**BÁO CÁO THỰC HÀNH**

**KIẾN TRÚC MÁY TÍNH LAB**

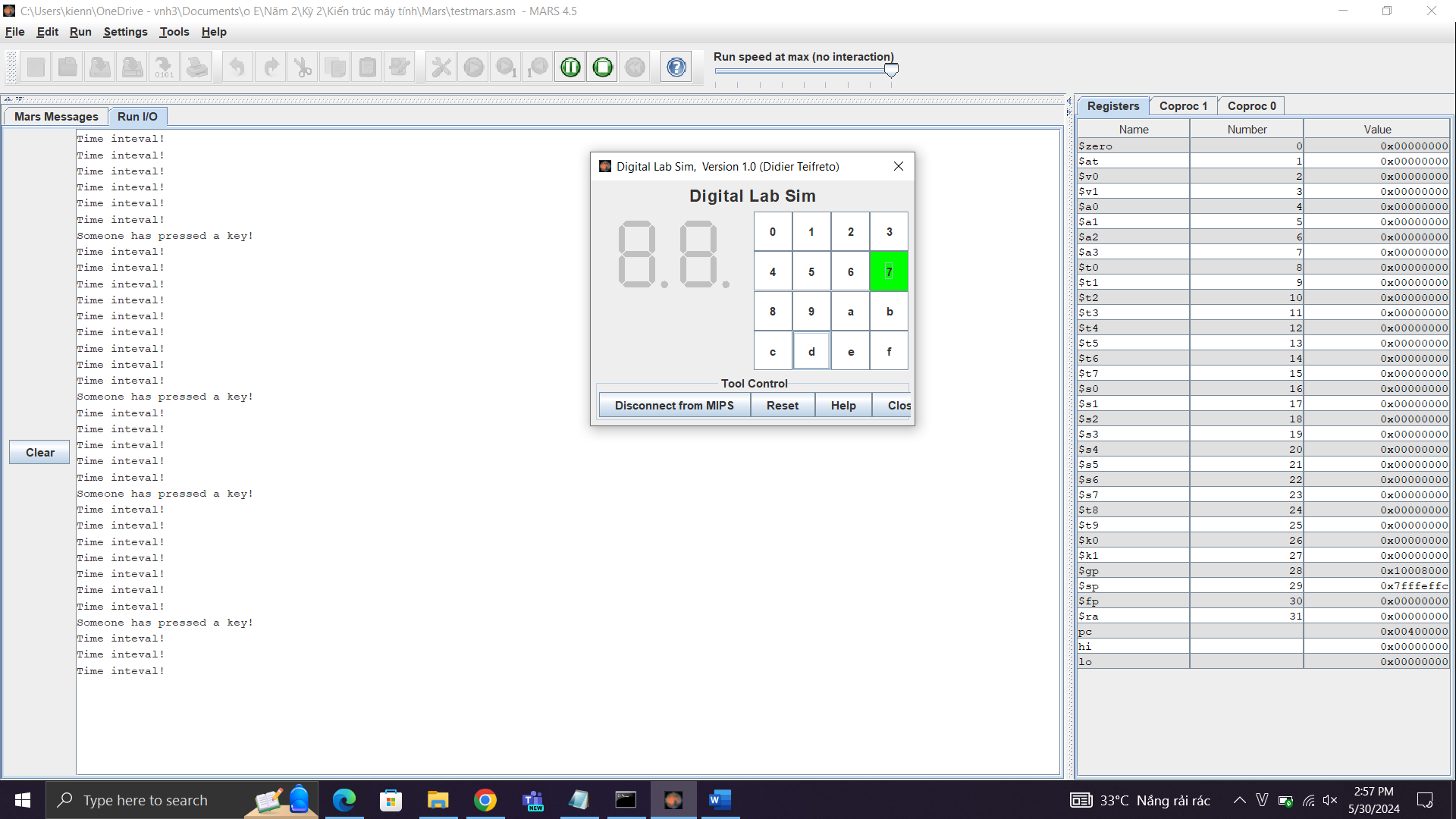
**Assignment 4:**

**Code:**

|  |
| --- |
| .eqv IN\_ADRESS\_HEXA\_KEYBOARD 0xFFFF0012  .eqv COUNTER 0xFFFF0013 # Time Counter  .eqv MASK\_CAUSE\_COUNTER 0x00000400 # Bit 10: Counter interrupt  .eqv MASK\_CAUSE\_KEYMATRIX 0x00000800 # Bit 11: Key matrix interrupt  .data  msg\_keypress: .asciiz "Ai do da bam cac nut nay!\n"  msg\_counter: .asciiz "Time inteval!\n"  #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  # MAIN Procedure  #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  .text  main:  #---------------------------------------------------------  # Enable interrupts you expect  #---------------------------------------------------------  # Enable the interrupt of Keyboard matrix 4x4 of Digital Lab Sim  li $t1, IN\_ADRESS\_HEXA\_KEYBOARD  li $t3, 0x80 # bit 7 = 1 to enable  sb $t3, 0($t1)  # Enable the interrupt of TimeCounter of Digital Lab Sim  li $t1, COUNTER  sb $t1, 0($t1)  #---------------------------------------------------------  # Loop an print sequence numbers  #---------------------------------------------------------  Loop:  nop  nop  nop  sleep:  addi $v0,$zero,32 # BUG: must sleep to wait for Time Counter  li $a0, 400 # sleep 0,4s  syscall  nop # WARNING: nop is mandatory here.  b Loop  end\_main:  #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  # GENERAL INTERRUPT SERVED ROUTINE for all interrupts  #~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~  .ktext 0x80000180  IntSR:  #--------------------------------------------------------  # Temporary disable interrupt  #--------------------------------------------------------  dis\_int:  li $t1, COUNTER # BUG: must disable with Time Counter  sb $zero, 0($t1)  # no need to disable keyboard matrix interrupt  #--------------------------------------------------------  # Processing  #--------------------------------------------------------  get\_caus:  mfc0 $t1, $13 # $t1 = Coproc0.cause  IsCount:  li $t2, MASK\_CAUSE\_COUNTER # if Cause value confirm Counter..  and $at, $t1,$t2  beq $at,$t2, Counter\_Intr  IsKeyMa:  li $t2, MASK\_CAUSE\_KEYMATRIX # if Cause value confirm Key..  and $at, $t1,$t2  beq $at,$t2, Keymatrix\_Intr  others:  j end\_process # other cases  Keymatrix\_Intr:  li $v0, 4 # Processing Key Matrix Interrupt  la $a0, msg\_keypress  syscall  j end\_process  Counter\_Intr:  li $v0, 4 # Processing Counter Interrupt  la $a0, msg\_counter  syscall  j end\_process  end\_process:  mtc0 $zero, $13 # Must clear cause reg  en\_int:  #--------------------------------------------------------  # Re-enable interrupt  #--------------------------------------------------------  li $t1, COUNTER  sb $t1, 0($t1)  #--------------------------------------------------------  # Evaluate the return address of main routine  # epc <= epc + 4  #--------------------------------------------------------  next\_pc:  mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc  addi $at, $at, 4 # $at = $at + 4 (next instruction)  mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at  return:  eret # Return from exception |

**Kết quả:**

-Lần lượt ấn các phím 4,5,6,7 (tổng số phím là 4) kết quả sẽ như này:



**Assignment 5:**

**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 MASK\_CAUSE\_KEYBOARD 0x0000034 # Keyboard Cause  .text  li $k0, KEY\_CODE  li $k1, KEY\_READY  li $s0, DISPLAY\_CODE  li $s1, DISPLAY\_READY  loop:  nop  WaitForKey:  lw $t1, 0($k1) # $t1 = [$k1] = KEY\_READY  beq $t1, $zero, WaitForKey # if $t1 == 0 then Polling  MakeIntR:  teqi $t1, 1 # if $t0 = 1 then raise an Interrupt  j loop  #---------------------------------------------------------------  # Interrupt subroutine  #---------------------------------------------------------------  .ktext 0x80000180  get\_caus:  mfc0 $t1, $13 # $t1 = Coproc0.cause  IsCount:  li $t2, MASK\_CAUSE\_KEYBOARD # if Cause value confirm  Keyboard:  and $at, $t1,$t2  beq $at,$t2, Counter\_Keyboard  j end\_process  Counter\_Keyboard:  ReadKey:  lw $t0, 0($k0) # $t0 = [$k0] = KEY\_CODE  WaitForDis:  lw $t2, 0($s1) # $t2 = [$s1] = DISPLAY\_READY  beq $t2, $zero, WaitForDis # if $t2 == 0 then Polling  Encrypt:  addi $t0, $t0, 1 # change input key  ShowKey:  sw $t0, 0($s0) # show key  nop  end\_process:  next\_pc:  mfc0 $at, $14 # $at <= Coproc0.$14 = Coproc0.epc  addi $at, $at, 4 # $at = $at + 4 (next instruction)  mtc0 $at, $14 # Coproc0.$14 = Coproc0.epc <= $at  return:  eret # Return from exception |

**Kết quả:**

