

## DSP on the Xilinx Spartan-3E Starter Board

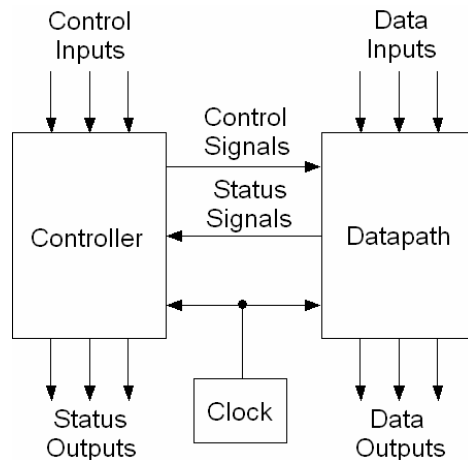
This “How To” tutorial is an excerpt from a recently published text *Embedded Design Using Programmable Gate Arrays* that provides complete Xilinx ISE FPGA projects in Verilog using finite state machines (FSM), controller-datapath modules and Xilinx LogiCORE blocks for real-time processing in DSP, digital communications and digital control on the Xilinx Spartan-3E Starter Board. This text is intended as a supplementary text and laboratory manual for undergraduate students in a contemporary course and for professionals who have not had an exposure to the FSM, the controller-datapath construct and the Xilinx EDA.

By Dennis Silage

### Introduction

Embedded design in Verilog using FPGAs can utilize controller and datapath modules to facilitate the implementation of real-time tasks. The controller module accepts external control and status signals from the datapath module and uses one or more finite state machines (FSM) to coordinate the process. The controller module provides the datapath module control input signals that route the input data, perform processing and output the data. The datapath module stores and manipulates data in registers using combinational and sequential logic and can use one or more FSMs to output the data but not autonomously. The controller can also accept external control signals from and return status signals to an external processor or internal soft core processor to augment the performance of the embedded system.

The controller and datapath construct partitions the design into modules that can be separately verified in simulation. Rather than one module that encapsulates the entire process, the controller and datapath modules then each have a reduced number of interconnections which facilitates the Verilog structural and behavioral synthesis into FPGA hardware. Datapath modules also support the concept of design reuse.



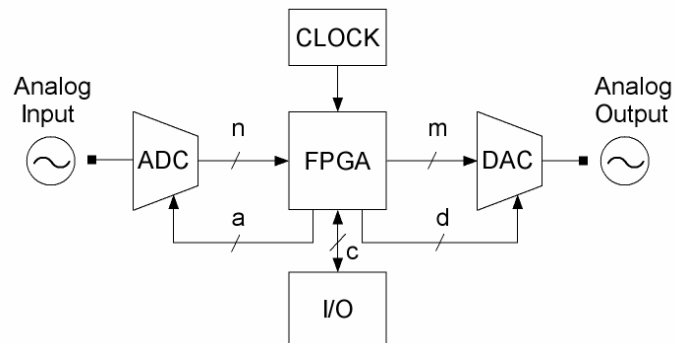
**Figure 1** Configuration of a typical controller and datapath construct

The controller module can be easily modified to accommodate a new task, which can then even include additional datapath modules. The configuration of a typical controller and datapath construct is shown in Figure 1. The synchronous clock input schedules the state transitions of the FSMs of the controller and datapath. Registers can be initialized by a global reset signal, a local reset signal or by a declaration in the behavioral synthesis of the controller and datapath. The reset signal is not shown in Figure 1.

The controller has control input logic signals that initiate the process and status output logic signals that signify the completion of the process. The datapath has only data as an input and output and no external process control logic signals other than those derived from the controller. The datapath outputs status logic signals to the controller to coordinate the process. A clock signal is used to provisionally evoke a state transition in the FSM if utilized in the controller and datapath. The controller control signals and the datapath status signals are required to have a state transition.

## DSP System

The DSP system consisting of an ADC, FPGA and DAC is shown in Figure 2. The ADC provides n-bit data to the FPGA and receives an a-bit data packet for command and control. The DAC receives both m-bit data and a d-bit data packet for command and control from the FPGA. A crystal oscillator provides a clock signal to the FPGA for synchronization and timing of the data transfers and to establish the sampling rate  $f_s$  of the DSP system.



**Figure 2** DSP embedded hardware system

This DSP system executes on the Xilinx Spartan-3E Starter Board using Verilog structural and behavioral synthesis modules that are developed using the Xilinx ISE EDA tools. The Verilog modules are configured as FSMs and the controller and datapath construct. The DSP system is initially implemented as a straight-through processor that inputs and outputs an analog signal without any manipulation to assess the maximum data throughput rate with the Verilog top module s3eadcdac.v in Listing 1. The five Verilog modules operate in parallel in the top module.

### Listing 1 ADC-DAC system top module for the Xilinx Spartan-3E Starter Board

// c 2008 Embedded Design using Programmable Gate Arrays Dennis Silage

```
module s3eadcdac (input CCLK, SW0, SPIMISO, output SPIMOSI, SPISCK, DACCS, DACCLR,
                output SPISF, AMPCS, AMPSD, ADCON, SFCE, FPGAIB, JC1);
```

```
wire adcdav, davadc, adcsck, adcsposd, ampdav, davamp, ampsck, ampssid, sdamp;
wire dacdav, davdac, dacsck, dacssid, clrdac, csdac, csamp, conad, CLKOUT, CLKFX;
wire [11:0] dacdata;
wire [13:0] adc0data, adc1data;
wire [3:0] ampcmd0, ampcmd1, daccmd, dacaddr;
```

```

assign SPISCK = adcsck | ampsck | dacsck;    // SPI clock
assign SPIMOSI = ampspid | dacspsid;        // SPI data in
assign adcspsid = SPIMISO;                  // SPI data out

assign DACCS = csdac;                       // disable DAC
assign DACCLR = clrdac;                     // DAC clear

assign AMPCS = csamp;                       // select prog amp
assign AMPSD = sdamp;                       // disable prog amp shutdown
assign ADCON = conad;                       // ADC convert command

assign SPISF = 1;                           // disable serial Flash
assign SFCE = 1;                           // disable StrataFlash
assign FPGAIB = 1;                          // disable Platform Flash

assign JC1 = conad;                         // monitor sampling rate

s3eadc M0 (CLKOUT, adcdav, davadc, adc0data, adc1data, adcsck, adcspsid, conad);
s3eprogamp M1 (CLKOUT, ampdav, davamp, ampcmd0, ampcmd1, ampsck, ampspid, csamp,
              sdamp);
s3edac M2 (CLKFX, dacdav, davdac, dacdata, dacaddr, daccmd, dacsck, dacspsid, csdac, clrdac);
genadcdac M3 (CLKOUT, SW0, ampdav, davamp, ampcmd0, ampcmd1, adcdav, davadc, adc0data,
             adc1data, dacdav, davdac, dacdata, dacaddr, daccmd);
dacs3edcm M4 (.CLKIN_IN(CCLK), .RST_IN(0), .CLKFX_OUT(CLKFX),
             .CLKIN_IBUFG_OUT(CLKINIBO), .CLK0_OUT(CLKOUT), .LOCKED_OUT(LOCK));

endmodule

```

The Verilog top module utilizes the `se3adc.v` and `se3progamp.v` modules for the ADC and programmable gain amplifier (PGA) and the `se3dac.v` and `dacs3edcm.v` modules for the DAC and the Digital Clock Manager (DCM) of the Xilinx Spartan-3E Starter Board. The DCM is an available Xilinx Architecture Wizard module which facilitates the embedded design. These four Verilog modules are FSM controllers for the ADC, PGA and DAC external peripherals and the soft-core peripheral DCM and are described in the text.

An output pin of one of the 6-pin peripheral ports is used to monitor the ADC conversion command signal (`conad`) and the sampling rate  $f_s \approx 282$  ksamples/sec here. This maximum sampling rate is far below the cutoff frequency of the anti-aliasing filter of the Spartan-3E Starter Board ADC (1.54 MHz). To avoid aliasing, an analog filter with a cutoff frequency  $f_{cutoff} < f_s / 2$  or  $f_{cutoff} < 141$  kHz would have been appropriate.

Although data from the two ADC channels are available simultaneously, the DAC can only accept one channel of data at this sampling rate. In addition, the Xilinx Spartan-3E Starter Board DAC shares access to the serial peripheral interface (SPI) bus with the ADC (and other external peripherals) which leads to SPI bus contention and an even lower sampling rate.



The Xilinx Spartan-3E Starter Board ADC outputs 14-bit two's complement binary data and the DAC requires 12-bit straight binary data. The conversion is accomplished by complementing the MSB of the ADC output (bit 13) as the sign bit. The twelve MSBs of the ADC output (bit 13 through bit 2) are

outputted to the DAC. Since the PGA inverts the analog input signal, the resulting straight binary data is complemented for comparison of the analog input and output signals on an oscilloscope.

**Listing 2** ADC-DAC system controller module for the Xilinx Spartan-3E Starter Board

// c 2008 Embedded Design using Programmable Gate Arrays Dennis Silage

```
module genadcdac (input genclk, SW0, output reg ampdav = 0, input davamp,
                 output reg [3:0] ampcmd0 = 1, output reg [3:0] ampcmd1 = 1,
                 output reg adcdav = 0, input davadc, input [13:0] adc0data,
                 input [13:0] adc1data, output reg dacdav = 0, input davdac,
                 output reg [11:0] dacdata, output reg [3:0] dacaddr = 0,
                 output reg [3:0] daccmd = 3);
```

```
reg [1:0] gstate = 0;          // state register
reg [13:0] value;
```

```
always@(posedge genclk)
begin
```

```
    case (gstate)
    0: begin
        ampdav = 1;          // programmable amplifier
        if (davamp == 1)
            gstate = 1;
        end
    1: begin
        adcdav = 1;          // ADC conversion
        if (davadc == 1)
            begin
                adcdav = 0;
                if (SW0 == 0)    // select one ADC channel
                    value = adc0data;
                else
                    value = adc1data;
                value[13] = ~value[13];    // sign bit
                dacdata[11:0] = ~value[13:2];    // inversion
                gstate = 2;
            end
        end
    2: begin
        dacdav = 1;          // DAC conversion
        if (davdac == 1)
            begin
                dacdav = 0;
                gstate = 1;
            end
        end
        default: gstate = 1;
    endcase
end
```

```
endmodule
```

The module genadcdac.v in the top module is a three state FSM controller for the ADC-DAC system. The ADC 14-bit signed 2's complement data is converted to a 12-bit straight binary data for the DAC. Slide switch SW0 selects one of the two available ADC channels. The DCM frequency synthesizer module

dacs3edcm.v inputs the 50 MHz crystal clock oscillator and outputs a  $50 \text{ MHz} \times 5/3 = 83.333 \text{ MHz}$  clock for the DAC. The DAC has a maximum SPI bus rate of 50 Mb/sec and is running at  $83.333/2 = 41.666 \text{ Mb/sec}$  from the controller.

For comparison a second straight-through ADC-DAC system utilizes the Digilent PmodAD1 ADC and Digilent PmodDA2 DAC and is described in the text. The PmodAD1 and the PmodDA2 hardware modules are connected to two of the 6-pin peripheral module ports of the Xilinx Spartan-3E Starter Board. An output pin of another of the 6-pin ports is again used to monitor the ADC conversion command signal and the sampling rate is  $f_s \approx 714 \text{ ksamples/sec}$ . This maximum sampling rate is above the cutoff frequency of the anti-aliasing filter of the Digilent PmodAD1 ADC (500 kHz). To avoid aliasing, an analog filter with a cutoff frequency  $f_{cutoff} < f_s/2$  or  $f_{cutoff} < 357 \text{ kHz}$  would be appropriate.

Although data from the two channels of the PmodAD1 ADC are available simultaneously, the PmodDA1 DAC can only accept one channel of data at this sampling rate. However, unlike the ADC and DAC of the Spartan-3E Starter Board, the PmodAD1 and PmodDA2 do not share an SPI data communication channel and can be processed in parallel. The output data of the ADC and the input data of the DAC are both 12-bit and utilize a straight binary data representation and no data conversion is thus required.

The PmodAD1 ADC and PmodDA2 DAC status signals from their datapath modules are processed in parallel by the controller and both provide a logic 1 signal to indicate that processing has completed. The DAC is outputting the previous analog signal sampled by the ADC while the ADC is obtaining the current analog signal sample. This illustrates wavefront processing and results in a sampling rate 2.5 times that of the straight-through ADC-DAC system using the Xilinx Spartan-3E Starter Board ADC and DAC. This primarily occurs because there is no SPI bus contention.

### **Xilinx FIR Compiler LogiCORE Block**

The straight-through ADC-DAC systems implemented with the controller and datapath modules can be easily modified to perform a DSP function. The Xilinx CORE Generator provides a FIR Compiler LogiCORE block which is a common interface for the design of FPGA resource efficient FIR digital filters with either multiply-accumulate (MAC) or distributed arithmetic (DA) architectures. The FIR Compiler LogiCORE block is evoked from within the Xilinx ISE project and provides design windows as described in the text.

The Xilinx LogiCORE block FIR Compiler provides from 2 to 1024 taps and integer signed or unsigned coefficients that range from 1-bit to 32-bit precision and is configured as a datapath module in Verilog. The signed or unsigned integer data widths can be as large as 32-bit and are accommodated by accumulator widths up to 74-bits.

The LogiCORE FIR Compiler here uses a coefficient file and implements a single-rate FIR MAC digital filter with one data channel (although up to 256 channels are possible), a clock frequency of 50 MHz and a sampling rate of 250 kHz. The clock and sample frequency parameters and the input and output bus widths specified are those utilized with the Spartan-3E Starter Board and the integral ADC and DAC. The FIR Compiler has the capability to reload coefficients on a separate data bus, change the digital filter function with an external command and utilize different coefficient sets for each data channel.

The FIR Compiler digital filter is implemented as part of a DSP system. A portion of the Verilog top module s3eadcfirdac.v is given in Listing 3. The complete module is similar to that of the top module in Listing 1. Additional details of the implementation of the DSP system are provided in the text. The six Verilog modules operate in parallel in the top module. An output pin of a 6-pin port is again used to monitor the ADC conversion command signal (conad). The estimated cycle latency of the FIR filter from the FIR Compiler LogiCORE block design windows is  $16 \times T_{clock} = 0.32 \mu\text{sec}$  since  $f_{clock} = 50 \text{ MHz}$ .

The actual latency of the digital filter can be determined by measuring the time interval between the assertion of the new data input signal (nd) and the appearance of the data ready output signal (rdy) of the FIR Compiler datapath module. The latency is measured to be 0.39  $\mu\text{sec}$ , which is less than the available

in-sequence computation time of 0.46  $\mu$ sec, since the sampling period  $T_s = 4 \mu$ sec and the sampling period for the straight-through or ADC-DAC system in Listing 1 is  $T_{ADC-DAC} \approx 3.54 \mu$ sec (282 ksamples/sec).

The controller module genadcfirdac.v utilizes a five state FSM controller for the FIR Compiler LogiCORE block DSP system. The FIR filter LogiCORE block datapath ready for data signal (rfd) sets the sampling rate of 250 ksamples/sec. The controller module responds with the FIR filter LogiCORE block new input data available signal (nd). The data ready status signal (rdy) indicates that the 25-bit FIR digital filter data (firdout) is available.

**Listing 3** FIR Compiler LogiCORE block DSP system top module for the Xilinx Spartan-3E Starter Board (portion)

```
// c 2008 Embedded Design using Programmable Gate Arrays Dennis Silage
```

```
module s3eadcfirdac (input CCLK, SW0, SW1, SW2, SW3, SPIMISO, output SPIMOSI, SPISCK,
                   output DACCS, DACCLR, SPISF, AMPCS, AMPSD, ADCON, SFCE,
                   output FPGAIB, JC1, JC2, JC3, JC4);

s3eadc M0 (CLKOUT, adcdav, davadc, adc0data, adc1data, adcsck, adcspod, conad);
s3eprogamp M1 (CLKOUT, ampdav, davamp, ampcmd0, ampcmd1, ampsck, ampspid, csamp,
              sdamp);
s3edac M2 (CLKFX, dacdav, davdac, dacdata, dacaddr, daccmd, dacsck, dacspsid, csdac, clrdac);
genadcfirdac M3 (CLKOUT, SW0, SW1, ampdav, davamp, ampcmd0, ampcmd1, adcdav, davadc,
               adc0data, adc1data, dacdav, davdac, dacdata, dacaddr, daccmd, sclr, rfd, rdy, nd,
               firdout, firdin, SW2, SW3);
dacs3edcm M4 (.CLKIN_IN(CCLK), .RST_IN(0), .CLKFX_OUT(CLKFX),
             .CLKIN_IBUFG_OUT(CLKINIBO), .CLK0_OUT(CLKOUT),
             .LOCKED_OUT(LOCK));
fir1 M5 (.sclr(sclr), .clk(CLKOUT), .nd(nd), .din(firdin), .rfd(rfd), .rdy(rdy), .dout(firdout));
```

### Xilinx PicoBlaze Soft Core Processor

The availability of the soft core processor within the FPGA has significantly altered the ambiance of embedded design. Although hard core processors are available for some FPGAs, the soft core processor seemingly has more flexibility including the capability of utilizing several within the embedded design. Since the soft core processor can be modified easily, new and specialized architectures can also be readily developed.

The Xilinx 8-bit PicoBlaze soft core processor is quite resource efficient and is suitable for debouncing and reading slide switches and push buttons and external peripherals with a low data throughput, such as the LCD of the Spartan-3E Starter Board. The Verilog controller and datapath construct is seemingly more suited to high throughput internal soft core peripherals and external peripherals, such as an ADC, the FIR Compiler LogiCORE block and a DAC in a DSP system. However, the Xilinx PicoBlaze soft core processor and the Verilog controller and datapath construct can also interact synergistically through port signals. An embedded design then would consist of processing elements appropriate for the task.

### Summary

This short description illustrates the concept and utility of the FSM and controller-datapath Verilog modules for real-time processing in DSP on the Spartan-3E Starter Board. The text *Embedded Design Using Programmable Gate Arrays* features complete Xilinx ISE projects for the PS/2 mouse and keyboard, PGA, ADC, DAC of the Xilinx Spartan-3E Starter Board and reference material for the Verilog hardware description language, the Xilinx EDA, DSP, digital communications, digital control and the Xilinx PicoBlaze soft core processor.

This 320 page, soft cover text is available from Bookstand Publishing:  
<http://www.bookstandpublishing.com/m/dennissilage>  
These complete Xilinx ISE project files for the Xilinx Spartan-3E Starter Board and the Digilent Basys Board are available for download (password protected) as described in the text.



## **Chapter 1 Verilog Hardware Description Language**

- Programmable Logic Devices
- Hardware Description Languages
- Verilog Syntax and Concepts
- Structural Models in Verilog
- Behavioral Models in Verilog
- Finite State Machines
- Controller-Datapath Construct
- C to Verilog Translation
- Arithmetic Functions
- Programmable Gate Array and Microprocessor Comparison

## **Chapter 2 Verilog Design Automation**

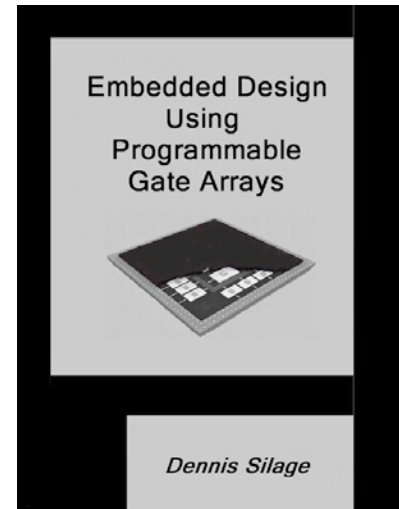
- Xilinx ISE WebPACK
- Xilinx CORE Generator
- Xilinx Floorplanner
- Xilinx Simulator
- Xilinx Verilog Language Templates
- Xilinx Architecture Wizard
- Xilinx LogiCORE Blocks
- Warnings and Errors in Synthesis

## **Chapter 3 Programmable Gate Array Hardware**

- Evaluation Boards
- Selection of an Evaluation Board
- User Constraints File
- Hardware Components and Peripherals
  - Crystal Clock Oscillator
  - Light Emitting Diodes
  - Push Buttons and Slide Switches
  - Rotary Shaft Encoder
  - Seven Segment Display
  - Liquid Crystal Display
  - PS/2 Keyboard Port
  - PS/2 Mouse Port
  - Digital-to-Analog Converter
  - Analog-to-Digital Converter
  - Auxiliary Ports and Peripherals

## **Chapter 4 Digital Signal Processing, Communications and Control**

- Sampling and Quantization
- Discrete Time Sequences
- Discrete Frequency Response
- Analog Output
- Digital Signal Processing Embedded System



- IIR Digital Filter
- FIR Digital Filter
- FIR Compiler LogiCORE Block
  - Digital Signal Processing System
- Sine-Cosine Look-Up Table LogiCORE Block
  - DTMF Generator
- Direct Digital Synthesis Compiler LogiCORE Block
  - Frequency Generator
  - Frequency Shift Keying Modulator
  - Phase Shift Keying Modulator
  - Quaternary Phase Shift Keying Modulator
- Linear Finite Shift Register
- Data Communication
  - RS-232 Standard
  - Manchester Encoder-Decoder
- Digital Control
  - Pulse Width Modulation
  - DC Servomotor Speed Control

**Chapter 5 Embedded Soft Core Processors**  
Programmable Gate Array Processors  
Xilinx PicoBlaze Development Tools  
Xilinx PicoBlaze Processor Architecture  
Xilinx PicoBlaze Reference Projects  
Soft Core Processors and Peripherals

