UVM Based Mixed-Signal Verification of a Display PMIC Designed for OLED Display Applications

Vijay Kumar & Adnan Malik
Samsung Semiconductor India Research

accellera SYSTEMS INITIATIVE

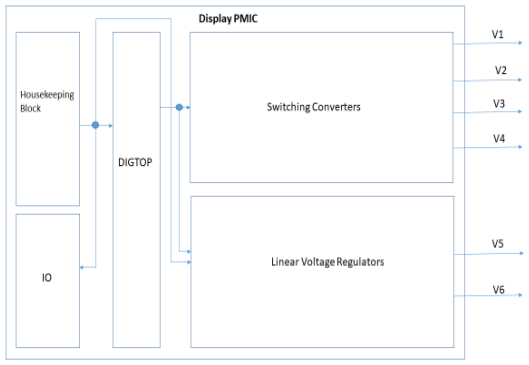
2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES
SAN JOSE, CA, USA
FEBRUARY 27-MARCH 2, 2023

INTRODUCTION (or REQUIREMENTS)

- The Display PMIC consists of **Switching Converters, Housekeeping Circuits, viz., Reference voltage and Bias current generators, On-chip LDO and Clock generation unit**
- Contains Protection logic circuits to **guard it against various fault scenarios** such as TSD, UVLO, SCP, IVP, OVP, UVP, OCD, SSD
- Comes with two slave addresses, each with its own set of register banks
  - One set of registers programmable by I2C protocol using the serial clock and data lines
  - Another set of registers programmable by a separate set of lines
  - **GPIO input** decides which set of lines are used for I2C read and write
- Finds its applications in **OLED power supply for mobile devices, camcorders, digital cameras and multimedia players**

Main Features of the Display PMIC are:

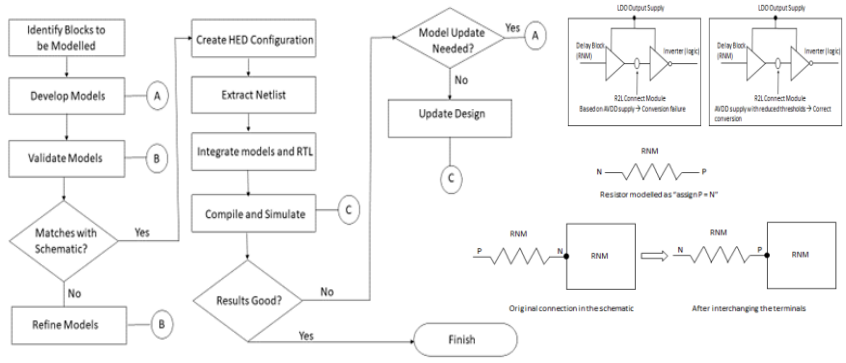
- Adaptive power-on sequence & shutdown mode
- Fast discharge of outputs after shutdown
- Output high-Z by Always-On Display (AOD) mode
- Single-byte and multi-byte write & read through one of the I2C interfaces
- Soft-start with inrush current limitation
- Thorough protection by triggering interrupts against various fault scenarios



OBJECTIVES

- **To compile the major challenges faced in verifying the PMIC in a UVM based mixed-signal verification environment (MSV)**
- **How we addressed those challenges?**

Major challenges were:
(a) Handling interaction of signals between blocks at different abstraction levels (logic and real number) under different supply domains
(b) Mechanism to cover fault scenarios in a real number model (RNM) based Digital and Mixed-Signal (DMS) verification environment with UVM, that are typically addressed in the transistor-level abstraction
(c) Handling mixed-signal specific driver conflicts arising out of logic and real drivers, modelling style, etc.



RESULTS

SV-UVM ASSERTIONS TO CHECK THE STATUS REGISTERS AND CONVERTER OUTPUTS TO DISPLAY THE PASS/FAIL STATUS
Enabled regression by adding **SV-UVM assertions** in each test case
- Checks for the status registers and converter outputs to display the pass/fail status

```
p_sequencer.i2c_intf.I2C_WRITE (SLAVE_ADDR, REG1, VALUE1); //Enabling Converter 1 (Converters 2 & 3 are auto-enabled)
#t_converter_1_enable; // Converters 1-3 enable wait time
p_sequencer.i2c_intf.I2C_WRITE (SLAVE_ADDR, REG1, VALUE2); //Enabling converters 4 & 5

#t_converter_4_5_enable; //Converters 4 & 5 enable wait time and display all converter output values
$display("%f %f %f %f %f", V_Converter_1, V_Converter_2, V_Converter_3, V_Converter_4, V_Converter_5);

//POK Registers Check
p_sequencer.i2c_intf.I2C_READ (SLAVE_ADDR, REG2, i2c_rdata);
$display("POK_STATUS at %f = %h", $time, i2c_rdata);
if(i2c_rdata == expected_data)
`uvm_info(get_full_name(), "POK Good\n", UVM_LOW)
else
`uvm_error("MAIN_CHIP", "POK Bad\n");
```

UVM TEST CASES FOR TRIGGERING FAULT SCENARIOS IN THE RNM MODELS
Since we use RNM and not the transistor level schematic, the fault scenarios may not occur in the simulation as we do not model those scenarios in the RNM. So in order to run test cases that check for the protection logic against such scenarios, we had to set the corresponding flag for each fault condition and see if the fault status registers and interrupt requests (IRQB) are triggered accordingly.

```
p_sequencer.i2c_intf.I2C_WRITE (SLAVE_ADDR, REG1, VALUE1); //Enabling Converter 1 (Converters 2 & 3 are auto-enabled)
#t_converter_1_enable; //Converters 1-3 enable wait time
p_sequencer.i2c_intf.I2C_WRITE (SLAVE_ADDR, REG1, VALUE2); //Enabling converters 4 & 5
```

RESULTS

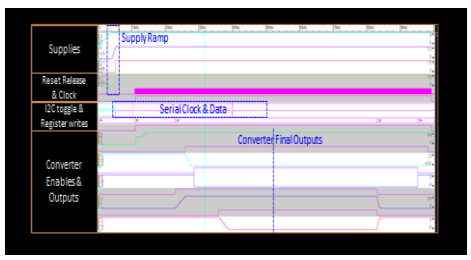
```
#t_wait_time_for_flag_set; //Wait time before setting the flag
uvm_hdl_force("top.Main.I_CONV_1.OVP", 1'b1); //Triggering OVP condition for Converter 1

//OVP Register Check
p_sequencer.i2c_intf.I2C_READ (SLAVE_ADDR1, REG_OVP, i2c_rdata);
$display("OVP_STATUS = %h", i2c_rdata);
if(i2c_rdata ==  expected_data) //Checking the OVP register status
`uvm_info(get_full_name(), "OVP interrupt generated", UVM_LOW)
else
`uvm_error("MAIN_CHIP", "OVP interrupt generation failed");

#t_wait_time_for_flag_release; //Wait time before releasing the flag
uvm_hdl_force("top.Main.I_CONV_1.OVP", 1'b0); //Releasing OVP condition for Converter 1
```

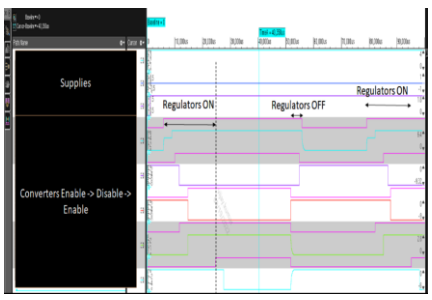
Once the OVP condition is triggered for Converter 1, all the converters shut down during which we check the status registers and display pass/fail status as shown above. After this, we re-enable the converters to see if they would power-up, followed by power down of all the converters.

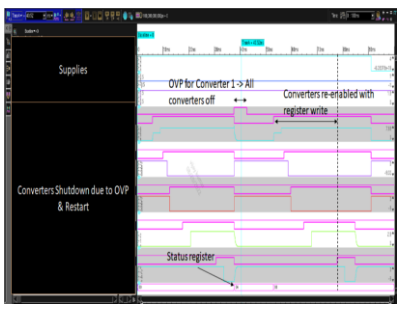
```
//Power Up
p_sequencer.i2c_intf.I2C_WRITE (SLAVE_ADDR, REG1, VALUE1); //Enabling Converter 1 (Converters 2 & 3 are auto-enabled)
#t_converter_1_enable; //Converters 1-3 enable wait time
p_sequencer.i2c_intf.I2C_WRITE (SLAVE_ADDR, REG1, VALUE2); //Enabling Converters 4 & 5

//Power Down
#t_wait_time;
p_sequencer.i2c_intf.I2C_WRITE (SLAVE_ADDR, REG1, VALUE1); //Disabling Converters 4 & 5
#t_converter_1_enable;
p_sequencer.i2c_intf.I2C_WRITE (SLAVE_ADDR, REG1, VALUE0); //Disabling Converters 1-3
```

CONCLUSIONS

- 142 test cases were completed on time with this approach spanning across various scenarios:
  - Power-on and power-off
  - Write and read from dedicated clock and data lines or separate set of lines using I2C protocol
  - Dynamic Voltage Scaling
  - Protection logic
  - Pin connectivity checks between analog and digital
  - Level shifter checks across supply domains
- 45 VAMS wreal models were developed and validated against the schematic
- 7-8 weeks TAT for the whole activity







REFERENCES

1. Kiyeong Kim, Hwan-woo Shim, Chulsoon Hwang, "Analysis and Solution for RF Interference Caused by PMIC Noise in Mobile Platforms", IEEE Transactions on Electromagnetic Compatibility, June 2019
2. Abdullah Abdulslam, Patrick P. Mercier, "A Symmetric Modified Multilevel Ladder PMIC for Battery-Connected Applications", IEEE Journal of Solid-State Circuits, December 2019
3. Chunlei Shi, Brett Walker, Eric Zeisel, Brian Hu, Gene McAllister, "A Highly Integrated Power Management IC for Advanced Mobile Applications", IEEE Custom Integrated Circuits Conference, September 2006
4. Nicola Dall'Ora, Sara Vinco, Franco Fummi, "Functionality and Fault Modeling of a DC Motor with Verilog-AMS", IEEE 18th International Conference on Industrial Informatics, July 2020
5. Constantina Tsechelidou, Nikolaos Georgoulopoulos, Alkiviadis Hatzopoulos, "Design of a SystemVerilog-Based Sigma-Delta ADC Real Number Model", 22nd Euromicro Conference on Digital System Design, August 2019
6. Vijay Kumar, Shrikant Pattar, Yaswanth Chebrolu and Vinayak Hegde, "Harnessing SV-RNM Based Modelling and Simulation Methodology for Verifying a Complex PMIC designed for SSD Applications", DVCON INDIA 2022
7. Nikolaos Georgoulopoulos, Alkiviadis Hatzopoulos, "Real number modeling of a flash ADC using SystemVerilog", Panhellenic Conference on Electronics and Telecommunications, November 2017
8. Nikolaos Georgoulopoulos, Alkiviadis Hatzopoulos, "Efficiency evaluation of a SystemVerilog-based real number model", 7th International Conference on Modern Circuits and Systems Technologies, May 2018
9. Nikolaos Georgoulopoulos, Athanasios Mekras, Alkiviadis Hatzopoulos, "Design of a SystemVerilog-Based VCO Real Number Model", 8th International Conference on Modern Circuits and Systems Technologies, May 2019

Add acknowledgements, organization involved, logo's contact infomation here

1. Manish Parmar, Manish Goel, Balajee Sowrirajan, Deepa Devendran, Shyam Prasad Venkata Voleti, Divij Kumar and Vinayak Hegde of **Samsung Semiconductor India Research**
2. YuBeom Jeon of **Samsung Electronics**

SSIR Samsung Semiconductor India R&D Center