**LAB REPORT**

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

**Lab 11: Interrupts & IO programming**

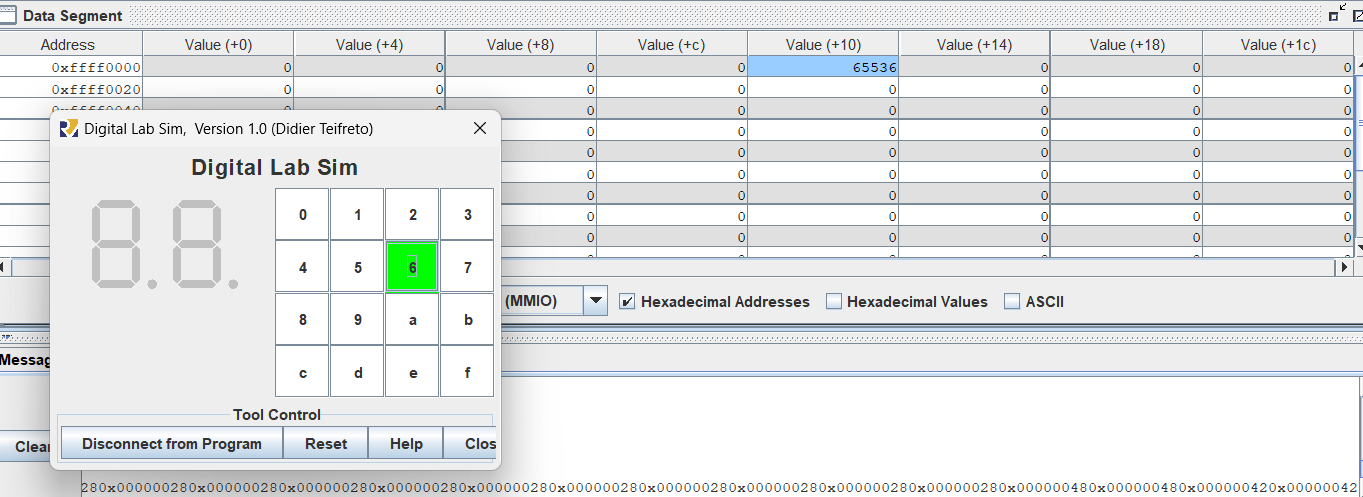
**Assignment 1:**

*Create a new project, type in, and build the program of Home Assignment 1. Run the program step by step to understand each line of the source code. Upgrade the source code so that it could detect all 16 key buttons, from 0 to F*

* Source code:

|  |
| --- |
| # ------------------------------------------------------  # col 0x1 col 0x2 col 0x4 col 0x8  # row 0x1 0 1 2 3  # 0x11 0x21 0x41 0x81  # row 0x2 4 5 6 7  # 0x12 0x22 0x42 0x82  # row 0x4 8 9 a b  # 0x14 0x24 0x44 0x84  # row 0x8 c d e f  # 0x18 0x28 0x48 0x88  # ------------------------------------------------------  # Command row number of hexadecimal keyboard (bit 0 to 3)  # Eg. assign 0x1, to get key button 0,1,2,3  # assign 0x2, to get key button 4,5,6,7  # NOTE must reassign value for this address before reading,  # eventhough you only want to scan 1 row  .eqv IN\_ADDRESS\_HEXA\_KEYBOARD 0xFFFF0012  # Receive row and column of the key pressed, 0 if not key pressed  # Eg. equal 0x11, means that key button 0 pressed.  # Eg. equal 0x28, means that key button D pressed.  .eqv OUT\_ADDRESS\_HEXA\_KEYBOARD 0xFFFF0014  .text  main:  li t1, IN\_ADDRESS\_HEXA\_KEYBOARD  li t2, OUT\_ADDRESS\_HEXA\_KEYBOARD  li t3, 0x01 # check row 4 with key C, D, E, F  polling:  sb t3, 0(t1) # must reassign expected row  lb a0, 0(t2) # read scan code of key button  beqz a0, no\_key\_pressed # check if key is pressed  print:  li a7, 34 # print integer (hexa)  ecall  sleep:  li a0, 50 # sleep 50ms  li a7, 32  ecall  no\_key\_pressed:  slli t3, t3, 1 # shift left to get the next row (0x02, 0x04, 0x08)  li t4, 0x10 # maximum value for row is 0x08 (row 8)  bne t3, t4, polling # if not reached row 8, continue scanning  li t3, 0x01 # reset to start  back\_to\_polling:  j polling # continue polling |

* Result:



A screenshot of a computer

Description automatically generated

**Assignment 2:**

*Create a new project, type in, and build the program of Home Assignment 2. Run the program step by step to understand each line of the source code.*

* Source code:

|  |
| --- |
| .eqv IN\_ADDRESS\_HEXA\_KEYBOARD 0xFFFF0012  .data  message: .asciz "Someone's presed a button.\n"  # -----------------------------------------------------------------  # MAIN Procedure  # -----------------------------------------------------------------  .text  main:  # Load the interrupt service routine address to the UTVEC register  la t0, handler  csrrs zero, utvec, t0    # Set the UEIE (User External Interrupt Enable) bit in UIE register  li t1, 0x100  csrrs zero, uie, t1 # uie - ueie bit (bit 8)  # Set the UIE (User Interrupt Enable) bit in USTATUS register  csrrsi zero, ustatus, 0x1 # ustatus - enable uie (bit 0)    # Enable the interrupt of keypad of Digital Lab Sim  li t1, IN\_ADDRESS\_HEXA\_KEYBOARD  li t3, 0x80 # bit 7 = 1 to enable interrupt  sb t3, 0(t1)    # ---------------------------------------------------------  # No-end loop, main program, to demo the effective of interrupt  # ---------------------------------------------------------  loop:  nop  # Delay 10ms  li a7, 32  li a0, 10  ecall  nop  j loop  end\_main:    # -----------------------------------------------------------------  # Interrupt service routine  # -----------------------------------------------------------------  handler:  # ebreak # Can pause the execution to observe registers  # Saves the context  addi sp, sp, -8  sw a0, 0(sp)  sw a7, 4(sp)    # Handles the interrupt  # Shows message in Run I/O  li a7, 4  la a0, message  ecall    # Restores the context  lw a7, 4(sp)  lw a0, 0(sp)  addi sp, sp, 8    # Back to the main procedure  uret |

* Result:

|  |  |
| --- | --- |
|  |  |

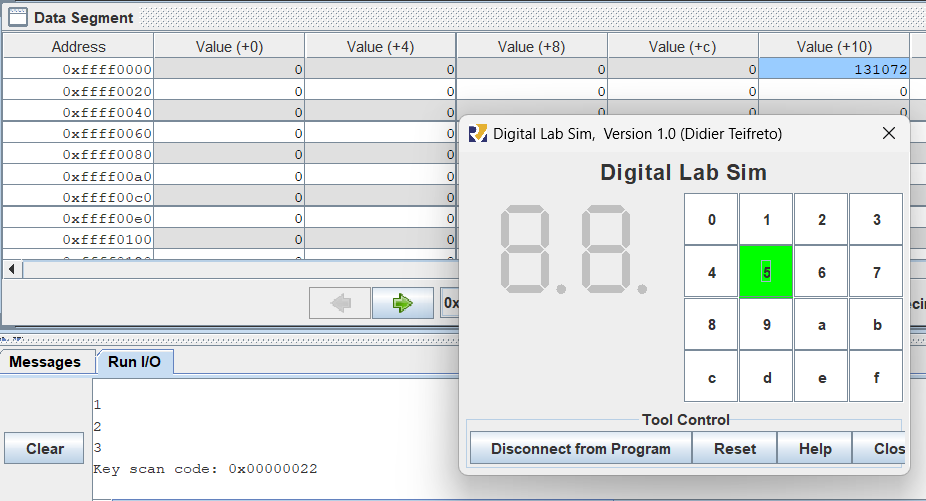
**Assignment 3:**

*Create a new project, type in, and build the program of Home Assignment 3. Run the program step by step to understand each line of the source code. Upgrade the source code so that it could detect all 16 key buttons, from 0 to F.*

* Source code:

|  |
| --- |
| .eqv IN\_ADDRESS\_HEXA\_KEYBOARD 0xFFFF0012  .eqv OUT\_ADDRESS\_HEXA\_KEYBOARD 0xFFFF0014  .data  message: .asciz "Key scan code: "  # -----------------------------------------------------------------  # MAIN Procedure  # -----------------------------------------------------------------  .text  main:  # Load the interrupt service routine address to the UTVEC register  la t0, handler  csrrs zero, utvec, t0    # Set the UEIE (User External Interrupt Enable) bit in UIE register  li t1, 0x100  csrrs zero, uie, t1  # Set the UIE (User Interrupt Enable) bit in USTATUS register  csrrsi zero, ustatus, 0x1    # Enable the interrupt of keypad of Digital Lab Sim  li t1, IN\_ADDRESS\_HEXA\_KEYBOARD  li t3, 0x80 # bit 7 = 1 to enable interrupt  sb t3, 0(t1)    # ---------------------------------------------------------  # Loop to print a sequence numbers  # ---------------------------------------------------------  xor s0, s0, s0 # count = s0 = 0  loop:  addi s0, s0, 1 # count = count + 1  prn\_seq:  addi a7, zero, 1  mv a0, s0 # Print auto sequence number  ecall  addi a7, zero, 11  li a0, '\n' # Print EOL  ecall  sleep:  addi a7, zero, 32  li a0, 300 # Sleep 300 ms  ecall  j loop  end\_main:    # -----------------------------------------------------------------  # Interrupt service routine  # -----------------------------------------------------------------  handler:  # Save context  addi sp, sp, -16  sw a0, 0(sp)  sw a7, 4(sp)  sw t1, 8(sp)  sw t2, 12(sp)    prn\_msg:  addi a7, zero, 4  la a0, message  ecall  get\_key\_code:  li t2, 0x01 # start with row 1  row\_scan:  li t1, IN\_ADDRESS\_HEXA\_KEYBOARD  sb t2, 0(t1) # Activate row  li t1, OUT\_ADDRESS\_HEXA\_KEYBOARD  lb a0, 0(t1) # read key scan code  bnez a0, prn\_key\_code # if non-zero, key is detected  slli t2, t2, 1 # move to the next row  li t3, 0x10 # 0x10 for 4 rows (0x01, 0x02, 0x04, 0x08)  blt t2, t3, row\_scan  li t2, 0x01  j clear\_interrupt # no key detected  prn\_key\_code:  li a7, 34  ecall  li a7, 11  li a0, '\n' # Print EOL  ecall    wait\_key\_release:  li t1, OUT\_ADDRESS\_HEXA\_KEYBOARD  lb a0, 0(t1) # Read the current state  bnez a0, wait\_key\_release # Wait until key is released  clear\_interrupt:  li t1, IN\_ADDRESS\_HEXA\_KEYBOARD  li t3, 0x80 # Re-enable interrupt bit  sb t3, 0(t1)  j restore\_context    restore\_context:  # Restore context  lw t2, 12(sp)  lw t1, 8(sp)  lw a7, 4(sp)  lw a0, 0(sp)  addi sp, sp, 16    # Back to the main procedure  uret |

* Result:



A screenshot of a computer

Description automatically generated