

Implementation of high speed
streaming video data transfer
application on FPGA

Kaushal D. Buch



Implementation of high speed streaming video data transfer application on FPGA

Abstract

High speed streaming video data transfer operations are evolving with the onset of an era of high-end multimedia applications in compact digital devices. These data transfer applications become more crucial in high resolution video imaging. This article talks about System-on-Chip architecture for storing and transferring high speed data from image sensor to USB. The design is implemented on low power FPGA device. The critical design issues with compact digital devices intended to be used on the move are low power consumption and smaller device size. This paper highlights a FPGA based SoC architecture for a high resolution streaming video capture device which takes care of low power and high speed design issues.

Introduction

The system under consideration is a high resolution camera with a CCD image sensor as the input and USB port as the user interface/output. The application can be used in variety of streaming media applications like mobile camera to USB based web-camera, video conferencing.



The system consists of hardware-software co-design, which comprises mainly of a FPGA for video data transfer, a host microcontroller for controlling the entire data transfer operation, SDRAM for intermediate storage of this video data and USB Controller chip for implementation of USB PHY. The overall product is a having a layered architecture in which the data transfer through FPGA is governed through the microcontroller. The microcontroller is also responsible for controlling other devices on-board and also for programming USB interface etc. This is achieved via applications written in software and drivers for USB interface. Thus it is a application that involves both – FPGA development, Embedded Software development and board design.

The video processing board along with the CMOS imaging sensor is interfaced to the personal computer through USB port. The FPGA operations like data capture from the imaging sensor, storing the data into the memory and retrieving it back to transfer it to the USB interface etc. are controlled by the host microcontroller. The host application running on the PC is used to control the imaging options. An additional I2C interface with the imaging sensor is provided for transferring the commands and data to and from the imaging processing chip. The focus of this paper shall be on the FPGA implementation of streaming video application.

System-on-Chip Design:

Architecture

The Actel FPGA based SoC consists of Video Buffer and Frame Controller, DMA Controller, SDRAM Controller, USB Interface Controller, I2C Controller and Host Interface. The system level block diagram is as shown in figure 2. The streaming data input is at 108 MHz, 8-bit at 30 frames per second. The internal architecture of the SoC works at 60MHz 16-bit. The host microcontroller interface works at 25MHz.

Buffering of the video data is done in Video Buffer and Frame Controller block using 8KB of Block RAM. The data is transferred to SDRAM through Channel 0 of DMA Controller. After one frame is transferred to the SDRAM, the host microcontroller halts the channel 0 transfer and starts the data transfer between SDRAM and USB via channel 1. The data rate available on this data path is about 960 mbps. There is a 1KB FIFO in the USB Interface Controller block for continuous transfer to the USB chip. After one transaction on channel 1, channel 0 is re-activated and thus data transfer to the SDRAM memory resumes. The SDRAM has a depth of 16MB, which can store approximately 10 frames (1024 x 768 @ 30 fps). Thus, reading frame by frame from memory, in a chunk of 512 bytes or 1KB, helps in maintaining balance of both performance and avoids frame dropping. The data rate in the bulk mode of USB transfer is about 150 mbps.

The unique implementation aspect of this design in the ability of the Frame Controller to detect the pixels per line and lines per frame of the incoming streaming video and also the dual channel DMA controller which is able to detect and transfer line by line data into the memory under the control of microcontroller. These features make the hardware more versatile and also help in better hold and flexibility of data transfer with the microcontroller.

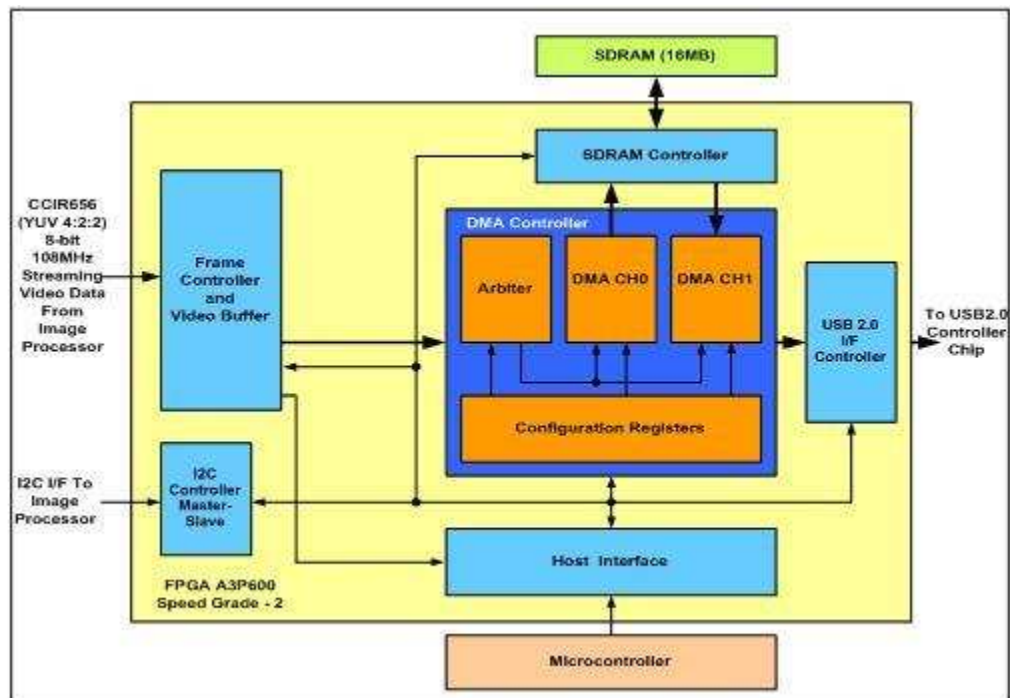


Figure 2: Block Level System for Streaming Video Data Transfer Application

Video Buffer and Frame Controller

The basic task of this module is to buffer the data. It has 8KBytes of Block RAM storage which can store about 4 lines of data for 1024 x 768 resolution. Streaming Video input is provided to this module which stores the data in asynchronous FIFO (FPGA block RAM in this case) at video clock rate. The buffering requirement is set according to the data rates available at the video interface and at the DMA interface. With the storage capacity of about 8192 bytes (4 lines of data), there is almost no back pressure at the video interface section. The interface possesses Hsync, Vsync, video data and video clock as per ITU-R BT.656-4 specifications.

This module also provides the pixel count and line count to the DMA Controller in terms of pixels per line and lines per frame. Frame Controller can work in normal as well as Embedded Hsync/Vsync mode as per ITU-T BT.656-4 specifications.

Dual Channel DMA Controller

The Dual Channel DMA Controller is the key module for optimizing data transfer performance. It has 2 channels – Channel 0 and Channel 1. Channel 0 is the default activated channel and starts after one-line buffer of video data in the Video Buffer and Frame Controller module. It is used to transfer line by line data from the Video Buffer and Frame Controller module to SDRAM Controller. This channel can be stopped through the configuration register under the control of the host microcontroller.

To ensure higher speed data rate availability, the DMA Controller operates as follows. Channel 1 is activated by the host microcontroller. The host can start the SDRAM to USB data transfer looking at the number of lines written in the SDRAM. Channel 0 transfer is halted before starting the operation through Channel 1. This is done by checking the status of lines written – DMA Controller module has a register that tracks the lines written in to the SDRAM. Due to a single data path between the SDRAM and DMA, this operation has to be judiciously done to achieve optimum performance. The DMA module is generic and accepts the pixels per line and lines per frame information from the Video and Frame controller module. This is a major advantage against conventional DMA controllers wherein the DMA has to be configured by the host processor for the amount of data transfer required.

Dual channel mechanism and its effective utilization by the host microcontroller are the prime factors leading to higher data transfer rates. Also due to pixel and line count data available, it frees the controller from configuration of count details.

SDRAM Controller

SDRAM Controller module enables data transfer between DMA Controller channels and SDRAM memory chip. The controller possesses logic for implementation of SDRAM Read & Write protocol and supports incremental, burst and interleaved-burst modes.

USB Interface Controller

USB Interface Controller forms an interface conversion between the custom interface of DMA and the USB controller chip external to the FPGA. It is also used to program the external chip through the microcontroller via the host interface.

I2C Controller

I2C bus is required to program the image processor chip via the microcontroller. The module can work as a master or slave as configured. Also it has a facility to read or write a burst data of 256 bytes, by using FIFO in the module.

Host Interface

The module interfaces with the microcontroller outside the FPGA. The module has the register bank for configuration of modules and decodes the addresses as per defined register map. The other functions include reflection of status, configuration of commands and interrupt control.

Resource Utilization and Power Consumption

The resource utilization in case of any FPGA can be considered on the basis of the total percentage of the available resources. In case of the FPGA used in the current implementation we use almost all the different hard macros available. Here, core utilization is the basic logic blocks utilized. These logic blocks contain function generators and sequential elements for implementation of logic. PLL (Phase Lock Loop) is used in order to provide phase shifted clocks to the SDRAM controller module. The buffering on input video data and output data buffering has been done using the internal block memory modules. The RTL was implemented on Actel ProASIC A3P600 -2 FG 256 and the resource utilization and power consumption statistics is as below:

Resource	Utilization	Available	Percentage
Core	9367	13824	67.76
I/O	121	165	73.33
PLL	1	6	16.67
RAM/FIFO	19	24	79.17

Type	Power
Static	7.5 mW
Dynamic	1022 mW (approx)

Note: Synthesis and Power Calculation is done using Actel Libero 7.0 and Smart Power.

The power consumption of any FPGA is classified in two categories - static and dynamic. The static consumption is the consumption in the idle state when the FPGA is not operational functionally. The dynamic power consumption depends on the clock frequency and is the overall power consumed by the FPGA during normal operating conditions. The dynamic power consumption is dependent on hardware architecture and other device parameters apart from clock frequency.

Conclusion

The architecture provides efficient technique for transferring high speed streaming video data. The implementation on Actel's Flash based FPGA helps in low power consumption and elimination of reconfiguration of FPGA on power-up. The low power consumption makes it an ideal architecture for bus-powered applications. The arbitration of the DMA channels helps in maintaining a balance between buffering and the transfer rates. It also helps in reducing the back pressure on the input video buffer. Thus, the architecture suffices the requirements of a low power streaming video data transfer applications.