#

# Внутренняя организация микроконтроллеров AVR

Микроконтроллеры AVR выпускаются фирмой Atmel и предназначены для свободного использования в проектируемой аппаратуре.

Общие характеристики микроконтроллеров AVR следующие:

* Напряжение питания – 3-5 В
* Количество регистров общего назначения – 32
* Наличие внутренних ОЗУ, памяти программ, энергонезависимой памяти
* Поддержка интерфейсов USART(RS 232), SPI(I2C)

Архитектура микроконтроллера АТ9052313

Микроконтроллеры AVR являются 8-разрядными. Это означает, что все действия по выполнению арифметических операций, работы с портами, таймерами, интерфейсами выполняются побайтно.

32 регистра общего назначения выполняют функцию аккумулятора. Это означает, что в каждый из этих регистров можно записать и считывать информацию а также то, что результат выполнения арифметических команд можно сохранять в любой из этих регистров.

Арифметика логического устройства предназначена для выполнения операции сложения, вычитания, а также логических операций на аппаратном уровне.

128 байт оперативной памяти, предназначены для хранения данных пока на МК подано питание.

128 EEPROM предназначены для хранения информации после выключения питания МК.

1 Кб памяти программ это флэш-память, в которой находится программа, в соответствии с которой работает МК, а также размещаются таблицы данных (если они необходимы). Эта информация записывается в память на этап программирования и в отличии от EEPROM в процессе работы контролера не может быть изменена.

Счетчик программ – это счетчик 16-разрядный, который начинает считать после включения питания МК и содержание этого счетчика используется для выбора следующего для исполнения команды из памяти программ. Выбранная команда считывается из память программ и помещается в регистр инструкции, а декодер инструкции преобразует информацию в действие.

15 линий ввода/вывода физически являются ножками МК, объединенные в 2 порта ввода/вывода – порт В и порт D. Порт В является 8-разрядным, порт D является 7-разрядным.

Обращение к линиям ввода/вывода может быть как индивидуальным, так и порту в целом, т.е. каждая из линий порта ввода/вывода может быть настроена как на ввод информации, так и на вывод, а данные выводятся и считываются из порта как по битам, так и по байтам.

8-битный таймер-счетчик предназначен для подсчета импульсов внутреннего тактового генератора, а также импульсов, поступающих на счетный вход Т0.

16-разрядный таймер-счетчик предназначен для подсчета тактов, импульсов либо импульсов, поступающих на вход Т1. 8-разрядный таймер-счетчик считает до 256 с коэффициентом предделения от 1 до 1024.

USART – аппаратно реализуемый интерфейс приема/передачи между двумя устройствами в старт/стоповом режиме.

Алгоритм передачи данных и формат пакета передаваемых данных полностью совпадает с интерфейсом RS-232. однако, контролер выдает лишь логическую 1 или 0, а для сопряжения с двухполярным интерфейсом RS-232 необходимо дополнительно использовать соответствующий преобразователь полярности. С помощью этого интерфейса можно передавать информацию со скоростью от нескольких Кбит до 1 Мбит в двух направлениях одновременно.

SPI-модуль в данном МК он предназначен лишь для программирования и является синхронным интерфейсом, в отличии от USART. В более сложных МК он предназначен для синхронной передачи информации между несколькими МК со скоростью до 4 Мбит/с.

Модуль прерываний – это функциональный блок, который обеспечивает прерывания основной программы и переход к выполнению подпрограммы, которая задействуется после соответствующего аппаратного действия.

Аналоговый компаратор предназначен для сравнения напряжений, поданных на 2 соответствующих входа МК. Если напряжение на положительном входе больше, то происходит соответствующее прерывание.

WATCH DOG таймер – это таймер, отсчитывающий интервал времен, в течении которого МК находится в спящем режиме. По окончании счета МК выходит из спящего режима. Спящий режим часто используется в устройствах, которые работают от аккумуляторов, когда нужно снизить энергопотребление и продлить время работы без подзарядки.

Регистр статуса – это регистр, в котором отображается результат выполнения большинства команд МК. Этот регистр представляет собой совокупность битов флагов, каждый из которых установлен в 1 либо сбрасывается в 0, в результате выполнения соответствующей команды.

Флаги в регистре статуса обновляются автоматически, как только выполняется соответствующая команда, обновляющая эти флаги.

Сравнительный анализ современных микроконтролеров представлен ниже:


# Система команд микроконтроллеров серии AVR представлена ниже:

# *Арифметические конструкции МК AVR*

ADD Rd, Rr - суммирование содержимого регистра Rd с содержимым регистра Rr. Результат помещается в Rd.

ADD R16, R17

Если результат больше 256, то флаг переноса – это флаг С в регистре статуса. Регистры Rd и Rr – любые из диапазона 0 Rd, Rr31

ADC Rd, Rr – суммирование с переносом. Содержимое Rd добавляется содержимому Rr и значению флага переноса С.

Пусть R1615

R172

ADC R16, R17

SUB Rd, Rr – вычитание без переноса, т.е. Rd= Rd – Rr.

SUBI Rd, k – вычитание константы k из содержимого регистра Rd, т.е. Rd= Rd – k.

SBC Rd, Rr – вычитание с переносом, т.е. Rd= Rd – Rr – c.

AND Rd, Rr – логическая "И" содержимого двух регистров Rd и Rr.

R16=8=00001000

R17=00001010

AND R16, R17=00001000

ANDI Rd, k – логическая "И" содержимого регистра Rd и константы k.

OR Rd, Rr – логическая "ИЛИ" содержимого двух регистров.

ORI Rd, k - логическая "ИЛИ" содержимого регистра Rd и константы k.

EOR Rd, Rr – исключающая "ИЛИ".

COM Rd – побитная инверсия Rd.

NEG Rd – изменение знака числа.

SBR Rd, k – установить биты в регистре.

Пример. Установить нулевой и третий биты в регистре R17.

SBR R17,0b00001001

SBR R17,9

SBR R17,$09 (0x09)

Устанавливать биты в регистре можно только для регистров 16 – 31.

CBR Rd, k – очистить биты в регистре.

INC Rd – увеличить на 1 содержимое регистра Rd.

DEC Rd – уменьшить на 1 содержимое регистра Rd.

CLR Rd – очистка содержимого регистра Rd.

SER Rd – установить в 1 все биты регистра.

MUL Rd, Rr – умножение без знака.

MULS Rd, Rr – умножение со знаком.

*Инструкции ветвления*

RJMP k – относительный переход на метку k. Переход осуществляется в пределе 2000 команд от места вызова.

RJMP Res

- - - - - - -

Res. ADD R16, R17

IJMP – косвенный переход на адрес, хранящийся в регистровой паре Z.

В микроконтроллере AVR существуют 3 регистровые пары X, Y, Z. Регистровые пары используются при косвенной адресации и позволяют осуществлять переход в пределах 216.

JMP k – переход на метку k в пределах 4 млн. инструкций.

RCALL k – относительный вызов подпрограмм, т.е. переход на метку k, с которой начинается вызываемая подпрограмма. При этом подпрограмма должна заканчиваться командой RET.

ICALL – косвенный вызов подпрограммы, начальный адрес которой хранится в регистровой паре Z.

CALL k – вызов подпрограммы удаленной на расстояние до 64 000 команд от места вызова.

RETI – возврат из прерывания.

При возникновении прерываний в процессоре прекращается выполнение основной программы микроконтроллера. Адрес следующей команды, которую необходимо выполнить помещается в стек и вызывается подпрограмма обрабатывающая это прерывание. При выполнении этой команды адрес с которого мы начинаем переписывать обратно в счетчик прерываний и мы возвращаемся в ту точку с которой прервались.

CPSE Rd, Rr – сравнить содержимое двух регистров и пропустить следующую команду, если значения в эти регистрах одинаково.

CP Rd, Rr – сравнить содержимое двух регистров.

CPI Rd, k – сравнить содержимое регистра с константой.

BREQ k – переход на метку, если выполнилось равенство.

BRNE k – переход на метку, если значение

BRCS k – флаг переноса установлен, т.е. перейти на метку, если установлен флаг переноса.

BRCC k – перейти на метку, если флаг переноса сброшен в 0.

BRSH k – перейти на метку, если равно или больше.

CP R16 R17

BRSH RCS

---------------

RES -------

BRLO k - перейти на метку, если меньше.

BRMI k – перейти на метку, если установлен флаг отрицательного числа.

BRPL k – перейти на метку, если флаг отрицательного числа сброшен в 0.

BRGE k – перейти на метку, если больше или равно с учетом знака.

BRLT k – перейти на метку, если меньше с учетом знака.

BRHS k – перейти на метку, если флаг внутреннего переноса установлен.

BRHC k – перейти на метку, если флаг внутреннего переноса сброшен в 0.

BRTS k – перейти на метку, если флаг Т установлен.

BRTC k – перейти на метку, если флаг Т сброшен в 0.

BRVS k – перейти на метку, если флаг перевыполнения установлен.

BRVC k – перейти на метку, если флаг перевыполнения сброшен в 0.

BRIE k – перейти на метку, если разрешены прерывания.

BRID k – перейти на метку, если прерывания запрещены.

SBRC Rd, b – пропустить следующую команду, если бит b в регистре Rd сброшен в 0.

SBRS Rd, b – пропустить следующую команду, если бит b в регистре Rd установлен.

SBIC P, b – пропустить следующую команду, если бит b в порту Р сброшен в 0.

SBIS P, b – пропустить следующую команду, если бит b в порту Р установлен.

BRBS S, k – перейти на метку k, если флаг S в регистре статуса установлен.

BRBC S, k – перейти на метку k, если флаг в регистре статуса сброшен в 0.

*Команды работы с битами.*

LSL Rd – логический сдвиг влево. В результате логического сдвига влево старший бит в сдвигаемом регистре перемещается во флаг переноса С, все биты регистра сдвигаются на 1 бит влево, а в младший бит записывается 0.

LSR Rd - логический сдвиг вправо.

ROL Rd – циклический сдвиг влево.

При циклическом сдвиге влево, содержимое сдвигаемого регистра сдвигается на один бит влево, при этом старший бит переходит во флаг переноса B, а то что было во флаге С до этого записывается в младший бит записываемого регистра.

ROR Rd – циклический сдвиг вправо.

ASR Rd – арифметический сдвиг вправо.

SWAP Rd – перестановка тетрад в регистре.

SBI P, b – установить бит b в порту Р.

CBI P, b – сбросить бит b в порту Р.

BST Rd, b – установить либо сбросить флаг Т в регистре статуса в зависимости от того установлен или сброшен бит в регистре Rd.

BLD Rd, b – загрузить в бит b регистра Rd содержимое флага Т.

SEC – установить в 1 флаг переноса.

CLC – сбросить в 0 флаг переноса.

SEN – установить в 1 флаг отрицательного числа.

CLN – сбросить в 0 флаг отрицательного числа.

SEZ – установить в 1 флаг 0.

CLZ – сбросить в 0.

SEI – разрешить прерывания.

CLI – запретить прерывания.

SES – установить в 1 флаг числа со знаком.

CLS – сбросить в 0 флаг числа со знаком.

SEV - установить в 1 флаг переполнения.

CLV – сбросить в 0 флаг переполнения.

SET – установить в 1 флаг Т.

CLT – сбросить в 0 флаг Т

SEH – установить в 1 флаг внутреннего переноса.

CLH – очистить флаг внутреннего переноса.

NOP – пустая команда.

SLEEP – переход в режим пониженного энергопотребления.

WDR – сброс сторожевого таймера.

*Команды передачи данных*

MOV Rd, Rr – копирование содержимого регистра Rr в Rd.

LDI Rd, k – загрузить константу k в регистр Rd. Команда справедлива только для регистров с 16 по 31.

LDS Rd, k – прямая загрузка из оперативной памяти значений хранящихся по адресу k в регистре Rd.

LD Rd, x – загрузка из ОЗУ значений хранящихся по адресу, который предварительно записан в регистровую пару х, значение в регистр Rd

# LDI R26, 0X01

LDI R27, 0X02

LD R17, x

LD Rd, x+ (LD Rd, y+ LD Rd, z+) – косвенная загрузка в регистр Rd значений из ОЗУ, адрес которой записан в регистровой паре х с последующим инкрементом адреса.

LD Rd, -x – загрузка в регистр Rd значений из ОЗУ, адрес которых предварительно записан регистровой паре х с прединкрементом адреса.

STS k, Rr – прямое сохранение содержимого регистра Rr в ячейку ОЗУ, адрес которой указан константой k.

STS 0X69, R17

ST x, Rr (ST Y, Rr ST Y, Rr) – сохранение содержимого регистра Rr в ячейку оперативной памяти, адрес которой предварительно записан в регистровой паре х.

ST x+, Rr (ST Y+, Rr ST Z+, Rr) – сохранение содержимого регистра Rr в ячейку оперативной памяти с адресом, записанным в регистровую пару х и последующим инкрементом адреса.

ST -x, Rr (ST -Y, Rr ST -Z, Rr) - сохранение содержимого регистра Rr в ячейку оперативной памяти с адресом, предварительно записанным в регистровую пару х и преддектрементом адреса.

LDD Rd, Y+q (LDD Rd, Z+q) – загрузка из памяти в регистр Rd, когда адрес хранится в регистровой паре и указывается смещение адреса q.

IN Rd, P – загрузка из порта Р его содержимого в регистр Rd.

OUT P, Rr – вывод в порт Р содержимого регистра Rr.

PUSH Rd – занесение содержимого регистра в стек.

POP Rd – извлечение из стека.

LPM – загрузка значений из памяти программ.

LPM Rd, Z+ - загрузка из памяти программ с постинкрементом.