MINISTRY OF SCIENCE AND HIGHER EDUCATION OF THE RUSSIAN Federation
Federal State Budgetary Educational Institution of Higher Education
"Kazan National Research Technical University named after A. N. Tupolev-KAI"
(KNRTU-KAI)
Institute of Computer Technologies and Information Security
Department of Computer Systems

Report № 5

«Proteus Virtual System Modeling (VSM)
Part I. Some Logic Function Design
Part II. Delay Loops Applications Flasher & Counter »

«Architecture of embedded systems»

Student 4167 (group number)  (signature, data) Edwin G. Carreno (full name)

Associate professor of the computer systems' department Daria V. Shirshova

Grade _____

_____
(signature, data)

Kazan 2021

Content

1. Part I tasks – Some Logic Function Design

1. Write an assembly program to achieve the addition/subtraction of two numbers entered to port B and display the result on port A.

   For this laboratory practice, the following program allows to the user add and subtract two binary number entered in the Port B (first number in least significant nibble, and second number most significant nibble). Any user can choose the operation using the signal in the 4th pin of the Port A, zero for subtraction and one for addition.

```
;==========================================================================
; Main.asm file generated by New Project wizard
;
; Created:   Sun Apr 20 2021
; Processor: PIC16F84A
; Compiler:  MPASM (Proteus)
;==========================================================================

;==========================================================================
; DEFINITIONS
;==========================================================================

#include p16f84a.inc                ; Include register definition file

;==========================================================================
; VARIABLES
;==========================================================================
CBLOCK 0x20
                     FIRSTNUMBER
                     SECONDNUMBER
      endc
;==========================================================================
; RESET and INTERRUPT VECTORS
;==========================================================================
      ; Reset Vector
RST   code  0x0
goto start
;==========================================================================
; CODE SEGMENT
;==========================================================================

start:
      BSF               STATUS, RP0   ; select Register Bank 1
      MOVLW          0x10                ; Setting Output direction
      MOVWF          TRISA

      MOVLW          0x77                ; setting Input direction RB[2:0] and RB[2:0] or
0b0111_0111
      MOVWF          TRISB

      BCF               STATUS, RP0
      goto Loop

Loop:

      ; extracting first number from PORTB --> PORTB[3:0]
      MOVF           PORTB, W
      ANDLW          0x07
      MOVWF          FIRSTNUMBER

      ; extracting second number from PORTB --> PORTB[7:4]
```

```
        MOVF            PORTB, W
        ANDLW           0x70
        MOVWF           SECONDNUMBER
        SWAPF           SECONDNUMBER, W
        MOVWF           SECONDNUMBER

        ;decision making about operation
        BTFSC           PORTA,4
        goto addition
        goto substraction

        goto  Loop

addition:
        MOVF            FIRSTNUMBER, W
        ADDWF           SECONDNUMBER, W
        MOVWF           PORTA
        goto Loop

substraction:
        MOVF            FIRSTNUMBER, W
        SUBWF           SECONDNUMBER, W
        MOVWF           PORTA
        goto Loop

 END
;=========================================================================
```

2. Simulate the program using the circuit shown in figure via Proteus software. Verify it operates properly when simulated.

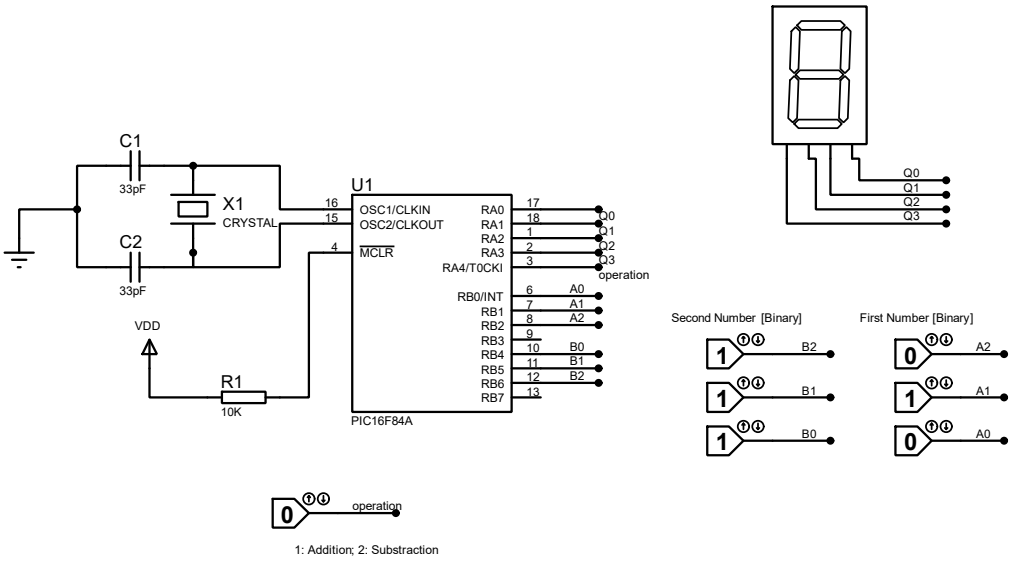

Figure 1. Schematic diagram for laboratory 5, addition and subtraction of binary numbers.

Addition

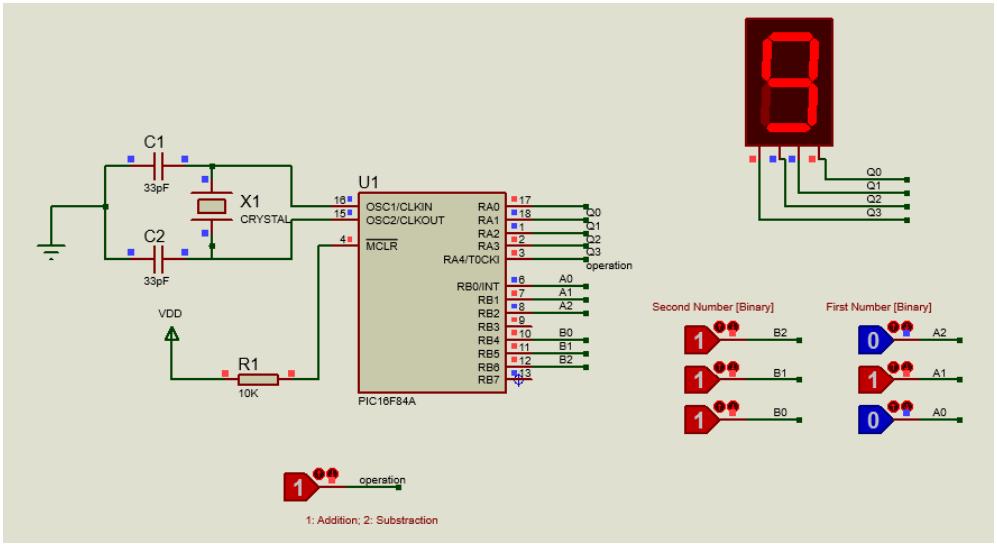

Figure 2. Addition of two numbers, first number is 7 and second number is 2, the result is 9.

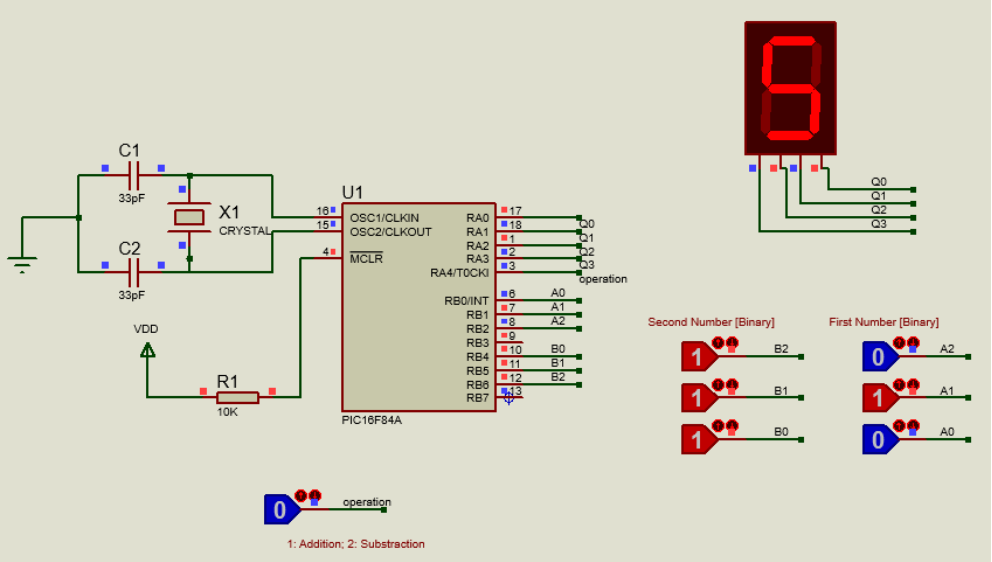

Figure 3. Subtraction of two numbers, first number is 7 and second number is 2, the result is 5.

3. Program a PIC16F84A using the QL2006 programmer.

4. Build the circuit using the programmed PIC16F8A and then observe its operation. Demonstrate the circuits operation to the instructor.

2. Part II tasks – Delay Loops Applications Flasher & Counter

Flasher

1. Write an assembly program to make a Flasher on RB0 (Hint: Use the Delay Subroutine).

```
;===============================================================
; Main.asm file generated by New Project wizard
;
; Created:   Sun May 9 2021
; Processor: PIC16F84A
; Compiler:  MPASM (Proteus)
;===============================================================

;===============================================================
; DEFINITIONS
;===============================================================

#include p16f84a.inc                ; Include register definition file

;===============================================================
; VARIABLES
;===============================================================

;===============================================================
; RESET and INTERRUPT VECTORS
;===============================================================

      ; Reset Vector
RST   code  0x0
      goto  Start

;===============================================================
; CODE SEGMENT
;===============================================================

PGM   code
Start:
      BSF           STATUS, RP0    ; select Register Bank 1
      MOVLW      0x00              ; Setting Output direction
      MOVWF      TRISB

      BCF           STATUS, RP0
```

```
      MOVLW     0x00
      MOVWF     PORTB
      goto Loop

Loop:
      MOVF      PORTB, W
      XORLW     0x01
      MOVWF     PORTB
      goto          Delay
      goto          Loop

Delay:
      MOVLW     .255
      MOVWF     0CH
      LOOP1
      MOVLW     .255
      MOVWF     0DH
      LOOP0
      NOP
      DECFSZ    0DH, F
      GOTO          LOOP0
      DECFSZ    0CH, F
      GOTO      LOOP1
      GOTO      Loop
 ;=====================================================================
END
```

2. Simulate the program using the circuit shown in figure via Proteus software. Verify it operates properly when simulated.

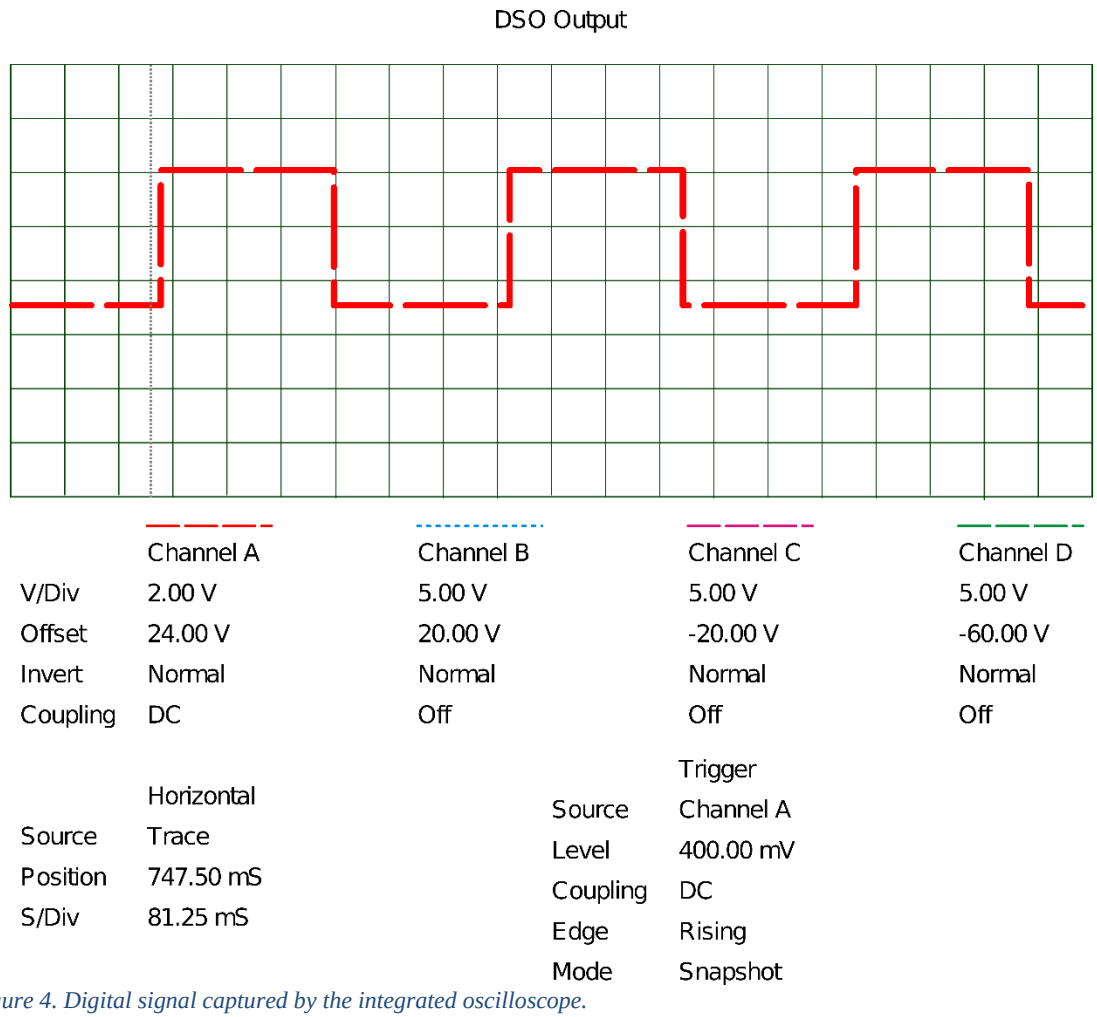

Figure 4. Digital signal captured by the integrated oscilloscope.

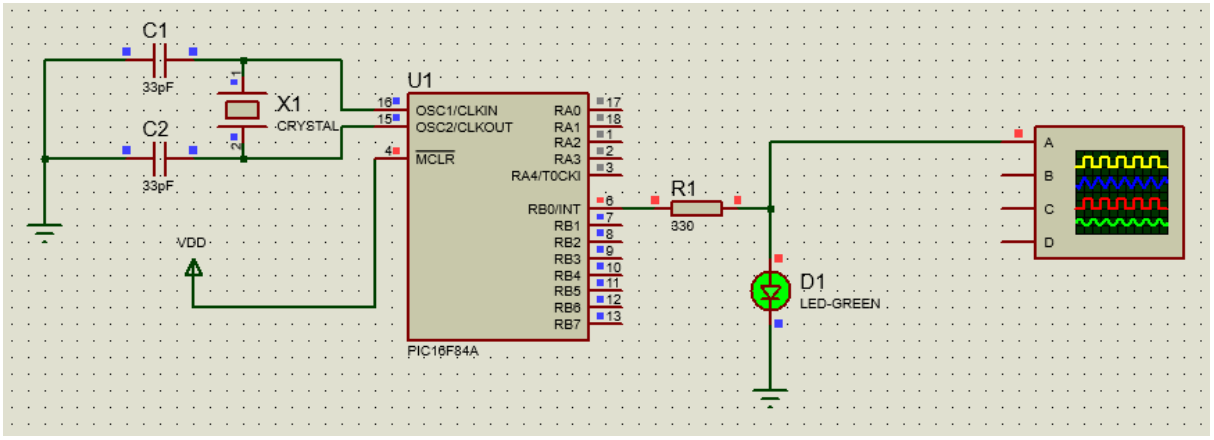

Figure 5. High state at the RB0 output.

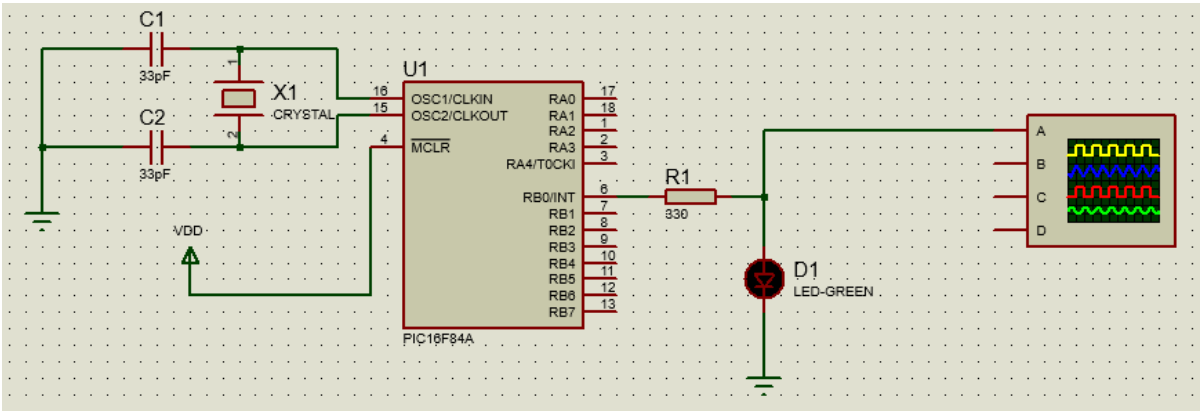

Figure 6. Low state at the RB0 output.

3. Program a PIC16F84A using the QL2006 programmer.

4. Build the circuit using the programmed PIC16F84A and then observe its operation. Demonstrate the circuits operation to the instructor.

Counter

1. Write an assembly program to make a Counter (0-9) on Port A. RB0 (Hint: use the delay subroutine).

```
;=====================================================================
; Main.asm file generated by New Project wizard
;
; Created:   Sun Apr 20 2021
; Processor: PIC16F84A
; Compiler:  MPASM (Proteus)
;=====================================================================

#include p16f84a.inc                ; Include register definition file

;=====================================================================
; VARIABLES
;=====================================================================
CBLOCK 0x20
      COUNTNUMBER
endc
;=====================================================================
; RESET and INTERRUPT VECTORS
;=====================================================================

      ; Reset Vector
RST   code  0x0
      goto  Start

;=====================================================================
```

```
; CODE SEGMENT
;=====================================================================

PGM   code
Start:
      BSF             STATUS, RP0   ; select Register Bank 1
      MOVLW       0x10; Setting Output direction
      MOVWF       TRISA

      BCF             STATUS, RP0
      GOTO Loop

Loop:
                  CLRF          COUNTNUMBER

      INNERLOOP
                   MOVF       COUNTNUMBER,0
                   MOVWF      PORTA
                   CALL       DELAY
                   INCF       COUNTNUMBER, 1
                   MOVF       COUNTNUMBER, 0
                   XORLW      .10
                   BTFSS      STATUS, Z
                   GOTO       INNERLOOP
     GOTO Loop

DELAY:
      MOVLW       .255
      MOVWF       0CH
      LOOP1
      MOVLW       .255
      MOVWF       0DH
      LOOP0
      NOP
      DECFSZ      0DH, F
      GOTO            LOOP0
      DECFSZ      0CH, F
      GOTO        LOOP1
      RETURN
 ;=====================================================================
END
```

2. Simulate the program using the circuit shown in figure via Proteus software. Verify it operates properly when simulated.

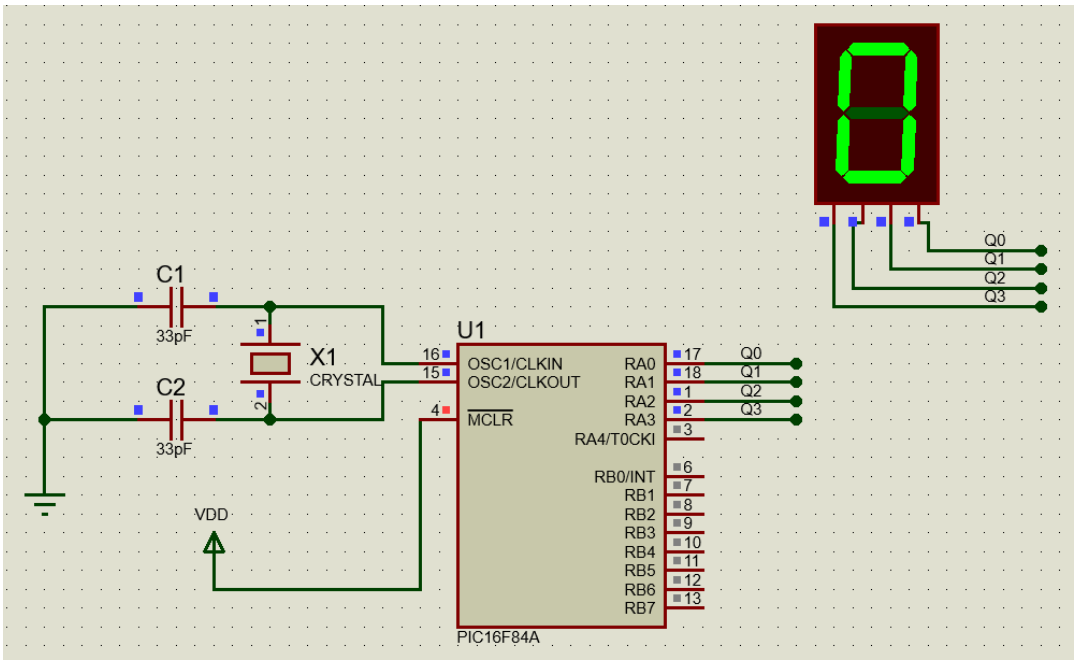

Figure 7. Output value at PORTA

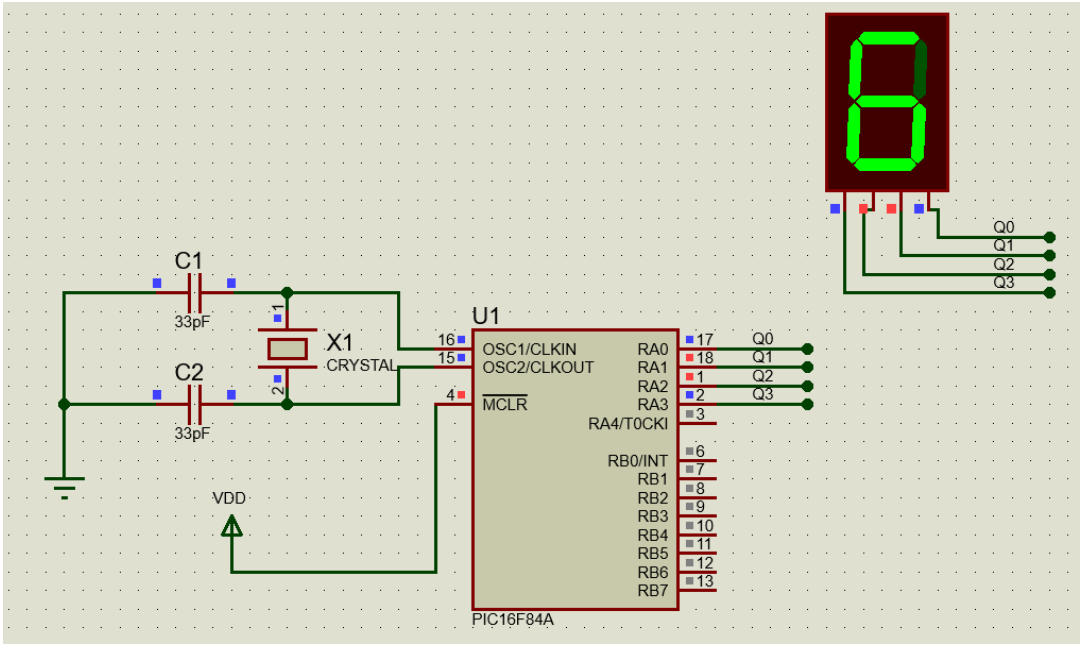

Figure 8. Output value at PORTA

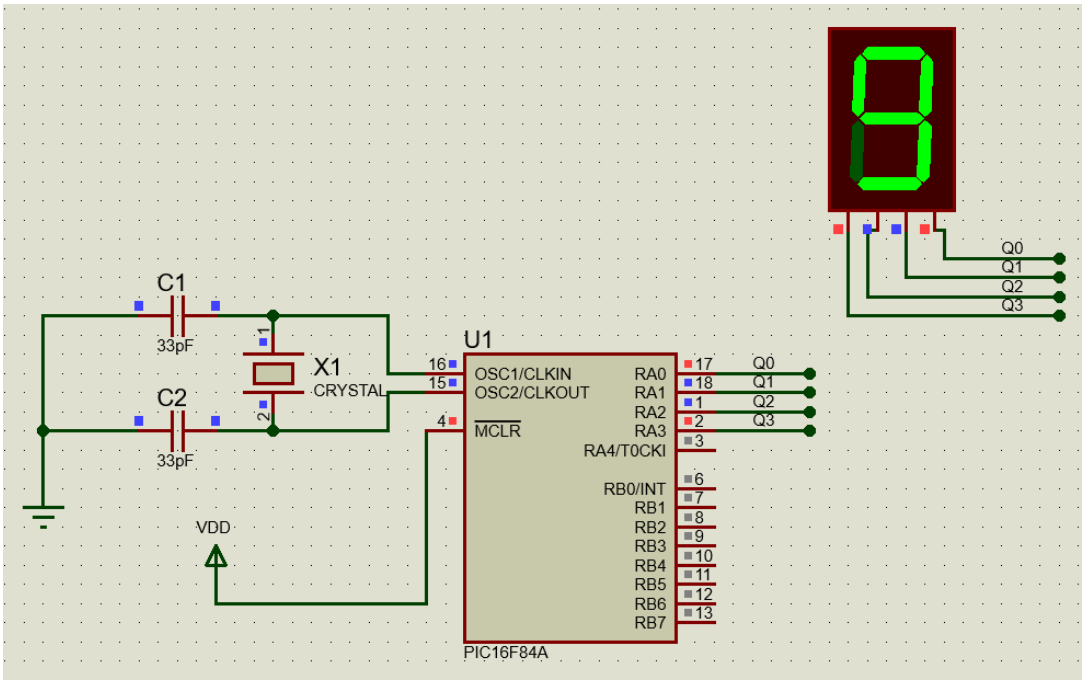

Figure 9.Output value at PORTA

3. Program a PIC16F84A using the QL2006 programmer.

4. Build the circuit using the programmed PIC16F84A and then observe its operation. Demonstrate the circuits operation to the instructor. Present your results in a lab report including a copy of the source codes.

   Those codes are as a attached files with this report. However, they can be consulted at: github.com/ecarrenolozano/ArchitectureEmbeddedSystems2021

3. Control Questions

a. How to write the addition function?

   In C language you could use the operator +, but in assembler you need the ADDWF, or the INCF if your intention is adding1 to one register.

b. What is the method of masking?

   The idea of masking is utilizing some subsets of a register without affecting the remaining bits you are not interested in use. This is a method that allow to treat bits independently.

c. How many methods do you know generating Delay in PIC16F84A?

   Two methods, creating delays with loops in general and using timers.

d. What are these methods?

   Loops: this method is inefficient, and it consist of generate empty operations and “waste time” doing nothing. The time wasted is considered a delay.

   Timer: this method creates interruptions once an internal and precise counter reaches an overflow condition.

4. **Summary**

   For this laboratory practice the main purpose was getting started with the Assembly language for microcontrollers. A variety of circuits were tested in Proteus, the personal computer abstract microcontroller capabilities and it is able to simulate the Instruction Set Architecture of the chosen microcontroller (PIC16F84A) and to generate the expected outputs for each circuit.