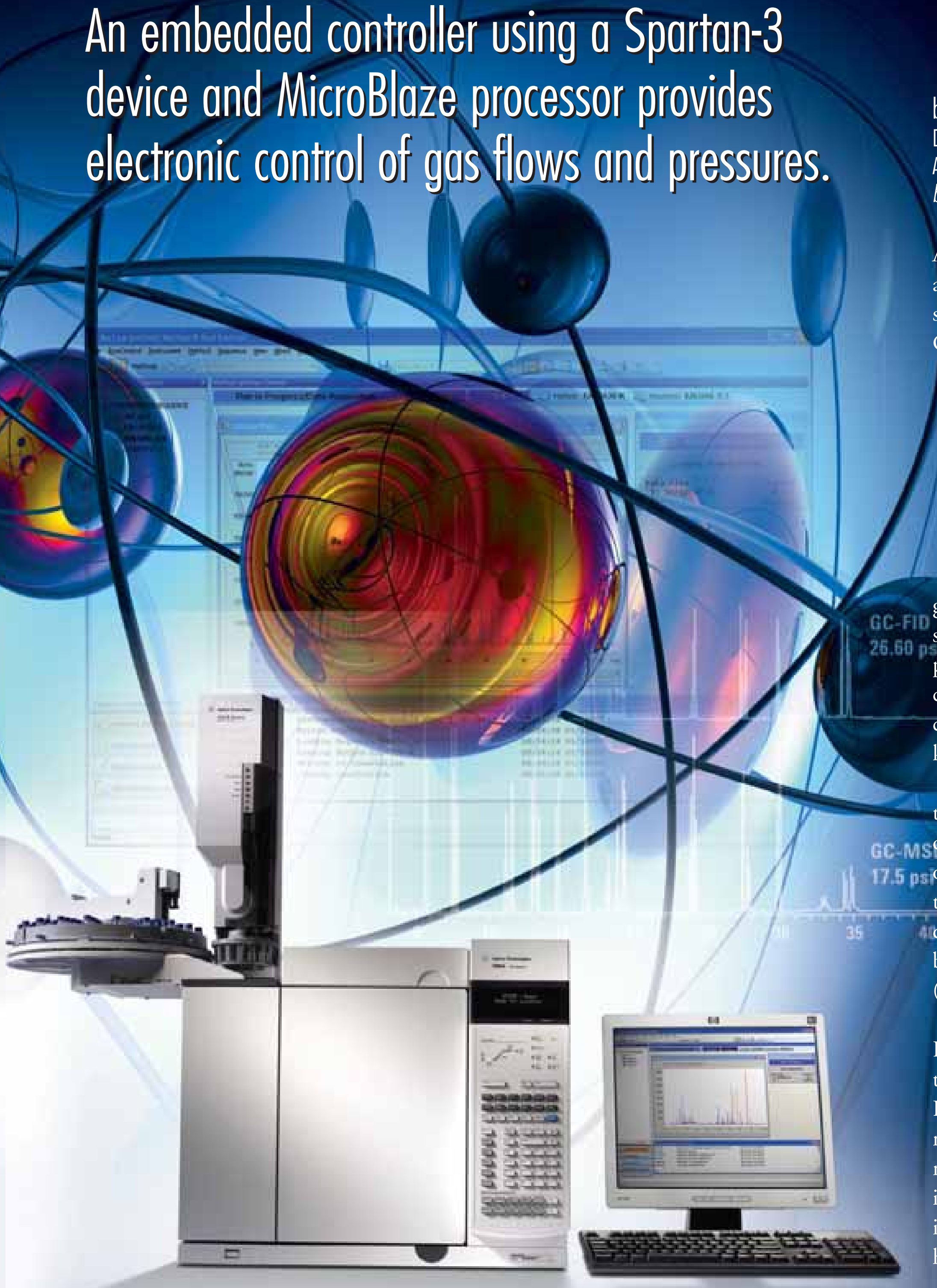
FPGA-Based Controller Enables Precise Chemical Analysis

An embedded controller using a Spartan-3 device and MicroBlaze processor provides electronic control of gas flows and pressures.

by Robert Henderson
Development Engineer
Agilent Technologies
bob_henderson@agilent.com



A gas chromatograph (GC) is a chemical analysis instrument that separates volatile substances in a complex chemical sample. GCs help answer such basic questions as:

- "What contaminants are in this ground water"?
- "Is this gasoline formulated correctly"?
- "Are there any residual solvents in this pill"?

The basic mechanism for gas chromatographic separation is the distribution of a sample between two phases (a stationary phase and a gas mobile phase). In modern chromatography, the stationary phase is coated on the inside of a long narrow tube known as a column.

As the sample passes down the column, the different chemical components travel at different rates and are detected at the exit of the column by a detector. In addition to the gas supporting the flow of the sample down the column, each detector also uses between one and three supporting gases (see Figure 1).

Agilent Technologies (spun off from Hewlett Packard in 1999) has been a part of the GC business since 1965. In 1973, Hewlett Packard introduced the world's first microprocessor-controlled analytical instrument, the 5830 GC, and over the years introduced a series of next-generation instruments, including the Agilent 7890A high-performance GC in 2007.

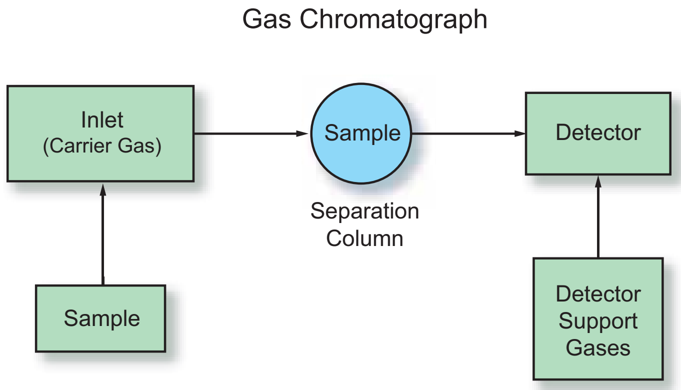


Figure 1 – Block diagram of gas flows in a gas chromatograph

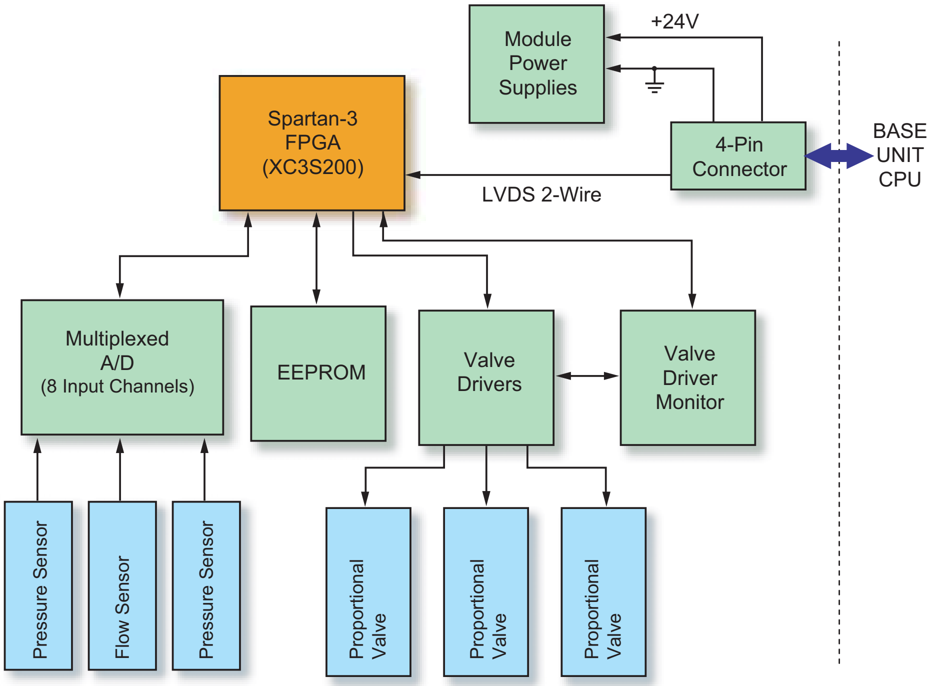


Figure 2 – Block diagram of Spartan-3 FPGA-based pneumatic module

GCs have evolved measurably over the last 20 years, from simple mechanical pressure and flow regulators to sophisticated electronic closed-loop pressure and flow controllers that allow for the storage and retrieval of the entire instrument setup. Additionally, this capability also allows for programmable setpoints and for the logging of any deviation from the setpoint(s).

Spartan-3 FPGA-Based Pneumatic Modules

Because of the disparate different application needs of our customers, we designed the 7890A GC with a modular structure, allowing as many as six pneumatic modules to communicate with a common "base unit" that provides the central CPU and memory, communication ports, keyboard and display interfaces, and power supplies.

Pneumatic modules can regulate up to three channels of gas for each inlet or detector. Each module is essentially an embedded controller that receives commands from the CPU in the base unit, but otherwise operates independently. The module controls the three independent pneumatic channels in a closed-loop manner, offloading the CPU in the base unit from this real-time task (Figure 2).

Four-Wire Interface (Module to Base Unit)

From the base unit to the module, the entire interface comprises an unshielded four-wire cable: two wires are for power and two are for communications. The power supplied is an unregulated, full-wave-rectified +24V, and the communications path is a bus LVDS system.

The LVDS communications path is a differential signal path that minimizes noise susceptibility issues and allows for cable lengths long enough to place modules anywhere inside the instrument, maximizing instrument configurability options. The bus LVDS communications operate in a half-duplex mode, allowing the same pair of wires that sends commands to the module to provide command responses from the module.

Additionally, because the communications interface in the base unit is also implemented with an FPGA, the LVDS communications link between the two FPGAs required essentially no hardware, saving cost, reducing complexity, and improving reliability.

MicroBlaze Embedded Processor Core

In order to implement the embedded controller, we configured the Xilinx® MicroBlaze™ embedded processor in the FPGA using the Platform Studio tool provided in the EDK development system. This tool not only defines, synthesizes, and routes user logic, utilizing MicroBlaze processor IP and associated bus interface designs, but also develops and compiles the program code for the MicroBlaze processor. All FPGA hardware description is in VHDL, and all MicroBlaze processor coding is in C.

Specifying the optional barrel shifter and hardware divider in the microprocessor hardware specification file (system.mhs) enabled the processor system to function as a real-time controller. Performing operations such as bit shifting or division in C code instead of VHDL hardware takes too long for some of the time-critical PID controller processes.

Using the Xilinx-supplied template VHDL file, the user logic interfaces to the MicroBlaze processor OPB bus with an OPB to IPIF (IP interface) block. This block takes care of the OPB bus protocol and interface signals and presents a simplified interface to the user logic called the IP InterConnect (IPIC).

The major user logic VHDL blocks complement the external hardware controlled by the MicroBlaze processor, resulting in the overall FPGA system design (Figure 3). Let's review the major components.

Multiplexed A/D Converter Control

A common delta-sigma analog-to-digital converter (ADC) is multiplexed between eight inputs. With this type of ADC, when the input changes the output filter must be "flushed" of the previous channel's data. A VHDL block manages the ADC and its serial data output.

When the MicroBlaze processor selects a specific channel, the VHDL subsystem reads and throws away the old channel's readings, then reads and averages data automatically from the ADC. The MicroBlaze processor specifies to the VHDL block how many readings to average, and thus can perform other operations (such as executing a new command from the base unit CPU) until an averaged reading is ready.

Serial EEPROM Control

The VHDL hardware manages the commands, addresses, and 8-bit data to and from the serial EEPROM and presents a memory-mapped, 32-bit-word-wide interface to the MicroBlaze processor. The EEPROM stores calibration coefficients for the sensors in the pneumatic module and PID coefficients, as well as other configuration and identification information.

Valve PWM Drive and Monitor

After each PID calculation, the MicroBlaze embedded processor outputs another valve drive setpoint to the memory-mapped addresses for the valve drive. The valve drive VHDL state machine uses these values to generate a pulse-width modulated (PWM) drive to the proportional valves at 65 kHz. This high a frequency is used so that in addition to being above the audible range, it is filtered out well by the inductive time constant of the valve, resulting in a smooth current profile.

Signals from the valve drive are read back into the FPGA for diagnostic purposes. This allows the MicroBlaze processor to examine the voltages and determine if a valve is installed and if the PWM drive is able to drive both high and low.

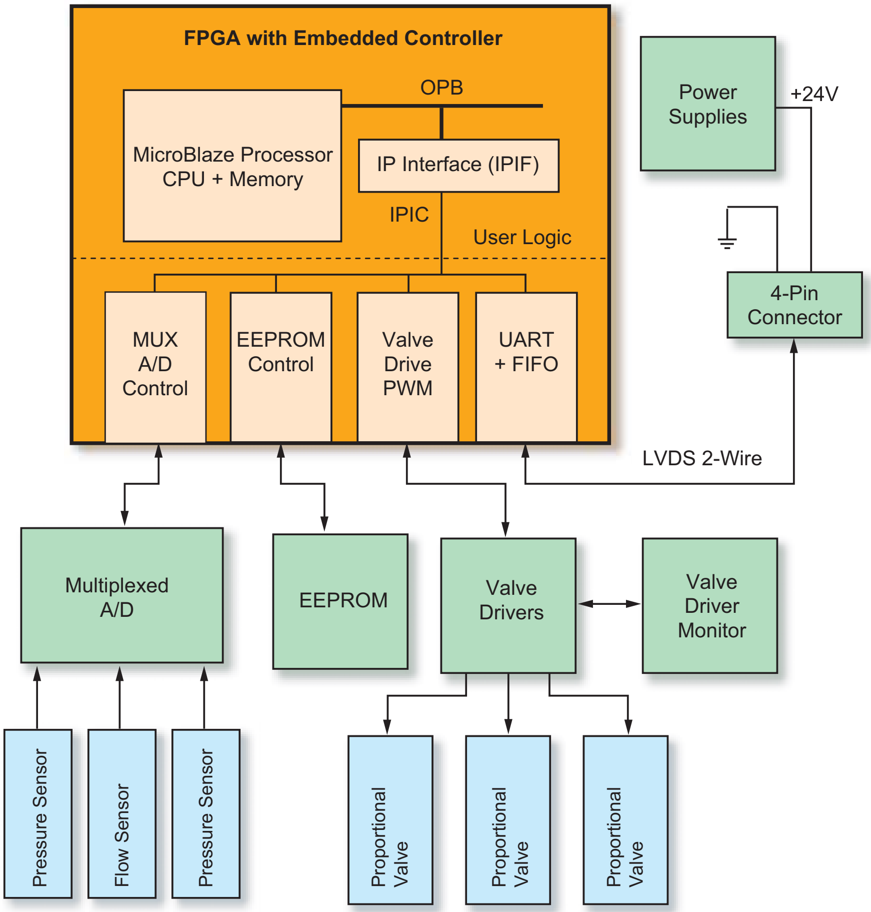


Figure 3 – Overall pneumatic controller block diagram

UART/FIFO

We implemented a UART and FIFO in VHDL hardware to handle the commands and responses with the base unit CPU. Because of the half-duplex protocol, the pneumatic module must only be in the transmit mode during the response to commands.

VHDL hardware automatically returns to receive mode a fixed time after the command response and presents to the MicroBlaze processor a transmit and receive FIFO along with the necessary flags (FIFO empty, FIFO full).

Basic Pneumatic Module Operation

As I mentioned earlier, the physical setpoints supplied by the customer may be static or programmed. They are expressed in physical units such as psig (gauge pressure) or ml/minute (flow rate) and can be set with a resolution of 0.001 psig and 0.01 ml/min. These setpoints are sent at a rate of 50 Hz to each pneumatic control module.

The basic job of the pneumatic controller module is to regulate the pressure or flow to the desired setpoint. This is done with a modified proportional integral derivative (PID) controller.

Command Interpreter

Because there is no clock shared between the base unit CPU and the LVDS pneumatic modules, commands to the pneumatic control module are not synchronized with the closed-loop operation of the embedded controller in the module.

For example, you can send commands while the MicroBlaze processor is in the middle of PID calculations. Because the control loop takes precedence, the command parsing and execution will be temporarily delayed until the MicroBlaze embedded processor is idle. This delay is short enough that it allows the command response to fall within the acceptable response-time limits set by the base unit CPU.

The simple command interface defines commands that set setpoints, read actuals, read and write the EEPROM, specify the gas type (H2, Helium, N2, ArCh4), as well as a number of diagnostic tests on the system, A/D converter, and valve drives.

Filtered Sensor Readings

The sensors read data hundreds of times a second. Two IIR filters in the MicroBlaze processor process the data, which makes use of the barrel shifter hardware configured as part of the processor configuration file.

The first low-pass filter helps reduce raw sensor noise above the bandwidth of the control loop that would result in a wider control band. The second low-pass filter is used on data returned to the base unit CPU. It limits the bandwidth of the data to below 25 Hz so that there is no aliasing of the data in the 50-Hz rate of data to the base unit CPU.

Control Loop

The PID controller function implements a standard PID control with a couple of necessary twists. Figure 4 shows the overall operation of the PID controller.

The first modification of a standard PID is due to the +24V power supply. A fully loaded instrument could have 18 proportional valves to power. Rather than implement a 20W-regulated +24V supply, we decided to use a simple and reliable full-wave-rectified and unregulated supply. A supply like this will track the AC voltage and ripple at twice the line frequency.

The magnitude of the +24V supply can range from 22V to 28V, depending on the AC voltage to the instrument. This results in a variation in the open-loop gain of the system, which can affect stability.

Additionally, the AC ripple on the +24V supply modulates the drive to the valve and can result in pressure or flow variations that are at too high a frequency for the control loop to attenuate away (the disturbances are above the bandwidth of the controller).

We solved these problems by reading the value of the +24V supply with the multiplexed A/D converter and having the MicroBlaze processor calculate a feed-forward compensation term. If the +24V supply increases, the valve drive is automatically reduced to compensate. This requires a divide operation. Because this happens at over 100 Hz and takes away from PID calculation time, the hardware divider was implemented in the MicroBlaze processor system specification.

The second modification from a standard PID controller is due to the pneumatic system itself. The range of setpoints you can define covers a broad dynamic range, with pressure setpoints from 0.2 psi up to 150 psi and flows from 3 ml/min up to 1,250 ml/min.

Not surprisingly, the dynamics of the pneumatic system over this range of operation change a lot. It is possible, for example, to have more than a 30-dB gain change in the transfer function of the pneumatic system (ratio of sensor out to valve drive in) over these ranges. In addition, the bandwidth (poles) of the pneumatic system varies greatly over this operating range.

To obtain good control and step response performance, you can alter the values of the PID coefficients based on the setpoints specified. This alters both the gain and frequency response of the PID controller to match the characteristics of the pneumatic system at that setpoint.

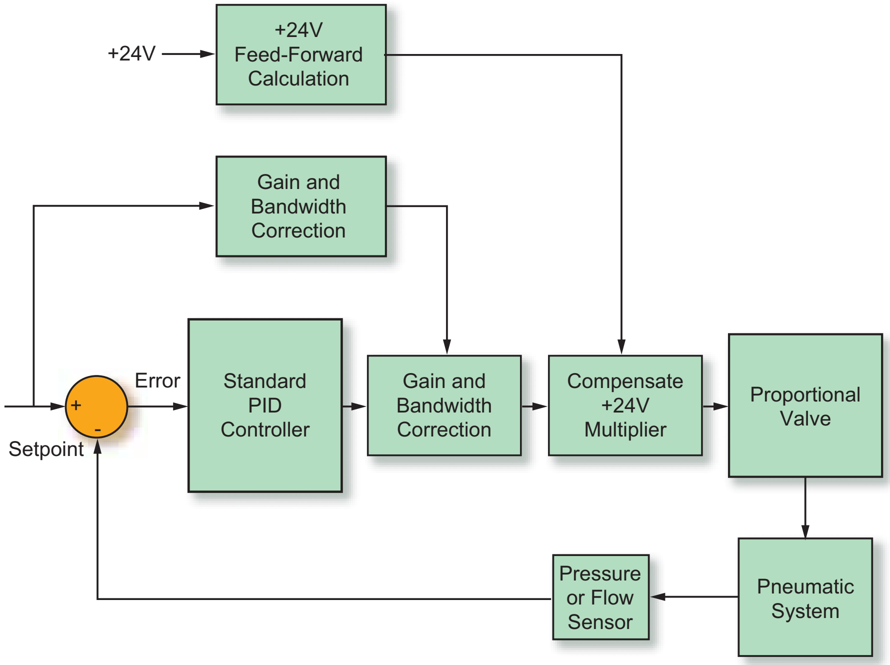


Figure 4 – Modified PID controller

Conclusion

New techniques and improved precision in chemical analysis techniques have enabled our customers to meet increasingly stringent requirements for chemical identification.

Part of that improvement comes from the increased accuracy and precision of gas supplies within the GC. The new Agilent 7890A network gas chromatograph continues this tradition, while the Xilinx Spartan-3 XC3S200 FPGA with MicroBlaze embedded processor helped the product meet its functionality, precision, cost, and modularity requirements.