ECE 471 – Embedded Systems
Lecture 3

Vince Weaver

https://web.eece.maine.edu/~vweaver

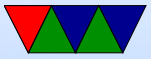
vincent.weaver@maine.edu

8 September 2025

Announcements

- HW#1 was posted, due Friday
  it's relatively easy, short answer, be sure to follow directions and submit via e-mail

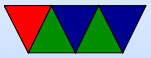
- If you have questions on the homework, you can get a faster response if you indicate it's a question in the e-mail subject



Tradeoffs

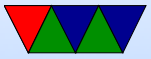
It's all about tradeoffs

- Power / Thermal
- Performance
- Availability (!!!)
- Cost
- Compatibility
- Time to Market
- Features
- Size / Footprint



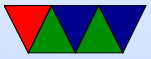
Challenges vs Regular Systems

- Programming in constrained environment (cross-compiling / emulators)
- Security
- Safety
- Real-time
- Power consumption
- Long-life (embedded device might be in use for decades)
- Testing
- Bug-fixing



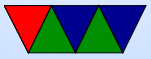
Discussion

- What concerns might you have when designing an embedded system?
  Security is a big one these days
- What language might you write your code in?
  C is still popular despite security issues.



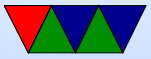
The ARM Architecture

- What does ARM stand for?
  (these days it's not an acronym anymore)
- Will ARM be around forever?
  Even the most prominent companies can collapse suddenly. Intel is currently being a cautionary example.



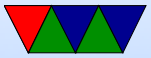
Ancient ARM History

- Acorn Computer, 1980s in UK
- Computers like Electron, Archimedes, BBC Micro
- BBC Micro used 6502
  - 6502 also used in Commodore 64, Apple II, NES, Atari 2600
  - Fun fact: 6502 design led by UMaine alum Chuck Peddle



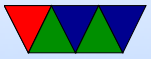
ARM: End of 8-bit era

- Acorn wanted new chip for newer machines
- MOS Technology had 16-bit 65816 (used by Apple IIgs, SNES) but it had some downsides
- Acorn first visited Natl Semiconductor who had hundreds of engineers/expensive equipment making chips
- They later visited MOS (6502) and found was only developed by a few people by hand. how hard could it be...
- Decided to make one themselves.



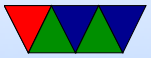
ARM: Original Design

- Furber and Wilson made ARMv1
- 27k transistors, 3um
- Low power, aiming for 1W so could use cheaper plastic vs ceramic case
- Actually was 0.1W



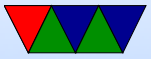
ARM: Medium History

- **A**corn **R**ISC **M**achine.
- 1990 ARM founded by Acorn, Apple, VLSI Technologies, changed to Advanced RISC Machines (at request of Apple?)
- 1998 at IPO changed to just "ARM Holdings"



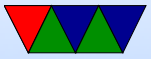
Recent ARM History

- Bought by Softbank (Japan) in 2016
- Name changed to “Arm” in 2017
- Softbank was in talks to sell ARM to NVIDIA (2020) but that fell through
- China subsidiary went rogue
- Had another IPO in 2023



ARM Business Plan

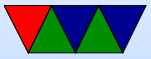
- IP Licensing company. Does not fab own chips. License to other companies
- Other companies take the design, put on SoC, attach whatever other logic blocks are needed
- Relatively small company compared to Intel which not only deigns the chip, but fabs, etc.
- Can buy full core (Cortex-AX) or just rights to ISA and make your own (Apple A14 / M1)



AMBA Bus Protocol for SoC

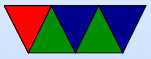
Advanced Microcontroller Bus Architecture

- Common bus, various companies can provide logic blocks for it, can swap in and out ARM cores as needed.
- ARM System Bus (ASB), ARM Peripheral Bus (APB)
- ARM High Performance Bus (AHB)
- You might recognize those prefixes from register names in ECE271



RISC / CISC Discussion

- Simple decode. Load/store. Fixed instruction width. 3-operand.
- MIPS is classic RISC
- x86 is classic CISC (with complex instructions)
  Though internally x86 executes uops, RISC
- ARM (predication, auto-increment, barrel shifter)
  Called RISC but has complex instructions



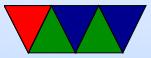
RISC / CISC Example

Memory copy: Load a byte from pointer, store byte to another pointer, increment pointers, loop until counter counted down.

CISC	RISC

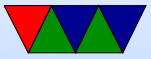
`rep movsb`	`ldb r0,[r1]`
	`add r1,r1,#1`
	`stb r0,[r2]`
	`add r2,r2,#1`
	`sub r3,r3,#1`
	`cmp r3,#0`
	`bne loop`

Note: if ARM32 can optimize a bit



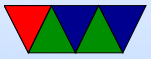
ARM Architecture vs Family (old)

- ARMv1 : ARM1
- ARMv2 : ARM2, ARM3 (26-bit, status in PC register)
- ARMv3 : ARM6, ARM7
- ARMv4 : StrongARM, ARM7TDMI, ARM9TDMI
- ARMv5 : ARM7EJ, ARM9E, ARM10E, XScale



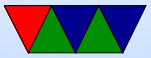
ARM Architecture vs Family (newer)

- ARMv6 : ARM11 (Raspberry Pi A/B), ARM Cortex-M0
- ARMv7 : Cortex A8, A9, A15, A7 (Pi2), Cortex-M3
- ARMv8 : (64-bit) Cortex A50, A53 (Pi3), A57, A72 (Pi4)
  ARMv8.2 A76 (Pi5)
  ARMv8.1, 8.2, 8.3, 8.4, 8.5
  ARMv8-A, ARMv8-R
- ARMv9 : Cortex-X2, Cortex-X3, Cortex-A710/A510 (big/little)



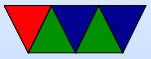
Various abbreviations in Model Names

- Modern Cortex Processors
  - "Application" ARM Cortex-A
  - "Real-time" ARM Cortex-R
  - "Micro-controller" ARM Cortex-M
- ARM7 Processors (example armv4 ARM7TDMI)
  - "E" means DSP instructions
  - "M" improved multiplier
  - "T" THUMB
  - "J" Jazelle (java bytecodes)



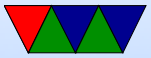
- “D” Debug
- “I” ICE (In-circuit Emulator)
- “EE” ThumbExecutionEnvironment, Just-in-time
- NEON – SIMD

- ARM11 Processors (Raspberry Pi is armv6 BCM2835 ARM1176JZF-S)
  - (All have Thumb)
  - S – Synthesizable
  - J – Java Extension
  - Z – TrustZone
  - F – Vector Floating Point Coprocessor

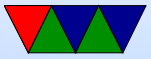


Single Board Computers

- Small boards with lots of I/O, often for use in embedded systems
- Often put out by companies trying to encourage use of their chips
- Examples
  - TS-7600 (ECE has a zillion of these)
  - Pandaboard / Beagleboard/ Beaglebone
  - Gumstix (used in 471 years ago)
  - Raspberry Pi

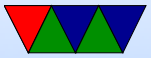


- Pi clones (Orange, Banana, etc)
- Arduino, Pi-pico, esp32
- STM boards (like used in ECE271)
- MOS KIM-1 (1976)



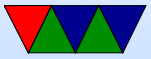
Example: STM NUCLEO-L476RG

- Used in ECE 271
- 32-bit Cortex-M4, 80MHz, FPU
- Thumb2 ISA
- Low-power (30nA shutdown, 120nA standby)
- Peripherals
  - ADC, RTC
  - Timers
  - 1MB Flash, 128k SRAM
  - USB/i2c/USART/Canbus



We'll Use Raspberry Pis in ECE471

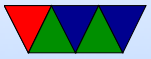
- High perf
- Relatively low cost
- Well maintained Linux port
- Relatively well documented
- Widely used so support easy to find online



Raspberry Pi

Note there are two separate (but related) organizations:

- Raspberry Pi Foundation – charitable group to encourage computer science education
- Raspberry Pi, Ltd – company that makes and sells Raspberry Pi Boards
- Their goals don't always line up
- During parts shortage extremely limited board availability, Ltd was prioritizing businesses over education institutions



What is a Raspberry Pi?

- Raspberry Pi Foundation wanted small board to encourage CS in schools
- Easy to use and cheap enough that students can experiment without worrying too much about bricking it
- Back in the day small micro-computers encouraged hacking, modern Windows systems not so much
- There are other small embedded boards (BeagleBone, etc.) but Pi is a nice combination of performance, cost, and available software