**BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH (IT3280) TUẦN 3**

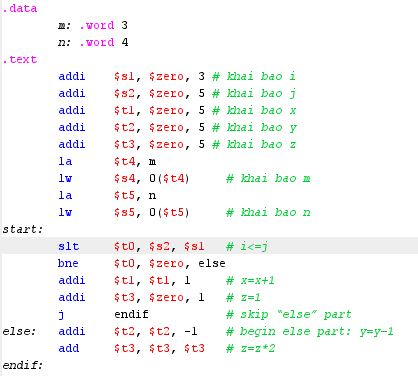
***Họ và tên: Phạm Đức Long***

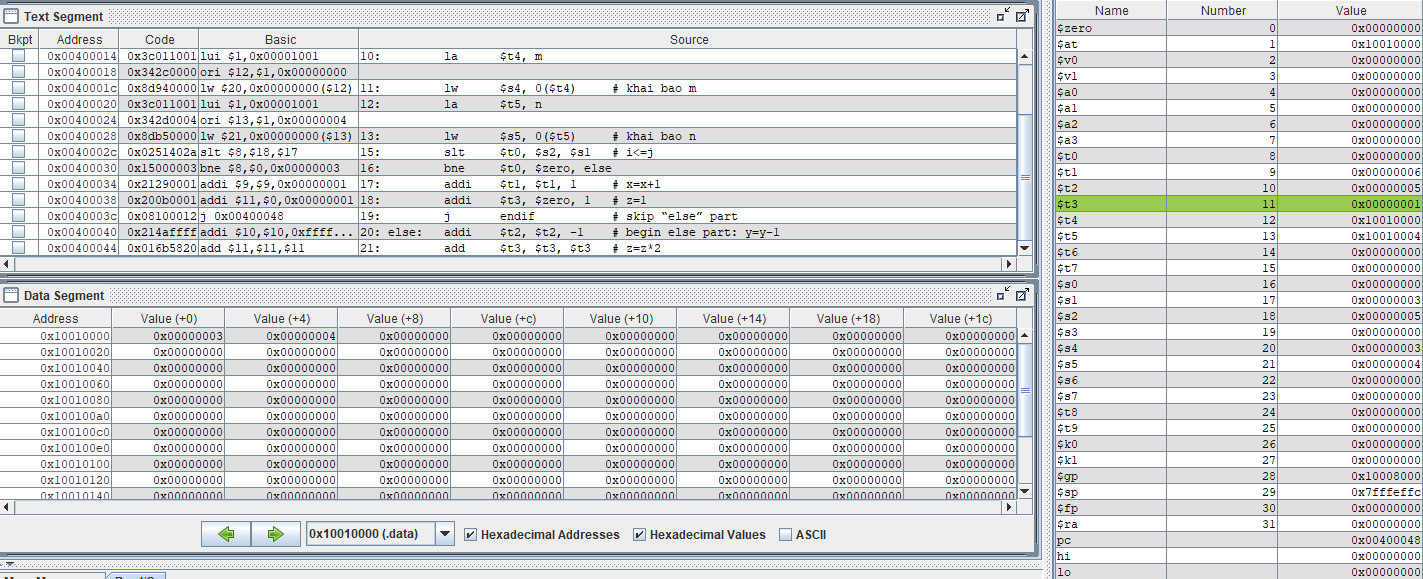
***MSSV: 20225737***

Assignment 1:

Ta xét hai trường hợp i = 3, j = 5 và ngược lại.

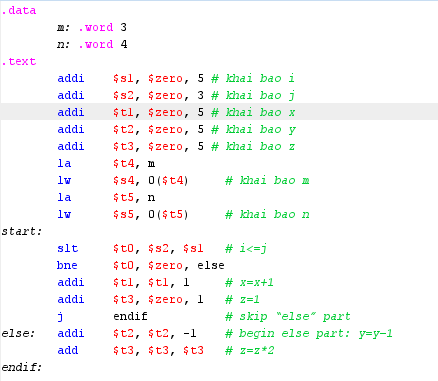
* TH1: i = 3 và j = 5 ( i <= j):

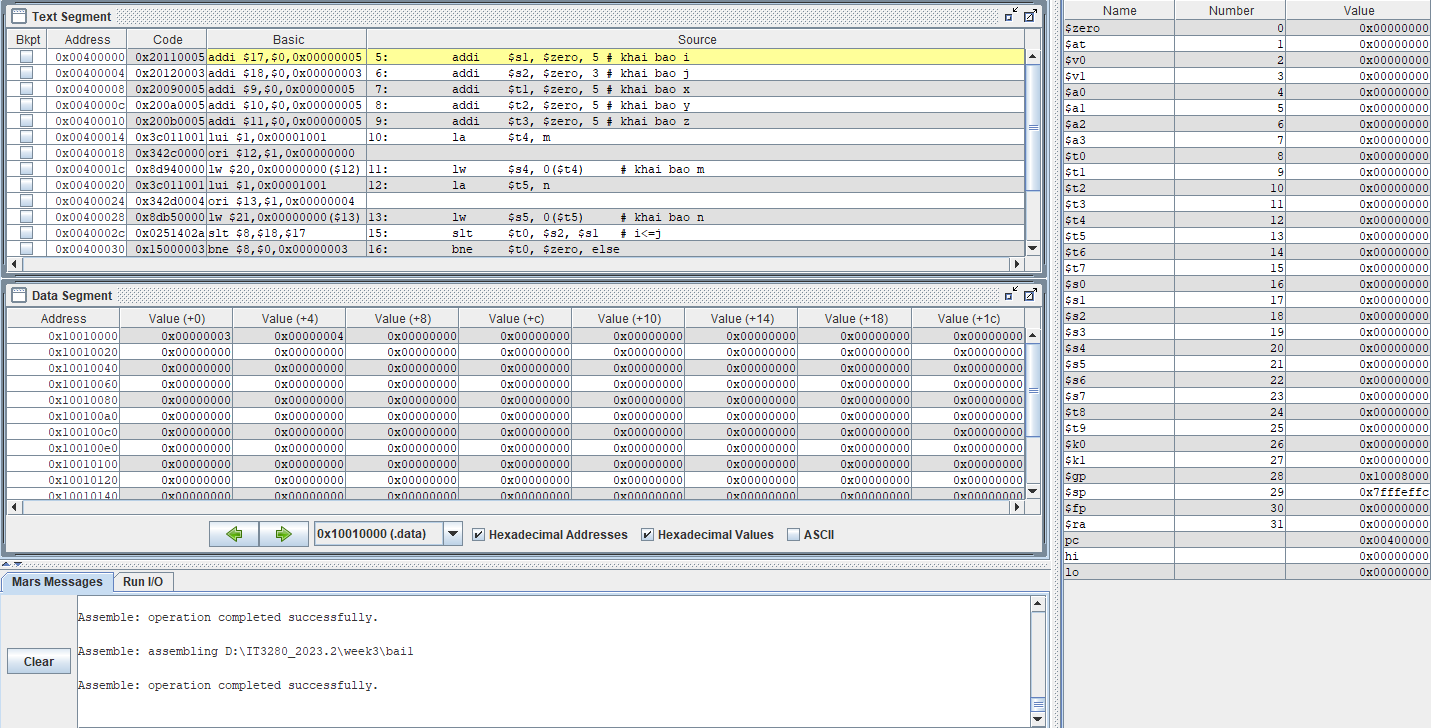




|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | $s1 | $s2 | $t0 | $t1 | $t2 | $t3 | pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x40000000 |
| Khai báo i | 0x00000003 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x40000004 |
| Khai báo j | 0x00000003 | 0x00000005 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x40000008 |
| Khai báo x, y, z | 0x00000003 | 0x00000005 | 0x00000000 | 0x00000005 | 0x00000005 | 0x00000005 | 0x40000014 |
| Chạy lệnh slt | 0x00000003 | 0x00000005 | 0x00000000 | 0x00000005 | 0x00000005 | 0x00000005 | 0x40000030 |
| Chạy lệnh bne | 0x00000003 | 0x00000005 | 0x00000000 | 0x00000005 | 0x00000005 | 0x00000005 | 0x40000034 |
| Chạy lệnh addi thứ 1 của start | 0x00000003 | 0x00000005 | 0x00000000 | 0x00000006 | 0x00000005 | 0x00000005 | 0x40000038 |
| Chạy lệnh addi thứ 2 của start | 0x00000003 | 0x00000005 | 0x00000000 | 0x00000006 | 0x00000005 | 0x00000001 | 0x4000003c |

* Việc kiểm thử là đúng khi ta thấy j >= i (với giá trị thanh ghi $s1 là 3 (0x00000003) và giá trị thanh ghi $s2 là 5 (0x00000005)) thì giá trị thanh ghi $t0 sẽ trả về 0 (địa chỉ 0x00000000) và giá trị thanh ghi này đúng với thanh ghi $zero nên trình biên dịch không thực hiện lệnh rẽ nhánh mà tiếp tục các lệnh addi, từ đó tính toán và gán giá trị cho thanh ghi $t1 là 6 (0x00000006) và gán giá trị thanh ghi $t3 là 1 (0x00000001).
* TH2: i = 5 và j = 3 ( i > j):

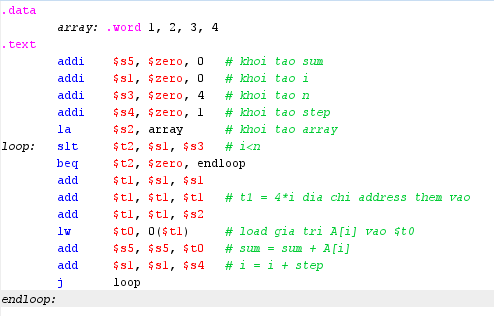




|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | $s1 | $s2 | $t0 | $t1 | $t2 | $t3 | pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x40000000 |
| Khai báo i, j | 0x00000005 | 0x00000003 | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 | 0x40000008 |
| Khai báo x, y, z | 0x00000005 | 0x00000003 | 0x00000000 | 0x00000005 | 0x00000005 | 0x00000005 | 0x40000014 |
| Chạy lệnh slt | 0x00000005 | 0x00000003 | 0x00000001 | 0x00000005 | 0x00000005 | 0x00000005 | 0x40000030 |
| Chạy lệnh bne | 0x00000005 | 0x00000003 | 0x00000001 | 0x00000005 | 0x00000005 | 0x00000005 | 0x4000001c |
| Chạy lệnh addi thứ 1 của else | 0x00000005 | 0x00000003 | 0x00000001 | 0x00000000 | 0x00000004 | 0x00000000 | 0x40000044 |
| Chạy lệnh add của else | 0x00000005 | 0x00000003 | 0x00000001 | 0x00000000 | 0x00000004 | 0x0000000a | 0x40000048 |

* Việc kiểm thử là đúng khi ta thấy j < i (với giá trị thanh ghi $s1 là 5 (0x00000005) và giá trị thanh ghi $s2 là 3 (0x00000003)) thì giá trị thanh ghi $t0 sẽ trả về 1 (địa chỉ 0x00000001) và giá trị thanh ghi này khác với thanh ghi $zero nên trình biên dịch đã thực hiện lệnh rẽ nhánh, từ đó tính toán và gán giá trị cho thanh ghi $t2 là 4 (địa chỉ là 0x00000004) và gắn giá trị thanh ghi $t3 bằng 2 lần giá trị chính nó là 10 (địa chỉ là 0x0000000a) .

Assignment 2:



* Đầu tiên ta khởi tạo mảng có 4 phần tử có các giá trị là 1, 2, 3, 4.
* Tiếp đến ta khởi tạo các giá trị i, n, step (bước nhảy), step = 4 => sẽ có 4 lần lặp => sau 4 vòng lặp ta sẽ thu được tổng các phần tử trong chuỗi chính là giá trị của thanh ghi $s5.
* Sự thay đổi của các thanh ghi:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | $t1 | $t2 | $s1 | $s5 |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| Vòng loop 1 | 0x10010000 | 0x00000001 | 0x00000001 | 0x00000001 |
| Vòng loop 2 | 0x10010004 | 0x00000001 | 0x00000002 | 0x00000003 |
| Vòng loop 3 | 0x10010008 | 0x00000001 | 0x00000003 | 0x00000006 |
| Vòng loop 4 | 0x1001000c | 0x00000001 | 0x00000004 | 0x0000000a |
| Vòng ‘loop’ 5 | 0x1001000c | 0x00000000 | 0x00000004 | 0x0000000a |

* Nhận xét:
* Vòng lặp dừng lại khi $t2 = 0x00000000, tương ứng với điều kiện câu lệnh: beq $t2, $zero, endloop
* Kết quả giá trị sum = $s5 = 0x0000000a = 10 = 1 + 2 + 3 + 4, điều này đúng với tính toán khi ta tính tổng các phần tử của mảng.

Assignment 3:

* TH1: text = 1:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | $s1 | $s2 | $s3 | pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| Sau khi khởi tạo | 0x00000001 | 0x00000000 | 0x00000000 | 0x00400018 |
| Sau lệnh beq case\_1 và case\_1 | 0x00000001 | 0xffffffff | 0x00000000 | 0x00400034 |

* Do $s1 = $t1 = 1 nên lệnh hàm sẽ thực hiện hàm case\_1 và thanh ghi pc nhảy đến giá trị của case\_1 là 0x00400030 thực hiện lệnh trừ thanh ghi $s2 cho 1 ta được $s2 = -1 ( thỏa mãn).
* TH2: text = 0:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

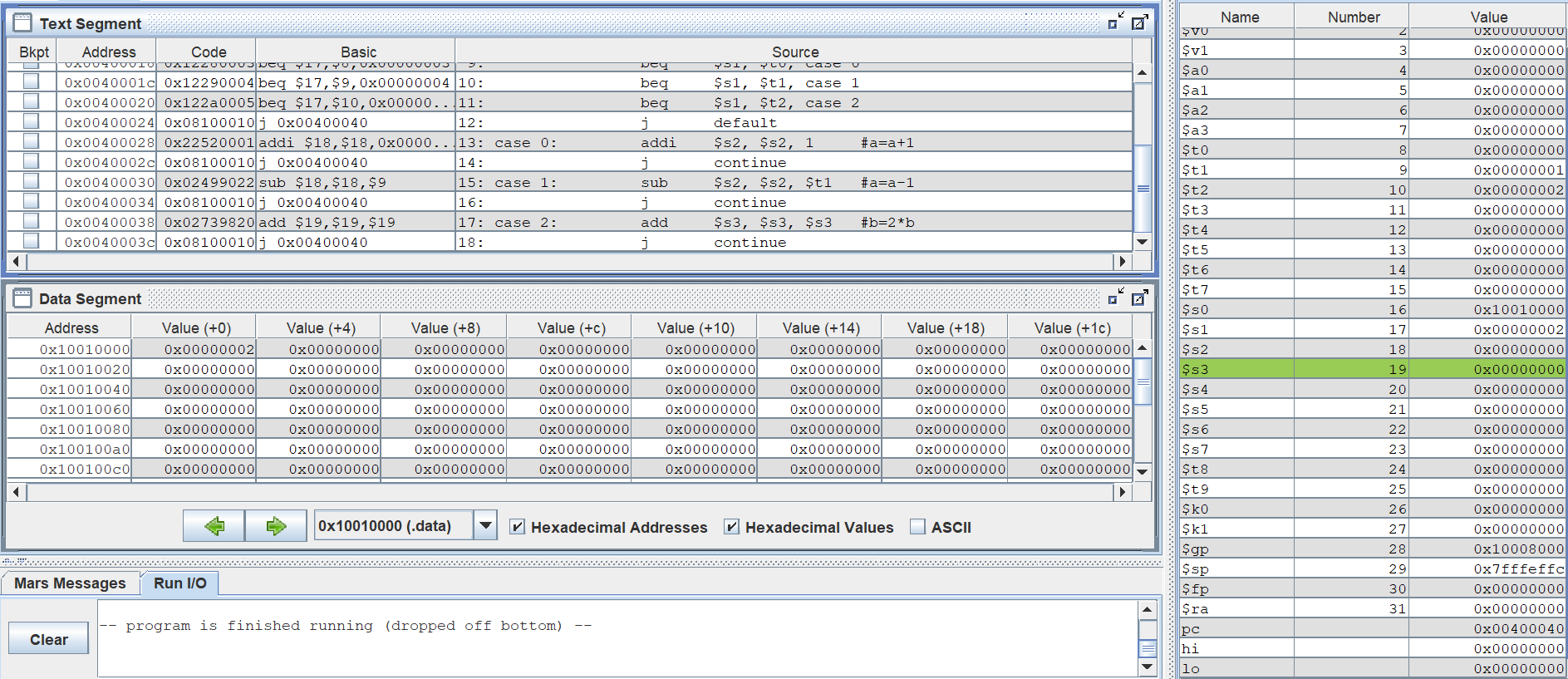
Description automatically generated

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | $s1 | $s2 | $s3 | pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| Sau khi khởi tạo | 0x00000000 | 0x00000000 | 0x00000000 | 0x00400018 |
| Sau lệnh beq case\_0 và case\_0 | 0x00000000 | 0x00000001 | 0x00000000 | 0x0040002c |

* Do $s1 = $t0 = 0 nên lệnh hàm sẽ thực hiện hàm case\_0 và thanh ghi pc nhảy đến giá trị của thẻ case\_0 là 0x00400028 thực hiện lệnh cộng thanh ghi $s2 cho 1 ta được $s2 = 1 ( thỏa mãn).
* TH3: text = 2:

A screenshot of a computer program

Description automatically generated



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | $s1 | $s2 | $s3 | pc |
| Ban đầu | 0x00000000 | 0x00000000 | 0x00000000 | 0x00000000 |
| Sau khi khởi tạo | 0x00000002 | 0x00000000 | 0x00000000 | 0x00400018 |
| Sau lệnh beq case\_2 và case\_2 | 0x00000002 | 0x00000000 | 0x00000000 | 0x0040003c |

🡪 Như vậy do $s1 = $t2 = 2 nên lệnh hàm sẽ thực hiện hàm case\_2 và thanh ghi pc nhảy đến giá trị của thẻ case\_0 là 0x00400038 thực hiện lệnh cộng thanh ghi $s3 với thanh ghi $s3 ta được $s3 = 0 ( thỏa mãn).

Assignment 4:

Assignment 5:

Assignment 6: