Table

Description automatically generated





**import** numpy **as** np

**global** reg,memory

reg **=** np.zeros((16), dtype **=** np.uint8)

memory **=** np.zeros((16), dtype **=** np.uint8)

rows **=** 45

columns **=** 2

values **=** rows **\*** columns

matrix **=** np.zeros((rows,columns), dtype **=** np.uint8)

**def** run\_program(num\_bytes):

  pc **=** 0

  sb **=** np.uint8

  fb **=** np.uint8

  m **=** np.uint8

  counter **=** np.uint8

  counter **=** (num\_bytes**/**2)   *#8*

**while**( pc **<** counter ):

      pc **=** pc **+** 1

**if** pc **==** counter:

**break**

      fb **=** matrix[pc,0]

      sb **=** matrix[pc,1]

      m **=** fb **>>** 4      *# shift*

**if** m **==** 0:                    *# MOV Rn, direct*

        reg[fb **&** 0x0f] **=** memory[sb]

**print**("bu reg sıfır...",reg[0])

**elif** m **==** 1:                 *# MOV direct, Rn*

        memory[sb] **=** reg[fb **&** 0x0f]

**elif** m **==** 2:                  *#  MOV @Rn, Rm*

        memory[reg[fb **&** 0x0f]]

**elif** m **==** 3:                  *# MOV Rn, #immed.*

        reg[fb **&** 0x0f] **=** sb

**elif** m **==** 4:                 *# ADD Rn, Rm*

        reg[fb **&** 0x0f] **+=** reg[sb **>>** 4]

**elif** m **==** 5:                *# SUB Rn, Rm*

        reg[fb **&** 0x0f] **-=** reg[sb **>>** 4]

**elif** m **==** 6:               *#  JZ Rn, relative*

        pc **=** pc **+** sb

**else**:

**print**('wrong choice...')

**return**

**def**  print\_memory\_contents():

**print**("Printing memory")

**for** i **in** range(0,16,1):

**print**("Memory Address...",i,"Value...",memory[i])

**def** print\_register\_contents():

**print**("Printing registers")

**for** i **in** range(0,16,1):

**print**("R [%d] : %d"**%**(i,reg[i]))

**with** open('tests.bin','rb') **as** file:

*# read HEX value*

**for** x **in** range(0,rows,1):

**for** y **in** range(0,columns,1):

       byte **=** file.read(2)

       result\_int **=** int(byte)

       matrix[x,y] **=** result\_int

file.close()

**print**(matrix)

run\_program(values)

print\_memory\_contents()

print\_register\_contents()

**Binary file content as below: (tests.bin)**

480049105001510097006416813299009700641681329900970064168132990097006416813299009700641681329900970064168132990097006416813299009700641681329900970064168132990097006416813299009700