IBIS 5.0 does not support Non-LTI transmitter models! Solution: Input to Tx GetWave should be a digital stimulus waveform.

Walter Katz

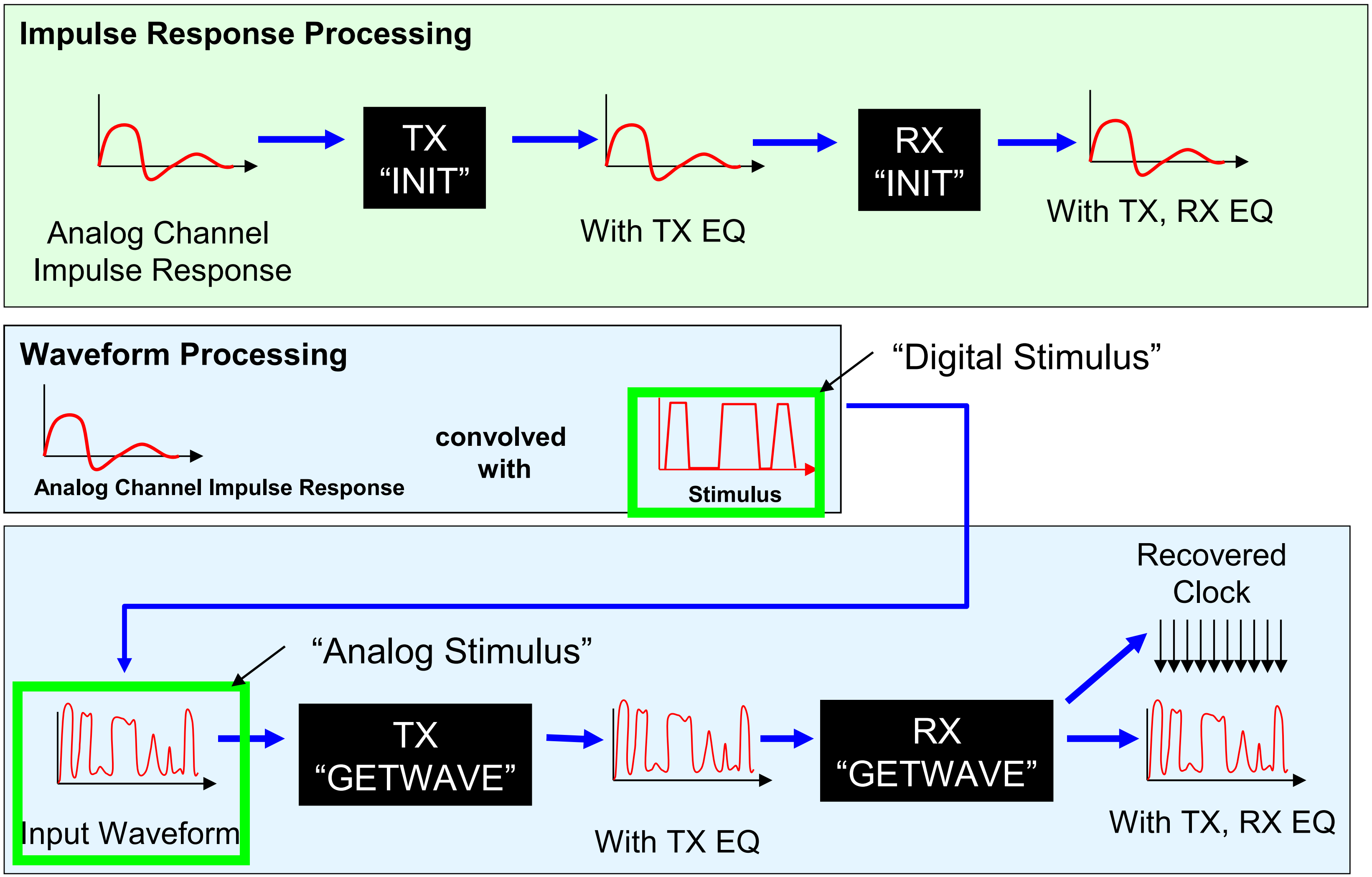
IBIS AMI

August 18, 2009

SiSoft™

Signal Integrity Software, Inc.

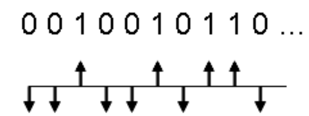
IBIS-AMI Algorithmic Models



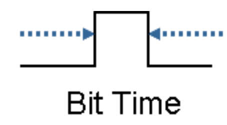
IBIS_Summit_2009_DAC_IBIS_AMI_Terminology.pdf

IBIS-AMI Terminology

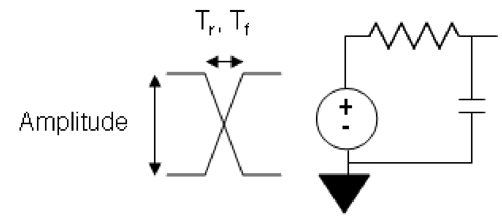
- Bit stream b(t)
  - Sum of delta functions



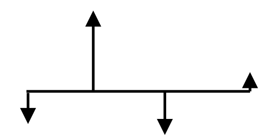
- Data symbol p(t)
  - Single bit width pulse



- TX analog characteristic $h_{TX}(t)$ *
  - Rise/fall time
  - Voltage swing
  - Drive impedance
  - Capacitance



- TX "Init" equalization $h_{TEI}(t)$ *
  - Sum of weighted delta functions
    - Coefficients & delays



- TX "Getwave" equalization $g_{TEG}()$ *
  - Not considered LTI
  - Waveform in, waveform out

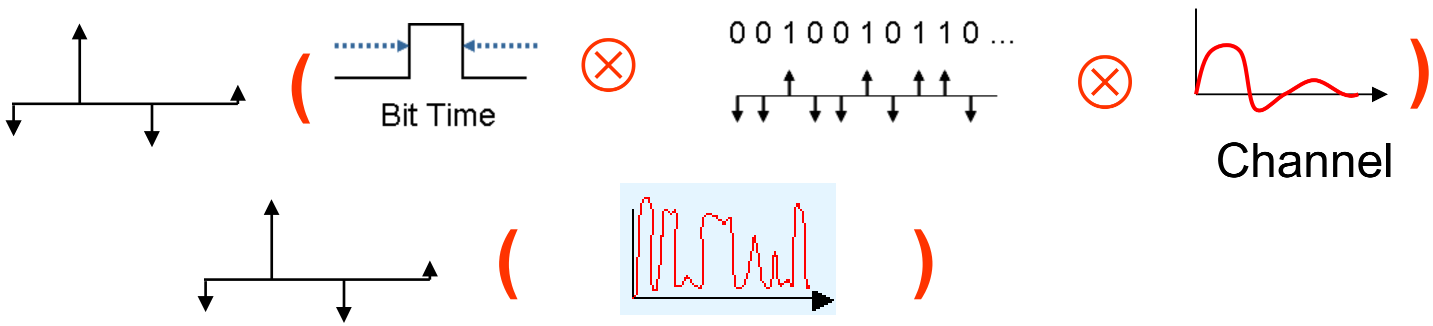
* Similar terms exist for RX

TX Use_Init_Output = False , RX Use_Init_Output = False

- Impulse response input to TX AMI_Init
  - $h_{AC}(t)$
- Impulse response input to RX AMI_Init
  - $h_{AC}(t)$
- Impulse response after RX AMI_Init
  - $h_{AC}(t)$
- Waveform input to TX Silicon (“Digital Stimulus”)
  - $p(t) \otimes b(t)$
- Waveform input to TX AMI_Getwave (“Analog Stimulus”)
  - $p(t) \otimes b(t) \otimes h_{AC}(t)$
  - Should be $p(t) \otimes b(t)$
- Waveform input to RX AMI_Getwave
  - $\mathbf{g_{TEG}(}\ p(t) \otimes b(t) \otimes h_{AC}(t)\ \mathbf{)}$
  - Should be $\mathbf{g_{TEG}(}\ p(t) \otimes b(t)\ \mathbf{)} \otimes h_{AC}(t)$
- Waveform output from RX AMI_Getwave
  - $g_{REG}(\ g_{TEG}(\ p(t) \otimes b(t) \otimes h_{AC}(t)\ )\ )$
  - Should be $g_{REG}(\ \mathbf{g_{TEG}(}\ p(t) \otimes b(t)\ \mathbf{)} \otimes h_{AC}(t))$

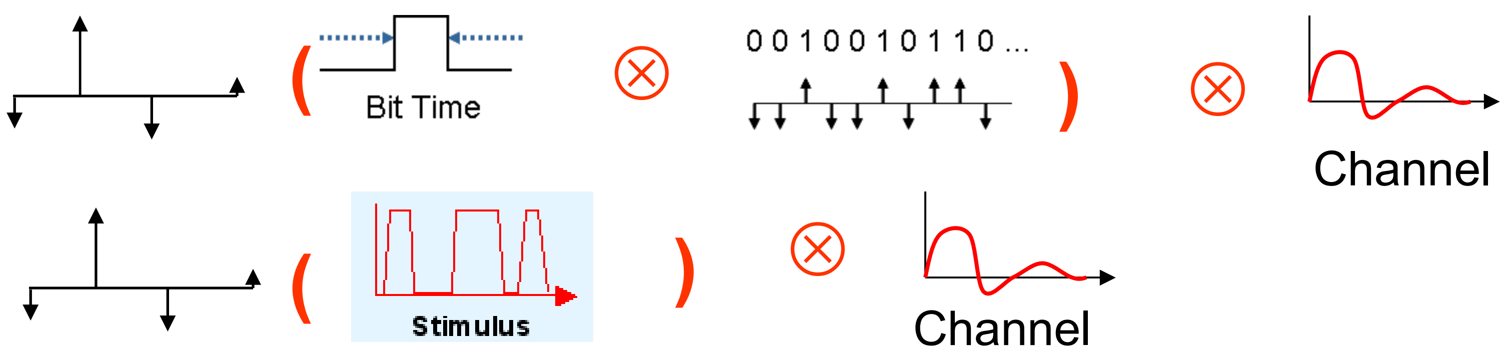
If not LTI these give different result, Proposed is correct.

g_{TEG} ($p(t) \otimes b(t) \otimes h_{AC}(t)$) **IBIS 5.0**



---

g_{TEG} $(p(t) \otimes b(t)) \otimes h_{AC}(t)$ **Proposed**



Examples of Non-LTI Tx Buffers

- **DCD**
  - **odd and even pulse widths different**
- **n-tap table lookup buffer**
  - **Amplitude of a symbol is determined from table lookup based on value of cursor (current) bit, and next (n-1) post cursor bits.**

IBIS 5.0 Tx GetWave Flow Broken Example: 3-tap table lookup

- Tx contains a 3-tap table lookup filter
- Tx GetWave Exists
- Tx Init does not modify the Impulse Response
- Rx Init does not modify the Impulse Response
- Long, lossy channel

3-tap table lookup

Table

- 000 4
- 100 3
- 010 2
- 110 0
- 001 9
- 101 7
- 011 6
- 111 5

Challenge, take the below "Stimulus", convolve with a lossy channel, and try to implement the equalization described in adjacent table so that one would get the same answer as one would get by convolving the channel with the Proposed Tx Equalized waveform.

Stimulus 0001011100010111000101110001011100…
Tx Equalized 4927650349276503492765034927650 3…

SiSoft™

5.0 Tx GetWave Flow Broken Example: Tx DCD, lossless channel

- Tx is a two tap FIR filter
  - High levels: 3,2
  - Low levels: 0,1
- Tx GetWave Exists
- Tx Init does not modify the Impulse Response
- Rx Init does not modify the Impulse Response
- Four time steps per bit
- Tx and Rx are both 100 Ohm differential impedance
- Short, low loss channel, 100 Ohm differential impedance

Stimulus patterns A and B (identical but shifted by 1 bit):

```
A 1100110011001100110011001100110011001100110011001100
B 0110011001100110011001100110011001100110011001100110
```

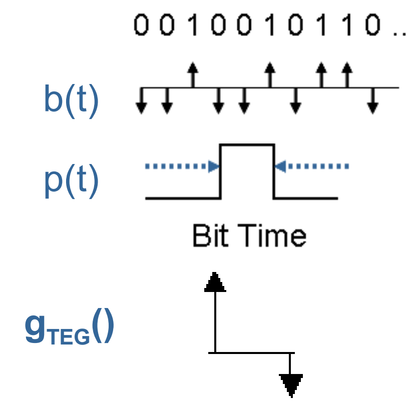
Tx Equalization Primer
Simple Step Function

```
Step function (Rising edge)
0   0   0   0   0   1   1   1   1   1   1   1
000000000000000000001111111111111111111111111111
111111111111111111113333222222222222222222222222

                    3333
                        222222222222222222222222
11111111111111111111



Step function (Falling edge)
1   1   1   1   1   0   0   0   0   0   0   0
111111111111111111110000000000000000000000000000
222222222222222222220000111111111111111111111111

22222222222222222222
                        111111111111111111111111
                    0000
```

b(t)

p(t) ⊗ b(t)

g_{TEG}(p(t) ⊗ b(t))



How Tx works when no Tx DCD

```
A   (Red are odd bits)
1   1   0   0   1   1   0   0   1   1   0   0   1   1   0   0
1111111100000000111111110000000011111111000000001111111100000000
3333222200001111333322220000111133332222000011113333222200001111

3333            3333            3333            3333
    2222            2222            2222            2222
            1111            1111            1111            1111
        0000            0000            0000            0000

B   (Red are odd bits)
0   1   1   0   0   1   1   0   0   1   1   0   0   1   1   0
0000111111110000000011111111000000001111111100000000111111110000
1111333322220000111133332222000011113333222200001111333322220000

    3333            3333            3333            3333
        2222            2222            2222            2222
1111            1111            1111            1111
            0000            0000            0000            0000
```

How Tx works with 75% DCD

```
A    (Red are odd bits)
1      10      01      10      01      10      01      10      0
1111111100000000111111110000000011111111000000001111111100000000
3333333200000001333333320000000133333332000000013333333200000001

3333333         3333333         3333333         3333333
       2               2               2               2
               1               1               1               1
        0000000         0000000         0000000         0000000

B     (Red are odd bits)
0      11      00      11      00      11      00      11      0
0000000111111110000000011111111000000001111111100000000111111110
1111111322222220111111132222222011111113222222201111111322222220

       3               3               3               3
        2222222         2222222         2222222         2222222
1111111         1111111         1111111         1111111
               0               0               0               0
```

A and B with 75% DCD give dramatically different results!

<u>This proves that DCD is not LTI!</u>

This is clearly a real effect in silicon that must be simulatable by AMI models!

Input to Tx GetWave according to IBIS 5.0 is the same for all four cases (except time shift)

```
A   0% DCD 1111111100000000111111110000000011111111000000001111111100000000
B   0% DCD 0000111111110000000011111111000000001111111100000000111111110000
A 75% DCD 1111111100000000111111110000000011111111000000001111111100000000
B 75% DCD 0000000111111110000000011111111000000001111111100000000111111110
```

Another affect is DCD amplification. Demonstrating this this would require a lossy channel, and comparison of real simulation results.

Need to make input to Tx GetWave a digital stimulus pattern.

Input to Tx GetWave should be stimulus (e.g. 11001100...) but in the form a waveform that is .5 (or >0.) when the signal is high and -.5 (or <0.) when the signal is low. Transitions occur when the waveform crosses 0.

And Very Importantly

IC Vendors currently use digital stimulus pattern input to the Tx models that they have written for MatLab and other proprietary SerDes analysis tools.

End of presentation on digital stimulus input to Tx GetWave

