Hardware Implementation of 1024-bit RSA Decryption using Modular Exponentiation

Engr 697 Presentation

Group Members
Zach, Lukas, Kevin, Muhib

Zach Bachman
Electrical Engineering

Lukas Pettersson
Computer Engineering

Kevin Manago
Computer and Electrical Engineering

Muhib Noorali
Computer Engineering

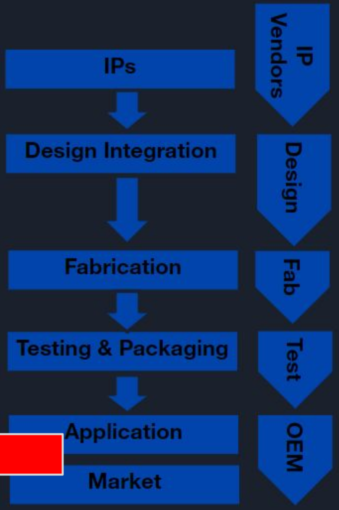
Summary of proposal

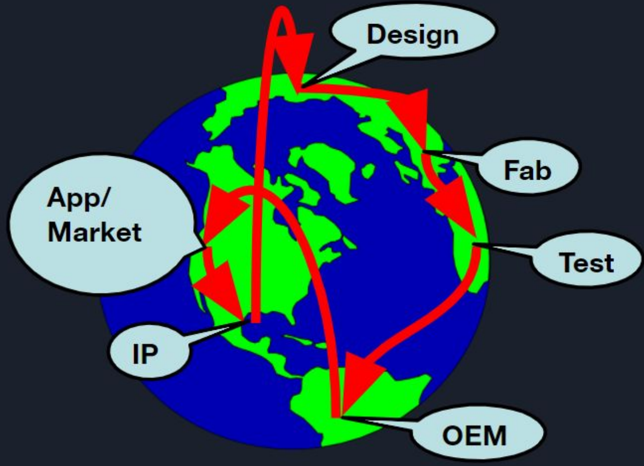
Multiple stages of Manufacturing leads to vulnerabilities

Obfuscation of Hardware to improve IP privacy

RSA Algorithm

1024-bit RSA Decryption module on an FPGA Board





Original Design Goals and Justification

Design Goal	Justification
1024-bit cipher text	Bigger cipher text = More Security
UART for input	Easier to input larger cipher values
Edge Detector	Visualization of correct output
Push it onto an FPGA	Confirm Viability

Discussion of methodologies

RSA Decryption	Montgomery Modular Multiplication	Extended Euclidean Algorithm	Non-restoring division
$M = C^d \bmod (n)$ $\Phi(n) = (p-1)(q-1)$	$n'_0 = -n^{-1} \bmod (2^w)$ $r = 2^{sw} \bmod (n)$	$ax + by = \gcd(a,b)$	
$d = e^{-1} \bmod \Phi(n)$	$t = r^2 \bmod (n)$		

Summary of results

Share screen

What’s next? What could be done with this in the future?

Pushing to Board- Optimizing

Connecting with UART- Ease of inputting values

Connecting with the Edge detector- Visual Confirmation

Upscaling- Potential for improved Security

Proper resetting of design- Bring design back to initial state and reset values

Thank you!

Zach Bachman
Electrical Engineering

Lukas Pettersson
Computer Engineering

Kevin Manago
Computer and Electrical Engineering

Muhib Noorali
Computer Engineering

QUESTIONS?