

Beams

16 Июль 2008 в 23:29

Безумно рад такому прекрасному проекту, что получится...

Поживём — увидим, доживём — узнаем, а если переживём — то обязательно напьемся!!!

С удовольствием окажу посильную помощь, если таковой не требуется буду доблестным флудерастом :)

NTRNO

21 Июль 2008 в 9:54

Уважаемый авторб подскажите в чем проблема компиляции при такой конструкции...

Только начал изучать...

Ниже кусок учебного примера который должен обнулять регистры с R0 по R29

```
.device AT90S1200
.nolist
.include «C:\Program Files\Atmel\AVR Tools\AvrAssembler\Appnotes\1200def.inc»
.list
.org 0x0030
;=====

.def temp = r16

Start:
clr ZL ;
clr ZH ;
ClearLoop:
st ZH,Z ; На эту строку РУГАЕТСЯ!!!!
inc ZL ;
```

```
сpi ZL,30 ;  
brne ClearLoop ;  
rjmp Start ;
```

★ DI HALT

21 Июль 2008 в 11:01

Ну так, а что ты хотел? Пытаешься адресовать ячейку с 16ти разрядной адресацией регистром ZH, который восьмиразрядный. Да и засунуть в эту восьмизрядную ячейку пытаешься шестнадцатизрядную регистровую пару. Вот у тебя компилер и офигевает от твоей наглости.

Лезем в даташит на AT90S1200 и глядим:

ST Z, Rr ; Store Register Indirect (Z) ← Rr

Поменяй операнды местами и будет тебе счастье.

ST Z,ZH ; Должно сработать.

Внимательней!

Кстати, если потом, когданибудь, надо будет тебе обнулить большой массив, не вздумай юзать этот пример. Гляди, у тебя в качестве нуля юзается ZH, а он нулем будет только первые 256 байт ,потом инкремент индексного регистра дойдет и до него.

NTRNO

21 Июль 2008 в 11:12

Классно, когда есть кому помочь!! Пример взят букву в букву из книги :)

Про то что, этот пример имеет недостаток в книге указано, но то что в книге будет не на своих местах стоять ZH и Z я не подозревал,

отсюда и был вопрос что я делаю не так :)

Спасибо!

★ **DI HALT**

21 Июль 2008 в 11:15

А не стоит доверять безоговорочно книгам :)))) Там ляпы бывают и похлеще. Авторы же пишут примеры из головы и опечатки тут не редкость.

FAndrey

10 Август 2008 в 1:42

Я бы сказал резче недоверяй ничему что не проверил :)

А в книгах как правильно заметили куча ляпов и от авторов и от редакторов и от наборщиков. а общем примеры надо применять крайне обдуманно.

Hryam

8 Июнь 2009 в 20:22

Я так понимаю это книжка Джона Мортонa AVR вводый курс. Там вообще много ошибок. я тоже с неё начал изучать МК. Скачал сначала перевод там уйма ошибок потом скачал оригинал, там тоже есть ошибки но меньше...

DGrees

29 Январь 2011 в 18:45

По этой книге и начал изучать МК, то есть изучаю. Книга очень хорошая в плане того, что все понятно и что при написании программ надо много думать самому, усваивается хорошо.

Ну а пока ищешь ошибку в книге или у себя, начинаешь разбираться во всем еще лучше))))

Ктонибудь знает аналогичный курс, но на Си? Или по какому пути изучать Си после книги Джона Мортонна?

★ **DI HALT**

29 Январь 2011 в 20:08

Только K&R а уж к МК привяжешься сам.

DGrees

6 Февраль 2011 в 16:41

Ого. Это получается что после книги Мортонна становишься настолько крутым, что язык можно изучать по его описанию? :) :) :)

Хотелось просто пособий по написанию программ от простого к сложному, примерно как у Мортонна, но на Си.

Хотя так подумал, архитектура МК уже понятна, как что работает понятно, наверное надо просто изучать разные примеры устройств с программами на Си.

★ **DI HALT**

6 Февраль 2011 в 17:28

Так так оно и есть. Си хоть на контроллере хоть на суперкомпьютере остается одним и тем же языком. Тут лишь одна заморочка — обращение к прогмем и нет stdio функций (точнее они есть, но очень уж прожорливые до ресурсов).

А в периферию данные толкать что на ассемблере что на Си — разницы никакой.

Примеры работы с Си есть у меня в рубрике AVR. Учебный курс (статей уже под десяток наверное) Самое сложное там разобрано.

DGrees

7 Февраль 2011 в 11:06

Отлично! Буду тогда изучать. Первые шаги наверное все же по учебному курсу, а дальше уже и в ту книгу можно залезть.

Dghost

5 Сентябрь 2008 в 12:13

Хорший проект, автору респект! Нашел его случайно по нику автора после прочтения известного журнала, который выписываю уже несколько лет.

Здесь все описано доступным языком! Не все конечно понимаю, но пока впитываю знания как губка в разных областях(электроника, коддинг, архитектура мк и тд.) и уже многие термины на слуху и общая картина проясняется!

Вот собираюсь завтра идти в радио магазин за всем всем всем :D

В прайсе нашел почти все нужное, кроме микросхемы l293d, но там есть

L297 и L298n (105 руб.) . Подойдут ли они для простых роботов и экспериментов?

А из мк есть такие: ATmega8L-8PU (58 руб.), ATmega88-20PU(139 руб.), ATmega16-16PU(88 руб.), ATmega8515-16PU(85 руб.), ATtiny2313-20PU(46 руб.),

ATtiny12L-4PU (39 руб.). Я думаю взять для начала :)) 2 ATmega8L-8PU и 2 ATtiny2313-20PU. Что скажите?

★ DI HALT

5 Сентябрь 2008 в 13:15

L298 это то же самое что и L293 НО раза в 4 мощней :) Так что она любому лучше. НО! К ней еще нужны и диоды. Так что купи 8 диодов Шоттки на ток в 1А и впаяй их как показано у меня на схеме.

По поводу МК... возьми лучше АТМега168 она пофаршированней, а по корпусу полностью совместима с АТМега8. Я роботу скоро апгрейд сделаю :) Заменяв процессор, не меняя платы. Если денех не жалко, все же вдвое дороже она вроде бы.

А тини2313 тоже возьми, клевая вещь.

Dghost

5 Сентябрь 2008 в 14:14

Как оперативно! Спасибо за ответ, с l298n понятно(кстати что означает буква n?)

А мк такого как я понял в прайсе нет((((в моем списке его же нет? :))

Из того что я написал какие лучше взять?

★ DI HALT

5 Сентябрь 2008 в 15:38

Тада лучше 88 возьми. в нем фарша больше.

N.. эээ точно не помню, по моему тип корпуса.

Кстати, если захочешь порулить шаговиком, то возьми L297 еще до кучи.

Dghost

5 Сентябрь 2008 в 16:42

Спасибо за советы, еще вопрос — где брать моторчики, щд?

как они маркируются?

а то в прайсе искал — не нашел ничего подобного....

может машинку китайскую купить? если да то какую (дешевую)?)))

может кто знает конкретное название?

★ DI HALT

5 Сентябрь 2008 в 19:50

Маркировку не помню. Отличить его просто — из него выходит четыре или шесть проводов. Это главный признак шаговика. Навыдирать можно из флопов 5дюймовых. В 3.5 они тоже есть, но там шаговик обычно одношарнирный и если его выдрать из родного шасси, то работать не будет, т.к. вал будет болтаться.

★ DI HALT

5 Сентябрь 2008 в 19:50

В китайских машинках стоят говенные моторы. Лучше в магнитофонах.

Dghost

5 Сентябрь 2008 в 23:33

Кстати вопрос какую взять мс из предложенных?

_MAX232CWE-T — 49.00

_MAX232D PBF — 26.00

_MAX232CPE — 29.50

★ DI HALT

5 Сентябрь 2008 в 23:36

они отличаются только корпусом. Бери ту которую тебе удобней будет паять.

Скорей всего тебе будет нужна D как я полагаю, это DIP корпус.

Granit

25 Октябрь 2008 в 17:31

DI HALT — и я автоматчик))) Beams — я тоже учусь в Курганском Государственном Университете на Автоматизации Технологических Процессов в производстве. Правда я на втором курсе. Также интересуюсь микроконтроллерами, но применительно в робототехнике=)

Приятно узнать что такой хороший сайт сделал автоматчик;) Спасибо, всё очень ясно расписано=)

outsider

3 Декабрь 2008 в 4:48

Совсем ничего нет про fuse-биты. Где лежат, как управляются, на что влияют. Как раз сейчас сижу пытаюсь при помощи avrdude выставить сменить у атмеги8 частоту со штатной 1МГц хотя бы на 4, а лучше на 8. В даташите либо плохо ищу, либо информация неполная 8(

★ **DI HALT**

3 Декабрь 2008 в 4:51

В даташите есть все. Про фуз биты я уже писал.

outsider

3 Декабрь 2008 в 4:55

Тогда извиняюсь. Пошел снова рыскать.

Ridik911

15 Апрель 2009 в 0:57

смотри

например атмега8

открываем

кликаем по странице с названием «System Clock and Clock Options»

далее «Clock Sources» и изучаем (совместно с Сократом:))

вот впринципе и все.. аналогично и прочие AVR

RMiSe

15 Апрель 2009 в 23:14

Здравствуй DI HALT!

Не подскажите чем отличается ATMmega8535L-8AI от ATМега8535-16PU, кроме максимальной частоты?

Заранее благодарен за ответ.

★ **DI HALT**

16 Апрель 2009 в 0:53

A — корпус SOIC (поверхностный монтаж под паяльник на ура)

P — корпус PDIP (монтаж в дырки)

M (не уверен) — MLF корпус (поверхностный монтаж — без фена не соваться)

U — коммерческий температурный диапазон (0...50c)

I — промышленный температурный диапазон (-30...80C)

Если паяльник в руки взял вот недавно, то бери PDIP его пять НАААМНОГО проще.

RMiSe

16 Апрель 2009 в 0:59

А в плане программирования?

★ **DI HALT**

16 Апрель 2009 в 1:02

Никакой.

Кстати, Мега8535 изнутри практически ничем не отличается от Мега8 на которой у меня тут все примеры заточены. Там разница только в том, что у мега8 ног поменьше, да таблица векторов прерываний чуток отличается. А так один в один.

RMiSe

16 Апрель 2009 в 1:12

А вы интернет магазином пользовались (покупка МК или других радиодеталей)?

★ **DI HALT**

16 Апрель 2009 в 1:44

Пользовался и не раз. Доставка только кусачая очень.

RMiSe

16 Апрель 2009 в 6:42

А каким? Просто их куча, и к каждому доверие нет.

★ **DI HALT**

16 Апрель 2009 в 13:33

Платан

ЭФО

Терразлектроника

это в каких покупал.

johnspace

30 Октябрь 2009 в 15:02

Отличный сайт, пожалуй, лучше и не встречал))) Может подскажите как новичку возможно ли снятие дорожек с микросхем и каким образом???

С Ув. JohNSpace

Osergey0

11 Ноябрь 2009 в 2:35

Комрад, вопрос ты плантых консультаций по построению схем на AVR не оказываешь? Хорошему человеку по диплому нужно со знающим человеком вопрос решить...

Может кого-нибудь рекомендуешь?

Maximka

19 Февраль 2010 в 11:50

Добрый день! Вливаюсь в вашу дружную компанию. Читаю «в запой» две недели, но вот и появились вопросы. Как я понял в памяти МК с 0 адреса идут PОН, за ними PВВ, и потом такие же регистры для настройки остальных перифирийных устройств? Таких как таймеры, счетчики, компараторы...?

Maximka

19 Февраль 2010 в 11:53

и в догонку, а регистр SREG? Это какой то общий регистр? Не относящийся ни к перифирии ни к чему? И какие вообще бывают регистры?

★ DI HALT

19 Февраль 2010 в 11:55

SREG это регистр флагов. В нем стоят хитрые биты по которым можно определить результат операции предыдущей команды. Например вычитаем и появился нулевой результат — возникает флаг Z и так далее. Там много комбинаций. И все команды переходов условных делаются на основе этого регистра.

★ DI HALT

19 Февраль 2010 в 11:57

PBB это и есть регистры периферийных устройств.

Т.е. фактически вся память делится на

РОН

регистры периферии и ОЗУ.

Maximka

19 Февраль 2010 в 13:40

Ё-маё!! Понял! :) Дело в том, что у Евстифеева регистры PBB, во вступлении в тему, расписаны как регистры для физических Входов Выходов (Портов) это и внесло непонятку.

Та-а-а-ак, а все таки про регистр SREG, это наверно регистр АЛУ? (в него заносятся значения математических результатов.)

★ DI HALT

19 Февраль 2010 в 14:12

Это регистр состояния процессора. Почитай внимательней. Результат работы АЛУ остается в первом аргументе команды.

А sreg это регистр флагов. Прочитай описание его битов и поймешь что это и зачем.

200_OK

21 Март 2010 в 20:27

Di Halt, поправь опечатки плиз — в тексте несколько раз встречается регистр R32.

Как я понимаю, всего есть 32 РОН (R0-R31). (Или я что-то не понимаю?)

P.S.

Сайт — супер!

★ **DI HALT**

21 Март 2010 в 20:33

Спасиб, поправил.

zloypacifist

24 Сентябрь 2010 в 11:59

ATMEGA16-16PU тоже самое что и ATMEGA16A-16PU?

Живу в Екатеринбурге, искал AVR в магазинах, нашел то что рекомендовли, но чет дорого очень 200-300 рублей! ATMEGA16-16PU(227p), а в коментах смотрю дык там AVR некоторые и за 100 не выходят, может кто посоветует где в Ё-бурге дешевле купить можно?

★ **DI HALT**

24 Сентябрь 2010 в 13:38

Почити. А-версия имеет пониженно энергопотребление. В остальном разницы нет.

AVR сейчас сильно подорожали, связано это с тем, что сейчас кризис производства — Atmel продал свои заводы и вынес производство в китай, а те не успевают штамповать контроллеры. Поэтому возник резкий дефицит контроллеров AVR по всему миру (некоторых позиций днем с огнем не достать). Будет это еще несколько месяцев так, потом подешевеет вновь.

LamaK

26 Октябрь 2010 в 0:36

могу посоветовать магаз <http://www.ekits.ru> довольно дешевые детали ,особенно если брать в больших количествах, + доставка пару недель

<http://www.ekits.ru/shop/index.php?categoryID=622> — собственно МКшки AVR

★ **DI HALT**

26 Октябрь 2010 в 0:42

Они охренели! Я беру все в 2-3 раза дешевле. =)))

LamaK

26 Октябрь 2010 в 22:37

а где?) думаю, никому не помешало бы знать)

★ **DI HALT**

26 Октябрь 2010 в 22:49

elitan.ru но там сильно зависит от количества и текущего прихода. Сейчас долго и дорого, но можно поймать момент (как я) и купить мегу16, например, за 60р.

LamaK

26 Октябрь 2010 в 0:34

«— Power-down Mode: < 1 μ A

(Потребляемая мощность в разных режимах. 1mA даже в активном режиме это фигня. В 10 раз меньше самого тухлого светодиода)»

буквой «мю» обозначаются микроамперы вроде? а миллиамперы — просто «m»...

★ **DI HALT**

26 Октябрь 2010 в 0:40

Да мю это микро. Но речь то идет не о Power Down режиме, а о активном.

LamaK

26 Октябрь 2010 в 22:37

мда, чет не не внимательно читал... Наверно, ночами надо спать все-таки)

ТЕНЬ63

28 Октябрь 2010 в 13:53

У меня в городе ~~щ~~ас можно достать вот такие

AT mega 48 L — 20 PU 145,0

AT mega 8 L — 8 PU 95,0

AT mega 8 — 16 PU 160,0

AT mega 16 — 16 PU 175,0

AT tiny 2313 — 20 PU 87,0

AT tiny 2313 — 20 SU 77,0

AT 27 C 020 — 55 PU 155,0

Какие желательно приобрести для начала обучения ?

★ **DI HALT**

28 Октябрь 2010 в 14:04

AT mega 16 — 16 PU 175,0

оптимальный вариант.

dinaryar

6 Июнь 2011 в 23:43

а чем отличается микропроцессор attiny 25 от attiny 45???

★ **DI HALT**

6 Июнь 2011 в 23:46

Размером памяти. У 25го ее 2кб у 45 четыре.

igor_

1 Август 2011 в 1:16

Вот операция: ld R17,Z

Объясните пожалуйста, как выполняется команда, если Z-регистровая пара, R17 — один регистр? все не влезет в R17. и ZH, и ZL вместе.

★ **DI HALT**

1 Август 2011 в 6:47

Не путай LD и LDI

LDI загрузка непосредственного значения, т.е. числа

LD — загрузка из памяти по адресу Z в Z лежит адрес ячейки откуда надо загрузить. Адрес ессесно двубайтный.

igor_

1 Август 2011 в 10:32

спс. разобрался.

mian

13 Октябрь 2011 в 12:07

Возможно дурацкий вопрос, но все же..

Число циклов перезаписи EEPROM = 100 000.

Имеется в виду, что каждую из ячеек можно перезаписать 100 000 раз, или вообще к EEPROM на запись можно обратиться не более 100 000 раз (например, перезаписать 100 ячеек 1000 раз)?

★ **DI HALT**

13 Октябрь 2011 в 12:21

Каждую ячейку

katelz

15 Сентябрь 2012 в 21:59

Здравствуйте, подскажите пожалуйста. А вот Фьюзы — это часть оперативной памяти, памяти программ или отдельная память?

★ **DI HALT**

16 Сентябрь 2012 в 0:11

Это вообще штука вне мозга мк. Чисто аппаратная заморочка.

FedorSumkin

12 Ноябрь 2012 в 14:22

Здравствуйте Дихалт! Подскажите если можно.

Никак не могу разобраться, как AVR сложение за один такт производит. Есть предположим в регистрах (например R16 и R17) два числа. Строкой ADD R16, R17 мы за один такт складываем оба числа и перезаписываем результат в R16.

АЛУ по документации вроде бы прямо к шине данных подсоединено. А как же АЛУ умудряется за один такт извлечь число из R16 и тут же его перезаписать в R16?

Или у АЛУ все-таки какие-то регистры-защелки на входах есть, где оно промежуточно хранит пришедшие в R16 данные? Или там небольшой временной сдвиг между выборкой из R16 и перезаписью в R16 делается, и таким образом выборка из R16 и перезапись в R16 не мешают друг другу? Заранее Вам спасибо!

★ DI HALT

13 Ноябрь 2012 в 12:47

Полагаю там конвейер, который глядит на следующую инструкцию и начинает ее выполнять еще до того как до нее дойдет очередь. В итоге несколько инструкций выполняются параллельно, просто в разной стадии готовности. Завершаясь каждая на очередном такте.

FedorSumkin

14 Ноябрь 2012 в 14:26

Спасибо Вам большое.

Можно еще спросить. А как микроконтроллер завершает программу (как решает, когда стоп)? Он натывается на пустые инструкции FFFF, когда программа заканчивается, и автоматически стопорится? Я с CVAVR экспериментировал (дизассемблировал в AVRStudo), так он в конце jmp PC-1 ставит. Микроконтроллер как до конца дойдет, тут же мечется на один шаг назад, и так до бесконечности. Хорошо ли это? Может его лучше в сон вгонять по завершении линейной программы?

И еще. Компилятор CVAVR в самом начале полсотни строк каких-то ненужных действий совершает.

. Обнуляет некоторые регистры, определяет прерывания и т п, даже когда это не нужно

Это просто глюк компилятора, я так понимаю, который пытается скомпилировать максимально универсальную программу, и в результате неоптимальный код создает?

Спасибо!

★ **DI HALT**

14 Ноябрь 2012 в 15:51

Нет, пробегает по FFFF и влетает в начало :) ПО хорошему программа должна не иметь конца. Т.е. все возвращается в бесконечном цикле. Ну или как делает компилятор — загоняет программу в пустой цикл.

Стартап код у всех компиляторов жирноват. Инициализация переменных, очистка памяти, настройка стека, подготовка векторов. Это нормально.

FedorSumkin

14 Ноябрь 2012 в 19:20

А, то есть если вырезать в конце команду `jmp PC-1`, программа пробежит до конца все 16 килобайт кода (для мег16) независимо от того, что там записано, обнуляется и самозацикливается? Ух, ну тогда действительно лучше ее искусственно зациклить.

А FFFF в данном случае равносильно команде NOP (хоть это не совсем грамотно)?

koliakrasnoff

13 Февраль 2013 в 8:06

Я вот тоже не пойму, как схемотехнически АЛУ прямо к регистрам общего назначения подсоединено. Отдельная шина данных для них что ли делается? И таким образом можно что-то вычислять и не занимать при этом основную шину данных. В принципе было бы удобно, и быстро.

Это уже не гарвардская, а какая-то тройная гарвардская архитектура получается, с тремя независимыми шинами от памяти программ, от памяти данных и у АЛУ

:-D

Lord_Prizrak

2 Декабрь 2014 в 5:34

У меня тоже вопрос возник, может он и глупый совсем, но он есть.

Ядро контроллера 8ми битное, а большинство команд занимают в памяти в два раза больше — 16 бит, при этом выполняются за один такт. Как это?

★ DI HALT

2 Декабрь 2014 в 19:57

Когда говорят, что контроллер восьмибитный, то имеется в виду его АЛУ. Оно да, имеет 8 разрядов. Так что операции могут осуществляться только с байтом. А вот длина команды может быть любой, за это отвечает совсем другая подсистема.

Lord_Prizrak

18 Декабрь 2014 в 17:24

А можно подробнее про «другая подсистема» или тырк носом где читать?

★ DI HALT

18 Декабрь 2014 в 18:19

Кури устройство управления, АЛУ, внутренняя архитектура. Это аппаратно зашито внутри и программисту недоступно. Но на той же ПЛИС можно самому создать свой МК и тогда все это вылезет в полный рост. На marsohod.org товарищи делали эмуляцию AVR на плис, простенькую, но работала.

bondyara

16 Май 2016 в 18:31

По поводу регистров.

Было бы не плохо, если бы в статье была рекомендация по их применению с высоты опыта автора.

Для себя принял решение распределить их по назначению.

Пока решил так:

R0-R15 не использовать. Регистров и так много (по сравнению с x86, например)

R16-R17 отвел для использования в макросах без сохранения в стек. В коде буду использовать временно на кусках кода, не перекрывающихся макросами. Отпадает необходимость сохранения в стек. Это сократит код, ускорит выполнение, снизит потребление памяти. Количество регистров можно увеличить (???)

Для вывода результатов функции (процедуры) использовать **R20-R25**. Логично — R20 — 0-й результат, R21 — 1-й и т.д.

В то же время R18-R25 можно использовать для прочих нужд, при необходимости сохраняя в стек.

R26-R31 использовать для работы с адресами-словами.

Если регистров не хватит, можно применить их для прочих целей.

Теперь вопрос, насколько оправдано такое распределение? Стоит ли расширить диапазон временных регистров R16-R17, чтобы не менять концепцию в будущем?

Как поступают опытные разработчики?

bondyara

16 Май 2016 в 18:36

Забыл добавить. В некоторых макросах, предлагаемых автором в курсе, сохраняются в стек все используемые регистры, в некоторых — нет, например, используется R16 без сохранения. Чтобы не запоминать, какие макросы портят регистры, целесообразно привести все к общему знаменателю. Либо везде сохранять, либо выделить несохраняемые, как поступил я.