|  |
| --- |
| PRACTICE EXERCISES OF THE MICROPROCESSORS & MICROCONTROLLERS  Instructor: The Tung Than  Student's name: Nguyễn Tiến Ngọc  Student code: 21522381 |

PRACTICE REPORT NO 5

Lab5

I. **Practice content**

**1. Example code:**

A screenshot of a computer code

Description automatically generated with medium confidence

- Khởi tạo giá trị ban đầu cho 2 thanh ghi al, và bl bằng lệnh mov (dòng 5, 6).

- Thực hiện cộng thanh ghi bl với thanh ghi al, kết quả lưu vào thanh ghi al.

- Thực hiện trừ thanh ghi bl đi 1, kết quả lưu vào thanh ghi bl.

- In ra kết quả ở dạng nhị phân:

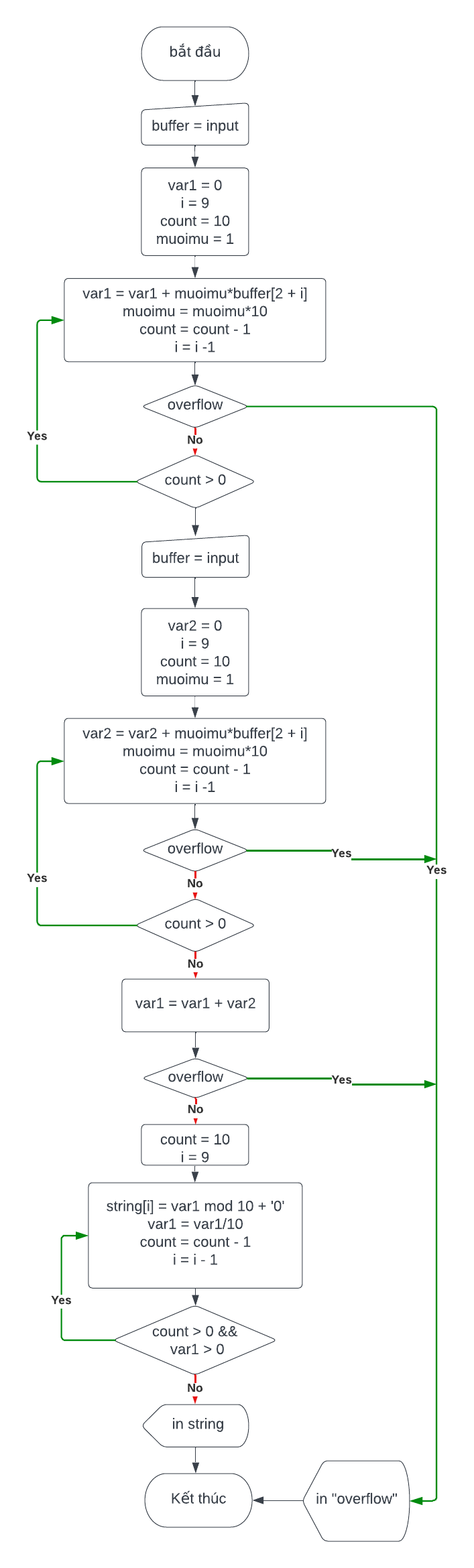
+ Dùng ngắt int 21h với ah = 2 để in ra từng kí tự.

+ khởi tạo cho thanh ghi dl (thanh ghi chứa kí tự cần in ra) là kí tự ‘0’.

+ Kiểm tra xem bit trọng số 7 có bằng 1 hay không, nếu đúng thì gán dl = ‘1’ (kiểm tra bằng lệnh test và lệnh jz-jump if ZF=1, lệnh test thực hiện and 2 toán tử, set cờ zero nếu kết quả bằng 0, hai toán tử không thay đổi giá trị).

+ in ra kí tự bằng lệnh int 21h với ah = 2.

+ shift thanh ghi bl sang trái (chèn 0 vào bit có trọng số 0), sau đó lặp lại quá trình trên 8 lần (thanh ghi dl chứa 8 bit giá trị).

**2. Write a program that adds 2 32-bit numbers.**

**a. Flowchart**

**b. Code**

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 program

Description automatically generated with low 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

**c. Explanation**

\* Nhập vào string, chuyển sang nhị phân:

- Khai báo 1 buffer gồm 11 kí tự, 10 chữ số và 1 kí tự kết thúc chuỗi. (2^32 - 1 có thể biểu diễn bằng 10 chữ số).

- Thực hiện chuyển đổi string sang mã bcd, từ bcd chuyển sang binary bằng cách lấy từng chữ số nhân với 10^(trọng số của chữ số). Vd 432 = 4\*10^2 + 3\*10^1 + 2\*10^0.

- 8086 chỉ hỗ trợ phép nhân 16\*16 bit, mà phép nhân lớn nhất ta cần thực hiện là 4\*10^9 (2^32 - 1 = 4294967296) là phép nhân 16 bit \* 32 bit, kết quả là 1 số 48 bit.

+ giả sử số A 32 bit gồm phần word cao H, word thấp L1, số L2 16 bit. A\*L2 = (H\*2^16 + L1)\*L2 = H\*L2\*2^16 + H\*L1. Như vậy ta đã đơn giản hóa phép nhân 32b\*16 thành 16b\*16 kết hợp với phép dịch (nhân 2^16 là dịch 16 bit, nhưng thực tế trong code không dùng phép dịch mà gán kết quả của H\*L2 vào phần bit 48-16 của kết quả).

\* tính toán var1 + var2:

+ lấy word thấp của var1 + word thấp của var2

+ lấy word cao của var1 + word cao của var2 + bit nhớ từ phép tính trước.

\* In ra kết quả:

- chia lấy dư kết quả cho 10, gán string[i] = phần dư + ‘0’(I = 9 -> 0). Kết quả = kết quả / 10, i = i – 1. Lặp lại cho đến khi kết quả = 0 hoặc lặp đủ 10 lần.

+ Thực hiện phép chia 32 bit cho số 10 (16 bit): 8086 hỗ trợ phép chia 32 bit cho 16 bit, nhưng kết quả có thể bị tràn do kết quả trong phép chia chỉ được lưu ở thanh ghi 16 bit (32 bit chia 16 bit kết quả phải là 32 bit).

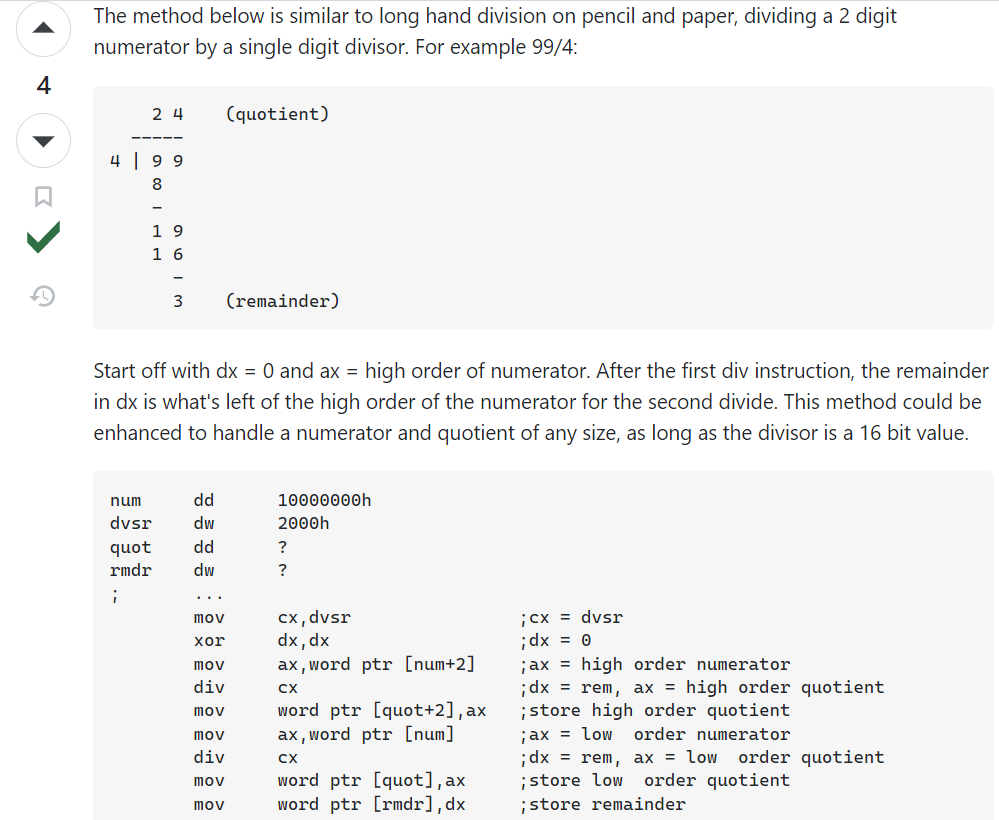
+ Giải sử số 32 bit có dạng H + L1, số 16 bit có dạng L2.

+ Thực hiện chia phần cao H (16 bit) cho L2: gán DX = 0 (phần cao của số 32 bit), AX = H, BX = L2. Vậy đây có thể coi như là phép chia 16 bit cho 16 bit. Kết quả là 16 bit sẽ không bị tràn. (\*1)

+ Phần nguyên của phép chia 1 chính là phần cao của kết quả phép chia mong muốn (32b/16b).

+ Thực hiện chia (DX AX) cho BX: DX = phần dư của phép chia trước đó, AX = L1, BX = L2. Đây vẫn là 1 phép chia 32 bit cho 16 bit. Nhưng sẽ không bị tràn vì phần DX < L2 (vì DX là phần dư của phép chia trước, luôn bé hơn L2). Phần nguyên của phép chia này là word thấp trong phần nguyên của kết quả. Phần dư DX là phần dư của toàn bộ phép chia 32b/16b.

+ Tham khảo thêm về phép chia 32b/16b:



- Sau khi lặp, i + 1 chính là địa chỉ của số đầu tiên cần in ra, thực hiện in ra chuỗi kể từ i (ví dụ kết quả là 123 thì chỉ in ra 123 thay vì 0000000123).

**d. Demonstration**

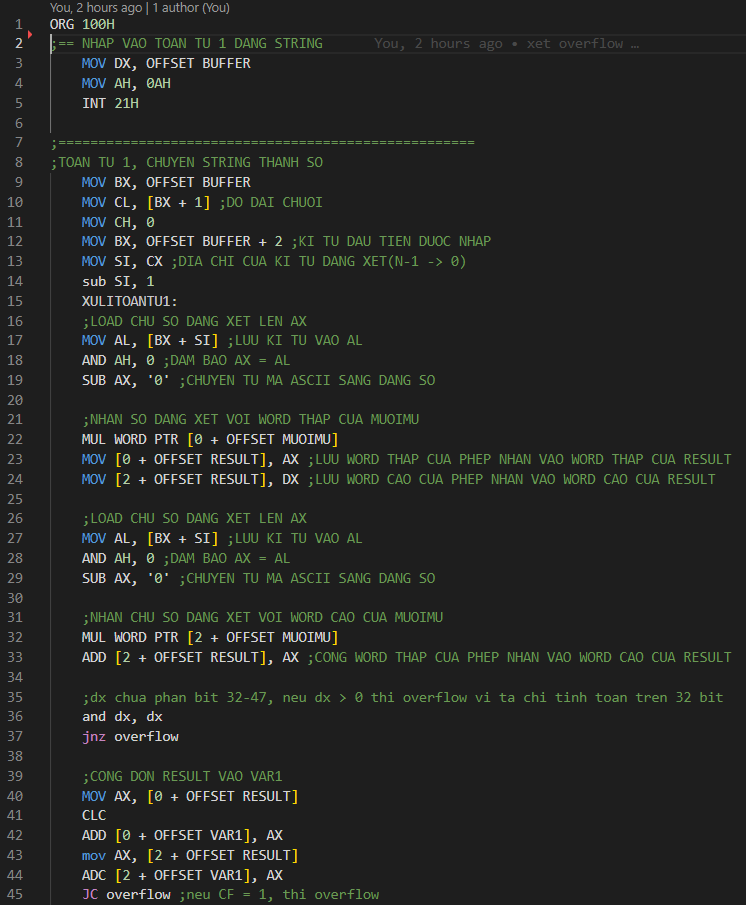
- Link drive: https://drive.google.com/file/d/1qHrkXzcZBjpND1mI-Ugg1a7SIEbp3oeO/view?usp=sharing

A picture containing text, diagram, plan, font

Description automatically generated**III. Exercise**

**1. Flowchart**

**2. Code**



A screenshot of a computer program

Description automatically generated with medium confidence

A screenshot of a computer program

Description automatically generated with low 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 picture containing text, screenshot, font, number

Description automatically generated

**3. Explanation**

- Mọi thứ giống như bài phép cộng, chỉ khác là phép trừ không xét overflow ở kết quả (2 số dương trừ nhau không thể overflow).

- Xét xem kết quả có âm hay không bằng cờ CF, nếu âm thì in ra dấu trừ và kết quả = 0 – kết quả (lấy giá trị tuyệt đối).

**4. Demonstration**

- Link Drive: https://drive.google.com/file/d/1UpLTkgakvgzOD4B6Qxqb6xHLfh6ePYx1/view?usp=sharing