-Hardware para sistemas empotrados-

pedraza@datsi.fi.upm.es

Sistemas empotrados y ubicuos 2022. MUII.

Índice

- Introducción / HW para sistemas empotrados
- Características y componentes de un sistema empotrado

- Procesadores:
  - CPU / FPGA / DSP / VLIW
- Computadores modulares
- Problemas/Ideas/Técnicas utilizadas en s. empotrados

- Normativa (Actualmente en grado en II: 'Proyecto de Instalación Informática')

Bibliografía

- ***Marwedel, P. Embedded System Design. Springer 2006 (4ª ed, 2021).***
  https://link.springer.com/book/10.1007/978-3-030-60910-8
- ***Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems, A Cyber-Physical Systems Approach, (Second Edition, V. 2.2 2022).***
  http://leeseshia.org/index.html

- Yaghmour, K., Masters, J., Ben-Yossef, G., Gerum, P., *Building Embedded Linux Systems* (2nd ed.), O'Reilly, 2008.
- Catsoulis, J. *Designing Embedded Hardware*. 2nd Edition. O'Reilly Media, 2005.
- https://es.coursera.org/learn/iot (Coursera, Introduction to the Internet of Things and Embedded Systems)
- https://www.edx.org/es/course/introduction-to-the-internet-of-things-iot (edX, Introduction to the Internet of Things –IOT)

- Stallings, W. *Organización y arquitectura de computadores*. Prentice Hall. 7ª Edición, 2006.

Introducción

- **Lee and Seshia**
  ***Introduction to Embedded Systems* (v2.2 2022)**

- The most visible use of computers and software is processing information for human consumption. The vast majority of computers in use, however, are much less visible. They run the engine, brakes, seatbelts, airbag, and audio system in your car. They digitally encode your voice and construct a radio signal to send it from your cell phone to a base station. They command robots on a factory floor, power generation in a power plant, processes in a chemical plant, and traffic lights in a city.
  These less visible computers are called embedded systems, and the software they run is called embedded software.

- The principal challenges in designing and analyzing embedded systems stem from their interaction with physical processes.

Introducción

- No existe

  "Hardware estándar de sistemas empotrados"

  ya que no existe un

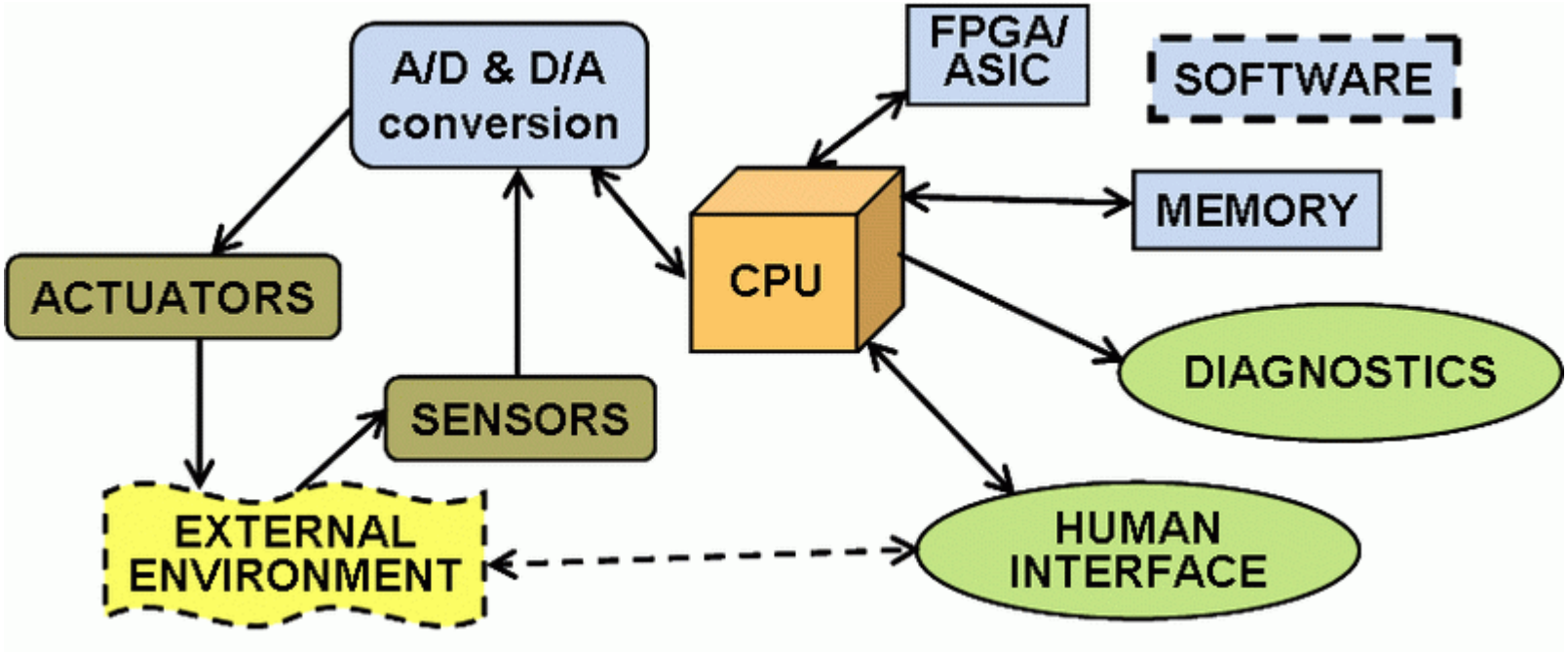
  "Sistema empotrado estándar"

- Sí hay características particulares de *un* sistema empotrado:
  - Bajo consumo / consumo regulable (alimentación, memoria)
  - Tiempo real. Tolerancia a fallos
  - Inmunidad a radiación / a ruido electromagnético
  - Robustez del dispositivo/encapsulado del sistema
  - Altas prestaciones: en cómputo, en comunicaciones, …
  - Bajo coste de fabricación / de desarrollo, …

Sistema empotrado (MSerra, U Victoria)

Cualquiera que incluya un dispositivo programable, con propósito específico[1] y que disponga de múltiples unidades de interfaz para medir, controlar y/o manipular el entorno.

(http://webhome.cs.uvic.ca/~mserra/HScodesign.html Micaela Serra)



[1] nota: “no de propósito general”

Sistema empotrado (CMU-2005)

- Undergraduate Embedded System Education at Carnegie Mellon
  (ACM Transactions on Embedded Computing Systems, Vol. 4, No. 3, August 2005, Pages 500–528)

  - Embedded **applications** are very diverse and span a tremendous range of complexity.

  - Indeed, embedded **computing** is more readily defined by what it is not (it is not generic application software executing on the main CPU of a "desktop computer") than what it is.

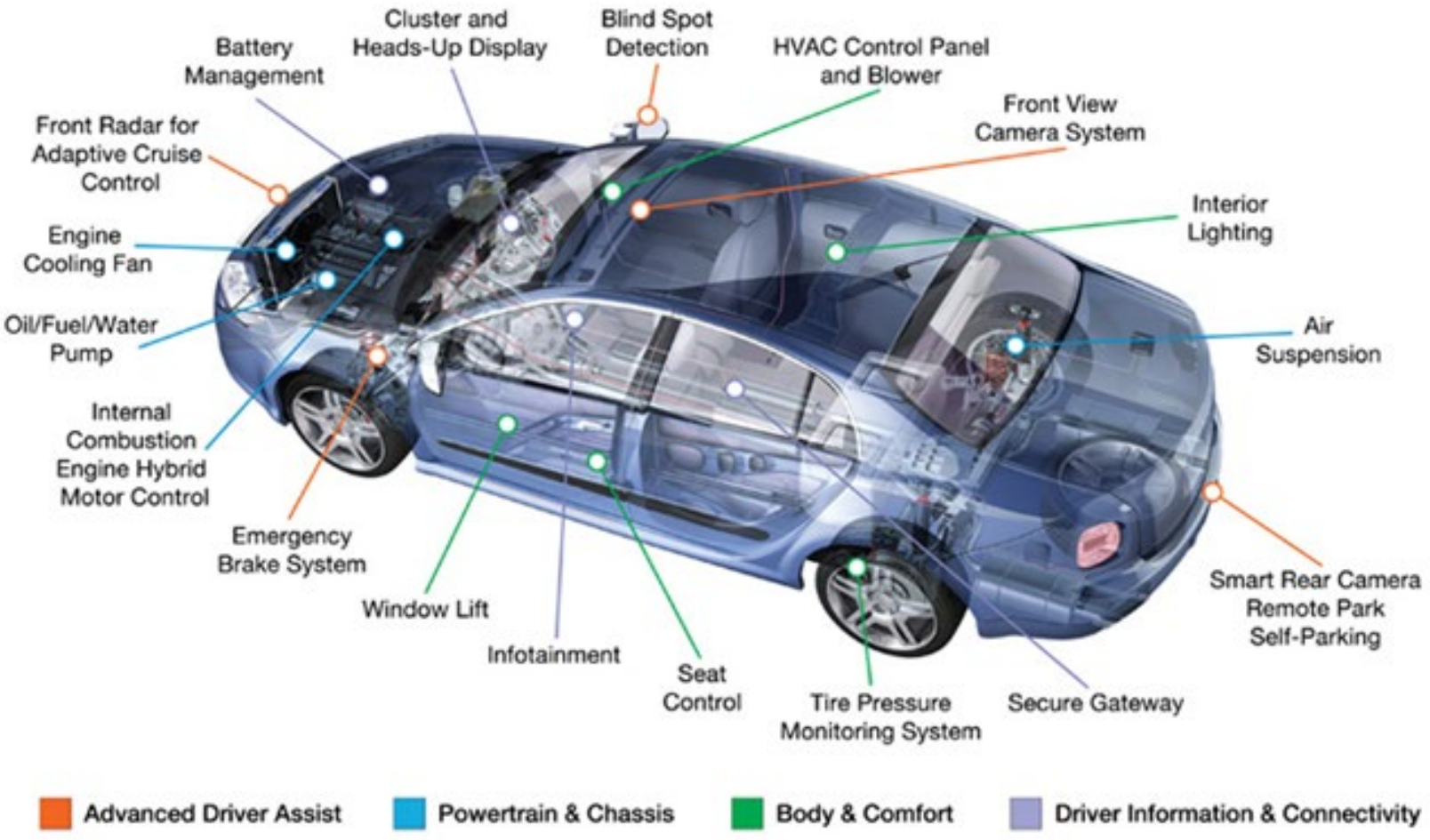
  - Embedded **CPUs** comprise the vast majority of processors, although many of those processors are used in very high-volume applications.

  - We believe that it is the rare **embedded system developer** who will need, or even be able to absorb, expertise in all areas of embedded systems.

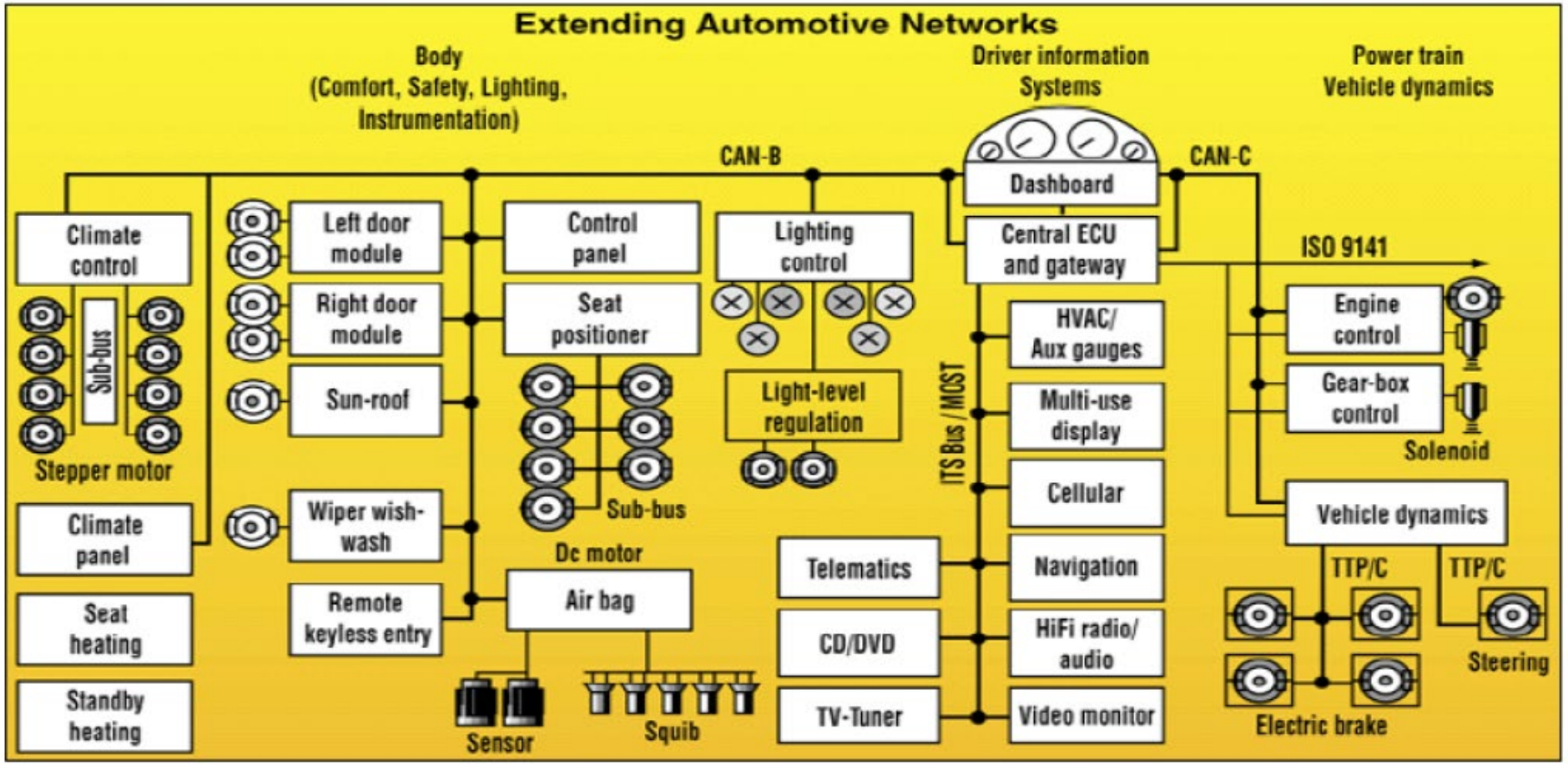
Características S. empotrado (CMU)

- From an application perspective, CMU think of embedded computing as falling into the following distinct categories:
  - Small and single microcontroller applications. Contador de personas, Riego automático, CO2…
  - Control systems. Control de posición de un motor eléctrico (robot, seguimiento cámara, control de velocidad de vehículo), control de temperatura HVAC, etc.
  - Distributed embedded control. Sistemas de automoción: ABS, Airbags, ESP, etc
  - System on chip. Reproductores multimedia, máquina de fotos, teléfono inalámbrico, móvil, etc.
  - Networking. Dispositivos de conexión: routers, switches, wi-fi, etc.
  - Embedded PCs. Sistemas de control de una fábrica, una planta industrial, grua robotizada, etc.
  - Critical systems. Seguridad de vehículos (coches, aviones, satélites), instrumental médico, etc.
  - Robotics. Robots industriales, exploradores, vehículos autónomos, robots caminantes, juguetes, etc.
  - Computer peripherals. Controladores gráficos, de red, de audio, Impresoras, proyectores, etc.
  - Wireless data systems. Mandos a distancia, dispositivos bluetooth (p.e. manos libres), etc.
  - Signal processing. Procesamiento de audio/vídeo, filtros digitales de sonido, etc.
  - Command and control. Sistemas de control de vehículos (coches, aviones, trenes, tranvías). Sistemas de señalización (p.e. trenes)

Ejemplo: automoción (I)



Automoción (II)



(V. Rodellar, Informática Industrial, GII)

IoT: Far-reaching Implications

1. It connects **things** beyond people
   - Can track and control many devices

2. It is **physical** beyond information
   - Can directly impact physical aspects of our life (comfort, health, safety, green, ...)

3. It empowers **sensing and reasoning** about the environment
   - Can be an assistant or agent for human beings, robots, or autonomous vehicles

All these mean new interdisciplinary opportunities and challenges, in areas such as devices, networking, systems, machine learning, security and privacy

IoT in Two Markets

1. IoT for **enterprises**
   - Smart meters (utility companies), smart vehicles (car companies), intelligent healthcare (insurance companies), ...
2. IoT for **consumers**
   - Wearables such as smart watches, smart bands, and intelligent headsets
   - Smart homes which are open to various consumer IoT appliances
   - Many new IoT devices being introduced every week

Curso Harvard : Secure and Intelligent Internet of Things (2014)

- Population of the world:
  7 billion
- Population of Internet users:
  3 billion
- Population of mobile phone users:
  1 billion

A game of asserting multiples

- Suppose there will have 10 IoT devices per mobile user
  ➔ then we have a ***multiple of 10***

- This means 10 billion IoT devices in year like 2020. Indeed:
  26 billion predicted by Gartner,
  30 billion by ABI Research, and
  50 billion by Cisco

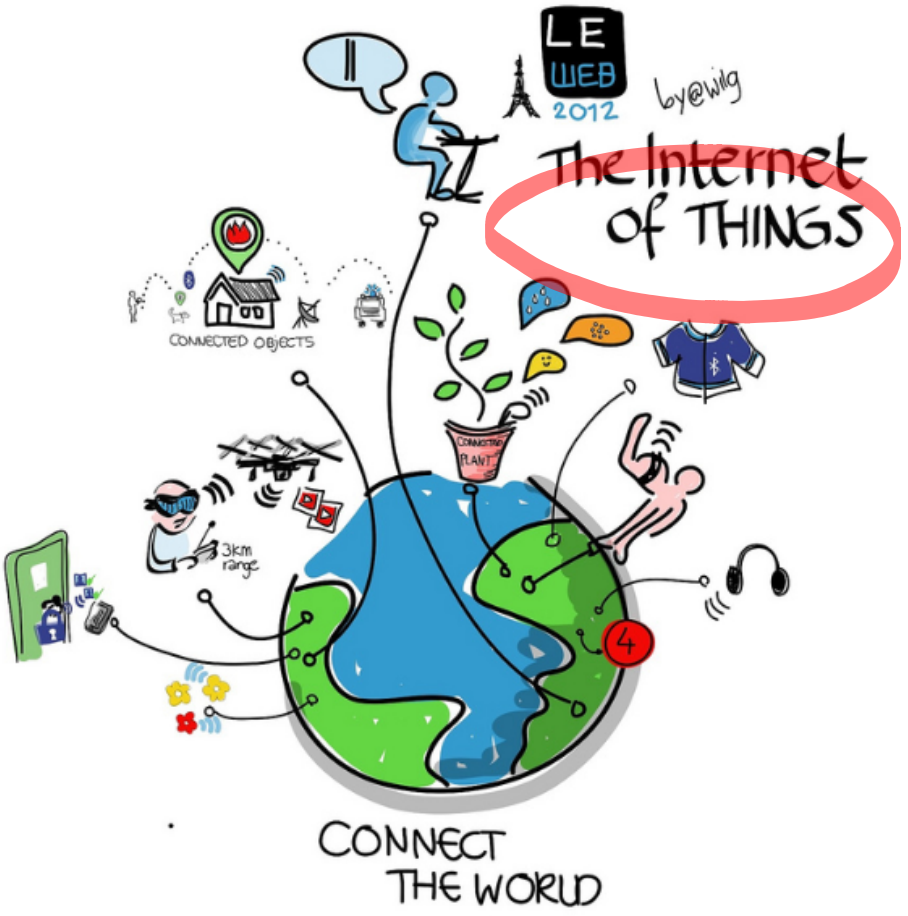


Image courtesy: Wilgengebroed

Véase también https://findstack.com/es/internet-of-things-statistics/

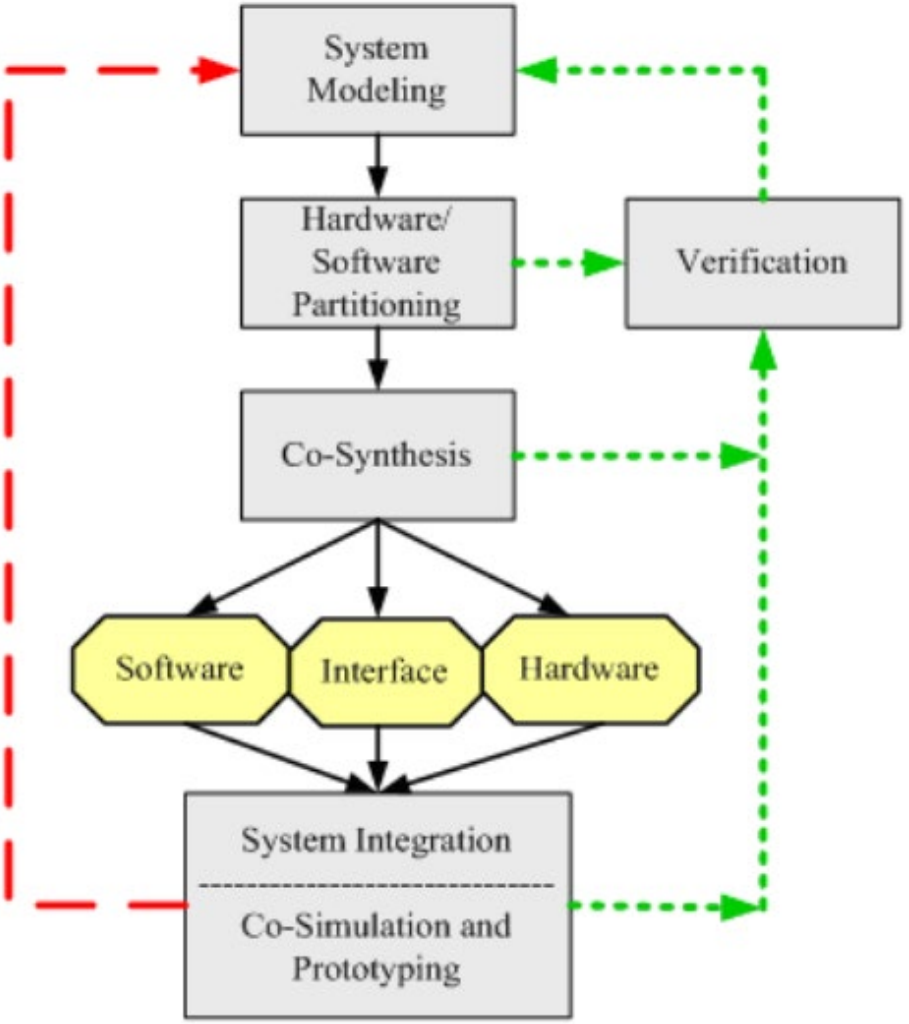
Diseño de un sistema empotrado (I)

Hardware-Software Codesign. Micaela Serra, University of Victoria.

- Necesario considerar:
  → Hardware + Software
- Metodología apropiada: basada en reutilización:
  → Codiseño HW-SW

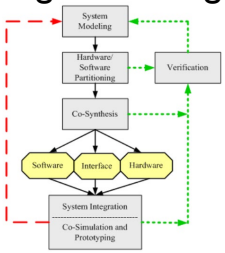
- The co-design of HW/SW systems may be viewed as composed of four main phases as illustrated in in the side diagram:

1. Modeling
2. Partitioning
3. Co-Synthesis
4. Co-Simulation



Diseño de un sistema empotrado (II)

- **Modeling** involves specifying the concepts and the constraints of the system to obtain a refined specification. This phase of the design also specifies a software and hardware model. The first problem is to find a suitable specification methodology for a target system. Some researchers favour a formal language which can yield provably-correct code. But most prefer a hardware-type language (e.g., VHDL, Verilog), a software-type language (C, C++, Handel-C, SystemC), or other formalism lacking a hardware or software bias (such as Codesign Finite State Machines). There are three different paths the modeling process can take, considering its starting point:
  - An existing software implementation of the problem.
  - An existing hardware, for example a chip, is present.
  - None of the above is given, only specifications leaving an open choice for a model.



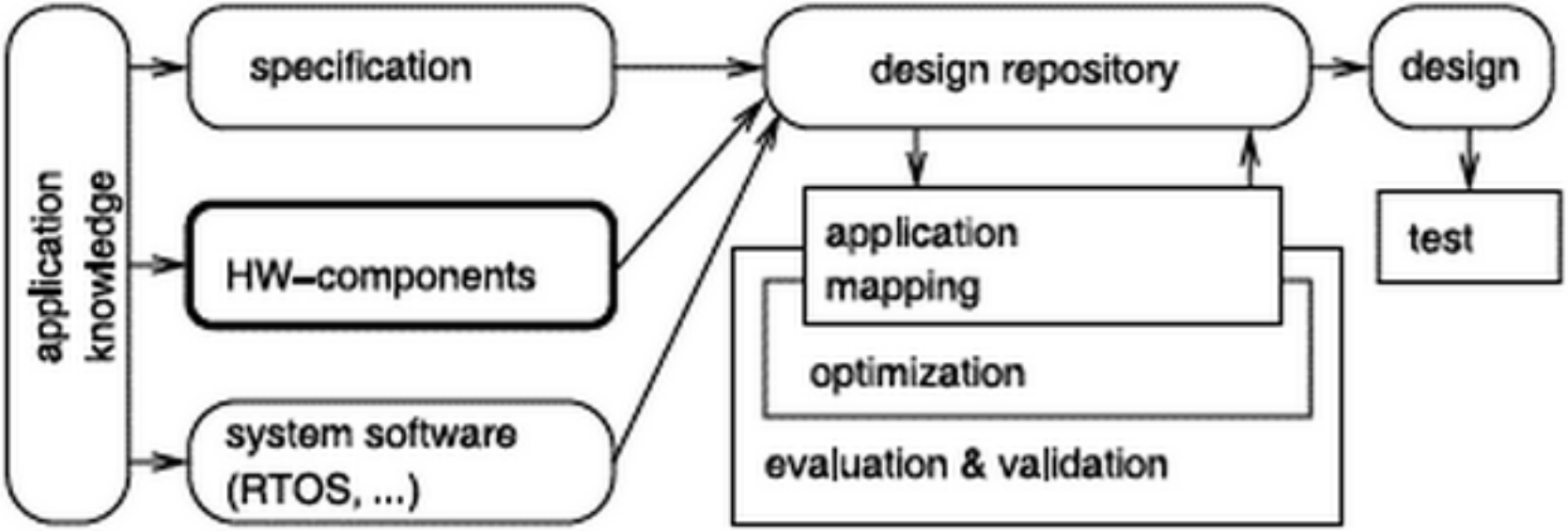
- **Partitioning** identifies the parts in the model which are best suited in hardware software given constraints such as time, size, cost and power. Recent reports indicate that automatic partitioning is currently making little headway, and that researchers are turning to semiautomatic "design space exploration," relying on tools for fast evaluation of user-directed partitions.

- **Co-synthesis** uses the available tools to synthesize the software, hardware and interface implementation. This is done concurrently with as much interaction as possible between the three implementations. Hardware synthesis is built on existing CAD tools, typically via VHDL or Verilog. Software synthesis is usually in any high level language. Codesign tools should generate hardware/software interprocess communication automatically, and schedule software processes to meet timing constraints (see also the diagram at the top of the page). DSP software is a particular challenge, since few good compilers exist for these idiosyncratic architectures. Retargetable compilers for DSPs, ASIPs (application specific instruction processors), and processor cores are a special research problem.

- **Co-simulation** executes all three components, functioning together in real time. This phase helps with the verification of the original design and implied constraints by verifying if input and output data are presented as expected.

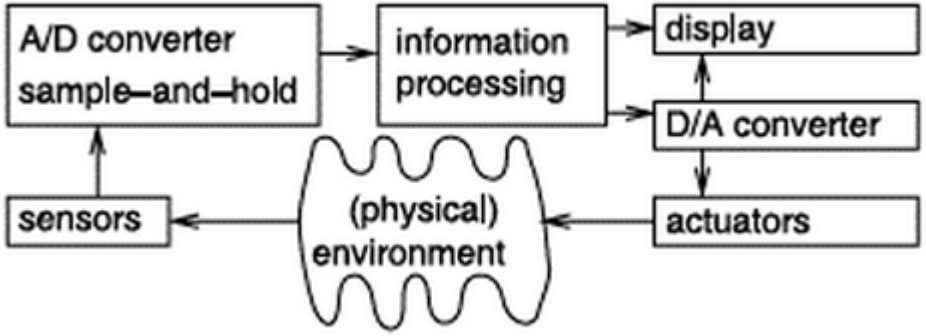
Diseño de un sistema empotrado (III)

- Otra forma de verlo (Marwedel, Embedded System Design)



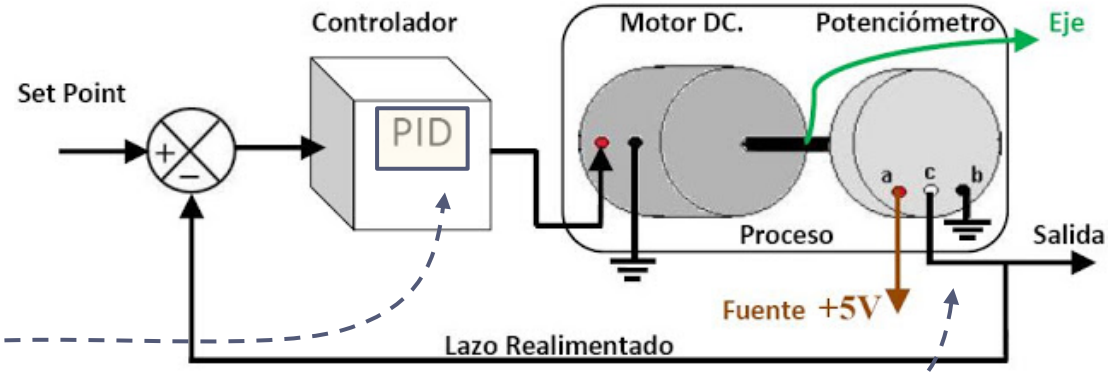
Sistema empotrado: componentes

- En general, el sistema empotrado se usa como controlador de un bucle:

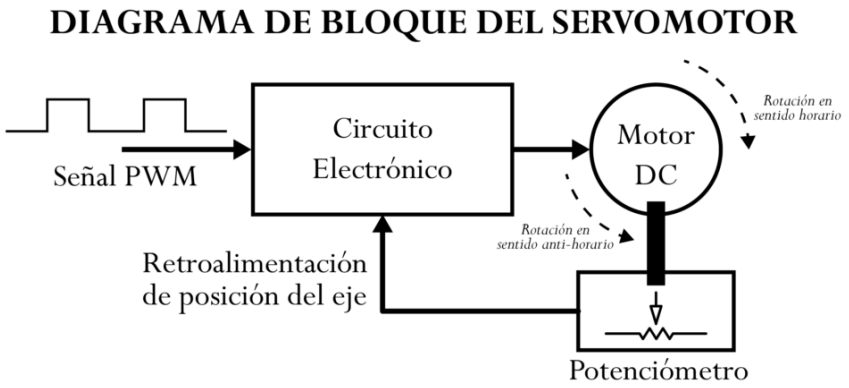


- Componentes:
  - Entradas. Sensores, Circuitos Sample-&-Hold, Conversores A/D …
  - Salidas. Displays, Conversores D/A, Actuadores (p.e. electro-mecánicos)
  - Comunicaciones. Requisitos: Eficiencia, mantenibilidad, robustez ante ruidos, …
  - Unidades de proceso. ASICs, Procesadores, DSPs, Multimedia, VLIW, Microcontroladores, lógica reconfigurable, …
  - Memorias

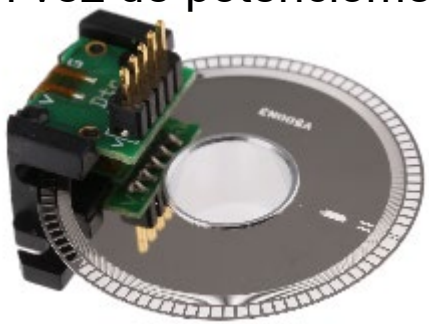
Ejemplo de uso: control de motor DC



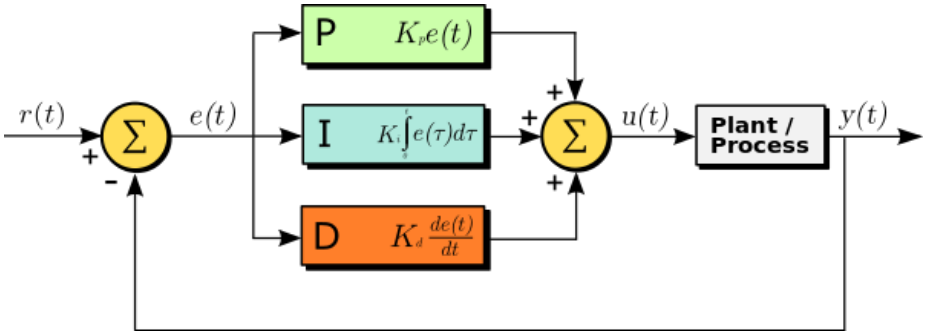
Control PID, un ejemplo:
https://www.youtube.com/watch?v=HRJiow_k-V0

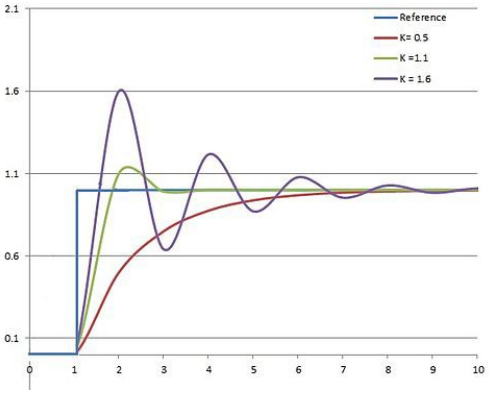


Codificador óptico
(en vez de potenciómetro)

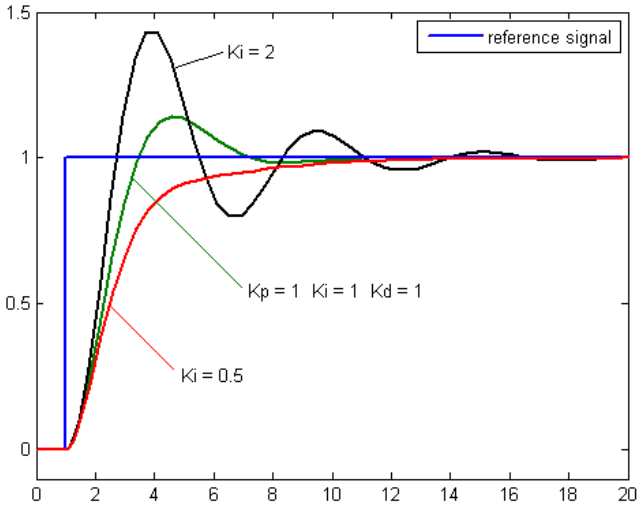


Control PID

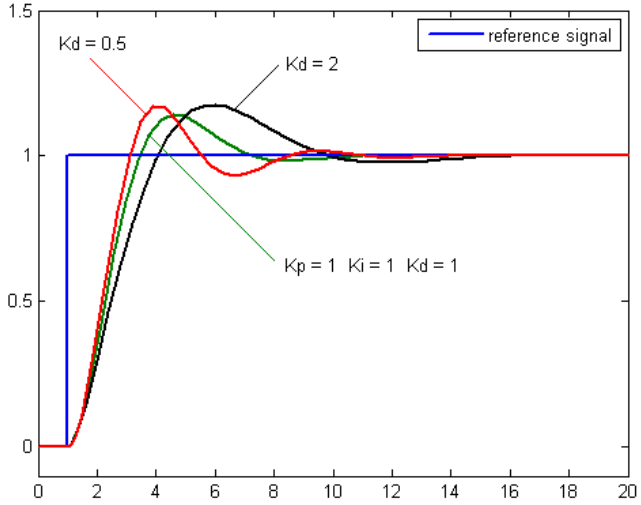




Response of PV to step change of SP vs time, for three values of Kp (Ki and Kd held constant)



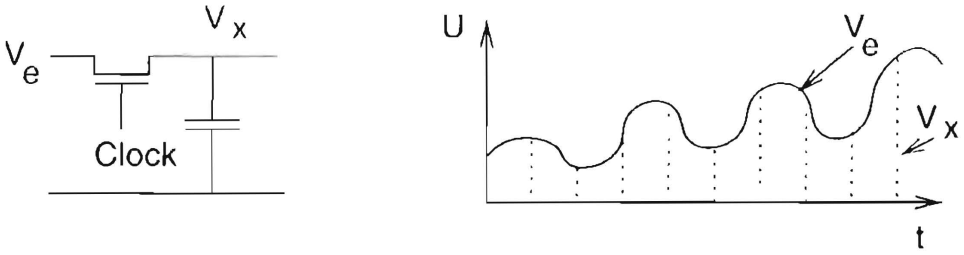
Response of PV to step change of SP vs time, for three values of K_i (K_p and K_d held constant)



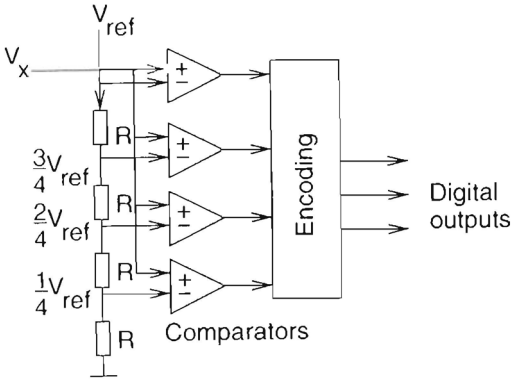
Response of PV to step change of SP vs time, for three values of K_d (K_p and K_i held constant)

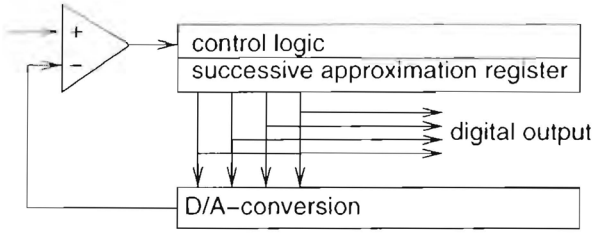
Sistema empotrado: Entradas.

- Sensores:
  - Prácticamente para cada magnitud física, p.e. Peso, Velocidad, Aceleración, Temperatura, Voltaje, Corriente eléctrica, …, detectores de sustancias químicas, humedad, imagen, biométricos, biomédicos, …
- Circuitos Sample-&-Hold



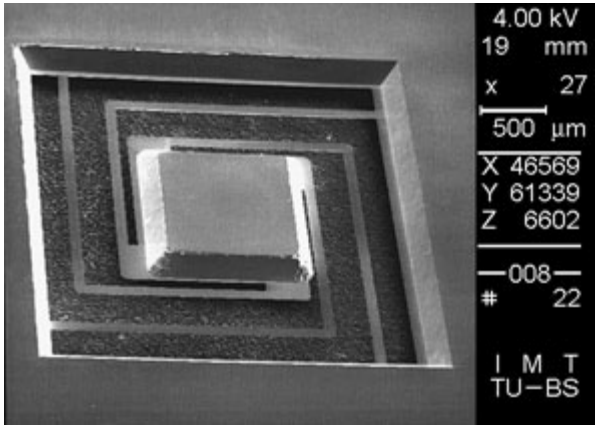
- Conversores A/D
  a) Flash
  b) Aproximaciones sucesivas





Sensores

- Prácticamente cualquier magnitud física. Ejemplos:
- Aceleración (pequeña masa, mueve los cables conductores que la rodean, cambia su resistencia).



- Lluvia (vehículos convencionales, diodos).
- Imagen: CCD (mejor calidad de imagen) / CMOS (más económicos, permiten ser integrados junto con circuitería lógica, menor consumo).
- Biometría (huellas, iris, cara, retina…).
- Ojos artificiales (cámara más interface eléctrico/sonoro).
- RFID (Radio Frequency Identification). Identificación de objetos, animales, personas. (tarjeta de crédito, tarjeta de transportes, etc).
- Otros: presión, proximidad, control de motores (cod. óptico), efecto Hall, etc.

(Efecto Hall https://short.upm.es/3uob3)

Sensores (características)

- Miden alguna magnitud física. Pueden estar empaquetados junto con procesadores, conversores A/D o D/A, constituyendo módulos "inteligentes" que forman parte del IoT.
- Características:
  - Magnitud → medida: x(t) → f(x(t)) frecuentemente función afín: $f(x(t)) = A \cdot x(t) + B$
  - Sensibilidad. Constante de proporcionalidad "A"
  - Rango operativo: Normalmente los valores a medir son limitados entre L y H, p.e. temperatura de -25°C a +75°C (función afín con límites)
  - Precisión *p*: diferencia más pequeña en valor absoluto que es capaz de distinguir (p.e. 0,01°C)
  - Rango dinámico: D=(H-L)/*p* normalmente en decibelios: $D_{dB}=20 \log_{10}(H-L)/p$ *(p.e. 80dB)*
  - Quantization: sensor digital mide con n-bits, solo distingue 2^n medidas. Idealmente coincide con la precisión (dos medidas con valores consecutivos difieren en 1 unidad)
  - Noise: diferencia entre la magnitud observada (medida) y el valor real de la misma. El ruido RMS (root mean square) es el valor cuadrático medio del ruido en todo el rango operativo.
  - Signal to Noise ratio: Si X es el valor cuadrático medio de la señal y *N* el del ruido, se mide en decibelios y es: $SNR=20 \log_{10} X/N$
  - Sampling: cuando hay muestreo, la señal se obtiene a intervalos normalmente regulares (periodo de muestreo)

Sensores (otras características)

Distorsión Armónica (ejemplo)

- Se da cuando *un sensor se comporta dentro de su rango operativo de forma no-lineal*
- Por ejemplo, que la sensibilidad del sensor (o actuador) dependa de la magnitud de la señal
- Un ejemplo más concreto: un micrófono que responda con mayor sensibilidad ante medidas de menor presión (señal de audio menos intensa) que ante otras más intensas
- En lugar de $f(x(t))=A \cdot x(t) + B$, que se comporte como $f(x(t))=A \cdot x(t) + B + C \cdot (x(t))^2$
- Supongamos que se reproduce una señal sinusoidal pura frente al micrófono:

$x(t) = \cos(\omega_0 t)$ donde t es el tiempo y ω_0 la frecuencia en radianes por segundo

Si el sensor se comporta como $f(x(t))=A \cdot x(t) + B + C \cdot (x(t))^2$ en el instante t medirá:

$$x'(t) = A \cdot x(t) + B + C \cdot (x(t))^2 = A \cdot \cos(\omega_0 t) + B + C \cdot \cos^2(\omega_0 t)$$

Y, dado que $\cos^2(\varphi) = \frac{1}{2}(1 + \cos(2\varphi))$ resulta:

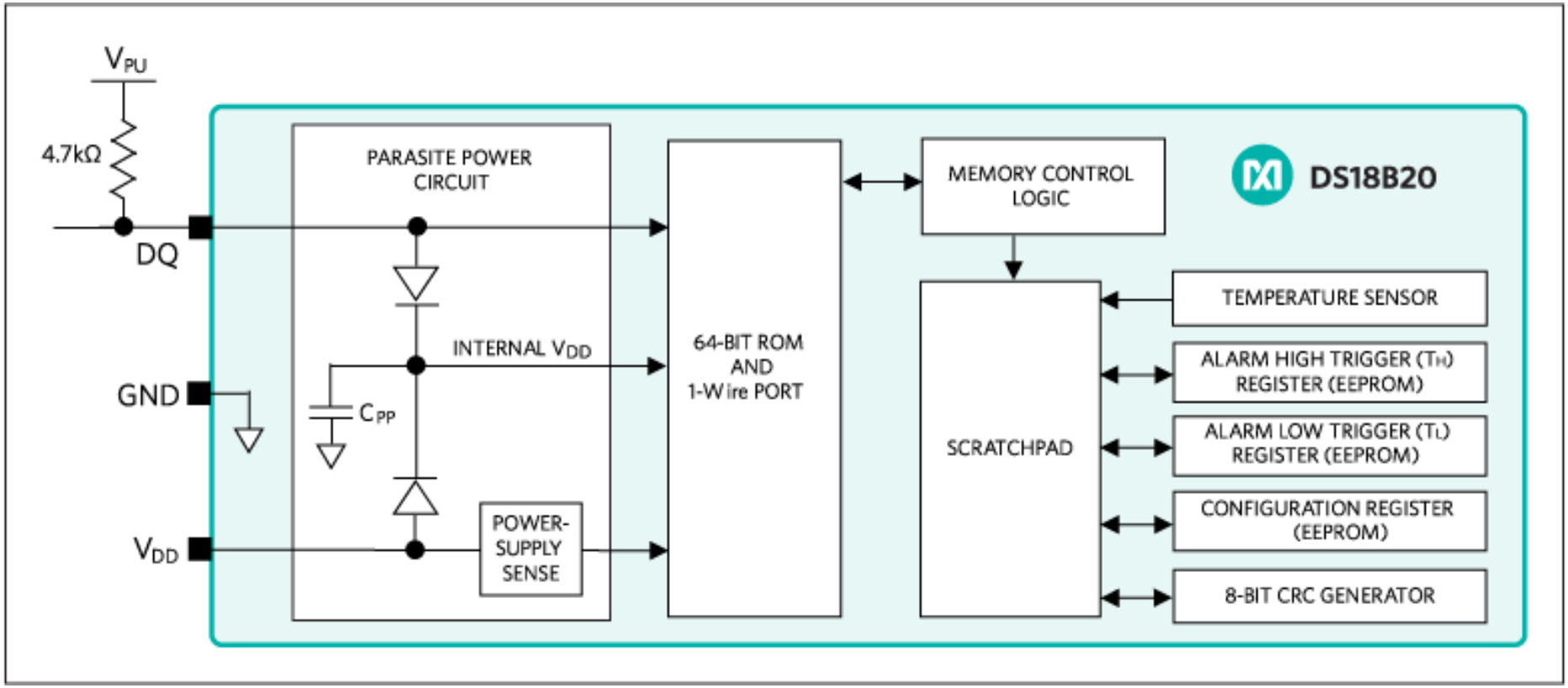
$x'(t) = A \cdot \cos(\omega_0 t) + B + \frac{1}{2}(C + C \cdot \cos(2\omega_0 t))$, que para el oído humano equivale a:

$$x'(t) = A \cdot \cos(\omega_0 t) + C/2 \cdot \cos(2\omega_0 t)$$

(se registra una señal que tiene cierta componente de frecuencia doble a la real)

Sensores (ejemplo: temperatura, DS18B20)

- Sensor "inteligente": genera temperatura en digital, dispara alarmas, programable



Sensores (ejemplo: temperatura, DS18B20)

Key Features

Unique 1-Wire® Interface Requires Only One Port Pin for Communication

Reduce Component Count with Integrated Temperature Sensor and EEPROM

- Measures Temperatures from -55°C to +125°C (-67°F to +257°F)
- ±0.5°C Accuracy from -10°C to +85°C
- Programmable Resolution from 9 Bits to 12 Bits
- No External Components Required

Parasitic Power Mode Requires Only 2 Pins for Operation (DQ and GND)

Simplifies Distributed Temperature-Sensing Applications with Multidrop Capability

- Each Device Has a Unique 64-Bit Serial Code Stored in On-Board ROM

Flexible User-Definable Nonvolatile (NV) Alarm Settings with Alarm Search Command Identifies Devices with Temperatures Outside Programmed Limits

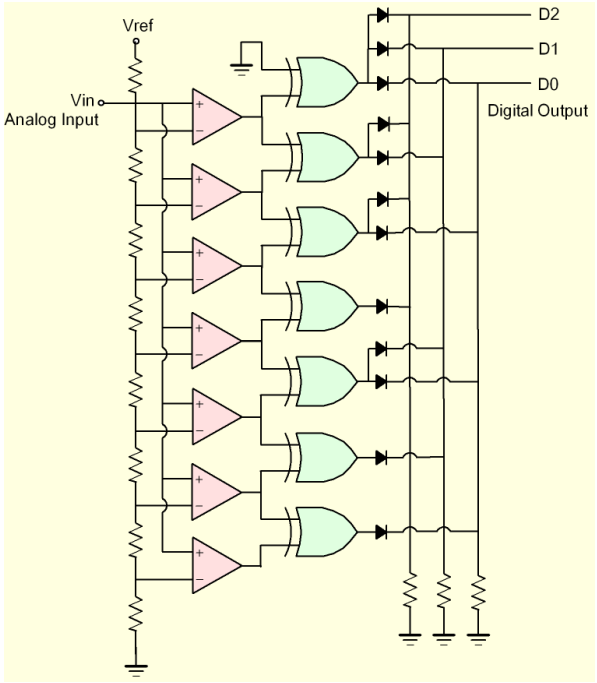
Available in 8-Pin SO (150 mils), 8-Pin µSOP, and 3-Pin TO-92 Packages

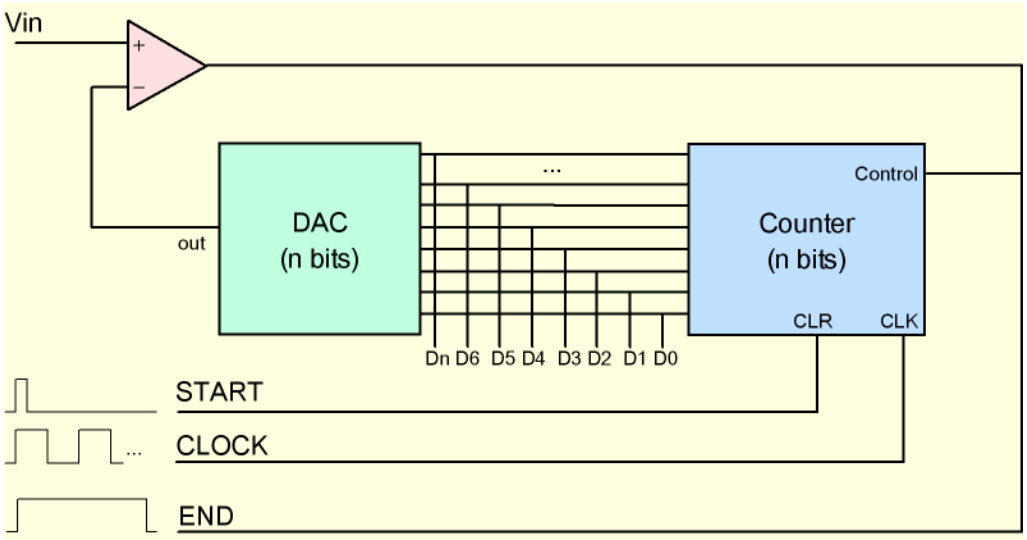
Conversores A/D

- Realizan la conversión de una señal analógica a digital. Varias técnicas.

Flash converters: rápidos, costosos (paralelos). 2^n -1 comparadores.
p.e. 12 bits → 4095 comparadores.
¿ Precisión de las resistencias ?

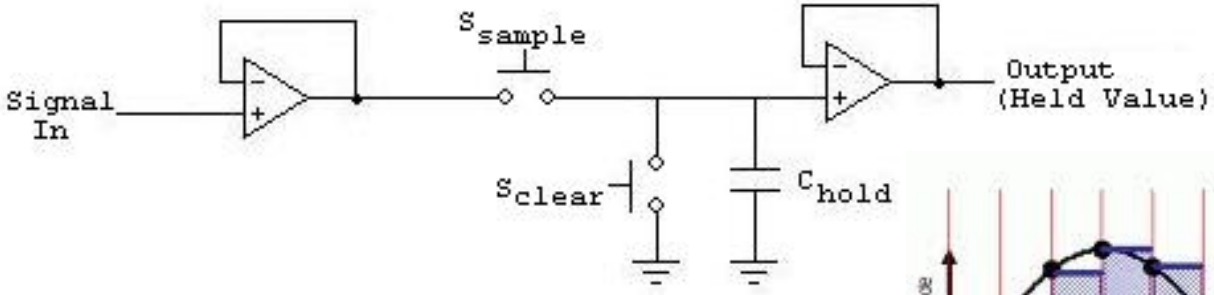
Aproximaciones sucesivas:
Basados en conversión D/A, más lento (búsqueda binaria)

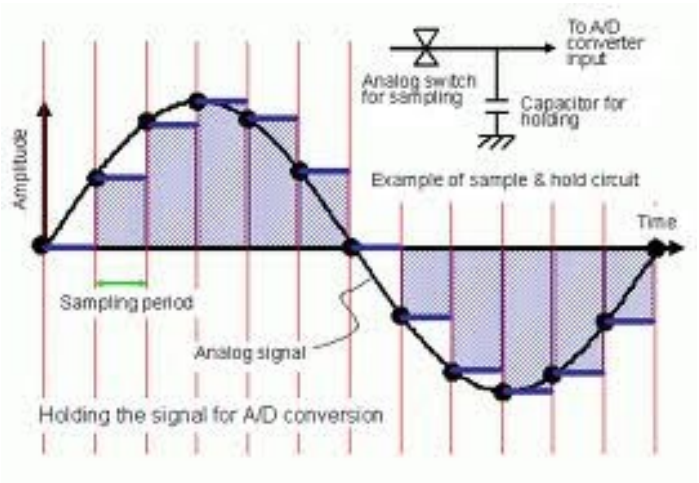




Circuitos Sample & Hold

- Evitan variaciones en la señal de entrada de los conversores A/D que podrían corromper la medida.
- Son "memorias analógicas", recuerdan el nivel de una señal durante un tiempo determinado.
- Permiten **conectar varias entradas** analógicas a un único conversor A/D.
- Permiten construir conversores A/D por aproximaciones sucesivas.





Sistema empotrado: Salidas (I)

- Displays

Área muy importante para los sistemas empotrados. Por ejemplo (pero no limitado a) en:

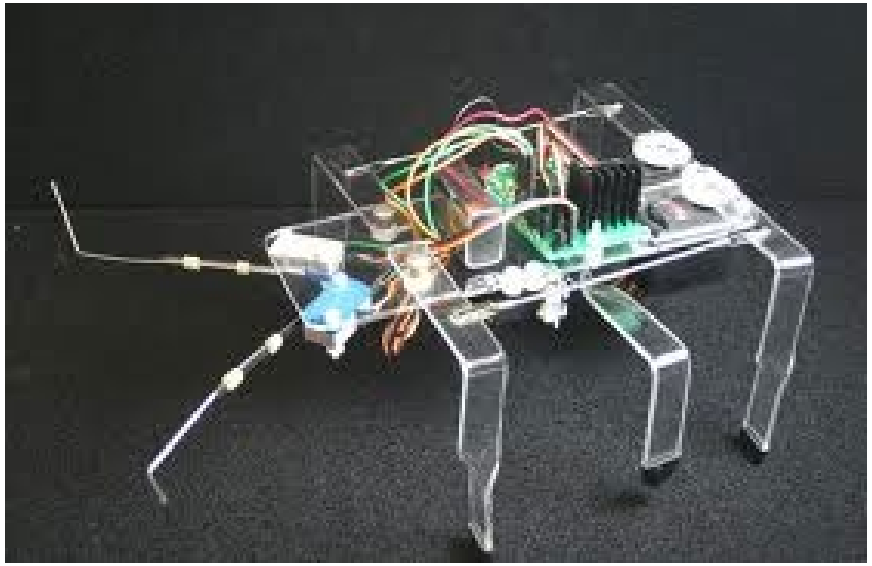
- Teléfonos móviles, Navegadores GPS, Instrumentación de automóviles (cuadro de instrumentos, radio-CD, …), Instrumentación de laboratorio, Relojes inteligentes, etc…

Diferentes tipos de dispositivos:

- Cristal líquido (LCD, https://en.wikipedia.org/wiki/Liquid-crystal_display)
- Organic displays (OLED), Active-Matrix OLED (AMOLED) … (https://en.wikipedia.org/wiki/OLED)

Sistema empotrado: Salidas (II)

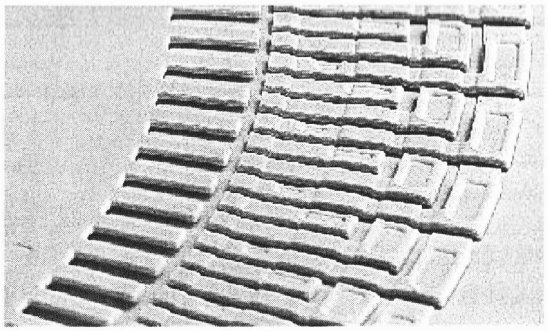
- Dispositivos Electro-mecánicos
  - Robótica, Automatización, Modelismo, etc.



Sistema empotrado: Salidas (III)

- Actuadores, mecatrónica: une ingeniería *mecánica*, ingeniería *electrónica*, ingeniería *de control* e ingeniería *informática*

https://www.sensorsmag.com/components/introduction-to-piezoelectric-motors



- Motores *piezoeléctricos*, de ultrasonidos: p.e. autofocus de cámara, instrumentación médica para odontología, trabajos bajo RMN:

https://www.micromo.com/applications/medical-lab-automation-equipment/piezo-motors-power-mri-robot

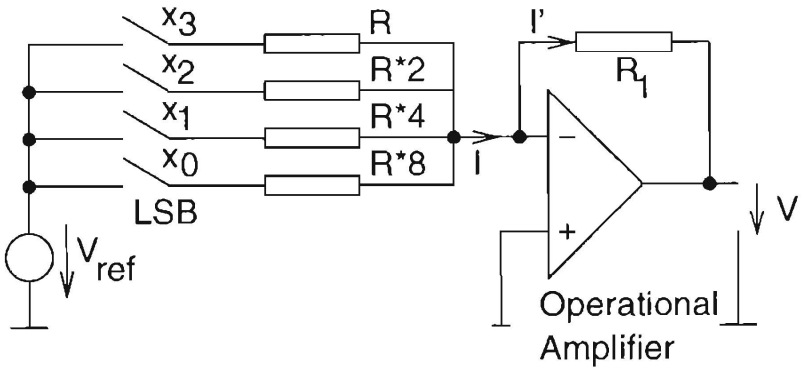
- No se ven afectados por campos magnéticos. Pequeños, par motor elevado.

https://www.piezomotor.com/applications/

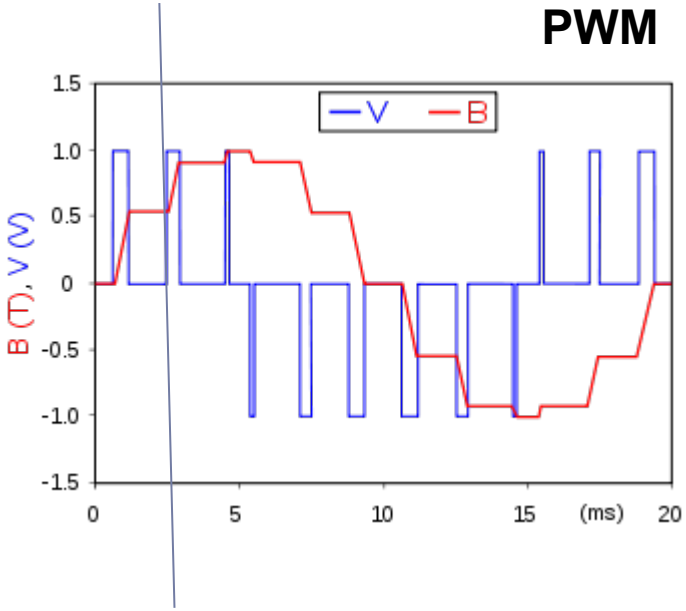
Sistema empotrado: Conversores D/A

- Realizan la conversión de una señal digital, binaria, a analógica. Varias técnicas.
  - Redes de N resistencias o condensadores preajustados *(quantization 2^N medidas)*
  - Modulación de ancho de pulsos (ajusta el *duty-cycle*)
  - Conversión de 1-bit mediante la modulación *sigma-delta*

Red de resistencias

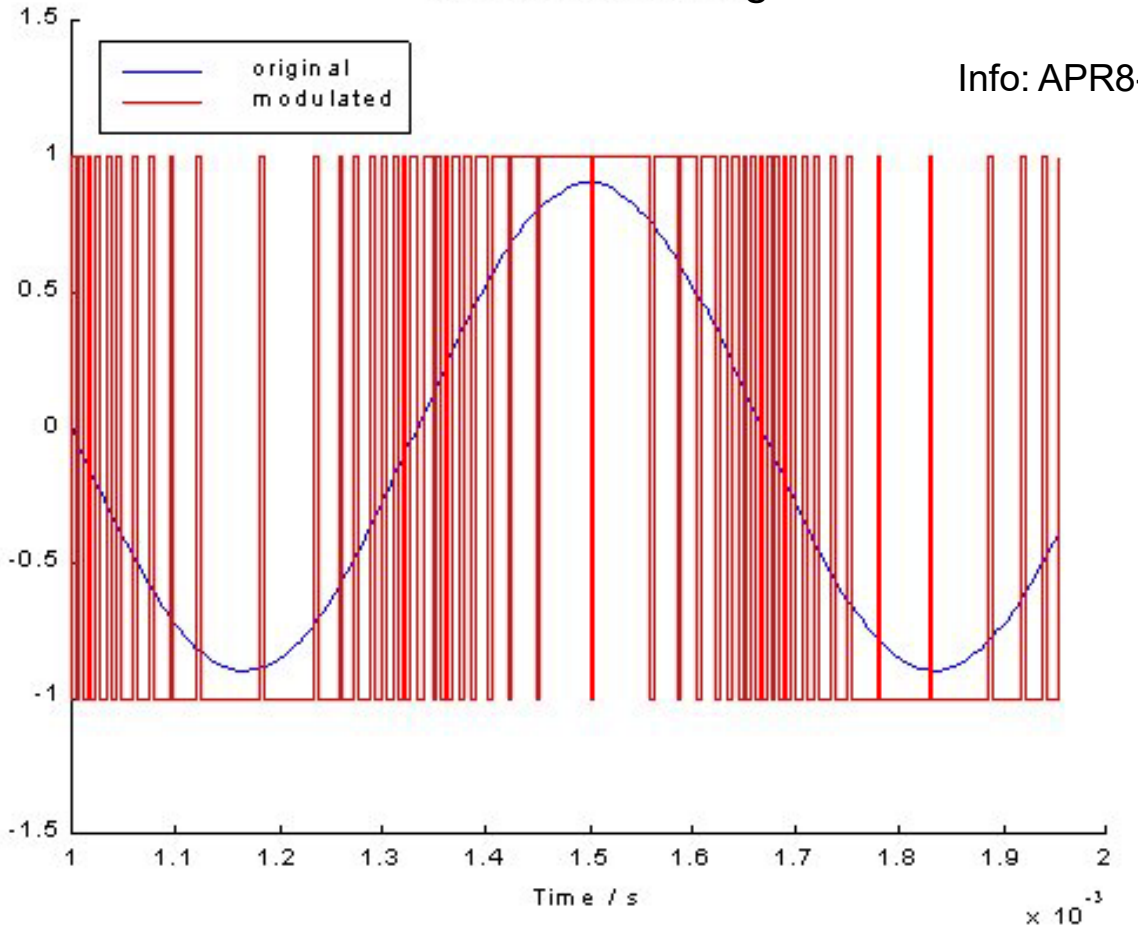


PWM



S. empotrado: Conversión Sigma-Delta

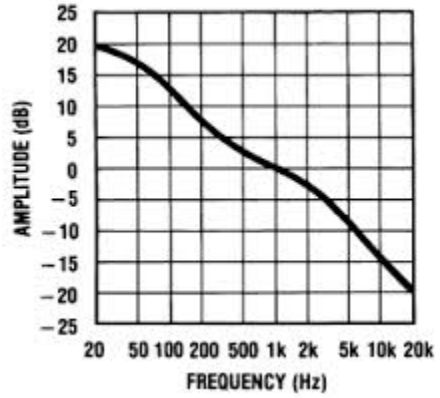
- Realizan la conversión de una señal digital, binaria, a analógica. Varias técnicas.
  - Conversión de 1-bit mediante la modulación *sigma-delta*



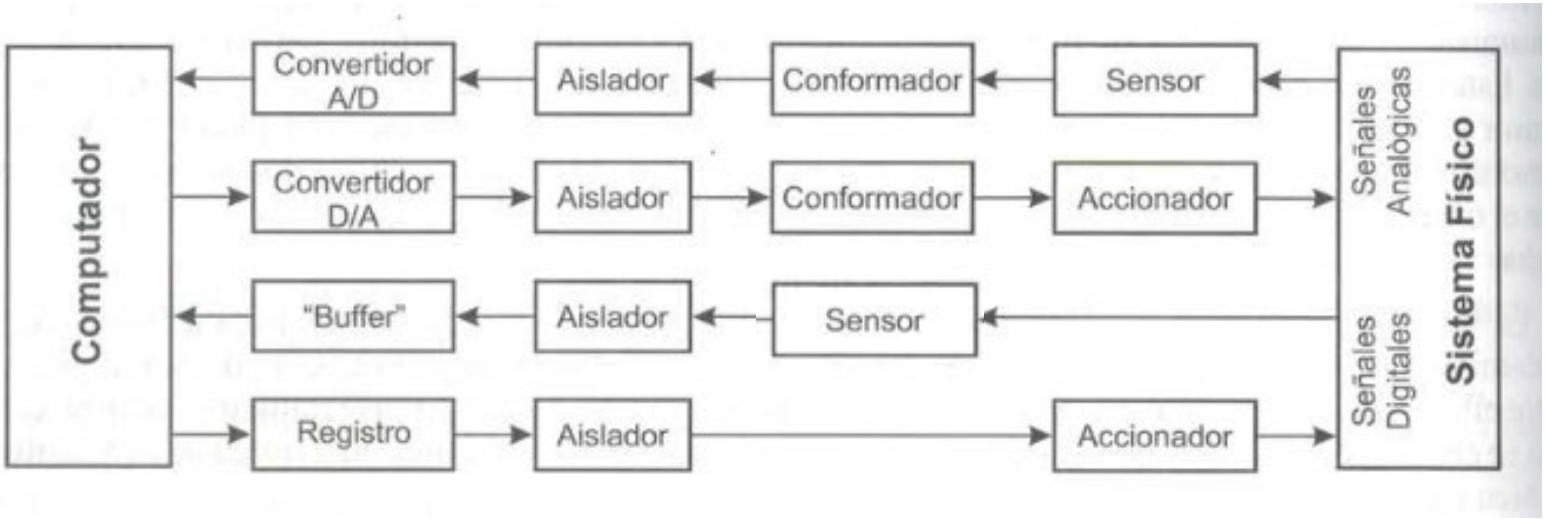
Info: APR8-sigma-delta.pdf

Periféricos analógicos/de control (I)

- Cuatro tipos de periféricos de control:
  - Lectura analógica
  - Accionamiento analógico
  - Lectura digital
  - Accionamiento digital



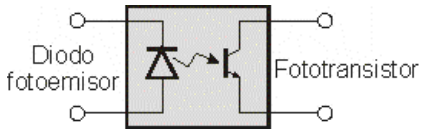
RIAA
(ejemplo de conformador)



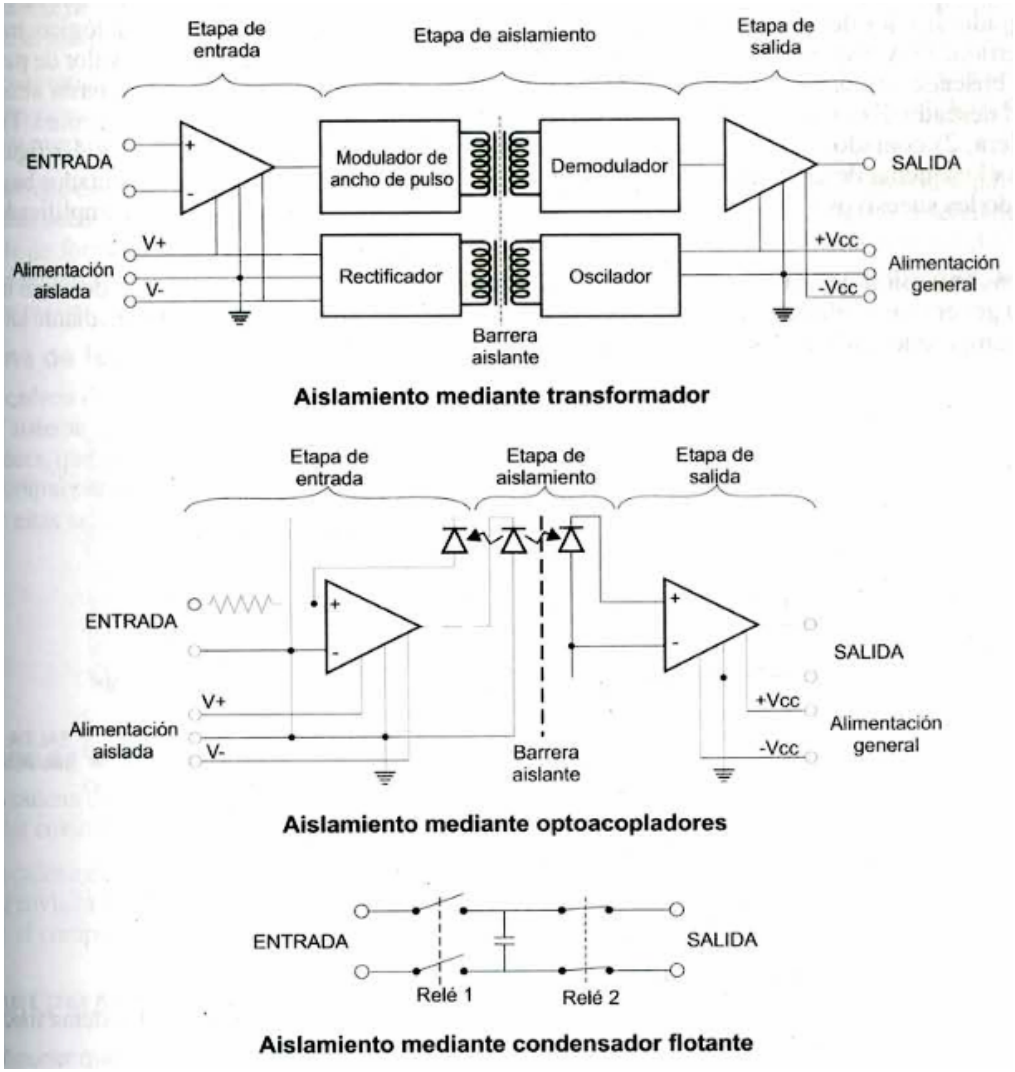
(Fundamentos de los Computadores, Pedro de Miguel, Paraninfo 2004)

Periféricos analógicos/de control (II)

- Etapa de aislamiento:

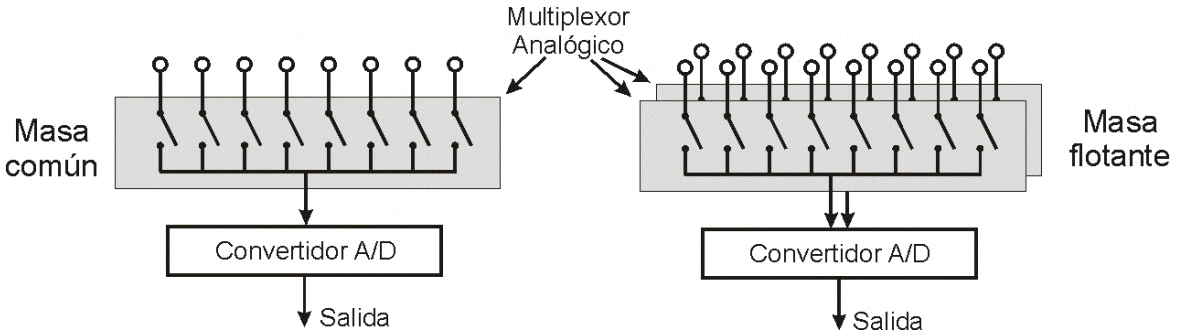


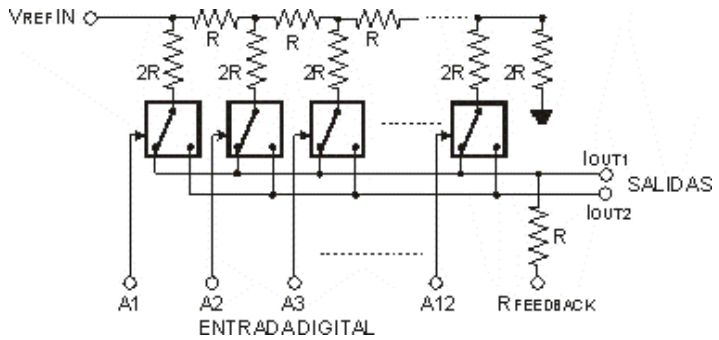
(para señal digital)



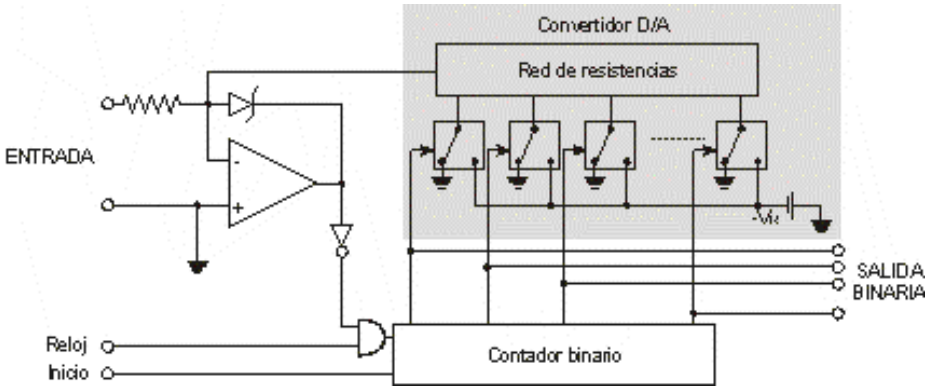
Periféricos analógicos/de control (III)

- Manejo de señales analógicas:





Conversor digital-analógico

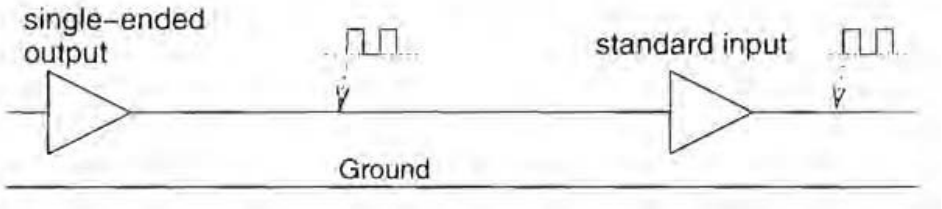


Conversor analógico-digital (aprox. sucesivas)

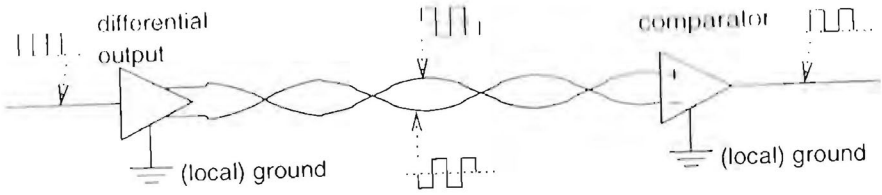
Sistema empotrado: Comunicaciones

- Canales: entidades abstractas, caracterizadas por las propiedades de la comunicación, como *ancho de banda* o *inmunidad al ruido*.
- Medio de transmisión: soporte físico de la comunicación, como cable, radio (wi-fi, bluetooth), medio óptico (infrarrojos, fibra óptica).
- Múltiples requisitos, p.e. Eficiencia, Mantenibilidad, Robustez, Capacidad de trabajar bajo condiciones de Tiempo Real y/o Tolerancia a fallos …
- Robustez ante ruidos: cableado simple/diferencial

a) Cableado simple



b) cableado diferencial



Sistema empotrado: Comunicaciones

- Buses de sensores/actuadores: Comunicación entre dispositivos simples, como pulsadores o iluminadores, con el equipo de proceso. Pueden ser buses serie como RS232 o bucle de corriente 4-20mA (RS485)
- Buses de campo (field busses): Con el mismo propósito de los buses de sensores y actuadores pero estructurados para obtener mayores prestaciones. P.e. CAN-Bus (cars), MAP (car factories), EIB-European Installation Bus (smart buildings).

"Bus de campo" es el nombre de una familia de protocolos de red industriales utilizados para **control distribuido en tiempo real**, estandarizado como IEC 61158. Un sistema industrial complejo y automatizado — como las líneas de ensamblaje — usualmente requiere de un sistema de control distribuido — una **organización jerárquica de controladores** — para funcionar. En el nivel superior de esta jerarquía existe comúnmente una **interfaz hombre-máquina (HMI)**, desde la cual el operador puede monitorizar u operar el sistema. Ésta está típicamente ligada a una capa en los **niveles medios** de la jerarquía, constituida por **controladores lógicos programables (PLC)** a través de un **sistema de comunicación** que no es de tiempo crítico (por ejemplo, Ethernet). En el **nivel más bajo** de la cadena de control está el **bus de campo que enlaza los PLCs a los componentes que realizan la tarea**, como sensores, actuadores, motores eléctricos, luces de consola, interruptores, válvulas y contactores. (wikipedia).

Sistema empotrado: U. de Proceso.

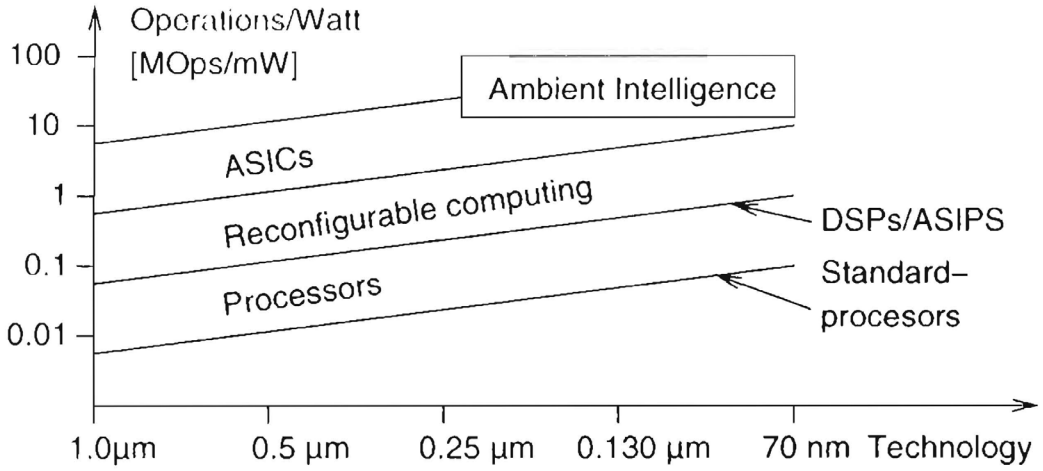
- Diferentes alternativas: de lo específico a lo general… Eficiencia.

- ASICs: "custom hardware"

  Desarrollo máscara[1] > $1.000.000

- Lógica reconfigurable: FPGA

  Excelente relación potencia/coste



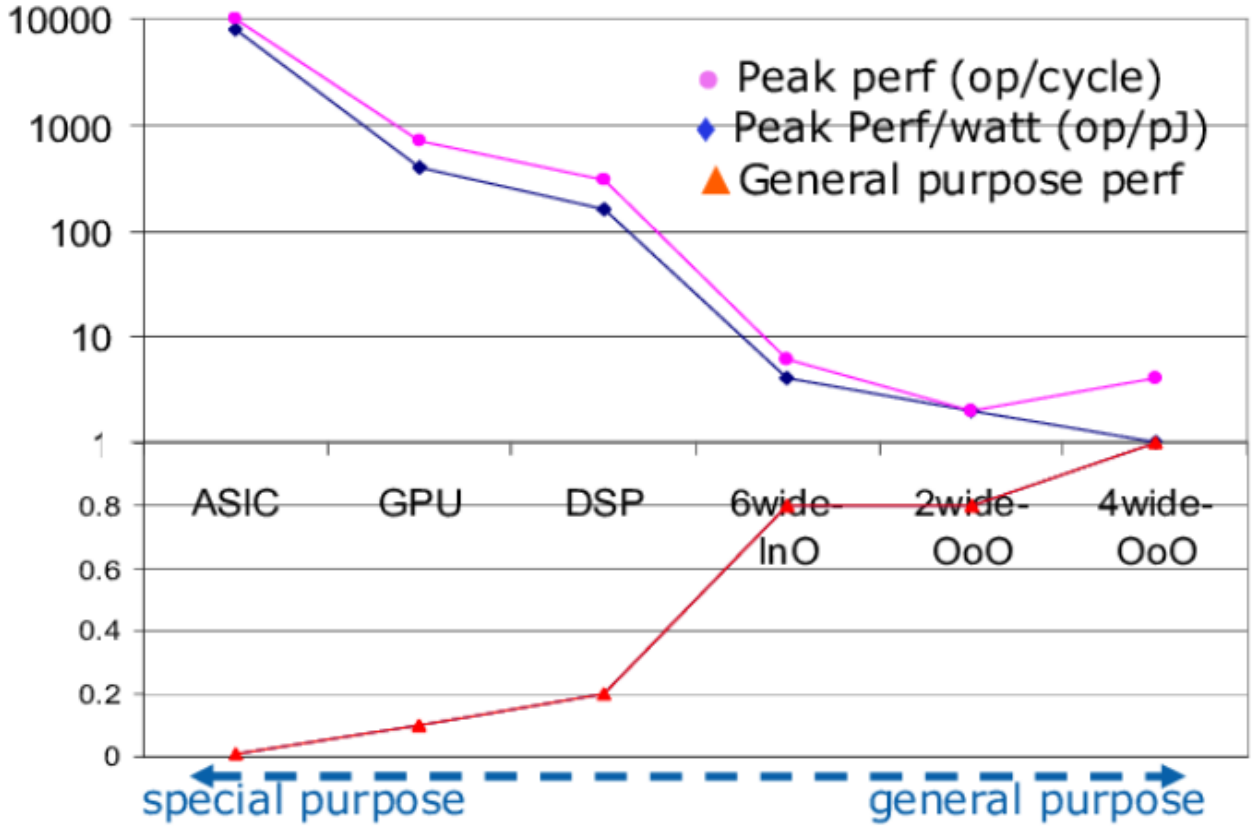
- Procesadores de propósito general: CPU/DSP

  Gran flexibilidad. Múltiples posibilidades: Procesadores 8-32 bits, Microcontroladores, DSPs, VLIW, Procesadores multimedia, etc.

https://electronics.stackexchange.com/questions/7042/how-much-does-it-cost-to-have-a-custom-asic-made

https://europractice-ic.com/schedules-prices-2022/

Sistema empotrado: U. de Proceso.

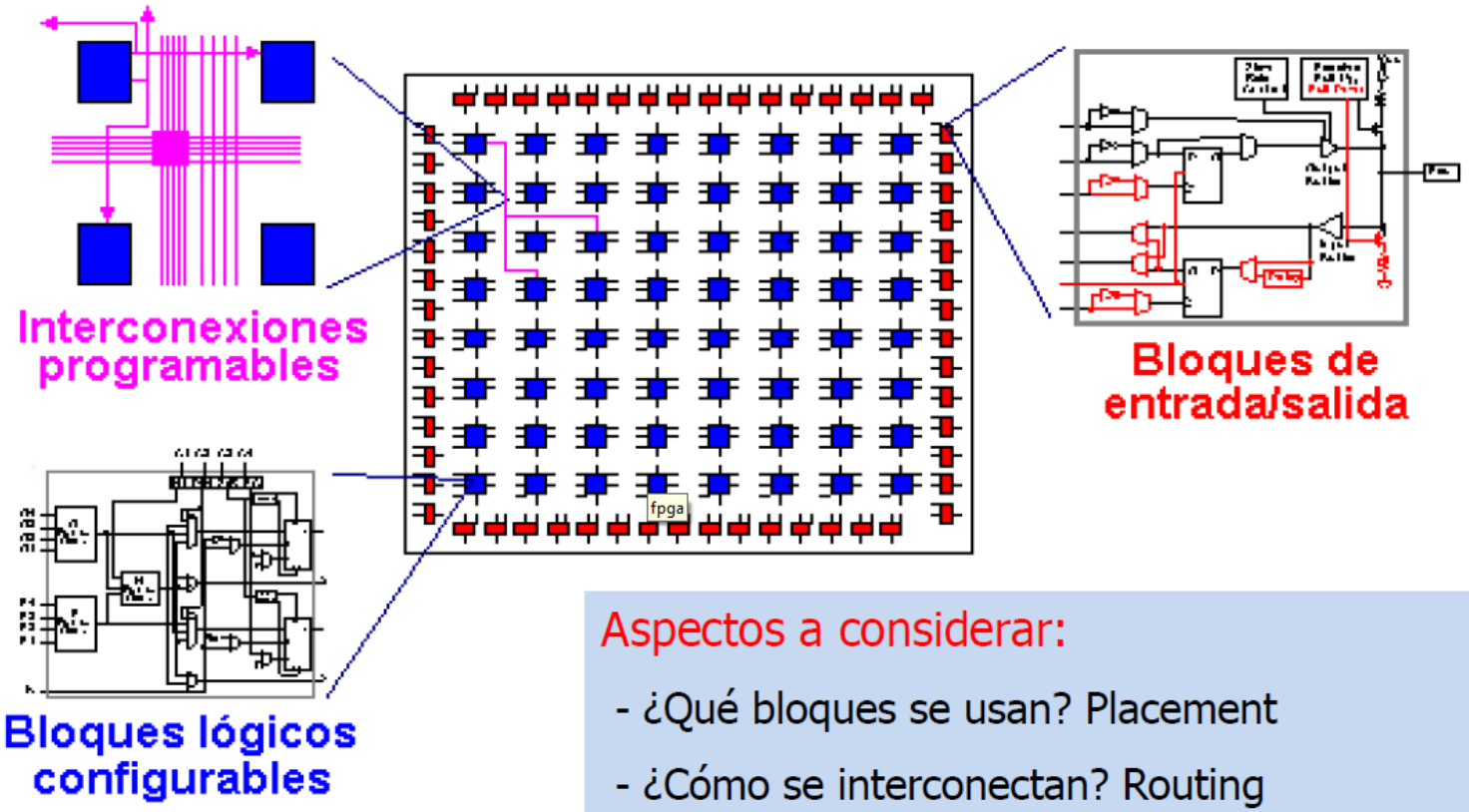


Performance-per-watt, Peak Performance and General-purpose Performance of General Purpose Processor vs. Others

Paper: “A HW/SW co-designed heterogeneous multi-core virtual machine for energy-efficient general purpose computing”

Sistema empotrado: FPGA

Puede incluir sistemas completos heterogéneos en un único chip (SoC)



Aspectos a considerar:

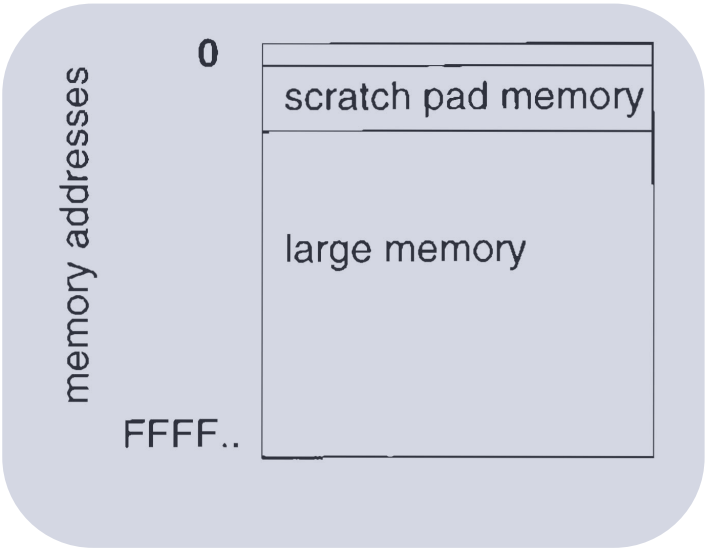
- ¿Qué bloques se usan? Placement
- ¿Cómo se interconectan? Routing
- ¿Cómo se programan? Bitstream generation

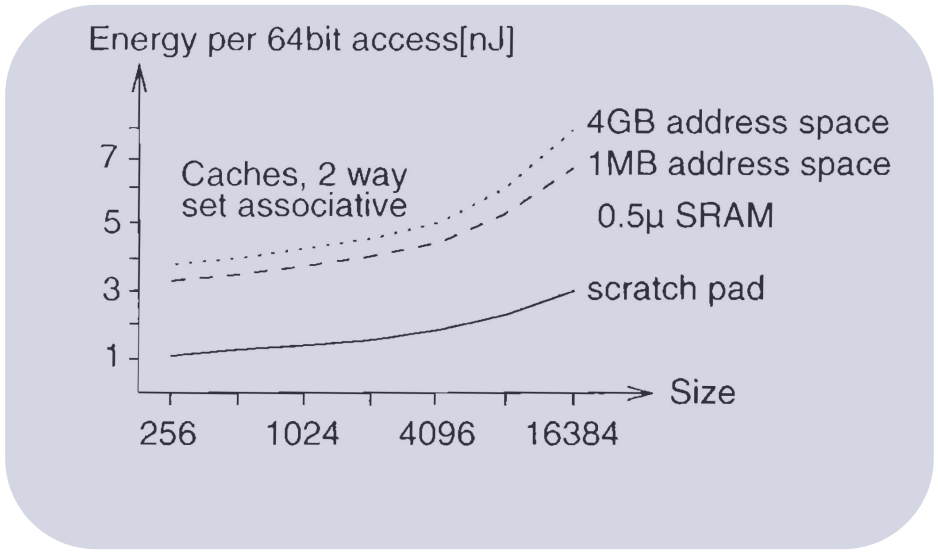
(V. Rodellar, Informática Industrial, GII)

Sistema empotrado: Memorias.

Almacenamiento de datos, programas, configuraciones en: CPU / ASICs / FPGA

- Eficiencia en cuanto a:
  - Tiempo de ejecución: ¿ influencia de la memoria ?
  - Tamaño del código: ¿ dispositivos de memoria / compilador / ISA ?
  - Consumo de energía: ¿ jerarquía de memoria / tamaño ?





Sistema empotrado: Memorias.

"Energy Conservation in Memory Hierarchies using Power-Aware Cached-DRAM"
Nevine AbouGhazaleh, Bruce Childers, Daniel Moss´e, and Rami Melhem
(http://people.cs.pitt.edu/~childers/papers/dagstuhl05-pacdram.pdf)

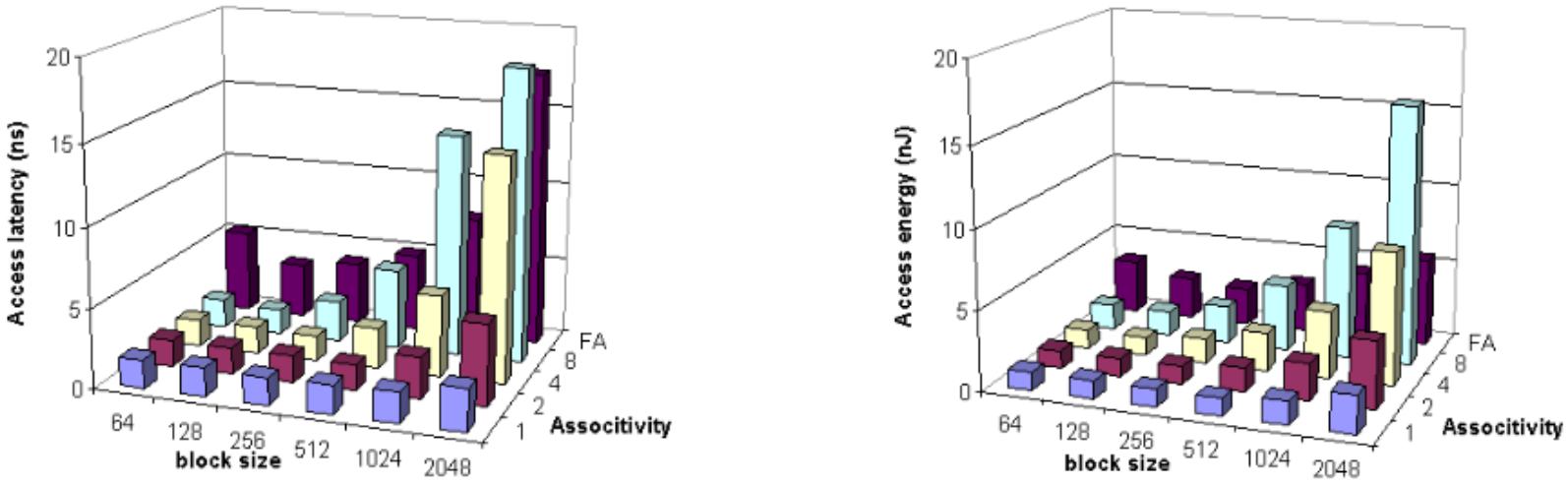


Fig. 3. The effect of varying the cache block size and associativity on the cache access latency and energy consumption for a 256KB on-memory cache.

Ejemplos:

- Robot escalador (A tailless timing belt climbing platform utilizing dry adhesives with mushroom caps)
  https://www.youtube.com/watch?v=tont-BzM1II
- Robot "Sand Flea" (Boston Dynamics: Changing Your Idea of What Robots Can Do)
  http://www.youtube.com/watch?v=6b4ZZQkcNEo
- "STARMAC" (The Stanford/Berkeley Testbed of Autonomous Rotorcraft for Multi-Agent Control)
  https://www.youtube.com/watch?v=rJ9r2orcaYo
- MIT Robotic Cheetah (MIT Biomimetic Robotics Laboratory)
  https://youtu.be/XMKQbqnXXhQ http://newsoffice.mit.edu/2014/mit-cheetah-robot-runs-jumps-0915
- Boston Dynamics

(Atlas ,The World's Most Dynamic Humanoid –28 articulaciones–)
https://www.youtube.com/watch?v=_sBBaNYex3E

(Spot, dispositivo commercial ~ $75.000 2021)
https://www.bostondynamics.com/products/spot

Índice

- Introducción / HW para sistemas empotrados
- Características y componentes de un sistema empotrado
- Procesadores:
  - CPU / FPGA / DSP / VLIW
- Computadores modulares
- Problemas/Ideas/Técnicas utilizadas en s. empotrados

- Normativa (Actualmente en grado en II: 'Proyecto de Instalación Informática')

Procesador

- Componente del computador que ejecuta las instrucciones del programa.
- Manipula y mueve los datos.
  - Inicialmente implementado en placas (múltiples chips)
  - Actualmente un solo chip que integra otros componentes (Memorias caché, MMU, controlador gráfico, etc.)
  - Se caracteriza por:
    - Ancho de palabra (4, 8, 16, 32, 64 bits, … 1024 bits)
    - Velocidad de reloj (pocos MHz hasta 5 GHz)
    - Juego de instrucciones: CISC / RISC / VLIW / DSP
    - Implementación: "Convencional" / Segmentada / Superescalar

Procesador: CISC / RISC / VLIW / DSP

CISC

Énfasis en el hardware

Utiliza instrucciones complejas,
(la mayoría multiciclo)

Arquitectura **memoria-memoria**:
Carga y almacenamiento pueden
formar parte de otras instrucciones

Tamaño de código reducido,
baja velocidad de reloj

Unidad de control compleja.
Área de silicio importante

RISC

Énfasis en el software

Define/emplea instrucciones simples uniciclo
(implementa también multiciclo)

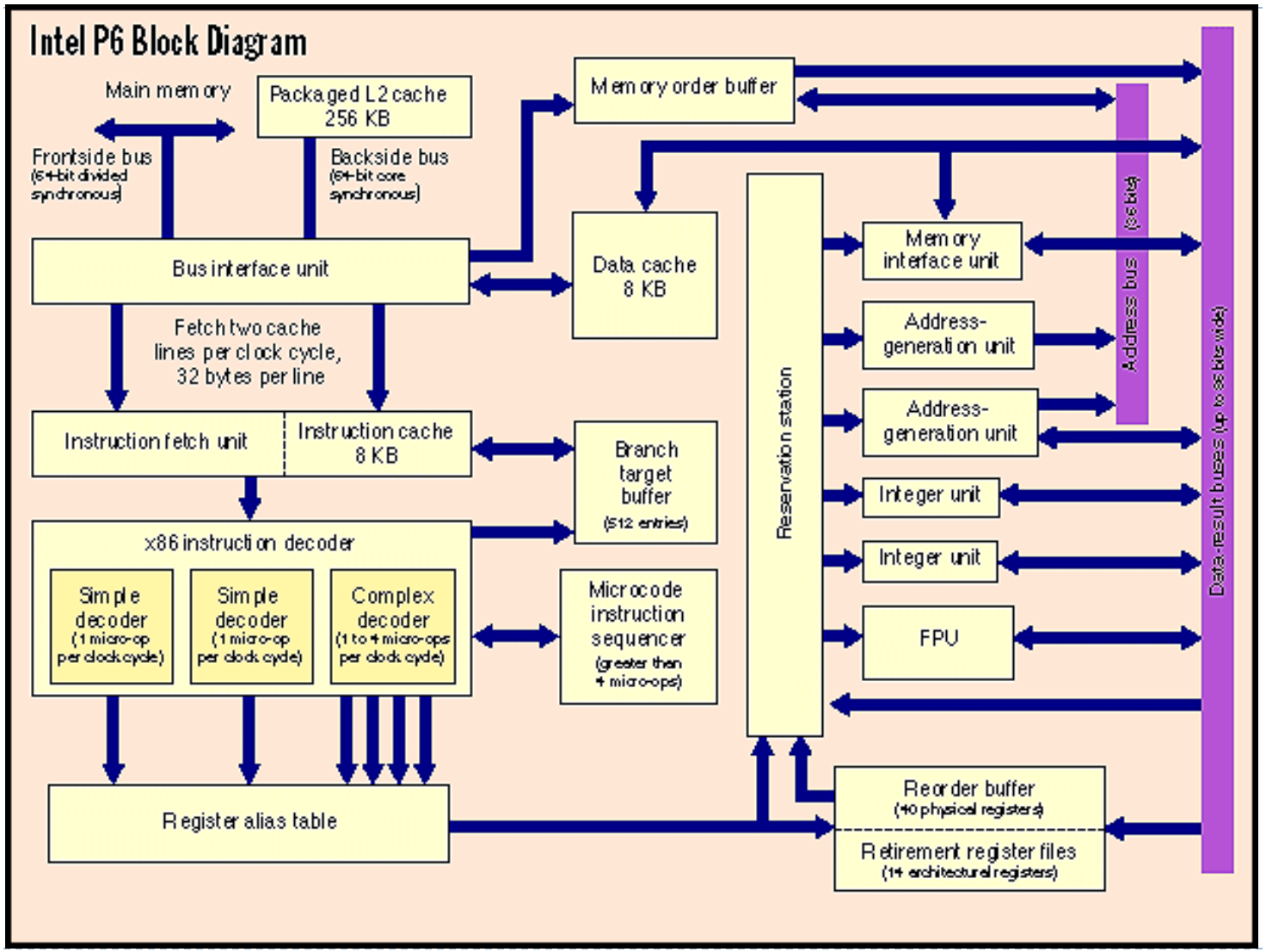
Arquitectura **registro-registro**:
instrucciones " load / store " *independientes*

Unidad de control sencilla.
Silicio disponible para otras funciones:
registros, caché, MMU, etc.

Facilita implementación de menor consumo

Tamaño de código considerable,
elevada velocidad de reloj

Procesador (ejemplo CISC→RISC -P6-)

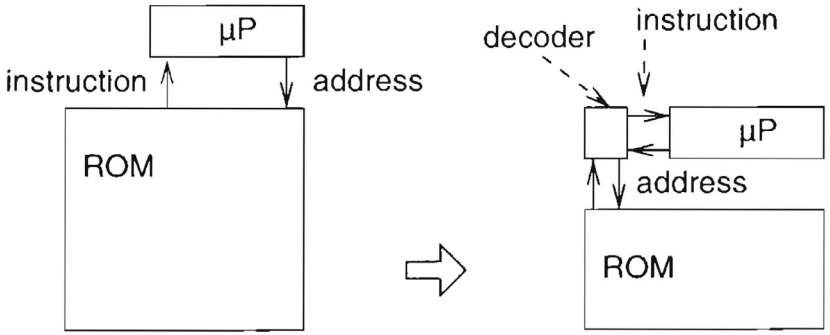


Memoria de programa eficiente

- CISC. Es uno de sus principales objetivos (reducción mem):
  - CISC: Una instrucción tiene mayor contenido semántico
  - RISC: ejecución más rápida
  - Frente a la tendencia actual de los µP de propósito general (→RISC), *tiene sentido* usar CISC en los sistemas empotrados… al menos en ciertos casos
  - Ejemplo (segundo enlace, hoy "*legacy*"):
  - https://www.nxp.com/applications/automotive:SECURE-CONNECTED-VEHICLE
  - low-cost-32-bit-microprocessor-including-hc000-hc001-ec000-and-sec000:MC68000
- Técnicas de compresión:
  - Almacenamiento comprimido y uso de decodificador
  - Uso de juego de instrucciones alternativo (ARM THUMB)
  - Uso de diccionarios

Instrucciones comprimidas (I)

- Con instrucciones comprimidas se limita:
  - El tamaño de la memoria (RAM y ROM) necesaria
  - La energía necesaria para hacer el *fetch* de instrucciones

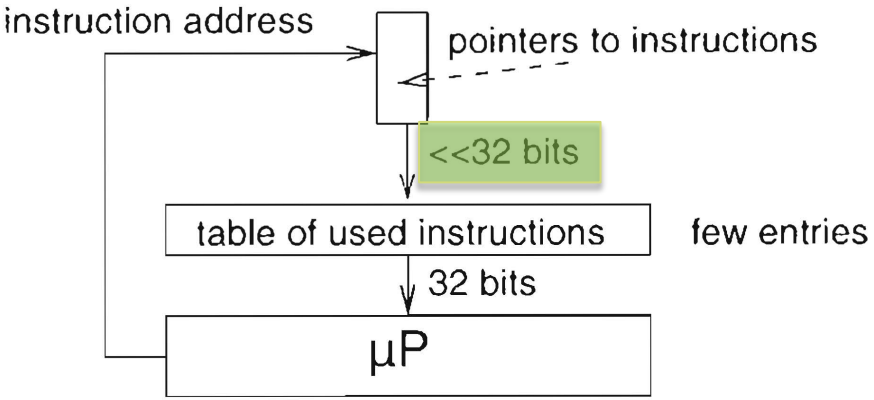


  - Objetivos de diseño:
    - Evitar que se añada penalización al ejecutar (tiempo).
    - Decodificación muy rápida. La *codificación* puede ser lenta.
    - Resolver el problema de los saltos.

(Memoria de programa eficiente)

Instrucciones comprimidas (II)

- Uso de diccionarios:
  - Almacenar cada instrucción diferente una única vez
    - ¿Qué se hace con los datos inmediatos?
  - Uso de una '*lookup table*' para acceder al 'diccionario'
  - Los punteros pueden ser de tamaño muy reducido

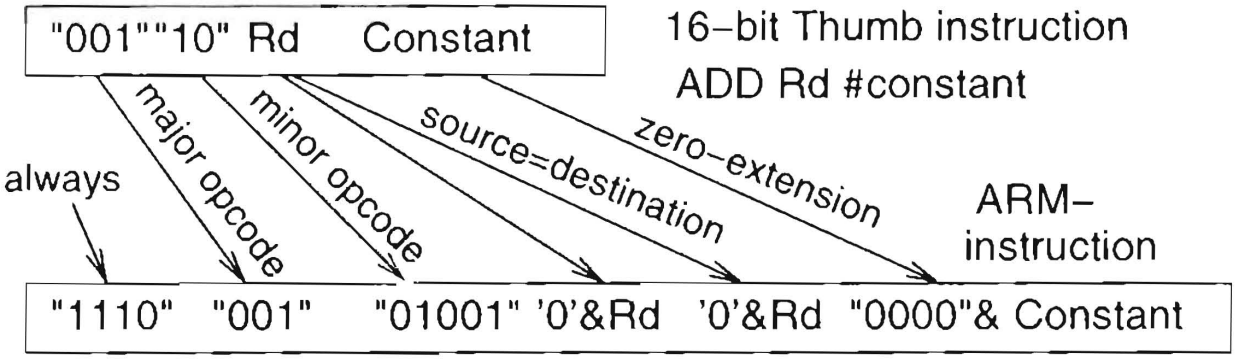


  - Denominaciones: almacenamiento de dos niveles, '*procedure exlining*', nanoprogramación

(Memoria de programa eficiente)

Instrucciones comprimidas (III)

- Segundo juego de instrucciones:
  - Utilización de un formato más compacto
  - Se restringe/impide el uso de algunas características: p.e. 'predicados' en cada instrucción
  - Ejemplo: ARM (32 bits) instrucciones THUMB



  - Coste incrementado en herramientas de desarrollo

(Memoria de programa eficiente)

Ejecución eficiente (Procesador)

"Convencional"

pipeline

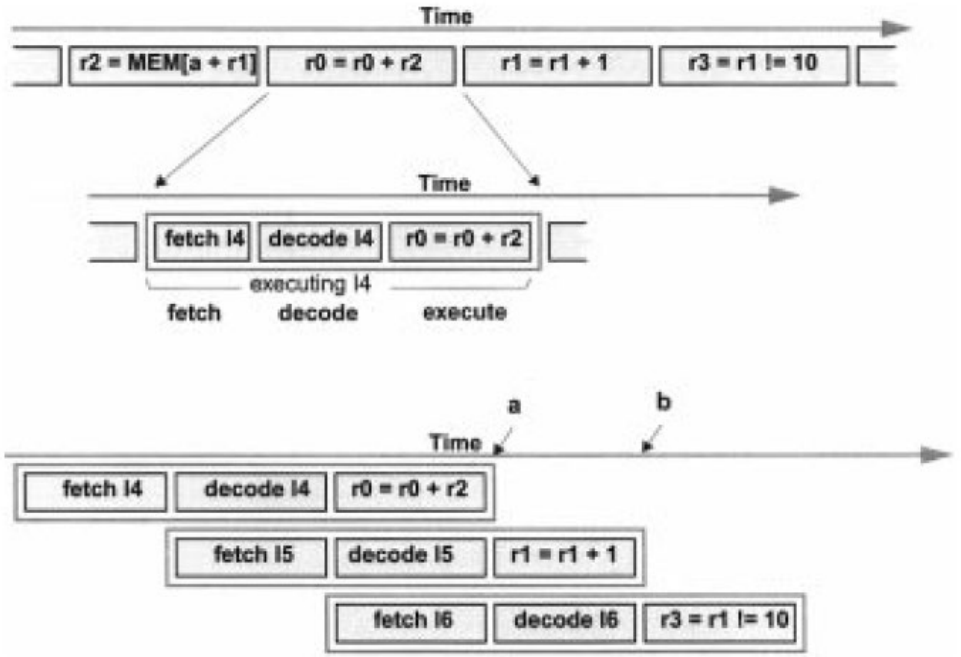
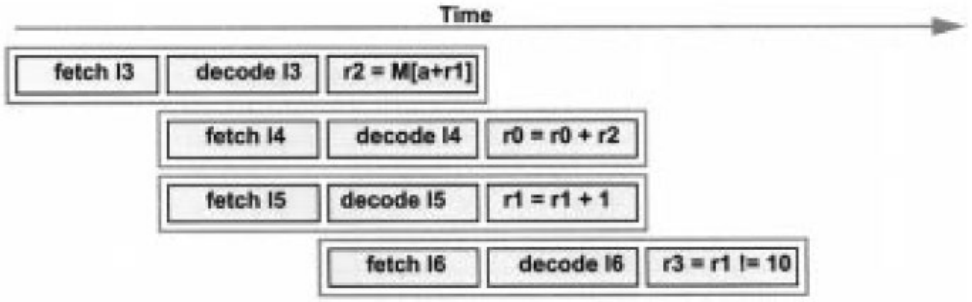
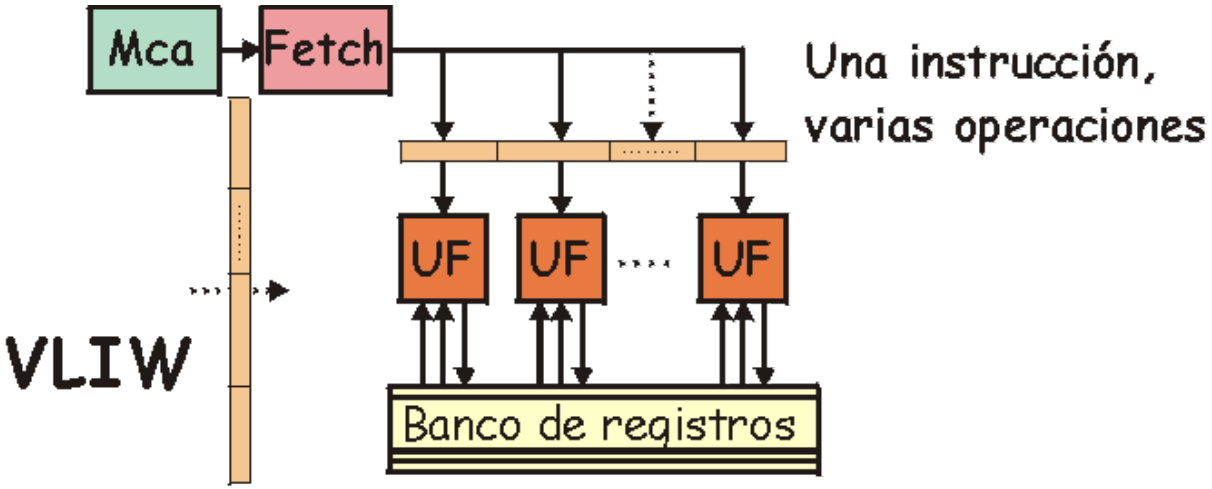


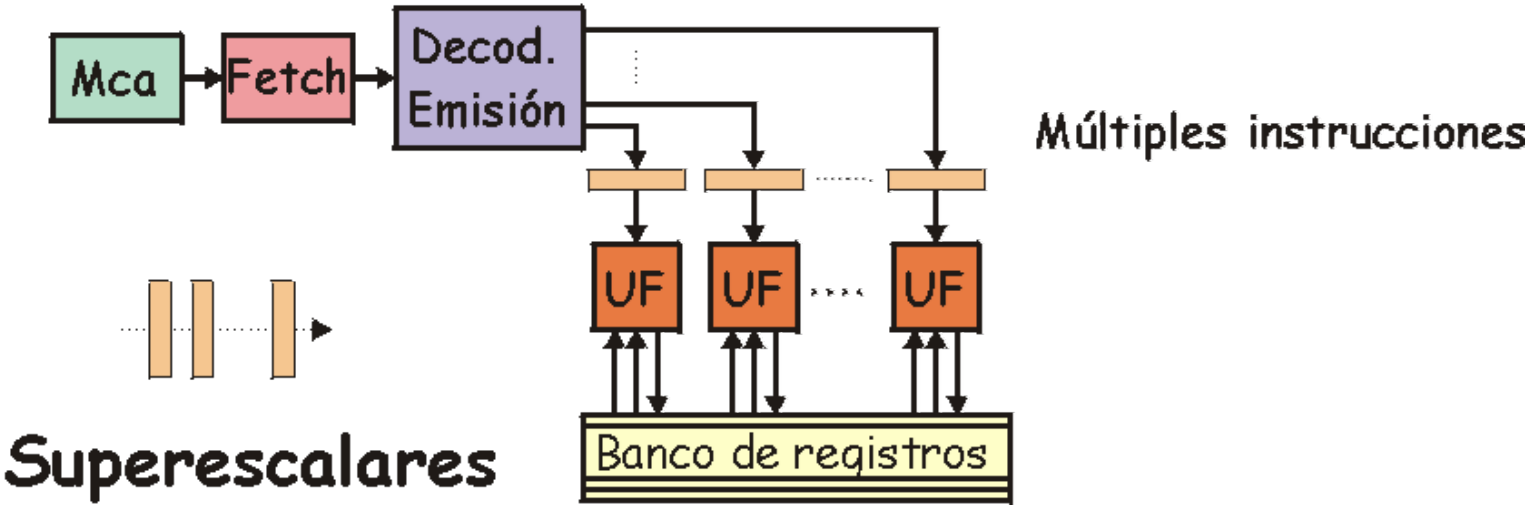
Fig. 2. (a) Execution without pipelining. (b) Steps required for executing a single instruction. (c) Execution with pipelining: the various execution steps of different instructions are overlapped.

superescalar



Procesador (VLIW / Superescalar)



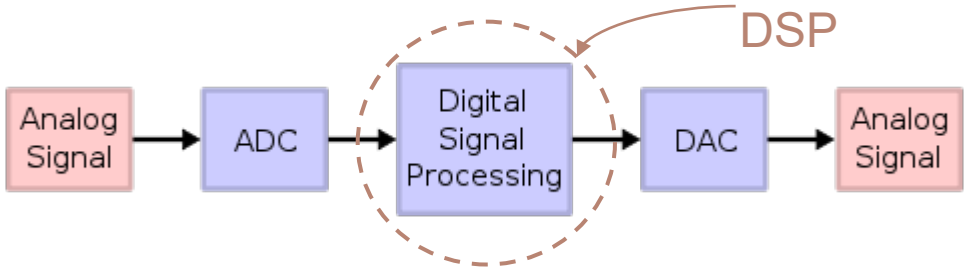


Procesador (VLIW - Superescalar)

- Hardware más sencillo, simplifica:
  - Decodificación
  - Detección de dependencias de datos
  - Emisión de instrucciones
  - Shelving
  - Renombrado
  - Despacho de instrucciones
  - Reordenamiento de las instrucciones
- Facilita:
  - Elevada frecuencia de reloj
  - Mayor número de unidades funcionales
- Requiere:
  - Mayor tamaño y ancho de banda para la memoria
  - Imposible programar en ensamblador
- Compilador más complejo:
  - Detectar y resolver dependencias:
    - Datos: Verdaderas. Antidependencias. Dependencias de escritura.
    - Control
  - Conocer detalles particulares de la *arquitectura* e *implementación* del procesador y sistema de memoria:
    - Tipo y número de unidades funcionales
    - Latencia y "tasa de repetición" de cada unidad funcional
    - Tiempos de espera del sistema de memoria (acierto/fallo en cache)

Procesador DSP (I)

- DSP, Digital Signal Processor.
  - Procesador especializado en ejecutar de forma eficiente las operaciones propias del procesado digital de señal.



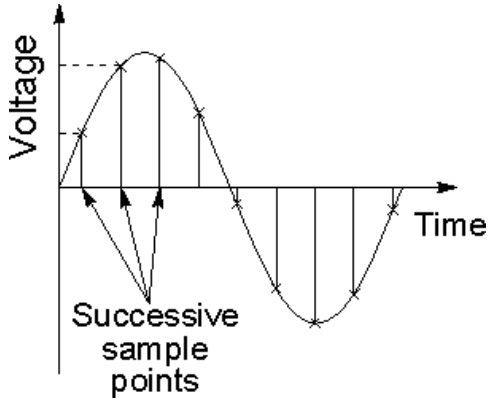
  - La operación típica del procesado de señal es el "multiply-accumulate":

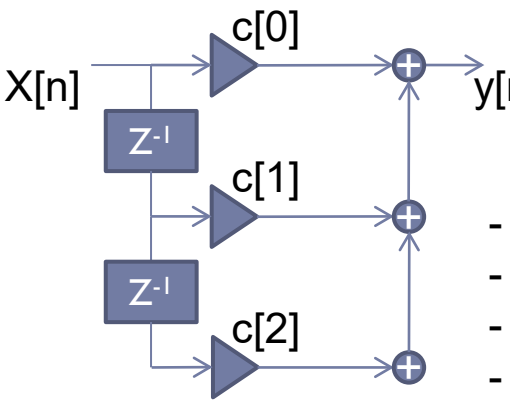
$$A \leftarrow A + B \times C$$

  - Se puede hacer con enteros o en coma flotante.
  - El redondeo (FP) puede ser doble (en cada operación) o único (como si fuese una sola operación: fused MAC o FMAC).

Procesador DSP (II)

Proceso digital de señal:





$y[n] = \Sigma (c[k] * x[n-k])$

- Sumas
- Productos
- Retardos
- Manejo de vectores/matrices

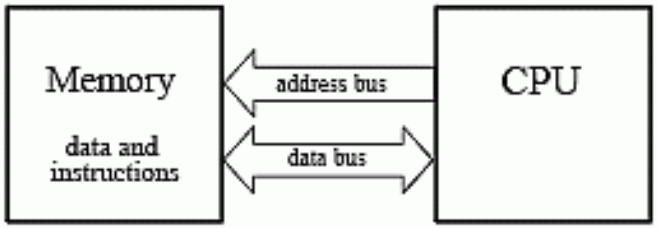
Normalmente se requieren múltiples operaciones de suma y producto.

A su vez, suma y producto necesitan:
- *fetch* de dos operandos/ciclo
- realizar suma y producto
- almacenar el resultado

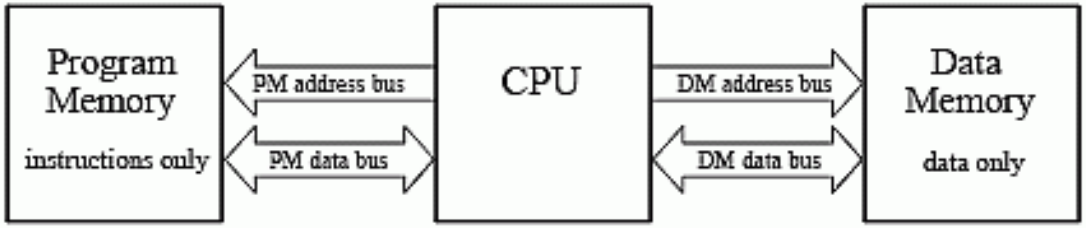
El aspecto crítico puede ser el *fetch* de dos operandos en un solo ciclo de instrucción.

DSP (III)

a. Von Neumann Architecture (*single memory*)



b. Harvard Architecture (*dual memory*)



c. Super Harvard Architecture (*dual memory, instruction cache, I/O controller*)

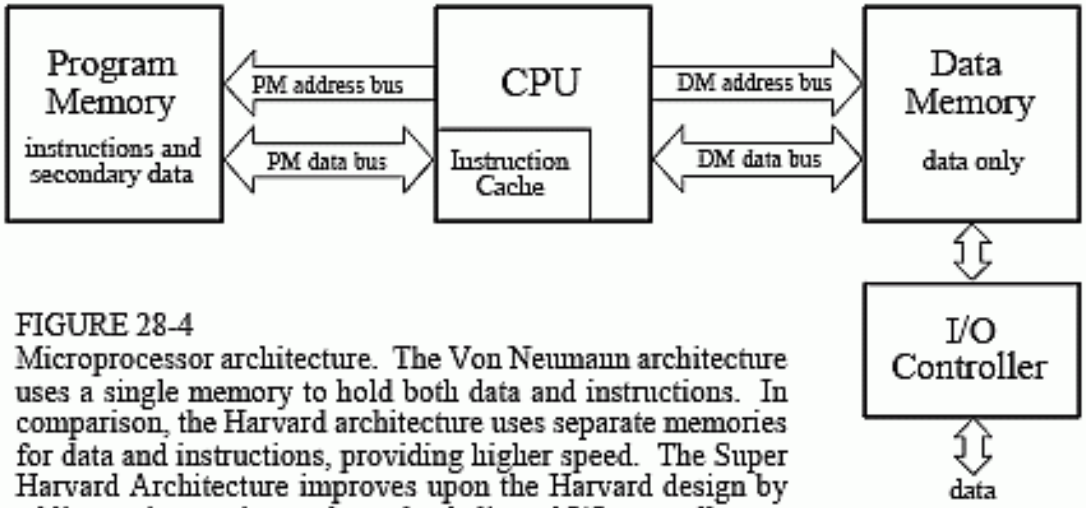
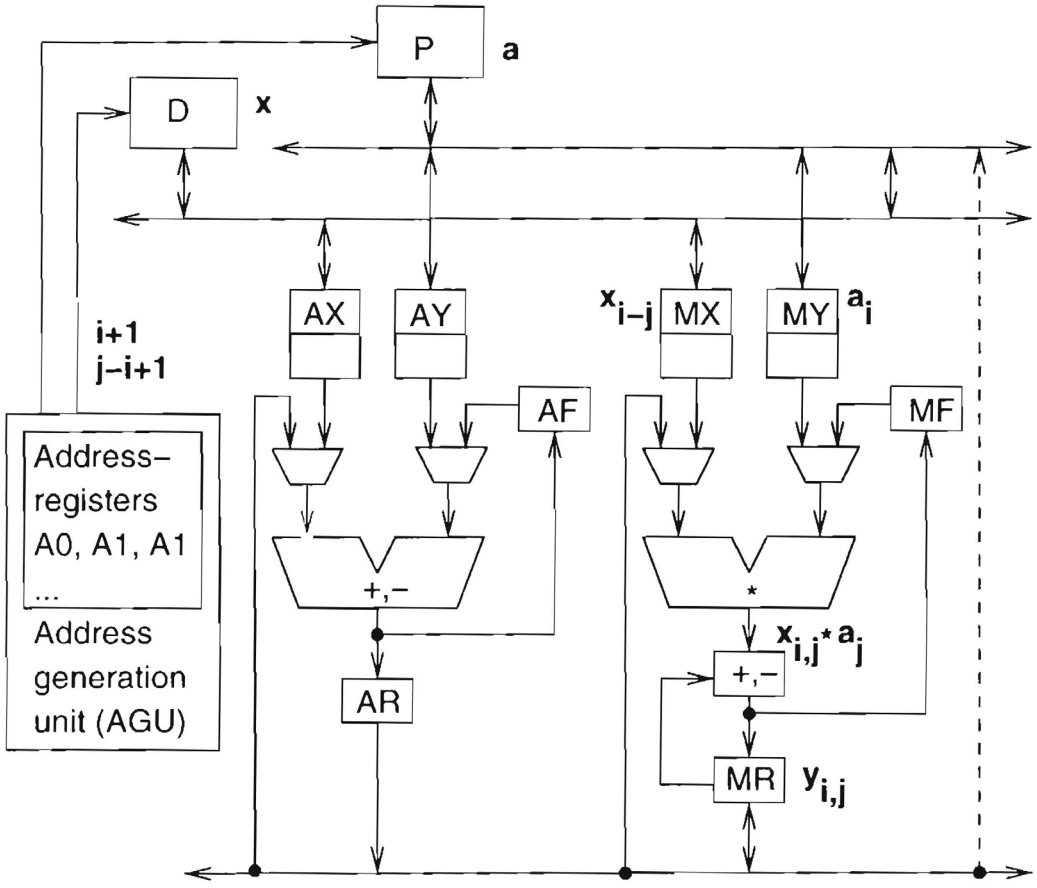


FIGURE 28-4
Microprocessor architecture. The Von Neumann architecture uses a single memory to hold both data and instructions. In comparison, the Harvard architecture uses separate memories for data and instructions, providing higher speed. The Super Harvard Architecture improves upon the Harvard design by adding an instruction cache and a dedicated I/O controller.

DSP (IV)



ADSP 2100 DSP processor

$$y_i = \sum_{j=0}^{n-1} x_{i-j} * a_j$$

```
MR:=0; A1:=1; A2:=n-2;
MX:=x[n-1]; MY:=a[0];

for (j=1; j<=n; j++) {
     MR:= MR + MX*MY;
     MX:= x[A2];
     MY:= a[A1];
     A1++; A2--;
}

(+ 0 overhead loop insts.)

TOT en DSP: 1 inst/iteration
```

Procesador DSP. Técnicas

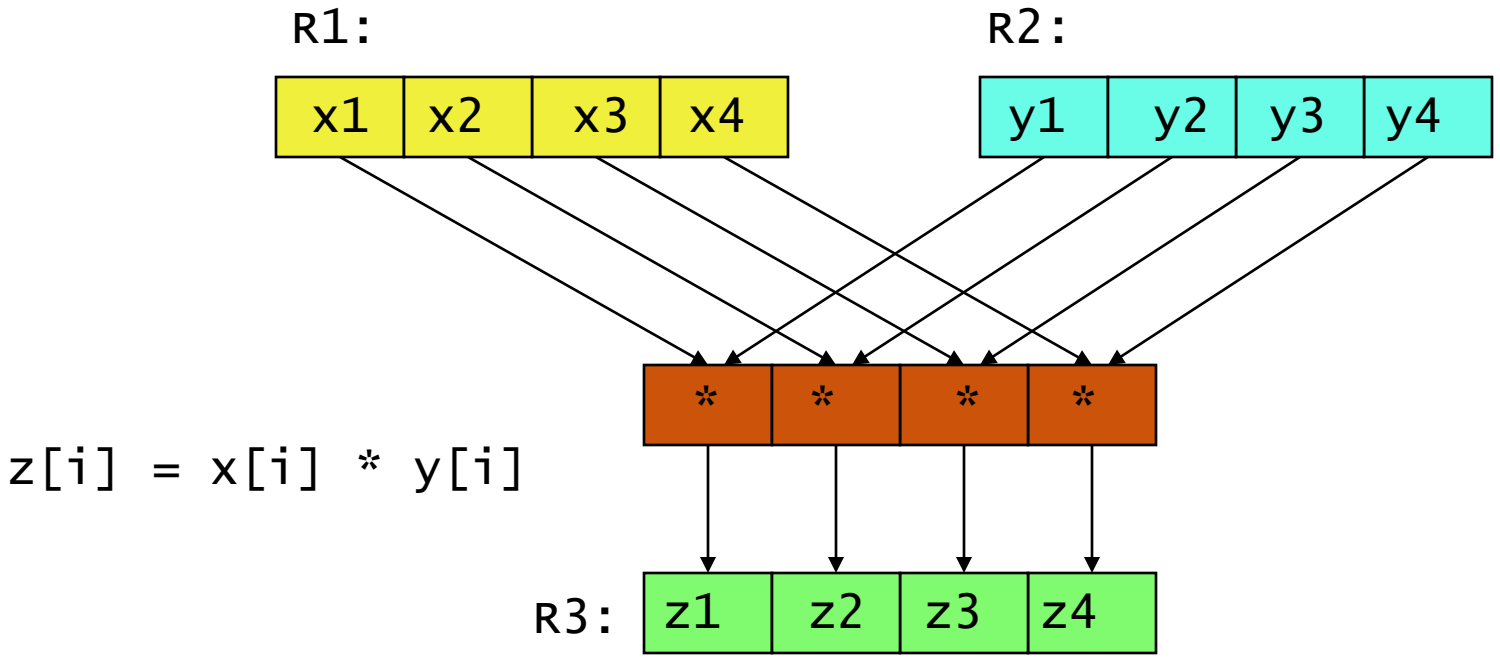
- Unidades independientes de generación de direcciones
- Para trabajar en tiempo real
- Bancos de memoria múltiples
- Bancos de registros heterogéneos (propósito general + específicos)
- Instrucciones producto/acumulación (FMAC)
- Aritmética de saturación

  Ejemplo:

```
0111+1001 = (1)0000 -aritmética convencional-
0111+1001 =    1111 -aritmética de saturación-
```

- Aritmética de coma fija (reducir coste y consumo):

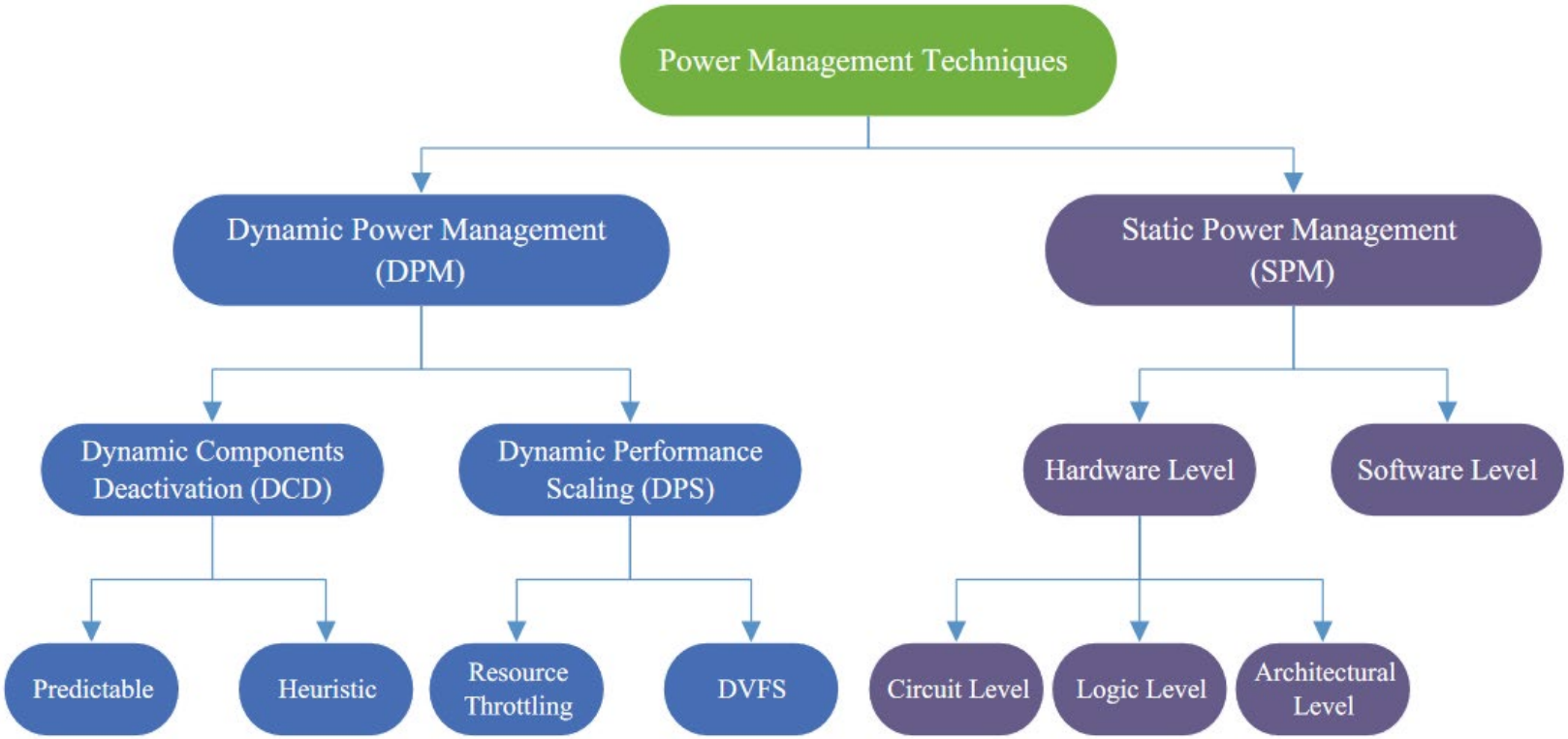
Procesador (CPU: extensiones multimedia)



- Modelo SIMD: una operación común a todos los datos
- Datos empaquetados en registros compartidos o especializados

Eficiencia. Consumo de energía

Clasificación de métodos para mejorar la gestión de energía[1]

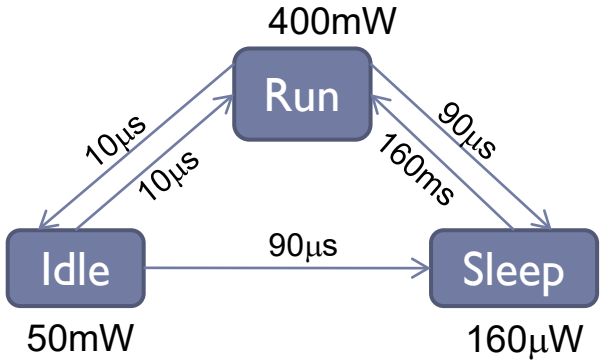


[1] Energy Efficient High Performance Processors: Recent Approaches for Designing Green High Performance Computing (Texto completo disponible en ResearchGate y a través de *Ingenio*)

Eficiencia. Consumo de energía I.

Dynamic Power Management

- Se dispone de varios modos de bajo consumo, además del normal ("operación").
- El cambio de modo tiene coste. El consumo global puede reducirse.
- Ejemplo (StrongARM):



Run state: Procesador *operativo*
Idle state: Solo monitoriza *interrupciones*
Sleep state: *Toda* la actividad del chip está desactivada

La aplicación óptima de esta técnica es compleja -normalmente basada en heurísticas- debido al *consumo y tiempo extra* necesario para cambiar de modo

Eficiencia. Consumo de energía II.

Dynamic Voltage Scaling

- Fundamento:

  A) El consumo de energía en circuitos CMOS crece de forma cuadrática con la tensión de alimentación y de forma lineal con la frecuencia (aprox.)

$$P = \kappa V_{dd}^2 f$$

  B) La **máxima** frecuencia de reloj crece de forma lineal con la tensión de alimentación (aprox.)

$$f = \lambda V_{dd}$$

  - → Al **reducir** la tensión de alimentación:
    - → disminuye el consumo (energía) de forma cuadrática
    - → crece el tiempo de proceso (disminuye la frecuencia máxima) de forma lineal

- Implementación:

  → Permitir diversos niveles de tensión de alimentación y conmutar de uno a otro en función de los requisitos que se dan en cada instante.

Eficiencia. Consumo de energía III.

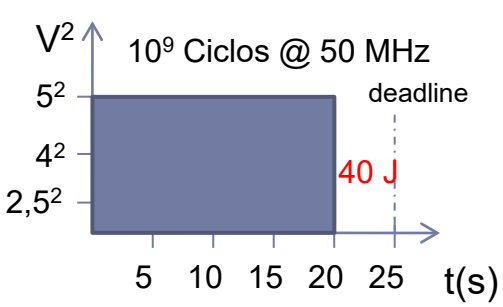
- Ejemplo[1] (Ishihara – Yasuura, 1998)

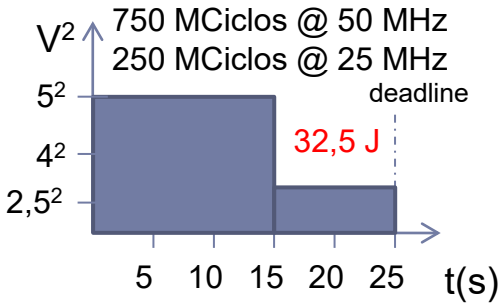
Dynamic Voltage Scaling

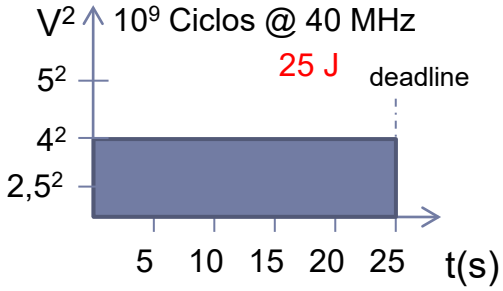
  - Procesador que puede trabajar a 2,5V, 4,0V y 5,0V.
  - Energía consumida por ciclo/frecuencia máxima/Período de reloj según tabla:

V_{dd}[volts]	5,0	4,0	2,5
Energía/ciclo [nJ]	40	25	10
f_{max} [MHz]	50	40	25
t de ciclo [ns]	20	25	40

  - Tarea a realizar: requiere 10^9 ciclos y tiene un plazo de ejecución de 25 segundos
  - Soluciones:







Ejemplo: Procesadores ARM

- ARM: Advanced RISC Machine. RISC de 32 bits (arquitectura Load/Store).
- Instrucciones de tamaño fijo (32 bits) + juego Thumb, de 16 bits. ARMv8: 64 bits.
- Procesadores simples, inicialmente para aplicaciones de baja potencia / consumo.
- Para compensar su simplicidad, se añaden algunas particularidades:
  - Ejecución condicional de la mayoría de las instrucciones.
  - Unidad de desplazamientos hardware de 32-bit que puede utilizarse con la mayoría de las instrucciones aritméticas y en el cálculo de direcciones.
- ARM: arquitectura con licencia comercial. Múltiples fabricantes: Alcatel-Lucent, Apple Inc., Atmel, Broadcom, Cirrus Logic, Digital Equipment Corporation, Freescale, Intel (through DEC), LG, Marvell Technology Group, Microsoft, NEC, Nuvoton, Nvidia, NXP (previously Philips), Oki, Qualcomm, Samsung, Sharp, STMicroelectronics, Symbios Logic, Texas Instruments, VLSI Technology, Yamaha and ZiiLABS.
- Algunas denominaciones: … Apple: ARM6, DEC: StrongARM, Intel: Xscale
- En enero de 2008 se habían fabricado más de 10.000 millones de ARM cores
- Solo en 2010 se vendieron más de 6.100 millones de procesadores ARM.
- en.wikipedia.org: 2011, ARM's customers reported 7.9 billion ARM processors shipped: 95% of smartphones, 90% of hard disk drives, 40% of digital televisions …

Difusión comercial: Procesadores ARM

Sales of chips containing ARM cores[148][149][150][151][152]

Year	Billion units	Relative size
2017	21.3	
2016	17.7	
2015	15	
2014	12	
2013	10	
2012	8.7	
2011	7.9	
2010	6.1	
2009	3.9	
2008	4.0	
2007	2.9	
2006	2.4	
2005	1.662	
2004	1.272	
2003	0.782	
2002	0.456	
2001	0.420	
2000	0.367	
1999	0.175	
1998	0.051	
1997	0.009	
Total	**150**[68]	

As of February 2020, over 160 billion chips with ARM IP have been shipped worldwide.

Ltd, Arm. "Record shipments of Arm-based chips in previous quarter". Arm | The Architecture for the Digital World. Retrieved 26 February 2020.

https://en.wikipedia.org/wiki/Arm_Ltd.

Ejemplo procesadores ARM

- Ejecución condicional ("predicados" en cada instrucción):
  - Algoritmo de Euclides para calcular el MCD:

```
while(i!=j) {
        if (i > j)
                i = i - j;
        else
                j = j - i;
}
```

  - Con instrucciones ARM:

```
loop    CMP Ri, Rj          ; set condition "NE" if (i != j),
                            ; "GT" if (i > j),
                            ; or "LT" if (i < j)
        SUBGT Ri, Ri, Rj    ; if "GT" (greater than), i = i-j;
        SUBLT Rj, Rj, Ri    ; if "LT" (less than), j = j-i;
        BNE loop            ; if "NE" (not equal), then loop
```

- Desplazamientos y rotaciones incorporados a otras instrucciones. Ejemplo:

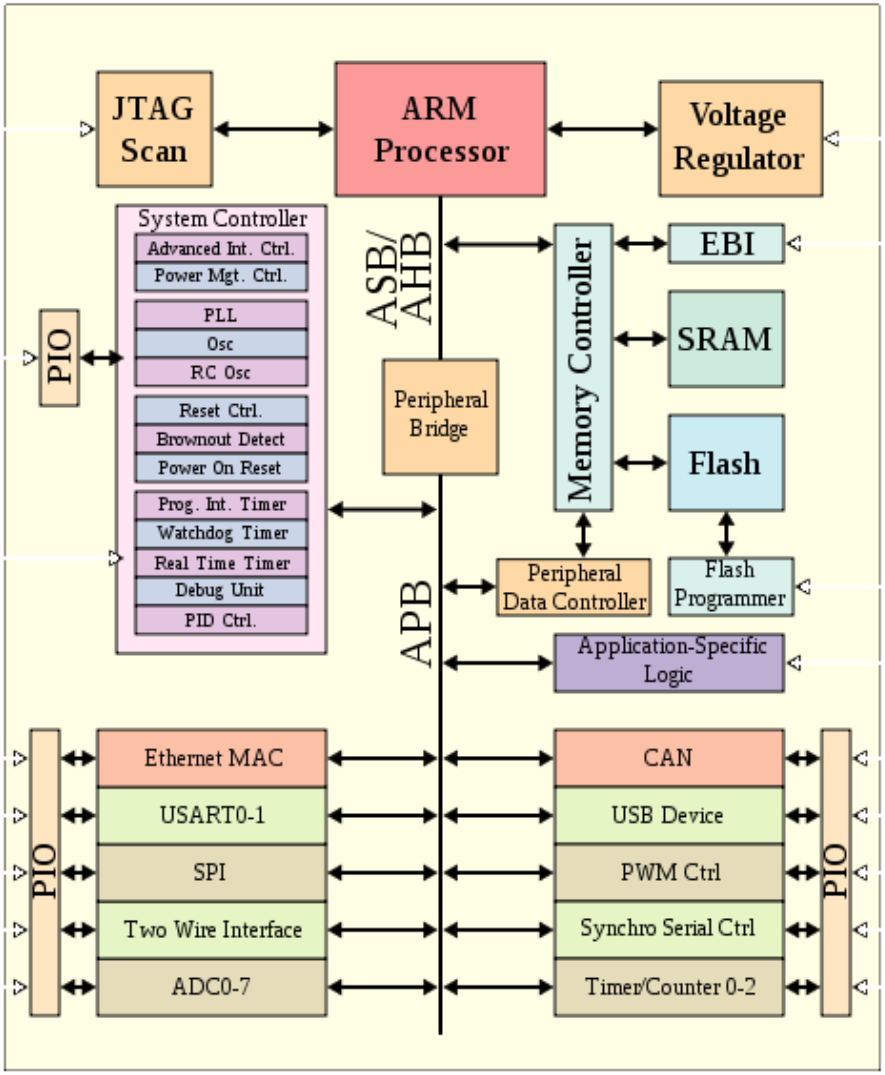
```
a = a + (j << 2);
```

  - Se puede implementar con una única instrucción de una palabra:

```
ADD Ra, Ra, Rj, LSL #2
```

Procesadores (System on a Chip)

Ejemplo de System-on-a-chip

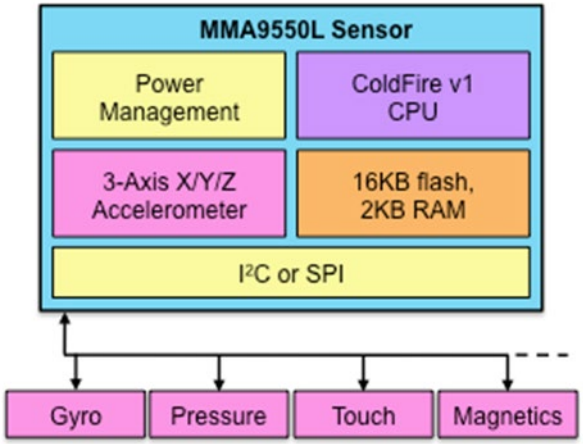


Ejemplos (ARM+sensores)

FREESCALE ADDS SMARTS TO SENSORS

Xtrinsic Embeds ColdFire into Accelerometers, Gyros, Other Sensors

By Jim Turley {07/19/10-02} (Microprocessor-Reports July-2010)



"Freescale hasn't announced pricing for its Xtrinsic 9550L three-axis accelerometer chip, except to suggest it will likely cost **around $2.50** in small quantities—more than triple the price of a typical, non-intelligent equivalent."

Fully 90% of smartphones will include an **accelerometer** by 2014, according to ARCchart, and 60% will have a **gyroscope** and a magnetometer (**compass**).

Figure 1. MMA9550L block diagram. The first Xtrinsic "smart" sensor chip includes a three-axis accelermeter, ColdFire v1 CPU core, power-management, and serial interfaces to an upstream host processor and up to 12 optional downstream slave sensors.

Índice

- Introducción / HW para sistemas empotrados
- Características y componentes de un sistema empotrado
- Procesadores:
  - CPU / FPGA / DSP / VLIW
- Computadores modulares
- Problemas/Ideas/Técnicas utilizadas en s. empotrados

- Normativa (Actualmente en grado en II: 'Proyecto de Instalación Informática')

Sistemas modulares (ejemplo: PC/104)

http://www.pc104.org/

Specification: PC/104-*Plus*
Product Type: CPU or Single Board Computer (SBC)
Features:
High-performance Pentium M Processor
Extreme Graphics 2 Video
10/100 Ethernet
High-speed DDR RAM
CompactFlash socket
Small PC/104-*Plus* form factor
RoHS-compliant versions

The Cheetah is a high-performance stand-out in a compact PC/104-*Plus* system. It offers flexible RAM options and scalable processing for optimum application performance with minimal power draw.

VersaLogic Cheetah

Specification: PC/104
Product Type: Motor Controller

Single board DC servo motor solution,0º to +70ºC
Two independent motor interfaces
Control position, velocity and acceleration using dedicated motor control chipset
Two full bridges for direct motor connection
60 V, 10 A onboard MOSFET H-bridges
Incremental Encoder inputs
Galvanic isolation (ISC629)
Output control port for external power stage
24 TTL Digital I/O, 8255 based

RTD Embedded Technologies, Inc
DC Servo Motor Controller - ESC629/ISC629

Prototipos de bajo coste. Arduino

What is arduino?

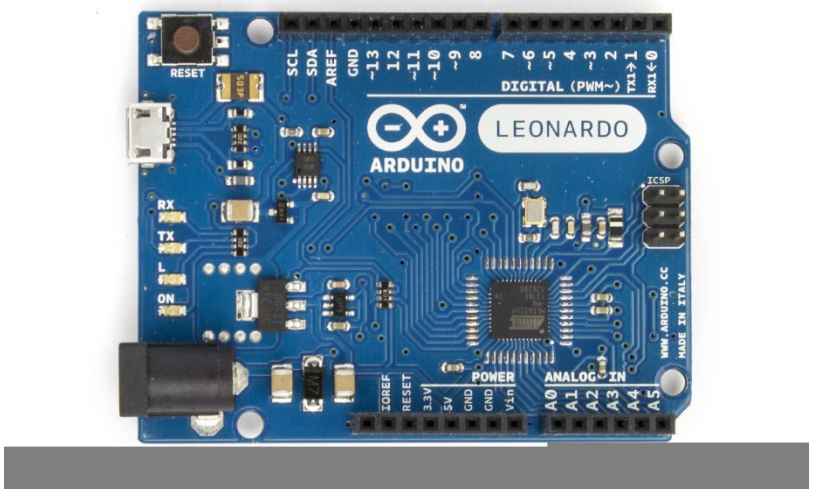
Arduino is an open-source electronics platform based on easy-to-use hardware and software. It's intended for anyone making interactive projects.

ARDUINO BOARD

Arduino senses the environment by receiving inputs from many sensors, and affects its surroundings by controlling lights, motors, and other actuators.

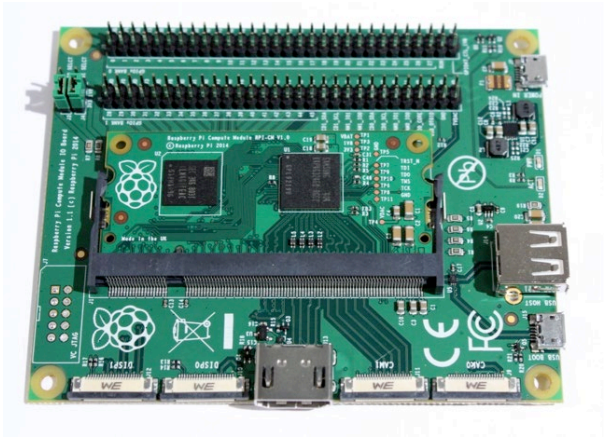
ARDUINO SOFTWARE

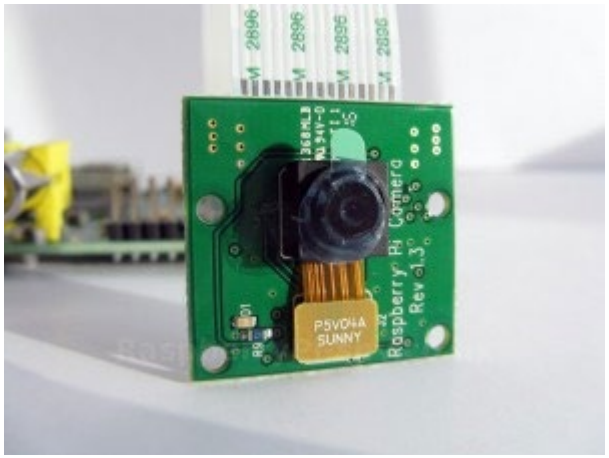
You can tell your Arduino what to do by writing code in the Arduino programming language and using the Arduino development environment.



Prototipos de bajo coste. Raspberry-Pi

The Raspberry Pi is a credit-card sized computer that plugs into your TV and a keyboard. It is a capable little computer which can be used in electronics projects, and for many of the things that your desktop PC does, like spreadsheets, word-processing and games. It also plays high-definition video. We want to see it being used by kids all over the world to learn how computers work, how to manipulate the electronic world around them, and how to program.





Raspberry-Pi: modelos recientes

	Raspberry Pi 400	Raspberry Pi 4 B 8GB	Raspberry Pi 4 B	Raspberry Pi 3 Model A+	Raspberry Pi 3 B+
Imagen					
Release date	2/11/2020	28/5/2020	24/6/2019	15/11/2018	14/3/2018
Descripción	Raspberry Pi 4 directly integrated into a keyboard with improved CPU clock speed	Raspberry Pi 4 B version with 8GB memory and layout of power supply components updated.			
Product details					
Precio	70,00 $	75,00 $	35,00 $	25,00 $	35,00 $
SOC					
SOC Type	Broadcom BCM2711	Broadcom BCM2711	Broadcom BCM2711	Broadcom BCM2837B0	Broadcom BCM2837B0
Core Type	Cortex-A72 (ARM v8) 64-bit	Cortex-A72 (ARM v8) 64-bit	Cortex-A72 (ARM v8) 64-bit	Cortex-A53 64-bit	Cortex-A53 64-bit
No. Of Cores	4	4	4	4	4
GPU	VideoCore VI	VideoCore VI	VideoCore VI	VideoCore IV	VideoCore IV
CPU Clock	1.8 GHz	1.5 GHz	1.5 GHz	1.4 GHz	1.4 GHz
RAM	4 GB LPDDR4	8 GB LPDDR4	1 GB , 2 GB, 4 GB LPDDR4	512 MB DDR2	1 GB DDR2

Índice

- Introducción / HW para sistemas empotrados
- Características y componentes de un sistema empotrado
- Procesadores:
  - CPU / FPGA / DSP / VLIW
- Computadores modulares
- Problemas/Ideas/Técnicas utilizadas en s. empotrados

- Normativa (Actualmente en grado en II: 'Proyecto de Instalación Informática')

Sistemas empotrados. Problemas

- Presupuesto limitado / producción a gran escala:
  - Sistemas a la medida, ahorro de componentes, uso limitado de memoria,
- Alimentación por baterías → bajo consumo:
  - Sistemas a la medida, técnicas de reducción de consumo, modos de funcionamiento seleccionables
- Plazo de respuesta estricto → tiempo real (*hard*):
  - Evitar técnicas de mejora "estadística": memorias caché, predicción de saltos, paginación
- Tolerancia a fallos → sistemas críticos (funcionamiento asegurado):
  - Técnicas de tolerancia a fallos
- Altas prestaciones:
  - Multiprocesador, VLIW, Superescalar, DSP, etc.
- Robustez mecánica, tolerancia a golpes:
  - Ausencia de partes móviles, componentes adaptados a la función (vehículos, máquinas, etc)
- Inmunidad a la radiación y a ruido electromagnético
  - Encapsulado apropiado, componentes de mayor tamaño

Problemas/Soluciones I.

- Presupuesto limitado / producción a gran escala:
  - **Ahorro de componentes, uso limitado de memoria: microcontroladores/ASIC** (Application Specific Integrated Circuits) Ejemplos:
  - Tarjetas musicales (audio), juguetes, relojes y temporizadores electrónicos
  - Teléfonos, GPS, Cámaras de fotos/vídeo, Libros electrónicos
  - Tarjetas bancarias / de identificación (incluyen cifrado)
  - Ejemplo, el DNI 3.0:

    http://www.dnielectronico.es/PDFs/Guia_de_Referencia_DNIe_con_NFC.pdf
    - Chip SLE78CLFx408AP de Infineon Technologies.
    - Características:
    - 400KB memoria Flash (código + personalización)
    - 8 KB memoria RAM
    - Dual Interface (contacto + sin contacto)
    - Criptolibrería RSA

Problemas/Soluciones II.

- Alimentación por baterías → bajo consumo:
  - **Sistemas a la medida, técnicas de reducción de consumo: DPM y DVS**
  - **Modos de funcionamiento/espera: Suspensión e Hibernación (ACPI)**
  - Ejemplos: Intel SpeedStep (*DVS*), AMD PowerNow! (*DVS*), IBM EnergyScale (*DVS*)
  - Apagado de componentes: Intel Core functional units power control, AMD CoolCore

- Tiempo Real:
  - Memoria Virtual:
    - Ventaja para facilitar protección entre procesos, pero desventaja para TR:
      - Genera plazos de respuesta impredecibles
      - Fallos de página → críticos (evitables, gestión de la MV hw/sw)
  - Memorias Caché:
    - Ventaja para obtener mayor eficiencia en la ejecución, pero desventaja para TR:
      - Fragmentos de código con tiempo de ejecución no predecible / máximo tiempo de ejecución mayor que sin M Caché

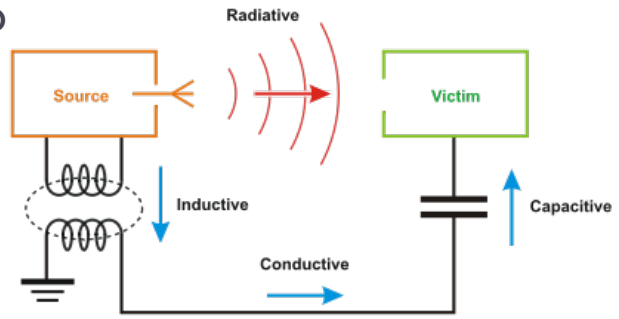
Problemas/Soluciones III.

- Alternativas a las Memorias Caché:
  - Uso de zonas "*no-cacheables*"
  - Uso de *scratchpad* RAM:
    - Zona de memoria especializada (mayor velocidad)
    - Non-uniform Memory Access (NUMA)
    - No tiene porqué contener una copia del siguiente nivel (L1-L2-L3-Mp)
    - Gestionada por las aplicaciones, no por el HW

- Tolerancia a fallos → sistemas críticos:
  - Un sistema se dice ***tolerante a fallos*** cuando puede continuar trabajando, posiblemente en un modo degradado, cuando alguno de sus componentes falla.
  - Soluciones:
    - Redundancia (HW y/o SW). Ejemplos:
      - Discos duros (RAID), Fuentes de alimentación, Procesadores/Memoria
  - Algunas desventajas:
    - Coste. Riesgos de "ocultación":
      - del fallo(s) al ser compensado
      - de fallos en otros componentes

Problemas/Soluciones IV.

- Altas prestaciones:
  - Multiprocesador, VLIW, Superescalar, DSP, etc.
- Inmunidad a la radiación y al ruido electromagnético
  - Encapsulado apropiado, componentes de mayor tamaño
  - Interferencia Electromagnética (EMI):
    - Emisiones electromagnéticas de un dispositivo o sistema que interfieren con el funcionamiento normal de otro dispositivo o sistema.
    - Se denominan también Interferencias de Radiofrecuencia (RFI)



  - Compatibilidad Electromagnética (EMC)
    - Capacidad de los equipos o sistemas para trabajar satisfactoriamente en su entorno electromagnético (EME) sin introducir ruido o influencia electromagnética indeseable en cualquier otro sistema de dicho entorno.
  - Resumen:
    - Debe evitarse la generación de interferencias que sobrepasen un cierto nivel prefijado
    - Debe tolerarse un cierto nivel predeterminado de interferencias.
    - Los sistemas deben ser "auto-compatibles" electromagnéticamente.

Problemas/Soluciones V.

- Robustez mecánica, tolerancia a golpes:
  - Encapsulado especial: problema de ingeniería mecánica:
    - PC industrial
    - PC plano, pantalla táctil, industrial
  - Ausencia de partes móviles:
    - Eliminación de ventiladores
    - Discos duros sustituidos por discos de estado sólido

Fanless touch screen industrial panel PC

Embedded panel PC

Point of sale panel PC (POS)

Stainless steel industrial panel PC

Industrial panel PC **(IP65)**

Touch screen panel PC for medical applications **(IP65/54)**

Sistemas Empotrados: HW