

## Declarations

### Ports:

```
BCRES_FROM_TIM_M : std_logic
BX                : std_logic_vector(11 DOWNT0 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 DOWNT0 0)
F0300            : std_logic_vector(20 DOWNT0 0)
F0704            : std_logic_vector(20 DOWNT0 0)
F1108            : std_logic_vector(20 DOWNT0 0)
F1512            : std_logic_vector(20 DOWNT0 0)
F1916            : std_logic_vector(20 DOWNT0 0)
F2320            : std_logic_vector(20 DOWNT0 0)
F_DAQ0           : std_logic_vector(20 DOWNT0 0)
F_DAQ1           : std_logic_vector(20 DOWNT0 0)
GT_STATUS        : std_logic_vector(3 DOWNT0 0)
ID               : std_logic_vector(2 DOWNT0 0)
INACTIVE         : std_logic
L1A_FROM_TIM_M  : std_logic
L1RES_FROM_TIM_M : std_logic
MON_F3124       : std_logic_vector(31 DOWNT0 0)
RDRQST          : std_logic
RESET_TCSM_CHIP : std_logic
STOP_MON        : std_logic
STROB           : std_logic_vector(2 DOWNT0 0)
VA_I            : std_logic_vector(19 DOWNT0 1)
WR_TCSM         : std_logic
CLK_FBOU_TCSM   : std_logic
CLK_LOCKED_TCSM : std_logic
DAQ_CLK         : std_logic
DAQ_D           : std_logic_vector(27 DOWNT0 0)
FASTIN          : std_logic_vector(81 DOWNT0 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 DOWNT0 0)
```

### Diagram Signals:

```
SIGNAL CLK0           : std_ulogic := '0'
SIGNAL CLK2X          : std_ulogic := '0'
SIGNAL CLKIN          : std_logic
SIGNAL accum_mode     : std_logic
SIGNAL bres_tim       : std_logic
SIGNAL board_nr       : std_logic_vector(2 DOWNT0 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 DOWNT0 0)
SIGNAL ground         : std_logic
SIGNAL gsr            : STD_ULOGIC
SIGNAL gt_evm_stat    : std_logic_vector(15 DOWNT0 0)
SIGNAL gts            : std_ulogic
SIGNAL inc_red_cntrs  : std logic vector(41 DOWNT0 0)
```

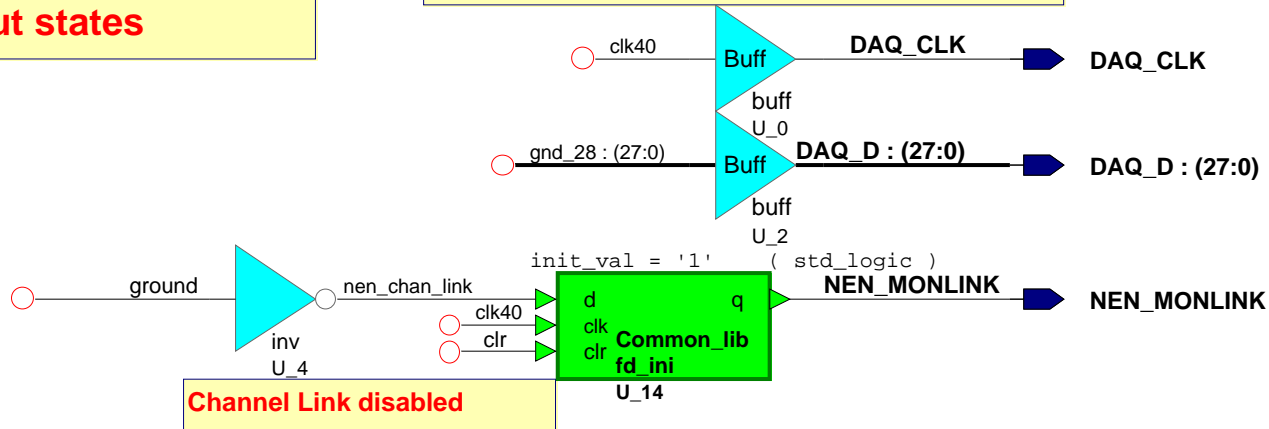
## Package List

```
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;
```

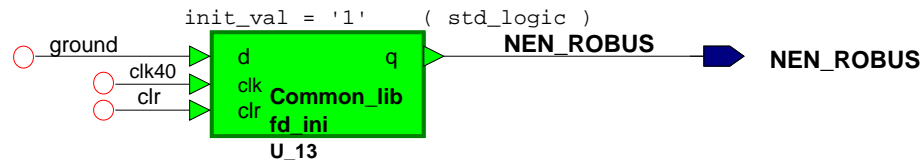
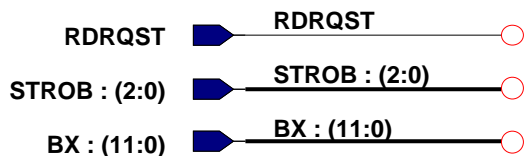
|                |  |                       |          |
|----------------|--|-----------------------|----------|
| <company name> |  | Project:              | tcs_chip |
| Title:         |  | <enter comments here> |          |
| Path:          |  |                       |          |
| Edited:        |  |                       |          |

**ROP to be implemented  
- actual input states**

**ChannelLink to GTFE disabled**



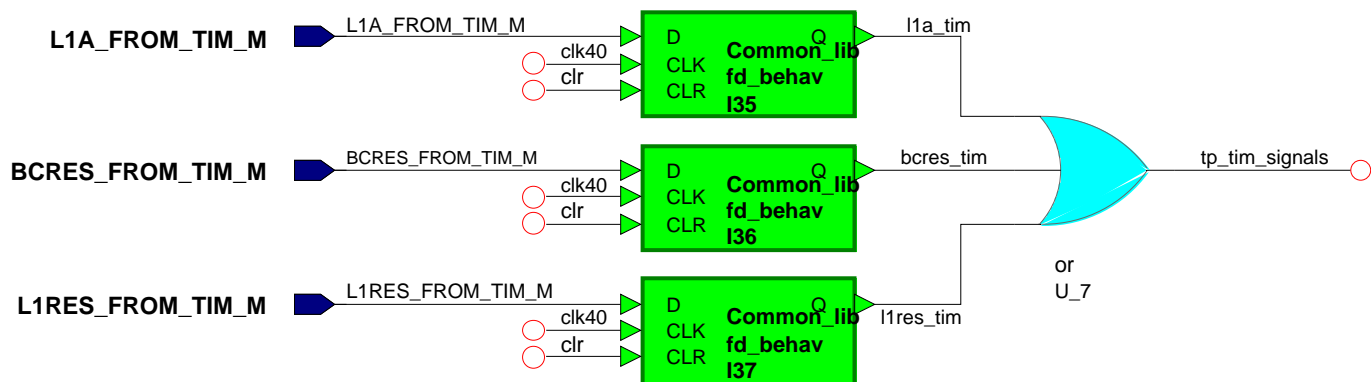
**ROBUS enabled, but signals are not used**



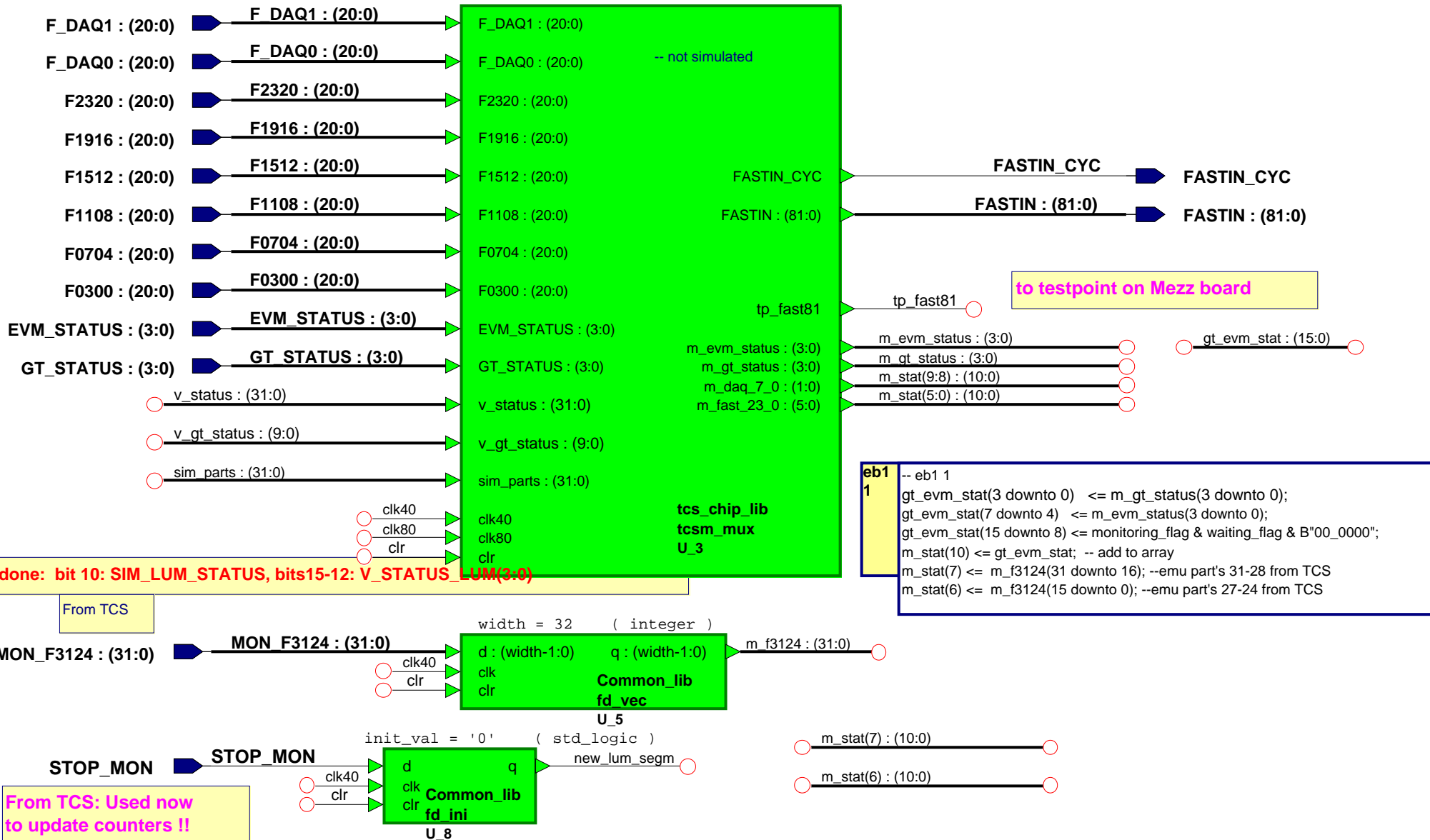
**ROBUS enabled, but signals are not used**

**delayed bcr to testpoint**

**Signals from TIM not used**

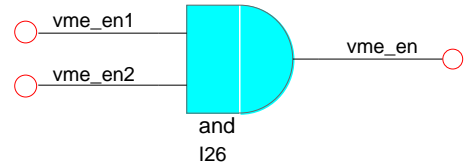
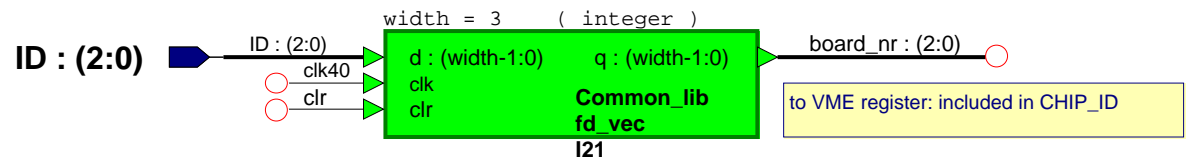
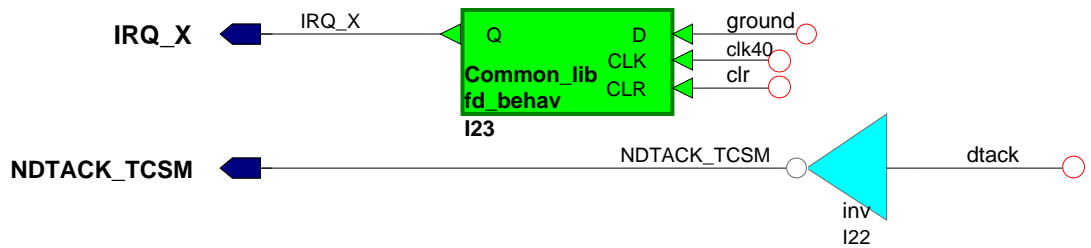


# STATUS MULTIPLEXER

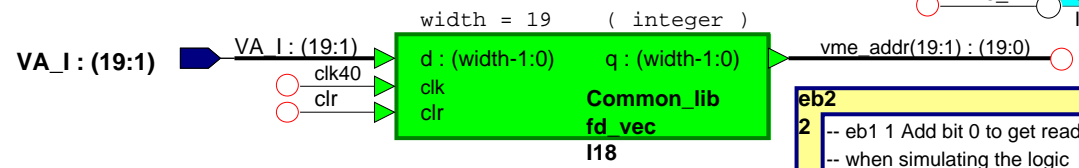
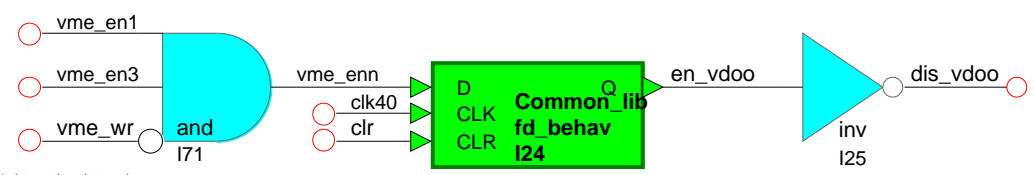
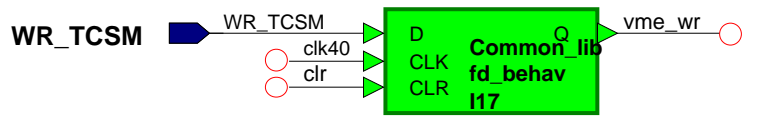
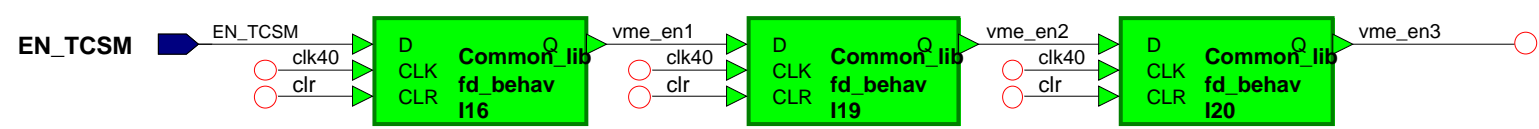


SIGNAL inc\_yellow\_cntrs : std\_logic\_vector(41 DOWNTO 0)  
SIGNAL lla\_tim : std\_logic  
SIGNAL llres\_tim : std\_logic  
SIGNAL m\_evm\_status : std\_logic\_vector(3 DOWNTO 0)  
SIGNAL m\_f3124 : std\_logic\_vector(31 DOWNTO 0)  
SIGNAL m\_gt\_status : std\_logic\_vector(3 DOWNTO 0)  
SIGNAL m\_stat : 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 tcsmd\_cmd\_pulse : std\_logic\_vector(15 DOWNTO 0)  
SIGNAL tcsmd\_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 vdo0 : 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



EN\_TCS\_TCSM



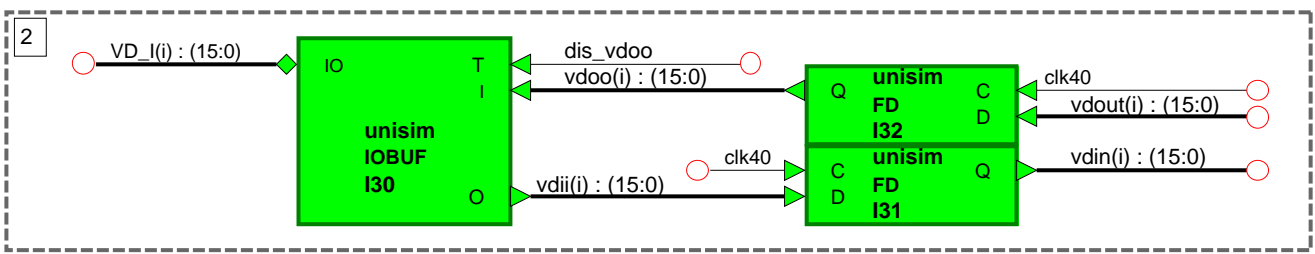
```

eb2
2
-- eb1 1 Add bit 0 to get readable address
-- when simulating the logic
--Assign unused bit
vme_addr(0) <= '0';
    
```

VD\_I : (15:0)

IOBUF:  
 T= 1: IO=Z, O=X vme writing or inactive  
 T= 0: IO=I, O=I vme read

g1: FOR i IN 0 TO 15 GENERATE

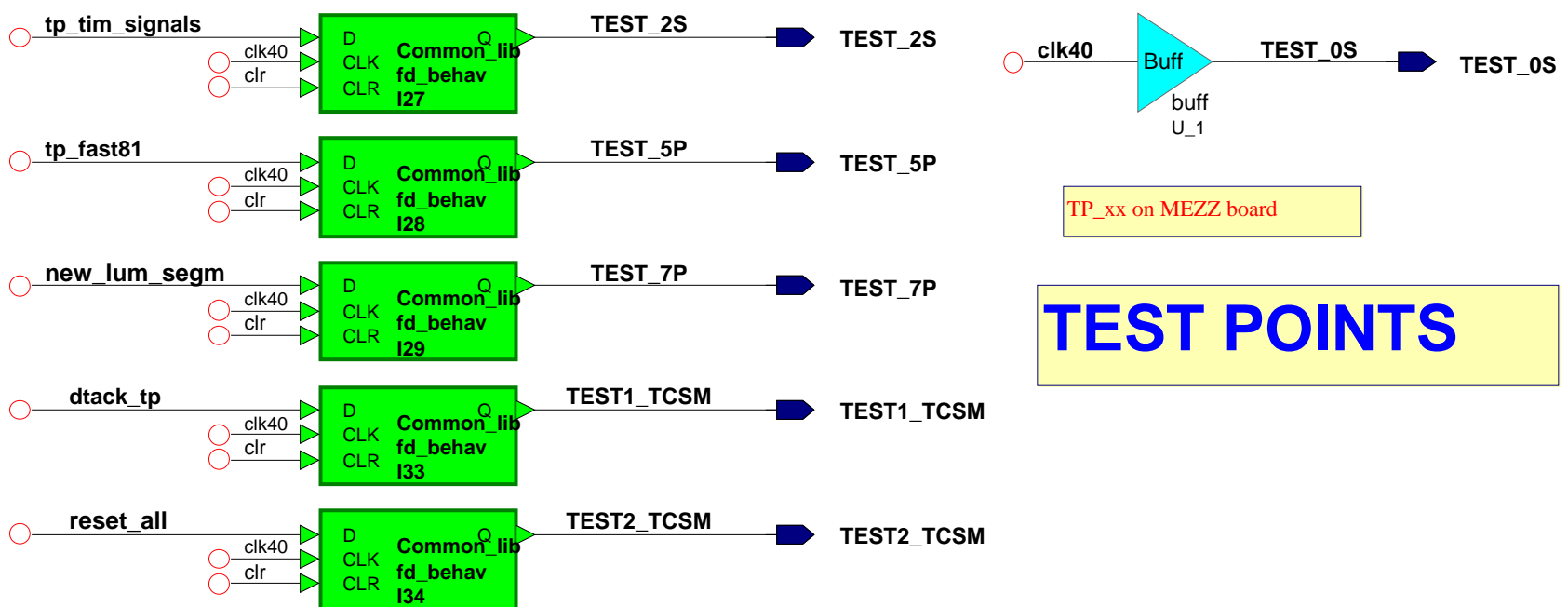
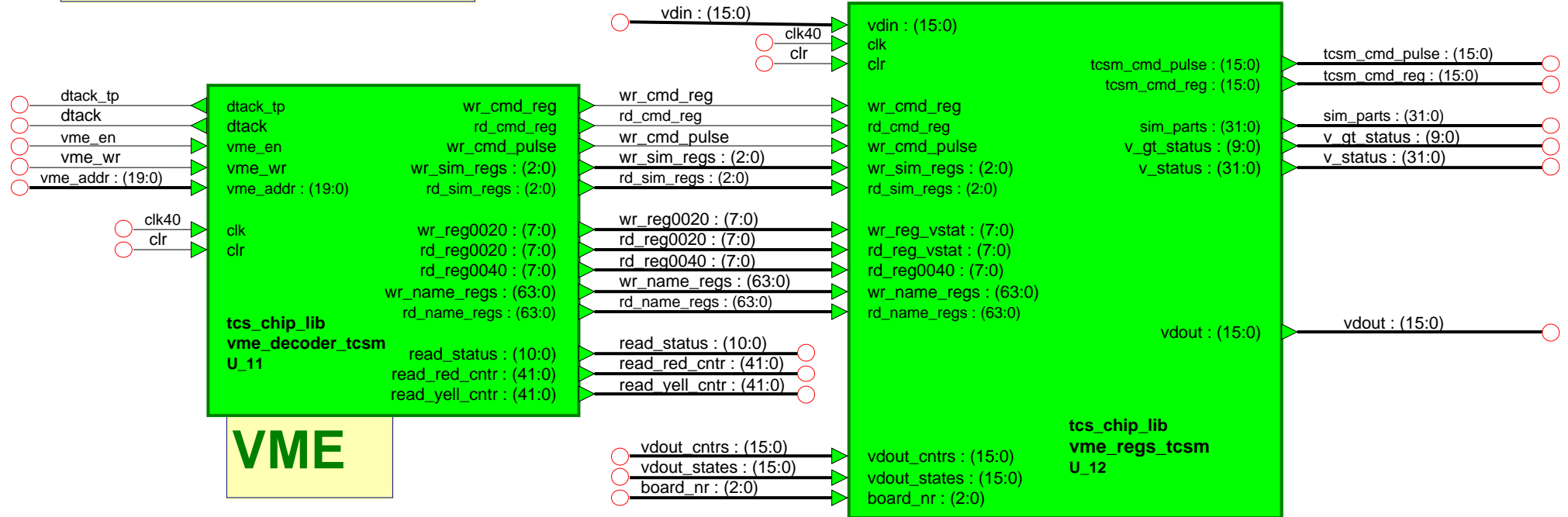


## BIDIRECTIONAL VME DATA BUS

Frame Declarations

# VERSION NR

version\_nr\_h = X"0000" ( std\_logic\_vector(15 downto 0) )  
 version\_nr\_l = X"0003" ( std\_logic\_vector(15 downto 0) )



```

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_POSEGE = 0.000 ns           ( VitalDelayType )
tperiod_PSCLK_POSEGE = 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   )
DESKW_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 )

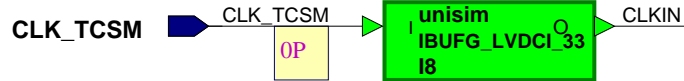
```

CLK80\_TCSM CLK80\_TCSM

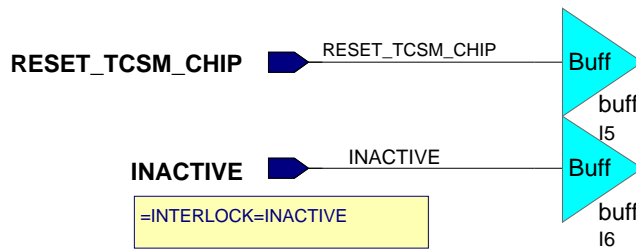
# DCM & CLOCK

We could also take clock\_dcm from psb\_lib or clock\_module from common\_lib.

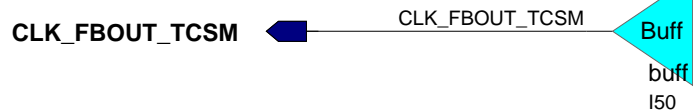
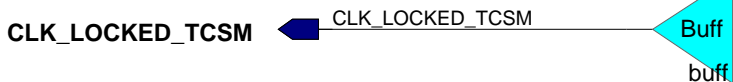
CLKFB not used, maybe later.  
==> to unused



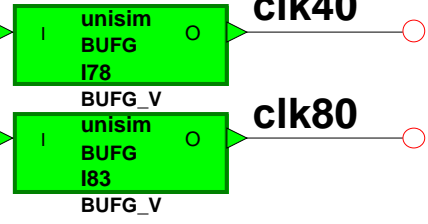
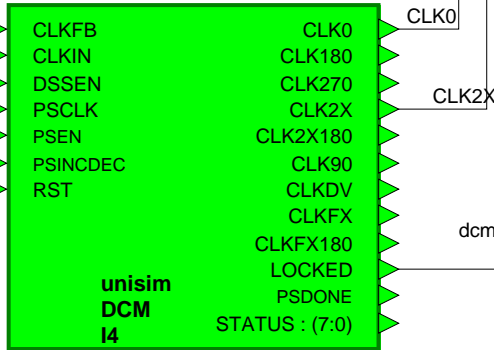
IBUFG and IBUF are placed on uppermost hierarchy.



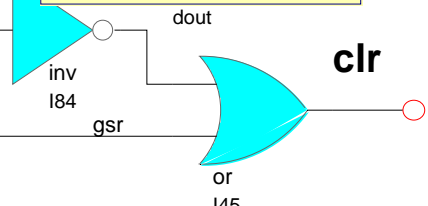
=INTERLOCK=INACTIVE



GTS=1 ==> All IOB into highZ  
GSR=1 ==> Reset/Set set all FF



Simulation ok.  
DCM locks after 1.7ms



TCS: CLEAR all by VME chip

FDL, PSB, TIM  
clr <= not clk\_locked

For Synthesis add --pragma synthesis\_off/on around generics  
rightMouse==> select Object Properties==>Generics  
and activate 'add pragmas around generics'

```

eb8
8
gnd_28 <= X"0000000";
-- gnd_16 <= X"0000";
-- gnd_4 <= "0000";
-- gnd_2 <= "00";
-- gnd_1 <= '0';
ground <= '0';
-- high <= '1';
    
```

# STATUS MONITORING

STATUS CODES:  
 0=disconnected  
 1=warning  
 2=out\_of\_sync  
 4=busy  
 8=ready  
 C=error  
 F=disconnected

VME

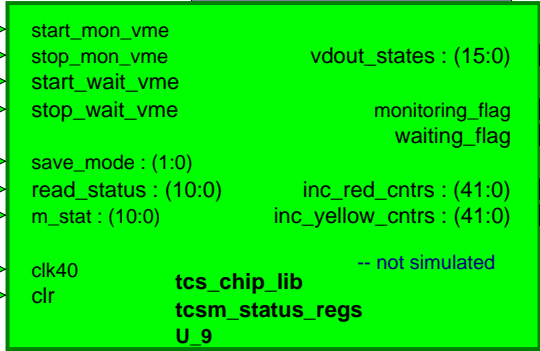
vdout\_states = simulated=ok

eb3  
 3  
 -- eb3 3  
 save\_mode <= tcsm\_cmd\_reg(7 downto 6);  
 accum\_mode <= tcsm\_cmd\_reg(8);

see eb1 1 above

eb4  
 4  
 -- eb4 4  
 reset\_counters <= tcsm\_cmd\_pulse(5) or tcsm\_cmd\_pulse(4);  
 reset\_all <= tcsm\_cmd\_pulse(4);  
 stop\_wait\_vme <= tcsm\_cmd\_pulse(3) or tcsm\_cmd\_pulse(4);  
 start\_wait\_vme <= tcsm\_cmd\_pulse(2);  
 stop\_mon\_vme <= tcsm\_cmd\_pulse(1) or tcsm\_cmd\_pulse(4);  
 start\_mon\_vme <= tcsm\_cmd\_pulse(0);

From TCSM\_MUX



Counters: --> 42 words  
 bit 15-8: errors  
 bit 7- 0: warnings

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

