

# **LAPORAN PRAKTIKUM**

## **SISTEM OPERASI**

**(DOSEN PENGAMPU : IWAN LESMANA. S.KOM., M.KOM)**

### **MODUL 1**



**DISUSUN OLEH :**

**NAMA: MOHAMAD ABAN SY'BANA**

**NIM : 20230810012**

**KELAS : TINFC-2023-04**

**TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

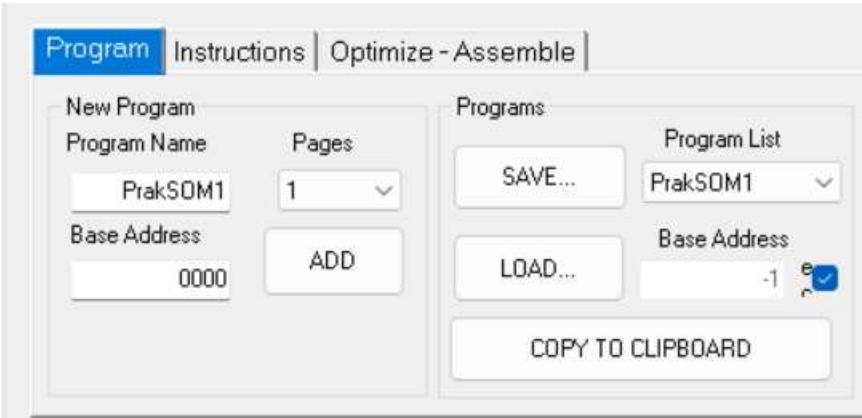
**UNIVERSITAS KUNINGAN**

**2024**

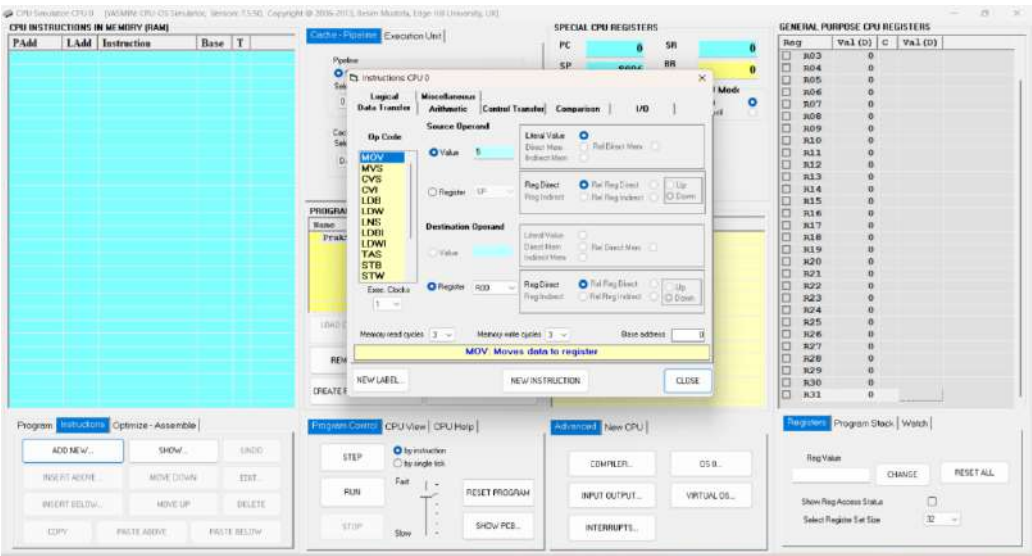
Praktikum

a. Transfer Data

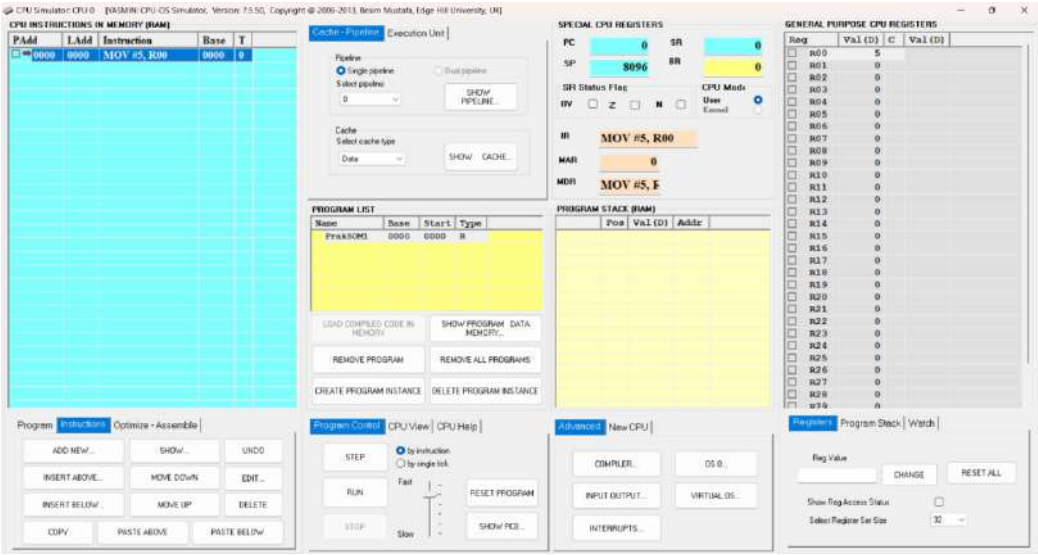
- 1. Buatlah Instruksi yang memindahkan (move) angka 5 Ke register R00
  - Pertama buka software yang Bernama Cpu Simulator yg telah di download kemudian ke bagian bawah kiri di bagian program isi kan Program Name dan Base Address setelah di isi lalu klik ADD lalu nanti di PROGRAM LIST ada nama folder yang telah kalian isi.



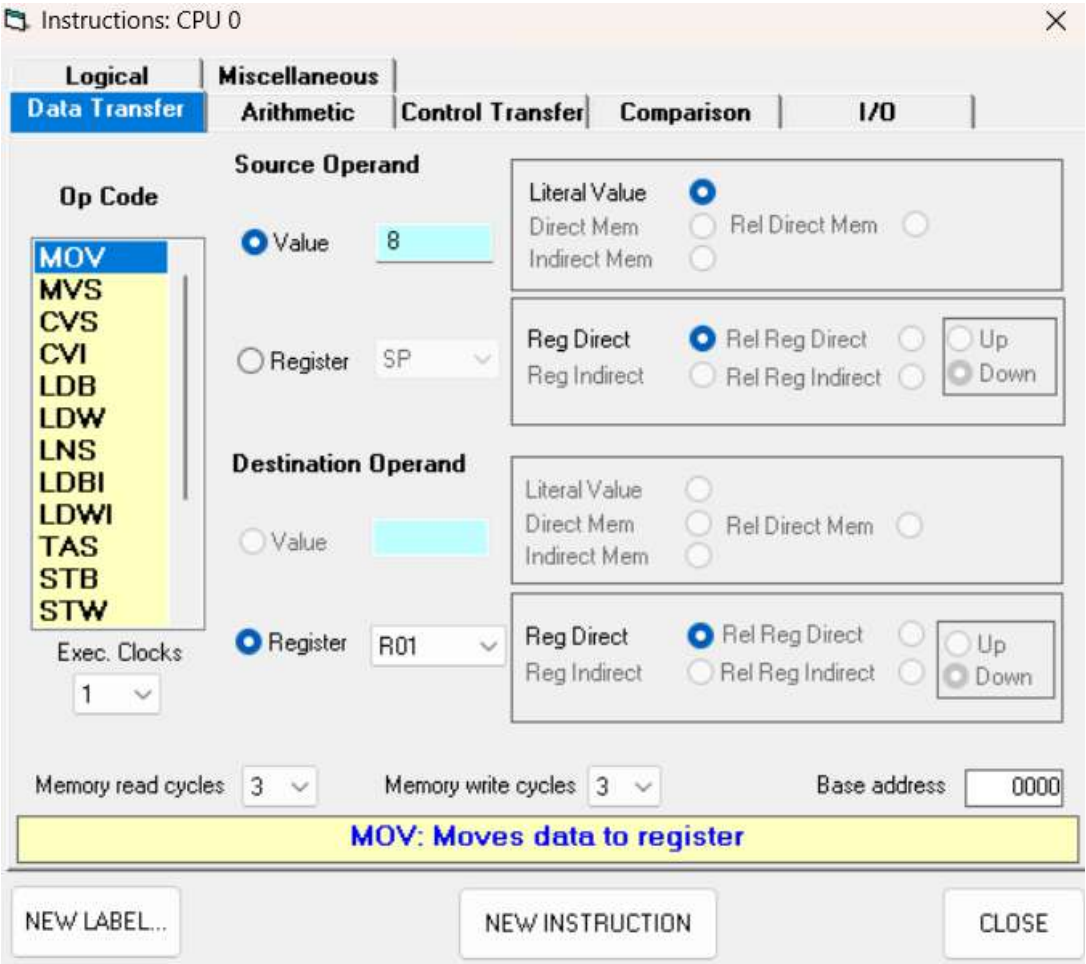
- Lalu ke bagian kiri bawah klik Intructions lalu klik ADD NEW setelah itu di OP CODE pilih MOV di bagian Source Operand isi Value 5 setelah di isi, di Destination Operand di bagian register pilih R00 lalu klik NEW INSTRUCTION.

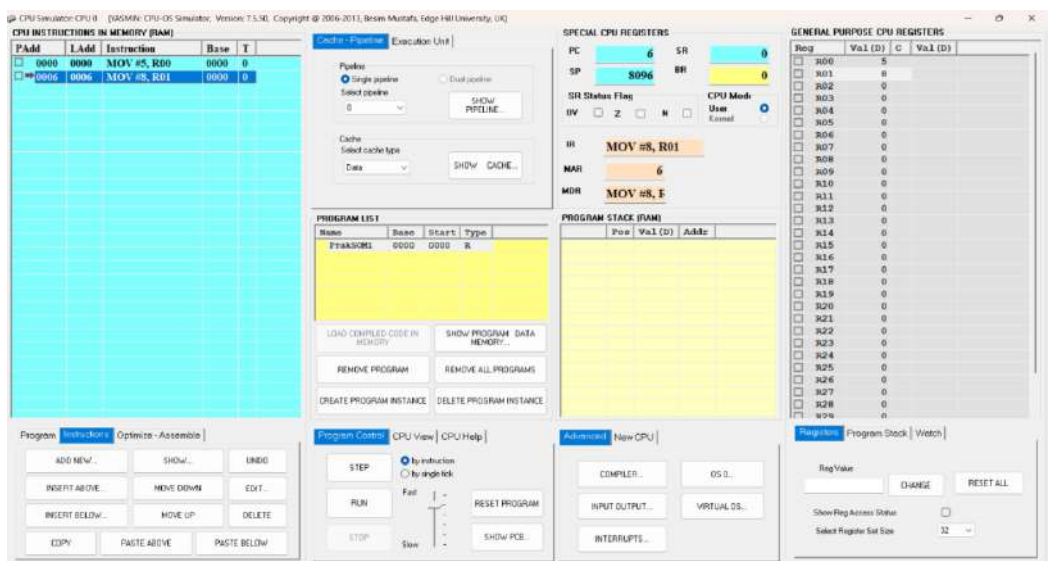


- Setelah di NEW INSTRUCTION lalu di bagian CPU INSTRUCTION IN MEMORY (RAM) double klik lalu lihat di GENERAL PURPOSE CPU REGISTERS di R00 akan memiliki Val(D) dengan value yang tadi telah di isi.



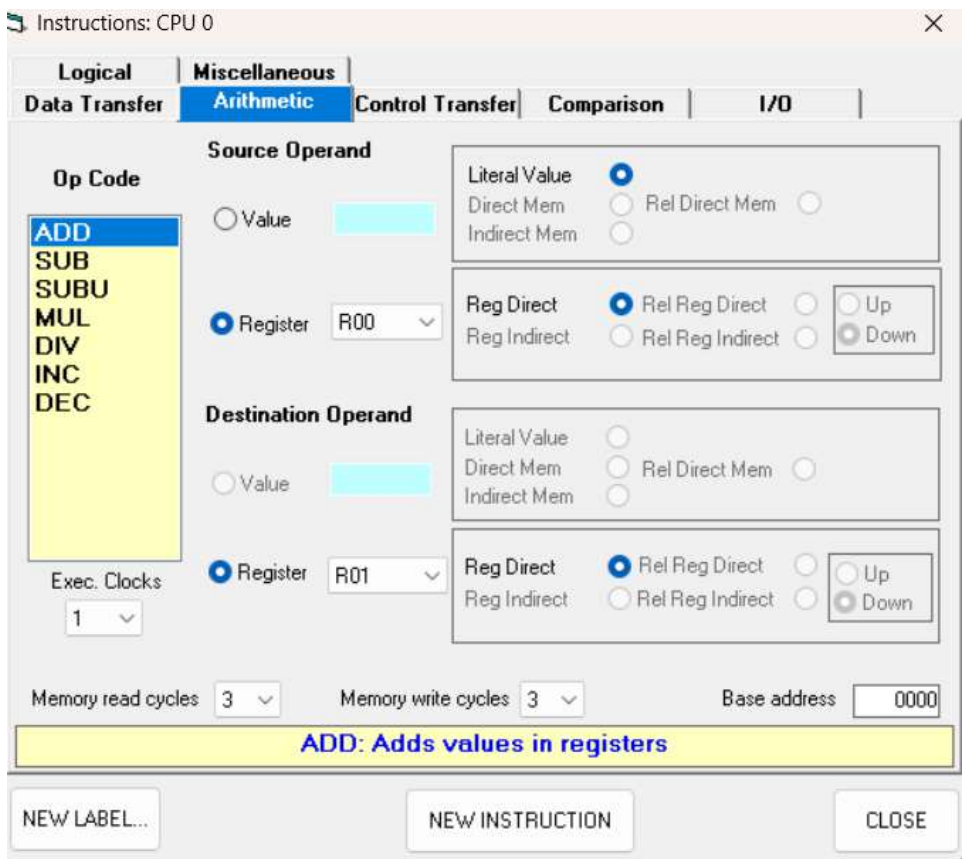
2. Buatlah Instruksi yang memindahkan (move) angka 8 Ke register R01
- Ulangi hal yang sama tetapi dengan Value dan Register yang berbeda.



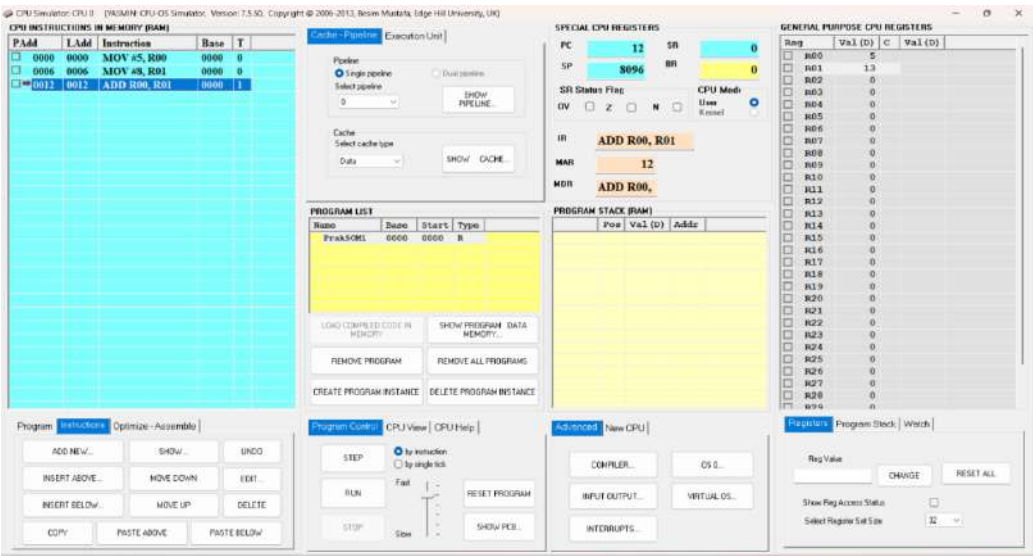


### b. Aritmatika

1. ADD NEW lalu pindah ke bagian Arithmetic setelah itu di Op Code pilih ADD dan di Source Operand pilih register dan isi dengan R00 dan di Destination Operand pilih register dan isi dengan R01 setelah di isi klik NEW INSTRUCTION.

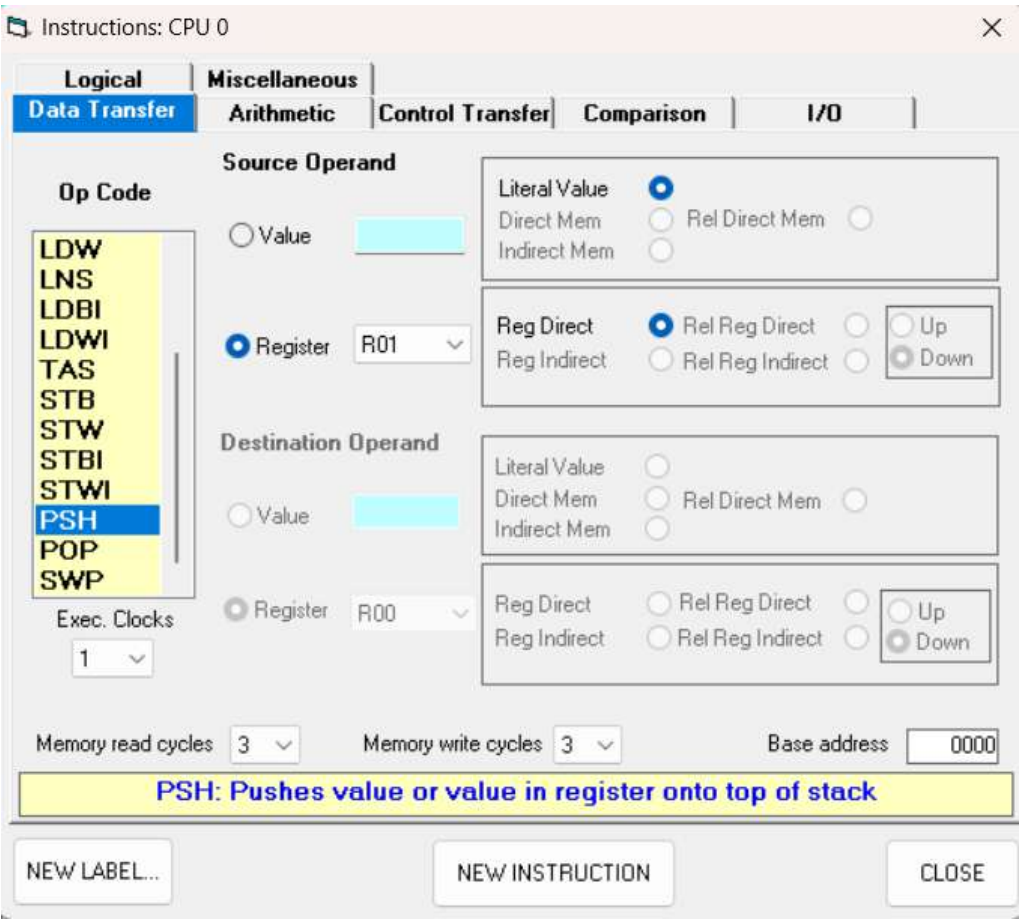


2. Lalu di CPU INSTRUCTIONS IN MEMORY (RAM) double klik lihat lagi di bagian GENERAL PURPOSE CPU REGISTERS ada perubahan di R01 yang sebelumnya nilai dari R01 adalah 8 akan berubah menjadi 13, hal itu di karenakan angka 8 pada R01 akan di jumlahkan dengan angka 5 yang ada pada di R00 dan hasil penjumlahan tersebut akan di simpan di R01.



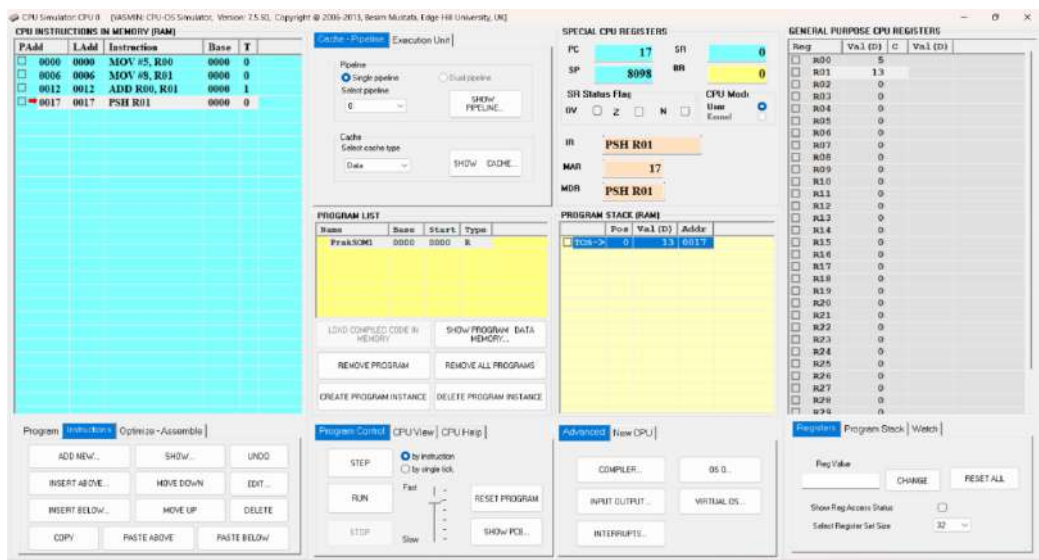
c. Stack Pointer (SP)

1. ADD NEW lagi lalu pilih data transfer lalu di Op Code scrool ke bawah pilih PSH dan di Source Operand klik register dan pilih R01 setalah itu klik NEW INSTRUCTION.

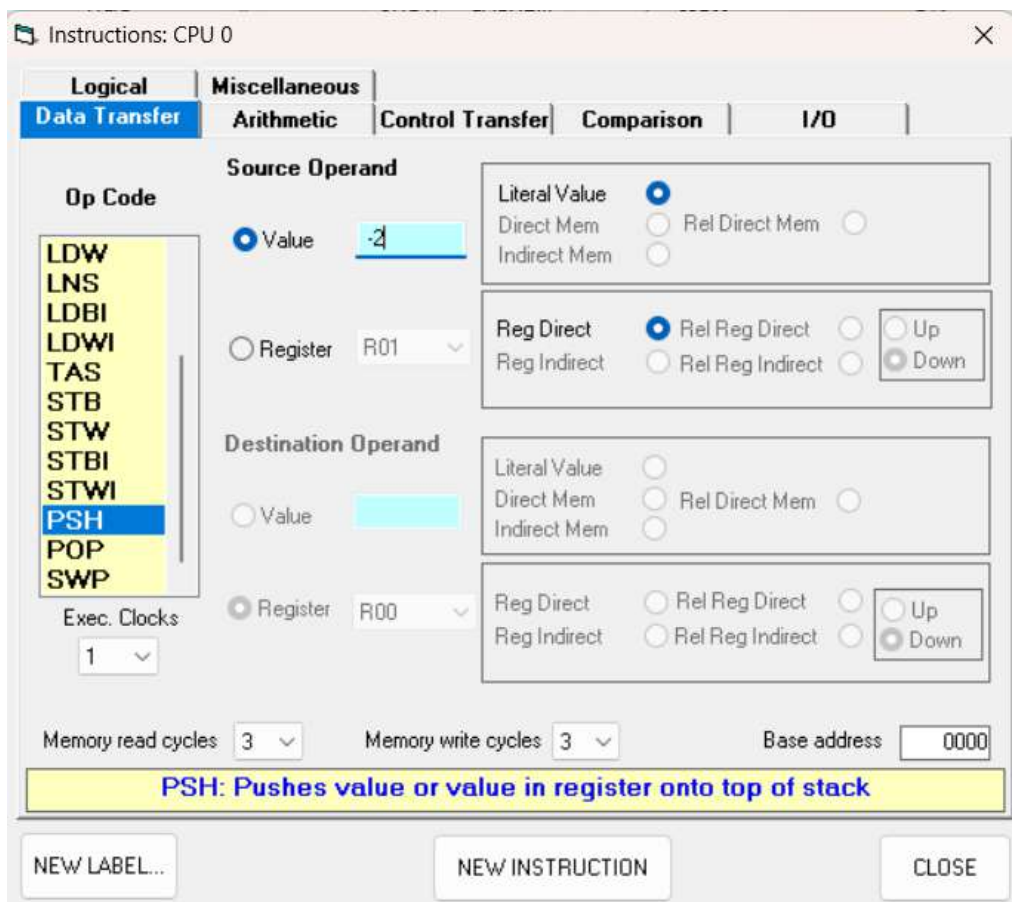




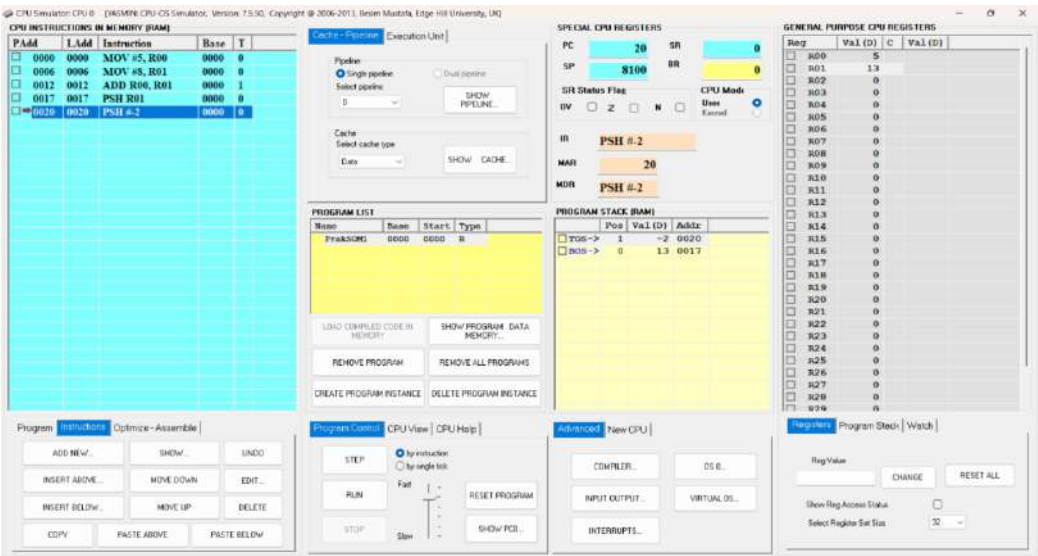
2. Di bagian CPU INSTRUCTIONS IN MEMORY (RAM) double klik Instruction PSH R01 lalu lihatlah di bagian PROGRAM STACK (RAM) akan muncul seperti gambar di bawah ini.



3. ADD NEW lagi lalu lakukan seperti tadi tapi isikan Source Operand nya dengan value -2 dan klik NEW INSTRUCTION.

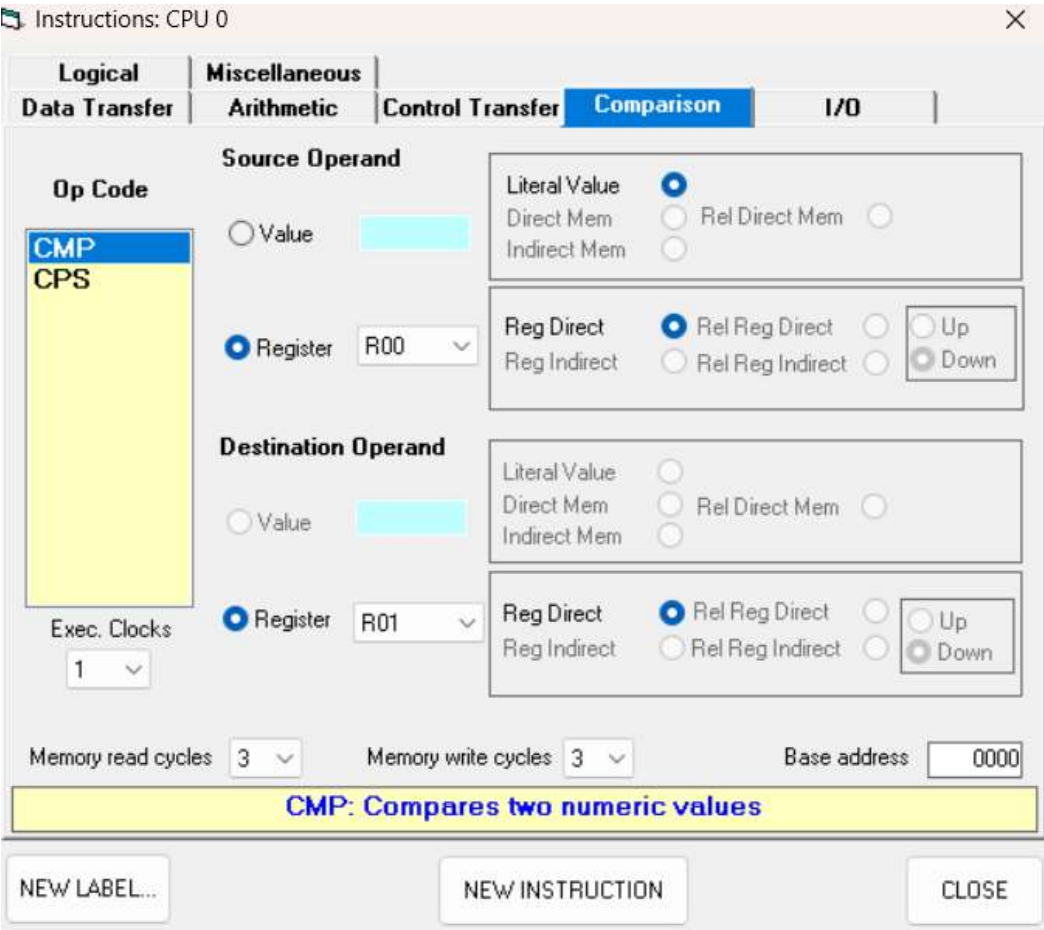


4. Double klik lagi di instruction PSH#-2 pada bagian SPECIAL CPU REGISTERS, nilai SP nya berubah menjadi 8100 yang sebelumnya 8098 di karenakan Setiap kali data didorong ke stack menggunakan instruksi PSH, nilai SP akan menyesuaikan untuk menunjuk ke lokasi berikutnya di memori dan di bagian PROGRAM STACK juga akan menampilkan nilai -2 dan akan muncul di bagian paling teratas.

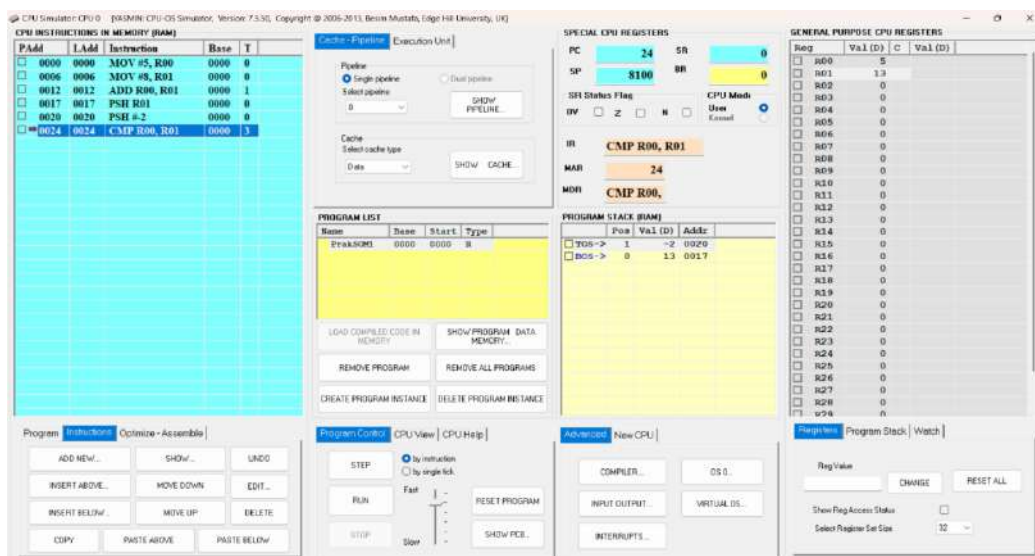


d. **Pembandingan**

1. ADD NEW lagi lalu pindah ke bagian Comparison dan Op Code nya pilih cmp dan isi register Source Operand dengan R00 dan register Destination Operand isi dengan R01 lalu klik NEW INSTRUCTION.

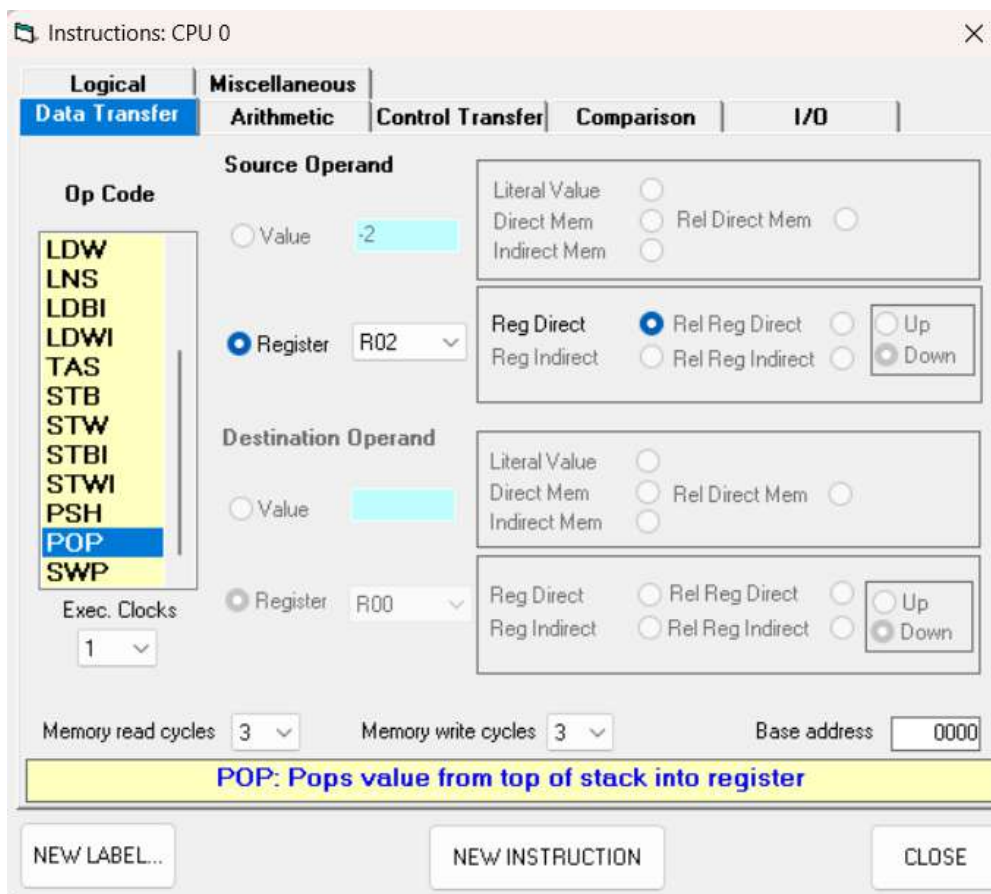


2. Double klik lagi di Instruction CMP R00,R01 Ketika instruksi dieksekusi, PC akan bertambah sesuai dengan panjang instruksi atau sesuai dengan alamat instruksi berikutnya. jika nilai berbeda dan nilai R00 lebih besar dari R01, maka bit Z dan N di bagian SR tidak akan di ceklis, dan SR akan tetap nol karena tidak ada kondisi zero atau negative yang terpenuhi begitupun dengan bit OV karena tidak hasil yang sama (zero), dan bit OV tetap nol.



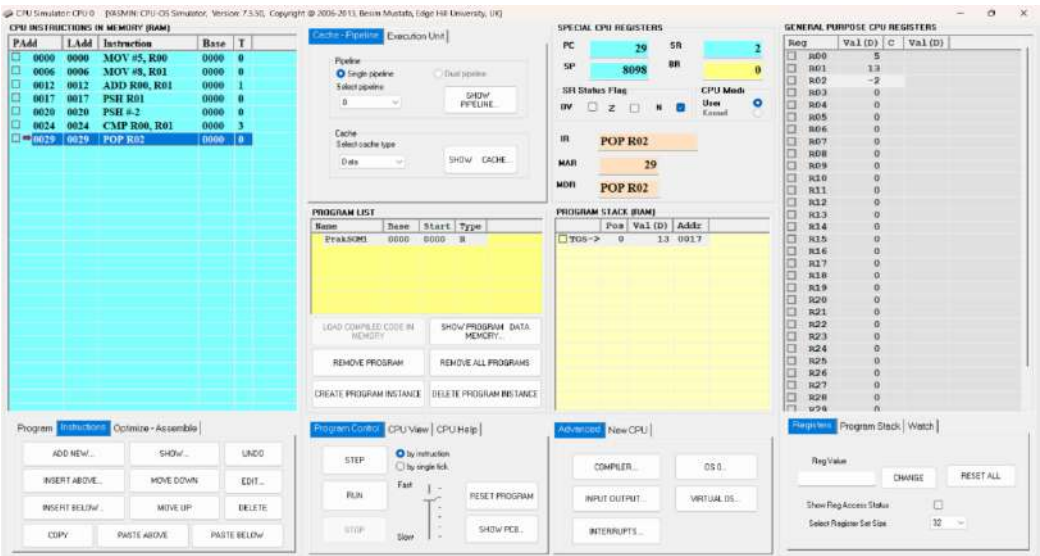
**e. Stack Pointer**

1. ADD NEW lagi lalu ke bagian Data Transfer di Op code Scrool Ke bawah pilih POP dan di Source Operand klik register lalu pilih R02 setelah itu klik NEW INSTRUCTION.

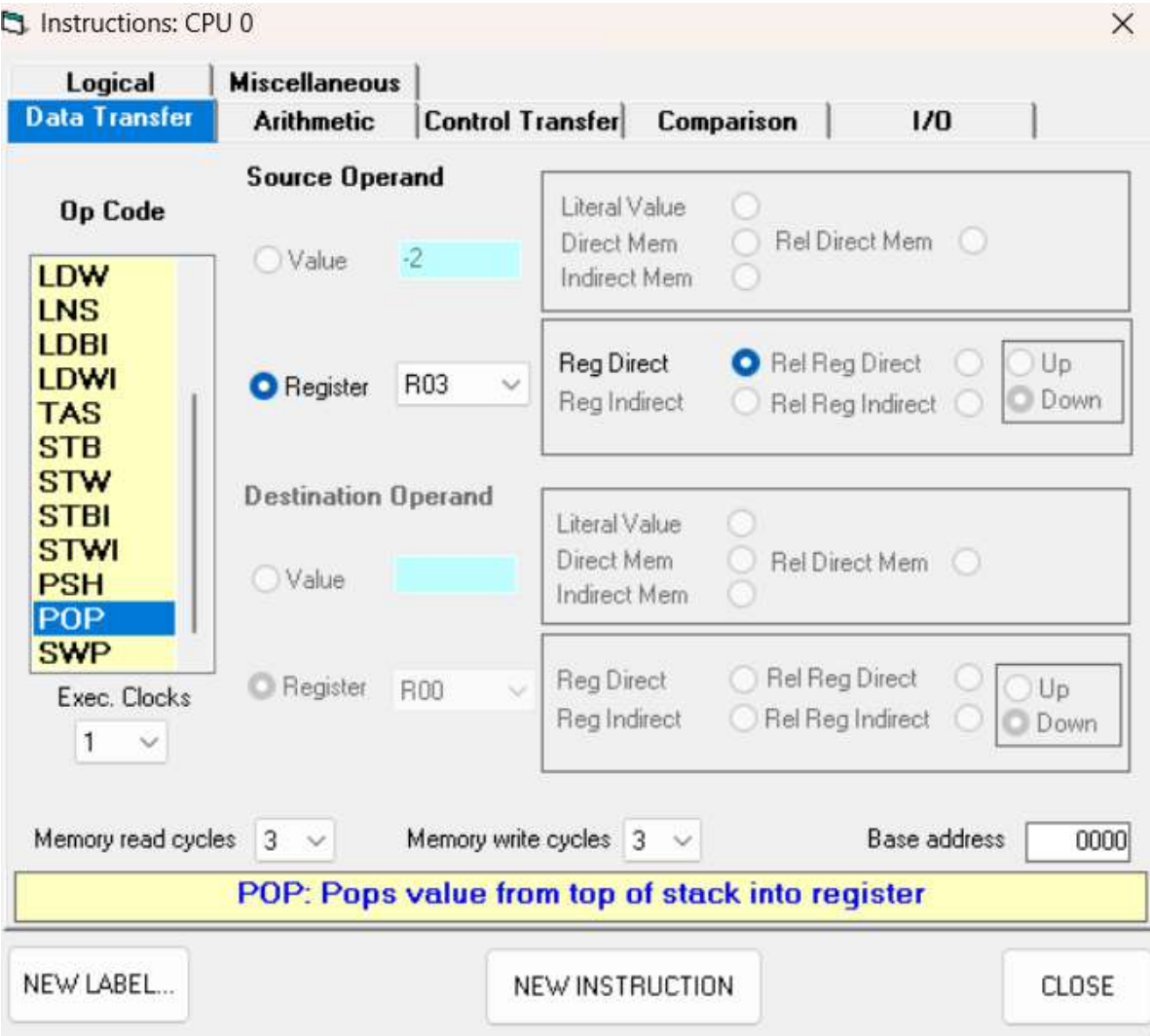


2. Double klik Instruction POP R02 lalu lihat di bagian GENERAL PURPOSE CPU REGISTERS nilai R02 berubah menjadi -2 dan sesuai dengan yang di instruksikan pada bagian teratas di PROGRAM STACK. Lalu nilai register SP berubah menjadi 8098 yang sebelumnya 8100.

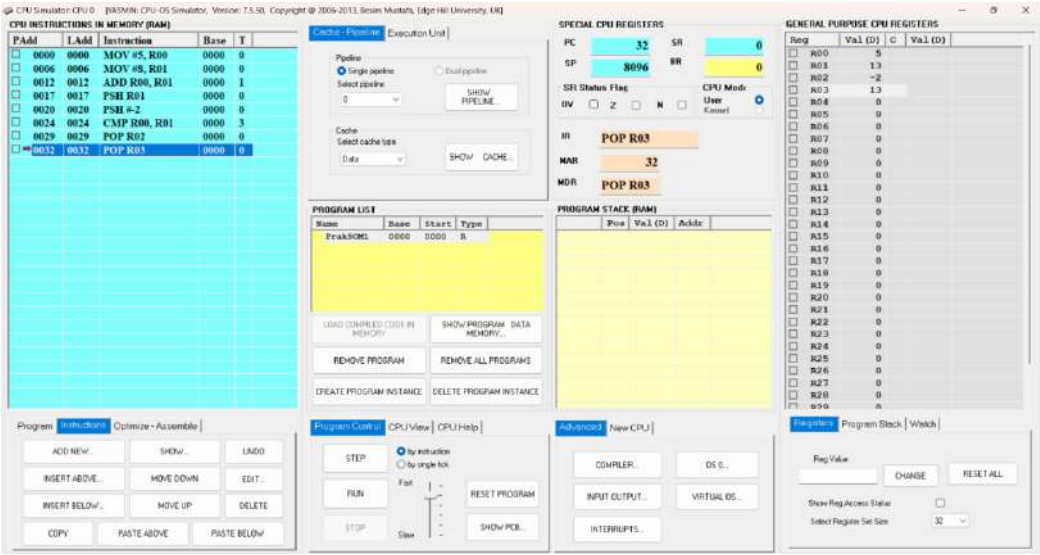




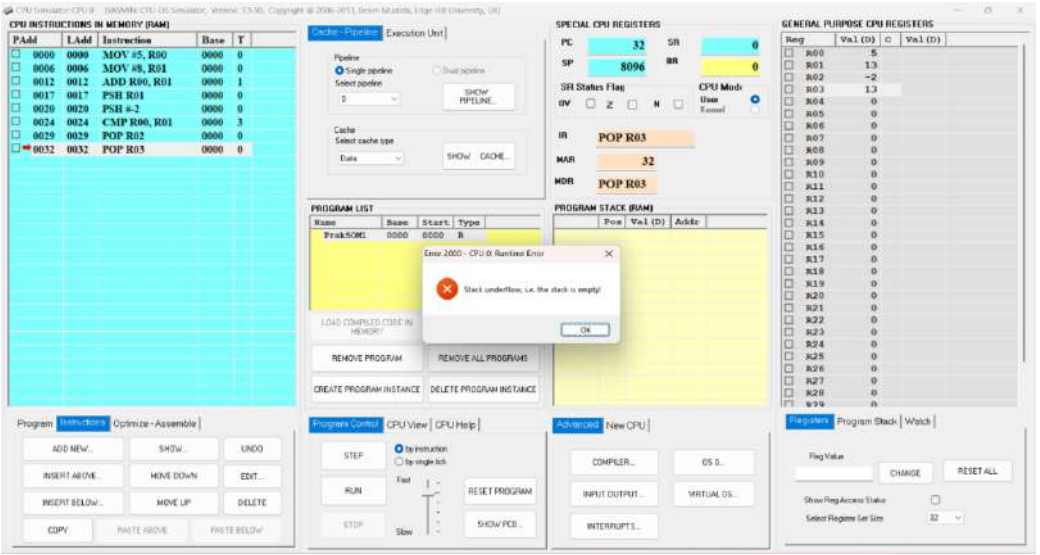
3. ADD NEW lagi pilih Data Transfer lalu di bagian Op code scrool ke bawah pilih POP dan pilih Source Operand klik register pilih R03 dan klik NEW INSTRUCTION.



4. Double Klik Instruction POP R03 lihat ke bagian GENERAL PURPOSE CPU REGISTERS nilai R03 berubah menjadi 13 dan sesuai dengan yang di instruksikan pada bagian teratas di PROGRAM STACK dan sekarang bagian PROGRAM STACK akan kosong.



5. Ketika di eksekusi akan muncul pesan eror seperti gambar di bawah ini kenapa bisa eror karena program mencoba untuk mengambil nilai dari stack ketika stack sudah kosong dan untuk memastikan ada cukup nilai di dalam stack untuk di ambil.



Pre Test

1. Apa itu CPU Simulator?  
Jawab :  
CPU Simulator adalah aplikasi yang memungkinkan pengguna untuk merancang dan mensimulasikan CPU komputer, serta menjalankan program bahasa mesin atau bahasa

rakitan. CPU Simulator dapat digunakan untuk mensimulasikan berbagai arsitektur CPU, seperti arsitektur berbasis akumulator, mirip RISC, atau berbasis tumpukan.

2. Sebutkan dan jelaskan dua intruksi yang dapat dieksekusi oleh CPU Simulator!

Jawab :

- **Instruksi LOAD** digunakan untuk mengambil data dari memori dan memasukkannya ke dalam register CPU. Instruksi ini berfungsi untuk memindahkan data dari alamat memori tertentu ke register, sehingga data tersebut siap digunakan oleh CPU dalam proses selanjutnya.
- Instruksi ADD digunakan untuk melakukan operasi penjumlahan antara dua nilai, baik yang berada dalam register maupun dengan nilai langsung (immediate). Hasil penjumlahan ini kemudian disimpan kembali ke dalam sebuah register, yang nantinya bisa digunakan untuk operasi lanjutan.

3. Apa fungsi dari Register PC, SR dan SP dalam CPU? Jelaskan masing masing!

Jawab :

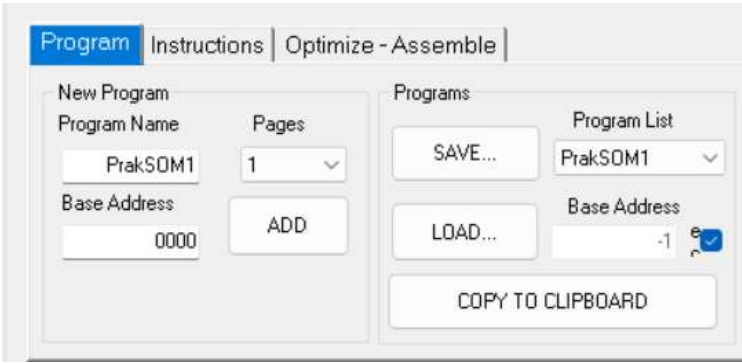
- **Register Program Counter (PC)** berfungsi untuk menyimpan alamat instruksi berikutnya yang akan dieksekusi oleh CPU. Setiap kali CPU mengeksekusi sebuah instruksi, nilai dalam PC akan ditingkatkan untuk menunjuk ke instruksi berikutnya. Ini memungkinkan CPU untuk melanjutkan proses eksekusi instruksi secara berurutan dalam program.
- **Register Status Register (SR)** menyimpan informasi tentang status atau kondisi terakhir operasi yang dilakukan oleh CPU. Register ini berisi flag atau tanda yang menunjukkan hasil operasi, seperti tanda overflow, carry, zero (hasil nol), dan negatif. Flag-flag ini sangat penting dalam pengambilan keputusan, misalnya dalam operasi percabangan atau dalam menentukan hasil dari operasi logika dan aritmatika.
- **Register Stack Pointer (SP)** digunakan untuk menunjuk ke alamat memori di mana data sementara disimpan dalam struktur yang disebut stack. Stack berfungsi untuk menyimpan alamat pengembalian, variabel lokal, atau data sementara lainnya selama eksekusi program, khususnya dalam pemanggilan fungsi atau subrutin. Stack pointer secara otomatis diperbarui saat data didorong (push) ke atau diambil (pop) dari stack.

## Post Test

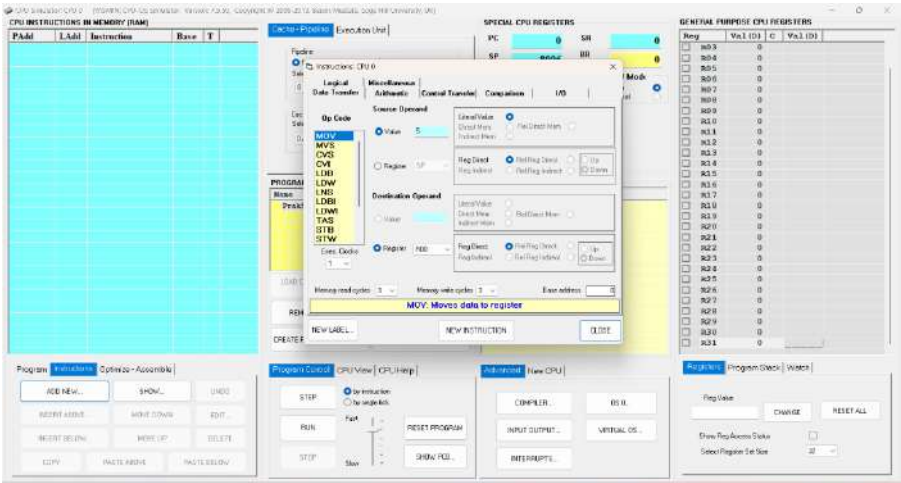
1. Jelaskan langkah langkah untuk memindahkan data ke register menggunakan CPU Simulator!

Jawab :

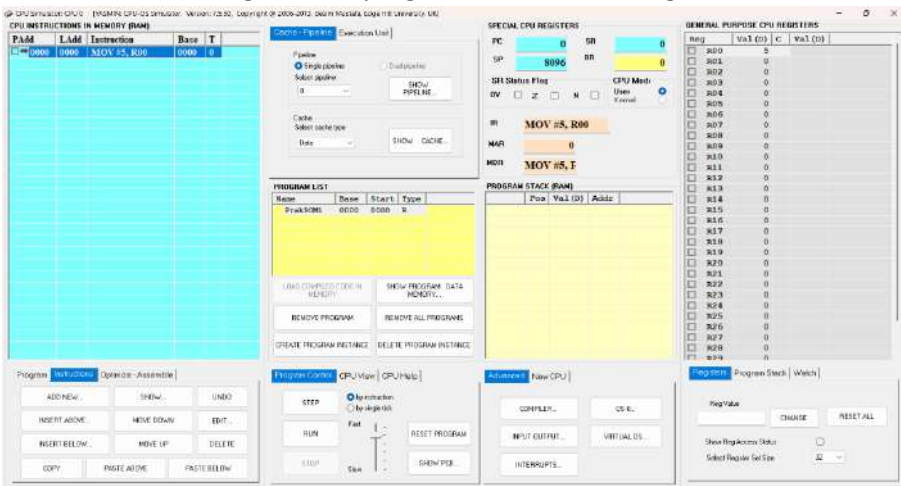
- a) Langkah Pertama, Pada bagian bawah kiri di bagian program isi kan Program Name Dan Base Address sesuai keinginan setelah di isi lalu klik ADD lalu nanti di PROGRAM LIST ada nama folder yang telah kalian isi. Lihat gambar di bawah ini:



- b) Langkah Kedua, Lalu ke bagian kiri bawah klik INTRUCTIONS lalu klik ADD NEW setelah itu di bagian DATA TRANSFER OP CODE pilih MOV di bagian SOURCE OPERAND isi Value setelah di isi, di DESTINATION OPERAND di bagian Register pilih mana yang akan kalian tuju (Value dan register di isi sesuai selera) lalu klik NEW INSTRUCTION. lihat gambar di bawah ini :



- c) Langkah Ketiga, Setelah di New Instruction lalu di bagian CPU INSTRUCTION IN MEMORY (RAM) double klik lalu lihat di GENERAL PURPOSE CPU REGISTERS akan memiliki Val(D) dengan value yang tadi telah di isi. lihat gambar di bawah ini :



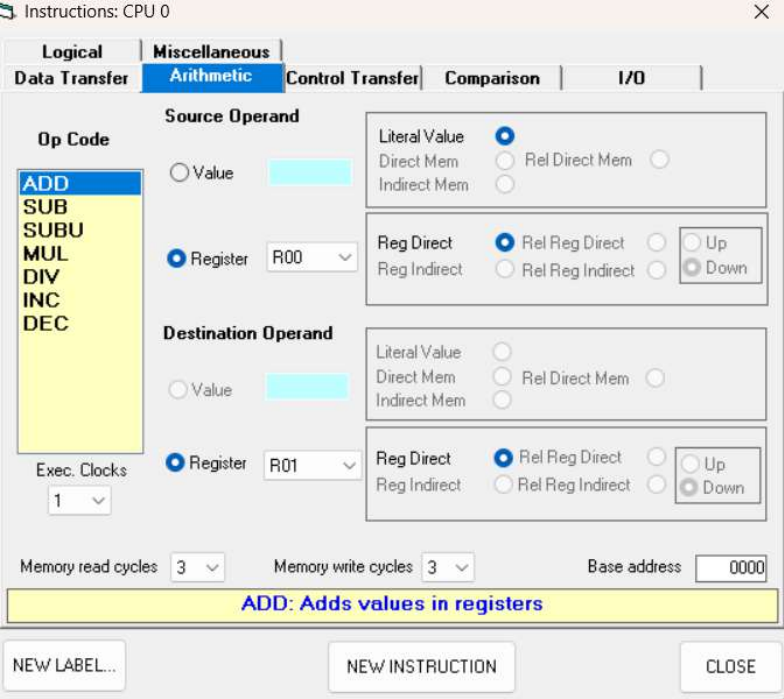
2. Bagaimana cara menambahkan nilai dalam register menggunakan CPU Simulator!

Jawab :

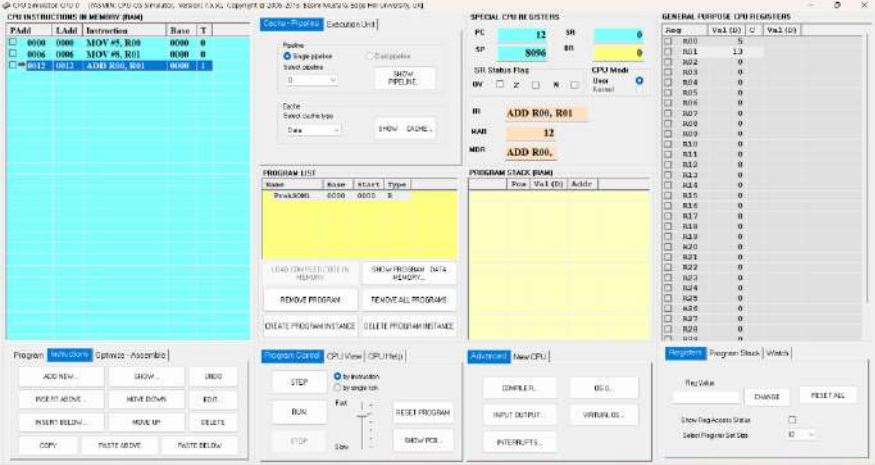
- a) Langkah Pertama, ADD NEW lalu pindah ke bagian Arithmetic setelah itu di op code pilih ADD dan di source operand pilih register dan disini saya isi dengan R00 dan di



destination operand pilih register dan isi dengan R01 setelah di isi klik new instruction.



- b) Langkah Kedua, Lalu di CPU INSTRUCTIONS IN MEMORY (RAM) double klik lihat lagi di bagian GENERAL PURPOSE CPU REGISTERS ada perubahan di R01 yang sebelumnya nilai dari R01 adalah 8 akan berubah menjadi 13, hal itu di karenakan angka 8 pada R01 akan di jumlahkan dengan angka 5 yang ada pada di R00 dan hasil penjumlahan tersebut akan di simpan di R01.



3. Jelaskan perbedaan antara register dan stack dalam konteks CPU!

Jawab :

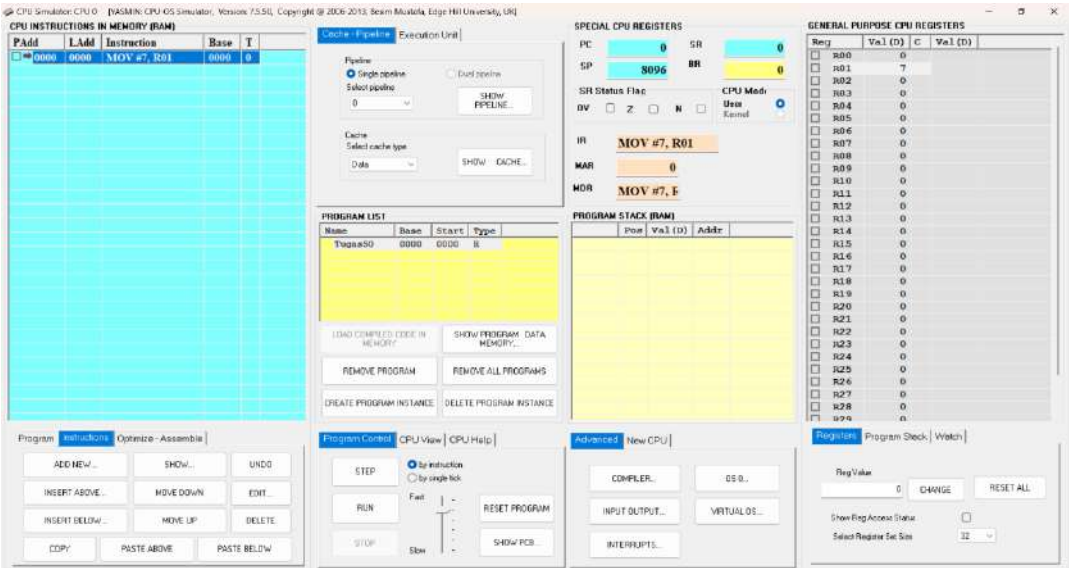
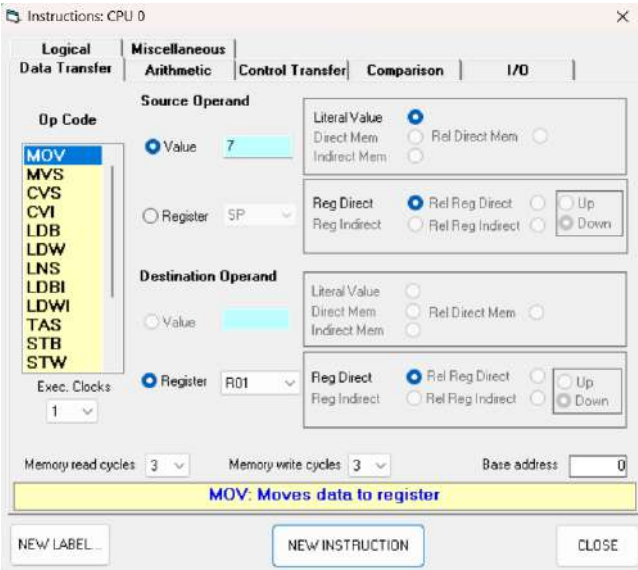
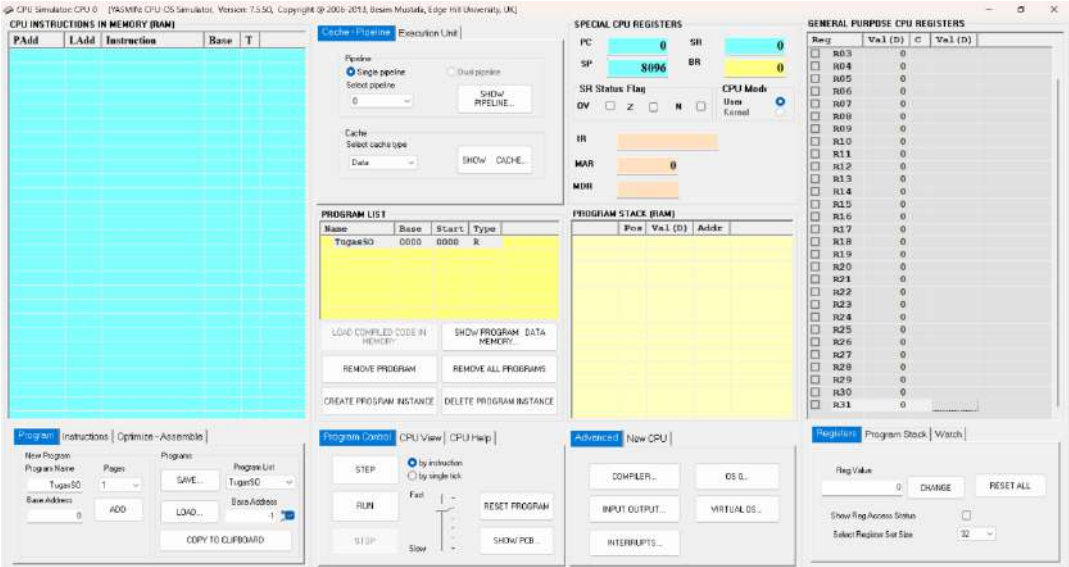
- Register adalah unit penyimpanan memori kecil di dalam CPU yang berfungsi untuk menyimpan data yang sedang dikerjakan CPU. Register memungkinkan CPU untuk menjalankan tugas dengan cepat dan efisien. Register digunakan untuk berbagai keperluan, seperti menyimpan instruksi program sebelum dijalankan atau menyimpan hasil sementara dari perhitungan.
- Stack atau Tumpukan adalah suatu struktur data yang terbentuk dari barisan hingga yang terurut dari satuan data. Pada Stack, penambahan dan penghapusan elemennya hanya dapat dilakukan pada satu posisi, yaitu posisi akhir stack.

Tugas

1. Buatlah program CPU sesuai tahap tahap yang telah dijelaskan. Jalankan serta buat kesimpulan!

Jawab :

- Buat Folder baru Pindahkan angka 7 ke R01



- Menambahkan Aritmetic ADD dengan Value 4 ke Register R01

Instructions: CPU 0

Logical

Miscellaneous

Data Transfer

Arithmetic

Control Transfer

Comparison

I/O

Op Code

ADD

SUB

SUBU

MUL

DIV

INC

DEC

Source Operand

Literal Value

Direct Mem

Indirect Mem

Value

4

Register

R00

Destination Operand

Literal Value

Direct Mem

Indirect Mem

Value

Register

R01

Exec. Clocks

1

Memory read cycles

3

Memory write cycles

3

Base address

0000

ADD: Adds values in registers

NEW LABEL...

NEW INSTRUCTION

CLOSE

CPU Simulator: CPU 0 (NASMITE CPU-02 Simulator, Version: 7.3.30, Copyright © 2005-2013, Essam Mustafa, Edge Hill University, UK)

CPU INSTRUCTIONS IN MEMORY (RAM)

PAddr	LAddr	Instruction	Base	T
0000	0000	MOV #7, R01	0000	0
0006	0006	ADD #4, R01	0000	1

Cache - Expanded

Execution Unit

OpCode

Single opcode

Dual opcode

Select opcode

0

Cache

Select cache type

Data

SHOW PIPELINE...

SHOW CACHE...

PROGRAM LIST

Name	Base	Start	Type
Program0	0000	0000	R

LOAD COMPILED CODE IN MEMORY

SHOW PROGRAM DATA MEMORY...

REMOVE PROGRAM

REMOVE ALL PROGRAMS

CREATE PROGRAM INSTANCE

DELETE PROGRAM INSTANCE

Program Control

CPU View

CPU Help

STEP

Full

Slow

STOP

RESET PROGRAM

SHOW PCB...

SPECIAL CPU REGISTERS

PC

6

SR

0

SP

8096

BR

0

SR Status Flag

OV

Z

N

CFU Mode

User

Kernel

IR

ADD #4, R01

MAR

6

MDR

ADD #4, R

PROGRAM STACK (RAM)

Pos	Val (D)	Addr
-----	---------	------

GENERAL PURPOSE CPU REGISTERS

Reg	Val (D)	C	Val (D)
R00	0		
R01	11		
R02	0		
R03	0		
R04	0		
R05	0		
R06	0		
R07	0		
R08	0		
R09	0		
R10	0		
R11	0		
R12	0		
R13	0		
R14	0		
R15	0		
R16	0		
R17	0		
R18	0		
R19	0		
R20	0		
R21	0		
R22	0		
R23	0		
R24	0		
R25	0		
R26	0		
R27	0		
R28	0		
R29	0		

Registers

Program Stack

Watch

Reg Value

0

CHANGE

RESET ALL

Show Reg Access Status

Select Register Set Size

32

- Menambahkan Op Code Data Transfer PSH dengan Value 7 Untuk di simpan pada stack teratas.

Instructions: CPU 0

Logical

Miscellaneous

Data Transfer

Arithmetic

Control Transfer

Comparison

I/O

Op Code

LDW

LNS

LDBI

LDWI

TAS

STB

STW

STBI

STWI

PSH

POP

SWP

Source Operand

Literal Value

Direct Mem

Indirect Mem

Value

7

Register

SP

Destination Operand

Literal Value

Direct Mem

Indirect Mem

Value

Register

R00

Exec. Clocks

1

Memory read cycles

3

Memory write cycles

3

Base address

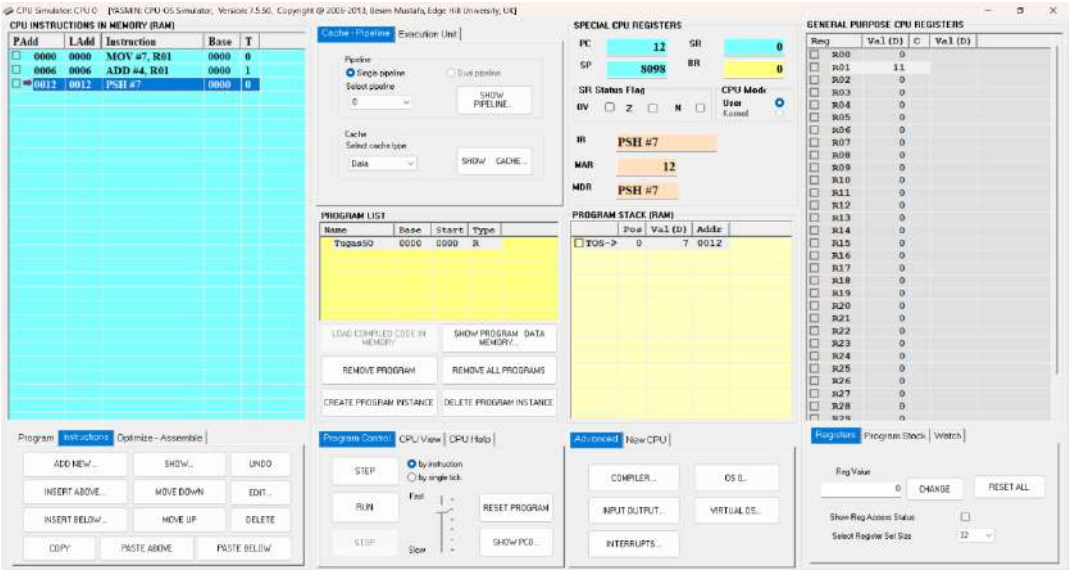
0000

PSH: Pushes value or value in register onto top of stack

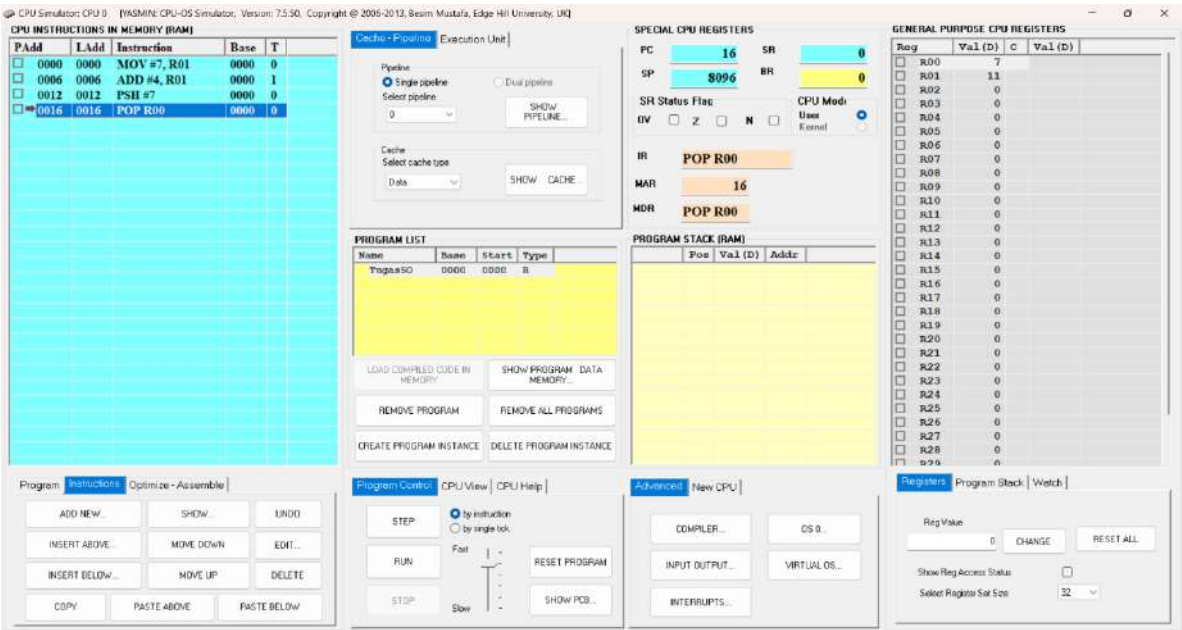
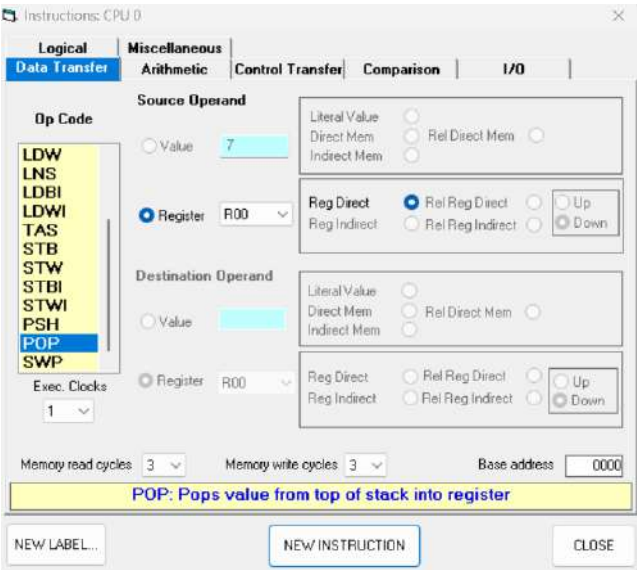
NEW LABEL...

NEW INSTRUCTION

CLOSE

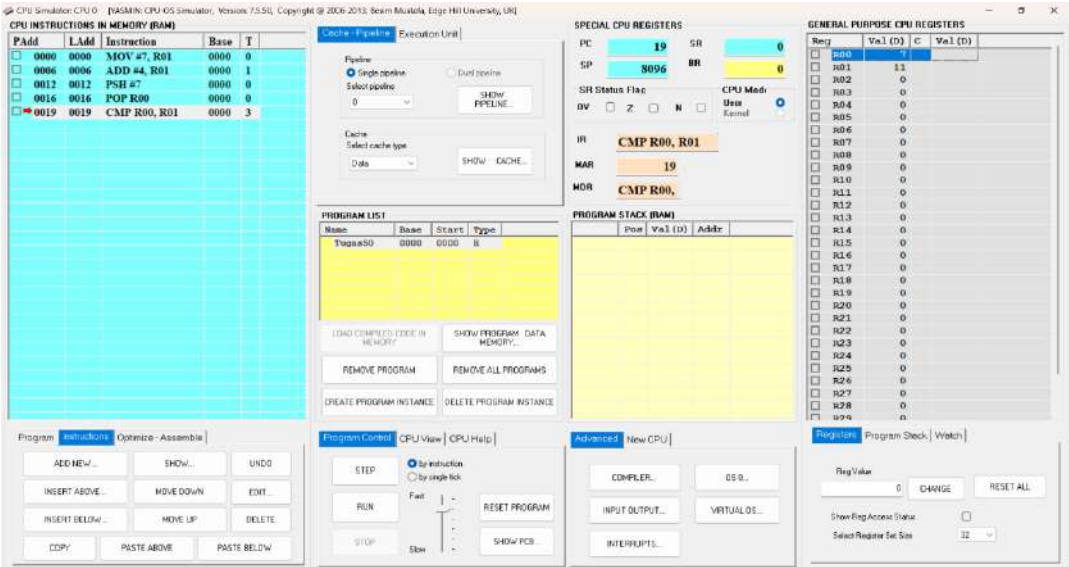
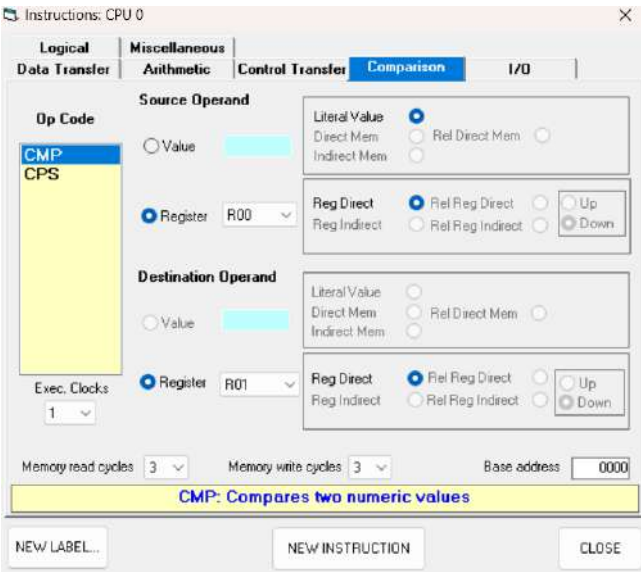


- Mengambil Op Code Data Transfer POP dengan Value 7 dengan Register R00 Untuk di simpan pada stack teratas.





- Membandingkan CMP Register R00 dan R01



Kesimpulan

Kesimpulan yang saya ambil, secara keseluruhan cpu simulator sangat efektif dalam mempelajari dasar- dasar CPU dan juga memahami bagaimana data di simpan dan di akses. tapi mungkin saat saya pertama kali mencoba di bagian Pembandingan di situ saya kebingungan karena nilai value dan register tidak tahu mau di bagaimana kan lagi.dan saya pun terus mencoba saat telah menemukan Dimana yang salah nya ternyata Cuma salah di bagian penempatan nya saja.selain itu cpu simulator juga lebih aman dan tanpa merusak perangkat keras yang asli.