Performing Functional ECO on Hierarchical Designs having Multiply Instantiated Modules

Jurair Hamid Bhat Associate Staff Engineer
Samsung Semiconductor India Research

Alphyn Stanley Application Staff Engineer
Synopsys

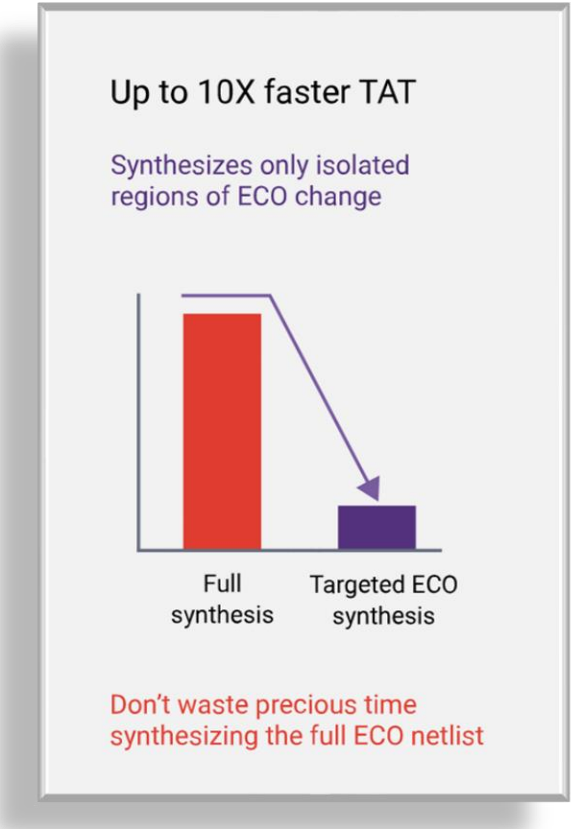
Nageswara Rao Kunchapu Associate Director
Samsung Semiconductor India Research

Agenda

- Formality ECO Introduction
- General Formality ECO flow overview
- Detailed Formality ECO flow stepwise
- Hierarchical Flow Challenges
- Formality ECO Hierarchical Flow Methodology
- Bit Blasting
- Bit Blasting - Solution
- Instance Addition In Hierarchical Design
- Instance Addition in Hierarchical Design - Solution
- Hierarchical Formality ECO results
- Conclusion
- Future Scope

Formality ECO Introduction

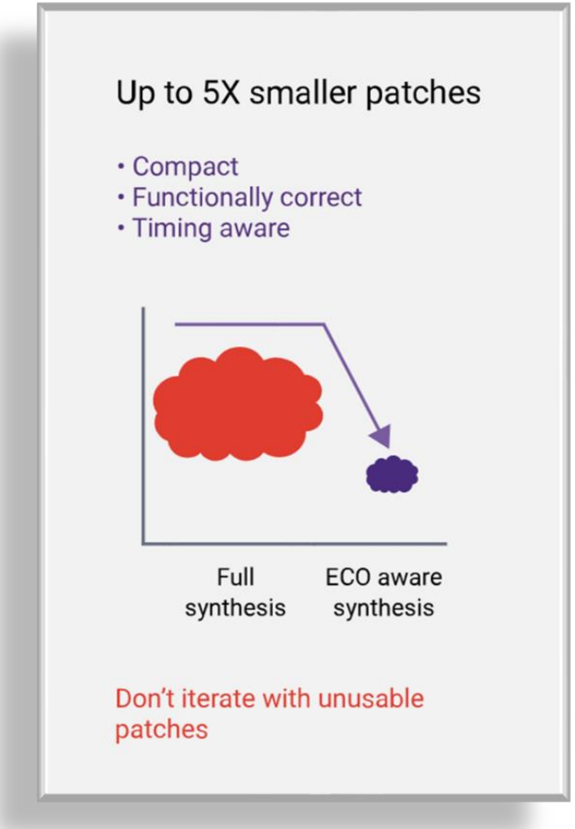
- A tool to generate ECO patch on Netlist, functionally equivalent to its ECO'd RTL
- Faster way of doing ECO which is functionally verified.
- Generates ECO patch file compatible with all Synopsys Implementation tools



Up to 10X faster TAT

Synthesizes only isolated regions of ECO change

Don't waste precious time synthesizing the full ECO netlist



Up to 5X smaller patches

- Compact
- Functionally correct
- Timing aware

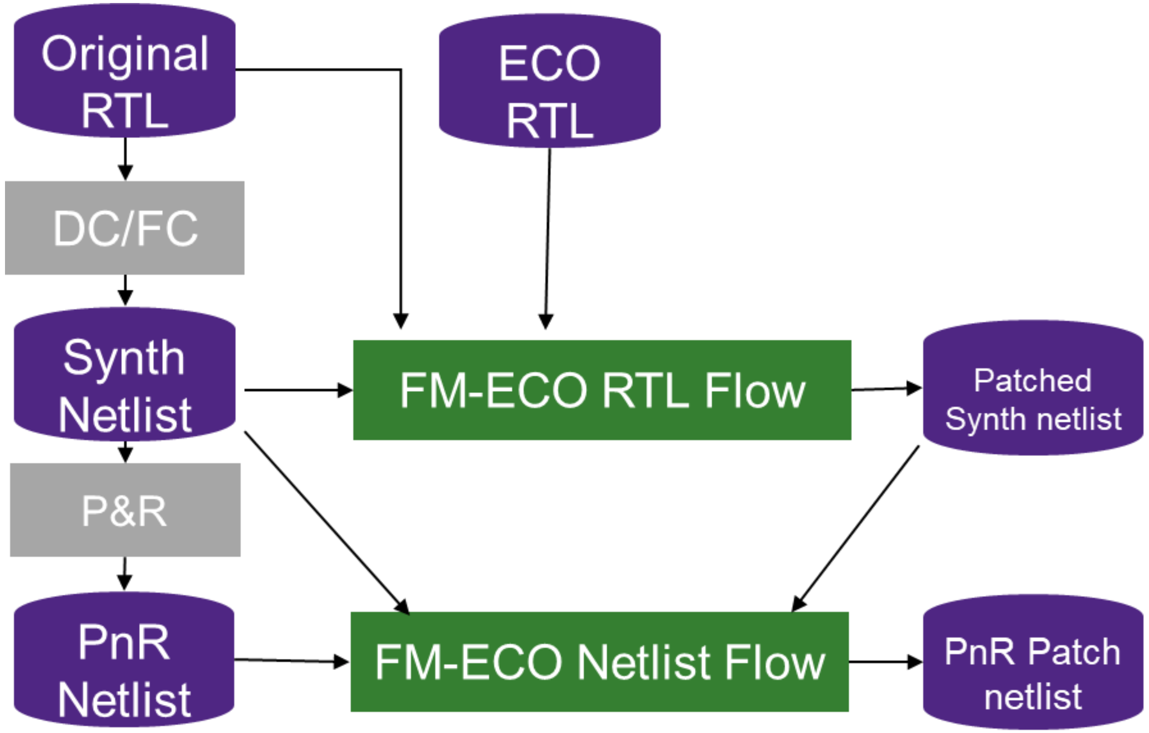
Don't iterate with unusable patches

Achieve maximal QoR

Handles all optimizations:
- Multibit banking
- Datapath

Don't sacrifice synthesis QoR

General Formality ECO Flow Overview



ECO RTL Flow

Inputs:
- Original (pre-ECO) RTL (ORTL)
- Original (pre-ECO) netlist (ONET)
- Original SVF(OSVF)
- ECO RTL (ERTL)
- Formality/Synthesis scripts

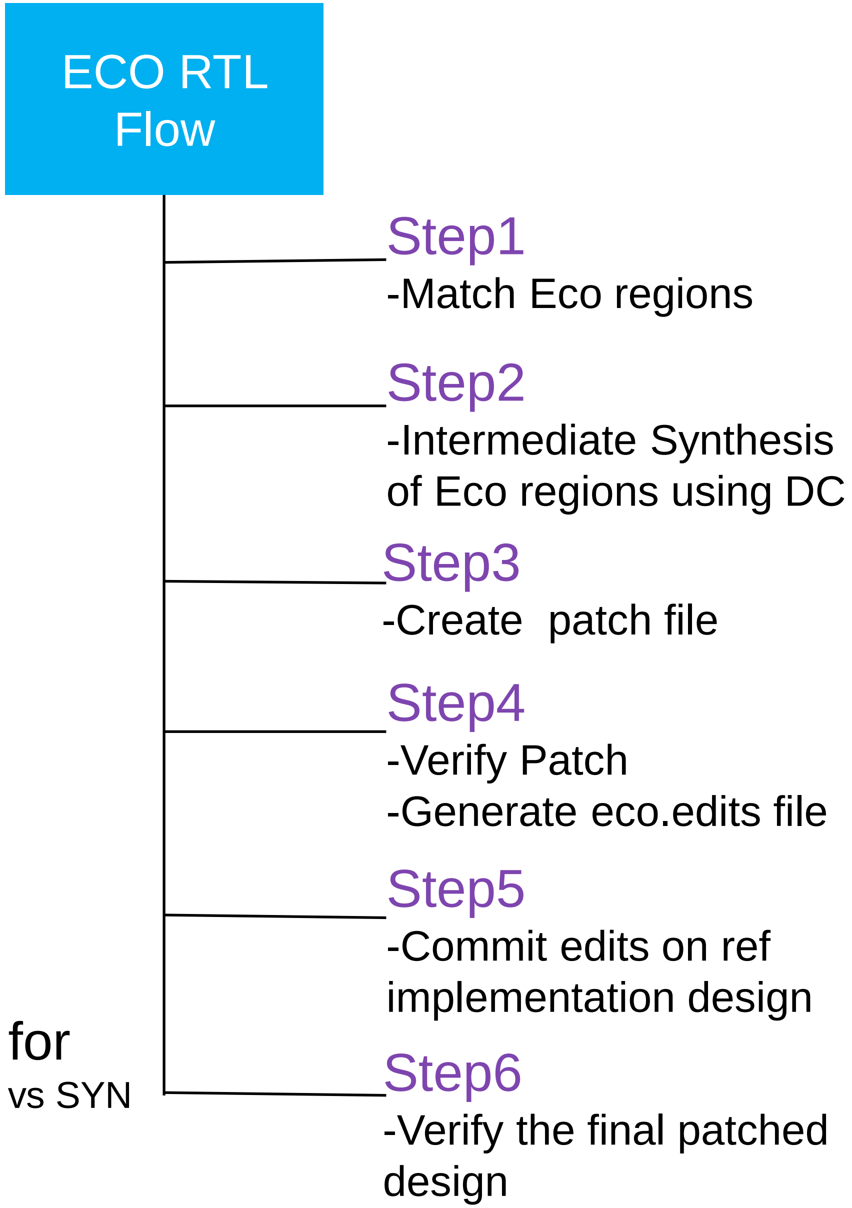
Outputs:
- Patch script

ECO Netlist Flow

Inputs:
- Original (pre-ECO) netlist (ONET)
- Previously ECO-patched netlist (PNET)
- Original netlist P&R/DFT netlist
- Formality scripts

Outputs:
- Patch script

Detailed Formality ECO Flow step-wise



ECO RTL Flow

Step1
-Match Eco regions

Step2
-Intermediate Synthesis of Eco regions using DC

Step3
-Create patch file

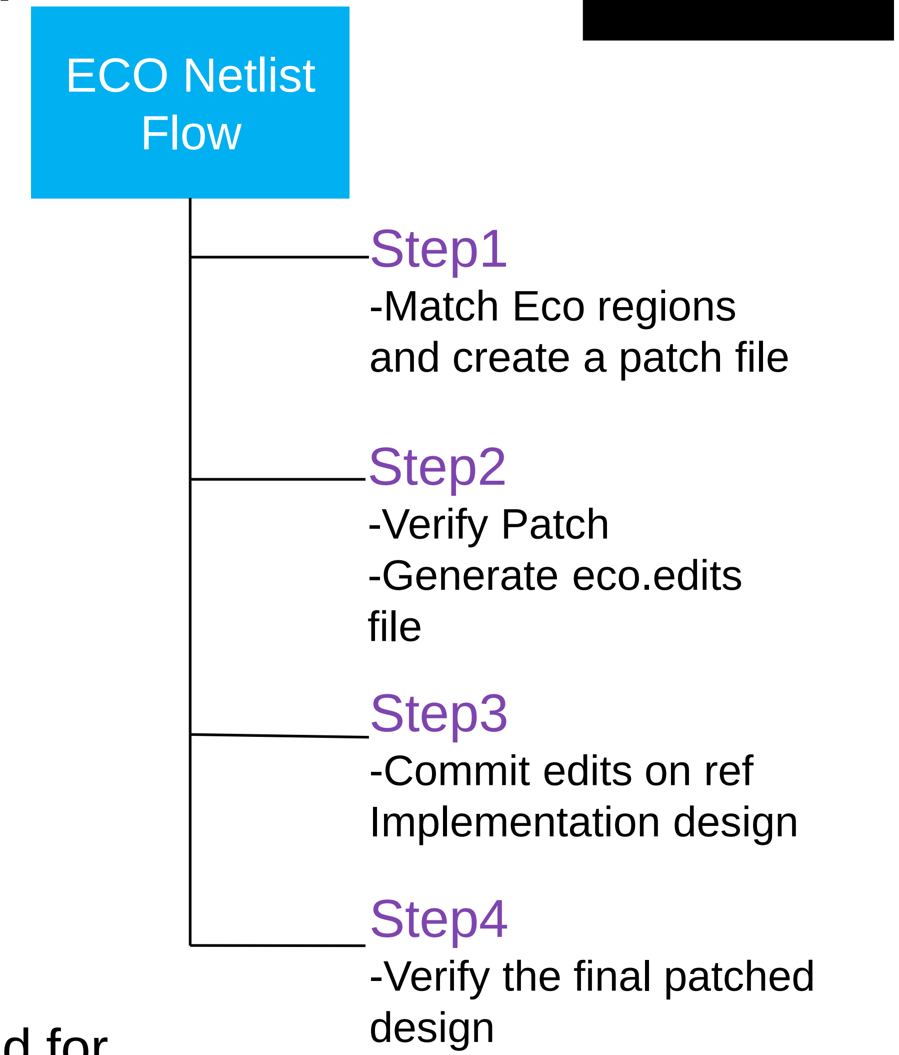
Step4
-Verify Patch
-Generate eco.edits file

Step5
-Commit edits on ref implementation design

Step6
-Verify the final patched design

Used for

- RTL vs SYN



ECO Netlist Flow

Step1
-Match Eco regions and create a patch file

Step2
-Verify Patch
-Generate eco.edits file

Step3
-Commit edits on ref Implementation design

Step4
-Verify the final patched design

Used for

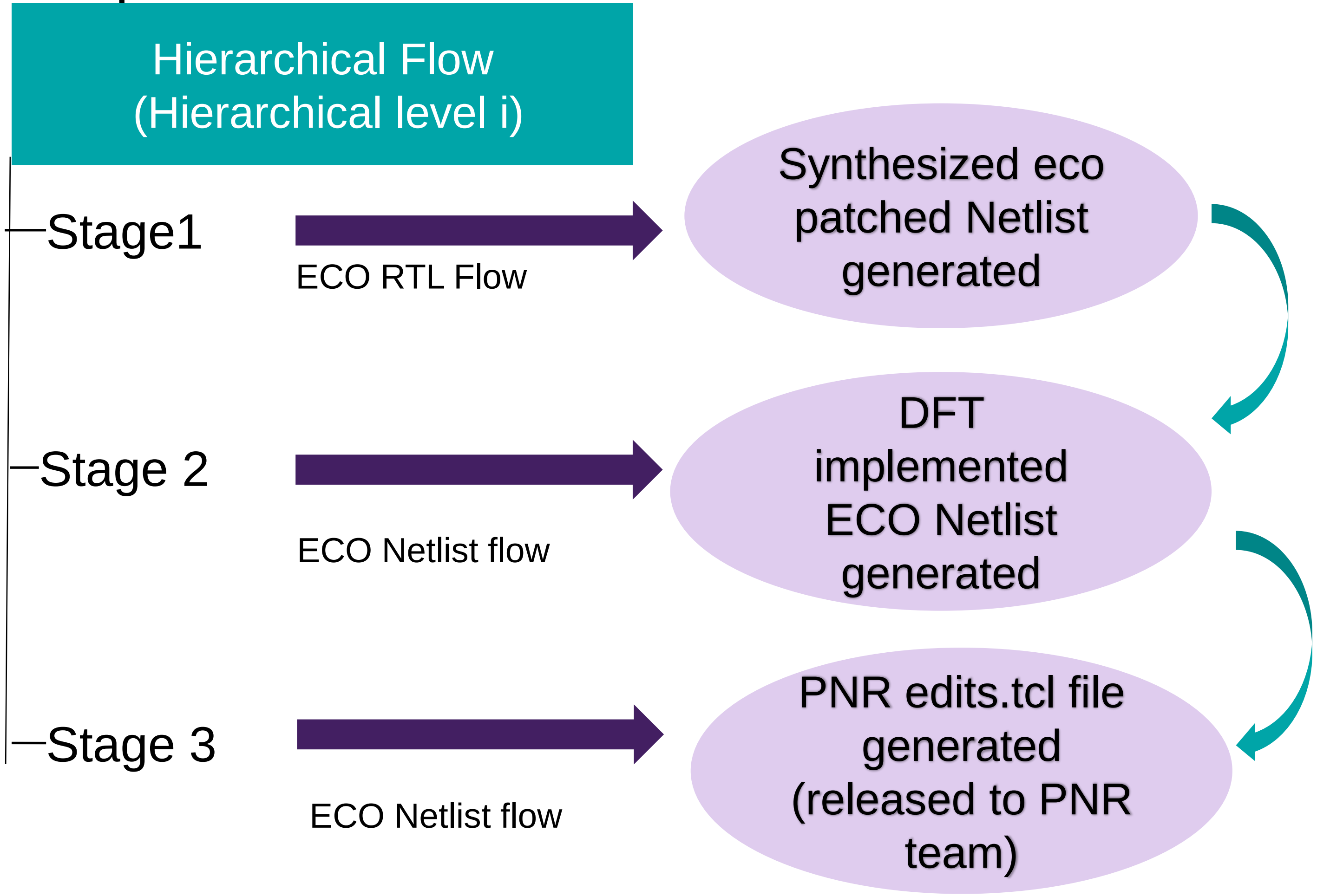
- SYN VS DFT
- DFT vs PNR

Hierarchical Flow Challenges

- Availability of Hierarchical Methodology targeting Functional ECO
- Manual ECO very risky :
  - Human errors
  - Complexity of the functional ECO and hierarchical design
  - Implementation of ECO into optimized netlist
  - Functional equivalence of the ECO; block-wise and top
  - QoR impact
- Using Generic Formality ECO flow:
  - Challenges in handling extra instance addition of submodules
  - Challenges in handling the bit blasting of buses in implemented designs

Formality ECO Hierarchical Flow - Methodology

Formality ECO performed at all hierarchical levels independently with subsequent lower hierarchies black boxed



Hierarchical Flow
(Hierarchical level i)

- Stage1 → ECO RTL Flow → Synthesized eco patched Netlist generated
- Stage 2 → ECO Netlist flow → DFT implemented ECO Netlist generated
- Stage 3 → ECO Netlist flow → PNR edits.tcl file generated (released to PNR team)

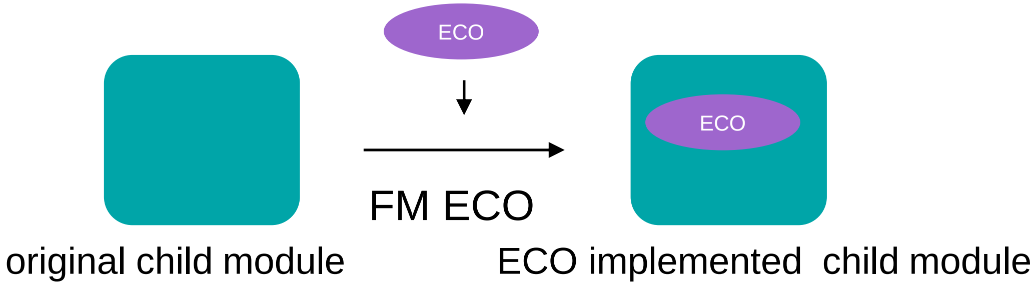
Features :

- ECO can be implemented parallelly for different blocks at different hierarchies
  - Less complicated
  - Time efficient
- No dependency between different blocks .
- ECO introduced in one block does not alter other blocks
  - ECO of any block can be altered as per need

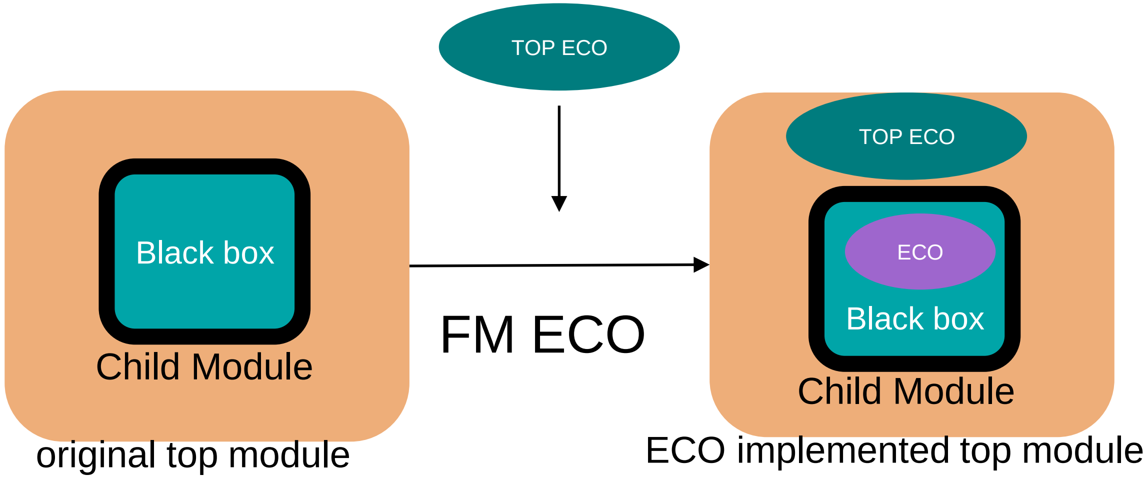
Formality ECO Hierarchical Flow - Methodology

Black Box subsequent lower hierarchies while performing hierarchical ECO on top

Child Block ECO



Top Block ECO

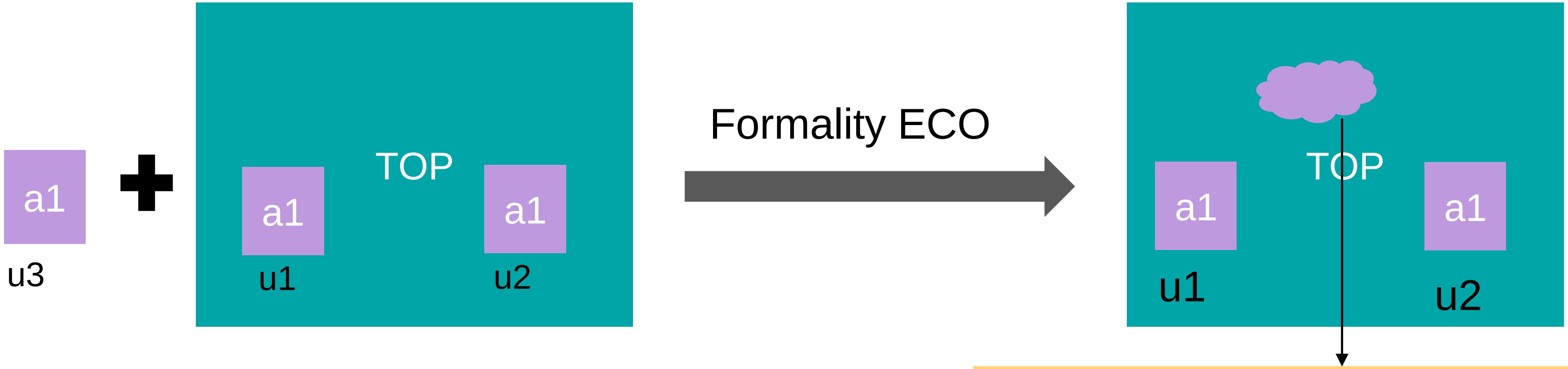


Child module black boxed

- Child module remains untouched inside
- Only ECO happens at or outside the interface of child module

Instance addition in Hierarchical design

- If extra instances of submodules is added as a part of ECO . The tool is not able to make and maintain that hierarchy in ECO implemented design
- The tool optimizes and flattens that hierarchy resulting in unwanted ECO implemented design

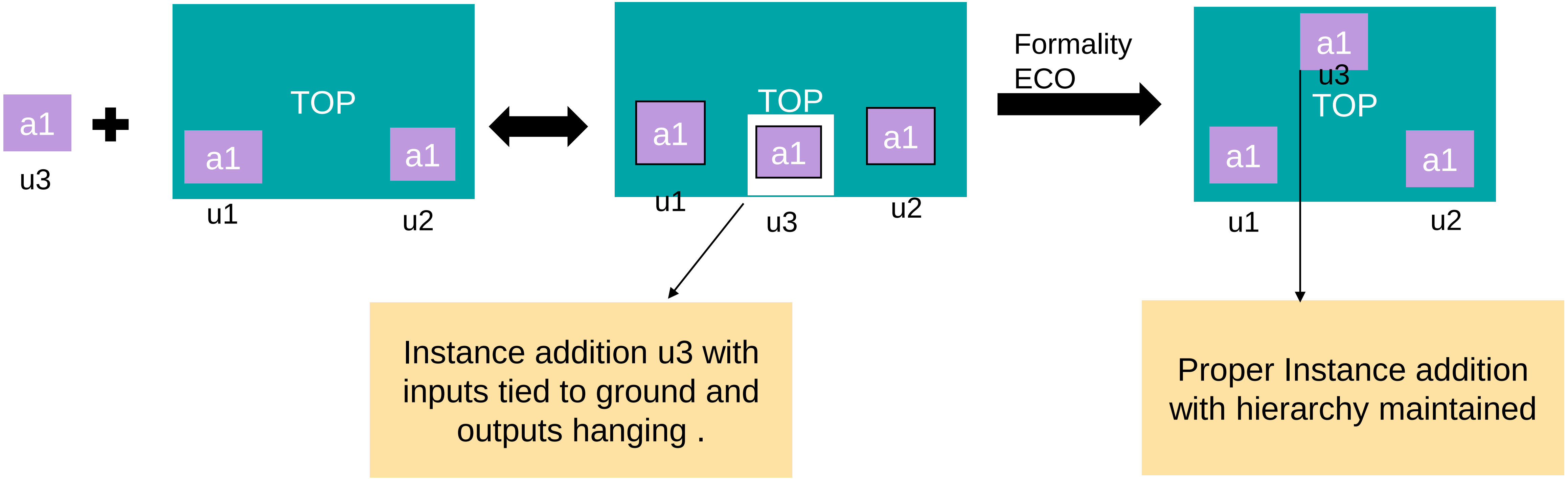


Instance u3 of sub block a1 to be implemented as part of ECO in top block

- ECO implemented but design flattened
- Hierarchy removed

Instance addition in Hierarchical design - Solution

- Add extra instances with inputs tied to ground and outputs hanging.
- Then black box the all child modules
- The eco flow will automatically update connections on the input and output side

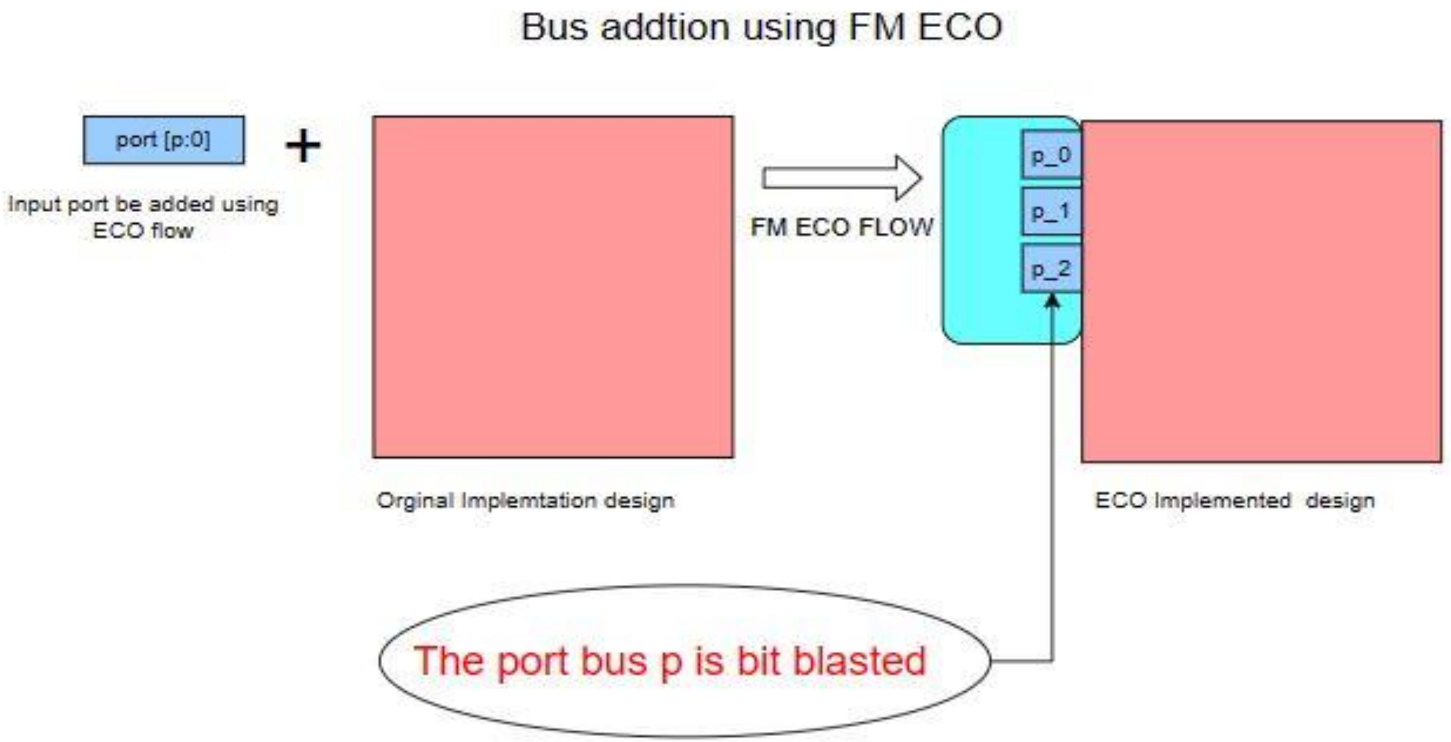


Instance addition u3 with inputs tied to ground and outputs hanging .

Proper Instance addition with hierarchy maintained

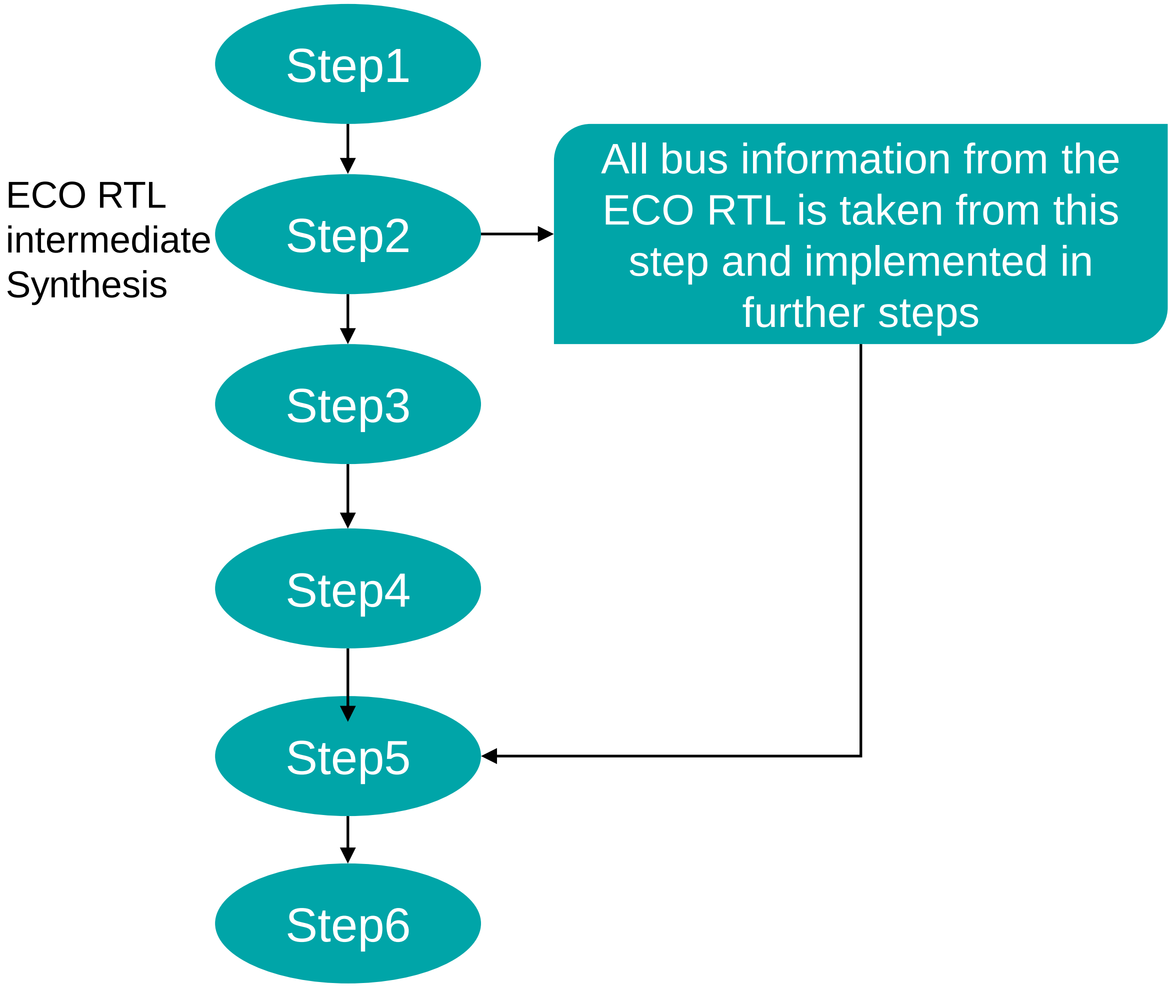
Bit Blasting

- If the bus addition or bus width elongation happens in ECO , the Formality ECO dumps these buses in final ECO implemented design as **bit blasted .**
- This results in linking issues when instantiated in subsequent top design


Bus addtion using FM ECO
port [p:0]
+
Input port be added using ECO flow
FM ECO FLOW
p_0
p_1
p_2
Orginal Implemtation design
ECO Implemented design
The port bus p is bit blasted

Bit Blasting - Solution

- The bus information is taken from intermediate netlist generated from synthesis of ECO RTL in second step of RTL Netlist flow
- The bus information is used to club the ports, which form the bus, together in all ECO implemented designs


Step1
ECO RTL intermediate Synthesis
Step2
All bus information from the ECO RTL is taken from this step and implemented in further steps
Step3
Step4
Step5
Step6

Bus information is taken from intermediate synthesis and reused in later stages to get bus information implemented correctly

Hierarchical Formality ECO Results

RTL vs Syn

Child block	Runtime (hrs)
Child FM ECO	(.255,.24,.257,.23)
Top	.295
Total	**Max (child,top) = .295**

Syn vs DFT

Step	Runtime (hrs)
Child FM ECO	(.04, .042, .039, .041)
Top	.053
Total	**Max (child,top) = .053**

DFT vs PNR

Step	Runtime (hrs)
Child FM ECO	(.035, .033, .032, .031)
Top	.05
Total	**Max (child,top) = .05**

Faster, parallel and functionally correct

Future Scope

- Bus creation during ECO should be supported
- Extra black boxed instance addition should be supported

Conclusion

- Hierarchical ECO achieved
- Generalized hierarchical FM ECO flow for any number of hierarchies achieved
- Turn around time very less
- First time right and verified patch achieved

SAMSUNG snug

THANK YOU

Our Technology, **Your Innovation**™