

Write a program in 8085 to multiply two 8-bit numbers without using shifting multiplicand.

Address	Assembly Code	Hex code	Comments
0000	LHLD 2050H	2A 50 20	Loading data to HL reg pair
0003	MOV E, L	5D	Move data in L to E
0004	MOV A, H	7C	Move data in H to Acc
0005	MVI D, 00H	16 00	Moving 00 to D
0007	MVI C, 08H	0E 08	Move 8 to C
0009	LXI H, 0000H	21 00 00	Loading 0000 to HL reg pair
000C	BACK: DAD H	1F	HL <- HL+HL (left shift)
000D	RAL	17	Rotate accumulator left
000E	JNC NEXT	D2 11 00	Jump to next if no carry
0011	DAD D	19	HL <- HL + DE
0012	NEXT: DCR C	EB	Decrease C by 1
0013	JNZ BACK	29	Jump to Back if C not zero
0016	SHLD 2052	22 52 20	Store HL reg pair to 2052 and 2053
0019	HLT	76	Halt

Procedure:

Step – 1: Writing program in memory.

1. Press Reset
2. Press SET/MEM
3. Type in Address 0000
4. Press Enter
5. Type 1st Hex Code (Here 21)
6. Press Enter
7. Follow step 5 and 6 to type in all Hex Code

Step – 2: Assigning Value to the Address Location

1. Press Reset
2. Press SET/MEM
3. Type in Address of 1st Location (Here 2050)
4. Press Enter
5. Enter value of Multiplicand
6. Press Enter
7. Enter value of Multiplier
8. Press Enter

Step – 3: Executing the program

1. Press Reset to clear Buffer
2. Press Go
3. Enter Starting Address of Program (Here 0000)
4. Press Execute

Step – 4: Checking output

1. Press reset and Clear the Buffer
2. Press Go
3. Enter Result Location (Here 2052 for LB and 2053 for UB)
4. You will find the product of the numbers

Output:

The screenshot displays the 8085 Simulator interface. The **Assembler** window on the left shows a list of instructions with their addresses, labels, mnemonics, hexcodes, bytes, M-Cycles, and T-States. The **Memory Editor** window on the right shows the memory content from address 0000 to FFFF. The memory range is set to 0000 to FFFF. The memory content is displayed in a table with columns for Memory Address and Value. The values are: 0000: 2A, 0001: 50, 0002: 20, 0003: 5D, 0004: 7C, 0005: 16, 0006: 00, 0007: 0E, 0008: 08, 0009: 21, 000A: 00, 000B: 00, 000C: 29, 000D: 17, 000E: D2, 000F: 12, 0010: 00, 0011: 19, 0012: 0D, 0013: C2, 0014: 0C, 0015: 00, 0016: 22, 0017: 52, 0018: 20, 0019: 76, 2050: AA, 2051: 10, 2052: A0, 2053: 0A. A red 'X' is drawn over the memory address 0017, and a red checkmark is drawn over the memory address 0018.

8085 Simulator

File Edit Tools Settings Simulation Subroutine View Load Sample Program Help

Editor Assembler

* Address	Label	Mnemonics	Hexcode	Bytes	M-Cycles	T-States
✓ 0003		MOV E,L	5D	1	1	4
✓ 0004		MOV A,H	7C	1	1	4
✓ 0005		MVI D,00	16	2	2	7
0006			00			
✓ 0007		MVI C,08	0E	2	2	7
0008			08			
✓ 0009		LXI H,0000	21	3	3	10
000A			00			
000B			00			
✓ 000C	BACK	DAD H	29	1	3	10
✓ 000D		RAL	17	1	1	4
✓ 000E		JNC 0012	D2	3	3	10
000F			12			
0010			00			
✓ 0011		DAD D	19	1	3	10
✓ 0012	NEXT	DCR C	0D	1	1	4
✓ 0013		JNZ 000C	C2	3	3	10
0014			0C			
0015			00			

Simulate

Start From → 0000

Run all At a Time Step By Step

Registers Memory Devices

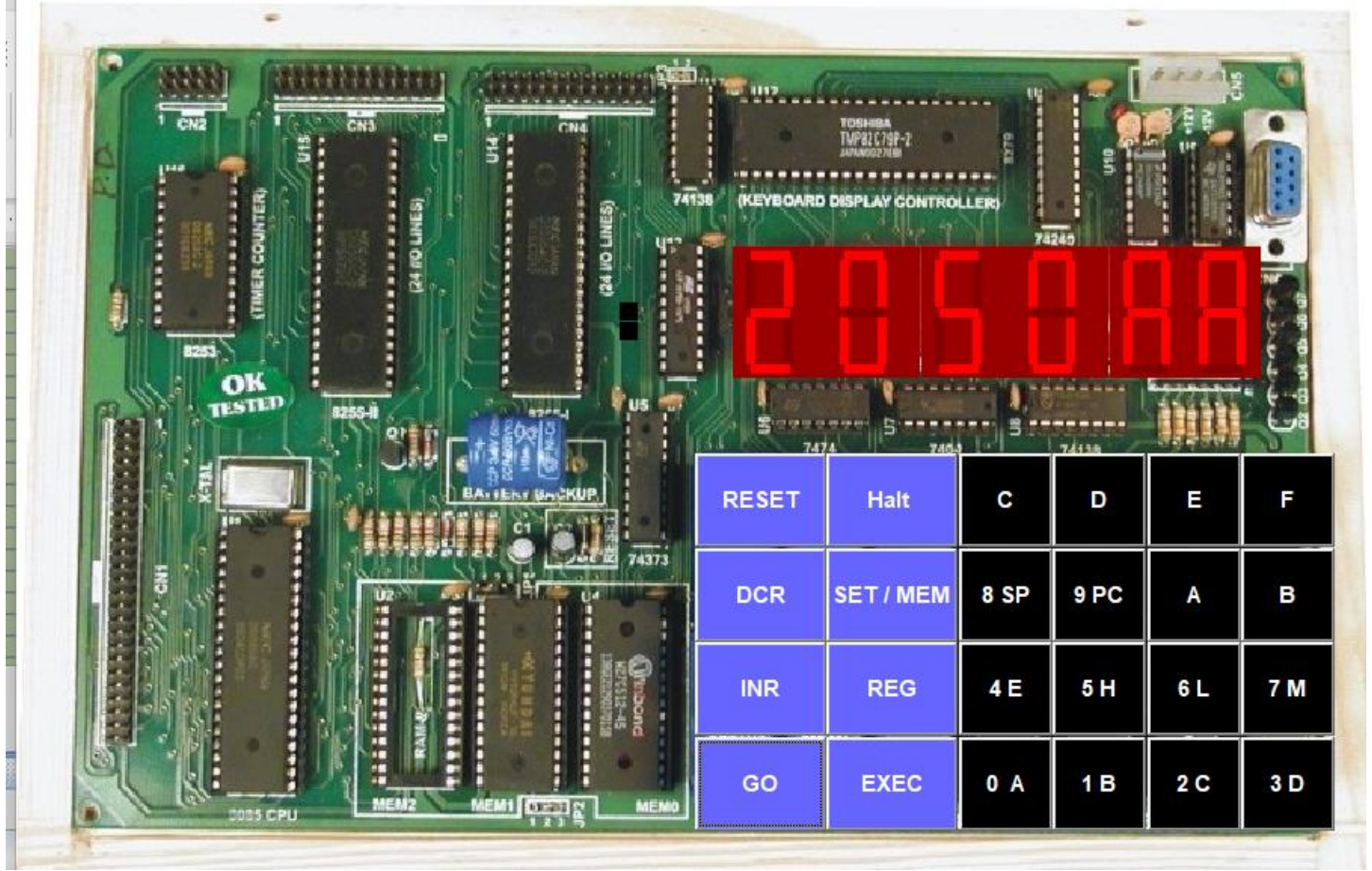
Memory Editor

Memory Range: 0000 ---- FFFF

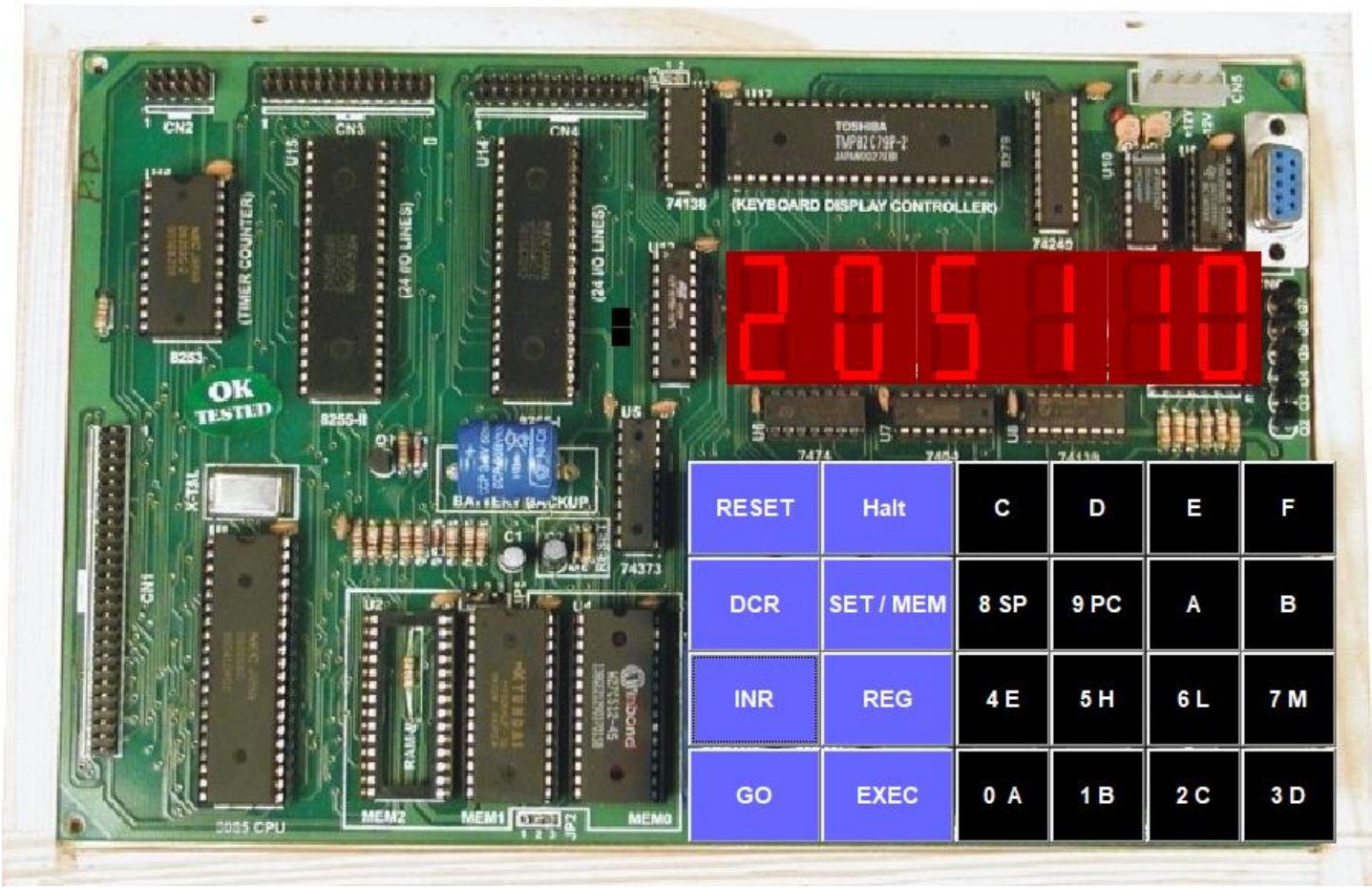
Memory Address	Value
0000	2A
0001	50
0002	20
0003	5D
0004	7C
0005	16
0007	0E
0008	08
0009	21
000C	29
000D	17
000E	D2
000F	12
0011	19
0012	0D
0013	C2
0014	0C
0016	22
0017	52
0018	20
0019	76
2050	AA
2051	10
2052	A0
2053	0A

☐ Show entire memory content
☒ Show only loaded memory location
☐ Store directly to specified memory location

8085 MICROPROCESSOR TRAINER KIT



8085 MICROPROCESSOR TRAINER KIT



8085 MICROPROCESSOR TRAINER KIT

