# ----------------------------------------------------------------------

# Name Type Size Value

# ----------------------------------------------------------------------

# uvm\_test\_top udp\_reader\_uvm\_test - @359

# env udp\_reader\_uvm\_env - @371

# agent udp\_reader\_uvm\_agent - @379

# agent\_ap\_compare uvm\_analysis\_port - @413

# agent\_ap\_output uvm\_analysis\_port - @404

# drvr udp\_reader\_uvm\_driver - @545

# rsp\_port uvm\_analysis\_port - @562

# seq\_item\_port uvm\_seq\_item\_pull\_port - @553

# mon\_cmp udp\_reader\_uvm\_monitor\_compare - @579

# mon\_ap\_compare uvm\_analysis\_port - @589

# mon\_out udp\_reader\_uvm\_monitor\_output - @571

# mon\_ap\_output uvm\_analysis\_port - @599

# seqr uvm\_sequencer - @422

# rsp\_export uvm\_analysis\_export - @430

# seq\_item\_export uvm\_seq\_item\_pull\_imp - @536

# arbitration\_queue array 0 -

# lock\_queue array 0 -

# num\_last\_reqs integral 32 'd1

# num\_last\_rsps integral 32 'd1

# sb udp\_reader\_uvm\_scoreboard - @387

# compare\_fifo uvm\_tlm\_analysis\_fifo #(T) - @684

# analysis\_export uvm\_analysis\_imp - @728

# get\_ap uvm\_analysis\_port - @719

# get\_peek\_export uvm\_get\_peek\_imp - @701

# put\_ap uvm\_analysis\_port - @710

# put\_export uvm\_put\_imp - @692

# output\_fifo uvm\_tlm\_analysis\_fifo #(T) - @631

# analysis\_export uvm\_analysis\_imp - @675

# get\_ap uvm\_analysis\_port - @666

# get\_peek\_export uvm\_get\_peek\_imp - @648

# put\_ap uvm\_analysis\_port - @657

# put\_export uvm\_put\_imp - @639

# sb\_export\_compare uvm\_analysis\_export - @622

# sb\_export\_output uvm\_analysis\_export - @613

# ----------------------------------------------------------------------

#

# UVM\_INFO ../uvm/udp\_reader\_uvm\_sequence.sv(37) @ 0: uvm\_test\_top.env.agent.seqr@@seq [SEQ\_RUN] Loading file ../source/test.pcap...

# UVM\_INFO ../uvm/udp\_reader\_uvm\_sequence.sv(54) @ 0: uvm\_test\_top.env.agent.seqr@@seq [SEQ\_RUN] Packet size: 1066

# UVM\_INFO ../uvm/udp\_reader\_uvm\_sequence.sv(54) @ 10665: uvm\_test\_top.env.agent.seqr@@seq [SEQ\_RUN] Packet size: 1066

# UVM\_INFO ../uvm/udp\_reader\_uvm\_sequence.sv(54) @ 21325: uvm\_test\_top.env.agent.seqr@@seq [SEQ\_RUN] Packet size: 1066

# UVM\_INFO ../uvm/udp\_reader\_uvm\_sequence.sv(54) @ 31985: uvm\_test\_top.env.agent.seqr@@seq [SEQ\_RUN] Packet size: 949

# UVM\_INFO ../uvm/udp\_reader\_uvm\_sequence.sv(54) @ 41475: uvm\_test\_top.env.agent.seqr@@seq [SEQ\_RUN] Packet size: 0

# UVM\_INFO ../uvm/udp\_reader\_uvm\_sequence.sv(70) @ 41475: uvm\_test\_top.env.agent.seqr@@seq [SEQ\_RUN] Closing file ../source/test.pcap...

# UVM\_INFO /vol/mentor/questa\_sim-2019.3\_2/questasim/verilog\_src/uvm-1.2/src/base/uvm\_objection.svh(1270) @ 51565: reporter [TEST\_DONE] 'run' phase is ready to proceed to the 'extract' phase

# UVM\_INFO ../uvm/udp\_reader\_uvm\_monitor.sv(129) @ 51565: uvm\_test\_top.env.agent.mon\_cmp [MON\_CMP\_FINAL] Closing file ../source/test.txt...

# UVM\_INFO ../uvm/udp\_reader\_uvm\_monitor.sv(59) @ 51565: uvm\_test\_top.env.agent.mon\_out [MON\_OUT\_FINAL] Closing file ../source/test\_output.txt...

# UVM\_INFO /vol/mentor/questa\_sim-2019.3\_2/questasim/verilog\_src/uvm-1.2/src/base/uvm\_report\_server.svh(847) @ 51565: reporter [UVM/REPORT/SERVER]

# --- UVM Report Summary ---

#

# \*\* Report counts by severity

# UVM\_INFO : 15

# UVM\_WARNING : 0

# UVM\_ERROR : 0

# UVM\_FATAL : 0

# \*\* Report counts by id

# [MON\_CMP\_FINAL] 1

# [MON\_OUT\_FINAL] 1

# [Questa UVM] 2

# [RNTST] 1

# [SEQ\_RUN] 7

# [TEST\_DONE] 1

# [UVM/RELNOTES] 1

# [UVMTOP] 1

#

# \*\* Note: $finish : /vol/mentor/questa\_sim-2019.3\_2/questasim/verilog\_src/uvm-1.2/src/base/uvm\_root.svh(517)

# Time: 51565 ns Iteration: 70 Instance: /udp\_reader\_uvm\_tb

# End time: 13:39:08 on Feb 19,2024, Elapsed time: 0:00:01

# Errors: 0, Warnings: 0

Simulation Cycle Count = **5166 cycles** (UVM simulation)

Simulation Total Time = **51,656 ns** (UVM simulation)

Throughput = **77,435 packets/second** (4 packets / 51656 ns or 0.000051656 s)

Max Frequency = **108.5 MHz**

Total combinational functions 625 of 6272 ( 9%)

Logic element usage by number of inputs

4 input functions 311

3 input functions 94

[=2 input functions 220

Logic elements by mode

normal mode 460

arithmetic mode 165

Total registers 361 of 6272 ( 5%)

I/O pins 24 of 180 (13%), total I/O based on largest package of this part.

Number of I/O registers

Input DDRs :0

Output DDRs :0

DSP Blocks: 0 (0 nine-bit DSP elements).

DSP Utilization: 0.00% of available 15 blocks (30 nine-bit).

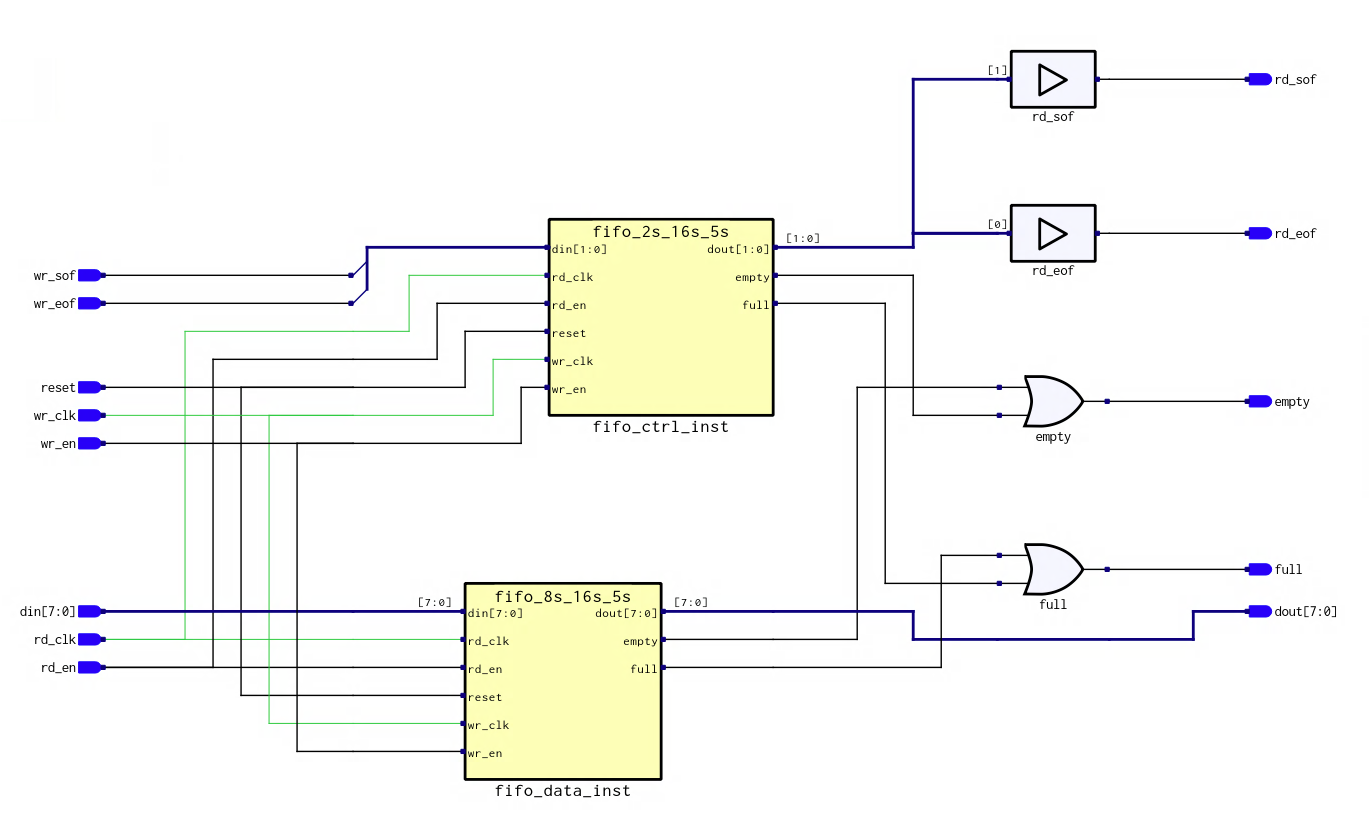
ShiftTap: 0 (0 registers)

Total ESB: 8192 bits

**udp\_reader\_top:A diagram of a computer circuit

Description automatically generated**

**fifo\_ctrl:**



**FSM State Diagram:**

A diagram of a circle with blue circles and green lines

Description automatically generated

**Architecture discussion:**

My udp\_reader works like how we discussed it in class: a fifo\_ctrl buffer that includes a start of file and end of file signal takes in the UDP packet from the PCAP file, and then feeds it byte by byte to udp\_reader. There are 4 packets back-to-back inside the PCAP file so as soon as an end of file is asserted, the start of file signal for the next packer is asserted in the next clock cycle.

Once a start of file signal has been detected, the udp\_reader goes through each header field, checking those than need to be checked against a default value and adding the necessary ones into a checksum buffer. When the header has been read, the reader goes into a data reading state where it takes in the data bytes byte at a time per clock cycle and adds them to the checksum buffer. When the end of file signal is asserted, it goes back into WAIT\_FOR\_SOF\_STATE to wait for the next start of file signal. This data is passed directly into an output FIFO buffer that will hold the data while the checksum is being validated.

After this happens, in a parallel process outside of the state machine, the checksum is folded down to 16 bits and its two's complement value is taken. This value is then compared to the checksum value that was given in the UDP header. If the checksum value calculated does not match up, an error signal is asserted, and the output FIFO buffer is reset or cleared. If it is correct, then the output FIFO buffer can start bursting out data byte by byte. A counter, set to the UDP data length (excluding header bytes), is decremented every clock cycle to make sure that the correct number of bytes is burst out of the output FIFO buffer. This is because while the data is being burst out, another UDP packet's data could be streaming into the output buffer, but we cannot burst that data yet as its checksum has not yet been validated. While the checksum is being validated, a checksum\_valid signal is kept low which asserts the output FIFO buffer's empty signal, not allowing any data to be streamed out even if the FIFO itself is not empty. When checksum is validated, then checksum\_valid is asserted causing the output FIFO buffer's empty signal to be pulled low and for data to be burst out.