

Sistemas Embebidos en Tiempo Real

Maestría en Inteligencia Artificial

2024

1

Preguntas Previas

- ¿Qué es un sistema embebido (Embedded System) y para qué sirve?
 - Un sistema embebido es un pequeño sistema informático diseñado para realizar una función específica dentro de un dispositivo más grande.
 - En mi opinión, los sistemas embebidos pueden permitir capturar una gran cantidad de datos del mundo físico para generar modelos de inteligencia artificial que pueden ser usados para predicción, anomalías, mantenimiento predictivo, seguridad, etc.
 - Un sistema embebido cuenta con diversas capacidades gracias a sus sistemas (serie de componentes que tienen un objetivo en común), CPU, Memoria, entre otros. En la captura de información, IoT, TB en el uso eficiente de los modelos de IA con TinyML
 - No he tenido experiencia previa con el tema, pero en mi opinión se refiere a una combinación entre software y hardware para realizar una tarea en específico
 - Para mí, los sistemas embebidos es el que se realiza un conjunto de funciones (dónde el software está embebido con el hardware), operando muchas veces sin intervención humana.
 - Son sistemas que realizan tareas acotadas o puntuales limitados al hardware donde se ejecuta
 - un sistema embebido es el agrupamiento de componentes electrónicos y pueden servir para capturar datos o funcionar como pequeñas computadoras

2

Sistemas Embebidos

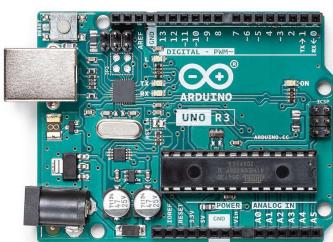
- ¿Cuál es la pertinencia de un sistema embebido en el desarrollo de aplicaciones relacionadas con IA?
- SBC (single-board computer):
 - Raspberry Pi
 - Arduino
 - Arduino Opta (PLC)
 - Nvidia Jetson Nano
 - Orange Pi
 - Rock Pi
 - Google Coral



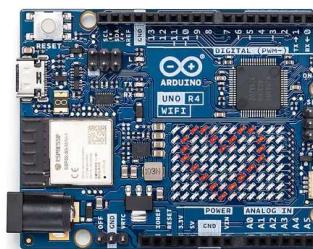
3

Sistemas Embebidos

- Arduino UNO



R3
μP 8bit



R4
μP 32bit

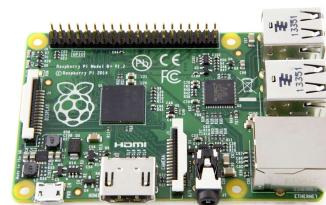
4

Preguntas Previas

- ¿Qué es un sistema embebido (Embedded System) y para qué sirve?
 - desconozco el termino y es la primera vez que lo escucho
 - son minicomputadoras que cumplen tareas en base a sensores.
 - sistemas que hacen funciones específicas
 - es un sistema que realiza funciones específicas, tareas particulares dentro de un sistema mas complejo
 - son sistemas de computación especializados que están integrados dentro de un dispositivo o producto más grande y que se diseñan específicamente para realizar una o varias funciones concretas de manera eficiente
 - un claro ejemplo son los microcontroladores
 - forman parte de un sistema mayor, diseñado con un propósito particular.
 - Es un sistema de computación especializado que forma parte de un dispositivo mayor.
 - es un conjunto de componentes electrónicos y software diseñados para realizar tareas específicas dentro de un entorno controlado, integrándose en un dispositivo mayor.
 - Sistema compuesto de diversos sensores que recopilan datos para una tarea en comun de manera práctica, facil y rápida
 - combinación de hardware y software, se encuentra "embebido" en el sentido de que no opera de manera independiente; más bien, está integrado para controlar funciones en tiempo real en dispositivos como automóviles, electrodomésticos, teléfonos inteligentes, dispositivos médicos, entre otros.

5

Preguntas Previas



- ¿Qué es un sistema embebido (Embedded System) y para qué sirve?
 - Es una computadora de placa reducida (SBC – Single Board Computer), no solamente de dimensiones, sino de periféricos y recursos.
 - Es un sistema electrónico en cuál comple las funciones escritas en un firmware.
 - Da soluciones a requerimientos de algún proceso a desarrollar.
 - ¿Lenguajes de programación?
 - Arduino – Similar al C, no es lo mas óptimo.
 - RPI – Python, C, Assembler

6

Sistemas Embebidos

- ¿Cuándo utilizar Arduino y cuándo utilizar RPI?
 - Dependerá de los requerimientos de la aplicación
 - Hay aplicaciones muy sencillas como muy complejas
- ¿Diferencias de programación?
 - Arduino – Lenguaje muy similar al C
 - RPI – Múltiples lenguajes: C, Python

7

Agenda

- Conocimientos previos
- Definición de Sistema Embebido

8

Conocimientos previos

- Fundamentos de la electricidad y electrónica
- Fundamentos de sistemas digitales
- Fundamentos de arquitectura de computadores

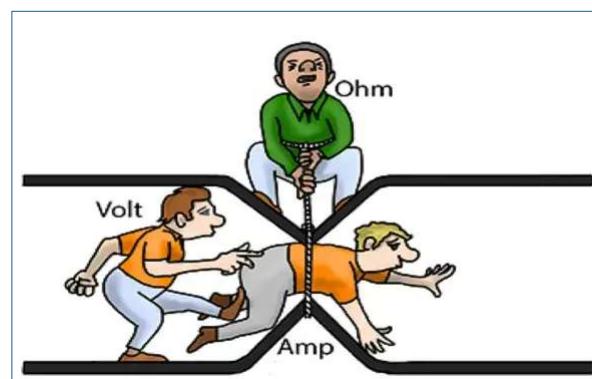
9

Conocimientos previos

• Ley de Ohm: Relación entre Voltaje, corriente y resistencia

- V representa el voltaje aplicado en el circuito, medido en voltios (V).
- I representa la corriente que circula por el circuito, medido en amperios (A).
- R representa la resistencia del circuito, medida en ohmios (Ω).

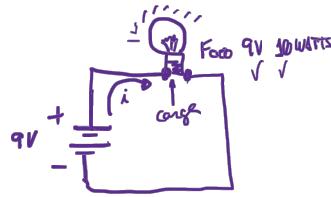
$$V = I \times R$$



10

Ejemplo de circuitos eléctricos

- Elementos activos → entregan energía
- Elementos pasivos → consumen energía



Ley Ohm's
 $V = R \cdot I$
 Ley Watt's
 $P = V \cdot I$
 $I = \frac{P}{V}$
 $I = \frac{10}{9}$ amperios
 $I = 1.11A$

11

Dimensionamiento de elementos de protección eléctrica.

En tu casa:

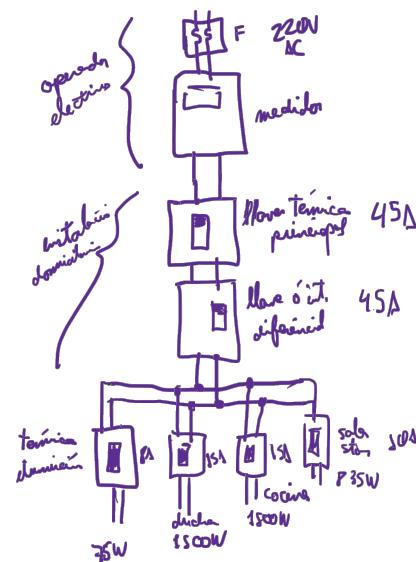


5 focos 15W
 1 televisor 35W
 1 ducha eléctrica 1500W
 1 horno microondas 500W
 1 lavadora de agua 1000W
 1 PC 800W
 $\sum 4000W$ aprox

$$P = V \cdot I$$

$$I = \frac{P}{V} = \frac{4000W}{220V}$$

$$I = 18.18A$$



12

Conocimientos previos

- Potencia eléctrica
 - Cantidad de energía utilizada en un tiempo determinado

The diagram consists of three separate triangles, each with a hand pointing to a different part of the formula. The first triangle has 'V' at the top vertex and 'I' at the bottom vertex. The second triangle has 'P' at the top vertex and 'I' at the bottom vertex. The third triangle has 'P' at the top vertex and 'V' at the bottom vertex.

$$P = V \cdot I$$

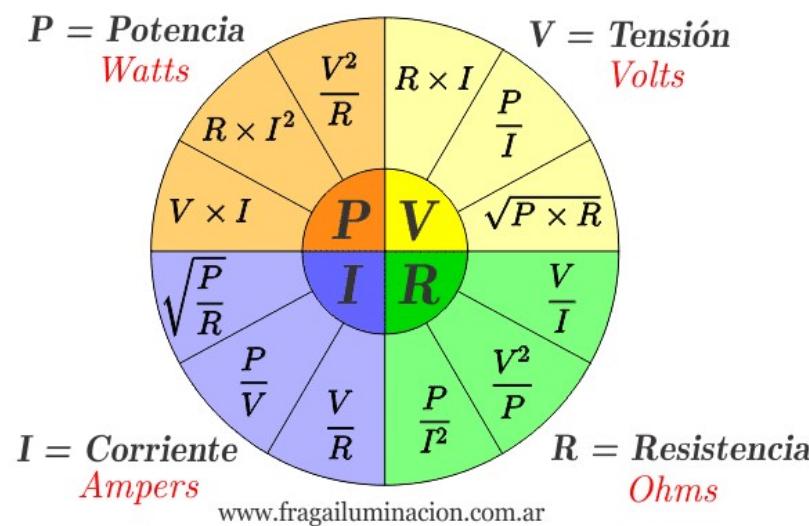
$$V = \frac{P}{I}$$

$$I = \frac{P}{V}$$

13

Conocimientos previos

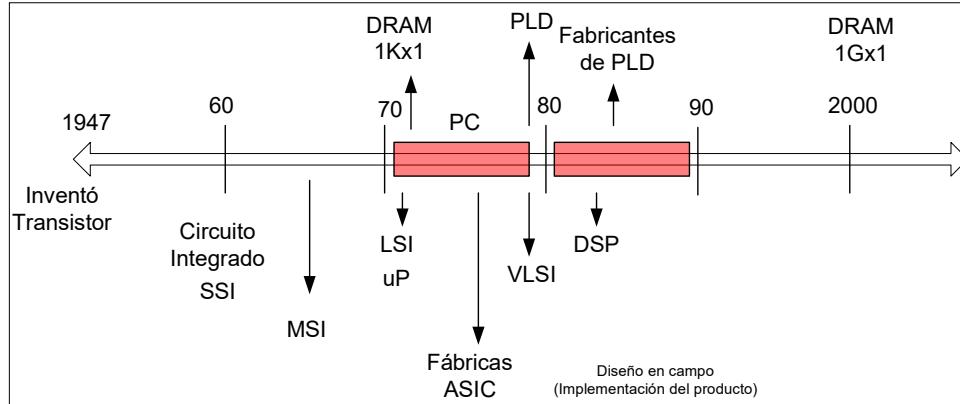
- Combinando ambas leyes:



14

Conocimientos previos

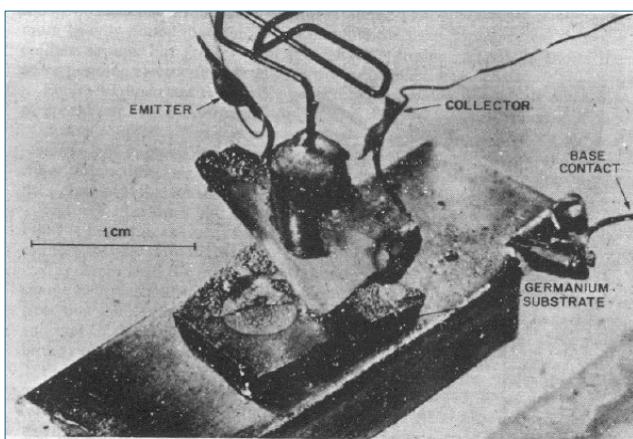
Evolución de los Sistemas Digitales



15

Conocimientos previos

El Primer Transistor (1947)



- Inventado el 23 de Diciembre de 1947
- William Shockley, John Bardeen y Walter Brattain.
- Material empleado: Germanio



16

Conocimientos previos

El Transistor

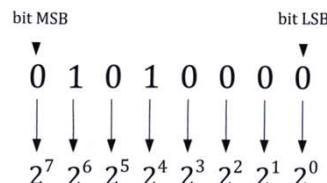
- El **transistor** es un dispositivo electrónico semiconductor que cumple funciones de amplificador, oscilador, conmutador o rectificador.
 - El término "transistor" es la contracción en inglés de *transfer resistor* ("resistencia de transferencia").
 - Actualmente se los encuentra prácticamente en todos los enseres domésticos de uso diario: radios, televisores, grabadores, reproductores de audio y vídeo, hornos de microondas, lavarropas automáticos, automóviles, equipos de refrigeración, alarmas, relojes de cuarzo, computadoras, calculadoras, impresoras, lámparas fluorescentes, equipos de rayos X, tomógrafos, ecógrafos, reproductores mp3, celulares, etc.

17

Conocimientos previos

- ### • Aritmética binaria

Decimal: 165
Binario: 10100101
Octal: 245
Hexadecimal: A5



Tamaño	Valor mínimo y máximo	Denominación
1 bit	0 - 1	Bit
4 bits	0000 - 1111	Nibble
8 bits	0000 0000 - 1111 1111	Byte
16 bits	0000 0000 0000 0000 - 1111 1111 1111 1111	Word
32 bits	0000 0000 0000 0000 0000 0000 0000 0000 - 1111 1111 1111 1111 1111 1111 1111 1111	Double Word

En binario:

$$1 + 1 = 10$$

Operador aritmético

$$1 \text{ (or)} 1 = 1$$

Operador lógico

18

Conocimientos previos

- Operadores lógicos

NOT	AND	NAND	OR	NOR	XOR	XNOR																																																																																																
\bar{A}	AB	\overline{AB}	$A+B$	$\overline{\overline{A+B}}$	$A \oplus B$	$\overline{A \oplus B}$																																																																																																
<table border="1"> <tr> <th>A</th> <th>X</th> </tr> <tr> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> </tr> </table>	A	X	0	1	1	0	<table border="1"> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	B	A	X	0	0	0	0	1	0	1	0	0	1	1	1	<table border="1"> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	B	A	X	0	0	1	0	1	1	1	0	1	1	1	0	<table border="1"> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	0	<table border="1"> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </table>	B	A	X	0	0	1	0	1	0	1	0	0	1	1	0	<table border="1"> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	B	A	X	0	0	0	0	1	1	1	0	1	1	1	1	<table border="1"> <tr> <th>B</th> <th>A</th> <th>X</th> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </table>	B	A	X	0	0	1	0	1	0	1	0	0	1	1	1
A	X																																																																																																					
0	1																																																																																																					
1	0																																																																																																					
B	A	X																																																																																																				
0	0	0																																																																																																				
0	1	0																																																																																																				
1	0	0																																																																																																				
1	1	1																																																																																																				
B	A	X																																																																																																				
0	0	1																																																																																																				
0	1	1																																																																																																				
1	0	1																																																																																																				
1	1	0																																																																																																				
B	A	X																																																																																																				
0	0	0																																																																																																				
0	1	1																																																																																																				
1	0	1																																																																																																				
1	1	0																																																																																																				
B	A	X																																																																																																				
0	0	1																																																																																																				
0	1	0																																																																																																				
1	0	0																																																																																																				
1	1	0																																																																																																				
B	A	X																																																																																																				
0	0	0																																																																																																				
0	1	1																																																																																																				
1	0	1																																																																																																				
1	1	1																																																																																																				
B	A	X																																																																																																				
0	0	1																																																																																																				
0	1	0																																																																																																				
1	0	0																																																																																																				
1	1	1																																																																																																				

19

Conocimientos previos

- Álgebra de Boole

Ley commutativa de la suma para dos variables

$$A+B = B+A$$

Ley commutativa de la multiplicación para dos variables

$$AB = BA$$

Ley asociativa de la suma para tres variables

$$A + (B + C) = (A + B) + C$$

Ley asociativa de la multiplicación para tres variables

$$A(BC) = (AB)C$$

Ley distributiva para tres variables

$$A(B + C) = AB + AC$$

20

Conocimientos previos

- Álgebra de Boole (cont.)

1. $A + 0 = A$	5. $A + A = A$	9. $\bar{A} = A$
2. $A + 1 = 1$	6. $A + \bar{A} = 1$	10. $A + AB = A$
3. $A \cdot 0 = 0$	7. $A \cdot A = A$	11. $A + \bar{A}B = A + B$
4. $A \cdot 1 = A$	8. $A \cdot \bar{A} = 0$	12. $(A + B)(A + C) = A + BC$

¿ $\bar{A} + AB$?

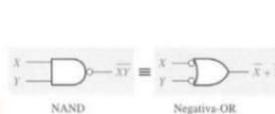
21

Conocimientos previos

- Teoremas de Demorgan

- Teorema 1

$$\overline{XY} = \overline{X} + \overline{Y}$$



Entradas		Salida	
X	Y	XY	$\bar{X} + \bar{Y}$
0	0	1	1
0	1	0	1
1	0	0	1
1	1	0	0

- Teorema 2

$$\overline{X + Y} = \overline{X} \cdot \overline{Y}$$



Entradas		Salida	
X	Y	$X + Y$	$\bar{X} \cdot \bar{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

En la práctica, los teoremas de Demorgan nos permiten expresar las funciones lógicas en términos de suma de productos o en términos de productos de suma, también podemos representar una función con un solo tipo de operador lógico (compuertas universales).

22

Conocimientos previos

- Mapas de Karnaugh

A	B	C	D	F1	F2
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	0
0	0	1	1	0	1
0	1	0	0	0	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	0	0
1	0	0	1	0	0
1	0	1	0	0	0
1	0	1	1	0	1
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

C D	A B	00	01	11	10
00	0	0	1	0	
01	0	0	1	0	
11	0	0	1	0	
10	0	0	1	0	

$$F = A \cdot B$$

Nos permiten obtener funciones lógicas simplificadas a partir de tablas de verdad, funciones sumatoria de minterm o producto de maxterm.

C D	A B	00	01	11	10
00	0	0	1	0	
01	0	1	0	0	
11	1	1	1	0	
10	0	1	1	0	

$$F = B \cdot C + A \cdot B \cdot D + A \cdot B \cdot \bar{D} + \bar{A} \cdot C \cdot D$$

23

Conocimientos previos

- Método de Quine Mccluskey

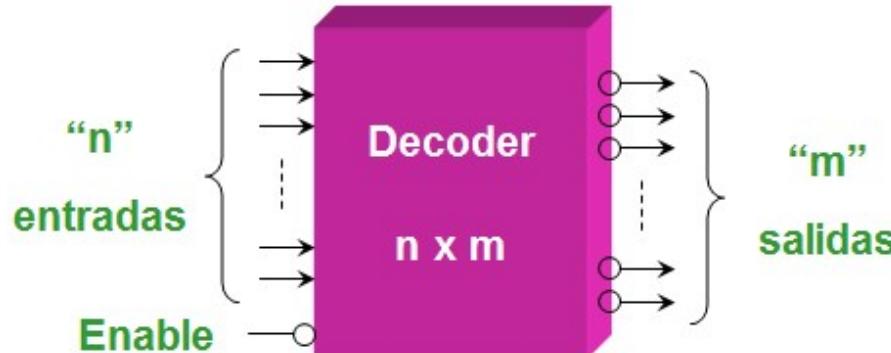
	0	1	2	3	5	7	8	10	12	13	15
$\bar{A} \bar{B}$	*	*	*	*							
$\bar{B} D$	*		*		*	*	*	*			
$\bar{A} D$		*		*	*	*	*				
$\bar{B} \bar{D}$			*	*	*	*					
$\bar{A} \bar{C} \bar{D}$					*				*		*
$\bar{A} B C$							X	X		X	X
Essential	X		X		X	X	X	(X)		X	(X)

- Método tabular similar a los mapas de Karnaugh
- Aplicable a funciones con cualquier número de variables (útil para funciones de a partir de 5 variables)
- Tedioso realizarlo a mano, pero puede ser automatizado (algoritmo en procesador)

24

Conocimientos previos

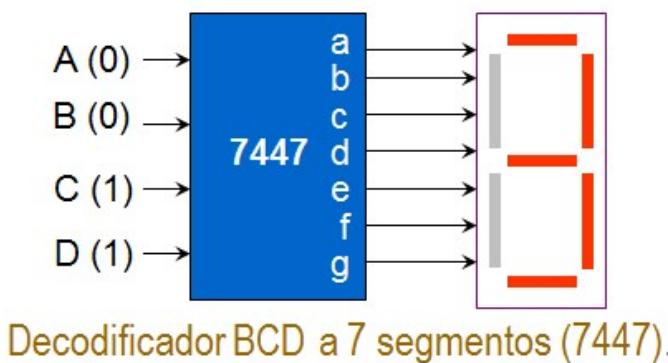
- Circuitos MSI: Decodificadores
 - Son circuitos lógicos que tienen “n” entradas y 2^n salidas.
 - Posee una tabla de decodificación



25

Conocimientos previos

- Circuitos MSI: Decodificadores (cont.)
 - Hay decodificadores BCD a display de siete segmentos

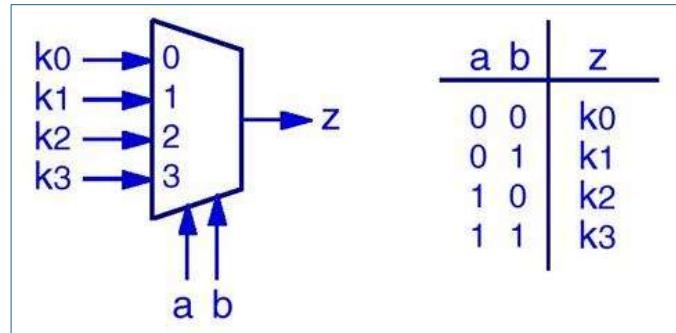


Decodificador BCD a 7 segmentos (7447).

26

Conocimientos previos

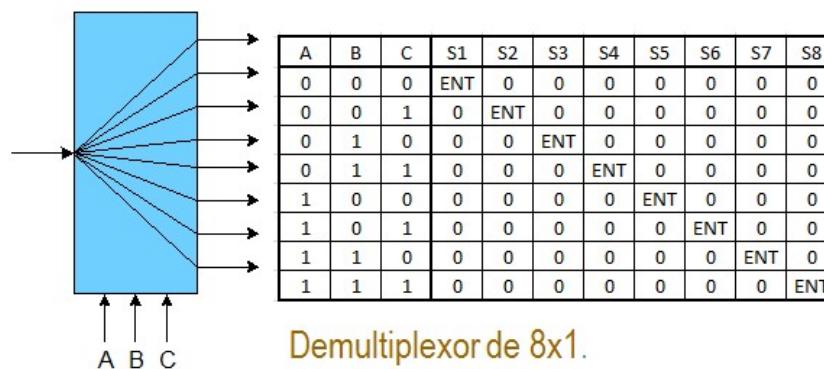
- Circuitos MSI: Multiplexores
 - Dispositivos selectores de datos



27

Conocimientos previos

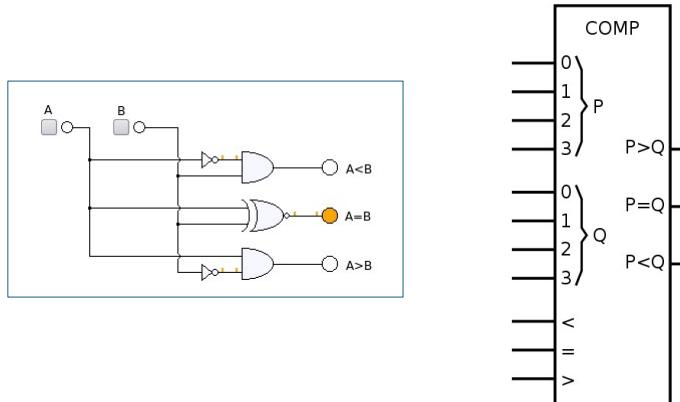
- Circuitos MSI: Demultiplexores
 - Dispositivos distribuidores de datos



28

Conocimientos previos

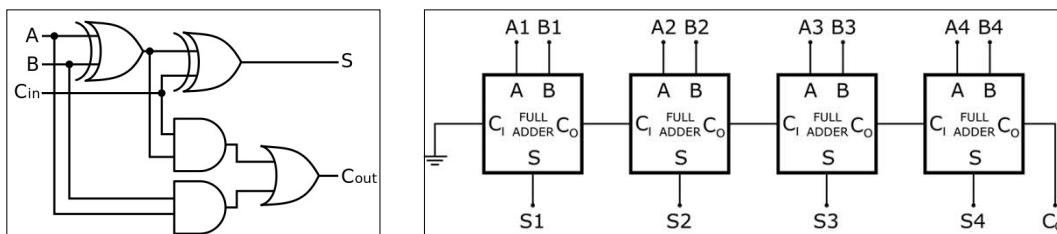
- Circuitos MSI: Comparadores de magnitud



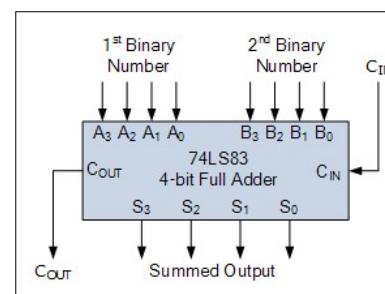
29

Conocimientos previos

- Circuitos MSI: Sumadores



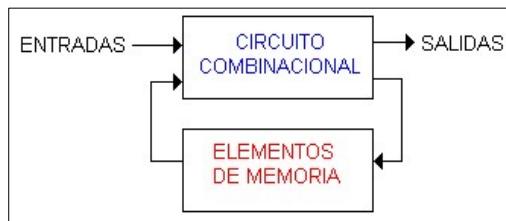
El carry (acarreo) es uno o cero, no otro valor



30

Conocimientos previos

- Sistemas Secuenciales

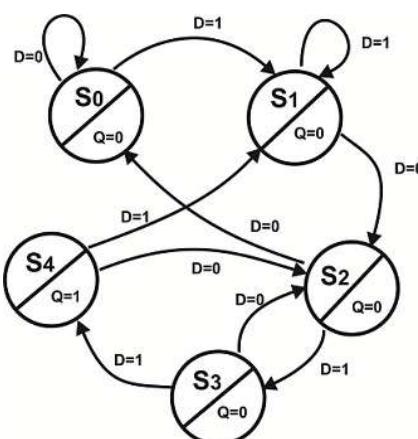


Name / Symbol	Characteristic (Truth Table)	State Diagram / Characteristic Equations	Excitation Table																																																								
SR	<table border="1"> <thead> <tr> <th>S</th><th>R</th><th>Q</th><th>Q_{next}</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>X</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>X</td></tr> </tbody> </table>	S	R	Q	Q _{next}	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1	0	X	1	1	1	X	<p>SR=00 or 01 SR=10 SR=01 SR=00 or 10</p> <p>$Q_{next} = S + R'Q$ $SR = 0$</p>	<table border="1"> <thead> <tr> <th>Q</th><th>Q_{next}</th><th>S</th><th>R</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>X</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>X</td><td>0</td></tr> </tbody> </table>	Q	Q _{next}	S	R	0	0	0	X	0	1	1	0	1	0	0	1	1	1	X	0
S	R	Q	Q _{next}																																																								
0	0	0	0																																																								
0	0	1	1																																																								
0	1	0	0																																																								
0	1	1	0																																																								
1	0	0	1																																																								
1	0	1	1																																																								
1	1	0	X																																																								
1	1	1	X																																																								
Q	Q _{next}	S	R																																																								
0	0	0	X																																																								
0	1	1	0																																																								
1	0	0	1																																																								
1	1	X	0																																																								
JK	<table border="1"> <thead> <tr> <th>J</th><th>K</th><th>Q</th><th>Q_{next}</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	J	K	Q	Q _{next}	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	0	<p>JK=00 or 01 JK=10 or 11 JK=01 or 11 JK=00 or 10</p> <p>$Q_{next} = J'K'Q + JK' + JKQ'$ $= J'K'Q + JK'Q + JKQ' + JKQ'$ $= K'Q'J' + JI + JQ'(K' + K)$ $= K'Q' + JQ'$</p>	<table border="1"> <thead> <tr> <th>Q</th><th>Q_{next}</th><th>J</th><th>K</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>X</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>X</td></tr> <tr><td>1</td><td>0</td><td>X</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>X</td><td>0</td></tr> </tbody> </table>	Q	Q _{next}	J	K	0	0	0	X	0	1	1	X	1	0	X	1	1	1	X	0
J	K	Q	Q _{next}																																																								
0	0	0	0																																																								
0	0	1	1																																																								
0	1	0	0																																																								
0	1	1	0																																																								
1	0	0	1																																																								
1	0	1	1																																																								
1	1	0	1																																																								
1	1	1	0																																																								
Q	Q _{next}	J	K																																																								
0	0	0	X																																																								
0	1	1	X																																																								
1	0	X	1																																																								
1	1	X	0																																																								
D	<table border="1"> <thead> <tr> <th>D</th><th>Q</th><th>Q_{next}</th></tr> </thead> <tbody> <tr><td>0</td><td>X</td><td>0</td></tr> <tr><td>1</td><td>X</td><td>1</td></tr> </tbody> </table>	D	Q	Q _{next}	0	X	0	1	X	1	<p>D=0 D=1 D=0 D=1</p> <p>$Q_{next} = D$</p>	<table border="1"> <thead> <tr> <th>Q</th><th>Q_{next}</th><th>D</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	Q	Q _{next}	D	0	0	0	0	1	1	1	0	0	1	1	1																																
D	Q	Q _{next}																																																									
0	X	0																																																									
1	X	1																																																									
Q	Q _{next}	D																																																									
0	0	0																																																									
0	1	1																																																									
1	0	0																																																									
1	1	1																																																									
T	<table border="1"> <thead> <tr> <th>T</th><th>Q</th><th>Q_{next}</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	T	Q	Q _{next}	0	0	0	0	1	1	1	0	1	1	1	0	<p>T=0 T=1 T=0 T=1</p> <p>$Q_{next} = TQ' + T'Q = T \oplus Q$</p>	<table border="1"> <thead> <tr> <th>Q</th><th>Q_{next}</th><th>T</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	Q	Q _{next}	T	0	0	0	0	1	1	1	0	1	1	1	0																										
T	Q	Q _{next}																																																									
0	0	0																																																									
0	1	1																																																									
1	0	1																																																									
1	1	0																																																									
Q	Q _{next}	T																																																									
0	0	0																																																									
0	1	1																																																									
1	0	1																																																									
1	1	0																																																									

31

Conocimientos previos

- Máquinas de estado (MEF)

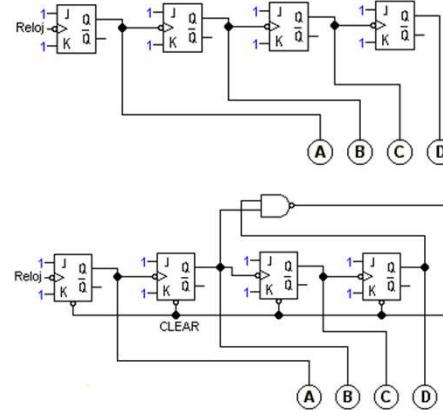
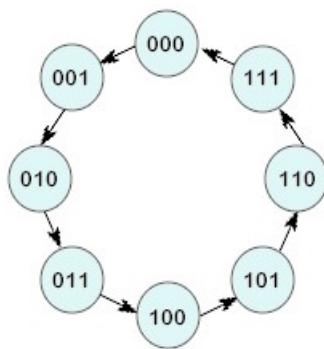


Nos permiten modelar el funcionamiento de sistemas tanto combinacional como secuencial

32

Conocimientos previos

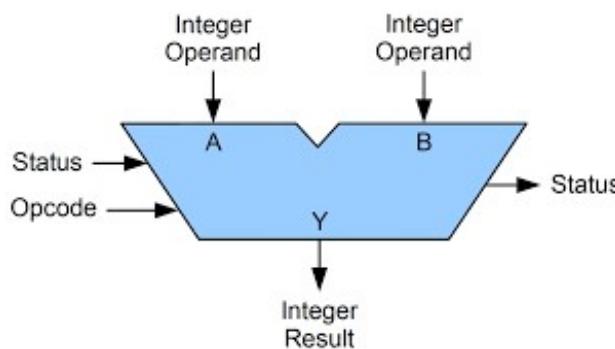
- Circuitos MSI: Contadores



33

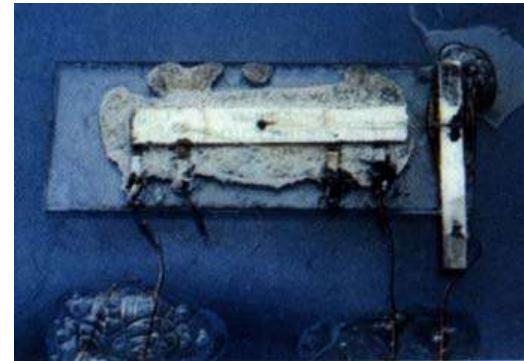
Unidad Aritmética Lógica

- Es el dispositivo esencial para el procesamiento de datos en el microprocesador.



34

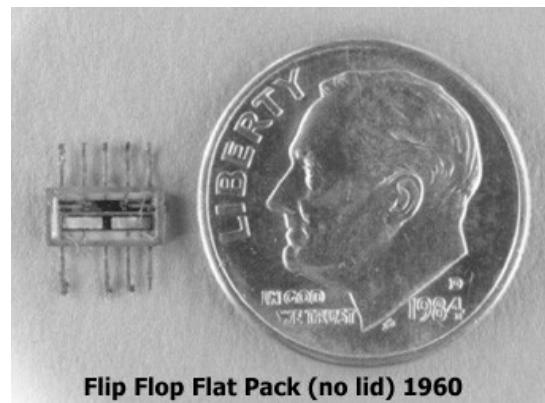
El Primer Circuito Integrado (1958)



Jack Kilby y Robert Noyce (Texas Instrument)

35

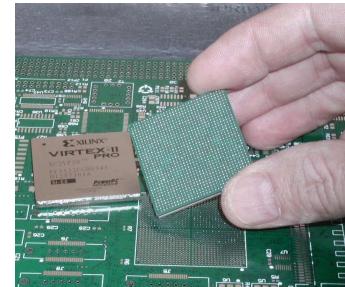
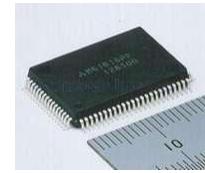
El Primer IC Comercial (1960)



SN502 Solid Circuit Flip Flop (USD\$450)

36

IC's hoy en día:



37

Conocimientos previos

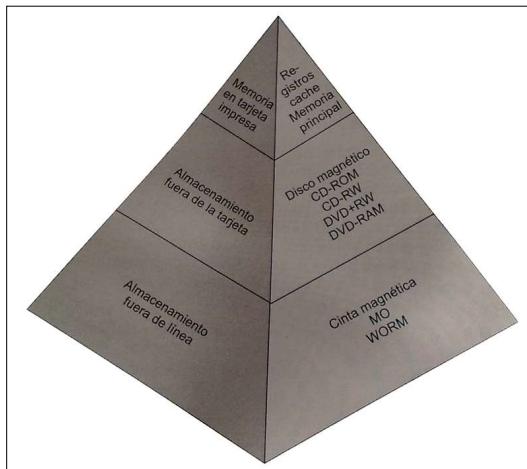
- Memorias
 - Dispositivos que almacenan información de manera ordenada
 - Concepto de “dirección única”



38

Conocimientos previos

- Jerarquía de memorias



- Registros
- Caché Nivel 1
- Caché Nivel 2
- Memoria Principal
- Caché de Disco Duro
- Disco Duro
- Discos Ópticos
- Cintas Tape Backup

39

Conocimientos previos

- Jerarquía de memorias

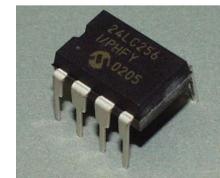
Relaciones:

- A menor tiempo de acceso, mayor coste por bit.
- A mayor capacidad, menor coste por bit
- A mayor capacidad, mayor tiempo de acceso

40

Conocimientos previos

- Tecnologías en memorias de estado sólido
 - ROM
 - PROM (OTP)
 - EPROM
 - EEPROM
- RAM
- SRAM
- DRAM

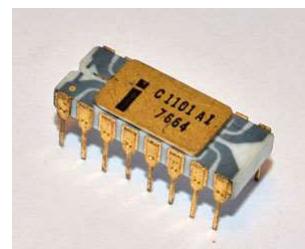


41

Conocimientos previos

Dispositivos de almacenamiento

- SRAM (John Schmidt 1964 – Fairchild Semiconductor)



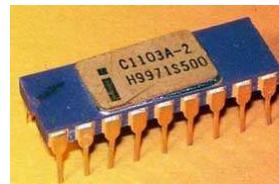
SRAM Intel 1101 (1969)

42

Conocimientos previos

Dispositivos de almacenamiento

- DRAM (Robert Dennart – 1968)



(Intel C1103 - 1970)

43

Conocimientos previos

Dispositivos de almacenamiento



Tape data storage

44

Conocimientos previos

Dispositivos de almacenamiento

- El disco duro (1953)

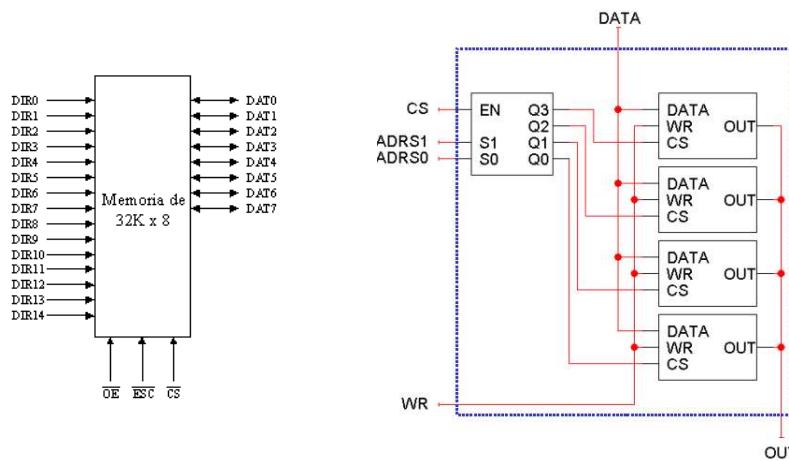


Disco duro 5MByte para IBM305 RAMAC (1956)

45

Conocimientos previos

- Lineas de datos, direcciones y de control en una memoria

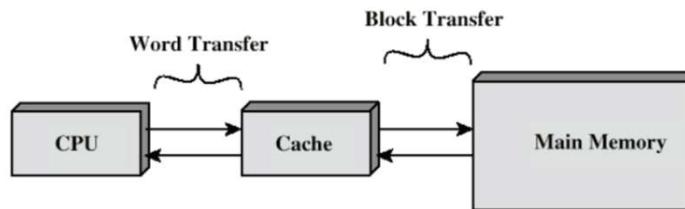


46

Conocimientos previos

Memoria CACHÉ

- Basado en memoria del tipo SRAM
- En pequeña cantidad
- Se encuentra entre la memoria principal y el µP
- Puede estar embebida en el mismo µP o de manera externa



47

Conocimientos previos

Funcionamiento de la Memoria CACHÉ

- El µP pide el contenido de una posición de memoria
- Revisa la caché para ver si contiene esa información
- Si está presente, lo obtiene de la caché
- Si no está presente, lee el bloque requerido desde la memoria principal hacia la caché.
- Luego la información es entregada desde la caché hacia el µP.
- La caché incluye etiquetas para identificar cuál bloque de la memoria principal está en una casilla de la caché.

48

Conocimientos previos

¿Porqué Memoria CACHE?

- Un sistema de memoria el cual almacena temporalmente información accedida frecuentemente y obtenida de dispositivos de almacenamiento con mayor tiempo de acceso.
- Con esto, se busca aminorar el tiempo de acceso, mientras la memoria CACHE es mayor, nuestra PC será mas rápida

49

Conocimientos previos

¿Necesitamos velocidad?

- Es posible implementar una PC con sólamente memoria del tipo SRAM
- Sería sumamente rápido
- No se necesitaría memoria CACHE
- El costo sería extremadamente exorbitante!

50

Conocimientos previos

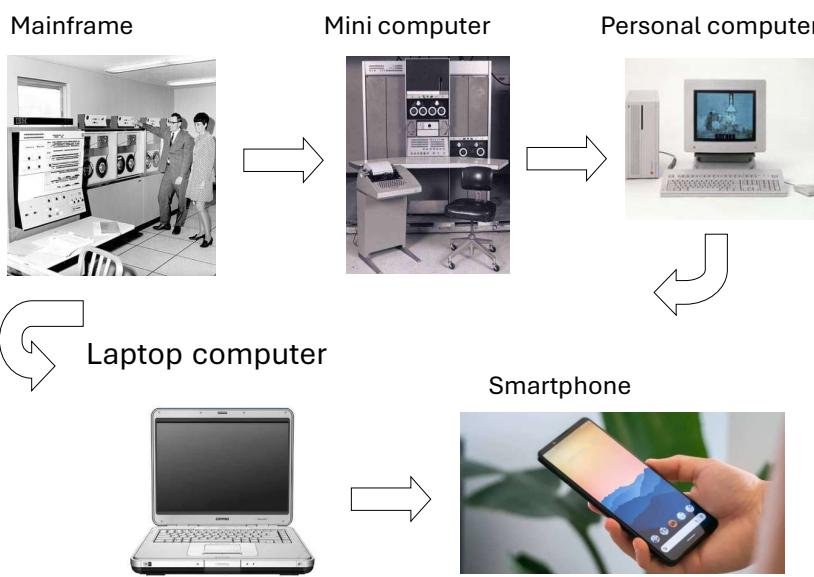
Segun Wikipedia:

- Una **computadora u ordenador** es un sistema digital con tecnología microelectrónica, capaz de procesar datos a partir de un grupo de instrucciones denominado programa. La estructura básica de una computadora incluye microprocesador (CPU), memoria y dispositivos de entrada/salida (E/S), junto a los buses que permiten la comunicación entre ellos. En resumen la computadora es una dualidad entre hardware (parte física) tales como: la pantalla, el teclado o el disco duro y software (parte lógica), que interactúan entre sí para una determinada función.

51

Conocimientos previos

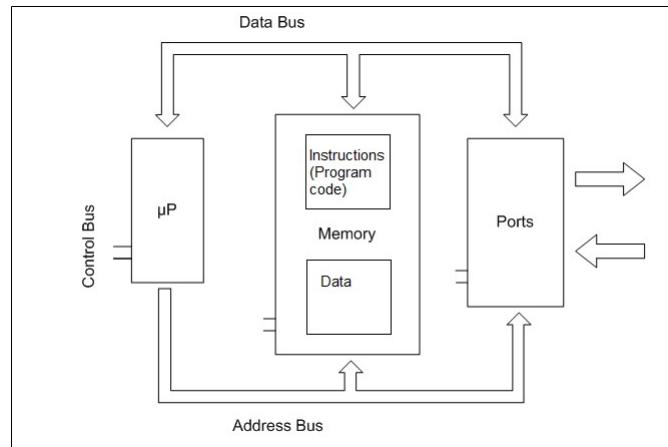
Evolución de la Computadora



52

Conocimientos previos

- Arquitectura de Computadores



53

Conocimientos previos

Arquitecturas: Harvard vs Von Neumann

- Von Neumann
 - Instrucciones y Datos en una sola memoria
 - Cuando la CPU se dirige a la memoria principal, primero saca la instrucción y después saca los datos necesarios para ejecutarla, esto retarda el funcionamiento de la CPU
- Harvard
 - Instrucciones en una memoria dedicada
 - Datos en una memoria dedicada
 - Cada memoria dispone de su respectivo bus, lo que permite, que la CPU pueda acceder de forma independiente y simultánea a la memoria de datos y a la de instrucciones

54

Conocimientos previos

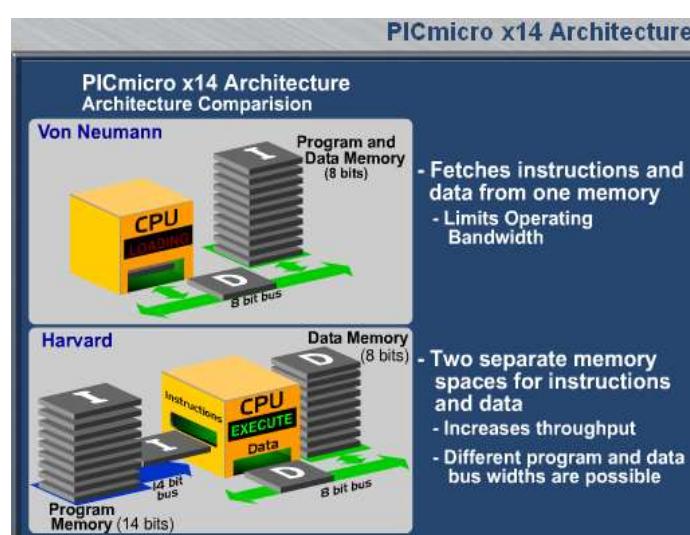
Neumann vs. Harvard

- Tipos de arquitecturas presentes en procesadores
- Harvard viene a ser una evolución de Neumann
 - Neumann -> Lo antiguo
 - Harvard -> Lo nuevo
- Neumann -> Tu PC
- Harvard -> Tu disp. Móvil

55

Conocimientos previos

Arquitecturas: Harvard vs Von Neumann



56

Conocimientos previos

CISC vs RISC

- **CISC (Complex Instruction Set Computer)**
 - Las instrucciones son de diferente longitud
 - El software resulta mucho más caro que el hardware.
 - El nivel del lenguaje era cada vez más complicado.
- **RISC (Reduced Instruction Set Computer)**
 - Las instrucciones son de un solo tamaño
 - Gran número de registros de uso general
 - Repertorio de instrucciones limitado y sencillo.
 - Énfasis en la optimización de la segmentación de instrucciones

57

Conocimientos previos

Persistencia de Neumann hasta hoy

- ¿Por qué se sigue usando Neumann en las PCs sabiendo que Harvard es mejor?
 - ¿Precio y fácil de fabricar?
 - ¿Uso más eficiente de la memoria?
 - ¿Compatibilidad?
 - ¿Qué es compatibilidad?
 - ¿A qué se debe la compatibilidad? (Causas y efectos)
 - ¿Sigue habiendo compatibilidad?
 - Motivos económicos por parte del fabricante

58

Conocimientos previos

Punto de quiebre: 20 de Mayo del 2024

- Microsoft anuncia la introducción de una nueva categoría de PCs con soporte a Windows Copilot. Denominados Copilot+ PC. (Microsoft, Mayo del 2024)
- Este anuncio incluye la cooperación de los principales fabricantes en el rubro de PCs: Acer, Asus, Dell, HP, Lenovo y Samsung.
- A parte de las nuevas funcionalidades de AI, anunciaron que se usarán procesadores Qualcomm Snapdragon X Elite y Snapdragon X Plus en esta nueva categoría.
- ¿Podrá significar esto un declive en el uso de procesadores Intel para computadoras que usan el sistema operativo de Microsoft?



59

Conocimientos previos

Sistema de Procesamiento de Datos

- Microprocesador (μ P)
 - Encargado de procesar instrucciones.
 - Se apoya en registros de trabajo.
- Memoria (Memory)
 - Encargado de almacenar datos.
- Puertos (Ports)
 - Interfase con el mundo exterior.

60

Conocimientos previos

Sistema de Procesamiento de Datos

- Bus de Datos (Data Bus)
 - Intercambio de información (datos e instrucciones) entre los componentes del sistema.
- Bus de Direcciones (Address Bus)
 - Le permite al µP elegir el recurso a trabajar (memoria o puerto).
- Bus de Control
 - Para que el µP pueda leer ó emitir eventos de lectura, escritura, estado, etc.

61

Conocimientos previos

Sistema de Procesamiento de Datos

- Cada recurso del sistema posee una dirección única el cual el microprocesador utiliza para realizar diversas operaciones (lectura, escritura, etc)

62

Conocimientos previos

Ley de Moore

“Aproximadamente cada dos años se duplica el número de transistores en un circuito integrado”

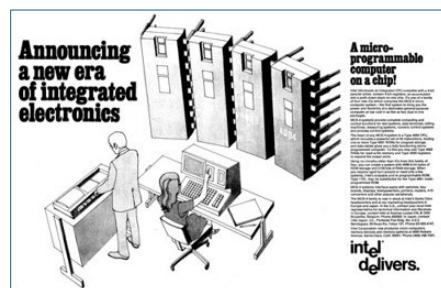


63

Conocimientos previos

Intel 4004

- Primer microprocesador microelectrónico (1971)



64

Conocimientos previos

¿Primer procesador microelectrónico?

- Hasta hace poco se indicaba al Intel 4004 de 1971 como el primer microprocesador de la historia (Intel, 2012)
- Sin embargo, un documento desclasificado por la milicia de EEUU en 1998 (Hackaday, Enero del 2024) detallan un desarrollo de un microprocesador con datación anterior frente al i4004, se trata del MP944 desarrollado por los hermanos Ray y Bill Holt en las instalaciones de Garrett AiResearch. (Wikipedia, 2024)
- El MP944 se concibió y desarrolló como la computadora de vuelo del avión de combate F14 Tomcat entre los años 1968-1970. (Ray Holt – First Microprocessor, 1997)

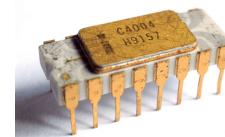


Figura 1: Procesador i4004.
Recuperado de Wikipedia



Figura 2: El MP944.
Recuperado de Hackaday

65

Conocimientos previos

Historia de los procesadores modernos

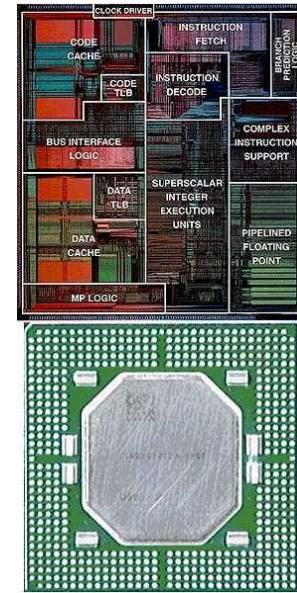
- IBM – Muy ligado a apoyo de procesos del gobierno de EEUU
 - Desarrollo de la PC (Intel 8086)
- Intel x86
- Motorola 68HC11
- MOS 6502
 - Apple (I y II)
 - Commodore 64
 - Atari
 - Nintendo etc
- MIPS (RISC)
- Compañía ARM (Advanced RISC Machines)
 - Desarrolla arquitectura (no produce circuitos) ARM7, ARM9, ARM11
 - Brinda licencias: Qualcomm (Snapdragon), Apple (A12, A13, A14, M1), Samsung (Exynos), Mediatek, Huawei (Kirin)
- RISC-V (Open hardware)

66

Conocimientos previos

Secciones internas del μP

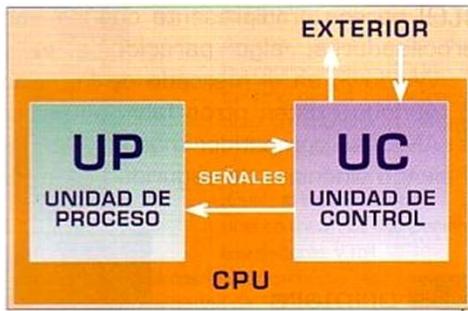
- Como podemos apreciar en la imagen, el interior del microprocesador está conformado por muchos elementos.
- Es como una pequeña “ciudad”.



67

Conocimientos previos

El interior del CPU



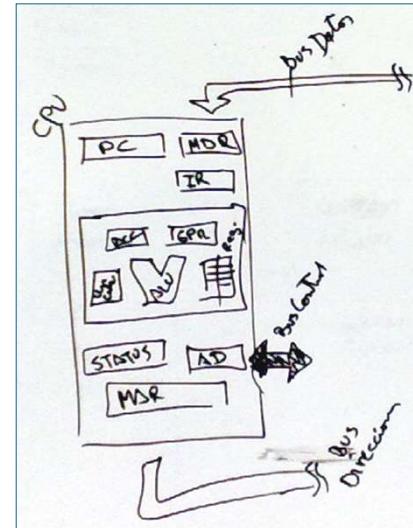
- Lo conforman dos componentes lógicos con funciones específicas.
- Al igual que en un automóvil, hay un motor y un conductor que lo controla.
 - El motor es la UP
 - El conductor es la UC

68

Conocimientos previos

¿Cómo se ejecuta una instrucción?

- Si has aprendido un lenguaje de alto nivel (C++, Java, etc), podrías pensar que la ejecución de las instrucciones es una cosa sencilla, esto no es cierto!.
- Actualmente, una sentencia en un lenguaje de alto nivel puede conllevar a realizar cientos de operaciones en la CPU.



69

Conocimientos previos

Instruction pipeline

- Técnica empleada en el diseño de microprocesadores para aumentar la eficiencia en el procesamiento de las instrucciones.
- En un ciclo de reloj puede estarse ejecutando varias instrucciones en diferentes etapas de su procesamiento.

Instr. No.	Pipeline Stage						
	IF	ID	EX	MEM	WB		
1							
2				EX	MEM	WB	
3			IF	ID	EX	MEM	WB
4				IF	ID	EX	MEM
5					IF	ID	EX
Clock Cycle	1	2	3	4	5	6	7

70

Conocimientos previos

IBM PC (1981)



71

Conocimientos previos

¿Procesador Super-Escalares?

- Lo esencial del enfoque superescalar es su habilidad para ejecutar instrucciones de manera independiente en diferentes cauces. El concepto puede llevarse más lejos, permitiendo que las instrucciones se ejecuten en un orden diferente al del programa.
- Hay múltiples unidades funcionales, cada una de las cuales está implementada como un cauce segmentado, que admiten la ejecución en paralelo de varias instrucciones

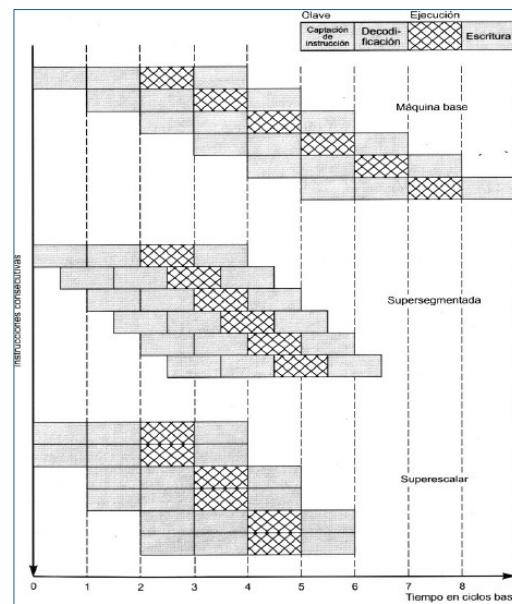
Fuente: <http://informatica.utm.cl/~hlatorre/Arq-Comp/Introduccion/Paralelo-Segment.pdf>

72

Conocimientos previos

Super-Escalares vs Super-Segmentados

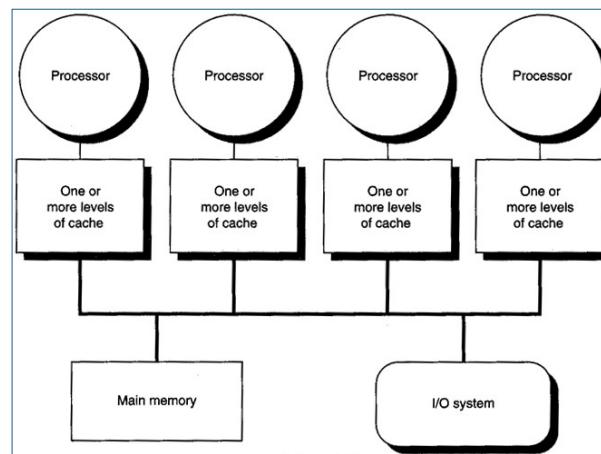
- La supersegmentación aprovecha el hecho de que muchas etapas del cauce realizan tareas que requieren menos de la mitad de un ciclo de reloj
- La implementación superescalar es capaz de ejecutar en paralelo dos instrucciones en cada etapa



73

Conocimientos previos

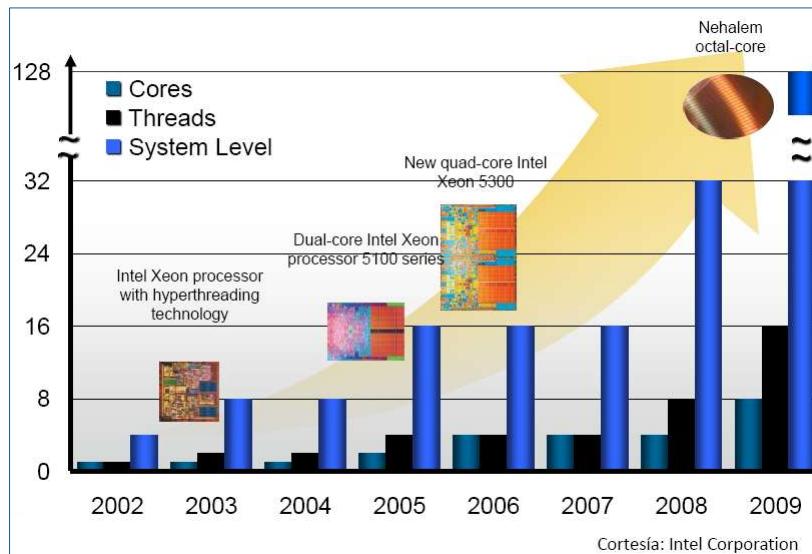
Sistemas Multi-procesadores (multicore)



74

Conocimientos previos

Avances de la Tecnología Multinúcleo



75

Conocimientos previos

Lenguaje Assembler

- El lenguaje ensamblador es el lenguaje de programación utilizado para escribir programas informáticos de bajo nivel, y constituye la representación más directa del "Código máquina" específico para cada arquitectura de computadoras legible por un programador.

Fuente: https://www.ecured.cu/Lenguaje_ensamblador

76

Conocimientos previos

Assembler: ¿Es difícil?

- Existe un prejuicio entre los diseñadores de software y programadores: Assembler es muy difícil.
- Al igual que todo lenguaje de programación, existe una curva de aprendizaje. El Assembler al ser un lenguaje de bajo nivel, es necesario conocer la arquitectura del procesador en donde va a correr el programa.
- Es esencial aprender a desarrollar algoritmos previamente (ej. Diagramas de Flujo)

77

Conocimientos previos

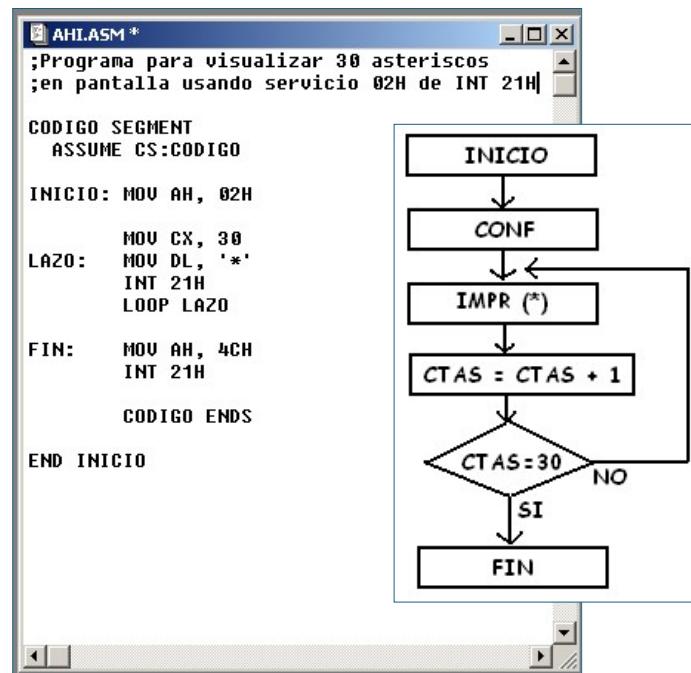
- Los programas hechos en lenguaje ensamblador, al ser programado directamente sobre [Hardware](#), son generalmente más rápidos y consumen menos recursos del sistema (memoria [RAM](#) y [ROM](#)). Al programar cuidadosamente en lenguaje ensamblador se pueden crear programas que se ejecutan más rápidamente y ocupan menos espacio que con lenguajes de alto nivel.

78

39

Macro Assembler (Intel)

- Ejemplo de programa en MASM: Imprimir treinta veces el carácter “*” en pantalla



79

Sistemas Embebidos

- Definición

El término “sistema embebido” hace referencia a todo circuito electrónico digital capaz de realizar operaciones de computación, generalmente en tiempo real, que sirven para cumplir una tarea específica en un producto.

Fuente: Sergio Salas. (2015). Todo sobre Sistemas Embebidos. Lima, Perú:
Editorial Universidad Peruana de Ciencias Aplicadas.

80

Sistemas Embebidos

- La arquitectura de un sistema embedded es similar a la de un computador pero orientado a aplicaciones específicas.



81

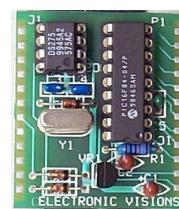
Sistemas Embebidos

Sistemas embedded usando microcontroladores

¿Porque usar un microcontrolador?



“Un microcontrolador es un circuito integrado que incluye todos (o casi) los componentes necesarios para tener un sistema de control completo.”



82

Sistemas Embebidos

Sistemas embedded usando microcontroladores



¿Porque usar un microcontrolador?

Debido a la simplicidad de diseño (pocos componentes externos) y de programación (RISC), es posible diseñar sistemas embedded de una manera rápida.

Las herramientas de desarrollo para microcontroladores nos permite depurar nuestro código en forma rápida y eficiente

83

Sistemas Embebidos

Sistemas embedded usando microcontroladores



No. Rank	1990 Rank	1991 Rank	1992 Rank	1993 Rank	1994 Rank	1995/96 Rank	1997-00 Rank
1	Motorola	Motorola	Motorola	Motorola	Motorola	Motorola	Motorola
2	Mitsubishi	Mitsubishi	Mitsubishi	Mitsubishi	Mitsubishi	Mitsubishi	Microchip
3	NEC	NEC	Intel	NEC	NEC	SGS-Thomson	NEC
4	Intel	Intel	NEC	Hitachi	Philips	NEC	Hitachi
5	Hitachi	Hitachi	Philips	Philips	Intel	Microchip	ST-Micro
6	Philips	Philips	Hitachi	Intel	Zilog	Philips	Infinion
7	Matsushita	Matsushita	Matsushita	SGS	Zilog	Zilog	Mitsubishi
8	National	SGS-Thomson	SGS	Microchip	SGS	Hitachi	Philips
9	Siemens	Siemens	National	Matsushita	Fujitsu	Fujitsu	Toshiba
10	TI	TI	TI	Toshiba	Hitachi	Intel	Atmel
11	Sharp	National	Zilog	National	Toshiba	Siemens	Zilog
12	Oki	Toshiba	Toshiba	Zilog	National	Toshiba	Fujitsu
13	Toshiba	Sony	Siemens	TI	TI	Matsushita	Matsushita
14	SGS-Thomson	Sharp	Microchip	Siemens	Ricoh	TI	Realtek
15	Zilog	Oki	Sharp	Sharp	Fujitsu	National	Samsung
16	Matra MHS	Zilog	Sanyo	Oki	Siemens	Temic	National
17	Sony	Microchip	Matra MHS	Sony	Sharp	Sanyo	Sanyo
18	Fujitsu	Matra MHS	Sony	Sanyo	Oki	Ricoh	Elan
19	AMD	Fujitsu	Oki	Fujitsu	Sony	Oki	TI
20	Microchip	Sanyo	Fujitsu	AMD	Temic	Sharp	Sony

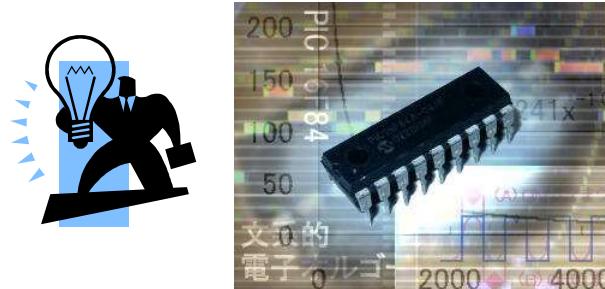
Ranking de fabricantes de microcontroladores

84

Sistemas Embebidos

Sistemas embedded usando microcontroladores

El microcontrolador es la solución en un chip!

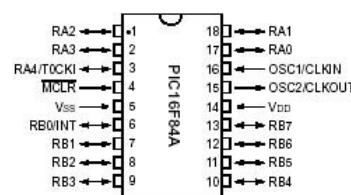


85

Sistemas Embebidos

Sistemas embedded usando microcontroladores

El Microcontrolador PIC



- Diversos microcontroladores de acuerdo a la necesidad de la aplicación
- Arquitectura RISC-Harvard
- Diversos periféricos internos
- Amplia gama de herramientas de diseño

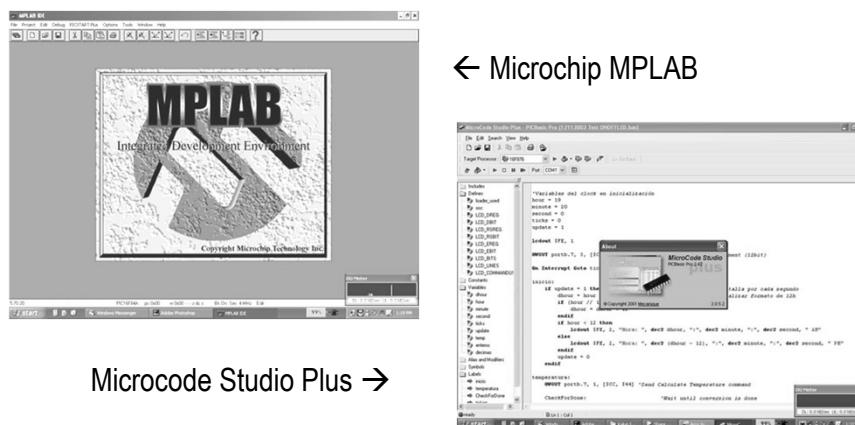


86

Sistemas Embebidos

Sistemas embedded usando microcontroladores

Herramientas de Desarrollo para Microcontroladores PIC



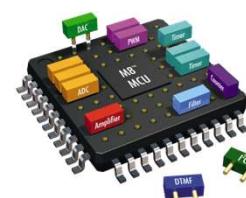
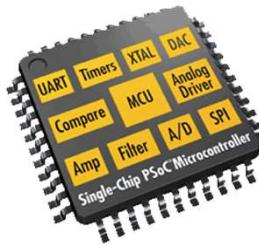
87

Sistemas Embebidos

Nuevas Tecnologías: El PSoC

- ¿Qué es el PSoC?

- PSoC = Programmable System-on-Chip
- Es un arreglo configurable de señales mixtas con un microcontrolador integrado.
- Creación de un chip “a la medida”.



88

Sistemas Embebidos

Especificaciones de un PSoC

- Cypress Enhanced Analog (CEA)
 - Rail-to-rail inputs
 - Low opamp input offset (5 mV)
 - Low opamp noise (80 nV/rtHz) and low ground path noise
 - 8 selectable analog reference points
 - Low power comparator (< 15 uA)
 - Differential inputs on opamps
 - PGA gain up to 48x
 - Three opamp instrumentation amp topology

89

Sistemas Embebidos

Especificaciones de un PSoC

- Embedded M8C Microprocessor Core
 - Programmable processor speeds
 - Up to 24 MHz (4 MIPS) Operation at 5V
 - Up to 12 MHz Operation at 3.3V
 - Harvard architecture
 - Used in Cypress USB products
- Single-cell (1.2V to start) Operation at up to 24MHz
 - with Built-in Voltage Pump and three passive components
- Internal system supervisor for PSoC
 - Eight-level Low Voltage Detection/Alert
- 2.5% Accurate Oscillator with no ext. components
 - PLL for precise time-base with inexpensive watch crystal
- Flexible Sleep Modes, as low as 3.0 μ A in Standby

90

Sistemas Embebidos

Especificaciones de un PSoC

- 16 KBytes Flash Program Memory
- EEPROM Emulation in Flash
- 256 Bytes SRAM
 - User defined stack length
- Built-In Multiply-Accumulate Hardware (MAC)
 - 8 X 8 Multiply, 32-bit Accumulate
 - Answer available immediately on next instruction cycle
- Four Memory Protection Modes
 - Allows factory or field upgrade on individual 64-byte blocks
 - From one block up to the entire Flash memory protectable
 - Robust read/write protection algorithm for added security
- In-System Serial Programmable (ISSP™)
 - Supports BIST or production test/calibration re-programming
 - Recommended that ISSP interface be designed into PCB

91

Sistemas Embebidos

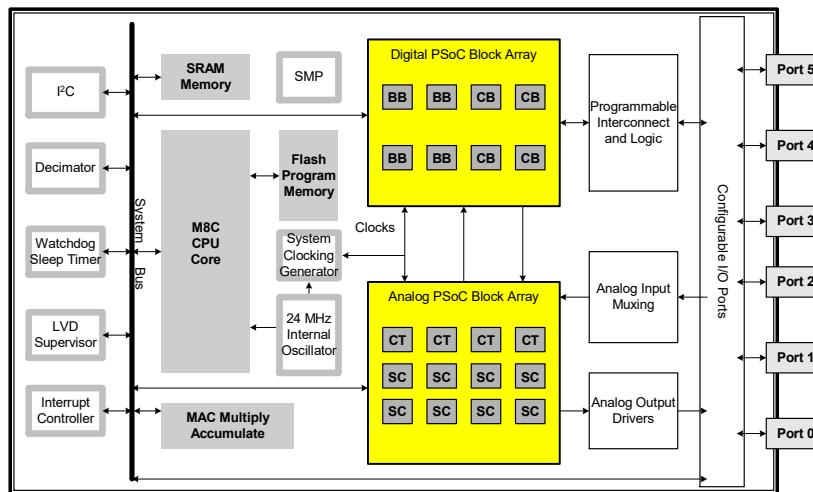
Especificaciones de un PSoC

- Configurable I/O Pins
 - Every pin can source 10mA and sink 25mA
 - Integrated/selectable pull-up and pull-down resistors
 - Selectable as interrupt source on either edge/change in state
 - Analog input with Schmitt trigger disabled
 - Disconnect logical inputs for lower digital feedback noise
 - Reduced noise level at nominal logic threshold
 - Slope controlled strong output
 - Reduced current transients, slower rise times for reduced radiated emissions
 - Open Drain and Open Source outputs (actually open-drain N-channel and open-drain P-channel)
- 8 Muxable Analog Inputs (except 8-pin device)
- 4 Analog Outputs each w/ 40mA drive
- 4 direct input analog lines (except 8-pin device)

92

Sistemas Embebidos

Diagrama de bloques de un PSoC



93

Sistemas Embebidos

Disminución de costos usando PSoC

Solución Tradicional

8-bit Micro	\$2.00
Crystal + Caps	\$0.57
Filters	\$0.30
Amps	\$0.20
Speaker Driver	\$0.15
LED Drivers	\$0.05
Circuit Board	\$1.20
Assembly	\$1.60

Costo = \$6.07

Solución de Cypress

PSoC Micro.	\$2.50
Circuit Board	\$0.90
Assembly	\$1.40

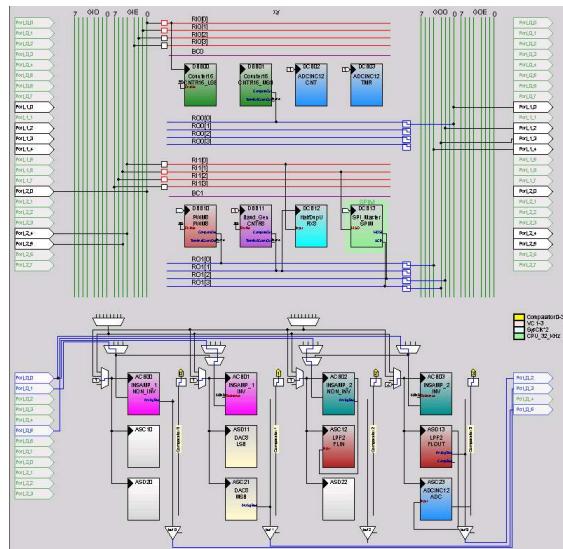


Costo = \$4.80

94

Sistemas Embebidos

Interconexión interna de un PSoC



- Se define la conexión entre los puertos y los bloques funcionales
- Se define las conexiones entre bloques funcionales
- Se define las rutas de reloj
- Posibilidad de cambiar las rutas dinámicamente

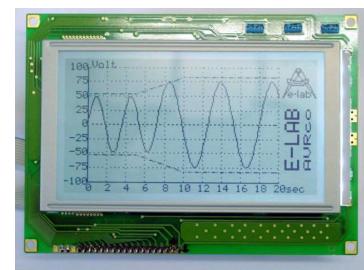
95

Sistemas Embebidos

Aplicaciones sobre sistemas embedded



- Sistemas de seguridad programables con interfase de usuario
- Monitoreo remoto de variables físicas
- Estaciones meteorológicas
- Edificios inteligentes (domótica)
- Servidores web embedados (embedded webservers)
- etc



96

Sistemas Embebidos

- Proceso de diseño con microcontroladores
 - Hardware prototipo
 - Algoritmo
 - Codificación
 - Simulación / Pruebas

97

Fin de la sesión

98