Mode-Based Obfuscation using Control-Flow Modifications

Sandhya Koteshwara, Chris H. Kim, Keshab K. Parhi

University of Minnesota
Driven to Discover℠

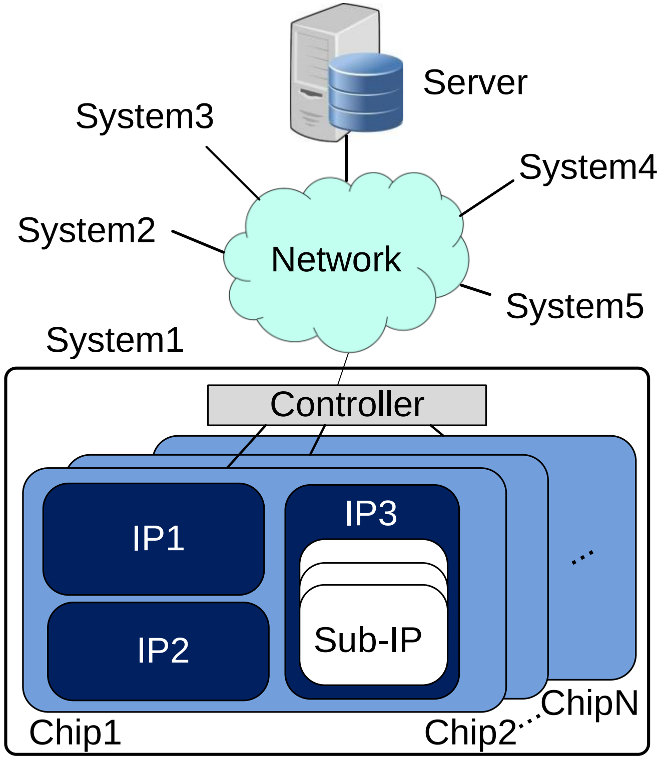
Outline

- Background and motivation
- Mode-based hardware obfuscation
- Basic concepts : Folding, Control-flow modifications
- Complete design : Obfuscated datapath and controlpath
- Mode mapping
- Analysis of obfuscated modes
- Simulation and synthesis results
- Conclusions and Future work

Background and Motivation

Background:

- Shifts of design challenges → reliability and security.
- Globalization of Integrated Circuits (ICs) and systems design and fabrication.
- Lost revenue and jobs due to counterfeit ICs.
- Hardware security is critical to national defense.



Background and Motivation

Key need:

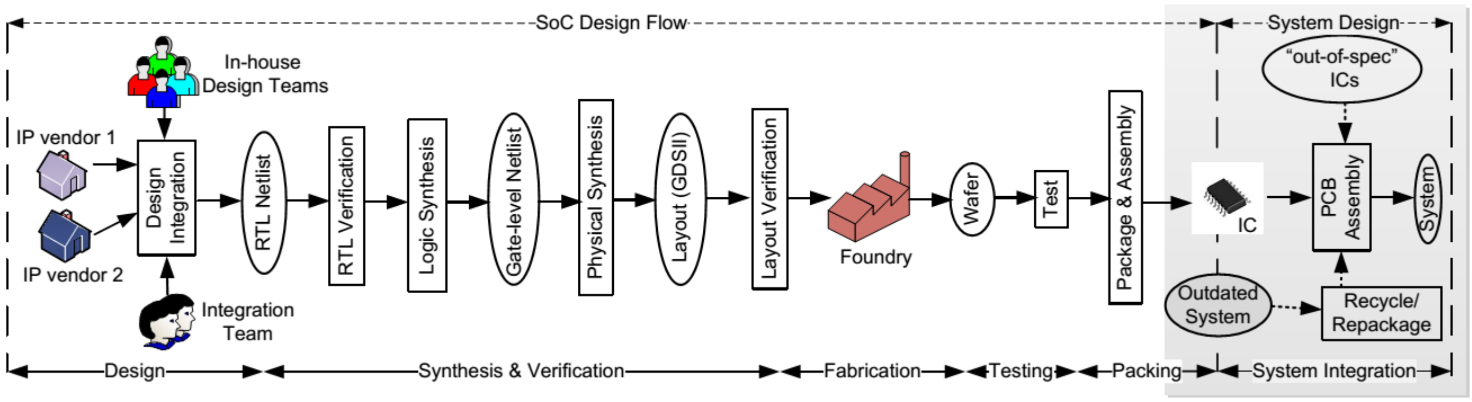
- Design of integrated circuits that can be authenticated and obfuscated for systems with deep, heterogeneous, and complex hierarchies

Application example:

- Recycled electronics products (accounting for 80 to 90 percent of counterfeit parts in circulation, according to a 2010 estimate by SMT Corp., based in Sandy Hook, Conn)*

* J. Villasenor, and M. Tehranipoor. "Chop-shop Electronics." *IEEE SPECTRUM* 50.10 (2013): 41-45

Hardware obfuscation



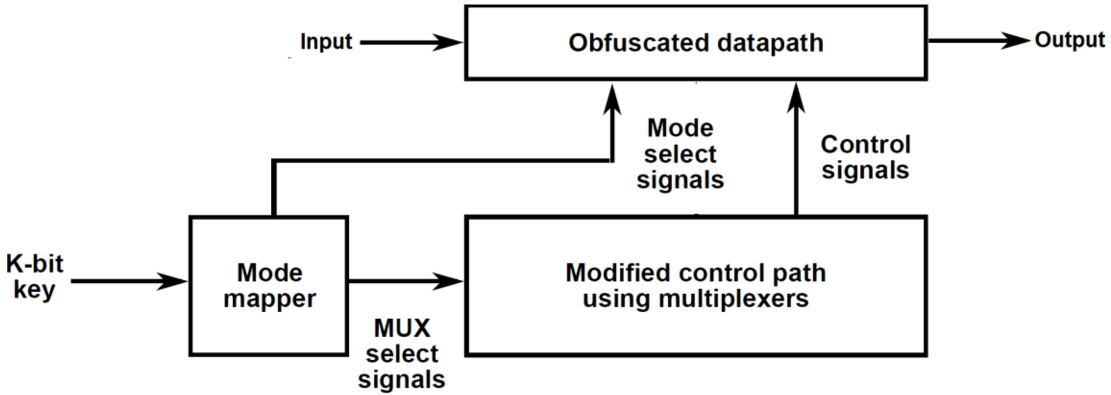
- Vulnerable to threats such as IP piracy, IC overbuilding, reverse engineering, counterfeiting, Trojans, side channel attacks etc.
- Obfuscation involves hiding functionality of a design.

Reference : M. Rostami, F. Koushanfar, J. Rajendran, and R. Karri, "Hardware security: Threat models and metrics," in Proceedings of the International Conferenceon Computer-Aided Design, 2013, pp. 819–823

Goals of obfuscation

- Address obfuscation of Digital Signal Processing (DSP) circuits.
- Highly control-driven circuits and hence obscuring control-flow is required.
- Properties of these circuits such as number of taps of filter, length of FFT which dictate performance, area, power need to be hidden.
- No existing methods of obfuscation target these specific concerns.

Mode-based obfuscation



- Design meaningful and non-meaningful modes by obfuscation of both data-path and control-path.
- Only a correct *key* applied to the system can make the circuit operate in a desired correct mode.

Reference: Y. Lao and K. K. Parhi, “Obfuscating DSP circuits via high-level transformations,” IEEE Transactions on VLSI Systems, pp. 819–830, May 2015.

Basic concepts

- Folding:

High-level transformation on circuits to create time-multiplexed architectures.

- Control-flow modifications:

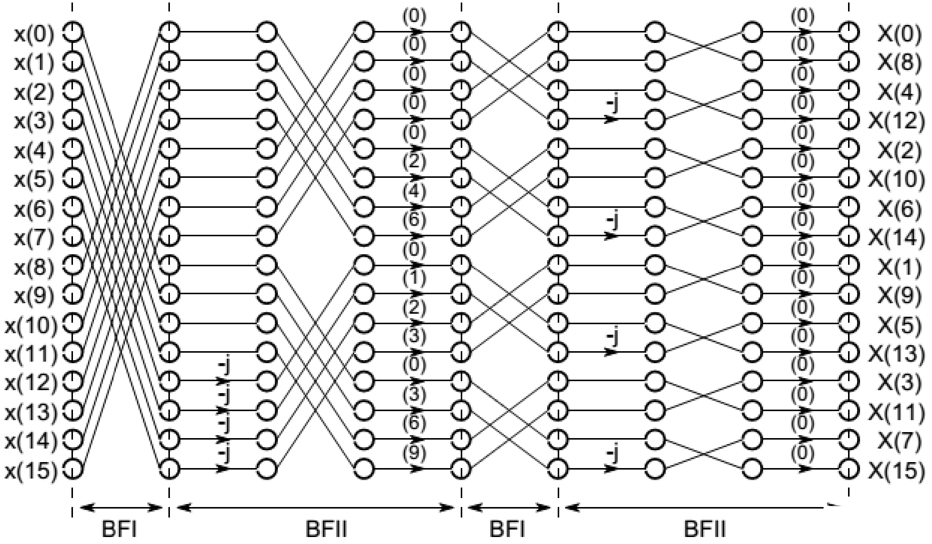
Components of the folded circuits require precise control for correct operation. Modifications to these control signals to compute incorrect outputs.

References:

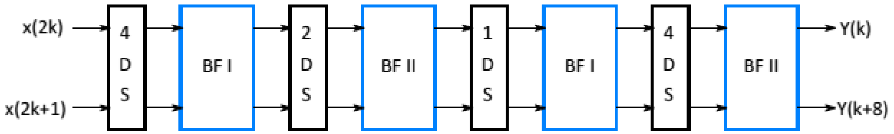
K. K. Parhi, *VLSI digital signal processing systems: design and implementation*. John Wiley & Sons, 1999.

K.K. Parhi, C.-Y. Wang, A.P. Brown, "Synthesis of control circuits in folded pipelined DSP architectures," *IEEE Journal of Solid-State Circuits*, Jan. 1992

Folding on FFT circuits



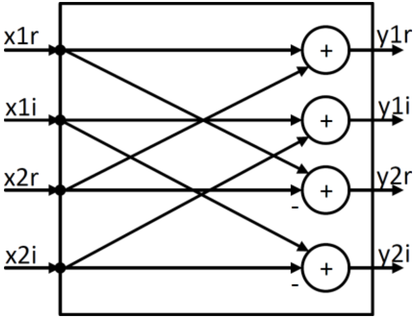
Folding



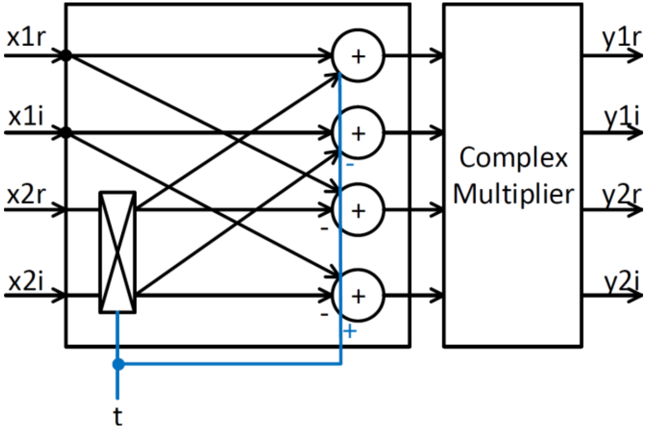
Reference: M. Ayinala, M. Brown, and K. K. Parhi, “Pipelined parallel FFT architectures via folding transformation,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 20, no. 6, pp. 1068–1081, 2012.

Components of a folded FFT

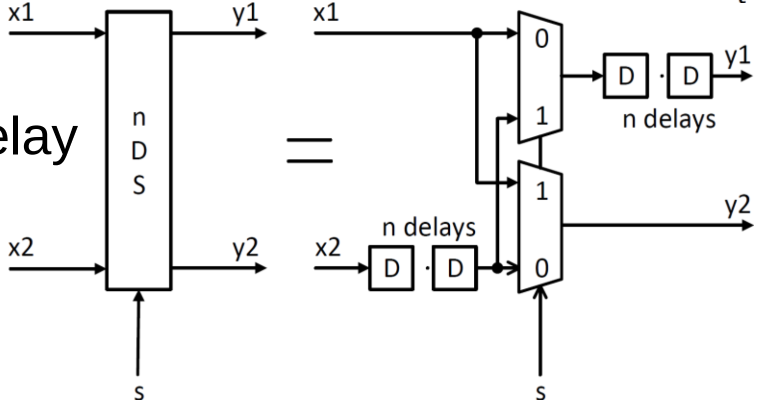
BFI



BFII



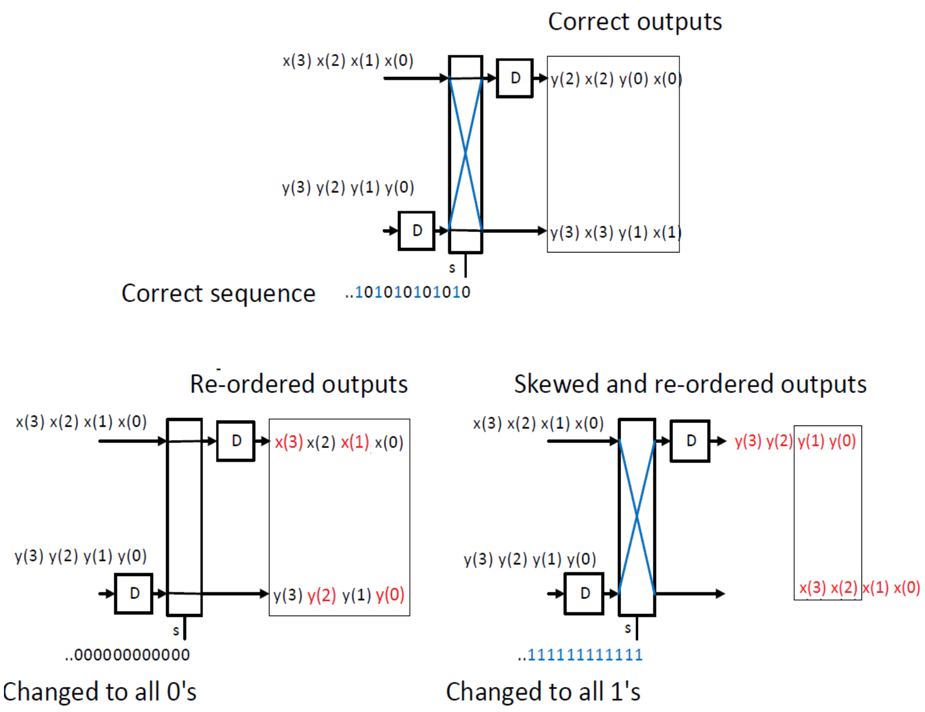
Delay-switch-delay



Control-Flow Modifications: Switches

- Any deviation from *correct Control sequence* leads to incorrect, randomized outputs.

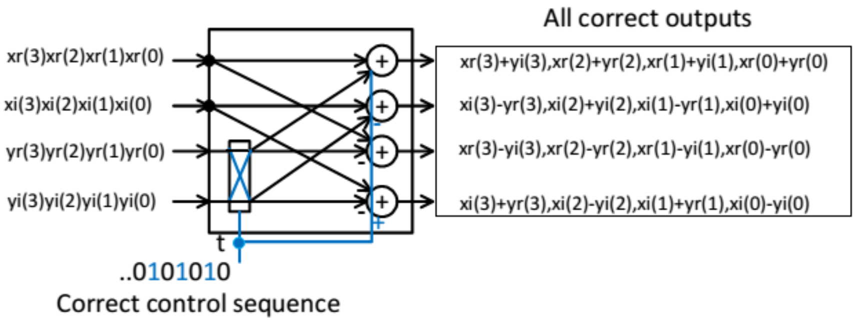
- The correct control signals *s* for these structures depend on the number of associated delays and position with respect to complete datapath.

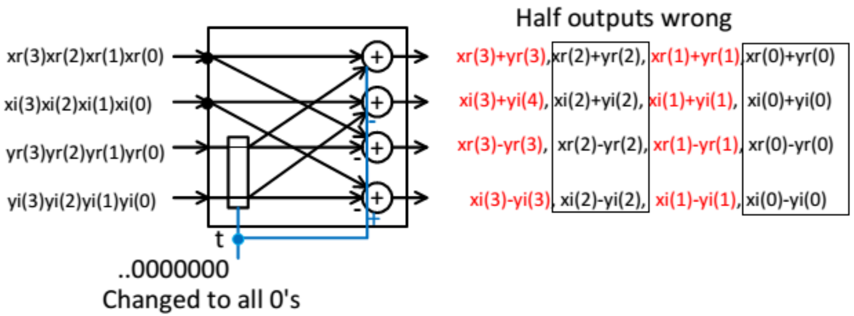


Control-Flow Modifications: Butterfly

- The control input *t* here, dictates the multiplication of inputs by a –j factor.

- Modifications to this control input leads to controlled corruption of computed outputs.

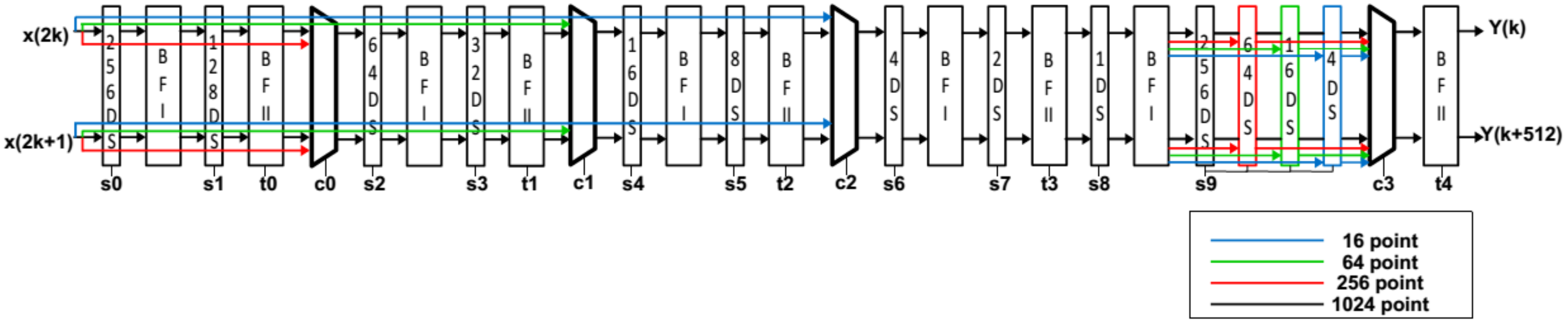




Complete Design

- An obfuscated 1024-point FFT is built as an example.
- The obfuscated datapath is built using folding transformation to produce a 16/64/256/1024-point reconfigurable design.
- The obfuscated controlpath is built using different combinations of correct and incorrect control sequences, selected using multiplexers.

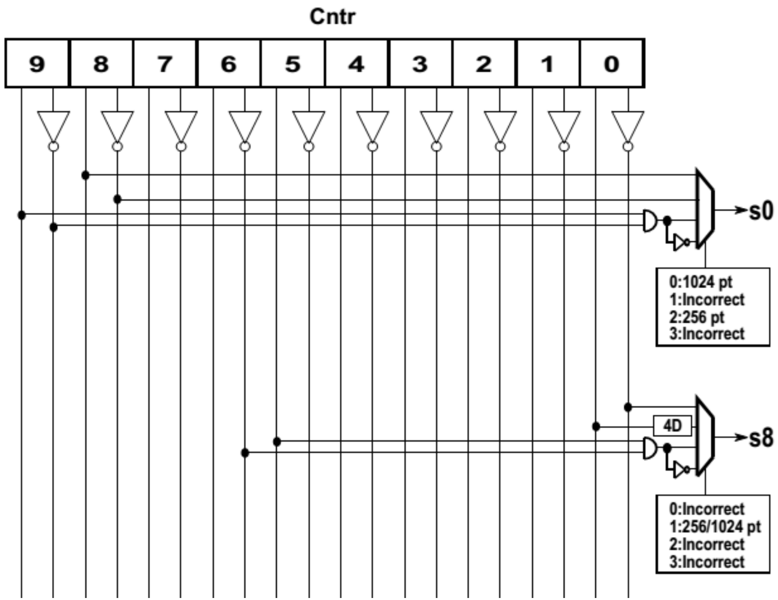
Obfuscated Datapath



- Four different architectures are combined to generate one reconfigurable architecture using only a few additional switches and wires.

Obfuscated Controlpath

- The obfuscated datapath has 15 different controls s0-s9 and t0-t4.
- Correct and incorrect control sequences for these are derived from a 10-bit counter.
- For example, s0 derived from cntr[8], !cntr[8], 0(cntr[9] & !cntr[9]), 1
- Multiplexers are used at the control outputs and select signals are generated using the *key*.



Mode Mapping

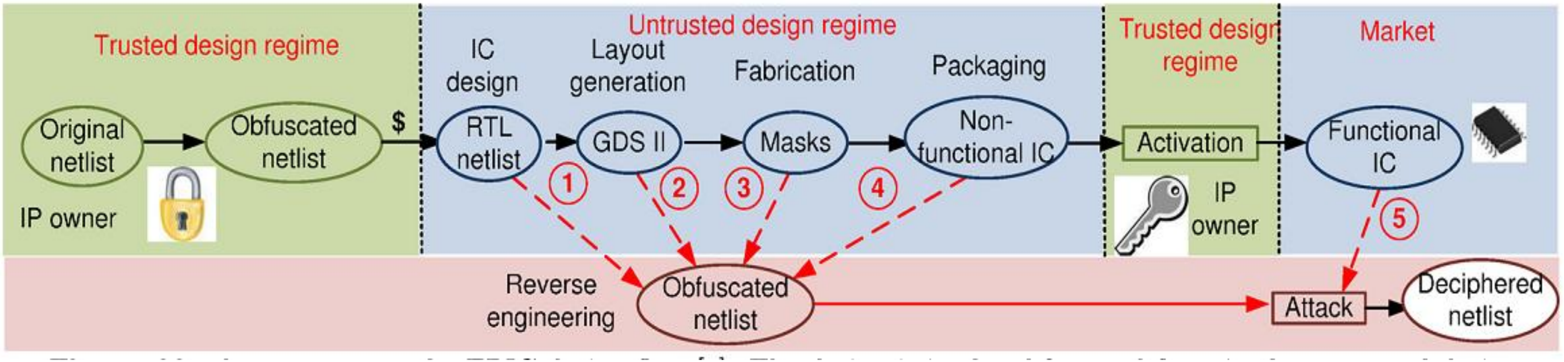
- Correct key maps to all correct control signal combinations.
- Incorrect key maps to modes which are non-meaningful or partially correct.
- For each non-meaningful mode, at least 50% deviation from correct control signal combination is used.
- Examples:

1. Non-meaningful mode with correct control signals for s0, s2, s4, s6, s8, t0 and t2 and incorrect signals for the rest.
2. Partially correct mode with 25% correct outputs by choosing incorrect controls for t0 and t4.

Analysis of Obfuscated Modes

- Attack model :

Availability of obfuscated netlist from various sources and a functional IC from the market is assumed.



Reference : J. Rajendran, Y. Pino, O. Sinanoglu, and R. Karri, “Security analysis of logic obfuscation,” in Proceedings of the 49th Annual Design Automation Conference, 2012, pp. 83–89.

Analysis of Obfuscated Modes

Obscurity of control-flow :

- For *C* different control signals in the design, each is obfuscated to degree *L* using a *L:1* mux. This gives us L^C different signal combinations.
- For example, in our implementation, L = 4, C = 15 (corresponding to s = 10 and t = 5 variables) giving 4^{15} different combinations.
- For each mode, *M* different incorrect signals can be used to create modes. M=0 implies a meaningful mode of operation. For sufficient randomness *M>C/2* is used.

Analysis of Obfuscated Modes

- For each M value, we can modify the signal in L ways using the mux. This gives us a total of $\binom{C}{M} * L^M$ modes, for every chosen value of M.

Protection of length of FFT :

- System can operate in 4 different modes (*16/64/256/1024* point FFT) and attacker has no way to know which is the desired. Partially correct modes confuse attacker.

Simulation and Synthesis Results

Design Compiler used with a 65nM technology library, clock speed 100MHz. All overhead comparisons done with respect an unobfuscated 1024-point FFT.

- First design uses different number of meaningful modes. 2 is a nominal value. (Mux size at controlpath=4, Key size=16)

Table 1: Overhead due to meaningful modes

No. of meaningful modes	Total area overhead	Total power overhead
1 (1024 point)	0.2%	0.5%
2 (256/1024 point)	8.5%	10.6%
3 (64/256/1024 point)	38%	15.5%
4 (16/64/256/1024 point)	41.5%	17.3%

Simulation and Synthesis Results

- Next, the mux size at controlpath is varied. (Meaningful modes=2, Key size=16).
- This has direct correlation to security.

Table 2: Overhead due to size of mux at controlpath

Mux size of controlpath (L)	Controlpath overhead		Total overhead	
	Area	Power	Area	Power
2	2%	0.7%	8.2%	10.1%
4	5.5%	1.7%	8.35%	10.5%
8	22%	4.2%	8.5%	11.4%
16	41%	6.7%	9.1%	12.1%

Simulation and Synthesis Results

- Finally, the key size is increased using mode mapping. (Meaningful modes=2, Mux size=4)
- Once multiplexer size at control path is set, increase in overhead is not high.

Table 3: Overhead for different key sizes

Key size (Modes)	Total Overhead	
	Area	Power
4 (16)	8.29%	10.53%
8 (256)	8.33%	10.55%
16 (65536)	8.35%	10.59%
20 (1048576)	8.42%	10.63%
28 (268435456)	8.47%	10.66%

Conclusion and Future Work

- Demonstration of mode-based method of obfuscation using FFT.
- Control-flow modifications to design modes of operation of circuit.
- Analysis of the various modes and their role in security.
- Low overheads (8% area overhead and 10% power overhead) can be achieved.
- Formal derivation of metrics of obfuscation and automation of the obfuscation technique are future areas to be explored.

Thank you!

University of Minnesota
Driven to Discover[SM]

UniversityofMinn UMNews UofMN