ARM Processor Architecture

Pao-Ann Hsiung
National Chung Cheng University

Brief History of ARM

- ARM is short for **Advanced Risc Machines** Ltd.
- Founded 1990, owned by **Acorn**, Apple and VLSI
- Known as computer manufacturer before becoming ARM
- Acorn developed a 32-bit RISC processor for it’s own use Acorn Archimedes

Why ARM here?

- ARM is one of the most licensed and thus widespread processor cores in the world
- Used especially in portable devices due to low power consumption and reasonable performance (MIPS / watt)
- Several interesting extensions available or in development like Thumb instruction set and Jazelle Java machine

ARM

- Processor cores: ARM6, ARM7, ARM9, ARM10, ARM11
- Extensions: Thumb, El Segundo, Jazelle etc.
- IP-blocks: UART, GPIO, memory controllers, etc

CPU	Description	ISA	Process	Voltage	Area mm2	Power mW	Clock / MHz	Mips / MHz
ARM7TDMI	Core	V4T	0.18u	1.8V	0.53	<0.25	60-110	0.9
ARM7TDMI-S	Synthesizable core	V4T	0.18u	1.8V	<0.8	<0.4	>50	0.9
ARM9TDMI	Core	V4T	0.18u	1.8V	1.1	0.3	167-220	1.1
ARM920T	Macrocell 16+16kB cache	V4T	0.18u	1.8V	11.8	0.9	140-200	1.05
ARM940T	Macrocell 8+8kB cache	V4T	0.18u	1.8V	4.2	0.85	140-170	1.05
ARM9E-S	Synthesizable core	V5TE	0.18u	1.8V	?	~1	133-200	1.1
ARM1020E	Macrocell 32+32kB cache	V5TE	0.15u	1.8V	~10	~0.85	200-400	1.25

ARM architecture

- 32-bit RISC-processor core (32-bit instructions)
- 37 pieces of 32-bit integer registers (16 available)
- Pipelined (ARM7: 3 stages)
- Cached (depending on the implementation)
- Von Neuman-type bus structure (ARM7), Harvard (ARM9)
- 8 / 16 / 32 -bit data types
- 7 modes of operation (usr, fiq, irq, svc, abt, sys, und)
- Simple structure -> reasonably good speed / power consumption ratio

ARM7TDMI

- **ARM7TDMI is a core processor module embedded in many ARM7 microprocessors, such as ARM720T, ARM710T, ARM740T, and Samsung's KS32C50100. It is the most complex processor core module in ARM7 series.**
  - **T: capable of executing Thumb instruction set**
  - **D: Featuring with IEEE Std. 1149.1 JTAG boundary-scan debugging interface.**
  - **M: Featuring with a Multiplier-And-Accumulate (MAC) unit for DSP applications.**
  - **I: Featuring with the support of embedded In-Circuit Emulator.**
- **Three Pipe Stages: Instruction fetch, decode, and Execution.**

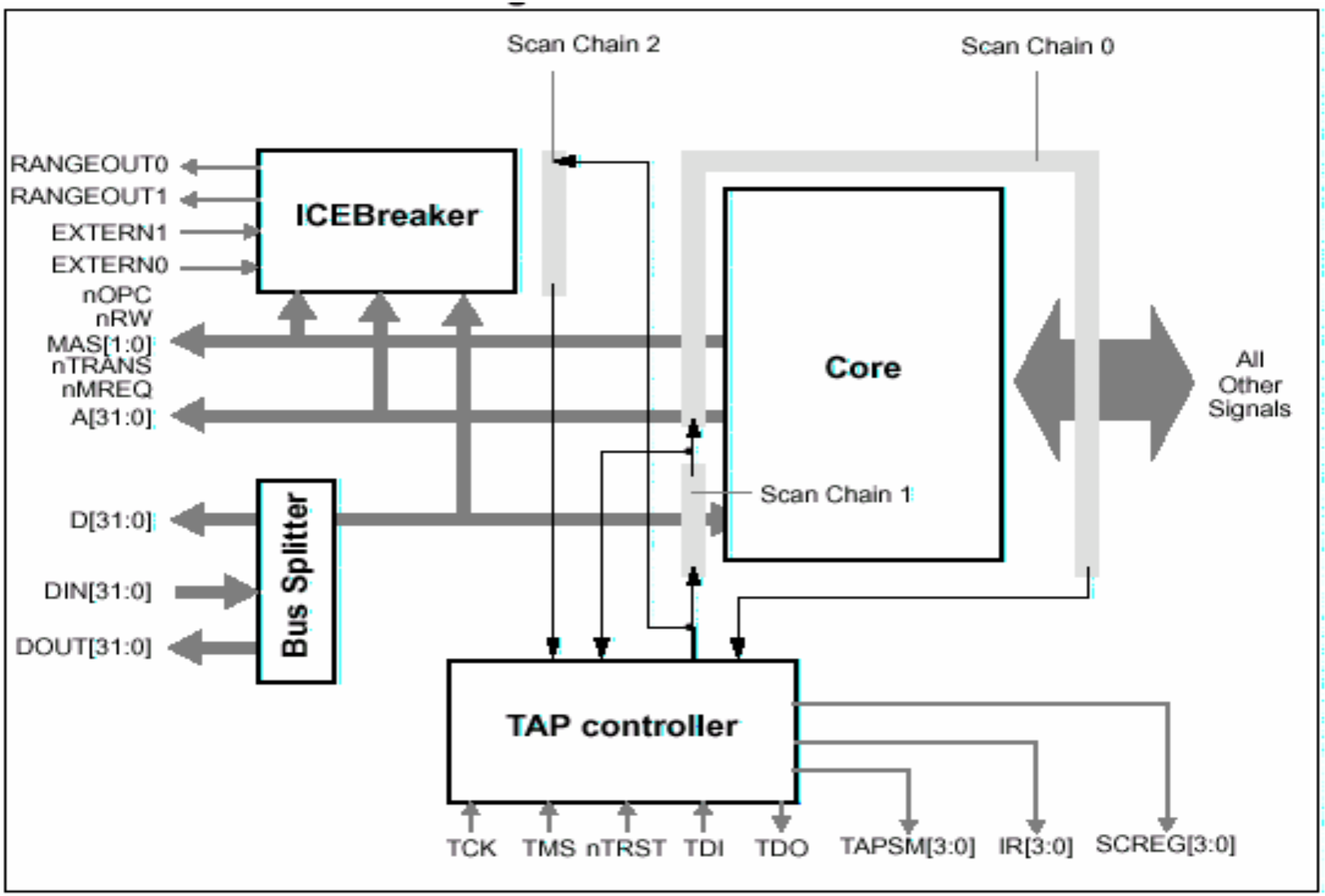
Features

- **A 32-bit RSIC processor core capable of executing 16-bit instructions (Von Neumann Architecture)**
  - **High density code**
    - **The Thumb's set's 16-bit instruction length allows it to approach about 65% of standard ARM code size while retaining ARM 32-bit processor performance.**
  - **Smaller die size**
    - **About 72,000 transistors**
    - **Occupying only about 4.8mm^2 in a 0.6*um* semiconductor technology.**
  - **Lower power consumption**
    - **dissipate about 2*mW*/*MHZ* with 0.6*um* technology.**

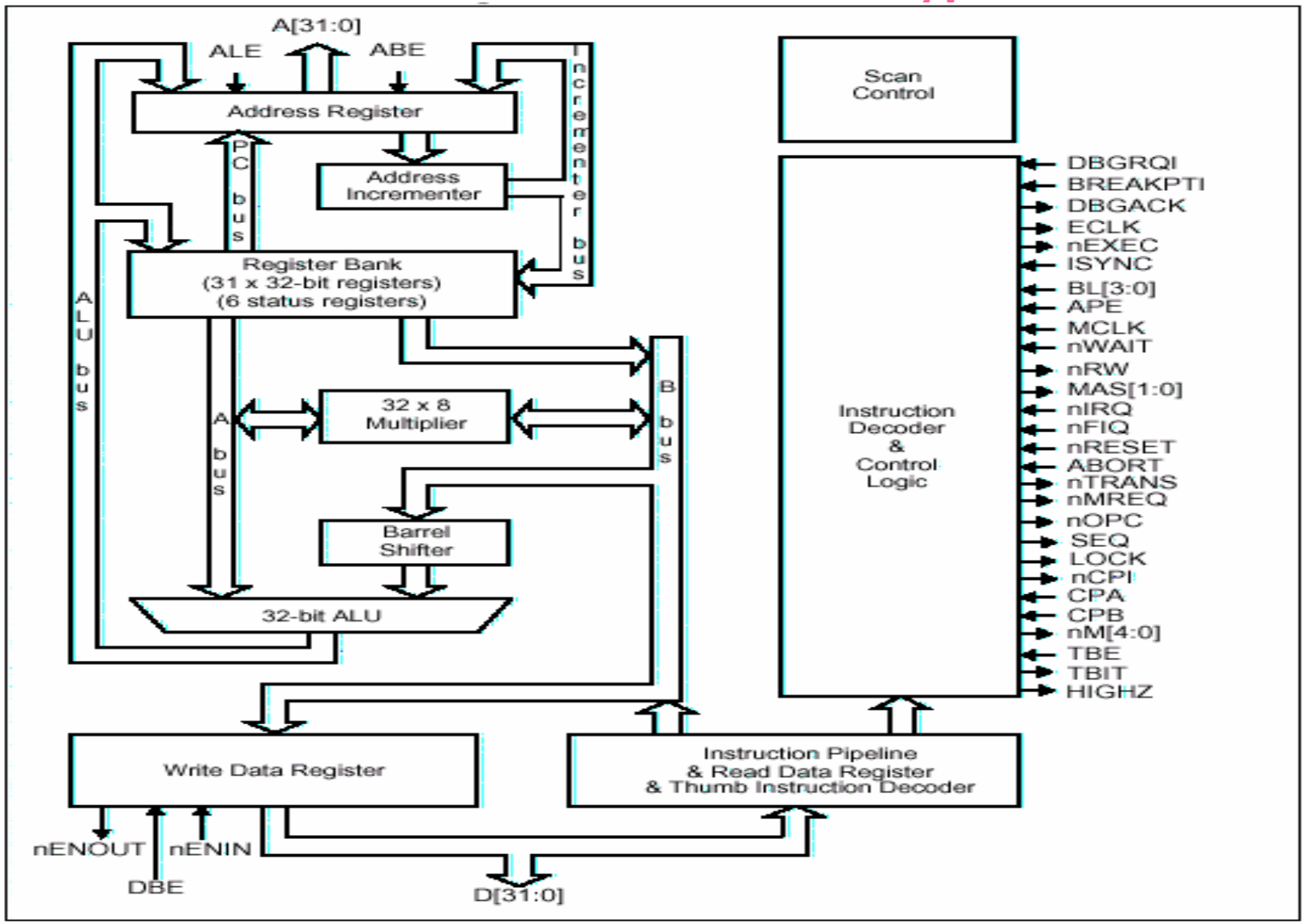
Features Cont!

- **Memory Access**
  - **Data can be**
    - **8-bit (bytes)**
    - **16-bit (half words)**
    - **32-bit (words)**
- **Memory Interface**
  - **Can interface to SRAM, ROM, DRAM**
  - **Has four basic types of memory cycle**
    - **idle cycle**
    - **nonsequential cycle**
    - **sequential cycle**
    - **coprocessor register cycle**

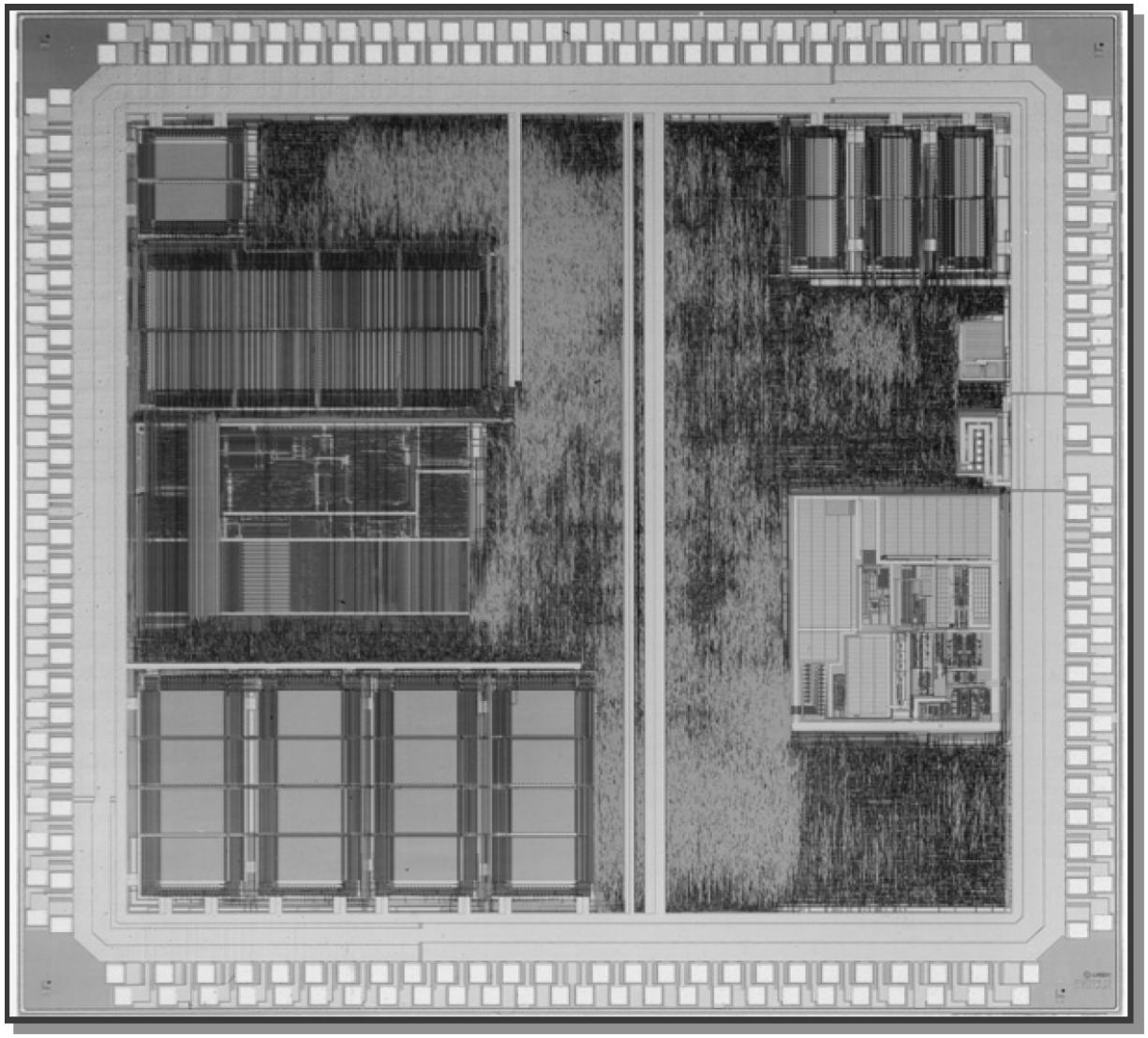
ARM7TDMI Block Diagram



ARM7TDMI Core Diagram



ARM7TDMI on Silicon



Memory Interface

- **32-bit address bus**
- **32-bit data bus**
  - **D[31:0]: Bidirectional data bus**
  - **DIN[31:0]: Unidirectional input bus**
  - **DOUT[31:0]: Unidirectional output bus**
- **Control signals**
  - **Specify the size of the data to be transferred and the direction of the transfer**

Cycle Types (1)

- **Non-sequential cycle**
  - **Requests a transfer to or from an address which is unrelated to the address used in the preceding cycle.**
- **Sequential cycle**
  - **Requests a transfer to or from an address which is either the same as the address in the preceding cycle, or is one word or halfword after the preceding cycle. But should not occur at the page end.**
- **Internal cycle**
  - **Does not require a transfer.**
- **Coprocessor register transfer**
  - **Wishes to use the data bus to communicate with a coprocessor, but does not require any action by the memory system.**

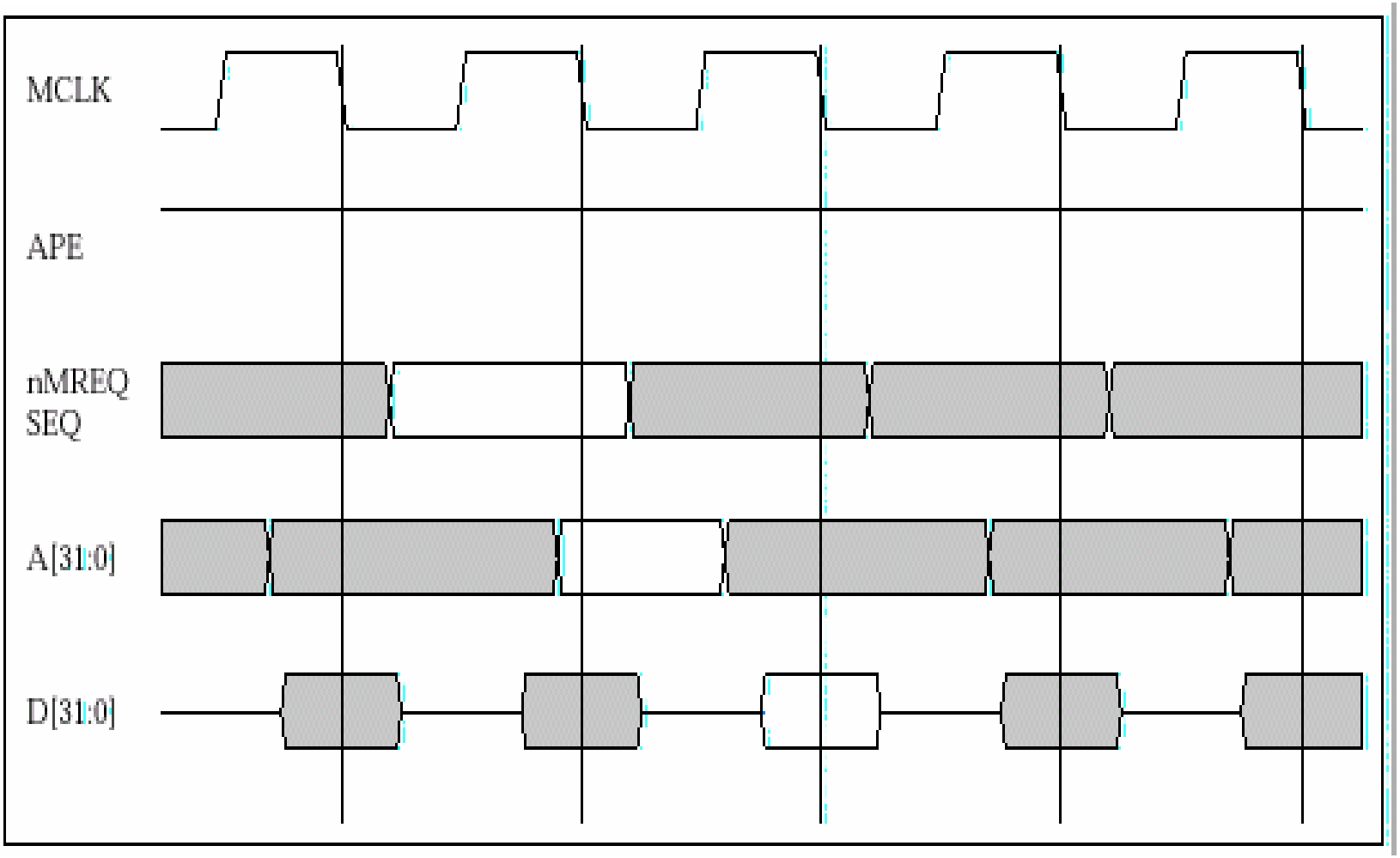
Cycle Types (2)

nMREQ	SEQ	Cycle type
0	0	Non-sequential (N-cycle)
0	1	Sequential (S-cycle)
1	0	Internal (I-cycle)
1	1	Coprocessor register transfer (C-cycle)

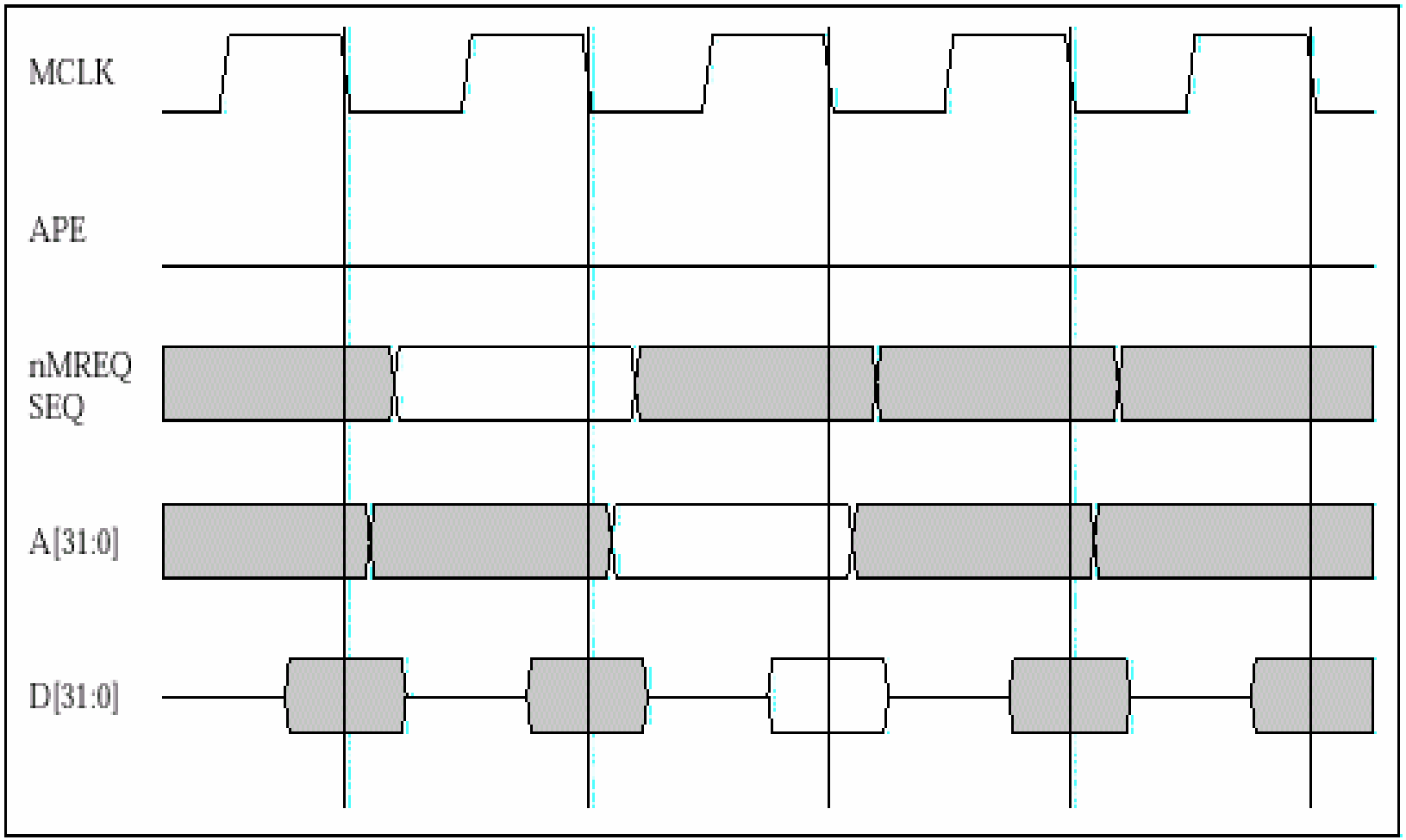
Bus Configuration

- **Pipelined**
  - **Interface to DRAM**
  - **Set *APE* to 1**
- **De-pipelined**
  - **Interface to SRAM or ROM**
  - **Set *APE* to 0**

Pipelined Addresses



De-pipelined Addresses

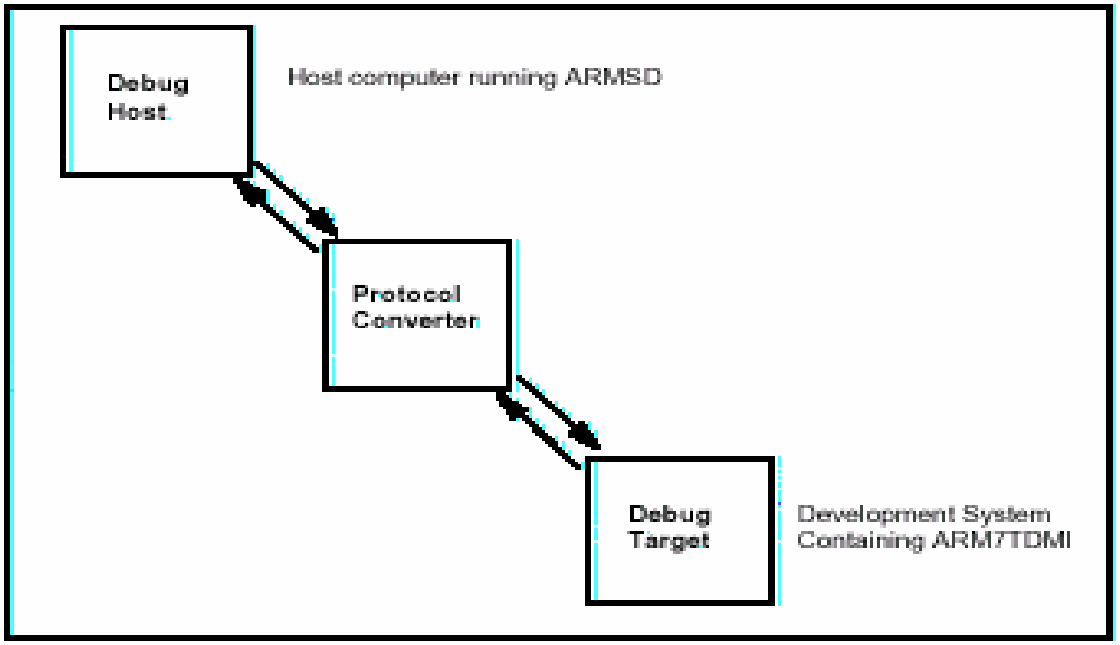


Coprocessor Interface

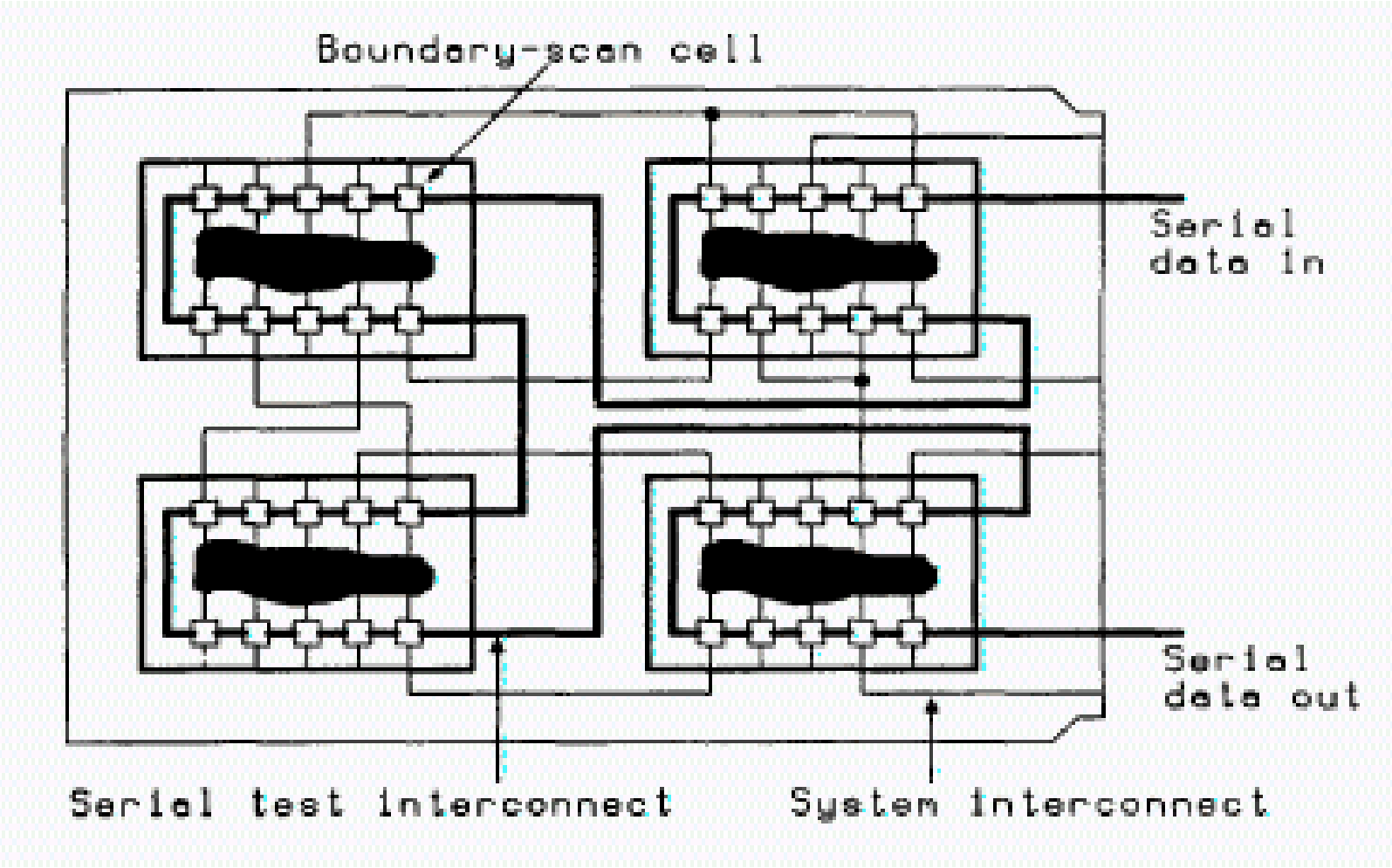
- **Can have 16 coprocessors**
- **If the designated coprocessor**
  - **exists, a coprocessor instruction will be executed by the coprocessor;**
  - **does not exist, the ARM7TDMI will take the undefined instruction trap and a software will be executed to emulate the coprocessor. The execution of a coprocessor instruction is done by software.**

Debug Interface

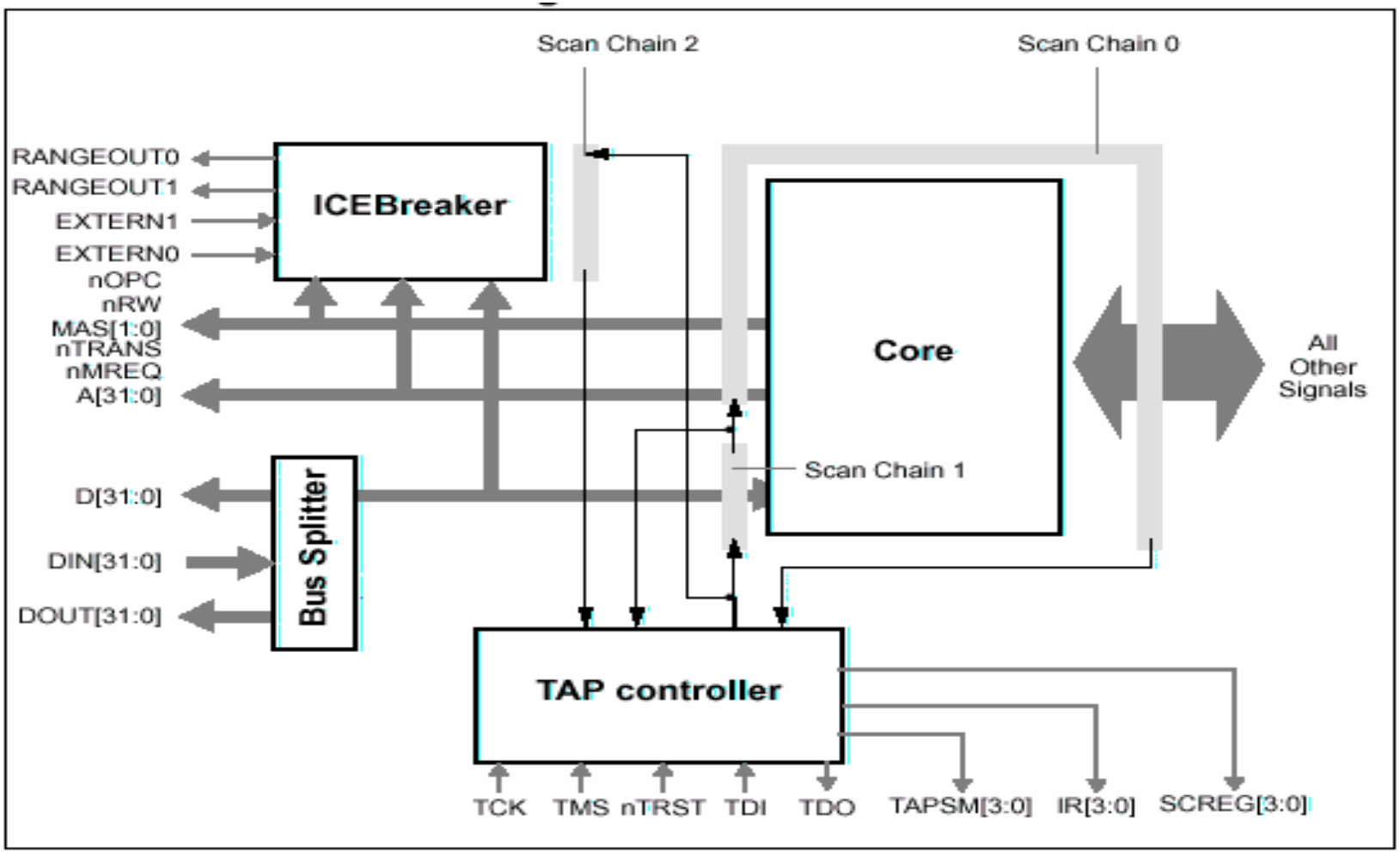
- **Based on IEEE Std. 1149.1-1990, “Standard Test Access Port and Boundary-Scan Architecture”.**
- **Debug systems**



The Concept of Boundary Scan Design



ARM7TDMI Debug Architecture



Enter Debugging State

- **ARM7TDMI is forced into debug state after a breakpoint, watchpoint, or debug request has occurred.**
  - **Breakpoint: Set for *a particular instruction*. When this instruction is executed, the machine is forced into debug state.**
  - **Watchpoint: Set for a data access. When data access occurs for *a particular address* for *a particular data*, the machine is forced into debug state.**
- **Setting the breakpoint and watchpoint by**
  - **issuing debug request, DBGRQ.**
  - **ICEBreaker programming**

Action In Debug State

- **Once entered into debug state**
  - **The internal states of the machine can be examined.**
  - **The system's external state can be examined.**
  - **The memory bus of the machine (ARM7TDMI) is forced to indicate internal cycles and the machine's outputs will change asynchronously to the memory system.**
  - **Then, the internal state of the machine can be scanned out through scan chain for examination.**

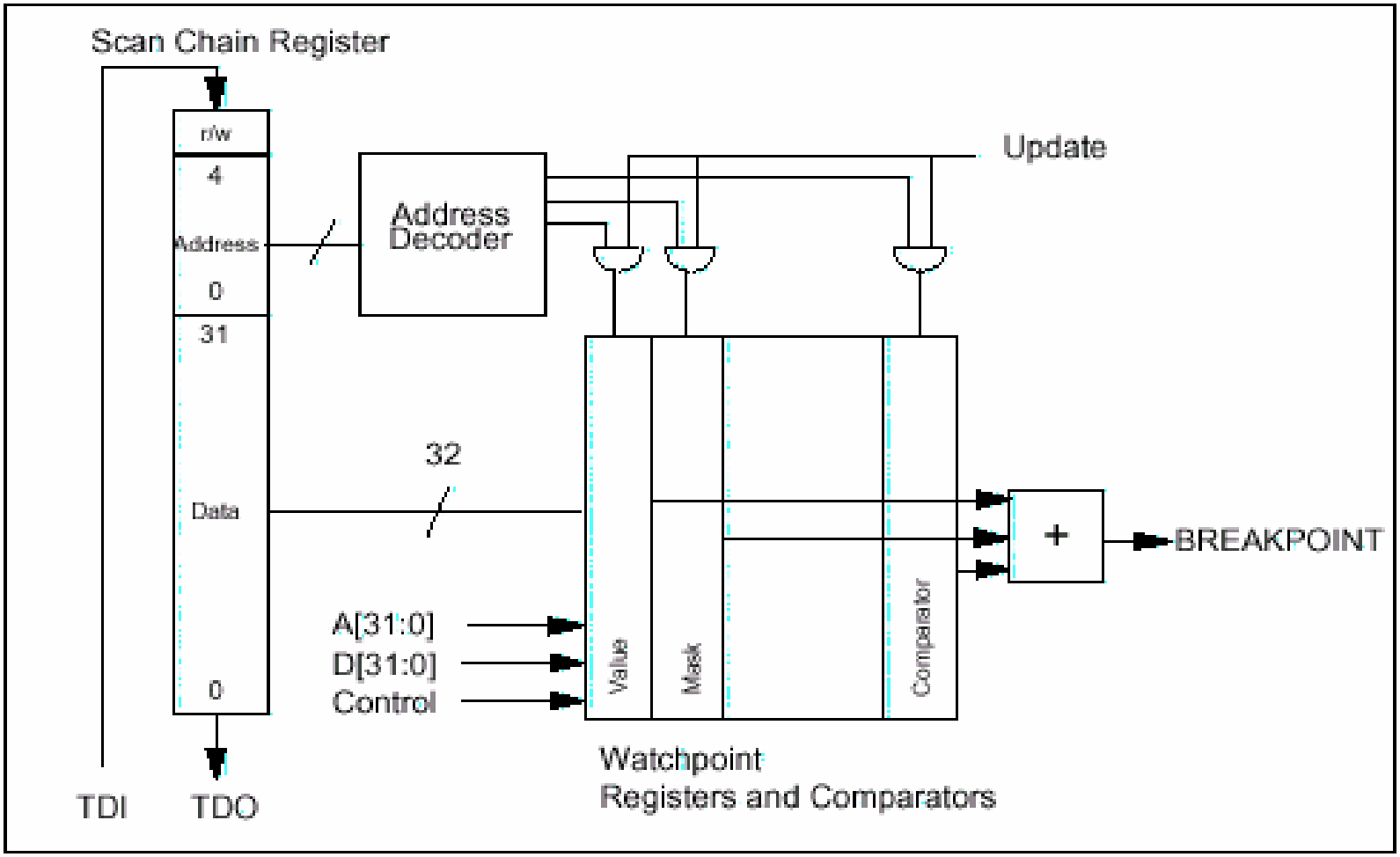
ARM7TDMI ICEBreaker

- **It is programmed through TAP controller.**
- **It consists of two real-time watchpoint units with a control and status register.**
  - **Each watchpoint unit can be configured to be a watchpoint or a breakpoint.**
- **Execution of ARM7TDMI is halted when**
  - **a match occurs between the values programmed into ICEBreaker and the values currently appearing on the address bus, data bus and various control signals.**

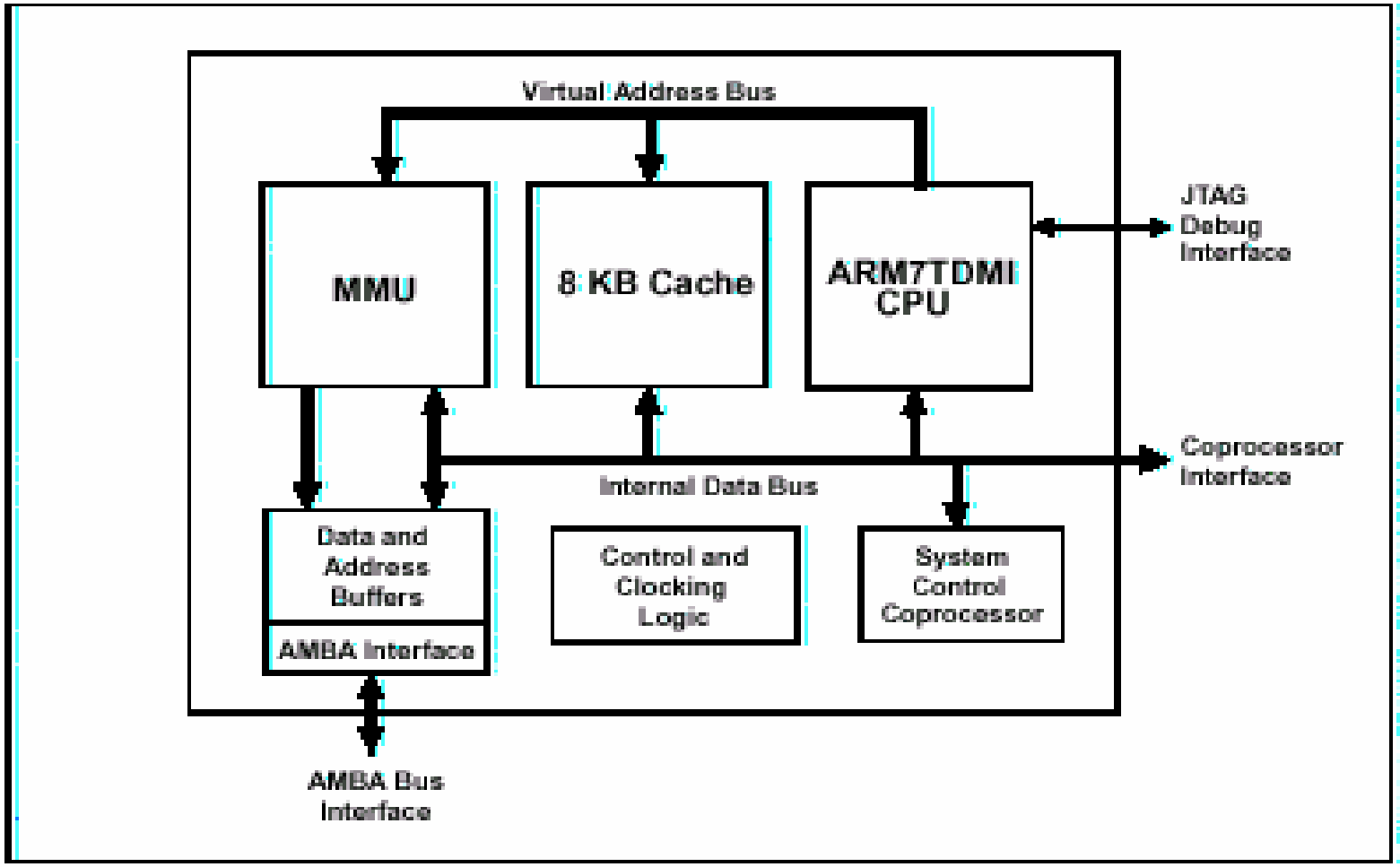
Watchpoint Registers

Address	Width	Function
00000	3	Debug Control
00001	5	Debug Status
00100	6	Debug Comms Control Register
00101	32	Debug Comms Data Register
01000	32	Watchpoint 0 Address Value
01001	32	Watchpoint 0 Address Mask
01010	32	Watchpoint 0 Data Value
01011	32	Watchpoint 0 Data Mask
01100	9	Watchpoint 0 Control Value
01101	8	Watchpoint 0 Control Mask
10000	32	Watchpoint 1Address Value
10001	32	Watchpoint 1 Address Mask
10010	32	Watchpoint 1 Data Value
10011	32	Watchpoint 1 Data Mask
10100	9	Watchpoint 1 Control Value
10101	8	Watchpoint 1 Control Mask

ICEBreaker Block Diagram



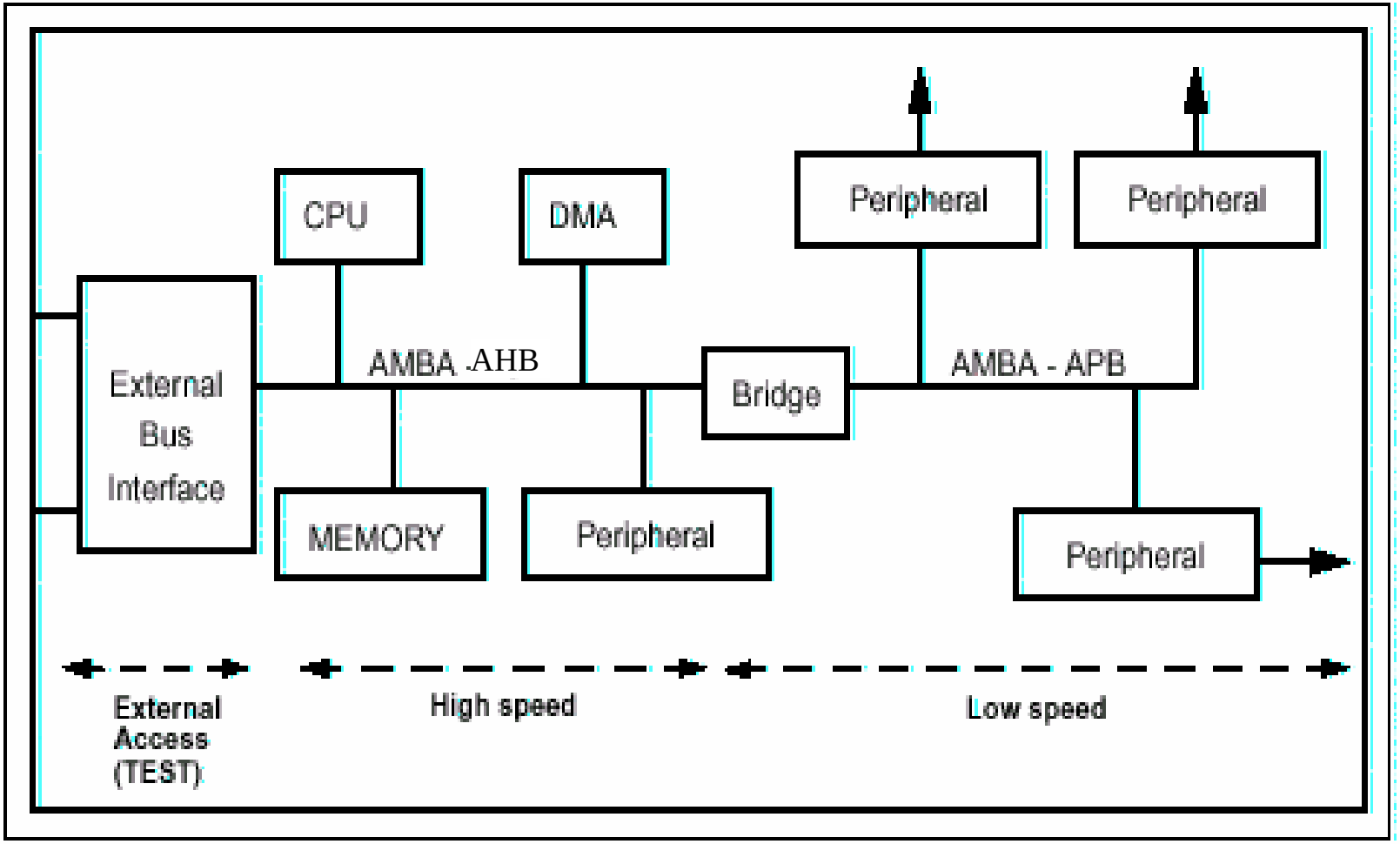
ARM720T


Virtual Address Bus
MMU
8 KB Cache
ARM7TDMI CPU
JTAG Debug Interface
Coprocessor Interface
Internal Data Bus
Data and Address Buffers
AMBA Interface
Control and Clocking Logic
System Control Coprocessor
AMBA Bus Interface

AMBA

- **AMBA: Advanced Microcontroller Bus Architecture**
  - **It is a specification for an on-chip bus, to enable macrocells (such as a CPU, DSP, Peripherals, and memory controllers) to be connected together to form a microcontroller or complex peripheral chip.**
  - **It defines**
    - **A high-speed, high-bandwidth bus, the Advanced High Performance Bus (AHB).**
    - **A simple, low-power peripheral bus, the Advanced Peripheral Bus (APB).**
    - **Access for an external tester to permit modular testing and fast test of cache RAM**
    - **Essential house keeping operations (reset/power-up, …)**

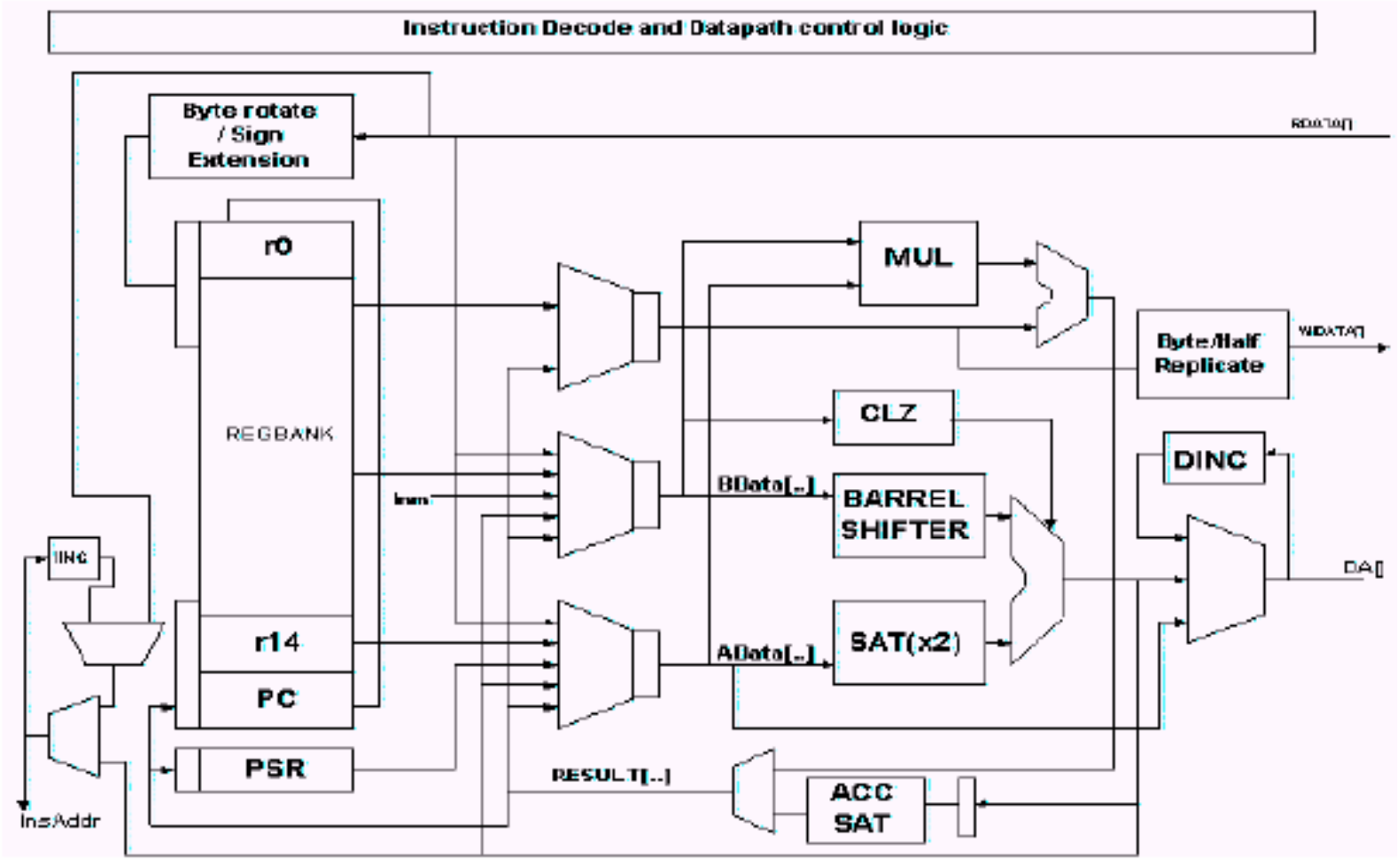
A Typical AMBA-based Microcontroller



ARM9(TDMI)

- ARM7 microarchitecture is getting old and will be replaced with ARM9
- ARM9 realizes the same (v4T) instruction set as ARM7 and is thus binary compatible
- Pipeline length is **5 stages** instead of ARM7 3 stages. This allows for faster clocking.
- Available with TDMI extensions
- ARM92x: ARM9TDMI and caches as a macrocell
- Caches are separate for instructions and data (Harvard Architecture)

ARM9 Processor Core



ARM10(TDMI)

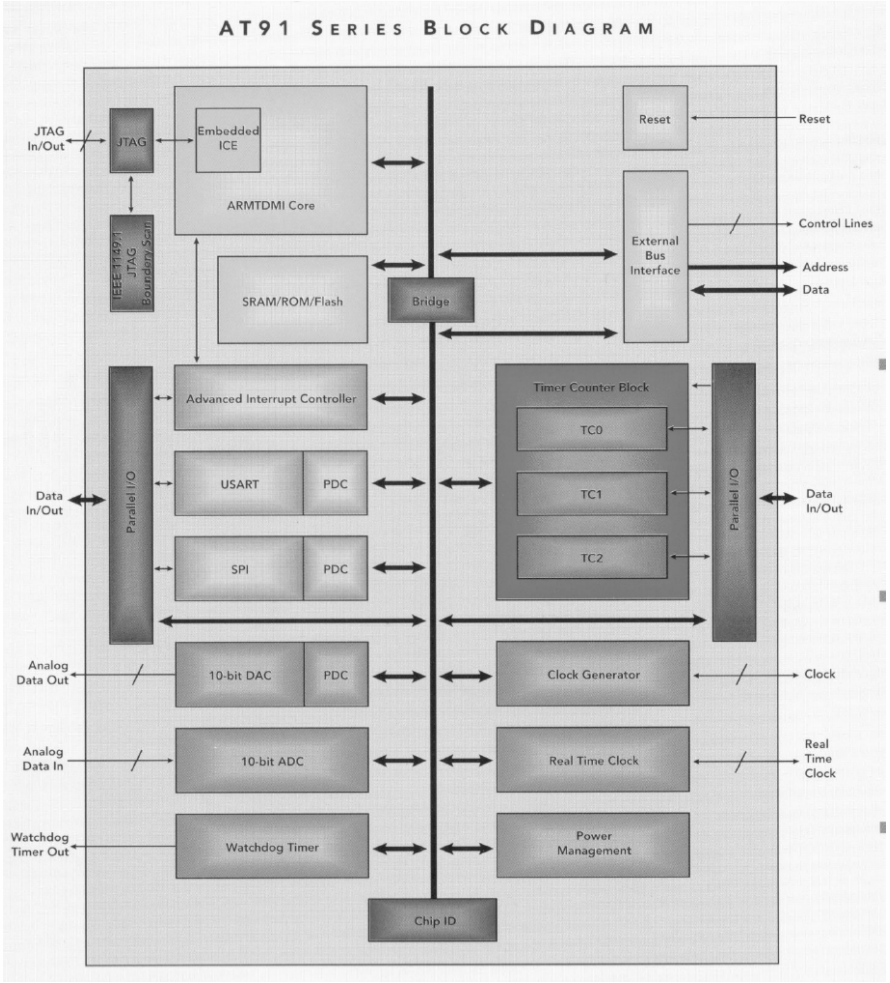
- ARM10TDMI processor core:
- Realizes the ARM instruction set with binary compatibility including the Thumb extension
- Instruction set expanded to version 5 (v5TE), 32x16 MAC-multiplier
- 6-stage pipeline for fixed point instructions

ARM as a standard component

- Even tough ARM is mostly used as a processor core in SoC and other ASICs, some manufacturers have brought ARM-based standard products to market
- Examples of manufacturers: Atmel, Cirrus Logic, Hyundai, Intel, Oki, Samsung, Sharp …
- Most of the products are based on 7TDMI-core, some are based on 720T and 920T-cores
- ARM + FPGA are implemented by Altera and Triscend
- In addition, there are a number of ASSP (Application Specific Standard Product)-chips available
  - for example to communication applications (Philips VWS22100 with ARM7-based GSM baseband chip).

Atmel ARM

- AT91-series:
  - ARM7TDMI-core
  - External bus controller
  - A load of peripherals
  - Variable amount of SRAM on die (up to 2 megabits)



Altera ARM + FPGA

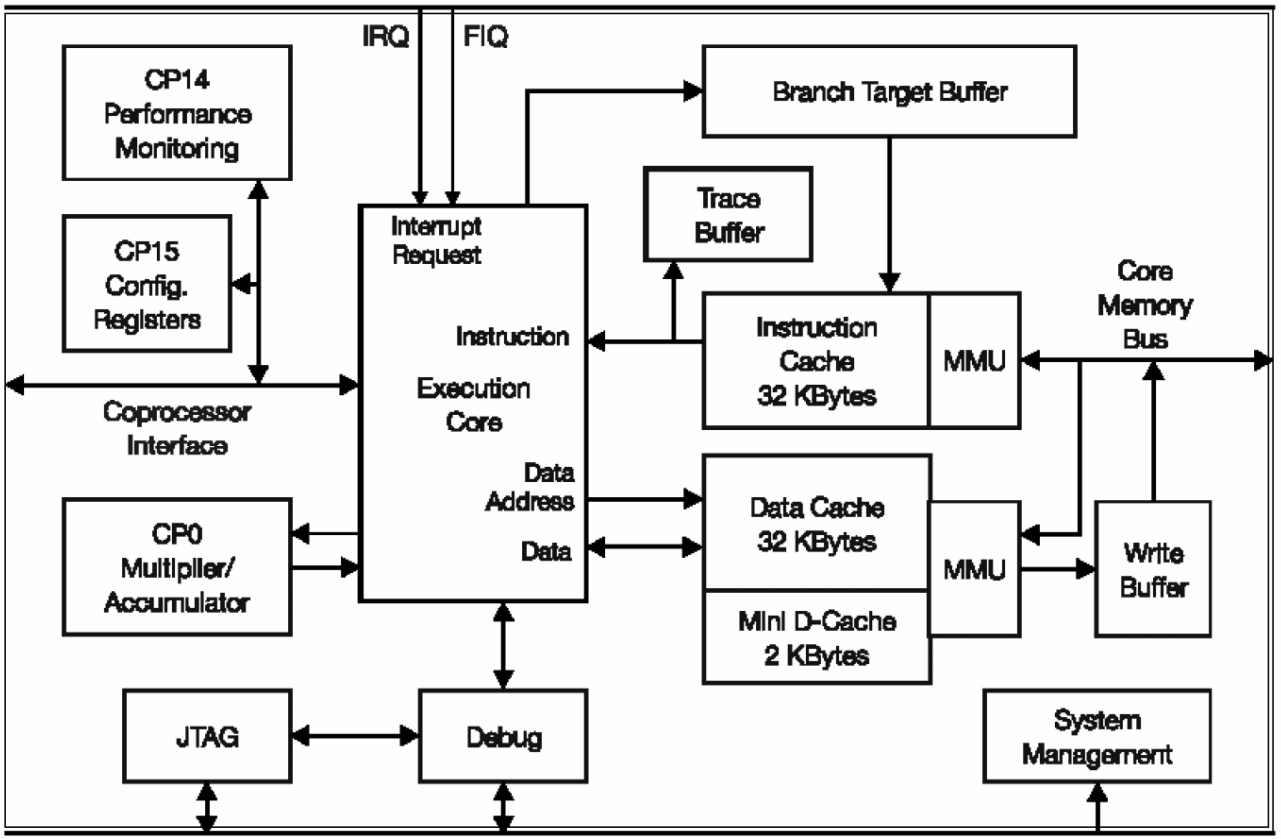
- ARM922T macrocell and programmable logic on same chip
  - System-on-a-programmable-chip

Table 1. Current ARM-Based Embedded Processor Device Features (1)

Feature	EPXA1	EPXA4	EPXA10
Maximum system gates	263,000	1,052,000	1,772,000
Typical gates 100,000	400,000	1,000,000	
Logic elements (LEs)	4,160	16,640	38,400
Embedded system blocks (ESBs)	26	104	160
Maximum RAM bits	53,248	212,992	327,680
Maximum macrocells	416	1,664	2,560
Maximum user I/Os	178	360	521
Single-port SRAM	32 Kbytes	128 Kbytes	256 Kbytes
Dual-port SRAM 16 Kbytes	64 Kbytes	128 Kbytes	

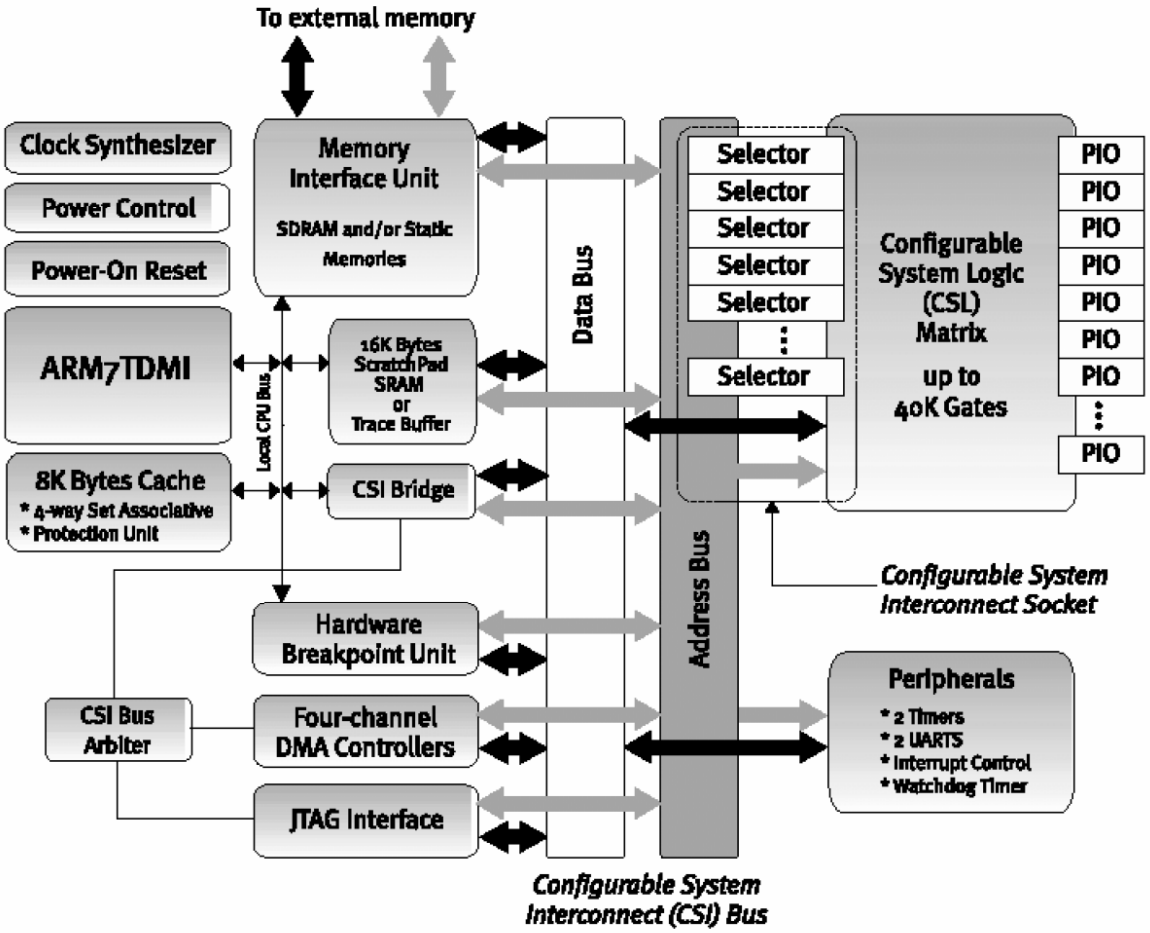
Intel ARM derivatives

- StrongARM
  - DEC developed ARM variant
  - Being phased out with XScale
- XScale
  - ARM v5TE instruction set
  - Intel developed microarchitecture
  - Coprocessor instructions used for extensions



Triscend ARM + FPGA

- Triscend A7:



Block Diagram of the Triscend A7 Configurable System-on-Chip