A coverage-driven AMS verification of a 4Mb Z-RAM® macro

Yogitech and Innovative Silicon

Monia Chiavacci

Michel Bron

Session 1

Presented at

cadence designer network

A coverage-driven AMS verification of a 4Mb Z-RAM® macro

Chiavacci Monia, Pescari Egidio, Froio Giuseppe, Yogitech SPA
Michel Bron. Anant P. Singh, Hamid Daghighian, Paul de Champs, Florent Haik, Innovative Silicon

Abstract

Memory macros are typical examples of mixed signals circuits in which the amount of information to be processed in verification is very high (many operative modes, many configurations, many signals to control, etc) and where the analog and digital portions are closely linked and generally cannot be separated in the design hierarchy.

In a traditional methodology, beside high automated approach for the digital side, the verification of analog part is usually based on visual inspection of waveforms, poorly automated and mainly executed in the post-processing phase. A methodology and a flow able to provide a rigorous checking on both digital and analog signals and to extend to the analog domain the concept of functional coverage is now required.

The YOGITECH AMSvKit makes all of this possible as a fully automated verification environment using Cadence Specman Elite as the best-in-class verification tool and a mixed signal simulator.

A detailed verification plan has been created describing the goals of the verification and the architecture of the AMS verification components based on AMSvKit. Each component addresses a defined set of checks and/or stimuli and is e language re-use methodology (eRM) compliant to ensure interoperability and reusability.

One part of the verification plan was implemented and executed. The resulting verification environment was able to detect all the possible failures and out-of-spec conditions and the time spent for the measurement campaign was significantly reduced when compared to the manual approach.

1. Introduction

Most System-on-Chip designs today include both analog and digital blocks with a high degree of interaction. Their joint verification in a unified environment is extremely hard to automate due to the very different tools and methodologies used in each of the digital and analog domains.

Memories are very good examples of mixed circuit devices where the complexity of the digital has to dial with critical analog behaviour. For their specific nature, memory macros are inside a digital world but their core is analog. Ensuring the correct functionality of a memory macro means to ensure the capability to perform a lot of operations (many kinds of write and read tasks) among a wide amount of data (increasing dimension for address space and words length) and in many operating modes (normal mode, test mode, low power mode etc) but also match challenging requirements in timing.

Number of operating modes, behavioural complexity, and circuit size contribute to make the verification process more difficult. Functional failures can come in the form of inoperable modes and are often due to “simple” errors such as inverted signals, swapped bit lines, and incorrect power up sequencing. Functional failures tend to result in failed chips that are show stoppers as the end customer cannot begin bringing up the firmware that is typically run on the SoCs. In contrast, a performance failure is not as fatal because the system development can usually continue while the IC is re-spun for performance improvement.
The end result of functional failures is many design iterations that are usually at great expense in terms of non-recurring engineering costs (NRE) and missed market windows.

For a memory macro, timing requirements are conceptually related to performance but due to the nature of these circuits are very important.

This is especially true for last generation of such devices where performances in timing are pushed at the edge of the technology capability. Moreover, with technology processes downscaling, parasitic effects due to path lengths and interactions become the key factor for performance. They are difficult to predict and regardless of tricks and adhoc design practices, a validation at top level is strongly required.

Z-RAM provides SoC designers with speed, density and power advantages that aren't available with other memory solutions. A distinctive element to ensure features and performances is to rely on a well defined and robust verification flow and methodology.

The goal of the verification methodology is to systematically find error in a reproducible manner. It should give a rigorous approach to reduce as much as possible risk of having functional bugs in silicon but also give an objective metric to measure this risk. This is why functional coverage is so important.

Verification is a complex task and as such it requires a good methodology but it is also time consuming with a lot of repetitive steps. This is why it is important to have flows and tools introducing reusability and automation.

A good verification flow brings some interesting collaterals in terms of efficiency and predictability of design process and in identifying errors in specification document.

2. Z-RAM Memory macro

Innovative Silicon Inc. is a fast-growing, venture-backed company that develops and licenses the ultra-dense Z-RAM® (Zero-Capacitor DRAM) memory technology for Systems on-Chip (SoC), microprocessors and portable consumer applications [6].

Z-RAM® memory technology harnesses the floating body effect of silicon on insulator (SOI) semiconductor devices [5]. Unlike conventional SRAM, which uses 6 transistors, or DRAM, which uses 1 transistor and a complicated capacitor, Z-RAM® uses a single transistor – and nothing else – as the memory bitcell, as shown on Figure 1. The density and simplicity of this technology makes Z-RAM® the lowest-cost memory technology – both as stand-alone memory and as embedded memory on microprocessors and other leading-edge semiconductors.

Z-RAM® is 5x denser than SRAM and up to 2x denser than eDRAM, enabling you to…

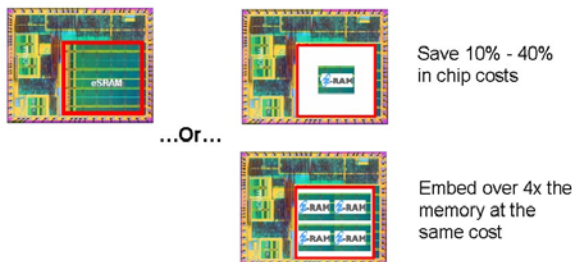

Figure 1:Innovative Silicon Z-RAM®

The Z-RAM memory macro design verified with this flow comprises 16 sub-arrays of 260 rows by 1120 columns

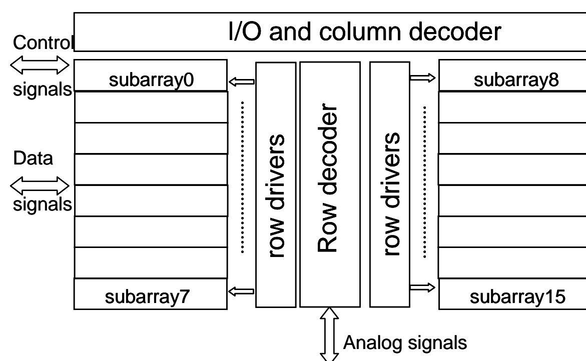

Figure 2: Z-RAM macro block diagram

A full digital functional verification of Z-RAM macro is possible by black boxing large part of the memory macro. The functionality of the black boxed areas meanwhile is verified using an analog simulator. The two environments are exercised in isolation from each other. This leaves a hole in the verification (both functional and timing) mainly at the interface of the digital and analog areas. The functional coverage on the analog portions is minimal as all the instantiations of these circuits in the design are not verified in their unique environment. A true mixed signal verification environment is required to verify the digital and analog portions together as they interact in real time. Such an environment facilitates the use of a large number of vectors in the simulation while rigorously measuring and reporting analog parameters and quality metrics across a large portion of the design. The verification components (vComponents) are ready-to-use to create verification environments (e.g. test-benches).

3. AMSvKit for mixed-signal functional verification

The AMS functional verification of the memory macro is implemented using the Yogitech Analog Mixed-Signal Verification Kit (AMSvKit) [1].

The AMSvKit is a tool able to link analog and digital approach extending to the analog domain verification techniques already used in the digital one providing a unified environment for mixed signal verification, based on Cadence Specman Elite and "e" language [3].

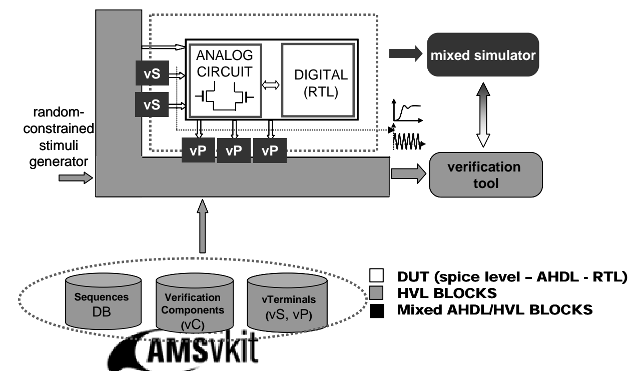

Figure 3: General verification environment using AMSvKit

As illustrated in Figure 3 the kit is composed of three libraries (vTerminals, vComponents and Sequences DB) and all the necessary infrastructure to make working the full environment (simulator scripts,structures/unit, etc.) .

The core of the kit is a library of "verification terminals" (vTerminals) that creates an interface between the analog and digital domains. The vTerminals are divided into two types:

- verification sources (vSources - vS), which are models of signal sources configured and controlled by digital commands from the verification environment that provide continuous and time-continuous voltage and current signals or analog events; they include DC, pulse and sinusoidal signal (current and voltage) generators, noise injectors and parameter spread emulators;
- verification probes (vProbes – vP), which transfer analog information from the mixed-signal simulator to

the verification environment; they provide values of voltage, current and timing parameters and include self checking mechanisms (e.g. check a sampled voltage level within a pre-defined range); examples of vProbes are voltage/current/time detectors, linear behaviour and total harmonic distortion calculators, AC gain extractor etc.

The verification components (vComponents) are ready-to-use to create verification environments (e.g. test-benches) for basic analog blocks such as band gap cells, oscillators, voltage regulators, comparators, operational amplifiers and buffers.

In order to calculate a non-trivial analog parameter it is necessary to properly control and configure a number of vSources and vProbes and to synchronize them. This is implemented using sequences: a structure that represents a stream of items signifying a high-level scenario of stimuli.

The sequences DB provided with the kit includes all the sequences needed in an analog context.

The AMSvKit includes a GUI used to create the environment and to instantiate, connect and configure the needed vTerminals and verification components.

Using AMSvKit the powerful generators of Cadence Specman Elite can be used to generate also analog stimuli; checking mechanisms can be applied to analog verification items and functional coverage can be evaluated also based on analog metrics according to the defined verification plan.

4. Verification strategy and planning

The aim of the verification strategy is to mitigate the design risks identified by the design team.

The fist step of the verification activity is to define the verification plan: it lists the issues and presents a brief description of how verification has to be done.

The first section of our verification plan describes meaning, expected behaviour and constraints of any I/O ports of the circuit under test (DUT).

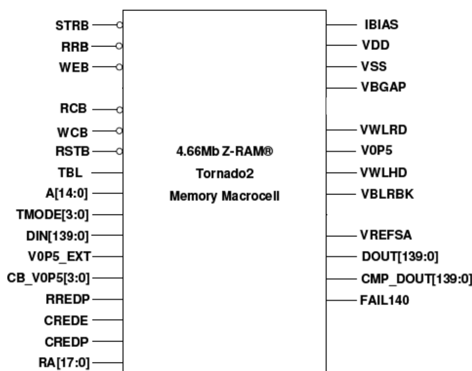

Figure 4: 4Mb Z-RAM pin out

For the 4Mb Z-RAM the pin-out is reported in Figure 4 and Table 1 describes the ports classification.

Group name	Ports list	Function
Control signals	STRB, RRB, WEB, TWL, RCB, WCB, RSTB, TBL	Selection of basic operation to be done in the memory and its timing
Data signals	DIN[139:0], DOUT[139:0], CMP_OUT[139:0], FAIL140, A[14:0]	Data and address busses handling, including some information used in BIST operations
Special signals	TMODE[3:0], V0P5_EXT, CB_V0P5[3:0], RREDP, CREDE, CREDP, RA[17:0]	Control of special functions; test mode 0, test mode 1, column test and programming of rows and columns redundancy.
Analog signals	IBIAS, VDD, VSS, VBGAP, VWLRD, V0P5, VWLHD, VBLRBK, VTT, VREFSA	References voltages, bias currents and power supply signals.

Table 1: I/Os ports description

Moreover we need also to take in account some internal signals, both analog and digital, in order to ensure their correct behaviour. These signals and their short description are reported in Table 2.

Signals set	Type	Description
XDEC signals (SEL)	Logic	row decoder output signals (two for each memory row)
SA control signals (SA, YMUX)	Logic	control signals for sense amplifier sections
WL drv	Analog	Analog driver for gate line of each row
Bitcell signals	Analog	Bit cell signals (2 for each bit cell)

Table 2: Internal signals to be monitored in verification

Another element to be considered during the planning phase is related to the environment conditions, i.e. process parameters and temperature.

Normally, designers perform a so called PVT (process, voltage, temperature) analysis to ensure the correct operation of the circuit in different conditions related to process parameters spread, power supply voltage and temperature. In this case the PVT analysis is needed as well and it is necessary to define a set of model cards (i.e. set of process parameters) and a range of temperature. In this way, for each set of input stimuli, it will be necessary to perform a number of simulations equal to the number of PVT corners. The results are cumulated and included in the functional coverage evaluation.

The core of the verification plan is the definition of the verification items. Each item, identified by an ID, represents a measurement involving analog and/or digital information to be performed by a well defined procedure for well specified conditions.

In this particular case, the verification items can be split in 3 sets as described in the following.

1) Memory functionality evaluated at I/Os level.

   This aims to check if the protocol of each possible operation is correctly implemented and if the memory functionality is there (i.e. writing a data in a specific address and then reading the same data from the same address).

   These checks are exactly the ones done in a pure digital verification and in principle do not involve any analog contents. The DUT can be described in RTL or by a gate-level netlist with a behavioural model for the bit cell array. It is worth to note that to perform this kind of verification is mandatory to use tools able to handle big amount of data and to

evaluate functional coverage in order to measure how thoroughly your test-bench exercises your design.

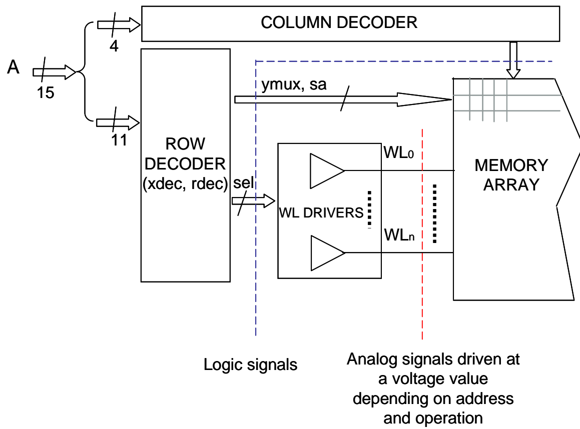

Figure 5: Addresses decoding in the Z-RAM macro

2) Address decoding evaluated looking into internal signals.

During normal operations, in the memory macro, the content of the address bus is decoded by internal logic to address the specific set of bit cells and to drive into their control signals proper voltage levels needed for the specific operation. A block diagram depicting this task is in Figure 5. In this case for each operation, the logic behaviour of SEL, YMUX and SA signals has to be checked together with the voltage levels of WLi. This set of items includes analog measurements. In Table 3 is reported the expected behaviour.

Operation	WL	
	selected	unselected
RR	VWLRD	VWLHD
WR - 0	VWLWR	VWLHD
WR - 1	VWLWR	VWLHD
RC	VWLHD	VWLHD
WC	VWLHD	VWLHD
LA/IDLE	VWLHD	VWLHD

Table 3: Expected behaviour of WL in normal mode; the abbreviations in the table represent voltage values within general 10% of approximation (e.g. VWLWR = 0.5V ± 0.005V)

3) Bit cell functionality and timing

Depending on the selected operation the addressed bit cell is driven in a defined way as shown in Figure 6 in two of the possible cases. The items to be checked are:

- voltage values of each signal (BL, WL) before, during and after the operation
- significant delay periods (e.g. tBSRw, tSBO, tWSRw in Figure 6)
- pulses width and slew rate of each expected transition

Considering all these conditions for each operation generates 35 verification items to be addressed and covered for each bit cells.

Of course it is not necessary to monitor all the cells. A smart strategy, in fact, suggests involving only the cells positioned at the corners of the array: these represent the worst case conditions for timing.

Considering the amount of verification items and the complexity in size of the circuit, run time (i.e. time duration of meaningful simulations) is a critical parameter. That's why the simulation strategy and circuit partition becomes really important in order to optimize the run time without loosing in results accuracy.

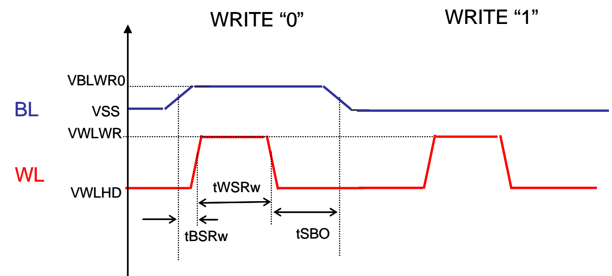

Figure 6: Bit cell (the addressed one) driving protocol during "write" operations

Depending on measurements and checks we have to perform it is possible to use different descriptions or abstraction levels of the same sub-circuit. For instance to address "memory functionality evaluated at I/Os level", as already stated, it is enough to use an RTL (fully digital) description of the boundary of the macro and a behavioural model of the array; instead once we want to check the "bit cell functionality and timing" it's mandatory to include the actual (analog) implementation at least of the monitored bit cells and the drivers of the voltages in question, and only for those in order to reduce as much as possible the run-time. The circuit partitioning is defined following some rules:

- using digital (RTL or gate-level) description where checks or measurements involve only "logic" information;
- using transistor level description only for sub-circuit for which voltage values and/or timing behaviours become important;
- taking advantages from the symmetrical structure of the memory array to limit analog measurements.
- introducing parasitic effect from layout (back-annotation) only for sub-circuits where this is strictly necessary;

The flexibility in changing descriptions of blocks is one of the required features driving the definition of the simulation environment.

Another element to optimize the run-time is related to the simulator speed. Surely a mixed signal simulator is needed but the analog engi ne can be a "spice" simulator

or a “fast-spice” one. Trade off between speed and accuracy drives the choice.

Verification environments

To address the verification items described in the previous section, we have defined two environments or test-benches. Both of them are eRM compliant [3].

In Figure 7 is shown the verification environment addressing both memory functionality evaluated at I/Os level and decoder section. It is composed of a verification component handling digital signals named MEM eVC (e Verification Component). MEM eVC drives control and data signals according to specification [1] and checks protocol rules and memory functionality.

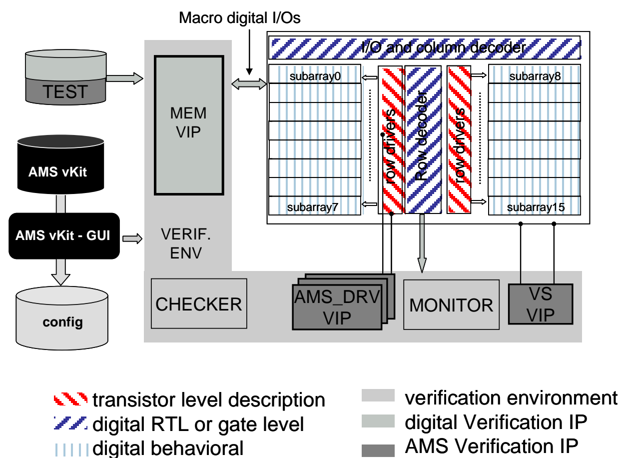

Figure 7: Verification environment addressing memory functionality and decoder section

Moreover there are other two kinds of verification IPs, both of them handling analog information. The first one is called AMS_DRV VIP and it is able to monitor and to check the WL drivers section of the addresses decoder. Each AMS_DRV VIP monitors signals related to a subset of 4 rows of a defined sub-array (this depends on the macro structure) so it is possible to instantiate many of them increasing the overall coverage. This is ensured by the eRM compatibility of every VIPs[3].

The second one is called VS VIPand it allows generating all the needed reference voltages, bias currents and power supplies.

The VIPs can be instantiated, connected to DUT nets and configured by the AMSvKit GUI. The configuration is also stored in a text file called config: this allows introducing modifications without a graphical approach.

As explicitly depicted in Figure 7, to optimize the run time a specific partition has been implemented involving spice (or transistor) level and RTL or gate-level descriptions together with behavioural models.

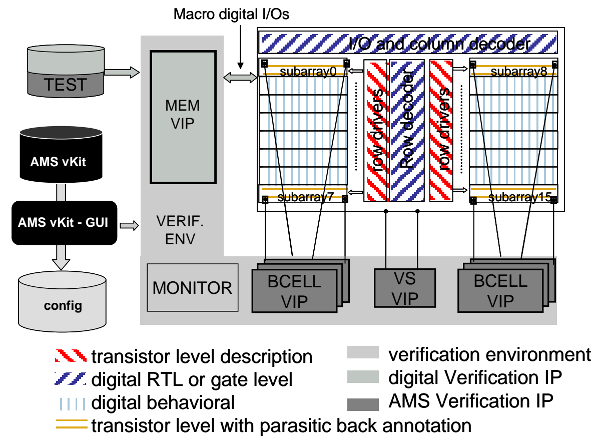

Figure 8: Verification environment addressing memory functionality and bit cell timing

The second environment, shown in Figure 8, has the same structure of the previous one but the AMS_DRV VPS is replaced by the BCELL VIP. The BCELL verification IPs instantiated are 8 in total to monitor the 8 bit cells located at the corners of the two sub-array sections (sub-array 0 to 7 and sub-array 8 to 15). Moreover the circuit partition changes accordingly: four sub-arrays are described at transistor level including parasitic effects extracted for layout views.

The simulation becomes very long making impossible a good coverage for memory functionality. Due to that the first environment proposed is mainly used to reach enough coverage about memory functionality; instead the second one is mainly focused on bit cells functionality and timing.

Demonstration

To prove the effectiveness of this approach for the verification of a memory macro we have developed a part of the first verification environment (Figure 7) including only one AMS_DRV VIP and without VS VIP. In fact voltage references and bias currents have been generated by ideal voltage and current generators.

To match the requirements of the described simulation strategy, the chosen simulator was Advanced AMS from Mentor Graphics together with HSIM from Synopsys: the fast spice engine allows speeding up the simulation itself when there is a big analog content.

The verification execution generates reports and logs to summarize results of checks and functional coverage; of course the waveform database is always available but visual inspection can actually limited to a further analysis of any potential issue highlighted into verification reports.

During simulation a log file is created in which one can find a detailed messaging about each event and operation both in the analog and digital domains including results of self checking mechanism. As shown in Figure 9, during a single read operation, some checks on analog information related to the WL drivers are done.

After the simulation every set of verification IPs generates its report file with all the measurements and results of the comparison towards the expected behaviour.

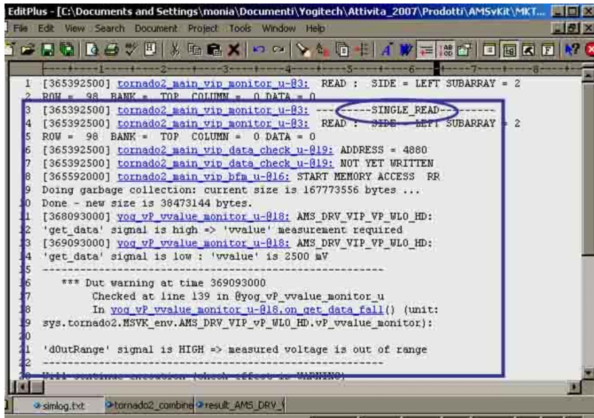

Figure 9: On-the fly log file; highlighted in blue events related to a single read operation

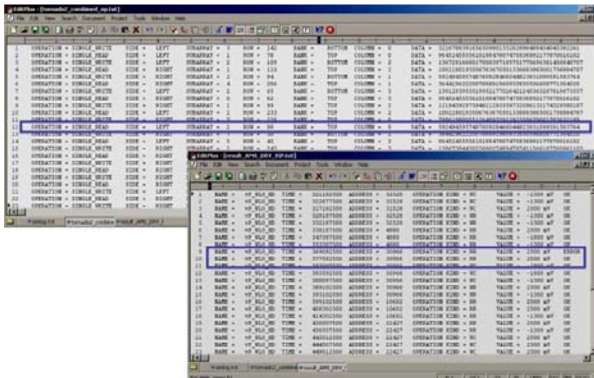

Figure 10: Reports generated by VIPS; the first one is from MEM VIP and shows details about the operation done (highlighted in blue the line corresponding to the operation highlighted in Figure 9); the second one is from AMS_DRV VIP and shows analog measurements

Figure 10 shows snapshots of the two reports available in the environment and summarizing the results of the execution.

Finally, in the graphical interface of Specman Elite it is possible to load the functional coverage as depicted in Figure 11 where the details about row address sets are highlighted.

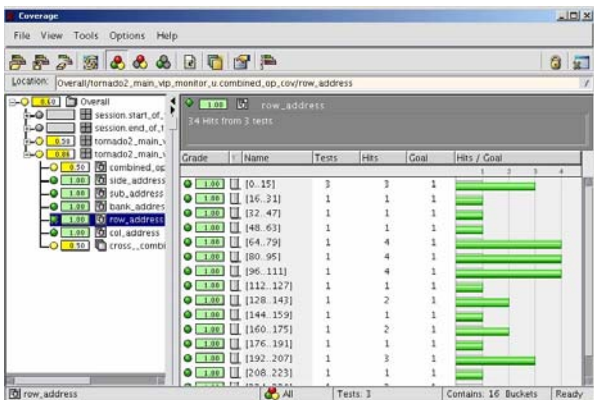

Figure 11: Functional coverage

The automation of the described flow limit the engineering effort spent to collect and analyse results, leaving a more accurate investigation on waveform database in case of unexpected behaviour highlighted by errors or warnings.

Conclusion and future work

The results of the demonstration show the capability of the described approach to create a true mixed signal verification environment as required verifying the interaction of analog array and drivers and digital decoders and state machines as they work in real time.

The high level of automation allows performing time consuming simulations with acceptable level of coverage without spending man effort in results evaluation unless we find unexpected behaviour (i.e. bugs requiring further evaluation based on waveforms inspection).

Moreover the reusability allows a strong reduction of set-up time for the verification of other macros. This is a very important advantage differentiating this approach towards the traditional ones.

The main limitation we faced in this phase was related to the implementation of the required simulation strategy. As explained in the previous sections, design partitioning (i.e. the flexibility in changing descriptions of blocks) plays an important role and a stable flow to change blocks description adding parasitic back-annotation where needed is mandatory. This element has to be fully defined before to proceed with the verification environments implementation and this is the goal of the next step.

Reference

[1] G. Bonfini, M. Chiavacci, R. Mariani and E. Pescari. “A mixed signal verification kit for verification of analog-digital circuits” Proc. of *IEEE Design Automation and Test in Europe*, Munich, Germany, pp 88-93, March 2006

[2] IEEE 1647: http://www.ieee1647.org/index.html

[3] CADENCE “e Reuse Methodology (eRM) Developer Manual” Version 6.1.

[4] S. Okhonin et al, “A SOI Capacitor-less 1T-DRAM Concept”, IEEE Int. SOI Conf., 2001, pp. 153-154

[5] D.Fisch et al, “Z-RAM Ultra-dense Memory for 90nm and Below”, HOT CHIPS Conf., 2006