A 64 MHz RISC Coprocessor Using the A1460 and VHDL Entry

Warren Miller
Product Planning Manager, Actel Corporation

Introduction

The Actel A1460 is the only Field Programmable Gate Array (FPGA) offering high capacity and high performance simultaneously. Additionally, the gate array architecture of the A1460 makes VHDL entry very efficient in terms of speed and capacity. This application note describes the implementation of a specialized coprocessor for a RISC CPU running at 64 MHz using VHDL entry.

RISC processors are used in a variety of high-performance applications, but even these speedy devices run out of processing power in some applications, or else the additional costs required to "pump up" the processor with faster memory systems or faster processor speed are prohibitive. In many cases, an FPGA can improve performance by off loading processor-intensive portions of an algorithm for direct hardware implementation. A RISC-based communications packet processor demonstrates this concept.

Communications Processor

In its simplest form, a packet processor is responsible for receiving data packets from a communications network, checking packets for correctness, and directing data to the correct port. A RISC processor could do all these functions, but matching the data rate would require a large portion of the processor's bandwidth. An FPGA can implement the packet checking portion of the algorithm and accelerate performance above that of a processor alone. The additional processor cycles thus made available can be used to improve performance of the higher level portions of the algorithm, adding intelligence and further improving throughput.

Figure 1 shows a block diagram of the communications processor. The RISC processor and the main memory provide the high-level control of the communications network. Packets received from the four communications lines are routed by the processor to the appropriate output port. The RISC processor establishes the connections between the input and output ports based on traffic levels and bandwidth limitations. It also measures traffic statistics to help determine the best solution of input and output ports. In addition, during reset and during times of low network activity, the processor supports diagnostics.

The FPGA implements the low level portions of the algorithm. Four data channels of 16 bits are received by the FPGA and four output ports are available for output data. The 16-word data packets are preceded by a command word that indicates the destination address of the packet and delineates the start of data. The FPGA counts each data word as it is received to ensure that the full 17 words exist in each packet. The 17th word is a simple CRC (an exclusive OR of each data word). The received data is selected for output at any of the four output ports.

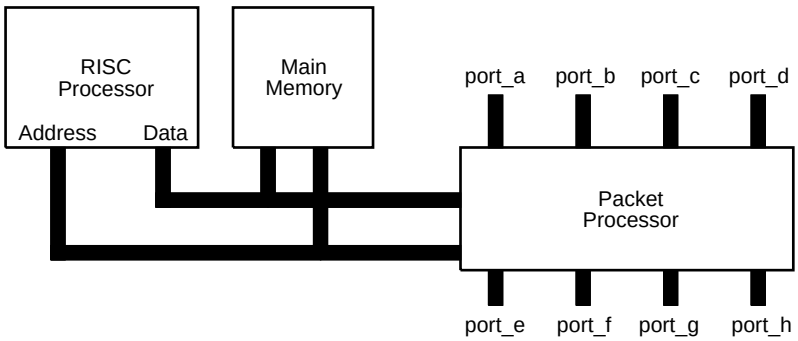

Figure 1 • *System Level Block Diagram*

Detailed FPGA Description

A block diagram of the packet processor FPGA is shown in Figure 2. The bus interface state machine manages the transactions with the RISC processor, thus allowing data registers to be read, status registers to be written to, and initialization commands to be executed. The packet processor state machine manages data flow. In normal operation, data flows from input to output without interrupting the RISC processor. If the number of data words is incorrect or a CRC error is generated, the RISC processor is informed of the error by the packet processor. The packet multiplexers are the main data path for the packer processor. Each of the four packet multiplexers implement the same functions. Input data is received by the packet multiplexer, command words detected, data words counted, CRC computed, and output data stream selected.

Referring to packet multiplexer A, input port A is the 16-bit input port, with associated data_valid, control_valid, and receive_valid signals. Input command words are separated from data words by the data_valid signal. Ports B, C, and D are the terminal versions of the data packet words that can be selected (in addition to A) by output port E by selecting signals S1 and S0.

FPGA VHDL Description

The complete packet processor design is described in a few pages of VHDL. Figure 3 gives the entire code for the packet multiplexer. The entity section defines the input and output signals of the packet multiplexer and the architecture section defines the operation of the packet multiplexer. Each of the two main parts of the design are described in separate block sections: input registers (inregs), output registers, the CRC register, and the shift register used to count data packets. The dffc and dff–v procedures call flip-flop routines from the library, simplifying the code for sequential elements (–v is used on a bit vector, 16 bits in this case). Notice also how the shift operation and the CRC operation are executed. These powerful operations require only a single line of code. As a result, a large amount of design work can be done very quickly in VHDL. The out_mux block implements the output select function so that any of the input ports can be routed to the output port.

The packet multiplexer description is called four times in the top-level design with different input and output signals to implement all four ports. The state machine section is added to complete the entire design. The design is targeted for the ACT 3 library and an Actel netlist is created. Figure 4 shows the results of running the complete design through the ACT'x'press software.

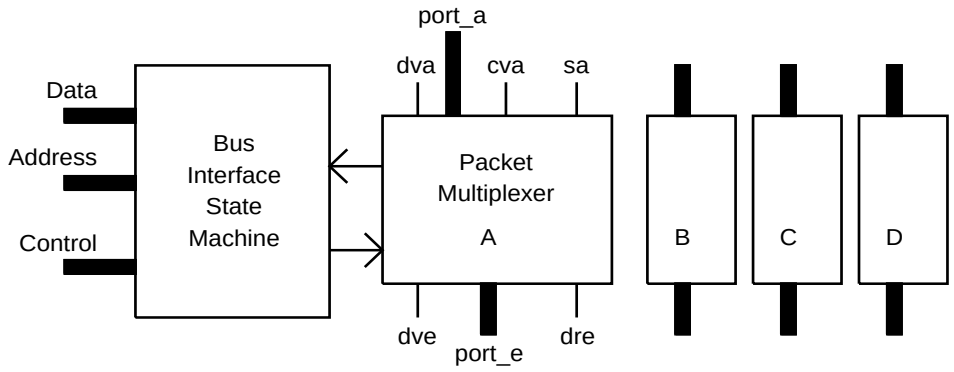

Figure 2 • *System Level Block Diagram*

```
library exemplar;
use work.exemplar.all;

entity pack_mux is
       port ( a_port, b, c, d : in bit_vector(0 to 15)
              e_port, a_reg : out bit_vector(0 to 15);
              pack_start, crc_error, pack_end : out bit;
              sa, dve : out bit;
              pack_addr : out bit_vector(0 to 1)
              clock, s2, s1, en_crc, cva, dva, clr, dre : in bit
       );
end pack_mux;

architecture packet of pack_mux is
       signal e, s, crc, s_reg, crc_reg : bit_vector(0 to 15)
       signal cvalid_reg, dvalid_reg, dre_reg : bit
       begin
       inregs : block
              begin
              dffc(cva, clr, clock, cvalid_reg);
              dffc(dva, clr, clock, dvalid_reg);
              dffc(en_crc, clr, clock, a_reg);
              dff_v(a_port, clock, a_reg);
              pack_start <='1' WHEN cvalid_reg = '1'1 and a_reg(15) = '0'
                     ELSE '0';
              pack-addr <= a_reg(1 downto 0)
              dffx_v(s, clr, clock, s_reg);
              s <= s_reg(14 downto 0) & '1' WHEN dvalid_reg = '1'
                     ELSE s_reg;
              ppack_end <= s_reg(15);
              dffc_v(crc, clr, clock, crc_reg);
              crc <= crc_reg xor a_reg WHEN en_crc = '1'
                     ELSE crc_reg;
              crc_error <= '0' WHEN crc_reg = x"0000"ELSE'1';
              end block;
       out_mux : block
              Begin
              dff_v(e, clock, e_port);
              dffc(dre, clr, clock, dre_reg);
              e <= e WHEN dre_reg = '1'ELSE
                     d WHEN (s1 = '1' and s2 = '1') ELSE
                     c WHEN (s1 = '1' and s2 = '0') ELSE
                     b WHEN (s1 = '0' and s2 = '1') ELSE
                     a_reg;
              end block;
       end packet;
```

Figure 3 • *VHDL Source for Packet Multiplexer*

FPGA Capacity and Performance Results

As indicated in Figure 4, the estimated performance of the design is 47 MHz (1/21.1 nanoseconds) and capacity of the design is 801 modules. The Actel place and route tool was run—the timer results for the longest path in the design are shown in Figure 5. The performance of the design is 81 MHz (1/12.3 ns), well over the goal of 64 MHz.

Conclusion

As shown in this example, a high-performance data path design is easily implemented using VHDL and the Actel A1460A. The high-capacity, high-speed, and fine-grained architecture of the Actel ACT 3 device family and the efficient results obtained from the ACT'x'press software make them ideal for time-to-market critical designs.

```
Exemplar Logic Synthesis System
Sun Jan 23 20:53:41 1994

Recourse Use Estimate

Design: pack_all
Technology: act3
File: PACK_MUX.VHD
Area: 801.0
Critical Path: 21.2 ns
```

Figure 4 • *Synthesis Results for Packet Processor Design*

```
; 1st longest path to $1I566:D (rising) (Rank: 0)
; Total    Delay    Typ      Load     Macro    Start pin        Net name
; 12.3     0.8      Tsu      0        DFC1B    $1I566:D
; 11.5     0.0      Tpd      1        AX1C     $1I315:A         PACK_MUX/CRC_0
; 11.5     8.7      Tcq      3        DF1      $1I549:CLK       A_REG0_internal
; 2.8      2.8      Psk      4                 $1I566:CLK       CLOCK_internal
```

Figure 5 • *Longest Path Timing Results (CRC Register)*