Features

- **32-bit RISC Architecture**
- **Two Instruction Sets:**
  - **ARM® High-performance 32-bit Instruction Set**
  - **Thumb® High-code-density 16-bit Instruction Set**
- **Very Low Power Consumption: Industry-leader in MIPS/Watt**
- **4G Bytes Linear Address Space**
- **Von Neumann Load/Store Architecture:**
  - **Single 32-bit Data Bus for Instructions and Data**
- **3-Stage Pipeline Architecture:**
  - **Fetch, Decode and Execute Stage**
- **8-, 16-, and 32-bit Data Types**
- **Single Cycle 32x8 Hardware Multiplier:**
  - **Multiplication is Accelerated when Upper Bytes Are All Zero or One**
- **On-chip JTAG Debug and In Circuit Emulation**
- **Extensive Range of Third-party Application Development Tools**

Description

The ARM7TDMI™ embedded microcontroller core is a member of the Advanced RISC Machines (ARM®) family of general purpose 32-bit microprocessors, which offer high performance and very lower power consumption. Its outstanding feature is the 16-bit Thumb® subset of the most commonly used 32-bit instructions. These are expanded at run time with no degradation of system performance. This gives 16-bit code density (saving memory area and cost) coupled with 32-bit processor performance.

The ARM architecture is based on Reduced Instruction Set Computer (RISC) principles, and the instruction set and related decode mechanism are much simpler than those of microprogrammed Complex Instruction Set Computers. This simplicity results in a high instruction throughput and impressive real-time interrupt response from a small and cost-effective chip.

Pipelining is employed so that all parts of the processing and memory systems can operate continuously. Typically, while one instruction is being executed, its successor is being decoded, and a third instruction is being fetched from memory.

The ARM memory interface has been designed to allow the performance potential to be realized without incurring high costs in the memory system. Speed-critical control signals are pipelined to allow system control functions to be implemented in standard low-power logic, and these control signals facilitate the exploitation of the fast local access modes offered by industry standard dynamic RAMs.

The ARM memory interface is also ideally suited to interfacing, either on-chip or off-chip, with Atmel's Flash memory blocks. These give the benefits of in-system programmability and security, reducing time-to-market and system cost.

The ARM7TDMI core is supported by an extensive range of application development tools. These are fully described in the AT91Business Partners section of Atmels's Web site (www.atmel.com).

Embedded RISC Microcontroller Core

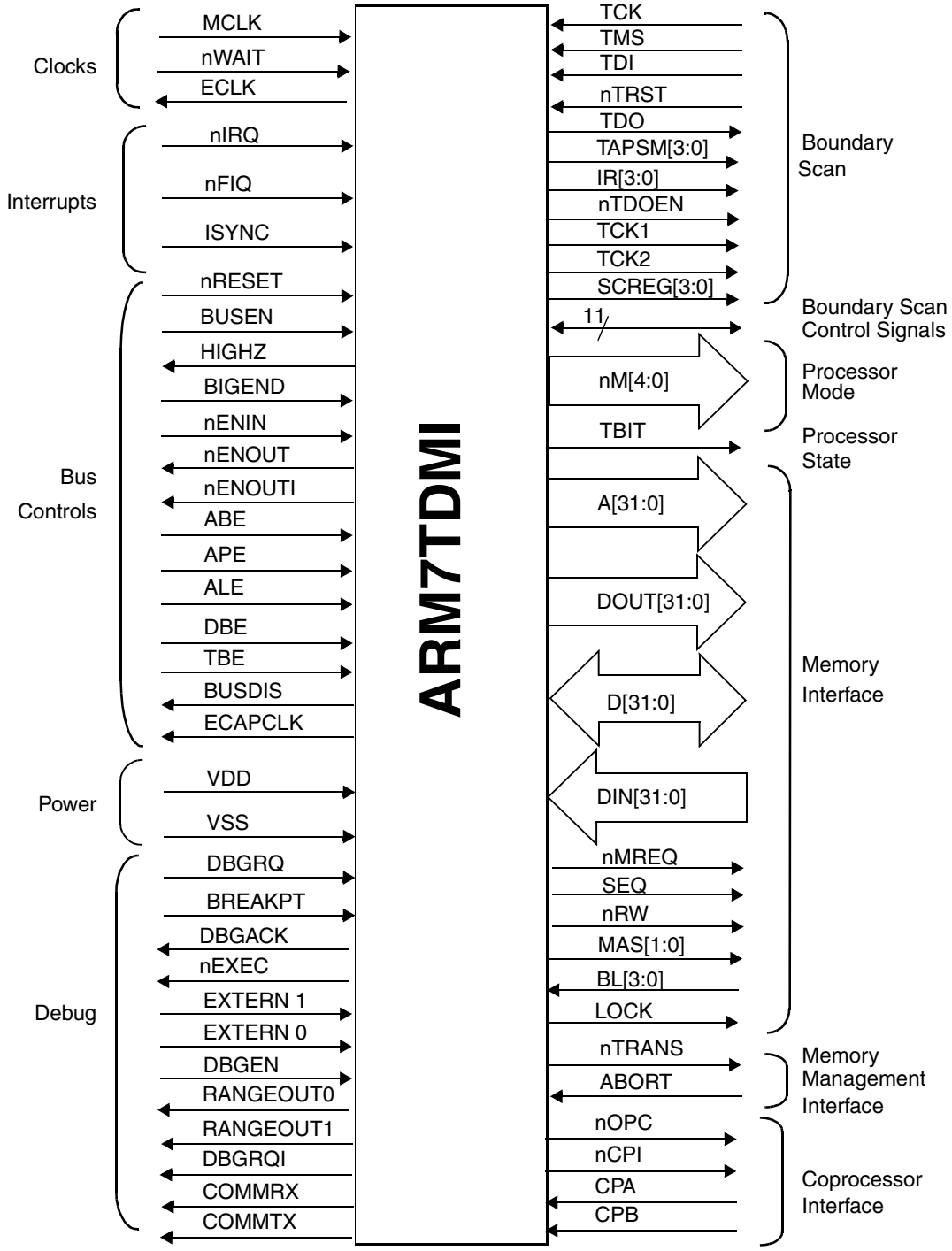
ARM7TDMI™

Rev. 0673CS–11/99

Note: This is a summary document. For the complete 204-page document, please visit our web site at *www.atmel.com* or e-mail at *literature@atmel.com* and request literature #0673B.

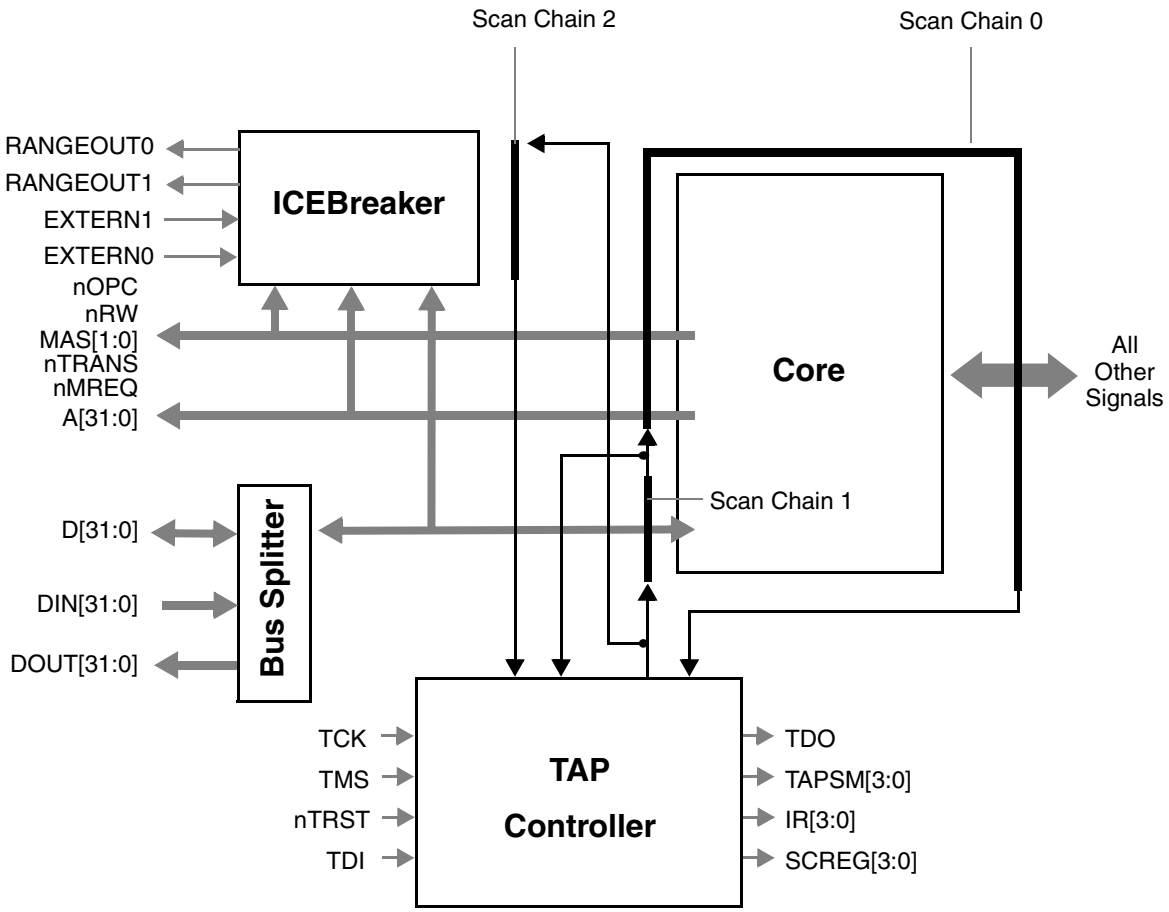
ARM7TDMI Input/Output Signals

Figure 1. ARM7TDMI Input/Output Signals



ARM7TDMI Block Diagram

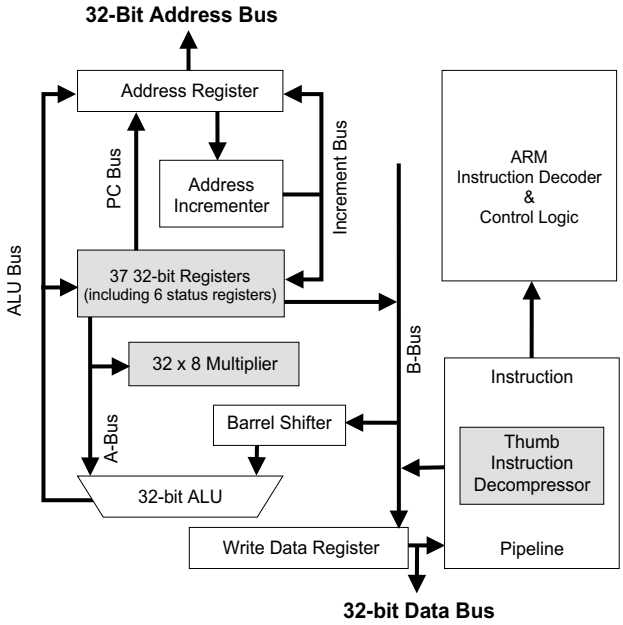
Figure 2. ARM7TDMI Block Diagram



As shown in Figure 1 and Figure 2, the ARM7TDMI consists of a processor, a TAP controller for boundary scan, and an in-circuit emulator (ICEBreaker). The bi-directional data bus D[31:0] is split into uni-directional input and output buses for compatibility with a wide range of external memories.

ARM7TDMI Processor

Figure 3. ARM7TDMI Processor



The ARM7TDMI processor is built around a bank of 37 32-bit registers and six status registers. It features an integral 32 x 8 multiplier and 32-bit barrel shifter. Five independent internal buses (the PC Bus, the Increment Bus, the ALU Bus and the A- and B-Buses) allow a high degree of parallelism in instruction execution.

Operating Modes

ARM7TDMI supports seven modes of operation:

- User (usr):
  The normal ARM program execution state
- FIQ (fiq):
  Designed to support a data transfer or channel process
- IRQ (irq):
  Used for general-purpose interrupt handling
- Supervisor (svc):
  Protected mode for the operating system
- Abort mode (abt):
  Entered after a data or instruction prefetch abort
- System (sys):
  A privileged user mode for the operating system
- Undefined (und):
  Entered when an undefined instruction is executed

Mode changes may be made under software control, or may be brought about by external interrupts or exception processing. Most application programs will execute in User mode. The non-user modes - known as privileged modes - are entered in order to service interrupts or exceptions, or to access protected resources.

Each operating mode has dedicated banked registers for fast exception handling. The FIQ mode has five additional banked working registers, r8_fiq to r12_fiq, to enhance interrupt processing speed.

Registers

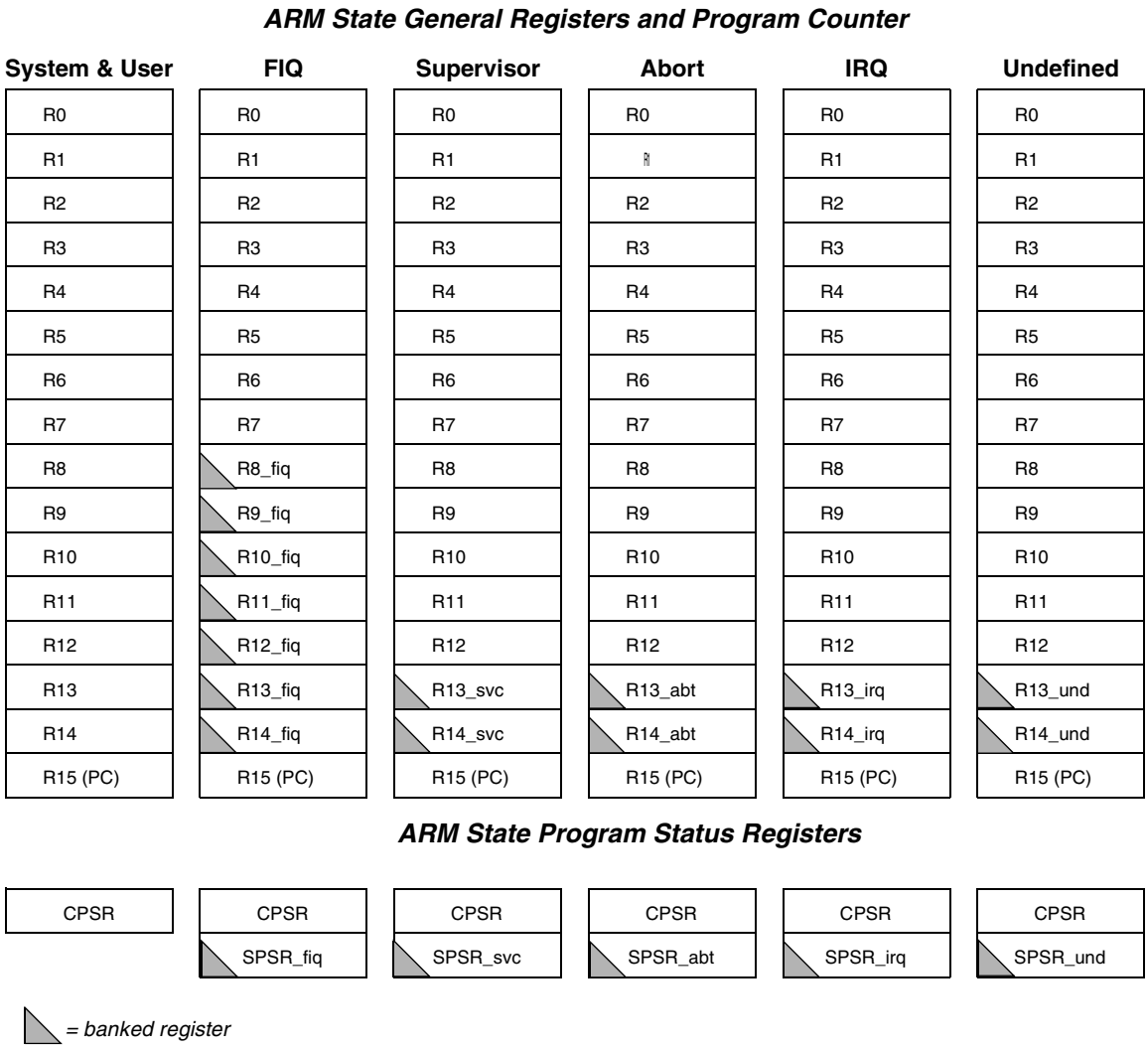
ARM7TDMI has a total of 37 registers – 31 general-purpose 32-bit registers and six status registers – but these cannot all be seen at once. The processor state and operating mode dictate which registers are available to the programmer.

The ARM State Register Set

In ARM state, 16 general registers and one or two status registers are visible at any one time. In privileged (non-User) modes, mode-specific banked registers are switched in. Figure 4 shows which registers are available in each mode: the banked registers are marked with a shaded triangle.

The ARM state register set contains 16 directly accessible registers: R0 to R15. All of these except R15 are general-purpose, and may be used to hold either data or address values. In addition to these, there is a seventeenth register used to store status information.

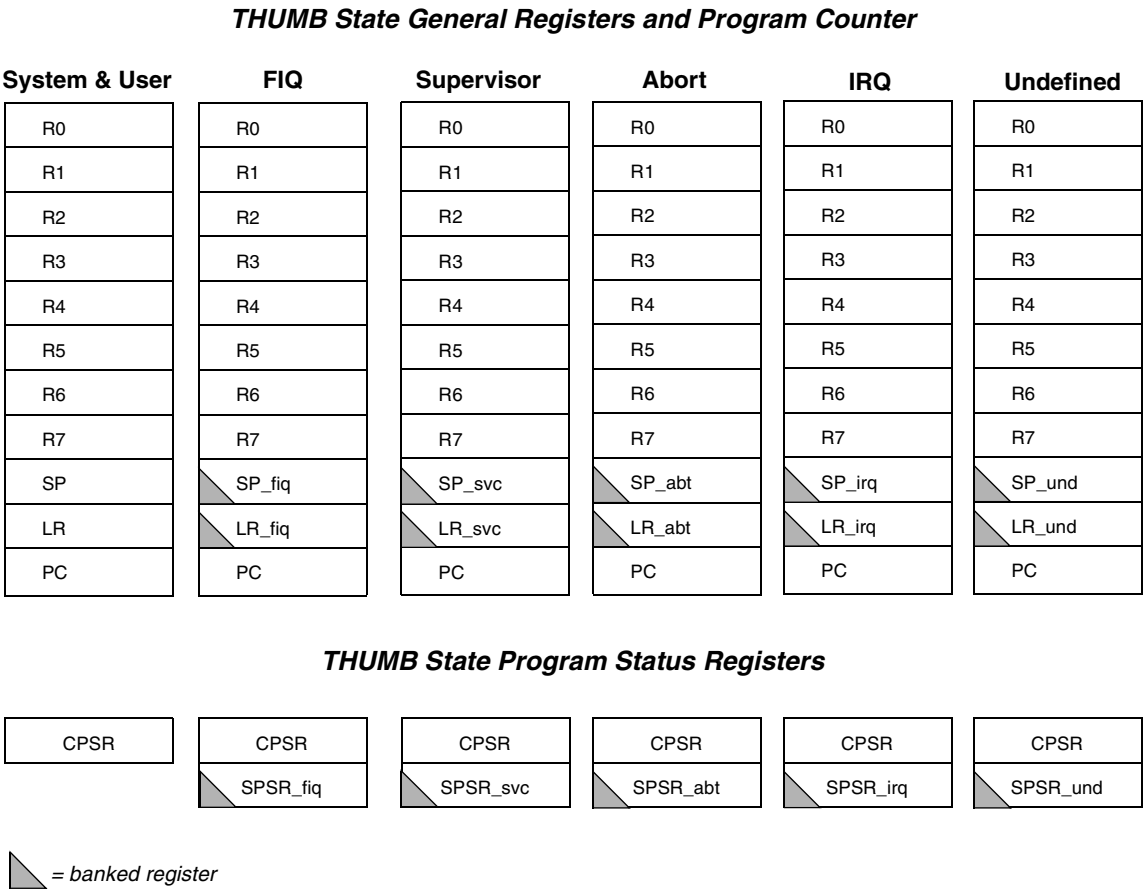
Figure 4. Register Organization in ARM State



The THUMB State Register Set

The THUMB state register set is a subset of the ARM state set. The programmer has direct access to eight general registers, R0-R7, as well as the Program Counter (PC), a stack pointer register (SP), a link register (LR), and the CPSR. There are banked Stack Pointers, Link Registers and Saved Process Status Registers (SPSRs) for each privileged mode.

Figure 5. Register Organization in Thumb State



ARM7TDMI Architecture

The ARM7TDMI is a 3-stage pipeline, 32-bit RISC processor. The processor architecture is Von Neumann load/store architecture, which is characterized by a single data and address bus for instructions and data. The CPU has two instruction sets, the ARM and the Thumb instruction set. The ARM instruction set has 32-bit wide instructions and provides maximum performance. Thumb instructions are 16-bits wide and give maximum code-density. Instructions operate on 8-, 16-, and 32-bit data types.

The THUMB Concept

The ARM7TDMI processor employs a unique architectural strategy known as *THUMB*, which makes it ideally suited to high-volume applications with memory restrictions, or applications where code density is an issue.

The key idea behind THUMB is that of a super-reduced instruction set. Essentially, the ARM7TDMI processor has two instruction sets:

- the standard 32-bit ARM set
- a 16-bit THUMB set

The THUMB set's 16-bit instruction length allows it to approach twice the density of standard ARM code while retaining most of the ARM's performance advantage over a traditional 16-bit processor using 16-bit registers. This is possible because THUMB code operates on the same 32-bit register set as ARM code.

THUMB code is able to provide up to 65% of the code size of ARM, and 160% of the performance of an equivalent ARM processor connected to a 16-bit memory system.

The Advantages of THUMB

THUMB instructions operate with the standard ARM register configuration, allowing excellent interoperability between ARM and THUMB states. Each 16-bit THUMB instruction has a corresponding 32-bit ARM instruction with the same effect on the processor model.

The major advantage of a 32-bit (ARM) architecture over a 16-bit architecture is its ability to manipulate 32-bit integers with single instructions, and to address a large address space efficiently. When processing 32-bit data, a 16-bit architecture will take at least two instructions to perform the same task as a single ARM instruction.

However, not all the code in a program will process 32-bit data (for example, code that performs character string handling), and some instructions, like Branches, do not process any data at all.

If a 16-bit architecture only has 16-bit instructions, and a 32-bit architecture only has 32-bit instructions, then overall the 16-bit architecture will have better code density, and better than one half the performance of the 32-bit architecture. Clearly 32-bit performance comes at the cost of code density.

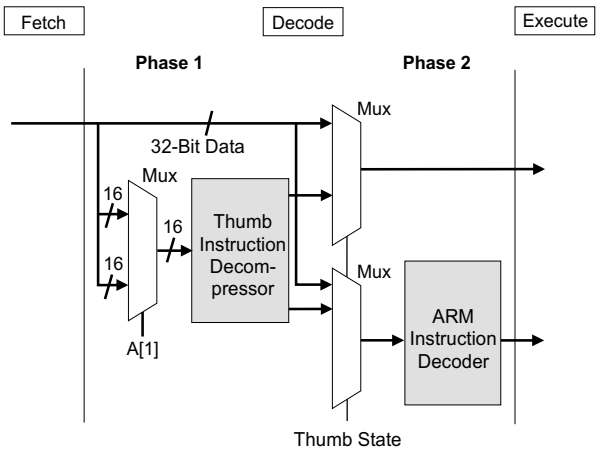
THUMB breaks this constraint by implementing a 16-bit instruction length on a 32-bit architecture, making the processing of 32-bit data efficient with a compact instruction coding. This provides far better performance than a 16-bit architecture, with better code density than a 32-bit architecture.

THUMB also has a major advantage over other 32-bit architectures with 16-bit instructions. This is the ability to switch back to full ARM code and execute at full speed. Thus critical loops for applications such as

- fast interrupts
- DSP algorithms

can be coded using the full ARM instruction set, and linked with THUMB code. The overhead of switching from THUMB code to ARM code is folded into sub-routine entry time. Various portions of a system can be optimized for speed or for code density by switching between THUMB and ARM execution as appropriate.

Figure 6. Flexible Selection of ARM or Thumb Instruction Set



Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686677
FAX (44) 1276-686697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road
Tsimshatsui East
Kowloon, Hong Kong
TEL (852) 27219778
FAX (852) 27221369

Japan
Atmel Japan K.K.
Tonetsu Shinkawa Bldg., 9F
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4 42 53 60 00
FAX (33) 4 42 53 60 01

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

0673CS–11/99/0M