

**Introduction to computer architecture (ce243)**

***Project : MARIE assembly language, scientific calculator***

Habiba Hassan 20104924

Rawan Mahmoud 20109528

Computer Science, semester 4

4B

Supervised by: Dr. Aliaa Youssef

TA. Mohanad Mahmoud

**Table of contents**

1. RTL/Dump memory/Registers …………Pages 3..113
2. Marie code with explanation comments ..Page 114..139
3. User instructions…………..…………….Page 140..141

**1-RTL LOG refers to addition operation:-**

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 1  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 2  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 3  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

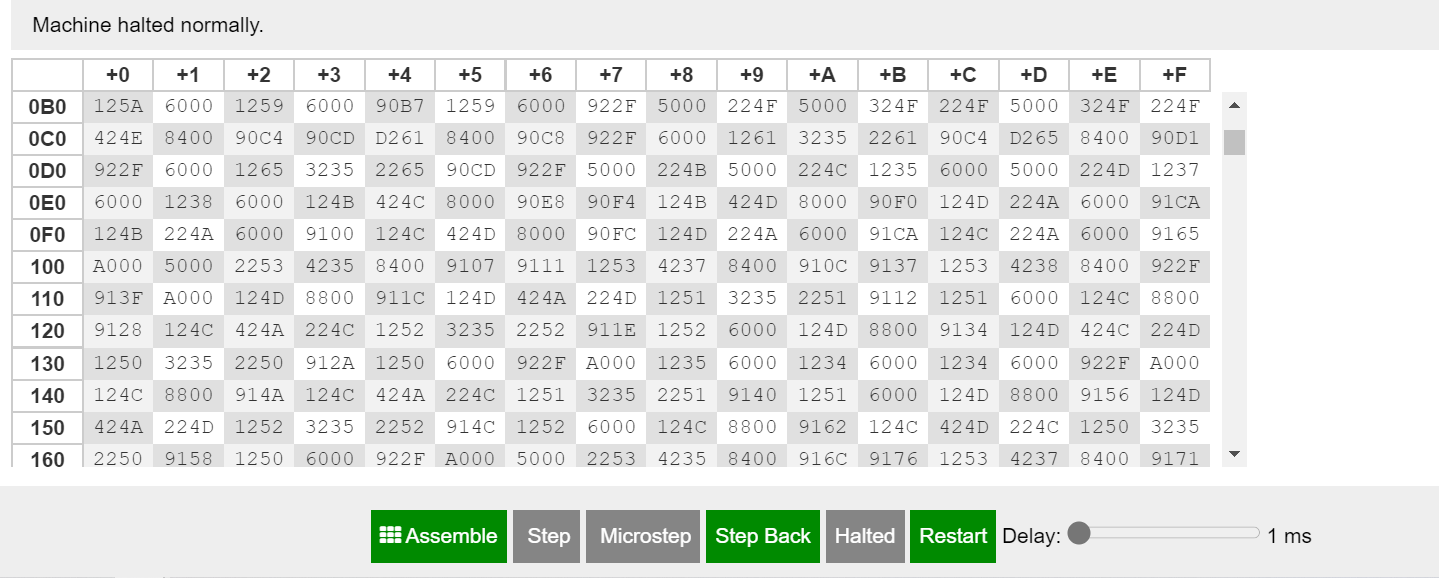
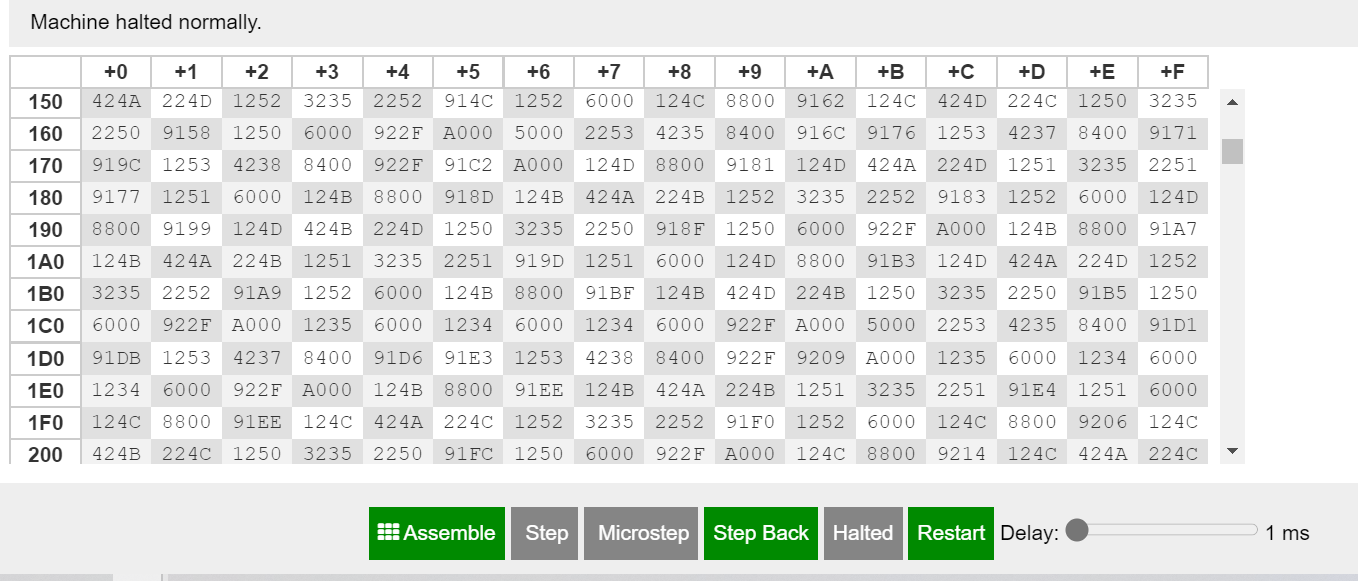
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

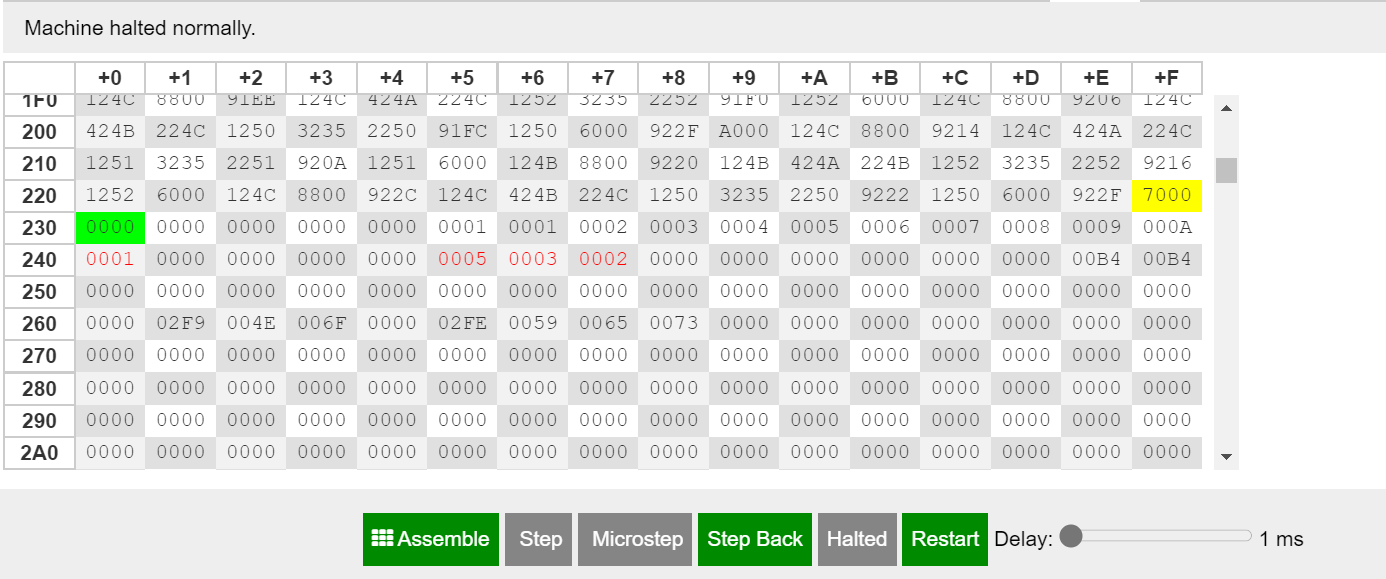
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 7 in IR[15-12] as halt  
----- halted -----

**Dump memory**







**Registers**

Text

Description automatically generated

**2-RTL LOG refers to subtraction operation:-**

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 2  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 4  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 2  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

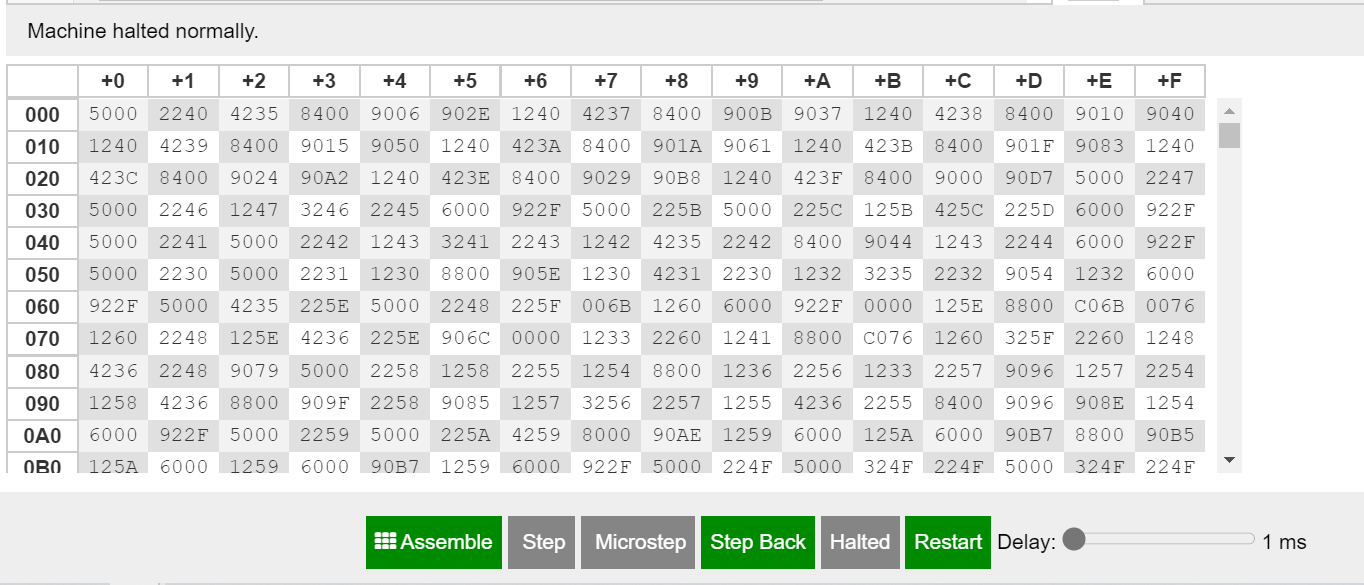
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

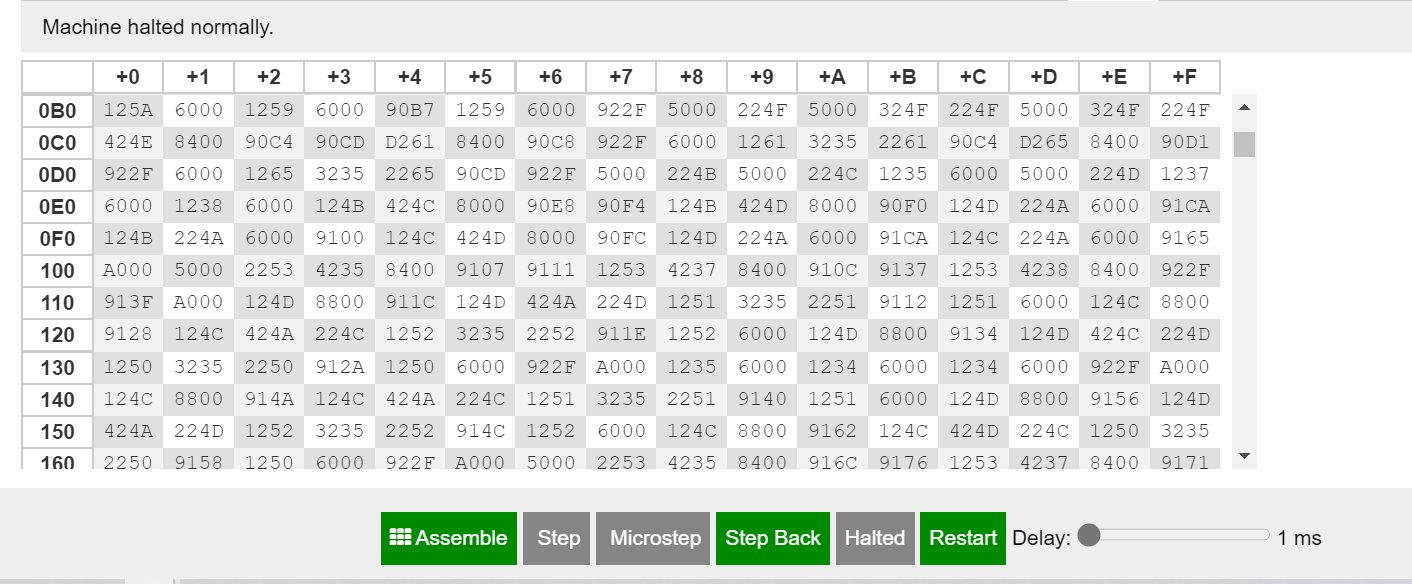
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

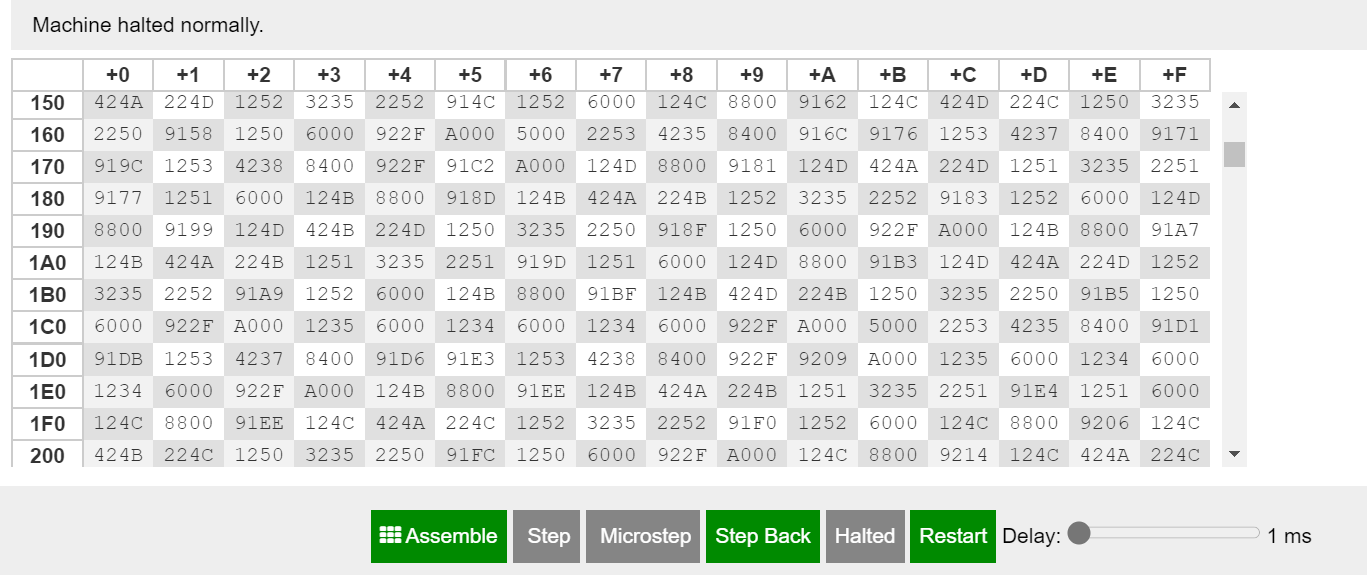
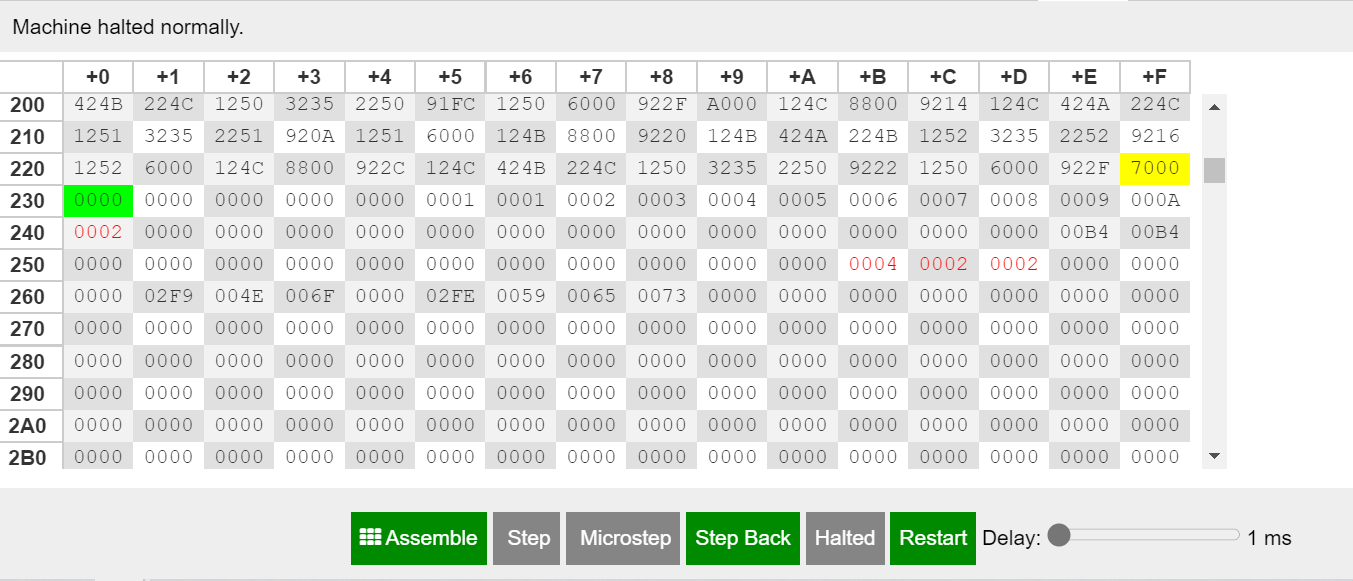
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 7 in IR[15-12] as halt  
----- halted -----

**Dump memory**



**Registers**

****

**3-RTL log refers to multiplication operation:-**

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 3  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 4  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 4  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

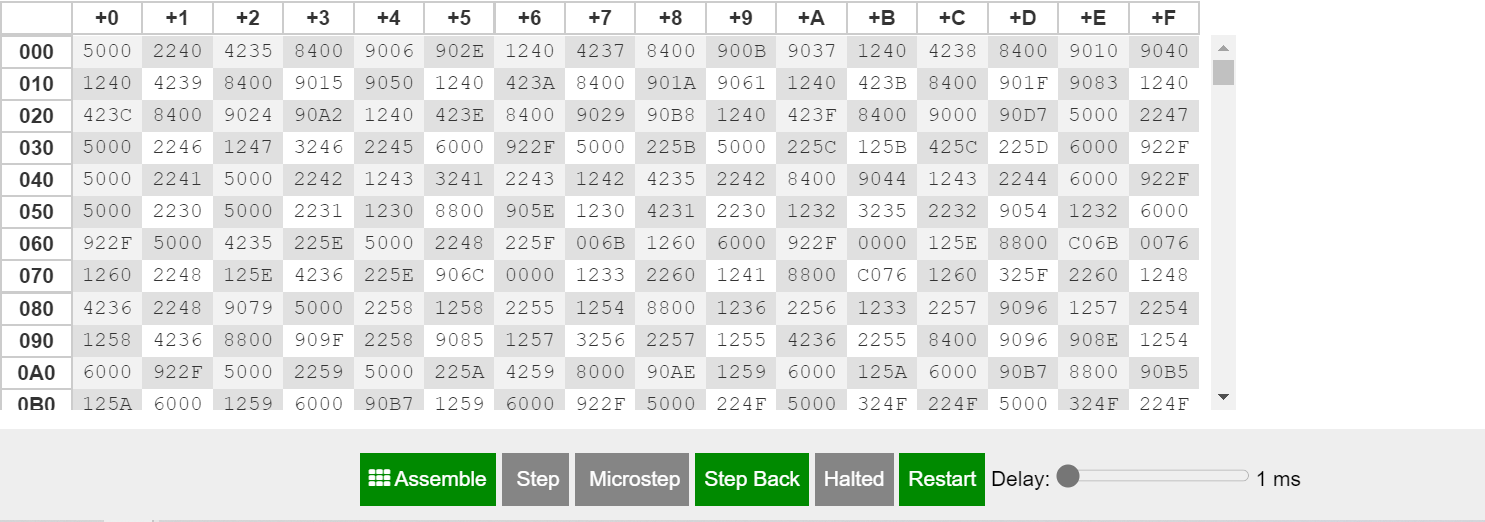
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

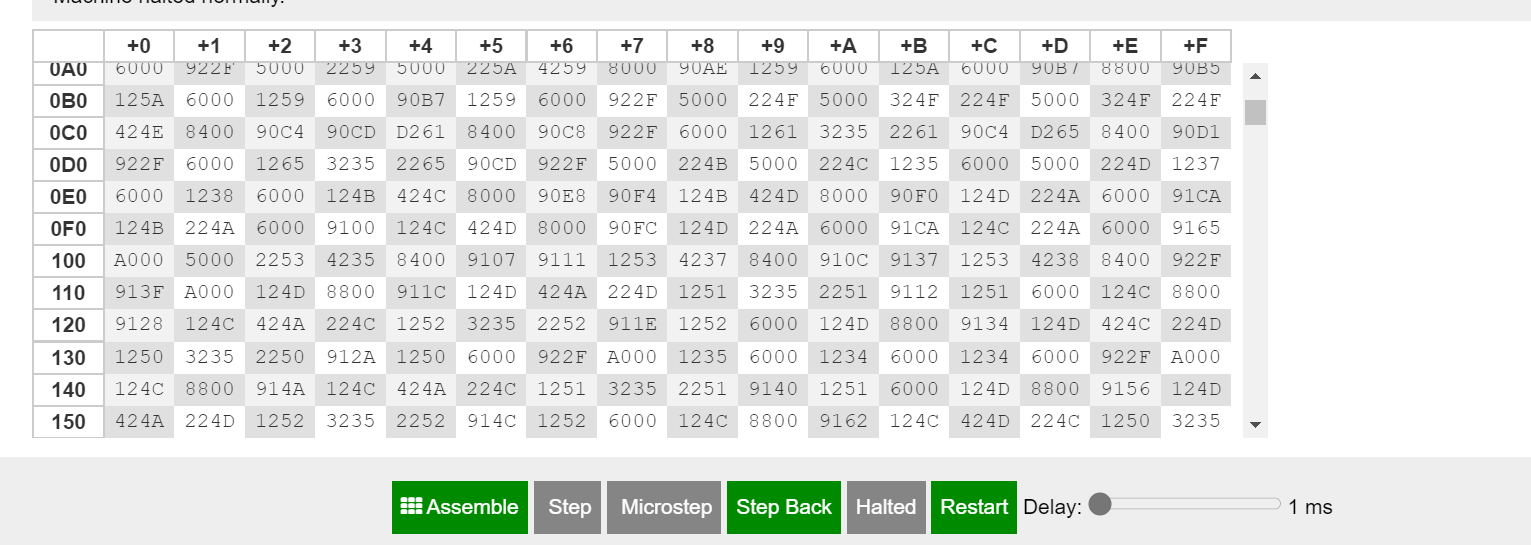
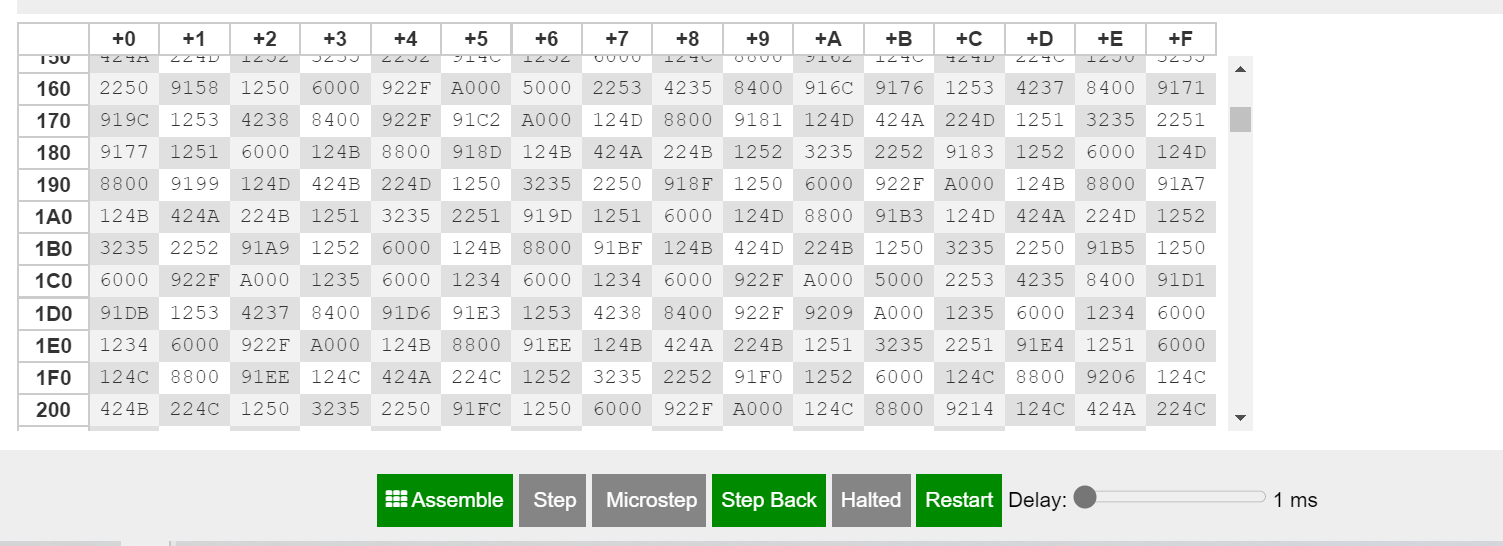
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

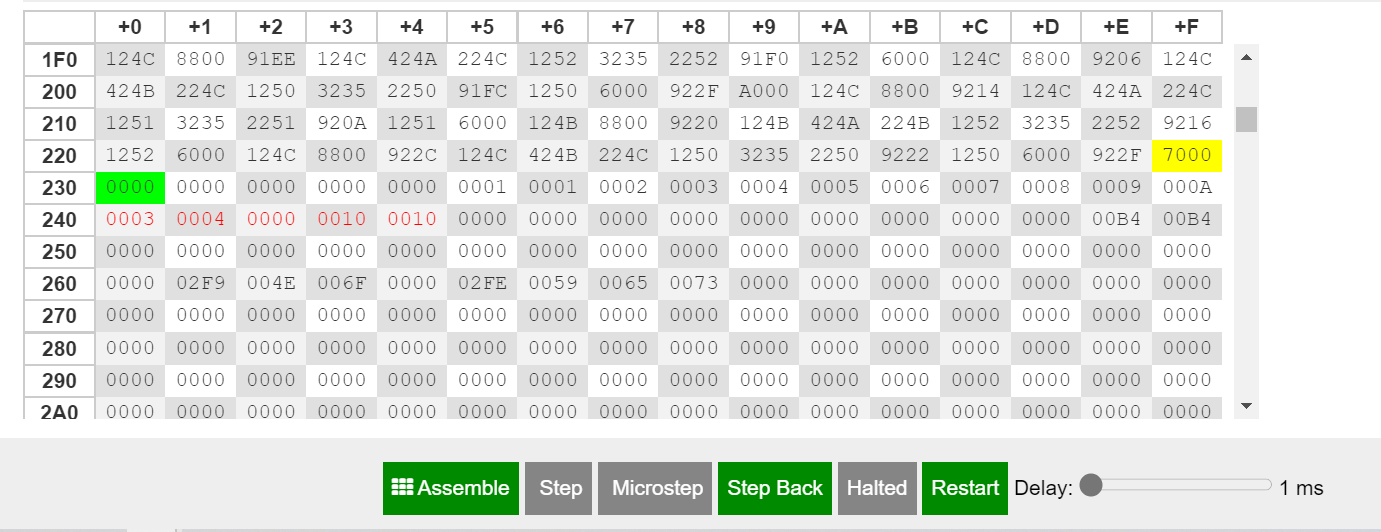
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 7 in IR[15-12] as halt  
----- halted -----

**Dump memory**







**Registers**

Text

Description automatically generated with medium confidence

**4-RTL LOG refers to division operation: -**

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 4  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← F  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 5  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

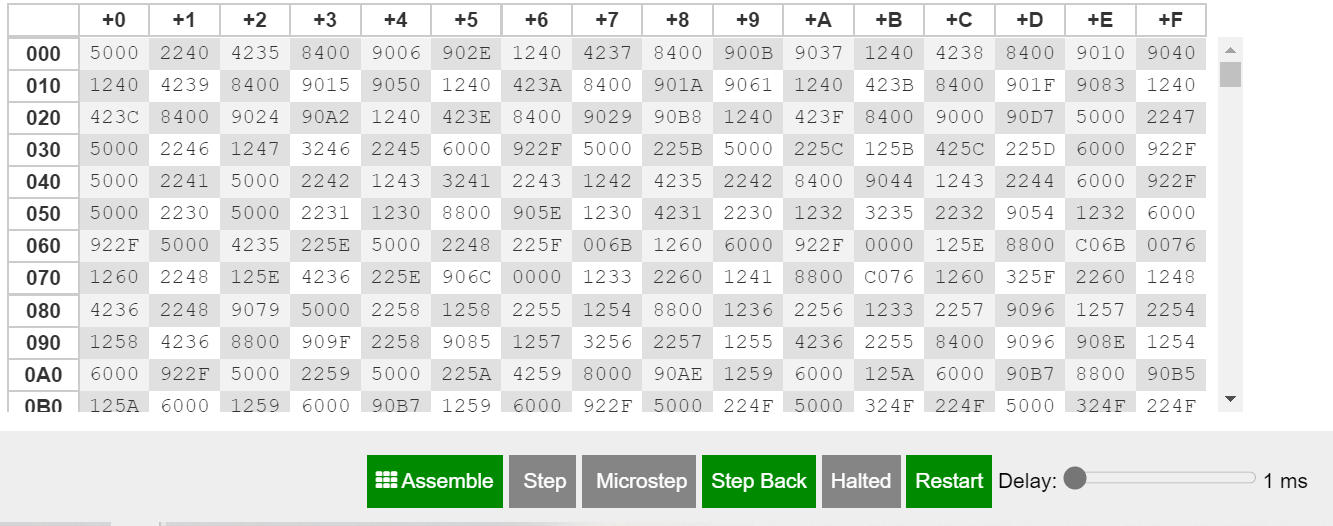
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

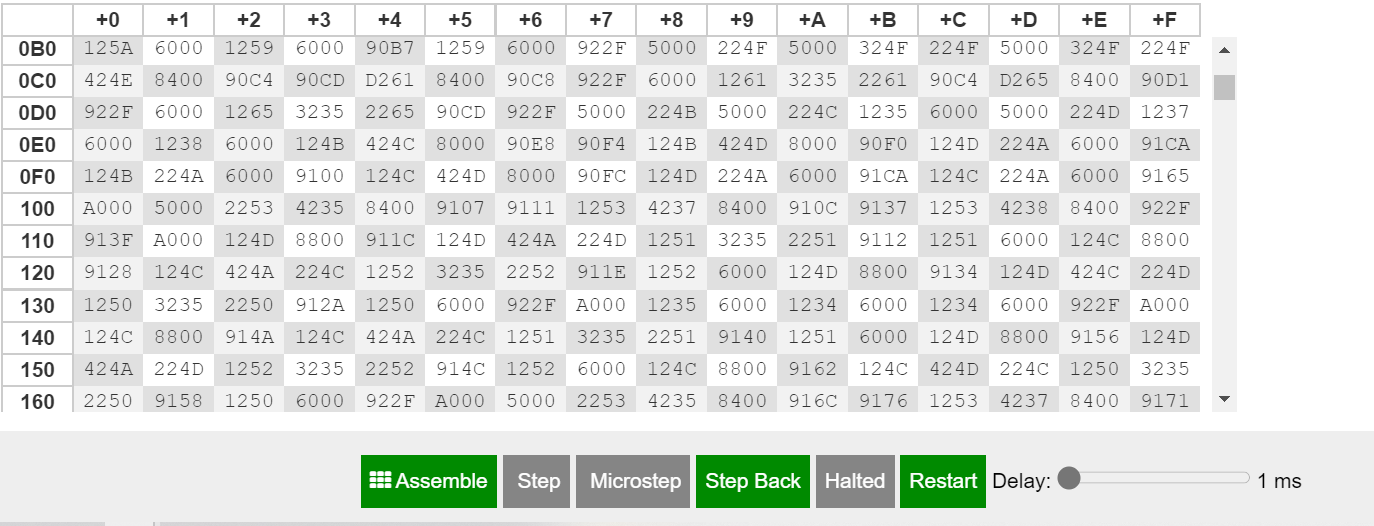
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

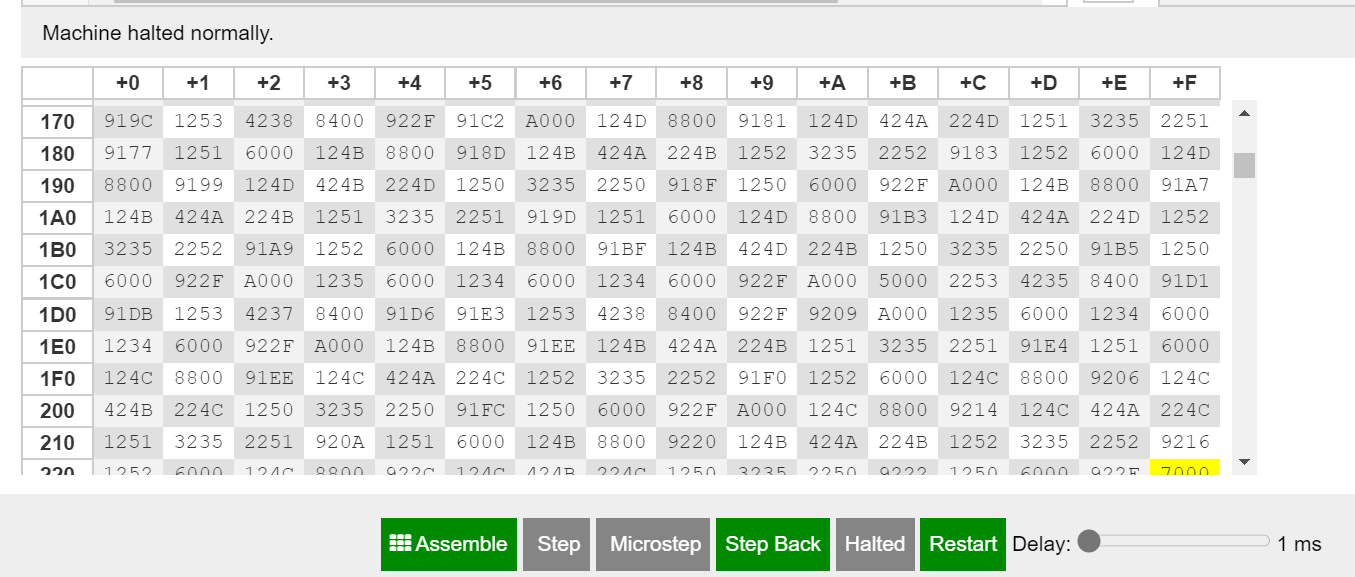
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

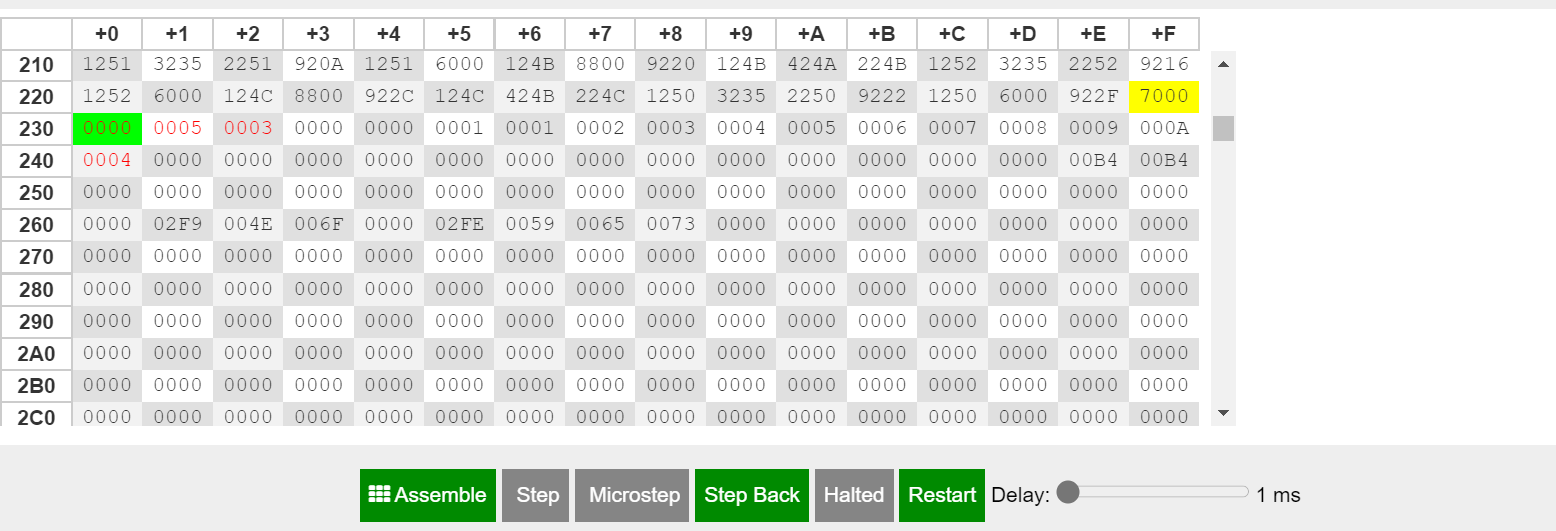
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 7 in IR[15-12] as halt  
----- halted -----

**Dump memory**

****







**Registers**

A screenshot of a computer

Description automatically generated

**5-RTL LOG refers to power operation: -**

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 5  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 2  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 3  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 0 in IR[15-12] as jns  
MAR ← IR[11-0]  
MBR ← PC  
M[MAR] ← MBR  
PC ← MAR  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 0 in IR[15-12] as jns  
MAR ← IR[11-0]  
MBR ← PC  
M[MAR] ← MBR  
PC ← MAR  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode C in IR[15-12] as jumpi  
MAR ← IR[11-0]  
MBR ← M[MAR]  
PC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode C in IR[15-12] as jumpi  
MAR ← IR[11-0]  
MBR ← M[MAR]  
PC ← MBR

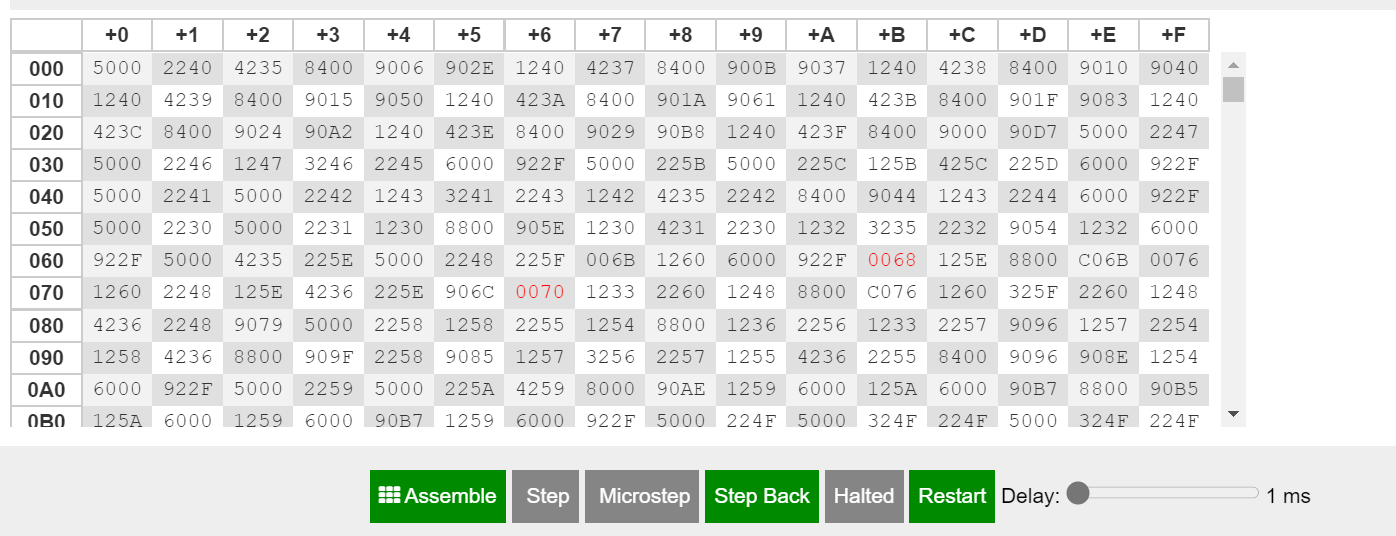
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

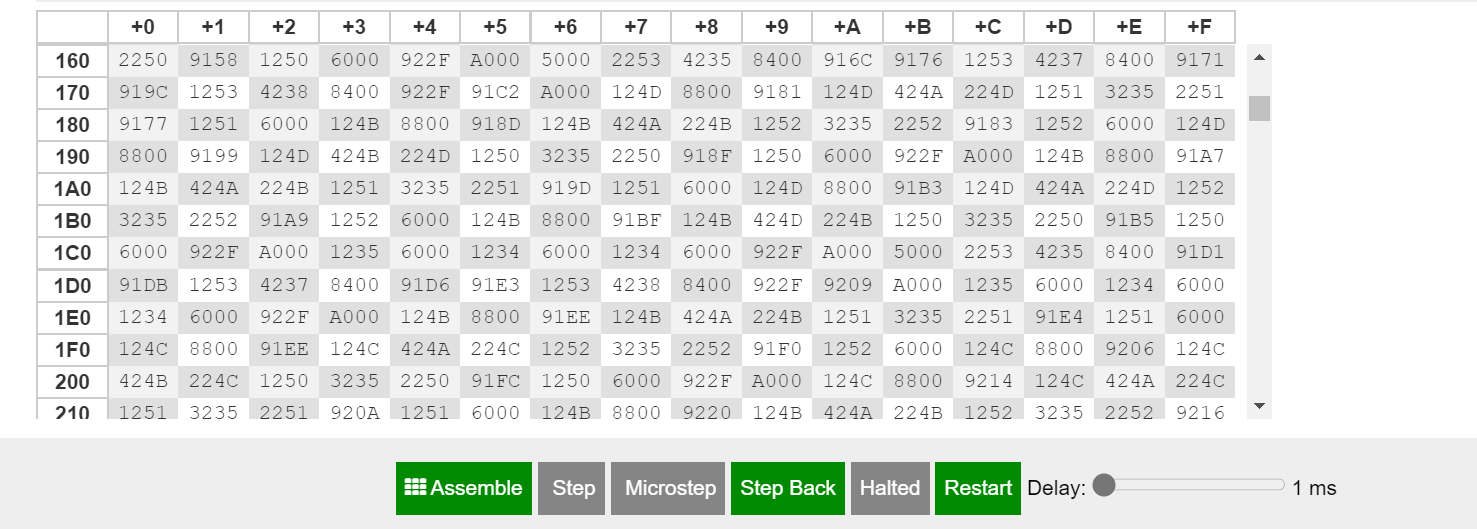
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

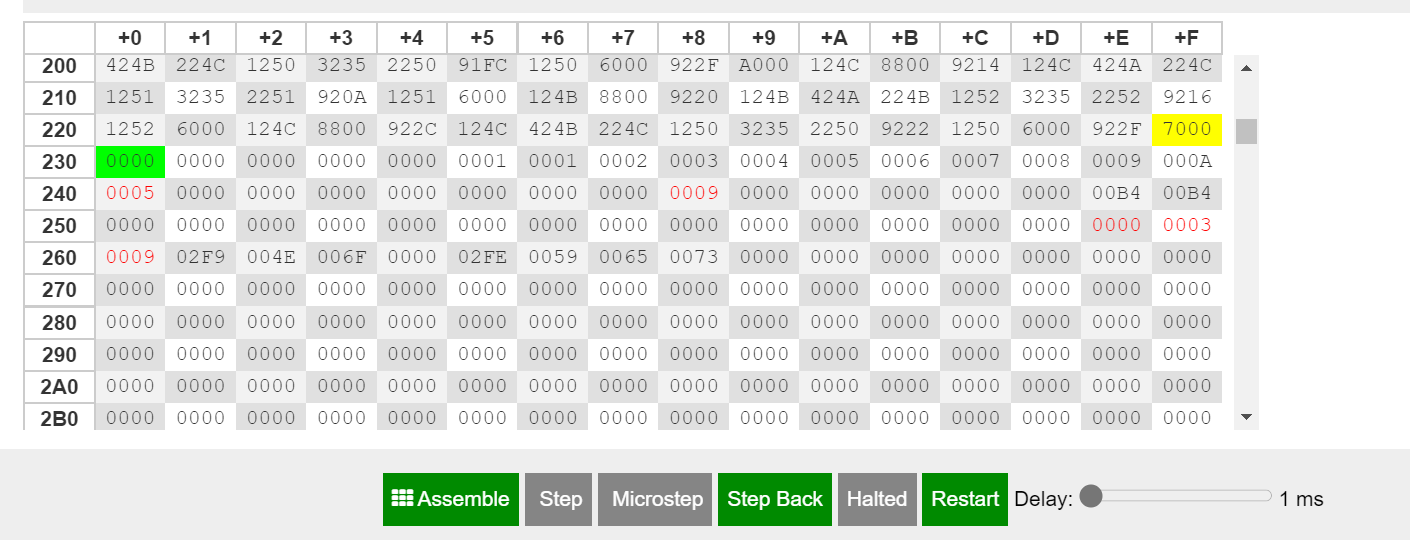
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 7 in IR[15-12] as halt  
----- halted -----

**Dump memory**









**Registers**

A screenshot of a computer

Description automatically generated

**6-RTL LOG refers to factorial operation: -**

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 6  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 5  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC > 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

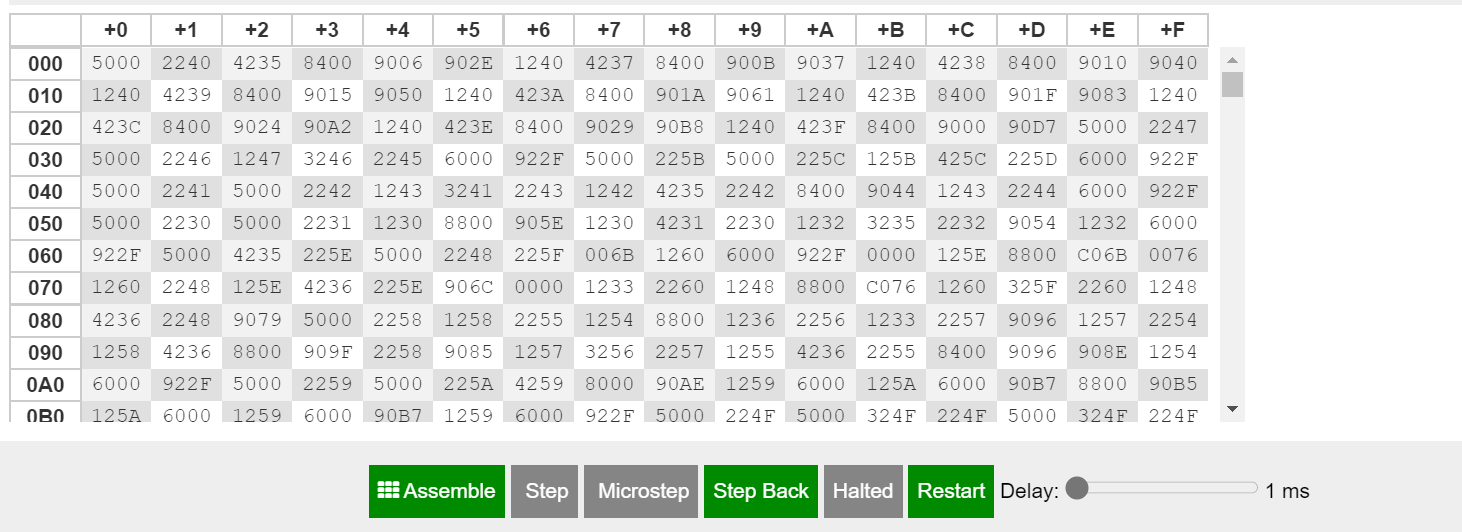
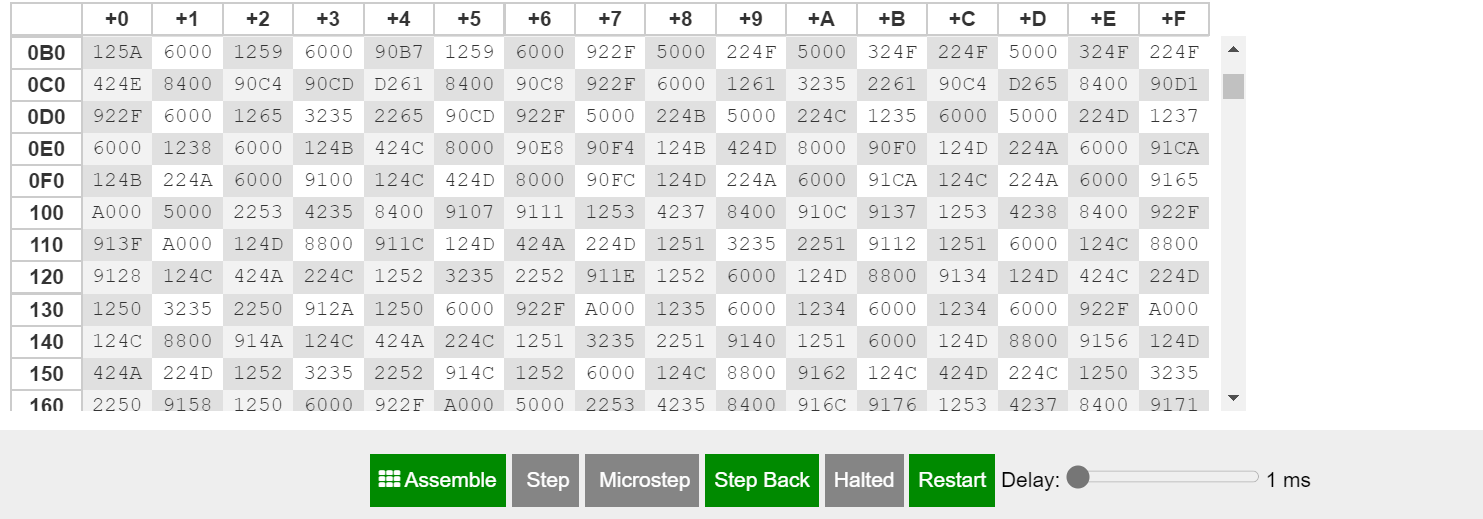
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

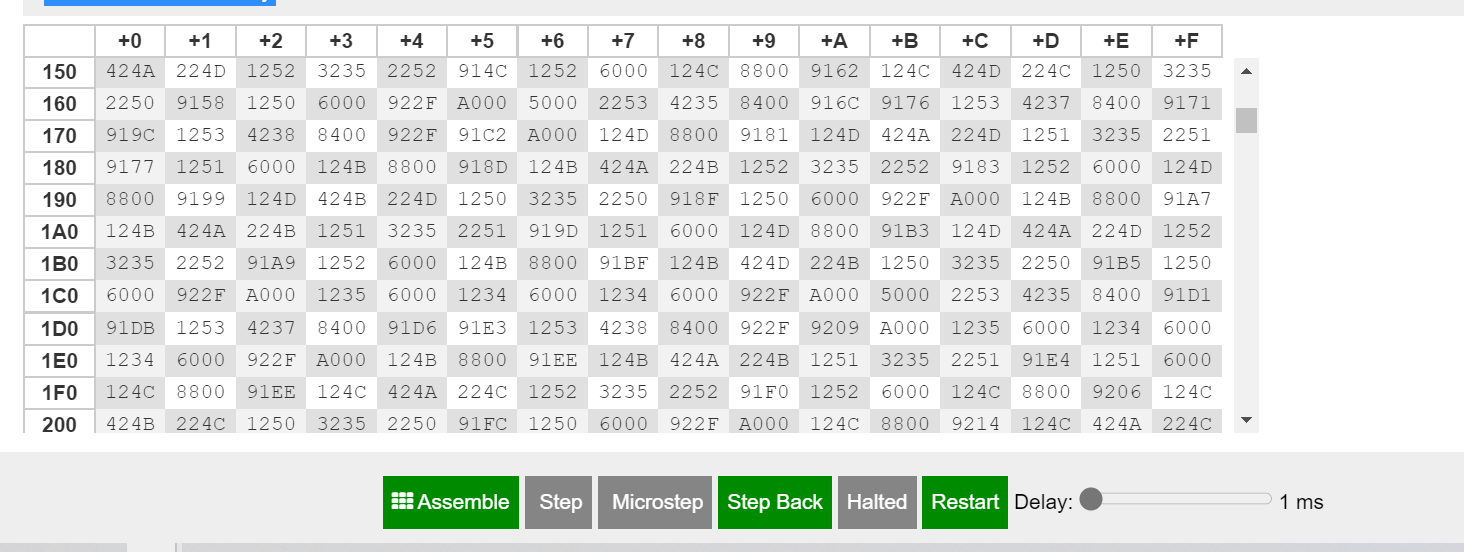
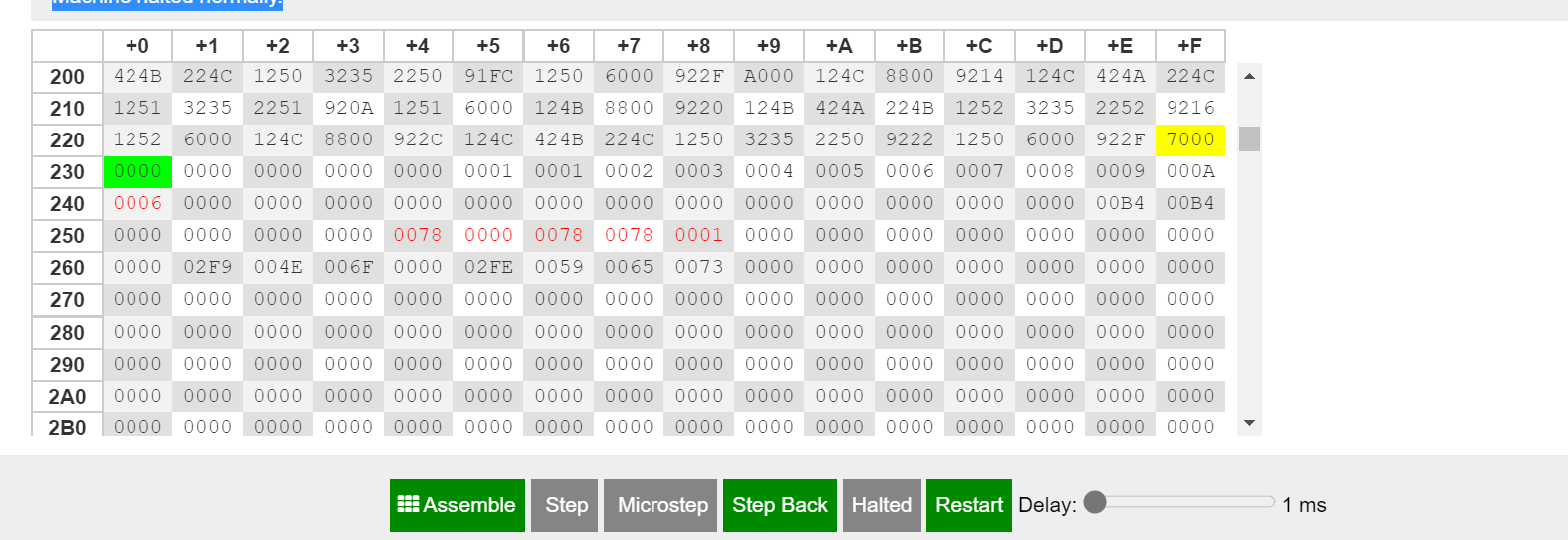
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 7 in IR[15-12] as halt  
----- halted -----

**Dump memory**



**Registers**

A screenshot of a computer

Description automatically generated

**7-RTL LOG refers to max, min, equal operations :-**

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 7  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 6  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 5  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC < 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

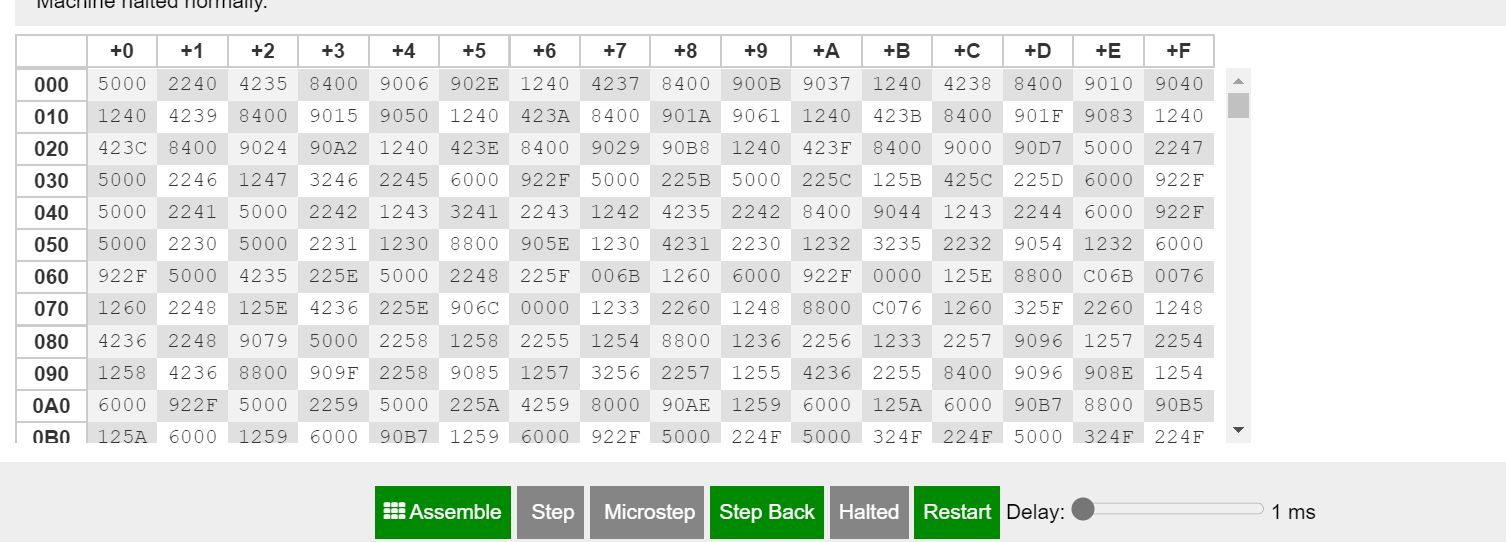
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

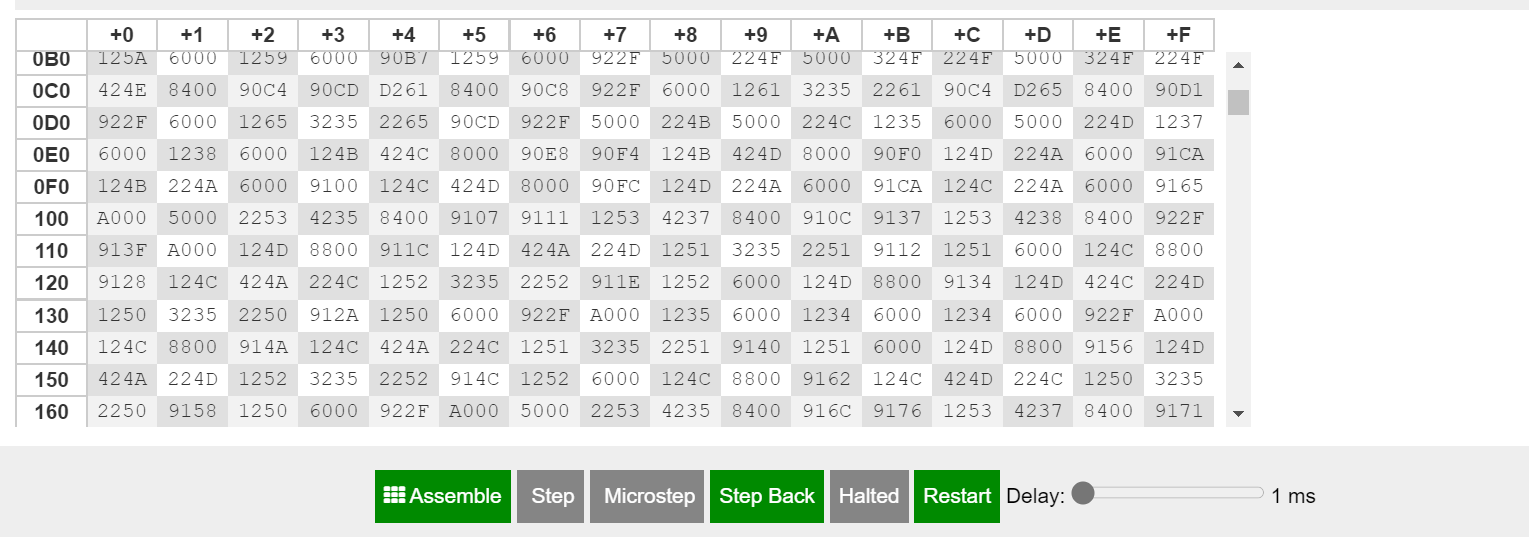
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

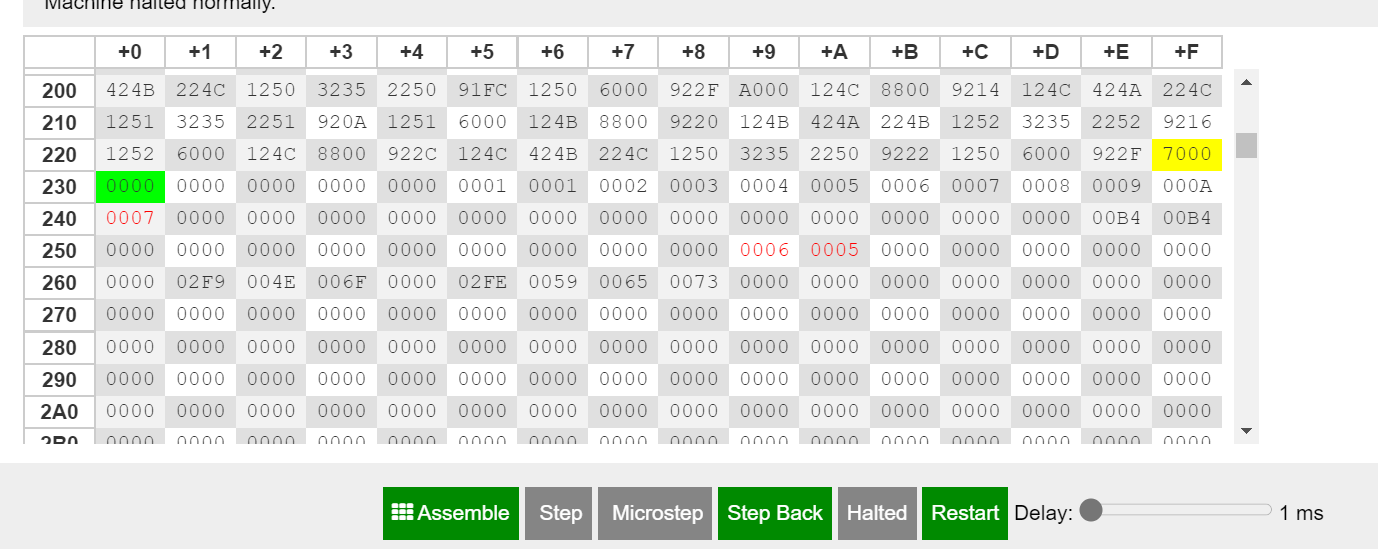
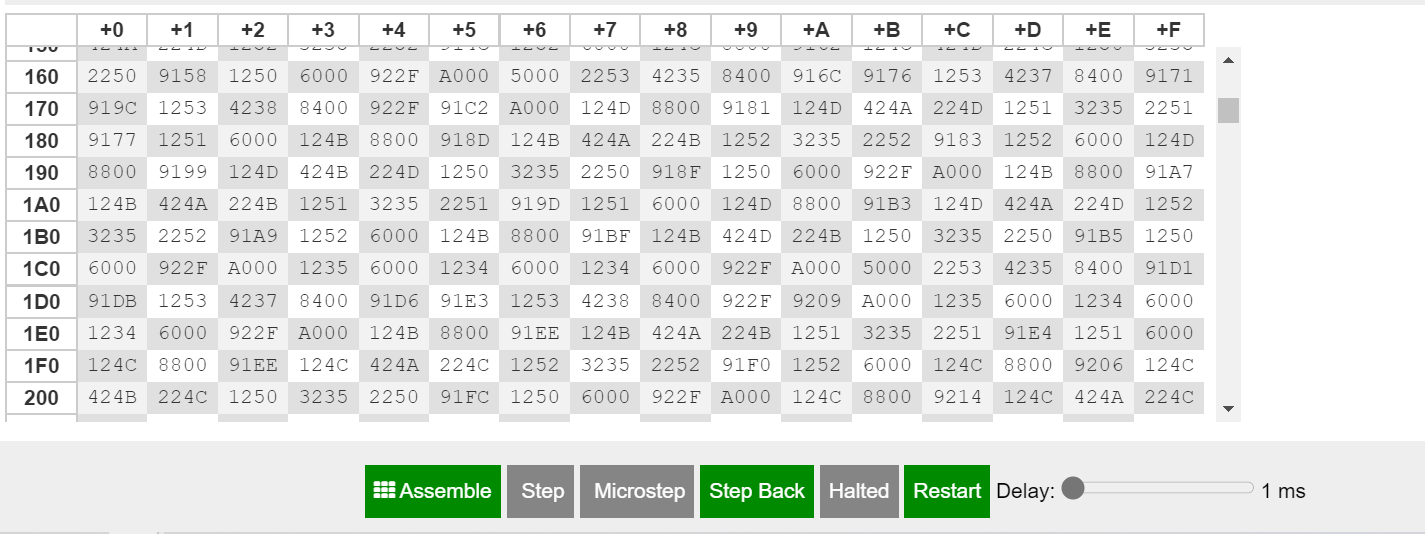
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 7 in IR[15-12] as halt  
----- halted -----

**Dump memory**



**Registers**

A screenshot of a computer

Description automatically generated

**9-RTL LOG refers to triangle detection operation :-**

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 9  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 32  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 32  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 50  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 7 in IR[15-12] as halt  
----- halted -----

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 9  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 50  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 50  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 64  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 3 in IR[15-12] as add  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC + MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 7 in IR[15-12] as halt  
----- halted -----

Graphical user interface, text

Description automatically generatedGraphical user interface, text

Description automatically generatedA screenshot of a computer

Description automatically generatedGraphical user interface, text

Description automatically generated**Dump Memory**

**Registers**

A screenshot of a computer

Description automatically generated

**10-RTL LOG refers to trigonometry operations :-**

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← A  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? No!

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 3  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 4  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 5  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC < 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC < 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode A in IR[15-12] as clear  
AC ← 0

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 5 in IR[15-12] as input  
IN ← 1  
AC ← IN

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 2 in IR[15-12] as store  
MAR ← IR[11-0]  
MBR ← AC  
M[MAR] ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 4 in IR[15-12] as subt  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← AC - MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 8 in IR[15-12] as skipcond  
Is AC = 0? Yes!  
PC ← PC + 1

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode A in IR[15-12] as clear  
AC ← 0

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

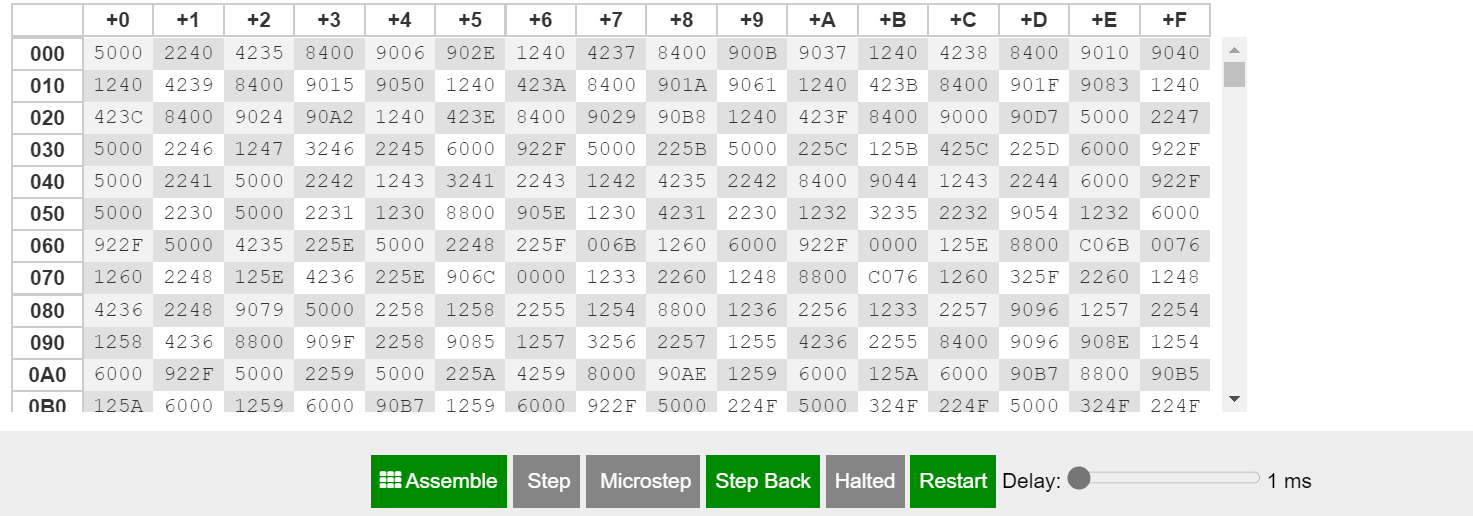
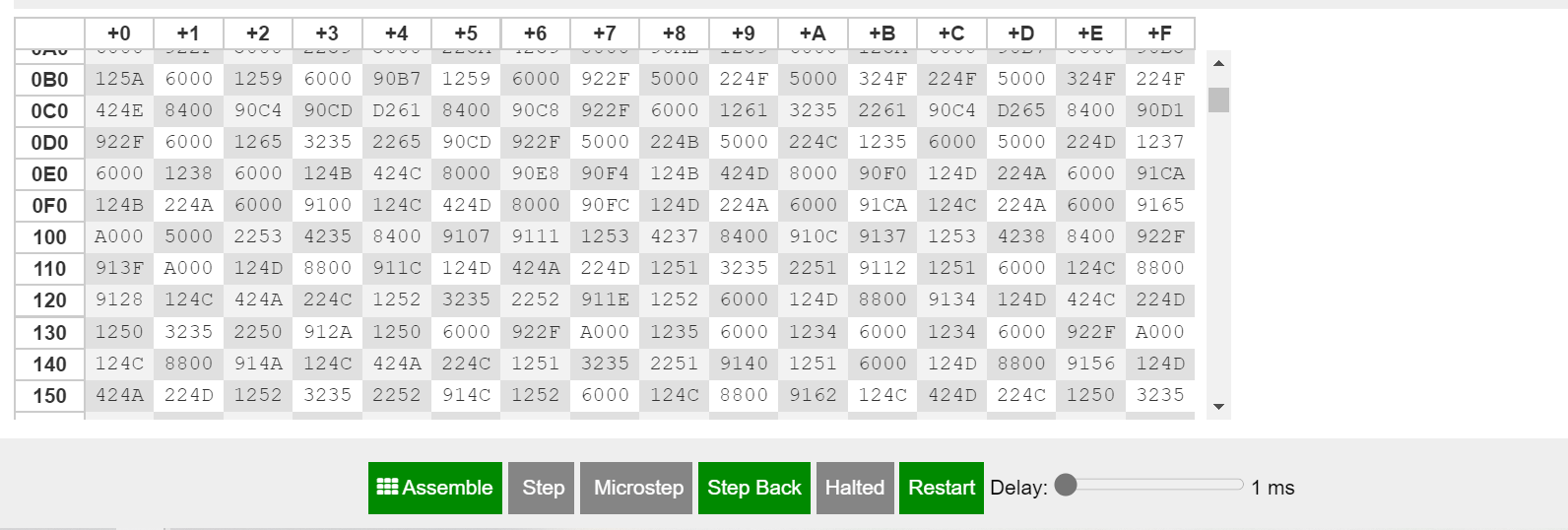
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 1 in IR[15-12] as load  
MAR ← IR[11-0]  
MBR ← M[MAR]  
AC ← MBR

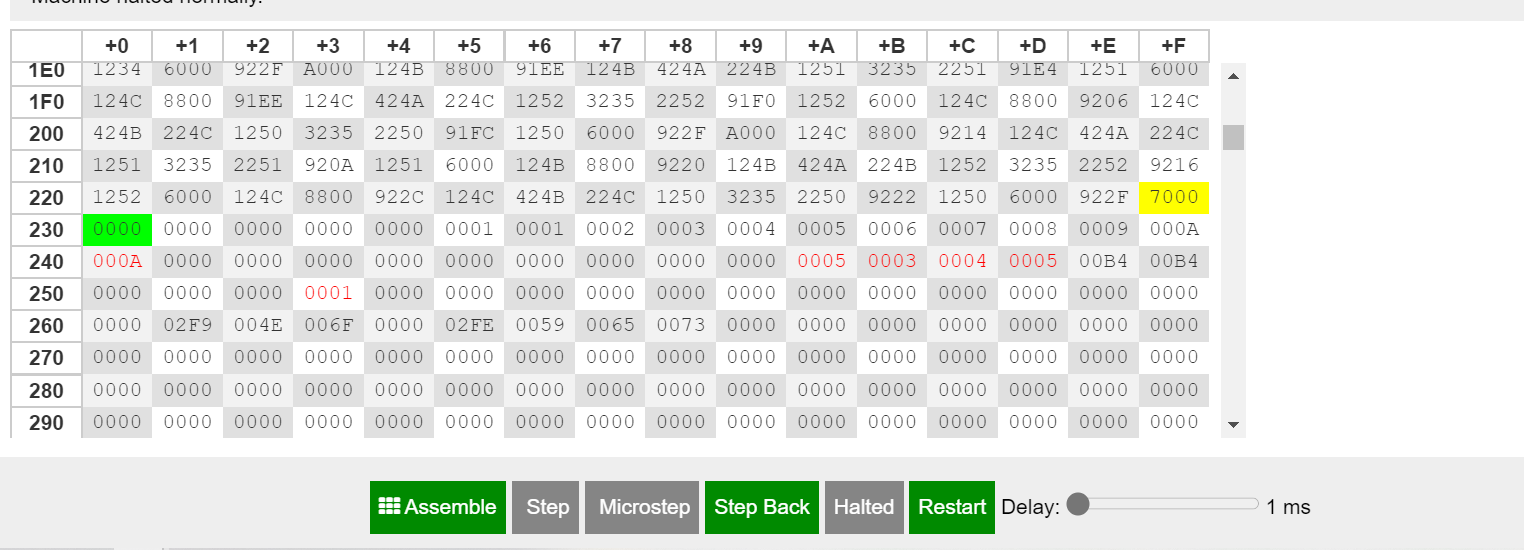
MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 6 in IR[15-12] as output  
OUT ← AC

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 9 in IR[15-12] as jump  
PC ← IR[11-0]

MAR ← PC  
MBR ← M[MAR]  
IR ← MBR  
PC ← PC + 1  
Decoded opcode 7 in IR[15-12] as halt  
----- halted -----

**Dump memory**





**Registers**

A screenshot of a computer

Description automatically generated

**The MARIE code**

//enter what operation the user wants

//the user enters the number assigned to the operarion wanted

//the the program detects what number he put

main, Input

Store o

subt one

Skipcond 400

Jump test1

Jump addition

test1, Load o

subt two

Skipcond 400

Jump test2

Jump subt

test2, Load o

subt three

Skipcond 400

Jump test3

Jump multiplication

test3, Load o

subt four

Skipcond 400

Jump test4

Jump div

test4, Load o

subt five

Skipcond 400

Jump test5

Jump power

test5, Load o

subt six

Skipcond 400

Jump test6

Jump factorial

test6, Load o

subt seven

Skipcond 400

Jump test8

Jump maxMin

test8, Load o

subt nine

Skipcond 400

Jump test9

Jump detection

test9, Load o

subt ten

Skipcond 400

Jump main

Jump trigonometry

//addition code

addition, Input

Store c

Input

Store d

load c

add d

store sumResult

Output

jump end

//subtraction code

subt,Input

Store e

Input

Store f

load e

Subt f

store subResult

Output

Jump end

//multiplication code

multiplication, Input

Store x

Input

Store y

mult, load w

add x

store w

load y

subt one

store y

Skipcond 400

jump mult

load w

store multResult

output

Jump end

//division code

div, Input

store a

Input

Store b

if, load a

Skipcond 800

jump endif

then, load a

Subt b

Store a

load z

add one

store z

jump if

endif, load z

output

Jump end

//power

power,Input / Enter the exponent Store i

Subt one

Store Count

Input / Enter the Base

Store t

Store i

Jns Exp

/ Ending the main program

Load Ans

Output

Jump end

Exp, Hex 0

Loop2, Load Count

Skipcond 800

JumpI Exp

JnS Multiplier

Load Ans

Store t

Load Count

Subt One

Store Count

Jump Loop2

/ Start of the subroutine Multiplier

Multiplier, Hex 0

Load Zero

Store Ans

Loop, Load t

Skipcond 800

JumpI Multiplier

Load Ans

Add i

Store Ans

Load t

Subt One

Store t

Jump Loop

// factorial

factorial,Input

Store Fact

Fac, Load Fact

Store MulOpA

Load Sum

Skipcond 800 /This is to prevent multiply wi

Load One /Start multiply by One

Store MulOpB /Store in MulOpB variable

Load Zero

Store MulSum

Jump Mul

FacLoop, Load MulSum

Store Sum

Load Fact

Subt One

Skipcond 800

Jump Finish

Store Fact

Jump Fac

Mul, Load MulSum

Add MulOpB

Store MulSum

Load MulOpA

Subt One

Store MulOpA

Skipcond 400

Jump Mul

Jump FacLoop

Finish, Load Sum

Output

jump end

//sort from max to min

//the user enter 2 numbers and the program will print them in descending order

/max and min

maxMin, input / get value for p

store p

input / get value for s

store s

Testm1, subt p / acc = s - p

skipcond 000 / if negative then p bigger, skip further tests

jump Testm2

load p / got here if p bigger, print p then s, then done

output

load s

output

jump Done

Testm2, skipcond 800 / if positive then s bigger, skip further tests

jump XYEq / if here then equal, go to XYeq print section

load s / got here if s bigger, print s then p, then done

output

load p / then print s

output

jump Done

XYEq, load p / just print p, then done

output

Done, jump end

//triangle detection

//the user enters the three angles of the triangle

//the program will then checck their sum

detection, Input

Store totalAngles

Input

Add totalAngles

Store totalAngles

Input

Add totalAngles

Store totalAngles

subt hundredEighty

skipcond 400 //if the sum is 180, the program will print 1

jump loopNo // if the sum is not 180, the program will print 0

load one

Output

jump end

loopNo, Load zero

Output

jump end

//sin/cos/tan

trigonometry, Input

//first, the user will enter the three sides of the triangle

Store side1

Input

Store side2

Load one

output

//the angle between side1 and side2 is named 1

Input

Store side3

load two

output

//the angle between side2 and side3 is named 2

load three

output

//the angle between side3 and side1 is named 3

//then, it will compare between the three sides to find the biggest one, which will be the hypothenuse

Load side1

subt side2

Skipcond 000 // if side 2 is bigger than side 1

jump check // if side 1 is >side2

Jump check1

//compare between side 1 and side 3

check, Load side1

subt side3

Skipcond 000 //if side 3 is bigger

Jump hypo

load side3

store hypothenuse

Output

Jump case3

//if side 1 is bigger

hypo,Load side1

Store hypothenuse

Output

Jump case1

check1,Load side2

subt side3

Skipcond 000 //if side 3 is bigger

Jump hypo1

load side3

store hypothenuse

Output

Jump case3

//if side 2 is bigger

hypo1, load side2

Store hypothenuse

Output

Jump case2

/when the hypothenuse is side 1

case1, clear

/the program will detect which angle is chosen

Input

Store ang

subt one

Skipcond 400

Jump testAngle1

Jump angle1

testAngle1, Load ang

subt two

Skipcond 400

Jump testAngle2

Jump angle2

testAngle2, Load ang

subt three

Skipcond 400

Jump end

Jump angle3

//the user will then choose the angle he wants its sin, cos and tan

// the program will give him all the results in one time

//sin

//divide side 3 which is the opposite side by the hypotenus

angle1, clear

if1, load side3

Skipcond 800

jump endif1

then1, load side3

Subt hypothenuse

Store side3

load sinResult

add one

store sinResult

jump if1

endif1, load sinResult

output

//cos

//divide side 2 which is the adjacent side by the hypotenuse

if2, load side2

Skipcond 800

jump endif2

then2, load side2

Subt hypothenuse

Store side2

load cosResult

add one

store cosResult

jump if2

endif2, load cosResult

output

//tan

//divide side 3 which is the opposite side by side2 which is the adjacent

if3, load side3

Skipcond 800

jump endif3

then3, load side3

Subt side2

Store side3

load tanResult

add one

store tanResult

jump if3

endif3, load tanResult

output

jump end

angle2, clear

//sin

load one

output

//cos

load zero

output

//tan

load zero

Output

jump end

angle3, clear

//sin

if4, load side2

Skipcond 800

jump endif4

then4, load side2

Subt hypothenuse

Store side2

load sinResult

add one

store sinResult

jump if4

endif4, load sinResult

output

//cos

//divide side 3 which is the adjacent side by the hypotenuse

if5, load side3

Skipcond 800

jump endif5

then5, load side3

Subt hypothenuse

Store side3

load cosResult

add one

store cosResult

jump if5

endif5, load cosResult

output

//tan

//divide side 3 which is the opposite side by side2 which is the adjacent

if6, load side2

Skipcond 800

jump endif6

then6, load side2

Subt side3

Store side2

load tanResult

add one

store tanResult

jump if6

endif6, load tanResult

output

jump end

//when the hypothenuse is side 2

case2, clear

//the program will detect which angle is chosen

Input

Store ang

subt one

Skipcond 400

Jump testAnglee1

Jump anglee1

testAnglee1, Load ang

subt two

Skipcond 400

Jump testAnglee2

Jump anglee2

testAnglee2, Load ang

subt three

Skipcond 400

Jump end

Jump anglee3

//sin

//divide side 3 which is the opposite side by the hypotenus

anglee1, clear

if7, load side3

Skipcond 800

jump endif7

then7, load side3

Subt hypothenuse

Store side3

load sinResult

add one

store sinResult

jump if7

endif7, load sinResult

output

//cos

//divide side 1 which is the adjacent side by the hypotenuse

if8, load side1

Skipcond 800

jump endif8

then8, load side1

Subt hypothenuse

Store side1

load cosResult

add one

store cosResult

jump if8

endif8, load cosResult

output

//tan

//divide side 3 which is the opposite side by side1 which is the adjacent

if9, load side3

Skipcond 800

jump endif9

then9, load side3

Subt side1

Store side3

load tanResult

add one

store tanResult

jump if9

endif9, load tanResult

output

jump end

anglee2, clear

//sin

if10, load side1

Skipcond 800

jump endif10

then10, load side1

Subt hypothenuse

Store side1

load sinResult

add one

store sinResult

jump if10

endif10, load sinResult

output

//cos

//divide side 3 which is the adjacent side by the hypotenuse

if11, load side3

Skipcond 800

jump endif11

then11, load side3

Subt hypothenuse

Store side3

load cosResult

add one

store cosResult

jump if11

endif11, load cosResult

output

//tan

//divide side 3 which is the opposite side by side1 which is the adjacent

if12, load side1

Skipcond 800

jump endif12

then12, load side1

Subt side3

Store side1

load tanResult

add one

store tanResult

jump if12

endif12, load tanResult

output

jump end

anglee3, clear

//sin

load one

output

//cos

load zero

output

//tan

load zero

Output

jump end

//when the hypothenuse is side 3

case3, clear

//the program will detect which angle is chosen

Input

Store ang

subt one

Skipcond 400

Jump testAngleee1

Jump angleee1

testAngleee1, Load ang

subt two

Skipcond 400

Jump testAngleee2

Jump angleee2

testAngleee2, Load ang

subt three

Skipcond 400

Jump end

Jump angleee3

//sin

//divide side 3 which is the opposite side by the hypotenus

angleee1,clear

//sin

load one

output

//cos

load zero

output

//tan

load zero

Output

jump end

angleee2, clear

//sin

if13, load side1

Skipcond 800

jump endif13

then13, load side1

Subt hypothenuse

Store side1

load sinResult

add one

store sinResult

jump if13

endif13, load sinResult

output

//cos

//divide side 2 which is the adjacent side by the hypotenuse

if14, load side2

Skipcond 800

jump endif13

then14, load side2

Subt hypothenuse

Store side2

load cosResult

add one

store cosResult

jump if14

endif14, load cosResult

output

//tan

//divide side 1 which is the opposite side by side2 which is the adjacent

if15, load side2

Skipcond 800

jump endif15

then15, load side2

Subt side1

Store side2

load tanResult

add one

store tanResult

jump if15

endif15, load tanResult

output

jump end

angleee3, clear

if16, load side2

Skipcond 800

jump endif16

then16, load side2

Subt hypothenuse

Store side2

load sinResult

add one

store sinResult

jump if16

endif16, load sinResult

output

//cos

//divide side 1 which is the adjacent side by the hypotenuse

if17, load side1

Skipcond 800

jump endif17

then17, load side1

Subt hypothenuse

Store side1

load cosResult

add one

store cosResult

jump if17

endif17, load cosResult

output

//tan

//divide side 2 which is the opposite side by side1 which is the adjacent

if18, load side2

Skipcond 800

jump endif18

then18, load side2

Subt side1

Store side2

load tanResult

add one

store tanResult

jump if18

endif18, load tanResult

output

jump end

end, Halt

a, dec 0

b, dec 0

z, dec 0

Zero, DEC 0

zero, DEC 0

one, dec 1

One, DEC 1

two, DEC 2

three, DEC 3

four, DEC 4

five, DEC 5

six, DEC 6

seven, DEC 7

eight, DEC 8

nine, DEC 9

ten, DEC 10

o, DEC 0

x, DEC 0

y, DEC 0

w, DEC 0

multResult,DEC 0

sumResult,DEC 0

d,DEC 0

c,DEC 0

t,DEC 0

count,DEC 0

hypothenuse, DEC 0

side1, DEC 0

side2, DEC 0

side3, DEC 0

hundredEighty, DEC 180

totalAngles, DEC 180

tanResult,DEC 0

sinResult,DEC 0

cosResult,DEC 0

ang,DEC 0

Sum, DEC 0

MulOpA, DEC 0

MulOpB, dec 0

MulSum, DEC 0

Fact, dec 0

p, dec 0

s, dec 0

e, dec 0

f, dec 0

subResult, dec 0

Count, dec 0

i, dec 0

Ans, dec 0

**User instructions**

Notes : - The program **can** manipulate negative numbers

* The program **cannot** deal with decimals, so do not use decimal numbers if you want accurate results

1. **Addition**

If you want to add two numbers, you have to run the program then enter ‘1’ as input. You then enter the first number and the second number, and you will get their sum in the output.

1. **Subtraction**

If you want to subtract two numbers, you have to run the program then enter ‘2’ as input. You then enter the first number and the second number, and you will get their difference in the output.

1. **Multiplication**

If you want to multiply two numbers, you have to run the program then enter ‘3’ as input. You then enter the first number and the second number, and you will get their product in the output.

1. **Division**

If you want to divide two numbers, you have to run the program then enter ‘4’ as input. You then enter the first number and the second number, and you will get the result in the output. Make sure your denominator is not 0.

1. **Power**

If you want to get the power of a number, you have to run the program then enter ‘5’ as input. You first have to enter the exponent and then the number, and you will get the result in the output.

1. **Factorial**

If you want to get the factorial of a number, you have to run the program then enter ‘6’ as input. You then enter the number you want its factorial, and you will get the result in the output.

1. **Max, Min**

If you want to know the maximum and minimum between two numbers. You have to run the program then enter ‘7’ as input. You then enter the first number and the second number, and the program will first print the bigger number and it will print the smallest one.

1. **Triangle Detection**

If you want to check whether the shape you have is a triangle or not. You have to run the program then enter ‘9’ as input. You then enter the three angles you have, the program will check their sum. If it is 180 then the shape is a triangle, otherwise it’s not a triangle.

In case it **is a triangle** you will get **1** as output.

In case if **it’s not**, you will get **0** as output.

1. **Trigonometry**

If you want to calculate the sin, cos or tan of any angle in your triangle, you have to run the program then enter ‘10’ as input.

First you will enter the value of the 1st side of the triangle, then the value of the 2nd side of your triangle. After this you will get **1** as an output, the means the program named the angle between side 1 and side 2**, angle 1**. Then you will enter the value of the 3rd side of the triangle you have. You will get **2** as output, this means the program named the angle between side 2 and side 3, **angle 2**. And then you will get **3** as output, this means the program named the angle between side 3 and side 1, **angle 3**.

After that the program will print the largest side between the three sides you have entered, so you make sure that this is the hypotenuse of the triangle.

**If you want the sin, cos and tan of angle 1 : enter 1**

**If you want the sin, cos and tan of angle 2 : enter 2**

**If you want the sin, cos and tan of angle 3 : enter 3**

After you enter 1, 2 or 3 you will get the sin, cos and tan of the angle you chose.

The first output is the sin, the second is the cos and the third is the tan.

Thank you for using our calculator 😊