International Journal of Technical Innovation in
Modern Engineering & Science (IJTIMES)
Impact Factor: 3.45 (SJIF-2015), e-ISSN: 2455-2585
Volume 4, Issue 5, May-2018

LOW COST FPGA DEVICE SELECTION FOR REAL TIME VIDEO SIGNAL PROCESSING

Dr. R. Prakash Rao
Associate Professor, Electronics and Communication Engineering,
Matrusri Engineering College, #16-1-486, Saidabad, Hyderabad-500059, India.
prakashiits@gmail.com

ABSTRACT

This work depicts the utilization of present day low price Field Programmable Gate Arrays (FPGAs) for real time broadcast video processing. Capabilities of selected device family (Altera Cyclone IV) are discussed with regard to video processing. Example IP core deinterlacer is designed in Verilog HDL and the design flow is described. The IP core is implemented in real hardware system. The overall hardware system is described together with individual FPGA components providing video input/output and other I/O functions.

Keywords: FPGA, video processing, deinterlacing, Verilog HDL, hardware design flow

I.INTRODUCTION

Nowadays, as the requirements for processing power of the embedded systems are growing, many systems are starting to use FPGAs for offloading the processing functions. This was made possible by the advancements in chip manufacturing technology as described by Moore's law [1], where programmable logic device parameters such as density, processing power, power consumption and cost improved to become viable alternatives to the traditional approaches. Additionally, a design using programmable logic offers specific advantages over other approaches, mainly the possibility to alter the configuration of the hardware in the field (hence the name), which is a very useful feature considering problems like bug fixes and frequent needs to modify the design after the product is finished. Of course, this flexibility comes at a premium compared to a dedicated 'hardened' CPU or ASIC, usually both in terms of power consumption and unit price. However, especially for small production series, the flexibility of programmable logic may more than balance the additional cost of the device; the CPU may not be exactly suited to the application and the ASIC development costs may be well out of bounds of the estimated product volumes. With the gradual transition of video signal representation from analog signals like VGA and SCART to the digital domain, programmable logic started to provide the processing functions where required. With its inherently parallel nature, these devices are well suited for algorithms requiring high bandwidth and the calculation of many operations in parallel on the video data.

II.REVIEW OF EXISTING SYSTEM:

2.1 Present day FPGAs

Nowadays, FPGAs are a standard off-the-shelf components, ranging in size and capabilities. Usually, the FPGA is composed of configurable logic, routing resources, embedded memory, multipliers and a range of hardened peripheral interfaces. Not physically present in the FPGA, but from the design standpoint an integral part of the device design flow, is the FPGA development software.

2.1.1 Programmable logic

The programmable logic is composed of LUTs (look up tables, sometimes also called LEs - logic elements), which are SRAM based cells performing user defined function given by the FPGA configuration bitstream. The exact LUT structure varies by manufacturer and device family, for illustration above figure 1 shows the LUT structure of Altera Cyclone IV device family.

.

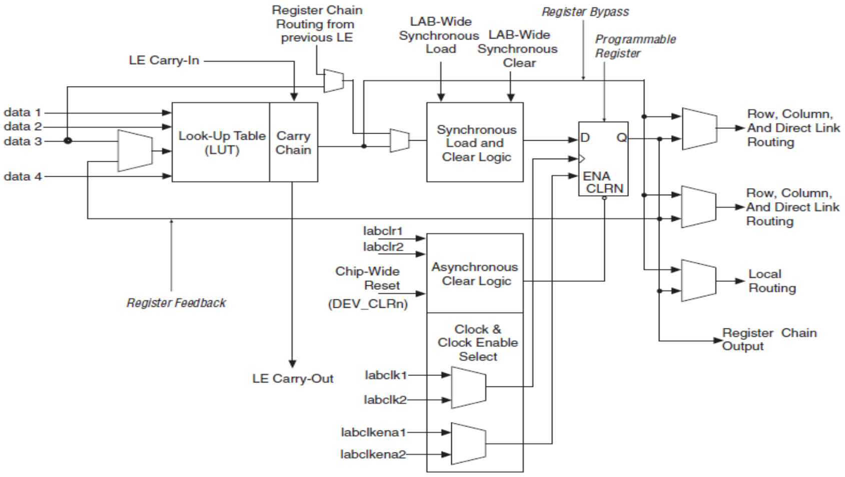
Register Chain Routing from previous LE
LAB-Wide Synchronous Load
LAB-Wide Synchronous Clear
Register Bypass
Programmable Register
LE Carry-In
data 1
data 2
data 3
data 4
Look-Up Table (LUT)
Carry Chain
Synchronous Load and Clear Logic
D
Q
ENA
CLRN
Row, Column, And Direct Link Routing
Row, Column, And Direct Link Routing
Local Routing
Register Chain Output
labclr1
labclr2
Chip-Wide Reset (DEV_CLRn)
Asynchronous Clear Logic
Register Feedback
Clock & Clock Enable Select
LE Carry-Out
labclk1
labclk2
labclkena1
labclkena2

Figure 1: Altera Cyclone IV LUT structure

2.1.2 Routing resources

To enable connections between logic elements themselves and between logic elements and any other parts of the chip, the FPGA contains the interconnect. These are ‘hardened’ connection paths inside the chip, either general purpose for user design or with a specific function, e.g. clock distribution networks. The clock distribution paths are designed in such a way as to provide uniform clock distribution with minimal skew over all parts of the chip. This is an important part of the interconnect fabric, since most FPGA designs are synchronous and the quality of clock distribution directly affects the maximum frequency at which the user design can properly function (this maximum frequency is usually called fmax). Also, this is the part of an FPGA occupying the most silicon resources of the chip. Some estimates quote up to 90% of the silicon die is dedicated to routing[2].

2.1.3 Embedded memory

Many of the FPGA designs require some kind of fast memory for temporary storage of intermediate results, data buffers and other. For this reason, the chip contains embedded memory blocks. These are hardened SRAM memory units, usually configurable for different memory sizes, data widths or single/dual port access.

2.1.4 Embedded multipliers

Since FPGAs are well suited for digital signal processing (DSP), most device families contain hardened multipliers. This provides the designer with optimized blocks with higher performance (fmax) than soft (in logic) implementations and also frees up logic resources, which would otherwise be needed to implement the multiplier function. The DSP blocks are usually fixed point, newer and high end FPGA families implement hardened floating-point-optimized components[3].

2.1.5 Development software

An important part of the FPGA development is the design software. This software package provides the designer with interface to all FPGA design stages, from design entry to programming of the configuration memory. This software is responsible for transferring the user design to a selected physical device and its structure while meeting the user requirements for design timing (timing constraints). Contrary to the software world, where the compilation times are relatively small and the iterative development method cycle is short, a larger FPGA design can take several hours to compile. The compiler must analyze the design, convert the algorithms into device-specific blocks and fit the resulting netlist into the selected device fabric. When the design uses a large portion of the device resources or has high requirements for maximum frequency, this is a computing challenge even for modern processors (Intel Sandy Bridge CPU i7-2600K@3.4GHz compiles the design described in this work in 9 minutes, although on a single core; and this is a relatively small design). For this reason, appropriate hardware is necessary for the development.

2.2 Future possibilities of FPGAs

Currently, the fastest performing FPGA is probably from the device family Speedster22 from Achronix[4]. Since the major performance limiting factor in current FPGAs is the interconnect, the Achronix device avoids this bottleneck by time multiplexing the routing resources. By doing this, the Speedster22i device is capable of providing 1.5GHz peak processing performance. Since today's highend is tomorrow's lowend in the semiconductor industry, we may see a rapid increase in processing power of even low cost FPGAs in the coming years. The discovery of memristor[9] may be an important step towards developing new generation FPGAs. HP is currently developing a memristor based FPGA. The standard PC architecture may also include elements of FPGA fabric in the future or be entirely replaced by programmable logic. This is signified by the Intel Stellarton CPU, which includes an Intel Atom processor together with an Altera Arria II FPGA die in a single package. The FPGA is currently used as an H264 encoding accelerator.

2.3 Video processing on an FPGA

Processing a video stream usually involves operations on either the video signal timing or on the raw bitmap data of individual frames or fields. The FPGA architecture is well suited for video processing for the following reasons:

- Video timing generation is relatively straightforward with an FPGA. Even the logic fabric of low cost FPGA families is usually capable of supporting 150+ MHz IP components, therefore allowing generation of HD resolutions.
- Processing the raw frame data can take advantage of the hardened DSP blocks to ease the timing requirements for the logic fabric itself. Together with pipelining the individual algorithm operations, this allows the design of complex video processing paths even with HD resolutions.
- By being "close to metal", the algorithms on an FPGA can be more effective in terms of power than systems using an CPU core to perform the processing functions.
- Due to the FPGA flexibility, the video processing path can be tailored to specific project requirements.
- The flexibility of the FPGA architecture may prove useful for small production series, where the development costs of ASIC solution may be prohibitive. For these reasons, the processing functions required for the project described in this work were implemented on an FPGA.

III Broadcast video transport standards

Today, with a few exceptions (e.g. the VGA interface), the video signal representation transitioned from analog to digital domain. The most obvious advantage of digital representation over analog is that the video data is not in any way altered by the transmission. With analog representation, this was not possible due to effects like noise and line losses, which in most cases corrupted the transmitted information. Regardless of the selected video interface standard, video data is divided into discrete images called frames. A frame is an bitmap image, transferred over the transport interface from top to bottom line by line, with each image line being transmitted from left to right. Therefore, the transmission of a frame starts with top left pixel and ends with bottom right pixel. The rate at which the video frames are transferred is called a frame rate. The video format can be either progressive or interlaced. In progressive video stream, a frame is transferred in whole, meaning it is a complete representation of the video image in one point in time. With interlaced stream, frames are divided in halves called fields. Fields can be either odd or even, where odd field contains odd lines of the frame and even field contains even lines. When the stream is transferred as interlaced video, the motion appears smoother because this format effectively doubles temporal resolution of the stream (compared to a progressive stream with the same resolution and bandwidth). The video data represent the scene in some predefined color space. The most commonly used color spaces are RGB and YCbCr. With RGB color space, the pixel has red, green and blue component to identify it's color. The RGB standard is widely used in the PC industry for video data representation and as graphics card output format. When using YCbCr color space, the pixel has luminance (brightness) and chrominance (color) coordinates to identify the color. Conversion between these color spaces can be from straightforward to fairly complex, depending on the requested conversion quality. The horizontal and vertical resolution of the frame, frame rate, color space and progressive/interlaced identifier together form a video format. Video formats are standardized by organizations such as VESA[16] or SMPTE[5]. This chapter gives an overview of video transport standards used for video input and output of the presented video processing system.

3.1 Parallel digital data

The representation of video data as a parallel clocked bus is most common when connecting different integrated circuits on a printed circuit board. The bus contains a master clock signal, horizontal and vertical synchronization signals, active picture indicator (data valid signal) field identifier for interlaced formats and the video data itself. This format with separate horizontal and vertical synchronization is most commonly used, probably for its universality. Although embedded synchronization can be used (synchronization signals are not separate wires but are embedded as special sequences directly in the video data), this may cause design complications when using video processing ICs which each expect differing embedded synchronization sequences because of differing standards (e.g. BT656 vs BT1120). The parallel transmission format requires that the appropriate individual bit wires have their lengths closely matched to each other to ensure that the pixel wavefront is properly aligned at the receiver side. With today’s high resolutions and therefore high pixel clock rates, this data format may also cause problems with signal crosstalk or reflections from impedance differences, therefore it is a good practice to use some kind of termination at both the transmitter and receiver sides.

3.2 Serial digital interface (SDI)

Serial digital interface[18] is a video transport standard used mainly in broadcast and medical industries. It uses shielded coaxial cable as a medium and allows for transfer rates from 270Mbit/s to 3Gbit/s. It can be thought of as an serial encapsulation of parallel digital data. On the transmitting side, the data is serialized to a high speed serial form and on the receiving side data is deserialized back to parallel format. SDI uses NRZI encoding scheme to encode data and a linear feedback shift register to scramble the data to control bit disparity. The video stream can also include CRC (Cyclic Redundancy Check) checksums to verify that the transmission occurred without an error.

3.3 Digital Video Interface (DVI)

DVI is an interface to transfer digital video and is used frequently in the PC industry. The interface uses TMDS (Transition Minimized Differential Signaling) to transfer data over four twisted pairs (three for data and one for clock) of wires. Because this interface is frequently used to connect a graphics port of a computer to a display, DVI also includes support data channels to allow the computer to identify the device being connected. This interface is called EDID and is basically a serial EEPROM with information about the display vendor and supported resolutions.

This interface can be also thought of as an serial encapsulation of parallel data, but compared to SDI it uses three serial data channels to transport the data. This reduces bandwidth requirements for a single serial channel and therefore reduces the quality requirements for used cabling.

IV Project requirements

This section describes the various requirements for the processing hardware. The device using the FPGA video processor is to be used in an medical environment for displaying live video from endoscopic cameras during surgeries. The system also has to be able to record the video and store the feed either locally or via network, but these functions are handled by a standard x86 system embedded in the device and as such are not the topic of this work.

4.1 Video deinterlacing

Based on customer requirements, the video processor must handle two input video formats, one progressive and one interlaced video feed. This requirement comes from the fact that with this system, a HD camera will be usually delivered which has two settings for output video resolution, 720p and 1080i. Since the customer wants to be able use a standard monitor (most of which do not handle interlaced video timings very well), the 1080i interlaced video must be internally converted to 1080p. This video format can be displayed on a standard monitor with no timing problems.

4.2 Low latency

The system is to be used for live video display during surgical operations. The device processes the video signal from endoscope which is then output on a monitor. The surgeon navigates by the displayed video image and so the processing delay must be as small as possible. If the delay was too large, the surgeon would see the operating tool later than he or she may do a critical intervention to the patient and would therefore be a hazardous behavior.

4.3 On-screen display generation

When displaying live video from the endoscopic camera, the system also has to mix into the picture some additional information. This information includes patient name, system settings, buttons for touch controls if the attached monitor has a touch panel and an indicator of free space available for the recorded video.

4.4 Video stream switching

One of the features that the customer requested was the ability to display both the live video feed and an administrative GUI application running on the system on a single monitor. From this stems the requirement to switch between two video streams seamlessly, not to cause the attached display monitor to resynchronize to a new timing should the transition be made by a simple switch.

4.5 Image capture

The system must be able to take snapshots of the displayed video feed. Although this could be handled by the embedded x86 system in a similar way as the video recording, because of another request by the customer that the captured image be freezed for a few seconds for a surgeon to see what the picture is, it was decided that this function will be handled by the hardware.

V Device family selection

This section discusses the selection of FPGA device family to realize the required functions of the system. After preliminary tests of video processing components on a separate board developed for said testing, it was concluded that even low cost FPGA families from major manufacturers were sufficient to implement Full HD video processing. Based on this conclusion, the family selection was limited to low cost field programmable gate arrays. FPGA families are usually divided into several generations, each generation contains devices with varying sizes and features and each device is manufactured in various packages and speed grades.

5.1 Design requirements

This work requirements described in previous chapter were transferred to design requirements for the FPGA chip performance and required peripheral functions. Since the design seemed to most likely require a frame buffer component, some form of large temporary memory was needed to store the incoming video frames. It was decided that the system will use DDR2 memory for it’s relatively low cost and sufficient performance. Based on the incoming video formats specified by the customer, the required memory bandwidth for the frame buffer was estimated (in bytes):

1920(width) _ 540(height) _ 60(fps) _ 4(Bpp) _ 2(R +W) = 474MB=s

Including a margin for read/write bank switching and memory refresh cycles, it was concluded that a single DDR2 x16 chip fulfills this bandwidth requirement, since (in bytes):

2(datawidth) _ 2(effectiveperclock) _ 200000000(frequency) = 800MiB=s

Therefore, the target device must be able to instantiate a DDR2 x16 memory controller core to interface to the external DDR2 x16 memory chip. The total number of pins required was estimated to be in the range of 150 to 180. This included two video inputs, USB link connection, DDR2 memory interface and support I/O functions of the FPGA. The maximum frequency required for any part of the design was estimated to be 150MHz-180Mhz for the most demanding components. Namely the DDR2 memory interface and the deinterlacer module. The selection of FPGA device family was based on these requirements together with a preference of wide availability and good online support.

5.2 Altera Cyclone family

Altera manufactures low cost FPGA chips under the Cyclone family name. This family includes devices from 3000 logic elements (LEs) to about 150000 LEs. The FPGA chips of this family also contain up to several megabits of embedded memory blocks, multipliers for DSP processing and are offered in a range of package sizes and pin counts. The Cyclone family supports the instantiation of DRAM memory device controllers. The Cyclone family is currently divided into four generations, Cyclone I to Cyclone IV (as of time of writing of this work, the Cyclone V family is announced by Altera with available samples). These generations differ in power consumption, densities, supported peripheral features and the maximum frequency the logic fabric of the device is able to support for a given HDL design. The family generation selection was reduced to Cyclone III and Cyclone IV. These families are more advanced than the I/II generations, due to advances in lithographic processes are cheaper and have better availability. Also, due to the Cyclone IV being basically a “shrink” of Cyclone III, the conversion of a given design between these families is a relatively simple task.

5.3 Xilinx Spartan family

The other major manufacturer of Field Programmable Gate Arrays, Xilinx Inc., offers device families with similar features as Altera. The Xilinx version is branded under the name Xilinx Spartan.

The Spartan devices are also divided into device generations based on advancements in FPGA design. The device families considered were Spartan-6 and Spartan-3 due to a relatively large community support for designs based on these devices. The FPGA chips from the Spartan-6 device family include hardened memory controller blocks for interfacing an external DRAM memory chip.

5.4 Lattice Semiconductor Corporation

Lattice Semiconductor is the third largest FPGA manufacturer and although it was also taken into consideration, for a perceived lack of good online support the devices from Lattice Semi were not given any further evaluation.

VI CONCLUSION

The device family selected to implement the requested functions of the system was Altera Cyclone III/IV. This decision was influenced by several factors. The low cost FPGA devices from Altera are on par with low cost devices offered by Xilinx when comparing features like price, performance, capabilities and package options. Since the selected manufacturer will probably be also used in future projects requiring some form of FPGA processing, availability of IP cores was taken into account. Since the company is trying to enter into medical video processing market, it is necessary to have video processing cores available. Although there exist many for the Xilinx devices, Altera offers a complete package for video processing, the Altera Video and Image Processing Suite (VIP)[12]. Both manufacturer’s FPGA development environments were evaluated, the Altera Quartus II and Xilinx ISE Design suite. It was concluded that the Altera Quartus II is a better solution, because it integrates all required functions (design entry, compilation, simulation, programming) into one package. Also taken into account was the large availability of cores adhering to the Altera Avalon Interconnect Fabric standard, which together with the SOPC Builder software simplifies system design. To provide a complete and realistic overview of the reasons influencing this decision, it must also be noted, that one of the reasons tipping the selection into Altera’s favor was the authors familiarity with devices of this manufacturer fromlectures at FI MU.

References:

[1] Crookes, D.: Design and implementation of a high level programming environment for FPGA-based image processing. IEE Proceedings on Vision, Image, and Signal Processing 147(4), 377 (2000)

[2] Rao, D.V., Patil, S., Babu, N.A., Muthukuma, V.: Implementation and Evaluation of Image Processing Algorithms on Reconfigurable Architecture using C-based Hardware Descriptive Languages. International Journal of Theoretical and Applied Computer Sciences 1(1), 9–34 (2006)

[3] Neoh, H., Hazanchuk, A.: Adaptive Edge Detection for Real-Time Video Processing using FPGAs. Global Signal Processing (2004)

[4] Spartan-3A DSP FPGA Video Starter Kit user Guide, http://www.xilinx.com

[5] Xilinx Inc. Embedded System Tools Reference Manual, http://www.xilinx.com

[6] Ramachandran, S.: ‘Digital VLSI System Design’, New York: Springer, chapter 11, 2007.