Declarations

Ports:

```
BCRES_FROM_TIM_M : std_logic
BX               : std_logic_vector(11 DOWNTO 0)
CLK80_TCSM       : std_logic
CLK_FBIN_TCSM    : std_logic
CLK_TCSM         : std_logic
EN_TCSM          : std_logic
EN_TCS_TCSM      : std_logic
EVM_STATUS       : std_logic_vector(3 DOWNTO 0)
F0300            : std_logic_vector(20 DOWNTO 0)
F0704            : std_logic_vector(20 DOWNTO 0)
F1108            : std_logic_vector(20 DOWNTO 0)
F1512            : std_logic_vector(20 DOWNTO 0)
F1916            : std_logic_vector(20 DOWNTO 0)
F2320            : std_logic_vector(20 DOWNTO 0)
F_DAQ0           : std_logic_vector(20 DOWNTO 0)
F_DAQ1           : std_logic_vector(20 DOWNTO 0)
GT_STATUS        : std_logic_vector(3 DOWNTO 0)
ID               : std_logic_vector(2 DOWNTO 0)
INACTIVE         : std_logic
L1A_FROM_TIM_M   : std_logic
L1RES_FROM_TIM_M : std_logic
MON_F3124        : std_logic_vector(31 DOWNTO 0)
RDRQST           : std_logic
RESET_TCSM_CHIP  : std_logic
STOP_MON         : std_logic
STROB            : std_logic_vector(2 DOWNTO 0)
VA_I             : std_logic_vector(19 DOWNTO 1)
WR_TCSM          : std_logic
CLK_FBOUT_TCSM   : std_logic
CLK_LOCKED_TCSM  : std_logic
DAQ_CLK          : std_logic
DAQ_D            : std_logic_vector(27 DOWNTO 0)
FASTIN           : std_logic_vector(81 DOWNTO 0)
FASTIN_CYC       : std_logic
IRQ_X            : std_logic
NDTACK_TCSM      : std_logic
NEN_MONLINK      : std_logic
NEN_ROBUS        : std_logic
TEST1_TCSM       : std_logic
TEST2_TCSM       : std_logic
TEST_0S          : std_logic
TEST_2S          : std_logic
TEST_5P          : std_logic
TEST_7P          : std_logic
VD_I             : std_logic_vector(15 DOWNTO 0)
```

Diagram Signals:

```
SIGNAL CLK0             : std_ulogic := '0'
SIGNAL CLK2X            : std_ulogic := '0'
SIGNAL CLKIN            : std_logic
SIGNAL accum_mode       : std_logic
SIGNAL bcres_tim        : std_logic
SIGNAL board_nr         : std_logic_vector(2 DOWNTO 0)
SIGNAL clk40            : std_logic
SIGNAL clk80            : std_logic
SIGNAL clkfb_unused     : STD_ULOGIC
SIGNAL clr              : std_logic
SIGNAL dcm_locked       : std_logic
SIGNAL dis_vdoo         : std_logic
SIGNAL dout             : std_logic
SIGNAL dtack            : std_logic
SIGNAL dtack_tp         : std_logic
SIGNAL en_vdoo          : std_logic
SIGNAL gnd_28           : std_logic_vector(27 DOWNTO 0)
SIGNAL ground           : std_logic
SIGNAL gsr              : STD_ULOGIC
SIGNAL gt_evm_stat      : std_logic_vector(15 DOWNTO 0)
SIGNAL gts              : std_ulogic
SIGNAL inc_red_cntrs    : std_logic_vector(41 DOWNTO 0)
```

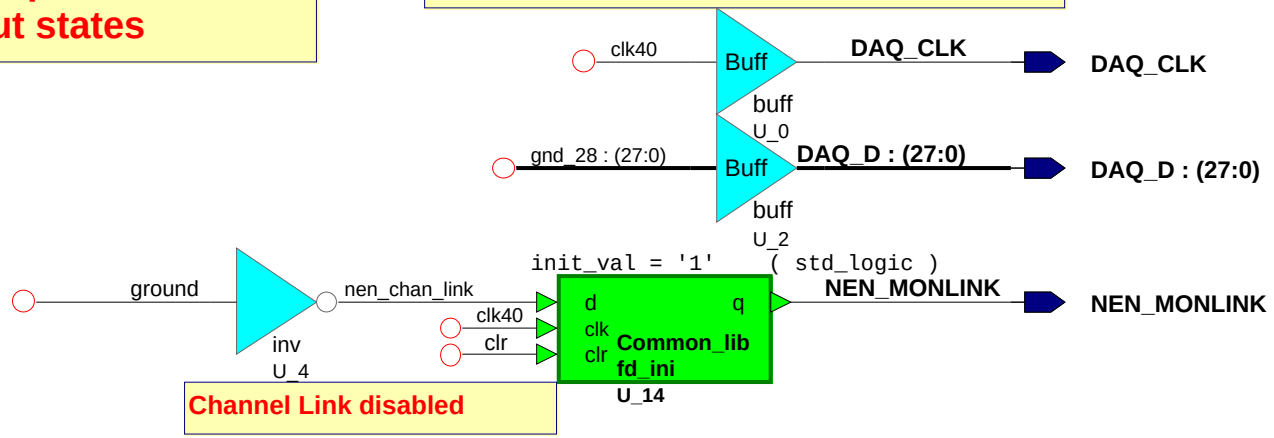
Package List

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
LIBRARY tcs_types;
USE tcs_types.tcs.all;
USE ieee.STD_LOGIC_UNSIGNED.all;
USE IEEE.VITAL_Timing.all;
LIBRARY STD;
USE STD.TEXTIO.all;
LIBRARY unisim;
USE unisim.VPKG.all;
```

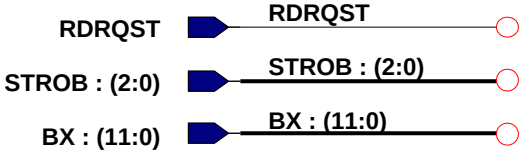
<company name>		Project:	tcs_chip
Title:	<enter diagram title here>	<enter comments here>	
Path:	tcs_chip_lib/tcsm_chip/struct		
Edited:	by taurok on 25 Mär 2008		

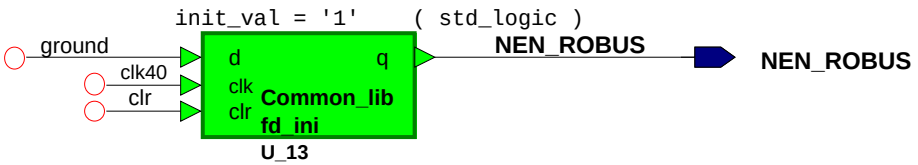
ROP to be implemented
- actual input states

ChannelLink to GTFE disabled



ROBUS enabled, but signals are not used

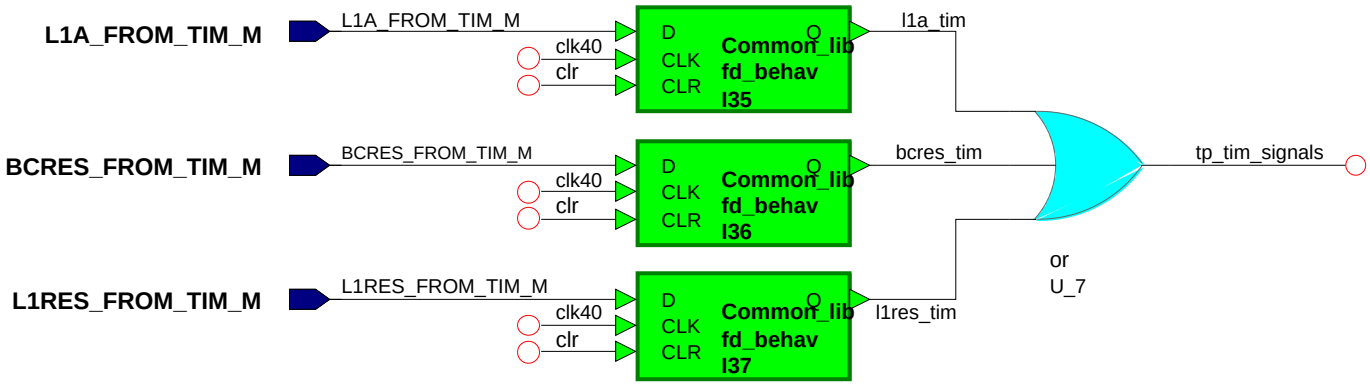




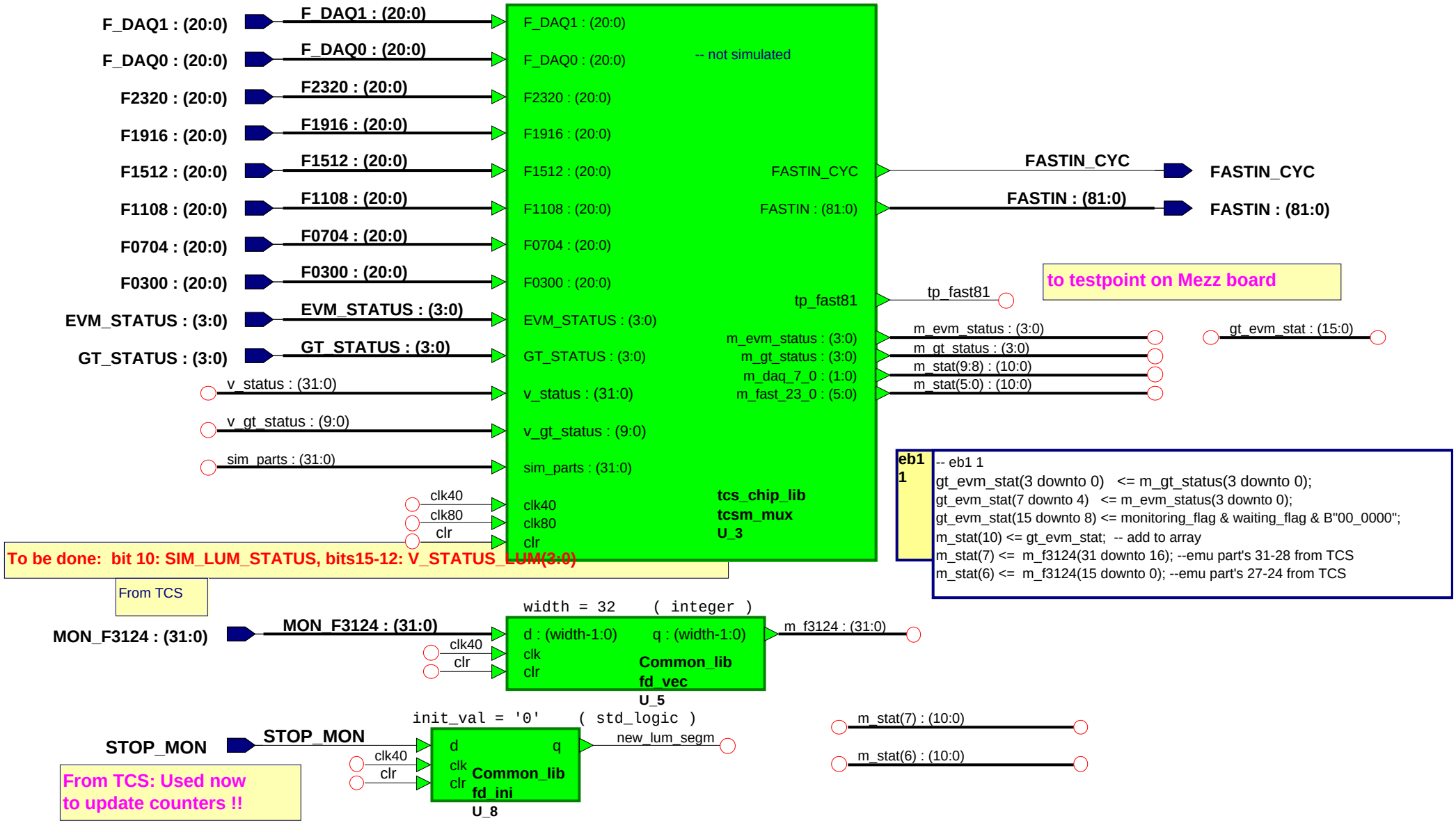
ROBUS enabled, but signals are not used

delayed bcr to testpoint

Signals from TIM not used

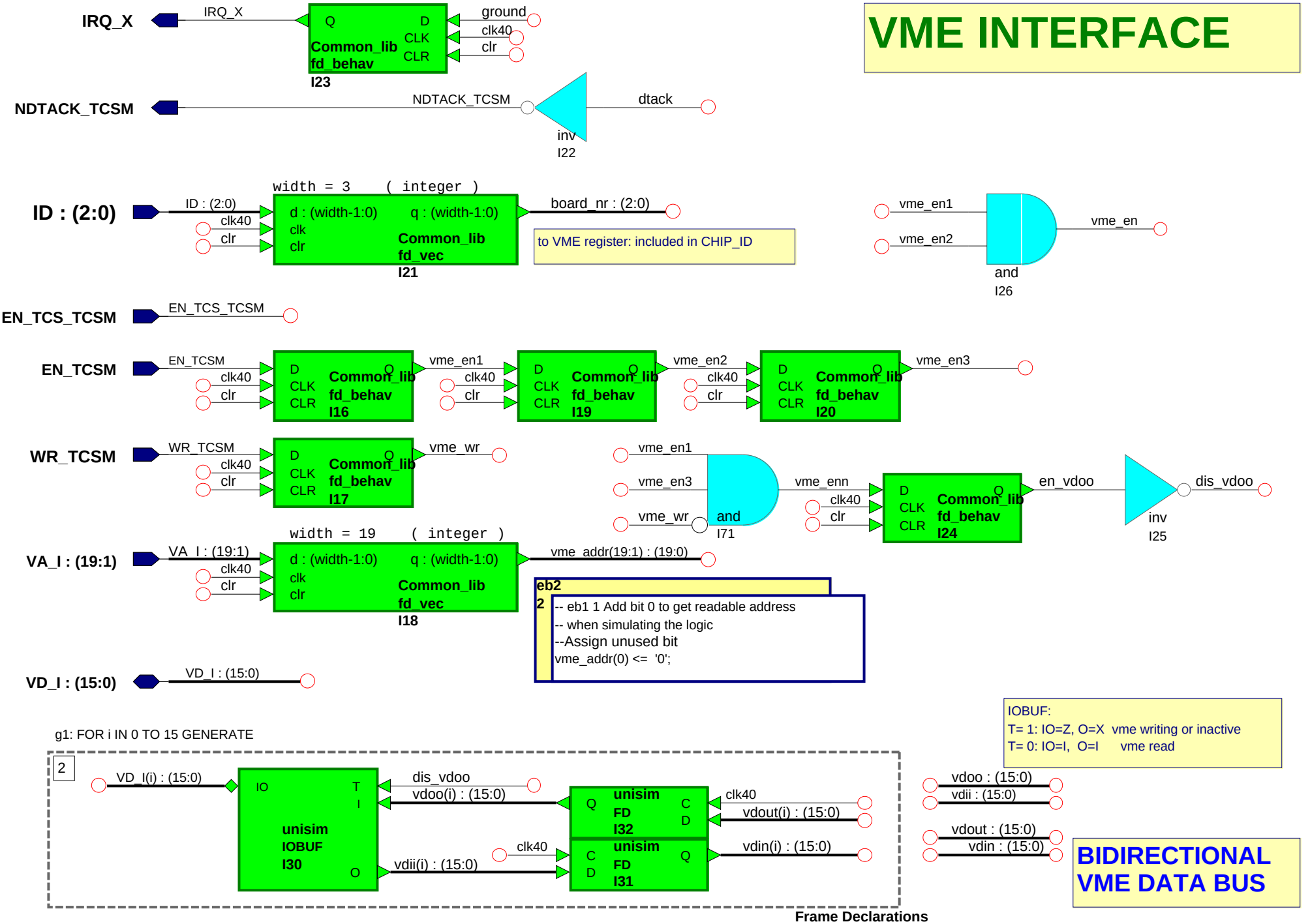


STATUS MULTPLEXER



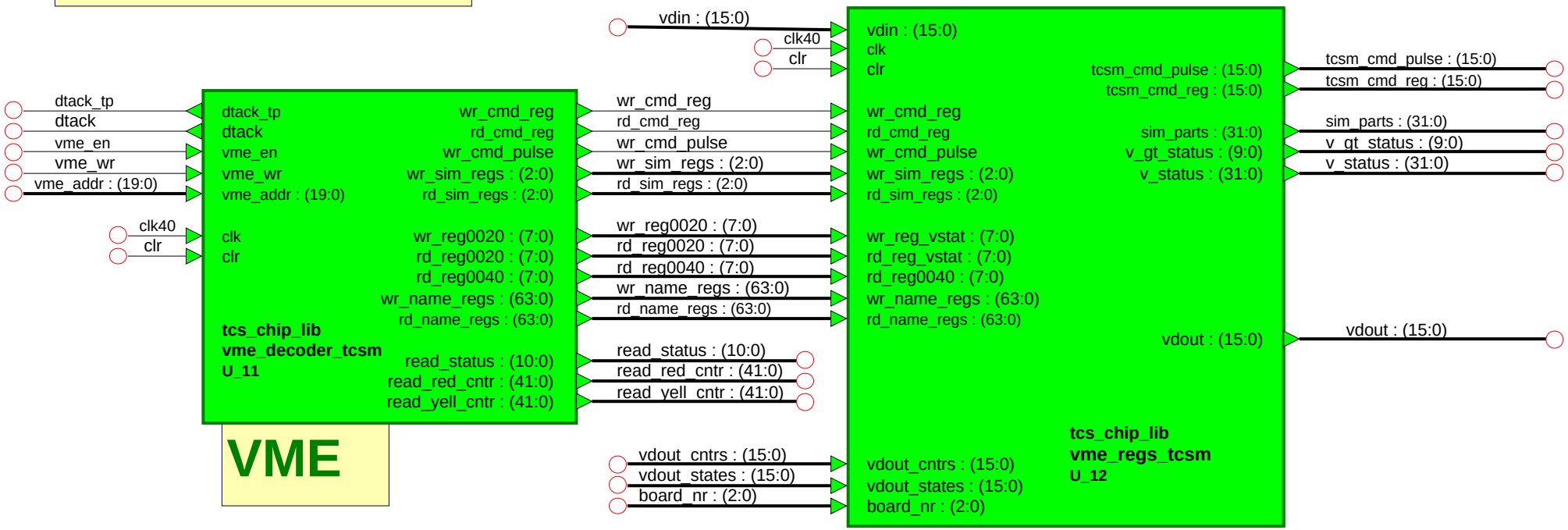
```
SIGNAL inc_yellow_cntrs : std_logic_vector(41 DOWNTO 0)
SIGNAL l1a_tim          : std_logic
SIGNAL l1res_tim        : std_logic
SIGNAL m_evm_status     : std_logic_vector(3 DOWNTO 0)
SIGNAL m_f3124          : std_logic_vector(31 DOWNTO 0)
SIGNAL m_gt_status      : std_logic_vector(3 DOWNTO 0)
SIGNAL m_stat           : vec16_array(10 DOWNTO 0)
SIGNAL monitoring_flag  : std_logic
SIGNAL nen_chan_link    : std_logic
SIGNAL new_lum_segm     : std_logic
SIGNAL rd_cmd_reg       : std_logic
SIGNAL rd_name_regs     : std_logic_vector(63 DOWNTO 0)
SIGNAL rd_reg0020       : std_logic_vector(7 DOWNTO 0)
SIGNAL rd_reg0040       : std_logic_vector(7 DOWNTO 0)
SIGNAL rd_sim_regs      : std_logic_vector(2 DOWNTO 0)
SIGNAL read_red_cntr    : std_logic_vector(41 DOWNTO 0)
SIGNAL read_status      : std_logic_vector(10 DOWNTO 0)
SIGNAL read_yell_cntr   : std_logic_vector(41 DOWNTO 0)
SIGNAL reset_all        : std_logic
SIGNAL reset_counters   : std_logic
SIGNAL save_mode        : std_logic_vector(1 DOWNTO 0)
SIGNAL sim_parts        : std_logic_vector(31 DOWNTO 0)
SIGNAL start_mon_vme    : std_logic
SIGNAL start_wait_vme   : std_logic
SIGNAL stop_mon_vme     : std_logic
SIGNAL stop_wait_vme    : std_logic
SIGNAL tcsm_cmd_pulse   : std_logic_vector(15 DOWNTO 0)
SIGNAL tcsm_cmd_reg     : std_logic_vector(15 DOWNTO 0)
SIGNAL tp_fast81        : std_logic
SIGNAL tp_tim_signals   : std_logic
SIGNAL v_gt_status      : std_logic_vector(9 DOWNTO 0)
SIGNAL v_status         : vec4_array(31 DOWNTO 0)
SIGNAL vdii             : std_logic_vector(15 DOWNTO 0)
SIGNAL vdin             : std_logic_vector(15 DOWNTO 0)
SIGNAL vdoo             : std_logic_vector(15 DOWNTO 0)
SIGNAL vdout            : std_logic_vector(15 DOWNTO 0)
SIGNAL vdout_cntrs      : std_logic_vector(15 DOWNTO 0)
SIGNAL vdout_states     : std_logic_vector(15 DOWNTO 0)
SIGNAL vme_addr         : std_logic_vector(19 DOWNTO 0)
SIGNAL vme_en           : std_logic
SIGNAL vme_en1          : std_logic
SIGNAL vme_en2          : std_logic
SIGNAL vme_en3          : std_logic
SIGNAL vme_enn          : std_logic
SIGNAL vme_wr           : std_logic
SIGNAL waiting_flag     : std_logic
SIGNAL wr_cmd_pulse     : std_logic
SIGNAL wr_cmd_reg       : std_logic
SIGNAL wr_name_regs     : std_logic_vector(63 DOWNTO 0)
SIGNAL wr_reg0020       : std_logic_vector(7 DOWNTO 0)
SIGNAL wr_sim_regs      : std_logic_vector(2 DOWNTO 0)
```

VME INTERFACE

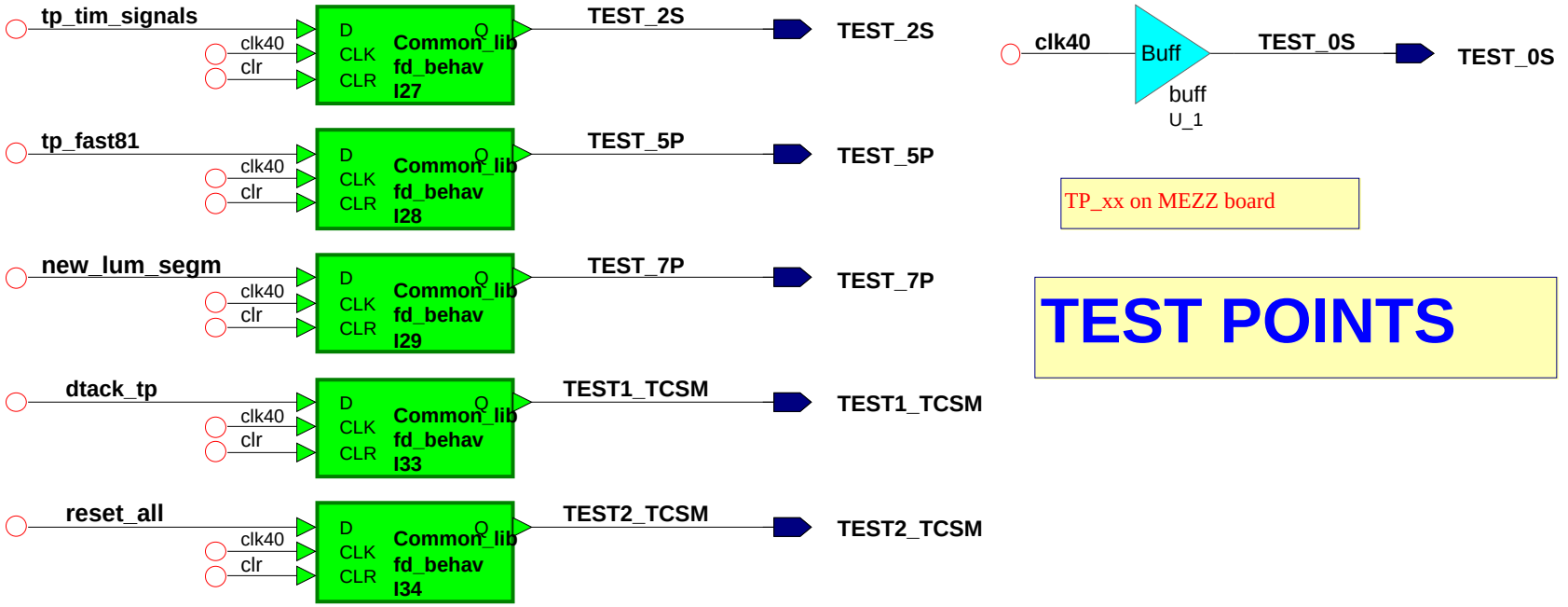


VERSION NR

```
version_nr_h = X"0000"    ( std_logic_vector(15 downto 0) )
version_nr_l = X"0003"    ( std_logic_vector(15 downto 0) )
```

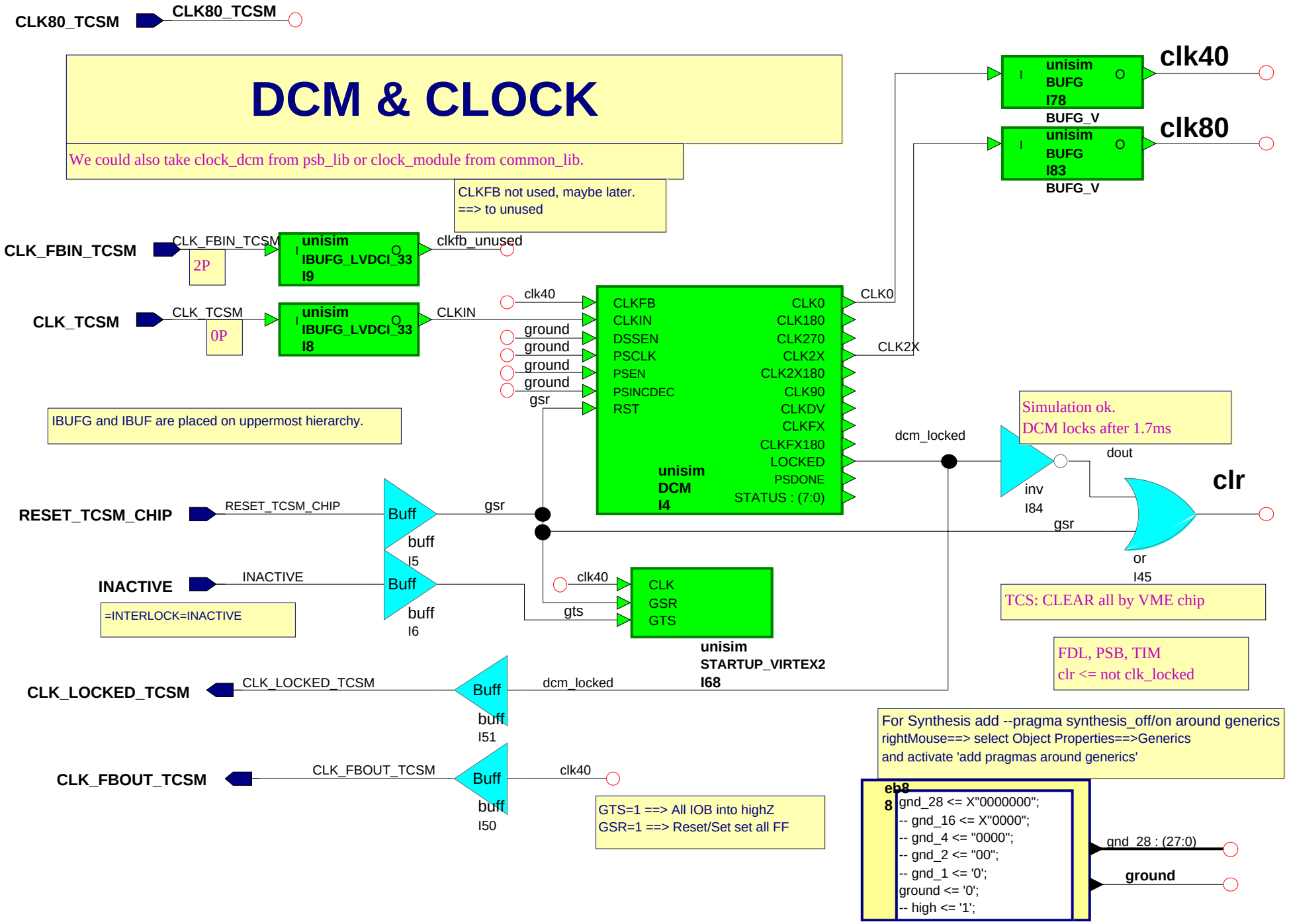


VME



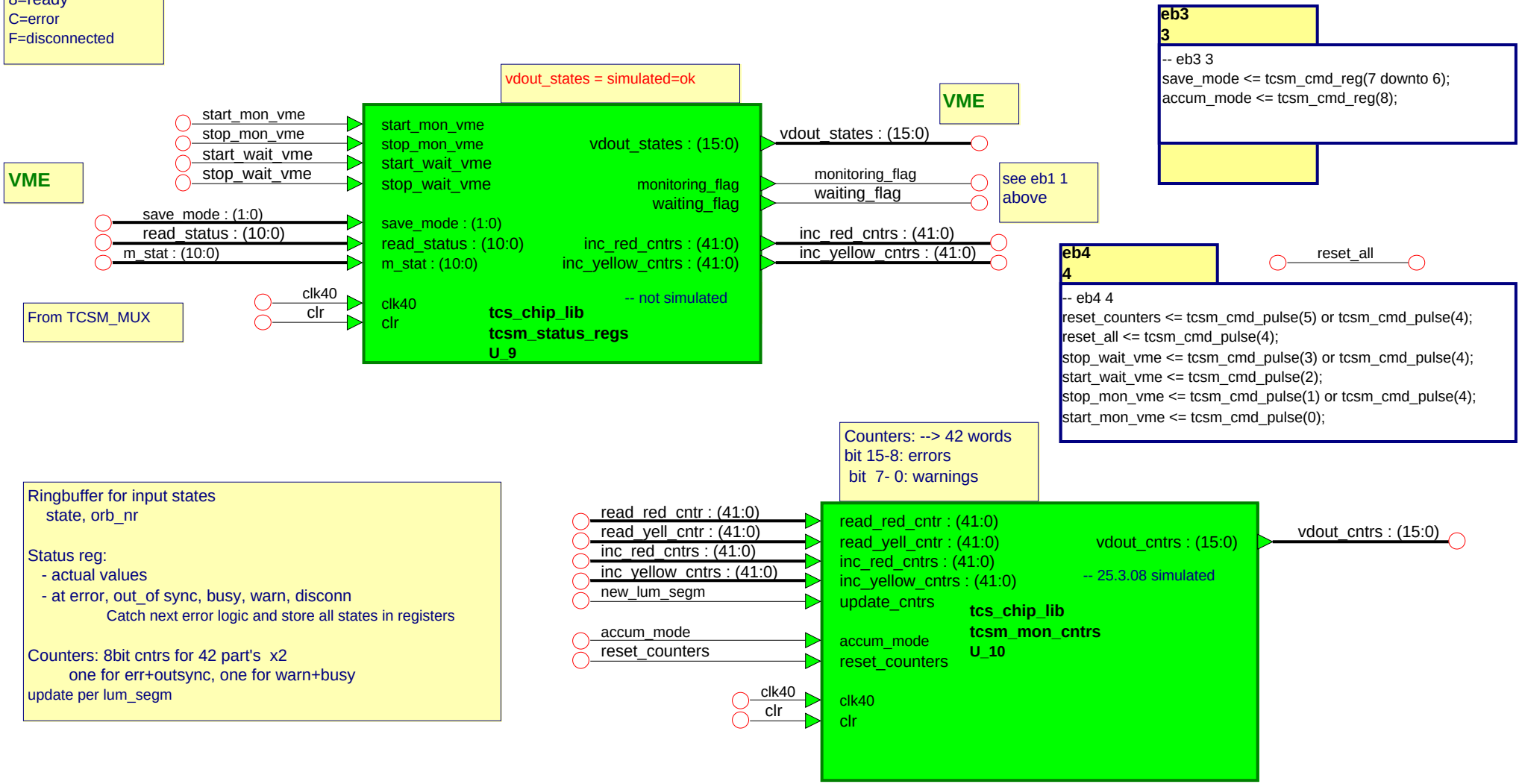
TEST POINTS

```
TimingChecksOn                        = true                    ( boolean          )
InstancePath                          = "*"                     ( string           )
Xon                                   = true                    ( boolean          )
MsgOn                                 = false                   ( boolean          )
thold_PSEN_PSCLK_negedge_posedge      = 0.000 ns                ( VitalDelayType   )
thold_PSEN_PSCLK_posedge_posedge      = 0.000 ns                ( VitalDelayType   )
thold_PSINCDEC_PSCLK_negedge_posedge  = 0.000 ns                ( VitalDelayType   )
thold_PSINCDEC_PSCLK_posedge_posedge  = 0.000 ns                ( VitalDelayType   )
tipd_CLKFB                            = (0.000 ns, 0.000 ns)    ( VitalDelayType01 )
tipd_CLKIN                            = (0.000 ns, 0.000 ns)    ( VitalDelayType01 )
tipd_DSSEN                            = (0.000 ns, 0.000 ns)    ( VitalDelayType01 )
tipd_PSCLK                            = (0.000 ns, 0.000 ns)    ( VitalDelayType01 )
tipd_PSEN                             = (0.000 ns, 0.000 ns)    ( VitalDelayType01 )
tipd_PSINCDEC                         = (0.000 ns, 0.000 ns)    ( VitalDelayType01 )
tipd_RST                              = (0.000 ns, 0.000 ns)    ( VitalDelayType01 )
tpd_CLKIN_LOCKED                      = (0.000 ns, 0.000 ns)    ( VitalDelayType01 )
tpd_PSCLK_PSDONE                      = (0.000 ns, 0.000 ns)    ( VitalDelayType01 )
tperiod_CLKIN_POSEDGE                 = 0.000 ns                ( VitalDelayType   )
tperiod_PSCLK_POSEDGE                 = 0.000 ns                ( VitalDelayType   )
tpw_CLKIN_negedge                     = 0.000 ns                ( VitalDelayType   )
tpw_CLKIN_posedge                     = 0.000 ns                ( VitalDelayType   )
tpw_PSCLK_negedge                     = 0.000 ns                ( VitalDelayType   )
tpw_PSCLK_posedge                     = 0.000 ns                ( VitalDelayType   )
tpw_RST_posedge                       = 0.000 ns                ( VitalDelayType   )
tsetup_PSEN_PSCLK_negedge_posedge     = 0.000 ns                ( VitalDelayType   )
tsetup_PSEN_PSCLK_posedge_posedge     = 0.000 ns                ( VitalDelayType   )
tsetup_PSINCDEC_PSCLK_negedge_posedge = 0.000 ns                ( VitalDelayType   )
tsetup_PSINCDEC_PSCLK_posedge_posedge = 0.000 ns                ( VitalDelayType   )
CLKDV_DIVIDE                          = 2.0                     ( real             )
CLKFX_DIVIDE                          = 1                       ( integer          )
CLKFX_MULTIPLY                        = 4                       ( integer          )
CLKIN_DIVIDE_BY_2                     = false                   ( boolean          )
CLKIN_PERIOD                          = 23.0                    ( real             )
CLKOUT_PHASE_SHIFT                    = "NONE"                  ( string           )
CLK_FEEDBACK                          = "1X"                    ( string           )
DESKEW_ADJUST                         = "SYSTEM_SYNCHRONOUS"    ( string           )
DFS_FREQUENCY_MODE                    = "LOW"                   ( string           )
DLL_FREQUENCY_MODE                    = "LOW"                   ( string           )
DSS_MODE                              = "NONE"                  ( string           )
DUTY_CYCLE_CORRECTION                 = true                    ( boolean          )
FACTORY_JF                            = X"C080"                 ( bit_vector       )
MAXPERCLKIN                           = 1000000 ps              ( time             )
MAXPERPSCLK                           = 100000000 ps            ( time             )
PHASE_SHIFT                           = 0                       ( integer          )
SIM_CLKIN_CYCLE_JITTER                = 300 ps                  ( time             )
SIM_CLKIN_PERIOD_JITTER               = 1000 ps                 ( time             )
STARTUP_WAIT                          = false                   ( boolean          )
```


CLK80_TCSM
CLK80_TCSM
DCM & CLOCK
We could also take clock_dcm from psb_lib or clock_module from common_lib.
CLKFB not used, maybe later.
==> to unused
unisim
BUFG
I78
BUFG_V
clk40
unisim
BUFG
I83
BUFG_V
clk80
CLK_FBIN_TCSM
CLK_FBIN_TCSM
2P
unisim
IBUFG_LVDCI_33
I9
clkfb_unused
CLK_TCSM
CLK_TCSM
0P
unisim
IBUFG_LVDCI_33
I8
CLKIN
clk40
ground
ground
ground
ground
gsr
CLKFB
CLKIN
DSSEN
PSCLK
PSEN
PSINCDEC
RST
CLK0
CLK180
CLK270
CLK2X
CLK2X180
CLK90
CLKDV
CLKFX
CLKFX180
LOCKED
PSDONE
STATUS : (7:0)
unisim
DCM
I4
CLK0
CLK2X
IBUFG and IBUF are placed on uppermost hierarchy.
dcm_locked
Simulation ok.
DCM locks after 1.7ms
dout
inv
I84
clr
gsr
or
I45
RESET_TCSM_CHIP
RESET_TCSM_CHIP
Buff
buff
I5
gsr
INACTIVE
INACTIVE
Buff
buff
I6
=INTERLOCK=INACTIVE
clk40
gts
CLK
GSR
GTS
unisim
STARTUP_VIRTEX2
I68
TCS: CLEAR all by VME chip
FDL, PSB, TIM
clr <= not clk_locked
CLK_LOCKED_TCSM
CLK_LOCKED_TCSM
Buff
buff
I51
dcm_locked
For Synthesis add --pragma synthesis_off/on around generics
rightMouse==> select Object Properties==>Generics
and activate 'add pragmas around generics'
CLK_FBOUT_TCSM
CLK_FBOUT_TCSM
Buff
buff
I50
clk40
GTS=1 ==> All IOB into highZ
GSR=1 ==> Reset/Set set all FF
eb8
8
gnd_28 <= X"0000000";
-- gnd_16 <= X"0000";
-- gnd_4 <= "0000";
-- gnd_2 <= "00";
-- gnd_1 <= '0';
ground <= '0';
-- high <= '1';
gnd_28 : (27:0)
ground

STATUS MONITORING

STATUS CODES:
0=disconnected
1=warning
2=out_of_sync
4=busy
8=ready
C=error
F=disconnected



Ringbuffer for input states
state, orb_nr

Status reg:
- actual values
- at error, out_of sync, busy, warn, disconn
Catch next error logic and store all states in registers

Counters: 8bit cntrs for 42 part's x2
one for err+outsync, one for warn+busy
update per lum_segm