CALYPTO®

Design · Optimize · Verify

State-of-the-art of HLS and What is Ahead

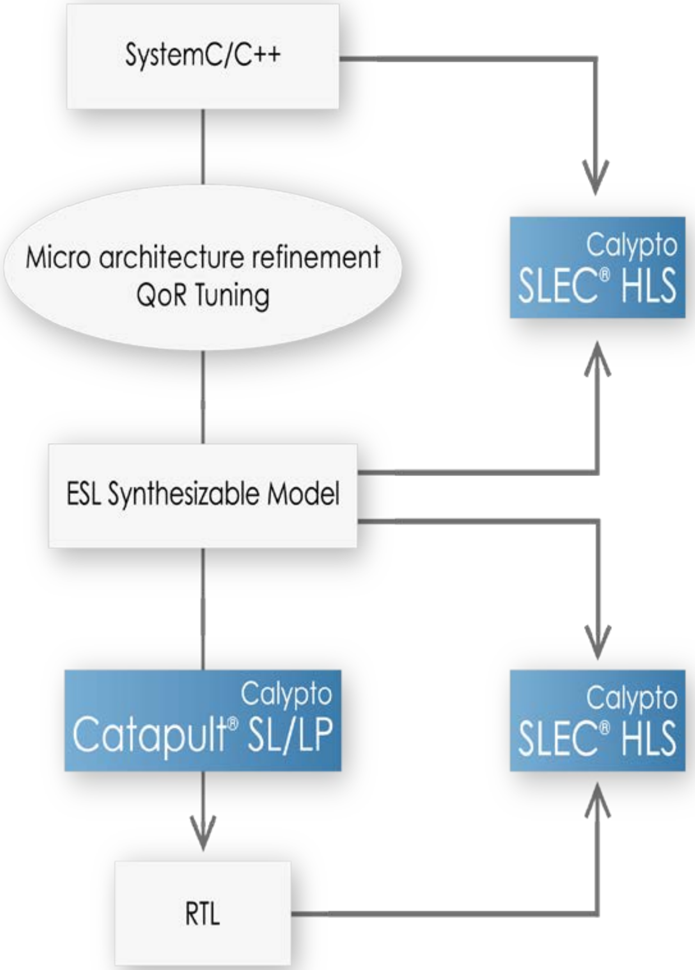
Andres Takach
Senior Architect

Overview

- High-level Synthesis and Verification at Calypto
- Adoption and Deployment
- User’s expectations
- Verification: Coverage and assertions
- Standards

HLS Synthesis and Verification

Catapult Platform

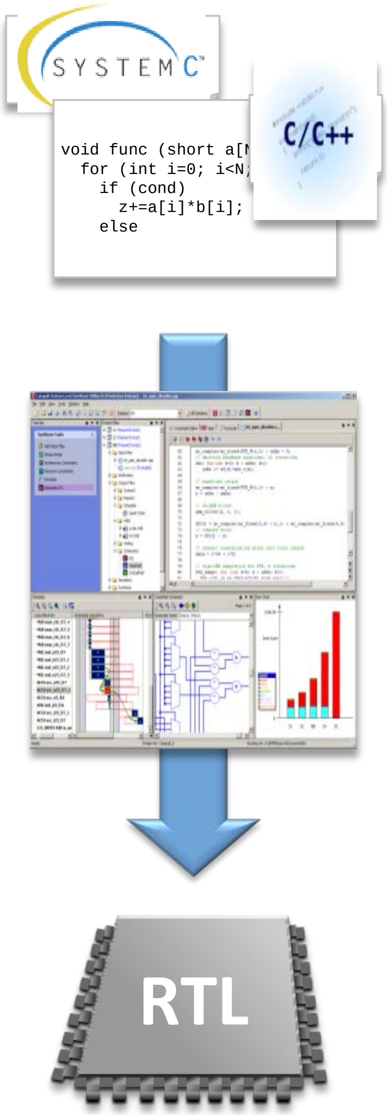


Catapult – HLS Synthesis

- Synthesizes C++ and SystemC to production quality RTL

- "Full-Chip" applicability
  - Algorithms
  - Control-logic
  - Interfaces and protocols

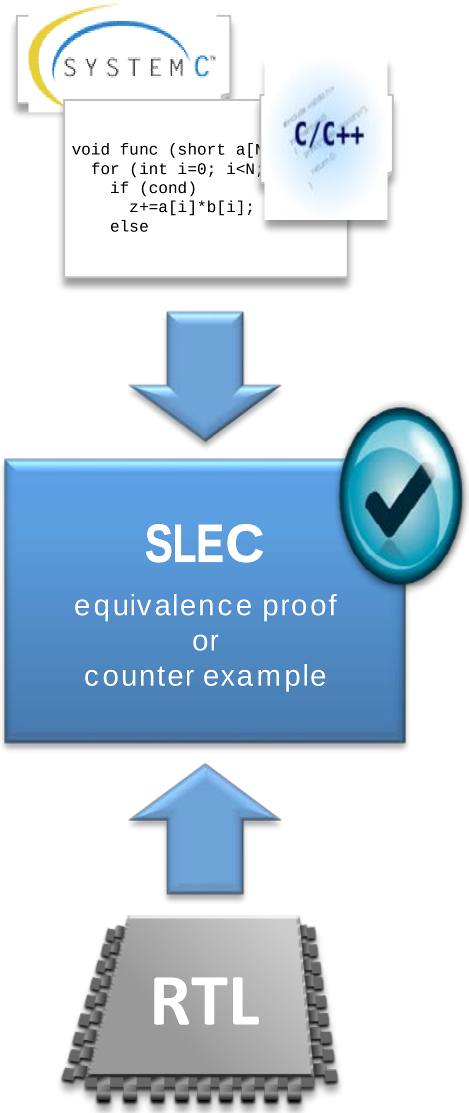
- 10x design and verification productivity gains
  - Design space exploration for best QoR
  - Integrate last minute specification changes
  - Efficient retargeting of high-level IP
  - Correct-by-construction RTL



```
void func (short a[N
  for (int i=0; i<N;
    if (cond)
      z+=a[i]*b[i];
    else
```

SLEC – HLS Formal Verification

- Sequential Logic Equivalence Checker
  - Handles timing differences in internal computation and at interfaces
- Verifies across the ESL to RTL continuum
  - C property checking
  - C-to-C formal verification
  - C-to-RTL formal verification
- Total verification confidence
  - Eliminates risk of design inconsistencies
  - No need for lengthy simulation runs
  - Unlocks the full potential of ESL



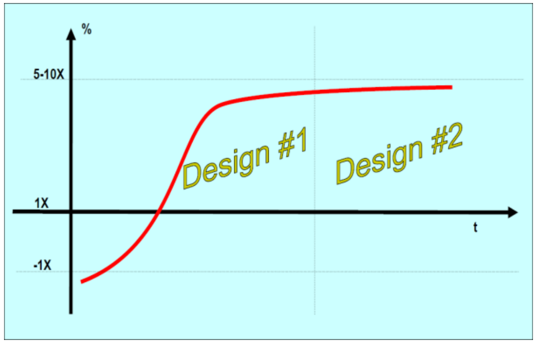
```
void func (short a[N
  for (int i=0; i<N;
    if (cond)
      z+=a[i]*b[i];
    else
```

Adoption

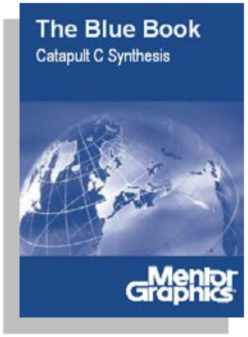
- Strength in wireless, image and video processing markets
- Adoption increasing worldwide as knowledge and confidence in methodology spreads
- 14 of top 20 semiconductor companies have multi-division deployment of HLS
- Continued steady growth in ASIC tapeout numbers

Adoption

- Public User Testimonials
  - Productivity Benefits > 5x
    - Higher for derivative designs
    - Enhanced with C-to-RTL formal equivalence checking
  - Verification benefits
    - Fewer bugs
    - C-to-RTL formal equivalence checking
  - QoR optimization through visibility, user control, incremental methodology
    - Power optimization through architectural exploration

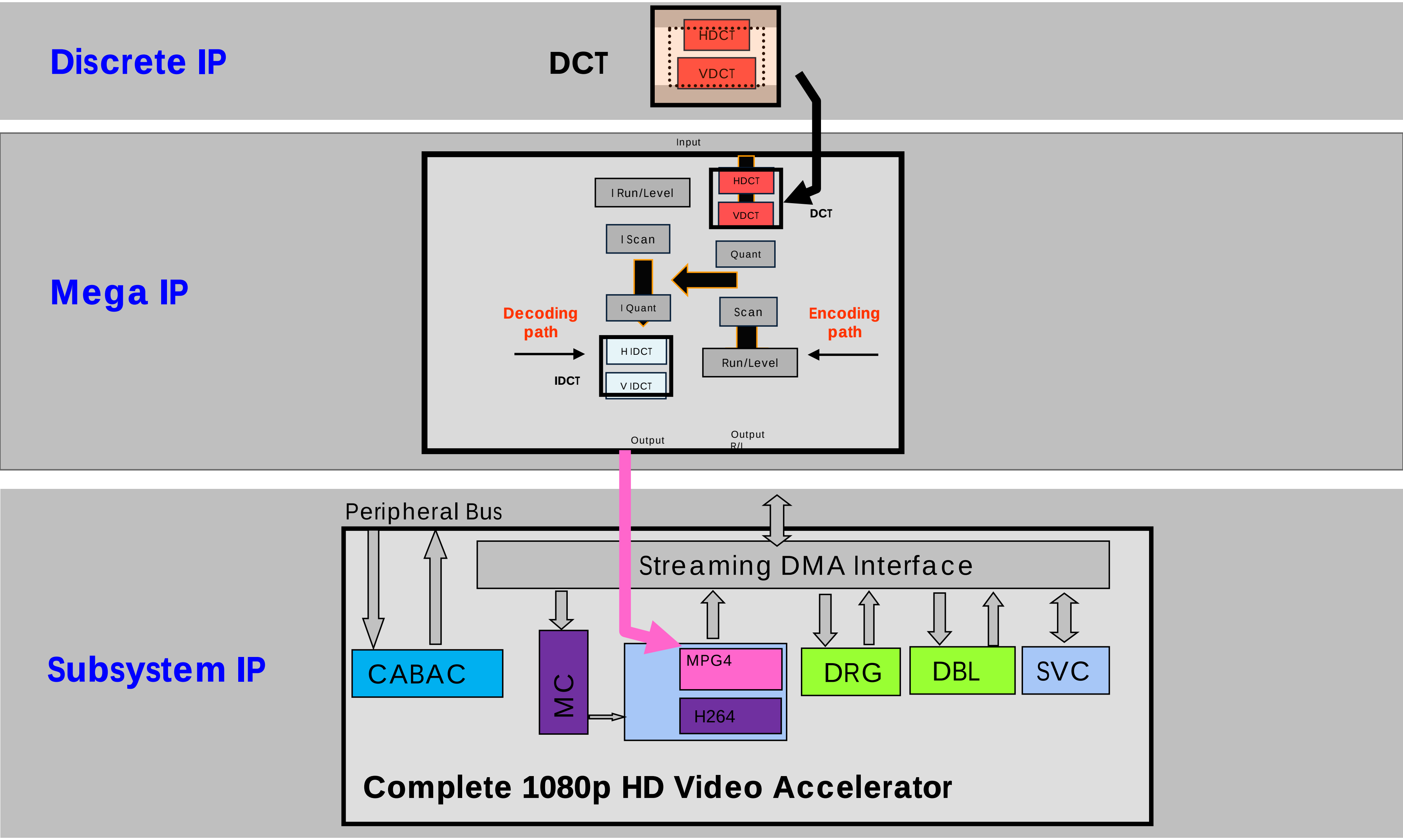


- Calypto's Blue Book:
  - minimize rampup and achieve best results



- Ecosystem
  - Silicon Vendor Partners
  - Certified EDA integrations
    - Leading RTL Synthesis, Simulators, RTL Equivalence, linting
    - Sequential Logic Equivalence
    - Power analysis and Optimization

Evolution of HLS Deployment



Discrete IP

Mega IP

Subsystem IP

Complete 1080p HD Video Accelerator

Customer Expectations

- HLS users expect to see the corresponding tools and flows that are available at the RTL level
  - Equivalence Checking
  - Power analysis and optimization flows
  - Assertion and property checking
  - Tools to check for C correctness
    - Check for out-of-bounds accesses
    - Check for use of undefined behavior
  - Tools to debug C++/SystemC
  - Linting tools
  - Coverage tools
  - ECO
- Synthesis Standard

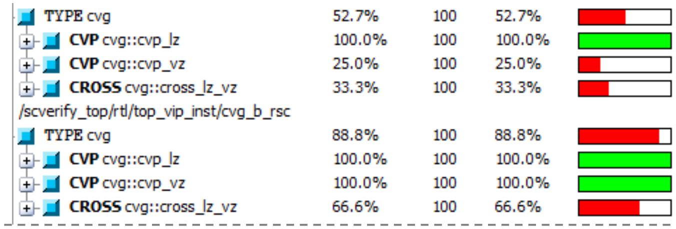
Coverage-Driven HLS Verification Flow

- Support for Assertions and Cover Points
- Catapult propagates assertions and cover points from C to RTL
- Formally check assertions with SLEC
- Facilitates functional coverage in the HLS flow
- Accellera OVL support

```
int alu(int a, int b, uint2 opcode)
{
  cover(opcode==ADD);
  cover(opcode==SUB);
  cover(opcode==MUL);
  cover(opcode==DIV);

  short r;

  switch(opcode) {
    case ADD:
      r = a+b;
      break;
    case SUB:
      r = a-b;
      break;
    case MUL:
      r = a*b;
      break;
    case DIV:
      assert(b!=0);
      r = a/b;
      break;
  }
  return r;
}
```



```
TYPE cvg                                52.7%     100    52.7%
 + CVP cvg::cvp_lz                      100.0%    100    100.0%
 + CVP cvg::cvp_vz                      25.0%     100    25.0%
 + CROSS cvg::cross_lz_vz               33.3%     100    33.3%
/scverify_top/rtl/top_vip_inst/cvg_b_rsc
TYPE cvg                                88.8%     100    88.8%
 + CVP cvg::cvp_lz                      100.0%    100    100.0%
 + CVP cvg::cvp_vz                      100.0%    100    100.0%
 + CROSS cvg::cross_lz_vz               66.6%     100    66.6%
```

Synthesis Subset Standard Draft

- Standards are important for wide deployment
- Accellera's Synthesis Working Group refining the draft for standardization
- What is covered:
  - C++ syntax that is supported
  - SystemC
    - Datatypes
    - Processes (SC_THREAD, SC_CTHREAD, SC_METHOD)
    - Signals, ports
- What is not (yet) covered:
  - Synthesis vs. TLM models for IO
  - Verification
- What needs to improve:
  - Datatypes (bit-accurate integer, fixed-point)

Summary

- Increasing Adoption
  - Strengths in wireless, image and video processing
- Verification is a key benefit that is driving adoption
- Ecosystem of tools around HLS
- Standards are important