**----8086----ASM**

BÀI TOÁN: Cộng trừ nhân chia 2 số nguyên dương bất kỳ 16 bit. (ngưỡng: 1000-50000)

Số lớn nhất có thể chứa là: (65.535) (0FFFFh) (1111.1111.1111.1111b)

- Nhập 1 dãy số bất kì, và kết thúc bằng phím Enter. Rồi xuất ra màn hình.

* 1. **Hiểu ý nghĩa 14 thanh ghi. Đặc biệt 8 thanh ghi đa dụng. Thanh ghi 8 bit, thanh ghi 16 bit.**
* **8 thanh ghi đa dụng cho mục đích chung (General Purpose) 16bit**

|  |  |  |  |
| --- | --- | --- | --- |
| **AX** | Accumulator register(AH,AL) | **SI** | Source index register |
| **BX** | Base address register(BH,BL) | **DI** | Destination index register |
| **CX** | Count register(CH,CL) | **BP** | Base pointer |
| **DX** | Data register(DH,DL) | **SP** | Stack pointer |

* **4 thanh ghi segment(phân đoạn): \*Lưu ý không sử dụng để chứa data**

|  |  |
| --- | --- |
| **CS - Code Segment** | Trỏ tới địa chỉ segment chứa Code |
| **DS - Data Segment** | Trỏ tới segment chứa các Data(Var) được định nghĩa |
| **ES - Extra Segment** |  |
| **SS - Stack Segment** | Trỏ tới địa chỉ vùng nhớ Stack |

* **2 thanh ghi có chức năng đặc biệt:\* Lưu ý không sử dụng để chứa data**

|  |  |
| --- | --- |
| **IP(PC)** | **Instruction Pointer (Program Counter)** kết hợp với CS để xác định địa chỉ lệnh chuẩn bị thực thi (trở tới offset chứa lệnh trong segment được CS trỏ) |
| **Flag register** | Thanh ghi cờ trạng thái( tự động thay đổi bởi CPU khi thực hiện phép toán, dùng để “nhảy” chương trình |

* 1. **Biết cách: tạo mới chương trình, chạy chương trình, xem kq thanh ghi.**

New 🡪COM template 🡪Viết code giữa “org” và “ret”

Chú thích viết sau “;”

Org 100h; dia chi offset bat dau

Khai báo sau “ret” cú pháp : <var>\_<datatype>\_<data>

Emulator:

View var:xem biến

View memory: xem bộ nhớ

View stack: xem stack

* 1. **Bạn biết 3 cách (thể hiện) để viết 1 con số trong code.(DEC, HEX, BIN)**
  2. **Tạo biến, lưu 1 kq vào biến, xem biến.**

**<var\_name> <dw/db/dd/> <data>**

**Mov <var\_name> <result>**

**Xem biến: view 🡪 variables**

* 1. Biến lưu ở đâu? **Memory**. Biến được quản lý thông qua địa chỉ của ô nhớ.   
     Xem địa chỉ ô nhớ chứa biến trong emulator ntn?
  2. Vấn đề “matching” (đồng bộ kích thước): các toán hạng phải đồng nhất về kích thước. vd: mov dx, so / so DB 0fh. Lỗi, cách giải quyết?
  3. **Chạy được từng bước của chương trình. Single Step**.
  4. Hiểu các Operand (toán hạng) trong phép toán. Có 4 loại.
  5. Biết sử dụng lệnh MOV, và các operand có thể sử dụng được cho lệnh MOV?
  6. Có những dạng kí tự nào?
  7. **Nhập 1 con số từ bàn phím, được không? KHÔNG ĐƯỢC!**
  8. **Câu lệnh nhập 1 kí tự từ bàn phím**.

Mov AH,1

INT 21h

* 1. **1 kí tự được lưu trong bộ nhớ của VĐK/VXL ntn?**
     + Bản chất VĐK/VXL không hiểu ngôn ngữ con người. Nên thực tế, kí tự sẽ được lưu dưới dạng 1 con số (tương ứng trong bảng mã ASCII)
  2. **Kí tự ‘a’ được lưu trong bộ nhớ tương ứng con số bao nhiêu?**
     + 97(decimal) = 61(hexa)
  3. **Kí tự ‘5’ được lưu trong bộ nhớ tương ứng con số bao nhiêu?**
     + 53(decima) = 35(hexa)
  4. **Cách chuyển 1 kí tự số thành 1 con số?**
     + Lấy kí tự đó – trừ đi cho 30h.
  5. Nhập 1 kí tự số, chuyển sang số, lưu vào 1 biến
  6. CT có cho nhập 1 số từ bàn phím không? Nếu muốn nhập, thì phải làm sao?
     + Không cho nhập số. Nhập 1 ký tự số, sau đó chuyển nó thành số.
  7. Vậy 1 lần nhập, thì mình chỉ nhập được 1 con số có mấy chữ số?
     + 1 chữ số (vì mỗi lần mình chỉ nhập được 1 kí tự)
  8. **Mình sử dụng phép toán gì để nhập kí tự?**
     + Dùng INT21 với AH=1, kết quả lưu vào AL(1byte). Lý do, 1 kí tự chiếm 1 byte.
  9. Muốn dừng chương trình (đợi nhấn phím bất kì) thì làm sao?
     + Dùng INT16/AH=0
  10. **Xuất 1 chuỗi ntn?**
      + Dùng **int21/ah=9**
  11. **Muốn nhập số có 2 chữ số thì làm sao?**
      + Nhập 1 số, lưu kq vào 1 biến.
      + Tiếp tục nhập số tiếp theo.
      + Nhân kq trước đó với 10, rồi cộng thêm số vừa nhập.
      + (Gợi ý: Sử dụng bộ nhớ stack để lưu số vừa nhập)
  12. **Nhập số có nhiều chữ số bất kì thì sao?**
      + Dùng lệnh JMP để nhập lại liên tục.
  13. **Nhập số có nhiều chữ số, dừng lại khi bấm phím Enter?**
      + Kí tự Enter có mã ASCII là 0Dh (13)
      + Dùng lệnh nhảy có điều kiện. Lệnh nhảy dựa vào giá trị của Flag (C, Z, S).
      + Ý nghĩa của 3 cờ sẽ dùng.
      + Để tác động vào cờ thì dùng lệnh CMP (compare) – ý nghĩa là thực hiện phép trừ mà không lưu kết quả, để tác động vào Flag.
  14. **Xuất số có nhiều chữ số ntn?**
      + Chia số cho 10, lấy phần dư (đơn vị) lưu vào Stack.  
        (Mỗi lần lưu vào Stack thì cần INC tăng 1 biến CX để đếm số có bao nhiêu chữ số)
      + Rồi tiếp tục lấy phần thương chia tiếp cho 10 và lặp lại bước trên cho đến khi phần thương = 0.
      + Sau đó, bạn đọc số từ Stack ra, rồi chuyển thành kí tự và xuất ra màn hình (đây đúng là trình tự bạn cần). Mỗi lần đọc thì giảm CX xuống 1 đến khi CX=0 (tức xuất đủ số chữ số).