EEE1024: Fundamentals of Electrical and Electronics Engineering

Dr. Sanchit Khatavkar

Course Outline

DC Circuits – Basics

AC Circuits - Basics

Digital Systems

Semiconductor Devices

µ processor & µ controller

Sensors

Comm. Systems

Microprocessor & Microcontroller

Overview of *ARM* Architecture

What is ARM?

- ARM is a CPU architecture
(A family of related CPU architectures).

- If this CPU is put on the chip at by itself,
  then → microprocessor

- If you combine it with ROM, RAM and peripherals on one chip, then → microcontroller

ARM (Advance RISC Machine) History

- First ARM processor was developed in the year 1978 by Cambridge University.

- Project taken up in 1983, to replace 8-bit 6502 microprocessor from BBC computers. First ARM RISC processor was produced by the *Acorn* Group of Computers in the year 1985.

- In 1990, new company - *ARM* was formed, which was jointed owned by *Acorn*, *Apple* and *VLSI*

- RISC - Reduced Instruction Set Computer
  (Intel used a CISC - Complex Instruction Set Computing)

- RISC - use only one cycle to execute a command, reduces functions.

Why ARM ?

- One of the most licensed and thus widespread processor cores in the world
  - Used in cell phones, multimedia players, handheld game console digital TV and cameras , digital TV and cameras
  - ARM7: GBA, iPod
  - ARM9: NDS PSP Sony Ericsson BenQ , PSP, Sony Ericsson, BenQ
  - ARM11: Apple iPhone, Nokia N93, N800
  - 90% of 32-bit embedded RISC processors till 2010

- Used especially in portable or battery-operated devices due to its low power consumption and reasonable performance

Embedded system is a combination of computer hardware and software designed for a specific function or functions within a larger **system**


ATAP Partners
Tools Partners
ARM
in Partnership
RTOS Partners
Software Partners

ARM Powered Products

About ARM Processors

- A simple RISC-based architecture with powerful design

- A whole family of ARM processors exist –

share similar design principles and common instruction set (backward compatibility)

- Design philosophy

➢ *Small processor* (size) for low power consumption (suitable for embedded applications)

➢ *High code density* (Instruction and Data in same memory, space scarcity) for limited memory and physical size restrictions.

➢ Can *interface* with slow and low-cost memory systems

➢ *Reduced die size* for processor to accommodate more peripherals.

Popular ARM architectures

- ARM7TDMI
  - 3 pipeline stages (fetch/decode/execute) – how instructions are executed
  - High code density/low power consumption
  - One of the most used ARM-version for low-end systems – where high power not required
  - All ARM cores after ARM7TDMI include TDMI even if they do not include TDMI in their labels
- ARM9TDMI
  - Compatible with ARM7
  - 5 stages (fetch/decode/execute/memory/write)
  - Separate instruction and data cache (Instruction and Data in same memory till ARM7)
- ARM10

  - 6 stages (fetch/issue/decode/execute/memory/write)

ARM Family Comparison

ARM family attribute comparison.

year	1995	1997	1999	2003
	ARM7	ARM9	ARM10	ARM11
Pipeline depth	three-stage	five-stage	six-stage	eight-stage
Typical MHz	80	150	260	335
mW/MHz[a]	0.06 mW/MHz	0.19 mW/MHz (+ cache)	0.5 mW/MHz (+ cache)	0.4 mW/MHz (+ cache)
MIPS[b]/MHz	0.97	1.1	1.3	1.2
Architecture	Von Neumann	Harvard	Harvard	Harvard
Multiplier	8 × 32	8 × 32	16 × 32	16 × 32

Clock frequency

Power

Throughput

$$Power \propto \frac{1}{Clock}$$

Throughput – how fast instructions can be executed

MIPS – Million instructions per second

ARM RISC

- RISC: simple but powerful instructions that execute within a single cycle at high clock speed.

- Four major design rules:
  - *Instructions*: reduced set/single cycle/fixed length(decoding easy)
  - *Pipeline*: decode in one stage/no need for microcode (complicated program)
  - *Registers*: a large set of general-purpose registers(GPRs) (data can be stored temporarily in between calculations)
  - *Load/store architecture*: Data processing (ALU)instructions apply to registers only (-they do not access memory); load/store to transfer data between registers and memory

- The distinction blurs because modern day CISC (Complex Instruction Set Computer) implements RISC concepts

RISC – Load/Store

Load-Store architecture:

Instructions are classified into 2 categories –

Memory access (load and store between memory and registers) and

ALU operations (which only occur between registers)

For example –

In a load–store approach, both operands and destination for an ADD operation must be in registers.

In register–memory architecture (for example, a CISC instruction set architecture such as x86) in which one of the operands for the ADD operation may be in memory, while the other is in a register

ARM specific features – which differ from RISC

- Variable cycle execution for certain instructions –

  (multiple-register load/store for higher code density)
- Inline barrel shifter leading to more complex instructions -

  (improves performance and code density)
- Thumb 16-bit instruction set:

  When 32-bit power is not needed, it can work with 16-bit thumb, resulting in 30% code density improvement

  (32-bit instructions that can be freely intermixed with 16-bit instructions in a program.)

- Conditional execution –

  reduces branches and improves performance

  (Add 2 numbers provided '0' flag is Set. This is common in other architectures' branch or jump instructions but ARM allows its use with most mnemonics.)

  Mnemonics:

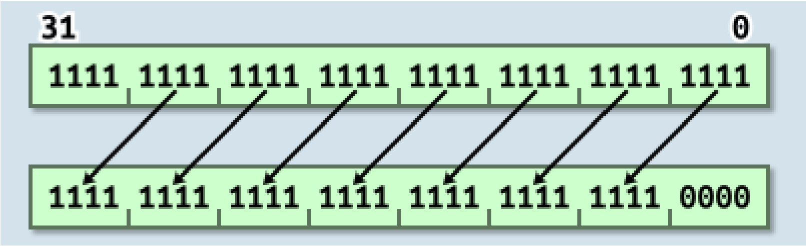
  ADD for add and CMP for compare
- Enhanced instructions –

  additional functions like MULTIPLY and ADD especially for DSP applications (-from voice to audio to sensor hubs to machine learning (ML))

ARM – Barrel Shifting

•Barrel shifter is a hardware that allows multiple bit shifting in 1 cycle.

•It performs SHIFT and ROTATE operations in *ARM* processors - 5 types



1. Logic Shift Left (LSL),

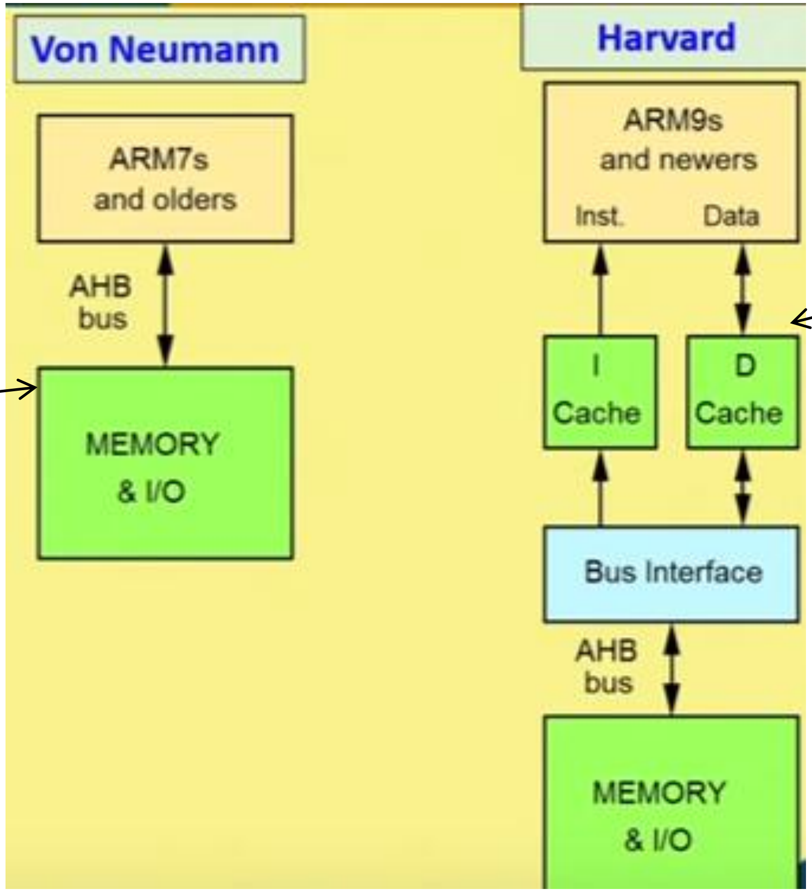
2.LSR,

3. Arithmetic Shift Right (ASR),

4. ROR,

5. RRX

Architectures



Single

Instruction and Data

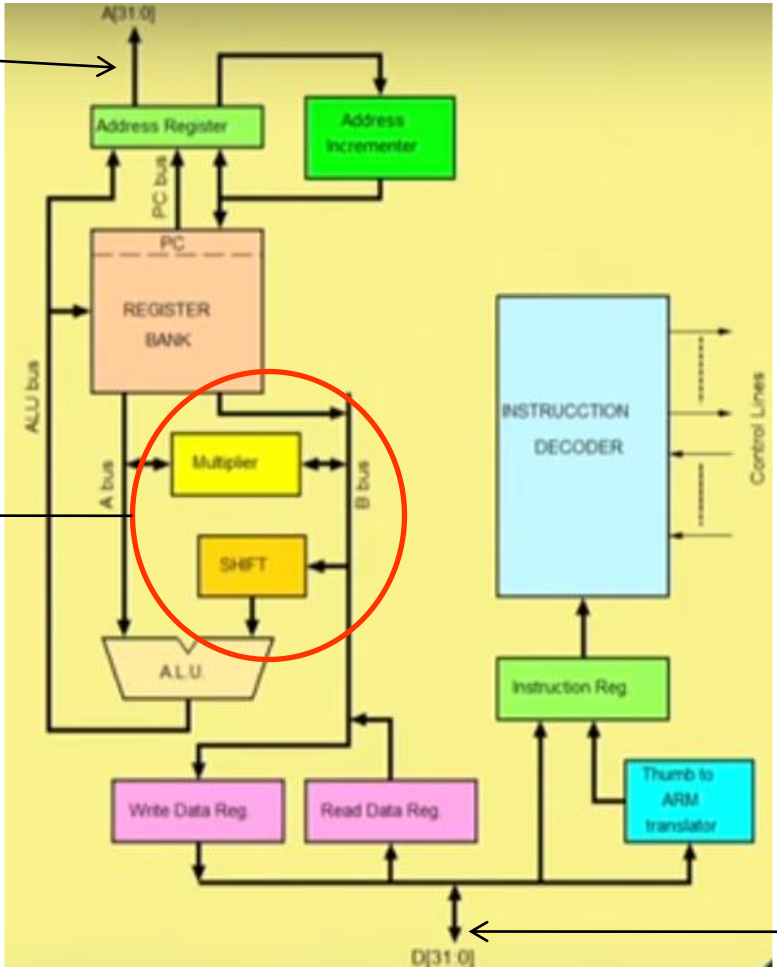
Memory-mapped I/O:

- No specific instructions for I/O
- Use Load/Store instr. for I/O
- Peripheral's registers at some memory addresses

Some part of memory is reserved for I/O

ARM7 Architecture

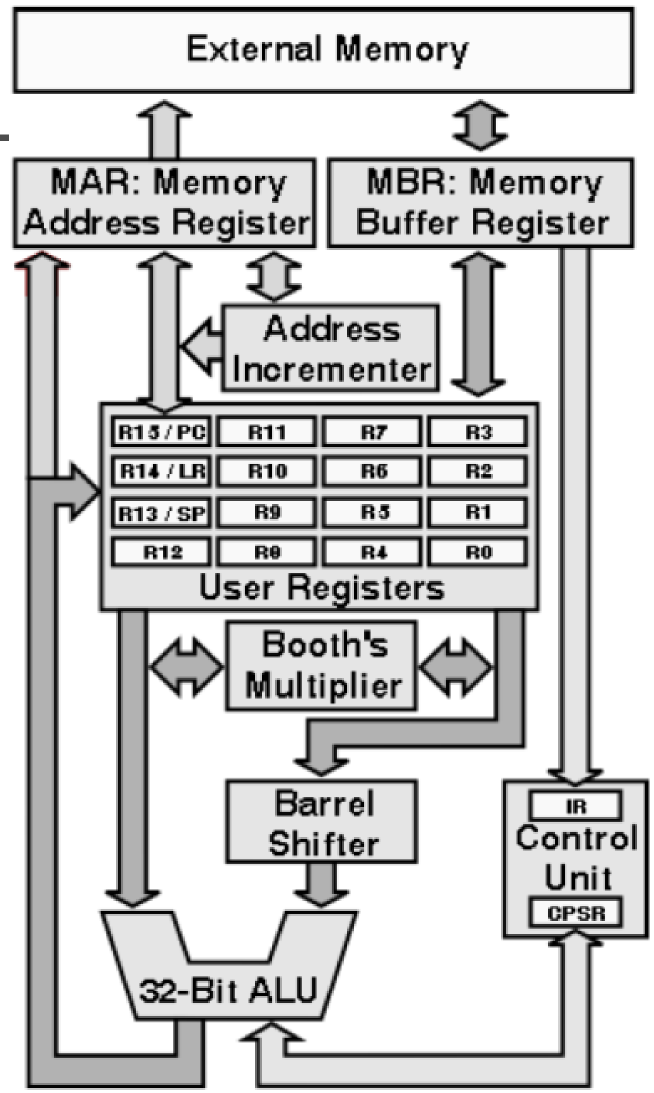
Address Bus



Unique feature of ***ARM***

Data Bus

ARM Architecture!



Acknowledgements

1. https://www.watelectronics.com/arm-processor-architecture-working/
2. http://www.davespace.co.uk/arm/introduction-to-arm/barrel-shifter.html
3. https://developer.arm.com/documentation/dui0471/i/key-features-of-arm-architecture-versions/thumb-2-technology
4. https://www.arm.com/why-arm/technologies/dsp

- Overview of ARM architecture
- Different modes of ARM processor
- Various instructions
- 8051 Microcontroller architecture
- Applications