V8-uRISC 8-bit RISC Microprocessor

February 8, 1998 | Product Specification

VAutomation, Inc.

20 Trafalgar Square
Nashua, NH 03063
Phone: +1 603-882-2282
Fax: +1 603-882-1587
E-mail: sales@vautomation.com
URL: www.vautomation.com

Features

- RISC architecture for low gate count and high performance
- Single Cycle instruction execution for all register to register opcodes
- High Code density, many opcodes are a single byte
- 33 opcodes, 4 addressing modes, 2 user defined opcodes
- 8-bit ALU
- 64K byte addressing capability
- Two banks of 8-bit general purpose registers
- 16-bit program counter and stack pointer
- Seven maskable interrupts, one non-maskable
- Assembler, debugger and simulator available now, C Compiler (May '98)
- IntelliCore™ Prototyping System available for evaluation and rapid product development
- Includes VHDL/Verilog source code for the following peripherals:
  - UART
  - DMA controller
  - I2C EEPROM interface
  - P1284 parallel interface
  - Page register

AllianceCORE™ Facts

Core Specifics[2]	
Device Family	XC4000E
Macrocells Used	268
IOBs Used	54[1]
CLKIOBs Used	1
System Clock fmax	15.9 MHz
Device Features Used	SelectRAM

Supported Devices/Resources Remaining[2]		
	I/O	**CLBs**
XC4013E-1 PQ160	74[1]	308

Provided with Core	
Documentation	Reference Manual
Design File Formats	XNF, EDIF, VHDL or Verilog netlist
Constraint Files	Example .ucf
Verification Tool	Test Bench
Schematic Symbols	None
Evaluation Model	None
Reference designs & application notes	Includes VHDL/Verilog source for UART, DMA controller, I2C EE-PROM I/F, P1284 I/F, page register
Additional Items	IntelliCore™ Prototype system available extra

Design Tool Requirements	
Xilinx Core Tools	Alliance 1.3
Entry/Verification Tool	FPGA Synthesis tool required for supplied peripherals; Model Tech for simulation

Support
Support provided by VAutomation.

Notes:
1. Assuming all core signals are routed off-chip.
2. For CPU and register file only. Figures do not include peripherals.

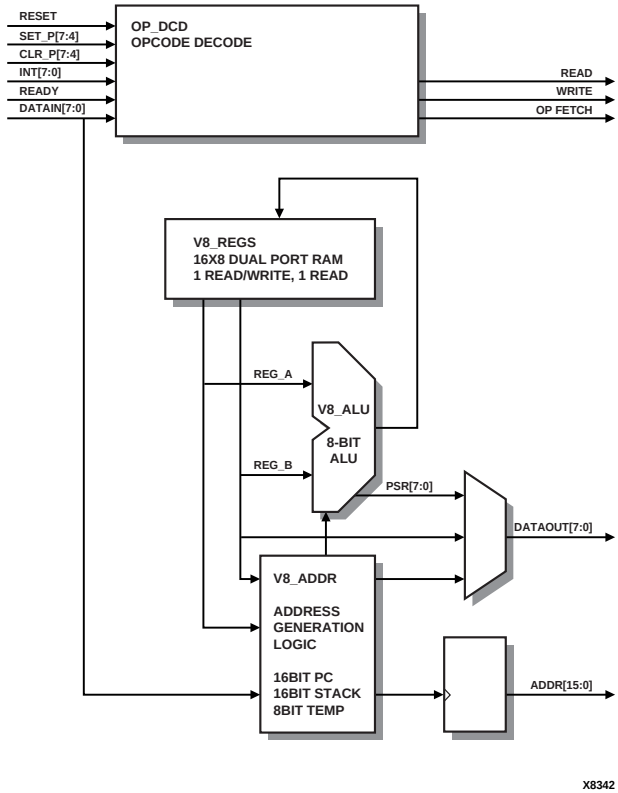

Figure 1: V8-uRISC 8-bit RISC Microprocessor Block Diagram

Applications

- 8-bit processing applications which need a combination of small size, good performance and low power
- Custom microcontroller for any embedded application

General Description

The V8-uRISC 8-bit RISC microprocessor is a general purpose processor core designed and optimized specifically for programmable logic. It combines a small gate count with single clock cycle execution for many instructions to deliver a high performance 8-bit microprocessor with a very small footprint.

The V8-uRISC is delivered with a set of standard peripherals including a timer, UART, DMA controller, I2C EEPROM interface, P1284 parallel interface, and page register design to expand the address space to 8 megabytes. The peripherals are delivered in non-optimized VHDL/Verilog source code. The V8-uRISC and its' peripherals allow high performance custom microcontrollers to be implemented within Xilinx FPGAs.

Functional Description

The V8-uRISC core is partitioned into modules as shown in Figure 1, and described below.

Instruction Decode (OP_DCD)

The Instruction Decode block controls the operation of the V8-uRISC. Opcodes are fetched from memory and loaded into the Opcode Register. The opcode is then decoded and the appropriate control signals are output. The control signals depend on the opcode and the number of clock cycles that have been executed for the particular opcode.

Most of the control outputs are decodes and thus may glitch and have a significant amount of delay before settling to their final state. However, WRITE comes directly from a Flip-Flop to insure there are no glitches.

The READY signal will hold the current state of the CPU and deactivates many of the register load control signals. The INT and RESET lines force an INT opcode to be loaded into the opcode register. Special processing is performed depending on the type of interrupt.

Arithmetic Logic Unit (V8_ALU)

The ALU block performs the 8 bit arithmetic and logical operations. The inputs to the ALU come directly from the Register File or from the TEMP register in the V8-uRISC_ADDR module. Certain constants such as 0 and 1 can also be selected as inputs to the ALU for increment and decrement operations.

The ALU includes bit-manipulation logic to allow testing, setting and clearing of a single bit of a byte. This is a common function in embedded microcontrollers.

The Processor Status Register (PSR) shown in Table 1, is also located within the V8_ALU block. Most opcodes pass data through the ALU so that the appropriate PSR flags can be updated. A logic 1 on the SET_P signals will cause the corresponding bit in the PSR register to be set; a logic 1 on the CLR_P pin will clear the corresponding PSR bit to zero. These signals are typically connected to IO device status lines that allow the V8-uRISC to quickly test and branch on these bits.

Table 1: Program Status Register (PSR)

Bit	Name	Operation
7	P7	FLAG bit, Set by SET_P[7], Cleared by CLR_P[7]
6	P6	FLAG bit, Set by SET_P[6], Cleared by CLR_P[6]
5	P5	FLAG bit, Set by SET_P[5], Cleared by CLR_P[5]
4	P4	FLAG bit, Set by SET_P[4], Cleared by CLR_P[4]
3	I	Interrupt MASK bit (1=ignore interrupts, 0=enable interrupts)
2	N	Negative bit (1 when ALU result has bit 7=1)
1	C	Carry bit (NOT borrow for subtract)
0	Z	Zero bit (1 when ALU result is zero)

Register File (V8_REGS)

The Register File is a 16x8, dual read, single write register file with two 8x8 banks. It is implemented as a dual port RAM with one port being read/write and the other port being read only.

The registers in the first bank are referred to as R0 through R7, as shown in Table 2. R0 can be thought of as the accumulator as it is the default source/destination for most opcodes. Typically R4 through R7 are used as index registers into RAM although they can be used for any purpose.

Table 2: Register File Use

Register	Typical Purpose
R0 [7:0]	Accumulator
R1 [7:0]	General Purpose Registers
R2 [7:0]	
R3 [7:0]	
R4 [7:0]	Register pairs form 16-bit addresses for indexed and offset addressing modes
R5 [7:0]	
R6 [7:0]	
R7 [7:0]	

The Interrupt Mask (I) bit of the PSR is used to select the second bank of registers which are used only during interrupt processing. This allows interrupts to be serviced without having to save and restore all of the registers, saving several microseconds for each interrupt.

Additional banks of registers can be added to allow rapid changes in context or for any purpose where the current state of the CPU needs to be retained. Typically the bank select bits are sourced from the FLAG bits in the PSR.

Address Generation Logic (V8_ADDR)

The address generation logic module consists of the 16 bit program counter, the stack pointer and an 8 bit temporary register. The output of the V8_ALU block is actually the address for the next clock (ADDR_D). The ADDR_D bus is then registered external to the V8_ALU block so that the address can come directly from a Flip-Flop.

The performance limiting paths in a V8-uRISC microcontroller design are typically from the ADDR bus, through address decodes to external IO, RAM or ROM and then into the DATAIN bus of the V8-uRISC. Providing the ADDR bus directly from Flip-Flops saves time in this critical path. Critical address decodes can also be done a cycle early by decoding the ADDR_D bus and then registering the address decode signal.

The 16 bit Program Counter (PC) provides the address from which opcodes are fetched. The counter is incremented whenever an opcode is fetched or the data bytes of an opcode are read. The counter is reloaded whenever a Jump or Branch opcode is executed or when processing an interrupt.

The stack pointer (SP) is a 16 bit up/down counter. It provides the address when data is pushed or popped from the stack. The SP points to the byte that was most recently pushed onto the stack. Typically the upper 8 bits of the SP are set to 0x01, but they can easily be set to any desired location in RAM. With the source-code version of the core, the counter can be reduced to less than 16 bits by simply changing a few constants in the HDL code to reduce the gate count of the V8-uRISC CPU.

DATAOUT Multiplexor

The Data Out Mux is an 8-bit 3 to 1 mux. It routes data from the register file or the high and low halves of the PC. The PC is required to be output when executing an interrupt or jumping to a subroutine.

Core Modifications

An optional HDL source license to the V8-uRISC allows the user to make unlimited modifications to the core. The core has been specifically designed to allow the implementation of two user defined opcodes.

Analysis of your application code can typically identify one or two functions that require most of the CPU bandwidth. By coding these functions as a single opcode, performance can be significantly increased without increasing the clock rate. This in turn can keep the power requirements of your product low.

Applications for custom opcodes include: 512 or 1024 bit math for high speed Public Key Encryption algorithms, multiply/accumulate for DSP applications, table lookups for decoding, block moves, and trigonometric functions commonly used in GPS applications.

Other modifications include adding register files, or reducing the size of the program counter. Consult VAutomation for any of these modifications.

Pinout

Signal names for the core are shown in Figure 1, and described in Table 3.

Verification Methods

VAutomation prototypes and verifies all of its cores using Xilinx devices. The V8-uRISC core has been implemented within the Xilinx-based IntelliCore™ Prototyping System (IPS) and extensively tested in real-world applications (e.g. command processing for USB audio). The IPS is available from VAutomation in support of a V8-uRISC evaluation and early software testing while hardware design is underway. Please consult the IntelliCore™ Prototyping System datasheet for more details.

The V8-uRISC has also been tested in simulation using a test bench which implements the core as a XC4013-based microcontroller. This microcontroller is identical to one implemented and tested within the IPS. The test bench also consists of a self-checking assembly program (Intel hex format object file) that is meant to be run on the test bench. The test bench is supplied with the core.

Table 3: Core Signal Pinout

Signal	Signal Direction	Description
ADDR[15:0]	Output	Address Bus
DATAIN[7:0]	Input	Data Input Bus
DATAOUT[7:0]	Output	Data Output Bus
READ	Output	Read Cycle - High when data is being loaded from the DATAIN bus.
WRITE	Output	Write Cycle - High when data is valid on the DATAOUT bus.
CLK	Input	Clock - All Flip-Flops clock on the rising edge of the clock; uses 1 FPGA CLKIOB pin.
RESET	Input	Reset - Must be asserted for at least 3 rising edges of CLK.
INT[7:0]	Input	Interrupt Request - INT[0] is Non-maskable and is generally reserved for JTAG debugging purposes.
READY	Input	Ready - Freezes the current state when low, operation completes when high. Must be high when RESET is asserted.
OP_FETCH	Output	Opcode Fetch Cycle. Typically used for single step debugging.
CLR_P[7:4]	Input	Clear PSR bit
SET_P[7:4]	Input	SET PSR bit
PSR	Output	Program Status Register

Recommended Design Experience

Users should be familiar with Xilinx design flows and tools, have experience with microprocessor systems and be skilled in assembly and/or C language programming.

Available Support Products

The V8-uRISC is supplied with an assembler (DOS application). An Instruction Level Simulator and debugger are available now. The debugger operates seamlessly with the Instruction Level Simulator or the IPS. A C Compiler will be available in May '98. The V8-uRISC is also supported by the IntelliCore™ Prototyping Systems (IPS) which is available separately. Please consult the IPS datasheet for more details.

Ordering Information

The V8-uRISC core is provided under license from VAutomation for use in Xilinx devices in either netlist or HDL source form. Please contact VAutomation at the location listed on the first page of this datasheet for ordering details.

Related Information

Xilinx Programmable Logic

For information on Xilinx programmable logic or development system software, contact your local Xilinx sales office, or:

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
URL: www.xilinx.com

For general Xilinx literature, contact:

Phone: +1 800-231-3386 (inside the US)
+1 408-879-5017 (outside the US)
E-mail: literature@xilinx.com

For AllianceCORE™ specific information, contact:

Phone: +1 408-879-5381
E-mail: alliancecore@xilinx.com
URL: www.xilinx.com/products/logicore/alliance/tblpart.htm