Arithmetic unit capable of performing modular exponentiation $C=M^E \bmod N$.

Himabindu Sajja, Vamsikrishna Paladugu, Deepak Somesula.

Objective : The objective is to Design an arithmetic unit capable of performing modular exponentiation $C=M^E \bmod N$, where M, N are arbitrary 768-bit unsigned integers, and $E = F_4 = 2^{16} + 1$.

Requirements: The arithmetic unit here performs modular exponentiation using three operands M, N and E optimized for minimum area and latency.
M,N are 768-bit operands and E is a 17-bit operand.

Applications: This modular exponentiation $C=M^E \bmod N$ is used in modern public-key ciphers like RSA, Diffe-Hellman and Elliptic curve crypto systems.

Optimization criteria: We are optimizing the design for minimum Area and minimum Latency for hardware.

Cad Tools: we are using mentor graphics version 5.4d available in ECE labs for simulation and verification of our hardware design.

Assumptions: The modular exponentiation involves modular multiplication. Here we are planning to use the Montgomery method of multiplication for modular multiplication .The basic components of Modular multiplier are carry save adders and carry propagate adder. The Carry-save adder uses full adder as its building block. Each full adder has 2 gate delays considering cin -> cout and has an area equal to 6 gates. The area and delay of carry save adder and carry propagate adder depends on the number of full adders used.

Test Plan: The VHDL code of the design is complied using ModelSim and tested using a set of 10 test vectors in a test bench.
The circuit speed is determined by calculating the delay of each building block of the design in gate levels and the circuit area is determined by counting the number of gates and maximum number of inputs to a gate.

References:
Hand book of Applied Cryptography by Alfred J Menezes , Paul C. Van Oorschot.
Cryptography and nework security by William stallings.
Computer Arithmetic Algorithms and Hardware Designs by Behrooz Parhami.
"A Scalable Architecture for Montegomery Multiplication," A. F. Tenca and C. F. Koc, Proc.CHES'99.