Production DDU Design: Parts, Purpose and IO Usage

Connections, Front Panel

15 fibers (input) from DMB
- -Differential outputs from Fiber transceivers go directly to 2 Xilinx Virtex-2-Pro FPGAs (XC2VP20-6FG676c, 404 I/O each); these "InCtrl" FPGAs have 8 Rocket I/O each. The DMB-DDU fiber transmission is based on 80 MHz clocks provided by matched on-board oscillators on the DMB and DDU boards. The fiber connections are duplex, and the primary data direction is DMB-DDU, but the board is designed to support limited communication from DDU-DMB.
- -Each InCtrl FPGA has 2 associated TI FIFO pairs (2 TI 74LV3690-6 FIFOs serially cascaded, 0.125 MB+ capacity each) operated with 18-bit in/36-bit out, Write clock @160MHz, Read clock @156MHz. LVCMOS 3.3V communication is used for the FIFOs.
- -There are 2 separate Readout Control paths per InCtrl FPGA, with one Read Control path controlling its own TI FIFO pair.

- -Logical Fiber Input Overview:
  4 DMB Fibers ---> 1 Read Control ---> 1 TI FIFO pair
- -Physical Fiber Input Overview:
  7-8 DMB Fibers ---> 1 InCtrl FPGA ---> 2 TI FIFO pairs

1 fiber for Gigabit Ethernet (GbE) readout
- -Driven by Rocket IO on the "DDUctrl" FPGA (XC2VP7-6FF672c, 396 I/O)
- -The data for GbE packets is buffered in an on-board "Output FIFO" (IDT 72V72100, 72-in/18-out) prior to transmission. More below.

1 SLINK board for TF use
- -Driven by DDUctrl FPGA, in parallel with Output FIFO input bus
- -Has particular power and control requirements. More below.

1 FMM connector (RJ45 type)
- -4 LVDS signals driven by "VMEctrl" FPGA (XC2V500-5FG456c, 264 I/O)
- -Conforms to CMS TTS/FMM requirements. More below.

1 Block of 4 LEDs
- -Green, orange, yellow and red LEDs are hard-wired to show 3.3V power, 2.5V power, FPGA Program_Done and VME DTACK respectively. The DTACK LED is driven by on-board discrete logic.

16 pairs of LEDs
- -Each green/yellow LED pair is connected by light-pipes that correspond to its fiber connection. The green LED indicates a connected live channel (FOKout). The yellow LED flashes to show data activity (DAV) on the link.
- -The 15 DMB Fiber LEDs are driven by InCtrl FPGA logic. The GbE LEDs are driven by DDUctrl FPGA logic.

Connections, Backplane

1 "standard" 5-row VME connector on P1
1 HSC connector for 3.2Gbs differential communication to DCC on custom P3
- -2 differential High-Speed Connection paths per DDU, 3.12 Gb/s each
  ---> 620 MB/sec of real data throughput (due to 8b10b)
- -These connectors also carry clock & control signals from the TTCrx mounted on the DCC.

Clocks and Distribution

156 MHz PECL on-board; goes to DDUctrl FPGA only. Used for DDU-DCC Gigabit Transceiver (GT) links (3.12Gb/s), and an internal DCM provides divide-by-2 for 78MHz TI FIFO control and data processing in DDUctrl. Data goes out to GbE FIFO and SLINK at 78MHz on a common bus, 64-bits wide (plus control signals).
80.00 MHz PECL on-board; drives a 1-5 PECL clock driver. 4 outputs to the InCtrl FPGAs & 1 output to VMEctrl, 1 load each.
62.5 MHz LVTTL on-board; goes only to DDUctrl for GbE clocking
40.08 MHz LVTTL LHC clock from TTC via DCC/backplane connection; DDUctrl FPGA serves as 1-3 clock driver with 1-output to each FPGA..
10 MHz LVTTL SCLK (Serial/Slow Clock); derived from 80 MHz clock by DCM inside VMEctrl FPGA and fanned-out to All Other FPGAs and serial devices on a common bus (4 loads) with termination. Requires 24 mA IO drive setting.

FPGAs

XC2VP20 "InCtrl FPGAs" (need 2 for Input Control, use 254+ IO each)
- -use FG676 package (404 I/O), -6 speed grade (faster -7 costs more)
- -TWO linked XC18V04 PROMs used to program both FPGAs in parallel
  ---> both InCtrl FPGAs use identical Firmware
- -8 IO: use parallel load DIN[7:0] and CCLK line w/linked DONE pins
- -set one FPGA as SelectMAP Master, other as SelectMAP Slave
  ---> SelectMAP Master provides program clock CCLK
- -each FPGA uses 7-8 Rocket IO to read in 7-8 DMB fibers
- -JTAG is used for all PROM programming via the VME FPGA; JTAG is available on all FPGAs for slow control
- -Main Clock is 80.00 MHz DMB-DDU-matched oscillator (fanout from PECL clock driver)
  ---> drives 2 differential input pairs on InCtrl FPGA (4 BREF IO, not BREF2) for Fiber Reference clocks (1 Top edge, 1 Bottom edge) as well as logic clocks (via BUFG) and GT USRCLKs [GT is Gigabit Transceiver logic module from Virtex2 library]

Each InCtrl FPGA is functionally split into a "top half" and a "bottom half", each responsible for processing the data from the 4 top-edge RocketIO and 4 bottom-edge RocketIO respectively. Each half-FPGA has the following functions (see file ddu_in.pdf for logic schematics):
- -44 Block RAMs configured as 22 FIFO MemUnits, 4 kB each, which are assigned to the RocketIO on an as-needed basis. In this way, any RocketIO that need more storage may be allotted any combination of the 22 MemUnits as they are available.
- -A MemControl Unit to control assignment of the BRAM FIFOs. There is a priority alogorithm to select the assignments preferentially.
- -4 InUnits, each containing one RocketIO, a buffer to record MemUnits in use, and data monitoring logic. This logic watches for Status Codes sent from the DMBs and flags & corrects any 64-bit word boundary violations that occur. InUnits can also transmit data via fiber to any selected DMBs using a common data bus from the DDUctrl FPGA. Before the MemUnit in use becomes full, the InUnit will request another from MemCtrl, then switch to the new unit when the current one goes Full. InUnits will "free" a MemUnit for reuse when it goes Empty. Each InUnit has a 4-bit path to the VMEctrl FPGA to report status information.
- -A ReadControl Unit that writes exclsively to one on-board TI FIFO pair (0.25 MB external storage). The RdCtrl gets the data from each RocketIO and checks L1A numbers for an event as the data is read out. RdCtrl forwards the Kill signal from DDUctrl to the appropriate RocketIOs and watches for timeouts or changes in the link status of the InUnits. It also monitors the AlmostFull/Full status of the external FIFOs and the internal MemCtrl, then sends out a Warning (to TTS) or Busy/Stop signal (to TTS & DMBs respectively) as needed. The event number and a summary of all the checks done within InCtrl are sent out to the FIFO as the last word of an event; this information will be assimilated in the DDUctrl FPGA. Each RdCtrl has a 4-bit path to the DDUctrl FPGA to report its operating conditions.

Both halves of the InCtrl FPGA share a common JTAG control unit. This is used primarily as an interactive monitoring/debug tool.

>>>InCtrl FPGA IO on-board hardware connections (for one chip):
- 4 dedicated input lines (2 pairs) from PECL clock driver
- 2 global Reset inputs (from DDUctrl)
- 1 LVTTL clock input (SCLK) from VMEctrl
- 58 dedicated lines for TI FIFOs (50 Out, 8 In)
- 36 configurable IO to VMEctrl FPGA

59 configurable IO to DDUctrl FPGA
24 IO for Fiber support (3 per fiber)
70 Other IO, 12 required for parallel PROM load and INIT signals
---> optional 8 input from switches, 8 output to LEDs
---> optional 42 IO for header/testpoints for InCtrl FPGA#0
---> optional 6 IO for testpoints for InCtrl FPGA#1
32 additional signals for RocketIO (4 per RocketIO)
1 signal to specify chip as InCtrl FPGA#0 or #1

>>>How exactly do we use these IO in the DDU InCtrl?
-111 IO; each InCtrl FPGA uses these signals:
80MHz Main Clock (4 IO), 40MHz LHC Clock (from DCC via DDUctrl FPGA), 10MHz SlowCLK from VMEctrl
5-bits from DDUCtrl: L1A, EvCntRst, L1RST, SoftReset, BC0/BXR
13-bits to/from DDUCtrl FPGA as spare (include 1 clock I/O pin)
16-bits input data from DDUCtrl for TxData to DMB (via Rocket IO)
1 DLL Lock/Ready (to DDUctrl), 2 FIFO_Active (1 per RdCtrl)
4 control/IO signals left free: INIT, CS, WRITE, DOUT (Don't Use)
2 * 18-bit DOUT to FIFOs (bits 16-17 flag Fill/LW, Hdr/Tr)
2 * 11-bit FIFO control: WCLK, ~WEN, ~PAE, ~PAF, FF, ~3/4Full SEN, SerLD, SerDat, Mode0, Mode1
2 * 4-bit FPGA status bits (to DDUctrl, 1 4-bit set per RdCtrl)
-72 IO; need per Rocket IO used on each FPGA (multiply below by 8):
2-bits for Tx control from DDUctrl FPGA: TxEn, Kill Fiber
3-bits for Fiber/FIFO Status: FOK (in), FOKout, DAV (to FP-LEDs)
4 for DMB status: T[3:0] for FMM (to VMEctrl)
-58 IO; added header connectors for 2 Logic Analyzer (LA) ports (18 bits each, only InCtrl FPGA#0), 6 testpoints, 8 LEDs, 8 switches
-8 IO dedicated for parallel PROM load
-InCtrl FPGA#1 "kills" RocketIO #6 (the 16th RocketIO) as it is unused

XC2VP7 "DDUctrl FPGA" (uses 391+ I/O)
-use FF672 package (396 I/O), requires -6 speed grade or faster (-7) with flip-chip package
-TWO linked XC18V04 PROMs, use parallel load DIN[7:0] and CCLK lines
---> use SelectMAP Master mode for program clock, FPGA drives CCLK
-2 Rocket IO to backplane (data to DCC via HSC connector, 3.2 Gb/s)
-1 Rocket IO to front panel output Gigabit Ethernet fiber (1.25 Gb/s)

The DDUctrl FPGA reads in data from the on-board TI FIFOs, 1 event at a time. Each event is checked for the following during readout:
-improper event formatting & data corruption
-status/error flags passed from InCtrl via TI FIFO
-event & bunch number synchronization from each source
-CFEB wordcount & CRC from each source
-DMB CRC & LCT/DAV consistency from each source
-TMB wordcount & CRC from each source
-ALCT wordcount & CRC from each source
-missing DMB, CFEB, TMB or ALCT data from each source
-Full and Near Full condition from each CSC
-timeout conditions
DDUctrl performs the following tasks (logic schematics ddu_ctrl.pdf):
-event readout, processing and monitoring
-receives & decodes TTC Command Bus instructions from DCC (via backplane); e.g. BC0, Sync/Soft Reset, Start/Stop data taking
-receives & distributes CMS Clock & L1A from TTC/DCC to other FPGAs
-watches for changes in status of the RdCtrl units
-monitors the status feedback from the DCC/SLINK paths and Stops TI FIFO readout as needed
-sends event information in the headers and trailers, conforming to CMS Common Data Format rules
-generates a CRC word for all data transmitted for the event
-sends all data out via DCC/SLINK paths
-sends data to the external GbE FIFO on a prescalable basis
-creates & transmits GbE packets from data stored in external GbE FIFO
-reads Board ID & Killed Channel constants from Flash RAM and sends Kill Channel signals to InCtrl FPGAs
-controls a 16-bit data bus to send control or data signals out on any Fiber via InCtrl FPGAs
-monitors the DDU Board Operation condition, summarizing the 4 RdCtrl and DDUctrl status on one 4-bit path to the VMEctrl FPGA for TTS

>>>DDUctrl FPGA IO on-board hardware connections:
3 dedicated input lines from PECL/LVTTL sources (4 IO)
1 LVTTL clock input (SCLK) from VMEctrl
2 IO for local 78 MHz LVTTL clock feedback
2 global resets Out (Sync & Soft reset)
18 control lines from TTC (15), DCC (2) & VME (1)
72 dedicated lines for TI FIFOs (36 data, 36 control)
96 configurable IO to InCtrl FPGAs
19 configurable IO to VMEctrl FPGA
116 IO for SLINK & GbE
---> Min. 83 for SLINK; 33 more are for GbE only.
62 Other IO, 12 required for parallel PROM load and INIT signals
---> optional 8 input from switches, 8 PROM lines go to LEDs
---> optional 41 IO to header/testpoints, 1 pushbutton switch
12 additional signals for RocketIO (4 per RocketIO)

>>>How exactly do we use these IO in DDUctrl?
-7 IO used for 4 ext. CLKs for DDUctrl: Main Clock (156 MHz) MUST use LVDS IO (IBUFGDS) & divide-by-2 for 78 MHz, one 62.5 MHz for GbE, 40.08 MHz LHC clock, 10 MHz serial shift clock (SCLK)
---> 78 MHz feedback LVTTL goes out * back for DCM lock (2 IO)
-18 IO; DDU board control
1 pushbutton input (debounced, goes to DDUctrl) used for logic RST
1 SYSRESET input from VME, [+4 JTAG lines]
3 DLL-Lock signals (1 from each FPGA, tells DDUctrl they're ready)
4 FIFO_Active (1 per RdCtrl from InCtrl FPGAs)
5-bits control Output from DDUctrl, common for other FPGAs: L1Aout, EvCntRst, L1RST(SyncRst), SoftReset, BC0/BXR
4 control/IO signals left free: CS, WRITE, DOUT, INIT (Don't Use)
-30 IO; each InCtrl Rocket IO has individual controls (below times 15) :
Enable (for TxEn), Kill (for Rx/Tx)
-16-bit output for DMB Tx data bus (parallel to both InCtrl FPGAs)
-16 status bits from InCtrl FPGAs (4-bits from each RdCtrl)
-26 bits from/to InCtrl FPGAs as spare (13 each, inc. 1 clock I/O pin)
-38 IO; 1 36-bit input data bus to read 4 TI FIFO pairs, plus 2 RCLK signals shared by the TI FIFOs (2 loads each)
-32 IO; each TI FIFO pair has individual controls (mult. by 4 below):
~OE, ~REN, EF, ~HF, ~PAF, FF, TransferWEN, TransferCLK
-2 IO; common FIFO control (for all TI FIFOs):
~MRST (after soft reset), ~PRST (after sync/logic reset)
-16 IO; control input from TTC/DCC
cmd_bus[8:0], L1A, Link Ready/Full, plus 4 others
-66-bit output data (to GbE FIFO/SLINK)
-18-bit input data from GbE FIFO (for GbE packets out via RocketIO)
-12 IO; GbE FIFO control (SDR IDT Supersync):
WCLK, RCLK, WEN, REN, FF, ~PAF, ~PAE, MT, ~HF
Output fiber FOK, FOKout LED, DataValid LED
-18 IO; SLINK control
10 Inputs: LFF, LDOWN, LRL[3:0], LSF[3:0]
8 Outputs: UWEN, UCTRL, URST, UTEST, USF[3:0]
-19 IO; communication to/from VMEctrl
4 FMM bits to VMEctrl
4 FMM bits from VMEctrl (also go to FMM cable)
2 control bits for SLoad, SData input from VMEctrl
-from serial-flash memory for "kill" fiber [16], Board ID [32], etc.
9 bits from/to VMEctrl FPGA as spare (includes 1 clock I/O pin)
-57 other IO; 8 switches, 8 LEDs (share PROM DIN), 2 LA Ports (36 IO), 5 extra IO to "test points"

```
XC2V500-5FG456c "VMEctrl FPGA"  (uses 264 IO)
    -One XC18V04 PROM, use parallel load DIN[7:0] and CCLK lines
        ---> use SelectMAP Master mode for program clock
    -Program VME PROM from VME FPGA, JTAG cable or Discrete on-board
        VME Logic; uses parallel load DIN[7:0] and CCLK lines to FPGA
    -Main Clock is 80.00 MHz DMB-DDU-matched oscillator (fanout from PECL
     clock driver)

  VMEctrl performs the following tasks (logic schematics ddu_vme.pdf):
    -decodes VME Bus signals into on-board Parallel, Serial and JTAG paths
    -controls all Read and Write functions for the on-board 1 Mb Serial
        Flash RAM; constants are read out and sent to the appropriate
        devices after every Soft Reset
    -continually receives 4-bit status summaries from DDUctrl and 15 fiber
        sources, uses this to determine the instantaneous status of the
        entire 20-degree DDU Sector, sending the result to TTS via FMM

  >>>VMEctrl FPGA IO on-board hardware connections:
        2 dedicated clock input lines from PECL clock driver
        2 global resets In (Sync & Soft reset from DDUctrl)
        15 configurable IO to DDUctrl FPGA
        68 configurable IO from InCtrl FPGAs
        67 dedicated lines for VME Backplane IO
        4 dedicated output lines to RJ45 for FMM/TTS; LVTTL-LVDS
           conversion done on-board
        32 IO for 8 JTAG paths (for FPGAs, PROMs & FIFOs)
        13 IO for other Serial controls (FIFO threshold, etc. from SRAM)
        59 Other IO, 12 required for parallel PROM load and INIT signals
            ---> optional 8 input from switches, 8 PROM lines go to LEDs
            ---> optional 39 IO for header/testpoints

  >>>How exactly do we use these IO in VMEctrl?
    -47 VME input IO:  ADR[23:1], AM[5:0], GA[5:0], IRQ1, IackIN,
        sysreset, sysfail, clk, write, iack, lword, berr, as, ds0, ds1
    -20 VME output IO:  dtack, ToVME, doe, D[15:0], IackOUT
    -70 FMM IO: 4-bit FMM output (to DDUctrl [TTL] and FMM cable [LVDS]),
        60-bit DMB status input (15 DMBs * 4 bits each from InCtrl FPGAs),
        4-bit DDUctrl status inputs, 2 LEDs
    -9 control IO
        80 MHz PECL clock, 2 global Reset inputs (from DDUctrl)
        1 DLL Lock/Ready (to DDUctrl)
        4 control/IO signals left free:  CS, WRITE, DOUT, INIT
    -14 Serial control IO  (for 4 Input FIFOs, 1 Output FIFO, 1 DDUctrl)
        Common SCLK (4 loads),SData (2 loads), 1 FIFO_SEN,
        6 bits for FlashMemory Control (M_SCLK, SI, SO, ~CS, ~WP, ~RESET)
        4 bits for SerADC Control (ADCclk, ADCdin, ADCsel, ADCout),
        1 Serial Enable bit to DDUctrl
    -9 spare signals to DDUctrl FPGA
    -8 spare signals to INctrl FPGAs
    -32 IO for 8 JTAG paths (4 IO each):
        1 for DDUctrl PROM
        1 for DDUctrl FPGA
        1 for VMEctrl PROM (program by VMEctrl, VME-Recovery & JTAG cable)
        2 for InCtrl FPGAs (1 each)
        1 for InCtrl PROMs (in series, matched with DONE/PROGRAM order)
        1 for S-Link
        1 for output FIFO
    -55 other IO; 8 switches, 8 LEDs (share PROM DIN), 2 LA Ports (36 IO),
        3 extra IO to "test points"



FIFOs & on-board Flash Memory
-----------------------------
Input FIFO: use 8 TI 74LV3690-6 (128 PQFP, 32K x 36-bits)
       -FIFOs are paired for a combined depth of 64K
         > first FIFO: 18-bits In (160MHz), 36-bits Out (156MHz)
         > second FIFO: 36-bits In (156MHz), 36-bits Out (156MHz)
Output FIFO: use IDT 72V72100 (256 PBGA, 64K x 72-bits, 72/18)
       -use 66-bits In (78MHz), 18-bits Out (62.5MHz)
Serial Flash RAM: use 1 AT45DB011B (over 1 Mb, 8-pin SOIC), 20 MHz max.
       -uses 10 MHz derived from SCLK; can do 20 MHz, but the GbE FIFO
          serial clock is limited to 10 MHz max.
       -stores constants in Pages for DDU Board ID, Killed channels, etc.
     Useful OpCodes: d2h, d7h, 82h
       -all constants only use ~100 bits of memory; over 1 Mb is available
          for additional constants if desired
     Use Block0: page1=KillCh (16b),    page7=BoardID (8b),
                 page4=free (32b),      page5=GbE FIFO Thresh (34b)


Other Notes
-----------
Hard_Reset has 3 sources logical OR'ed together w/discrete on-board logic:
    -TTC via DCC/Backplane connection
    -Power-On sensor
    -PROGRAM button input (debounced)
   Hard_Reset goes to all FPGA PROGRAM pins.
The 2nd on-board pushbutton (debounced) makes a Soft_Reset in DDUctrl,
   which fans it out to other FPGAs, and sends sync'd MRST to all
   FIFOs. (This could be reconfigured for another purpose as needed.)
    -VMEctrl sets the LD & FWFT/SI signals in time with the MRST
The board has discrete "Emergency VME-Recovery" logic on board to allow
   for bootstrap VMEctrl PROM programming.  There is also a 10-pin JTAG
   header connector for programming this PROM via laptop if desired.  All
   other PROMs _must_ be programmed via VME, with no alternative provided.



================================================================================
 The 3 sections below are the UCF files containing the Location and Timing
 Constraints for all FPGA firmware.
================================================================================


DDU InCtrl FPGA UCF file
------------------------
# use with "D785/DDU5in: IN5CTRL" Logic schematic
#PINLOCK_BEGIN
#Sun Sep 27 12:48:16 2005
#INPUTS
# Clock & Control signals in:
#   from switches (added 4)
NET "$1I162/~SOFTRST"  LOC = "AB4";
NET "$1I162/IN_EV_RST" LOC = "AB1"; #l1arst
NET "$1I162/~L1RST"    LOC = "AA5"; #syncrst
#NET "$1I162/BXRIN"    LOC = "";   # not used
NET "$1I162/IN_BC0"    LOC = "AB2"; # in from DDUctrl: "bc0out"
NET "$1I162/MODEIN7"  LOC = "H6";#to Switch8
NET "$1I162/MODEIN6"  LOC = "H7";
NET "$1I162/MODEIN5"  LOC = "G1";
NET "$1I162/MODEIN4"  LOC = "G2";
NET "$1I162/MODEIN3"  LOC = "G3";
NET "$1I162/MODEIN2"  LOC = "G4";
NET "$1I162/MODEIN1"  LOC = "H5";
NET "$1I162/MODEIN0"  LOC = "G5";#to Switch1
NET "$1I162/L1AIN"    LOC = "AA2"; #l1aout
NET "$1I162/IN_FPGAID" LOC = "C25"; #fpgaid
#NET "$1I162/SCLKIN"   LOC = "E14"; #sclk, not used
NET "$1I162/CLKIN40"   LOC = "AD14";#spare_clk, gclk_s
NET "$1I162/CK80P_I"   LOC = "B13"; #clk80p0, gclk_s
NET "$1I162/CK80N_I"   LOC = "C13"; #clk80n0, gclk_p
NET "$1I162/CK80P_B"   LOC = "AD13";#clk80p1, gclk_p
NET "$1I162/CK80N_B"   LOC = "AE13";#clk80n1, gclk_s
#
# Spare signals lines:
#NET "$1I162/DDUSPARE0" LOC = "AD25"; #spare0/12 from DDUctrl
#NET "$1I162/DDUSPARE1" LOC = "AC24"; #spare1/13 from DDUctrl
```

```
#NET "$1I162/DDUSPARE2" LOC = "AC25"; #spare2/14 from DDUctrl
#NET "$1I162/DDUSPARE3" LOC = "AC26"; #spare3/15 from DDUctrl
#NET "$1I162/DDUSPARE4" LOC = "AB23"; #spare4/16 from DDUctrl
#NET "$1I162/DDUSPARE7" LOC = "AB26"; #spare7/19 from DDUctrl
#NET "$1I162/DDUSPARE8" LOC = "AA25"; #spare8/20 from DDUctrl
#NET "$1I162/DDUSPARE9" LOC = "AA12"; #spare9/21 from DDUctrl
NET "$1I162/~IDMB_FUL0" LOC = "AA22"; #spare5/17 to DDUctrl In0-3 DMB_Full
NET "$1I162/~IDMB_FUL1" LOC = "AB25"; #spare6/18 to DDUctrl In4-7 DMB_Full
#NET "$1I162/~FMT0IN" LOC = "Y12"; #spare10/22 from DDUctrl, "DoFake0"
#NET "$1I162/~FMT1IN" LOC = "W12"; #spare11/23 from DDUctrl, "DoFake1"
#NET "$1I162/VMESPARE0"  LOC = "F22"; #spare "INPUT0/4" to-from VMEctrl
#NET "$1I162/VMESPARE1"  LOC = "E25"; #spare "INPUT1/5" to-from VMEctrl
NET "$1I162/ORST_TO_VME" LOC = "E26"; # out to VMEctrl for EvCntRst or BC0
#                                              "INPUT2/6"
NET "$1I162/INFAKEL1_EN" LOC = "F21"; # in from VMEctrl for Fake_L1A_En
#                                              "INPUT3/7"
# TestPoints:
#NET "$1I162/TP_56"  LOC = "E5"; #Testpoint56
#NET "$1I162/TP_57"  LOC = "E6"; #Testpoint57
#NET "$1I162/TP_58"  LOC = "G7"; #Testpoint58
#NET "$1I162/TP_59"  LOC = "E7"; #Testpoint59
#NET "$1I162/TP_60"  LOC = "B14"; #Testpoint60
#NET "$1I162/TP_61"  LOC = "C14"; #Testpoint61
#
#FIFO Status Signals in:
#   from output FIFO
NET "$1I162/~PAE0IN" LOC = "K22"; #not really needed, monitor only?
NET "$1I162/~HFA0IN" LOC = "M22"; #eff0in, use for Warn flag or Release?
NET "$1I162/~PAF0IN" LOC = "J23";
NET "$1I162/FF0IN"   LOC = "J22";
NET "$1I162/~PAE1IN" LOC = "U23"; #not really needed, monitor only?
NET "$1I162/~HFA1IN" LOC = "V22"; #eff1in, use for Warn flag or Release?
NET "$1I162/~PAF1IN" LOC = "T22";
NET "$1I162/FF1IN"   LOC = "R22";
#
#   from input FIBERs
NET "$1I162/IFOK0"    LOC = "D1"; #fok0
NET "$1I162/IFOK1"    LOC = "D2";
NET "$1I162/IFOK2"    LOC = "C1";
NET "$1I162/IFOK3"    LOC = "C2";
NET "$1I162/IFOK4"    LOC = "AA1";
NET "$1I162/IFOK5"    LOC = "Y5";
NET "$1I162/IFOK6"    LOC = "W5";
NET "$1I162/IFOK7"    LOC = "Y4"; #floating IO on 2nd FPGA
#   KILL input FIBERs from DDU Control
NET "$1I162/KILL0"    LOC = "AB14";
NET "$1I162/KILL1"    LOC = "AD15";
NET "$1I162/KILL2"    LOC = "AC15";
NET "$1I162/KILL3"    LOC = "AA14";
NET "$1I162/KILL4"    LOC = "AB15";
NET "$1I162/KILL5"    LOC = "AA15";
NET "$1I162/KILL6"    LOC = "AA16";
NET "$1I162/KILL7"    LOC = "AD17";
#   Data signals in
NET "$1I162/RXP0"    LOC = "A22"; #rd0+
NET "$1I162/RXN0"    LOC = "A23"; #rd0-
NET "$1I162/RXP1"    LOC = "A17"; #rd1+
NET "$1I162/RXN1"    LOC = "A18"; #rd1-
NET "$1I162/RXP2"    LOC = "A11";
NET "$1I162/RXN2"    LOC = "A12";
NET "$1I162/RXP3"    LOC = "A6";
NET "$1I162/RXN3"    LOC = "A7";
#
NET "$1I162/RXP4"    LOC = "AF6";
NET "$1I162/RXN4"    LOC = "AF7";
NET "$1I162/RXP5"    LOC = "AF11";
NET "$1I162/RXN5"    LOC = "AF12";
NET "$1I162/RXP6"    LOC = "AF17"; #floating IO on 2nd FPGA
NET "$1I162/RXN6"    LOC = "AF18"; #floating IO on 2nd FPGA
NET "$1I162/RXP7"    LOC = "AF22";
NET "$1I162/RXN7"    LOC = "AF23";
#
NET "$1I162/TXDIN0"  LOC = "AC1";
NET "$1I162/TXDIN1"  LOC = "AC2";
NET "$1I162/TXDIN2"  LOC = "AC3";
NET "$1I162/TXDIN3"  LOC = "AD2";
NET "$1I162/TXDIN4"  LOC = "AD1";
NET "$1I162/TXDIN5"  LOC = "AE1";
NET "$1I162/TXDIN6"  LOC = "AB7";
NET "$1I162/TXDIN7"  LOC = "AC7";
NET "$1I162/TXDIN8"  LOC = "AD7";
NET "$1I162/TXDIN9"  LOC = "AA8";
NET "$1I162/TXDIN10" LOC = "AB8";
NET "$1I162/TXDIN11" LOC = "AE8";
NET "$1I162/TXDIN12" LOC = "AF8";
NET "$1I162/TXDIN13" LOC = "AA9";
NET "$1I162/TXDIN14" LOC = "AB9";
NET "$1I162/TXDIN15" LOC = "AC9";
NET "$1I162/TXEN0"   LOC = "AC17";
NET "$1I162/TXEN1"   LOC = "AB16";
NET "$1I162/TXEN2"   LOC = "AB17";
NET "$1I162/TXEN3"   LOC = "AD18";
NET "$1I162/TXEN4"   LOC = "AC18";
NET "$1I162/TXEN5"   LOC = "AB18";
NET "$1I162/TXEN6"   LOC = "AA18";
NET "$1I162/TXEN7"   LOC = "AF19";
#
#OUTPUTS
#   diagnostic signals out to LEDs:
NET "$1I162/LED0"   LOC = "J3";
NET "$1I162/LED1"   LOC = "J4";
NET "$1I162/LED2"   LOC = "J5";
NET "$1I162/LED3"   LOC = "J6";
NET "$1I162/LED4"   LOC = "J7";
NET "$1I162/LED5"   LOC = "J8";
NET "$1I162/LED6"   LOC = "H1";
NET "$1I162/LED7"   LOC = "H2";
#
# New FIFO Configuration controls
NET "$1I162/~PLD0"    LOC = "J19"; #default HIGH for m=n=7
NET "$1I162/~OSEN0"  LOC = "J25"; #default HIGH
NET "$1I162/SI0"     LOC = "J20"; #HIGH during MRST for FWFT
NET "$1I162/FSEL0_0" LOC = "K23"; #afs0, default HIGH for m=n=7
NET "$1I162/FSEL0_1" LOC = "K24"; #afs1, default HIGH for m=n=7
NET "$1I162/~PLD1"    LOC = "P23"; #default HIGH for m=n=7
NET "$1I162/~OSEN1"  LOC = "R25"; #default HIGH
NET "$1I162/SI1"     LOC = "P22"; #HIGH during MRST for FWFT
NET "$1I162/FSEL1_0" LOC = "U22"; #bfs0, default HIGH for m=n=7
NET "$1I162/FSEL1_1" LOC = "U24"; #bfs1, default HIGH for m=n=7
#
#Data signals out:
#   output FIFO control
NET "$1I162/~OWEN0"   LOC = "J24";
NET "$1I162/CLKOUTX0" LOC = "J21"; #clkout0
NET "$1I162/~OWEN1"   LOC = "R26";
NET "$1I162/CLKOUTX1" LOC = "R23"; #clkout1
#   data out to DMBs
NET "$1I162/TXP0"    LOC = "A21"; #td0+
NET "$1I162/TXN0"    LOC = "A20"; #td0-
NET "$1I162/TXP1"    LOC = "A16"; #td1+
NET "$1I162/TXN1"    LOC = "A15"; #td1-
NET "$1I162/TXP2"    LOC = "A10";
NET "$1I162/TXN2"    LOC = "A9";
NET "$1I162/TXP3"    LOC = "A5";
NET "$1I162/TXN3"    LOC = "A4";
#
NET "$1I162/TXP4"    LOC = "AF5";
NET "$1I162/TXN4"    LOC = "AF4";
```

```
NET "$1I162/TXP5"    LOC = "AF10";
NET "$1I162/TXN5"    LOC = "AF9";
NET "$1I162/TXP6"    LOC = "AF16";
NET "$1I162/TXN6"    LOC = "AF15";
NET "$1I162/TXP7"    LOC = "AF21";
NET "$1I162/TXN7"    LOC = "AF20";
#
#   data out to FIFOs
NET "$1I162/DOUT0"   LOC = "N18";
NET "$1I162/DOUT1"   LOC = "L20";
NET "$1I162/DOUT2"   LOC = "M19";
NET "$1I162/DOUT3"   LOC = "M20";
NET "$1I162/DOUT4"   LOC = "M21";
NET "$1I162/DOUT5"   LOC = "N21";
NET "$1I162/DOUT6"   LOC = "N22";
NET "$1I162/DOUT7"   LOC = "N23";
NET "$1I162/DOUT8"   LOC = "N19";
NET "$1I162/DOUT9"   LOC = "K19";
NET "$1I162/DOUT10"  LOC = "L19";
NET "$1I162/DOUT11"  LOC = "N24";
NET "$1I162/DOUT12"  LOC = "M26";
NET "$1I162/DOUT13"  LOC = "M25";
NET "$1I162/DOUT14"  LOC = "L24";
NET "$1I162/DOUT15"  LOC = "K26";
NET "$1I162/DOUT16"  LOC = "K25";
NET "$1I162/DOUT17"  LOC = "J26";
#
NET "$1I162/DOUT18"  LOC = "P19";
NET "$1I162/DOUT19"  LOC = "P21";
NET "$1I162/DOUT20"  LOC = "R20";
NET "$1I162/DOUT21"  LOC = "R21";
NET "$1I162/DOUT22"  LOC = "T21";
NET "$1I162/DOUT23"  LOC = "T20";
NET "$1I162/DOUT24"  LOC = "T19";
NET "$1I162/DOUT25"  LOC = "U19";
NET "$1I162/DOUT26"  LOC = "V21";
NET "$1I162/DOUT27"  LOC = "R18";
NET "$1I162/DOUT28"  LOC = "R19";
NET "$1I162/DOUT29"  LOC = "P18";
NET "$1I162/DOUT30"  LOC = "V24";
NET "$1I162/DOUT31"  LOC = "V25";
NET "$1I162/DOUT32"  LOC = "V26";
NET "$1I162/DOUT33"  LOC = "U25";
NET "$1I162/DOUT34"  LOC = "U26";
NET "$1I162/DOUT35"  LOC = "T24";
#
# Send Ready & Active 0-1 status to DDU_Ctrl
NET "$1I162/~RDYOUT"   LOC = "AD12"; #dll_ready
NET "$1I162/~ACTOUT0"  LOC = "AA13"; #fifo_active0
NET "$1I162/~ACTOUT1"  LOC = "AC12"; #fifo_active1
#
# Send RdCtrl 0-1 status to DDU_Ctrl
NET "$1I162/RD0_BUSY"  LOC = "AE19"; #rdstat0
NET "$1I162/RD0_WARN"  LOC = "AB19"; #rdstat1
NET "$1I162/RD0_SYNC"  LOC = "AA19"; #rdstat2
NET "$1I162/RD0_ERROR" LOC = "AD20"; #rdstat3
NET "$1I162/RD1_BUSY"  LOC = "AC20"; #rdstat4
NET "$1I162/RD1_WARN"  LOC = "AB20"; #rdstat5
NET "$1I162/RD1_SYNC"  LOC = "AE26"; #rdstat6
NET "$1I162/RD1_ERROR" LOC = "AD26"; #rdstat7
#
# Send DMB/Fiber 0-7 status to VME_Ctrl
NET "$1I162/F0_BUSY"  LOC = "H14"; #dmb_stat0
NET "$1I162/F0_WARN"  LOC = "J14"; #dmb_stat1
NET "$1I162/F0_SYNC"  LOC = "C15"; #dmb_stat2
NET "$1I162/F0_ERROR" LOC = "D15"; #dmb_stat3
NET "$1I162/F1_BUSY"  LOC = "F14"; #dmb_stat4
NET "$1I162/F1_WARN"  LOC = "E15";
NET "$1I162/F1_SYNC"  LOC = "F15";
NET "$1I162/F1_ERROR" LOC = "G15";
NET "$1I162/F2_BUSY"  LOC = "H15"; #dmb_stat8
NET "$1I162/F2_WARN"  LOC = "J15";
NET "$1I162/F2_SYNC"  LOC = "F16";
NET "$1I162/F2_ERROR" LOC = "G16";
NET "$1I162/F3_BUSY"  LOC = "C17"; #dmb_stat12
NET "$1I162/F3_WARN"  LOC = "D17";
NET "$1I162/F3_SYNC"  LOC = "E16";
NET "$1I162/F3_ERROR" LOC = "E17";
NET "$1I162/F4_BUSY"  LOC = "H16"; #dmb_stat16
NET "$1I162/F4_WARN"  LOC = "H17";
NET "$1I162/F4_SYNC"  LOC = "C18";
NET "$1I162/F4_ERROR" LOC = "D18";
NET "$1I162/F5_BUSY"  LOC = "E18"; #dmb_stat20
NET "$1I162/F5_WARN"  LOC = "F18";
NET "$1I162/F5_SYNC"  LOC = "G18";
NET "$1I162/F5_ERROR" LOC = "H18";
NET "$1I162/F6_BUSY"  LOC = "A19"; #dmb_stat24
NET "$1I162/F6_WARN"  LOC = "B19";
NET "$1I162/F6_SYNC"  LOC = "E19";
NET "$1I162/F6_ERROR" LOC = "F19";
NET "$1I162/F7_BUSY"  LOC = "G19"; #dmb_stat28
NET "$1I162/F7_WARN"  LOC = "H19";
NET "$1I162/F7_SYNC"  LOC = "C20";
NET "$1I162/F7_ERROR" LOC = "D20";
#
# Send FiberOK 0-7 & DAV 0-7 signals to Front Panel LEDs
NET "$1I162/FOKLED0"  LOC = "E1"; #fokout0
NET "$1I162/FOKLED1"  LOC = "E2";
NET "$1I162/FOKLED2"  LOC = "F5";
NET "$1I162/FOKLED3"  LOC = "E4";
NET "$1I162/FOKLED4"  LOC = "Y3";
NET "$1I162/FOKLED5"  LOC = "Y2";
NET "$1I162/FOKLED6"  LOC = "Y1";
NET "$1I162/FOKLED7"  LOC = "W7";
NET "$1I162/DVLED0"  LOC = "F1"; #dav0
NET "$1I162/DVLED1"  LOC = "F2";
NET "$1I162/DVLED2"  LOC = "G6";
NET "$1I162/DVLED3"  LOC = "F6";
NET "$1I162/DVLED4"  LOC = "W6";
NET "$1I162/DVLED5"  LOC = "W2";
NET "$1I162/DVLED6"  LOC = "W1";
NET "$1I162/DVLED7"  LOC = "V8";
#
#   Reserve 18 *2 Pads for 2 Logic Analyzer paths:
NET "$1I162/L0_0"   LOC = "K8"; #la0_0
NET "$1I162/L0_1"   LOC = "L8";
NET "$1I162/L0_2"   LOC = "K5";
NET "$1I162/L0_3"   LOC = "K4";
NET "$1I162/L0_4"   LOC = "K3";
NET "$1I162/L0_5"   LOC = "L3";
NET "$1I162/L0_6"   LOC = "K2";
NET "$1I162/L0_7"   LOC = "K1";
NET "$1I162/L0_8"   LOC = "L7";
NET "$1I162/L0_9"   LOC = "M7";
NET "$1I162/L0_10"  LOC = "L6";
NET "$1I162/L0_11"  LOC = "L5";
NET "$1I162/L0_12"  LOC = "L2";
NET "$1I162/L0_13"  LOC = "L1";
NET "$1I162/L0_14"  LOC = "M9";
NET "$1I162/L0_15"  LOC = "M8";
NET "$1I162/L0_16"  LOC = "J1";  #la0_clk1
NET "$1I162/L0_17"  LOC = "J2";  #la0_clk2
#
NET "$1I162/L1_0"   LOC = "P4";  #la1_0
NET "$1I162/L1_1"   LOC = "P5";
NET "$1I162/L1_2"   LOC = "P8";
NET "$1I162/L1_3"   LOC = "P9";
NET "$1I162/L1_4"   LOC = "R1";
```

```
NET "$1I162/L1_5"   LOC = "R2";
NET "$1I162/L1_6"   LOC = "R4";
NET "$1I162/L1_7"   LOC = "R5";
NET "$1I162/L1_8"   LOC = "P6";
NET "$1I162/L1_9"   LOC = "R6";
NET "$1I162/L1_10"  LOC = "R8";
NET "$1I162/L1_11"  LOC = "R9";
NET "$1I162/L1_12"  LOC = "T1";
NET "$1I162/L1_13"  LOC = "T2";
NET "$1I162/L1_14"  LOC = "T5";
NET "$1I162/L1_15"  LOC = "T6";
NET "$1I162/L1_16"  LOC = "P3";  #la1_clk1
NET "$1I162/L1_17"  LOC = "P2";  #la1_clk2
#
# Foundation 3.1 does not understand high/low 50%....
#    but required for 4.2 & 5.1
NET "$1I162/CLKIN40" period=23ns high 50%;  # T_clk40 = 4.40ns was met
NET "$1I162/CLK"  period=10.8ns high 50%;   # T_clk = 10.61ns was met
NET "$1I162/BREF" period=11.3ns high 50%;   # T_bref    = N/A
NET "$1I162/BREF_B" period=11.3ns high 50%; # T_bref_B  = N/A
#
## VIRTEX 2 Geometry Summary (XC2VP20-FG676):
#       4 Slices per CLB, 2 TBUFs per CLB
#       2 LUTs per Slice, 2 FDs per Slice (share common RST/CE/CLK)
#       2 Fast Carry MUXCYs per Slice, plus 1 Fast ORCY
#       1 MUXF5 and 1 MUXFx per Slice (x=6,7 or 8)
#
# For "Normal" Coords (x,y) Slices, TBUFs, RAMBs, etc all have
#       independent coordinate origins, so use GRID coords if
#       you must combine diffferent types of components.
#       Component coordinate ranges:
#               SLICE  x0y0 to x91y111
#               TBUF   x0y0 to x90y111, even x only
#               RAMB16 x0y0 to x7y13 (88 total)
#               GT     x0y0 to x3y1  (8 total)
#               DCM    x0y0 to x3y1  (8 total)
#               MULT18X18 x0y0 to x7y13 (88 total)
#
# In GRID Coords (X,Y) bottom left is X3Y4, top right is X138Y227
#       X value cycles through SLICE-TBUF-SPECIAL with increasing X
#               -X3Y4 is a SLICE, so is X138Y227
#               -SPECIAL is RAMB/MULT or GT or DCM
#               -SPECIAL only exists at X=8,26,44,62,80,98,116,134
#                  RAMB/MULT exist at all these X locations
#                  GTs at X=26,62,80,116  .and.  Y=19,227
#                  DCMs at X=8,44,98,134  .and.  Y=3,231
#       Y values are continuous for Slices, except for PPC holes
#               -TBUFs are only even in Y (2 per CLB)
#               -RAMBs are spaced 16 apart in Y,
#                  starting at Y=35 on GT columns, Y=19 otherwise
#                  (GTs effectively take up a RAMB spot)
#               -MULTs are 1 lower in Y than RAMB (at Y_RAMB - 1)
# PPC "Dead Zone" holes:
#    No TBUFS from x14y40 to x58y71   .and.   x62y40 to x76y71
#           GRID:  X23Y88 to X47Y143  .and.  X95Y88 to X119Y143
#
###################
# JTAG AREA GROUP
###################
INST "$1I162/$2I4683" AREA_GROUP = AG_JTAG; #Def AreaGrp
INST "$1I162/$2I4683" LOC = SLICE_X8Y46:SLICE_X35Y65;#LocSLx8y48x55y63
AREA_GROUP "AG_JTAG" COMPRESSION = 1;
#
#################
# IN_UNIT0-3 RLOC
#################
# #0  AF_CLB_5x31:
SET "$1I162/$1I4142/$1I4152/af_clb_5x31rpm/mem/distmem/dist_mem/DPRAM" RLO
C_ORIGIN = X102Y204; #mAF_CLB sli_x66y100-->sli_x68y102 RLOC=X0Y0 X105Y208
SET "$1I162/$1I4142/$1I4152/af_clb_5x31rpm/control/rd_blk" RLOC_ORIGIN = X
108Y216;     #rAF_CLB_5x31 RLOC = X6Y12 X111Y220
SET "$1I162/$1I4142/$1I4152/af_clb_5x31rpm/control/wr_blk" RLOC_ORIGIN = X
102Y216;     #wAF_CLB_5x31 RLOC = X0Y12 X105Y220
INST "$1I162/$1I4142/$1I4328" RLOC_ORIGIN = X105Y208; #GT9, SLI_X68Y102
INST "$1I162/$1I4142/$1I3863" LOC = GT_X3Y1; #GT7, SLI_X44Y102
# #1  AF_CLB_5x31:
SET "$1I162/$3I4142/$1I4152/af_clb_5x31rpm/mem/distmem/dist_mem/DPRAM" RLO
C_ORIGIN = X66Y204; #mAF_CLB
SET "$1I162/$3I4142/$1I4152/af_clb_5x31rpm/control/rd_blk" RLOC_ORIGIN = X
72Y216;     #rAF_CLB_5x31
SET "$1I162/$3I4142/$1I4152/af_clb_5x31rpm/control/wr_blk" RLOC_ORIGIN = X
66Y216;     #wAF_CLB_5x31 RLOC = X0Y12 X105Y220
INST "$1I162/$3I4142/$1I4328" RLOC_ORIGIN = X69Y208; #GT7, SLI_X44Y102
INST "$1I162/$3I4142/$1I3863" LOC = GT_X2Y1; #GT7, SLI_X44Y102
# #2  AF_CLB_5x31:
SET "$1I162/$3I4243/$1I4152/af_clb_5x31rpm/mem/distmem/dist_mem/DPRAM" RLO
C_ORIGIN = X48Y204; #mAF_CLB
SET "$1I162/$3I4243/$1I4152/af_clb_5x31rpm/control/rd_blk" RLOC_ORIGIN = X
54Y216;     #rAF_CLB_5x31
SET "$1I162/$3I4243/$1I4152/af_clb_5x31rpm/control/wr_blk" RLOC_ORIGIN = X
48Y216;     #wAF_CLB_5x31 RLOC = X0Y12 X105Y220
INST "$1I162/$3I4243/$1I4328" RLOC_ORIGIN = X51Y208; #GT7, SLI_X32Y102
INST "$1I162/$3I4243/$1I3863" LOC = GT_X1Y1; #GT7, SLI_X44Y102
# #3  AF_CLB_5x31:
SET "$1I162/$3I4274/$1I4152/af_clb_5x31rpm/mem/distmem/dist_mem/DPRAM" RLO
C_ORIGIN = X12Y204; #mAF_CLB
SET "$1I162/$3I4274/$1I4152/af_clb_5x31rpm/control/rd_blk" RLOC_ORIGIN = X
18Y216;     #rAF_CLB_5x31
SET "$1I162/$3I4274/$1I4152/af_clb_5x31rpm/control/wr_blk" RLOC_ORIGIN = X
12Y216;     #wAF_CLB_5x31 RLOC = X0Y12 X105Y220
INST "$1I162/$3I4274/$1I4328" RLOC_ORIGIN = X15Y208; #GT7, SLI_X8Y102
INST "$1I162/$3I4274/$1I3863" LOC = GT_X0Y1; #GT7, SLI_X44Y102
#
#################
# IN_UNIT4-7 RLOC
#################
# #4  AF_CLB_5x31:
SET "$1I162/$5I4382/$1I4152/af_clb_5x31rpm/mem/distmem/dist_mem/DPRAM" RLO
C_ORIGIN = X12Y0; #mAF_CLB sli_x6y0-->sli_x8y0 RLOC = X0Y0 X12Y0
SET "$1I162/$5I4382/$1I4152/af_clb_5x31rpm/control/rd_blk" RLOC_ORIGIN = X
18Y8;     #rAF_CLB_5x31 RLOC = X6Y12 X18Y12
SET "$1I162/$5I4382/$1I4152/af_clb_5x31rpm/control/wr_blk" RLOC_ORIGIN = X
12Y12;     #wAF_CLB_5x31 RLOC = X0Y12 X12Y12
INST "$1I162/$5I4382/$1I3863" RLOC_ORIGIN = X15Y4; #GT21, SLI_X8Y0
# #5  AF_CLB_5x31:
SET "$1I162/$6I4412/$1I4152/af_clb_5x31rpm/mem/distmem/dist_mem/DPRAM" RLO
C_ORIGIN = X48Y0; #mAF_CLB
SET "$1I162/$6I4412/$1I4152/af_clb_5x31rpm/control/rd_blk" RLOC_ORIGIN = X
54Y8;     #rAF_CLB_5x31 RLOC = X6Y12 X54Y12
SET "$1I162/$6I4412/$1I4152/af_clb_5x31rpm/control/wr_blk" RLOC_ORIGIN = X
48Y12;     #wAF_CLB_5x31 RLOC = X0Y12 X48Y12
INST "$1I162/$6I4412/$1I3863" RLOC_ORIGIN = X51Y4; #GT19, SLI_X32Y0
# #6  AF_CLB_5x31:
SET "$1I162/$6I4415/$1I4152/af_clb_5x31rpm/mem/distmem/dist_mem/DPRAM" RLO
C_ORIGIN = X66Y0; #mAF_CLB
SET "$1I162/$6I4415/$1I4152/af_clb_5x31rpm/control/rd_blk" RLOC_ORIGIN = X
72Y8;     #rAF_CLB_5x31 RLOC = X6Y12 X72Y12
SET "$1I162/$6I4415/$1I4152/af_clb_5x31rpm/control/wr_blk" RLOC_ORIGIN = X
66Y12;     #wAF_CLB_5x31 RLOC = X0Y12 X66Y12
INST "$1I162/$6I4415/$1I3863" RLOC_ORIGIN = X69Y4; #GT18, SLI_X44Y0
# #7  AF_CLB_5x31:
SET "$1I162/$6I4446/$1I4152/af_clb_5x31rpm/mem/distmem/dist_mem/DPRAM" RLO
C_ORIGIN = X102Y0; #mAF_CLB
SET "$1I162/$6I4446/$1I4152/af_clb_5x31rpm/control/rd_blk" RLOC_ORIGIN = X
108Y8;     #rAF_CLB_5x31 RLOC = X6Y12 X108Y12
SET "$1I162/$6I4446/$1I4152/af_clb_5x31rpm/control/wr_blk" RLOC_ORIGIN = X
102Y12;     #wAF_CLB_5x31 RLOC = X0Y12 X102Y12
INST "$1I162/$6I4446/$1I3863" RLOC_ORIGIN = X105Y4; #GT16, SLI_X68Y0
```

```
#
#################
# MEM_UNIT RLOC
#################
# --Corner 0--
#  Define Global Location for MU0:
INST "$1I162/$1I4143/$1I4507" RLOC_ORIGIN = X129Y100;#WR_SEL FDCE
INST "$1I162/$1I4143/$1I4488" RLOC_ORIGIN = X129Y100;#SF_BRAM,SL_X84Y48
INST "$1I162/$1I4143" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU1:
INST "$1I162/$3I4417/$1I4507" RLOC_ORIGIN = X129Y132;#WR_SEL FDCE
INST "$1I162/$3I4417/$1I4488" RLOC_ORIGIN = X129Y132;#SF_BRAM,SL_X84Y64
INST "$1I162/$3I4417" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU2:
INST "$1I162/$3I4392/$1I4507" RLOC_ORIGIN = X129Y164;#WR_SEL FDCE
INST "$1I162/$3I4392/$1I4488" RLOC_ORIGIN = X129Y164;#SF_BRAM,SL_X84Y80
INST "$1I162/$3I4392" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU3:
INST "$1I162/$3I4329/$1I4507" RLOC_ORIGIN = X129Y196;#WR_SEL FDCE
INST "$1I162/$3I4329/$1I4488" RLOC_ORIGIN = X129Y196;#SF_BRAM,SL_X84Y96
INST "$1I162/$3I4329" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU4:
INST "$1I162/$3I4526/$1I4507" RLOC_ORIGIN = X111Y148;#WR_SEL FDCE
INST "$1I162/$3I4526/$1I4488" RLOC_ORIGIN = X111Y148;#SF_BRAM,SL_X72Y72
INST "$1I162/$3I4526" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU5:
INST "$1I162/$3I4496/$1I4507" RLOC_ORIGIN = X111Y180;#WR_SEL FDCE
INST "$1I162/$3I4496/$1I4488" RLOC_ORIGIN = X111Y180;#SF_BRAM,SL_X72Y88
INST "$1I162/$3I4496" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU6:
INST "$1I162/$3I4365/$1I4507" RLOC_ORIGIN = X93Y196; #WR_SEL FDCE
INST "$1I162/$3I4365/$1I4488" RLOC_ORIGIN = X93Y196; #SF_BRAM,SL_X60Y96
INST "$1I162/$3I4365" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU7:
INST "$1I162/$3I4551/$1I4507" RLOC_ORIGIN = X93Y164; #WR_SEL FDCE
INST "$1I162/$3I4551/$1I4488" RLOC_ORIGIN = X93Y164; #SF_BRAM,SL_X60Y80
INST "$1I162/$3I4551" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for *MU8*:
INST "$1I162/$4I4630/$1I4507" RLOC_ORIGIN = X75Y148;#WR_SEL FDCE
INST "$1I162/$4I4630/$1I4488" RLOC_ORIGIN = X75Y148;#SF_BRAM,SL_X48Y72
INST "$1I162/$4I4630" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU9:
INST "$1I162/$3I4489/$1I4507" RLOC_ORIGIN = X75Y132; #WR_SEL FDCE
INST "$1I162/$3I4489/$1I4488" RLOC_ORIGIN = X75Y132; #SF_BRAM,SL_X48Y64
INST "$1I162/$3I4489" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU10:
INST "$1I162/$3I4464/$1I4507" RLOC_ORIGIN = X75Y180; #WR_SEL FDCE
INST "$1I162/$3I4464/$1I4488" RLOC_ORIGIN = X75Y180; #SF_BRAM,SL_X48Y88
INST "$1I162/$3I4464" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
# --Corner 1--
#  Define Global Location for MU16:
INST "$1I162/$4I4329/$1I4507" RLOC_ORIGIN = X3Y132; #WR_SEL FDCE
INST "$1I162/$4I4329/$1I4488" RLOC_ORIGIN = X3Y132; #SF_BRAM,SLI_X0Y64
INST "$1I162/$4I4329" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU17:
INST "$1I162/$4I4399/$1I4507" RLOC_ORIGIN = X3Y164; #WR_SEL FDCE
INST "$1I162/$4I4399/$1I4488" RLOC_ORIGIN = X3Y164; #SF_BRAM,SLI_X0Y80
INST "$1I162/$4I4399" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU18:
INST "$1I162/$4I4365/$1I4507" RLOC_ORIGIN = X3Y196; #WR_SEL FDCE
INST "$1I162/$4I4365/$1I4488" RLOC_ORIGIN = X3Y196; #SF_BRAM,SLI_X0Y96
INST "$1I162/$4I4365" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU19:
INST "$1I162/$4I4426/$1I4507" RLOC_ORIGIN = X21Y148; #WR_SEL FDCE
INST "$1I162/$4I4426/$1I4488" RLOC_ORIGIN = X21Y148; #SF_BRAM,SL_X12Y72
INST "$1I162/$4I4426" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU20:
INST "$1I162/$4I4475/$1I4507" RLOC_ORIGIN = X21Y180; #WR_SEL FDCE
INST "$1I162/$4I4475/$1I4488" RLOC_ORIGIN = X21Y180; #SF_BRAM,SL_X12Y88
INST "$1I162/$4I4475" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU21:
INST "$1I162/$4I4476/$1I4507" RLOC_ORIGIN = X39Y196; #WR_SEL FDCE
INST "$1I162/$4I4476/$1I4488" RLOC_ORIGIN = X39Y196; #SF_BRAM,SL_X24Y96
INST "$1I162/$4I4476" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU22:
INST "$1I162/$4I4501/$1I4507" RLOC_ORIGIN = X39Y164; #WR_SEL FDCE
INST "$1I162/$4I4501/$1I4488" RLOC_ORIGIN = X39Y164; #SF_BRAM,SL_X24Y80
INST "$1I162/$4I4501" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU23:
INST "$1I162/$4I4526/$1I4507" RLOC_ORIGIN = X57Y100; #WR_SEL FDCE
INST "$1I162/$4I4526/$1I4488" RLOC_ORIGIN = X57Y100; #SF_BRAM,SL_X36Y48
INST "$1I162/$4I4526" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for *MU24*:
INST "$1I162/$4I4657/$1I4507" RLOC_ORIGIN = X39Y132;#WR_SEL FDCE
INST "$1I162/$4I4657/$1I4488" RLOC_ORIGIN = X39Y132;#SF_BRAM,SL_X24Y64
INST "$1I162/$4I4657" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU25:
INST "$1I162/$4I4560/$1I4507" RLOC_ORIGIN = X57Y148; #WR_SEL FDCE
INST "$1I162/$4I4560/$1I4488" RLOC_ORIGIN = X57Y148; #SF_BRAM,SL_X36Y72
INST "$1I162/$4I4560" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
#  Define Global Location for MU26:
INST "$1I162/$4I4611/$1I4507" RLOC_ORIGIN = X57Y180; #WR_SEL FDCE
INST "$1I162/$4I4611/$1I4488" RLOC_ORIGIN = X57Y180; #SF_BRAM,SL_X36Y88
INST "$1I162/$4I4611" AREA_GROUP = AG_MEMTBUF_UP; #Def AreaGrp
AREA_GROUP "AG_MEMTBUF_UP" RANGE = TBUF_X0Y62:TBUF_X90Y106; #LocTBUF
# --Corner 2--
#  Define Global Location for MU32:
INST "$1I162/$5I4143/$1I4507" RLOC_ORIGIN = X3Y100; #WR_SEL FDCE
INST "$1I162/$5I4143/$1I4488" RLOC_ORIGIN = X3Y100; #SF_BRAM,SLI_X0Y48
INST "$1I162/$5I4143" LOC = TBUF_X0Y5:TBUF_X0Y49;
INST "$1I162/$5I4143" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU33:
INST "$1I162/$6I4143/$1I4507" RLOC_ORIGIN = X3Y68; #WR_SEL FDCE
INST "$1I162/$6I4143/$1I4488" RLOC_ORIGIN = X3Y68; #SF_BRAM,SLI_X0Y32
INST "$1I162/$6I4143" LOC = TBUF_X2Y5:TBUF_X2Y49;
INST "$1I162/$6I4143" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU34:
INST "$1I162/$6I4479/$1I4507" RLOC_ORIGIN = X3Y36; #WR_SEL FDCE
INST "$1I162/$6I4479/$1I4488" RLOC_ORIGIN = X3Y36; #SF_BRAM,SLI_X0Y16
INST "$1I162/$6I4479" LOC = TBUF_X4Y5:TBUF_X4Y49;
INST "$1I162/$6I4479" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU35:
INST "$1I162/$6I4504/$1I4507" RLOC_ORIGIN = X3Y4; #WR_SEL FDCE
INST "$1I162/$6I4504/$1I4488" RLOC_ORIGIN = X3Y4; #SF_BRAM,SLI_X0Y0
INST "$1I162/$6I4504" LOC = TBUF_X6Y5:TBUF_X6Y49;
INST "$1I162/$6I4504" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU36:
INST "$1I162/$6I4529/$1I4507" RLOC_ORIGIN = X21Y52; #WR_SEL FDCE
INST "$1I162/$6I4529/$1I4488" RLOC_ORIGIN = X21Y52; #SF_BRAM,SL_X12Y24
INST "$1I162/$6I4529" LOC = TBUF_X8Y5:TBUF_X8Y49;
INST "$1I162/$6I4529" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU37:
INST "$1I162/$6I4575/$1I4507" RLOC_ORIGIN = X21Y20; #WR_SEL FDCE
INST "$1I162/$6I4575/$1I4488" RLOC_ORIGIN = X21Y20; #SF_BRAM,SLI_X12Y8
INST "$1I162/$6I4575" LOC = TBUF_X10Y5:TBUF_X10Y49;
INST "$1I162/$6I4575" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU38:
INST "$1I162/$6I4598/$1I4507" RLOC_ORIGIN = X39Y4; #WR_SEL FDCE
INST "$1I162/$6I4598/$1I4488" RLOC_ORIGIN = X39Y4; #SF_BRAM,SLI_X24Y0
INST "$1I162/$6I4598" LOC = TBUF_X30Y5:TBUF_X30Y49;
INST "$1I162/$6I4598" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU39:
INST "$1I162/$6I4614/$1I4507" RLOC_ORIGIN = X39Y36; #WR_SEL FDCE
INST "$1I162/$6I4614/$1I4488" RLOC_ORIGIN = X39Y36; #SF_BRAM,SL_X24Y16
INST "$1I162/$6I4614" LOC = TBUF_X12Y5:TBUF_X12Y49;
INST "$1I162/$6I4614" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for *MU40*:
INST "$1I162/$7I5013/$1I4507" RLOC_ORIGIN = X39Y52; #WR_SEL FDCE
```

```
INST "$1I162/$7I5013/$1I4488" RLOC_ORIGIN = X39Y52; #SF_BRAM,SL_X24Y24
INST "$1I162/$7I5013" LOC = TBUF_X32Y5:TBUF_X32Y49;
INST "$1I162/$7I5013" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU41:
INST "$1I162/$6I4641/$1I4507" RLOC_ORIGIN = X57Y68; #WR_SEL FDCE
INST "$1I162/$6I4641/$1I4488" RLOC_ORIGIN = X57Y68; #SF_BRAM,SL_X36Y32
INST "$1I162/$6I4641" LOC = TBUF_X34Y5:TBUF_X34Y49;
INST "$1I162/$6I4641" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU42:
INST "$1I162/$6I4686/$1I4507" RLOC_ORIGIN = X57Y20; #WR_SEL FDCE
INST "$1I162/$6I4686/$1I4488" RLOC_ORIGIN = X57Y20; #SF_BRAM,SLI_X36Y8
INST "$1I162/$6I4686" LOC = TBUF_X36Y5:TBUF_X36Y49;
INST "$1I162/$6I4686" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#AREA_GROUP "AG_MEMTBUF_C2" RANGE = TBUF_X0Y0:TBUF_X44Y31; #LocTBUF
# --Corner 3--
#  Define Global Location for MU48:
INST "$1I162/$7I4614/$1I4507" RLOC_ORIGIN = X129Y68; #WR_SEL FDCE
INST "$1I162/$7I4614/$1I4488" RLOC_ORIGIN = X129Y68; #SF_BRAM,SL_X84Y32
INST "$1I162/$7I4614" LOC = TBUF_X58Y5:TBUF_X58Y49;
INST "$1I162/$7I4614" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU49:
INST "$1I162/$7I4615/$1I4507" RLOC_ORIGIN = X129Y36; #WR_SEL FDCE
INST "$1I162/$7I4615/$1I4488" RLOC_ORIGIN = X129Y36; #SF_BRAM,SL_X84Y16
INST "$1I162/$7I4615" LOC = TBUF_X56Y5:TBUF_X56Y49;
INST "$1I162/$7I4615" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU50:
INST "$1I162/$7I4616/$1I4507" RLOC_ORIGIN = X129Y4; #WR_SEL FDCE
INST "$1I162/$7I4616/$1I4488" RLOC_ORIGIN = X129Y4; #SF_BRAM,SLI_X84Y0
INST "$1I162/$7I4616" LOC = TBUF_X54Y5:TBUF_X54Y49;
INST "$1I162/$7I4616" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU51:
INST "$1I162/$7I4863/$1I4507" RLOC_ORIGIN = X111Y52; #WR_SEL FDCE
INST "$1I162/$7I4863/$1I4488" RLOC_ORIGIN = X111Y52; #SF_BRAM,SL_X72Y24
INST "$1I162/$7I4863" LOC = TBUF_X52Y5:TBUF_X52Y49;
INST "$1I162/$7I4863" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU52:
INST "$1I162/$7I4838/$1I4507" RLOC_ORIGIN = X111Y20; #WR_SEL FDCE
INST "$1I162/$7I4838/$1I4488" RLOC_ORIGIN = X111Y20; #SF_BRAM,SLI_X72Y8
INST "$1I162/$7I4838" LOC = TBUF_X50Y5:TBUF_X50Y49;
INST "$1I162/$7I4838" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU53:
INST "$1I162/$7I4762/$1I4507" RLOC_ORIGIN = X93Y4; #WR_SEL FDCE
INST "$1I162/$7I4762/$1I4488" RLOC_ORIGIN = X93Y4; #SF_BRAM,SLI_X60Y0
INST "$1I162/$7I4762" LOC = TBUF_X46Y5:TBUF_X46Y49;
INST "$1I162/$7I4762" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU54:
INST "$1I162/$7I4731/$1I4507" RLOC_ORIGIN = X93Y36; #WR_SEL FDCE
INST "$1I162/$7I4731/$1I4488" RLOC_ORIGIN = X93Y36; #SF_BRAM,SLI_X60Y16
INST "$1I162/$7I4731" LOC = TBUF_X48Y5:TBUF_X48Y49;
INST "$1I162/$7I4731" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU55:
INST "$1I162/$7I4712/$1I4507" RLOC_ORIGIN = X75Y100; #WR_SEL FDCE
INST "$1I162/$7I4712/$1I4488" RLOC_ORIGIN = X75Y100; #SF_BRAM,SL_X48Y48
INST "$1I162/$7I4712" LOC = TBUF_X44Y5:TBUF_X44Y49;
INST "$1I162/$7I4712" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for *MU56*:
INST "$1I162/$7I4974/$1I4507" RLOC_ORIGIN = X75Y68; #WR_SEL FDCE
INST "$1I162/$7I4974/$1I4488" RLOC_ORIGIN = X75Y68; #SF_BRAM,SL_X48Y32
INST "$1I162/$7I4974" LOC = TBUF_X42Y5:TBUF_X42Y49;
INST "$1I162/$7I4974" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU57:
INST "$1I162/$7I4632/$1I4507" RLOC_ORIGIN = X75Y52; #WR_SEL FDCE
INST "$1I162/$7I4632/$1I4488" RLOC_ORIGIN = X75Y52; #SF_BRAM,SLI_X48Y24
INST "$1I162/$7I4632" LOC = TBUF_X40Y5:TBUF_X40Y49;
INST "$1I162/$7I4632" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
#  Define Global Location for MU58:
INST "$1I162/$7I4651/$1I4507" RLOC_ORIGIN = X75Y20; #WR_SEL FDCE
INST "$1I162/$7I4651/$1I4488" RLOC_ORIGIN = X75Y20; #SF_BRAM,SLI_X48Y8
INST "$1I162/$7I4651" LOC = TBUF_X38Y5:TBUF_X38Y49;
INST "$1I162/$7I4651" AREA_GROUP = AG_MEMTBUF_DN; #Def AreaGrp
# AREA_GROUP "AG_MEMTBUF_DN" RANGE = TBUF_X0Y0:TBUF_X90Y55; #LocTBUF
# AREA_GROUP "AG_MEMTBUF_C3" RANGE = TBUF_X46Y0:TBUF_X90Y31; #LocTBUF
#
#################
# MEM_CTRL0 RLOC
#################
INST "$1I162/$1I3963" AREA_GROUP = AG_MEM_CTRL0; #Def AreaGrp
INST "$1I162/$1I3963" LOC = SLICE_X8Y66:SLICE_X35Y71;#LocSL x8y64
#AREA_GROUP "AG_MEM_CTRL0" RANGE = SLICE_X8Y66:SLICE_X35Y71;#LocSL x8y64
AREA_GROUP "AG_MEM_CTRL0" COMPRESSION = 1;
#
#################
# MEM_CTRL1 RLOC
#################
INST "$1I162/$5I3963" AREA_GROUP = AG_MEM_CTRL1; #Def AreaGrp
INST "$1I162/$5I3963" LOC = SLICE_X8Y40:SLICE_X35Y45;#LocSLx35y47
#AREA_GROUP "AG_MEM_CTRL1" RANGE = SLICE_X8Y40:SLICE_X35Y45;#LocSLx35y47
AREA_GROUP "AG_MEM_CTRL1" COMPRESSION = 1;
#
#################
# READ_CTRL0 RLOC
#################
INST "$1I162/$1I4156/$1I3760/$1I96" RLOC = X0Y0; #OUT0 FD18CE_RPM x0y12
INST "$1I162/$1I4156/$1I3760/$1I99" RLOC = X3Y0; #OUT1 FD18CE_RPM x0y0
INST "$1I162/$1I4156/$1I3780/$1I96" RLOC = X0Y0; #DO0 FD18CE_RPM
INST "$1I162/$1I4156/$1I3780/$1I99" RLOC = X3Y0; #DO1 FD18CE_RPM
INST "$1I162/$1I4156/$1I3760/$1I96" RLOC_ORIGIN = X69Y148; # X69Y132
INST "$1I162/$1I4156/$1I3780/$1I96" RLOC_ORIGIN = X123Y148; #
#
INST "$1I162/$1I4156" AREA_GROUP = AG_RD_CTRL0; #Def AreaGrp
AREA_GROUP "AG_RD_CTRL0" RANGE = SLICE_X56Y56:SLICE_X83Y79; #Loc slice
#AREA_GROUP "AG_RD_CTRL0" RANGE = SLICE_X56Y56:SLICE_X83Y71; #Loc slice
AREA_GROUP "AG_RD_CTRL0" COMPRESSION = 1;
#
#################
# READ_CTRL1 RLOC
#################
INST "$1I162/$5I4156/$1I3760/$1I96" RLOC = X0Y0; #OUT0 FD18CE_RPM x0y12
INST "$1I162/$5I4156/$1I3760/$1I99" RLOC = X3Y0; #OUT1 FD18CE_RPM x0y0
INST "$1I162/$5I4156/$1I3780/$1I96" RLOC = X0Y0; #DO0 FD18CE_RPM
INST "$1I162/$5I4156/$1I3780/$1I99" RLOC = X3Y0; #DO1 FD18CE_RPM
INST "$1I162/$5I4156/$1I3760/$1I96" RLOC_ORIGIN = X69Y68; # X69Y132
INST "$1I162/$5I4156/$1I3780/$1I96" RLOC_ORIGIN = X123Y68; #
#
INST "$1I162/$5I4156" AREA_GROUP = AG_RD_CTRL1; #Def AreaGrp
AREA_GROUP "AG_RD_CTRL1" RANGE = SLICE_X56Y32:SLICE_X83Y55; #Loc slice
#AREA_GROUP "AG_RD_CTRL1" RANGE = SLICE_X56Y40:SLICE_X83Y55; #Loc slice
AREA_GROUP "AG_RD_CTRL1" COMPRESSION = 1;
#PINLOCK_END



DDU Control FPGA UCF file
-------------------------
# use with "D785/DDU5ctrl: DDU5CTRL" Logic schematic
#PINLOCK_BEGIN
#Sun Sep 27 12:38:16 2005
#INPUTS
# Control signals in:
#   from switches
NET "$1I135/~ISYSRST" LOC = "D6";   #~sysreset
NET "$1I135/~ARSTIN"  LOC = "C9";   #rst_sw
NET "$1I135/MODEIN0"  LOC = "AF24"; #to Switch1
NET "$1I135/MODEIN1"  LOC = "AE24";
NET "$1I135/MODEIN2"  LOC = "AD23";
NET "$1I135/MODEIN3"  LOC = "AC24";
NET "$1I135/MODEIN4"  LOC = "AE26";
NET "$1I135/MODEIN5"  LOC = "AF25";
```

```
NET "$1I135/MODEIN6"  LOC = "AD25";
NET "$1I135/MODEIN7"  LOC = "AD26"; #to Switch8
#
# Control Input from other FPGAs
NET "$1I135/~RDYIN0"  LOC = "AE1"; #dll_ready0
NET "$1I135/~RDYIN1"  LOC = "AD15";
NET "$1I135/~RDYIN2"  LOC = "C4";
NET "$1I135/SCLKIN"   LOC = "D14"; #sclk
NET "$1I135/~SENIN"   LOC = "H13"; #sen5
NET "$1I135/SDIN"     LOC = "H12"; #si
#
# TTC/CCB Signals in:
NET "$1I135/CLKIN40"  LOC = "C13"; #ccbclk, 3.3V
NET "$1I135/CKFB_IN"  LOC = "AE13";#clk_fb
NET "$1I135/FCKIN_P"  LOC = "B14"; #clk120p
NET "$1I135/FCKIN_N"  LOC = "C14"; #clk120n
#NET "$1I135/CK625IN"  LOC = "D13"; #clk625n, 3.3V
NET "$1I135/CK625IN"  LOC = "E13"; #clk625p, 3.3V
#
# TTC Command bus from DCC:
NET "$1I135/RAWL1A"  LOC = "C12"; #L1A
NET "$1I135/BUS0"    LOC = "E11";  #bus0, really IS EvCntRst signal
NET "$1I135/BUS1"    LOC = "G12";  #bus1, really IS BXR signal
NET "$1I135/BUS2"    LOC = "G11";  #bus2
NET "$1I135/BUS3"    LOC = "D10";
NET "$1I135/BUS4"    LOC = "E10";
NET "$1I135/BUS5"    LOC = "F11";
NET "$1I135/BUS6"    LOC = "F10";
NET "$1I135/BUS7"    LOC = "H11";
NET "$1I135/BUS8"    LOC = "G10";  # really the CCB_CMD_STROBE signal
NET "$1I135/~STOP_DATA"   LOC = "F13";  # From DCC
NET "$1I135/~LINK_READY"  LOC = "G13";  # From DCC
NET "$1I135/SBDATA"       LOC = "D12";  # Serial TTC data, Not Used
NET "$1I135/TDSTRB"       LOC = "E12";  # Parallel TTC data, Not Used
NET "$1I135/TDOUT0"       LOC = "F12";  # From TTC?  Not used
NET "$1I135/TDOUT1"       LOC = "D11";  # From TTC?  Not used
#
#FIFO Status Signals in:
#   from output Alt.FIFO
NET "$1I135/OFULL"        LOC = "L25";
NET "$1I135/~OPAF"        LOC = "K26";
NET "$1I135/~OHALF"       LOC = "C18";
NET "$1I135/~OPAE"        LOC = "D17";
NET "$1I135/OEMPTY"       LOC = "J25";
NET "$1I135/OFIBER_OK"    LOC = "E14";       #fok
#   from input FPGAs
NET "$1I135/RDCTRL0STAT0" LOC = "W1";   #rdstat0
NET "$1I135/RDCTRL0STAT1" LOC = "V7";
NET "$1I135/RDCTRL0STAT2" LOC = "V6";
NET "$1I135/RDCTRL0STAT3" LOC = "V5";
NET "$1I135/RDCTRL1STAT0" LOC = "V4";   #rdstat4
NET "$1I135/RDCTRL1STAT1" LOC = "V3";
NET "$1I135/RDCTRL1STAT2" LOC = "V2";
NET "$1I135/RDCTRL1STAT3" LOC = "U6";
NET "$1I135/RDCTRL2STAT0" LOC = "AC11"; #rdstat8
NET "$1I135/RDCTRL2STAT1" LOC = "AB11";
NET "$1I135/RDCTRL2STAT2" LOC = "Y12";
NET "$1I135/RDCTRL2STAT3" LOC = "Y11";
NET "$1I135/RDCTRL3STAT0" LOC = "AC10"; #rdstat12
NET "$1I135/RDCTRL3STAT1" LOC = "AB10";
NET "$1I135/RDCTRL3STAT2" LOC = "AA11";
NET "$1I135/RDCTRL3STAT3" LOC = "AA10"; #rdstat15
#   from input FIFOs
NET "$1I135/MTIN0"  LOC = "E3"; #ef0in, FIFO-B-Empty signal
NET "$1I135/MTIN1"  LOC = "H4";
NET "$1I135/MTIN2"  LOC = "H5";
NET "$1I135/MTIN3"  LOC = "M5";
NET "$1I135/EFAIN0" LOC = "M7"; #efa0in, FIFO-A-Empty signal
NET "$1I135/EFAIN1" LOC = "N7"; #efa1in
NET "$1I135/EFAIN2" LOC = "T6"; #efa2in
NET "$1I135/EFAIN3" LOC = "B3"; #efa3in
NET "$1I135/HFIN0" LOC = "E4"; #hf0in, really FIFO-B-Full signal
NET "$1I135/HFIN1" LOC = "G4"; #hf1in
NET "$1I135/HFIN2" LOC = "H6"; #hf2in
NET "$1I135/HFIN3" LOC = "L5"; #hf3in
NET "$1I135/~PAFIN0" LOC = "T5";#paf0in, these 4 aren't really needed...
NET "$1I135/~PAFIN1" LOC = "T4";
NET "$1I135/~PAFIN2" LOC = "R4";
NET "$1I135/~PAFIN3" LOC = "P4";
NET "$1I135/FFIN0"  LOC = "T3"; #ff0in, these 4 aren't really needed...
NET "$1I135/FFIN1"  LOC = "R3";
NET "$1I135/FFIN2"  LOC = "P3";
NET "$1I135/FFIN3"  LOC = "N4";
NET "$1I135/~INFIFO_USE0" LOC = "AC3"; #fifo_active0
NET "$1I135/~INFIFO_USE1" LOC = "AF2";
NET "$1I135/~INFIFO_USE2" LOC = "AB15";
NET "$1I135/~INFIFO_USE3" LOC = "AC15";
# send Input FIFO READY out to InFPGAs (for bit Test5In):
#NET "$1I135/~OFRDY0"  LOC = "Y10"; #spare10 DDUtest "DoFake0"
#NET "$1I135/~OFRDY1"  LOC = "AD9"; #spare11 DDUtest "DoFake1"
#NET "$1I135/~OFRDY2"  LOC = "P7"; #spare22 DDUtest "DoFake2"
#NET "$1I135/~OFRDY3"  LOC = "P8"; #spare23 DDUtest "DoFake3"
#   Data signals in
NET "$1I135/I0"      LOC = "M6"; #data0  *~all Data In pins changed!
NET "$1I135/I1"      LOC = "L6";
NET "$1I135/I2"      LOC = "R6";
NET "$1I135/I3"      LOC = "R5";
NET "$1I135/I4"      LOC = "P5";
NET "$1I135/I5"      LOC = "P6";
NET "$1I135/I6"      LOC = "N3";
NET "$1I135/I7"      LOC = "M4";
NET "$1I135/I8"      LOC = "M3";
NET "$1I135/I9"      LOC = "L4";
NET "$1I135/I10"     LOC = "L3";
NET "$1I135/I11"     LOC = "K4";
NET "$1I135/I12"     LOC = "K3";
NET "$1I135/I13"     LOC = "J3";
NET "$1I135/I14"     LOC = "T2";
NET "$1I135/I15"     LOC = "T1";
NET "$1I135/I16"     LOC = "R2";
NET "$1I135/I17"     LOC = "R1";
NET "$1I135/I18"     LOC = "P2";
NET "$1I135/I19"     LOC = "N2";
NET "$1I135/I20"     LOC = "M1";
NET "$1I135/I21"     LOC = "M2";
NET "$1I135/I22"     LOC = "L1";
NET "$1I135/I23"     LOC = "L2";
NET "$1I135/I24"     LOC = "K1";
NET "$1I135/I25"     LOC = "J1";
NET "$1I135/I26"     LOC = "J2";
NET "$1I135/I27"     LOC = "H1";
NET "$1I135/I28"     LOC = "H2";
NET "$1I135/I29"     LOC = "G1";
NET "$1I135/I30"     LOC = "F1";
NET "$1I135/I31"     LOC = "G3";
NET "$1I135/I32"     LOC = "E1";
NET "$1I135/I33"     LOC = "E2";
NET "$1I135/I34"     LOC = "D1";
NET "$1I135/I35"     LOC = "D2";
#
#S-LINK Command & Control Signals:
NET "$1I135/ILSF0"   LOC = "R26"; #lsf0
NET "$1I135/ILSF1"   LOC = "N24";
NET "$1I135/ILSF2"   LOC = "R25";
NET "$1I135/ILSF3"   LOC = "N23";
NET "$1I135/ILRL0"   LOC = "L24"; #lrl0
NET "$1I135/ILRL1"   LOC = "L23";
NET "$1I135/ILRL2"   LOC = "M24";
```

```
NET "$1I135/ILRL3"   LOC = "M23";
NET "$1I135/~ILDOWN" LOC = "K24"; #ldown
NET "$1I135/~ILFF"   LOC = "K23"; #lff
NET "$1I135/USF0"    LOC = "P24"; #usf0
NET "$1I135/USF1"    LOC = "P23";
NET "$1I135/USF2"    LOC = "R24";
NET "$1I135/USF3"    LOC = "R23";
NET "$1I135/~OUWEN"  LOC = "J24"; #uwen
NET "$1I135/~OUCTRL" LOC = "H23"; #uctrl
NET "$1I135/~UTEST"  LOC = "H24"; #utest
NET "$1I135/~URESET" LOC = "J23"; #ureset
#
#OUTPUTS
#Error Signals out:
#   to LEDs (use parallel-load-data pins).
# Set in Schematic now: LED1-8.   cdata[7:0]
#
# TestPoints:
NET "$1I135/TP_42"   LOC = "AE14"; #brefclk_n_free
NET "$1I135/TP_43"   LOC = "AD14"; #brefclk_p_free
NET "$1I135/TP_45"   LOC = "AC8";  #unused1
NET "$1I135/TP_48"   LOC = "A3";   #unused3
NET "$1I135/TP_1"    LOC = "R8";   #unused
NET "$1I135/TP_9"    LOC = "AC6";  #unused
NET "$1I135/TP_11"   LOC = "AC14"; #unused
#
# Rocket IO: Gigabit Transceiver pins
NET "$1I135/RXP0"    LOC = "A16";  #unused rx2+, Rocket 6
NET "$1I135/RXN0"    LOC = "A15";  #unused rx2-
NET "$1I135/TXP0"    LOC = "A17";  #tx2+, Rocket 6
NET "$1I135/TXN0"    LOC = "A18";  #tx2-
NET "$1I135/RXP1"    LOC = "A21";  #rx3+, Rocket 4
NET "$1I135/RXN1"    LOC = "A20";  #rx3-
NET "$1I135/TXP1"    LOC = "A22";  #tx3+
NET "$1I135/TXN1"    LOC = "A23";  #tx3-
NET "$1I135/RXP_GBE" LOC = "AF21"; #rd4+, Rocket 21
NET "$1I135/RXN_GBE" LOC = "AF20"; #rd4-
NET "$1I135/TXP_GBE" LOC = "AF22"; #td4+
NET "$1I135/TXN_GBE" LOC = "AF23"; #td4-
#
# Send G-Bit fiber status signals to Front Panel LEDs
NET "$1I135/FOKOUT"  LOC = "AB9";  # G-Bit Fiber OK
NET "$1I135/DAV"     LOC = "AB8";  # G-Bit DAV (Xmit data to PC)
#
#Control Signals out:
#   input FIFO control
NET "$1I135/OUT_EVCNTRST" LOC = "AB18"; #was called L1ARST
NET "$1I135/~PRST"     LOC = "J7";   #~fifo_prst
NET "$1I135/~MRST"     LOC = "J6";   #~fifo_mrst
NET "$1I135/BC0OUT"    LOC = "AC19";
NET "$1I135/L1AOUT"    LOC = "AB19";
NET "$1I135/~OSYNCRST" LOC = "AD19"; #syncrst
NET "$1I135/~OSOFTRST" LOC = "AE19"; #softrst
NET "$1I135/$1I3030/~FOE0"     LOC = "G5"; #infifo_oe0
NET "$1I135/$1I3030/~FOE1"     LOC = "J4";
NET "$1I135/$1I3030/~FOE2"     LOC = "K5";
NET "$1I135/$1I3030/~FOE3"     LOC = "C1";
NET "$1I135/$1I3030/~RENFIFO0" LOC = "G6"; #infifo_ren0
NET "$1I135/$1I3030/~RENFIFO1" LOC = "H3";
NET "$1I135/$1I3030/~RENFIFO2" LOC = "J5";
NET "$1I135/$1I3030/~RENFIFO3" LOC = "N5";
#
#Data signals out:
#   Output FIFO control
NET "$1I135/CKFB_OUT"  LOC = "AD13";# 78 MHz out from FPGA-DCM
NET "$1I135/OWCLK"     LOC = "M25"; #  "  as above  "
NET "$1I135/IRCLK0"    LOC = "K6";  #  "  as above  " for InFIFO-B 0&1
NET "$1I135/IRCLK1"    LOC = "N6";  #  "  as above  " for InFIFO-B 2&3
NET "$1I135/ORCLK"     LOC = "J26"; # 62.5 MHz out, oscillator freq.
NET "$1I135/FIFO_CLK0" LOC = "Y8"; # 156 MHz out, FIFO-A-RCLK/FIFO-B-WCLK
NET "$1I135/FIFO_CLK1" LOC = "AD4";
NET "$1I135/FIFO_CLK2" LOC = "AF3";
NET "$1I135/FIFO_CLK3" LOC = "AE3";
NET "$1I135/~OWEN"     LOC = "L26"; #owen
NET "$1I135/~OREN"     LOC = "H26"; #oren
#   TX output to DMBs via InCtrl FPGAs
NET "$1I135/KILL0"   LOC = "AD1";
NET "$1I135/KILL1"   LOC = "AC2";
NET "$1I135/KILL2"   LOC = "AC1";
NET "$1I135/KILL3"   LOC = "AB4";
NET "$1I135/KILL4"   LOC = "AB3";
NET "$1I135/KILL5"   LOC = "Y6";
NET "$1I135/KILL6"   LOC = "Y5";
NET "$1I135/KILL7"   LOC = "Y4";
NET "$1I135/KILL8"   LOC = "W14";
NET "$1I135/KILL9"   LOC = "Y14";
NET "$1I135/KILL10"  LOC = "AA14";
NET "$1I135/KILL11"  LOC = "AB14";
NET "$1I135/KILL12"  LOC = "AC13";
NET "$1I135/KILL13"  LOC = "AB13";
NET "$1I135/KILL14"  LOC = "AA13";
NET "$1I135/~TXEN0"  LOC = "Y3";
NET "$1I135/~TXEN1"  LOC = "AA1";
NET "$1I135/~TXEN2"  LOC = "Y1";
NET "$1I135/~TXEN3"  LOC = "W6";
NET "$1I135/~TXEN4"  LOC = "W5";
NET "$1I135/~TXEN5"  LOC = "W4";
NET "$1I135/~TXEN6"  LOC = "W3";
NET "$1I135/~TXEN7"  LOC = "W2";
NET "$1I135/~TXEN8"  LOC = "Y13";
NET "$1I135/~TXEN9"  LOC = "W13";
NET "$1I135/~TXEN10" LOC = "W12";
NET "$1I135/~TXEN11" LOC = "AD12";
NET "$1I135/~TXEN12" LOC = "AC12";
NET "$1I135/~TXEN13" LOC = "AB12";
NET "$1I135/~TXEN14" LOC = "AA12";
NET "$1I135/TXDIN0"  LOC = "AF19";
NET "$1I135/TXDIN1"  LOC = "Y18";
NET "$1I135/TXDIN2"  LOC = "AA18";
NET "$1I135/TXDIN3"  LOC = "AC18";
NET "$1I135/TXDIN4"  LOC = "AD18";
NET "$1I135/TXDIN5"  LOC = "Y17";
NET "$1I135/TXDIN6"  LOC = "W16";
NET "$1I135/TXDIN7"  LOC = "AA17";
NET "$1I135/TXDIN8"  LOC = "AA16";
NET "$1I135/TXDIN9"  LOC = "AB17";
NET "$1I135/TXDIN10" LOC = "AC17";
NET "$1I135/TXDIN11" LOC = "Y16";
NET "$1I135/TXDIN12" LOC = "Y15";
NET "$1I135/TXDIN13" LOC = "AB16";
NET "$1I135/TXDIN14" LOC = "AC16";
NET "$1I135/TXDIN15" LOC = "AA15";
#
#   Spares to FPGAs
# (spare_clk[1:0] + spare[23:0] to InCtrls, vme[8:0] to VMEctrl)
NET "$1I135/CLK40-0" LOC = "AD2"; #spare_clk0, 40 MHz w/DCM-phase adjust
NET "$1I135/CLK40-1" LOC = "W15"; #spare_clk1,  "  as above  "
NET "$1I135/CLK40-2" LOC = "E8"; #vme8 out,    "  as above  "
NET "$1I135/~LD_RDY" LOC = "E9"; #vme7 out, Ready for Serial load?
NET "$1I135/~SEN6"   LOC = "D8"; #vme6 in, Serial load CE for Board ID?
NET "$1I135/~AUTOSLD_EN" LOC = "C8"; #vme5 in, Skip Serial Load phase
                                       # from VME_Ctrl ModeSW7 (bit6)
NET "$1I135/~INSLNKWT_EN" LOC = "B8"; #vme4 in, for Kill SLink_Wait
NET "$1I135/INPRSCL_SEL2" LOC = "A8"; #vme3 in, for GbE_PreScale[bit2]
NET "$1I135/INPRSCL_SEL1" LOC = "G9"; #vme2 in, for GbE_PreScale[bit1]
NET "$1I135/INPRSCL_SEL0" LOC = "F9"; #vme1 in, for GbE_PreScale[bit0]
NET "$1I135/INFAKEL1_EN"  LOC = "D9"; #vme0 in, turns on L1A_Fake
#
```

```
NET "$1I135/~IDMB_FUL0" LOC = "U7";#spare5 InCtrl0 InUnit detects DMB_Full
NET "$1I135/~IDMB_FUL1" LOC = "T8";#spare6 InCtrl1 InUnit detects DMB_Full
NET "$1I135/~IDMB_FUL2" LOC = "Y9";#spare14 InCtrl2 InUnt detects DMB_Full
NET "$1I135/~IDMB_FUL3" LOC = "AF8";#spare15 InCtl3 InUnt detects DMB_Full
#NET "$1I135/SPARE0"  LOC = "U5"; #spare0 to InFpga#0
#NET "$1I135/SPARE1"  LOC = "U4"; #spare1 to InFpga#0
#NET "$1I135/SPARE2"  LOC = "U3"; #spare2 to InFpga#0
#NET "$1I135/SPARE3"  LOC = "V1"; #spare3 to InFpga#0
#NET "$1I135/SPARE4"  LOC = "U1"; #spare4 to InFpga#0
#NET "$1I135/SPARE7"  LOC = "T7"; #spare7 to InFpga#0
#NET "$1I135/SPARE8"  LOC = "R7"; #spare8 to InFpga#0
#NET "$1I135/SPARE9"  LOC = "W11"; #spare9 to InFpga#0
#NET "$1I135/SPARE12" LOC = "AC9"; #spare12 to InFpga#1
#NET "$1I135/SPARE13" LOC = "AA9"; #spare13 to InFpga#1
#NET "$1I135/SPARE16" LOC = "AE8"; #spare16 to InFpga#1
#NET "$1I135/SPARE17" LOC = "AD8"; #spare17 to InFpga#1
#NET "$1I135/SPARE18" LOC = "K7"; #spare18 to InFpga#1
#NET "$1I135/SPARE19" LOC = "L7"; #spare19 to InFpga#1
#NET "$1I135/SPARE20" LOC = "M8"; #spare20 to InFpga#1
#NET "$1I135/SPARE21" LOC = "N8"; #spare21 to InFpga#1
#
#   data out to Output FIFO/S-Link
NET "$1I135/DOUT0"   LOC = "J20"; #outdata0
NET "$1I135/DOUT1"   LOC = "K20";
NET "$1I135/DOUT2"   LOC = "L19";
NET "$1I135/DOUT3"   LOC = "E16";
NET "$1I135/DOUT4"   LOC = "D15";
NET "$1I135/DOUT5"   LOC = "F14";
NET "$1I135/DOUT6"   LOC = "P25";
NET "$1I135/DOUT7"   LOC = "N25";
NET "$1I135/DOUT8"   LOC = "M26";
NET "$1I135/DOUT9"   LOC = "H22";
NET "$1I135/DOUT10"  LOC = "H21";
NET "$1I135/DOUT11"  LOC = "G21";
NET "$1I135/DOUT12"  LOC = "J22";
NET "$1I135/DOUT13"  LOC = "J21";
NET "$1I135/DOUT14"  LOC = "K22";
NET "$1I135/DOUT15"  LOC = "E23";
NET "$1I135/DOUT16"  LOC = "N19";
NET "$1I135/DOUT17"  LOC = "M19";
NET "$1I135/DOUT18"  LOC = "L20";
NET "$1I135/DOUT19"  LOC = "G14";
NET "$1I135/DOUT20"  LOC = "H14";
NET "$1I135/DOUT21"  LOC = "C15";
NET "$1I135/DOUT22"  LOC = "E18";
NET "$1I135/DOUT23"  LOC = "P19";
NET "$1I135/DOUT24"  LOC = "C19";
NET "$1I135/DOUT25"  LOC = "E24";
NET "$1I135/DOUT26"  LOC = "K21";
NET "$1I135/DOUT27"  LOC = "L22";
NET "$1I135/DOUT28"  LOC = "L21";
NET "$1I135/DOUT29"  LOC = "M22";
NET "$1I135/DOUT30"  LOC = "N21";
NET "$1I135/DOUT31"  LOC = "N22";
NET "$1I135/DOUT32"  LOC = "D19";
NET "$1I135/DOUT33"  LOC = "F15";
NET "$1I135/DOUT34"  LOC = "E19";
NET "$1I135/DOUT35"  LOC = "D20";
NET "$1I135/DOUT36"  LOC = "G15";
NET "$1I135/DOUT37"  LOC = "E20";
NET "$1I135/DOUT38"  LOC = "F18";
NET "$1I135/DOUT39"  LOC = "F16";
NET "$1I135/DOUT40"  LOC = "F17";
NET "$1I135/DOUT41"  LOC = "F19";
NET "$1I135/DOUT42"  LOC = "G23";
NET "$1I135/DOUT43"  LOC = "G24";
NET "$1I135/DOUT44"  LOC = "P22";
NET "$1I135/DOUT45"  LOC = "P21";
NET "$1I135/DOUT46"  LOC = "R22";
NET "$1I135/DOUT47"  LOC = "H16";
NET "$1I135/DOUT48"  LOC = "G22";
NET "$1I135/DOUT49"  LOC = "G17";
NET "$1I135/DOUT50"  LOC = "D18";
NET "$1I135/DOUT51"  LOC = "C21";
NET "$1I135/DOUT52"  LOC = "C23";
NET "$1I135/DOUT53"  LOC = "G18";
NET "$1I135/DOUT54"  LOC = "D21";
NET "$1I135/DOUT55"  LOC = "D24";
NET "$1I135/DOUT56"  LOC = "G16";
NET "$1I135/DOUT57"  LOC = "H15";
NET "$1I135/DOUT58"  LOC = "N20";
NET "$1I135/DOUT59"  LOC = "P20";
NET "$1I135/DOUT60"  LOC = "R19";
NET "$1I135/DOUT61"  LOC = "T25";
NET "$1I135/DOUT62"  LOC = "T26";
NET "$1I135/DOUT63"  LOC = "R21";
#   Status Bits to Output FIFO
NET "$1I135/DOUT64"  LOC = "M21"; #outdata64
NET "$1I135/DOUT65"  LOC = "M20";
#   data in from Output FIFO
NET "$1I135/IN0"     LOC = "D16"; #gig_data0
NET "$1I135/IN1"     LOC = "G26";
NET "$1I135/IN2"     LOC = "H25";
NET "$1I135/IN3"     LOC = "E17";
NET "$1I135/IN4"     LOC = "E26";
NET "$1I135/IN5"     LOC = "F26";
NET "$1I135/IN6"     LOC = "E15";
NET "$1I135/IN7"     LOC = "D26";
NET "$1I135/IN8"     LOC = "E25";
NET "$1I135/IN9"     LOC = "D25";
NET "$1I135/IN10"    LOC = "C26";
NET "$1I135/IN11"    LOC = "C25";
NET "$1I135/IN12"    LOC = "B24";
NET "$1I135/IN13"    LOC = "B26";
NET "$1I135/IN14"    LOC = "A25";
NET "$1I135/IN15"    LOC = "A19";
NET "$1I135/IN16"    LOC = "B19";
NET "$1I135/IN17"    LOC = "A24";
#   Use 4 I/O pins for RealFMM[3:0] from VME_Ctrl
NET "$1I135/REALFMM0" LOC = "F8"; #fmm0
NET "$1I135/REALFMM1" LOC = "D7";
NET "$1I135/REALFMM2" LOC = "E7";
NET "$1I135/REALFMM3" LOC = "C6";
#   Use 4 I/O pins for DDUFMM[3:0] to VME_Ctrl
NET "$1I135/DDUFMM0"  LOC = "D3"; #ctrl_stat0
NET "$1I135/DDUFMM1"  LOC = "A2";
NET "$1I135/DDUFMM2"  LOC = "B1";
NET "$1I135/DDUFMM3"  LOC = "C2";
#
#   Reserve 18 *2 Pads for 2 Logic Analyzer paths:
NET "$1I135/L0_0"   LOC = "AB23"; #la0_0
NET "$1I135/L0_1"   LOC = "AB24";
NET "$1I135/L0_2"   LOC = "Y21";
NET "$1I135/L0_3"   LOC = "Y22";
NET "$1I135/L0_4"   LOC = "Y23";
NET "$1I135/L0_5"   LOC = "Y24";
NET "$1I135/L0_6"   LOC = "AA26";
NET "$1I135/L0_7"   LOC = "Y26";
NET "$1I135/L0_8"   LOC = "W21";
NET "$1I135/L0_9"   LOC = "W22";
NET "$1I135/L0_10"  LOC = "W23";
NET "$1I135/L0_11"  LOC = "W24";
NET "$1I135/L0_12"  LOC = "W25";
NET "$1I135/L0_13"  LOC = "W26";
NET "$1I135/L0_14"  LOC = "V20";
NET "$1I135/L0_15"  LOC = "V21";
NET "$1I135/L0_16"  LOC = "AC26";  #la0_clk1
NET "$1I135/L0_17"  LOC = "AC25";  #la0_clk2
```

```
#
NET "$1I135/L1_0"   LOC = "V24";  #la1_0
NET "$1I135/L1_1"   LOC = "V25";
NET "$1I135/L1_2"   LOC = "U21";
NET "$1I135/L1_3"   LOC = "U22";
NET "$1I135/L1_4"   LOC = "U23";
NET "$1I135/L1_5"   LOC = "U24";
NET "$1I135/L1_6"   LOC = "V26";
NET "$1I135/L1_7"   LOC = "U26";
NET "$1I135/L1_8"   LOC = "U20";
NET "$1I135/L1_9"   LOC = "T19";
NET "$1I135/L1_10"  LOC = "T20";
NET "$1I135/L1_11"  LOC = "R20";
NET "$1I135/L1_12"  LOC = "T21";
NET "$1I135/L1_13"  LOC = "T22";
NET "$1I135/L1_14"  LOC = "T23";
NET "$1I135/L1_15"  LOC = "T24";
NET "$1I135/L1_16"  LOC = "V23";  #la1_clk1
NET "$1I135/L1_17"  LOC = "V22";  #la1_clk2
#
# Foundation 3.1 does not understand high/low 50%....but required
#       for 4.2 & 5.1
NET "$1I135/SCLKIN" period=75ns high 50%; #SCLKIN = sclk, 10mhz
NET "$1I135/CLKIN40" period=23ns high 50%; #CLKIN40 = ccbclk, 40mhz
#         T_clk40 = 11.43ns was met
NET "$1I135/CK625IN" period=13.8ns high 50%; #CK625IN = clk625, 62.5mhz
#         T_ck625 = 10.10ns was met     T_ck125 = 6.23ns was met
NET "$1I135/FCKIN_N" period=5.9ns high 50%;#FCKIN=clk156,156mhz: max=6.2ns
NET "$1I135/CKFB_IN" period=12.0ns high 50%;#CKFB=clk_fb,78mhz: max=12.0ns
#         T_clk = 11.04ns was met     T_fclk = 5.32ns was met
NET "$1I135/$1I3030/DRCK1" period=30ns high 50%;
NET "$1I135/$1I3030/DRCK2" period=30ns high 50%;
#
# FCKIN -> CLK156 comes from 156MHz oscillator, goes to FD to drive CLK78;
#  these are used to drive CKFB_OUT (78MHz, 1 load to FPGA:CKFB_IN, Slow
#  8mA) and IRCLK0/1 (156MHz, 2 loads each to InFIFOs, Fast 24/16mA).
#  CKFB_OUT is phase-delayed  by ~1.7ns due to Drive settings, but IRCLKs
#  get delayed up to 0.5ns due to extra loads, so:
#      at the pins, CKFB_IN is slower than IRCLKs by 1.2-1.6ns
#  Roundtrip time to FIFO is 0.3-0.6ns, and this can be absorbed into the
#  clock delays such that
#      **effectively, CKFB_IN is slower than IRCLKs 0.6 - 1.3ns **
#  This matches well with the FIFO RCLK-to-Output lag of 2-4.5ns.
# Meas'd FIFO RCLK-to-Output lag=2.5-3.6ns for Empty, 2.5-4.3ns for Data.
# Note that (CKFB_IN -> CLK/FCLK) has an IBUF delay and so does FIFO Data,
#  so the IBUF delays CANCEL OUT: check constraints for double-counting.
#  Constraints need only use Effective CKFB lag & FIFO RCLK-to-Output lag.
#
# Specify timing for CLK_REG-to-FCLK_REG as 6ns each way:
INST "$1I135/$1I3030/$8I2986" TNM=FFS FWCREG;
INST "$1I135/$1I3030/$8I3205" TNM=FFS FWCREG;
INST "$1I135/$1I3030/$8I3206" TNM=FFS FWCREG;
INST "$1I135/$1I3030/$8I3207" TNM=FFS FWCREG;
TIMESPEC "TS_FtoS" = FROM "FCLK" TO "CLK0" 6.0ns;# < 5.09ns was met
TIMESPEC "TS_StoF" = FROM "CLK0" TO "FCLK" 6.0ns;# < 2.67ns was met
# Specify timing for Empty-to-FCLK_REG accounting for FIFO-Output lag,
#   with Period=6.0ns:    < 2.87ns was met!
TIMESPEC TS_MTtoF=FROM:"INMT":TO:"FWCREG" 3.2ns;#assume ISE inc.IBUF delay
#         < 3.08ns was met
#TIMESPEC TS_MTtoF=FROM:"INMT":TO:"FWCREG" 2.4ns;#assume ISE inc.NoIBUFdly
#  MT & CLK IBUF delays cancel, but ISE probably double-counts: add 0.8ns?
#  Period - 3.6ns worstcase FIFOoutput + 0.8ns IBUF dly = 3.4ns TS_MTtoF
# Specify timing from ~CLK IFDDRs to CLK REGs as 6.0ns:
TIMESPEC "TS_QtoS" = FROM "FDAT" TO "CLK0" 6.0ns;# < 5.67ns was met
#
# FIFO REN has 1.5ns Setup, plus 1ns clk/rclk difference: 6.4-2.5=3.9ns
# Use when IRCLK driven by CLK156 via OFDDR:  assume that ISE inc. CKFB_IN
TIMEGRP "FREN" OFFSET = OUT 3.9ns AFTER "$1I135/CKFB_IN"; #    IBUFG delay
#         < 2.78ns was met
TIMEGRP "FOE" OFFSET = OUT 4.9ns AFTER "$1I135/CKFB_IN";
#         < 2.79ns was met
#PINLOCK_END
```

DDU VMEctrl FPGA UCF file

```
# use with "D785/DDU5vme: VME5CTRL" Logic schematic
#PINLOCK_BEGIN
#Sun Sep 27 20:30:16 2005
#INPUTS
# Control signals in:
NET "$1I135/CLK80P"   LOC = "V11";
NET "$1I135/CLK80N"   LOC = "W11";
NET "$1I135/CLKIN40"  LOC = "Y11"; #vme8 spare, on GCLKp pin?
NET "$1I135/~SYNCRST" LOC = "V10";
NET "$1I135/~SOFTRST" LOC = "V9";
#   from switches
NET "$1I135/SW0"  LOC = "B19";#to Switch1
NET "$1I135/SW1"  LOC = "A19";
NET "$1I135/SW2"  LOC = "D18";
NET "$1I135/SW3"  LOC = "C18";
NET "$1I135/SW4"  LOC = "B18";
NET "$1I135/SW5"  LOC = "A18";
NET "$1I135/SW6"  LOC = "D17";
NET "$1I135/SW7"  LOC = "C17";#to Switch8
#
#VME Signals in (and in/out): (all new)
#   VME Data (2-way I/O)
NET "$1I135/VMECLKIN" LOC = "AB12"; #clk_16
NET "$1I135/VMED0"    LOC = "E18";
NET "$1I135/VMED1"    LOC = "F18";
NET "$1I135/VMED2"    LOC = "D21";
NET "$1I135/VMED3"    LOC = "D22";
NET "$1I135/VMED4"    LOC = "E19";
NET "$1I135/VMED5"    LOC = "E20";
NET "$1I135/VMED6"    LOC = "E21";
NET "$1I135/VMED7"    LOC = "E22";
NET "$1I135/VMED8"    LOC = "G21";
NET "$1I135/VMED9"    LOC = "G22";
NET "$1I135/VMED10"   LOC = "H19";
NET "$1I135/VMED11"   LOC = "H20";
NET "$1I135/VMED12"   LOC = "H21";
NET "$1I135/VMED13"   LOC = "H22";
NET "$1I135/VMED14"   LOC = "J17";
NET "$1I135/VMED15"   LOC = "J18";
#   VME Control
NET "$1I135/~ISYSFAIL"  LOC = "J19";
NET "$1I135/~ISYSRESET" LOC = "J21";
NET "$1I135/OTOVME"     LOC = "C21";
NET "$1I135/~IDS0"      LOC = "K17";
NET "$1I135/~IDS1"      LOC = "J22";
NET "$1I135/~IWRITE"    LOC = "K19";
NET "$1I135/~IBERR"     LOC = "J20";
NET "$1I135/~ILWORD"    LOC = "K18";
NET "$1I135/~IAS"       LOC = "K20";
NET "$1I135/~INACK"     LOC = "L21";
NET "$1I135/~ODTACK"    LOC = "L22";
NET "$1I135/~IACKIN"    LOC = "AA15"; #was testpoint14
NET "$1I135/~IACKOUT"   LOC = "AB15"; #was testpoint13
NET "$1I135/~IRQ1"      LOC = "AA12"; #was testpoint7
# Ground ~DOE inside FPGA:
NET "$1I135/~DOE"    LOC = "C22"; # also for DDU4?
#   VME Address Mode
NET "$1I135/AM5"     LOC = "K21";
NET "$1I135/AM0"     LOC = "K22";
NET "$1I135/AM1"     LOC = "L17";
```

```
NET "$1I135/AM2"     LOC = "L18";
NET "$1I135/AM3"     LOC = "L19";
NET "$1I135/AM4"     LOC = "L20";
#   VME Address
NET "$1I135/VMEA1"   LOC = "M20";
NET "$1I135/VMEA2"   LOC = "M19";
NET "$1I135/VMEA3"   LOC = "M18";
NET "$1I135/VMEA4"   LOC = "M17";
NET "$1I135/VMEA5"   LOC = "N17";
NET "$1I135/VMEA6"   LOC = "N22";
NET "$1I135/VMEA7"   LOC = "N21";
NET "$1I135/VMEA8"   LOC = "N20";
NET "$1I135/VMEA9"   LOC = "N19";
NET "$1I135/VMEA10"  LOC = "N18";
NET "$1I135/VMEA11"  LOC = "P18";
NET "$1I135/VMEA12"  LOC = "P22";
NET "$1I135/VMEA13"  LOC = "P21";
NET "$1I135/VMEA14"  LOC = "P20";
NET "$1I135/VMEA15"  LOC = "P19";
NET "$1I135/VMEA16"  LOC = "R22";
NET "$1I135/VMEA17"  LOC = "R21";
NET "$1I135/VMEA18"  LOC = "R20";
NET "$1I135/VMEA19"  LOC = "R19";
NET "$1I135/VMEA20"  LOC = "R18";
NET "$1I135/VMEA21"  LOC = "P17";
NET "$1I135/VMEA22"  LOC = "T22";
NET "$1I135/VMEA23"  LOC = "M21";
#   Global Address
NET "$1I135/IGA4"    LOC = "V22";
NET "$1I135/IGA3"    LOC = "V21";
NET "$1I135/IGA2"    LOC = "V20";
NET "$1I135/IGA1"    LOC = "V19";
NET "$1I135/IGA0"    LOC = "W22";
NET "$1I135/IGAP"    LOC = "T21";
#
# DMB Status signals in
NET "$1I135/DMB_STAT0"  LOC = "F9";
NET "$1I135/DMB_STAT1"  LOC = "E9";
NET "$1I135/DMB_STAT2"  LOC = "A8";
NET "$1I135/DMB_STAT3"  LOC = "B8";
NET "$1I135/DMB_STAT4"  LOC = "E8";
NET "$1I135/DMB_STAT5"  LOC = "E7";
NET "$1I135/DMB_STAT6"  LOC = "A6";
NET "$1I135/DMB_STAT7"  LOC = "B6";
NET "$1I135/DMB_STAT8"  LOC = "C6";
NET "$1I135/DMB_STAT9"  LOC = "D6";
NET "$1I135/DMB_STAT10" LOC = "A5";
NET "$1I135/DMB_STAT11" LOC = "B5";
NET "$1I135/DMB_STAT12" LOC = "C5";
NET "$1I135/DMB_STAT13" LOC = "C4";
NET "$1I135/DMB_STAT14" LOC = "A4";
NET "$1I135/DMB_STAT15" LOC = "B4";
NET "$1I135/DMB_STAT16" LOC = "E6";
NET "$1I135/DMB_STAT17" LOC = "E5";
NET "$1I135/DMB_STAT18" LOC = "C2";
NET "$1I135/DMB_STAT19" LOC = "C1";
NET "$1I135/DMB_STAT20" LOC = "D2";
NET "$1I135/DMB_STAT21" LOC = "D1";
NET "$1I135/DMB_STAT22" LOC = "E4";
NET "$1I135/DMB_STAT23" LOC = "E3";
NET "$1I135/DMB_STAT24" LOC = "E2";
NET "$1I135/DMB_STAT25" LOC = "E1";
NET "$1I135/DMB_STAT26" LOC = "G2";
NET "$1I135/DMB_STAT27" LOC = "G1";
NET "$1I135/DMB_STAT28" LOC = "H5";
NET "$1I135/DMB_STAT29" LOC = "J6";
NET "$1I135/DMB_STAT30" LOC = "H4";
NET "$1I135/DMB_STAT31" LOC = "H3";
NET "$1I135/DMB_STAT32" LOC = "H2";
NET "$1I135/DMB_STAT33" LOC = "H1";
NET "$1I135/DMB_STAT34" LOC = "J4";
NET "$1I135/DMB_STAT35" LOC = "J3";
NET "$1I135/DMB_STAT36" LOC = "J2";
NET "$1I135/DMB_STAT37" LOC = "J1";
NET "$1I135/DMB_STAT38" LOC = "J5";
NET "$1I135/DMB_STAT39" LOC = "K5";
NET "$1I135/DMB_STAT40" LOC = "K6";
NET "$1I135/DMB_STAT41" LOC = "L6";
NET "$1I135/DMB_STAT42" LOC = "K4";
NET "$1I135/DMB_STAT43" LOC = "K3";
NET "$1I135/DMB_STAT44" LOC = "K2";
NET "$1I135/DMB_STAT45" LOC = "K1";
NET "$1I135/DMB_STAT46" LOC = "L5";
NET "$1I135/DMB_STAT47" LOC = "L4";
NET "$1I135/DMB_STAT48" LOC = "L3";
NET "$1I135/DMB_STAT49" LOC = "L2";
NET "$1I135/DMB_STAT50" LOC = "W6";
NET "$1I135/DMB_STAT51" LOC = "Y6";
NET "$1I135/DMB_STAT52" LOC = "W8";
NET "$1I135/DMB_STAT53" LOC = "Y8";
NET "$1I135/DMB_STAT54" LOC = "AA8";
NET "$1I135/DMB_STAT55" LOC = "AB8";
NET "$1I135/DMB_STAT56" LOC = "W9";
NET "$1I135/DMB_STAT57" LOC = "Y9";
NET "$1I135/DMB_STAT58" LOC = "AA9";
NET "$1I135/DMB_STAT59" LOC = "AB9";
#  DDU_Ctrl Status signals in
NET "$1I135/CTRL_STAT0" LOC = "Y10";
NET "$1I135/CTRL_STAT1" LOC = "AA10";
NET "$1I135/CTRL_STAT2" LOC = "AB10";
NET "$1I135/CTRL_STAT3" LOC = "U10";
#
#OUTPUTS
NET "$1I135/~DLL_READY2"    LOC = "U11";
#   Use 6 I/O pins for RealFMM[5:0]
NET "$1I135/REALFMM0"   LOC = "Y21"; #fmm0
NET "$1I135/REALFMM1"   LOC = "W20";
NET "$1I135/REALFMM2"   LOC = "W21";
NET "$1I135/REALFMM3"   LOC = "Y22";
NET "$1I135/REALFMM4"   LOC = "AA20"; #RJ45 LED  yel
NET "$1I135/REALFMM5"   LOC = "Y12"; #RJ45 LED? grn
# Error Signals out:
#   to LEDs (use parallel-load-data pins).  Set in Schematic now: LED1-8.
#
# Control Signals out:
#   fake JTAG ports
NET "$1I135/OTDI1"  LOC = "B11";  #tdi1, to VME Prom
NET "$1I135/ITDO1"  LOC = "F12";  #tdo1, from VME Prom
NET "$1I135/OTCK1"  LOC = "C11";  #tck1
NET "$1I135/OTMS1"  LOC = "A11";  #tms1
NET "$1I135/OTDI2"  LOC = "P6"; #tdi2, to InCtrl FPGA 0
NET "$1I135/ITDO2"  LOC = "P5"; #tdo2, from InCtrl FPGA 0
NET "$1I135/OTCK2"  LOC = "P4";
NET "$1I135/OTMS2"  LOC = "P3";
NET "$1I135/OTDI3"  LOC = "R4"; #tdi3, to InCtrl FPGA 1
NET "$1I135/ITDO3"  LOC = "R3"; #tdo3, from InCtrl FPGA 1
NET "$1I135/OTCK3"  LOC = "R2";
NET "$1I135/OTMS3"  LOC = "R1";
NET "$1I135/OTDI4"  LOC = "P2";  #tdi4, to InCtrl Proms
NET "$1I135/ITDO4"  LOC = "P1";  #tdo4, from InCtrl Proms
NET "$1I135/OTCK4"  LOC = "N6";
NET "$1I135/OTMS4"  LOC = "N5";
NET "$1I135/OTDI5"  LOC = "U4"; #tdi5, to SLINK
NET "$1I135/ITDO5"  LOC = "U3"; #tdo5, from SLINK
NET "$1I135/OTCK5"  LOC = "T2";
NET "$1I135/OTMS5"  LOC = "T1";
NET "$1I135/OTDI6"  LOC = "V4"; #tdi6, to DDU_Ctrl Prom
NET "$1I135/ITDO6"  LOC = "V3"; #tdo6, from DDU_Ctrl Prom
```

```
NET "$1I135/OTCK6"  LOC = "W2";
NET "$1I135/OTMS6"  LOC = "W1";
NET "$1I135/OTDI7"  LOC = "V5";  #tdi7, to Output FIFO
NET "$1I135/ITDO7"  LOC = "U5";  #tdo7, from Output FIFO
NET "$1I135/OTCK7"  LOC = "Y2";
NET "$1I135/OTMS7"  LOC = "Y1";
NET "$1I135/OTDI8"  LOC = "N4"; #tdi8, to DDU_Ctrl FPGA
NET "$1I135/ITDO8"  LOC = "N3"; #tdo8, from DDU_Ctrl FPGA
NET "$1I135/OTCK8"  LOC = "N2";
NET "$1I135/OTMS8"  LOC = "N1";
#   Serial path control
#     Flash RAM:
NET "$1I135/M_WP"   LOC = "AA18";
NET "$1I135/M_SO"   LOC = "AB18";
NET "$1I135/M_SI"   LOC = "W17";
NET "$1I135/M_SCLK" LOC = "Y17";
NET "$1I135/M_RST"  LOC = "AA17";
NET "$1I135/M_CS"   LOC = "AB17";
#     Serial ADC (temp & volt monitors)
NET "$1I135/ADC_CLK"  LOC = "U14";#
NET "$1I135/ADC_DIN"  LOC = "V14";#
NET "$1I135/ADC_DOUT" LOC = "W14";#
NET "$1I135/~ADC_CS"  LOC = "W12";#
#     Other Serial destinations:
NET "$1I135/OSCLK"  LOC = "AA5";#sclk, out to destinations
NET "$1I135/SI"     LOC = "AB5";# out to destinations
NET "$1I135/~SEN4"  LOC = "U12";#Output FIFO
NET "$1I135/~SEN5"  LOC = "V12";#DDU_Ctrl FPGA
#
#Data signals out:
#   Spare signals to DDU_Ctrl, InCtrl
NET "$1I135/VME0"  LOC = "AB14"; # out for Fake_L1A_Enable to DDUctrl
NET "$1I135/VME1"  LOC = "U13";  # out for GbE_PreScale[bit0]
NET "$1I135/VME2"  LOC = "V13";  # out for GbE_PreScale[bit1]
NET "$1I135/VME3"  LOC = "W13";  # out for GbE_PreScale[bit2]
NET "$1I135/VME4"  LOC = "Y13";  # out for Kill SLink_Wait/Enable DAVLCT
#                                                    check from DMB
NET "$1I135/VME5"  LOC = "AA13";#use as AutoSLD_EN signal to DDU_Ctrl
NET "$1I135/VME6"  LOC = "AB13";#use as ~LD6 (Board ID) signal to DDUctrl
NET "$1I135/VME7"  LOC = "Y14"; #use as ~LD_Ready signal from DDU_Ctrl
#NET "$1I135/INPUT0"  LOC = "M6";# to/from InFPGA0
#NET "$1I135/INPUT1"  LOC = "M5";
NET "$1I135/INPUT2"  LOC = "M4"; # in for EvCntRst from InFPGA0
NET "$1I135/INPUT3"  LOC = "M3"; # out for Fake_L1A_Enable to InFPGA0
#NET "$1I135/INPUT4"  LOC = "M2";# to/from InFPGA1
#NET "$1I135/INPUT5"  LOC = "M1";
NET "$1I135/INPUT6"  LOC = "V7"; # in for BC0 from InFPGA1
NET "$1I135/INPUT7"  LOC = "V6"; # out for Fake_L1A_Enable to InFPGA1
#    Testpoints:
#NET "$1I135/TP_7"   LOC = "AA12";# testpoint7
NET "$1I135/TP_12"  LOC = "W10";# testpoint12
#NET "$1I135/TP_13"  LOC = "AB15";# testpoint13
#NET "$1I135/TP_14"  LOC = "AA15";# testpoint14
NET "$1I135/TP_76"  LOC = "AA11";# testpoint76
#NET "$1I135/TP_77"  LOC = "Y11";# testpoint77
NET "$1I135/L0_0"   LOC = "E17"; #la0_0
NET "$1I135/L0_1"   LOC = "E16";
NET "$1I135/L0_2"   LOC = "D15";
NET "$1I135/L0_3"   LOC = "C15";
NET "$1I135/L0_4"   LOC = "B15";
NET "$1I135/L0_5"   LOC = "A15";
NET "$1I135/L0_6"   LOC = "D14";
NET "$1I135/L0_7"   LOC = "C14";
NET "$1I135/L0_8"   LOC = "B14";
NET "$1I135/L0_9"   LOC = "A14";
NET "$1I135/L0_10"  LOC = "E14";
NET "$1I135/L0_11"  LOC = "E13";
NET "$1I135/L0_12"  LOC = "D13";
NET "$1I135/L0_13"  LOC = "C13";
NET "$1I135/L0_14"  LOC = "B13";
NET "$1I135/L0_15"  LOC = "A13";
NET "$1I135/L0_16"  LOC = "A17"; #la0_clk1
NET "$1I135/L0_17"  LOC = "B17"; #la0_clk2
NET "$1I135/L1_0"   LOC = "D12"; #la1_0
NET "$1I135/L1_1"   LOC = "E12";
NET "$1I135/L1_2"   LOC = "F13";
NET "$1I135/L1_3"   LOC = "D11";
NET "$1I135/L1_4"   LOC = "F11";
NET "$1I135/L1_5"   LOC = "E11";
NET "$1I135/L1_6"   LOC = "A10";
NET "$1I135/L1_7"   LOC = "B10";
NET "$1I135/L1_8"   LOC = "C10";
NET "$1I135/L1_9"   LOC = "D10";
NET "$1I135/L1_10"  LOC = "F10";
NET "$1I135/L1_11"  LOC = "E10";
NET "$1I135/L1_12"  LOC = "A9";
NET "$1I135/L1_13"  LOC = "B9";
NET "$1I135/L1_14"  LOC = "C9";
NET "$1I135/L1_15"  LOC = "D9";
NET "$1I135/L1_16"  LOC = "C12"; #la1_clk1
NET "$1I135/L1_17"  LOC = "B12"; #la1_clk2
#
# Foundation 3.1 does not understand high/low 50%....but required
#       for 4.2 & 5.1 & beyond
NET "$1I135/CKIN80" period=11.5ns high 50%; # 80mhz, 10.18ns was met
NET "$1I135/CLKIN40" period=23ns high 50%; # 40mhz, 7.51ns was met
#          T_clk10 = 90.0ns was met
#PINLOCK_END
```