AMI Simulation Flow Round 3

Fangyi Rao

Motivations

- Handle Init-only Rx properly in both time domain and statistical flows for normal and redriver channels
- Provide full redriver channel impulse to Rx Init for optimization
- Eliminate the need for deconvolution

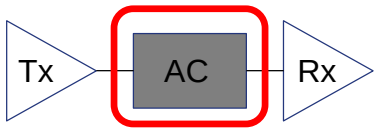
Summary

- No change to Tx Init
- Augment Rx Init impulse matrix by two columns for total impulse and Rx DFE

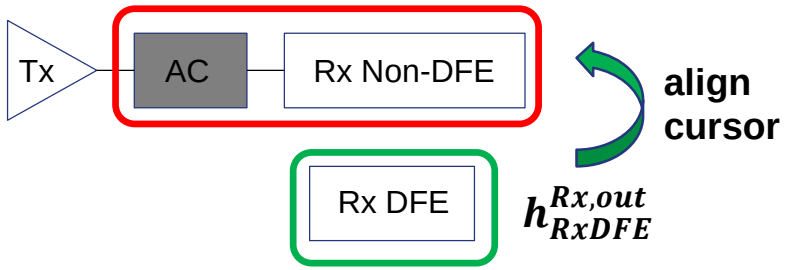
Rx Init Input Impulse		Rx Init Output Impulse	
Symbol	Definition	Symbol	Definition
$h_{partial}^{Rx,in}$	Impulse from upstream Tx input or output, depending on whether Tx has GetWave and whether simulation is in time domain or statistical, to Rx input	$h_{partial,RxNonDFE}^{Rx,out}$	Combined impulse of $h_{partial}^{Rx,in}$ and Rx's non-DFE portion (including gain and linear EQ)
$h_{total}^{Rx,in}$	Impulse from terminal Tx input to Rx input. Rx Init performs optimization based on this impulse	$h_{total,RxAll}^{Rx,out}$	Combined impulse of $h_{total}^{Rx,in}$ and the entire Rx (including gain , linear EQ and DFE)
$h_{RxDFE}^{Rx,in}$	Empty place holder for Rx Init to return DFE impulse	$h_{RxDFE}^{Rx,out}$	Rx DFE. Aligned cursors to $h_{partial,RxNonDFE}^{Rx,out}$
$h_{xtlk}^{Rx,in}$	Impulses from aggressors to Rx input	$h_{xtlk}^{Rx,out}$	Combined impulse of $h_{xtlk}^{Rx,in}$ and Rx's non-DFE portion (including gain and linear EQ)

Normal Time Domain Flow: GetWave Tx

$$h_{partial}^{Rx,in} = h_{AC}$$

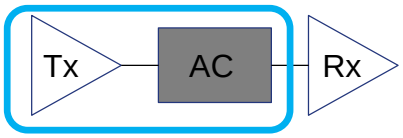


$$h_{partial,RxNonDFE}^{Rx,out} = h_{partial}^{Rx,in} * h_{RxNonDFE}$$

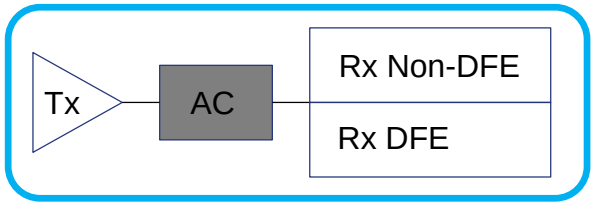


Rx Init

$$h_{total}^{Rx,in} = \textbf{Tx Init output}$$



$$h_{total,RxAll}^{Rx,out}$$



- GetWave Rx

  Rx output = Tx GetWave output $*$ h_{AC} → Rx GetWave

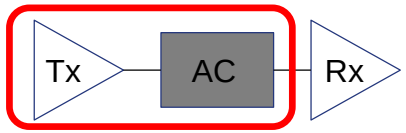
- Init-only Rx

  Rx output = Tx GetWave output $*$ $h_{partial,RxNonDFE}^{Rx,out}$ + Tx digital input $*$ $h_{RxDFE}^{Rx,out}$

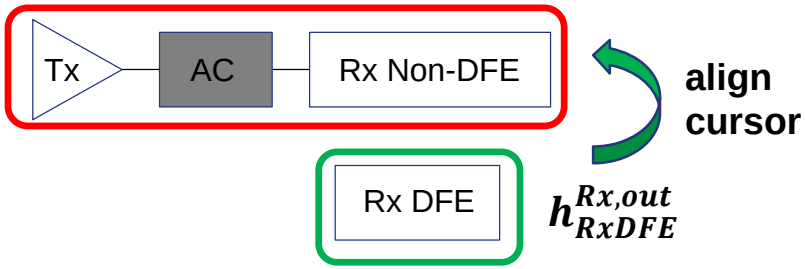
(note: EDA tool must align Tx digital input and Tx GetWave output)

Normal Time Domain Flow: Init-only Tx

$$h_{partial}^{Rx,in} = Tx\ Init\ output$$

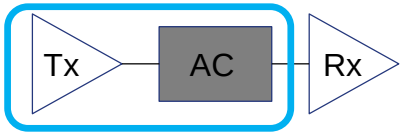


$$h_{partial,RxNonDFE}^{Rx,out} = h_{partial}^{Rx,in} * h_{RxNonDFE}$$

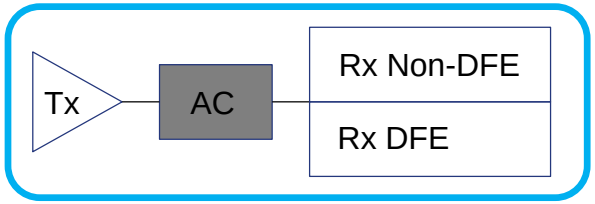


Rx Init

$$h_{total}^{Rx,in} = Tx\ Init\ output$$



$$h_{total,RxAll}^{Rx,out}$$



- GetWave Rx

  Rx output = Tx digital input $*$ $h_{partial}^{Rx,in}$ → Rx GetWave

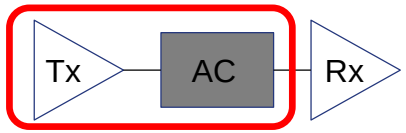
- Init-only Rx

  Rx output = Tx digital input $*$ $h_{total,RxAll}^{Rx,out}$

Note: $h_{total,RxAll}^{Rx,out} = h_{total,RxNonDFE}^{Rx,out} + h_{RxFE}^{Rx,out}$

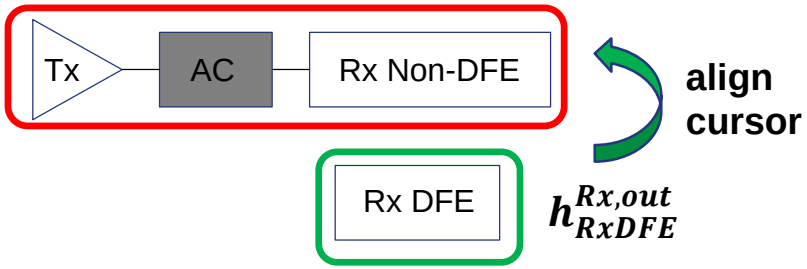
Normal Statistical Flow

$$h_{partial}^{Rx,in} = Tx\ Init\ output$$

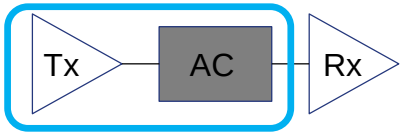


Rx Init

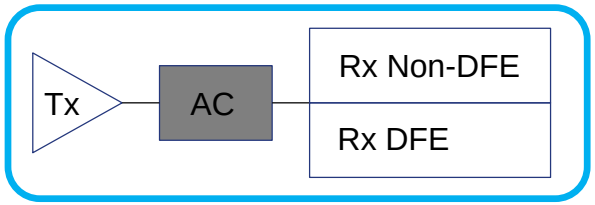
$$h_{partial,RxNonDFE}^{Rx,out} = h_{partial}^{Rx,in} * h_{RxNonDFE}$$



$$h_{total}^{Rx,in} = Tx\ Init\ output$$



$$h_{total,RxAll}^{Rx,out}$$

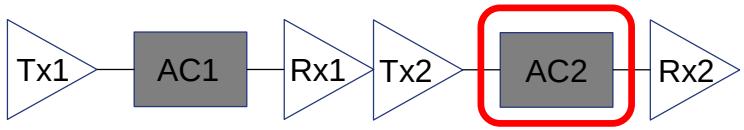


- Statistical uses $h_{total,RxAll}^{Rx,out}$

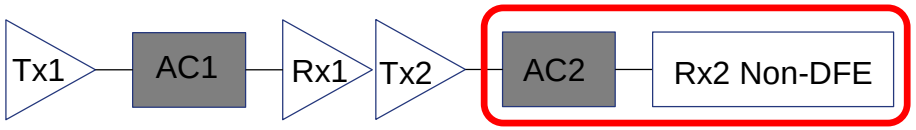
Note: $h_{total,RxAll}^{Rx,out} = h_{total,RxNonDFE}^{Rx,out} + h_{RxFE}^{Rx,out}$

Redriver Time Domain Flow: GetWave Tx2

$$h^{Rx,in}_{partial} = h_{AC2}$$



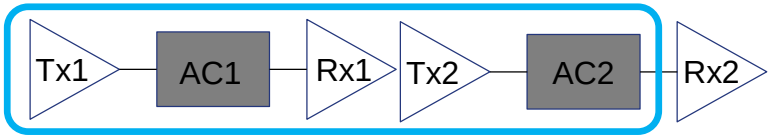
$$h^{Rx,out}_{partial,RxNonDFE} = h^{Rx,in}_{partial} * h_{Rx2NonDFE}$$



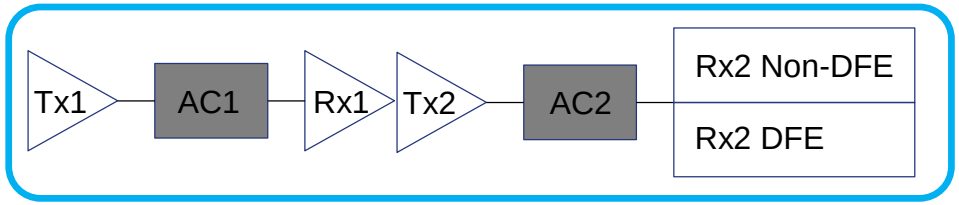
$h^{Rx,out}_{RxDFE}$

Rx2 Init

$$h^{Rx,in}_{total} = Rx1\ Init\ output * Tx2\ Init\ output$$



$$h^{Rx,out}_{total,RxAll}$$



- GetWave Rx2

  Rx2 output = Tx2 GetWave output $* h_{AC2}$ → Rx2 GetWave

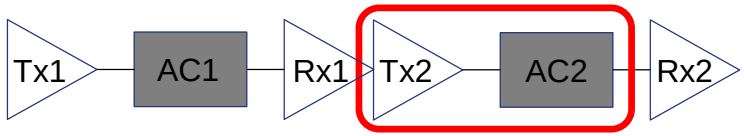
- Init-only Rx2

  Rx2 output = Tx2 GetWave output $* h^{Rx,out}_{partial,RxNonDFE}$ + Tx1 digital input $* h^{Rx,out}_{RxDFE}$

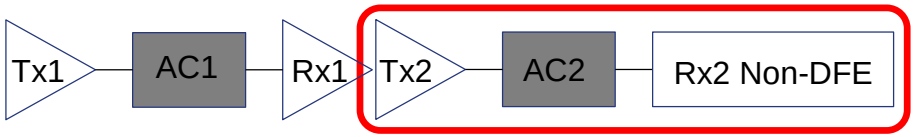
(note: EDA tool must align Tx1 digital input and Tx2 GetWave output)

Redriver Time Domain Flow: Init-only Tx2

$$h^{Rx,in}_{partial} = Tx2\ Init\ output$$



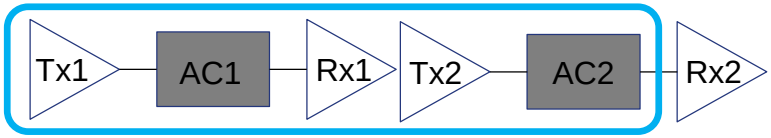
$$h^{Rx,out}_{partial,RxNonDFE} = h^{Rx,in}_{partial} * h_{Rx2NonDFE}$$



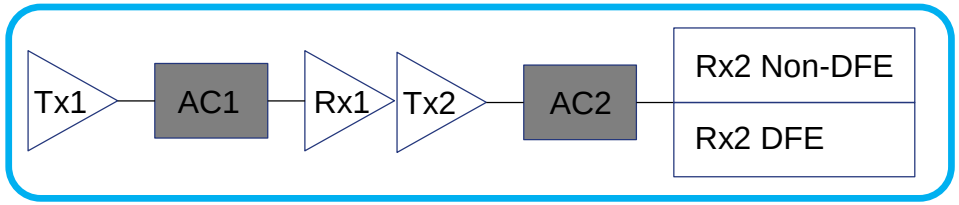
$h^{Rx,out}_{RxDFE}$

Rx2 Init

$$h^{Rx,in}_{total} = Rx1\ Init\ output * Tx2\ Init\ output$$



$h^{Rx,out}_{total,RxAll}$



- Getwave Rx2

  Rx2 output = Rx1 output $* h^{Rx,in}_{partial}$ → Rx2 GetWave

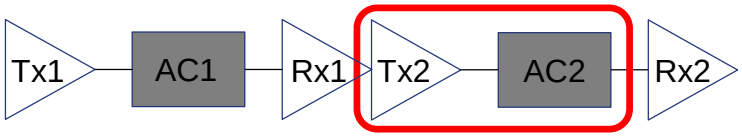
- Init-only Rx2

  Rx2 output = Rx1 output $* h^{Rx,out}_{partial,RxNonDFE}$ + Tx1 digital input $* h^{Rx,out}_{RxDFE}$

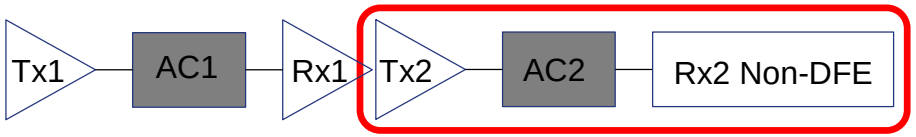
(note: EDA tool must align Tx1 digital input and Rx1 output)

Redriver Statistical Flow

$$h^{Rx,in}_{partial} = Tx2\ Init\ output$$



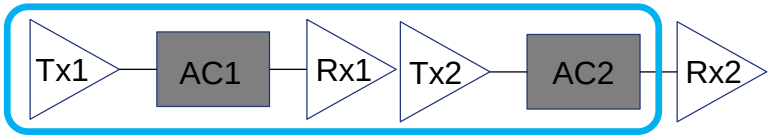
$$h^{Rx,out}_{partial,RxNonDFE} = h^{Rx,in}_{partial} * h_{Rx2NonDFE}$$



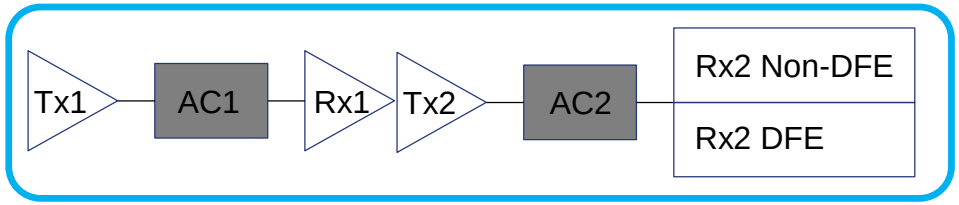
$h^{Rx,out}_{RxDFE}$

Rx2 Init

$$h^{Rx,in}_{total} = Rx1\ Init\ output * Tx2\ Init\ output$$



$h^{Rx,out}_{total,RxAll}$



- Statistical uses $h^{Rx,out}_{total,RxAll}$ for victim and $h^{Rx,out}_{partial,RxNonDFE}$ for aggressors received by Rx1

Summary

- No change to Tx Init
- Augment Rx Init impulse matrix by two columns for total impulse and Rx DFE

Rx Init Input Impulse		Rx Init Output Impulse	
Symbol	Definition	Symbol	Definition
$h_{partial}^{Rx,in}$	Impulse from upstream Tx input or output, depending on whether Tx has GetWave and whether simulation is in time domain or statistical, to Rx input	$h_{partial,RxNonDFE}^{Rx,out}$	Combined impulse of $h_{partial}^{Rx,in}$ and Rx's non-DFE portion (including gain and linear EQ)
$h_{total}^{Rx,in}$	Impulse from terminal Tx input to Rx input. Rx Init performs optimization based on this impulse	$h_{total,RxAll}^{Rx,out}$	Combined impulse of $h_{total}^{Rx,in}$ and the entire Rx (including gain , linear EQ and DFE)
$h_{RxDFE}^{Rx,in}$	Empty place holder for Rx Init to return DFE impulse	$h_{RxDFE}^{Rx,out}$	Rx DFE. Aligned cursors to $h_{partial,RxNonDFE}^{Rx,out}$
$h_{xtlk}^{Rx,in}$	Impulses from aggressors to Rx input	$h_{xtlk}^{Rx,out}$	Combined impulse of $h_{xtlk}^{Rx,in}$ and Rx's non-DFE portion (including gain and linear EQ)

New Reserved Parameters

Simulator_Supports_Augmented_Rx_Init_Impulse_Matrix

- Boolean, In, Optional, Default=False
- Set by simulator in the *AMI_parameters_in* input string of Rx AMI_Init to inform Rx model whether simulator supports the augmented impulse matrix

Rx_Init_Supports_Augmented_Impulse_Matrix

- Boolean, Info, Optional, Default=False
- Rx parameter that informs simulator whether Rx AMI_Init supports the augmented impulse matrix

New Reserved Parameters (cont'd)

Rx_Init_Supports_Augmented_Impulse_Matrix = False

- Simulator doesn't include *Simulator_Supports_Augmented_Rx_Init_Impulse_Matrix* in the *AMI_parameters_in* input string of Rx AMI_Init
- Simulator sends unaugmented impulse matrix to Rx AMI_Init
- Rx AMI_Init modifies the unaugmented impulse matrix
- Simulator proceeds according to the existing flow

Rx_Init_Supports_Augmented_Impulse_Matrix = True &

Simulator supports the new flow

- Simulator sets *Simulator_Supports_Augmented_Rx_Init_Impule_Matrix* to *True* in the *AMI_parameters_in* input string of Rx AMI_Init
- Simulator sends augmented impulse matrix to Rx AMI_Init
- Rx AMI_Init modifies the augmented impulse matrix
- Simulator proceeds according to the new flow

New Reserved Parameters (cont'd)

Rx_Init_Supports_Augmented_Impulse_Matrix = True &

Simulator doesn't support the new flow

- Simulator doesn't include *Simulator_Supports_Augmented_Rx_Init_Impule_Matrix* in the *AMI_parameters_in* input string of Rx AMI_Init
- Simulator sends unaugmented impulse matrix to Rx AMI_Init
- If Rx supports the existing flow, its AMI_Init modifies the unaugmented impulse matrix. Simulator proceeds according to the existing flow.
- If Rx doesn't support the existing flow, its AMI_Init errors out