Hardware Security Project

*Boyou Zhou**

*Boston Univeristy, MA

Created on Feb 8 2014, Modified on March 17, 2015

Outline

1. Plans for the ICCAD
   - More Detailed Simulation
   - Metal Pattern
2. Reviews from DAC
3. python imported files
   - scripts problems
   - Standard cells in layout
4. Prep for ICCAD
   - Previous work
   - Optical PUF
5. Gdspy works
6. gds2txt
   - Integration with python tool

Basic Thoughts

- We can have the simulation including all the metal layers, material inside of standard cells, like poly-silicon.
- Design a metal structure inside of metal layer, in *Metal1* or other metal layers in order to improve the ability to detect Hardware Trojans.

Basic Thoughts

- We can have the simulation including all the metal layers, material inside of standard cells, like poly-silicon.
- Design a metal structure inside of metal layer, in *Metal1* or other metal layers in order to improve the ability to detect Hardware Trojans.

Basic Thoughts

- We can have the simulation including all the metal layers, material inside of standard cells, like poly-silicon.
- Design a metal structure inside of metal layer, in *Metal1* or other metal layers in order to improve the ability to detect Hardware Trojans.

Outline

1. Plans for the ICCAD
   - More Detailed Simulation
   - Metal Pattern
2. Reviews from DAC
3. python imported files
   - scripts problems
   - Standard cells in layout
4. Prep for ICCAD
   - Previous work
   - Optical PUF
5. Gdspy works
6. gds2txt
   - Integration with python tool

Two possible methods to simulate the entire chip

opt 1 First, we use the rectlinear decomposition to divide all the parts inside standard cells into rectangles. And then use the def file to locate all the cells positions. At last, we combine all these informations with metal connections to one file. The problem is that lef only contains **signal pins, power and ground pins, vias, and power and ground stripes**.

opt 2 Use python to read the gds file and then use rectlinear decomposition to all the parts inside the chip.

Two possible methods to simulate the entire chip

opt 1 First, we use the rectlinear decomposition to divide all the parts inside standard cells into rectangles. And then use the def file to locate all the cells positions. At last, we combine all these informations with metal connections to one file. The problem is that lef only contains **signal pins, power and ground pins, vias, and power and ground stripes**.

opt 2 Use python to read the gds file and then use rectlinear decomposition to all the parts inside the chip.

Two possible methods to simulate the entire chip

opt 1 First, we use the rectlinear decomposition to divide all the parts inside standard cells into rectangles. And then use the def file to locate all the cells positions. At last, we combine all these informations with metal connections to one file. The problem is that lef only contains **signal pins, power and ground pins, vias, and power and ground stripes**.

opt 2 Use python to read the gds file and then use rectlinear decomposition to all the parts inside the chip.

Simple Idea on Rectlinear Decomposition

The basic algorithm has been applied to the *Metal1* and it seems to be working. The next step is to ensure the accurate of the algorithm. The first order of testing the algorithm is to use the area to verify the program.

Simple Idea on Rectlinear Decomposition

The basic algorithm has been applied to the *Metal1* and it seems to be working. The next step is to ensure the accurate of the algorithm. The first order of testing the algorithm is to use the area to verify the program.

Outline

1. Plans for the ICCAD
   - More Detailed Simulation
   - Metal Pattern
2. Reviews from DAC
3. python imported files
   - scripts problems
   - Standard cells in layout
4. Prep for ICCAD
   - Previous work
   - Optical PUF
5. Gdspy works
6. gds2txt
   - Integration with python tool

Metal Pattern

I need to look into the document to look for the specification of filler cell positions.

Metal Pattern

I need to look into the document to look for the specification of filler cell positions.

Feb 17th, 2015

The first reviewer, 2

- 16 did such imaging for temperature measurements and Trojan detection. The drawback here and in 16 is that the imaging must be performed on unpackaged die. In BISA technique (HOST 13 and TCAD 14), they do not need unpacked ICs.
- If Trojans insertion is performed by replacing standard cells in the design with smaller custom cells, this will not impact the filler cells or watermark so such Trojans are undetectable.
- Another limitation is the speed of the technique. ?The authors claim a few hours would be required to test an IC and they think this is an advantage.
- The authors use FDTD simulation for most experiments, but do not describe the software or its limitations.
- Detection rates vs. SNR are given to show the accuracy of the approach, but the authors never mention what an expected SNR might be so the results are not as easy to interpret.

The first reviewer, 2

- 16 did such imaging for temperature measurements and Trojan detection. The drawback here and in 16 is that the imaging must be performed on unpackaged die. In BISA technique (HOST 13 and TCAD 14), they do not need unpacked ICs.
- If Trojans insertion is performed by replacing standard cells in the design with smaller custom cells, this will not impact the filler cells or watermark so such Trojans are undetectable.
- Another limitation is the speed of the technique. ?The authors claim a few hours would be required to test an IC and they think this is an advantage.
- The authors use FDTD simulation for most experiments, but do not describe the software or its limitations.
- Detection rates vs. SNR are given to show the accuracy of the approach, but the authors never mention what an expected SNR might be so the results are not as easy to interpret.

The first reviewer, 2

- 16 did such imaging for temperature measurements and Trojan detection. The drawback here and in 16 is that the imaging must be performed on unpackaged die. In BISA technique (HOST 13 and TCAD 14), they do not need unpacked ICs.
- If Trojans insertion is performed by replacing standard cells in the design with smaller custom cells, this will not impact the filler cells or watermark so such Trojans are undetectable.
- Another limitation is the speed of the technique. ?The authors claim a few hours would be required to test an IC and they think this is an advantage.
- The authors use FDTD simulation for most experiments, but do not describe the software or its limitations.
- Detection rates vs. SNR are given to show the accuracy of the approach, but the authors never mention what an expected SNR might be so the results are not as easy to interpret.

The first reviewer, 2

- 16 did such imaging for temperature measurements and Trojan detection. The drawback here and in 16 is that the imaging must be performed on unpackaged die. In BISA technique (HOST 13 and TCAD 14), they do not need unpacked ICs.
- If Trojans insertion is performed by replacing standard cells in the design with smaller custom cells, this will not impact the filler cells or watermark so such Trojans are undetectable.
- Another limitation is the speed of the technique. ?The authors claim a few hours would be required to test an IC and they think this is an advantage.
- The authors use FDTD simulation for most experiments, but do not describe the software or its limitations.
- Detection rates vs. SNR are given to show the accuracy of the approach, but the authors never mention what an expected SNR might be so the results are not as easy to interpret.

The first reviewer, 2

- 16 did such imaging for temperature measurements and Trojan detection. The drawback here and in 16 is that the imaging must be performed on unpackaged die. In BISA technique (HOST 13 and TCAD 14), they do not need unpacked ICs.
- If Trojans insertion is performed by replacing standard cells in the design with smaller custom cells, this will not impact the filler cells or watermark so such Trojans are undetectable.
- Another limitation is the speed of the technique. ?The authors claim a few hours would be required to test an IC and they think this is an advantage.
- The authors use FDTD simulation for most experiments, but do not describe the software or its limitations.
- Detection rates vs. SNR are given to show the accuracy of the approach, but the authors never mention what an expected SNR might be so the results are not as easy to interpret.

The second reviewer, 4

- It would be better if there are details regarding design overhead and evaluation results using fabricated chips.
- How to engineer and where to insert the fill cells?
- Any rules to choose the location and number of embedded cells?
- Adding maximal metal into the cell which doesn't violate the design rules doesn't necessarily lead to no violations for the entire circuit.
- Inserting fill cells into blank area is easy, but should be careful if among certain functional cells, because it may interrupt the metal connections among functional cells.
- Too many fill cells with high metal density in a certain area may cause polishing problem during fabrication, not high enough metal density may not create significant reflectance compare to its adjacent area in real chip.

The second reviewer, 4

- It would be better if there are details regarding design overhead and evaluation results using fabricated chips.
- How to engineer and where to insert the fill cells?
- Any rules to choose the location and number of embedded cells?
- Adding maximal metal into the cell which doesn't violate the design rules doesn't necessarily lead to no violations for the entire circuit.
- Inserting fill cells into blank area is easy, but should be careful if among certain functional cells, because it may interrupt the metal connections among functional cells.
- Too many fill cells with high metal density in a certain area may cause polishing problem during fabrication, not high enough metal density may not create significant reflectance compare to its adjacent area in real chip.

The second reviewer, 4

- It would be better if there are details regarding design overhead and evaluation results using fabricated chips.
- How to engineer and where to insert the fill cells?
- Any rules to choose the location and number of embedded cells?
- Adding maximal metal into the cell which doesn't violate the design rules doesn't necessarily lead to no violations for the entire circuit.
- Inserting fill cells into blank area is easy, but should be careful if among certain functional cells, because it may interrupt the metal connections among functional cells.
- Too many fill cells with high metal density in a certain area may cause polishing problem during fabrication, not high enough metal density may not create significant reflectance compare to its adjacent area in real chip.

The second reviewer, 4

- It would be better if there are details regarding design overhead and evaluation results using fabricated chips.
- How to engineer and where to insert the fill cells?
- Any rules to choose the location and number of embedded cells?
- Adding maximal metal into the cell which doesn't violate the design rules doesn't necessarily lead to no violations for the entire circuit.
- Inserting fill cells into blank area is easy, but should be careful if among certain functional cells, because it may interrupt the metal connections among functional cells.
- Too many fill cells with high metal density in a certain area may cause polishing problem during fabrication, not high enough metal density may not create significant reflectance compare to its adjacent area in real chip.

The second reviewer, 4

- It would be better if there are details regarding design overhead and evaluation results using fabricated chips.
- How to engineer and where to insert the fill cells?
- Any rules to choose the location and number of embedded cells?
- Adding maximal metal into the cell which doesn't violate the design rules doesn't necessarily lead to no violations for the entire circuit.
- Inserting fill cells into blank area is easy, but should be careful if among certain functional cells, because it may interrupt the metal connections among functional cells.
- Too many fill cells with high metal density in a certain area may cause polishing problem during fabrication, not high enough metal density may not create significant reflectance compare to its adjacent area in real chip.

The second reviewer, 4

- It would be better if there are details regarding design overhead and evaluation results using fabricated chips.
- How to engineer and where to insert the fill cells?
- Any rules to choose the location and number of embedded cells?
- Adding maximal metal into the cell which doesn't violate the design rules doesn't necessarily lead to no violations for the entire circuit.
- Inserting fill cells into blank area is easy, but should be careful if among certain functional cells, because it may interrupt the metal connections among functional cells.
- Too many fill cells with high metal density in a certain area may cause polishing problem during fabrication, not high enough metal density may not create significant reflectance compare to its adjacent area in real chip.

The third reviewer, 4

- The big issue is that the work appears to be theoretical with no real experimental data to validate these claims. The image quality and the speed may change when applied to real designs.

The fourth reviewer, 2

- Whether there is any process variation to this approach and how much. Second, a design to be checked for Trojan must be compared with a golden design for the pattern (watermark). How to obtain this golden design?
- The 2% leakage overhead seems to be small, but the 0.1% of the total area is still large for some Trojan. (for a chip with 2 million gates, this 0.1% means 2000 gates!) It will be interesting and might be challenging to reduce this 0.1% to a much smaller number, maybe 10^{-6} or smaller.

The fourth reviewer, 2

- Whether there is any process variation to this approach and how much. Second, a design to be checked for Trojan must be compared with a golden design for the pattern (watermark). How to obtain this golden design?
- The 2% leakage overhead seems to be small, but the 0.1% of the total area is still large for some Trojan. (for a chip with 2 million gates, this 0.1% means 2000 gates!) It will be interesting and might be challenging to reduce this 0.1% to a much smaller number, maybe 10^{-6} or smaller.

The fifth reviewer, 4

- The limitations and/or comparison to existing like techniques are not exactly described. For example, I am not sure how much better detection this paper achieves compared to the imaging references below.
  while the paper cites a number of marginally related work, e.g., [2][13][15], the authors miss the important related work on using the optical and thermal imaging to detect hardware Trojans, in particular:
  - P. Song et al., "MARVEL - Malicious Alteration Recognition and Verification by Emission of Light" HOST 2011
  - F. Stellari et al., "Functional block extraction for hardware security detection using time-integrated and time-resolved emission measurements," VTS 2014?
  - Nowroz et al, "Novel Techniques for High-Sensitivity Hardware Trojan Detection Using Thermal and Power Maps", TCAD 2014

The fifth reviewer, 4

- The limitations and/or comparison to existing like techniques are not exactly described. For example, I am not sure how much better detection this paper achieves compared to the imaging references below.
  while the paper cites a number of marginally related work, e.g., [2][13][15], the authors miss the important related work on using the optical and thermal imaging to detect hardware Trojans, in particular:
  - P. Song et al., "MARVEL - Malicious Alteration Recognition and Verification by Emission of Light" HOST 2011
  - F. Stellari et al., "Functional block extraction for hardware security detection using time-integrated and time-resolved emission measurements," VTS 2014?
  - Nowroz et al, "Novel Techniques for High-Sensitivity Hardware Trojan Detection Using Thermal and Power Maps", TCAD 2014

The fifth reviewer, 4

- The limitations and/or comparison to existing like techniques are not exactly described. For example, I am not sure how much better detection this paper achieves compared to the imaging references below.
  while the paper cites a number of marginally related work, e.g., [2][13][15], the authors miss the important related work on using the optical and thermal imaging to detect hardware Trojans, in particular:
  - P. Song et al., "MARVEL - Malicious Alteration Recognition and Verification by Emission of Light" HOST 2011
  - F. Stellari et al., "Functional block extraction for hardware security detection using time-integrated and time-resolved emission measurements," VTS 2014?
  - Nowroz et al, "Novel Techniques for High-Sensitivity Hardware Trojan Detection Using Thermal and Power Maps", TCAD 2014

The fifth reviewer, 4

- The limitations and/or comparison to existing like techniques are not exactly described. For example, I am not sure how much better detection this paper achieves compared to the imaging references below.
  while the paper cites a number of marginally related work, e.g., [2][13][15], the authors miss the important related work on using the optical and thermal imaging to detect hardware Trojans, in particular:
  - P. Song et al., "MARVEL - Malicious Alteration Recognition and Verification by Emission of Light" HOST 2011
  - F. Stellari et al., "Functional block extraction for hardware security detection using time-integrated and time-resolved emission measurements," VTS 2014?
  - Nowroz et al, "Novel Techniques for High-Sensitivity Hardware Trojan Detection Using Thermal and Power Maps", TCAD 2014

The sixth reviewer, 3

- By analyzing multiple optical responses of Trojan free chips, can the attacker identify the location of the fill cells?
- I can?t see how the author can differentiate between figure 2.b and 3.b by looking at their images only as he/she claims.
- questionable claim of capability to deal with process variations
- need comparison against TJ Watson’s PICO technique

The sixth reviewer, 3

- By analyzing multiple optical responses of Trojan free chips, can the attacker identify the location of the fill cells?
- I can?t see how the author can differentiate between figure 2.b and 3.b by looking at their images only as he/she claims.
- questionable claim of capability to deal with process variations
- need comparison against TJ Watson’s PICO technique

The sixth reviewer, 3

- By analyzing multiple optical responses of Trojan free chips, can the attacker identify the location of the fill cells?
- I can?t see how the author can differentiate between figure 2.b and 3.b by looking at their images only as he/she claims.
- questionable claim of capability to deal with process variations
- need comparison against TJ Watson’s PICO technique

The sixth reviewer, 3

- By analyzing multiple optical responses of Trojan free chips, can the attacker identify the location of the fill cells?
- I can?t see how the author can differentiate between figure 2.b and 3.b by looking at their images only as he/she claims.
- questionable claim of capability to deal with process variations
- need comparison against TJ Watson's PICO technique

The seventh reviewer, 5

- Will modifying the fill cells affect the analog characteristics of the chip?
- For Replacement_Type1 and Replacement_Type2 tests, how many gates/cells were changed? What is the lower limit on the number of fill cells/gates that can be modified and still be detected?
- For Figure 4, how are these detection error rates calculated? Is this empirical or can you guarantee no false positives/negatives at a threshold of 0.65?

The seventh reviewer, 5

- Will modifying the fill cells affect the analog characteristics of the chip?
- For Replacement_Type1 and Replacement_Type2 tests, how many gates/cells were changed? What is the lower limit on the number of fill cells/gates that can be modified and still be detected?
- For Figure 4, how are these detection error rates calculated? Is this empirical or can you guarantee no false positives/negatives at a threshold of 0.65?

The seventh reviewer, 5

- Will modifying the fill cells affect the analog characteristics of the chip?
- For Replacement_Type1 and Replacement_Type2 tests, how many gates/cells were changed? What is the lower limit on the number of fill cells/gates that can be modified and still be detected?
- For Figure 4, how are these detection error rates calculated? Is this empirical or can you guarantee no false positives/negatives at a threshold of 0.65?

Outline

1. Plans for the ICCAD
   - More Detailed Simulation
   - Metal Pattern
2. Reviews from DAC
3. python imported files
   - scripts problems
   - Standard cells in layout
4. Prep for ICCAD
   - Previous work
   - Optical PUF
5. Gdspy works
6. gds2txt
   - Integration with python tool

info in gds

- gds file does not contain info of metal structures inside a standard cell. It contains the positions of the cells, types of the cells, and all the vias.
- The info inside gds file is listed below.

```
CellReference("XNOR2_X1", (26.98, 23.1), 0.0, None, True),
CellReference("XNOR2_X1", (22.8, 20.3), 180.0, None, False),
CellReference("XNOR2_X1", (20.33, 17.5), 180.0, None, True),
CellReference("XNOR2_X1", (22.42, 25.9), 0.0, None, True),
CellReference("XNOR2_X1", (21.47, 25.9), 180.0, None, True),
CellReference("XNOR2_X1", (21.47, 17.5), 0.0, None, False),
CellReference("XNOR2_X1", (28.31, 20.3), 0.0, None, False),
CellReference("top_VIA2", (11.385, 16.1), None, None, False),
CellReference("top_VIA8", (11.385, 16.1), None, None, False),
CellReference("top_VIA9", (11.385, 16.1), None, None, False),
```

Outline

1. Plans for the ICCAD
   - More Detailed Simulation
   - Metal Pattern
2. Reviews from DAC
3. python imported files
   - scripts problems
   - Standard cells in layout
4. Prep for ICCAD
   - Previous work
   - Optical PUF
5. Gdspy works
6. gds2txt
   - Integration with python tool

info from encounter

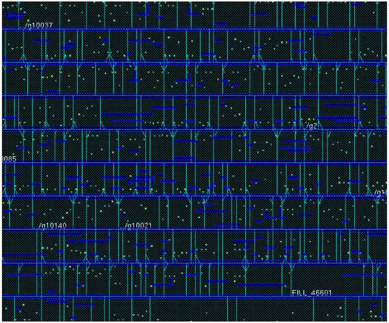


Figure : AES from DAC paper

info from encounter

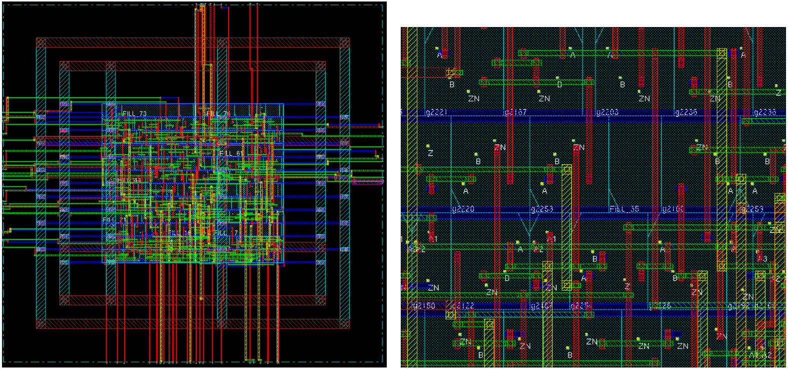


Figure : A much simpler example[1]

[1] Sheng Wei et al. "Hardware Trojan horse benchmark via optimal creation and placement of malicious circuitry". In: *Proceedings of the 49th Annual Design Automation Conference*. ACM. 2012, pp. 90–95.

Outline

1. Plans for the ICCAD
   - More Detailed Simulation
   - Metal Pattern
2. Reviews from DAC
3. python imported files
   - scripts problems
   - Standard cells in layout
4. Prep for ICCAD
   - Previous work
   - Optical PUF
5. Gdspy works
6. gds2txt
   - Integration with python tool

IC counterfeiting defs

- IC counterfeiting category[2]
  - unauthorized copy
  - not conform to original design, model and/or performance standards
  - off specs, defective or used design sold as new
  - incorrect or false markings and/or documents

[2] Ujjwal Guin, Daniel DiMase, and Mohammad Tehranipoor. "Counterfeit integrated circuits: detection, avoidance, and the challenges ahead". In: *Journal of Electronic Testing* 30.1 (2014), pp. 9–23.

FPGA IP protection

Main methods for FPGA IP protection.[3]

- encryption Commercially available encryption-based techniques are limited to single large FPGA configuration.[4]
- encryption-based licensing Requires TTY[5]
- HW-IP binding methods use mechanisms in scured ROM or flash memory. They are vulnerable to side-channel attacks.[6]
- PUF[7]

[3] Jiliang Zhang et al. "FPGA IP protection by binding finite state machine to physical unclonable function". In: *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on.* IEEE. 2013, pp. 1–4.

[4] "Design security in Stratix III devices Altera White Paper 0101". In: 2009.

[5] Roel Maes, Dries Schellekens, and Ingrid Verbauwhede. "A pay-per-use licensing scheme for hardware IP cores in recent SRAM-based FPGAs". In: *Information Forensics and Security, IEEE Transactions on* 7.1 (2012), pp. 98–108.

[6] Yousra Alkabani, Farinaz Koushanfar, and Miodrag Potkonjak. "Remote activation of ICs for piracy prevention and digital right management". In: *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design.* IEEE Press. 2007, pp. 674–677.

[7] Jiliang Zhang et al. "FPGA IP protection by binding finite state machine to physical unclonable function". In: *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on.* IEEE. 2013, pp. 1–4.

PUF with FSM

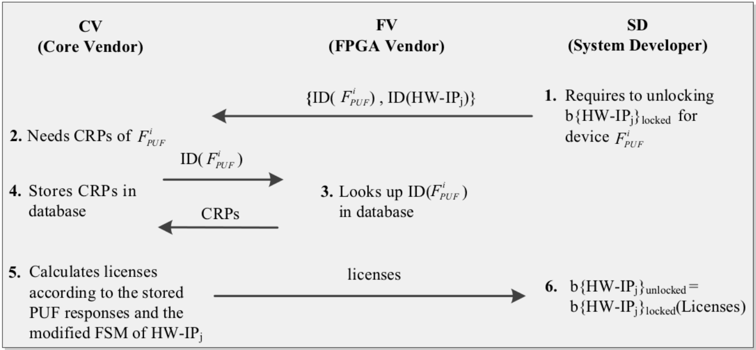

CV
(Core Vendor)
FV
(FPGA Vendor)
SD
(System Developer)
{ID(F^i_{PUF}) , ID(HW-IP$_j$)}
1. Requires to unlocking b{HW-IP$_j$}$_{locked}$ for device F^i_{PUF}
2. Needs CRPs of F^i_{PUF}
ID(F^i_{PUF})
4. Stores CRPs in database
3. Looks up ID(F^i_{PUF}) in database
CRPs
5. Calculates licenses according to the stored PUF responses and the modified FSM of HW-IP$_j$
licenses
6. b{HW-IP$_j$}$_{unlocked}$ = b{HW-IP$_j$}$_{locked}$(Licenses)

Figure : PUF with FSM lock[8]

[8] Jiliang Zhang et al. “FPGA IP protection by binding finite state machine to physical unclonable function”. In: *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on*. IEEE. 2013, pp. 1–4.

ASIC IP protection

Main methods for FPGA IP protection.[9]

- TTY, encryption related[10][11][12][13]
- PUF

[9] Roel Maes, Dries Schellekens, and Ingrid Verbauwhede. “A pay-per-use licensing scheme for hardware IP cores in recent SRAM-based FPGAs”. In: *Information Forensics and Security, IEEE Transactions on* 7.1 (2012), pp. 98–108.

[10] Yousra Alkabani, Farinaz Koushanfar, and Miodrag Potkonjak. “Remote activation of ICs for piracy prevention and digital right management”. In: *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design.* IEEE Press. 2007, pp. 674–677.

[11] Jarrod A Roy, Farinaz Koushanfar, and Igor L Markov. “EPIC: Ending piracy of integrated circuits”. In: *Proceedings of the conference on Design, automation and test in Europe.* ACM. 2008, pp. 1069–1074.

[12] Jarrod A Roy, Farinaz Koushanfar, and Igor L Markov. “Protecting bus-based hardware IP by secret sharing”. In: *Proceedings of the 45th annual Design Automation Conference.* ACM. 2008, pp. 846–851.

[13] Roel Maes et al. “Analysis and design of active IC metering schemes”. In: *Hardware-Oriented Security and Trust, 2009. HOST’09. IEEE International Workshop on.* IEEE. 2009, pp. 74–81.

Outline

1. Plans for the ICCAD
   - More Detailed Simulation
   - Metal Pattern
2. Reviews from DAC
3. python imported files
   - scripts problems
   - Standard cells in layout
4. Prep for ICCAD
   - Previous work
   - Optical PUF
5. Gdspy works
6. gds2txt
   - Integration with python tool

optical PUF against illegal copy of IP

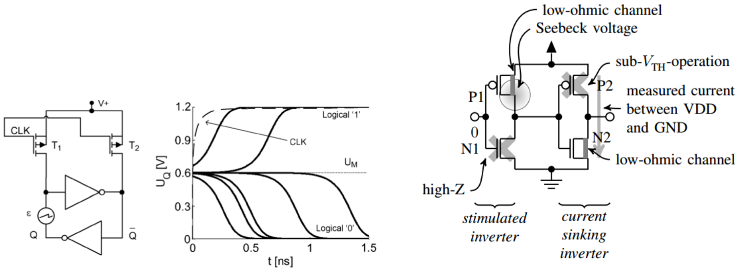


Figure : metastability[14] and laser stimulation[15]

[14] Piotr Zbigniew Wieczorek and Krzysztof Golofit. "Dual-metastability time-competitive true random number generator". In: *Circuits and Systems I: Regular Papers, IEEE Transactions on* 61.1 (2014), pp. 134–145.

[15] Dmitry Nedospasov et al. "Invasive PUF analysis". In: *Fault Diagnosis and Tolerance in Cryptography (FDTC), 2013 Workshop on.* IEEE. 2013, pp. 30–38.

optical PUF

This method requires IP core provide layouts.

- Design LFSR and seed generator inside IP core.
- With the scan with laser, metastability will not be maintained. Therefore, the seeds will be generated. And so does the key.
- For different users, LFSR generates different keys.

optical PUF

This method requires IP core provide layouts.

- Design LFSR and seed generator inside IP core.
- With the scan with laser, metastability will not be maintained. Therefore, the seeds will be generated. And so does the key.
- For different users, LFSR generates different keys.

optical PUF

This method requires IP core provide layouts.

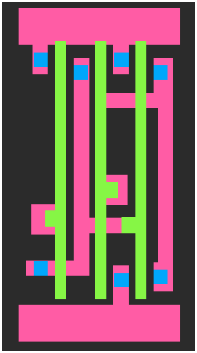
- Design LFSR and seed generator inside IP core.
- With the scan with laser, metastability will not be maintained. Therefore, the seeds will be generated. And so does the key.
- For different users, LFSR generates different keys.

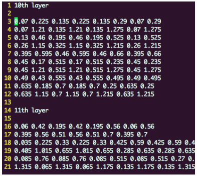
optical PUF

This method requires IP core provide layouts.

- Design LFSR and seed generator inside IP core.
- With the scan with laser, metastability will not be maintained. Therefore, the seeds will be generated. And so does the key.
- For different users, LFSR generates different keys.

Gdspy Read files





```
10th layer

0.07 0.225 0.135 0.225 0.135 0.29 0.07 0.29
0.07 1.21 0.135 1.21 0.135 1.275 0.07 1.275
0.13 0.46 0.195 0.46 0.195 0.525 0.13 0.525
0.26 1.15 0.325 1.15 0.325 1.215 0.26 1.215
0.395 0.595 0.46 0.595 0.46 0.66 0.395 0.66
0.45 0.17 0.515 0.17 0.515 0.235 0.45 0.235
0.45 1.21 0.515 1.21 0.515 1.275 0.45 1.275
0.49 0.43 0.555 0.43 0.555 0.495 0.49 0.495
0.635 0.185 0.7 0.185 0.7 0.25 0.635 0.25
0.635 1.15 0.7 1.15 0.7 1.215 0.635 1.215

11th layer

0.06 0.42 0.195 0.42 0.195 0.56 0.06 0.56
0.395 0.56 0.51 0.56 0.51 0.7 0.395 0.7
0.035 0.225 0.33 0.225 0.33 0.425 0.59 0.425 0.59 0.4
0.405 1.015 0.655 1.015 0.655 0.285 0.635 0.285 0.635
0.085 0.76 0.085 0.76 0.085 0.515 0.085 0.515 0.27 0.
1.315 0.065 1.315 0.065 1.175 0.135 1.175 0.135 1.315
```

Figure : Python Imported layout and outputted file

Layer info

LayerName	Layer#	Abbreviation
active	1	active
pwell	2	pwell
nwell	3	nwell
nimplant	4	nimp
pimplant	5	pimp
vtg	6	vtg
vth	7	vth
thkox	8	thkox
poly	9	poly
contact	10	contact
metal1	11	metal1
via1	12	via1
metal2	13	metal2
via2	14	via2

Table : Layer mapping info

Mar 4th, 2015

- Toy circuit has been designed, consulted with Zhen. Ronen's informed but he did not replied. I am running triggering possiblity tests for the circuit to ensure the low triggering rate.
- I have finished integrating gds2txt tool. Now it is fully automated, and it can be run on celnode.
- DAC paper updated. Figures have not been updated.

- Toy circuit has been designed, consulted with Zhen. Ronen's informed but he did not replied. I am running triggering possiblity tests for the circuit to ensure the low triggering rate.
- I have finished integrating gds2txt tool. Now it is fully automated, and it can be run on celnode.
- DAC paper updated. Figures have not been updated.

- Toy circuit has been designed, consulted with Zhen. Ronen's informed but he did not replied. I am running triggering possiblity tests for the circuit to ensure the low triggering rate.
- I have finished integrating gds2txt tool. Now it is fully automated, and it can be run on celnode.
- DAC paper updated. Figures have not been updated.

Outline

1. Plans for the ICCAD
   - More Detailed Simulation
   - Metal Pattern
2. Reviews from DAC
3. python imported files
   - scripts problems
   - Standard cells in layout
4. Prep for ICCAD
   - Previous work
   - Optical PUF
5. Gdspy works
6. gds2txt
   - Integration with python tool

Integration with python tool

Mar 17, 2015

Cadence Virtuoso

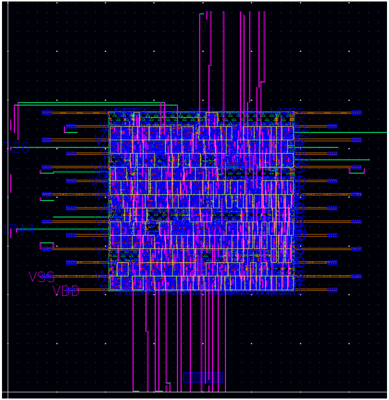


Figure : cds2python2cds

Encounter to Python

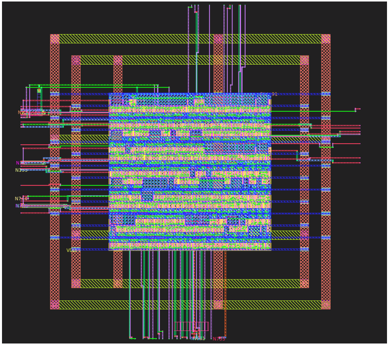


Figure : enc2python

Encounter to Matlab

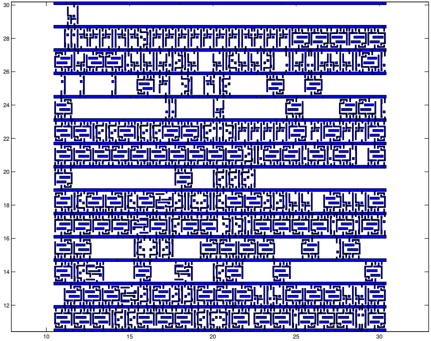


Figure : gds2txt

Thank you

Thank you.