SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)
Coimbatore-35

DEPARTMENT OF BIOMEDICAL ENGINEERING

19BMB303 & Fundamentals of Microprocessors and Microcontrollers

Unit V - 32- BIT ARM PROCESSOR

III Year/ VI Sem

Dr. K. Manoharan,
ASP / BME / SNSCT

MICROCONTROLLER BASED SYSTEM DESIGN

Reduced Instruction Set Computer

Design Physiology

RISC Vs CISC Architecture

ARM Processor Architecture

ARM Core data flow model, Barrel Shifter

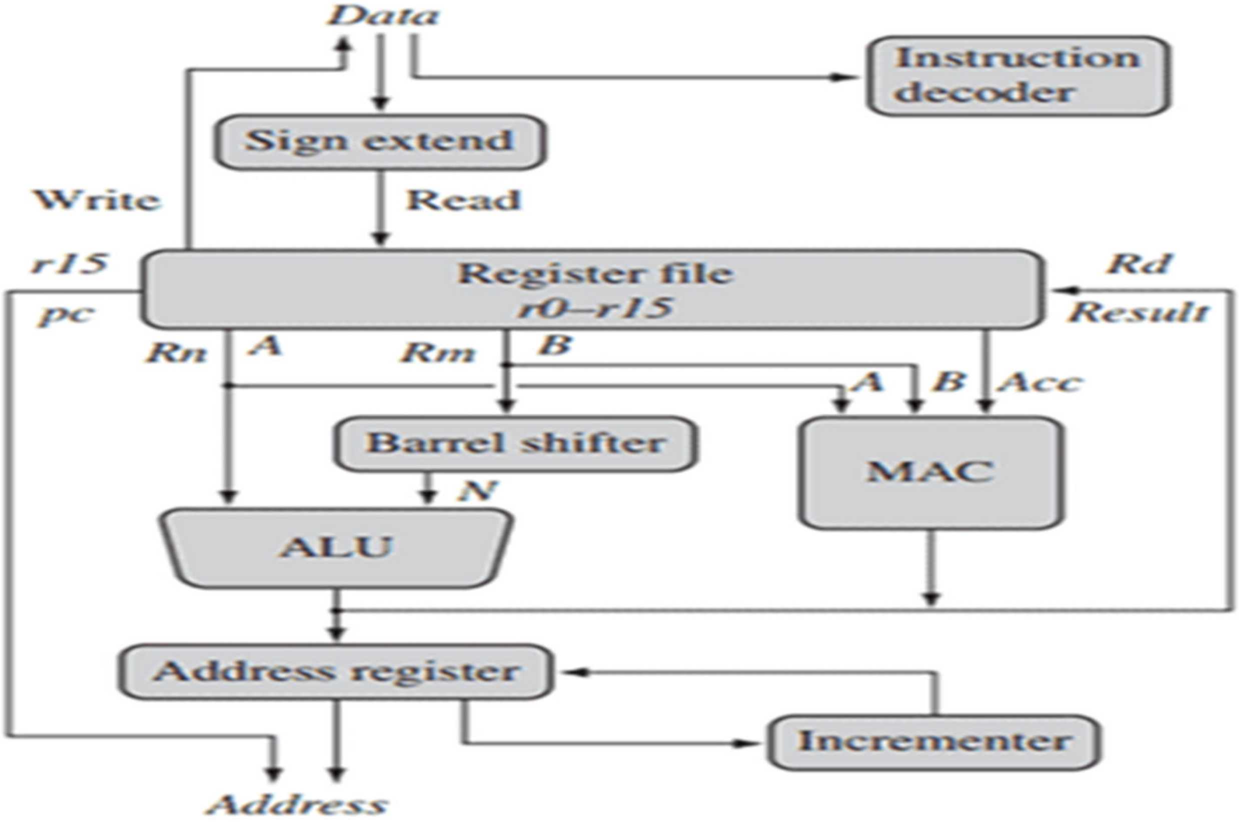
ARM processor modes and families

Pipelining

ARM instruction Set and its Programming

Pulse oximeter using ARM processor

ARM Core data flow model, Barrel Shifter



ARM Core data flow model, Barrel Shifter

As depicted in figure, an ARM core dataflow model may be thought of as functional units connected by data buses, with the arrows representing data flow, the lines representing buses, and the boxes representing either an operating unit or a storage region.

The diagram depicts both the data flow and the abstract components that make up an ARM core. The Data bus is where data enters the CPU core.

The data might be an executable command or a data object.

The ARM is implemented by Von Neumann in figure, with data items and instructions sharing the same bus.

The ARM implementations at Harvard, on the other hand, employ two separate buses.

Before instructions are executed, the instruction decoder interprets them.Each instruction that is executed is part of a certain instruction set.

ARM Core data flow model, Barrel Shifter

The load-store architecture is used by the ARM processor, as it is by all RISC processors.

This implies there are two sorts of instructions for moving data in and out of the processor: load instructions copy data from memory to core registers, while store instructions copy data from registers to memory.

Data processing instructions that directly manipulate data in memory are not available.

As a result, data processing is limited to registers.

The register file—a storage bank made up of 32-bit registers—is where data objects are stored.

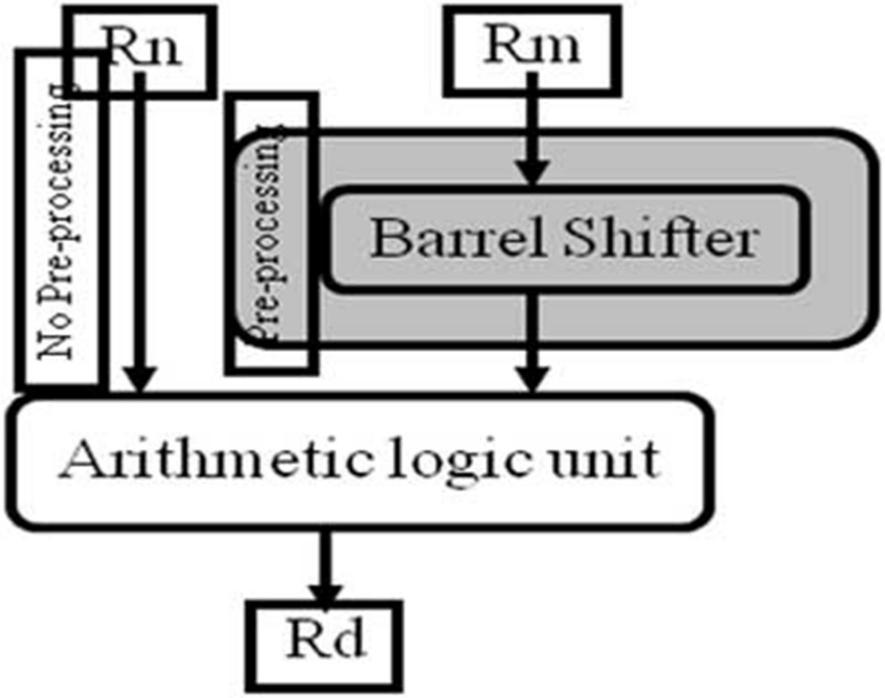
Most instructions consider the registers as having signed or unsigned 32-bit values since the ARM core is a 32-bit processor.

ARM Core data flow model, Barrel Shifter

- ✓ When signed 8-bit and 16-bit integers are read from memory and stored in a register, the sign extend hardware transforms them to 32-bit values.
- ✓ Rn and Rm are the two source registers of ARM instructions, while Rd is the result or destination register.
- ✓ The internal buses A and B are used to read source operands from the register file.
- ✓ The ALU (arithmetic logic unit) or MAC (multiply-accumulate unit) computes a result using the Rn and Rm register values from the A and B buses.
- ✓ The outcome of data processing instructions is written directly to the register file in Rd.
- ✓ The ALU is used in load and store instructions to create an address that is stored in the address register and broadcast over the Address bus. Because the ARM7 architecture is based on the von Neuman architecture, the same bus is used to load both instructions and data.

Barrel Shifter

- An **ARM Barrel Shifter** is a special hardware block inside ARM processors that efficiently performs **bit shifts** and **rotations** during instruction execution — without needing a separate shift instruction first. It's tightly integrated with the ALU (Arithmetic Logic Unit)



Barrel Shifter

It supports different shift types:

- **LSL (Logical Shift Left)** – shifts bits left, inserting zeros.
- **LSR (Logical Shift Right)** – shifts bits right, inserting zeros.
- **ASR (Arithmetic Shift Right)** – shifts bits right, preserving the sign bit (good for signed numbers).
- **ROR (Rotate Right)** – rotates bits to the right.
- **RRX (Rotate Right with Extend)** – rotates right through the carry flag.