# Schematic

## Modified DFE

There are some modification of DFE code. See the appendix for complete DFE Verilog code.

### DC Level Shift

The original DFE is incorrect, and will cause DC level shift. The following two lines are changed.

|  |
| --- |
| // Wrong !! This will change DC level of both signal.  // Use data\_history = 00000 to verify this.  // V(out) <+ slew(V(in) +tap1\*(2\*data\_history[0]-1)+**...**;  // V(outbar) <+ slew(V(inbar)-tap1\*(2\*data\_history[0]-1)-**...**;  // This is the correct one.  V**(**out**)** **<+** slew**(**V**(**in**)** **+**tap1**\***data\_history**[**0**]+****...;**  V**(**outbar**)** **<+** slew**(**V**(**inbar**)-**tap1**\***data\_history**[**0**]-...;** |

### External DFE controller

I use external DFE controller to provide clock and reset. The clk and rst should be wire.

|  |
| --- |
| // Change from logical to wire, since I am using external DFE\_controller,  // which provides clk and rst signal. See layout.  // logical clk, rst;  **wire** clk**,** rst**;**  **...**  **always@(posedge(**clk**),** rst**)** **begin**  // Since rst is changed to wire, I have to explicitly specify rst == 1 condition.  // if(rst) begin  **if(**rst **==** 1**)** **begin**  data\_history **<=** 5'b00000**;**  Dout **<=** 1'b0**;**  **End**  **...** |

## Simulation

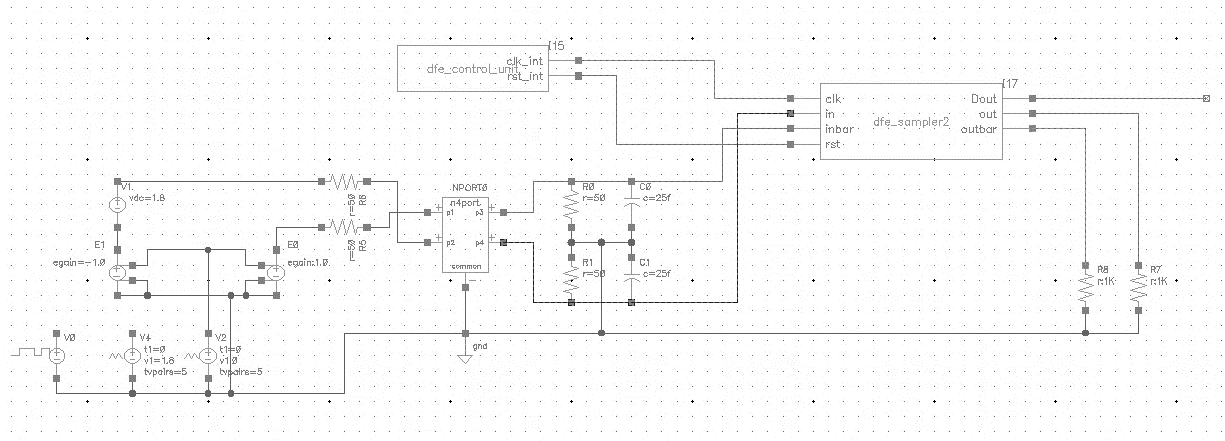
### Lower corner are the PRBS or PWL voltage sources

Initial delay for both sources are 1ns.

### Simulation setup

Maximum time step = 1ps. The larger time step will degrade eye, since DFE works on continuous time.

Maximum simulation time for eye measurement = 300 ns, or eye will be too coarse.



# Pulse Response before DFE (Tapi = 0)

## Differential Received Pulse Response and post cursors



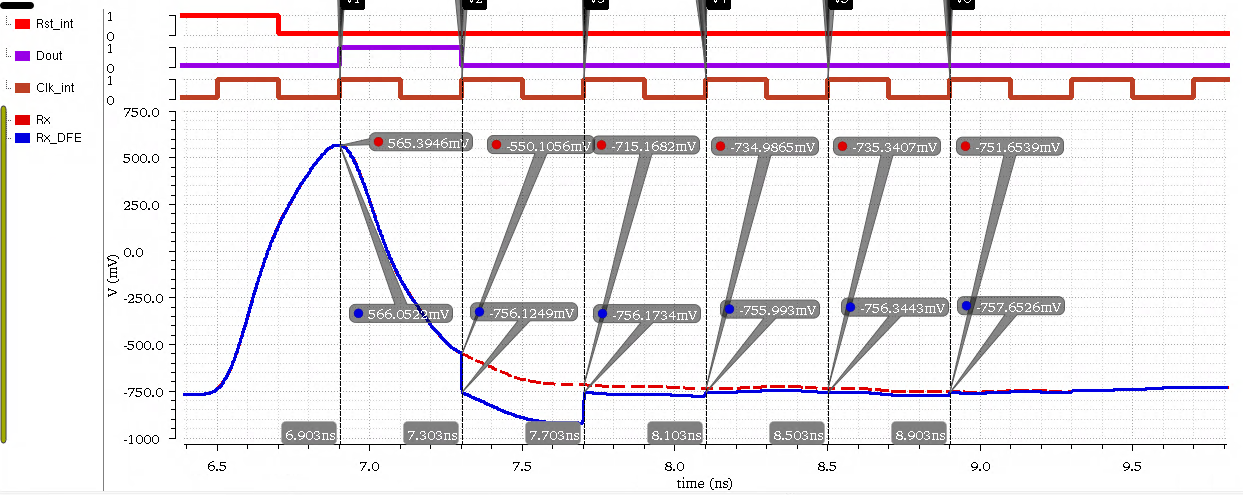
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Time (ns) | 6.901 | 7.301 | 7.701 | 8.101 | 8.501 | 8.901 | Steady |
| Rx\_Diff (mV) | 565 | 552 | 717 | 737 | 737 | 752 | 758 |
| Taps (m) |  | -103 | -20.5 | -10.5 | -10.5 | -3 |  |

|  |  |
| --- | --- |
| Rx\_Diff (Zoomed) | Tx\_Diff (Solid) , Rx\_Diff (Dash) |
|  |  |

# Pulse Response after DFE

To show the pulse response of DFE, the maximum time step of simulator must <=1ps, since DFE is working in continuous time. The resolution of DFE response is related to maximum time step significantly. Clk, Rst, and Dout signals also listed in figure.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Time |  | (ns) | 6.901 | 7.301 | 7.701 | 8.101 | 8.501 | 8.901 | Steady |
| Rx\_Diff | (Dash) | (mV) | 565 | 552 | 717 | 737 | 737 | 752 | 758 |
| Rx\_Diff after DFE | (Solid) | (mV) | 566 | 756 | 756 | 756 | 756 | 756 | 757 |
| Dout |  |  | 1 | 0 | 0 | 0 | 0 | 0 | 0 |



# Eyes

## Setup

Maximum time step = 1ps

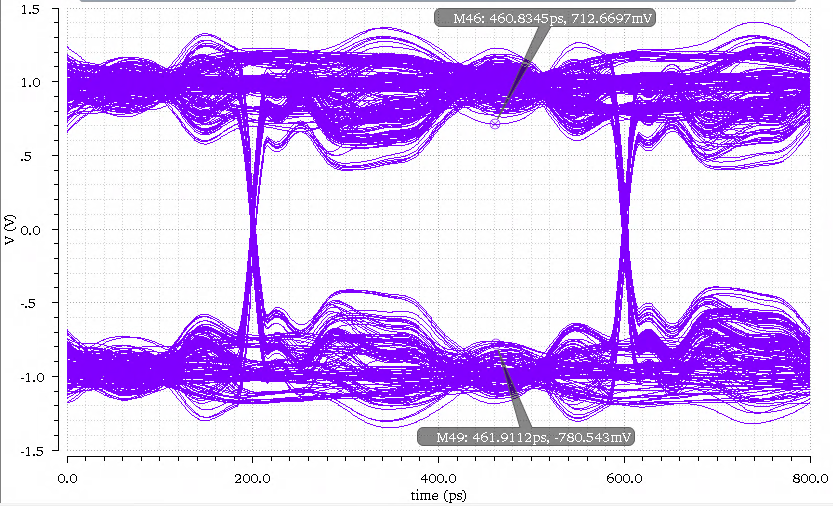
Maximum simulation time = 300ps.

## Eye Measure

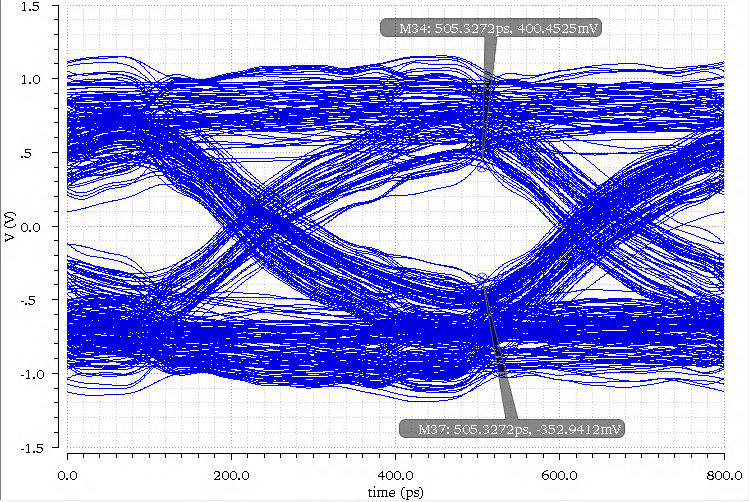
|  |  |  |  |
| --- | --- | --- | --- |
|  | Tx Diff | Rx Diff before DFE | Rx Diff after DFE |
| Upper (mV) | 712 | 400 | 312 |
| Lower(mV) | -780 | -352 | -576 |
| Eye Height(mV) | 1492 | 752 | 888 |

We can see the improvement of DFE is roughly 130 mV, which is very close to tap1 of DFE, since the first ISI is much larger than the others.

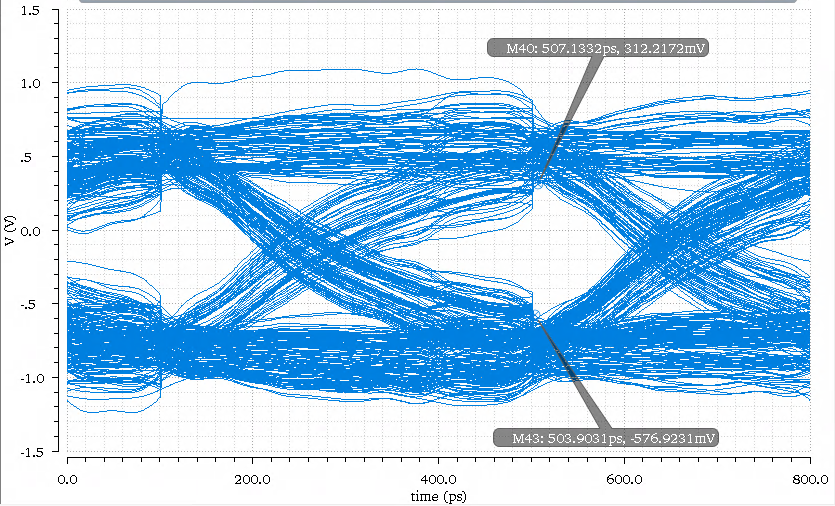
## Transmitted



## Received Before DFE

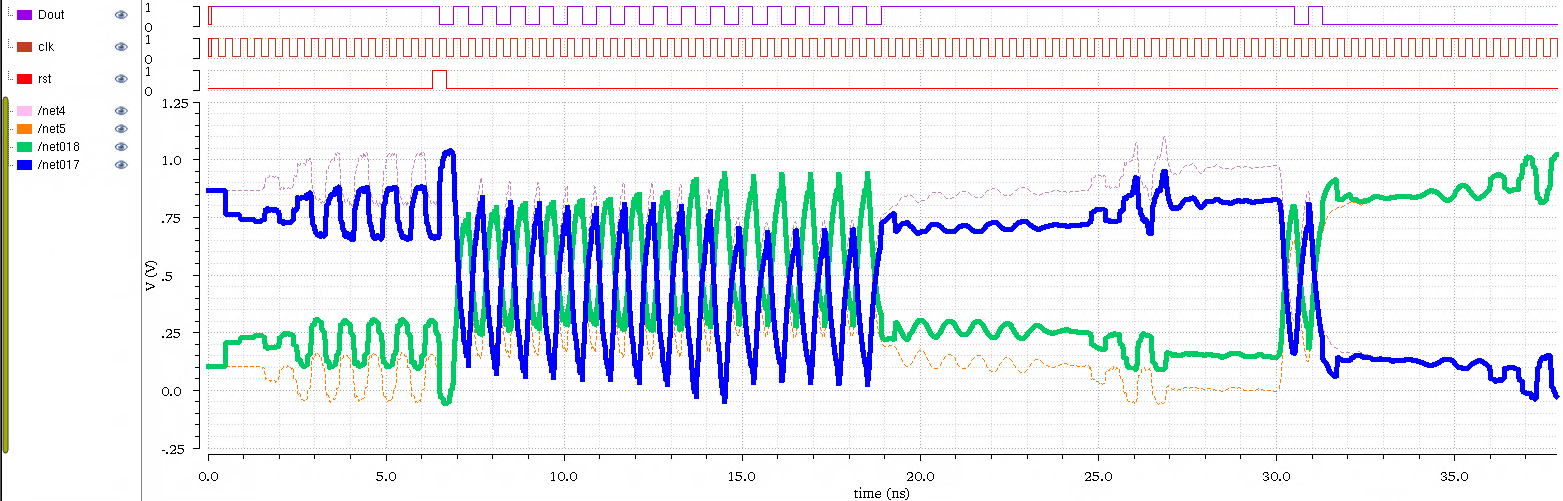


## Received After DFE



## Time Domain Signal (Partial)

Dash: Before DFE , Solid: After DFE



# Verilog Code

## Modified DFE

|  |
| --- |
| //Verilog-AMS HDL for "ece546", "dfe\_sampler2" "verilogams"  `include "constants.vams"  `include "disciplines.vams"  module dfe\_sampler2 (in, inbar, out, outbar, Dout, clk, rst);  input in, inbar, clk, rst;  output Dout, out, outbar;  reg Dout;  electrical in, inbar, out, outbar;  // Change from logical to wire, since I am using external DFE\_controller,  // which provides clk and rst signal. See layout.  // logical clk, rst;  wire clk, rst;  parameter real tap1 = - 0.103;  parameter real tap2 = - 0.0205;  parameter real tap3 = - 0.0105;  parameter real tap4 = - 0.003;  parameter real tap5 = - 0.003;  reg[4:0] data\_history;  analog begin  // Wrong !! This will change DC level of both signal. Use data\_history = 00000 to verify this, and negative slew rate should be negative(produce warning).  // V(out) <+ slew(V(in) +tap1\*(2\*data\_history[0]-1)+tap2\*(2\*data\_history[1]-1)+tap3\*(2\*data\_history[2]-1)+tap4\*(2\*data\_history[3]-1)+tap5\*(2\*data\_history[4]-1),1e11, 1e11);  // V(outbar) <+ slew(V(inbar)-tap1\*(2\*data\_history[0]-1)-tap2\*(2\*data\_history[1]-1)-tap3\*(2\*data\_history[2]-1)-tap4\*(2\*data\_history[3]-1)-tap5\*(2\*data\_history[4]-1),1e11, 1e11);  // This is the correct one.  V(out) <+ slew(V(in) +tap1\*data\_history[0]+tap2\*data\_history[1]+tap3\*data\_history[2]+tap4\*data\_history[3]+tap5\*data\_history[4],1e11);  V(outbar) <+ slew(V(inbar)-tap1\*data\_history[0]-tap2\*data\_history[1]-tap3\*data\_history[2]-tap4\*data\_history[3]-tap5\*data\_history[4],1e11);  end  always@(posedge(clk), rst) begin  // Since rst is changed to wire, I have to explicitly specify rst == 1 condition.  // if(rst) begin  if(rst == 1) begin  data\_history <= 5'b00000;  Dout <= 1'b0;  end  else begin  if(V(out) - V(outbar) > 0.2)  Dout <= 1'b1;  else if (V(out) - V(outbar) < -0.2)  Dout <= 1'b0;  data\_history[4:0] = {data\_history[3:0],Dout};  end  end  endmodule |

## DFE Controller

|  |
| --- |
| //Verilog-AMS HDL for "ECE546", "dfe\_control\_unit" "verilogams"  `include "constants.vams"  `include "disciplines.vams"  `timescale 10ps/1ps  // Make sure the transient maximum time step must <= 1ps to see the  // significance of DFE, or the time response will not be accurate,  // which degrade the performance of DFE.  module dfe\_control\_unit (clk\_int, rst\_int);  output clk\_int, rst\_int;  parameter real delayUnit = 690.131; // delay is approx. 6.9013 ns  parameter cycle = 20; // clock period (20\*10ps)  reg rst\_int;  reg clk; // generate the internal clk.  initial begin  clk = 1;  end  // The reset signal has to a cycle earlier than the real signal  // Another half cycle guarantee the correctness.  initial begin  rst\_int = 0;  #(delayUnit - cycle/2 - cycle) rst\_int = 1;  #(cycle) rst\_int = 0;  end  always #(cycle/2) clk = ~clk;  //Delay the clk to clk\_int  wire clk\_int;  assign #(delayUnit - $floor(delayUnit/cycle)\*cycle) clk\_int = clk;  endmodule |