

Silicon Interfaces'

Silicon Cores

presents -

GEMAC
OVC SI80GEOVC10



Agenda

- **Silicon Interfaces'** Overview.
- Added value to the OVM community.
- Products or services that support the OVM.
- Success using OVM.
- Future commitments and offerings related to the OVM.



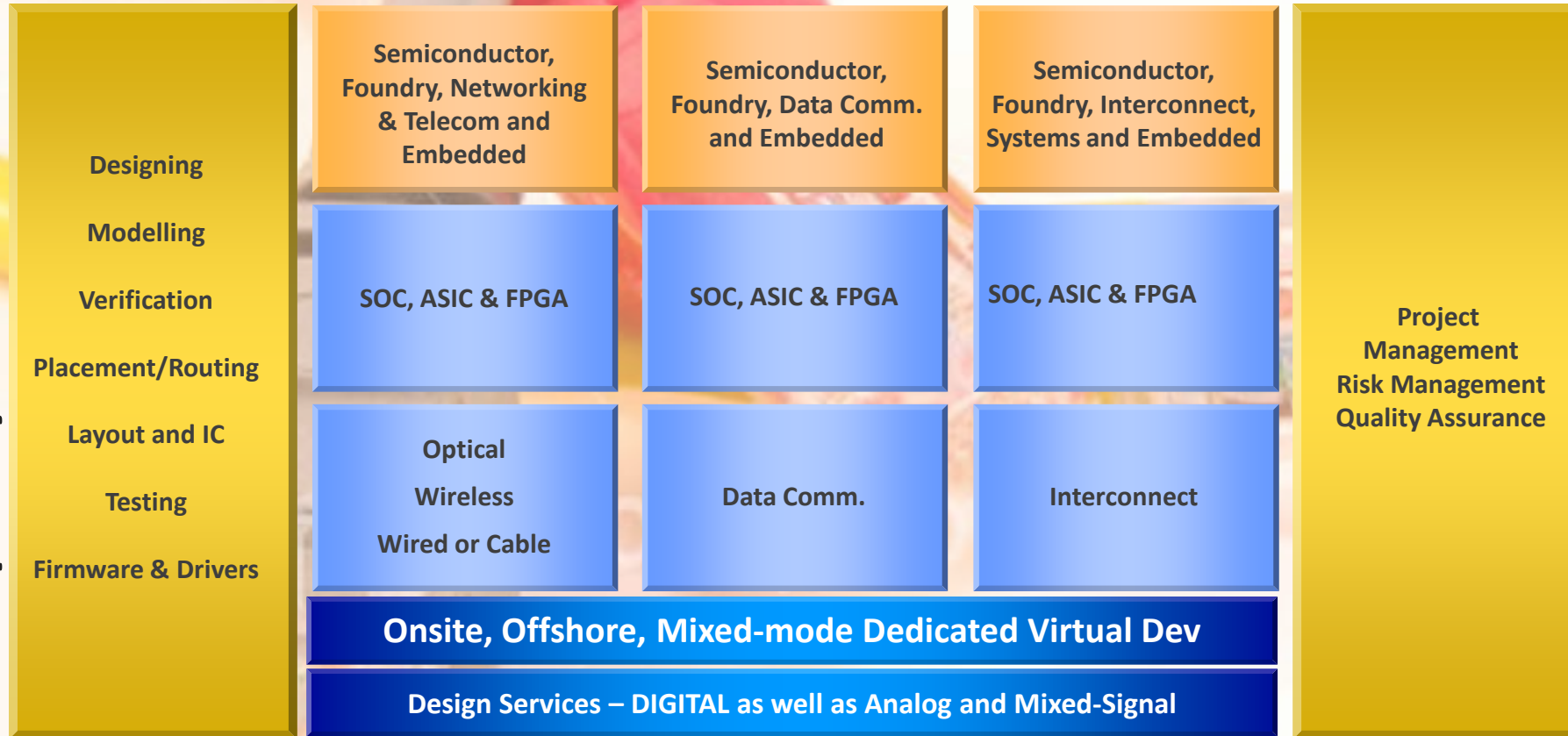
***Silicon Interfaces'* core competencies:**

- Design Services:
 - Frontend Design
 - Verification
 - Backend/Layout
 - FPGA Validation Services
 - EDA Product Services
- Products:
 - Silicon Cores – an IP Development Program



Symphony – *Silicon Interfaces'* Design Services

Company Overview



✓ *perfectly tuned for all requirements*



Silicon Cores

Core to the Intelligent Systems™

Wireless, Communication, Networking & Interconnect

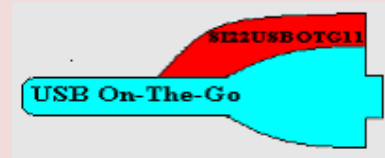
12+ IPs NOW!!!

6+ VIPs NOW!!!



802.11 a/b/g MAC

- ✓ OVCs
- ✓ uVC
- ✓ eVCs



USB On-The-Go



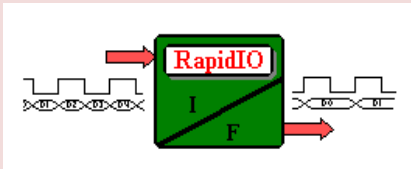
Bluetooth BaseBand



GEMAC OVC



USB2 Function Controller



RapidIO



IEEE 1394



Infiniband

IP Development Program by

Silicon Interfaces®

a software and vlsi design center



GEMAC

Value Add to OVM Community

- A stable platform to verify third party Gigabit IP and Ethernet Interface part of SOC
- Customer adoption of OVM will strengthen based on availability of VIPs and Silicon Interfaces efforts will give confidence and reduce time to market guidelines for SOC verification requiring GEMAC
- Silicon Interfaces has promoted OVM to its Design Services and these efforts are aiding the availability of resources who having working experience and knowledge of OVM



Silicon Cores' Gigabit Ethernet Media Access Control (MAC) System Verilog OVC VIP is fully documented, off-the-shelf component for the developers of the Gigabit Ethernet MAC.

The GEMAC OVC based VIP provides a concise, declarative mechanism to code the specification of sequences of events and activities of Gigabit Ethernet MAC Protocol.

GEMAC VIP is developed using the OVM Class Library that includes `ovm_sequence`, `ovm_driver` etc. base classes and capabilities to create modular, reusable components.

Silicon Interfaces' GEMAC core implements the Ethernet Media Access Control (MAC) protocol according to IEEE 802.3 specification. The MAC has a standard Gigabit Media Independent Interface (GMII) to connect to any PHY interface. A single channel MAC with PCI controller would provide an ideal solution for inexpensive NIC cards.

Products supporting OVM

SI80GEOVC10

Product Features:

- **Fully OVM-compliant** with **OVM2.0.2 Methodology** and completely configurable with **SystemVerilog** environment, as per user requirements and OVM user guide for OVCs.
- Accurately verifies **IEEE 802.3-2000** standard Gigabit Ethernet MAC specifications.
- Full programmability and versatility of the VIP allows connection to any standard GEMAC device.
- Separate set of tasks provided for Transmit as well as Receive sections in Full-duplex mode.
- Coverage driven Verification Environment with Constraint Random test cases using **OVM sequences**.
- Protocol Checkers.
- Application of Stimulus to the Generic Microcontroller Interface as well as PHY Interface.
- **OVM** based transactions using Data Item, Driver, Monitors, Sequencers and Scoreboards.
- Supports scoreboard checking.
- Built in Coverage model implemented for all transaction types.
- Built in Monitors for protocol monitoring and checking.
- Provides checking of valid Inter Packet Gap.
- Support for checking of normal as well as VLAN tagged Packets.
- Supports checking of Pause Frames.
- Provides Checkers for Minimum and Maximum payload sizes, for both normal as well as VLAN tagged Packets.
- Checkers to determine validity of Preamble bits along with SFD.
- Parameterized configuration file supporting various modes operation and types of packets.
- Separate Configuration file for Parameter definitions.

Products supporting OVM



Silicon Interfaces' Services

Silicon Interfaces offers Design Services in the areas of **Wireless, Networking, Communication** and **Interconnect**, primarily using **SystemVerilog/Verilog/VHDL/SystemC** for Design, Modeling and Verification of complex SoC, ASIC and FPGAs. Further, as a recognized guru in the space of Verification Methodologies, it is a one stop solution for Verification in terms of coverage on Languages, Tools and Methodologies.

Silicon Interfaces offers solutions in **SystemVerilog** based on Mentor **OVM/AVM**. That may give you some idea about our experience in churning out solutions in the Verification IP space.

Apart from **Front-end Design and Verification services** we provide **Backend Physical IC Layout services** and we offer a complete solution to our Customer. We also provide specialized **FPGA Validation services**, **IP Cores Development** and **EDA Product Support Services**.

You can choose to have consultant(s) working **Onsite** or have the complete project developed **Offshore** or even choose to have a **mix** of the two modes.



GEMAC



SI80GEOVC10

Challenges in Verification:

- **Shrinking Silicon Geometries** accompanied by multi-protocol Implementation on a single Silicon die.
- **Zero Bug sign-off**
 - Identifying **Corner Cases**.
 - **Unpredictable bugs**.
- Quicker and Shorter **Time-to-market** requires:
 - **Reusability** of Existing Test Environment:
 - New feature or variation to be tested in IP or SOC requires writing a new test environment
 - Reduced load on the Verification Engineer
 - **Better Interface** to Integrate to SOC
 - Legacy test code difficult to integrate because of non-standard interface
 - Ability to **Incorporate Change of Specifications** in Existing Test Environment
 - Change of Specifications requires a lot of re-coding. This can be a major hurdle
 - Reusability
 - Versatility
 - Modularity
 - User-friendliness

Demonstrated Successes using OVM



GEMAC

SI80GEOVC10

Silicon Interfaces had an advantage of using **OVM** for developing a **VIP** for it's IP – **GEMAC** i.e. **GEMAC OVC**

Huge challenges that we had, keeping the big question in mind, is our design fault free ...

- Have an **integrated environment**. The challenge here for us were:
 - keep up with **robust designs and changes/additions to features** that keep happening within projects.
 - test environment had to be **reusable** and **independent** to accommodate any such modifications and additions required.
- Needed **comprehensives tests plan** and an **automated regression process** that submits **thousands of tests to all possible scenarios** and **automated reporting** of each tests and bugs for failures.
- **Reporting** had to be presented **graphically** that showed statistics of tests (passed, failed, crashed etc) and for the bugs over time.
- Developing tests that **verify the correctness** of the design, like self **checking randomization**.
- **Layering of sequences**, so as to **reduce the line of code** and **improve reusability**.

You just cannot think of easily accommodating these in your traditional verification environment.



GEMAC



SI80GEOVC10

First encounter with OVM

- are we using a most complex weapon to kill a housefly?

- Comprehensive Library with huge number of macros, methods and base classes in OVM.
- Heavily relied on object-oriented concepts.

Having progressed,



we realized that this most sophisticated weapon could also be customized to achieve our goals. The difficulties what we faced during this development was soon surpassed by the advantages that we got using OVM.

OVM user guide and Reference manual were so much helpful and also the xbus example helped us to understand the concepts more clearly.

Key Features of OVM:

- First **Open, Interoperable Verification Methodology** with robust OVM class library and compliant to **IEEE 1800-2005 SystemVerilog Standard**.
- Functional Verification Methodology for digital hardware (any HDL at any appropriate abstraction level), primarily using simulation.
- OVM test benches are complete Verification Environments composed of Reusable Verification Components and provide the best framework to achieve Coverage Driven Verification (CDV)
- OVM provides a full-fledged solution for verification.
- Rich set of macros, methods and base classes.
- Abstract classes for all verification components.



GEMAC

OVC

SI80GEOVC10

Products supporting OVM

Advantages of OVM Based VIP

- **Re-usable** Verification Components (**OVC**)
- **Reusable constrained-random stimulus** in the form of sequences and virtual sequences
- **OVM Configuration Mechanism**
- **Standard TLM communication** channel for inter/intra-component communication enable re-modification of the components.
- **Factory automation** that facilitates modifying ovm objects on the fly.
- **Abstract classes** available for all Verification Components.
- **Rich set of Methods & Base Classes**, which makes more efficient Verification environment.
- **Automation Macros**; macros make common code easier to write.
- **Verbosity level setting**; the verbosity can be set from the script file also.
- **Sequence Layering.**
- **Global Functions and Variables.**
- **Reporting.**
- **Flexible layered Testbench architecture** that enables unit to **system level Verification.**
- **Multi-language support** enables **interoperability** of Verification Components.



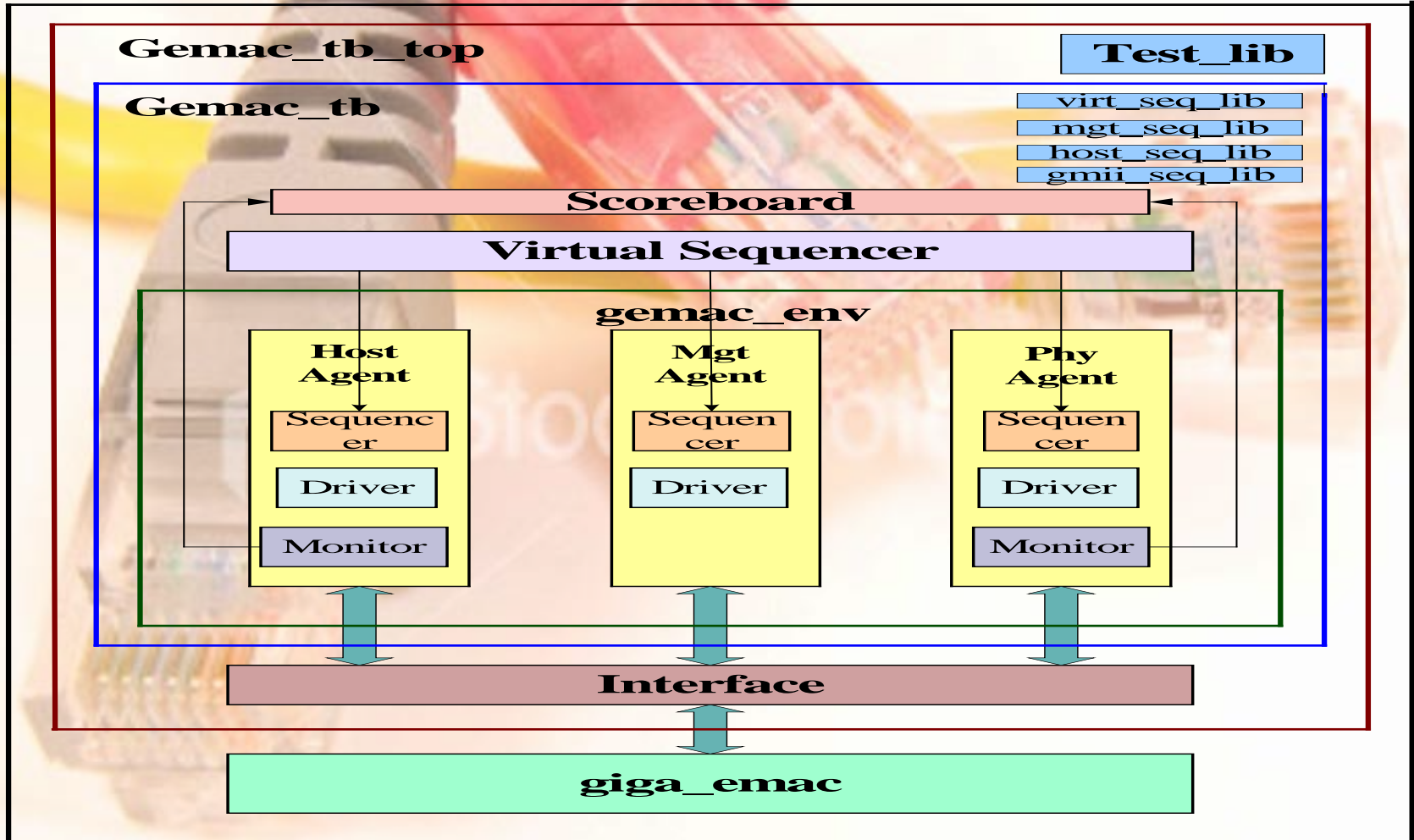


GEMAC

SI80GEOVC10

GEMAC OVC VIP Architecture

Demonstrated Successes using OVM





GEMAC OVC

SI80GEOVC10

GEMAC OVC VIP Test Architecture

- The GEMAC testbench consists of single GEMAC Environment and module components.
- The link between the Environment and the DUT is the SystemVerilog Interface.
- The Agents are the Interface Components and are configured as ACTIVE or PASSIVE.
- This Environment has ACTIVE Agents.
- GEMAC ENV consist of a Host Agent , PHY Agent and MGT agent and are Configured as Active thus driving the HOST and PHY side of the DUT.
- Scoreboard connects to the Env Host and Phy Monitors via the TLM analysis Ports.
- Ability to configure component fields setting using hierarchy fields during the build phase.

Products supporting OVM



GEMAC OVC

SI80GEOVC10

Our GEMAC VIP has three agents in one environment. We started with gemac_host_agent then created the others. Although block level verification is different from top-level verification, using an OVM environment it was easy to develop a top-level environment from block level, by reusing much of the verification components. Most of the changes required were in development of the driver component module.

```
Class gemac_host_agent extends ovm_agent;
.....
function void build();
super.build();
monitor = gemac_host_monitor::type_id::create("monitor", this);
if(is_active == OVM_ACTIVE)
begin
$cast(driver, create_component("gemac_crc_driver", "driver"));
driver.build();
.....
end
.....
endclass
```

The same components can be reused to check the top-level functionality by overriding the driver component for the same verification environment. The method set_type_override() is very powerful and it overrides the existing driver components with new driver, without altering other components.

```
Class gemac_test extends ovm_test;
gemac_env env;
function void build();
ovm_factory::set_type_override("gemac_crc_driver", "gemac_host_driver");
.....
endfunction
.....
endclass
```



GEMAC

SI80GEOVC10

Among the three agents, gemac_mgt_agent is only responsible for configuring the DUT. It was very difficult for us to verify the DUT at different CSR configuration using tradition verification methods. For each test, you have to reconfigure the DUT manually. OVM's virtual sequencer with macro `ovm_do_on and `ovm_do_with provided a new light to this problem using which we can use various combinations of sequence from different sequence libraries.

```
class virt_seq_1 extends ovm_sequence;
`ovm_sequence_utils(virt_seq_1, gemac_virtual_sequencer)
host_normal_pkt_seq_1 host_normal_1;
host_vlan_pkt_seq_1 host_vlan_1;
mgt_seq_2 mgt2;
mgt_seq_8 mgt8;
mgt_seq_33 mgt33;
gmii_normal_pkt_seq gmii_normal_1;
.....
function new();
host_normal_1 = new();
host_vlan_1 = new();
mgt2 = new();
mgt8 = new();
mgt33 = new();
gmii_normal_1 = new();
.....
endfunction

virtual task body();
`ovm_do_on(mgt8, p_sequencer.mgt_seqr)
`ovm_do_on(host_vlan_1, p_sequencer.host_seqr)
`ovm_do_on(mgt2, p_sequencer.mgt_seqr)
`ovm_do_on(host_normal, p_sequencer.host_seqr)
`ovm_do_on(mgt33, p_sequencer.mgt_seqr)
`ovm_do_on(gmii_normal_1, p_sequencer.gmii_seqr)
.....
endtask
endclass : virt_seq_1
```

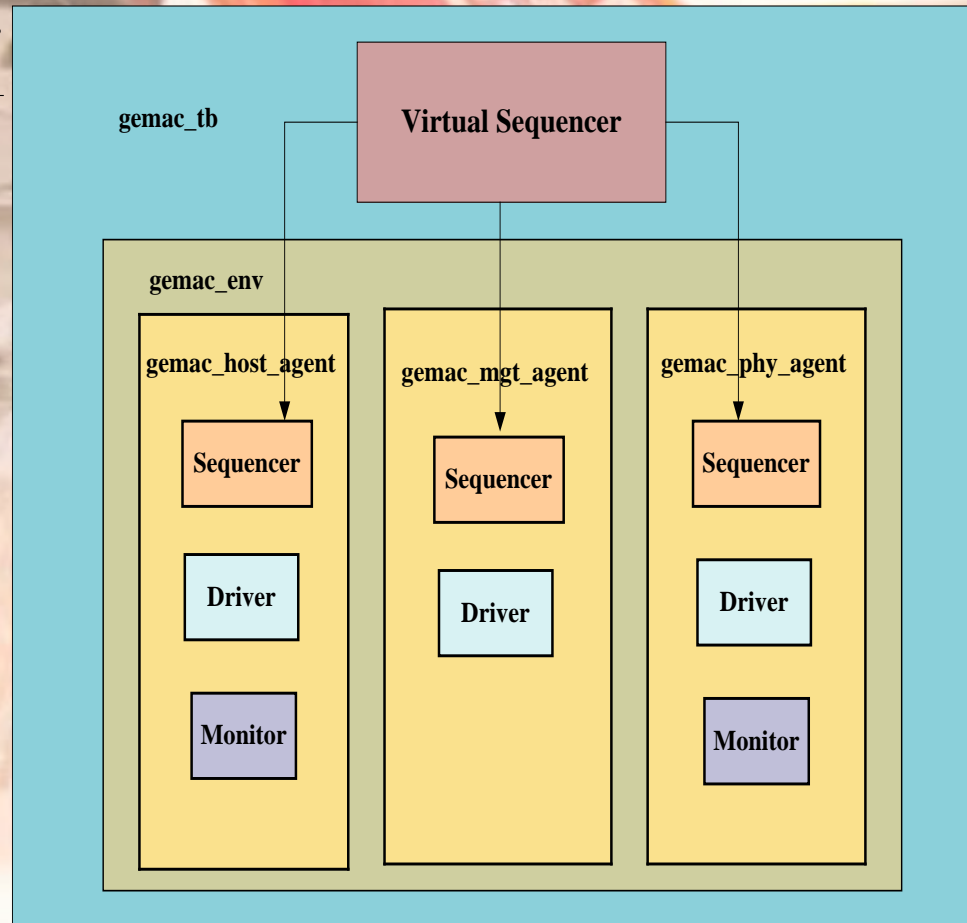
Demonstrated Successes using OVM



GEMAC

SI80GEOVC10

Each `ovm_do_on macro will execute sequentially. It is also possible to execute the sequences in parallel. A simple fork/join with macros `ovm_do or `ovm_do_with will serve this purpose. There is another method start() from OVM that can also be used to execute the sequences in parallel. In case of GEMAC VIP most of our efforts for generating different scenarios were significantly lowered by making virtual sequencer as a mastermind. The diagram below shows how a virtual sequencer takes control of driver sequencers and executes sequences from different sequence library.

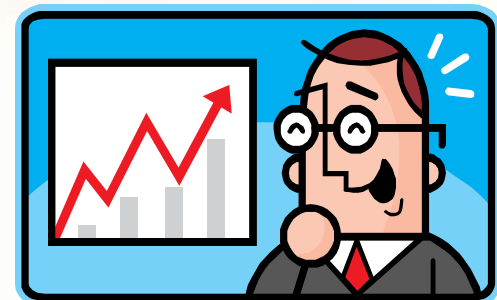


Demonstrated Successes using OVM



Silicon Interfaces' OVM Roadmap

- ✓ Cross platform validation of existing Gemac OVM VIP checked for VCS
- ✓ Cross platform validation USB and 1394 VIPs under OVM framework
- ✓ Developing PCIe OVC





Thank You ...

For more info on:

Services – www.siliconinterfaces.com

IPs & VIPs – www.siliconcores.com

Email – info@siliconinterfaces.com