

**AVR MICROCONTROLLER**

**LAB MANUAL (P21CS405)**

**DEPARTMENT OF COMPUTER SCIENCE**

**AND ENGINEERING**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Unit** | **Program no.** | **Problem set** |
| Unit-2 | 1 | Write a program to find greatest of three numbers. |
| 2 | Write a program to div two numbers. |
| 3 | Write a program to whether the given number is power of 2 or not. |
| Unit-3 | 4 | Write a program to find the factorial of given positive number. |
| 5 | Write a program to accept two 8 bit numbers from PORTA and POTRB. Multiply two numbers and send the result to PORTC (lower byte) and PORTD (higher byte). |
| 6 | Write a program to monitor the bit 1 of PORTC. If set send ‘Y’ to PORTA else send ‘N’ to PORTB. |
| Unit-4 | 7 | Write a program to convert packed BCD to ASCII. |
| 8 | Write a program to add 10 bytes of data stored starting from $300. Store the sum in R21 and carry in R22. (Use direct addressing mode). |
| 9 | Write a program to count number of odd and even numbers among n bytes of data stored starting from $600 (Use indirect addressing mode). |
| Unit-5 | 10 | Write c program to blink LED connected to 1st pin of PORTC with appropriate delay. |
| 11 | A switch is connected to PORTA. If it is pressed the led should glow. Write C program to perform the above said operation. |
| 12 | Write a program to read the temperature using sensor and display the following message on the monitor according to the condition given below**.** If temperature is greater than 32 display “high**”** else if the temperature is lesser than the 32 display “low”. |

**Program 1. Program to find the largest among three numbers**

/\*

Example- Input:r16=-27 r17=27 r18=-10 (all numbers are decimal)

Output: 1B(Dec 27) at Memory Location 0x0060

\*/

.org 0

.equ largest=0x0060

; load r16,r17 and r18 with three numbers

ldi r16,-27

ldi r17,27

ldi r18,-10

;logic to find larsest number and r16 contains the largest number

cp r16,r17

brge l1

mov r16,r17

l1: cp r16,r18

brge l2

mov r16,r18

; store the largest number in memory location

l2: sts largest,r16

here: jmp here

## Program 2. Write ALP program to divide two numbers.

/\*

Example- Input: r17=-10, r18=2 (all numbers are decimal)

Output: 5(result will be in Hexadecimal) at Memory Location 0x0300

\*/

.org 0

.equ result =0X0300 ;address of result

.def dividend =r17 ;dividend

.def divisor =r18 ;divisor

.def counter =r19 ;loop counter

ldi dividend,10

ldi divisor,2

ldi counter,0

clc

l1:inc counter

sub dividend,divisor

brcc l1

dec counter

; store the result in memory location

sts result,counter

Here: rjmp Here

**Program 3. Write ALP program to check whether the given number is power of 2 or not.**

/\*

Example- Input: R16=128 (decimal)

Output: Finally, if the content of register R17 is 0(zero) then the given number is power of 2, otherwise not.

\*/

.org 0

ldi r16,128 ;load the number into r16

mov r17,16 ;copy the content of r16 to r17

subi r17,1 ;find the previous number of the given number

andi r17,r16 ; perform logical AND on r17 and r16

here:rjmp here

**Program – 4 program to find factorial of a number.**

/\*

Example-Input: r19=5

Output:porta=0x78

\*/

.org 0

; stack initialization

ldi r18,high(ramend)

out sph,r18

ldi r18,low(ramend)

out spl,r18

ldi r19,0x05

ldi r18,0x01

call fact ; call subroutine

; make PORTA as output port to send result

ldi r16,0xff

out ddra,r16

out porta,r18

here: rjmp here

; subroutine

fact: nop

loop: mul r18,r19

mov r18,r0

dec r19

brne loop

ret ; return to main code

## Program – 5 Write ALP program to accept two 8 bit numbers from PORTA and POTRB. Multiply two numbers and send the result to PORTC (lower byte) and PORTD (higher byte).

/\*

Example-Input: PORTA=0x45 and PORTB=0x20

Output: PORTD=0x08(higher byte) and PORTC=0xA0(lower byte)

\*/

.org 0

; configure PORTA,PORTB,PORTC and PORTD as output port

ldi r16,0xff

out ddra,r16

out ddrb,r16

out ddrc,r16

out ddrd,r16

; load the forst number to PORTA and second number to PORTB

ldi r16,0x45

out porta,r16

ldi r17,0x20

out portb,r17

; configure PORTA and PORTB as input port

ldi r20,0x00

out ddra,r20

out ddrb,r20

; read the two numbers from PINA and PINB

in r24,pina

in r25,pinb

; multiply two numbers

mul r24,r25

; send result to PORT C and PORTD

out portc,r0

out portd,r1

here:rjmp here

**Program –6 Write ALP program to monitor the bit 1 of PORTC. If set send ‘Y’ to PORTA else send ‘N’ to PORTB.**

/\*

Example 1- Input: PORTC=0x20(00100000)

Output: PORTB=0x4E(ASCII for ‘N’)

Example 2- Input: PORTC=0x23(00100011)

Output: PORTB=0x59(ASCII for ‘Y’)

\*/

.org 0

; configure PORTA,PORTB and PORTC as output port

ldi r16,0XFF

out ddra,r16

out ddrb,r16

out ddrc,r16

; load some value into PORTC

ldi r17,0X20

out portc,r17

; configure pin 1 of PORTC as Input pin

cbi ddrc,1

; skip next instruction if PIN 1 of PORTC is set

sbis pinc,1

rjmp over

; send ‘Y’ to PORTA

ldi r16,'Y'

out porta,r16

rjmp here

; send ‘N’ to PORTB

over: ldi r16,'N'

out portb,r16

here: rjmp here

## Program –7 Write ALP program to convert packed BCD to ASCII.

/\*

Example-Input:r20=0x48

Output: memory location 0x301 = 0x34 and 0x302=0x38

\*/

.equ packedbcd=0x300

.equ ascii1=0x301

.equ ascii2=0x302

; initialize stack

ldi r16,high(ramend)

out sph,r16

ldi r16,low(ramend)

out spl,r16

; store packed BCD at location 0x300

ldi r20,0x38

sts packedbcd,r20

call conversion ; call subroutine

here:jmp here

; subroutine

Conversion:

lds r20,packedbcd ; load packedBCD into R20

mov r21,r20

ldi r22,0x30

andi r20,0x0f ; unpack BCD and obtain lowerbyte number

add r20,r22 ; convert unpacked BCD into ASCII

sts ascii1,r20 ; store first ASCII value into memory location 0x301

swap r21

andi r21,0x0f ; unpack BCD and obtain higher byte number

add r21,r22 ; convert unpacked BCD into ASCII

sts ascii2,r21 ;s tore first ASCII value into memory location 0x302

ret ; return to main program

## Program – 8 Write ALP program to add 10 bytes of data stored starting from $300. Store the sum in R21 and carry in R22. (Use direct addressing mode).

/\*

Example: 0x300=10, 0x301=20, 0x302=30, 0x303=40, 0x304=50, 0x305=60, 0x306=70, 0x307=80, 0x308=90 and 0x309=100.(all numbers are in decimal)

Output: R21=0x226 and R22=1(carry value)

\*/

.include "m32def.inc"

.org 0

; store 10 bytes of data starting from location 0x300 to 0x309

ldi r16,10

sts 0x0300,r16

ldi r16,20

sts 0x0301,r16

ldi r16,30

sts 0x0302,r16

ldi r16,40

sts 0x0303,r16

ldi r16,50

sts 0x0304,r16

ldi r16,60

sts 0x0305,r16

ldi r16,70

sts 0x0306,r16

ldi r16,80

sts 0x0307,r16

ldi r16,90

sts 0x0308,r16

ldi r16,100

sts 0x0309,r16

ldi r21,0

ldi r22,0

; read each byte from location 0x300 to 0x309 and add each value to r21

lds r20,0x0300

add r21,r20

lds r20,0x0301

add r21,r20

lds r20,0x0302

add r21,r20

lds r20,0x0303

add r21,r20

lds r20,0x304

add r21,r20

lds r20,0x0305

add r21,r20

lds r20,0x0306

add r21,r20

lds r20,0x0307

add r21,r20

lds r20,0x0308

add r21,r20

lds r20,0x0309

add r21,r20

; read the content of status register

in r19,sreg

lsr r19 ; obtain the content of carry bit

brcc here

ldi r22,1 ; store the carry bit value into r22

here: rjmp here

**Program –9 Write ALP program to count number of odd and even numbers among n bytes of data stored starting from $600 (Use indirect addressing mode).**

/\*

Example- Input:if n=6, 0x600=2, 0x601=3, 0x602=4, 0x603=7, 0x604=10 and 0x605=12

Output: R20 = 2 (odd numbers) and R21 = 4 (even numbers)

Note : n value can be anything

\*/

.org 0

ldi r16,10 ; counter(n)

ldi xl, 0X00 ; lower byte of address

ldi xh, 0X06 ; higher byte of address

ldi r20,0 ; odd number counter

ldi r21,0 ; even number counter

; store n=6 bytes of data from memory location 0x600 to 0x605

ldi r17,2

st x+,r17

ldi r17,3

st x+,r17

ldi r17,4

st x+,r17

ldi r17,7

st x+,r17

ldi r17,10

st x+,r17

ldi r17,12

st x+,r17

; initialize the register x with address 0x600

ldi xl, 0X00 ; lower byte of address

ldi xh, 0X06 ; higher byte of address

; obtain each byte from memory location 0x600 to 0x605

l1: ld r18,x+

ror r18 ; obtain the last bit of each byte

brcs l2 ; jump to l2 if the last bit is 1

inc r21 ; increment odd counter

rjmp next

l2: inc r20 ; increment even counter

next: dec r16

brne l1

here: rjmp here

**Program 10. Write C program to blink LED connected to 1st pin of PORTC with appropriate delay.**

#include <avr/io.h>

#include <util/delay.h> //it has delay function

int main(void)

{

DDRC=0b00000001;

while(1)

  {

     PORTC=0b00000001;

  \_delay\_ms(1000); /\*so that LED would remain in ON state for 1 sec and then  turns off  for  1 sec.  \*/

     PORTB=0b00000000;

     \_delay\_ms(1000);

  }

}

**Program 11. A switch is connected to PORTA. If it is pressed the led should glow. Write C program to perform the above said operation.**

**Program 12. Write a program to read the temperature using sensor and display the following message on the monitor according to the condition given below. If temperature is greater than 32 display “high” else if the temperature is lesser than the 32 display “low”.**