MIDDLE EAST TECHNICAL UNIVERSITY

NORTHERN CYPRUS CAMPUS

DEPARTMENT OF COMPUTER ENGINEERING

CNG331
TERM PROJECT: PART 2

NAME: MERT CAN
SURNAME: BİLGİN
STUDENT ID: 2453025

Register File

Register File is implemented by using 8 registers, 2 MUXs, and 1 Decoder. Enable is not included because WE signal directly comes from the Control Unit.

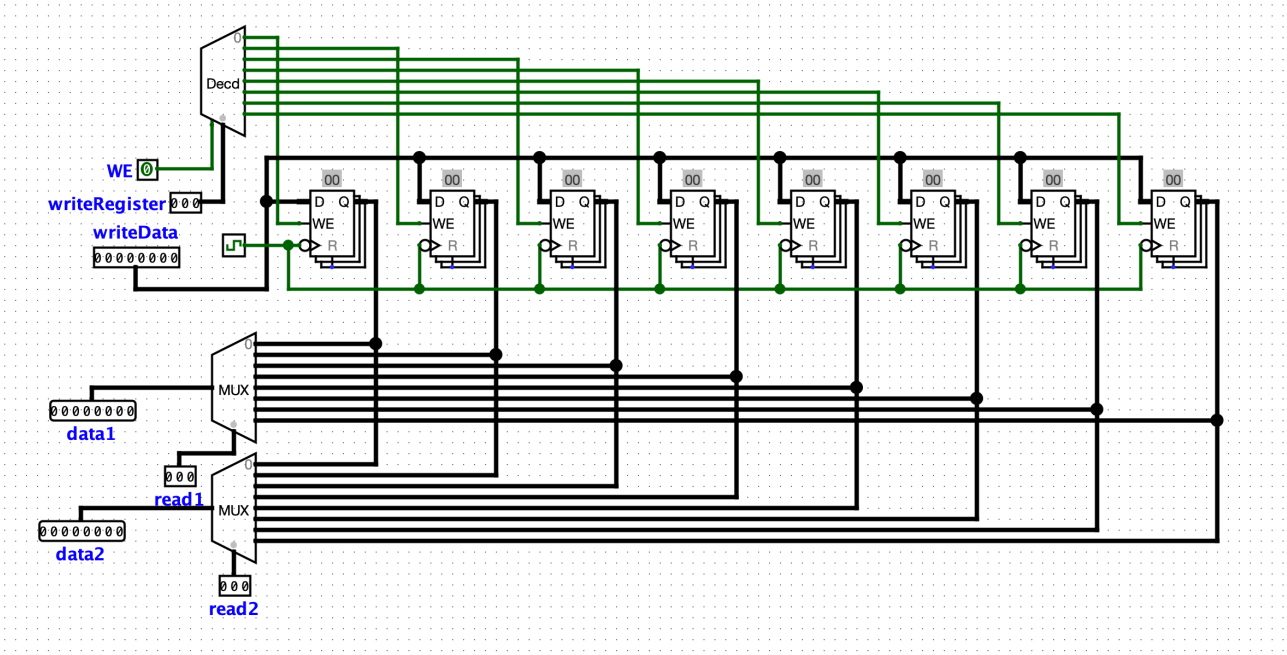


Figure 1: Register File

Data Memory

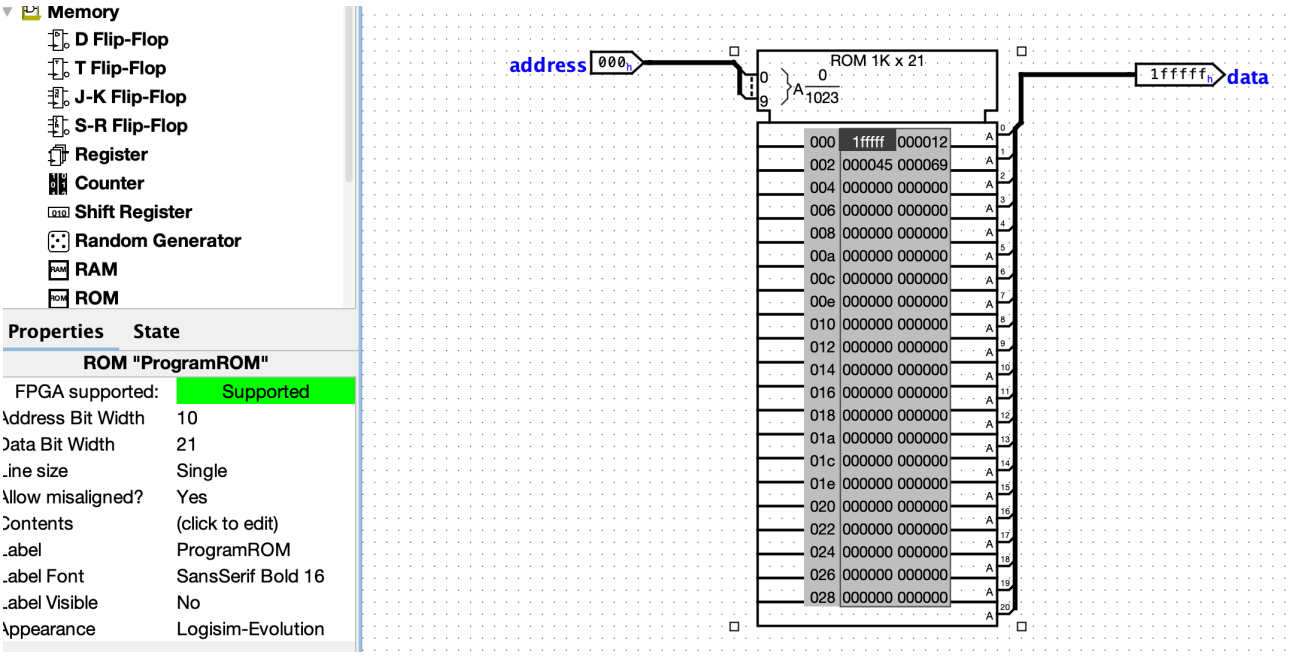


Figure 2: ROM

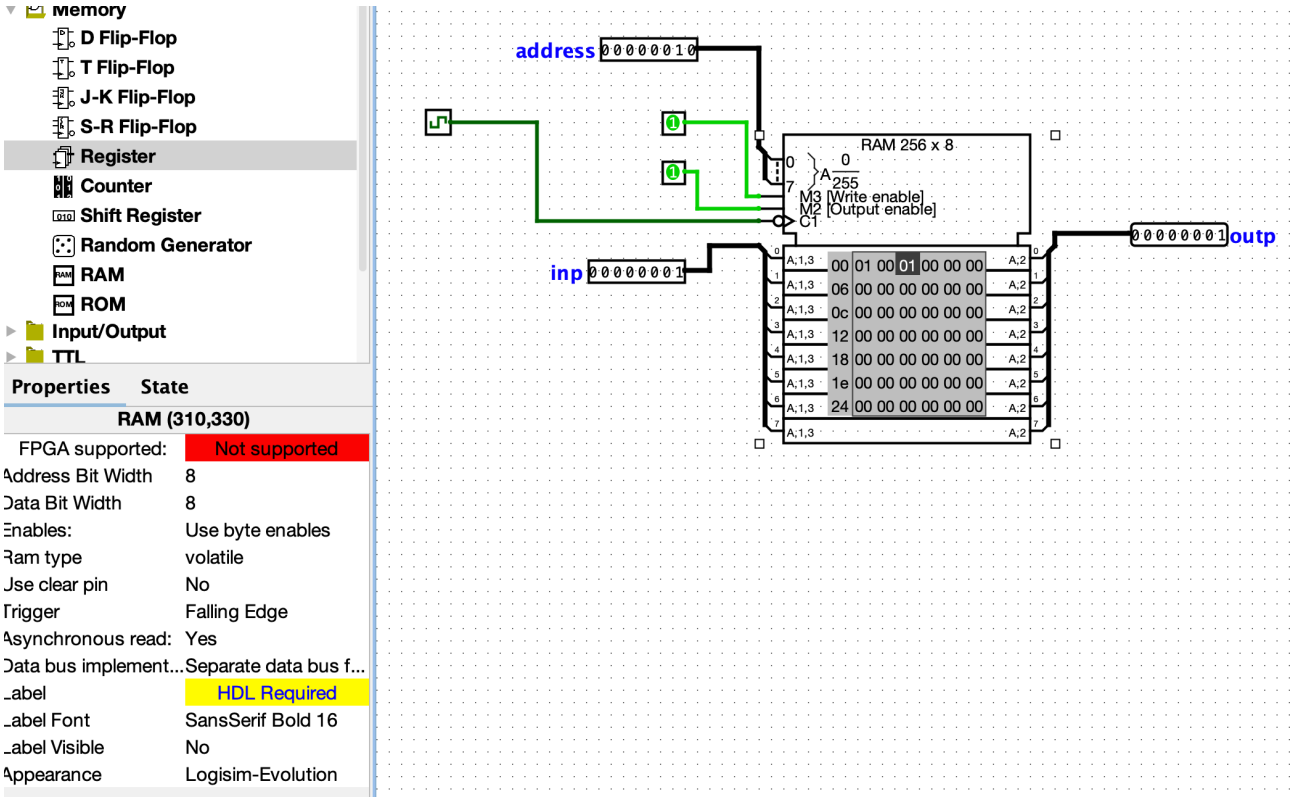


Figure 3: RAM

RAM and ROM are types of computer memory. RAM stands for random access memory and ROM stands for read-only memory. RAM is a type of volatile memory, which means it requires power to retain data. RAM is used for storing data that the computer is currently using or processing. When the computer is powered off, the data in RAM is erased. ROM, on the other hand, is a type of non-volatile memory, which means it retains data even when the power is turned off. ROM is used to store data that the computer needs to boot up and start running. Because it is read-only, data stored in ROM cannot be easily modified or deleted. In short, the main difference between the two types of memory is that RAM is temporary and ROM is permanent. RAM is used for storing data that the computer is currently working with, while ROM is used for storing data that the computer needs to start up and run its basic functions.

Control Unit

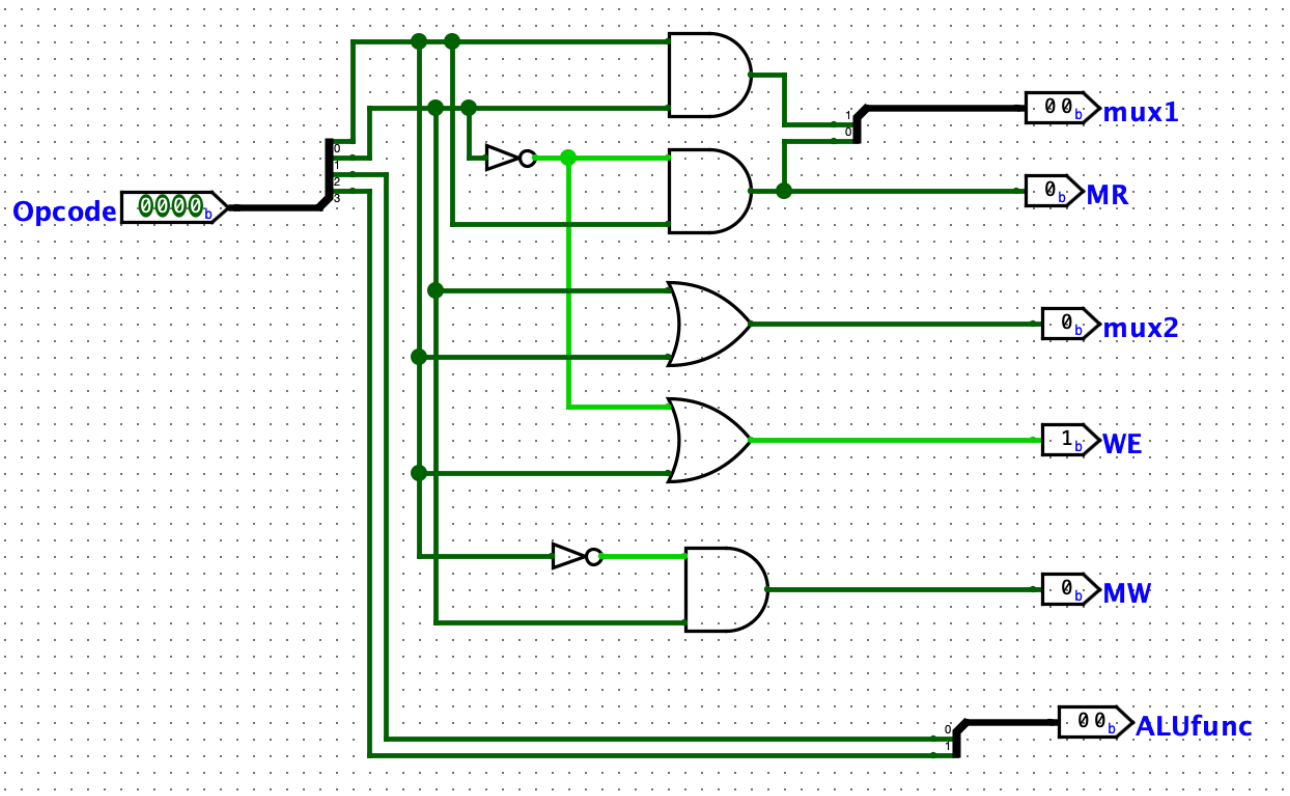
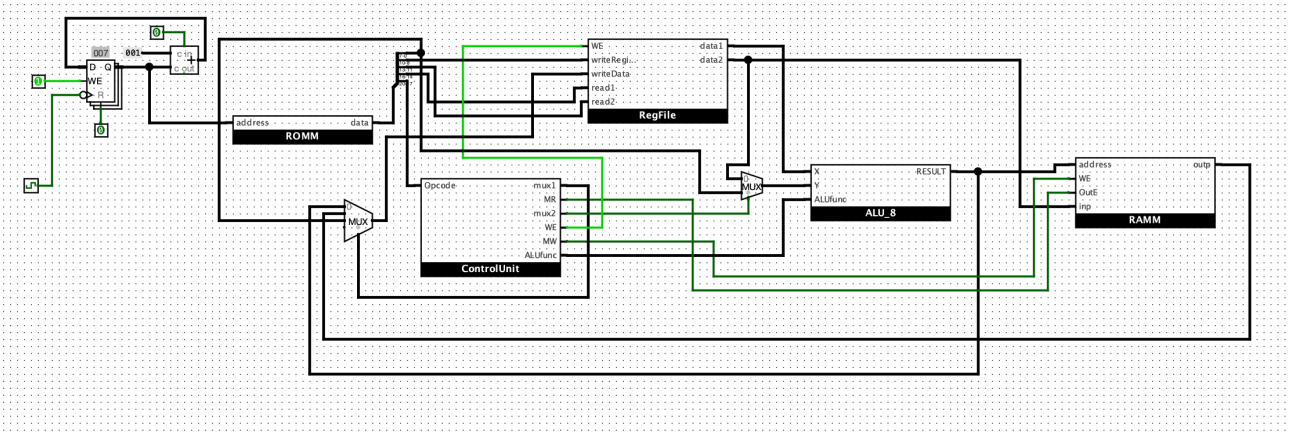


Figure 4: Control Unit

CPU



Assembly Code:

LI $001, IMM(0000 1000)
LI $010, IMM(0000 0100)
AND $011, $010, $001
ADD $100, $010, $001
ROTL $101, $010, $001
ST $010, IMM($001)
LD $110, IMM($001)

1st step: Load the image to ROM.

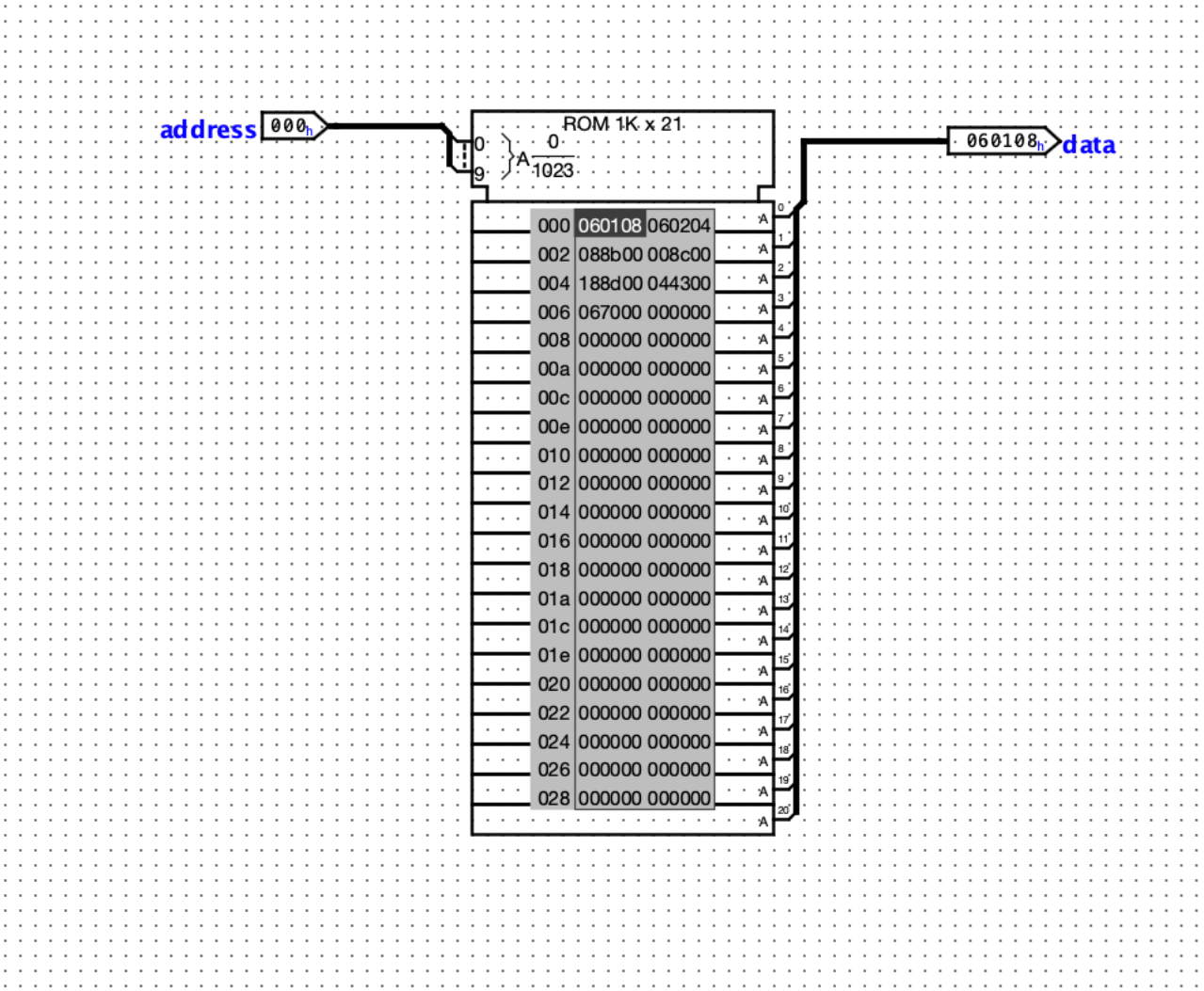


Figure 6: Loaded ROM

After executing the first line, register1 is loaded as 8. The register file is as following:

2nd line: Register2 is loaded as 4.

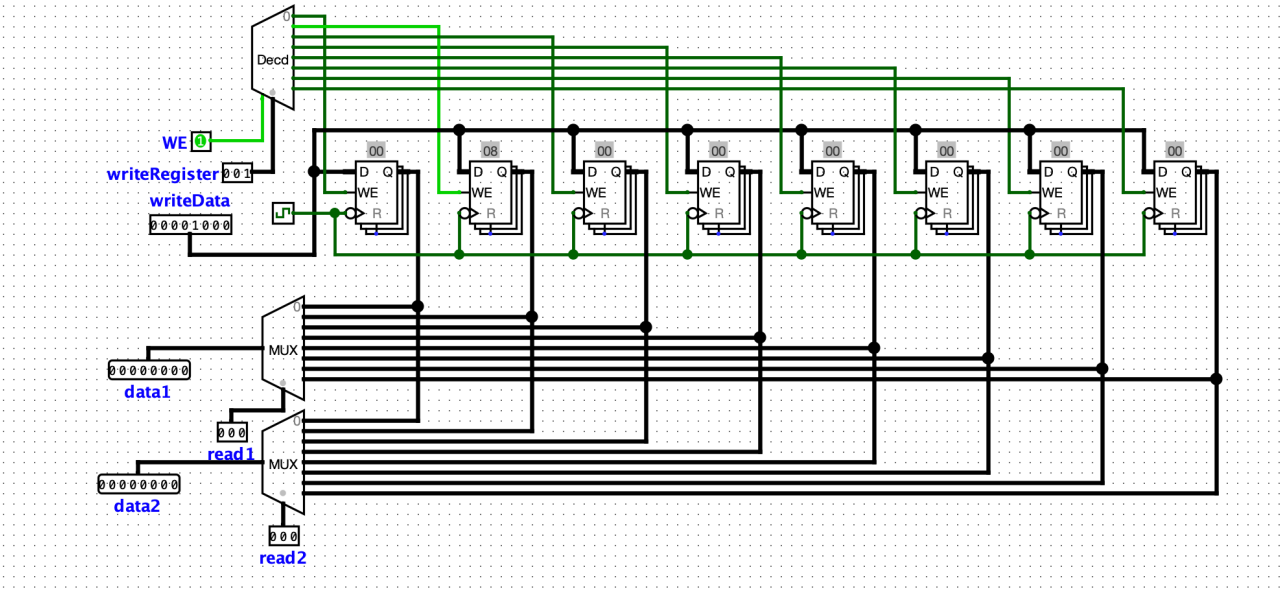


Figure 7

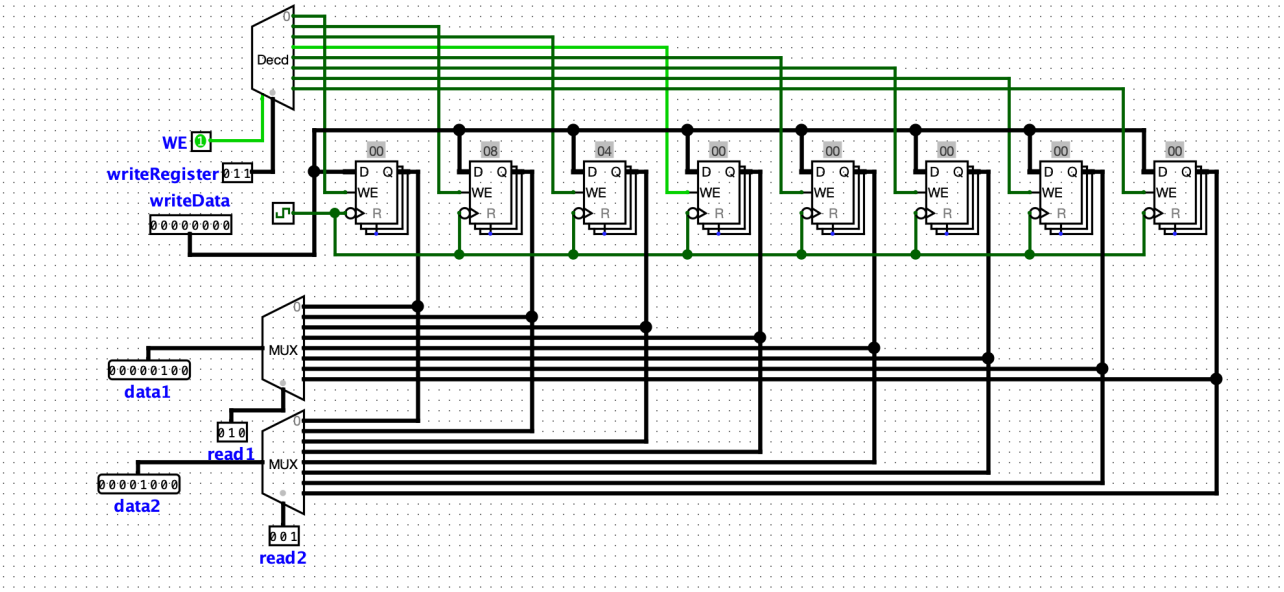


Figure 8

3rd line: 0000 1000 & 0000 0100 operation is executed and loaded to register3.

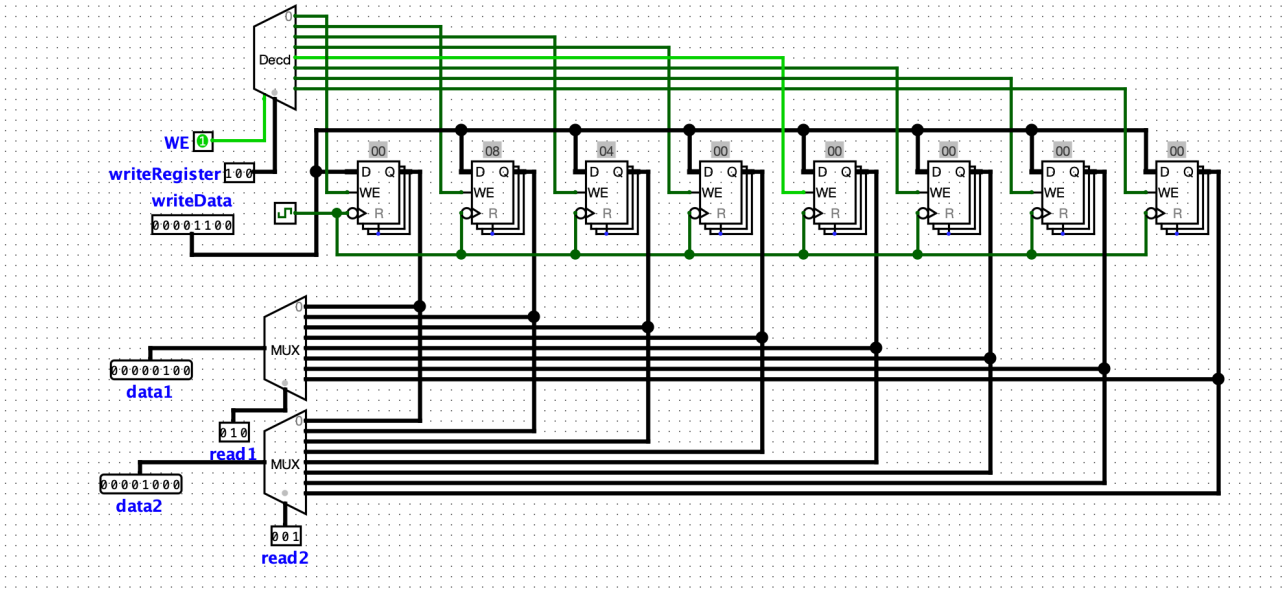


Figure 9

4th line: 0000 1000 + 0000 0100 operation is executed and loaded to register4

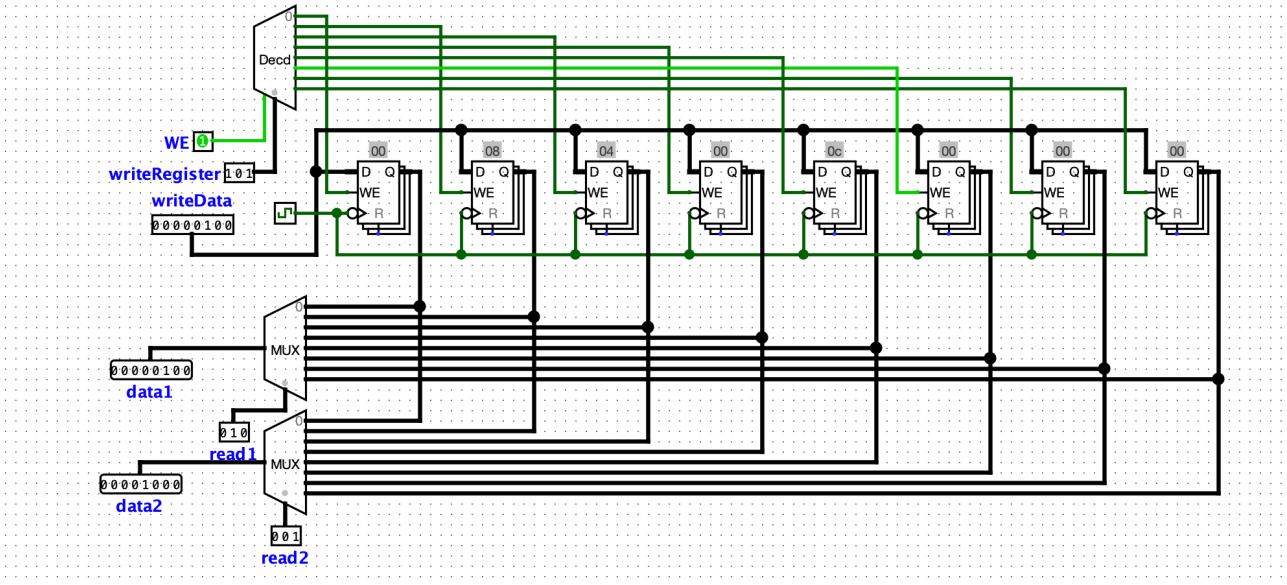


Figure 10

5th line: 0000 0100 is rotated by 000, and loaded to register5.

6th line: store operation is executed

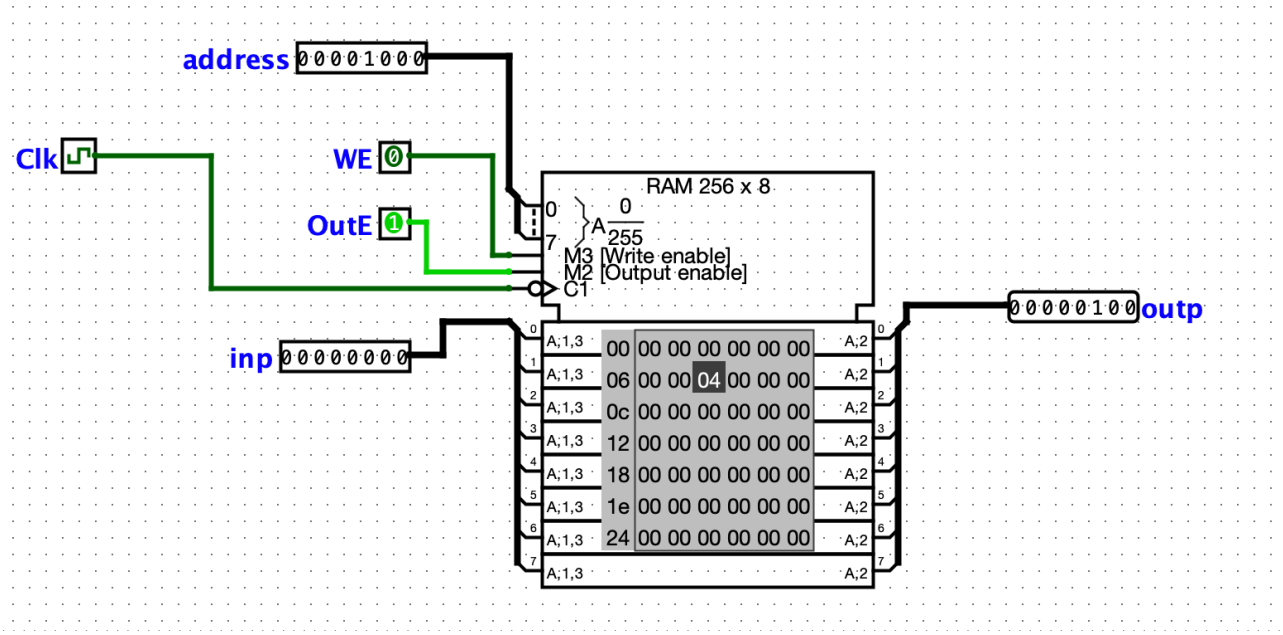


Figure 12

7th line: register7 is loaded as 4

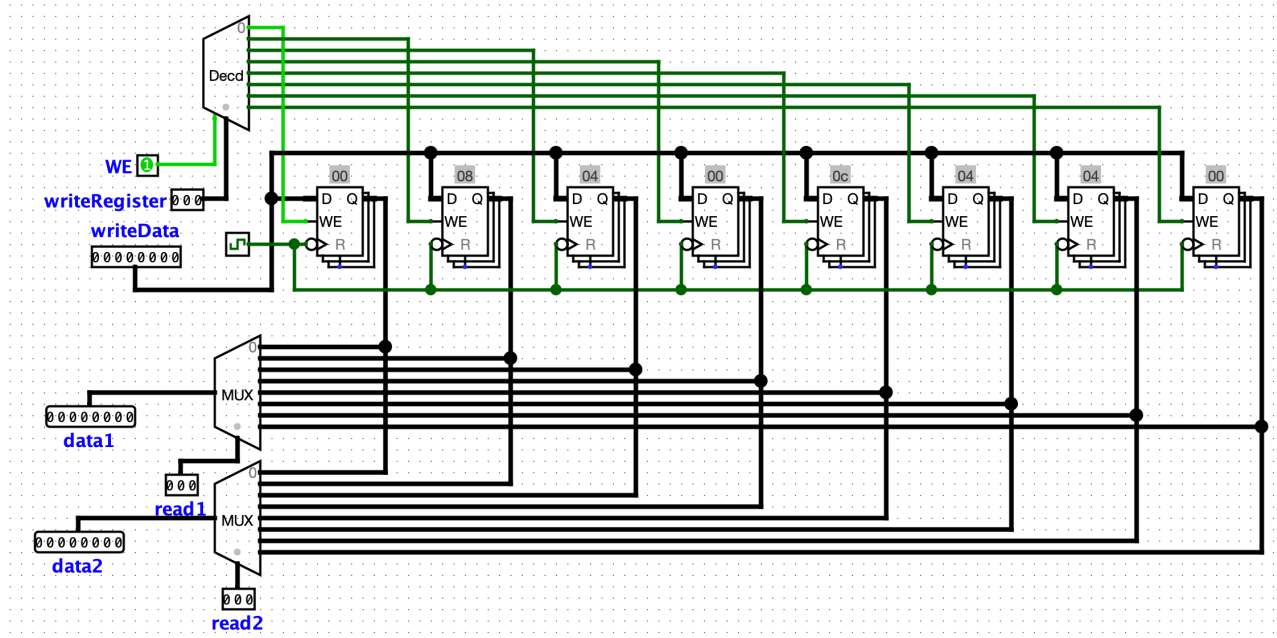


Figure 13