AMI Backchannel Response to Sigrity/Gennum Proposal

Walter Katz
SiSoft

IBIS-ATM Committee
May 10, 2011

SiSoft™
We Are Signal Integrity

SiSoft AMI Backchannel Proposal

- New AMI Reserved Parameter
  - Backchannel_Protocol
    - Tells User which Backchannel Protocols Supported by this Model
    - User tells model what protocol to use
    - Allows User to disable Backchannel
      - Model must support BIRD-120 Flows

SiSoft™
We Are Signal Integrity

Example .ami File

```
(Backchannel_Tx (Description "Generic backchannel transmitter model")
  (Reserved_Parameters
    (Ignore_Bits (Value 4) (Usage Info) (Type Integer)
        (Description "Ignore four bits to fill up tapped delay line."))
    (Init_Returns_Impulse (Value True) (Usage Info) (Type Boolean)
        (Description "Model supports statistical flow."))
    (GetWave_Exists (Value True) (Usage Info) (Type Boolean)
        (Description "Model supports time domain flow."))
    (Backchannel_Protocol List "None" "PCIeG3" "802.3KR")
        (Type String) (Usage InOut)(
        (Description "Model supports PCIeG3 and 802.3KR"))
  ) | End Reserved_Parameters
   (Model_Specific
|       Normal Model Specific Parameters
     (Backchannel (Description "Backchannel Protocols Branch")
        (802.3KR  (Description "802.3KR Protocol Branch")
|              802.3KR Specific Parameters
        )
        (PCIeG3  (Description "PCIeG3 Protocol Branch")
|              PCIeG3 Specific Parameters
        )
      )  | End Backchannel Branch
   )  | End Model Specific Parametere
)  | End Backchsnnel Tx
```


SiSoft™
We Are Signal Integrity

Model Specific Protocol Simulator Parameters

```
!            Note, Red items are  "Simulator_Commands"
(Backchannel (Description "Backchannel Protocols Branch")
   (802.3KR  (Description "802.3KR Protocol Branch")
     (Simulator_Commands
       (Training (List "Off" "On") (Usage InOut) (Type String)(Description
              "Model or simulator can turn Training On or Off.
               Simulator disables Training by setting it Off.
               Rx Getwave can stop Training by returning Off."))
       (Max_Train_Bits  (Range 500000 50000 1000000000)
            (Usage InOut) (Type Integer)
            (Description "Simulator should stop training after 500000 bits")
       (Training_Pattern (Description
              "On fist call to Tx GetWave when (Training On) input
               stimulus to Tx GetWave shall be Preamble followed by
               Data repeated until Rx GetWave sets (Training Off).
               When Rx GetWave sets (Training Off), the next input
               stimulus will be Postamble then simulation stimulus")
          (Preamble (Usage Info) (Type String)
             (Value "b11111111111111110000000000000000 1")
             (Description "Training Preamble sent once"))
          (Data (Usage Info) (Type String) (List "PRBS 11 b11110000111 2")
             (Description "Training Pattern is PRBS11, seed b11110000111
                           repeated twice(4092 bits)."))
          (Postamble (Usage Info) (Type String) (Value "b0011 5"
             (Description "Training Postamble sent five times"))))
```


SiSoft™
We Are Signal Integrity

Model Specific Protocol GetWave Parameters

```
(GetWave_Commands
   (BackChanControls (Description "Tap controls")
         (Max_Train_Bits  (Value 500000) (Usage In) (Type Integer)
           (Description "Model will stop training after 500000 bits")
         (Description "From Rx
                              -n Decrements tap by n
                               0 Tap is unchanged
                               n Increments tap by n
                         From Tx
                              -1 Low limit has been reached
                               0 Setting is adjustable
                               1 High limit has been reached")
       (-1 (Range 0 -1 1) (Usage InOut) (Type Integer)
           (Description "Pre-cursor tap control"))
       (0  (Range 0 -1 1) (Usage InOut) (Type Integer)
           (Description "Main tap control"))
       (1 (Range 0 -1 1) (Usage InOut) (Type Integer)
           (Description "First post-cursor tap control"))
    )   | End BackChanControls
  ) | End GetWave_Commands
) | End 802.3KR
(PCIeG3 (Value "PCIeG3.pmi") (Type String) (Usage Info)
    (Description "Include file PCIeG3.pmi here"))
) | End Backchannel
```


SiSoft™
We Are Signal Integrity

Defining Preamble, Data and Postamle Stimulus Patterns

- We should be able to easily define any of these waveforms as a concatenated list of waveform snippets.
- Example using Format Table:

```
(Data (Type String) (Usage Info)
   (Table
      ("b01111111111111111000000000000000 5")
      ("h0123456789ABCDEF0123456789ABCDEF 10")
      ("o01234567012345670123456701234567 10")
      ("PRBS 11 b11110000111 2"))
   (Description "
      Strings that begin b,h,o, denote Binary, Hex, Octal.
         These bit patterns are followed by a repeat count
         (default is 1, which means the pattern is added once).
      Strings that begin with PRBS generate a Pseudo Random Binary Sequence
      using a Linear Feedback Shift Register. PRBS is followed by 3 fields:
         PRBS <duty cycle> <seed> <repeat count>
           <duty cycle> A positive, integer number.
                 The PRBS patter will repeat every 2^<duty cycle> bits.
           <seed> A non-negative integer number (can be represented as b…)
           <repeat count> is non-negative integer number.
                 The number of times this bit pattern is to be inserted
                 into the stimulus. If Repeat count is 0, the pattern is to
                 be repeated until the Rx Model turns Training Off.")
```


SiSoft™
We Are Signal Integrity

Example Flow Tracing
AMI_parameters_in and AMI_parameters_out

```
Note: Using AMI_parameters_io to describe usage of AMI_parameters_out in
accordance with BIRD 128.

AMI_parameters_in from simulator to Rx AMI_Init
  (rx_root …)(PCIeG3(Simulator_Commands (Training "On")
                                        (Max_Train_Bits 500000)))
AMI_parameters_out from Rx AMI_Init
  (rx_root …)(PCIeG3(Simulator_Commands (Training "On"))
             (GetWave_Commands (BackChanControls (-1 0) (0 0) (1 0))))
AMI_parameters_io to Tx AMI_GetWave
  (PCIeG3(Simulator_Commands (Training "On"))
         (GetWave_Commands (BackChanControls (-1 0) (0 0) (1 0)))
AMI_parameters_io from Tx AMI_GetWave
  (tx_root …)( PCIeG3(GetWave_Commands(BackChanControls(-1 0)(0 0)(1 0))))
AMI_parameters_io to Rx AMI_GetWave
  ( PCIeG3(GetWave_Commands(BackChanControls(-1 0)(0 0)(1 0))))
AMI_parameters_io from Rx AMI_GetWave
   (rx_root …)(PCIeG3(Simulator_Commands (Training "Off"))
         (GetWave_Commands (BackChanControls (-1 -1) (0 +1) (1 -1)))
AMI_parameters_io to Tx AMI_GetWave
  (PCIeG3(Simulator_Commands (Training "off"))
         (GetWave_Commands (BackChanControls (-1 -1) (0 +1) (1 -1)))
```


SiSoft™
We Are Signal Integrity

Who Owns What?

- IBIS Owns
  - Backchannel_Protocol
  - (Backchannel (<protocol>
    - (Simulator_Commands
      - (Training
      - (Max_Train_Bits
      - (Training_Pattern
        - (Preamble
        - (Data
        - (Postamble
    - (GetWave_Commands (...))
- Protocol Owns
  - (Backchannel (<protocol> (GetWave_Commands (...))))

SiSoft™
We Are Signal Integrity

Who controls (<protocol>(GetWave_Commands (...))

- Protocol Registry
  - IBIS should maintain a registry of Protocols
  - Model Makers may submit Protocols to the Registry
  - IBIS may from time to time officially approve specific Protocols in consultation with other standards committees.
- Unregistered Protocols
  - A Model Maker can create a new Protocol, and develop Tx and Rx models to that Protocol
  - New Protocols do not need to be Registered

Summary

- No need for new communications file
- Single model supports multiple protocols
- Clear definition of what Model Maker needs to tell EDA Tool, under the control of IBIS
- Clear definition of what Tx and Rx models need to communicate
- Clear definition of how Protocols can be designed, registered, and formally approved.

SiSoft™
We Are Signal Integrity