Input Output Port Programming

Instructor

Zhizheng Wu

吴智政

School of Mechatronic Engineering and Automation

- Hardware properties of the ports in 8051

- Examples in assembly language

HARDWARE PROPERTIES OF THE PORTS

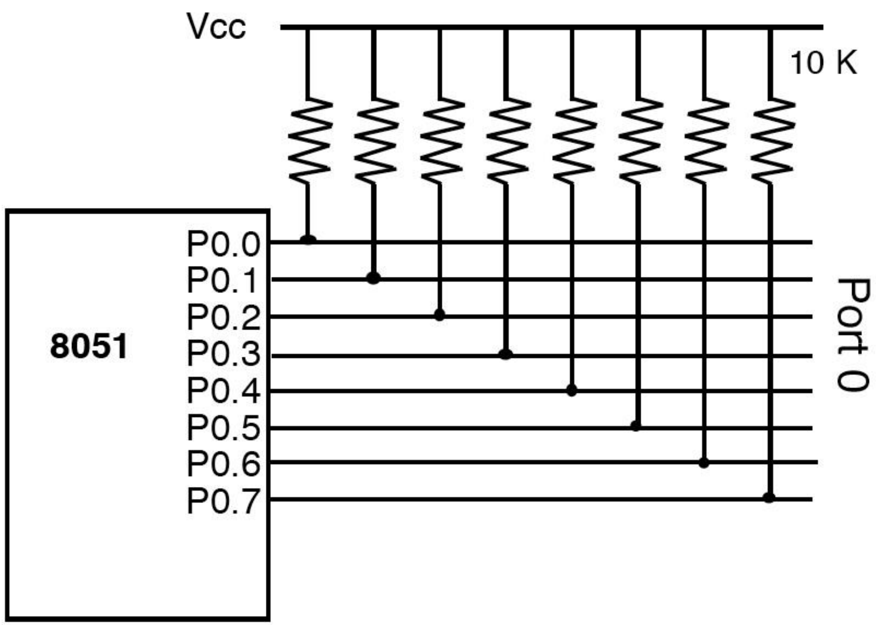
- 32 pins P0, P1, P2 and P3 function as I/O port lines
  - **Port 0:** a dual purpose port on pins 32-39
  - **Port 1:** a dedicated I/O port on pins 1-8
  - **Port 2:** dual purpose port on pins 21-28 (could be a general purpose I/O or high byte of the address bus for external memory)
  - **Ports 3:** dual purpose port on pins 10-17
  - All the ports upon RESET are configured as **input**, ready to be used as output ports

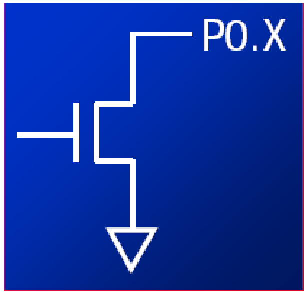
HARDWARE PROPERTIES OF THE PORTS

- Port 0 is designated as AD0-AD7, allowing it to be used for both address and data
  - When connecting an 8051/31 to an external memory, port 0 provides both address and data
  - The 8051 multiplexes address and data through port 0 to save pins
  - ALE indicates if P0 has address or data
    - When ALE=0, it provides data D0-D7
    - When ALE=1, it has address A0-A7

HARDWARE PROPERTIES OF THE PORTS

- Port 0 can be used for input or output, each pin must be connected externally to a 10K ohm pull-up resistor
  - ➢ This is due to the fact that P0 is an open drain, unlike P1, P2, and P3
  - ➢ Open drain is a term used for MOS chips in the same way that open collector is used for TTL chip





- Writing 1 to all the bits configures the ports as an input port.

Example:

```
MOV A, #0FFH        ; Load FFH into the accumulator
MOV P0, A           ; Make P0 an input port
MOV A, P0           ; Move data from port 0 into the
                      accumulator
```

HARDWARE PROPERTIES OF THE PORTS

- In 8051-based systems with no external memory connection
  - Both P1 and P2 are used as simple I/O without pull-up resistor
- In 8031/51-based systems with external memory connections
  - Port 2 must be used along with P0 to provide the 16-bit address for the external memory
    - P0 provides the lower 8 bits via A0 – A7
    - P2 is used for the upper 8 bits of the 16-bit address, designated as A8 – A15, and it cannot be used for I/O

HARDWARE PROPERTIES OF THE PORTS

- Port 3 can be used as input or output
  - Port 3 does not need any pull-up resistors
- Port 3 has the additional function of providing some extremely important signals
  - Alternate pin functions for Port 3

P3 Bit	Function	Pin
P3.0	RxD	10
P3.1	TxD	11
P3.2	$\overline{\text{INT0}}$	12
P3.3	$\overline{\text{INT1}}$	13
P3.4	T0	14
P3.5	T1	15
P3.6	$\overline{\text{WR}}$	16
P3.7	$\overline{\text{RD}}$	17

Serial communications

External interrupts

Timers

Read/Write signals of external memories

HARDWARE PROPERTIES OF THE PORTS

- There are two possibilities for the 8051 to read a I/O port
  - ➢ Some instructions read the status of port pins
  - ➢ Others read the status of an internal port latch

- Confusion between them is a major source of errors in 8051 programming. Especially where external hardware is concerned.

HARDWARE PROPERTIES OF THE PORTS

① Reading data directly from an input port: In general, reading data from an input port can be done using one of the following options

Mnemonic	**Example**	**Description**
MOV A, PX	**MOV A, P2**	**Bring into A the data at P2 pins**
JNB PX.Y, ..	**JNB P2.1, Target**	**Jump if pin P2.1 is low**
JB PX.Y, ..	**JB P1.3, Target**	**Jump if pin P1.3 is high**
MOV C, PX.Y	**MOV C, P2.4**	**Copy status of pin P2.4 into CY**

Note: X is port number 0, 1, 2, or 3 for P0 – P3, Y is the pin number

② Some instructions read the latch for the output port: The instructions below read the contents of an internal port latch instead of reading the status of an external output pin.

Mnemonics	Example
ANL PX	ANL P1,A
ORL PX	ORL P2,A
XRL PX	XRL P0,A
JBC PX.Y,TARGET	JBC P1.1,TARGET
CPL PX.Y	CPL P1.2
INC PX	INC P1
DEC PX	DEC P2
DJNZ PX.Y,TARGET	DJNZ P1,TARGET
MOV PX.Y,C	MOV P1.2,C
CLR PX.Y	CLR P2.3
SETB PX.Y	SETB P2.3

HARDWARE PROPERTIES OF THE PORTS

- Reading the latch for the output port

Example:
ORL P2, A

➢ First, the contents of the internal latch are read.

➢ The contents of the latch are ORed with the contents of the accumulator.

➢ The result is written back to the port latch

➢ The port pin data is changed to the same value as that of the port latch.

☞ These operations are referred to as "Read-Modify-Write" operations.

HARDWARE PROPERTIES OF THE PORTS

- This feature of Read-modify-write technique saves many lines of code by combining in a single instruction all three actions

1. Reading the port
2. Modifying it
3. Writing to the port

- I/O Ports' Bit-addressability
  - Any bit on any of the four ports can be accessed and modified using the instructions below:

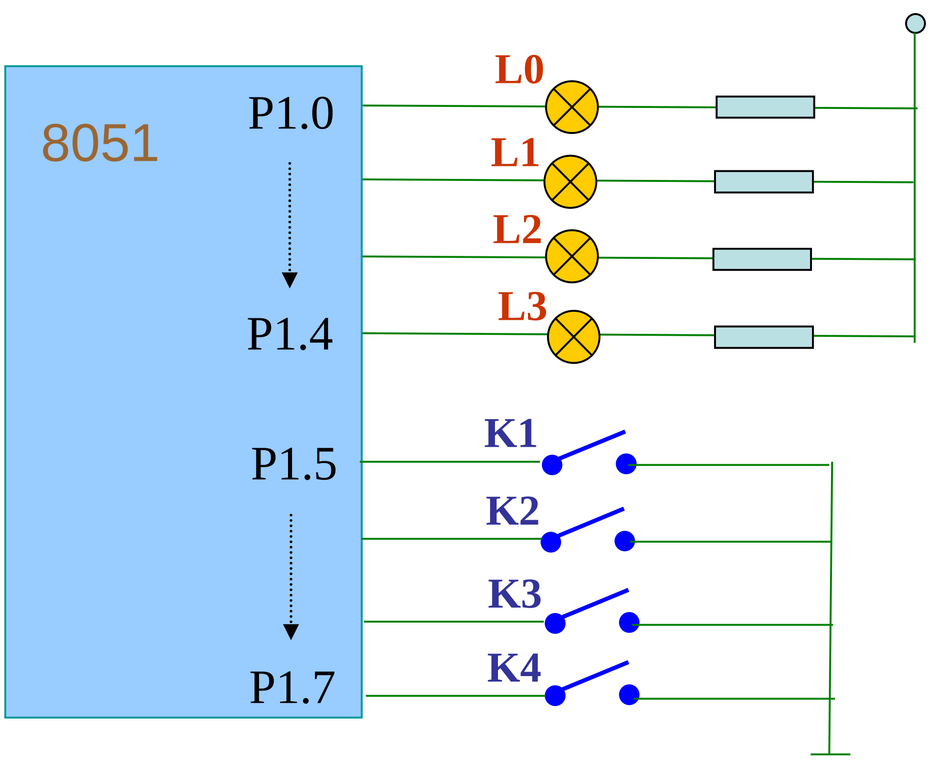
Instruction	Function
SETB bit	Set the bit (bit = 1)
CLR bit	Clear the bit (bit = 0)
CPL bit	Complement the bit (bit = NOT bit)
JB bit,target	Jump to target if bit = 1 (jump if bit)
JNB bit,target	Jump to target if bit = 0 (jump if no bit)
JBC bit,target	Jump to target if bit = 1, clear bit (jump if bit, then clear)

Example:

SETB P1.2 ; Set bit 2 in port 1 to high.

EXAMPLE 1

- The circuit connection is shown in the figure. The L0-L3 are lights and K1-K3 are switches. Program a subroutine to make the status of the lights follow the those of the switches. Namely, if the switch Ki is closed, then the light Li is on.




```
SW:   PUSH  ACC
      PUSH  PSW
      ORL  P1, #0F0H ;set P1.5-7 as input pins
      MOV  A, P1
      SWAP  A;
      MOV  P1, A
      POP  PSW
      POP  ACC
      RET
```

EXAMPLE 2

- Program a delay subroutine
  - CPU executing an instruction takes a certain number of clock cycles These are referred as to as machine cycles
  - In 8051, one machine cycle lasts 12 oscillator periods
  - Find the period of the machine cycle for 11.0592 MHz crystal frequency

Solution:
11.0592/12 = 921.6 kHz;
machine cycle is 1/921.6 kHz = 1.085µs

EXAMPLE 3 (3-15)

- Find the size of the delay in following program, if the crystal frequency is 11.0592MHz.

```
        MOV A, #55H
AGAIN:  MOV P1, A
        ACALL  DELAY
        CPL A
        SJMP  AGAIN
        ;---time delay-------
DELAY:  MOV R3, #200
HERE:   DJNZ R3, HERE
        RET
```

Solution:

	Machine cycle
DELAY: MOV R3, #200	1
HERE: DJNZ R3, HERE	2
RET	1

Answer: $[(200\text{x}2)+1+1]\text{x}1.085\mu s = 436.17\mu s$.

EXAMPLE 4 (3-17)

- Find the size of the delay in following program, if the crystal frequency is 11.0592MHz.

```
                                  Machine Cycle
DELAY:  MOV R2, #200                   1
AGAIN:  MOV R3, #250                   1
HERE:   NOP                            1
        NOP                            1
        DJNZ  R3, HERE                 2
        DJNZ  R2, AGAIN                2
        RET                            1
```

Solution:

For HERE loop, we have (4x250)x1.085μs＝1085μs. For AGAIN loop repeats HERE loop 200 times, so we have 200x1085μs＝217000μs. But “MOV R3,#250” and “DJNZ R2,AGAIN” at the start and end of the AGAIN loop add (3x200x1.085)=651μs. As a result we have 217000+651+2x1.085 = 217653.17μs.

EXAMPLE 5 (8-1)

- Write a program to create a square wave of 50% duty cycle on bit 0 of port 1.

Solution:

The 50% duty cycle means that the “on” and “off” state (or the high and low portion of the pulse) have the same length. Therefore, we toggle P1.0 with a time delay in between each state.

```
SQW:    SETB  P1.0          ;set to high bit 0 of port 1
        LCALL  DELAY        ;call the delay subroutine
        CLR  P1.0           ;P1.0=0
        LCALL  DELAY
        SJMP  SQW           ;keep doing it
DELAY: MOV R3, #200
HERE:   DJNZ R3, HERE
        RET
```

EXAMPLE 6 (8-3)

- Assume that bit P2.3 is an input and represents the condition of an oven. If it goes high, it means that the oven is hot. Monitor the bit continuously. Whenever it goes high, send a high-to-low pulse to port P1.5 to turn on a buzzer.

Solution:

```
HERE:  JNB  P2.3, HERE          ;keep monitoring for high
       SETB  P1.5               ;set bit P1.5=1
       CLR  P1.5                ;make high-to-low
       SJMP  HERE               ;keep repeating
```

EXAMPLE 7

```
0000                     ORG
0000 7455       BACK:    MOV A, #55H           ;load A with 55H
0002 F590                MOV P1, A             ;send 55H to P1
0004 7C99                MOV R4, #99H
0006 7D67                MOV R5, #67H
0008 12                  LCALL DELAY           ;time delay
000B 74AA                MOV A, #0AAH          ;load A with AA
000D F590                MOV P1, A             ;send AAH to P1
000F 12                  LCALL DELAY
0012 80                  SJMP BACK             ;keeping doing this
0014            ;-------this is the delay subroutine------
0300                     ORG
0300 C004       DELAY:   PUSH 4                ;push R4
0302 C005                PUSH 5                ;push R5
0304 7CFF                MOV R4, #0FFH         ;R4=FFH
0306 7DFF       NEXT:    MOV R5, #0FFH         ;R5=FFH
0308 DD         AGAIN:   DJNZ R5, AGAIN
030A DC                  DJNZ R4, NEXT
030C D005                POP 5                 ;POP into R5
030E D004                POP 4                 ;POP into R4
0310 22
0311
```

1. Fill the box
2. Calculate the delay time