BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH TUẦN 4

Họ và tên: Nguyễn Quý Đức MSSV: 20235682

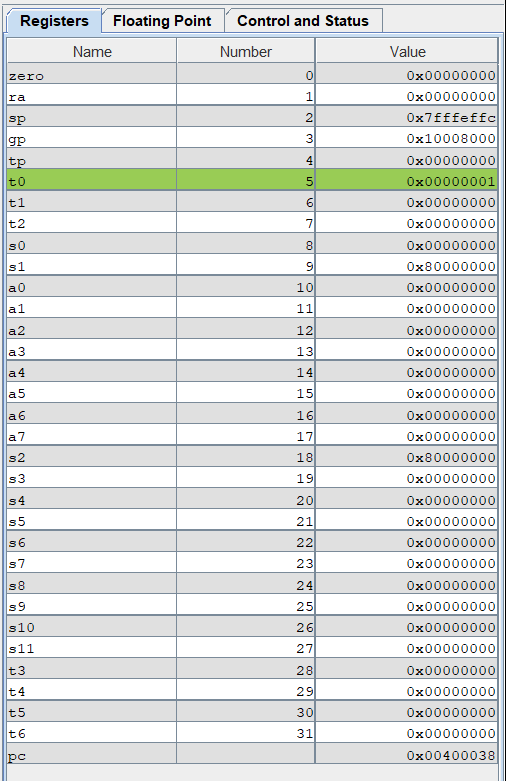
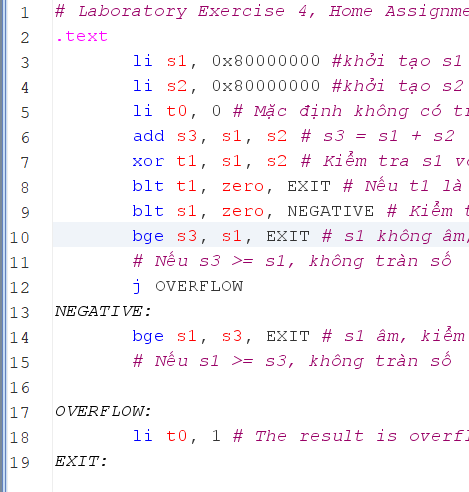
Mã HP: IT3280 Mã lớp: 156788

1. Assignment 1
   1. HWA 1

- Giải thích:

* Khởi tạo tạo t0 = 0, s1, s2
* Sau khi thực hiện phép cộng s3 = s1 + s2, chương trình kiểm tra dấu của s1 và s2 bằng phép XOR (t1 = s1 ⊕ s2).
  + Nếu t1 âm (do bit dấu của s1 và s2 khác nhau, nghĩa là s1 và s2 khác dấu), chắc chắn không có tràn số.
  + Nếu s1 âm, chương trình kiểm tra xem s3 có lớn hơn s1 không, nếu có thì tràn số. Ngược lại, nếu s1 không âm, kiểm tra s3 < s1, nếu có thì tràn số. Nếu có tràn số, đặt t0 = 1, ngược lại thoát chương trình.

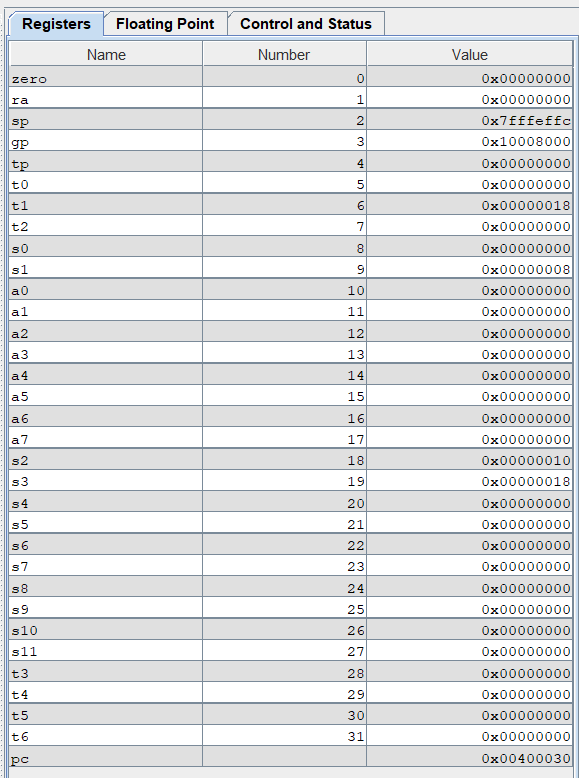
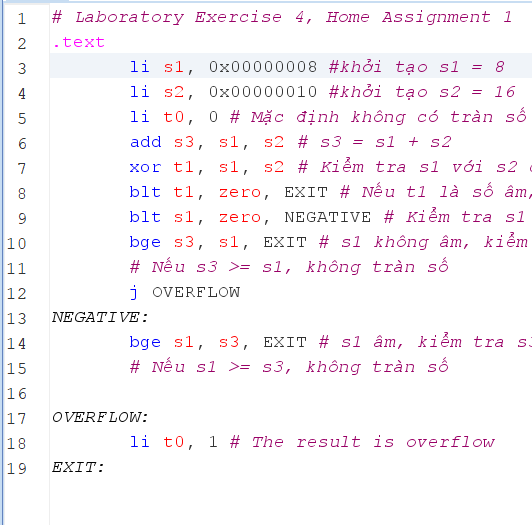
TH1: s1, s2 âm, s3 tràn số



* Trong TH1, s1 = s2 = 0x80000000 = -2147483648, đáng lẽ ra s3 = s1 + s2 = -4294967296 nhưng lại trả về s3 = 0x00000000 = 0 > s1 => tràn số => t0 = 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trạng thái | S3 | T0 | T1 | Pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00400000 |
| Sau lệnh addi | 0x00000000 | - | - | 0x00400018 |
| Sau lệnh xor | - | - | 0x00000000 | 0x0040001c |
| Sau lệnh blt kiểm tra s1 s2 cùng dấu | - | - | - | 0x00400020 |
| Sau lệnh blt kiểm tra s1 s2 âm/dương | - | - | - | 0x0040002c |
| Sau lệnh bge (ở lable NEGATIVE) | - | - | - | 0x00400030 |
| Sau lệnh li check tràn số | - | 0x00000001 | - | 0x00400038 |

TH2: s1, s2 dương, s3 không tràn số

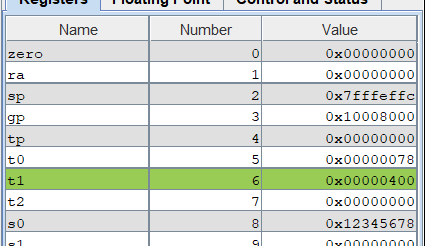
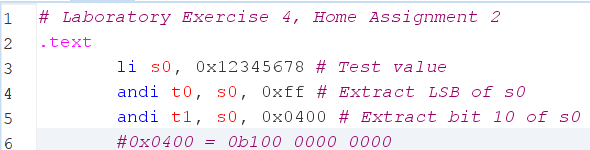


* Trong TH2, s1 = 8, s2 = 0x10 = 16, s3 = s1 + s2 = 0x18 = 24 > s1 => không xảy ra tràn số => t0 = 0

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trạng thái | S3 | T0 | T1 | Pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00400000 |
| Sau lệnh addi | 0x00000018 | - | - | 0x00400010 |
| Sau lệnh xor | - | - | 0x00000018 | 0x00400014 |
| Sau lệnh blt kiểm tra s1 s2 cùng dấu | - | - | - | 0x00400018 |
| Sau lệnh blt kiểm tra s1 s2 âm/dương | - | - | - | 0x0040001c |
| Sau lệnh bge | - | 0x00000000 | - | 0x0040002c  (sau đó nhảy đến lable EXIT) |

­

* 1. HWA 2



- Giải thích:

s0 = 0x12345678 = 0b 0001 0010 0011 0100 0101 0110 0111 1000

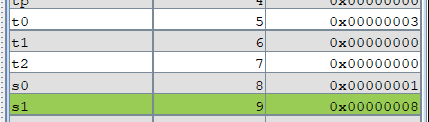
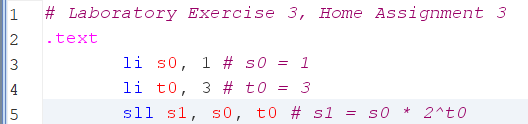
0xff = 0b 1111 1111

0x0400 = 0b 0000 0100 0000 0000

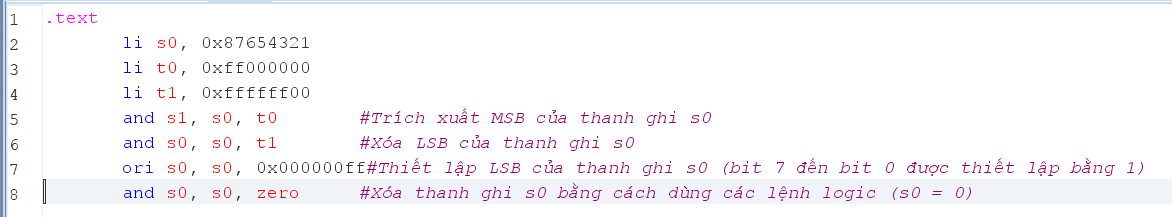
t0 = (s0 and 0xff) = 0b 01111000 = 0x78

t1 = (s0 and 0x0400) = 0b 0000 0100 0000 0000 = 0x400

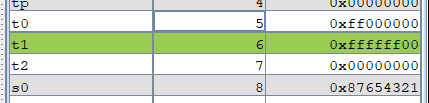
* 1. HWA 3



1. Assignment 2



Khởi tạo s0 = 0x87654321 và t0, t1 cần thiết



Trích xuất MSB của s0 và lưu vào s1

Xóa LSB của s0  


Thiết lập LSB của s0 

Xóa thanh ghi s0



1. Assignment 3
   1. neg s0, s1

=> sub s0, zero, s1

* 1. mv s0, s1

=> addi s0, s1, 0

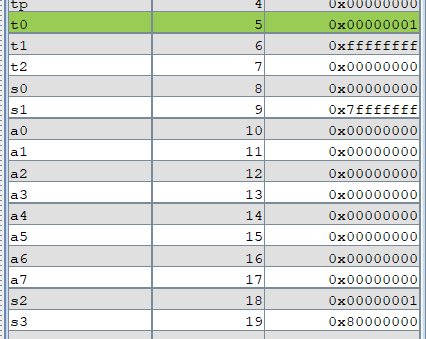
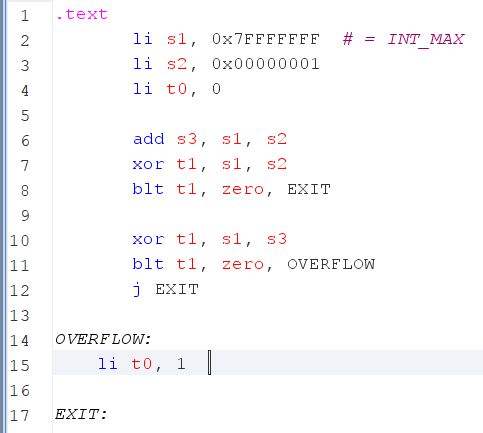
* 1. not s0, s1

=> xori s0, s1, -1

* 1. ble s1, s2, label

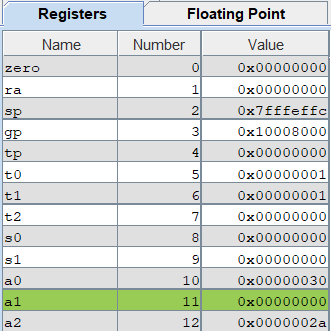
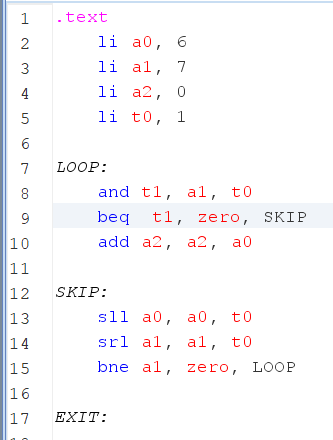
=> bge s2, s1, lbl

1. Assignment 4



Do s1 + s0 = 0x7FFFFFFF + 0x1 (= 2147483647 + 1) = 0x800000000 (= -2147483648 trái dấu s1) => tràn số => t0 = 1

1. Assignment 5



- Giải thích:

* Kiểm tra bit thấp nhất của a1
  + Nếu là 1, cộng a0 vào a2
  + Nếu là 0, bỏ qua
* Dịch trái a0
* Dịch phải a1 (loại bỏ bit cuối cùng)
* Lặp lại đến khi a1 = 0 (xét hết bit)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Loop | A1 | T1 | A2 | A0 |
| 0 | 0000 0111 (7) | 0 | 0000 0000 (0) | 0000 0110 (6) |
| 1 | 0000 0111 (7) | 1 | 0000 0110 (6) | 0000 1100 (12) |
| 2 | 0000 0011 (3) | 1 | 0001 0010 (18) | 0001 1000 (24) |
| 3 | 0000 0001 (1) | 1 | 0010 1010 (42) | 0011 0000 (48) |
| 4 | 0000 0000 (0) | 0 | - | - |

Kiểm tra 6 \* 7 = 42 = 0x2a => thỏa mãn