Methodology for Mixed Mode Design & Verification of Complex SoCs

ChipEx2013

Jan Pleskac, S3 Group

Agenda

Introduction

Mixed Signal SoC examples

Analog vs. digital design flow

Lessons from Mixed Mode Design & Verification projects

Conclusion

Mixed Signal SoC examples

RF analog Front End with digital signal pre-processing

TX – DAC, Filter, Mixer, VGA, PA Driver

RX – LNA, Mixer, Programmable Filter, ADC

PLL, LDOs, Auxiliary ADCs, DACs, Temp sensor

RX/TX data interface, IC programming interface, GPIOs

Digital filtering and DSP, Power management

Analog blocks calibration, testing

Power management IC

POR, Reset/Clock distribution, ADCs, Temperature monitors/alarms, Programming interface

Smart Sensors

Ultra low power, Closed loop compensation/linearization control, Signal processing, uController

Mixed Mode Design & Verification

Architecture

Decide on TOP implementation flow – “analog on top” vs. “digital on top”

Early IC Floor Plan – package constrains (analog sensitive pads)

Design partitioning

– signal flow, operation modes, mission critical, power domains

– leveraging existing design data (internally developed building blocks)

– IP blocks selection

Identify Analog/Digital critical parts

– mixed mode co-simulation candidate – calibration, control loops, power sequencing

Top-level Functional Verification – co-simulation

– Integration checks on functionally verified analog and digital blocks

– Functional Requirements sign-off

Analog vs. Digital Characteristics

Analog

Bottom-Up

– focus on block level design (schematic, models, performance)

– GUI based tools, visual inspection

– Sub-systems & TOP integration

– Block/Sub-Systems Model creation for top verification (functional models)

– Power Estimations accurate early

– Hand Layout

Digital

Top-Down

– block level design and early integration (reuse of verification environment)

– Text based, regressions, self-checking

– methodology and standards (strong reuse of design patterns or blocks, coding style)

– Active specification – (IP-XACT registers description)

– Accurate power estimations requires simulation (post layout + extractions)

– Tool based automatic layout

Lesson #1

Analog and Digital team communication

Problem: Two different worlds, different mindset.

Solution:

Clear and consistent naming on IC architecture and interface signals from Day #1.

High level functional/architecture specification.

Focus on USE-CASEs – common language.

Co-Working rather then forcing someone to work in unfriendly environment.

Lesson #2

Respect the difference in flow/domain

Problem: Flows are disconnected and way different.

Solution:

Use appropriate tool or view for given task (e.g. Modeling trade-off accuracy vs. speed).

Implement small digital macros for non-critical analog routing (e.g. configuration bus, decoders).

Uniform data formats for use in Analog/Digital standalone env., especially when 3rd party tool is involved (e.g. Matlab data generators, files).

Lesson #3

Implementation phasing

Problem: It is not possible to finish digital controller until analog block is designed, but you have to start someday… Late changes happen. Tight layout or performance issues may require functional modifications (Power architecture, Filtering changes).

Solution:

Start to implement core functionality and structural parts (state machine, interval timers).

Use simplified functional models of analog block, or create behavioral representation of digital block rather then creating RTL in early stages.

Run several implementation iterations, each step needs consistent views. Final implementation "just configures soft core RTL".

This require multiple runs of full digital flow RTL2-GDS2, important for change impact analysis (area, power, routing).

Keep working top-level schematic – co-simulation on netlist with "dummy" views.

Lesson #4

Reuse of digital verification environment

Problem: Digital standalone verification covers majority of analog scenarios, but with rubbish data.

Solution:

Build digital standalone verification test environment aware of analog content (constrain meaningful data for USE-CASEs, Min-Mid-Max VGA code, ADC start-up, etc.)

Digital verification techniques like assertions, coverage are directly applicable to analog design.

Interface monitors can be re-connected from digital boundary to analog blocks instances.

Top-level mixed mode verification is just rerun of existing digital test.

UVM based digital verification environment is easily reconfigurable for co-simulation environment.

Conclusion

Mixed mode design & verification is iterative process

Top – Down approach is must for parallel execution

Modeling at early design stages helps with implementation and verification

Effective reuse of digital verification environment require top-level verification planning

Mixed-Mode Functional Verification can be a DIGITAL task executed in ANALOG environment

Mixed-Mode Design & Verification is team work – COMMUNICATION matter

Thank You!

jan.pleskac@s3group.com