Dennis Lang

ECE 111 Winter 2024

Professor John Eldon

23 March 2024

Final Project

Part 1 - Priority Encoder with Viterbi Decoder Implementation

1)

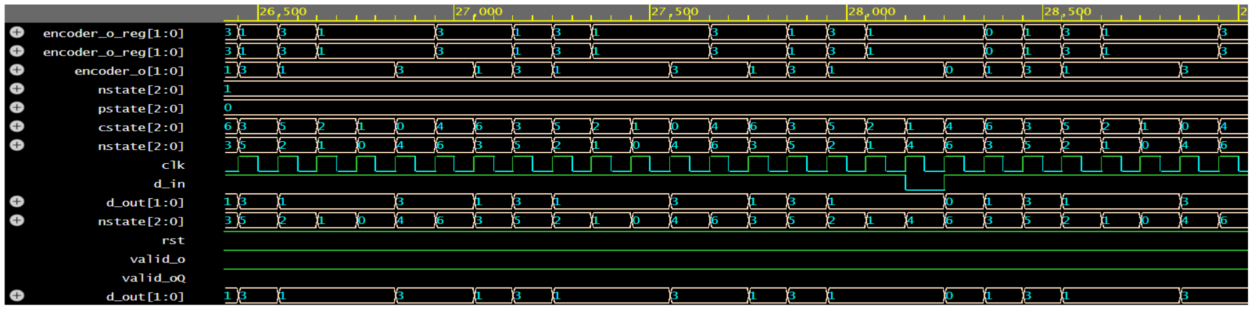
A. Working SV Code (see folder)

B. Questa/ModelSim Transcript (see below)

C. Proof of synthesis - Mentor Precision Netlist and Utilization Report (see "precision.log")

2) Explain how this encoder works, since it differs somewhat from the Homework 7 unit, although similar in principle. Use this encoder with the decoder – your Homework 7 encoder will provide decoder input vectors that won't work with this design.

The use of states in the project encoder is what differentiates it from the homework encoder. Using a state machine "cstate", the encoder uses three bits for encoding the input signals into output of 2-bits. The homework uses a convolutional encoder with taps. The *priority encoder* in the project is used for compression of binary inputs into a smaller number of outputs. With each input assigned a unique priority level, the encoder is able to determine using this number which to select first, converting the index to binary output. From the EPWave waveform viewer, we can see cstate's state machine behavior and priority handling.



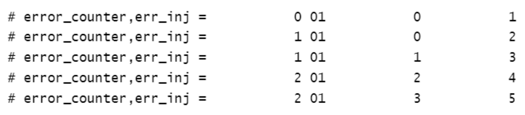
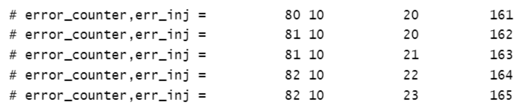
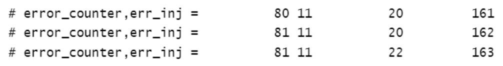
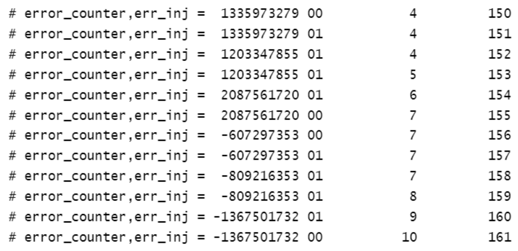
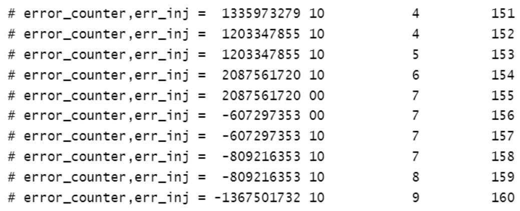
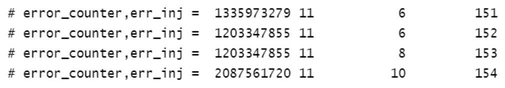
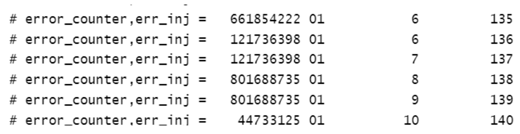
3) Explain how the decoder works and show that the minimum branch metric = 0 when no errors are present. How does the decoder determine what sequence of bits was sent to the encoder?

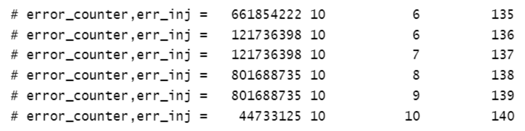
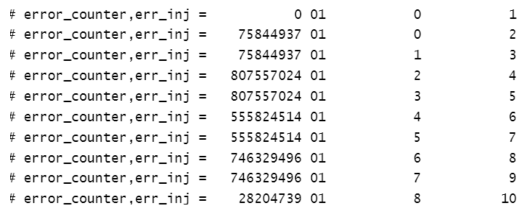
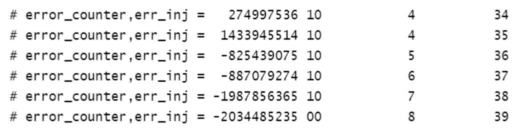
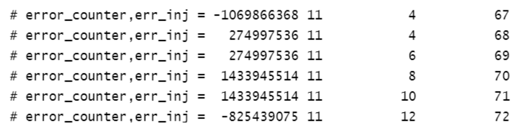
The decoder used in the project is a Viterbi decoder, which uses 8 instantiated Branch Metric Calculation modules to represent the trellis in the diagram. The Add-Compare-Select operation is performed, to assign the path for the correct sequence (first or second, then compare and select lower metric). There are memory modules for storing the path metric as seen in the RTL diagram. The traceback unit reconstructs the original data based on the traceback algorithm, after which, the decoded output is stored in d_out. When no errors are present, the branch metric is used as an indicator for the correct signal being transmitted, and it returns 0 for the number of errors. The decoder traces back through the path with the lowest value for this variable in order to recover the sequence. This algorithm relies on the fact that errors are highly unlikely to occur, giving the path back a lower number of invalid sequences. Finding the shortest path through the trellis is the goal of the Viterbi decoder.

```
[2024-03-22 20:13:06 UTC] qrun -autoorder -batch -access=rw+/.  -vcom.options
'-timescale' '1ns/1ns'  -end -mfcu design.sv testbench.sv  -vsim.options
'-voptargs=+acc=npr'  -end  -do  " run -all; exit"
QuestaSim-64 qrun 2021.3 Utility 2021.07 Jul 13 2021
Start time: 16:13:06 on Mar 22,2024
qrun -autoorder -batch -access=rw+/. -vcom.options -timescale 1ns/1ns -end -mfcu
design.sv testbench.sv -vsim.options -voptargs="+acc=npr" -end -do " run -all; exit"
Creating library 'qrun.out/work'.
QuestaSim-64 vlog 2021.3 Compiler 2021.07 Jul 13 2021
Start time: 16:13:06 on Mar 22,2024
vlog -mfcu design.sv testbench.sv -work qrun.out/work -csession=incr -writesessionid
"+qrun.out/top_dus" -statslog qrun.out/stats_log
-- Compiling module ACS
-- Compiling module bmc000
-- Compiling module bmc001
-- Compiling module bmc010
-- Compiling module bmc011
-- Compiling module bmc100
-- Compiling module bmc101
-- Compiling module bmc110
-- Compiling module bmc111
-- Compiling module decoder
-- Compiling module encoder
-- Compiling module mem_disp
-- Compiling module mem
-- Compiling module tbu
-- Compiling module viterbi_tx_rx
-- Compiling module viterbi_tx_rx_tb

Top level modules:
        viterbi_tx_rx_tb
End time: 16:13:07 on Mar 22,2024, Elapsed time: 0:00:01
Errors: 0, Warnings: 0
QuestaSim-64 vopt 2021.3 Compiler 2021.07 Jul 13 2021
Start time: 16:13:07 on Mar 22,2024
vopt -access=rw+/. -mfcu -findtoplevels /home/runner/qrun.out/work+0+ -work
qrun.out/work -statslog qrun.out/stats_log -o qrun_opt

Top level modules:
        viterbi_tx_rx_tb

Analyzing design...
-- Loading module viterbi_tx_rx_tb
-- Loading module viterbi_tx_rx
-- Loading module encoder
-- Loading module decoder
-- Loading module bmc000
-- Loading module bmc001
-- Loading module bmc010
-- Loading module bmc011
-- Loading module bmc100
-- Loading module bmc101
-- Loading module bmc110
-- Loading module bmc111
-- Loading module ACS
-- Loading module mem
-- Loading module tbu
-- Loading module mem_disp
Optimizing 16 design-units (inlining 27/28 module instances):
-- Inlining module encoder(fast)
-- Inlining module bmc000(fast)
-- Inlining module bmc001(fast)
-- Inlining module bmc010(fast)
-- Inlining module bmc011(fast)
-- Inlining module bmc100(fast)
-- Inlining module bmc101(fast)
-- Inlining module bmc110(fast)
-- Inlining module bmc111(fast)
-- Inlining module ACS(fast)
-- Inlining module mem(fast)
-- Inlining module tbu(fast)
-- Inlining module mem_disp(fast)
-- Inlining module decoder(fast)
-- Inlining module viterbi_tx_rx(fast)
-- Optimizing module viterbi_tx_rx_tb(fast)
Optimized design name is qrun_opt
End time: 16:13:07 on Mar 22,2024, Elapsed time: 0:00:00
Errors: 0, Warnings: 0
# vsim -batch -voptargs="+acc=npr" -lib qrun.out/work -do " run -all; exit" -statslog
qrun.out/stats_log qrun_opt -appendlog -l qrun.log
# Start time: 16:13:07 on Mar 22,2024
# //  Questa Sim-64
# //  Version 2021.3 linux_x86_64 Jul 13 2021
# //
# //  Copyright 1991-2021 Mentor Graphics Corporation
# //  All Rights Reserved.
# //
# //  QuestaSim and its associated documentation contain trade
# //  secrets and commercial or financial information that are the property of
# //  Mentor Graphics Corporation and are privileged, confidential,
# //  and exempt from disclosure under the Freedom of Information Act,
# //  5 U.S.C. Section 552. Furthermore, this information
# //  is prohibited from disclosure under the Trade Secrets Act,
# //  18 U.S.C. Section 1905.
# //
# Loading sv_std.std
# Loading work.viterbi_tx_rx_tb(fast)
#
# run -all
# yaa! in = 0, out = 0, w_ct =           0, err = 00
# yaa! in = 0, out = 0, w_ct =           1, err = 00
# yaa! in = 0, out = 0, w_ct =           2, err = 00
# yaa! in = 0, out = 0, w_ct =           3, err = 00
# yaa! in = 0, out = 0, w_ct =           4, err = 00
# yaa! in = 0, out = 0, w_ct =           5, err = 00
# yaa! in = 0, out = 0, w_ct =           6, err = 00
# yaa! in = 0, out = 0, w_ct =           7, err = 00
# yaa! in = 0, out = 0, w_ct =           8, err = 00
# yaa! in = 0, out = 0, w_ct =           9, err = 00
# yaa! in = 0, out = 0, w_ct =          10, err = 00
# yaa! in = 1, out = 1, w_ct =          11, err = 00
# yaa! in = 0, out = 0, w_ct =          12, err = 00
# yaa! in = 0, out = 0, w_ct =          13, err = 00
# yaa! in = 1, out = 1, w_ct =          14, err = 00
# yaa! in = 1, out = 1, w_ct =          15, err = 00
# yaa! in = 0, out = 0, w_ct =          16, err = 00
# yaa! in = 0, out = 0, w_ct =          17, err = 00
# yaa! in = 0, out = 0, w_ct =          18, err = 00
# yaa! in = 1, out = 1, w_ct =          19, err = 00
# yaa! in = 1, out = 1, w_ct =          20, err = 00
# yaa! in = 1, out = 1, w_ct =          21, err = 00
# yaa! in = 0, out = 0, w_ct =          22, err = 00
# yaa! in = 0, out = 0, w_ct =          23, err = 00
# yaa! in = 0, out = 0, w_ct =          24, err = 00
# yaa! in = 0, out = 0, w_ct =          25, err = 00
# yaa! in = 1, out = 1, w_ct =          26, err = 00
# yaa! in = 1, out = 1, w_ct =          27, err = 00
# yaa! in = 1, out = 1, w_ct =          28, err = 00
# yaa! in = 1, out = 1, w_ct =          29, err = 00
# yaa! in = 0, out = 0, w_ct =          30, err = 00
# yaa! in = 0, out = 0, w_ct =          31, err = 00
# yaa! in = 0, out = 0, w_ct =          32, err = 00
# yaa! in = 0, out = 0, w_ct =          33, err = 00
# yaa! in = 0, out = 0, w_ct =          34, err = 00
# yaa! in = 1, out = 1, w_ct =          35, err = 00
# yaa! in = 1, out = 1, w_ct =          36, err = 00
# yaa! in = 1, out = 1, w_ct =          37, err = 00
# yaa! in = 1, out = 1, w_ct =          38, err = 00
# yaa! in = 1, out = 1, w_ct =          39, err = 00
# yaa! in = 0, out = 0, w_ct =          40, err = 00
# yaa! in = 1, out = 1, w_ct =          41, err = 00
# yaa! in = 0, out = 0, w_ct =          42, err = 00
# yaa! in = 0, out = 0, w_ct =          43, err = 00
# yaa! in = 1, out = 1, w_ct =          44, err = 00
# yaa! in = 1, out = 1, w_ct =          45, err = 00
# yaa! in = 0, out = 0, w_ct =          46, err = 00
# yaa! in = 0, out = 0, w_ct =          47, err = 00
# yaa! in = 0, out = 0, w_ct =          48, err = 00
# yaa! in = 1, out = 1, w_ct =          49, err = 00
# yaa! in = 1, out = 1, w_ct =          50, err = 00
# yaa! in = 1, out = 1, w_ct =          51, err = 00
# yaa! in = 0, out = 0, w_ct =          52, err = 00
# yaa! in = 0, out = 0, w_ct =          53, err = 00
# yaa! in = 0, out = 0, w_ct =          54, err = 00
# yaa! in = 0, out = 0, w_ct =          55, err = 00
# yaa! in = 1, out = 1, w_ct =          56, err = 00
# yaa! in = 1, out = 1, w_ct =          57, err = 00
# yaa! in = 1, out = 1, w_ct =          58, err = 00
# yaa! in = 1, out = 1, w_ct =          59, err = 00
# yaa! in = 0, out = 0, w_ct =          60, err = 00
# yaa! in = 0, out = 0, w_ct =          61, err = 00
# yaa! in = 0, out = 0, w_ct =          62, err = 00
# yaa! in = 0, out = 0, w_ct =          63, err = 00
# yaa! in = 0, out = 0, w_ct =          64, err = 00
# yaa! in = 1, out = 1, w_ct =          65, err = 00
# yaa! in = 1, out = 1, w_ct =          66, err = 00
# yaa! in = 1, out = 1, w_ct =          67, err = 00
# yaa! in = 1, out = 1, w_ct =          68, err = 00
# yaa! in = 1, out = 1, w_ct =          69, err = 00
# yaa! in = 1, out = 1, w_ct =          70, err = 00
# yaa! in = 1, out = 1, w_ct =          71, err = 00
# yaa! in = 1, out = 1, w_ct =          72, err = 00
# yaa! in = 1, out = 1, w_ct =          73, err = 00
# yaa! in = 1, out = 1, w_ct =          74, err = 00
# yaa! in = 1, out = 1, w_ct =          75, err = 00
# yaa! in = 1, out = 1, w_ct =          76, err = 00
# yaa! in = 1, out = 1, w_ct =          77, err = 00
# yaa! in = 1, out = 1, w_ct =          78, err = 00
# yaa! in = 1, out = 1, w_ct =          79, err = 00
# yaa! in = 1, out = 1, w_ct =          80, err = 00
# yaa! in = 1, out = 1, w_ct =          81, err = 00
# yaa! in = 1, out = 1, w_ct =          82, err = 00
# yaa! in = 1, out = 1, w_ct =          83, err = 00
# yaa! in = 1, out = 1, w_ct =          84, err = 00
# yaa! in = 1, out = 1, w_ct =          85, err = 00
# yaa! in = 1, out = 1, w_ct =          86, err = 00
# yaa! in = 1, out = 1, w_ct =          87, err = 00
# yaa! in = 1, out = 1, w_ct =          88, err = 00
# yaa! in = 1, out = 1, w_ct =          89, err = 00
# yaa! in = 1, out = 1, w_ct =          90, err = 00
# yaa! in = 1, out = 1, w_ct =          91, err = 00
# yaa! in = 1, out = 1, w_ct =          92, err = 00
# yaa! in = 1, out = 1, w_ct =          93, err = 00
# yaa! in = 1, out = 1, w_ct =          94, err = 00
# yaa! in = 1, out = 1, w_ct =          95, err = 00
# yaa! in = 1, out = 1, w_ct =          96, err = 00
# yaa! in = 1, out = 1, w_ct =          97, err = 00
# yaa! in = 1, out = 1, w_ct =          98, err = 00
# yaa! in = 1, out = 1, w_ct =          99, err = 00
# yaa! in = 1, out = 1, w_ct =         100, err = 00
# yaa! in = 1, out = 1, w_ct =         101, err = 00
# yaa! in = 1, out = 1, w_ct =         102, err = 00
# yaa! in = 1, out = 1, w_ct =         103, err = 00
# yaa! in = 1, out = 1, w_ct =         104, err = 00
# yaa! in = 1, out = 1, w_ct =         105, err = 00
# yaa! in = 1, out = 1, w_ct =         106, err = 00
# yaa! in = 1, out = 1, w_ct =         107, err = 00
# yaa! in = 1, out = 1, w_ct =         108, err = 00
# yaa! in = 1, out = 1, w_ct =         109, err = 00
# yaa! in = 1, out = 1, w_ct =         110, err = 00
# yaa! in = 1, out = 1, w_ct =         111, err = 00
# yaa! in = 1, out = 1, w_ct =         112, err = 00
# yaa! in = 1, out = 1, w_ct =         113, err = 00
# yaa! in = 1, out = 1, w_ct =         114, err = 00
# yaa! in = 1, out = 1, w_ct =         115, err = 00
# yaa! in = 1, out = 1, w_ct =         116, err = 00
# yaa! in = 1, out = 1, w_ct =         117, err = 00
# yaa! in = 1, out = 1, w_ct =         118, err = 00
# yaa! in = 1, out = 1, w_ct =         119, err = 00
# yaa! in = 1, out = 1, w_ct =         120, err = 00
# yaa! in = 1, out = 1, w_ct =         121, err = 00
# yaa! in = 1, out = 1, w_ct =         122, err = 00
# yaa! in = 1, out = 1, w_ct =         123, err = 00
# yaa! in = 1, out = 1, w_ct =         124, err = 00
# yaa! in = 1, out = 1, w_ct =         125, err = 00
# yaa! in = 1, out = 1, w_ct =         126, err = 00
# yaa! in = 1, out = 1, w_ct =         127, err = 00
# yaa! in = 1, out = 1, w_ct =         128, err = 00
# yaa! in = 1, out = 1, w_ct =         129, err = 00
# yaa! in = 1, out = 1, w_ct =         130, err = 00
# yaa! in = 1, out = 1, w_ct =         131, err = 00
# yaa! in = 1, out = 1, w_ct =         132, err = 00
# yaa! in = 1, out = 1, w_ct =         133, err = 00
# yaa! in = 1, out = 1, w_ct =         134, err = 00
# yaa! in = 1, out = 1, w_ct =         135, err = 00
# yaa! in = 1, out = 1, w_ct =         136, err = 00
# yaa! in = 1, out = 1, w_ct =         137, err = 00
# yaa! in = 1, out = 1, w_ct =         138, err = 00
# yaa! in = 1, out = 1, w_ct =         139, err = 00
# yaa! in = 1, out = 1, w_ct =         140, err = 00
# yaa! in = 1, out = 1, w_ct =         141, err = 00
# yaa! in = 1, out = 1, w_ct =         142, err = 00
# yaa! in = 1, out = 1, w_ct =         143, err = 00
# yaa! in = 1, out = 1, w_ct =         144, err = 00
# yaa! in = 1, out = 1, w_ct =         145, err = 00
# yaa! in = 1, out = 1, w_ct =         146, err = 00
# yaa! in = 1, out = 1, w_ct =         147, err = 00
# yaa! in = 1, out = 1, w_ct =         148, err = 00
# yaa! in = 1, out = 1, w_ct =         149, err = 00
# yaa! in = 1, out = 1, w_ct =         150, err = 00
# yaa! in = 1, out = 1, w_ct =         151, err = 00
# yaa! in = 1, out = 1, w_ct =         152, err = 00
# yaa! in = 1, out = 1, w_ct =         153, err = 00
# yaa! in = 1, out = 1, w_ct =         154, err = 00
# yaa! in = 1, out = 1, w_ct =         155, err = 00
# yaa! in = 1, out = 1, w_ct =         156, err = 00
# yaa! in = 1, out = 1, w_ct =         157, err = 00
# yaa! in = 1, out = 1, w_ct =         158, err = 00
# yaa! in = 1, out = 1, w_ct =         159, err = 00
# yaa! in = 1, out = 1, w_ct =         160, err = 00
# yaa! in = 1, out = 1, w_ct =         161, err = 00
# yaa! in = 1, out = 1, w_ct =         162, err = 00
# yaa! in = 1, out = 1, w_ct =         163, err = 00
# yaa! in = 1, out = 1, w_ct =         164, err = 00
# yaa! in = 1, out = 1, w_ct =         165, err = 00
# yaa! in = 1, out = 1, w_ct =         166, err = 00
# yaa! in = 1, out = 1, w_ct =         167, err = 00
# yaa! in = 1, out = 1, w_ct =         168, err = 00
# yaa! in = 0, out = 0, w_ct =         169, err = 00
# yaa! in = 1, out = 1, w_ct =         170, err = 00
# yaa! in = 1, out = 1, w_ct =         171, err = 00
# yaa! in = 1, out = 1, w_ct =         172, err = 00
# yaa! in = 1, out = 1, w_ct =         173, err = 00
# yaa! in = 1, out = 1, w_ct =         174, err = 00
# yaa! in = 1, out = 1, w_ct =         175, err = 00
# yaa! in = 1, out = 1, w_ct =         176, err = 00
# yaa! in = 1, out = 1, w_ct =         177, err = 00
# yaa! in = 1, out = 1, w_ct =         178, err = 00
# yaa! in = 1, out = 1, w_ct =         179, err = 00
# yaa! in = 0, out = 0, w_ct =         180, err = 00
# yaa! in = 1, out = 1, w_ct =         181, err = 00
# yaa! in = 1, out = 1, w_ct =         182, err = 00
# yaa! in = 1, out = 1, w_ct =         183, err = 00
# yaa! in = 1, out = 1, w_ct =         184, err = 00
# yaa! in = 1, out = 1, w_ct =         185, err = 00
# yaa! in = 1, out = 1, w_ct =         186, err = 00
# yaa! in = 1, out = 1, w_ct =         187, err = 00
# yaa! in = 1, out = 1, w_ct =         188, err = 00
# yaa! in = 1, out = 1, w_ct =         189, err = 00
# yaa! in = 1, out = 1, w_ct =         190, err = 00
# yaa! in = 1, out = 1, w_ct =         191, err = 00
# yaa! in = 1, out = 1, w_ct =         192, err = 00
# yaa! in = 1, out = 1, w_ct =         193, err = 00
# yaa! in = 1, out = 1, w_ct =         194, err = 00
# yaa! in = 1, out = 1, w_ct =         195, err = 00
# yaa! in = 1, out = 1, w_ct =         196, err = 00
# yaa! in = 1, out = 1, w_ct =         197, err = 00
# yaa! in = 1, out = 1, w_ct =         198, err = 00
# yaa! in = 1, out = 1, w_ct =         199, err = 00
# yaa! in = 1, out = 1, w_ct =         200, err = 00
# yaa! in = 1, out = 1, w_ct =         201, err = 00
# yaa! in = 1, out = 1, w_ct =         202, err = 00
# yaa! in = 1, out = 1, w_ct =         203, err = 00
# yaa! in = 1, out = 1, w_ct =         204, err = 00
# yaa! in = 1, out = 1, w_ct =         205, err = 00
# yaa! in = 1, out = 1, w_ct =         206, err = 00
# yaa! in = 1, out = 1, w_ct =         207, err = 00
# yaa! in = 1, out = 1, w_ct =         208, err = 00
# yaa! in = 1, out = 1, w_ct =         209, err = 00
# yaa! in = 1, out = 1, w_ct =         210, err = 00
# yaa! in = 1, out = 1, w_ct =         211, err = 00
# yaa! in = 1, out = 1, w_ct =         212, err = 00
# yaa! in = 1, out = 1, w_ct =         213, err = 00
# yaa! in = 1, out = 1, w_ct =         214, err = 00
# yaa! in = 1, out = 1, w_ct =         215, err = 00
# yaa! in = 1, out = 1, w_ct =         216, err = 00
# yaa! in = 1, out = 1, w_ct =         217, err = 00
# yaa! in = 1, out = 1, w_ct =         218, err = 00
# yaa! in = 1, out = 1, w_ct =         219, err = 00
# yaa! in = 1, out = 1, w_ct =         220, err = 00
# yaa! in = 1, out = 1, w_ct =         221, err = 00
# yaa! in = 1, out = 1, w_ct =         222, err = 00
# yaa! in = 1, out = 1, w_ct =         223, err = 00
# yaa! in = 1, out = 1, w_ct =         224, err = 00
# yaa! in = 1, out = 1, w_ct =         225, err = 00
# yaa! in = 1, out = 1, w_ct =         226, err = 00
# yaa! in = 1, out = 1, w_ct =         227, err = 00
# yaa! in = 1, out = 1, w_ct =         228, err = 00
# yaa! in = 1, out = 1, w_ct =         229, err = 00
# yaa! in = 1, out = 1, w_ct =         230, err = 00
# yaa! in = 1, out = 1, w_ct =         231, err = 00
# yaa! in = 1, out = 1, w_ct =         232, err = 00
# yaa! in = 1, out = 1, w_ct =         233, err = 00
# yaa! in = 1, out = 1, w_ct =         234, err = 00
# yaa! in = 1, out = 1, w_ct =         235, err = 00
# yaa! in = 1, out = 1, w_ct =         236, err = 00
# yaa! in = 1, out = 1, w_ct =         237, err = 00
# yaa! in = 1, out = 1, w_ct =         238, err = 00
# yaa! in = 1, out = 1, w_ct =         239, err = 00
# yaa! in = 1, out = 1, w_ct =         240, err = 00
# yaa! in = 1, out = 1, w_ct =         241, err = 00
# yaa! in = 1, out = 1, w_ct =         242, err = 00
# yaa! in = 1, out = 1, w_ct =         243, err = 00
# yaa! in = 1, out = 1, w_ct =         244, err = 00
# yaa! in = 1, out = 1, w_ct =         245, err = 00
# yaa! in = 1, out = 1, w_ct =         246, err = 00
# yaa! in = 1, out = 1, w_ct =         247, err = 00
# yaa! in = 1, out = 1, w_ct =         248, err = 00
# yaa! in = 1, out = 1, w_ct =         249, err = 00
# yaa! in = 1, out = 1, w_ct =         250, err = 00
# yaa! in = 1, out = 1, w_ct =         251, err = 00
# yaa! in = 1, out = 1, w_ct =         252, err = 00
# yaa! in = 1, out = 1, w_ct =         253, err = 00
# yaa! in = 1, out = 1, w_ct =         254, err = 00
# yaa! in = 1, out = 1, w_ct =         255, err = 00
# good =         256, bad =          0,           0 bad_bits
# ** Note: $stop    : viterbi_tx_rx_tb.sv(372)
#    Time: 1381700 ns  Iteration: 0  Instance: /viterbi_tx_rx_tb
# Break in Module viterbi_tx_rx_tb at viterbi_tx_rx_tb.sv line 372
# exit
# End time: 16:13:08 on Mar 22,2024, Elapsed time: 0:00:01
# Errors: 0, Warnings: 0
# *** Summary *********************************************
#     qrun: Errors:   0, Warnings:   0
#     vlog: Errors:   0, Warnings:   0
#     vopt: Errors:   0, Warnings:   0
#     vsim: Errors:   0, Warnings:   0
#   Totals: Errors:   0, Warnings:   0
Done
```

Part 2 - Priority Encoder with Viterbi Decoder Debugging (Table)

Filename	Description	Number of Errors Injected	Pattern of Errors Injected	Number of Errors Uncorrected
viterbi…**2a1**.sv	Invert bit[0] of the channel once every 8 samples	15	01	0
viterbi…**2a2**.sv	Invert bit[1] of the channel once every 8 samples	15	10	0
viterbi…**2a3**.sv	Invert bit[0] and bit[1] of the channel once every 16 samples	14	11	0
viterbi…**2a4**.sv	Invert bit[0] of the channel twice in a row every 16 samples	23	0101	0
viterbi…**2a5**.sv	Invert bit[1] of the channel twice in a row every 16 samples	23	1010	0
viterbi…**2a6**.sv	Invert bit[0] four times in a row out of every 32 samples	39	01010101	32
viterbi…**2a7**.sv	Invert bit[1] four times in a row out of every 32 samples	32	10101010	15
viterbi…**2a8**.sv	Invert bits 1 and 0 twice in a row out of every 32 samples	35	1111	21
viterbi…**2b1**.sv	(Random) Invert bit[0] of the channel once every 8 samples	18	01	6
viterbi…**2b2**.sv	(Random) Invert bit[1] of the channel once every 8 samples	18	10	2
viterbi…**2b3**.sv	(Random) Invert bit[0] and bit[1] of the channel once every 16 samples	24	11	7
viterbi…**2b4**.sv	(Random) Invert bit[0] of the channel twice in a row every 16 samples	18	0101	7
viterbi…**2b5**.sv	(Random) Invert bit[1] of the channel twice in a row every 16 samples	18	1010	2
viterbi…**2b6**.sv	(Random) Invert bit[0] four times in a row out of every 32 samples	16	01010101	16
viterbi…**2b7**.sv	(Random) Invert bit[1] four times in a row out of every 32 samples	16	10101010	16
viterbi…**2b8**.sv	(Random) Invert bits 1 and 0 twice in a row out of every 32 samples	32	1111	21

Filename	Screenshot (Error Causing Behavior)	Biggest Cluster	Errors
viterbi…**2a1**.sv		01	0
viterbi…**2a2**.sv		10	0
viterbi…**2a3**.sv		11	0
viterbi…**2a4**.sv		0101	0
viterbi…**2a5**.sv		1010	0
viterbi…**2a6**.sv	 # error_counter,err_inj = 0 01 0 1 # error_counter,err_inj = 1 01 0 2 # error_counter,err_inj = 1 01 1 3 # error_counter,err_inj = 2 01 2 4 # error_counter,err_inj = 2 01 3 5	01010101	32
viterbi…**2a7**.sv	 # error_counter,err_inj = 80 10 20 161 # error_counter,err_inj = 81 10 20 162 # error_counter,err_inj = 81 10 21 163 # error_counter,err_inj = 82 10 22 164 # error_counter,err_inj = 82 10 23 165	10101010	15
viterbi…**2a8**.sv	 # error_counter,err_inj = 80 11 20 161 # error_counter,err_inj = 81 11 20 162 # error_counter,err_inj = 81 11 22 163	1111(2E)	21
viterbi…**2b1**.sv	 # error_counter,err_inj = 1335973279 00 4 150 # error_counter,err_inj = 1335973279 01 4 151 # error_counter,err_inj = 1203347855 01 4 152 # error_counter,err_inj = 1203347855 01 5 153 # error_counter,err_inj = 2087561720 01 6 154 # error_counter,err_inj = 2087561720 00 7 155 # error_counter,err_inj = -607297353 00 7 156 # error_counter,err_inj = -607297353 01 7 157 # error_counter,err_inj = -809216353 01 7 158 # error_counter,err_inj = -809216353 01 8 159 # error_counter,err_inj = -1367501732 01 9 160 # error_counter,err_inj = -1367501732 00 10 161	010101 (2C)	2x3
viterbi…**2b2**.sv	 # error_counter,err_inj = 1335973279 10 4 151 # error_counter,err_inj = 1203347855 10 4 152 # error_counter,err_inj = 1203347855 10 5 153 # error_counter,err_inj = 2087561720 10 6 154 # error_counter,err_inj = 2087561720 00 7 155 # error_counter,err_inj = -607297353 00 7 156 # error_counter,err_inj = -607297353 10 7 157 # error_counter,err_inj = -809216353 10 7 158 # error_counter,err_inj = -809216353 10 8 159 # error_counter,err_inj = -1367501732 10 9 160	101010 (2D)	2x1
viterbi…**2b3**.sv	 # error_counter,err_inj = 1335973279 11 6 151 # error_counter,err_inj = 1203347855 11 6 152 # error_counter,err_inj = 1203347855 11 8 153 # error_counter,err_inj = 2087561720 11 10 154	111111	7
viterbi…**2b4**.sv	 # error_counter,err_inj = 661854222 01 6 135 # error_counter,err_inj = 121736398 01 6 136 # error_counter,err_inj = 121736398 01 7 137 # error_counter,err_inj = 801688735 01 8 138 # error_counter,err_inj = 801688735 01 9 139 # error_counter,err_inj = 44733125 01 10 140	01010101	7

viterbi…**2b5**.sv	# error_counter,err_inj = 661854222 10 6 135 # error_counter,err_inj = 121736398 10 6 136 # error_counter,err_inj = 121736398 10 7 137 # error_counter,err_inj = 801688735 10 8 138 # error_counter,err_inj = 801688735 10 9 139 # error_counter,err_inj = 44733125 10 10 140	10101010	2
viterbi…**2b6**.sv	# error_counter,err_inj = 0 01 0 1 # error_counter,err_inj = 75844937 01 0 2 # error_counter,err_inj = 75844937 01 1 3 # error_counter,err_inj = 807557024 01 2 4 # error_counter,err_inj = 807557024 01 3 5 # error_counter,err_inj = 555824514 01 4 6 # error_counter,err_inj = 555824514 01 5 7 # error_counter,err_inj = 746329496 01 6 8 # error_counter,err_inj = 746329496 01 7 9 # error_counter,err_inj = 28204739 01 8 10	01010100	19
viterbi…**2b7**.sv	# error_counter,err_inj = 274997536 10 4 34 # error_counter,err_inj = 1433945514 10 4 35 # error_counter,err_inj = -825439075 10 5 36 # error_counter,err_inj = -887079274 10 6 37 # error_counter,err_inj = -1987856365 10 7 38 # error_counter,err_inj = -2034485235 00 8 39	10101010	16
viterbi…**2b8**.sv	# error_counter,err_inj = -1069866368 11 4 67 # error_counter,err_inj = 274997536 11 4 68 # error_counter,err_inj = 274997536 11 6 69 # error_counter,err_inj = 1433945514 11 8 70 # error_counter,err_inj = 1433945514 11 10 71 # error_counter,err_inj = -825439075 11 12 72	11111111	21

2C) From the table, with an injected string of 3+ consecutive bad bit[0] we can see that there begins to produce output errors.

2D) From the table, with an injected string of 3+ consecutive bad bit[1] we can see that there begins to produce output errors.

2E) From the table, with an injected string of 2+ consecutive bad bit[0] and bad bit[1] we can see that there begins to produce output errors.

This proves the clustering of injected errors breaks the Forward Error Correction protocol between the priority encoder and Viterbi decoder.