WARRIOR STRONG

WAYNE STATE
UNIVERSITY

Verilog to Digital Chip – An Introduction

Harpreet Singh, Ihfaz Islam
Department of Electrical and Computer Engineering
Wayne State University
Detroit MI 48202

Introduction

- In this presentation, we present some steps of designing basic digital chips from Verilog. We consider elementary Boolean gates and circuits and give the design steps so that the chips can be made using the open-source Tiny Tapeout procedure. We consider the following example, F = AB + C'.
- Steps using Cadence to Verilog to GDS file – quick overview
- Steps using efabless from Verilog to GDS file – quick overview
- Sending chip for fabrication – MOSIS, TinyTapeout

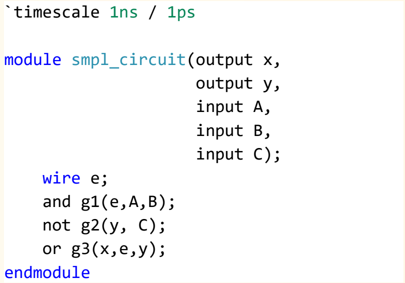
Steps using Cadence

1. Write the code in Verilog
2. Synthesis
3. Run Innovus and Automatic Place and Route flow
4. Run Virtuoso and Pad ring creation: This step will create rings of pad cells around the chip.
5. Physical Verification
6. Sending the design to MOSIS for fabrication

Steps using Cadence

1. **Write the code in Verilog**

We assume that Cadence tools and required PDKs are already installed on our systems. The procedure starts with writing the RTL for your circuit as shown below.



```
`timescale 1ns / 1ps

module smpl_circuit(output x,
                    output y,
                    input A,
                    input B,
                    input C);
    wire e;
    and g1(e,A,B);
    not g2(y, C);
    or g3(x,e,y);
endmodule
```

We will also be needing the following files:

- TCL Script for Genus
- NCSU CDK Library
- UofU Technology Library ("UofU_TechLib_ami06/")
- UofU Standard Cell Library ("UofU_Digital_v1_2/")

Steps using Cadence

2. Synthesis

Run the command **genus -legacy_ui -files genus.s** to perform synthesis.

Steps using Cadence

3. Run Innovus and Automatic Place and Route

Run the command **innovus -files innovus.cmd** to start Innovus tool.

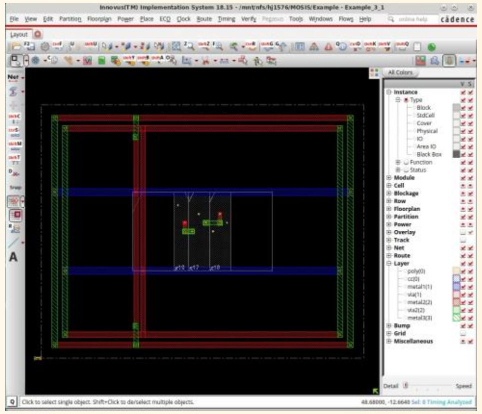

Innovus(TM) Implementation System 18.15 - /mnt/nfs/hj1576/MOSIS/Example - Example_3_1
File View Edit Partition Floorplan Power Place ECO Clock Route Timing Verify Pegasus Tools Windows Flows Help
cādence
Layout
All Colors
Instance
Layer
Click to select single object. Shift+Click to de/select multiple objects.

Figure 1: Innovus cell layout

Save netlist file with new name by going to Files → save → Verilog, now save as *nlopt.v*. Now save the def file with new name Files → save → def, now save as *Example_3_1.def.*

Steps using Cadence

4. Run Virtuoso & Pad ring creation

We will run Virtuoso to create the layout and pad ring. Run the two commands below to start Virtuoso.

source cshrc_virtuoso

virtuoso

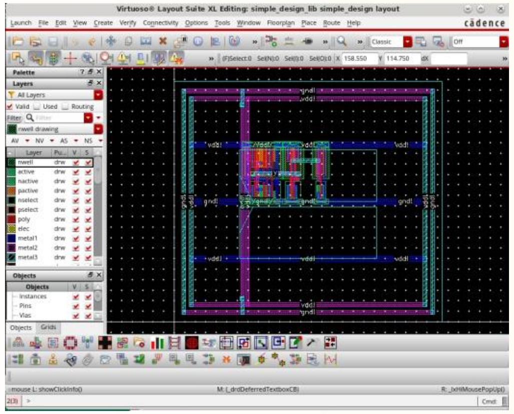

Virtuoso® Layout Suite XL Editing: simple_design_lib simple_design layout

Figure 2: Virtuoso design layout

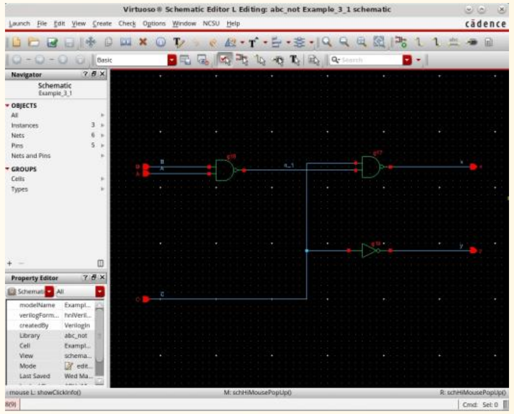

Virtuoso® Schematic Editor L Editing: abc_not Example_3_1 schematic

Figure 3: Virtuoso schematic editor

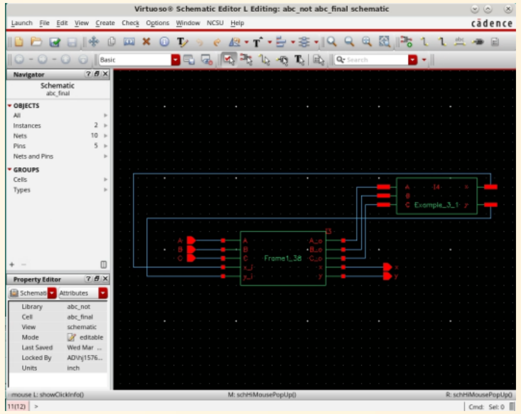

Virtuoso® Schematic Editor L Editing: abc_not abc_final schematic

Figure 4: Symbol generation

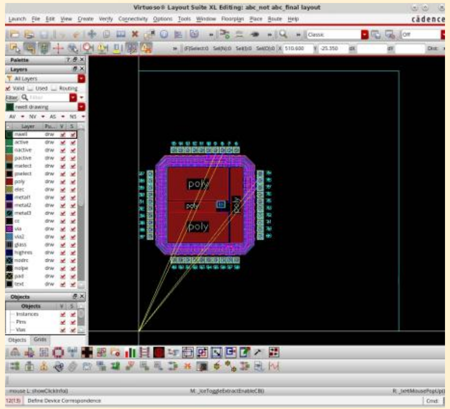


Figure 5: Virtuoso final layout

Steps using Cadence

5. Physical Verification

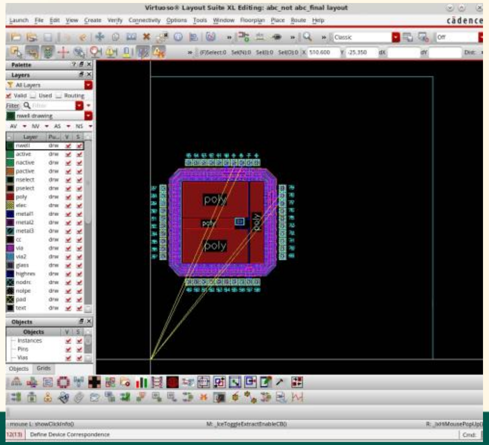
Verify → DRC

Verify → Extract

Verify → LVS

Go to the main home log window → file → Export → click on stream → click on Translate → ok → verify errors → yes.

You should get zero errors upon completion. The GDS file will be saved in the example folder which may be sent to MOSIS if required.



Virtuoso® Layout Suite XL Editing: abc_not abc_final layout

(F)Select:0 Sel(N):0 Sel(I):0 Sel(O):0 X 510.600 Y -25.350 dX dY Dist:

mouse L: showClickInfo() M: _lceToggleExtractEnableCB() R: _lxHiMousePopUp()

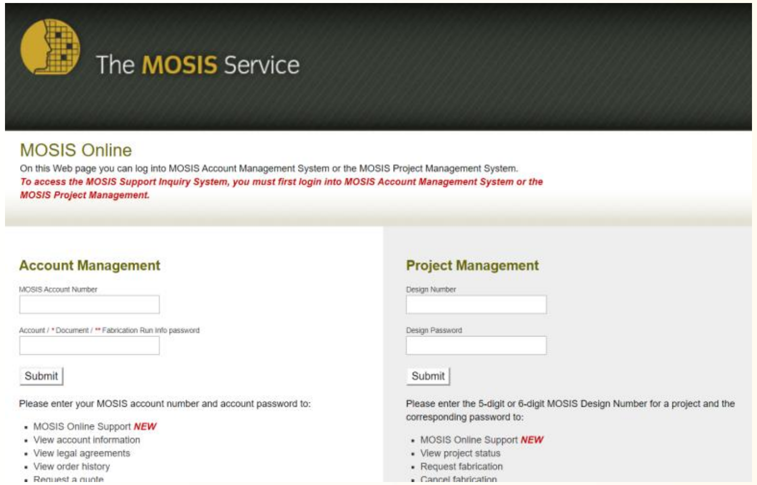
12(13) Define Device Correspondence Cmd:

Figure 5: Virtuoso final layout

Steps using Cadence

6. Sending the design to MOSIS for fabrication

- MOSIS stands for Metal Oxide Semiconductor Implementation Service and is a well-regarded provider of multi-project wafer (MPW) fabrication services.
- They also provide support services to help ensure a successful tapeout, which is the final step in the design process before chips are manufactured.



MOSIS Online

On this Web page you can log into MOSIS Account Management System or the MOSIS Project Management System.

To access the MOSIS Support Inquiry System, you must first login into MOSIS Account Management System or the MOSIS Project Management.

Account Management

MOSIS Account Number

Account / * Document / ** Fabrication Run info password

Please enter your MOSIS account number and account password to:

- MOSIS Online Support ***NEW***
- View account information
- View legal agreements
- View order history
- Request a quote

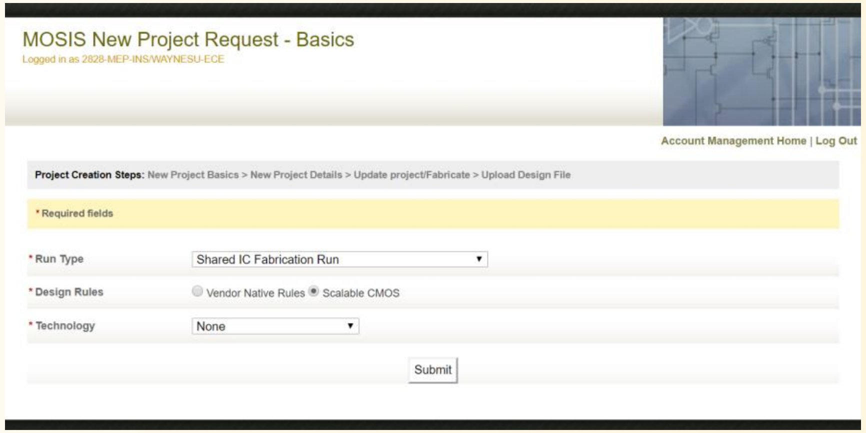
Project Management

Design Number

Design Password

Please enter the 5-digit or 6-digit MOSIS Design Number for a project and the corresponding password to:

- MOSIS Online Support ***NEW***
- View project status
- Request fabrication
- Cancel fabrication



MOSIS New Project Request - Basics

Logged in as 2828-MEP-INS/WAYNESU-ECE

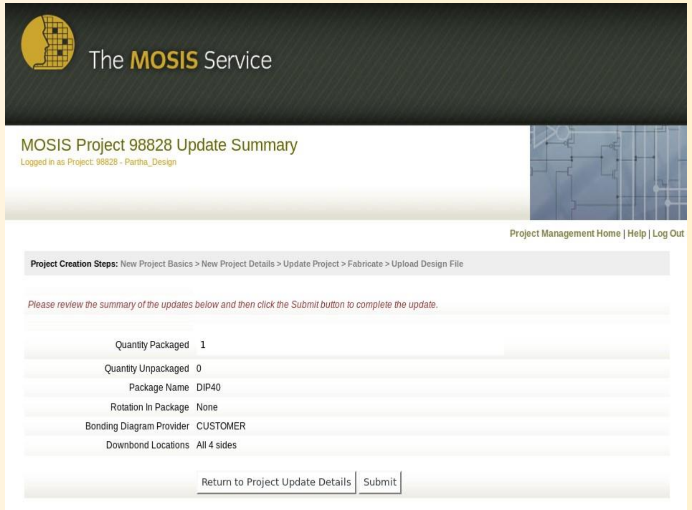
Project Creation Steps: New Project Basics > New Project Details > Update project/Fabricate > Upload Design File

* Required fields

* Run Type: Shared IC Fabrication Run ▾

* Design Rules: ○ Vendor Native Rules ● Scalable CMOS

* Technology: None ▾



MOSIS Project 98828 Update Summary

Logged in as Project: 98828 - Partha_Design

Project Creation Steps: New Project Basics > New Project Details > Update Project > Fabricate > Upload Design File

Please review the summary of the updates below and then click the Submit button to complete the update.

Quantity Packaged	1
Quantity Unpackaged	0
Package Name	DIP40
Rotation In Package	None
Bonding Diagram Provider	CUSTOMER
Downbond Locations	All 4 sides

Steps using efabless

1. Write the code for both the design and simulation in Verilog HDL.

2. Simulation is done to check the correctness of the design.

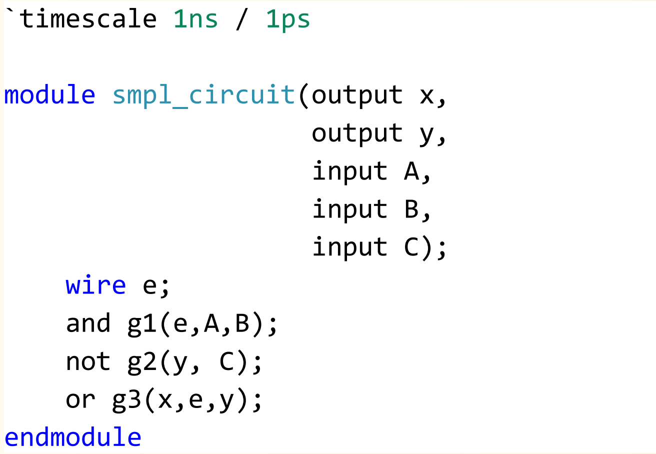
3. Running Automatic Place and Route flow is performed after the generation of Gate-Level Netlist automatically.

4. Sending design for fabrication

Steps using efabless

1. **Write the code for both the design and simulation in Verilog HDL.**

Let’s take F = AB + C’ as an example.



```
`timescale 1ns / 1ps

module smpl_circuit(output x,
                    output y,
                    input A,
                    input B,
                    input C);
    wire e;
    and g1(e,A,B);
    not g2(y, C);
    or g3(x,e,y);
endmodule
```

OpenLane ASIC flow needs to be installed inside the Docker container on an Ubuntu machine.

Steps using efabless

2. Simulation (Optional, not required for GDS generation)

- Beneficial to do simulation in different simulation environments and compare to ensure consistency and accuracy
- *iverilog* and *gtkwave* tools are used to view the ".vcd" files generated during simulation.

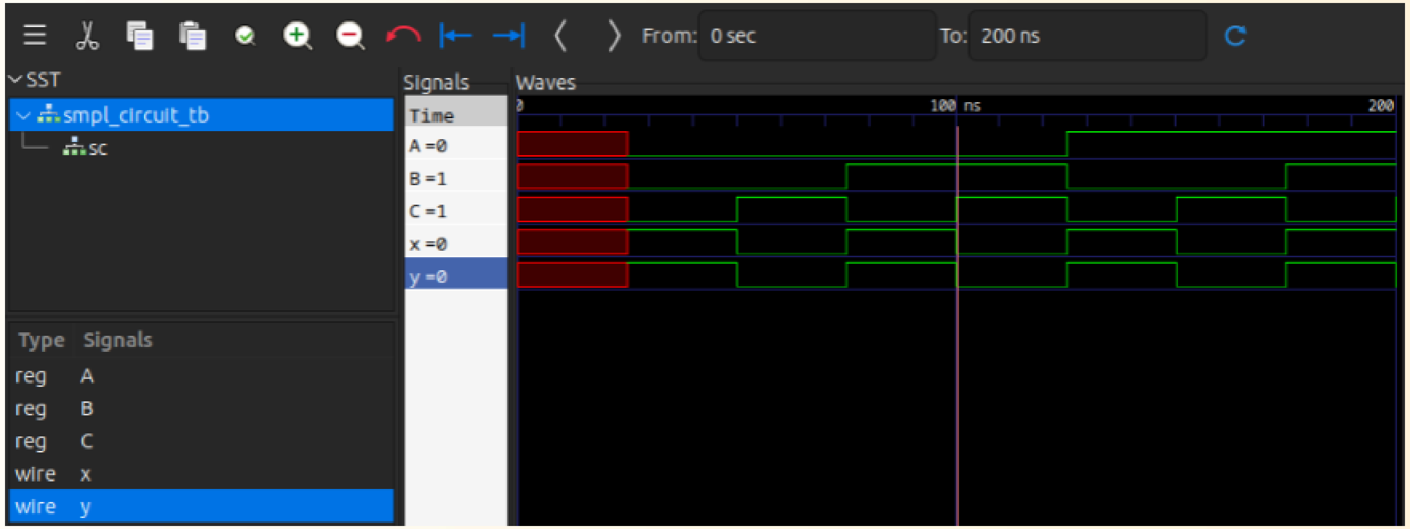


Figure 6: efabless simulation results of F = AB + C'

Steps using efabless

3. **Running Automatic Place and Route (APR) flow is performed after the generation of Gate-Level Netlist automatically.**

- We run the efabless APR to enter the docker environment to generate the GDS layout.
- The *klayout* tool can be used to view the GDS.

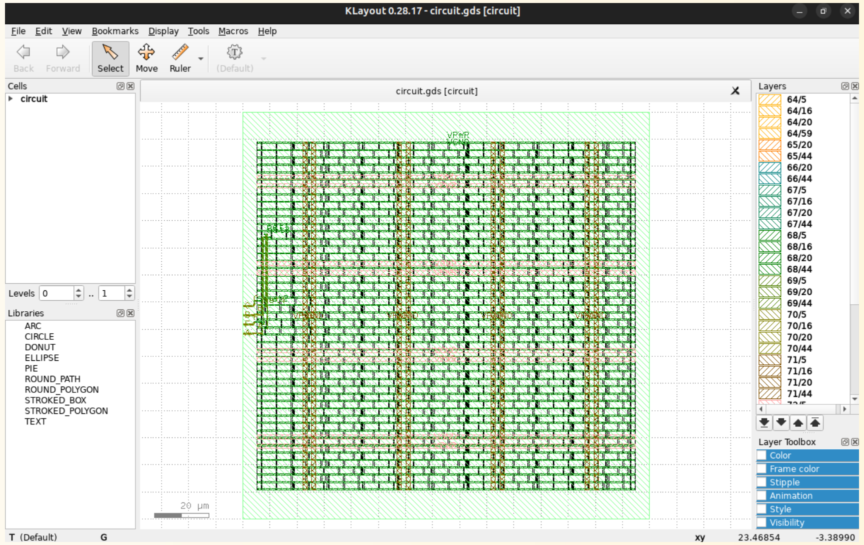


Figure 7: klayout GDS

Steps using efabless

4. Sending design for fabrication

The design must be submitted to the efabless MPW shuttle program. Users will need to create an account on the efabless shuttle website.

1. Create a GitHub project link including the caravel_repo

2. Perform pre-verification checks on the website to find out any potential conflicts or DRC errors when placed with other people’s project

3. Include a LICENSCE file in your repo

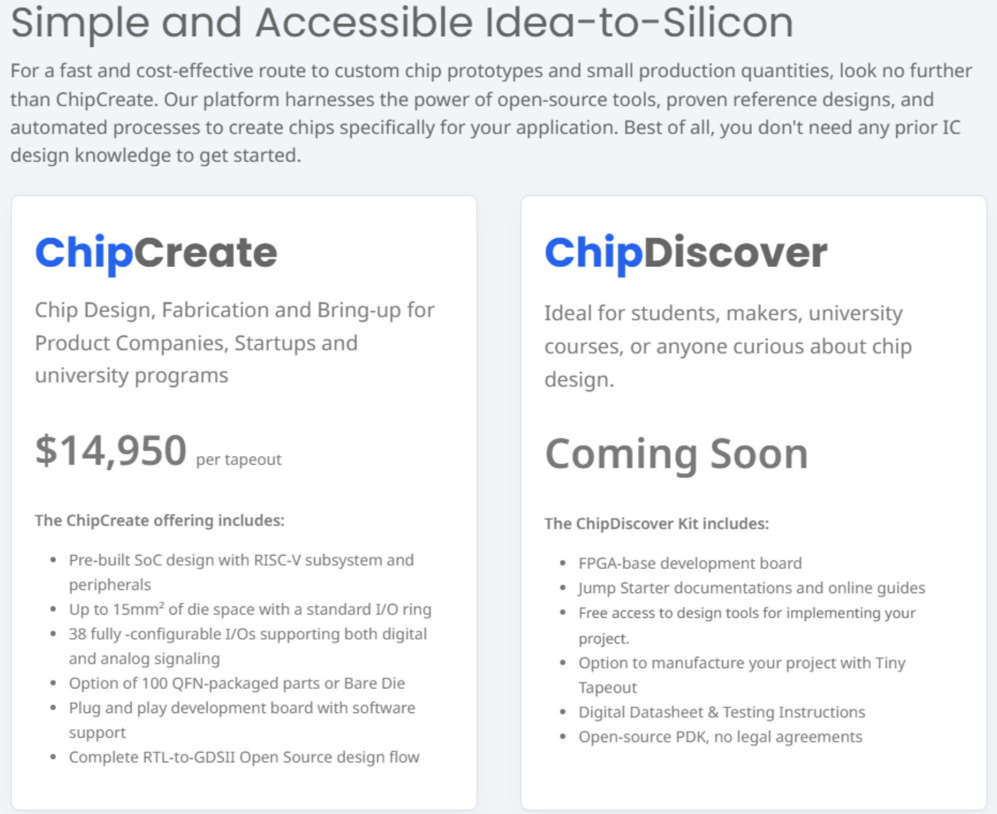
4. Submit your design once it’s clean. This may take a while depending upon the complexity of your design

5. The design will be received on a MPW caravel package which will have an ASIC containing the submissions of multiple users including ours.

Sending the design for fabrication

- MOSIS – not being supported for students
- Chipignite - $9750
- Chipignite mini - $3500
- Chipignite ML - Pricing starts at $14,750. Proto + Production Option: Includes full features for production-ready applications, starting at $30,000.
- TinyTapeout - $300

More Fabrication Options - ChipFoundry



Simple and Accessible Idea-to-Silicon

For a fast and cost-effective route to custom chip prototypes and small production quantities, look no further than ChipCreate. Our platform harnesses the power of open-source tools, proven reference designs, and automated processes to create chips specifically for your application. Best of all, you don't need any prior IC design knowledge to get started.

ChipCreate

Chip Design, Fabrication and Bring-up for Product Companies, Startups and university programs

$14,950 per tapeout

The ChipCreate offering includes:

- Pre-built SoC design with RISC-V subsystem and peripherals
- Up to 15mm² of die space with a standard I/O ring
- 38 fully -configurable I/Os supporting both digital and analog signaling
- Option of 100 QFN-packaged parts or Bare Die
- Plug and play development board with software support
- Complete RTL-to-GDSII Open Source design flow

ChipDiscover

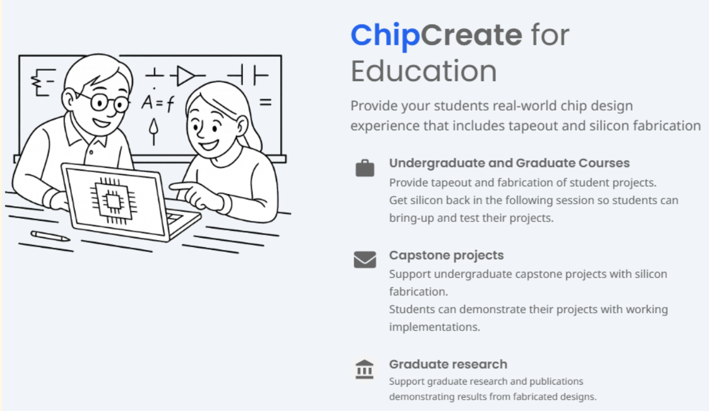
Ideal for students, makers, university courses, or anyone curious about chip design.

Coming Soon

The ChipDiscover Kit includes:

- FPGA-base development board
- Jump Starter documentations and online guides
- Free access to design tools for implementing your project.
- Option to manufacture your project with Tiny Tapeout
- Digital Datasheet & Testing Instructions
- Open-source PDK, no legal agreements

https://chipfoundry.io/



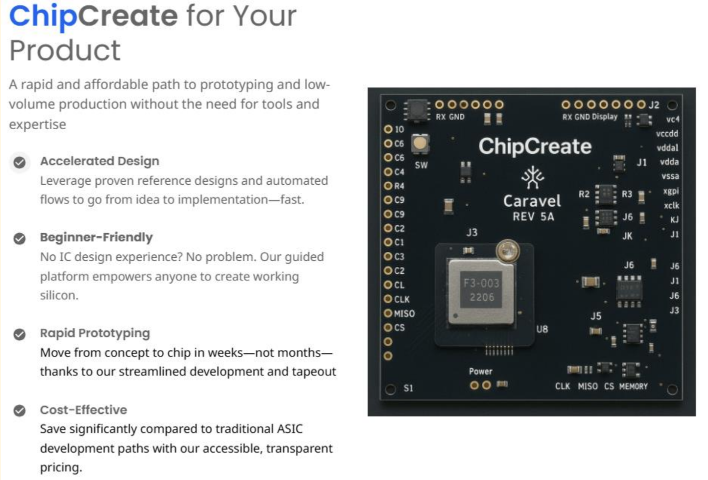
ChipCreate for Education

Provide your students real-world chip design experience that includes tapeout and silicon fabrication

Undergraduate and Graduate Courses
Provide tapeout and fabrication of student projects.
Get silicon back in the following session so students can bring-up and test their projects.

Capstone projects
Support undergraduate capstone projects with silicon fabrication.
Students can demonstrate their projects with working implementations.

Graduate research
Support graduate research and publications demonstrating results from fabricated designs.



ChipCreate for Your Product

A rapid and affordable path to prototyping and low-volume production without the need for tools and expertise

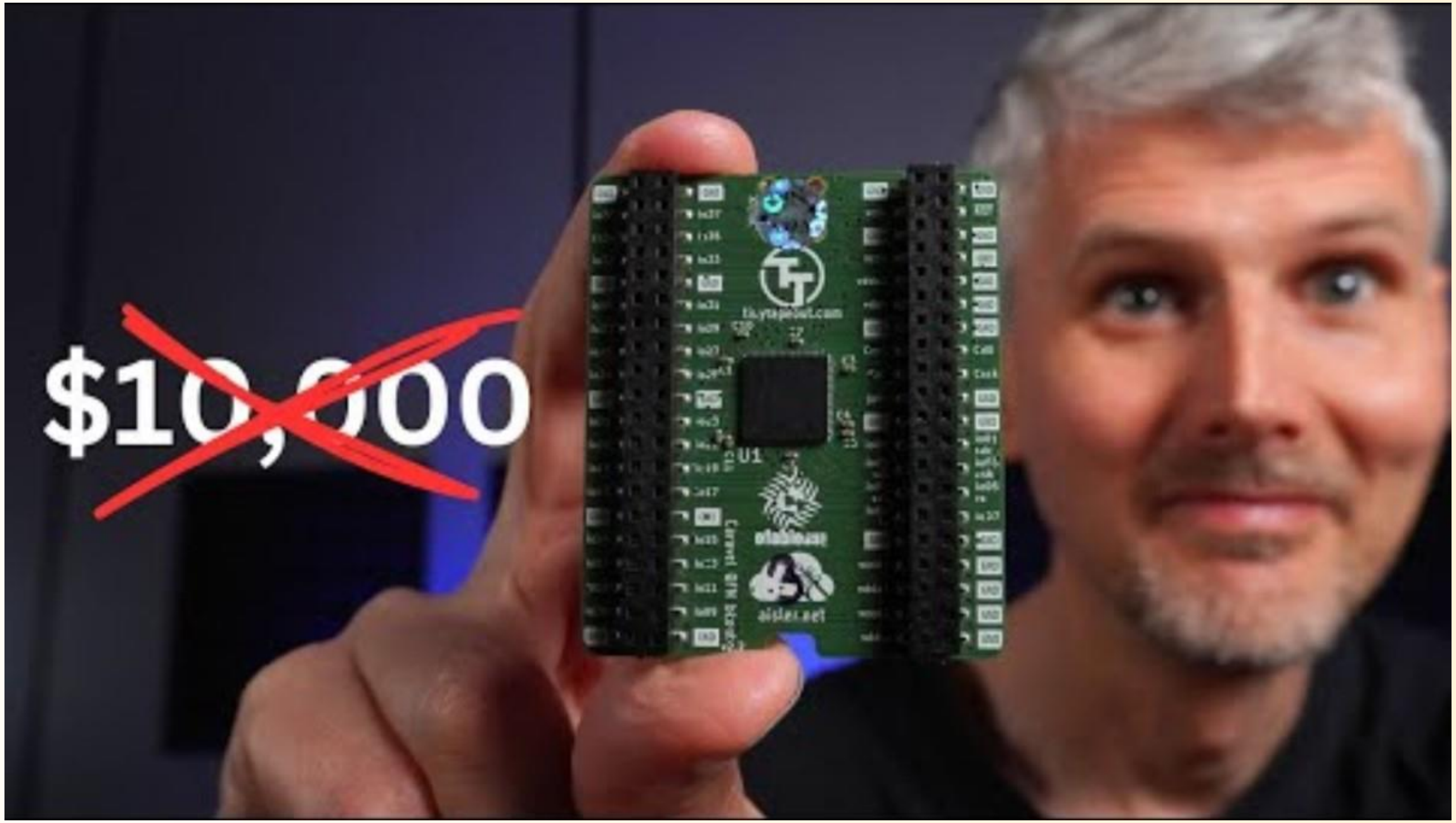
Accelerated Design
Leverage proven reference designs and automated flows to go from idea to implementation—fast.

Beginner-Friendly
No IC design experience? No problem. Our guided platform empowers anyone to create working silicon.

Rapid Prototyping
Move from concept to chip in weeks—not months—thanks to our streamlined development and tapeout

Cost-Effective
Save significantly compared to traditional ASIC development paths with our accessible, transparent pricing.

TinyTapeout Introduction Video



https://tinytapeout.com/

Tiny Tapeout is an educational project that aims to make it easier and cheaper than ever to get your digital and analog designs manufactured on a real chip.

TinyTapeout

- NAND Gate
- **Simple Boolean Circuit $F=AB+C'$**
- 1-bit Half Adder
- 1-bit Full Adder
- 2-4 Decoder
- 2-1 Multiplexer

TinyTapeout Procedure for F=AB+C’

Step 1 – Set up your Verilog project

Add your Verilog files to the source (`src`) folder.

Edit the `project.v` file and use the following modified Verilog program.

Change the source file name from `project.v` to `tt_um_islam_ihfaz_simple_circuit.v`.

Make sure to use your own name and information where applicable. **It is suggested that the Verilog file and module name match.**

TinyTapeout Procedure for F=AB+C’

Step 2 - Edit the `info.yaml` file.

Edit the `info.yaml` file and update information about your project as shown below, paying special attention to the `source_files` and `top_module` properties.

TinyTapeout Procedure for F=AB+C’

Step 3 - Edit `docs/info.md` and add a description of your project.

The `info.md` will be the datasheet for your digital design project and will explain how the design works, how it can be tested, if there is any external hardware used, and your pinout if included.

TinyTapeout Procedure for F=AB+C’

Step 4 – Set up the testbench (cocotb using Python)

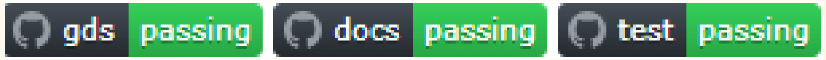
Tiny Tapeout uses *cocotb* to drive the design-under-test (DUT) and check the outputs.

TinyTapeout Procedure for F=AB+C’

Step 5 – View the GDS results including the 2D and 3D schematic views

Sending design to TinyTapeout

- On the main page of your GitHub repository, at the top of the README, if all the prior steps were followed properly, then all three flows (gds, docs, test) should be successful and will be displaying a green “passing” icon.

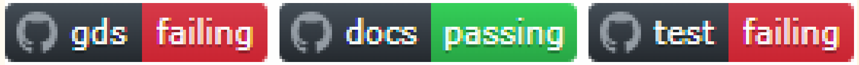


gds passing docs passing test passing

- This means that the design passes GitHub workflows and is ready for submission. The design may not be fabricated if it fails backend checks after submission, but most likely it will pass those checks as well.
- Once you are ready, you may upload your GitHub repository link to TinyTapeout for fabrication, paying the $300 fee, at this link https://app.tinytapeout.com/.
- Currently, the IHP 25B and Sky 25A shuttles are open. Each shuttle has its own GitHub template which needs to be filled out. F=AB+C’ design was done using Skywater shuttle #10 template, https://github.com/TinyTapeout/tt10-verilog-template.

Example of Unsuccessful TinyTapeout Design

- This will be showing on your repository main page if your design fails one or more of the GitHub workflows.
- For example, the design below is failing gds and test workflows.



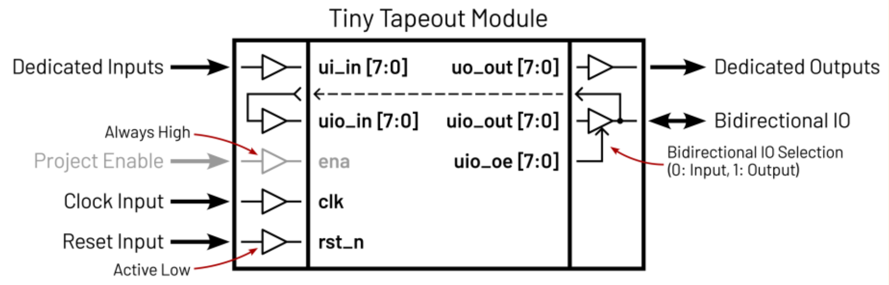
gds failing

docs passing

test failing

Limitations of TinyTapeout

- Limited number of I/O pins

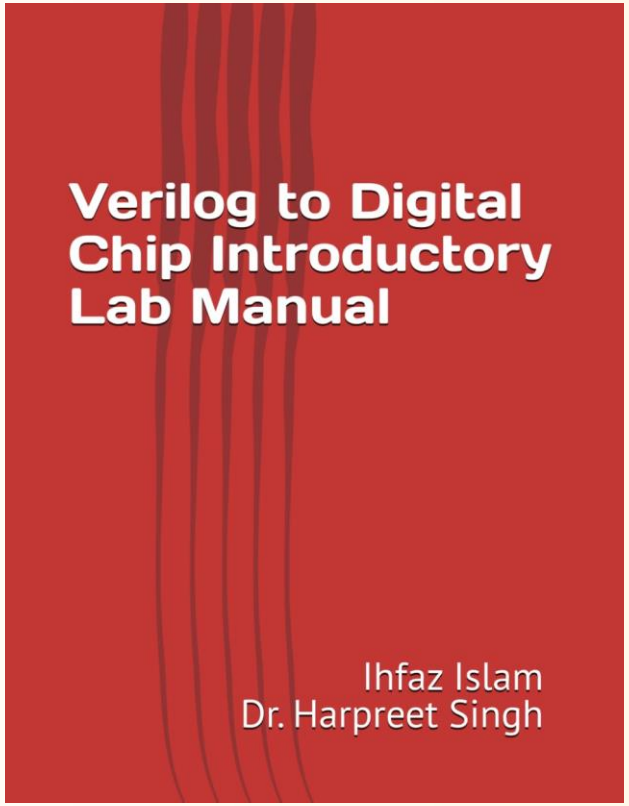


- A number of designs can be put on a single board.
- F=AB+C’ design uses 1x1 tile which is the smallest size chip. A single tile is about 167x108 μm.
- Add more tiles for larger designs (1x2, 2x2, 3x2, 4x2, 6x2 or 8x2) for more space on the board. However, the number of pins will remain the same regardless of tiles.

Conclusion

- TinyTapeout procedure for chip fabrication
- Limitations of TinyTapeout
- Looking for low-cost procedures for getting the digital chips made so that the students can learn the entire chip fabrication process without paying any money.
- What makes TinyTapeout so easy to use is that everything is done through GitHub, made possible by GitHub Actions and workflows.
- No terminal commands or local machine needed, but can be used.
- Online open-source silicon community

Verilog to Digital Chip Book on Amazon



Verilog to Digital Chip Introductory Lab Manual

Ihfaz Islam
Dr. Harpreet Singh

In this manual, we present some steps of designing basic digital chips from Verilog. We consider elementary Boolean gates and circuits and give the design steps so that the chips can be made using the open-source Tiny Tapeout procedure. We consider the following below.

NAND Gate
Simple Boolean Circuit F=AB+C'
1-bit Half Adder
1-bit Full Adder
2-4 Decoder
2-1 Multiplexer

We give all the steps from Verilog program to testing the final digital chip. The steps are as listed below.

Verilog Program
Testbench
Generating Truth Table
Vivado Simulation
Pipelining
Tiny Tapeout ASIC Design Flow
GDS
Testing

All the steps are given in Microsoft Word format so that the beginners can easily copy-paste the steps and code and verify the procedure. The authors do not take any guarantee if some steps fail to work on their machines. However, if there are any mistakes in any of the steps or code errors, please advise the authors to make corrections. The responsibility is entirely with the beginner. We hope these steps will be highly useful for the students in digital design to get the various digital chips designed and fabricated.

Ihfaz Islam

Mr. Islam graduated from Wayne State University Detroit, MI in August 2025 with a B.Sc. in electrical and computer engineering. He has done research in VLSI chip design and has presented at national conferences. His interests are in chip design, AI, and machine learning.

Dr. Harpreet Singh

Dr. Singh (M'73, SM'85, LSM'2009) received his B.Sc. Eng. degree from Punjabi University India, in 1963 and his M.S. and Ph.D degrees from the University of Roorkee, India, in 1966 and 1971. He taught at the University of Roorkee from 1963 to 1981. Since 1981 he has been with Wayne State University, Detroit, MI, where he has been a professor in the Department of Electrical and Computer Engineering.

He has worked in diversified areas of systems, networks, controls, fuzzy logic, DNA Nano-digital circuits, bio-circuits and VLSI design. Dr. Singh has about 400 publications in international journals and conferences and has received a number of awards. He has organized a number of national and international conferences and has supervised more than 25 Ph.D. theses in different areas of electrical and computer engineering. He has been awarded the distinguished Alumni Award of prestigious IIT Roorkee in 2012. As a founder and chair of Life Affinity Group of Southeastern Michigan section, the section received the 2017 IEEE Life Member Affinity Group Achievement award.

ISBN 9798289926722
90000
9 798289 926722

https://www.amazon.com/dp/B0FG2MFJP9

TinyTapeout Demo

$F = AB + C'$

Questions