

An Intelligent SOC Verification Platform

Deyong.Yang(ydy12345678@163.com)

Fabo.Deng(fabo.deng@unisoc.com)



Background Introduction

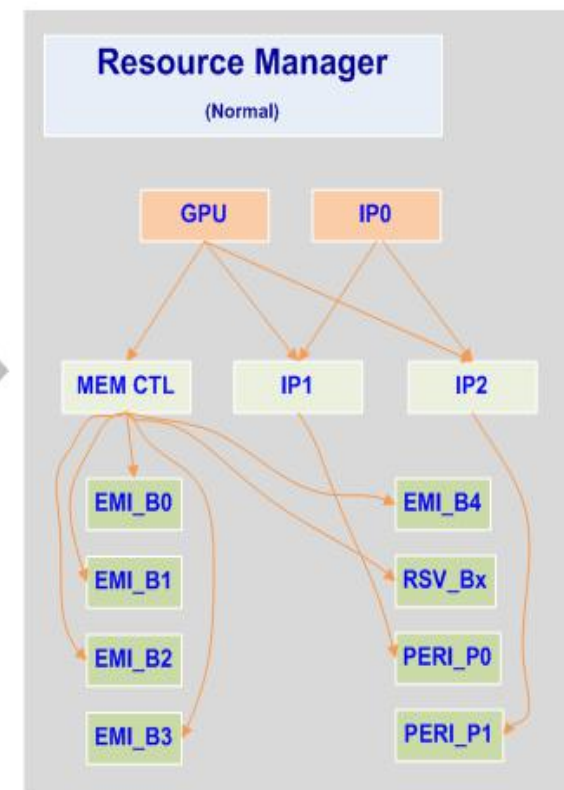


- The features of SOC are more and more complex, but time cost and manpower cost needed to verify these features are more and more.
- To rapidly meet changing market demand and further reduce cost, the project team wishes the faster SOC verification and less manpower.
- SOC features have a strong regularity, So we can build a universal SOC model.

SOC Universal Model

- SOC Model can be abstracted into relationships of master, slave and region.
- Build excel format to express this relationships.
- Design universal verification model to this relationships.

Subsystem_Name APSYS			
Region_Name	Start_Address	End_Address	RW
EMI_B0	0x0000_0000	0x0FFF_FFFF	RW
EMI_B1	0x1000_0000	0x1FFF_FFFF	RW
EMI_B2	0x2000_0000	0x2FFF_FFFF	RW
EMI_B3	0x3000_0000	0x3FFF_FFFF	RW
EMI_B4	0x4000_0000	0x4FFF_FFFF	RW
RSV_B5	0x5000_0000	0x5FFF_FFFF	RW
RSV_B6	0x6000_0000	0x6FFF_FFFF	RW
PERI_P0	0x7001_0000	0x7001_FFFF	RO
PERI_P1	0x7002_0000	0x7002_FFFF	WO
Slave_Name	Region_Name	Power	
MEM_CTL	EMI_B*+RSV_B*	ON	
IP1	PERI_P0	ON	
IP2	PERI_P1	ON	
Master_Name	Slave_Name	Power_Domain	Clock_Dom
GPU	ALL_SLAVE	NA	NA
IP0	IP*	NA	NA



Design Universal SOC Verification Model



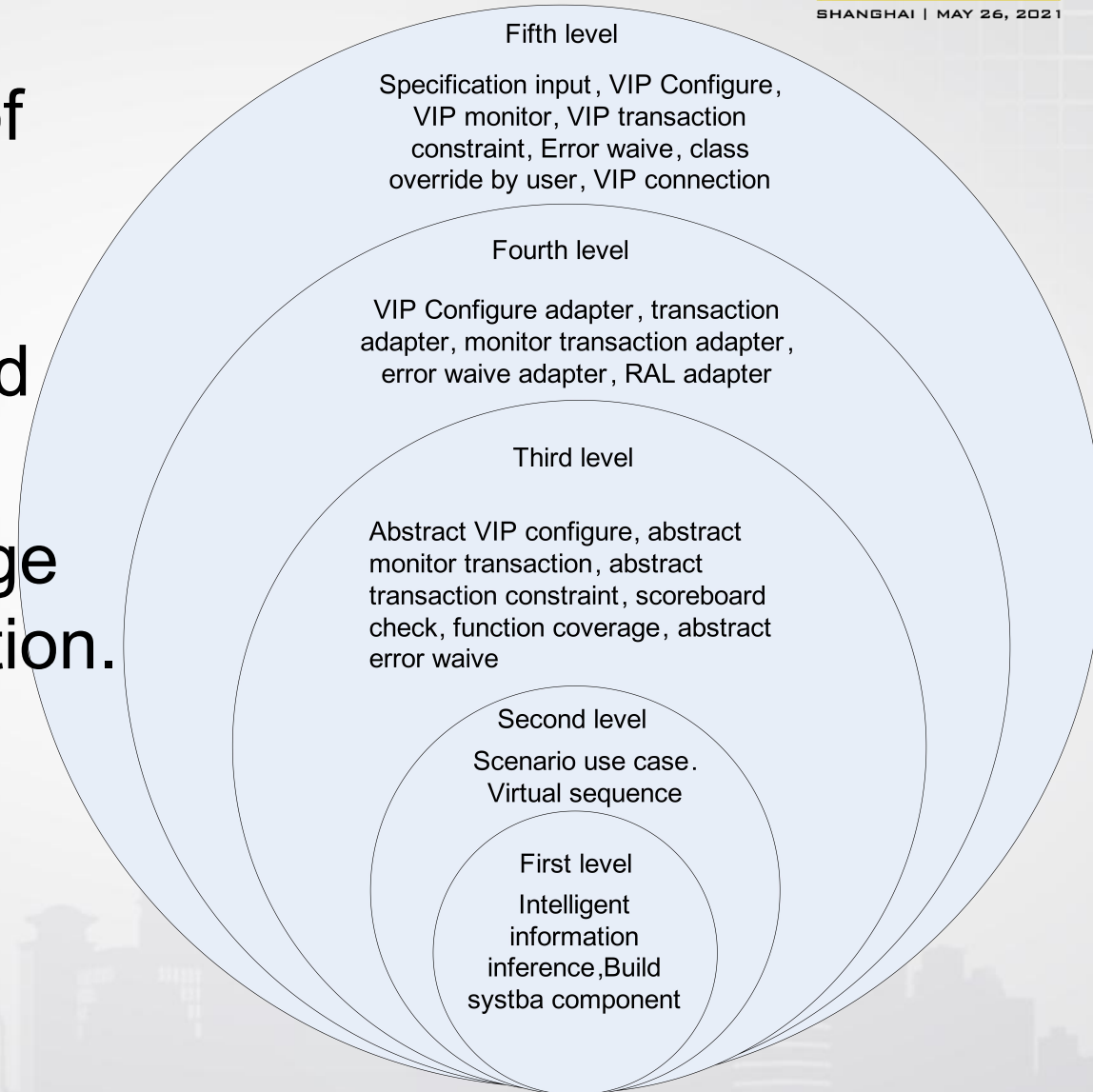
- Systba(System Test-bench automaton) based on SOC model
 - Common sequences and tests
 - Golden behavior model for check
 - Common coverage database for common features
- Systba test features
 - BusMatrix Connections
 - BusMatrix address decoder and remap
 - Channel Performance
 - SOC Low-power
 - SOC Anti-hang and so on

Contrast Between Systba and Others

Function	Cadence Interconnect TB	Systba
Auto hook VIP(APB/AHB/AXI/ACE...)	YES	YES
Modify VIP configure/constraint on fly	NO	YES
Auto generate common features sequence	NO	YES
Auto common features check Mechansim	NO	YES
Auto generate common features coverage	NO	YES
Easy bottom-to-up integration multi-subsys	NO	YES
Compatible C code	NO	YES
Easy Debug	YES	YES
Easy review	YES	YES
Isolation testbench with VIP vendor	NO	YES
Performance measurement	YES	YES

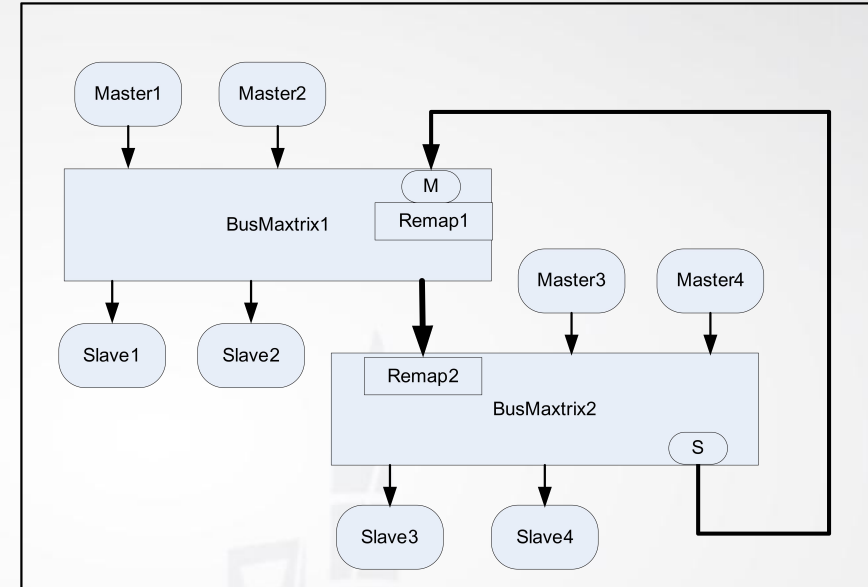
Systba Architecture

- To layer implementation of relationships intelligent inference, configuration ,stimulus and checker.
- To isolate test-bench usage with vendor's IP specification.



Systba's Common Pattern (1/2)

- Auto create Common Pattern
 - Special constraints based on known information;
 - Special transaction;
 - Special sequence;
 - Special test;



Slave Name	Function Description	Addr(Master1/2)	Addr(Master3/4)
Slave1	DMA Register	0x0000_0000:0x0000_0FFF	0x3000_0000:0x0000_0FFF
Slave2	UART Register	0x0000_1000:0x0000_1FFF	0x3000_1000:0x0000_1FFF
Slave3	IIS Register	0x0000_2000:0x0000_2FFF	0x3000_2000:0x0000_2FFF
Slave4	RAM	0x6000_0000:0x607F_FFFF	0x9000_0000:0x607F_FFFF

Master/Slave	Slave1	Slave2	Slave3	Slave4
Master1	YES	YES	YES	YES
Master2	YES	NO	YES	YES
Master3	YES	YES	NO	YES
Master4	YES	YES	YES	YES

Systba's Common Pattern (2/2)

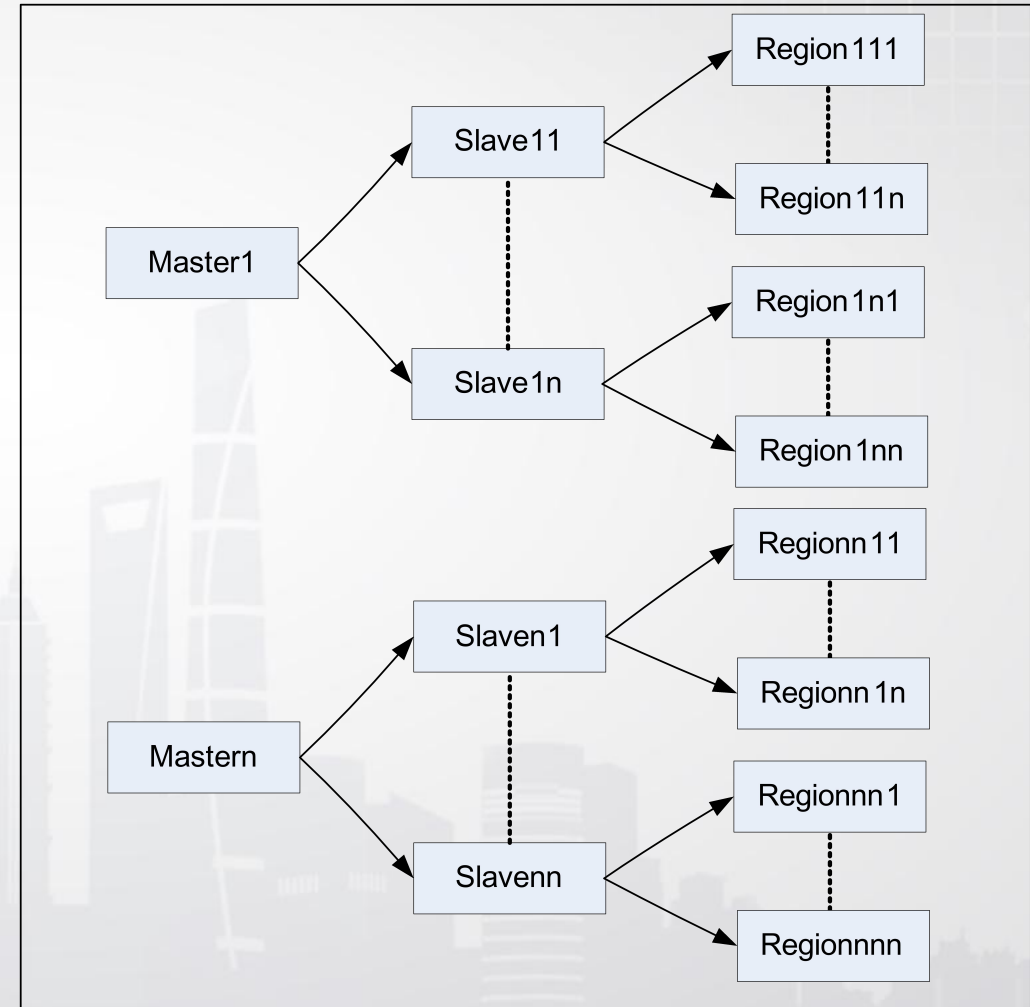
- Users can quickly and easily create their own sequence;
- Support for using regular expressions to represent model features;
- Easy add features to special master, special slave, or a special path;

```
Class no_send_burst_length16 extends systba_axi_trans;  
    function new();  
        super.new();  
    endfunction  
  
    constraint no_length16{  
        if(tr.direct==WRITE) {tr.burst_length != 16;}  
    }  
endclass
```

```
.....  
no_send_burst_length16 m_no_burst_length16;  
m_no_burst_length16 = new("No_burst_length16");  
m_resource_man.add_constraint(m_no_burst_length16, "master1->slave1->*");  
.....
```

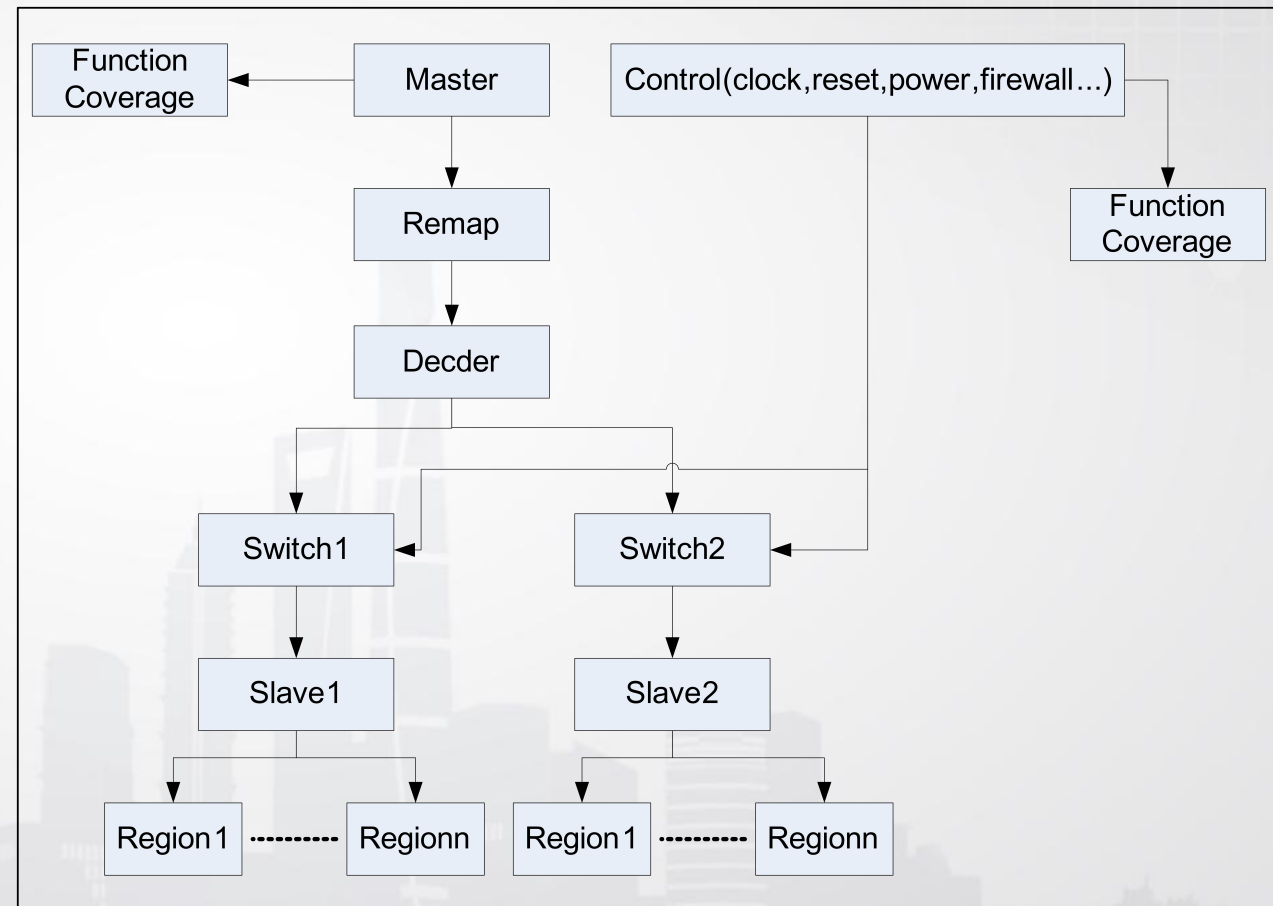
Systba Golden Behavior Model (1/2)

- The master knows that it can access the corresponding slave and region, similarly, also know it cannot access the slave and region.
- So can build access check mechanism based-on relationship of master to slave



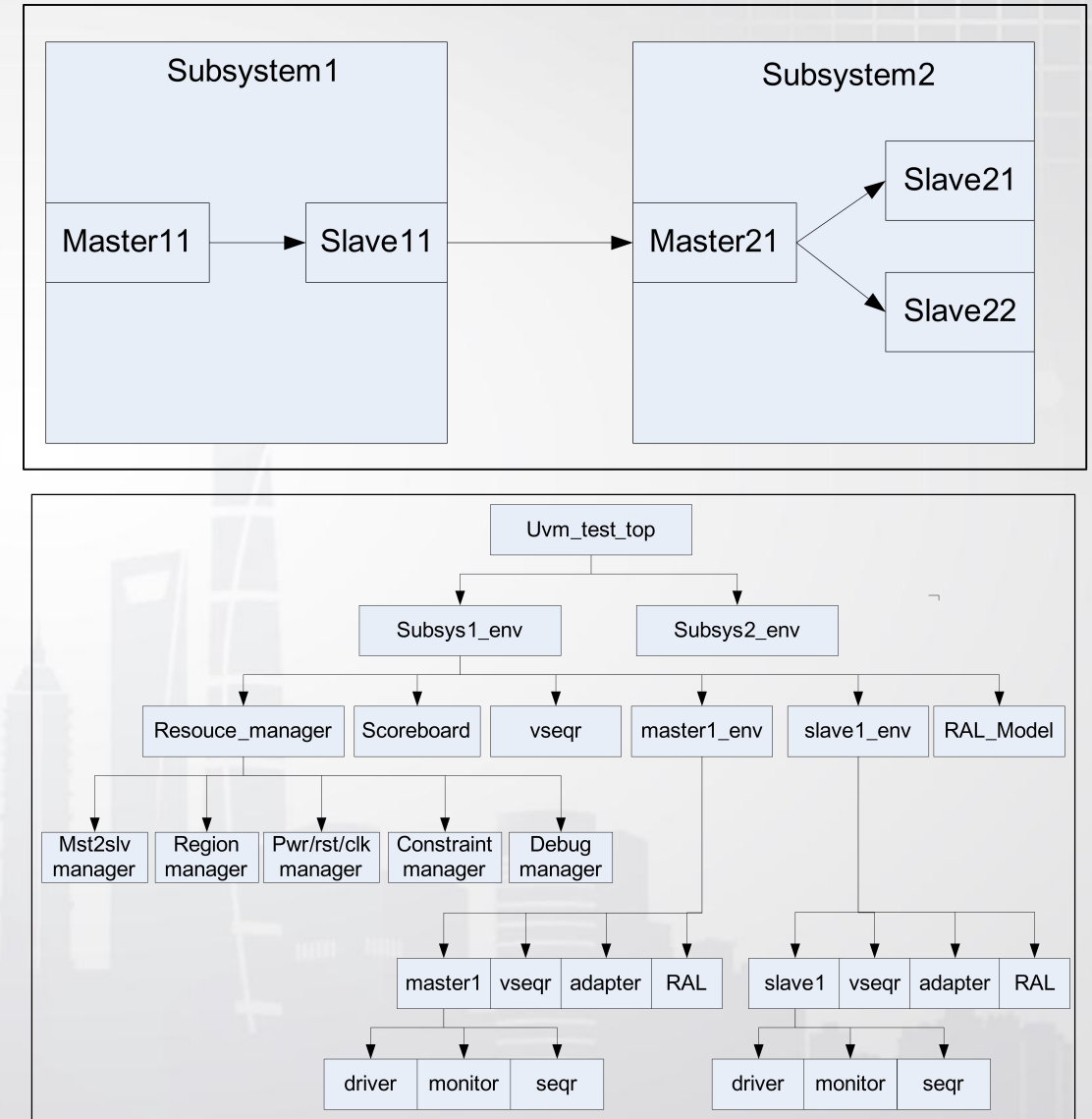
Systba Golden Behavior Model (2/2)

- Behavior model
 - Address remap feature;
 - Address decode feature;
 - Access switch feature;
 - Control mechanism, including clock, reset, low-power and firewall;



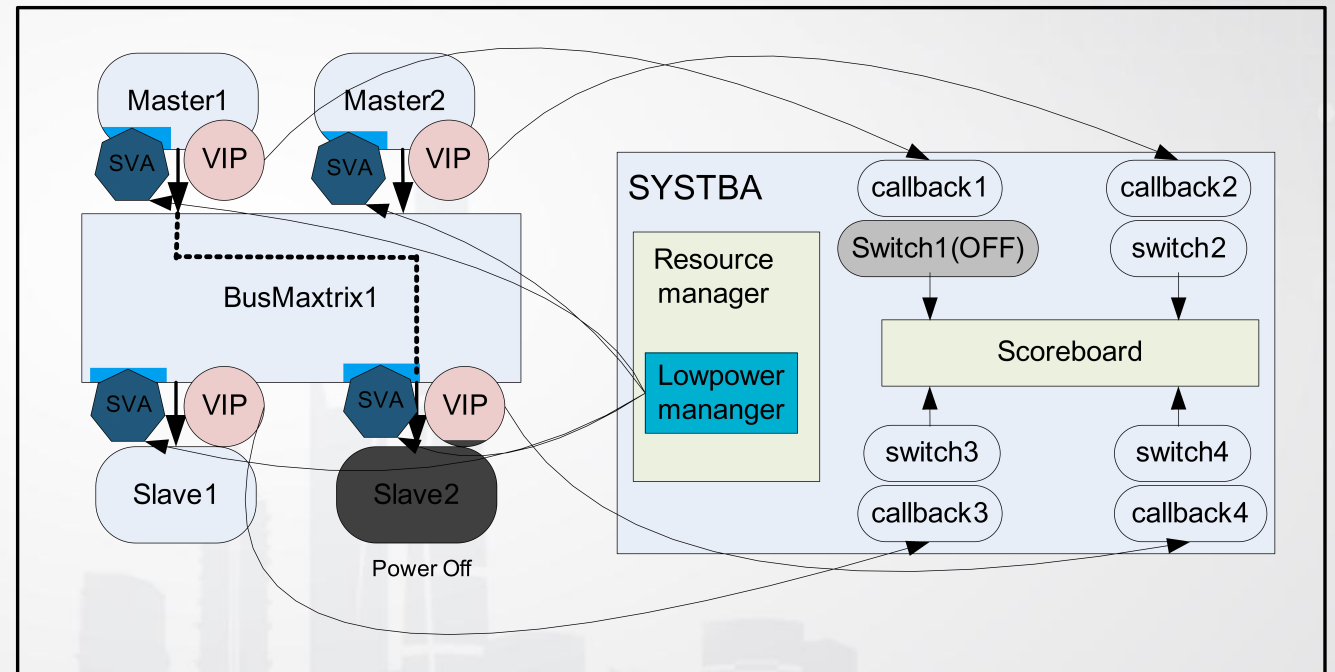
Bottom to Up Reuse Subsystem

- Support bottom to up reuse subsystem
- The structure of each subsystem is the same, so can easy cascade subsystem by Systba attribute configure
- The platform architecture of N subsystem is the same as a subsystem



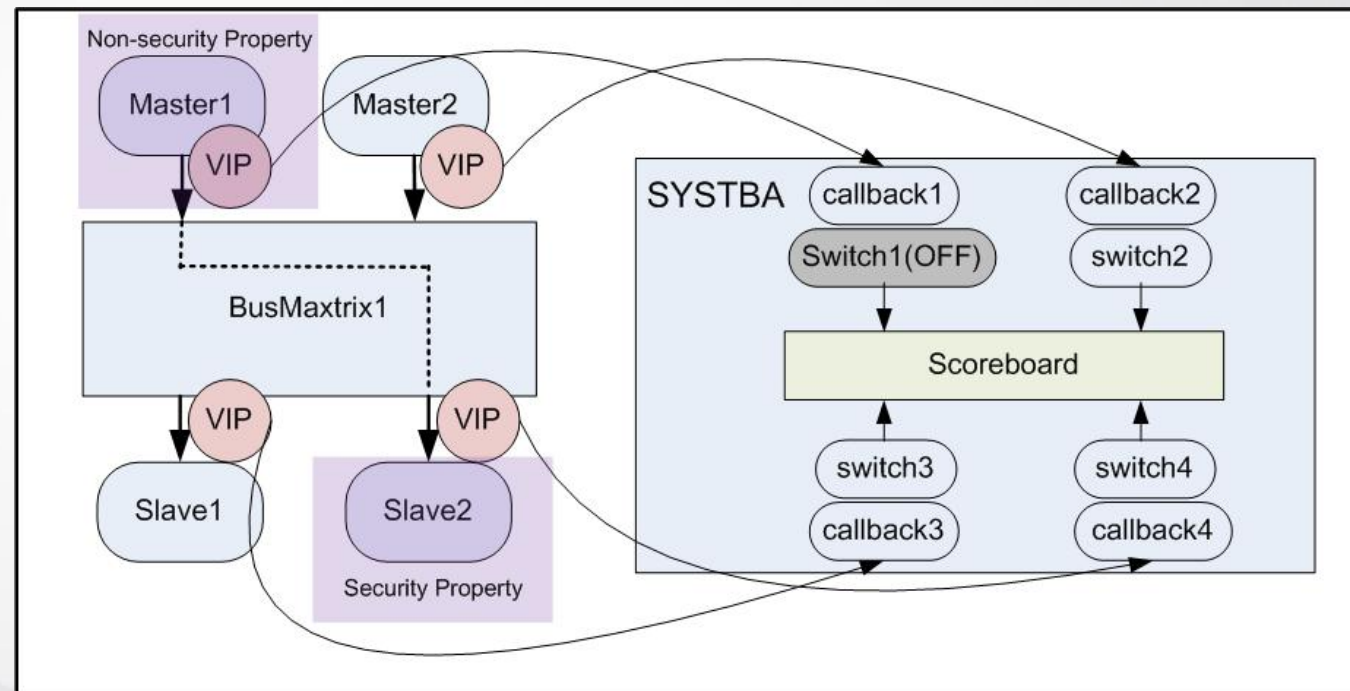
Low-Power Verification Demo by Systba

- SOC low-power include clock gating, power-gating and DVFS;
- SYSTBA intelligent generate low-power stimulus;
- SYSTBA low-power manager control clock/power SVA check enable and check low-power behavior;



Security Verification Demo by Systba

- SOC security scenario include security/non-security master access security/non-security slave;
- Check data-path behavior to meet security scenario;
- Check security information transform correctly;



Summary

- SOC common features and possibility of abstraction into model;
- Characteristics of SOC model, including architecture, test pattern and behavior model;
- Bottom to Up Reuse Subsystem ;
- Introduction verification demo for Low-power and security scenario;



Thanks you

