AN69211

Using High-Density Programmable FIFO In Video And Imaging Applications

Author: M. Sivashankar
Associated Project: No
Associated Part Family: CY7Fxxxx

AN69211 explains how to use Cypress high-density programmable FIFOs in video and imaging applications and the benefits offered over conventional FPGA and DDR SDRAM solutions. This information is intended for engineers who work on video and imaging applications and are familiar with FPGA and video standards.

Introduction

Video frame storage and synchronization are a part of almost any video and imaging application. Because the nature of video data is sequential, high-density FIFOs (HDFIFOs) are best suited for these applications.

This application note describes a few video applications to explain the data path and data handling required. It also compares implementing a frame buffer using Cypress HDFIFO (CY7FXXXX) to the conventional method of using an FPGA and DDR SDRAM, along with memory size and bandwidth calculations. Advantages of using Cypress HDFIFO over the conventional solution are described.

Overview of Video Applications

Figure 1 shows the system block diagram of an IPTV. The input transport streams in any encoded format such as DVB–ASI, MPEG2, or SDI are passed through a multi-format codec to be transcoded into an H.264 transport stream. The encoded transport stream is encapsulated with channel information and sent over Ethernet. On the receiving path, the incoming transport stream is decoded. Post-processing such as noise reduction, color enhancement, scaling, and deinterlacing occurs before display.

Figure 1. Block Diagram of an IPTV

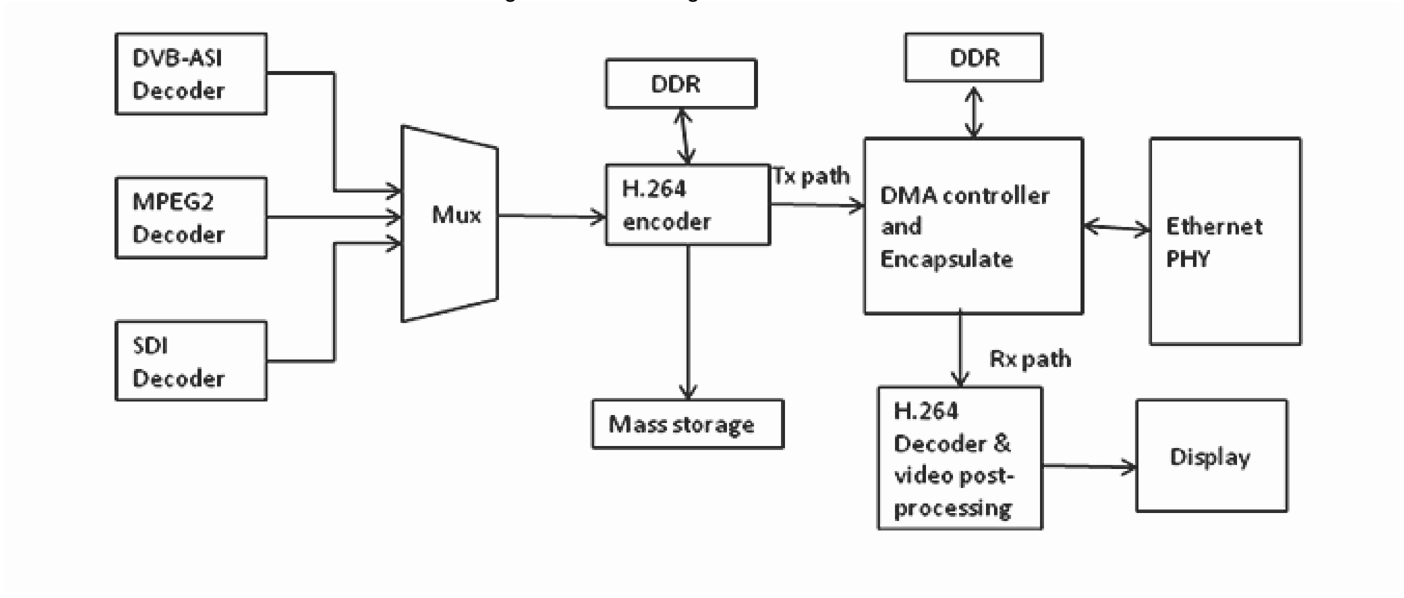


Figure 2 shows a system block diagram of a professional high-definition (HD) camera used in filmmaking. The image captured is passed through an image-processing unit. This unit is usually an FPGA-based design because image processing is typically a proprietary process with frequent changes.

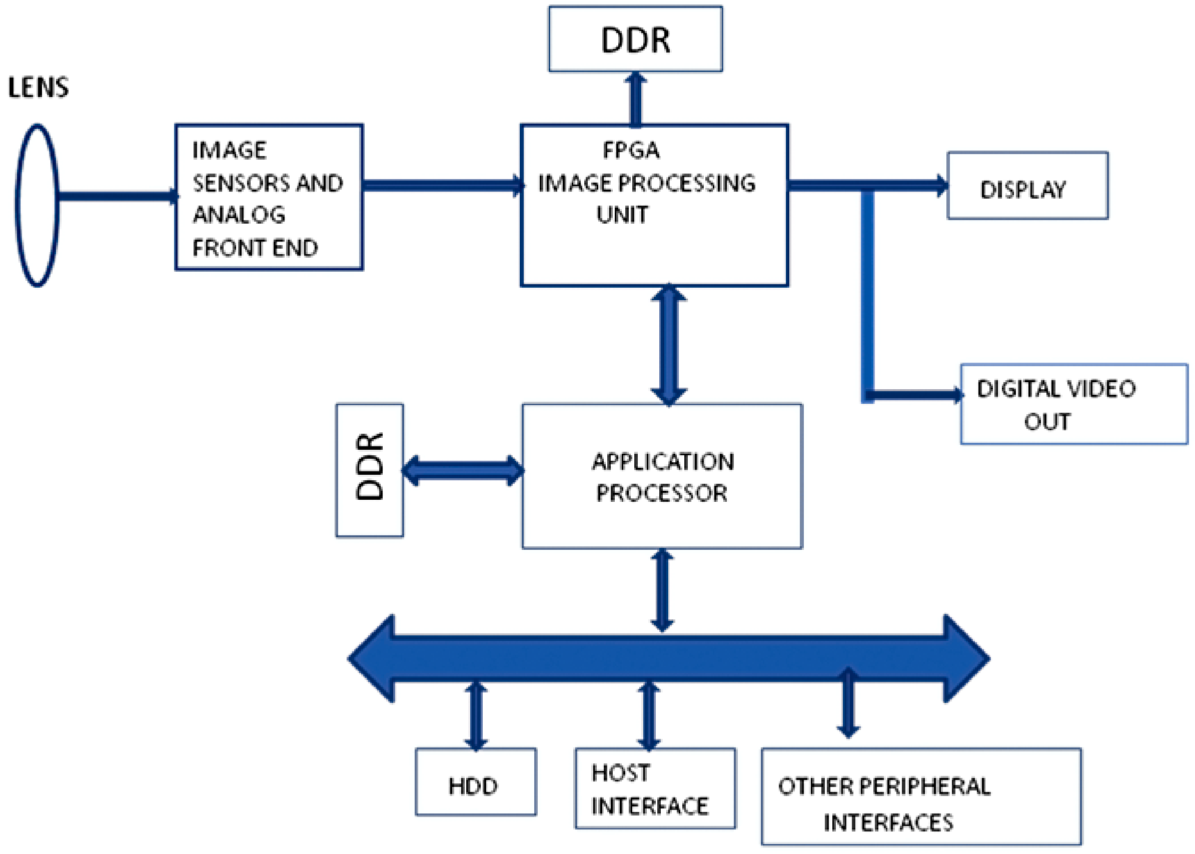
The application processor manages communication with other equipment, and it compresses and stores the captured content onto mass storage. The application processor also contains a graphics engine to generate the on-screen display (OSD) that is blended with incoming video to be displayed.

Two types of data handling are involved:

- **Frame synchronization:** Required in situations such as transmission and reception over Ethernet in which the bit-rates vary. The decoder needs a constant bit-rate transport stream. Though the memory required for synchronization may seem small, it can be significant when multiple streams are involved. An asynchronous FIFO achieves the synchronization.
- **Frame storage:** Required wherever any temporal processing such as frame rate conversion, trans-coding, or deinterlacing occurs. The number of frames to be stored increases with the amount of temporal information required. Because the video data is sequential in nature, the frame buffer must be essentially a FIFO.

The previous discussion shows that all of the storage and synchronization can be achieved using FIFOs. For an approximation of the size of these FIFOs, a typical 1080p frame in 10-bit 4:2:2 format would require a memory size of (1920*1080*20) = 39.55 Mbit. You can estimate the total size by multiplying this figure by the number of frames to be stored. Typical video processing algorithms require 2 to 3 frames to be stored, which means the total size can be up to 120 Mbit. Because it is not possible to have such large on-chip SRAM-based FIFO memory, the general approach is to use a DRAM to buffer this data.

Figure 2. Block Diagram of an HD Camera


DDR
LENS
IMAGE SENSORS AND ANALOG FRONT END
FPGA IMAGE PROCESSING UNIT
DISPLAY
DIGITAL VIDEO OUT
DDR
APPLICATION PROCESSOR
HDD
HOST INTERFACE
OTHER PERIPHERAL INTERFACES

`

Conventional Frame Buffer Implementation

Frame buffers are actually just high-density FIFOs. They are usually implemented by using external DDR SDRAMs.

Figure 3 shows the data path for a typical scenario in which four video streams from different sources are to be displayed on a single display. Four high-definition cameras capturing video with 1080p60 (24-bit RGB) resolution are connected to the system using a LVDS (cameralink) interface. After color space conversion (from RGB to YCbCr) and chroma downsampling (4:4:4 to 4:2:2), the frames are downscaled by a ratio of two both horizontally and vertically and stored in the DDR2 SDRAM memory. The stored frames are read back and positioned according to what is required. The resulting frame with merged frames is then upsampled and color space converted to drive the panel with an LVDS link.

Size Requirement

To avoid tearing effects, no temporal processing is involved. Two frames of each source are stored so that when one frame is being written, the other can be read back. The size of two frames is ((1920*1080*16)/4)*2 ~= 63.3Mbit.

Bandwidth Requirement

Because the read and write path is multiplexed, the bandwidth required is the sum of write and read path bandwidth. Write path frequency is:

(frequency of each client)*(number of clients)
which is (148.5/4)*4 =148.5 MHz.

Read path frequency is equal to the output frame resolution frequency which is equal to 148.5MHz. Actual operating frequency is:

((read frequency + write frequency) / 2 + overhead)

The interface is at double-data rate and there are overheads such as DRAM memory refresh cycles and bank address switching. Assuming 80% efficiency, the frequency of operation would be around 185MHz.

Memory Interface Size and I/O Requirements

Because the frames are stored in 16-bit 4:2:2 format, a 16-bit interface is sufficient. The total number of I/Os required from the FPGA would be 46 pins:

Clock pins (2 for differential clock, 1 for clock enable) = 3 pins

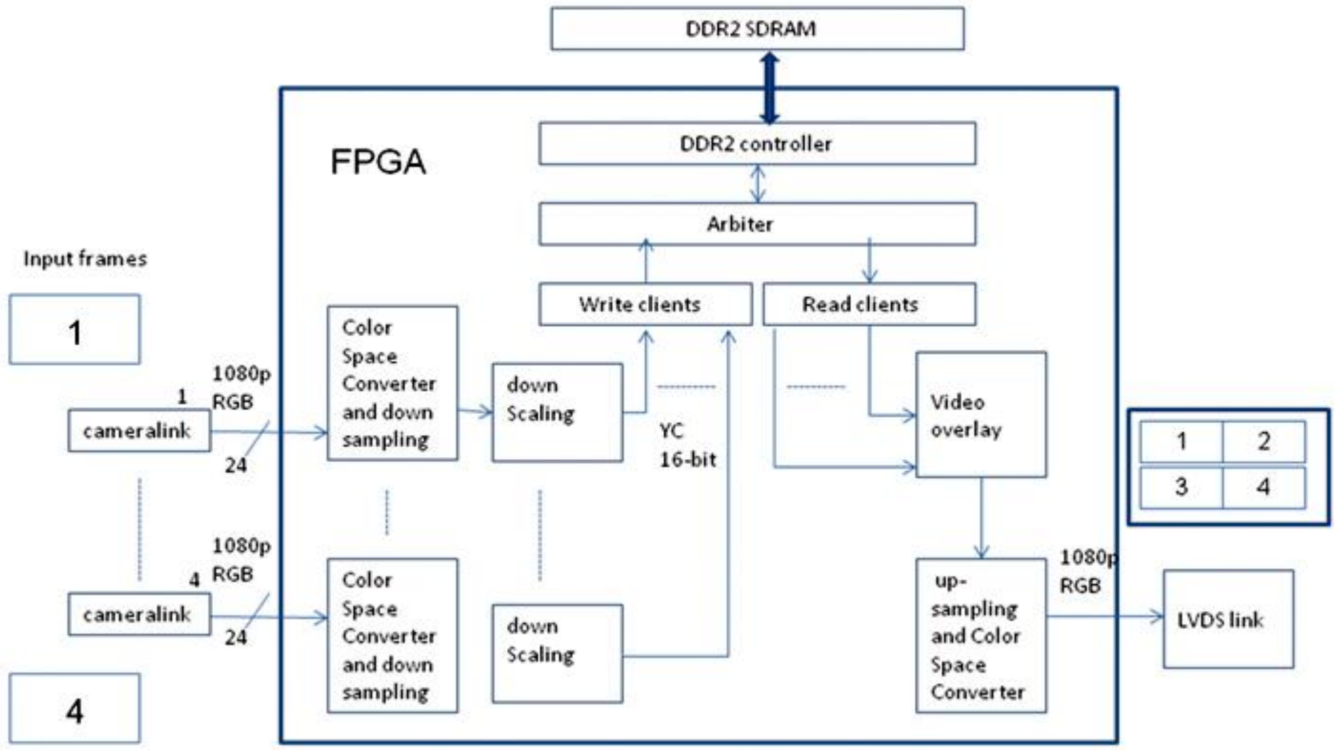
Command pins (chip select, RAS, CAS, WE) = 4 pins

Address pins (14 address lines, 3 bank address lines) =17 pins

Data lines (X16 interface) = 16 pins

Data strobe and mask (4 pins for 2 differential DQS, 2 pins for data mask) = 6 pins.

Figure 3. Data Path With Multiple Video Streams



Frame Buffer Implementation Using HDFIFO

Figure 4 shows the example application with Cypress HDFIFO instead of the DDR2 SDRAM.

Size Requirement

The size storage is the same as that of the DDR2 SDRAM memory, which is equal to two frames: ((1920*1080*16)/4)*2 ~= 63.3Mbit.

Bandwidth Requirement

Because the read and write path is separate, the operating frequency of read and write can be different. This is a major advantage over the DDR2 SRAM memory. Write path frequency equals (frequency of each client)*(number of clients) which is (148.5/4)*4 =148.5MHz. Read path frequency is equal to the output frame resolution frequency, which is equal to 148.5 MHz. Because no overhead (such as memory refresh cycles and memory bank switching latencies) is involved, actual operating frequency would be 148.5 MHz single data rate for both read and write path. Cypress HDFIFO has a 36-bit wide read and write port, which can further lower the operating frequency to 74.25 MHz.

Memory Interface Size and I/O Requirements

Because frames are stored in a 16-bit 4:2:2 format, a 16-bit interface is sufficient. The total number of I/Os required from the FPGA would be 48 pins:

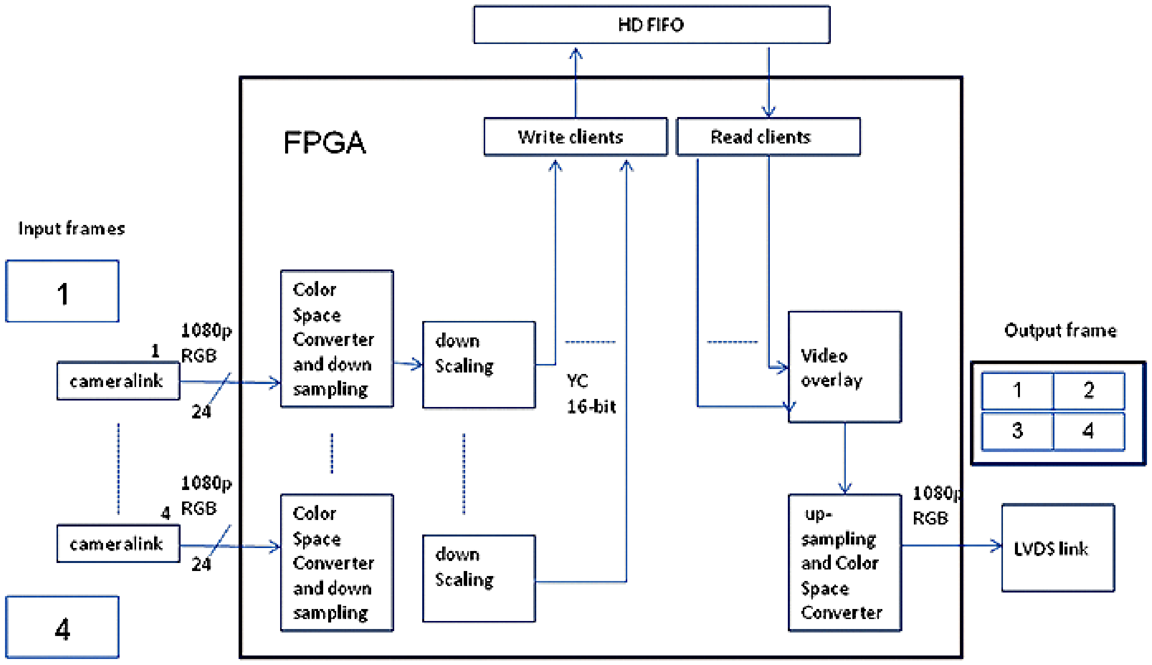
Clock pins (1 pin for write clock, 1 pin for read clock) = 2 pins

Command pins (write enable, read enable, input enable, output enable, 3 pins to select which of the 8-queue to write, 3 pins to select which of the 8-queue to read, 1 pin for mark, 1 pin for retransmit) = 12 pins

Data pins (16 pins for write data, 16 pins for read data) = 32 pins

Flags (1 pin for empty flag, 1 pin for full flag) = 2 pins

Figure 4. Data Path Using HDFIFO With Multiple Video Streams



HDFIFO Advantages

HDFIFO has several types of advantages—architectural, electrical, and economical—over conventional implementations.

Architectural Advantages

Because the read and write paths are separate with no operating overhead, the operating frequency is reduced by more than half, which gives a significant advantage. The FPGA internal logic becomes simpler because the SDRAM controller and arbiter are eliminated. The signal switching frequency is reduced by more than half, which allows increased setup time margins and relaxed clock-to-output constraints when compared to a DDR2 interface. The number of clock domains in the design is smaller, which reduces handoff and cross-clock domain-related timing issues.

Electrical Advantages

A lower signal-switching frequency reduces the amount of switching noise on the board. The IO logic with an HDFIFO can be any LVCMOS interface that has more noise margin compared to the SSTL2 logic of DDR2 SDRAM.

Cost Savings

The FPGA resources on a high-end FPGA solution that can be saved by using an HDFIFO are a result of the following features:

- SDRAM Controller, which reduces the required memory, I/Os, and logic
- Video processing features that can be implemented using the multi-queue feature on the HDFIFO, such as: interlacing /deinterlacing of video signal, PIP implementation, and processing interweaved signals.

Summary

This application note shows that using Cypress HDFIFO results in reduction of operating frequency to half as compared to FPGA and DDR SDRAM solutions. Cypress HDFIFO also offers better noise margin, reduced design complexity, and more efficient data handling when compared to the conventional solution.

Document History

Document Title: Using High-Density Programmable FIFO in Video and Imaging Applications - AN69211

Document Number: 001-69211

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	3243245	SIVS	04/28/2011	New Application Note
*A	4370520	ADMU	05/06/2014	Updated template.

Worldwide Sales and Design Support

Cypress maintains a worldwide network of offices, solution centers, manufacturer's representatives, and distributors. To find the office closest to you, visit us at Cypress Locations.

Products

Automotive	cypress.com/go/automotive
Clocks & Buffers	cypress.com/go/clocks
Interface	cypress.com/go/interface
Lighting & Power Control	cypress.com/go/powerpsoc cypress.com/go/plc
Memory	cypress.com/go/memory
PSoC	cypress.com/go/psoc
Touch Sensing	cypress.com/go/touch
USB Controllers	cypress.com/go/usb
Wireless/RF	cypress.com/go/wireless

PSoC® Solutions

psoc.cypress.com/solutions

PSoC 1 | PSoC 3 | PSoC 4 | PSoC 5LP

Cypress Developer Community

Community | Forums | Blogs | Video | Training

Technical Support

cypress.com/go/support

Cypress Semiconductor 198 Champion Court San Jose, CA 95134-1709	Phone Fax Website	: 408-943-2600 : 408-943-4730 : www.cypress.com