Physics and Astronomy Dept.
UBC
SCUBA-2 Project

Ron Johnson
6/6/2007 12:33 PM
SC2-ELE-S589-201
Version 1.1

Sync Box CPLD Firmware Description

Revision History:

Rev. 1.a RJ Mar. 29, 2007 Draft

Rev. 1.1 MA Jun. 7, 2007 added document number and revision history

(For Firmware version SC2-SyncBox-6c)

Files:

File: SyncBox.bdf
Type: Block Diagram File
Function: This is the top-level file, that connects all the files in the project.

File: PIO_Interface.vhd
Function: This forms the interface between the AT89C5131 Command Processor and the Altera MAXII CPLD which does the counting, etc., for sync generation. It also flags whether the Sync output is to be taken from the external **DV_RTS**, or from the internal **DV_FreeRun** counter, and buffers the Reset and Enable signals. There are eight output bits, and eight inputs, used for ACDCU control and status io.

This version of PIO_Interface uses an all programmed io from the cmd-processor with an 8 bit address and 8 bits of data. A 32 bit byte-addressable Command_Data register buffers the value to be loaded; the endian problem is handled by the Command Processor. A following write to the Command_Select address in the CPLD will load the value in the CmdData register to the designated counter preset register.

```
CMD_SL_LOAD          : STD_LOGIC_VECTOR := X"01"; --
CMD_FR_LOAD          : STD_LOGIC_VECTOR := X"02"; --
CMD_DV_CNTR_LOAD     : STD_LOGIC_VECTOR := X"04"  --
```

Interface signals:

```
PIO_DAT      : INOUT STD_LOGIC_VECTOR(7 downto 0); = CmdProcessor P0
PIO_ADR      : IN STD_LOGIC_VECTOR(7 downto 0);    = CmdProc P2
PIO_ALE      : IN STD_LOGIC;      not used
PIO_nWR      : IN STD_LOGIC; = CmdProc P3.6
PIO_nRD      : IN STD_LOGIC; = CmdProc P3.7
PIO_Reset    : IN STD_LOGIC; = CmdProc P1.0
PIO_nEnable  : IN STD_LOGIC; = CmdProc P1.1
```

File: Sync_Len.vhd
Function: This is the Sync_Length counter, which counts cycles of 25 MHz. It counts from a settable number equal to (**Row_Length** x **Num_Rows**) -1, down to zero. When the count equals zero, the counter will output **isAddr_Zero**, reload the count on the next clock cycle, and continue.

File: FreeRun.vhd
Function: This implements a counter that functions similar to the Sync_Length counter, but which continuously down-counts occurrences of isAddr_Zero from a settable count, and outputs DV_FreeRun, when a zero count is reached.

File: ManchEncode.vhd
Function: This module contains:

1. Logic to handle the registering and selection of the **Data_Valid** source, and a counter to stretch DV_Out to 1µS for external use.
2. A 32 bit Frame Count register that counts occurences of Data_Valid.
3. Two 40 bit shift registers, one shifting at 25 MHz, and the other at 5 MHz. These registers get loaded when an Addr_Zero occurs. If a Data_Valid has occurred during the prior Sync_Length count, then the Frame Count and StatusBits will also be loaded and sent as well as the Addr_Zero and Datat_Valid bits. Otherwise only the Addr_Zero and all ones are loaded and sent.
4. Logic to generate the Manchester encoded 25 MHz serial bit stream from the output of the shift register.
5. A six bit Status Register, which is used to flag that a **DV_Error** has occurred, and whether DV_FreeRun mode is enabled.

File: TestPnts1.vhd
Function: This handles the output of various internal signals to TESTHDR1, P22, for test and development use.

File: TestPnts2.vhd
Function: This outputs signals to dedicated test points, and also drives the DV_Error and FR_Mode front panel LEDs.

File: ClkDiv.vhd
Function: As the name implies, it divides the 100 MHz clock down to the other frequencies required.

Other Bits In The Source Directory:

The following directories, located in the main source directory, contain test and simulation versions of the various files.

ManchEncode-5a
ZZ-clkdiv

ZZ-edgit
ZZ-FreeRun-4c
ZZ-PIOInterface-3b
ZZ-sync_len

The following directory contains an alternitive, reference, version of the PIO_Interface.
ZZ-PIO-Xilinx8051Interface

VECTOR_WAVEFORM_FILE "testo-S1.vwf"
VECTOR_WAVEFORM_FILE "testo-m1.vwf"