EEE3096S: Embedded Systems II

LECTURE 6:

THE ADVANCED RISC MACHINE (ARM)

Presented by:

Dr Yaaseen Martin

Electrical Engineering
University of Cape Town

The origins of ARM….

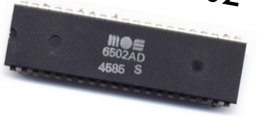
Acorn Computers

- Started in 1978 as a subsidiary of Cambridge Processing Unit, Ltd.
- The company's first project was a slot machine, developed on a tiny budget
- Bought by Olivetti in 1985

Olivetti today: Italian corporation that specialises in very expensive office equipment for the European market — especially very highly priced printers, fax machines and such.

ACORN COMPUTERS BBC MICRO

BBC Micro's processor:
Motorola 65002



Acorn's BBC Micro, from 1982, was one of the best home computers of that decade; powered by Motorola 65002 processor

ARM HISTORY

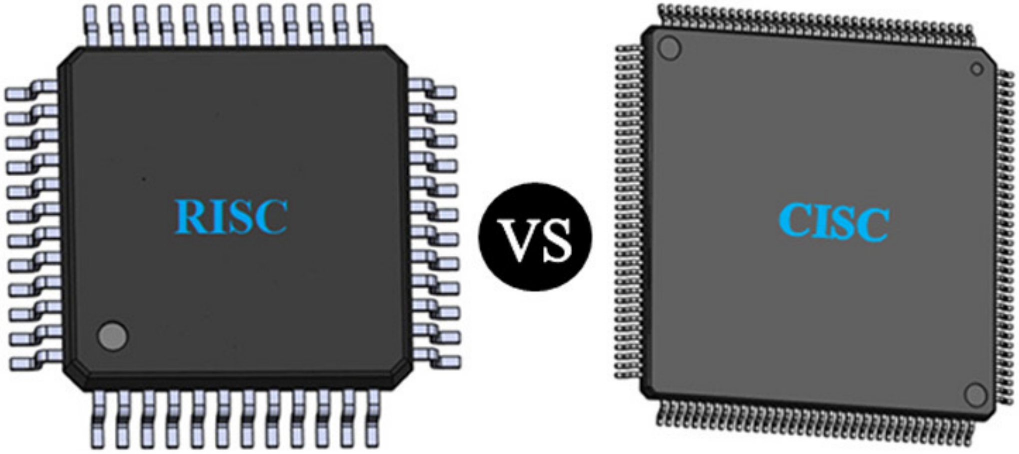
- 1983: Acorn begins design of a processor - the Acorn RISC* Machine
- 1985: The ARM1 is launched, fabricated by VLSI Technology
- 1987: ARM processor debuts in a commercial product
- 1990: ARM Ltd spun off from Acorn with support from Apple
- 1991: ARM6 launched – ARM's first embeddable RISC core

ARM = Advanced RISC Machine

* RISC = Reduced Instruction Set Computer

RISC VS CISC

- **<u>RISC:</u>** Reduced Instruction Set Computer
- **<u>CISC:</u>** Complex Instruction Set Computer
- RISC has **fewer** instructions, therefore needs **more** code/software
- CISC has **more** instructions, therefore needs **less** code/software
- Majority of today's microprocessors are RISCs



CHARACTERISTICS OF CISC

- Large number of instructions (100 – 250 instructions)
- Some instructions perform specialized tasks and are used only infrequently
- Many different addressing modes (say 5 or more)
- **Variable-length** instruction formats
- Possibly **variable length instruction execution cycles** (e.g., ADD taking more clock cycles than LOAD)
- Instructions that **manipulate operands directly in memory** (e.g. add value at address X to reg. A)

CHARACTERISCTICS OF RISC

- Small number of instructions (usually <100)
- Just enough types of instructions (or close to minimal)
- Few addressing modes (maybe be just two or three)
- **Fixed-length instruction formats**, easy to decode
- **All instructions take the same number of clock cycles** to complete (typically)
- All **operations** (e.g., ADD, COMPARE, etc) **work only with registers** (no memory access)
- Often use micro-programmed control (i.e., CPU instructions a set of even lower level instructions)
- Memory access limited to LOAD and STORE instructions (i.e. no instructions manipulate operands in memory)

THINKING ABOUT ALTERNATIVE PROCESSOR OPTIONS

- You are aware of ARM (e.g., STM, Texas Instruments, NXP/Freescale, Intel)
- An ES developer should also be aware of these often-used microcontrollers:
  - **PIC** (Peripheral Interface Controller)
  - **AVR** (or ‘Alf and Vegard's RISC processor’, an Atmel)

PIC and AVR are essential for designing low-power, low-cost systems!

KEY FEATURES OF PIC

- Developed by Microchip Technology
  - (Derived from the PIC1650 originally designed by General Instrument's Microelectronics Division)
- Available since 1976*
- Some use pure Harvard Architecture, i.e. program memory is protected; others use Modified Harvard where the program memory can be read (and sometimes even written)
- All models use flash memory for program storage
- Low cost, low power, ease of reprogramming with built-in EEPROM
- Abundant development tools and application notes
- Often marketed as "PICmicro"
- A long history of use, probably the most used microcontroller of all time!

KEY FEATURES OF AVR

- AVR – “Alf and Vegard's RISC processor”
  - Modified Harvard architecture 8-bit RISC
  - Even lower power options, generally low cost (maybe the **cheapest AVR is not as cheap as the cheapest PIC**)
  - ATmega328P one of the most popular, very low power but fairly powerful micros
  - Major claim to fame: one of the first micros to have on-chip flash, instead of one-time programmable ROM or hassles of using EEPROM
- Inspiring historical fact:
  - The AVR architecture was conceived by two students from the Norwegian Institute of Technology (NTH), Alf-Egil Bogen and Vegard Wollan*
  - The name “AVR” is commonly thought to stand for “Alf and Vegard's RISC” processor

QUICK ACTIVITY!

Which micro-processor (PIC, AVR, ARM) would be best suited to each of these products?
(Power and price: ARM > AVR > PIC)

- ABS brakes
- Radio control of a model airplane
- Car radio (e.g., auto tuning, stored stations)
- Washing machine
- Portable battery-operated game console
- Weather sensor
- Garage door opener
- iPad
- Burglar alarm

… and which of these applications might you instead use an AVR or PIC? Or even some other option?

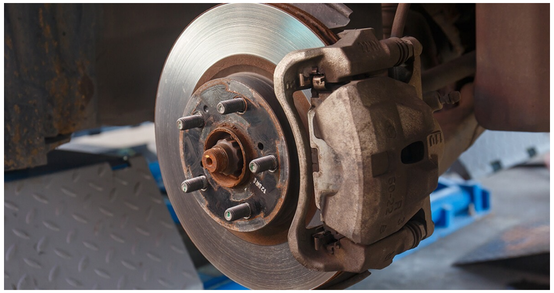
I’d probably make the following choices…

- ABS brakes - ARM or AVR
- Radio control of a model airplane – PIC or AVR
- Car radio – PIC for auto turning feature at least
- Washing machine – Probably a PIC
- Portable battery-operated game console - ARM
- Weather sensor – PIC (lowest power options, usually doesn’t need speed)
- Garage door opener – PIC or AVR, something cheap
- iPad – ARM
- Burglar alarm - probably a PIC, if there’s not much signal processing

WHERE ARE ARM PROCESSORS USED?

In a wide range of embedded computers and control systems e.g.

- Highly reliable – ABS braking
- Consumer – cameras, smartphones, laptops, tablets
- Gaming systems (Nintendo Switch, etc)
- Network systems (routers, switches, firewalls)



ABS brakes

iPhone

MacBook

Nintendo Switch

Network (not Nintendo) Switch

COMPANIES THAT USE ARM

Just a some of the companies that use ARM cores:

Apple, Agere, Broadcom, Fujitsu, Infineon, Intel, NEC, Nokia, Philips, Qualcomm, Texas Instruments, Toshiba, Sega, Sony, Nintendo, Cirrus Logic, Motorola, Thomson Multimedia.

ARM licenses 19 of the top 20 semiconductor companies.