Lightfoot 32-bit Java Processor Core

August 24, 2001 Product Specification

Digital Communication Technologies

Unit 5, Windmill Business Village
Brooklands Close, Sunbury
Middlesex TW16 7DY
Phone: +44 (0)1932 784 858
Fax: +44 (0)1932 782 811
E-mail: info@dctl.com
URL: www.dctl.com

Features

- 32-bit Harvard architecture CPU
- Simple 3-stage pipeline
- Stack-based design
- 32-bit data memory interface, 24-bit address
- 8-bit program memory interface, 24-bit address
- Optional unified memory interface
- No microcode memory
- Small runtime program memory footprint
- Efficient hardware Java byte-code execution
- Multi language development in Java/C
- Low interrupt latency
- Single clock domain

AllianceCORE™ Facts

Core Specifics	
See Table 1	
Provided with Core	
Documentation	Data sheet, SDK
Design File Formats	EDIF netlist
Constraints File	.ucf
Verification	XC2S200-based development board, Verisity Specman Elite Testbench, VHDL Testbench
Instantiation Templates	VHDL, Verilog
Reference designs & application notes	Available
Additional Items	XC2S200-based development board, Java/C run time environment, RTOS, TCP/IP, Java/C compiler, simulator/debugger
Simulation Tool Used	
Model Tech ModelSim V5.4	
Support	
Support provided by Digital Communication Technologies	

Applications

- Internet appliances (TCP/IP)
- Multimedia controllers (HAVi, MHP)
- Automotive (D2B)
- Network/switch processors
- Set-top box (MPEG)
- Networked embedded controllers
- Terminals, PDAs, TVs, Printers, copiers
- Smart Cards

.

Table 1: Core Implementation Data

Supported Family	Device Tested	CLB Slices	Clock IOBs[1]	IOBs[1]	Performance (MHz)	Xilinx Tools	Special Features
Virtex™-II	2V1000-5	1711	1	218	40	F3.1i	1 Block RAM
Virtex-E	V400E-8	1710	1	218	33	F3.1i	2 Block RAMs
Spartan™-II	2S200-6	1710	1	218	31	F3.31i	2 Block RAMs

Notes:
1. Assuming all core I/Os are routed off-chip.

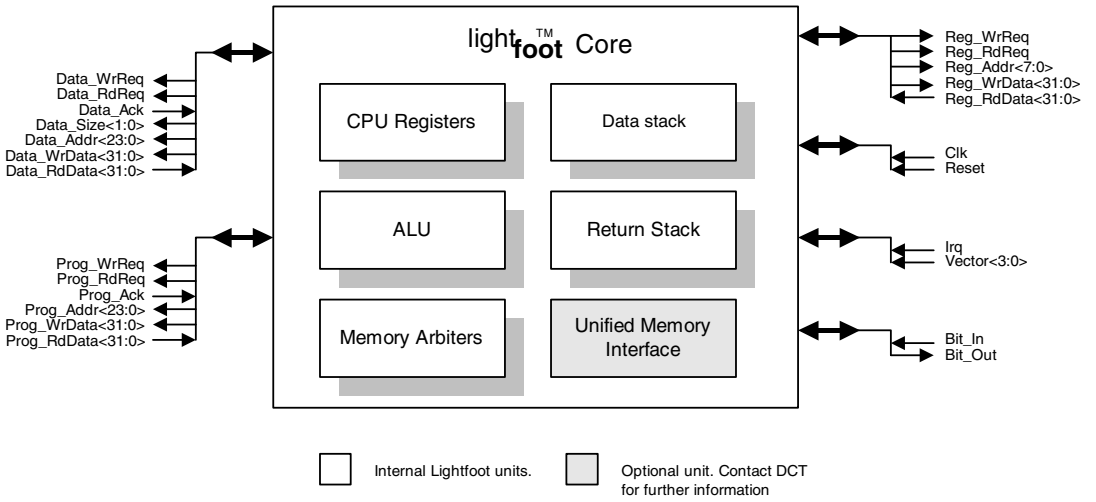

Figure 1: Lightfoot 32-bit Java Processor Block Diagram

General Description

Offered as part of the Xilinx AllianceCORE intellectual property program, the 32-bit LIghtfoot processor operates from a tiny memory footprint, providing an ideal design solution for embedded system OEMs. This implementation of the core can provide performance levels of over 30 and 40 MIPs respectively from the Spartan-II and Virtex-II FPGA families, thereby allowing designers to realize powerful single-chip control solutions for connected embedded systems without the cost and time associated with ASIC development.

The Lightfoot core is compatible with KVM, J2ME, CLDC and CDC and executes native Java bytecodes directly in hardware. It provides an exceptionally efficient means of exploiting the benefits of the language in real-time systems. In addition, it supports C, which provides a practical means of migrating legacy code to new-generations of embedded system architectures.

Purpose-designed for embedded systems, the DCT Lightfoot core uses less than 30,000 gates in its conventional off-the-shelf chip form, and requires just 1710 'CLB slices' of Xilinx logic in its IP form. This is extremely compact, equating to around 3% of a top-of-the-line Virtex-II FPGA.

Functional Description

The Lightfoot 32-bit core is a hybrid 8/32-bit processor based on a Harvard architecture.

The memory referenced by programs falls into two categories: Program Memory and Data Memory.

Program memory is 8-bits wide and is used to store program instructions and constant data.

Data memory is 32-bits wide and is byte addressable. Words (32-bits wide) must be word aligned, halfwords (16-bits wide) must be halfword aligned. The memory interface detects illegal accesses and signals a Bus Error trap.

The key blocks of the core, the Control Unit, ALU, Data and Return Stacks, CPU and Parameter Registers are described below.

Control Unit

The Control Unit is responsible for fetching, decoding and sequencing the execution of instructions in the processor. It also contains modules for implementing run-time checks and handling traps.

ALU

The *ALU* is a “traditional” 32-bit design, featuring a 32-bit barrel shifter and a 2-bit multiply step unit (allowing a 32 x 32 bit multiply to execute in 16 cycles), in addition to the usual arithmetic and logic capabilities.

Data and Return Stacks

The Data Stack plays the role of a register bank in traditional architectures. It consists of a hardware part (implemented as a bank of 8, 32-bit on-chip registers) and a memory extension. The memory extension is supported by a dedicated register (the EP or Extension Pointer) together with a fill/spill circuit. The data stack is used to hold temporary data; it is not used to implement the stack frame, for which special support is provided. The top elements of the Data Stack are connected to the inputs of the ALU.

The Return Stack plays a threefold role in the processor: it holds return addresses for subroutines, its top-of-stack element is used as an index register to access program memory, and it can be used as an auxiliary stack for programs.

The organization of the Return Stack is similar to the organization of the Data Stack, in that it also consists of a hardware part and a memory extension. The hardware part of the Return Stack consists of four 32-bit registers. The memory extension is supported by a dedicated CPU register (called the REP Return Extension Pointer) and a fill/spill circuit.

CPU and Parameter Registers

The Register Bank resides in a 256-word long register space. The sixteen addresses at the bottom of the register space are reserved for the various CPU registers (such as the Processor Status Word (PSW), the stack extension pointers, multiplier partial product, constant and parameter pool pointer registers), with the remainder available for interfacing to system peripherals such as memory management units or cryptographic co-processors. The register bank contains four Parameter cache registers that hold the first four method parameters.

The Instruction Set

The processor architecture specifies three different instruction formats, called IF0, IF1 and IF2. The first (IF0) is the format of the (128 possible) soft bytecode instructions. The second (IF1) is used by the (64) non returnable instructions, and the third (IF2) is used by 32 single-byte instructions that can be folded with a return operation.

Execution of a soft bytecode instruction causes the processor to branch to one of 128 locations in low program memory, where the implementation of the soft bytecodes resides. This operation is performed by the Fetch Unit and carries a single cycle overhead on the current 3-stage pipeline organization. The address of the following instruction is pushed on to the Return Stack.

The opcode field of the remaining Lightfoot instructions is always 8 bits wide. Some instructions can be followed by a single 8-bit immediate operand, which is interpreted as a signed or unsigned value, depending on the instruction.

If an instruction, which uses an immediate operand is prefixed by the WIDE opcode, the immediate operand is taken to be 16-bits wide. The resulting 16-bit value is interpreted as a signed or unsigned value, depending on the particular instruction. Lightfoot instructions (assuming that WIDE is a part of the following instruction) can thus be 8, 16 or 32 bits wide.

The 32 format IF2 instructions have a return bit. This bit is tested by the fetch unit, and if set, it causes the value of the program counter register to be loaded with the value popped from the Return Stack. This mechanism implements a zero-overhead return feature of the processor. The low overhead of invoking instruction sequences defined by the soft bytecodes allow the implementation of extremely effective interpretative machines.

From a functional point of view, the various machine instructions fall into one of the following groups:

- Arithmetic-Logic
- Data Stack Manipulation
- Interstack
- Constants
- Local Variables
- Parameter Registers
- Program Memory Access
- Data Memory Access
- Control Flow
- Register

The instruction set was chosen to map well onto the JVM. In particular, all simple JVM instructions that can be efficiently executed in hardware have Lightfoot counter-parts.

Special instructions are provided for supporting the complex JVM bytecodes. This group includes instructions for creating stack frames.

To allow system-level code to be written, the instruction set contains low-level memory and device register access instructions.

Verification Methods

The Lightfoot CPU has been verified on Spartan-II, Virtex and Virtex-II devices. The low-cost evaluation platform contains a typical microcontroller configuration in a Spartan-II device that can be used to develop and run code in real time. Standard configurations are also available for Xilinx AFX cards.

For simulation based verification an extensive test suite is available for Verisity Specman, VHDL users. Simulation testbenches allow target system environments to be quickly developed. Customers wishing to use a Verilog design flow should contact DCT.

Recommended Design Experience

Users should be familiar with digital electronic system design, HDL design flows and Xilinx FPGA technologies.

Pinout

The DCT Lightfoot core has not been fixed to a specific FPGA I/O, thereby allowing flexibility with a user's application. The pin signal names are shown in Figure 1 are described in Table 2.

Table 2: Core Signal Pinout

Signal	Signal Direction	Description
Clk	Input	System clock
Reset	Input	System reset
Prog_RdData [7:0]	Input	Program memory read bus
Prog_WrData [7:0]	Output	Program memory write bus
Prog_Add[23: 0]	Output	Program memory address bus
Prog_RdReq	Output	Program memory read request
Prog_WrReq	Output	Program memory write request
Prog Ack	Input	Program memory acknowledge
Data_RdData [31:0]	Input	Data memory read bus
Data_WrData [31:0]	Output	Data memory write bus
Data_Size[1: 0]	Output	Data memory transfer size bus (word, halfword, or byte)
Data_Add[23: 0]	Output	Data memory address bus
Data_RdReq	Output	Data memory read request
Data_WrReq	Output	Data memory write request
Data Ack	Input	Data memory acknowledge
Reg_RdData[31:0]	Input	Register read data bus
Reg_WrData[31:0]	Output	Register write data bus
Reg_Addr[7: 0]	Output	Register address bus
Reg_RdReq	Output	Register read request
Reg_WrReq	Output	Register write request
Irq_Vector[3: 0]	Input	Interrupt vector bus
Irq	Input	Interrupt request
Bit_Out	Output	Bit data output (general purpose debug output pin)

Available Support Products

The DCT SDK includes the following software tools: Java and C compilers, assembler, linker/librarian and a simulator/source level debugger.

In addition to the software tools the SDK also includes a Spartan-II based development card with Lightfoot CPU, 256k byte data memory, 512k bytes program memory, ethernet controller, RS-232 port, 32-bit timer/counter, ¼ VGA LCD and touch panel interface.

Various system software components are also available including: C/Java runtime environment, TCP/IP stacks and real time operating systems.

Ordering Information

Lightfoot is provided as an EDIF netlist, under license from Digital Communication Technologies Limited. Contact Digital Communication Technologies when ordering.

Xilinx Programmable Logic

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

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

For general Xilinx literature, contact:

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