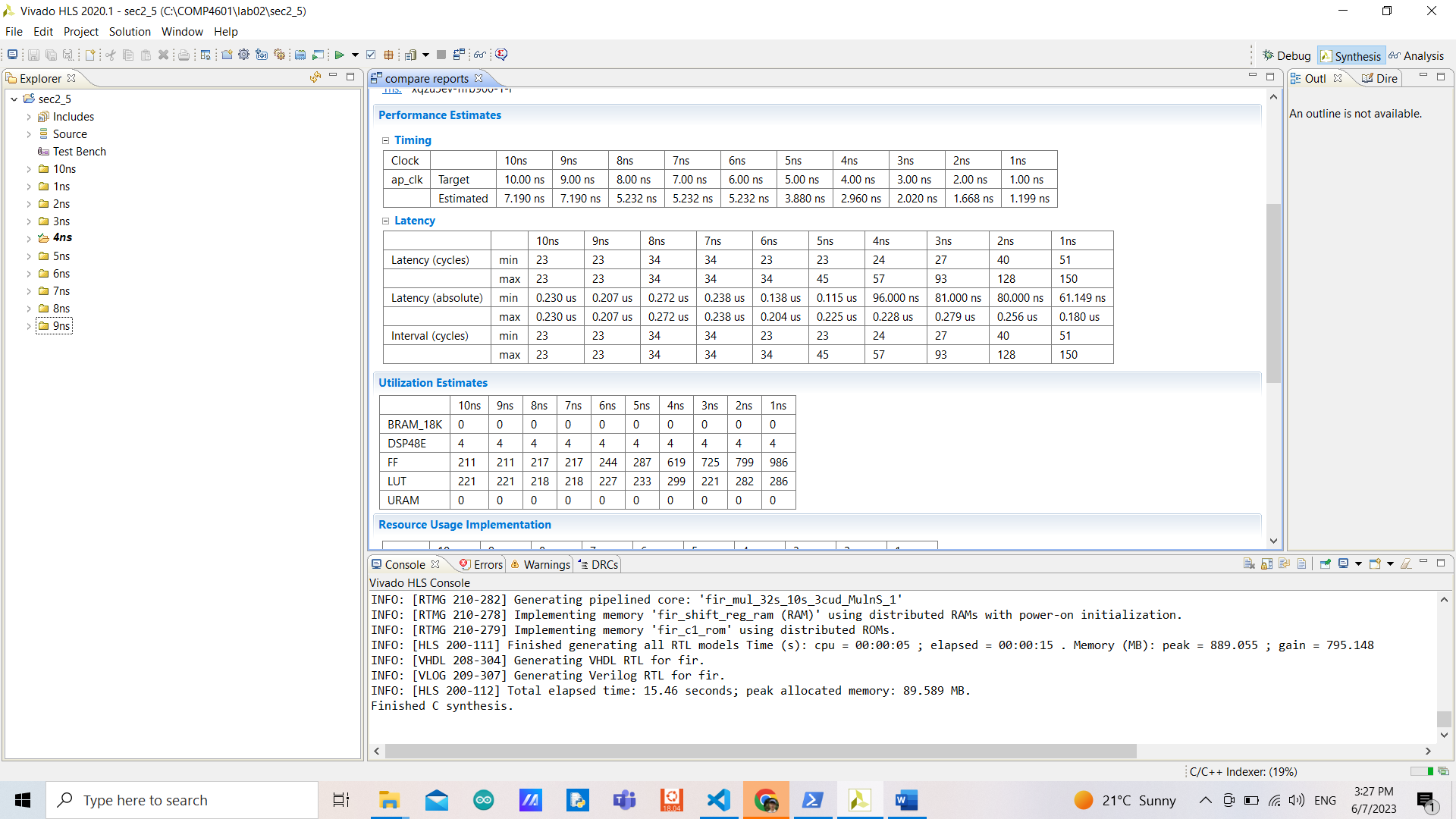
Mr. Phot Koseekrainiramon z5387411

23T2 COMP4601 Week 2 Lab Hand-in exercises

**Exercise 1:**

i.



ii. More clock period provides smaller area usage, with fewer intervals, fewer cycles, and fewer gaps between min and max, but more absolute latency.

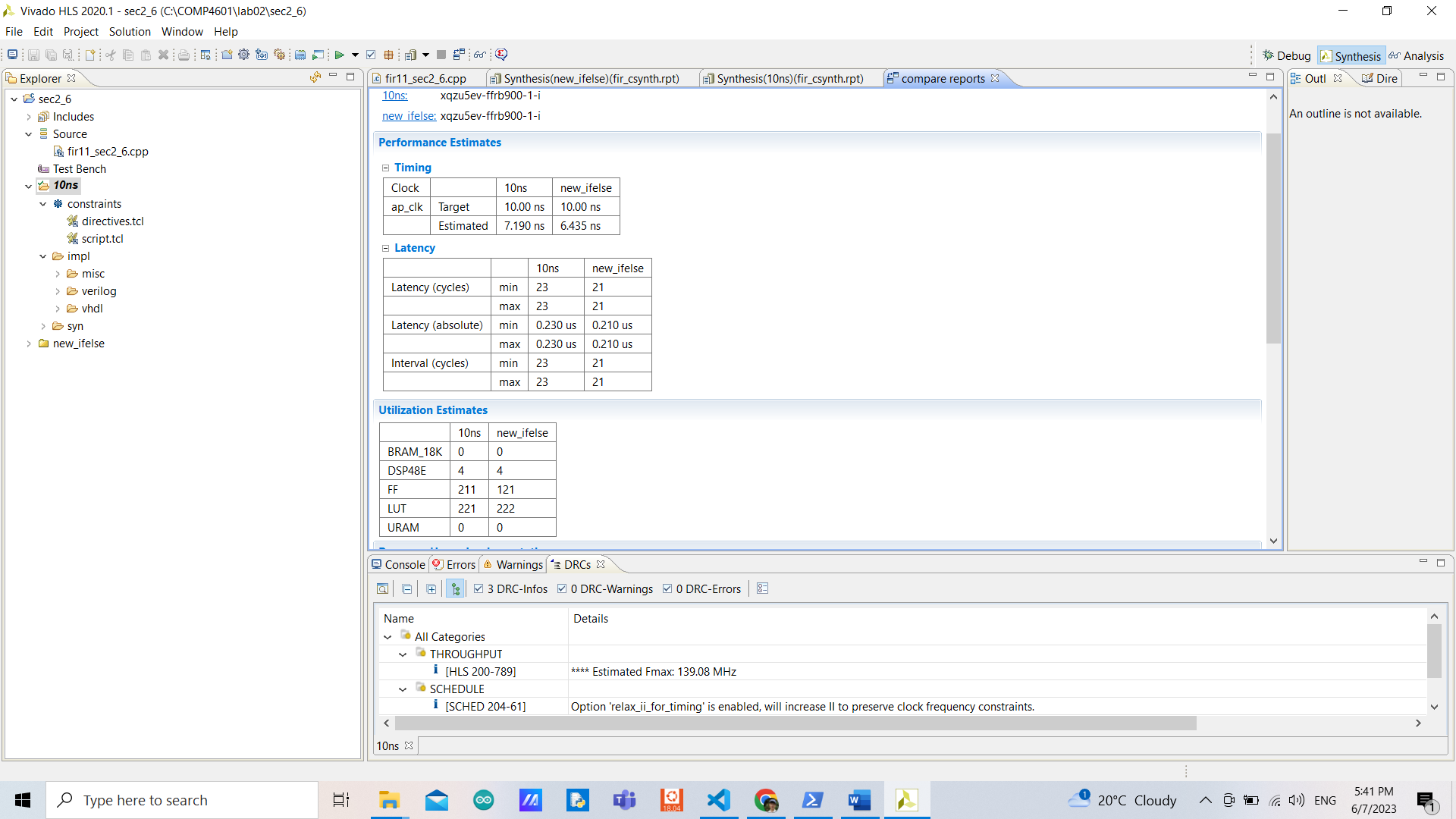
iii. In terms of latency, a clock period of 1ns performs the best with an average of 120ns absolute latency even though it takes the highest number of cycles. In terms of utilization, a clock period of 9ns performs the best with the least number of both flip-flops and look-up-table usages (with the same number of usages as a clock period of 10ns but having less absolute latency).

iv. The FPGA has 234240 flip-flops available and 117120 look-up tables available. By comparing 10ns and 1ns solutions, the usage of flip-flops is reduced by 0.003% while the usage of look-up tables is reduced by 0.0005%. This means the reduction of the usage of flip-flops is more significant than the usage of look-up tables even though the number of look-up tables available is less than the flip-flops.

v. The factor influencing the estimated clock period (worst-case delay) is the resources used since it needs to be optimized to meet the timing requirement.

vi. In the if-else statement, the first branch requires only 4 cycles but the other requires 13 cycles. That’s why the minimum iteration latency is 4 and the maximum is 13.

**Exercise 2:**

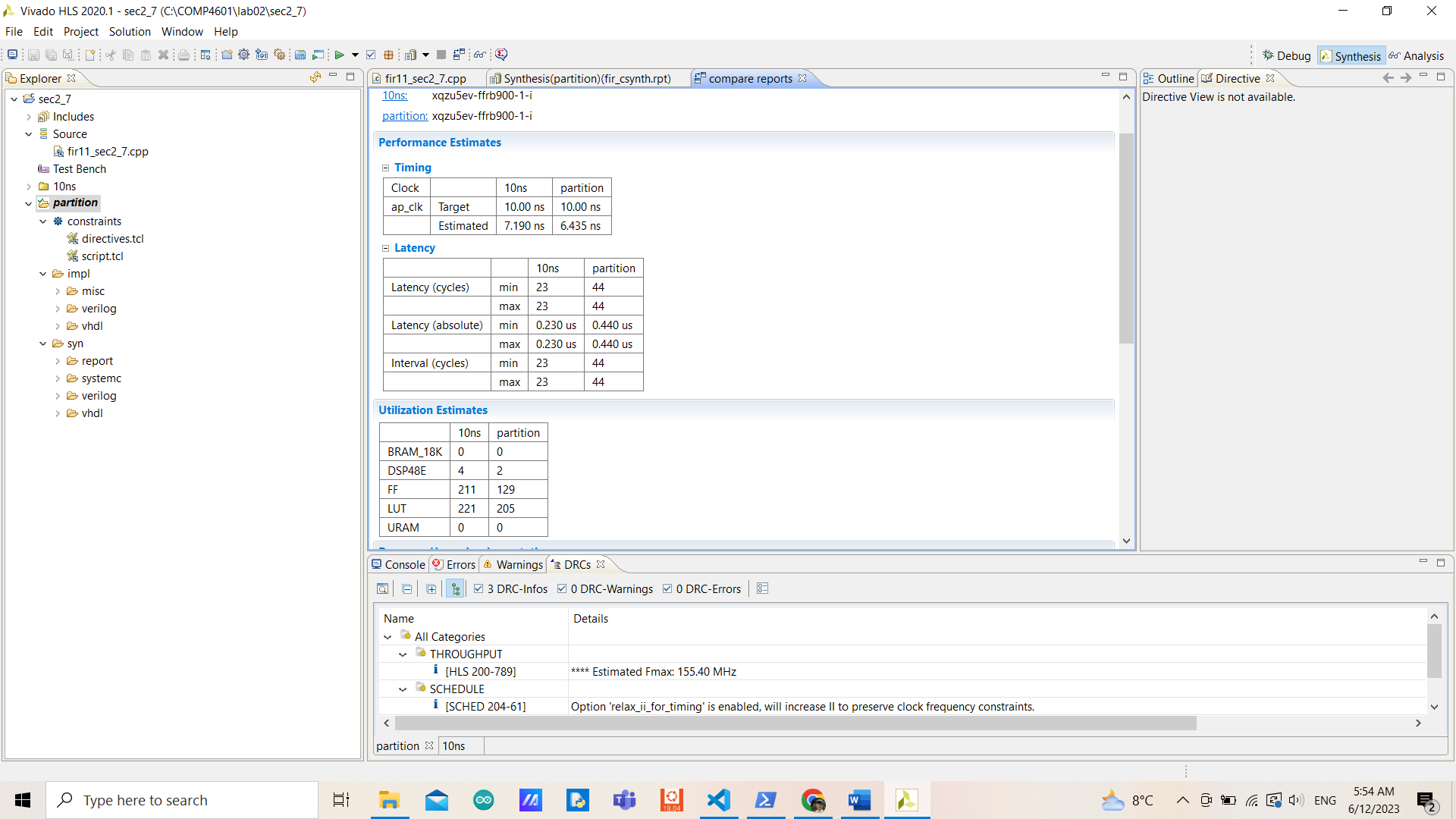


i. The new design has less latency and interval cycles, and less estimated clock period, with 1 extra look-up table and 90 fewer flip-flops.

ii. The new solution has a smaller estimated clock period with the same number of uncertainty. In terms of latency, the new solution takes 2 fewer cycles from iterating 1 less iteration.

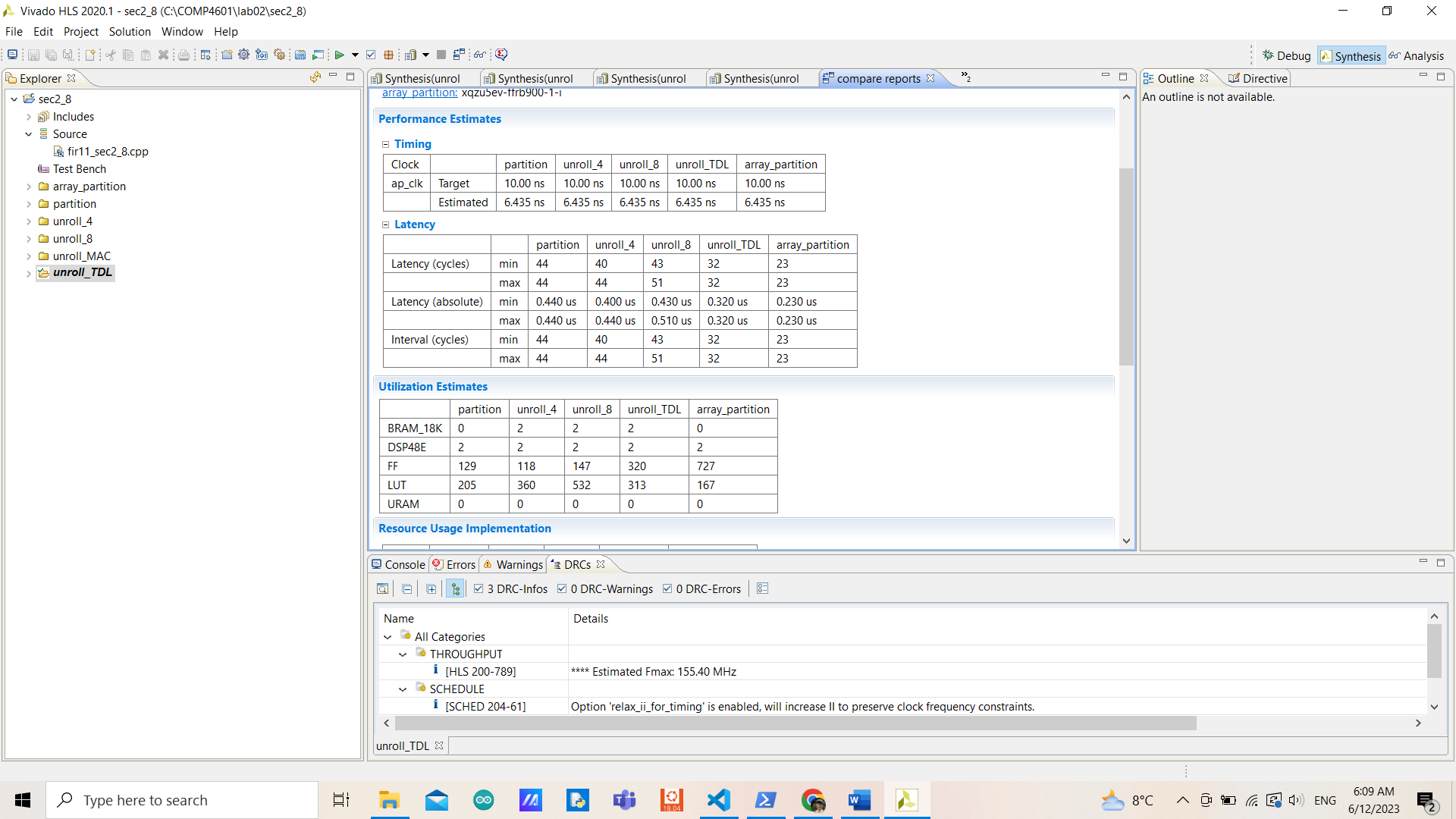
iii. Instead of iterating the loop 11 times, the new design only iterates 10 times and computes the last multiplication separately from the iteration.

**Exercise 3:**

****

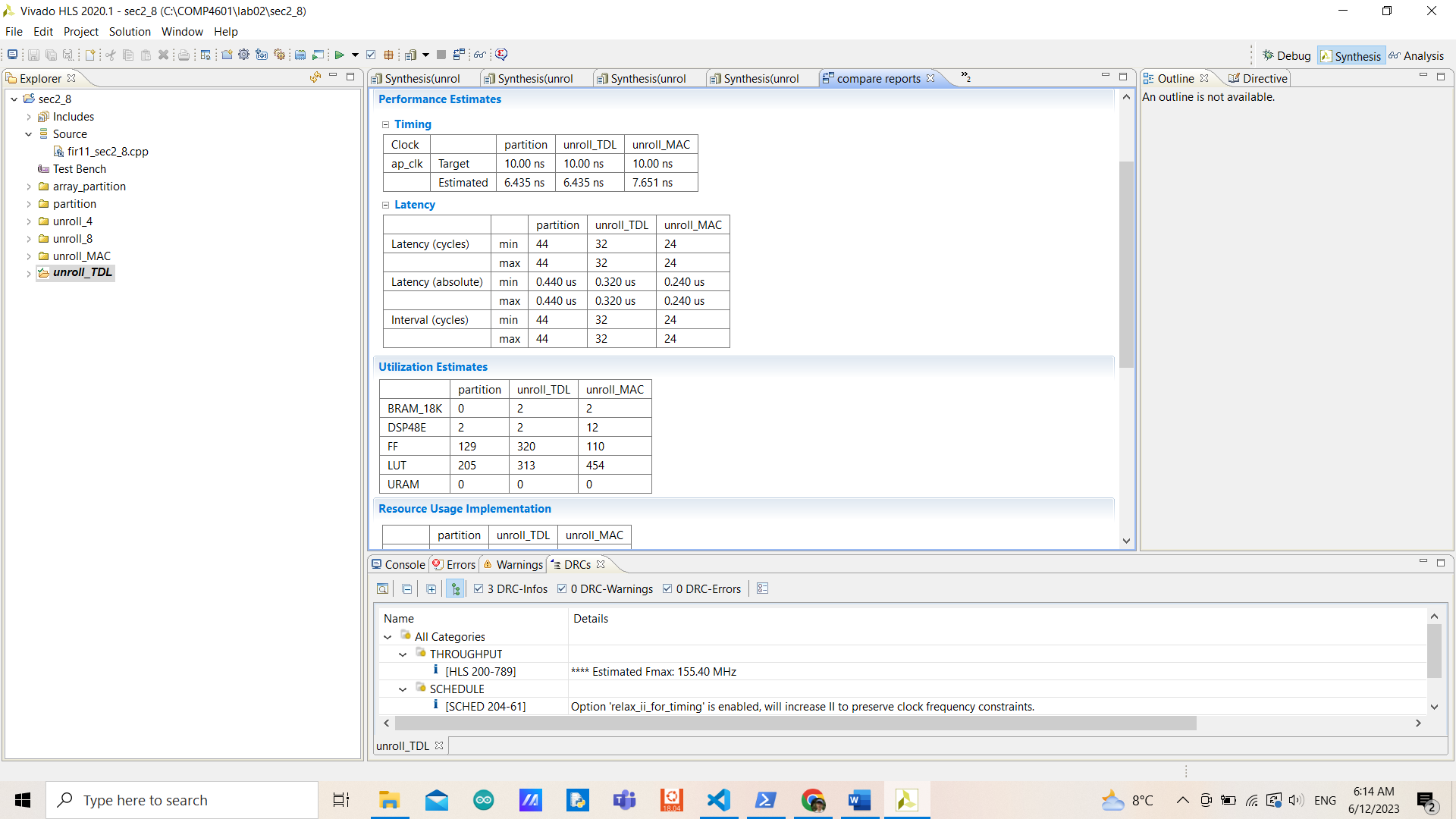
The new design has less estimated clock period. It has a higher number of interval and latency cycles since it is not pipelined and has to iterate 2 loops (TDL and MAC). It uses up fewer areas/resources (flip-flops, look-up tables, DSP) because it does fewer operations in one iteration.

**Exercise 4:**

****

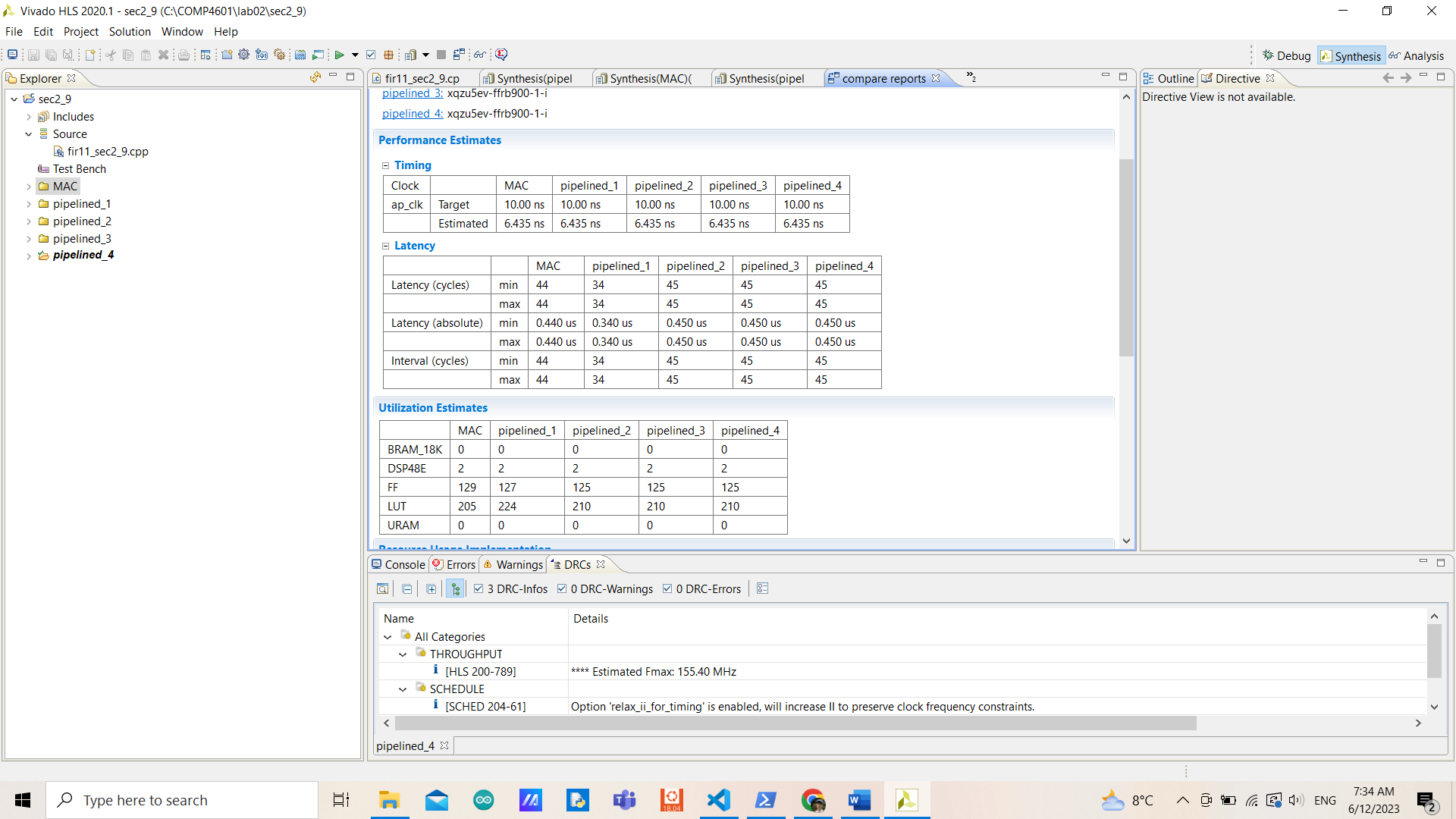
As the unroll factor is increased, the number of resources used is also increased for both flip-flips and look-up tables but not the others. Interval and latency cycles are also increased accordingly. In terms of latency, unrolling with a factor of 4 is slightly better compared to not unrolling, but with a factor of 8 is much worse with an increasing number of DSP, FF, and LUT. When unrolling completely, the number of resources used is changed (in this case, the number of flip-flops used is increased but the look-up table is decreased compared to using other factors) to gain the best performance in terms of latency (better performances but used up more resources compared to not unrolling). However, when trying to partition the arrays while unrolling completely, it achieves a better performance in terms of latency with no BRAM used, more flip-flops used but fewer look-up tables used.

**Exercise 5:**



By unrolling the MAC loop, it has a higher estimated clock period but better performance in terms of latency (since TDL’s load and write take many cycles). It uses up more BRAM, DSP, and URAM but less FF. This is because it needs to do more multiplication at a time compared to the other two designs.

**Exercise 6:**



In terms of utilization, there are not many differences between them. In terms of latency, having pipelined with 1 II beats all the other solutions.

Increasing the II tends to increase latency cycles and decrease the number of resources used compared to less II, however, it stops changing after II equals to the number of the loop’s iteration latency (2 in this case).