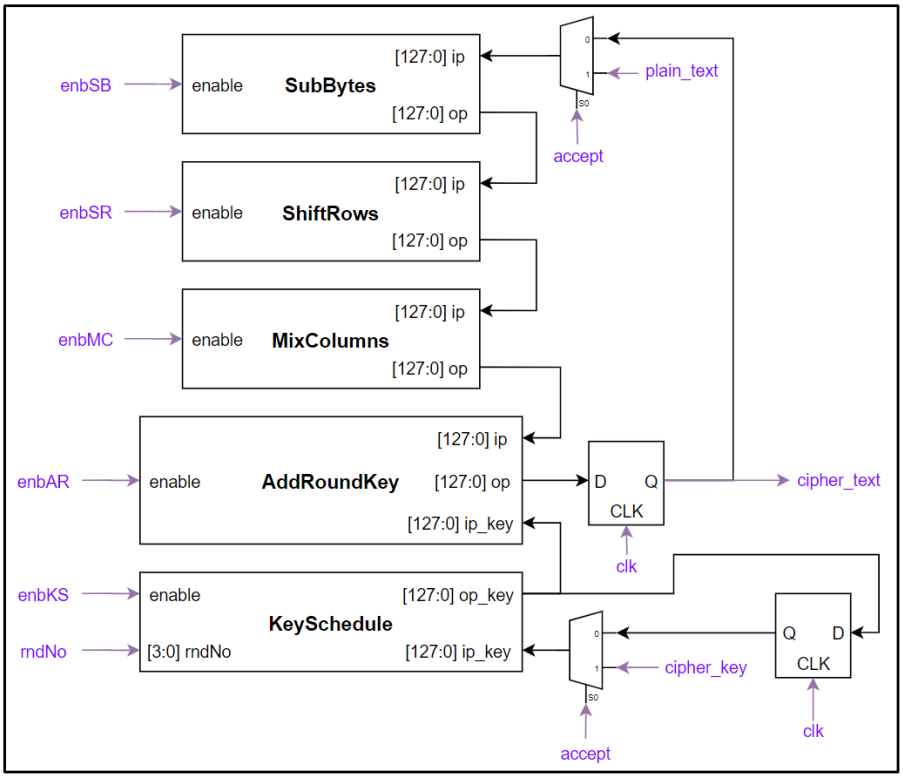
ARCHITECTURE



The figure above shows the connections between blocks in the AES core. The signals in purple are derived from the AES controller and the test bench.

REGISTERS

- AESCore.v
    1. *reg [127:0] round_key* is necessary to save the output of KeySchedule for the current round, which will then be fed into the input of KeySchedule in the next round for the generation of the round key for that round.
    2. *reg [127:0] cipher_text* is necessary to save the output of AddRoundKey for the current round, which will then be fed into the input of SubBytes in the next round for the generation of the intermediate cipher text for that round.
- AEScntx.v
    1. *reg [3:0] rndNo* is necessary to track the overall progress of the AES encryption operation. It serves as the state for our implicit FSM.
    2. *reg done* is used to signal the completion of the AES encryption operation for a given input.

SISO

- Area (μm^2) = 33309.694
- Critical path delay (ns) = 1.860
- Maximum throughput (MS/s) = $\frac{1}{2\times10^{-9}} \times \left(\frac{1}{11}\right) = 45.454$

The SISO implementation provides us with the necessary references in terms of area and maximum throughput required to compare the MIMO and N-slowing implementations below.

MIMO (N=4)

- Area (μm^2) = 135624.078
- Critical path delay (ns) = 1.955
- Maximum throughput (MS/s) = $\frac{1}{2\times10^{-9}} \times \left(\frac{4}{11}\right) = 181.818$

Essentially, parallelism (hardware replication) is employed in the MIMO implementation. We replicate the fundamental operator (i.e. AES core) N=4 times and sequentially apply all incoming inputs (when ready) to the N=4 replicas. This allows us to do N=4 computations at the same time.

Parallelism allows us to achieve an improvement in the throughput by a factor of N=4 (when compared to the SISO implementation). However, it also results in a substantial trade off in terms of area. The area increases by a factor of approximately N=4 (when compared to SISO).

We can also clearly see that the critical path delay has increased when compared to SISO. This could be due to the increase in fan-out as a result of increasing the number of replicas (i.e. AES core), which led to increased capacitive load and consequently, longer propagation delay on the corresponding driving gates in the circuit.

N-SLOWING (N=4)

- Area (μm^2) = 62210.291
- Critical path delay (ns) = 0.490
- Maximum throughput (MS/s) = $\frac{1}{5\times10^{-10}} \times \left(\frac{4}{44}\right) = 181.818$

In essence, we are performing time interleaving (N-slowing + retiming) in this implementation. N-slowing increases the number of registers in the circuit by a factor of N=4. Retiming distributes the added registers across the entire circuit in an attempt to reduce the overall clock cycle. This allows us to achieve a better result when compared to the MIMO implementation.

Time interleaving allows us to achieve an overall improvement in the throughput by a factor of N=4 (when compared to the SISO implementation) since we now have essentially N=4 channels, where each channel has the same throughput as the original SISO circuit. The trade-off in terms of area is also not as bad compared to that of the MIMO implementation. Time interleaving only increases the REG area by a factor of N=4 while parallelism increases the total circuit area by a factor of N=4.

CONCLUSION

N-slowing is the best strategy for obtaining high throughput when compared to SISO and MIMO implementations. As seen from the analysis above, N-slowing provides us with an equal throughput as MIMO despite using significantly lower area. When compared to SISO, N-slowing provides a significantly higher throughput at the expense of a much smaller increase in the area (than what MIMO was able to achieve). Hence, when we look at the throughput per area, N-slowing is definitely the superior option. In summary, N-slowing is able to provide us with a huge increase in throughput (when compared to SISO) without having to increase its area significantly (when compared to MIMO).