March 5, 2025

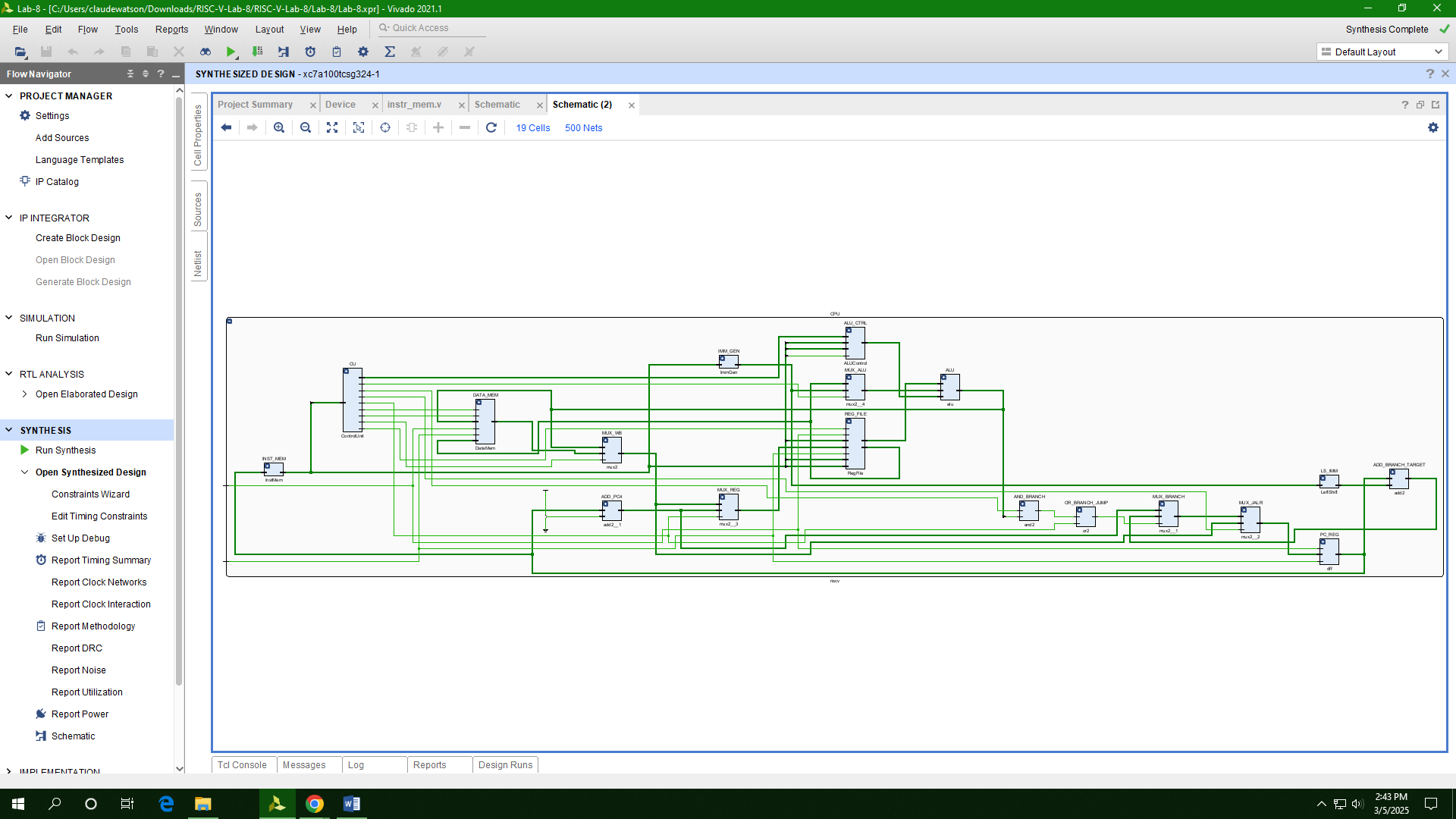
In class work:

T1: Why do we not need the lower two bits of the address when reading from the instruction memory?

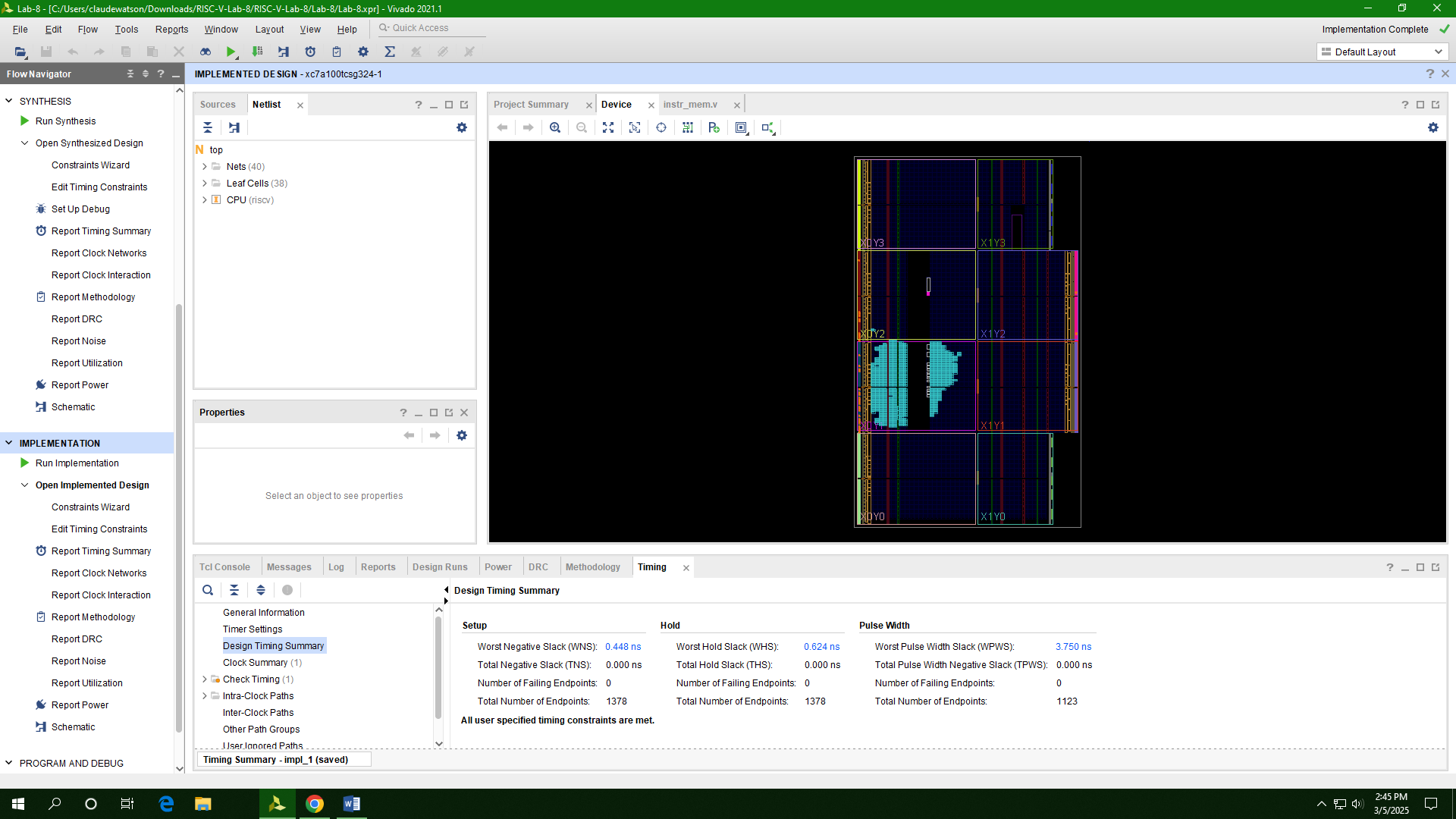
We don’t need the lower 4 bits because each RISC-V instruction is 4 bytes (32 bits), and stored in address that are multiples of 4 (0, 4, 8, 12…). Since the CPU always fetches the full instructions, the lower two bits will always be 00 and can be ignored. So instead of using the full address, we divide by 4, (addr[ADDRESS\_WIDTH – 1 : 2]) to access the correct instruction in memory.

T2: Take a screenshot of the block diagram generated by Vivado. Remember to click into the

CPU block before taking the screenshot so it shows the actual design.



T3: Take a screenshot of the FPGA view in the Device tab.



T4: Report the utilization of logic resources (LUTs, FFs, DSPs) used by the design

|  |  |  |
| --- | --- | --- |
| LUTs | FFs | DSPs |
| 2605 | 1056 | 3 |

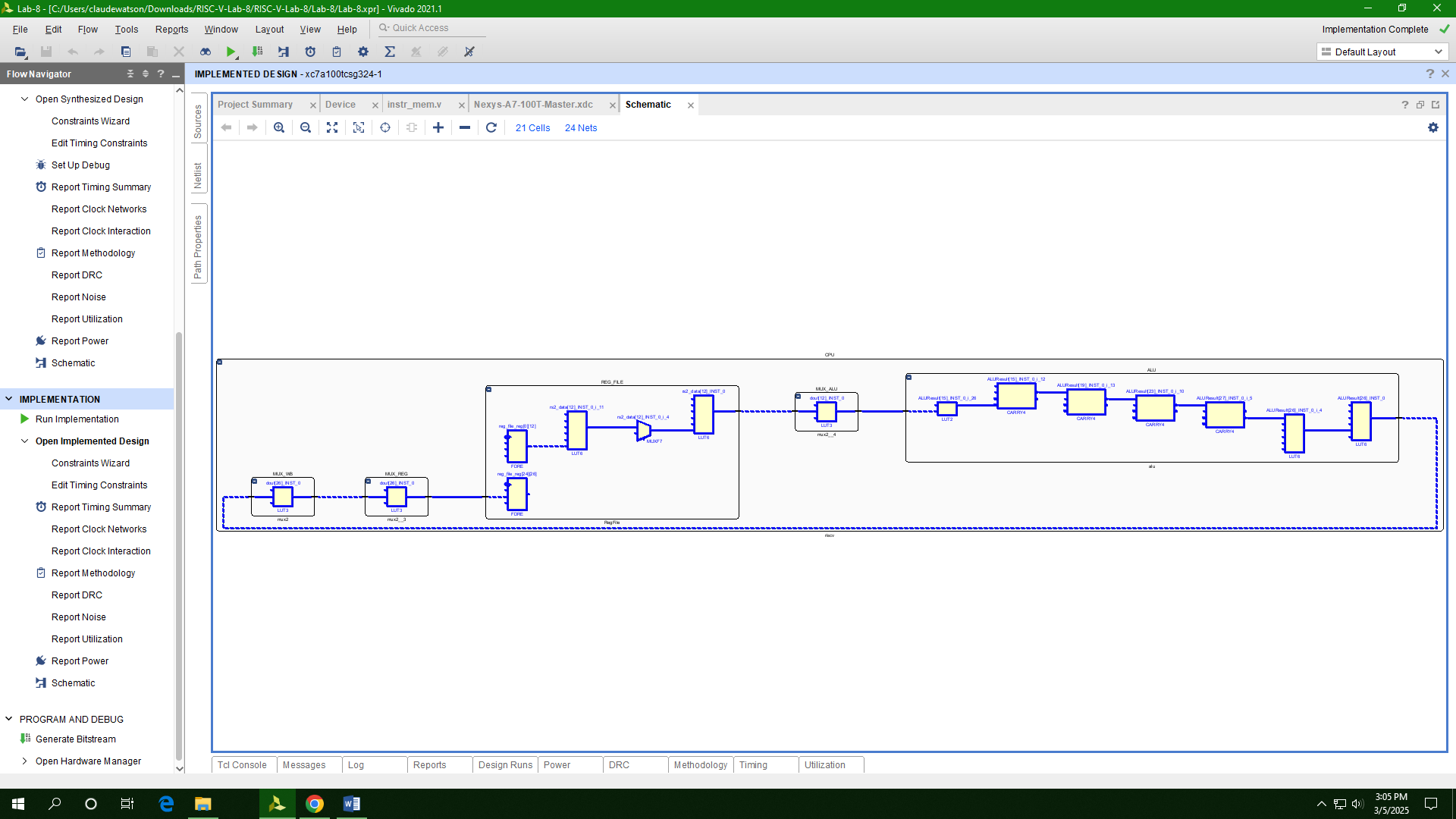
T5: What is the clock period/frequency for this design according to the create\_clock command?

* Period: 10ns
* Frequency: 100MHz

T6: What is the length (in nanoseconds) of the critical path? This will be the path with the highest Total Delay. Based on the clock period, how much “slack” is there in the timing? (That is, how much time is “leftover”.)

* The critical path is path 2, and it has a total delay of 9.411ns. Hence, there is a slack of 0.589ns.

T7: Take a screenshot of the schematic showing the critical path. What instruction(s) would you expect to use this path?

Critical paths involve the ALU operations, memory address and control signals propagations.. Typical instructions in RISC-V may include Arithmetic instructions, Load and store instructions and branch instructions.