Learn the OpenAccess API

Using Python

Initial Contribution By

James Masters
Intel - 2013
Updates & Additions
by Silicon Integration Initiative - 2020

Section 13 Introduction
Embedded Module Hierarchy

- What is an EMH?
- Learned how to import Verilog and make an EMH
- Learned how to traverse the EMH hierarchy

Section 13: Embedded Module Hierarchy (EMH)

- OpenAccess Hierarchy has three domains
  - Module Domain
    - Logical domain
  - Block Domain
    - Physical domain
  - Occurrence Domain
    - Union of Module and Block domains
- Some information is automatically transferred to all domains
- Information can be made specific to the Block domain
- Verilog export comes from Module domain
- LEF/DEF, GDS exports come from the Block domain

Section 13: Module Domain

- Used for logical view of the design
- Can have modules, within modules, within modules,….
- The verilog2oa translator can create the hierarchy
  - Try the –designPerMod option

Section 13: Module Domain

- Translators verilog2oa and oa2Verilog
  - Independent of OpenAccess, run independently
  - Read from and write to an OpenAccess database
  - Read data from the module domain
  - Only write data that has been saved to the design
- Note the translators only read and write architectural Verilog
  - Behavioral Verilog will not be interpreted
  - Example:

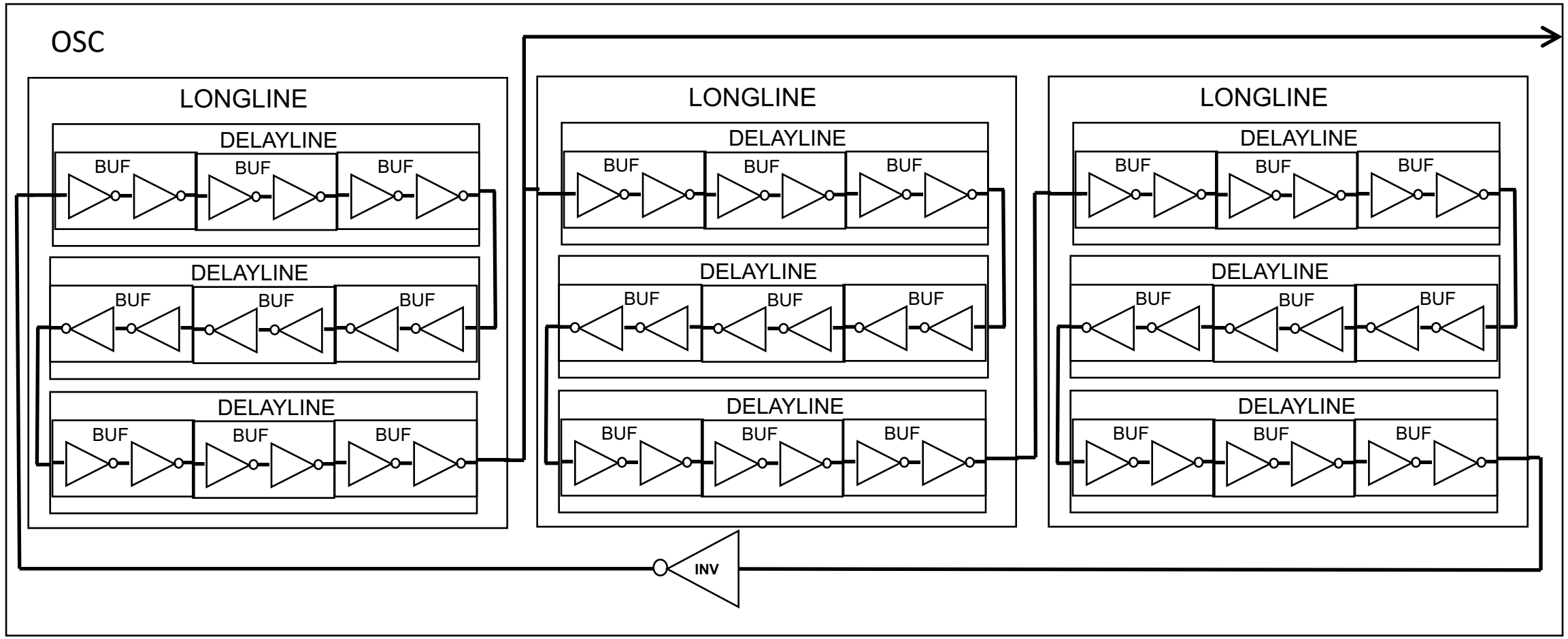
This works

```
module INV_X1 (A, ZN);
    input A;
    output ZN;
endmodule
```

This does not work

```
module INV_X1 (A, ZN);
    input A;
    output ZN;
    not(ZN, A);
endmodule
```

Section 13: Lab 13



Section 13: Lab 13.1

- Goal: learn how Verilog can be used to create a design hierarchy
- Write a script to import a hierarchtical Verilog design using the verilog2oa translator
  - OSC.v is available in the lab folder
- Open the design and print the name of the top module
- Compare your script to labs/lab13.1/buildEMHdesign.py

Section 13: Lab 13.2

- Goal: learn how to traverse an EMH hierarchy
- Write a script to find instances and nets on each level of hierarchy
  - If an instance has no instances in it's master, consider it a leaf cell
  - Note, the code must be recursive as you do not know how many levels of hierarchy there are.
- Compare your script to EMH_walk.py

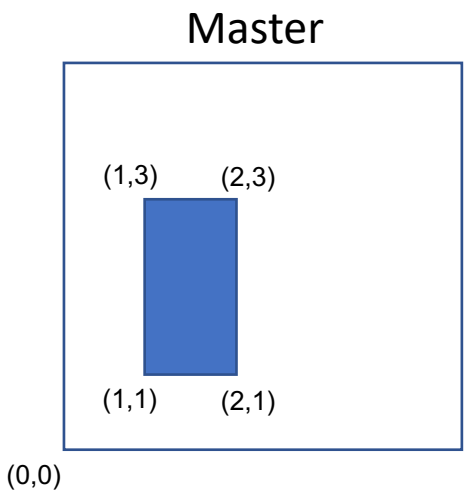
EMH Application

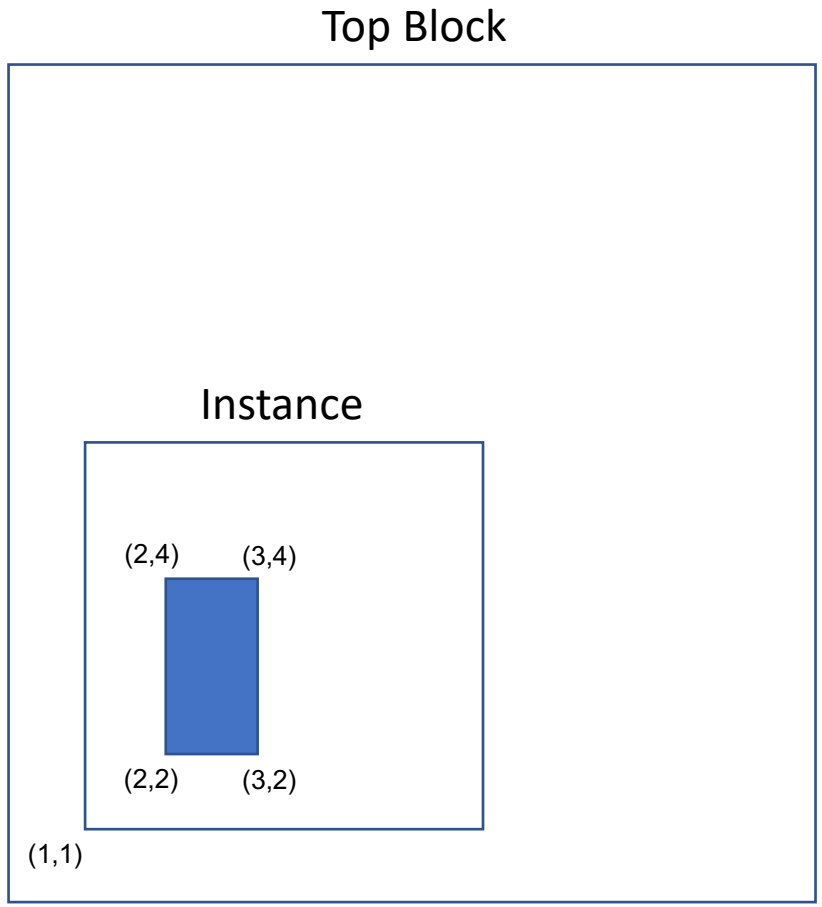
- An Embedded Module Hierarchy can be used to develop a Logic Macro/Block if it has complexity to warrant needing a hierarchy
  - Otherwise you would stay in the Block Domain
- At chip-level multiple Logic Macros and Analog Blocks from the Block Domain are integrated into a final system (Module Domain). This is a common application of the EMH hierarchy.

Section 13: Block Domain Hierarchy

- OpenAccess Hierarchy has three domains
  - Module Domain
    - Logical domain
  - Block Domain
    - Physical domain
  - Occurrence Domain
    - Union of Module and Block domains
- The block domain hierarchy can also be traversed using the instance master
- The problem arises with physical designs where all shape coordinates are relative to their design, so an instance shape does not have the coordinates of the occurrence but of the master

Section 13: Block Domain Hierarchy





- To map the shape from the master coordinates to the instance coordinates we can use oaTransform
- We can capture the transform of an instance and use the transform to move or copy the shape.

```
instance.getTransform(mytransform)
shape.copy(mytransform, new_block)
```

- Note that you can concatenate the transforms as you traverse the hierarchy to constantly reference the top level

Section 13: Lab 13.3

- Goal: learn how to flatten a block domain hierarchy
  - “flatten” implies the design is rendered a single level of hierarchy

- Write a script to flatten a hierarchical physical design
  - Run build_OSC.py to build the OSC design
  - Flatten the design as a new design using your script
  - Concatenate the instance transform with the shape to map the coordinates
  - Note: recursive code will help with multiple levels of hierarchy

- Compare your script to flatten_hier.py

Section 13 Summary

- Learned how to traverse the hierarchy
  - Both EMH and Block Hierarchy
- Where The EMH hierarchy fits within a design flow
- “Flattening” a Block Hierarchy to a single level
- Another use of oaTransform

Silicon Integration Initiative

www.si2.org

For details contact Marshall Tiner

Director of Production Standards

mtiner@si2.org