onespin

AI Chips Must Get the Math Right

Accelerate FPU Verification

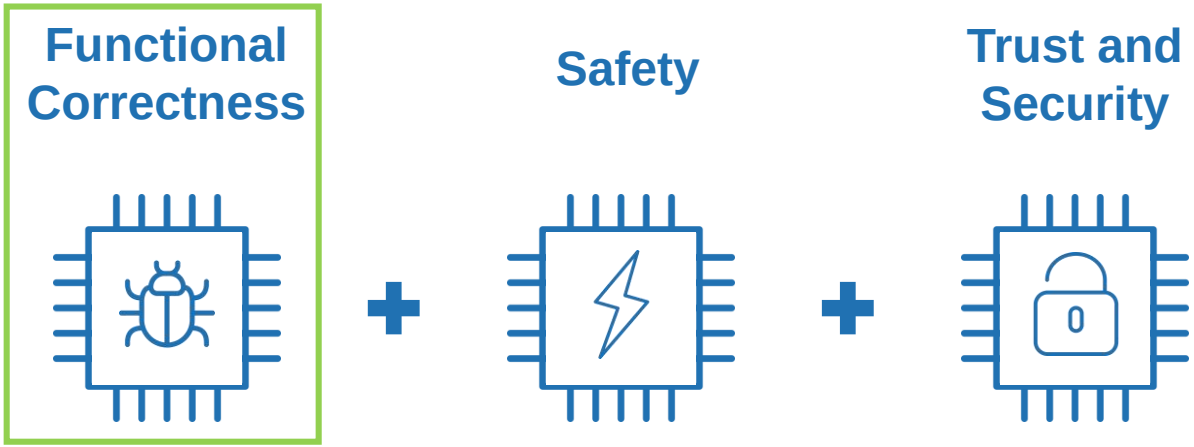
DVClub Europe, April 21, 2020

Verification of AI Designs

sergio.marchese@onespin.com

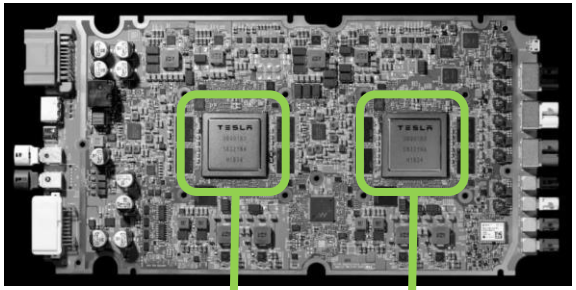
IC Integrity

Functionally correct, safe, secure, and trusted SoCs/ASICs/FPGAs

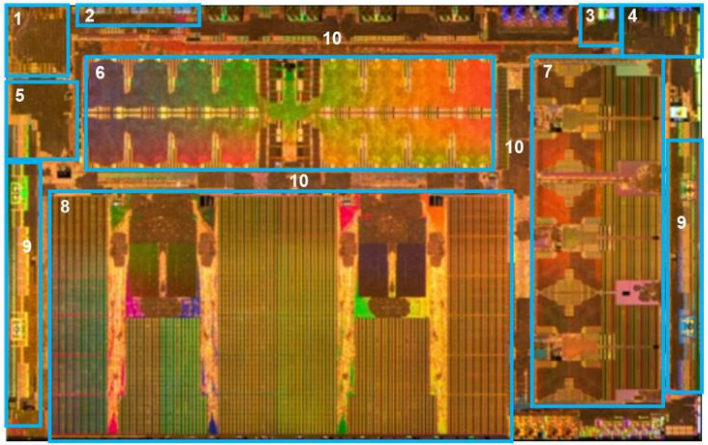


OneSpin provides certified **IC Integrity Verification Solutions** to develop functionally correct, safe, secure, and trusted integrated circuits.

Tesla's Full Self-Driving (FSD) Chip (2019)



Tesla Full Self-Driving (FSD) Chip
14nm FinFET CMOS (Samsung) – 6 Billion Transistors – 260 mm²

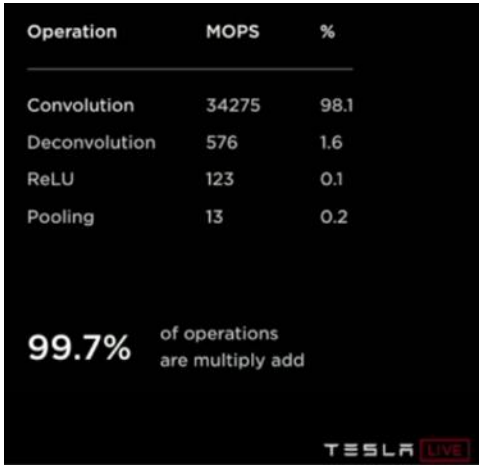


1. **Image Signal Processor (1 G pixel/s)**
2. **Video Input (serial, 2.5 G pixel/s)**
3. **Safety System**
   Dual-core lockstep CPU
4. **Security System**
5. **Video Encoder (H.265)**
6. **GPU (600 GFLOPS, FP32/16, 1 GHz)**
7. **Processor (12 Cortex-A72, 2.2 GHz)**
8. **Neural Network Accelerator (2 GHz)**
   96x96 mul/add array (9216 madd/cycle)
   32 MB SRAM
   36 TOPS
   2 Instances (72 TOPS)
9. **LPDDR4 Memory Controller**
   68 GB/s peak bandwidth
10. **Network-on-Chip**

Tesla FSD Computer (2 FSD Chips)
15W consumed by NNAs
72W total to run autopilot software stack
2300 processed frames per second

Arithmetic Engines and Vision

Image classification and object recognition



Multiply/add loops in convolution layer of a CNN

```
for(r=0; r<R; r++)              //output feature map
  for(q=0; q<Q; q++)            //input feature map
    for(m=0; m<M; m++)          //row in feature map
      for(n=0; n<N; n++)        //column in feature map
        for(k=0; k<K; k++)      //row in convolution kerenel
          for(l=0; l<L; l++) //column in convolution kernel
            Y[r][m][n]+=W[r][q][k][l]*X[q][m+k][n+l];
```

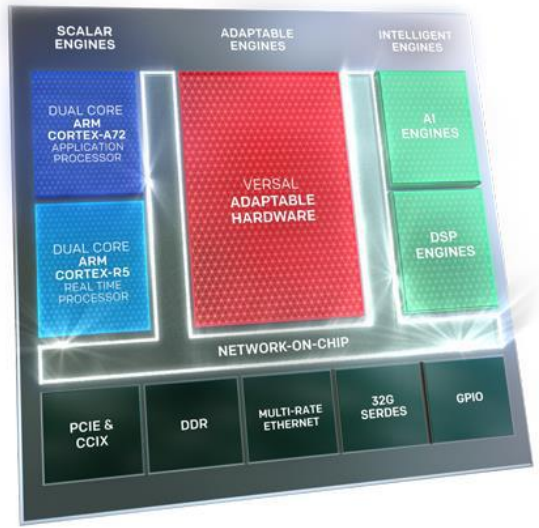
Tesla's NNAs mostly perform multiplications and additions (integer)

Low precision (8/12/32 bit) FP operands often sufficient

Xilinx Versal AI Core

https://www.xilinx.com/products/silicon-devices/acap/versal-ai-core.html

Portfolio's Highest Compute for Maximum AI and Workload Acceleration



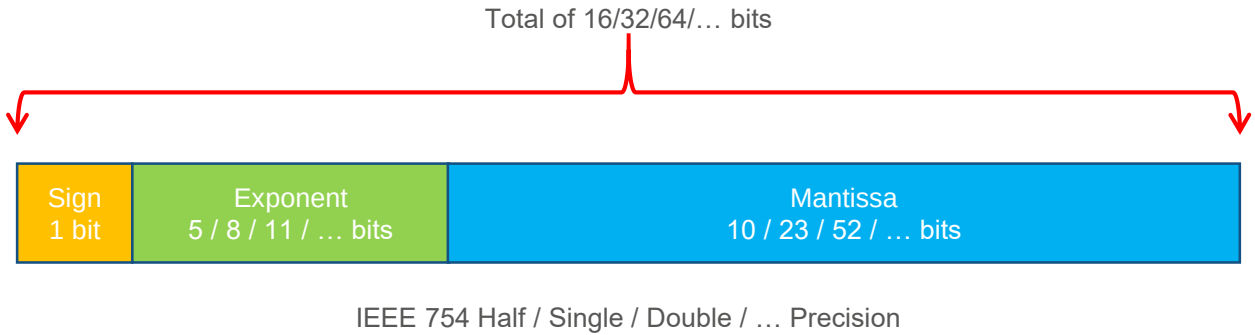
Large array of AI engines with support for floating-point operations

Floating-Point Arithmetic

IEEE-754 Standard

Compared to fixed-point arithmetic

- Covers wider range of values
- No loss of precision, higher accuracy
- More complex hardware
- Notoriously hard to verify



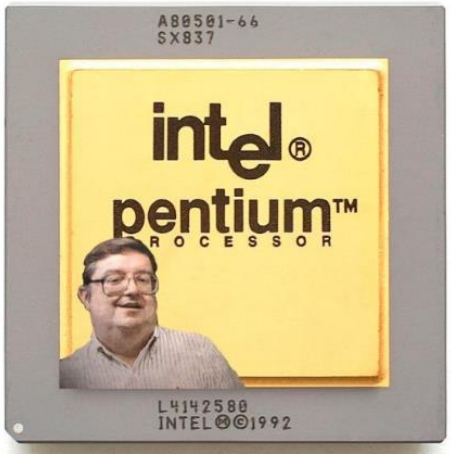
FPU Verification is Key

Arithmetic operations are crucial in AI engines

- Take up a lot of the power and area budget

Even a “tiny” bug can be devastating

- Over many iterations, a tiny bug can lead to a huge error
- Remember the Intel Pentium FPU bug (~$500M cost)



A tiny chance of a huge error

FPU Verification Challenges

IEEE 754 standard is complex

- Arithmetic, comparison, and conversion operations
- Multiple precisions supported (half, single, double, …)
- Five rounding modes, five exception flags
- Special cases (denormals, NaN, ±∞, ±0, ...)

Verification is incomplete, time-consuming

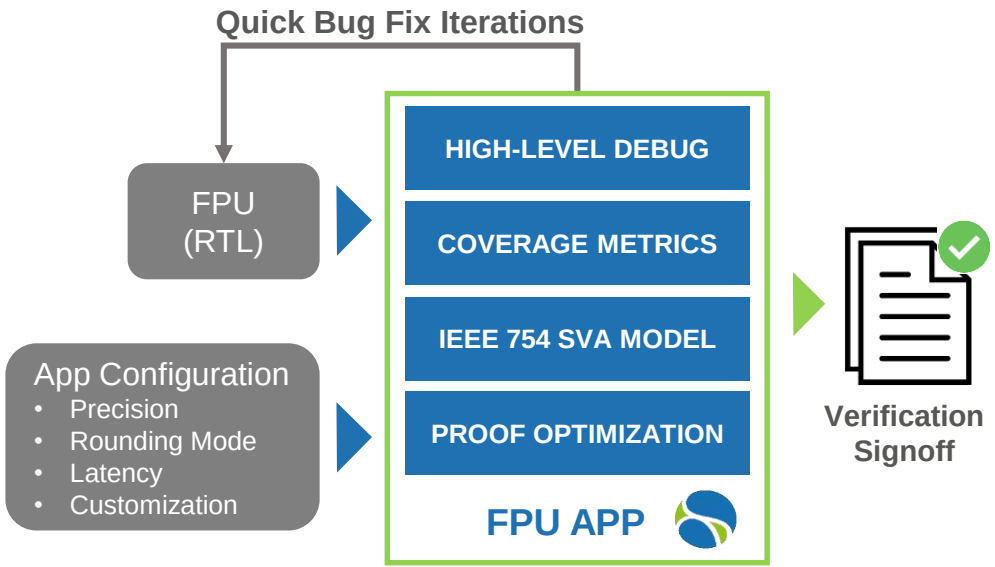
- Expertise and time to setup testbench
- Simulation/emulation unlikely to hit corner-cases

- Sequential Equivalence Checking (SEC) RTL vs. C++/SystemC reference model
- Requires additional model and knowledge of implementation details

- Expertise and time to write the right assertions
- Inconclusive proofs

OneSpin FPU Verification App

Accelerate verification, prove correctness



- Easy to setup
- Supports half/single/double bfloat16 and custom precisions
- Supports all 5 rounding modes and 5 exceptions flags
- add, sub, mul, fma, abs, neg
- Conversion and comparison operations
- Parameters to specify ambiguities in the standard
- RISC-V configuration
- No need for C++ model of the FPU
- Easy to model intended deviations from the IEEE-754 standard

FPU App Use Cases

AI arithmetic engines, RISC-V and proprietary ISA cores

Successful applications include

- AI Platform Chip
- Automotive core
- Open-source: multiple RISC-V cores
- Open-source: multiple FPU IPs

Bugs found in all applications

Days of effort compared to months with simulation

Xilinx FPU Verification

Formal is now key component of the flow

Design

- Compliant with IEEE-754
- Few minor intended deviations
- Single-precision
- fadd/fsub/fmul operations

Results

- All known bugs detected
- 1 new issue identified
- 5X reduction in verification effort …
- … even when accounting for tool and app ramp-up

Project

- IP already verified with simulation and emulation
- FPU app tested on RTL versions with known bugs

Operation	# bugs	Runtime	Result
FADD	0	3 minutes	Full proof
FSUB	0	1 minute	Full proof
FMUL	1	4 minutes	Full proof

Summary

FP operations crucial in AI and other applications

OneSpin FPU App

- Reduces verification effort from months to days
- Includes SVAs capturing IEEE-754 standard and coverage
- Detects bugs in seconds, supports efficient debug
- Optimal proof strategies and engines
- Proven on many proprietary and open-source designs

Formal verification of FPU datapath during RTL development!

For more information visit **onespin.com/fpu**

*"Results of the application of the FP ABIP as part of the OneSpin FPU App in industrial applications show that corner-case bugs can be unveiled within seconds, and unbounded proof achieved within minutes, even for the multiplication operation. These results were obtained **without** the use of **abstractions or assume-guarantee partitioning**."*

Ravi Ram, principal engineer, verification architecture, Xilinx

Thank you!