|  |  |  |
| --- | --- | --- |
|  | **BỘ MÔN KỸ THUẬT MÁY TÍNH – VIỄN THÔNG**  **THỰC TẬP KIẾN TRÚC MÁY TÍNH**  **MMH: COAP416964**  **Tuần:**  **Ngày 13 tháng 4 năm 2018** | |
| **Họ và tên** | Phạm Nguyễn Hoàng Hải | 15119017 |
|  |  |

1. **Cơ sở lý thuyết (20%)**
2. **Mục Tiêu**
   * So sánh 8086 và 8088
   * Chương trình hợp ngữ : Cộng 2 mảng số nguyên
   * Giải thích chương trình hợp ngữ robot
   * Các câu lệnh điều kiện, rẽ nhánh trong hợp ngữ
3. **Lý Thuyết**

* **So sánh 8086 và 8088**

|  |  |  |
| --- | --- | --- |
| **Tiêu chí so sánh** | **8086** | **8088** |
| **Kích thước** | 16 bit | 8 bit |
| **Bus** | Dùng chung đường bus 16 bit để đánh địa chỉ và trao đổi dữ liệu (AD15:0) | Dùng chung đường bus 8 bit để đánh địa chỉ và trao đổi dữ liệu (AD7:0)  Có riêng đường bus 8 bit để đánh địa chỉ (AD15:8) |
| **Bộ nhớ** | Bộ nhớ được chia làm 2 bank :   * Chẵn (Bank thấp) * Lẻ (Bank cao) | Bộ nhớ không bị chia |
| **Hàng đợi Chỉ dẫn** | Dài 6 bit | Dài 4 bit |
| **Tiêu thục năng lượng** | Dòng tiêu thụ tối đa 360 mA | Dòng tiêu thụ tối đa 340 mA |

* **Các câu lệnh điều kiện, rẽ nhánh trong hợp ngữ**

**Lệnh CMP**

**CMP** destination, source

So sánh destination (được lưu trong bộ nhớ hoặc thanh ghi) với source (là một hằng số hoặc một giá trị được lưu trong bộ nhớ hoặc thanh ghi). Giá trị so sánh được lưu trong các cờ thanh ghi.

**Nhảy không cần điều kiện với lệnh JMP**

**JMP** label

Nhảy tới đoạn chương trình được gán nhãn label

**Nhảy theo điều kiện**

Các lệnh sau dùng cho các phép tính số học với số có dấu

|  |  |  |
| --- | --- | --- |
| **Lệnh** | **Mô tả** | **Cờ dùng để kiểm tra kết quả** |
| JE/JZ | Jump Equal hoặc Jump Zero | ZF |
| JNE/JNZ | Jump Not Equal hoặc Jump Not Zero | ZF |
| JG/JNLE | Jump Greater hoặc Jump Not Less/Equal | OF, SF, ZF |
| JGE/JNL | Jump Greater/Equal hoặc Jump Not Less | OF, SF |
| JL/JNGE | Jump Less hoặc Jump Not Greater/Equal | OF, SF |
| JLE/JNG | Jump Less/Equal hoặc Jump Not Greater | OF, SF, ZF |

Các lệnh sau dùng cho các phép tính logic với số không dấu

|  |  |  |
| --- | --- | --- |
| **Lệnh** | **Mô tả** | **Cờ dùng để kiểm tra kết quả** |
| JE/JZ | Jump Equal hoặc Jump Zero | ZF |
| JNE/JNZ | Jump Not Equal hoặc Jump Not Zero | ZF |
| JA/JNBE | Jump Above hoặc Jump Not Below/Equal | CF, ZF |
| JAE/JNB | Jump Above/Equal hoặc Jump Not Below | CF |
| JB/JNAE | Jump Below hoặc Jump Not Above/Equal | CF |
| JBE/JNA | Jump Below/Equal hoặc Jump Not Above | AF, CF |

Các lệnh nhảy đặc biệt sau được thực thi sau khi kiểm tra giá trị của cờ

|  |  |  |
| --- | --- | --- |
| **Lệnh** | **Mô tả** | **Cờ dùng để kiểm tra kết quả** |
| JXCZ | Jump if CX is Zero  Nhảy nếu cờ CX có giá trị bằng 0 | Không có |
| JC | Jump If Carry  Nhảy nếu mượn | CF |
| JNC | Jump If No Carry  Nhảy nếu không mượn | CF |
| JO | Jump If Overflow  Nhảy nếu tràn | OF |
| JNO | Jump If No Overflow  Nhảy nếu không tràn | OF |
| JP/JPE | Jump Parity or Jump Parity Even  Nhảy nếu có bit Parity hoặc Nhảy nếu bit Parity chẵn | PF |
| JNP/JPO | Jump No Parity or Jump Parity Odd  Nhảy nếu không có bit Parity hoặc Nhảy nếu bit Parity lẻ | PF |
| JS | Jump Sign (negative value)  Nhảy có dấu (có giá trị âm) | SF |
| JNS | Jump No Sign (positive value)  Nhảy không dấu (có giá trị dương) | SF |

1. **Kết quả và nhận xét (50%)**

**Chương trình hợp ngữ: Cộng 2 mảng số nguyên**

Ý tưởng : Tạo 2 mảng với độ dài và giá trị cho trước, sử dụng vòng lặp cộng từng phần tử của 2 mảng và lưu kết quả vào mảng thứ nhất.

ORG 100H

;INDEX

MOV BX,0

;ARRAY1:

A1 DB 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

;ARRAY2:

A2 DB 9, 8, 7, 6, 5, 4, 3, 2, 1, 0

ADDEACH:

MOV AL, A1[BX] ;COPY BX-th ELEMENT OF ARRAY1 TO AL

ADD AL, A2[BX] ;ADD BX-th ELEMENT OF ARRAY2 TO AL

MOV A1[BX], AL ;COPY RESULT TO BX-th POSITION OF ARRAY1

ADD BX,1 ;INCREASE BX BY 1

CMP BX,10 ;COMPARE BX WITH 10

JL ADDEACH ;LOOP IF LOWER THAN 10

JLE END ;ELSE JUMP TO END

END:

RET

1. **Giải thích chương trình (30%)**

**Giải thích chương trình hợp ngữ robot.asm**

**Nguyên lý hoạt động**

Chương trình điều khiển robot di chuyển ngẫu nhiên và dùng robot để bật/tắt đèn. Vì robot là một sản phẩm cơ hữu nên chương trình có cơ chế kiểm tra tính khả thi rồi mới thực thi công việc tiếp theo.

Chương trình điều khiển giao tiếp với robot thông qua 3 port :

* **command(**out) **–** **port 9**: Điều khiển công việc robot phải thực thi (di chuyển, bật/ tắt đèn). Tín hiệu command được mã hóa bằng 3 bit LSB
* xxxxx001 : Tiến
* xxxxx010 : Quay trái
* xxxxx011 : Quay phải
* xxxxx100 : Dùng cảm biến kiểm tra khu vực
* xxxxx101 : Bật đèn
* xxxxx110 : Tắt đèn
* **data(**in**) – port 10**: Dữ liệu từ cảm biến trên robot gửi về chương trình điều khiển.
* Tất cả các bit là mức ‘0’ : Không đụng vật thể
* Tất cả các bit là mức ‘1’ : Đụng tường
* Có một vài bit mang mức ‘1’ : Đụng vật thể
* **Status(**in**) – port 11**: Trạng thái của robot gửi về chương trình điều khiển. Tín hiệu status sử dụng 3 bit LSB. 3 bit LSB lên mức ‘1’ nghĩa là robot đang bận xử lý công việc trước đó.