

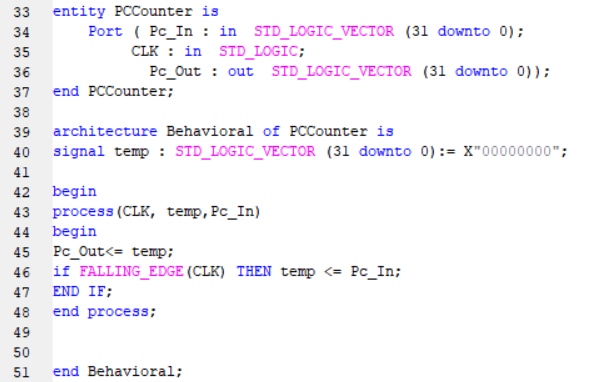
MIPS

**Name: Mahmoud Mostafa Mahmoud Ramadan  
Reg No.: 19102461  
Dr. Ahmed Abo El Farag  
Eng. Esraa Khattab**

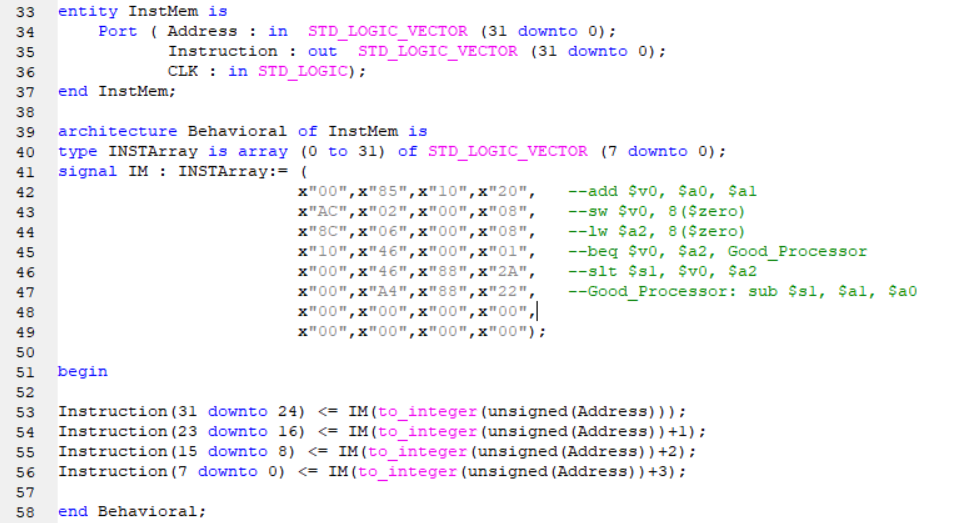
**Sources Needed for Mips**

**1) Program Counter**

VHDL Module:

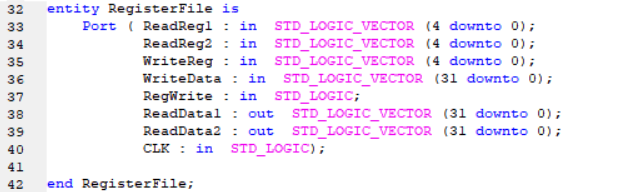


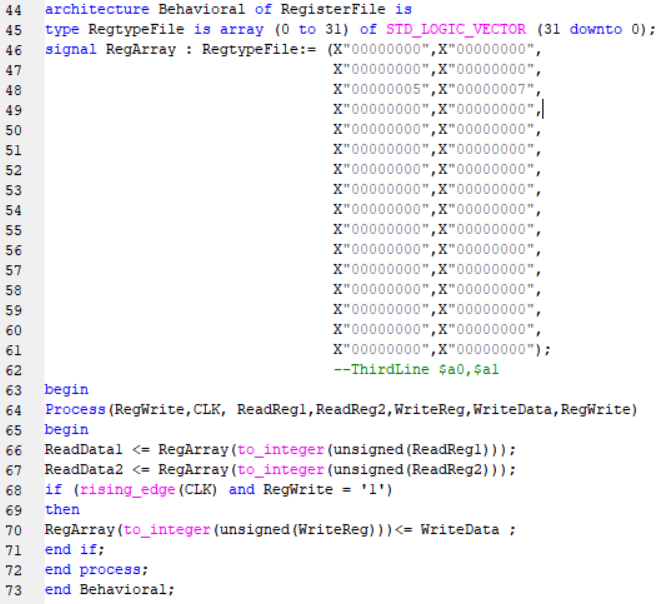
**2) Instruction Memory**

VHDL Module :  


**3) Register File**

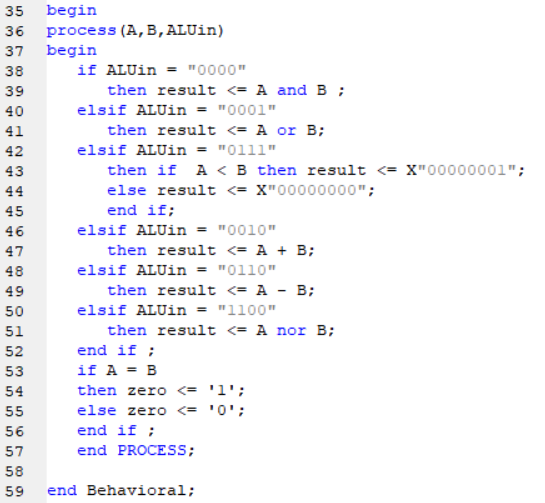
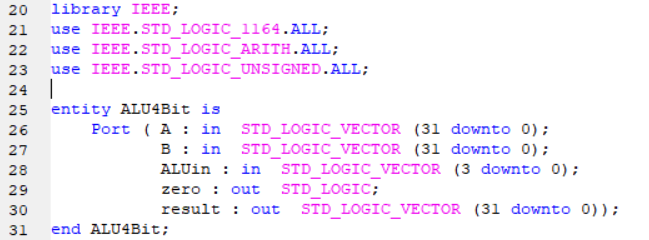
VHDL Module:





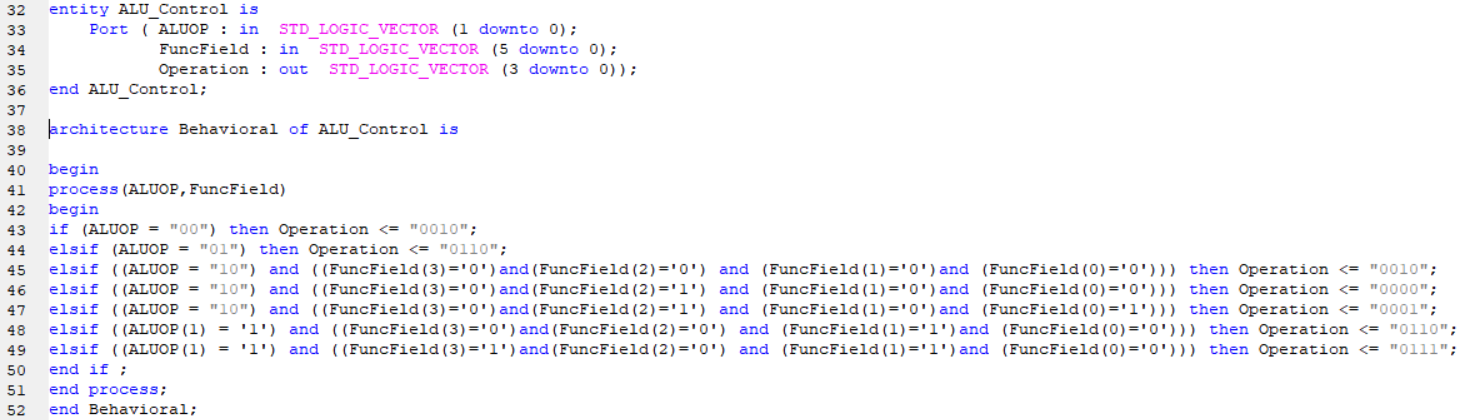
**4) ALU**

VHDL Module:



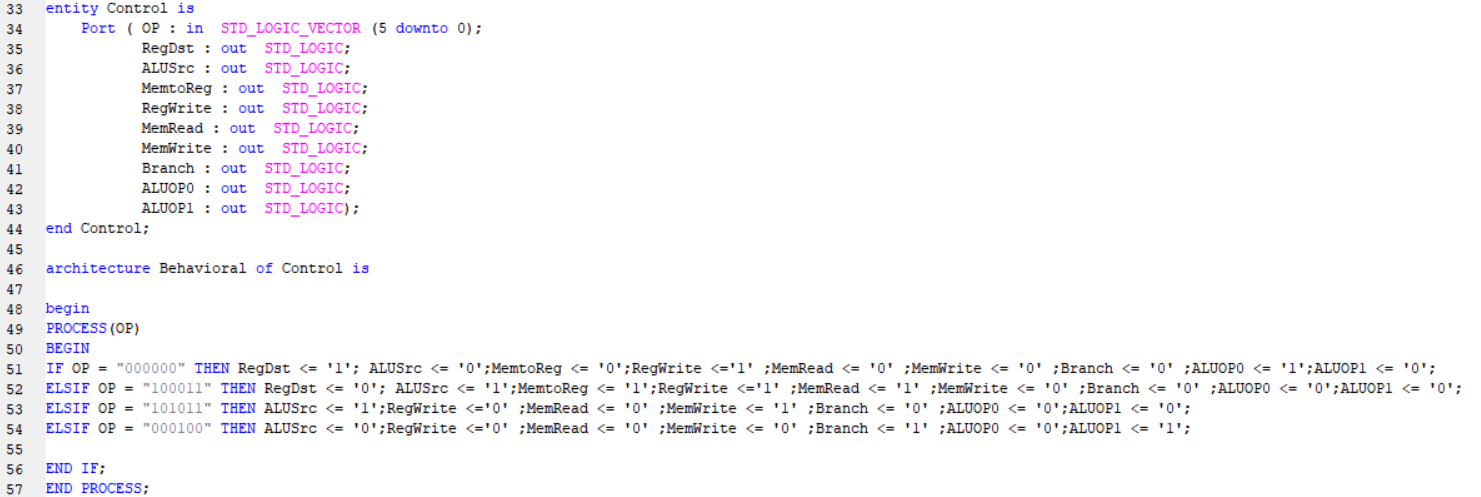
**5) ALU Control**

VHDL Module:



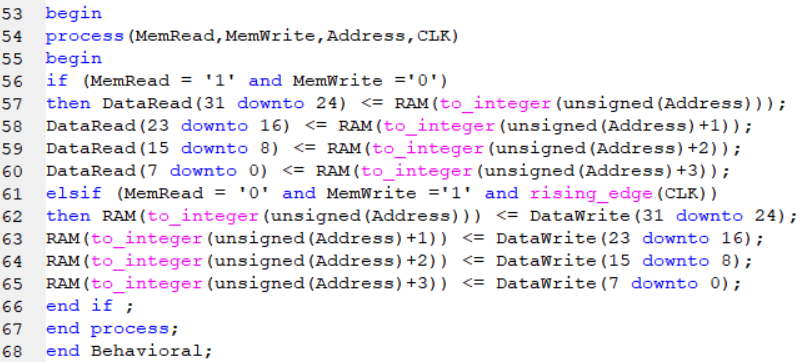
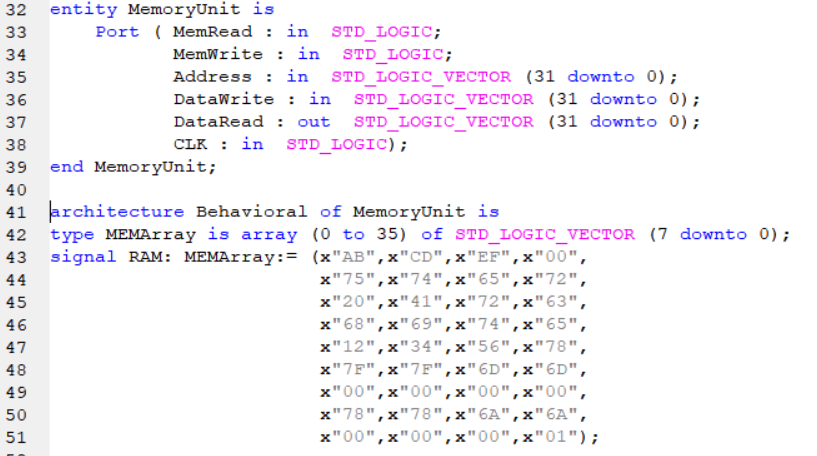
**6) Control Unit**

VHDL Module:

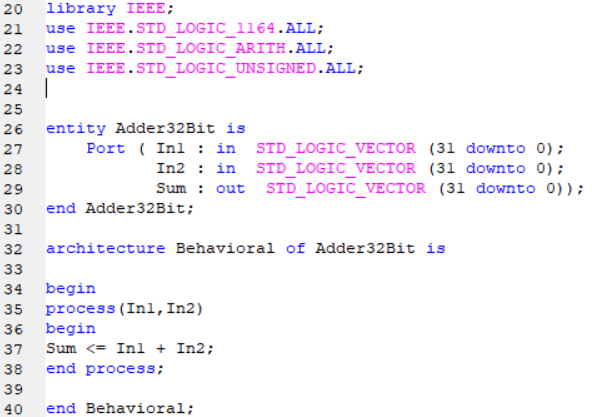


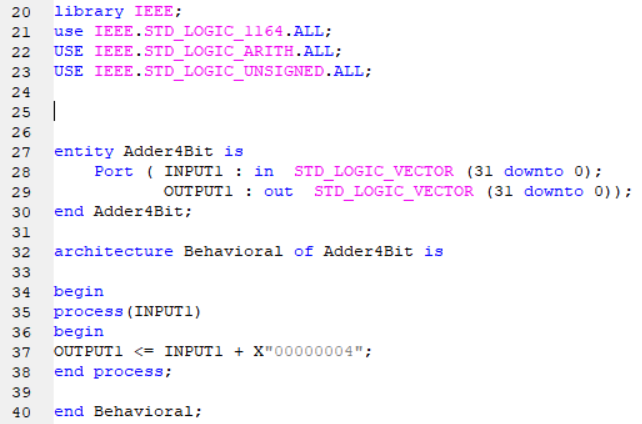
**7) Memory Unit**

VHDL Module:



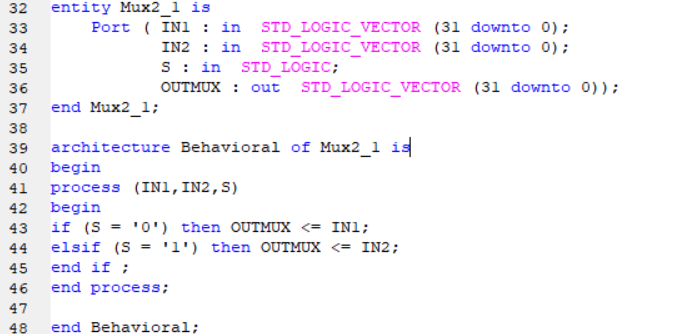
**8) Adders**  
**ADDER for 32 bit inputs**VHDL Module:



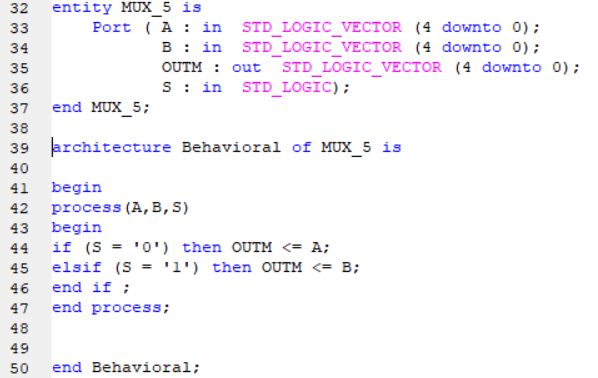
**ADDER for PC+4**   
VHDL Module:

**9) Muxes**

**2-1 32Bit Mux**VHDL Module:

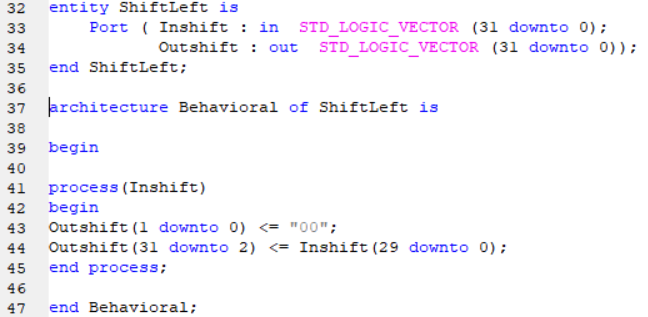


**2-1 5Bit Mux**VHDL Module:

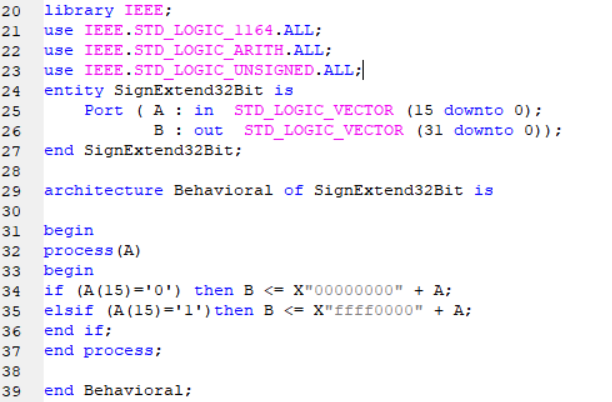


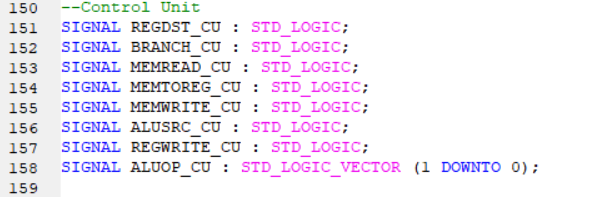
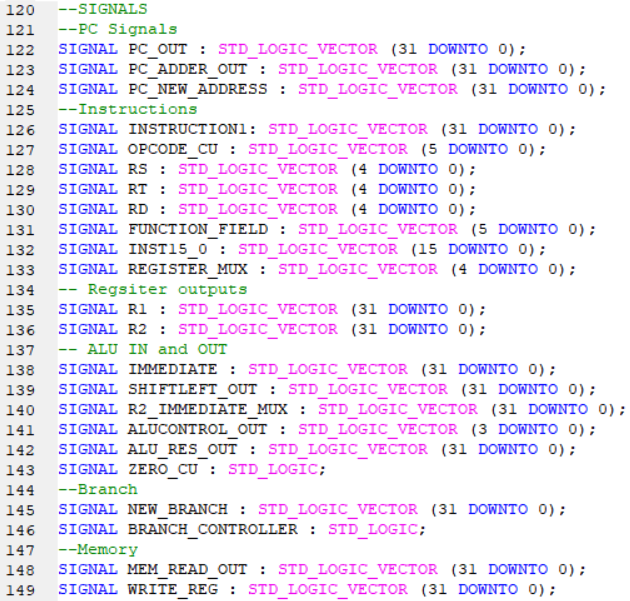
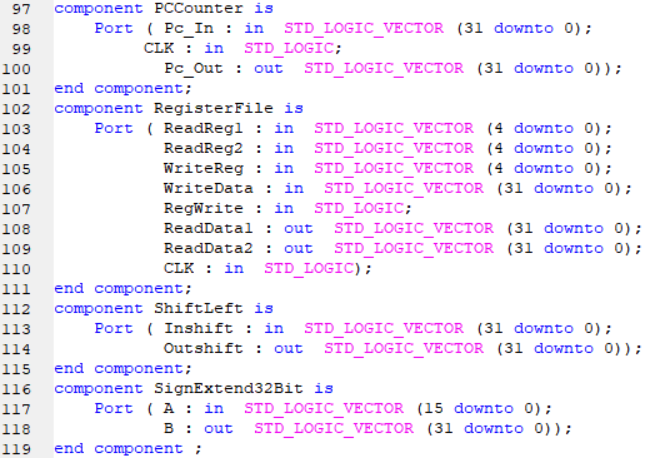
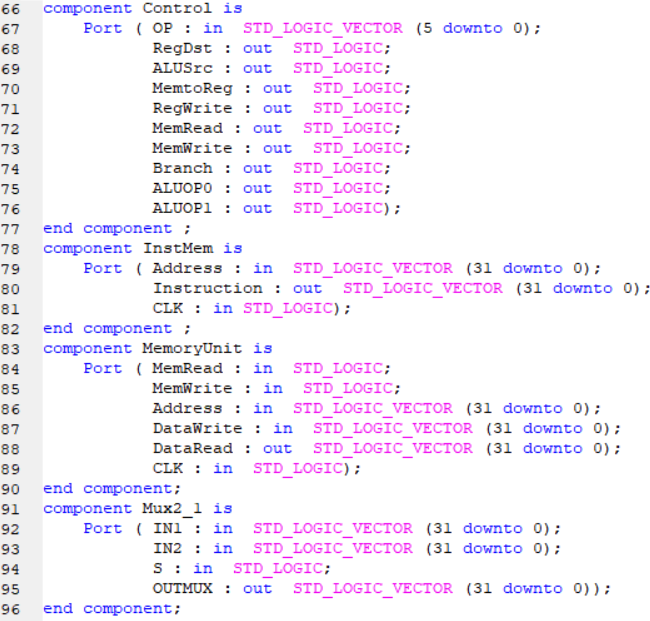
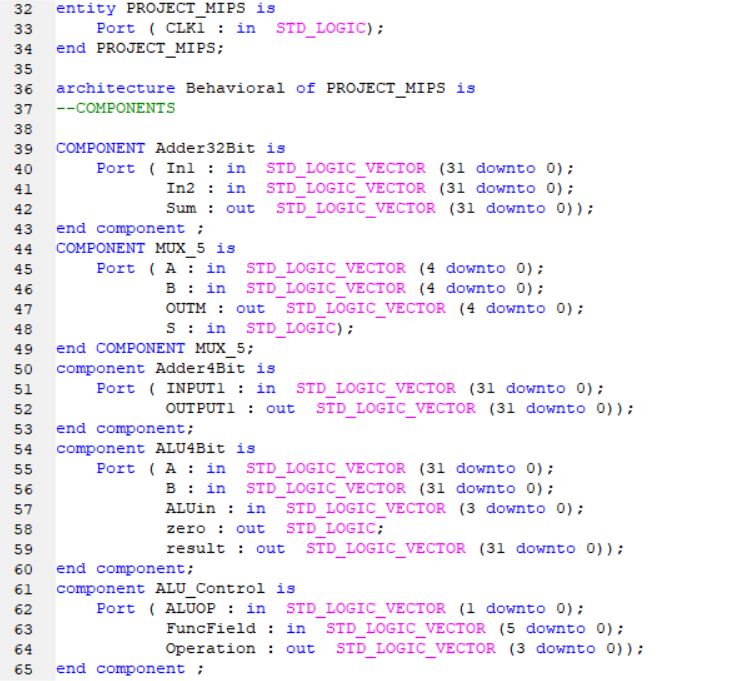
**10) Shift Left**

VHDL Module:

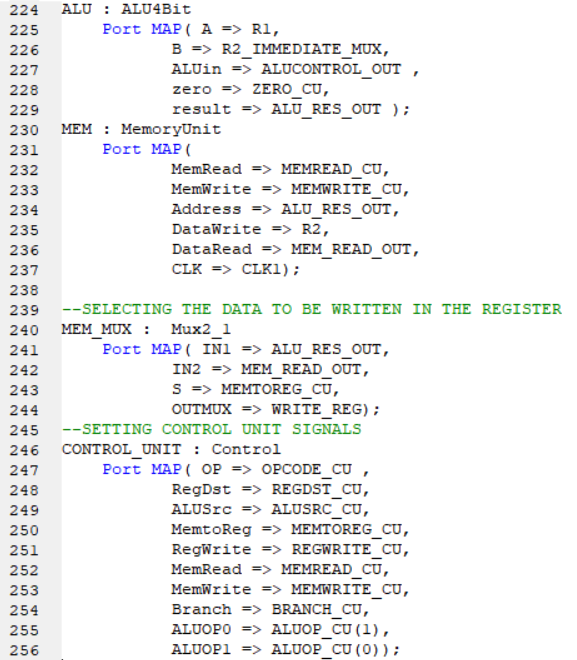
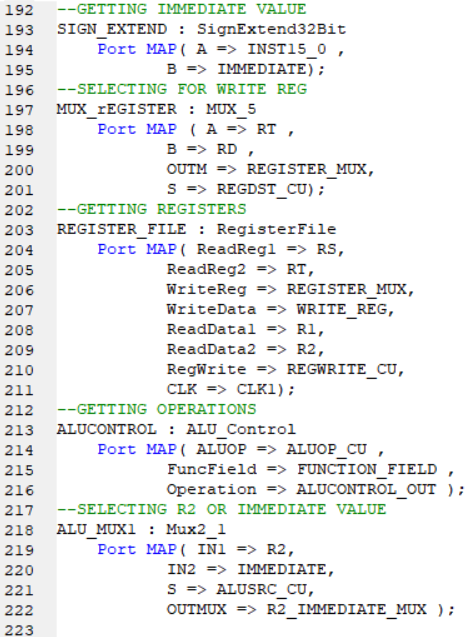
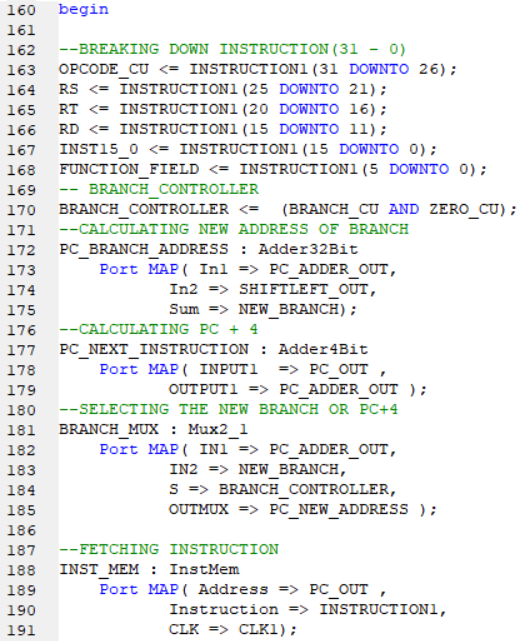
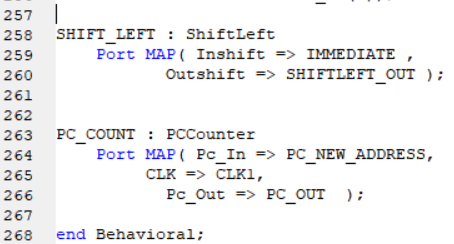
**11) Sign Extend**

VHDL Module:



**The Main Program Port Mapping and Components**  
  


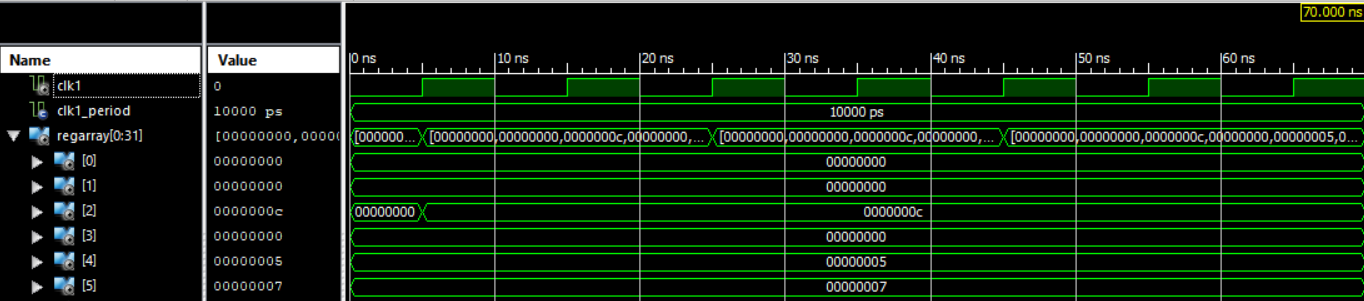
**Port Mapping:**

**Simulation of the Design:**

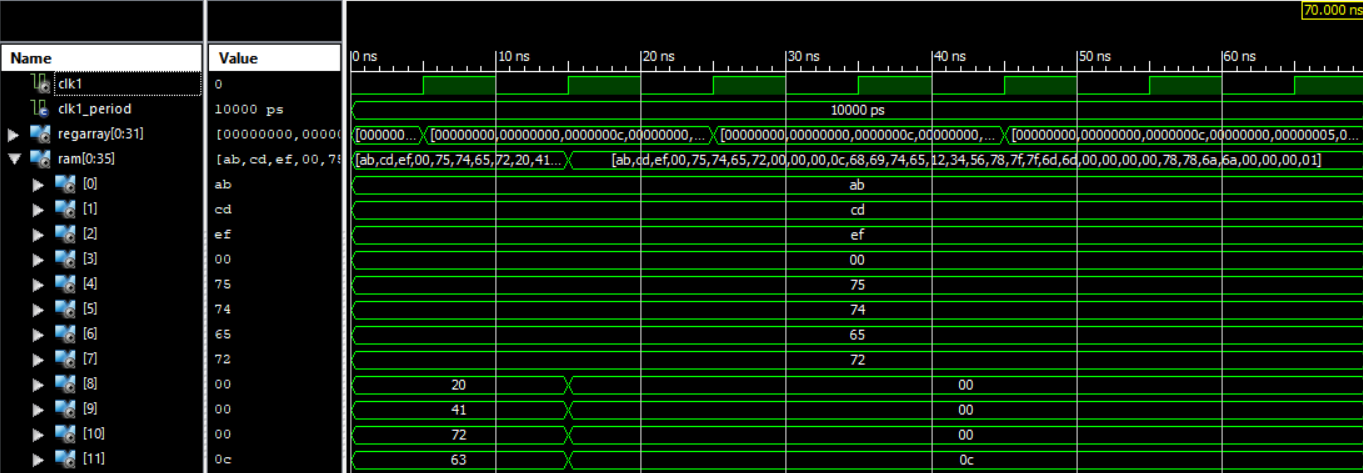
**1- add $v0, $a0, $a1**

This is R-Type Instruction so the operation is the addition of the values in registers $a0(4) and $a1(5) and save the result in $v0(2)   
5+7 = x(c) as the simulation shows the operation is made correctly.



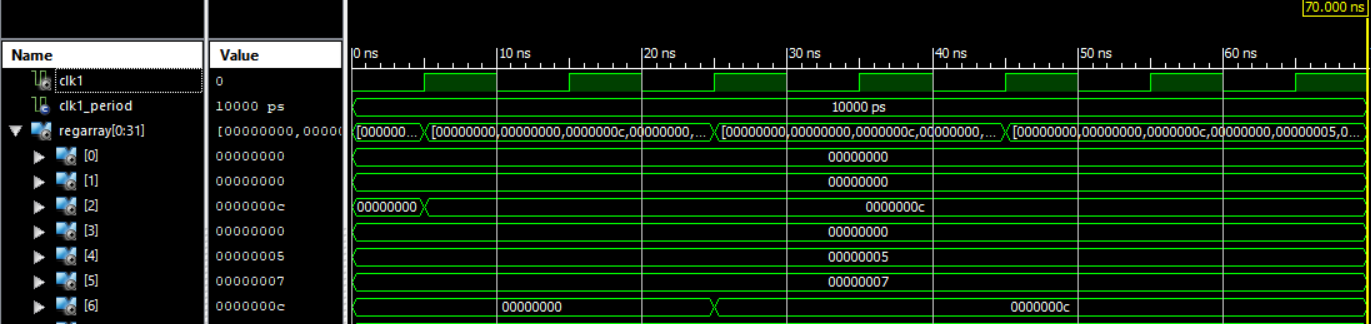
**2- sw $v0, 8($zero)**

This I-Type Instruction were the operation is done by calculating the address to which we will save the data of Register $v0 in the memory by the addition of the offset \*4 with the reference address which is $zero (As we know from the previous instruction the value in $v0 is X”0000000c” from 8to 11)



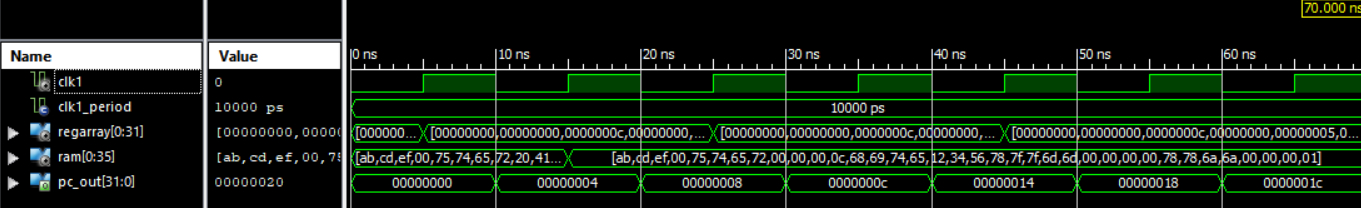
**3- lw $a2, 8($zero)**

This I-Type Instruction were the operation is done by reading the value stored in the memory at Address 8\*4 +$zero and save it in the register $a2(6)

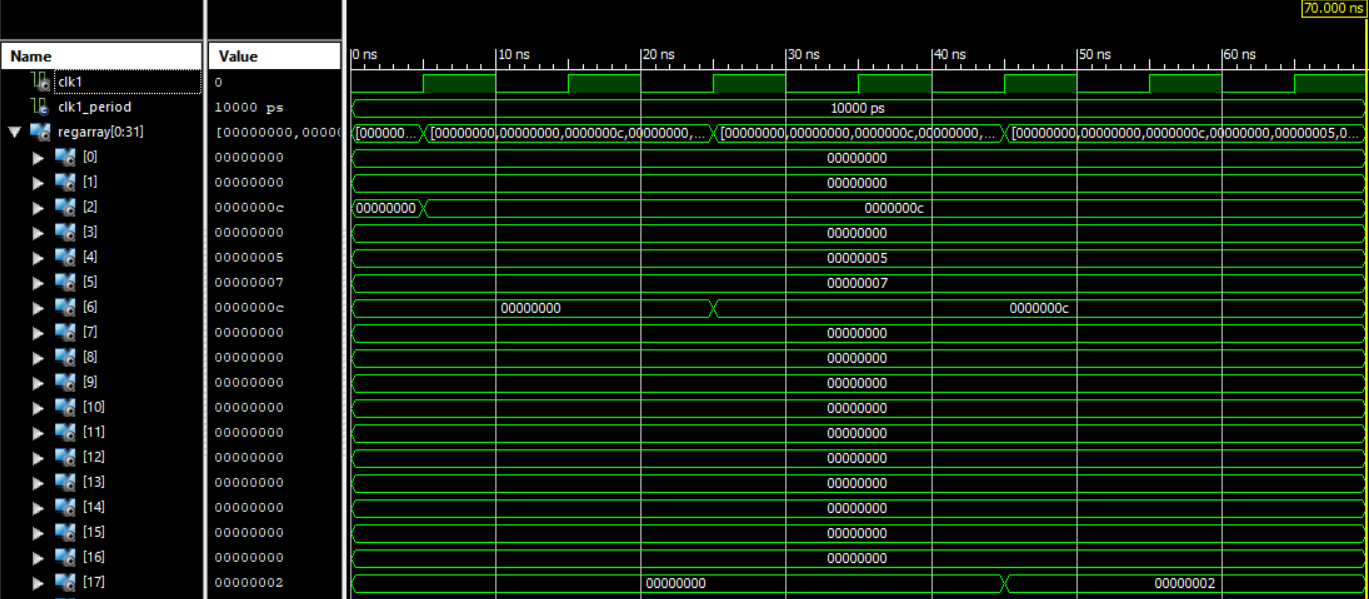


**4- beq $v0, $a2, Good\_Processor**

This I-Type Instruction were the operation is done by subtracting the values stored in $v0(X”0000000c”) and $a2(X”0000000c”) if the result is zero then a new address is calculated by the addition of the immediate value which is sign extended and shifted to the left by 2 bits with the PC+4 signal calculated   
(after the 4 clock the address after c is 10 but Because the condition was true the address changed to 14 which is the place where our label refers)



**5- slt $s1, $v0, $a2**

This is R-Type Instruction so the operation is the comparison of the values in registers $v0(2) and $a2(6) and if the value of $v0 is greater than $a2 then it will set $s1(17) to be equal 1  
this instruction didn’t work because the condition of beq was true and caused the sequence of the program to change 

**6- Good\_Processor: sub $s1, $a1, $a0**

This is R-Type Instruction so the operation is the Subtraction of the value in registers $a1(5) from $a0(4) and save the result in $s1(17)   
7-5 = x(2) as the simulation shows the operation is made correctly  
