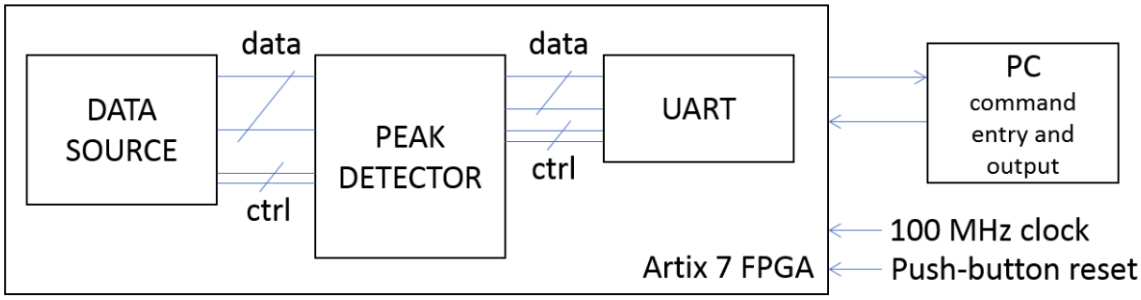

Fig. 2.1 - Peak detector system overview

Table 2.1 - Peak Detector Commands

Command	Description
ANNN or aNNN where NNN are 3 decimal digits.	Start command to process NNN bytes of data. Print each byte processed in the terminal window in hexadecimal format, separated by spaces. For example bytes "11111111" and "10101001" in sequence are to be printed as "FF A9".
L or l	Print the 3 bytes preceding the peak in the order they were received, the peak byte itself and the 3 bytes following the peak in the order received. They should be printed in hexadecimal format and separated by spaces. For example, the output of typing this command after processing a data sequence may be "09 FA A0 FD BC 10 DE".
P or p	Print the peak byte itself, followed by a space and the peak index in decimal format. For example, the output of typing this command after processing a data sequence may be "FD 197".
Push button BTN1 (reset)	Initialise the system when pressed at any point, including in the middle of a run. This is a synchronous reset and a reset command should not be required for the system to function normally, except after first loading the design.

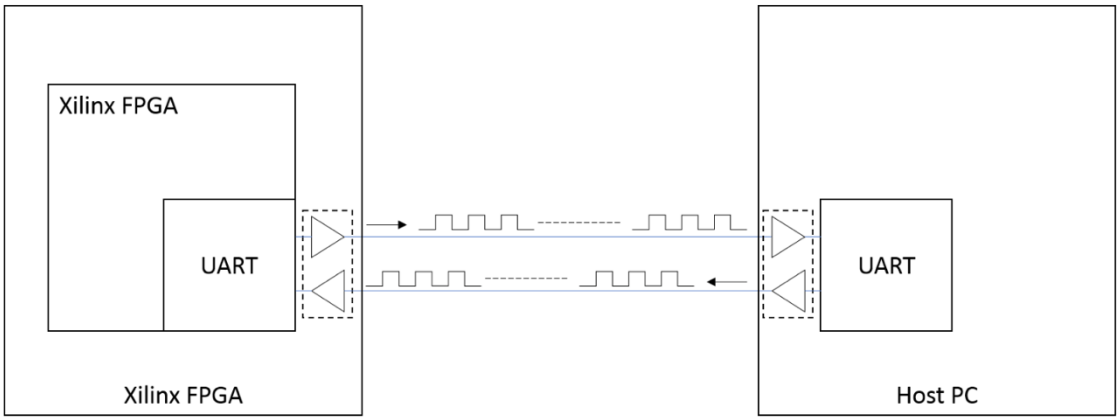

Fig. 3.1 - Peak detector system overview

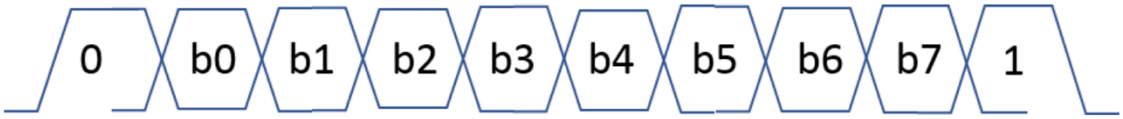

Fig. 3.2 - Peak detector system overview

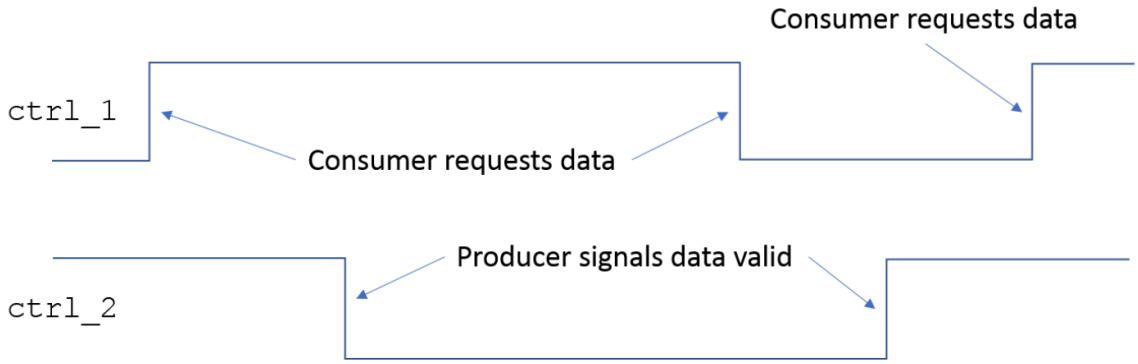

Fig. 3.5 - Peak detector system overview

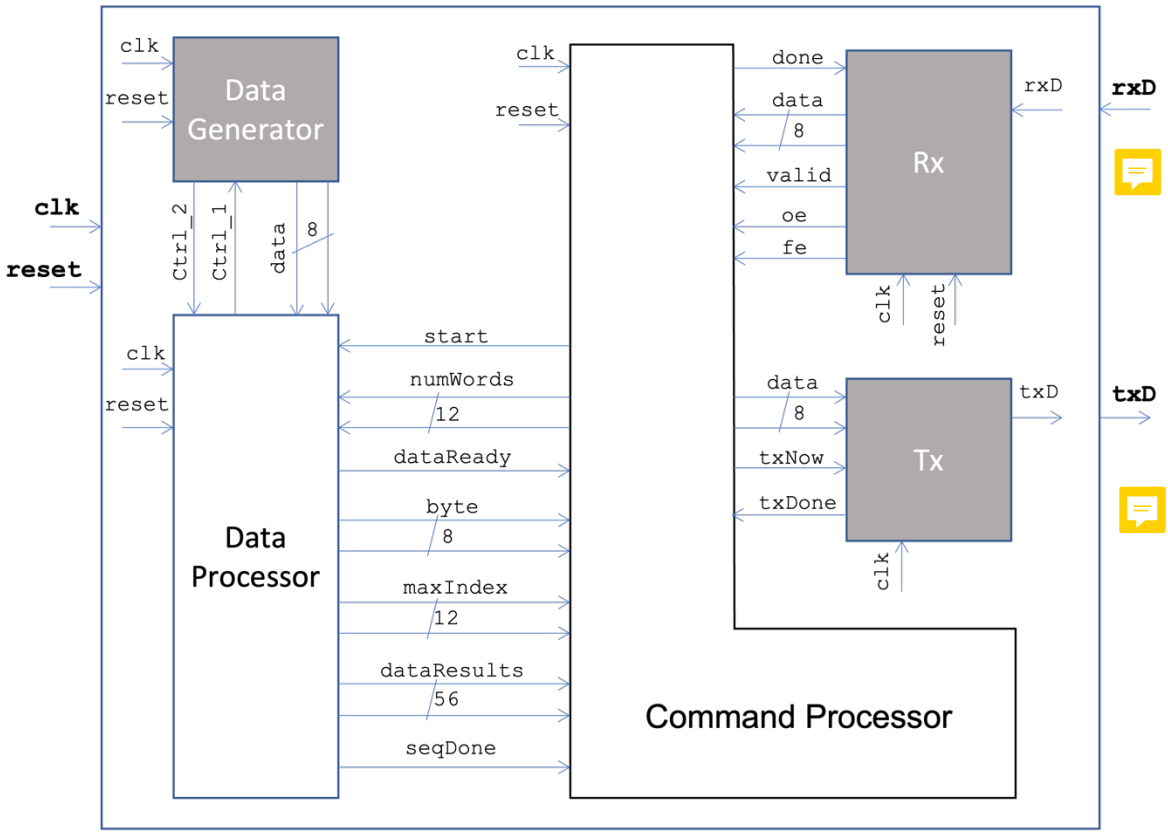

Table 4.1 - UART Transmitter (Tx) Interface

Signal	Description
`txD`	Serial output data line. This signal is connected to the external output pin on the FPGA, that, in turn, is connected to the serial data input of the receiver on the PC side.
`data`	Contains the data to be serialised and transmitted. Must be valid in the clock cycle that `send` is high.
`txNow`	Called `send` in the Tx port description. Used to trigger a send operation. The ***Command Processor*** should set this signal high for a single clock cycle to trigger a send. When this signal is set high `data` must be valid. Should not be asserted unless `txDone` is high.
`txDone`	Called `ready` in the Tx port description. This signal goes low once a send operation has begun and remains low until it has completed and the module is ready to send another byte.
`clk`	System clock. A 100 MHz clock is expected.

Table 4.2 - UART Receiver (Rx) Interface

Signal	Description
`rxD`	Serial input data line. This signal is connected to the external input pin on the FPGA, that, in turn, is connected to the serial data output of the transmitter on the PC side.
`data`	The received data is parallelised and made available on this 8-bit wide bus. Data for current word is valid when `valid` is high.
`valid`	Called `dataReady` in the Rx port description. Goes high when all of the bits in a serial transmission of a word have been detected. It goes low when `done` is set high by the upper layer logic, to be ready to receive the next word and indicate completion of the current word.
`done`	Called `rxDone` in the Rx port description. Used to signal to the receiver that data on the bus has been successfully read, and the register can be cleared. Should be set high by the upper layer logic (i.e. Command Processor) for 1 clock cycle. This signal is read by the Rx module to reset the data ready signal and also to check for overrun errors.
`clk`	System clock. A 100 MHz clock is expected.
`reset`	Synchronous reset.
`oe`	Goes high when an overrun error is detected; i.e. a new word is received, but upper layer logic has not signalled that previous byte has been read.
`fe`	The start and stop bits are not detected in order according to the RS232 protocol; i.e. a framing error has occurred.

Table 4.3 - Command Processor <=> Data Processor Interface

Signal	Description
`start`	A data retrival cycle is initiated by `start` being driven high. The ***Data Consumer*** should check this value on a rising edge of the clock. The value of `start` being high at the beginning of a clock cycle is a signal for data retrieval to take place; if the value is low, data retrieval will not take place. Used to start and halt data retrieval in the ***Data Processor***, while the ***Command Processor*** communicates with the PC via the serial link, which operates at a much lower frequency than the clock rate.
`numWords`	A 12 bit wide set of data lines that contain the number of bytes to process, in binary-coded-decimal (BCD) format. Each decimal digit is encoded in 4 bits.
`dataReady`	Asserted (active-high) by the ***Data Processor*** to signify that a new byte of data that has been supplied by the ***Data Generator*** is ready on the 8-bit `byte` line.
`byte`	Contains the latest 8-bits wide data word retrieved from the ***Data Generator***. Data is valid when `dataReady` is high.
`seqDone`	Asserted (active high) for one clock cycle when the number of bytes specified in `numWords` has been processed, and the 7 bytes comprising the peak and the 3 bytes either side of it are contained in `dataResults`, which is 56 bits wide.
`dataResults`	A 56-wide set of lines that contain the 7 bytes comprising the peak in the middle (i.e. bit indices 31 down to 23), and the 3 bytes either side of it. Data is valid when `seqDone` is high.
`maxIndex`	Contains the index of the peak byte in BCD format.