Calculate Hamming Weight of a Parallel Sequence

Problem Statement

Please design a RTL module that computes Hamming weight of a binary sequence of length 1024. The hamming weight of the sequence is always less than or equal to 31. The module also need to output the locations of 1's after receiving all the bits of a sequence in no more than 31 clock cycles (one location at a time). There is a start packet (frame) signal and this is asserted every 129 clock cycles. After the assertion of start packet signal, the bits would be input for 128 clock cycles at the rate of 8 bits/clk. This process is continuous.

Design Output format

The design counts the number of "1" received. There is a valid signal it means the output reported is valid only if valid signal is one. In special case of all zero frame the Hamming weight is zero and valid is one. The position would be zero it ***does not hold any meaning***.

Files and I/O:

Matlab file (IO_generate.m) generates the test vectors and expected outputs.
Do not run this ".m" in a folder that contains ".txt" files you might need. Running this file will erase them.
Put the ".txt" files in the Xilinx ISE 13.1 project folder.
Use the Project file with Xilinx ISE or the files the top module for test is "Test.v" file and for synthesis the top module is "Parallel.v"
Inputs:
"rst" is Reset signal (generated in the verilog test bench)
"clk" is Clock Signal (generated in the verilog test bench)
"seq" input sequence (Input.txt)
Outputs:
"pos" reports the positions of "1"s (Pos.txt)
"v" if one the out puts are valid (V.txt)
"hw" Hamming Weight of the previous frame (HW.txt)

Solution Statement

The Idea is to first remove the zero bytes (bytes with all bits equal to zero). we save the byte alongside the byte position (which cycle of frame) it is received. Then we use a state machine to convert the position of bit inside the byte in addition to the byte address to a unite bit-address for each "1" in the frame.

In case a byte has more than a bit of "1" we report the next "1" in the next clock cycle. This process continues for as many as eight clock cycles (if we have an all "1" byte) then we repeat the process for next byte.

Utilization Summary

```
Device utilization summary:
---------------------------

Selected Device : 3s500efg320-4

 Number of Slices:                130  out of   4656     2%
 Number of Slice Flip Flops:       77  out of   9312     0%
 Number of 4 input LUTs:          307  out of   9312     3%
    Number used as logic:         187
    Number used as RAMs:          120
 Number of IOs:                   30
 Number of bonded IOBs:            30  out of    232    12%
 Number of GCLKs:                   1  out of     24     4%
```

Timing Summary

The clock frequency is 92MHz. the throughput will be 92x8x128/129 that is approximately 3.25 Gbps.

```
=========================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 10.865ns (frequency: 92.039MHz)
  Total number of paths / destination ports: 18069 / 708
-------------------------------------------------------------------------
Delay:               10.865ns (Levels of Logic = 7)
  Source:            p2s_c_2 (FF)
  Destination:       memout_0 (FF)
  Source Clock:      clk falling
  Destination Clock: clk falling

  Data Path: p2s_c_2 to memout_0
                        Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     FDRS_1:C->Q          13   0.591   1.018  p2s_c_2 (p2s_c_2)
     LUT3_D:I2->O          3   0.704   0.535  _and0000<5>1 (_and0000<5>)
     LUT4:I3->O            1   0.704   0.455  P1/Exp<0>11 (P1/Exp<0>11)
     LUT4:I2->O            5   0.704   0.712  P1/Exp<0>38 (Exp<0>)
     LUT4:I1->O            3   0.704   0.610  P2/InM<5>1 (P2/InM<5>)
     LUT4:I1->O            6   0.704   0.673  p2s_c_mux0000<2>11 (N22)
     LUT4_D:I3->O         14   0.704   1.035  memout_mux0000<0>21 (N52)
     LUT4:I2->O            1   0.704   0.000  memout_mux0000<8>28 (memout_mux0000<8>)
     FDE_1:D                   0.308          memout_8
    ----------------------------------------
    Total                    10.865ns (5.827ns logic, 5.038ns route)
                                 (53.6% logic, 46.4% route)
=========================================================================
```

Output Snapshot

Figure 1 demonstrate the snapshot at the time of an output. The output matched the expected values. There is one clock cycle delay between the actual outputs and the value passed to the Verilog tasks due to these values being defined as register.

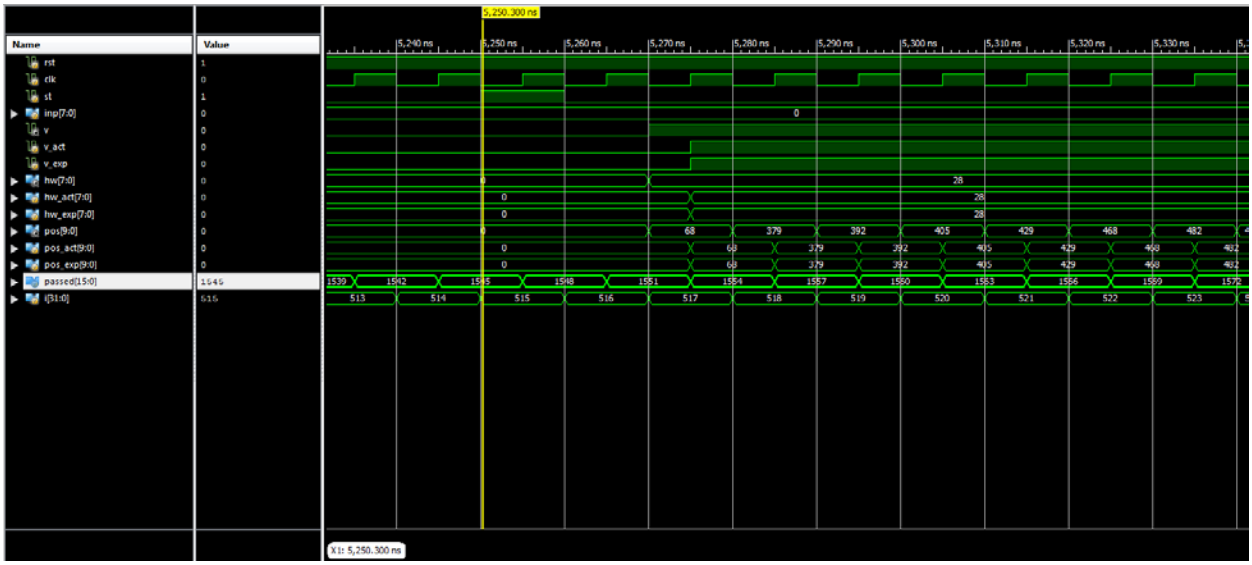

Figure 1 Snapshot of the outputs and expected values.

Figure 2 demonstrates the test format of the self diagnosing TestBench that shows all tests have passed the check. This means all the expected value were equal to actual values.

```
Clock     13028 test starts
Output v match
Output pos match
Output hw match
ALL output signals in this clock cycle matched match
Clock     13028 test ends

|-=-=-=-=-=-=-=-=-=-=-=-=-=-|
        |!!All tests passed!!|
|-=-=-=-=-=-=-=-=-=-=-=-=-=-|
```

Figure 2 All test passed

Time Table

Spend approximately 16 hours on this project. Most of the time I spend was on removing Bugs in each step. This all is done while I wasn't able to seat and I had to lay back and use laptop due to backache I have right now.

Task one (Making Matlab model to generate the results) took one hour mostly spend on formatting the output ".txt" files.

Task two (Verilog RTL code and testing it Vs. some simple input sequence) 10 hours. I needed to change the inputs of the module from Friday version so it match the new description.

Task three (writing the TestBench and running expected outputs against actual outputs) . Two hours mostly spend on finding and solving occasional errors (all zero frame and a frame with last element “1”).

Task four (Synthesis and timing analysis) 2 hours. Spend on changing the memory design to read only on clock edge and removing warnings and unused signals. Also I had to define the input and output time constrains.

Task five (Time spend on writing this document) approximately an hour.