

8051 Quick Reference

Basic ALU Instructions

- INC operand (0x) - Increment
- DEC operand (1x) - Decrement
- ADD A,operand (2x) - Add
- ADDC A,operand (3x) - Add including the carry bit (C)
- ORL A,operand (4x) - Logical OR
- ANL A,operand (5x) - Logical AND
- XRL A,operand (6x) - Logical Exclusive-OR (XOR)
- MOV operand,#data (7x) - Move immediate data
- MOV address,operand (8x) - Move data to IRAM or SFR
- SUBB A,operand (9x) - Subtraction with borrow
- MOV operand,address (Ax) - Move data from an IRAM or SFR register
- CJNE operand,#data,offset (Bx) - Compare and Jump if Not Equal
- XCH A,operand (Cx) - Exchange (switch) the accumulator and the operand
- DJNZ operand,offset (Dx) - Decrement and Jump if Not Zero
- MOV A,operand (Ex) - Move operand to the accumulator
- MOV operand,A (Fx) - Move accumulator to the operand

Irregular Instructions

- 0x00 (NOP) - No Operation
- 0x01 (AJMP page0) - Address Jump
- 0x02 (LJMP addr16) - Long Jump
- 0x03 (RR A) - Rotate Right
- 0x04 (INC A) - Increment
- 0x10 (JBC bit,offset) - Jump if Bit is set and then Clear the bit
- 0x11 (ACALL page0) - Absolute Call
- 0x12 (LCALL addr16) - Long Call
- 0x13 (RRC A) - Rotate Right with Carry (>A C>); a rotation right where the value in A.7 becomes CY and CY becomes A.0
- 0x14 (DEC A) - Decrement
- 0x20 (JB bit,offset) - Jump if Bit set; jump if bit is one

- 0x21 (AJMP page1) - Address Jump
- 0x22 (RET) - Return; pops the high-order and low-order bytes of PC from the stack and decrements SP by two
- 0x23 (RL A) - Rotate Left
- 0x24 (ADD A,#data) - Addition
- 0x30 (JNB bit,offset) - Jump if Bit Not set; jump if bit is zero
- 0x31 (ACALL page1) - Absolute Call
- 0x32 (RETI) - Return from Interrupt
- 0x33 (RLC A) - Rotate Left with Carry (
- 0x34 (ADDC A,#data) - Add including the carry bit (C)
- 0x40 (JC offset) - Jump if the Carry bit is set (CY is "1")
- 0x41 (AJMP page2) - Address Jump
- 0x42 (ORL address,A) - Logical OR
- 0x43 (ORL address,#data) - Logical OR
- 0x44 (ORL A,#data) - Logical OR
- 0x50 (JNC offset) - Jump if Carry Bit is Not set (CY is "0")
- 0x51 (ACALL page2) - Absolute Call
- 0x52 (ANL address,A) - Logical AND
- 0x53 (ANL address,#data) - Logical AND
- 0x54 (ANL A,#data) - Logical AND
- 0x60 (JZ offset) - Jump if Zero
- 0x61 (AJMP page3) - Address Jump
- 0x62 (XRL address,A) - Logical XOR
- 0x63 (XRL address,#data) - Logical XOR
- 0x64 (XRL A,#data) - Logical XOR
- 0x70 (JNZ offset) - Jump if Not Zero
- 0x71 (ACALL page3) - Absolute Call
- 0x72 (ORL C,bit) - Logical OR
- 0x73 (JMP @A+DPTR) - Jump
- 0x74 (MOV A,#data) - Move
- 0x80 (SJMP offset) - Short Jump

- 0x81 (AJMP page4) - Address Jump
- 0x82 (ANL C,bit) - Logical AND
- 0x83 (MOVC A,@A+PC) - Move a byte from code or program memory
- 0x84 (DIV AB) - Division/Divide
- 0x90 (MOV DPTR, #imm16) - Move
- 0x91 (ACALL page4) - Absolute Call
- 0x92 (MOV bit,C) - Move the value of C to bit
- 0x93 (MOVC A,@A+DPTR) - Move a byte from code or program memory
- 0x94 (SUBB A,#data) - Subtraction with borrow
- 0xA0 (ORL C,bit) - Logical OR
- 0xA1 (AJMP page5) - Address Jump
- 0xA2 (MOV C,bit) - Move the value of bit to C
- 0xA3 (INC DPTR) - Increment
- 0xA4 (MUL AB) - Multiply
- 0xB0 (ANL C,bit) - Logical AND
- 0xB1 (ACALL page5) - Absolute Call
- 0xB2 (CPL bit) - Logical Complement; Logical NOT; 0 becomes 1 and vice versa
- 0xB3 (CPL C) - Logical Complement; Logical NOT; 0 becomes 1 and vice versa
- 0xB4 (CJNE A,#data,offset) - Compare and Jump if Not Equal
- 0xB5 (CJNE A,address,offset) - Compare and Jump if Not Equal
- 0xC0 (PUSH address) - Increments SP
- 0xC1 (AJMP page6) - Address Jump
- 0xC2 (CLR bit) - Clear the specified bit (set to "0")
- 0xC3 (CLR C) - Clear the carry bit (set to "0")
- 0xC4 (SWAP A) - Swap the low and high nibble
- 0xD0 (POP address) - Decrements SP
- 0xD1 (ACALL page6) - Absolute Call
- 0xD2 (SETB bit) - Set Bit to "1"; opposite of CLR bit
- 0xD3 (SETB C) - Set Bit to "1"; opposite of CLR C

8051 Quick Reference

- 0xD4 (DA A) - Decimal Adjust
- 0xD6 (XCHD A,@R0) - Exchange least-significant nibble
- 0xD7 (XCHD A,@R1) - Exchange least-significant nibble
- 0xE0 (MOVX A,@DPTR) - External Move
- 0xE1 (AJMP page7) - Address Jump
- 0xE2 (MOVX A,@R0) - External Move
- 0xE3 (MOVX A,@R1) - External Move
- 0xE4 (CLR A) - Clear (fill with zeros)
- 0xF0 (MOVX @DPTR,A) - Extended Move
- 0xF1 (ACALL page7) - Absolute Call
- 0xF2 (MOVX @R0,A) - External Move
- 0xF3 (MOVX @R1,A) - External Move
- 0xF4 (CPL A) - Logical Complement; Logical NOT; 0 becomes 1 and vice versa

Directives

- DB - Define Byte; Use with labels to create constants - DATA: DB 10101010B
- END - End of code
- EQU - Equate; Define a constant - DATA EQU 10101010B
- ORG address - Origin; Begin the program code at the specified address

Addressing Modes

- 4 - Immediate or (if no immediate) the accumulator
- 5 - Direct Memory
- 6 - Indirect Register (@R0)
- 7 - Indirect Register (@R1)
- 8 - Direct Register (R0)
- 9 - Direct Register (R1)
- A - Direct Register (R2)
- B - Direct Register (R3)
- C - Direct Register (R4)
- D - Direct Register (R5)
- E - Direct Register (R6)
- F - Direct Register (R7)

Registers

- Stack-pointer (SP) [0x81] - 8-bit register
- Data-pointer (DP or DPTR) [0x82 and 0x83] - 16-bit register used to access PMEM and XRAM
 - DPH - The higher 8-bits of DPTR
 - DPL - The lower 8-bits of DPTR
- Program Status Word (PSW) [0xD0] - 8-bit register that is bit-addressable
 - PSW.0 - Parity (P); Stores the parity bit of the accumulator
 - PSW.1 - User Defined (UD)
 - PSW.2 - Overflow flag (OV); Set when addition produces a signed overflow
 - PSW.3 - Register Select 0 (RS0); Bank selection
 - PSW.4 - Register Select 1 (RS1); Bank selection
 - PSW.5 - Flag 0 (F0); Same use as PSW.1
 - PSW.6 - Auxiliary Carry (AC); Set when addition produces a carry from bit "3" to bit "4"
 - PSW.7 - Carry bit (C or CY); General carry register
- Accumulator (A) [0xE0] - The accumulator
- B register (B) [0xF0] - Used by the accumulator for multiplication and division instructions
- R0 - R7 - Used to store data

P3 Interrupts

- **Bit/Pin Function**
- 0/10 RxD
- 1/11 TxD
- 2/12 $\overline{\text{INT0}}$
- 3/13 INT1
- 4/14 T0
- 5/15 T1
- 6/16 $\overline{\text{WR}}$
- 7/17 $\overline{\text{RD}}$