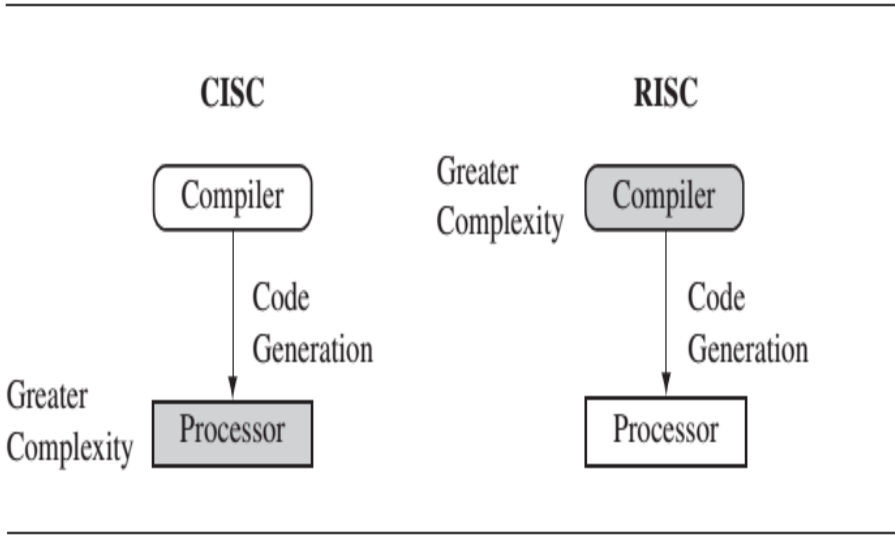
BTEC-701 Embedded Systems

ARM

- Almost every one of us owns one.
- ARM cores are widely used in mobile phones, handheld organizers, and a multitude of other everyday portable consumer devices.
- First Prototype came out in 1985.
- ARM core is not a single core but a family of designs
- ARM is based on RISC design philosophy

RISC Design Philosophy

- ARM cores use RISC Architecture
- Simple instruction which are fast and take one machine cycle to execute
- Aim is to reduce the hardware complexity but to more put more intelligence into the software
- On the other hand CISC relies more on Hardware than software



Major design rules for RISC
1. Instructions

- Reduced Number of instructions
- Takes Single cycle to execute
- Each instruction is of fixed length which facilitates the pipeline process
- CISC instructions are often of variable size and take many cycles to execute

2.Pilelines

- The processing of instructions is broken down into smaller units
- These smaller units can be executed in parallel by pipelines
- Ideally pipeline advances by one step in each cycle

3.Registers

- RISC machines have a large general-purpose register set.
- Any register can contain either data or an address.
- Registers act as the fast local memory store for all data processing operations.
- In contrast, CISC processors have dedicated registers for specific purposes.

Load Store Architecture

- The processor operates on data held in registers.
- Separate load and store instructions transfer data between the register bank and external memory.
- Memory accesses are costly, so separating memory accesses from data processing provides an advantage because you can use data items held in the register bank multiple times without needing multiple memory accesses.
- Only allows memory to be accessed by load and store operations.
- In contrast, with a CISC design the data processing operations can act on memory directly.

ARM Design Philosophy

- The ARM design philosophy is aimed at Portable embedded systems operating on battery
- Reduced power consumption
- High code density due to memory constraints
- Low cost memory
- Minimizing the size

More on ARM

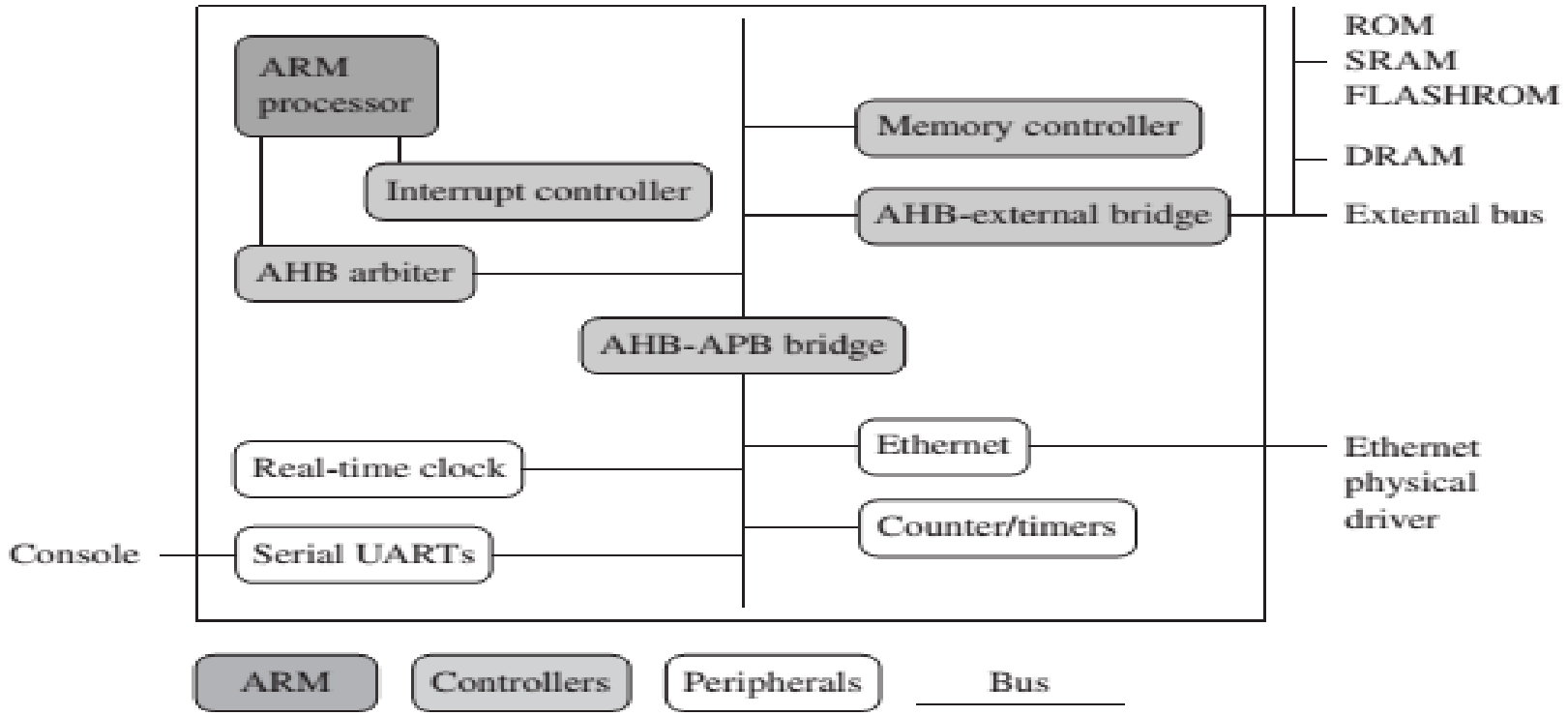
- ARM has incorporated hardware debug technology within the processor.
- Software engineers can view what is happening while the processor is executing code. With greater visibility.
- They can resolve issues faster which has a direct effect on the time to market and reduces overall development costs.
- The ARM core is not a pure RISC architecture because of the constraints of its primary application—the embedded system.
- In some sense, the strength of the ARM core is that it does not take the RISC concept too far.
- In today's systems the key is not raw processor speed but total effective system performance and power consumption.

Instruction Set for ARM Processor

- Variable cycle execution for certain instructions
- Not every ARM instruction executes in a single cycle. For example, load-store-multiple instructions vary in the number of execution cycles depending upon the number of registers being transferred. The transfer can occur on sequential memory addresses, which increases performance since sequential memory accesses are often faster than random accesses.
- Code density is also improved since multiple register transfers are common operations at the start and end of functions.
- Inline barrel shifter leading to more complex instructions—The inline barrel shifter is a hardware component that preprocesses one of the input registers before it is used by an instruction. This expands the capability of many instructions to improve core performance and code density.

- Thumb 16-bit instruction set—ARM enhanced the processor core by adding a second 16-bit instruction set called Thumb that permits the ARM core to execute either 16- or 32-bit instructions. The 16-bit instructions improve code density by about 30% over 32-bit fixed-length instructions.
- Conditional execution—An instruction is only executed when a specific condition has been satisfied. This feature improves performance and code density by reducing branch instructions.
- Enhanced instructions—The enhanced digital signal processor (DSP) instructions were added to the standard ARM instruction set to support fast 16×16-bit multiplier operations and saturation. These instructions allow a faster-performing ARM processor in some cases to replace the traditional combinations of a processor plus a DSP.

Example of an ARM based Embedded Device



The example device shown in Figure has three buses: an AHB bus for the high performance peripherals, an APB bus for the slower peripherals, and a third bus for external peripherals, proprietary to this device. This external bus requires a specialized bridge to connect with the AHB bus.

- The ARM processor controls the embedded device. Different versions of the ARM processor are available to suit the desired operating characteristics. An ARM processor comprises a core (the execution engine that processes instructions and manipulates data) plus the surrounding components that interface it with a bus. These components can include memory management and caches.
- Controllers coordinate important functional blocks of the system. Two commonly found controllers are interrupt and memory controllers.
- The peripherals provide all the input-output capability external to the chip and are responsible for the uniqueness of the embedded device.
- A bus is used to communicate between different parts of the device.

ARM Bus Technology

- Embedded systems use different bus technologies than those designed for x86 PCs.
- The most common PC bus technology, the Peripheral Component Interconnect (PCI) bus, connects such devices as video cards and hard disk controllers to the x86 processor bus.
- This type of technology is external or off-chip (i.e., the bus is designed to connect mechanically and electrically to devices external to the chip) and is built into the motherboard of a PC.
- In contrast, embedded devices use an on-chip bus that is internal to the chip and that allows different peripheral devices to be interconnected with an ARM core.

- There are two different classes of devices attached to the bus. The ARM processor core is a bus master—a logical device capable of initiating a data transfer with another device across the same bus.
- Peripherals tend to be bus slaves—logical devices capable only of responding to a transfer request from a bus master device.
- A bus has two architecture levels. The first is a physical level that covers the electrical characteristics and bus width (16, 32, or 64 bits). The second level deals with protocol—the logical rules that govern the communication between the processor and a peripheral.
- ARM is primarily a design company. It seldom implements the electrical characteristics of the bus, but it routinely specifies the bus protocol.

AMBA Bus Protocol

- The Advanced Microcontroller Bus Architecture (AMBA) was introduced in 1996 and has been widely adopted as the on-chip bus architecture used for ARM processors.
- The first AMBA buses introduced were the ARM System Bus (ASB) and the ARM Peripheral Bus (APB).
- Later ARM introduced another bus design, called the ARM High Performance Bus (AHB).
- Using AMBA, peripheral designers can reuse the same design on multiple projects.
- This plug-and-play interface for hardware developers improves availability and time to market.

Memory

- An embedded system has to have some form of memory to store and execute code.
- Price, performance and power consumption are critical parameters of memory
- Hierarchy, width and type are the specific memory characteristics to look for.

Hierarchy

- All computer systems have memory arranged in some form of hierarchy.
- The fastest memory cache is physically located nearer the ARM processor core.
- Slowest secondary memory is located further away.
- Generally closer is the memory to the core more is the cost and smaller its capacity.
- Cache memory is used speed up data transfer between the processor and the main memory
- The main memory is large depending upon the application.
- Secondary memory is largest.

Width

- The memory width is the number of bits the memory returns on each access.
- This can be 8, 16, 32, or 64 bits.
- The memory width has a direct effect on the overall performance and cost ratio.
- 32 bit arm processor will have to execute two memory fetch cycles in order to fetch 32 bit instruction from a 16 bit memory chip.

Fetching instructions from memory.

Instruction size	8-bit memory	16-bit memory	32-bit memory
ARM 32-bit	4 cycles	2 cycles	1 cycle
Thumb 16-bit	2 cycles	1 cycle	1 cycle

TYPE

- ROM contains the an image that is permanently set at production time and cannot be reprogrammed.
- ROMs are used to hold the boot code.
- Flash ROM cab be written or read, used to hold device firmware
- DRAM and SRAM
- DRAM is low cost needs refresh cycles and is slower
- SRAM is fast but requires more silicon area, no refreshing required, High cost
- SDRAM is synchronous DRAM with higher speed.

Peripherals

- Peripherals device performs the input and output functions
- All ARM peripherals are memory mapped
- Controllers are specialized peripherals
- Two important type of controllers are memory controller and interrupt controller.
- Memory controllers connect different types of memory to the processor bus.
- Interrupt controller provides a policy about the device that can interrupt the processor at specific time.
- ARM processor have standard interrupt controller and vector interrupt controller.