**ReadMe\_Rising\_Falling\_Edge**

(Entity) Inputs and Outputs have been defined (in: "resetn", "sysclk" /   
out: "clk\_6", "clk\_7", "signal\_A\_q", "signal\_A\_q\_not", "signal\_A\_r", "signal\_A\_f").  
  
(Architecture) New signals have been created: "sig\_counter", "sig\_Aclock\_q", "sig\_Aclock\_q\_not", "sig\_Araising", "sig\_Afaling".  
The "Rising\_Falling\_Edge" source fille contains 2 processes:  
In the first process named "clk\_bit\_6\_7" the outputs "clk\_6" and "clk\_7" was created.  
If "resetn" equals to '0' then "sig\_counter" equals to '0 0 0 0 0 0 0 0' (8-bits), else, every rising edge "sig\_counter" increased by 1 (change every 20nSec for frequency of 50MHZ). "sig\_counter (6)" (the 6 bit of "sig\_counter") is clk\_6 and "sig\_counter (7)" (the 7 bit of "sig\_counter") is clk\_7, bit number 6 changes every 64 ( ) bits and bit number 7 changes every 128 ( ) bits.  
"clk\_6" will change every 1280nSec ().   
"clk\_7" will change every 2560nSec ().  
   
In the second process named "raising\_falling" the signals "sig\_Aclock\_q" and "sig\_Aclock\_q\_not" was created. If "resetn" equals to '0' then "sig\_Aclock\_q" equals to '0' and "sig\_Aclock\_q\_not" equals to '1', else, every rising edge "sig\_Aclock\_q" is   
"sig\_counter (7)" (clk\_7) and "sig\_Aclock\_q\_not" is(**not** "sig\_Aclock\_q") **(logic operator)**.   
That mean the signals "sig\_Aclock\_q" and "sig\_Aclock\_q\_not" are reversed. After the second process the signal "sig\_Araising" equals to ("sig\_Aclock\_q" **AND** "sig\_Aclock\_q\_not") **(logic operator)**.   
That mean the signal "sig\_Araising" equals to '1' every rising edge of "sig\_Aclock\_q" because the execution of the operations is performed in parallel and when comes the operation that "sig\_Aclock\_q" equals to '1' the signal "sig\_Aclock\_q\_not" is(**not** "sig\_Aclock\_q") when "sig\_Aclock\_q" is still '0' and not '1' like he change in this particular operation, as a result "sig\_Araising" equals to ("sig\_Aclock\_q"(1) **AND** "sig\_Aclock\_q\_not"(1)) which equals to '1' for only one operation (20nSec pulse) because in the next operation "sig\_Aclock\_q\_not" will equal to '0' and as a result "sig\_Araising" will equal to '0'.

Additionally, the signal "sig\_Afaling" equals to (**not** ("sig\_Aclock\_q" **or** "sig\_Aclock\_q\_not")) **(logic operator)**. That mean the signal "sig\_Afaling" equals to '1' every falling edge of "sig\_Aclock\_q" because the execution of the operations is performed in parallel and when comes the operation that "sig\_Aclock\_q" equals to '0' the signal "sig\_Aclock\_q\_not" is(**not** "sig\_Aclock\_q") when "sig\_Aclock\_q" is still '1' and not '0' like he change in this particular operation, as a result "sig\_Afaling" equals to   
( **not** ( "sig\_Aclock\_q" (0) **or** "sig\_Aclock\_q\_not"(0) ) ) which equals to '1' for only one operation (20nSec pulse) because in the next operation "sig\_Aclock\_q\_not" will equal to '1' and as a result "sig\_Afaling" will equal to '0'.