COMPUTER ARCHITECTURE AND MICROCONTROLERS

S4 B TECH

PROJECT REPORT ON

8-BIT BINARY UP DOWN COUNTER

Done by

- Abhishek Babu T (Reg No: TCR19EC004)
- Ashik M M (Reg No: TCR19EC019)
- Sivaprasad A S (Reg No: TCR19EC051)

Under guidance of :

Prof. Shaniba SMI

Department of Electronics & Communication Engineering

Government Engineering college

Thrissur-680009

OCT 5,2021

TABLE OF CONTENTS

AIM

To design and implement an 8-Bit Binary Up/Down Counter on 8051 microcontroller using LED interfacing and simulate the design using Edsim51

INTRODUCTION

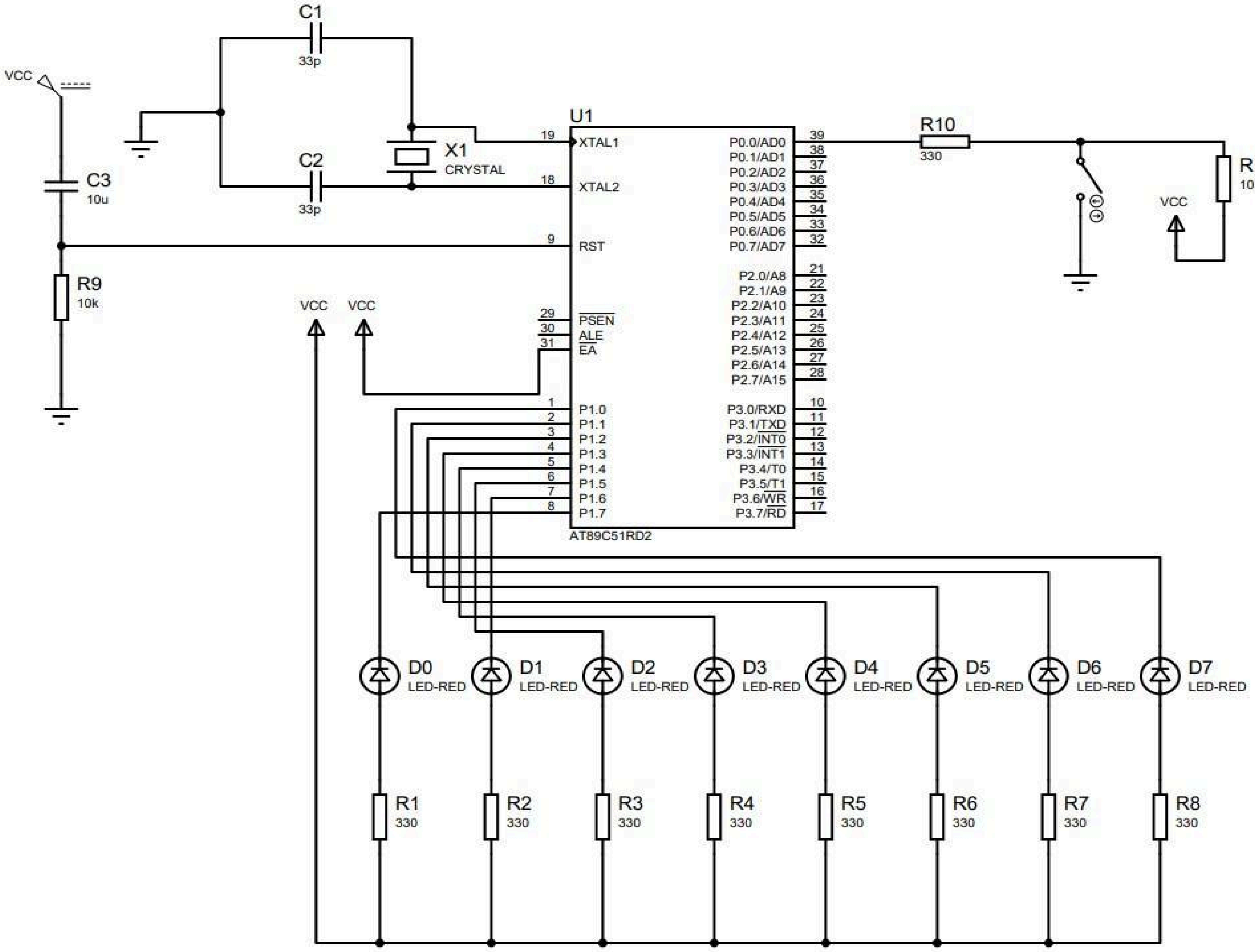
A binary counter is a hardware circuit that is made out of a series of flip-flops. The output of one flip-flop is sent to the input of the next flip-flop in the series. Counters are used in many different applications these days.

Some count up from zero and provide a change in state of output upon reaching a predetermined value; others count down from a preset value to zero to provide an output state change. However, some counters can operate in both up and down count mode, depending on the state of an up/down count mode input pin.

A Binary Up/Down counter is a bidirectional counter capable of counting in either the up direction or the down direction. In this project, we have designed an 8-bit up/down counter which counts from 0 to 255 as an up counter and vice versa. LED interfacing is used to implement the counter.

A push button switch is provided to decide whether to count up or down. The counter counts up when the button is pushed and vice versa.

CIRCUIT DIAGRAM


VCC
C1
33p
C2
33p
X1
CRYSTAL
C3
10u
R9
10k
U1
19
XTAL1
18
XTAL2
9
RST
29
PSEN
30
ALE
31
EA
1
P1.0
2
P1.1
3
P1.2
4
P1.3
5
P1.4
6
P1.5
7
P1.6
8
P1.7
AT89C51RD2
P0.0/AD0
39
P0.1/AD1
38
P0.2/AD2
37
P0.3/AD3
36
P0.4/AD4
35
P0.5/AD5
34
P0.6/AD6
33
P0.7/AD7
32
P2.0/A8
21
P2.1/A9
22
P2.2/A10
23
P2.3/A11
24
P2.4/A12
25
P2.5/A13
26
P2.6/A14
27
P2.7/A15
28
P3.0/RXD
10
P3.1/TXD
11
P3.2/INT0
12
P3.3/INT1
13
P3.4/T0
14
P3.5/T1
15
P3.6/WR
16
P3.7/RD
17
R10
330
R11
10k
VCC
VCC
VCC
D0
LED-RED
D1
LED-RED
D2
LED-RED
D3
LED-RED
D4
LED-RED
D5
LED-RED
D6
LED-RED
D7
LED-RED
R1
330
R2
330
R3
330
R4
330
R5
330
R6
330
R7
330
R8
330

COMPONENTS REQUIRED

- AT89C51 Microcontroller: The AT89C51 is an age old 8-bit microcontroller from the Atmel family. It works with the popular 8051 architecture and hence is used by most beginners till date. It is a 40 pin IC package with 4Kb flash memory. It has four ports and all together provide 32 Programmable GPIO pins.

- 8 LEDs: A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. It is a two terminal device (anode & cathode) used heavily in electronics & automotive industries.

- 8 Resistors: Resistors are used to limit the current flowing through the LEDs in order to prevent them from damage.

- Power Supply (V): A dc power supply is provided to the LED interface and the microcontroller.

- 8051 Programmer: Edsim51 with Assembly language programming is used for the project.

- Push Button Switch: Push button switches are widely used in embedded system projects and the knowledge about interfacing them to 8051 is very essential in designing such projects. A typical push button switch has two active terminals that are normally open and these two terminals get internally shorted when the push button is depressed.

WORKING

PUSH BUTTON INTERFACING:

Push button switch is connected to the first bit of PORT 0 (**P0.0**) which is configured as an input pin. Which is connected to a pull up resistor as there is no internal pull up resistors for port P0**.** Thus P0.0 pin is at Vcc potential when the switch is not pressed. When the switch is pressed this pin P0.0 will be grounded.

Pushing the button makes the counter function as Up counter and the counter counts down when not pressed.

LED INTERFACING:

To use an LED as the output device, LED should be connected to Microcontroller port and the Microcontroller has to programmed inside to make LED ON/OFF. In 8051 we are connecting LEDs to port P1 of microntroller.

The anodes of all LEDS are connected to the power supply and the cathodes are connected to the P1 port pins. Hence common anode connection of LEDs is used.

The LED glows only when port pin value of microcontroller is low(0). When the port value is low, the LED is forward biased and hence current flows through the LED resulting in the glow. This is called current sinking of LED.

Initially port P1 value is set to FF and hence all the LEDS are turned off. Change in P1 pins values from 1 to 0 results in LEDs turning ON and vice versa.

The resistors connected limits the current flowing and prevents the LEDs and microcontroller from damage.

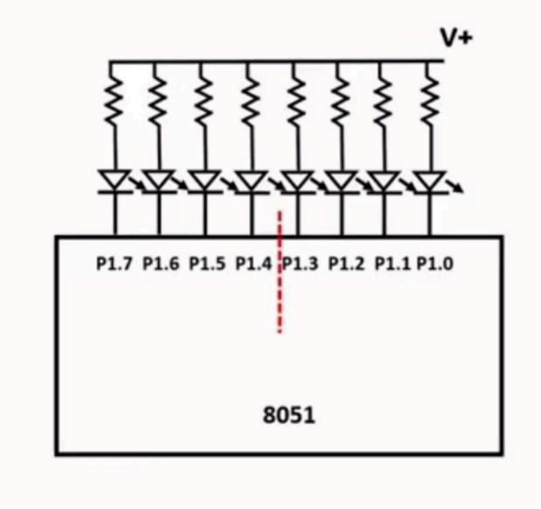


Fig 1. 8051 LED Schematics

UP COUNTER:

The upcounter functions when pin value of P0.0 is logic low (0). The value of accumulator (A) is initially set to FF so that all LEDs are turned OFF. The contents of A is decremented for the counter to count up. The counter counts from 0 to 255. CJNE function checks whether the value of A is 0 and the counter runs until the condition is satisfied.

DOWN COUNTER:

The downcounter functions when pin value of P0.0 is logic high (1). The value of accumulator is set to 00 so that all LEDs are glowing. The counter counts from 255 to 0. For the counter to count down, the accumulator value is incremented in each clock cycle. The process goes on until the value of Accumulator is FF.

DELAY:

Loops are used to generate delay between consecutive LED blinking. The loop variable values are stored in R0 & R1 registers. The value of R0 is initially set to 80H and R1 to FFH. Delay subroutine is called in every clock cycle to increase the time between each states of the counter.

FUNCTIONS & SUBROUTINES

Functions used:

- DEC A: Decrement Accumulator.

- INC A: Increment Accumulator.
- ACALL: Absolute call within 2K block. ACALL unconditionally calls a subroutine located at the indicated address. The instruction increments the PC twice to obtain the address of the following instruction, then pushes the 16-bit result onto the stack and increments the stack pointer twice. The destination address is obtained by successively concatenating the five highorder bits of the incremented PC, opcode bits 7-5, and the second byte of the instruction. The subroutine called must therefore start within the same 2K block of the program memory as the first byte of the instruction following ACALL.
- CJNE: Compare and jump if not equal. CJNE compares the magnitudes of the first two operands and branches if their values are not equal. The carry flag is set if the unsigned integer value of <dest-byte> is less than the unsigned integer value of <src-byte>; otherwise the carry is cleared. Neither operand is affected.
- SJMP: Short jump. Program control branches unconditionally to the address indicated.
- DJNZ: Decrement and Jump if Not Zero. DJNZ decrements the location indicated by 1, and branches to the address indicated by the second operand if the resulting value is not zero.

- RET: Return from subroutine.

Subroutines used:

- Start: For getting input from switch.
- Up: Functions as up counter.
- Repeat1: Loop for counting up.
- Down: Functions as down counter.
- Repeat2: Loop for counting down.
- Delay: For generating delay using two loops.

CODE

```
                Org 00H

Start:          Mov P0,#0FFh                  ;Making P0 as input port

                Acall Delay                   ;Call Delay

                Mov A,P0                      ;Getting switch value
                CJNE A,#00H,Down
                SJMP Up

Up:             Mov A,#0FFh                   ;Data
                Mov P1,A                      ; Output Data to LEDs
                Acall Delay
Repeat1:                                      ;Count Up
                Dec A                         ;Decrement A by one
                Mov P1,A                      ;Output Data to LEDs
                Acall Delay
                CJNE A,#00H,Repeat1
                SJMP Start

Down:           Mov A,#00H
                Mov P1,A
                Acall Delay
Repeat2:                                      ;Count Down
                Inc A                         ;Increment A by one
                Mov P1,A                      ;Output Data to LEDs
                Acall Delay
                CJNE A,#0FFH,Repeat2
                SJMP Start                    ;Repeat

Delay:          Mov R0,#80h                   ;Outer Loop
Again:          Mov R1,#0FFh                  ;Inner Loop
Here:           DJNZ R1,Here                  ;jump if R1 not 0
                DJNZ R0,Again                 ;jump if R0 not 0
                Ret
                End
```

CONCLUSION

Designed an 8-bit binary up down counter using LED interface on 8051 and simulated the design on Edsim51. Learned about the internal architecture and components of AT89C51 8051 microcontroller.

REFERENCES

- Youtube
- Electronics Tutorials
- Elprocus.com
- Engineers garage