8051 - Arithmetic and Logic

EE4380 Fall 2001

Class 8

Pari vallal Kannan

Center for Integrated Circuits and Systems

University of Texas at Dallas

Signed Arithmetic - Concepts

- Representation of the sign
  - Allocate one bit of all numeric quantities for the sign
  - Usually MSB (most significant bit) is assigned for the sign
  - The remaining bits represent the *magnitude*
- 8051 has only 8-bit registers
  - Signed numbers can have only a 7 bit magnitude
  - Positive numbers in 8051 = 0 to +127 (7 bits)
  - Negative numbers ???

Signed Arith. – Negative Numbers

- Negative Number representation in signed arithmetic
  - The sign bit (MSB) is 1
  - Magnitude is in 2’s complement form
- Examples

Represent –5

5 = 0000 0101

Cpl = 1111 1010

+1 = 1111 1011

Hex = FBH

Hence –5 = FBh

Represent –34H

34H = 0011 0100

Cpl = 1100 1011

+1 = 1100 1100

Hex = CCH

Hence –34H = CCH

Represent –128

128 = 1000 0000

Cpl = 0111 1111

+1 = 1000 0000

Hex = 80H

Hence –128 = 80H

Range

-128 = 80H

-127 = 81H

…..

-1 = FFH

0 = 00H

1 = 01H

+127 = 7FH

Signed Numbers - Usage

- Application may require a specific quantity be represented as a signed number
  - Temperature measurement –20deg, +10deg etc
  - Water level measurement in a tank
  - Gas tank monitor
- Data is collected and stored as an array of signed numbers
  - Some of the array elements can be negative, while others are positive
  - Identify negative numbers by the MSB. If MSB=1, the number is negative
- Same arithmetic operations (add, sub, mul, div etc) may need to be performed on the array elements, and the result can be positive or negative.

8051 – Signed Arithmetic

- 8051 uses negative number representation in the sub instruction. Not useful.
- When signed numbers are needed, programmer has to take care of signed arithmetic
- Overflow has to be dealt with. Carry flag is not enough, because only 7 bits carry the magnitude in signed numbers
- The 8051 provides another flag – OV (Overflow) for this purpose.

8051 - Signed Arithmetic (contd.)

- Addition

A+B A = 01H, B = FFH

A = +1, B = -1

A =   0000 0001

B =   1111 1111

+ = 1 0000 0000

A+B = 0H

A+B A = FEH, B = FFH

A = -2, B = -1

A =   1111 1110

B =   1111 1111

+ = 1 1111 1101

A+B = FDH = -3

- Subtraction

A-B A = 01H, B = FFH

A = +1, B = -1

2’s(B) = 0000 0000 +1 = 0000 0001

A   =   0000 0001

2’s(B) =   0000 0001

+   = 0 0000 0010

A-B = 02H

A-B A = FEH, B = 01H

A = -2, B = +1

2’s(B) = 1111 1110 +1 = 1111 1111

A   =   1111 1110

2’s(B) =   1111 1111

+   = 1 1111 1101

A-B = FDH = -3

8051 Signed Arith. - Overflow

- Overflow can occur from the magnitudes of the signed numbers, which can change the sign bit.
- Example

A+B, A=+96 (60H), B=+70(46H)

A = 0110 0000

B = 0100 0110

+ = 1010 0110 = A6H = -90 (wrong)

OV = 1, CY=0

96+70 = 166 > +127

- OV Flag is to be checked for error in signed arithmetic

8051 – OV Flag

- After arithmetic operations, OV is set if
  - Carry from D6 to D7 but no carry from D7
  - Carry from D7 but no carry from D6 to D7
  - These cases indicate a wrong result due to signed arithmetic
- After arithmetic operation involving signed numbers, check OV flag, for error detection
  - Use jb PSW.2 or jnb PSW.2
  - PSW.2 = OV

8051 Logic Instructions

- AND
  - anl dest, source ; dest = dest AND source
  - Commonly used to mask out (set to 0) a few bits in an operand
- OR
  - orl dest, source ; dest = dest OR source
  - Commonly used to set a few bits in an operand
- XOR
  - xrl dest, source ; dest = dest XOR source
  - Commonly used to clear a register, check if two registers have the same value and toggle a few bits
- Complement
  - cpl A ; A = A'
- None of these instructions affect any flags

8051 – Compare Instruction

- CJNE
  - Cjne dest, source, rel address
  - Compare dest and source and jump to relative address if not equal
  - Basically a subtract operation which does not change the operands but affects the CY flag
  - dest > source ➔ CY=0
  - dest < source ➔ CY=1

```
Cmp:        cjne R5, #80, NEQ
EQ:         ….             ;R5= #80
NEQ:        jnc GREAT
LESS:       …              ;R5< #80
GREAT:      ….             ;R5 > #80
```

- Example
  - Monitor P1 continuosly and
  - exit if P1=63H

```
Loop:       mov A, P1
            cjne A, #63, loop
```

8051 – Rotate and Swap

- Bitwise rotation is required in many apps like serial comm., control etc.
- Rotate right
  - rr A ; rotate right A
  - mov A, #AAH, rr A ; A = 55H
- Rotate left
  - rl A ; rotate left A
  - Mov A, #55H, rl A; A = AAH
- Rotate right/left with Carry
  - Use CY in the rotate sequence (9 bit rotate)
  - rlc A and rrc A
- Swap nibbles
  - swap A ; swaps D7-D4 with D3-D0

8051 – Single Bit Instructions

- Set a bit
  - set bit ;bit = 1
- Clear a bit
  - clr bit ;bit = 0
- Complement a bit
  - cpl bit ;bit = bit’
- Conditional Jump on bit value
  - jb (jump if bit=1), jnb (jump if bit=0), jbc (jump if bit=1 and clear the bit)

Bit addressable Regs and Memory

- All I/O ports (P0 – P3), B, PSW, IP, IE, ACC, SCON and TCON are bit addressable (BARs)
- The bits of BARs can be referred to as Register.bitnum (P0.1, PSW.2, IE.4 etc) or by their bit address
- Bit address is the base address of the register + the bit number
  - ACC Base address is E0H, hence ACC.1=E1H, ACC.7=E7H
  - P0, Base address is 80H, hence P0.0=80H, P0.5=84H and so on
- 16 bytes of the internal RAM is bit addressable
  - 20H to 2FH has a bit address of 00H to 7FH
  - clr 67H ; clear bit D7H of RAM location 2CH
  - setb 05H ; set bit 5 of RAM location 20H

Single Bit Opn with CY flag

- 8051 has special instructions that directly manipulate CY flag
  - setb C; clr C; cpl C; mov b,C; mov C,b; jnc, jc, anl C,b; anl C,/b; orl C,b; orl C,/b
  - anl C, /b ;C = CY AND b'
- Example: Turn ON fan (P2.2) and turn OFF light (P2.3)

```
Fan_on:      setb C
             orl C,P2.2    ;CY = CY OR P2.2
             mov P2.2, C   ;turn on fan if not already ON
Light_off:   clr C
             anl C,P2.3    ;CY = CY AND P2.3
             mov P2.3,C    ;turn off light if not already OFF
```