

Introducción al microcontrolador MSP430

FAMILIA MSP430

Los Procesadores de Señales Mixtas ¹ (Mixed Signal Processor) MSP 430 de Texas Instruments son una familia de microcontroladores con capacidad añadida para operaciones de procesamiento analógico de complejidad simple a media, consumo de corriente ultra bajo, integrada por una muy amplia gama de dispositivos con el mismo núcleo y diferentes capacidades de memoria y periféricos, todos ellos compatibles en software, lo que permite una migración sencilla entre diseños a dispositivos con mayor capacidad de memoria o periféricos adicionales.

Operan en un rango de voltaje de 1.8V a 3.6V, con velocidades de operación desde 0 HZ hasta 25 MHz. La memoria de programa varía desde 512 Bytes hasta 256 KBytes

Están diseñados en torno a un CPU de 16 bits, con arquitectura Von Newman, por lo que la RAM, ROM y periféricos residen en el mismo espacio de memoria. No cuentan con buses externos, así que no hay operación en modo microcomputadora, únicamente en modo microcontrolador.

Estos dispositivos están enfocados a aplicaciones embebidas de bajo costo y baja potencia, como equipos operados por baterías. Pueden incluir un bootstrap loader BSL (cargador de arranque) integrado, una interfaz JTAG (Joint Test Action Group) la cual en este contexto representa una interfaz estandarizada, que puede ser usada para control y/o monitoreo durante la programación, depuración y pruebas del sistema o una variación de JTAG llamada Spy-Bi-Wire, usada por Texas Instruments en donde solo se usan 2 líneas, identificadas como **clock y bidirectiona data**.

Las configuraciones de estos dispositivos incluyen osciladores internos, temporizadores, generadores de PWM, temporizador de seguridad o watchdog timer, USART, SPI, I²C, A/D de 10,12,14, ó 16 bits, D/A de hasta 12 bits, comparadores analógicos, amplificadores operacionales para acondicionamiento de señales, manejadores para displays LCD, multiplicadores por hardware, USB y DMA para recuperar los resultados del A/D

Identificación de los dispositivos.

MSP430F2618ATZQWT-EP

MSP430. familia MSP F= memoria Flash 2= Generación del dispositivo 6= Modelo dentro de la generación 18= cantidad de memoria del dispositivo (ver tabla de sufijo)
A= Revisión T=Rango de temperatura ZQW = Encapsulado, en este caso *ball grid array* T = Entregado en carrete pequeño -EP = Características adicionales.

Hay seis generaciones de procesadores MSP430, que son en orden de desarrollo, la 3xx, la 1xx, la 4xx, la 2xx, la 5xx, y la 6xx. Cada una de estas generaciones tiene dispositivos con características específicas. Sin embargo, la característica relevante de esta familia es su bajo consumo de energía. Para aprovechar al máximo el ahorro, los diseños deberían contemplar el establecer un modo de bajo consumo el mayor tiempo posible dentro del proceso y salir de él por medio de una interrupción, en vez de estar inactivo en modo de poleo. Esto implica que el desarrollo del software debe integrar políticas de ahorro desde su planeación para que el sistema sea realmente un producto final con el máximo ahorro de corriente.

¹ " Un circuito de señales mixtas puede ser definido como un circuito que cuenta con elementos analógicos y digitales integrados en la misma área semiconductor" -- página 1 "An Introduction to Mixed-Signal IC Test and Measurement" Mark Burns and Gordon W. Roberts 2001

MSP430F2618ATZQWT-EP

Tipo de Procesador
 CC = Embedded RF Radio
 MSP = Mixed Signal Processor
 XMS = Experimental Silicon

Plataforma 430 MCU

Tipo de dispositivo

Tipo de memoria
 C = ROM
 F = Flash
 FR = FRAM
 G = Flash (Value Line)
 L = Sin memoria No Volatil

Aplicaciones Especializadas
 FG = Flash Uso Médico
 CG = ROM Uso Médico
 FE = Flash Medición de Energía (Energy Meter)
 FW = Flash Medición Electrónica de Flujo (Electronic Flow Meter)
 AFE = Circuitos de entrada Analógica (Analog Front End)
 BT = Preprogramado con Bluetooth
 BQ = Energizado sin contacto (Contactless Power)

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

Encapsulado
 Consultar www.ti.com/packaging

Opcional: Formato de Distribución
 T = Carrete chico (7 in)
 R = Carrete Grande (11 in)
 Sin marca = Tubo plástico o charola

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

Departamento de Electrónica. FI, UNAM Enero 2012

SUFIJO	RAM	ROM	SUFIJO	RAM	ROM
0	128	1 K	10	5 K	32 K
1	128	2 K	11	10 K	48 K
2	256	4 K	12	5 K	55/56 K
3	256	8 K	13		
4	512	12 K	14		
5	512	16 K	15		
6	1 K	24 K	16	4 K	92 K
7	1 K	32 K	17	8 K	92 K
8	2 K	48K	18	8 K	116 K
9	2 K	60K	19	4 K	120 K

Dentro de la gama x2xxx, Texas Instruments desarrolla una gama de microcontroladores, la G2XXX, de muy bajo precio a la que denomina **Value Line**. El nombre es sinónimo de

un producto económico que provee funcionalidad y servicio con características semejantes a las de un producto de mayor precio y capacidad (mid line o high end) y que esta integrada por más de 40 dispositivos:

Serie G2XX1

Series	Part Number	Flash (KB)	SRAM (B)	I/O (max)	Timers		Watchdog	BOR	USI: I ² C/SPI	USCI: I ² C/SPI/UART	Comp_ A+	Temp Sensor	ADC Ch/Res	Additional Features	Packages	1ku Price ¹
					Total	A'										
G2xx1	MSP430G2001	0.5	128	10	1	2	●	●	—	—	—	—	—	—	14PW, N; 16RSA	\$0.34
	MSP430G2101	1	128	10	1	2	●	●	—	—	—	—	—	—	14PW, N; 16RSA	\$0.44
	MSP430G2121	1	128	10	1	2	●	●	●	—	—	—	—	—	14PW, N; 16RSA	\$0.46
	MSP430G2201	2	128	10	1	2	●	●	—	—	—	—	—	—	14PW, N; 16RSA	\$0.47
	MSP430G2221	2	128	10	1	2	●	●	●	—	—	—	—	—	14PW, N; 16RSA	\$0.49
	MSP430G2111	1	128	10	1	2	●	●	—	—	●	—	Slope	—	14PW, N; 16RSA	\$0.46
	MSP430G2211	2	128	10	1	2	●	●	—	—	●	—	Slope	—	14PW, N; 16RSA	\$0.49
	MSP430G2131	1	128	10	1	2	●	●	●	—	—	●	8ch ADC10	—	14PW, N; 16RSA	\$0.49
	MSP430G2231	2	128	10	1	2	●	●	●	—	—	●	8ch ADC10	—	14PW, N; 16RSA	\$0.55

Serie G2XX2

Series	Part Number	Flash (KB)	SRAM (B)	I/O (max)	Timers		Watchdog	BOR	USI: I ² C/SPI	USCI: I ² C/SPI/UART	Comp_ A+	Temp Sensor	ADC Ch/Res	Additional Features	Packages	1ku Price ¹
					Total	A'										
G2xx2	MSP430G2102	1	256	16	1	3	●	●	●	—	—	—	—	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.48
	MSP430G2202	2	256	16	1	3	●	●	●	—	—	—	—	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.50
	MSP430G2302	4	256	16	1	3	●	●	●	—	—	—	—	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.55
	MSP430G2402	8	256	16	1	3	●	●	●	—	—	—	—	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.65
	MSP430G2112	1	256	16	1	3	●	●	●	—	●	—	Slope	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.49
	MSP430G2212	2	256	16	1	3	●	●	●	—	●	—	Slope	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.55
	MSP430G2312	4	256	16	1	3	●	●	●	—	●	—	Slope	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.60
	MSP430G2412	8	256	16	1	3	●	●	●	—	●	—	Slope	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.65
	MSP430G2132	1	256	16	1	3	●	●	●	—	—	●	8ch ADC10	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.55
	MSP430G2232	2	256	16	1	3	●	●	●	—	—	●	8ch ADC10	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.55
	MSP430G2332	4	256	16	1	3	●	●	●	—	—	●	8ch ADC10	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.60
	MSP430G2432	8	256	16	1	3	●	●	●	—	—	●	8ch ADC10	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.70
	MSP430G2152	1	256	16	1	3	●	●	●	—	●	●	8ch ADC10	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.55
	MSP430G2252	2	256	16	1	3	●	●	●	—	●	●	8ch ADC10	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.60
	MSP430G2352	4	256	16	1	3	●	●	●	—	●	●	8ch ADC10	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.65
	MSP430G2452	8	256	16	1	3	●	●	●	—	●	●	8ch ADC10	Cap touch I/O	14PW; 20PW, N; 16RSA	\$0.70

Serie G2XX3

Series	Part Number	Flash (KB)	SRAM (B)	I/O (max)	Timers		Watchdog	BOR	USI: I ² C/SPI	USCI: I ² C/SPI/UART	Comp_ A+	Temp Sensor	ADC Ch/Res	Additional Features	Packages	1ku Price ¹
					Total	A'										
G2xx3	MSP430G2203	2	256	24	2	3,3	●	●	—	●	—	—	—	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.60
	MSP430G2303	4	256	24	2	3,3	●	●	—	●	—	—	—	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.65
	MSP430G2403	8	512	24	2	3,3	●	●	—	●	—	—	—	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.75
	MSP430G2213	2	256	24	2	3,3	●	●	—	●	—	—	Slope	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.65
	MSP430G2313	4	256	24	2	3,3	●	●	—	●	—	—	Slope	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.70
	MSP430G2413	8	512	24	2	3,3	●	●	—	●	—	—	Slope	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.75
	MSP430G2513	16	512	24	2	3,3	●	●	—	●	—	—	Slope	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.90
	MSP430G2233	2	256	24	2	3,3	●	●	—	●	—	●	8ch ADC10	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.65
	MSP430G2333	4	256	24	2	3,3	●	●	—	●	—	●	8ch ADC10	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.70
	MSP430G2433	8	512	24	2	3,3	●	●	—	●	—	●	8ch ADC10	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.75
	MSP430G2533	16	512	24	2	3,3	●	●	—	●	—	●	8ch ADC10	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.90
	MSP430G2153	1	256	24	2	3,3	●	●	—	●	—	●	8ch ADC10	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.60
	MSP430G2253	2	256	24	2	3,3	●	●	—	●	—	●	8ch ADC10	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.65
	MSP430G2353	4	256	24	2	3,3	●	●	—	●	—	●	8ch ADC10	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.70
	MSP430G2453	8	512	24	2	3,3	●	●	—	●	—	●	8ch ADC10	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.80
	MSP430G2553	16	512	24	2	3,3	●	●	—	●	—	●	8ch ADC10	Cap touch I/O	20PW, N; 28PW; 32RHB	\$0.90

El conjunto de instrucciones es tipo RISC, ortogonal, aunque no todas las instrucciones se codifican en una sola palabra de instrucción. Hay siete modos de direccionamiento, todos ellos ortogonales (todas las instrucciones usan todos los modos de direccionamiento).

En este curso de introducción se usara la familia MSP430G2XXX y los microcontroladores MSP430G2231IN14, MSP430G2211IN14 y el MSP430G2452IN20

La arquitectura de la familia se muestra a continuación

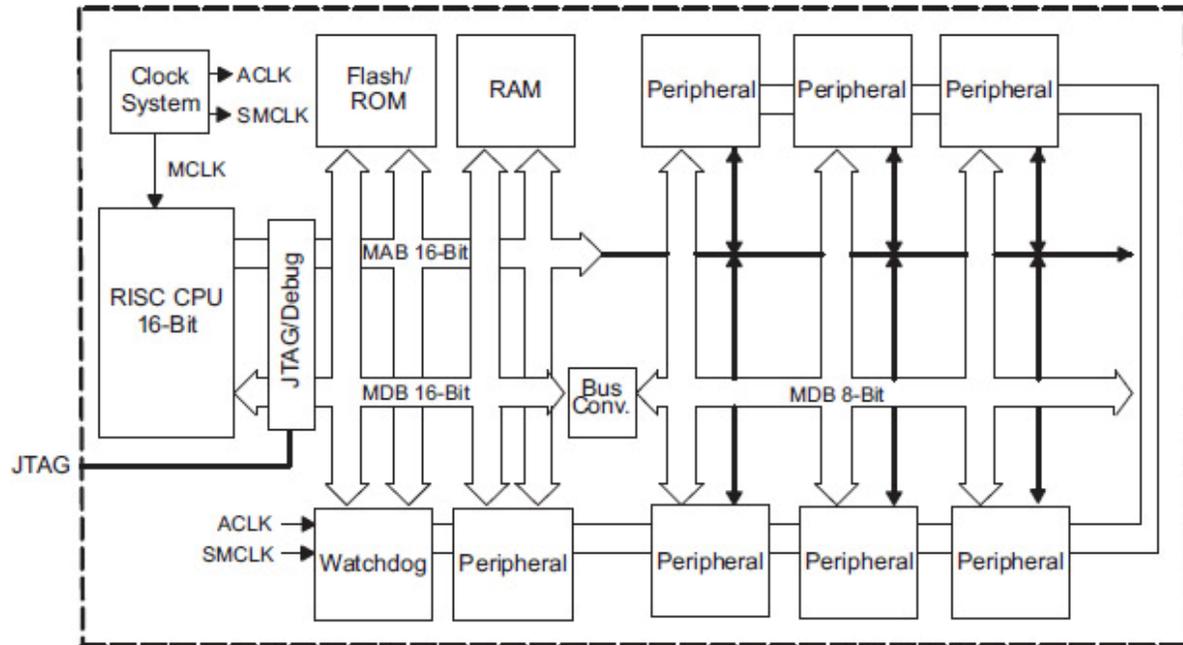


Figura 1. Arquitectura MSP430

-Ref. slau144h.pdf p.26-

La aplicación del desarrollo determinará los periféricos con los que contará el dispositivo. La máxima cantidad de memoria flash con la que cuenta la familia G2XXX es de 128 KB.

El mapa de memoria de la familia se muestra en la figura 2. la organización en bits, Bytes y Words se muestra en la figura 3. En todos los casos, tratándose de Words, la organización es el Byte bajo en dirección par y el Byte alto en la siguiente dirección non.

Los registros de funciones especiales (SFR's) y los periféricos de 8 Bits se acceden en modo Byte, los periféricos de 16 Bits en modo Word y el resto de la memoria puede ser accedida en modo Byte o Word, de manera indistinta.

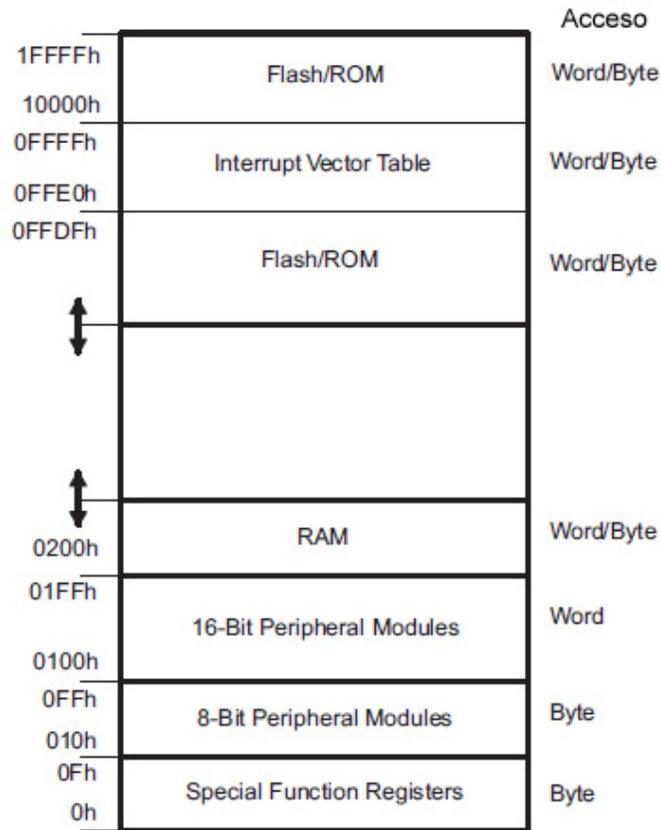


Figura 2. Mapa de Memoria

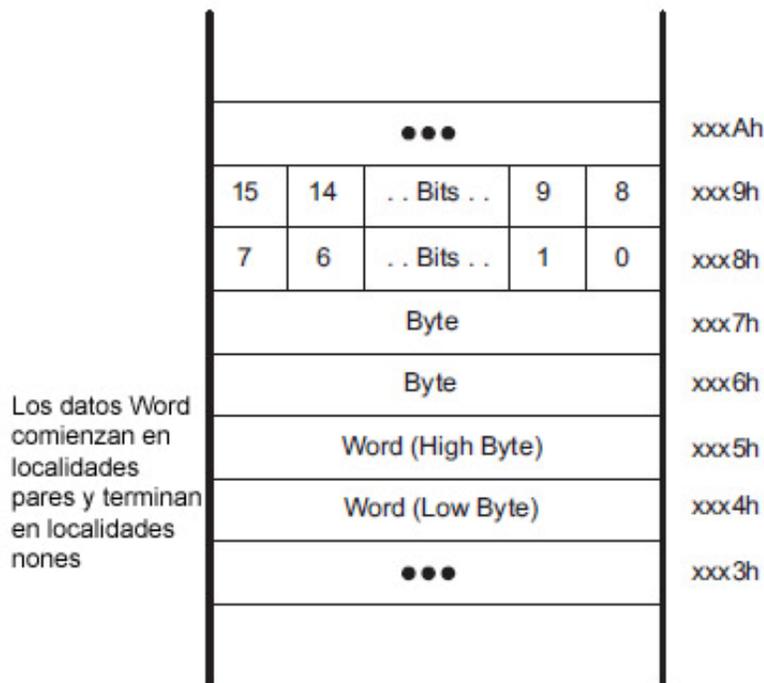


Figura 3. Bits, Bytes y Words en una memoria organizada en Bytes

-Ref. slau144h.pdf p.27,28-

Se usarán los encapsulados DIP de 14 y 20 terminales. Estos dispositivos no cuentan con el módulo cargador de arranque BSL

Dispositivo	EEM Flash	RAM	Timer_A	Comp_A+	Clock	I/O	COM	ADC10	
MSP430G2211	SI	2K	128B	1 x TA2	8 Canales	LF, DCO, VL	10	NO	NO
MSP430G2231	SI	2K	128B	1 x TA2	NO	LF, DCO, VL	10	USI	8 Can
MSP430G2452	SI	8K	256B	1x TA3	8 Canales	LF, DCO, VL	16	USI	8 Can

Organización de memoria:

	MSP430G2452	MSP430G2231	MSP430G2211
SFR's (8 bits)	00h-0Fh	00h-0Fh	00h-0Fh
Periféricos			
8 bits	010h-0FFh	010h-0FFh	010h-0FFh
16 bits	0100-01FFh	0100-01FFh	0100-01FFh
RAM	0200h-02FFh (256)	0200h-027Fh (128)	0200h-027Fh (128)
Memoria de Información	1000h-10FFh (256)	1000h-10FFh (256)	1000h-10FFh (256)
Memoria de Programa	0E000h-0FFBFh	0F800h-0FFBFh	0F800h-0FFBFh
Vectores de Interrupción	0FFC0h-0FFFFh	0FFC0h-0FFFFh	0FFC0h-0FFFFh

REGISTROS DE FUNCIONES ESPECIALES (SFR'S)

Registros de habilitación de interrupción 1 y 2

Address	7	6	5	4	3	2	1	0
00h			ACCVIE	NMIIE			OFIE	WDTIE
			rw-0	rw-0			rw-0	rw-0

WDTIE – Interrupción del temporizador Watchdog habilitada. Inactiva si el modo Watchdog está seleccionado. Activa si el temporizador está configurado en el modo de intervalo.

OFIE – Habilidad de falla del oscilador.

NMIIE – Habilidad de interrupción NMI.

ACCVIE – Habilidad de violación de acceso a la flash.

Address	7	6	5	4	3	2	1	0
01h								

Registros de banderas de Interrupción 1 y 2

Address	7	6	5	4	3	2	1	0
02h				NMIIFG	RSTIFG	PORIFG	OFIFG	WDTIFG
				rw-0	rw-(0)	rw-(1)	rw-1	rw-(0)

WDTIFG – 1 al desbordarse el Watchdog o en violación de la clave de seguridad. 0 al encender Vcc o en condición de reset en el modo reset de la terminal **RST**/NMI.

RSTIFG – Bandera de interrupción Externa. 1 al presentarse un evento en **RST**/NMI cuando se usa en modo RESET. 0 al encender VCC

PORIFG – Power-On reset. 1 al encender VCC

OFIFG – 1 en falla del oscilador (Oscillator Fail)

NMIIFG –1 a través de la terminal **RST**/NMI.

Definición de ortogonalidad:

The ability to use different addressing modes for both source and destination is referred to as instruction *orthogonality*. The '430 is considered to be fully orthogonal, since any instruction can effectively use any addressing mode for both source and destination operands. This flexibility is something of a rarity in small microcontrollers.

Full orthogonality has several advantages. It allows the programmer to write very compact code. Many operations which require multiple statements in other controllers can be accomplished with one properly addressed command in the '430. Also, and perhaps most importantly, this is a very compiler-friendly feature. C constructs, such as pointers, implement much more readily in the '430 than in most other microcontrollers. The downside, however, is added complexity. In order to implement a fully orthogonal set, linear instruction timing is sacrificed (i.e., instructions can take anywhere from 1 to 6 clock cycles, depending on addressing).

RESETS E INTERRUPCIONES

El circuito de reinicio del sistema que se muestra en la figura 4 es el que se encarga de operar tanto la señal de Power-On Reset (POR) como la de Power-Up Clear (PUC). Diferentes eventos activan estas señales de reinicio y existen diferentes condiciones iniciales en función de la señal que se genere.

El POR es un mecanismo de reinicio del sistema que sólo se genera con los siguientes eventos:

- El encendido del microcontrolador.
- Un nivel bajo en la terminal **RST/NMI** cuando se configura en modo de reset.
- Un nivel bajo generado por SVS (supply voltaje supervisor) cuando PORON=1.

PUC se genera después que POR se ha generado, pero no sucede a la inversa. Los siguientes eventos activan el PUC:

- La existencia de la señal de POR.
- La expiración del temporizador Watchdog, sólo en modo Watchdog.
- Una violación en la clave de seguridad del Watchdog timer.
- Una violación en la clave de seguridad de la memoria flash.

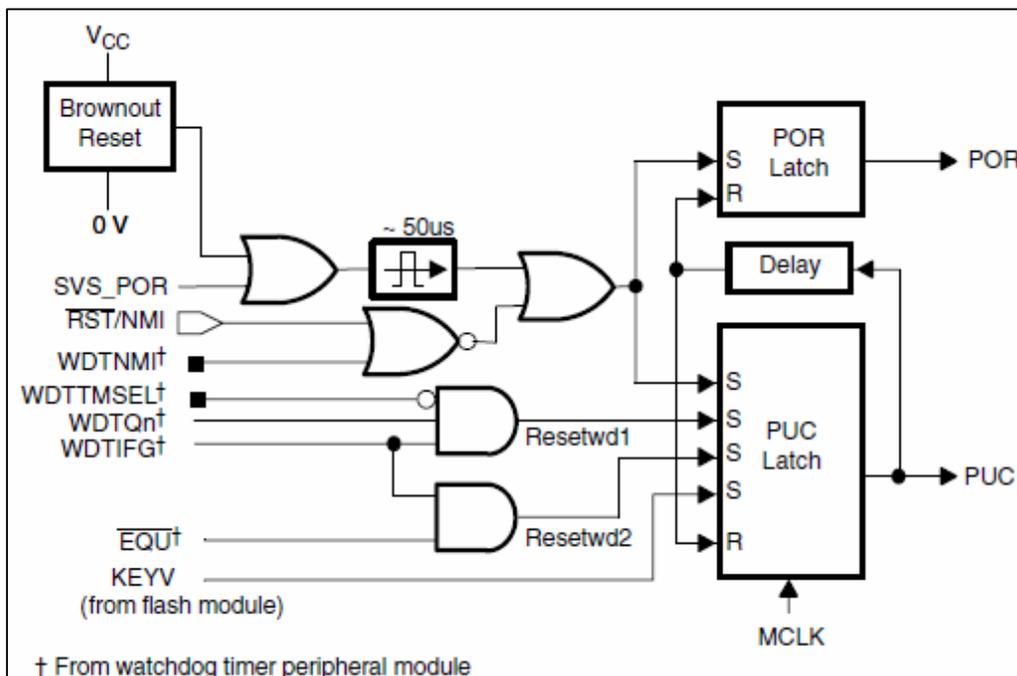


Figura 4 Esquema POR y PUC (Power On Reset, Power On Clear)

CONDICIONES INICIALES DEL DISPOSITIVO DESPUÉS DEL RESET

- La terminal **RST/NMI** se ajusta en el modo reset o reinicio *** definir.
- Las terminales de entrada y salida de los puertos están ajustadas en el modo de entrada por defecto.
- Los distintos periféricos y registros se inicializan como se describe en sus apartados correspondientes.

- El registro de estado o Status Register (SR) se reinicia.
- El temporizador Watchdog se inicia en el modo de Watchdog.
- El Program Counter (PC) se carga con la dirección contenida en la posición del vector de Reset (0FFFEh). La ejecución comienza en esa dirección.

INICIALIZACIÓN DEL SOFTWARE

Después del reinicio del sistema, el software de usuario debe inicializar el microcontrolador para los requerimientos de la aplicación, que generalmente serán los siguientes pasos::

- La inicialización del Stack Pointer (SP), generalmente en la parte superior de la RAM.
- La inicialización del Watchdog a los requisitos de la aplicación.
- Configuración de los módulos periféricos a los requisitos de la aplicación.

Adicionalmente, las banderas del WatchDog, la memoria flash y de falla de oscilador pueden ser consultadas para determinar la causa del reset.

Interrupciones

Las prioridades de las interrupciones se fijan y definen por la disposición de los módulos conectados en cadena como aparece en la figura 5. El módulo más cercano al CPU/NMIRS tiene la mayor prioridad. Las prioridades de las interrupciones determinan que interrupción será atendida cuando más de una estén pendientes al mismo tiempo.

Existen tres tipos de interrupciones:

- System reset o reinicio del sistema.
- No enmascarable, Non maskable, NMI.
- Enmascarable , Maskable.

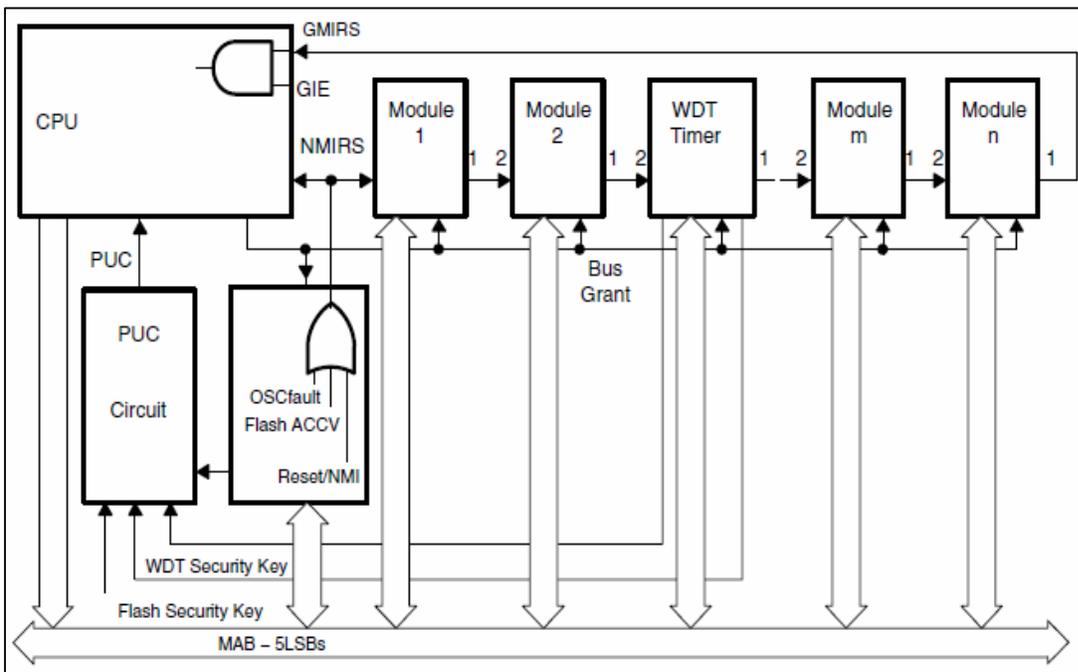


Figura 5. Estructura de las interrupciones

Una interrupción NMI puede ser generada por tres vías distintas:

- Cuando la Terminal **RST**/NMI es configurado en el modo NMI.
- Falla de un oscilador.
- Una violación de acceso a la memoria flash.

RESET/NMI

En el arranque, la terminal **RST**/NMI se configura en el modo RESET. La función de las terminales de **RST** o NMI se selecciona en el Watchdog Control Register (WDTCTL). Si la terminal **RST**/NMI se configura en RESET, el CPU se mantiene en RESET mientras la terminal **RST**/NMI siga en estado 0. Cuando ésta cambia a 1, el CPU inicia la ejecución del programa en la dirección de la palabra almacenada en el vector de reset, 0FFFEh. Si el usuario configura por software la terminal **RST**/NMI, el flanco seleccionado por el bit WDTNMIES genera una interrupción NMI si el bit NMIIE está a 1, estableciendo la bandera NMIIFG a 1.

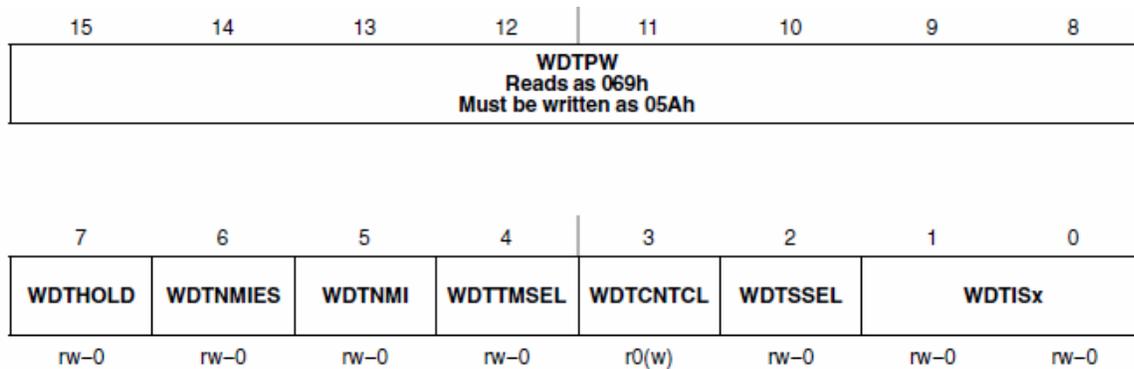


Figura 7. Registro WDTCTL

- WDTNMIES** Bit-6 Selección de flanco NMI del WDT. Éste bit selecciona el flanco para la interrupción NMI cuando WDTNMI = 1. Modificar éste bit solo cuando WDTNMI = 0 para evitar la activación de una NMI accidental.
 0 NMI en flanco de subida
 1 NMI en flanco de bajada
- WDTNMI** Bit-5 Selección de NMI del WDT. Éste bit selecciona la función para el pin **RST**/NMI.
 0 Función reset
 1 Función NMI

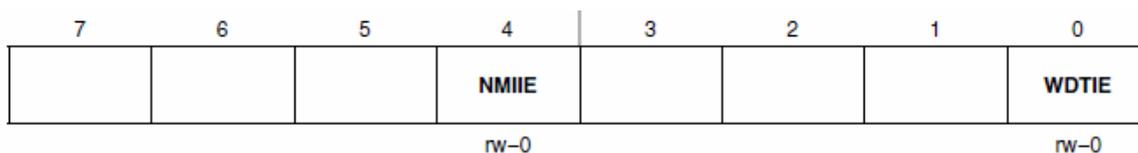


Figura 8 IE1, registro de habilitación de interrupción 1

NMIIE Bit-4 Habilitación de interrupción NMI. Habilita la interrupción NMI. Debido a que otros bits de este registro se pueden usar por otros módulos, se recomienda poner a 1 o a 0 este bit usando las instrucciones BIS.B o BIC.B, en lugar de las instrucciones MOV.B o CLR.B.
 0 Interrupción no habilitada
 1 Interrupción habilitada

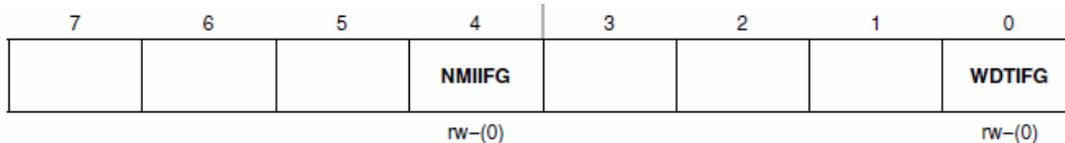


Figura 9 IFG1, registro de banderas de interrupción 1

NMIIFG Bit 4 Bandera de interrupción no enmascarable. En 1 Indica interrupción NMI Pendiente. Debe ser restablecida a 0 en software. Debido a que otros bits de este registro se pueden usar por otros módulos, se recomienda poner a 1 o a 0 este bit usando las instrucciones BIS.B o BIC.B, en lugar de las instrucciones MOV.B o CLR.B.
 0 No hay interrupción pendiente
 1 Interrupcion pendiente

Temporizador de seguridad Watchdog timer Watchdog+

El temporizador Watchdog (WDT) es un temporizador de 16 bits que puede usarse como Watchdog o como temporizador de intervalo. La principal función de éste modulo WDT es llevar a cabo el reinicio controlado del sistema después de que se haya producido un problema en el software. Si el intervalo de tiempo seleccionado expira, se genera un reinicio del sistema. Si la función de Watchdog no se necesita en una aplicación, el módulo se puede configurar como un temporizador de intervalo y puede generar interrupciones en intervalos de tiempo seleccionados.

Las características del temporizador Watchdog son:

- Cuatro intervalos de tiempo seleccionables por software
- Modo de Watchdog
- Modo de intervalos
- El acceso al control WDT está protegido por contraseña
- Control de la función del pin **RST**/NMI
- Fuente de reloj seleccionable
- Se puede detener para ahorrar energía
- Característica de reloj a prueba de fallos en WDT+

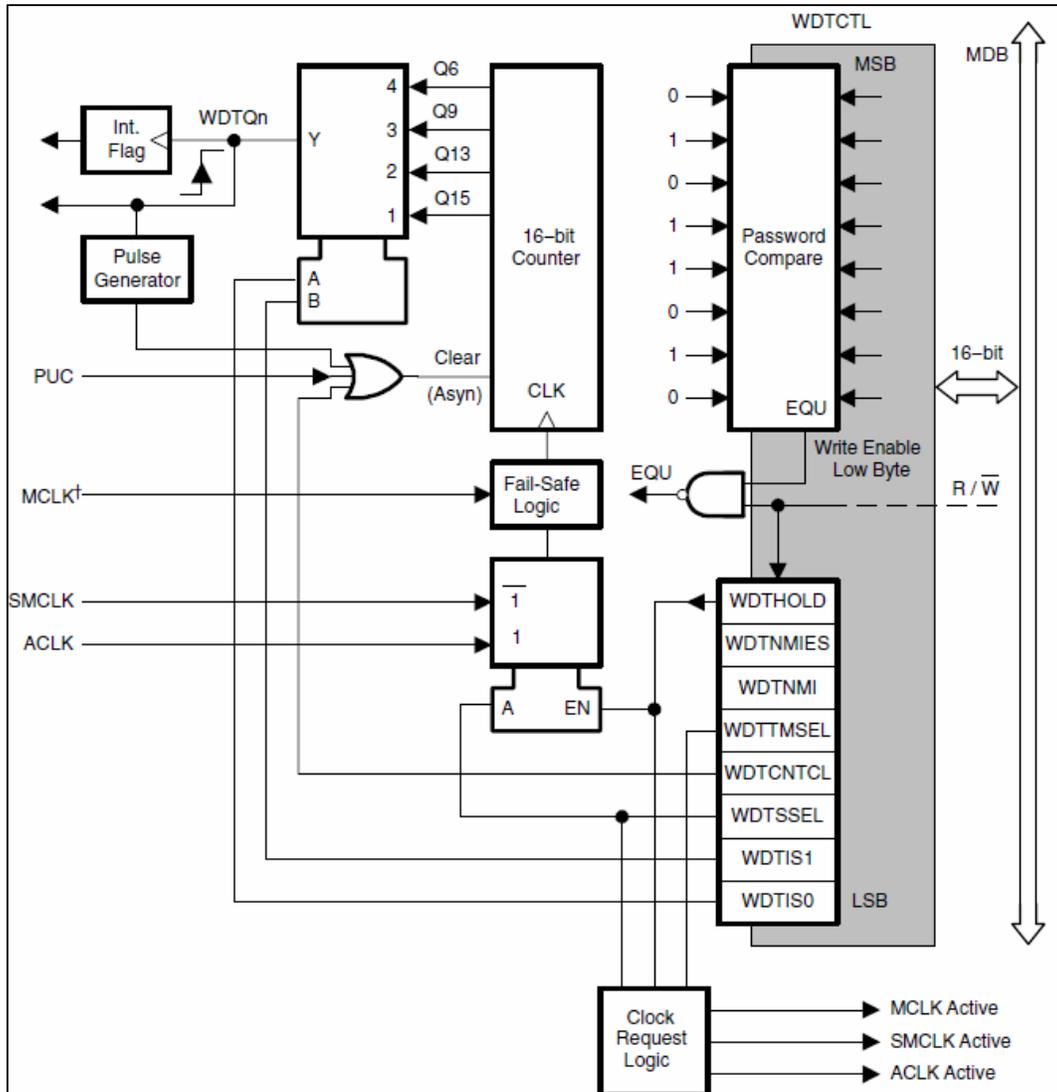


Figura 10 Diagrama de bloques del WatchDog

Después de un POR o PUC, el WDT reinicia en su modo de WatchDog con un ciclo de reset de 32768 ciclos del DCLOCK. El usuario debe deshabilitarlo o limpiarlo antes que este ciclo termine.

Funcionamiento del Watch Dog Timer

Para configurar el WDT como Watchdog o como temporizador de intervalo, se debe utilizar el registro WDTCTL, que es un registro de lectura/escritura de 16-bit protegido por contraseña. Cualquier acceso de lectura/escritura debe usar instrucciones de *word* y los accesos de escritura deben incluir la contraseña de escritura 05Ah en el byte más alto. Cualquier escritura en el registro WDTCTL diferente a 05Ah en el byte más alto es una violación de la clave de seguridad y dispara un PUC. Cualquier lectura en el registro WDTCTL lee como 069h el byte alto. El reloj del WDT+ deberá ser menor o igual que la frecuencia de sistema (MCLK).

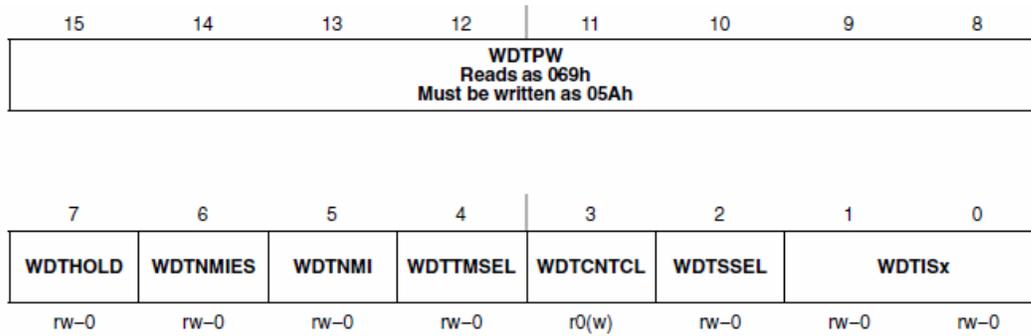


Figura 11 Registro de control del WatchDog Timer

WDTPW	Bits 15-8	Contraseña WDT. Siempre se lee como 096h. Debe ser escrita como 0A5h o se generará un PUC .
WDTTHOLD	Bit-7	Detener WDT. Éste bit detiene el WDT. Establecer WDTTHOLD = 1 cuando WDT no está en uso ahorra energía. 0 WDT Activo 1 WDT Deshabilitado
WDTTMSSEL	Bit-4	Selección del modo de WDT. 0 Modo Watchdog 1 Modo temporizador de intervalo
WDCNTCL	Bit-3	Borrado del contador WDT. Ajustando WDCNTCL = 1 borra el valor del contador quedando 0000h. WDCNTCL Se reinicia automáticamente. 0 Ninguna acción 1 WDCNT = 0000h
WDTSSSEL	Bit-2	Selección de la fuente de reloj de WDT. 0 SMCLK 1 ACLK
WDTISx	Bits 1-0	Selección del intervalo del WDT. Éstos bits eligen el intervalo del temporizador Watchdog para establecer en 1 WDTIFG y/o generar un PUC. 00 Fuente de reloj del WDT / 32768 01 Fuente de reloj del WDT / 8192 10 Fuente de reloj del WDT / 512 11 Fuente de reloj del WDT / 64

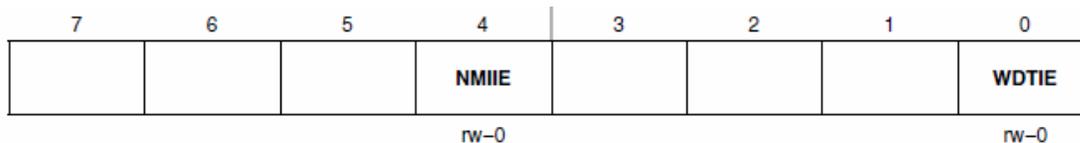


Figura 12 Habilitación del WDT

WDTIE	Bit-0	Habilitación de interrupción WDT. Éste bit habilita a la interrupción WDTIFG para el modo de intervalo. No es necesario poner a 1 para el modo de Watchdog. Debido a que otros bits de este registro se pueden usar por otros módulos, se recomienda poner a 1 o a 0 este bit usando las
--------------	-------	--

instrucciones BIS.B o BIC.B, en lugar de las instrucciones MOV.B o CLR.B.
0 Interrupción no habilitada
1 Interrupción habilitada

El contador Watchdog+

El contador del Watchdog (WDTCNT) es un contador de 16 bits que no es directamente accesible por software. El WDTCNT y los intervalos de tiempo seleccionados se controlan a través del registro de control del temporizador Watchdog WDTCTL.

El reloj del WDTCNT puede ser generado por la señales ACLK o SMCLK. La fuente de reloj se selecciona con el bit WDTSSSEL.

Modo de Watchdog

Después de una condición de PUC, el módulo WDT se configura en el modo de Watchdog con un intervalo inicial de ciclo de reinicio de 32768, usando DCOCLK. El usuario debe configurar, detener o poner a 0 el WDT antes que termine el intervalo inicial de reinicio, o se generará otro PUC. Cuando el WDT se configura para funcionar en el modo Watchdog, genera un PUC bien sea por la escritura de una contraseña incorrecta en WDTCTL o que expire el intervalo de tiempo seleccionado. Un PUC reinicia el WDT a su condición por defecto y configura el pin /NMI en el modo reset.

Modo de temporizador de intervalo

Estableciendo el bit WDTTMSSEL en 1 se selecciona el modo de temporizador de intervalo. Éste modo se puede usar para proporcionar interrupciones periódicas. En el modo de temporizador de intervalo, la bandera WDTIFG se pone a 1 cuando expira el intervalo de tiempo seleccionado. De este modo, el PUC no se genera cuando expira el intervalo de tiempo y el WDTIFG y el bit de habilitación WDTIE permanecen sin cambio.

Cuando los bits WDTIE y GIE están a 1, la bandera WDTIFG solicita una interrupción. La bandera de interrupción WDTIFG se reinicia automáticamente cuando la llamada a la interrupción es atendida, o puede ser reiniciada por software. El vector de interrupción en modo WatchDog es diferente al vector de temporizador de intervalo.

Modificando el Watchdog Timer+

El intervalo del WDT+ debe ser cambiado al mismo tiempo que se establece WDTCNTCL = 1, en una sola instrucción, para evitar una interrupción o PUC no esperados.

El WDT+ debe ser detenido antes de cambiar su fuente de reloj para evitar un posible intervalo incorrecto.

Interrupciones del temporizador WDT+

El WDT utiliza dos bits del SFR para el control de interrupciones, que son, la bandera de interrupción, WDTIFG ubicada en IFG1.0, y la habilitación de interrupción, WDTIE ubicada en IE1.0.

Cuando se utiliza el WDT en modo Watchdog, la bandera WDTIFG genera un reinicio por vector de interrupción. WDTIFG se puede utilizar por la rutina de servicio de interrupción para determinar si el WDT causó que el dispositivo reiniciara. Si la bandera está en 1, significa que el WDT inició la condición de reset, ya sea por la expiración del tiempo o por una violación de la clave de seguridad. Si en caso contrario WDTIFG está a 0 la condición de reset la causó una fuente diferente.

Cuando se utiliza el WDT en modo de temporizador de intervalo, la bandera WDIFG se pone a 1 después de seleccionar el intervalo de tiempo y solicitar una interrupción del temporizador de intervalo si el WDTIE y el GIE están a 1. El vector de interrupción del temporizador de intervalo es diferente del vector de reset utilizado en el modo Watchdog. En el modo de intervalo, el flag WDTIFG se reinicia automáticamente cuando la interrupción se atiende o puede ser reiniciado por software.

WDT+ mejorado, con operación a prueba de fallo del reloj

El módulo WDT+ proporciona una funcionalidad mejorada sobre el WDT. Éste módulo proporciona una característica de sincronización a prueba de fallos para asegurar que el reloj para el WDT+ no se pueda desactivar mientras está en modo Watchdog. Esto quiere decir que los modos de bajo consumo se pueden ver afectados por la elección del reloj WDT+. Por ejemplo, si ACLK es la fuente de reloj del WDT+, LPM4 no está disponible, porque el WDT+ impide al ACLK ser deshabilitado. Además, si ACLK o SMCLK fallan mientras generan el WDT+, la fuente de reloj de éste se cambia automáticamente a MCLK. En este caso, si MCLK se genera con un cristal de cuarzo y éste falla, la característica a prueba de fallos activa al DCO y lo utiliza como fuente para el MCLK.

Cuando el WDT+ se utiliza en modo de temporizador de intervalo, no aplican estas características de seguridad

Funcionamiento en modos de bajo consumo

Los dispositivos de MSP430 tienen varios modos de bajo consumo que pueden disponer de diferentes señales de reloj. Los requisitos de las aplicaciones del usuario y el tipo del reloj utilizado determinan la configuración del WDT. Por ejemplo, no se debe configurar en modo Watchdog con SMCLK como fuente de reloj si se quiere usar LPM3, porque el WDT+ conservara a SMCLK activo en LPM3 y entonces no funcionaría WDT. Si WDT+ se genera desde SMCLK, éste permanece habilitado durante LPM3, lo cual incrementa el consumo de corriente en el modo LPM3. Cuando el temporizador Watchdog no es necesario, el bit WDT HOLD se puede utilizar para deshabilitar WDT CNT, reduciendo el consumo de energía.

Modos de operación

El microcontrolador MSP430 tiene un modo activo de operación y cinco modos de selección de programa de bajo consumo. Un evento de interrupción puede despertar el dispositivo de cualquiera de los modos de bajo consumo, atendiendo a la solicitud y restaurando de nuevo al modo de bajo consumo en el retorno del programa de la interrupción.

Los cinco modos de bajo consumo se configuran con los bits CPUOFF, OSCOFF, SCG0 y SCG1 en el registro de estado.

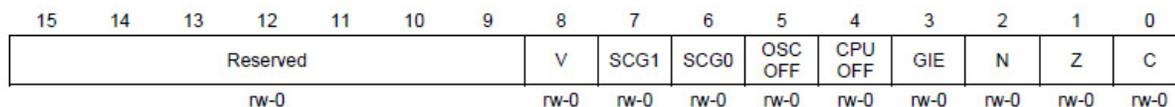


Figura 13. Registro de estado SR y bits de control de modos de operación

La ventaja de incluir estos bits de control de los modos de funcionamiento en el SR, es que el modo de funcionamiento presente se almacena en la pila durante la rutina de servicio de interrupción. El flujo de programa vuelve al modo de funcionamiento anterior si el valor del SR guardado no se altera durante la rutina de servicio de interrupción. El flujo de programa puede ser devuelto a un modo de funcionamiento diferente mediante la

manipulación del valor del SR de la pila dentro de la rutina de servicio de interrupción. Se puede acceder con cualquier instrucción a los bits de control de modos de funcionamiento y a la pila.

Al ajustar cualquiera de los bits de control de modos de funcionamiento, el modo elegido entra en vigor inmediatamente. Los periféricos que funcionan con algún reloj, se desactivan hasta que se vuelve a activar el correspondiente reloj. Los periféricos también se pueden desactivar con su ajuste individual del registro de control. Todos los puertos de entrada y salida y los registros de memoria RAM permanecen invariables. La reactivación es posible por medio de cualquier interrupción habilitada.

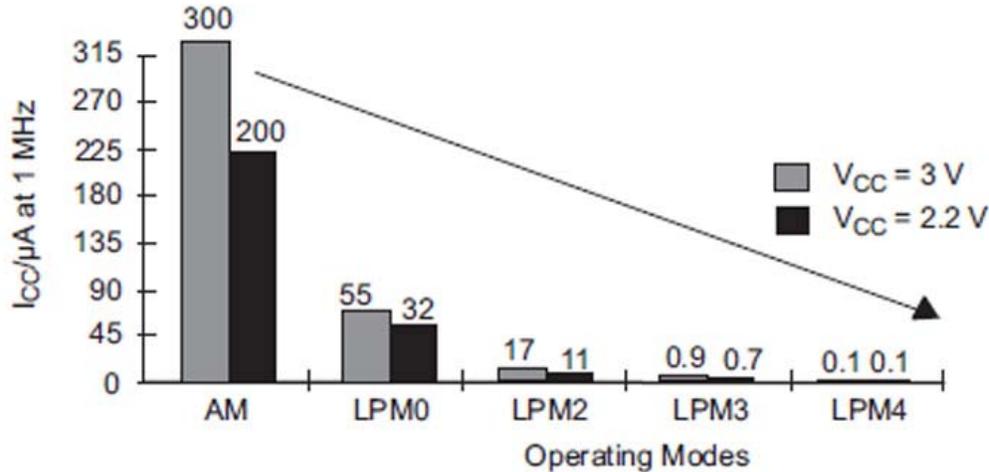


Figura 14. Consumo vs. Modos de operación

Los modos de funcionamiento son los siguientes:

SCG1	SCG0	OSCOFF	CPUOFF	Mode	CPU and Clocks Status
0	0	0	0	Active	CPU is active, all enabled clocks are active
0	0	0	1	LPM0	CPU, MCLK are disabled, SMCLK, ACLK are active
0	1	0	1	LPM1	CPU, MCLK are disabled. DCO and DC generator are disabled if the DCO is not used for SMCLK. ACLK is active.
1	0	0	1	LPM2	CPU, MCLK, SMCLK, DCO are disabled. DC generator remains enabled. ACLK is active.
1	1	0	1	LPM3	CPU, MCLK, SMCLK, DCO are disabled. DC generator disabled. ACLK is active.
1	1	1	1	LPM4	CPU and all clocks disabled

Figura 15. Modos de operación

Active mode (AM)

- CPU activo. Todos los relojes se activan.

Low-power mode 0 (LPM0)

- El CPU se desactiva.
- El reloj de sub-sistema (SMCLK) y auxiliar (ACLK) permanecen activos.
- El reloj MCLK maestro se desactiva.

Low-power mode 1 (LPM1)

- El CPU se desactiva.
- El reloj maestro MCLK se desactiva.
- DCO y el generador de DC se deshabilitan si el DCO no se usa para generar SMCLK
- El reloj auxiliar ACLK permanece activo.

Low-power mode 2 (LPM2)

- El CPU, MCLK y SMCLK se deshabilitan.
- El DCO se desactiva. El generador DC permanece habilitado.
- El ACLK permanece activo.

Low-power mode 3(LPM3)

- El CPU, MCLK y SMCLK se deshabilitan.
- El DCO y el generador DC se deshabilitan.
- El ACLK permanece activo.

Low-power mode 4 (LPM4)

- El CPU se desactiva.
- Todas las fuentes de reloj se deshabilitan.

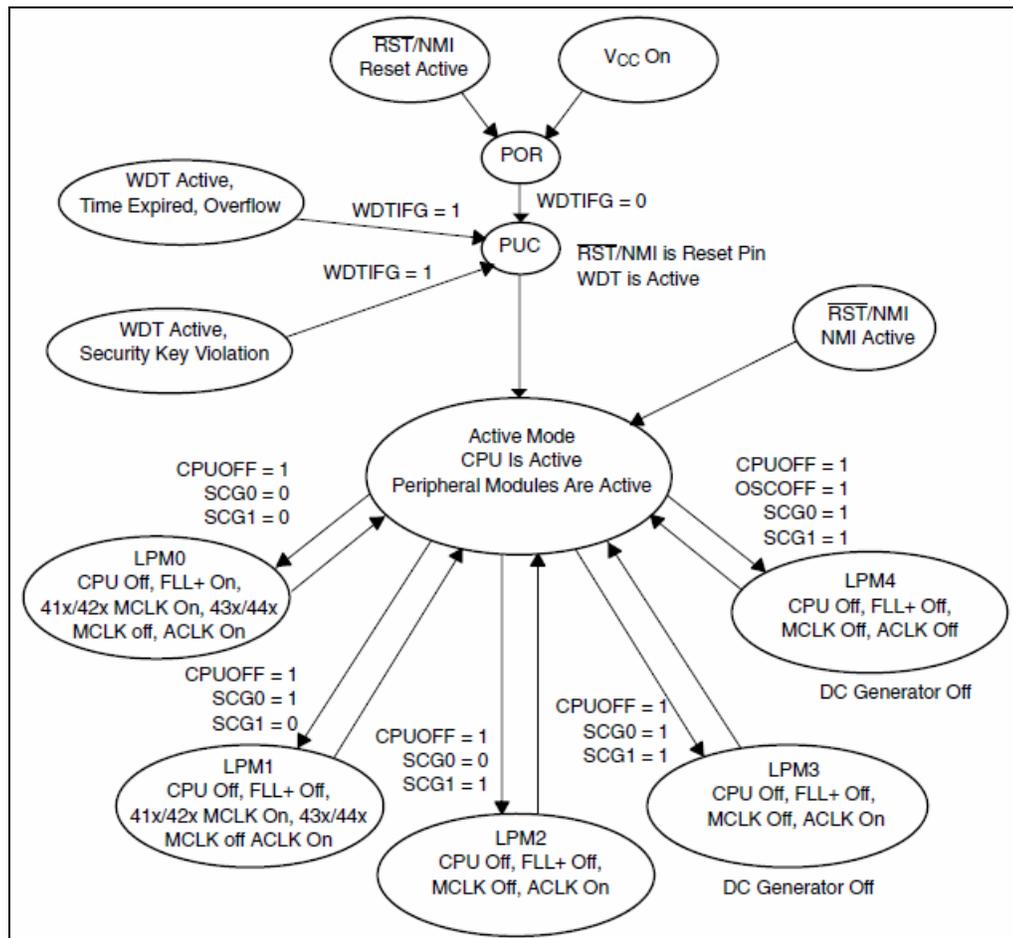


Figura 16. Secuencia de modo activo a modos de bajo consumo.

Sistema de reloj. Basic Clock Module +

Los microcontroladores encuentran un compromiso entre relojes de alta velocidad, que puedan ser habilitados y deshabilitados rápidamente para conservar energía, que no necesitan ser particularmente precisos y relojes de baja velocidad y bajo consumo, como los usados para monitorear tiempo real, donde la precisión de suma importancia. La familia MSP430 cuenta con un sistema de reloj que soluciona el compromiso alto rendimiento, bajo consumo y precisión en la frecuencia al usar tres relojes internos

- Master clock o **MCLK**, se usa en el CPU y algunos periféricos.
- Subsystem Master Clock o **SMCLK**, usado por los periféricos.
- Auxiliary Clockm **ACLK**, también distribuido a los periféricos.

De manera típica, SMCLK opera a la misma frecuencia que MCLK, ambos en el rango de MHz y ambos son generados por un Oscilador controlado Digitalmente o DCO.

Una de las principales características del DCO es que arranca muy rápido a máxima velocidad, en el orden de unos pocos microsegundos (1 uS en la familia G2XXX), siendo esto primordial para sistemas de muy bajo consumo de potencia.

ACLK por lo general se deriva de un cristal de cuarzo de los usados en relojes de pulsera, y por lo tanto, opera a una frecuencia mucho menor.

La mayoría de los periféricos pueden seleccionar la fuente de reloj que usarán, ya sea SMCLK o ACLK.

Nota: No todas las opciones del módulo básico de reloj están disponibles para todas las familias del MSP430:

MSP430x20xx, MSP430G2xxx: LFXT1 no soporta el modo HF, XT2 no está implementado, no soporta el modo ROSC.

MSP430x21x1: El oscilador interno LP/LF no está implementado, XT2 no está implementado, no soporta el modo ROSC.

MSP430x21x2: XT2 no está implementado.

MSP430x22xx, MSP430x23x0: XT2 no está implementado.

Después de un PUC, la fuente para MCLK y SMCLK es DCOCKL con una frecuencia de ~ 1,1 MHz, y la fuente de ACLK es LFXT1CLK en modo LF con una capacitancia interna de 6 pF. Los bits de control SCG0, SCG1, OSCOFF, y CPUOFF, en el registro de estado, configuran los modos de operación y habilitan o deshabilitan porciones del módulo básico de reloj.

Los registros DCOCTL, BCCTL1, BCCTL2, y BCCTL3 controlan la operación del módulo básico de reloj.

Aplicaciones en baja potencia

Existen requisitos conflictivos en aplicaciones de dispositivos operados a baterías:

- Una frecuencia de reloj baja para conservar energía y funciones de tiempo real.

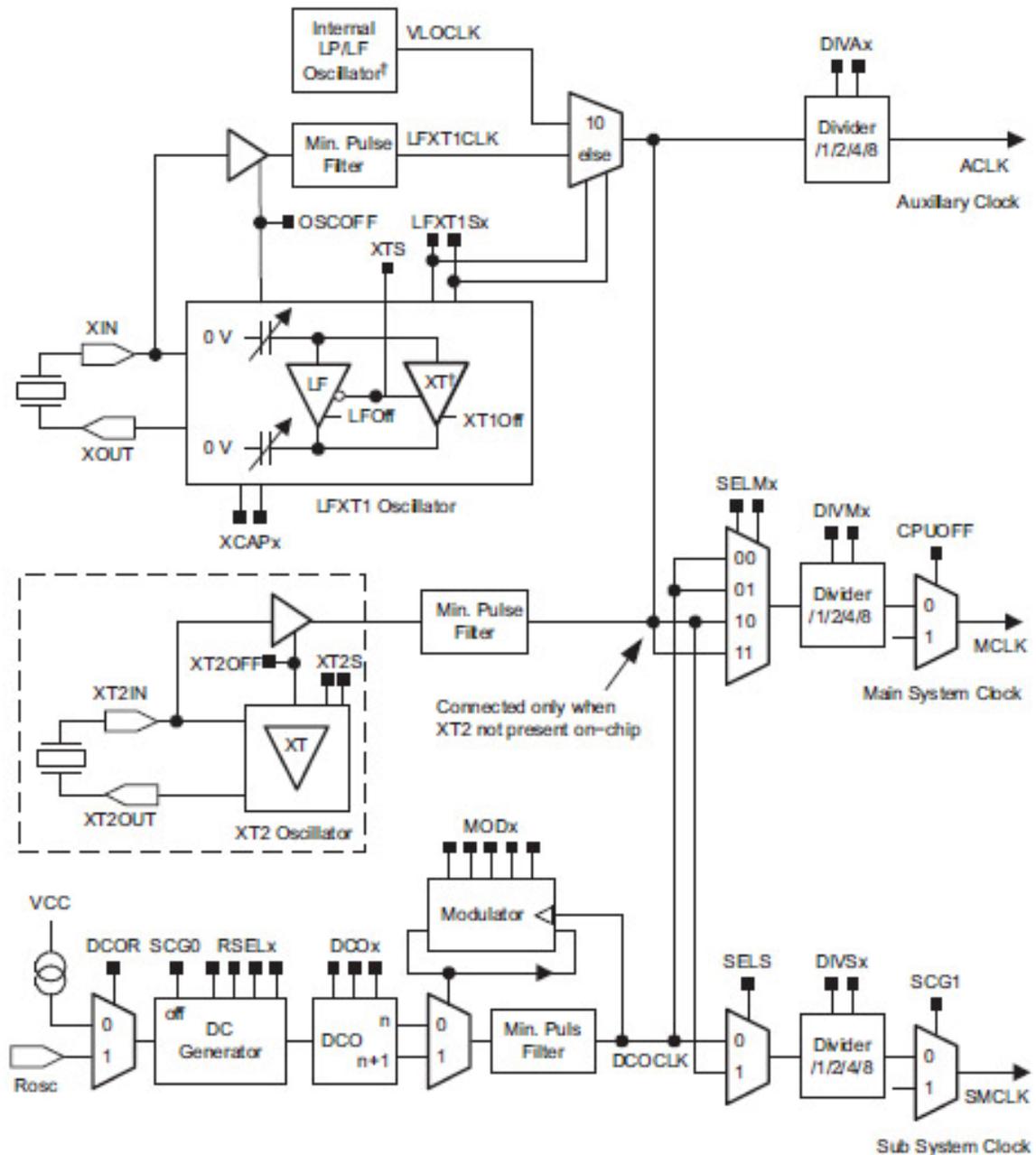


Figura 17. Diagrama de bloques del sistema del módulo básico de reloj +.

- Una frecuencia de reloj alta para una reacción rápida a eventos y capacidad de respuesta a procesos súbitos y que requieren respuestas veloces.
- Estabilidad del reloj en el rango de la temperatura de operación y voltaje de alimentación.

El módulo básico de reloj + maneja los requisitos anteriores permitiendo al usuario seleccionar la fuente de las tres señales disponibles: ACLK, MCLK y SMCLK. Para una operación óptima en baja potencia, ACLK puede ser generada con un cristal de reloj de

32768 Hz, proveyendo una base de tiempo estable para el sistema y operación en espera (stand-by) , o también puede ser generada con oscilador de baja frecuencia interno si la precisión no es un requisito fundamental.

Hay que hacer notar que ACLK requiere un cristal externo, que viene incluido con muchos módulos de demostración del MSP430, tal como el Launch Pad. Sin embargo, muchas familias incluyen un oscilador de muy baja potencia y baja frecuencia (VLO) que puede ser seleccionada como fuente del ACLK si no hay cristal.

MCLK y SMCLK son generados por XT2 o por el DCO, que en varias familias es controlado a su vez por un FLL (frequency locked loop). Esto enclava la frecuencia en x32 la frecuencia de ACLK, que es de ~1.1 MHz (1048576 Hz) .

Pero si no hay cristal significa que no hay ACLK,. Eso significaría que el CPU está detenido. Sin embargo, el generador de reloj detectará una falla en ACLK y permite al DCO funcionar en su frecuencia mas baja, de modo que MCLK y SMCLK permanecen disponibles. Muchas aplicaciones no requieren de una frecuencia precisa, de modo que el DCO brinda una operación aceptable, pero como al momento no hay una alternativa a los cristales de cuarzo para una frecuencia precisa, muchos sistemas requieren tener un cristal en ACLK.

Registros de control del sistema del módulo básico de reloj +.

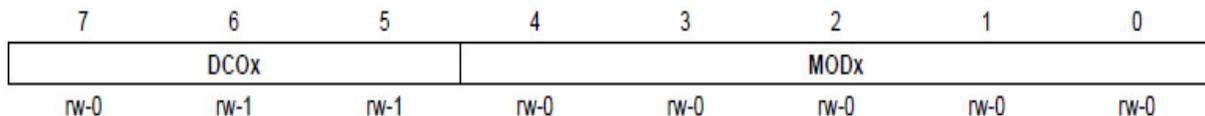


Figura 18. DCOCTL Registro de control del DCO

- DCOx** bits 7-5 Estos bits seleccionan cual de las 8 frecuencias discretas dentro del rango definido por RSELx es establecida

- MODx** bits 4-0 Selección del modulador. Estos bits definen que tan seguido la frecuencia f_{DCO+1} es usada dentro de un periodo de 32 ciclos DCOCLK. Durante los siguientes ciclos (32 – MOD) se usa la frecuencia f_{DCO} . No disponible cuando DCOx=7

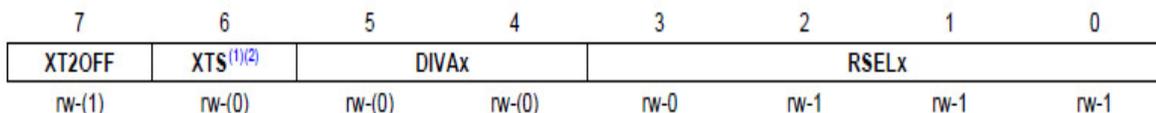


Figura 19. BCSTL1 Basic Clock System Control Register 1

- XT2OFF** bit 7 Este bit enciende o apaga el oscilador Xt2
 0 XT2 encendido
 1 XT2 apagado.

XTS	bit 6	Selección de modo de LFXT1 0 Modo para cristales de baja frecuencia 1 Modo para cristales de alta frecuencia
DIVAs	bits 5-4	Divisor para ACLK 00 /1 01 /2 10 /4 11 /8
RSELx	bits 3-0	Selección de rango. 16 diferentes rangos de frecuencias están disponibles. El rango más bajo de frecuencia es seleccionado cuando RSELx=0. Rsel3 se ignora cuando DCOR=1

- (1) XTS=1 no es soportado con dispositivos MSP430X20XX
- (2) Este es un bit reservado en dispositivos MSP430AFE2XX

CPU

El CPU MSP430 tiene una arquitectura RISC de 16-bits con 51 instrucciones . Todas las operaciones, además de las instrucciones de control de programa, se llevan a cabo como operaciones de registro. Existen siete modos de direccionamiento para el operando de la fuente y cuatro modos de direccionamiento para el operando de destino.

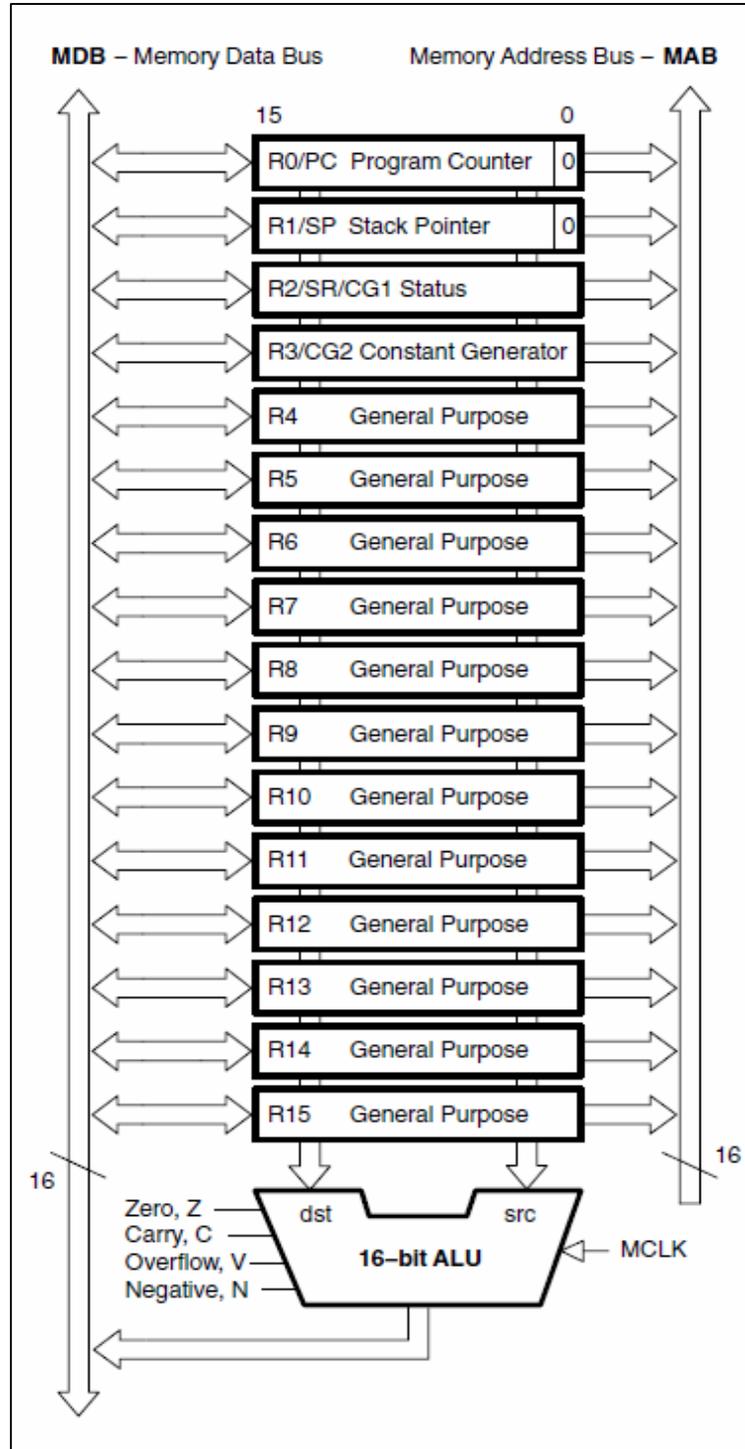


Figura 20. Registros y Modelo de programación del CPU MSP430

Las características del CPU incluyen:

- 27 instrucciones implementadas en hardware y 27 emuladas usando CG1 y CG2 con 7 modos de direccionamiento
- Arquitectura ortogonal: todas las instrucciones se ejecutan en todos los modos de direccionamiento.
- Acceso total a los registros, incluyendo al Program Counter, Status Register, y Stack Pointer
- Operaciones entre registros de un ciclo.
- Registros de 16 bits, que reducen accesos a memoria,
- Bus de dirección de 16 bits, con acceso y salto a todo el rango de memoria.
- Generador de constantes que provee 6 de las constantes más usadas.
- Transferencia directa entre memoria sin registros de respaldo intermedios
- Formatos de instrucción de Byte y Word.

Registros

El CPU está integrado por 16 registros de 16 bits que reducen el tiempo de ejecución de instrucciones. El tiempo de ejecución de la operación registro a registro es un ciclo del reloj del CPU. Cuatro de los registros, R0 a R3, se dedican a las funciones de contador de programa, apuntador de pila, registro de estado, y un generador de constantes, respectivamente. Los registros restantes son de propósito general. Los periféricos están conectados al CPU por medio de los buses datos, dirección y control, y se puede manejar con todas las instrucciones.

- **CONTADOR DE PROGRAMA (PC)**

El contador de programa de 16 bits (PC/R0) apunta a la siguiente instrucción que se va a ejecutar. Cada instrucción utiliza un número par de Bytes (2, 4 ó 6), incrementándose el PC de manera consecuente. El acceso a instrucciones en el espacio de 64 KB se efectúa en fronteras de palabras y el PC se alinea en direcciones pares.

- **STACK POINTER (SP)**

El CPU utiliza el Stack Pointer (SP/R1) para almacenar las direcciones de retorno de las llamadas de subrutinas e interrupciones. Utiliza una combinación de predecremento y postincremento. Además puede ser usado por el software con todas las instrucciones y modos de direccionamiento. Es inicializado por el usuario y debe ser alienado en direcciones pares.

- **REGISTRO DE ESTADO (SR)**

El registro de estado (SR/R2), utilizado como registro fuente o destino, puede ser utilizado en el modo de registro únicamente con instrucciones de Word. El resto de combinaciones de modos de direccionamiento se utilizan para apoyar el generador de constantes.

- **REGISTROS GENERADORES DE CONSTANTES (CG1 Y CG2)**

Seis constantes comúnmente usadas son generadas por los registros R2 y R3, sin necesitar una palabra de 16-bit adicional del código de programa. Las constantes son seleccionadas con el modo de direccionamiento del registro de origen (As – Addressing source).

Register	As	Constant	Remarks
R2	00	-----	Register mode
R2	01	(0)	Absolute address mode
R2	10	00004h	+4, bit processing
R2	11	00008h	+8, bit processing
R3	00	00000h	0, word processing
R3	01	00001h	+1
R3	10	00002h	+2, bit processing
R3	11	0FFFFh	1, word processing

Figura 21. Constantes generadas por R2 y R3.

- **REGISTROS DE PROPÓSITO GENERAL**

Son doce registros de propósito general, R4 a R15. Todos se pueden utilizar como registros de datos, apuntadores de dirección o índice, y pueden ser accedidos mediante instrucciones de Byte o Word.

Modos de direccionamiento

Los valores EDE, LEO, TONI, TOM y LEO son usados como etiquetas genéricas sin ningún otro significado.

Existen siete modos de direccionamiento para el operando fuente y cuatro modos para el operando destino que pueden direccionar todo el espacio de direcciones sin excepciones.

As/Ad	Addressing Mode	Syntax	Description
00/0	Register mode	Rn	Register contents are operand
01/1	Indexed mode	X(Rn)	(Rn + X) points to the operand. X is stored in the next word.
01/1	Symbolic mode	ADDR	(PC + X) points to the operand. X is stored in the next word. Indexed mode X(PC) is used.
01/1	Absolute mode	&ADDR	The word following the instruction contains the absolute address. X is stored in the next word. Indexed mode X(SR) is used.
10/-	Indirect register mode	@Rn	Rn is used as a pointer to the operand.
11/-	Indirect autoincrement	@Rn+	Rn is used as a pointer to the operand. Rn is incremented afterwards by 1 for .B instructions and by 2 for .W instructions.
11/-	Immediate mode	#N	The word following the instruction contains the immediate constant N. Indirect autoincrement mode @PC+ is used.

Figura 22. Modos de direccionamiento

Los modos soportados por el CPU son:

- Registro
- Indexado
- Simbólico
- Absoluto
- Registro indirecto
- Indirecto con autoincremento
- Inmediato

Las instrucciones pueden ser orientadas a byte (MOV.B) o a palabra (MOV). Si no hay prefijo, se interpreta como instrucción de Word.

1) Modo de Registro

Longitud: Una o dos palabras

Transfiere el contenido del registro R10 al registro R11

Valido para fuente y destino

```
MOV R10,R11
```

Assembler Code	Content of ROM
MOV R10,R11	MOV R10,R11
<p>Length: One or two words</p> <p>Operation: Move the content of R10 to R11. R10 is not affected.</p> <p>Comment: Valid for source and destination</p> <p>Example: MOV R10,R11</p>	
Before:	After:
R10 0A023h	R10 0A023h
R11 0FA15h	R11 0A023h
PC PC_{old}	PC PC_{old} + 2

Figura 23. Modo registro.

El dato en los registros puede ser accedido usando instrucciones de byte o Word. Si se usan instrucciones de byte, el byte alto siempre tendrá 0 como resultado. Los bits de estado se manejan de acuerdo al resultado de las instrucciones de byte.

2) Modo Indexado

Longitud: Dos o tres palabras.

Mueve el contenido de la dirección fuente (R5+2) a la dirección destino (R6 + 6) Los registros de fuente y destino (R5 y R6) no son afectados. El PC se incrementa automáticamente de modo que el programa continúa con la siguiente instrucción.

Valido para fuente y destino

MOV 2(R5),6(R6)

Assembler Code		Content of ROM	
MOV 2(R5),6(R6)		MOV X(R5),Y(R6)	
		X = 2	
		Y = 6	

Length: Two or three words

Operation: Move the contents of the source address (contents of R5 + 2) to the destination address (contents of R6 + 6). The source and destination registers (R5 and R6) are not affected. In indexed mode, the program counter is incremented automatically so that program execution continues with the next instruction.

Comment: Valid for source and destination

Example: MOV 2(R5),6(R6);

Before:		After:	
Address Space	Register	Address Space	Register
0FF16h 00006h	R5 01080h	0FF16h 00006h	R5 01080h
0FF14h 00002h	R6 0108Ch	0FF14h 00002h	R6 0108Ch
0FF12h 04596h	PC	0FF12h 04596h	
01094h 0xxxxh	0108Ch +0006h	01094h 0xxxxh	
01092h 05555h	01092h	01092h 01234h	
01090h 0xxxxh		01090h 0xxxxh	
01084h 0xxxxh	01080h +0002h	01084h 0xxxxh	
01082h 01234h	01082h	01082h 01234h	
01080h 0xxxxh		01080h 0xxxxh	

3) Modo Simbólico

Longitud: Dos o tres palabras.

Mueve el contenido de la dirección fuente EDE, (contenido del PC + X), a la dirección destino TONI (contenido del PC + Y). El PC se incrementa automáticamente. El ensamblador calcula automáticamente los corrimientos (offsets) X y Y. El PC se incrementa automáticamente a la siguiente dirección

Valido para fuente y destino

MOV EDE,TONI ;Dirección Fuente EDE = 0F016h
;Direccion destino TONI = 01114h

Assembler Code		Content of ROM	
MOV EDE,TONI		MOV X(PC),Y(PC) X = EDE - PC Y = TONI - PC	
Length:	Two or three words		
Operation:	Move the contents of the source address EDE (contents of PC + X) to the destination address TONI (contents of PC + Y). The words after the instruction contain the differences between the PC and the source or destination addresses. The assembler computes and inserts offsets X and Y automatically. With symbolic mode, the program counter (PC) is incremented automatically so that program execution continues with the next instruction.		
Comment:	Valid for source and destination		
Example:	MOV EDE,TONI ;Source address EDE = 0F016h ;Dest. address TONI=01114h		
Before:	Address Space	Register	After: Address Space Register
	0FF16h 011FEh		0FF16h 011FEh PC
	0FF14h 0F102h		0FF14h 0F102h
	0FF12h 04090h PC		0FF12h 04090h
	0F018h 0xxxxh	0FF14h	0F018h 0xxxxh
	0F016h 0A123h	+0F102h	0F016h 0A123h
	0F014h 0xxxxh	-0F016h	0F014h 0xxxxh
	01116h 0xxxxh	0FF16h	01116h 0xxxxh
	01114h 05555h	+011FEh	01114h 0A123h
	01112h 0xxxxh	-01114h	01112h 0xxxxh

4) Modo Absoluto

Longitud: Dos o tres palabras.

Mueve el contenido de la localidad de memoria fuente EDE a la localidad de memoria destino TONI. El PC se incrementa automáticamente.

Valido para fuente y destino

MOV &EDE,&TONI ;Dirección fuente EDE = 0F016h
;Dirección destino TONI = 01114h

Assembler Code		Content of ROM	
MOV &EDE, &TONI		MOV X(0), Y(0) X = EDE Y = TONI	
Length:	Two or three words		
Operation:	Move the contents of the source address EDE to the destination address TONI. The words after the instruction contain the absolute address of the source and destination addresses. With absolute mode, the PC is incremented automatically so that program execution continues with the next instruction.		
Comment:	Valid for source and destination		
Example:	MOV &EDE, &TONI ;Source address EDE=0F016h, ;dest. address TONI=01114h		
Before:	Address Space	Register	After:
			Address Space
			Register
			PC
0FF16h	01114h		0FF16h
0FF14h	0F016h		0FF14h
0FF12h	04292h	PC	0FF12h
0F018h	0xxxxh		0F018h
0F016h	0A123h		0F016h
0F014h	0xxxxh		0F014h
01116h	0xxxxh		01116h
01114h	01234h		01114h
01112h	0xxxxh		01112h

Este modo de direccionamiento es principalmente para módulos periféricos de hardware con una dirección fija absoluta. Son direccionados en modo absoluto para asegurar portabilidad del software.

5) Modo Indirecto

Longitud: Una o dos words

Mueve el contenido de la localidad de memoria fuente (contenido de R10), a la localidad destino, contenido de R11. Los registros no son modificados.

Valido solo para el operando fuente. El sustituto para el operando destino es 0(Rd)

MOV.B @R10,0(R11)

Assembler Code		Content of ROM	
MOV @R10,0(R11)		MOV @R10,0(R11)	
Length: One or two words			
Operation: Move the contents of the source address (contents of R10) to the destination address (contents of R11). The registers are not modified.			
Comment: Valid only for source operand. The substitute for destination operand is 0(Rd).			
Example: MOV.B @R10,0(R11)			
Before:	Address Space	Register	After:
	0xxxxh		Address Space
0FF16h	0000h	R10 0FA33h	0FF16h
0FF14h	04AEBh	PC R11 002A7h	0FF14h
0FF12h	0xxxxh		0FF12h
0FA34h	0xxxxh		0FA34h
0FA32h	05BC1h		0FA32h
0FA30h	0xxxxh		0FA30h
002A8h	0xxh		002A8h
002A7h	012h		002A7h
002A6h	0xxh		002A6h

6) Modo Indirecto con Autoincremento

Longitud: Una o dos words

Mueve el contenido de la localidad de memoria fuente (contenido de R10), a la localidad de memoria destino (contenido de R11). R10 se incrementa en uno para operaciones de Byte o se incrementa en dos para operaciones de Word después del fetch, apuntando a la próxima dirección. Esto evita pérdida de ciclos en la siguiente instrucción. Esto es útil para manejo de tablas.

Valido solo para el operando fuente. El sustituto para el operando destino es 0(Rd) mas una segunda unstruccion INCD Rd.

Valido solo para el operando fuente

MOV @R10+,0(R11)

Assembler Code		Content of ROM	
MOV @R10+,0(R11)		MOV @R10+,0(R11)	
Length:	One or two words		
Operation:	Move the contents of the source address (contents of R10) to the destination address (contents of R11). Register R10 is incremented by 1 for a byte operation, or 2 for a word operation after the fetch; it points to the next address without any overhead. This is useful for table processing.		
Comment:	Valid only for source operand. The substitute for destination operand is 0(Rd) plus second instruction INCD Rd.		
Example:	MOV @R10+,0(R11)		
Before:	Address Space	Register	After:
			Address Space
			Register
0FF18h	0xxxxh	R10	0FF18h
0FF16h	00000h	R11	0FF16h
0FF14h	04ABBh	PC	0FF14h
0FF12h	0xxxxh		0FF12h
0FA34h	0xxxxh		0FA34h
0FA32h	05BC1h		0FA32h
0FA30h	0xxxxh		0FA30h
010AAh	0xxxxh		010AAh
010A8h	01234h		010A8h
010A6h	0xxxxh		010A6h

7) Modo Inmediato

Longitud: Dos o tres words

Mueve el valor inmediato #45h, que esta en la palabra siguiente a la instrucción, a la direccion de destino TONI

Valido solo para el operando fuente

MOV #45h,TONI

Assembler Code		Content of ROM	
MOV #45h, TONI		MOV @PC+, X (PC)	
		45	
		X = TONI - PC	
Length:	Two or three words It is one word less if a constant of CG1 or CG2 can be used.		
Operation:	Move the immediate constant 45h, which is contained in the word following the instruction, to destination address TONI. When fetching the source, the program counter points to the word following the instruction and moves the contents to the destination.		
Comment:	Valid only for a source operand.		
Example:	MOV #45h, TONI		
Before:	Address Space	Register	After:
			Address Space
			Register
			PC
0FF16h	01192h		0FF18h 0xxxxh
0FF14h	00045h		0FF16h 01192h
0FF12h	040B0h	PC	0FF14h 00045h
			0FF12h 040B0h
010AAh	0xxxxh		010AAh 0xxxxh
010A8h	01234h	0FF16h +01192h 010A8h	010A8h 00045h
010A6h	0xxxxh		010A6h 0xxxxh

Conjunto de Instrucciones. († Instrucciones emuladas)

Mnemonic	S-Reg, D-Reg	Description	Operation
ADC(.B)†	dst	Add C to destination	dst + C → dst
ADD(.B)	src,dst	Add source to destination	src + dst → dst
ADDC(.B)	src,dst	Add source and C to destination	src + dst + C → dst
AND(.B)	src,dst	AND source and destination	src .and. dst → dst
BIC(.B)	src,dst	Clear bits in destination	.not.src .and. dst → dst
BIS(.B)	src,dst	Set bits in destination	src .or. dst → dst
BIT(.B)	src,dst	Test bits in destination	src .and. dst
BR†	dst	Branch to destination	dst → PC
CALL	dst	Call destination	PC+2 → stack, dst → PC
CLR(.B)†	dst	Clear destination	0 → dst
CLRC†	src,dst	Clear C	0 → C
CLRN†	dst	Clear N	0 → N
CLRZ†	src,dst	Clear Z	0 → Z
CMP(.B)	dst	Compare source and destination	dst - src
DADC(.B)†	dst	Add C decimally to destination	dst + C → dst (decimally)
DADD(.B)	dst	Add source and C decimally to dst.	src + dst + C → dst (decimally)
DEC(.B)†	dst	Decrement destination	dst - 1 → dst
DECD(.B)†	dst	Double-decrement destination	dst - 2 → dst
DINT†	label	Disable interrupts	0 → GIE
EINT†	label	Enable interrupts	1 → GIE
INC(.B)†	label	Increment destination	dst + 1 → dst
INCD(.B)†	label	Double-increment destination	dst+2 → dst
INV(.B)†	label	Invert destination	.not.dst → dst
JC/JHS	label	Jump if C set/Jump if higher or same	
JEQ/JZ	label	Jump if equal/Jump if Z set	
JGE	label	Jump if greater or equal	
JL	src,dst	Jump if less	
JMP	dst	Jump	PC + 2 x offset → PC
JN	src	Jump if N set	
JNC/JLO	dst	Jump if C not set/Jump if lower	
JNE/JNZ	dst	Jump if not equal/Jump if Z not set	
MOV(.B)	dst	Move source to destination	src → dst
NOP†	dst	No operation	
POP(.B)†	dst	Pop item from stack to destination	@SP → dst, SP+2 → SP
PUSH(.B)	src,dst	Push source onto stack	SP - 2 → SP, src → @SP
RET†	src,dst	Return from subroutine	@SP → PC, SP + 2 → SP
RETI	dst	Return from interrupt	
RLA(.B)†	dst	Rotate left arithmetically	
RLC(.B)†	dst	Rotate left through C	
RRA(.B)	src,dst	Rotate right arithmetically	
RRC(.B)	dst	Rotate right through C	
SBC(.B)†	src,dst	Subtract not(C) from destination	@SP → dst, SP+2 → SP
SETC†	src,dst	Set C	SP - 2 → SP, src → @SP
SETN†	src,dst	Set N	@SP → PC, SP + 2 → SP
SETZ†	src,dst	Set Z	@SP → dst, SP+2 → SP
SUB(.B)	src,dst	Subtract source from destination	SP - 2 → SP, src → @SP
SUBC(.B)	src,dst	Subtract source and not(C) from dst.	@SP → PC, SP + 2 → SP
SWPB	dst	Swap bytes	
SXT	Dst	Extend sign	
TST(.B)†	Dst	Test destination	dst + 0FFFFh + 1
XOR(.B)	src,dst	Exclusive OR source and destination	src .xor. dst → dst

PUERTOS DE ENTRADA SALIDA

El microcontrolador MSP430G2 XXX tiene implementados puertos de entrada / salida (I/O) digital. Cada terminal I/O se puede configurar individualmente como entrada o salida digital, y se puede escribir en cada línea I/O o leer de ella individualmente. En el caso de los puertos 1 y 2, existe capacidad de interrupción en cada una de las terminales, que pueden ser configuradas para responder en flanco de subida o flanco de bajada.

Funcionamiento básico de las entradas/salidas digitales

Las entradas/salidas digitales se configuran con el software del usuario. Cada registro de puerto es un registro de 8-bits y se accede con instrucciones de byte.

Existen 3 registros básicos de control para cada puerto, y pueden existir registros adicionales para controlar otras funciones integradas, como las interrupciones, resistencias de pull y entradas sensibles a capacitancia variable (función Touch)

Funciones básicas de los Puertos

PxIN Puerto de entrada.

Cada bit en cada registro PxIN refleja el valor de la señal de entrada en la terminal correspondiente I/O cuando se configura como tal.

Bit = 0, la entrada es baja (0)

Bit = 1, la entrada es alta (1)

PxOUT Puerto de salida

Cada bit en cada registro PxOUT refleja el valor de la señal en la terminal correspondiente I/O cuando se configura como salida.

Bit = 0, la salida es baja (0)

Bit = 1, la salida es alta (1)

PxDIR

Configuración de la dirección del puerto. La configuración es bit a bit.

Bit= 0 . La terminal se configura como entrada

Bit= 1 . La terminal se configura como salida

PxREN registro de habilitación de resistencias de pull

Cada bit en cada registro PxREN habilita o deshabilita la resistencia integrada a la entrada.

Bit = 0: Resistencia de pullup/pulldown deshabilitada.

Bit = 1: Resistencia de pullup/pulldown habilitada.

Cuando PxDIR configura a la Terminal como entrada y se habilita PxREN, PxOUT configura el modo

Bit = 0: La resistencia es de pull down

Bit = 1: La resistencia es de pull down

Cuando la dirección del puerto es de salida, no se deberían habilitar las resistencias.