AN FPGA-BASED LOW-COST FRAME GRABBER FOR IMAGE PROCESSING APPLICATIONS

Donglai Xu, Said Boussakta,* and John P Bentley

School of Science and Technology, University of Teesside,Middlesbrough, TS1 3BA, UK
d.xu{j.p.bentley}@tees.ac.uk
* School of Electronic and Electrical Engineering, University of Leeds, Leeds, LS2 9JT, UK
s.boussakta@ee.leeds.ac.uk

Abstract: This paper presents a low-cost frame grabber, which was specifically designed as part of a real-time motion detection system for high-resolution images. The frame grabber is FPGA-based to minimise size of the PCB and improve reliability of the system. It also acts as a backend add-on card for an IBM-PC compatible. The experimental tests carried out on different machines show that the board implemented meets all specifications required by the system, and performs well. The captured frames are clear, well contrasted and jitter-free in both live and still video modes and their quality is comparable to that available from equivalent commercial systems.

1. INTRODUCTION

The work reported in this paper is part of a research programme to develop a real-time motion detection system for high-resolution images. A key component in such a system is a frame grabber to store digitised images [1]. The customised low-cost system which we have designed and implemented makes use of field-programmable logic arrays (FPGAs), allowing rapid alteration of functionality during development. More fundamentally, as users of the system have control over the hardware, it is possible to add hardware image processing blocks which interface directly with the frame-grabber, sitting on the same PCB, by means of either commercial signal and image processing chips or user-designed FPGAs. This makes a powerful but low-cost test-bed to prototype systems where image processing is shared between hardware and a host computer.

2. FPGA-BASED FRAME GRABBER

The frame grabber presented here has the following specification:

- To capture images with a resolution of 512×512 pixels using a standard CCIR input video signal from commercially available monochrome cameras;
- To digitise to an accuracy of 8 bits using an on-board ADC;
- To store 4 images using 1 MB of on-board VRAM;
- To display either live images or a selected stored frame on an attached monitor using an on-board DAC;
- To implement all the control logic in FPGAs;
- To create the system as a 2-layer PCB occupying one ISA expansion slot;
- To save a selected frame to the PC's hard disk in TIF format;
- To support the use of C software to control the capture, storage and display of images and implement various image processing algorithms.

The frame grabber is connected to a PC through its ISA local-bus as a backend add-on card [2], [3], [4]. During operation, video pictures from a video camera are captured, and either stored in VRAM for processing later or passed to a D/A converter for live display. The operations are performed by a digitizer, PLL (phase locked loop), D/A converter and VRAM under the control of an FPGA-based system controller. Figure 1 shows the architecture diagram for the complete frame grabber. Its major components are presented below in some detail.

2.1 Digitizer module

The digitizer module was implemented using a Plessey SP94308 video ADC combined with an LM1881 sync separator.

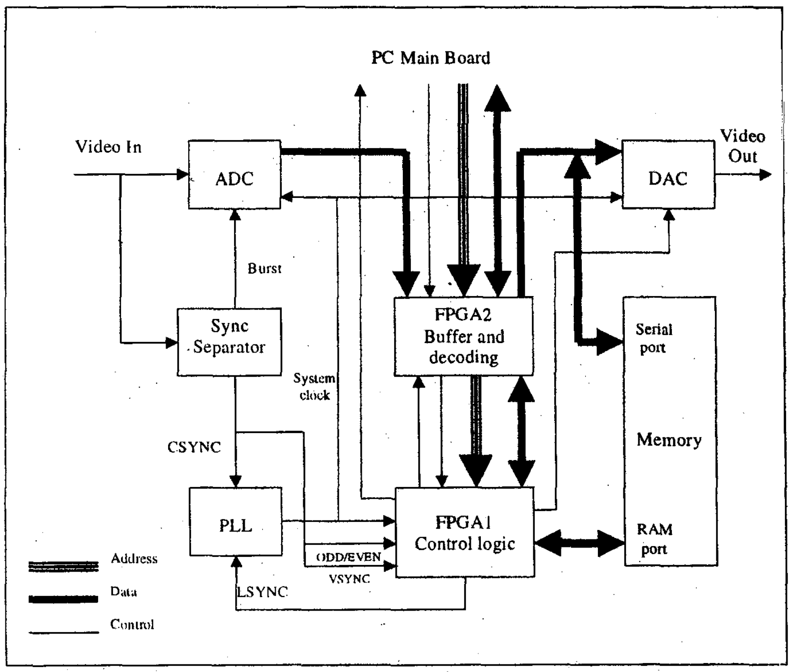

Figure 1 Architecture diagram of the frame grabber

The ADC samples the input waveform on the rising edge of the input clock and produces a latched digital output. A tri-state buffer is connected to the digital output in order to isolate the ADC during the still video mode. This is necessary because, during the still video mode, the DAC receives data from memory while in live video mode the ADC sends data to both the DAC and the memory.

Either the video sync or the video burst pulses can be used to drive the clamping circuit within the ADC. In our case the burst pulse derived from the LM1881 is used. The LM1881 provides all the necessary timing information associated with the input composite video signal. Of its other outputs, CSYNC (composite sync) is used to trigger the PLL, VSYNC (vertical sync) controls frame synchronisation and OD/EV (odd/even field signal) is combined with other signals to address the memory rows.

2.2 DAC module

The DAC (Plessey MV95308) latches the data on *the falling* edge of *the clock while* the *ADC* samples the input on the rising edge of the clock, which means that the same clock signal can be used for both. All the necessary video pedestals are mixed with data at the DAC output in order to produce a standard composite video signal.

2.3 PLL module

The PLL (74HCT4046) is a device that is employed to produce the 10 MHz global clock of the system. This clock must be fully synchronised with the video composite sync.

Concisely, the PLL consists of a phase detector, a low-pass filter and a VCO (voltage controlled oscillator), acting as a closed loop. The phase detector compares two input frequencies and generates an output which is a measure of their phase difference. This phase error signal, after being filtered and amplified, causes the VCO output frequency to deviate in order to eliminate the detected difference. If a frequency divider (i.e. a synchronous counter) is used between the VCO and the phase detector input, the result can be any fixed multiple of the base frequency. In this case a counter counting up to 640 is used in order to achieve the required 10 MHz frequency because the unbroken sequence of the video sync pulses has a frequency of *15.625 KHz*, *which* gives *10 MHz if* multiplied by 640.

The output of the VCO is the global clock of the system. It clocks the 10-bit counter which is reset every 640 pulses. The MSB of this counter (LSYNC) is fed back to the other input of the phase detector in order to be compared with the video composite sync. The PLL keeps these two signals locked in phase and frequency so that the VCO output runs at 10 MHz.

2.4 Video memory module

The 1 MB on-board video memory which is required to store up to four 512×512×8 bit image frames is implemented using eight TC524256B VRAMs. Each chip is organised in 512 rows by 512 columns by 4 bits so that two of them can be combined to accommodate a single frame. Thus for every 8-bit pixel in a frame one nibble is stored in every chip of the pair.

2.5 FPGA-based controller

At the heart of the frame grabber are two Xilinx FPGAs [5], [6], which minimise the component count and PCB size, and which allow on-board re-programmability. Two relatively small devices from the Xilinx range are used to avoid routing complexity on the PCB, since a single device would need a large number of pins; also, the devices are cheap and may be reconfigured individually. They include all the required digital logic for controlling the other support chips on the board, namely the sync separator (which generates important timing signals from the input video), the ADC, the DAC, the VRAM memory and the PLL (which is needed to generate the 10 MHz global clock signal from the 15.625 kHz line frequency embedded in the video signal).

FPGA1 holds a number of logic blocks. First, there is a 10 bit counter, used to count pixels within each line of active video and also, in combination with the PLL, to produce the 10 MHz global clock by frequency division of 640. There is a 9-bit counter, used to count the lines and produce the necessary row addressing for the memory. An address multiplexer feeds the memory with the appropriate address at all times, while a command arbiter prevents conflicts when reading, writing or refreshing the VRAM memory. It generates the signals RAS, CAS, DT\OE, WR\WE and SC needed to create refreshing cycles, internal memory transfer cycles and cycles which read from or write to the computer.

The video control logic needed to produce the required signals for the ADC and DAC is also contained within FPGA1 as is the logic needed to control the flow within the video data bus. When the grabber is operated in live video mode, the ADC feeds both the memory and the DAC, whereas the memory feeds only the DAC in still video mode.

Finally in FPGA1 there is PC interface logic to handle the bi-directional data bus, including three registers, two of 9-bit width for row and column addresses respectively and one of 4-bit width, for controlling the status of the system. The PC data bus is switched between the three registers and the memory by decoding the A1 and A2 PC address lines. By using appropriate I/O addresses the user is able to:

1) Load the row or column address register with the desired address;
2) Switch from live video mode to still video mode;
3) Select between the stored frames;
4) Read from or write to the video memory at a position defined from the row and column registers;
5) Initialise an FPGA re-programming cycle.

FPGA2 contains the external address decoding logic allowing FPGA1 to be accessed like an I/O mapped peripheral occupying 8 I/O addresses. The address offset can be selected via on-board dip-switches with a choice of 32 different octal address sets from Hex 300 to Hex 3F8. FPGA2 also has internal buffering and tri-state buffering as all the address, data and control lines which are used directly by FPGA1 need to be externally buffered and Xilinx input IBUF buffers do not offer sufficient drive capability.

The software required to integrate, test and demonstrate the capabilities of the frame grabber is written in C. The main functions of the program are to:

1) Save the selected frame into TIF format;
2) Reprogram the FPGAs;
3) Produce a standard pattern on the screen;
4) Control the operation mode (Live or Still video).
5) Select which of the 4 stored frames is to be displayed;
6) Copy a user defined window from a position on the screen and display it at another position;
7) Draw a diagonal line using a user specified grey level;
8) Control the execution of various image processing algorithms.

3. CONCLUSIONS

The final result is a PC-compatible image frame grabber which meets the required specification in every respect at a relatively low cost. Captured frames are clear, well contrasted and jitter-free in both live and still video modes and their quality is comparable to that available from equivalent commercial systems costing about three times as much. In this design, the frame grabber acts as a backend add-on card for an IBM-PC compatible through its ISA local-bus, which can be plugged into one of the PC's expansion slots. The work is currently being extended to produce a similar product for the PCI bus around which it is intended to build a real-time motion detection system.

REFERENCES

1. D. M. Avedon, *Electronic Image System: Design, Application and Management*, McGraw-Hill, 1994.
2. D. C. Sastry and J. Jagadeesh, "A High-speed Low-cost Frame Grabber", *Computer*, Vol. 29, 1997, pp102-103.
3. G. Bush, M. J. Smith and D. H. Evans, "Frame Grabber for Sequential Real-time Video Image Digitisation and Transfer to Microcomputer System", *Medical and Biological Engineering and Computing*, Vol. 32, 1994, pp476-478.
4. T. Shanley and D. Anderson, *ISA System Architecture*, MindShare, Inc., 1995.
5. Xilinx Inc., *XACT 2000/3000/4000 Programmable Gate Array Development System Reference Guide*, 1998.
6. Xilinx Inc., *The Programmable Logic Data Book*, 1998.