**ReadMe\_Shift\_Register\_X**

(Entity) Inputs and Outputs have been defined (in: "resetn", "sysclk" /   
out: "shift\_reg\_out" (8-bit vector), "Afalling", "Araising", "q\_rom\_out" (8-bit vector), "Timing\_Pulse", "Address\_cnt" (6-bit vector)).  
  
(Architecture) First was defined the "Rom\_X" component that we created in other fille, this component contains 4 ports: (in: "address", "clock" / out: "q").   
New signals have been created: "sig\_shift\_reg\_out" (8-bit vector),   
"sig\_q\_rom\_out" (8-bit vector), "sig\_sysclk\_cnt" (8-bit vector), "sig\_Address\_cnt"   
(6-bit vector), "sig\_Aclock\_q", "sig\_Aclock\_q\_not", "sig\_Afalling", "sig\_Araising",   
"sig\_Timing\_Pulse".  
Note: this component is not part of the "Shift\_Register\_X" but it is useful for system control of the data transportation in the project.  
Port map was defined when "address" was mapped to "sig\_Address\_cnt", "clock" was mapped to "sysclk" and "q" was mapped to "sig\_q\_rom\_out".  
This operation uses every "Rom\_X" outputs and inputs as new signals or I/O of the source. As a result, we can use the data in the rom registers for "Shift\_Register\_X" source.  
   
The "Shift\_Register\_X" source fille contains 4 processes:   
In the first process named "raising\_falling" the signals "sig\_Aclock\_q" and "sig\_Aclock\_q\_not" are created and with them outputs "sig\_Araising" and "sig\_Afaling" were created (read "ReadMe\_Rising\_Falling\_Edge").   
  
In the second process named "address\_counter" the signal "sig\_Address\_cnt" was created to set to output "Address\_cnt". If "resetn" equals to '0' then "sig\_Address\_cnt" equals to '1 1 1 1 1 1' (6-bits vector) because we want to start from address '0 0 0 0 0 0', else, every rising edge of "sysclk" there is condition " if "sig\_Araising" = '1' " then "sig\_Address\_cnt" increased by 1 (like this "sig\_Address\_cnt" go through all the addresses). Because every "sig\_Araising" = '1' there is advance in the address and every "sig\_Afalling" = '1' there is a reading of the content of the address and because "sig\_Araising" comes before "sig\_Afalling" the reading reads in the second address and not from the first, that why we started from "sig\_Address\_cnt" equals to '1 1 1 1 1 1' (6-bits vector) to read the first address. The signal "sig\_Afalling" reads the content of the addresses because we set this signal in the port map of the Rom\_X component that ("rden" **=>** "sig\_Afalling**")**, when "rden" equals to '1' it reads the content of the addresses and so when "sig\_Afalling" equals to '1' it reads the content of the addresses too.  
   
  
  
  
In the third process named "shift\_register" the signal "sig\_shift\_reg\_out" was created to set to output "shift\_reg\_out". If "resetn" equals to '0' then " sig\_shift\_reg\_out " equals to '0 0 0 0 0 0 0 0' (8-bit vector), else, every rising edge of "sysclk" there is condition   
" if "sig\_Araising" = '1' " then " sig\_shift\_reg\_out" equals to the signal "sig\_q\_rom\_out" when this signal equals to the content of the current address because in the port map of the Rom\_X component that ("q" **=>** "sig\_q\_rom\_out")and "q" indicates the content of the current address. So, every "sig\_Araising" equals to '1' "sig\_q\_rom\_out" will be equal to one of the address data in the "Rom\_X", else, we will shift 1 register to left with '0' enter from the left like this '1 1 1 1 1 1 1 1' 🡺 '1 1 1 1 1 1 1 0' (the 7 bit gone).   
  
In the fourth process named "tim\_pulse" the signal "sig\_Timing\_Pulse" was created to set to output "Timing\_Pulse". If "resetn" equals to '0' then "sig\_Timing\_Pulse" equals to '0', else, every rising edge of "sysclk" there is condition   
" if "sig\_shift\_reg\_out(3)" = '0' AND "sig\_shift\_reg\_out(4)" = '1' " then "sig\_Timing\_Pulse" equals to '1', else, "sig\_Timing\_Pulse" equals to '0'. Like this pulse was created from the shift register operation till we fulfill this condition when the 3 bit is '0' and the 4 bit is '1'.

.