

# F8 Anhang zum ATmega8A

## Befehlssatz des ATmega8A

(Quelle: Atmel® Datenblatt ATmega8A)

| Mnemonics                                | Operands | Description                              | Operation   | Flags      | #Clocks   |
|--|----------|--|---|------------|-----------|
| <b>ARITHMETIC AND LOGIC INSTRUCTIONS</b> |          |  |   |            |           |
| 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                    | $Rd \leftarrow Rd \cdot Rr$                           | Z,N,V      | 1         |
| ANDI                                     | Rd, K    | Logical AND Register and Constant        | $Rd \leftarrow Rd \cdot K$                            | 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                 | $Rd \leftarrow Rd \cdot (0xFF - K)$                   | 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                   | $Rd \leftarrow Rd \cdot Rd$                           | 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         |
| MUL                                      | Rd, Rr   | Multiply Unsigned                        | $R1:R0 \leftarrow Rd \times Rr$                       | Z,C        | 2         |
| MULS                                     | Rd, Rr   | Multiply Signed                          | $R1:R0 \leftarrow Rd \times Rr$                       | Z,C        | 2         |
| MULSU                                    | Rd, Rr   | Multiply Signed with Unsigned            | $R1:R0 \leftarrow Rd \times Rr$                       | Z,C        | 2         |
| FMUL                                     | Rd, Rr   | Fractional Multiply Unsigned             | $R1:R0 \leftarrow (Rd \times Rr) \ll 1$               | Z,C        | 2         |
| FMULS                                    | Rd, Rr   | Fractional Multiply Signed               | $R1:R0 \leftarrow (Rd \times Rr) \ll 1$               | Z,C        | 2         |
| FMULSU                                   | Rd, Rr   | Fractional Multiply Signed with Unsigned | $R1:R0 \leftarrow (Rd \times Rr) \ll 1$               | Z,C        | 2         |
| <b>BRANCH INSTRUCTIONS</b>               |          |  |   |            |           |
| 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 |
| SBRSC                                    | 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     |

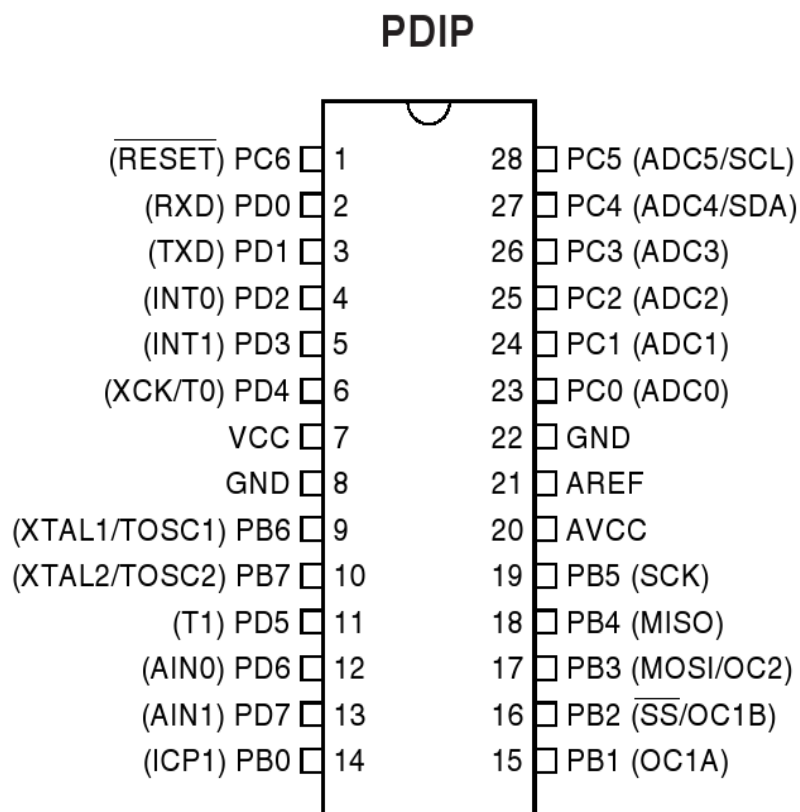
|                                      |         |                                  |  |         |       |
|--------------------------------------|---------|----------------------------------|--|---------|-------|
| BRIE                                 | k       | Branch if Interrupt Enabled      | if ( I = 1) then PC ← PC + k + 1         | None    | 1 / 2 |
| BRID                                 | k       | Branch if Interrupt Disabled     | if ( I = 0) then PC ← PC + k + 1         | None    | 1 / 2 |
| <b>DATA TRANSFER INSTRUCTIONS</b>    |         |                                  |  |         |       |
| MOV                                  | Rd, Rr  | Move Between Registers           | Rd ← Rr                                  | None    | 1     |
| MOVW                                 | Rd, Rr  | Copy Register Word               | Rd+1:Rd ← Rr+1:Rr                        | None    | 1     |
| LDI                                  | Rd, K   | Load Immediate                   | Rd ← K                                   | None    | 1     |
| LD                                   | Rd, X   | Load Indirect                    | Rd ← (X)                                 | None    | 2     |
| LD                                   | Rd, X+  | Load Indirect and Post-Inc.      | Rd ← (X), X ← X + 1                      | None    | 2     |
| LD                                   | Rd, -X  | Load Indirect and Pre-Dec.       | X ← X - 1, Rd ← (X)                      | None    | 2     |
| LD                                   | Rd, Y   | Load Indirect                    | Rd ← (Y)                                 | None    | 2     |
| LD                                   | Rd, Y+  | Load Indirect and Post-Inc.      | Rd ← (Y), Y ← Y + 1                      | None    | 2     |
| LD                                   | Rd, -Y  | Load Indirect and Pre-Dec.       | Y ← Y - 1, Rd ← (Y)                      | None    | 2     |
| LDD                                  | Rd, Y+q | Load Indirect with Displacement  | Rd ← (Y + q)                             | None    | 2     |
| LD                                   | Rd, Z   | Load Indirect                    | Rd ← (Z)                                 | None    | 2     |
| LD                                   | Rd, Z+  | Load Indirect and Post-Inc.      | Rd ← (Z), Z ← Z+1                        | None    | 2     |
| LD                                   | Rd, -Z  | Load Indirect and Pre-Dec.       | Z ← Z - 1, Rd ← (Z)                      | None    | 2     |
| LDD                                  | Rd, Z+q | Load Indirect with Displacement  | Rd ← (Z + q)                             | None    | 2     |
| LDS                                  | Rd, k   | Load Direct from SRAM            | Rd ← (k)                                 | None    | 2     |
| ST                                   | X, Rr   | Store Indirect                   | (X) ← Rr                                 | None    | 2     |
| ST                                   | X+, Rr  | Store Indirect and Post-Inc.     | (X) ← Rr, X ← X + 1                      | None    | 2     |
| ST                                   | -X, Rr  | Store Indirect and Pre-Dec.      | X ← X - 1, (X) ← Rr                      | None    | 2     |
| ST                                   | Y, Rr   | Store Indirect                   | (Y) ← Rr                                 | None    | 2     |
| ST                                   | Y+, Rr  | Store Indirect and Post-Inc.     | (Y) ← Rr, Y ← Y + 1                      | None    | 2     |
| ST                                   | -Y, Rr  | Store Indirect and Pre-Dec.      | Y ← Y - 1, (Y) ← Rr                      | None    | 2     |
| STD                                  | Y+q, Rr | Store Indirect with Displacement | (Y + q) ← Rr                             | None    | 2     |
| ST                                   | Z, Rr   | Store Indirect                   | (Z) ← Rr                                 | None    | 2     |
| ST                                   | Z+, Rr  | Store Indirect and Post-Inc.     | (Z) ← Rr, Z ← Z + 1                      | None    | 2     |
| ST                                   | -Z, Rr  | Store Indirect and Pre-Dec.      | Z ← Z - 1, (Z) ← Rr                      | None    | 2     |
| STD                                  | Z+q, Rr | Store Indirect with Displacement | (Z + q) ← Rr                             | None    | 2     |
| STS                                  | k, Rr   | Store Direct to SRAM             | (k) ← Rr                                 | None    | 2     |
| LPM                                  |         | Load Program Memory              | R0 ← (Z)                                 | None    | 3     |
| LPM                                  | Rd, Z   | Load Program Memory              | Rd ← (Z)                                 | None    | 3     |
| LPM                                  | Rd, Z+  | Load Program Memory and Post-Inc | Rd ← (Z), Z ← Z+1                        | None    | 3     |
| SPM                                  |         | Store Program Memory             | (Z) ← R1:R0                              | None    | -     |
| IN                                   | Rd, P   | In Port                          | Rd ← P                                   | None    | 1     |
| OUT                                  | P, Rr   | Out Port                         | P ← Rr                                   | None    | 1     |
| PUSH                                 | Rr      | Push Register on Stack           | STACK ← Rr                               | None    | 2     |
| POP                                  | Rd      | Pop Register from Stack          | Rd ← STACK                               | None    | 2     |
| <b>BIT AND BIT-TEST INSTRUCTIONS</b> |         |                                  |  |         |       |
| 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     |
| SES                                  |         | Set Signed Test Flag             | S ← 1                                    | S       | 1     |
| CLS                                  |         | Clear Signed Test Flag           | S ← 0                                    | S       | 1     |
| SEV                                  |         | Set Twos Complement Overflow.    | V ← 1                                    | V       | 1     |
| CLV                                  |         | Clear Twos Complement Overflow   | V ← 0                                    | V       | 1     |
| SET                                  |         | Set T in SREG                    | T ← 1                                    | T       | 1     |
| CLT                                  |         | Clear T in SREG                  | T ← 0                                    | T       | 1     |
| SEH                                  |         | Set Half Carry Flag in SREG      | H ← 1                                    | H       | 1     |
| CLH                                  |         | Clear Half Carry Flag in SREG    | H ← 0                                    | H       | 1     |
| <b>MCU CONTROL INSTRUCTIONS</b>      |         |                                  |  |         |       |
| 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     |

Im Befehlssatz werden für Operanden folgende Abkürzungen verwendet:

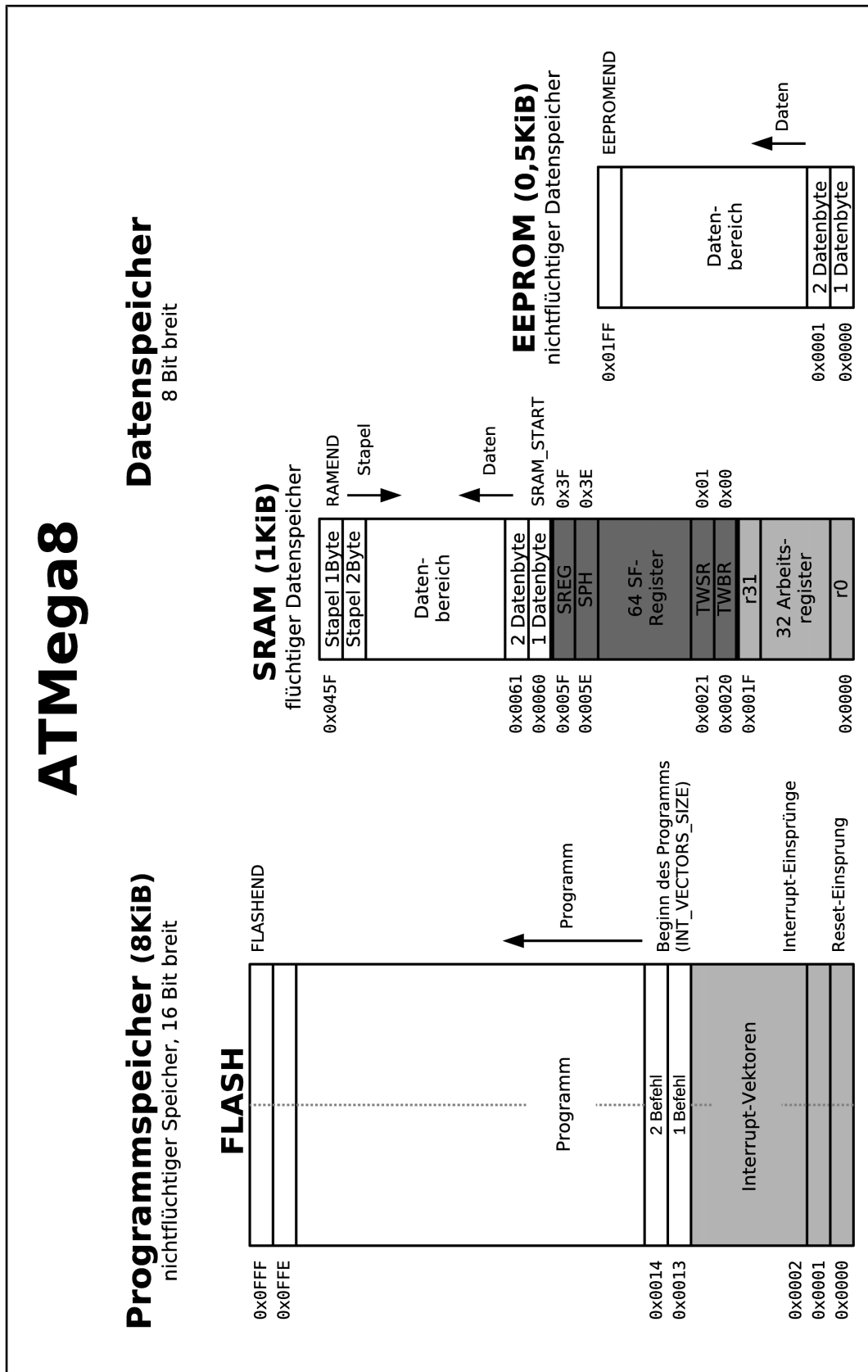
|                       |   |
|-----------------------|---|
| <b>Rd</b>             | Meist Ziel-Arbeitsregister (destination, bei einigen Befehlen auch Quelle)<br><b>r0-r31</b> (bei immediate Befehlen nur <b>r16-r31</b> )  |
| <b>Rd<sub>L</sub></b> | Niederwertiges Byte (LByte) eines 16 Bit Ziel-Arbeitsregister   |
| <b>Rr</b>             | Quell- oder Sende-Arbeitsregister ( <b>r0-r31</b> , source)   |
| <b>K</b>              | Daten-Konstante 8 Bit (0-255)   |
| <b>k</b>              | Adress-Konstante für Operationen mit dem "program counter PC".<br>(Bsp.: Label für einen Sprung)  |
| <b>b</b>              | Bitkonstante 3 Bit (0-7), zum Auswählen eines Bits in einem Arbeits- oder SF-Registers  |
| <b>P</b>              | Adresse eines SF-Registers 6 Bit (0-63)   |
| <b>s</b>              | Bitkonstante 3 Bit (0-7), zum Auswählen eines Bits im Statusregister  |
| <b>X, Y, Z</b>        | Doppelregister (Pointer, Adresszeiger) zur direkten Adressierung<br><b>X</b> $\Rightarrow$ <b>r27:r26</b> ; <b>Y</b> $\Rightarrow$ <b>r29:r28</b> , <b>Z</b> $\Rightarrow$ <b>r31:r30</b> |

## Pinbelegung des ATmega8A

(Quelle: Atmel® Datenblatt ATmega8A)



# Speicherorganisation des ATmega8A





## Interrupt-Vektortabelle des ATmega8A

| Vektor-nummer | Adresse im Flash | Name in "m32def.inc" | Quelle des Interrupts | Verantwortlich für den Interrupt   |
|---------------|------------------|----------------------|-----------------------|--|
| 19            | 0x0024           | SPMRaddr             | SPM_RDY               | Programmspeicher (Flash)-Programmierung fertig   |
| 18            | 0x0022           | TWIaddr              | TWI                   | I <sup>2</sup> C-Interface (TWI-Interface)   |
| 17            | 0x0020           | ACIaddr              | ANA_COMP              | Analog-Komparator  |
| 16            | 0x001E           | ERDYaddr             | EE_RDY                | EEPROM-Programmierung (schreiben) fertig   |
| 15            | 0x001C           | ADCCaddr             | ADC                   | AD-Wandlung vollständig  |
| 14            | 0x001A           | UTXCaddr             | USART, TXC            | USART, Zeichen gesendet (Senderegister leer)   |
| 13            | 0x0018           | UDREaddr             | USART, UDRE           | USART, UDR-Register leer   |
| 12            | 0x0016           | URXCaddr             | USART, RXC            | USART, Empfangsregister voll   |
| 11            | 0x0014           | SPIaddr              | SPI, STC              | Serielle Übertragung beendet (SPI)   |
| 10            | 0x0012           | OVF0addr             | TIMER0 OVF            | Overflow von Timer 0   |
| 9             | 0x0010           | OVF1addr             | TIMER1 OVF            | Overflow von Timer 1   |
| 8             | 0x000E           | OC1Baddr             | TIMER1 COMPB          | Compare Match B von Timer 1  |
| 7             | 0x000C           | OC1Aaddr             | TIMER1 COMPA          | Compare Match A von Timer 1  |
| 6             | 0x000A           | ICP1addr             | TIMER1 CAPT           | Capture Event von Timer 2  |
| 5             | 0x0008           | OVF2addr             | TIMER2 OVF            | Overflow Match von Timer 2   |
| 4             | 0x0006           | OC2addr              | TIMER2 COMP           | Compare Match von Timer 2  |
| 3             | 0x0004           | INT1addr             | INT1                  | Interrupt-Eingang 1 (externer PIN PD3)   |
| 2             | 0x0002           | INT0addr             | INT0                  | Interrupt-Eingang 0 (externer PIN PD2)   |
| 1             | 0x0000           |                      | RESET                 | RESET Pin (extern Pin 9), Power-On-Reset, Brown-Out-Reset, Watchdog Reset und JTAG AVR-Reset |

**Bemerkungen:** Das eigentliche Programm kann erst an der Adresse **0x0026** beginnen. Der in der Definitionsdatei vorgesehene Name für diese Adresse heißt: **INT\_VECTORS\_SIZE**.

Wenn das **BOOTRST**-Fuse-Bit programmiert wurde springt der Controller nach einem **RESET** automatisch in den Bootbereich (Bootloader-Programm) im oberen Adressbereich des Flash-Speichers. Mit Hilfe des **IVSEL**-Bit im SF-Register **GICR** kann die Interrupt-Vektortabelle in den Anfang des Bootbereichs verlegt werden.

# SF-Registersatz des ATmega8A

| 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      | 8                |     |
| 0x3E (0x5E)                               | SPH      | –  | –      | –      | –      | –          | SP10   | SP9    | SP8    | 11               |     |
| 0x3D (0x5D)                               | SPL      | SP7  | SP6    | SP5    | SP4    | SP3        | SP2    | SP1    | SP0    | 11               |     |
| 0x3C (0x5C)                               | Reserved |  |        |        |        |            |        |        |        |                  |     |
| 0x3B (0x5B)                               | GICR     | INT1   | INT0   | –      | –      | –          | –      | IVSEL  | IVCE   | 48, 68           |     |
| 0x3A (0x5A)                               | GIFR     | INTF1  | INTF0  | –      | –      | –          | –      | –      | –      | 69               |     |
| 0x39 (0x59)                               | TIMSK    | OCIE2  | TOIE2  | TICIE1 | OCIE1A | OCIE1B     | TOIE1  | –      | TOIE0  | 73, 104, 124     |     |
| 0x38 (0x58)                               | TIFR     | OCF2   | TOV2   | ICF1   | OCF1A  | OCF1B      | TOV1   | –      | TOV0   | 74, 104, 104     |     |
| 0x37 (0x57)                               | SPMCR    | SPMIE  | RWWSB  | –      | RWWSRE | BLBSET     | PGWRT  | PGERS  | SPMEN  | 224              |     |
| 0x36 (0x56)                               | TWCR     | TWINT  | TWEA   | TWSTA  | TWSTO  | TWWC       | TWEN   | –      | TWIE   | 191              |     |
| 0x35 (0x55)                               | MCUCR    | SE   | SM2    | SM1    | SM0    | ISC11      | ISC10  | ISC01  | ISC00  | 36, 67           |     |
| 0x34 (0x54)                               | MCUCSR   | –  | –      | –      | –      | WDRF       | BORF   | EXTRF  | PORF   | 43               |     |
| 0x33 (0x53)                               | TCCR0    | –  | –      | –      | –      | –          | CS02   | CS01   | CS00   | 73               |     |
| 0x32 (0x52)                               | TCNT0    | Timer/Counter0 (8 Bits)                              |        |        |        |            |        |        |        | 73               |     |
| 0x31 (0x51)                               | OSCCAL   | Oscillator Calibration Register                      |        |        |        |            |        |        |        | 31               |     |
| 0x30 (0x50)                               | SFIOR    | –  | –      | –      | –      | ACME       | PUD    | PSR2   | PSR10  | 57, 77, 125, 196 |     |
| 0x2F (0x4F)                               | TCCR1A   | COM1A1   | COM1A0 | COM1B1 | COM1B0 | FOC1A      | FOC1B  | WGM11  | WGM10  | 99               |     |
| 0x2E (0x4E)                               | TCCR1B   | ICNC1  | ICES1  | –      | WGM13  | WGM12      | CS12   | CS11   | CS10   | 101              |     |
| 0x2D (0x4D)                               | TCNT1H   | Timer/Counter1 – Counter Register High byte          |        |        |        |            |        |        |        | 102              |     |
| 0x2C (0x4C)                               | TCNT1L   | Timer/Counter1 – Counter Register Low byte           |        |        |        |            |        |        |        | 102              |     |
| 0x2B (0x4B)                               | OCR1AH   | Timer/Counter1 – Output Compare Register A High byte |        |        |        |            |        |        |        | 103              |     |
| 0x2A (0x4A)                               | OCR1AL   | Timer/Counter1 – Output Compare Register A Low byte  |        |        |        |            |        |        |        | 103              |     |
| 0x29 (0x49)                               | OCR1BH   | Timer/Counter1 – Output Compare Register B High byte |        |        |        |            |        |        |        | 103              |     |
| 0x28 (0x48)                               | OCR1BL   | Timer/Counter1 – Output Compare Register B Low byte  |        |        |        |            |        |        |        | 103              |     |
| 0x27 (0x47)                               | ICR1H    | Timer/Counter1 – Input Capture Register High byte    |        |        |        |            |        |        |        | 103              |     |
| 0x26 (0x46)                               | ICR1L    | Timer/Counter1 – Input Capture Register Low byte     |        |        |        |            |        |        |        | 103              |     |
| 0x25 (0x45)                               | TCCR2    | FOC2   | WGM20  | COM21  | COM20  | WGM21      | CS22   | CS21   | CS20   | 121              |     |
| 0x24 (0x44)                               | TCNT2    | Timer/Counter2 (8 Bits)                              |        |        |        |            |        |        |        | 123              |     |
| 0x23 (0x43)                               | OCR2     | Timer/Counter2 Output Compare Register               |        |        |        |            |        |        |        | 123              |     |
| 0x22 (0x42)                               | ASSR     | –  | –      | –      | –      | AS2        | TCN2UB | OCR2UB | TCR2UB | 123              |     |
| 0x21 (0x41)                               | WDTCSR   | –  | –      | –      | WDCE   | WDE        | WDP2   | WDP1   | WDP0   | 43               |     |
| 0x20 <sup>(1)</sup> (0x40) <sup>(1)</sup> | UBRRH    | URSEL  | –      | –      | –      | UBRR[11:8] |        |        |        |                  | 160 |
|   | UCSRC    | URSEL  | UMSEL  | UPM1   | UPM0   | USBS       | UCSZ1  | UCSZ0  | UCPOL  | 159              |     |
| 0x1F (0x3F)                               | EEARH    | –  | –      | –      | –      | –          | –      | –      | EEAR8  | 19               |     |
| 0x1E (0x3E)                               | EEARL    | EEAR7  | EEAR6  | EEAR5  | EEAR4  | EEAR3      | EEAR2  | EEAR1  | EEAR0  | 19               |     |
| 0x1D (0x3D)                               | EEDR     | EEPROM Data Register                                 |        |        |        |            |        |        |        | 19               |     |
| 0x1C (0x3C)                               | EECR     | –  | –      | –      | –      | EERIE      | EEMWE  | EEWE   | EERE   | 19               |     |
| 0x1B (0x3B)                               | Reserved |  |        |        |        |            |        |        |        |                  |     |
| 0x1A (0x3A)                               | Reserved |  |        |        |        |            |        |        |        |                  |     |
| 0x19 (0x39)                               | Reserved |  |        |        |        |            |        |        |        |                  |     |
| 0x18 (0x38)                               | PORTB    | PORTB7   | PORTB6 | PORTB5 | PORTB4 | PORTB3     | PORTB2 | PORTB1 | PORTB0 | 65               |     |
| 0x17 (0x37)                               | DDRB     | DDB7   | DDB6   | DDB5   | DDB4   | DDB3       | DDB2   | DDB1   | DDB0   | 65               |     |
| 0x16 (0x36)                               | PINB     | PINB7  | PINB6  | PINB5  | PINB4  | PINB3      | PINB2  | PINB1  | PINB0  | 65               |     |
| 0x15 (0x35)                               | PORTC    | –  | PORTC6 | PORTC5 | PORTC4 | PORTC3     | PORTC2 | PORTC1 | PORTC0 | 65               |     |
| 0x14 (0x34)                               | DDRC     | –  | DDC6   | DDC5   | DDC4   | DDC3       | DDC2   | DDC1   | DDC0   | 65               |     |
| 0x13 (0x33)                               | PINC     | –  | PINC6  | PINC5  | PINC4  | PINC3      | PINC2  | PINC1  | PINC0  | 65               |     |
| 0x12 (0x32)                               | PORTD    | PORTD7   | PORTD6 | PORTD5 | PORTD4 | PORTD3     | PORTD2 | PORTD1 | PORTD0 | 65               |     |
| 0x11 (0x31)                               | DDRD     | DDD7   | DDD6   | DDD5   | DDD4   | DDD3       | DDD2   | DDD1   | DDD0   | 65               |     |
| 0x10 (0x30)                               | PIND     | PIND7  | PIND6  | PIND5  | PIND4  | PIND3      | PIND2  | PIND1  | PIND0  | 66               |     |
| 0x0F (0x2F)                               | SPDR     | SPI Data Register                                    |        |        |        |            |        |        |        | 135              |     |
| 0x0E (0x2E)                               | SPSR     | SPIF   | WCOL   | –      | –      | –          | –      | –      | SPI2X  | 134              |     |
| 0x0D (0x2D)                               | SPCR     | SPIE   | SPE    | DORD   | MSTR   | CPOL       | CPHA   | SPR1   | SPR0   | 133              |     |
| 0x0C (0x2C)                               | UDR      | USART I/O Data Register                              |        |        |        |            |        |        |        | 156              |     |
| 0x0B (0x2B)                               | UCSRA    | RXC  | TXC    | UDRE   | FE     | DOR        | PE     | U2X    | MPCM   | 157              |     |
| 0x0A (0x2A)                               | UCSRB    | RXCIE  | TXCIE  | UDRIE  | RXEN   | TXEN       | UCSZ2  | RXB8   | TXB8   | 158              |     |
| 0x09 (0x29)                               | UBRRL    | USART Baud Rate Register Low byte                    |        |        |        |            |        |        |        | 160              |     |
| 0x08 (0x28)                               | ACSR     | ACD  | ACBG   | ACO    | ACI    | ACIE       | ACIC   | ACIS1  | ACIS0  | 196              |     |
| 0x07 (0x27)                               | ADMUX    | REFS1  | REFS0  | ADLAR  | –      | MUX3       | MUX2   | MUX1   | MUX0   | 208              |     |
| 0x06 (0x26)                               | ADCSRA   | ADEN   | ADSC   | ADFR   | ADIF   | ADIE       | ADPS2  | ADPS1  | ADPS0  | 209              |     |
| 0x05 (0x25)                               | ADCH     | ADC Data Register High byte                          |        |        |        |            |        |        |        | 210              |     |
| 0x04 (0x24)                               | ADCL     | ADC Data Register Low byte                           |        |        |        |            |        |        |        | 210              |     |
| 0x03 (0x23)                               | TWDR     | Two-wire Serial Interface Data Register              |        |        |        |            |        |        |        | 193              |     |
| 0x02 (0x22)                               | TWAR     | TWA6   | TWA5   | TWA4   | TWA3   | TWA2       | TWA1   | TWA0   | TWGCE  | 194              |     |
| 0x01 (0x21)                               | TWSR     | TWS7   | TWS6   | TWS5   | TWS4   | TWS3       | –      | TWPS1  | TWPS0  | 193              |     |
| 0x00 (0x20)                               | TWBR     | Two-wire Serial Interface Bit Rate Register          |        |        |        |            |        |        |        | 191              |     |

- Note:
1. Refer to the USART description for details on how to access UBRRH and UCSRC.
  2. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
  3. Some of the Status Flags are cleared by writing a logical one to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

## Blockschaltbild des ATmega8A

(Quelle: Atmel® Datenblatt ATmega8A)

