

Design and Verification Challenges of ODC-based Clock Gating

Chaiyasit Manovit
Sridhar Narayanan
Sridhar Subramanian

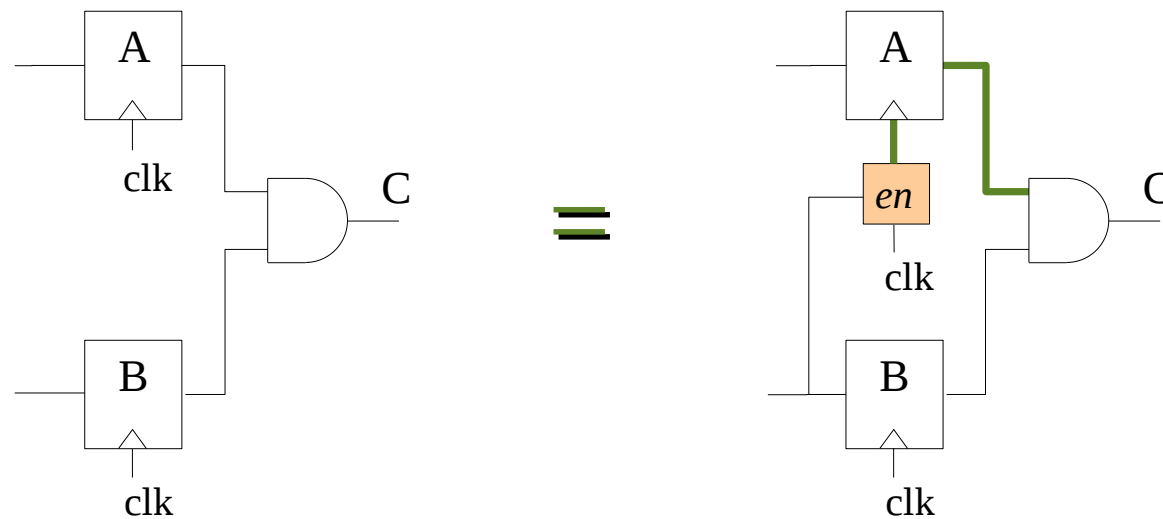
PwrLite, Inc.
Santa Clara, USA
info@pwrlite.com



DESIGN AUTOMATION CONFERENCE

ODC-based Clock Gating Example

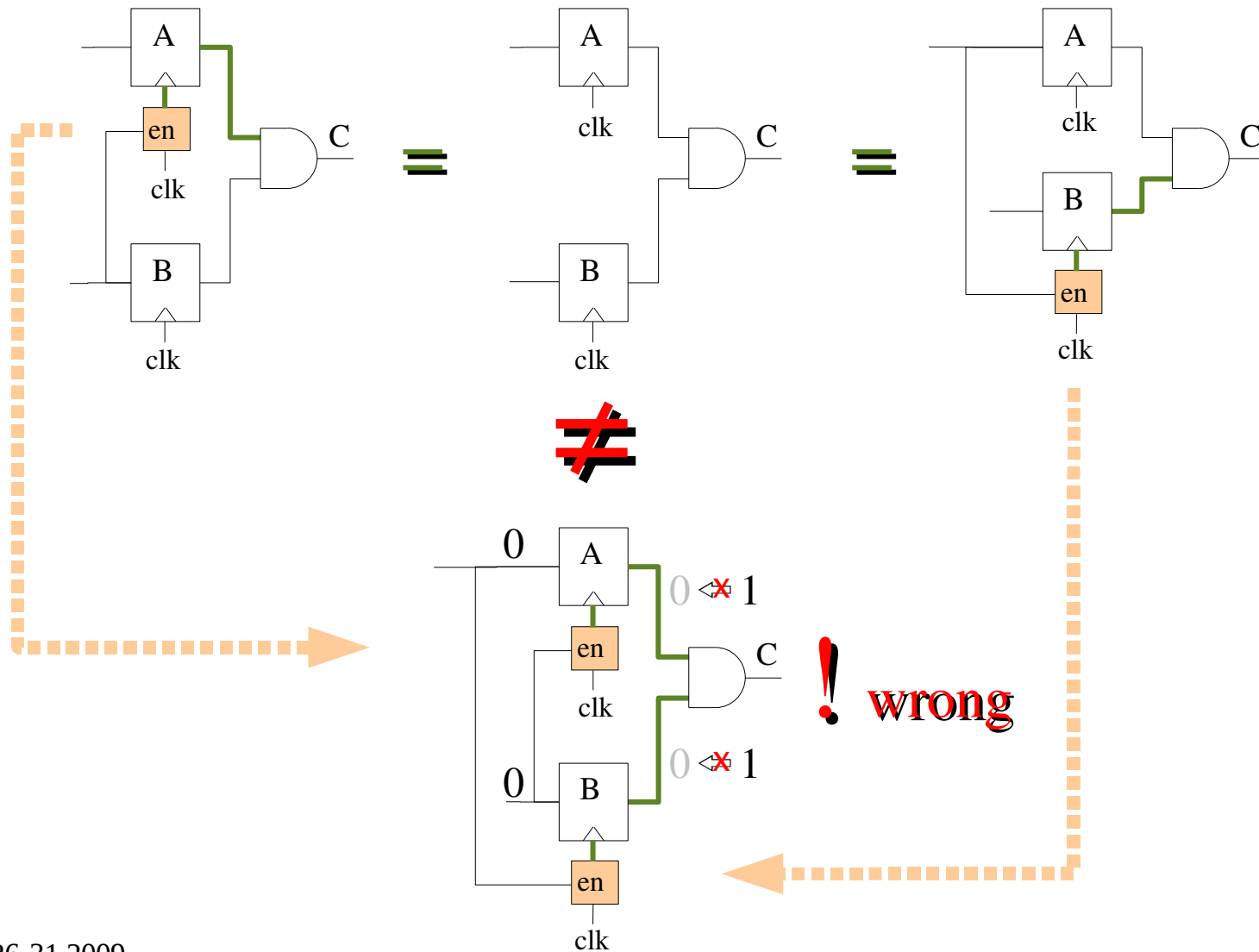
- A's output not observable if B's output is 0
→ No need to clock A's *input* if B's *input* is 0
- Energy saving on clock *and* A's output toggling



Note: *en* = clock enable
(clock is gated when *en* is 0)

Design Challenges: Unsafe Combination

- Can't straightforwardly combine A's & B's clock gating

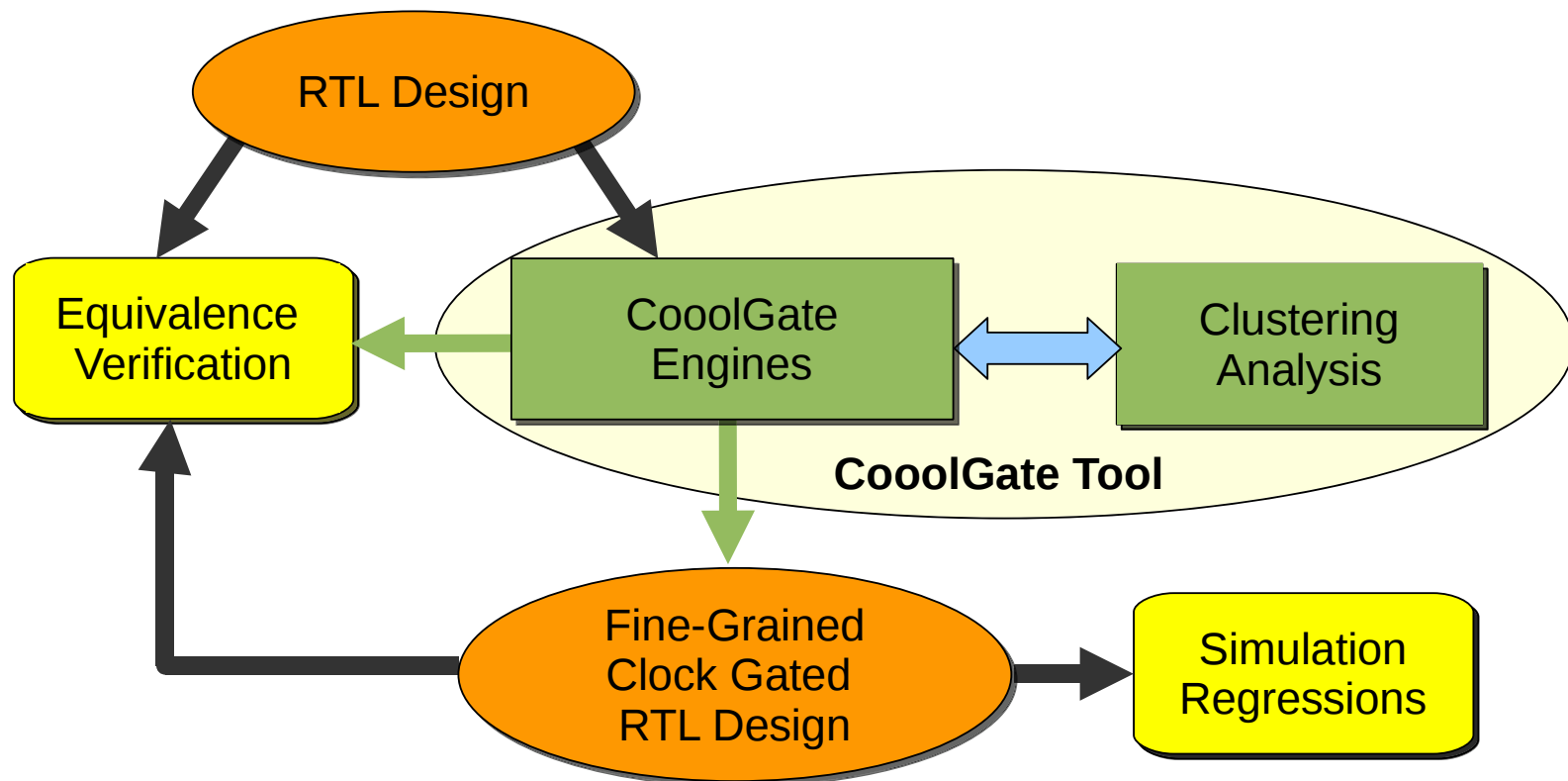


Solutions to Unsafe Combination

- Identify inter-dependency and *refine* enable functions
 - Remove inter-dependency between A and B by not gating B
- Incremental analysis
 - Include A's clock gating in analysis of B
 - B's *en* = A's input is 1 OR A's output is 1
- Flops ordering may affect quality of results
 - Heuristic to favor larger saving opportunities

CoolGateTM

- PwrLite's solution to clock gating at RTL/netlist level
 - Suggests clock gating functions with estimated efficiencies (efficiency is defined as fraction of time toggling is inhibited)
 - Can automatically modify RTL



Results

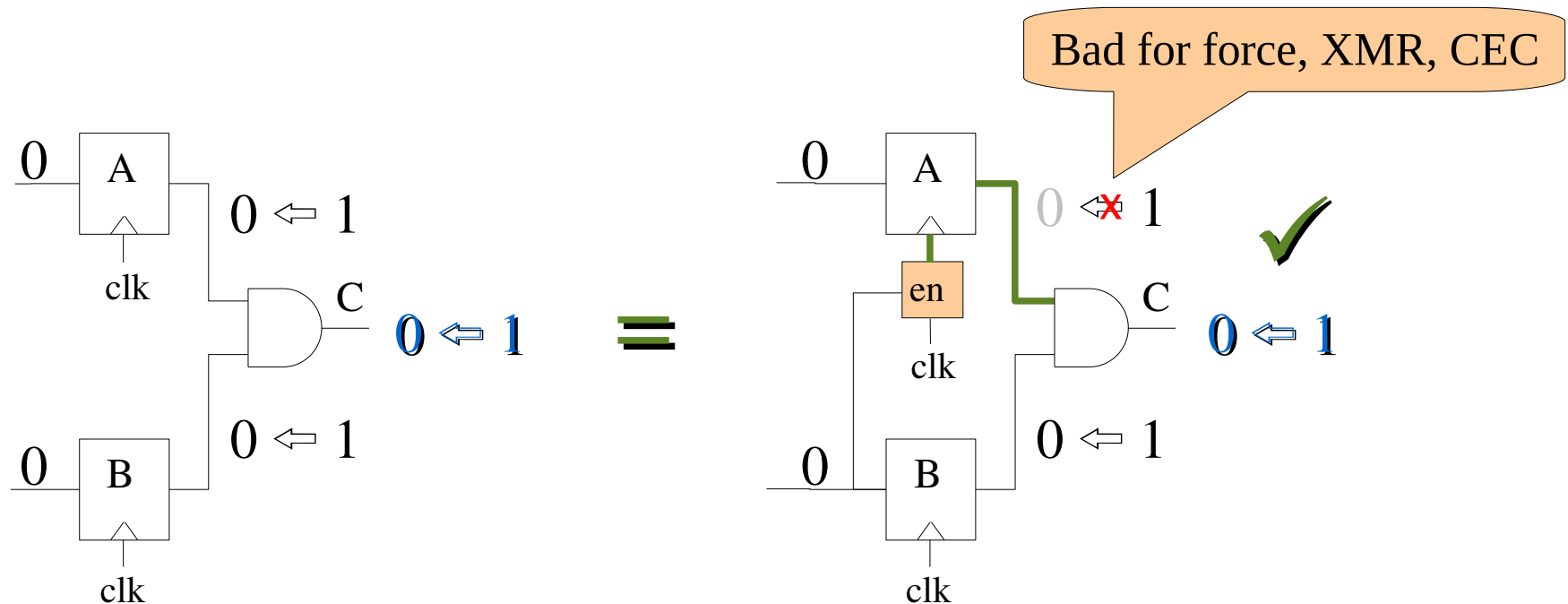
	Design#1 (Open AES)	Design#2 (OpenSPARC's dramctl)	Design#3 (A processor block)	Design#4 (A networking block)
Total flops	2239	7363	3018	17979
Gated flops with ODC-based	1562	1057	373	10009
Gated flops as percentage	70%	14%	12%	56%
% Gated flops in unsafe combinations & resolved	0%	19%	20%	6%
Efficiency ¹ - reduced clock toggling on gated flops	41%	98%	13%	9%
Efficiency ¹ - reduced data toggling on gated flops	61%	11%	0%	9%
Average number of flops per enable function	156	132	13	31
Tool's run time (seconds; on 2 GHz x86)	33	192	482	269

Notes

1. Efficiency is evaluated against given simulation dumps

Verification Challenges: False Alarms

- Simulation mismatches
 - Force on A's input has no effect if clock gated
 - Cross module reference (XMR) to A's output gets stale value
- Combinational Equivalence Check (CEC) fails on A



Simulation Aids

Monitor Code for RTL Simulation

```
module odc_checker(q, en, clk);  
  
parameter size=1;  
  
input [size-1:0] q;  
input en;  
input clk;  
  
always @(posedge clk)  
begin  
    release q;  
    if (en!=1'b1) begin  
        force q = {size{1'bx}};  
    end  
end  
  
endmodule
```

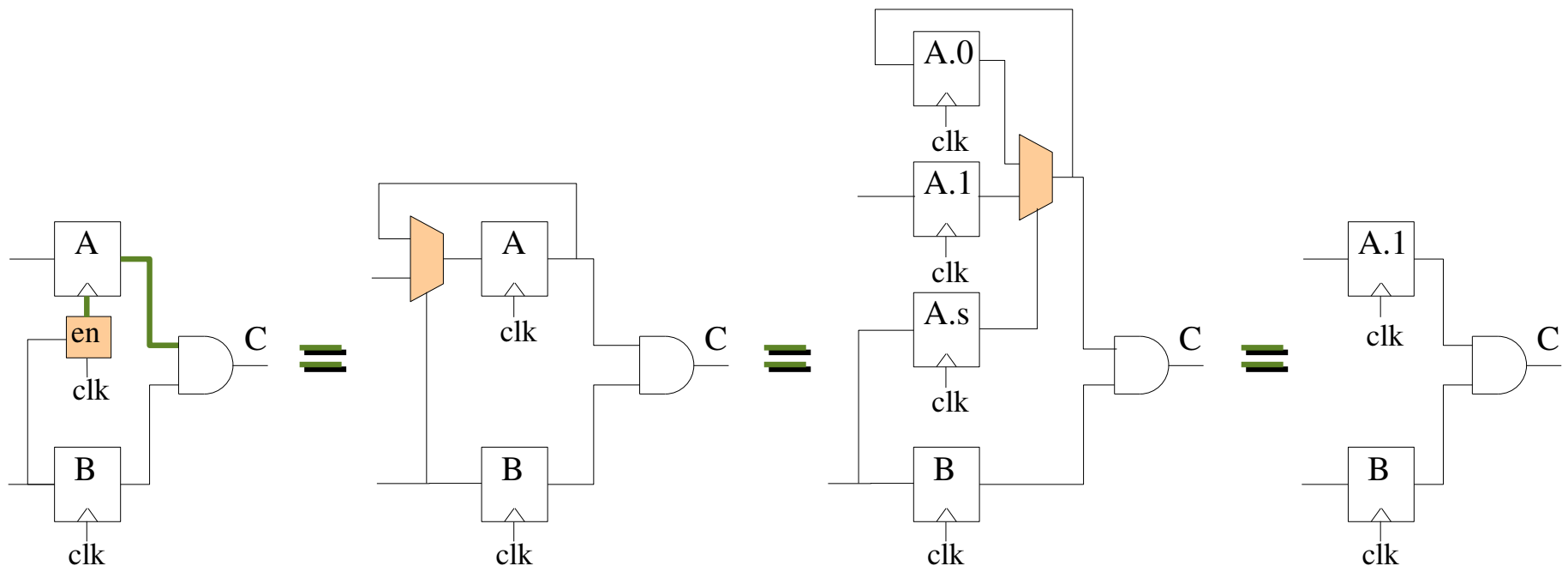
Guidelines for Simulation

- Force on “reg”, not “wire”
- Avoid XMR, or create a shadow flop

Combinational Equivalence Check

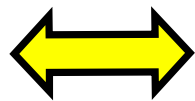
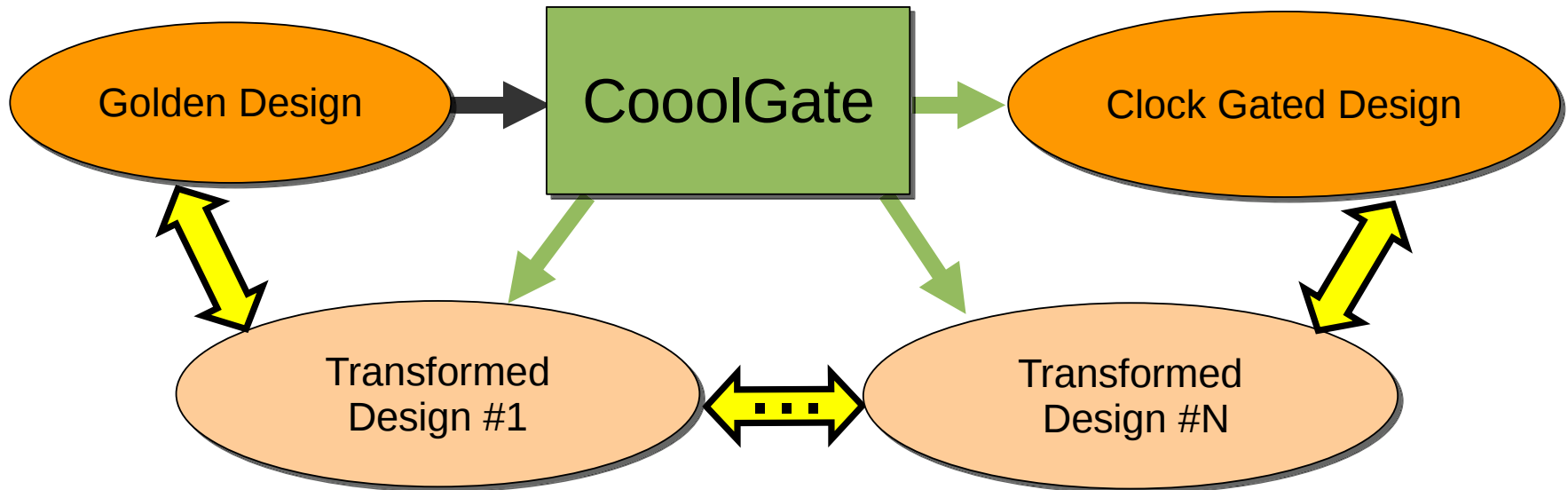
- Multiple steps of CEC (utilizing tool's advanced features)

[C. Manovit, S. Narayanan, S. Subramanian, "Method and system for verifying power-optimized electronic designs using equivalency checking," *Patent pending.*]



- Sequential Equivalence Check (SEC) not required

Equivalence Checking Flow



= Combinational Equivalence Check
(N is expected to be ≤ 4)

Summary

- ODC-based clock gating is useful but can be intricate
- Test bench may require changes regarding force/XMR
- CEC does not work as is; proof requires multiple steps
- Tools for design and verification of clock gating can be helpful