10GB/s Multi-Stream FPGA Data Recording

Using StoreEngine™ and StorePak XMC™ in Recording Mode

Abstract

The Critical I/O StoreEngine/StorePak XMC combination provides ultra-high performance data recording capabilities with a high degree of configurability and scalability. This paper describes a demonstration recording configuration that includes recording sensor data from four VPX FPGA boards connected to two StoreEngine/StorePak XMC VPX boards, recording at a sustained aggregate rate of 10 GB/sec. The system also includes an optional controller SBC, connected to both StoreEngine boards, which can be used to provide high level control of the recording functionality. A proof of concept demonstration system was assembled and tested. Benchmarking of this system verified that StoreEngine/StorePak XMC combination can record at the required 10GB/s aggregate rate as well as playback recorded data to the FPGAs at the same 10GB/s aggregate rate.

StoreEngine and StorePak – Recorder Building Blocks

StoreEngine and StorePak are flexible storage building blocks that can be used to implement a wide range of data storage systems. In the system described in this paper, StoreEngine functions as an ultra-high performance VPX storage controller blade providing high performance recording functionality. StoreEngine has an integrated Xeon-D storage management processor, along with 8GB to 16GB of DDR4 buffer memory with a bandwidth of 34GB/s to support high performance recording.

StoreEngine provides rich PCIe connectivity, with four x4 PCIe backplane ports per 3U VPX board. These ports are used for connections to the recording data sources as well as for interconnections between the StoreEngines (and external StorePak VPX blades, if used). StoreEngine also features a PCIe switch which is fully partitionable and supports NT bridging, providing greatly increased system PCIe architecture options.

StoreEngine can host a StorePak XMC that provides up to 12 TB of on-board SSD storage, all in a single 3U VPX slot. Together, StoreEngine and StorePak XMC provide unmatched storage capability, ultra-high performance and high capacity within a very small size, weight, and power (SWaP) footprint.

Figure 1. 3U CC StoreEngine + StorePak XMC

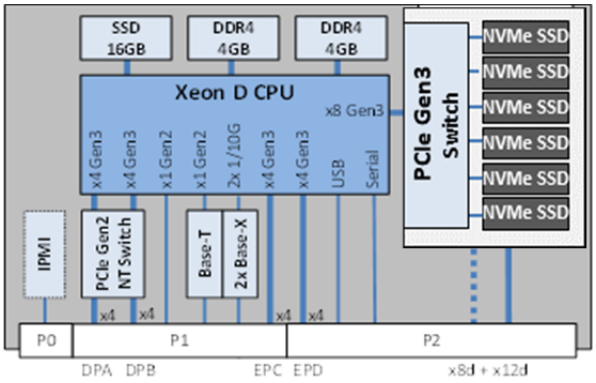

Figure 2. Block Diagram of 3U StoreEngine + StorePak XMC

Section 1: StoreEngine/StorePak XMC Data Recorder Overview

StoreEngine is pre-loaded with highly configurable data recorder software that allows data streams from multiple sources to be recorded to StorePaks (either XMC or 3U VPX blade) storage at ultra-high data rates. The recorder software supports recording from multiple stream sources of multiple types.

Data flow and recording is completely managed by StoreEngine and stored as files using a specialized low overhead file system. The recorded data can then be played back to the same devices, or to different devices through either PCIe or Ethernet connections.

Recording Source Stream Type Options

Stream sources are devices that source the data into memory buffers, which are then written to StorePak SSD storage by StoreEngine recording software. The recording software supports a number of stream source types, with dedicated recording mode software for each stream type. Stream source

options include Ethernet (UDP, TCP), sensors (FPGA, ADC, etc.), Fibre Channel, PCIe connected SBCs, and Local. Local is a unique stream type in that it is used for recording data from buffers hosted on the StoreEngine, where the user has implemented their own alternate functionality to move the data into these buffers. Depending on the source type, the data sources may be connected over the backplane (PCIe, 1/10 GbE) or through the use of interface XMCs (1/10/25 GbE, Fibre Channel).

This FPGA recording demonstration uses the FPGA stream type.

Recording Modes of Operation

The StoreEngine recording software supports two basic modes of operation: Buffered and Direct. The key difference is where data buffers are located. For Buffered mode, data buffers are hosted on the StoreEngine, and the source device "pushes" or writes data into these buffers. For Direct mode, the data buffers are located on the data source device (i.e. the FPGA or ADC or SBC), and the StoreEngine (or StorePak) "pulls" or reads data from these buffers. Each recording data stream type (FPGA, Ethernet, etc.) may support direct and/or buffered mode, depending on the source type. Note however that some recording source types support only one mode.

The demonstration system described later uses Buffered mode, where data is sent by the source devices through the PCIe backplane connections and is buffered in StoreEngine data buffers before being written to StorePak SSDs.

Recording Control Options

There are three methods of controlling recording and playback. The first is a web-based interface which is mainly used for configuration of the system but can also be used for run-time controls. The second, (which has the same capabilities) is the Recorder Network Control Protocol which is a client/server-based TCP socket interface. This control interface provides lower latency than the web-based method and can be built into an application that runs on a customer device. The third uses the Recording Driver hosted on a user's System Controller SBC, which implements a control interface over a PCIe link. The first two methods can control any non-System Controller source. The System Controller interface can control recording from the System Controller or from other sources connected to StoreEngine.

The table below contrasts the capabilities of these control options.

Control Type	Network	Recording Driver	Web
PC stream control	Limited	Yes	No
ADC stream control	Yes	Limited	Yes
FPGA stream control	Yes	Limited	Yes
UDP stream control	Yes	Yes	Yes
TCP stream control	Yes	Yes	Yes
EMU stream control	Yes	No	Yes
Recorder/LUN/mode controls	Yes	Limited	Yes

The System Controller is an optional PCIe linked SBC running either Linux or VxWorks. If a System Controller is used, a recorder driver is provided that is used to communicate with StoreEngine. Through this driver, an application can control recording from the System Controller itself or from an external source. As an intelligent device which is connected to StoreEngine and potentially to external sources,

the System Controller is positioned to manage all aspects of recording up to transferring data to/from storage. In the case of recording data directly from the System Controller, the System Controller directs the allocation of individual data recorder buffers and specifies when the data transfers to storage should occur. For external source recording, the System Controller directs the StoreEngine to start and stop the stream of data recorder buffers provided to the external source.

Recording File System

For all stream types, StoreEngine manages the recording of streams using user-defined constant-sized blocks of data, and using a specialized file system. The file system has a simple hierarchy with files organized into groups by LUN and channel. Each group of files is a file name space in which files are numbered sequentially as they are created. For each recording, three files are generated. The first contains the actual data blocks, the second contains file level metadata, and the third contains block level metadata, including timestamps, for each recorded block. All three files are readable through a standard file system interface which can be externally accessed using a network protocol (NFS, FTP or CIFS/SMB). This provides a method for retrieving recorded data outside of the recording source path.

Section 2: FPGA Recording Demonstration System

The demonstration system described here requires both significant PCIe connectivity as well as high aggregate recording rates. The system includes two StoreEngines, each equipped with a StorePak XMC, four FPGA boards, and a System Controller SBC. Each StoreEngine is connected to two FPGA boards and records data from them at an aggregate rate of 5 GB/sec using two Gen3 x4 PCIe interfaces. The System Controller SBC is connected to both StoreEngines through separate PCIe connections. It is used to provide high level controls to the StoreEngines and may also be a destination for playback of recorded data.

The StoreEngines and StorePak XMCs are Critical I/O products. Critical I/O also supplies the recording software running on the StoreEngines and the recording driver for the System Controller (is used) to facilitate communication with the StoreEngines. The FPGAs, System Controller and backplane are normally customer-supplied.

The system performance requirement is an aggregate recording rate (10 GB/s). This was analytically determined to be practical, and the principle goal of this recording demonstration was to demonstrate that this recording rate could be reliably achieved in a real system, given the overheads and inefficiencies of the real world.

Demonstration System Requirements Summary

The demonstration system has the following architecture and performance requirements:

1) The desired aggregate recording rate for each StoreEngine is 5 GB/sec for a system aggregate of 10 GB/sec. The desired aggregate playback rate is the same.
2) Each StoreEngine/StorePak XMC combo connects to two FPGAs for a total of four FPGAs. Each connection is a PCIe Gen3 x4 link.
3) The System Controller card connects to each of the StoreEngines using PCIe backplane connections. Each connection is a PCIe Gen2 x4 link.

4) The System Controller is to be able to control recording on each StoreEngine over the PCIe connections. These are high level controls such as “start recording” and “stop recording”.
5) The System Controller is to be able to playback FPGA-recorded data from each StoreEngine over the PCIe connection at an aggregate rate of 1.5 GB/sec.

Demonstration System Architecture

Figure 3 shows the demonstration system architecture. Each block in figure 3 is a separate VPX board. A single backplane connects all boards. Note that in this demonstration system, the StoreEngines operate independently of each other. That is, each maintains its own set of files and responds directly to System Controller commands. Note also that there is no direct communication between the two StoreEngines. Each pair of FPGA boards is connected to the StoreEngines’ PCIe Gen3 x4 ports. The optional System Controller SBC communicates with both StoreEngines using two PCIe G2 x4 ports.

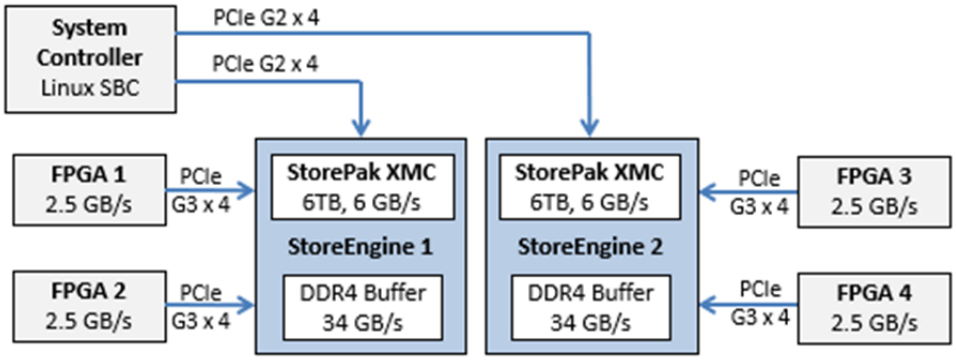

Figure 3: PCIe connectivity of demonstration system

The backplane for a fielded system like this would likely be a custom design. For this demonstration system, Meritec controlled impedance multi-lane backplane cables were used to connect the boards.

PCIe Topology Issues

The port used to connect StoreEngine to the System Controller is configured as an NTB. Since a System Controller will typically only have root ports an NTB is required so that the StoreEngine can be presented to the System Controller as an endpoint. StoreEngine and the System Controller are then able to map some or all of each other’s memory space into their own memory map. The memory mappings are done partly by the OS and partly by the recording software.

The other two StoreEngine ports are PCIe root ports and are connected to the FPGAs. The FPGAs thus appear as endpoint devices to the StoreEngines. StoreEngine accesses device memory and registers through these mappings. Each device can have up to three mapping windows, which can allow for separation of functionality if desired.

FPGA Recording Top Level Data Flow

The top level data flow for the demonstration system is diagramed in figure 4. The FPGAs for the system host a ring buffer of records that StoreEngine uses to submit a series of data transfer records to be used in subsequent data transfers. Each record includes a buffer address, buffer size, direction of transfer, a

logical channel number, a transfer handle and a status field. When the FPGA is done with a buffer it sets the status in the ring buffer record and notifies StoreEngine.

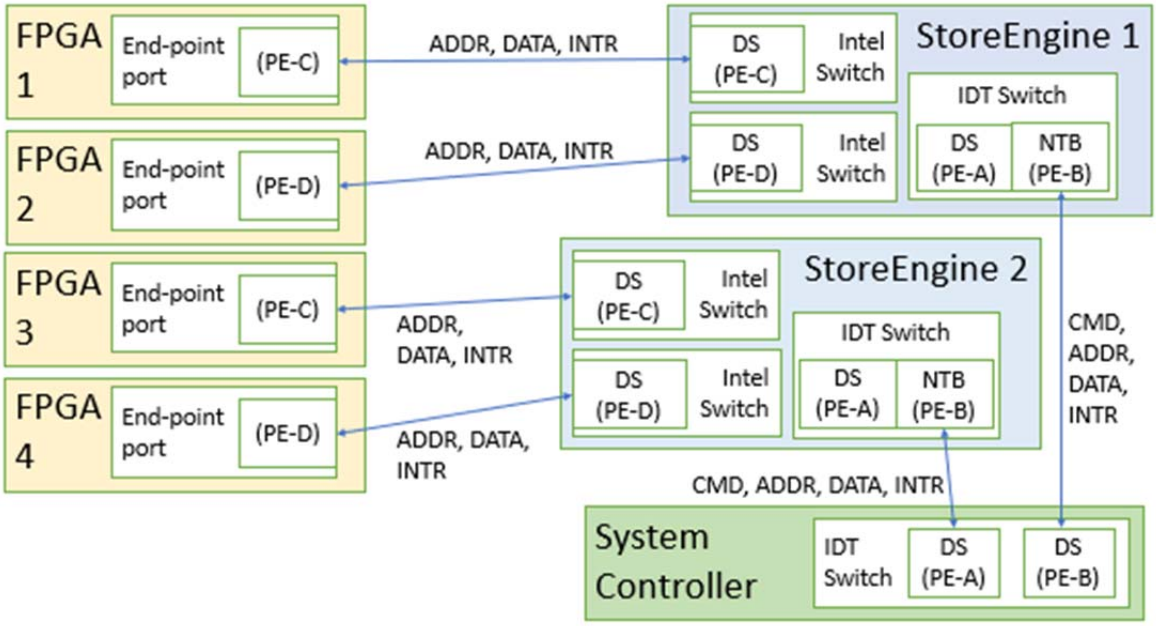

Figure 4. Top Level System Data Flow

FPGA PCIe and DMA Interface - Detailed Description

This section provides some additional details of the StoreEngine to FPGA logical interface, including both details of the FPGA hardware implementation, as well as a brief description of initialization and recording data and control flow.

The main logical components of the FPGA interface are: 1) a buffer address queue (BAQ), 2) a DMA engine to move received data to the data buffers, and 3) a method of interrupting StoreEngine when a buffer has been filled with data. Each FPGA has its own PCIe switch port, source buffers and DMA engine. Recorder software writes buffer addresses to the buffer address queue (BAQ) in FPGA memory. The FPGA fetches a buffer address from the BAQ and uses a local DMA to copy data from local source buffers to StoreEngine data recorder buffers. The StoreEngine is then interrupted as buffers are filled, at which point the StoreEngine schedules the buffers to be written to storage on the StorePak.

The StoreEngine to FPGA interface uses a single PCIe mapping window, through which StoreEngine does two things: 1) performs an initialization handshake to verify that the device is ready, and 2) writes data transfer request records into a ring buffer.

The initialization handshake is designed to verify that the FPGA is ready. When the FPGA comes up, it writes a token to a pre-defined offset in the mapping window that StoreEngine reads. StoreEngine then writes a start command to FPGA memory. The FPGA responds by writing another token, this time to a StoreEngine memory location, and interrupts StoreEngine. In the interrupt handler, StoreEngine verifies the token in memory and if valid, marks the FPGA as ready to be used.

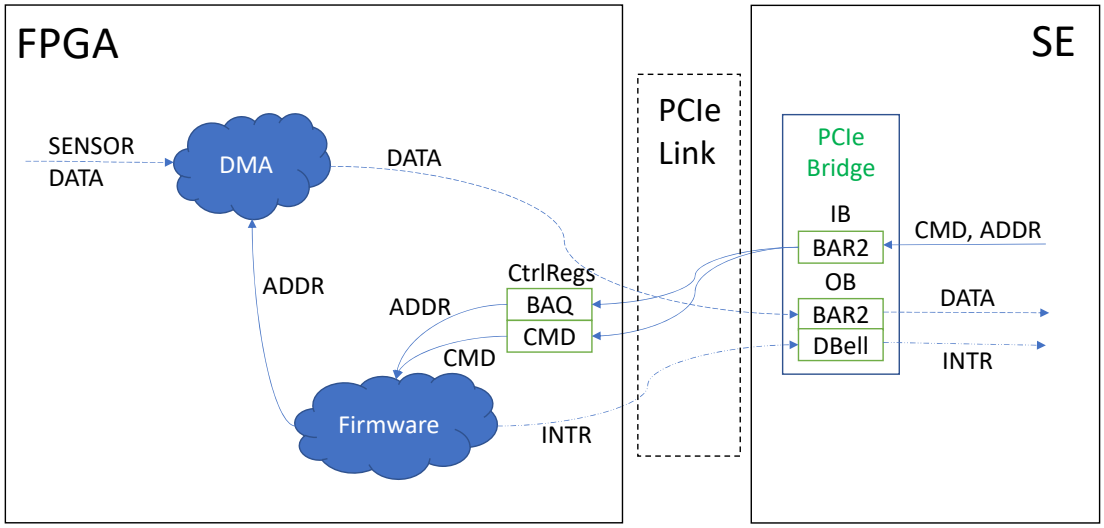

Figure 5. FPGA Data Flow Diagram

The data transfer request record from the StoreEngine provides the FPGA with both the information needed about a block transfer as well as the location for the FPGA to provide status when the transfer has completed. The record includes the buffer address, buffer size, direction and logical channel number. The buffer address and size define the PCIe address range of the data buffer. Direction specifies whether the operation is recording or playback. Logical channel number can be used to work on logically separate streams of data, each of which uses a different set of files; this demonstration uses a single channel. The record also includes fields for status and transfer length. The status indicates whether the transfer completed successfully. The transfer length (which is only relevant for recording) indicates the amount of data transferred into the data buffer; the FPGA does not need to fill the buffer completely. In the event it does not, StoreEngine will write a partially filled block of data to storage, but the block will still consume a full block of storage space. The amount of data in the block is tracked by StoreEngine using block metadata records. When playback is performed, the block metadata is used to determine how much data should be sent. The block metadata records also include the disk location, file position and time stamp information.

Note that StoreEngine FPGA source type software has flexibility in the FPGA interface designs that it can support; the interface used in the demonstration is only one possibility.

Note that for the demonstration system the FPGA hardware was actually emulated using SBCs running a software module (SE-FPGA) that is designed to exactly emulate the hardware functionality of a true FPGA. This software module first performs some basic interface initialization, then retrieves buffer addresses provided by the StoreEngine, fills the buffers with test data using DMAs to StoreEngine, and generates interrupts to the StoreEngine as the buffers are filled.

Recording Setup

The setup of the FPGA stream recording and playback functionality on the StoreEngine can be done using the StoreEngine's web based recorder management interface, an example of which is shown in Figure 6. The basic configuration steps including creating recorder LUNs (the basic storage container),

defining the PCIe topology, adding new recorder streams of the type FPGA, and then configuring those streams.

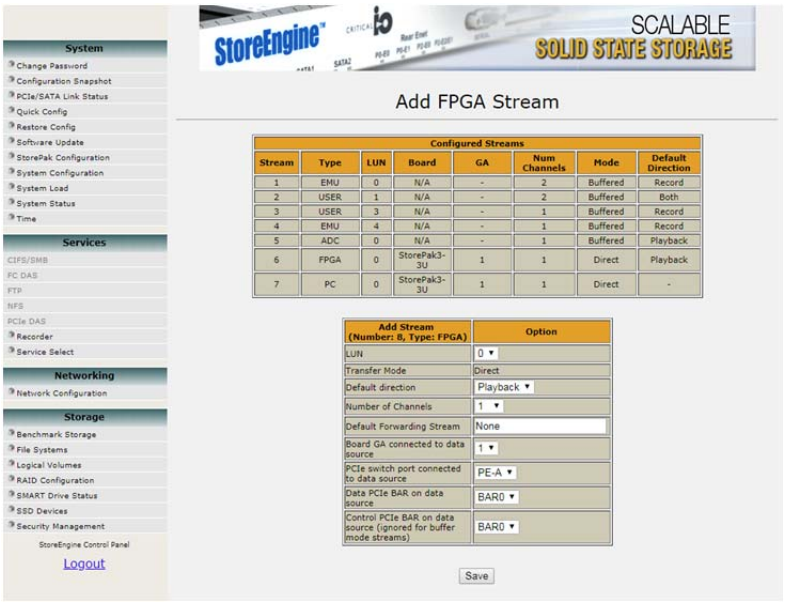

Figure 6. Example of StoreEngine Recorder Web Management Controls

Operation - Recording

StoreEngine recorder software first performs some basic initialization with the FPGAs to bring them online and verify they are ready to operate. After initialization, the FPGAs can begin writing data into StoreEngine hosted data buffers. The recorder software processes control commands received (such as Start and Stop) via the Recorder Network Control (RNC) interface, and provides buffers to the FPGAs through FPGA hosted buffer address queues. The FPGAs signal the StoreEngine when a buffer has been filled. The recorder software then manages writing the data in these buffers to SSD storage that is hosted on the StorePaks using the Critical I/O data recorder file system.

Operation - Playback

Playback of recorded data can be performed to either the FPGAs or the System Controller SBC. In each case the System Controller can be used to control playback but the software operation is significantly different.

Playback to FPGA

Playback operation to the FPGA is basically the reverse of recording operation. The main difference is the direction of data flow. Instead of being provided empty buffers that are returned to StoreEngine and written to storage, StoreEngine fills data recorder buffers by sequentially reading blocks of data from the specified file in storage. As buffers are filled, StoreEngine queues them with the FPGA using the same interface used for recording. The FPGA uses the direction field of the request record to determine that the operation is a send. When the FPGA has sent the data in the buffer, StoreEngine is notified by

interrupt and the buffer is reclaimed to be used again. StoreEngine queues multiple buffers with the FPGA so that there is always a data buffer available to be sent.

StoreEngine continues queuing buffers until a stop command is received or the end-of-file is reached. The start and stop commands can come from any of the three control interfaces, including the System Controller.

Playback to System Controller

Playback operation to the System Controller is controlled differently, through the System Controller interface. This is because StoreEngine provides buffer addresses in response to buffer request commands from the System Controller and, unlike the FPGA interface, there is no buffer address queue to push addresses into. Data buffers are only filled as they are requested by the System Controller. The completion of requests is asynchronous, so the System Controller can queue multiple requests for better efficiency.

When StoreEngine receives a buffer request from the System Controller, it fetches an address from a free buffer queue and schedules a read of the next block of the current file from storage. When the buffer has been filled StoreEngine completes the buffer request command by providing the buffer address and size in a command response to the System Controller. The System Controller then continues to request buffers at its discretion. When the System Controller is done with the buffer it sends a buffer complete command to StoreEngine with the buffer address.

Summary and System Performance Results

The StoreEngine / StorePak combination offers high PCIe connectivity and high recording bandwidth capabilities with up to 12TB of on board SSD storage. StoreEngine also features highly configurable high performance Recording Mode software. A demonstration FPGA recording system was constructed, tested, and benchmarked that consisted of two StoreEngine boards recording (and playing back) data from four FPGA boards, using StoreEngine's standard Recording Mode software.

Benchmarking results showed the two StoreEngines could reliably record and playback FPGA data at the desired 10GB/s aggregate sustained rate.