digital

Digital Semiconductor SA-110 Microprocessor Tools Brochure

SA-110 Microprocessor Tools

The SA-110 microprocessor (SA-110) program offers a robust development environment to help OEMs and third-party software applications providers incorporate the SA-110 into their products easily and quickly. The environment includes familiar Advanced RISC Machines Ltd. (ARM) tools and an evaluation board kit. This document lists the software and hardware tools available for use with the SA-110. The main features of each of the tools are also listed.

SA-110 Description

The SA-110 is the first member of Digital Semiconductor's StrongARM family of high-performance, low-power microprocessors. The SA-110 is the latest implementation of Advanced RISC Machines Ltd. (ARM) Version 4 architecture and offers significant advances in microprocessor design.

The SA-110 is a general-purpose, 32-bit microprocessor with a 16KB instruction cache (Icache); a 16KB, write-back data cache (Dcache); a write buffer; and a memory-management unit (MMU) combined in a single chip. The five-stage pipeline distributes tasks evenly over time to remove bottlenecks, ensuring high throughput for the core logic. The SA-110 offers high-level RISC performance, yet it ensures minimal power consumption, making it ideal for portable, low-cost systems.

The SA-110 onchip MMU supports a conventional 2-level page-table structure, with a number of extensions, which makes it ideal for embedded control systems and object-oriented systems.

Software Tools

The ARM Software Developer's Kit includes all the essential tools needed to get your software development off to a quick start. The suite of tools helps developers generate and run benchmarks on computer code, simulate processors and memory, download computer code to target hardware, and debug code quickly and efficiently. All software is supplied on CD–ROM. The ARM Software Developer's Kit consists of:

- ARM C Compiler and ANSI C Library
- ARM Assembler
- ARM Linker
- ARM Symbolic Debugger
- ARM Instruction Set Simulator (ARMulator)
- Hardcopy manuals

These tools are available now and they are compatible with DOS, UNIX (SunOS), UNIX (HP), Windows 3.1, Windows NT (Intel), Windows NT (Alpha), and Windows 95.

Hardware Tools

The SA-110 Evaluation Board (EBSA-110) Kit provides a flexible hardware environment to help you develop your SA-110 application quickly. The non-proprietary example design, using the SA-110, combines a fast memory subsystem with a flexible I/O subsystem.

ARM C Compiler and Library

The ARM C compiler is a mature, industrial-strength compiler, based on the multitarget, multilanguage Norcroft compiler suite from Codemist Limited.

Both 26-bit and the more recent 32-bit addressing modes are supported along with big and little endian modes. A mature and efficient procedure call standard (APCS) is defined with variants targeted to specific requirements.

Features

The ARM C Compiler and Library has the following features:

- Provides compatibility with ANSI and BSD pcc language compile-time options
- Provides full ANSI C library
- Meets Plum-Hall C Validation Suite standard
- Provides floating-point hardware plus emulation in software
- Establishes few implementation limits—limits are defined by available memory
- Compiles quickly—800 lines/s on a Pentium 100 system
- Provides full range of pragma directives, controlling stack checking, warnings, optimization, and so on
- Allows inline software interrupt instruction generation
- Supports writing low-level functions in C (IRQ handlers)
- Allows functions to be compiled one per code area, allowing the link step to reduce the final image size by removing unused functions
- Supports different procedure call standards, for example, with stack limit checking performed in software or hardware
- Provides comprehensive shared library support, allowing user-code libraries to be shared between applications running in the same address space
- Provides high optimization (by default), balancing code size and execution speed with the following optional items:
  - Common expression elimination
  - Loop constant elimination
  - Live range splitting (for dynamic register allocation)
  - Constant folding
  - Tail calling
  - Cross-jump elimination
  - Using table-driven peepholes
  - Structure splitting

ARM Assembler

The ARM Assembler translates ARM assembly language into ARM Object Format. The assembler supports all ARM instructions by using standard mnemonics. It can produce debugging information suitable for use with the ARM Symbolic Debugger.

The ARM Assembler is easy and fast to use, reducing training and development time. Easy linking of assemble code modules with C, through use of the ARM Procedure Call Standard (APCS), means that only very speed-critical or low-level routines need to be written in assembler language.

Features

The ARM assembler has the following features:

- Supports all ARM instructions including floating-point, cache control, and memory-management unit (MMU) control
- Supports big and little endian modes
- Allows specification of the target processor, and generates warnings if that processor does not support a particular instruction
- Produces listings and cross-referencing information
- Provides support for local labels and macros to simplify complex coding problems
- Provides support for accessing and placing multiple literal pools
- Supports full range of binary, logical, and relational operators
- Provides comprehensive documentation

ARM Linker

The ARM Linker produces an executable image from ARM Object Format and ARM Library Format files produced by the ARM C and C++ Compilers, ARM Assembler, and ARM Librarian.

The linker supports scatter-load image creation and contains benchmarking facilities for code-size and data-size reports and summaries.

The linker is fast and memory-efficient. It provides flexible output options, so that a customized conversion program or loader is seldom required.

Features

The ARM Linker has the following features:

- Provides a highly reliable, industrial-strength, high-performance linker
- Provides the facility for shared libraries of code that can be accessed by a number of clients within an address space, reducing overall memory usage
- Provides support for static and dynamic overlays, which is useful for reducing memory usage in some circumstances
- Produces a full link map showing code and data areas
- Allows for a wide range of useful and easy to use output formats
- Allows production of partially linked output (as ARM Object Format), which can be used as input to another link step
- Allows for removal of unreferenced areas, reducing code and data size

ARM Symbolic Debugger

The ARM Symbolic Debugger provides debugging facilities for ARM programs, normally in ARM Image Format.

Debugging can be done on a simulated ARM by using ARMulator, or over a serial link to an ARM Development System.

The debugger supports both assembler and C modes, with facilities for single stepping, setting breakpoints and watchpoints, and examining and changing variables, as well as low-level debugging.

Features

The ARM Symbolic Debugger has the following features:

- Allows remote debugging over a serial link
- Provides support for debugging under the ARMulator
- Contains a count-based profiler that develops the dynamic call graph and uses PC-sampling to determine where call graph time is spent
- Understands a full range of C operators
- Permits single stepping and stepping over procedure calls
- Allows navigation up and down the stack, to examine local variables at any level
- Provides access to machine registers (including floating-point) and other low-level debugging features
- Supports a configuration file to load a program and bring it to a predefined state
- Makes the following list of extensive views available:
    - Registers
    - Variables
    - Functions
    - Backtrace
    - Memory
    - Symbols
    - Debugger internal/statistics
    - Expressions
- Supports complex breakpoints
- Supports complex watchpoints

ARMulator

The ARMulator provides the user with a means of simulating an ARM processor on the development host computer. All variants of the ARM processor core are supported, and the simulator can be extended to handle entire ARM-based systems and application-specific integrated circuits (ASICs).

The ARMulator is useful for several reasons.

- It allows development and testing of ARM software on non-ARM host systems.
- It allows accurate benchmarking of ARM software.
- It supports the simulation of prototype ARM designs, before the hardware is actually available, allowing the development of hardware and low-level software to proceed in parallel.
- It can be incorporated into other simulators (for example, for other hardware) to simulate a complete system.

The ARMulator is supplied with user-modifiable source for SWI handlers and memory interfacing, event scheduling, and coprocessor interface modules.

Features

The ARMulator has the following features:

- Interfaces to the ARM Symbolic Debugger to allow debugging of the simulated system
- Provides benchmark-accurate simulation
- Communicates with the Debugger by using the ARM Remote Debug Interface
- Allows an operating system to be placed on top of the emulator, using SWIs, I/O areas, and so on
- Provides high-performance, equivalent to ARM 6 at around 1 MHz on a Sun SPARCstation 10
- Supplies ARMulator Operating System in source form, which provides filing operations and full IEEE floating-point support
- Provides full ANSI C library

SA-110 Evaluation Board

The SA-110 evaluation board (EBSA-110) is an example design that hardware designers can use to expedite product development and software developers can use to test their application before a complete hardware system is available. It provides an excellent environment for the following activities:

- Designing software and hardware prototypes
- Porting operating systems and applications
- Running ARM V4 architecture-compliant software test suites
- Running software benchmarks

Availability

The design details of the EBSA-110 are available in the SA-110 Hardware Developer's Kit. The EBSA-110 systems are not available for purchase but are available for loan to qualified software and hardware application developers and customers. Contact your Digital Semiconductor sales representative.

EBSA-110 Features

The EBSA-110 board conforms to the Baby AT form factor and is packaged in a PC desktop enclosure with a standard PC power supply (only 5 V and 12 V are used).

Configuration Selection

The EBSA-110 provides the following configuration selections:

- Core clock (**cclk**) and system clock (**mclk**) frequency selection
- Up to four boot options from flash ROM
- Flash ROM or EPROM boot
- Etch links for CPU verification options (tristate controls, asynchronous bus mode, and so on)
- SA-110 core voltage

LED Indicators

There are five LED indicators on the EBSA-110. They are provided for these purposes:

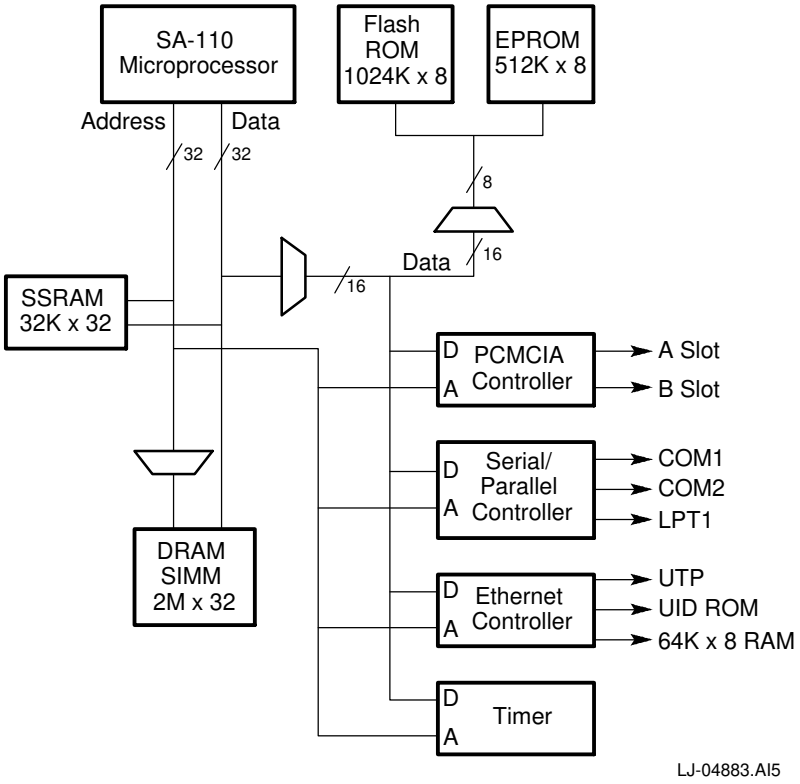
- One for debug
- Four for Ethernet TP status

EBSA-110 Hardware Components

The main hardware components of the EBSA-110 are listed here and are shown in Figure 1:

- SA-110 microprocessor
- Memory
  - 1024K x 8 flash ROM
  - 512K x 8 EPROM (in socket)
  - 32K x 32 SSRAM
  - 2 SIMM slots with 2M x 32 DRAM (EDO/burst EDO) per slot
- PCMCIA controller with two slots (supports types I, II, and III cards)
- Serial/parallel controller with two serial and one parallel port
- Ethernet controller with one UTP port, a RAM, and a UID ROM
- Two of three 8254 timers are available

Figure 1 EBSA-110 Evaluation Board Block Diagram



digital

For More Information

To learn more about the availability of the SA-110 microprocessor, contact your local semiconductor distributor. To learn more about Digital Semiconductor's product portfolio, contact the Digital Semiconductor Information Line:

United States and Canada
1–800–332–2717
Outside North America
1–508–628–4760

While Digital believes the information in this publication is correct as of the date of publication, it is subject to change without notice.

© Digital Equipment Corporation 1996.

All rights reserved.

Printed in U.S.A.

EC–QPWJB–TE

Digital, Digital Semiconductor, and the DIGITAL logo are trademarks of Digital Equipment Corporation.

Digital Semiconductor is a Digital Equipment Corporation business.

ARM, the ARM Powered logo, and StrongARM are trademarks of Advanced RISC Machines Ltd.

BSD is a trademark of the University of California, Berkeley, CA.

HP is a registered trademark of Hewlett-Packard Company.

IEEE is a registered trademark of The Institute of Electrical and Electronic Engineers, Inc.

Intel and Pentium are registered trademarks of Intel Corporation.

SPARCstation and SunOS are trademarks and Sun is a registered trademark of Sun Microsystems Inc.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Ltd.

Windows 95 is a registered trademark and Windows and Windows NT are trademarks of Microsoft Corporation.

All other trademarks and registered trademarks are the property of their respective owners.

ARM POWERED ™