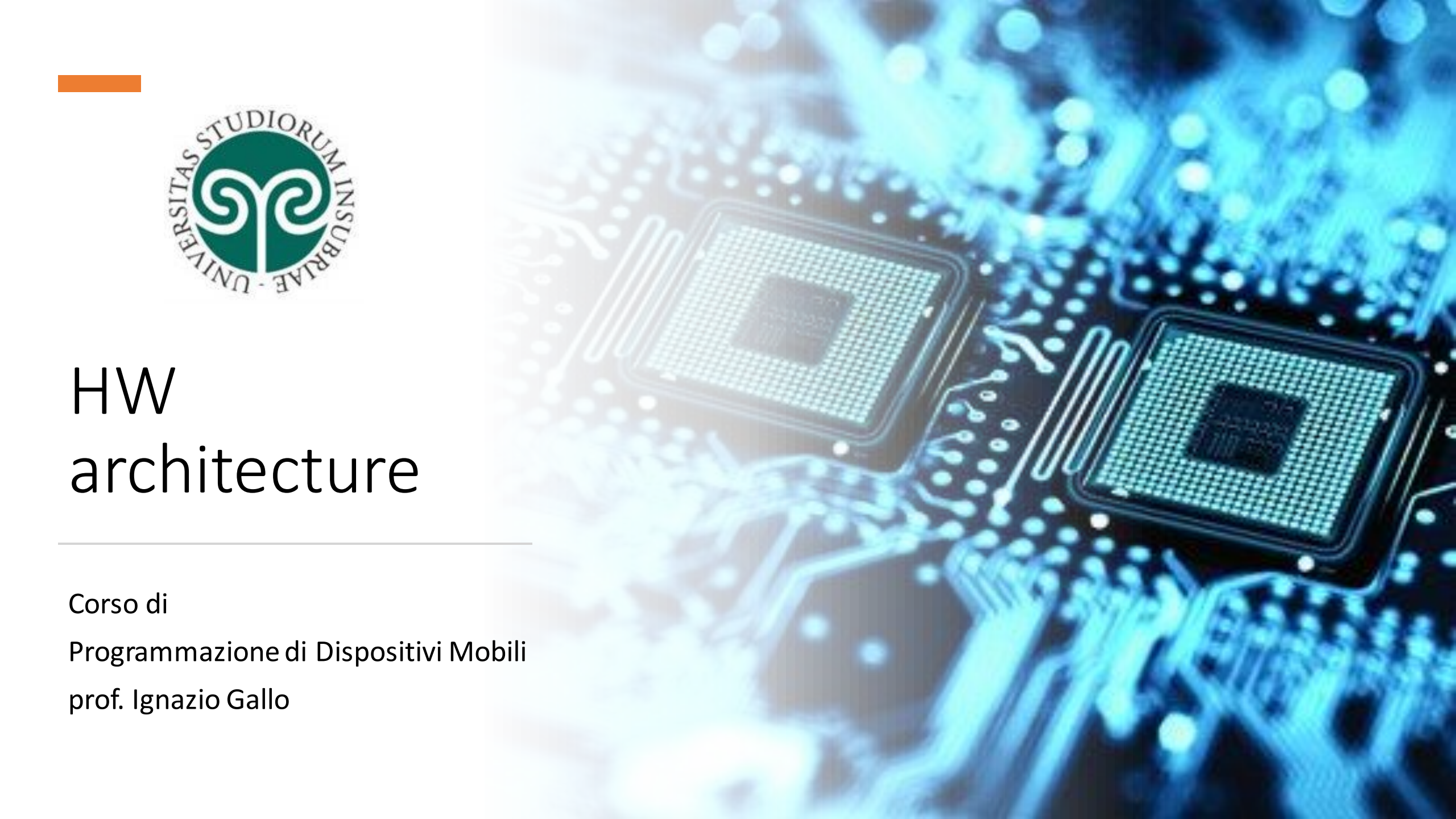


HW architecture

Corso di

Programmazione di Dispositivi Mobili

prof. Ignazio Gallo

Evoluzione storica

All'inizio della storia dell'informatica, per facilitare il lavoro dei programmatori e anche perchè la RAM costava tanto, i progettisti dei processori decisero di inserire **istruzioni anche molto complesse** in modo da simulare le funzioni ad alto livello dei linguaggi di programmazione direttamente nei processori.

```
00000000
00000001
00000003
00000007
00000008
0000000C
0000000F
00000011
00000014
00000016
00000019
0000001B
0000001D
0000001F
00000022
```

```
push    ebp
mov     ebp, esp
movzx   ecx, [ebp+arg_0]
pop     ebp
movzx   dx, cl
lea     eax, [edx+edx]
add     eax, edx
shl     eax, 2
add     eax, edx
shr     eax, 8
sub     cl, al
shr     cl, 1
add     al, cl
shr     al, 5
movzx   eax, al
retn
```


```
#include<stdio.h>

int main()
{
   printf("Hello, World!\n");
   return 0;
}
```

hello_world.c

Compiler

```
11000000011111111110000001
11110001101010100011000011
00110001000100110001111110
00000011111111110000001111
10001101010100011000110011
10001000100110001111100000
00011111111110000001111100
01101010100011000110011000
01000100110001111100000000
11111111100000011111000011
```

hello_world.o

Gli studi dimostrarono che molte istruzioni assemply esotiche venivano utilizzate molto raramente e a volte erano più lente del codice scritto con le istruzioni generiche.

L'obiettivo di ridurre le istruzioni portò all'ideazione del nome *reduced instruction set computing*

HW: ARM architecture

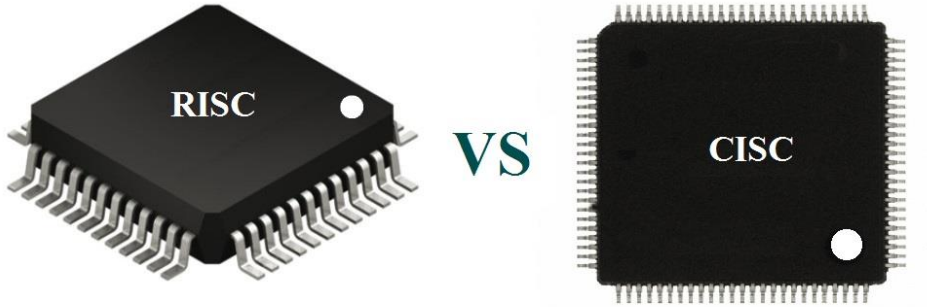
- **ARM** = Advanced RISC Machines (opposed to x86)
- **RISC** = Reduced Instruction Set Computer
- ARM is a family of instruction set architectures based on a RISC architecture developed by British company ARM Holding .
- Partners:
  - Apple, IBM, Texas Instruments, Samsung, ST-Ericsson …
- ARM develops the instruction set and architecture for ARM-based products but does not manufacture products.
- Products are manufactured by licensees
- ARM is one of the most popular architectures used in embedded Linux systems


intel
AMD
X86
ARM
NVIDIA

HW: ARM architecture

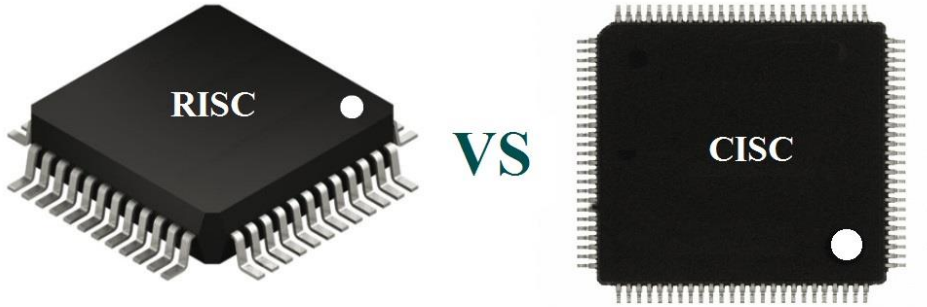
- **RISC** = Reduced Instruction Set Computing
- Processors require significantly fewer transistors than **CISC** (Complex Instruction Set Computing) processors.
  - This reduces costs, heat and power use.
    - Desirable traits for light, portable, battery-powered devices
  - A simpler design facilitates more efficient multi-core CPUs and higher core counts at lower cost.
- **ARM** designs CPU cores
  - instruction sets, caches memory management unit (MMU) and sells the design to licensees.

HW: CISC vs RISC



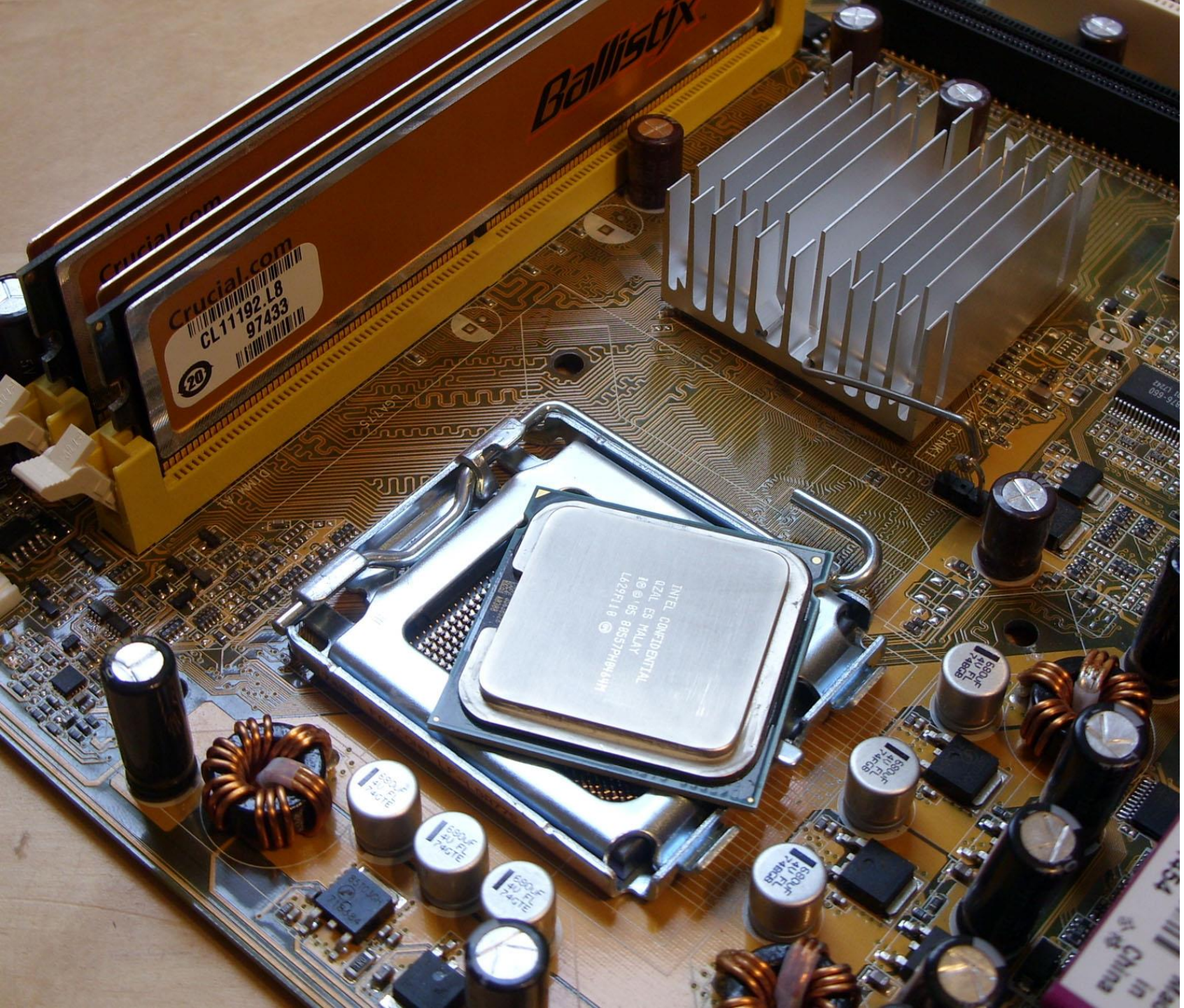
- Let we take an example of multiplying two numbers.
  - `A = A * B; // this is C statement`
- The **CISC** Approach:
  - complete a task in as few lines of assembly as possible
    - `MULT A,B  # this is assembly statement`
- The **RISC** Approach:
  - only use simple instructions that can be executed within one clock cycle
    - `LOAD R1, A  # this is assembly statement`
    - `LOAD R2, B  # this is assembly statement`
    - `PROD A, B  # this is assembly statement`
    - `STORE R3, A  # this is assembly statement`

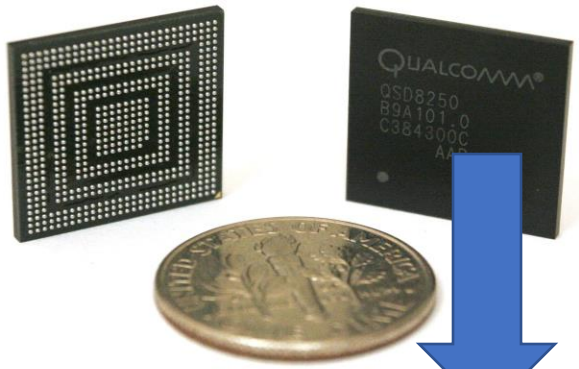
HW: CISC vs RISC

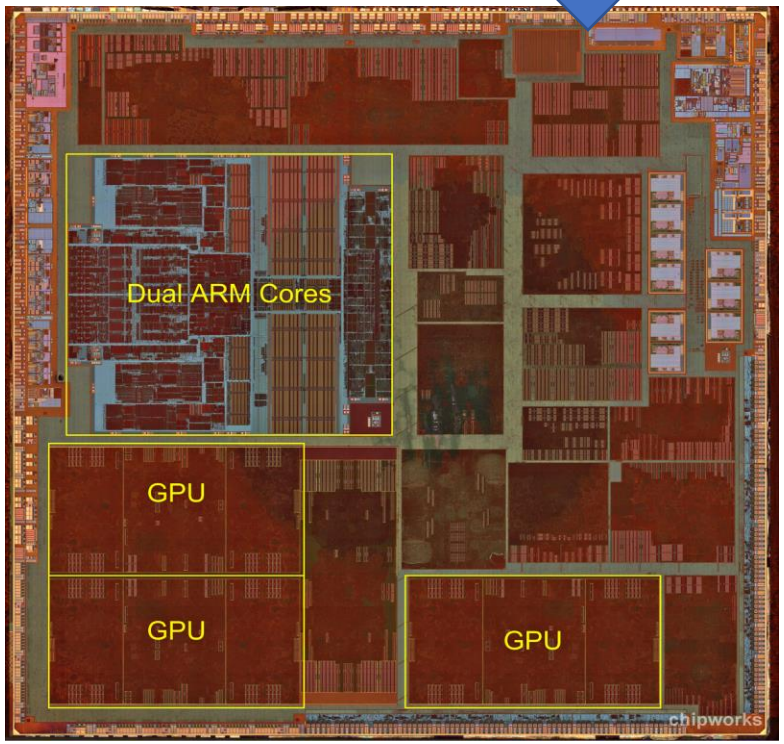


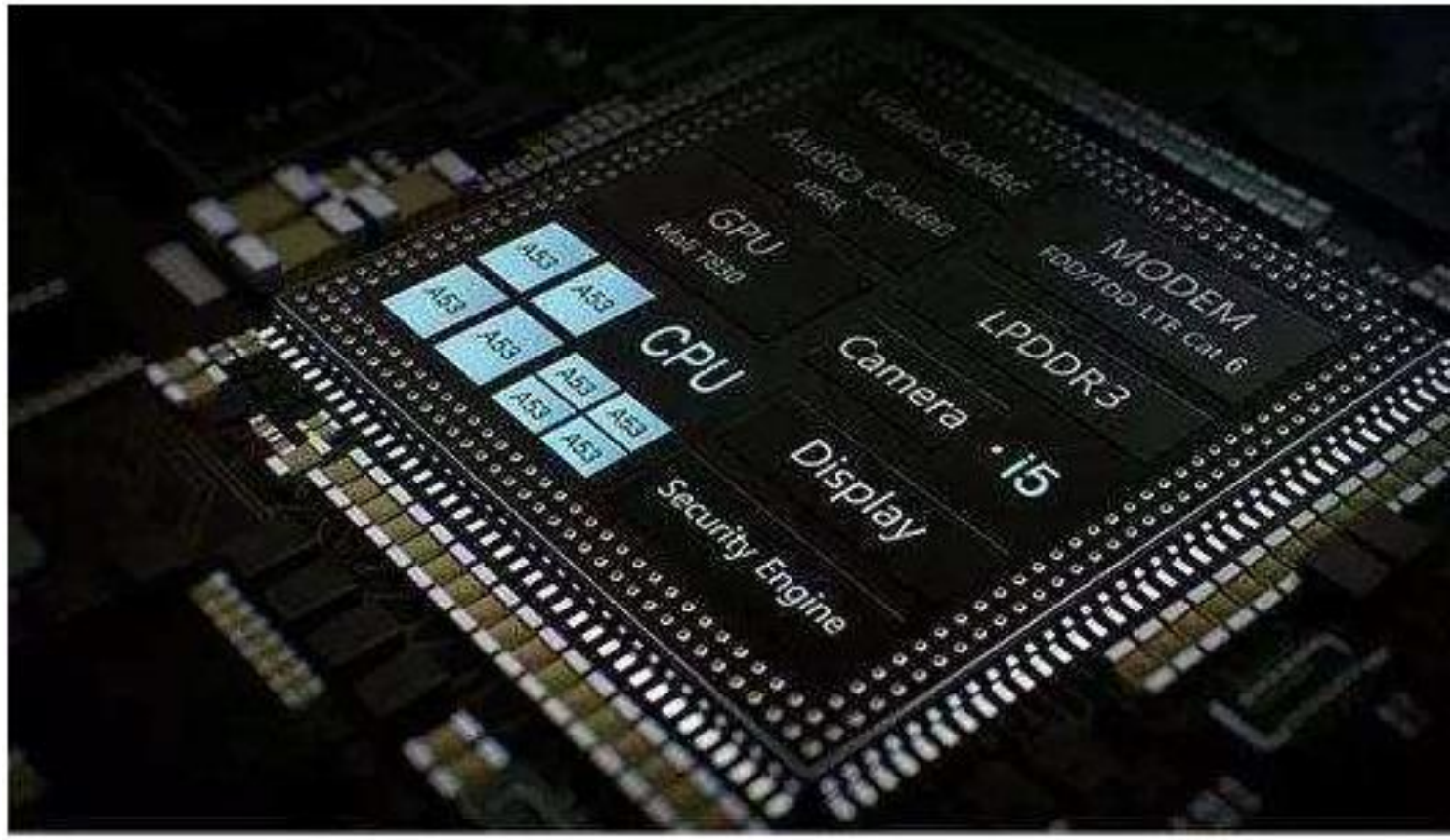
- Let we take an example of
- multiplying two numbers.
  - `A = A * B; // this is C statement`
- **CISC** Advantages:
  - Compiler must do very little work to translate a high-level language statement into assembly
  - Length of the code is relatively short
  - Very little RAM is required to store instructions
  - The emphasis is put on building complex instructions directly into the hardware.
- **RISC** Advantages:
  - Each instruction requires only one clock cycle to execute, the entire program will execute in approximately the same amount of time as the multi-cycle “MULT” command.
  - These RISC “reduced instructions” require less transistors of hardware space than the complex instructions, leaving more room for general purpose registers. Because all of the instructions execute in a uniform amount of time (i.e. one clock)

Laptop vs Mobile





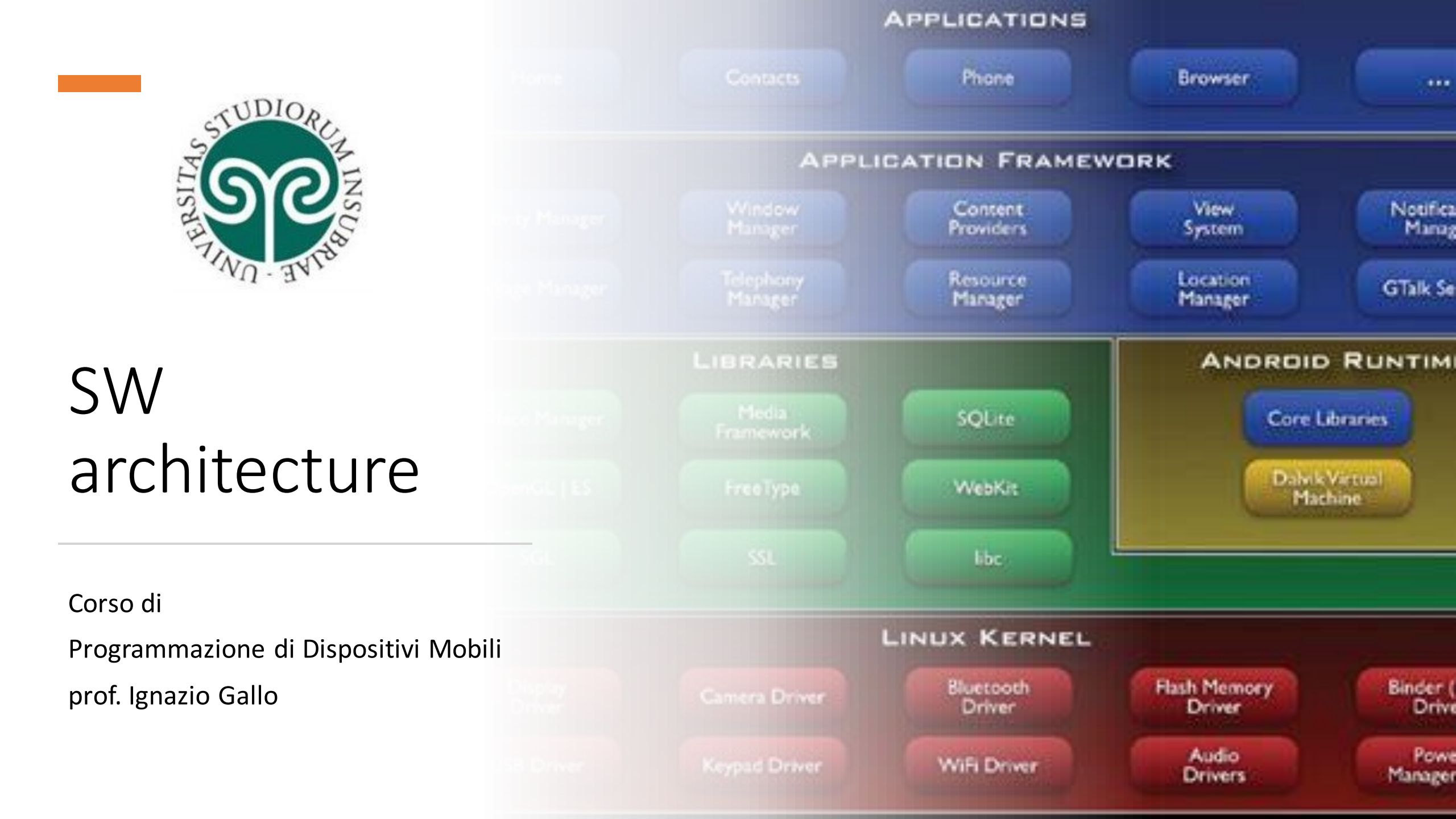




SoC: System on Chip

- La maggior parte dei componenti che possiamo vedere in una scheda madre di un laptop o desktop, sono integrate all'interno di **un singolo chip** nello smartphone.
- **ARM progetta il SoC** e vende le license per poterli costruire
- Il sistema operativo di uno smartphone è ottimizzato per poter sfuttare questa tecnologia

Video di approfondimento:
https://youtu.be/NKfW8ijmRQ4



SW architecture

Corso di

Programmazione di Dispositivi Mobili

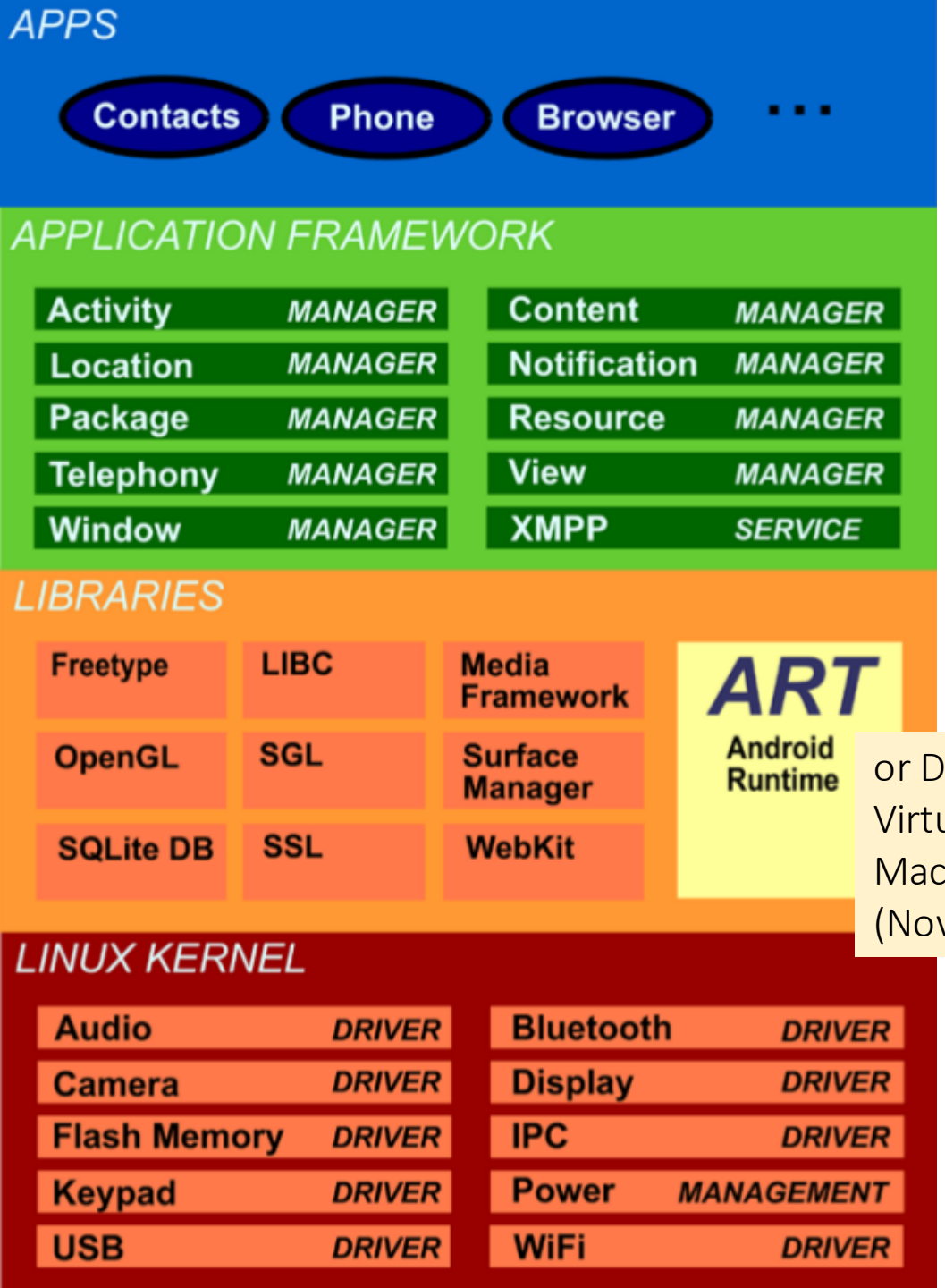
prof. Ignazio Gallo

Platform for mobile development

- Android is an ecosystem made up of:
- A free, open-source Linux-based OS for embedded devices
- An open-source development platform for creating applications (SDK)
- Devices
  (mobile phones, tablets, ….) that run the Android operating system and the applications created for it

Linux versions

Android Version	API Level	Linux Version in AOSP	Header Vers
1.5 Cupcake	3	(2.6.27)	
1.6 Donut	4	(2.6.29)	2.6.18
2.0/1 Eclair	5-7	(2.6.29)	2.6.18
2.2.x Froyo	8	(2.6.32)	2.6.18
2.3.x Gingerbread	9, 10	(2.6.35)	2.6.18
3.x.x Honeycomb	11-13	(2.6.36)	2.6.18
4.0.x Ice Cream San	14, 15	(3.0.1)	2.6.18
4.1.x Jelly Bean	16	(3.0.31)	2.6.18
4.2.x Jelly Bean	17	(3.4.0)	2.6.18
4.3 Jelly Bean	18	(3.4.39)	2.6.18
4.4 Kit Kat	19, 20	(3.10)	2.6.18
5.x Lollipop	21, 22	(3.16.1)	3.14.0
6.0 Marshmallow	23	(3.18.10)	3.18.10
7.0 Nougat	24	3.18.48 4.4.0	4.4.1
7.1 Nougat	25	?	4.4.1
8.0 Oreo	26	3.18.72 4.4.83 4.9.44	4.10.0
8.1 Oreo	27	3.18.70 4.4.88 4.9.56	4.10.0
9.0 Pie	28	4.4.146 4.9.118 4.14.61	4.15.0
10.0 Q	29	4.9.191 4.14.142 4.19.71	5.0.3



The Android Operating System

or Dalvik Virtual Machine (Novel)

Key points

- HW architecture
- CISC vs RISC
- SW architecture