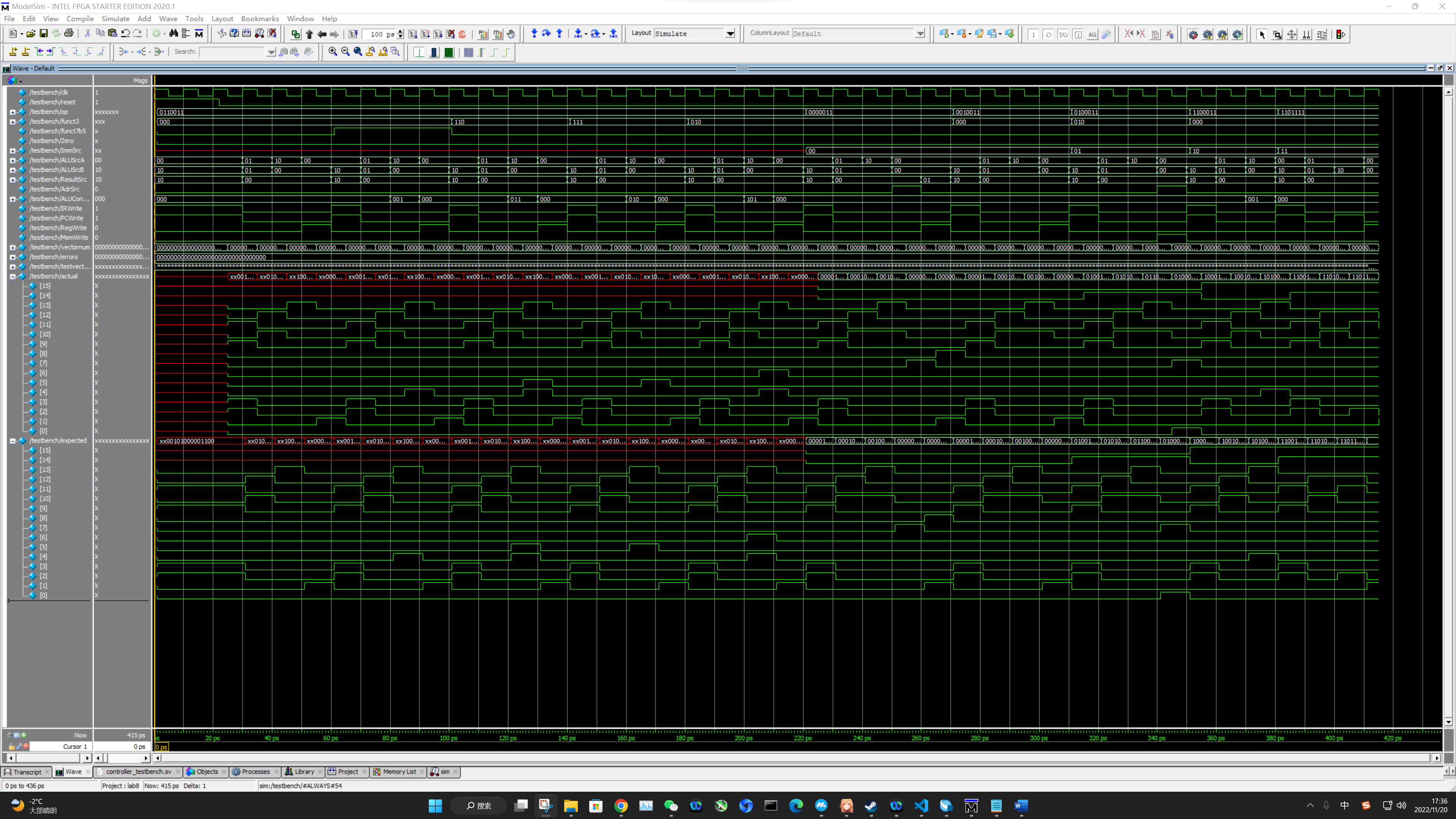
Lab 8

Time Spent: 10hours



The results match exactly what I want. The testbench file passed with no errors.

RTL:

图示

描述已自动生成

图示, 示意图

描述已自动生成图示, 示意图

描述已自动生成

文本

描述已自动生成Code:

文本

描述已自动生成

图形用户界面

低可信度描述已自动生成

文本

描述已自动生成Test Bench

Test Vector:

//op[6:0]\_funct3[2:0]\_funct7b5\_Zero\_ImmSrc[1:0]\_ALUSrcA[1:0]\_ALUSrcB[1:0]\_ResultSrc[1:0]\_AdrSrc\_ALUControl[2:0]\_IRWrite\_PCWrite\_RegWrite\_MemWrite

// add

0110011\_000\_0\_0\_\_XX\_00\_10\_10\_0\_000\_1\_1\_0\_0 // Fetch

0110011\_000\_0\_0\_\_XX\_01\_01\_00\_0\_000\_0\_0\_0\_0 // Decode

0110011\_000\_0\_0\_\_XX\_10\_00\_00\_0\_000\_0\_0\_0\_0 // ExecuteR

0110011\_000\_0\_0\_\_XX\_00\_00\_00\_0\_000\_0\_0\_1\_0 // ALUWB

// sub (only differs in ALUControl during ExecuteR)

0110011\_000\_1\_0\_\_XX\_00\_10\_10\_0\_000\_1\_1\_0\_0 // Fetch

0110011\_000\_1\_0\_\_XX\_01\_01\_00\_0\_000\_0\_0\_0\_0 // Decode

0110011\_000\_1\_0\_\_XX\_10\_00\_00\_0\_001\_0\_0\_0\_0 // ExecuteR

0110011\_000\_1\_0\_\_XX\_00\_00\_00\_0\_000\_0\_0\_1\_0 // ALUWB

// or (only differs in ALUControl during ExecuteR)

0110011\_110\_0\_0\_\_XX\_00\_10\_10\_0\_000\_1\_1\_0\_0 // Fetch

0110011\_110\_0\_0\_\_XX\_01\_01\_00\_0\_000\_0\_0\_0\_0 // Decode

0110011\_110\_0\_0\_\_XX\_10\_00\_00\_0\_011\_0\_0\_0\_0 // ExecuteR

0110011\_110\_0\_0\_\_XX\_00\_00\_00\_0\_000\_0\_0\_1\_0 // ALUWB

// and (only differs in ALUControl during ExecuteR)

0110011\_111\_0\_0\_\_XX\_00\_10\_10\_0\_000\_1\_1\_0\_0 // Fetch

0110011\_111\_0\_0\_\_XX\_01\_01\_00\_0\_000\_0\_0\_0\_0 // Decode

0110011\_111\_0\_0\_\_XX\_10\_00\_00\_0\_010\_0\_0\_0\_0 // ExecuteR

0110011\_111\_0\_0\_\_XX\_00\_00\_00\_0\_000\_0\_0\_1\_0 // ALUWB

// slt (only differs in ALUControl during ExecuteR)

0110011\_010\_0\_0\_\_XX\_00\_10\_10\_0\_000\_1\_1\_0\_0 // Fetch

0110011\_010\_0\_0\_\_XX\_01\_01\_00\_0\_000\_0\_0\_0\_0 // Decode

0110011\_010\_0\_0\_\_XX\_10\_00\_00\_0\_101\_0\_0\_0\_0 // ExecuteR

0110011\_010\_0\_0\_\_XX\_00\_00\_00\_0\_000\_0\_0\_1\_0 // ALUWB

// Test I-type instructions (lw, addi)

// lw takes five cycles to execute, addi takes four cycles

// ImmSrc = 00

// Other outputs based on Fig Multi-cycle controller FSM

//{Op[6:0]}\_{Funct3[2:0]}\_{Funct7b5}\_{Zero}\_\_{ImmSrc[1:0]}\_{ALUSrcA[1:0]}\_{ALUSrcB[1:0]}\_{ResultSrc[1:0]}\_{AdrSrc}\_{ALUControl[2:0]}\_{IRWrite}\_{PCWrite}\_{RegWrite}\_{MemWrite}

// lw

0000011\_010\_0\_0\_\_00\_00\_10\_10\_0\_000\_1\_1\_0\_0 // Fetch

0000011\_010\_0\_0\_\_00\_01\_01\_00\_0\_000\_0\_0\_0\_0 // Decode

0000011\_010\_0\_0\_\_00\_10\_01\_00\_0\_000\_0\_0\_0\_0 // MemAdr

0000011\_010\_0\_0\_\_00\_00\_00\_00\_1\_000\_0\_0\_0\_0 // MemRead

0000011\_010\_0\_0\_\_00\_00\_00\_01\_0\_000\_0\_0\_1\_0 // MemWB

// addi

0010011\_000\_0\_0\_\_00\_00\_10\_10\_0\_000\_1\_1\_0\_0 // Fetch

0010011\_000\_0\_0\_\_00\_01\_01\_00\_0\_000\_0\_0\_0\_0 // Decode

0010011\_000\_0\_0\_\_00\_10\_01\_00\_0\_000\_0\_0\_0\_0 // ExecuteI

0010011\_000\_0\_0\_\_00\_00\_00\_00\_0\_000\_0\_0\_1\_0 // ALUWB

// Test S-type instructions (sw)

// sw takes four cycles to execute

// ImmSrc = 01

// Other outputs based on Fig Multi-cycle controller FSM

//{Op[6:0]}\_{Funct3[2:0]}\_{Funct7b5}\_{Zero}\_\_{ImmSrc[1:0]}\_{ALUSrcA[1:0]}\_{ALUSrcB[1:0]}\_{ResultSrc[1:0]}\_{AdrSrc}\_{ALUControl[2:0]}\_{IRWrite}\_{PCWrite}\_{RegWrite}\_{MemWrite}

// sw

0100011\_010\_0\_0\_\_01\_00\_10\_10\_0\_000\_1\_1\_0\_0 // Fetch

0100011\_010\_0\_0\_\_01\_01\_01\_00\_0\_000\_0\_0\_0\_0 // Decode

0100011\_010\_0\_0\_\_01\_10\_01\_00\_0\_000\_0\_0\_0\_0 // MemAdr

0100011\_010\_0\_0\_\_01\_00\_00\_00\_1\_000\_0\_0\_0\_1 // MemWrite

// Test B-type instructions (beq)

// beq takes three cycles to execute

// ImmSrc = 10

// Other outputs based on Fig Multi-cycle controller FSM

//{Op[6:0]}\_{Funct3[2:0]}\_{Funct7b5}\_{Zero}\_\_{ImmSrc[1:0]}\_{ALUSrcA[1:0]}\_{ALUSrcB[1:0]}\_{ResultSrc[1:0]}\_{AdrSrc}\_{ALUControl[2:0]}\_{IRWrite}\_{PCWrite}\_{RegWrite}\_{MemWrite}

// beq

1100011\_000\_0\_0\_\_10\_00\_10\_10\_0\_000\_1\_1\_0\_0 // Fetch

1100011\_000\_0\_0\_\_10\_01\_01\_00\_0\_000\_0\_0\_0\_0 // Decode

1100011\_000\_0\_0\_\_10\_10\_00\_00\_0\_001\_0\_0\_0\_0 // BEQ

// Test J-type instructions (jal)

// beq takes four cycles to execute

// ImmSrc = 11

// Other outputs based on Fig Multi-cycle controller FSM

//{Op[6:0]}\_{Funct3[2:0]}\_{Funct7b5}\_{Zero}\_\_{ImmSrc[1:0]}\_{ALUSrcA[1:0]}\_{ALUSrcB[1:0]}\_{ResultSrc[1:0]}\_{AdrSrc}\_{ALUControl[2:0]}\_{IRWrite}\_{PCWrite}\_{RegWrite}\_{MemWrite}

1101111\_000\_0\_0\_\_11\_00\_10\_10\_0\_000\_1\_1\_0\_0 // Fetch

1101111\_000\_0\_0\_\_11\_01\_01\_00\_0\_000\_0\_0\_0\_0 // Decode

1101111\_000\_0\_0\_\_11\_01\_10\_00\_0\_000\_0\_1\_0\_0 // JAL

1101111\_000\_0\_0\_\_11\_00\_00\_00\_0\_000\_0\_0\_1\_0 // ALUWB