The Backchannel Crossroads

Ken Willis, Ambrish Varma, Kumar Keshavan, Brad Brim

IBIS Summit – June 2014
Santa Clara

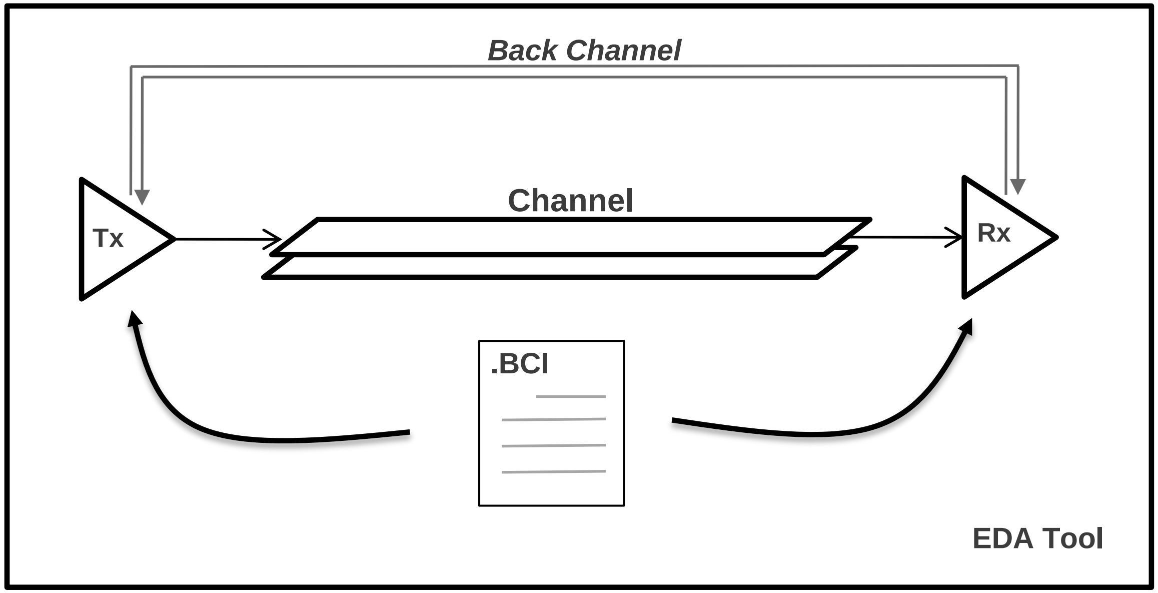
Agenda

- BIRD 147 review (Readers Digest version)
- The SISoft proposal
- Contrasting the two
- Summary

Agenda

- ***BIRD 147 review (Readers Digest version)***
- The SISoft proposal
- Contrasting the two
- Summary

Both Tx and Rx AMI Models point to a “BCI” File to Define the Backchannel Communication

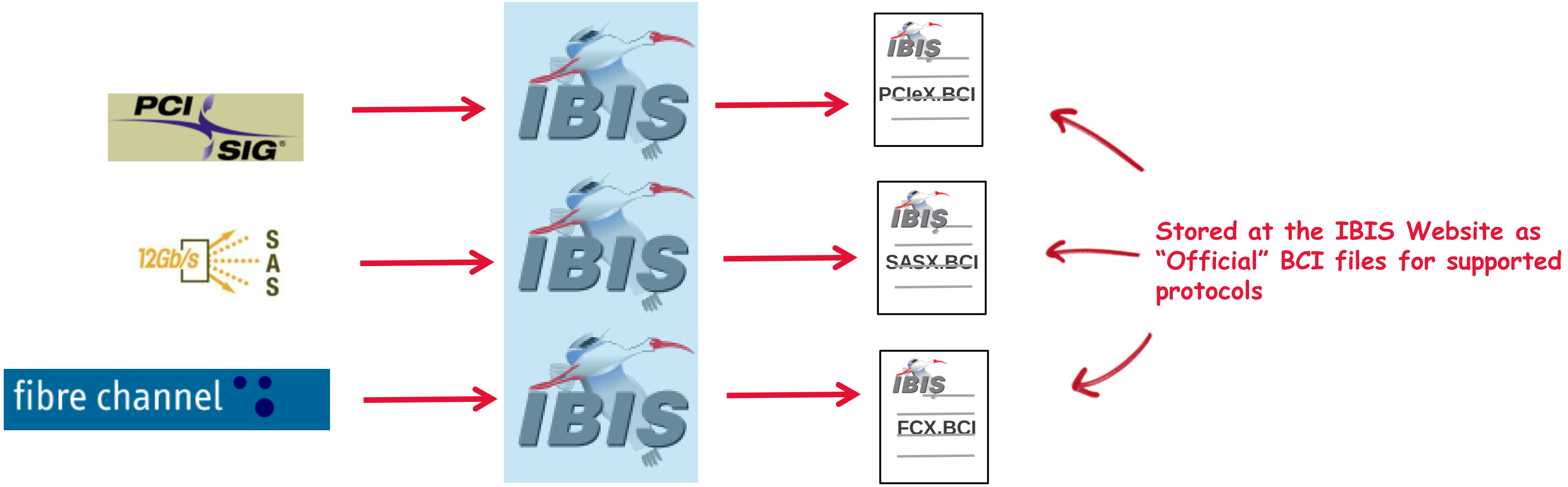


- Just two new Reserved_Parameters needed for AMI models:
  - Training > turns backchannel training on/off
  - Backchannel_Protocol > pointer to the specific BCI file being used
- BCI file contains protocol-specific info:
  - Training pattern and length
  - Min number of Tx taps available for training

Why is this a good proposal?

- Tx and Rx AMI models remain “black box” entities (as originally intended for AMI)
  - AMI modeler free to put whatever they want in the black boxes (as always)
  - Can evolve as industry needs, without an (annual) IBIS spec uprev
- The role of the EDA tool is simply to pass parameters back and forth between Tx and Rx
- Protocol_Specific section for backchannel parameters compartmentalized in BCI file
  - Analogous to “Model_Specific” section of “.ami” file
  - Isolates the protocol-specific churn to that section, which can be modified and released without an (annual) IBIS spec uprev or Golden Parser update
  - Proprietary backchannel protocols are easily supported, with no IBIS spec churn
- Simple, elegant, and extendable, which is typically what works over the long haul

The BCI File – Who, Why and When



Stored at the IBIS Website as "Official" BCI files for supported protocols

- 'IBIS Approved' BCI Files will be produced every time there is a need/demand of supporting a new BC protocol
- Eliminate any interpretation errors between what is supported and what gets implemented by the Tx/Rx/EDA tools
  - The model makers otherwise will have to read relevant sections of the protocols to understand, interpret and implement the BC communication.

‘Private’ BCI File

- Company A can make a ‘private’ .BCI file and send it to Company B to disclose what/how the Backchannel interaction is performed for their Tx/Rx.
  - Convenient, wholly contained
  - Separate from .AMI
  - Works in the same fashion as published protocol Backchannel training.
  - EDA tool does not need to do anything different for private .BCI.

Agenda

- BIRD 147 review (Readers Digest version)
- ***The SISoft proposal***
- Contrasting the two
- Summary

The SISoft Proposal

- Hard-code the Tx AMI model
  - They are all the same, aren’t they?
- Then hard-code the Rx AMI model with a few dozen back-channel Reserved_Parameter keywords
  - Leaves NO room for anything other than general optimization.
- If a new backchannel protocol comes up, add some more Reserved_Parameter keywords in an (annual) IBIS spec update
- Wait for a new Golden Parser to be funded, developed, and tested
- Lather, rinse, repeat

Agenda

- BIRD 147 review (Readers Digest version)
- The SISoft proposal
- ***Contrasting the two***
- Summary

Contrast #1 - Backchannel Protocol Definition

- BIRD 147
  - Isolated to Protocol_Specific section of BCI file
  - Both Tx and Rx point to it, so no conflicts
  - BCI file flexible to accommodate whatever new parameters are required to pass between Tx and Rx
  - EDA Tool participation limited to opening a channel for communication

- SiSoft proposal
  - Hardcode backchannel parameters into the Tx and Rx AMI models with many Reserved_Parameter AMI keywords
  - When a new protocol comes up, go through a full (annual) IBIS spec uprev
  - Then go update the IBIS Golden Parser
  - Active EDA tool participation in the communication using Reserved Parameters

Contrast #2 - The AMI “Black Box” Principle

- BIRD 147
  - An AMI model is a “black box” owned by the AMI model developer
  - We don’t define its contents; this is left open and flexible to accommodate whatever EQ technology may develop
  - We won’t pretend to know every innovation that could occur
  - Allows vendor specific ‘Private Protocol’ backchannel
- SiSoft proposal
  - Go ***inside*** the black box and hardcode its contents
  - Remove the ability to innovate and develop something slightly different
  - We are smart enough to know there will never be any innovation in this area
  - NO scope for Private Protocols

A Look Back …

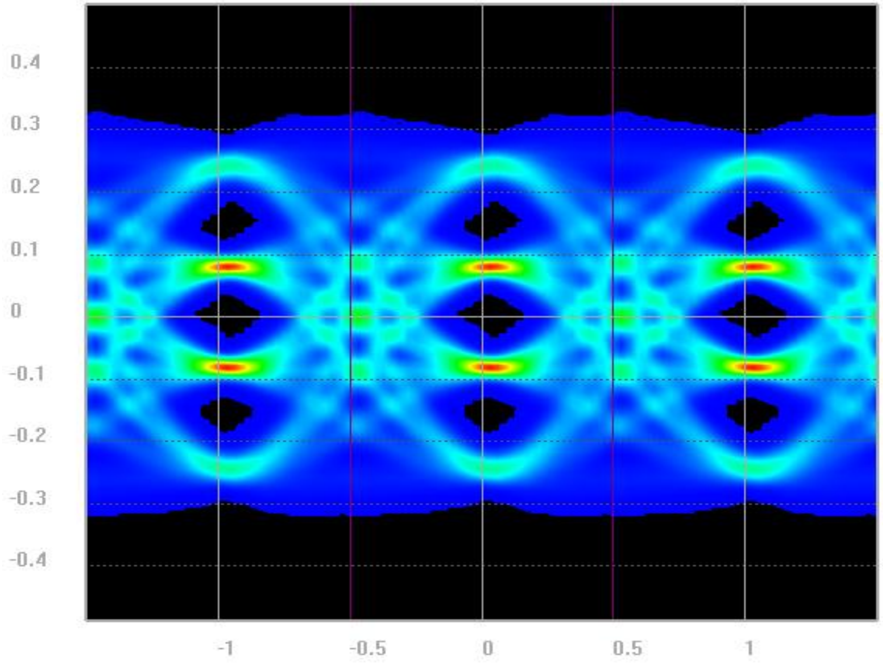
Where has hard-coding and keyword explosion limited us before?

- IO buffers
  - Went beyond new keywords to [External Model] general syntax
  - Analog BIRDs 116 – 118 later passed to enable generic SPICE “ISS” syntax
- Packages
  - Lumped per-pin package parasitics became distributed “Package Models”
  - Then decided we needed generic SPICE “ISS” syntax

- Do we ***really*** think hardcoding multi-gigabit equalization models is the right approach?
- Black box / general syntax has ***always*** proved more durable than hardcoding

What could go wrong with hard-coding?

- Supplier “x” directing exactly how supplier “y” will set its parameters
- Irregularly spaced sub-UI taps
- Pattern dependent EQ algorithms
- Adaptive analog EQ (ex. with Tcoils or caps turning on or off)
- Proprietary protocols
- PAM4
- WHO KNOWS WHAT ELSE?



?

Summary

- Not at all comfortable with hardcoding the contents of AMI models “inside the black box”
  - Haven’t seen this as a successful approach yet
  - Very slippery slope
- BCI file gives us an opportunity to:
  - Maintain AMI models as a “black box”
  - Compartmentalize the churn associated with a given protocol to an editable section of the BCI file
- This will keep the ability to innovate in the hands of the IP suppliers:
  - Maintain status quo of “black box” AMI models
  - Enable new back-channel protocols to be supported without a new IBIS spec uprev, or a new Golden Parser

BIRD 147 is Simple, Traditional Backchannel Communication

cadence®