IBIS-AMI Time-Domain Reference Flow

–James Zhou, QLogic Corp.

Aug. 29, 2011

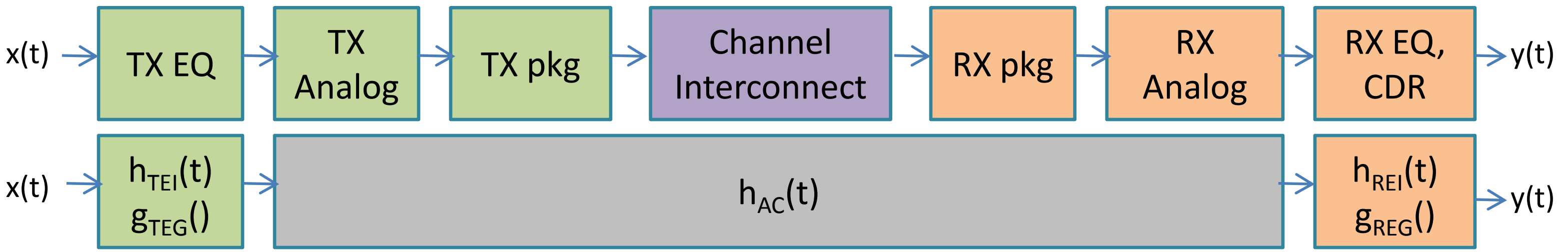
Background and Objectives

- The subject of IBIS-AMI cookbook was raised by Mike LaBonte, Chairman of IBIS Quality Task Group, during the Aug 9, 2011 teleconference.
- The AR was to create a “starter” presentation summarizing the current status of IBIS-AMI reference flows and modeling approaches for the purpose of exploring end-user interests and concerns on IBIS-AMI.
- With the end goal being to create materials for end-user education and training, the feedbacks and comments generated during this process may also help to identify issues in the Specification requiring clarification or modification.
- This presentation only covers IBIS-AMI time-domain reference flow.
- Thanks to T. Westerhoff and W. Katz of SiSoft and, Greg Edlund of IBM for insightful comments to the original version of the presentation.

AMI Reference Flow – Brief History

- BIRD 104.1, (10/2007)
  - First public proposal of IBIS-AMI
- BIRD 107.1, and IBIS Specification 5.0, (05/2008, 08/2008)
  - Introduced Use_Init_Output to solve the double counting issue when filtering exist in both AMI_Init and AMI_Getwave functions
  - Added dedicated section to describe reference flow
- BIRD 120.1 (04/2011)
  - Deprecated Use_Init_Output
  - Revised reference flow section to separate statistical and time-domain flows
  - Corrected inconsistencies in IBIS 5.0 flow for NLTV systems

Reference Flow – IBIS 5.0



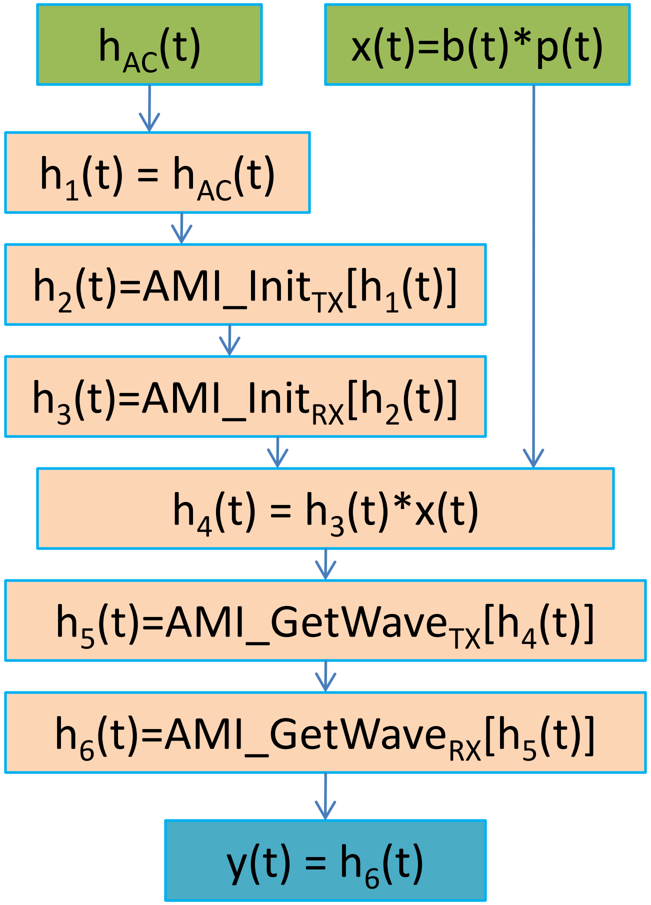
Step 1: $h_1(t) = h_{AC}(t)$	
Step 2a: $h_{2a}(t) = AMI\_Init_{TX}[h_1(t)] = h_{TEI}(t)*h_{AC}(t)$	(Tx Use_Init_Output = TRUE)
Step 2b: $h_{2b}(t) = h_1(t) = h_{AC}(t)$	(Tx Use_Init_Output = FALSE)
Step 3a: $h_{3a}(t) = AMI\_Init_{RX}[h_2(t)] = h_{REI}(t)*h_2(t)$	(Rx Use_Init_Output = TRUE)
Step 3b: $h_{3b}(t) = h_2(t)$	(Rx Use_Init_Output = FALSE)
Step 4: $h_4(t) = h_3(t) *b(t)*p(t)$	
Step 5a: $h_{5a}(t) = AMI\_GetWaveTX[h_4(t)]$	(Tx GetWave_Exists = TRUE)
Step 5b: $h_{5b}(t) = h_4(t)$	(Tx GetWave_Exists = FALSE)
Step 6a: $h_{6a}(t) = AMI\_GetWaveRX[h_5(t)]$	(Rx GetWave_Exists = TRUE)
Step 6b: $h_{6b}(t) = h_5(t)$	(Rx GetWave_Exists = FALSE)
Step 7: $y(t) = h_6(t)$	

Observations

- $h_{AC}(t)$ is the end-to-end analog channel impulse response
- b(t)*p(t) is the input waveform to Tx AMI block
- It is not obvious how to map this process to system equations relating output to input
- Init_Returns_Impulse indicates whether output AMI_Init is modified
- Naming conventions for impulse and AMI_GetWave functions in this presentation follow that of DAC 2009 IBIS Summit Presentation by W. Katz

Reference Flow Diagram – IBIS 5.0

- Output of AMI_Init convolves with stimulus to become the input to AMI_GetWave.
- Analog channel $h_{AC}(t)$ participates in both Init and GetWave calls.
- Init_Returns_Impulse and Tx/Rx GetWave_Exists only impact AMI_Init and AMI_GetWave functions locally



$h_{AC}(t)$

$x(t)=b(t)*p(t)$

$h_1(t) = h_{AC}(t)$

$h_2(t)=AMI\_Init_{TX}[h_1(t)]$

$h_3(t)=AMI\_Init_{RX}[h_2(t)]$

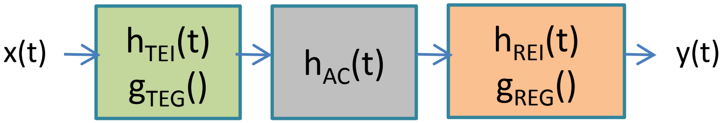
$h_4(t) = h_3(t)*x(t)$

$h_5(t)=AMI\_GetWave_{TX}[h_4(t)]$

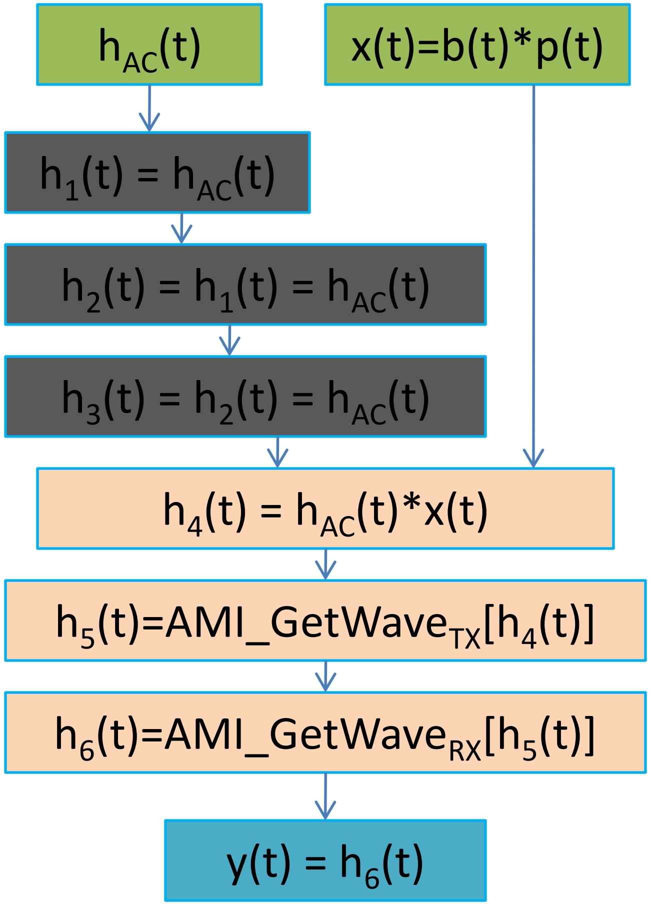
$h_6(t)=AMI\_GetWave_{RX}[h_5(t)]$

$y(t) = h_6(t)$

Reference Flow Diagram – IBIS 5.0



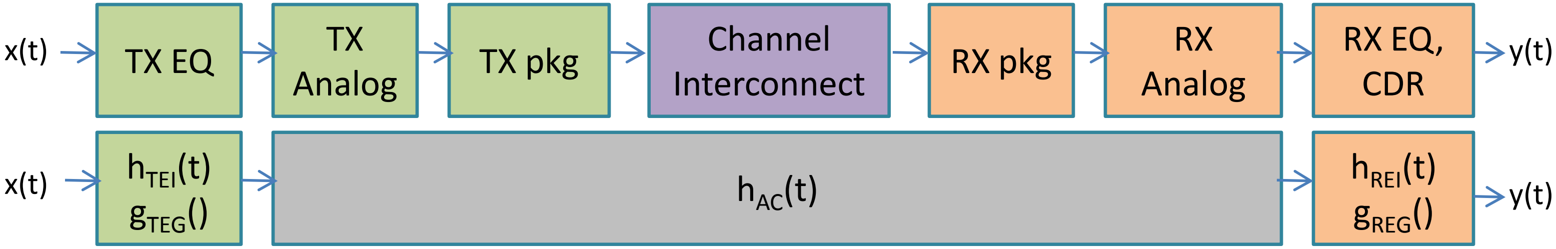
- If Use_Init_Output = FALSE, Init phase is pass-thru only
- Convolving x(t) directly with $h_{AC}(t)$ without including the Tx AMI block (Step 4) makes this flow invalid for NLTV Tx AMI block


$h_{AC}(t)$
$x(t)=b(t)*p(t)$
$h_1(t) = h_{AC}(t)$
$h_2(t) = h_1(t) = h_{AC}(t)$
$h_3(t) = h_2(t) = h_{AC}(t)$
$h_4(t) = h_{AC}(t)*x(t)$
$h_5(t)=AMI\_GetWave_{TX}[h_4(t)]$
$h_6(t)=AMI\_GetWave_{RX}[h_5(t)]$
$y(t) = h_6(t)$

Double-Counting

- Tx equalization may be double-counted if AMI_GetWave contains a different equalization than AMI_Init.
- Ambiguities exist in input and output variables of AMI_GetWave calls.
- Use_Init_Output was introduced to allow bypassing of AMI_Init function calls by directly convolving the analog channel with stimulus before calling AMI_GetWave functions.
- The reference flows become complicated when all combinations of Use_Init_Output, GetWave_Exists must be dealt with in a consistent manner.

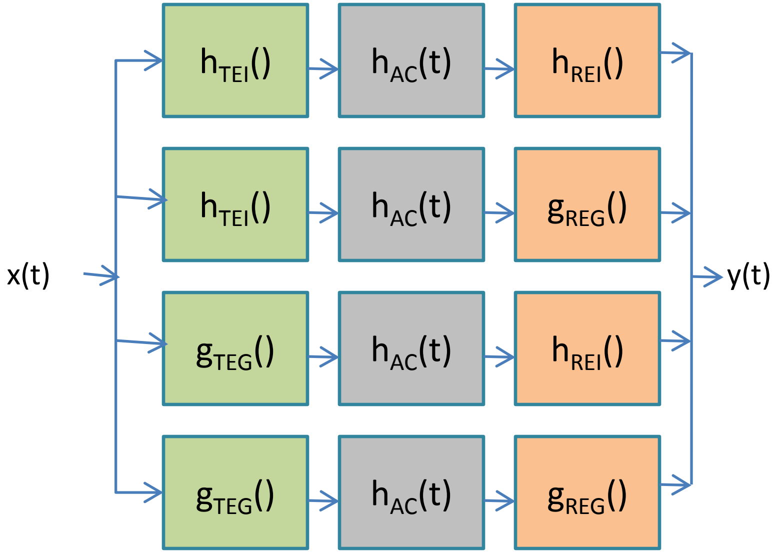
Reference Flow - BIRD 120.1



Step 1: $h_1(t) = h_{AC}(t)$
Step 2: $h_2(t) = Tx\_AMI\_Init[h_1(t)] = h_{TEI}(t)*h_{AC}(t)$
Step 3: $h_3(t) = Rx\_AMI\_Init[h_2(t)] = h_{REI}(t)*h_{TEI}(t)*h_{AC}(t)$
Step 4: $h_4(t) = x(t) = b(t)*p(t)$
Step 5: $h_5(t) = g_{TEG}[h_4(t)];$ (TxGE = TRUE)
Step 6a: $h_{6a}(t) = g_{REG}[h_1(t)*h_5(t)];$ (TxGE=TRUE;RxGE=TRUE)
Step 6b: $h_{6b}(t) = g_{REG}[h_2(t)*h_5(t)];$ (TxGE=FALSE;RxGE=TRUE)
Step 6c: $h_{6c}(t) = h_3(t)*h_4(t);$ (TxGE=FALSE;RxGE=FALSE)
Step 6d: $h_{6d}(t) = h_{REI}(t)*h_1(t)*h_5(t);$ (TxGE=TRUE;RxGE=FALSE)
Step 7: $h_{7a,b}(t) = g_{REG}[h_{6a,b}(t)];$
Step 8: $h_8(t) = \{h_{7a}(t), h_{7b}(t), h_{6c}(t), h_{6d}(t)\}$

- [Note]: TxGE is TX GetWave_Exists; RxGE is RX GetWave_Exists

Block Diagram and Equations



FF: $y(t) = h_{REI}(t)*h_{AC}(t)*h_{TEI}(t)*x(t)$

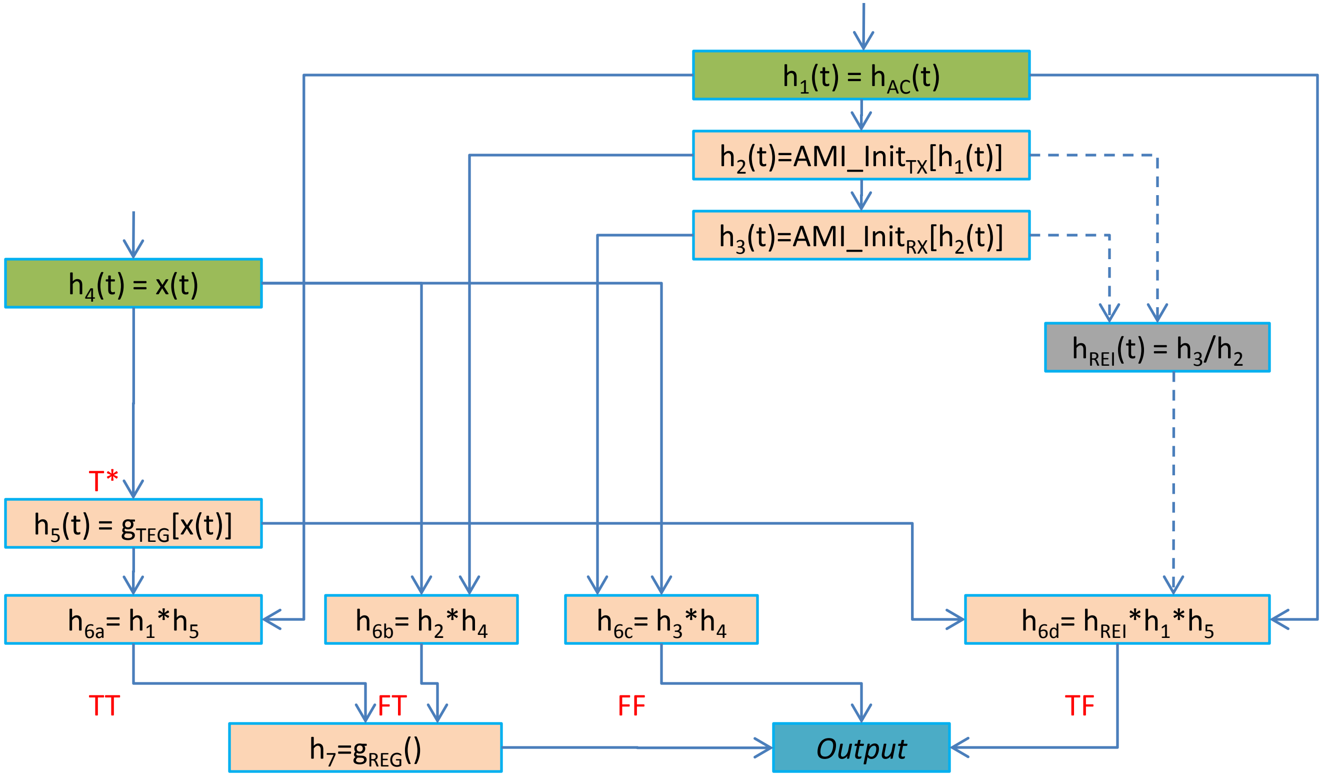
FT: $y(t) = g_{REG}[h_{AC}(t)*h_{TEI}(t)*x(t)\ ]$

TF: $y(t) = h_{REI}(t)*h_{AC}(t)*g_{TEG}[x(t)]$

TT: $y(t) = g_{REG}[h_{AC}(t)*g_{TEG}[x(t)]]$

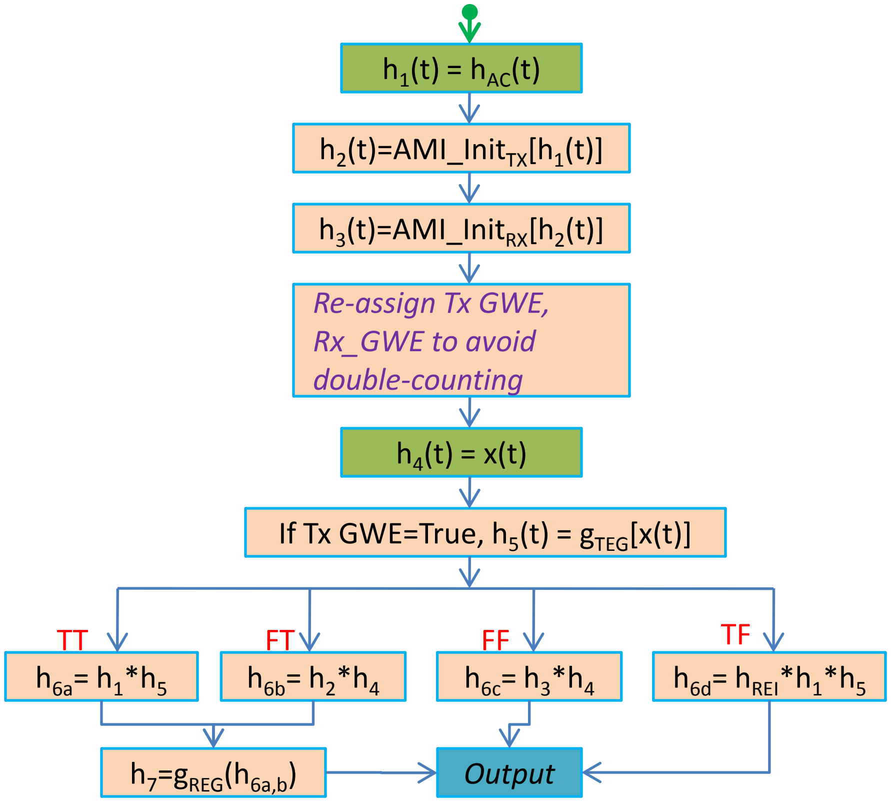
- Four possible cases of Tx and Rx AMI system with analog channel in between
  - [Tx GetWave_Exists, Rx GetWave_Exists] = {FF,FT,TF,TT}

Diagram of Equations



- Four possible combinations of Tx GetWave_Exists and Rx GetWave_Exists are: FF,FT,TF and TT

Reference Flow – Original View



$h_1(t) = h_{AC}(t)$

$h_2(t)$=AMI_Init$_{TX}[h_1(t)]$

$h_3(t)$=AMI_Init$_{RX}[h_2(t)]$

Re-assign Tx GWE, Rx_GWE to avoid double-counting

$h_4(t) = x(t)$

If Tx GWE=True, $h_5(t) = g_{TEG}[x(t)]$

TT

$h_{6a}= h_1*h_5$

FT

$h_{6b}= h_2*h_4$

FF

$h_{6c}= h_3*h_4$

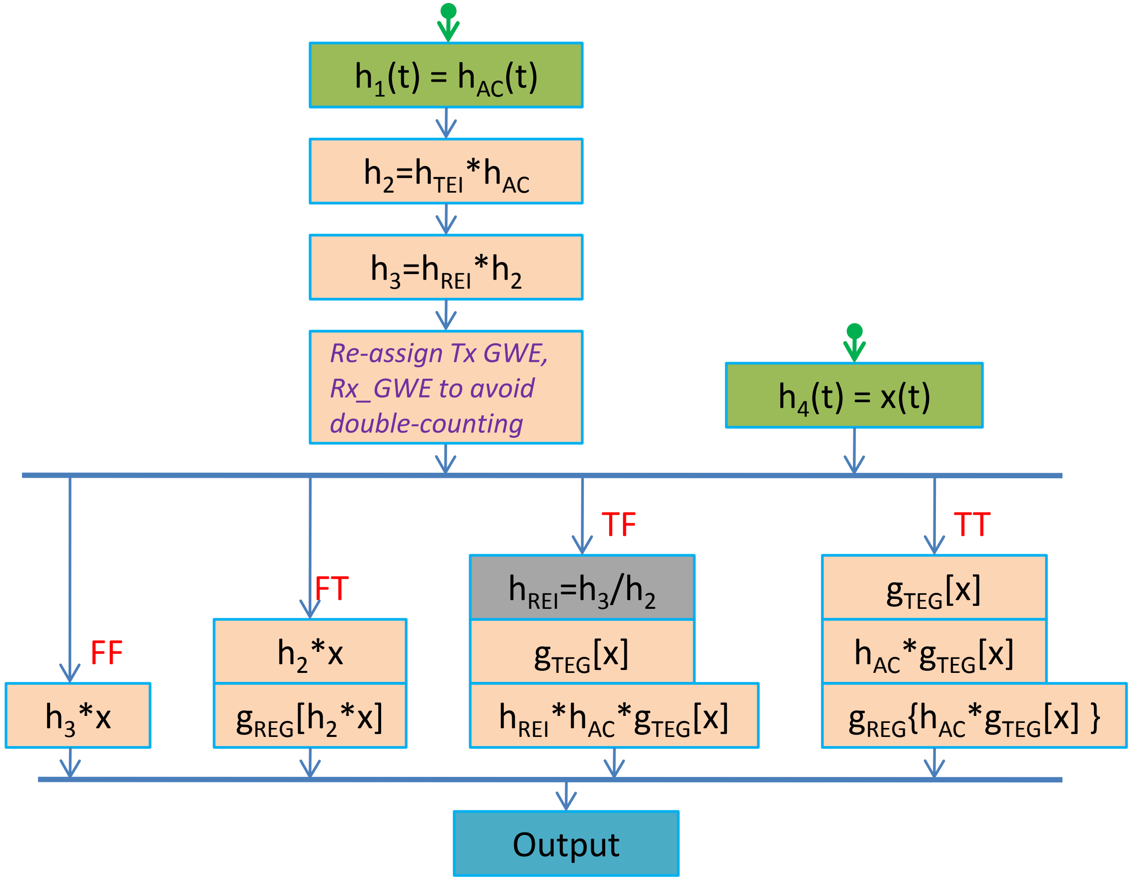
TF

$h_{6d}= h_{REI}*h_1*h_5$

$h_7=g_{REG}(h_{6a,b})$

Output

Reference Flow – Alternative View



- This is equivalent to reference flow on previous page

Observations

- Flow can be mapped to system equations from input to output for each block
- There are four branches in reference flow based on combinations of Tx and Rx GetWave_Exists
- Use_Init_Output was deprecated in BIRD120.1
- The same reference flow applies to both LTI and NLTV AMI blocks.
- GetWave_Exists may be re-assigned at simulation time to avoid double-counting

AMI_Init

- If Init_Returns_Impulse = TRUE, AMI_Init returns the convolution of input impulse response with impulse response of the equalization
- If Init_Returns_Impulse = FALSE, AMI_Init passes the input to output without changing it
  - the AMI block represents an all pass filter which impulse response is the Dirac delta function with unit amplitude.
- The output can always be interpreted as the convolution of the input with the impulse responses of the AMI block.

AMI_GetWave

- Only applies to time-domain flow; does not apply to statistical flow
- Can represent either NLTV or LTI AMI blocks
- Explicit relationship between output and input may not exist

Conclusion

- Deprecation of Use_Init_Output simplified the reference flow without comprising functionality
- Init_Returns_Impulse is information only
- Output can be related to input at each block
- Default reference flow can be changed at simulation time to avoid double-counting by reassigning Tx and Rx GetWave_Exists values