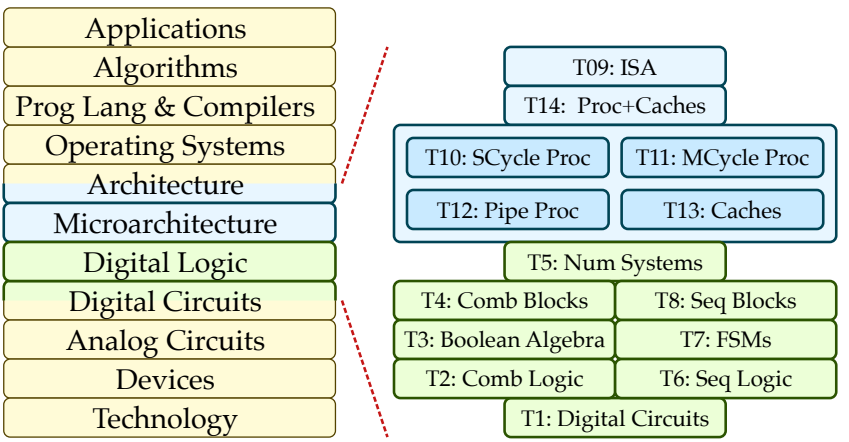
ECE 2300 Digital Logic and Computer Organization Fall 2025

Section 13: FPGAs to ASICs

School of Electrical and Computer Engineering
Cornell University

revision: 2025-12-05-12-44



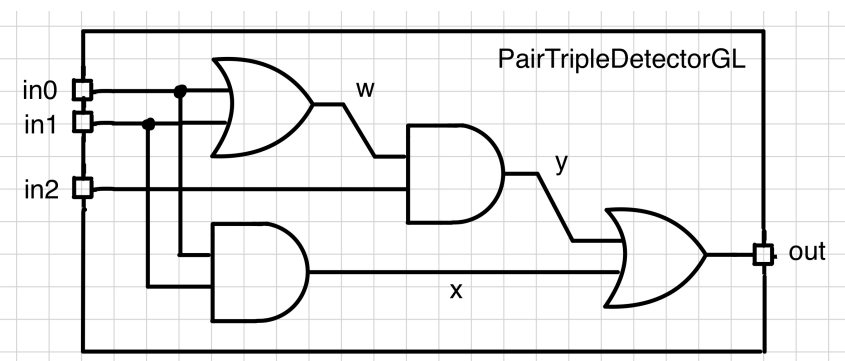
- Use VS Code on workstation to connect to `ecelinux`
- Clone discussion section GitHub repo and look at provided files

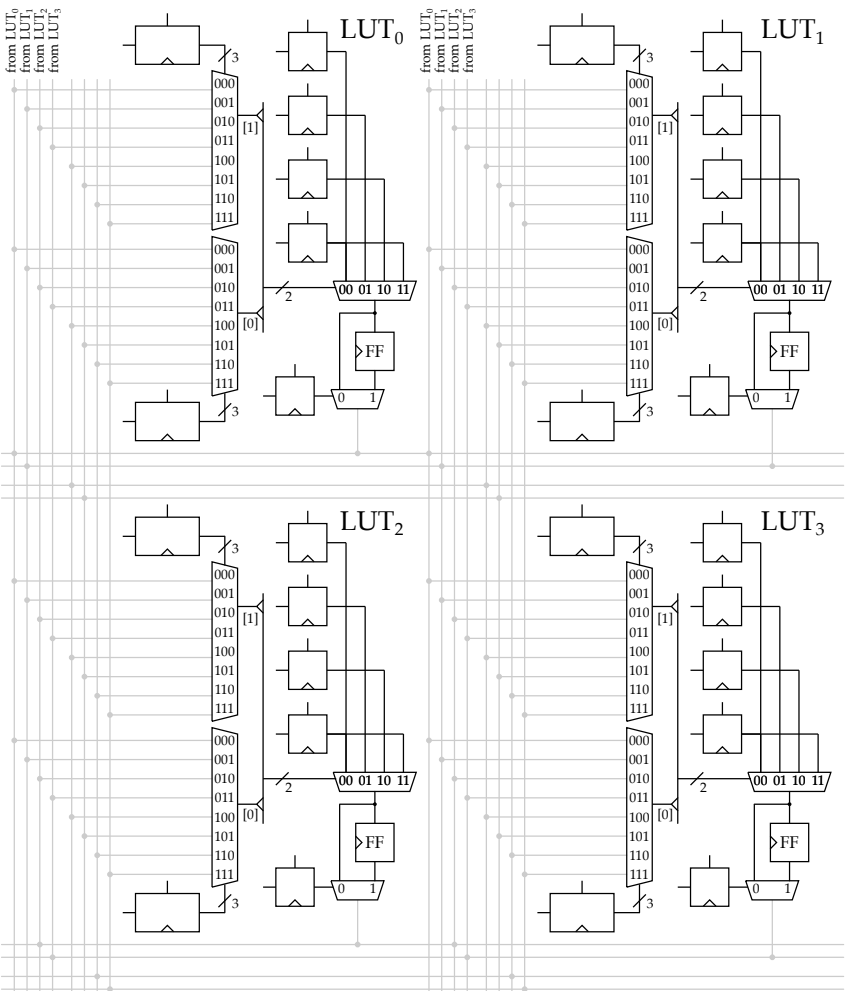
```
% source setup-ece2300.sh
% source setup-ece2300-asic.sh
% mkdir -p $HOME/ece2300
% cd $HOME/ece2300
% git clone git@github.com:cornell-ece2300/ece2300-sec13 sec13
% cd sec13
% tree
```

- Run provided tests

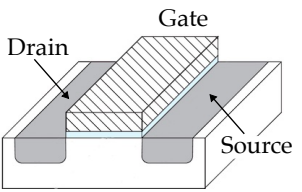
```
% cd $HOME/ece2300/sec13
% mkdir build
% cd build
% ../configure
% make check
```

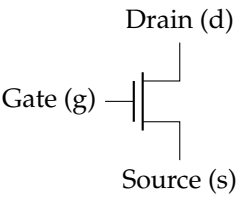
1. Field Programmable Gate Arrays



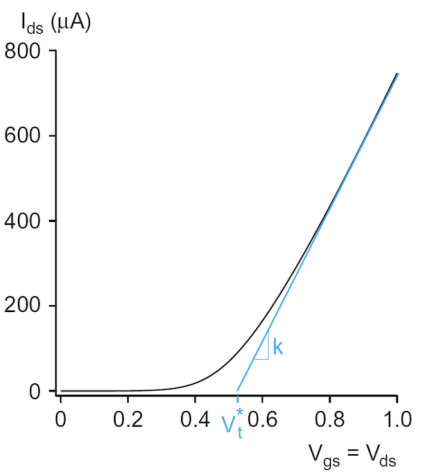

from LUT0
from LUT1
from LUT2
from LUT3
LUT0
LUT1
LUT2
LUT3
000
001
010
011
100
101
110
111
[1]
[0]
3
2
00 01 10 11
FF
0 1

2. Metal-Oxide Semiconductor Field-Effect Transistors





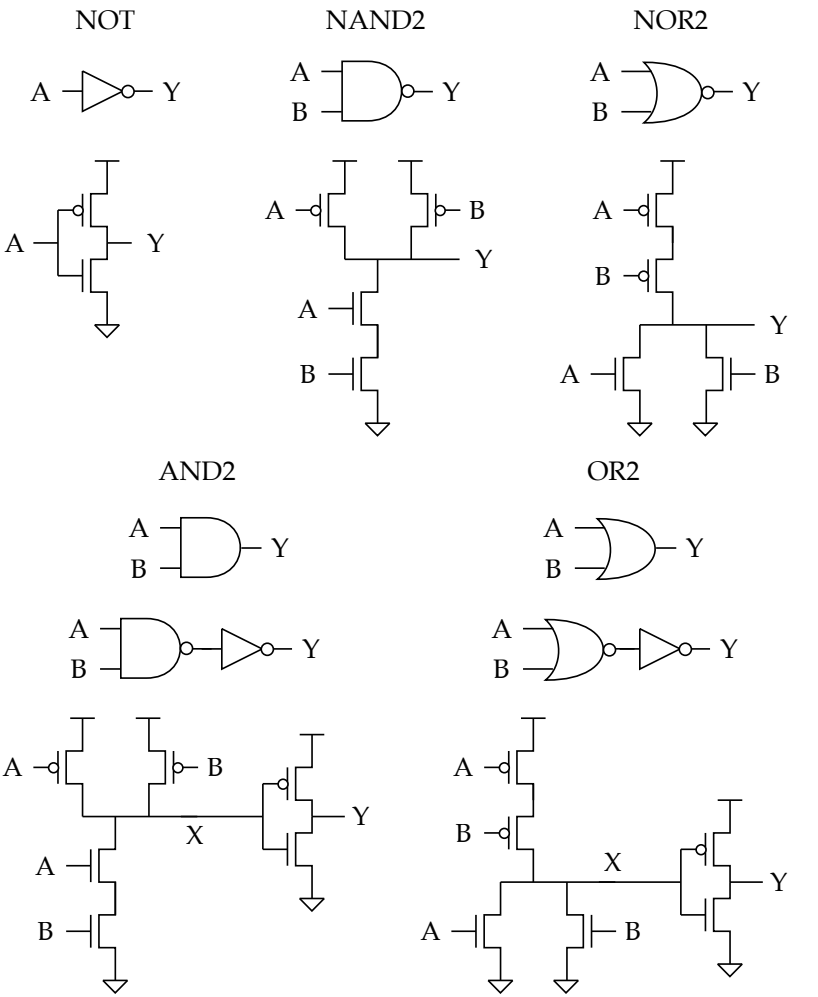
- MOS: Need three materials to make a transistor
  - _____ strong conductor of current
  - _____ insulator (does not conduct)
  - _____ conduction can be controlled
- FET: Electric field causes conduction between source and drain



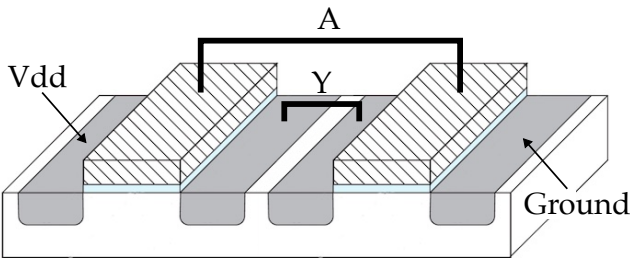
- MOSFET operation
  - As gate voltage increases, current from drain to source also increases
  - At specific threshold voltage, current starts to increase very rapidly
- MOSFET analog applications
  - Various amplifiers
  - Operational amplifiers
  - Current sources and mirrors
  - Voltage-controlled oscillators
  - Voltage regulators
  - Filters

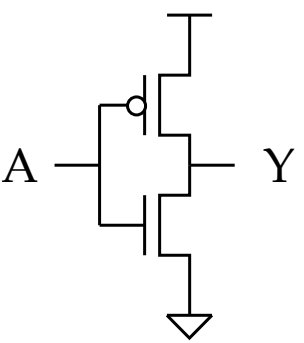
We will simply use transistors as voltage controlled switches

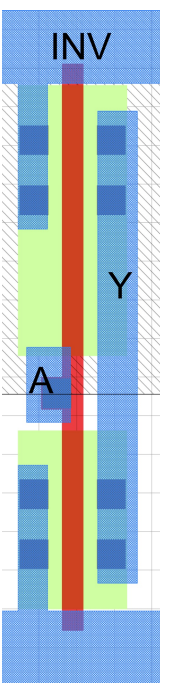
3. Digital Circuits

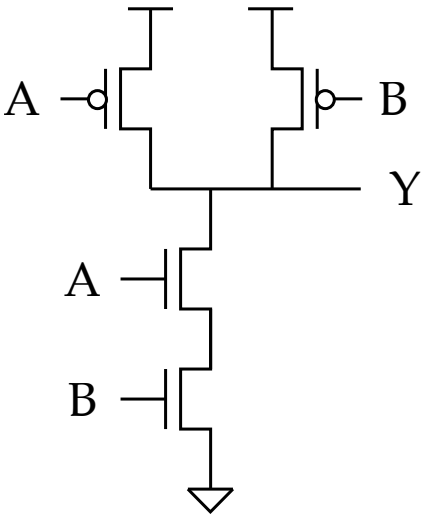


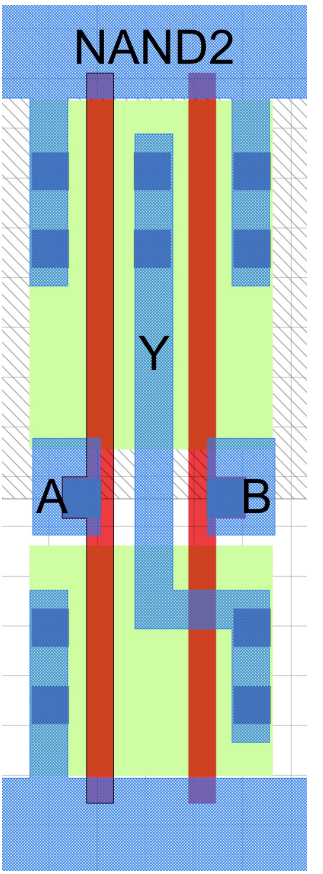
4. Layout

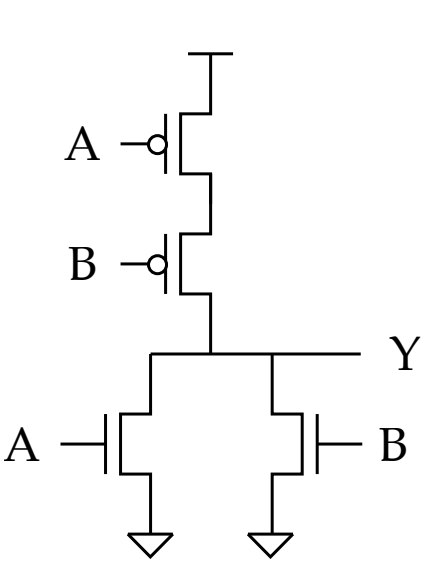


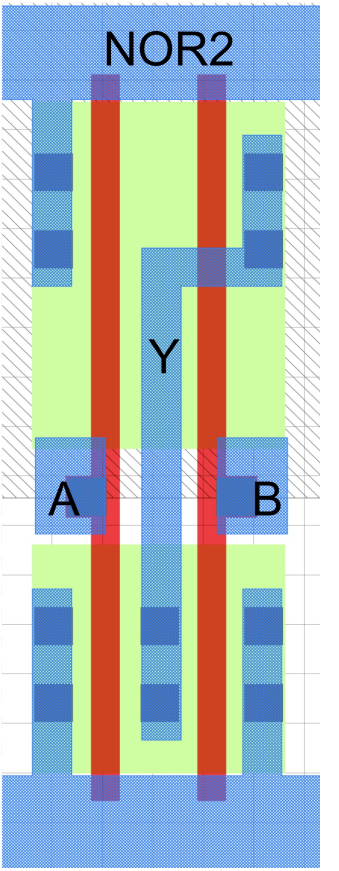




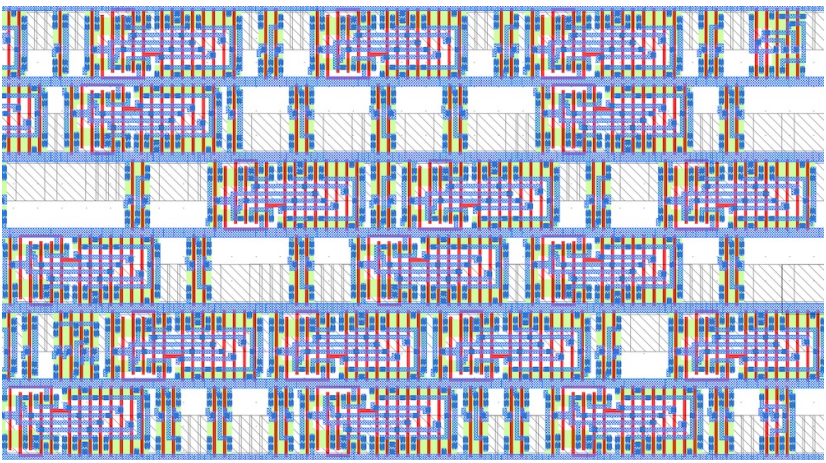

A
B
Y
A
B


NAND2
Y
A
B

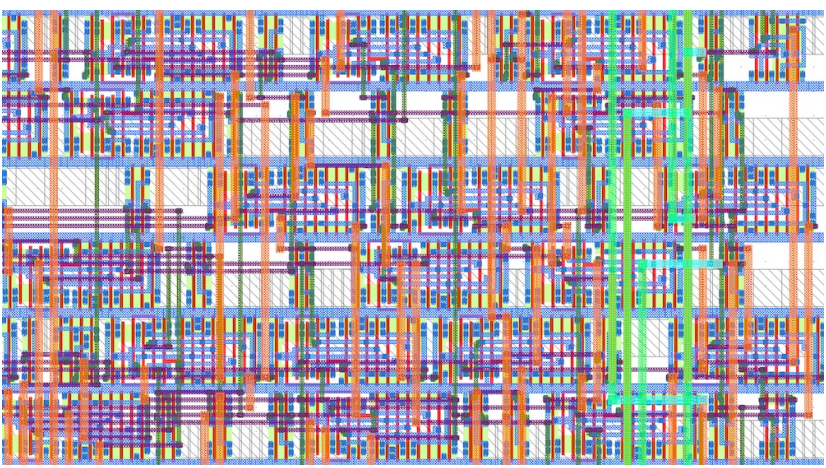

A
B
Y
A
B


NOR2
Y
A
B

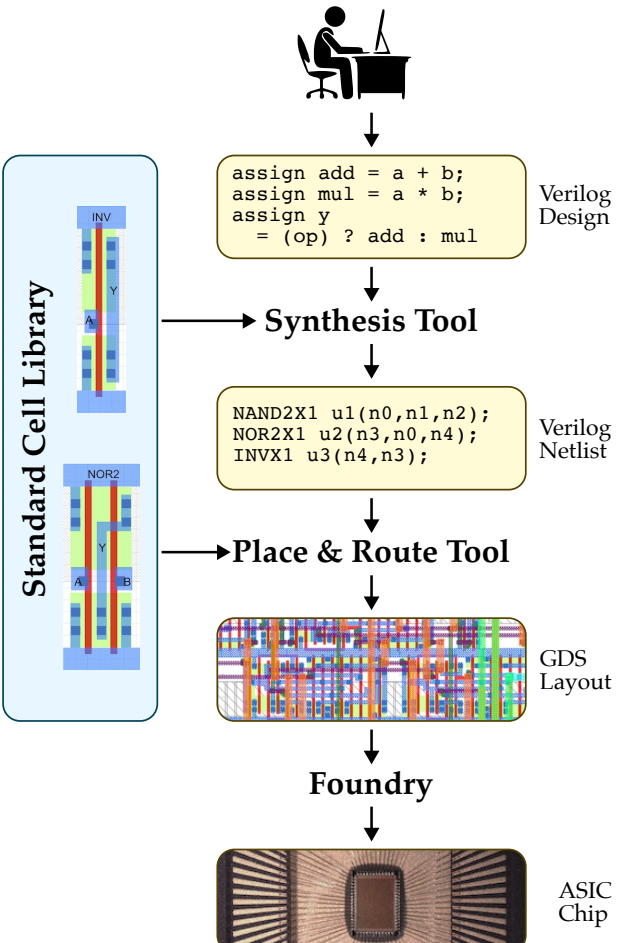
Standard cells placed into rows



Wires between standard cells routed on metal layers

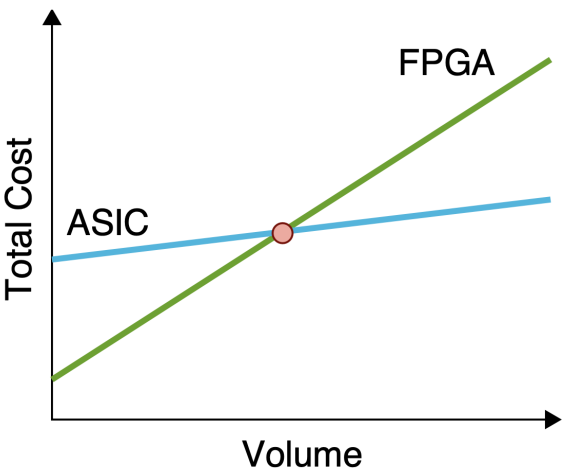


5. ASIC Toolflow



6. FPGA vs ASIC

	NRE $	Marginal $	Area	Delay	Generality
Software					
FPGA					
ASIC					



7. Hands-On Demo

- Let's start by creating the flow for the pair triple detector

```
% mkdir -p $HOME/ece2300/sec13/build-ptd
% cd $HOME/ece2300/sec13/build-ptd
% pyhflow ../asic/designs/sec13-ptd.yml
```

- Now let's synthesize the design

```
% cd $HOME/ece2300/sec13/build-ptd
% ./01-synopsys-dc-synth/run
% cat 01-synopsys-dc-synth/post-synth.v
```

- Now let's place and route the synthesized gate-level netlist

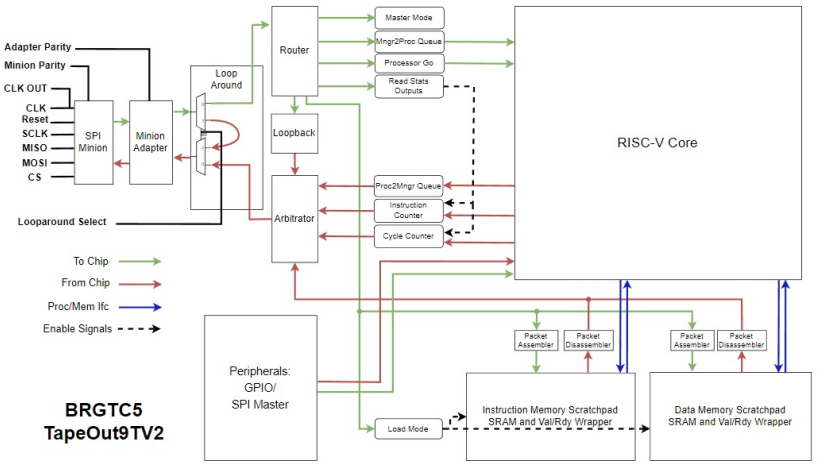
```
% cd $HOME/ece2300/sec13/build-ptd
% ./02-cadence-innovus-pnr/run
% cat ./02-cadence-innovus-pnr/timing-setup.rpt
% cat ./02-cadence-innovus-pnr/area.rpt
```

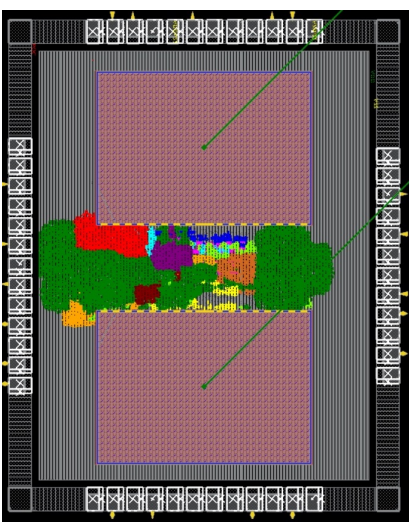
- Right click on `./02-cadence-innovus-pnr/post-pnr.gds` in the VS Code file explorer to download the final layout to the workstation
- Open using Klayout on the workstation
- Now try pushing the RTL calculator through the ASIC toolflow

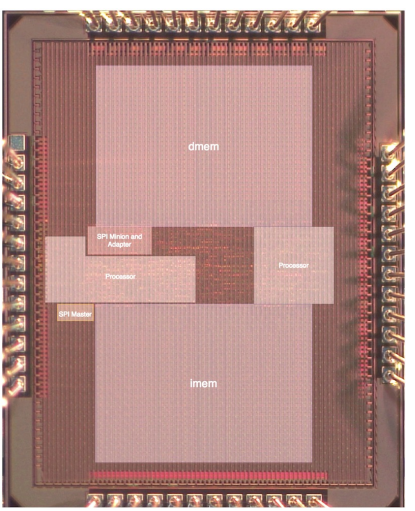
```
% mkdir -p $HOME/ece2300/sec13/build-calc
% cd $HOME/ece2300/sec13/build-calc
% pyhflow ../asic/designs/sec13-calc.yml
% ./run-flow
```

- Look at the post-synthesis gate-level netlist
- Look at the post-pnr timing and reports
- Look at the final layout using Klayout

8. BRGTC5


Adapter Parity
Minion Parity
CLK OUT
CLK
Reset
SCLK
MISO
MOSI
CS
SPI Minion
Minion Adapter
Loop Around
Router
Loopback
Arbitrator
Looparound Select
Master Mode
Mngr2Proc Queue
Processor Go
Read Stats Outputs
Proc2Mngr Queue
Instruction Counter
Cycle Counter
RISC-V Core
To Chip
From Chip
Proc/Mem Ifc
Enable Signals
Packet Assembler
Packet Disassembler
Packet Assembler
Packet Disassembler
Peripherals: GPIO/ SPI Master
Load Mode
Instruction Memory Scratchpad SRAM and Val/Rdy Wrapper
Data Memory Scratchpad SRAM and Val/Rdy Wrapper
BRGTC5
TapeOut9TV2




dmem
SPI Minion and Adapter
Processor
Processor
SPI Master
imem