**ReadMe\_Simple\_BS- ChatGPT**

1. "**cut\_bi\_phase**": This process is responsible for cutting the bi-phase filtered signal. When the "resetn" input is '0', indicating a reset condition, it sets the signals "sig\_bi\_phase\_filterd\_cut" and "sig\_bi\_phase\_filterd\_cut\_not" to '0' and '1', respectively. During the rising edge of the sysclk signal, it assigns the value of "bi\_phase\_filtered" to "sig\_bi\_phase\_filterd\_cut", and takes the logical negation of "sig\_bi\_phase\_filterd\_cut" and assigns it to "sig\_bi\_phase\_filterd\_cut\_not".
2. "**ena**": This process manages the enable signal "sig\_enable" based on the bi-phase filtered signal and the counter "cnt\_clk". When the "resetn" input is '0', it initializes "cnt\_clk" to all zeros and sets "sig\_enable" to '0'. On the rising edge of "sysclk", it increments the value of "cnt\_clk" by 1. When "cnt\_clk" reaches a count of 12288, it sets "sig\_enable" to '1'. If either "sig\_bi\_phase\_filterd\_r" or "sig\_bi\_phase\_filterd\_f" is '1' (indicating a rising or falling edge of the bi-phase filtered signal) and "sig\_enable" is '1', it resets "sig\_enable" to '0' and resets "cnt\_clk" back to all zeros.
3. "**clk\_00**": This process generates the clock signal "sig\_00\_clk" used during the '00' phase. If the "resetn" input is '0', it sets "sig\_00\_clk" to '1'. During the rising edge of "sysclk", it checks the value of "cnt\_clk". When "cnt\_clk" reaches a count of 8192, it sets "sig\_00\_clk" to '0'. If either "sig\_bi\_phase\_filterd\_r" or "sig\_bi\_phase\_filterd\_f" is '1' and "sig\_enable" is '1', it sets "sig\_00\_clk" to '1'.
4. "**cut\_clk\_00**": This process cuts the "sig\_00\_clk" signal. When "resetn" is '0', it sets "sig\_00\_cut" to '0' and "sig\_00\_cut\_not" to '1'. During the rising edge of "sysclk", it assigns the value of "sig\_00\_clk to sig\_00\_cut", and takes the logical negation of "sig\_00\_cut" and assigns it to "sig\_00\_cut\_not".
5. "**clk\_90**": This process generates the clock signal "sig\_90\_clk" used during the '90' phase. When the "resetn" input is '0', it sets "sig\_90\_clk" to '0'. During the rising edge of "sysclk", it checks the value of "cnt\_clk". When "cnt\_clk" reaches a count of 4096, it sets "sig\_90\_clk" to '1'. If "cnt\_clk" reaches a count of 12288, it sets "sig\_90\_clk" back to '0'.
6. "**cut\_clk\_90**": This process cuts the "sig\_90\_clk" signal. When resetn is '0', it sets "sig\_90\_cut" to '0' and "sig\_90\_cut\_not" to '1'. During the rising edge of "sysclk", it assigns the value of "sig\_90\_clk" to "sig\_90\_cut", and takes the logical negation of "sig\_90\_cut" and assigns it to "sig\_90\_cut\_not".
7. "**sample\_bs**": This process samples the bi-phase filtered signal ("bi\_phase\_filtered") during the '90' and '00' phases and stores the samples in flip-flops "sig\_ff\_A" and "sig\_ff\_B". When the "resetn" input is '0', it sets both "sig\_ff\_A" and "sig\_ff\_B" to '0'. On the rising edge of "sysclk", it checks the values of "sig\_90\_r" (indicating the rising edge of the '90' phase clock) and "sig\_90\_f" (indicating the falling edge of the '90' phase clock). If "sig\_90\_r" is '1' and "sig\_00\_clk" is '1', it assigns the value of "bi\_phase\_filtered" to "sig\_ff\_A". If "sig\_90\_f" is '1' and "sig\_00\_clk" is '0', it assigns the value of "bi\_phase\_filtered" to "sig\_ff\_B".
8. "**nrzl\_creation**": This process creates the NRZL (Non-Return-to-Zero-Level) data signal ("sig\_nrzl\_data"). When the "resetn" input is '0', it sets "sig\_nrzl\_data" to '0'. On the rising edge of "sysclk", it checks if 'sig\_00\_r" is '1' (indicating the rising edge of the '00' phase clock). If true, it calculates the XNOR (exclusive NOR) between "sig\_ff\_A" and "sig\_ff\_B" and assigns the result to "sig\_nrzl\_data".
9. Finally, the "nrzl\_data" output port is assigned the value of "sig\_nrzl\_data", and the "main\_clk" output port is assigned the value of "sig\_00\_clk".

Note that the code provided represents the architecture of the entity "Simple\_BS". It includes various processes that generate clock signals, sample, and manipulate signals based on these clocks, and generate the NRZL data signal. The purpose of this code is implementing a simple baseband modulation scheme using the bi-phase filtered input signal.