A Java Based Component Identification Tool for Measuring Circuit Protections

James D. Parham

J. Todd McDonald

Michael R. Grimaila

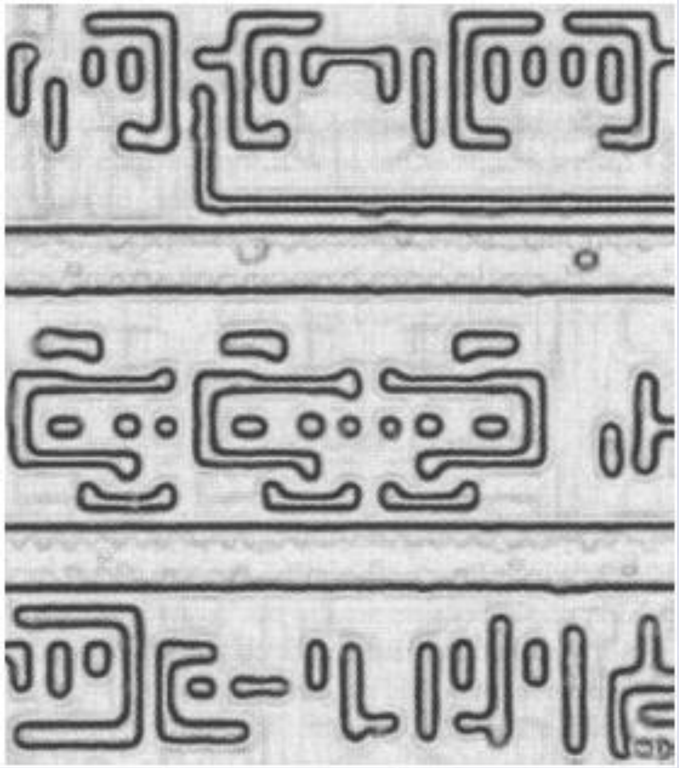
Yong C. Kim

Background – Program Protection

- Software (programs) are the 1s and 0s representing language statements able to execute on hardware processors[1]
- Circuits implemented using Field Programmable Gate Arrays (FPGAs) are essentially programs
- Embedded systems using FPGAs are able to realize circuits consisting of many different components
  - Gates
  - Controllers
  - Arithmetic Logic Units
- Protecting circuits from adversarial attack is in turn protecting programs

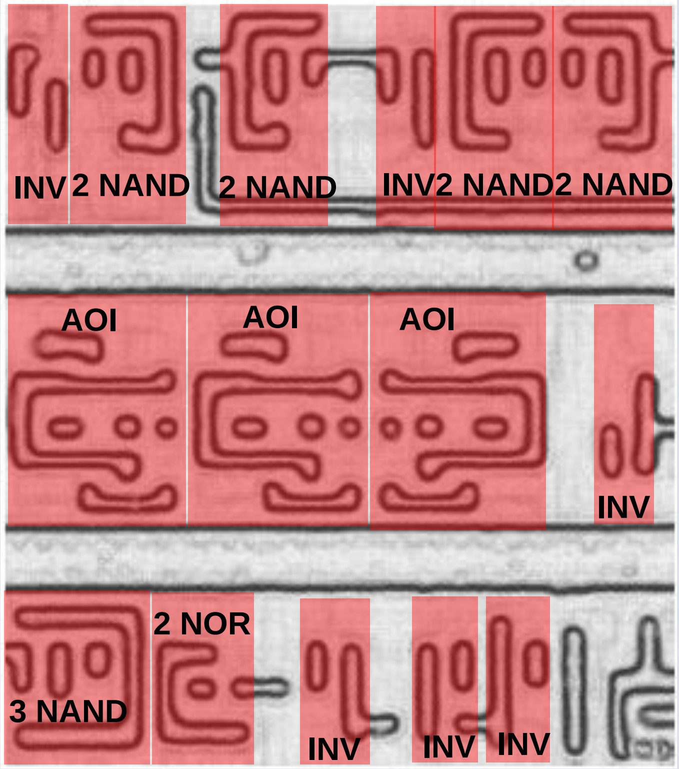
Background - Motivation

- Reverse engineering of Mifare Classic RFID tag
  - Dutch government previously invested over $2 billion in new transit ticketing system
  - Nohl et al. exposed transistors to identify gate level structures[3]
  - From gate level structures components are identifiable
  - Revealed cryptographic keys enabling free access to Dutch transit system



Background - Motivation

- Reverse engineering of Mifare Classic RFID tag
  - Dutch government previously invested over $2 billion in new transit ticketing system
  - Nohl et al. exposed transistors to identify gate level structures[3]
  - From gate level structures components are identifiable
  - Revealed cryptographic keys enabling free access to Dutch transit system

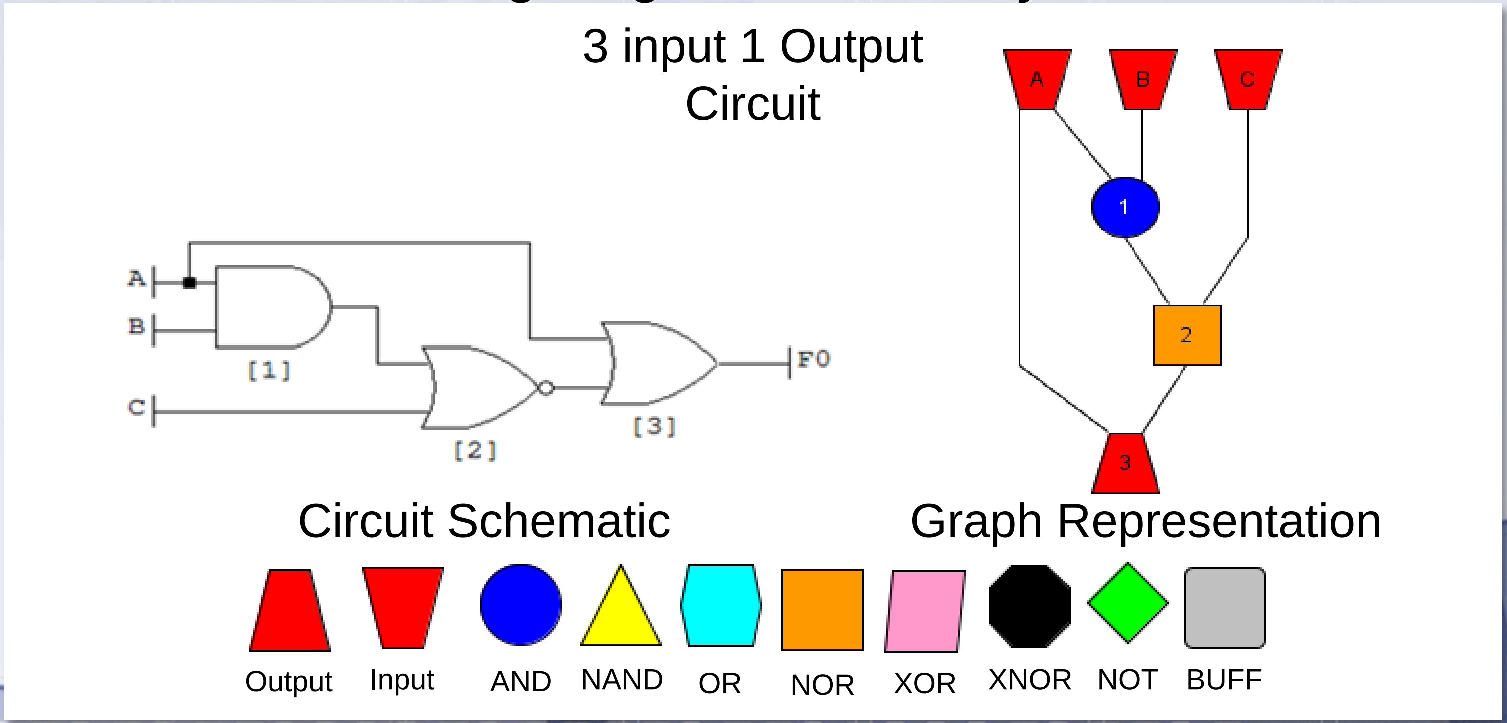


Background - Problem Statement

- 2009 DoD procurement and R&D budget over $182 billion
- An adversary with access to critical technologies may use them against the United States
  - Defeat systems that ensure national security
  - Develop equivalent systems faster and cheaper
- **We must develop a method for measuring the strength of protection applied to an individual circuit**
- Component identification tools provide measure of protection against component identification
- No component identification tool exists in our protection tool kit

Background – Modeling Circuits

- A Directed Acyclic Graph ***G*** is a triple consisting of a vertex set ***V(G)***, an edge set ***E(G)*** and a relation representing each edge with its endpoints
  - Each vertex, with its shape and color, represents a logic gate
  - Each edge represents a connection between them
  - Directed indicates edge signal flow in only one direction



3 input 1 Output Circuit

Circuit Schematic

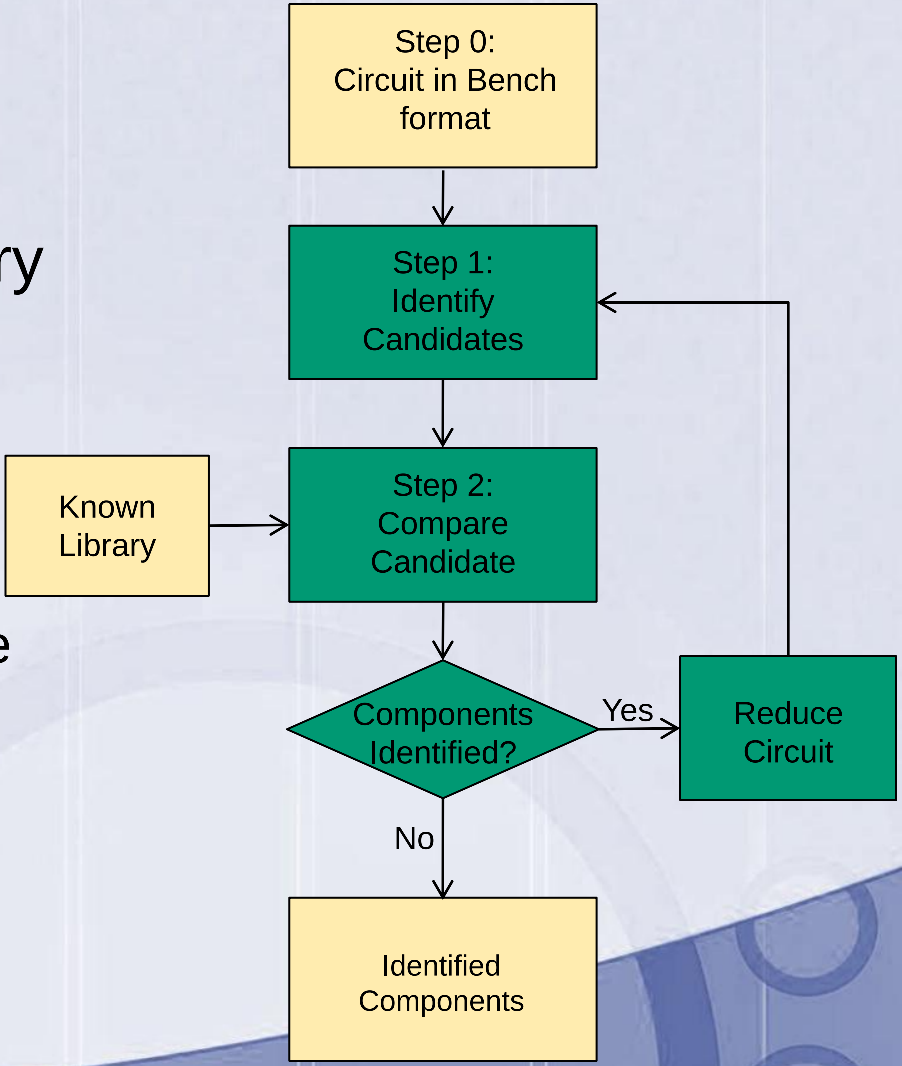
Graph Representation

Background – Candidate Enumeration

- Enumerating all candidate subcircuits is intractable for even small circuits
  - Upper bound is $n!$ where n is the number of circuit gates
- White et al. in their publication entitled, “Candidate Subcircuits For Functional Module Identification In Logic Circuits” outlines a candidate subcircuit enumeration algorithm[2]
  - Enables candidate enumeration
- No source code available for our use
- We implemented in Java using our interpretation
- Complexity ***O(n³)***

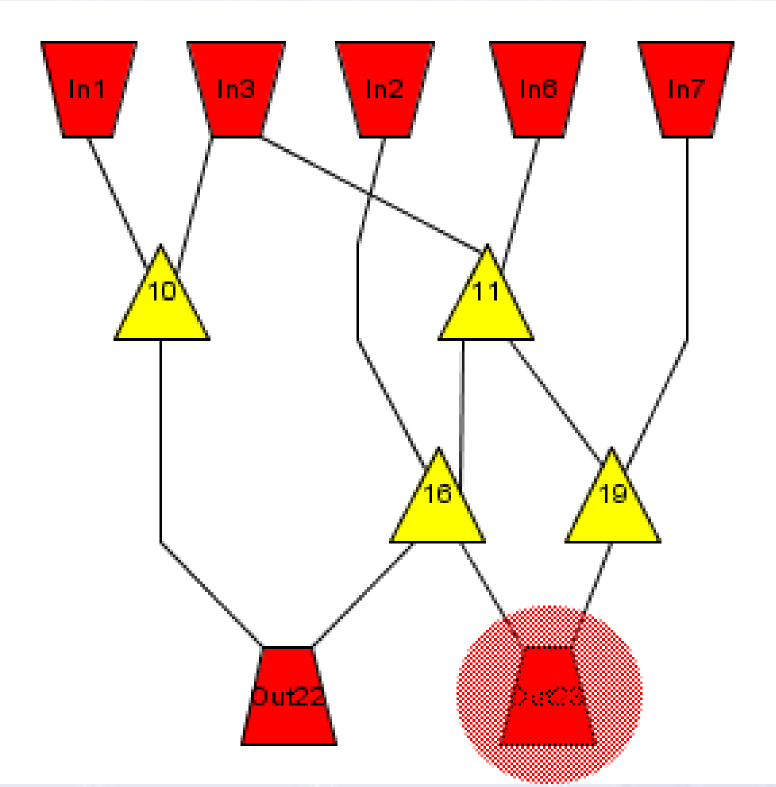
Component Identification Tool

- Provide circuit of interest to component ID tool
- Identify candidate cut sets for comparison against known library modules
- Compare candidate using truth table analysis
  - Only compare candidates with matching I/O space
  - Input and output order may require permuting for matching
- Check if any components identified
  - Yes - Circuit reduced then steps 1 and 2 repeated
  - No – Search terminates

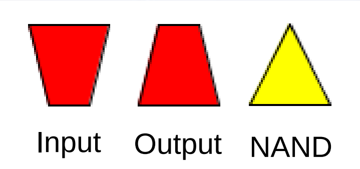


Component Identification Tool - Identify Candidates Step 1

- Enumeration begins with the highest index in the circuit. In this case Out23
- This becomes the index of the subgraph
- Vertices are “looked” at in decreasing order

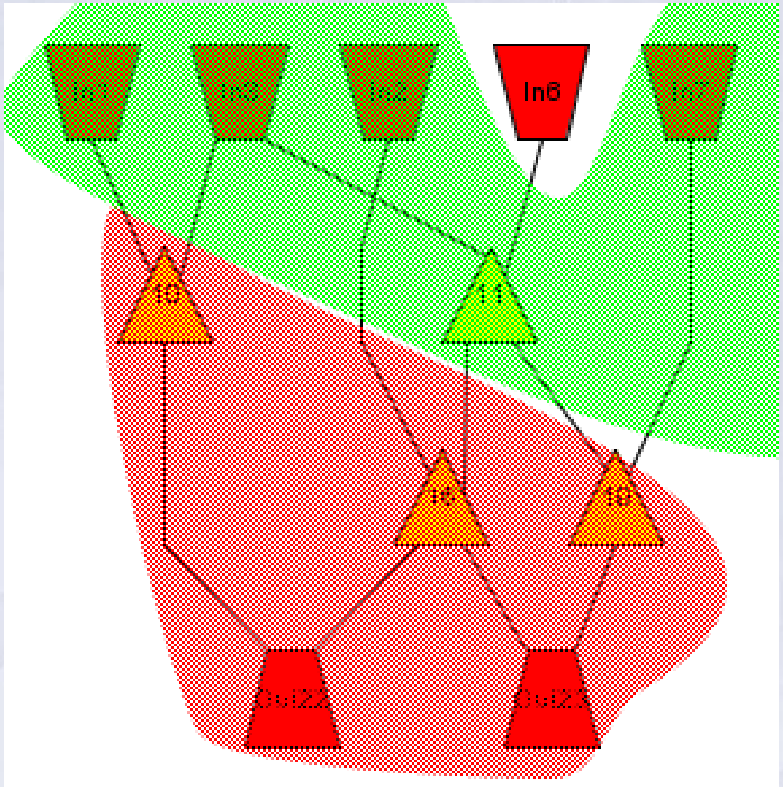


Creation Path = {23}

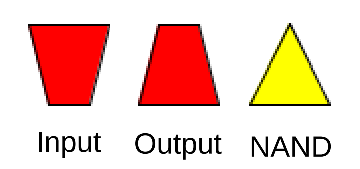


Component Identification Tool - Identify Candidates Step 1

- No rule violations
- Candidate subcircuit

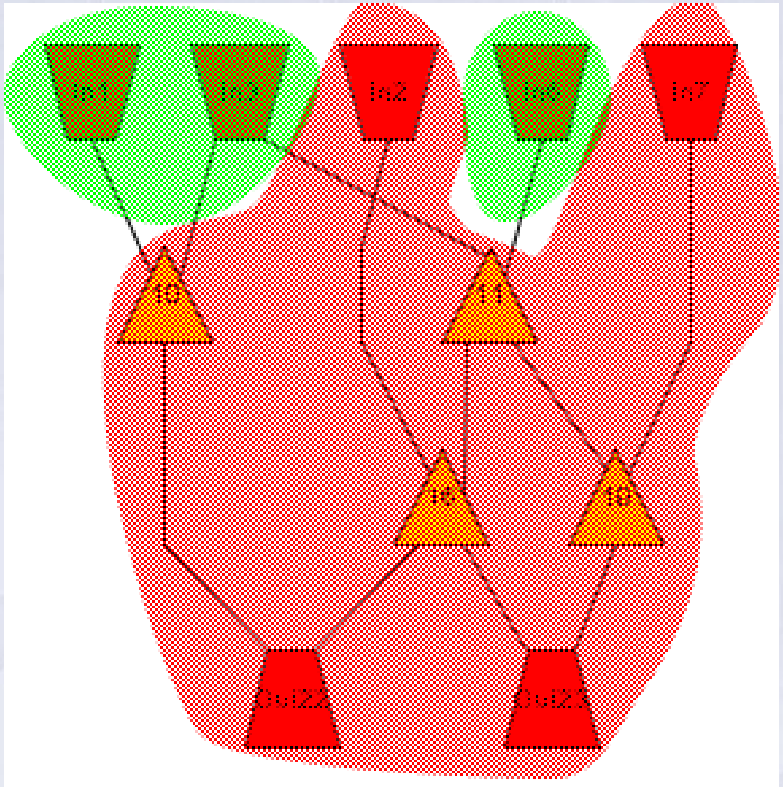


Creation Path = {23,19,16,22,10}
Reachable Frontier = {11,7,3,2,1}

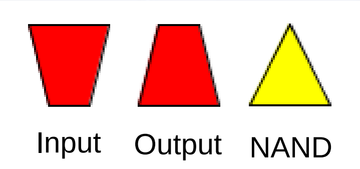


Component Identification Tool - Identify Candidates Step 1

- No rule violations
- Candidate subcircuit

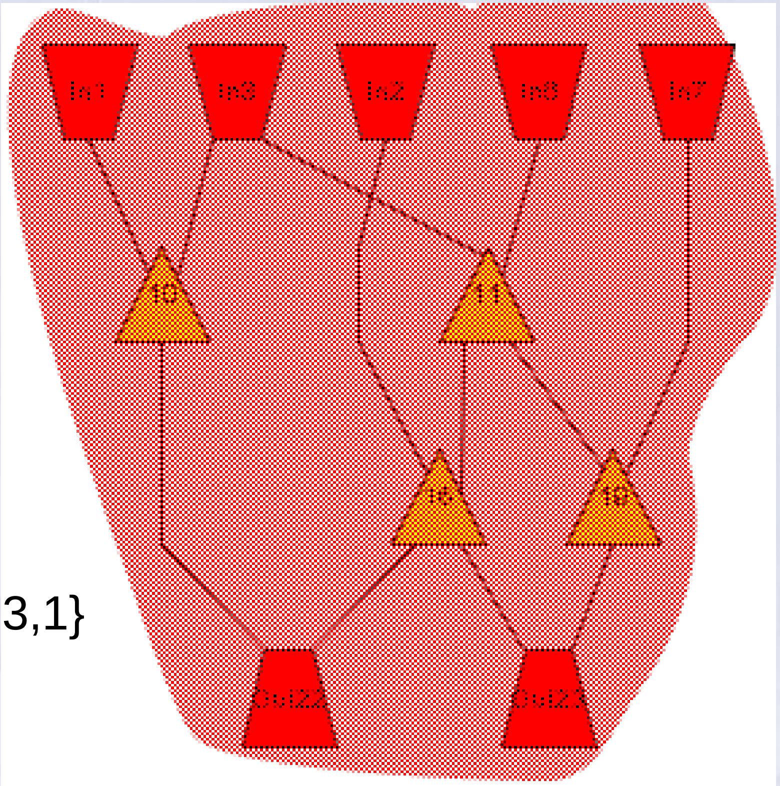


Creation Path = {23,19,16,22,10,11,7,2}
Reachable Frontier = {6,3,1}

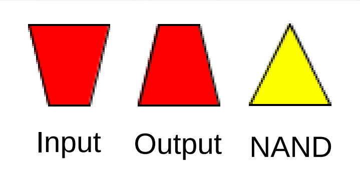


Component Identification Tool - Identify Candidates Step 1

- No rule violations
- The candidate subcircuit is the actual circuit

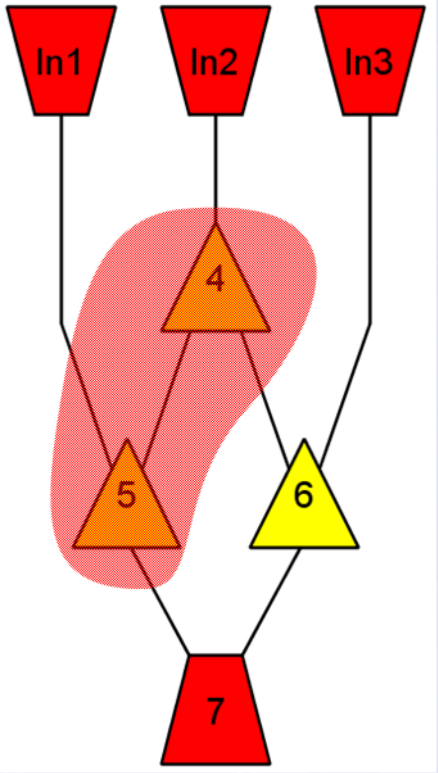


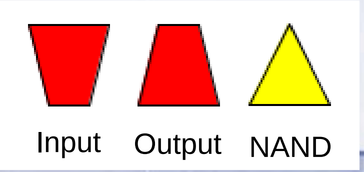
Creation Path = {23,19,16,22,10,11,7,2,6,3,1}
Reachable Frontier = {∅}



Component Identification Tool - Identify Candidates Step 1

- Example with two rule violations
- Vertex four violates rule three because only one of its successors is contained in the highlighted subgraph
- Vertex five violates rule two because only one of its predecessors is contained in the subgraph



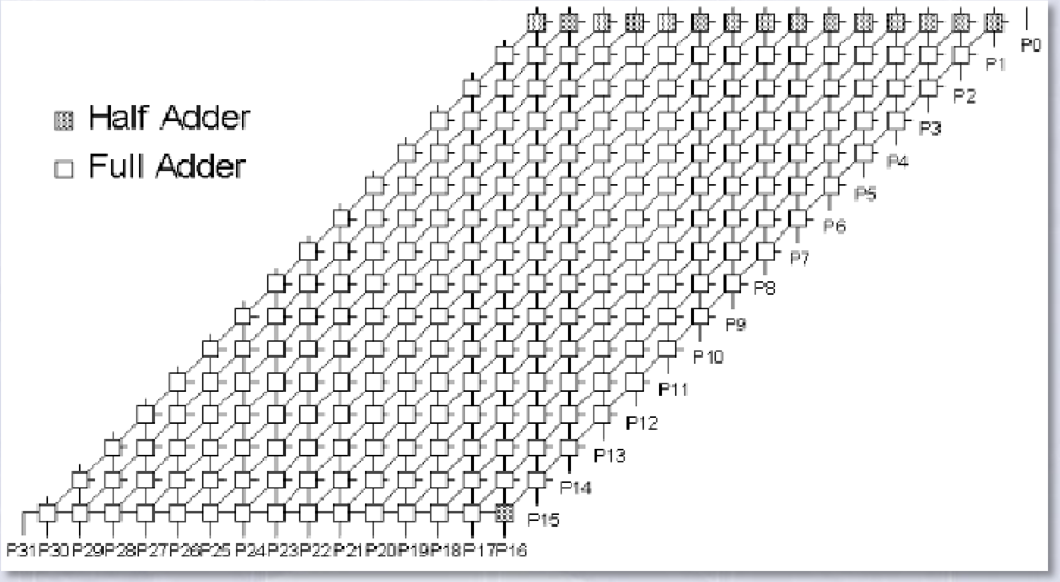


Component Identification Tool– Compare Candidates Step 2

- Created custom benchmark set containing 16 components
  - Input and output size no greater than size six
  - Used for constructing larger test circuits and verifying component comparison
- Candidate with I/O space matching component from known library compared using truth table analysis
  - Comparison runtime ***O(n!m!)*** *where* ***n*** is input size and ***m*** is output size

Component Identification Tool– ISCAS-85 16-Bit Multiplier (C6288)

- 32 input 32 output test circuit
  - Composed of 224 full adder components and 16 half adder components
  - All components identified with a single pass in 1.167 minutes using search set {12,11}

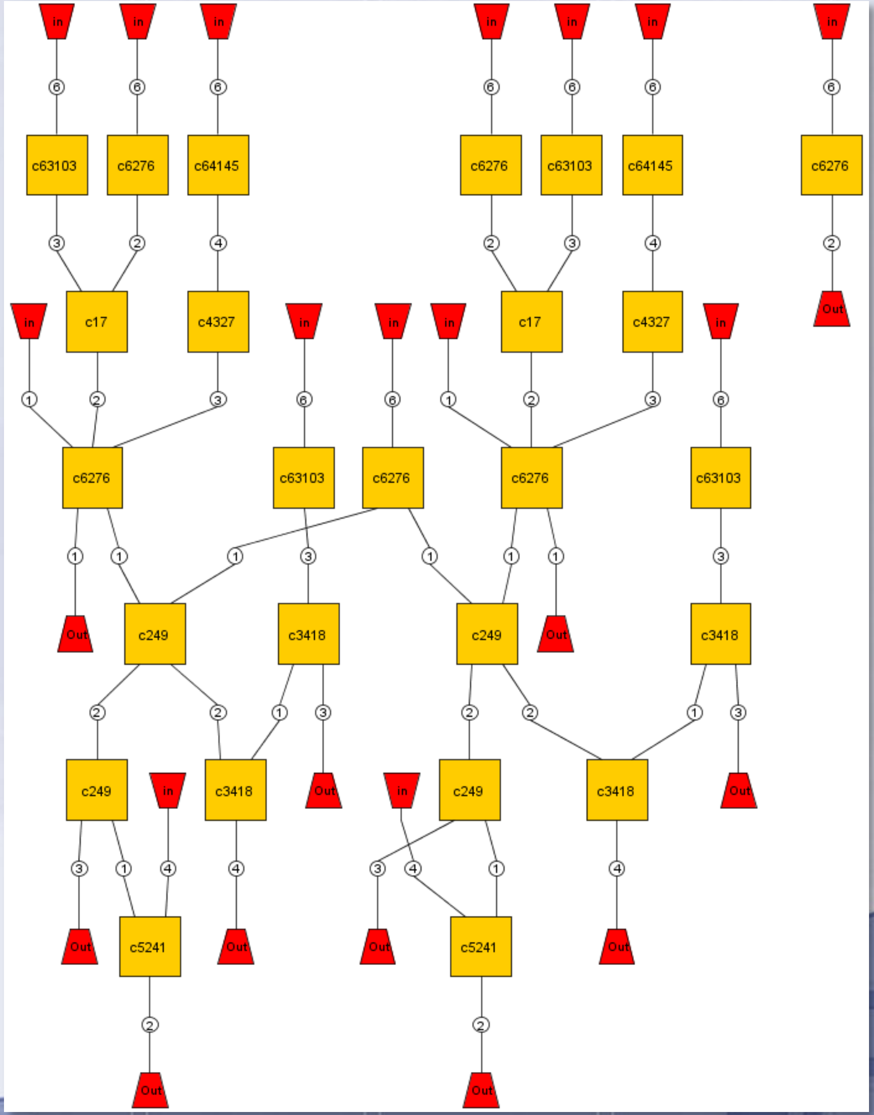


▨ Half Adder
□ Full Adder
P0
P1
P2
P3
P4
P5
P6
P7
P8
P9
P10
P11
P12
P13
P14
P15
P31 P30 P29 P28 P27 P26 P25 P24 P23 P22 P21 P20 P19 P18 P17 P16

Component Topology – Each block is either full or half adder

Component Identification Tool– Circuit with Large I/O Space

- Largest test circuit has 70 inputs 28 outputs and contain 1374 gates
  - All 26 components identified with 4 passes in 40.58 minutes using search set {145,103,76, 41,27,18,11,9}



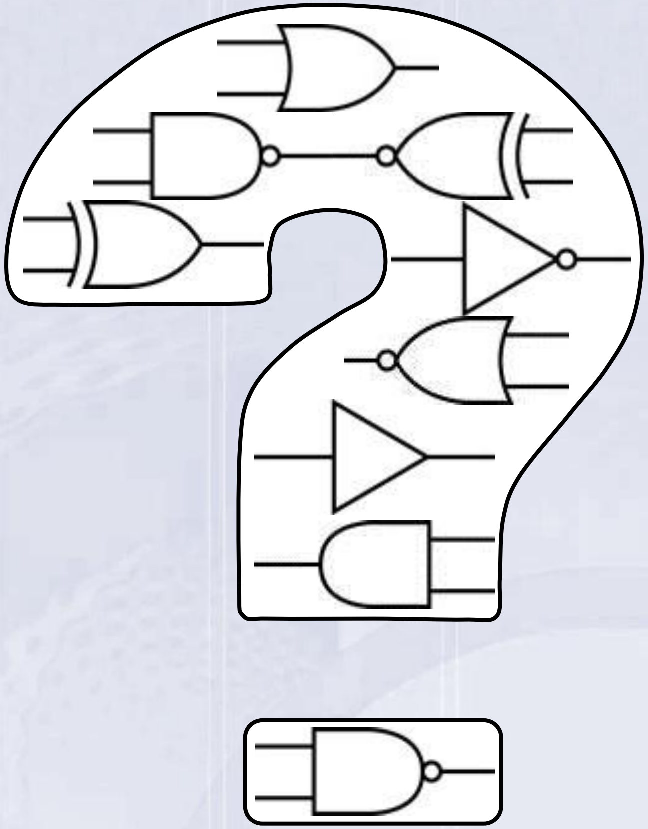
in in in in in in in
6 6 6 6 6 6 6
c63103 c6276 c64145 c6276 c63103 c64145 c6276
3 2 4 2 3 4 2
in c17 c4327 in in in c17 c4327 in Out
1 2 3 6 6 1 2 3 6
c6276 c63103 c6276 c6276 c63103
1 1 1 3 1 1 1 3
Out c249 c3418 c249 Out c3418
2 2 1 3 2 2 1 3
c249 in c3418 Out in c249 c3418 Out
3 1 4 4 3 4 1 4
Out c5241 Out Out c5241 Out
2 2
Out Out

Component Identification Tool– Measuring Circuit Protection

- Three variants of C6288 produced and component identification ran to measure circuit protection

C6288 Variant	Gate Size	Components Identified	Identification Time
Unprotected	2448	100%	18.8 Minutes
Variant One	2468	92%	18.9 Minutes
Variant Two	5784	.02%	44.5 Minutes
Variant Three	7052	0	54.3 Minutes

Questions…



Bibliography

1. Kim, Yong C. and Lt. Col. J. Todd McDonald. “Considering Software Protection for Embedded Systems”. *Crosstalk The Journal of Defense Software Engineering,* 22(6):4-8, 2009.
2. White, J. L., Wojcik, A. S., Chung, M., and Doom, T. E. 2000. Candidate subcircuits for functional module identification in logic circuits. In *Proceedings of the 10th Great Lakes Symposium on VLSI* (Chicago, Illinois, United States, March 02 - 04, 2000). GLSVLSI '00. ACM, New York, NY, 34-38. DOI= http://doi.acm.org/10.1145/330855.332575
4. Nohl, Karsten, David Evans, Starbug Starbug, and Henryk PlÄotz. \Reverse-engineering a cryptographic RFID tag". *SS'08: Proceedings of the 17th conference on Security symposium, 185{193. USENIX Association, Berkeley, CA, USA, 2008.*