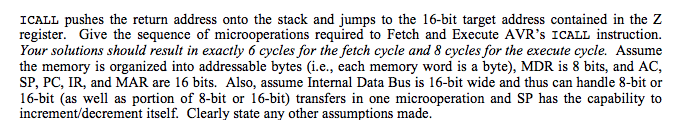
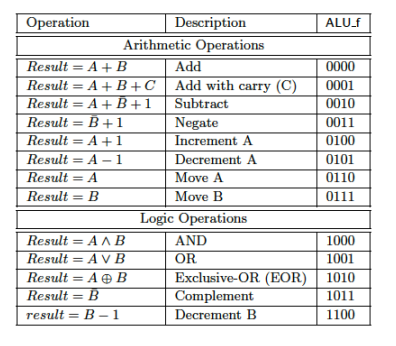
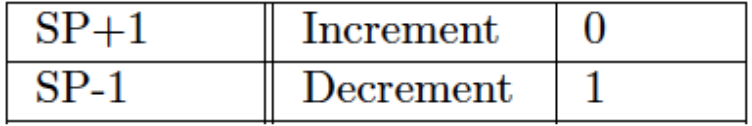
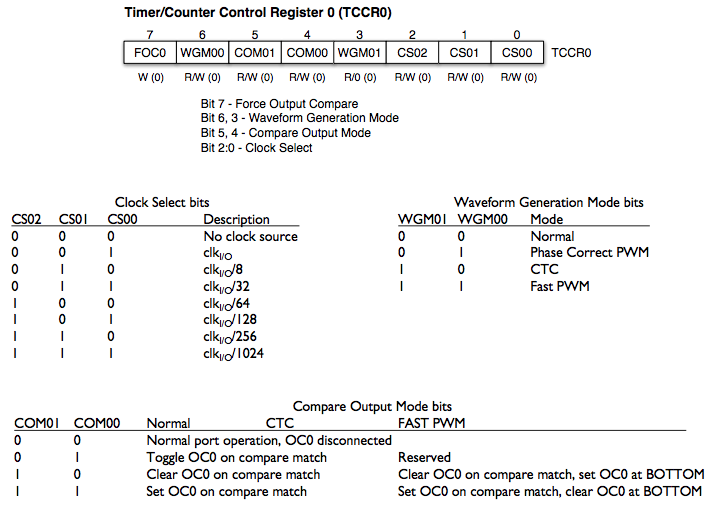
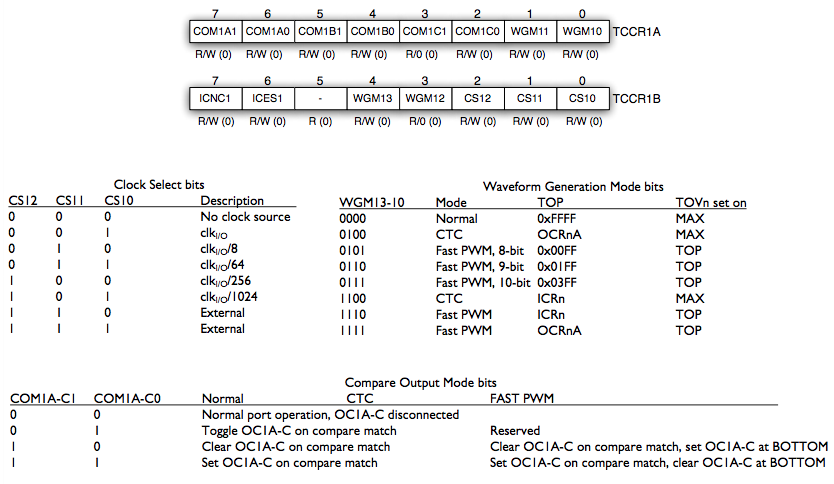
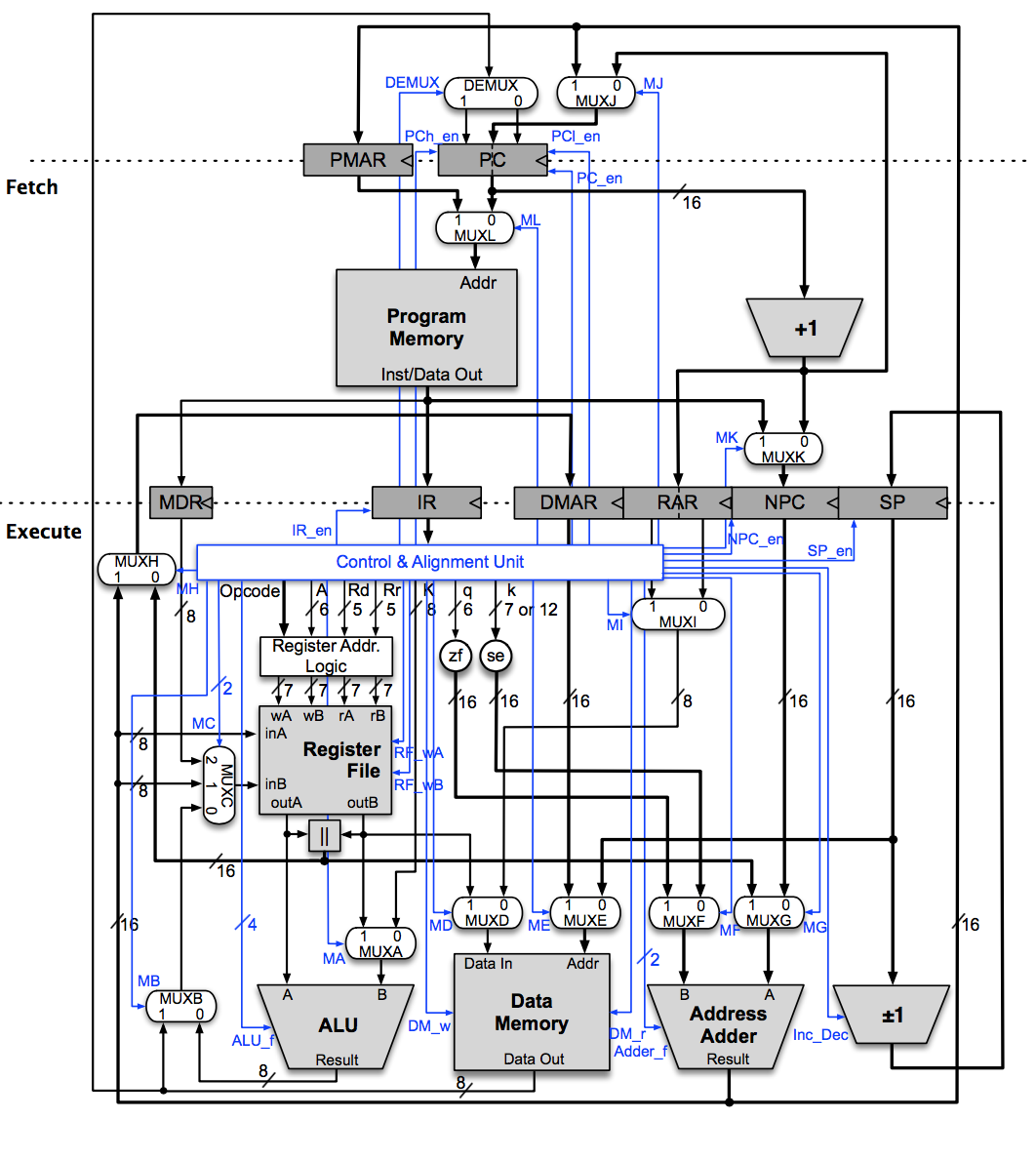


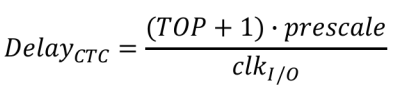
|  |  |  |
| --- | --- | --- |
| [ADC - Add with Carry](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_ADC.html)  [ADD - Add without Carry](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_ADD.html)  [ADIW - Add Immediate to Word](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_ADIW.html)  [AND - Logical AND](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_AND.html)  [ANDI - Logical AND with Immediateand](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_ANDI.html)  [ASR - Arithmetic Shift Right](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_ASR.html)  [BCLR - Bit Clear in SREG](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BCLR.html)  [BLD - Bit Load from the T Flag in SREG to a Bit in Register.](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BLD.html)  [BRBC - Branch if Bit in SREG is Cleared](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRBC.html)  [BRBS - Branch if Bit in SREG is Set](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRBS.html)  [BRCC - Branch if Carry Cleared](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRCC.html)  [BRCS - Branch if Carry Set](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRCS.html)  [BREAK - Break](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BREAK.html)  [BREQ - Branch if Equal](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BREQ.html)  [BRGE - Branch if Greater or Equal Signed)](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRGE.html)  [BRHC - Branch if Half Carry Flag is Cleared](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRHC.html)  [BRHS - Branch if Half Carry Flag is Set](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRHS.html)  [BRID - Branch if Global Interrupt is Disabled](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRID.html)  [BRIE - Branch if Global Interrupt is Enabled](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRIE.html)  [BRLO - Branch if Lower (Unsigned)](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRLO.html)  [BRLT - Branch if Less Than (Signed)](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRLT.html)  [BRMI - Branch if Minus](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRMI.html)  [BRNE - Branch if Not Equal](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRNE.html)  [BRPL - Branch if Plus](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRPL.html)  [BRSH - Branch if Same or Higher (Unsigned)](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRSH.html)  [BRTC - if the T Flag is Cleared](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRTC.html)  [BRTS - Branch if the T Flag is Set](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRTS.html)  [BRVC - Branch if Overflow Cleared](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRVC.html)  [BRVS - Branch if Overflow Set](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BRVS.html)  [BSET - Bit Set in SREG](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BSET.html)  [SEH - Set Half Carry Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SEH.html)  [SEI - Set Global Interrupt Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SEI.html)  [SEN - Set Negative Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SEN.html)  [SER - Set all bits in Register](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SER.html)  [SES - Set Signed Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SES.html)  [SET - Set T Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SET.html)  [SEV - Overflow Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SEV.html)  [SEZ - Set Zero Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SEZ.html)  [SLEEP- Sleep mode](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SLEEP.html)  [SPM - Store Program Memory](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SPM.html)  [ST - Store Indirect From Register to data space using Index X](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_ST.html)  [ST (STD) - Store Indirect From Register to data space using Index Y](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_STD.html) | [BST - Bit Store from Bit in Register to T Flag in SREG](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_BST.html)  [CALL - Long Call to a Subroutine](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CALL.html)  [CBI - Clear Bit in I/O Register](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CBI.html)  [CBR - Clear Bits in Register](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CBR.html)  [CLC - Clear Carry Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CLC.html)  [CLH - Clear Half Carry Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CLH.html)  [CLI - Clear Global Interrupt Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CLI.html)  [CLN - Clear Negative Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CLN.html)  [CLR - Clear Register](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CLR.html)  [CLS - Clear Signed Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CLS.html)  [CLT - Clear T Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CLT.html)  [CLV - Clear Overflow Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CLV.html)  [CLZ - Clear Zero Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CLZ.html)  [COM- One's Complement](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_COM.html)  [CP- Compare](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CP.html)  [CPC- Compare with Carry](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CPC.html)  [CPI- Compare with Immediate](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CPI.html)  [CPSE- Compare Skip if Equal](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_CPSE.html)  [DEC- Decrement](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_DEC.html)  [EICALL - Extended Indirect Call to Subroutine](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_EICALL.html)  [EIJMP - Extended Indirect Jump](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_EIJMP.html)  [ELPM - Extended Load Program Memory](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_ELPM.html)  [EOR - Exclusive OR](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_EOR.html)  [FMUL- Fractional Multiply Unsigned](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_FMUL.html)  [FMULS - Fractional Multiply Signed](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_FMULS.html)  [FMULSU - Fractional Multiply Signed with Unsigned](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_FMULSU.html)  [ICALL - Indirect Call to Subroutine](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_ICALL.html)  [IJMP - Indirect Jump](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_IJMP.html)  [IN - Load an I/O Location to Register](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_IN.html)  [INC- Increment](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_INC.html)  [JMP - Jump](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_JMP.html)  [LD - Load Indirect from data space to Register using Index X](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_LD.html)  [LAT - Load and Toggle](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_LAT.html)  [LAS - Load and Set](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_LAS.html)  [ST (STD) - Store Indirect From Register to data space using Index Z](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_STD_Z.html)  [STS - Store Direct to data space](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_STS.html)  [STS - Store Direct to SRAM](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_STS_-_Store_Direct_to_SRAM.html)  [SUB- Subtract without Carry](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SUB.html)  [SUBI- Subtract Immediate](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SUBI.html)  [SWAP - Swap Nibbles](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SWAP.html)  [TST- Test for Zero or Minus](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_TST.html)  [WDR - Watchdog Reset](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_WDR.html)  [XCH - Exchange](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_XCH.html) | [LAC - Load and Clear](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_LAC.html)  [LD (LDD)- Load Indirect from data space to Register using Index Y](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_LDD.html)  [LD (LDD) - Load Indirect From data space to Register using Index Z](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_LDD_Z.html)  [LDI - Load Immediate](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_LDI.html)  [LDS - Load Direct from data space](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_LDS.html)  [LDS - Load Direct from SRAM](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_LDS_-_Load_direct_from_SRAM.html)  [LPM - Load Program Memory](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_LPM.html)  [LSL- Logical Shift Left](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_LSL.html)  [LSR- Logical Shift Right](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_LSR.html)  [MOV - Copy Register](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_MOV.html)  [MOVW - Copy Register Word](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_MOVW.html)  [MUL- Multiply Unsigned](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_MUL.html)  [MULS - Multiply Signed](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_MULS.html)  [MULSU - Multiply Signed with Unsigned](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_MULSU.html)  [NEG- Two's Complement](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_NEG.html)  [NOP - No Operation](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_NOP.html)  [OR- Logical OR](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_OR.html)  [ORI- Logical OR with Immediate](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_ORI.html)  [OUT - Store Register to I/O Location](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_OUT.html)  [POP - Pop Register from Stack](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_POP.html)  [PUSH - Push Register on Stack](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_PUSH.html)  [RCALL - Relative Call to Subroutine](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_RCALL.html)  [RET - Return from Subroutine](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_RET.html)  [RETI - Return from Interrupt](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_RETI.html)  [RJMP- Relative Jump](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_RJMP.html)  [ROL- Rotate Left trough Carry](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_ROL.html)  [ROR- Rotate Right through Carry](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_ROR.html)  [SBC- Subtract with Carry](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SBC.html)  [SBCI- Subtract Immediate with Carry](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SBCI.html)  [SBI - Set Bit in I/O Register](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SBI.html)  [SBIC - Skip if Bit in I/O Register is Cleared](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SBIC.html)  [SBIS - Skip if Bit in I/O Register is Set](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SBIS.html)  [SBIW - Subtract Immediate from Word](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SBIW.html)  [SBR- Set Bits in Register](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SBR.html)  [SBRC - Skip if Bit in Register is Cleared](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SBRC.html)  [SBRS - Skip if Bit in Register is Set](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SBRS.html)  [SEC - Set Carry Flag](http://www.microchip.com/webdoc/avrassembler/avrassembler.wb_SEC.html) |

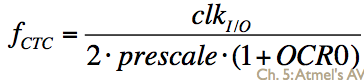


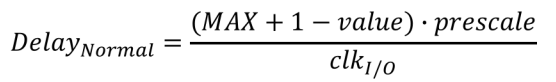


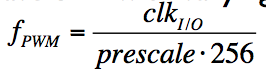


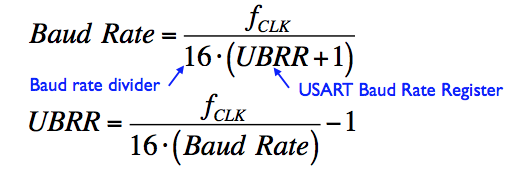


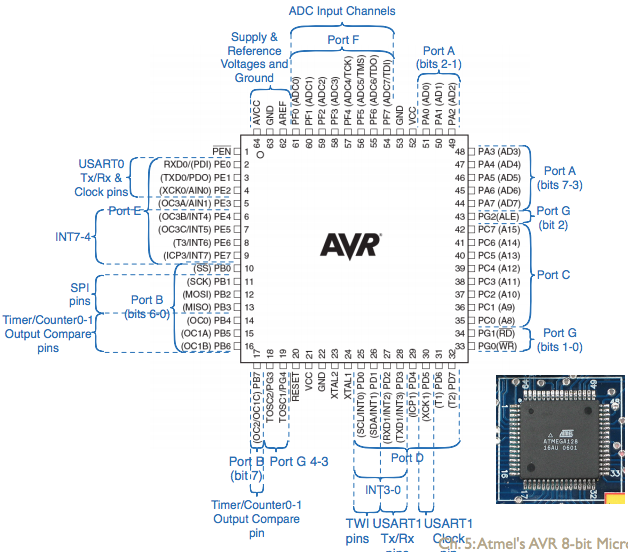


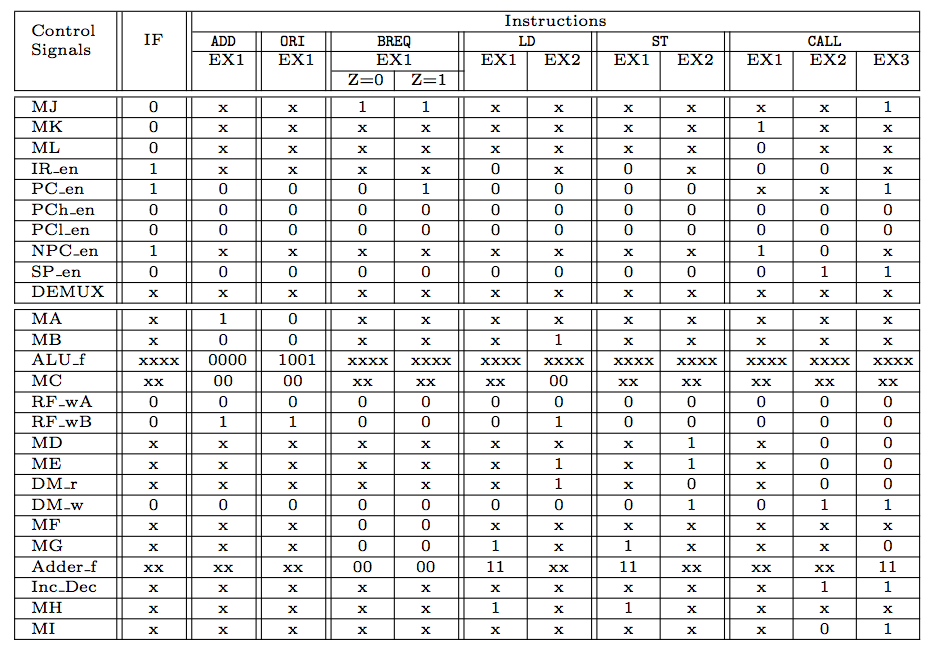


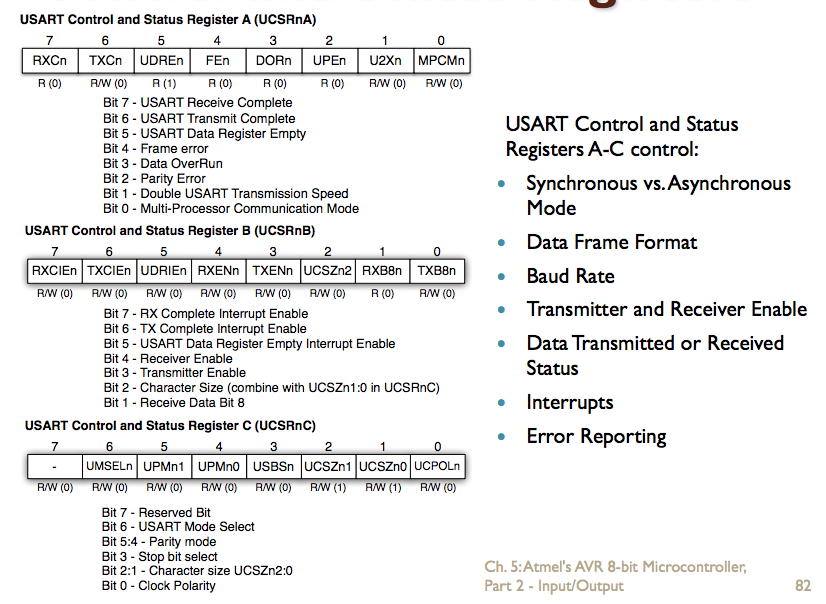
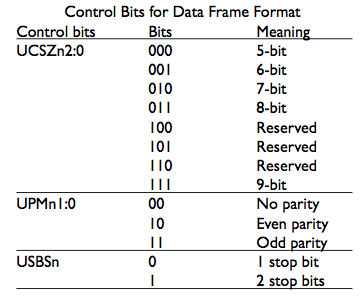


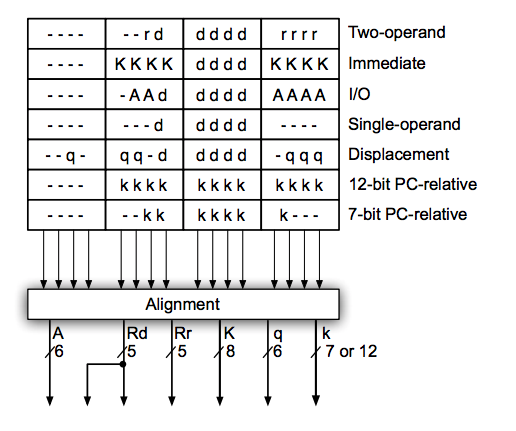


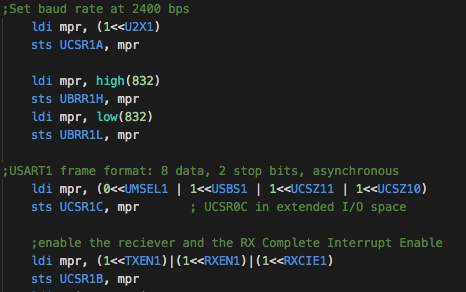
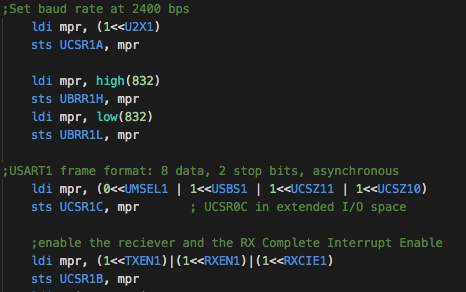
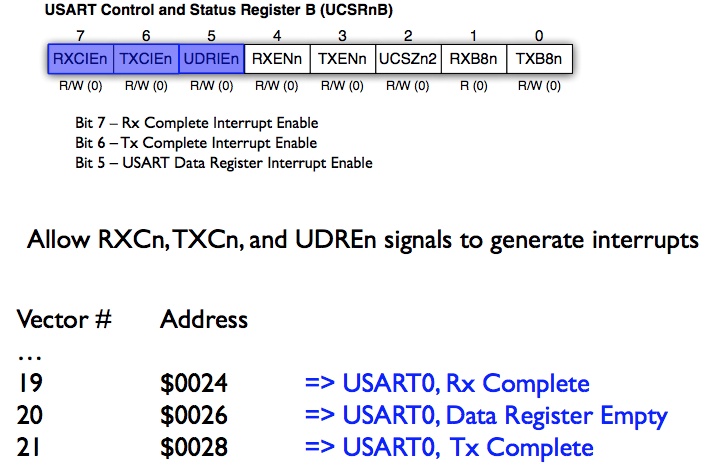










|  |  |
| --- | --- |
| PC = Size of address  MDR = Size of Data memory  AC = Size of Data memory  IR = Size of opcode | DDRF PINF PORTG DDRG PING are in the extended IO space so you have to use LDS and STS to use them  AtMega128 has 6 8-bit IO ports and 1 5-bit |
| LPM can only use Z register | MDR is the only interface to memory(therefore same width) |
| Result of MUL lower byte is in R0 and upper byte is in R1 (even if 00) | RJMP and RCALL = 16-bit. JM and CALL = 32-bit |
| AtMega128: 32 8-bit GPRs. 64 8-bit IO regs. 160 Extended 8-bit IO regs | * X is registers 27:26 * Y is registers 29:28 * Z is registers 31:30 |
| * CALL and JMP are 32-bit instructions * If working in Program Memory (DIR<<1) * Set ZL low(DIR>>1) | * INT7-4 connected to PORTE pins 7-4. * INT3-0 connected to PORTD pins 3-0. |
| CTC Mode   * Load OCR0 with a value. * TCNT0 starts at 0 and counts up to OCR0 (TOP) * OCF0 is set when TCNT0 reaches OCR0 * OCRx is used to control when counting ends. * TCNTx is reset to 0 when TCNTx and OCRx match. * Need to write a 1 to TOVx and OCFx to use them again * OCO is part of a PORT so DDRB must be set to 1 (output) in order to be read from | Normal Mode   * TOV0 is set when TCNT0 reaches MAX * Smaller the prescale, the higher the resolution * Counting stops when TCNTx rolls over   Fast PMW Mode   * Uses both OCF0 and TOV0 * Clear OC0 on output compare match, set OC0 at TOP   Timers   * TOIEx = Timer overflow interrupt enable (Normal mode) * OCIEx = Output compare interrupt enable (CTC mode) * CTC Mode: move value into OCR and 0 into TCNT. Now when TCNT==OCR, OCIE will be set to call the interrupt. IF the count is to be repeated then clear TCNT to 0 and set CTCx bit which clears it on compare automatically * TCNT is compared to OCR. IF match OCF is set. IF 16-bit then OCR1A sets OCF1A * TCCRx is register that customizes timer |
| In PC relative instructions like RCALL and BREQ, you have to find the address by using PC + 1 | * Must set bit in EIMSK to be detected * You know which INT was triggered by bit testing EIFR * 00 = low, 11 = rising, 10 = falling |
| USART   * Start bit is always low * Stop bit is always high * Idle is high * Even Parity means P-bit is set to make number of 1’s even * By setting TXCIE and RXCIE in UCSRxB you can set interrupts. Vectors for 0 = 0x0028 and 0x0024 and for USRART1 = 0x0040 and 0x003C | USART   * UMSELx = 0 = Async, 1 = Sync * UCSZx 2:0 is data frame size * UPMx 1:0 is parity == 00 none, 10 = even, 01 = odd * USBSx = 0 = one stop bit, 1 = two stop bits * TXENx = transmitter enable * UDREx = 0 = UDRx Full, 1 = UDRx Empty * TXCx = 0 = transmit incomplete, 1 = complete * RXCx = 0 = receive incomplete, 1 = complete |
| Effective address (is pointer) routes to the left of Address Adder and Target Address(is raw address) routes to the right | sbr mpr, (1<<WskrR)|(1<<WskrL) ;setting bits in flag register to 1 since pull up resistor  out EIFR, mpr                   ;cancel out any staged interrupts |
|  | ldi mpr, 0b01101111              ;Activate Fast PWM mode with toggle. Done by bit 3 and 6 being 1  out TCCR0, mpr                    ;(non-inverting), and set prescaler to 1024. Normal Mode  ;later out OCR0, newSpeed |
|  | lds mpr, UCSR1A   sbrs mpr, UDRE1    ;If byte is still in UDR, wait till it shifts to transmit reg  ;enable the reciever and the RX Complete Interrupt Enable  ldi mpr, (1<<TXEN1)|(1<<RXEN1)|(1<<RXCIE1)  sts UCSR1B, mpr  ;later lds recvdCmmd, UDR1 |