Event Driven Analog Modeling of RF Frontends

Stefan Joeres
Chair of Integrated Analog Circuits
Sommerfeldstrasse 24
RWTH Aachen, Germany
joeres@ias.rwth-aachen.de

Hans-Werner Groh
Atmel Germany GmbH
Theresienstrasse 2
74072 Heilbronn, Germany
Hans-Werner.Groh@hno.atmel.com

Stefan Heinen
Chair of Integrated Analog Circuits
Sommerfeldstrasse 24
RWTH Aachen, Germany
mailbox@ias.rwth-aachen.de

ABSTRACT

The modeling language VerilogAMS supports a new double precision datatype (wreal) that enables analog accuracy in the digital simulation domain. It is therefore possible to seperate high frequency signal paths, like those in RF frontends, from the rest of the chip, which comes in very handy for fast verification purposes. After an introduction to analog and digital modeling, a strategy to model the RF and LO signal flow path for a bluetooth transceiver system is presented. Especially the nonuniform oversampling, which is introduced through wreal to electrical and vice versa conversion, is analyzed and compared to traditional uniform sampling ratios. The proposed approach is demonstrated for an industrial available RF frontend, including biasing and analog to digital conversion. Simulation comparisons for different complexities of the frontend and different modeling approaches like passband, baseband and transistorlevel conclude the paper.

General Terms

Modeling, VerilogAMS, wreal, nonuniform sampling

1. INTRODUCTION

Verification is becoming a key component in today's design of complex, highly integrated circuits. More and more digital functionality enters the formerly analog-only radio frequency (RF) frontend. Self calibrating, self adjusting and reconfigurable building blocks need a huge number of digital connections and programming routines. Since these blocks are need to be connected to each other, a verification of the complex interaction between all of them is necessary. This so called functional verification has particularly to be done on the final tape out database prior to fab tape in, when every wire connection and signal routing already has been done. Nevertheless, the target is to provide a simulation strategy that can verify the system functionality "over night" - already during the design process. This leads to a "golden schematic" strategy, where parts of the system design can already be done inside a design framework like Cadence®. Each building block is represented by a behavioural model which implements the basic functionality. These can then be subsequently replaced by transistorlevel schematics and refined models.

A transient system simulation on transistorlevel of the analysed complex transceiver for one data burst of 1 ms would take approx 500 days to complete. Using the proposed approach we can reduce this time to approx 5 hours. A lot of information is lost at the cost of this speedup, but it is still sufficient to provide a basic functionality and connectivity test.

In the following an introduction to analog and digital domain simulations, including supported datatypes and their advantages including event driven modeling is given. With the new approach of event driven analog modeling the sampling time is becoming a critical factor which is investigated in the following section. The next following section deals with the problem of enabling constant sampling times over different amplitude precisions. Section 3 demonstrates the approach using a commercially available bluetooth transceiver system.

2. SIMULATIONS IN ANALOG AND DIGITAL DOMAINS

Transient simulations are the only feasible ones for a mixed signal system verification, since they are closest to the real functionality.

The design methods to create digital and analog building blocks are very different. It is therefore beneficial to use signals with different disciplines, simulated in different domains. Typically, digital signals (discipline logic or the continous value datatype wreal) are simulated in the discrete time domain, while analog signals (discipline electrical - voltage and current) are simulated in the continous time domain. This enables the most efficient use of today's simulators, when trying to keep the "traditional" design methodology for both domains.

Signals in the discrete time domain can either be simulated by a fixed sample frequency, where calculations are carried out once every period or using a trigger/sensitivity scheme, where calculations are carried out when a signal at the input of a building block changes. From the digital designer's point of view this is called synchronous or asynchronous cal-

978-1-4244-1567-0/07/$25.00 © 2007 IEEE

culation.

In modern simulators like the Incisive Unified Simulator from Cadence the discrete time domain is always simulated using the asynchronous approach, since this is far more efficient in terms of simulation speed.

Analog signals are simulated in the continous time domain, where the equations that are derived from the schematics are simulated at varying time intervals. These intervals are changed by the simulator according to the required accuracy. The equations are solved at the simulation time t and at $t + \Delta t$. If the simulation results are within the accuracy requirements, the simulation is continued, else the time step Δt is reduced and simulation is done again. This approach is known as simulating in the continous time domain.

The mentioned equations form a two dimensional matrix of current and voltage dependencies, that is derived from the electrical characteristics of the nodes, branches and devices in the circuit. It is solved using different algorithms, depending on the simulator and the desired precision. As shown before, the time steps between each of the solutions are continously shortened, until the desired accuracy criteria (e.g. maximum voltage difference at each node) between two time steps is fulfilled. The main problem in analog simulation is that the whole matrix has to be solved for each timestep, regardless of the number of nodes that changed their value. Single high frequency nodes - like those in RF frontends - determine the number of matrix calculations for the complete chip, although most signal changes are far below the accuracy criteria.

2.1 Analog Domain Modeling

Since analog signals are simulated in the continous time domain, the models are normally also written using continous time equations. These include equations for electrical characteristics like those for capacitors, where $U = \frac{1}{C} \int I dt$ or $I = C \cdot \frac{\delta U}{\delta t}$. Filter functions can either be implemented using discrete RLC component equations or using the Laplace function of a transfer function. For Laplace based filters, it is important to define the maximum allowable timestep for the simulator so that the filter characteristics match the desired functionality in simulation.

2.2 Digital Domain Modeling

In contrast to the analog domain, digital domain simulation as it is used in computer algebra programs like Matlab^{TM} uses discrete, predefined time steps (synchronous simulation). The time difference between each of these steps is defined by the sampling frequency, which must be at least twice the highest frequency of the desired signal to avoid aliasing effects. The functional transfer from analog signals to discrete time sampled signals is well known. Especially the work on digital filters with analog accuracy for a given frequency range has been researched by [5]. In contrast to the work there, the event driven modeling leads to nonuniform sampling, which will be explained in the next subsections.

2.2.1 Event driven Modeling

Another approach is known from the digital logic designers, that calculate a result for each input signal variation (asynchronous simulation). This is only possible for building blocks that do not have an internal oscillation, where signal changes would remain undetected.

No time step definition is necessary for this calculation type, because every change of an input signal propagates through the building blocks to the output. Each node must therefore have a sensitivity list, where signal changes cause a computation. It is important to note that this only leads to calculations for nodes affected by the change of the input signal. Single high frequency nodes trigger only those sensitive nodes, other nodes in the system remain unaffected and don't need to be recalculated. However, these digital models lack the feasibility of handling analog signals and are not sensitive to accuracy definitions. When using a central clock to trigger each block simultaneously, the calculations for conversions from analog to digital filters from [5] can be used.

The Verilog-AMS modeling language supports wreal datatypes, which are handled in the digital domain. The theoretical accuracy is that of standard analog signals (double precision real datatypes), albeit simulated using event driven (asynchronous) methods. Every change of the wreal signal leads to a trigger event for the connecting nodes that starts the digital simulation engine. It is therefore possible to use analog accuracy in event driven simulations when the time intervals are small enough - so that the changes in amplitude between two events match the accuracy requirements.

One major problem persists when dealing with wreal datatypes in a mixed signal implementation. Electrical wires contain two types of information, potential (voltage) and flow (current). In contrast to this, wreal wires do just contain a single value, which can be interpreted as either one of those information types. It is therefore impossible to model traditional electrical characteristics like charge distribution or impedance matching with wreal datatypes without the knowledge of the corresponding impedance. It is therefore crucial to determine e.g. signal flow paths and bias points in the schematic, which can be treated either as wreal datatypes or need to maintained as electrical discipline. An example is shown in Figure 1 where signal flow wires that could be treated as wreal signals are shown in bold and bias signals that need to be maintained in the electrical discipline are drawn with fine lines. This coarse description will be refined for the demonstration example in the last but one section.

To join the discrete time and continous time domain, connect modules are inserted. These define the corresponding functions that convert an analog (continous time) to a discrete time value (either digital or wreal). A sample implementation of a E2R (electrical to wreal) and R2E (wreal to electrical) connect module is shown in Figures 2 and 3.

The default connect module E2R for a conversion of electrical to wreal signals has an input port resistance of 50 Ohm which determines the current flowing into the port by

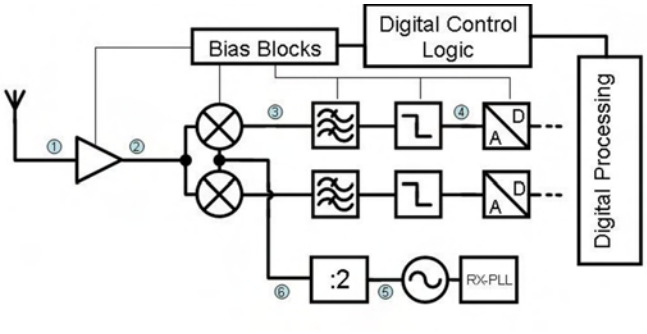

Figure 1. Typical RF Frontend

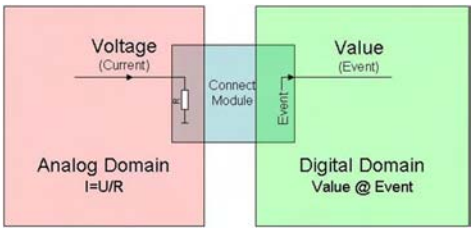

Figure 2. Connect Module E2R

the adjacent voltage. The connect module uses an *absdelta* statement to determine a change of the input voltage, releases an event and the new value at the output when a predefined difference of *vdelta* occurs. For a pure "signal flow" simulation the input port resistance is set to infinite to minimize the number of equations. Considering impedance matching as it is necessary in RF frontends, these matching dependancies must be previously known and can be calculated into the model transfer function following the connect module. More complex effects like LO feedthrough or alike are lost with this simplified signal flow. For a verification of the functionality, this is still sufficient.

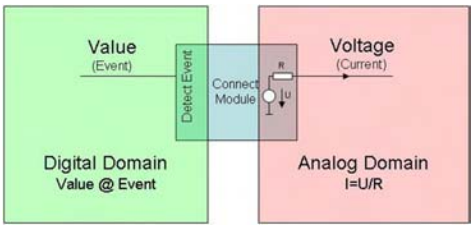

Figure 3. Connect Module R2E

The default connect module from Cadence Design Systems "R2E" for a conversion of a wreal signal to an electrical signal uses a voltage source and a series resistance of 200 Ohm for the signal output. Digital noise that might occur on the wreal wire is considered through changing the analog signal only for changes of at least *vdelta*. For pure signal flow simulations, the series resistance is changed to 0 Ohm to obtain the given voltage value at the output, regardless of the load impedance.

Both connect modules produce nonuniform sampling points for the signal, depending on the input amplitude. The following chapter investigates the necessary criterias for the definition of *vdelta* to maintain a given sampling criteria.

2.2.2 *Sampling Considerations*

In typical digital simulations a central clock generates the sample period for the signal. According to the Nyquist theorem, the sampling frequency f_s must then be at least twice as high as the signal bandwidth to avoid aliasing effects (see Fig. 4).

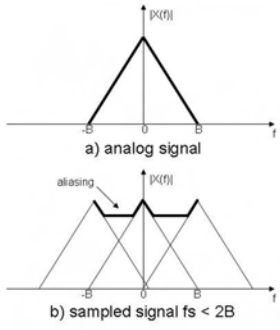

Figure 4. Sampled Signal and Aliasing.

A common rule of thumb is to set the sampling rate from 10 to 15 times the highest frequency of interest. This practice accurately models the upper skirt of a filter to about one decade above the upper cutoff frequency [3].

When dealing with event driven simulations, it is necessary to keep this sampling rate considerations in mind. Since there is no central clock to determine the sampling rate, a sampling (event) occurs only according to the signal difference. This leads to nonuniform sampling (Fig. 5). Although the signal was sampled nonuniformly, it can be perfectly reconstructed. This was proven by Freeman et.al. in 1965 [2]. A function f(t), bandlimited to $-B/2 \leq \omega \leq B/2$ (where $B = 2\pi/T$) can be uniquely reconstructed from a set of samples which are nonuniformly spaced but satisfy the condition that there be precisely N distinct samples to every interval of length NT where N is some finite integer. For our verification approach it is not necessary to do perfect reconstruction and would furthermore lead to too much effort. In the proposed approach a simple linear reconstrunction is done between the sampling points (see section 2.3).

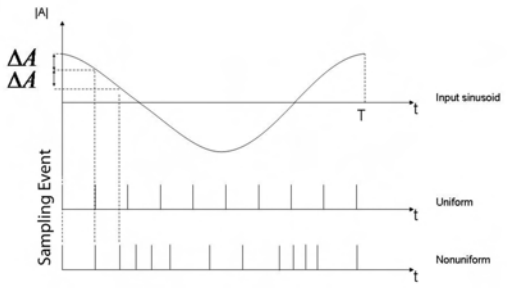

Figure 5. Uniform and nonuniform sampling times.

Assuming uniform sampling, the amplitude difference *vdelta* of a sampled sinusoidal signal between two time samples is

$$vdelta = \Delta A = A_0 \cdot [cos(\omega t) - cos(\omega(t + \Delta t))] \quad (1)$$

with $\Delta t = 1/f_s$.

In the following formulas, the resulting oversampling ratio for a cosine signal allowing a maximum signal change of *vdelta* per sample is calculated and hence at least the accuracy from the comparable uniform sampling is guaranteed. This is done by calculating the minimum *vdelta* that occurs for a given uniform sampling rate and applying this *vdelta* to the amplitude-dependant sampling.

For a single cosine signal with the frequency ω_0, *vdelta* will become minimal at

$$t = 0, \quad (2)$$

since this is the minimum flat point of the curve. Since the cosine signal is periodic and even, this point is periodic with π. In the following, only the first quarter of the cosine signal is observed.

Assuming a minimum (uniform) sampling resolution of

$$\Delta t = \frac{T}{N} = \frac{2\pi}{\omega_0 \cdot N} \quad (3)$$

with the uniform oversampling factor N ($\geq$ 2 to satisfy Niquist criteria), this leads to the desired minimum amplitude difference ΔA that will occur during one period T:

$$\Delta A_T = A_0(1 - cos(\frac{2\pi}{N})). \quad (4)$$

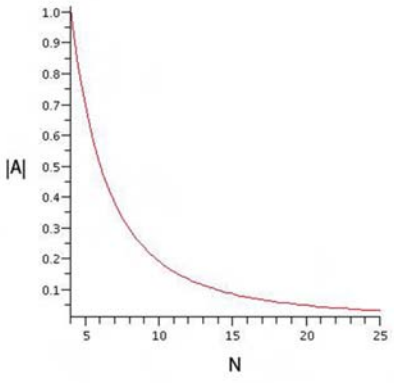

Figure 6. Necessary vdelta for uniform oversampling ratio N.

Due to the fact that the cosine signal has a different steepness over one cycle, an additional nonuniform oversampling will occur for one complete period.

Since the signal is amplitude limited, the resulting nonuniform oversampling factor will be

$$OSR_{nUn} = 4 \cdot \frac{A_0}{\Delta A_T} = \frac{4}{1 - cos(\frac{2\pi}{N})} \quad (5)$$

.

The resulting number of samples per period for a connectmodule using a vdelta according to equation 4 is shown in fig. 7 for different oversampling ratios.

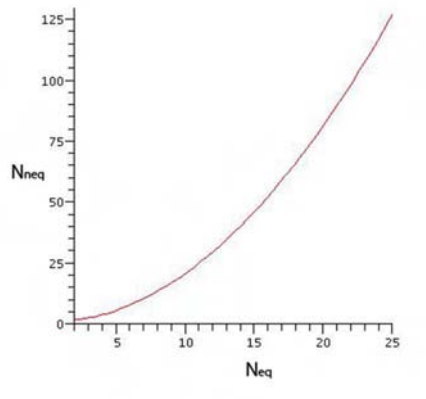

Figure 7. Resulting nonuniform number of samples N_{neq} vs. uniform oversampling ratio N_{eq}.

2.3 Different precision requirements

The LNA input and the mixer output signal amplitude differs by a factor of typically 20 dB - 40 dB. This makes it necessary to insert different connect modules with different analog sensitivities, to minimize the nonuniform oversampling.

According to section 2.2.2 an LNA input signal of -40 dBm at 50 Ohm input resistance would require an input amplitude sensitivity of $\Delta A \approx 1\mu V$ to provide an oversampling ratio of 10. At the output, the same oversampling ratio at the new intermediate frequency is generated when using a sensitivity of only $\Delta A \approx 9.5mV$ assuming a voltage gain of 40 dB.

The occuring problem is how to automatically insert connectrules for the corresponding input and output nets.

This paper focusses on the Cadence design framework and therefore uses its automatic insertion of connect modules (AICM). Up to now it is not possible to give the corresponding blocks an attribute on which the AICM routine inserts different parameters for the same connectmodule. Therefore, another approach is proposed.

According to section 2.2.2 the number of different connect modules which need to be used in the design with its different voltage peak to peak levels is determined. Each of these get labeled as e.g. $Efine2R$, $Ecoarse2R$ and has its own set of sensitivity factors implemented by the parameter "fineness" according to listings 1 and 2. The input wire needs therefore to be redefined as a new discipline in the analog domain as electricalfine or electricalcoarse.

The connect rule itself is set up as shown in listing 3. After the definitions of the desired new electrical disciplines fine and coarse, the corresponding connect modules are included and defined using their fineness parameters. The resolveto statements at the end of the listing enable the simulator to

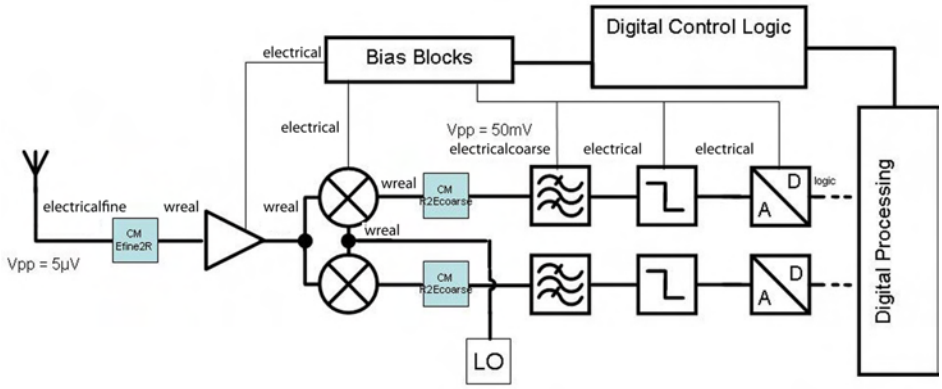

Figure 8. Setup including netset disciplines.

Listing 1. Electrical fine to wreal connect module.

```
‘include ” disciplines .vams”
‘timescale 1ns / 100p

discipline electricalfine
 potential Voltage;
 flow Current;
enddiscipline

connectmodule Efine2R (Ain, Dout);
  input Ain;
  //input electrical
   electricalfine Ain;
  // discrete domain output
  output Dout;
  wreal Dout;
  \logic Dout;
  parameter real fineness = 1000;
  parameter real vdelta =3.3/64/fineness;
  parameter real vtol =vdelta/4;
  parameter real ttol =10p;
   real Dreg;
   assign Dout = Dreg;
  always @(absdelta(V(Ain), vdelta , ttol , vtol ))
        Dreg = V(Ain);
endmodule
```

Listing 2. Electrical coarse to wreal connect module.

```
‘include ” disciplines .vams”
‘timescale 1ns / 100p

discipline electricalcoarse
 potential Voltage;
 flow Current;
enddiscipline

connectmodule Ecoarse2R (Ain, Dout);
   [...]
  parameter real fineness = 1;
   [...]
endmodule
```

tap standard electrical signals (e.g. for spice level models) immediately from the electricalfine or electricalcoarse wires.

With the new definitions, the setup shown in figure 8 is going to be simulated. The nets labeled electricalfine and electricalcoarse have been set explicitly to the corresponding discipline to enable the automatic insertion of connect modules using AICM.

3. DEMONSTRATION EXAMPLE

In a transceiver design, a couple of specifications can be made for each node. These include the most important factors signal frequency and amplitude. At the antenna (pos. 1 in Figure 1), the input signal is typically quite small in amplitude but has a high frequency. It is amplified by several dB (pos. 2) and then downconverted to an intermediate frequency (pos. 3), that is typically some decades lower than the original input frequency. At position 4 the signal was amplified again and prepared for digital to analog conversion. The necessary local oscillator frequency is generated by a phase-locked-loop (PLL, pos. 5) at twice the target frequency. A divide by two circuit generates the necessary in- and quadrature phase signals (pos. 6) for the complex frequency conversion.

The following numbers are derived from a typical bluetooth transceiver frontend.

The signal frequency at the antenna is 2.451GHz with an amplitude between -40 and -60 dBm. The first LNA raises the amplitude by about 18 dB. The mixer itself has two input frequencies from the LNA output and the LO divider (2.451 GHz and 2.45 GHz) which generate the lower output frequency (1 MHz). Due to integrated filtering, the higher sideband mixing can be neglected. The 2.45 GHz is generated from an LO running at twice the frequency to generate I/Q signals using a divide by 2 circuit. The change in amplitude for the signal is noted as 22 dB after the mixer. Filters and following limiters provide an undefined signal gain, only limited by the supply voltage of 3.3 V before entering the analog to digital converter which is clocked with 52 MHz.

The system therefore has a frequeny range of 4.9 GHz down to 1 MHz, DC biasing in the analog part and an additional 52 MHz for A/D conversion in the mixed signal part. The voltage scale of interest is from $5\mu V$ up to the supply voltage of $3.3.V$. The high frequency usually leads to high simulation times in the analog modeling, while the voltage range limits the usability of earlier event driven mixed signal simulations.

Listing 3. Connect rule setup.

```
`include " disciplines.vams"
discipline electricalfine [...]
enddiscipline
discipline electricalcoarse [...]
enddiscipline

`define Vsup 3.0
[... more defines ...]
`define Vdelta `Vsup/64
`define finenessfine 1024
`define finenesscoarse 1

connectrules ConnRules_3V_full_wreal;
  connect E2R #(.vdelta(`Vdelta), .vtol(`Vdelta_tol),
        .ttol(`Tr_delta));
  connect R2E #(.vdelta(`Vdelta), .tr(`Tr_delta),
        .tf(`Tr_delta), .rout(`Rlo));
  connect Efine2R #(.vdelta(`Vdelta/`finenessfine),
        .vtol(`Vdelta_tol/`finenessfine), .ttol(`Tr_delta));
  connect R2Efine #(.vdelta(`Vdelta/`finenessfine),
        .tr(`Tr_delta), .tf(`Tr_delta), .rout(`Rlo));
  connect Ecoarse2R #(.vdelta(`Vdelta/`finenesscoarse),
        .vtol(`Vdelta_tol/`finenesscoarse), .ttol(`Tr_delta));
  connect R2Ecoarse #(.vdelta(`Vdelta/`finenesscoarse),
        .tr(`Tr_delta), .tf(`Tr_delta), .rout(`Rlo));
  connect electrical, electricalfine resolveto electrical;
  connect electrical, electricalcoarse resolveto electrical;
endconnectrules
```

Using the proposed approach from section 2.2.2, the necessary number of computations can be minimized, so that a desired OSR (to satisfy nyquist and oversampling demands) is ensured, but not excessively exceeded.

3.1 Baseband Modeling

Using the proposed baseband modeling approach from [4] and [1], the simulation speed is supposed to be further improved. The only crucial thing about baseband simulations using the event driven approach is, that the number of digital events is approximately doubled, because each path (I/Q) triggers its own event.

4. SIMULATION RESULTS

Simulations were carried out using a Athlon 64 3200+ Linux PC with 1.5Gb of memory. For comparison, a small bluetooth receiver RF frontend with $\approx$ 150 transistors and another, more complex one, with additional 1000 transistors were simulated using the same testbench and biasing. On the left hand side of Figure 9 the simulation time for a transient simulation of $200ns$ on transistorlevel is shown. In comparison to this, the simulation time for behavioural passband models of the LNA, mixer and LO blocks are shown in the next column. The improvement in terms of simulation speed is dependend on the percentage of the size of the analog matrix that has been reduced. The posterior row has approximately ten times the number of analog equations compared to the simple receiver. The benefits of using wreal datatypes for high frequency signals can be seen in the third row. As the performance gain of the simple receiver is only about a factor of 2, it is more than a factor of 16 for the more complex receiver. This is due to the fact that wreal nets are simulated in the digital domain and the (larger) analog matrix does not have to been solved for each digital event. Using baseband behavioural models, as shown in the last two columns, even gets a lot more out of it, for the cost of information loss. The resulting oversampling ratio in the analog domain from the wreal implementation can be observed as a small raise in simulation time in the last column.

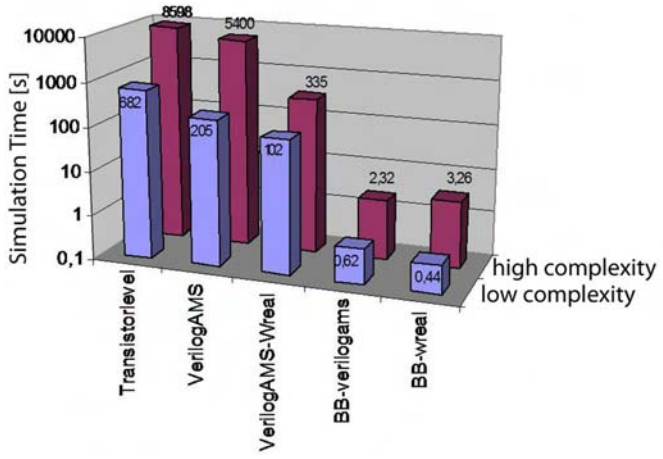

Figure 9. Simulation duration for 200ns of different configurations.

5. CONCLUSIONS

As shown in section 4, simulations using the wreal datatype improve the simulation duration for complex transceivers with a large number of analog equations. Nonuniform sampling introduces an oversampling by conversions from electrical to wreal and vice versa. To minimize the resulting oversampling ratio, chapter 2.2.2 and 2.3 proposed two methods to ensure a sufficient, but not excessive oversampling ratio of factor N. Simulation results showed a sufficient accuracy for full chip verification. In addition to the demonstrated RF frontend, additional more complex interactions between analog and digital domain are to be investigated in the future.

Acknowledgement

The authors would like to thank the German Government, represented through the BMBF, for the financial support of this work in the frame of "VeronA" 01M3079.

6. REFERENCES

[1] J. E. Chen. *Modeling RF Systems*. The Designers Guide Community.

[2] H. Freeman. *Discrete Time Systems - An Introduction to Theory*. John Wiley and Sons, 1965.

[3] V. K. Handley. Modeling analog signal processing systems.

[4] S. Joeres and S. Heinen. *Functional Verification of Radio Frequency SoCs using Mixed-Mode and Mixed-Domain Simulations*. BMAS 2006, 2006.

[5] F. J. Taylor. *Digital Filter Design Handbook*. Marcel Dekker, 1983.