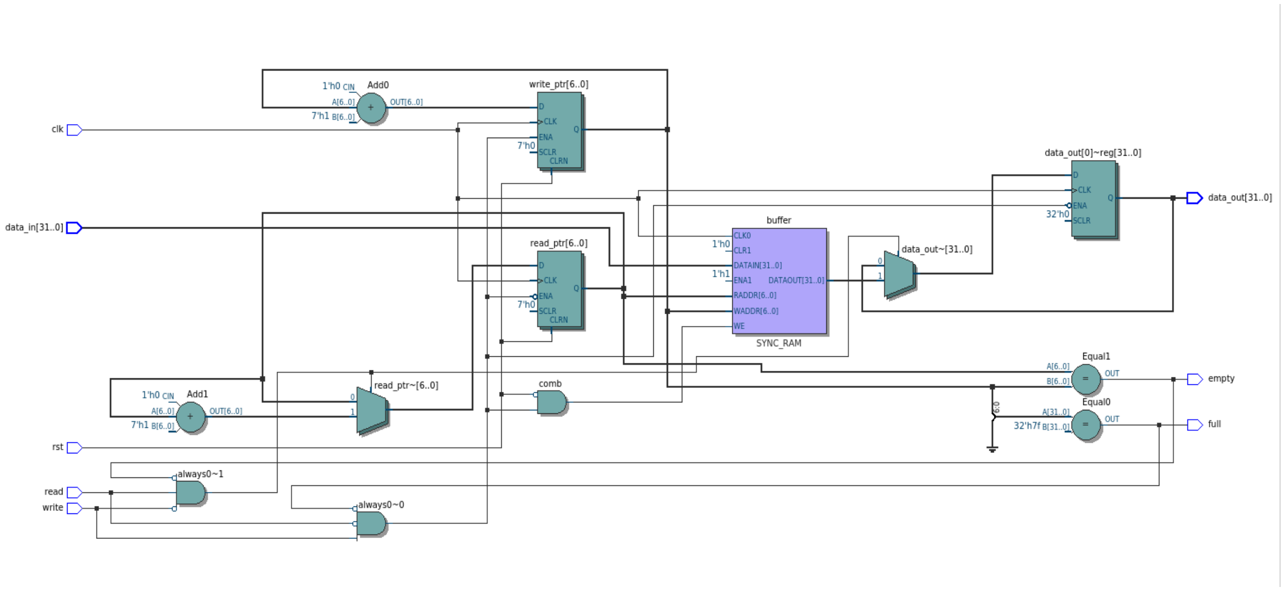
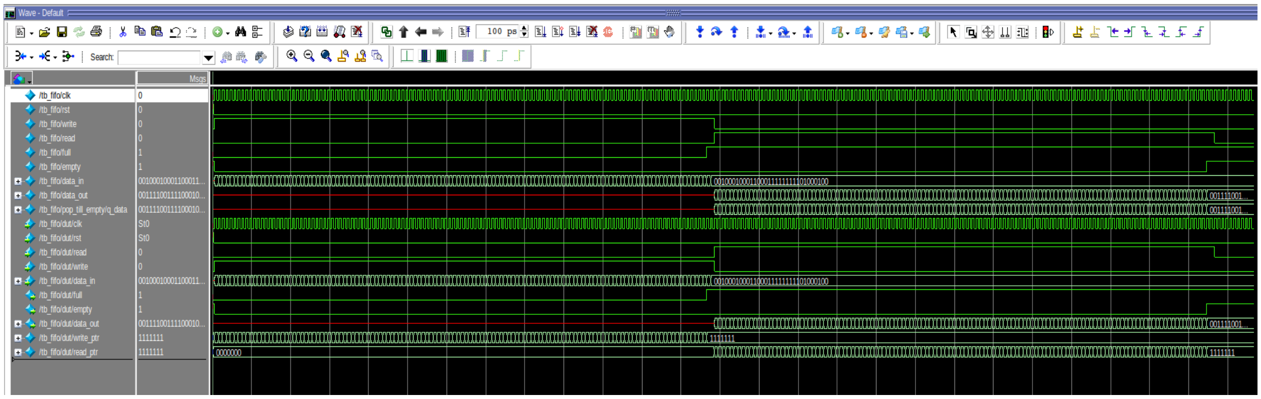
“3. Single Clock FIFO buffer 32wx128d”

Design:



- The fifo module implements a single clock buffer with 32 bit width and 128 depth capacity.
- Data is written into the buffer when write is asserted and not full, advancing the write pointer.
- Data is read out when read is asserted and not empty, advancing the read pointer.
- Reset clears both pointers and sets the output to unknown for proper initialization.
- Full and empty flags are generated by comparing write and read pointers, they are controlling buffer usage.

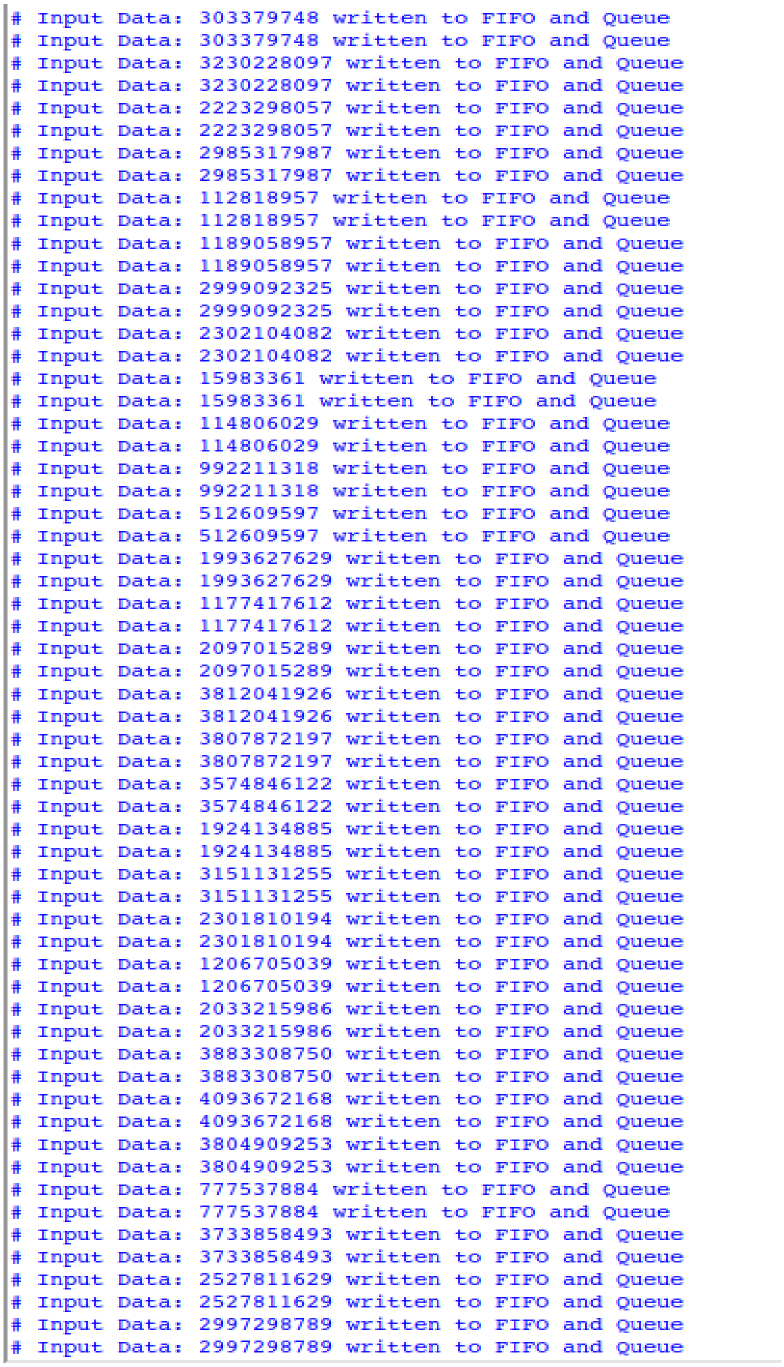
Verification (using Queue):



	Msgs
/tb_fifo/clk	0
/tb_fifo/rst	0
/tb_fifo/write	0
/tb_fifo/read	0
/tb_fifo/full	1
/tb_fifo/empty	1
/tb_fifo/data_in	00100010001100011...
/tb_fifo/data_out	00111100111100010...
/tb_fifo/pop_till_empty/q_data	00111100111100010...
/tb_fifo/dut/clk	St0
/tb_fifo/dut/rst	St0
/tb_fifo/dut/read	0
/tb_fifo/dut/write	0
/tb_fifo/dut/data_in	00100010001100011...
/tb_fifo/dut/full	1
/tb_fifo/dut/empty	1
/tb_fifo/dut/data_out	00111100111100010...
/tb_fifo/dut/write_ptr	1111111
/tb_fifo/dut/read_ptr	1111111

1. *Continuous Write operation before Full:*

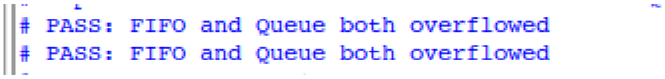
- Random data words are generated and written into the FIFO while simultaneously pushed into the queue.
- The write pointer advances until the FIFO reaches its maximum depth of 128 entries.
- Each successful write is confirmed by displaying the input data stored in both FIFO and queue.



```
# Input Data: 303379748 written to FIFO and Queue
# Input Data: 303379748 written to FIFO and Queue
# Input Data: 3230228097 written to FIFO and Queue
# Input Data: 3230228097 written to FIFO and Queue
# Input Data: 2223298057 written to FIFO and Queue
# Input Data: 2223298057 written to FIFO and Queue
# Input Data: 2985317987 written to FIFO and Queue
# Input Data: 2985317987 written to FIFO and Queue
# Input Data: 112818957 written to FIFO and Queue
# Input Data: 112818957 written to FIFO and Queue
# Input Data: 1189058957 written to FIFO and Queue
# Input Data: 1189058957 written to FIFO and Queue
# Input Data: 2999092325 written to FIFO and Queue
# Input Data: 2999092325 written to FIFO and Queue
# Input Data: 2302104082 written to FIFO and Queue
# Input Data: 2302104082 written to FIFO and Queue
# Input Data: 15983361 written to FIFO and Queue
# Input Data: 15983361 written to FIFO and Queue
# Input Data: 114806029 written to FIFO and Queue
# Input Data: 114806029 written to FIFO and Queue
# Input Data: 992211318 written to FIFO and Queue
# Input Data: 992211318 written to FIFO and Queue
# Input Data: 512609597 written to FIFO and Queue
# Input Data: 512609597 written to FIFO and Queue
# Input Data: 1993627629 written to FIFO and Queue
# Input Data: 1993627629 written to FIFO and Queue
# Input Data: 1177417612 written to FIFO and Queue
# Input Data: 1177417612 written to FIFO and Queue
# Input Data: 2097015289 written to FIFO and Queue
# Input Data: 2097015289 written to FIFO and Queue
# Input Data: 3812041926 written to FIFO and Queue
# Input Data: 3812041926 written to FIFO and Queue
# Input Data: 3807872197 written to FIFO and Queue
# Input Data: 3807872197 written to FIFO and Queue
# Input Data: 3574846122 written to FIFO and Queue
# Input Data: 3574846122 written to FIFO and Queue
# Input Data: 1924134885 written to FIFO and Queue
# Input Data: 1924134885 written to FIFO and Queue
# Input Data: 3151131255 written to FIFO and Queue
# Input Data: 3151131255 written to FIFO and Queue
# Input Data: 2301810194 written to FIFO and Queue
# Input Data: 2301810194 written to FIFO and Queue
# Input Data: 1206705039 written to FIFO and Queue
# Input Data: 1206705039 written to FIFO and Queue
# Input Data: 2033215986 written to FIFO and Queue
# Input Data: 2033215986 written to FIFO and Queue
# Input Data: 3883308750 written to FIFO and Queue
# Input Data: 3883308750 written to FIFO and Queue
# Input Data: 4093672168 written to FIFO and Queue
# Input Data: 4093672168 written to FIFO and Queue
# Input Data: 3804909253 written to FIFO and Queue
# Input Data: 3804909253 written to FIFO and Queue
# Input Data: 777537884 written to FIFO and Queue
# Input Data: 777537884 written to FIFO and Queue
# Input Data: 3733858493 written to FIFO and Queue
# Input Data: 3733858493 written to FIFO and Queue
# Input Data: 2527811629 written to FIFO and Queue
# Input Data: 2527811629 written to FIFO and Queue
# Input Data: 2997298789 written to FIFO and Queue
# Input Data: 2997298789 written to FIFO and Queue
```

2. *Write Operation after Full (Full Flag Check):*

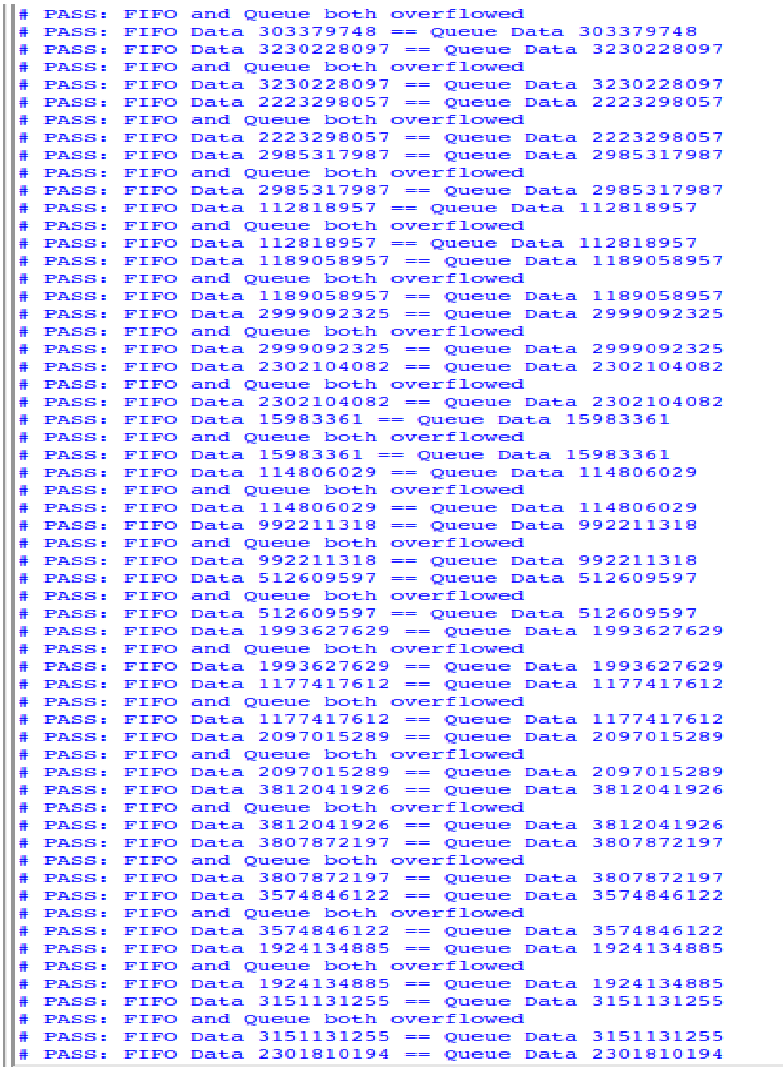
- Once the FIFO reaches capacity, the full flag is asserted to prevent further writes.
- The queue size is checked against the depth to confirm overflow behavior matches the FIFO.
- A PASS message is displayed when both FIFO and queue indicate overflow.



```
# PASS: FIFO and Queue both overflowed
# PASS: FIFO and Queue both overflowed
```

3. *Continuous Read Operation before Empty (Order Check):*

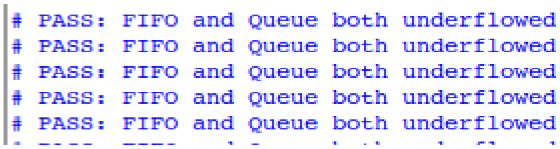
- Data is read from the FIFO while simultaneously popped from the front of the queue.
- Each read compares FIFO output with queue data to verify correct order preservation.
- A PASS or fail message is displayed depending on whether the FIFO output matches the expected queue value.



```
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 303379748 == Queue Data 303379748
# PASS: FIFO Data 3230228097 == Queue Data 3230228097
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 3230228097 == Queue Data 3230228097
# PASS: FIFO Data 2223298057 == Queue Data 2223298057
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 2223298057 == Queue Data 2223298057
# PASS: FIFO Data 2985317987 == Queue Data 2985317987
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 2985317987 == Queue Data 2985317987
# PASS: FIFO Data 112818957 == Queue Data 112818957
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 112818957 == Queue Data 112818957
# PASS: FIFO Data 1189058957 == Queue Data 1189058957
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 1189058957 == Queue Data 1189058957
# PASS: FIFO Data 2999092325 == Queue Data 2999092325
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 2999092325 == Queue Data 2999092325
# PASS: FIFO Data 2302104082 == Queue Data 2302104082
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 2302104082 == Queue Data 2302104082
# PASS: FIFO Data 15983361 == Queue Data 15983361
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 15983361 == Queue Data 15983361
# PASS: FIFO Data 114806029 == Queue Data 114806029
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 114806029 == Queue Data 114806029
# PASS: FIFO Data 992211318 == Queue Data 992211318
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 992211318 == Queue Data 992211318
# PASS: FIFO Data 512609597 == Queue Data 512609597
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 512609597 == Queue Data 512609597
# PASS: FIFO Data 1993627629 == Queue Data 1993627629
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 1993627629 == Queue Data 1993627629
# PASS: FIFO Data 1177417612 == Queue Data 1177417612
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 1177417612 == Queue Data 1177417612
# PASS: FIFO Data 2097015289 == Queue Data 2097015289
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 2097015289 == Queue Data 2097015289
# PASS: FIFO Data 3812041926 == Queue Data 3812041926
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 3812041926 == Queue Data 3812041926
# PASS: FIFO Data 3807872197 == Queue Data 3807872197
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 3807872197 == Queue Data 3807872197
# PASS: FIFO Data 3574846122 == Queue Data 3574846122
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 3574846122 == Queue Data 3574846122
# PASS: FIFO Data 1924134885 == Queue Data 1924134885
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 1924134885 == Queue Data 1924134885
# PASS: FIFO Data 3151131255 == Queue Data 3151131255
# PASS: FIFO and Queue both overflowed
# PASS: FIFO Data 3151131255 == Queue Data 3151131255
# PASS: FIFO Data 2301810194 == Queue Data 2301810194
```

4. Read Operation after Empty (Empty Flag Check):

- When all entries are read, the FIFO asserts the empty flag to block further reads.
- The queue size is checked to ensure it is also empty, confirming underflow behavior.
- A PASS message is displayed when both FIFO and queue correctly indicate empty state.



```
# PASS: FIFO and Queue both underflowed
# PASS: FIFO and Queue both underflowed
# PASS: FIFO and Queue both underflowed
# PASS: FIFO and Queue both underflowed
# PASS: FIFO and Queue both underflowed
```