ECE 471 – Embedded Systems
Lecture 3

Vince Weaver

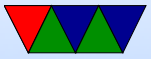
http://web.eece.maine.edu/~vweaver

vincent.weaver@maine.edu

3 September 2021

Announcements

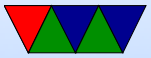
- HW#1 was posted, due Friday
- Don't forget Monday is Labor Day



Computer System Tradeoffs

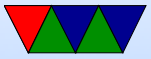
It's all about tradeoffs

- Power / Thermal
- Performance
- Cost
- Compatibility
- Time to Market
- Features
- Size/footprint

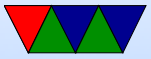


Challenges vs Regular Systems

- 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

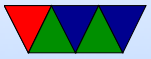


The ARM Architecture



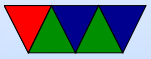
Brief ARM History

- **A**corn **RI**SC **M**achine. Acorn was a computer company in the UK in the 1980s
- Wanted a chip to succeed 6502. Decided to make one themselves. (Good idea, 65816 a pain and only 16-bit)
- 6502 was the chip in Commodore 64, Apple II, NES, Atari 2600
- Fun fact: 6502 design led by UMaine alum Chuck Peddle
- Bought by Softbank (Japan) in 2016
- Softbank possibly in talks to sell ARM to NVIDIA (2020)



RISC / CISC Discussion

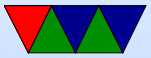
- Simple decode. Load/store. Fixed instruction width. 3-operand.
- MIPS is classic RISC
- x86 is classic CISC (with complex instructions)
  Though internally x86 executes uops, RISC
- ARM (predication, auto-increment, barrel shifter)
  Called RISC but has complex instructions



RISC / CISC Example

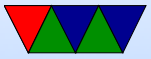
CISC	RISC

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



ARM Business Plan

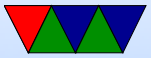
- IP Licensing company. Does not fab own chips. License to other companies
- Other companies take the design, put on SoC, attach whatever other logic blocks are needed
- Relatively small company compared to Intel which not only deigns the chip, but fabs, etc.
- Can buy full core (Cortex-AX) or just rights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



AMBA Bus Protocol

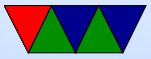
Advanced Microcontroller Bus Architecture

- ARM System Bus (ASB), ARM Peripheral Bus (APB)
- ARM High Performance Bus (AHB)
- Common bus, various companies can provide logic blocks for it, can swap in and out ARM cores as needed.



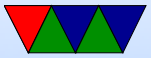
ARM Architecture vs Family

- ARMv1 : ARM1
- ARMv2 : ARM2, ARM3 (26-bit, status in PC register)
- ARMv3 : ARM6, ARM7
- ARMv4 : StrongARM, ARM7TDMI, ARM9TDMI
- ARMv5 : ARM7EJ, ARM9E, ARM10E, XScale
- ARMv6 : ARM11, ARM Cortex-M0 (Raspberry Pi A/B)
- ARMv7 : Cortex A8, A9, A15, A7, Cortex-M3 (Pi2)
- ARMv8 : Cortex A50, A53, A57 (64-bit), Pi3, Pi4
- ARMv9 : announced



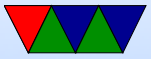
Various abbreviations in Model Names

- 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)



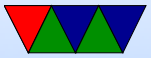
- “D” Debug
- “I” ICE (In-circuit Emulator)
- “EE” ThumbExecutionEnvironment, Just-in-time
- NEON – SIMD

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

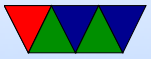


STM32L476-Discovery

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



Raspberry Pi



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 (BeagbleBone, etc.) but Pi is a nice combination of performance, cost, and available software