PROJECT 3 – PIPELINE – CHECKLIST

Verify that each item on this check list is correct in your output before submitting. ***If your output does not conform to every requirement listed below, do not submit your project.*** Go back and rework your project so that all the listed items are provided correctly.

If you submit without all the information detailed on this check list, there may be an automatic 5-point deduction and the TA will send it back to you to rework.

Item 1:

Turn in through Blackboard.

Item 2:

Both your program code **AND** the complete output (*copy and paste from emulator into a* ***.txt file***) are attached to the email submission, either as individual files or as a zipped file (**.zip** format only).

Item 3:

The 32 user registers are printed out at **every single clock cycle**.

Item 4:

Each **clock cycle is labeled**, with its number (for instance, "Clock Cycle 9"), in the output.

Item 5:

Print out the pipeline registers **ONLY** as they are **before the copy from write to read side**.

Item 6:

All the fields as displayed in the Pipelines handout **must be displayed in the output**.

Item 7:

All the fields **are labeled with the same labels** as in the Pipelines handout.

Item 8:

The main memory array is called **Main_Mem** and the register array is called **Regs**. No exceptions.

Item 9:

Page 2 of this document is a screen shot of the Pipelines handout. **Model your output on this format.** You do not need to display items in the pipeline registers that deal only with branches.

If you have any questions at all, email the professor and the TA. That's what we're here for. 😊

Handout Screen Shot

Clock Cycle 5 (*Before* we copy the write side of pipeline registers to the read side)
IF/ID_Write (written to by the IF stage)
Inst = 0x10c8fffb [beq $6, $8, label] IncrPC = 7A014
IF/ID_Read (read by the ID stage)
Inst = 0x00625022 [sub $10, $3, $2] IncrPC = 7A010

ID/EX_Write (written to by the ID stage)
Control: RegDst=1, ALUSrc=0, ALUOp=10, MemRead=0, MemWrite=0,
Branch=0, MemToReg=0, RegWrite=1, [sub]
Incr PC = 7A010 ReadReg1Value = 30003 ReadReg2Value = 30002
SEOffset = X WriteReg_20_16 = 2 WriteReg_15_11 = 10 Function = 22

ID/EX_Read (read by the EX stage)
Control: RegDst=X, ALUSrc=1, ALUOp=00, MemRead=0, MemWrite=1,
Branch=0, MemToReg=X, RegWrite=0, [sw]
Incr PC = 7A00C ReadReg1Value = 30008 ReadReg2Value = 30009
SEOffset = FFFFFFFC WriteReg_20_16 = 9 WriteReg_15_11 = 31 Function = x

EX/MEM_Write (written to by the EX stage)
Control: MemRead=0, MemWrite=1, Branch=0, MemToReg=X, RegWrite=0, [sw]
CalcBTA = X Zero = F ALUResult = 30004
SWValue = 30009 WriteRegNum = X

EX/MEM_Read (read by the MEM stage)
Control: MemRead=1, MemWrite=0, Branch=0, MemToReg=1, RegWrite=1, [lw]
CalcBTA = X Zero = F ALUResult = 3000C
SWValue = 3000F WriteRegNum = 15

MEM/WB_Write (written to by the MEM stage)
Control: MemToReg=1, RegWrite=1, [lw]
LWDataValue = mem contents @ 3000C ALUResult = 3000C WriteRegNum = 15

MEM/WB_Read (read by the WB stage)
Control: MemToReg=0, RegWrite=1, [add]
LWDataValue = X ALUResult = 6000B WriteRegNum = 7

($7 is set to a new value of 6000B)

NOTE:

You do not need to *format* your output exactly as in the handout. If it's easier to put all your fields one after another on separate lines, or any other reasonable formatting, that's fine. As long as everything is **labeled clearly and correctly**, the formatting is less important.