

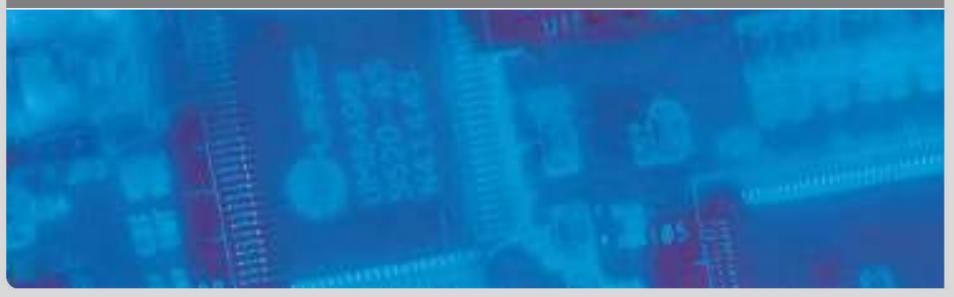


AxLS: An Open-Source Framework for Netlist Transformation Approximate Logic Synthesis

Jorge Castro-Godínez^{1,2}, Humberto Barrantes-García², Muhammad Shafique³, Jörg Henkel¹

¹Karlsruhe Institute of Technology, ²Instituto Tecnológico de Costa Rica, ³New York University Abu Dhabi

CES – Chair for Embedded Systems



Approximate Computing



Computational *quality* (accuracy of results) vs. computational *effort* (execution time, area, power, or energy).



Input image



PDP = 0.62 / PSNR = 22.4 dB



PDP = 1.00 / PSNR = inf dB



PDP = 0.61 / PSNR = 19.52 dB



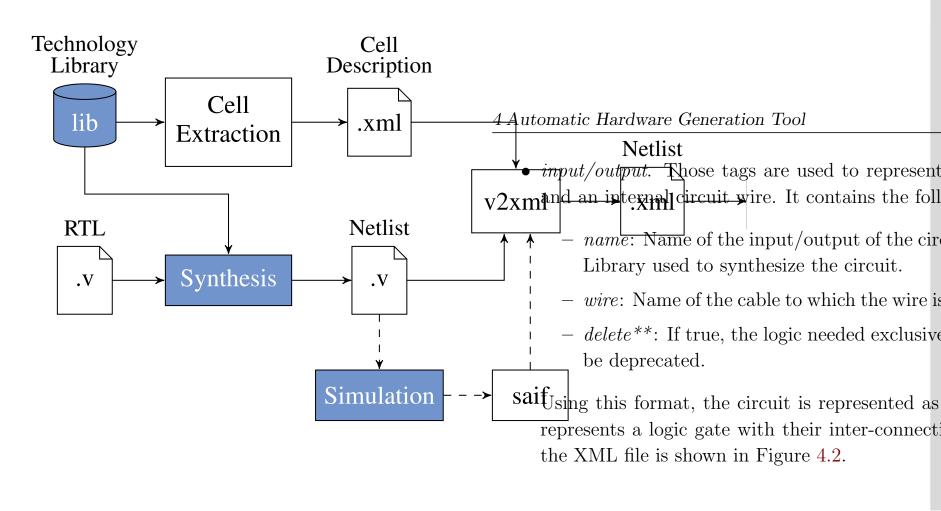
PDP = 0.69 / PSNR = 29.8 dB

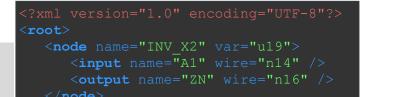
Motivation



- Approximate Logic Synthesis (ALS)
 - Generate approximate circuits from accurate implementations.
- Functional simplification:
 - Netlist transformation.
 - Boolean re-writing.
 - Approximate High-Level Synthesis.
- Missing an open-source framework for netlist transformation-based.







4

represents a logic gate with their inter-connections to other gate the XML file is shown in Figure 4.2.

AxLS – XML-based Netlist

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
  <node name="INV_X2" var="u19">
    <input name="A1" wire="n14" />
    <output name="ZN" wire="n16" />
  </node>
  <node name="OAI21 X1" var="u23">
    <input name="A1" wire="n14" />
    <input name="A2" wire="n20" />
    <output name="ZN" wire="n21" />
  </node>
  <node name="NAND2_X2" var="u33">
    <input name="A1" wire="n16" />
    <input name="A2" wire="n14" />
    <output name="ZN" wire="S[1]" />
  </node>
  [ ... ]
  <circuitinputs>
    <input var="in[0]"/>
    <input var="in[1]"/>
  </circuitinputs>
  <circuitoutputs>
    <output var="S[0]"/>
    <output var="S[1]"/>
  </circuitoutputs>
</root>
```

XML Netlist Description

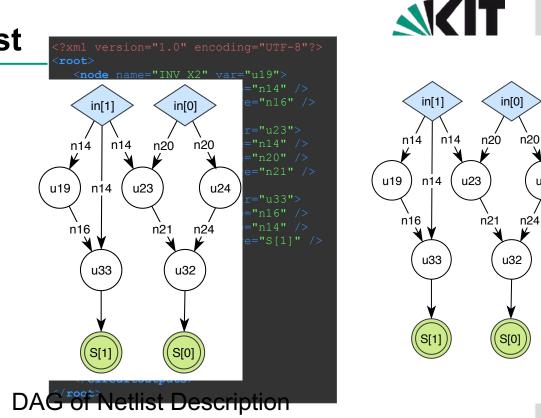


Figure 4.2: Sample of the circuit represented as an XML

υź

Some of the benefits of this representation is that these files co loaded quickly. The Python Element Tree library allows searches in as if it were a database using XPath syntax. XPath syntax could filtering them by their attribute values or navigate through the cir with the following line:

root.findall("./node/input[@wire='n54']/..")

J. Castro-Godínez @ WOSET '20, Virtual Workshop

5

The algorithm gets every node which has an input connected t ces.itec.kit.edu the circuit is represented in XML, two techniques were implement

AxLS Framework



Accuracy Threshold tic Hardware Generation Tool Approximate Transformed Netlist Netlist Netlist *it/output.* Those tags are used to represent a connection between a logic gate an internal dircuit wire. It contains the following attributesml2v .V Criteria name: Name of the input/output of the circuit. Obtained from the Technology Library used to synthesize the circuit. wire: Name of the cable to which the wire is connected in a given input/output. *delete*^{**}: If true, the logic needed exclusively to produce this wire signal could Circuit be deprecated. Synthesis Metrics ng this format, the circuit is represented as a list of nodes in XML, each node

resents a logic gate with their inter-connections to other gates. An example of XML file is shown in Figure 4.2.

sion="1.0" encoding="UTF-8"?>
name="INV_X2" var="u19">
put name="A1" wire="n14" />
tput name="ZN" wire="n16" />
>
name="OAI21 X1" var="u23">





AxLS implemented using Python language.

External tools:

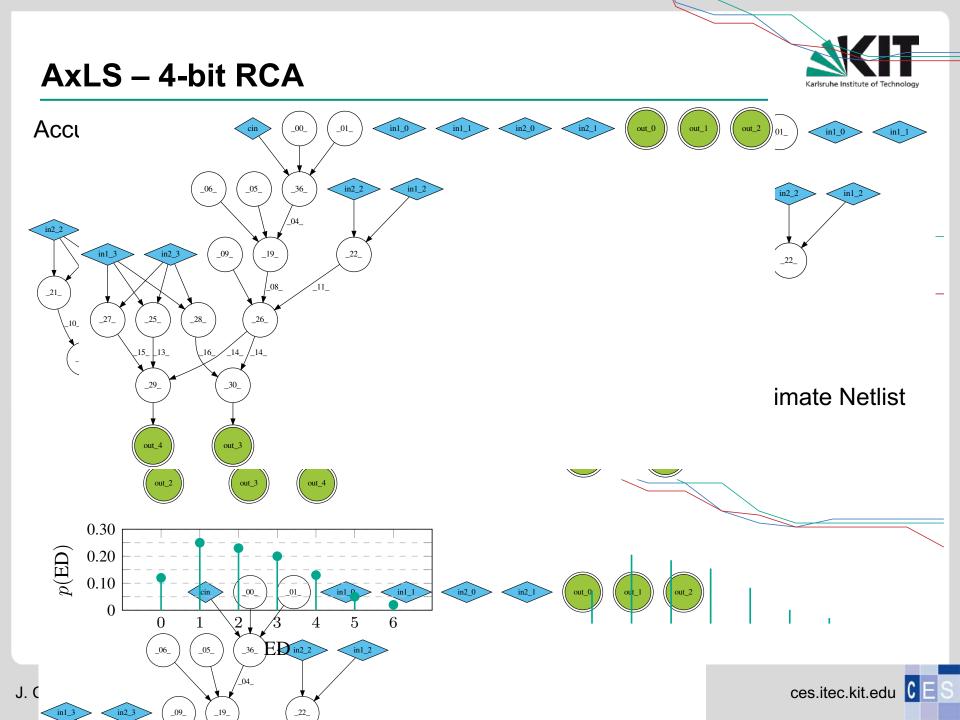
- Yosys for circuit synthesis.
- Icarus Verilog for netlist simulation.

Arithmetic circuits, particularly standard adders.

NanGate 15 nm technology library.

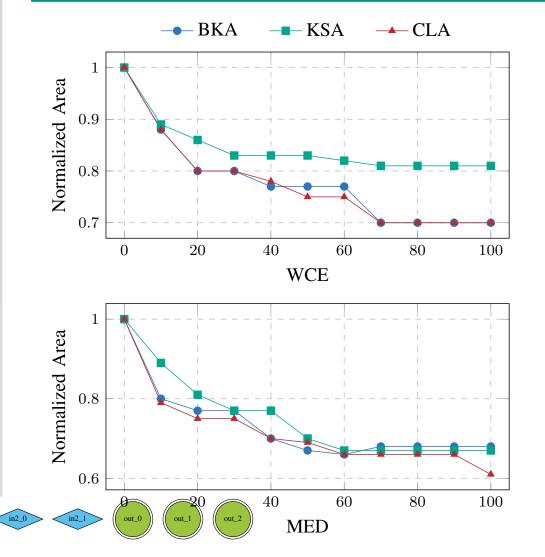


- Example of pruning technique, as Approximation Criteria:
- Primary input constant.
 - All dependencies explored.
- Primary output constant.
 - Explore nodes affecting such output.
- Nodes removed, replaced with 0 value.
- Starting from LSB.



AxLS – Evaluation





Virtual Workshop

ces.itec.kit.edu

Summary



- AxLS is an open-source framework for ALS techniques based on netlist transformation.
- Available at https://github.com/ECASLab/AxLS
- Example of pruning techniques that can be applied, for different accuracy metrics, and tested with standard adders.
- As future work, ML-based techniques for error estimation of gate pruning.

The End





Schloss Karlsruhe (Karlsruhe Palace)

Thanks for your watching!



J. Castro-Godínez @ WOSET '20, Virtual Workshop