LAPORAN PRAKTIKUM ARSITEKTUR DAN ORGANISASI KOMPUTER

CPU SIMULATOR

Agus Pranata Marpaung
13323033
DIII Teknologi Komputer

INSTITUT TEKNOLOGI DEL
FAKULTAS VOKASI

Judul Praktikum

Minggu/Sesi	:	V/2
Kode Mata Kuliah	:	1031103
Nama Mata Kuliah	:	ARSITEKTUR DAN ORGANISASI KOMPUTER
Setoran	:	Laporan Materi CPU Simulator dikirimkan dalam bentuk PDF dan beri nama "*Prak4_CPU_Simulator_NIM.pdf*"
Batas Waktu Setoran	:	*3 Oktober 2023 jam 8:00*
Tujuan	*:*	*1. Use the CPU simulator to create basic CPU instructions.* *2. Use the simulator to execute basic CPU instructions.* *3. Use CPU instructions to move data to registers,compare values in registers, push data to the stack,pop data from the stack,jump to address locations and add values held in registers.* *4. Explain the functions of special CPU registers such as the PC,SR and SP registers.*

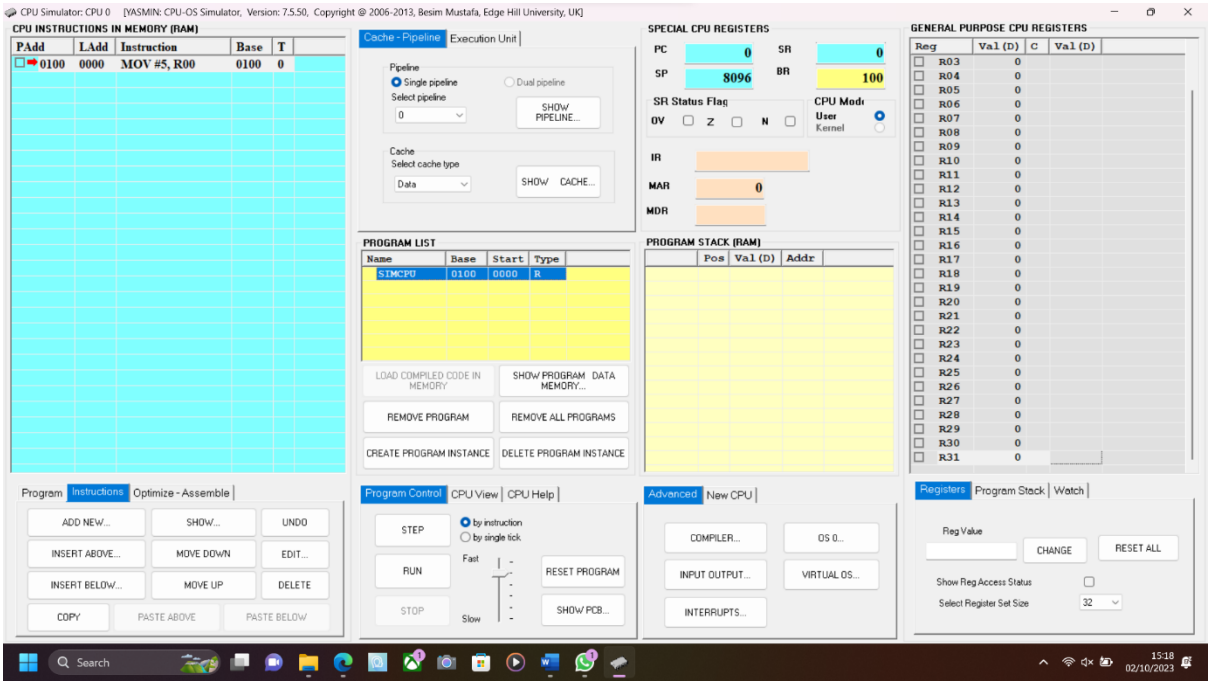
Petunjuk

1. Laporan praktikum dikerjakan secara individu (tidak berkelompok)
2. Setiap individu diperbolehkan memberikan pertanyaan dan diskusi melalui WAG pada sesi kedua di hari praktikum
3. Lapoan praktikum akan dikirimkan sesi praktikum melalui e-cource dan mengikutiformat yang telah disediakan sebelumnya
4. Tidak ada toleransi keterlambatan, jika terlambat maka akan terjadi pengurangan nilai.
5. Dalam pengerjaan laporan praktikum, dilarang keras melakukan plagiasi(mencontek).

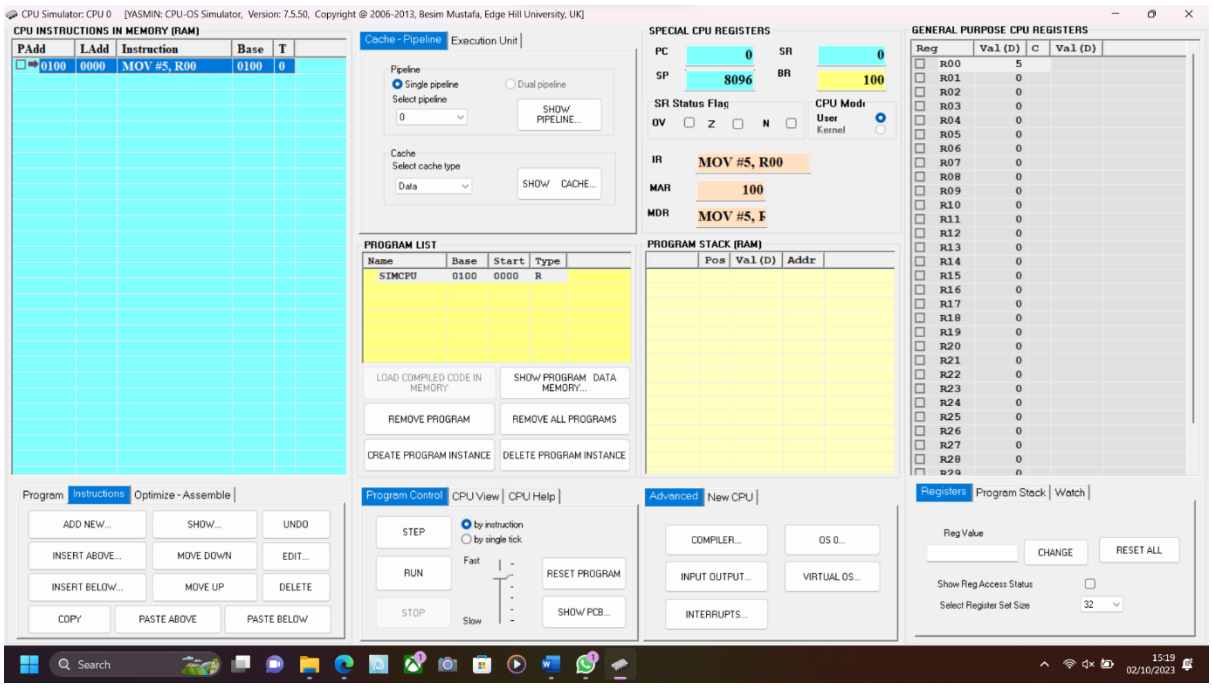
Arsitektur dan Organisasi Komputer

Tugas

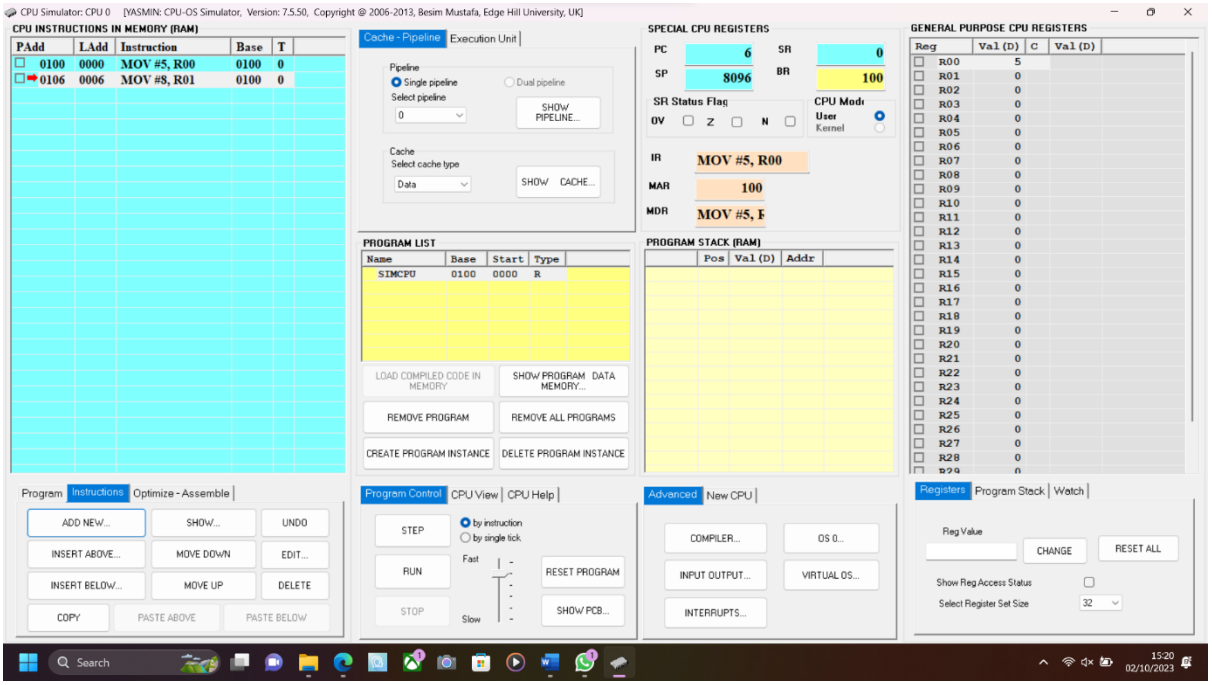
1. Create an instruction, which moves number 5 to register R00.



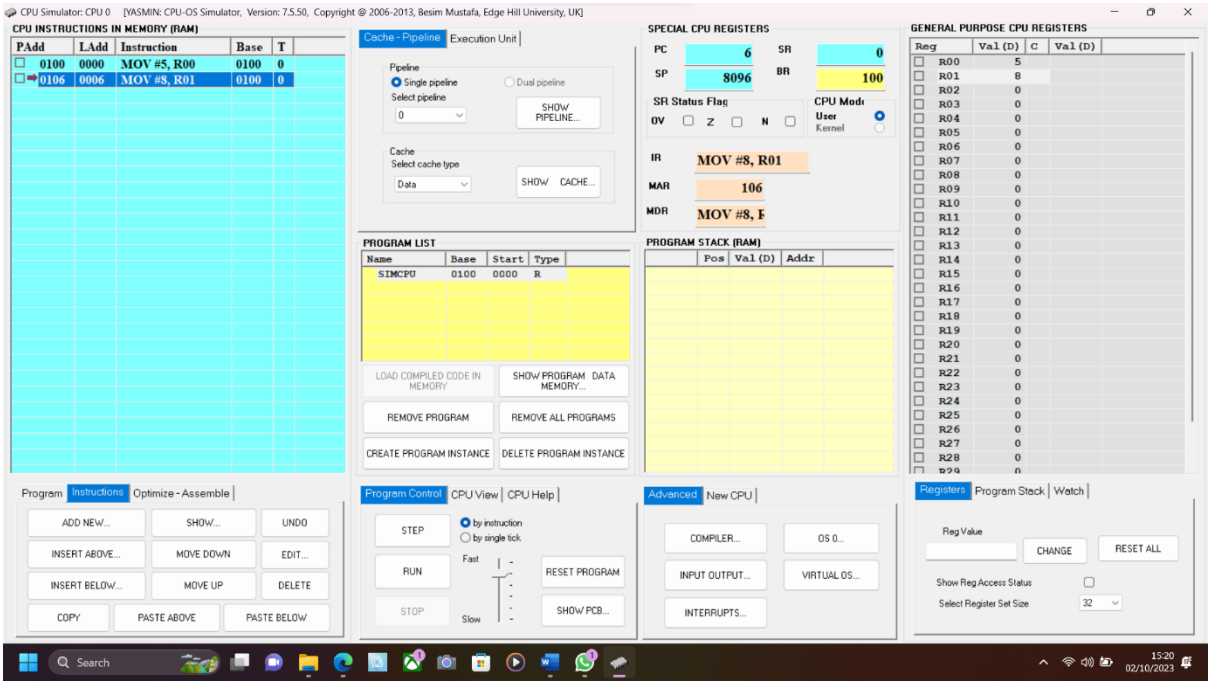
2. Execute the above instruction (to do this simply double click on it in the **Instruction Memory View**). Observe the result in the **CPU Registers** view (Image 4).



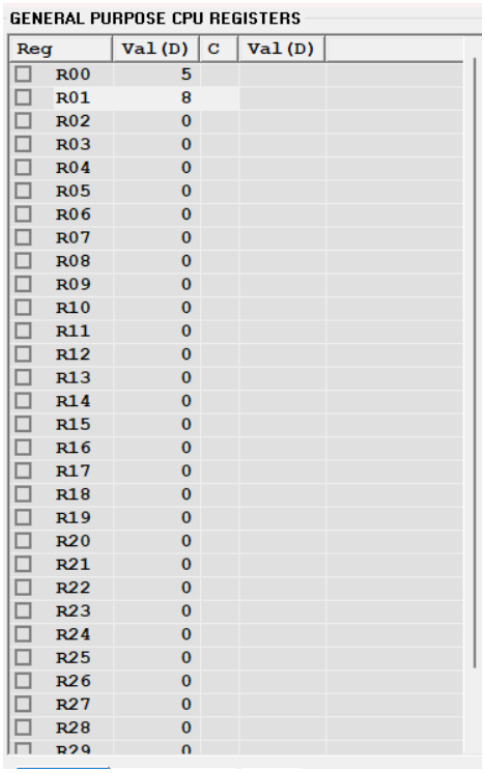
3. Create an instruction, which moves number 8 to register R01.



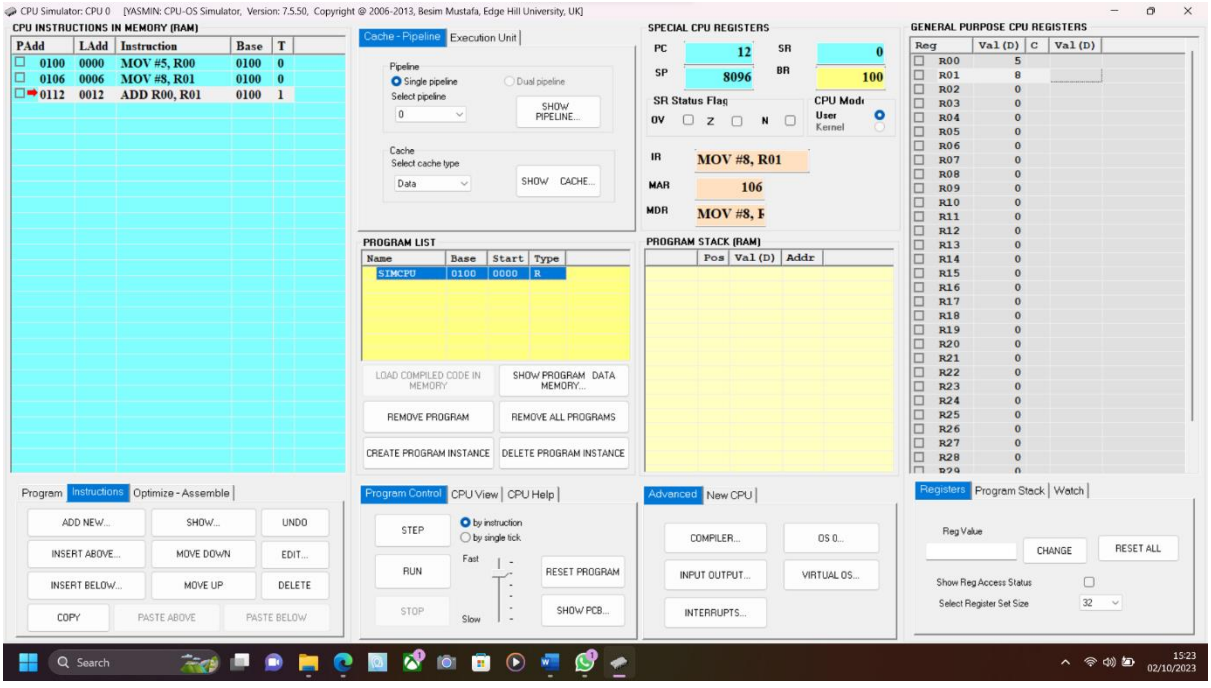
4. Execute it (You do this by double-clicking on the instruction).



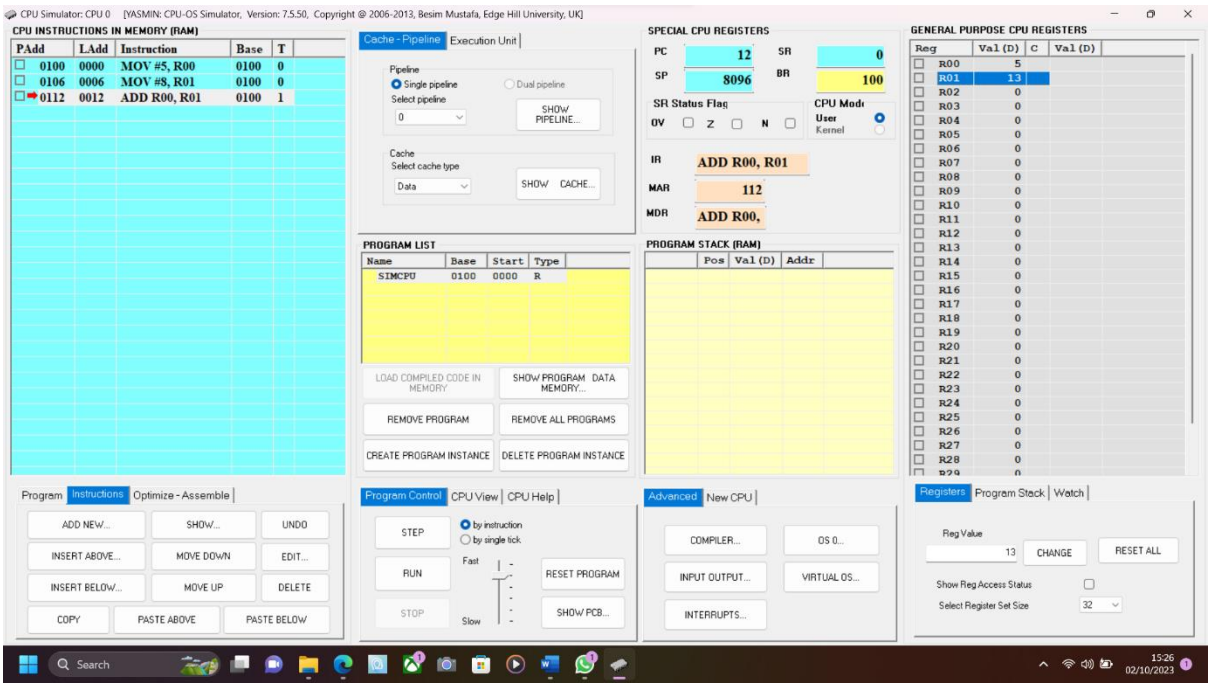
5. Observe the contents of R00 and R01 in the **CPU Registers** view (Image 4).



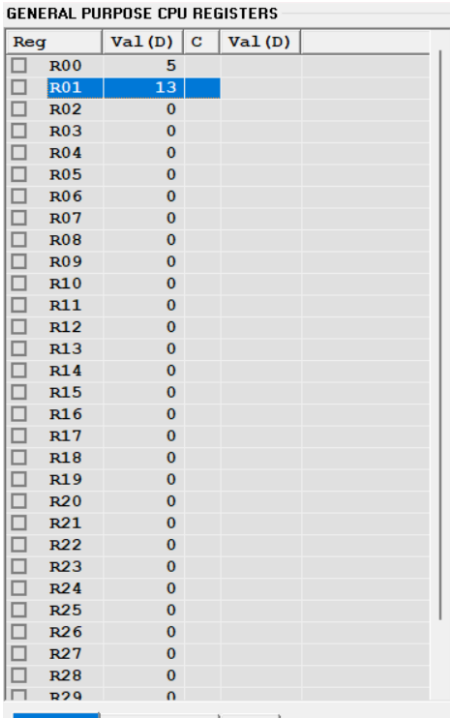
6. Create an instruction, which <u>adds</u> the contents of R00 and R01.



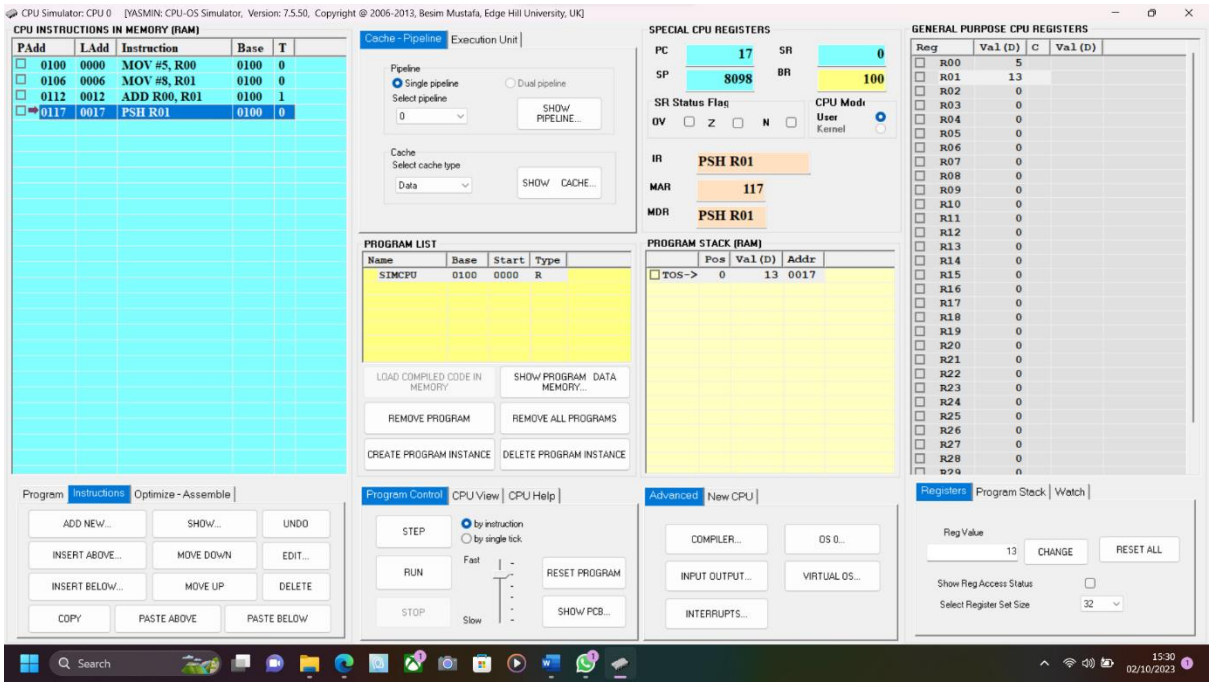
7. Execute it.



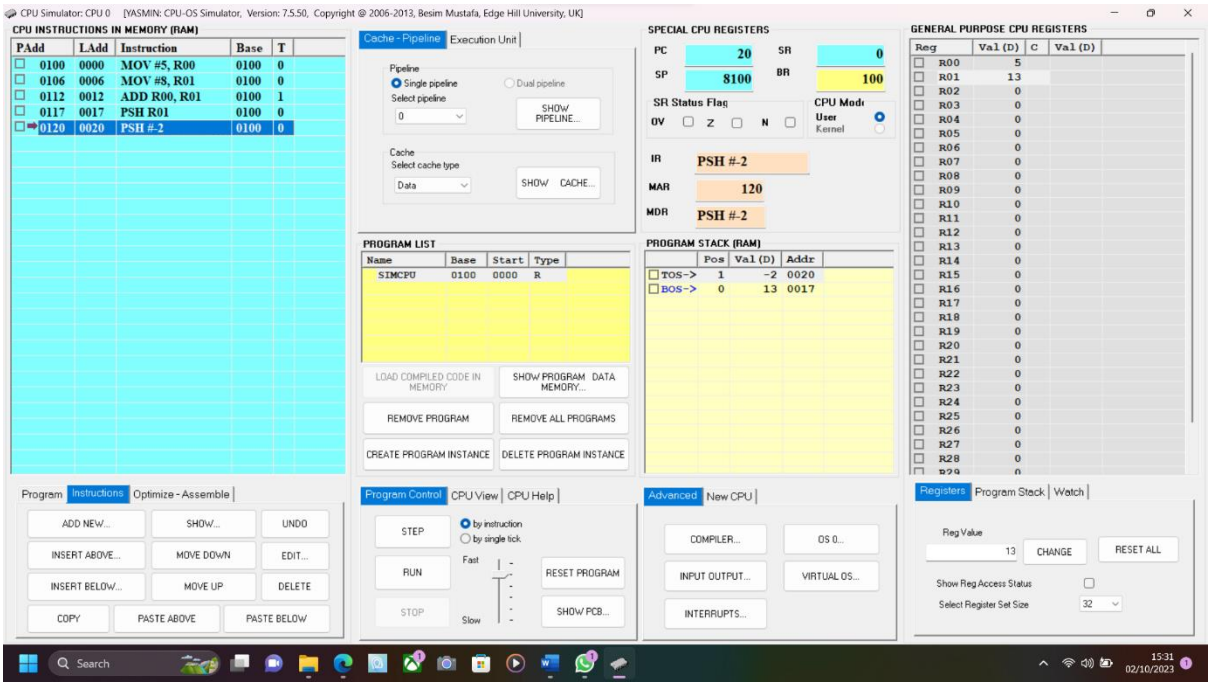
8. Observe which register the result is put in.



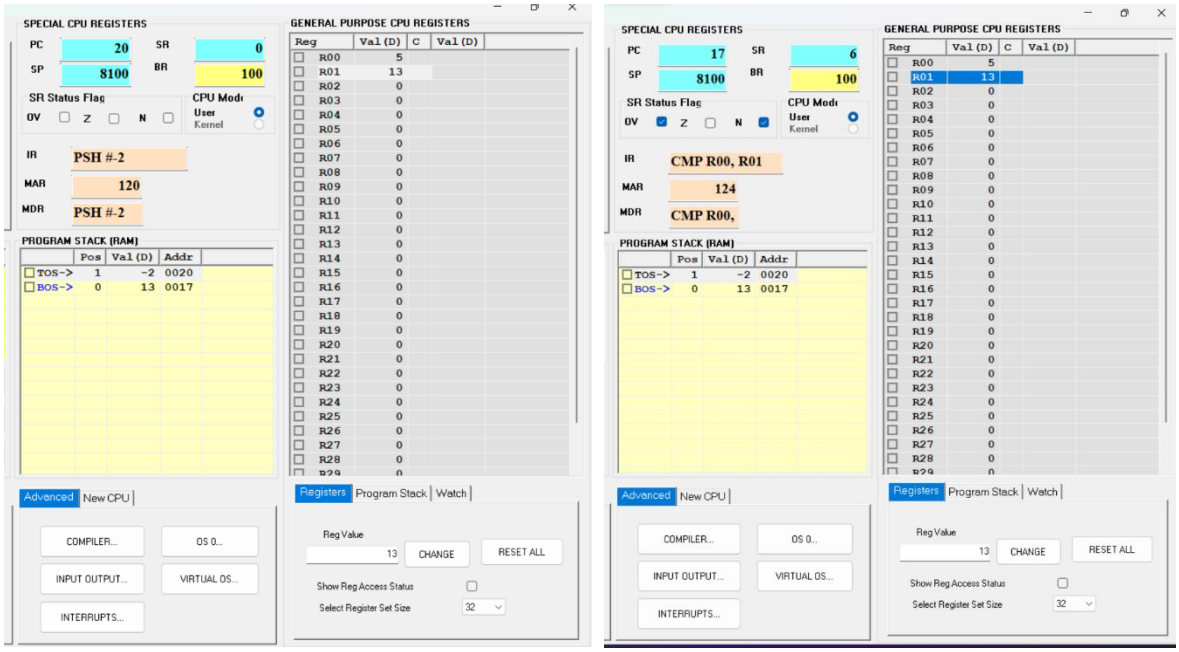
9. Create an instruction, which pushes the above result to the top of the hardware stack, and then execute it.



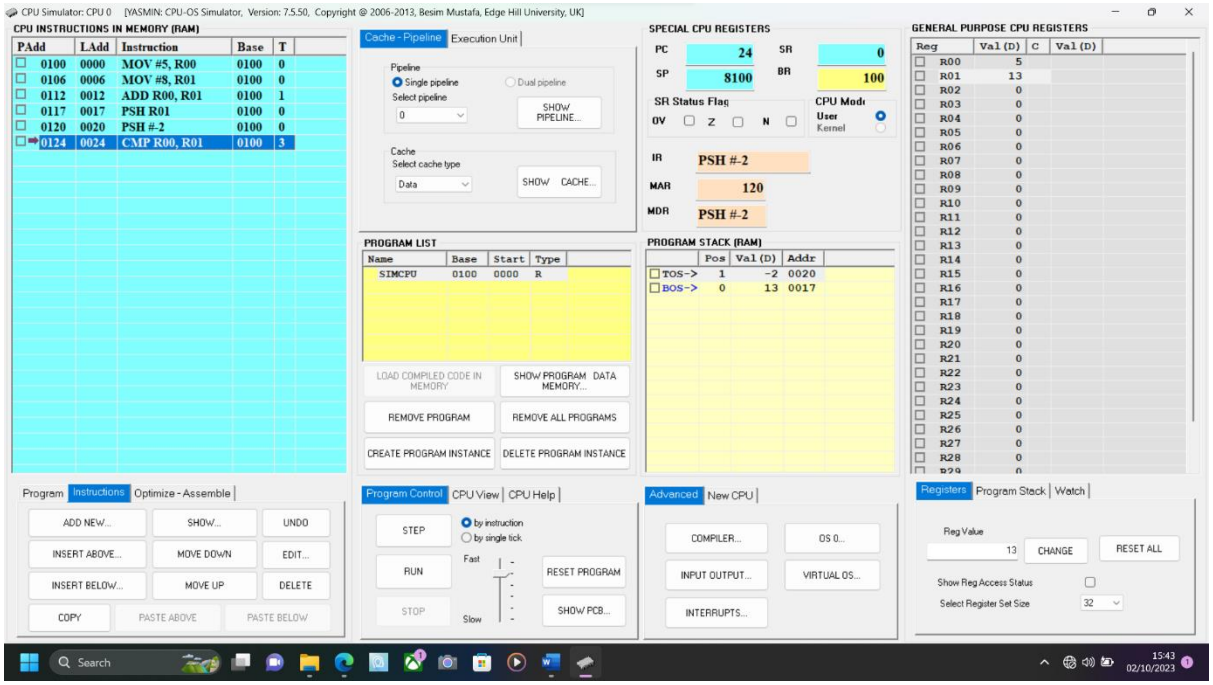
10. Create an instruction to push number-2 on top of the stack and execute it. Observe the value in **Program Stack** (Image 5).



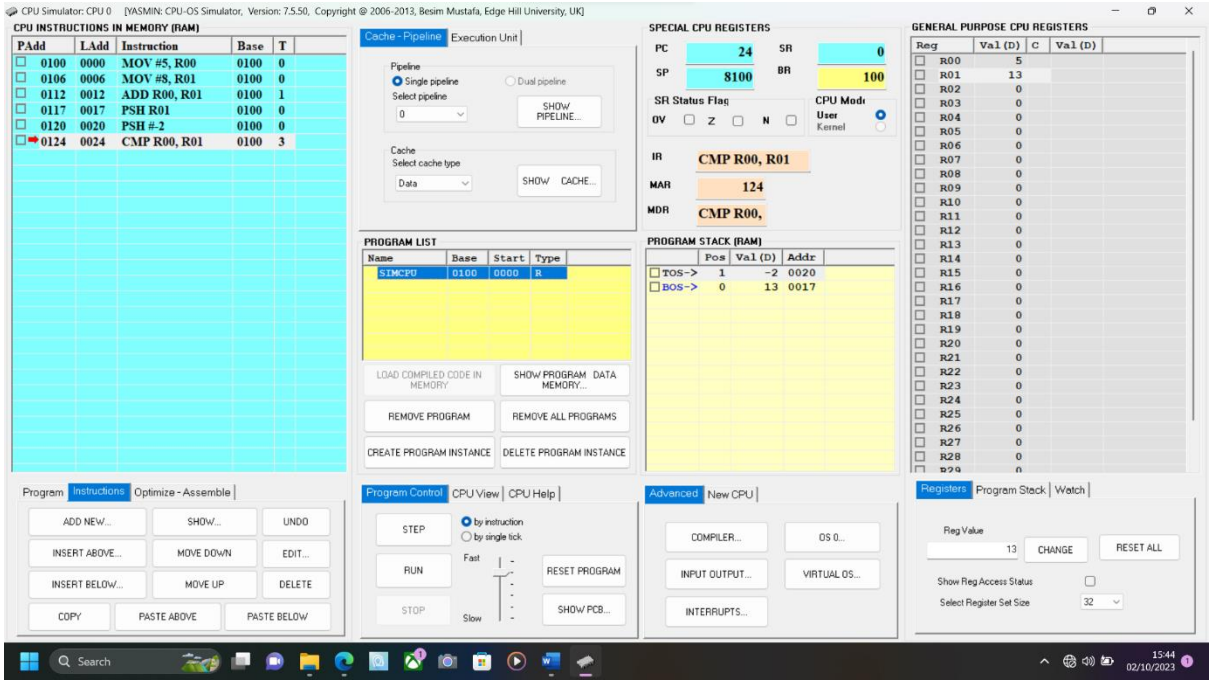
11. Observe the value in the **SP** register (**Special CPU Registers** view – Image 3). Whenever you push a value on **Program Stack**, the **SP** register is updated.



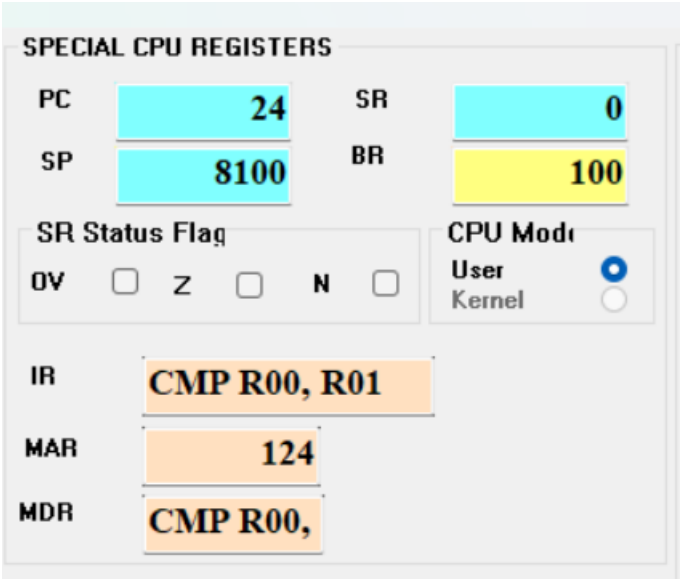
12. Create an instruction to <u>compare</u> the values in registers R00 and R01.



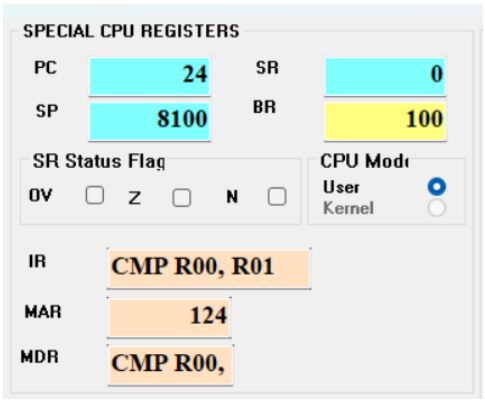
13. Execute it.

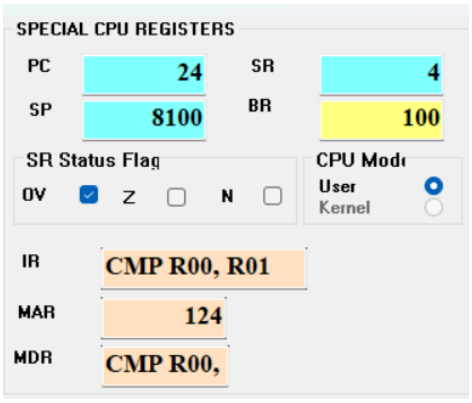


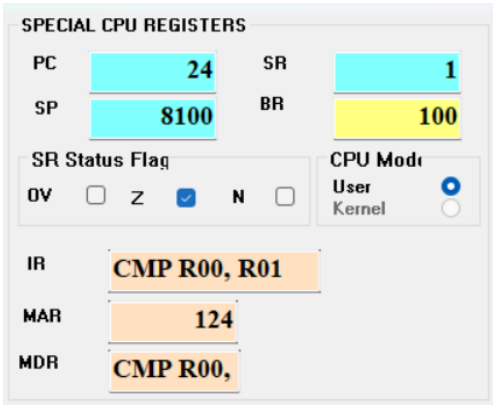
14. Observe the value in the **SR** register (**Special CPU Registers** view–Image 3).

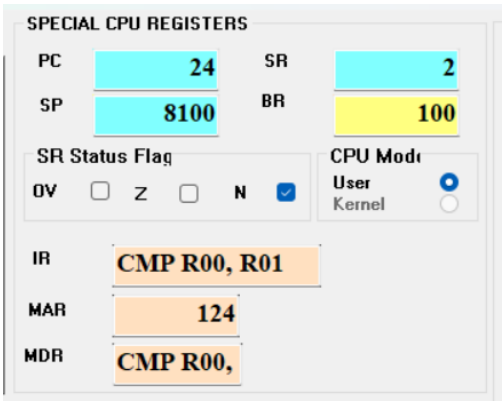


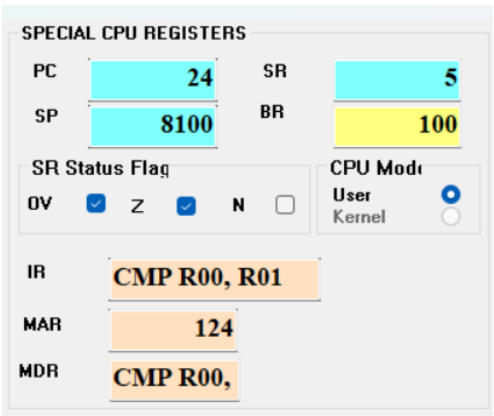
15. Observe the status of the **OV/Z/N** parts of the status register. Which boxes are checked and which are not? What do they indicate?

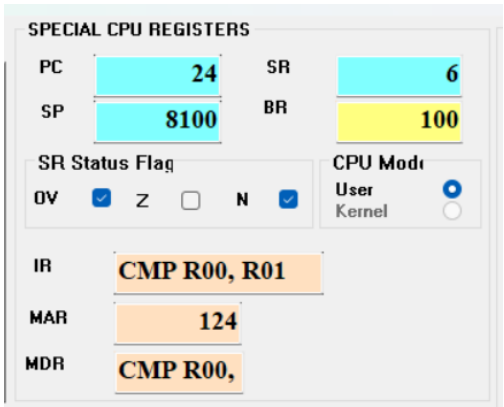


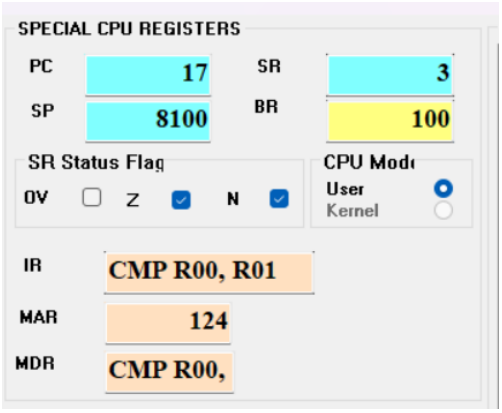


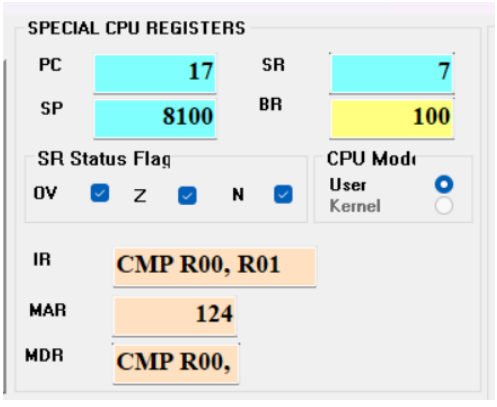




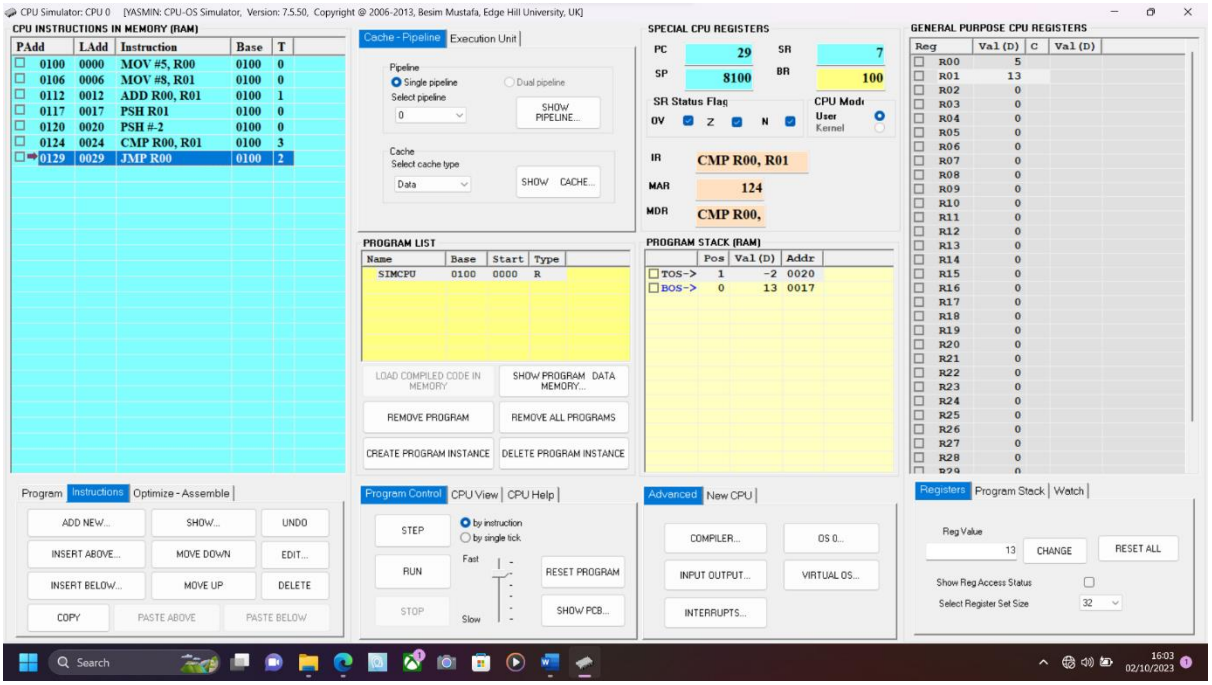




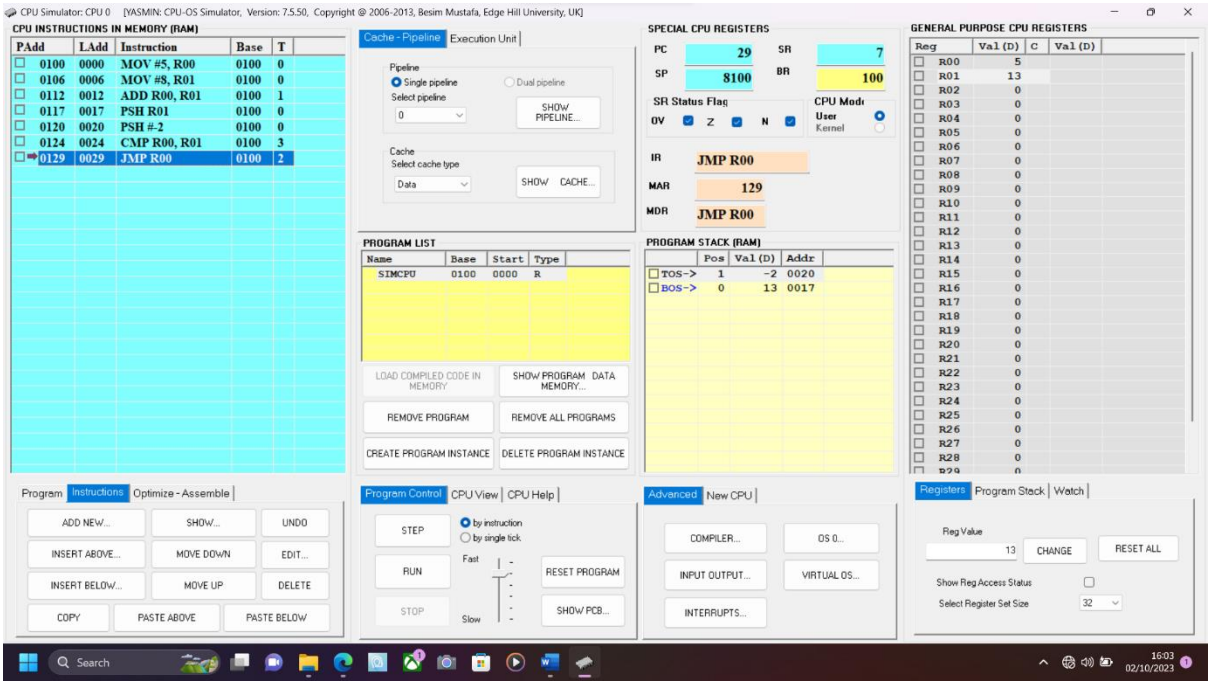




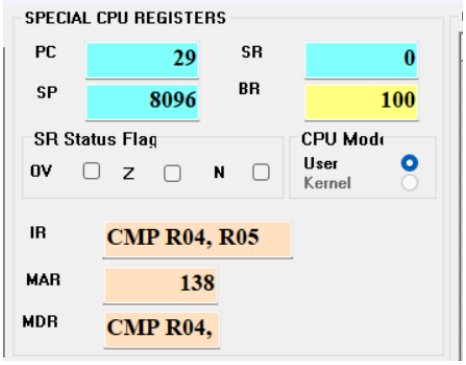
16. Create an instruction to unconditionally jump to the first instruction.



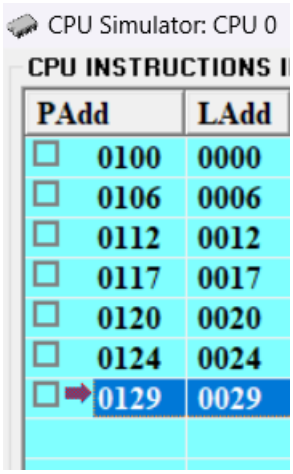
17. Execute it.



18. Observe the value in the **PC** register. This is the address ofthe nextinstruction to be executed. Make a note of which instruction itis pointing to?

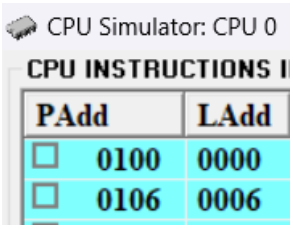


19. Observe the values in the **PAdd** and **LAdd** columns. What do these values indicate?Are they different (**Hint**: Check out the Base Address value)?

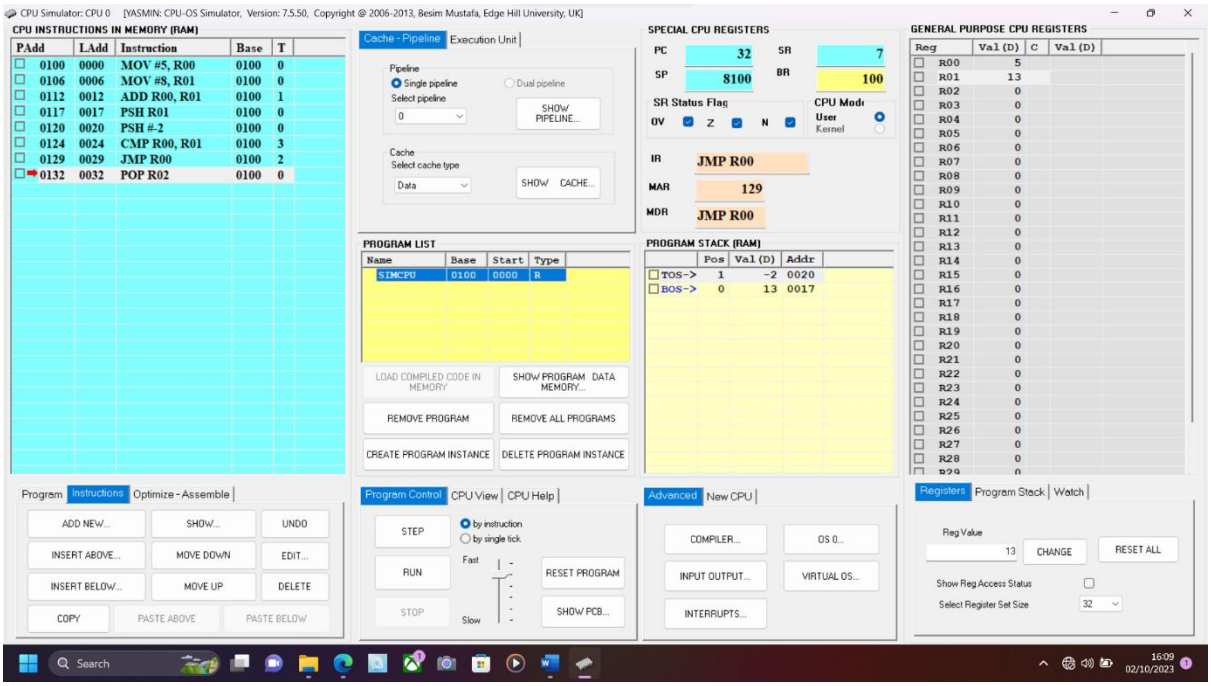


20. What is the difference between the **LAdd** value of the first instruction and the **LAdd** value ofthe second instruction? What does this value indicate (**Hint**: Think of the instruction lengths in bytes)?

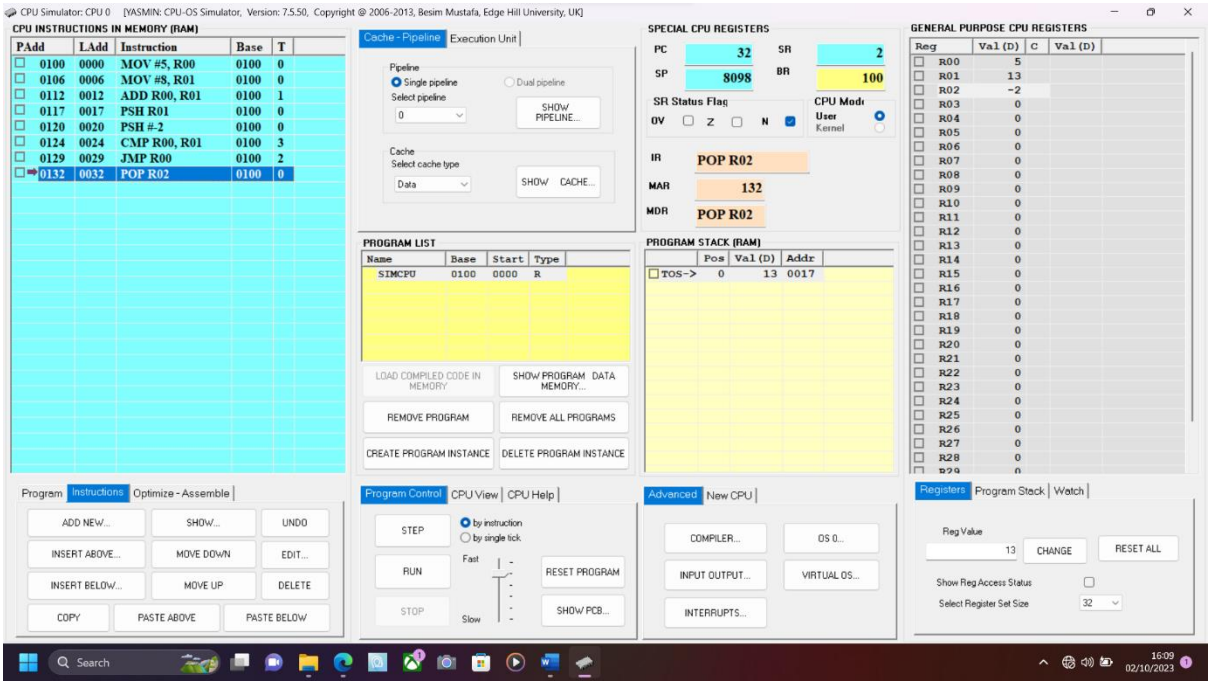
    Jawab: Nilai Ladd pertama dan kedua berbeda karna nilainya ditambah 6



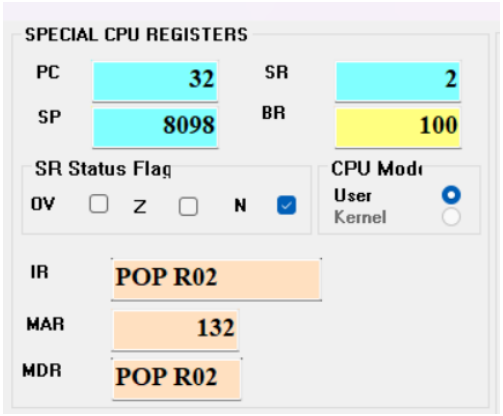
21. Create an instruction to pop the value on top of the **Program Stack** into register R02.



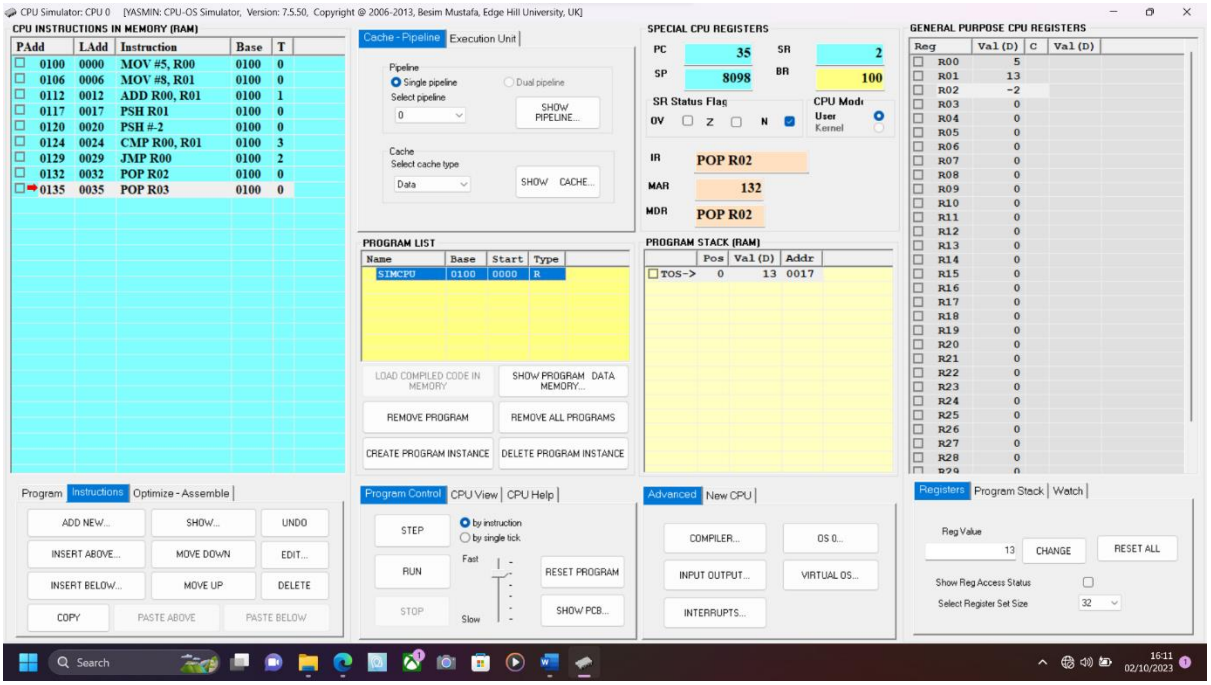
22. Execute it.



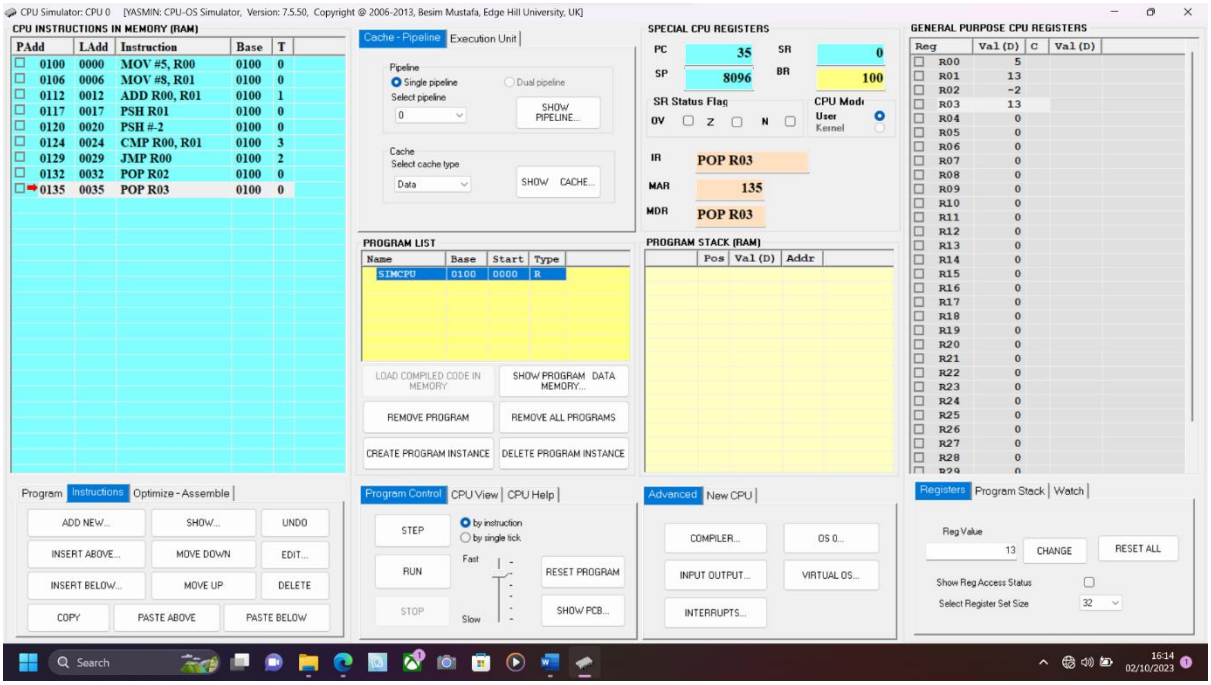
23. Observe the value in the **SP** register.



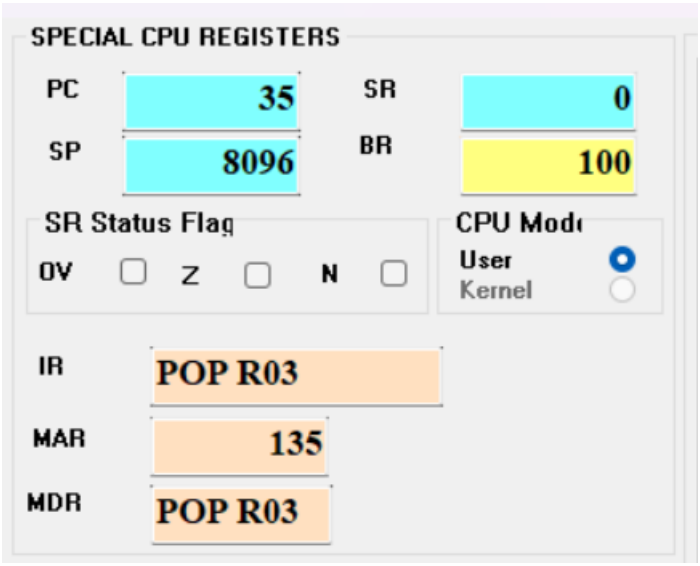
24. Create an instruction to pop the value on top of the **Program Stack** into register R03.



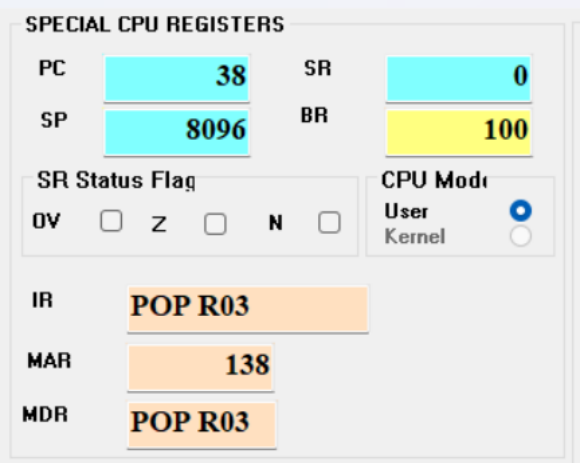
25. Execute it.



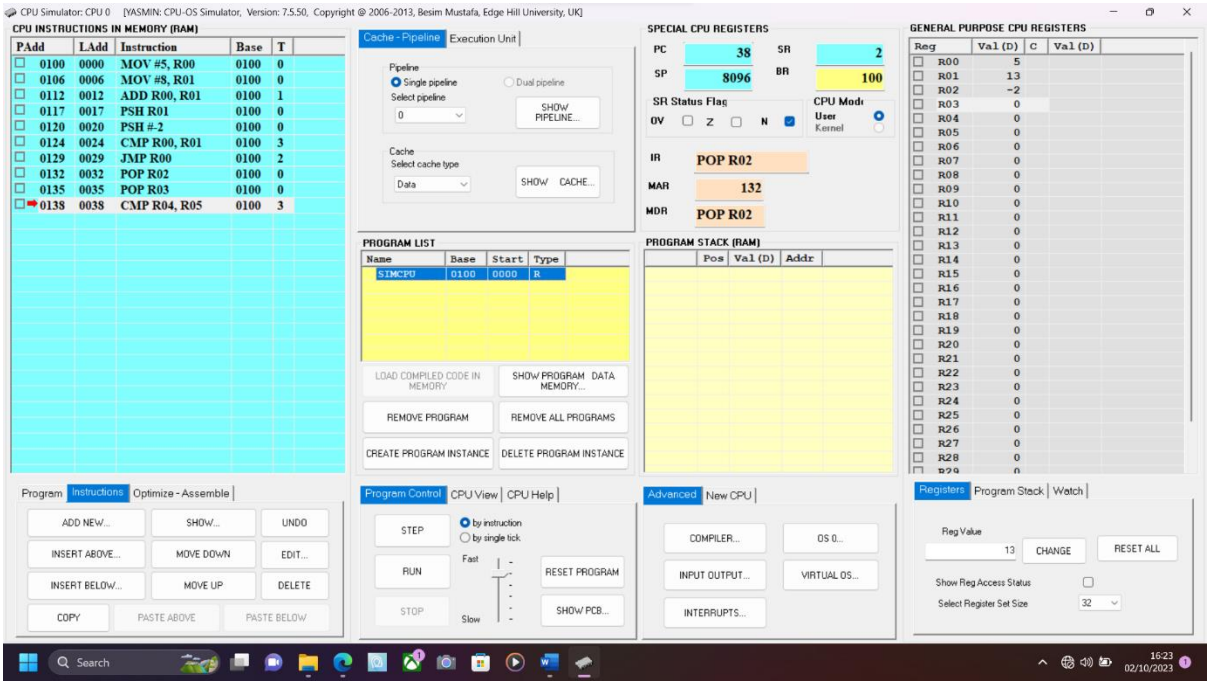
26. Observe the value in the **SP** register.



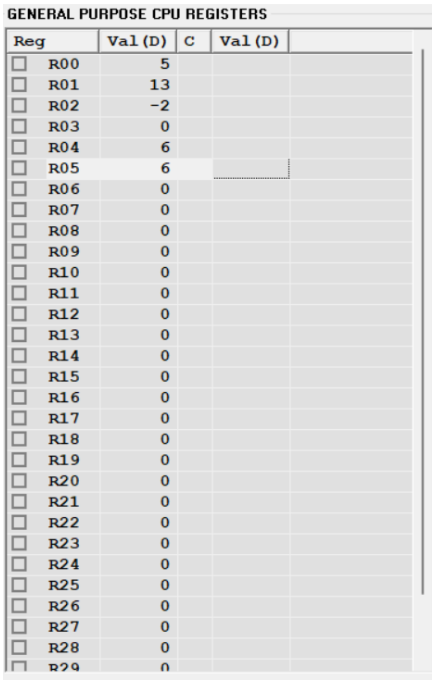
27. Execute the last instruction again. What happened? Explain.
    Jawab: Adanya perubahan pada PC dan MAR



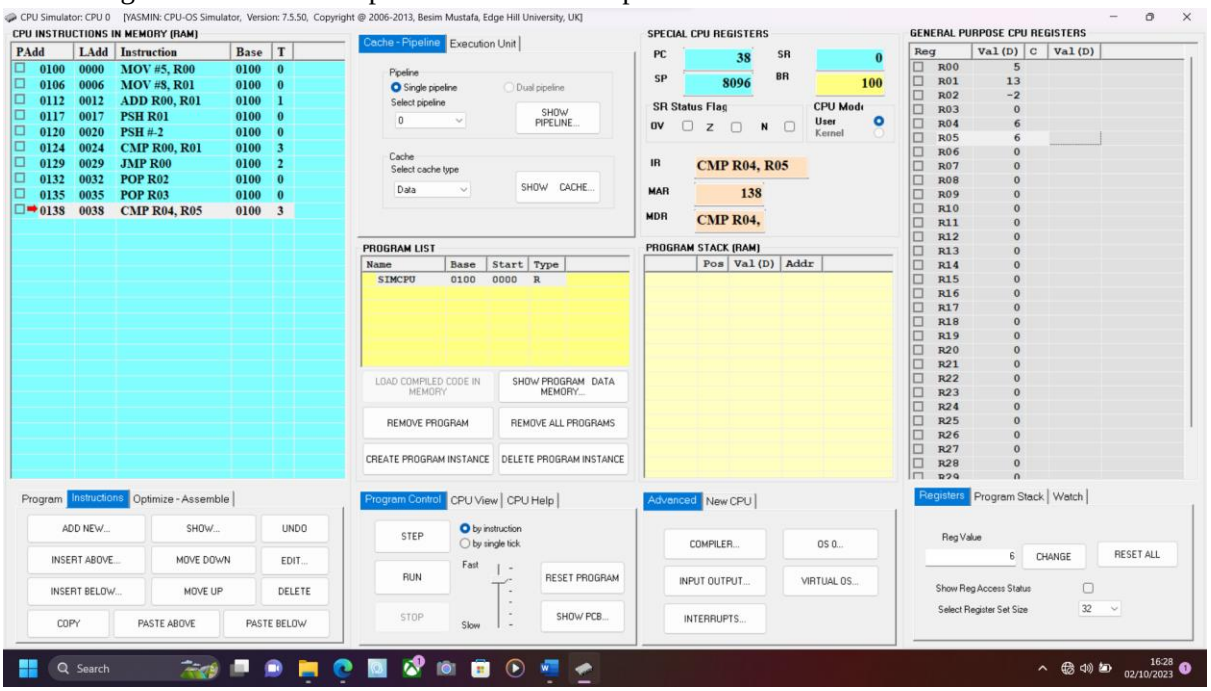
28. Create a compare instruction, which compares values in registers R04 and R05.



29. Manually insert two equal values in registers R04 and R05 (Image 4).

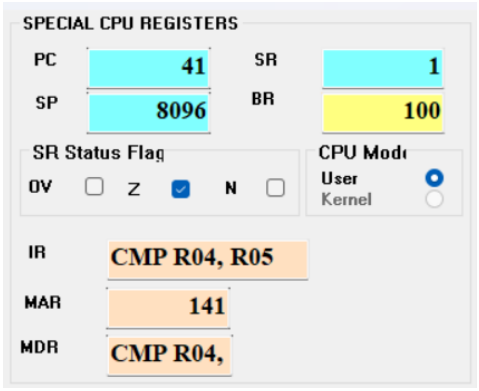


30. One again execute the compare instruction in step 28 above.

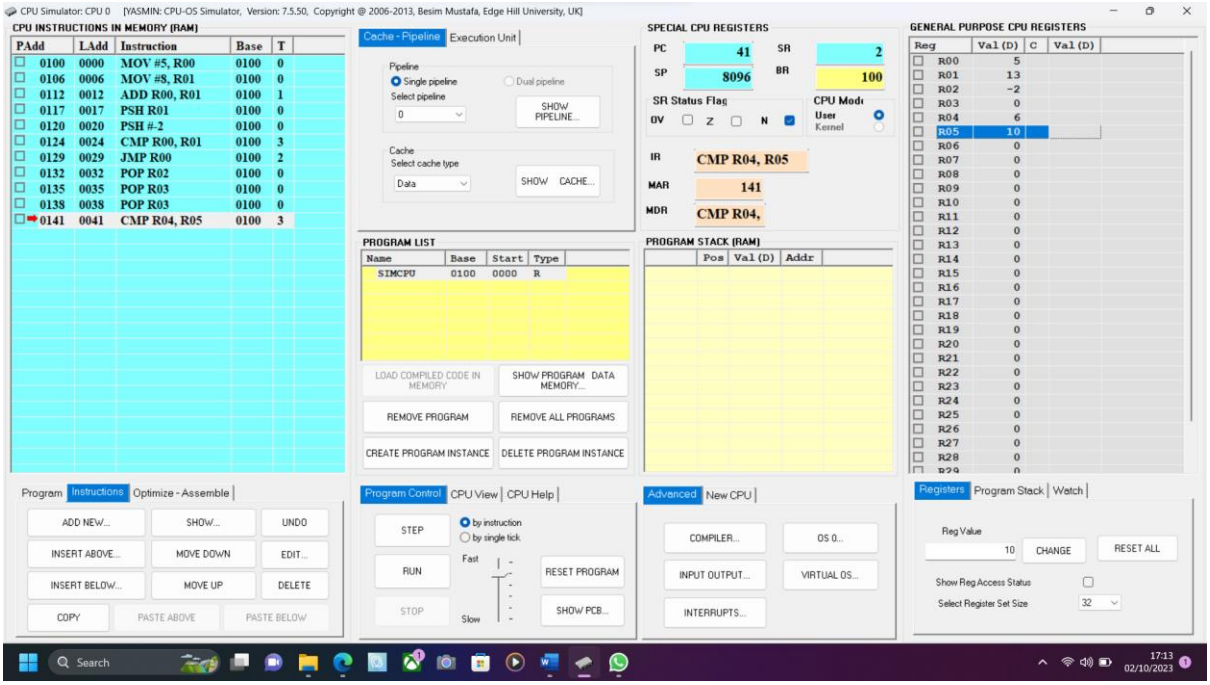


31. Which of the status flags **OV/Z/N** is set (i.e. box is checked)? Why?

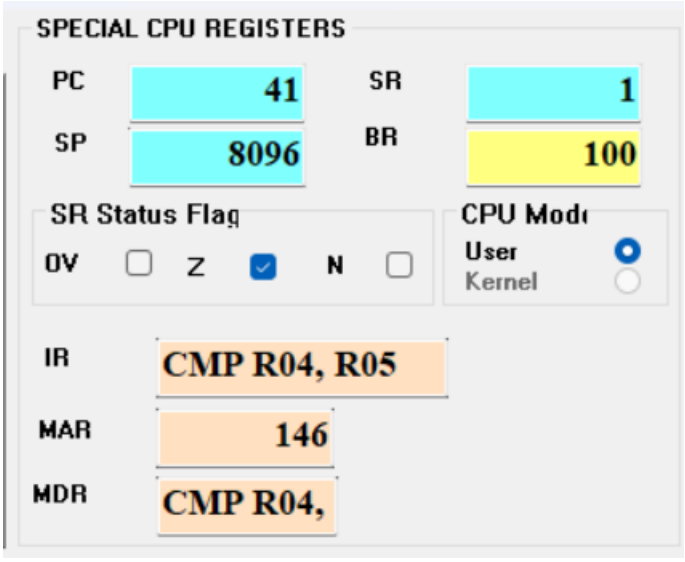
Jawab: Z menandakan zero. Karena R04=R05,maka yang dicentang adalah Z (zero).



32. Manually insert a value in register R05 greater than that in register R04.



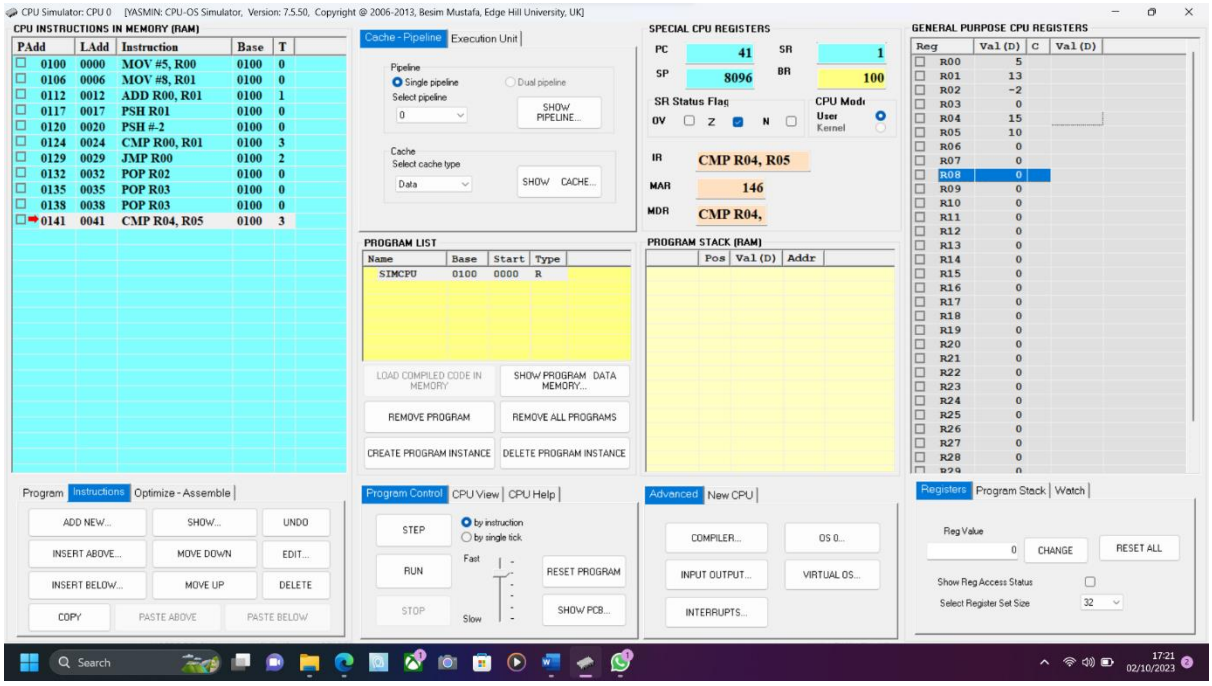
33. Execute the compare instruction in step 28 above.



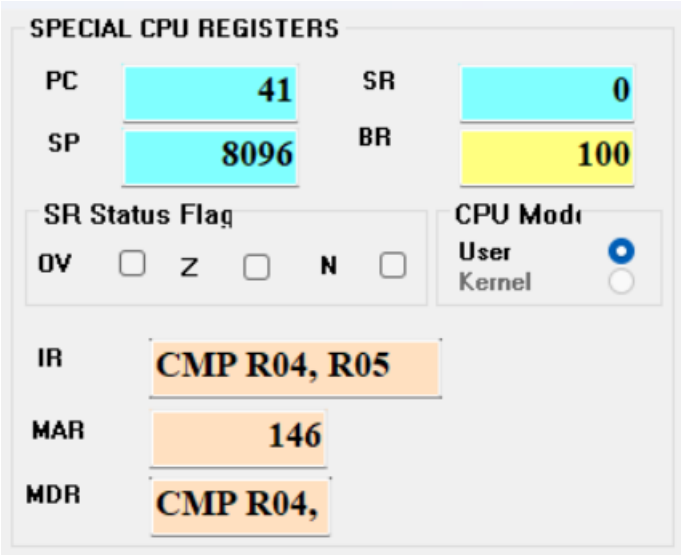
34. Which of the status flags **OV/Z/N** is set? Why?

    Jawab: Status Nol tidak lagi dicentang karena nilai yang telah saya masukkan sudah memiliki perbandingan (positif).

35. Manually insert a value in register R04 greater than that in register R05.

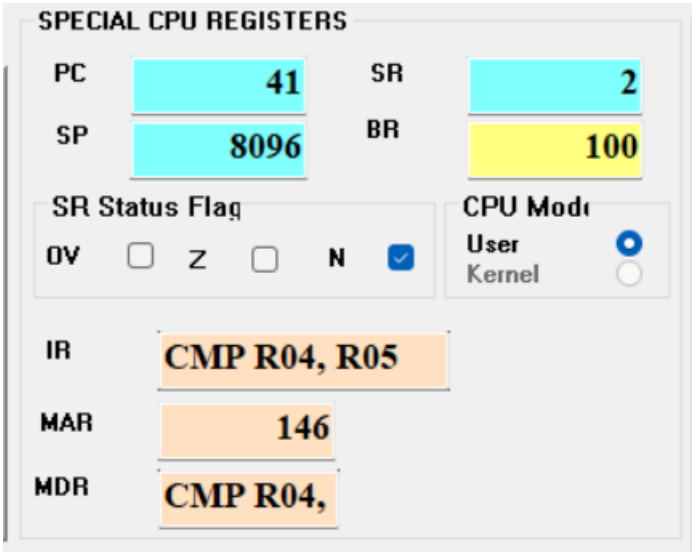


36. Execute the compare instruction in step 28 above.

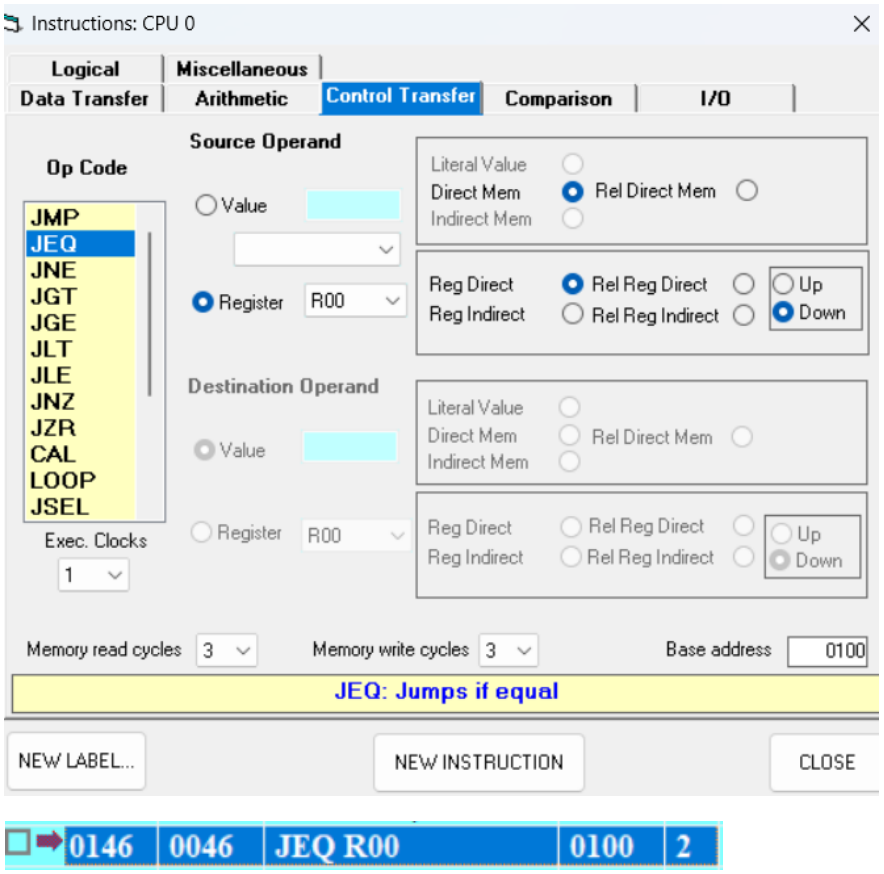


37. Which of the status flags **OV/Z/N** is set? Why?

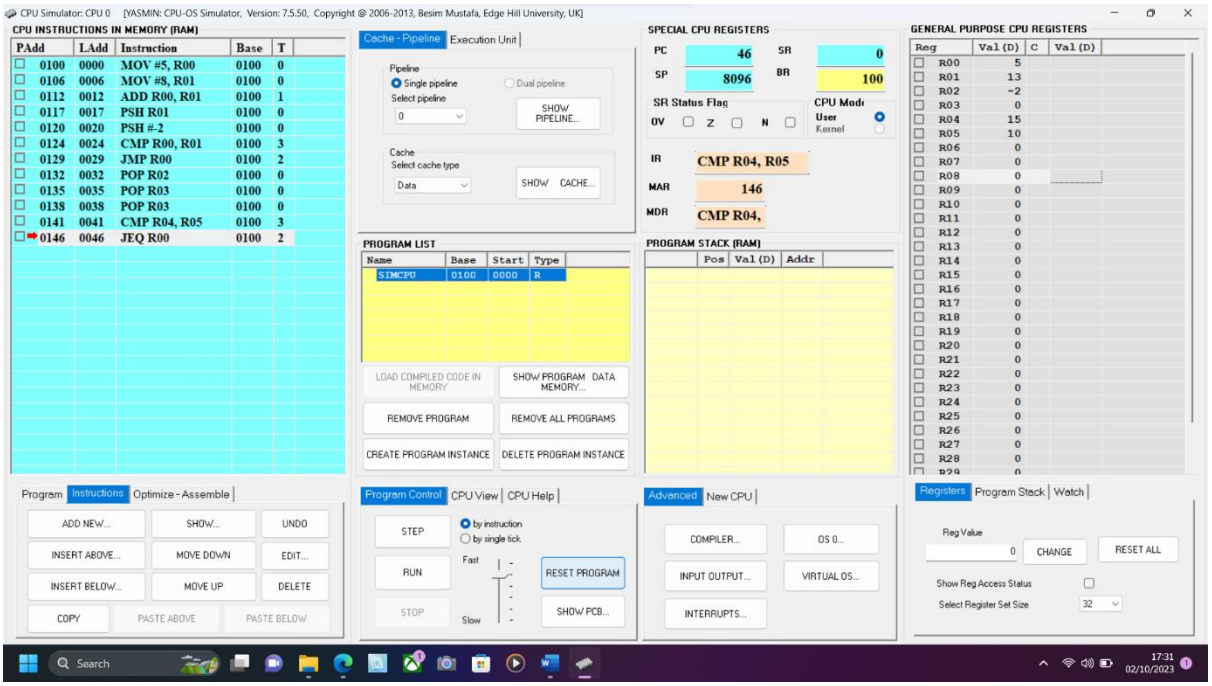
Jawab: SR status flag berubah menjadi N, karena R04 lebih besar dari R05



38. Create an instruction, which will jump to the first instruction if the values in registers R04 and R05 are equal.



39. Test the above instruction by manually putting equal values in registers R04 andR05, then first executing the compare instruction followed by executing the jump instruction (**Remember**: You execute an instruction by double-clicking on it). Did it work?



40. Save the instructions in the **Instruction Memory View** in a file by clicking on the **SAVE…** button (Image 7).