

# Why Now? Growth and Innovation using HLS and the Need for Standardization

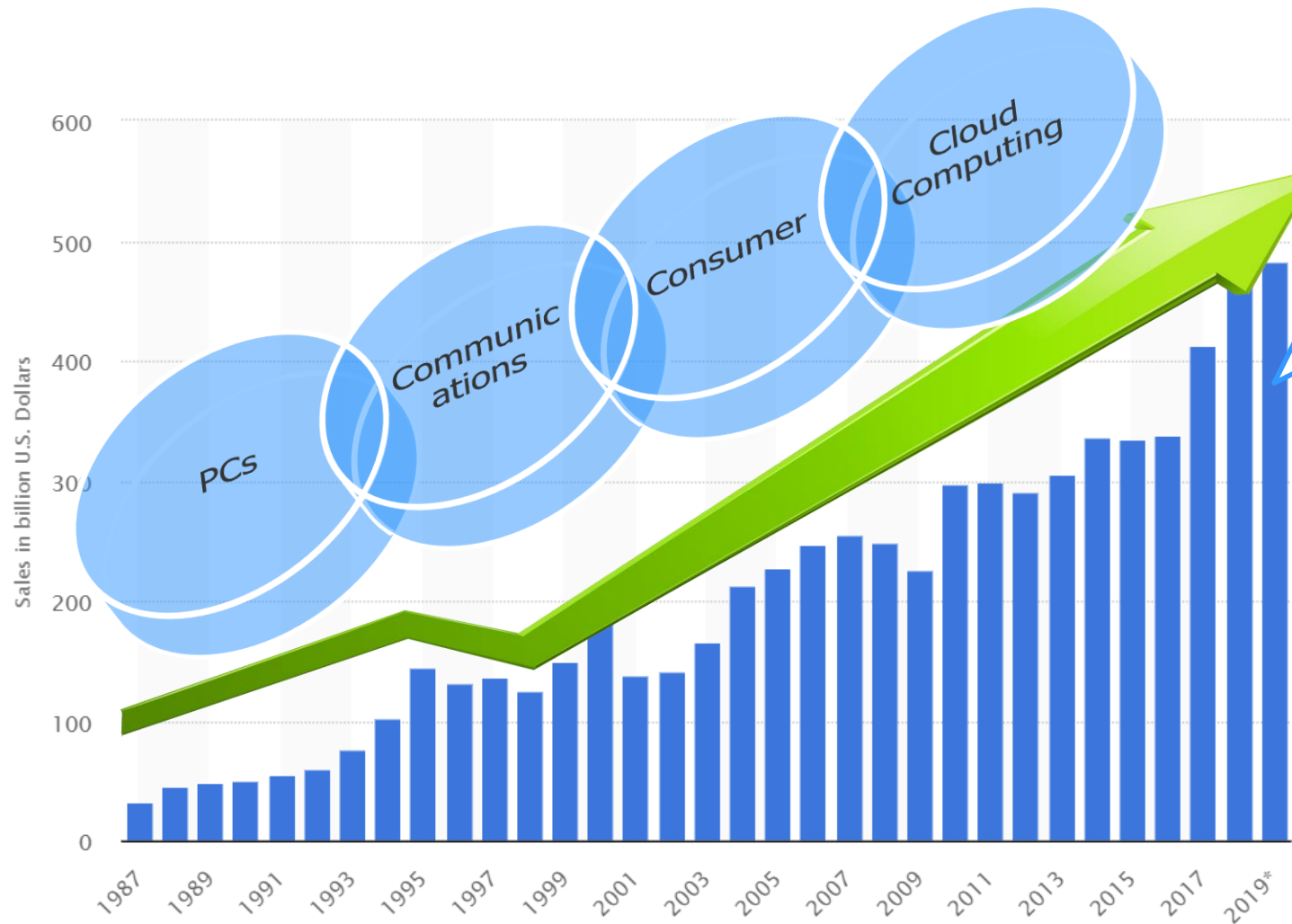
Ellie Burns

Director of Marketing

Digital Design Implementation Solution



# New Markets fueling Semiconductor Growth



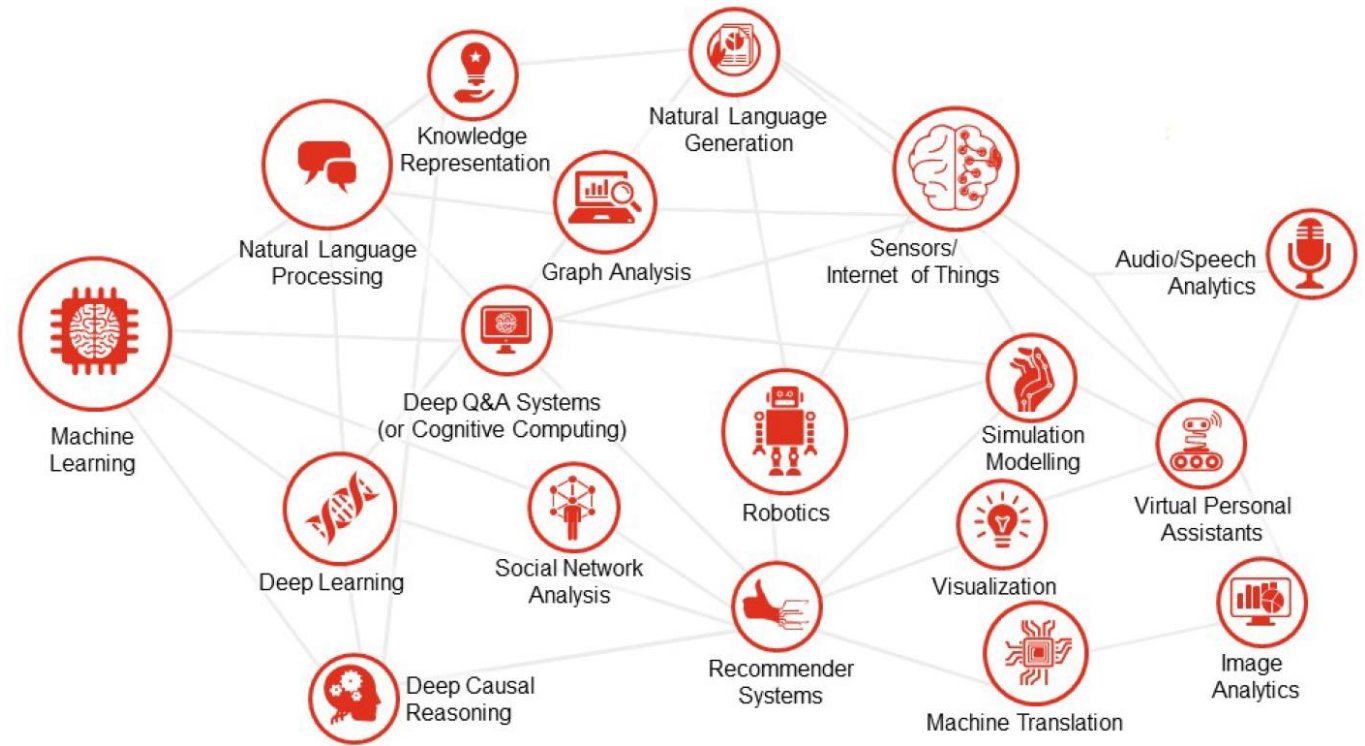
## Next 5-10 years

- AI, ML, Compute Vision
- Communication, 5G
- Virtual Reality
- Internet of Things
- Autonomous vehicles
- Systems/Software companies designing hardware
- Mil-Aero moving to ASIC

Source: [www.statista.com](http://www.statista.com)

# Impact of AI is in all domains

- Cloud based deep learning
- ADAS
- Smart Cameras
- Financial Technologies
- Healthcare and Medicine
- Manufacturing
- Robotics
- Voice Assistants



# Artificial Intelligence & Machine Learning Brings Opportunity



“... artificial intelligence will likely be the catalyst that will drive another **decade-long growth cycle** for the semiconductor sector...

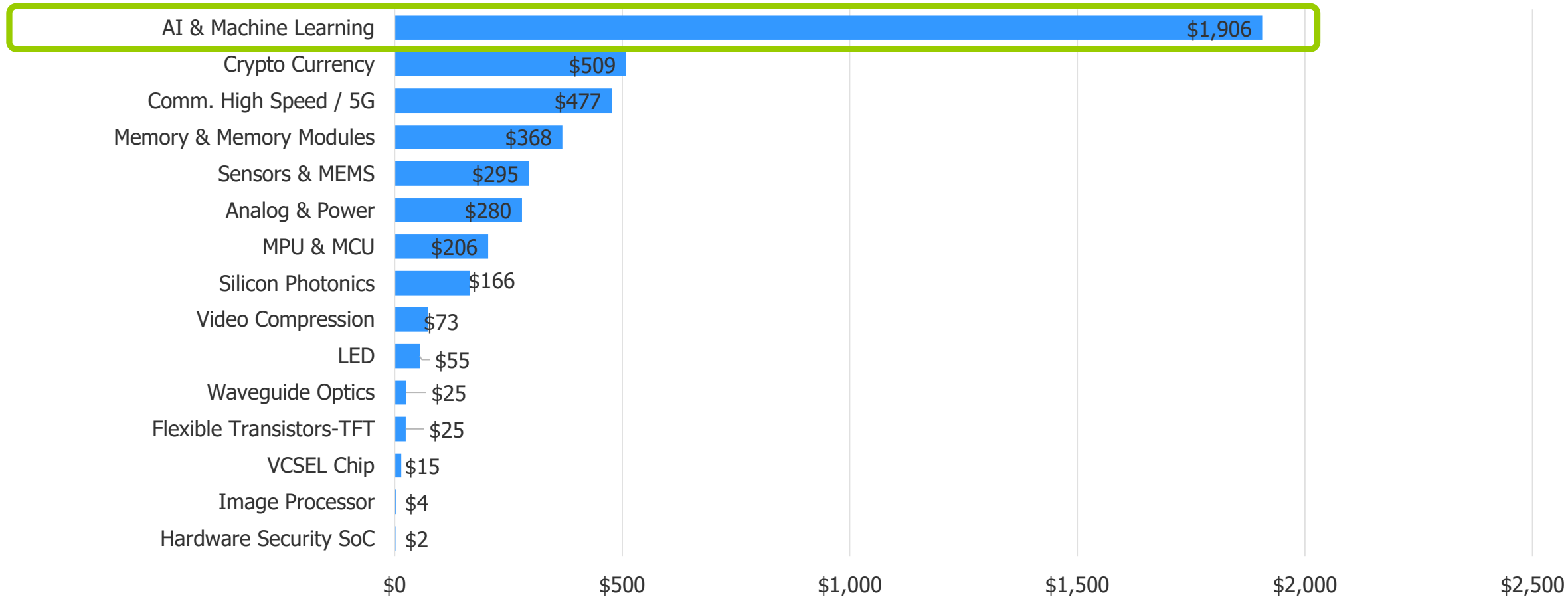
The semiconductor firms that are able to take the most advantage of this growth and fully realize their market potential will likely be those that **harness the possibilities that AI brings.**”

Source: PWC, Opportunities for the Global Semiconductor Market, April 3, 2019

# Startups Dominated by Domain Specific Architectures

## Worldwide Fabless Company Venture Capital Funding (Rounds 1-3)

Market Segments Funded 2012 – 2019 YTD  
By Funding Dollars (\$M)



# Multiple Markets are Moving to HLS

## *The Perfect Storm*

Have Significant New Algorithmic Design

Need Faster Time to Market with Good QofR

Need to Reduce Verification Cost & Debug Time

Need Lowest Power for Critical Performance

Need to Handle Frequently Changing Specifications

Computer  
Vision and  
Neural  
Computing



High  
Bandwidth  
and Cellular  
Comm



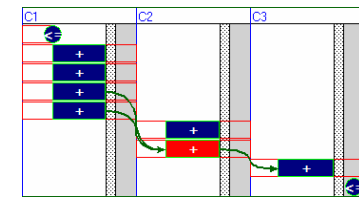
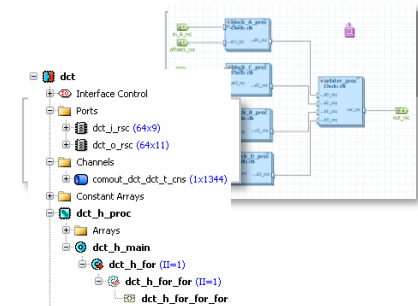
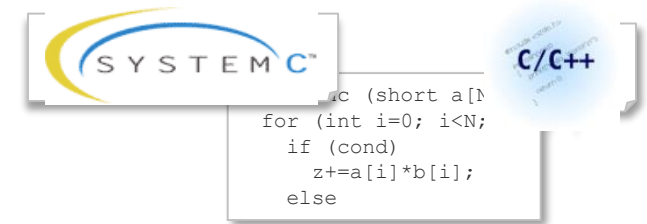
Image  
Processing,  
Video &  
Compression



HLS is the Only Way to Address Multiple Design and Verification Challenges

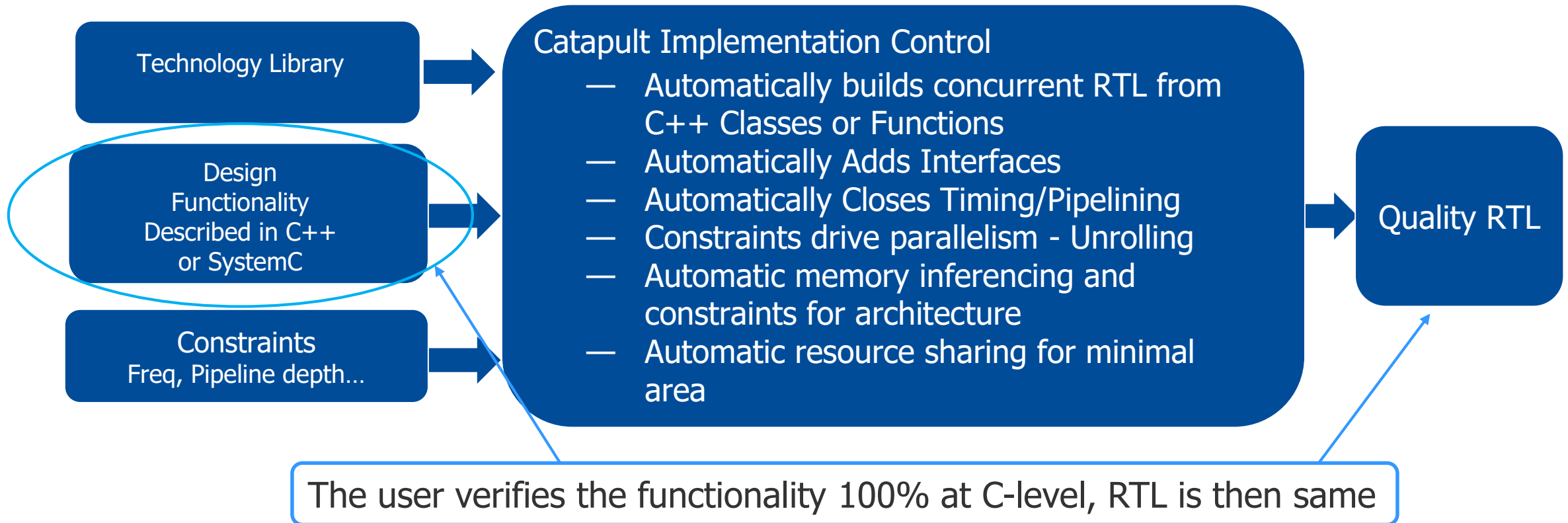
# Benefits and Expectations of HLS

- Deliver high quality RTL with QoR comparable to hand-coded RTL...in much less time
  - 1 Year reduced to a few months
  - New features added in days not weeks
- Cut verification costs with 500x speedup vs RTL
  - Faster/easier functional verification and debug
- Enable late functional changes without impacting schedule
  - Algorithms can be easily modified and RTL regenerated
  - New technology nodes are easy (or FPGA to ASIC)
- Quickly evaluate power and performance of algorithms
  - Rapidly explore multiple options for optimal solution



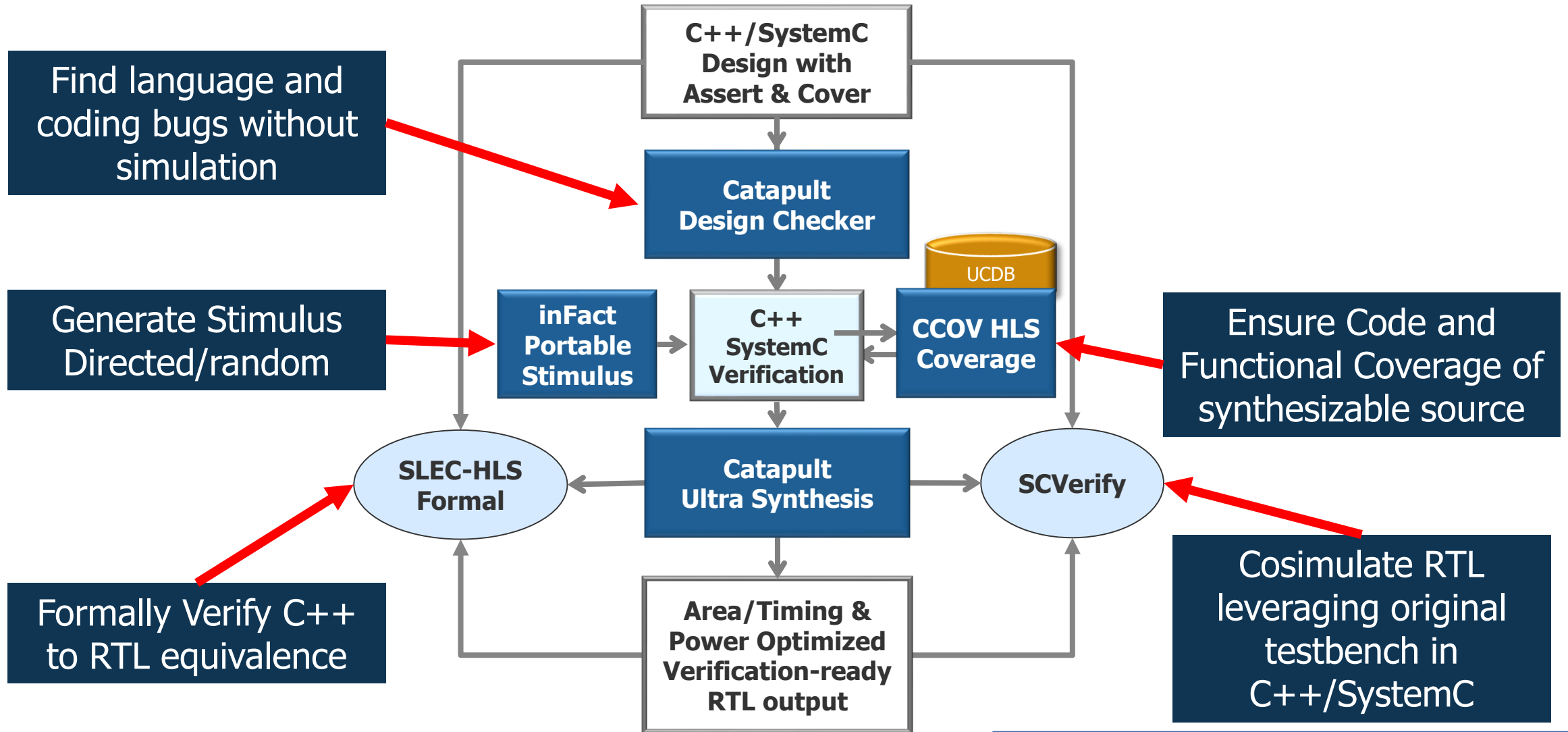
# Why HLS is So Much More Productive than RTL

- Catapult HLS Separates Functionality from Implementation and has powerful capabilities to control implementation for best QofR





# Adding Verification Tools to the C++ / SystemC HLS EcoSystem



# Common misconceptions & wrong expectations for HLS

- HLS does NOT “translate” any working C++ code into a good HW
  - You still need to design HW in C++ (no magic!)
  - Pure software models, overly abstracted, results in sub-optimal HW
- HLS does NOT turn a SW engineer into a HW designer
  - You must have HW skills to understand what HLS does, drive it and get better QofR
- HLS does NOT replace RTL designers, it empowers them
  - They need to change approach, it’s more a cultural than technical barrier

The good news is that ALL the RTL designers that use HLS,  
“*..after a few months never want to go back!*” (ST’s G.Trunde, DAC 2015)

# Catapult HLS Proving Value in Production Designs Today and Growing Rapidly



HLS enables availability of HW accelerators on day 1 after AV1 spec freeze

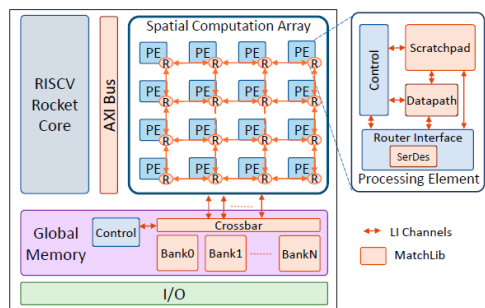
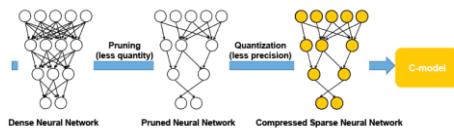
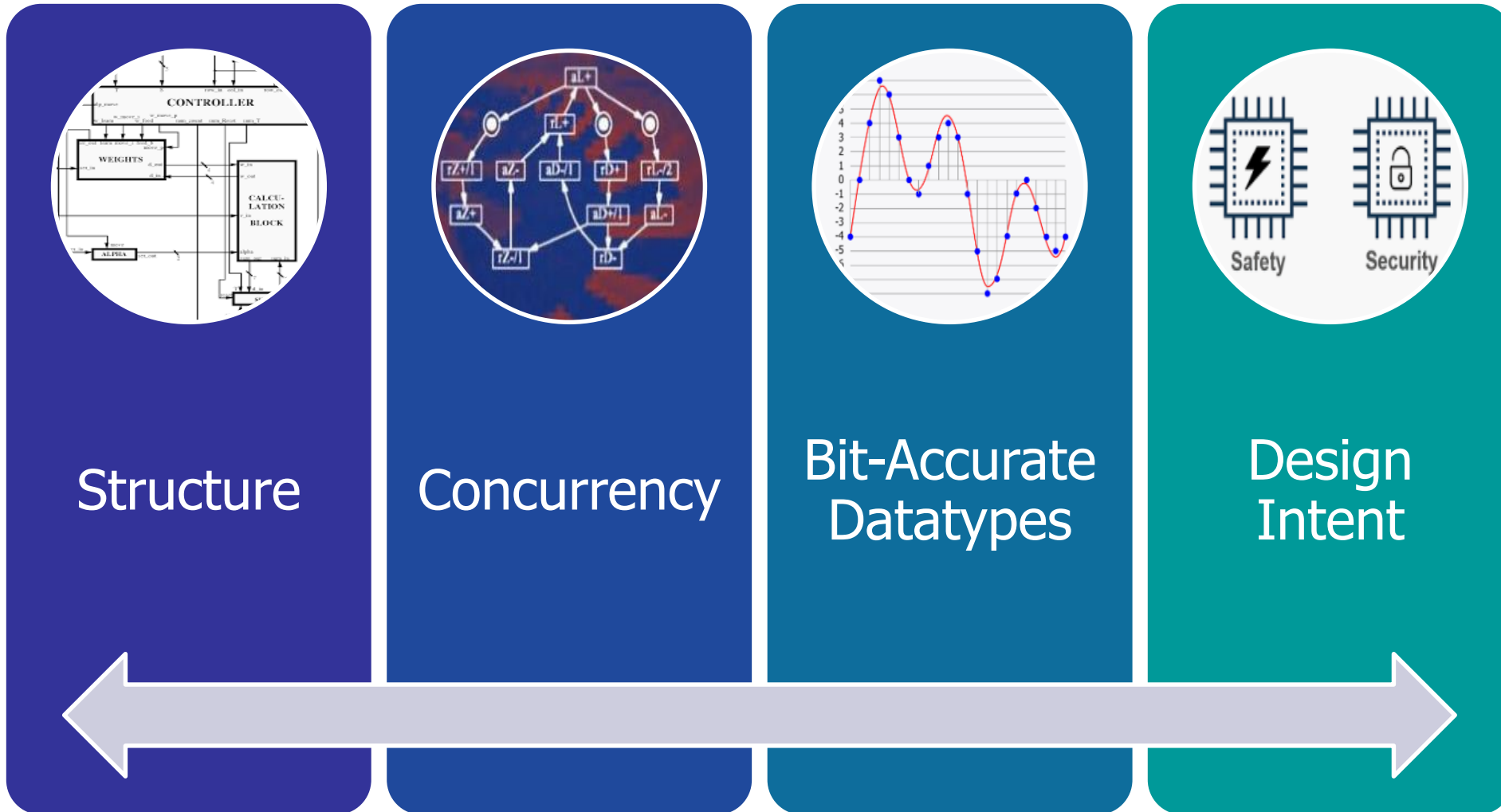


Figure 5: Prototype SoC

- **NVIDIA** – Video - Cut design time by 50%, verification cost by 80%
- **NVIDIA Research** – AI – Taped out 2 AI accelerator SoC's built completely in HLS – “~10x RTL design and verification effort reduction compared to manual RTL”
- **Facebook** – AR/VR- New team that just could not get RTL done on new Algorithms on time
- **Bosch** - CV/Image Processing- Delivered new design IP with Improved quality and ahead of schedule of 7 months
- **ST** - Image Processing - 50+ IP's with HLS ; Reduced average time for IP from 24 weeks to 4 weeks with better quality
- **Google** – Video - Time to Verified RTL: 2x faster ; Simulation Speed: 500x faster. >99% bugs caught in C simulation
- **Leading 5G** - 5G - 4X faster than hand-coded RTL, Much easier and faster implementation of fast changing 5G specs
- **Chips&Media** - CV/AI – IP for real-time object detection using DNN; 5 months to 2.5 months on first project; fast time to FPGA demonstrator to win business
- **FotoNation** – CV/AI - Facial Recognition used in ADAS; ML HW IP 3 weeks from Caffe to FPGA; 4X Faster than RTL for best QofR
- **Qualcomm** – Video/Image Processing - 1.5X-2X faster time to development using HLS

**STANDARDIZATION IS  
NEEDED TO ENABLE HLS AND  
LOWER COST OF ECOSYSTEM  
DEVELOPMENT**

# Areas where Standardization is Needed for HLS



# HLS Standards Today

## SystemC - IEEE 1666-2005, 1666-2011

- SystemC - IEEE 1666-2005, 1666-2011
  - SystemC language contributes hardware constructs to express structure and concurrency based on events to the C++ language
  - Synthesis standard produced for SystemC in 2016 and continues to update
  - Synthesis standard mostly focuses on SystemC but does cover C++ syntax and built-in datatypes that are synthesizable
- However multiple HLS tools use plain C++
  - Infer structure from C++ Class Hierarchy or functions
  - Models Kahn Process Network (KPN) style concurrency with light weight modeling of processes communicating with channels
  - Preferred by many users due to simplicity
- SystemC and C++ modeling styles are complementary
  - Could model subsystems with C++ and integrate into larger SystemC design

# Most Immediate Need for IEEE standardization

## Bit-Accurate Datatypes in C++

- Orthogonal and easy to decouple from complexities of structure and concurrency
- More immediate benefit to HLS community before a more general IEEE synthesis standard covering all aspects of hardware modeling could be approved
- Precedence of value of datatypes in IEEE
  - The IEEE 754 (1985, 2008, 2019) standards for floating-point
  - The IEEE 1164-1993 standard which addressed Multi-value Logic for VHDL
- Focus on C++ bit-accurate integer, fixed-point, floating-point and complex datatypes

# Value of Focus on C++ Bit-Accurate Datatypes

- Floating point is available
  - Increasing interest for HLS in signal processing algorithms, machine learning, augmented reality, etc.
  - IEEE compliant floating-point, fully parametrized to extend IEEE semantics also Google's TensorFlow "brain float"
- Yes, there is are SystemC bit-accurate types, however
  - They present some serious deficiencies not addressable without affecting legacy
  - There is no floating point or complex types
- C++ types are much higher performance in verification/simulation
  - 100X faster than SystemC equivalents
- Proven in production since 2006 (floating point added in 2018)



# C++ Bit-Accurate Datatypes Contributed to Open-Source in 2016 - *HLS Libs on Github*

## ■ HLS Libs **Open-Source**

Github Repository <http://hlslibs.org>

- Deployment examples
- Apache license
- Options for high-perf or high-accuracy

## ■ AC Datatypes – Now

- **Including New floating point– Very good for AI**

## ■ AC Math - Now

- Basic math/trig functions
- Matrix/Linear Algebra class/funcs

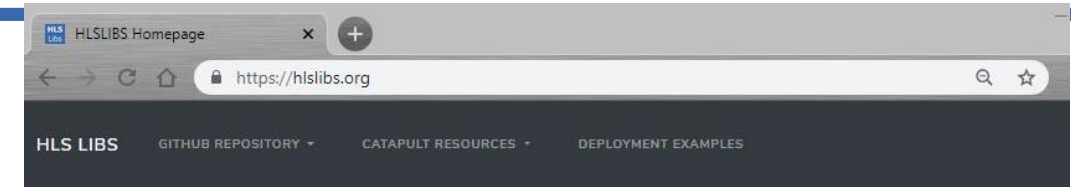
## ■ AC DSP - Now

- 1-D Filter blocks
- Various FFT architectures

## ■ AC Image Processing Library” (AC IPL) – Coming soon

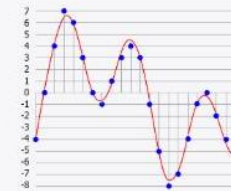
- Image scaling
- Building blocks for compression, line-buffers/windowing classes

## ■ Matchlib from NVIDIA – Available NOW – coming to hlslibs soon



### WELCOME TO HLSLIBS!

HLSLibs is a free and open set of libraries implemented in standard C++ for bit-accurate hardware and software design. The goal of HLSLibs is to create an open community for exchange of knowledge and IP for HLS (High-Level Synthesis) that can be used to accelerate both research and design. The libraries are targeted to enable a faster path to hardware acceleration by providing easy-to-understand, high-quality fundamental building blocks that can be synthesized into both FPGA and ASIC. HLSLibs are delivered as an open-source project on GitHub under the Apache 2.0 license and contributions are welcome.



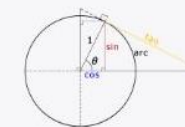
### AC DATATYPES

The Algorithmic C datatypes include a numerical set of datatypes and an interface datatype for modeling channels in communicating processes in C++. The numerical datatypes provide an easy way to model static bit-precision with minimal runtime overhead. They include bit-accurate integer, fixed-point, floating-point and complex datatypes. The numerical datatypes were developed in order to provide a basis for writing bit-accurate algorithms to be synthesized into hardware.

[LEARN MORE](#)

[DOCUMENTATION](#)

[BROWSE GITHUB](#)



### AC MATH

The Algorithmic C Math Library contains synthesizable C++ functions commonly used in Digital Signal Processing applications. The functions use the Algorithmic C data types and are meant to serve as examples on how to write parameterized models and to facilitate migrating an algorithm from using floating-point to fixed-point arithmetic where the math functions either need to be computed dynamically or via lookup tables or piecewise linear approximations. The library includes basic math functions (reciprocal, log, exponent, square-root, sin/cos/tan, etc) as well as a matrix storage class and linear algebra functions like multiplication, determinant, Cholesky Inverse/Decomposition, etc. Each function comes with a unit test to demonstrate usage and measurement of errors due to approximations.

# Summary

---

- Markets that are driving the growth of semiconductor market today are the ones where HLS is a good fit
- HLS delivers QofR and quality RTL in production today and proven to cut project time in half or more
- IEEE Standardization is needed to drive the HLS ecosystem to the next level
  - Recommended first and easier step are the datatypes
- Next steps to start?

# Mentor<sup>®</sup>

A Siemens Business

[www.mentor.com](http://www.mentor.com)