



Organización del Computador II

Microarquitectura - Estado del arte de la Tecnología

Alejandro Furfaro

Departamento de Computación - FCEyN - UBA

11 de abril de 2018

Temario

- 1 Tecnología de Integración
- 2 Arquitectura de Computadores
 - Instruction Set Architecture (ISA)
 - Organización y Hardware
- 3 Objetivos de estudiar organización y hardware

Escenario Actual

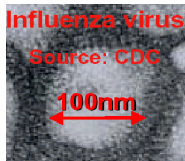


Figura: Vista microscópica de una célula del virus de la gripe

- En 2012 se lanzó la primer implementación con transistores MOS tri-gate en 22nm.
- En 2014 se comenzó a trabajar con tecnología de 14 nm
- En 2017 se implementó el proceso de 10 nm
- Y para 2019 y 2021 se prevé implementar respectivamente tecnologías de 7 y 5 nm.
- ¿Y luego?

Escenario Actual

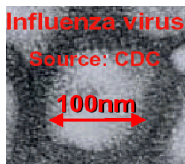


Figura: Vista microscópica de una célula del virus de la gripe

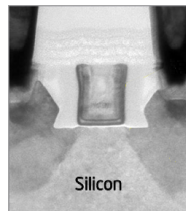


Figura: Vista en un Microscopio termoelectrónico de un CMOS de 32 nm high-K gate

- En 2012 se lanzó la primer implementación con transistores MOS tri-gate en 22nm.
- En 2014 se comenzó a trabajar con tecnología de 14 nm
- En 2017 se implementó el proceso de 10 nm
- Y para 2019 y 2021 se prevé implementar respectivamente tecnologías de 7 y 5 nm.
- ¿Y luego?

Escenario Actual

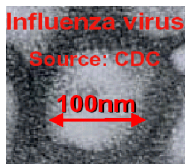


Figura: Vista microscópica de una célula del virus de la gripe

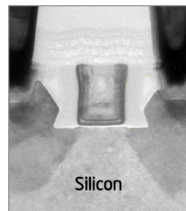


Figura: Vista en un Microscopio termoelectrónico de un CMOS de 32 nm high-K gate

- En 2012 se lanzó la primer implementación con transistores MOS tri-gate en 22nm.
- En 2014 se comenzó a trabajar con tecnología de 14 nm
- En 2017 se implementó el proceso de 10 nm
- Y para 2019 y 2021 se prevé implementar respectivamente tecnologías de 7 y 5 nm.
- ¿Y luego?

Escenario Actual

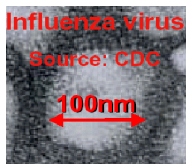


Figura: Vista microscópica de una célula del virus de la gripe

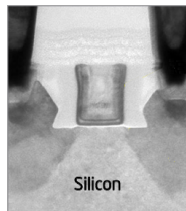


Figura: Vista en un Microscopio termoelectrónico de un CMOS de 32 nm high-K gate

- En 2012 se lanzó la primer implementación con transistores MOS tri-gate en 22nm.
- En 2014 se comenzó a trabajar con tecnología de 14 nm
- En 2017 se implementó el proceso de 10 nm
- Y para 2019 y 2021 se prevé implementar respectivamente tecnologías de 7 y 5 nm.
- ¿Y luego?

Escenario Actual

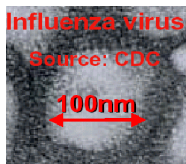


Figura: Vista microscópica de una célula del virus de la gripe

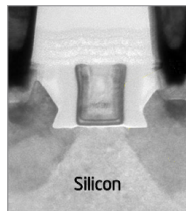


Figura: Vista en un Microscopio termoelectrónico de un CMOS de 32 nm high-K gate

- En 2012 se lanzó la primer implementación con transistores MOS tri-gate en 22nm.
- En 2014 se comenzó a trabajar con tecnología de 14 nm
- En 2017 se implementó el proceso de 10 nm
- Y para 2019 y 2021 se prevé implementar respectivamente tecnologías de 7 y 5 nm.
- ¿Y luego?

Escenario Actual

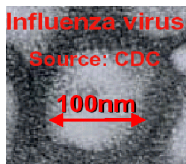


Figura: Vista microscópica de una célula del virus de la gripe

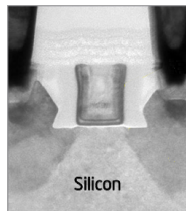


Figura: Vista en un Microscopio termoelectrónico de un CMOS de 32 nm high-K gate

- En 2012 se lanzó la primer implementación con transistores MOS tri-gate en 22nm.
- En 2014 se comenzó a trabajar con tecnología de 14 nm
- En 2017 se implementó el proceso de 10 nm
- Y para 2019 y 2021 se prevé implementar respectivamente tecnologías de 7 y 5 nm.

● ¿Y luego?

Escenario Actual

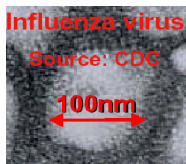


Figura: Vista microscópica de una célula del virus de la gripe

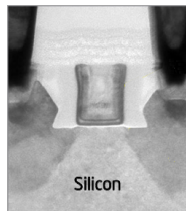


Figura: Vista en un Microscopio termoelectrónico de un CMOS de 32 nm high-K gate

- En 2012 se lanzó la primer implementación con transistores MOS tri-gate en 22nm.
- En 2014 se comenzó a trabajar con tecnología de 14 nm
- En 2017 se implementó el proceso de 10 nm
- Y para 2019 y 2021 se prevé implementar respectivamente tecnologías de 7 y 5 nm.
- ¿Y luego?

Ley de Moore



“The number of transistors incorporated in a chip will approximately double every 24 months.” – Gordon Moore, Intel co-founder.

©Cortesía Intel

Evolución Tecnológica

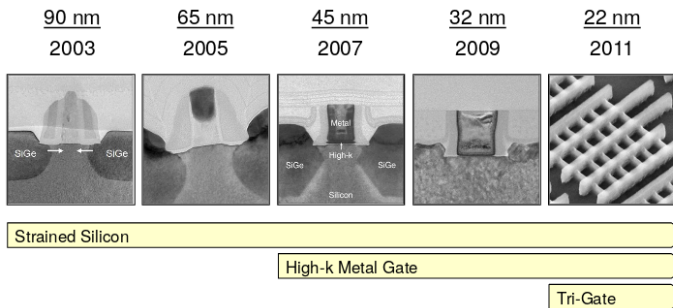


Figura: Evolución de las tecnologías de integración. ©Cortesía Intel

Los progresos en Scaling se logran en base a modificaciones permanentes en la estructura de los transistores y en los materiales que los componen.

Tendencias tecnológicas

La tarea de un diseñador es permanente e inevitablemente moldeada por el rumbo de las tecnologías

- 1 La densidad de transistores por unidad de superficie aumenta 35 % por año en promedio. (Otra forma de la ley de Moore).
- 2 El tamaño del die aumenta 10 a 20 % por año. Esto deriva en un crecimiento en la cantidad de transistores de entre 40 % y 55 % de un año a otro.

Tendencias tecnológicas

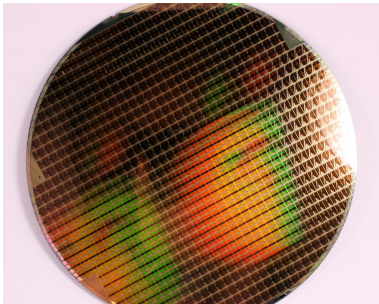
La tarea de un diseñador es permanente e inevitablemente moldeada por el rumbo de las tecnologías

- 1 La densidad de transistores por unidad de superficie aumenta 35 % por año en promedio. (Otra forma de la ley de Moore).
- 2 El tamaño del die aumenta 10 a 20 % por año. Esto deriva en un crecimiento en la cantidad de transistores de entre 40 % y 55 % de un año a otro.

Tendencias tecnológicas

La tarea de un diseñador es permanente e inevitablemente moldeada por el rumbo de las tecnologías

- 1 La densidad de transistores por unidad de superficie aumenta 35 % por año en promedio. (Otra forma de la ley de Moore).
- 2 El tamaño del die aumenta 10 a 20 % por año. Esto deriva en un crecimiento en la cantidad de transistores de entre 40 % y 55 % de un año a otro.



Tendencias tecnológicas

- 3 La velocidad de clock ya no crece. Parecería haber alcanzado un techo en los 3 GHZ aproximadamente.
- 4 La capacidad de almacenamiento de las memorias DRAM crece a razón de un 40 % por año.
- 5 Los discos rígidos aumentan su capacidad 25 a 30 % por año. Su costo por bit de almacenamiento se mantiene entre 50 y 100 veces por debajo del costo de un bit de memoria DRAM

Tendencias tecnológicas

- 3 La velocidad de clock ya no crece. Parecería haber alcanzado un techo en los 3 GHz aproximadamente.
- 4 La capacidad de almacenamiento de las memorias DRAM crece a razón de un 40 % por año.
- 5 Los discos rígidos aumentan su capacidad 25 a 30 % por año. Su costo por bit de almacenamiento se mantiene entre 50 y 100 veces por debajo del costo de un bit de memoria DRAM

Tendencias tecnológicas

- 3 La velocidad de clock ya no crece. Parecería haber alcanzado un techo en los 3 GHz aproximadamente.
- 4 La capacidad de almacenamiento de las memorias DRAM crece a razón de un 40 % por año.
- 5 Los discos rígidos aumentan su capacidad 25 a 30 % por año. Su costo por bit de almacenamiento se mantiene entre 50 y 100 veces por debajo del costo de un bit de memoria DRAM

Scaling

- El proceso de un circuito integrado está caracterizado por un solo parámetro: *tamaño*, que es el mínimo valor en la dimensión de un transistor en las dimensiones x o y .
- El tamaño de un transistor en 1971 era de 10 micrones.
- Actualmente es de 0,01 micrones: 1000 veces mas pequeño. . .
- Los transistores se cuentan por mm^2 de silicio, de modo que podemos esperar una función de incremento del tipo cuadrática.

Scaling

- El proceso de un circuito integrado está caracterizado por un solo parámetro: *tamaño*, que es el mínimo valor en la dimensión de un transistor en las dimensiones x o y .
- El tamaño de un transistor en 1971 era de 10 micrones.
- Actualmente es de 0,01 micrones: 1000 veces mas pequeño. . .
- Los transistores se cuentan por mm^2 de silicio, de modo que podemos esperar una función de incremento del tipo cuadrática.

Scaling

- El proceso de un circuito integrado está caracterizado por un solo parámetro: *tamaño*, que es el mínimo valor en la dimensión de un transistor en las dimensiones x o y .
- El tamaño de un transistor en 1971 era de 10 micrones.
- Actualmente es de 0,01 micrones: 1000 veces mas pequeño. . .
- Los transistores se cuentan por mm^2 de silicio, de modo que podemos esperar una función de incremento del tipo cuadrática.

Scaling

- El proceso de un circuito integrado está caracterizado por un solo parámetro: *tamaño*, que es el mínimo valor en la dimensión de un transistor en las dimensiones x o y .
- El tamaño de un transistor en 1971 era de 10 micrones.
- Actualmente es de 0,01 micrones: 1000 veces mas pequeño. . .
- Los transistores se cuentan por mm^2 de silicio, de modo que podemos esperar una función de incremento del tipo cuadrática.

Scaling

- Otro parámetro importante en un transistor es su rendimiento.
- Al disminuir el tamaño en sentido vertical un transistor requiere reducir su tensión de alimentación. De otro modo su rendimiento decae o puede dañarse.
- Como no es posible cambiar la tensión de operación cada vez que se reduce la escala, la mejora en el rendimiento con cada avance en scaling no es cuadrática, sino que tiende a ser lineal.

Scaling

- Otro parámetro importante en un transistor es su rendimiento.
- Al disminuir el tamaño en sentido vertical un transistor requiere reducir su tensión de alimentación. De otro modo su rendimiento decae o puede dañarse.
- Como no es posible cambiar la tensión de operación cada vez que se reduce la escala, la mejora en el rendimiento con cada avance en scaling no es cuadrática, sino que tiende a ser lineal.

Scaling

- Otro parámetro importante en un transistor es su rendimiento.
- Al disminuir el tamaño en sentido vertical un transistor requiere reducir su tensión de alimentación. De otro modo su rendimiento decae o puede dañarse.
- Como no es posible cambiar la tensión de operación cada vez que se reduce la escala, la mejora en el rendimiento con cada avance en scaling no es cuadrática, sino que tiende a ser lineal.

Scaling

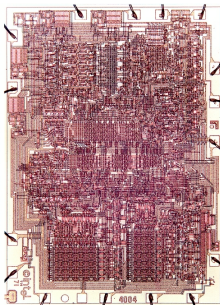


Figura: Procesador 4004, 2300 transistores 10 micrones. 1971
©Cortesía Intel

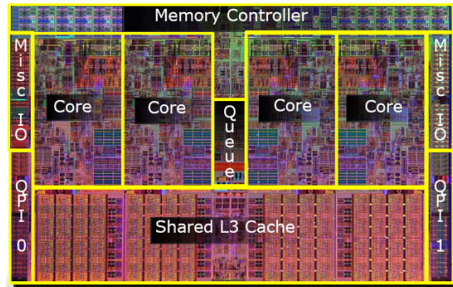


Figura: Procesador Core i7, 2.000.000.000 transistores 22 nm. 2012 ©Cortesía Intel

Scaling

- A medida que disminuye el parámetro *tamaño*, los transistores ganan linealmente en rendimiento.
- Los “alambres” son los caminos de señal que conectan los diferentes componentes.
- Conforman los buses internos.
- Estos caminos de señal, en las escalas actuales y a las frecuencias de trabajo actuales generan dos problemas:

Scaling

- A medida que disminuye el parámetro *tamaño*, los transistores ganan linealmente en rendimiento.
- Los “alambres” son los caminos de señal que conectan los diferentes componentes.
- Conforman los buses internos.
- Estos caminos de señal, en las escalas actuales y a las frecuencias de trabajo actuales generan dos problemas:
 - Delays.
 - Interferencia de señales.

Scaling

- A medida que disminuye el parámetro *tamaño*, los transistores ganan linealmente en rendimiento.
- Los “alambres” son los caminos de señal que conectan los diferentes componentes.
- Conforman los buses internos.
- Estos caminos de señal, en las escalas actuales y a las frecuencias de trabajo actuales generan dos problemas:
 - Delays.
 - Consumo de Energía.

Scaling

- A medida que disminuye el parámetro *tamaño*, los transistores ganan linealmente en rendimiento.
- Los “alambres” son los caminos de señal que conectan los diferentes componentes.
- Conforman los buses internos.
- Estos caminos de señal, en las escalas actuales y a las frecuencias de trabajo actuales generan dos problemas:
 - 1 Delays.
 - 2 Consumo de Energía.

Scaling

- A medida que disminuye el parámetro *tamaño*, los transistores ganan linealmente en rendimiento.
- Los “alambres” son los caminos de señal que conectan los diferentes componentes.
- Conforman los buses internos.
- Estos caminos de señal, en las escalas actuales y a las frecuencias de trabajo actuales generan dos problemas:
 - 1 Delays.
 - 2 Consumo de Energía.

Delays

- A pesar de que los “alambres” también se acortan a medida que reducimos las dimensiones, sus efectos no se reducen en consecuencia.
- Por donde pasar un bus, es un aspecto muy complejo en el diseño.
- Estos “nano alambres” finalmente son un medio de transmisión.
- Con la frecuencia de trabajo actual cobran relevancia las capacidades e inductancias mutuas entre cada “nanoalambre” con su vecino.
- Esto genera que la señal inyectada en un extremo se propague al otro con una determinada demora.
- El problema es si esta demora no es la misma para cada “nanoalambre”.
- A las frecuencias de trabajo actuales las señales pueden llegar a destiempo al extremo receptor.

Delays

- A pesar de que los “alambres” también se acortan a medida que reducimos las dimensiones, sus efectos no se reducen en consecuencia.
- Por donde pasar un bus, es un aspecto muy complejo en el diseño.
- Estos “nano alambres” finalmente son un medio de transmisión.
- Con la frecuencia de trabajo actual cobran relevancia las capacidades e inductancias mutuas entre cada “nanoalambre” con su vecino.
- Esto genera que la señal inyectada en un extremo se propague al otro con una determinada demora.
- El problema es si esta demora no es la misma para cada “nanoalambre”.
- A las frecuencias de trabajo actuales las señales pueden llegar a destiempo al extremo receptor.

Delays

- A pesar de que los “alambres” también se acortan a medida que reducimos las dimensiones, sus efectos no se reducen en consecuencia.
- Por donde pasar un bus, es un aspecto muy complejo en el diseño.
- Estos “nano alambres” finalmente son un medio de transmisión.
- Con la frecuencia de trabajo actual cobran relevancia las capacidades e inductancias mutuas entre cada “nanoalambre” con su vecino.
- Esto genera que la señal inyectada en un extremo se propague al otro con una determinada demora.
- El problema es si esta demora no es la misma para cada “nanoalambre”.
- A las frecuencias de trabajo actuales las señales pueden llegar a destiempo al extremo receptor.

Delays

- A pesar de que los “alambres” también se acortan a medida que reducimos las dimensiones, sus efectos no se reducen en consecuencia.
- Por donde pasar un bus, es un aspecto muy complejo en el diseño.
- Estos “nano alambres” finalmente son un medio de transmisión.
- Con la frecuencia de trabajo actual cobran relevancia las capacidades e inductancias mutuas entre cada “nanoalambre” con su vecino.
- Esto genera que la señal inyectada en un extremo se propague al otro con una determinada demora.
- El problema es si esta demora no es la misma para cada “nanoalambre”.
- A las frecuencias de trabajo actuales las señales pueden llegar a destiempo al extremo receptor.

Delays

- A pesar de que los “alambres” también se acortan a medida que reducimos las dimensiones, sus efectos no se reducen en consecuencia.
- Por donde pasar un bus, es un aspecto muy complejo en el diseño.
- Estos “nano alambres” finalmente son un medio de transmisión.
- Con la frecuencia de trabajo actual cobran relevancia las capacidades e inductancias mutuas entre cada “nanoalambre” con su vecino.
- Esto genera que la señal inyectada en un extremo se propague al otro con una determinada demora.
- El problema es si esta demora no es la misma para cada “nanoalambre”.
- A las frecuencias de trabajo actuales las señales pueden llegar a destiempo al extremo receptor.

Delays

- A pesar de que los “alambres” también se acortan a medida que reducimos las dimensiones, sus efectos no se reducen en consecuencia.
- Por donde pasar un bus, es un aspecto muy complejo en el diseño.
- Estos “nano alambres” finalmente son un medio de transmisión.
- Con la frecuencia de trabajo actual cobran relevancia las capacidades e inductancias mutuas entre cada “nanoalambre” con su vecino.
- Esto genera que la señal inyectada en un extremo se propague al otro con una determinada demora.
- El problema es si esta demora no es la misma para cada “nanoalambre”.
- A las frecuencias de trabajo actuales las señales pueden llegar a destiempo al extremo receptor.

Delays

- A pesar de que los “alambres” también se acortan a medida que reducimos las dimensiones, sus efectos no se reducen en consecuencia.
- Por donde pasar un bus, es un aspecto muy complejo en el diseño.
- Estos “nano alambres” finalmente son un medio de transmisión.
- Con la frecuencia de trabajo actual cobran relevancia las capacidades e inductancias mutuas entre cada “nanoalambre” con su vecino.
- Esto genera que la señal inyectada en un extremo se propague al otro con una determinada demora.
- El problema es si esta demora no es la misma para cada “nanoalambre”.
- A las frecuencias de trabajo actuales las señales pueden llegar a destiempo al extremo receptor.

Consumo / Energía

- En cada "nano alambre" se disipan solo algunos picoJoules de energía cada vez que se transmite una señal de un extremo a otro.
- La energía que disipa cada alambre es despreciable.
- Pero cientos de millones de transistores requieren cientos de millones de cables conectores. Hagan cuentas. . .

Consumo / Energía

- En cada "nano alambre" se disipan solo algunos picoJoules de energía cada vez que se transmite una señal de un extremo a otro.
- La energía que disipa cada alambre es despreciable.
- Pero cientos de millones de transistores requieren cientos de millones de cables conectores. Hagan cuentas. . .

Consumo / Energía

- En cada "nano alambre" se disipan solo algunos picoJoules de energía cada vez que se transmite una señal de un extremo a otro.
- La energía que disipa cada alambre es despreciable.
- Pero cientos de millones de transistores requieren cientos de millones de cables conectores. Hagan cuentas. . .

Consumo / Energía

- La potencia dinámica es la que se disipa en un transistor CMOS en conmutación.
- Es proporcional a la Capacidad de carga del dispositivo, al cuadrado de la tensión de alimentación y a la frecuencia de conmutación.
- En procesadores destinados a dispositivos portátiles, para dimensionar la capacidad de una batería y su tiempo de duración, mas que la potencia, interesa la energía en Joules:

Consumo / Energía

- La potencia dinámica es la que se disipa en un transistor CMOS en conmutación.
- Es proporcional a la Capacidad de carga del dispositivo, al cuadrado de la tensión de alimentación y a la frecuencia de conmutación.
- En procesadores destinados a dispositivos portátiles, para dimensionar la capacidad de una batería y su tiempo de duración, mas que la potencia, interesa la energía en Joules:

Consumo / Energía

- La potencia dinámica es la que se disipa en un transistor CMOS en conmutación.
- Es proporcional a la Capacidad de carga del dispositivo, al cuadrado de la tensión de alimentación y a la frecuencia de conmutación.

$$P_d = \frac{1}{2} C_c V^2 f_c \quad (1)$$

- En procesadores destinados a dispositivos portátiles, para dimensionar la capacidad de una batería y su tiempo de duración, mas que la potencia, interesa la energía en Joules:

Consumo / Energía

- La potencia dinámica es la que se disipa en un transistor CMOS en conmutación.
- Es proporcional a la Capacidad de carga del dispositivo, al cuadrado de la tensión de alimentación y a la frecuencia de conmutación.

$$P_d = \frac{1}{2} C_c V^2 f_c \quad (1)$$

- En procesadores destinados a dispositivos portátiles, para dimensionar la capacidad de una batería y su tiempo de duración, mas que la potencia, interesa la energía en Joules:

Consumo / Energía

- La potencia dinámica es la que se disipa en un transistor CMOS en conmutación.
- Es proporcional a la Capacidad de carga del dispositivo, al cuadrado de la tensión de alimentación y a la frecuencia de conmutación.

$$P_d = \frac{1}{2} C_c V^2 f_c \quad (1)$$

- En procesadores destinados a dispositivos portátiles, para dimensionar la capacidad de una batería y su tiempo de duración, mas que la potencia, interesa la energía en Joules:

$$E_d = C_c V^2 \quad (2)$$

Consumo / Energía

Conclusiones de (1) y (2)

- 1 La tensión de alimentación se redujo en los últimos 30 años de 5V a 0,85V. Esto por sí solo es una reducción drástica en el consumo de un transistor.
- 2 La capacidad de carga depende de la cantidad de dispositivos que se conecten a la salida de un transistor y de la tecnología de integración empleada.
- 3 Para una tarea fija, reducir la frecuencia de clock disminuye la potencia disipada pero no tiene efecto con la energía consumida.

Consumo / Energía

Conclusiones de (1) y (2)

- 1 La tensión de alimentación se redujo en los últimos 30 años de 5V a 0,85V. Esto por sí solo es una reducción drástica en el consumo de un transistor.
- 2 La capacidad de carga depende de la cantidad de dispositivos que se conecten a la salida de un transistor y de la tecnología de integración empleada.
- 3 Para una tarea fija, reducir la frecuencia de clock disminuye la potencia disipada pero no tiene efecto con la energía consumida.

Consumo / Energía

Conclusiones de (1) y (2)

- 1 La tensión de alimentación se redujo en los últimos 30 años de 5V a 0,85V. Esto por sí solo es una reducción drástica en el consumo de un transistor.
- 2 La capacidad de carga depende de la cantidad de dispositivos que se conecten a la salida de un transistor y de la tecnología de integración empleada.
- 3 Para una tarea fija, reducir la frecuencia de clock disminuye la potencia disipada pero no tiene efecto con la energía consumida.

Consumo / Energía

Conclusiones de (1) y (2)

- 1 La tensión de alimentación se redujo en los últimos 30 años de 5V a 0,85V. Esto por sí solo es una reducción drástica en el consumo de un transistor.
- 2 La capacidad de carga depende de la cantidad de dispositivos que se conecten a la salida de un transistor y de la tecnología de integración empleada.
- 3 Para una tarea fija, reducir la frecuencia de clock disminuye la potencia disipada pero no tiene efecto con la energía consumida.

¿Por que razón es un problema el Consumo?

- El incremento en la cantidad de transistores CMOS por mm^2 de superficie tiene preeminencia por sobre los ahorros de energía individuales de cada transistor debidos al cambio de tecnología.
- Por lo tanto cada vez es mas crítico el manejo del consumo.
- El procesador 4004 de Intel en 1971 tenía poco mas de 2300 transistores y su consumo era de algunas décimas de Watts. Su clock era de 108 KHz (si... leyó bien... Kilo Hertz)
- El procesador Pentium IV Extreme Edition desarrollado en 2001 (30 años después), llegó a consumir 135 Watts. Tenía cerca de 40 millones de transistores y un clock de 3GHz
- Por lo tanto cada vez existen mas limitaciones tanto con la distribución de la alimentación como con el ahorro de potencia y energía.

¿Por que razón es un problema el Consumo?

- El incremento en la cantidad de transistores CMOS por mm^2 de superficie tiene preeminencia por sobre los ahorros de energía individuales de cada transistor debidos al cambio de tecnología.
- Por lo tanto cada vez es mas crítico el manejo del consumo.
- El procesador 4004 de Intel en 1971 tenía poco mas de 2300 transistores y su consumo era de algunas décimas de Watts. Su clock era de 108 KHz (si... leyó bien... Kilo Hertz)
- El procesador Pentium IV Extreme Edition desarrollado en 2001 (30 años después), llegó a consumir 135 Watts. Tenía cerca de 40 millones de transistores y un clock de 3GHz
- Por lo tanto cada vez existen mas limitaciones tanto con la distribución de la alimentación como con el ahorro de potencia y energía.

¿Por que razón es un problema el Consumo?

- El incremento en la cantidad de transistores CMOS por mm^2 de superficie tiene preeminencia por sobre los ahorros de energía individuales de cada transistor debidos al cambio de tecnología.
- Por lo tanto cada vez es mas crítico el manejo del consumo.
- El procesador 4004 de Intel en 1971 tenía poco mas de 2300 transistores y su consumo era de algunas décimas de Watts. Su clock era de 108 KHz (si... leyó bien... Kilo Hertz)
- El procesador Pentium IV Extreme Edition desarrollado en 2001 (30 años después), llegó a consumir 135 Watts. Tenía cerca de 40 millones de transistores y un clock de 3GHz
- Por lo tanto cada vez existen mas limitaciones tanto con la distribución de la alimentación como con el ahorro de potencia y energía.

¿Por que razón es un problema el Consumo?

- El incremento en la cantidad de transistores CMOS por mm^2 de superficie tiene preeminencia por sobre los ahorros de energía individuales de cada transistor debidos al cambio de tecnología.
- Por lo tanto cada vez es mas crítico el manejo del consumo.
- El procesador 4004 de Intel en 1971 tenía poco mas de 2300 transistores y su consumo era de algunas décimas de Watts. Su clock era de 108 KHz (si... leyó bien... Kilo Hertz)
- El procesador Pentium IV Extreme Edition desarrollado en 2001 (30 años después), llegó a consumir 135 Watts. Tenía cerca de 40 millones de transistores y un clock de 3GHz
- Por lo tanto cada vez existen mas limitaciones tanto con la distribución de la alimentación como con el ahorro de potencia y energía.

¿Por que razón es un problema el Consumo?

- El incremento en la cantidad de transistores CMOS por mm^2 de superficie tiene preeminencia por sobre los ahorros de energía individuales de cada transistor debidos al cambio de tecnología.
- Por lo tanto cada vez es mas crítico el manejo del consumo.
- El procesador 4004 de Intel en 1971 tenía poco mas de 2300 transistores y su consumo era de algunas décimas de Watts. Su clock era de 108 KHz (si... leyó bien... Kilo Hertz)
- El procesador Pentium IV Extreme Edition desarrollado en 2001 (30 años después), llegó a consumir 135 Watts. Tenía cerca de 40 millones de transistores y un clock de 3GHz
- Por lo tanto cada vez existen mas limitaciones tanto con la distribución de la alimentación como con el ahorro de potencia y energía.

Tendencias en reducción del Consumo

- La mayoría de los procesadores actuales contiene bloques de hardware para control de consumo, que se encargan de mantener alimentados solo los bloques funcionales que se necesitan en cada momento.
- Si el procesador no está ejecutando operaciones de Punto Flotante, entonces la Unidad de Punto Flotante se mantiene apagada.
- Lo mismo con cada Unidad interna.

Tendencias en reducción del Consumo

- La mayoría de los procesadores actuales contiene bloques de hardware para control de consumo, que se encargan de mantener alimentados solo los bloques funcionales que se necesitan en cada momento.
- Si el procesador no está ejecutando operaciones de Punto Flotante, entonces la Unidad de Punto Flotante se mantiene apagada.
- Lo mismo con cada Unidad interna.

Tendencias en reducción del Consumo

- La mayoría de los procesadores actuales contiene bloques de hardware para control de consumo, que se encargan de mantener alimentados solo los bloques funcionales que se necesitan en cada momento.
- Si el procesador no está ejecutando operaciones de Punto Flotante, entonces la Unidad de Punto Flotante se mantiene apagada.
- Lo mismo con cada Unidad interna.

Corrientes de fuga (leakage)

- A pesar de que un transistor esté al corte, circula una muy pequeña de todos modos
- Interesa determinar la Potencia estática, relacionada con esta corriente:
- V = Tensión de Alimentación, I_e = corriente de fuga (leakage). Cada transistor tiene así una componente adicional de potencia cuando está en corte.
- A medida que aumenta la cantidad de transistores esta corriente se hace mas significativa.
- En 2006 los principales diseñadores establecieron como meta que esta corriente represente solo el 25 % del consumo total del chip. Aún así los modelos de mas alto rendimiento no lograron esta marca.

Corrientes de fuga (leakage)

- A pesar de que un transistor esté al corte, circula una muy pequeña de todos modos
- Interesa determinar la Potencia estática, relacionada con esta corriente:
- V = Tensión de Alimentación, I_e = corriente de fuga (leakage). Cada transistor tiene así una componente adicional de potencia cuando está en corte.
- A medida que aumenta la cantidad de transistores esta corriente se hace mas significativa.
- En 2006 los principales diseñadores establecieron como meta que esta corriente represente solo el 25 % del consumo total del chip. Aún así los modelos de mas alto rendimiento no lograron esta marca.

Corrientes de fuga (leakage)

- A pesar de que un transistor esté al corte, circula una muy pequeña de todos modos
- Interesa determinar la Potencia estática, relacionada con esta corriente:

$$E_e = I_e V^2 \quad (3)$$

- V = Tensión de Alimentación, I_e = corriente de fuga (leakage). Cada transistor tiene así una componente adicional de potencia cuando está en corte.
- A medida que aumenta la cantidad de transistores esta corriente se hace mas significativa.
- En 2006 los principales diseñadores establecieron como meta que esta corriente represente solo el 25 % del consumo total del chip. Aún así los modelos de mas alto rendimiento no lograron esta marca.

Corrientes de fuga (leakage)

- A pesar de que un transistor esté al corte, circula una muy pequeña de todos modos
- Interesa determinar la Potencia estática, relacionada con esta corriente:

$$E_e = I_e V^2 \quad (3)$$

- V = Tensión de Alimentación, I_e = corriente de fuga (leakage). Cada transistor tiene así una componente adicional de potencia cuando está en corte.
- A medida que aumenta la cantidad de transistores esta corriente se hace mas significativa.
- En 2006 los principales diseñadores establecieron como meta que esta corriente represente solo el 25 % del consumo total del chip. Aún así los modelos de mas alto rendimiento no lograron esta marca.

Corrientes de fuga (leakage)

- A pesar de que un transistor esté al corte, circula una muy pequeña de todos modos
- Interesa determinar la Potencia estática, relacionada con esta corriente:

$$E_e = I_e V^2 \quad (3)$$

- V = Tensión de Alimentación, I_e = corriente de fuga (leakage). Cada transistor tiene así una componente adicional de potencia cuando está en corte.
- A medida que aumenta la cantidad de transistores esta corriente se hace mas significativa.
- En 2006 los principales diseñadores establecieron como meta que esta corriente represente solo el 25 % del consumo total del chip. Aún así los modelos de mas alto rendimiento no lograron esta marca.

Corrientes de fuga (leakage)

- A pesar de que un transistor esté al corte, circula una muy pequeña de todos modos
- Interesa determinar la Potencia estática, relacionada con esta corriente:

$$E_e = I_e V^2 \quad (3)$$

- V = Tensión de Alimentación, I_e = corriente de fuga (leakage). Cada transistor tiene así una componente adicional de potencia cuando está en corte.
- A medida que aumenta la cantidad de transistores esta corriente se hace mas significativa.
- En 2006 los principales diseñadores establecieron como meta que esta corriente represente solo el 25 % del consumo total del chip. Aún así los modelos de mas alto rendimiento no lograron esta marca.

Distribución de la tensión de alimentación

- La alimentación en un circuito integrado moderno es otro tema a considerar, por varios factores.
- Se debe distribuir la tensión de alimentación a todo el circuito integrado. Esto motiva
- Desde hace mas de una década que los circuitos integrados dediquen una buena cantidad de terminales de conexión a V_{DD} y Tierra.

Arquitectura vs Microarquitectura

Arquitectura

Es el conjunto de recursos accesibles para el programador, que por lo general se mantienen a lo largo de los diferentes modelos de procesadores de esa arquitectura (puede evolucionar pero la tendencia es mantener compatibilidad hacia los modelos anteriores).

- Registros
- Set de instrucciones
- Estructuras de memoria (descriptores de segmento y de página p. ej.)

Micro Arquitectura

Es la implementación en el silicio de la arquitectura. Lo que está detrás del set de registros y del modelo de programación. Puede ser muy simple o sumamente robusta y poderosa. Cambia de un modelo a otro dentro de una misma familia.

Arquitectura vs Microarquitectura

Arquitectura

Es el conjunto de recursos accesibles para el programador, que por lo general se mantienen a lo largo de los diferentes modelos de procesadores de esa arquitectura (puede evolucionar pero la tendencia es mantener compatibilidad hacia los modelos anteriores).

- Registros
- Set de instrucciones
- Estructuras de memoria (descriptores de segmento y de página p. ej.)

Micro Arquitectura

Es la implementación en el silicio de la arquitectura. Lo que está detrás del set de registros y del modelo de programación. Puede ser muy simple o sumamente robusta y poderosa. Cambia de un modelo a otro dentro de una misma familia.

Arquitectura vs Microarquitectura

Arquitectura

Es el conjunto de recursos accesibles para el programador, que por lo general se mantienen a lo largo de los diferentes modelos de procesadores de esa arquitectura (puede evolucionar pero la tendencia es mantener compatibilidad hacia los modelos anteriores).

- Registros
- Set de instrucciones
- Estructuras de memoria (descriptores de segmento y de página p. ej.)

Micro Arquitectura

Es la implementación en el silicio de la arquitectura. Lo que está detrás del set de registros y del modelo de programación. Puede ser muy simple o sumamente robusta y poderosa. Cambia de un modelo a otro dentro de una misma familia.

Arquitectura vs Microarquitectura

Arquitectura

Es el conjunto de recursos accesibles para el programador, que por lo general se mantienen a lo largo de los diferentes modelos de procesadores de esa arquitectura (puede evolucionar pero la tendencia es mantener compatibilidad hacia los modelos anteriores).

- Registros
- Set de instrucciones
- Estructuras de memoria (descriptores de segmento y de página p. ej.)

Micro Arquitectura

Es la implementación en el silicio de la arquitectura. Lo que está detrás del set de registros y del modelo de programación. Puede ser muy simple o sumamente robusta y poderosa. Cambia de un modelo a otro dentro de una misma familia.

Arquitectura vs Microarquitectura

Arquitectura

Es el conjunto de recursos accesibles para el programador, que por lo general se mantienen a lo largo de los diferentes modelos de procesadores de esa arquitectura (puede evolucionar pero la tendencia es mantener compatibilidad hacia los modelos anteriores).

- Registros
- Set de instrucciones
- Estructuras de memoria (descriptores de segmento y de página p. ej.)

Micro Arquitectura

Es la implementación en el silicio de la arquitectura. Lo que está detrás del set de registros y del modelo de programación. Puede ser muy simple o sumamente robusta y poderosa. Cambia de un modelo a otro dentro de una misma familia.

Arquitectura vs Microarquitectura

Ejemplo

La arquitectura IA-32 se inicia con el procesador 80386 en 1985, y llega hasta los procesadores Intel Core i7, i5, i3, ATOM y Xeon actuales.

En el camino han pasado diferentes generaciones de Micro-Arquitectura para mas de 25 modelos de procesadores.

Definición de la Arquitectura de un Computador

- Es sin dudas la tarea mas ardua de un diseñador.
- Se trata de definir los aspectos mas relevantes en la arquitectura de un computador que maximicen su rendimiento, sin dejar de satisfacer otras limitaciones impuestas por los usuarios, como un costo económico que lo haga accesible, o un consumo de energía moderado.
- Comprende el diseño del set de instrucciones, el manejo de la memoria y sus modos de direccionamiento, los restantes bloques funcionales que componen el CPU, el diseño lógico, y el proceso de implementación
- La implementación comprende el diseño del circuito integrado, su encapsulado, montaje, alimentación y refrigeración.

Definición de la Arquitectura de un Computador

- Es sin dudas la tarea mas ardua de un diseñador.
- Se trata de definir los aspectos mas relevantes en la arquitectura de un computador que maximicen su rendimiento, sin dejar de satisfacer otras limitaciones impuestas por los usuarios, como un costo económico que lo haga accesible, o un consumo de energía moderado.
- Comprende el diseño del set de instrucciones, el manejo de la memoria y sus modos de direccionamiento, los restantes bloques funcionales que componen el CPU, el diseño lógico, y el proceso de implementación
- La implementación comprende el diseño del circuito integrado, su encapsulado, montaje, alimentación y refrigeración.

Definición de la Arquitectura de un Computador

- Es sin dudas la tarea mas ardua de un diseñador.
- Se trata de definir los aspectos mas relevantes en la arquitectura de un computador que maximicen su rendimiento, sin dejar de satisfacer otras limitaciones impuestas por los usuarios, como un costo económico que lo haga accesible, o un consumo de energía moderado.
- Comprende el diseño del set de instrucciones, el manejo de la memoria y sus modos de direccionamiento, los restantes bloques funcionales que componen el CPU, el diseño lógico, y el proceso de implementación
- La implementación comprende el diseño del circuito integrado, su encapsulado, montaje, alimentación y refrigeración.

Definición de la Arquitectura de un Computador

- Es sin dudas la tarea mas ardua de un diseñador.
- Se trata de definir los aspectos mas relevantes en la arquitectura de un computador que maximicen su rendimiento, sin dejar de satisfacer otras limitaciones impuestas por los usuarios, como un costo económico que lo haga accesible, o un consumo de energía moderado.
- Comprende el diseño del set de instrucciones, el manejo de la memoria y sus modos de direccionamiento, los restantes bloques funcionales que componen el CPU, el diseño lógico, y el proceso de implementación
- La implementación comprende el diseño del circuito integrado, su encapsulado, montaje, alimentación y refrigeración.

Definición de la Arquitectura de un Computador

skills necesarios

No es posible realizar esta tarea con éxito sin tener manejar de manera solvente varias tecnologías diferentes:

- Diseño lógico.
- Tecnología de encapsulado
- Funcionamiento y diseño de compiladores y de Sistemas Operativos.

Definición de la Arquitectura de un Computador

skills necesarios

No es posible realizar esta tarea con éxito sin tener manejar de manera solvente varias tecnologías diferentes:

- Diseño lógico.
- Tecnología de encapsulado
- Funcionamiento y diseño de compiladores y de Sistemas Operativos.

Definición de la Arquitectura de un Computador

skills necesarios

No es posible realizar esta tarea con éxito sin tener manejar de manera solvente varias tecnologías diferentes:

- Diseño lógico.
- Tecnología de encapsulado
- Funcionamiento y diseño de compiladores y de Sistemas Operativos.

Definición de la Arquitectura de un Computador

skills necesarios

No es posible realizar esta tarea con éxito sin tener manejar de manera solvente varias tecnologías diferentes:

- Diseño lógico.
- Tecnología de encapsulado
- Funcionamiento y diseño de compiladores y de Sistemas Operativos.

Temario

- 1 Tecnología de Integración
- 2 **Arquitectura de Computadores**
 - **Instruction Set Architecture (ISA)**
 - Organización y Hardware
- 3 Objetivos de estudiar organización y hardware

Definiendo un ISA

Nos referimos a *Instruction Set Architecture*, como el set de instrucciones visibles por el programador. Es además el límite entre el software y el hardware.

Clases de ISA. ISAs con Registros de Propósito general vs. Registros dedicados. ISAs registro-memoria vs. ISAs Load Store.

Direccionamiento de Memoria. Alineación obligatoria de datos vs. administración de a bytes.

Modos de Direccionamiento. Como se especifican los operandos.

Tipos y tamaños de operandos. Enteros, Punto Flotante, Punto Fijo. Diferentes tamaños y precisiones.

Operaciones. Pocas Simples (RISC) o Muchas Complejas (CISC).

Instrucciones de control de flujo Saltos condicionales, calls.

Longitud del código Instrucciones de tamaño fijo vs. variable.

Definiendo un ISA

Nos referimos a *Instruction Set Architecture*, como el set de instrucciones visibles por el programador. Es además el límite entre el software y el hardware.

Clases de ISA. ISAs con Registros de Propósito general vs. Registros dedicados. ISAs registro-memoria vs. ISAs Load Store.

Direccionamiento de Memoria. Alineación obligatoria de datos vs. administración de a bytes.

Modos de Direccionamiento. Como se especifican los operandos.

Tipos y tamaños de operandos. Enteros, Punto Flotante, Punto Fijo. Diferentes tamaños y precisiones.

Operaciones. Pocas Simples (RISC) o Muchas Complejas (CISC).

Instrucciones de control de flujo Saltos condicionales, calls.

Longitud del código Instrucciones de tamaño fijo vs. variable.

Definiendo un ISA

Nos referimos a *Instruction Set Architecture*, como el set de instrucciones visibles por el programador. Es además el límite entre el software y el hardware.

Clases de ISA. ISAs con Registros de Propósito general vs. Registros dedicados. ISAs registro-memoria vs. ISAs Load Store.

Direccionamiento de Memoria. Alineación obligatoria de datos vs. administración de a bytes.

Modos de Direccionamiento. Como se especifican los operandos.

Tipos y tamaños de operandos. Enteros, Punto Flotante, Punto Fijo. Diferentes tamaños y precisiones.

Operaciones. Pocas Simples (RISC) o Muchas Complejas (CISC).

Instrucciones de control de flujo Saltos condicionales, calls.

Longitud del código Instrucciones de tamaño fijo vs. variable.

Definiendo un ISA

Nos referimos a *Instruction Set Architecture*, como el set de instrucciones visibles por el programador. Es además el límite entre el software y el hardware.

Clases de ISA. ISAs con Registros de Propósito general vs. Registros dedicados. ISAs registro-memoria vs. ISAs Load Store.

Direccionamiento de Memoria. Alineación obligatoria de datos vs. administración de a bytes.

Modos de Direccionamiento. Como se especifican los operandos.

Tipos y tamaños de operandos. Enteros, Punto Flotante, Punto Fijo. Diferentes tamaños y precisiones.

Operaciones. Pocas Simples (RISC) o Muchas Complejas (CISC).

Instrucciones de control de flujo Saltos condicionales, calls.

Longitud del código Instrucciones de tamaño fijo vs. variable.

Definiendo un ISA

Nos referimos a *Instruction Set Architecture*, como el set de instrucciones visibles por el programador. Es además el límite entre el software y el hardware.

Clases de ISA. ISAs con Registros de Propósito general vs. Registros dedicados. ISAs registro-memoria vs. ISAs Load Store.

Direccionamiento de Memoria. Alineación obligatoria de datos vs. administración de a bytes.

Modos de Direccionamiento. Como se especifican los operandos.

Tipos y tamaños de operandos. Enteros, Punto Flotante, Punto Fijo. Diferentes tamaños y precisiones.

Operaciones. Pocas Simples (RISC) o Muchas Complejas (CISC).

Instrucciones de control de flujo Saltos condicionales, calls.

Longitud del código Instrucciones de tamaño fijo vs. variable.

Definiendo un ISA

Nos referimos a *Instruction Set Architecture*, como el set de instrucciones visibles por el programador. Es además el límite entre el software y el hardware.

Clases de ISA. ISAs con Registros de Propósito general vs. Registros dedicados. ISAs registro-memoria vs. ISAs Load Store.

Direccionamiento de Memoria. Alineación obligatoria de datos vs. administración de a bytes.

Modos de Direccionamiento. Como se especifican los operandos.

Tipos y tamaños de operandos. Enteros, Punto Flotante, Punto Fijo. Diferentes tamaños y precisiones.

Operaciones. Pocas Simples (RISC) o Muchas Complejas (CISC).

Instrucciones de control de flujo Saltos condicionales, calls.

Longitud del código Instrucciones de tamaño fijo vs. variable.

Definiendo un ISA

Nos referimos a *Instruction Set Architecture*, como el set de instrucciones visibles por el programador. Es además el límite entre el software y el hardware.

Clases de ISA. ISAs con Registros de Propósito general vs. Registros dedicados. ISAs registro-memoria vs. ISAs Load Store.

Direccionamiento de Memoria. Alineación obligatoria de datos vs. administración de a bytes.

Modos de Direccionamiento. Como se especifican los operandos.

Tipos y tamaños de operandos. Enteros, Punto Flotante, Punto Fijo. Diferentes tamaños y precisiones.

Operaciones. Pocas Simples (RISC) o Muchas Complejas (CISC).

Instrucciones de control de flujo Saltos condicionales, calls.

Longitud del código Instrucciones de tamaño fijo vs. variable.

Definiendo un ISA

Nos referimos a *Instruction Set Architecture*, como el set de instrucciones visibles por el programador. Es además el límite entre el software y el hardware.

Clases de ISA. ISAs con Registros de Propósito general vs. Registros dedicados. ISAs registro-memoria vs. ISAs Load Store.

Direccionamiento de Memoria. Alineación obligatoria de datos vs. administración de a bytes.

Modos de Direccionamiento. Como se especifican los operandos.

Tipos y tamaños de operandos. Enteros, Punto Flotante, Punto Fijo. Diferentes tamaños y precisiones.

Operaciones. Pocas Simples (RISC) o Muchas Complejas (CISC).

Instrucciones de control de flujo Saltos condicionales, calls.

Longitud del código Instrucciones de tamaño fijo vs. variable.

Temario

- 1 Tecnología de Integración
- 2 **Arquitectura de Computadores**
 - Instruction Set Architecture (ISA)
 - **Organización y Hardware**
- 3 Objetivos de estudiar organización y hardware

Microarquitectura = Organización + Hardware

Organización

Se refiere a los detalles de implementación de la ISA. Es decir

- Organización e interconexión de la memoria.
- Diseño de los diferentes bloques de la CPU y para implementar el set de instrucciones.
- Implementación de paralelismo a nivel de Instrucciones, y/o de datos.
- Podemos así encontrar procesadores que poseen la misma ISA pero una organización muy diferente.

Microarquitectura = Organización + Hardware

Organización

Se refiere a los detalles de implementación de la ISA. Es decir

- Organización e interconexión de la memoria.
- Diseño de los diferentes bloques de la CPU y para implementar el set de instrucciones.
- Implementación de paralelismo a nivel de Instrucciones, y/o de datos.
- Podemos así encontrar procesadores que poseen la misma ISA pero una organización muy diferente.

Microarquitectura = Organización + Hardware

Organización

Se refiere a los detalles de implementación de la ISA. Es decir

- Organización e interconexión de la memoria.
- Diseño de los diferentes bloques de la CPU y para implementar el set de instrucciones.
- Implementación de paralelismo a nivel de Instrucciones, y/o de datos.
- Podemos así encontrar procesadores que poseen la misma ISA pero una organización muy diferente.

Microarquitectura = Organización + Hardware

Organización

Se refiere a los detalles de implementación de la ISA. Es decir

- Organización e interconexión de la memoria.
- Diseño de los diferentes bloques de la CPU y para implementar el set de instrucciones.
- Implementación de paralelismo a nivel de Instrucciones, y/o de datos.
- Podemos así encontrar procesadores que poseen la misma ISA pero una organización muy diferente.

Microarquitectura = Organización + Hardware

Organización

Se refiere a los detalles de implementación de la ISA. Es decir

- Organización e interconexión de la memoria.
- Diseño de los diferentes bloques de la CPU y para implementar el set de instrucciones.
- Implementación de paralelismo a nivel de Instrucciones, y/o de datos.
- Podemos así encontrar procesadores que poseen la misma ISA pero una organización muy diferente.

Ejemplo:

Los procesadores AMD FX y los Intel Core i7, tienen la misma ISA, sin embargo organizan su caché y su motor de ejecución de maneras diferentes.

Microarquitectura = Organización + Hardware

Hardware

- Se refiere a los detalles de diseño lógico y tecnología de fabricación.
- Existirán por lo tanto, procesadores con la misma ISA y la misma organización, pero absolutamente distintos a nivel de hardware y diseño lógico detallado.

Microarquitectura = Organización + Hardware

Hardware

- Se refiere a los detalles de diseño lógico y tecnología de fabricación.
- Existirán por lo tanto, procesadores con la misma ISA y la misma organización, pero absolutamente distintos a nivel de hardware y diseño lógico detallado.

Microarquitectura = Organización + Hardware

Hardware

- Se refiere a los detalles de diseño lógico y tecnología de fabricación.
- Existirán por lo tanto, procesadores con la misma ISA y la misma organización, pero absolutamente distintos a nivel de hardware y diseño lógico detallado.

Microarquitectura = Organización + Hardware

Hardware

- Se refiere a los detalles de diseño lógico y tecnología de fabricación.
- Existirán por lo tanto, procesadores con la misma ISA y la misma organización, pero absolutamente distintos a nivel de hardware y diseño lógico detallado.

Ejemplo:

El Pentium 4, diseñado para desktop, y el Pentium 4 M para notebooks. Este último tiene una cantidad de lógica para control del consumo de energía, siendo su hardware muy diferente del del Pentium 4 desktop.

¿Porque necesitamos saber todo esto?

- Para comprender lo que ocurre debajo del software.
- Comprender como los diseños de hardware impactan en el software y en el programador
- Estar en condiciones de cruzar verticalmente las capas que separan el silicio de la aplicación.
- Comprender las nociones básicas que rigen el diseño de un sistema de cómputo moderno

¿Porque necesitamos saber todo esto?

- Para comprender lo que ocurre debajo del software.
- Comprender como los diseños de hardware impactan en el software y en el programador
- Estar en condiciones de cruzar verticalmente las capas que separan el silicio de la aplicación.
- Comprender las nociones básicas que rigen el diseño de un sistema de cómputo moderno

¿Porque necesitamos saber todo esto?

- Para comprender lo que ocurre debajo del software.
- Comprender como los diseños de hardware impactan en el software y en el programador
- Estar en condiciones de cruzar verticalmente las capas que separan el silicio de la aplicación.
- Comprender las nociones básicas que rigen el diseño de un sistema de cómputo moderno

¿Porque necesitamos saber todo esto?

- Para comprender lo que ocurre debajo del software.
- Comprender como los diseños de hardware impactan en el software y en el programador
- Estar en condiciones de cruzar verticalmente las capas que separan el silicio de la aplicación.
- Comprender las nociones básicas que rigen el diseño de un sistema de cómputo moderno

¿Porque necesitamos saber todo esto?

- Para comprender lo que ocurre debajo del software.
- Comprender como los diseños de hardware impactan en el software y en el programador
- Estar en condiciones de cruzar verticalmente las capas que separan el silicio de la aplicación.
- Comprender las nociones básicas que rigen el diseño de un sistema de cómputo moderno

¿Preguntas?