

ECE 471 – Embedded Systems

Lecture 2

Vince Weaver

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

`vincent.weaver@maine.edu`

6 September 2019

Announcements

- Reminder: The class notes are posted to the website.
- HW#1 will be assigned, watch your e-mail



Continuation from Last Time

Type			Speed	RAM	Disk	GPU
Intel	Xeon	64-bit	3GHz	8GB	1TB	Nvidia
ARM	A53	64-bit(?)	1GHz	1GB	8GB	VC4
ARM	M0	32-bit	32MHz	16kB	128kB	none
MOS	6502	8-bit	1MHz	64kB	140kB	none



Common Low-End Embedded Architectures

- Somewhat dated list, from EE Times 2003. Multiple answers so doesn't necessarily sum up to 100%
- 8-bit processors
 - Microchip PIC – 43%
 - AVR, etc. 8051 – 55%
 - Motorola 68xx – 36%
 - Zilog Z80 – 15%
- 16-bit processors



- 8086/80186/80286 - 41%
- 68HC12 - 21%



What Processors Commonly Used?

As reported by IDC at the SMART Technology conference in San Francisco for 2011

- ARM 71%
- MIPS 11%
- Other 9% (Linux supports 20+ architectures)
- x86 8% (at least Intel's desperately trying)
- Power 2%
- RISC-V?



We'll Mostly Use ARM in this Class

- Widely used
- You'll see it if you move to industry
- Other classes in ECE using it

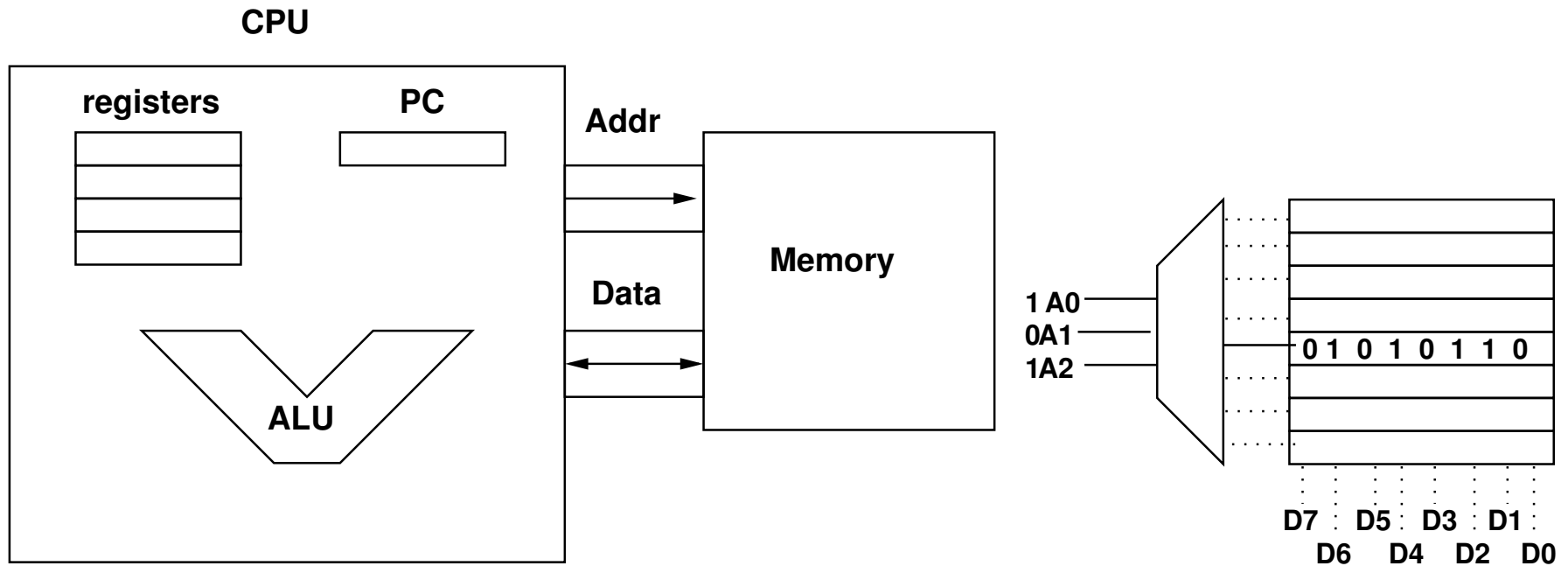


Microprocessors

- First one considered to be 4004 by Intel (for use in calculator)
- First to include all of a CPU on one chip. Before that there were processors, but often were made out of many discrete chips (sometimes entire boards full of logic)



8-bit vs 16-bit vs 32-bit



- What makes a processor 8-bit vs 16-bit vs 32-bit?
 - The size of the registers?
 - The size of the address bus?



- The size of the data bus?
- The size of the ALU (integer math unit)?
- The size of the PC (program counter)?
- On modern systems it typically is the integer registers, as well as the maximum size of a memory pointer (which typically is the same as the integer register size)
- On many systems though it is not as clear cut.
- A “pure” 8-bit system would have 8-bit registers (0-255), 8-bit ALU, and an 8-bit data bus. However an 8-bit address bus (only 256 bytes of RAM) is too limiting so most 8-bit processors (6502, z80, 8080, etc) had 16-



bit address busses, 16-bit PCs, and often 16-bit register capability

- Most 16-bit processors were equally complex. The 8086 had 16-bit registers and 16-bit data bus, but a 20-bit address bus with complex addressing. To complicated things, the 8088 was 8086 compatible but had only an 8-bit data bus (to save cost, with the side effect of making memory accesses take twice as long)
- Most 32-bit processors have 32-bit registers and 32-bits of address space, but that limits to 4GB so some have extensions (x86 and ARM) allowing 36-bits of address



space. And data bus has been made complex by caches and are often quite large. And often there are larger registers on chip (64-bit or 80-bit floating point, 128-bit SSE, 256-bit AVX)

- Most 64-bit processors have 64-bit registers, but their address bus is often limited (to 36 - 40 bits, sometimes 48-bits, this is complicated by virtual memory)
- Do machines have to be a power-of-two in bitness? No, not necessarily. 36-bit machines were once quite popular.



System-on-a-Chip / Microcontroller

- Moore's law allows lots of transistors
- Discrete Chips: CPU, GPU, Northbridge, Southbridge, (and older days, FPU, MMU, etc)
- System-on-a-Chip (SoC): All parts of computer on-chip CPU, DSP, memory, timers, USB, voltage regulators, memory controllers
- System-in-Package (SiP): various chips in one package



Extra Features on SoCs

- Parallel and Serial I/O
- A/D, D/A converters
- GPIO pins
- i2c, CAN, SPI, 1-wire, USB busses
- FPGA?
- Low-power
- Sound, DSP
- Video, GPU, Video Codecs
- Timers, PWM



ASIC, FPGA, Micro-controller

- ASIC – Application Specific Integrated Circuit
direct wiring of state machines / logic on silicon die
- FPGA – reprogrammable low-level logic
- Microcontroller – can do what above do, but in software
- Why use ASIC: could be faster, but what if mistake?
Why use FPGA: could be faster, more expensive
Why use microcontroller: Cost. Time to market. Bug-fixes (easier to fix in software)



Example Embedded Platforms

- 8051 microcontroller from Intel, 8-bit
 - 8 8-bit registers (one special 16-bit), 8-bit ALU
 - 8-bit data bus, 16-bit address bus (64kB)
 - bank-selectable register set
 - interrupts, timers
 - 128 Bytes on-chip RAM
 - 4k on-chip ROM
 - special boolean engine



- 68HC12 microcontroller
- ARM Cortex-M0 in the STM32L. See later.
- Other common embedded systems you might have heard of: PIC – generally 8 or 16bit but complicated (instructions can be 12 or 14)
ARDUINO – various AVR 8/16/32-bit micro controllers
AVR



Tradeoffs

It's all about tradeoffs

- Power
- Performance
- Cost
- Compatibility
- Time to Market
- Features



Challenges vs Regular Systems

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

