The 8051 Microcontroller and Embedded Systems

I/O PORT PROGRAMMING

OBJECTIVES

- List the 4 ports of the 8051
- Describe the dual role of port 0 in providing both data and addresses
- Code Assembly language to use the ports for input or output
- Explain the dual role of port 0 and port 2
- Code 8051 instructions for I/O handling
- Code I/O bit-manipulation programs for the 8051

SECTION 4.1: 8051 I/O PROGRAMMING

- **I/O port pins and their functions**

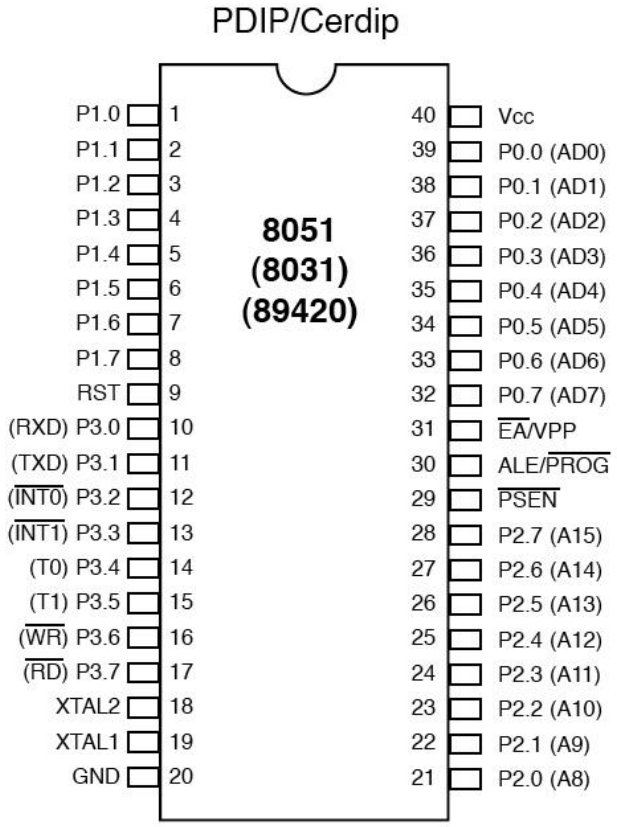

Figure 4–1 8051 Pin Diagram

SECTION 4.1: 8051 I/O PROGRAMMING

- All the ports upon RESET are configured as inputs, ready to be used as input ports.
- When the first 0 is written to a port, it becomes an output.
- To reconfigure it as an input, a 1 must be sent to the port.
- To use any of these ports as an input port, it must be programmed.

SECTION 4.1: 8051 I/O PROGRAMMING

- **Port** 0

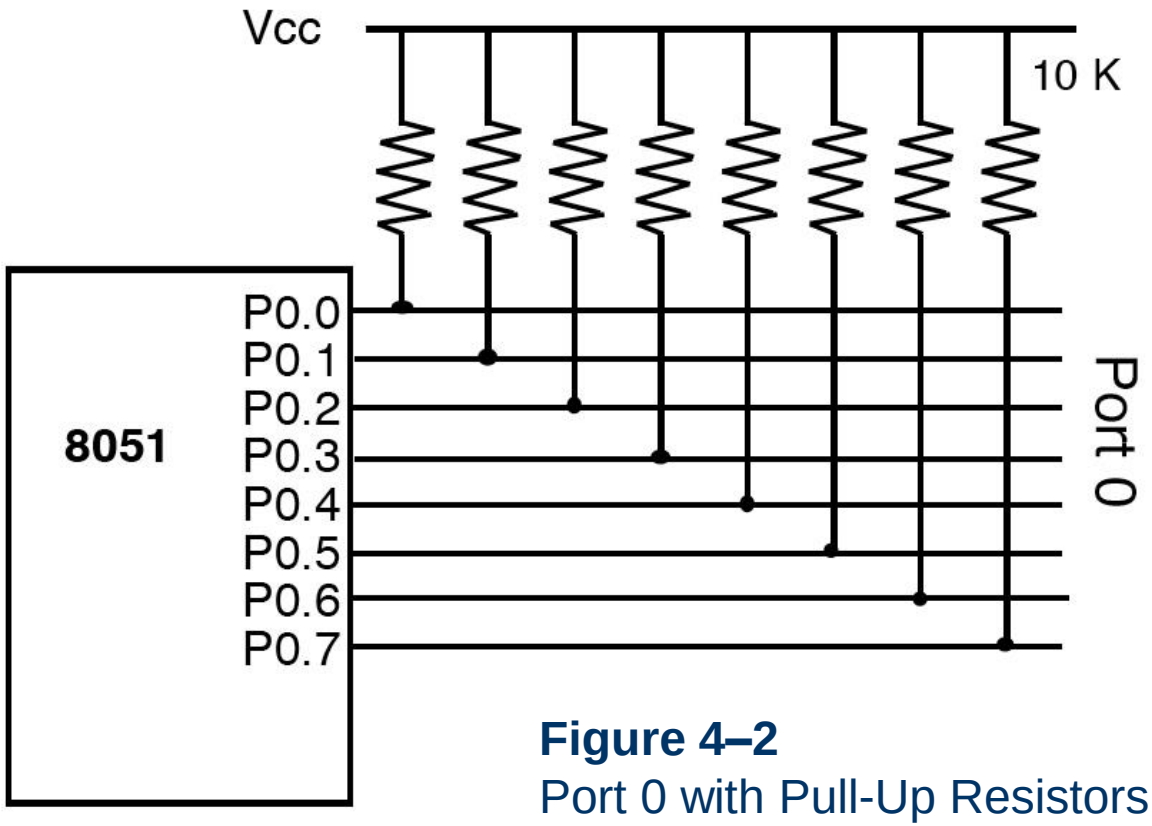

Figure 4–2
Port 0 with Pull-Up Resistors

Port 0

- It can be used for input or output.
- To use the pins of port 0 as both input and output each pin must be connected externally to a 10 Kohm pull-up resistor.
- This is due to the fact that P0 is an open drain, unlike P 1, P2, and P3

Toggle all bits of P0

Continuously send out to port 0 the alternating values 55H & AA H

```
        MOV     A,#55H
BACK:   MOV     P0,A
        ACALL   DELAY
        CPL     A
        SJMP    BACK
```

Port 0 as input

- **With resistors connected to port 0, in order to make it an input, the port must be programmed by writing 1 to all the bits.**

```
;Get a byte from P0 and send it to P1
        MOV  A,#0FFH    ;A = FF hex
        MOV  P0,A       ;make P0 an input port
                        ;by writing all 1s to it
BACK:   MOV  A,P0       ;get data from P0
        MOV  P1,A       ;send it to port 1
        SJMP BACK       ;keep doing it
```

Dual role of port 0

- Port 0 is also 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.

SECTION 4.1: 8051 I/O PROGRAMMING

- Port 1
- It can be used as input or output.
- This port does not need any pull-up resistors since it already has pull-up resistors internally.
- Upon reset, port I is configured as an input port.

Port 1 as input

- If port 1 has been configured as an output port, to make it an input port again, it must programmed as such by writing 1 to all its bits.

Port 1 as input

- Port 1 is configured first as an input port by writing 1 s to it, then data is received from that port and saved in R7, R6, and R5.

```
MOV   A,#0FFH  ;A=FF hex
MOV   P1,A     ;make P1 an input port
               ;by writing all 1s to it
MOV   A,P1     ;get data from P1
MOV   R7,A     ;save it in reg R7
ACALL DELAY    ;wait
MOV   A,P1     ;get another data from P1
MOV   R6,A     ;save it in reg R6
ACALL DELAY    ;wait
MOV   A,P1     ;get another data from P1
MOV   R5,A     ;save it in reg R5
```

SECTION 4.1: 8051 I/O PROGRAMMING

- **Port** 2
- **Port 2 occupies a total of 8 pins (pins 21 through 28).**
- **It can be used as input or output.**
- **Port** 2 **does not need any pull-up resistors since it already has pull-up resistors internally.**
- **Upon reset, port 2 is configured as an input port.**

Port 2 as input

- To make port 2 an input, it must programmed as such by writing 1 to all its bits.
- In the following code, port 2 is configured first as an input port by writing is to it.

```
;Get a byte from P2 and send it to P1
          MOV  A,#0FFH    ;A=FF hex
          MOV  P2,A       ;make P2 an input port by
                          ;writing all 1s to it
BACK:     MOV  A,P2       ;get data from P2
          MOV  P1,A       ;send it to Port 1
          SJMP BACK       ;keep doing that
```

Dual role of port 2

- In 8031-based systems, port 2 must be used along with P0 to provide the 16-bit address for external memory.
- Port 2 is also designated as A8 - A15, indicating its dual function.
- Since an 8051/31 is capable of accessing 64K bytes of external memory, it needs a path for the 16 bits of the address.
- P0 provides the lower 8 bits via A0 - A7
- P2 provides bits A8 - A 15 of the address.
- When the 8051 /31 is connected to external memory, P2 is used for the upper 8 bits of the 16-bit address, and it cannot be used for I/O.

Port 3

- Port 3 can be used as input or output.
- P3 does not need any pull-up resistors.
- Port 3 is configured as an input port upon reset.
- Port 3 has the additional function of providing some extremely important signals such as interrupts, serial I/O, timer/counter and read/write control for external memory.

SECTION 4.1: 8051 I/O PROGRAMMING

- **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

Table 4–1
Port 3 Alternate Functions

SECTION 4.1: 8051 I/O PROGRAMMING

- **Different ways of accessing the entire 8 bits**

```
BACK:   MOV   A,#55H
        MOV   P1,A
        ACALL DELAY
        MOV   A,#0AAH
        MOV   P1,A
        ACALL DELAY
        SJMP  BACK
```

Different ways of accessing the entire 8 bits

```
BACK:   MOV   P1,#55H
        ACALL DELAY
        MOV   P1,#0AAH
        ACALL DELAY
        SJMP  BACK
```

```
        MOV   A,#55H    ;A=55 HEX
BACK:   MOV   P1,A
        ACALL DELAY
        CPL   A         ;complement reg. A
        SJMP  BACK
```

SECTION 4.1: 8051 I/O PROGRAMMING

- **Ports status upon reset**

Register	Reset Value (Binary)
P0	11111111
P1	11111111
P2	11111111
P3	11111111

Table 4–2 Reset Value of Some 8051 Ports

SECTION 4.2: I/O BIT MANIPULATION PROGRAMMING

- A powerful feature of 80511/0 ports is their capability to access individual bits of the port without altering the rest of the bits in that port.
- Of the four 8051 ports, we can access either the entire 8 bits or any single bit without altering the rest.
- "SETB X. Y" where X is the port number 0, 1, 2, or 3, and Y is the desired bit number from 0 to 7 for data bits DO to D7.
- "SETB P1.5" sets high bit 5 of port 1.

SECTION 4.2: I/O BIT MANIPULATION PROGRAMMING

- The following code toggles bit P1.2 continuously.

```
BACK:     CPL   P1.2      ;complement P1.2 only
          ACALL DELAY
          SJMP  BACK

;another variation of the above program follows
AGAIN:    SETB  P1.2      ;change only P1.2=high
          ACALL DELAY
          CLR   P1.2      ;change only P1.2=low
          ACALL DELAY
          SJMP  AGAIN
```

SECTION 4.2: I/O BIT MANIPULATION PROGRAMMING

- **I/O ports and bit-addressability**

P0	P1	P2	P3	Port Bit
P0.0	P1.0	P2.0	P3.0	D0
P0.1	P1.1	P2.1	P3.1	D1
P0.2	P1.2	P2.2	P3.2	D2
P0.3	P1.3	P2.3	P3.3	D3
P0.4	P1.4	P2.4	P3.4	D4
P0.5	P1.5	P2.5	P3.5	D5
P0.6	P1.6	P2.6	P3.6	D6
P0.7	P1.7	P2.7	P3.7	D7

Table 4–3
Single-Bit Addressability of Ports

SECTION 4.2: I/O BIT MANIPULATION PROGRAMMING

- **I/O ports and bit-addressability**

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)

Table 4–4 Single-Bit Instructions

SECTION 4.2: I/O BIT MANIPULATION PROGRAMMING

- **Checking an input bit**

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 P1.2 is high
`MOV C, PX.Y`	`MOV C, P2.4`	Copy status of pin P2.4 to CY

Table 4–5 Instructions For Reading an Input Port

Reading a single bit into the carry flag

Example 4-6

A switch is connected to pin P1.7. Write a program to check the status of the switch and perform the following:
(a) If switch = 0, send letter 'N' to P2.
(b) If switch = 1, send letter 'Y' to P2.

Use the carry flag to check the switch status. This is a repeat of the last example.

Solution:

```
        SETB P1.7          ;make P1.7 an input
AGAIN:  MOV  C,P1.2        ;read the SW status into CF
        JC   OVER          ;jump if SW = 1
        MOV  P2,#'N'       ;SW = 0, issue 'N' to P2
        SJMP AGAIN         ;keep monitoring
OVER:   MOV  P2,#'Y'       ;SW = 1, issue 'Y' to P2
        SJMP AGAIN         ;keep monitoring
```

Reading input pins vs. port latch

- **Some instructions read the status of port pins while others read the status of an internal port latch.**
- **When reading ports there are two possibilities:**
  - **1. Read the status of the input pin.**
  - **2. Read the internal latch of the output port.**

Instructions for reading input ports

- To make any bit of any 8051 port an input port, we must write 1 (logic high) to that bit.
- After we configure the port bits as input, we can use only certain instructions in order to get the external data present at the pins into the CPU.

Reading latch for output port

Mnemonic		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

Note: x is 0, 1, 2, or 3 for P0 - P3.

Table 4–6 Instructions Reading a Latch (Read-Modify-Write)

Read-modify-write feature

- **The ports in the 8051 can be accessed by the read-modify-write technique.**
  - **(1) reading the port**
  - **(2) modifying its value**
  - **(3) writing to the port**

```
        MOV   P1,#55H       ;P1 = 01010101
AGAIN:  XLR   P1,#0FFH      ;EX-OR P1 with 11111111
        ACALL DELAY
        SJMP  AGAIN
```