# ARC 32-Bit Configurable RISC Processor

July 3, 2000

Product Specification

## ARC CORES

ARC House,
The Waterfront,
Elstree Road, Elstree
Herts WD6 3BS,
United Kingdom
Phone: +44 (0) 20 8236 2800
Fax: +44 (0) 20 8236 2801
E-mail: info@arcores.com
URL: www.arcores.com

ARC Cores
6862 Santa Teresa Boulevard
San Jose, CA 95119, USA
Tel: +1 408 360 2120
Fax +1 408 360 2139
E-mail: info@arcores.com
URL: www.arcores.com

## Features

- RISC architecture for low gate count and high performance
- Full RISC orthogonal instruction set
- 4-stage pipeline
- 16 single-cycle instructions (basecase)
- 32-bit ALU; all ALU instructions are conditional
- 32-bit data bus
- 32-bit Load/Store address bus
- 32-bit instruction bus
- 24-bit instruction address bus
- 32 general purpose core registers
- 24-bit program counter and stack pointer
- Maskable external interrupts
- Jumps/branches with single instruction delay slot
- Delay slot execution modes
- Zero overhead loops
- Integrated PC parallel port debug interface allowing the debugger to access the processor registers and memory.
- C Compiler, debugger, and simulator available from MetaWare Inc. GNU version also available.
- ARCangel™ development system available for evaluation and rapid product development
- Custom versions of processor available through ARC Certified Design Centers

| AllianceCORE™ Facts | |
|---|---|
| **Core Specifics** | |
| See Table 1 | |
| **Provided with Core** | |
| Document | Programmer's Guide |
| Design File Formats | EDIF netlist |
| Constraints File | ucf file |
| Verification | XCV1000-based development board |
| Instantiation Templates | VHDL, Verilog, schematic symbol |
| Reference Designs and Application Notes | White Eagle |
| Additional Items (at additional cost) | XCV1000-based hardware demonstration board, Metaware C compiler and debugger, DSP Software Library |
| **Simulation Tool Used** | |
| Model Tech ModelSim™ V5.3a | |
| **Support** | |
| Support provided by ARC Cores and ARC Certified Design Centers | |

## Applications

- 32-bit processing applications
- DSP Applications
- Systems that require a 32-bit processor with custom interfaces or instructions
- Wireless and consumer networks

**Table 1: Core Implementation Data**

| Example Implementation | Basecase ARC | Basecase ARC | ARC with DSP extensions |
|---|---|---|---|
| Device Tested | 2S150-6 | V400E-8 | V400E-8 |
| CLB Slices | 1538 | 1517 | 4800 |
| Clock IOBs | 2 | 2 | 2 |
| IOBs [1] | 82 | 82 | 82 |
| Performance (MHz) | 37 | 41 | 23 |
| Xilinx Tools | M2.1i SP6 | M2.1i SP6 | M2.1i SP6 |
| Special Features | 9 Block RAMs | 9 Block RAMs | 29 Block RAMs |
| ARC Extensions Used | 2Kb I-Cache | 2Kb I-Cache | 4Kb I-Cache, 24x24 XMAC, barrel shifter, 4Db XY mem, min, max, norm, sat add, sat subb |

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

## General Description

ARC Cores supplies the ARC processor in a basecase configuration and in a larger more powerful DSP configuration. The ARC processor is a configurable 32 bit RISC core; this means that the powerful, compact RISC core can be extended to include additional components.

The ARC processor has been designed to make the addition of custom instructions, condition flags, special registers and custom interfaces very easy. ARC Cores has developed a set of extension modules that can be used to develop an application specific version of the processor. These extensions include a set of DSP instructions that greatly increase the numerical processing power of the ARC processor. The cache size and type can be varied to match application requirements, as can the memory and interrupt systems.

ARC Cores Certified Design Centers supply the ARC processor in any configuration required by the application. Customers can also contract the worldwide ARC Cores Certified Design Centers to develop and/or integrate application specific instructions, custom logic or pre-existing customer IP to the ARC processor.

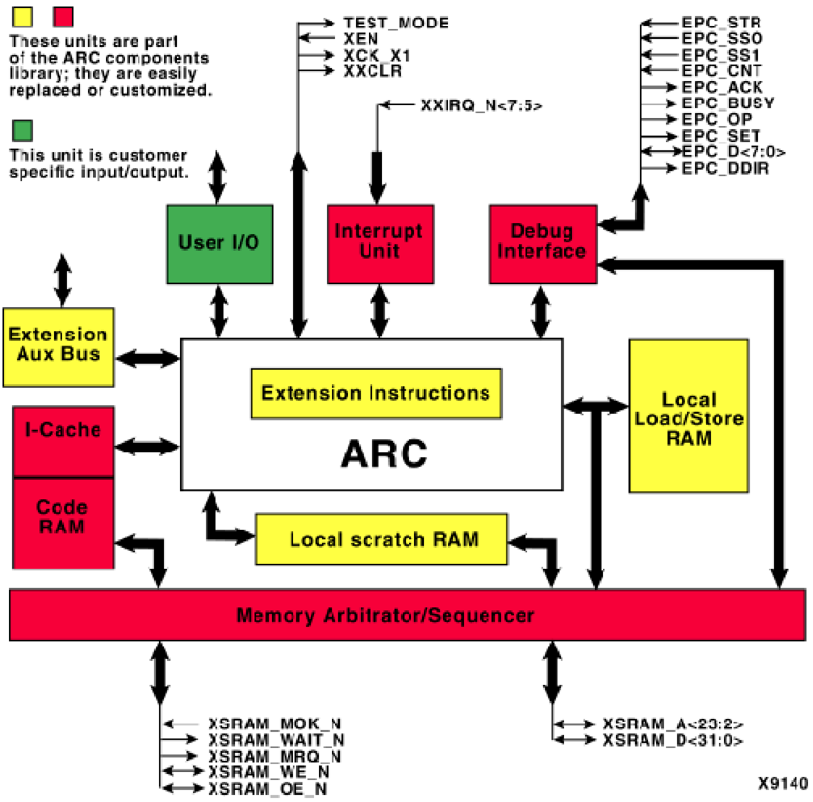

**Figure 1: ARC 32-bit RISC Processor System Block Diagram**

## Functional Description

The ARC processor is a 4-stage pipeline processor incorporating full 32-bit instruction, data and addressing. The instruction set is orthogonal with all data addressing modes implemented on all arithmetic and logical instructions and with optional conditional execution on all instructions. The processor has separate instruction and data buses (Harvard architecture) and a number of external interrupt signals. In the basecase configuration, the instruction and data buses are mapped to the same SRAM based memory space for ease of implementation and use.

It is also possible to build multiprocessor systems with the ARC processor, as it can be configured as a "master" processor or a "slave".

### Debugging Features

The ARC processor has a unique hardware based host interface which gives external devices full access to the internal registers and memory; this means that the user does not have to run a monitor program in order to debug application software or control the processor externally. External access to the processor control registers means that the processor can easily be halted, single stepped and interrogated at all levels. A PC parallel port based host interface is supplied as a standard interface; however, it can be replaced with a JTAG, RS232 or any user defined interface. The MetaWare tools make full use of this interface to provide a powerful source- level debugging capability. The MetaWare interface to ARC host interface is DLL based so that the same debugging power is immediately available through all types of host interface modules.

### Interrupt Unit

The interrupt unit contains the exception and interrupt vector positions, the logic to tell the ARC which level of interrupt has occurred, and the arbitration between the interrupts and exceptions. The interrupt unit can be modified to alter the priority of interrupts, the vector positions, and the number of interrupts. There are eight interrupts in the basecase ARC, each of which has its own vector address. Three interrupts are available externally on the basecase ARC, but many more can easily be added by an ARC Certified Design Center.

## Memory Units

The flexible memory system of the ARC processor possesses the ability to include client-own memory. An ARC processor system consists of multiple memory channels which allows a system architect to choose which channel to connect to the 'client own memory'. Each memory channel can be connected to its own independent memory.

The ARC processor has an additional memory arbitration unit, which allows the multiple independent memory channels to be connected to a memory system via a single interface. The arbitration unit can have up to 14 individual client interfaces which it arbitrates using a simple priority system. Full arbitration may not be required if the preservation of a Harvard Architecture is needed as, in this case, the program instruction fetch logic (or I-Fetch unit) is connected directly to a separate memory, and there is no contention with access to data memory.

## Instruction Cache

Another unique feature of the ARC processor is the configurable instruction cache. The I-cache improves the performance by reducing the dependency of the processor on comparatively slow accesses to the system memory. For maximum performance each application will require a different variation of cache set up. During the design process, selections from various options for the cache configuration (e.g. size, line length, memory address width and a selection of modes of operation) can be made and tested with the ARC tools.

There are various modes of I-Cache operation:

- Cache Debug Mode
- Cache Bypass Mode (processor executes from memory as cache line is loaded)
- Mixed Code RAM/Instruction Cache Mode (fixed mapping of a portion of cache to the memory)
- Line Locking Mode (prevents certain data lines being replaced in the cache)
- Virtual Cache Mode (cache size and line length can be artificially reduced, allowing for optimization of cache design)
- Multi-Way Associative Instruction Cache (allows a cache line to be loaded in a choice of cache memory)

## Load/Store Unit

The load/store unit contains the register scoreboard for tracking which registers are waiting for data from delayed loads. The size of the scoreboard can be modified according to the number of delayed loads that the memory controller can accommodate.

## Instruction set

The ARC processor has 16 basecase instructions, with an additional 12 variations to provide a set of 28 arithmetic and logical instructions, load/store, and branch/jump instructions. A further 16 dual operand and 53 single operand instructions slots are available for the addition of application specific extensions. Any ARC Certified Design Center can populate these spare instruction slots with ARC extension instructions or application specific instructions for a minor additional charge. ARC extension instructions are fully supported by the MetaWare C tools; with minor modifications the tools can also support customer specific instructions.

**Table 2: Basecase ARC Instruction Set**

| Mnemonic | Description |
|---|---|
| LD | Load indirect, register + register/offset |
| LR | Load from auxiliary register |
| ST | Store indirect, register + signed offset |
| SR | Store to auxiliary register |
| FLAG | Set condition-code flags |
| BRK | Breakpoint |
| SLEEP | Sleep |
| ASR | Arithmetic shift right 1 |
| LSR | Logical shift right 1 |
| ROR | Rotate right 1 |
| RRC | Rotate right through carry 1 |
| SEX | Sign-extend 8 – 32 |
| SEX | Sign-extend 16 – 32 |
| EXT | Zero-extend 8 – 32 |
| EXT | Zero-extend 16 – 32 |
| Bcc | Conditional branch |
| BLcc | Conditional branch and link |
| LPcc | Conditional enter loop mode |
| Jcc | Conditional jump |
| JLcc | Conditional jump and link |
| NOP | Null Instruction |
| ADD | Add |
| ADC | Add with carry |
| SUB | Subtract |
| SBC | Subtract with borrow |
| AND | Logical AND |
| OR | Logical OR |
| BIC | Logical AND-invert |
| XOR | Logical exclusive-OR |

## Core registers

The ARC processor has 64 core registers; 36 of these are used in the basecase processor for general purpose registers, link registers and loop count and immediate data indicators. Register positions 32 to 59 are free for customers own custom extensions.

### Auxiliary registers

The auxiliary register set, whose function can be compared to that of an I/O register space, is similar to a memory space that is accessed with special load and store instructions; it contains configuration registers that would otherwise occupy core register slots or be memory mapped, a condition which would interfere with real memory accesses. The auxiliary registers are addressed using 32-bit addresses and can be accessed in one clock cycle. The basecase ARC processor uses 6 status and control registers and reserves the additional registers 0x60 to 0x7F, leaving the rest of the 232 registers for extension purposes.

### Condition codes

There are 32 condition codes, the first 16 (00-0F) being defined in the basecase ARC processor. The remaining 16 condition codes (10-1F) are available for customers own extensions and are used to provide additional tests on the internal condition flags, test extension status flags from external sources and test combinations of external and internal flags.

### Multiple interface buses

As well as providing the conventional I/O approach to communicate to external devices through the Auxiliary Register Set and the Load/Store Bus, the extendibility of the ARC processor additionally opens up the interfaces to the Core Register Set, the ALU and the Condition Code Unit. The ARC processor provides the most open and user customizable interface to any external device. This allows program flow to be easily controlled, extremely fast and have direct access to extension functionality, and the choice of the optimal connection according to the frequency and response required.

### DSP Capabilities

The ARC processor also provides a powerful package of DSP features to make it capable of addressing demanding DSP tasks. This includes features such as a multiply accumulate instruction that is available in 24x24, 16x16 and dual 16x16 bit modes, a local XY memory space that has DSP based addressing modes, and saturating add and subtract instructions. A DSP software function library containing many standard DSP algorithms is also available from ARC Cores.

## Core Modifications

The ARC processor has been designed to be easily modified and an ARC Certified Design Center can make unlimited modifications to the core. Refer to the Related Information Section of the datasheet to contact the design centers. Typical modifications to improve the performance and to customize to specific applications include:

- implementation of a multiway cache,
- tuning cache size to suit application,
- addition of DSP instructions,
- extensions to the interrupt system,
- different debug host interface (e.g., RS232)
- different memory sequencer (e.g., synchronous RAM).
- mapping of peripherals to auxiliary register bus
- addition of custom application specific instructions
- addition of custom interfaces to enable rapid data flow.
- addition of existing customer IP

### Available Extensions

- 16 User-defined Dual Operand Instruction Codes
- 55 User-defined Single Operand Instruction Codes
- 28 User-defined Core Registers
- 32 Bit User-defined Auxiliary Register Set
- 16 User-defined Condition Codes

## Pinout

Signal names are shown in Figure 1 and described in Table 3.

## Verification Methods

The ARC processor core has been verified on Xilinx devices in the ARCangel development system. The ARCangel development system is available from ARC Cores and can be used to evaluate the ARC processor and for software development.

The ARC processor has also been tested in simulation using testbenches that include self checking programs. There are several tests for the core itself and core extensions each include additional testbenches that are used to test the extension component within the processor system.

## Ordering Information

The ARC Cores 32-bit configurable RISC processor is provided under license from ARC Cores for use in Xilinx devices in netlist form. Please contact your local ARC Cores Sales representative for more details.

**Table 3: Core Signal Pinout**

| Signal | Signal Direction | Description |
|---|---|---|
| XSRAM_MOK_N | Output | SRAM bus master acknowledge. Set low to indicate that the bus master request which had been received on sram_mrq_n has been granted. This signal is used in the pad ring for Z control over xsram_a, xsram_we_n, xsram_oe_n and xsram_d. When other master has finished with memory interface, it should take sram_mrq_n high, and xsram_mok_n will be released. |
| XSRAM_WAIT_N | Input | SRAM bus wait input to sequencer. Prevents the current transaction from being completed until it is removed. Active low. |
| XSRAN_NRQ_N | Input | SRAM bus master request. Another master on the memory interface may request ownership of the bus. After the current transaction has been completed, the mok signal will be taken low to indicate that the other master may now access the bus. Active low. |
| XSRAM_WE_N | Input/Output | SRAM bus byte write enable signals. 31-24, 23-16, 15-8, 7-0 respectively. Active low. |
| XSRAM_OE_N | Input/Output | SRAM bus output enable. Active low. |
| TEST_MODE | Input | Production test mode signal. Only enabled during ATPG generation and production test in the factory. Always disabled for normal operation. Is this necessary? |
| XEN | Output | Global enable. True when ARC is running in normal operation. |
| XCK_X | Input | System clock. |
| XXCLR | Input | Asynchronous hardware reset. |
| XXIRQ_N<7:5> | Input | SRAM bus address output. Z-state when sram_mok_n ='0' |
| XSRAM_A<23:2> | Input/Output | Receive Clock Enable: an active high user input, synchronous with RXC. |
| XSRAM_D<31:0> | Input/Output | SRAM bus data. |
| EPC_STR | Input | Parallel port control register set. Connector pin 14. |
| EPC_SS0 | Input | Parallel port control signal for Sun communications. Connector pin 16. |
| EPC_SS1 | Input | Transmit Underrun: an active high output pulse, synchronous to TXC, from the transmitter indicating an underrun error. This occurs upon start of frame transmission, if TX_DATA_VALID is deasserted when TX_LOAD is asserted. |
| EPC_CNT | Input | Parallel port control register set. Connector pin 14. |
| EPC_ACK | Output | Parallel port acknowledge. Connector pin 10. |
| EPC_BUSY | Output | Parallel port fsm busy signal. Connector pin 11. |
| EPC_OP | Output | Parallel port access rejected signal. Connector pin 12 |
| EPC_SEL | Output | Receive Status: an active high pulse, synchronous to RXC, to inform the user that receive frame status is being output on the RX_DATA bus. RX_STATUS is coincident with the RX_READY signal. |
| EPC_D<7:0> | Input/Output | Parallel port data. Connector pins 2-9. |
| XSRAM_A<23:2> | Input/Output | SRAM bus address output. Z-state when sram_mok_n = '0' |
| XSRAM_D<31:0> | Input/Output | SRAM bus data. |

## Recommended Design Experience

Users should be familiar with processor technology and skilled in programming with C and/or assembly.

## Available Support Products

1. The ARCangel development system, based on a Virtex XCV1000FG680-6, is available from ARC Cores.
2. A DSP software function library containing many standard DSP algorithms is also available from ARC Cores.
3. The C compiler, assembler and source level Debugger are available from ARC Cores, contact:

   MetaWare, Inc.
   2161 Delaware Avenue,
   Santa Cruz, CA, 95060-5706 (www.metaware.com).
4. A GNU version of the C tools is also available, please contact ARC cores for more details.

## Related Information

### ARC Cores Certified Design Centers

For more information on obtaining custom configurations of the ARC processor, please contact the worldwide ARC Cores Certified Design Centers.

**DELTA, Denmark**

Venlighedsvej 4
DK-2970 Horsholm, Denmark
Tel +45 45 867722
Fax +45 45 86 58 98
Email: europrac@delta.dk
URL: http://www.delta.dk

**Fraunhofer-Institut, Germany**

Am Weichselgarten 3
D-91058 Erlangen, Germany
Tel +49 9131 776 401
Fax +49 9131 776 499
Email: europrac@iis.fhg.de
URL: http://www.iis.fhg.de

**White Eagle Systems Technology, USA**

3150 Almaden Expy, Suite 229
San Jose, CA 95118, USA
Tel +1-408.445.6840
FAX 408.445.6860
Email: sales@westinc.com
URL: www.westinc.com

### 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