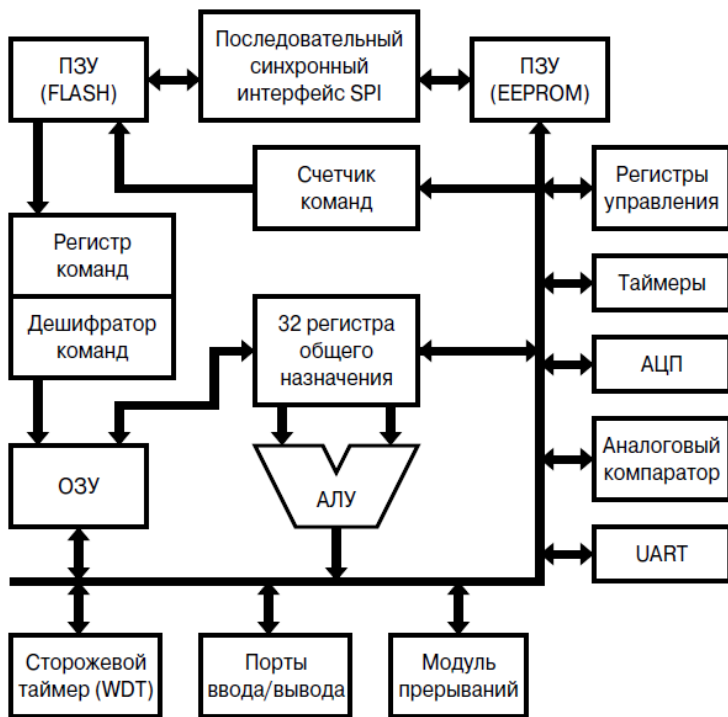
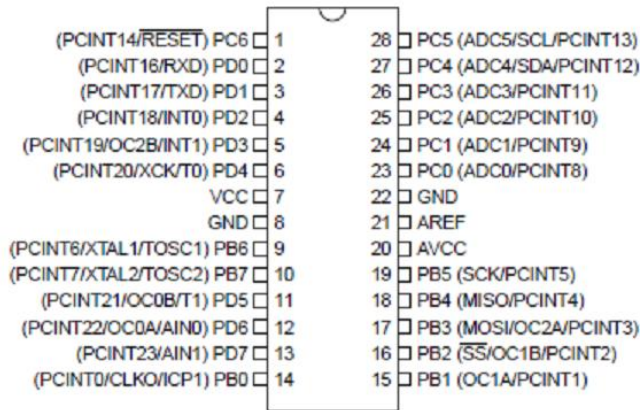


АТmega88 микроконтроллер

28-Pin PDIP Package ja 32-pin TQFP package.



Архитектура ядра микроконтроллера AVR АТmega88

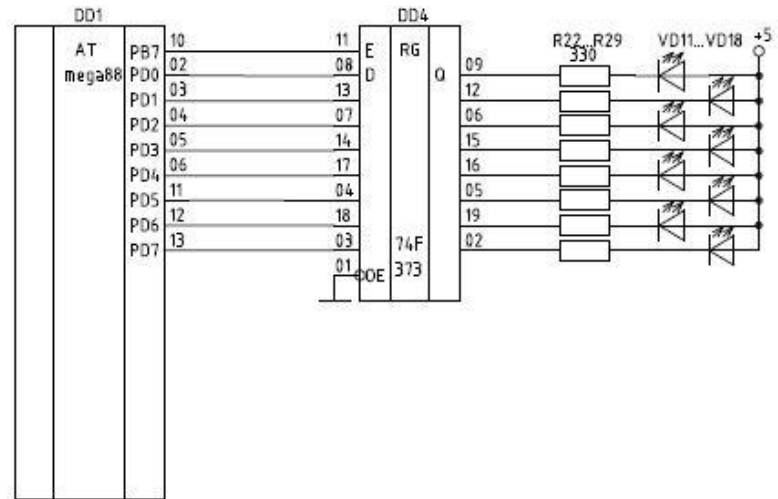
Atmel АТmega88

8-битный микроконтроллер AVR фирмы ATMEL АТmega88

Характеристики контроллера:

- **Усовершенствованная RISC архитектура:**
 - 131 мощных команд – большинство команд выполняется за один такт;
 - 32 x 8 рабочих регистров, объединенные в регистровый файл;
 - вычислений скорость до 20 MIPS/МГц;
- **Высокая выносливость энергонезависимой памяти:**
 - FLASH-память программ с объемом 8 Кбайт (число циклов стирания/записи не менее 10000);
 - память данных на основе статического ОЗУ (SRAM) с объемом 1 Кбайт;
 - память данных на основе ЭСППЗУ (EEPROM) с объемом 512 байт (число циклов стирания/записи не менее 100000);
- **Программирование микроконтроллера непосредственно в системе (ISP - In System Programming), используя программу загрузки микроконтроллера (OBP - On-chip Boot Program).**
- **Периферийные устройства:**
 - 8-разрядный таймер/счетчик с предделителем (таймер T0);
 - 16-разрядный таймер/счетчик с предделителем (таймер T1);
 - 8-разрядный таймер/счетчик с возможностью работы в асинхронном режиме из внешнего генератора (таймер T2);
 - сторожевой таймер (WDT - Watchdog)1;
 - одно- или двухканальный генератор сигнала с Широтно-Импульсной Модуляцией (ШИМ) (PWM - Pulse-width modulation);
 - одноканальный 8-разрядный генератор сигнала с ШИМ;
 - аналоговый компаратор;
 - 10-разрядный Аналого-Цифровой Преобразователь (АЦП) с 6-ю каналами;
 - полнодуплексный универсальный асинхронный приемопередатчик (UART);
 - последовательный синхронный интерфейс (SPI - Serial Peripheral Interface);
 - прерывание и пробуждение от изменения на выводах контроллера;
- **Специальные характеристики микроконтроллера**
 - сброс (Reset) при включении питания;
 - внутренняя настройка генератора (Calibrated Oscillator);
 - внутреннее и внешнее прерывание;
 - пять вида «спящих» режима: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby.
- **Порты ввода/вывода (В/В) и корпуса контроллера**
 - 23 программируемых выводов В/В;
 - нагрузочная способность всех выводов составляет до 20 мА,
 - четыре разных типа корпусов: 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF;
- **Напряжение и ток питания микроконтроллера**
 - 1.8 - 5.5V для АТmega88V;
 - 2.7 - 5.5V для АТmega88;
 - в активном режиме:
 - 250 μА при частоте генератора 1 MHz и питание 1.8V
 - 15 μА при 32 kHz и 1.8V
 - в «спящих» режиме Power-down 0.1μА при питания 1.8V

Обозначение	Операнд	Описание	Функция	Флаги
ADD	Rd, Rr	Add two Registers Сложить без переноса	$Rd < Rd + Rr$	Z,C,N,V,H
ADC	Rd, Rr	Add with Carry two Registers Сложить с переносом	$Rd < Rd + Rr + C$	Z,C,N,V,H
SUB	Rd, Rr	Subtract two Registers Вычесть без переноса	$Rd < Rd - Rr$	Z,C,N,V,H
SUBI	Rd, K	Subtract Constant from Register Вычесть непосредственное значение	$Rd < Rd - K$	Z,C,N,V,H
SBC	Rd, Rr	Subtract with Carry two Registers Вычесть с переносом	$Rd < Rd - Rr - C$	Z,C,N,V,H
SBCI	Rd, K	Subtract with Carry Constant from Reg. Вычесть непосредственное значение с переносом	$Rd < Rd - K - C$	Z,C,N,V,H
AND	Rd, Rr	Logical AND Registers Выполнить логическое «И»	$Rd < Rd \cdot Rr$	Z,N,V
ANDI	Rd, K	Logical AND Register and Constant Выполнить логическое «И» с непосредственным значением	$Rd < Rd \cdot K$	Z,N,V
INC	Rd	Increment; Инкрементировать	$Rd < Rd + 1$	Z,N,V
DEC	Rd	Decrement; Декрементировать	$Rd < Rd - 1$	Z,N,V
SEC		Set Carry; Установить флаг переноса	$C < 1$	C
CLC		Clear Carry; Сброс флаг переноса	$C < 0$	C
TST	Rd	Test for Zero or Minus; Проверить на ноль или минус	$Rd < Rd \cdot Rd$	Z,N,V
SBI	P,b	Set Bit in I/O Register Установить бит в регистр I/O (B/B)	$I/O(P,b) < 1$	None
CBI	P,b	Clear Bit in I/O Register Сброс бит в регистр I/O (B/B)	$I/O(P,b) < 0$	None
LSL	Rd	Logical Shift Left; Логически сдвинуть влево	$Rd(n+1) < Rd(n), Rd(0) < 0$	Z,C,N,V
ROL	Rd	Rotate Left Through Carry Сдвинуть влево через перенос	$Rd(0) < C, Rd(n+1) < Rd(n), C < Rd(7)$	Z,C,N,V
ROR	Rd	Rotate Right Through Carry Сдвинуть вправо через перенос	$Rd(7) < C, Rd(n) < Rd(n+1), C < Rd(0)$	Z,C,N,V
MOV	Rd, Rr	Move Between Registers Копировать регистр	Rd, Rr	None
LDI	Rd, K	Load Immediate Загрузить непосредственное значение	$Rd < K$	None
IN	Rd, P	In Port; Загрузить данные из порта I/O в регистр	Rd,P	None
OUT	P, Rr	Out Port; Записать данные из регистра в порт I/O	P,Rr	None
RJMP	k	Relative Jump; Относительный переход	$PC < PC + k + 1$	None
RCALL	k	Relative Subroutine Call; Относительный переход на подпрограмму	$PC < PC + k + 1$	None
RET		Subroutine Return Возврат из подпрограммы	$PC < STACK$	None
BRNE	s, k	Branch if Not Equal; Переход если не равно	if (Z = 0) then $PC < PC + k + 1$	None
BREQ	s, k	Branch if Equal; Переход если равно	if (Z = 1) then $pc < PC + k + 1$	None



Input/Output Ports

PORTB	0x05(0x25)	0x00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DDRB	0x04(0x24)		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PINB	0x03(0x23)		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PORTC	0x08(0x28)	0x00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DDRC	0x07(0x27)		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PINC	0x06(0x26)		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PORTD	0x0B(0x2B)		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
DDRD	0x0A(0x2A)		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
PIND	0x09(0x29)		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```

ldi r16,0x80 ; 1000 0000
out DDRB,r16 ; PB7=1 Выход - для импульса Enable.
; оставшиеся входы
ldi r16,0xFF ; 1111 1111
out DDRD,r16 ; PD все выходы
ldi r16, 0xfe ; раб. регистр r16 = 0xfe или 0b11111110
out portd, r16 ; содержание рег. r16 осваивается к portD
;Если разряд=0 то LED «горит»

```

```

;--- ENABLE строб для регистра 74F373 -----
sbi portb, 7 ; установка 7-го разряда порта portB
cbi portb, 7 ; сброс 7-го разряда порта portB
Loop: rjmp Loop ;STOP (бесконечный цикл)

```