UCLA

Tutorial on OpenAccess Electronic Design Database and API

EE 201A – VLSI Design Automation – Winter 2024

UCLA Electrical Engineering

Instructor: Prof. Puneet Gupta

TA: George Karfakis

What is OpenAccess (OA)?

- API: EDA industry-standard C++ programming interface
- Database: efficiently represent complex and sizable design data
- Facilitates sharing data between CAD applications in design flow
- Official information
  - www.si2.org/?page=69 – “What is OpenAccess?”
  - www.si2.org/oac_index.php – landing page
  - Not free and open source in general, but we have license for this course

Getting Started

- SEASNET Linux server for course: `eeapps.seas.ucla.edu`
  - **IMPORTANT:** If you haven't done this yet, get a SEASNET account and ask to be added to ee201a ASAP!
- All setup files you need for course will be on Linux server at:
  - `/w/class.1/ee/ee201o/ee201ota`
- OA v22.04 installation root at:
  - `/w/class.1/ee/ee201o/ee201ota/oa/`
- Working in `csh` shell:
  - `$ source /w/class.1/ee/ee201o/ee201ota/csh_ee201a_setup`
  - Alternatively add the source command to your .cshrc file
  - Setup file is subject to updates over the quarter
- Read OA documentation!
  - `$ firefox /w/class.1/ee/ee201o/ee201ota/oa/doc/oa/html/index.html`
- Do not copy anything from OA installation to your personal machine – it is not permitted by the license.
- Do all homeworks/labs/projects in your own directory
  - Do NOT use your home SEASNET directory! Limited disk space and TA visibility.
  - These ee201o class directories will be wiped at the end of the quarter.
  - ***Always back up your data!***

Building Your First OA "Hello World" Program

- Log in to eeapps

```
$ ssh -X YOUR_SEASNET_USERNAME@eeapps.seas.ucla.edu
```

- Set up appropriate shell environment

```
$ source /w/class.1/ee/ee201o/ee201ota/csh_ee201a_setup
```

- Copy OATutorial_HelloWorld to your personal directory:

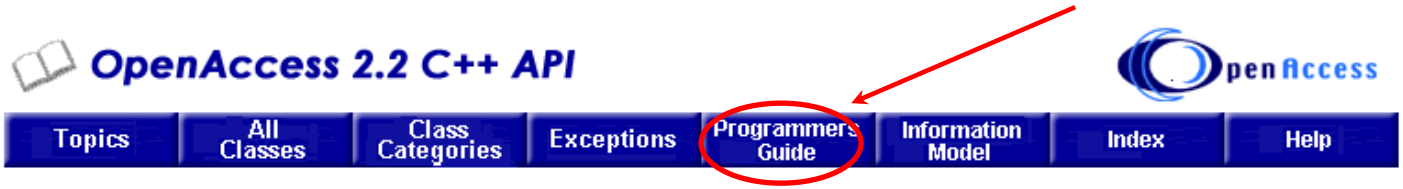
```
$ cp -r /w/class.1/ee/ee201o/ee201ota/oaTutorial_HelloWorld
<YOUR_HOME_DIRECTORY>/oaTutorial_HelloWorld
```

- Change to the project, build it, verify output

```
$ cd <YOUR_HOME_DIRECTORY>/oaTutorial_HelloWorld
$ make
$ ./HelloWorld
The design is created and opened in 'write' mode.
        The library name for this design is : DesignLib
        The cell name for this design is : And
        The view name for this design is : view
The following nets exist in this design.
        Hello
        World
```

- More advanced examples in `/w/class.1/ee/ee201o/ee201ota/oa/examples/oa/`

OA Documentation



OpenAccess C++ API Programmers Guide

This guide is for programmers who are developing new applications, or converting existing applications, to run in the OpenAccess environmer programming using the C++ programming language. The OpenAccess C++ API Programmers Guide consists of the following sections:

- OpenAccess Overview
  - API Programming Examples
- Compatibility for OpenAccess Applications and Data
- Getting Started—A HelloWorld Example
- OpenAccess Classes
- OpenAccess Libraries and Design Management (DM)
  - Turbo DM System
  - FileSys DM System
- Using Technology Databases
- Understanding Logical Connectivity
  - Defined Connections
- Global Nets
- Understanding Routes
- Physical Routing Segments (Orthogonal or Diagonal)
- Modeling Parasitics in OpenAccess
- Understanding the Derived Layer and Layer Operation Interfaces
- Undo and Redo Use Models in OpenAccess
- Using Transforms
- Extending the Database
- oaDesign Observer Notification, Binding, and Loading
- File Usage by OpenAccess Databases
- Name Mapping
- Deriving Your Own Namespace
- Creating and Modeling Process Rules and Constraints Via Parameters
- OpenAccess Hierarchy Domains
- Region Query Plug-In Architecture
- Tcl Bindings for OpenAccess APIs
- Support for Pcells
- Implementing Pcells Through Tcl
- Glossary
- A Typical Design Translation Flow and Related Issues

Start reading documentation on these topics

/w/class.1/ee/ee201o/ee201ota/oa/doc/oa/html/index.html

OA Translators

- Use *translators* to import and export design data from OA database to use in other CAD tools
- Create directory `DesignLib` in your project directory
- Example: Convert Verilog HDL to OA representation

```
$ cd YOUR_HOME_DIRECTORY/YOUR_PROJECT
$ mkdir DesignLib
$ verilog2oa -lib DesignLib -verilog file.v [Optional Args]
```

- More info in OA HTML documentation

OA Translators Documentation

OpenAccess C++ API Documentation

The OpenAccess API is a C++ programming interface for integrating design tools and design methodologies through a common, open database. OpenAccess documentation consists of documentation from this page or by using the navigation bar at the top of every page. For additional information about using the documentation, click the Help icon on the navigation bar.

Getting Started

- OpenAccess Release Notes
- What's New in OpenAccess 2.2
- Installing OpenAccess
- Adding a Search Solution to OpenAccess Documentation
- Known Problems and Solutions (KP&S)
- Getting Started — A HelloWorld Example

OpenAccess

- OpenAccess API Specification
- Programmers Guide
- Coding Style Guide and Standards
- Information Model Diagrams

Migrating to Newer OpenAccess Versions

- Compatibility for OpenAccess Applications and Data
- Features by Data Model

Extension Languages

- Tcl Bindings for OpenAccess APIs

Read on how to use translators to import/export design data to/from OA DB

Translators

- Using OpenAccess Translators
- LEF/DEF to OpenAccess Mapping
- LEF and DEF Translators
- Stream to OpenAccess Mapping
- Stream Translators
- Verilog to OpenAccess Mapping
- Verilog Translators
- SPEF to OpenAccess Mapping
- SPEF Translators

Plug-In Interfaces

- OpenAccess Plug-In API Specification
- How to Write a Plug-In
- Writing a C++ Pcell Generator
- How to Write a Pcell Evaluator Plug-In
- How to Write a Plug-In to Calculate bBoxes for Text
- How to Write Change Management System (CMS) Plug-Ins
- Implementing Pcells Through Tcl
- Region Query Plug-In Architecture

Return to top of page

OA Documentation

OA C++ classes and their member functions

OpenAccess API List of Classes

1 | 2 | a | b | c | d | e | f | g | h | i | l | m | n | o | p | r | s | t | u | v | w

1
oa1DLookupTbl
2
oa2DLookupTbl
A
oaAbstractType
oaAlignmentType
oaAnalysisLib
oaAnalysisOpPoint
oaAnalysisPoint
oaAnalysisPointArray
oaAntennaArea
oaAntennaAreaArray
oaAntennaData
oaAntennaModel
oaAntennaRatioArrayValue
oaAntennaRatioValue
oaAppDef
oaAppObject
oaAppObjectDef
oaAppObjectDefCollection
oaAppProp
oaArc
oaAreaBlockage
oaAreaBoundary
oaAreaHalo
oaArray

oaInstHeader
oaInstPropDisplay
oaInstQuery
oaInstTerm
oaInstTermAttrType
oaInt1DTblValue
oaInt2DTblValue
oaIntAppDef
oaIntDualIntArrayTblValue
oaInterPointerAppDef
oaInterpolateType
oaIntFltTblValue
oaIntProp
oaIntRange
oaIntRangeArray
oaIntRangeArray1DTblValue
oaIntRangeArray2DTblValue
oaIntRangeArrayValue
oaIntRangeProp
oaIntRangeValue
oaIntraPointerAppDef
oaIntValue
oaIter
L
oaLayer
oaLayerArray
oaLayerArrayConstraint

oaPath
oaPathSeg
oaPathStyle
oaPcellDef
oaPcellFileObserver
oaPcellLink
oaPcellObserver
oaPhysicalLayer
oaPiElmore
oaPin
oaPinConnectMethod
oaPinFig
oaPinType
oaPiPoleResidue
oaPlacementBlockageQuery
oaPlacementStatus
oaPoint
oaPointArray
oaPoleResidue
oaPolygon
oaPRBoundary
oaPrefRoutingDir
oaProp
oaPropDisplay
oaPurpose
oaPurposeArray
oaPurposeType

OA Classes – oaNet Example (1)

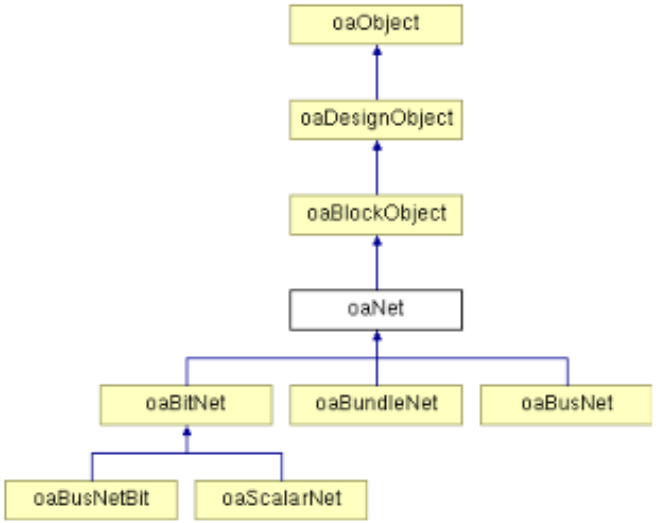
OpenAccess 2.2 C++ API

OpenAccess

Topics | All Classes | Class Categories | Exceptions | Programmers Guide | Information Model | Index | Help

oaNet Class Reference

Inheritance diagram for oaNet:



All member functions including inherited functions. Search for functions here!

Inherited Members | Schema Diagram

Member functions you may want to use

Public Methods

void **destroy** ()
oaBoolean isEmpty () const
oaBoolean isImplicit () const
oaBoolean isGlobal () const
oaUInt4 getNumBits () const

OA Classes – oaNet Example (2)

oaCollection oaNet::getInstTerms(oaUInt4 *filterFlags* = oacInstTermIterNotImplicit) const

This function returns a collection of instTerms in this net. The filter flag bits are defined below and may be logically OR'd together to refine the contents of the collection.

Parameters:

filterFlags Specifies what the collection contains. The bits flags are defined as follows:

- *oacInstTermIterNotImplicit:* the collection will only contain instTerms that were explicitly created
- *oacInstTermIterEquivNets:* the collection will also contain the instTerms on the nets that are equivalent to this one

Detail about class functions

Exceptions:

oacInstTermIterFlagNotApplicableOnNets

OA Classes – oalter Example

Public Methods

oalter (const **oaBaseCollection** &coll)
oalter (const oalter< obj > &iterIn)
obj * **getNext** ()

Detailed Description

template<class obj>
class oalter< obj >

The oalter class is used to iterate over the objects in an **oaCollection**. oaCollections a
multiple objects in some relationship to a single starting object. The oalter class allows
at a time and to test when all the objects are returned.

The intended usage of the oalter class is shown in the following example:

```
oaIter<oaShape>  sIter(view->getShapes());
oaShape          *myshape;
while (myshape = sIter.getNext()) { ... }
```

Note that some collections use special case iterators that do not belong to the oalter cl
has a different signature.

- Class quiz referenced “oaIter<oaNet> netIter(block->getNets());”
- What does this mean, now?

- It’s an iterator of <class_obj=oaNet> that can be used to iterate over the objects in an oaCollection.

Preparing for First Lab Assignment

- We will now cover a short tutorial on physical and logical connectivity as well as some important associated classes to help you prepare for the first Lab assignment.

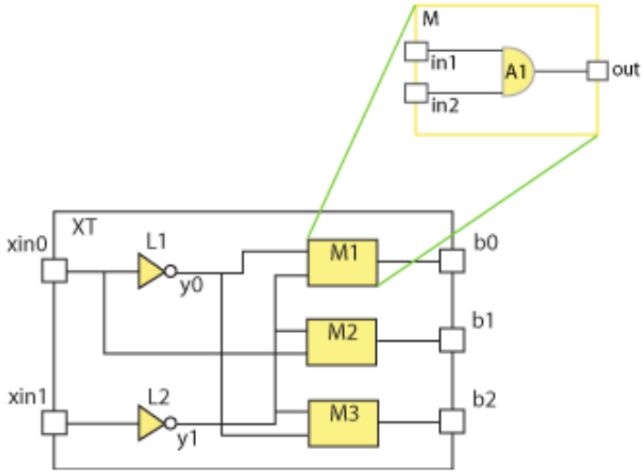
Review of Key OA Classes

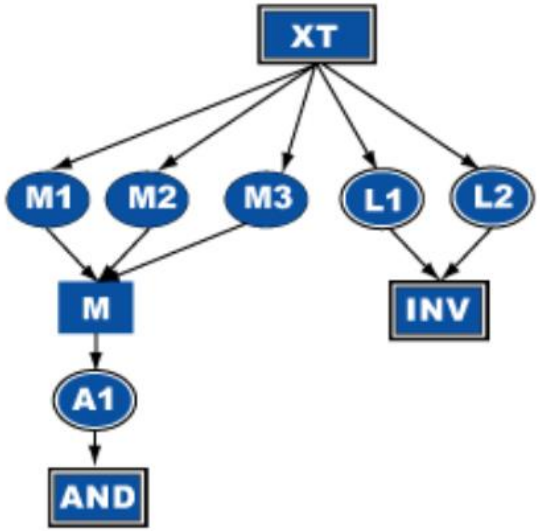
- oaDesign
  - Database containing full design information in various views - netlist, schematic, layout, etc.
- oaBlock
  - Container inside oaDesign for physical information
- oaInst
  - Instance (reference copy) of a different oaDesign in current one to maintain hierarchy.
  - This can mean standard cells, macros or sub-modules in a hierarchy.
- oaNet
  - Logical connectivity/wiring between endpoints (oaTerms, oaInstTerms)
- oaTerms, oaInstTerms
  - Interface connection points.

Hierarchy in an OA Design

- Simple example from OA docs

```
module XT(b0,b1,b2,xin0,xin1);

        input xin0,xin1;
        output b0,b1,b2;
        M M1 (b0,y0,y1);
        M M2 (b1,y1,xin0);
        M M3 (b2,y1,y0);

        INV L1 (y0,xin0);
        INV L2 (y1,xin1);

endmodule

module M(out,in1,in2);
        input in1, in2;
        output out;

        AND A1 (in1,in2,out);
endmodule
```

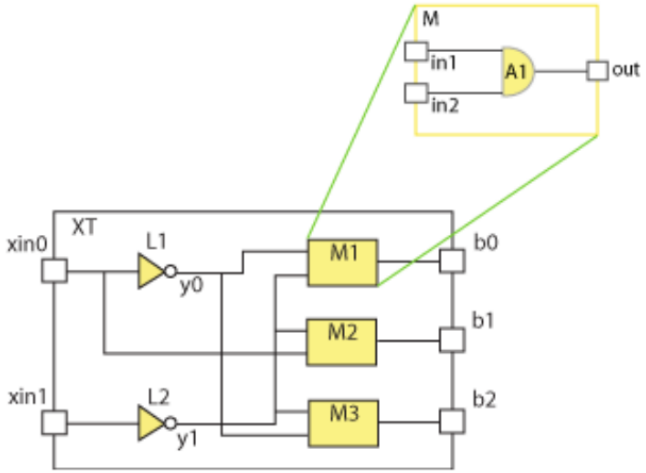


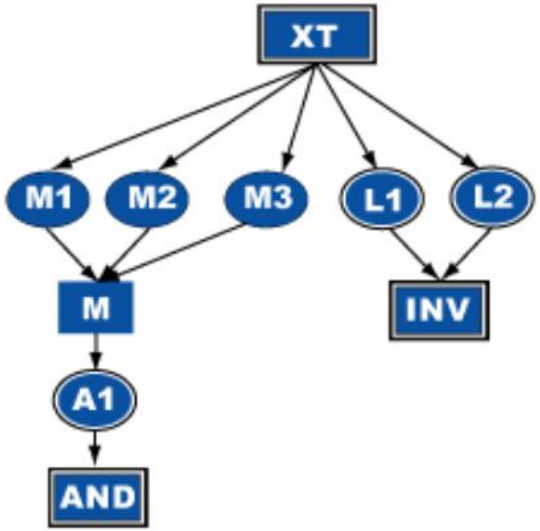


oaInsts

- Every unique reference to a different module/cell in the current block.

- Top block - XT:
  - L1, L2 of type INV
  - M1, M2, M3 of type M
- M block:
  - A1 of type AND

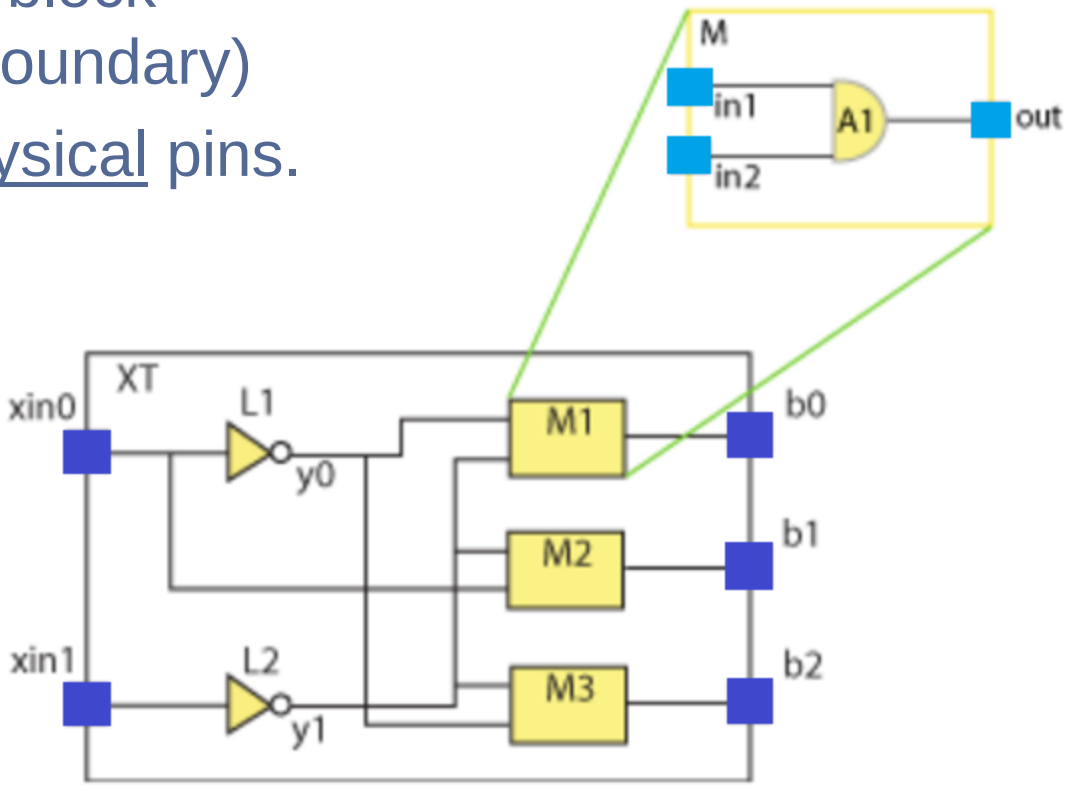




oaTerms

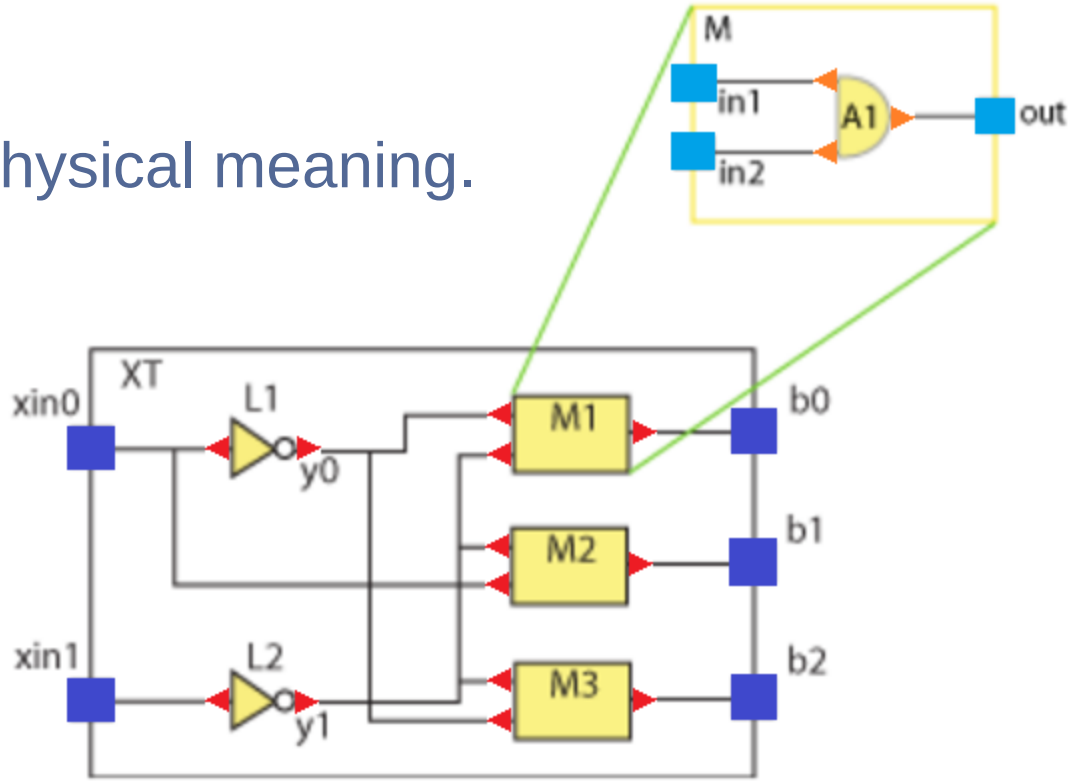
- I/O connection points at interface of current block (commonly at the boundary)
- Associated with physical pins.

- XT block:
  - xin0, xin1, b0, b1, b2
- M block:
  - in1, in2, out



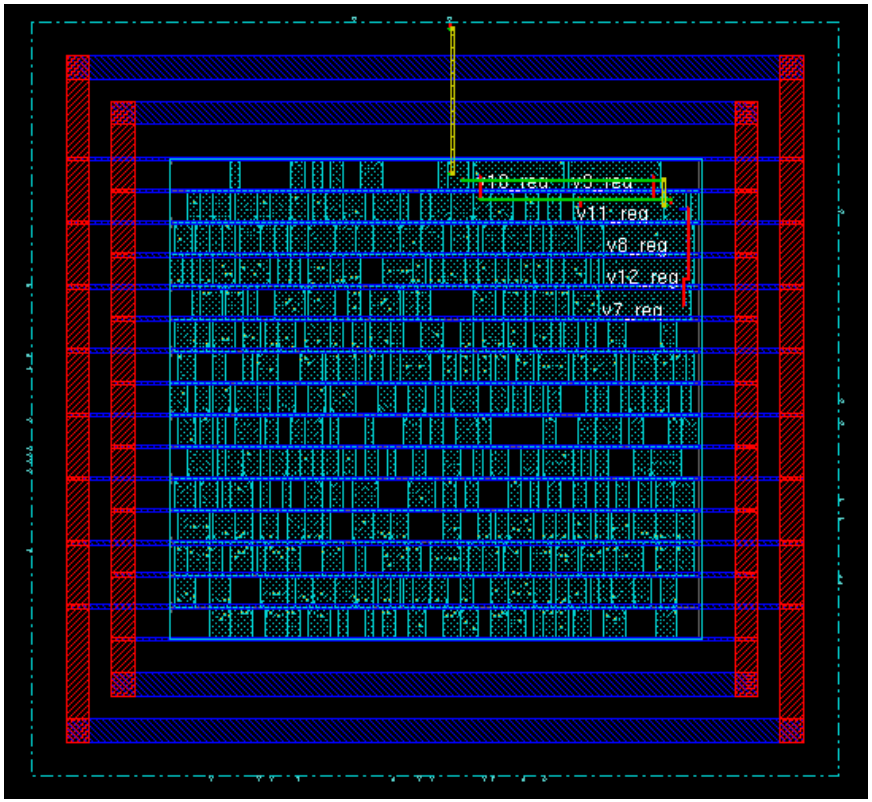
oaInstTerms

- I/O connection interfaces to respective oaInsts.
- Purely logical, no physical meaning.
- Associated with oaTerms inside the instantiated design block.



oaTerm / oaInstTerm Physical Locations

- oaTerms
  - Physical pins associated.
  - Pin figure → bounding box.
- oaInstTerms
  - Purely logical – no associated pins.
  - Simple approach for Labs: oaInst origin.
  - Why was this okay?
    - Good enough estimate for standard cells.



Preparing for First Lab Assignment (1)

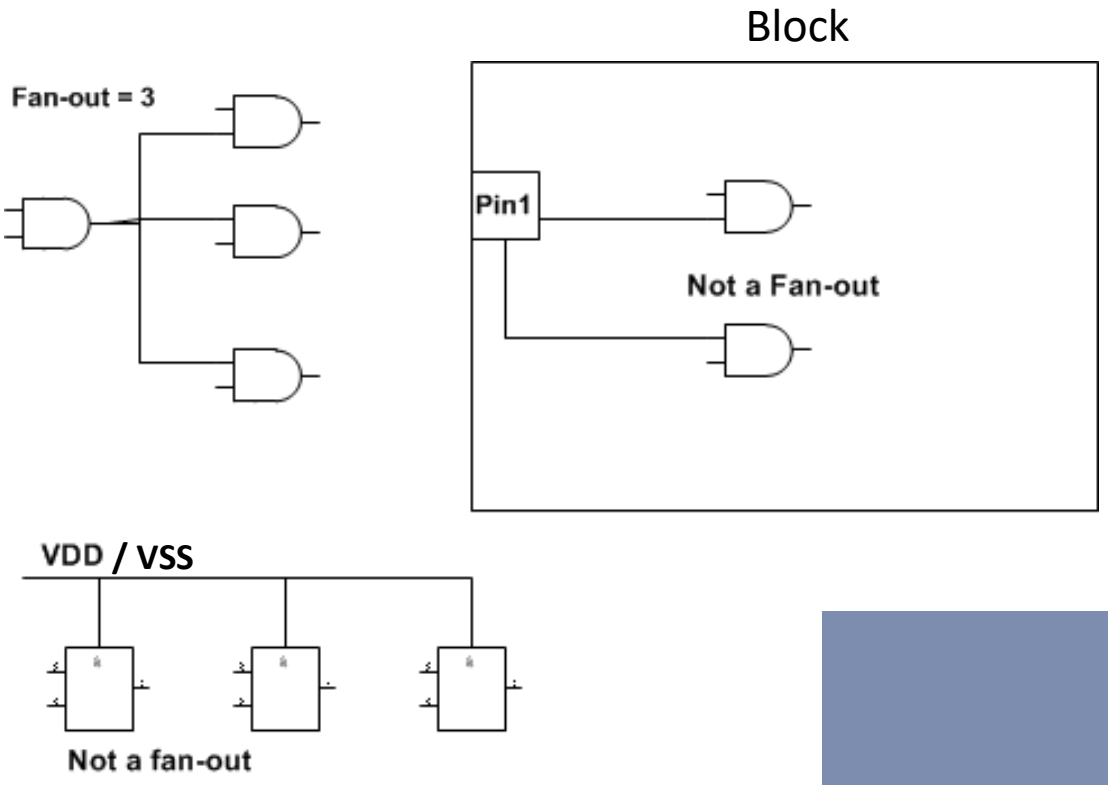
- Read about *logical connectivity* on OA docs
- Read these classes in docs:
  - `oaBlock`, `oaIter`, `oaCollection`, `oaNet`, `oaInstTerm`, `oaPin` (`oaTerm`), `oaInst`, `oaPoint`
- Look through OA examples

Preparing for First Lab Assignment (2)

OpenAccess C++ API Programmers Guide

This guide is for programmers who are developing new applications, or converting existing applications, to run in the OpenAccess environmer programming using the C++ programming language. The OpenAccess C++ API Programmers Guide consists of the following sections:

Fanout and Half-Perimeter Wirelength



Next Steps

- Use Piazza!
  - Ask and answer questions
  - Assignment will be posted soon