

Accellera Forum Report

105062600 Yi-Cheng, Chao 20:38, May 7, 2017

Accellera Update (9:10~9:30)

As the first program on Accellera forum, Mr. Dennis do a basically overview of Accellera. As we learn in “Electronic System Level” course, Accellera is dedicate to providing a platom, for example, systemC, in which the electronics industry can collaborate to innovate and deliver global standards that improve design and verification productivity for electronics products. For more and more complexity program, there is a huge gap between specification model and RTL model, if we can take advantage of the concepts of transaction level modeling, we can start from specification model and level by level to achieve RTL model level like the graph as shown below:

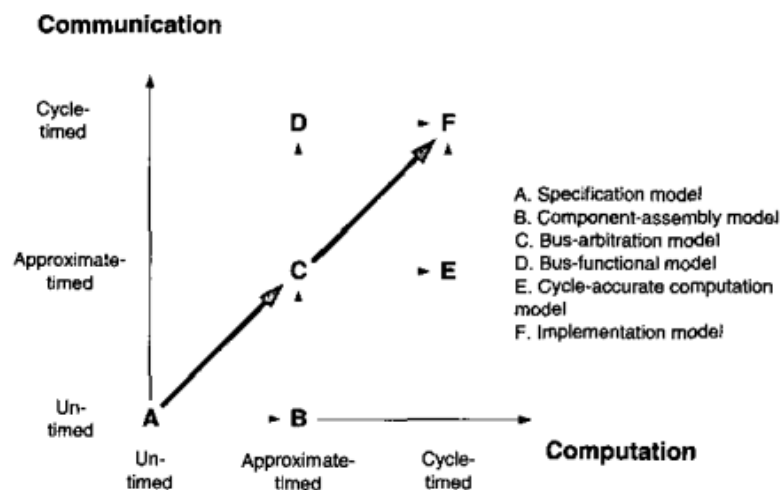


Figure 1: System modeling graph

“Transaction Level Modeling: An Overview”, L. Cai and D. Gajski, CODES+ISSS’03

At that time, SysyemC become the bridge to complete the communication between each model level. As a EDA engineering in the future, we cannot over-emphasized the importance of transaction level modeling.

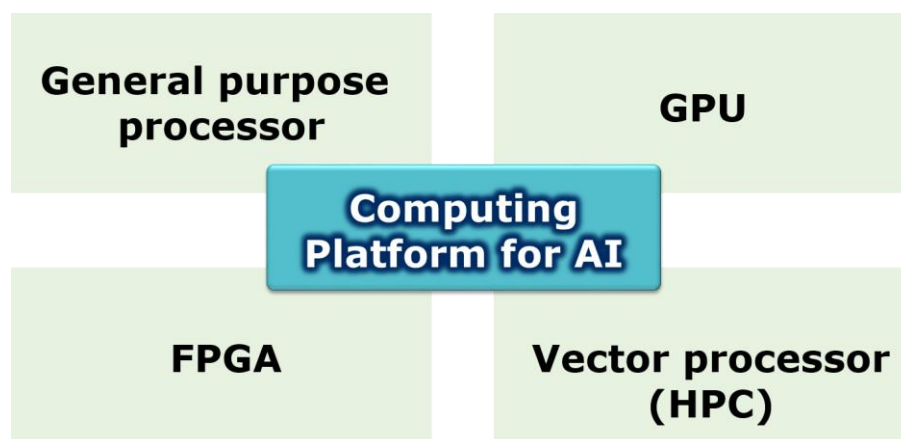
The corporate members and associate members shown in slides also confirm the importance of Accellera in Electronic Design Automation, in company with the growing of accellera, I convinced that leraning systemC is a good experience and will become a useful tool in my career.

IoT and AI Acceleration with FPGA + HLS (9:10~9:30)

As a representative of NEC Corporation, Ph.D. Kazutoshi presents the slides with a very heavy accent. Because I have taken “Parallel Programming” course last semester, in this course, we learn some parallel model and use parallel language to implement. Like use MPI to perform distributed memory model to do even-odd sort, use OpenMP to perform shared memory model to calculate mandelbrot set and single source shortest path, and use CUDA to implement blocked all pair shortest path. FPGA speeding up provide a new view for performance improvement, so I have the largest interest in this topic.

Jensen Huang, the Co-CEO of Nvidia, say that ” Today, NVIDIA is at the center of the AI revolution ”, CUDA is a very powerful GPU computing language to improve the performance, or acceleration the computing in another word, the main concept of acceleration of GPU is to parallel the computing, compact the computation time through multiple GPU, but hardware acceleration through FPGA provide another view to handle this problem.


NEC, as a TOP semiconductor company which is devoted to focusing on the social value creation to solve these issues both in the world, and in Japan. In near future, such surveillance systems for public safety will be strongly empowered by AI. As we known, Artificial Intelligence based on some complex computing like neural network, data training, deep learning and data mining. Operated in real-time is a big issue of such AI-powered surveillance systems because real-time operations can change the value to be created. And the computing time which based on performance is the problem we have to deal with, nowadays, how to speed up is the problem, like use parallel programming through GPU, the graph shown some platform for AI computing.



The heterogeneous computing is the mixture use of the specialized processor which is best suited to the application. As the semiconductor company mention before , NEC would like to focus on its activities for FPGAs.

FPGA is the programmable hardware, which enables to construct the best suited architecture to the target application. To construct its own dedicated circuits which are best

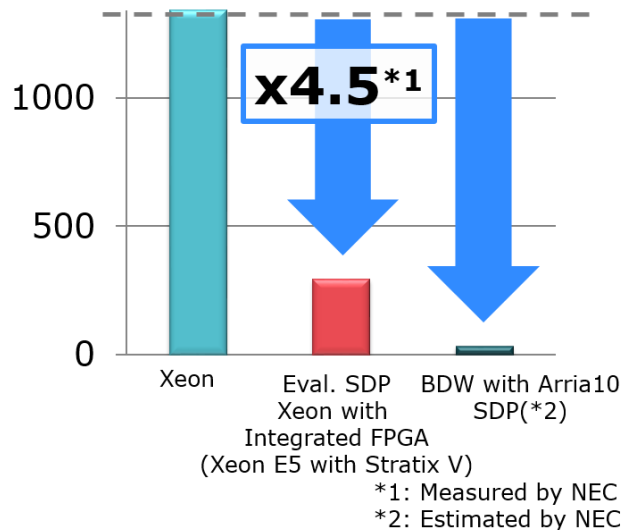
suited to the AI-algorithm. FPGA is capable to do that, by changing its configuration. By construct the optimized architecture for the application, FPGA can enjoy high performance and low energy consumption at the same time.

Feature	Type	example
General & slow	CPU	Xeon
	many core	Xeon Phi
	CPU+GPU	Nvidia+CUDA
	CPU+ FPGA	HLS or, RTL
Special & fast	CPU+ ASIC	Xeon + Crest

The result of hardware speeding up amaze me, because it can improve better performance than GPU speeding up and speed ip from different views. The main advantage of CPU+FPGA is the broadband interconnect between CPU and FPGA and a cache coherent mechanism for the shared. In these years, FPGA is getting popular in data center acceleration to improve performance while keeping the power consumption minimum. In order to utilize CPU+FPGA for AI computing platform, there are two challenge have to face.

The first challenge is communication between AI and FPGA engineers. Recently AI algorithms are quickly improved day by day. To implement the latest AI algorithm on FPGA quickly is the problem. However, for AI engineers, FPGA is hard to implement their algorithms because hareware-specific language is required. On the other hand, for FPGA engineers, the latest algorithms are too hard to understand without AI engineers' help. So FPGA speeding up need the same language for these engineers to communicate each other. Another challenge is the communication between CPU and FPGA. CPU+FPGA has wideband interconnect, so it is important to fully utilize its potential.

However, it is time consuming to tune the performance for each algorithm. The slides show the optimized for the CPU-FPGA communication. With this library, we can fully utilize the potential of the broadband interconnect between them. In addition to the performance, the design time will be also drastically reduced by the combination use of the cyberworkbench.

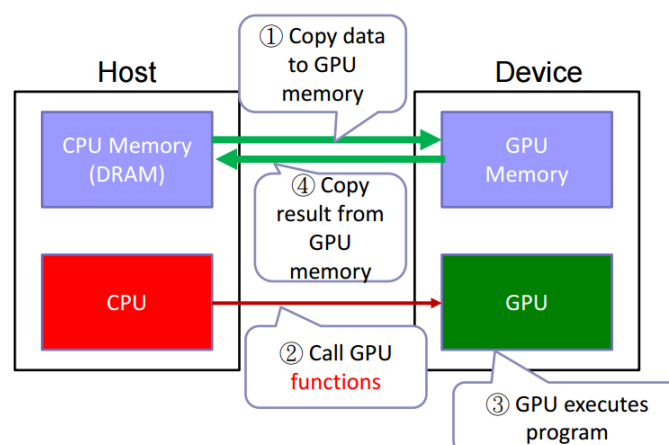


The improvement is very significantly, It achieved 4.5 times performance improvement compared to the general purpose processor.

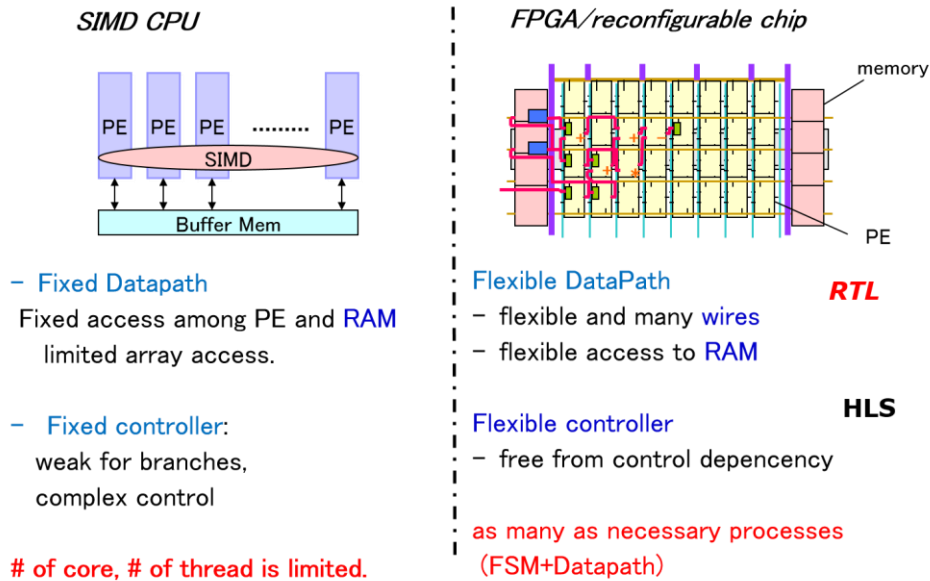
The slides also present the comparison between GPGPU speeding up and FPGA speeding up. The GPGPU speeding up always have the bottleneck on I/O time, and as learning in “Parallel Programming” course, most of the time, I/O time cannot be parallel handle (Except some case like sorting), and memory access time always cost a lot, not only PCIe but also memory, even used the shared memory.

Before compare with FPGA speeding up, we have to know how GPU speeding up implement. A basical GPU program flow, take CUDA for example as shown below

CUDA program flow

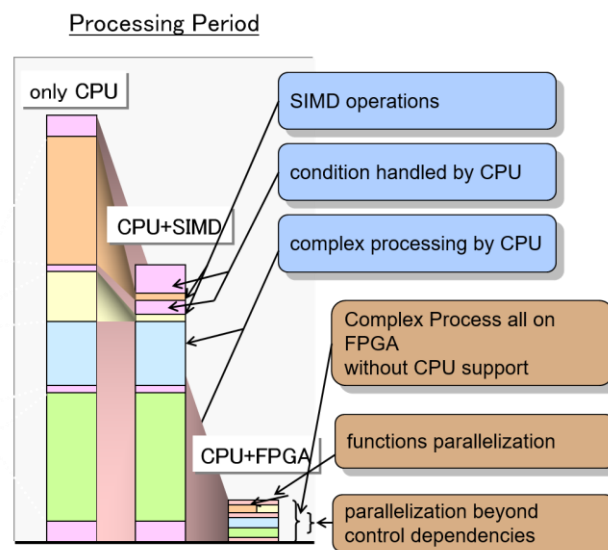


The transmission from CPU (Host site) to GPU (Device site) will happen many times based on program design. PCIe transmission always be the bottleneck in GPU speeding up, but FPGA speeding up flexible datapath will deal with this problem, and lead to the better performance, like the graph below:



As the result, FPGA is a good choice for low latency area, and if we take advantage of high level synthesis and FPGA, free from control dependencies cause the better performance than GPGPU. Today, to achieve the better performance is more and more importance, if we can improve performance through both algorithm and hardware, I convinced that the speeding up result of the final combinational performance will be quite awesome! Through this presentation, I come to understand the improvement of performance is the duty not only the software but also can implement on hardware!

There is a briefly described graph shown on the slide, and I also cite on below:

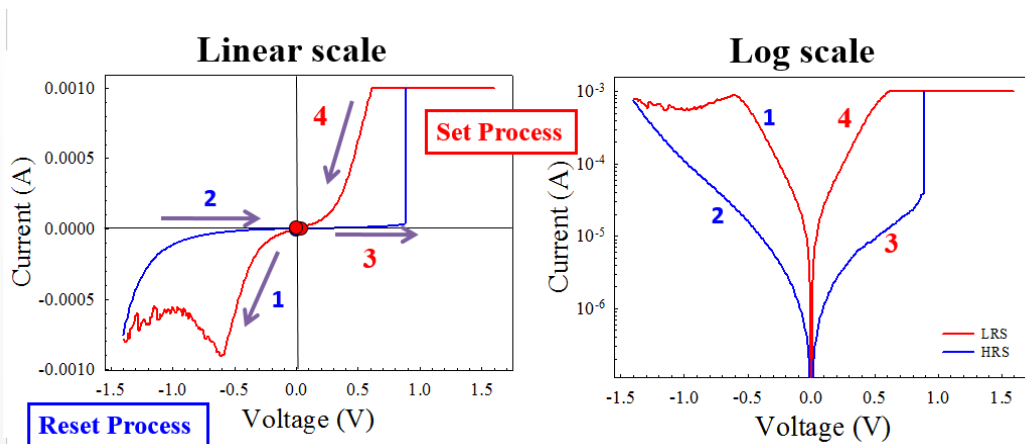
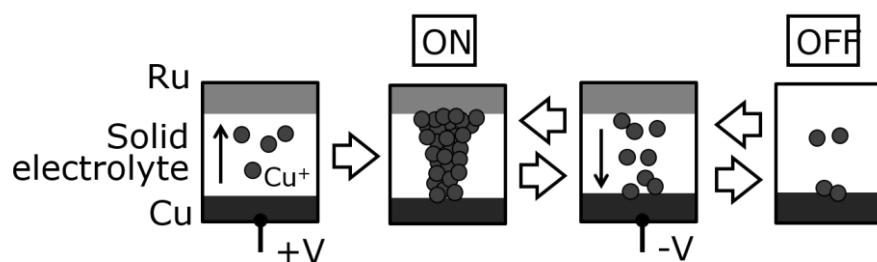


Look at this graph can easily know the advantage of CPU+FPGA speeding up. In CPU+SIMD, or we can say software parallel programing, the timing we can compact only account for a tiny ratio ,which the performance improvement based on. The complex processing

by CPU, I/O timing and so on may not be executed in parallel. On the other hand, the CPU+FPGA speeding up can perform well, which can parallel the process in realistic, and the performance improvement can easily observed through the above graph.

Integration the computing platform mention above, NEC has several research activities on each layer, like two topics introduce in this slides: data management and heterogeneous computing technologies.

The non-volatile wiring FPGA using “nano bridge” cost smaller and lower power can also reduce the heat waste. The mechanism of nano bridge, or atom switch, is quite like my university research about resistance RAM. Use the positive/negative voltage to control the tunnel formation, the I-V sweep and the Cu bridge forms between two electrodes via electrochemical reaction as shown below:



The mechanism through the voltage assignment create the tunnel to communicate from top metal layer (Ru in atom switch in these slide case) to low metal layer (Cu in atom switch in these slide case) , and allow the current easily passing, to perform the logical signal “1”, and recover the initial state (In another word, bury the created tunnel) to create resistance perform the logical signal “0”, with the tunnel create/disappear acts as the On/Off signal, and based on this architecture, it features with non-volatility, small write-erase time, w program energy, small cell size also optimize the performance. Finally, from high level synthesis + FPGA to reality hardware usage construct the whole architecture to reach the best performance.

SoC Verification (10:30~12:00)

“System-on-Chip” is a architecture consist of one or more embedded processors, some IP (include standard I/O, system infrastructure, differentiators), and also with some software, SoC combines functionality that used to be distributed across multiple chips and perhaps even some discrete devices. It’s hard to think of a system that does not contain some sort of processor, so the practical definition says that the SoC must include at least one processor. The presence of processors is the key to what makes SoC verification different from verification of other chips. Smaller, less complex chips, as well as many of the blocks within the SoC, can be verified effectively using a simulation testbench that provides data to the chip inputs and checks resulting data on the chip outputs. Traditional testbenches may be as simple as a framework that allows the user to provide a series of binary values to the inputs and check output results using a waveform viewer. Of course, such a manual setup can verify little of the intended functionality for a complex design.

In this tutorial tell me why SoC integration verification, lint/super-lint are required before designers block go into verification, formal-based integration “Linting” before integrated subsystem/system go into system-level verification. Similar to RTL linting, many issues can be cleaned up a lot quicker to enable system-level functional verification.

System specification (IP-XACT or other format) defining integration information, verify that the integrated system meets the specifications. It is not SoC functional or system verification. It verifies that the System level integration specification matches the RTL, for example: verify that a master is reaching a slave via expected path, we check: the physical connection along the path, the address mapping, mostly to ensure that the transaction passes through the interconnect, the interconnect is configured correctly, there is no other paths that the expected path, does not verify the full data-integrity.

Cite the verification flow shown on the slides will get briefly understand, as shown below:



- Clock and Reset Verification
- Control & Status Register Verification
- Master/Slave Path Verification
- Connectivity Verification

In verification stage, the key 3C’s used to prove all connections in the chip are correct, controllability, coverage, completeness respectfully. Traditional simulation methods lack

efficient way to exercise the connections, significant manual effort to apply stimulus is the controllability, very difficult to assess coverage in a simulation environment, exercise all combinations is the coverage problem. Finally, it's a big challenge to identify whether all connections are tested by the testbench, checks for all connections is the completeness key. In these tutorial we learn how to deal with each challenge.

This tutorial provide another view at the issue of performance improvement, the speaker say that multi-core is not the only thing that's changed in the last 20 years, Cache memory hierarchy has become increasingly important for performance optimizations, in the 90's, linked-list was considered perfectly fine, but today the view has changed, keeping your elements in a contiguous vector is much better! Careful consideration of modern server architecture is key. Optimizing any 20-year-old architecture for new servers is challenging, and a new simulation engine, re-architected from scratch, is a solution, and also a new algorithm. It cause that a new simulator, Xcelium Parallel Simulator, has been created. Xcelium™ Parallel Simulator, the industry's first production-ready hird-generation simulator, which have multi-core engine architected for fast SoC simulation single-core engine refactored for fast IP simulation, productivity features enable efficient verification.

At the end of this verification tutorial, the speaker also mention the portable stimulus usage flow, and let me have a basic knowledge of the next tutorial.

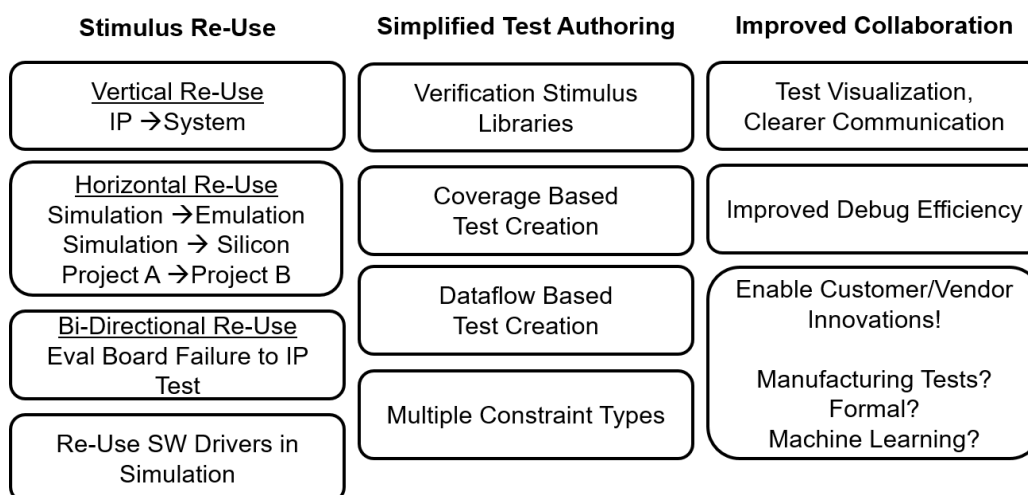
Portable Stimulus Tutorial (13:20~17:00)

The afternoon tutorial is all focused on portable stimulus, as the mention concepts of portable stimulus in the morning, portable stimulus is considered as the next verification productivity after universal verification methodology, to reduce product life cycle efficiency gains via portable content, need a system which can support different stimulus languages on different execution platforms, and in order to enforce single interpretation of a product specification and enable mainstream and methodical automation of test content reuse, encourage verification and validation plans to become a continuum, portable stimulus is necessary.

In functional verification, several different languages and techniques are used to generate verification stimulus depending on whether a block, sub system, SoC or system is being verified. When verify the RTL module and subsystem, there are some languages like systemVerilog, SystemC, VHDL are common used. At SoC and system level, embedded software is frequently used to exercise the design. And the compromise between two level result very challenge. So portable stimulus is dedicated to create a standard in the area of enabling verification stimulus to be captured in such a manner that enables stimulus generation automation, and enables the same specification to be reused in multiple verification languages and contexts.

UVM used to be a good verification model and have some advantage like common language / framework for verification engineers, smart testbench architecture to allow for user, but today, UVM is not sufficient for engineers anymore.

The below graph shows the portability use cases and potential capabilities, and let me know the necessity of portable stimulus:



PS standard enable expansion of VIP Ecosystem beyond UVM simulation VIP and innovation for re-use across platforms from EDA vendors. A standard levels playing field and focus innovation on next set of challenges. Increase predictability for mobility across platforms and vendors make the portable stimulus important again.

Portable stimulus rising the abstraction level from transaction level to scenario, it declarative partial specification of key intent and randomize scenarios based on system-level constraints, stimulus at a higher level let engineers to get a greater platform for operation.

After these PSS tutorial, we learning some reality implement via some example to know the potential of portable stimulus. Nowadays, portable stimulus is a perfect solution for many real problems. Stretch productivity and quality across platforms, users, integrations, and configurations let the portable stimulus become powerful and flexibility.

Cut Your Design Time in Half with Higher Abstraction (Not Attend)

In this subject introduce the high level synthesis, like the paper “*Transaction Level Modeling: An Overview*”, L. Cai and D. Gajski, CODES+ISSS’03 say that, HLS becomes a tool transform synthesizable SystemC into RTL code, we can take advantage of these tool, avoid directly across the gap between specification software code (for example: C, C++ code) and RTL code.

These subject also introduce SystemC as professor teach in class, and roughly introduce the basic concept of SystemC, use these model, we can increasing the design complexity and shorter

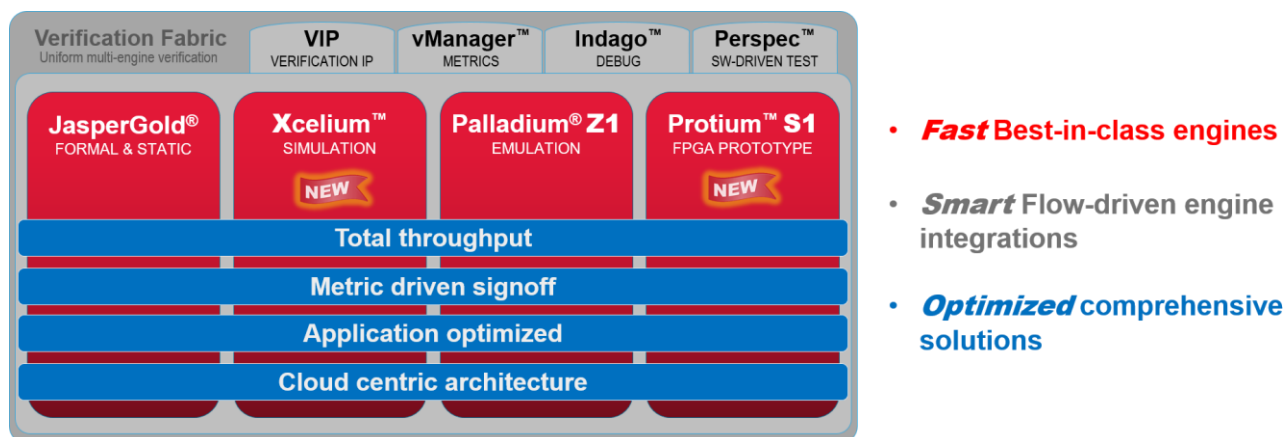
design cycle, and move to higher level of abstraction for not only design but also debug.

The Art of Power, Key to Soc Success – Introduction to Low Power Verification Methodolgy (Not Attend)

In these subject, it may focus on power-aware verification, include static power verification and dynamic power verification. The usage of adjective “Art” is similar with the book “The Art of Computer Science”, I think the speaker must be convinced that power-aware stimulus must get many benefit at a differ view from functional intent.

Latest Advanced Verification Technologies (No Attend)

In these subject, the speaker introduce a serial of innovation technology commercial tool, which are faster, smarter, optimizer engine. The graph shown below briefly show each of the tool.



The Formal Verification EcoSystem Extending to SystemC/C++ (No Attend)

In these subject, the speaker emphasize the importance of formal verification, and use systemC as an example, and meanwhile, also show the high level synthesis flow to deal with the formal verification.