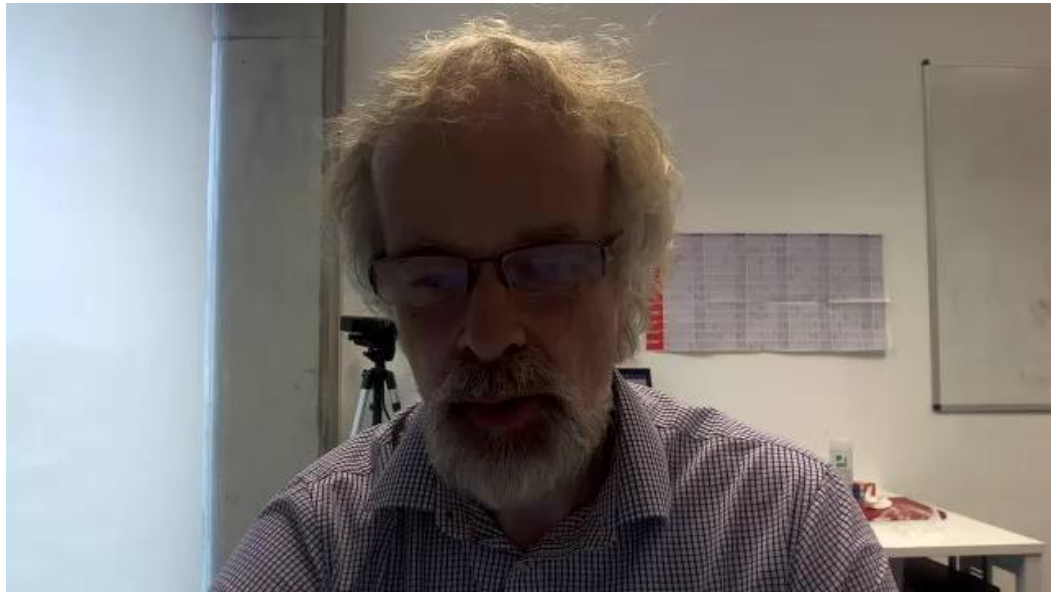
“Modern Architectures”

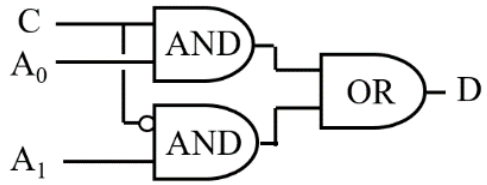
Dr. Stephen Brown
Dr. Charles Markham

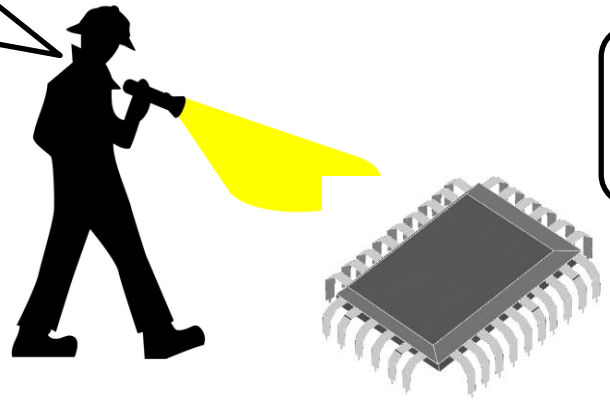
Objective: To understand how a modern computer operates and use this knowledge to be a better programmer and understand the implications of architecture for small-scale and large-scale computing



“Modern Architectures”

Last look at hardware

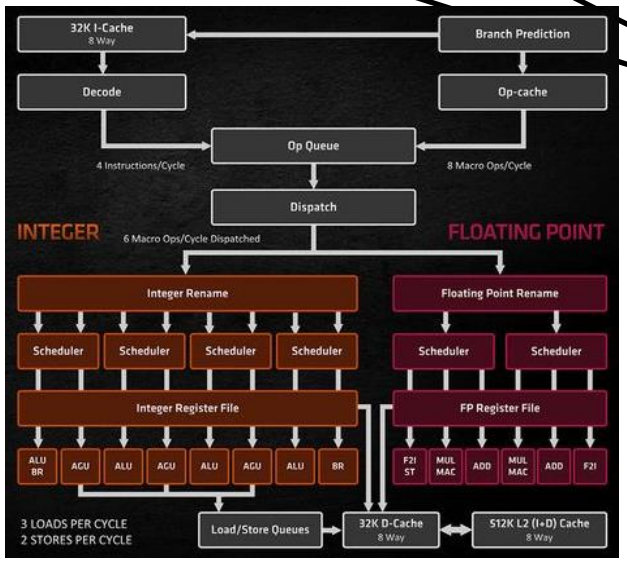




Closer look at code

```
back:   mov bx,OFFSET msg1
        mov dx,[bx]   ;dl=letters

        cmp dl,'$'
        jz  done

        mov ah,02h
        int 021h

        inc bx
        jmp back

done:   nop
```

Examine Modern CPUs



Understand Implications

Topics

1. Assembly Language [cm]
Traditional Machine Cycle,
Call conventions
TSC, MMX, SSE, AVX, AES, RDRAND,
Atomic operations, locks, mutex
Encryption

2. Advanced memory & storage architectures [sb]
Storage, Memory, Cache, prefetching

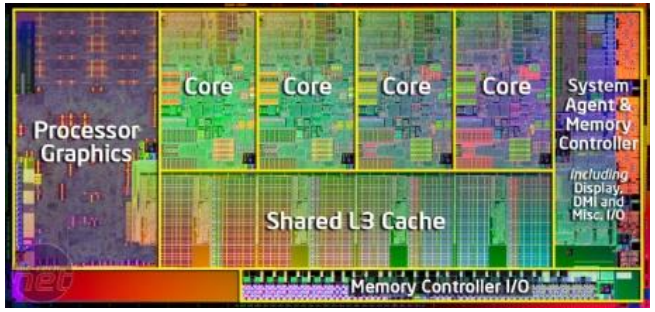
3. Advanced processing architectures [sb]
Modern machine cycle, Multi-die architectures

4. Interfacing [cm]
Arduino emulator

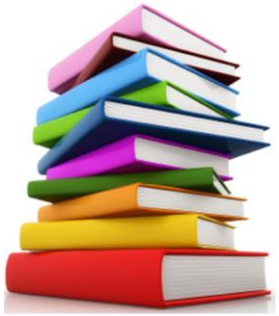
5: Special processing hardware [cm]
Threads, affinity, race conditions, self-mod code
FPGA, GPU, AIPU

6. Modern system architectures [sb]
Using IoT as an example:
sensor nodes to data centers

```
.code
public pass_xmm
pass_xmm:
movdqa     xmm0, [rcx+ 0*16]
movdqa     xmm1, [rcx+ 1*16]
movdqa     xmm2, [rcx+ 2*16]
paddq xmm0, xmm1
movdqa [rdx+ 0*16], xmm0;
movdqa [rdx+ 1*16], xmm1;
movdqa [rdx+ 2*16], xmm2;
ret ;
end
```



Assessment



Model:

bookwork – concepts and knowledge
mathematics – analysis and problem solving
programming – application and synthesis

Exam:

Traditional exam, focus on problem solving rather than book work
8 questions, do 6
Worth 75% of module grade

Continuous assessment:

Quiz most weeks – in total worth 25% of module grade
Some quizzes may require calculations or programming to answer the questions

* This is indicative and subject to change

Books & online resources

Intel white papers (technical publications) x86, x64 Assembly Language

The Art of Assembly Language – useful reference – Hybrid code HLA

Computer Architecture – A Quantitative Approach [Hennessy & Patterson]

Software

Visual Studio 2019 - Community edition (requires MS account)

Intel SDE Software Development Emulator

Nvidia CUDA SDK

Arduino SDK

AMD System Monitor

* This list will grow

IoT - Internet of Things

- IoT covers all aspects of computer architecture from smallest to largest

Data Collection

Storage & Processing

Display & Use

Why do we need performance?

Ever bigger data sets

Semantic gap is increasing

AI and Machine Learning

Video and Augmented reality

Why do we need low power?

Environment

Battery life

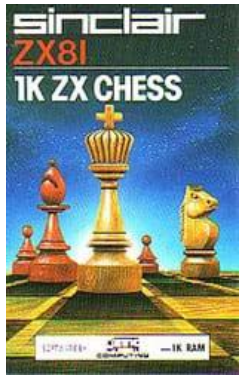
Thermal load and cooling

IoT Ubiquitous computing

Cost

Why learn about advanced architectures?

Chess in 1K



1982 by David Horne in Z80 Assembly with a code size of 672 bytes

ESP 32 - €6

Very popular with the IoT community

Processors:
CPU: Xtensa dual-core 240 MHz, ultra low power, 520K SRAM

Wireless connectivity:
Wi-Fi: 802.11 b/g/n, WPA2 also Bluetooth

Peripheral interfaces:
ADC, DACs, I²C, CAN, PWM, SD/SDIO

To write faster code.

To write efficient code (less code to do the same thing).

To understand the digital electronic computer from the device right through to the code.

To make informed design decisions: from nodes to data centers

It complements other modules such as programming, compilers, O/S, algorithms & data structures.

Understand the pace of development.