Extending Constrained Random Verification to mixed-signal Automotive Power Devices using a non-stationary Markov process

Thomas Nirmaier, Manuel Harrant, Georg Pelz

Infineon AG
Am Campeon 4, 85579 Munich - Neubiberg
e-mail: thomas.nirmaier@infineon.com

1. Abstract

Automotive Power Devices range from low pin-count devices, like power switches, up to complex system control SOCs for the airbag and brake system, that may contain multiple µC cores embedded with multiple sensor and communication interfaces, often all integrated on a single die. A strong trend towards the integration of digital functionality on power electronics devices is visible due to ever increasing monitoring and safety requirements. The combination of analog power electronics and increasing digital functionality leads to an explosion of the verification space that cannot be covered by traditional directed testing anymore.

In order to debug the wide range of mixed-signal functionality we propose to extend Constrained Random Verification methods, which are well known in digital verification. These methods are based on the principle "do anything the specification allows" and "do anything the application could possibly do". In this paper we show how to extend constrained-random digital verification to mixed-signal functionality, including test coverage for embedded switches, regulators, transceivers, sensor interfaces and other mixed-signal circuitry. There is no single approach that fits all needs, but several methods need to be applied for mixed-signal instead, including test pattern generation using non-stationary Markov processes to address random-resistance, random-power-up tests, post-processing on captured digital or analog results from ATE or other equipment. We show several case studies from devices of different complexity, starting from power switches, up to complex airbag control-system devices and how results link to requirements engineering.

2. Introduction

Latest automobile generations contain an ever an ever increasing electronic content due to technology drivers like energy efficiency, communication and safety. Several classes of electronic devices are built into cars, ranging from infotainment devices with moderate quality requirements to safety critical devices in restraint and brake systems. These devices have to satisfy high safety standards with respect to production quality, often referred to as "zero defect", but also with respect to functionality, thereby increasing the verification effort.

3. Constrained Random Verification

Constrained Random Verification is, besides directed testing, the most well known method for functional verification, both on pre- and post-Si side.

Any method involving algorithmic test generation with the help of a random selection process that is guided by constraints can be considered a CRV method, but we focus this paper on constrained-random tests for the power-up procedure and for constrained-random test pattern generation, as they have proven to be the most beneficial to guarantee functional device robustness.

4. Random Power-Up Tests

The power-up phase is a critical region of operation due to several reasons. First the shape and sequence of powering-up one or more supplies is often poorly specified, which leaves open a huge gap for verification. Most specifications only state that the device should be operational a specified time after "power-stable". In the meantime power supply can and will behave in an arbitrary manner, including intermediate power-down, spikes both in positive and negative or power glitches.

Mixed-signal and analog blocks, like charge pumps, oscillators and band-gap references are often sensitive to the power-up phase. Designs often implement an internal power-on sequence to switch on parts of the chip in exactly the right order, but interruptions in the power or spikes can partially or block-wise interrupt the predefined

power-up sequence leading to meta-stable internal states. Often an external power-on-reset is used to reset the device to a defined state, but not all circuitry may be resettable.

On the other hand, due to missing constraints given by most specifications, the power-up phase is the least covered region of operation by pre-Si verification. Therefore missing power-up robustness leads often enough to application failures.

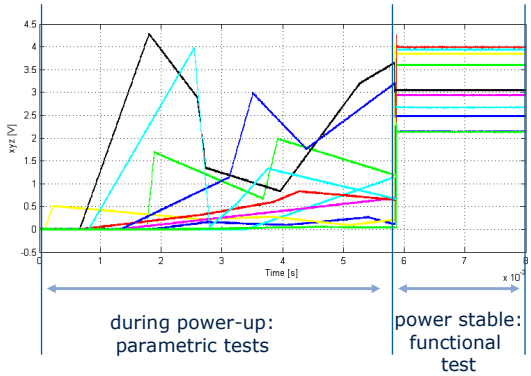

Figure 1: Power-Up test regions: During power-up with arbitrary supply shape, parametric tests may need to be applied. In any case the DUT must be operational in the power-stable phase, where extended functional tests should be applied.

We apply random power-up tests on ATE and lab equipment. From a verification point-of-view power-up can be divided in two distinct phases, see also Fig. 1. During the ramp time the DUT is not expected to be functional, but often enough other constraints like maximum power consumption, output drivers off, latch-up resistance, etc. must be guaranteed for. Any application may ramp-up in a different way, including voltage over/-undershoots, slope reversals, spikes. A way to cover such open conditions is to apply a random process to generate random arbitrary power-supply waveforms, only limited by maximum device ratings. During the second phase, the power-stable phase, the device should be operational, no matter how the first phase was, so extended functional testing at the full supply voltage range within specification is recommended.

5. Constrained-Random Test Pattern

While basic functionality of the DUT is most often tested for using directed test cases, functional robustness can only be guaranteed with the help of algorithmic test pattern generators based on the constrained random principle. This path is well known for digital post-Si verification, where commercial tooling like *e*, Specman, System- Verilog exists.

We focus on mixed-signal devices; therefore we extend the well known Markov-based test pattern generator principle with mixed-signal elements, testing for power-switching, transceiver, voltage regulator and other mixed-signal-behaviour.

As mentioned before, we use a layered model starting from Markov-based state generation, followed by timing randomization and conversion to pin-level test pattern. We will describe these steps hereafter, also see Fig. 2 for an overview.

6. Markov-based state sequence generation

A Markov model is a directed graph with weighed edges. For most digital designs, the edges represent commands to the device-under-verification. For Automotive Power devices, a simple Markov model, based on instructions to the DUT does not suffice, because many mixed-signal events may cause state-changes and must be included in the model. This may be events related to temperature, supply voltage, over-current etc., which may occur fully asynchronous to each other. Besides, special safety features are often implemented on Automotive Power devices, that check for certain timings. Here, a hardware watchdog must be regularly triggered after a certain time or timing window. In general the resulting Markov models contain many timing-driven transitions.

Direct observability may also not always be given for any possible mode-transition after each transition, but only after a larger sequence of transitions. For mixed-signal Constrained-Random Test generation we start from the approach described in [6, 8, 10] and extend it to mixed-signal events.

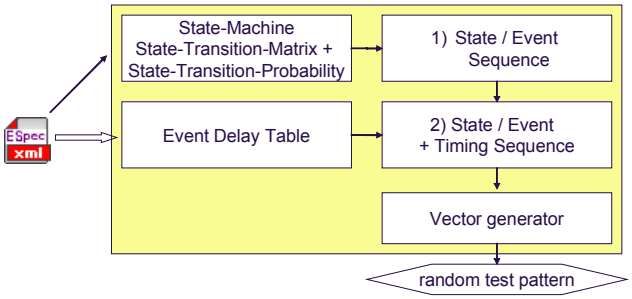

Figure 2: Modules overview for Markov-based Constrained Random Test generation.

In a first step all states and events that can occur are listed. A State-Transition-Table *STT(C|S)* is a direct representation of the weighed graph of the Markov-based model. In the *STT* also mixed-signal events are coded. At the level of the *STT* there is no information yet about timing or what stimulus has to be generated in order to generate the state transition coded in the table. In a textual form the state transitions may be described as follows:

event(WAKE) in state(SLEEP) => state(OPERATE)

event(TIMEOUT) in state(NORMAL) => state(RESET)

It is important to note that only state changes that are in some way observable from the outside are coded, not all internal states, like register settings. This Markov-model may be seen as a golden reference model for the device-under verification.

The weights from the Markov weighed graph are stored in a State-Transition-Probability Table *STP(C|S)* with the same size as the STT. Both tables are now used to generate allowed random state transitions as a Markov process. By calculating the steady-state distribution of the Markov chain by solving the appropriate Eigenvector solution coverage metrics can be derived.

7. Non-stationary extensions

We assumed here a stationary Markov chain generation, based on time independent *STT* and *STP* tables. For many applications the generated event chain is too homogenous and may either miss random-resistant state-transitions or would require test sequences too long to be simulated or tested within a reasonable time.

In this case we suggest to use time-dependent weighs on the Markov model, i.e. *STP = STP(t)*. Some or all weighs on the graph become time-depend, where this time dependence must fit the underlying functionality and may need careful balancing. A simple example for random resistance is a counter, where random counting up or down requires too long sequences to reach all counter states. Random-resistance is typically investigated in the context of Built-in-self test techniques [13], but the problem is the same for Markov-based test generation.

8. Timing calculation

The second layer of the test generation process is the timing calculation. We can code timings between events in a simple form like,

event(SPI) to event(SPI) = 10 us

event(POWER_ON) to event(INIT TIMEOUT) = 20 ms

….

From these descriptions an Event-Delay-Table *CDT(C|C)* is generated, that contains all minimal timing requirements. By applying the following operation to event time $t_{(n-1)}$

$$t_n = t_{(n-1)} + max(min_i(CDT(i, j_n) - \delta(i), 0)) + rand(\lambda)$$

the next event time t_n can be calculated. Here, a random timing adder *rand(λ)* with a non-negative probability distribution is added to achieve coverage in the timing domain, e.g. following an exponential probability distribution. The free timing parameter λ of this distribution must be chosen small enough to maximize the number of events within a given test length and large enough to maximize coverage in the time domain.

This approach works well, when there are only timing restrictions based on minimum timing requirements. We assumed in the generation process that all event changes are indeed possible to fulfill.

Some Automotive Power Devices have watchdog functionality implemented for safety reasons; then also maximum timing constraints must be fulfilled. It can indeed be the case that a generated random sequence based on minimum timing constraints is invalidated by maximum timing constraints and must be discarded or modified. An appropriate timing constraint formalism both for minimum and maximum timings is under investigation.

9. Coverage Estimation example

Figure 3 shows a coverage estimation example from a System-Basis-Chip, a device class that has the beforementioned highly configurable watchdog functionality. To test watchdog functionality in combination with other asynchronous events, it is of interest how many consecutive watchdog failures can be expected after a certain test pattern length. Based on the steady-state solution of the Markov chains we can calculate that we need at least ~300 ms pattern length or simulation time for covering 2 consecutive watchdog failures. For higher order coverage the required time dramatically increases and may lie outside a reasonable test and simulation time. This type of random resistance may also be solved by a time-dependent non-stationary Markov model.

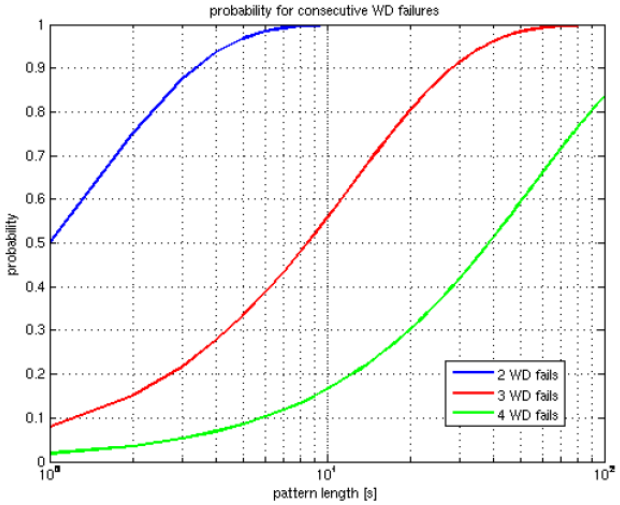

Figure 3: Probability for generating (and thereby verifying) the correct behavior of the SBC after 2 (blue), 3 (red) and 4 (green) consecutive watchdog failure events. Depending on programmable register settings the SBC is supposed to go to fail-safe mode after 1 or 2 of these events.

10. Conversion to protocol and pin-level

The last step of generating valid Markov-based constrained random tests for simulators or ATE consists of converting the events in the Markov chain to protocol or pin-level stimuli and generating expected values for the primary digital outputs, as well as expected analog functionality. On a digital protocol level, supported protocol types for Automotive Power products may include the SPI, LIN, CAN or FlexRay standards.

Generation of digital expected values, e.g. the device answer to an SPI command, is based on the current state of the DUV, including read-out of internal registers, status bits, etc. Also the generation of the result on ATE is accessible through standard digital compare with vector patterns.

Mixed-signal events are more difficult to handle. For each mixed-signal function an appropriate method, both on value and timing, has to be established for the expected output of the Markov-based generated test case. Examples include, high-side or low-side switches, on-board regulators, operational amplifiers, transceivers.

11. Coverage Report and Link to Requirements Engineering

Due to the constrained-random nature of this method the question may arise how the method fits to a classical verification planning, where a certain list of coverage items must be covered, not with a certain likelihood, but must be covered for sure or at least it must be known, which features from a verification list may have been missed.

For this purpose a coverage report is automatically generated for each constrained-random test cases. This report serves as an overview to the verification engineer in a human readable form (HTML), but also in machine-readable form (XML) as a link to requirements engineering software tools.

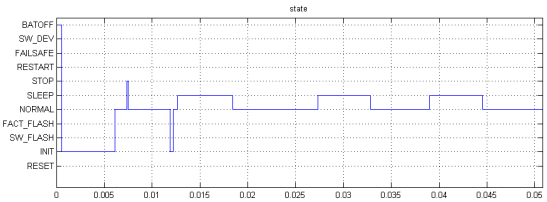

Figure 4: State over time from coverage report.

A number of coverage points are automatically extracted from the generated constrained-random test patterns without the need for the verification engineer to manually define coverage points. These are coverage of states, coverage of commands and commands at state, coverage for registers at state and combinations of commands at first, second or higher order, see excerpts in Fig. 4, 5 and 6.

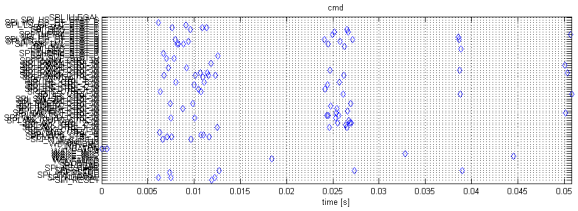

Figure 5: Command over time from coverage report.

Register Coverage

State	Coverage [%/range]@(state=ALL)	RESET	INIT	SW_FLASH	FACT_FLASH	NORMAL	SLEEP	STOP	RESTART
MODE	100	0	100	0	0	100	100	100	0
RES1	100	0	50	0	0	100	100	50	0
VCC2_ON_W	100	0	50	0	0	75	50	25	0
RES2	100	0	100	0	0	100	100	50	0
VCC1_RT_W	100	0	50	0	0	100	75	25	0
WD_STM_EN_0	50	0	50	0	0	50	50	50	0
WD_TIMER	25	0	25	0	0	25	25	25	0

Figure 6: Excerpt from sample coverage report, showing register coverage at different states relative to the full range of possible register settings.

12. Validation and Results

As mentioned before the range of complexity for Automotive Power devices goes from power-switches with few digital content up to SOCs with multiple digital sensor interfaces and µC cores. Out of this class, we chose devices from airbag control-system applications, a multi-port high-side power switch with SPI interface and a System-Basis-Chip with SPI interface, LIN and CAN transceivers and a complex watchdog functionality as examples. For both Markov-based test generators have been developed and the test cases have been applied both to simulation and post-Si verification.

13. Application example: Airbag Control System Devices

The airbag control system is, besides the brake system, the most safety critical control system in modern cars. It involves complexity on the chip level, devices including several uC cores, power switches and sensor interfaces on a single die and complexity on the system-level. Any malfunction may lead to serious injury or death, so to guarantee functional extended robustness is a crucial point in verification.

We applied random power-up tests and functional verification based on Constrained-Random Test Pattern to a power-management device test chips and to an airbag squib driver test chips, and present results for both methods hereafter.

Random Power-Up

Random-power-up tests based on random-power ramp profiles as mentioned before have been applied to an airbag power management device. This device manages the power distribution between battery, charge storage for airbag squibs and also handles cases like rupture of battery supply.

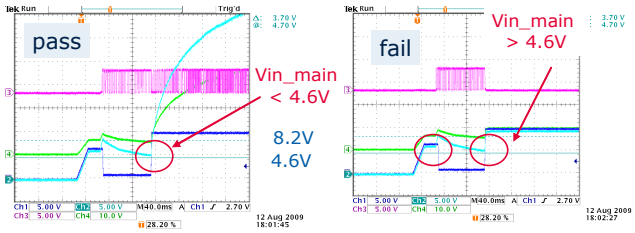

Figure 7: Power-up failure analysis. For a specific power-ramp profile (blue trace) the internal DLL starts clocking (cyan trace) but may fail after final supply voltage is reached (right side)

After running a few thousands of random tests at different temperatures several device issues have been identified, depending on shape of the ramps on different supply pins. The failure cases have been analyzed in root cause analysis. Fig. 7 shows an example of a DLL start-up failure under a specific power-up slope with an intermediate phase of low supply voltage. Under this specific condition an internal DLL fails.

Analysis and Debug Example

During device verification using random-power-up tests as described above in about 40% of all ramp cases output drivers have been observed to be wrongly driving after the power-up phase. This is a dangerous state, because the application expects the shared output data lines to be in high-Z state. The occurrence of the failure depended on the slope of the power-ramp, and besides only slightly on temperature. Not all samples were affected. Typically there were some probe pads for pico-probing inserted in the layout to observe signals with active or passive probes. Strange enough the failure disappeared, when placing an active probe on a probe pad in the output driver enable circuitry.

The capacitive load of the pico-probe obviously forced this node in the following circuitry to rise differently with power, than without the load. Due to lack of more probe pads several FIBs followed, narrowing down the failing block. Finally the only potentially meta-stable circuit element left was a latch. This latch was not connected to the internal power-on-reset detection. In the fail case it flipped in the wrong direction, wrongly enabling the output drivers.

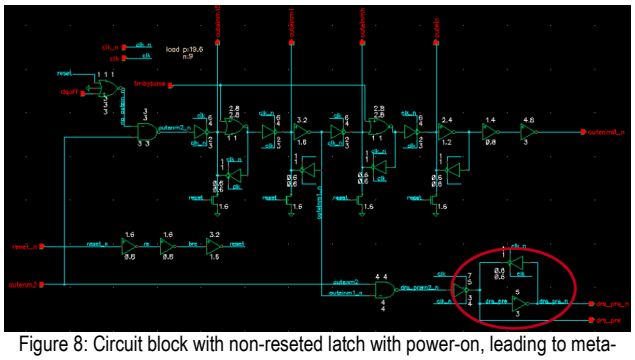

Figure 8: Circuit block with non-reseted latch with power-on, leading to meta-stable output driver enabling after power-up.

Constrained Random Test Pattern

Airbag squib controllers are SOCs that monitor the pyrotechnic airbag squibs during normal operation and provide a firing sequence for multiple connected airbags in case of a car accident. These SOCs communicates with an automotive uC on the airbag control unit through an SPI interface and some separate digital lines for extended safety reasons. It also provides diagnostic function, like ADC readout to the system uC.

Functional coverage must include digital interfaces, like SPI, but also analog functionality for squib diagnostic, firing currents and firing timings.

A reference behavioral-model for the Markov-based test generation has been set-up, in order to provide the expected digital state and firing state of the device as presented above.

While some of the functionality has been tested using the reference model directly on a digital tester, for example the expected digital SPI answer, the mixed-signal functionality requires capture and post-processing of data, including ADC read-outs, analog currents and voltages.

Fig. 9 shows a visualization of the expected pseudo-random switching of the 8 channels of the device in off, diagnostic or firing mode. The expected behavior is compared to captured waveforms in a post-processing step after test run, see Fig. 10.

During post-Si test[1] several device issues have been identified. In addition and for later root-cause analysis a lab based system[2] has been set-up. Through the use of a multi-platform verification management system (XVP) test patterns and conditions of interest for root-cause analyzes could be easily transferred between tester and lab equipment.

[1]on a Teradyne uFlex
[2]lab-based system based on a PXI system running with LabView

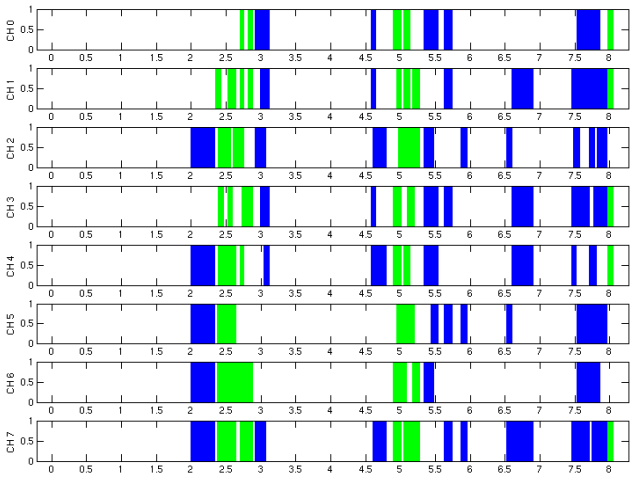

Figure 9: Constrained Random Switching of the 8 channels over time in diagnostic mode (blue) and firing mode (green).

Besides some other functional issues found, one is related to robustness of the SPI-based communication between the system uC and the airbag SOC. In SPI interfaces a transmission error can lead to SPI clocking errors. Then not the default length (16 bits) but less or more may be recognized by the DUT. Besides applying illegal commands to the DUT, in which case the command should be ignored, also SPI frames with illegal length are included in the constrained-random pattern generation process.

Here a specific failure case could be identified, where at a certain SPI clock frequency range an illegal SPI frame length leads to the fact, that a subsequent legal SPI frame is ignored, see Fig. 11 for details.

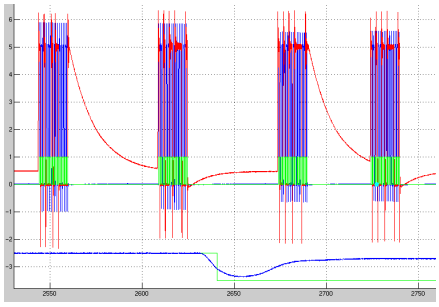

Figure 10: Post-processing of mixed-signal data. Here captured SPI frames are shown (upper blue and red traces), captured voltage over one of the power switches (blue trace below), and expected data from the model (green traces). In this test pattern the 2nd SPI frame enables the firing. The expected firing fits to the captured one within accuracy.

14. Application example: Multi-channel SPI controlled high-side power-switch

This device-type is used to drive high currents for bulbs, motors, etc. The functionality can be controlled via an SPI interface; it also includes supervision functions for over-current and temperature. Major state changes are based on power supply changes, so the Markov-based model also included mixed-signal extensions, like power supply changes, over-current and over-temperature events.

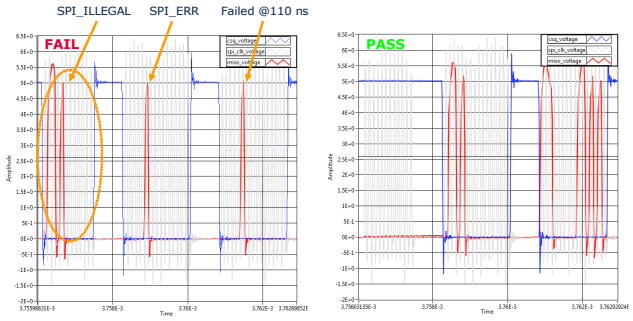

Figure 11: Wrong SPI response after an SPI frame with an illegal SPI frame length. The left side shows captured waveforms from a root cause analysis. Here three SPI frames are shown (blue = SPI chip select, red = SPI output from DUT). The first SPI frame shown is a frame with illegal length (16 bits). The other two are legal, but the DUT shows the wrong SPI answer for a specific SPI clock frequency (keft side). To confirm this the correct SPI answer is shown (right side) after deselecting the illegal SPI frame.

The generated functional tests then included checks for SPI answer and switching state of the channels. While the digital SPI answer can be directly compared to the model states, the switching state has to be probed by driving a probe current through the switch, see Figure 12 for a simulation result. Limitations to post-Si validation when applying the test cases on test equipment include slow response of power-supplies, need of conversion of currents into voltages and compare to an expected range, inability to control and generate temperature events due to slow temperature response or self-heating of the device.

Generated test cases for post-Si verification on a PXI system had a typical length of up to 100 ms, containing some hundreds of SPI command frames and some tens of power-mode transitions. As a result of this investigation various design issues including mixed-signal aspects could be identified.

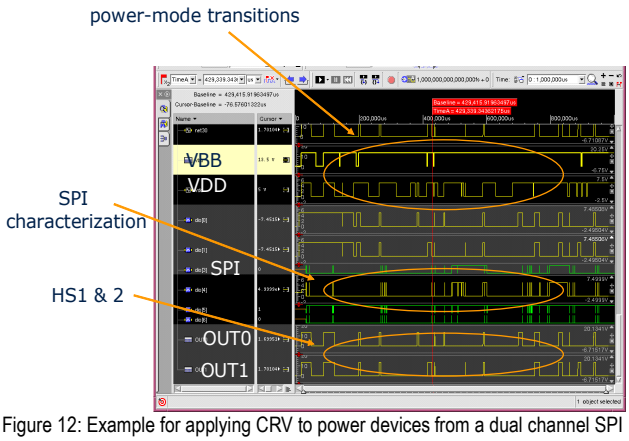

Figure 12: Example for applying CRV to power devices from a dual channel SPI controlled high-side switch. Random power-mode transitions on logic supply VCC and power supply VBB, random SPI commands, generating switching of the high-side power-switches HS1&2 on channels OUT0 and OUT1.

15. Application example: System-Basis Chip

The previously mentioned System-Basis Chips (SBC) contain approximately 2^{2592} externally addressable states due to high configurability of the analog, monitoring and timing behavior. SBCs provide and monitor several functionality on automotive Electronic Control Units (ECU). They are interesting devices from a test and verification point of view, because they combine heterogeneous functionality on a single die, including one or many voltage regulators, transceivers of different types, like CAN, LIN or FlexRay, power switches and controller interfaces like SPI. SBCs may also include different watchdog functionality with complex and / or configurable timings, including complex Finite State Machines.

The SBC supervises the functionality of a µC and other connected circuitry by regular checks of a response from µC. This watchdog functionality is also highly configurable, therefore expands the verification space in terms of states and timing. Besides, there is a large gap to be covered in terms of simulation time or pattern length, because both MHz-range signals (SPI, transceivers) as well as ms-range events (watchdog functionality) must be applied.

As mentioned above, it may not be possible to achieve state coverage targets with a static *STT*. In this particular case many transitions lead to default register states, which means that in most cases only a limited subset of the SBC configuration is verified. We therefore used a dynamic table $STT = STT(t)$ for test case generation; resulting in a non-stationary Markov process. As a drawback steady-state calculations for coverage metrics cannot be applied anymore, but must rather be based on post-generation examination of the coverage report, see also Fig. 13.

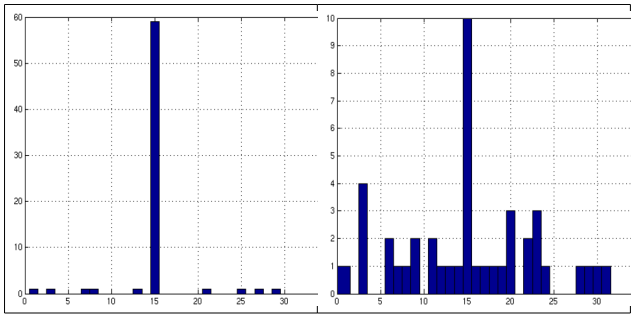

Figure 13: Distribution of configuration settings for watchdog timings showing random-resistance with a stationary *STT* (left), mostly only the default configuration 15 is visible, and increased coverage with a dynamic *STT* (right), with a broader distribution of register setting. The histograms show the frequency of occurrence of register settings related to a watchdog timer (5 bits) for same test pattern lengths.

Test cases were extended to a characterization program for digital interface timings and ran both on a mixed VHDL-AMS / Verilog model and for post-Si on a PXI and an ATE system (Teradyne µFlex).

Analysis and Debug Example

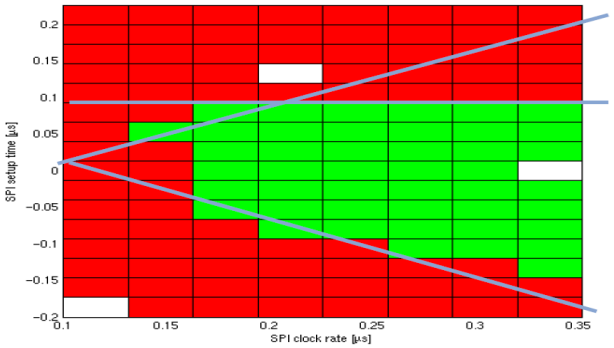

Figure 14: Pre-Si characterization of an SBC with SPI interface. (green = pass, red =fail, white = simu crashed). Upper left area for SPI clock frequency versus SPI setup time

Several functional issues could be identified during device verification. One of them was detected as a failure region in an SPI interface clock speed versus data setup-time shmoo, see Fig. 14. The typical triangular pass region of this type of shmoo was truncated for larger data setup-times.

Repeating the pattern with clock speed and setup-time in the fail region with a VHDL-AMS behavioral-model, the failing SPI output bit showed an usual small width of less than UI/2, see Fig. 15. In the following analysis a setup-time violation of the digital SPI interface for that specific SPI command was identified as root-cause of the issue and addressed in a re-design.

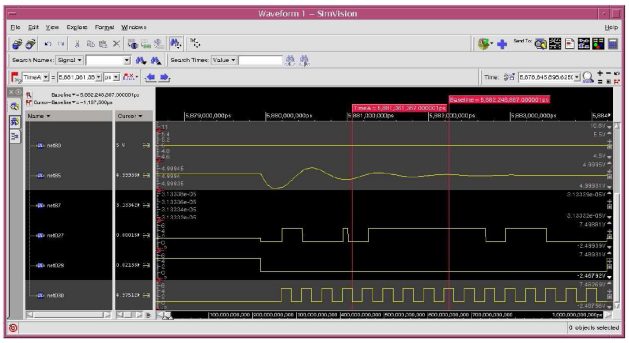

Figure 15: Simulation of issue from Fig. 14, showing SPI output trace with data width of less than 1/2 UI. Traces from below: SPI clock, SPI chip select, SPI output.

16. Implementation

We linked the Markov-based test generation to the environment of an Executable Verification Plan (XVP) [8,12] in order to allow access to specification data, pattern conversion tooling and automatic transfer of the test cases to different target platforms. The Markov-based model is implemented in Matlab / Simulink and uses XML to communicate with the verification environment. Available target platforms for simulation are Cadence simulators, Matlab / Simulink models, and for post-Si verification PXI and ATE. The number of test cases is mostly limited by simulation time and / or test pattern length, while the generation speed is uncritical.

17. Conclusion and Outlook

In this paper, we present the verification strategy and several methods for functional verification of mixed-signal automotive power devices. This device class follows a trend to an increasing integration of digital functionality and analog power electronics driven by increasing requirements for functionality and safety. The verification gap that follows complexity can be closed by extending constrained-random verification methods known from digital verification to mixed-signal aspects.

Several different strategies are presented, including non-stationary Markov-based test pattern generation to address random resistance, random-power-up tests and linking post-processing of captured digital and analog ATE data, when necessary. We present experimental results from several device types, how these approaches have been successfully applied to detect hidden design issues and show a link to verification engineering.

References

1. X. Shiyi, “Random-like testing of very large scale integration circuits”, Journal of Shanghai University (English Edition), Volume 2, Number 4 / December 1998.
2. A. Majumdar and S. B. K. Vrudhula, “Random-test length estimation: Analysis and techniques”, Tech. Rep. SIUC/DEE/TR-93-2, Dept. of Electrical Engineering,Southern Illinois University at Carbondale, 1993.
3. A. Majumdar and S. B. K. Vrudhula, “Fault Coverage and Test Length Estimation for Random Pattern”, IEEE Transactions on Computers, Vol. 44 , Issue 2,February 1995.
4. C. Lee et al., “Efficient Random Vector Verification, Method for an embedded 32 Bit RISC Core”, AP-ASIC. Proc. of the Second IEEE Asia Pacific, Conference, 2000.
5. N. Kitchen and A. Kuehlmann, “Stimulus Generation for Constrained Random Simulation”, Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design
6. T. Nirmaier; W. Spirkl; E. Liau Chee Hong, “Fully automated semiconductor operating condition testing”, Test Conference, 2006. ITC '06. IEEE International, Vol., Iss., Oct. 2006.
7. T. Nirmaier, W. Spirkl, “Method and apparatus for generating a random sequence of commands for a semiconductor device”, US patent 20090222779
8. A. Pirker-Frühauf, M. Kunze, “A novel methodology to combine and speed-up the verification process of simulation and measurement of integrated circuits”, Autotestcon 2008
9. T. Nirmaier, J. Zaguirre, E. Liau, W. Spirkl, D. Landsiedel, “Efficient high-speed interface verification and fault analysis”, International Test Conference 2008, Santa Clara
10. T. Zhang, T. Lv, S. LI, “An Abstraction-Guided Simulation Approach using Markov Models for Microprocessor Verification”, Proc. DATE 2010.
11. IEEE 1450, “Standard Test Interface Language (STIL)”
12. A. Pirker-Frühauf, W. Gallent, M. Kunze, G. Pelz, “Acceleration of IC verification process using advanced flexible modular measurement systems and software architectures”, I²MTC – IEEE
13. P. Bardell, W. McAnney, J. Savir, “Built-in-Test for VLSI: Pseudorandom Techniques”, John Wiley & Sons