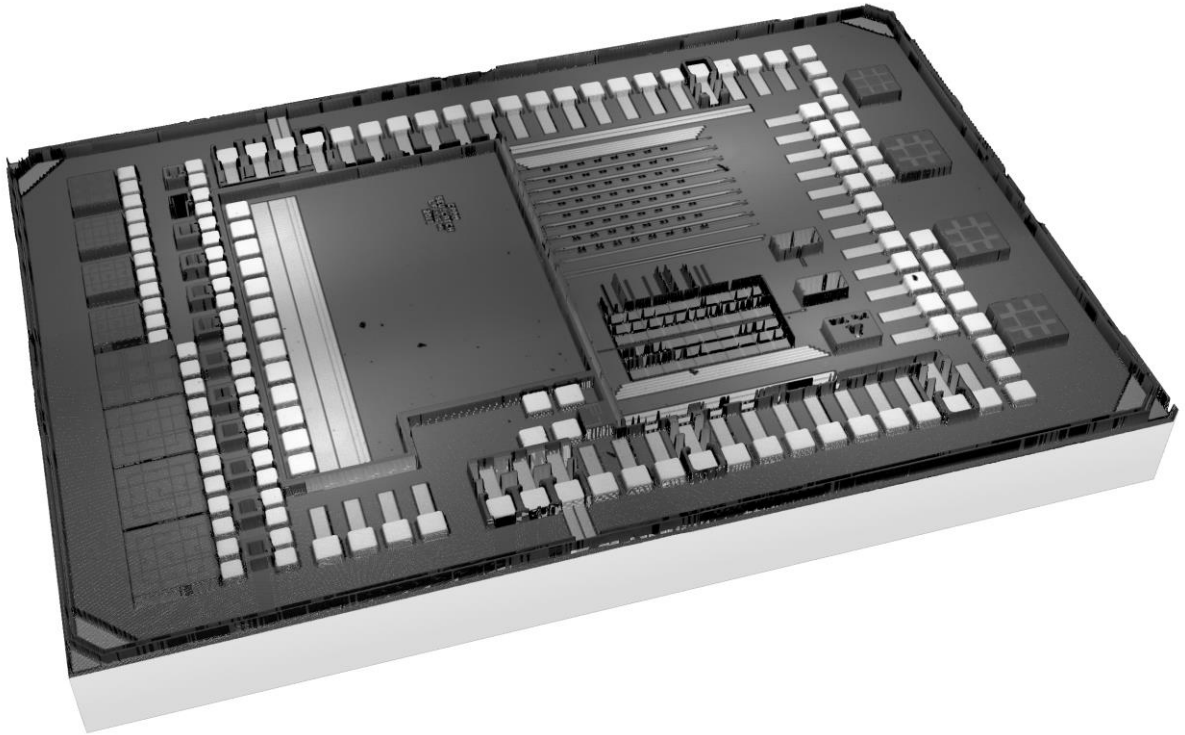


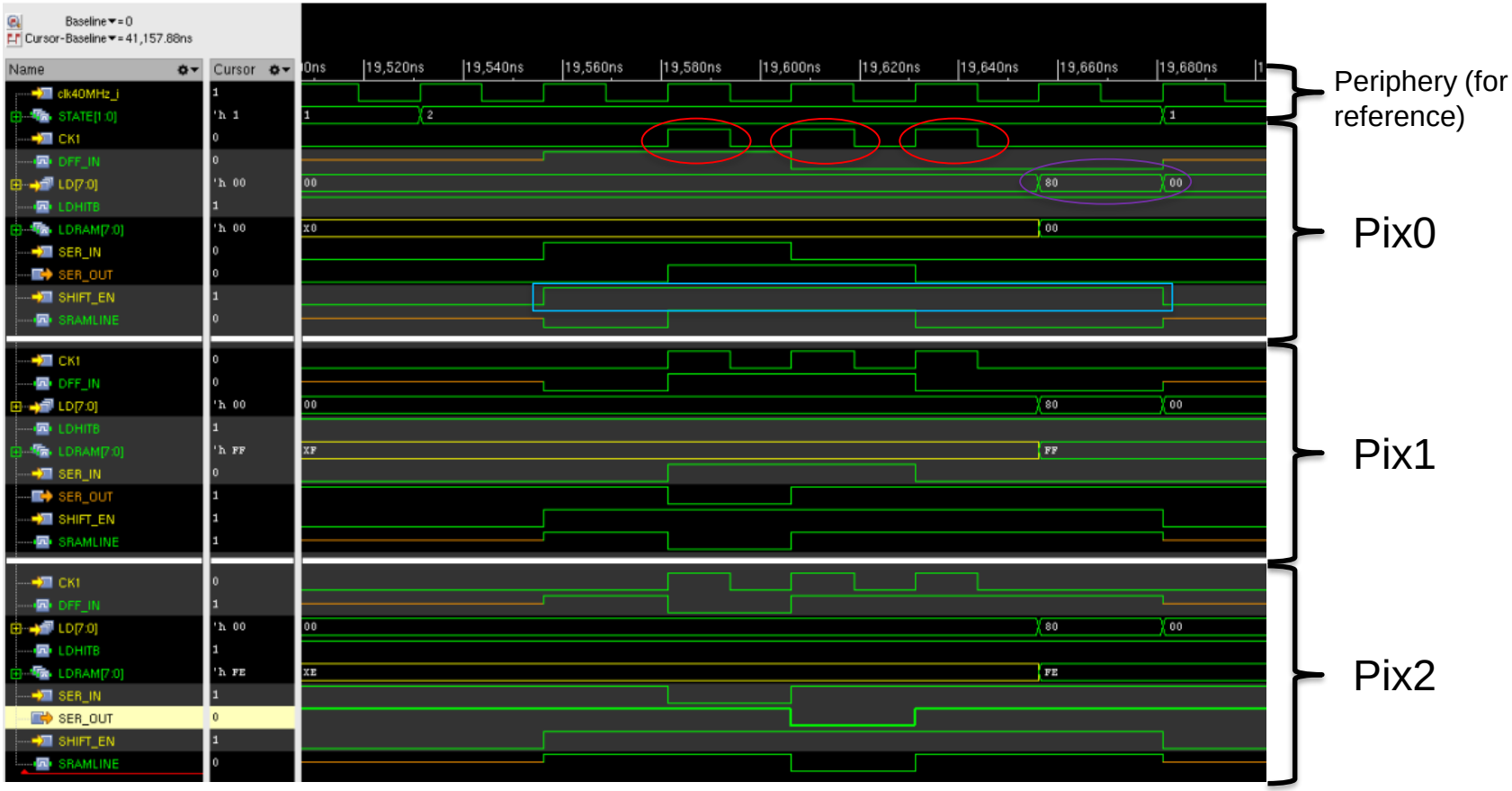
Status Digital Design

General comments

- Yellow = Inputs of the module
- Orange = Output of the module
- Green = Internal signals of the module

- Scripts available to produce these curves with 1 command
- Clock-gating cell from PDK used
  - *$LF15A_HOME* variable must be set (done during installing the PDK)
  - PDK library needs to be compiled (script available)

Shifting-WRITE Pixel



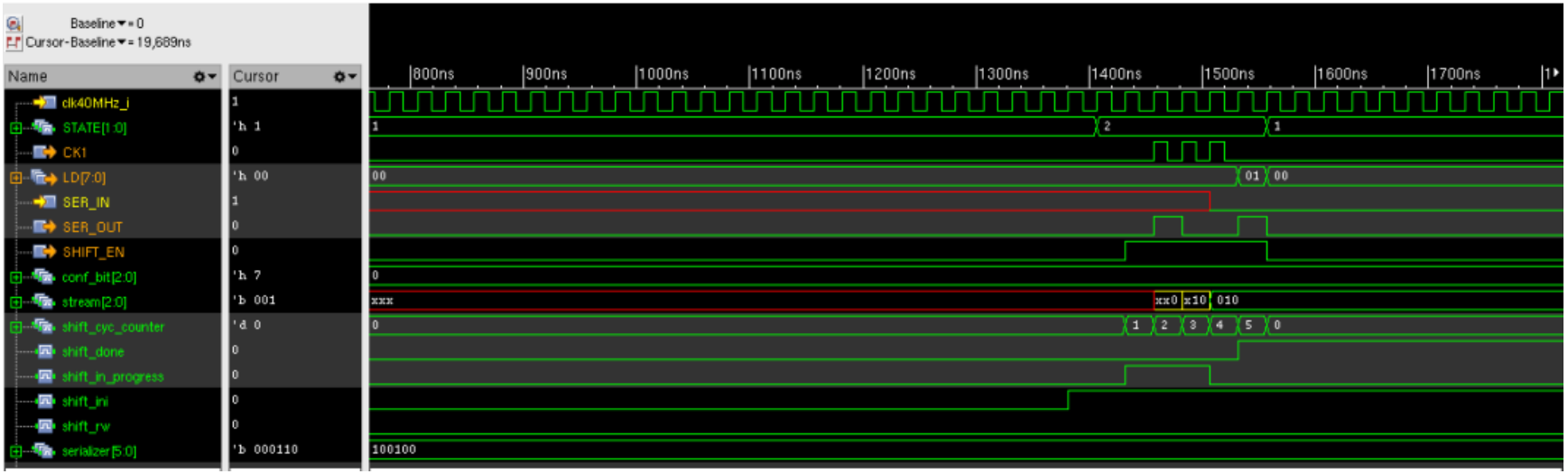
- 3 Clks to shift data in pixel
  - *SHIFT_EN* high
- *LD* Signal to write into SRAMs
  - *SHIFT_EN* **high for writing** to the SRAMs

READ Pixel

- *LD* Signal to write into SRAMs
  - *SHIFT_EN* **low for reading** to the SRAMs
- 3 Clks to shift data in pixel
  - *SHIFT_EN* high

Shift-WRITE EOC

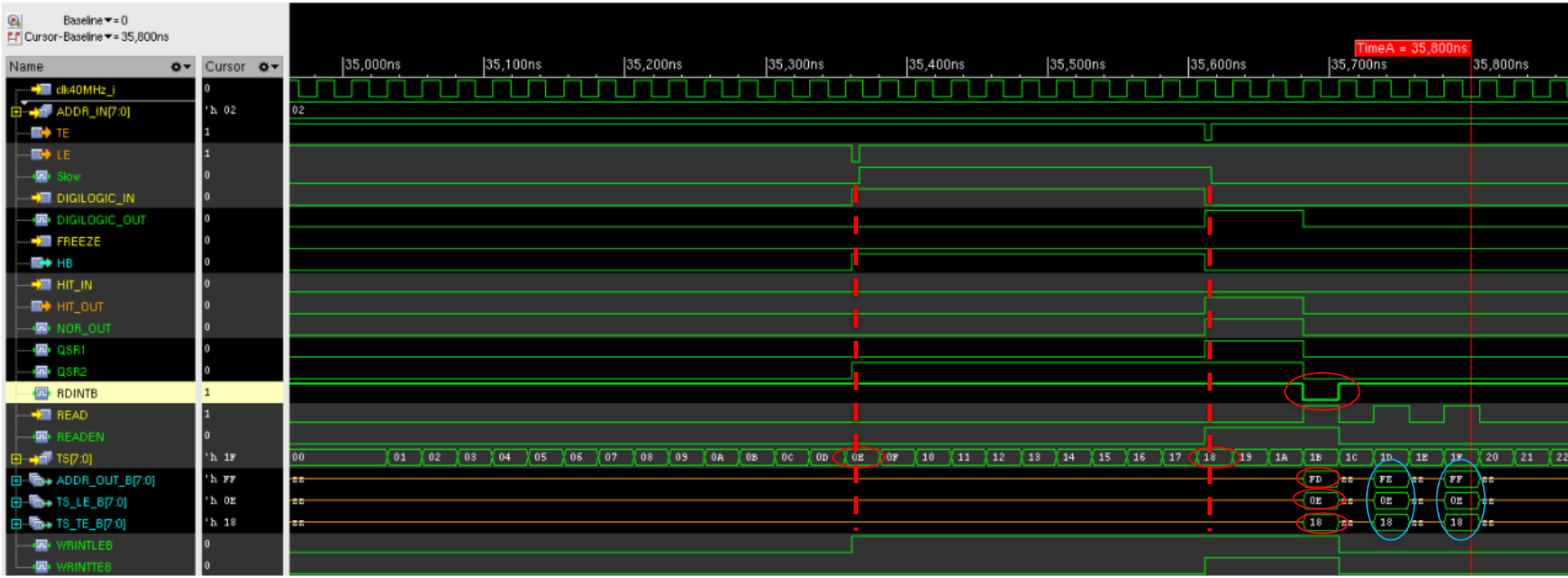
EOC0



- 6 clocks for 1 cycle:
  - 1 for recognizing *STATE*
  - 3 for shifting to/from the 3 pixels
  - 1 before shifting, to assert *LD* (Read only)
  - 1 after shifting, to assert *LD* (write only)

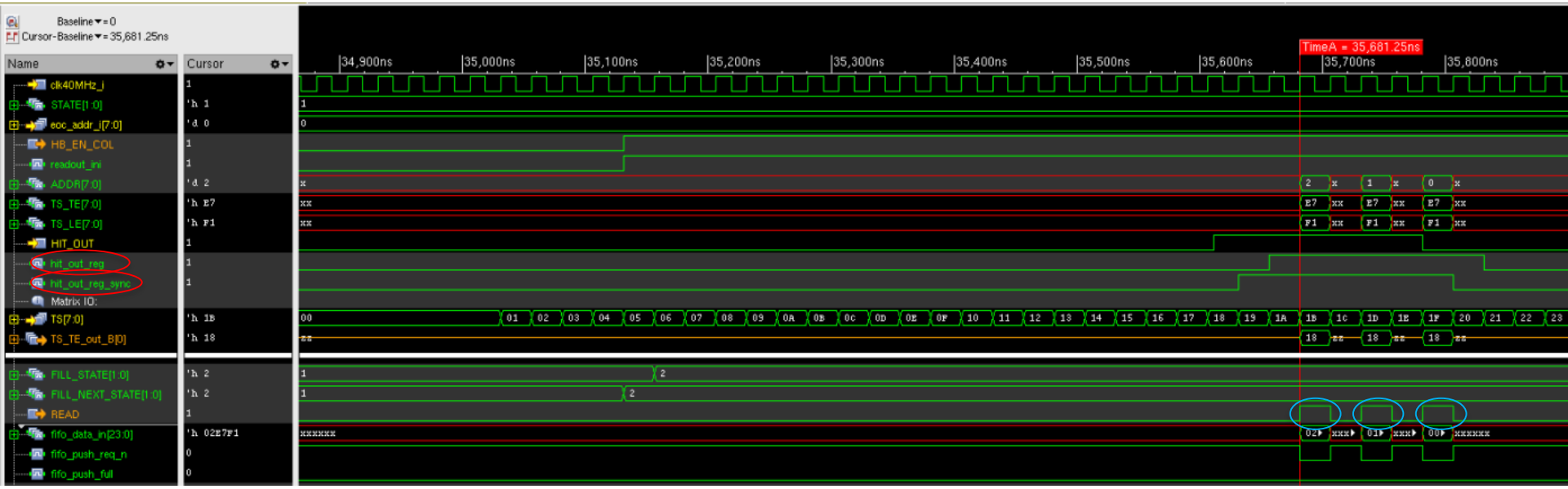
Recording Hits - Pixel

Pix0



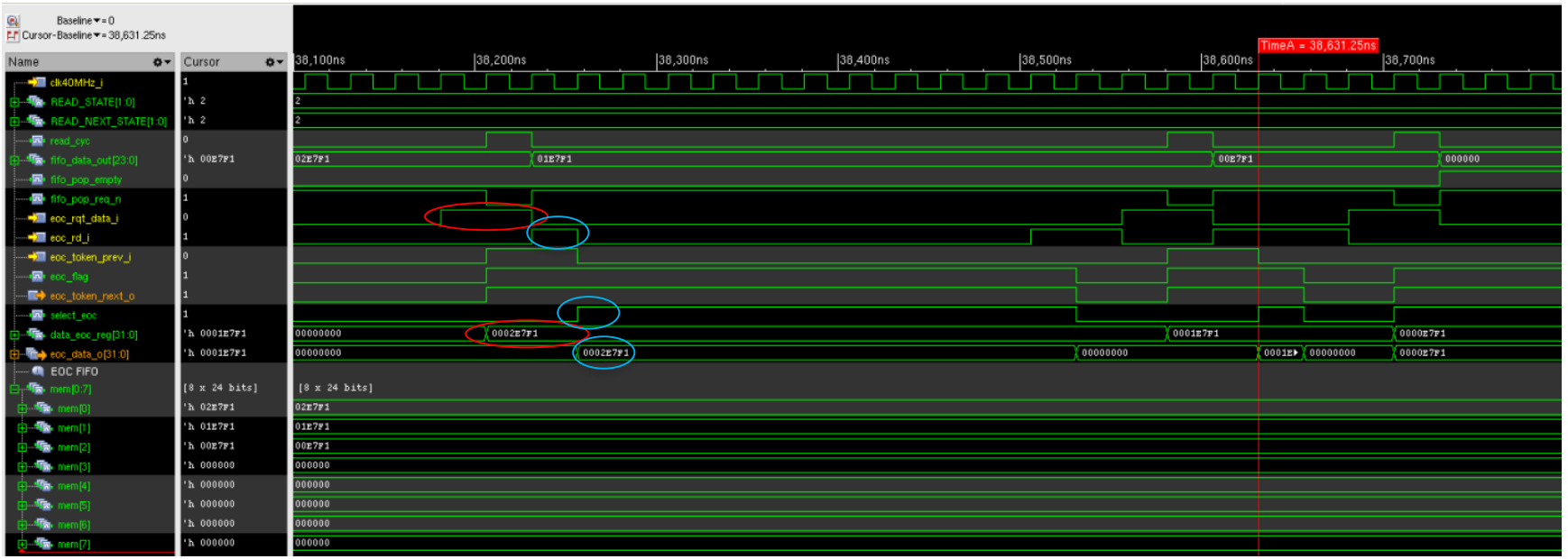
- Data at output bus only when ***RDINTB* is low**. (Only for 1 clk)
- Other values: Signals from different pixels.
- One clk pause, needed to reset the pixel after readout -> Cannot be done faster. (Limitation by the architecture)

Recording Hits - EOC



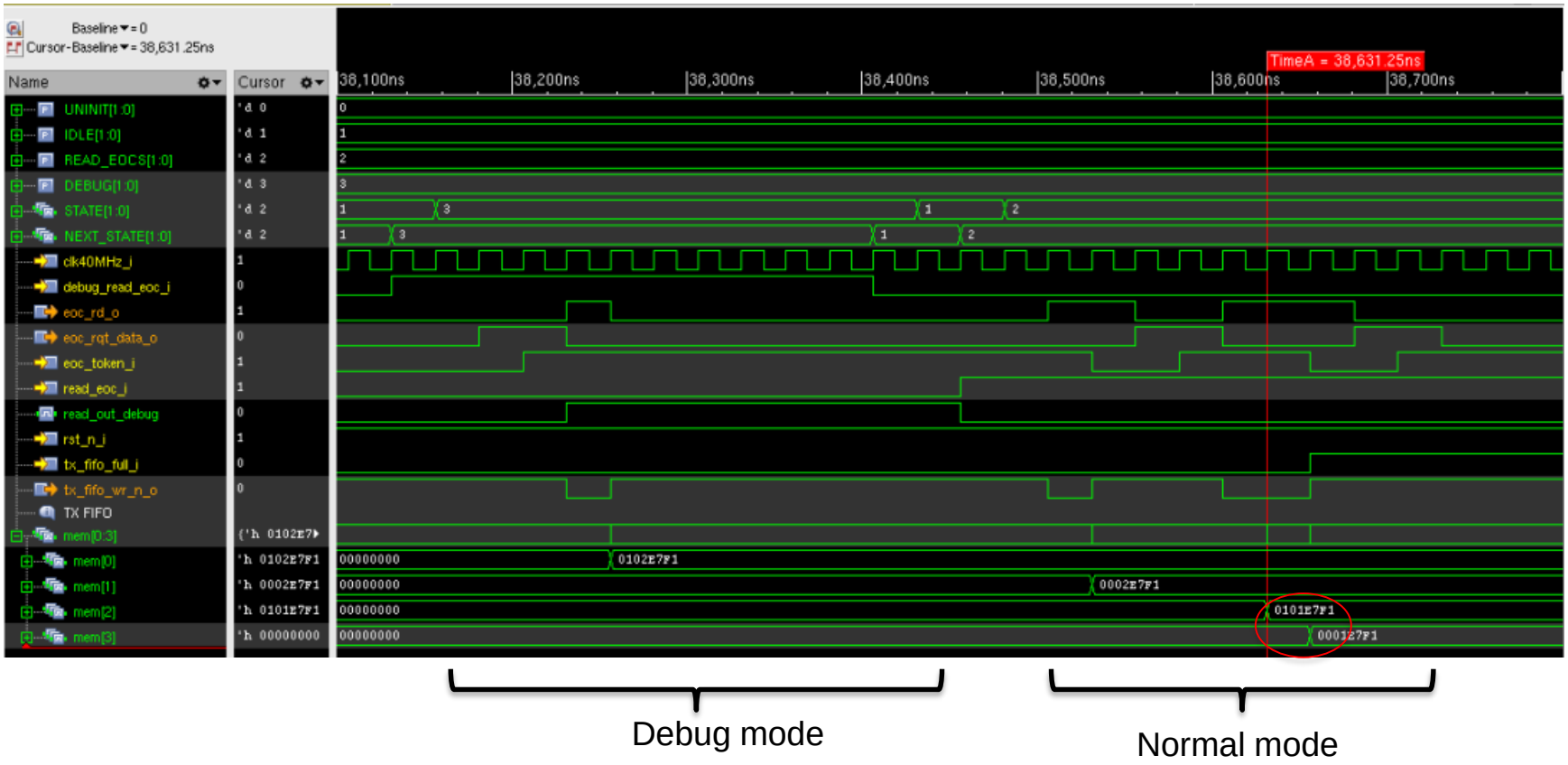
- 2 DFFs serve as synchronizer
  - *HIT_OUT* is <u>not</u> synchronous
- *READ* starts to toggle once a hit is registered in 2. DFF

Filling TX FIFO - EOC



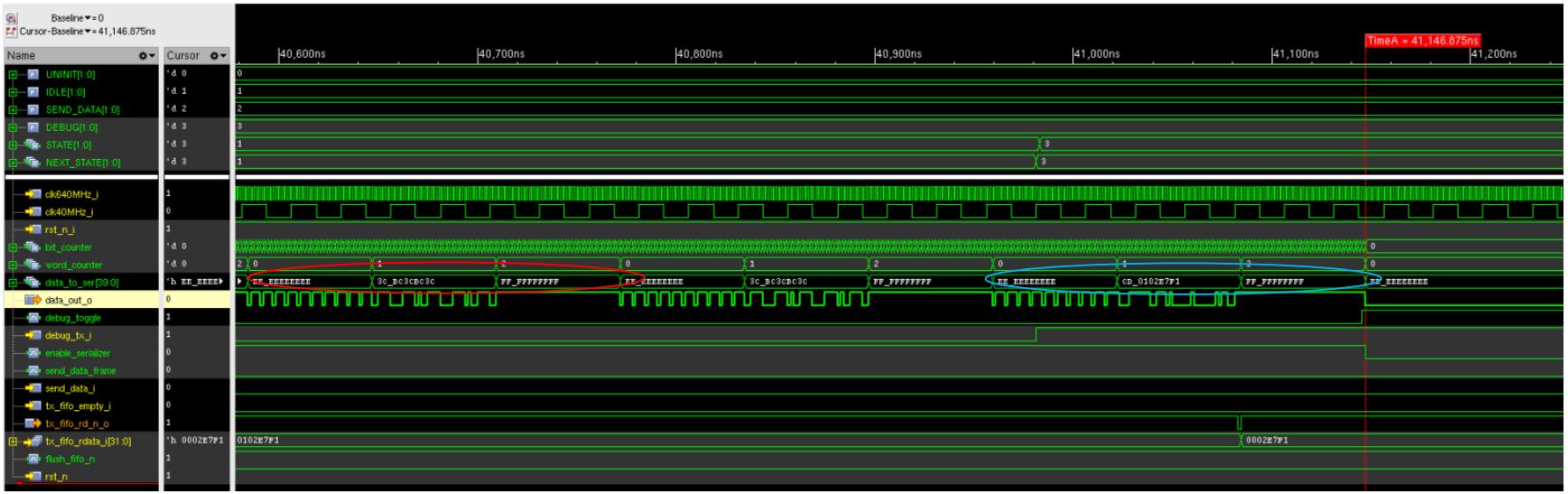
- EOCs take care of token handling
- *eoc_rqt_data* puts data into *data_out_reg*
  - Active, until there is at least 1 piece of data in 1 EOC
- data on output bus (*eoc_data_out*) only when token + *eoc_read* in previous cycle

Filling TX FIFO - CU



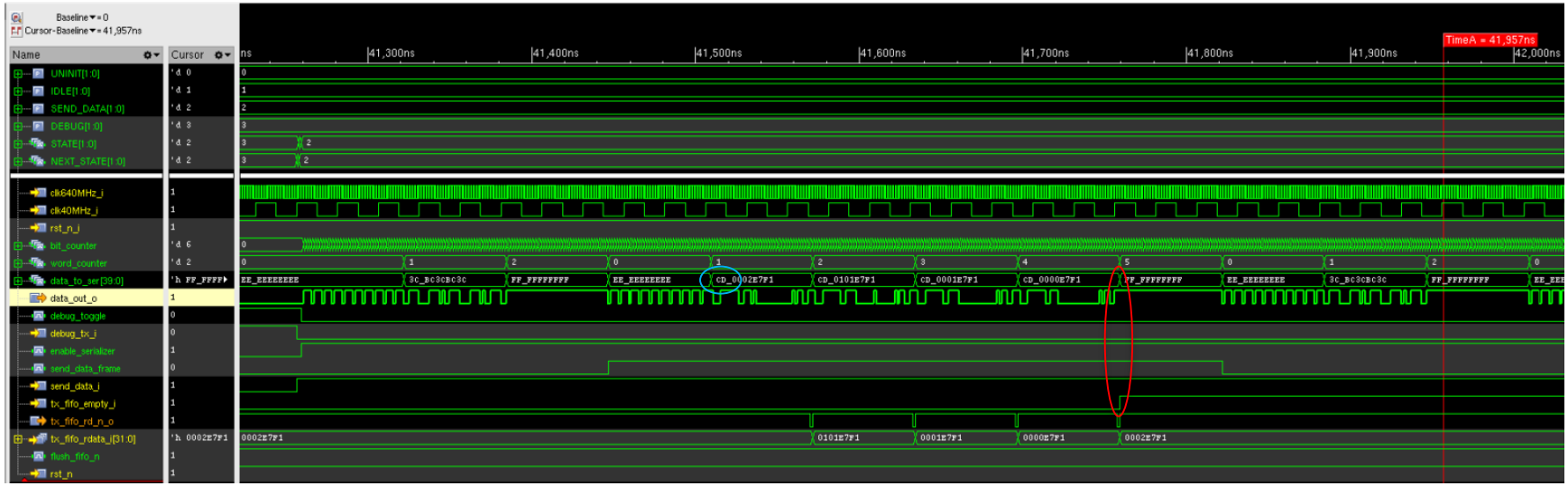
- In debug mode, only 1 piece of data is sent to TX-FIFO
  - Keeps the data on the output bus, until next readout cycle starts
- I normal mode, 1 piece of data every clock for maximum speed

Data readout – IDLE & DEBUG



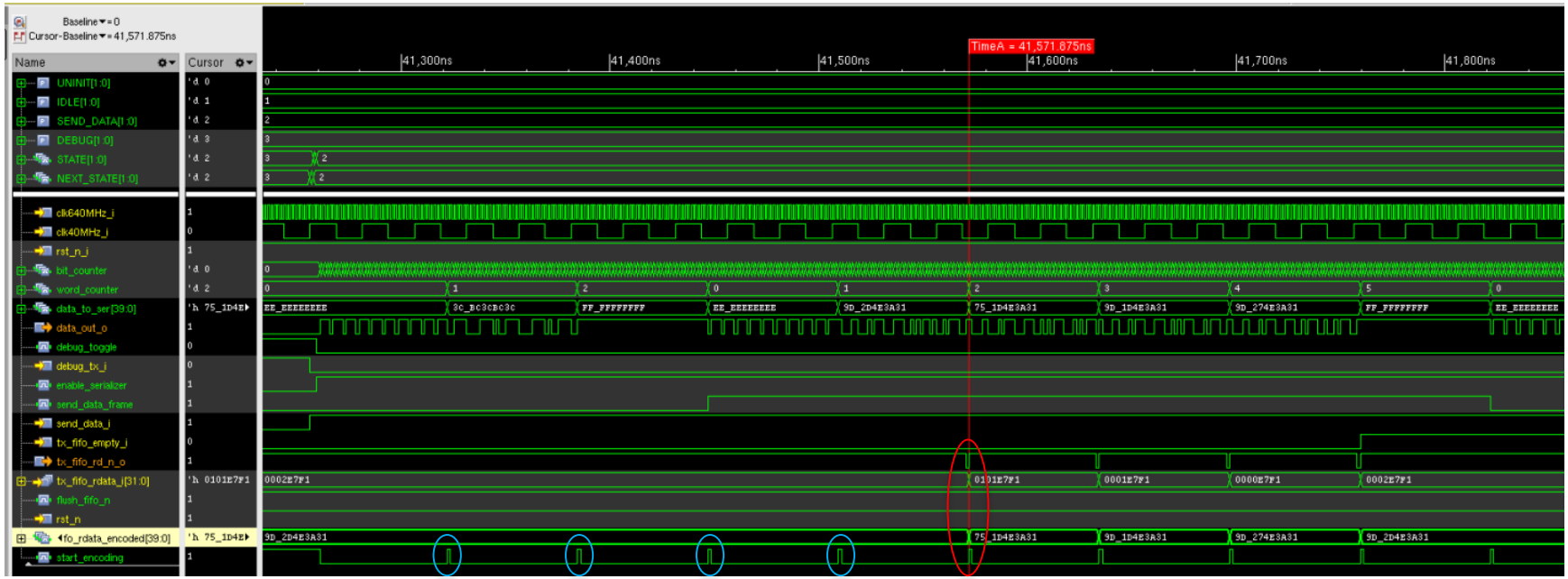
- One word is 40 bits
  - SOF: 10 bytes with EE
  - EOF: 10 bytes with FF
  - IDLE: 3C and BC alternating (3 times).
    - Kommas from 8bit10encoding
- In DEBUG mode, only one frame is sent (= data word + SOF + EOF)

Data readout – DATA



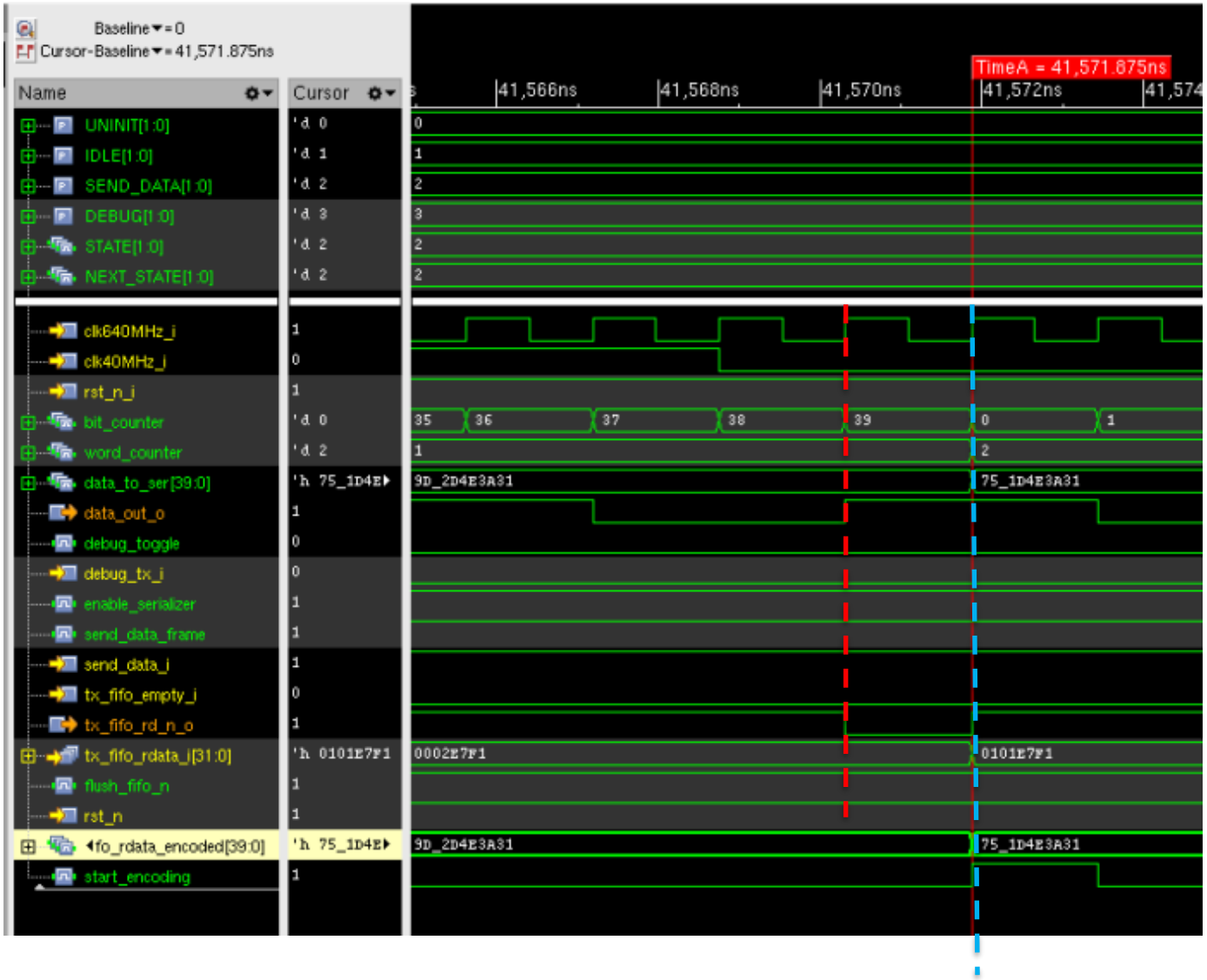
- In normal readout mode, the length of the frame is not determined.
  - Frame ends when FIFO is empty OR
  - Maximum frame size (= one full FIFO) is reached
- If no data, IDLE frame is sent
- Flag, which control which word is sent next are checked 1 clk (of the 640MHz) earlier, to guarantee no additional clks in the transmission
- Encoding: Here, ‘Encoding’ means simply attaching ‘CD’ -> much easier for understanding

Encoding



- Chipware IP from Cadence used
- Encoding takes 1 clk -> data is requested + encoded at end of every frame
  - Adds some unnecessary encoding
    - Still better the encoding at every clk edge

Encoding - Details



- New data requested at this edge (takes 1 edge)
- Encoded at this edge

Outlook

- Write something similar to this presentation into Documentation

- Add readout mode with FREEZE signal
- Convert TS to gray counter
- Some minor changes on Code missing (MUXing SER_OUT,… -> issues listed in gitlab)

- I2C – Wishbone currently checked by Jose
- Cross-check waveforms with Nissar
- Start with synthesis (~ 1-2 weeks from now)