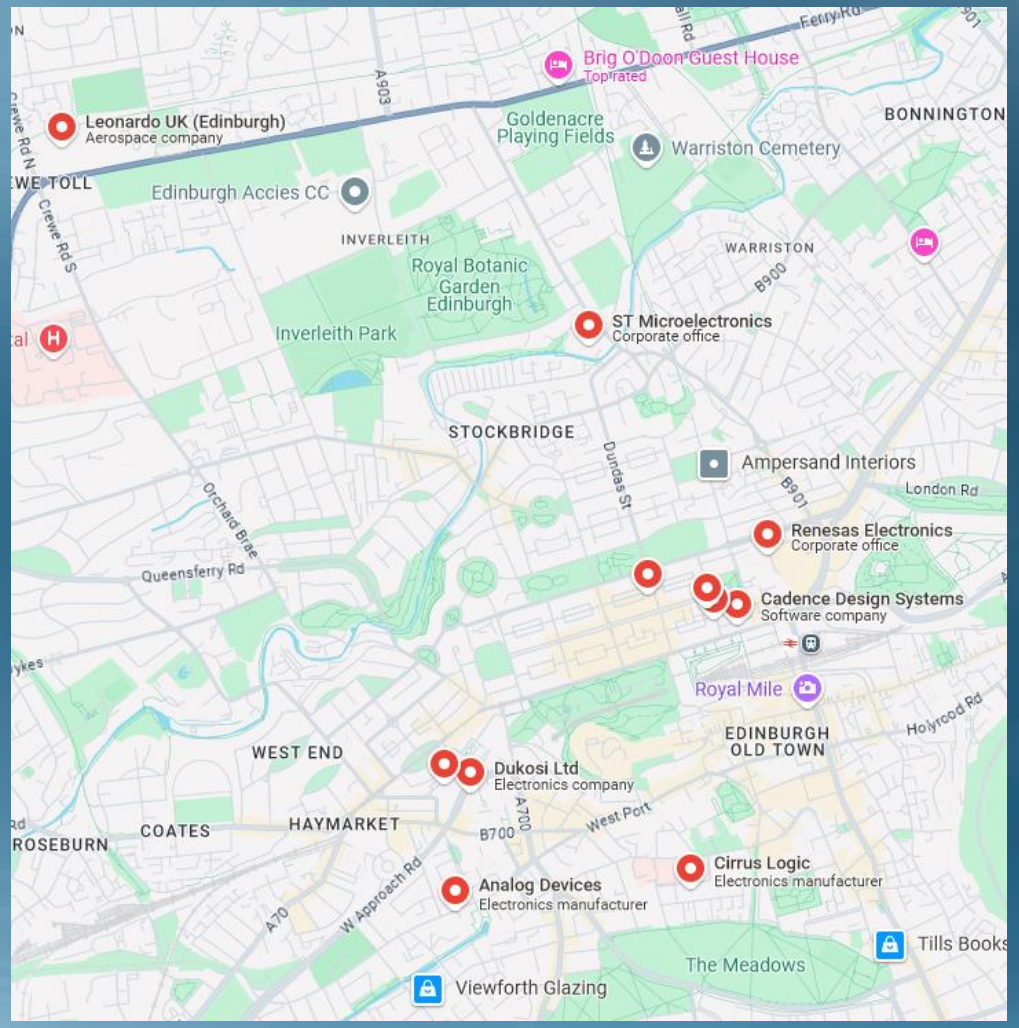
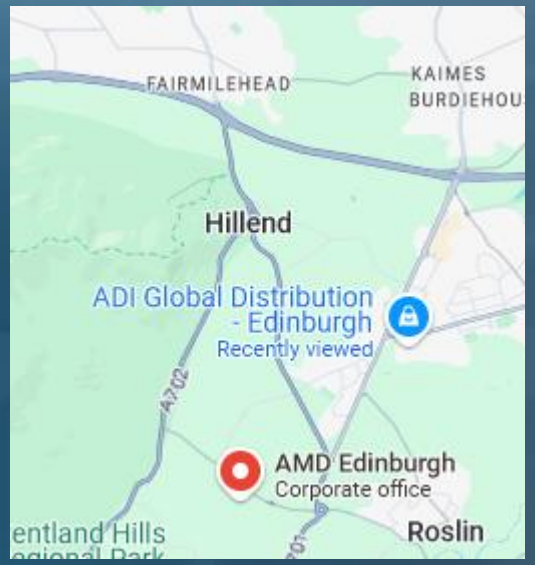
What is mixed signal verification and why you might want a *career* in it?

Graeme Nunn & Peter Grove

Companies in **Edinburgh** that do *Mixed Signal*





- ADI
- AMD
- AST Space Mobile

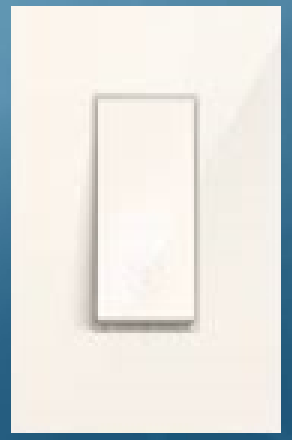
- Broadcom
- Cadence
- Cirrus
- Dukosi
- Indie
- Leonardo
- LiFi
- Renesas
- ST

Sorry if we have missed any!

Agenda

- **Back to Basics**
  - Analog/Digital Designs
  - Design Certification and Example
- **Digital-to-Analog Converter**
  - Analog Design and Verification
  - Digital Design and Verification
- **Questions before break**

Some Basic's



- Digital Circuits
  - 2 State On/Off represented as a logic1/0
  - Boolean logic/Karnaugh map
  - Event driven simulation
- Analog circuits
  - Infinite states. Is 1.0v and 1.0v+1uV the same?
  - Differential Equation based.
    - I = C * dV/dt
    - I = KB(Vds-Vth)^2
  - Continuous time simulation using matrices.

Mixed-Signal

We live in a ~~material~~ world

Historical Designs — 0 Software updates

- Analog based with probably no digital circuitry
- Very few options on top of core product function.
- Complete verification possible.
- Played one song probably from a drum of notes!

Modern Designs — 1000's of software updates

- Mixed design with CPU's, speakers, microphone's
- Complete verification of product *impossible*
- Lots of digital processing to interact with our analog World.
- Can play Oasis, book your tickets.....
- Music comes in a digital form that is converted to an analog signal for our ears!

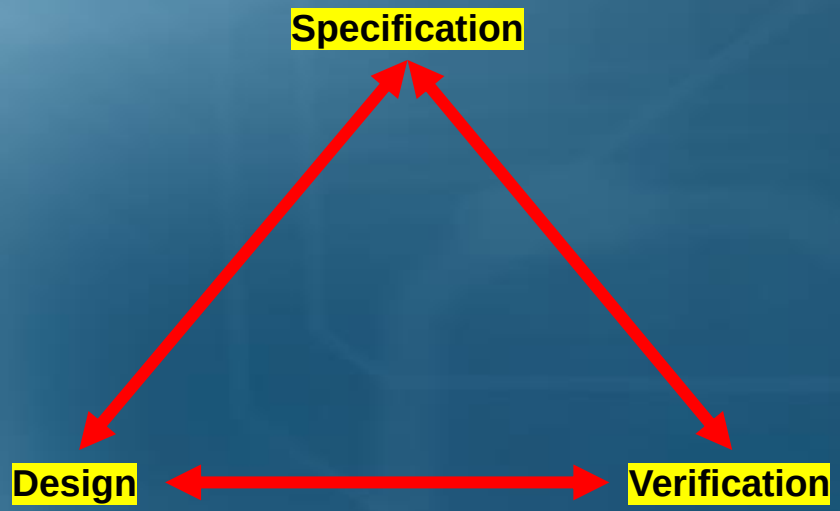
Why bother with verification?

- What makes customers happy?
  - Design meets their needs
  - Design delivered on time
- How we get there?
  - Architectural investigation
  - Detailed checking

The challenge

- Modern mixed signal designs are extremely complicated
  - Not just hardware!
  - Complete testing may need as many as five verification engineers per designer
  - Verification can easily become the gating factor in getting the design manufactured
- Modern mixed signal design flows help with
  - First time success
  - On time delivery

Design Certification Process (DCP)



A product (device) starts with a specification.

Distilled down to detailed block level specifications

Block level specification defines

What a designer should create.
What a verification engineer should test for.
Any ambiguity leads to bugs at block/system level
•Bugs can appear at integrated with other blocks

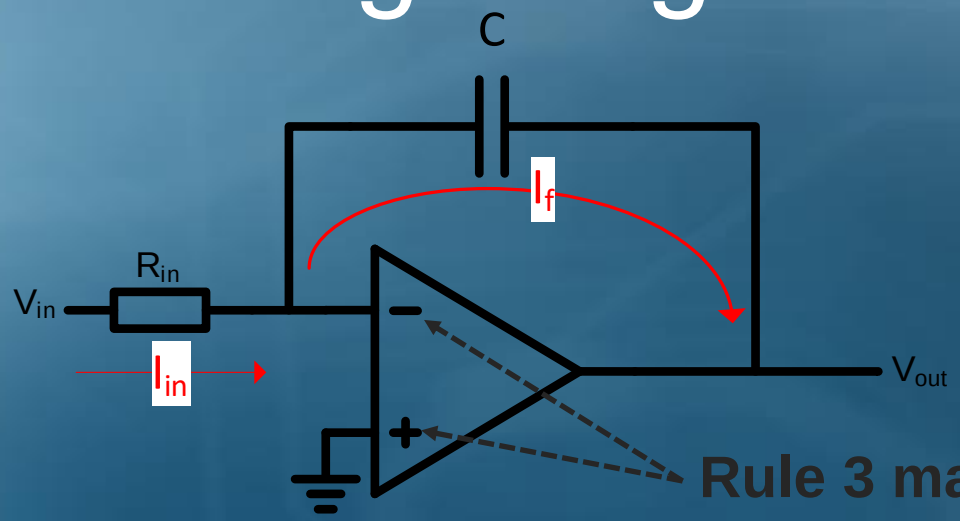
Verification treat the design is a block box

Should only use pins (io's) and not look inside.

Design & Verification should do 'what ifs' on the specification.

All three play a part in creating a robust specification.
Feedback should be ongoing throughout the process

Inverting Integrator



I_{in} Calculation

$$V_{in} - 0 = I_{in} * R_{in}$$
$$V_{in} = I_{in} * R_{in}$$
$$I_{in} = \frac{V_{in}}{R_{in}}$$

I_f Calculation

$$0 - V_{out} = \int \frac{I_f}{C} dt$$
$$V_{out} = -\int \frac{I_f}{C} dt$$

Rule 2 makes

$$I_{in} == I_f$$

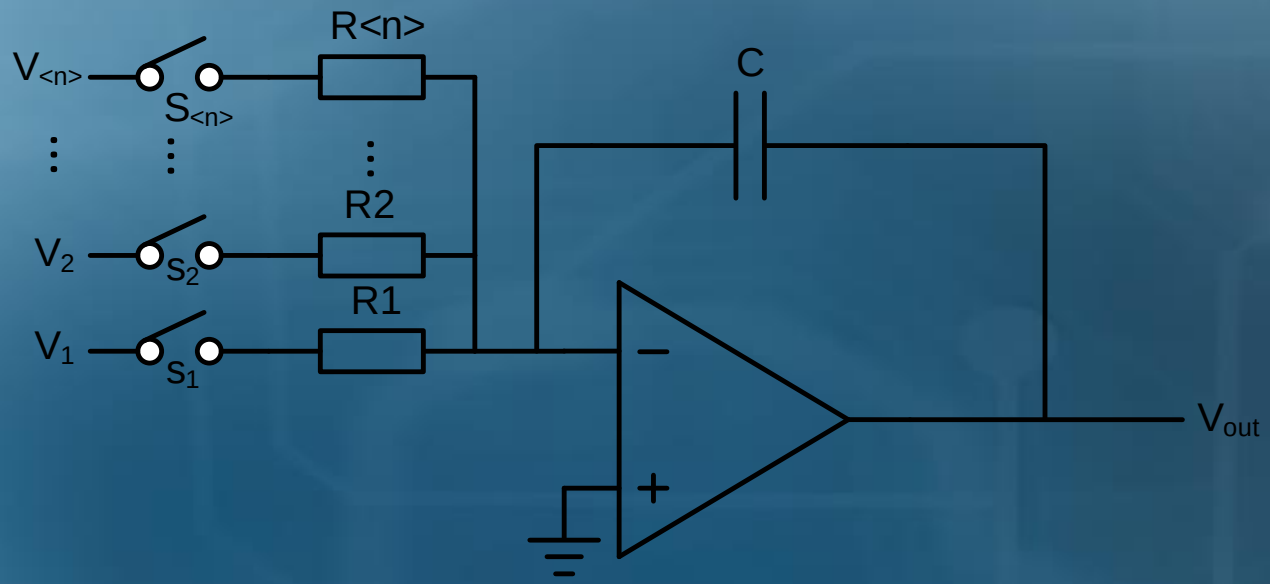
$$V_{out} = -\int \frac{V_{in}}{R_{in}C} dt$$

$$V_{out} = -\frac{1}{R_{in}C} \int V_{in}\, dt$$

Golden Rules of an Op-Amp

1. Infinite Open Loop Gain
2. No current flows into Inputs
3. No potential difference between inputs in closed loop

Digital-Analog-Converter (DAC)



$$I_{in} = \frac{V_1 * S_1}{R_1} + \frac{V_2 * S_1}{R_2} + \ldots \frac{V_{<n>} * S_{<n>}}{R_{<n>}}$$

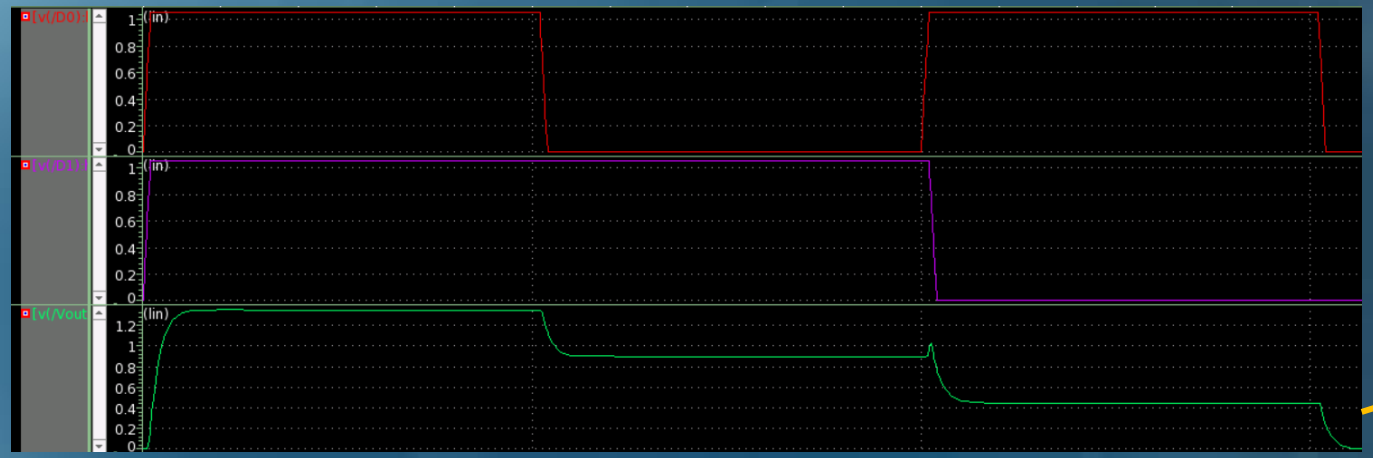
$$I_{in} = \sum_{1}^{n} \frac{V_{<n>} * S_{<n>}}{R_{<n>}}$$

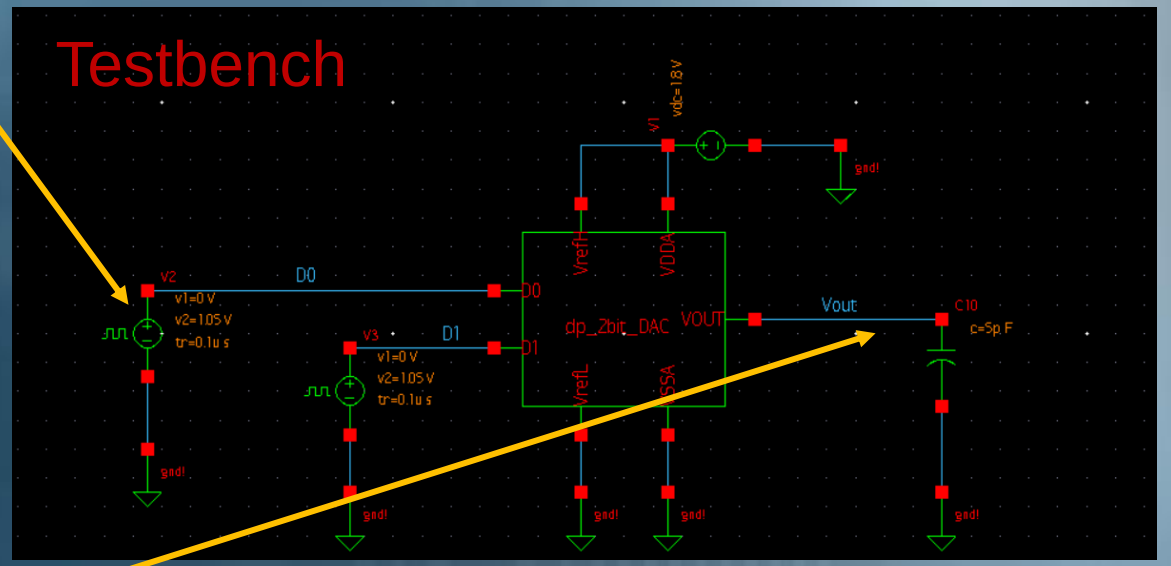
$$V_{out} = -\frac{1}{C} \int \sum_{1}^{n} \frac{V_{<n>} * S_{<n>}}{R_{<n>}} dt$$

- Assume $S_{<n>}$ are individually either on/off (1/0).
- V<n> are static Voltages
- Vout is the integration of the input Switch state, Voltages and Resistance

DAC Verification

<name> <plus> <minus> (time-voltage pair)
VS1 S1 0 t_0 V_0 t_1 V_1 … t_n V_n



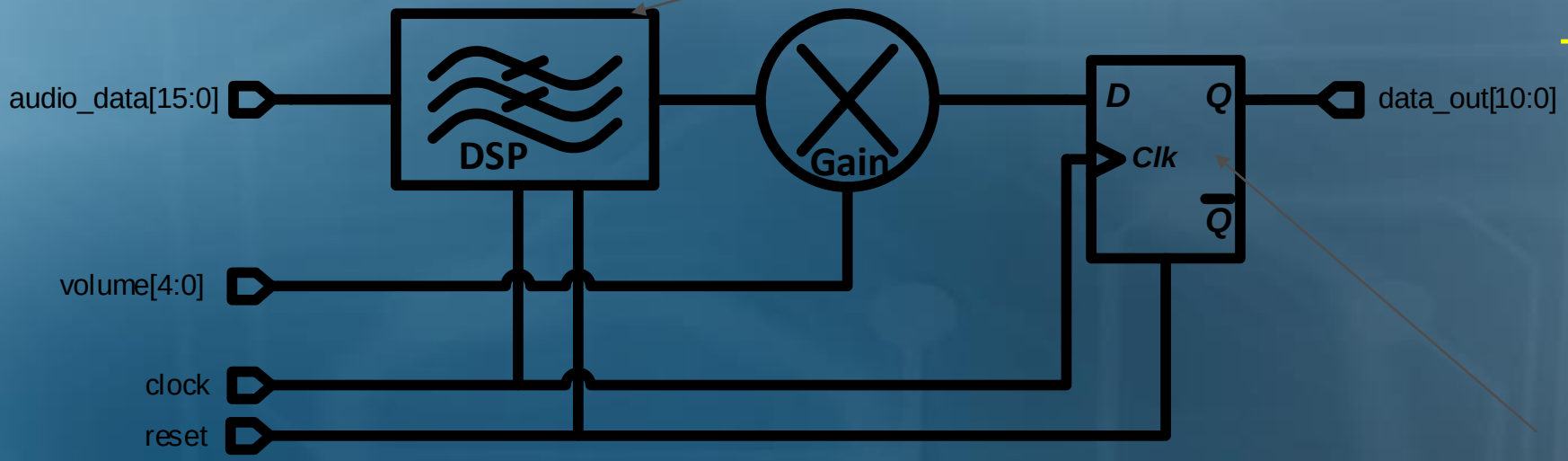


- Stimulus via Analog primitives, PWL, Static sources (Verilog-A can help)
- Checkers are often waveform based/Visual.
- Testing done by analog design and focus is performance/metrics rather functionality.
  - Have all possible inputs been tested?
- What are the assumptions here taken by the testing?

Digital Design

Digital **S**ignal **P**rocessing

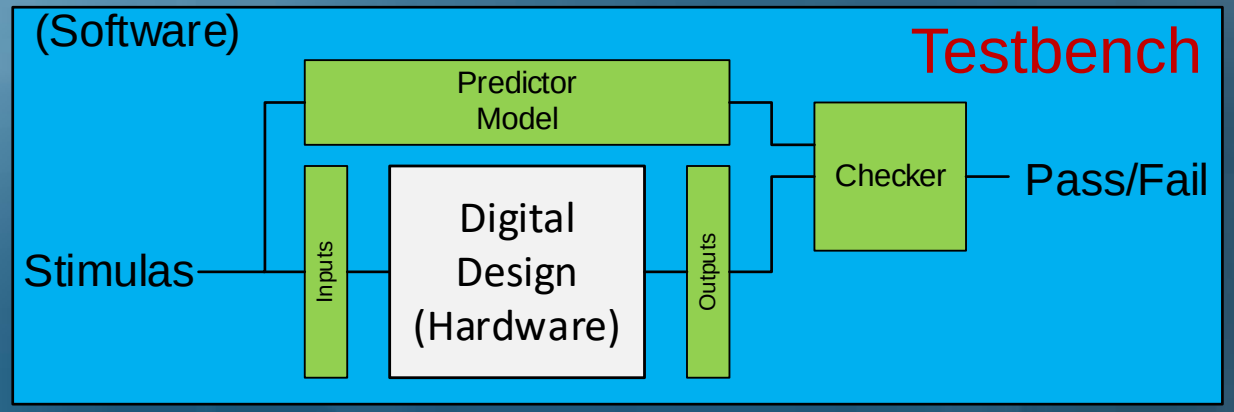
Written in Register Transfer Language (RTL) usually SystemVerilog

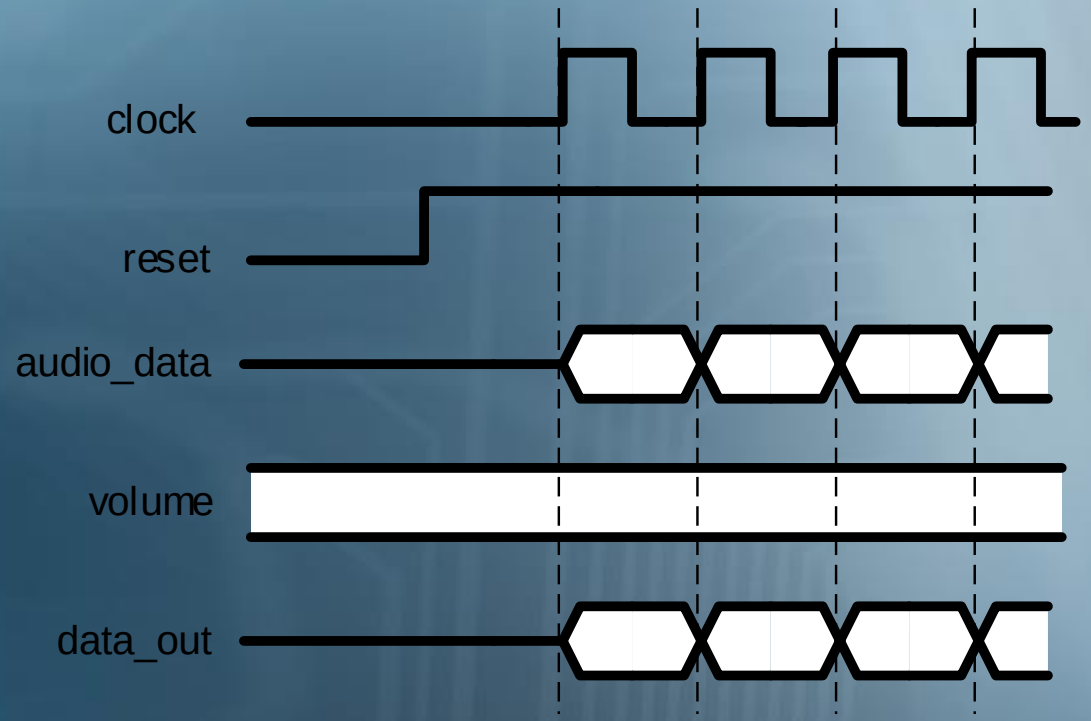


D-type Flip-Flop

- *audio_data* is a 16bit 48Khz mono channel audio stream
- *volume* controls the gain
- *clock* is the rate the circuit works at. E.g. 12.288MHz.
- *reset* holds the system in reset until is is released
- *dout_out* has 11 logic outputs to control the DAC switches $S_{<n>}$

Digital Verification





- Software based stimulus to drive Hardware components (Digital Design)
- Functionality driven rather than performance (Opposite of analog DAC)
- Can predict output based on input 100%, so self-checking.
  - Use randomisation on audio_data and volume to get high verification coverage.
- For a small block with limited combinations the digital design my verify

Question Time before the break

Agenda

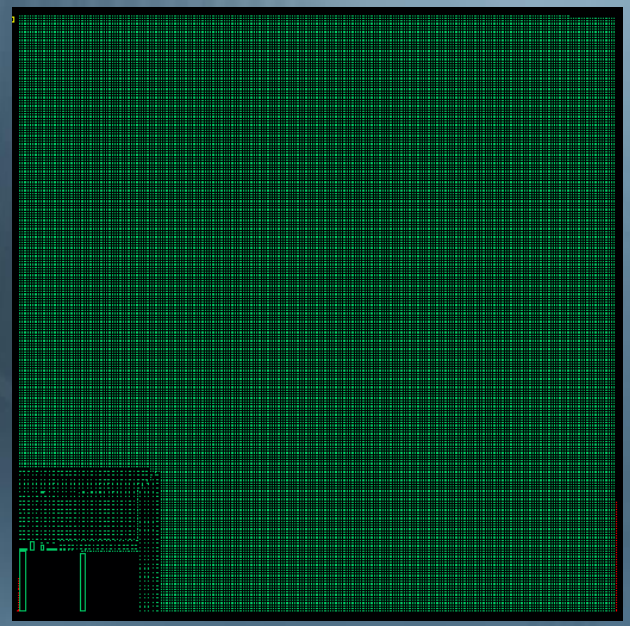
- Issues not simulating Digital and DAC together
- Possible Mixed Signal Simulation Solutions
  - Full Transistor / Partial transistor
  - Real Number Model (RNM) plus RTL
  - Spectre design or (Verilog AMS models) plus RTL
  - Speed vs Accuracy vs Scale
- Real design issues missed due to no Mixed Signal testing
- Conclusion

Issues not Simulating DAC+Digital together

- Signal Inversion. Is S set to 1 an open or closed switch?
- When should the block be taken out of reset
- Sequencing of signals
- Incorrect supplies and missing level shifters
- Any mistakes in the specification of the 2 blocks
- Invalid assumptions between designers (not my fault syndrome)
- ALL of these could lead to the whole chip being worthless

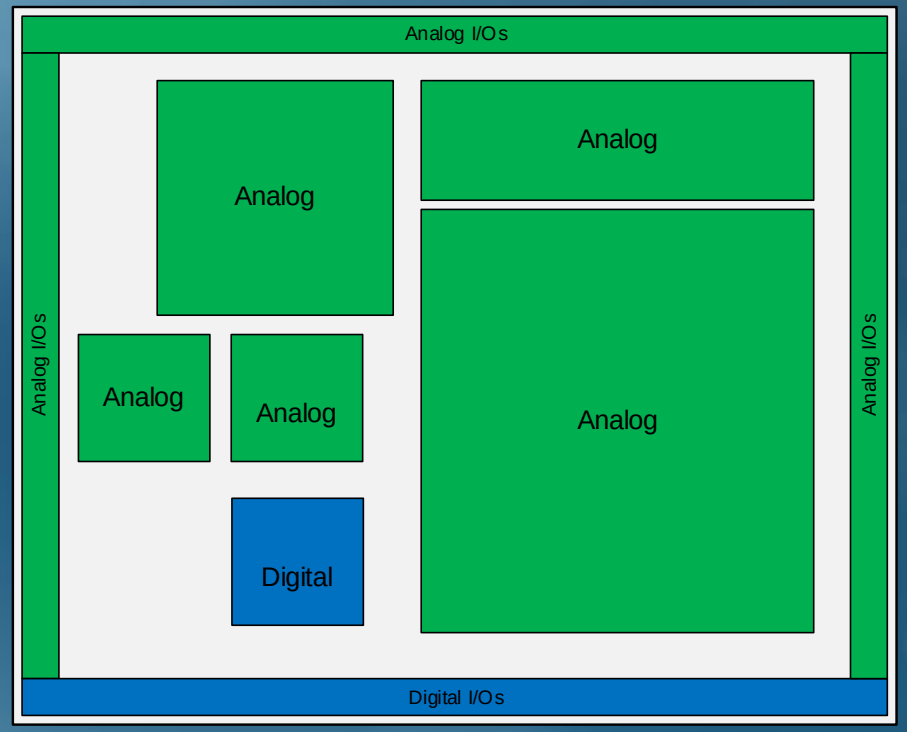
Possible Solutions for DAC+Digital Together

- Full Transistor simulation of whole design
  - Doesn't scale as the design get bigger.
  - Requires parasitic extracted digital netlist
    - Digital design has to be converted to Digital Gates (transistor levels)
- Pure analog simulation
  - Extract stimulus from digital simulations
    - Use this to "capture" digital
  - Purely feed forward. No digital to analog content
  - Hard to debug



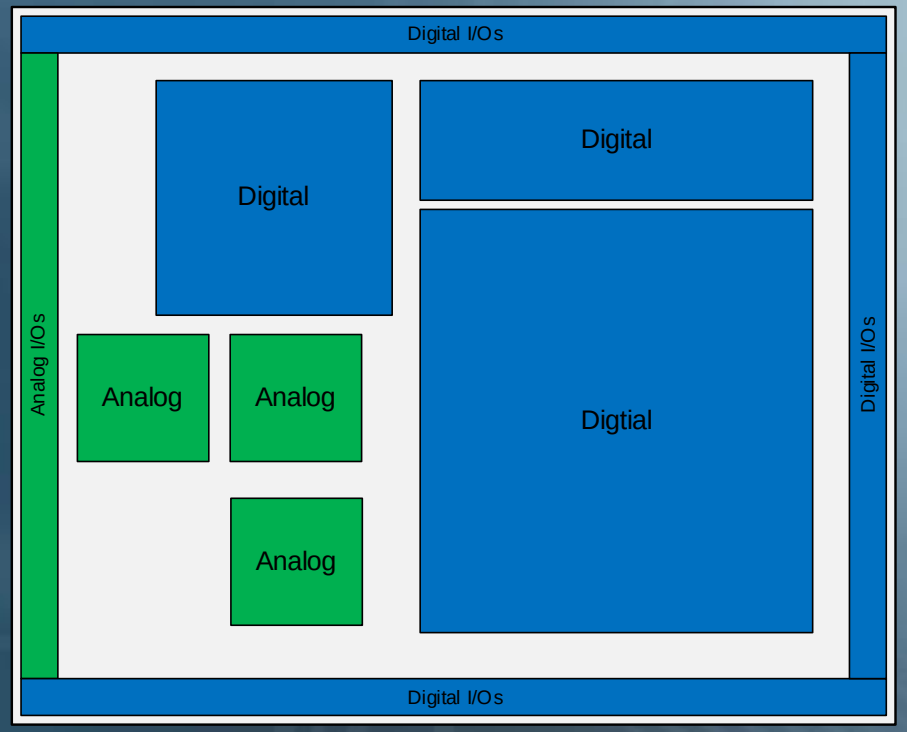
Possible Solutions for DAC+Digital Together

Big Analog Small Digital



Full transistor – Probably Not

Big Digital Small Analog



Full transistor – Probably

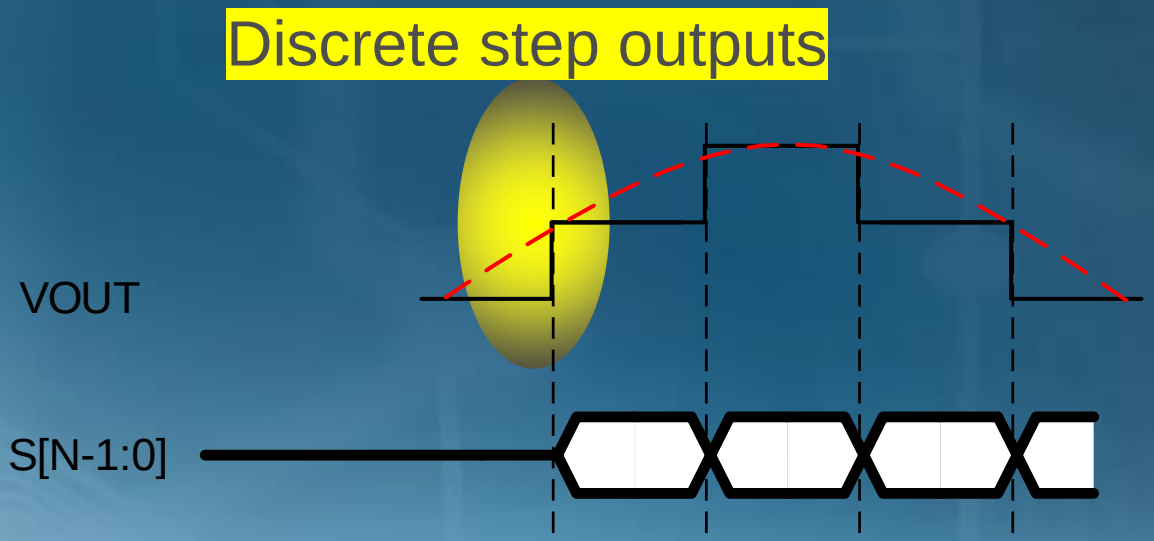
** One size does not fit all it depends, but generally ***NOT*** a solution **

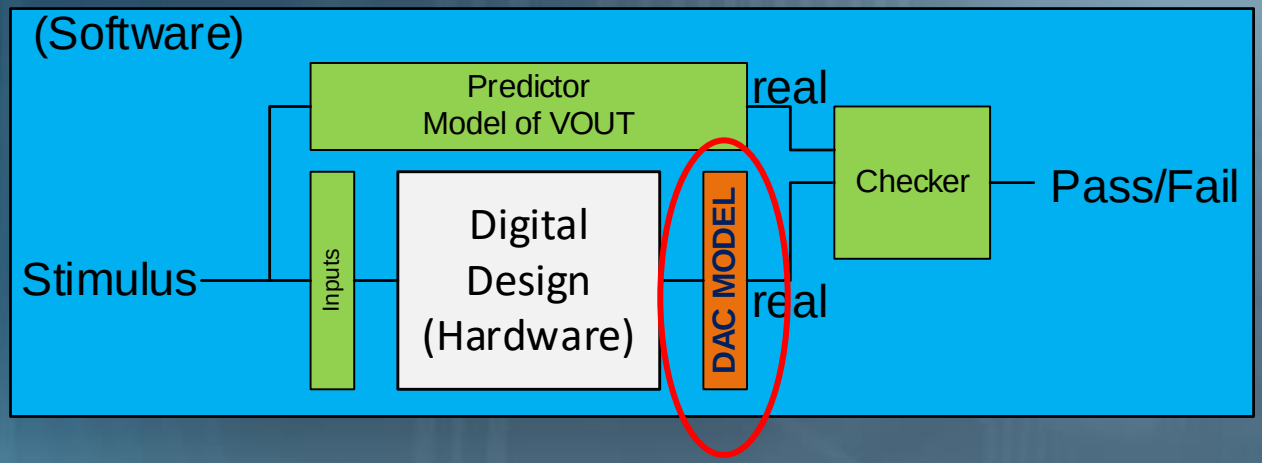
Possible Solutions for DAC+Digital Together

- **R**eal-**N**umber-**M**odelling of the DAC to simulate with the digital.
  - Pro's
    - Filters out basic mistakes. E.g. signal inversion, wiring mistakes.
    - Uses digital simulator so fast.
    - SW based testbench, with self checking.
    - Can scale to incredibly complex designs
  - Con's
    - How do we know the model is correct. (Model vs Schematic)
    - Hard to model some complex analog behaviours
    - Few have the knowledge. (automated tools still not good enough)

Possible Solutions for DAC+Digital Together

```
always@( * ) begin
    vout = 0.0; //Reset output calculation
    //Calculate next output
    for(int i = 0; i<=N; i++) vout <= vout + S[n]*V[n];
end
```



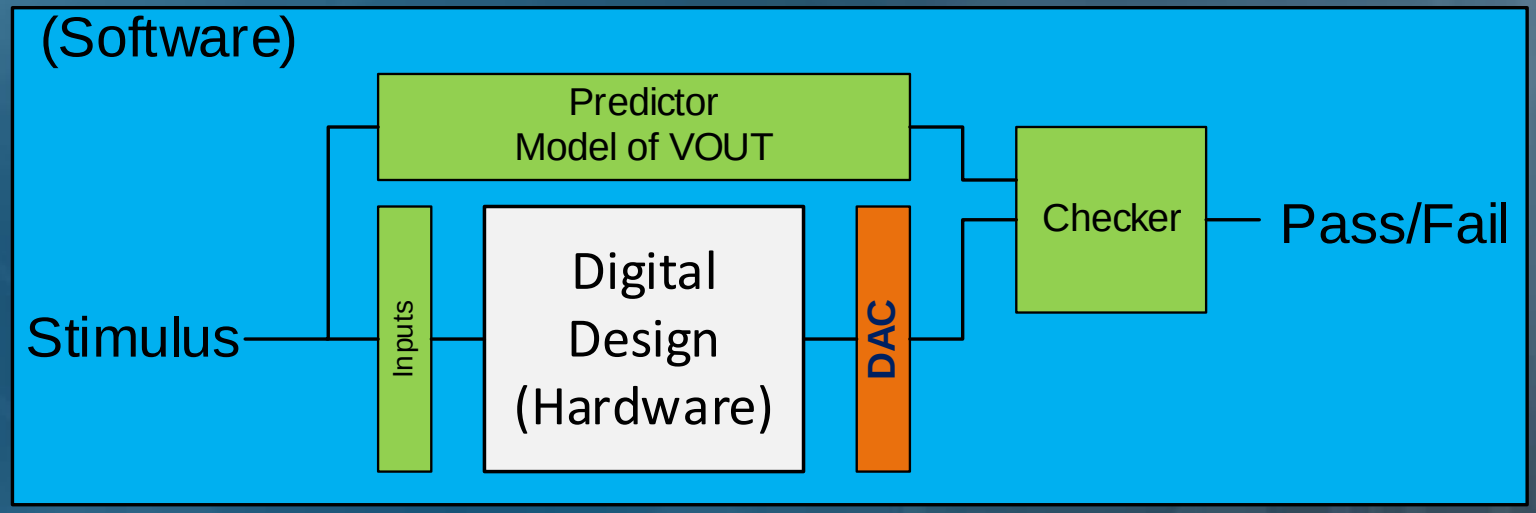


Possible Solutions for DAC+Digital Together

- Analog-Mixed-Signal (AMS)
  - Pro's
    - Has the flexibility to capture all analog behaviours
    - Can be faster than pure analog simulations
  - Con's
    - Speed is slower than RNM+RTL
    - Scalability is not as good as RNM+RTL
    - Few have the knowledge

Possible Solutions for DAC+Digital Together

- **AMS** allows users to mix many languages/solvers into a single simulation
- DAC could be a RNM, Analog Model, Transistors

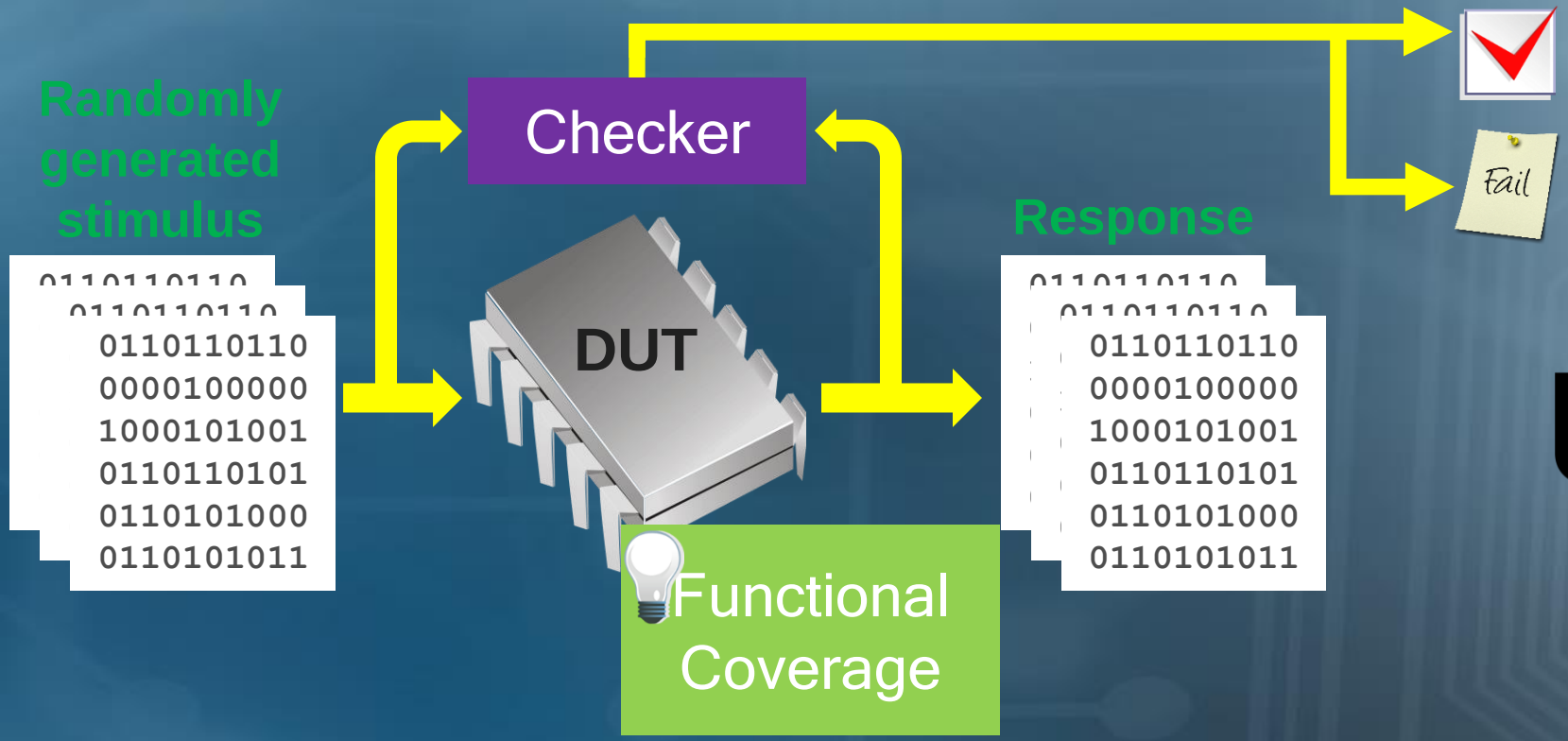


UVM MIXED-SIGNAL

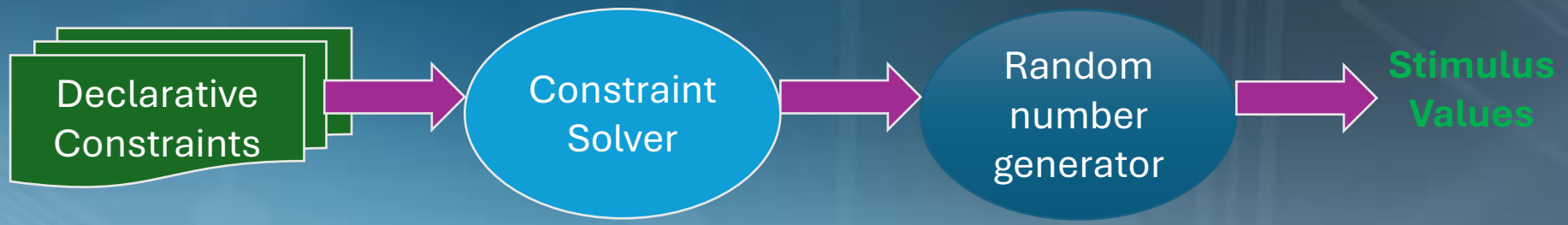
"Could we have **one** Testbench and change just DAC implementation?"

- Stimulus/Checkers/Predictor could still be 'software'
- Change abstraction of analog circuitry
- Scales with EDA technologies

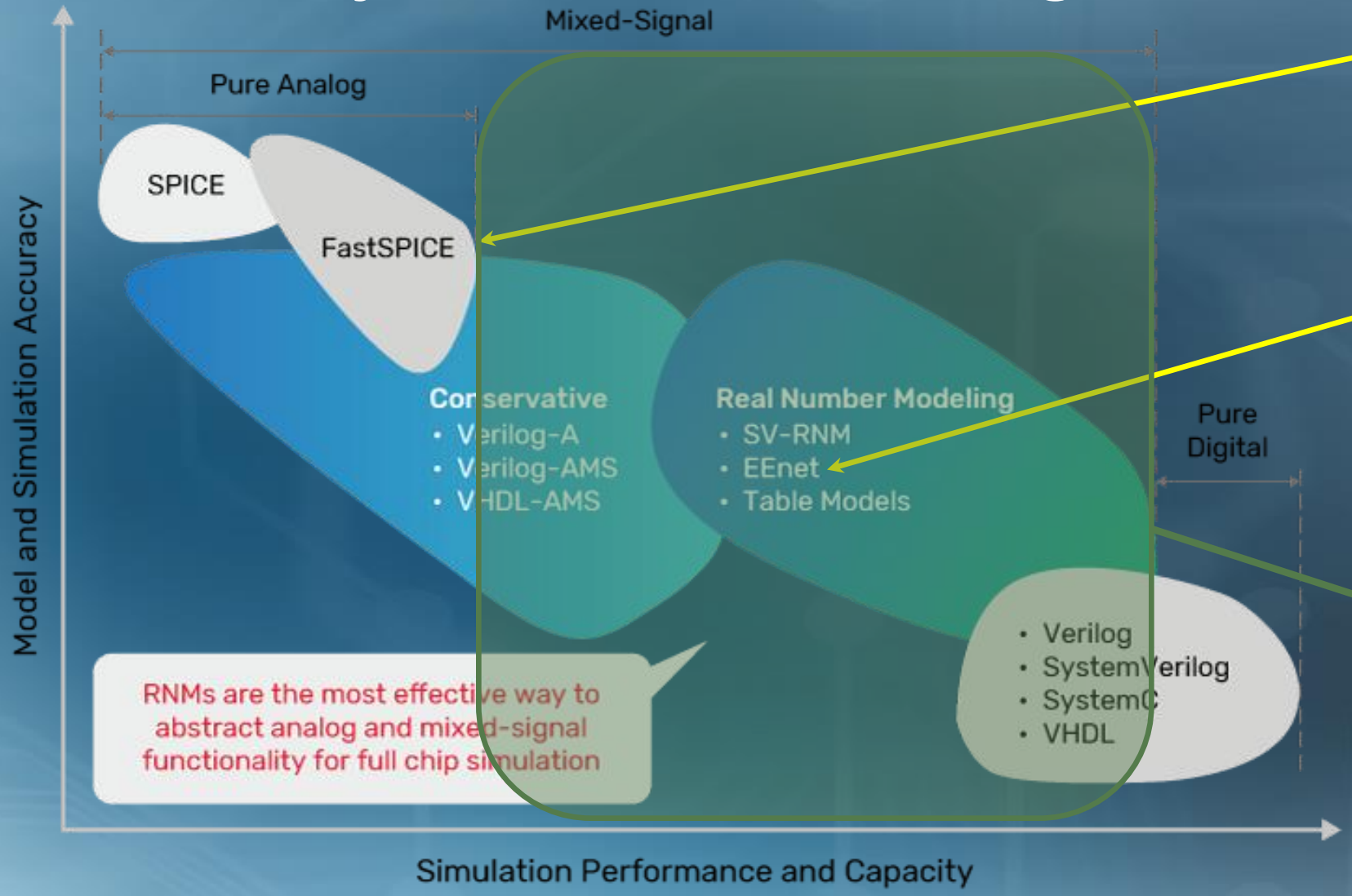
Overall Testbench



UVM MIXED-SIGNAL



Summary - *Cadence* Diagram



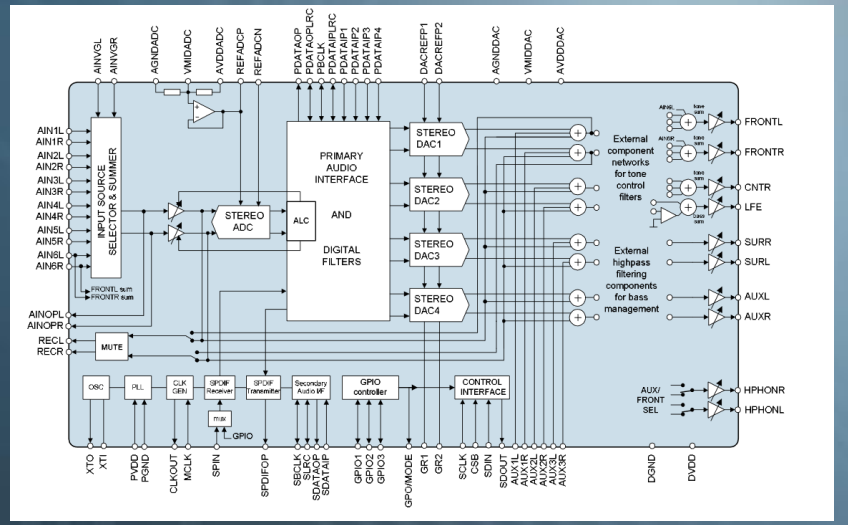
Spice/FastSpice is getting better with GPU's, multicore options

SystemVerilog User-Defined-NetType

Area of innovation by the industry.

Real Life (old) Example

- Specification missed some big use cases
  - ADC and DAC could run at different audio rates
- Design Bugs
  - Someone used powerdown instead of enable
  - Counter caused glitches in analog
  - A2D not captured at right time



- Verification missed many of the bugs, as focus was design!

Question Time

- Mixed Signal verification is still a developing technology with methodologies been patented by companies.
  - Analog Verification has not had a step changed in decades
    - No industry stand way.
  - Digital use Unified Verification Methodology (UVM 10+yrs old)
  - MS use UVM-MS (month old) (still innovation in this area.)
- Useful tools which can help throughout the design flow from architecture to final design off
- We need the next generation of *graduates* to enter this field.

Verification Future's Reading UK

- Call for Mixed-Signal abstracts, 1st July, Reading University
  - https://www.tessolve.com/verification-futures/vf2025-uk/
  - **Abstract Deadline:** 17th February 2025
    - To: VFutures at **vfutures@tessolve.com** or Mike Bartley at **mike.bartley@tessolve.com**
    - Subject: "Call for abstract – VF2025 UK – Mixed Signal Verification Track"
- Mixed videos from previous events.
  - Mixed-Signal Randomisation - Stimulus and Checkers
  - Renesas's Submission to the UVM-(A)MS working group
  - A Mixed-Signal Universal Testbench for RTL/RNM/DMS/AMS (UTB)
  - https://www.youtube.com/@TestAndVerification