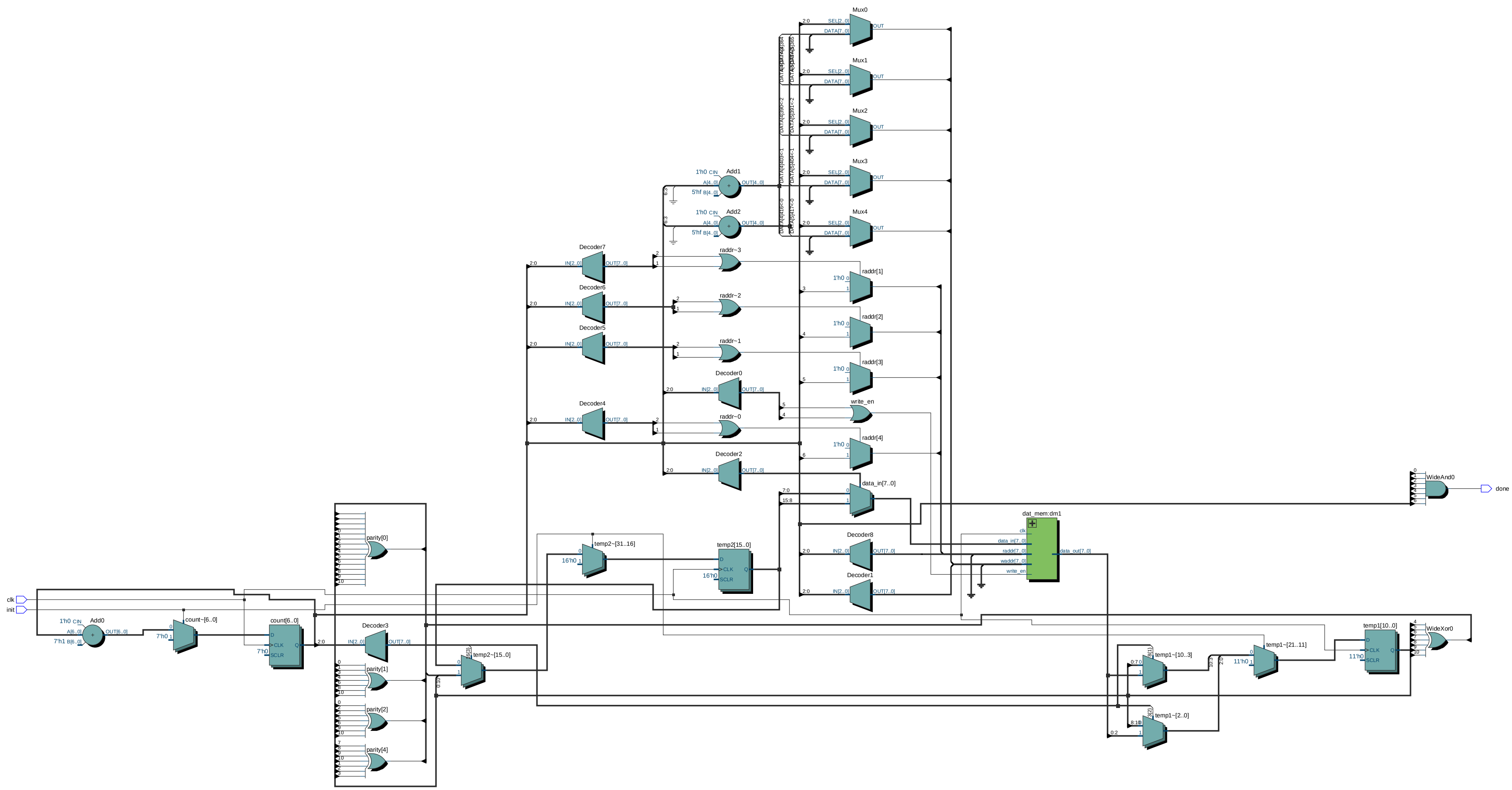
What worked / didn't work.

To start off with, the program worked as intended with given test cases.

Debug process. First, the walking 1 bit debug 'method' was used to systematically take care of incorrect bit rearrangements, one at a time. After solving all cases with walking 1's, the original testbench which applied $random to our test cases was used. Next, $random w/ shifting was also tested, and after multiple runs of the program (for each testbench edit), it was confirmed that the program continued to work as intended; producing a match between the expected and computed encoded messages.



clk
init
done
Add0
Add1
Add2
count~[6..0]
count[6..0]
Decoder0
Decoder1
Decoder2
Decoder3
Decoder4
Decoder5
Decoder6
Decoder7
Decoder8
parity[0]
parity[1]
parity[2]
parity[4]
temp2~[15..0]
temp2~[31..16]
temp2[15..0]
Mux0
Mux1
Mux2
Mux3
Mux4
raddr~0
raddr~1
raddr~2
raddr~3
raddr[1]
raddr[2]
raddr[3]
raddr[4]
write_en
data_in[7..0]
dat_mem:dm1
WideAnd0
WideXor0
temp1~[10..3]
temp1~[2..0]
temp1~[21..11]
temp1[10..0]

```
# Compile of dat_mem.sv was successful.
# Compile of lab4_tb.sv was successful.
# Compile of top_level.sv was successful.
# 3 compiles, 0 failed with no errors.
vsim -gui work.lab4_tb
# End time: 20:08:02 on Dec 10,2021, Elapsed time: 0:00:31
# Errors: 0, Warnings: 5
# vsim -gui work.lab4_tb
# Start time: 20:08:02 on Dec 10,2021
# Loading sv_std.std
# Loading work.lab4_tb
# Loading work.top_level
# Loading work.dat_mem
# WARNING: No extended dataflow license exists
add wave -position insertpoint sim:/lab4_tb/DUT/*
restart
run -all
# start lab4
#
# 10100100100  Message
# 1010010101010101  Expected Encoding
# 1010010101010101  Your Encoding
#
# 11010000001  Message
# 1101000100011101  Expected Encoding
# 1101000100011101  Your Encoding
#
# 11000001001  Message
# 1100000010011010  Expected Encoding
# 1100000010011010  Your Encoding
#
# 11001100011  Message
# 1100110000111100  Expected Encoding
# 1100110000111100  Your Encoding
#
# 01100001101  Message
# 0110000011001010  Expected Encoding
# 0110000011001010  Your Encoding
#
# 00110001101  Message
# 0011000011001111  Expected Encoding
# 0011000011001111  Your Encoding
#
# 10001100101  Message
# 1000110101001110  Expected Encoding
# 1000110101001110  Your Encoding
#
# 01000010010  Message
# 0100001000100100  Expected Encoding
# 0100001000100100  Your Encoding
```

```
#
# 01100000001  Message
# 0110000000001001  Expected Encoding
# 0110000000001001  Your Encoding
#
# 10100001101  Message
# 1010000011001001  Expected Encoding
# 1010000011001001  Your Encoding
#
# 00101110110  Message
# 0010111001110100  Expected Encoding
# 0010111001110100  Your Encoding
#
# 10100111101  Message
# 1010011011001111  Expected Encoding
# 1010011011001111  Your Encoding
#
# 11111101101  Message
# 1111110011001111  Expected Encoding
# 1111110011001111  Your Encoding
#
# 11110001100  Message
# 1111000011000011  Expected Encoding
# 1111000011000011  Your Encoding
#
# 00111111001  Message
# 0011111110011010  Expected Encoding
# 0011111110011010  Your Encoding
#
# ** Note: $stop    : C:/Users/Albert/Desktop/Course Related/cse
140L/Project_Folder_qPrime_modelsim/Lab_4.5/testbench_and_starter_verilog/lab4_tb.sv
(63)
#    Time: 1435 ns  Iteration: 0  Instance: /lab4_tb
# Break in Module lab4_tb at C:/Users/Albert/Desktop/Course Related/cse
140L/Project_Folder_qPrime_modelsim/Lab_4.5/testbench_and_starter_verilog/lab4_tb.sv
line 63
```