# Amjad Hamed ENG2025 Coursework Report – Timing Performance

Student ID: 230733071

Module: Digital Electronics

**Timing Comparison Between Parallel and Serial Multiplier**

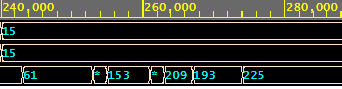
In this section, we compare the timing performance of the parallel multiplier (Design 1) and the serial multiplier (Design 2).

* Parallel Multiplier:
  + The parallel multiplier processes all bits at the same time, so the multiplication of two 4-bit numbers is completed in just one clock cycle per adder row.
  + This makes the parallel multiplier faster than the serial multiplier but requires more hardware components (AND gates, FAs, HAs).
* Serial Multiplier:
  + The serial multiplier operates on one bit at a time, processing the multiplicand and multiplier in order.
  + Each bit of the multiplier is processed sequentially, meaning the serial multiplier takes 4 clock cycles to complete the multiplication.
  + This approach requires fewer components (just shift registers and a single adder) but results in slower processing compared to the parallel approach.

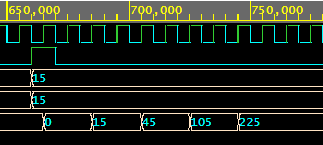
Trade-Off Analysis

* The parallel multiplier is faster because all the bits are processed simultaneously, but it requires more hardware resources (full adders, half adders, and gates). Hence, it will be better for applications where speed is the highest priority no matter the cost.
* The serial multiplier is slower due to the bit-by-bit processing but requires fewer hardware resources. Thus, it will be better for applications were cost and power are the most important.

Design Evidence and Conclusion:



Parallel Multiplier



Serial Multiplier

The performance of both the parallel multiplier and serial multiplier designs has been confirmed through extensive simulation in EDAPlayground. Simulation results have shown that the parallel multiplier performs faster (approximately 40,000 cycles) compared to the serial multiplier (approximately 100,000 cycles). The testbench for both designs used input values ranging from 0 to 15, with corresponding timing results captured using EPWave.

Therefore, the parallel multiplier offers higher speed but is more resource-intensive, while the serial multiplier, though slower, is more efficient in terms of hardware usage. The choice between the two designs depends on the specific needs of the application and trade off between cost and speed.