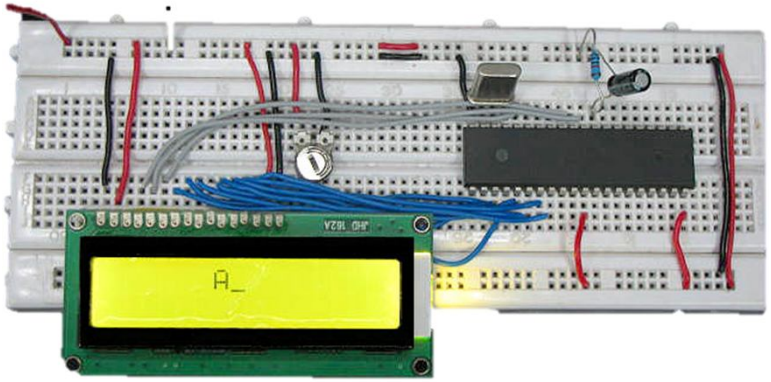
Microprocessors and Microcontrollers (EE-231)

Lab-4

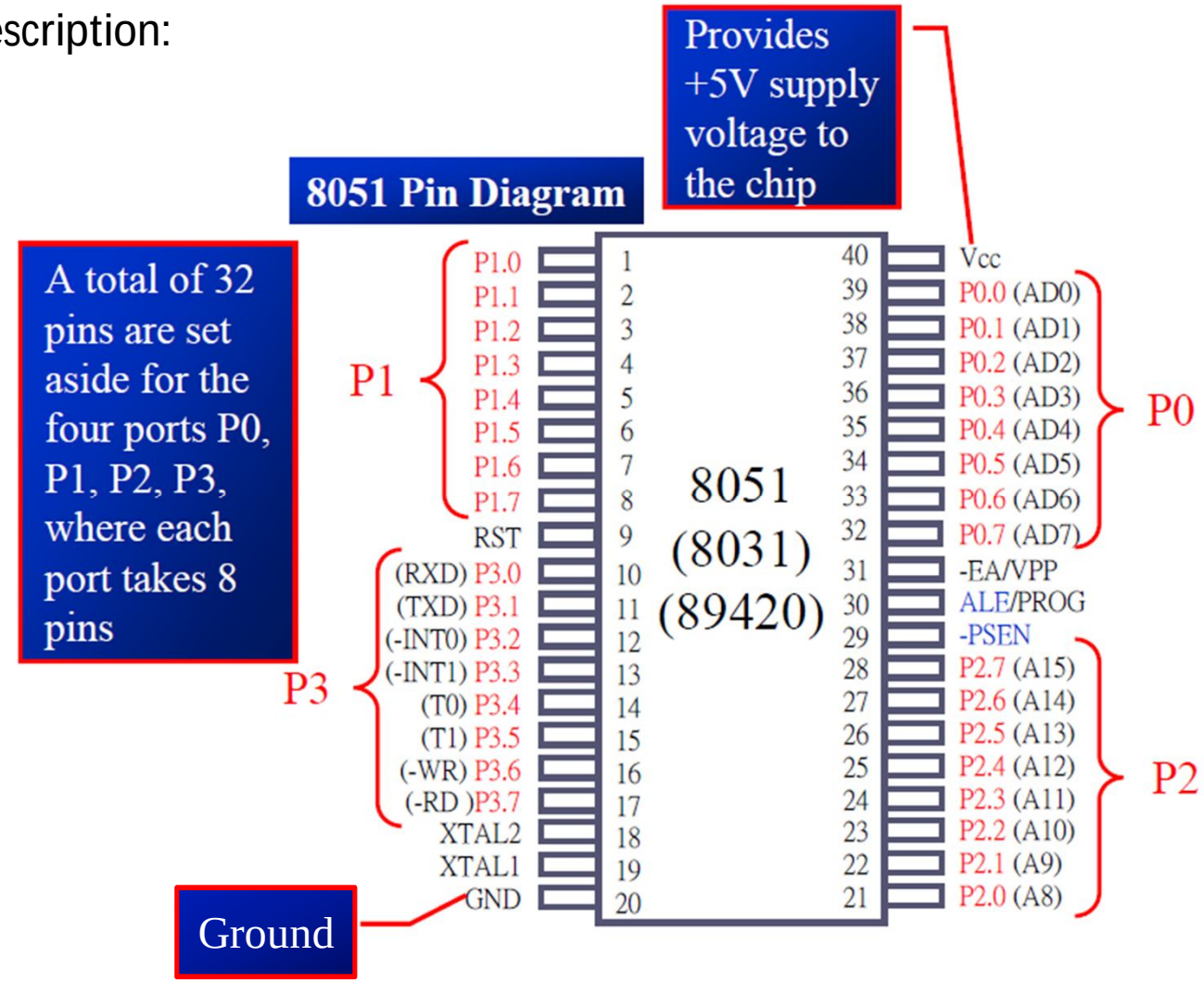
Main Objectives

- Brief introduction to 8051 hardware Pins
- Learning how to prototype 8051 hardware circuit
- How to view memory contents in 8051
- Implementation of a very simple Lock system in 8051 and its simulation in Proteus



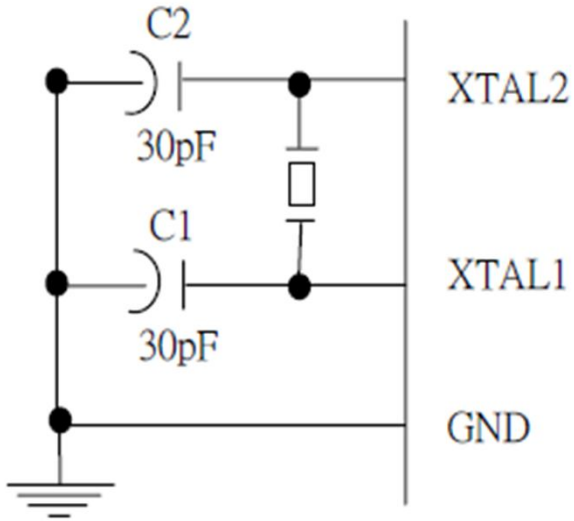
8051 Hardware Connections

- Pins Description:



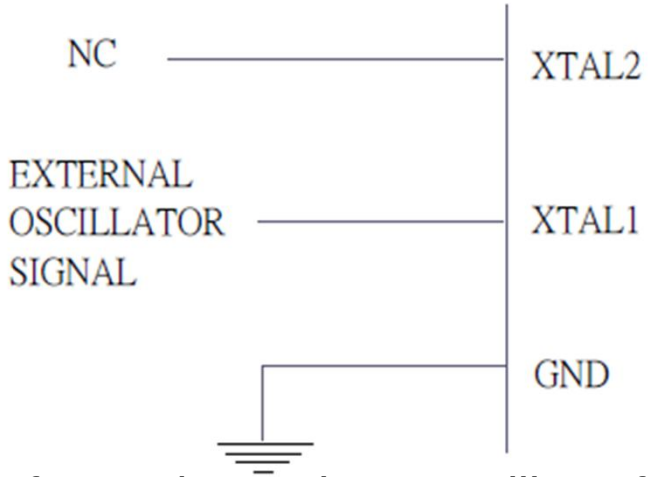
8051 Hardware Connections

- XTAL1 and XTAL2
- The 8051 requires an external clock to run it
- A quartz crystal oscillator is connected to inputs XTAL1 (pin19) and XTAL2 (pin18)
- The quartz crystal oscillator also needs two capacitors of 30 pF value



XTAL 1 and XTAL2

- XTAL1 and XTAL2
- If you use a frequency source other than a crystal oscillator, such as a TTL oscillator
- It will be connected to XTAL1 while XTAL2 is left unconnected



- The speed of 8051 refers to the maximum oscillator frequency connected to XTAL
- ex. A 12-MHz chip must be connected to a crystal with 12 MHz frequency or less
- We can observe the frequency on the XTAL2 pin using the oscilloscope

Reset

- RESET pin is an input and is active high (normally low)
- Upon applying a high pulse to this pin, the microcontroller will reset and terminate all activities
- This is often referred to as a power-on reset
- Activating a power-on reset will cause all values in the registers to be lost

RESET value of some 8051 registers

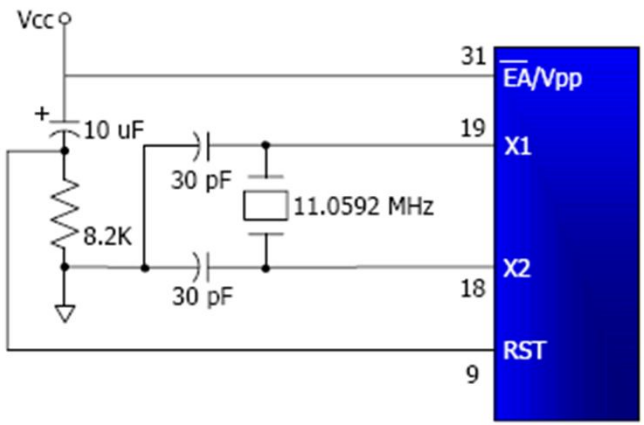
we must place the first line of source code in ROM location 0

Register	Reset Value
PC	0000
DPTR	0000
ACC	00
PSW	00
SP	07
B	00
P0-P3	FF

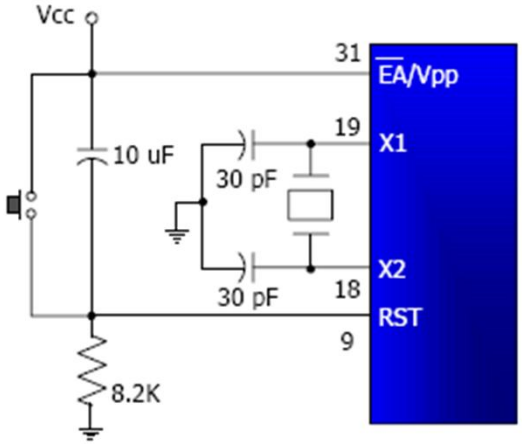
Reset

- In order for the RESET input to be effective, it must have a minimum duration of 2 machine cycles
- In other words, the high pulse must be high for a minimum of 2 machine cycles

Power-on RESET circuit



Power-on RESET with debounce



EA (External Access)

- EA, “external access’’, is an input pin and must be connected to **Vcc** or **GND**

1. The 8051 family members all come with on-chip ROM to store programs
   EA pin is connected to Vcc in that case.

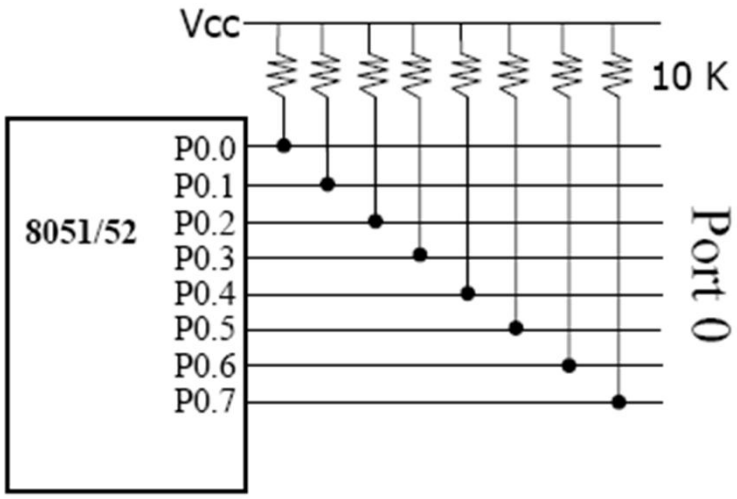
2. The 8031 and 8032 family members do no have on-chip ROM, so code is stored on an external ROM and is fetched by 8031/32
   EA pin must be connected to GND to indicate that the code is stored externally

‘PSEN’ & ‘ALE’

- The following two pins are used mainly in 8031-based systems
- PSEN, “program store enable’’, is an output pin
- This pin is connected to the OE pin of the ROM
- ALE, “address latch enable”, is an output pin and is active high
- The 8031 multiplexes address and data through port 0 to save pins
- ALE pin is used for demultiplexing the address and data by connecting to the G pin of the 74LS373 chip

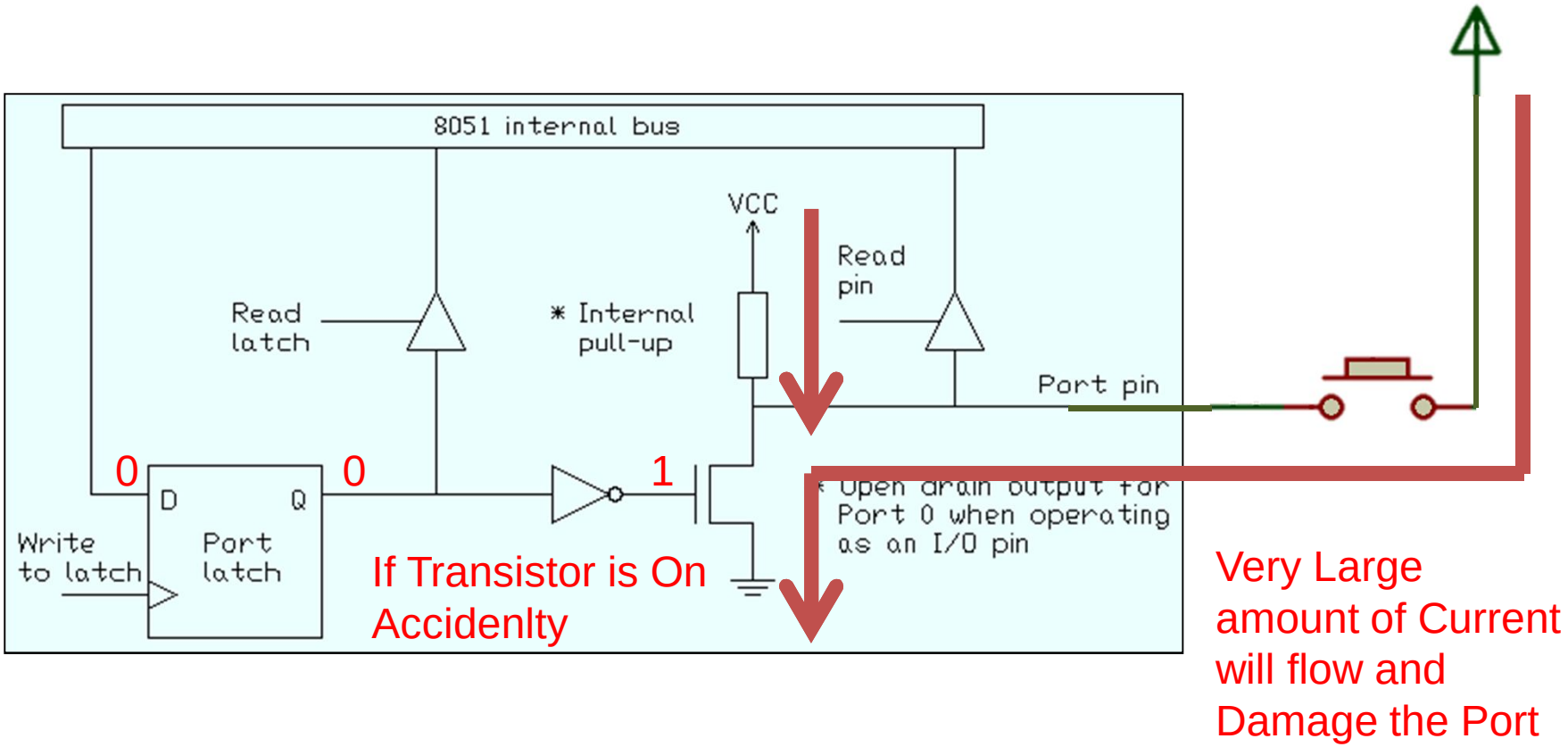
P0 Pull-up Resistors

- Port 0 has no internal pull-up resistors therefore It must be pulled-up externally



Never do this to I/O pin

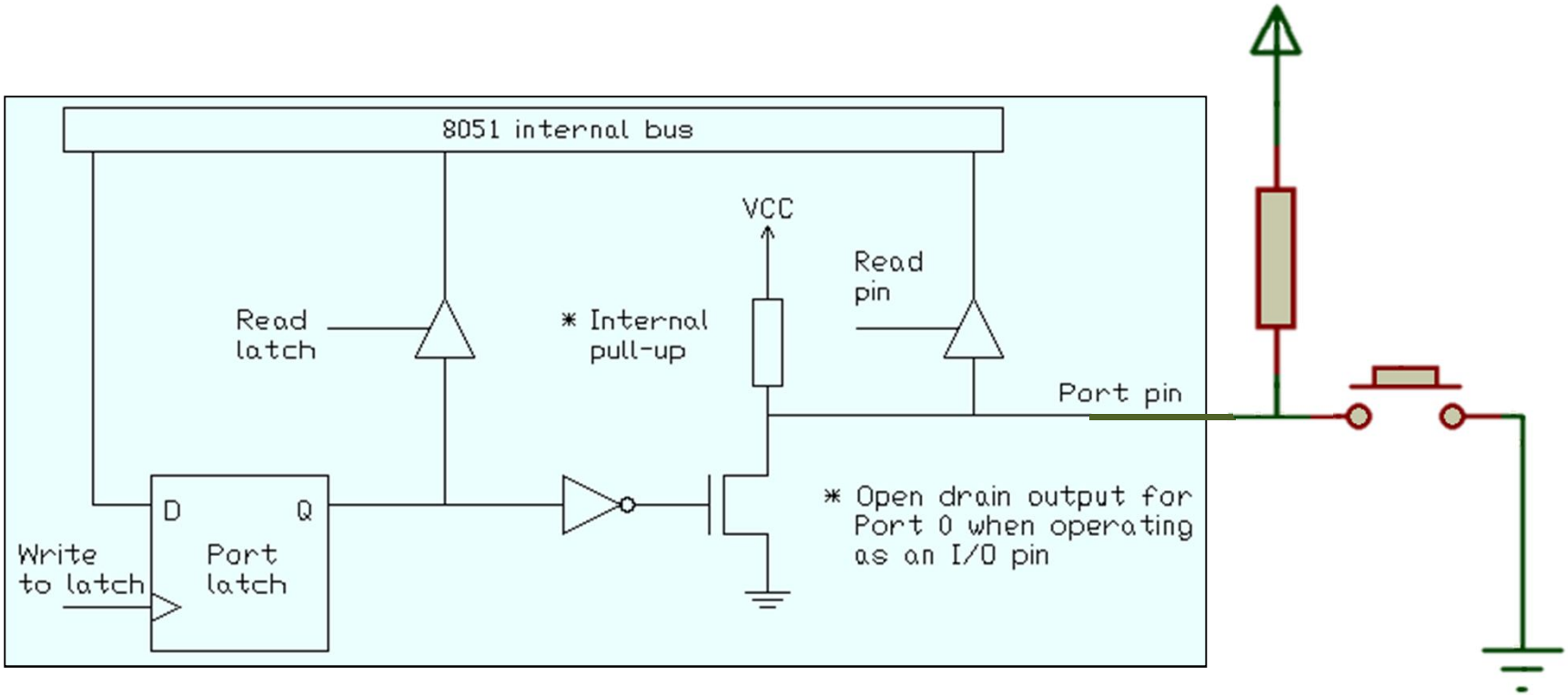
- Never give Vcc directly to a Pin. If you want to, then use A pull up Resistor.



The right way to connect switch to I/O

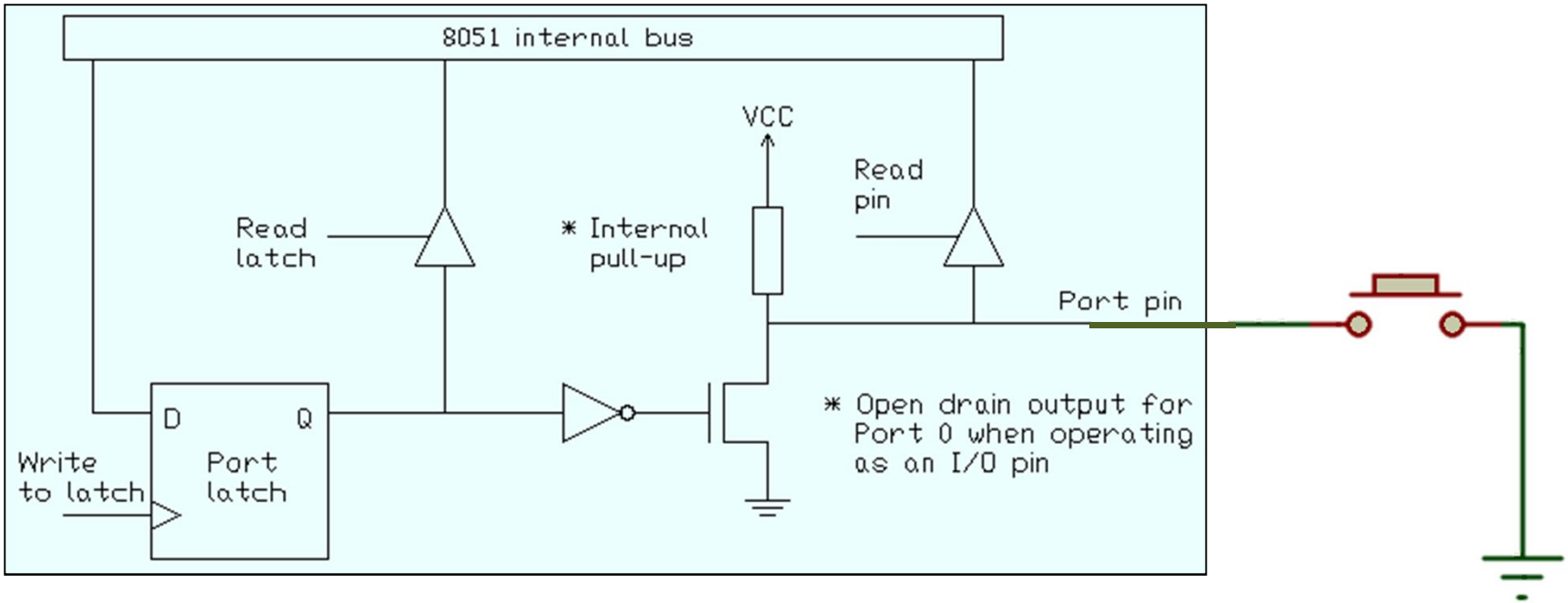
- This is the right way to do it..

Pull-up Resistor



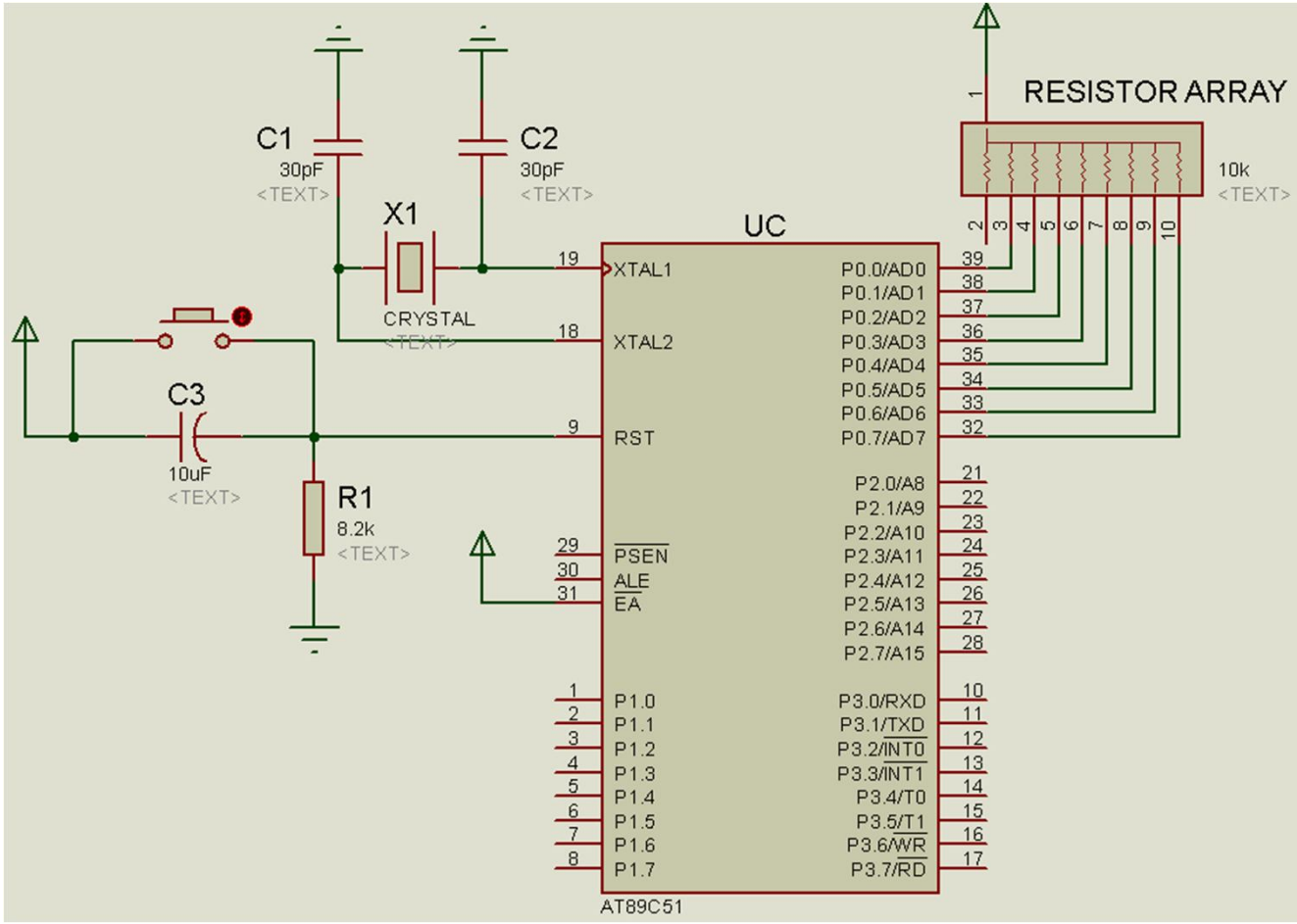
The right way to connect switch to I/O

- We can avoid Pull-Up Resistor when giving Gnd to Pin because port internally has one.



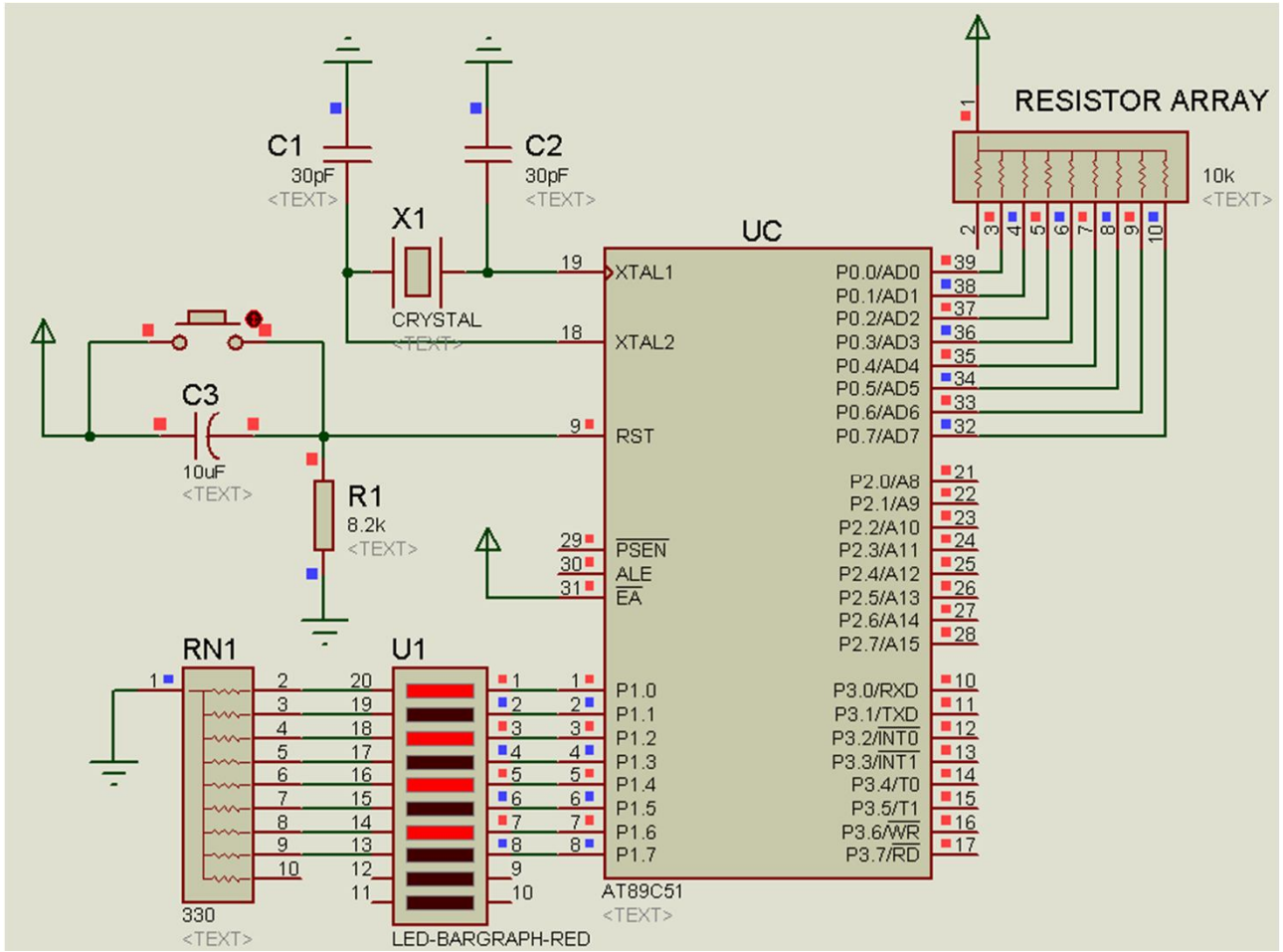
8051 Minimum Hardware Connections

- The following is a diagram showing minimum hardware connection to run the 8051 microcontroller



Next Week Assignment

- Connect the circuit according to the given scheme on breadboard and run a toggle program on BAR LED.



Todays Task

- You will have to implement todays task in Proteus.
- Write a code in assembly that accepts a security code and verifies it and then grants access to the user if that code is correct. Use DIP switch as the code input. When you have selected a combination at the dip switch. Press a Push button to request access. The Microcontroller will then verify your code and if that combination matches the built-in value stored in register, the controller will write 'A' on Seven Segment, otherwise it will write 'E' on seven segment.
- (Hint: store built-in value in program memory and access it using DPTR)

Todays Task

test.asm

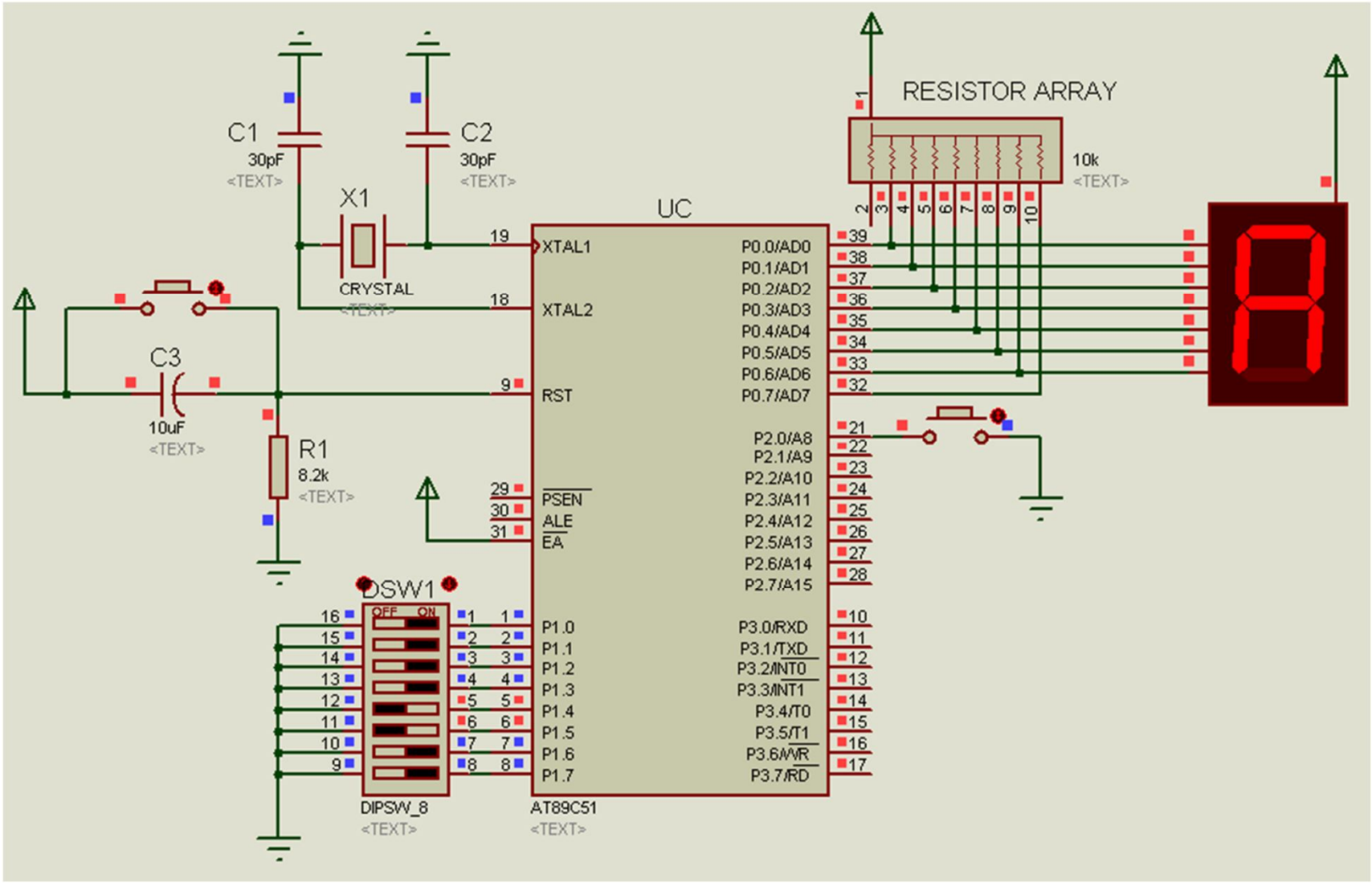
```
ORG 0
INPUT EQU P1     ; THE CODE INPUT READ FROM DIP SWITCH
BUTTON EQU P2.0; REQUEST ACCESS INPUT i.e. THE PUSH BUTTON
SS EQU P0; SEVEN SEGMENT

MOV INPUT,#0FFH; MAKE P1 INPUT
SETB BUTTON; MAKE P2.0 INPUT
MOV DPTR,#THEPASSWORD; OR I COULD HAVE WRITTEN "MOV DPTR,#300"

;----Now we will have to wait till button is pressed i.e. made "0"
START:JB BUTTON,START
;-----Now start the main code
CLR A
MOVC A,@A+DPTR
CJNE A,INPUT,SKIP
MOV SS,#88H ; THE ACCESS GRANTED (IF EQUAL)
SJMP START; DO NOT LET CONTROLLER TO GO BELOW THIS LINE
SKIP:MOV SS,#86H
SJMP START

ORG 300H
THEPASSWORD:
DB 30H
END
```

Todays Task


C1
30pF
<TEXT>
C2
30pF
<TEXT>
X1
CRYSTAL
<TEXT>
C3
10uF
<TEXT>
R1
8.2k
<TEXT>
RESISTOR ARRAY
10k
<TEXT>
UC
XTAL1
XTAL2
RST
PSEN
ALE
EA
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0/A8
P2.1/A9
P2.2/A10
P2.3/A11
P2.4/A12
P2.5/A13
P2.6/A14
P2.7/A15
P1.0
P1.1
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P3.0/RXD
P3.1/TXD
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
DSW1
OFF ON
DIPSW_8
<TEXT>
AT89C51
<TEXT>