ARM-FPGA VHDL Signal Naming.

Overview:

The back panel connector signals are fed to the Line driver boards, the FPGA connections of those boards are fed to the ST1 base board using the Enclustra designated IO pins. The design top-level file (`Mercury_XU5_ST1.vhd`) maps the signals to vectors for each axis. In the Vivado block diagram, these axis vectors are directly connected to the ports of the Panda1 RTL Module. The top level file of the RTL module (`Panda1.vhd`) then split sthem off into meaningful signal names, such as `pins_ENC_A_in(0)` which is the Encoder A input for the first axis.

Hardware:

```
    Back-Panel Axis_n  <--> {  Line-Driver Board.           } <--> { IO0_D0_P   Mercury ST1 }
                            {  FPGA schematic names:        }      { IO0_D2_P   (with XU6)  }
    powerPMAC  Axis_n  <--> {      FPGA_Inn_x, FPGA_OUTn_x  }      {    ...                 }

   (n is the axis number)     (these match the Line                  (these match the
                                      Driver Schematic)                    ST1 schematic)
```

Mercury_XU5_ST1.vhd:

```
Ports:
 IO0_D0_P  <-- These are directly assigned to -->  AXIS_n_IN  : in  std_logic_vector(7 downto 0);
 IO0_D2_P                                          AXIS_n_OUT : out std_logic_vector(7 downto 0);
   ...                                                          ...
 FMC_LA02_N
   ...

(these match the                                    (these match FPGA_INn_x and FPGA_OUTn_x
    ST1 schematic)                                               on the Line Driver Schematic)
```

Vivado Block Diagram:

```
    AXIS_n_IN    <-- are directly connected to -->    LD_FPGA_IN_AXIS_n
    AXIS_n_OUT                                        LD_FPGA_OUT_AXIS_n
      ...                                                  ...
                                          (these are ports on the Panda1.vhd RTL Module)
```

Panda1.vhd:

Ports:

```
        LD_FPGA_IN_AXIS_n
        LD_FPGA_OUT_AXIS_n
            ...
```

IO remapping (beginning at line 325)...

The individual bits (ie quad encoder A, B etc) are mapped from the 'vector-per-axis' to vectors containing one of the bits for all of the axes. For example:

```
            pins_ENC_A_in(0)      <= LD_FPGA_IN_AXIS_1(0);
            pins_ENC_B_in(0)      <= LD_FPGA_IN_AXIS_1(1);
                 ...
            pins_ENC_A_in(1)      <= LD_FPGA_IN_AXIS_2(0);
            pins_ENC_B_in(1)      <= LD_FPGA_IN_AXIS_2(1);
                 ...
            pins_PMAC_SCLK_RX(0)  <= LD_FPGA_IN_AXIS_1(3);
            pins_ENC_SDA_RX(0)    <= LD_FPGA_IN_AXIS_1(4);
                 ...
```

The vector `pins_ENC_A_in(7 downto 0)` now contains the A inputs for all 8 axes, so that the 'generate' function can step through them as needed.

Other Hardware Ports in Panda1.vhd:

```
--SPI to PIC for Mode Select (relays)

PIC_SPI_SC          : out std_logic;
PIC_SPI_CS          : out std_logic;
PIC_SPI_DI          : out std_logic;
PIC_SPI_DO          : in  std_logic;
```

These are for communication with the PIC24 micro-controller on the adaptor board which controls the relays on the Line Driver Boards. Turning them on/off enable/disables the serial pass-trough function.
The instance of the function handling this communication is at line 842.

```
--AUX SPI/TTL Board (Back Panel)

AUX_IO_DATA     : inout std_logic_vector(5 downto 1);    -- Bi-directional data lines

AUX_IO_DIR      : out std_logic_vector(5 downto 1);      -- Data direction...
                                                         -- 1: Output  0: Input

AUX_IO_TTL_IN   : in  std_logic_vector(1 downto 0);       -- 5V Logic Inputs

AUX_IO_TTL_OUT  : out std_logic_vector(1 downto 0);       -- 5V Logic Outputs
```

The Auxilliary SPI/TTL board provides 5xRS485 bi-directional line transceivers, primarily intended to communicate with external devices such as DACs. So far only a very simple hardware test has been implemented (line 1001).

There are also two TTL inputs and two TTL outputs, which have been coded to work with Panda TTL in/out blocks.

The front panel LEDs have also not yet been coded properly, but are attached as follows.

```
-- Front Panel PL LEDs

PL_LED_P          : out std_logic_vector(1 downto 0);
```

Finally, there are inputs for the EQU outputs from the PMAC (for fast signalling from the PMAC to the Panda) and also watchdog and abort inputs in case they come in useful (ie when writing a custom high-speed motor control loop using the external SPI DAC).

```
-- EQU inputs from PMAC

EQU_IN            : in  std_logic_vector(8 downto 1);


-- Watchdog and Abort inputs from PMAC

PMAC_WATCHDOG     : in  std_logic;
PMAC_ABORT        : in  std_logic;
```