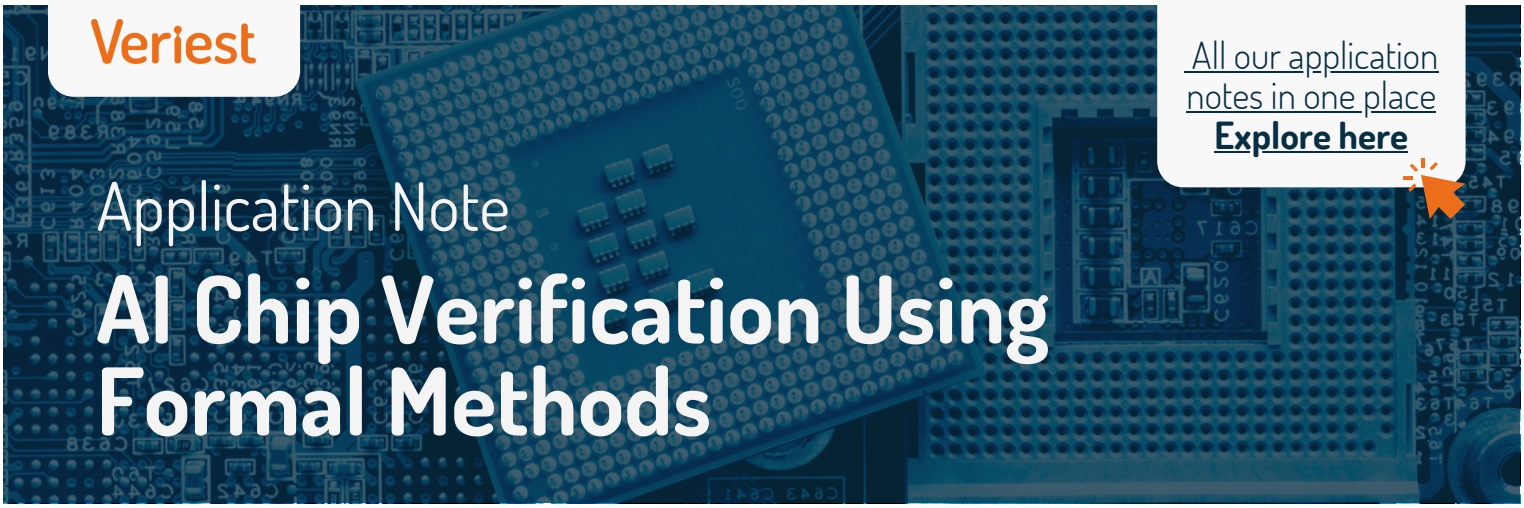
Veriest



All our application notes in one place **Explore here**

Application Note

AI Chip Verification Using Formal Methods

Traditional Challenges

- Number crunching has traditionally been a big challenge for verification
- The Intel FDIV Pentium bug incurred a cost of $475M
- Almost infinite number of input patterns make exhaustive simulation an impossibility
- Unlikely to reach most corner cases
- Traditional Formal Model Checking engines also not well suited for number crunching

Additional Current Challenges

New integer and floating point number formats are being created to exploit hardware bandwidth and power concerns

- Conversions between different formats are error prone and hard to diagnose
- Many legacy designs and verification reference libraries are now obsolete
- Debug also more complex, as new formats are hard to decode with traditional waveform viewers

Veriest Solution

Veriest has implemented a comprehensive Formal Verification strategy:

- Exploit new Formal C2RTL tools designed to deal with datapath (number crunching)
- These tools have very robust C/C++ support, enabling use of new number-format supporting C/C++ models written for simulation to be used almost as-is
- Tools also have features for easy waveform display of custom floating point formats
- Hybrid approach – C2RTL for datapath, Model Checking for random logic
- Work closely with customers to create designs which can easily be parsed into datapath vs. random logic, enabling exploitation of both C2RTL and Model Checking

Results

- Customer experience in getting up and running in days
- Bugs found in previously simulated designs
- Templates created for easily adapting testbench to additional designs