**Assignment 1**

Sự thay đổi của thanh ghi s0:

* Sau bước 1, giá trị của thanh ghi là 0x00003007
* Sau bước 2 giá trị của thanh ghi giữ nguyên

Sự thay đổi của thanh ghi pc: Sau mỗi bước giá trị trên thanh cộng thêm 4

Code: 0x20103007 = 0010 0000 0001 0001 0011 0000 0000 0111 ( I – type)

Phân tích code:

* Opcode: 001000
* Rs: 00000
* Rt: 10001
* Immediately: 0011 0000 0000 0111

Code: 0x00108020 = 0000 0000 0001 0000 1000 0000 0010 0000 (R – type)

Phân tích code:

* Opcode: 000000
* Rs: 00000
* Rt: 10000
* Rd: 10000
* Shamt: 00000
* Funct: 100000

Thay đổi sau khi đổi lệnh:

* Thanh ghi at được gán đến giá trị 0x21100000
* Thanh ghi at được gán giá trị 0x2110003d

Xảy ra hiện tượng 2 lần vì addi là cộng giá trị có dấu

**Assignment 2**

Thay đổi giá trị của thanh ghi s0

* Sau bước 1: $s0 có giá trị 0x21100000
* Sau bước 2: $s0 có giá trị 0x2110003d

Các byte đầu tiên của vùng lệnh trùng với cột tương ứng với giá trị của thanh pc là value(+0) và value(+4)

**Assignment 3**

Trong bước thứ nhất, lệnh gán li được chia làm 2 phần lui để gán giá trị cho các bit cao và ori để gán giá trị cho các bit thấp

Do 2 là số immediate quá nhỏ nên phép cộng trở thành phép cộng không dấu và đưa giá trị tổng của $s0 và $0x2 vào $s1

**Assignment 4**

Step 1: Gán giá trị 0 + 5 cho thanh ghi t1

Step 2: Gán giá trị 0 - 1 cho thanh ghi t2

Step 3: Gán giá trị của t1 bằng cách cộng thêm chính nó cho s0

Step 4: Gán giá trị của s0 bằng tổng của nó với s2

**Assignment 5**

Ở 2 bước 1 và 2, thanh ghi t1 và t2 lần lượt được gán giá trị 4 và 5 ( có dấu)

Ở bước 3, thanh ghi s0 có giá trị là 4\*5 = 20(hệ 10) = 14(hệ 16), và thanh gi lo có giá trị tưởng tự, thanh ghi hi giữ nguyên giá trị

Ở bước 4: Lệnh addi được thực hiện kèm theo để chuyển giá trị 0x3 sang thanh ghi $1

Sau đó tích của $1 và $s0 được lưu vào vị trí của $s0

Giá trị của $s0 và lo là 60( 3c trong hệ hex)

Ở bước 5, lệnh mflo di chuyển giá trị của thanh ghi lo vào thanh ghi s1