EE577b Viterbi Decoder Project

by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

EE577B Viterbi Decoder Project Report

EE577b Viterbi Decoder Project

by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

TABLE OF CONTENTS

EE577b Viterbi Decoder Project
by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

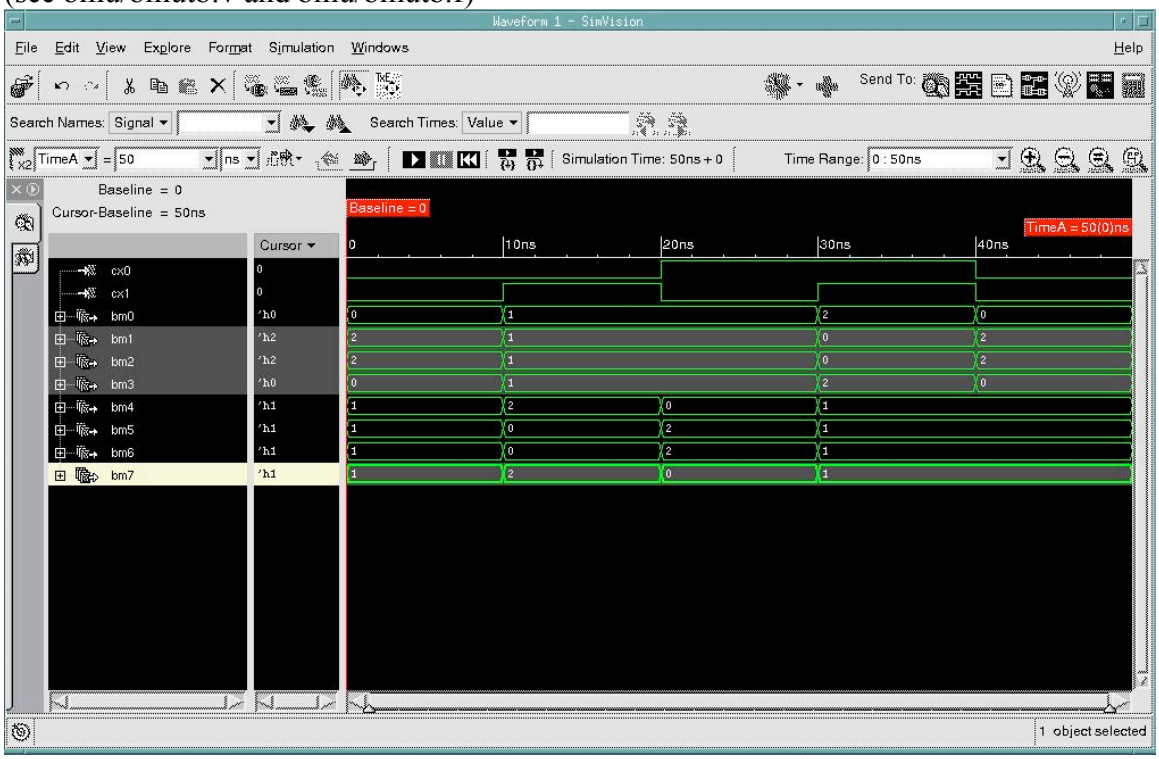
1 Verilog

1.1 BMU Design

The Branch Metric Unit (BMU) of the Viterbi Decoder computes the hamming distance between the received bits cx0 and cx1 and the defined outputs for each of the eight possible branches. The BMU is all combinational logic.
(see bmu/bmu.v)

1.2 BMU Test – Functionality

(see bmu/bmutb.v and bmu/bmutb.f)



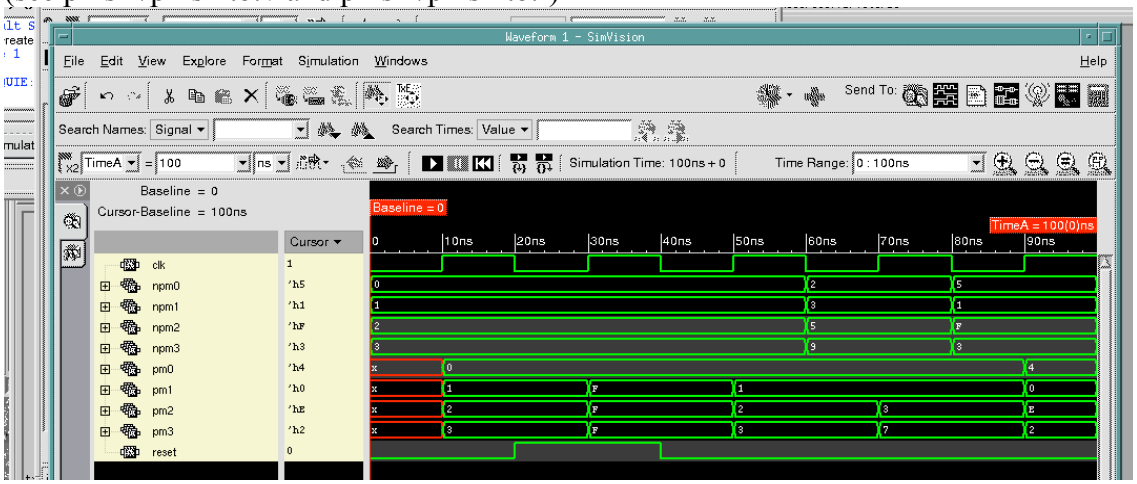
by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

1.3 PMSM Design

The Path Metric State Memory (PSMS) module is made of two blocks. The first block is combinational and normalizes the four inputs by subtracting the smaller of them from all 4 inputs. The second block is sequential and contains Flip flops to store the values of pm0-pm3.
(see pmsm/pmsm.v)

1.4 PMSM Test – Functionality

(see pmsm/pmsmtb.v and pmsm/pmsmtb.f)



1.5 SPD Design

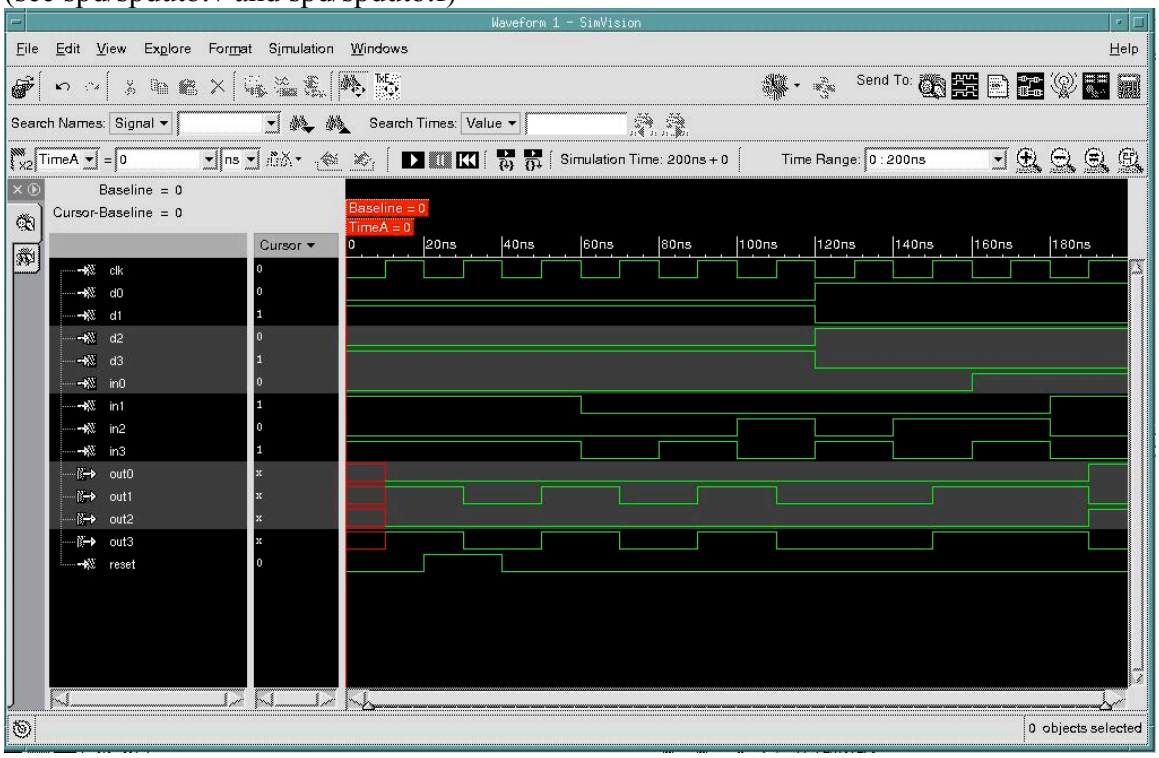
The SPD is composed of many sub parts. The Survivor Path Decoder (SPD) contains 15 identical Survival Path Decoder Units (SPDU), one 2-4 multiplexer, and a circuit block that chooses the smallest of four metrics (selector).

1.5.1 SPDU Design

Each SPDU is composed of two blocks. The first block is combinational and uses the for 'd' signals to select the appropriate 4 paths (like a set of 4 multiplexers).
The second block is sequential and contains Flip flops to store the values of the four surviving paths. There are 15 SPDUs in the SPD because this is 5 times the constraint length (3).
(see spd/spdu.v)

1.5.2 SPDU Test – Functionality

(see spd/spdutb.v and spd/spdutb.f)

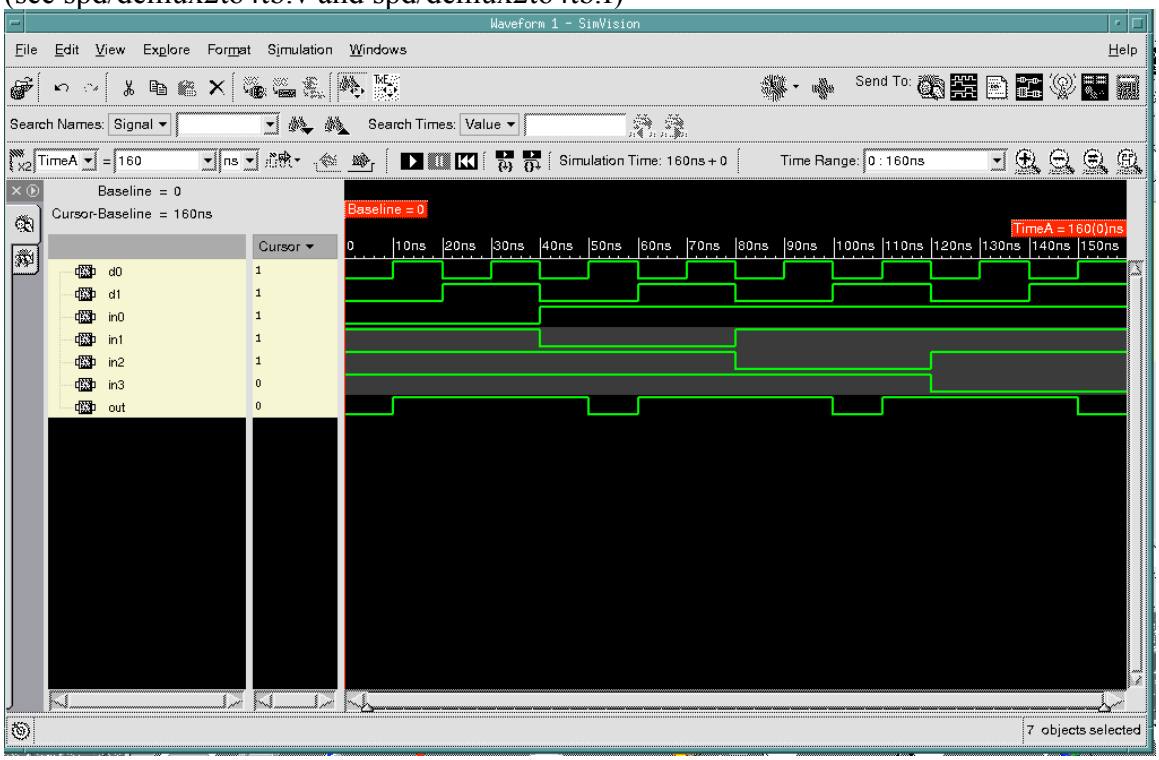


1.5.3 DEMUX2TO4 Design

The DEMUX2to4 is a combinational circuit. It implements a 4-to1 de-multiplexer. (see spd/demux2to4.v)

1.5.4 DEMUX2TO4 Functionality

(see spd/demux2to4tb.v and spd/demux2to4tb.f)



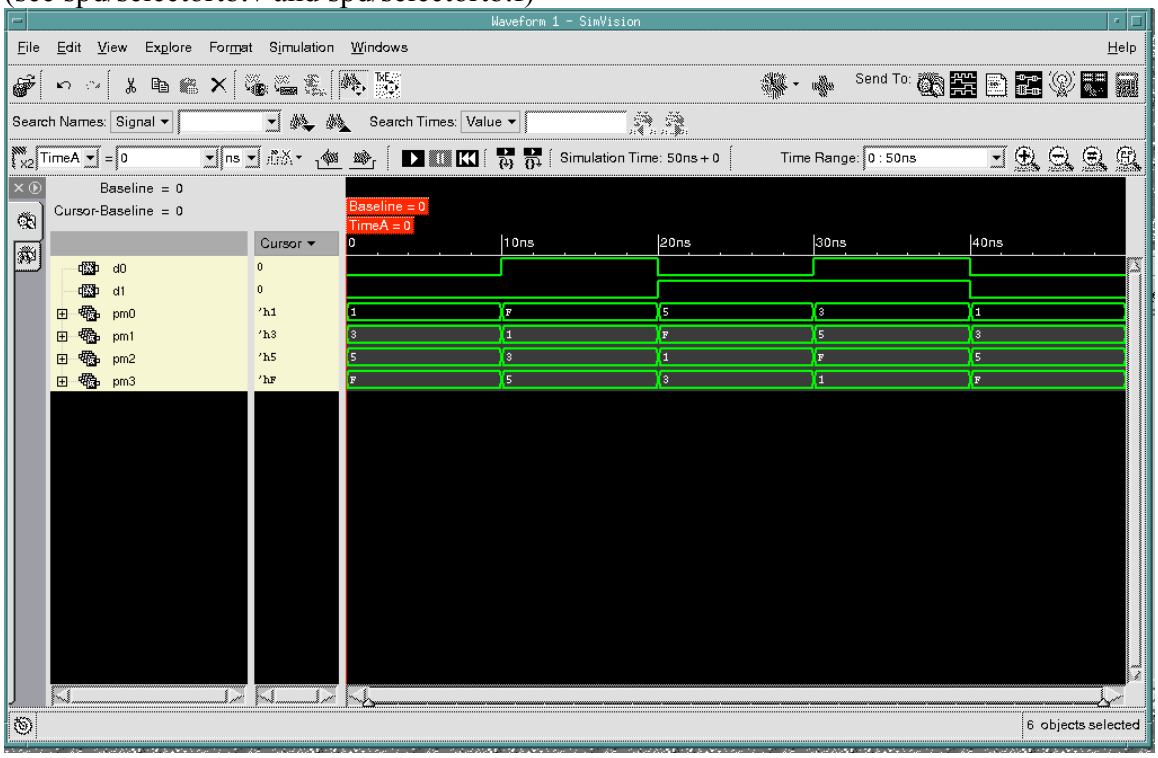
by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

1.5.5 SELECTOR Design

The selector is a combinational circuit. It inputs four 4-bit signals in ports 0-3 and outputs the port number (in binary) of the smallest input.
(see spd/selector.v)

1.5.6 SELECTOR Functionality

(see spd/selectortb.v and spd/selectortb.f)



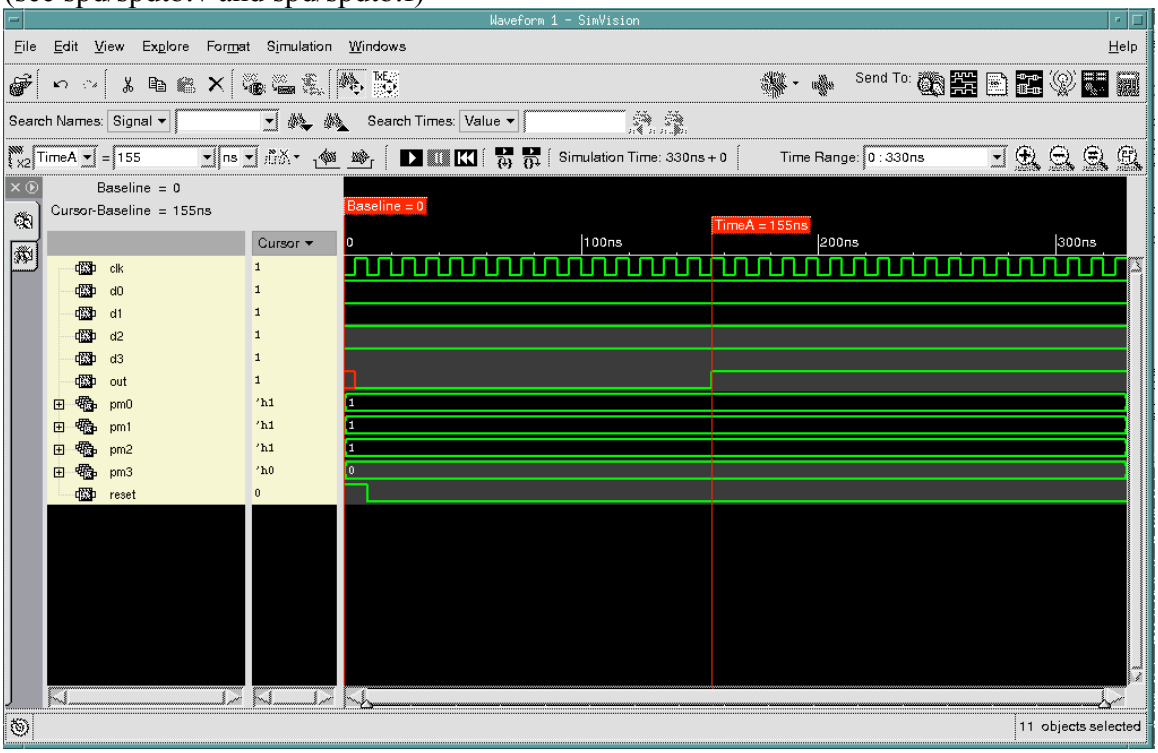
by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

1.5.7 SPD Top Level Design

There are 15 cascaded SPDUs. in0-in3 of the first SPDU are hard wired to 0's and 1's. All SPDU d0 signals are shorted together (and will be connected to the output of a single ACS cell in the top level design). The same is true about the d1, d2, and d3 signals. The final out singlals of the 15th SPDU are connected to the demux, which is controlled by the selector. The inputs to the selector are pm0-pm3 (these will be connected to the pmsm signals with the same names in the top level design). The output of the demux is the output of the Viterbi decoder.
(see spd/spd.v)

1.5.8 SPD Top Level Functionality

Test case 1 – see if d0-d3=1 and PM3 is constantly the lowest, see if the output is "1" 15 clocks after reset goes low.
(see spd/spdtb.v and spd/spdtb.f)

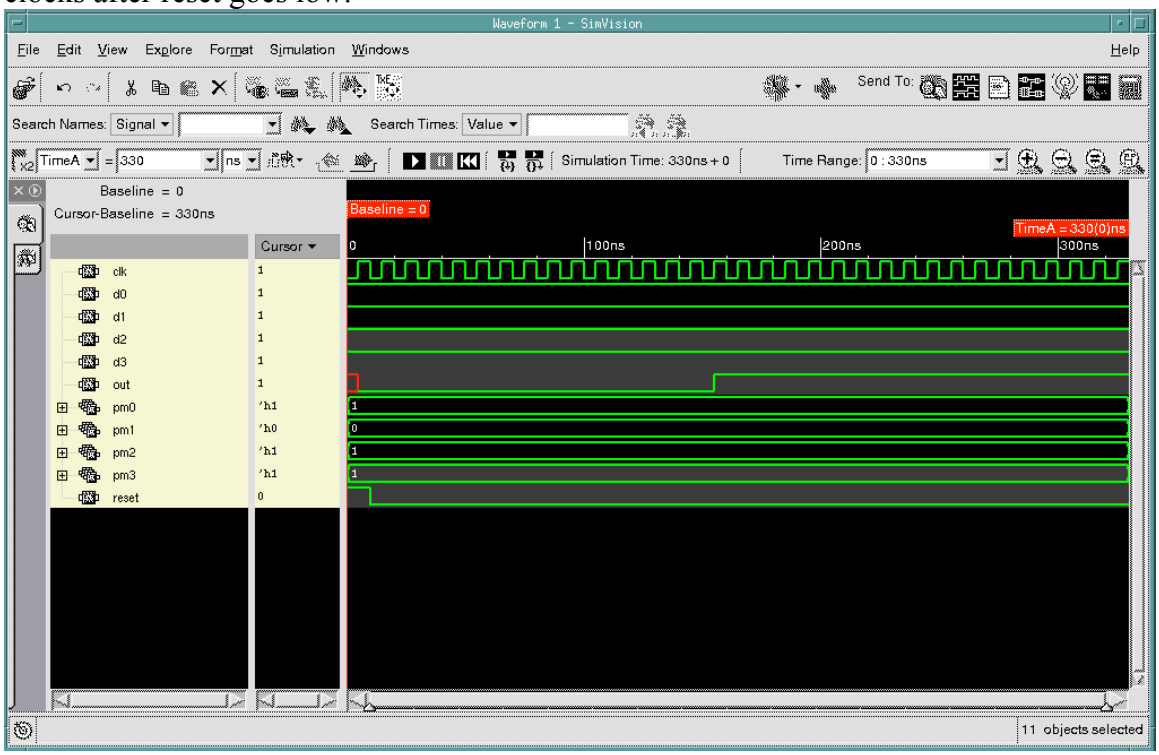


(Successful)

EE577b Viterbi Decoder Project

by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

Test case 2 – see if d0-d3=1 and PM1 is constantly the lowest, see if the output is “1” 15 clocks after reset goes low.

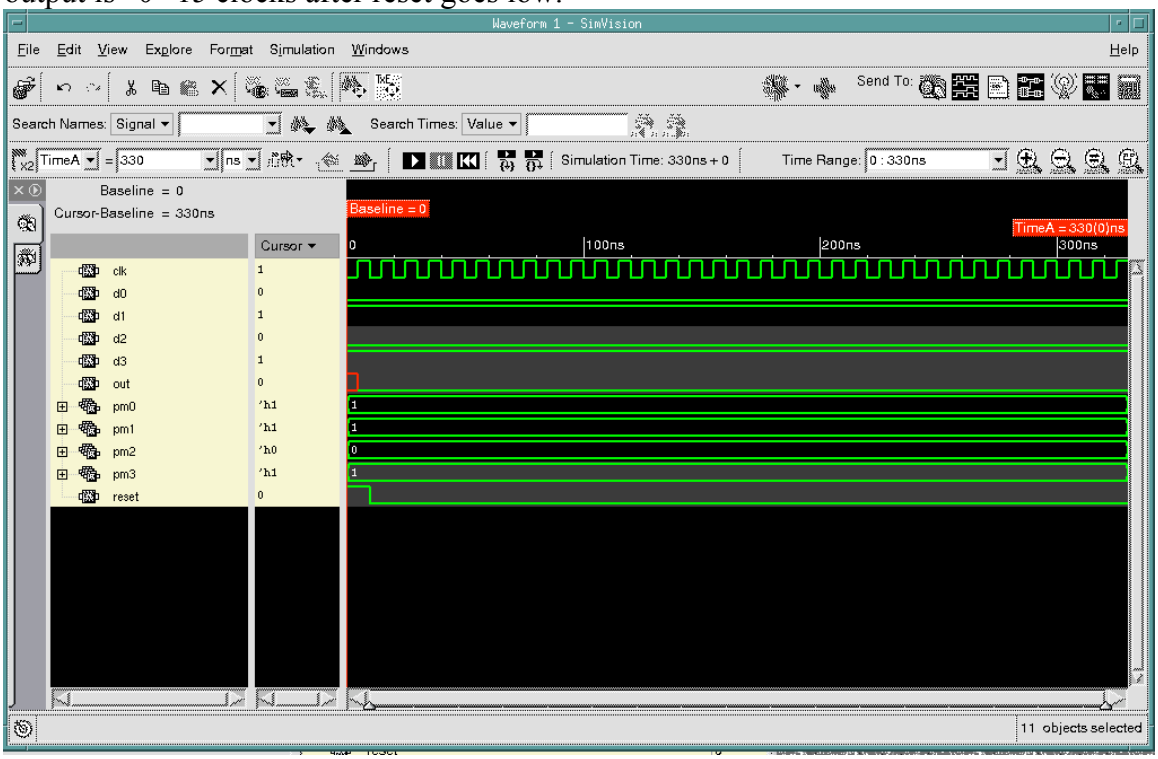


(Successful)

EE577b Viterbi Decoder Project

by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

Test case 3 – see if d0=d2=0 and d1=d3=1 and PM2 is constantly the lowest, see if the output is “0” 15 clocks after reset goes low.

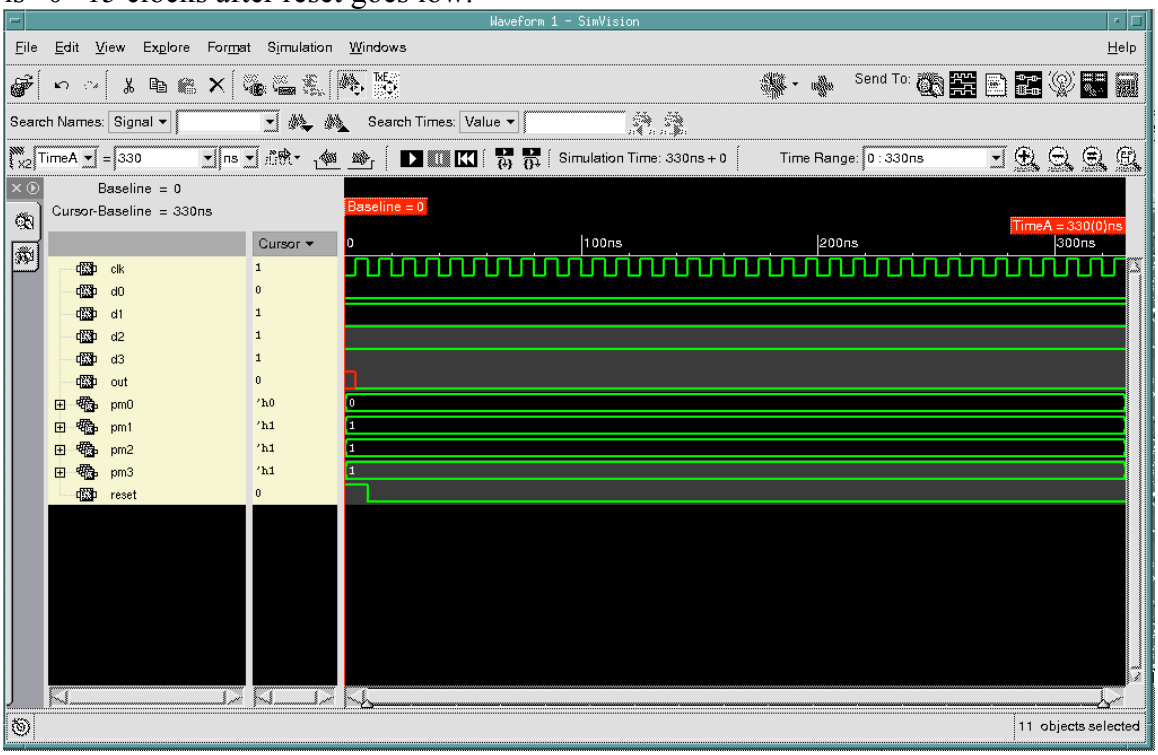


(Successful)

EE577b Viterbi Decoder Project

by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

Test case 4 – see if d0=0 and d1-d3=1 and pm0 is constantly the lowest, see if the output is “0” 15 clocks after reset goes low.



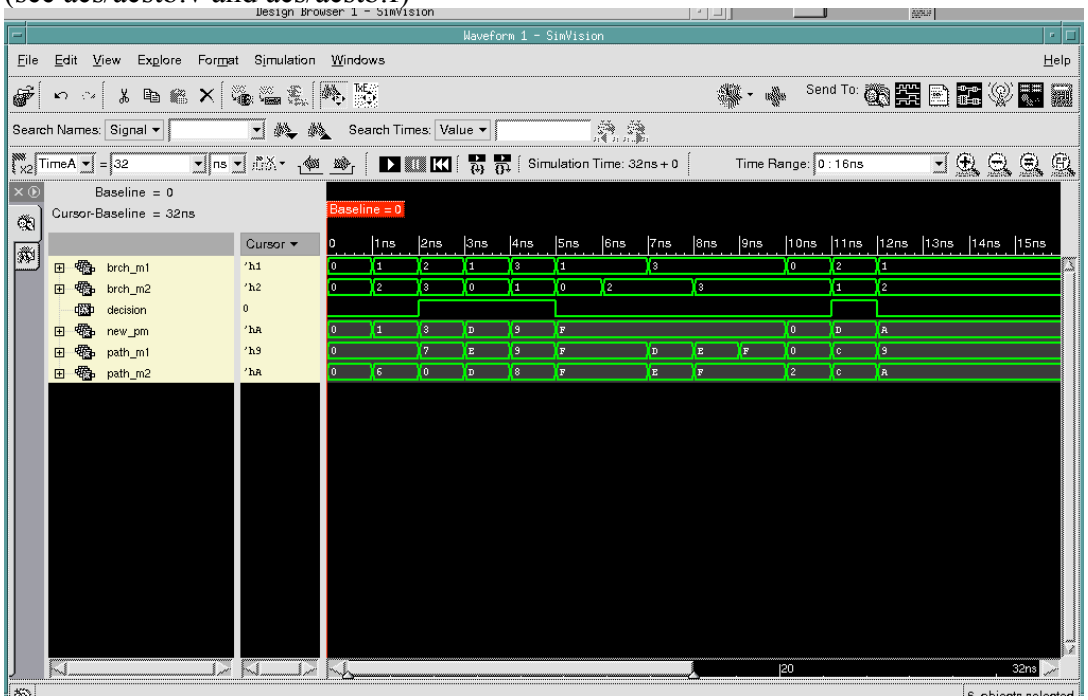
(Successful)

1.6 ACS Top Level Design

The ACS has 4 inputs – two 2-bit branch metrics and two 4-bit path metrics. The ACS has two outputs – the 4-bit new path metric, and a decision bit. The ACS adds branch metric0 and path metric0, and adds branch metric1 and path metric1. It then compares these two sums and selects the smaller of the two as the new path metric. If the sum of branch metric0 and path metric0 the smallest, the decision bit is set to 0. If the sum of branch metric1 and path metric1 is the smallest, the decision bit is set to 1. Special logic is included to ensure that if the total of the sum of the branch and path metric exceed 15, the output of the adder will be 15.
(see acs/acs.v)

1.7 SPD Top Level Functionality

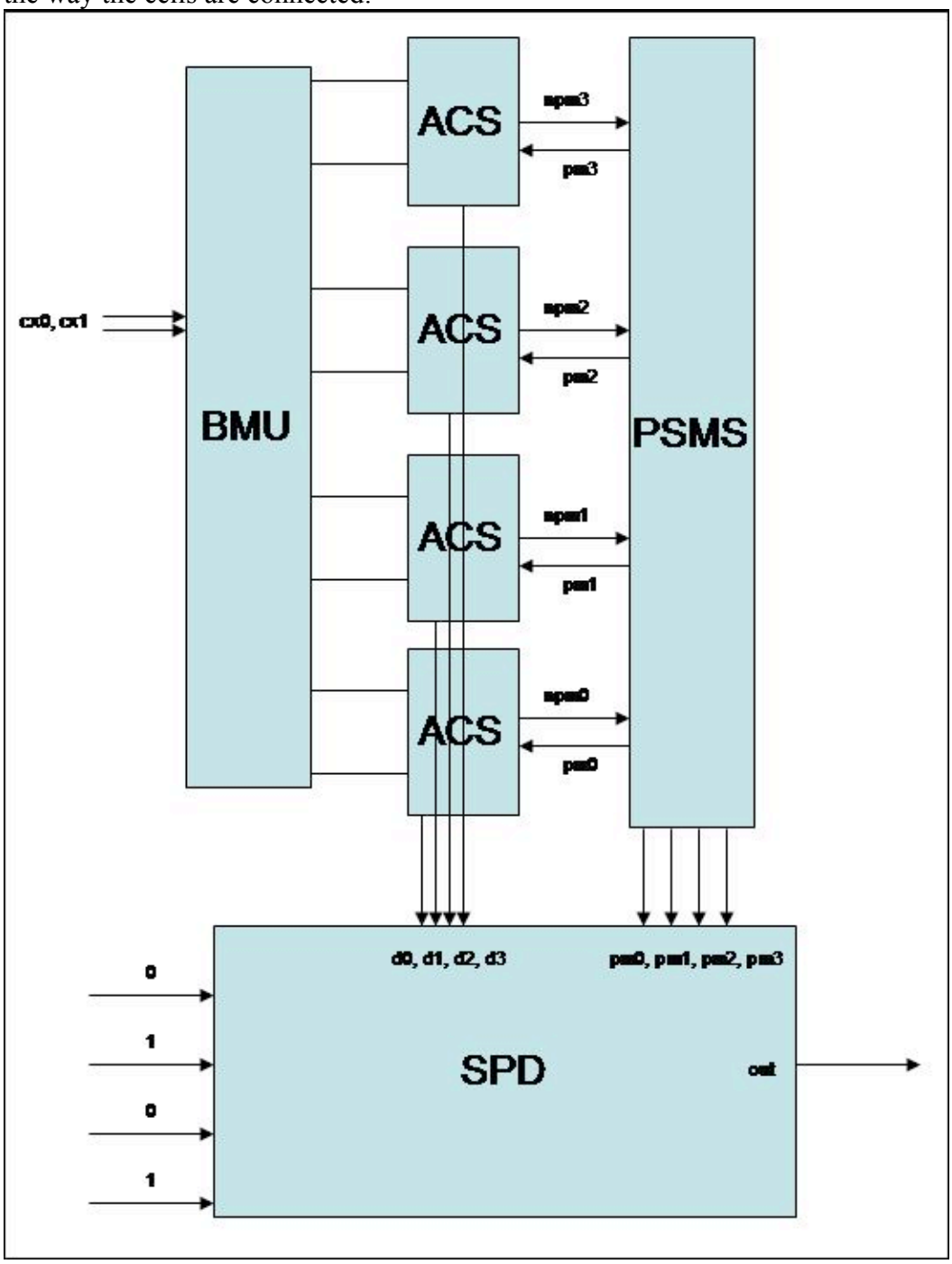
Various input combinations are tested to ensure that the new branch metric and decision bit are correctly produced. Several cases excercize the saturation logic built into the adders.
(see acs/acstb.v and acs/acstb.f)



by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

1.8 Viterbi Top Level Design

The Viterbi decoder is composed of many sub circuits. There is a single SDU (containing 15 stages, a 4-1 demux, and a comparator), 1 BMU, 4 ACSs, and 1 PSMS. The inputs are cx0, cx1 (connected to the inputs each BMU), clk, and reset (connected to the PSMS and SDU). The output is a serial bit stream at the pin "out", which is connected to the output of the SDU circuit. Internally, the following schematic explains the way the cells are connected:

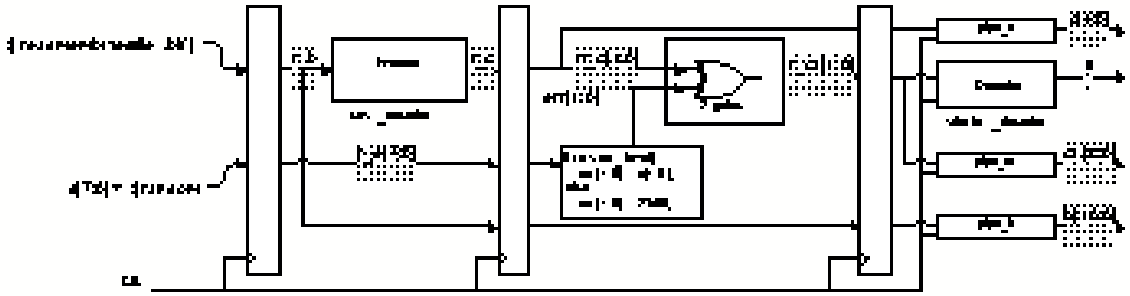


Note reset and clk signals not shown above for simplicity.

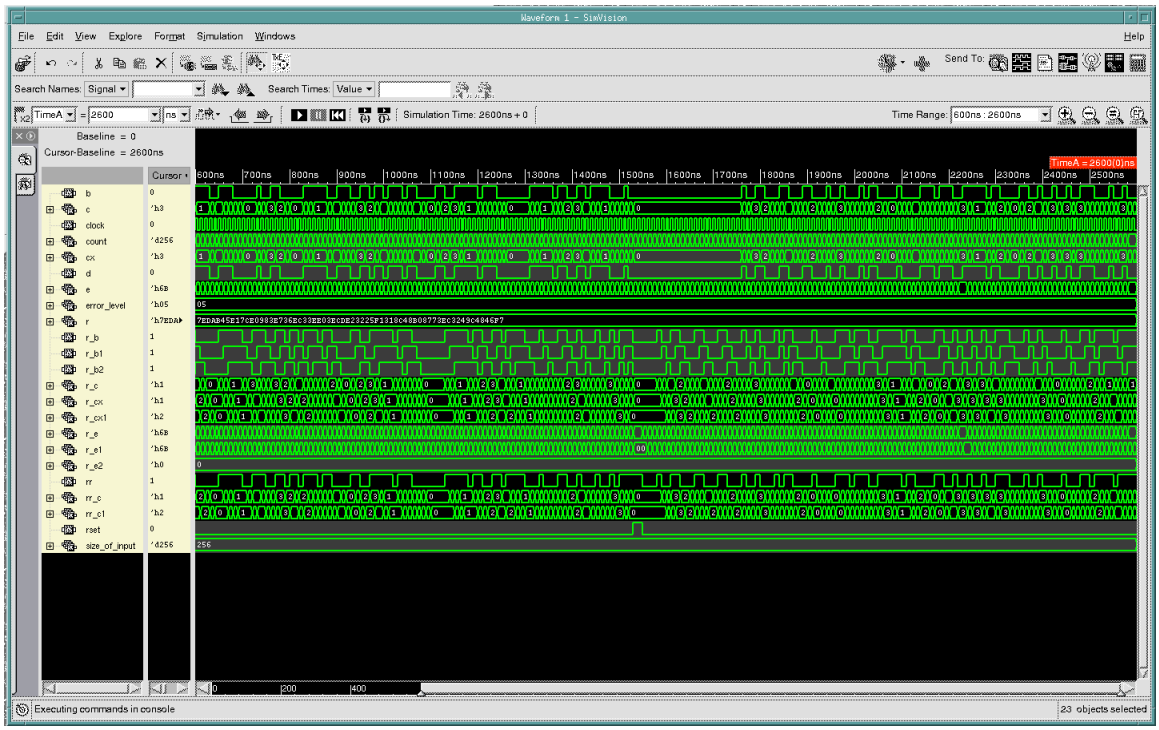
by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

1.9 Top level Viterbi module Functionality

The top level Viterbi decoder functionality was verified using the testbench described in the assignment (and shown below):



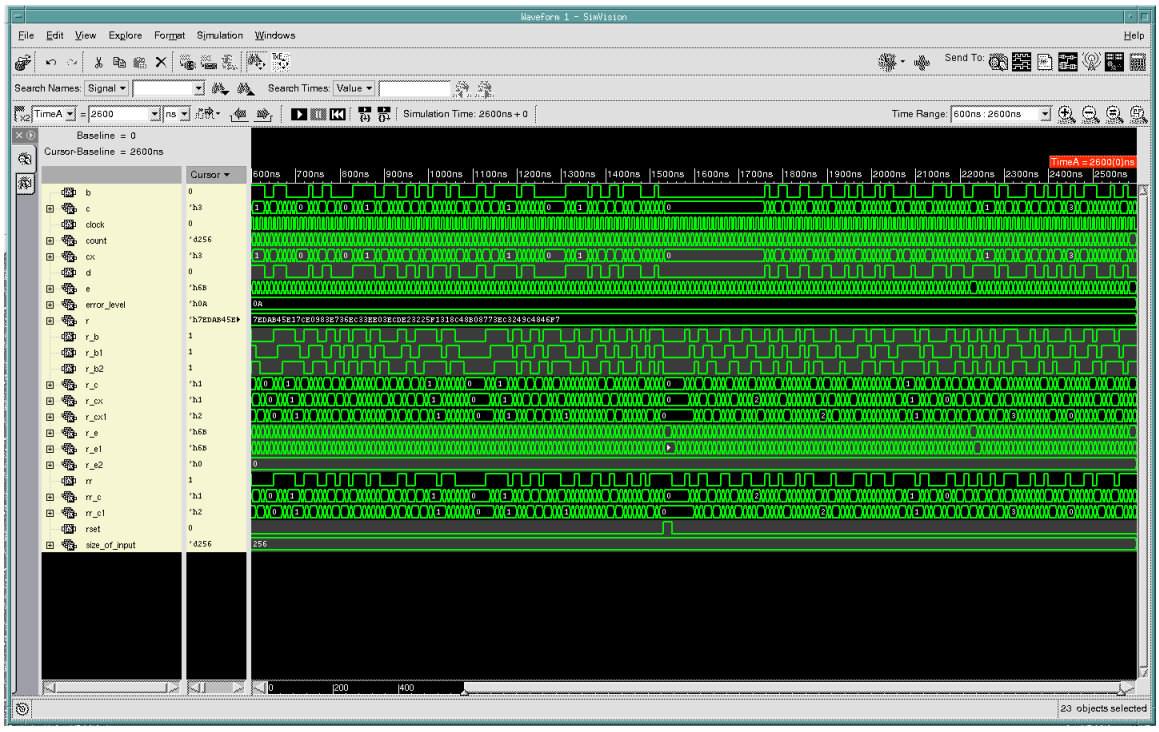
Error=5



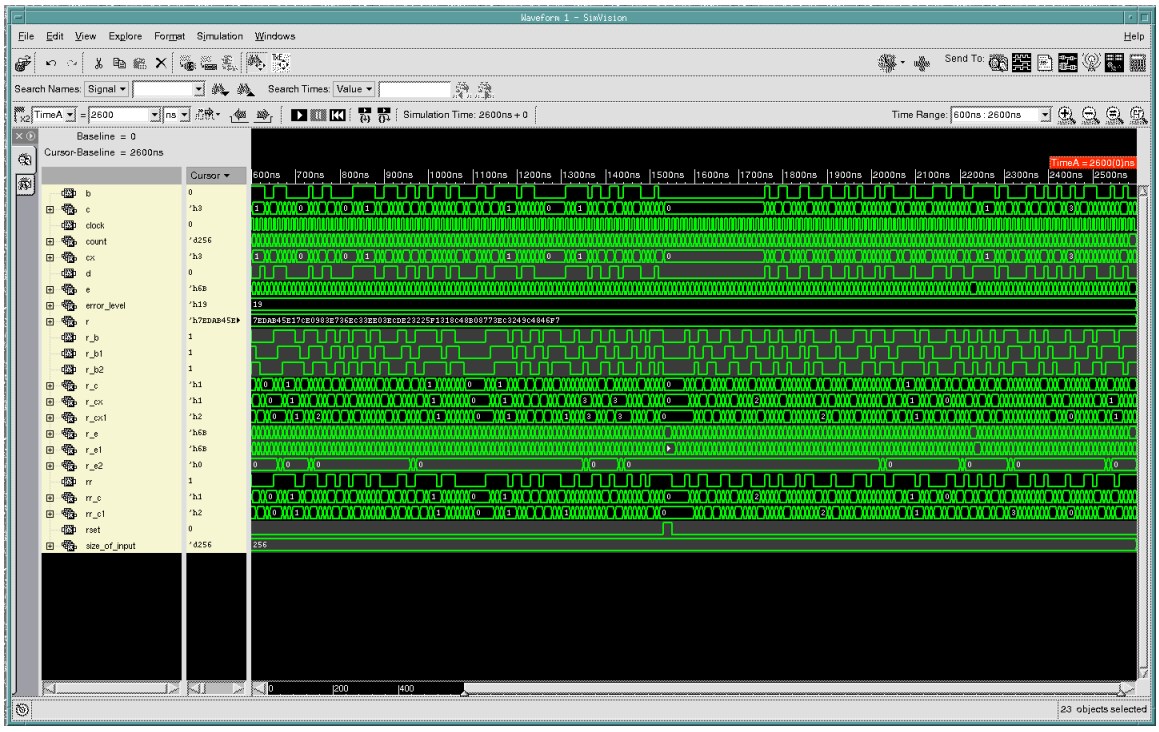
EE577b Viterbi Decoder Project

by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

Error = 10



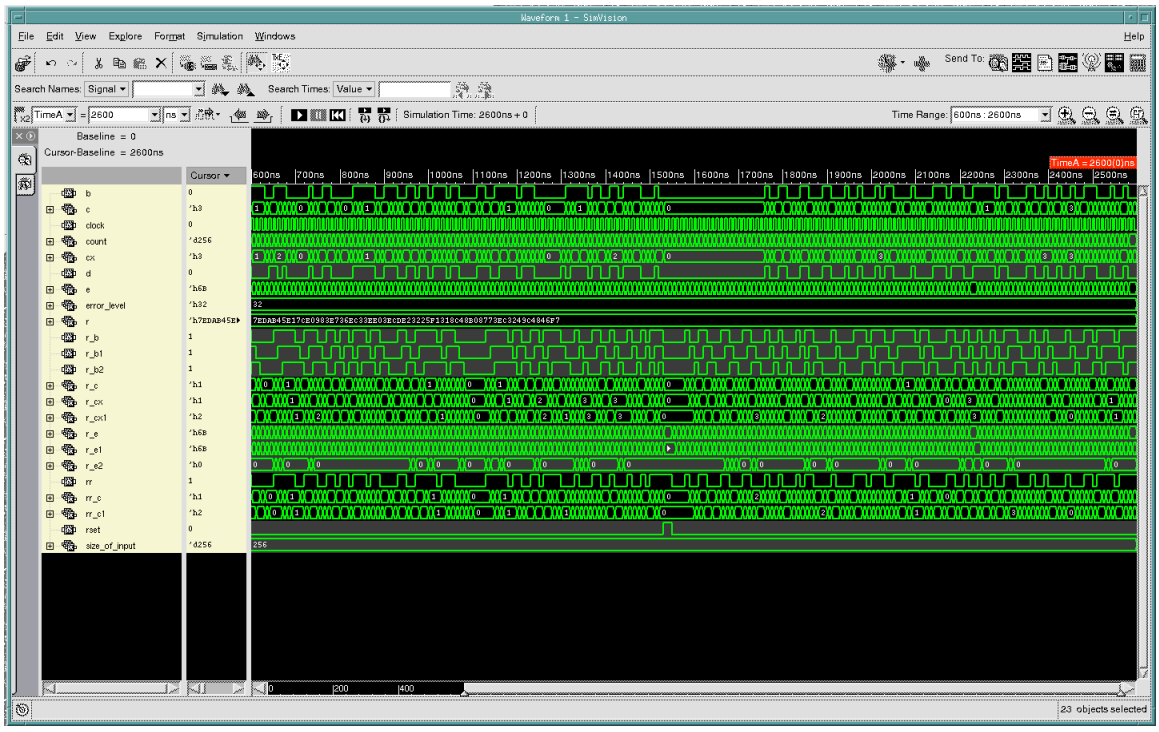
Error = 25



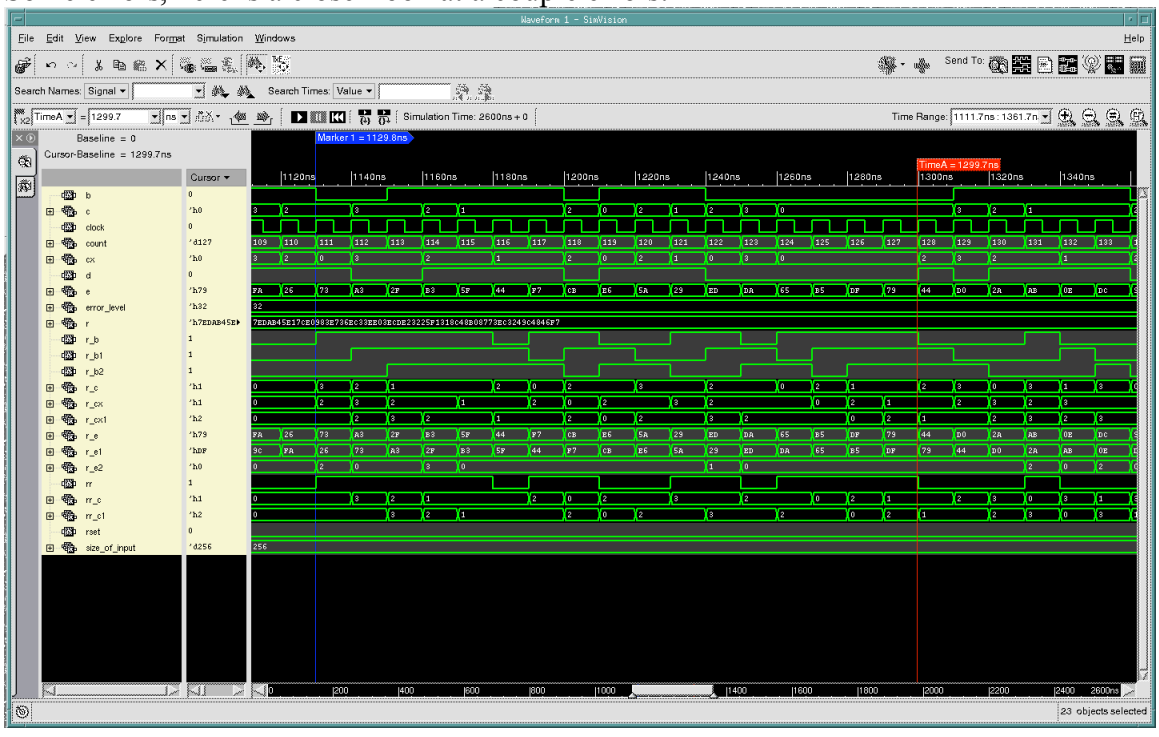
EE577b Viterbi Decoder Project

by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

Error = 50



Some errors, here is a closer look at a couple errors:

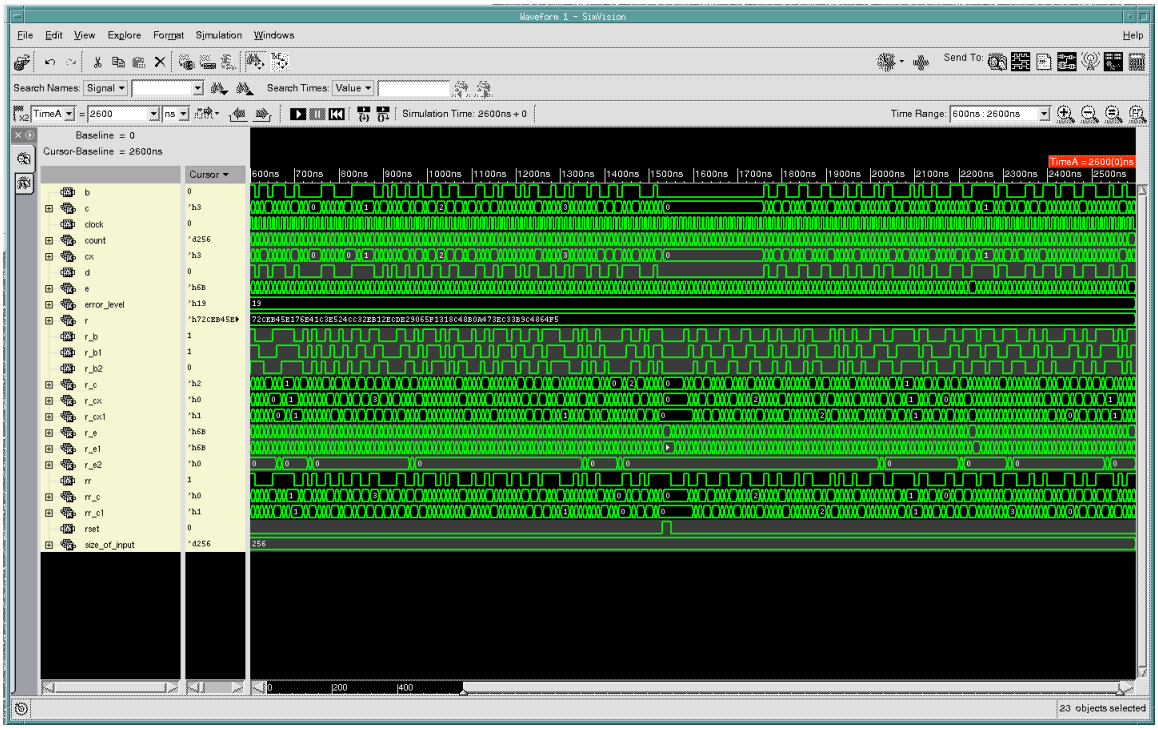


EE577b Viterbi Decoder Project

by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

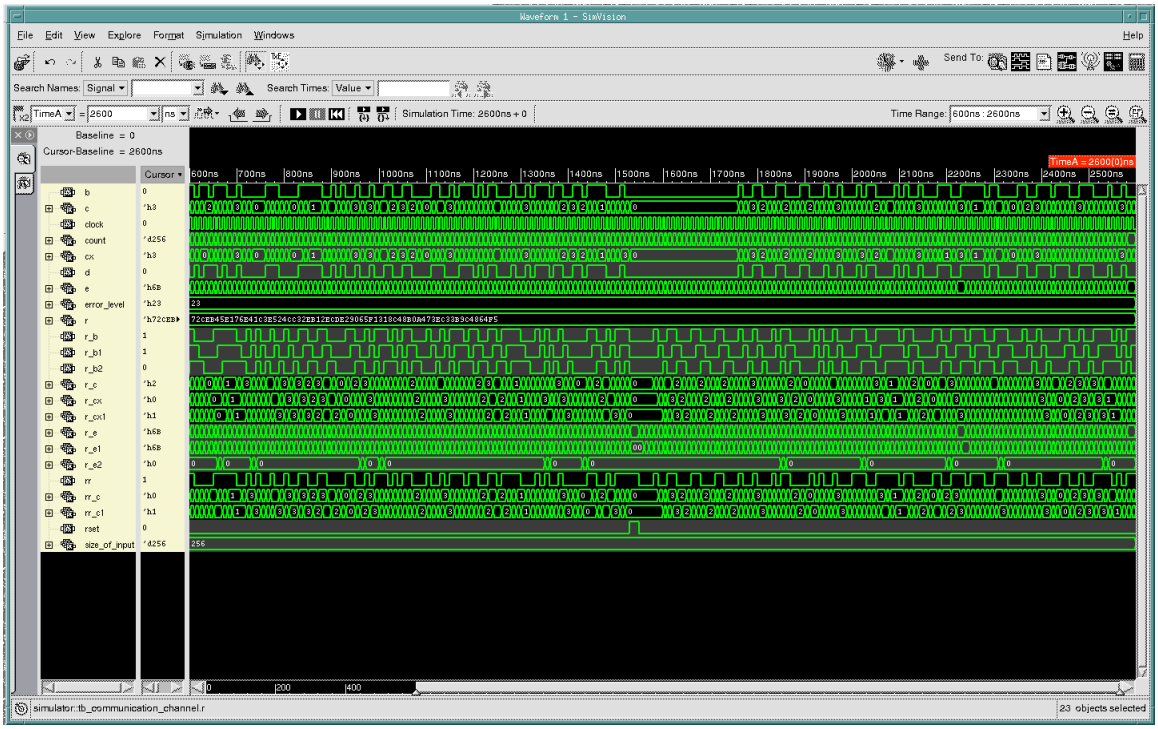
Second Data Set

Error=25



(No errors)

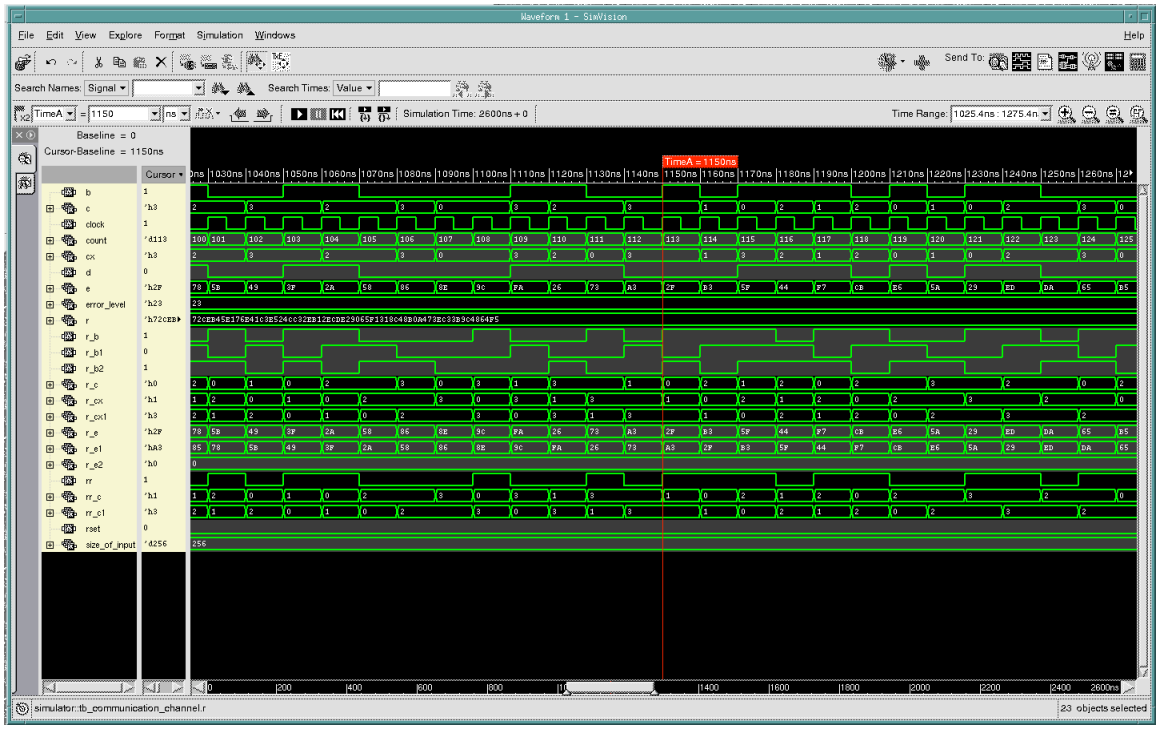
Error=35



Errors, see below:

EE577b Viterbi Decoder Project

by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com



2 Synthesis

Number of ports: 5
Number of nets: 57
Number of cells: 7
Number of references: 7
Area was calculated as 25455.000000 squares.
Timing was calculated to be 2.16r on the critical path.

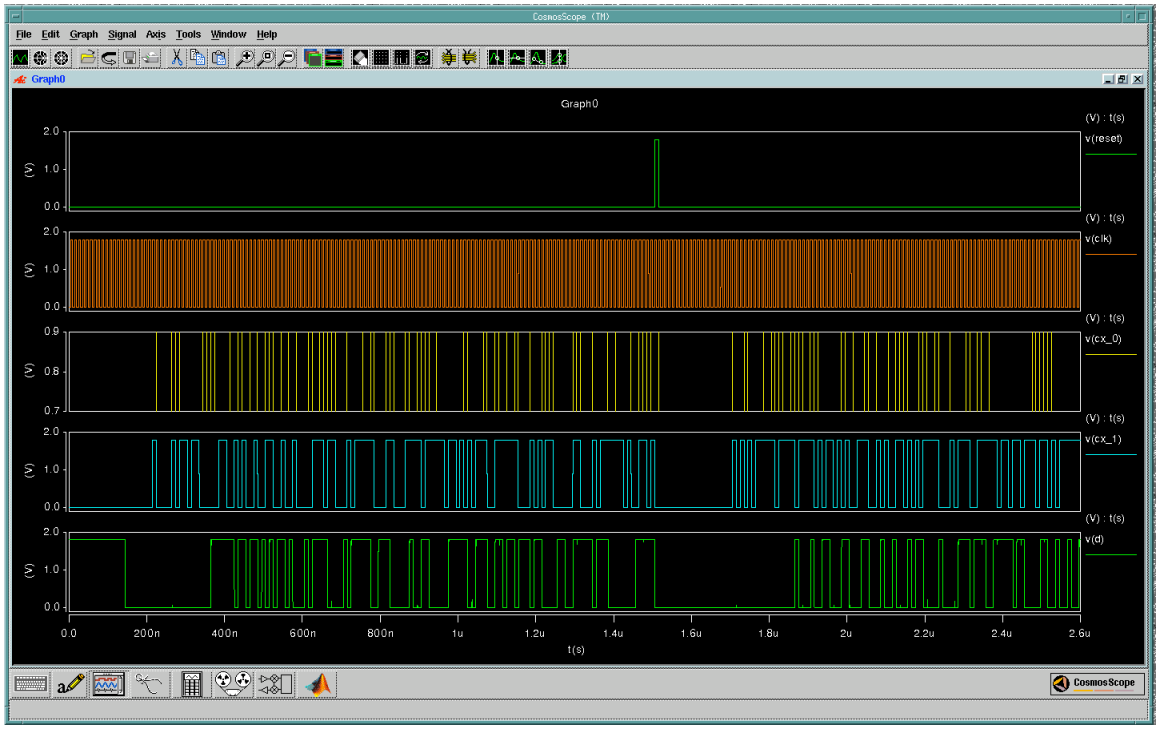
The authors were unable to create a SPICE netlist from the original synthesized Verilog netlist. However, the authors discovered that this error was due to the assign statements in Branch-Metric-Unit module of the synthesized Verilog netlist. In addition, the authors realized that some signals are redundant, and replaced them with buffers accordingly. That is, if signals a1[1]=a2[1] and a1[0]=a2[0], a buffer can be used to generate a2 from a1. This is applied for all redundant signals that cannot be uniquely determined, and depend on other signals for their logic values. Hence, assign statements in the behavioral Verilog code segment of the synthesized Verilog netlist have replaced with buffers in structural Verilog format. This allows the authors to proceed and perform SPICE netlist extraction in Cadence Virtuoso, and perform circuit simulation with Nanosim without impeding their progress.

3 NanoSim Results

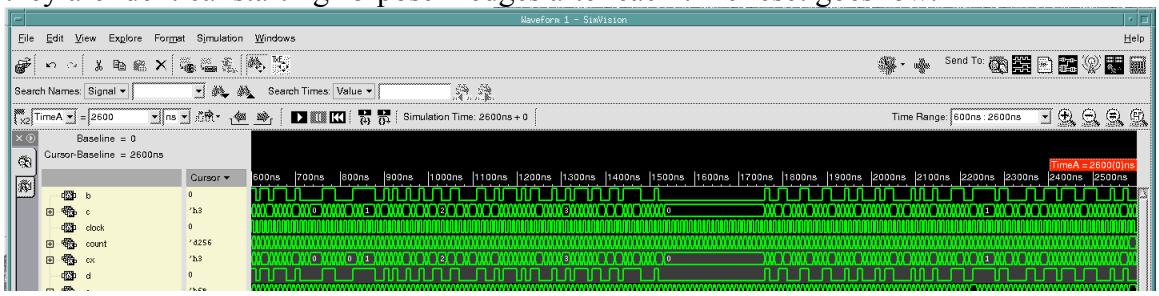
After extracting the netlist, nanosim was used to verify the functionality. At first a 10ns period clock was used just to verify functionality. The same data pattern from the second data set above was used. We did this by extracting the CX_0 and CX_1 values from the simulation data. The results are shown below.

EE577b Viterbi Decoder Project

by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

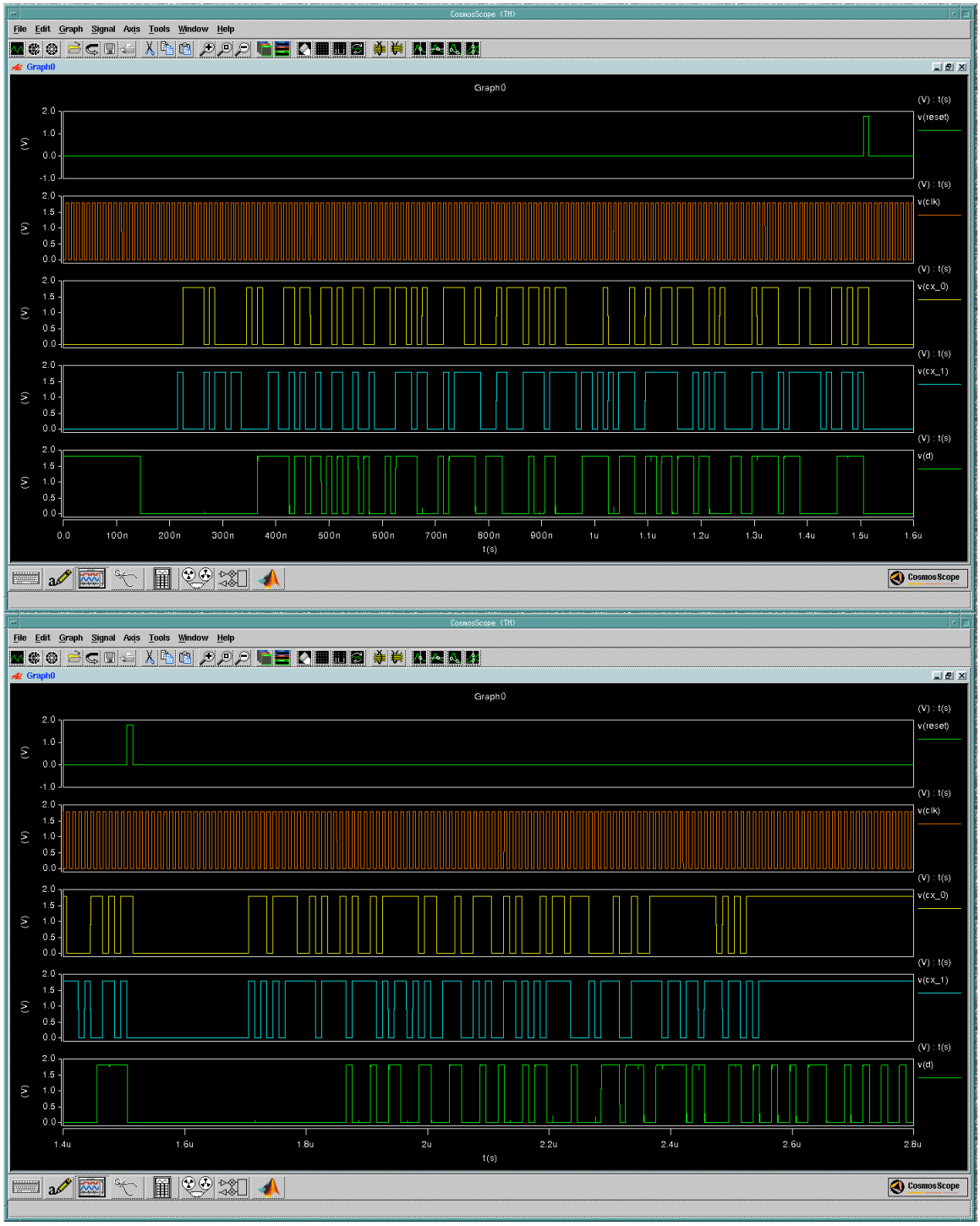


The above can be compared with the previous results (shown just below. As you can see, they are identical starting 15 posclk edges after each time reset goes low.



EE577b Viterbi Decoder Project

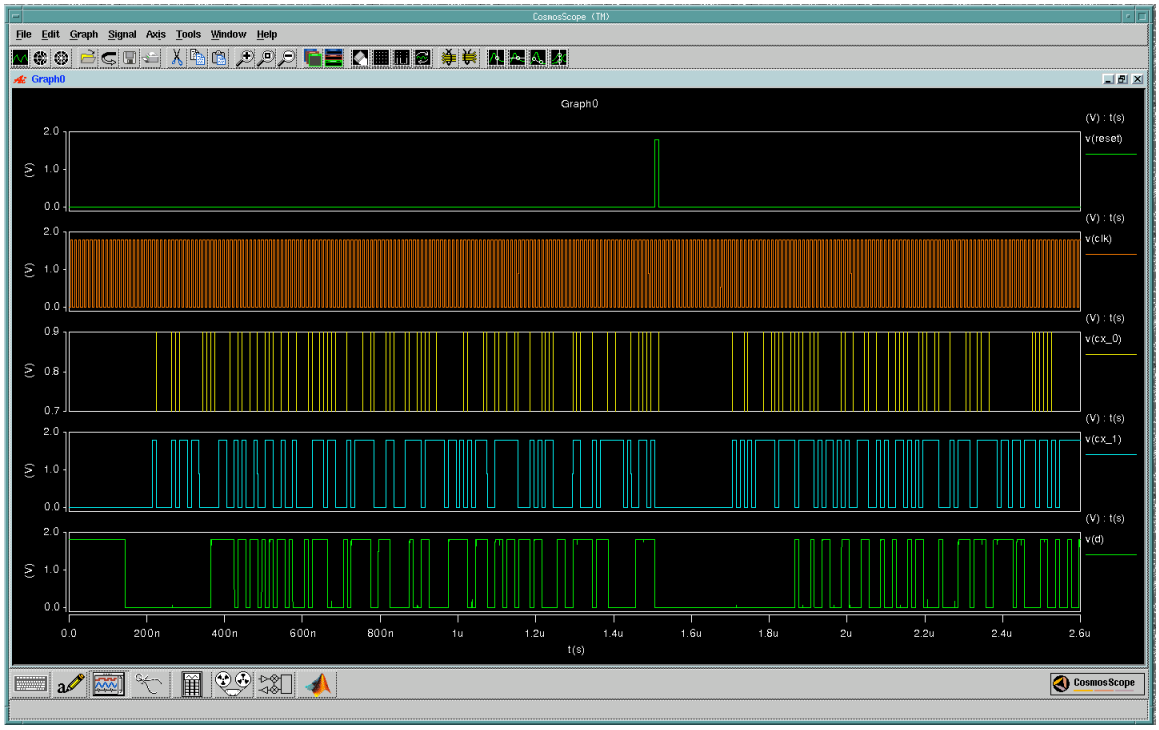
by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com



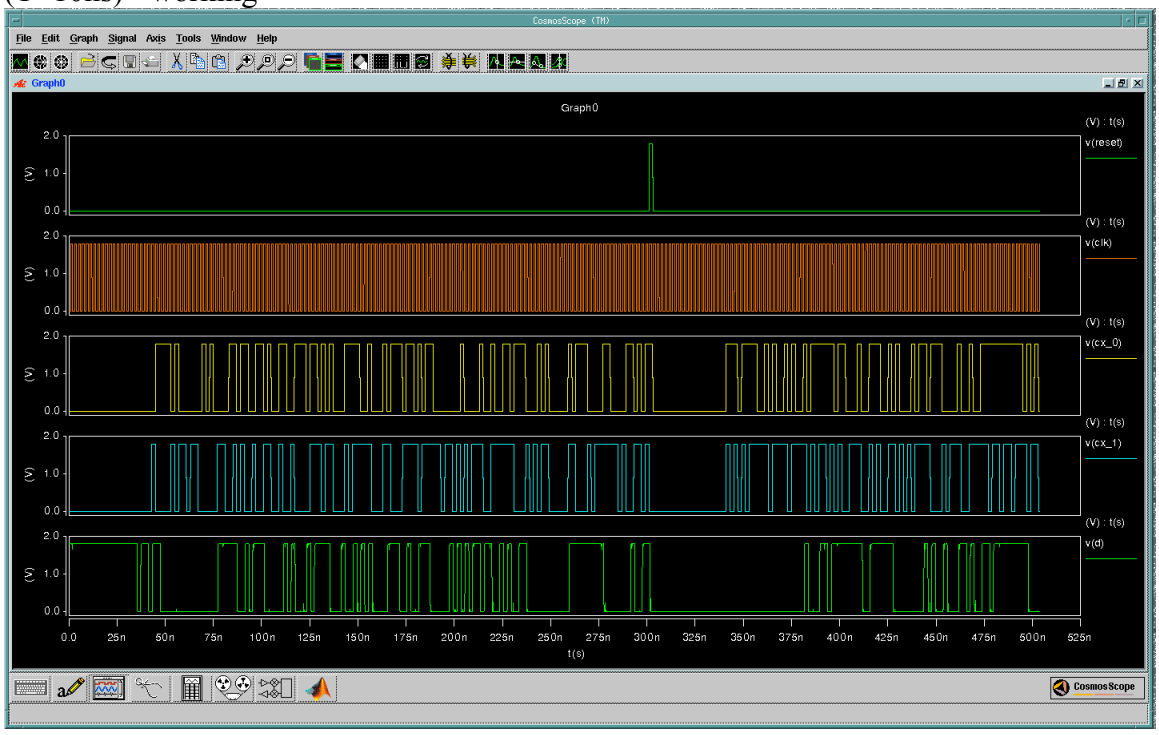
As you can see, the results match the functional test results when using the second data pattern.

EE577b Viterbi Decoder Project

by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com



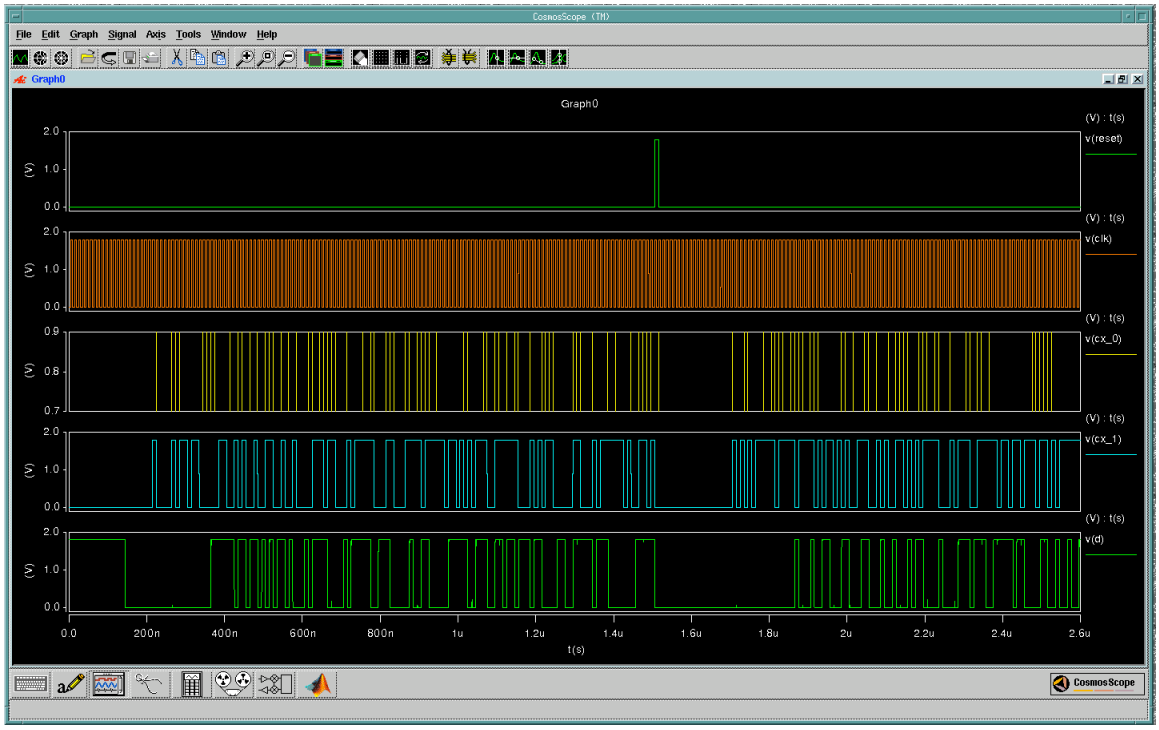
(T=10ns) - working



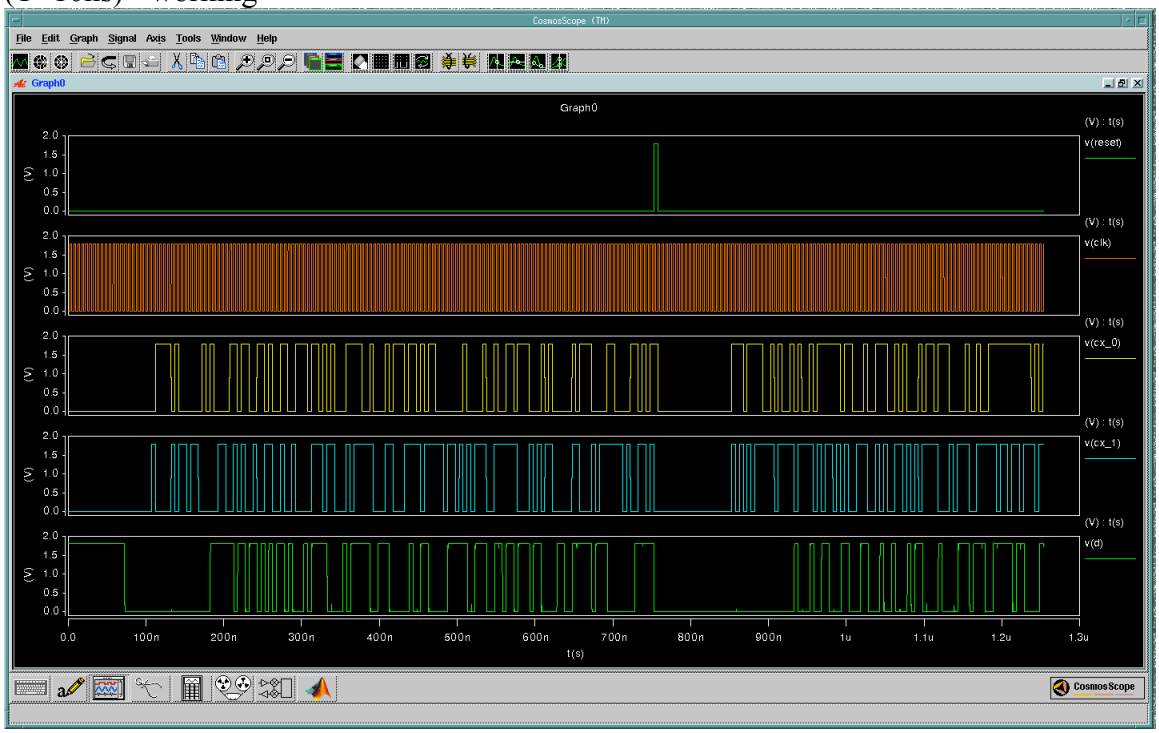
(T=2ns) – Not working

EE577b Viterbi Decoder Project

by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com



(T=10ns) - working



(T=5ns) – working

Therefore, the fastest period of clock is 5ns.

by Zhyang Ong zhiyang@ieee.org and Andrew Mattheisen amattheisen@gmail.com

4 Showtime Results

Unfortunately we were unable to complete the showtime part of this assignment.

5 Post Synthesis Results

The authors were unable to perform post-synthesis simulation as they were unable to compile their Verilog testbenches with the synthesized netlist. This is due to the inappropriate inclusion of the SDF file.

6 Make File Generation

The description for the make file is provided in the comments of the Make file. Each make target is used to run a particular test bench.

We attempted to create the make file and had it working for awhile, but it broke today and we have not been able to fix it - sorry.