Analog Modelling to Suit Emulation for Hardware-Software Co-Verification

Saranya Das

Senior Design Verification Engineer, Analog Devices, Bengaluru, India

INTRODUCTION

Boot code is the program that is burned into the ROM of the SoC (System on Chip).

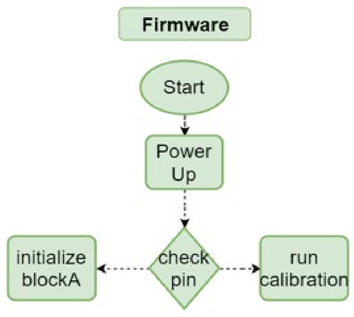
It boots up and configures the SoC.

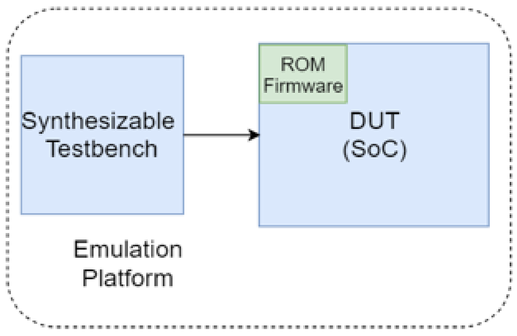
Verifying firmware routines pre-silicon is necessary because:

Any bug in the firmware code will be difficult to debug in silicon, delaying customer delivery

But the regular RTL simulation environment is very time consuming as it can go on for days. In emulation, tests can run from seconds to minutes depending on the size and code of DUT.

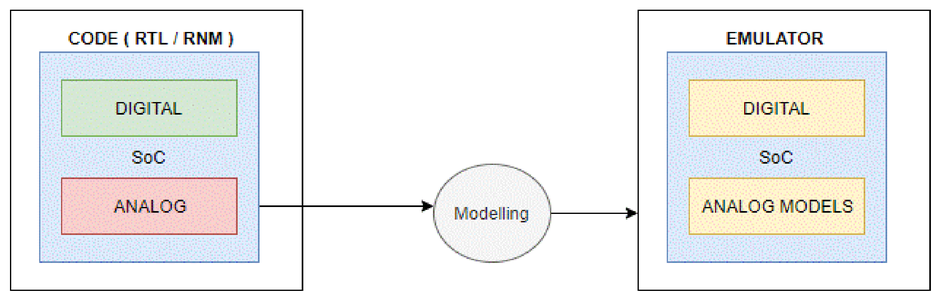
However, the main challenge is making the testbench synthesizable for which analog blocks need to be modelled.



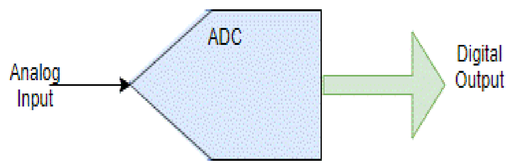


MODELLING OF ANALOG BLOCKS

Typical modelling flow is as shown where the analog circuits or real number models are converted to synthesizable models.



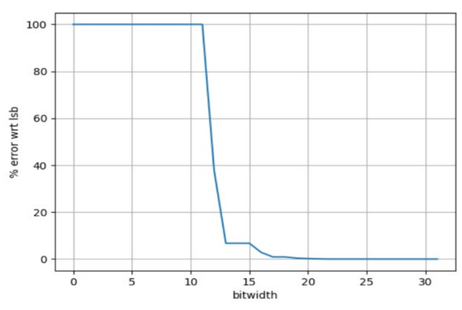
As an example, we showcase some logic of the modeling done for an ADC block which takes an analog input and gives a digital output. Data types such a float and real are non-synthesizable constructs. This was solved using **fixed point arithmetic.**



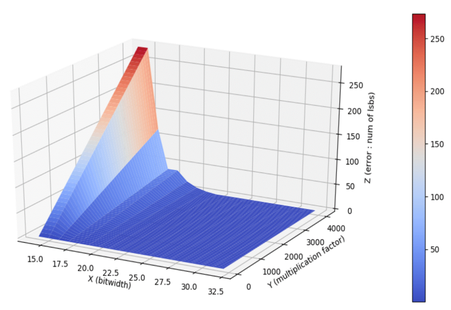

```
real vx=1.123
real va=0.234
real lsb=0.000392
real vc=1.002

vx=va-vc+lsb*256
```

Choosing fixed-point bus width by calculating error in resolution

Format	value A	Scaling factor	Input * scaling factor	Integer value	Integer/scaling factor (B)	error(A-B)/A
1.10	0.0025	2^10	2.56	2	2/(2^10)=0.00195	22%
1.20	0.0025	2^20	2621.44	2621	2621/(2^20)=0.00249	0.4 %



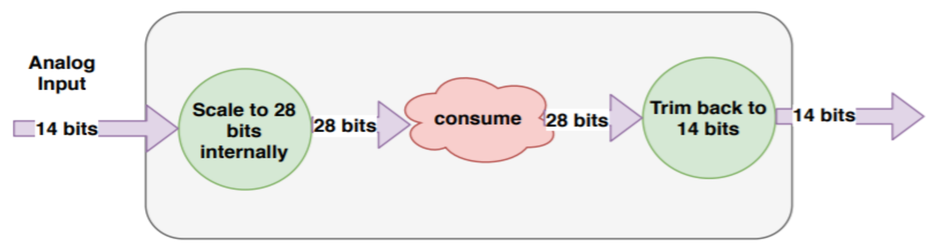
The graph shows bit-width chosen vs error resulting with the bit-width with reference to the actual real number.



The visualization shows how the error in decreases with increase in bit-width but saturates beyond a point. Hence, we take 14 as the bit-width.

Scaling to increase arithmetic accuracy

The below shows the scaling and trimming internally done to save the number of I/O pins for a module.



Consider the multiplication of two numbers 120 (‘a’) and 0.000392 (‘b’) as shown where b is the least significant bit of the ADC.

Format	(a*b)	Scaled a, rounded a (c)	Scaled b, rounded b (d)	e=(c*d)/2^(2*f)	(x-e)/lsb
1.14	0.0474	1966080,1966080	6.422,6	0.0439	8.8 lsb
1.28	0.0474	32212254720,32212254720	105226.69, 105226	0.0470	1.02 lsb

RESULTS

All the tests were run before silicon came. And this saved time in the crucial window between silicon arrival and sampling to customers.

Scenarios/Tests	Silicon Simulation time	RTL Simulation (wall clock time)	Emulation (wall clock time)
Firmware without patch (silicon use-case) Patch code is given by Firmware team to skip long running functions, reducing simulation time.	1s	4 days	1.2 hour
Chip Initialization: routines to initialize various modules/memory in DUT	500ms	2.5 days	40 min
Long running tests: Watch Dog Timer	100ms	1.5 days	10 min

Firmware Bugs	15
Firmware routines verified	6
Firmware iterations	8

As shown above, 15 firmware bugs were found pre-silicon, these issues if found post silicon, would have delayed time to market.

CONCLUSION

- To accelerate pre-silicon firmware verification, emulation is the way to go.
- It helps to uncover any surprises when the silicon arrives.
- To achieve the benefits of emulation, synthesizability and analog modeling must be kept in mind while developing the testbench in any project.
- One huge challenge in porting RTL for emulation is modelling analog blocks and this presentation shows how fixed-point modelling can be used to effectively model RNMs.
- The visualizations using python, acts as an effective tool for the reader to easily comprehend the trade-off between emulation speed and resolution error.

I would like to extend our gratitude to Raman Kumar, Sriram Madavswamy, Ponnambalam Lakshamanan, Shreya Dasgupta, Pablo Cholbi, Devaphanindra Kumar, Karan Punj and Rahul Sharma. I would also like to thank all the team members of Automotive Safety- Radar group of Analog Devices, Bangalore for their constant encouragement and support.