**LAB REPORT**

**IT3280E– 152049– Assembly Language and Computer Architecture Lab**

**Lab 10: Peripheral Devices**

**Assignment 1:**

*Implement the program in Home Assignment 1, change the values displayed on the LEDs to last two digits of StudentID.*

* Source code:

|  |
| --- |
| .data  student\_id: .asciz "20235977"  ascii\_msg: .asciz "\nInput: "  output\_msg: .asciz "\nOutput: "  .eqv SEVENSEG\_LEFT 0xFFFF0011 # Address of the LED on the left  # Bit 0 = segment a  # Bit 1 = segment b  # ...  # Bit 7 = dot sign  .eqv SEVENSEG\_RIGHT 0xFFFF0010 # Address of the LED on the right  .text  main:  # Display last 2 digits  la t0, student\_id  li t1, 6  add t0, t0, t1  lb t2, 0(t0)  lb t3, 1(t0)  addi a0, t2, -48  jal SHOW\_7SEG\_LEFT # Show the result  addi a0, t3, -48  jal SHOW\_7SEG\_RIGHT # Show the result  exit:  li a7, 10  ecall  end\_main:  SHOW\_7SEG\_LEFT:  li t0, SEVENSEG\_LEFT # Assign port's address  sb a0, 0(t0) # Assign new value  jr ra  SHOW\_7SEG\_RIGHT:  li t0, SEVENSEG\_RIGHT # Assign port's address  sb a0, 0(t0) # Assign new value  jr ra |

* Result:

A screenshot of a computer

Description automatically generated

**Assignment 2:**

*Write a program that lets user enter a character from the keyboard and the program will print the last two digits of the ASCII code of the characters.*

* Source code:

|  |
| --- |
| .data  ascii\_msg: .asciz "Enter a character: " # Message prompting user input  output\_msg: .asciz "\nLast two digits of ASCII code: "  buffer: .space 2 # Space for input character  .text  main:  # Prompt user for input  la a0, ascii\_msg  li a7, 4  ecall  # Read a character from the keyboard  li a7, 8  la a0, buffer  li a1, 2  ecall  # Extract ASCII value of input character  la t0, buffer  lb t4, 0(t0)    # Calculate last two digits of ASCII code  li t1, 100  rem t0, t4, t1  # Print output message  la a0, output\_msg  li a7, 4  ecall  # Print last two digits of ASCII code  li a7, 1  mv a0, t0  ecall  # Exit program  li a7, 10  ecall |

* Result:

A black text on a white background

Description automatically generated

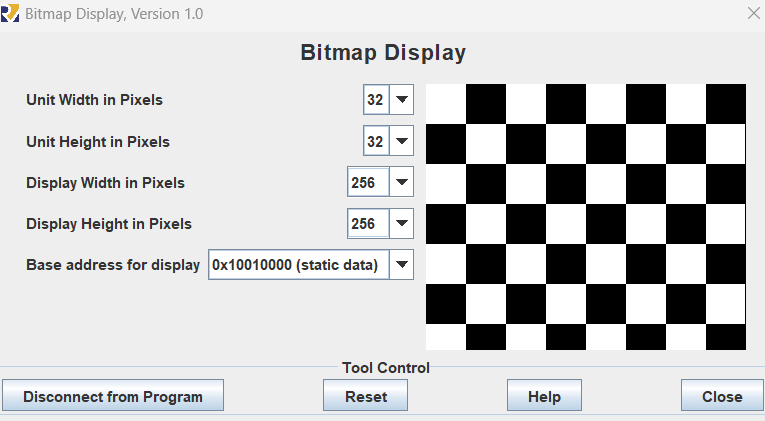
**Assignment 3:**

*Implement the program in Home Assignment 2, and then update the code so that it can  
draw a chess board.*

* Source code:

|  |
| --- |
| .eqv MONITOR\_SCREEN 0x10010000  .eqv RED 0x00FF0000  .eqv GREEN 0x0000FF00  .eqv BLUE 0x000000FF  .eqv WHITE 0x00FFFFFF  .eqv YELLOW 0x00FFFF00  .text  li t1, 0x10010180  li a0, MONITOR\_SCREEN  li t0, WHITE    start:  li t2, 4  j cont1  loop1:  li t2, 4  addi a0,a0 4  cont1:  sw t0, 0(a0)  addi a0,a0,8  bgt a0,t1, exit  addi t2,t2,-1  beqz t2,loop2    j cont1    loop2:  li t2,4  addi a0,a0,4  cont2:  sw t0, 0(a0)  bgt a0,t1, exit  addi t2,t2,-1  beqz t2,loop1  addi a0,a0,8  j cont2    exit:  li a7,10  ecall |

* Result:



**Assignment 4:**

*Implement the program in Home Assignment 3, then update the code so that it can be*

*executed as follows:*

*Enter a lowercase character => Display the corresponding uppercase character.*

*Enter an uppercase character => Display the corresponding lowercase character.*

*Enter a digit => Display the same digit*

*Enter another character => Display “\*”*

*The program will be exited if “exit” is entered.*

* Source code:

|  |
| --- |
| .eqv KEY\_CODE 0xFFFF0004 # ASCII code from keyboard, 1 byte  .eqv KEY\_READY 0xFFFF0000 # =1 if has a new keycode ?  # Auto clear after lw    .eqv DISPLAY\_CODE 0xFFFF000C # ASCII code to show, 1 byte  .eqv DISPLAY\_READY 0xFFFF0008 # =1 if the display has already to do  # Auto clear after sw  .eqv LOWERCASE\_MIN, 0x61 # ASCII 'a'  .eqv LOWERCASE\_MAX, 0x7A # ASCII 'z'  .eqv UPPERCASE\_MIN, 0x41 # ASCII 'A'  .eqv UPPERCASE\_MAX, 0x5A # ASCII 'Z'  .eqv DIGIT\_MIN, 0x30  .eqv DIGIT\_MAX, 0x39  .data  exit: .asciz "exit"  asterisk: .byte 0x2A # ASCII code for '\*'  .text  la s2, exit  li t4, 0 # counter matching  li a0, KEY\_CODE  li a1, KEY\_READY  li s0, DISPLAY\_CODE  li s1, DISPLAY\_READY  loop:  WaitForKey:  lw t1, 0(a1) # t1 = [a1] = KEY\_READY  beq t1, zero, WaitForKey # if t1 == 0 then Polling    ReadKey:  lw t0, 0(a0) # t0 = [a0] = KEY\_CODE    WaitForDis:  lw t2, 0(s1) # t2 = [s1] = DISPLAY\_READY  beq t2, zero, WaitForDis # if t2 == 0 then polling    check\_exit:  lb t3, 0(s2)  beq t3, zero, ShowKey  bne t0, t3, reset  addi t4, t4, 1  addi s2, s2, 1  li t3, 4  beq t4, t3, program\_exit  j loop  reset:  la s2, exit  li t4, 0  j check    check:  li t1, LOWERCASE\_MIN # t1 = 'a'  li t2, LOWERCASE\_MAX # t2 = 'z'  blt t0, t1, check\_upper # If t0 < 'a', check if it's a digit  ble t0, t2, uppercase # If t0 between 'a' and 'z', it's lowercase  check\_upper:  li t1, UPPERCASE\_MIN # t1 = 'A'  li t2, UPPERCASE\_MAX # t2 = 'Z'  blt t0, t1, check\_digit # If t0 < 'A', check if it's a digit  ble t0, t2, lowercase # If t0 between 'A' and 'Z', it's uppercase    check\_digit:  li t1, DIGIT\_MIN # t1 = 0  li t2, DIGIT\_MAX # t2 = 9  blt t0, t1, display\_asterisk  ble t0, t2, ShowKey  display\_asterisk:  la t0, asterisk  lb t0, 0(t0)  j ShowKey  uppercase:  addi t0, t0, -32 # change input key  j ShowKey  lowercase:  addi t0, t0, 32 # change input key  j ShowKey    ShowKey:  sw t0, 0(s0) # show key  j loop  program\_exit:  li a7, 10  ecall |

* Result:

A screenshot of a computer

Description automatically generated

# Assignment 5:

*Write a program that allows the user to enter 2 points with coordinates (x1, y1) and (x2,y2) (x1 is different from x2 and y1 is different from y2), draw and color a rectangle with 2 corners being the 2 entered points with a red border 1 unit wide and a green background. For example, with (x1, y1) = (3, 3) and (x2, y2) = (18, 11), or (x1, y1) = (3, 11) and (x2,y2) = (18, 3), we will have the result as the following figure.*

A grid with red and green squares

Description automatically generated