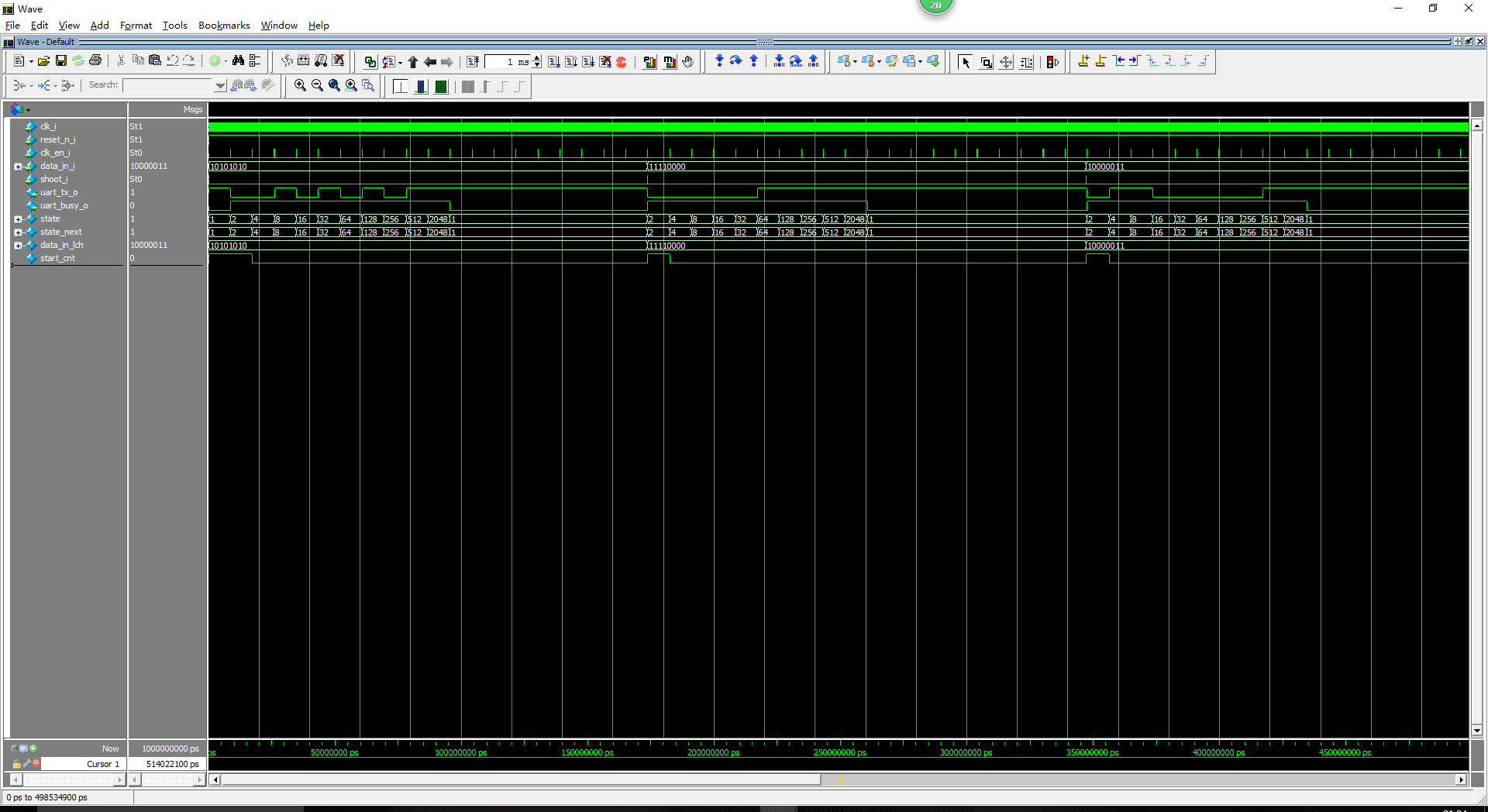
# uart\_tx\_op Module

## Design Idea

There are 13 states in the state machine of uart tx module, but not all of these states will be traversed during the state transition which is determined by the input parameters of the module. The input data of data\_in\_i signal is firstly buffered in a register when the shoot\_i signal is valid, and then it will be shift out one after another during the sending-data states. The output signals are totally determined by combinational logic.

## Simulation Result

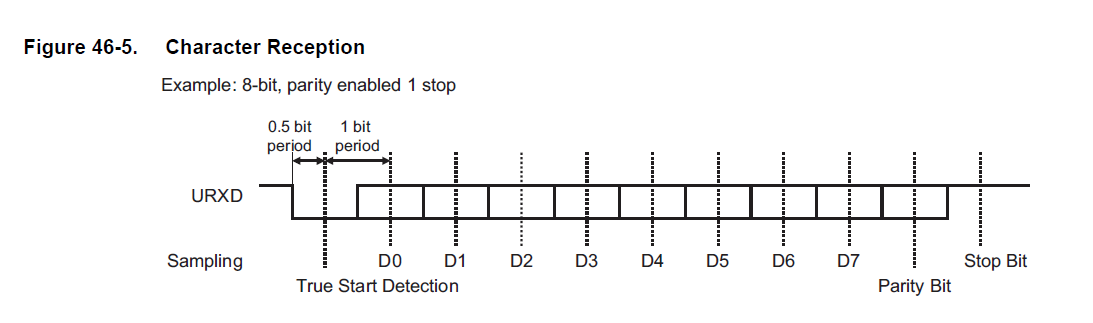


# uart\_rx\_op Module

## Design Idea

Besides the state machine of uart rx module, two extra modules are designed to corporate with this rx module.

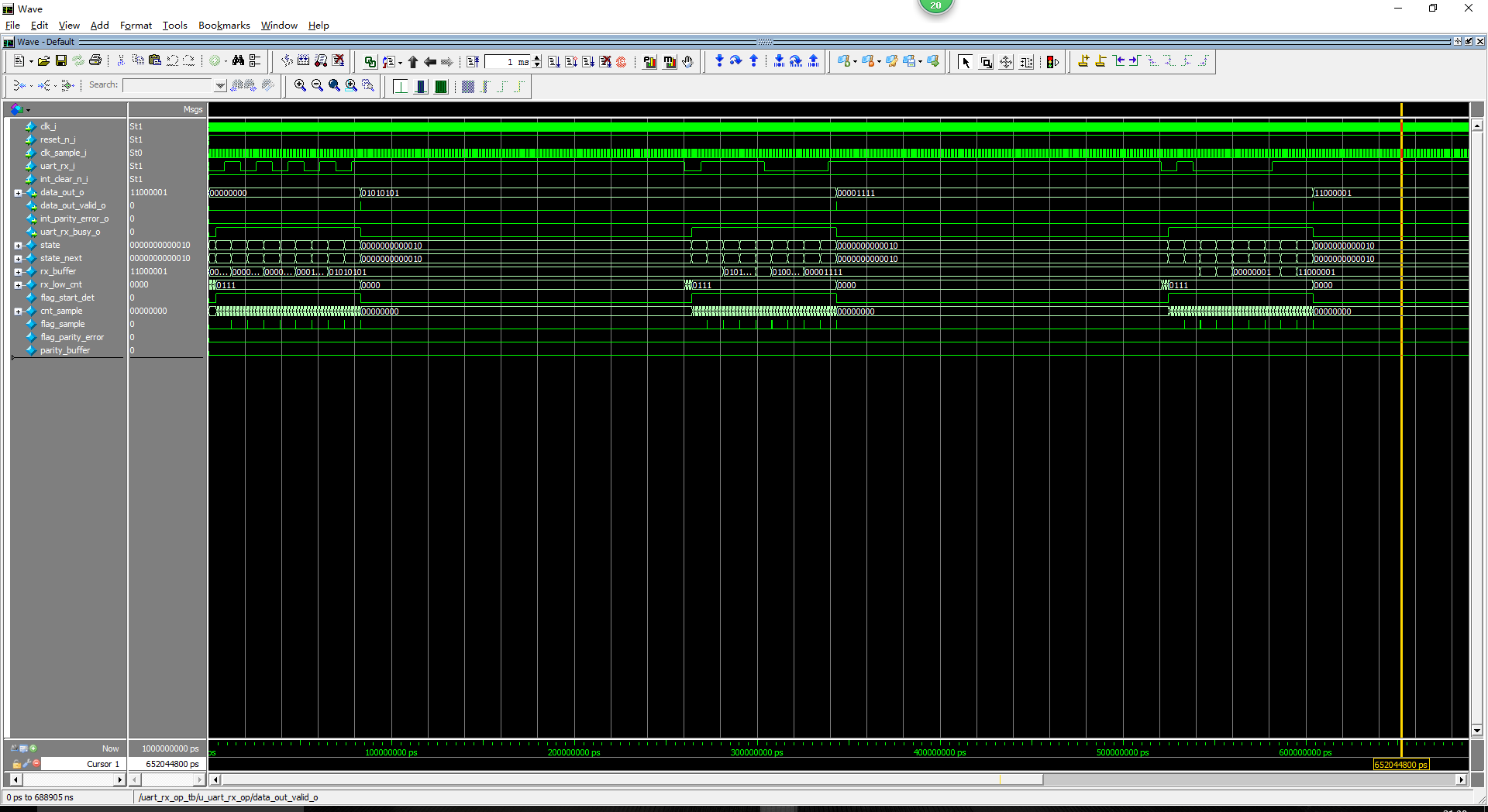
One module is used for start bit detection and spike filtering. Only when ( OVERSAMP\_RATE / 2 - 1 ) low rx input values have been sampled, this module will arbitrate the input signal as a start bit symbol, and assert the flag\_start\_det signal for state transition. The other module is used for sample pulse generation. When a start bit symbol has been detected, this module will generate sample pulses which is one clock cycle valid in the middle of each succeeding symbols, and all of these symbols will be sampled according to the sample pluses.



After all of the input symbols have been sampled, the output signals of the rx module will be handled in IDLE state, which is an intermediate state between start bit receiving and stop bit receiving. Parity bit will be check, and the interrupt will be set to high if any error exists. The data in the rx\_buffer register will be put on data\_out\_o signals, and the data\_out\_valid\_o signal will be assert for one clock cycle.

Frequency: clk\_sample\_i = OVERSAMP\_RATE \* frequency: clk\_i.

## Simulation Result



# rx\_package Module

## Design Idea

The main job of this module is to filter out desired GPS frames with specific preamble, and meanwhile substitute and pick certain number of bits in these frames.

Our state machine is designed for frame filtering, so several other modules need to corporate its work. The most important module is input FIFO module. This module will keep the rx data whenever the rx\_datavld\_i signal is set, and it will output the data of proper frames whenever the flag\_fifo\_out signal is set. All of the operations are performed based on write and read pointer of this FIFO.

There are also two FIFOs used for output ping-pong buffer, so if the following device could not handle the output frame immediately, it could access these two FIFOs to get the frame data. Besides the FIFO operation, several modules are designed to check the SOF condition and EOF condition, and flag signals are generated for state transition, like flag\_start\_det, flag\_sof\_det, flag\_eof\_det.

The substituting and picking operation for the selected frame are handled by two independent modules. Substituting of certain bits in selected frames is achieved by writing the corresponding contents into the input FIFO, so these contents will be sent out without any delay. Picking of certain bits in selected frames is performed by bit counting.

## Simulation Result

