HIGH-LEVEL SYNTHESIS REPORT 2011

SHAWN McCLOUD, CALYPTO DESIGN SYSTEMS

WHITEPAPER

CALYPTO®

ESL DESIGN & VERIFICATION

ABSTRACT

This High Level Synthesis (HLS) report is based on Calypto Design System's 3rd annual independent worldwide survey executed during January 2011.

A total of 1,133 engineers and engineering management responded. This report analyzes the survey results and identifies relevant emerging trends. Its scope spans HLS adoption, time savings of HLS versus manual RTL, most desired abstraction level, and ESL integration into ESL flows.

A key goal of the report is to define the critical elements semiconductor company management must consider as they evaluate HLS deployment. The topics covered are:

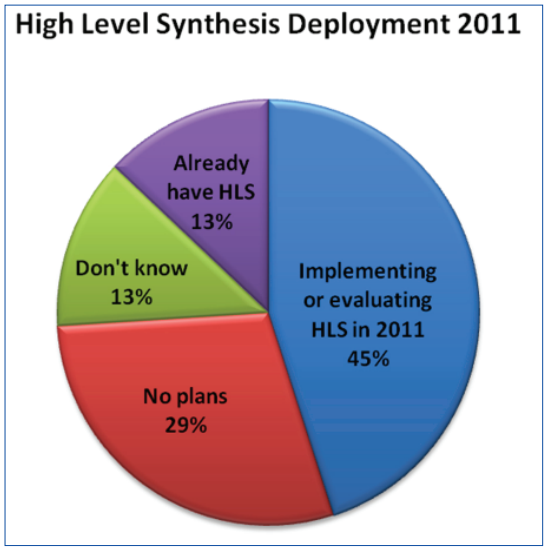
- Survey Methodology and Demographics
- High Level Synthesis - Current Adoption and 2011 Plans
- HLS Time to Verified RTL versus Manual RTL Implementation
- Length of Delays Due To Late Functional Changes and Bug Fixes
- General Criteria Used for HLS Tool Selection
- Preferred Level of HLS Abstraction for Design Implementation
- HLS Integration into ESL Flows
- Summary and Conclusions

SURVEY METHODOLOGY AND DEMOGRAPHICS

A blind, anonymous survey was emailed to several thousand SoC/IC design professionals worldwide by an independent consultant during January 2011. 1,133 engineers and managers completed the survey online.

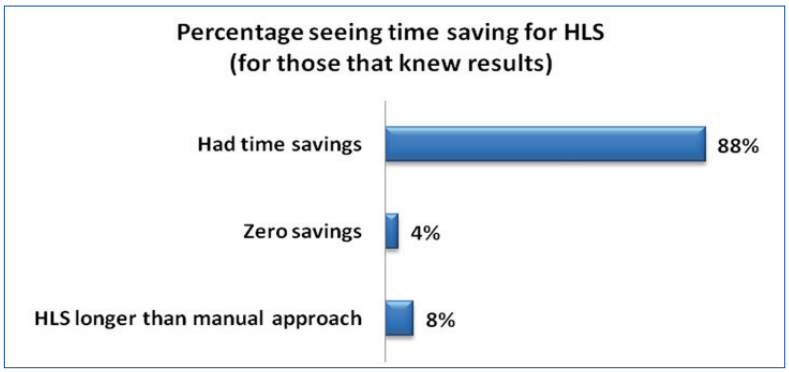
HIGH LEVEL SYNTHESIS - CURRENT ADOPTION AND 2011 PLANS

13 percent of respondents stated their organizations have deployed High Level Synthesis technology. Another 45 percent said their organizations intend to evaluate or implement HLS 2011, so that the total exposure of organizations to High Level Synthesis by the end of the year could run as high as 58 percent.

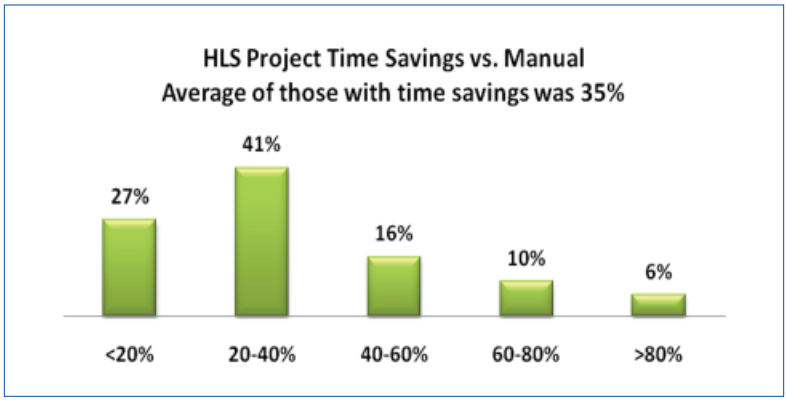


HLS TIME TO VERIFIED RTL VERSUS MANUAL RTL IMPLEMENTATION

452 of the 1,133 respondents indicated their organization had HLS experience. An overwhelming 88 percent that knew the relative time difference saw a time savings between using HLS to get to verified RTL versus manual approaches. This result is particularly notable as that many organizations are likely to still be in the early stages of setting up and deploying HLS. 4 percent saw no time difference, and 8 percent said it took longer to use HLS than manual RTL creation.

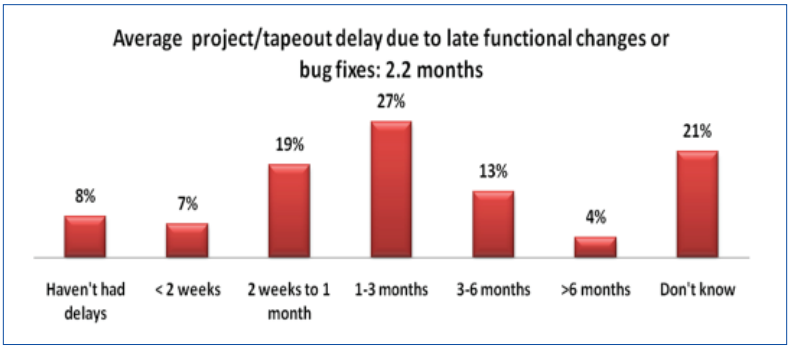


For those engineers that indicated they had seen time savings from HLS versus manual methods, the average time savings for using HLS was 35%. All time savings estimates reflect the time to implement RTL through verification.



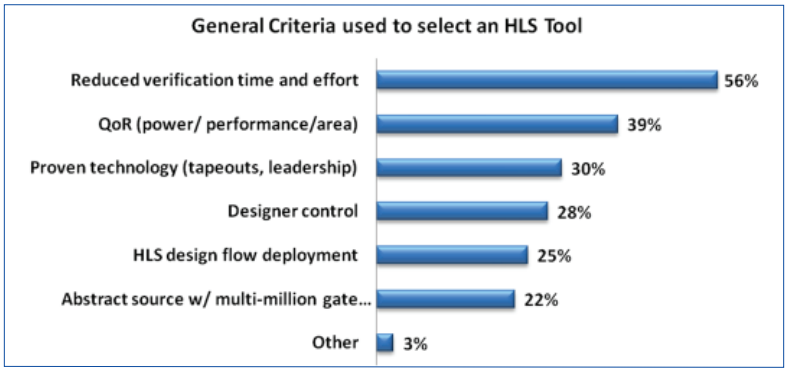
LENGTH OF DELAYS DUE TO LATE FUNCTIONAL CHANGES AND BUG FIXES

For those that had experienced project delays associated with late functional changes or bug fixes, the average delay was 2.2 months. This is relevant to an HLS review because the relative time savings for using HLS go up significantly for late design changes. This is because late functional changes and fixes can be readily made and verified at the C level, and automatically implemented as RTL.



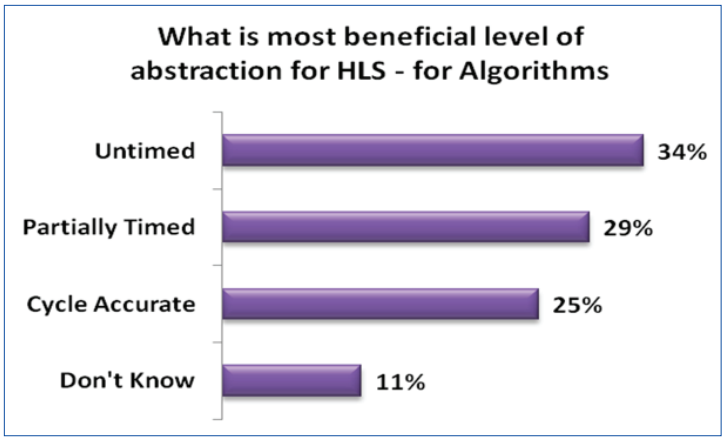
GENERAL CRITERIA USED FOR HLS TOOL SELECTION

Reduced verification time and effort ranked as the number one criteria in selecting a particular HLS tool (56%), followed by QoR (39%) and proven technology, based on tapeouts and leadership at 30%. 28 percent cited designer control, 25 percent cited HLS design flow deployment, and 22 percent mentioned abstract source. The verification time and effort closely relates to abstraction level and is covered in the next few questions.



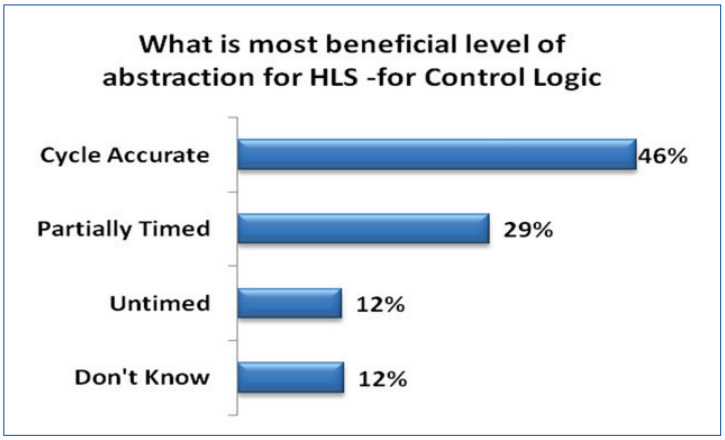
PREFERRED LEVEL OF HLS ABSTRACTION FOR DESIGN IMPLEMENTATION

Users cited their preferred level of abstraction for HLS. The preferred abstraction level varied depending on whether the design was algorithmic or control centric. Even then there was not a complete consensus for a particular level.



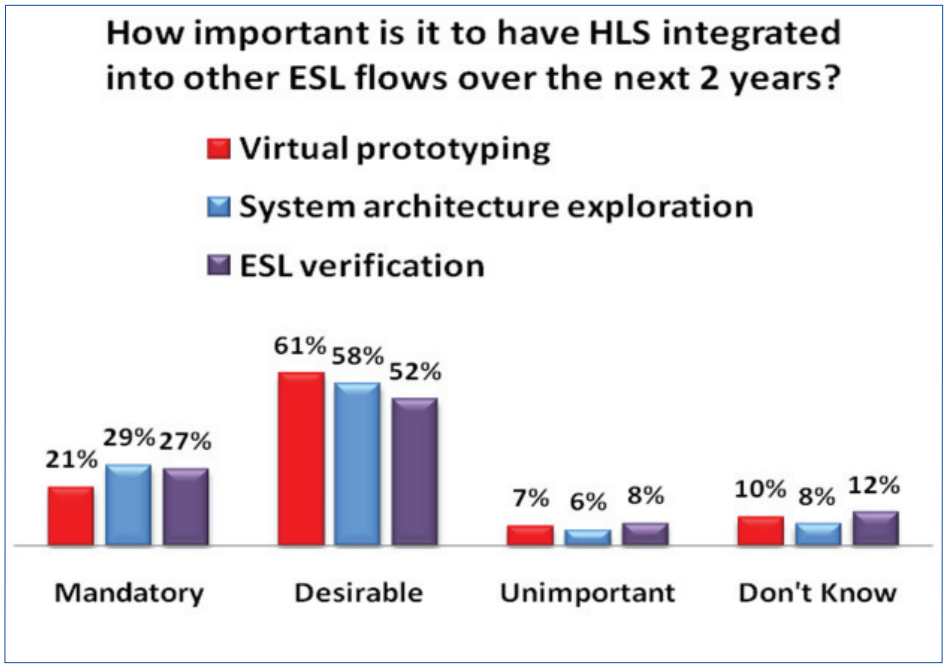
When implementing algorithms, users preferred a higher level of abstraction. Untimed HLS input languages ranked the highest at 32 percent, followed by partially timed at 30 percent, for a total of 62 percent favoring an untimed or partially timed input language. Only 25 percent picked cycle accurate as the preferred abstraction level for implementation of algorithms.

In contrast, the most beneficial level of abstraction for implementing control logic was considered to be the cycle accurate level (46%), followed by partially timed (29%), and untimed (12%).



HLS INTEGRATION INTO ESL FLOWS

Overall, from 79 percent to 87 percent of those surveyed felt it was desirable or mandatory to have HLS integrated into other ESL flows over the next 2 years. Only 6 percent to 8 percent felt HLS integration into ESL flows was unimportant. The ESL flow integrations evaluated were: Virtual prototyping, System architecture exploration, and ESL verification.



SUMMARY AND CONCLUSIONS

HLS continues to gain more organizational attention and traction worldwide.

The total exposure of organizations to High Level Synthesis by the end of 2011 could run as high as 58 percent, comprised of 13 percent of organizations that have already deployed HLS and 45 percent that intended to evaluate or implement HLS this year. For those with HLS experience, their average time savings to verified RTL was 35 percent over manual RTL methods.

For those that had experienced project delays associated with late functional changes or bug fixes, the average delay was 2.2 months. HLS productivity goes up with late functional changes and fixes, as the updates can be readily made and verified at the C level and automatically implemented as RTL.

Users cited their preferred level of abstraction for HLS, both for implementing algorithms, and for implementing control logic. When implementing algorithms, a higher level of untimed abstraction was the top choice. The most beneficial level of abstraction for implementing control logic was considered to be the cycle accurate level.

Overall, the vast majority of those surveyed felt it was desirable or mandatory to have HLS integrated into other ESL flows over the next 2 years. The ESL flow integrations evaluated were: Virtual prototyping, System architecture

exploration, and ESL verification. Reduced verification time and effort rank as the number one criteria in selecting a particular HLS tool, followed by QoR, and proven technology, based on tapeouts and leadership. This mirrors Mentor's view that the right HLS tool must minimally include the following elements:

- Integrated C/C++/SystemC to RTL verification methodology to reduce overall RTL verification and implementation time
- Excellent quality of results for power, performance, and area
- Designer control over HLS tool, including interactive analysis and incremental engineering change order (ECO) capabilities to meet design goals in the shortest time
- Tool capacity to handle multi-million gate, hierarchical designs
- Integrated ESL methodology between HLS, virtual platforms, and system verification
- Customer success track record to show product maturity and completeness
- Support for mixed abstraction levels to accommodate different design types

ABOUT CATAPULT

The Catapult C Synthesis tool automatically generates control and algorithmic RTL multi-block designs from pure ANSI C++ and SystemC sources. This process gives designers time and freedom to automatically perform detailed design exploration and quickly achieve fully optimized and error-free hardware implementation. By accelerating time to verified RTL without sacrificing quality of results, Catapult C provides the productivity boost required to tackle the design and verification challenges of modern ASIC design. Catapult C has been recognized as the High Level Synthesis market leader by Gary Smith EDA for 4 years in a row.

Shawn McCloud is the V.P of Marketing at Calypto. Prior to Calypto Mr. McCloud joined Mentor Graphics in 1994 after several years as a senior system architect responsible for RISC and CISC based micro-processor design. Shawn received his B.S. degree in electrical and computer engineering from Case Western Reserve University.