

**MIPS REPORT**

**Name: Mazen Aly Darwish.**

**Reg: 20100432.**

**Submitted to: Dr. Marwa ElShennawy and Eng. Esraa Khattab.**

**Components used in the project:**

**1-ALU**

**2-ALU CONTROL**

**3-INSTRUCTION MEMORY**

**4-DATA MEMORY**

**5-CONTROL UNIT**

**6-2\_1 32 BIT MUX**

**7-2\_1 5 BIT MUX**

**8-PC+4 ADDER**

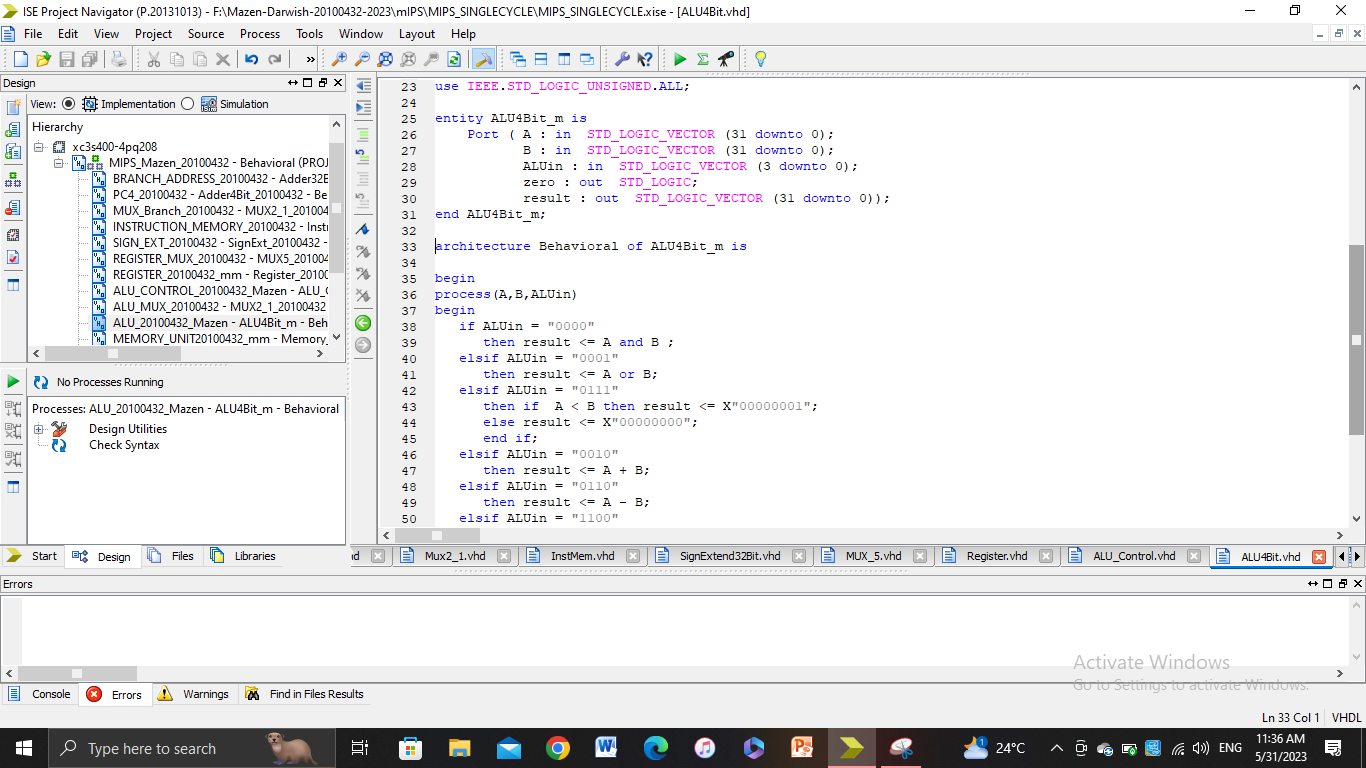
**9-32 BIT ADDER**

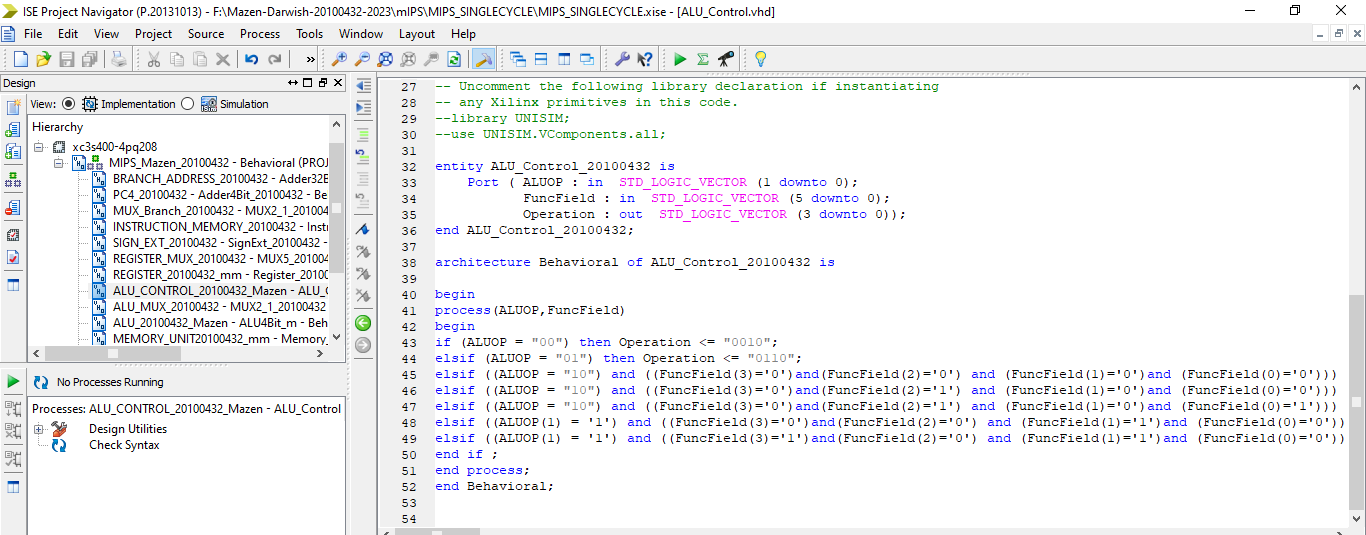
**10-REGISTER**

**11-PC COUNTER**

**12-SIGN EXTEND**

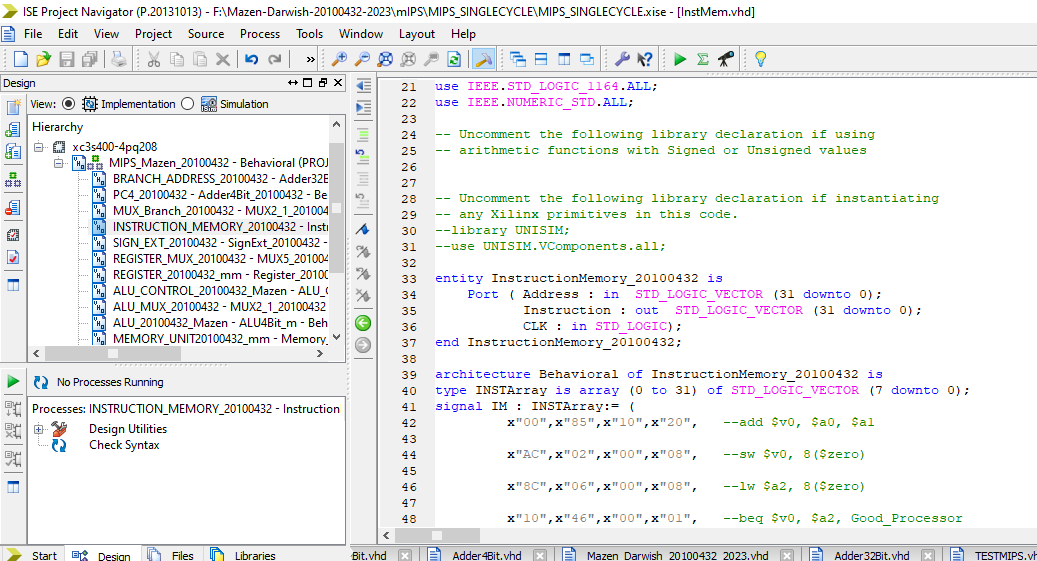
**13-SHIFT LEFT**

**1) ALU:**

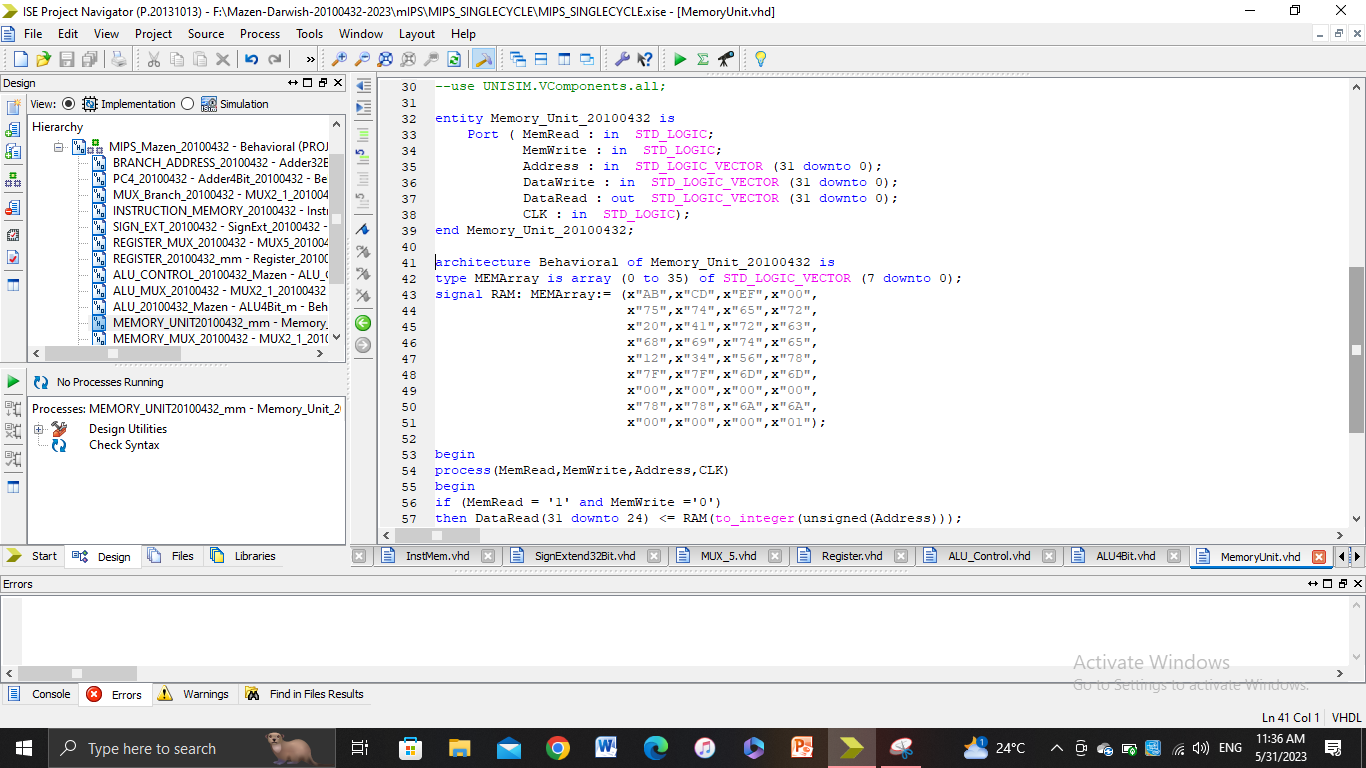
**2) ALU CONTROL:**

**3) INSTRUCTION MEMORY:**

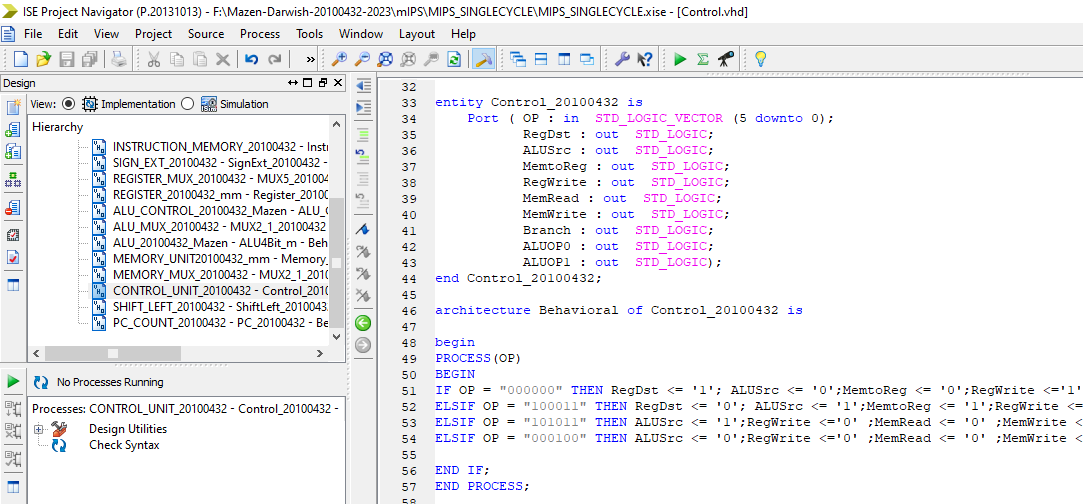
**(Sorry for the bad screenshot)**

****

**4) DATA MEMORY:**

****

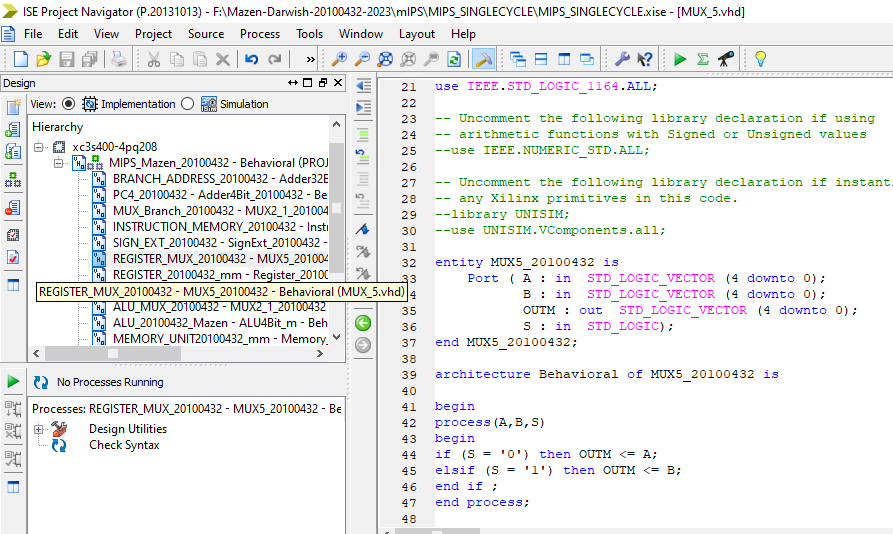
**5) CONTROL UNIT:**

****

**6) 2\_1 32 BIT MUX:A screenshot of a computer

Description automatically generated**

**7) 2\_1 5 BIT MUX:**

****

**8)PC+4 ADDER: A screenshot of a computer

Description automatically generated**

**9) 32 BIT ADDER:**

**A screenshot of a computer

Description automatically generated**

**10) Register:**

**A screenshot of a computer

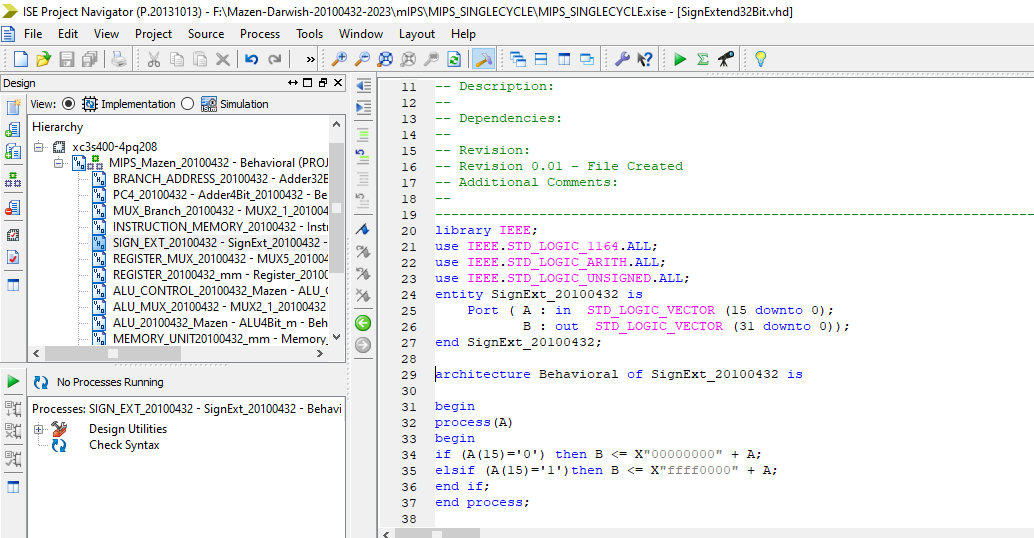
Description automatically generated**

**11) PC COUNTER:**

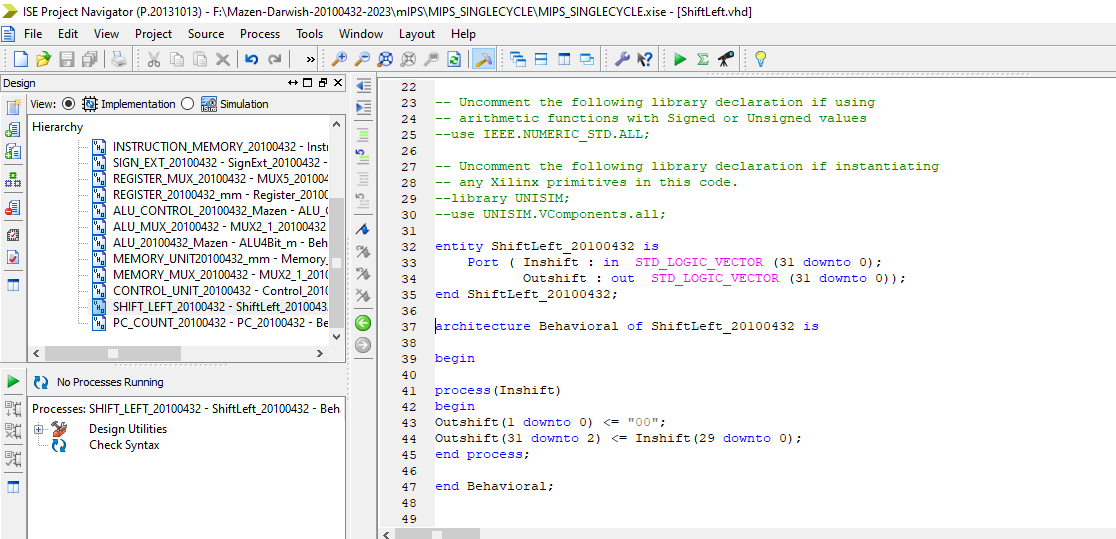
**A screenshot of a computer

Description automatically generated**

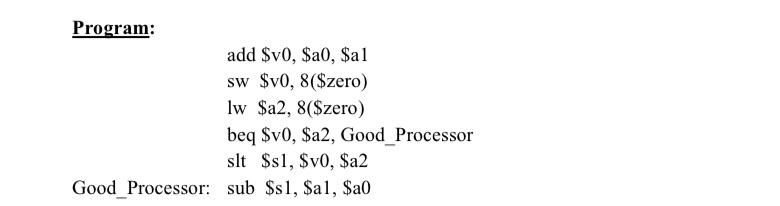
**12) SIGN EXTEND:**

****

**13) SHIFT LEFT:**

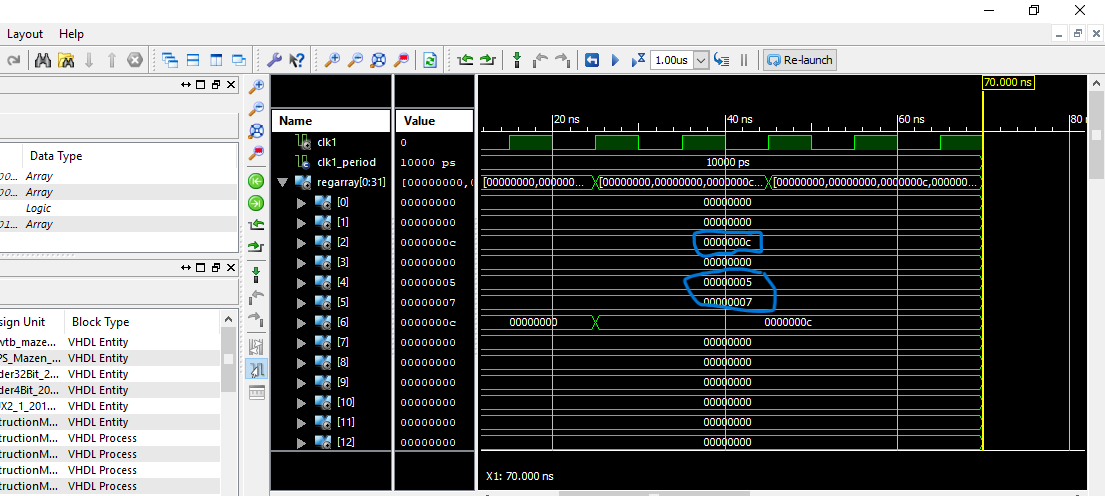
****

**SIMULATION:**

****

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

**Adds the values stored in registers: $a0 and $a1 and stores them in $v0 which is located in register 2.**

****

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

**This instruction will store the value in $v0 in the offset 8 (from 8 to 11) because mips is bit addressable.**

**A screenshot of a computer

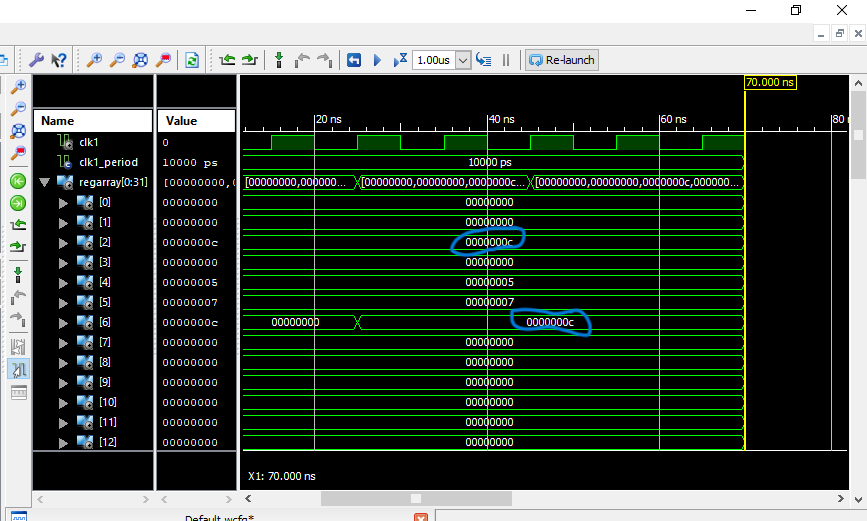
Description automatically generated**

**3- lw $a2, 8($zero)This instruction will load the value located in offset 8 (8 to 11) and store It in register $a2 which is register number 6. A screenshot of a computer

Description automatically generated**

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

**This instruction will subtract the values stored in registers $v0 and $a2 and if they are equal to 0 then it will proceed to the label.**

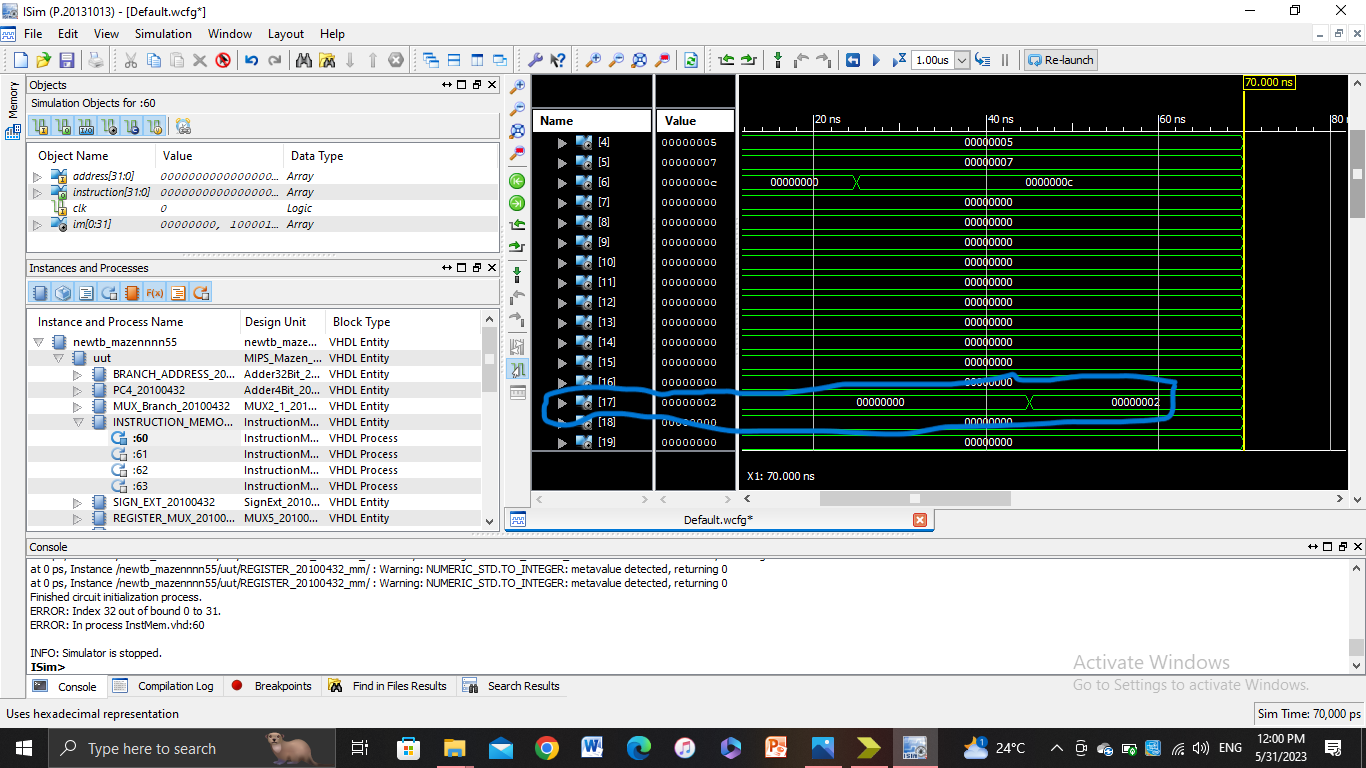
****

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

**This instruction will check if $v0 is less than $a2 and if this is wrong then it will set the value of $s1 to 1 and if the opposite is correct then it will set the value of $s1 to 0. A screen shot of a computer

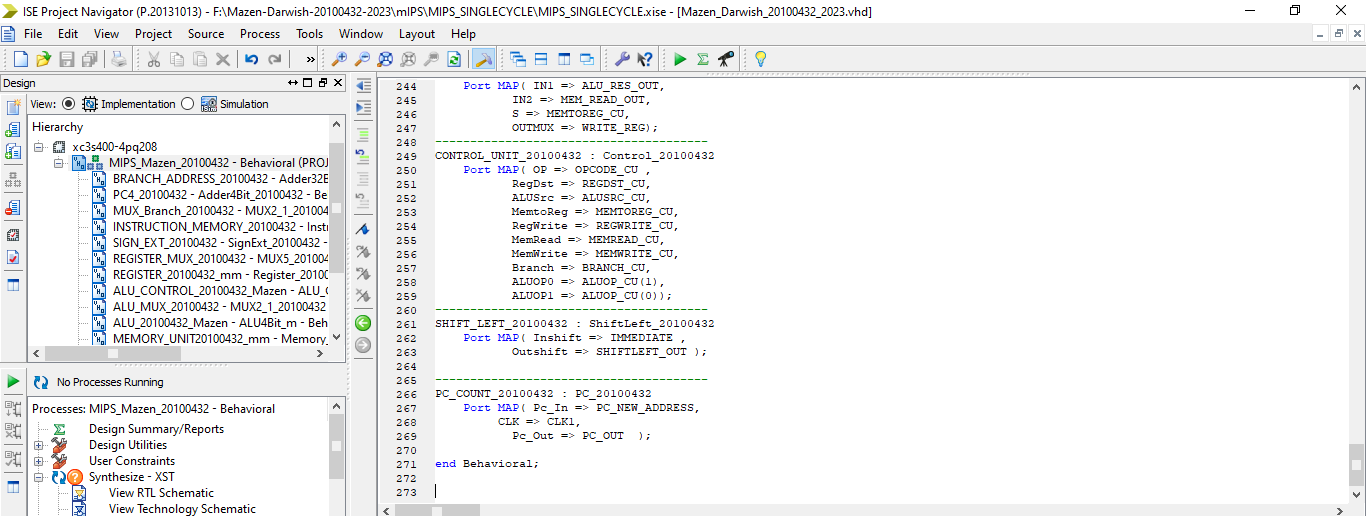
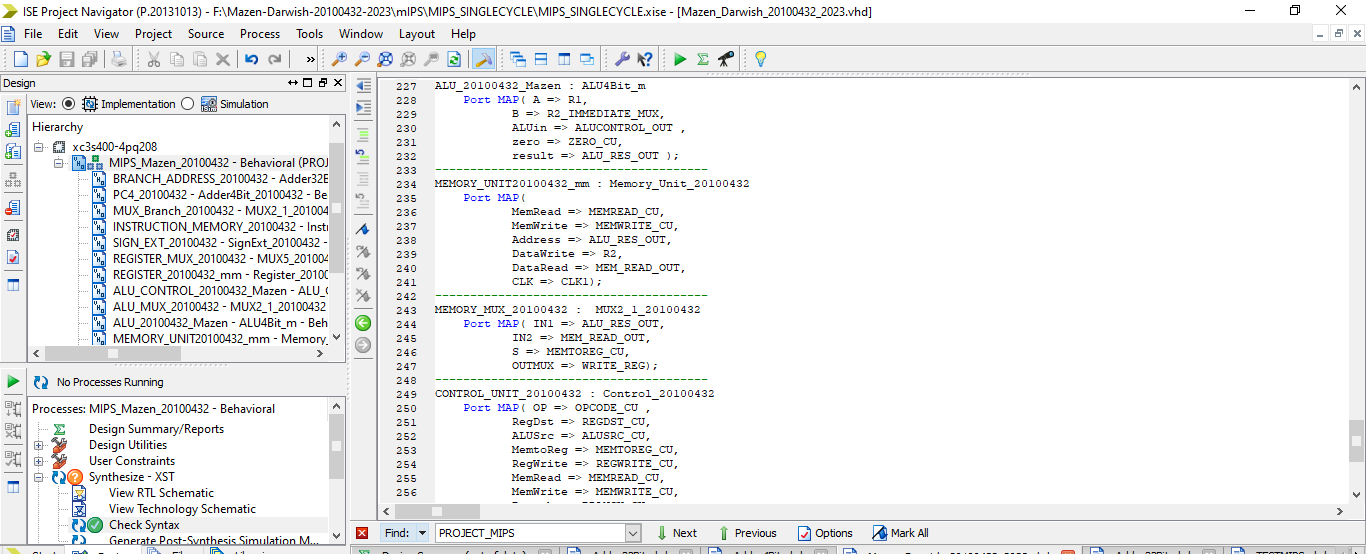
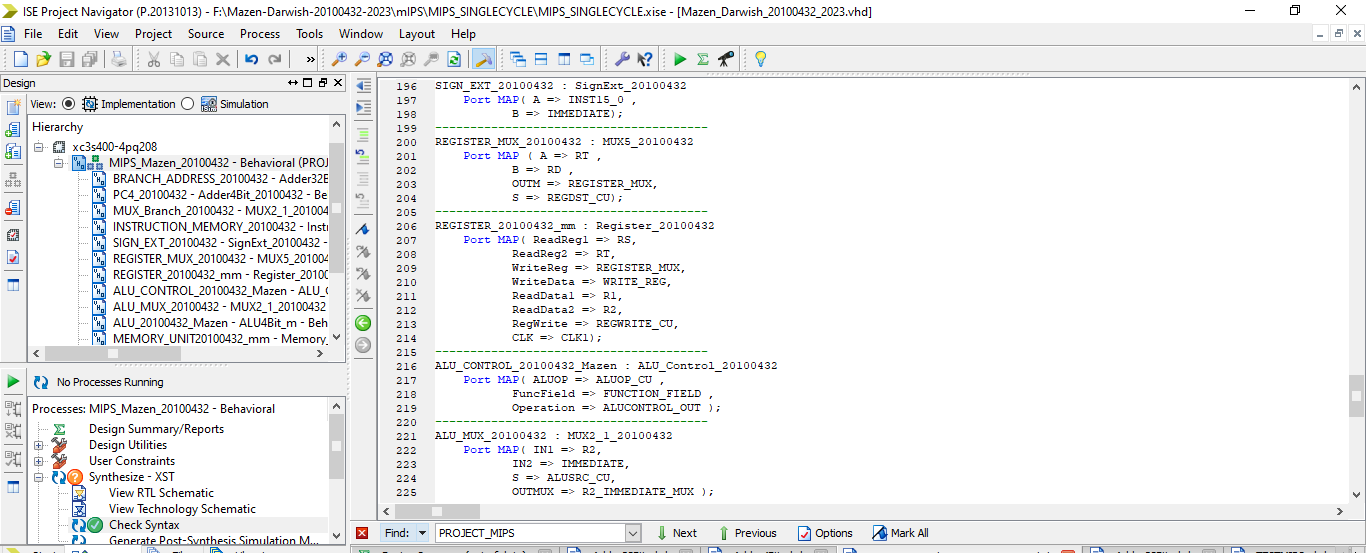
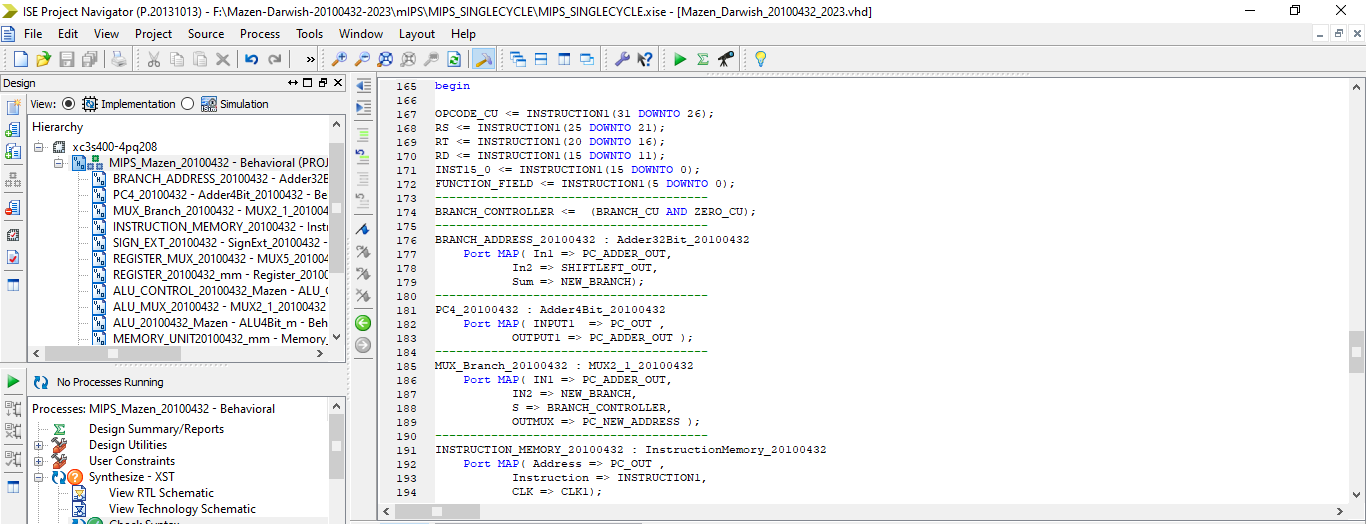
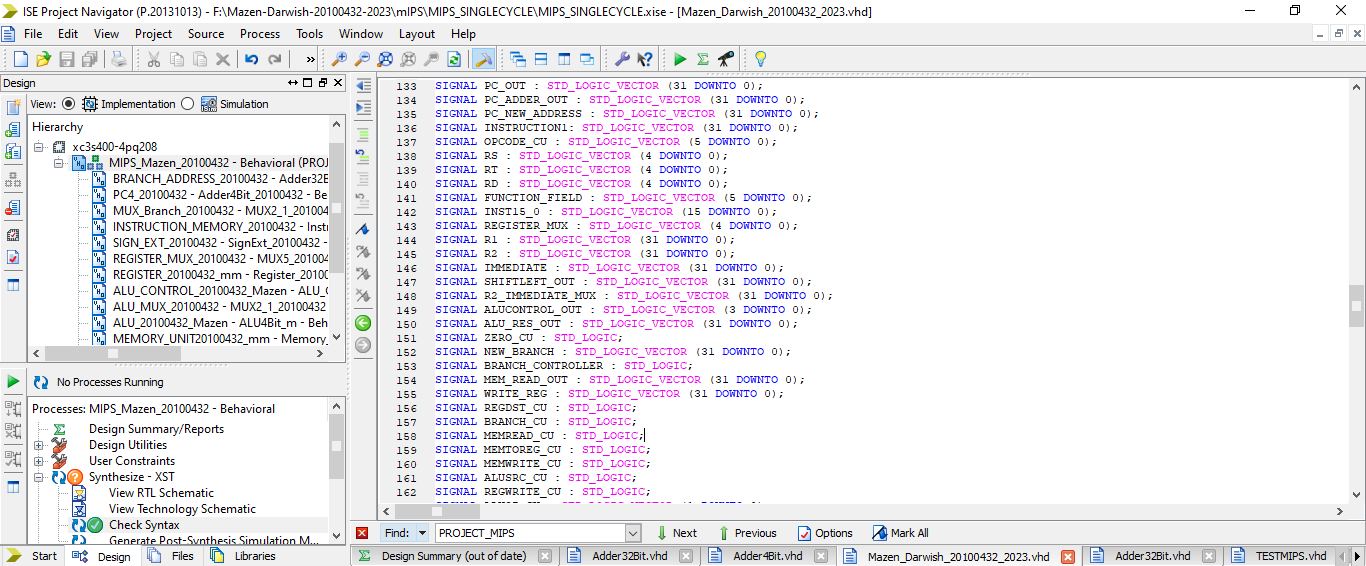
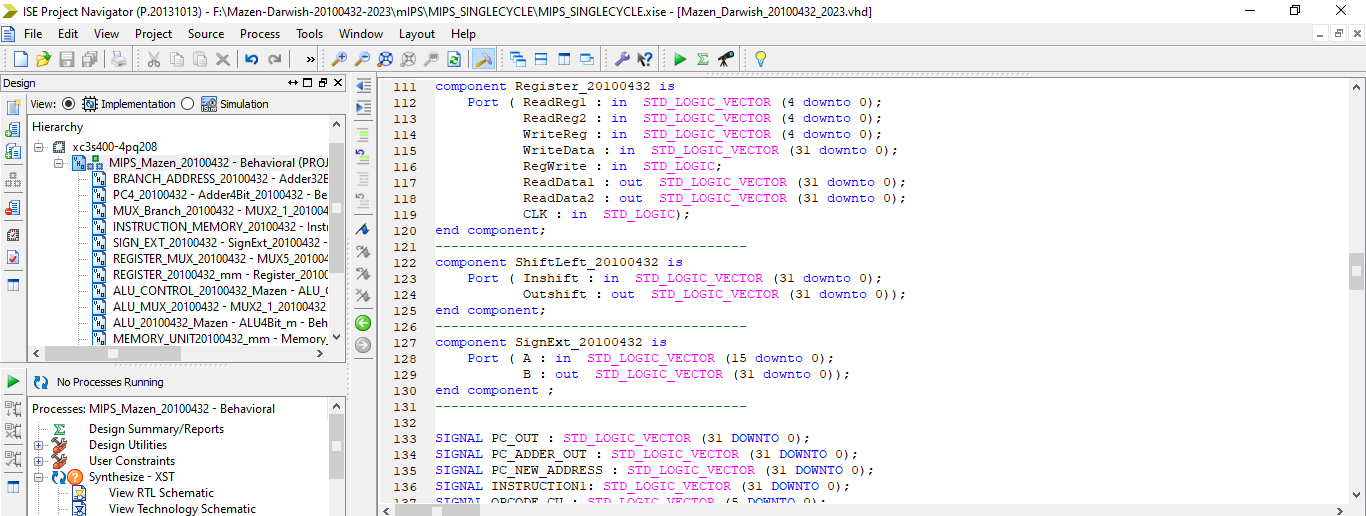
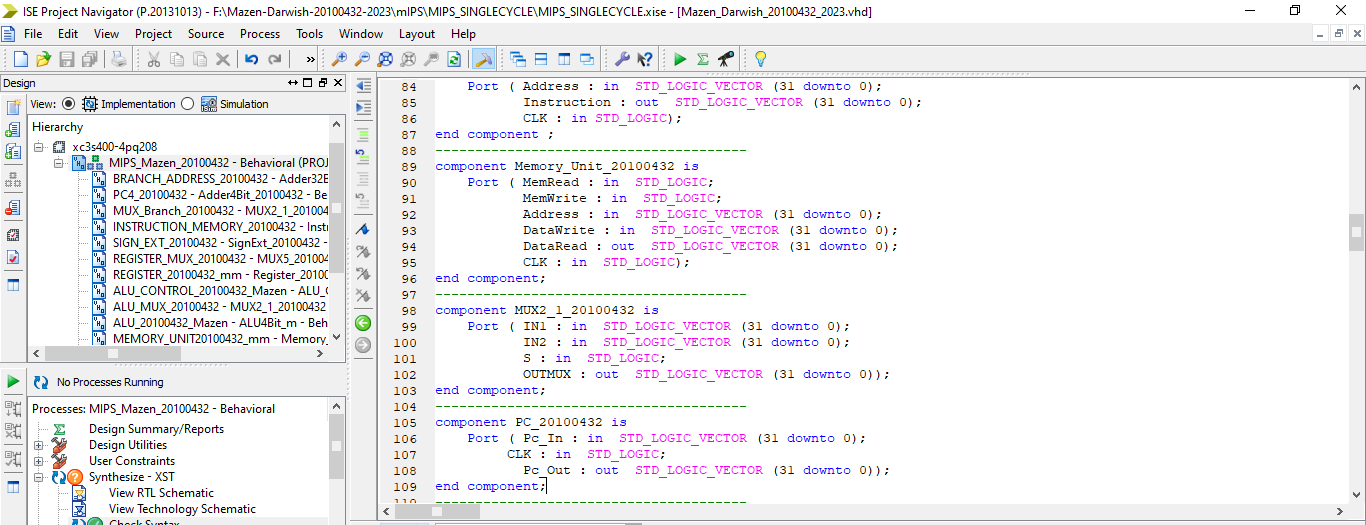
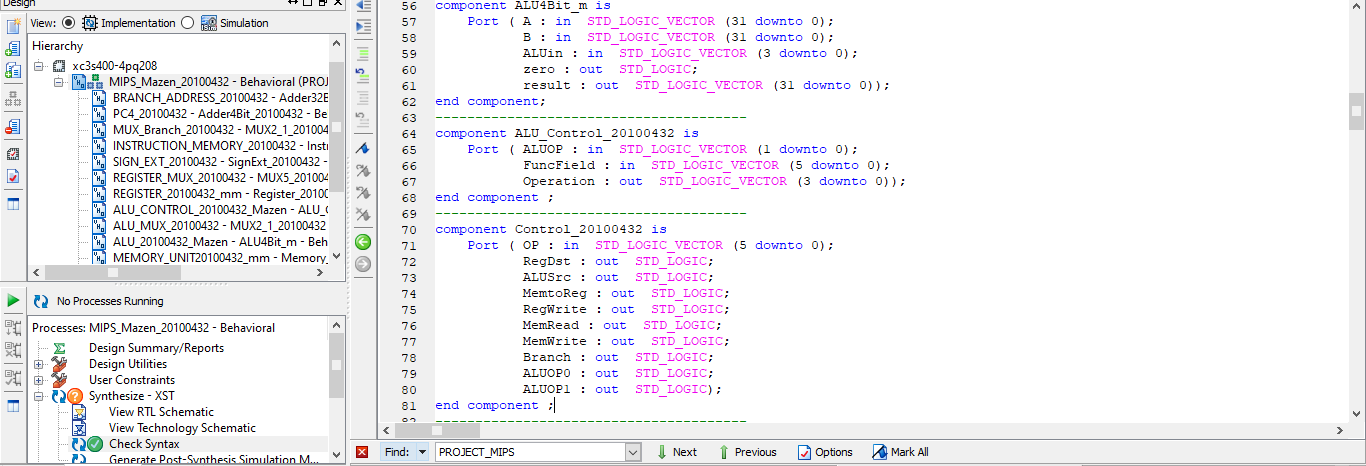
Description automatically generated with medium confidence**

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

**This instruction subtracts the values stored in registers: $a1 and $a0 and stores them in $s1 which is located in register 17.**

**PORT MAPPING:**

**A screenshot of a computer

Description automatically generated**