

Features

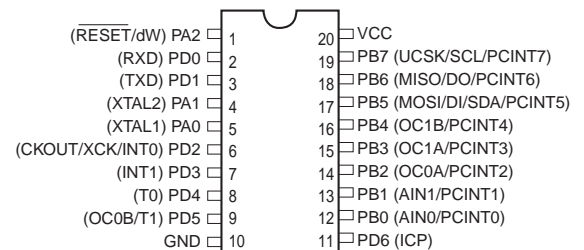
- Utilizes the AVR® RISC Architecture
- AVR – High-performance and Low-power RISC Architecture
 - 120 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
- Data and Non-volatile Program and Data Memories
 - 2K Bytes of In-System Self Programmable Flash
 - Endurance 10,000 Write/Erase Cycles
 - 128 Bytes In-System Programmable EEPROM
 - Endurance: 100,000 Write/Erase Cycles
 - 128 Bytes Internal SRAM
 - Programming Lock for Flash Program and EEPROM Data Security
- Peripheral Features
 - One 8-bit Timer/Counter with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Modes
 - Four PWM Channels
 - On-chip Analog Comparator
 - Programmable Watchdog Timer with On-chip Oscillator
 - USI – Universal Serial Interface
 - Full Duplex USART
- Special Microcontroller Features
 - debugWIRE On-chip Debugging
 - In-System Programmable via SPI Port
 - External and Internal Interrupt Sources
 - Low-power Idle, Power-down, and Standby Modes
 - Enhanced Power-on Reset Circuit
 - Programmable Brown-out Detection Circuit
 - Internal Calibrated Oscillator
- I/O and Packages
 - 18 Programmable I/O Lines
 - 20-pin PDIP, 20-pin SOIC, 20-pad QFN/MLF
- Operating Voltages
 - 1.8 - 5.5V (ATtiny2313V)
 - 2.7 - 5.5V (ATtiny2313)
- Speed Grades
 - ATtiny2313V: 0 - 4 MHz @ 1.8 - 5.5V, 0 - 10 MHz @ 2.7 - 5.5V
 - ATtiny2313: 0 - 10 MHz @ 2.7 - 5.5V, 0 - 20 MHz @ 4.5 - 5.5V
- Typical Power Consumption
 - Active Mode
 - 1 MHz, 1.8V: 230 µA
 - 32 kHz, 1.8V: 20 µA (including oscillator)
 - Power-down Mode
 - < 0.1 µA at 1.8V

8-bit Microcontroller with 2K Bytes In-System Programmable

Flash

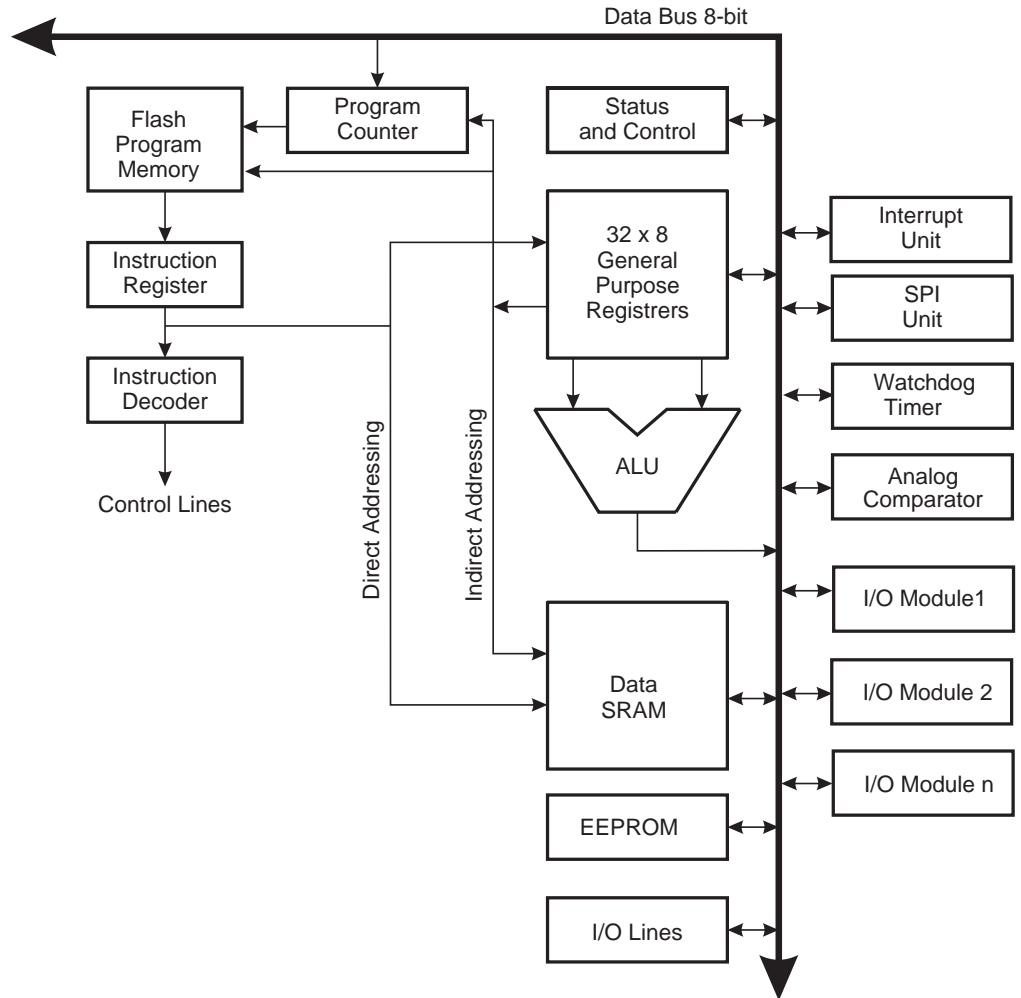
ATtiny2313/V

PDIP/SOIC



Overview

The ATtiny2313 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATtiny2313 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATtiny2313 provides the following features: 2K bytes of In-System Programmable Flash, 128 bytes EEPROM, 128 bytes SRAM, 18 general purpose I/O lines, 32 general purpose working registers, a single-wire Interface for On-chip Debugging, two flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, Universal Serial Interface with Start Condition Detector, a programmable Watchdog Timer with internal Oscillator, and three software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, or by a conventional non-volatile memory programmer. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATtiny2313 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATtiny2313 AVR is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x3F (0x5F)	SREG	I	T	H	S	V	N	Z	C	7
0x3E (0x5E)	Reserved	-	-	-	-	-	-	-	-	
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	10
0x3C (0x5C)	OCR0B	Timer/Counter0 – Compare Register B								78
0x3B (0x5B)	GIMSK	INT1	INT0	PCIE	-	-	-	-	-	60
0x3A (0x5A)	EIFR	INTF1	INTF0	PCIF	-	-	-	-	-	62
0x39 (0x59)	TIMSK	TOIE1	OCIE1A	OCIE1B	-	ICIE1	OCIE0B	TOIE0	OCIE0A	79, 110
0x38 (0x58)	TIFR	TOV1	OCF1A	OCF1B	-	ICF1	OCF0B	TOV0	OCF0A	79
0x37 (0x57)	SPMCSR	-	-	-	CTPB	RFLB	PGWRT	PGERS	SELFPRGEN	156
0x36 (0x56)	OCR0A	Timer/Counter0 – Compare Register A								78
0x35 (0x55)	MCUCR	PUD	SM1	SE	SM0	ISC11	ISC10	ISC01	ISC00	53
0x34 (0x54)	MCUSR	-	-	-	-	WDRF	BORF	EXTRF	PORF	37
0x33 (0x53)	TCCR0B	FOC0A	FOC0B	-	-	WGM02	CS02	CS01	CS00	77
0x32 (0x52)	TCNT0	Timer/Counter0 (8-bit)								78
0x31 (0x51)	OSCCAL	-	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	25
0x30 (0x50)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	-	-	WGM01	WGM00	74
0x2F (0x4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	WGM11	WGM10	105
0x2E (0x4E)	TCCR1B	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10	108
0x2D (0x4D)	TCNT1H	Timer/Counter1 – Counter Register High Byte								109
0x2C (0x4C)	TCNT1L	Timer/Counter1 – Counter Register Low Byte								109
0x2B (0x4B)	OCR1AH	Timer/Counter1 – Compare Register A High Byte								109
0x2A (0x4A)	OCR1AL	Timer/Counter1 – Compare Register A Low Byte								109
0x29 (0x49)	OCR1BH	Timer/Counter1 – Compare Register B High Byte								110
0x28 (0x48)	OCR1BL	Timer/Counter1 – Compare Register B Low Byte								110
0x27 (0x47)	Reserved	-	-	-	-	-	-	-	-	
0x26 (0x46)	CLKPR	CLKPCE	-	-	-	CLKPS3	CLKPS2	CLKPS1	CLKPS0	27
0x25 (0x45)	ICR1H	Timer/Counter1 - Input Capture Register High Byte								110
0x24 (0x44)	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								110
0x23 (0x43)	GTCCR	-	-	-	-	-	-	-	PSR10	82
0x22 (0x42)	TCCR1C	FOC1A	FOC1B	-	-	-	-	-	-	109
0x21 (0x41)	WDTCR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	42
0x20 (0x40)	PCMSK	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	62
0x1F (0x3F)	Reserved	-	-	-	-	-	-	-	-	
0x1E (0x3E)	EEAR	EEPROM Address Register								15
0x1D (0x3D)	EEDR	EEPROM Data Register								16
0x1C (0x3C)	EEDR	-	-	EEM1	EEM0	EERIE	EEMPE	EEPE	EERE	16
0x1B (0x3B)	PORTA	-	-	-	-	-	PORTA2	PORTA1	PORTA0	58
0x1A (0x3A)	DDRA	-	-	-	-	-	DDA2	DDA1	DDA0	58
0x19 (0x39)	PINA	-	-	-	-	-	PINA2	PINA1	PINA0	58
0x18 (0x38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	58
0x17 (0x37)	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	58
0x16 (0x36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	58
0x15 (0x35)	GPOR2	General Purpose I/O Register 2								20
0x14 (0x34)	GPOR1	General Purpose I/O Register 1								20
0x13 (0x33)	GPOR0	General Purpose I/O Register 0								20
0x12 (0x32)	PORTD	-	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	58
0x11 (0x31)	DDRD	-	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	58
0x10 (0x30)	PIND	-	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	58
0x0F (0x2F)	USIDR	USI Data Register								145
0x0E (0x2E)	USISR	USISIF	USIOIF	USIPF	USIDC	USICNT3	USICNT2	USICNT1	USICNT0	146
0x0D (0x2D)	USICR	USISIE	USIOIE	USIWM1	USIWM0	USICS1	USICS0	USICLK	USITC	147
0x0C (0x2C)	UDR	UART Data Register (8-bit)								130
0x0B (0x2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	UPE	U2X	MPCM	130
0x0A (0x2A)	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	132
0x09 (0x29)	UBRRH	UBRRH[7:0]								134
0x08 (0x28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	150
0x07 (0x27)	Reserved	-	-	-	-	-	-	-	-	
0x06 (0x26)	Reserved	-	-	-	-	-	-	-	-	
0x05 (0x25)	Reserved	-	-	-	-	-	-	-	-	
0x04 (0x24)	Reserved	-	-	-	-	-	-	-	-	
0x03 (0x23)	UCSRC	-	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	133
0x02 (0x22)	UBRRH	-	-	-	-	UBRRH[11:8]				134
0x01 (0x21)	DIDR	-	-	-	-	-	-	AIN1D	AIN0D	151
0x00 (0x20)	Reserved	-	-	-	-	-	-	-	-	

Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	Rd,K	Add Immediate to Word	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract two Registers	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	Rd,K	Subtract Immediate from Word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND Registers	$Rd \leftarrow Rd \bullet Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	$Rd \leftarrow Rd \bullet K$	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	One's Complement	$Rd \leftarrow 0xFF - Rd$	Z,C,N,V	1
NEG	Rd	Two's Complement	$Rd \leftarrow 0x00 - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	$Rd \leftarrow Rd \bullet (0xFF - K)$	Z,N,V	1
INC	Rd	Increment	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrement	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Test for Zero or Minus	$Rd \leftarrow Rd \bullet Rd$	Z,N,V	1
CLR	Rd	Clear Register	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Set Register	$Rd \leftarrow 0xFF$	None	1
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	$PC \leftarrow PC + k + 1$	None	2
IJMP		Indirect Jump to (Z)	$PC \leftarrow Z$	None	2
RCALL	k	Relative Subroutine Call	$PC \leftarrow PC + k + 1$	None	3
ICALL		Indirect Call to (Z)	$PC \leftarrow Z$	None	3
RET		Subroutine Return	$PC \leftarrow STACK$	None	4
RETI		Interrupt Return	$PC \leftarrow STACK$	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) $PC \leftarrow PC + 2$ or 3	None	1/2/3
CP	Rd,Rr	Compare	$Rd - Rr$	Z, N, V, C, H	1
CPC	Rd,Rr	Compare with Carry	$Rd - Rr - C$	Z, N, V, C, H	1
CPI	Rd,K	Compare Register with Immediate	$Rd - K$	Z, N, V, C, H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIC	P, b	Skip if Bit in I/O Register Cleared	if (P(b)=0) $PC \leftarrow PC + 2$ or 3	None	1/2/3
SBIS	P, b	Skip if Bit in I/O Register is Set	if (P(b)=1) $PC \leftarrow PC + 2$ or 3	None	1/2/3
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BREQ	k	Branch if Equal	if (Z = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRNE	k	Branch if Not Equal	if (Z = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRCS	k	Branch if Carry Set	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRCC	k	Branch if Carry Cleared	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRSH	k	Branch if Same or Higher	if (C = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRLO	k	Branch if Lower	if (C = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRMI	k	Branch if Minus	if (N = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRPL	k	Branch if Plus	if (N = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRGE	k	Branch if Greater or Equal, Signed	if (N \oplus V = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRLT	k	Branch if Less Than Zero, Signed	if (N \oplus V = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRTS	k	Branch if T Flag Set	if (T = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then $PC \leftarrow PC + k + 1$	None	1/2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then $PC \leftarrow PC + k + 1$	None	1/2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	$I/O(P,b) \leftarrow 1$	None	2
CBI	P,b	Clear Bit in I/O Register	$I/O(P,b) \leftarrow 0$	None	2
LSL	Rd	Logical Shift Left	$Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$	Z,C,N,V	1
LSR	Rd	Logical Shift Right	$Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	$Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$	Z,C,N,V	1

Mnemonics	Operands	Description	Operation	Flags	#Clocks
ROR	Rd	Rotate Right Through Carry	$Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	$Rd(n) \leftarrow Rd(n+1), n=0..6$	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	$Rd(3..0) \leftarrow Rd(7..4), Rd(7..4) \leftarrow Rd(3..0)$	None	1
BSET	s	Flag Set	$SREG(s) \leftarrow 1$	SREG(s)	1
BCLR	s	Flag Clear	$SREG(s) \leftarrow 0$	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	$T \leftarrow Rr(b)$	T	1
BLD	Rd, b	Bit load from T to Register	$Rd(b) \leftarrow T$	None	1
SEC		Set Carry	$C \leftarrow 1$	C	1
CLC		Clear Carry	$C \leftarrow 0$	C	1
SEN		Set Negative Flag	$N \leftarrow 1$	N	1
CLN		Clear Negative Flag	$N \leftarrow 0$	N	1
SEZ		Set Zero Flag	$Z \leftarrow 1$	Z	1
CLZ		Clear Zero Flag	$Z \leftarrow 0$	Z	1
SEI		Global Interrupt Enable	$I \leftarrow 1$	I	1
CLI		Global Interrupt Disable	$I \leftarrow 0$	I	1
SES		Set Signed Test Flag	$S \leftarrow 1$	S	1
CLS		Clear Signed Test Flag	$S \leftarrow 0$	S	1
SEV		Set Twos Complement Overflow	$V \leftarrow 1$	V	1
CLV		Clear Twos Complement Overflow	$V \leftarrow 0$	V	1
SET		Set T in SREG	$T \leftarrow 1$	T	1
CLT		Clear T in SREG	$T \leftarrow 0$	T	1
SEH		Set Half Carry Flag in SREG	$H \leftarrow 1$	H	1
CLH		Clear Half Carry Flag in SREG	$H \leftarrow 0$	H	1
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	$Rd \leftarrow Rr$	None	1
MOVW	Rd, Rr	Copy Register Word	$Rd+1:Rd \leftarrow Rr+1:Rr$	None	1
LDI	Rd, K	Load Immediate	$Rd \leftarrow K$	None	1
LD	Rd, X	Load Indirect	$Rd \leftarrow (X)$	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	$Rd \leftarrow (X), X \leftarrow X + 1$	None	2
LD	Rd, -X	Load Indirect and Pre-Dec.	$X \leftarrow X - 1, Rd \leftarrow (X)$	None	2
LD	Rd, Y	Load Indirect	$Rd \leftarrow (Y)$	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	$Rd \leftarrow (Y), Y \leftarrow Y + 1$	None	2
LD	Rd, -Y	Load Indirect and Pre-Dec.	$Y \leftarrow Y - 1, Rd \leftarrow (Y)$	None	2
LDD	Rd, Y+q	Load Indirect with Displacement	$Rd \leftarrow (Y + q)$	None	2
LD	Rd, Z	Load Indirect	$Rd \leftarrow (Z)$	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	$Z \leftarrow Z - 1, Rd \leftarrow (Z)$	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	$Rd \leftarrow (Z + q)$	None	2
LDS	Rd, k	Load Direct from SRAM	$Rd \leftarrow (k)$	None	2
ST	X, Rr	Store Indirect	$(X) \leftarrow Rr$	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	$(X) \leftarrow Rr, X \leftarrow X + 1$	None	2
ST	-X, Rr	Store Indirect and Pre-Dec.	$X \leftarrow X - 1, (X) \leftarrow Rr$	None	2
ST	Y, Rr	Store Indirect	$(Y) \leftarrow Rr$	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	$(Y) \leftarrow Rr, Y \leftarrow Y + 1$	None	2
ST	-Y, Rr	Store Indirect and Pre-Dec.	$Y \leftarrow Y - 1, (Y) \leftarrow Rr$	None	2
STD	Y+q, Rr	Store Indirect with Displacement	$(Y + q) \leftarrow Rr$	None	2
ST	Z, Rr	Store Indirect	$(Z) \leftarrow Rr$	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	$(Z) \leftarrow Rr, Z \leftarrow Z + 1$	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	$Z \leftarrow Z - 1, (Z) \leftarrow Rr$	None	2
STD	Z+q, Rr	Store Indirect with Displacement	$(Z + q) \leftarrow Rr$	None	2
STS	k, Rr	Store Direct to SRAM	$(k) \leftarrow Rr$	None	2
LPM		Load Program Memory	$R0 \leftarrow (Z)$	None	3
LPM	Rd, Z	Load Program Memory	$Rd \leftarrow (Z)$	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	$Rd \leftarrow (Z), Z \leftarrow Z + 1$	None	3
SPM		Store Program Memory	$(Z) \leftarrow R1:R0$	None	-
IN	Rd, P	In Port	$Rd \leftarrow P$	None	1
OUT	P, Rr	Out Port	$P \leftarrow Rr$	None	1
PUSH	Rr	Push Register on Stack	$STACK \leftarrow Rr$	None	2
POP	Rd	Pop Register from Stack	$Rd \leftarrow STACK$	None	2
MCU CONTROL INSTRUCTIONS					
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1
BREAK		Break	For On-chip Debug Only	None	N/A

AVR-RISC Mikrokontrollerite testprogrammid

ATtini2313 lülilid ja signaallambid

```

AVRASM ver. 1.30 SW_TEST.ASM Sun Feb 10 14:41:27 2002
;***** AT90S1200 *****
; **Test, mis kontrollib lüliteid ja signaallampe**
; Kui SW1-le või SW2-le vajutada SIIS loindatakse
; Kui SW3-le vajutada SIIS kustutatakse
; Kui SW4-le või SW5-le vajutada SIIS osa signaal-
; lampe põleb ja osa ei
; **** Initialization *****
000000 e000 ldi r16,$00 ; r16 on tööregister
; ajutise info jaoks.
000001 bb01 out $11,r16 ; PortD ($10) on sisend-
; register SW.
; Suuna määrab DDRD ($11)
; DDD0 - DDD7 := 00000000
; Kõik on sisendid.
000002 ef0f ldi r16,0xff ; r16 tööregister
000003 bb07 out 0x17,r16 ; PortB ($18) on väljund-
; register LED. Suund ($17)
; DDB0 - DDB7 := 11111111
; Kõik on väljundid.
; NB! 2-nd arv 1101 0111 võib AVR Assembleri
; jaoks kirjutada kujul kas $d7 või 0xd7
; **** Testida lüliteid *****
; LED-id põlevad, kui pordi väljund on "0" ja
; on kustunud, kui pordi väljund on "1".
; Algul kustutatakse kõik LED-id.
000004 bb08 LOOP: out 0x18,r16;Panna paika LED-id.
000005 9b80 sbis 0x10,0 ;Kui PortD, pin0 == 0 (SW1 sees)
000006 9503 inc r16 ; SIIS Loendada suuremaks
000007 9b81 sbis 0x10,1 ;Kui PortD, pin1 == 0 (SW2 sees)
000008 950a dec r16 ; SIIS Loendada väiksemaks
000009 9b82 sbis 0x10,2 ;Kui PortD, pin2 == 0 (SW3 sees)
00000a ef0f ldi r16,0xff; SIIS Kustutada
00000b 9b83 sbis 0x10,3 ;Kui PortD, pin3 == 0 (SW4 sees)
00000c e505 ldi r16,0x55; SIIS üks osa põlema
00000d 9b84 sbis 0x10,4 ;Kui PortD, pin4 == 0 (SW5 sees)
00000e ea0a ldi r16,0xAA; SIIS teine osa põlema
; **** Ootamine, enne kui väljastada LED-idele *****
00000f 951a DLY: dec r17
000010 f7f1 brne DLY
000011 952a dec r18
000012 f7e1 brne DLY
000013 cff0 rjmp LOOP
Assembly complete with no errors.

```