MIDDLE EAST TECHNICAL UNIVERSITY

NORTHERN CYPRUS CAMPUS

DEPARTMENT OF COMPUTER ENGINEERING

CNG331
TERM PROJECT: PART 3

NAME: MERT CAN
SURNAME: BİLGİN
STUDENT ID: 2453025

Firstly, I started to implement my design by modifying control unit. To do that, I used K-maps and got control bits as follows:

- ALUfunc[2] = opcode[3]. opcode[2]. opcode[1]. opcode[0]
- ALUfunc[1] = opcode[3]. Not(opcode[1]). Not(opcode[0])
- ALUfunc[0] = opcode[2]. Not(opcode[1]). Not(opcode[0])
- mux1[1] = Not(opcode[2]). opcode[1]. opcode[0]
- mux1[0] = Not(opcode[1]). opcode[0]
- mux2 = Not(opcode[2]).(opcode[1] + opcode[0])
- WE = Not(opcode[1]). Not(opcode[0]) + Not(opcode[3]).Not(opcode[2]).opcode[0]
  + opcode[3].opcode[2]
- Write = opcode[1]. Not(opcode[0])
- Read = Not(opcode[2]).Not(opcode[1]).opcode[0]
- PCmux[1] = opcode[3].Not(opcode[2]).opcode[1]
- PCmux[0] = Not(opcode[3]).opcode[2].opcode[0]
- Branch = Not(opcode[3]).opcode[2].opcode[1].opcode[0] + Not(opcode[3]). opcode[2]
  .Not(opcode[1]).opcode[0] + opcode[3].Not(opcode[2]).opcode[1].opcode[0]

According to these I completed the Control Unit as in Figure 1.

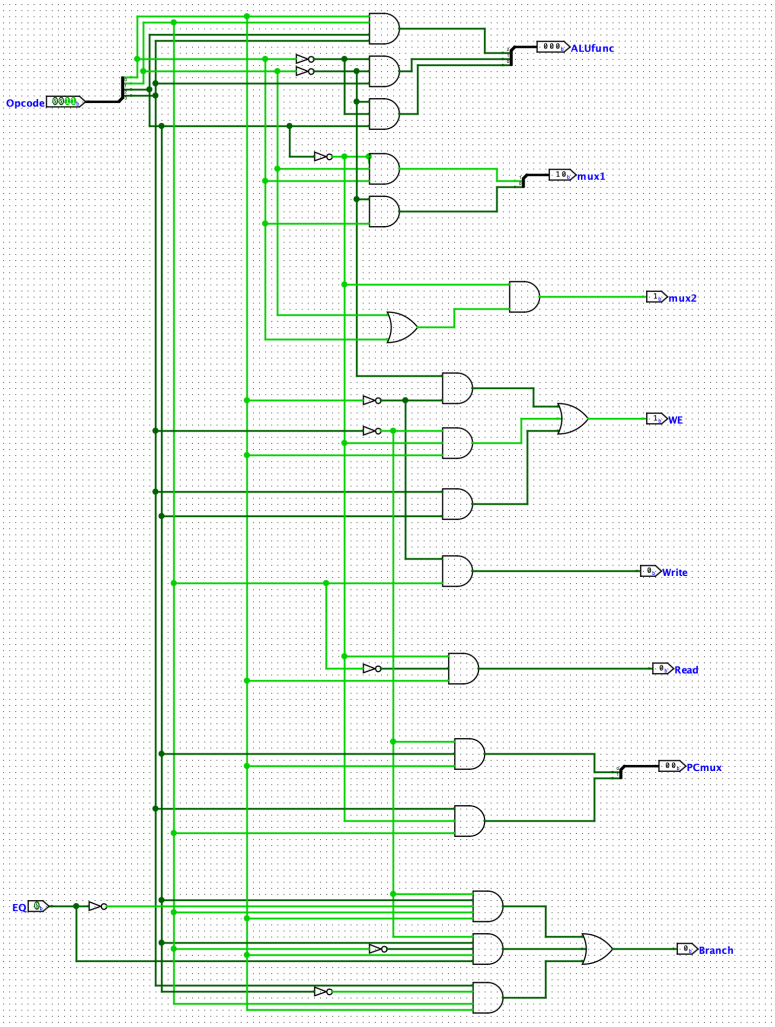


Figure 1: Control Unit

Then, I modified ALU to add SLT function and EQ signal as in Figure 2.

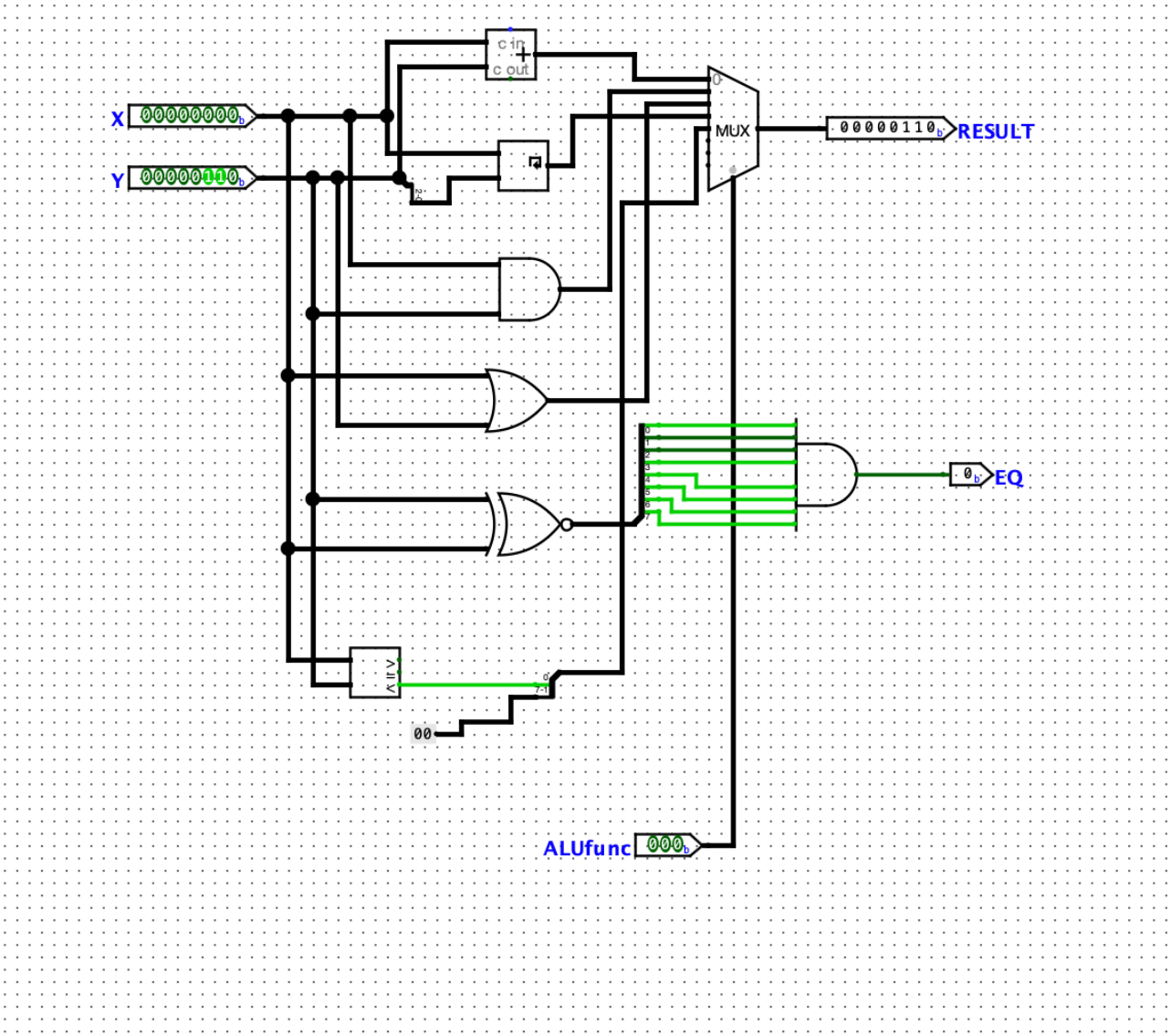
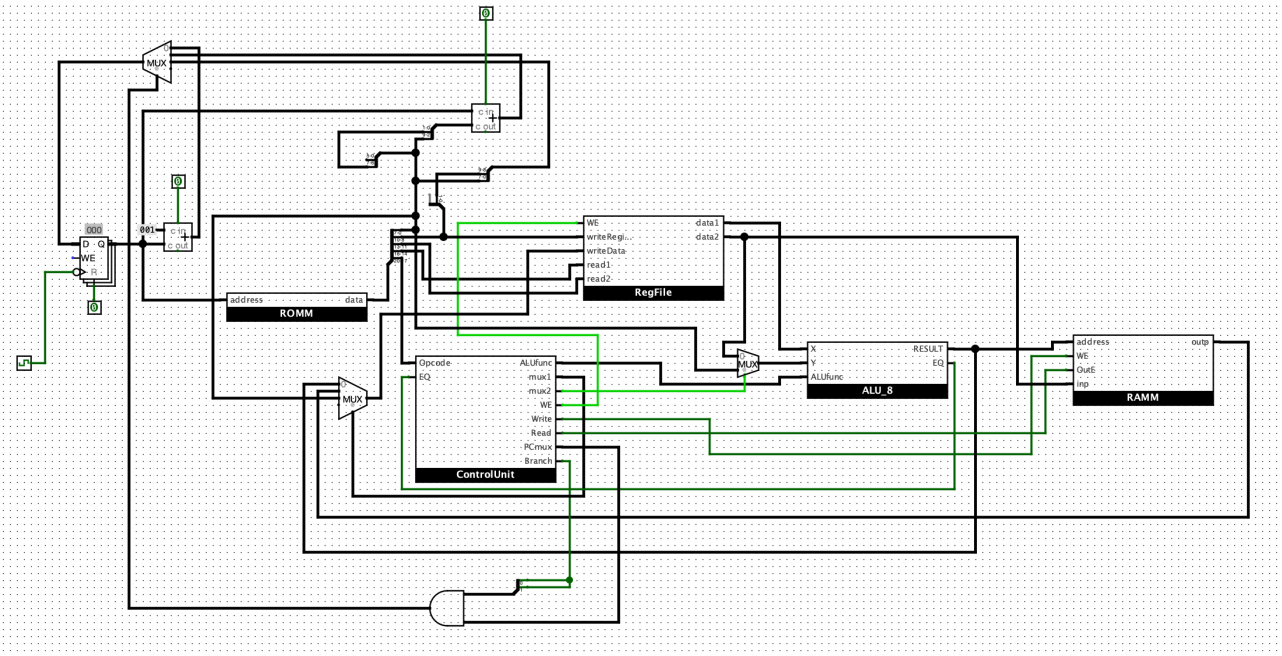


Figure 2: ALU

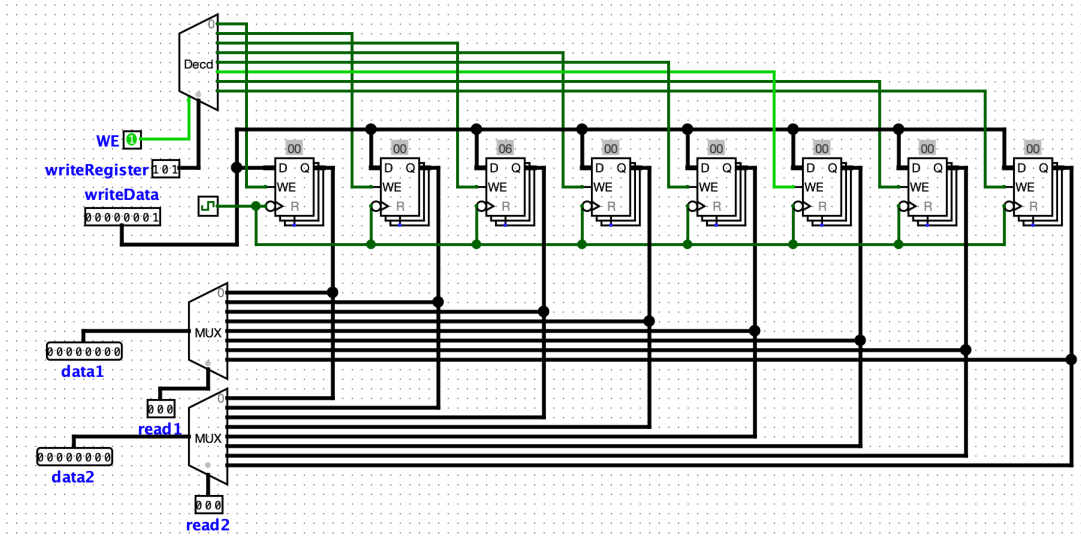
After that, I added a new mux to the circuit, and connected the necessary parts. The completed circuit is in Figure 3.



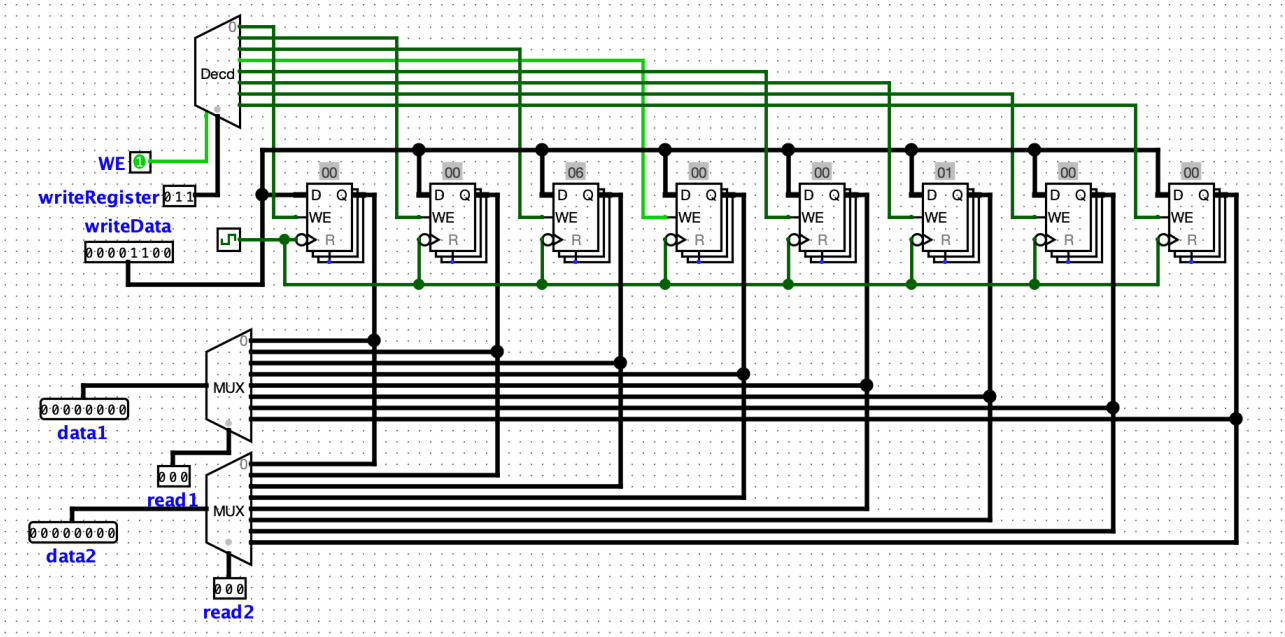
According to the test assembly code, I converted them to hex file as (I assumed that the registers are R1, R2, R3, …, R8):

LI R3, 6 = 060206
LI R6, 1 = 060501
LI R4, 12 = 06030C
ADD R7, R7, R6 = 01AE00
SLT R5, R4, R7 = 1EF400
BEQ R6,R5,L1 = 0B2882 (I assumed that the offset of L1 is 3, so we need to subtract 2)
J END = 160008
ADD R2,R2,R6 = 006900

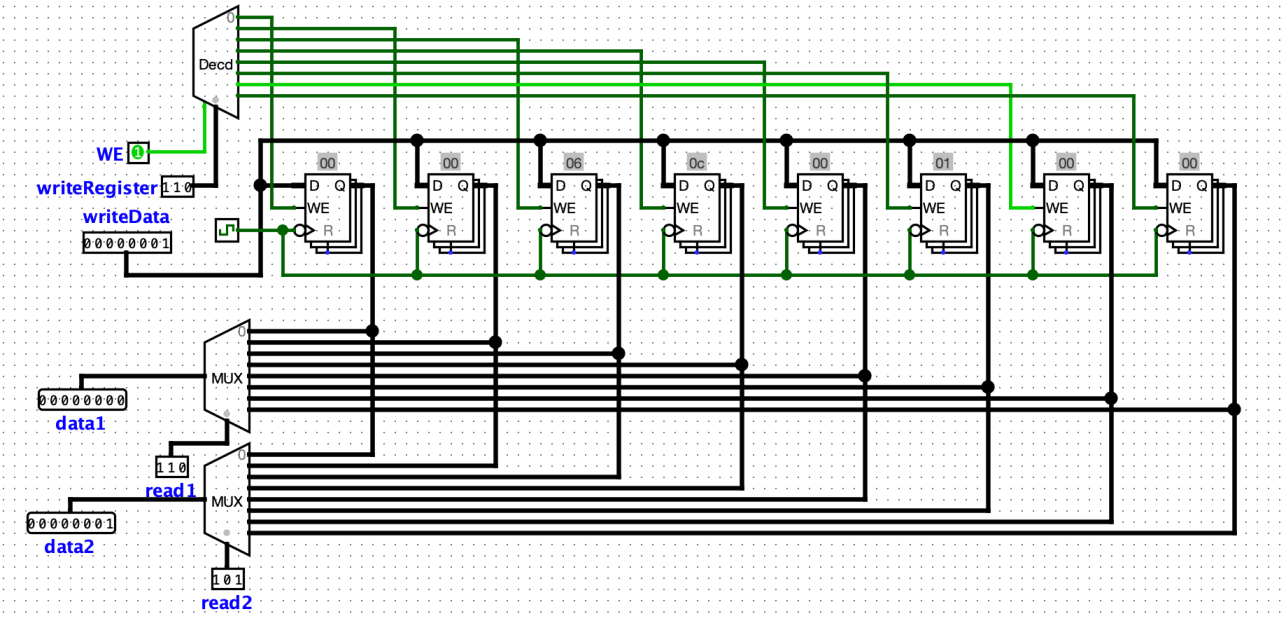
After executing LI R3,6, the registers file:



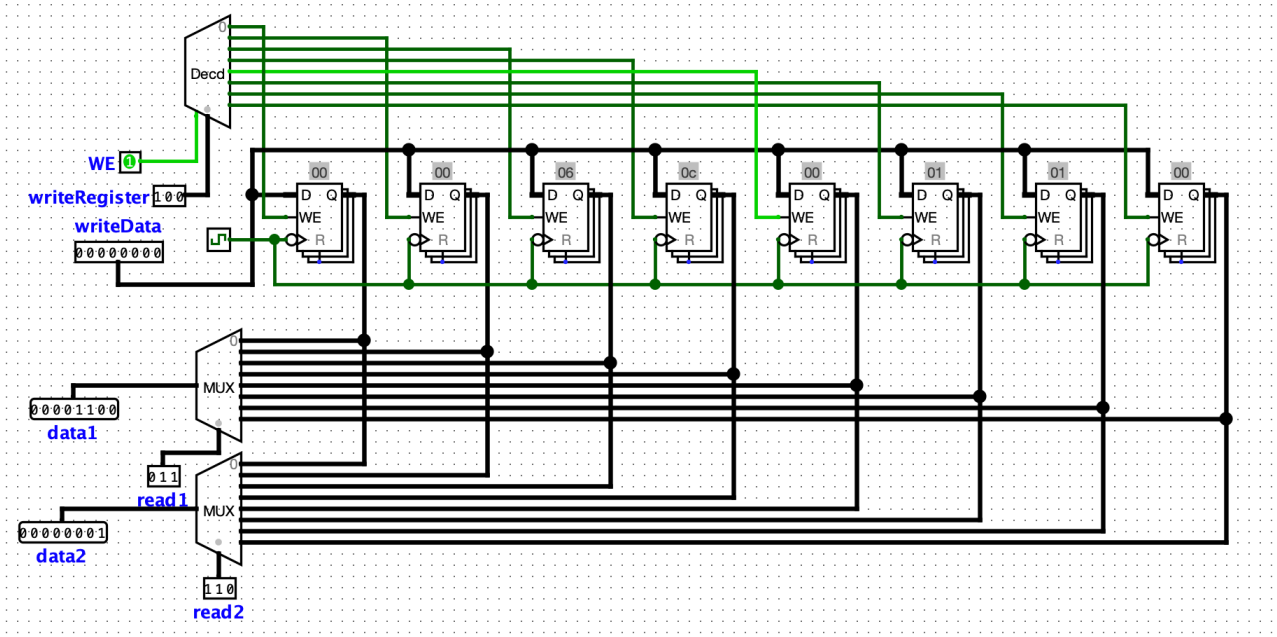
After executing LI R6,1, the registers file:



After executing LI R4,12, the registers file:



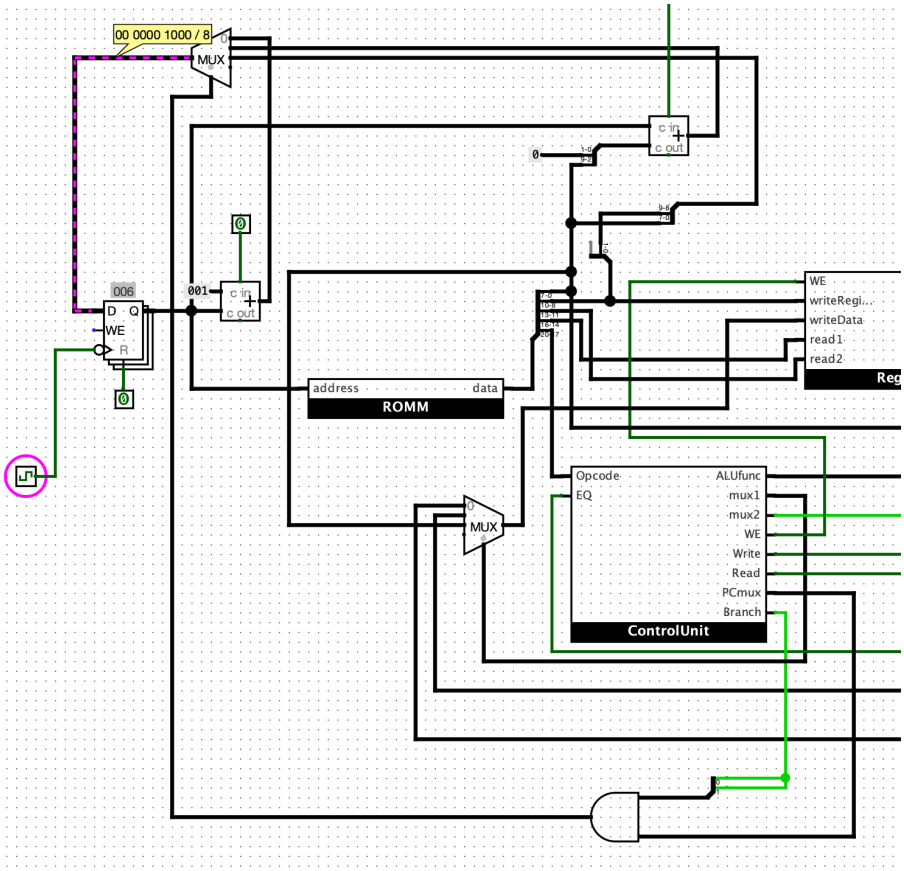
After executing ADD R7, R7, R6, the registers file:



After executing SLT R5, R4, R7, the registers file will not change, because the valueR4 is not less than the value in R7.

After executing BEQ R6, R5, L1, the registers file will not change, and it will not go on from L1 because the value in R6 is not equal to R5, so the next line will be executed.

After executing J END, PC will take the value from IMM in J-type like:



Because we know that there is no instruction in 8th line of ROM, I set the immediate value as 8.

At the end of the instructions, the registers file: