
1123021 Microcontroladores 15-P

Eduardo Rodríguez Martínez
Departamento de Electrónica
División de Ciencias Básicas e Ingeniería
Universidad Autónoma Metropolitana
Unidad Azcapotzalco

Email: erm@correo.azc.uam.mx
Sitio Web: <http://kali.azc.uam.mx/erm/>

Presentación

▷ Presentación

Metas y Objetivos

Contenido

Bibliografía

Tema 1

Tema 2

- **Inicio del curso:** 04 de Mayo del 2015
- **Fin del curso:** 24 de Julio del 2015
- **Grupo:** CEL01
- **Salón de clase:** F303
- **Días y hora de clase:** Lunes, y Miercoles de 11:30 a 13:00, y Jueves de 13:00 a 16:00 hrs
- **Asesorías:** Lunes, Miercoles y Viernes de 10:00 a 11:30

Presentación

▷ Presentación

Metas y Objetivos

Contenido

Bibliografía

Tema 1

Tema 2

- **Inicio del curso:** 04 de Mayo del 2015
- **Fin del curso:** 24 de Julio del 2015
- **Grupo:** CEL01
- **Salón de clase:** F303
- **Días y hora de clase:** Lunes, y Miercoles de 11:30 a 13:00, y Jueves de 13:00 a 16:00 hrs
- **Asesorías:** Lunes, Miercoles y Viernes de 10:00 a 11:30
- **Evaluación:** 30 % Exámenes, 70 % Trabajo en laboratorio.

Presentación

▷ Presentación

Metas y Objetivos

Contenido

Bibliografía

Tema 1

Tema 2

- **Inicio del curso:** 04 de Mayo del 2015
- **Fin del curso:** 24 de Julio del 2015
- **Grupo:** CEL01
- **Salón de clase:** F303
- **Días y hora de clase:** Lunes, y Miercoles de 11:30 a 13:00, y Jueves de 13:00 a 16:00 hrs
- **Asesorías:** Lunes, Miercoles y Viernes de 10:00 a 11:30
- **Evaluación:** 30 % Exámenes, 70 % Trabajo en laboratorio.
 - Tres exámenes parciales: 21 de Mayo, 18 de Junio, y 02 de Julio.
 - Evaluación global: 20 de Julio (Exención si el promedio de los globales es aprobatorio).

Presentación

▷ Presentación

Metas y Objetivos

Contenido

Bibliografía

Tema 1

Tema 2

- **Inicio del curso:** 04 de Mayo del 2015
- **Fin del curso:** 24 de Julio del 2015
- **Grupo:** CEL01
- **Salón de clase:** F303
- **Días y hora de clase:** Lunes, y Miercoles de 11:30 a 13:00, y Jueves de 13:00 a 16:00 hrs
- **Asesorías:** Lunes, Miercoles y Viernes de 10:00 a 11:30
- **Evaluación:** 30 % Exámenes, 70 % Trabajo en laboratorio.
 - Tres exámenes parciales: 21 de Mayo, 18 de Junio, y 02 de Julio.
 - Evaluación global: 20 de Julio (Exención si el promedio de los globales es aprobatorio).
 - Ocho sesiones de laboratorio + Proyecto final

Metas y Objetivos

Presentación

Metas y

► Objetivos

Contenido

Bibliografía

Tema 1

Tema 2

- Diferenciar un microcontrolador de un microprocesador.
- Identificar el campo de aplicación de los microcontroladores.
- Familiarizarse con el concepto de compilación cruzada.
- Construir aplicaciones en lenguaje ensamblador y lenguaje C.
- Dominar la configuración y el uso de los principales periféricos en un microcontrolador.
- Diseñar aplicaciones que usen eficientemente los recursos de un microcontrolador.

Contenido

Presentación

Metas y Objetivos

▷ Contenido

Bibliografía

Tema 1

Tema 2

- | | |
|---|-------|
| 1. Introducción a los microcontroladores | 3 hrs |
| 2. Programación en lenguaje ensamblador y de alto nivel | 3 hrs |
| 3. Introducción a periféricos | 6 hrs |
| 4. Uso de temporizadores | 3 hrs |
| 5. Uso de convertidores A/D y D/A | 3 hrs |
| 6. Manejo de interrupciones | 3 hrs |
| 7. Manejo del sistema de modulación por ancho de pulso | 3 hrs |
| 8. Interfaz de comunicación serie | 3 hrs |

Bibliografía

Presentación

Metas y Objetivos

Contenido

▷ Bibliografía

Tema 1

Tema 2

- John H. Davies, "MSP430 Microcontroller basics", Editorial Newnes-Elsevier, 2008.
- "MSP430x2xx Family: User's Guide", Ref. SLAU144J, Texas Instruments, 2004.

Tema 1 - Introducción a los microcontroladores - (3.0 hrs)

Presentación

Tema 1

▷ Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

MSP430G2xxx

Instrucciones

Tema 2

1. **Ámbito de aplicación de los microcontroladores**
2. Comparación de microcontroladores frente a microprocesadores.
3. Evolución de los microcontroladores.
4. Familias de microcontroladores.

Tema 1 - Introducción a los microcontroladores - (3.0 hrs)

Presentación

Tema 1

▷ Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

MSP430G2xxx

Instrucciones

Tema 2

1. **Ámbito de aplicación de los microcontroladores**
2. **Comparación de microcontroladores frente a microprocesadores.**
3. Evolución de los microcontroladores.
4. Familias de microcontroladores.

Tema 1 - Introducción a los microcontroladores - (3.0 hrs)

Presentación

Tema 1

▷ Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

MSP430G2xxx

Instrucciones

Tema 2

1. **Ámbito de aplicación de los microcontroladores**
2. **Comparación de microcontroladores frente a microprocesadores.**
3. **Evolución de los microcontroladores.**
4. **Familias de microcontroladores.**

Tema 1 - Introducción a los microcontroladores - (3.0 hrs)

Presentación

Tema 1

▷ Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

MSP430G2xxx

Instrucciones

Tema 2

1. Ámbito de aplicación de los microcontroladores
2. Comparación de microcontroladores frente a microprocesadores.
3. Evolución de los microcontroladores.
4. Familias de microcontroladores.

Ámbito de aplicación de los microcontroladores

Presentación

Tema 1

Contenido

▷ Aplicación

μ P vs. μ C

Evolución

Familias

MSP430G2xxx

Instrucciones

Tema 2



Ámbito de aplicación de los microcontroladores

Presentación

Tema 1

Contenido

▷ Aplicación

μ P vs. μ C

Evolución

Familias

MSP430G2xxx

Instrucciones

Tema 2



Ámbito de aplicación de los microcontroladores

Presentación

Tema 1

Contenido

▷ Aplicación

μ P vs. μ C

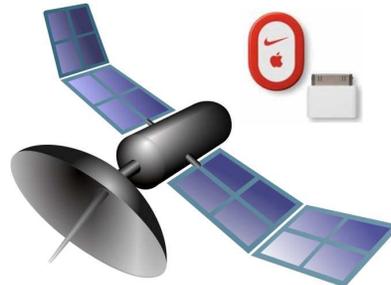
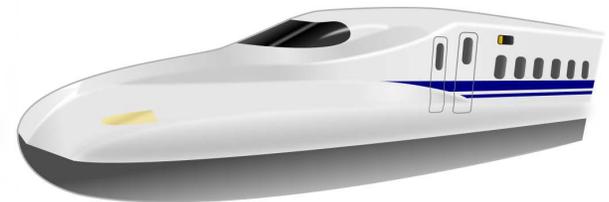
Evolución

Familias

MSP430G2xxx

Instrucciones

Tema 2



Ámbito de aplicación de los microcontroladores

Presentación

Tema 1

Contenido

▷ Aplicación

μ P vs. μ C

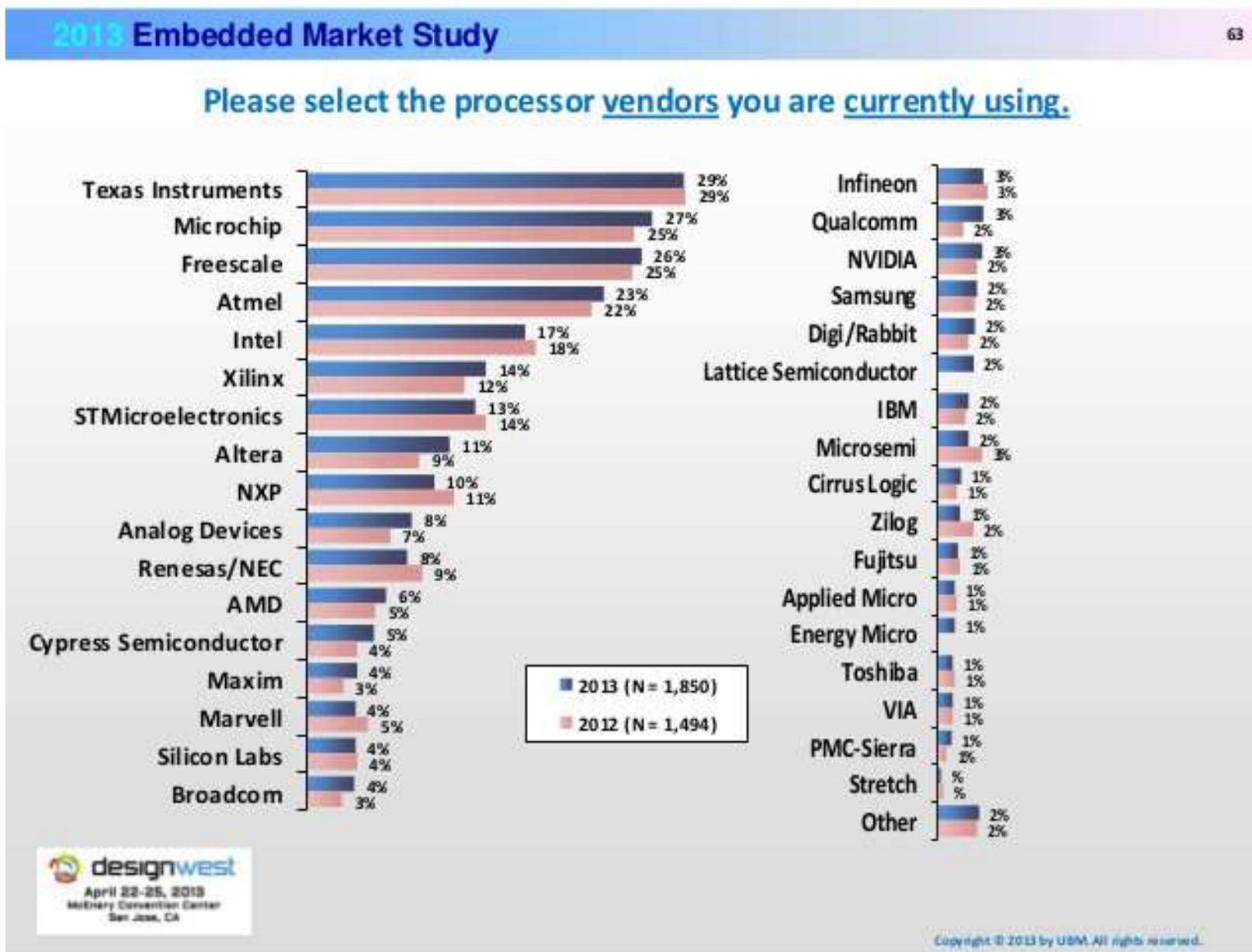
Evolución

Familias

MSP430G2xxx

Instrucciones

Tema 2



Ámbito de aplicación de los microcontroladores

Presentación

Tema 1

Contenido

▷ Aplicación

μ P vs. μ C

Evolución

Familias

MSP430G2xxx

Instrucciones

Tema 2



Ámbito de aplicación de los microcontroladores

Presentación

Tema 1

Contenido

▷ Aplicación

μ P vs. μ C

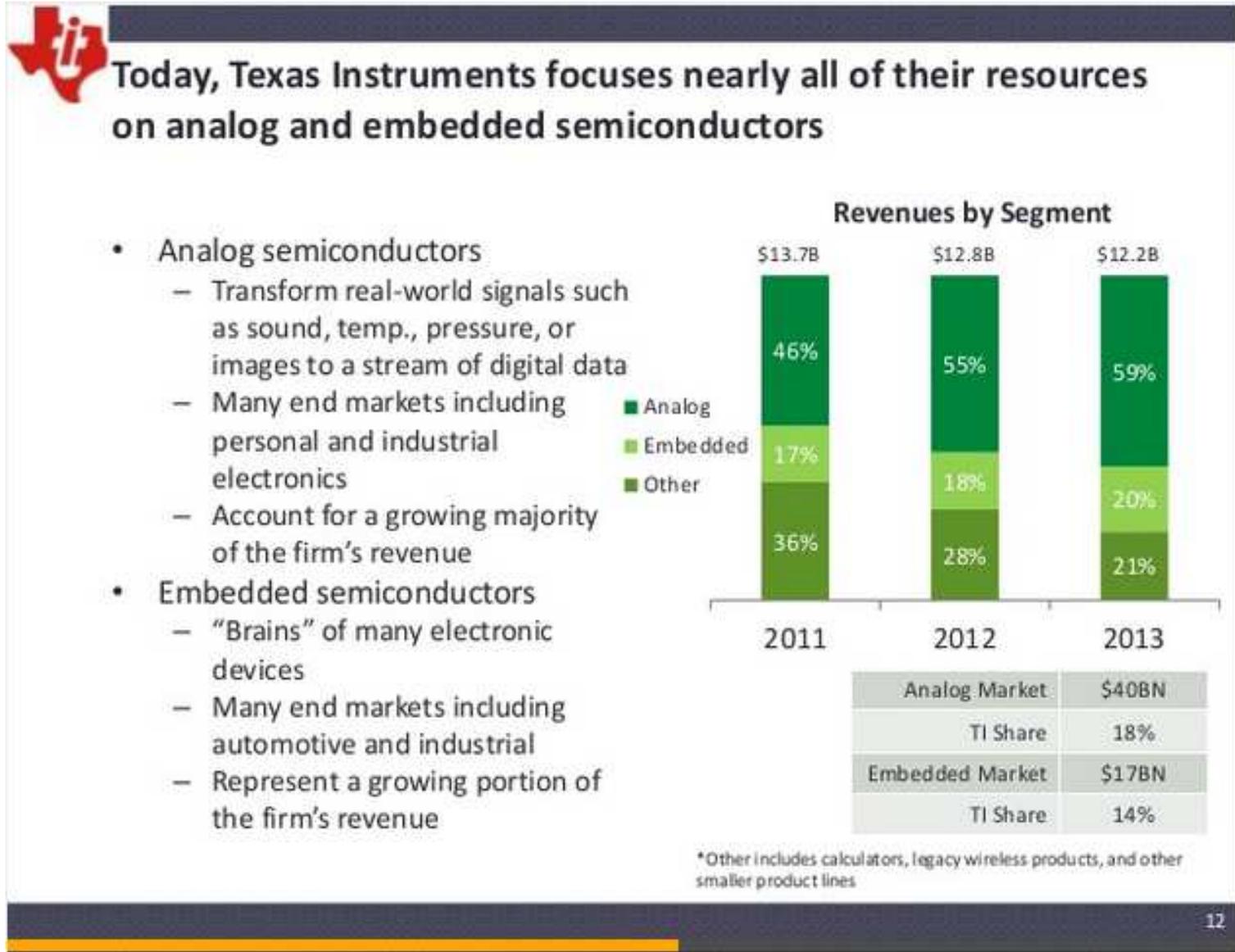
Evolución

Familias

MSP430G2xxx

Instrucciones

Tema 2



Comparación de microcontroladores frente a microprocesadores

Presentación

Tema 1

Contenido

Aplicación

▷ μP vs. μC

Evolución

Familias

MSP430G2xxx

Instrucciones

Tema 2



Microcontrolador

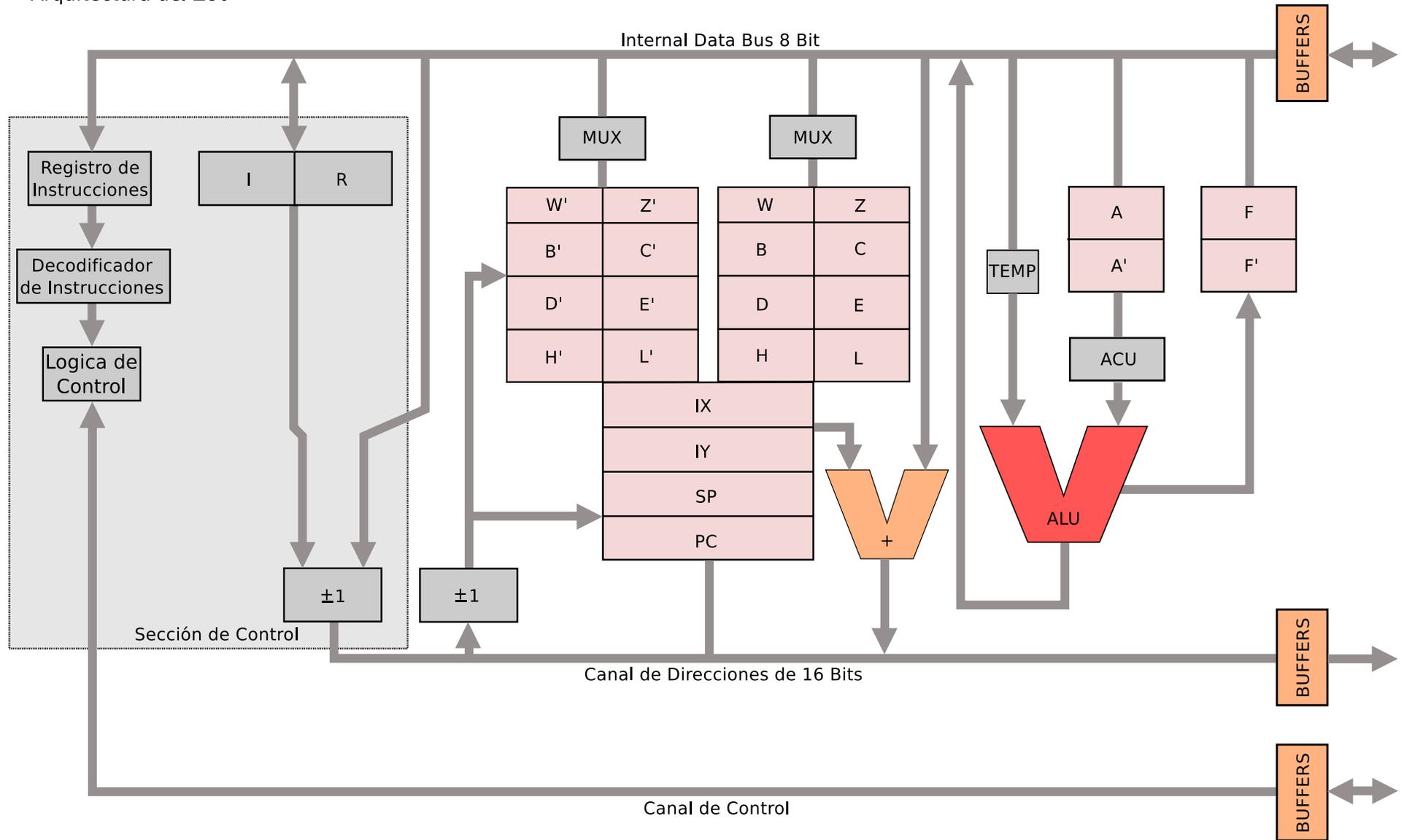
VS



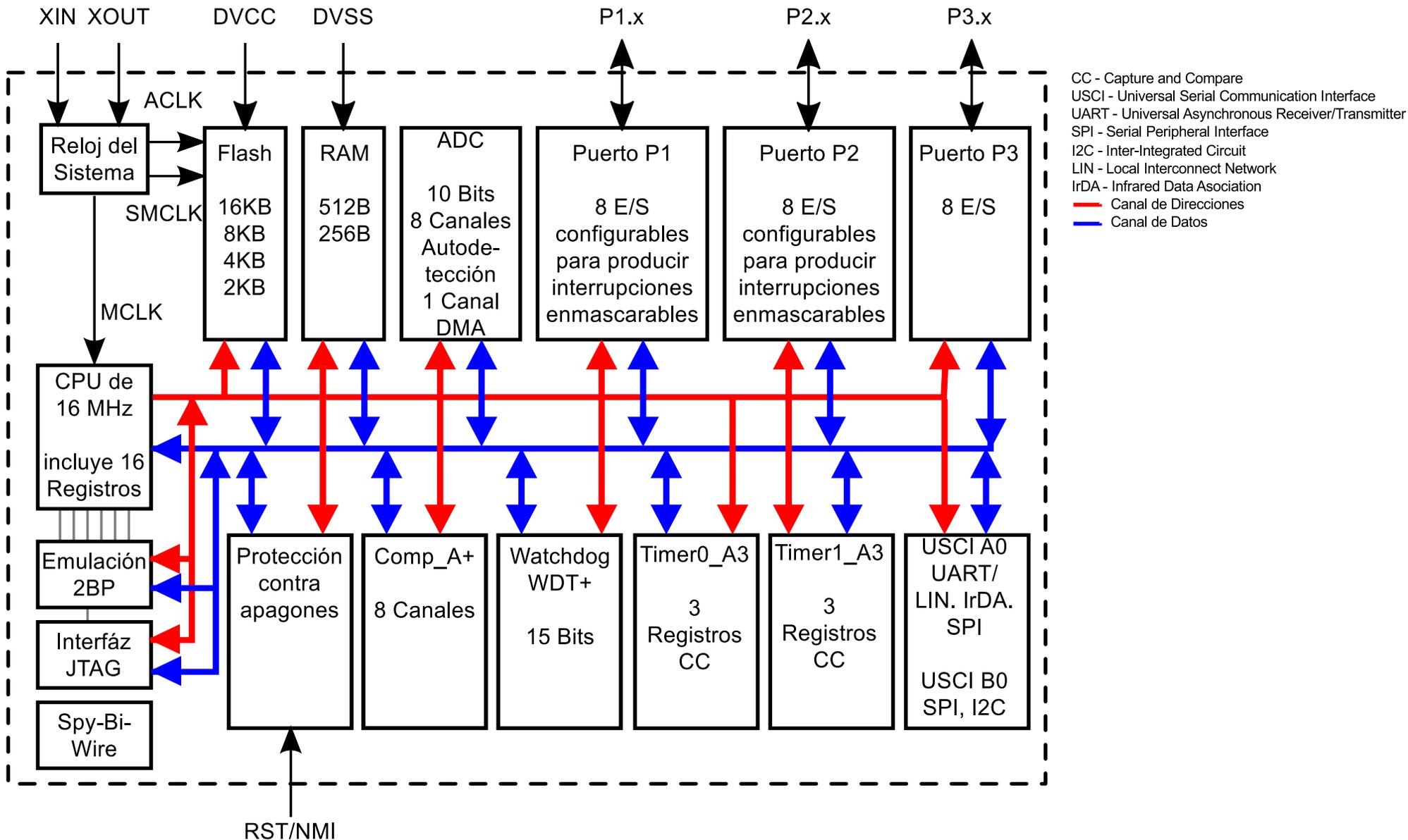
Microprocesador

Comparación de microcontroladores frente a microprocesadores

Arquitectura del Z80



Comparación de microcontroladores frente a microprocesadores



Evolución de los microcontroladores

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

▷ Evolución

Familias

MSP430G2xxx

Instrucciones

Tema 2

Evolución de los microcontroladores

En el principio existía el ábaco...

Evolución de los microcontroladores

En el principio existía el ábaco... pasaron los días y las necesidades de cómputo se incrementaron...

Evolución de los microcontroladores

En el principio existía el ábaco... pasaron los días y las necesidades de cómputo se incrementaron... surgieron las máquinas analíticas...

Evolución de los microcontroladores

En el principio existía el ábaco... pasaron los días y las necesidades de cómputo se incrementaron... surgieron las máquinas analíticas... con la invención del transistor se redujo su tamaño y se incrementó su velocidad de cálculo...

Evolución de los microcontroladores

En el principio existía el ábaco... pasaron los días y las necesidades de cómputo se incrementaron... surgieron las máquinas analíticas... con la invención del transistor se redujo su tamaño y se incrementó su velocidad de cálculo... surgió el microprocesador y todo era bueno...

Evolución de los microcontroladores

En el principio existía el ábaco... pasaron los días y las necesidades de cómputo se incrementaron... surgieron las máquinas analíticas... con la invención del transistor se redujo su tamaño y se incrementó su velocidad de cálculo... surgió el microprocesador y todo era bueno... con el avance tecnológico, se requirió usar una computadora para todo...

Evolución de los microcontroladores

En el principio existía el ábaco... pasaron los días y las necesidades de cómputo se incrementaron... surgieron las máquinas analíticas... con la invención del transistor se redujo su tamaño y se incrementó su velocidad de cálculo... surgió el microprocesador y todo era bueno... con el avance tecnológico, se requirió usar una computadora para todo... en las calculadoras, en las fotocopadoras, en las lavadoras, en los microondas, etc.,

Evolución de los microcontroladores

En el principio existía el ábaco... pasaron los días y las necesidades de cómputo se incrementaron... surgieron las máquinas analíticas... con la invención del transistor se redujo su tamaño y se incrementó su velocidad de cálculo... surgió el microprocesador y todo era bueno... con el avance tecnológico, se requirió usar una computadora para todo... en las calculadoras, en las fotocopiadoras, en las lavadoras, en los microondas, etc., por lo que el tamaño de los controladores tuvo que reducirse..

Evolución de los microcontroladores

En el principio existía el ábaco... pasaron los días y las necesidades de cómputo se incrementaron... surgieron las máquinas analíticas... con la invención del transistor se redujo su tamaño y se incrementó su velocidad de cálculo... surgió el microprocesador y todo era bueno... con el avance tecnológico, se requirió usar una computadora para todo... en las calculadoras, en las fotocopiadoras, en las lavadoras, en los microondas, etc., por lo que el tamaño de los controladores tuvo que reducirse... hasta convertirse en miniaturas...

Evolución de los microcontroladores

En el principio existía el ábaco... pasaron los días y las necesidades de cómputo se incrementaron... surgieron las máquinas analíticas... con la invención del transistor se redujo su tamaño y se incrementó su velocidad de cálculo... surgió el microprocesador y todo era bueno... con el avance tecnológico, se requirió usar una computadora para todo... en las calculadoras, en las fotocopiadoras, en las lavadoras, en los microondas, etc., por lo que el tamaño de los controladores tuvo que reducirse... hasta convertirse en miniaturas...

SURGIENDO ASI EL MICROCONTROLADOR

Familias de microcontroladores

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

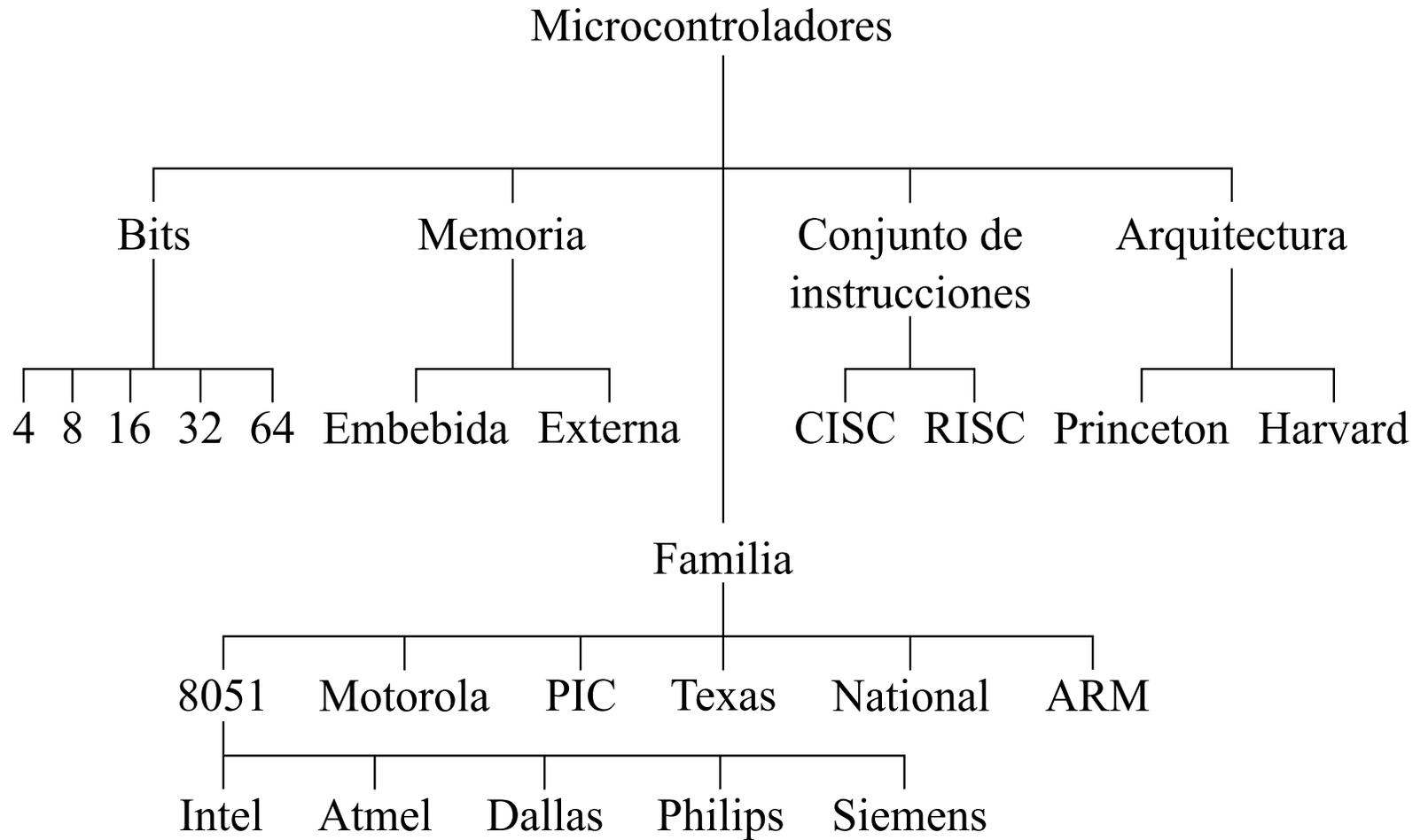
Evolución

▷ Familias

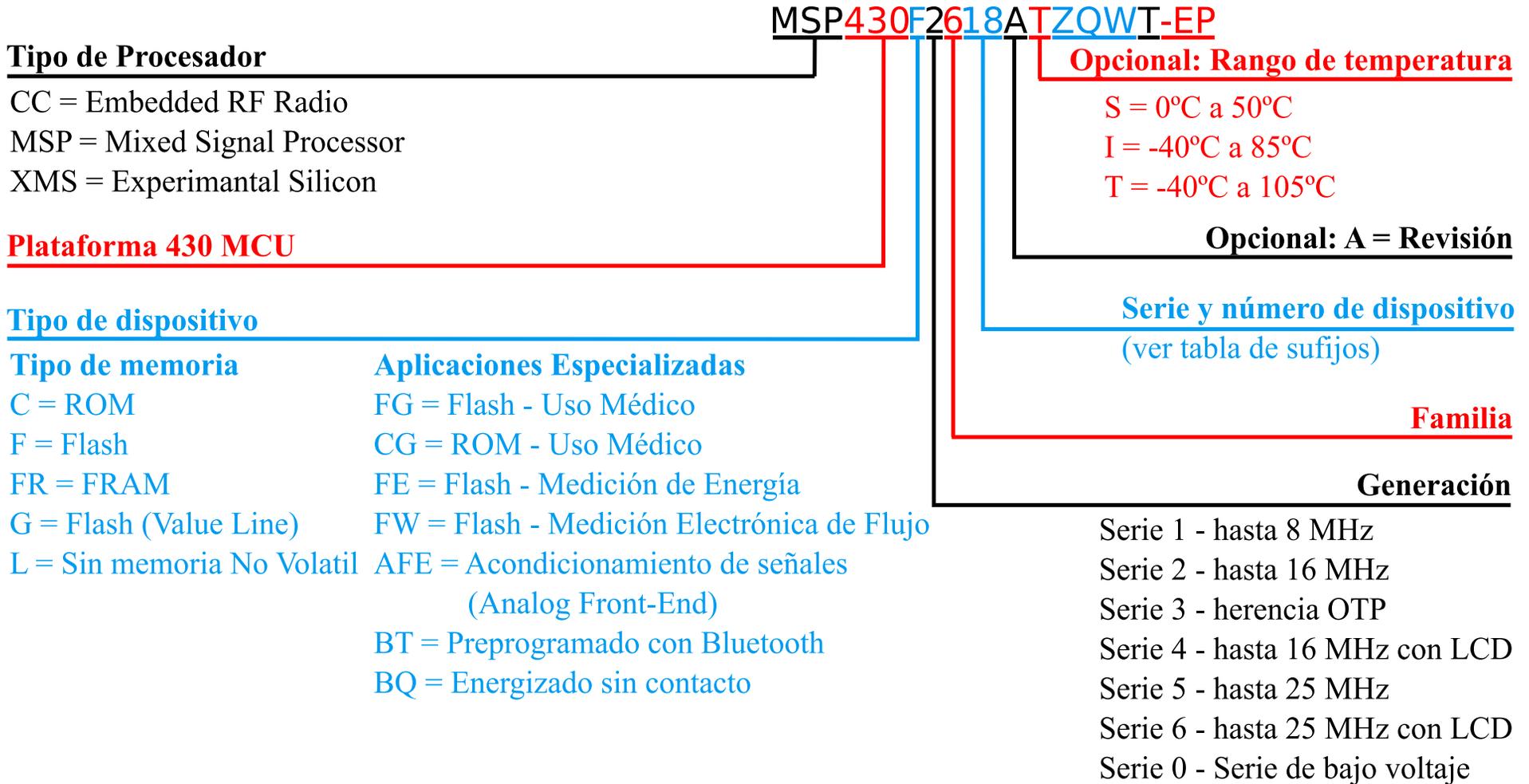
MSP430G2xxx

Instrucciones

Tema 2



Familias del MSP430



Familias del MSP430

MSP430F2618ATZQWT-EP

Generación

- Serie 1 - hasta 8 MHz
- Serie 2 - hasta 16 MHz
- Serie 3 - herencia OTP
- Serie 4 - hasta 16 MHz con LCD
- Serie 5 - hasta 25 MHz
- Serie 6 - hasta 25 MHz con LCD
- Serie 0 - Serie de bajo voltaje

Familia

Serie y número de dispositivo
(ver tabla de sufijos)

Opcional: A = Revisión

Opcional: Rango de temperatura

- S = 0°C a 50°C
- I = -40°C a 85°C
- T = -40°C a 105°C

Opcional: Características adicionales

- *-Q1 = Para uso automotriz
- *-EP = Producto mejorado (-40 a 125°C)
- *-HT = Productos para temperatura extrema (-55 a 150°C)

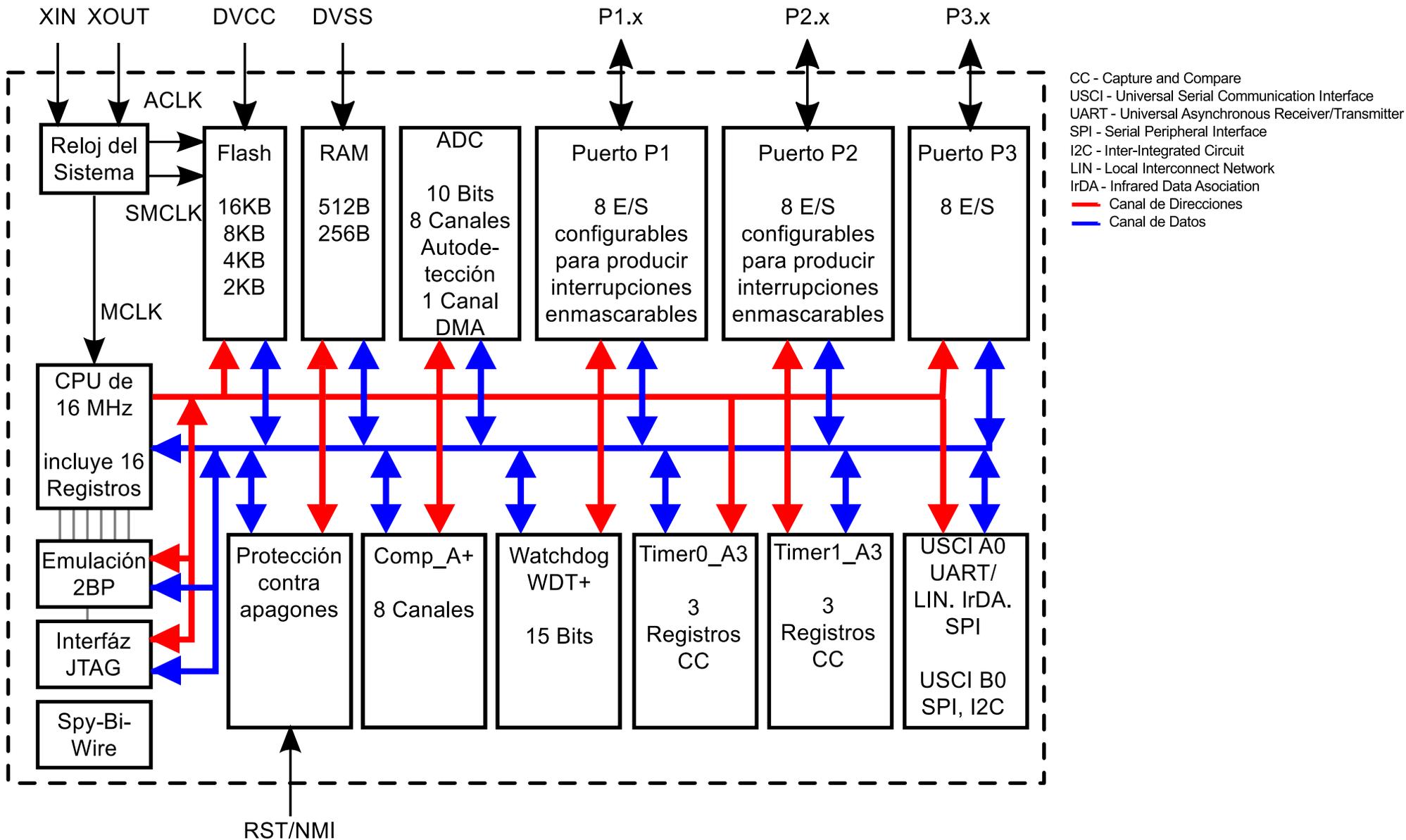
Opcional: Formato de distribución

- T = Carrete chico (7 in)
- R = Carrete Grande (11 in)
- Sin marca = Tubo plástico o charola

Encapsulado

(ver www.ti.com/packaging)

Familia MSP430G2xxx



Familia MSP430G2xxx

MSP430G2xx Series – Up to 16 MHz

Part Number	Frequency (MHz)	Non-volatile Memory (KB)	SRAM (kB)	GPIO	I ² C	SPI	UART	DMA	ADC	Comparators	Timers 16-Bit	Timers 32-Bit	Multiplier	AES	Additional Features	Operating Temperature Range (°C)	Package Group	1ku Price ¹ (U.S. \$)
MSP430G2452	16	8	0.25	16	1	1	0	0	ADC10-8ch	8	1	0	N/A	N/A	Temp sensor, BOR		PDIP, QFN, TSSOP	0.70
MSP430G2553	16	16	0.5	24	1	1	1	0	ADC10-8ch	8	2	0	N/A	N/A	Temp sensor, BOR, IrDA	-40 to 85	PDIP, TSSOP, VQFN	0.90

- Arquitectura: von Newman de 16 bits
- Conjunto de instrucciones: RISC
- Almacenamiento en memoria: Little-endian
- Todas las instrucciones se componen de entre 1 a 3 palabras, por lo que su dirección de inicio es par
- Los puertos de entrada/salida aparecen como direcciones de memoria (*memory mapped I/O*)
- Cada puerto tiene varios registros asociados que permiten configurarlo o realizar operaciones de lectura o escritura

Familia MSP430G2xxx - Unidad Central de Procesamiento

Presentación

Tema 1

Contenido

Aplicación

μP vs. μC

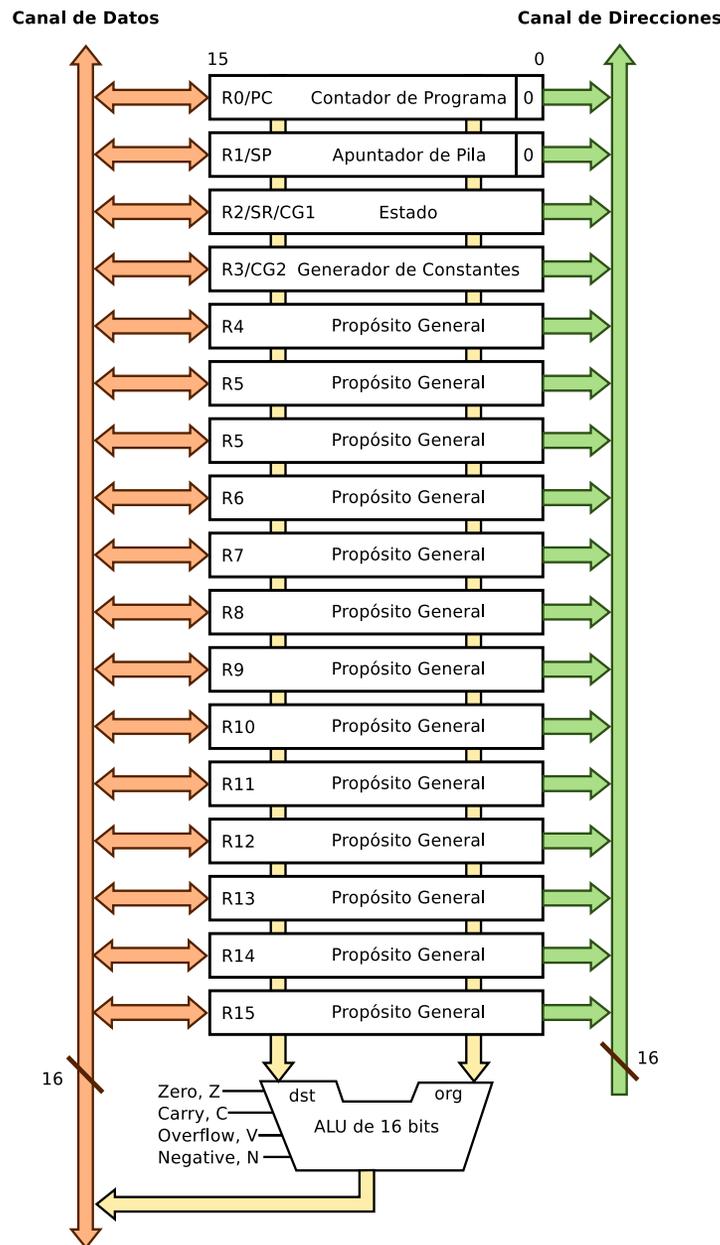
Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2



- 16 registros de 16 bits cada uno
- 4 son reservados para funciones especiales, el resto son de propósito general
- R0/PC - *Contador de programa*. Apunta a la dirección de memoria donde inicia la siguiente instrucción a ser ejecutada.
- R1/SP - *Apuntador de pila*. Apunta a la parte superior de la pila, utilizada para almacenar direcciones de retorno de subrutinas o interrupciones.
- R2/SR - *Registro de estado*. Proporciona información sobre el estado de las banderas aritméticas.
- CG1,CG2 - *Generador de constantes*. Mediante mecanismos de indexado, usando R2 y R3, provee seis constantes (0,+4,+8,+1,+2,-1) sin la necesidad de usar código adicional o localidades de memoria para su almacenamiento.

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera
V

Descripción

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera
V

Descripción

Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera

Descripción

V

Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.

N

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera

Descripción

V

Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.

N

Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera

Descripción

V

Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.

N

Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.

Z

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera

Descripción

V

Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.

N

Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.

Z

Bandera de cero. Este bit solo es igual a 1 cuando el resultado de una operación es cero.

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera

Descripción

V

Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.

N

Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.

Z

Bandera de cero. Este bit solo es igual a 1 cuando el resultado de una operación es cero.

C

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera

Descripción

V

Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.

N

Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.

Z

Bandera de cero. Este bit solo es igual a 1 cuando el resultado de una operación es cero.

C

Bandera de acarreo. Este bit solo es igual a 1 cuando el resultado de una operación produce un acarreo de salida.

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera	Descripción
V	Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.
N	Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.
Z	Bandera de cero. Este bit solo es igual a 1 cuando el resultado de una operación es cero.
C	Bandera de acarreo. Este bit solo es igual a 1 cuando el resultado de una operación produce un acarreo de salida.
SCG1	

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera	Descripción
V	Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.
N	Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.
Z	Bandera de cero. Este bit solo es igual a 1 cuando el resultado de una operación es cero.
C	Bandera de acarreo. Este bit solo es igual a 1 cuando el resultado de una operación produce un acarreo de salida.
SCG1	Generador del sistema de reloj (1). Cuando se escribe un 1 a este bit, el SMCKL se apaga.

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera	Descripción
V	Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.
N	Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.
Z	Bandera de cero. Este bit solo es igual a 1 cuando el resultado de una operación es cero.
C	Bandera de acarreo. Este bit solo es igual a 1 cuando el resultado de una operación produce un acarreo de salida.
SCG1	Generador del sistema de reloj (1). Cuando se escribe un 1 a este bit, el SMCKL se apaga.
SCG0	

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera	Descripción
V	Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.
N	Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.
Z	Bandera de cero. Este bit solo es igual a 1 cuando el resultado de una operación es cero.
C	Bandera de acarreo. Este bit solo es igual a 1 cuando el resultado de una operación produce un acarreo de salida.
SCG1	Generador del sistema de reloj (1). Cuando se escribe un 1 a este bit, el SMCKL se apaga.
SCG0	Generador del sistema de reloj (0). Cuando se escribe un 1 a este bit, el DCO se apaga.

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera	Descripción
V	Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.
N	Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.
Z	Bandera de cero. Este bit solo es igual a 1 cuando el resultado de una operación es cero.
C	Bandera de acarreo. Este bit solo es igual a 1 cuando el resultado de una operación produce un acarreo de salida.
SCG1	Generador del sistema de reloj (1). Cuando se escribe un 1 a este bit, el SMCKL se apaga.
SCG0	Generador del sistema de reloj (0). Cuando se escribe un 1 a este bit, el DCO se apaga.
OSCOFF	

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera	Descripción
V	Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.
N	Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.
Z	Bandera de cero. Este bit solo es igual a 1 cuando el resultado de una operación es cero.
C	Bandera de acarreo. Este bit solo es igual a 1 cuando el resultado de una operación produce un acarreo de salida.
SCG1	Generador del sistema de reloj (1). Cuando se escribe un 1 a este bit, el SMCKL se apaga.
SCG0	Generador del sistema de reloj (0). Cuando se escribe un 1 a este bit, el DCO se apaga.
OSCOFF	Interruptor del oscilador externo. Al escribir un 1 a este bit se desactiva la entrada del oscilador externo LFCT1.

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera	Descripción
V	Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.
N	Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.
Z	Bandera de cero. Este bit solo es igual a 1 cuando el resultado de una operación es cero.
C	Bandera de acarreo. Este bit solo es igual a 1 cuando el resultado de una operación produce un acarreo de salida.
SCG1	Generador del sistema de reloj (1). Cuando se escribe un 1 a este bit, el SMCKL se apaga.
SCG0	Generador del sistema de reloj (0). Cuando se escribe un 1 a este bit, el DCO se apaga.
OSCOFF	Interruptor del oscilador externo. Al escribir un 1 a este bit se desactiva la entrada del oscilador externo LFCT1.
CPUOFF	

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera	Descripción
V	Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.
N	Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.
Z	Bandera de cero. Este bit solo es igual a 1 cuando el resultado de una operación es cero.
C	Bandera de acarreo. Este bit solo es igual a 1 cuando el resultado de una operación produce un acarreo de salida.
SCG1	Generador del sistema de reloj (1). Cuando se escribe un 1 a este bit, el SMCKL se apaga.
SCG0	Generador del sistema de reloj (0). Cuando se escribe un 1 a este bit, el DCO se apaga.
OSCOFF	Interruptor del oscilador externo. Al escribir un 1 a este bit se desactiva la entrada del oscilador externo LFCT1.
CPUOFF	Interruptor de CPU. El CPU se apaga al escribir un 1 a este bit.

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera	Descripción
V	Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.
N	Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.
Z	Bandera de cero. Este bit solo es igual a 1 cuando el resultado de una operación es cero.
C	Bandera de acarreo. Este bit solo es igual a 1 cuando el resultado de una operación produce un acarreo de salida.
SCG1	Generador del sistema de reloj (1). Cuando se escribe un 1 a este bit, el SMCKL se apaga.
SCG0	Generador del sistema de reloj (0). Cuando se escribe un 1 a este bit, el DCO se apaga.
OSCOFF	Interruptor del oscilador externo. Al escribir un 1 a este bit se desactiva la entrada del oscilador externo LFCT1.
CPUOFF	Interruptor de CPU. El CPU se apaga al escribir un 1 a este bit.
GIE	

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Unidad Central de Procesamiento

Registro de estado

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reservados							V	SCG1	SCG0	OSCOFF	CPUOFF	GIE	N	Z	C

Bandera	Descripción
V	Bandera de sobreflujo. Este bit solo es igual a uno cuando el resultado de una operación aritmética sobrepasó el rango de representación permitido para números con signo.
N	Bandera de negativos. Este bit solo es igual a 1 cuando el resultado de una operación es negativo.
Z	Bandera de cero. Este bit solo es igual a 1 cuando el resultado de una operación es cero.
C	Bandera de acarreo. Este bit solo es igual a 1 cuando el resultado de una operación produce un acarreo de salida.
SCG1	Generador del sistema de reloj (1). Cuando se escribe un 1 a este bit, el SMCKL se apaga.
SCG0	Generador del sistema de reloj (0). Cuando se escribe un 1 a este bit, el DCO se apaga.
OSCOFF	Interruptor del oscilador externo. Al escribir un 1 a este bit se desactiva la entrada del oscilador externo LFCT1.
CPUOFF	Interruptor de CPU. El CPU se apaga al escribir un 1 a este bit.
GIE	Habilitación de interrupciones. Todas las interrupciones mascarables se habilitan al escribir un 1 a este bit.

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

Familia MSP430G2xxx - Organización de la memoria

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

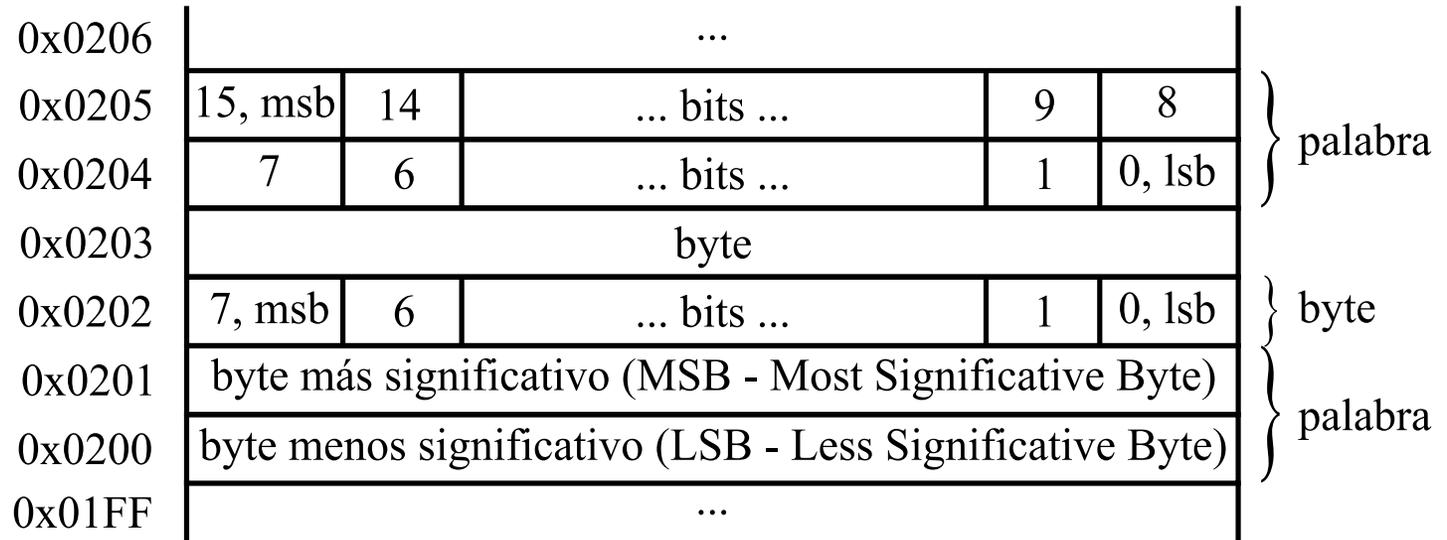
Familias

▷ MSP430G2xxx

Instrucciones

Tema 2

			MSP430G2452		MSP430G2553	
Memoria	Tamaño		8kB		16kB	
Principal: vector de interrupciones	Flash		0xFFFF	0xFFC0	0xFFFF	0xFFC0
Principal: código de usuario	Flash		0xFFBF	0xE000	0xFFBF	0xC000
			Espacio			
Memoria de información	Tamaño		256 bytes		256 bytes	
	Flash		0x10FF	0x1000	0x10FF	0x1000
			Espacio			
RAM	Tamaño		256 bytes		512 bytes	
			0x02FF	0x0200	0x03FF	0x0200
	Tamaño		512 bytes		512 bytes	
Periféricos	16 bits		0x01FF	0x0100	0x01FF	0x0100
	8 bits		0x00FF	0x0010	0x00FF	0x0010
	RFE de 8 bits		0x000F	0x0000	0x000F	0x0000



Familia MSP430G2xxx - Conjunto de instrucciones

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

MSP430G2xxx

▷ Instrucciones

Tema 2

- 27 instrucciones básicas (core instructions)
 - Código de operación único para cada instrucción
 - Se clasifican en tres formatos
 - Binarias
 - Unitarias
 - Saltos

- 24 instrucciones emuladas (emulated instructions)
 - No poseen un código de operación
 - Hacen los programas más fáciles de escribir y leer
 - Son remplazadas por instrucciones básicas mediante el ensamblador

Familia MSP430G2xxx - Conjunto de instrucciones

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

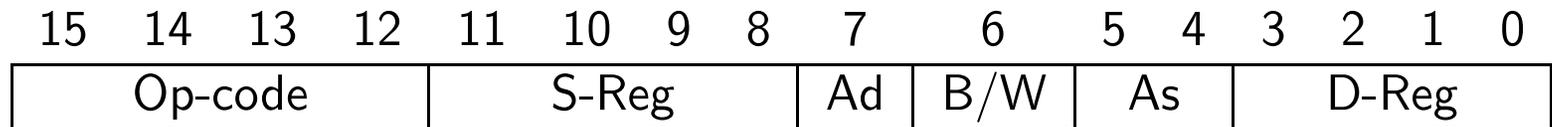
Familias

MSP430G2xxx

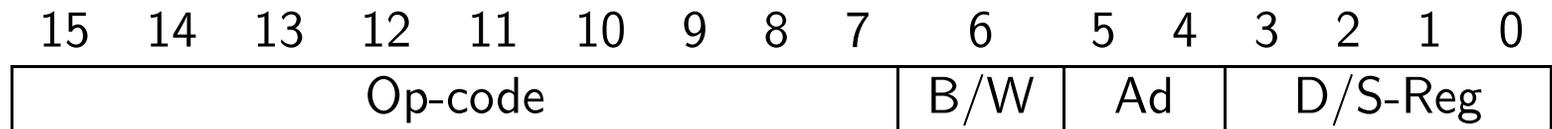
▷ Instrucciones

Tema 2

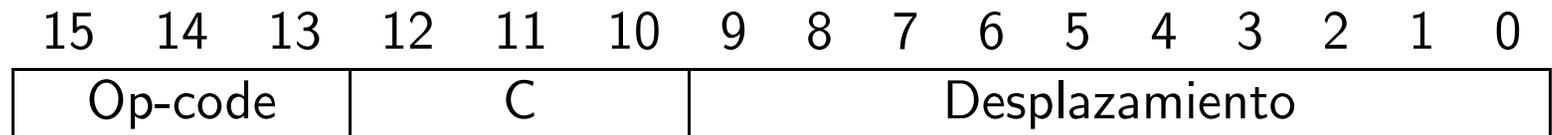
Formato de las instrucciones binarias



Formato de las instrucciones unitarias



Formato de las instrucciones de salto



As Indica el modo de direccionamiento usado para la fuente

S-reg El registro usado como fuente

Ad Indica el modo de direccionamiento usado para el destino

D-reg El registro usado como destino

B/W Indica si la instrucción opera sobre bytes (B/W=1) o palabras (B/W=0)

Familia MSP430G2xxx - Conjunto de instrucciones

Instrucciones binarias

- `mov.b/mov` - transfiere un byte o una palabra de la fuente `src` al destino `dst`: $src \rightarrow dst$
- `add.b/add` - $src + dst \rightarrow dst$
- `addc.b/addc` - $src + dst + C \rightarrow dst$
- `sub.b/sub` - $dst - src \rightarrow dst$
- `subc.b/subc` - Permite restar por partes, números de más de 16 bits: $dst - src - 1 + C \rightarrow dst$
- `cmp.b/cmp` - Solo afecta banderas, el resultado se descarta: $dst - src$
- `dadd.b/dadd` - Suma en BCD: $src + dst + C \rightarrow dst$
- `bit.b/bit` - $src \text{ AND } dst$
- `bic.b/bic` - $\text{NOT}(src) \text{ AND } dst \rightarrow dst$
- `bis.b/bis` - $src \text{ OR } dst \rightarrow dst$
- `xor.b/xor` - $src \text{ XOR } dst \rightarrow dst$
- `and.b/and` - $src \text{ AND } dst \rightarrow dst$

Presentación

Tema 1

Contenido

Aplicación

μP vs. μC

Evolución

Familias

MSP430G2xxx

▷ Instrucciones

Tema 2

Familia MSP430G2xxx - Conjunto de instrucciones

Instrucciones unitrias

- rrc.b/rrc - rotación a la derecha concatenado con el acarreo: $C \rightarrow \text{MSB}$, $\text{MSB} \rightarrow \text{MSB}-1$, ..., $\text{LSB}+1 \rightarrow \text{LSB}$, $\text{LSB} \rightarrow C$
- swpb - intercambia el byte bajo con el byte alto del operando
- rra.b/rra - rotación aritmética a la derecha: $\text{MSB} \rightarrow \text{MSB}$, $\text{MSB} \rightarrow \text{MSB}-1$, ..., $\text{LSB}+1 \rightarrow \text{LSB}$, $\text{LSB} \rightarrow C$
- sxt - extensión con signo de byte a palabra
- push.b/push - ingresa dato a pila: $\text{SP}-2 \rightarrow \text{SP}$, $\text{src} \rightarrow @\text{SP}$
- call
- reti - regreso de interrupción: $\text{TOS} \rightarrow \text{SR}$, $\text{SP}+2 \rightarrow \text{SP}$, $\text{TOS} \rightarrow \text{PC}$, $\text{SP}+2 \rightarrow \text{SP}$

Presentación

Tema 1

Contenido

Aplicación

μP vs. μC

Evolución

Familias

MSP430G2xxx

▷ Instrucciones

Tema 2

Familia MSP430G2xxx - Conjunto de instrucciones

Instrucciones de salto

- jeq/jz - salta si $Z=1$: $PC+2 \text{ offset} \rightarrow PC$
- jne/jnz salta si $Z=0$
- jc - salta si $C=1$
- jnc - salta si $C=0$
- jn - salta si $N=1$
- jge - salta si $N \text{ XOR } V=0$
- jl - salta si $N \text{ XOR } V=1$
- jmp - salto incondicional

Presentación

Tema 1

Contenido

Aplicación

μP vs. μC

Evolución

Familias

MSP430G2xxx

▷ Instrucciones

Tema 2

Familia MSP430G2xxx - Modos de direccionamiento

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

MSP430G2xxx

▷ Instrucciones

Tema 2

Los modos de direccionamiento es la forma en que se especifican los operandos en las instrucciones en ensamblador.

Familia MSP430G2xxx - Modos de direccionamiento

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

MSP430G2xxx

▷ Instrucciones

Tema 2

Los modos de direccionamiento es la forma en que se especifican los operandos en las instrucciones en ensamblador. En el MSP430 existen siete modos de direccionamiento para el operando fuente y cuatro para el operando destino.

Familia MSP430G2xxx - Modos de direccionamiento

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

MSP430G2xxx

▷ Instrucciones

Tema 2

Los modos de direccionamiento es la forma en que se especifican los operandos en las instrucciones en ensamblador. En el MSP430 existen siete modos de direccionamiento para el operando fuente y cuatro para el operando destino.

As/Ad	Modo de direccionamiento	Sintaxis	Descripción
00/0	Registro	Rn	El contenido del registro Rn sirve como operando
01/1	Indexado	X(Rn)	El operando se encuentra en la dirección (Rn+X), en donde X es una palabra almacenada en las siguientes dos direcciones de memoria
01/1	Simbólico	ADDR	El operando se encuentra en la dirección (PC+X), por lo que es equivalente a X(PC)

Familia MSP430G2xxx - Modos de direccionamiento

Los modos de direccionamiento es la forma en que se especifican los operandos en las instrucciones en ensamblador. En el MSP430 existen siete modos de direccionamiento para el operando fuente y cuatro para el operando destino.

As/Ad	Modo de direccionamiento	Sintaxis	Descripción
01/1	Absoluto	&ADDR	La palabra que le sigue al código de instrucción contiene la dirección absoluta de donde se leerá el operando. El modo indexado X(SR) es utilizado para encontrar S/D-Reg.
10/-	Registro-Indirecto	@Rn	Rn se usa como apuntador al operando
11/-	Indirecto-Autoincremento	@Rn+	Rn se usa como apuntador al operando. Después de ejecutar la instrucción, Rn se incrementa en 1 para instrucciones .b o en 2 para instrucciones .w
11/-	Inmediato	#N	La palabra siguiente al código de operación contiene el operando. Equivale al modo @PC+.

Presentación

Tema 1

Contenido

Aplicación

μ P vs. μ C

Evolución

Familias

MSP430G2xxx

▷ Instrucciones

Tema 2

Tema 2 - Programación en ensamblador y lenguaje C - (3.0 hrs)

Presentación

Tema 1

Tema 2

▷ Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

1. Uso de un ensamblador, un simulador y/o un compilador.
2. Estructura general de un programa en ensamblador.
 - 2.1 Estructuras de control en ensamblador.
 - 2.1.1 Selección if y case.
 - 2.1.2 Ciclos for, while y do-while.
 - 2.2 Operadores aritméticos y lógicos.
3. Estructura general de un programa en C
 - 3.1 Funciones y bibliotecas.
4. Ejemplos de programación

Tema 2 - Programación en ensamblador y lenguaje C - (3.0 hrs)

Presentación

Tema 1

Tema 2

▷ Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

1. Uso de un ensamblador, un simulador y/o un compilador.
2. Estructura general de un programa en ensamblador.
 - 2.1 Estructuras de control en ensamblador.
 - 2.1.1 Selección if y case.
 - 2.1.2 Ciclos for, while y do-while.
 - 2.2 Operadores aritméticos y lógicos.
3. Estructura general de un programa en C
 - 3.1 Funciones y bibliotecas.
4. Ejemplos de programación

Tema 2 - Programación en ensamblador y lenguaje C - (3.0 hrs)

Presentación

Tema 1

Tema 2

▷ Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

1. Uso de un ensamblador, un simulador y/o un compilador.
2. Estructura general de un programa en ensamblador.
 - 2.1 Estructuras de control en ensamblador.
 - 2.1.1 Selección if y case.
 - 2.1.2 Ciclos for, while y do-while.
 - 2.2 Operadores aritméticos y lógicos.
3. Estructura general de un programa en C
 - 3.1 Funciones y bibliotecas.
4. Ejemplos de programación

Tema 2 - Programación en ensamblador y lenguaje C - (3.0 hrs)

Presentación

Tema 1

Tema 2

▷ Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

1. Uso de un ensamblador, un simulador y/o un compilador.
2. Estructura general de un programa en ensamblador.
 - 2.1 Estructuras de control en ensamblador.
 - 2.1.1 Selección if y case.
 - 2.1.2 Ciclos for, while y do-while.
 - 2.2 Operadores aritméticos y lógicos.
3. Estructura general de un programa en C
 - 3.1 Funciones y bibliotecas.
4. Ejemplos de programación

Tema 2 - Programación en ensamblador y lenguaje C - (3.0 hrs)

Presentación

Tema 1

Tema 2

▷ Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

1. Uso de un ensamblador, un simulador y/o un compilador.
2. Estructura general de un programa en ensamblador.
 - 2.1 Estructuras de control en ensamblador.
 - 2.1.1 Selección if y case.
 - 2.1.2 Ciclos for, while y do-while.
 - 2.2 Operadores aritméticos y lógicos.
3. Estructura general de un programa en C
 - 3.1 Funciones y bibliotecas.
4. Ejemplos de programación

Tema 2 - Programación en ensamblador y lenguaje C - (3.0 hrs)

Presentación

Tema 1

Tema 2

▷ Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

1. Uso de un ensamblador, un simulador y/o un compilador.
2. Estructura general de un programa en ensamblador.
 - 2.1 Estructuras de control en ensamblador.
 - 2.1.1 Selección if y case.
 - 2.1.2 Ciclos for, while y do-while.
 - 2.2 Operadores aritméticos y lógicos.
3. Estructura general de un programa en C
 - 3.1 Funciones y bibliotecas.
4. Ejemplos de programación

Tema 2 - Programación en ensamblador y lenguaje C - (3.0 hrs)

Presentación

Tema 1

Tema 2

▷ Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

1. Uso de un ensamblador, un simulador y/o un compilador.
2. Estructura general de un programa en ensamblador.
 - 2.1 Estructuras de control en ensamblador.
 - 2.1.1 Selección if y case.
 - 2.1.2 Ciclos for, while y do-while.
 - 2.2 Operadores aritméticos y lógicos.
3. Estructura general de un programa en C
 - 3.1 Funciones y bibliotecas.
4. Ejemplos de programación

Tema 2 - Programación en ensamblador y lenguaje C - (3.0 hrs)

Presentación

Tema 1

Tema 2

▷ Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

1. Uso de un ensamblador, un simulador y/o un compilador.
2. Estructura general de un programa en ensamblador.
 - 2.1 Estructuras de control en ensamblador.
 - 2.1.1 Selección if y case.
 - 2.1.2 Ciclos for, while y do-while.
 - 2.2 Operadores aritméticos y lógicos.
3. Estructura general de un programa en C
 - 3.1 Funciones y bibliotecas.
4. Ejemplos de programación

Tema 2 - Programación en ensamblador y lenguaje C - (3.0 hrs)

Presentación

Tema 1

Tema 2

▷ Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

1. Uso de un ensamblador, un simulador y/o un compilador.
2. Estructura general de un programa en ensamblador.
 - 2.1 Estructuras de control en ensamblador.
 - 2.1.1 Selección if y case.
 - 2.1.2 Ciclos for, while y do-while.
 - 2.2 Operadores aritméticos y lógicos.
3. Estructura general de un programa en C
 - 3.1 Funciones y bibliotecas.
4. Ejemplos de programación

Uso de un ensamblador, un simulador y/o un compilador

Presentación

Tema 1

Tema 2

Contenido

▷ IDE

Programa en ensamblador

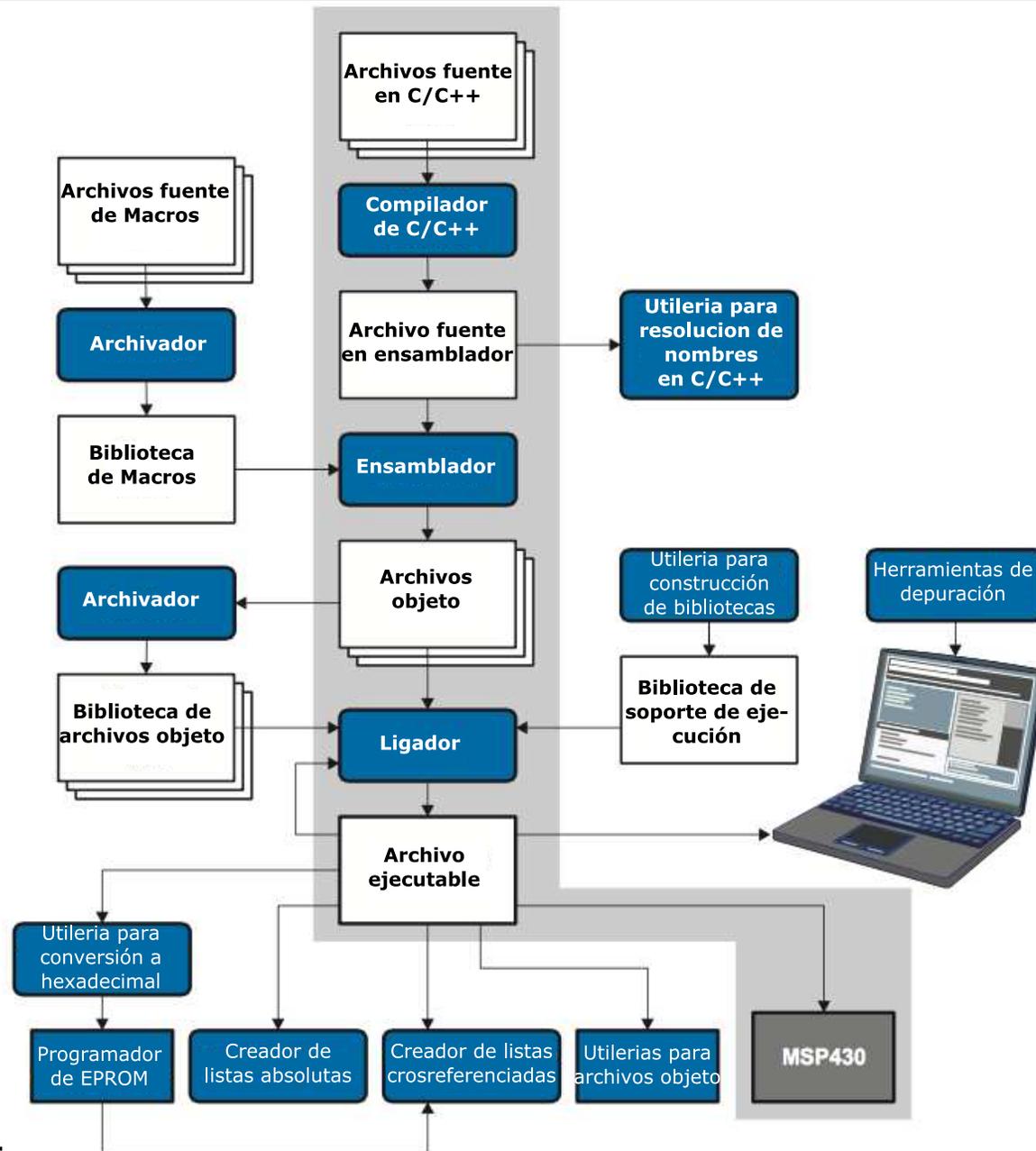
if-then-else

switch/case

ciclo for

ciclo while

La pila



Uso de un ensamblador, un simulador y/o un compilador

Presentación

Tema 1

Tema 2

Contenido

▷ IDE

Programa en

ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

- Ver el documento `lab00.pdf` para un tutorial sobre Code Composer Studio, que es el Entorno Integrado de Desarrollo (IDE - Integrated Development Environment) que provee Texas Instruments para uso bajo Windows.
- Para otros sistemas operativos, existen herramientas de código abierto, más ligeras pero que carecen de una interfaz gráfica. Ver el documento `tools_install_linux.pdf` para un pequeño tutorial sobre instalación y uso de dichas herramientas para el sistema operativo Linux, y el documento `tools_install_mac.pdf` con el mismo tutorial pero para el sistema operativo OS X.

Estructura general de un programa en ensamblador

Presentación

Tema 1

Tema 2

Contenido

IDE

Programa en
▷ ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

Código fuente - blink.asm

```
1      .include "msp430x2xx.inc"           ;biblioteca de macros
2
3      org 0xf800                          ;dirección de inicio del programa
4  start: mov.w #WDTPW|WDTHOLD, &WDTCTL    ;apaga el temporizador "watchdog"
5        mov.b #0x41, &PDIR              ;configura P1.0 y P1.6 como salidas
6  repeat: mov.w #0x01, r8                 ;r8.bit0=1 y r8.bit6=0
7        mov.b r8, &P1OUT                 ;P1.0 XOR 1; P1.6 XOR 0
8        xor.b #0x41, r8                  ;r8.bit0 XOR 1; r8.bit6 XOR 1
9  waiter: dec r9                           ;retardo: decrementa r9 hasta llegar a cero
10      jnz waiter                         ;fin del ciclo de retardo
11      jmp repeat                          ;ciclo infinito
12
13     org 0xfffe
14     dw start
```

Estructura general de un programa en ensamblador

Presentación

Tema 1

Tema 2

Contenido

IDE

Programa en
▷ ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

Resolución de macros - blink.asm

```
1      org 0xf800
2  start:  mov.w #0x5a80, &0x0120
3         mov.b #0x41, &0x0022
4         mov.w #1, r8
5  repeat: mov.b r8, &0x0021
6         xor.b #0x41, r8
7         mov.w #0x9c40, r9
8  waiter: sub.w #1, r9
9         jnz waiter
10        jmp repeat

11     org 0xfffe
12     dw start
```

Estructura general de un programa en ensamblador

Presentación

Tema 1

Tema 2

Contenido

IDE

Programa en
▷ ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

Lista absoluta - blink.asm

```
1          org 0xf800
2 start:   mov.w #0x5a80, &0x0120    ;5 ciclos
          0xf800: 0x40b2, 0xf802: 0x5a80, 0xf804: 0x0120
3          mov.b #0x41, &0x0022    ;5 ciclos
          0xf806: 0x40f2, 0xf808: 0x0041, 0xf80a: 0x0022
4          mov.w #1, r8              ;1 ciclo
          0xf80c: 0x4318
5 repeat:  mov.b r8, &0x0021         ;4 ciclos
          0xf80e: 0x48c2, 0xf810: 0x0021
6          xor.b #0x41, r8          ;2 ciclos
          0xf812: 0xe078, 0xf814: 0x0041
7          mov.w #0x9c40, r9        ;2 ciclos
          0xf816: 0x4039, 0xf818: 0x9c40
8 waiter: sub.w #1, r9              ;1 ciclo
          0xf81a: 0x8319
9          jne 0xf81a               ;2 ciclos
          0xf81c: 0x23fe
10         jmp 0xf80e                ;2 ciclos
          0xf81e: 0x3ff7
11         org 0xfffe
12         dw start
          0xfffe: 0xf800
```

Estructura general de un programa en ensamblador

Presentación

Tema 1

Tema 2

Contenido

IDE

Programa en
▷ ensamblador

if-then-else

switch/case

ciclo for

ciclo while

La pila

Lenguaje máquina - blink.asm

0xf800:	b2	40	80	5a	20	01
0xf806:	f2	40	41	00	22	00
0xf80c:	18	43	c2	48	21	00
0xf812:	78	e0	41	00	39	40
0xf818:	40	9c	19	83	fe	23
0xf81e:	f7	3f	??	??	??	??
.....					
0xffff:	00	f8	??	??	??	??

Estructuras de control en lenguaje ensamblador: if-then-else

Presentación

Tema 1

Tema 2

Contenido

IDE

Programa en ensamblador

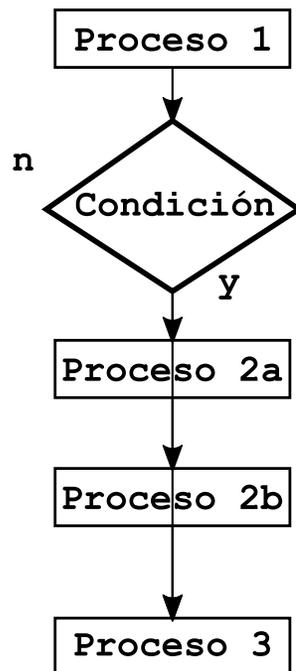
▷ if-then-else

switch/case

ciclo for

ciclo while

La pila

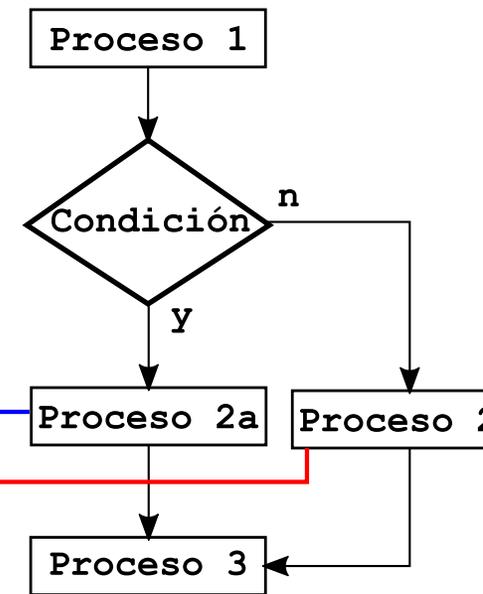


Lenguaje Ensamblador

```
cmp.w #1, entrada
jne ifFalse
bis.b #0x01, P1OUT
jmp endIf
ifFalse:
bic.b #0x01, P1OUT
endIf:
...
```

Lenguaje C

```
if (entrada==1) {
    prende_LED();
} else {
    apaga_LED();
}
```



Estructuras de control en lenguaje ensamblador: switch/case

Presentación

Tema 1

Tema 2

Contenido

IDE

Programa en
ensamblador

if-then-else

▷ switch/case

ciclo for

ciclo while

La pila

Lenguaje C

```
#define punto 0x002E
#define guion 0x002D

switch (tecla)
{
    case punto:
        muestra_punto();
        break;

    case coma:
        muestra_coma();
        break;

    ...

    default:
        ...

}
```

Lenguaje Ensamblador

```
sw_00:
    cmp.w #punto, &tecla
    jne sw_01
    call #muestra_punto
    jmp sw_default

sw_01:
    cmp.w #guion, &tecla
    jne sw_02
    call #muestra_guion
    jmp sw_default

sw_02:
    ...

sw_default:
    ...
```

Estructuras de control en lenguaje ensamblador: ciclo for

Presentación

Tema 1

Tema 2

Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

▷ ciclo for

ciclo while

La pila

Lenguaje C

Lenguaje Ensamblador

```
int i;
```

```
for(i=10; i>0; i--)
```

```
{
```

```
    prende_LED();
```

```
    retardo();
```

```
    apaga_LED();
```

```
    retardo();
```

```
}
```

```
.bss i,2
```

```
mov.w #10,&i
```

```
loop: call #prende_LED
```

```
call #retardo
```

```
call #apaga_LED
```

```
call #retardo
```

```
dec.w &i
```

```
jnz loop
```

```
fin_for:
```

Estructuras de control en lenguaje ensamblador: ciclo while

Presentación

Tema 1

Tema 2

Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

ciclo for

▷ ciclo while

La pila

Lenguaje C

Lenguaje Ensamblador

```
#define TRUE 1

while (TRUE)
{
    prende_LED();
    retardo();
    apaga_LED();
    retardo();
}
```

```
while_loop:
    bit.w #1,r4
    jnz fin_while
    call #prende_LED
    call #retardo
    call #apaga_LED
    call #retardo
    jmp while_loop
fin_while:
```

- Estructura LIFO (abreviatura del inglés Last In - First Out)
 - El apuntador de pila (SP por sus siglas en inglés) apunta al último elemento almacenado en la pila (TOS del inglés Top Of Stack).
 - Se ingresan datos a la pila mediante la instrucción push, y se recuperan datos de la pila con la instrucción pop.
 - Se pueden usar todos los modos de direccionamiento.
- Debe ser inicializada correctamente al inicio de cada programa.
- Se usa para almacenar:
 - variables locales (temporales).
 - estado de los registros.
 - el contador del programa y el registro de estado antes de atender una interrupción.
- Forma barata de almacenar datos temporales.

La Pila - Subrutinas

Presentación

Tema 1

Tema 2

Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

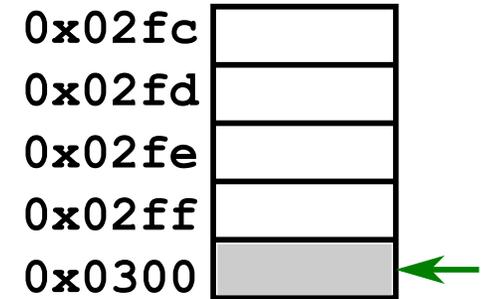
ciclo for

ciclo while

▷ La pila

```
repite:  mov.b R8, &P1OUT
0xf800:  0x48c2, 0xf802:  0x0021
        mov.b #0x9c40, R12
0xf804:  0x403c, 0xf806:  0x9c40
        call #retardo
→ 0xf808:  0x12b0, 0xf80a:  0xf812
        xor.b #0x41, R8
0xf80c:  0xe078, 0xf80e:  0x0041
        jmp  repite
0xf810:  0x3ff7
```

```
retardo: dec.w R12
0xf812:  0x831c
        jnz retardo
0xf814:  0x23fe
        ret
0xf816:  0x4130
```



SP **0x0300**

PC **0xf808**

La Pila - Subrutinas

Presentación

Tema 1

Tema 2

Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

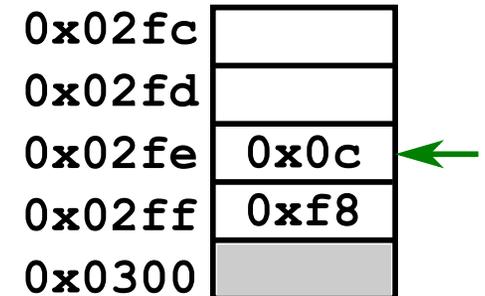
ciclo for

ciclo while

▷ La pila

```
repite:  mov.b R8, &P1OUT
0xf800:  0x48c2, 0xf802:  0x0021
        mov.b #0x9c40, R12
0xf804:  0x403c, 0xf806:  0x9c40
        call #retardo
0xf808:  0x12b0, 0xf80a:  0xf812
        xor.b #0x41, R8
0xf80c:  0xe078, 0xf80e:  0x0041
        jmp  repite
0xf810:  0x3ff7
```

```
retardo: dec.w R12
→ 0xf812: 0x831c
        jnz retardo
0xf814: 0x23fe
        ret
0xf816: 0x4130
```



SP 0x02fe

PC 0xf812

La Pila - Subrutinas

Presentación

Tema 1

Tema 2

Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

ciclo for

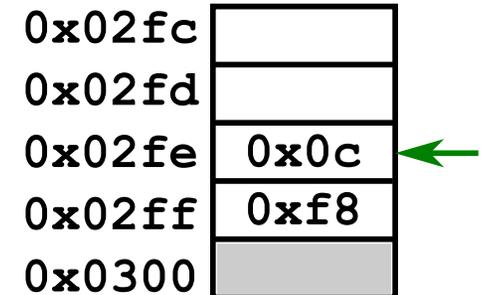
ciclo while

▷ La pila

```
repite:  mov.b R8, &P1OUT
0xf800:  0x48c2, 0xf802:  0x0021
        mov.b #0x9c40, R12
0xf804:  0x403c, 0xf806:  0x9c40
        call #retardo
0xf808:  0x12b0, 0xf80a:  0xf812
        xor.b #0x41, R8
0xf80c:  0xe078, 0xf80e:  0x0041
        jmp  repite
0xf810:  0x3ff7
```

```
retardo: dec.w R12
0xf812:  0x831c
        jnz retardo
0xf814:  0x23fe
        ret
```

→ 0xf816: 0x4130



SP 0x02fe

PC 0xf816

La Pila - Subrutinas

Presentación

Tema 1

Tema 2

Contenido

IDE

Programa en
ensamblador

if-then-else

switch/case

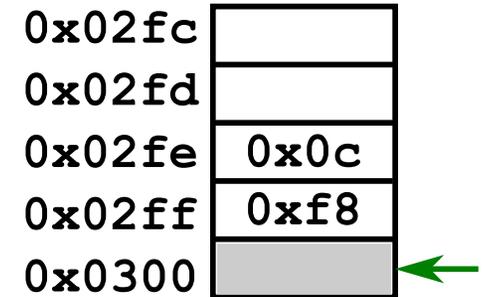
ciclo for

ciclo while

▷ La pila

```
repite:  mov.b R8, &P1OUT
0xf800:  0x48c2, 0xf802:  0x0021
        mov.b #0x9c40, R12
0xf804:  0x403c, 0xf806:  0x9c40
        call #retardo
        xor.b #0x41, R8
→ 0xf80c:  0xe078, 0xf80e:  0x0041
        jmp  repite
0xf810:  0x3ff7

retardo: dec.w R12
0xf812:  0x831c
        jnz retardo
0xf814:  0x23fe
        ret
0xf816:  0x4130
```



SP **0x0300**

PC **0xf80c**