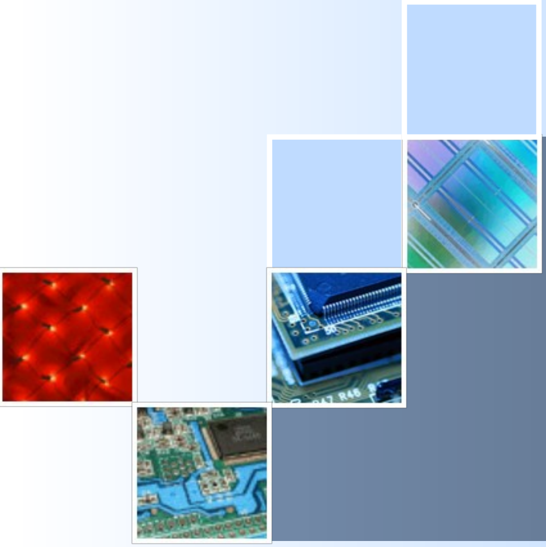


IBIS ATM: txgetwave

Kumar Keshavan

Aug 25, 2009

SIGRITY

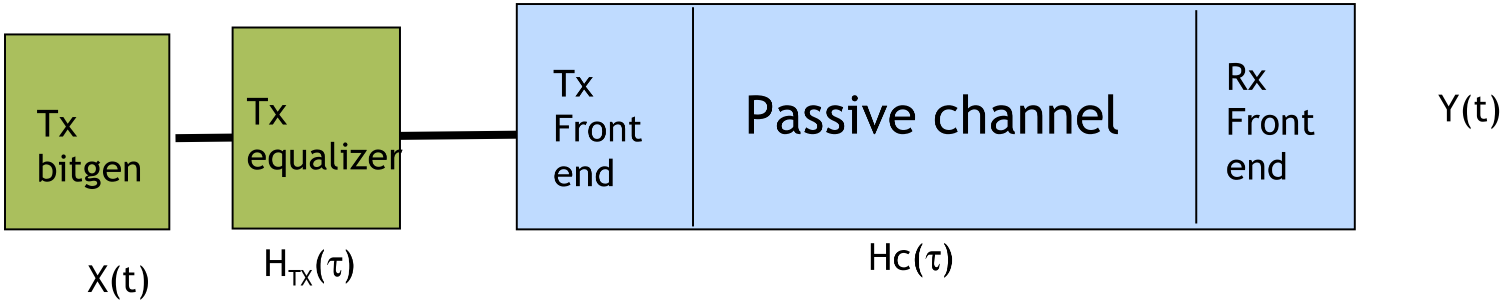
Objective

- Illustrate errors in last week txgetwave discussion
- Clarify where and when the ‘new’ txgetwave is needed

System schematic and terminology

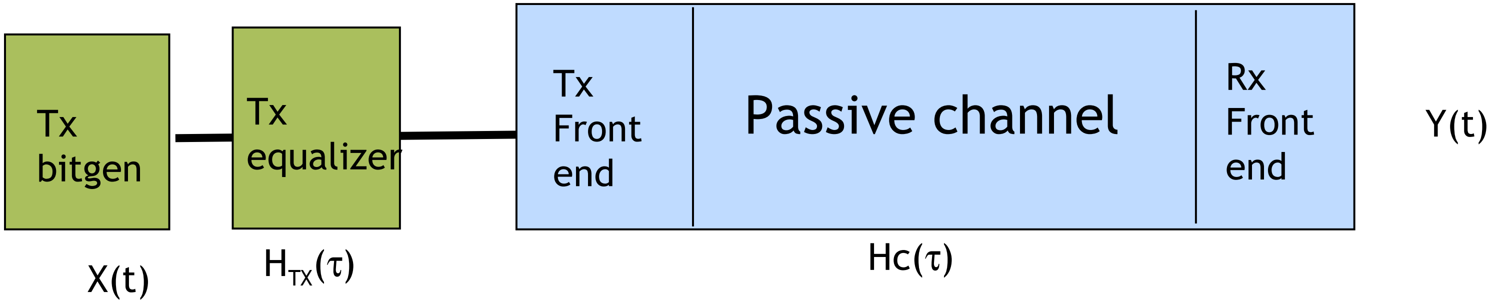
- Analog front end
- Tx input
  - Bit generateor
  - Equalizer
- Rx input

Wave form generation: case 1



- The Tx bit generator outputs a continuous ‘bit’ wave form.(x(t)) X(t) can include bit distortion including dcd and any other arbitrary jitter and noise
- The Tx equalizer; $H_{tx}(\tau)$; $H_{tx}(\tau)$ is the equalizer transfer function. In case1 it is independent of x(t)

Wave form generation: case 1



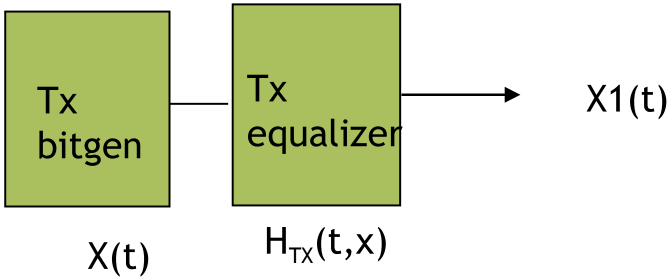
- If $H_{Tx}(\tau)$ and $Hc(\tau)$ is LTI(i.e they have time independent characteristic transfer functions) , then we can write the output wave as
  - $Y(t) = X(t) * Htx(\tau) * Hc(\tau) = X(t) * Htxc(\tau)$
  - where $Htxc(\tau) = Htx(\tau) * Hc(\tau)$ = combined transfer function of tx equalizer and the channel

Case 1 conclusion bit distortion like dcd does not make Tx equalizer non LTI

- In this case, since you can generate $H_{txc}(\tau)$, you do not need txgetwave
- This is true in all cases including the case when x(t) is ‘distorted’ with duty cycle distortion and any other arbitrary modification
  - In fact x(t) can be any arbitrary continuous wave form
- The reason for this that the tx equalizer is independent of the bit generator and can be characerized by a transfer function $h_{tx}(\tau)$ .. i.e it is LTI

When do you need tx getwave

- You need getwave only in the case where
  - $H_{tx} == H_{tx}(\tau, x(t))$
  - the equalizer depends on its input (x(t)) and is ‘time varying’
    - In general cases like this dependency you may not have a constant transfer function $H(\tau)$
  - Now you need txgetwave
  - Txgetwave should output x1(t) , a modified continuous bit stream



Rational for x1(t) - modified bit stream output

- The analog front end channel is in the eda domain; x1(t) is from the tx device and is neatly separated
- The eda tool can produce an wave form y(t) at the rx input in a flexible manner.
  - For example if it is deemed that front end channel non linearity has to be taken into account it can be done
- Removes cross talk drive complication
- This division (i.e tx ouputs only x1(t) and does not go out of its domain to produce rx input) is more natural and should be the only one supported for txgetwave

Note about bit stream value

- The continuous input bit stream x(t) should nominally swing between 0 and 1 and with 0.5 as the center point.
- This notion is consistent with the view of bits as 0 and 1.
- Multi level input signals are also more easily specified with normalized signals of 0-1