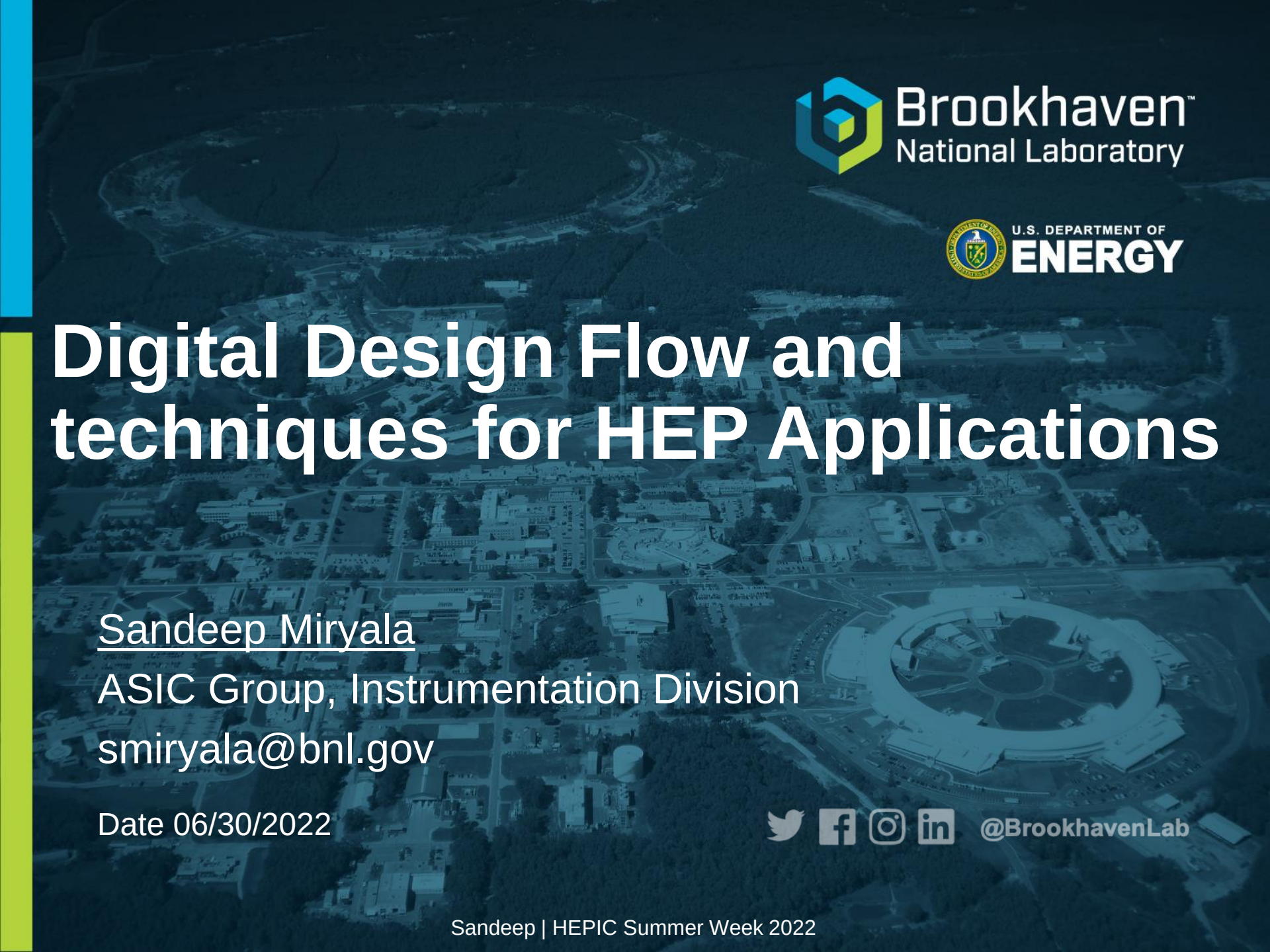


Digital Design Flow and techniques for HEP Applications

Sandeep Miryala
ASIC Group, Instrumentation Division
smiryala@bnl.gov

Date 06/30/2022

@BrookhavenLab

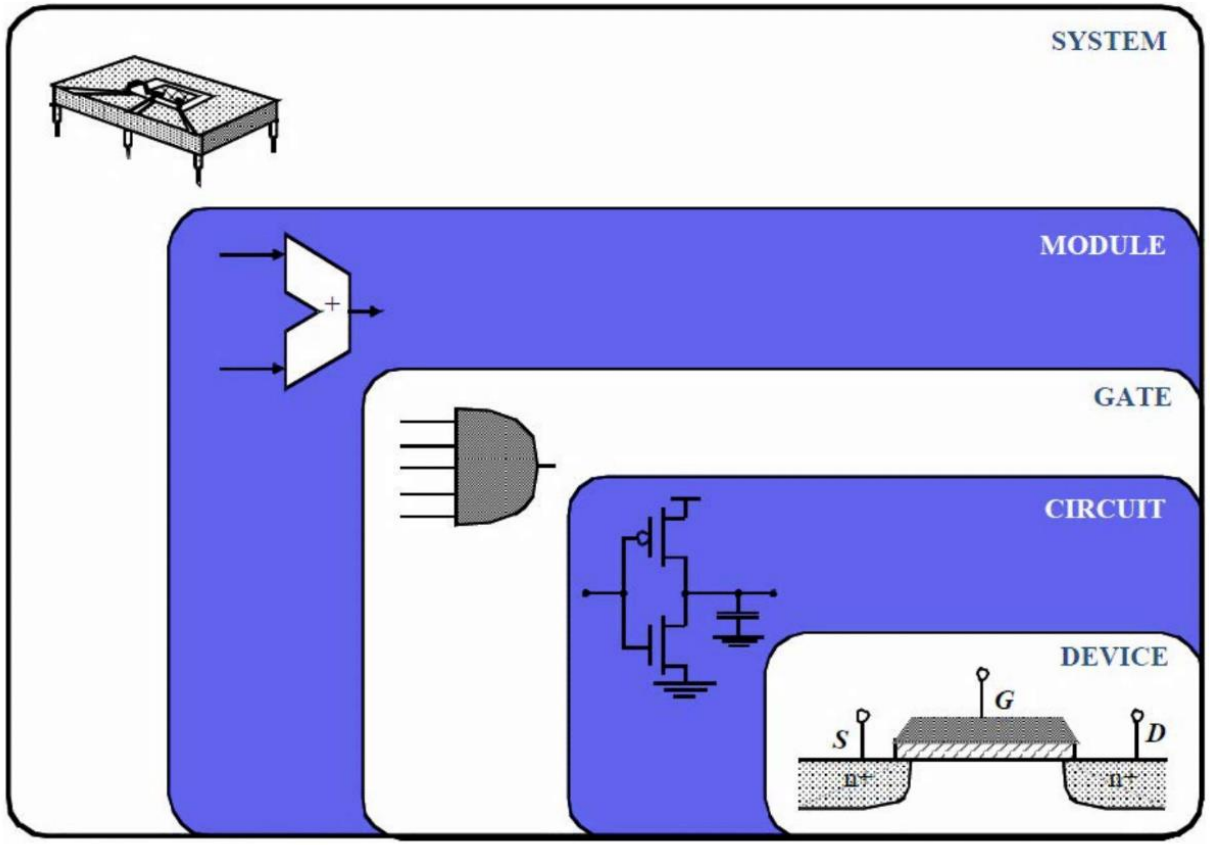
Outline

- Introduction
  - Semi-Custom Design Flow
  - Static Timing Analysis (STA)
  - Timing Library overview
- Synthesis overview
- Placement and Routing (PnR) Overview
- HEP ASIC requirements
  - TID tolerant digital design
  - SEE tolerant digital design
  - ASICs operating for cryogenic temperature
- AI on chip or Edge Computing
- DEMO of design flow using a simple counter

Custom vs Semi-Custom

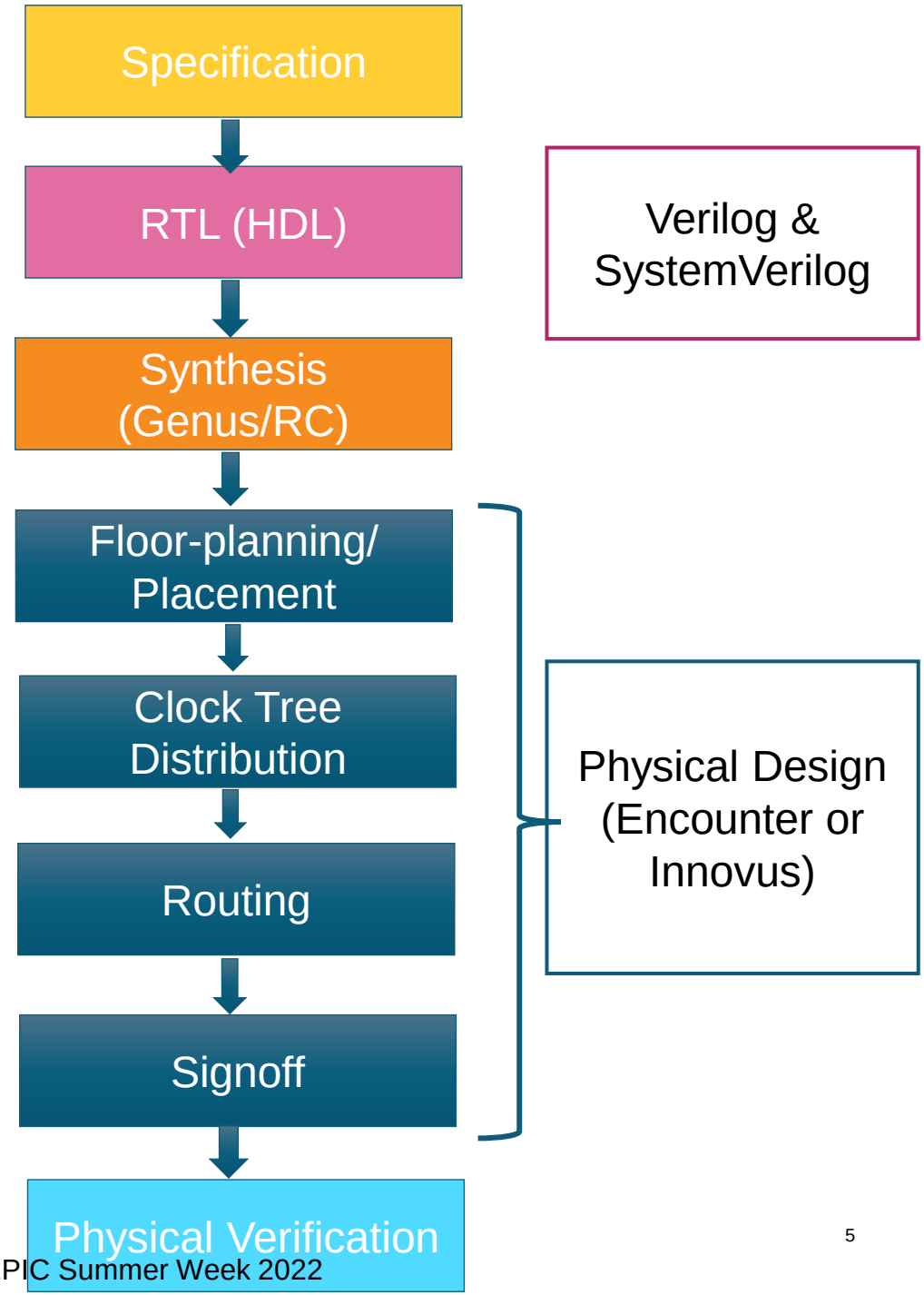
- Analog designs are often a full custom design
  - Designed from transistor level, approprate device sizing
  - Amplifiers, Phase Locked Loops (PLLs), Data Converters, Bias References, Drivers, High speed serializers
- Digital design is mostly a semi-custom design
  - Standard cell libraries as part of Process Design Kit (PDK)
  - 65nm process has ~730 cells in a standard cell library
  - Various standard cell libraries
    - Differ in terms of height of a standard cell tracks (7T, 9T, 12T)
    - Vt flavor of device (Standard, Low and High threshold voltage)
  - Command decoders, counters, pattern generators, encoders, error detection and correction alg., data compression, scrambler, frame formation, FIFOs Etc.

Design Abstraction

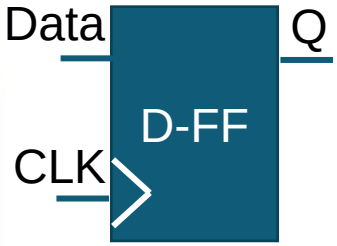


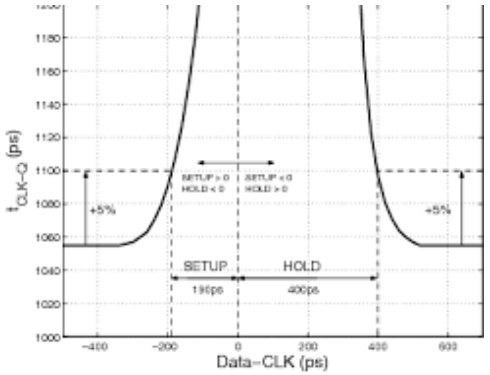
Semi-Custom Digital Design Flow

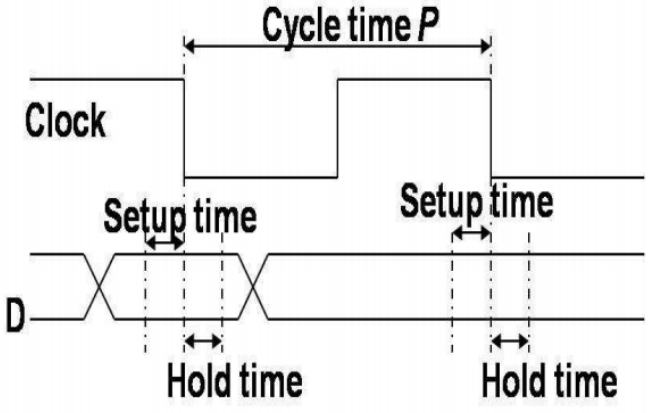
- Use of CAD tools at every stage of the design flow
- Automation is through TCL scripts
- Static Timing Analysis (STA) is run at all the stages



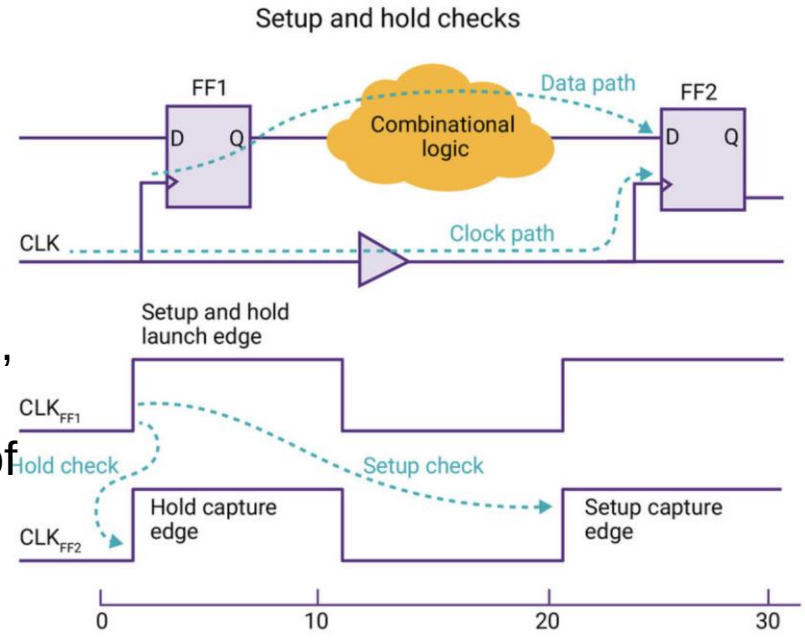
Static Timing Analysis







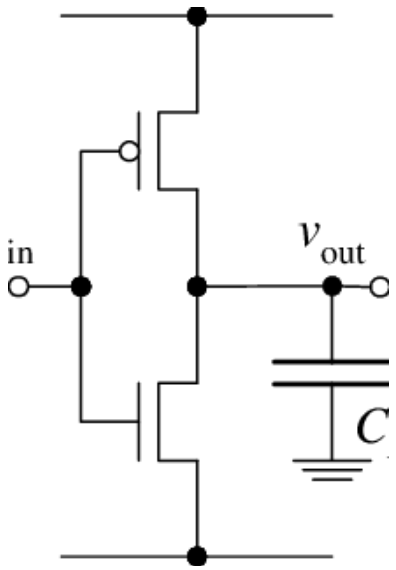
- Static timing analysis (STA) is a method of validating the timing performance of a design by checking all possible paths for timing violations
- STA breaks a design down into timing paths, calculates the signal propagation delay along each path, and checks for violations of timing constraints
- Algorithms determine the max and min propagation delays of the data path



- setup slack = (cycle_time - t_{ckq} - t_{pd} -t_{su}) - clock_skew
- hold slack = (t_{ckq} + t_{pd} -t_h) - clock_skew

Brookhaven National Laboratory

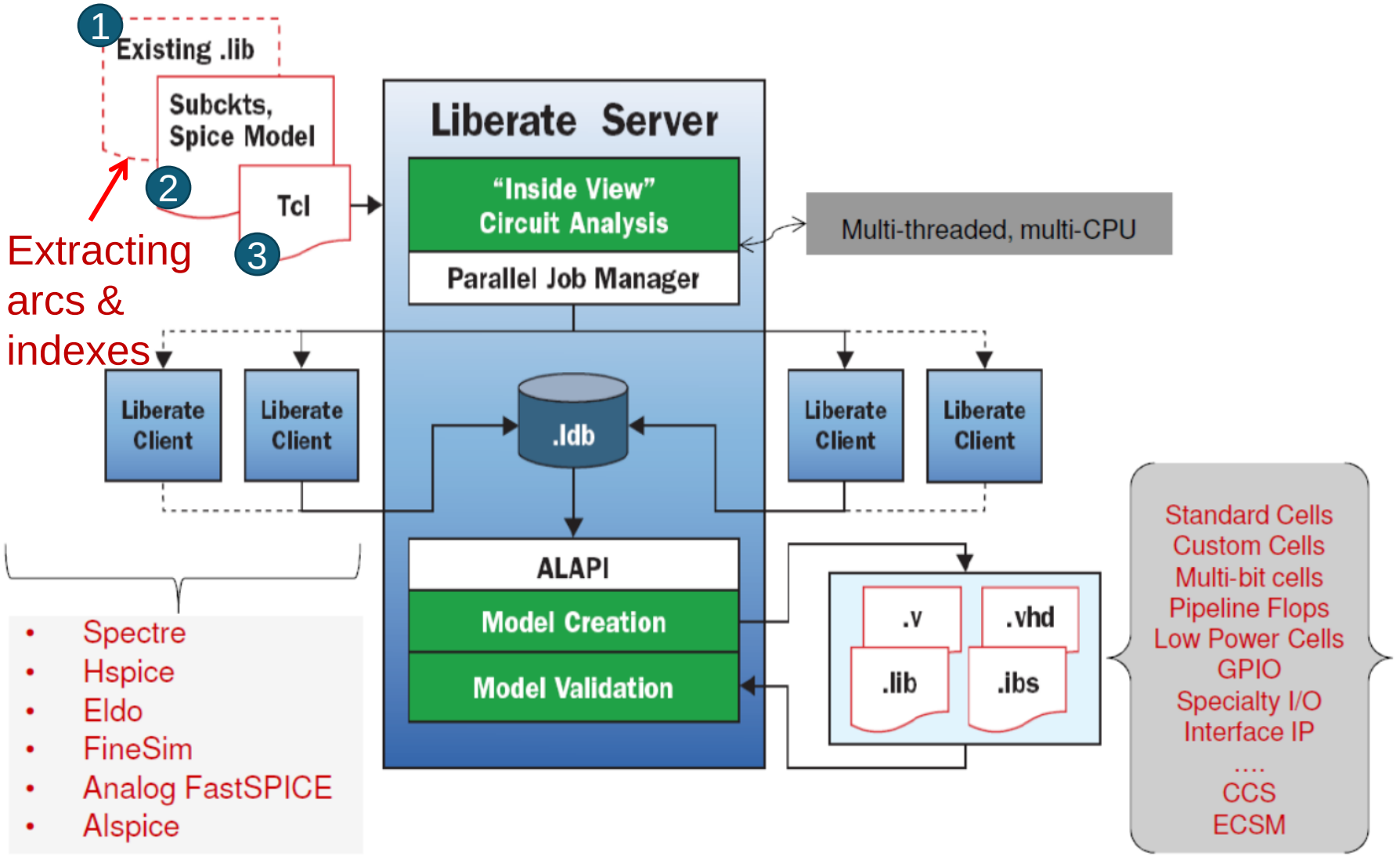
Timing Library Information (.lib)



- Timing
  - Combinational
    - Delay & Transition Time
  - Sequential
    - Delay & Transition Time
    - Hold and Setup
    - Recovery and Removal
- Power
  - Leakage
  - Internal Power
- Noise
  - Signal integrity analysis
- Data is arranged in Look Up Table (LUT) format
  - Sample 4 x 4 LUT

Index	cl1	cl2	cl3	cl4
Trin1	D11	D12	D13	D14
Trin2	D21	D22	D23	D24
Trin3	D31	D32	D33	D34
Trin4	D41	D42	D43	D44

Liberate Tool Flow



Define_Cell and Define_Arc

```
if {[ALAPI_active_cell "AND2_A_XL_TSMC_HVTN"]} {
define_cell \
     -input { A B } \
     -output { Z } \
     -pinlist { A B Z } \
     -delay delay_template_7x7 \
     -power power_template_7x7 \
     -si_immunity si_immunity_template_7x7 \
     AND2_A_XL_TSMC_HVTN
```

Cell Definition

```
define_leakage -when "!A&!B" AND2_A_XL_TSMC_HVTN
define_leakage -when "A&!B" AND2_A_XL_TSMC_HVTN
define_leakage -when "!A&B" AND2_A_XL_TSMC_HVTN
define_leakage -when "A&B" AND2_A_XL_TSMC_HVTN
```

Leakage

```
# power arcs from  => A  hidden
define_arc \
     -type hidden \
     -vector {Rxx} \
     -pin A \
     AND2_A_XL_TSMC_HVTN

# power arcs from  => A  hidden
define_arc \
     -type hidden \
     -vector {Fxx} \
     -pin A \
     AND2_A_XL_TSMC_HVTN

# power arcs from  => B  hidden
define_arc \
     -type hidden \
     -vector {xRx} \
     -pin B \
     AND2_A_XL_TSMC_HVTN

# power arcs from  => B  hidden
define_arc \
     -type hidden \
     -vector {xFx} \
     -pin B \
     AND2_A_XL_TSMC_HVTN
```

Dynamic power arcs

```
# delay arcs from A => Z positive_unate combinational
define_arc \
     -vector {RxR} \
     -related_pin A \
     -pin Z \
     AND2_A_XL_TSMC_HVTN

# delay arcs from A => Z positive_unate combinational
define_arc \
     -vector {FxF} \
     -related_pin A \
     -pin Z \
     AND2_A_XL_TSMC_HVTN

# delay arcs from B => Z positive_unate combinational
define_arc \
     -vector {xRR} \
     -related_pin B \
     -pin Z \
     AND2_A_XL_TSMC_HVTN

# delay arcs from B => Z positive_unate combinational
define_arc \
     -vector {xFF} \
     -related_pin B \
     -pin Z \
     AND2_A_XL_TSMC_HVTN

}
```

Delay arcs

```
define_template -type delay \
      -index_1 {0.0224 0.0608 0.12 0.32 0.72 1.6 3.0 } \
      -index_2 {0.0014 0.003 0.0062 0.0125 0.0251 0.0504 0.101 } \
      delay_template_7x7

define_template -type power \
      -index_1 {0.0224 0.0608 0.12 0.32 0.72 1.6 3.0 } \
      -index_2 {0.0014 0.003 0.0062 0.0125 0.0251 0.0504 0.101 } \
      power_template_7x7

define_template -type si_immunity \
      -index_1 {0.224 0.608 1.2 3.2 7.2 16.0 30.0 } \
      -index_2 {0.0014 0.003 0.0062 0.0125 0.0251 0.0504 0.101 } \
      si_immunity_template_7x7
```

Templates

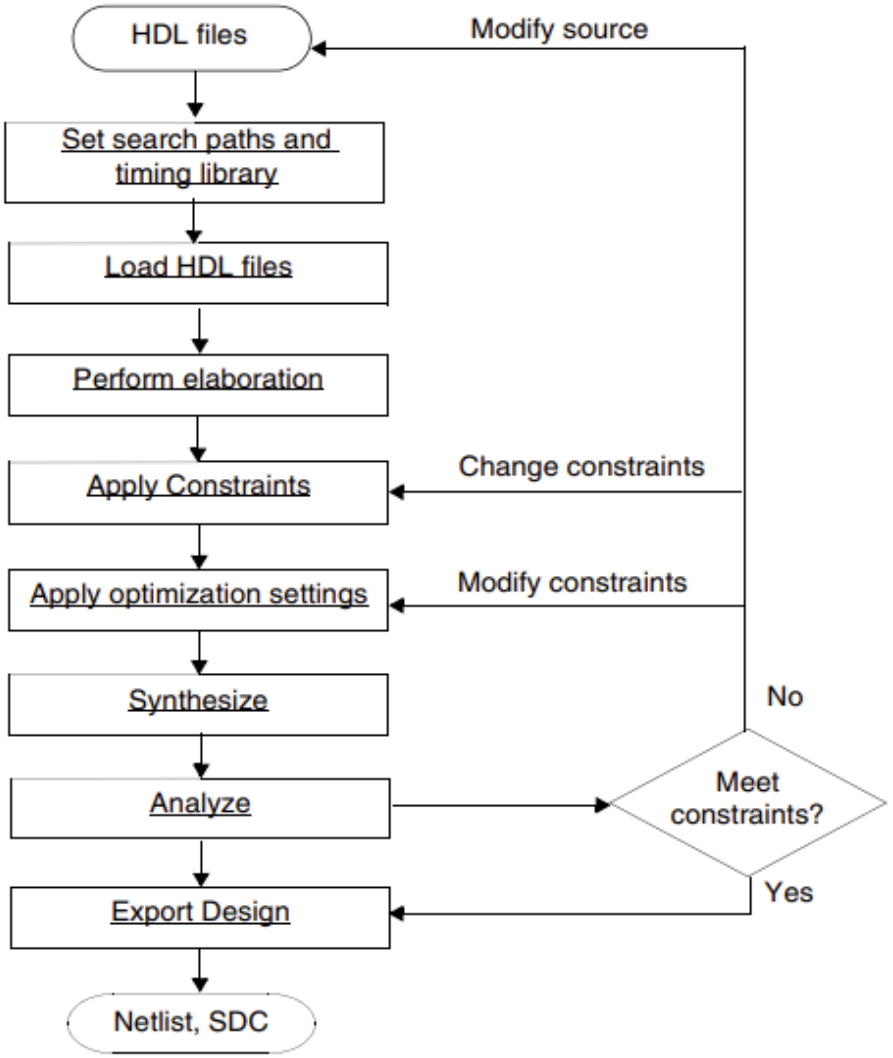
Brookhaven National Laboratory

Synthesis

Synopsis Design Constraints (SDC)

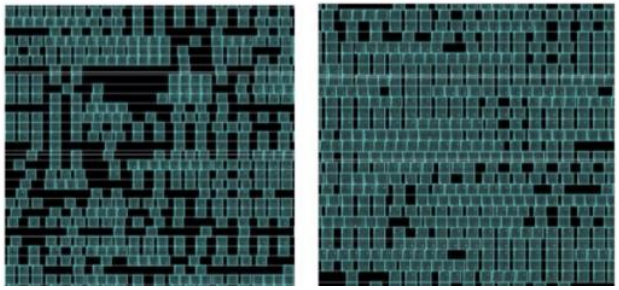
- Create Clocks
- IO delays
- Loads
- Timing exceptions
- Don’t touch nets

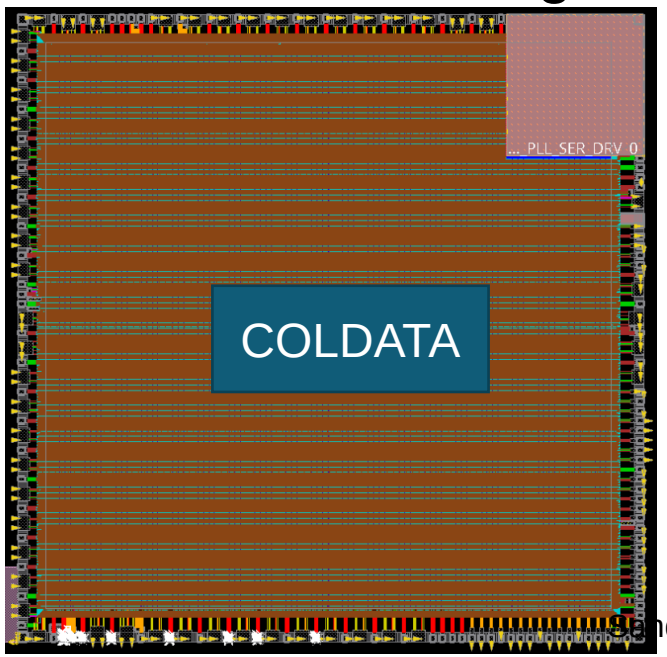
Genus Generic Synthesis Flow

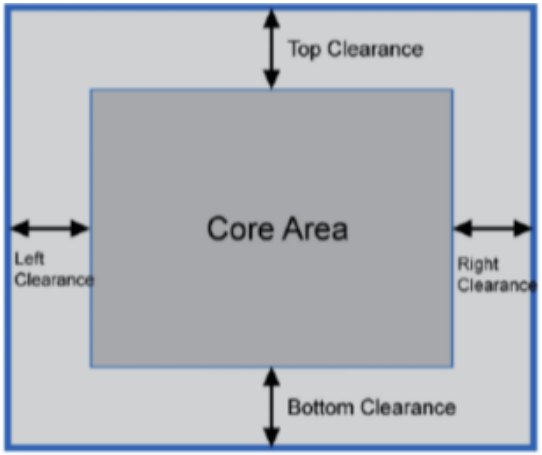


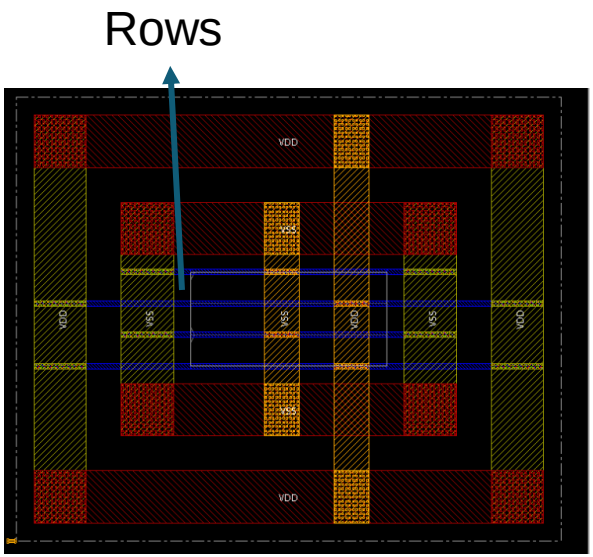
Floor planning

- ❑ Initialize with aspect ratio (AR), core utilization
- ❑ Initialize Row Configuration and Cell Orientation
- ❑ Specify core to pad/IO spacing
- ❑ Pins/Pads placement
- ❑ Macro placement and orientation
- ❑ Blockage Management
- ❑ Power Planning





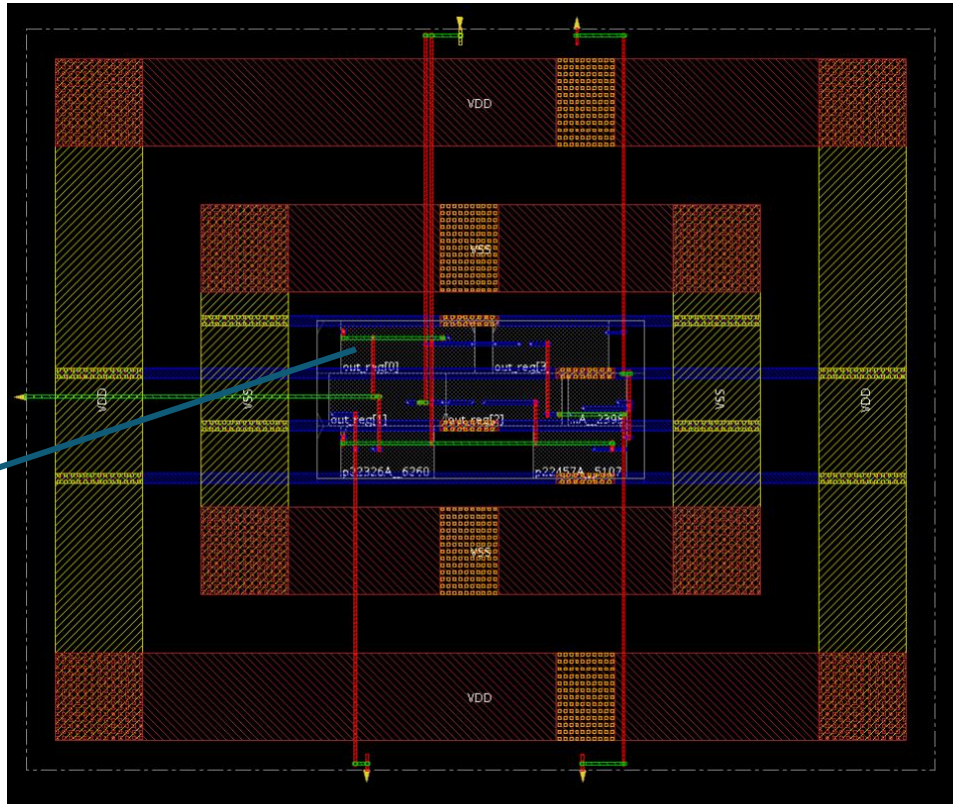




Placement

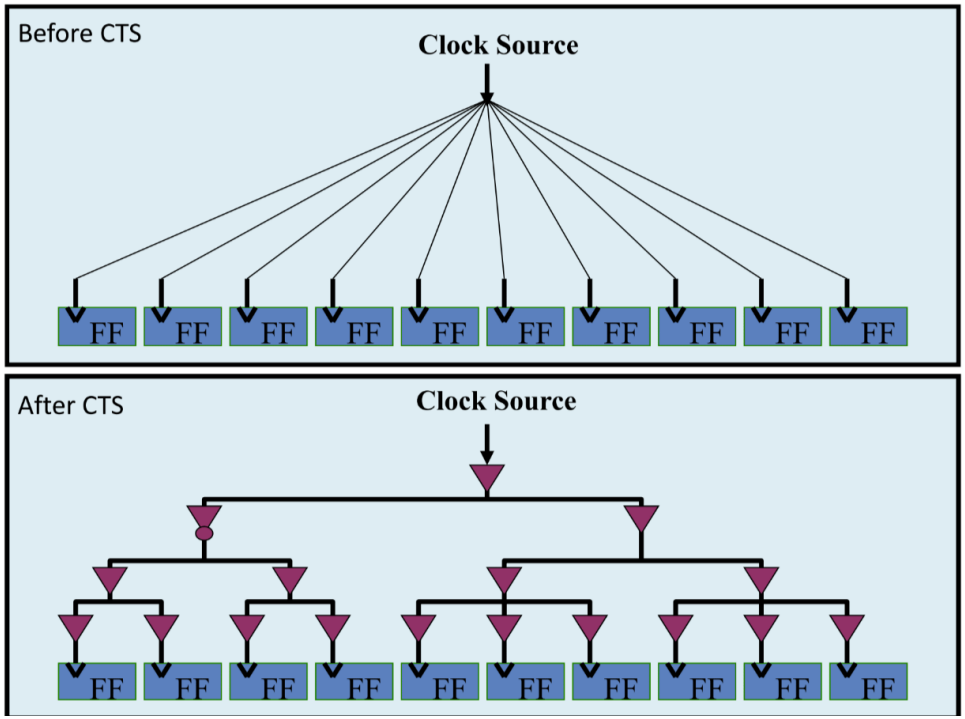
- Automated standard cell placement for placing the standard cells in placement tracks
- Placement Objectives
  - Total wire length
  - Routability
  - Performance
  - Power
  - Heat distribution

Standard cells in the rows



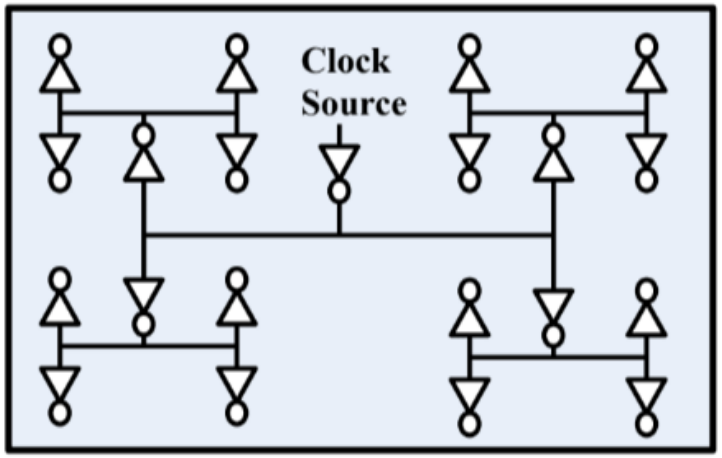
Clock Tree Synthesis

- Clock tree objectives and associated issues
  - Clock skew
  - Long clock insertion delay
  - Heavy clock net loading
  - Clock is power hungry
  - Clock to signal coupling effect

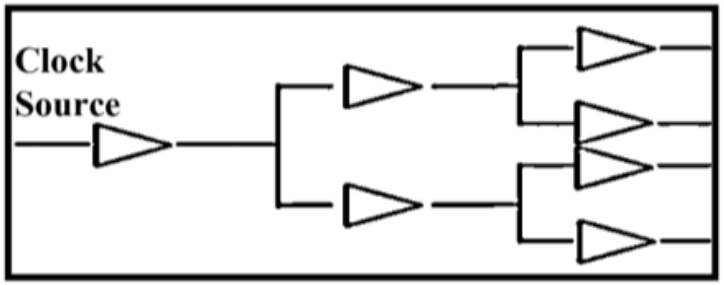


CTS Algorithms

- H Tree based algorithm
- Pi Configuration



H-Tree

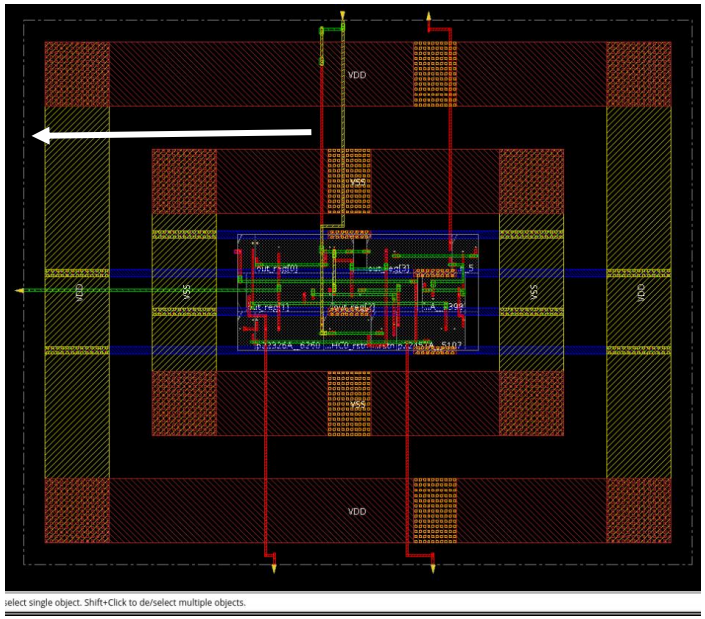


Pi Configuration

Routing

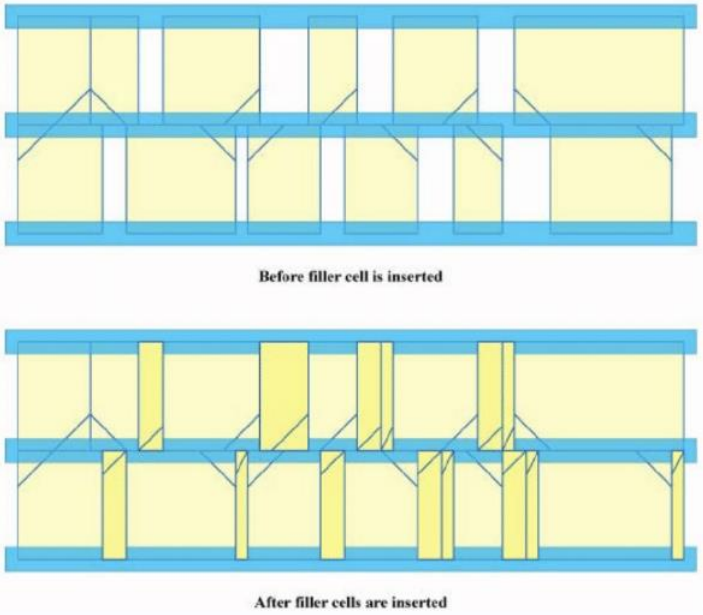
- Routing Objectives
  - Skew requirements
  - Open/Short circuit cleaning
  - Routed paths must meet setup and hold timing req
  - Metal traces must meet DRC constraints
  - Layout geometries must meet current density requirements

Routing wires

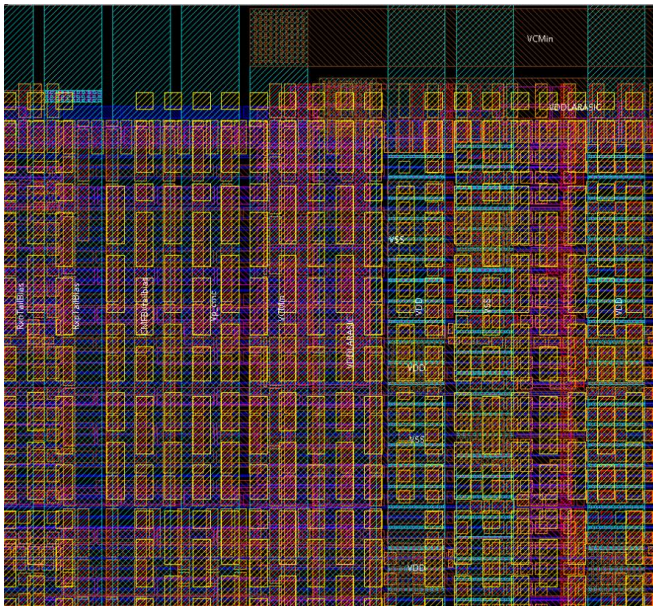


Signoff

Filler/Decap cells

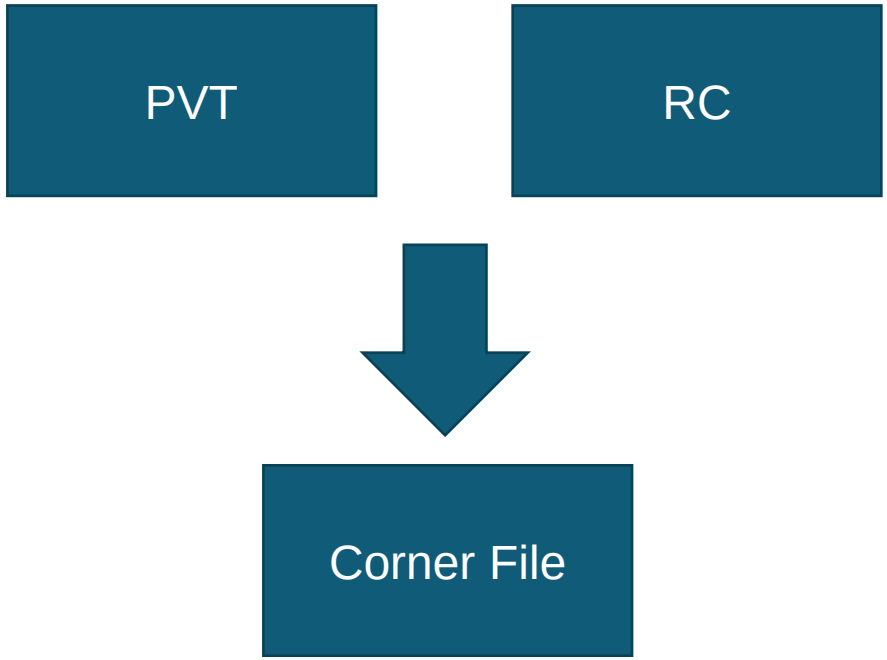


Metal Density



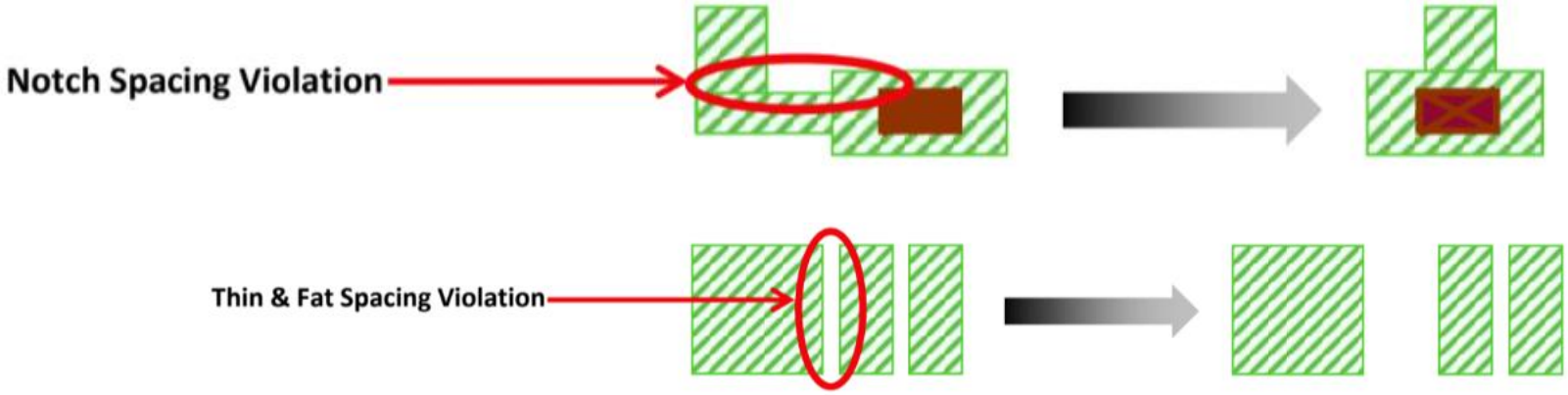
Corners & Analysis View

- ❑ Process (SS, TT, FF)
- ❑ Voltage (1.08, 1.2, 1.32)
- ❑ Temperature (-40, 25, 125)
- ❑ RC corners

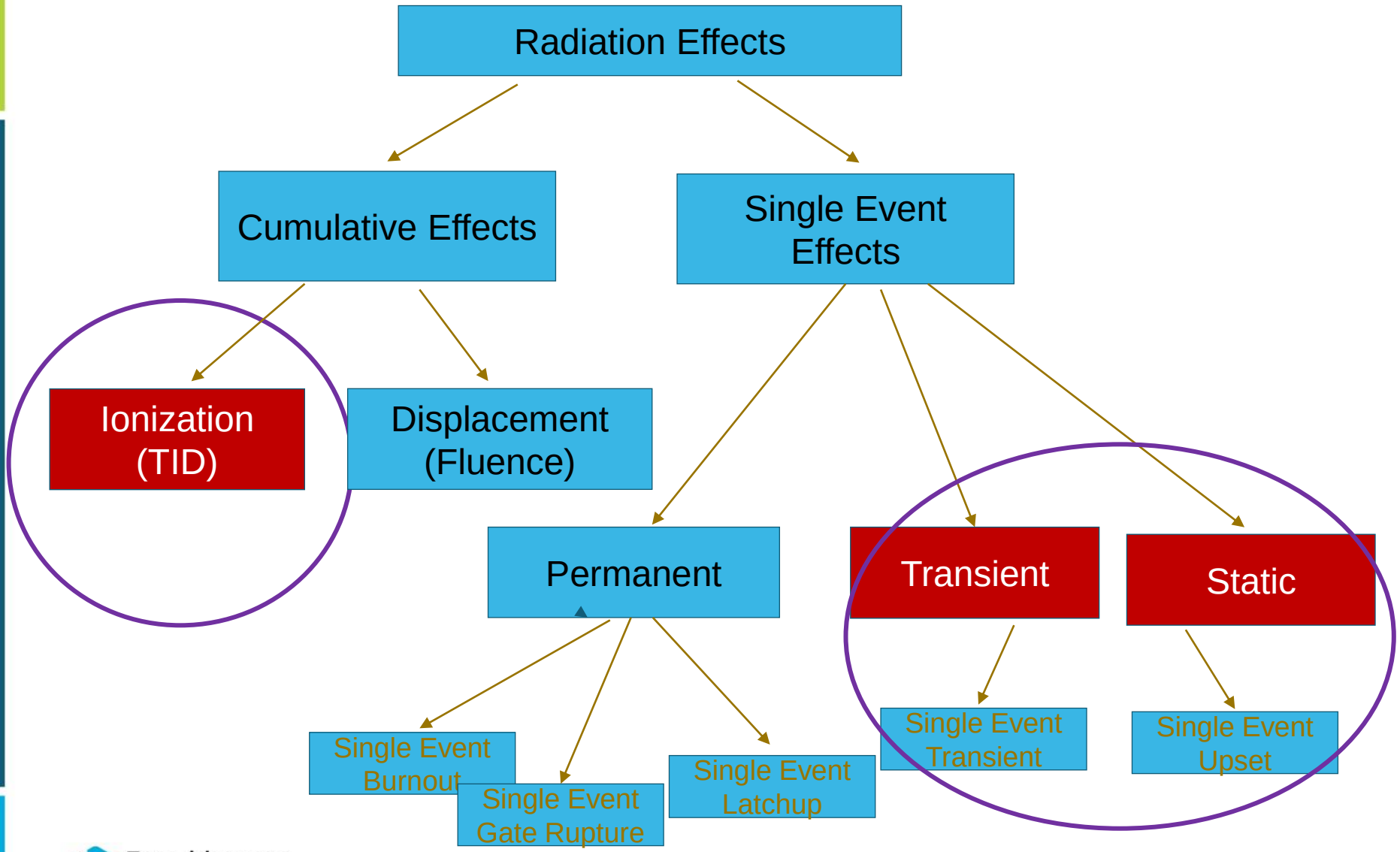


Physical Verification

- DRC : Process of checking physical layout data against foundry-specific rules to ensure successful fabrication
- LVS : Layout vs Schematic, e.g. check shorts and opens
- Antenna checks
- Wire/Bump bonding rules

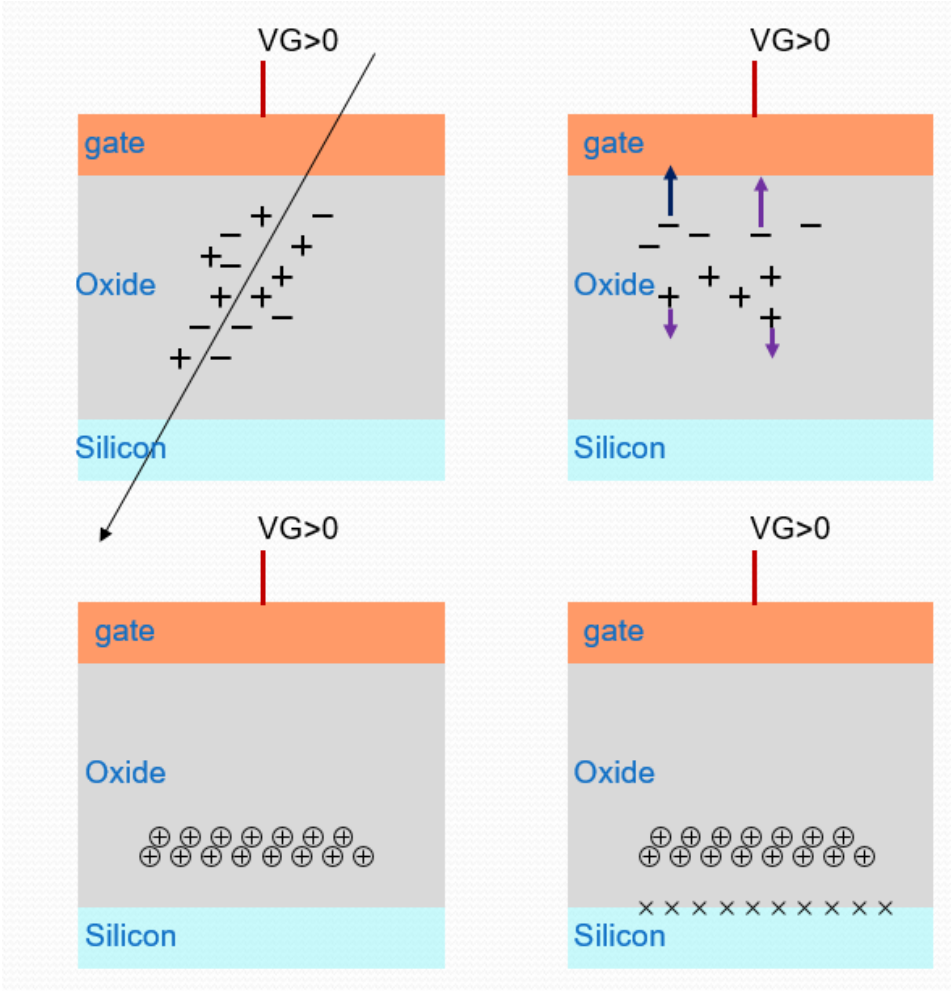


Radiation Effects on Semi-Conductor Devices

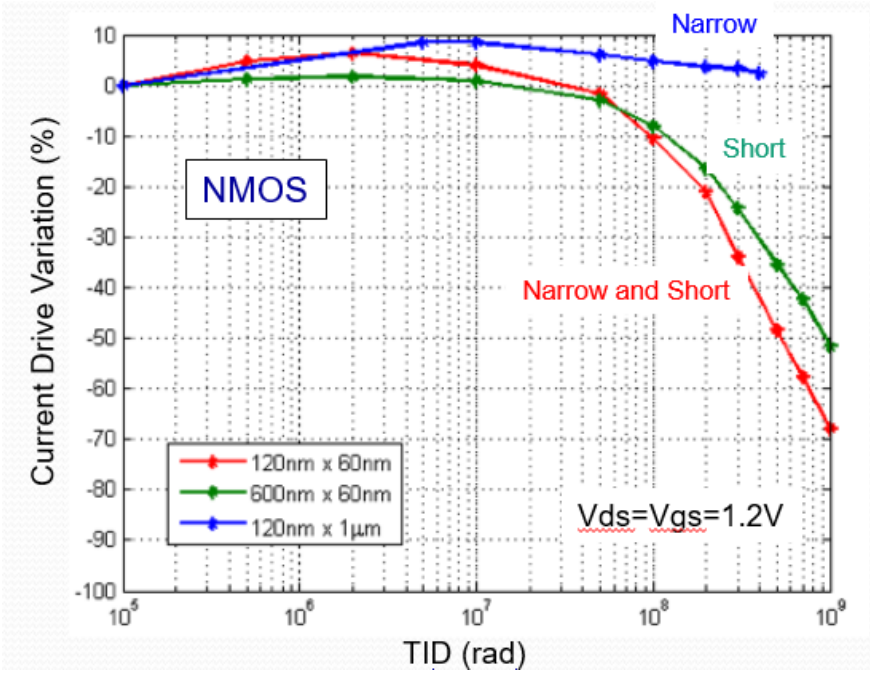


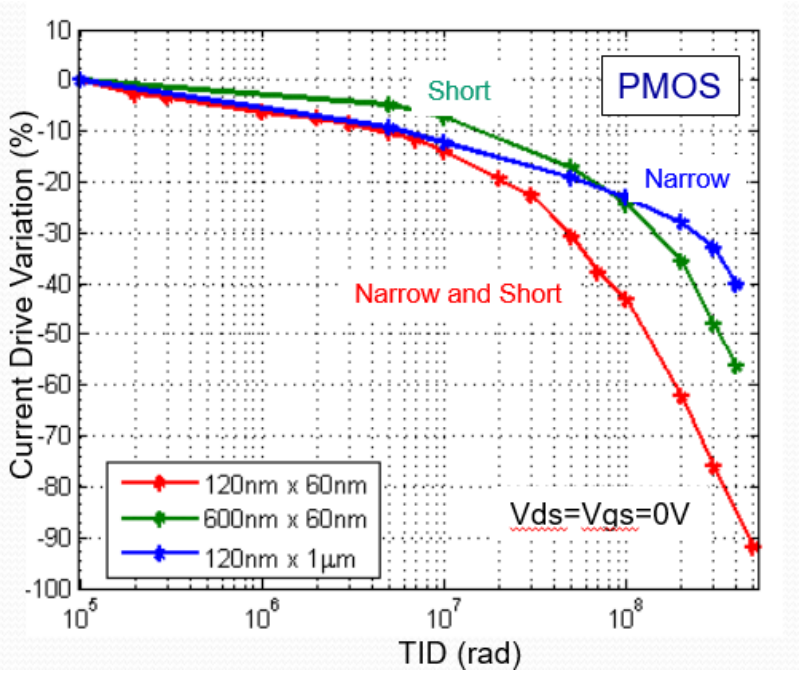
Total Ionizing Dose (TID)

- Ionizing radiation builds up interface trap states
- TID-induced charge in the oxide decreases with the thickness
- Thick oxide used for device isolation introduces leakage
- Parameters:
  - Temperature
  - Bias
  - Annealing
  - Dose Rate



Device Irradiation (65nm)





Narrow Transistors

- NMOS less affected by TID

F.Faccio et al., TWEPP 2015, TID effects in 65nm transistors: summary of long irradiation study at the CERN X-rays facility

Short Transistors

- PMOS and NMOS devices are degraded

Narrow and Short devices

- PMOS completely off for 1 Grad

Brookhaven National Laboratory

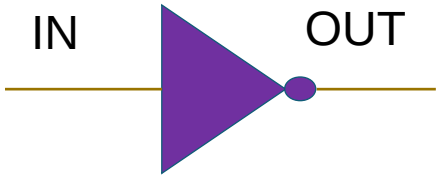
TID effects on Digital Circuits

- Analog Designs
  - Avoid excessively narrow and short transistors
  - Keep the same bias for branches of the differential structures
- For Analog Designs (upto 500 Mrad)
  - NMOS: L >= 120nm, any W ok
  - PMOS: L >= 120nm, W>= 300nm
- Digital Designs (Semi-Custom Design Flow)
  - Standard cells often use short and narrow devices
  - Speed degradation due to increase of Vth
  - Development of new radiation models
  - Characterize new timing libraries and included as additional corners
    - Using Liberate Tool

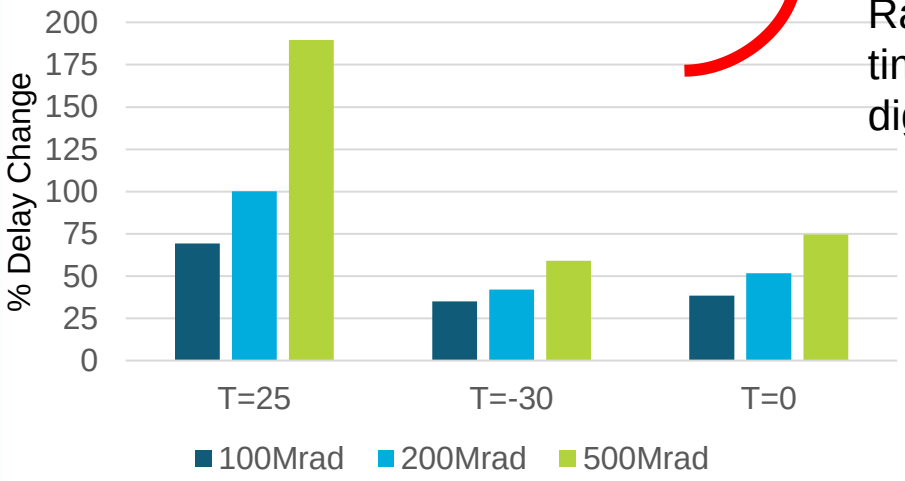
Recommendations from Radiation Working Group in RD53

Total Ionizing Dose (TID) Tolerant ASICs

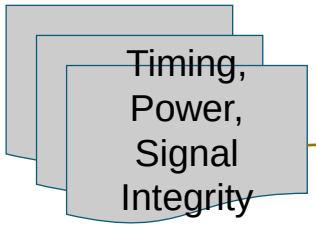
- TID alters the performance of CMOS device
  - Degrades logic gate performance



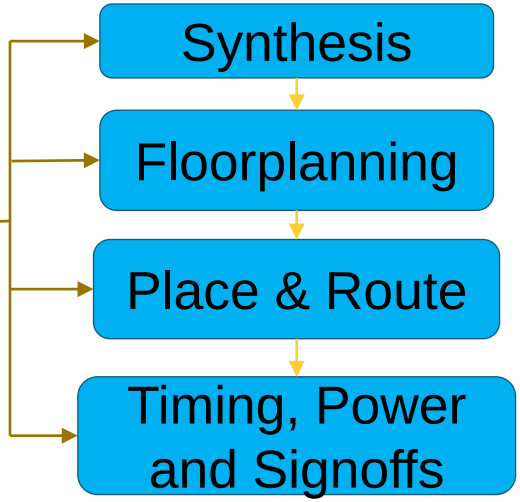
TID Effects on an Inverter (Typical Corner)



Liberty Timing Models

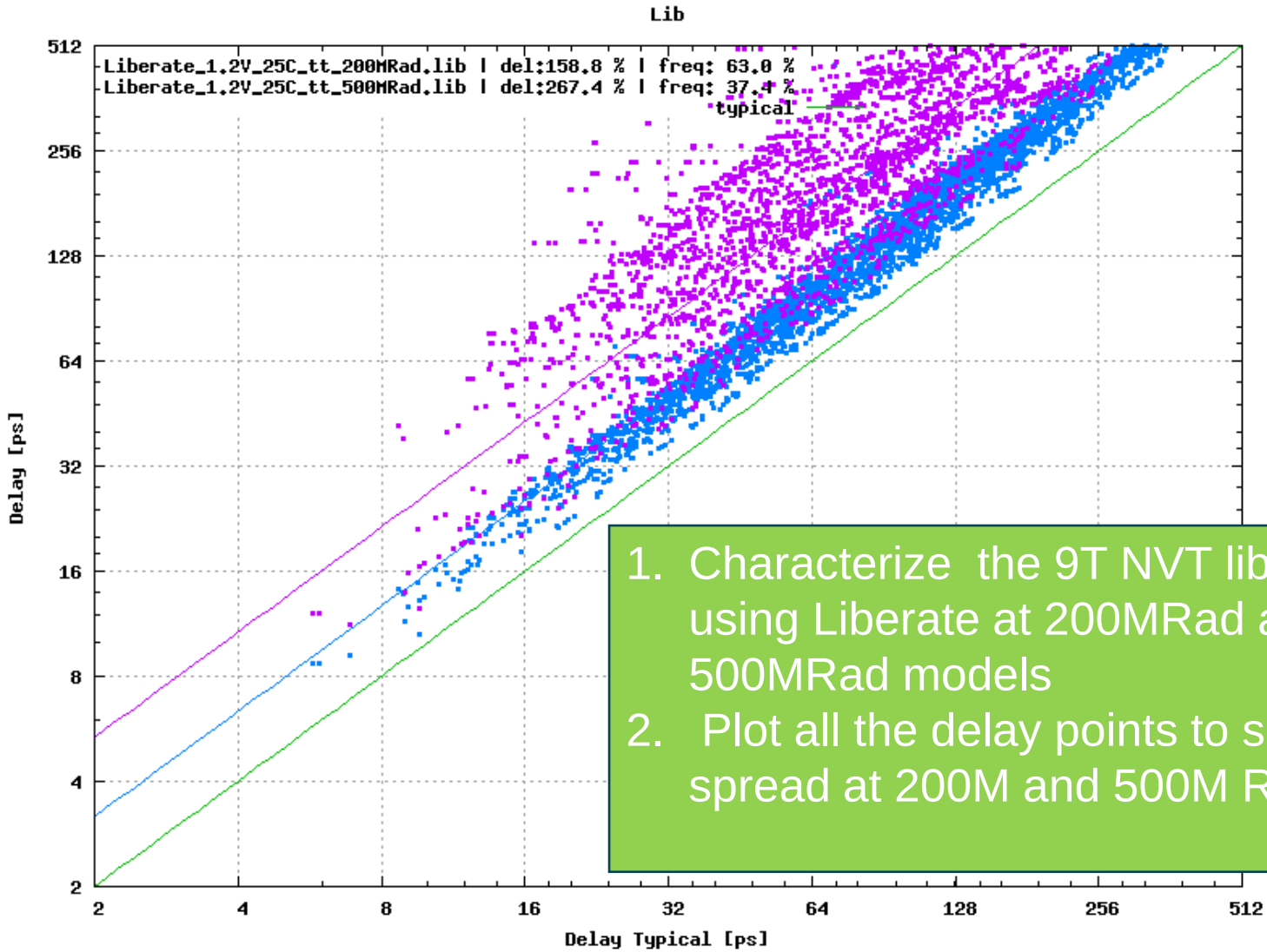


Radiation corner timing libraries in digital design flow



- Timing libraries are characterized using industry standard tools using rad corner device models
- Important for ASICs operating in extreme environments (e.g. DUNE)

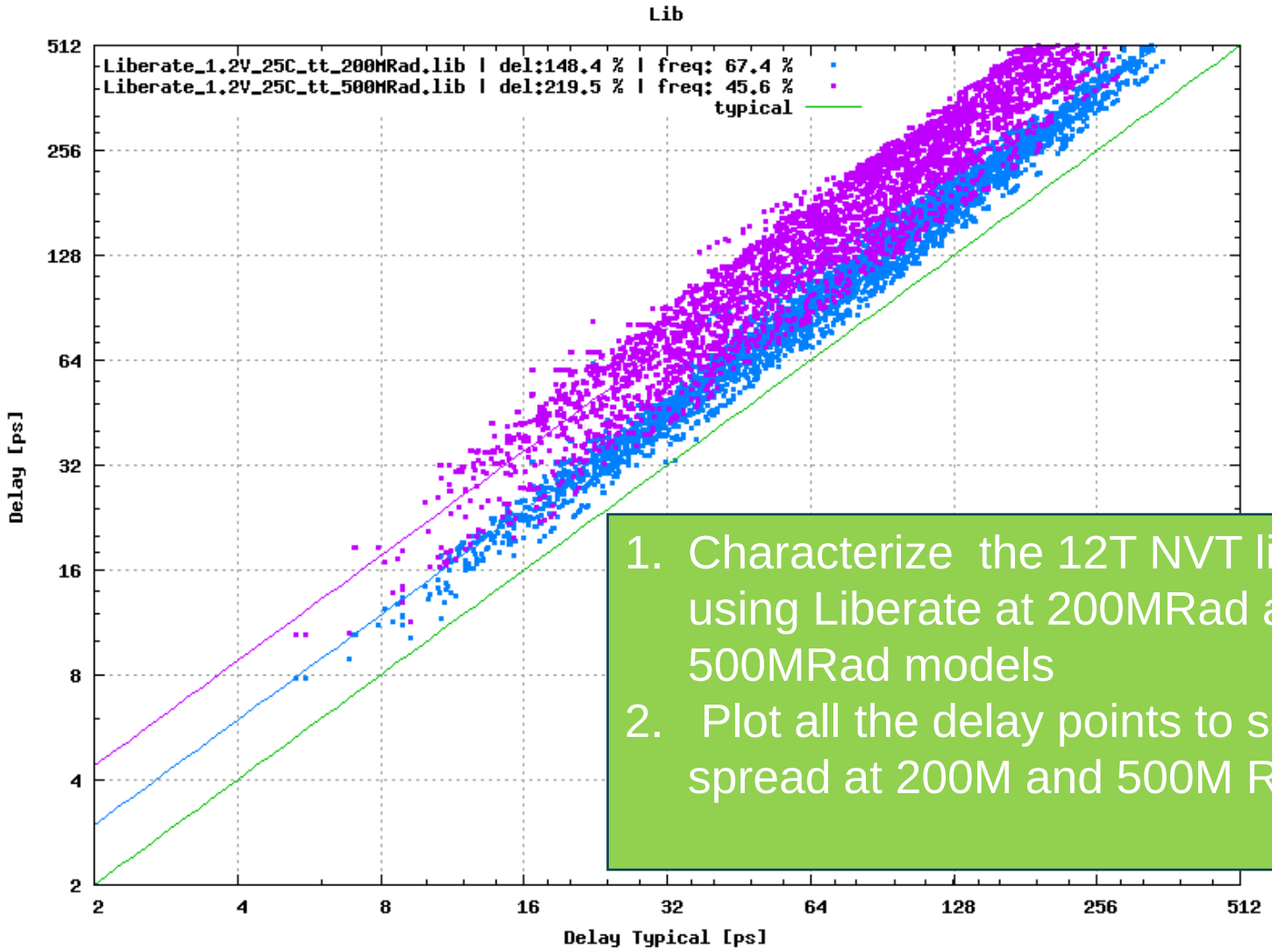
Radiation Effects on 9T NVT Library



1. Characterize the 9T NVT library using Liberate at 200MRad and 500MRad models
2. Plot all the delay points to see the spread at 200M and 500M Rad

Brookhaven National Laboratory

Radiation Effects on 12T NVT Library



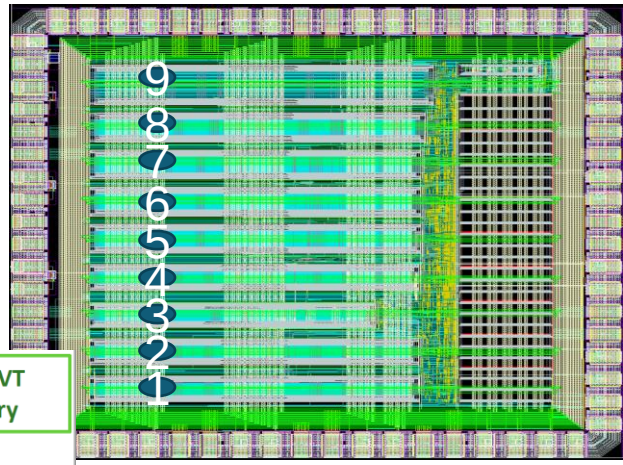
1. Characterize the 12T NVT library using Liberate at 200MRad and 500MRad models
2. Plot all the delay points to see the spread at 200M and 500M Rad

Brookhaven National Laboratory

Digital Radiation Test Chip (DRAD)

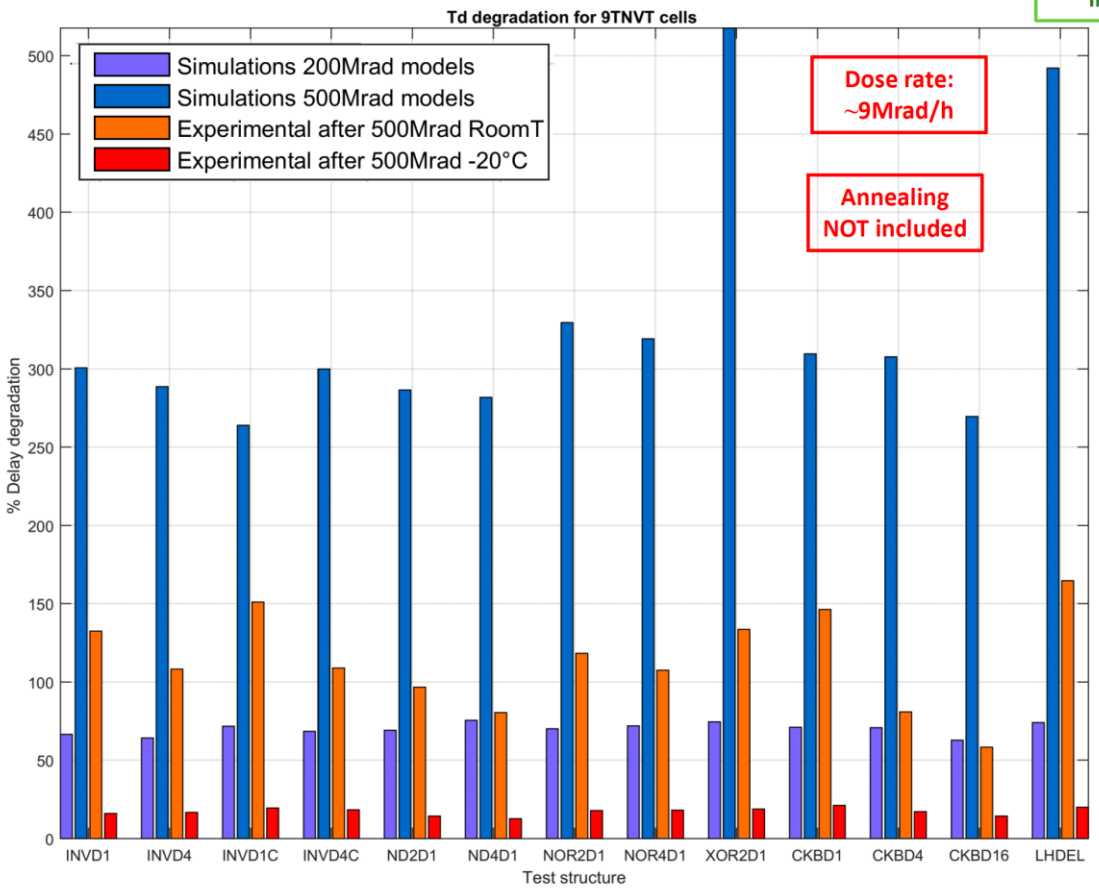
- 9 different standard cell libraries
- Investigated subset of cells of in each library (INV, NAND, NOR, XOR, DFF, Latch)

DRAD (03/2016)



9T_NVT library

Experimental and simulations results:

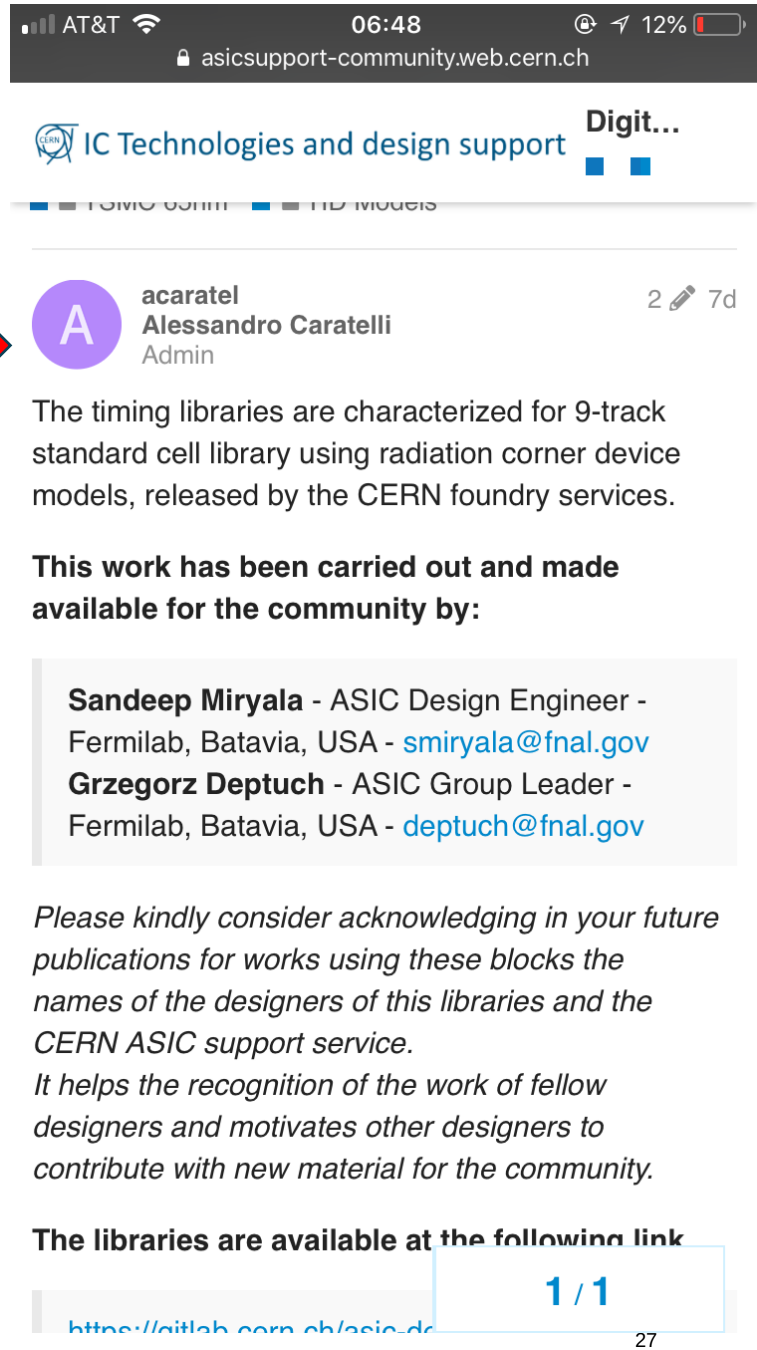


Simulation models were pessimistic

LMJ cases et.al., "Characterization of Radiation Effects in 65nm Digital Circuits using DRAD test chip", JINST, 2017

Synergetic Applications

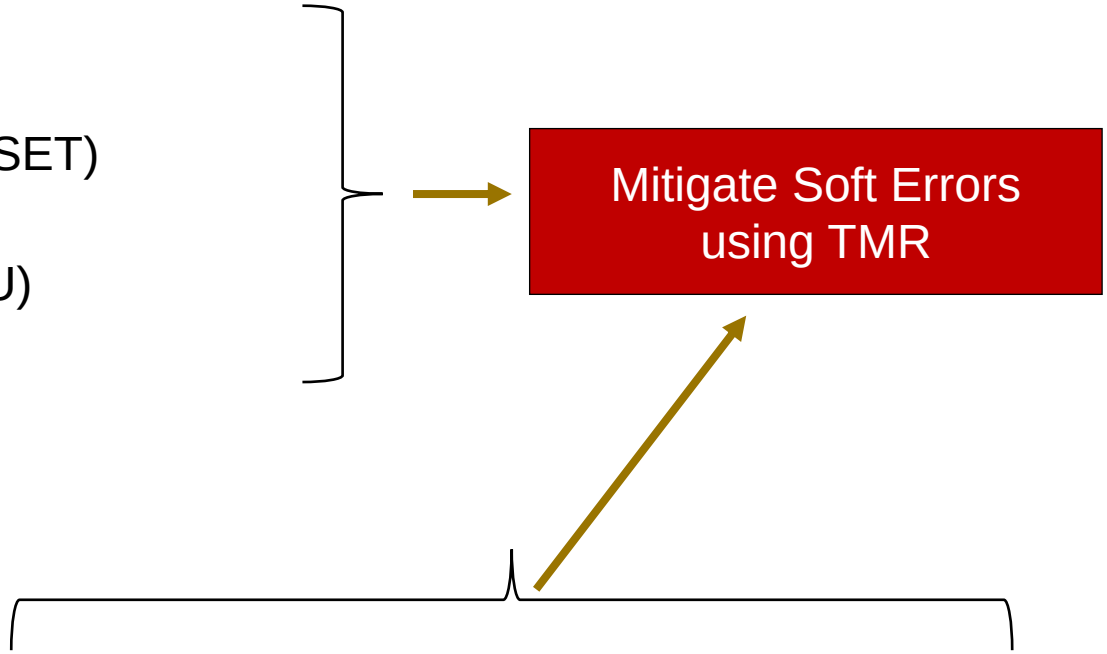
- Characterized timing libraries are shared with the community
  - New models are available from CERN foundry services
  - 100Mrd, 200Mrad and 500Mrad
  - Limited set to corners
- RD53 Chips
  - Same methodology adopted for RD53A and RD53B chips
  - Chips are operating as expected for TID of 500Mrad
- ASICs for DUNE Liquid Argon Time Projection Chamber (TPC)
  - Digital logic implementations in ColData and ColdADC chips
  - Must operate at cryogenic temperature 87K or -189ºC
  - ~225 custom standard cells (65nm → 90nm)
  - Timing libraries are characterized at all corners (regular as well as cryogenic temperature)
  - Never saw a failure in digital logic
- Cryo-CMOS ASICs for quantum computing applications
  - Must be designed for 4K



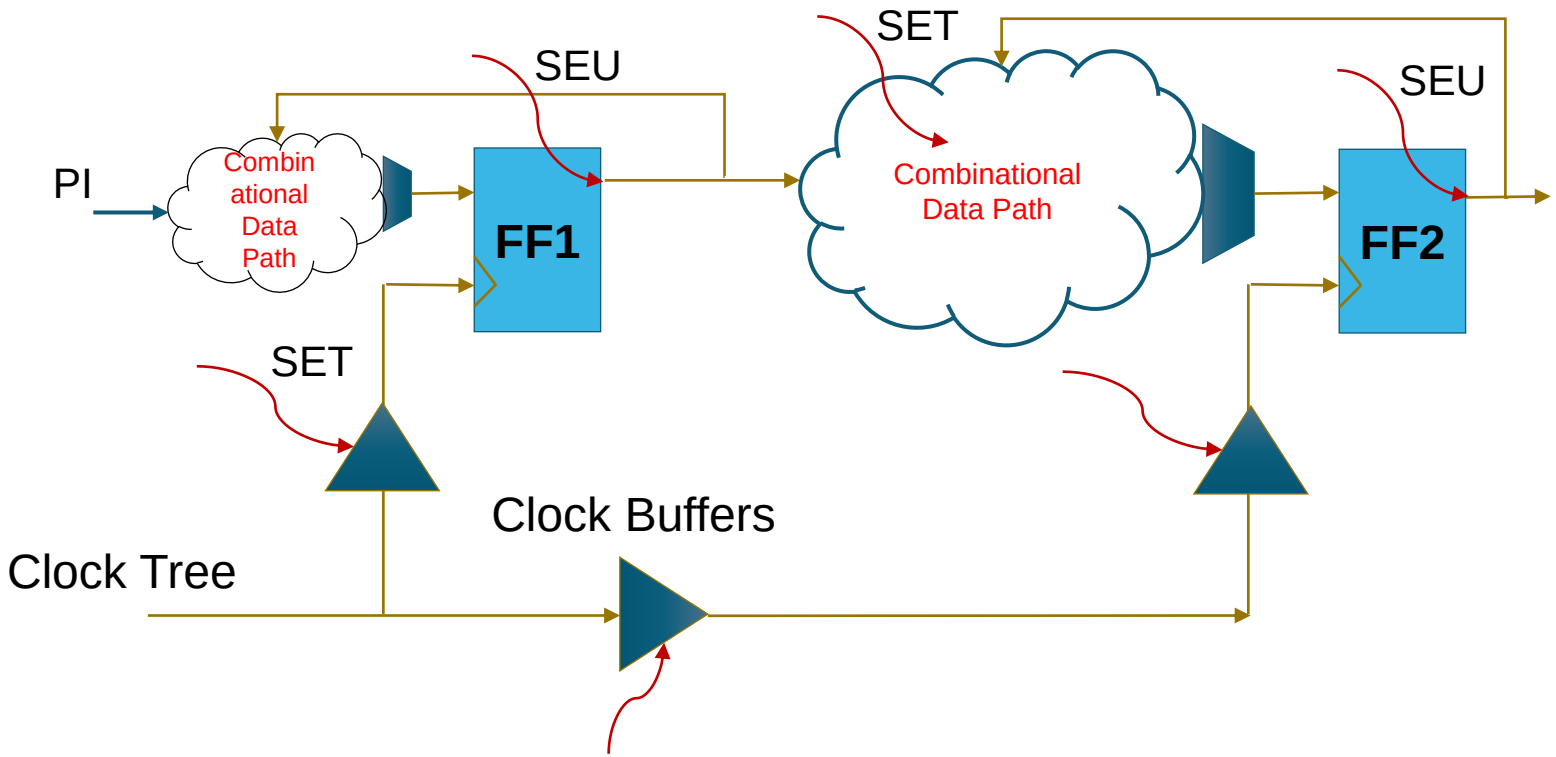
Brookhaven National Laboratory

Single Event Effects (SEE) Classification

- Soft Errors
  - Transient
    - Single Event Transient (SET)
  - Static
    - Single Event Upset (SEU)
- Soft Error mitigation
  - Technology level
    - Bulk CMOS Vs SOI
  - Cell level
    - Dice latch
  - System level redundancy : Triple Modular Redundancy (TMR)



SEE Issues in Digital Logic Overview



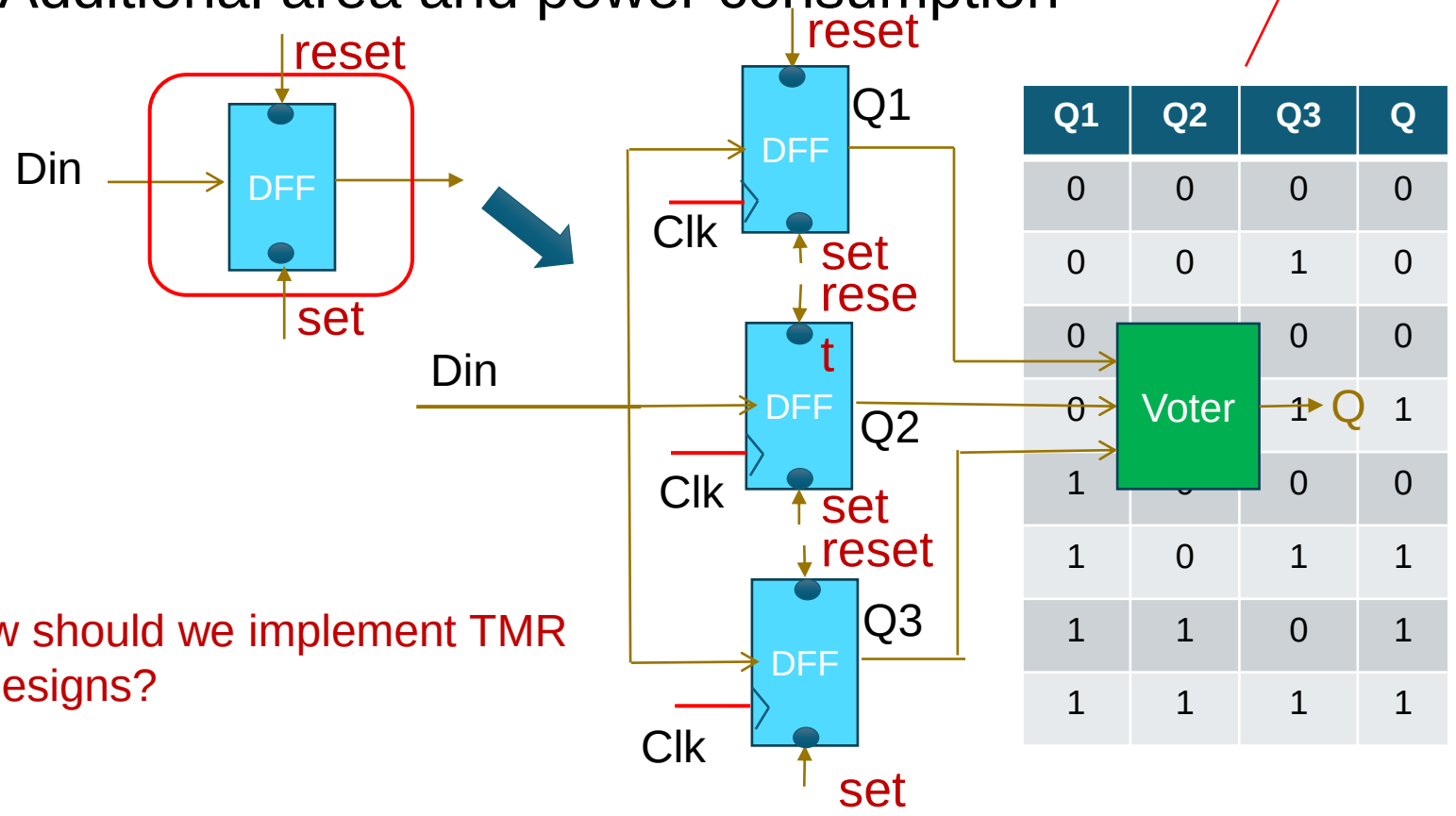
SEE possibilities in digital logic

- SEU's in registers
- SET's in clock buffers
- SET's in combinational data path

Triple Modular Redundancy (TMR)

- Data is replicated on multiple nodes
- Mitigation of SEUs in sequential elements
- Additional area and power consumption

Q = (Q1&Q2)| (Q2&Q3)| (Q3&Q1)

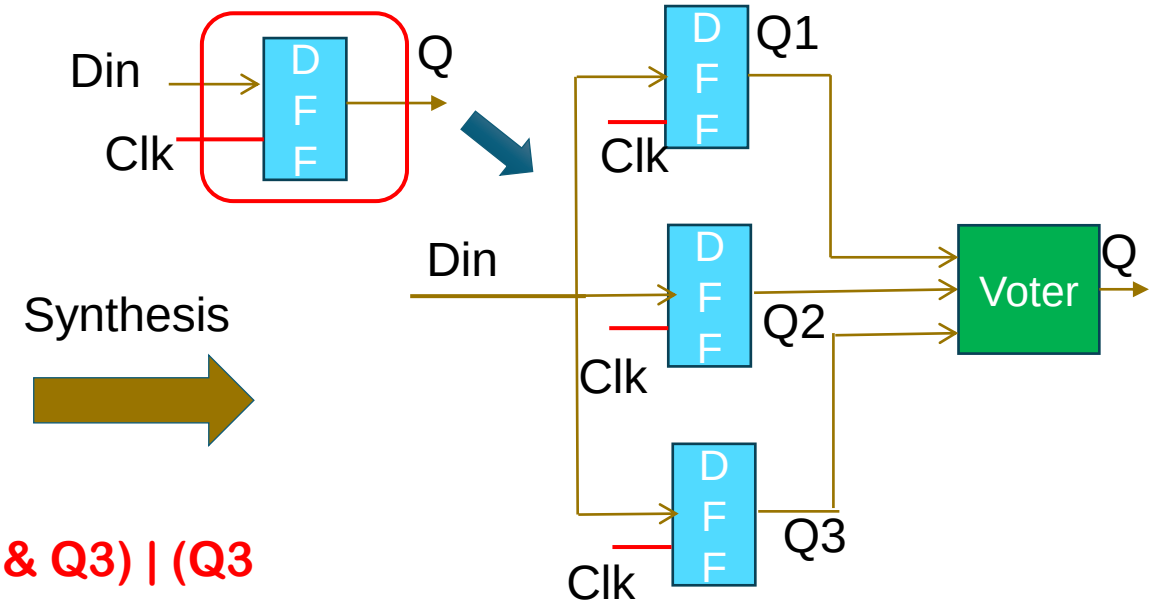


Q1	Q2	Q3	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

How should we implement TMR in designs?

TMR Insertion by RTL designers

```
module TMR (Din, Q, clk);
Input Din;
output Q;
reg Q1, Q2, Q3;
always @(posedge clk)
begin
        Q <= Din;
        Q1 <= Din;
        Q2 <= Din;
        Q3 <= Din;
end
assign Q = (Q1 & Q2) | (Q2 & Q3) | (Q3
& Q1)
endmodule
```

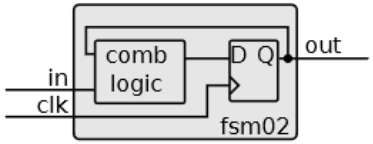


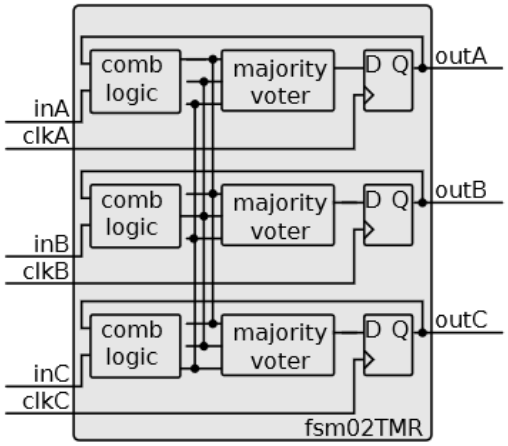
Disadvantages:

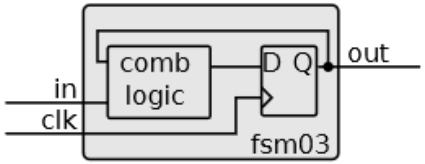
1. Redundant logic is removed by synthesis tools
2. Additional don't touch commands
3. Complex for chips with huge # of registers

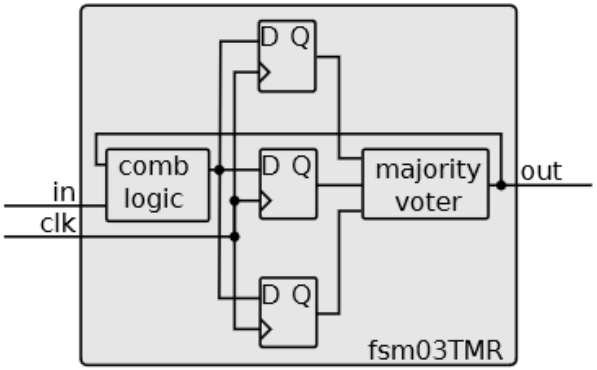
Triple Modular Redundancy Generator (TMRG)

- In-house tool developed by CERN
- HDL language dependent (only Verilog)
- Successfully used in LPGBT, MPA and SSA ASICs





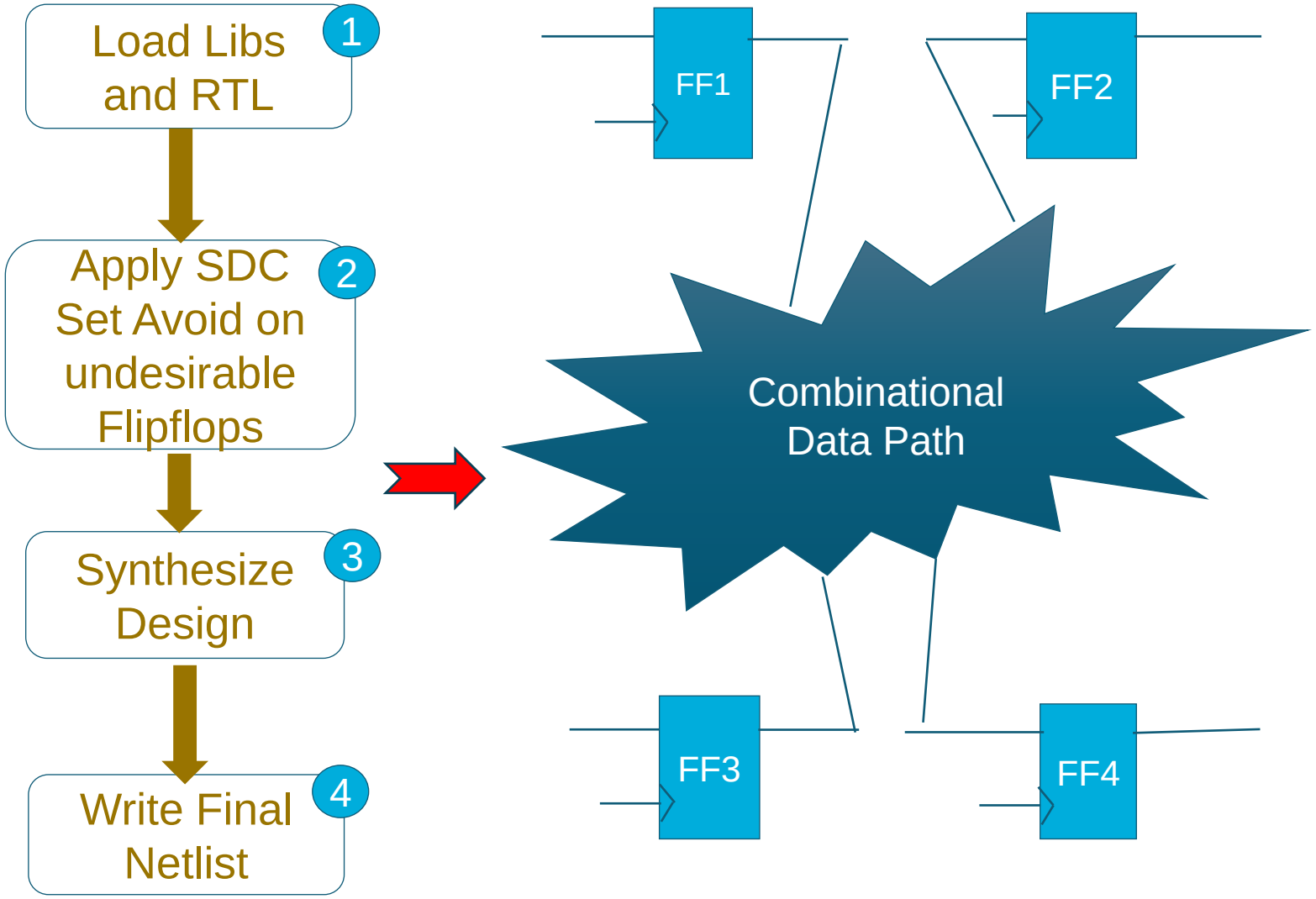




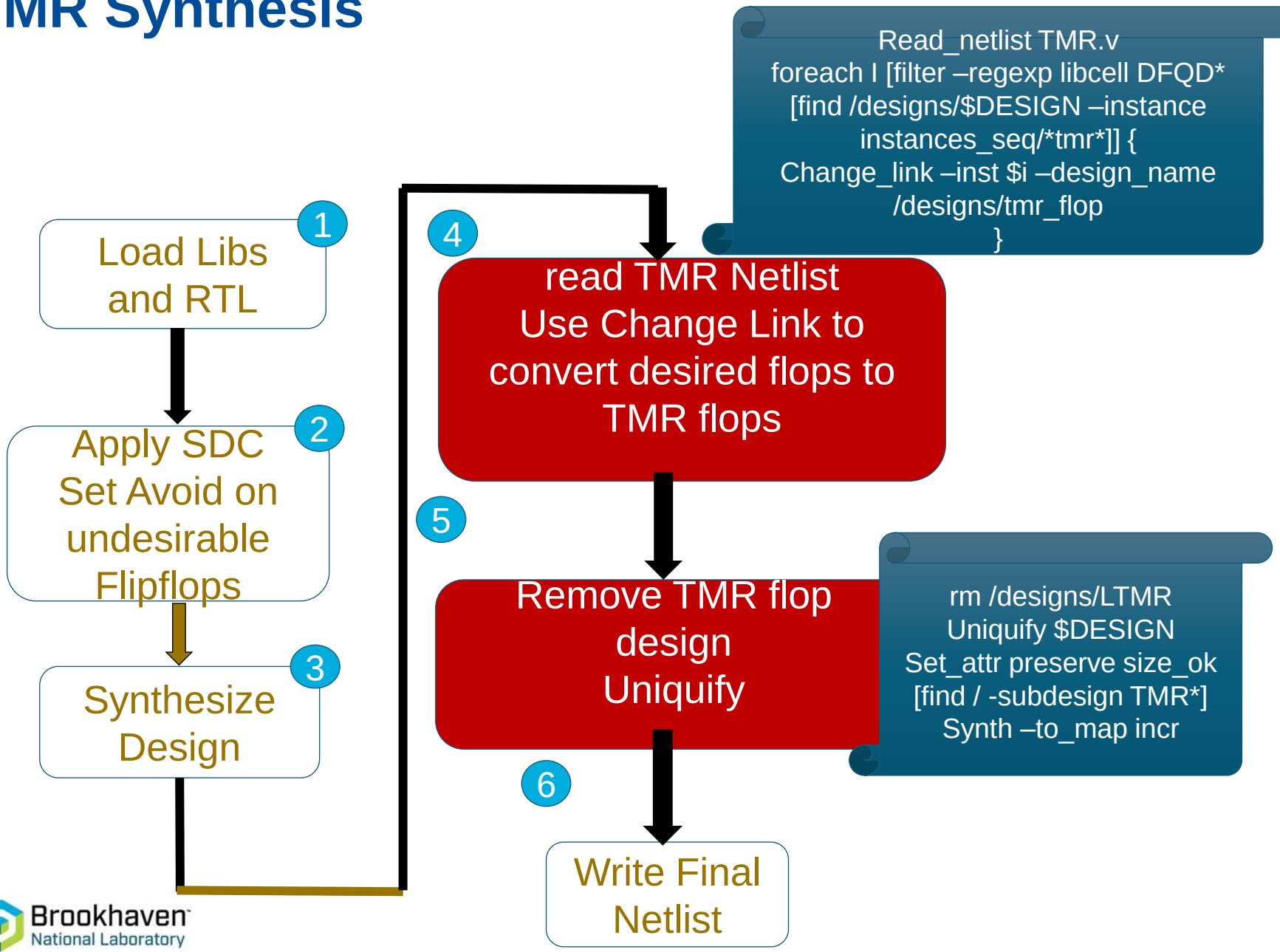
https://tmrg.web.cern.ch/tmrg/

Brookhaven National Laboratory

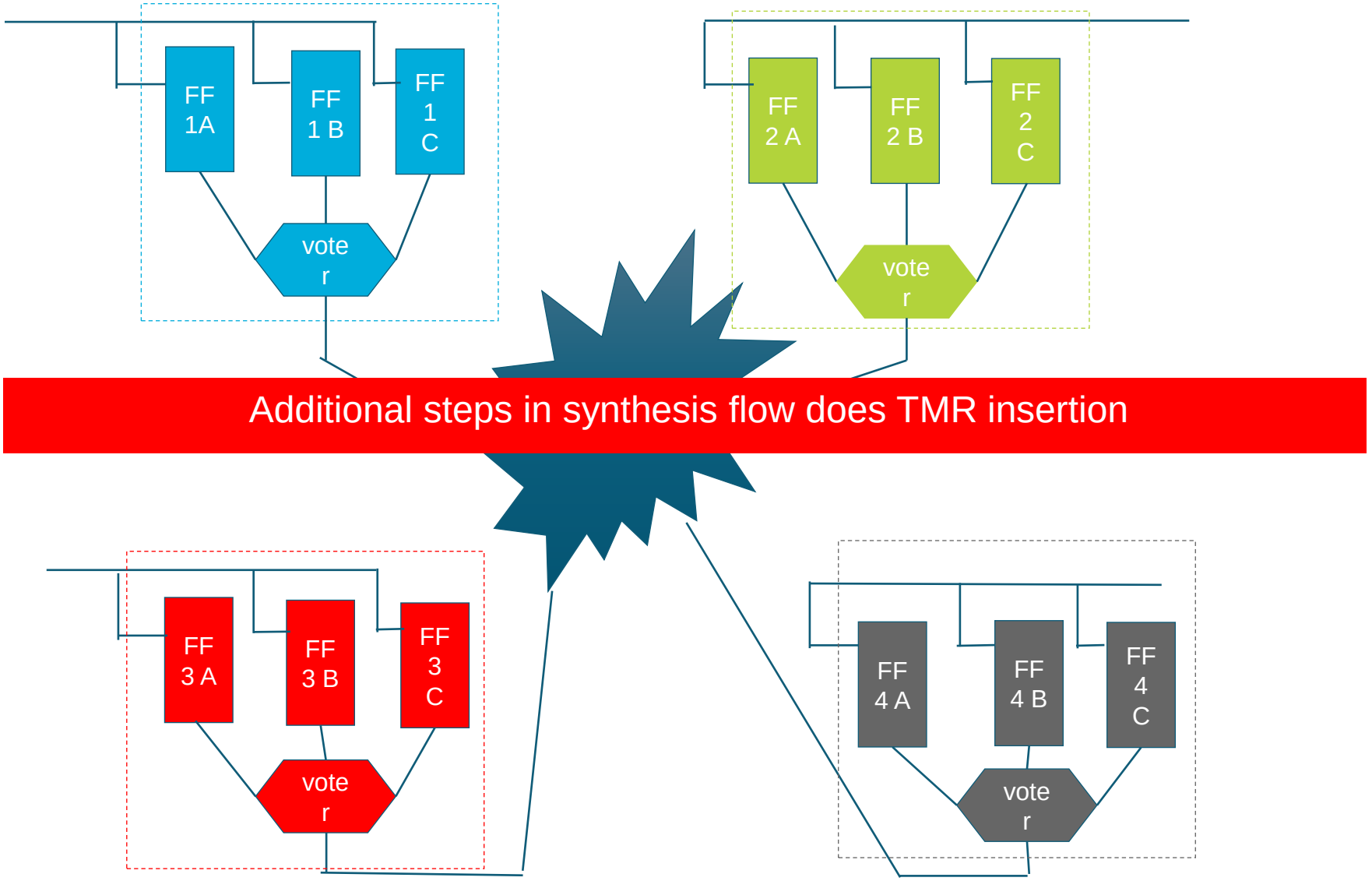
Logic Synthesis Flow : Generic



TMR Synthesis



TMR Insertion Automation



4

```
read_netlist TMR_DFQD.v
foreach I [filter –regexp libcell DFQD* [find /designs/$DESIGN –instance
instances_seq/*reg*]] {
Change_link –inst $i –design_name /designs/TMR_DFQD
}
```

TMR_DFQD.v

```
module TMR_DFQD (CP, D, Q);
input CP, D;
output Q;
wire CP, D1, D2, D3, Q, Q1, Q2, Q3, n_0;
```

Each flipflop type must have its corresponding TMR netlist

```
        DFQD1 DFFQ3_reg(.CP (CP), .D (D), .Q (Q3));
        MAOI222D0 p4324D(.A (Q1), .B (Q2), .C (Q3), .ZN (n_0));
        CKND0 Fp4461A(.I (n_0), .ZN (Q));
endmodule
```

TMR Insertion Scenarios

- Triplicating only the registers having *tmr* in RTL

```
module TOPModule ();
        …
        reg in_tmr;
        …
endmodule
```

```
read_netlist TMR_DFQD.v
foreach I [filter –regexp libcell DFQD* [find
/designs/$DESIGN/* –instance instances_seq/*tmr*]] {
Change_link –inst $i –design_name
/designs/TMR_DFQD
}
```

- Triplicating all the registers in the RTL

```
module TOPModule ();
        …
        reg in;
        …
endmodule
```

```
read_netlist TMR_DFQD.v
foreach I [filter –regexp libcell DFQD* [find
/designs/$DESIGN/* –instance instances_seq/*reg*]] {
Change_link –inst $i –design_name
/designs/TMR_DFQD
}
```

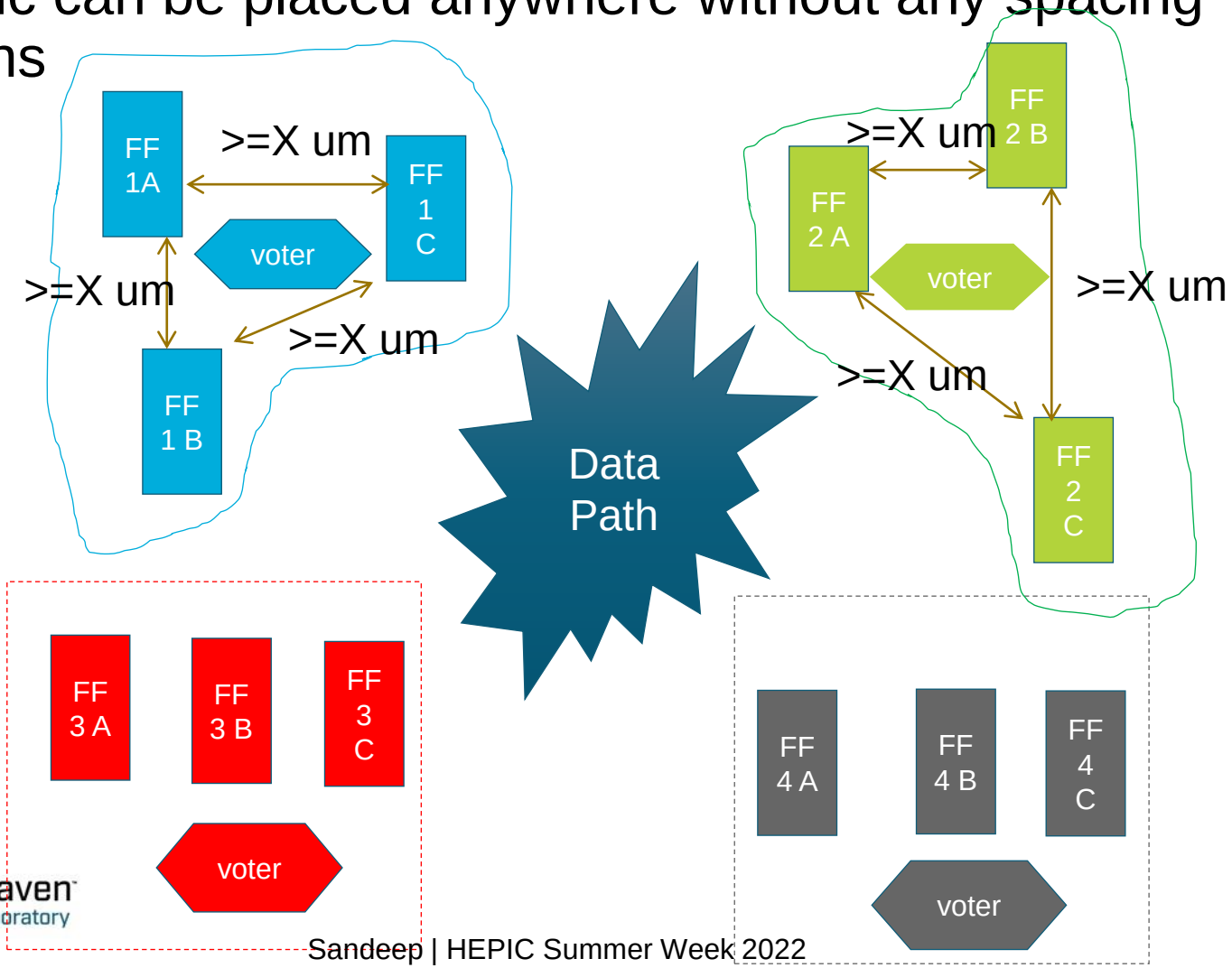
- Triplicating in one of the hierarchical module in RTL

```
module submodule();
        reg test;
        ….
endmodule
module TOPModule ();
        …
         submodule hierInst ();
endmodule
```

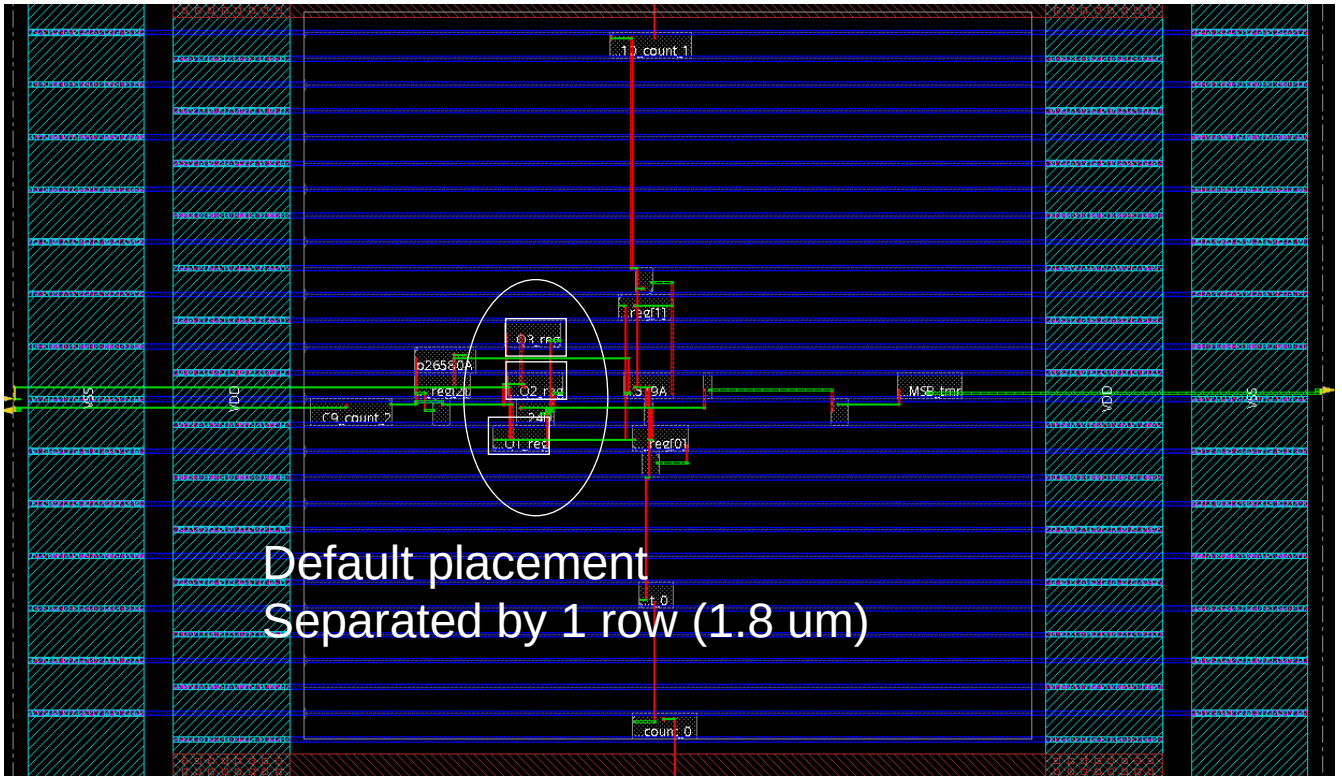
```
read_netlist TMR_DFQD.v
foreach I [filter –regexp libcell DFQD* [find
/designs/$DESIGN/instances_hier/hierInst –instance
instances_seq/*reg*]] {
Change_link –inst $i –design_name
/designs/TMR_DFQD
}
```

Physical Design

- Flops within the same TMR module must be at least X um apart
- Voter logic can be placed anywhere without any spacing restrictions



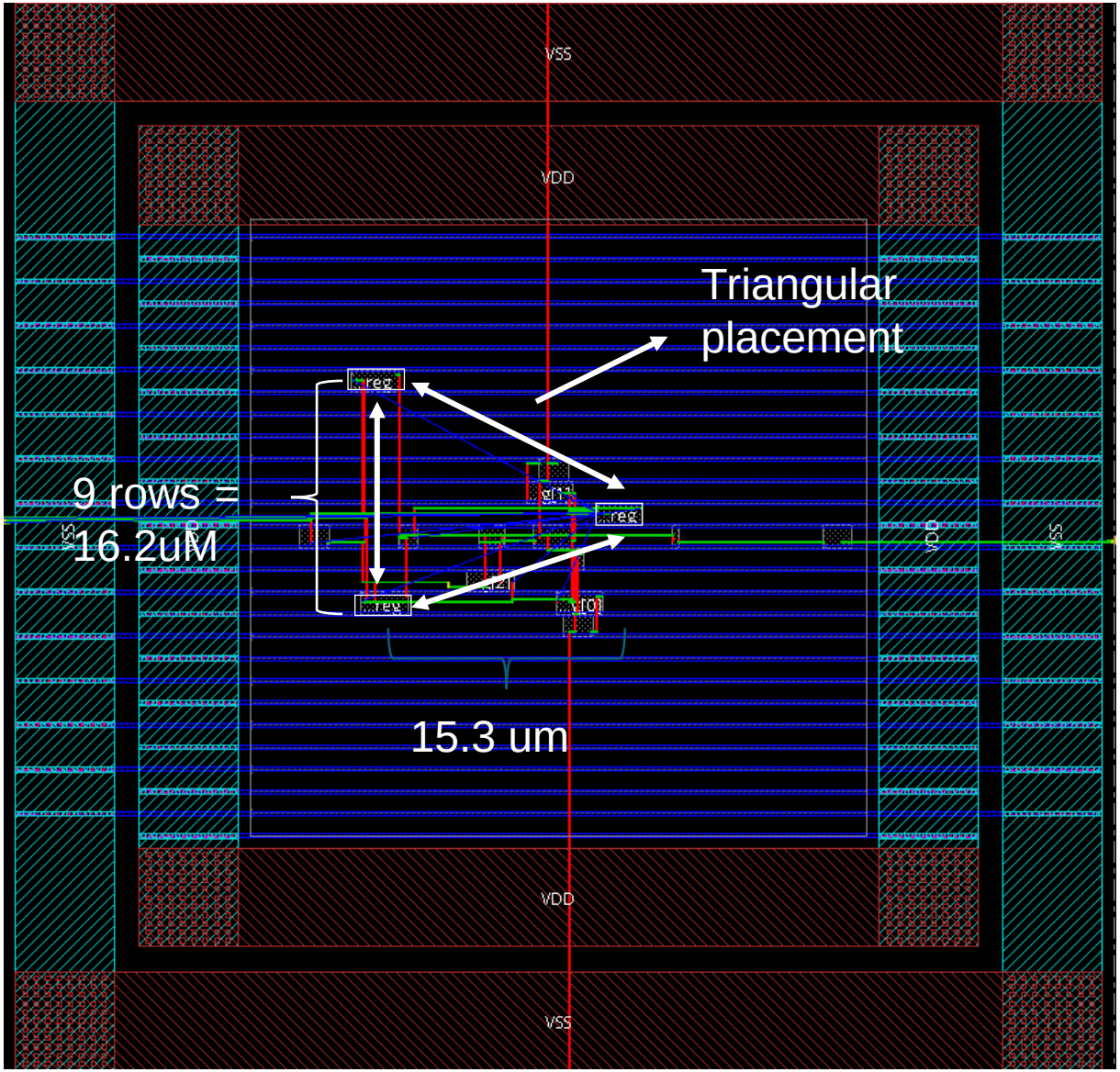
Innovus Implementation



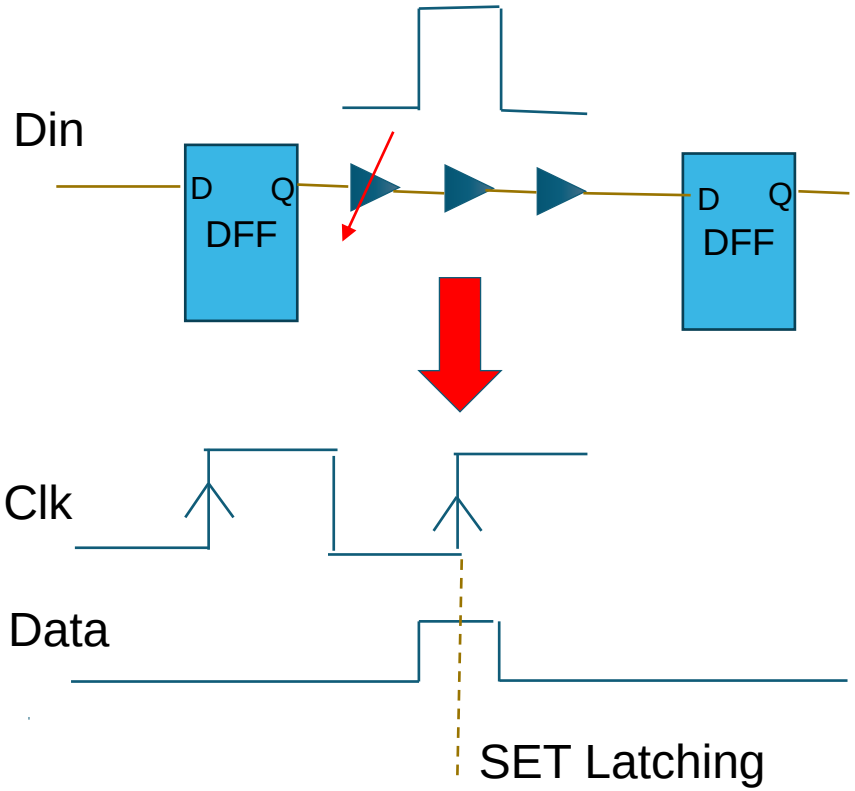
- Innovus 16.X has native space group constraints
  - Create space group constraint
  - Enable space group constraint during placement
  - Place design

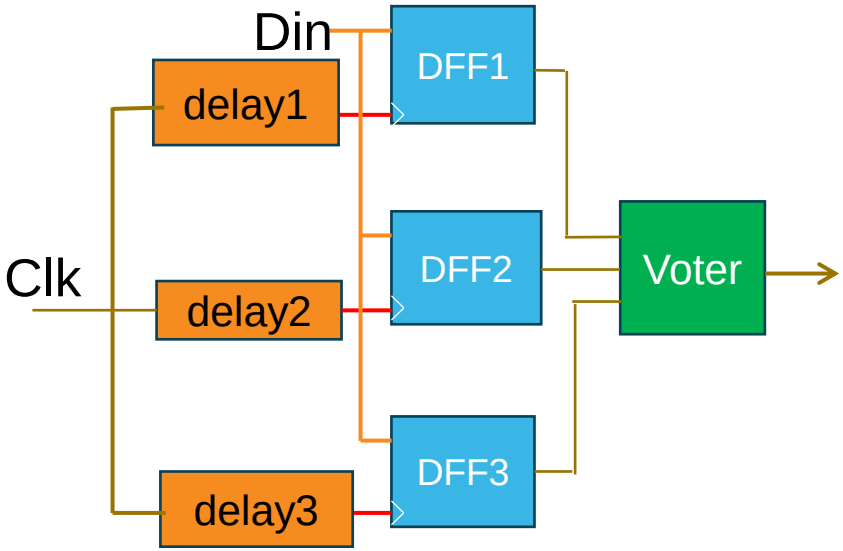
INNOVUS 16.2

- Takes horizontal and vertical spacing during place & route
- create_inst_space_group tmrSpaceGrp${n} -inst [dbget [dbget$thPtr.allInsts.cell.isSequential 1 -p2].name] –spacing_x $reg_spacing –spacing_y $reg_spacing



SET Mitigation: Combinational Data Path

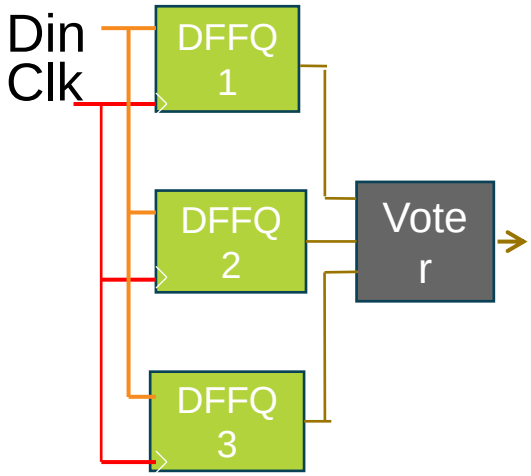




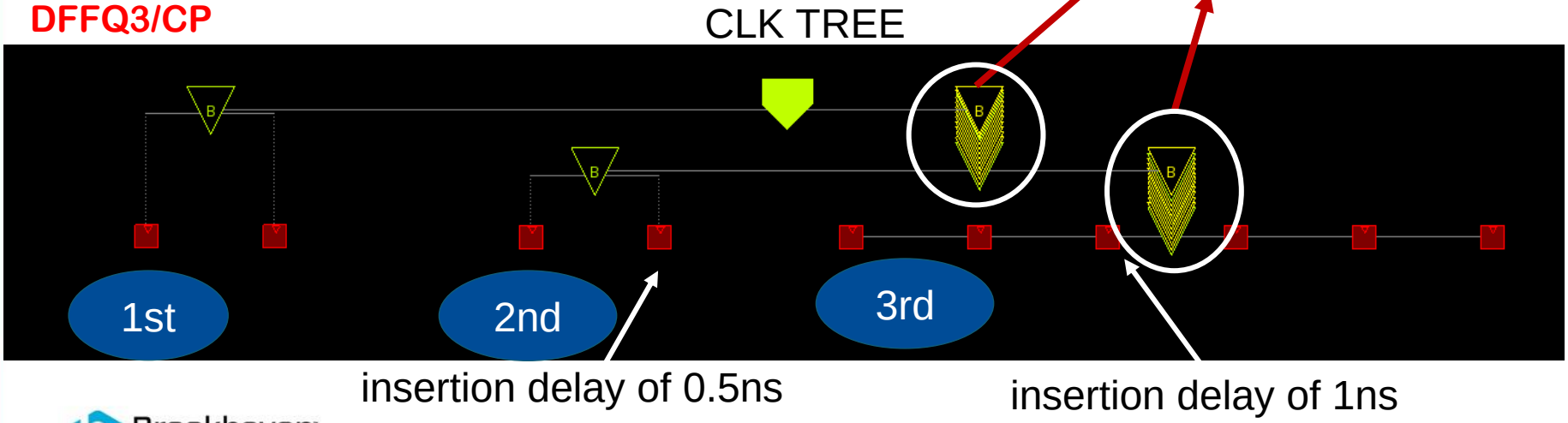
Simplest solution, Different delay values

- This delay insertion must be handled globally
- TMR cell level is expensive in area and timing

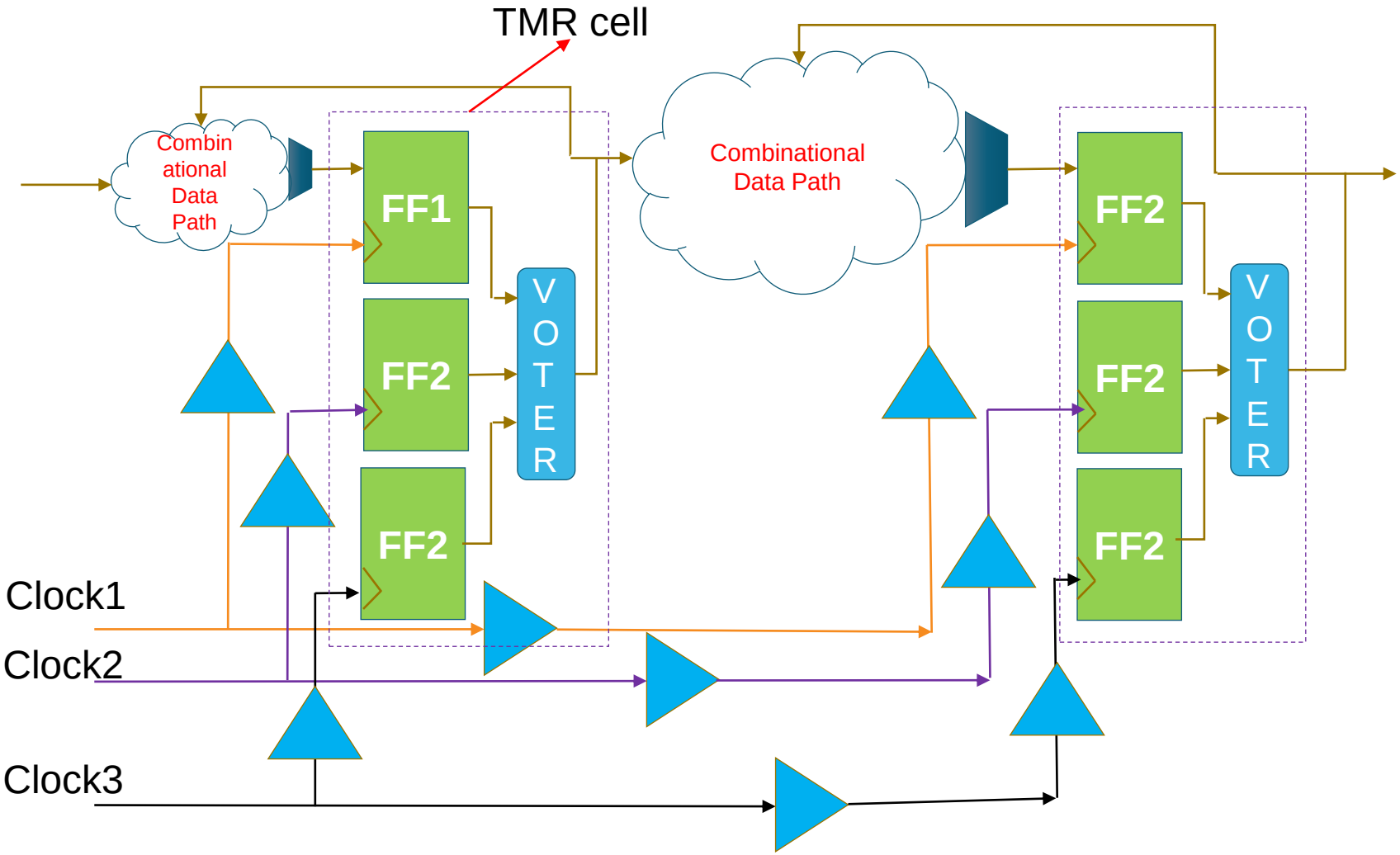
SET Mitigation : Clock Delay Insertion



set CkDel1 0.50
set CkDel2 1
 # Clock Insertion Delays on the TMR flops for SET Mitigation
set_ccopt_property insertion_delay -delay_corner {av_typ_dc} $CkDel1 -pin DFFQ2/CP
set_ccopt_property insertion_delay -delay_corner {av_typ_dc} $CkDel2 -pin DFFQ3/CP



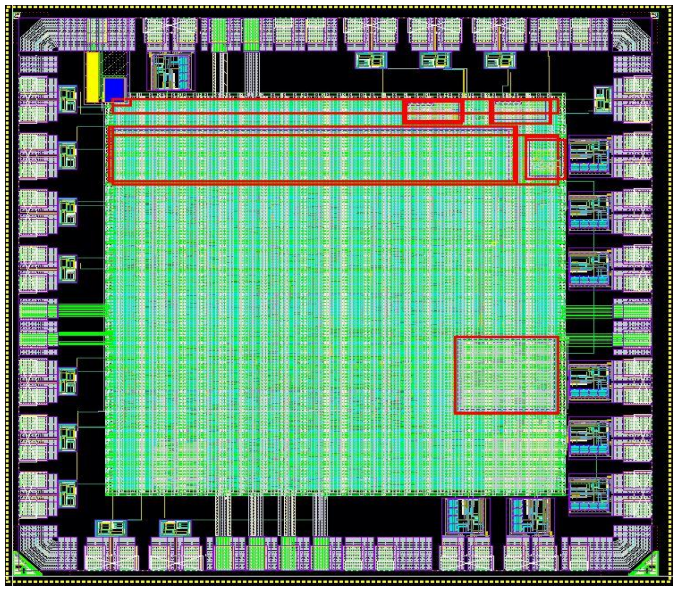
SEE Mitigation: Clock Buffers

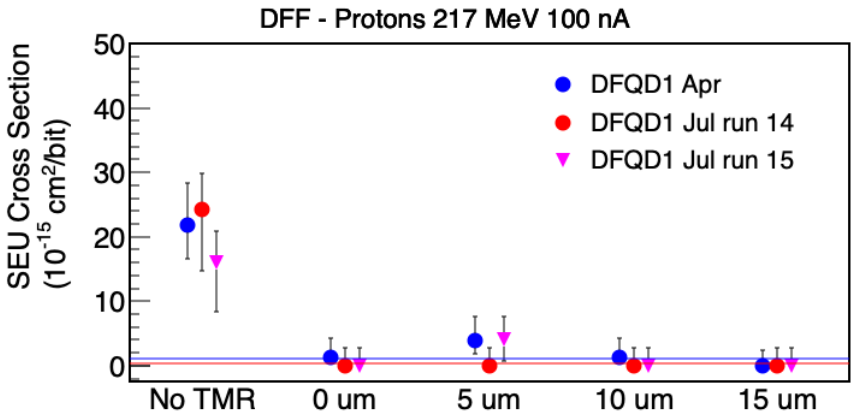


Automatically carried out while inserting TMR in Digital Flow

Measurements (Proton Beam)

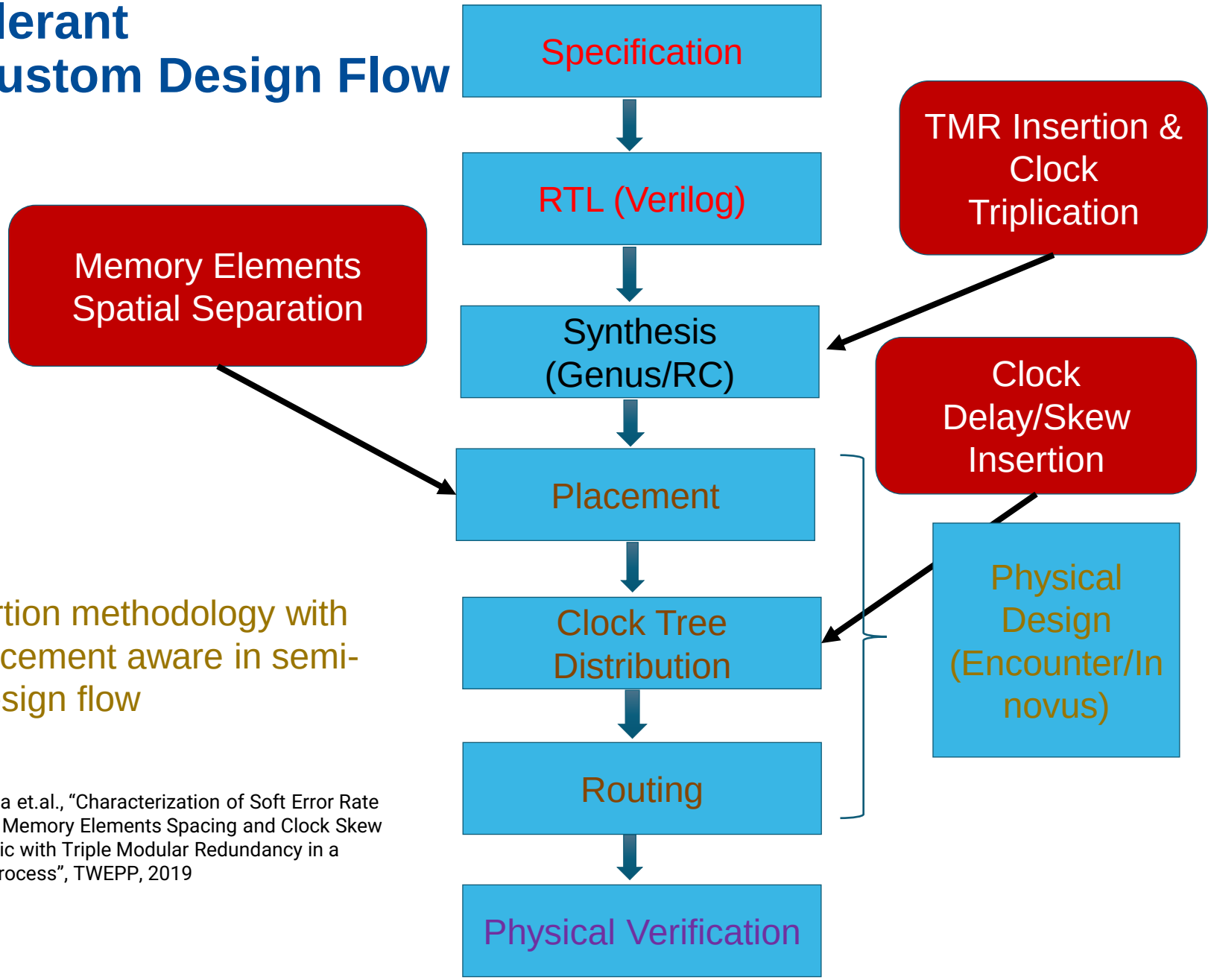
RD53SEU 08/2018





- Simple triplication, TMR improved SEU cross-section by a factor of ~10
- No improvement with memory element spacing

SEE Tolerant Semi-Custom Design Flow



TMR Insertion methodology with timing /placement aware in semi-custom design flow

S.Miryala et.al., "Characterization of Soft Error Rate Against Memory Elements Spacing and Clock Skew in a Logic with Triple Modular Redundancy in a 65nm Process", TWEPP, 2019

High Level Synthesis (HLS) tools for a neural processor design

Open-source framework in the community

Intermediate format, Python to C++

Commercial HLS tool from Siemens C/C++ to HDL

RTL Generation

Digital Design Flow

- **HLS4ML framework**
  - Generates C++ code for the neural network
  - Reads in weights and bias from Qkeras
  - Supports only MLP and CNN
- **Catapult**
  - Maps C++ code to RTL (Verilog / VHDL)
  - Also offers verification framework

S.Miryala et.al., Peak Prediction Using Multi Layer Perceptron (MLP) for Edge Computing ASICs Targeting Scientific Applications ISQED, 2022

Acknowledgements

- Velopix design team
- IC Design team, Nikhef
- RD53 Collaboration
- ASIC group, Fermilab
- DUNE Collaboration
- ASIC group, BNL
- CERN Microelectronics
- BNL LDRD 021-23, PI: Sandeep Miryala
- Cadence Support

!!! THANK YOU FOR YOUR ATTENTION !!!

Brookhaven National Laboratory