---VME subaddress decoder---

```
assign version_number_selected          =       (areg=='h01) & vme_pulse_int;
assign router_control_reg_selected      =       (areg=='h02) & vme_pulse_int;
assign router_status_selected1        =       (areg=='h03) & vme_pulse_int;
assign router_status_selected2        =       (areg=='h04) & vme_pulse_int;
assign router_status_selected3        =       (areg=='h05) & vme_pulse_int;

//assign data_register_selected       =       (areg=='h02) & vme_pulse_int;
//assign address_register_selected    =       (areg=='h03) & vme_pulse_int;
//assign  direct_access_linkrx_selected =       (areg=='h4) & vme_pulse_int;
//assign  router_dpm_selected          =       (areg=='h05) & vme_pulse_int;          // areg are bits [23:16]
//assign  router_spm_selected          =       (areg=='h06) & vme_pulse_int;          // areg are bits [23:21]
assign fee_status_word_for_DDL_selected =       (areg=='h07) & vme_pulse_int;
assign reset_detector_part_electronics  =       (areg=='h09) & vme_pulse_int;
assign read_start_address_selected  =           (areg=='h0a) & vme_pulse_int;
assign read_lenght_of_block_selected            =           (areg=='h0b) & vme_pulse_int;
assign write_histogram                =       (areg=='h0c) & vme_pulse_int;
assign read_histogram                 =       (areg=='h0d) & vme_pulse_int;
assign reset_ttcrx                    =           (areg=='h0e) & vme_pulse_int;
assign reset_linkrx                               =           (areg=='h0f) & vme_pulse_int;
assign send_sequence_tpl1l2a          =           (areg=='h10) & vme_pulse_int;
assign read_address_counter_fifo_sa             =           (areg=='h11) & vme_pulse_int;
assign flash_dpm                        =         (areg=='h12) & vme_pulse_int;
assign read_l1_id                     =           (areg=='h13) & vme_pulse_int;
assign read_number_of_fastor            =       (areg=='h14) & vme_pulse_int;
//assign router_jamplayer_register      =       (areg=='h15) & vme_pulse_int;  // in vme_fpga !!!
assign read_temp_vme_ch0                =       (areg=='h16) & vme_pulse_int;
assign read_temp_vme_ch1                =       (areg=='h17) & vme_pulse_int;
assign read_temp_vme_ch2                =       (areg=='h18) & vme_pulse_int;
assign read_temp_vme_ch3                =       (areg=='h19) & vme_pulse_int;
assign read_temp_vme_ch4                =       (areg=='h1a) & vme_pulse_int;
assign read_temp_vme_ch5                =       (areg=='h1b) & vme_pulse_int;
assign router_temp_limitMCM_ch0_selected  =       (areg=='h1c) & vme_pulse_int;
assign router_temp_limitMCM_ch1_selected  =       (areg=='h1d) & vme_pulse_int;
assign router_temp_limitMCM_ch2_selected  =       (areg=='h1e) & vme_pulse_int;
assign router_temp_limitMCM_ch3_selected  =       (areg=='h1f) & vme_pulse_int;
assign router_temp_limitMCM_ch4_selected  =       (areg=='h20) & vme_pulse_int;
assign router_temp_limitMCM_ch5_selected  =       (areg=='h21) & vme_pulse_int;
assign read_address_counter_fifo_ea             =           (areg=='h22) & vme_pulse_int;
//assign  vme_rst                       =       (areg=='h23) & vme_pulse_int; // in vme_fpga !!!
assign router_temp_limitBUS_ch0_selected   =       (areg=='h24) & vme_pulse_int;
assign router_temp_limitBUS_ch1_selected   =       (areg=='h25) & vme_pulse_int;
assign router_temp_limitBUS_ch2_selected   =       (areg=='h26) & vme_pulse_int;
assign router_temp_limitBUS_ch3_selected   =       (areg=='h27) & vme_pulse_int;
assign router_temp_limitBUS_ch4_selected   =       (areg=='h28) & vme_pulse_int;
assign router_temp_limitBUS_ch5_selected   =       (areg=='h29) & vme_pulse_int;
assign irq_push_button                =       (areg=='h2a) & vme_pulse_int;
assign read_l0_id                       =           (areg=='h2b) & vme_pulse_int;
assign read_l2a_id                                =       (areg=='h2c) & vme_pulse_int;
assign reset_bcnt                       =       (areg=='h2d) & vme_pulse_int;
assign write_router_header_word_selected =       (areg=='h2e) & vme_pulse_int;
assign router_header_fifo_reset           =       (areg=='h2f) & vme_pulse_int;
```

```
// jtag controller_0
assign jtag_data_in_selected_0                  =       (areg=='h31) & vme_pulse_int & ~vme_dir_int;
assign jtag_data_out_selected_0                 =       (areg=='h31) & vme_pulse_int & vme_dir_int;
assign write_enable_in_fifo_0           =       jtag_data_in_selected_0 | (common_write_enable_in_fifos &
jtagch0_common);
assign read_enable_out_fifo_0                   =       jtag_data_out_selected_0;
assign reset_jtag_controller_0                  =       ((areg=='h32) & vme_pulse_int) |
(common_reset_jtag_controllers & jtagch0_common);
assign reset_jtag_fifos_0               =       ((areg=='h33) & vme_pulse_int) | (common_reset_jtag_fifos &
jtagch0_common);
assign execution_start_jtag_controller_0=       ((areg=='h34) & vme_pulse_int & ~vme_dir_int) |
                                                        (common_execution_start_jtag_controllers &
jtagch0_common);
assign status_register_jtag_selected_0  =       (areg=='h35) & vme_pulse_int;
assign reset_jtag_ch_0                  =     ~(((areg=='h36) & vme_pulse_int) | (common_reset_jtag_channels
& jtagch0_common));
assign read_enable_in_fifo_0_vme  =       (areg=='h37) & vme_pulse_int;
assign read_current_number_of_in_fifo_0    =       (areg=='h38) & vme_pulse_int;
// jtag controller_1
assign jtag_data_in_selected_1                  =       (areg=='h39) & vme_pulse_int & ~vme_dir_int;
assign jtag_data_out_selected_1                 =       (areg=='h39) & vme_pulse_int & vme_dir_int;
assign write_enable_in_fifo_1           =       jtag_data_in_selected_1 | (common_write_enable_in_fifos &
jtagch1_common);
assign read_enable_out_fifo_1                   =       jtag_data_out_selected_1;
assign reset_jtag_controller_1                  =       ((areg=='h3a) & vme_pulse_int) |
(common_reset_jtag_controllers & jtagch1_common);
assign reset_jtag_fifos_1                       =       ((areg=='h3b) & vme_pulse_int) |
(common_reset_jtag_fifos & jtagch1_common);
assign execution_start_jtag_controller_1=       ((areg=='h3c) & vme_pulse_int & ~vme_dir_int) |
                                                        (common_execution_start_jtag_controllers &
jtagch1_common);
assign status_register_jtag_selected_1  =       (areg=='h3d) & vme_pulse_int;
assign reset_jtag_ch_1                  =     ~(((areg=='h3e) & vme_pulse_int) | (common_reset_jtag_channels
& jtagch1_common));
assign read_enable_in_fifo_1_vme  =       (areg=='h3f) & vme_pulse_int;
assign read_current_number_of_in_fifo_1    =       (areg=='h40) & vme_pulse_int;
// jtag controller_2
assign jtag_data_in_selected_2                  =       (areg=='h41) & vme_pulse_int & ~vme_dir_int;
assign jtag_data_out_selected_2                 =       (areg=='h41) & vme_pulse_int & vme_dir_int;
assign write_enable_in_fifo_2           =       jtag_data_in_selected_2 | (common_write_enable_in_fifos &
jtagch2_common);
assign read_enable_out_fifo_2                   =       jtag_data_out_selected_2;
assign reset_jtag_controller_2                  =       ((areg=='h42) & vme_pulse_int) |
(common_reset_jtag_controllers & jtagch2_common);
assign reset_jtag_fifos_2                       =       ((areg=='h43) & vme_pulse_int) |
(common_reset_jtag_fifos & jtagch2_common);
assign execution_start_jtag_controller_2=       ((areg=='h44) & vme_pulse_int & ~vme_dir_int) |
                                                        (common_execution_start_jtag_controllers &
jtagch2_common);
assign status_register_jtag_selected_2  =       (areg=='h45) & vme_pulse_int;
assign reset_jtag_ch_2                  =     ~(((areg=='h46) & vme_pulse_int) | (common_reset_jtag_channels
& jtagch2_common));
assign read_enable_in_fifo_2_vme  =       (areg=='h47) & vme_pulse_int;
assign read_current_number_of_in_fifo_2    =       (areg=='h48) & vme_pulse_int;
// jtag controller_3
assign jtag_data_in_selected_3                  =       (areg=='h49) & vme_pulse_int & ~vme_dir_int;
assign jtag_data_out_selected_3              =     (areg=='h49) & vme_pulse_int & vme_dir_int;
assign write_enable_in_fifo_3        =     jtag_data_in_selected_3 | (common_write_enable_in_fifos &
jtagch3_common);
assign read_enable_out_fifo_3                =     jtag_data_out_selected_3;
assign reset_jtag_controller_3               =     ((areg=='h4a) & vme_pulse_int) |
(common_reset_jtag_controllers & jtagch3_common);
assign reset_jtag_fifos_3                    =     ((areg=='h4b) & vme_pulse_int) |
(common_reset_jtag_fifos & jtagch3_common);
assign execution_start_jtag_controller_3=     ((areg=='h4c) & vme_pulse_int & ~vme_dir_int) |
                                                   (common_execution_start_jtag_controllers &
jtagch3_common);
assign status_register_jtag_selected_3  =     (areg=='h4d) & vme_pulse_int;
assign reset_jtag_ch_3               =    ~(((areg=='h4e) & vme_pulse_int) | (common_reset_jtag_channels
& jtagch3_common));
assign read_enable_in_fifo_3_vme  =     (areg=='h4f) & vme_pulse_int;
assign read_current_number_of_in_fifo_3    =     (areg=='h50) & vme_pulse_int;
// jtag controller_4
assign jtag_data_in_selected_4               =     (areg=='h51) & vme_pulse_int & ~vme_dir_int;
assign jtag_data_out_selected_4              =     (areg=='h51) & vme_pulse_int & vme_dir_int;
assign write_enable_in_fifo_4        =     jtag_data_in_selected_4 | (common_write_enable_in_fifos &
jtagch4_common);
assign read_enable_out_fifo_4                =     jtag_data_out_selected_4;
assign reset_jtag_controller_4               =     ((areg=='h52) & vme_pulse_int) |
(common_reset_jtag_controllers & jtagch4_common);
assign reset_jtag_fifos_4                    =     ((areg=='h53) & vme_pulse_int) |
(common_reset_jtag_fifos & jtagch4_common);
assign execution_start_jtag_controller_4=     ((areg=='h54) & vme_pulse_int & ~vme_dir_int) |
                                                   (common_execution_start_jtag_controllers &
jtagch4_common);
assign status_register_jtag_selected_4  =     (areg=='h55) & vme_pulse_int;
assign reset_jtag_ch_4               =    ~(((areg=='h56) & vme_pulse_int) | (common_reset_jtag_channels
& jtagch4_common));
assign read_enable_in_fifo_4_vme  =     (areg=='h57) & vme_pulse_int;
assign read_current_number_of_in_fifo_4    =     (areg=='h58) & vme_pulse_int;
// jtag controller_5
assign jtag_data_in_selected_5               =     (areg=='h59) & vme_pulse_int & ~vme_dir_int;
assign jtag_data_out_selected_5              =     (areg=='h59) & vme_pulse_int & vme_dir_int;
assign write_enable_in_fifo_5        =     jtag_data_in_selected_5 | (common_write_enable_in_fifos &
jtagch5_common);
assign read_enable_out_fifo_5                =     jtag_data_out_selected_5;
assign reset_jtag_controller_5               =     ((areg=='h5a) & vme_pulse_int) |
(common_reset_jtag_controllers & jtagch5_common);
assign reset_jtag_fifos_5                    =     ((areg=='h5b) & vme_pulse_int) |
(common_reset_jtag_fifos & jtagch5_common);
assign execution_start_jtag_controller_5=     ((areg=='h5c) & vme_pulse_int & ~vme_dir_int) |
                                                   (common_execution_start_jtag_controllers &
jtagch5_common);
assign status_register_jtag_selected_5  =     (areg=='h5d) & vme_pulse_int;
assign reset_jtag_ch_5               =    ~(((areg=='h5e) & vme_pulse_int) | (common_reset_jtag_channels
& jtagch5_common));
assign read_enable_in_fifo_5_vme  =     (areg=='h5f) & vme_pulse_int;
assign read_current_number_of_in_fifo_5    =     (areg=='h60) & vme_pulse_int;
// common mode for jtag controllers
assign common_reset_jtag_controllers    =     (areg=='h61) & vme_pulse_int;
assign common_reset_jtag_fifos         =     (areg=='h62) & vme_pulse_int;
assign common_reset_jtag_channels      =     (areg=='h63) & vme_pulse_int;
assign common_write_enable_in_fifos     =       (areg=='h64) & vme_pulse_int;
assign common_execution_start_jtag_controllers= (areg=='h65) & vme_pulse_int;
// resets for half sectors
assign reset_half_stave_ch0 = (areg=='h66) & vme_pulse_int;
assign reset_half_stave_ch1 = (areg=='h67) & vme_pulse_int;
assign reset_half_stave_ch2 = (areg=='h68) & vme_pulse_int;
assign reset_half_stave_ch3 = (areg=='h69) & vme_pulse_int;
assign reset_half_stave_ch4 = (areg=='h6a) & vme_pulse_int;
assign reset_half_stave_ch5 = (areg=='h6b) & vme_pulse_int;
// fastor registers
assign fastor_normal_reg_selected = (areg=='h6c) & vme_pulse_int;
assign fastor_coincidence_reg_selected = (areg=='h6d) & vme_pulse_int;
assign fastor_linkrx1_reg_selected = (areg=='h6e) & vme_pulse_int;
assign fastor_linkrx2_reg_selected = (areg=='h6f) & vme_pulse_int;
assign fastor_linkrx3_reg_selected = (areg=='h70) & vme_pulse_int;
assign time_for_fastor_counter_selected = (areg=='h71) & vme_pulse_int;
// scope options
assign scope_0_selected = (areg=='h72) & vme_pulse_int;
assign scope_1_selected = (areg=='h73) & vme_pulse_int;
assign scope_2_selected = (areg=='h74) & vme_pulse_int;

// time_l0_to_l1
assign time_l0_to_l1_selected = (areg=='h75) & vme_pulse_int;

// other resets
assign reset_to_default_temp_limit = (areg=='h76) & vme_pulse_int;


// fastor from VME bus - for test !!!
assign fastor_from_vme_for_test = (areg=='h7f) & vme_pulse_int;
// status from linkrx ch0
assign read_status0_reg_linkRx_ch0 = (areg=='h80) & vme_pulse_int;
assign read_status1_reg_linkRx_ch0 = (areg=='h81) & vme_pulse_int;
assign read_status2_reg_linkRx_ch0 = (areg=='h82) & vme_pulse_int;
assign read_status3_reg_linkRx_ch0 = (areg=='h83) & vme_pulse_int;
assign read_status4_reg_linkRx_ch0 = (areg=='h84) & vme_pulse_int;
assign read_status5_reg_linkRx_ch0 = (areg=='h85) & vme_pulse_int;
assign read_status6_reg_linkRx_ch0 = (areg=='h86) & vme_pulse_int;
assign read_status7_reg_linkRx_ch0 = (areg=='h87) & vme_pulse_int;
assign read_status8_reg_linkRx_ch0 = (areg=='h88) & vme_pulse_int;
assign read_status9_reg_linkRx_ch0 = (areg=='h89) & vme_pulse_int;
assign read_status10_reg_linkRx_ch0 = (areg=='h8a) & vme_pulse_int;
assign read_status11_reg_linkRx_ch0 = (areg=='h8b) & vme_pulse_int;
assign read_status12_reg_linkRx_ch0 = (areg=='h8c) & vme_pulse_int;
assign read_status13_reg_linkRx_ch0 = (areg=='h8d) & vme_pulse_int;
assign read_status14_reg_linkRx_ch0 = (areg=='h8e) & vme_pulse_int;
assign read_status15_reg_linkRx_ch0 = (areg=='h8f) & vme_pulse_int;
// status from linkrx ch1
assign read_status0_reg_linkRx_ch1 = (areg=='h90) & vme_pulse_int;
assign read_status1_reg_linkRx_ch1 = (areg=='h91) & vme_pulse_int;
assign read_status2_reg_linkRx_ch1 = (areg=='h92) & vme_pulse_int;
assign read_status3_reg_linkRx_ch1 = (areg=='h93) & vme_pulse_int;
assign read_status4_reg_linkRx_ch1 = (areg=='h94) & vme_pulse_int;
assign read_status5_reg_linkRx_ch1 = (areg=='h95) & vme_pulse_int;
assign read_status6_reg_linkRx_ch1 = (areg=='h96) & vme_pulse_int;
assign read_status7_reg_linkRx_ch1 = (areg=='h97) & vme_pulse_int;
assign read_status8_reg_linkRx_ch1 = (areg=='h98) & vme_pulse_int;
assign read_status9_reg_linkRx_ch1 = (areg=='h99) & vme_pulse_int;
assign read_status10_reg_linkRx_ch1 = (areg=='h9a) & vme_pulse_int;
assign read_status11_reg_linkRx_ch1 = (areg=='h9b) & vme_pulse_int;
assign read_status12_reg_linkRx_ch1 = (areg=='h9c) & vme_pulse_int;
assign read_status13_reg_linkRx_ch1 = (areg=='h9d) & vme_pulse_int;
assign read_status14_reg_linkRx_ch1 = (areg=='h9e) & vme_pulse_int;
assign read_status15_reg_linkRx_ch1 = (areg=='h9f) & vme_pulse_int;
// status from linkrx ch2
assign read_status0_reg_linkRx_ch2 = (areg=='ha0) & vme_pulse_int;
assign read_status1_reg_linkRx_ch2 = (areg=='ha1) & vme_pulse_int;
assign read_status2_reg_linkRx_ch2 = (areg=='ha2) & vme_pulse_int;
assign read_status3_reg_linkRx_ch2 = (areg=='ha3) & vme_pulse_int;
assign read_status4_reg_linkRx_ch2 = (areg=='ha4) & vme_pulse_int;
assign read_status5_reg_linkRx_ch2 = (areg=='ha5) & vme_pulse_int;
assign read_status6_reg_linkRx_ch2 = (areg=='ha6) & vme_pulse_int;
assign read_status7_reg_linkRx_ch2 = (areg=='ha7) & vme_pulse_int;
assign read_status8_reg_linkRx_ch2 = (areg=='ha8) & vme_pulse_int;
assign read_status9_reg_linkRx_ch2 = (areg=='ha9) & vme_pulse_int;
assign read_status10_reg_linkRx_ch2 = (areg=='haa) & vme_pulse_int;
assign read_status11_reg_linkRx_ch2 = (areg=='hab) & vme_pulse_int;
assign read_status12_reg_linkRx_ch2 = (areg=='hac) & vme_pulse_int;
assign read_status13_reg_linkRx_ch2 = (areg=='had) & vme_pulse_int;
assign read_status14_reg_linkRx_ch2 = (areg=='hae) & vme_pulse_int;
assign read_status15_reg_linkRx_ch2 = (areg=='haf) & vme_pulse_int;
// status from linkrx ch3
assign read_status0_reg_linkRx_ch3 = (areg=='hb0) & vme_pulse_int;
assign read_status1_reg_linkRx_ch3 = (areg=='hb1) & vme_pulse_int;
assign read_status2_reg_linkRx_ch3 = (areg=='hb2) & vme_pulse_int;
assign read_status3_reg_linkRx_ch3 = (areg=='hb3) & vme_pulse_int;
assign read_status4_reg_linkRx_ch3 = (areg=='hb4) & vme_pulse_int;
assign read_status5_reg_linkRx_ch3 = (areg=='hb5) & vme_pulse_int;
assign read_status6_reg_linkRx_ch3 = (areg=='hb6) & vme_pulse_int;
assign read_status7_reg_linkRx_ch3 = (areg=='hb7) & vme_pulse_int;
assign read_status8_reg_linkRx_ch3 = (areg=='hb8) & vme_pulse_int;
assign read_status9_reg_linkRx_ch3 = (areg=='hb9) & vme_pulse_int;
assign read_status10_reg_linkRx_ch3 = (areg=='hba) & vme_pulse_int;
assign read_status11_reg_linkRx_ch3 = (areg=='hbb) & vme_pulse_int;
assign read_status12_reg_linkRx_ch3 = (areg=='hbc) & vme_pulse_int;
assign read_status13_reg_linkRx_ch3 = (areg=='hbd) & vme_pulse_int;
assign read_status14_reg_linkRx_ch3 = (areg=='hbe) & vme_pulse_int;
assign read_status15_reg_linkRx_ch3 = (areg=='hbf) & vme_pulse_int;
// status from linkrx ch4
assign read_status0_reg_linkRx_ch4 = (areg=='hc0) & vme_pulse_int;
assign read_status1_reg_linkRx_ch4 = (areg=='hc1) & vme_pulse_int;
assign read_status2_reg_linkRx_ch4 = (areg=='hc2) & vme_pulse_int;
assign read_status3_reg_linkRx_ch4 = (areg=='hc3) & vme_pulse_int;
assign read_status4_reg_linkRx_ch4 = (areg=='hc4) & vme_pulse_int;
assign read_status5_reg_linkRx_ch4 = (areg=='hc5) & vme_pulse_int;
assign read_status6_reg_linkRx_ch4 = (areg=='hc6) & vme_pulse_int;
assign read_status7_reg_linkRx_ch4 = (areg=='hc7) & vme_pulse_int;
assign read_status8_reg_linkRx_ch4 = (areg=='hc8) & vme_pulse_int;
assign read_status9_reg_linkRx_ch4 = (areg=='hc9) & vme_pulse_int;
assign read_status10_reg_linkRx_ch4 = (areg=='hca) & vme_pulse_int;
assign read_status11_reg_linkRx_ch4 = (areg=='hcb) & vme_pulse_int;
assign read_status12_reg_linkRx_ch4 = (areg=='hcc) & vme_pulse_int;
assign read_status13_reg_linkRx_ch4 = (areg=='hcd) & vme_pulse_int;
assign read_status14_reg_linkRx_ch4 = (areg=='hce) & vme_pulse_int;
assign read_status15_reg_linkRx_ch4 = (areg=='hcf) & vme_pulse_int;
// status from linkrx ch5
assign read_status0_reg_linkRx_ch5 = (areg=='hd0) & vme_pulse_int;
assign read_status1_reg_linkRx_ch5 = (areg=='hd1) & vme_pulse_int;
assign read_status2_reg_linkRx_ch5 = (areg=='hd2) & vme_pulse_int;
assign read_status3_reg_linkRx_ch5 = (areg=='hd3) & vme_pulse_int;
assign read_status4_reg_linkRx_ch5 = (areg=='hd4) & vme_pulse_int;
assign read_status5_reg_linkRx_ch5 = (areg=='hd5) & vme_pulse_int;
assign read_status6_reg_linkRx_ch5 = (areg=='hd6) & vme_pulse_int;
assign read_status7_reg_linkRx_ch5 = (areg=='hd7) & vme_pulse_int;
assign read_status8_reg_linkRx_ch5 = (areg=='hd8) & vme_pulse_int;
assign read_status9_reg_linkRx_ch5 = (areg=='hd9) & vme_pulse_int;
assign read_status10_reg_linkRx_ch5 = (areg=='hda) & vme_pulse_int;
assign read_status11_reg_linkRx_ch5 = (areg=='hdb) & vme_pulse_int;
assign read_status12_reg_linkRx_ch5 = (areg=='hdc) & vme_pulse_int;
assign read_status13_reg_linkRx_ch5 = (areg=='hdd) & vme_pulse_int;
assign read_status14_reg_linkRx_ch5 = (areg=='hde) & vme_pulse_int;
assign read_status15_reg_linkRx_ch5 = (areg=='hdf) & vme_pulse_int;
assign reset_digital_pilot              = (areg=='he0) & vme_pulse_int;
assign reset_pll                        = (areg=='he1) & vme_pulse_int;
 // reset PPL inside Router FPGA - Stratix
assign reset_pixel                      = (areg=='he2) & vme_pulse_int;assign
assign timeout_for_event_ready_selected = (areg=='he3) & vme_pulse_int;
assign number_of_l1_in_l1fifo_selected = (areg=='he4) & vme_pulse_int;

router_status1 [31:0] = {
                        no_activity,
                        l2_fifo_check,
                        init_timeout_counter,
                        slm0_check,
                        del_s0c,
                        slm1_check,
                        del_s1c,
                        slm2_check,
                        del_s2c,
                        slm3_check,
                        del_s3c,
                        slm4_check,
                        del_s4c,
                        slm5_check,
                        del_s5c,
                        timeout_check,
                        slm_check_done,
                        event_header_word_1,
                        event_header_word_2,
                        event_header_word_3,
                        event_header_word_4,
                        event_header_word_5,
                        event_header_word_6,
                        event_header_word_7,
                        event_header_word_8,
                        inc_slm_counter,
                        load_slm_status,
                        del_lss,
                        read_errors_and_eventnumber,
                        del_reaen,
                        read_start_address,
                        del_rsa
                           };

assign router_status2 [31:0] = {
                        read_lenght_of_block,
                        del_rlob,
                        read_reserved_word,
                        del_rrw,
                        hit_transfer_to_daq,
                        wait_for_hit_transfer_to_daq,
                        readout_skip,
                        readout_done,
                        check_DDL_fifo_empty_flag,
                        1'b0,
                        1'b0,
                        1'b0,
                        sync_with_daq,
                        sync_with_trigger_masters,
                        trigger_fifos_update,
                        l2_fifo_error,
                        input_error,
                        daq_error,
                        n_of_triggers_in_l1_fifo[3],
                        n_of_triggers_in_l1_fifo[2],
                        n_of_triggers_in_l1_fifo[1],
                        n_of_triggers_in_l1_fifo[0],
                        n_of_l2_triggers[3],
                        n_of_l2_triggers[2],
                        n_of_l2_triggers[1],
                        n_of_l2_triggers[0],
                        nearly_full_address_counter_fifo_sa,
                 dpm_full_flag,
                        linkrx_busy,
                        no_sequence_tpl1l2a,
                        l2_fifo_empty,
                        empty_address_counter_fifo // <-- event ready in DPM
                          };

assign router_status3 [31:0] = {
                        error_data_header_missing_flag,
                        error_wrong_chip_number_flag,
                        error_wrong_event_number_flag,
              error_data_missing_flag,
              error_data_trailer_missing_flag,
              error_fill_word_missing_flag,
                        linkrx_dpm_full_flag,
                        idle_violation_flag,
                        busy_violation_flag,
                        seu_error_flag,
                        check_other_processes,
                        router_header_check,
                        router_header_length,
                        router_header_word,
                        router_header_check_n_of_words,
                        1'b0,
                        1'b0,
                        no_daq_activity,   // daq_state_machine
                        daq_check,
                        wait_for_daq_ready,
                        event_header_transfer,
                        wait_for_daq_ready_in_header,
                        event_body_transfer,
                        wait_for_daq_ready_in_body,
                        no_l1_trigger,    // l1_state_machine
                        send_l1,
                        l1_error,
                        no_l2_trigger,   // l2_state_machine
                        send_l2a,
                        send_l2r,
                        l2_error,
                        all_sm_idle
                           };

assign router_control [31:0] = {
                        1'b0,
                        1'b0,
                        enable_PP_to_be_TestPulse,
                        enable_fastor_coincidence,
                        disabled_fastor_linkrx3,
                        disabled_fastor_linkrx2,
                        disabled_fastor_linkrx1,
                        not_active_fastor_linkrx3_for_counter,
                        not_active_fastor_linkrx2_for_counter,
                        not_active_fastor_linkrx1_for_counter,
                        exclude_triggers_from_ttc,
                        enable_orbn_internal,
                        router_header_active_flag,
                        faster_jtag,
                        jtagch5_common,
                        jtagch4_common,
                        jtagch3_common,
                        jtagch2_common,
                        jtagch1_common,
                        jtagch0_common,
                        short_tditdo,
                        stop_all_sm,
                        enable_tp,
                        busy_set_by_vme,
                        disabled_channel[5],   // disabled_channels
                        disabled_channel[4],
                        disabled_channel[3],
                        disabled_channel[2],
                        disabled_channel[1],
                        disabled_channel[0],
                        dont_send_data_to_daq,
                        sample_enable_flag
                          };

              assign f_panel_scope[0] =(scope_0[0] & reset) |
                    (scope_0[1] & clk)  | // it is clk40_des1
                    (scope_0[2] & dclk) | // it is 10 Mhz clock
                    (scope_0[3] & bcntres)    |
                    (scope_0[4] & l0_from_ttc) |
                    (scope_0[5] & l1_from_ttc) |
                    (scope_0[6] & L1_header)   |
                    (scope_0[7] & l2a_from_ttc) |
                    (scope_0[8] & l2r_from_ttc) |
                    (scope_0[9] & router_busy)  |
                    (scope_0[10] & linkrx1_busy) |
                    (scope_0[11] & linkrx2_busy) |
                    (scope_0[12] & linkrx3_busy) |
                    (scope_0[13] & reset_for_FEE)  |
                    (scope_0[14] & L1_sent_to_det) |
                    (scope_0[15] & fastor_normal)  |
                    (scope_0[16] & fastor_coincidence) |
                    (scope_0[17] & dpm_full_flag)  |
                    (scope_0[18] & indications[0]) |
                    (scope_0[19] & indications[1]) |
                    (scope_0[20] & indications[2]) |
                    (scope_0[21] & indications[3]) |
                    (scope_0[22] & indications[4]) |
                    (scope_0[23] & indications[5]) |
                    (scope_0[24] & PP)             |
                    (scope_0[25] & reset_DP_sent_to_linkRx);

assign f_panel_scope[1] =(scope_1[0] & reset) |
                    (scope_1[1] & clk)  | // it is clk40_des1
                    (scope_1[2] & dclk) | // it is 10 Mhz clock
                    (scope_1[3] & bcntres)    |
                    (scope_1[4] & l0_from_ttc) |
                    (scope_1[5] & l1_from_ttc) |
                    (scope_1[6] & L1_header)   |
                    (scope_1[7] & l2a_from_ttc) |
                    (scope_1[8] & l2r_from_ttc) |
                    (scope_1[9] & router_busy)  |
                    (scope_1[10] & linkrx1_busy) |
                    (scope_1[11] & linkrx2_busy) |
                    (scope_1[12] & linkrx3_busy) |
                    (scope_1[13] & reset_for_FEE)  |
                    (scope_1[14] & L1_sent_to_det) |
                    (scope_1[15] & fastor_normal)  |
                    (scope_1[16] & fastor_coincidence) |
                    (scope_1[17] & dpm_full_flag)  |
                    (scope_1[18] & indications[0]) |
                    (scope_1[19] & indications[1]) |
                    (scope_1[20] & indications[2]) |
                    (scope_1[21] & indications[3]) |
                    (scope_1[22] & indications[4]) |
                    (scope_1[23] & indications[5]) |
                    (scope_1[24] & PP)             |
                    (scope_1[25] & reset_DP_sent_to_linkRx);

assign f_panel_scope[2] =(scope_2[0] & reset) |
                    (scope_2[1] & clk)  | // it is clk40_des1
                    (scope_2[2] & dclk) | // it is 10 Mhz clock
```

```
(scope_2[3] & bcntres)    |
(scope_2[4] & l0_from_ttc) |
(scope_2[5] & l1_from_ttc) |
(scope_2[6] & L1_header)   |
(scope_2[7] & l2a_from_ttc) |
(scope_2[8] & l2r_from_ttc) |
(scope_2[9] & router_busy)  |
(scope_2[10] & linkrx1_busy) |
(scope_2[11] & linkrx2_busy) |
(scope_2[12] & linkrx3_busy) |
(scope_2[13] & reset_for_FEE)  |
(scope_2[14] & L1_sent_to_det) |
(scope_2[15] & fastor_normal)  |
(scope_2[16] & fastor_coincidence) |
(scope_2[17] & dpm_full_flag)  |
(scope_2[18] & indications[0]) |
(scope_2[19] & indications[1]) |
(scope_2[20] & indications[2]) |
(scope_2[21] & indications[3]) |
(scope_2[22] & indications[4]) |
(scope_2[23] & indications[5]) |
(scope_2[24] & PP)             |
(scope_2[25] & reset_DP_sent_to_linkRx);
```