

8-BIT DIVISION

EXP NO: 4

AIM: To write an assembly language program to implement 8-bit division using 8085 processor.

ALGORITHM:

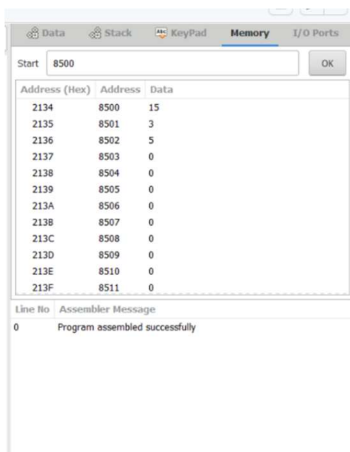
- 1) Start the program by loading a register pair with the address of memory location.
- 2) Move the data to a register.
- 3) Get the second data and load it into the accumulator.
- 4) Subtract the two register contents.
- 5) Increment the value of the carry.
- 6) Check whether the repeated subtraction is over.
- 7) Store the value of quotient and the remainder in the memory location.
- 8) Halt.

PROGRAM:

```
LDA 8501                                     1
MOV B,                                     1
LDA 8500      MVI C,00      LOOP: CMP      15
JC LOOP1      SUB      15
INR C      JMP LOOP      STA 8503      DCR      15
MOV A,      15

LOOP1: STA 8502
RST 1
```

INPUT:



OUTPUT:

The screenshot displays the 8085 processor simulator interface. The top bar shows the file name '53 KB' and a note 'Code 55% faster with GitHub Copilot'. The main window is divided into several sections:

- Registers:** A table showing the status of various registers. The PC (Program Counter) is at 42, and the SP (Stack Pointer) is at FF. The Z (Zero) flag is set to 1.
- Assembly Code:** A list of instructions being executed:
 1. MVI A, 0
 2. MVI B, 10
 3. MVI C, 5
 4. LOOP: CMP B
 5. JCZ C
 6. JNZ LOOP
 7. STA 8502
 8. HLT
- Memory:** A table showing the contents of memory locations. The address 8502 contains the value 05.
- I/O Ports:** A section for monitoring and controlling I/O ports.
- Assembly Messages:** A log showing the message 'Program assembled successfully'.

Register	Value	Flag
A	00	Z 0
BC	0A 00	Z 1
DE	00 00	
HE	00 00	
PSW	00 00	
PC	42 0F	P 1
SP	FF FF	C 1
Int-Flag	00	

Address (Hex)	Address	Data
2134	8500	15
2135	8501	3
2136	8502	5
2137	8503	0
2138	8504	0
2139	8505	0
213A	8506	0
213B	8507	0
213C	8508	0
213D	8509	0
213E	8510	0
213F	8511	0

RESULT: Thus the program was executed successfully using 8085 processor simulator.