

# Compact Web Manufacturing Process Defect-Detection System Uses a Camera and ADI Blackfin® Processor

By Hossain Hajimowlana, PhD, PE  
[hossain.hajimowlana@analog.com]

## INTRODUCTION

One of the aims of industrial machine vision is to develop computer and electronic systems to replace human vision in quality control of industrial production. Web inspection systems are currently used for defect detection and quality control in numerous applications, such as the manufacture of high-tension-cable insulation, paper, plastic bags, strip steel, fuel pellets, chip packaging, wood, cloth, and weaving machines. Automatic inspection systems have numerous advantages over manual inspection. The manual inspection of surface defects is a tedious, if not impossible, task—often because of the small size of many defects and the very large areas to be inspected.

Conventional inspection systems consist of a line-scan camera, host computer, frame grabber, and one or more dedicated processing circuit boards. In this article, we discuss the development of a new integrated design environment—intended for real-time defect detection—that eliminates the need for an external frame grabber and eliminates or reduces the need for other associated host computer peripheral systems. The processing board, containing a reconfigurable field-programmable gate array (FPGA), is mounted inside a DALSA CCD camera. The FPGA is directly connected to the video data stream and outputs data to an associated ADI Blackfin type ADSP-BF535P processor for further processing. Using an FPGA for low-level processing alone represents an excellent trade-off between software and special-purpose hardware implementations. The processed data may be transferred through a USB or Firewire port to a PC for storage, monitoring, and additional processing. The present system is targeted for web inspection but has potentially broader applicability. Figure 1 shows a basic block diagram of an industrial inspection system that operates without a frame grabber.

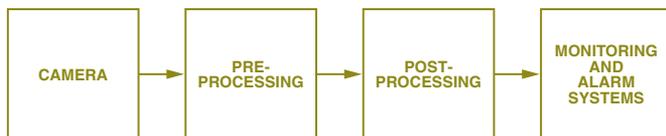


Figure 1. Block diagram of an industrial inspection process.

The defect detection task is delegated to two algorithms. The first one (*preprocessing*) undertakes the role of a conservative gross filter. Its objective is to detect all possible defects. This task will mainly be carried out in real time using the FPGA as a video-stream filter. The intent here is to provide a reliable means of rapidly identifying suspect regions that may or may not be finally classified as defective. In *postprocessing* we aim to identify the type and severity of the defect using an ADI Blackfin ADSP-BF535P processor. Typically this latter process has been carried out in the host computer, but a significant part of the process can be done locally in the modified camera system itself, using the powerful ADI Blackfin processor.

The system described in this article was developed at the University of Windsor, Canada as a PhD research project.

Establishing an appropriate computing environment for *real-time* applications, such as web inspection, is a challenging task. In this article, “real-time” describes any imaging system capable of receiving *and processing* continuous video data. A real-time system must perform *all* of the required operations within the critical time-frame allotted. Even under conditions of extreme system loading, the execution times and logical sequence of the system’s response must be correct. The system described in this note can achieve real-time video processing for up to 30 million samples per second.

## Test Setup

A test fixture that simulates a web manufacturing process with a provision for variable-speed operation has been set up as shown in Figure 2. The test setup comprises a DALSA TDI line-scan camera, a motorized drum with shaft encoder for TDI synchronization, and a dc light source with fiber-optic light guides. Our FPGA/DSP processing board is mounted above the regular camera control boards. Defective samples from various web sources are used for testing and verification of candidate algorithms.



Figure 2. Test setup for production-line simulation.

## Hardware

The complete hardware assembly is shown in Figure 3.

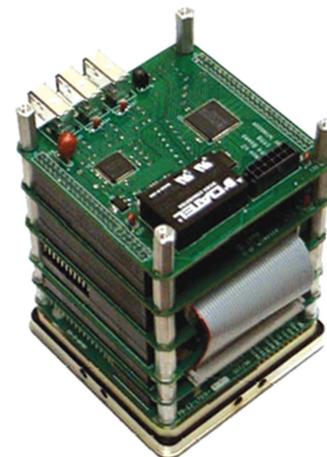


Figure 3. Processing hardware.

Figure 4 shows the block diagram of the processing system. The preprocessed data from the FPGA are stored in a FIFO, which buffers the data for further processing by the *digital signal processor* (DSP). The processing hardware assembly comprises three PCBs—the FPGA board, the DSP board, and a USB/Firewire board for linking to a PC. Other resources are shared between the boards.

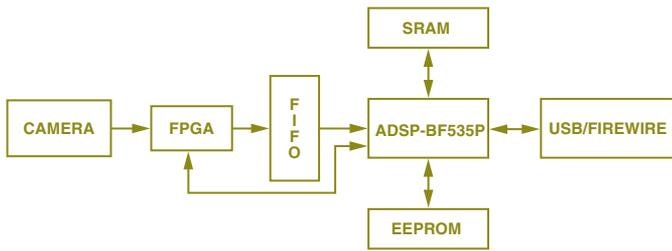


Figure 4. Block diagram of the processing board.

### FPGA versus DSP

Our processing board supports processing in either the FPGA or the DSP—or *both*. How does one choose the disposition of processing capability for an application?

A DSP is a specialized microprocessor, typically programmed in C, with the occasional use of assembly code to improve system performance. The DSP is well-suited to extremely complex math-intensive tasks, involving conditional processing. It is limited in performance by the clock rate—and the number of useful operations it can do per clock. In contrast, an FPGA is an uncommitted “sea of gates.” The device is programmed by connecting the gates together to form multipliers, registers, adders, and so forth. Math is done in hardware by interconnecting these building blocks. The blocks can range, in degrees of complexity, from a single gate to the very high level of an FIR filter or an FFT—given enough gates and the ability to interconnect them. Performance is limited by the number of available gates on the chip and by the clock rate.

FPGA and DSP thus represent two very different approaches to signal processing—each excelling at different things. There are many high-sampling-rate applications that an FPGA can do easily, but for which a DSP is unsuitable. Equally, there are many complex software problems—easy for a DSP—that the FPGA cannot address.

As a result of these complementary properties, the ideal system would split the work between FPGAs and a digital signal processor. In our web inspection system, most of the operations on an image *per se* are simple and very repetitive; so these primitive operations are best implemented in an FPGA. However, an imaging *pipeline* is often used to identify “blobs” or “regions of interest” in an object being inspected. These blobs can vary in size, and subsequent processing thus tends to be more complex. The algorithms used are often *adaptive*, depending on what the blob turns out to be. All things considered, a DSP-based approach is typically more effective at the back end of the imaging pipeline.

The Xilinx Spartan IIE series FPGA is used in our system, because it has additional *configurable logic block* (CLB) features that operate at greater speeds for memory-based designs—and it supports system clock rates of up to 200 MHz. A CLB includes a four-input function generator, carry logic, and storage element. Each CLB also contains logic that combines function generators to provide functions of five or six inputs.

Currently our design uses an XC2S200E, which has 5292 logic cells and 200K system gates. This FPGA has sufficient resources for many of our target applications and is packaged appropriately for building into a single-board in-camera system.

### Choosing the Right DSP

It was very important to choose the right processor for our application. Power, cost and packaging, speed, performance, and availability of the right peripherals and development tools were the main factors in our decision to choose the ADSP-BF535P.

The ADSP-BF535P is a member of the Analog Devices Blackfin DSP product family. It combines a dual-MAC DSP engine, RISC-like microprocessor instruction set, and single-instruction, multiple-data (SIMD) multimedia capabilities into a single instruction-set architecture.

### Power

Power plays an important role in web inspection systems. It is not uncommon to use a dozen of these cameras in the field. There may be applications where there is no need to further process the data by DSP, or the application can run at lower DSP clock speed. By using ADSP-BF535P, we did not need to sacrifice power for performance. In Blackfin ADSP-BF535P power can be reduced by reducing the core voltage and frequency. For this purpose, an external companion power management chip, the ADP3053, is available for dynamic control of the core voltage levels. Blackfin DSPs provide additional power control capability by allowing dynamic scheduling of clock inputs to each peripheral. Also, internal clocks are routed only to enabled portions of the device. For example, the 256KB on-chip L2 memory is divided into eight 32KB banks. This feature enables power to be reduced, as these banks are clocked only when they are accessed.

### Cost and Packaging

The Blackfin ADSP-BF535P, a general-purpose DSP, typically costs much less than its closest digital-processing counterparts. In this application, its compact PBGA260 packaging format fits neatly into our 3.5" × 3.5" PCB.

### Speed

Web inspection systems are demanding processing applications using intensive real-time algorithms. Thus, fast programmable, general-purpose digital signal processors are needed to handle the challenges presented by high speed data rates. Maximum core clock (CCLK) for the ADSP-BF535P is 350 MHz. We were able to successfully run our applications at 300 MHz (less in some cases to reduce power). CCLK pulses are generated via a PLL that has available CCLK to system clock (SCLK) ratios of 1 to 31. With a 20-MHz external oscillator we were able to achieve CCLK of 300 MHz. Depending on the CCLK, a maximum SCLK of 133 MHz can be achieved.

### Performance

The Blackfin Processor is highly optimized to execute DSP applications code efficiently. In image processing applications, we usually deal with different sizes and kinds of filters (*infinite impulse response*, IIR; and *finite impulse response*, FIR) or apply *fast Fourier transform* (FFT) to the data. Table 1 shows some benchmarks done on ADSP-BF535P.

Table 1. ADSP-BF535P signal processing algorithm benchmarks

Benchmark Description	Number of Clock Cycles
256-point complex FFT	3,176
Block FIR Filter	$[(\text{Number of Samples})/2] \times [(\text{Number of Taps})+2]$
Biquad IIR Filter	$2.5 \times (\text{Number of Biquad Sections}) + 3.5$

## Peripherals

The ADSP-21535P contains a rich set of peripherals connected to the core via several high-bandwidth buses, providing flexibility in system configuration as well as excellent overall system performance. It provides USB and PCI buses for glueless peripheral expansion without the need for costly external components.

For transferring processed data from the camera to a PC at medium data rates, USB seems a great solution. Because of the relatively high power consumption of the processing board, however, we are unable to use the bus-powered feature of USB. One of the most useful features of USB is that it is hot-pluggable, and the scanning camera can be plugged in or out of the monitoring system (a PC in this case) without a need to turn off the PC. For high data rate applications, where a series of monitoring cameras is used, IEEE Std. 1394 Firewire is recommended—it has 30 times more bandwidth than USB 1.1.

## Development Tools

We have used VisualDSP++™ to develop and debug our codes. VisualDSP++ includes an integrated development environment (IDE) and a debugger that provides efficient project management, enabling us to move easily between editing, building, and debugging of programs. An evaluation platform for ADSP-BF535 is also available.

## Algorithms

Different algorithms have been successfully simulated and implemented in the FPGA/DSP processing system. Here we briefly describe fuzzy logic and 1D AR algorithms. The interested reader can refer to the references for more detail.

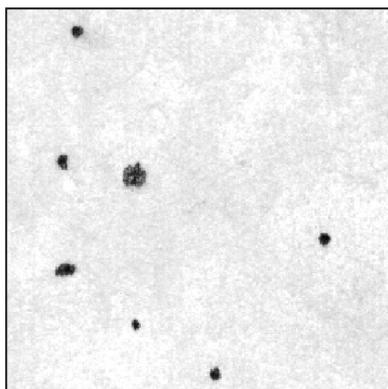


Figure 5. Stain defects on the sample.

Locating the exact position of the defects is performed using the ADSP-BF535P, employing the one-dimensional autoregressive (1D AR) algorithm.

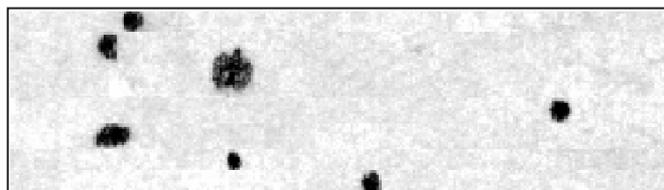


Figure 6. Image of selected defective lines of Figure 5.

## Fuzzy Logic

A new and exciting application of fuzzy logic is for defect detection in web inspection systems. Defects in a manual detection system are usually described and identified by linguistic variables, e.g., *darker or brighter* regions; *smaller or larger* objects, so fuzzy logic appears to be a good candidate for defect-detection applications.

To apply the algorithm, a set of texture features is derived off line from the “golden” (defect-free) template. These texture features are employed as inputs to the fuzzy decision engine. Outputs are obtained for the entire range of the possible inputs and stored in a look-up table (LUT). The proposed algorithm has been tested on a random texture sample with several stain defects (Figure 5); the image is digitized at a resolution of 256 rows  $\times$  256 columns with 8 bits of gray-level information. The result of applying the algorithm is shown in Figure 6.

## Hardware Implementation of the 1D AR Algorithm

Figure 7 shows the simplified signal flow of the 1D AR algorithm.

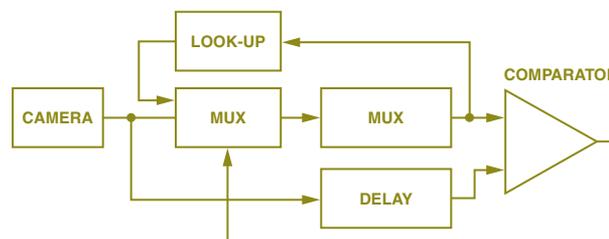


Figure 7. Simplified signal flow diagram of the 1D AR.

The 1D AR algorithm can easily be implemented in ADSP-BF535P and combined with the fuzzy-logic algorithm to detect the exact position of the defects in the defective lines.

The heart of the AR algorithm is an IIR filter (AR predictor). Since IIR filters are faster than FIR filters, they are more suitable for real-time applications. The experiments show that an 8<sup>th</sup>-order filter is suitable for most of the textures. The computation units perform single-cycle operations, and there is no computation pipeline. The gray levels of the defective lines' pixels can be stored in on-chip SRAM, whence they are transferred out invisibly to external memory—or to the PC, through DMA controllers.

## CONCLUSION

We have described an in-camera prototype processing board that basically consists of an FPGA and an ADI Blackfin processor. Some important issues for real-time web inspection systems were discussed, as well as the parameters—such as: power, cost, packaging, speed, performance, and the need for the right peripherals and development tools—that led us to choose ADSP-BF535 for our application. We showed that the Blackfin ADSP-BF535 provides an excellent platform for realizing low-power, high-performance, real-time embedded applications. 

## REFERENCES

- [1] S.H. Hajimowlana, R. Muscedere, G.A. Jullien, J.W. Roberts, “A Novel Approach for Defect Detection in Web Inspection Using Fuzzy Fusion of Texture Features,” *Journal of Vision*, a technical quarterly published by the Society of Manufacturing Engineers (SME), 4<sup>th</sup> quarter issue, 1999.
- [2] S.H. Hajimowlana, R. Muscedere, G.A. Jullien, J.W. Roberts, “An In-Camera Data Stream Processing System for Defect Detection in Web Inspection Tasks,” *Journal of Real Time Imaging*, special issue on real time detection of defects, Spring 1999, Academic Press.
- [3] S.H. Hajimowlana, R. Muscedere, G.A. Jullien, J.W. Roberts, “Defect Detection in Web Inspection Using Fuzzy Fusion of Texture Features,” *Proceedings of ISCAS 2000*, Vol. III, pp. 718-721, May 2000.

Trademarks and registered trademarks are the property of their respective owners.