॥वसुधैव कुटुम्बकम्॥

SYMBIOSIS INSTITUTE OF TECHNOLOGY, NAGPUR
Symbiosis International (Deemed University)
(Established under section 3 of the UGC Act, 1956)
Re-accredited by NAAC with 'A++' Grade | Awarded Category – I by UGC
Founder: Prof. Dr. S. B. Mujumdar, M. Sc., Ph. D. (Awarded Padma Bhushan and Padma Shri by President of India)

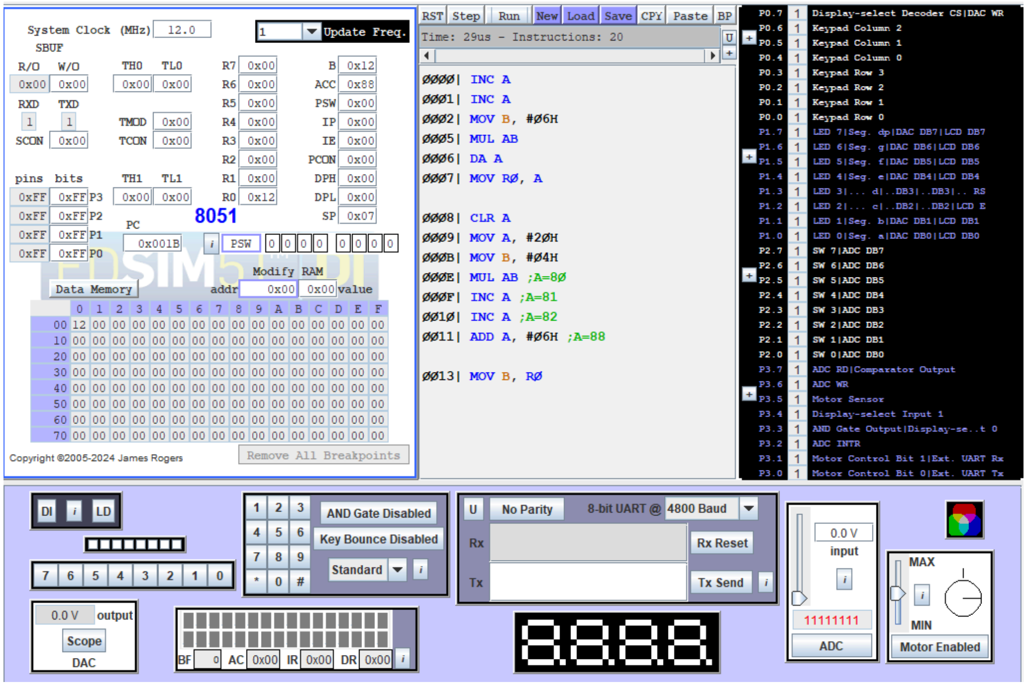
Microcontroller and Embedded Systems

CA-I (Assignment)

Name: Reshal Age
PRN: 24070521288
Section: D

Q1. Write an 8051 Assembly Language Program (ALP) to generate the last four digits of your PRN using any arithmetic instructions.

Sol:



```
Time: 29us - Instructions: 20

ØØØØ| INC A
ØØØ1| INC A
ØØØ2| MOV B, #Ø6H
ØØØ5| MUL AB
ØØØ6| DA A
ØØØ7| MOV RØ, A

ØØØ8| CLR A
ØØØ9| MOV A, #2ØH
ØØØB| MOV B, #Ø4H
ØØØE| MUL AB ;A=8Ø
ØØØF| INC A ;A=81
ØØ1Ø| INC A ;A=82
ØØ11| ADD A, #Ø6H ;A=88

ØØ13| MOV B, RØ
```

```
System Clock (MHz) 12.0
R7 0x00   B    0x12
R6 0x00   ACC  0x88
R5 0x00   PSW  0x00
R4 0x00   IP   0x00
R3 0x00   IE   0x00
R2 0x00   PCON 0x00
R1 0x00   DPH  0x00
R0 0x12   DPL  0x00
          SP   0x07
PC 0x001B
Data Memory 00: 12 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

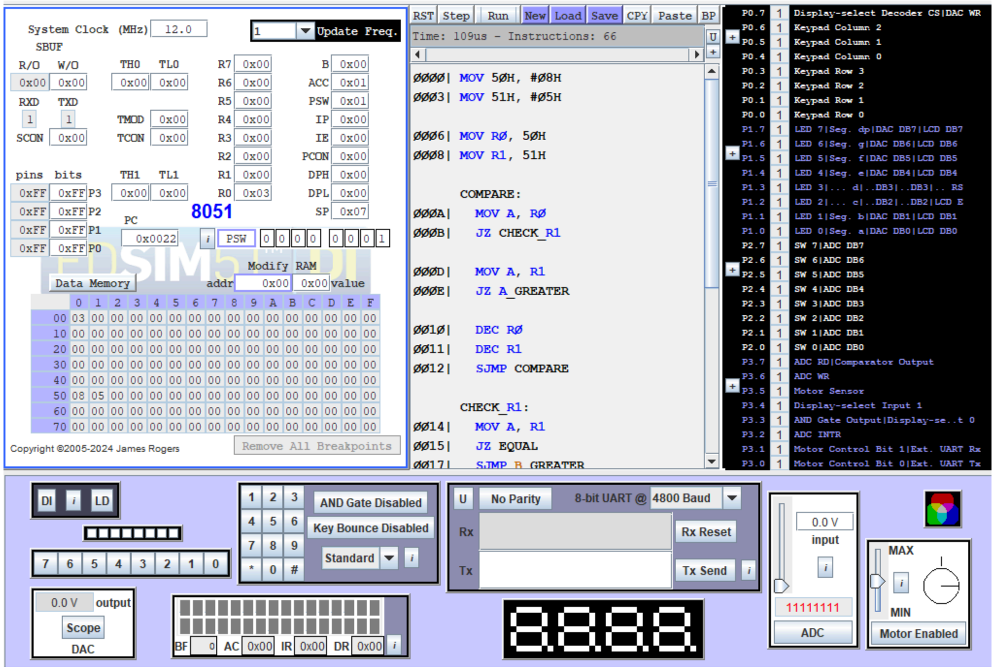
Values are generated using a combination of INC, ADD, and MUL instructions and converted to BCD using DA A.

Reshal Age_24070521288

Q2. Write an 8051 ALP to compare unsigned numbers at 50H and 51H using only basic instructions, storing 01H if greater, 00H if equal, and FFH if smaller.

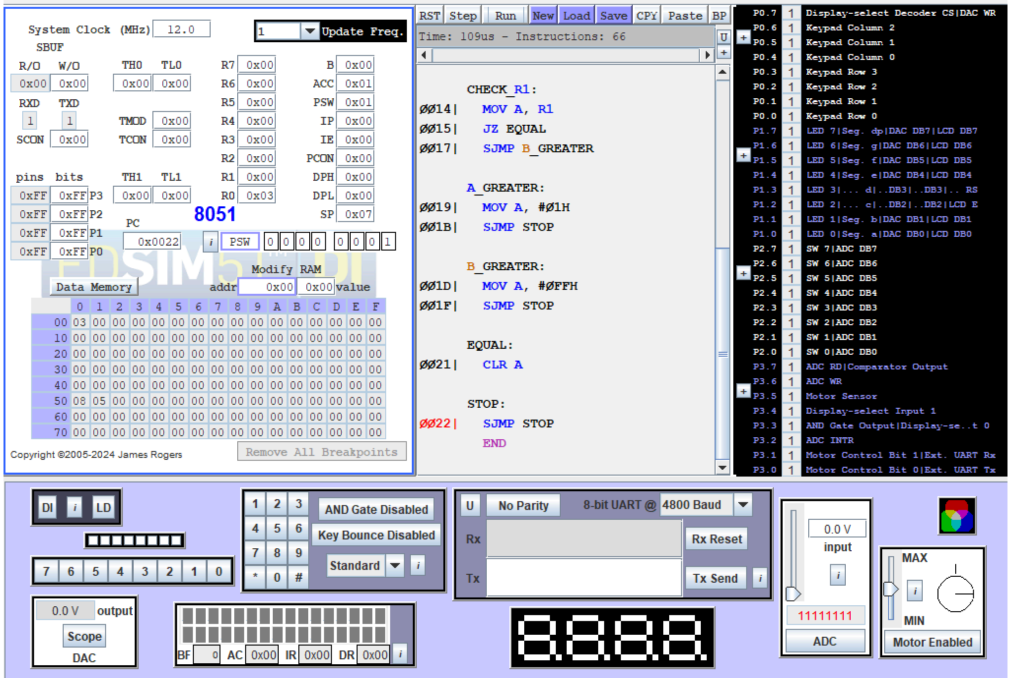
Sol:

Case 1: A > B — Example: 50H = #08H, 51H = #05H → Result stored should be 01H.



```
System Clock (MHz) 12.0        Update Freq. 1
R7 0x00  R6 0x00  R5 0x00  R4 0x00  R3 0x00  R2 0x00  R1 0x00  R0 0x03
B 0x00  ACC 0x01  PSW 0x01  IP 0x00  IE 0x00  PCON 0x00  DPH 0x00  DPL 0x00  SP 0x07
PC 0x0022   PSW 0 0 0 0 0 0 0 1
Data Memory
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50 08 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
Time: 109us - Instructions: 66

ØØØØ| MOV 5ØH, #Ø8H
ØØØ3| MOV 51H, #Ø5H

ØØØ6| MOV RØ, 5ØH
ØØØ8| MOV R1, 51H

      COMPARE:
ØØØA|   MOV A, RØ
ØØØB|   JZ CHECK_R1

ØØØD|   MOV A, R1
ØØØE|   JZ A_GREATER

ØØ1Ø|   DEC RØ
ØØ11|   DEC R1
ØØ12|   SJMP COMPARE

      CHECK_R1:
ØØ14|   MOV A, R1
ØØ15|   JZ EQUAL
ØØ17|   SJMP B_GREATER
```

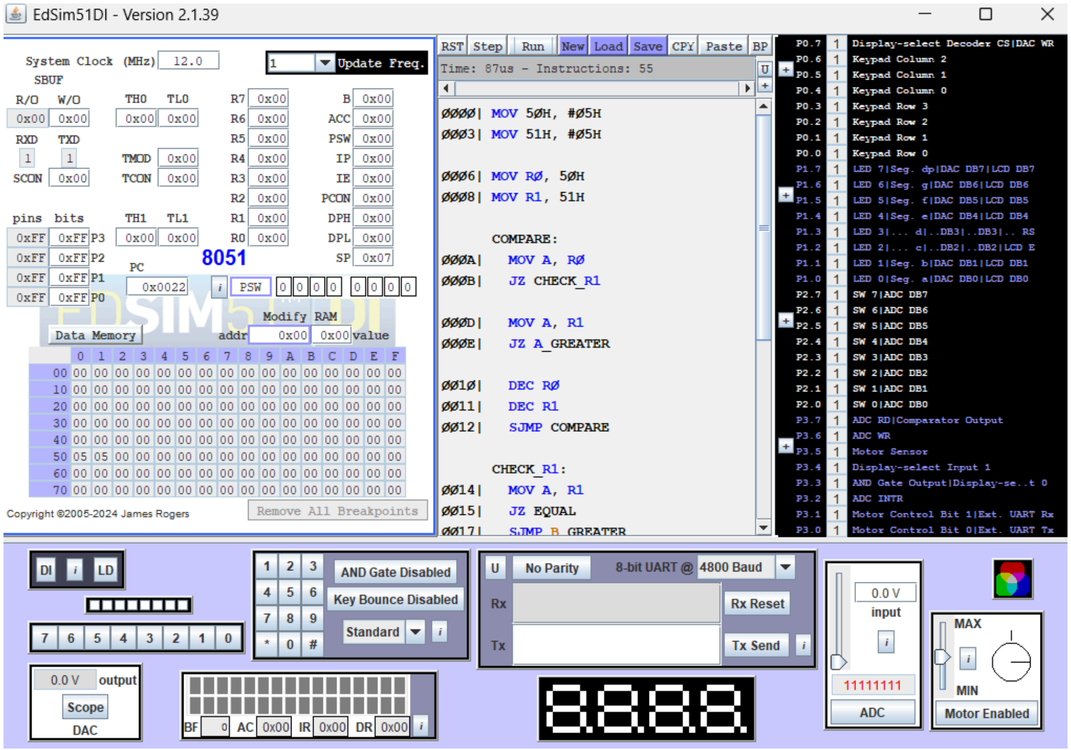


```
System Clock (MHz) 12.0        Update Freq. 1
R7 0x00  R6 0x00  R5 0x00  R4 0x00  R3 0x00  R2 0x00  R1 0x00  R0 0x03
B 0x00  ACC 0x01  PSW 0x01  IP 0x00  IE 0x00  PCON 0x00  DPH 0x00  DPL 0x00  SP 0x07
PC 0x0022   PSW 0 0 0 0 0 0 0 1
Data Memory
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50 08 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
Time: 109us - Instructions: 66

      CHECK_R1:
ØØ14|   MOV A, R1
ØØ15|   JZ EQUAL
ØØ17|   SJMP B_GREATER

      A_GREATER:
ØØ19|   MOV A, #Ø1H
ØØ1B|   SJMP STOP

      B_GREATER:
ØØ1D|   MOV A, #ØFFH
ØØ1F|   SJMP STOP

      EQUAL:
ØØ21|   CLR A

      STOP:
ØØ22|   SJMP STOP
        END
```

Reshal Age_24070521288

Case 2: A = B — Example: 50H = #05H, 51H =#05H → Result stored should be 00H.



EdSim51DI - Version 2.1.39

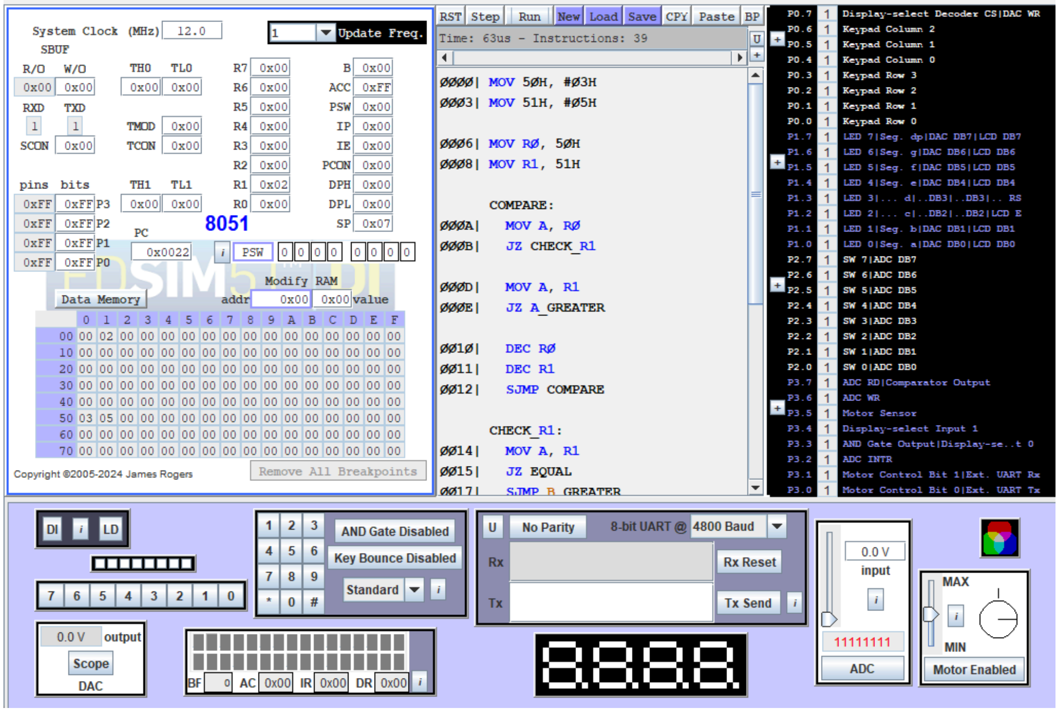
Time: 87us - Instructions: 55

```
0000| MOV 50H, #05H
0003| MOV 51H, #05H

0006| MOV R0, 50H
0008| MOV R1, 51H

      COMPARE:
000A|   MOV A, R0
000B|   JZ CHECK_R1

000D|   MOV A, R1
000E|   JZ A_GREATER

0010|   DEC R0
0011|   DEC R1
0012|   SJMP COMPARE

      CHECK_R1:
0014|   MOV A, R1
0015|   JZ EQUAL
0017|   SJMP B_GREATER
```

```
System Clock (MHz) 12.0
R7 0x00  R6 0x00  R5 0x00  R4 0x00  R3 0x00  R2 0x00  R1 0x00  R0 0x00
B 0x00  ACC 0x00  PSW 0x00  IP 0x00  IE 0x00  PCON 0x00  DPH 0x00  DPL 0x00  SP 0x07
PC 0x0022

Data Memory
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50  05 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Copyright ©2005-2024 James Rogers

Case 3: A < B - Example: 50H = 10H, 51H = 12H → Result stored should be FFH



Time: 63us - Instructions: 39

```
0000| MOV 50H, #03H
0003| MOV 51H, #05H

0006| MOV R0, 50H
0008| MOV R1, 51H

      COMPARE:
000A|   MOV A, R0
000B|   JZ CHECK_R1

000D|   MOV A, R1
000E|   JZ A_GREATER

0010|   DEC R0
0011|   DEC R1
0012|   SJMP COMPARE

      CHECK_R1:
0014|   MOV A, R1
0015|   JZ EQUAL
0017|   SJMP B_GREATER
```

```
System Clock (MHz) 12.0
R7 0x00  R6 0x00  R5 0x00  R4 0x00  R3 0x00  R2 0x00  R1 0x02  R0 0x00
B 0x00  ACC 0xFF  PSW 0x00  IP 0x00  IE 0x00  PCON 0x00  DPH 0x00  DPL 0x00  SP 0x07
PC 0x0022

Data Memory
     0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00  00 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50  03 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

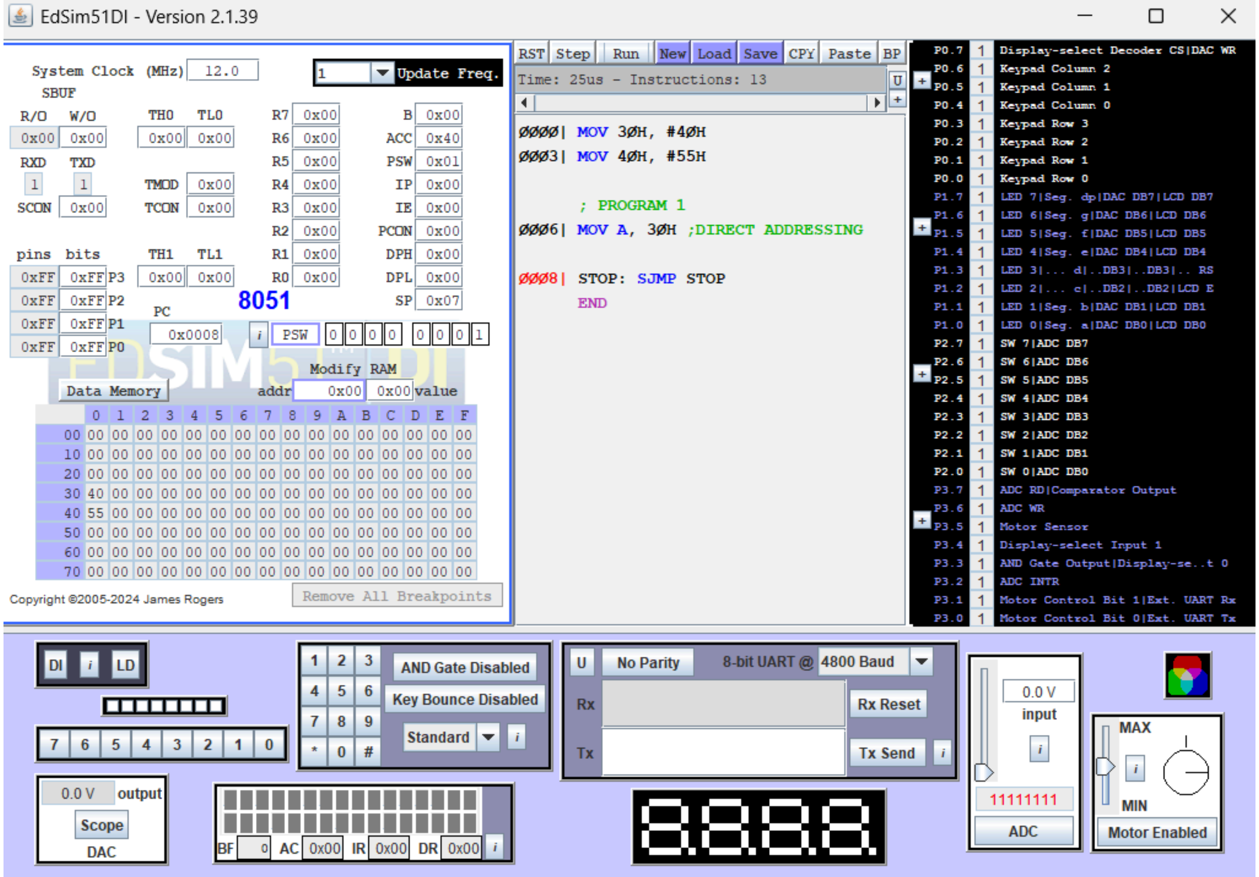
Copyright ©2005-2024 James Rogers

Reshal Age_24070521288

In this program, both numbers are changed step by step. When a value becomes zero, the Zero flag is set. The program checks this flag to decide which number is bigger or if both are equal.

Q3. Write two short 8051 assembly programs using direct and indirect addressing that reference the same RAM location and demonstrate through simulation that the outputs differ due to the addressing mode.

Case 1: Direct addressing



Time: 25us - Instructions: 13

```
ØØØØ| MOV 3ØH, #4ØH
ØØØ3| MOV 4ØH, #55H

      ; PROGRAM 1
ØØØ6| MOV A, 3ØH ;DIRECT ADDRESSING

ØØØ8| STOP: SJMP STOP
      END
```

System Clock (MHz) 12.0

R7	0x00	B	0x00	TH0	0x00
R6	0x00	ACC	0x40	TL0	0x00
R5	0x00	PSW	0x01	TMOD	0x00
R4	0x00	IP	0x00	TCON	0x00
R3	0x00	IE	0x00	TH1	0x00
R2	0x00	PCON	0x00	TL1	0x00
R1	0x00	DPH	0x00	SCON	0x00
R0	0x00	DPL	0x00	PC	0x0008
		SP	0x07		

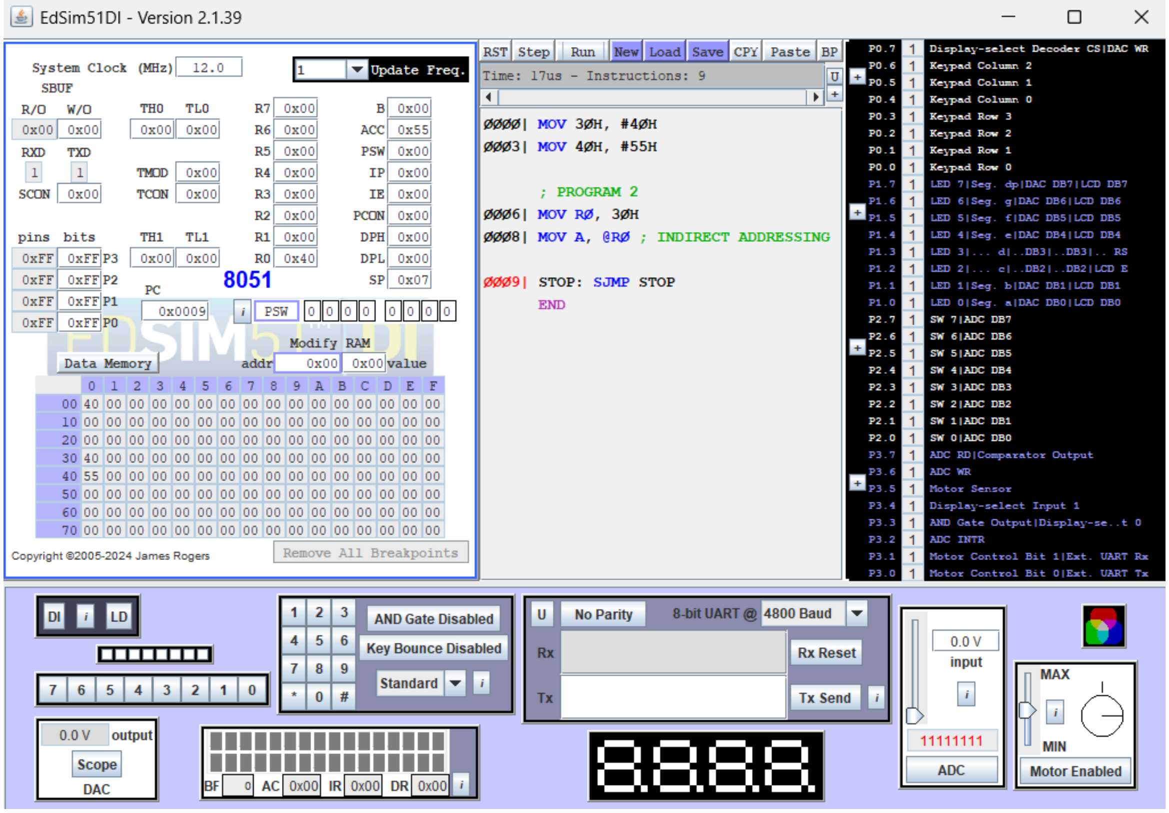
PSW 0 0 0 0 0 0 0 1

Data Memory

```
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40 55 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Reshal Age_24070521288

Case 2: Indirect addressing



EdSim51DI - Version 2.1.39

```
Time: 17us - Instructions: 9

ØØØØ| MOV 3ØH, #4ØH
ØØØ3| MOV 4ØH, #55H

      ; PROGRAM 2
ØØØ6| MOV RØ, 3ØH
ØØØ8| MOV A, @RØ ; INDIRECT ADDRESSING

ØØØ9| STOP: SJMP STOP
      END
```

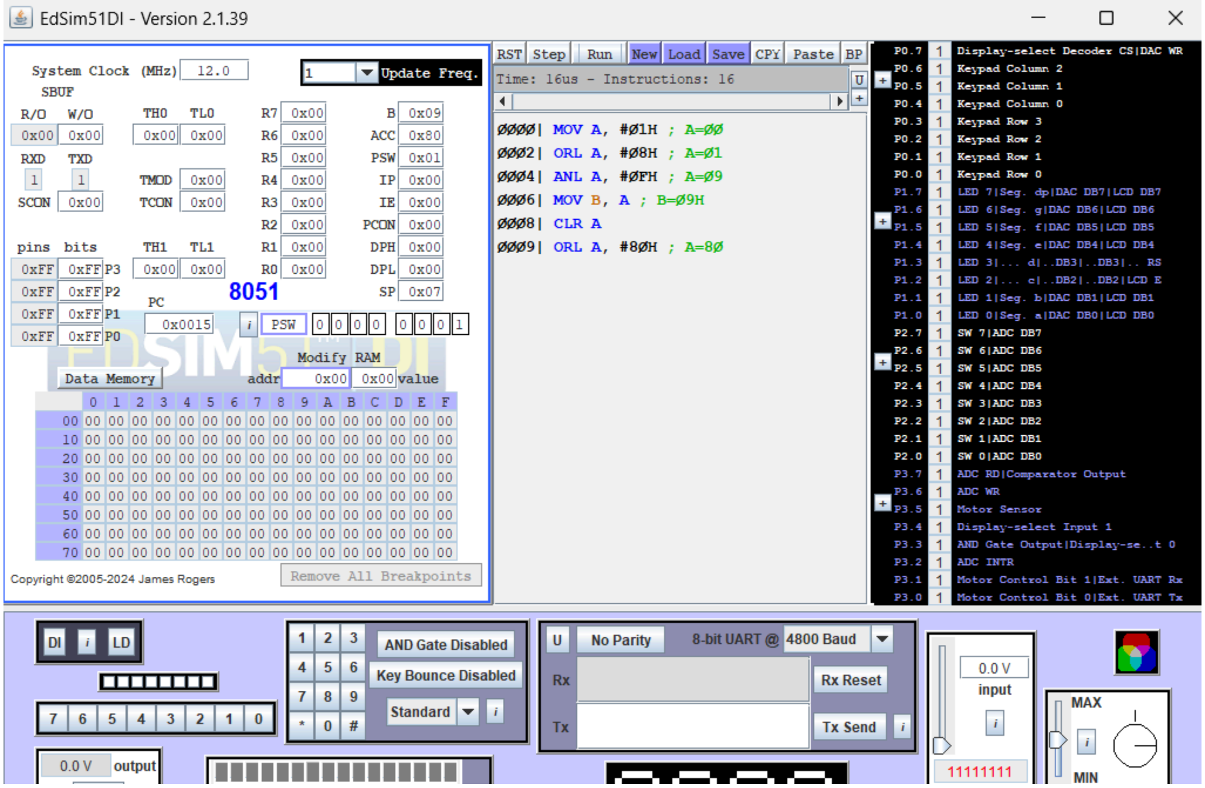
```
System Clock (MHz) 12.0
R7 0x00   B    0x00
R6 0x00   ACC  0x55
R5 0x00   PSW  0x00
R4 0x00   IP   0x00
R3 0x00   IE   0x00
R2 0x00   PCON 0x00
R1 0x00   DPH  0x00
R0 0x40   DPL  0x00
          SP   0x07
PC 0x0009
```

```
Data Memory
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40 55 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Even though both programs use the same RAM address, they give different outputs because of different addressing modes. In direct addressing, the data stored at the given address is accessed directly. In indirect addressing, the value stored at that address is treated as a pointer to another memory location. Therefore, different data is fetched, which results in different outputs.

Reshal Age_24070521288

Q4. Write an 8051 ALP to generate the last four digits of your mobile number using logical instructions (ANL, ORL, CLR) without directly loading the number and store the result in the Accumulator.



System Clock (MHz) 12.0

B 0x09 ACC 0x80 PSW 0x01 IP 0x00 IE 0x00 PCON 0x00 DPH 0x00 DPL 0x00 SP 0x07

R7 0x00 R6 0x00 R5 0x00 R4 0x00 R3 0x00 R2 0x00 R1 0x00 R0 0x00

PC 0x0015

Time: 16us - Instructions: 16

```
0000| MOV A, #01H ; A=00
0002| ORL A, #08H ; A=01
0004| ANL A, #0FH ; A=09
0006| MOV B, A ; B=09H
0008| CLR A
0009| ORL A, #80H ; A=80
```

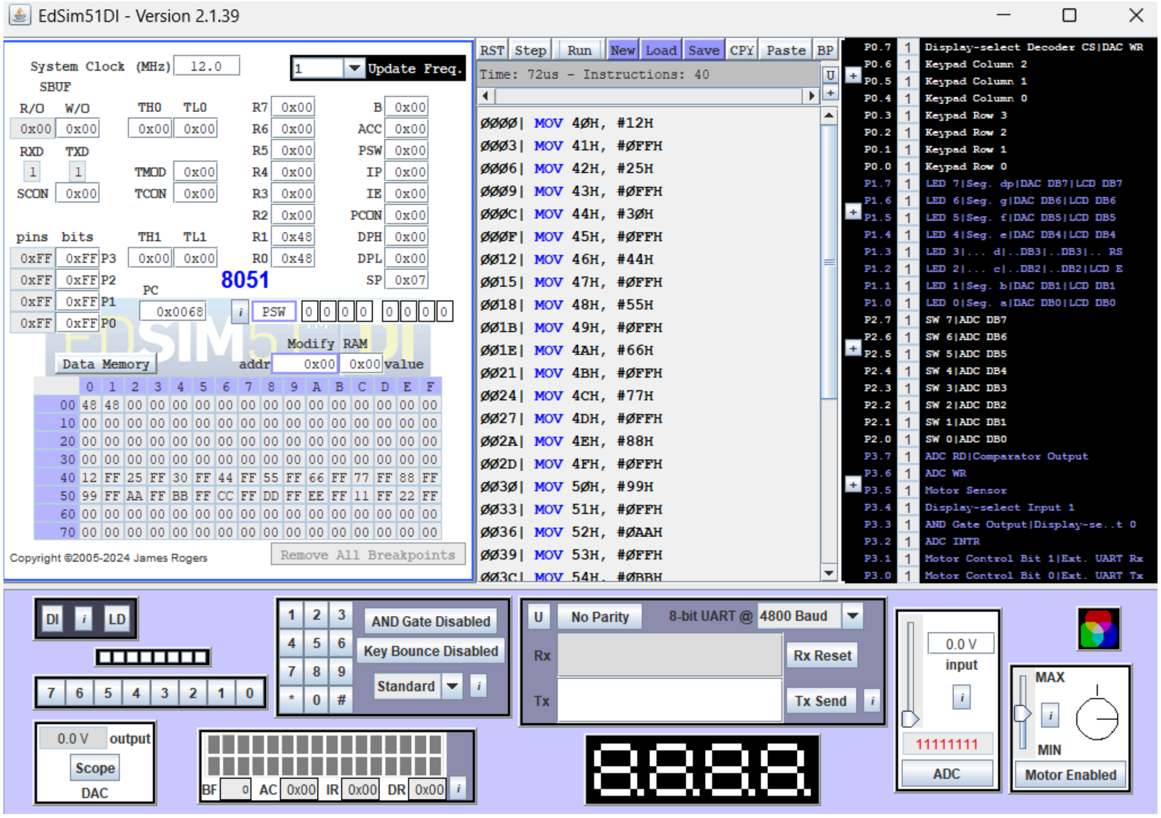
Copyright ©2005-2024 James Rogers

Reshal Age_24070521288

Q5. Write an 8051 ALP using indirect addressing to scan RAM locations 40H–5FH, remove all FFH values, shift valid data left without using extra memory or stack, fill remaining locations with 00H, and verify the result through simulation.

Sol:

Before Execution: RAM locations 40H–5FH contain event codes including invalid values (FFH) that need to be removed.



EdSim51DI - Version 2.1.39

System Clock (MHz) 12.0

Time: 72us - Instructions: 40

R7	0x00	B	0x00	TH0	0x00
R6	0x00	ACC	0x00	TL0	0x00
R5	0x00	PSW	0x00	TMOD	0x00
R4	0x00	IP	0x00	TCON	0x00
R3	0x00	IE	0x00	TH1	0x00
R2	0x00	PCON	0x00	TL1	0x00
R1	0x48	DPH	0x00	SCON	0x00
R0	0x48	DPL	0x00	PC	0x0068
		SP	0x07		

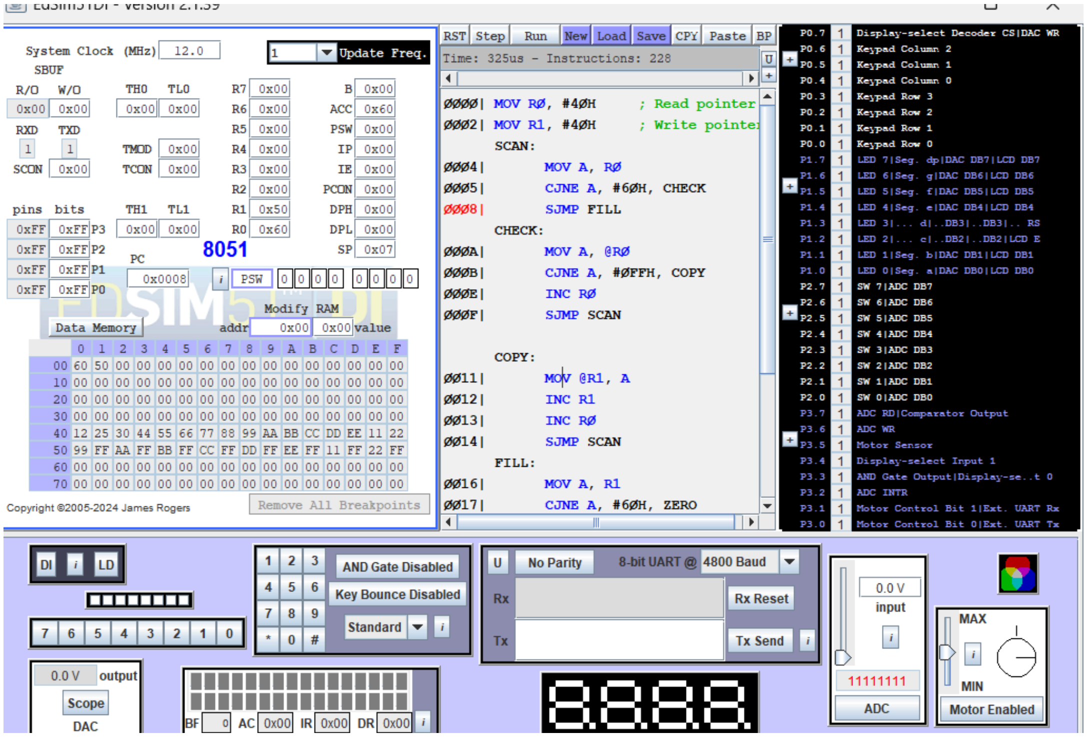
```
ØØØØ| MOV 4ØH, #12H
ØØØ3| MOV 41H, #ØFFH
ØØØ6| MOV 42H, #25H
ØØØ9| MOV 43H, #ØFFH
ØØØC| MOV 44H, #3ØH
ØØØF| MOV 45H, #ØFFH
ØØ12| MOV 46H, #44H
ØØ15| MOV 47H, #ØFFH
ØØ18| MOV 48H, #55H
ØØ1B| MOV 49H, #ØFFH
ØØ1E| MOV 4AH, #66H
ØØ21| MOV 4BH, #ØFFH
ØØ24| MOV 4CH, #77H
ØØ27| MOV 4DH, #ØFFH
ØØ2A| MOV 4EH, #88H
ØØ2D| MOV 4FH, #ØFFH
ØØ3Ø| MOV 5ØH, #99H
ØØ33| MOV 51H, #ØFFH
ØØ36| MOV 52H, #ØAAH
ØØ39| MOV 53H, #ØFFH
ØØ3C| MOV 54H, #ØBBH
```

Data Memory

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	48	48	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
20	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
30	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
40	12	FF	25	FF	30	FF	44	FF	55	FF	66	FF	77	FF	88	FF
50	99	FF	AA	FF	BB	FF	CC	FF	DD	FF	EE	FF	11	FF	22	FF
60	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
70	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Copyright ©2005-2024 James Rogers

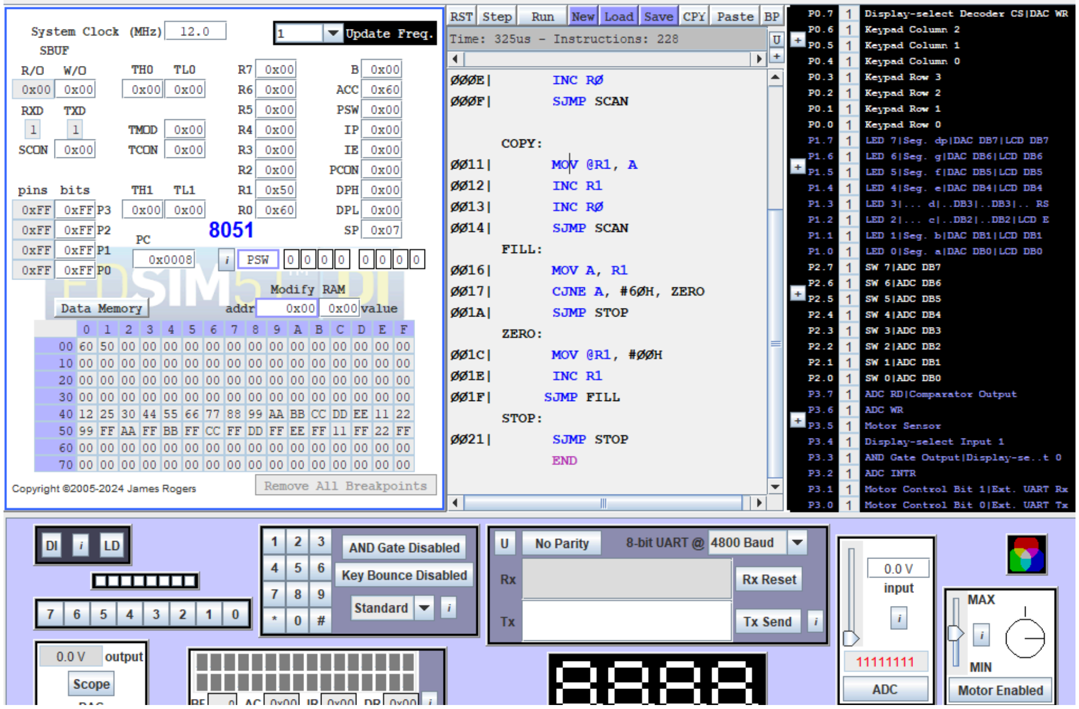
After Execution: All FFH values are removed, valid data is shifted left, and remaining memory locations are filled with 00H showing successful in-place compaction.



```
Time: 325us - Instructions: 228

0000| MOV R0, #40H      ; Read pointer
0002| MOV R1, #40H      ; Write pointer
      SCAN:
0004|         MOV A, R0
0005|         CJNE A, #60H, CHECK
0008|         SJMP FILL
      CHECK:
000A|         MOV A, @R0
000B|         CJNE A, #0FFH, COPY
000E|         INC R0
000F|         SJMP SCAN

      COPY:
0011|         MOV @R1, A
0012|         INC R1
0013|         INC R0
0014|         SJMP SCAN
      FILL:
0016|         MOV A, R1
0017|         CJNE A, #60H, ZERO
```

```
R7 0x00   B    0x00
R6 0x00   ACC  0x60
R5 0x00   PSW  0x00
R4 0x00   IP   0x00
R3 0x00   IE   0x00
R2 0x00   PCON 0x00
R1 0x50   DPH  0x00
R0 0x60   DPL  0x00
          SP   0x07
PC 0x0008
```

```
Data Memory
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 60 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40 12 25 30 44 55 66 77 88 99 AA BB CC DD EE 11 22
50 99 FF AA FF BB FF CC FF DD FF EE FF 11 FF 22 FF
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```



```
Time: 325us - Instructions: 228

000E|         INC R0
000F|         SJMP SCAN

      COPY:
0011|         MOV @R1, A
0012|         INC R1
0013|         INC R0
0014|         SJMP SCAN
      FILL:
0016|         MOV A, R1
0017|         CJNE A, #60H, ZERO
001A|         SJMP STOP
      ZERO:
001C|         MOV @R1, #00H
001E|         INC R1
001F|       SJMP FILL
      STOP:
0021|         SJMP STOP
              END
```

```
Data Memory
    0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
00 60 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
40 12 25 30 44 55 66 77 88 99 AA BB CC DD EE 11 22
50 99 FF AA FF BB FF CC FF DD FF EE FF 11 FF 22 FF
60 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
70 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```