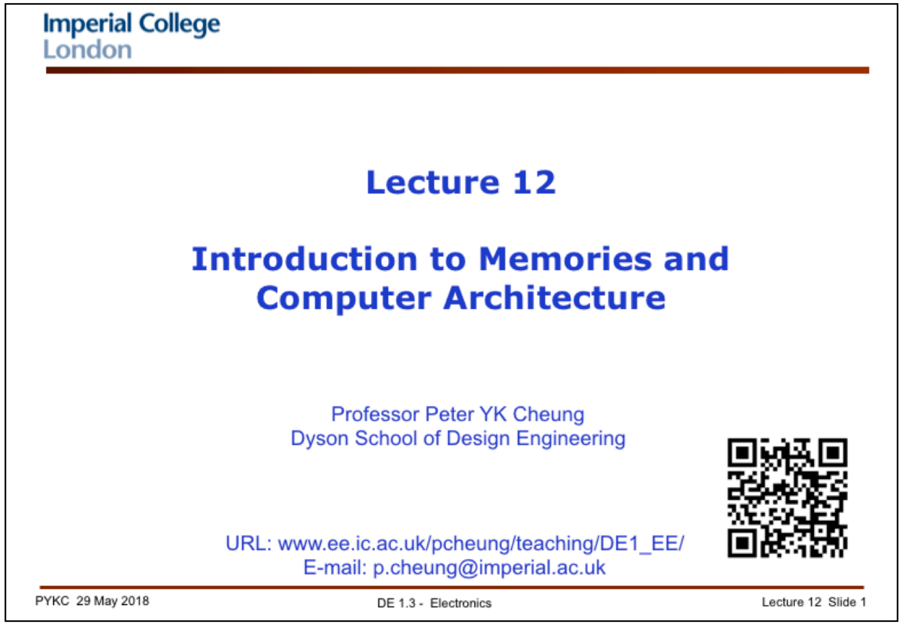


Imperial College London

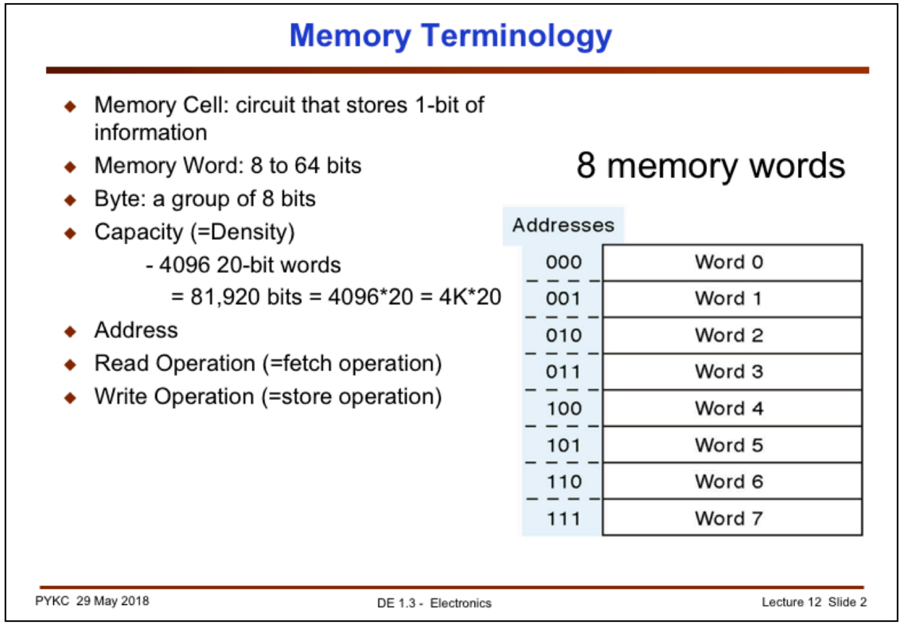
Lecture 12

Introduction to Memories and Computer Architecture

Professor Peter YK Cheung
Dyson School of Design Engineering

URL: www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/
E-mail: p.cheung@imperial.ac.uk

In this lecture, we will look at how storage (or memory) works with processor in a computer system. This is in preparation for the next lecture, in which we will examine how a microprocessor actually works inside.



Memory Terminology

- Memory Cell: circuit that stores 1-bit of information
- Memory Word: 8 to 64 bits
- Byte: a group of 8 bits
- Capacity (=Density)
  - 4096 20-bit words
    = 81,920 bits = 4096*20 = 4K*20
- Address
- Read Operation (=fetch operation)
- Write Operation (=store operation)

8 memory words

Addresses	
000	Word 0
001	Word 1
010	Word 2
011	Word 3
100	Word 4
101	Word 5
110	Word 6
111	Word 7

Let us first examine some common terminology related to memory.

A memory location always has a unique identifier called the address. Each memory unit is a cell which stores 1 bit of information. Therefore a D-FF is also a 1-bit memory.

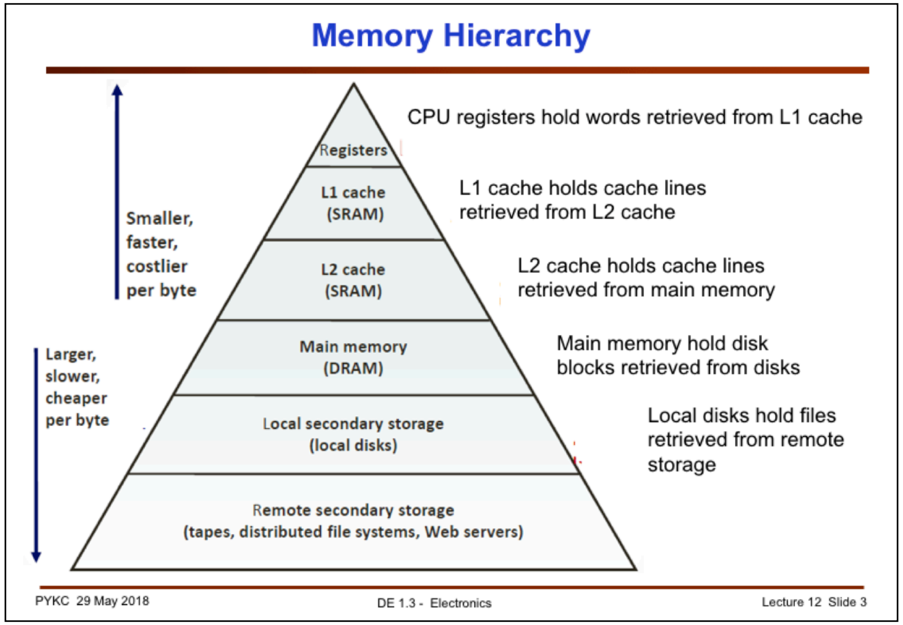
A memory word is a collection of bits to form a multi-bit unit that carries information. A word can be defined to be 8-bit, 16-bit, 32-bit or 64-bit. Other word sizes are possible, but not common.

An 8-bit word is known as a byte, which is the most common. For example, when you were sending or receiving the ASCII character '#' in Lab 1 or Lab 4, you were using a byte with the value 0x23 (0x … is the way to indicate the number following the prefix is a hexadecimal).

Memory capacity (i.e. How much memory is in a memory chip?) is measured in kb, Mb, Gb, Tb when measured on a chip. However, these memory bits could be organised in different way. For example, a 8Mb chip could be organised as 1M of 8-bit words, 2M of 4-bit words, or 8M of 1-bit storage.

The number of unique words in a memory module determines how many bits of address information is required to uniquely specify each location uniquely.

When using memory, we could be reading or fetching information from memory. This is a READ operation. We could also be storing information to the memory. This is a WRITE operation.



Memory Hierarchy

Smaller, faster, costlier per byte

Larger, slower, cheaper per byte

Registers

L1 cache (SRAM)

L2 cache (SRAM)

Main memory (DRAM)

Local secondary storage (local disks)

Remote secondary storage (tapes, distributed file systems, Web servers)

CPU registers hold words retrieved from L1 cache

L1 cache holds cache lines retrieved from L2 cache

L2 cache holds cache lines retrieved from main memory

Main memory hold disk blocks retrieved from disks

Local disks hold files retrieved from remote storage

In any computer systems, storage of information is always organiised in a hierarchy as shown here as a triangle. This has many layers representing different type of storage.

At the bottom is remote storage such as storage in the cloud (e.g. dropbox or iCloud), or central file server offered by an organisation for its employees. This is the largest and cheapest, and normally the slowest form of storage. This storage could be very large.

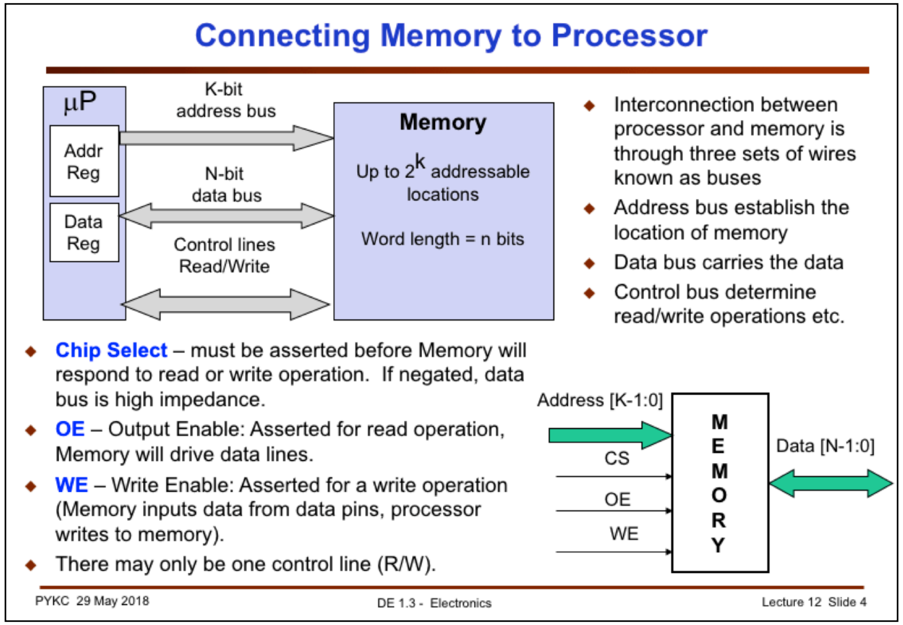
The next level is local disk storage, which is faster than the remote storage, but slower than the next layer up. Most disks are now 128 GB or more. (We use GB for giga byte and Gb for giga bit.)

The next level is the main memory or DRAM in the computer. Modern computer would come with 2GB or more of main memory. These normally come on small PCBs and are swappable. In that way, one could "upgrade" the memory, meaning that you can add more to the system.

The next two levels are SRAMs on the processor chip itself. They are L2 (level 2) and L1 (level 1) cache memory. Cache memory is usually not large, but has much faster access than all the other types of memory. Information are generally fetched in "cache lines" which is multiple bytes.

Finally the fastest memory are the registers inside the Central Processing Unit (CPU).

Information moves up and down this hierarchy in these order, going through each layer.



Connecting Memory to Processor

μP

Addr Reg

Data Reg

K-bit address bus

N-bit data bus

Control lines Read/Write

Memory

Up to 2^k addressable locations

Word length = n bits

- Interconnection between processor and memory is through three sets of wires known as buses
- Address bus establish the location of memory
- Data bus carries the data
- Control bus determine read/write operations etc.

- **Chip Select** – must be asserted before Memory will respond to read or write operation. If negated, data bus is high impedance.
- **OE** – Output Enable: Asserted for read operation, Memory will drive data lines.
- **WE** – Write Enable: Asserted for a write operation (Memory inputs data from data pins, processor writes to memory).
- There may only be one control line (R/W).

Address [K-1:0]

CS

OE

WE

MEMORY

Data [N-1:0]

Memory chips (not cloud storage or disks) are connected to the microprocessor through three bundles of signal connections. These are called "buses".

The address bus carries the location information or address of the memory. A k-bit address would allow 2^k locations to be accessed and uniquely specified. The address bus is usually driven by the processor chip.

The data bus carries the information read from or written to memory. It can be n-bit win width, where n is usually 8, 16, 32 or 64. for example, your College-issued computer is a 64-bit machine and the internal memory is organised with n=64. Everything you fetch a word from memory, each word is 64-bit.

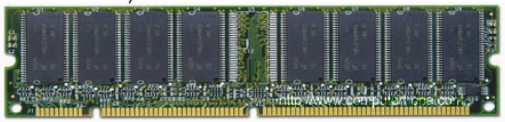
Finally the control bus provides all the control signal such as the signal to indicate whether you are doing a read or a write operation to memory.

Main memory characteristics

- Most devices are 8-bits wide (Byte-addressable); some are 16-bits, others 1 bit wide.
- Chip organisation examples: 1k x 8 (capacity = 8kb), 1G x 16 (16Gb)

Characteristics

- Access Times (read, write, erase)
  - The time from a valid address being placed on the address bus until valid data appears on the data bus.
  - Faster is Better (varies from minutes to a few ns)
- Volatility
  - Ability to Retain Data After Power is Removed
- Power Consumption
  - Less is Better (mW to nW typical)
- Density
  - Larger is Better (bits/sq. transistors/bit)
- Cost
  - Less is Better



This picture shows a memory module you may find inside your notebook computer. There are 8 chips, probably each providing 32-bit wide data to the edge connector at the bottom of the PCB, Assuming that the board has total capacity of 1G x 32 bit, or 4GB in total, then each memory chip could be organised as 1GB x 4 bit.

In memory, the important characteristics are:

Access time – how fast can one perform a read or write operation

Volatility – volatile memory are those that looses information when power is removed. Non-volatile memory are those that keep information even when no power is applied.

Power consumption – the lower the better

Density – how many bits can be stored?

Cost – this is normally directly related to capacity and hence silicon area of the chips.

Types of Semiconductor Memories

- **ROM** – Read Only Memory - a type of memory that cannot be written, can only be read. Contents determined a manufacture time.
- **PROM** – Programmable ROM – a type of memory whose contents can be programmed by the user
  - OTP – One Time Programmable, a PROM is OTP if contents can be programmed only once.
- **RAM** – Random Access Memory
- Memory that can be both read and written during normal operation.
- Contents are volatile, i.e.will be lost on power off.
- Two types of RAM:

  Static RAM
  - Fast access time (used for off-processor cache)
  - Does not have to be refreshed

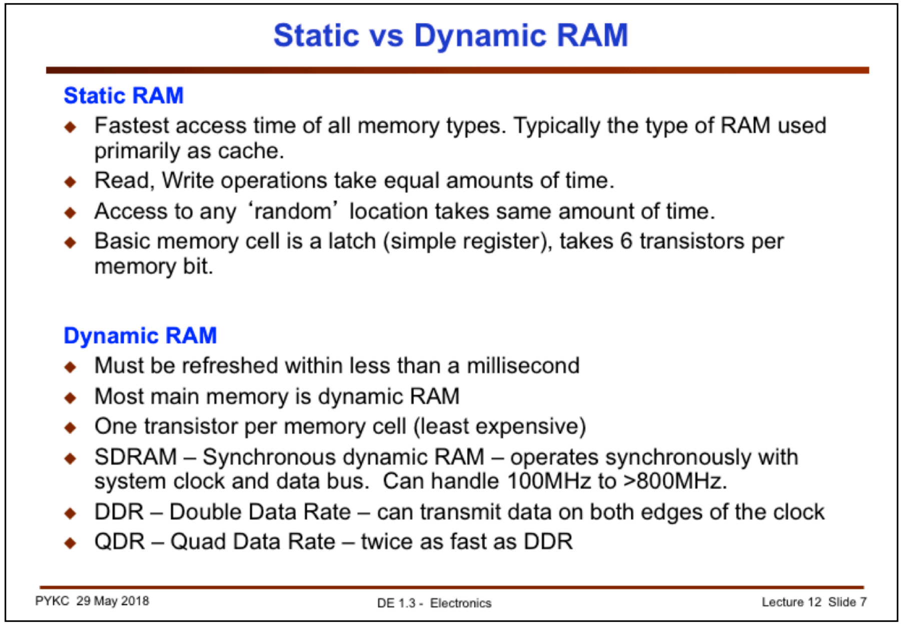
  Dynamic RAM
  - Slower access time
  - Must be refreshed
  - Much more dense

There are different types of memory chips. ROM is read-only. They are now very rare, because flash memory can act as both read-only and read/write memory.

PROM are those that you can program (usually once) in the filed.

RAM is the most common memory used. These are both read and write. They are called random access memory because you can go to any location individually and perform a read/write operation, only to that location.

There are two types of RAMs: static and dynamic.



Static vs Dynamic RAM

Static RAM

- Fastest access time of all memory types. Typically the type of RAM used primarily as cache.
- Read, Write operations take equal amounts of time.
- Access to any 'random' location takes same amount of time.
- Basic memory cell is a latch (simple register), takes 6 transistors per memory bit.

Dynamic RAM

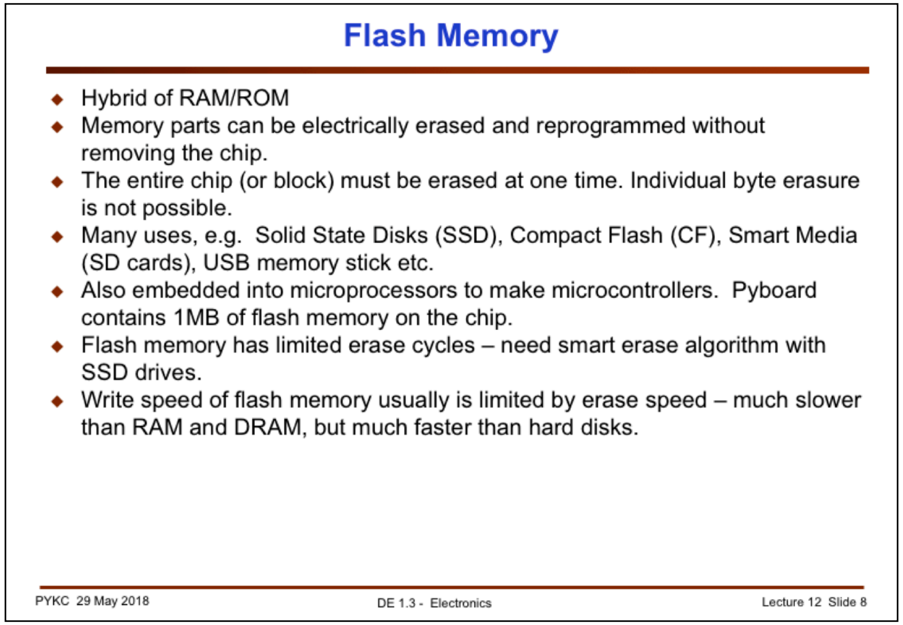
- Must be refreshed within less than a millisecond
- Most main memory is dynamic RAM
- One transistor per memory cell (least expensive)
- SDRAM – Synchronous dynamic RAM – operates synchronously with system clock and data bus. Can handle 100MHz to >800MHz.
- DDR – Double Data Rate – can transmit data on both edges of the clock
- QDR – Quad Data Rate – twice as fast as DDR

Static RAM are those that are generally used for cache memory. They are fast but expensive. The memory density is lower than the dynamic RAM. Static memory are large and expensive because each memory bit is made of a simple latch (a special type of flip-flop) which requires 6 transistors.

Dynamic RAMs are generally used for main memory or for the video RAM on a GPU. These are slower than static RAM in access and they are dynamic meaning that it uses a capacitor as the storage mechanism. By storing a small charge on the capacitor, it remembers whether the memory bit is '1' (say with charge) or '0' (no charge). DRAMs are high density because each memory only requires one transistor. Therefore they are also cheaper than static RAMs.

There are also many types of DRAMs. Most common are the SDRAM or synchronous DRAM. These use a clock signal to synchronise all read/write operations. Then there are those that are known as DDR memory. What it means is that on each clock cycle, there are TWO memory operations – one on the rising each and one on the falling each of the clock signal.

There even a QDR memory which transfers four times per clock cycles!



Flash Memory

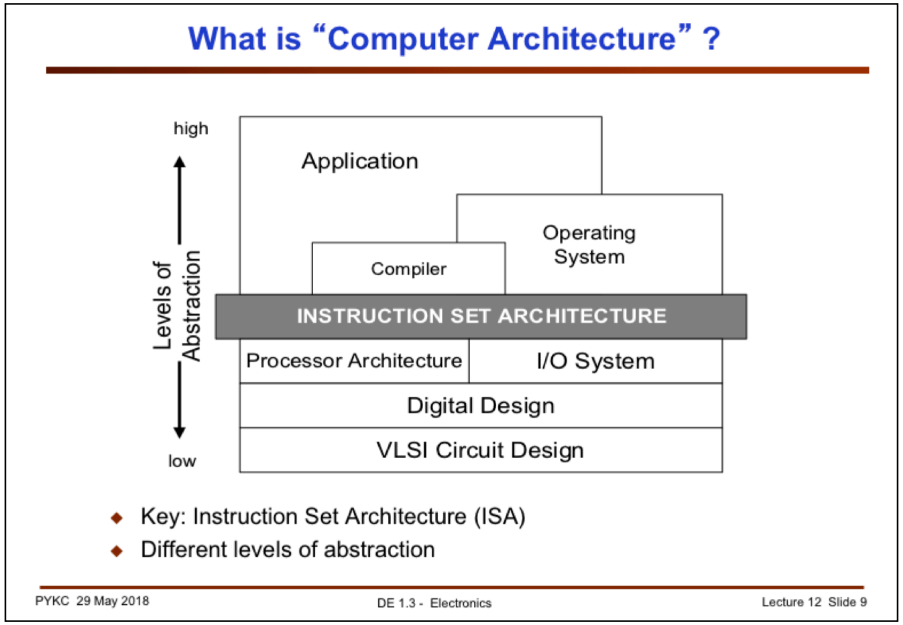
- Hybrid of RAM/ROM
- Memory parts can be electrically erased and reprogrammed without removing the chip.
- The entire chip (or block) must be erased at one time. Individual byte erasure is not possible.
- Many uses, e.g. Solid State Disks (SSD), Compact Flash (CF), Smart Media (SD cards), USB memory stick etc.
- Also embedded into microprocessors to make microcontrollers. Pyboard contains 1MB of flash memory on the chip.
- Flash memory has limited erase cycles – need smart erase algorithm with SSD drives.
- Write speed of flash memory usually is limited by erase speed – much slower than RAM and DRAM, but much faster than hard disks.

Finally, there are the flash memory. These are mostly behaving as ROM. However, a user can electrically erase its contents a block at a time and write back the block.

Flash memory therefore is generally reasonably fast (much faster than hard disk) in reading, and slower in writing due to this block erase requirement.

Furthermore, flash memory have limited write cycles in its life-time. Therefore when flash memory are used in solid state disks (SSD drives), there needs to have an algorithm that prevent the user using the same area of the disk (i.e. the same memory locations).

Flash memory are used in many applications including USB sticks and SD memory cards for portable equipment.



What is “Computer Architecture” ?

high

Application

Operating System

Compiler

INSTRUCTION SET ARCHITECTURE

Processor Architecture

I/O System

Digital Design

VLSI Circuit Design

Levels of Abstraction

low

- Key: Instruction Set Architecture (ISA)
- Different levels of abstraction

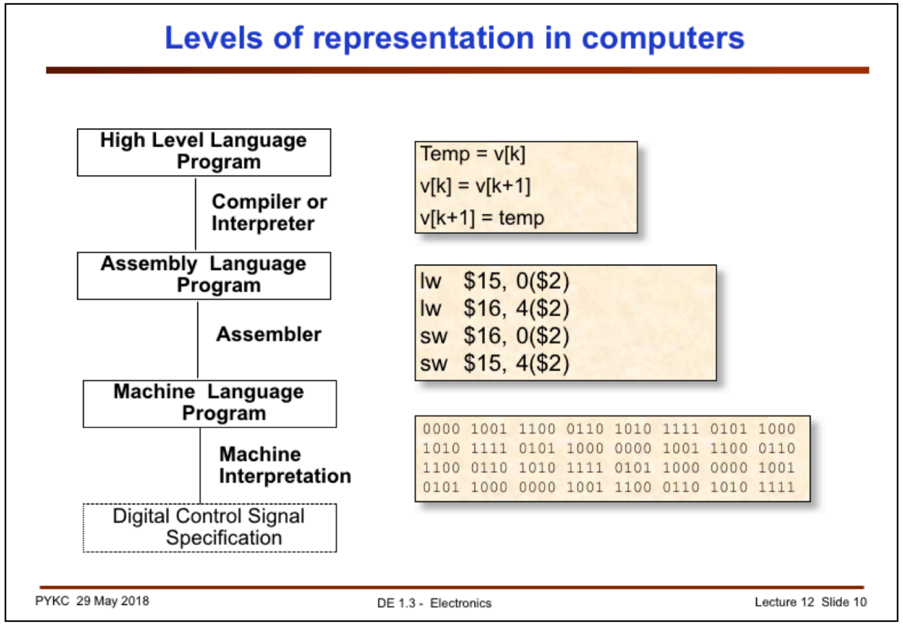
Now let us examine what makes up a computer system. Shown here is a hierarchy diagram from transistor level hardware (VLSI circuits) to the application programs. What is important is to appreciate the **levels of abstraction** used to present the system for different specialty of engineering.

DE students generally only need to know that application level. That’s why we are teaching you Python programming and you use the Pyboard. Your focus is applications.

Underneath that are generally two things: the operating system (such as OSX or Windows) and compiler. For Pyboard, the program on the board (known as “firmware”, partly because it is stored inside the flash memory and it is firm – or fixed) performs both functions of the operating system and the compiler.

Underneath that is the instruction set architecture. This is the instructions used by the central processing unit (CPU). For the Pyboard, we use the ARM instruction set.

Beneath that are what electronic engineers are trained to do – hardware design. Although most electronic engineers can also do all the other levels pretty well!



Levels of representation in computers

High Level Language Program

Compiler or Interpreter

Assembly Language Program

Assembler

Machine Language Program

Machine Interpretation

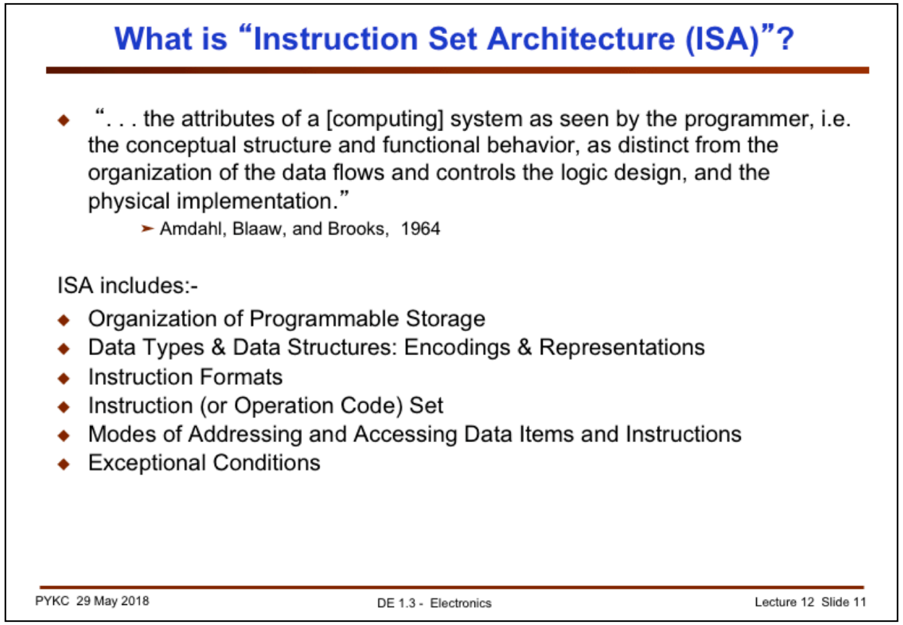
Digital Control Signal Specification

```
Temp = v[k]
v[k] = v[k+1]
v[k+1] = temp
```

```
lw   $15, 0($2)
lw   $16, 4($2)
sw   $16, 0($2)
sw   $15, 4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

Here is diagram showing how a high level language is used to specify instructions to the CPU. It goes through different steps. First there is the compiler which produces lower level language (usually called assembly language programs). This is then translated to machine instructions in binary form. These instructions are decoded by the CPU to become digital signals that control gates and flip-flops on chip itself.



What is "Instruction Set Architecture (ISA)"?

- "... the attributes of a [computing] system as seen by the programmer, i.e. the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls the logic design, and the physical implementation."
  - ➢ Amdahl, Blaaw, and Brooks, 1964

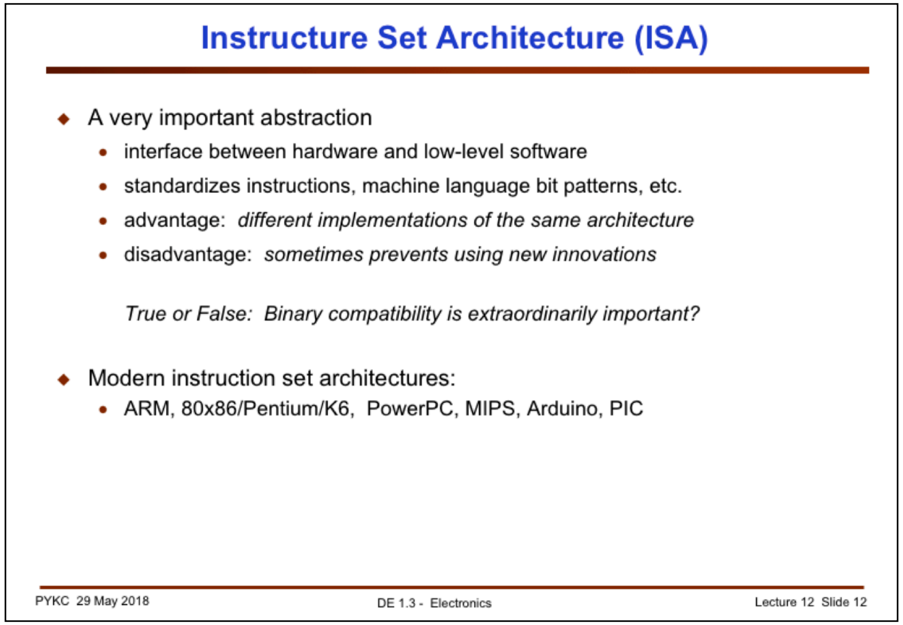
ISA includes:-

- Organization of Programmable Storage
- Data Types & Data Structures: Encodings & Representations
- Instruction Formats
- Instruction (or Operation Code) Set
- Modes of Addressing and Accessing Data Items and Instructions
- Exceptional Conditions

Central to any computer system is the Instruction Set Architecture (ISA). This defines what the CPU understands as instructions. We are using the ARM instruction set on the Pyboard but this is hidden from you because you interact with the CPU via Python only.

However, it is important to understand what is going on inside the processor – at least to some level of abstraction.

In the next lecture, I will be taking you through a very simple ISA (much simpler than the ARM ISA).



Instructure Set Architecture (ISA)

- A very important abstraction
  - interface between hardware and low-level software
  - standardizes instructions, machine language bit patterns, etc.
  - advantage: *different implementations of the same architecture*
  - disadvantage: *sometimes prevents using new innovations*

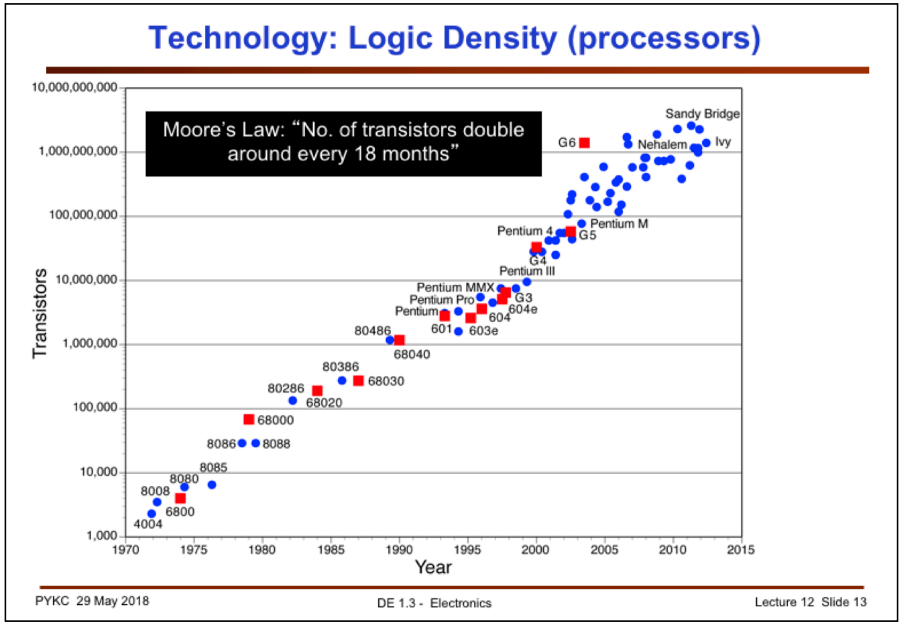
  True or False: Binary compatibility is extraordinarily important?

- Modern instruction set architectures:
  - ARM, 80x86/Pentium/K6, PowerPC, MIPS, Arduino, PIC

Once upon a time, there were MANY processor architectures. Now, there are only a few. For computers, Intel's 86x architecture dominates.

For mobile applications, the ARM architecture dominates. There are many more ARM processor being made and sold each day than Intel processors!

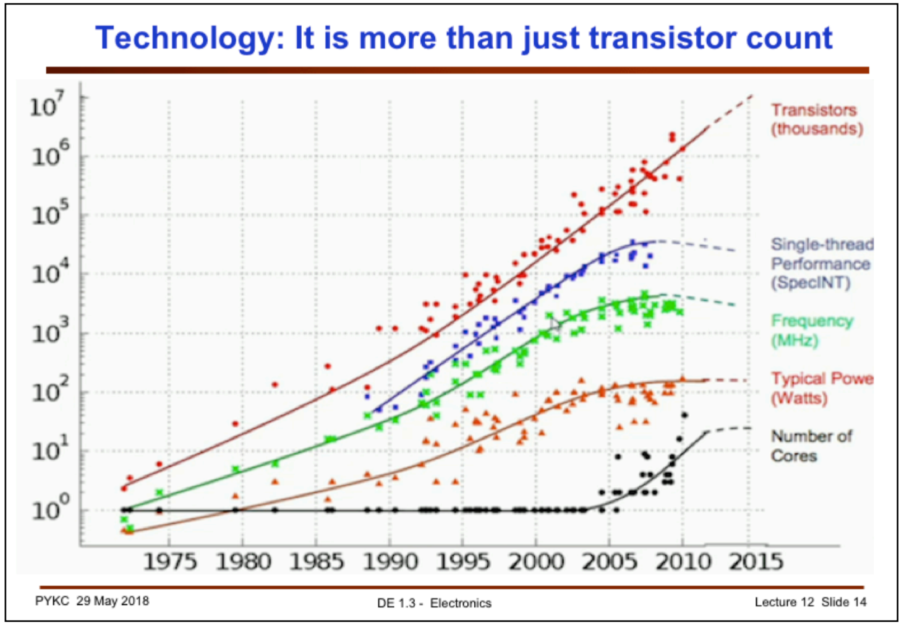
For hobbyist, the Arduino architecture dominates.



Technology: Logic Density (processors)

Moore's Law: "No. of transistors double around every 18 months"

PYKC 29 May 2018 DE 1.3 - Electronics Lecture 12 Slide 13



Technology: It is more than just transistor count

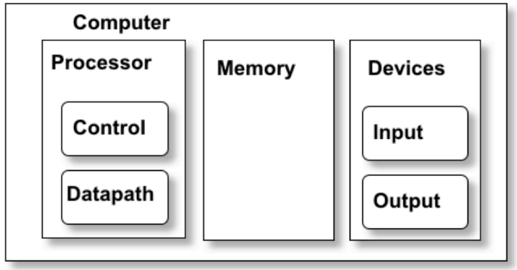
PYKC 29 May 2018 DE 1.3 - Electronics Lecture 12 Slide 14

Gordon Moore from Intel famously observed that transistor density on silicon doubles every year to year and a half. That became the Moore's law. It drove the semiconductor industry for the past four decades, and it became a self-fulfilling prophesy.

Here is a lot of how transistor density (i.e. no of transistors on a processor chip) over the years. Note that the y-axis is in log scale.

We often heard people say that this is the end of Moore's law. I think this view is incomplete. If fact if you look at the plot here, transistor density is still going up as Moore has predicted. However what becomes much more significant is NOT transistor density, but other factors including power consumption (by each chip) and the performance you get from processor. These other factors are NOT increasing. They are leveling out!

Internal Organisation


Computer

Processor

Control

Datapath

Memory

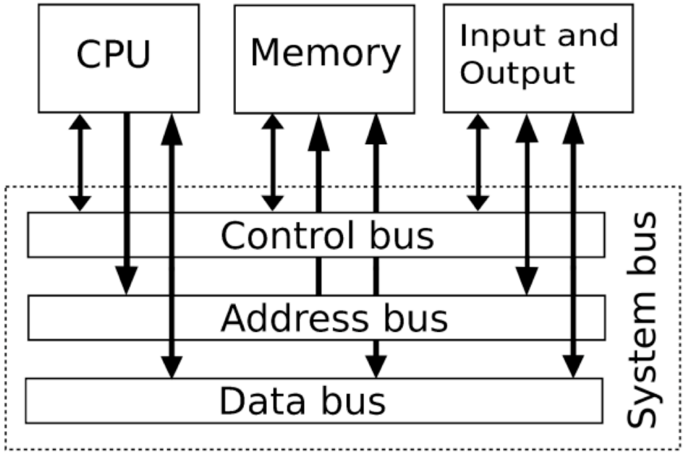
Devices

Input

Output

- Major components of Typical Computer System

A Typical Computer System with I/O


CPU

Memory

Input and Output

Control bus

Address bus

Data bus

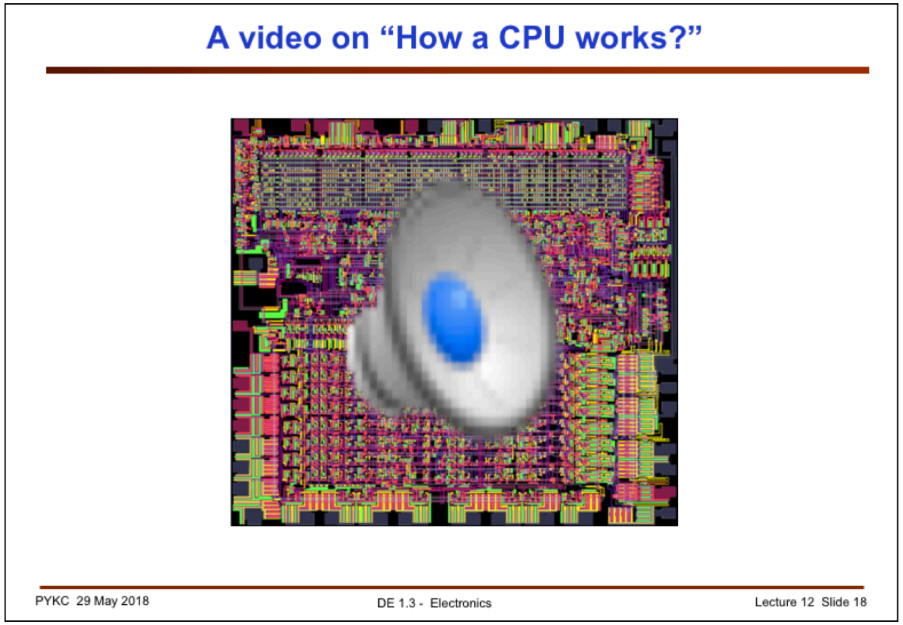
System bus

Here is a typical internal view of a computer. It has the processor (CPU), memory system (including disk storage) and input/output devices.

All these are interconnected via system buses: the address bus, the data bus and the control bus.

Summary

- All computers consist of five components
  - Processor: (1) datapath and (2) control
  - (3) Memory
  - (4) Input devices and (5) Output devices
- Not all "memory" are created equally
  - Cache: fast (expensive) memory are placed closer to the processor
  - Main memory: less expensive memory--we can have more
- Input and output (I/O) devices has the least regular organization
  - Wide range of speed: graphics vs. keyboard
  - Wide range of requirements: speed, standard, cost ... etc.
  - Least amount of research (so far)



A video on "How a CPU works?"

This is an excellent video to introduce you to the CPU. We will watch this together at the end of the lecture.

https://www.youtube.com/watch?v=cNN_tTXABUA&t=81s