WINTECHCON

HVDM Solution for Random Verification of Microcontroller based Subsystems

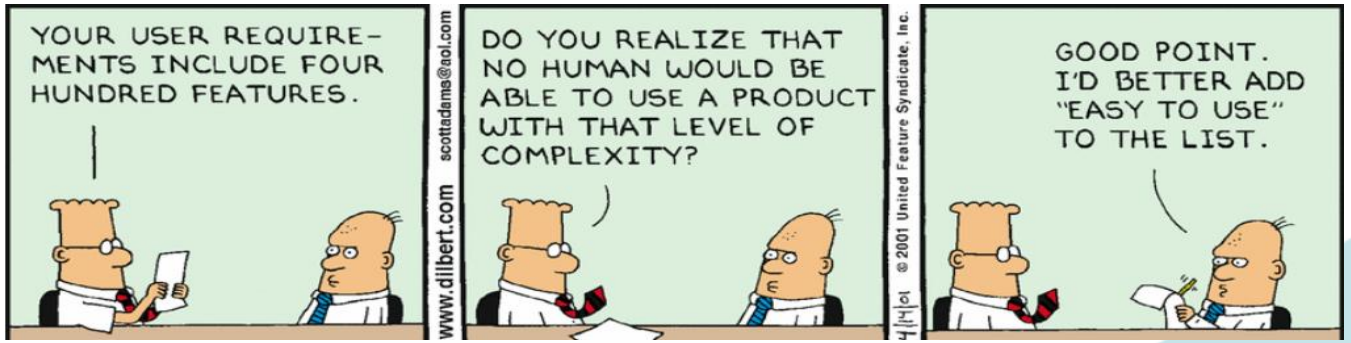
September 27, 2019

Ruchita Pathak
Vishal Dewan
Aravind Bhat
Intel

Problem Statement

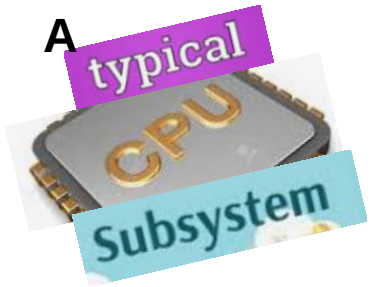
Scalable and Structured approach towards Verification of Microcontroller based Subsystem

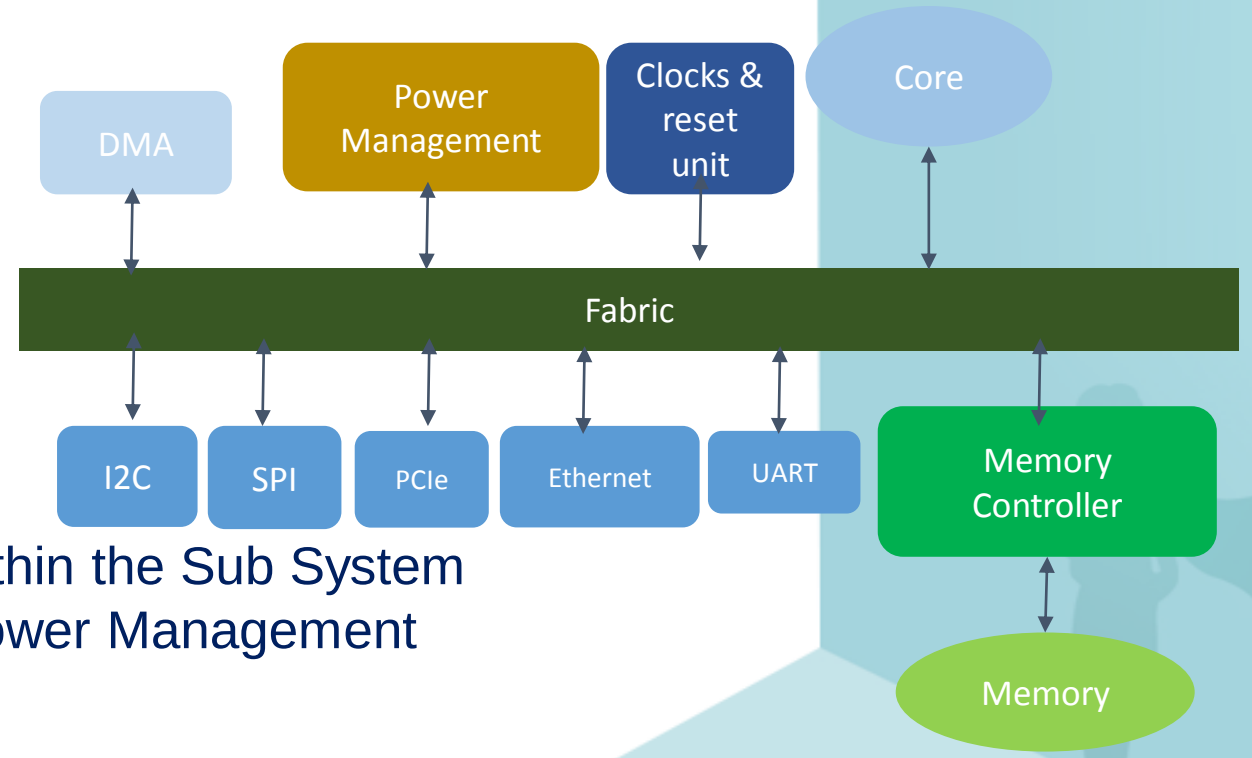
With growing Complexity of Chips, increase in number of features supported, operation at highest performance and the lowest power possible, lays down an unprecedented challenge for Verification to ensure the highest Quality standard in the shortest time possible to remain competitive in todays market !.



The proposed approach provides a scalable and structured Verification strategy and some tools to handle this complex Verification Challenge

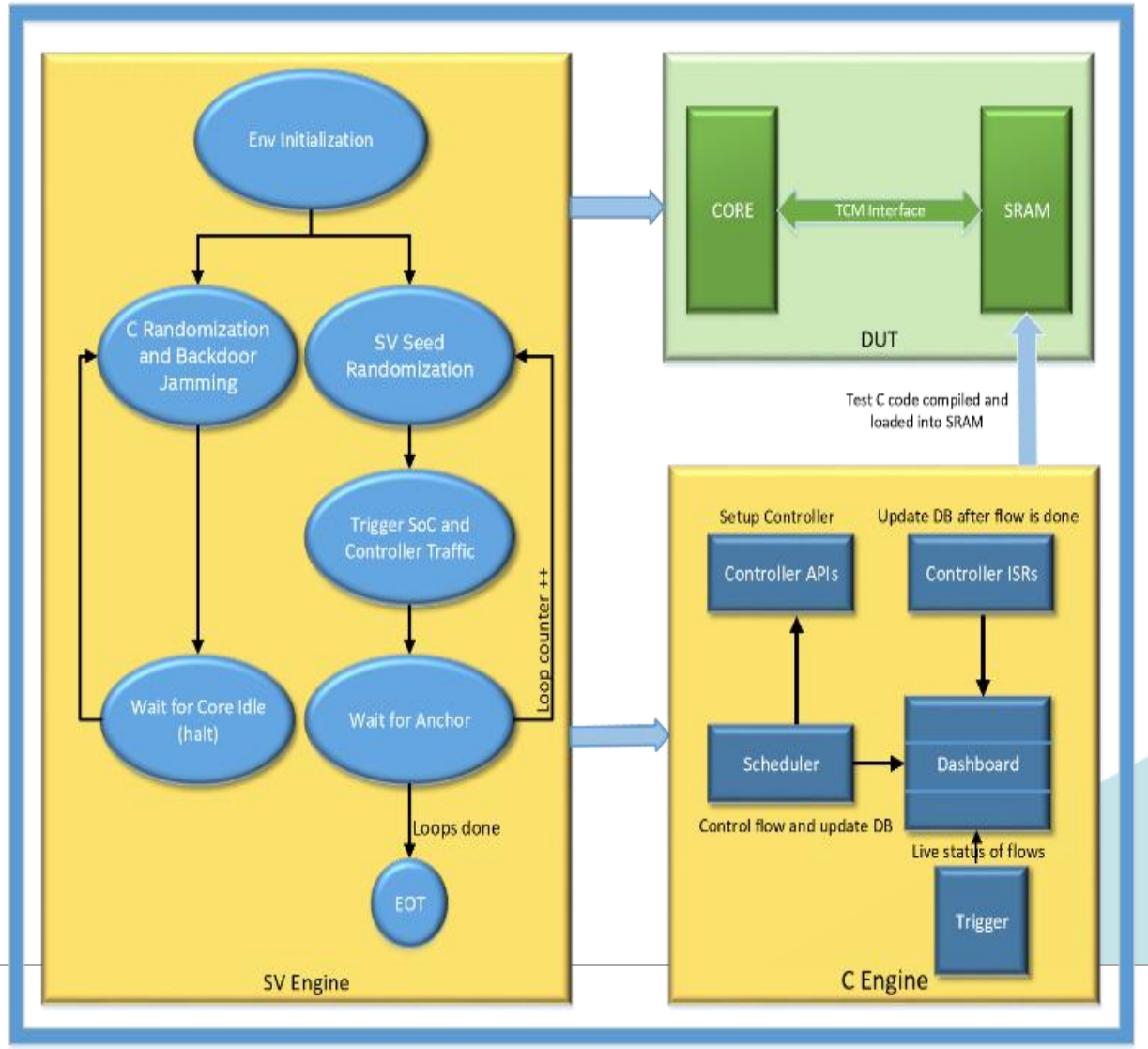
Problem Statement



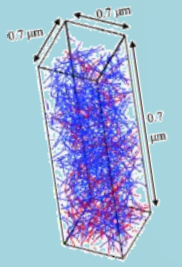


1. Enabling of all Functional Traffic within the Sub System
2. Enabling of all Clock Gating and Power Management Scenarios
3. Crossing 1 with 2
4. Debug ability of the Complex crosses
5. Ability to prove Coverage of the State Space
6. ***Ability to do all of the above in a Scalable and Modular manner for quick turn around and Future Proofing***

Proposed Methodology- Main Components



- "Randomizer" based on Constrained Random Verification(CRV)

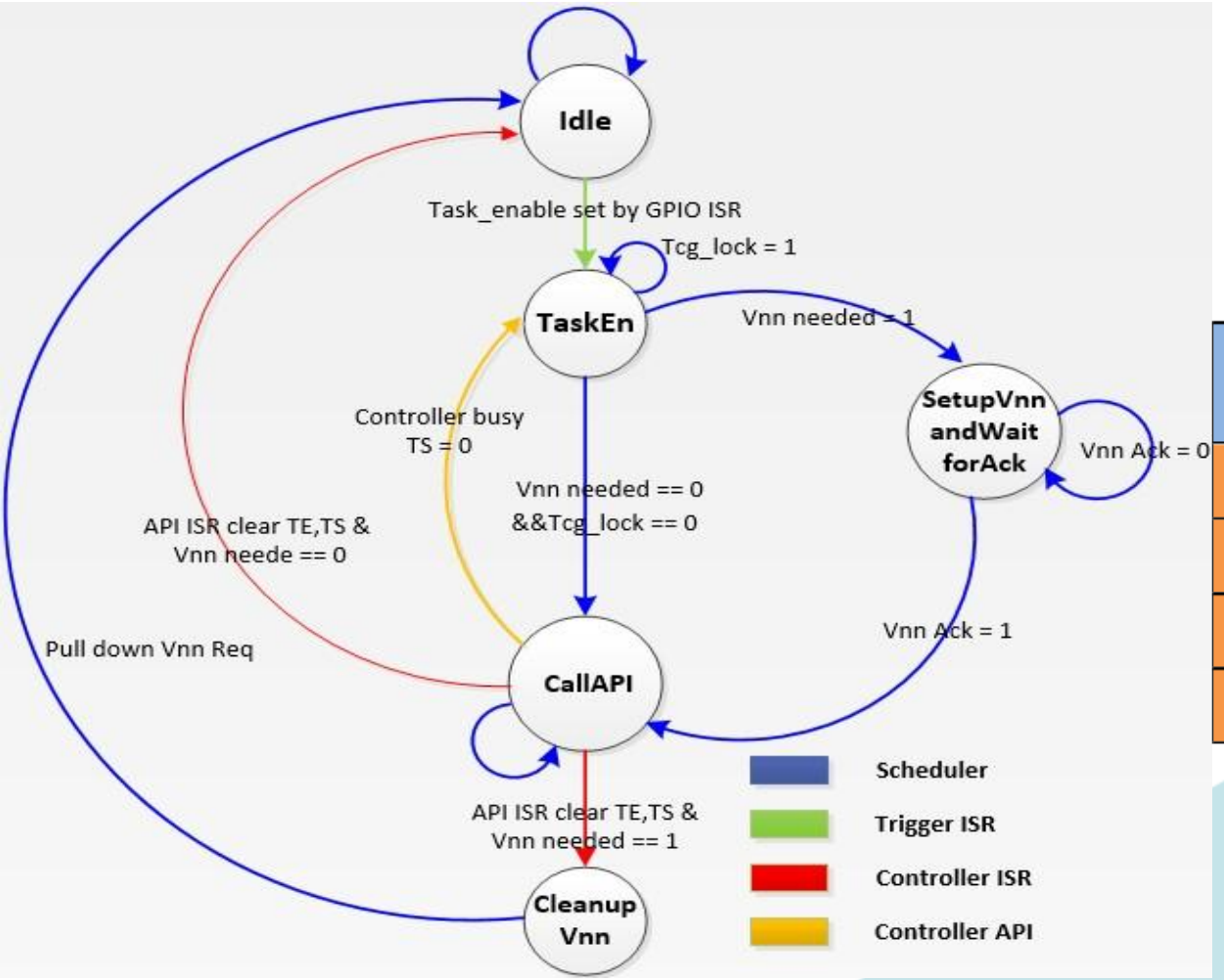


- "C Engine" to schedule all Core initiated Functional and Power Flows
- "SV Engine" to schedule all peripheral/ misc Functional Flows

- Use of "Anchors" to control Functional Flows to allow Entry to Low Power Modes

- Scalable "Dashboard" style Architecture to track all Functional and Low power Flows

Proposed Methodology- Dasboard Architecture



Project Specific Static bits **Flow Dynamic status bits**

AGENT Name	Vnn Dependency	TCG Dependency	Task Enable	Task Schedule	Vnn Req	Vnn Ack	TCG Lock	API Pointer
I2C0	0	1	1	1	0	0	0	API_I2C0
I2C1	0	1	0	0	0	0	0	API_I2C1
DMA0	0	1	1	0	0	0	0	API_DMA0
IPAPG	0	1	0	0	0	0	1	API_IPAPG

Proposed Methodology- Debug Monitor

-->DASHBOARD MONITOR<--

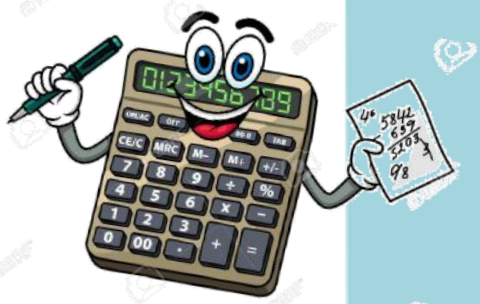
PROJECT SPECIFIC ENTRIES					DASHBOARD DYNAMIC ENTRIES			
TIME	TASKNAME	VNN DEP	TCG DEP	TASK EN	TASK SCH	VNN REQ	VNN ACK	TCG LOCK	DB STATE
1366857	AGENT_HPET0	0	0	0	0	0	0	0	IDLE
105492	AGENT_ISH2HOST_HOST0	1	1	0	0	0	0	0	IDLE_VNN
1232242	AGENT_HPET1	0	0	0	1	0	0	0	TASK_EN
104974	AGENT_DMA_CH0_INTR	0	1	1	1	0	0	0	CALL_API
105772	AGENT_ISH2NONHOST	1	0	1	0	1	1	0	SETUP_VNN_ACK_RCVD
105140	AGENT_DMA_CH1_EXTR	1	1	1	1	1	1	0	CLEANUP_VNN

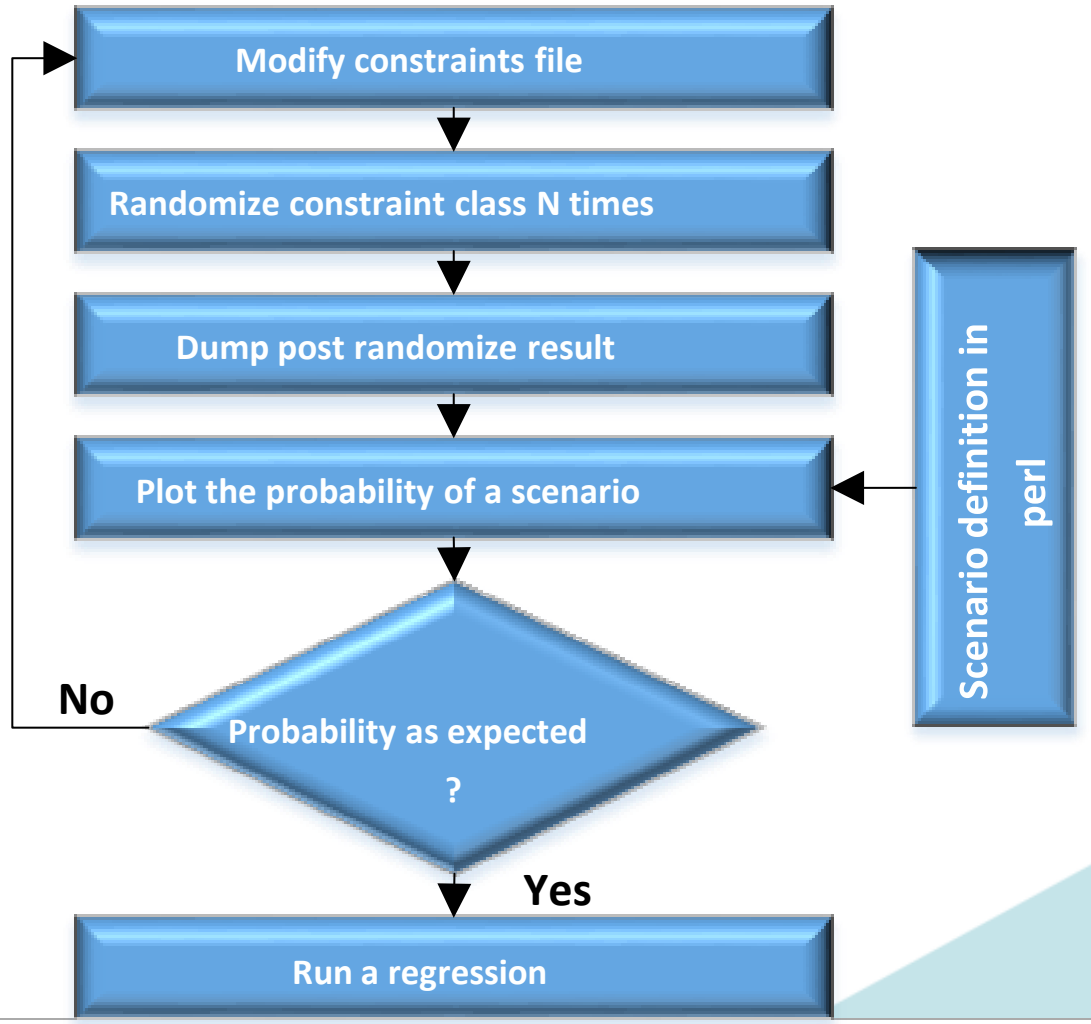
->SV Traffic DashBoard updated<-|-->Iteration Number: 1<--|-->Relaunch Enabled: 0<--|-->Max Relaunch cntr: 3<--

Thread num	Thread en	Th Delay	Th relaunch	Start TS	End TS	Duration	Status	Done Once	Relaunch cntr
->Primary Traffic Threads<									
GPIO	1	29	NO	791626	791626	0	0	1	1
IPIAPG	0	629	YES	0	0	0	0	0	0
HOST_POSTED	0	1587	YES	0	0	0	0	0	0
HOST_SRAM	1	296	YES	8132552	0	0	0	0	0
HOST_AON	0	581	YES	0	0	0	0	0	0
->Secondary Traffic Threads<									
->HOST SRAM Traffic Threads<									
0	0	3	YES	0	0	0	0	0	0
1	1	5	YES	8132667	0	0	1	0	0
2	0	8	YES	0	0	0	0	0	0

All Enabled Threads Done atleast once: 0

Proposed Methodology- Probability Calculator





Random Constraint Solver Output:

- Weighted Probability of all Low Power States: C1, C2, C3, C4
- Weighted Probability of all Low power sub-State: D1, D2, D3, D4
- Weighted Probablity of L1, l2, L3 Memories in Retention Vs PG'ted
- Weighted Probablity of "Anchor" allowing Full PG entry Vs PG Entry abort
- Weighted Probablity of all different kinds of Functional wakes: Timers, GPIO etc

Probability Calculator can help you find the Probablity of:

- C3 Low Power State, with D2 Low Power sub-State, L1 memory in Retention, L2 memory PG'ted, Full PG Entry aborted by a GPIO wake

Observed Results

Areas of Improvement	Traditional Method	Proposed Solution	%Improvement	Justification
Integration of new IP flow	~3days	~1day	66%	Scalable and structured framework
Debug time	~2days	~1day	50%	C and SV engine monitors
Functional Coverage Targets	~3days	~2days	33%	Flow controllability and Static probability calculator

Conclusions

The proposed verification Methodology provides:

- A Scalable and Structured verification Strategy to target any Microcontroller Based subsystem with Power Management Flows
- Ease of Debug for Faster Turn around
- Wider State Space Coverage
- Better Control on the stimulus to help target Coverage Gaps

References

- Hu Zhaohui, A. Pierres, Hu Shiqing, Chen Fang, P. Royannez, Eng Pek See,Yean Ling Hoon, “Practical and efficient SOC verification flow by reusing IPtestcase and testbench”, 2012 International SoC Design Conference (ISOCC),IEEE, 4-7 Nov. 2012.
- Sainath Karlapalem, Shashank Venugopal, “Scalable, Constrained Random Software driven Verification”, 17th International Workshop on Microprocessor and SOC Test and Verification, 2016.
- I. Silas, I. Frumkin, E. Hazan, E. Mor, G. Zobin, "System-level validation of the intel Pentium M processor", *Intel Technol. J.*, vol. 7, no. 2, pp. 38-43, May 2003.