

Testing Multiple Identical Cores

Teresa McLaurin



Purpose

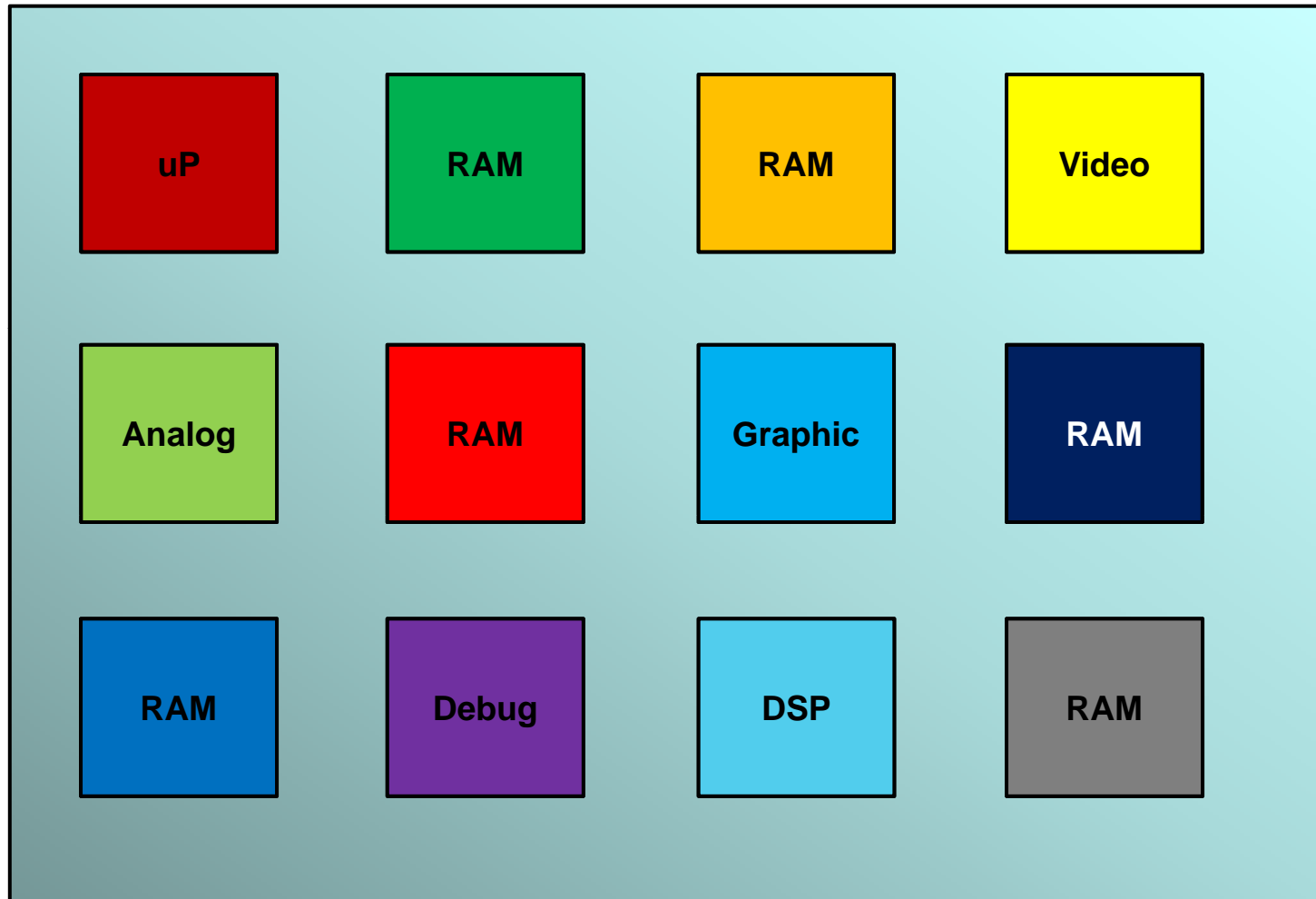
- To discuss the test implementation and execution for testing multiple identical cores
 - Preferably a methodology that is
 - automated
 - easily transferrable
 - supported by EDA tools
 - Economical in logic, pattern volume, etc.



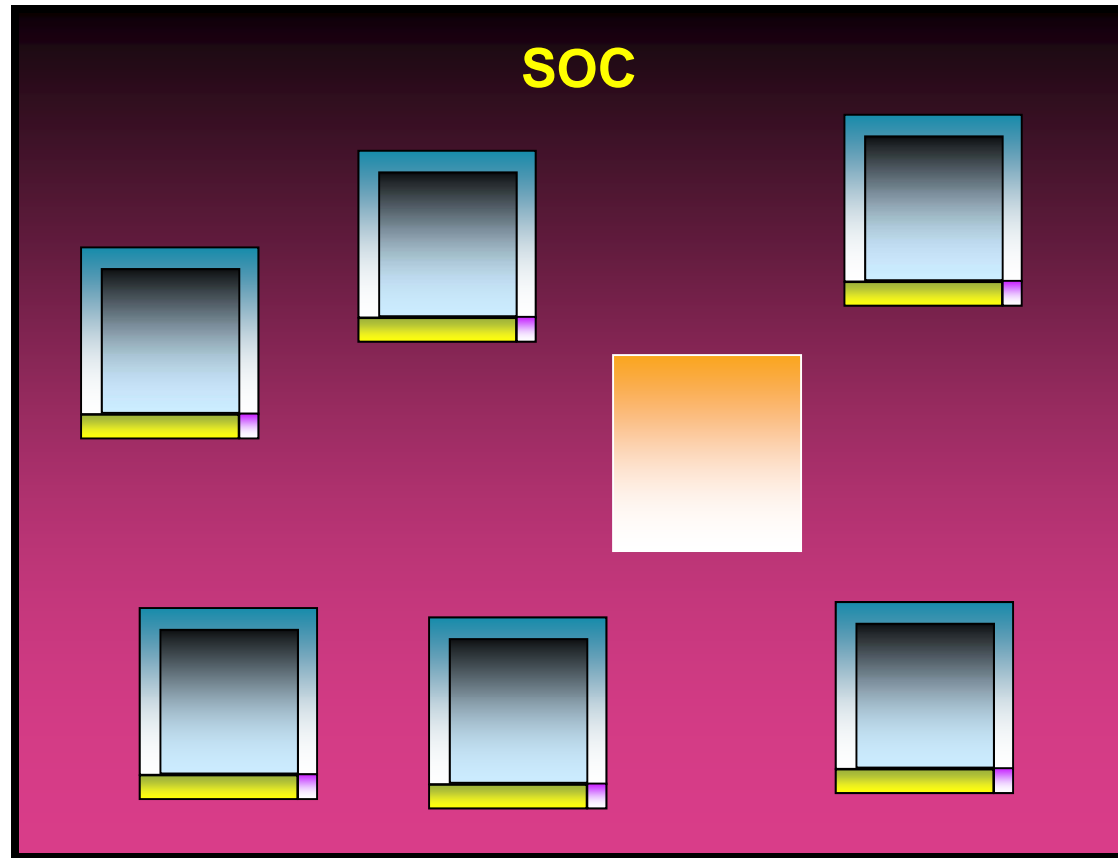
Background

- Some papers written about identical core testing
- Usually requires proprietary methodologies
- Need a simple methodology to test identical cores
 - EDA tools can support
 - Can be easily passed on to others
 - Works on all ATEs
 - Low Pin Count
 - Minimal Vectors

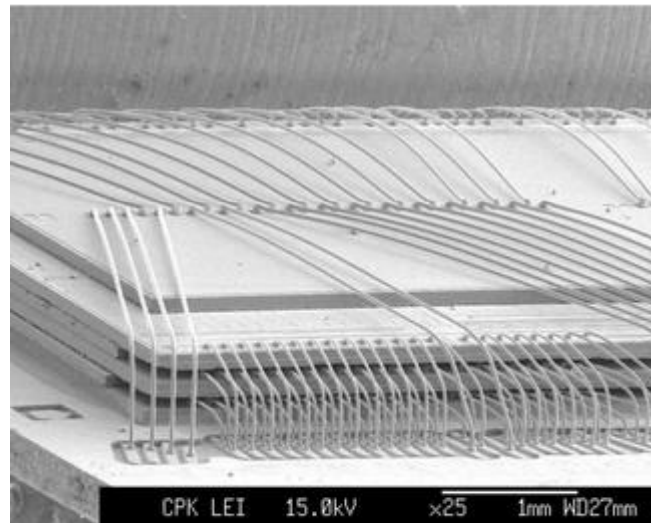
Multiple Cores



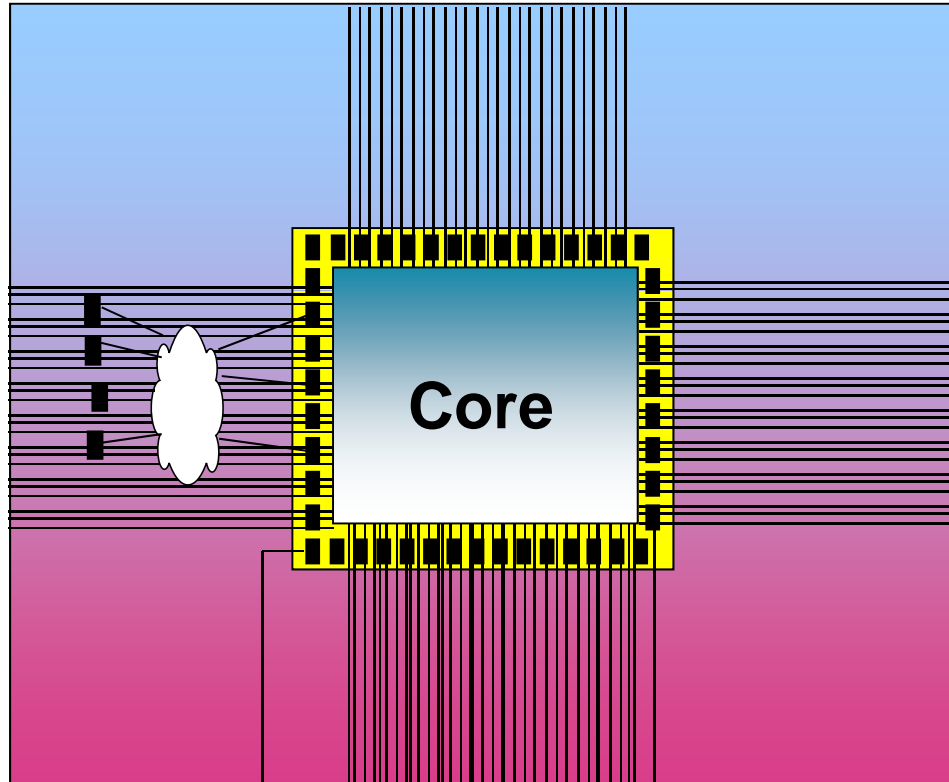
Why Isolate



Stacked Die

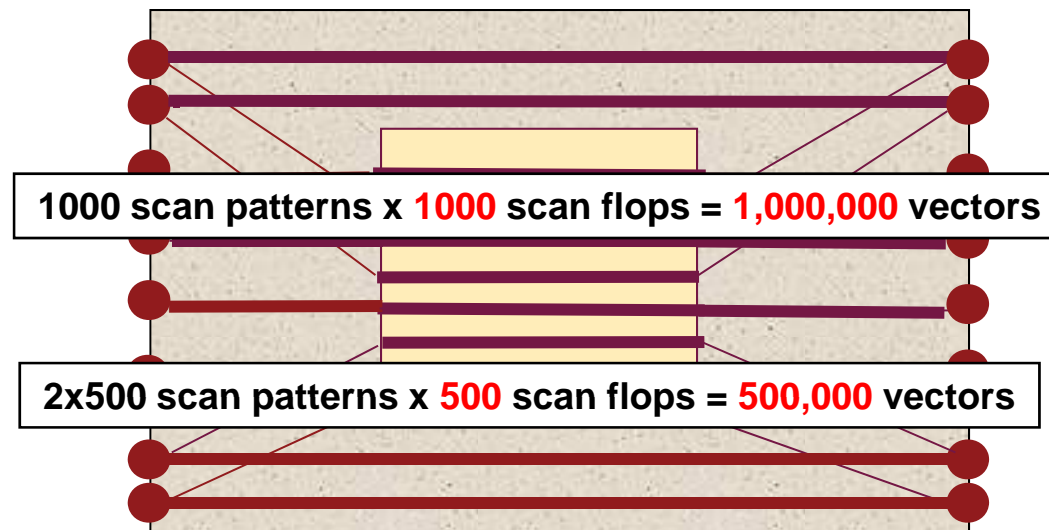


Single Core Isolation Strategy

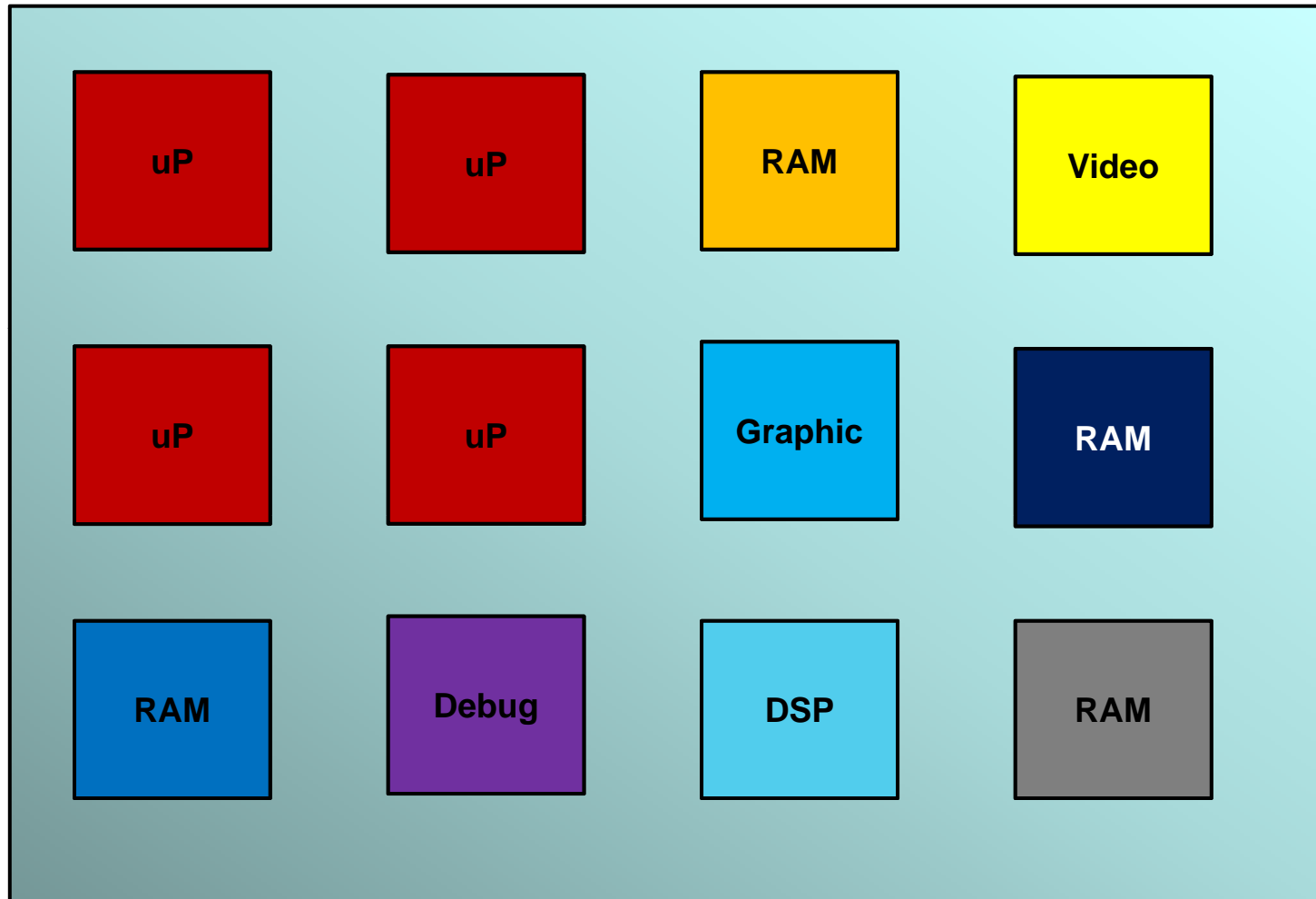


More isolation information

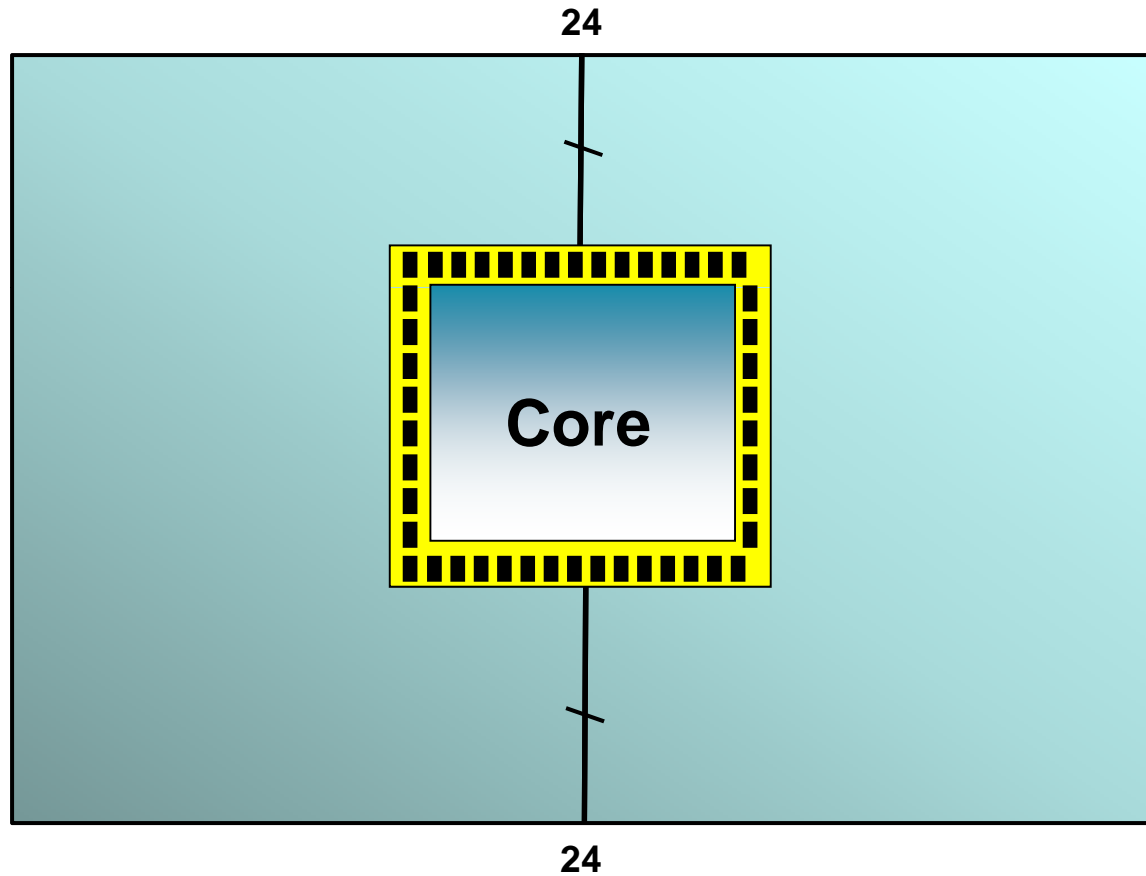
- Wrappers are a must nowadays
 - Allows for partition-based testing, debug and diagnosis – shorter scan chains/shorter test times



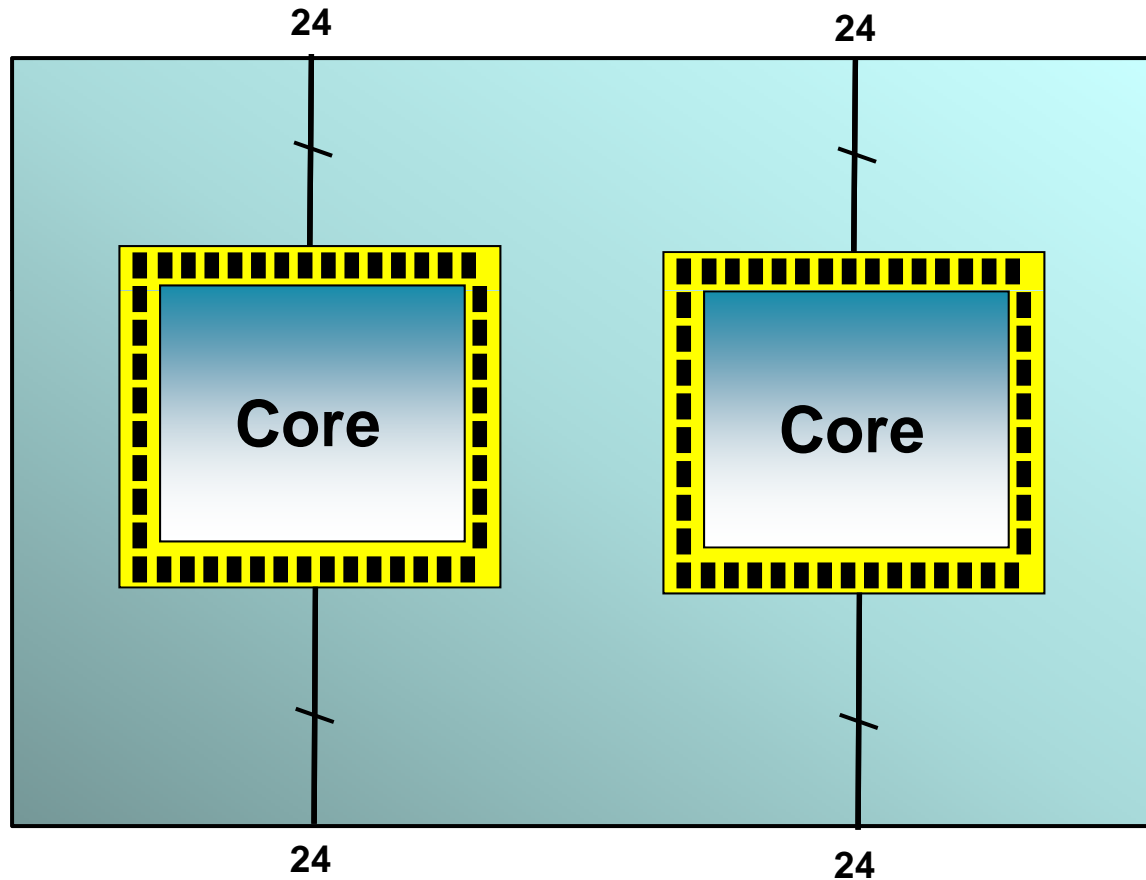
Multiple Cores



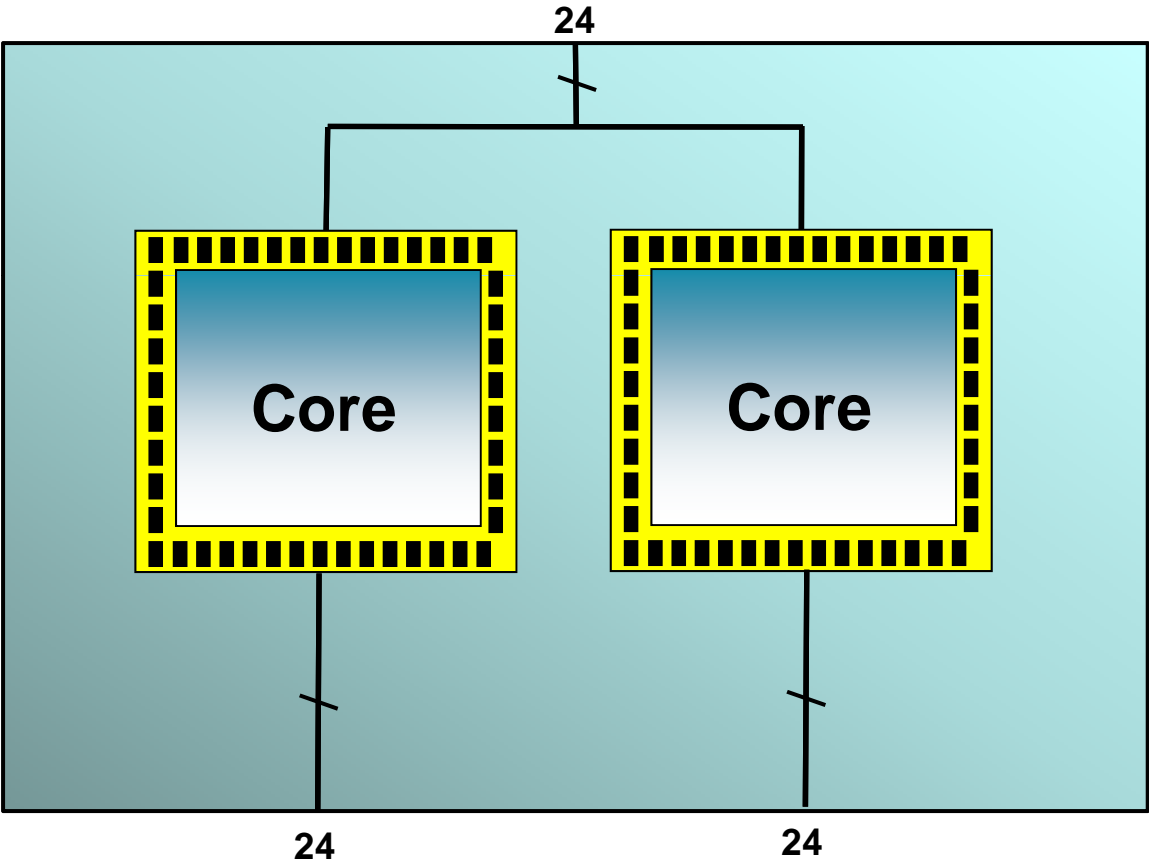
Single Core Connections



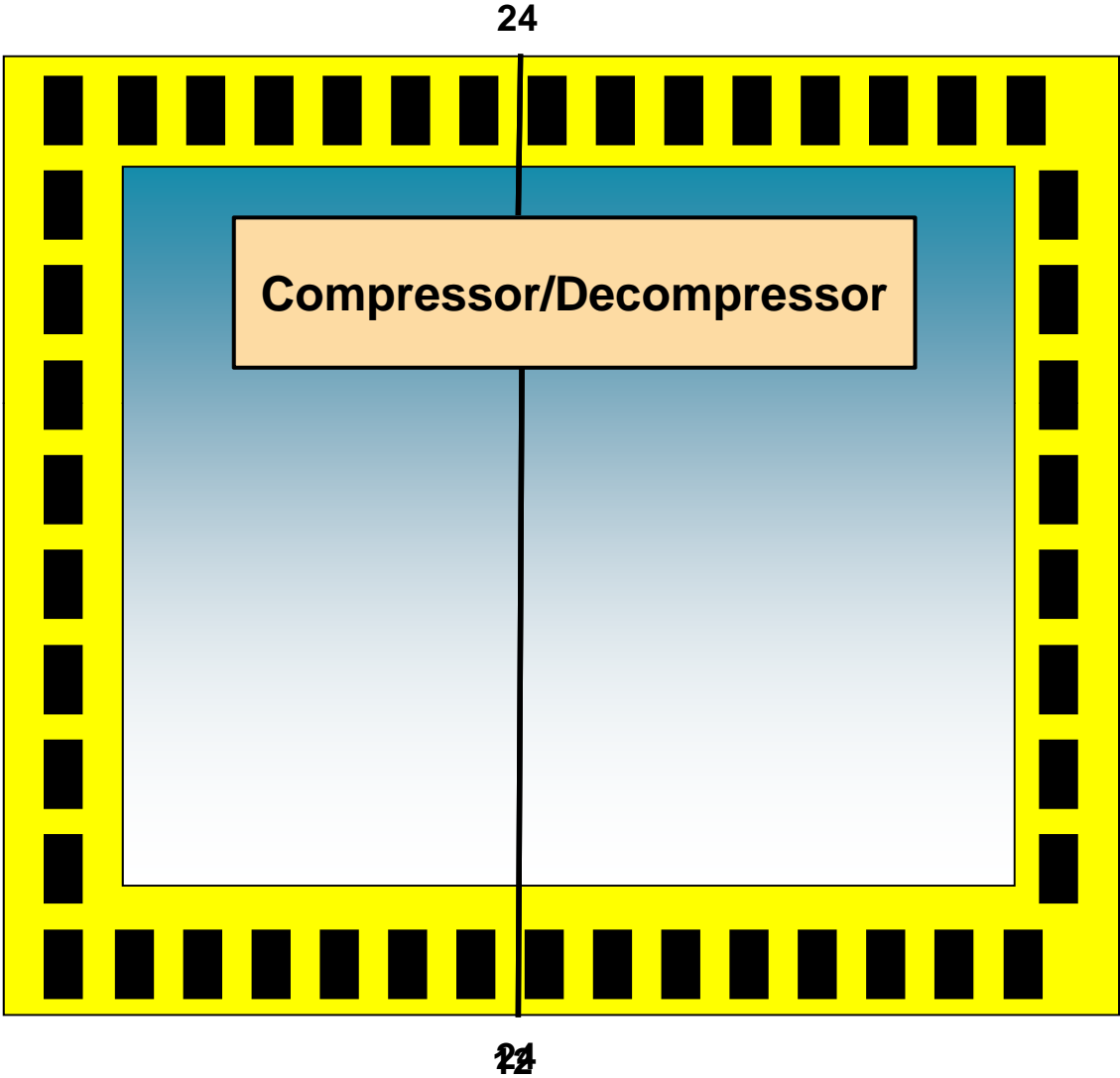
Single Core Connections



Two Identical Cores

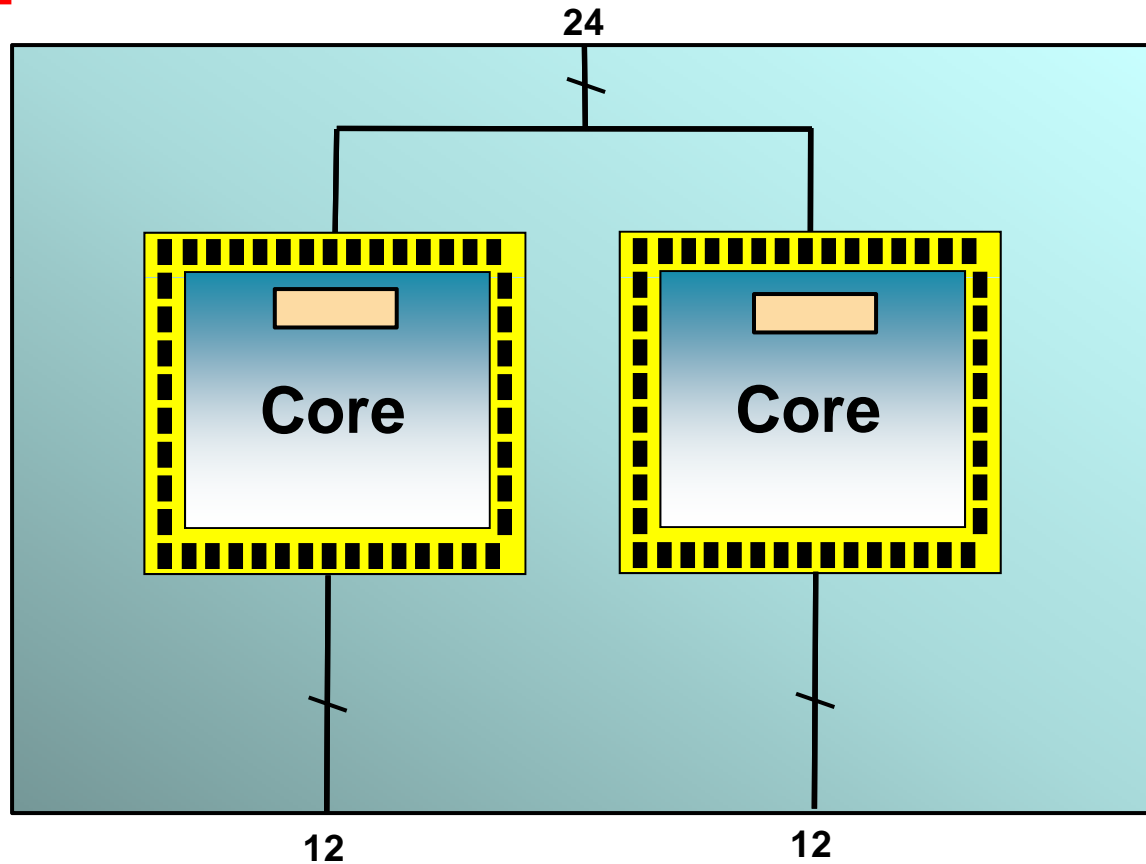


Two Identical Cores



Two Identical Cores

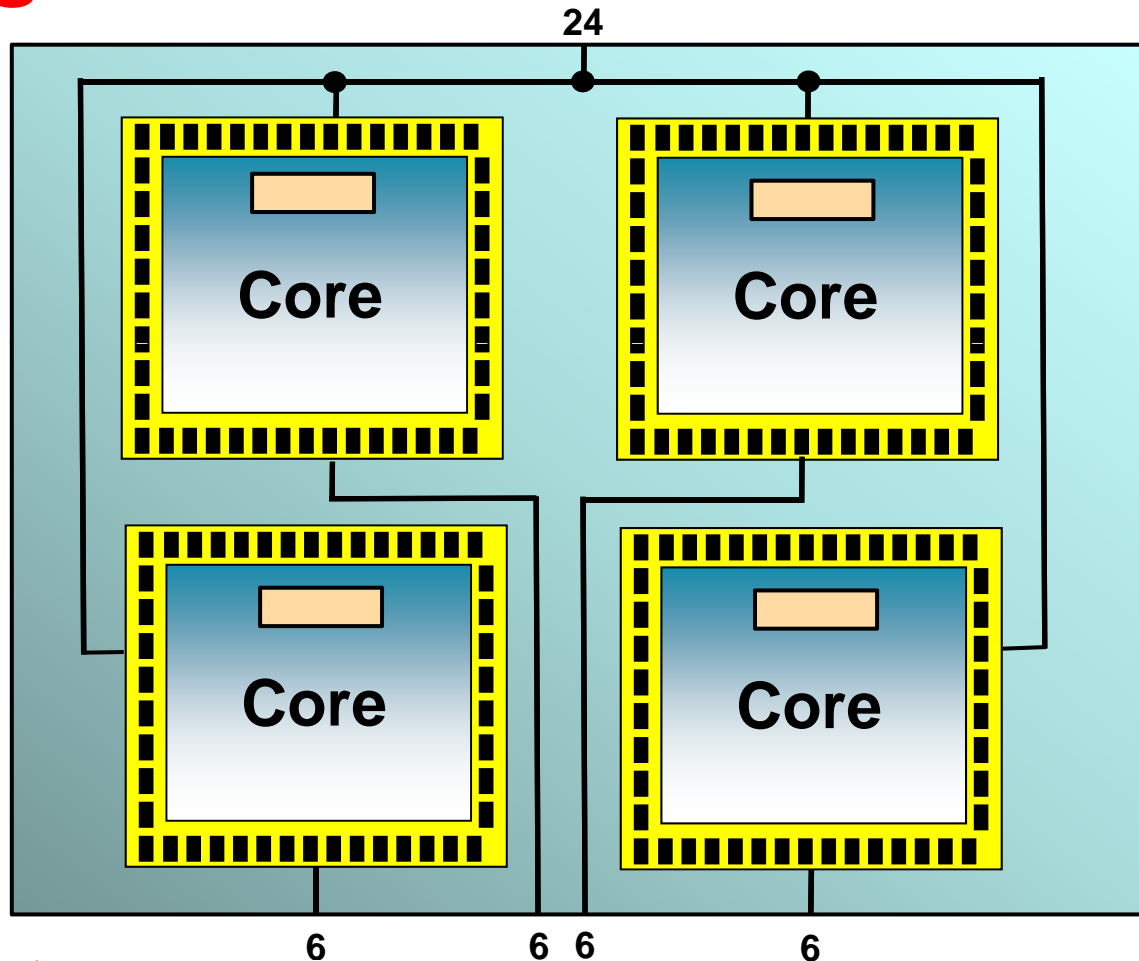
24 inputs



24 outputs

Four Identical Cores

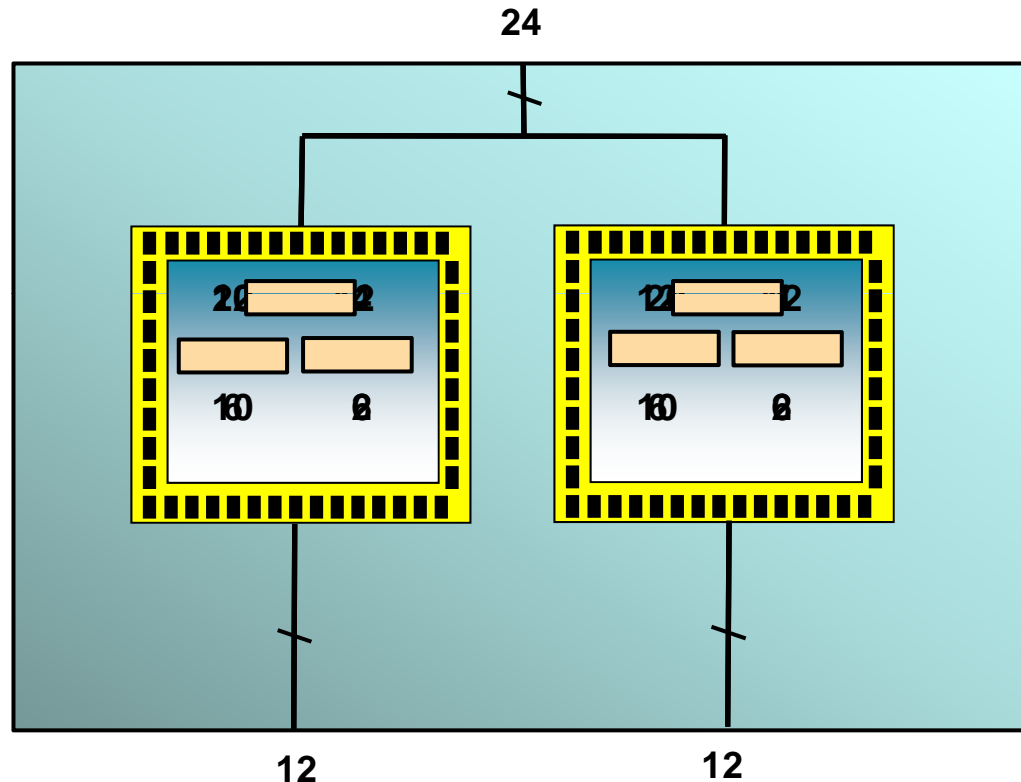
24 inputs



24 outputs

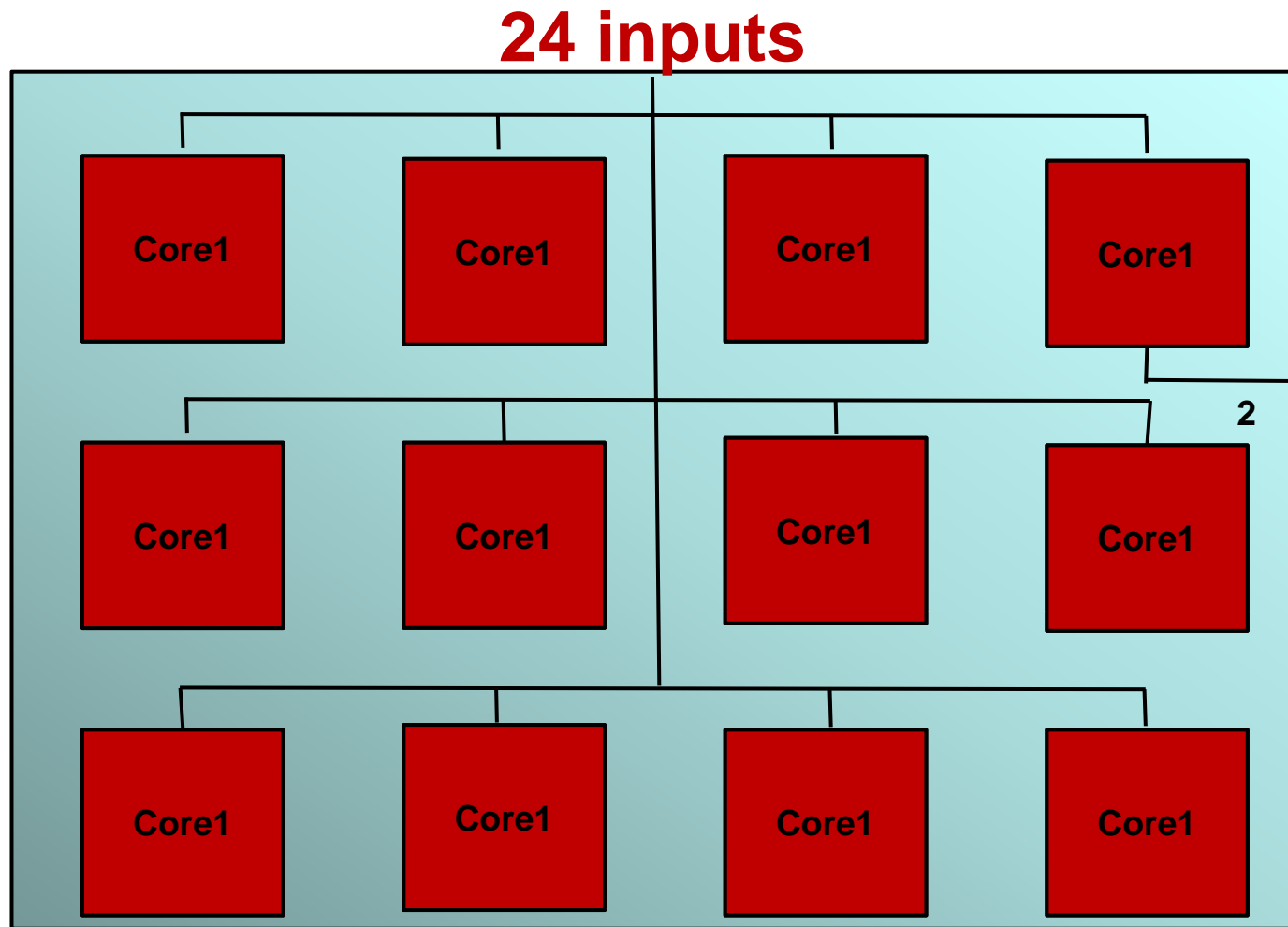
Multiple Compressors

- What if there are separate power domains



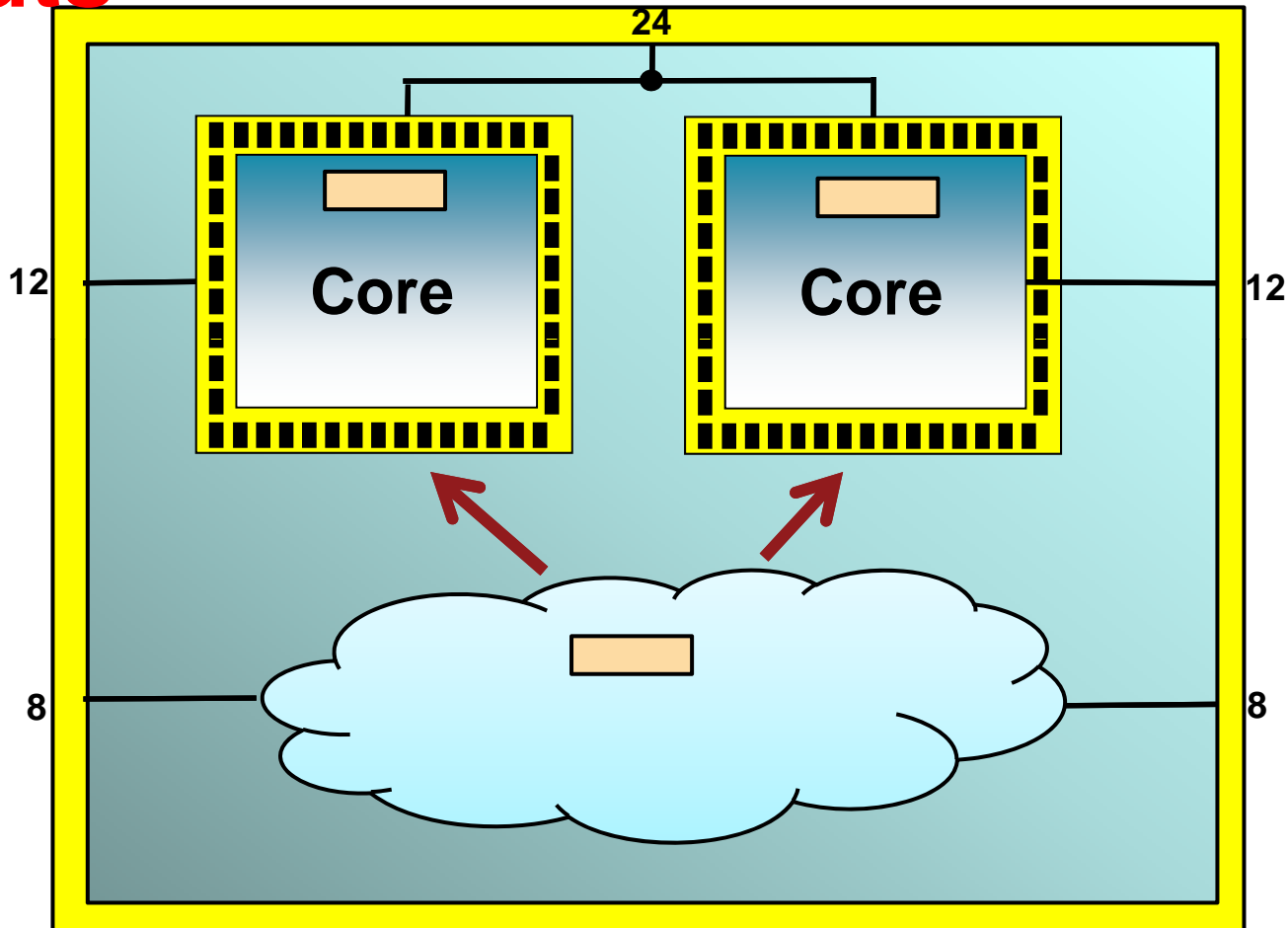
- Other things can cause the number of compressors to multiply – timing, hierarchy – leading to less test pins

Multiple Cores



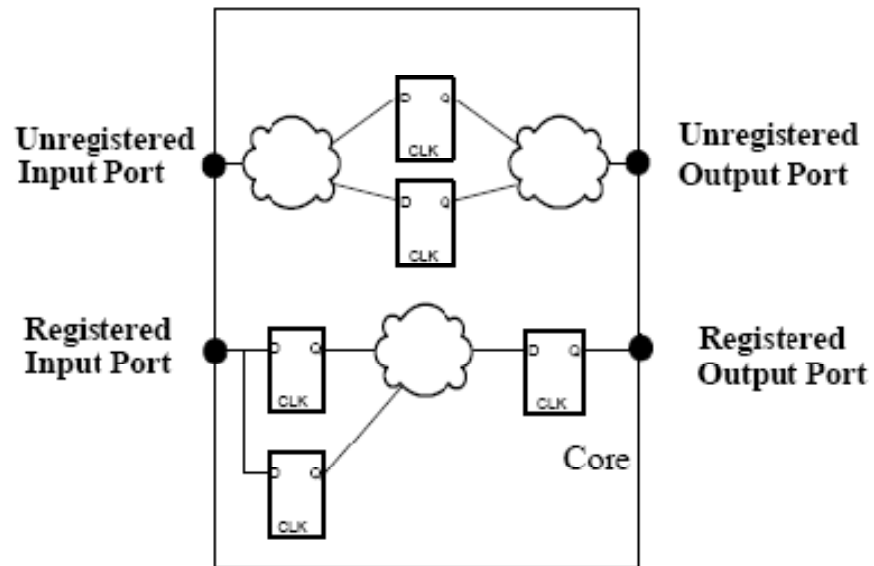
Identical Cores Plus Extra Logic

32 inputs

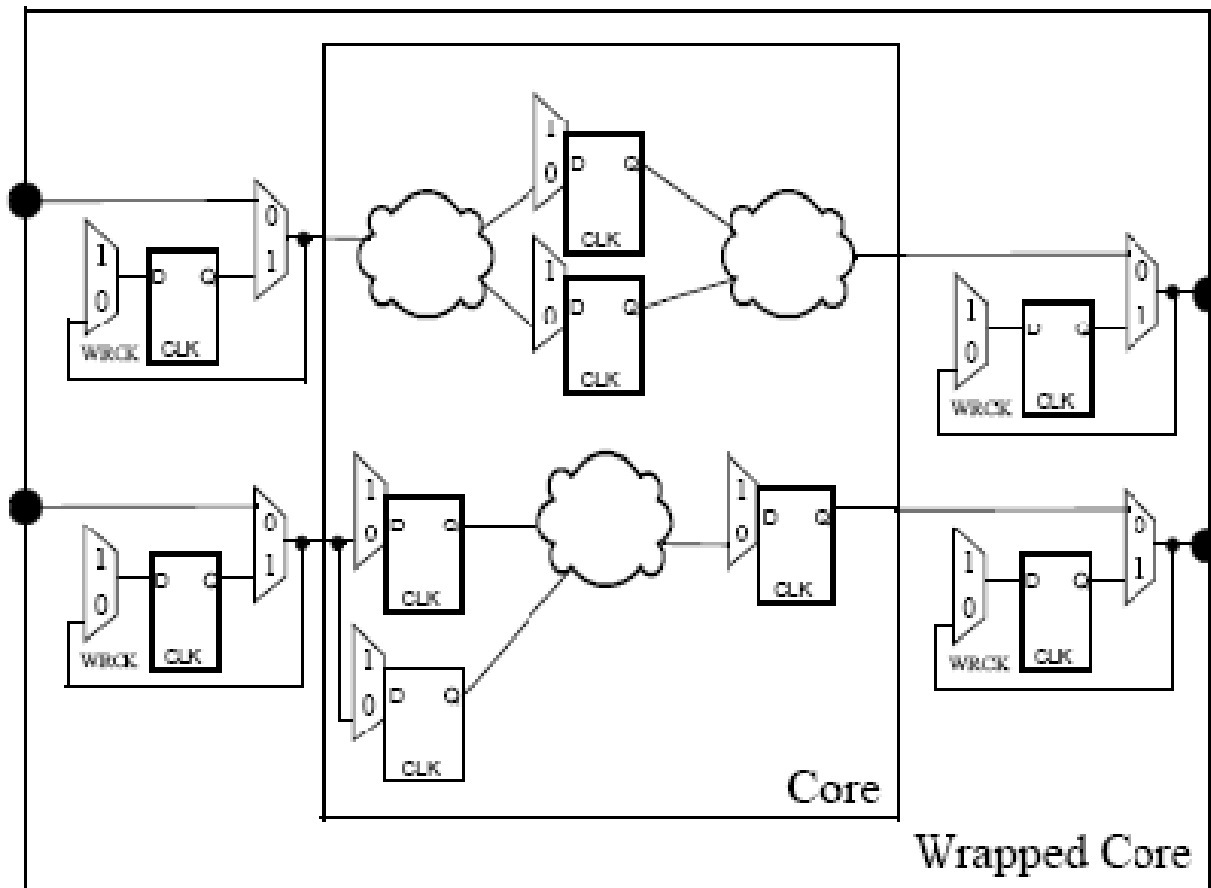


32 outputs

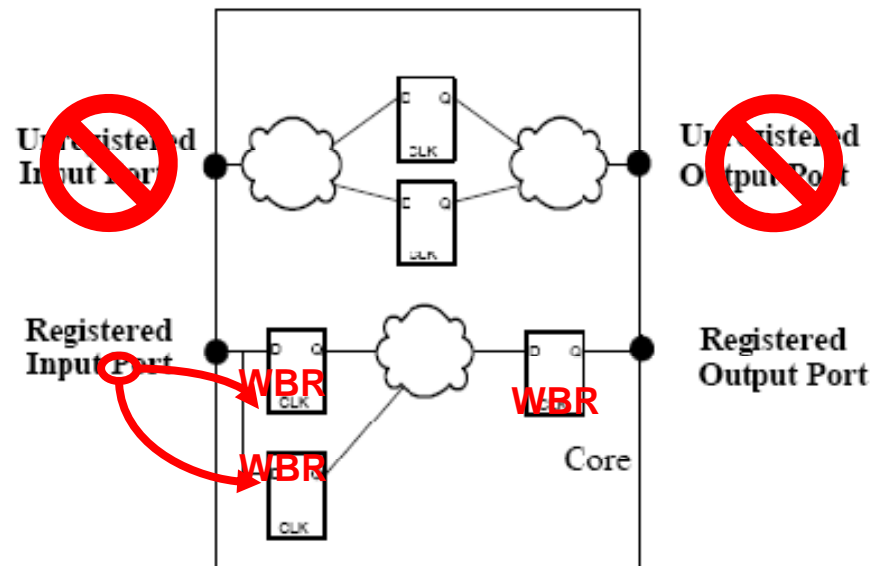
Port Registration



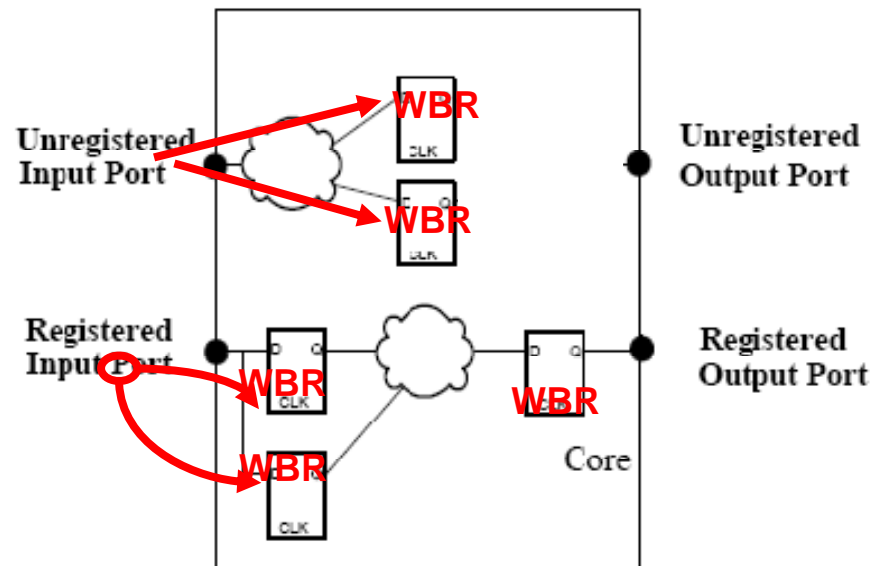
Dedicated Wrapper Cells



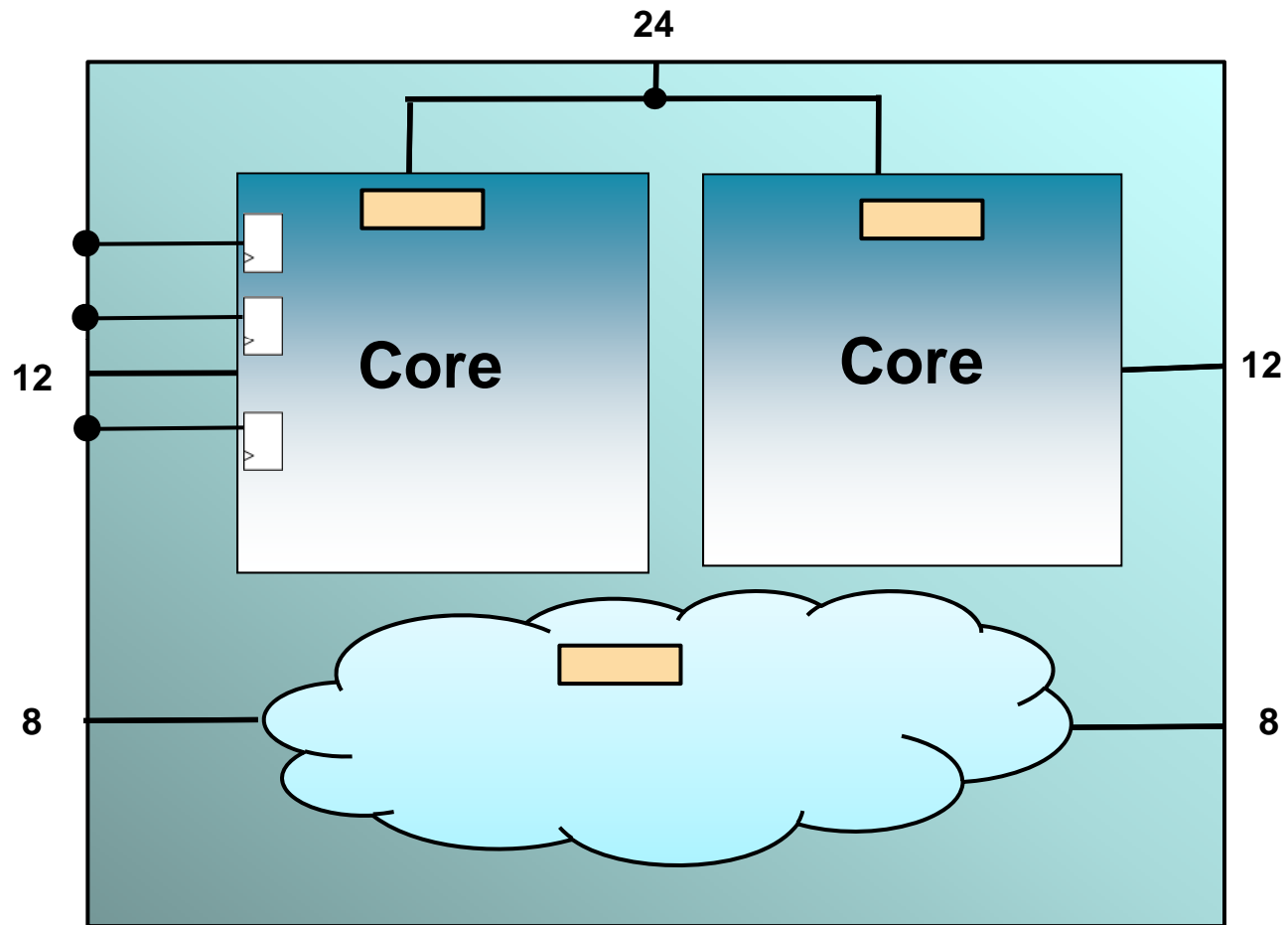
Shared Wrapper Cells



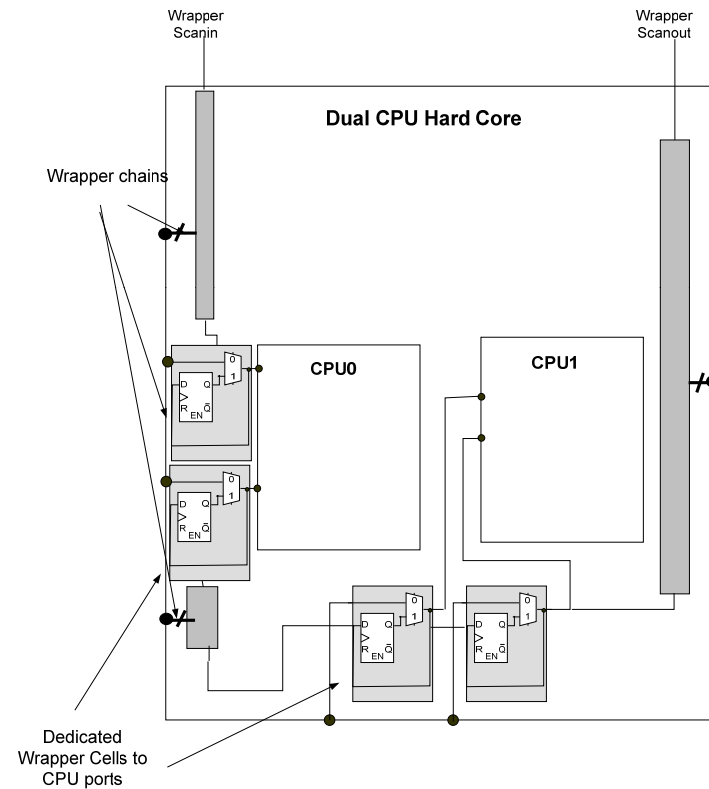
Shared Wrapper Cells



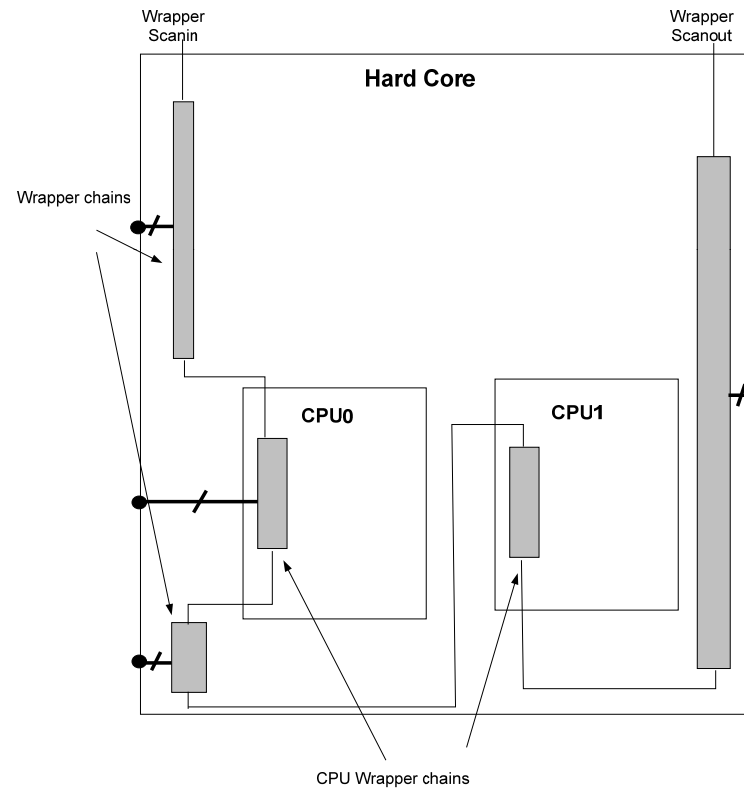
Wrappers



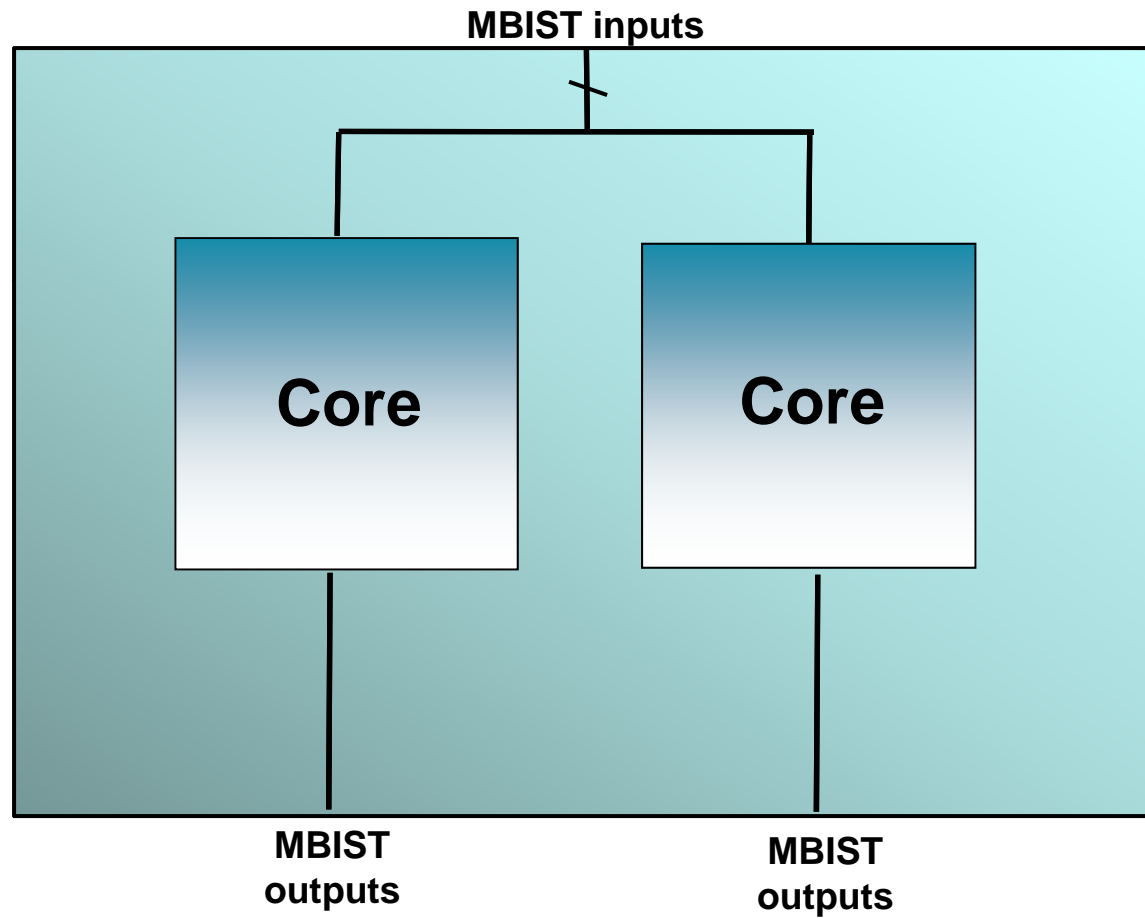
Using Wrapper Cells external to Macro



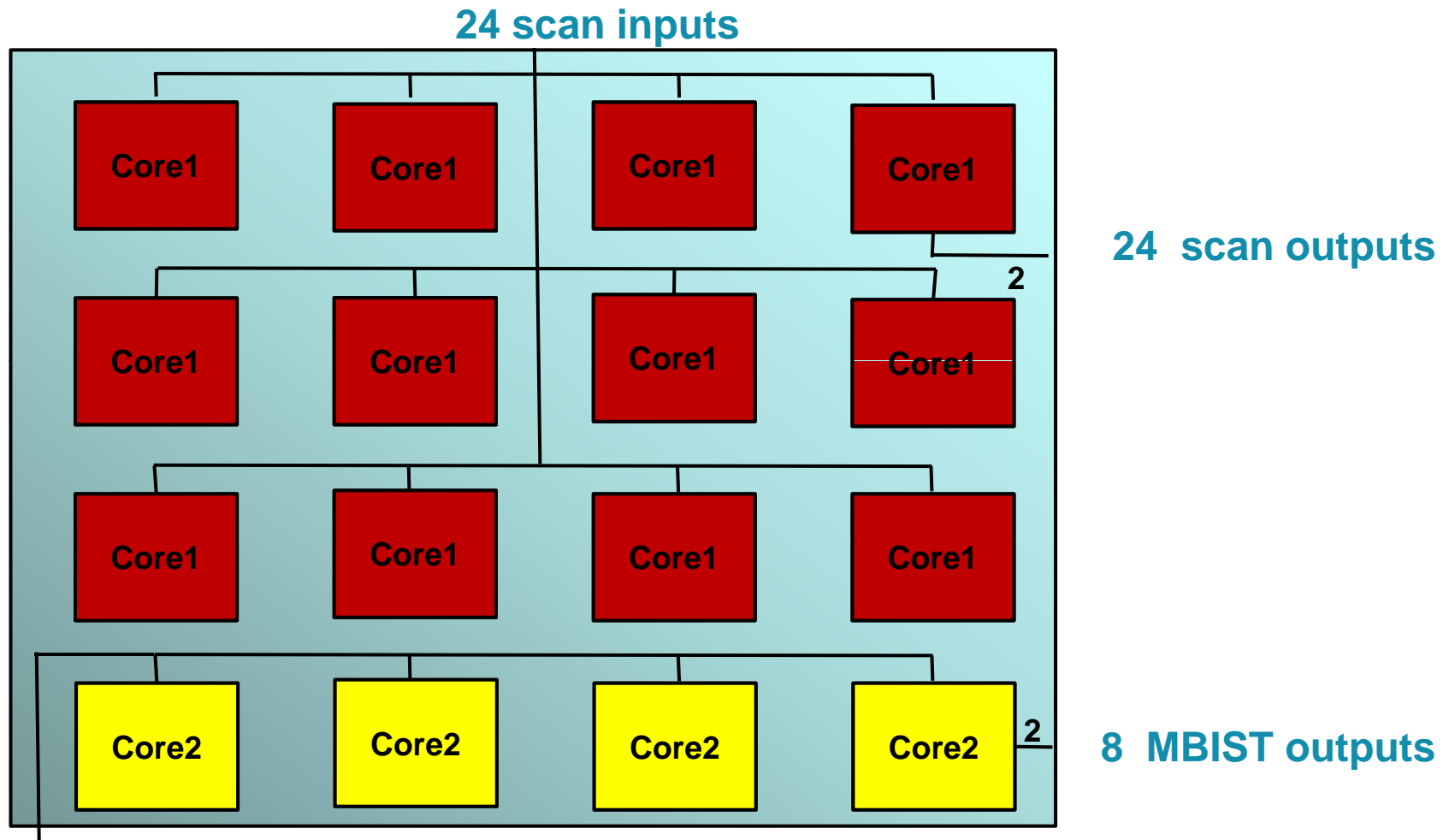
Utilizing Macro Wrapper Cells



MBIST



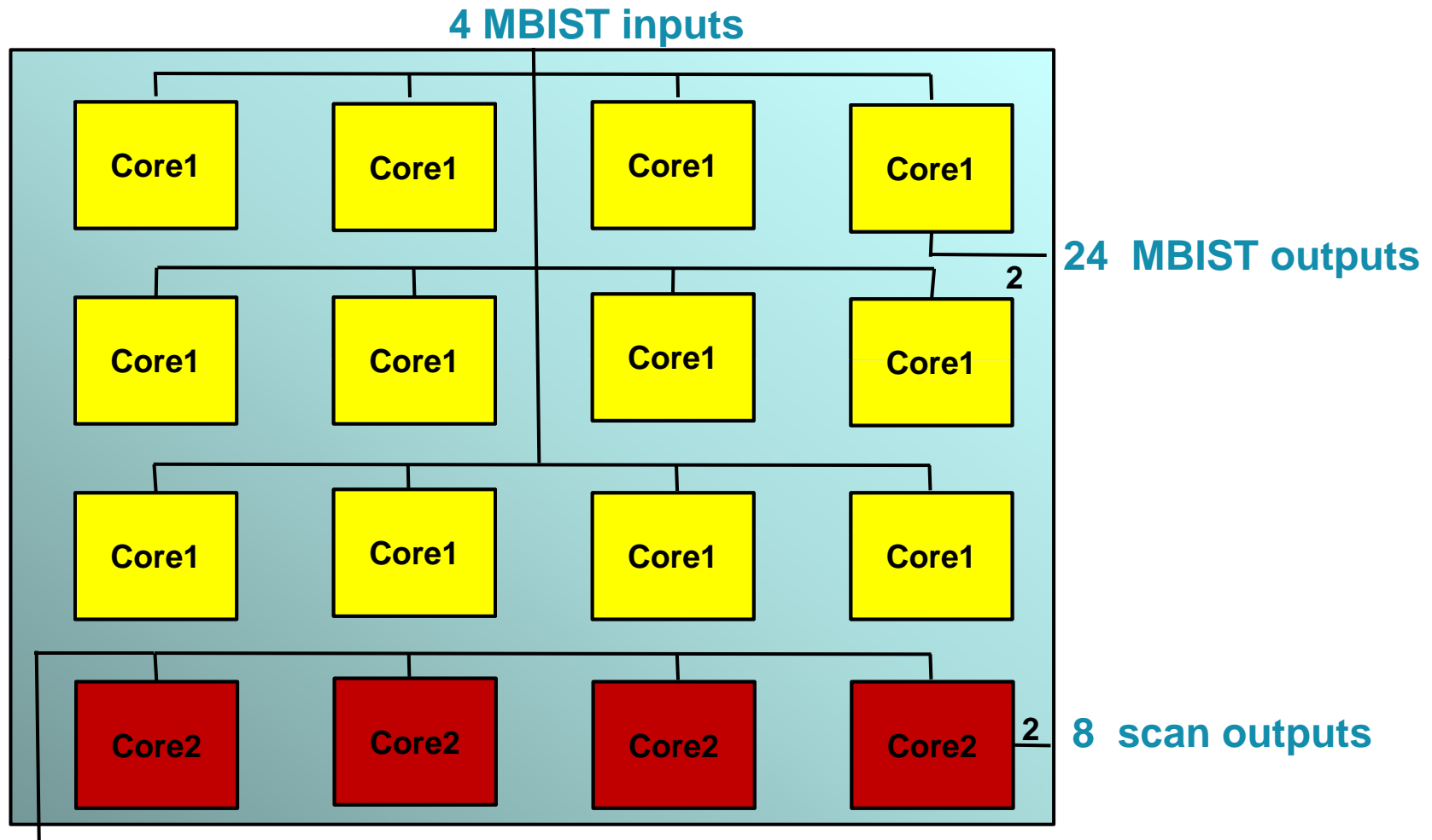
Multiple Cores



4 MBIST inputs
28 inputs

32 outputs

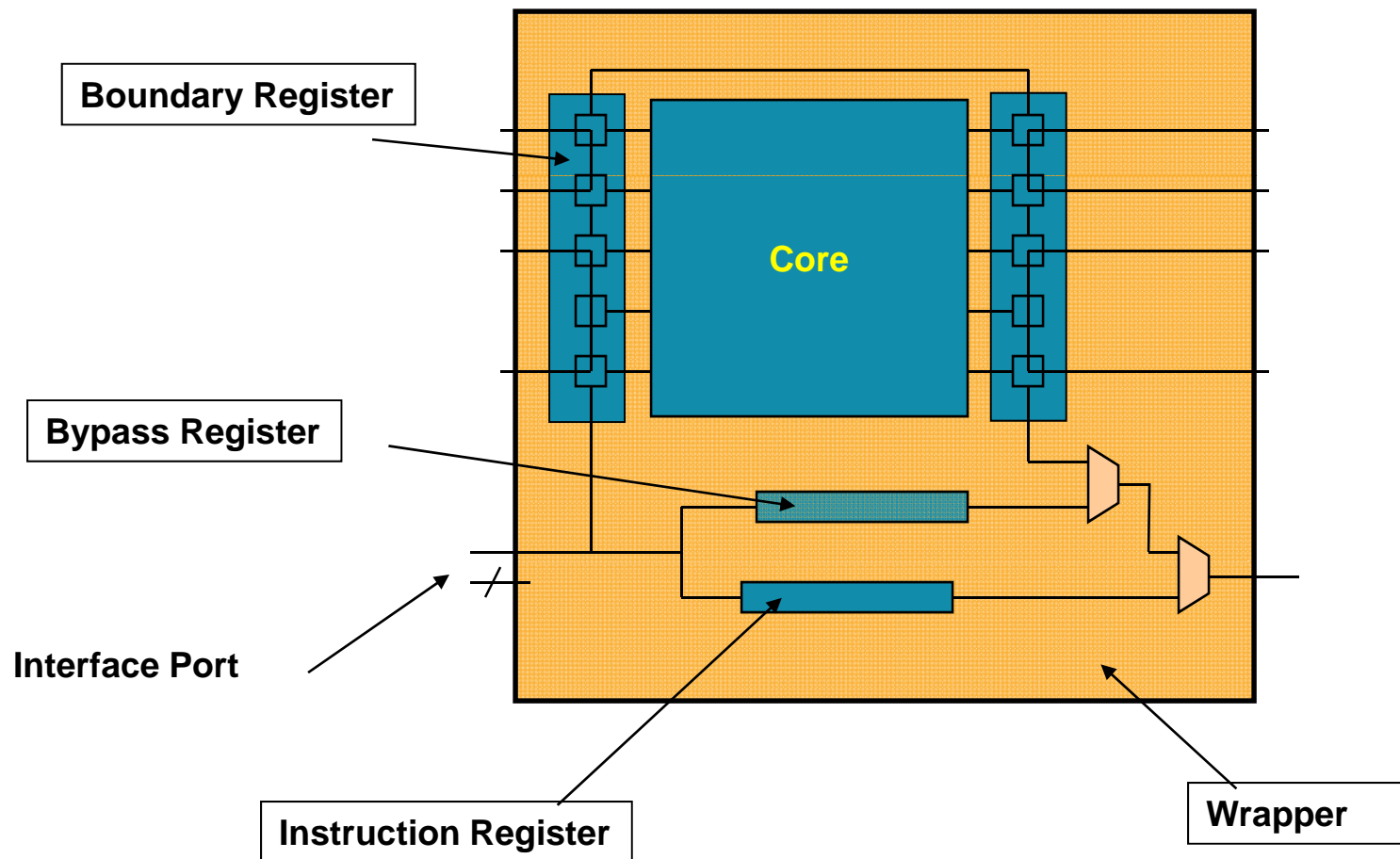
Multiple Cores



24 scan inputs
28 inputs

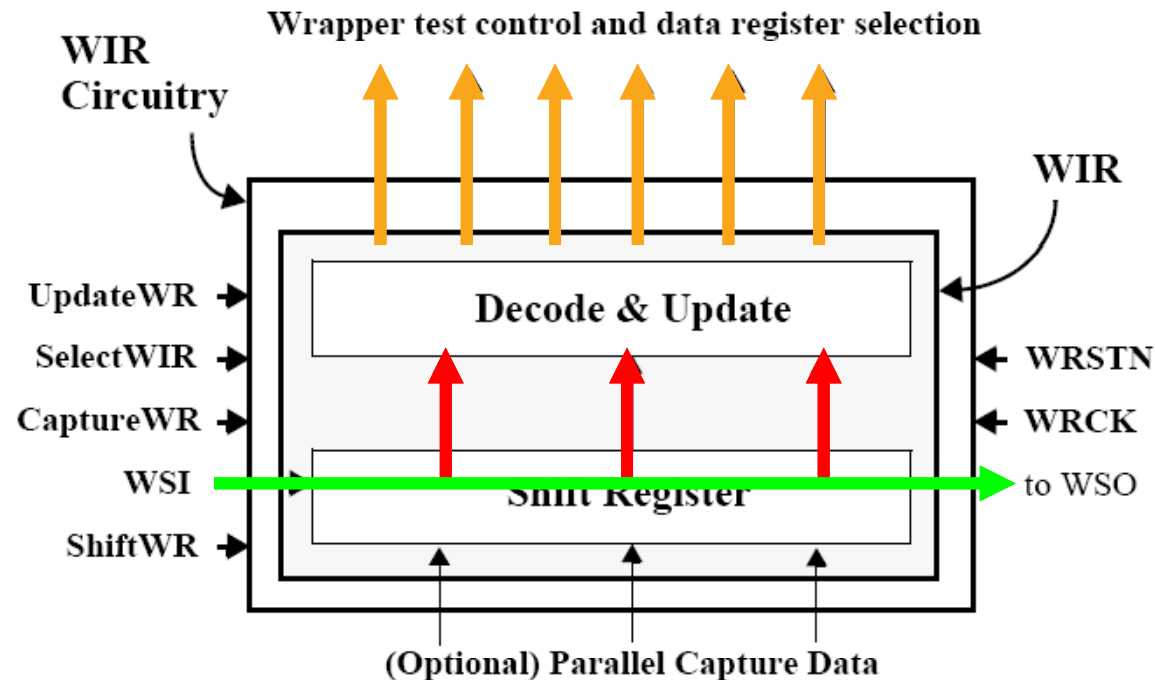
32 outputs

The 1500 Hardware Architecture



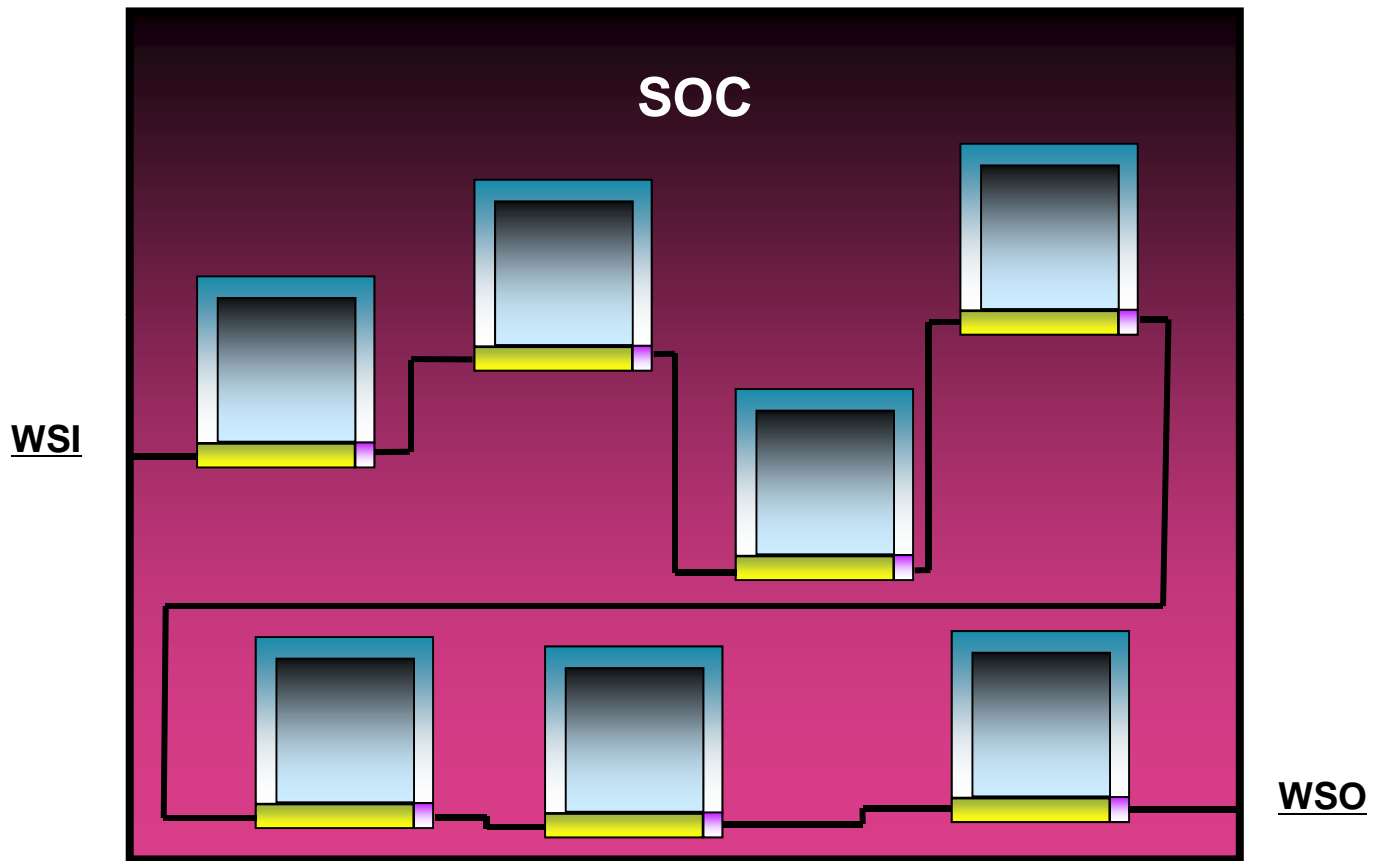
IEEE 1500 WIR Architecture

- Standard mechanism for handling test control



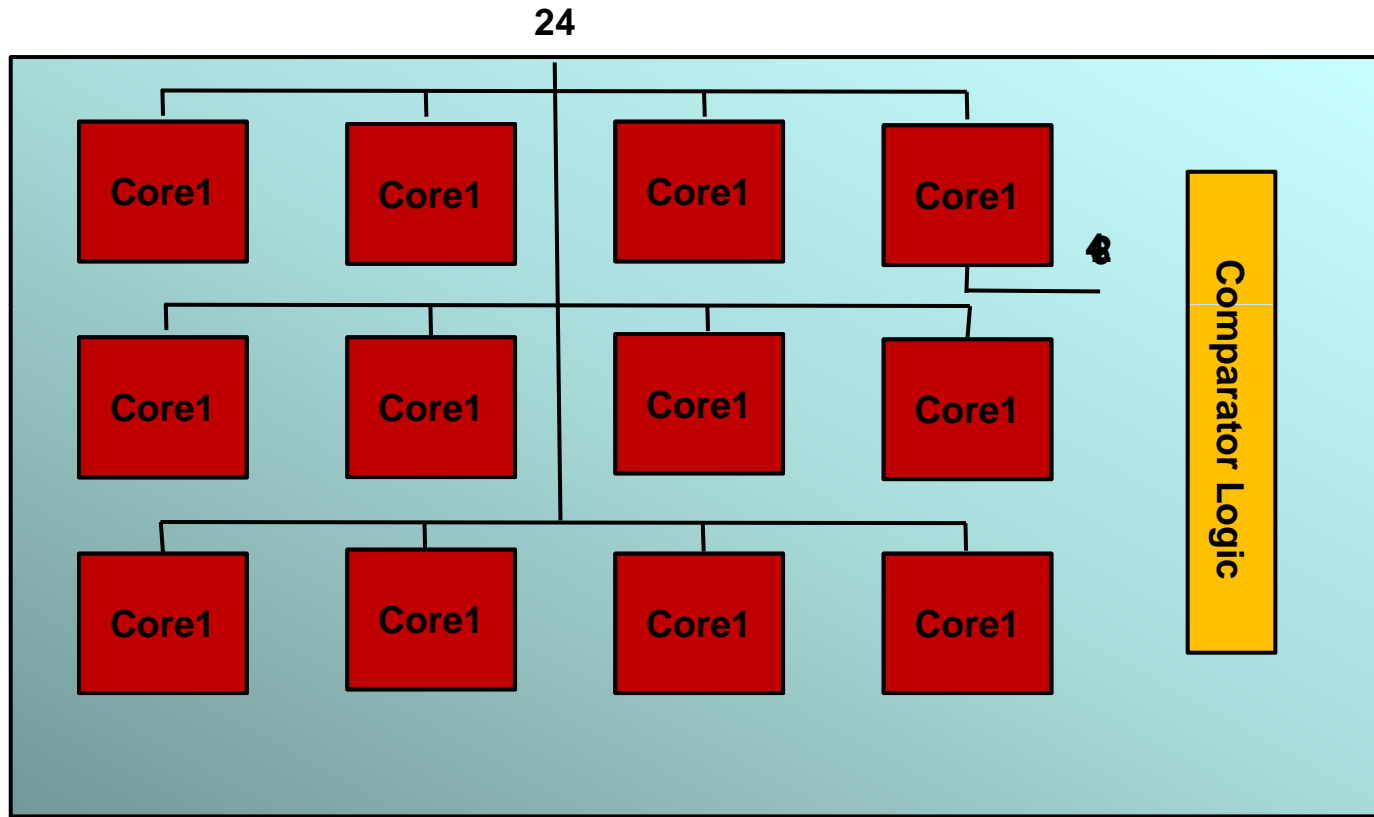
- Primarily meant to configure the WBR
 - Also controls the core embedded in a 1500 wrapper

IEEE 1500 SOC



Multiple Cores

24 inputs



24 outputs

Power Considerations

- How much is too much
 - If all cores can run during functional mode, we should be able to run during test – test mode power must mimic functional mode power or power grids must be enhance.

Key Take away Points

- Understand your design and your tool
- Work with your tool vendor to make sure it meets your needs
- If there is an equation that can describe it, it can be automated.