Final Assembly

- Your final project chip consists of a core and a pad ring
  - Core is the guts
  - Pad ring (or pad frame) connects the guts to the outside world
- It's critical to do a functional simulation of your whole chip, including the pads!
  - Make sure you can drive the chip from the external interface
  - Make sure you have the core connected to the pads correctly.

Chip Core

- The Chip Core is everything that is inside the Pad Ring
  - Try to floorplan your core so that it's as small a rectangle as possible
  - At the very least, make sure it fits in the frame you've chosen
  - Make sure to connect vdd and gnd in the core!
  - This core can be DRC and LVS checked
  - This core can be simulated for functionality
  - This core is then routed to the pads

Core Sizes

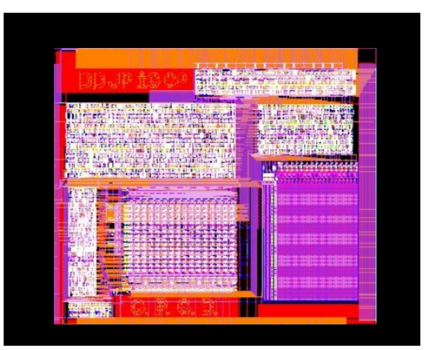
- All things are in terms of Tiny Chip Units (TCUs)
  - 1 TCU = 1.5x1.5mm outside dimension
  - 1 TCU = 900x900 usable core area
  - 2 TCU = 900x2300 usable core area
  - 4 TCU = 2300x2300 usable core area
- More on this later!

Connecting Core to Pads

- Once your core is complete, you need to connect it to the pad frame
  - Then you re-do the functional simulation, but through the pads this time
  - You should be able to re-use your testfixture
  - Also a final DRC and LVS which includes the pads
- Use vcar for connecting the core to the pads!
  - Chapter 12 in the CAD manual

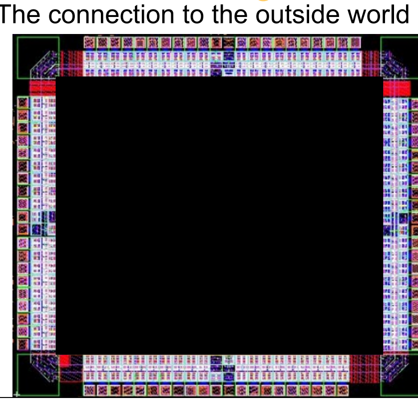
Core

- The guts of your chip

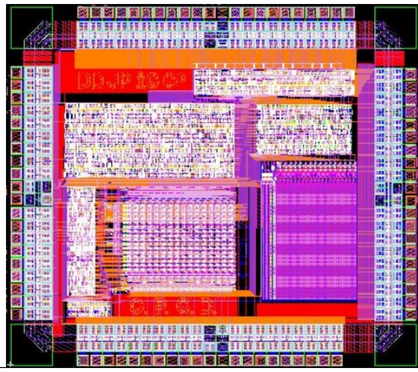


Pad Ring

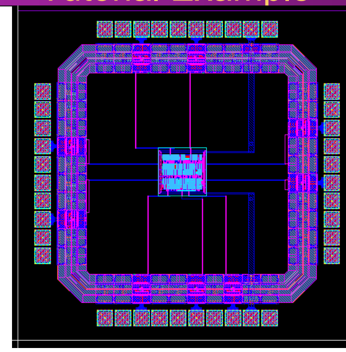
- The connection to the outside world



The Connected Chip



Tutorial Example

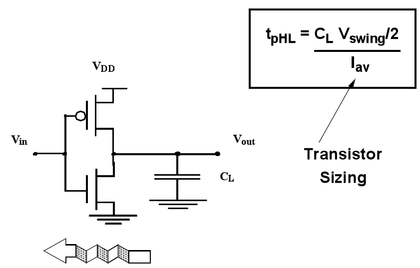


- A tiny state machine in a 1-tiny-chip frame

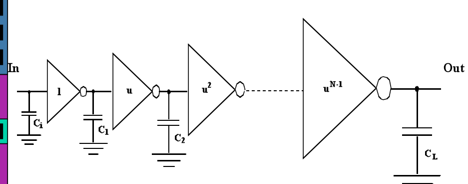
Pad Cells

- Started with a set of pads from MOSIS
  - Originally from Tanner Tools pads
  - Problem: the pads don't DRC, LVS, or simulate!
- Cameron Charles re-did the cells in 2002 (as a grad student) to fix these issues
- Result is UofU_Pads
  - /uusoc/facility/cad_common/local/Cadence/lib/OA/UofU_Pads
  - Use library manager to add this library
  - Name it UofU_Pads
  - They now DRC, LVS, and simulate!

Driving Large Capacitances

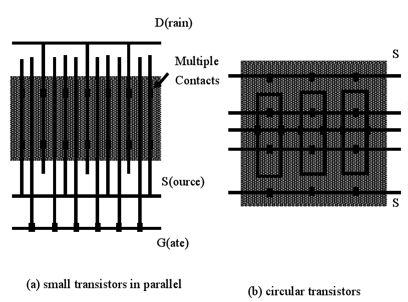


Using Cascaded Buffers

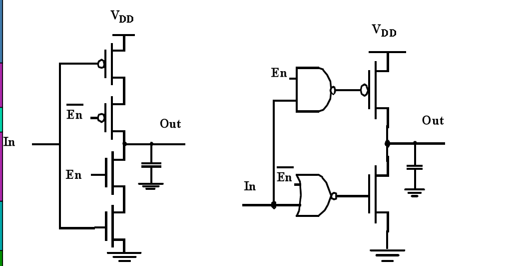


$u_{opt} = e$?

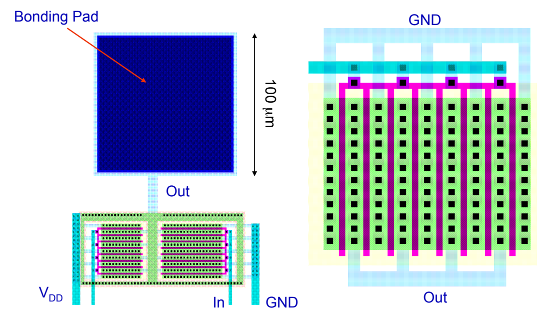
How to Design Large Transistors



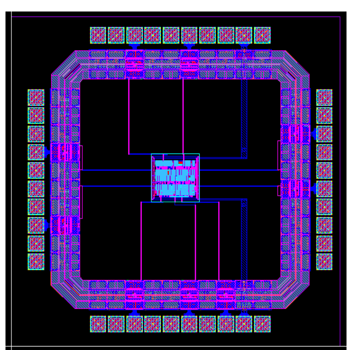
Tristate Buffers



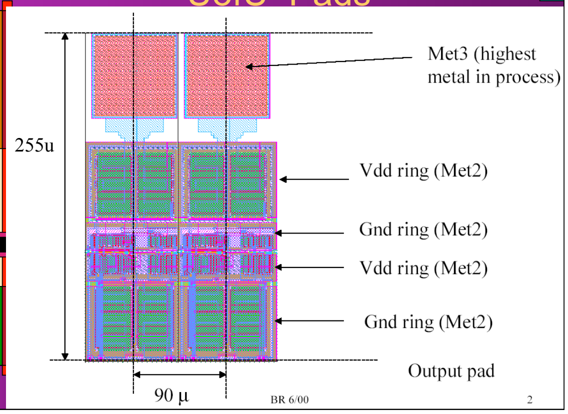
Bonding Pad Design



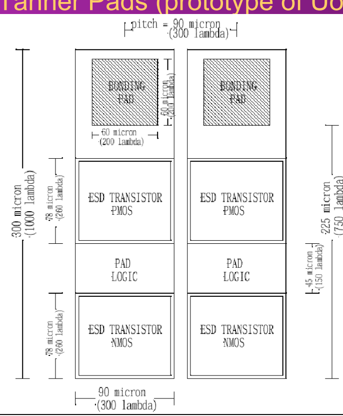
UofU_Pads



UofU_Pads



Tanner Pads (prototype of UofU_Pads)



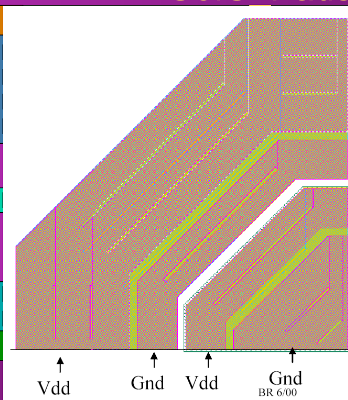
From Tanner documentation

Want minimum pitch pins so can get high pin count. ASIC pins tend to be long/narrow.

Tanner pads follow HP internal rules, not MOSIS rules.

3

UofU_Pads

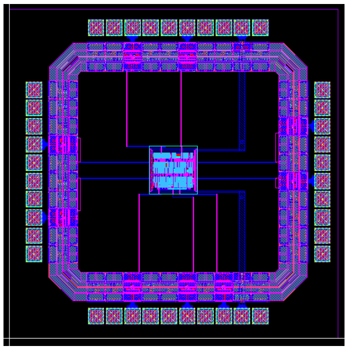


Corner pad

Used for continuity of power/gnd rings at corners

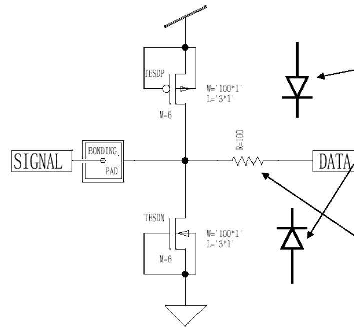
4

UofU_Pads



ESD and Analog Pads

Vdd/Gnd Input Signal clamping



Diodes Clamps input signals to Vdd/Gnd

Width large for high current capability

Resistor for input current limiting, diffusion or poly resistor

ESD Protection

A thick film device built from Met1, field oxide, and diffusion is used for ESD protection.

Field oxide device usually has threshold voltage in the 10's of volts.

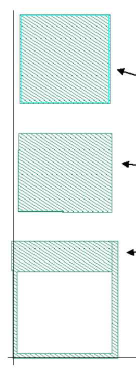
Metal 1 provides 'gate' of device, tied to pad, also tied to drain.

Width of device is very high to provide low on resistance, high current density.

Will handle any short duration spikes due to electro static discharge.

Most processes have special high threshold transistors that can be used for ESD protection instead of field oxide devices.

Pads from MOSIS



Diffusion areas in Tanner Input Pad

Diffusion under pad used for the thick film device

Diode protection

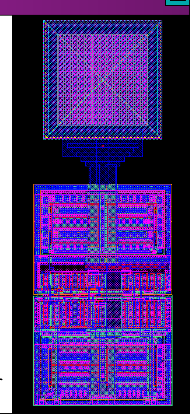
Input logic

ASIC Pads

- Most ASIC pads have multiple Vdd/Gnd Rings
- Leda Systems 0.25u, 0.35u pad library has the following rings
  - Avdd, Avss -- analog VDD/Gnd
  - Evdd, Evss – external Vdd/Gnd – supports pads with different Vdd/Gnd from core
  - Nvdd, Nvss – internal Vdd/Gnd – core VDD/GND
  - Vss/Vdd -- ?? (also core Vdd/Gnd??)

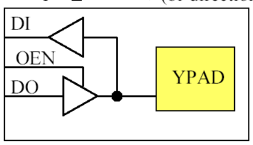
UofU_Pads

- pad_bidirhe
  - Bidirectional pad with high enable
- pad_in
  - Digital input pad
- pad_out
  - Digital output pad
- pad_vdd, pad_gnd
  - Power supply pads
- pad_io, pad_io_nores
  - Analog pads (with and without series resistor)
- pad_nc, pad_space
  - Non-connecting pad and spacer



Pad Interfaces

- pad_bidirhe (bi-directional pad)



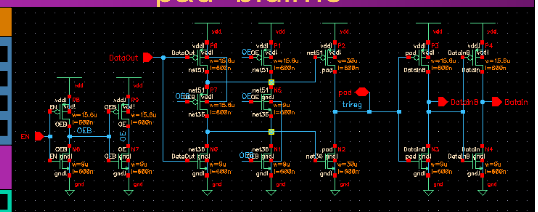
module pad_bidirhe (OEN, DO, DI, YPAD);

input OEN; (EN)
input DO; (DataOut)
output DI; (DataIn, DataInB)
inout YPAD; (pad)

If OEN = '1' , then YPAD = DO else YPAD = 'Z'.

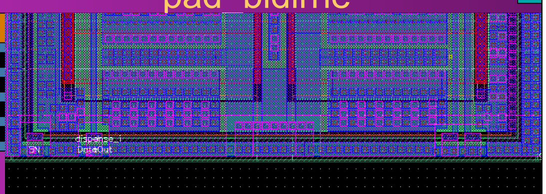
- DataOut drives a 78(p) x 45(n) inverter (30x)
- Which then drives a 200(p) x 200(n) output driver (133x)
- DataIn and DataInB come from 96(p) x 54(n) inverters (36x)
- EN drives a 16(p) x 9(n) inverter (6x)
- All signal pads are built from this one
  - All signals on are M2

pad_bidirhe



- Moderately complex pullup/pulldown structure

pad_bidirhe



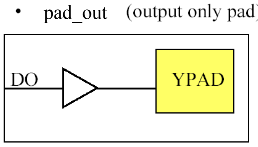
- M2 connections for EN, DataOut, DataIn, DataInB

pad_bidirhe

- Look at just the metal layers…
  - EN, DataOut, DataInB, DataIn is the order
- Middle connection is direct connection to the pad (don't use it!)
- You put metal2 shape pins over the connection points (for icc)

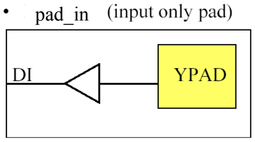
UofU Pads

- pad_out (output only pad)



module pad_out (DO, YPAD);
output YPAD; (pad)
input DO; (DataOut)

- pad_in (input only pad)

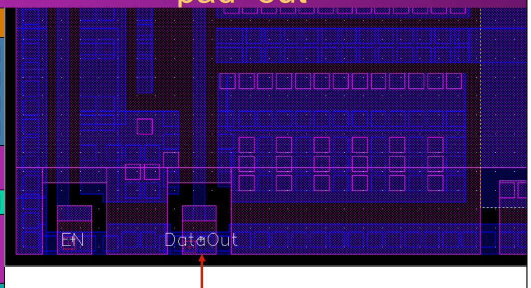


module pad_in (DI, YPAD);
output DI; (DataIn, DataInB)
input YPAD; (pad)

pad_out

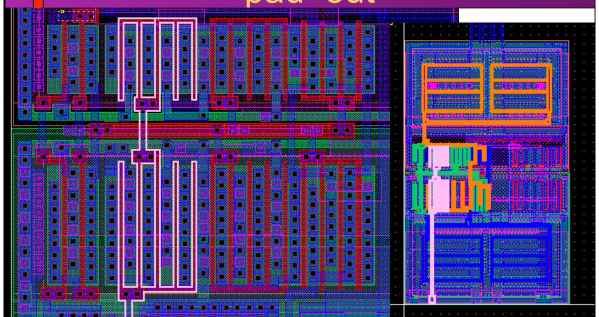
- Like pad_bidirhe but with EN already tied high for you
  - All you need to connect is DataOut

pad_out



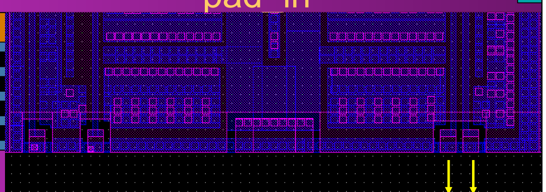
- You connect your signal to the DataOut connection into 78(p) x 45(n) inv (30x)

pad_out



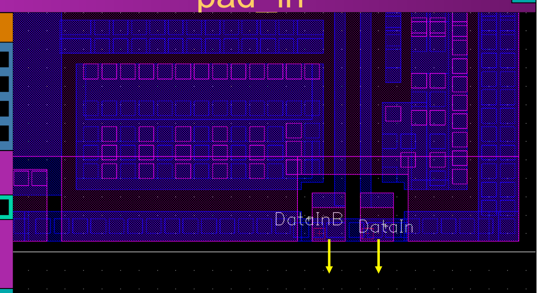
- You connect your signal to the DataOut connection into 78(p) x 45(n) inv (30x)

pad_in



- Like pad_bidirhe but with EN tied low already for you
  - Connect to the DataInB and DataIn port

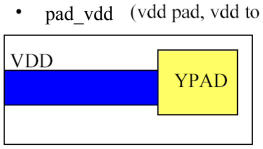
pad_in



- DataIn and DataInB provide input signals
- Driven from 94(p) x 54(n) inverters (36x)

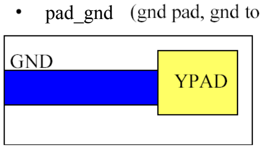
Power Supply Pads

- pad_vdd (vdd pad, vdd to core and pad rings)



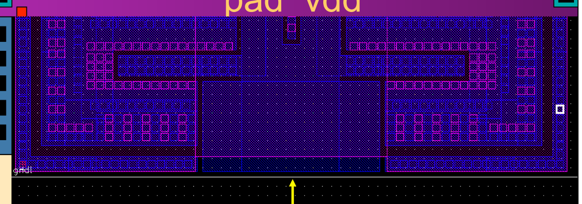
module pad_vdd (vdd, YPAD);
inout YPAD, vdd;

- pad_gnd (gnd pad, gnd to core and pad rings)



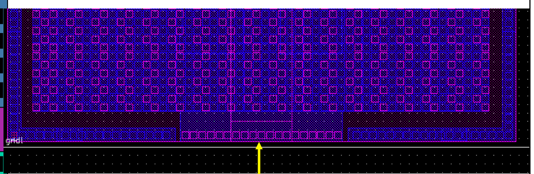
module pad_gnd (vdd, YPAD);
inout YPAD, gnd;

pad_vdd



- Vdd is on a big fat metal1 line
  - 28.8u wide

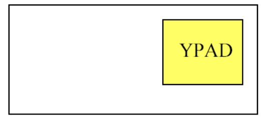
pad_gnd



- GND is also on a big fat metal1 line
  - Also 28.8u

More Pads

- PADNC (filler pad)



module PADNC ();

A no-connection pad is used to fill out the padframe if you don't need the I/Os. Could also use extra Vdd/GND pads.

You don't want to use input/output pads with unconnected inputs because these can consume power if the inputs float.

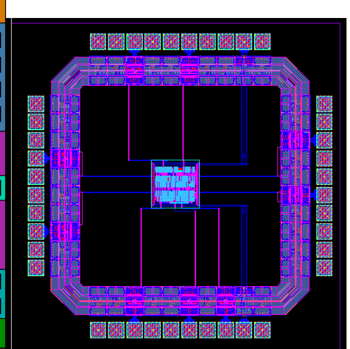
Timetable

- Final Chip Assembly
  - Due Wednesday, December 14th
  - Take the pad cells and make a pad ring
  - Connect your working core to the pad ring
- Remember that Tiny Chip Units are 1.5mm X 1.5mm and are not divisible
  - A 3.1mm X 2.8mm chip would cost 6 TCUs!
  - Preference will go to the well-simulated chips
  - Secondary preference will be for the smaller well-simulated chips

Available Frames

- Frame1_38
- Frame2h_70
- Frame2v_70
- Frame4_78, Frame4_80
  - 1,2,4 indicate how many Tiny Chip Units
  - h and v indicate horizontal and vertical for the rectangular core frames
  - _# indicates how many signal pins are available
  - Vdd and gnd are in the right spots – DON'T MOVE THEM!

Frame1_38

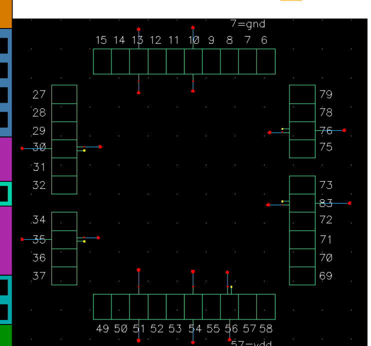


40 pins total
(38 signal pins)
10 on each side

990 x 990 core

Save room for
Routing to pads!
900 x 900
Usable core

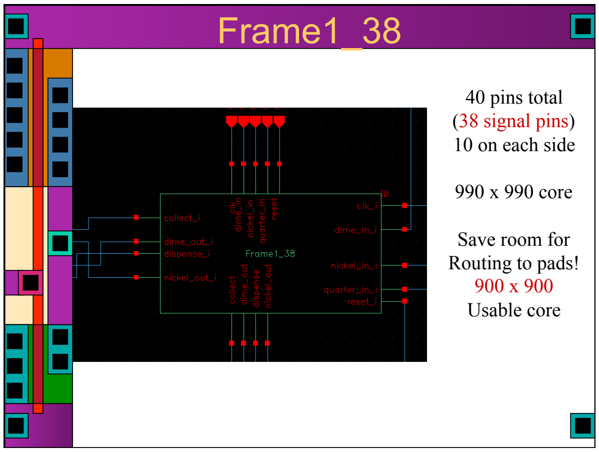
Frame1_38

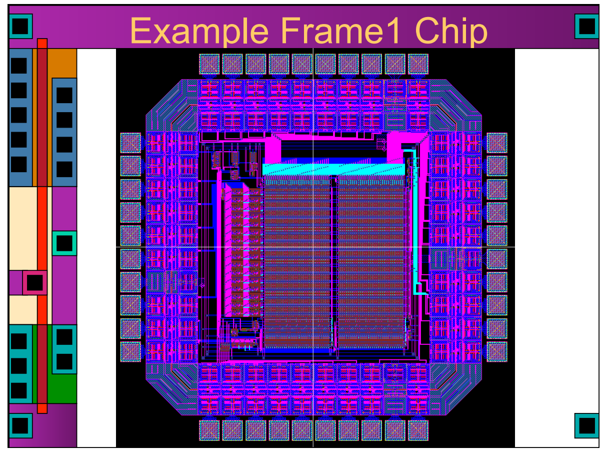


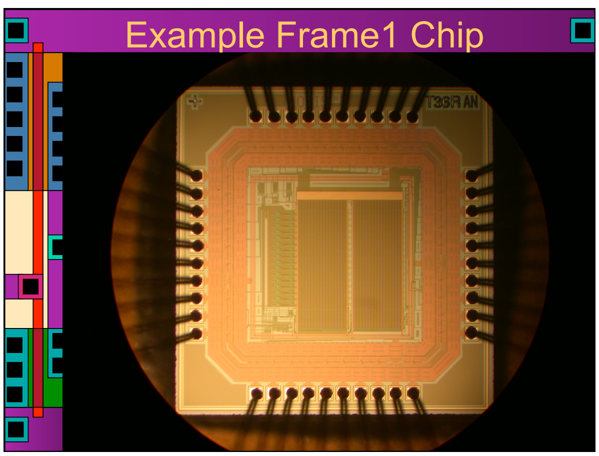
40 pins total
(38 signal pins)
10 on each side

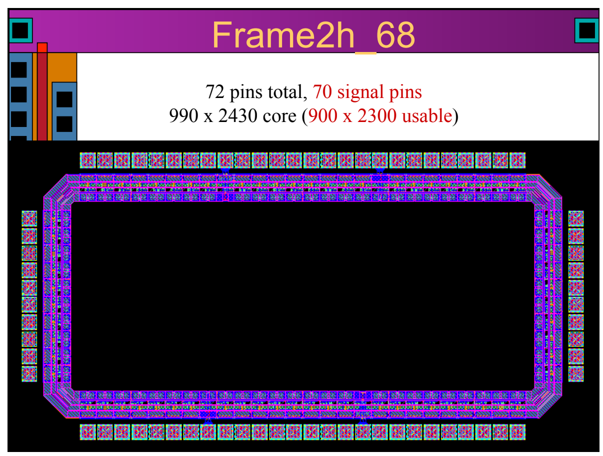
990 x 990 core

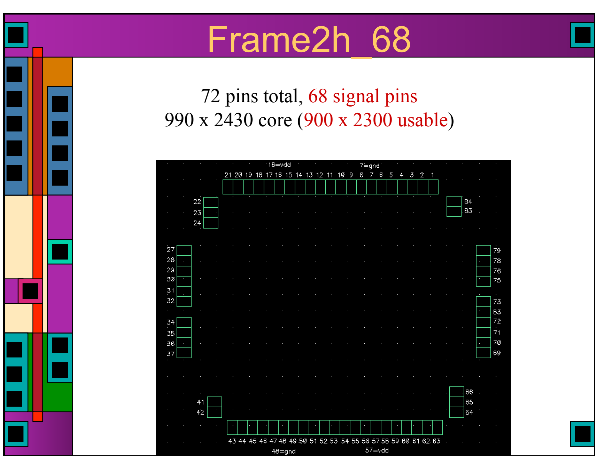
Save room for
Routing to pads!
900 x 900
Usable core

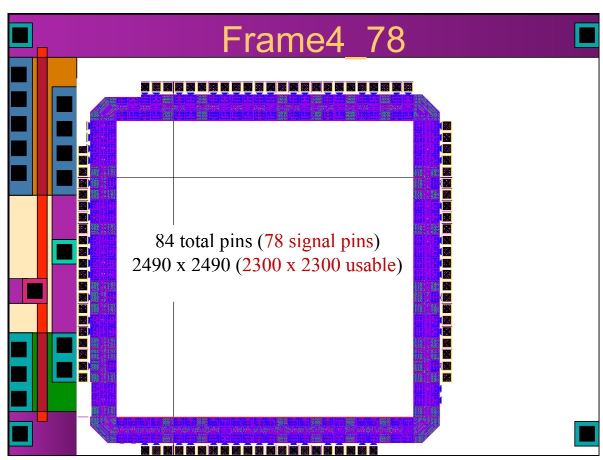

Frame1_38
40 pins total
(38 signal pins)
10 on each side
990 x 990 core
Save room for
Routing to pads!
900 x 900
Usable core


Example Frame1 Chip

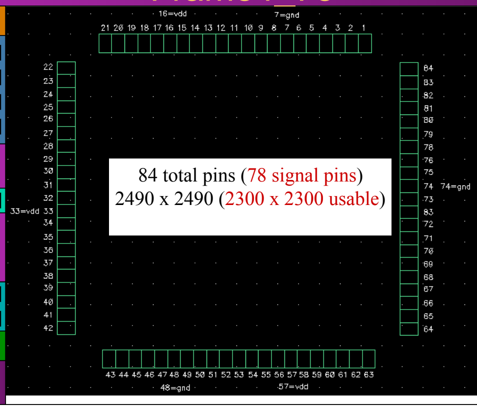

Example Frame1 Chip


Frame2h_68
72 pins total, 70 signal pins
990 x 2430 core (900 x 2300 usable)


Frame2h_68
72 pins total, 68 signal pins
990 x 2430 core (900 x 2300 usable)


Frame4_78
84 total pins (78 signal pins)
2490 x 2490 (2300 x 2300 usable)

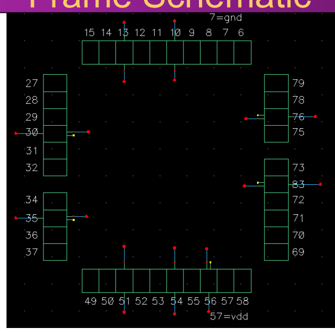
Frame4_78



How to Use the Rings

- Copy the pad ring of your choice
  - /uusoc/facility/cad_common/local/Cadence/lib/OA/UofU_Pads
  - From UofU_Pads
  - To your project directory
- Leave the pad_vdd and pad_gnd where they are!
- Select other pads, use properties to change to the pad type you want
  - DON'T move them!
  - Use pad_bidirhe, pad_out, and pad_in

Frame Schematic

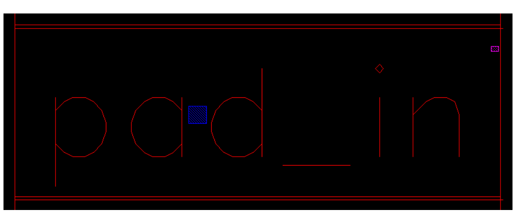


- Frame1_38 with the right pads for the drink_machine

Frame layout

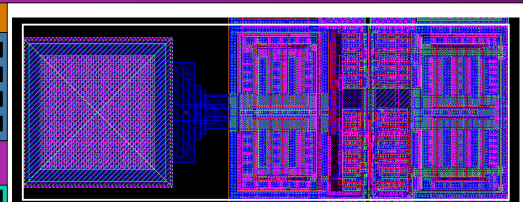
- Frame1_38 with the right pads for the drink_machine

Pins



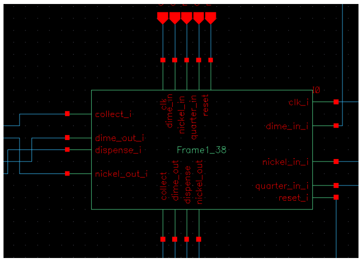
- Frame1_38 with the right pads for the drink_machine

Pins



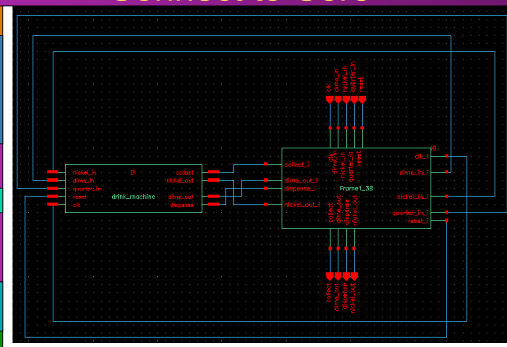
- Frame1_38 with the right pads for the drink_machine

Frame symbol



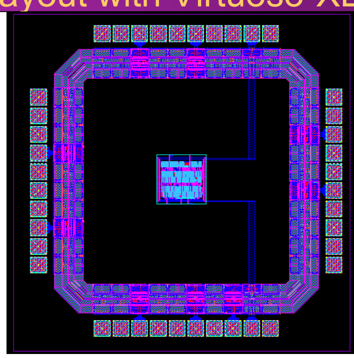
- Frame1_38 with the right pads for the drink_machine

Connect to Core



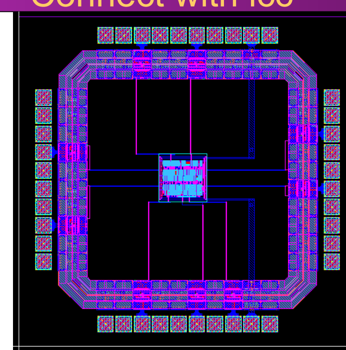
- Use this to start the ccar routing process

Layout with Virtuoso-XL



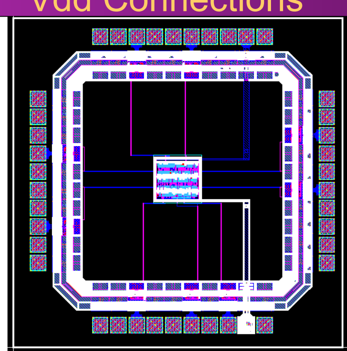
- Do placement, and connect vdd and gnd

Connect with icc



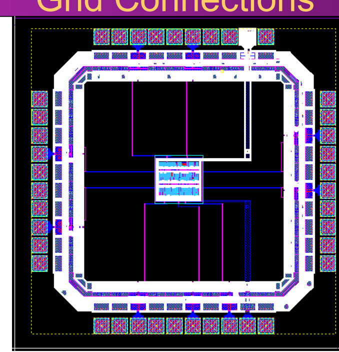
- Let ccar the routing

Vdd Connections



- Notice how the pad frame is connected

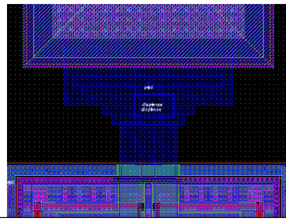
Gnd Connections



- Notice how the pad frame is connected

Now Simulate the Whole Chip

- Use essentially the same testbench that you used for the core
  - This time you'll be simulating with the pads in place
  - You'll need to place one more set of pins so that the wholechip cell has connection points



What Does This Mean?

- For now, concentrate on getting your chip core assembled, working, DRCed and LVSed.
  - You need a working core before you need pads!
- Make sure your core fits in the pad ring that you want to use
- Then, use vcar to assemble the frame and core
  - Simulate, LVS, DRC with the whole thing!

Output to GDS (Stream)

- Once everything is completely finished, you need to export the whole chip to GDS (stream) format
  - Use the export->stream function in CIW
  - use stream4gds.map as the Layer Map Table
  - From /uusoc/facility/cad_common/local/class/6710/F11/cadence/map_files
- Fill in Library Name, Top Cell Name and Output File Name
- I will read this GDS file in and re-DRC that layout...

Fabrication Schedule

- MOSIS educational run
  - Chips that go into the fab queue need to be absolutely and completely ready to go:
    - Friday Jan 10th at the LATEST
  - There are a few more steps that projects need to go through to make them fab-ready even after DRC/LVS
  - If you make logos and names, those have to pass DRC too!
    - Metal3 is recommended for logos…

Final Report

- Final Report, due Wed, December 14
- Three parts:
  - First: Technical Paper (about project)
    - Not more than 10 pages
    - IEEE two-column format
    - Describe what makes your chip interesting
    - This is a self-contained paper of the form that might be submitted to a conference or journal
  - Second: Project Details
    - Floorplan, pinout, and system block diagram
    - Schematics and layouts for all major parts
    - A table of contents or readme guide

Final Report

- Third: Standard Cells
  - Standard Cell layouts, schematics, etc.
  - User's guide
  - Email .lib, .lef, and .v files to me at teach-cs6710@list.eng.utah.edu
  - Also tell me where your Cadence libraries are.
  - I can slurp up the cell libraries if they are readable by your group.