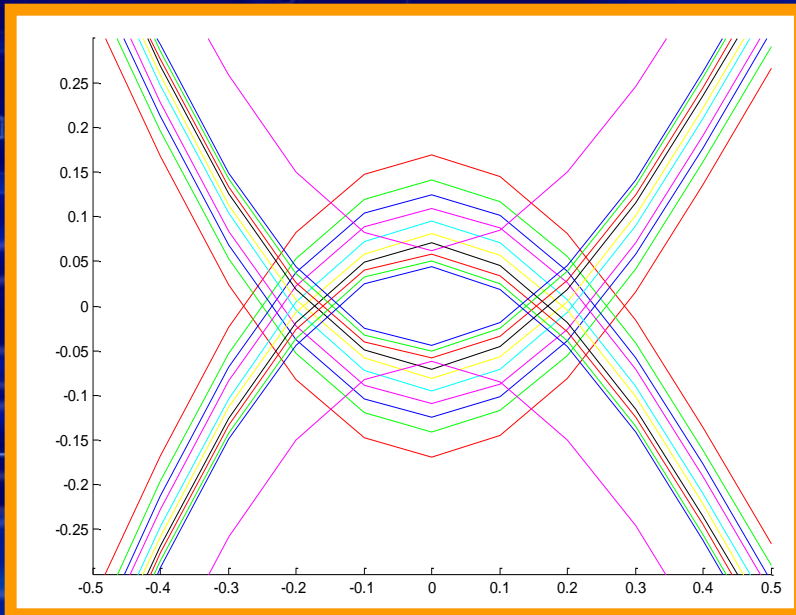


Back channel vs. Co-optimization

***IBIS ATM Teleconference
July 1, 2014***



Arpad Muranyi

© Mentor Graphics Corp., 2014, Reuse by written permission only. All rights reserved.

Mentor Graphics®

Introduction

- **The discussions on BIRD 147 and the corresponding SiSoft proposals came to a stalemate**
- **A vote attempting to make a group decision ended up with a tie in the IBIS-ATM meeting on May 27, 2014**
- **Two major member companies have different preferences**
 - Intel prefers BIRD 147 because it supports proprietary communications between Tx and Rx
 - Altera prefers the SiSoft proposal because it supports co-optimization between legacy Tx AMI models and new “optimizer” Rx AMI models
- **IBIS “cannot afford” to lose the support and interest of major companies in AMI modeling...**

High level summary of the proposals

- **BIRD 147 proposes the usage of a single .bci file for the back channel communication between Tx and Rx**
 - the parameter strings which are exchanged between the models are generated by the models, adhering to the rules in the .bci file
 - the EDA tool is responsible to take the parameter string from one model and pass it to the other model
 - the EDA tool is not expected (or allowed?) to make any modifications to these strings while passing them around
- **SiSoft proposes to use AMI parameters placed in the .ami file to facilitate Tx/Rx communication/optimization**
 - all optimization parameters are first read by the EDA tool from the .ami file and interpreted/processed according to the rules in the specification and passed to/from the DLLs as needed

What is the main difference?

- **Note that in both proposals the parameter strings are passed in/out of the AMI DLL by the EDA tool**
 - the DLL function signatures are not changed
 - BIRD 147 builds on BIRD 128 to allow AMI_parameters_out to be used for passing strings into the GetWave function
 - not stated (yet) but it seems that the SiSoft proposal will also need BIRD 128 or something equivalent
- **However, in BIRD 147, the strings are generated by the AMI DLLs based on the .bci parameters and the EDA tool only acts as a “mailman”**
- **In the SiSoft approach the strings are generated and processed by the EDA tool based on .ami parameters**
 - there may be a “mailman” mode in this proposal too

What do we need to keep Intel happy?

- **Intel likes the .bci file approach because the .bci file is allowed to have proprietary content**
 - another advantage is that both Tx and Rx uses the same .bci file, reducing the possibility for miscommunication which may arise when the Tx and Rx .ami files are incompatible
- **These are strong arguments for using the .bci files**
- **Could we achieve the same capabilities with the SiSoft approach using Model_Specific AMI parameters in the .ami file?**

What do we need to keep Altera happy?

- **Altera likes the SiSoft approach because it allows for co-optimization with legacy Tx AMI DLL-s without recompiling them**
 - additional (new) .ami file parameters are acceptable (and probably needed) to achieve this goal
- **Could we achieve the same capabilities with BIRD 147 if the EDA tool would be allowed to be “more involved”?**
 - let the EDA tool read/interpret .bci files for those DLLs which don't
 - e.g. an Rx DLL wouldn't know that it is not talking to a real Tx DLL
 - **this would only work with standard .bci files because the EDA tool would not be able to interpret proprietary .bci file content**
 - the .ami parameters which are needed in the SiSoft proposal to help the EDA tool to adjust the Tx DLL taps could also be used for this approach



Summary

- It seems that **BIRD 147** could be extended to support system level optimization by making provisions for the EDA tool to be “more involved”
- Not sure whether the SiSoft proposal can be extended in a similar manner to support proprietary protocols through **Model_Specific .ami** parameters
- I would recommend to look into these technical details and find a solution that supports the needs of both of our major semiconductor vendors
 - after all, we always complain that we don't get enough feedback from IC vendors
 - now we have feedback, we should act on them
- **This challenge doesn't seem to be unsolvable**