Mind the Gap: Ensuring a smooth passage from Matlab to FPGA code

David Wilson & John Bunton
Norbert Abel
“Mind the Gap” Development tools

High level modelling environment
Matlab/Simulink
Maths with “special functions”

Logic gates
Connections
clocks

(this is me)

(this is Norbert)
Block diagram (wide!)

Script editor

Quantisation results

Matlab

Need lots of screen real-estate
What I showed last time ...

Why we need a Signal Processing model

- To show the algorithms meet requirements
  - At least at high precision
  - Rapidly review spec changes
- To justify that the simplifications are reasonable
  - To inform design
- To optimize numerical precision

- Assist the firmware & hardware designers
- Assist in debugging/system verification

Feb 2017

What we’ll talk about today
Key challenge identified by CSP.LOW lead
High level model with quantiser hacks

Quantisers: Required for data transfer & to/from memory
What I showed last time...

Model Variants: Golden, Fixed

Golden Model
- Double precision (if this doesn’t work, we need to vary)
- Not considering any higher
- Implement in Matlab (>IEEE 754, 64~80bits)
- Status: essentially complete, no design choices

Fixed precision
- Double/Single/quantised
- Can vary individual blocks to arbitrary fixed-point notation
- Many design choices: Will need advice

Packetised
- Consider startup transients and buffer fills
- Independent of precision

In our analysis, where this should more strictly be in the LIFR-1 component. Hence our 1% correlator loss assumption should therefore be viewed as a conservative requirement.
Consider 2 simple tasks

1. A complex exponential (part of the SkyGen)

2. An FFT (channeliser)
The FFT version (in FPGA friendly code)

Doesn’t look too bad ...
But inside the FFT block!

- Lots of lines!
- Lots of lines going backwards!
Our bridge-building challenge:

• Does the deployment work as advertised? (Xilinx/Altera/HDL)
  • Ask Norbert

• How do we best organise/layout our Simulink blocks?
  • Continuous flow vs packetized
  • Agree on a layout
  • Data valid flags, debugging-blocks

• What makes it difficult to produce FPGA code?
  • Special functions? Complex indexing?

• Must we change our thinking?
  • Who needs to change? (e.g. Norbert)