

24. 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	10
0x3E (0x5E)	SPH	–	–	–	–	–	–	SP9	SP8	13
0x3D (0x5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	13
0x3C (0x5C)	OCR0B	Timer/Counter0 – Output compare register B								76
0x3B (0x5B)	GIMSK	–	INT0	PCIE1	PCIE0	–	–	–	–	47
0x3A (0x5A)	GIFR	–	INTF0	PCIF1	PCIF0	–	–	–	–	48
0x39 (0x59)	TIMSK0	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	77
0x38 (0x58)	TIFR0	–	–	–	–	–	OCF0B	OCF0A	TOV0	77
0x37 (0x57)	SPMCSR	–	–	–	CTPB	RFLB	PGWRT	PGERS	SPMEN	140
0x36 (0x56)	OCR0A	Timer/Counter0 – Output compare register A								76
0x35 (0x55)	MCUCR	BODS	PUD	SE	SM1	SM0	BODSE	ISC01	ISC00	47
0x34 (0x54)	MCUSR	–	–	–	–	WDRF	BORF	EXTRF	PORF	41
0x33 (0x53)	TCCR0B	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	75
0x32 (0x52)	TCNT0	Timer/Counter0								76
0x31 (0x51)	OSCCAL	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	29
0x30 (0x50)	TCCR0A	COM01	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	73
0x2F (0x4F)	TCCR1A	COM11	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10	97
0x2E (0x4E)	TCCR1B	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	99
0x2D (0x4D)	TCNT1H	Timer/Counter1 – Counter register high byte								100
0x2C (0x4C)	TCNT1L	Timer/Counter1 – Counter register low byte								100
0x2B (0x4B)	OCR1AH	Timer/Counter1 – Compare register A high byte								100
0x2A (0x4A)	OCR1AL	Timer/Counter1 – Compare register A low byte								100
0x29 (0x49)	OCR1BH	Timer/Counter1 – Compare register B high byte								101
0x28 (0x48)	OCR1BL	Timer/Counter1 – Compare register B low byte								101
0x27 (0x47)	DWDR	DWDR[7:0]								136
0x26 (0x46)	CLKPR	CLKPCE	–	–	–	CLKPS3	CLKPS2	CLKPS1	CLKPS0	29
0x25 (0x45)	ICR1H	Timer/Counter1 - Input capture register high byte								101
0x24 (0x44)	ICR1L	Timer/Counter1 - Input capture register low byte								101
0x23 (0x43)	GTCCR	TSM	–	–	–	–	–	–	PSR10	104
0x22 (0x42)	TCCR1C	FOC1A	FOC1B	–	–	–	–	–	–	100
0x21 (0x41)	WDTCR	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	41
0x20 (0x40)	PCMSK1	–	–	–	–	PCINT11	PCINT10	PCINT9	PCINT8	48
0x1F (0x3F)	EEARH	–	–	–	–	–	–	–	EEAR8	21
0x1E (0x3E)	EEARL	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	21
0x1D (0x3D)	EEDR	EEPROM data register								21
0x1C (0x3C)	EEDR	–	–	EEDR1	EEDR0	EEDR7	EEDR6	EEDR5	EEDR4	22
0x1B (0x3B)	PORTA	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	61
0x1A (0x3A)	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	61
0x19 (0x39)	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	62

- Notes:
- For compatibility with future devices, reserved bits should be written to logical zero if accessed. Reserved I/O memory addresses should never be written.
 - I/O registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
 - Some of the status flags are cleared by writing a logical one to them. Note that, unlike most other AVR®, the CBI and SBI instructions will only operation the specified bit, and can therefore be used on registers containing such status flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

24. Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x18 (0x38)	PORTB	–	–	–	–	PORTB3	PORTB2	PORTB1	PORTB0	62
0x17 (0x37)	DDRB	–	–	–	–	DDB3	DDB2	DDB1	DDB0	62
0x16 (0x36)	PINB	–	–	–	–	PINB3	PINB2	PINB1	PINB0	62
0x15 (0x35)	GPIOR2	General purpose I/O register 2								23
0x14 (0x34)	GPIOR1	General purpose I/O register 1								23
0x13 (0x33)	GPIOR0	General purpose I/O register 0								23
0x12 (0x32)	PCMSK0	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	48
0x11 (0x31)	Reserved	–								
0x10 (0x30)	USIBR	USI buffer register								111
0x0F (0x2F)	USIDR	USI data register								111
0x0E (0x2E)	USISR	USISIF	USIOIF	USIPF	USIDC	USICN3	USICN2	USICN1	USICN0	112
0x0D (0x2D)	USICR	USISIE	USIOIE	USIWM1	USIWM0	USICS1	USICS0	USICKL	USITC	112
0x0C (0x2C)	TIMSK1	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	101
0x0B (0x2B)	TIFR1	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	102
0x0A (0x2A)	Reserved	–								
0x09 (0x29)	Reserved	–								
0x08 (0x28)	ACSR	ACD	ACBG	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	116
0x07 (0x27)	ADMUX	REFS1	REFS0	MUX5	MUX4	MUX3	MUX2	MUX1	MUX0	129
0x06 (0x26)	ADCSRA	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0	132
0x05 (0x25)	ADCH	ADC data register high byte								133
0x04 (0x24)	ADCL	ADC data register low byte								133
0x03 (0x23)	ADCSRB	BIN	ACME	–	ADLAR	–	ADTS2	ADTS1	ADTS0	133
0x02 (0x22)	Reserved	–								
0x01 (0x21)	DIDR0	ADC7D	ADC6D	ADC5D	ADC4D	ADC3D	ADC2D	ADC1D	ADC0D	117,134
0x00 (0x20)	PRR	–	–	–	–	PRTIM1	PRTIM0	PRUSI	PRADC	32

- Notes:
1. For compatibility with future devices, reserved bits should be written to logical zero if accessed. Reserved I/O memory addresses should never be written.
 2. I/O registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions.
 3. Some of the status flags are cleared by writing a logical one to them. Note that, unlike most other AVR®, the CBI and SBI instructions will only operation the specified bit, and can therefore be used on registers containing such status flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

25. 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	Rdl, 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	Rdl, K	Subtract immediate from word	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND registers	$Rd \leftarrow Rd \times Rr$	Z,N,V	1
ANDI	Rd, K	Logical AND register and constant	$Rd \leftarrow Rd \times K$	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	$Rd \leftarrow Rd \times (0xFF - K)$	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	$Rd \leftarrow Rd \times Rd$	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

25. Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BREQ	k	Branch if equal	if (Z = 1) then PC ← PC + k + 1	None	1/2
BRNE	k	Branch if not equal	if (Z = 0) then PC ← PC + k + 1	None	1/2
BRCS	k	Branch if carry set	if (C = 1) then PC ← PC + k + 1	None	1/2
BRCC	k	Branch if carry cleared	if (C = 0) then PC ← PC + k + 1	None	1/2
BRSH	k	Branch if same or higher	if (C = 0) then PC ← PC + k + 1	None	1/2
BRLO	k	Branch if lower	if (C = 1) then PC ← PC + k + 1	None	1/2
BRMI	k	Branch if minus	if (N = 1) then PC ← PC + k + 1	None	1/2
BRPL	k	Branch if plus	if (N = 0) then PC ← PC + k + 1	None	1/2
BRGE	k	Branch if greater or equal, signed	if (N ⊕ V = 0) then PC ← PC + k + 1	None	1/2
BRLT	k	Branch if less than zero, signed	if (N ⊕ V = 1) then PC ← PC + k + 1	None	1/2
BRHS	k	Branch if half carry flag set	if (H = 1) then PC ← PC + k + 1	None	1/2
BRHC	k	Branch if half carry flag cleared	if (H = 0) then PC ← PC + k + 1	None	1/2
BRTS	k	Branch if T flag set	if (T = 1) then PC ← PC + k + 1	None	1/2
BRTC	k	Branch if T flag cleared	if (T = 0) then PC ← PC + k + 1	None	1/2
BRVS	k	Branch if overflow flag is set	if (V = 1) then PC ← PC + k + 1	None	1/2
BRVC	k	Branch if overflow flag is cleared	if (V = 0) then PC ← PC + k + 1	None	1/2
BRIE	k	Branch if interrupt enabled	if (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	Branch if interrupt disabled	if (I = 0) then PC ← PC + k + 1	None	1/2
Bit and Bit-Test Instructions					
SBI	P,b	Set bit in I/O register	I/O (P, b) ← 1	None	2
CBI	P,b	Clear bit in I/O register	I/O (P, b) ← 0	None	2
LSL	Rd	Logical shift left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical shift right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate left through carry	Rd(0) ← C, Rd(n+1) ← Rd(n), C ← Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate right through carry	Rd(7) ← C, Rd(n) ← Rd(n+1), C ← Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic shift right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap nibbles	Rd(3..0) ← Rd(7..4), Rd(7..4) ← Rd(3..0)	None	1
BSET	s	Flag set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit store from register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to register	Rd(b) ← T	None	1
SEC		Set carry	C ← 1	C	1
CLC		Clear carry	C ← 0	C	1
SEN		Set negative flag	N ← 1	N	1
CLN		Clear negative flag	N ← 0	N	1
SEZ		Set zero flag	Z ← 1	Z	1
CLZ		Clear zero flag	Z ← 0	Z	1
SEI		Global interrupt enable	I ← 1	I	1
CLI		Global interrupt disable	I ← 0	I	1

25. Instruction Set Summary (Continued)

Mnemonics	Operands	Description	Operation	Flags	#Clocks
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