**BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH**

**LAB 11**

Họ và tên: Phạm Vân Anh

MSSV: 20214988

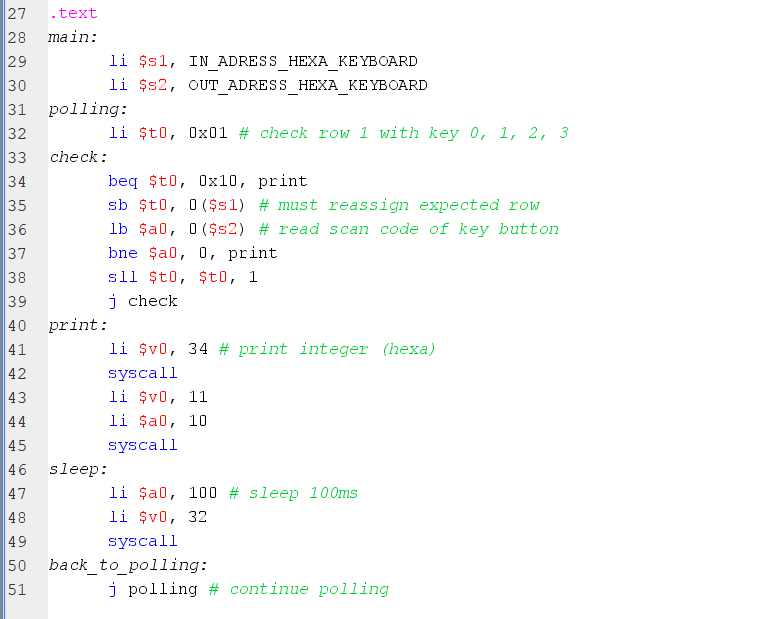
Mã lớp: 139365

***ASSIGNMENT 1:***

1. **Code**

**A screenshot of a computer program

Description automatically generated with medium confidence**



*Thực hiện gõ chương trình vào công cụ MARS*

1. **Giải thích**

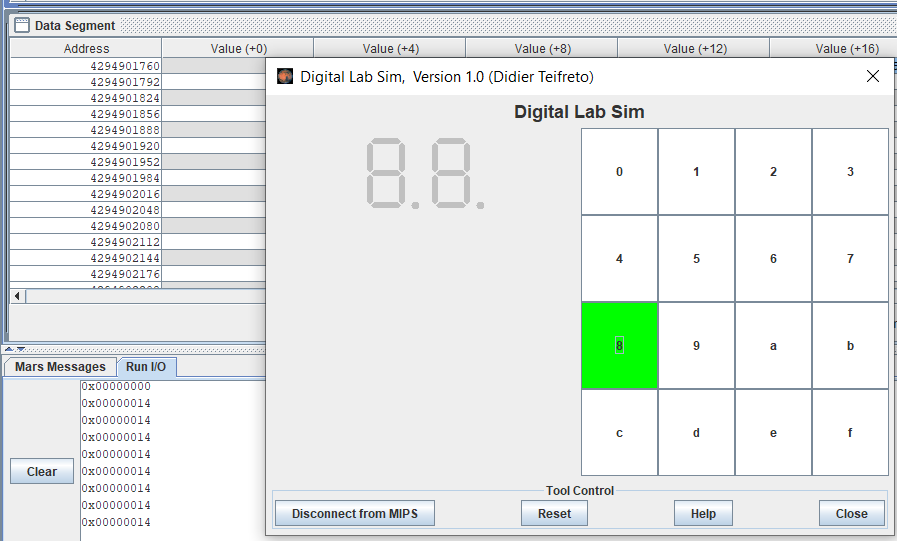
* Chương trình in ra màn hình mã scan của số được bấm trên Digital Lab Sim qua ma trận sau

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0x10 | 0x20 | 0x30 | 0x40 |
| 0x01 | 0 | 1 | 2 | 3 |
| 0x02 | 4 | 5 | 6 | 7 |
| 0x03 | 8 | 9 | A | B |
| 0x04 | C | D | E | F |

* IN\_ADDRESS\_HEXA\_KEYBOARD 0xFFFF0012: địa chỉ đầu vào
* OUT\_ADDRESS\_HEXA\_KEYBOARD 0Xffff0014: địa chỉ đầu ra nhận giá trị hàng và cột của phím được bấm.
* Hàm *main:*
* Gán địa chỉ vào các thanh ghi $s1 và $s2 tương ứng
* Bắt đầu vòng lặp gán nhãn *polling* (kiểm tra)
* Hàm *polling:* vòng lặp kiểm tra từng phần tử trong hàng
* Hàm *check:* vòng lặp kiểm tra từng cột
* Nếu $t0 = 0x10 (đã kiểm tra hết 4 cột) 🡪 *print*
* Nếu không, nó ghi giá trị trong thanh ghi $t0 vào địa chỉ được lưu trong thanh ghi $t1 🡪 gán hàng tiếp cho việc kiểm tra
* Đọc giá trị từ địa chỉ lưu trong $t2 vào $t0 🡪 đọc mã quét
* Nếu $a0 != 0 (đã nhận được giá trị từ bàn phím) 🡪 print
* Nếu không, nó dịch sang trái giá trị trong $t0 lên 1 (sang cột tiếp theo) 🡪 *check*
* Hàm *print:* in ra giá trị địa chỉ nút được bấm

1. Kết quả

* Thử ấn số 8(0x14)



* Thử ấn b

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

***ASSIGNMENT 2:***

1. **Code**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer program

Description automatically generated with low confidence**

*Thực hiện gõ chương trình vào công cụ MARS*

1. **Giải thích**

* Chương trình minh họa cho chương trình ngắt trên Digital Lab Sim
* Khi chương trình nhận tín hiệu ngắt, tự động nhảy tới địa chỉ *.ktext 8000180* để thực hiện chương trình con tự động ngắt. Sau khi thực hiện chương trình con, quay lại chương trình chính bằng cách nhảy tới địa chỉ của thanh ghi $14 (thanh ghi $co) nhưng lệnh tại địa chỉ của thanh ghi $14 đã thực hiện xong nên phải +4 để thực hiện lệnh tiếp theo.
* Hàm *main:*
* Kích hoạt phím ngắt bằng cách ghi giá trị 0x80 vào thanh ghi $t3 và lưu vào địa chỉ được lưu trong $t1
* Hàm  *loop:* Vòng lặp chờ sự kích hoạt ngắt từ bàn phím. Khi một nút được bấm (nhận được tín hiệu) 🡪 xử lí ngắt (Interript Servica Routine)

1. **Kết quả**

* Khi bấm số 8:

A screenshot of a computer

Description automatically generated



***ASSIGNMENT 3:***

1. **CodeA screenshot of a computer program

   Description automatically generated with medium confidence** **A screenshot of a computer program

   Description automatically generated with medium confidence** A screenshot of a computer program

   Description automatically generated with medium confidence A screenshot of a computer code

   Description automatically generated with low confidence

*Thực hiện gõ chương trình vào công cụ MARS*

1. **Giải thích**

* Chương trình đếm và in ra màn hình các số mà ta bấm trên ma trận phím sử dụng ngắt.
* Chương trình có 2 phần chính là .text để đếm và in ra màn hình, .ktext 0x80000180 xử lý chính khi gặp ngắt (khi bấm 1 phím trên bàn phím)
* Khởi tạo IN\_ADDRESS 0xFFFF0012 và OUT\_ADDRESS 0xFFFF0014, gán IN\_ADDRESS VÀO $t1
* $s0 là biến đếm. Vòng lặp *loop* tăng đếm thêm 1, sau đó in ra màn hình 🡪 delay 300ms. Sau khi thực hiện xong ta sẽ nhảy lại về loop
* Sau khi nhận được lệnh ngắt (ấn 1 phím ở trên ma trận), nó nhảy tới địa chỉ 0x80000180 và thực hiện chương trình con ngắt ở đó.
* Lưu $ra, $at, $v0, $a0, $t1, $t3 vào trong stack ở *IntSR* vì ta sẽ có thể thay đổi chúng trong chương trình con 🡪 kết thúc ta trả lại giá trị cho chúng bằng cách pop các phần tử lần lượt ra.
* In ra màn hình thông báo kèm mã của số ta vừa bấm vào. Xác định số vừa bấm vào giống ASSIGNMENT 1.
* Thực hiện lấy địa chỉ quay về thanh ghi $14 vào $at, sau đó cộng thêm 4 và gán ngược lại vào thanh ghi $14 để quay lại chương trình chính và thực hiện lại từ đầu

1. **Kết quả**

* Ấn số 8

A screenshot of a computer

Description automatically generated

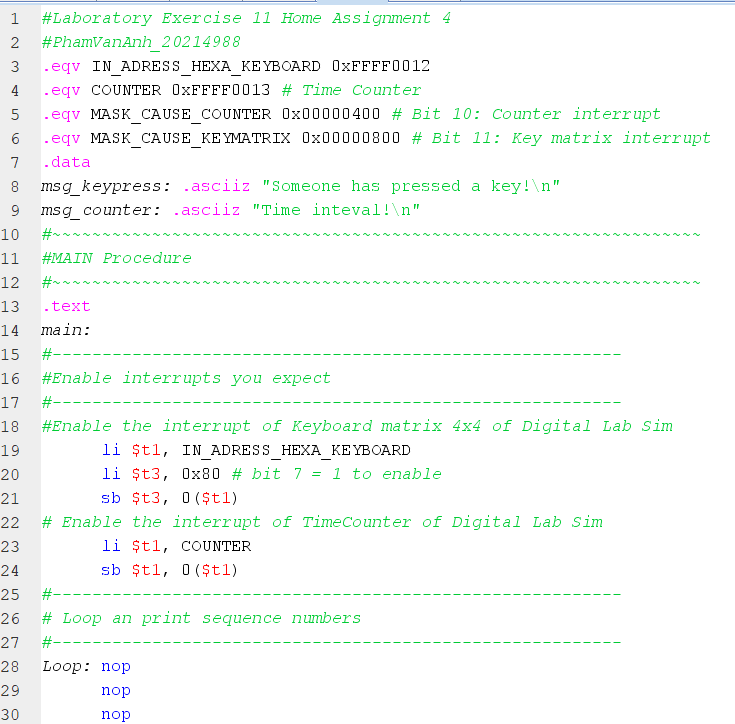
* Ấn b

**A screenshot of a computer

Description automatically generated**

***ASSIGNMENT 4:***

1. **Code**

****

**A picture containing text, font, line, screenshot

Description automatically generated**

**A screenshot of a computer program

Description automatically generated with medium confidence**

**A screenshot of a computer program

Description automatically generated with medium confidence**

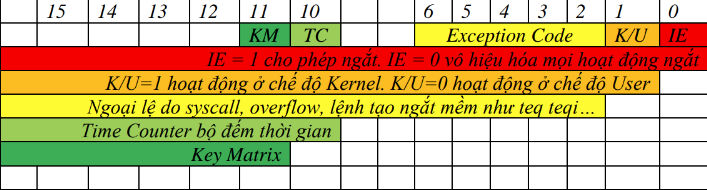
**A picture containing text, font, line, number

Description automatically generated**

*Thực hiện gõ chương trình vào công cụ MARS*

1. Giải thích

* Chương trình mô tả cách minh họa bằng nhiều cách ngắt, ngắt bằng ma trận phím và Bộ đếm thời gian. Nguyên nhân ngắt nằm ở thanh ghi $13 trong CO



* Nếu ngắt bằng Ma trận phím thì thanh ghi $13 có giá trị 0x800 hay bit 10 là 1, ngắt bằng Time Counter thì thanh ghi $13 có giá trị 0x400 hay bit 11 là 1.
* Đầu tiên, ta khởi tạo giá trị MASK\_CAUSE\_COUNTER (ngắt bằng Time Counter) 0x00000400 và MASK\_CAUSE\_KEYMATRIX (ngắt bằng Ma trận phím) 0x00000800
* Hàm *main:*
* Ngắt bằng Ma trận phím
* Gán $t1 = IN\_ADDRESS\_HEXA KEYBOARD, gán $t3 = 0x80 (bit 7 = 1) cho phép chương trình ngắt bằng ma trận phím.
* Gán giá trị của thanh ghi $t3 vào địa chỉ được trỏ bởi thanh ghi $t1
* Ngắt bằng Time Counter
* Gán $t1 = COUNTER, cho phép chương trình ngắt bằng bộ đếm thời gian
* Gán giá trị ở $t1 vào chính địa chỉ được lưu ở $t1
* Sau đó, ta có vòng lặp *Loop* với vài lệnh **nop** và 1 lệnh **sleep,** sau đó quay lại vòng lặp *loop* để đợi cho ngắt xảy ra. Khi quá thời gian đếm hay bấm phím nào trong ma trận, chương trình nhảy tới chương trình con phục vụ ngắt ở *.ktext* 0x80000180.
* Hàm *get\_caus:* lấy ra giá trị của thanh ghi $13 để biết được nguyên nhân ngắt.
* Hàm *IsCount:* check xem có phải nguyên nhắn ngắt là do bộ đếm thời gian hay không
* Gán $t2 = MASK\_CAUSE\_COUNTER 0x00000400 (nguyên nhân ngắt do Time Counter)
* Làm phép toán AND giữa $t1 và $t2, lưu kết quả ra thành $at
* Nếu $at = $t2 = MASK\_CAUSE\_COUNTER 0x00000400   
  🡪 ngắt do bộ đếm thời gian 🡪 in ra kết quả
* Nếu $at != $t2 🡪 check tiếp
* Hàm *IsKeyMa:* check xem có phải nguyên nhân ngắt là do Ma trận phím.
* Thực hiện tương tự như hàm *IsCount*
* Nếu $at = $t2 = MASK\_CAUSE\_KEYMATRIX 0x00000800   
  🡪 ngắt do Ma trận phím 🡪 In ra kết quả
* Nếu $at != $t2 🡪 *end\_process*
* Sau khi in thông báo, ta sẽ xóa dữ liệu thanh ghi $13 bằng cách lưu $13 = 0. Sau đó, ta lưu lại $t1 vào địa chỉ ở $t1 để bật lại cho phép ngắt bằng thời gian.
* Cuối cùng, ta sẽ quay lại chương trình chính bằng cách trích xuất giá trị ở thanh ghi $14, cộng thêm 4 sau đó lưu lại 🡪 dùng lệnh **eret**để quay lại chương trình chính.

1. Kết quả

* Không bấm gì:

A screenshot of a computer

Description automatically generated with medium confidence

* Ấn số 8:



***ASSIGNMENT 5:***

1. **Code**

**A screenshot of a computer code

Description automatically generated with low confidence**

**A screenshot of a computer code

Description automatically generated with low confidence**

**A screenshot of a computer program

Description automatically generated with medium confidence**

*Thực hiện gõ chương trình vào công cụ MARS*

1. Giải thích

* Chương trình minh họa chương trình ngắt với bàn phím. Bàn phím không tự động ngắt mà phải dùng lệnh **teqi** để ngắt.
* Khởi tạo
* KEY\_CODE (0xFFFF0004) giá trị ASCII ký tự được nhập từ bàn phím 🡪 gán $k0
* KEY\_READY 0xFFFF0000 = 1 khi có ký tự được nhập và bằng 0 sau lệnh lw 🡪 gán $k1
  + DISPLAY\_CODE (0xFFFF000C) giá trị ASCII của ký được được in ra màn hình 🡪 gán $s0
  + DISPLAY\_READY (0xFFFF0008) = 1 khi chuẩn bị in và sẽ = 0 sau lệnh sw 🡪 gán $1
  + MASK\_CAUSE\_KEYBOARD: 0x00000034 BIT MASK: nguyên nhân ngắt tạo bởi keyboard.
* Hàm *WaitForKey:* vòng lặp đọc ký tự đầu vào
* Nếu $t1 (KEY\_READY) = 0 🡪 chưa có ký tự nào được nhập  
  🡪 tiếp tục lặp đến khi $t1 = 1 (có ký tự từ bán phím được nhập)
* Sau khi xác nhận có ký tự được nhập, chương trình nhảy đến nhiệm vụ ngắt mềm bằng cách kiểm tra **teqi $1, 1** và nhảy đến địa chỉ ngắt *.ktext 0x80000180.*
* Cơ chế ngắt: khi có một ký tự được nhập từ bàn phím và KEY\_READY = 1
* Check nguyên nhân ngắt bằng cách kiểm tra thanh ghi $13 (được lưu vào $t1), và mặt nạ bit ngắt bàn phím (được lưu vào $t2). Sử dụng phép AND, nếu $1 trùng $t2 🡪 ngắt do bàn phím
* Đọc ký tự được ghi từ bán phím và kiểm tra DISPLAY\_READY có sẵn sàng (=1) hay không (=0)
* Nếu DISPLAY\_READY = 1 🡪 mã hóa ký tự (tăng mã ASCII lên 1) và sẵn sàng in ra màn hình
* Nếu DISPLAY\_READY = 0 🡪 tiếp tục polling bằng vòng lặp
* Kết thúc, ta cập nhật giá trị $14 cộng thêm 4 (địa chỉ của câu lệnh tiếp theo), dùng lệnh **eret** để trở về chương trình chính

1. Kết quả

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated