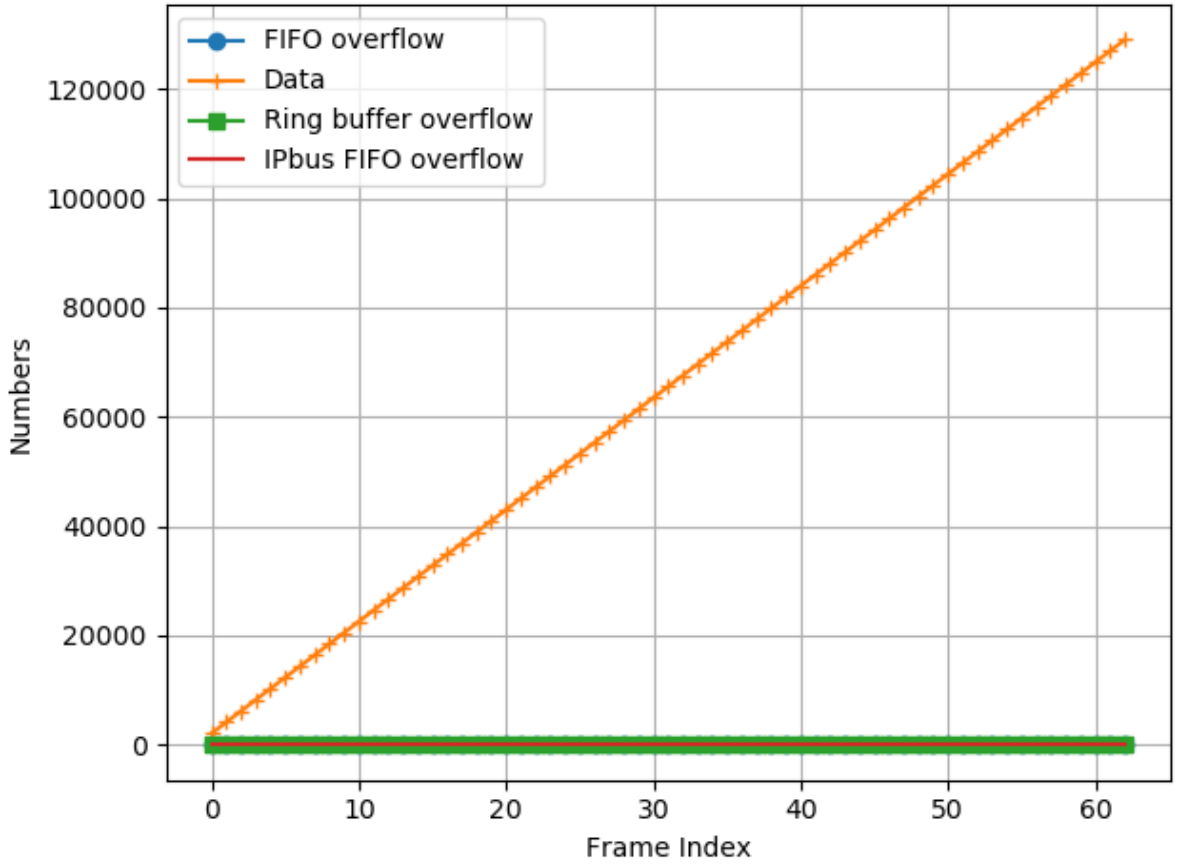
Jadepix3 FW and SW Status

Monday, November 2, 2020

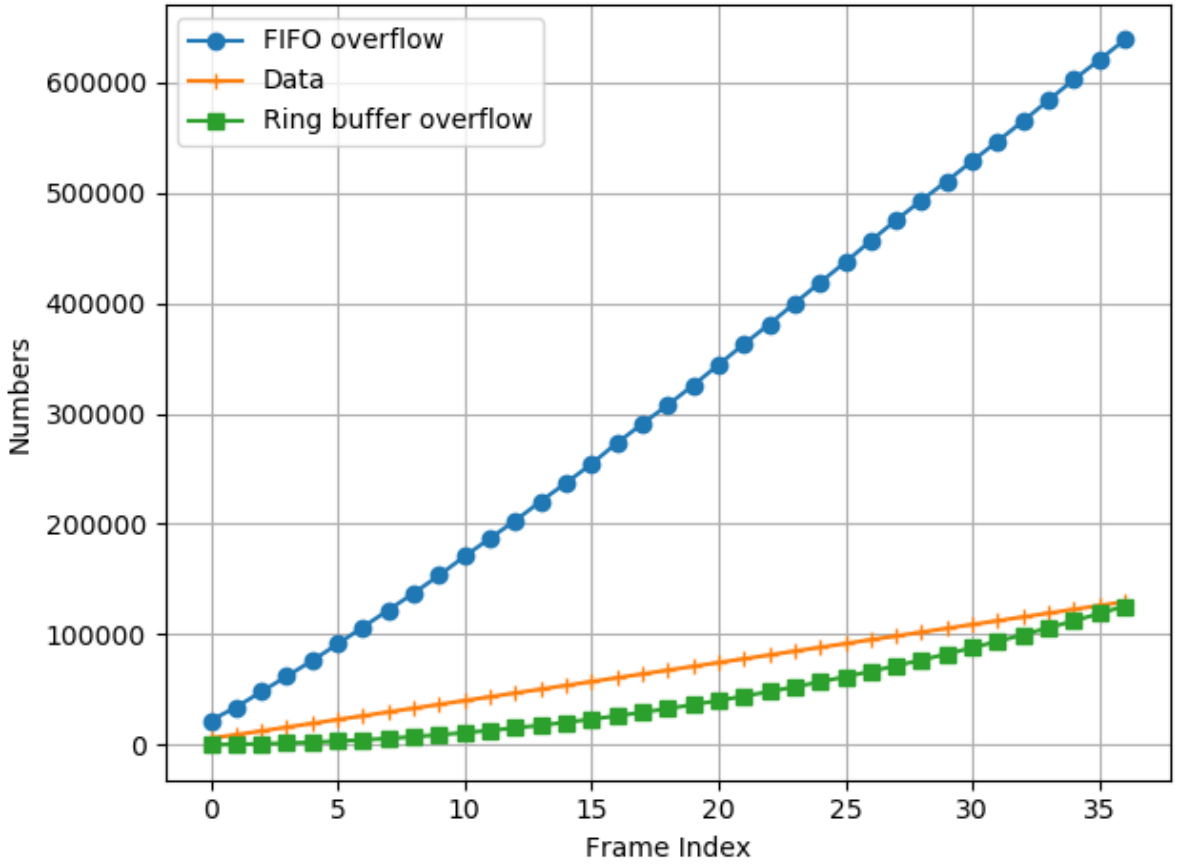
Sheng Dong, CCNU

s.dong@mails.ccnu.edu.cn

The first results of data readout

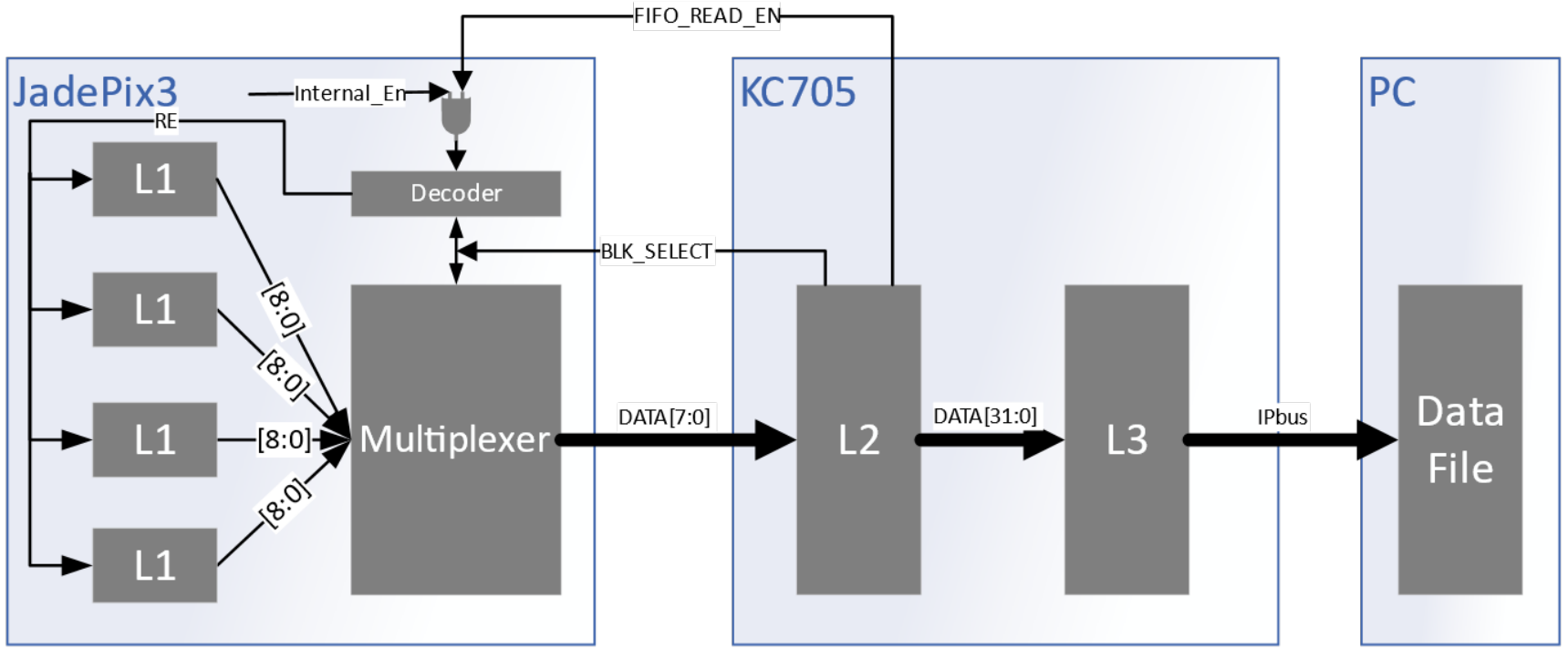


VALID_IN=(others=>clk_cache), DATA_IN =8X"FF"
Valid num per block = 1
Data number = Data FIFO Depth = 131072



VALID_IN = 4X"F", DATA_IN = 8X"FF"
Valid num per block = 16
Data number = Data FIFO Depth = 131072

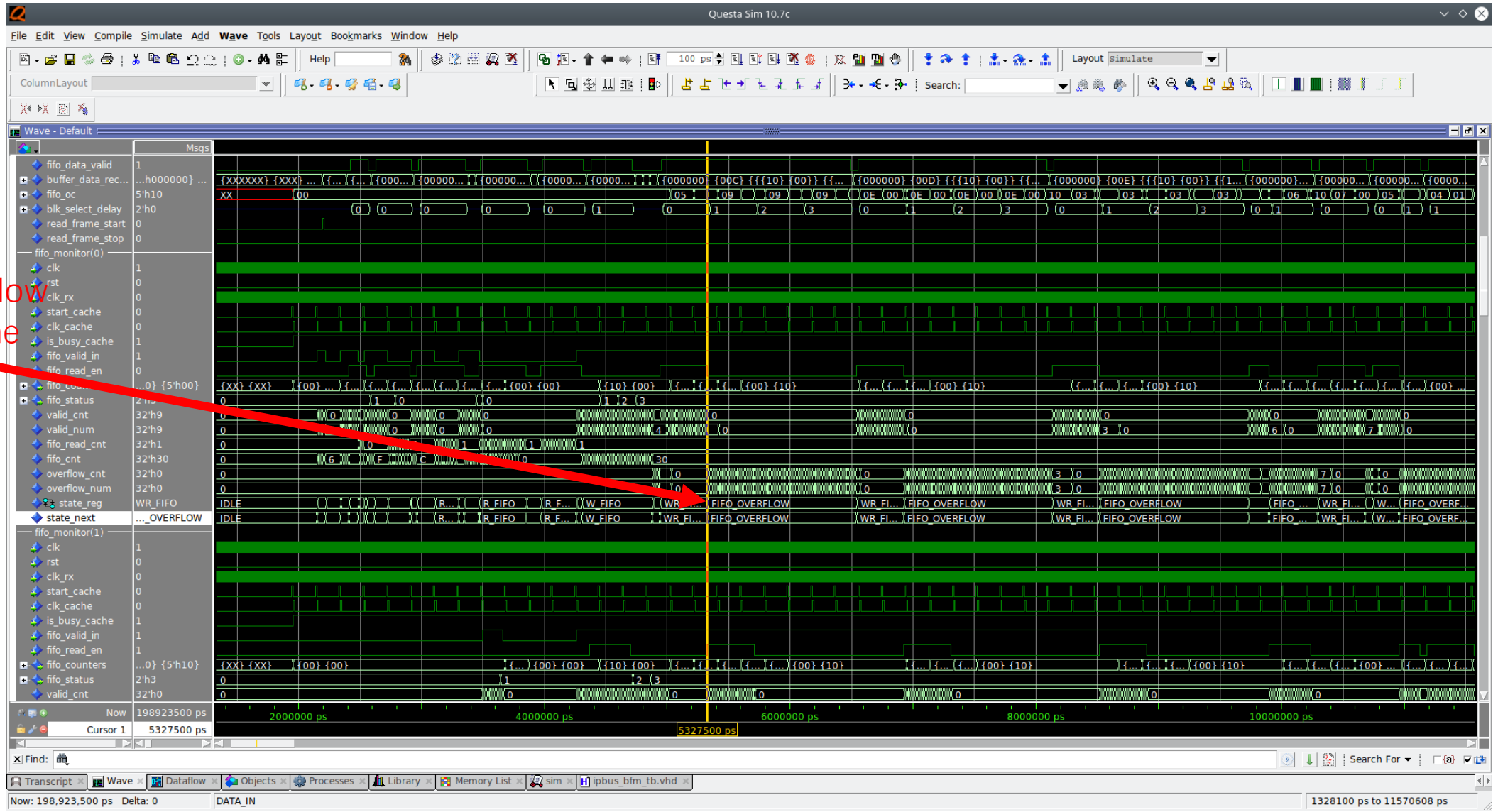
Data Loss Discussion



- Three Level Caches
  - L1: the cache FIFO in the chip, size=48*16 bits
  - L2: the ring buffer in the FPGA, size=192*86 bits
  - L3: the IPbus data FIFO in the FPGA, size=2^17*32 bits
- Since the FIFO in the chip is a synchronous FIFO, the readout speed is same as the write speed. We have 4 FIFOs input and 1 selected output. In principle, the readout speed should be 4x than the write speed. So the caches will be overflow soon if we have plenty of input data, then the data lost.
- L1 low speed readout -> L1 and L2 overflow
- PC software low speed readout -> L3 overflow

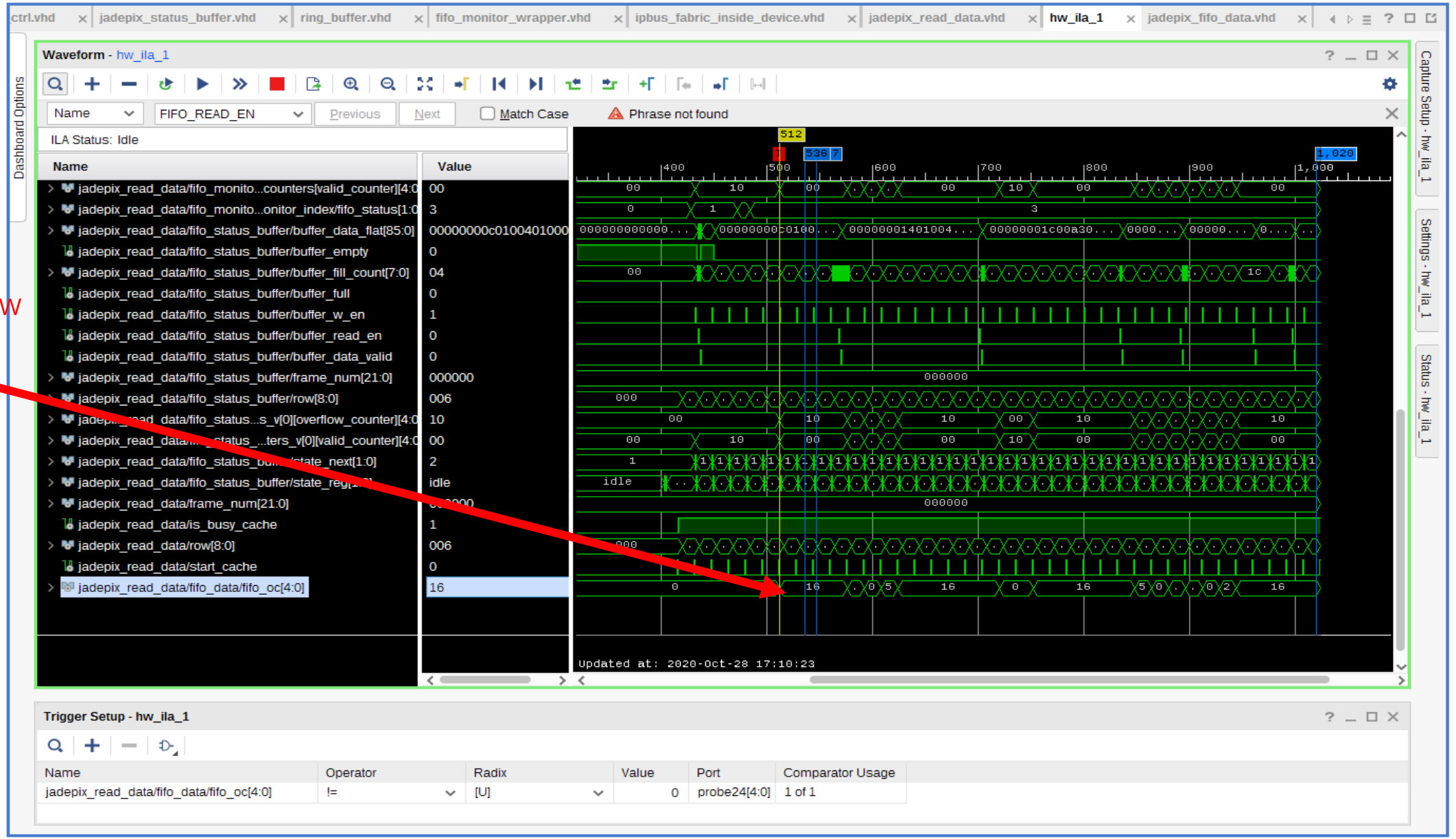
L1 Overflow -- Simulation

FIFO Overflow
@first frame



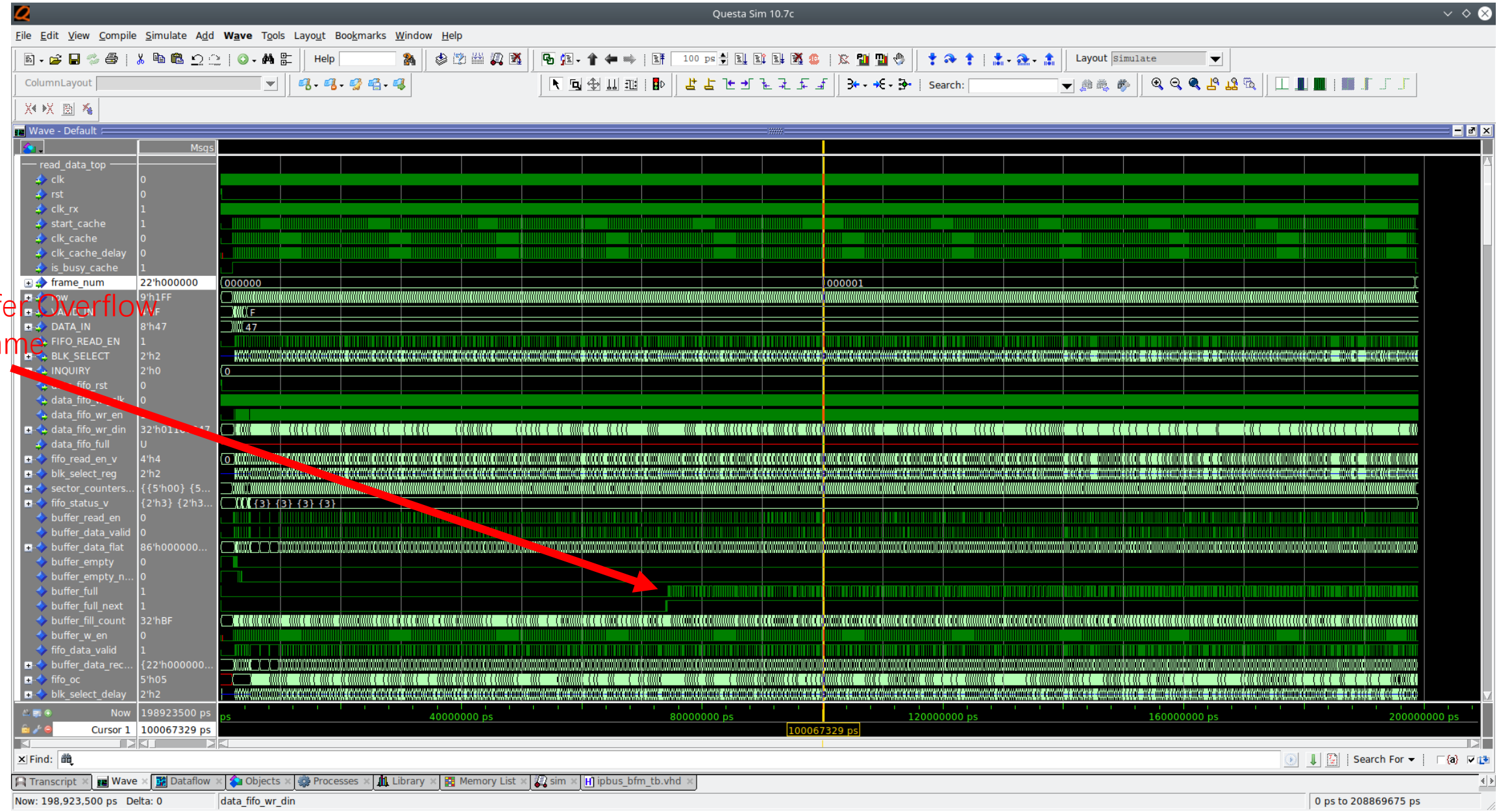
L1 Overflow– ILA Test

FIFO Overflow
@first frame



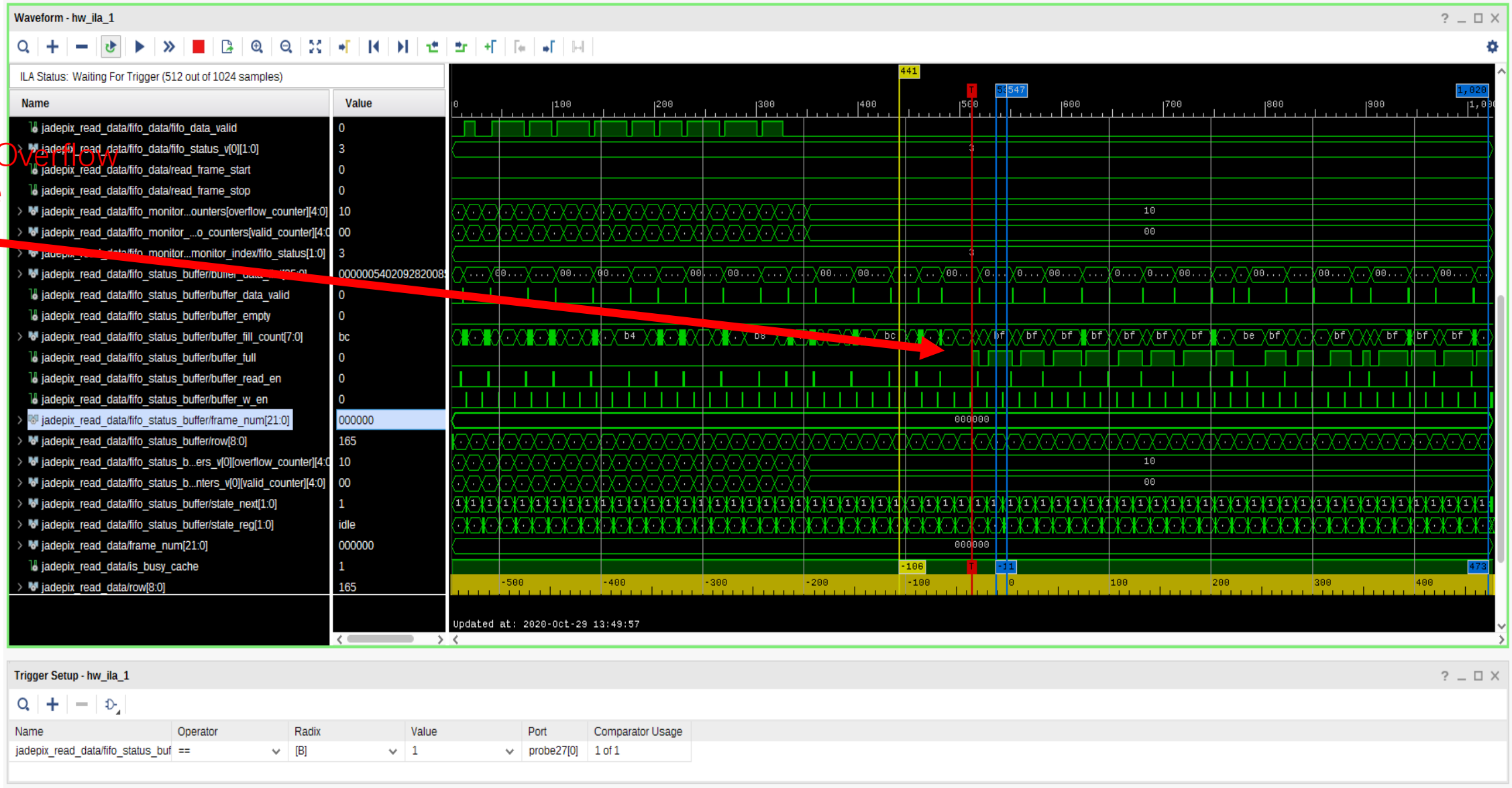
L2 Overflow– Simulation

Ring buffer Overflow @first frame



L2 Overflow– ILA Test

Ring buffer Overflow
@first frame



IPbus-based Data Readout

The maximum data amount per frame:

Data Amount$_{frame}$ = N_ROW * N_BLK * Valid$_{max}$ * DATA_WIDTH

= 512 * 4 * 16 * 16 bits = 0.13MByte

Time$_{frame}$ = 512*192ns=98.3 us

Valid output speed = (Data Amount$_{frame}$/ Time$_{frame}$)/4

= **1.334 Gbit/s**

The size of readout FIFO:

FIFO_Size = 2^{17}* 32bits = 0.52 MByte

>> The 1-client-to-1-device block read/write throughput for payloads larger than **1 Mbyte** is above **0.5 Gbit/s**

Speed test (Payload=0.52MByte) :

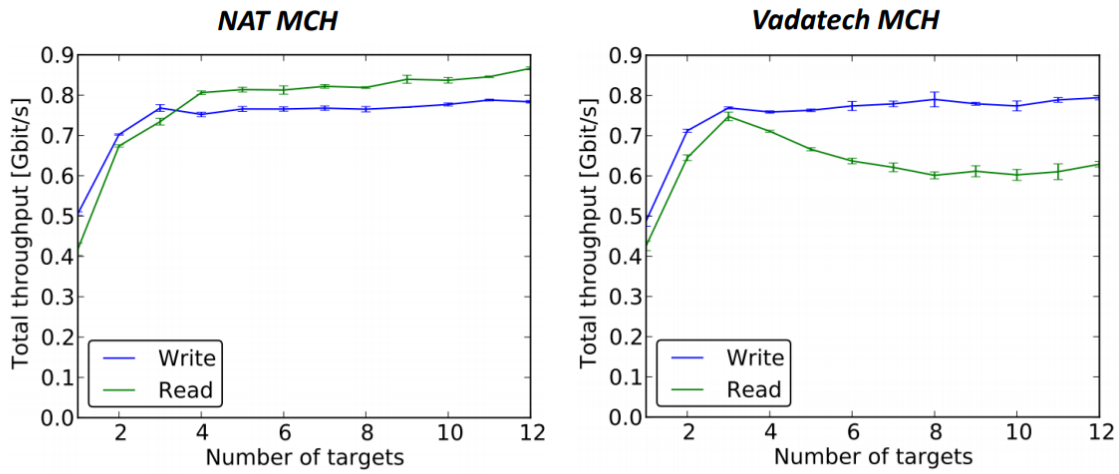
Write to .txt file: 200*0.52MB/23.4s = 35.6 Mbit/s

Write to memory: 200*0.52MB/14.8s = 56.2 Mbit/s

That means the IPbus readout FIFO can be overflow possible!

Performance (4)

- Block writes/reads, multiple targets
  - 1 client per target; 600MB read from / written to crate
  - Default IPbus software setup

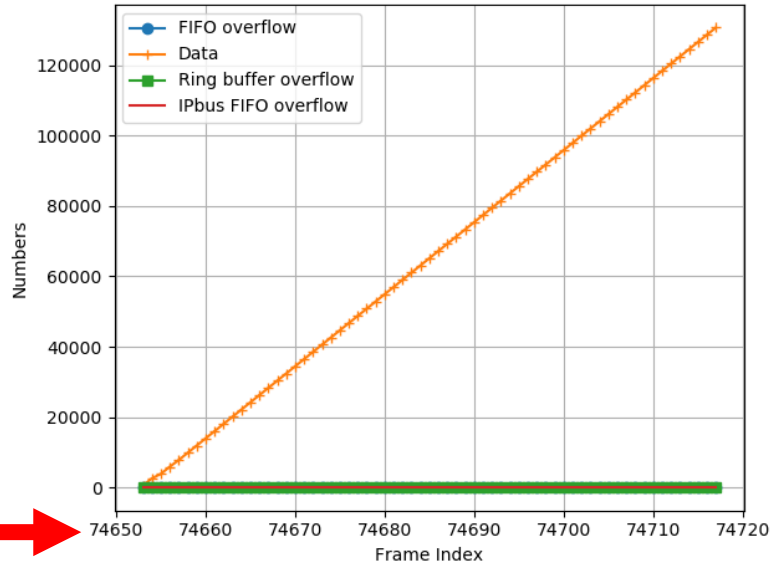


More Investigation

- The VALID_IN speed in reality. Then we can do some simulation and ILA/board level test.
- Remove empty frame (valid number = 0, OC=0) in L2 to reduce the OV?
- IPbus block readout logic. The block size and slice
- More accurate data analysis…(for example, use RO Class in root)
- RX_FPGA and clk_cache must be synchronized.

Bug:

Data cache somehow not cleaned at the beginning of each run. The frame index will not start from 0!



VALID_IN=(others=>clk_cache)
DATA_IN =8X"FF"

BACKUP

Frame Format

	Reserved [31:25]	Flag [24:23]	Payload[22:0]		
Head	0000000	01	FIFO_STATUS [22:15]	RBOF [14:0]	
Data	0000000	10	CHIP_FIFO_OC [22:18]	BLK_SELECT [17:16]	DATA [15:0]
Tail	0000000	00	0 [22:22]		Frame_Num [21:0]
Error (IPbus FIFO OF)	0000000	11	IPBUS FIFO_OC [22:0]		