Cho 1 đoạn code:

a. chữ else đó đc thay bằng hằng số và bằng bao nhiêu

= bao nhiêu lệnh ở giữa lệnh bne và lệnh cần nhảy 1 – **2 – 3 - 4 –** 5 => 3

La 3 boi vi no la lenh 2 chu ki. Bat dau tinh tu lenh nop

Chèn nop vào sau tất cả các lệnh có 2 chu kì : bne, j, lw, sw

Có thể đổi vị trí của lệnh ngay bên BNE xuống bên dưới BNE để tiết kiệm thời gian, thay cho việc thực việc NOP.

Hàm la thực ra là có thể lấy địa chỉ của lệnh, của biến => thực ra là gán hằng số thôi :3

Lệnh giả có gồm :

1. Những lệnh không có trong file của thầy

2. Tham số của lệnh có khuôn I gồm 32 bits : ví dụ addi có tham số quá lớn, 32 bits.

Jal có 2 bước chạy – là lệnh 2 chu kì

1. lưu địa chỉ lệnh tiếp theo (sau jal) vào $ra

2. trỏ pc tới hàm con

Jr $ra giúp nó trở về

Jr : lấy địa chỉ trong 1 thanh ghi nào đó để trở về

Mặc định lưu địa chỉ ào $ra nên là trong hàm con, ko nên động chạm đến $ra ( k nên gọi thêm hàm con nào nữa ?? )

Stack di tu duoi len

$s7 : store the starting address of stack

$t1 : status -1

$t2 : status -2

$t6 : store the integer read

$t7 : store the integer read

$a0 : store the last bit of the number

$s0 - $s5 : 10 -15

$a0 : store the last 4 bits of the number

$t3 = 0 if $a0 ( loaded from stack) < 10

1111111

1000000

0100000

Về viết sẵn hàm display số sang LED 7 đoạn SHOW\_1, SHOW\_2….

Double layer

1 13-20

2 10-23

3 8-10, 12-13, 22-25

4 7-8, 11-12, 22, 24-26

5 6-7, 9-11, 23, 26, 27

6 5-11, 23, 27-28

7 4-11, 24, 28-29

8 3-11, 24, 29

9 3, 5-16, 23-25, 29-30

10 2-3, 5-10, 16-26, 30-31

11 2, 4-9, 19-27, 30, 31

12 2, 4-8, 20-31