# Arquitectura de Computadoras

Risc-V: arquitectura RISC de código abierto

## Early development of computers

1st Generation

#### Vaccum Tubes

Querian estudiar las trayectorias balisticas

¿Como se programaba? Memoria de programa “a mano”. La forma en que se conectaban los tubos, junto con los voltajes aplicados a sus electrodos, determinaba el comportamiento del circuito. Manejamos físicamente los bits

#### von Neumann

Almacenar los datos de programa en la memoria. Memoria de rayos catódicos. Siguen diendo tubos de vacio

2nd Generation

Incorporacion del transistor.

Con cada nuevo lanzamiento la frecuencia aumenta pero no la memoria

### 3rd Generation

10k – 50 k transistors

Aparecen los circuitos integrados

Aqui unifican el S.O. que es compatible para este y futuros lanzamientos. Antes cada procesador lanzado por IBM tenia su propio software

#### Moore’s Law

Hace 20 años estamos en el limita, los procesadores no crecen, porque el tamaño disminuye y el calor aumenta (porque aumentamos mas transistores), empobreciendo la disipacion de calor, mucha densidad (creo)

### Last Generation

1er microprocesador (Intel 4004), todos los compontentes integrados

La 486 ya puede correr windows

Pentium 4 es un punto de infleion en el Desarrollo de microprocesadores. El avance de frecuencia es lineal hasta este punto donde se estanca en 1.8GHz. Para avanzar se paraleliza, etc

## Computer Architecture

Son 3 cosas:

1. ISA (Instruction Set Architecture)
   1. Classes: Si es load-store signifcia que la ALU solo puede hacer operaciones con registros (RISC y RISC-V), al ser un set corto de instrucciones, lo importante es que cada instruccion tiene el mismo largo. Si es register-memory puede realizar operaciones con la memoria y con los registros.
2. HW components
3. Organization (micro-architecture)

## Trends in Performance

Mas crecimiento del ancho de banda de las memorias con respecto a la velocidad de procesador

Pico de conusmo con el Pentium 4, ya que si aumento la frecuencia aumento el conusmo

## Trends in Technology

## Trends in Power and Energy

### Energia dinamica

energia de un pulso, para pasar de 0 a 1 o de 1 a 0.

d = C · V2 (0🡪1🡪0, 0🡪1🡪0)

d = ½ · C · V2 (0🡪1, 1🡪0)

potencia = d \* freq

Deduccion de que cada vez que aumentamos la frecuencia la potencia aumenta al cubo (Freq3)

### Techniques to improve efficiency

Do nothing well 🡪 bajar la frecuencia a las unidades no utilizadas

Dynamic 🡪 depende la demanda va adecuando la frecuencia (sleep)

Overclocking 🡪 subir la frecuencia de procesado

### Static Power

Statica por el voltaje

## Measuring Performance

Benchmarks, se hacen de distintas formas, suele ser el tiempo que utiliza el procesador para realizar ciertas tareas

## Amdahl’s Law

Para mejorar la performance del Sistema se debe mejorar la peor parte del mismo.

Trata la parelizacion, dependiendo que porcion del programa es paralelizable. Si un programa corre en 10s en un procesador…

## Processor

Nos importa el tiempo de CPU, que esta dado por la cantidad de ciclos de un programa por la frecuencia de ese ciclo. Cada intstrucion toma x ciclos de reloj (CPI)

La performance depende de 3 caracteristicas:

* Clock rate
* CPI
* Cantidad de Instrucciones

El **retardo F04** es el retardo de propagación (t) de una señal con 4 compuertas NOR conectadas secuencialmente. Se usa como medida de velocidad . Este “t” es el tiempo que tarda una señal en entrar a un combinacional y salir por el mismo

Si pensamos nuestro procesador como un combinacional, la frecuencia máxima que puede tener seria f = 1/t

Para aumentar la frecuencia debemos aumentar las etapas combinacionales, poniendo latches (guarda ua señal) entre combinacionales, es decir, hacienda mas chicos los combinacionales, por ende reduciendo el t, por ende pudiendo aumentar la frecuencia. Se puede hacer hasta la latencia de los latches

Que una instruccion tarde X ciclos significa que tiene X etapas

Me interesa que el Sistema sea muy eficiente, tenga poco retardo en los sistemas embebidos