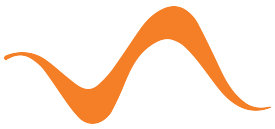
UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA



Advanced Center
for Electrical and Electronic Engineering

Proyecto de Memoria Hito 1

Manual de uso y Primeros pasos con MCU TMS320F28377D Control Card

Presentado por	Matias A. Licanqueo Mansilla
Profesor Guía	Marcelo A. Pérez
Correferente	Alan Wilson
Fecha	17 de junio de 2020
Contacto	matias.licanqueo.14@sansano.usm.cl
Versión	1.0.0

Índice

1. Acerca del MCU **1**

2. Primeros Pasos **2**

3. Periféricos **3**
3.1. ADC . . . 3
3.1.1. Introducción . . . 3
3.1.2. Resultados de Conversión . . . 4
3.1.3. Principio de operación del SOC . . . 4
3.1.3.1. Fuente del trigger . . . 5
3.1.3.2. Canal a convertir . . . 5
3.1.3.3. Ventana de adquisición . . . 5
3.1.3.4. Ejemplo configuración SOC . . . 5
3.1.4. Entradas del ADC . . . 5
3.1.4.1. Modo Single-Ended . . . 5
3.1.4.2. Modo Diferencial . . . 5
3.1.5. Prioridad de conversión del ADC . . . 6
3.1.5.1. Round Robin . . . 6
3.1.5.2. Modo de Alta Prioridad . . . 8
3.1.6. Principios de operación del EOC . . . 8
3.2. ePWM . . . 8
3.2.1. Introducción . . . 8
3.2.2. Principio de operación de la PWM . . . 9
3.2.3. Frecuencia de la PWM . . . 10
3.2.4. Comparador . . . 11
3.2.5. Eventos e Interrupciones . . . 12
3.2.6. Dead-Band generator . . . 13
3.2.6.1. Cálculo de RED y FED . . . 14

Referencias **15**

1. Acerca del MCU

A grandes rasgos, un MCU es un circuito integrado programable, capaz de ejecutar instrucciones grabadas en su memoria, y sus principales unidades son: CPU, memoria y periféricos.
En nuestro caso, el MCU utilizado es un Texas Instruments Familia C2000 serie Delfino modelo TMS320F28377D, el cual tiene doble CPU de 200Mhz, memoria flash de 1MB, 2 CLA, 4 módulos ADC, de 12-16 bits y hasta 12-24 canales, 24 canales PWM, entre otros.
Para mayor información revisar los siguientes documentos:

- Sitio web C2000 MCU: https://www.ti.com/microcontrollers/c2000-real-time-control-mcus/overview.html
- TMS320F2837xD Datasheet: https://www.ti.com/lit/ds/symlink/tms320f28377d.pdf

- Technical Reference Manual https://www.ti.com/lit/ug/spruhm8i/spruhm8i.pdf
- TMS320F28377D control CARD R1.1 Information Guide: Revisar C2000ware o Control Suite.
- Card Pinout: F28377D (180pin)
- DIM100 Pinout DIM PINOUT With adapter card
- TMS320C28x CPU and Instruction Set: http://www.ti.com/lit/ug/spru430f/spru430f.pdf
- TMS320C28x Floating Point Unit and Instruction Set: http://www.ti.com/lit/ug/sprueo2b/sprueo2b.pdf

2. Primeros Pasos

Antes de comenzar a incursionar en el mundo de los microcontroladores de Texas Instruments, primero se deben descargar ciertos recursos (enlaces en negrita):

- **Code Composer Studio** Es el entorno de desarrollo (IDE) a utilizar para desarrollar aplicaciones relacionadas con los microcontroladores de TI. Está disponible para Windows, Linux, MacOS, de 64b, versiones de 32b están obsoletas.
- **C2000WARE** Es un set de software y documentación, desarrollado por TI para ayudar a programar sus microcontroladores. Incluye Librerías, drivers, ejemplos de periféricos, datasheet, pinout diagram, etc.
- **controlSUITE**Es el software predecesor de C2000WARE, Contiene librerías, ejemplos, diagramas, datasheet, etc, con el objetivo de minimizar el tiempo de programación de los MCU de TI. Se recomienda descargar esto (a pesar de que su última actualización fue en 2018), ya que algunos ejemplos que contiene se programan de forma distinta que en C2000WARE, lo que puede enriquecer el estudio.

Luego, existen muchas formas de continuar, lo que dependerá del grado de conocimiento del usuario.La forma que recomienda TI es seguir su **C2000™ F2837xD Microcontroller Workshop**. Es un curso técnico con una serie de laboratorios en los que se explican a grandes rasgos el funcionamiento tanto de la arquitectura del MCU, el entorno de programación, inicialización de periféricos, etc.

Otra forma de iniciar es comenzar leyendo los ejemplos (ya que son plug&play, solo se debe conectar el microcontrolador y funcionan) que se encuentran en C2000WARE y controlSUITE (los cuales se deben cargar en el CCS) y entender su funcionamiento, apoyando la lectura con el Technical Reference Manual (TMR de aquí en adelante). El ejemplo más sencillo es "$led\_ex1\_blinky.c$", en el que se definen la salida de uno de los pines GPIO como el led y se enciende y apaga la salida cada cierto tiempo en un loop infinito.

3. Periféricos

Luego del acercamiento inicial, se debe comenzar a utilizar los periféricos del MCU para programar el código. En este documento no se señalan todos los detalles, sino si funcionamiento a grandes rasgos. Para mayor información, revisar el TRM.

3.1. ADC

3.1.1. Introducción

Por sus siglas, ADC significa Analog Digital Converter, o sea, un dispositivo que traduce los valores de una señal analógica en una número binario que puede interpretar el sistema. El ADC del MCU a utilizar es de tipo SAR (successive approximation), con una resolución seleccionable 12 o 16 bits. Dicho ADC está compuesto por 2 partes, el "*core*" y el "*wrapper*". El "*core*" está formado por los circuitos analógicos, como el multiplexor selector de canales, el circuito Sample-and-hold (S/H), etc. Por otro lado, el "*Wrapper*" está compuesto por los circuitos digitales que controlan el ADC.

El sistema de conversiones cuenta con 4 módulos ADC (A, B, C, D), y cada módulo cuenta con sus propios registros de selección de resolución, modos de entrada de señal (Single-ended y Differential), SOC, trigger sources, etc.

En la figura 1 se observa el diagrama de bloques del módulo ADC.

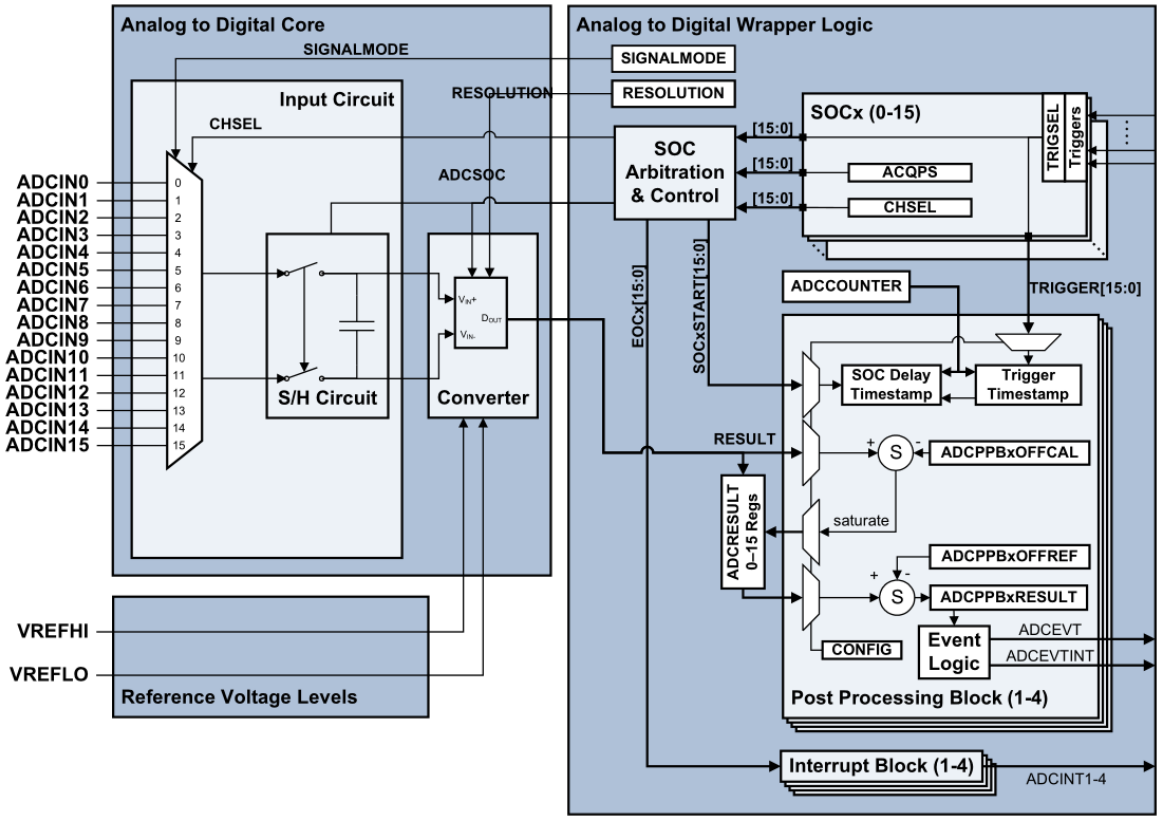


Figura 1: Diagrama de bloques del módulo ADC, reference [1, page 1555].

3.1.2. Resultados de Conversión

Para interpretar los resultados obtenidos en la conversión del ADC, se debe hacer uso de la tabla de la figura 2 la cual nos entregan la ecuación de conversión de cuentas del adc, tanto para 12 bits como para 16 bits. También observamos la tabla de resultados de conversión desde el valor digital hacia su correspondiente análogo, en la figura 3.

Table 11-2. Analog to 12-bit Digital Formulas

	Analog Input	Digital Result
Single-Ended	when ADCINy ≤ VREFLO	ADCRESULTx = 0
	when VREFLO < ADCINy < VREFHI	$ADCRESULTx = 4096\left(\frac{ADCINy - VREFLO}{VREFHI - VREFLO}\right)$
	when ADCINy ≥ VREFHI	ADCRESULTx = 4095
Differential	Invalid Mode	Invalid Mode

Table 11-3. Analog to 16-bit Digital Formulas

	Analog Input	Digital Result
Single-Ended	Invalid Mode	Invalid Mode
Differential	when ADCINyP - ADCINyN ≤ -VREFHI	ADCRESULTx = 0
	when -VREFHI < ADCINyP - ADCINyN ≤ VREFHI	$ADCRESULTx = 65536\left(\frac{ADCINyP - ADCINyN + VREFHI}{2\ VREFHI}\right)$
	when ADCINyP - ADCINyN ≥ VREFHI	ADCRESULTx = 65535

Figura 2: Resultados de conversión Análogo a digital, reference [1, page 1557].

Table 11-4. 12-Bit Digital-to-Analog Formulas

	Digital Value	Analog Equivalent
Single-Ended	when ADCRESULTy = 0	ADCINx ≤ VREFLO
	when 0 < ADCRESULTy < 4095	ADCINx = (VREFHI - VREFLO) $\left(\frac{ADCRESULTy}{4096}\right) + VREFLO$
	when ADCRESULTy = 4095	ADCINx ≥ VREFHI
Differential	Invalid Mode	Invalid Mode

Table 11-5. 16-Bit Digital-to-Analog Formulas

	Digital Value	Analog Equivalent
Single-Ended	Invalid Mode	Invalid Mode
Differential	when ADCRESULTy = 0	ADCINxP - ADCINxN ≤ -VREFHI
	when 0 < ADCRESULTy < 65535	$ADCINxP - ADCINxN = VREFHI\left(\frac{2\ ADCRESULTy}{65536} - 1\right)$
	when ADCRESULTy = 65535	ADCINxP - ADCINxN ≥ VREFHI

Figura 3: Resultados de conversión Digital a Análogo, reference [1, page 1557].

3.1.3. Principio de operación del SOC

El disparo del inicio de la secuencia de conversión del ADC está comandado por el SOC (Start-of-Conversion). Cada SOC es un conjunto de configuraciones que definen la conversión única de un solo canal, donde se configura: **Fuente del trigger** que inicia la conversión, el

Canal a convertir y la **Ventana de adquisición**. Multiples SOC pueden ser configurados para el mismo Trigger, lo que generará una secuencia de conversiones.

Cada SOC se configura desde su propio registro ADCSOCxCTL (donde la x es un numero entre 0 y 15), y donde cada modulo adc tiene sus propios registros SOC independientes.

3.1.3.1. Fuente del trigger Se configura el registro ADCSOCxCTL.TRIGSEL, definiendo la fuente que dará inicio a la conversión, los que pueden ser, timers del cpu, GPIO o ePWM.

3.1.3.2. Canal a convertir Se configura el registro ADCSOCxCTL.CHSEL. Selecciona qué ADC será convertido cuando llegue la señal de trigger.

3.1.3.3. Ventana de adquisición Se configura el registro ADCSOCxCTL.ACQPS. Corresponde al tiempo que tendrá permitido el circuito S/H para convertir el valor análogo en una palabra digital. Para calcular ACQPS utilizamos la siguiente fórmula:
Ventana de adquisición = $(ACQPS + 1) \cdot$ (System Clock (SYSCLK) cycle time)

3.1.3.4. Ejemplo configuración SOC

En este ejemplo se configura el ADC-A con una conversión simple en el canal ADC-A1 cuando el contador del ePWM3 alcance su periodo. Para esto, asumimos que el ePWM3 está configurado para generar la señal SOCA o SOCB (referido al SOC del PWM, no al SOC del ADC). El SOC5 es elegido de forma arbitraria, cualquiera de los otros 15 SOC puede ser utilizado. Si queremos una ventana de adquisición de 100ns, con SYSCLK de 200Mhz, la ventana de adquisición será 100ns/(1/200Mhz) = 20 ciclos SYSCLK, por lo que ACQPS = 20 - 1 = 19.

```
AdcaRegs.ADCSOC5CTL.bit.CHSEL = 1; //SOC5 will convert ADCINA1
AdcaRegs.ADCSOC5CTL.bit.ACQPS = 19; //SOC5 will use sample duration of 20 SYSCLK cycles
AdcaRegs.ADCSOC5CTL.bit.TRIGSEL = 10; //SOC5 will begin conversion on ePWM3 SOCB
```

Cuando el ePWM3 alcance su periodo y se genere la señal SOCB, el ADC comenzará a muestrear el canal ADC-A1 inmediatamente si el circuito S/H está desocupado. Si está ocupado, comenzará a muestrear dicho adc cuando el SOC5 gane prioridad. El resultado se guarda en el registro AdcaRegs.ADCRESULT5.

3.1.4. Entradas del ADC

Los modos de entradas del adc son 2, el modo Single-Ended, y el modo diferencial, los cuales se detallan a continuación:

3.1.4.1. Modo Single-Ended En este modo solo se utiliza 1 pin para la señal de entrada, pues está referenciado con tierra. Para este tipo de conversión se tiene 12 bits de resolución. Según el datasheet, el valor máximo de voltaje de entrada es 2.5 o 3V.

3.1.4.2. Modo Diferencial En este modo, se utilizan 2 pines para leer la diferencia de voltaje entre ellos. Para este tipo de conversión se tiene 16 bits de resolución.

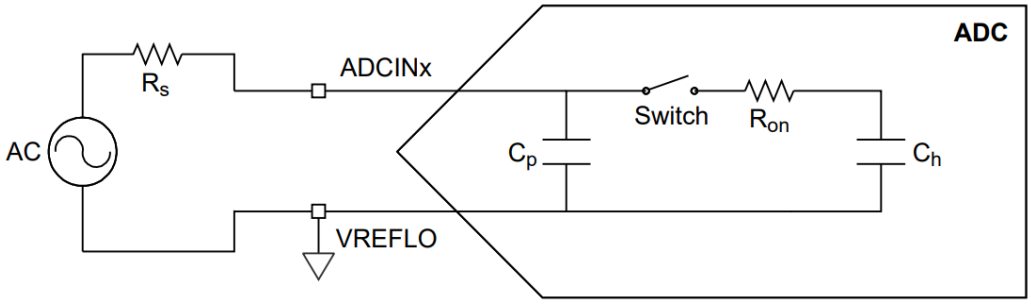


Figura 4: Single-Ended Input Model, reference [1, page 1560].

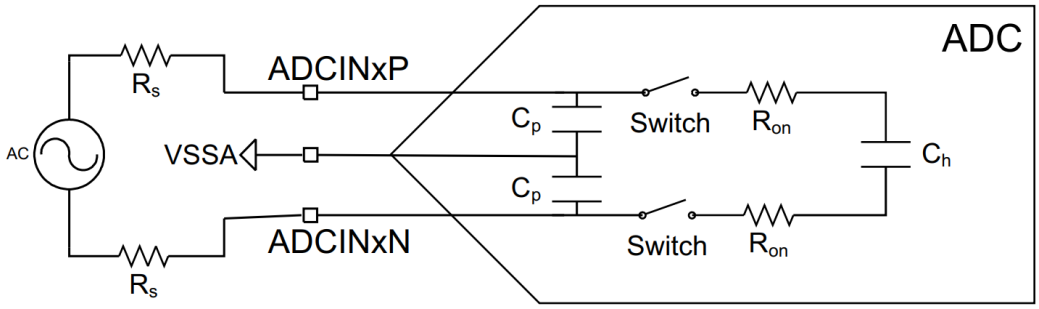


Figura 5: Differential Input Model, reference [1, page 1560].

3.1.5. Prioridad de conversión del ADC

En cada módulo del ADC, cuando muchas banderas de SOC se activan al mismo tiempo (pues solo se puede convertir 1 entrada cada vez, ya que solo se tiene un circuito s/h por cada módulo), se debe elegir qué SOC se debe convertir primero. Existen 2 formas para determinar el orden de conversión:

3.1.5.1. Round Robin

Es el método por defecto, consiste en que ningún SOC tiene una prioridad sobre otros, sino que la conversión de los ADC se realizará en función de un puntero llamado round robin pointer (RRPOINTER). Dicho puntero recorre de forma horaria los SOC desde el SOC0 hasta el SOC15. El RRPOINTER siempre apunta hacia el SOC siguiente al cual se ha realizado la conversión, por ejemplo, si la última conversión fue realizada por el SOC7, el RRPOINTER apuntará al SOC8, el cual ahora tendrá la mayor prioridad. Si se apunta hacia el SOC8 y simultáneamente llegan 2 señales de conversión de SOC6 y SOC9, debido a que el puntero recorre los SOC desde el 0 al 15, primero convertirá el SOC9 y luego el SOC6, y el puntero quedará con la prioridad de conversión apuntando hacia el SOC7. Un ejemplo se muestra en la figura 6 (ver TRM para más información).

A After reset, SOC0 is highest priority SOC ;
SOC7 receives trigger ;
SOC7 configured channel is converted immediately .

B RRPOINTER changes to point to SOC 7;
SOC8 is now highest priority SOC .

C SOC2 & SOC12 triggers rcvd . simultaneously ;
SOC12 is first on round robin wheel ;
SOC12 configured channel is converted while SOC2 stays pending .

D RRPOINTER changes to point to SOC 12;
SOC2 configured channel is now converted .

E RRPOINTER changes to point to SOC 2;
SOC3 is now highest priority SOC .

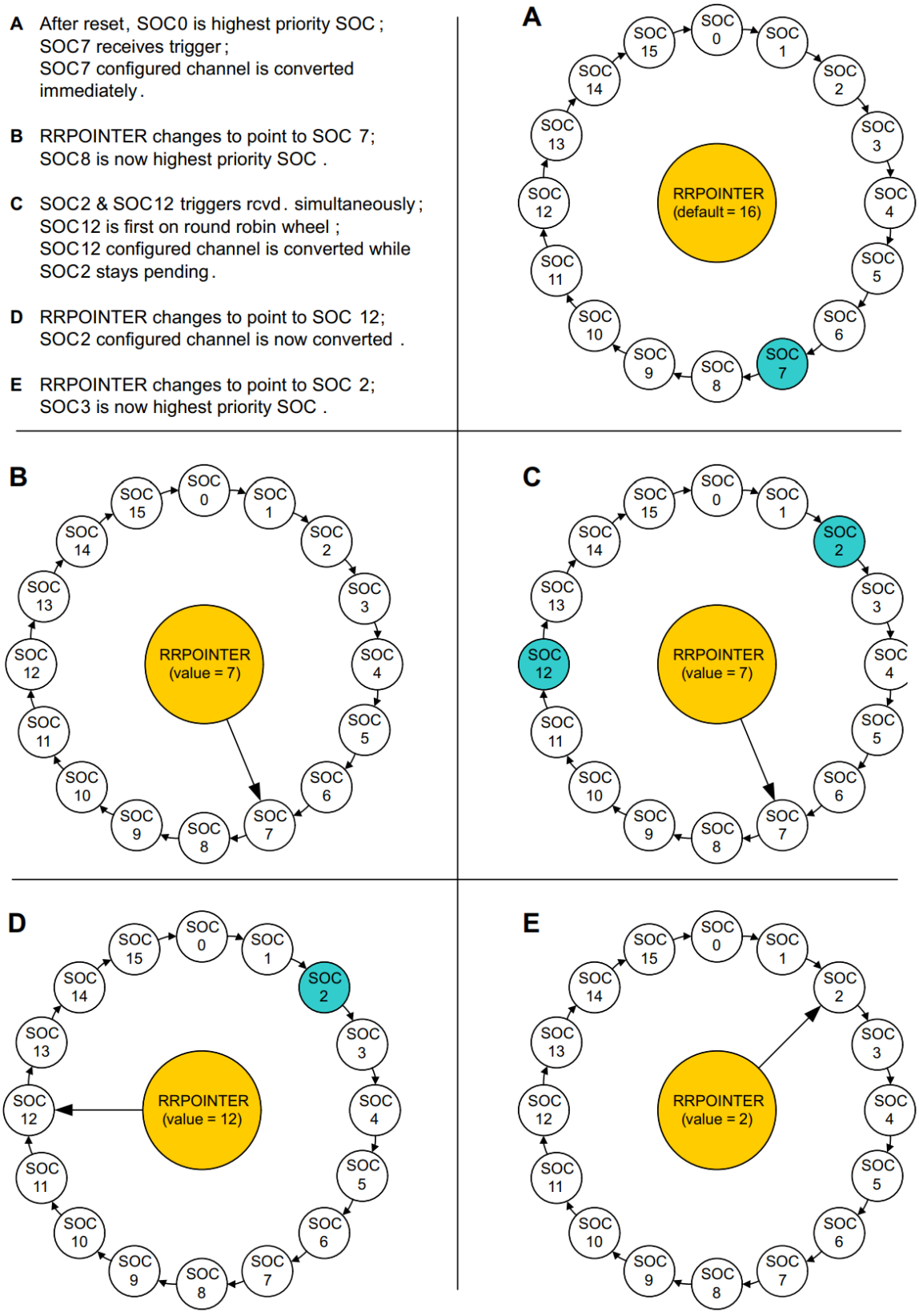


Figura 6: Round Robin Priority Example, from [1, page 1565].

3.1.5.2. Modo de Alta Prioridad

En este modo, también existen el RRPOINTER, solo que se definen SOC prioritarios, los cuales se convertirán primero, independiente del puntero.

3.1.6. Principios de operación del EOC

Cada SOC tiene una correspondiente señal EOC (end-of-conversion). Dicha señal puede ser utilizada para disparar una interrupción en el ADC. El ADC puede ser configurado para generar el pulso EOC tanto al final de la conversión de voltaje o al final de la ventana de adquisición, a través del registro ADCCTL1.INTPULSEPOS. Cada módulo ADC tiene 4 interrupciones configurables, las cuales pueden ser disparadas por cualquiera de las 16 señales EOC.

Esta señal es útil pues nos permite generar interrupciones una vez que tenemos el resultado de la conversión, y así poder utilizar dicho valor para realizar alguna operación o hacer trigger a algun periférico o utilizar el procesador matemático CLA.

3.2. ePWM

3.2.1. Introducción

Enhanced Pulse Width Modulator (ePWM) es un periférico clave en los sistemas de potencia, pues se utiliza en los sistema de control, necesarios para cualquier circuito electrónico que requiera conversión de energía.

El módulo ePWM representa un canal PWM compuesto de 2 salidas PWM: EPWMxA y EPWMxB. Los módulos ePWM están encadenados por sincronización del clock, lo que permite que operen como un solo sistema cuando es requerido. Cada módulo ePWM tiene las siguientes características:

- Contador de 16 bits con control de periodo (frecuencia).
- Dos salidas PWM (EPWMxA y EPWMxB).
- Generación de Dead-Band con control independiente de retardo de canto de subida y bajada.
- Los eventos pueden disparar las interrupciones del CPU y el inicio de conversión del ADC.
- Las salidas PWM están mapeadas a sus respectivos pines GPIO.
- Cada modulo tiene 2 señales de inicio de conversión para el adc (EPWMxSOCA and EPWMxSOCB).

En la figura 7 se observa el diagrama de bloques del módulo PWM.

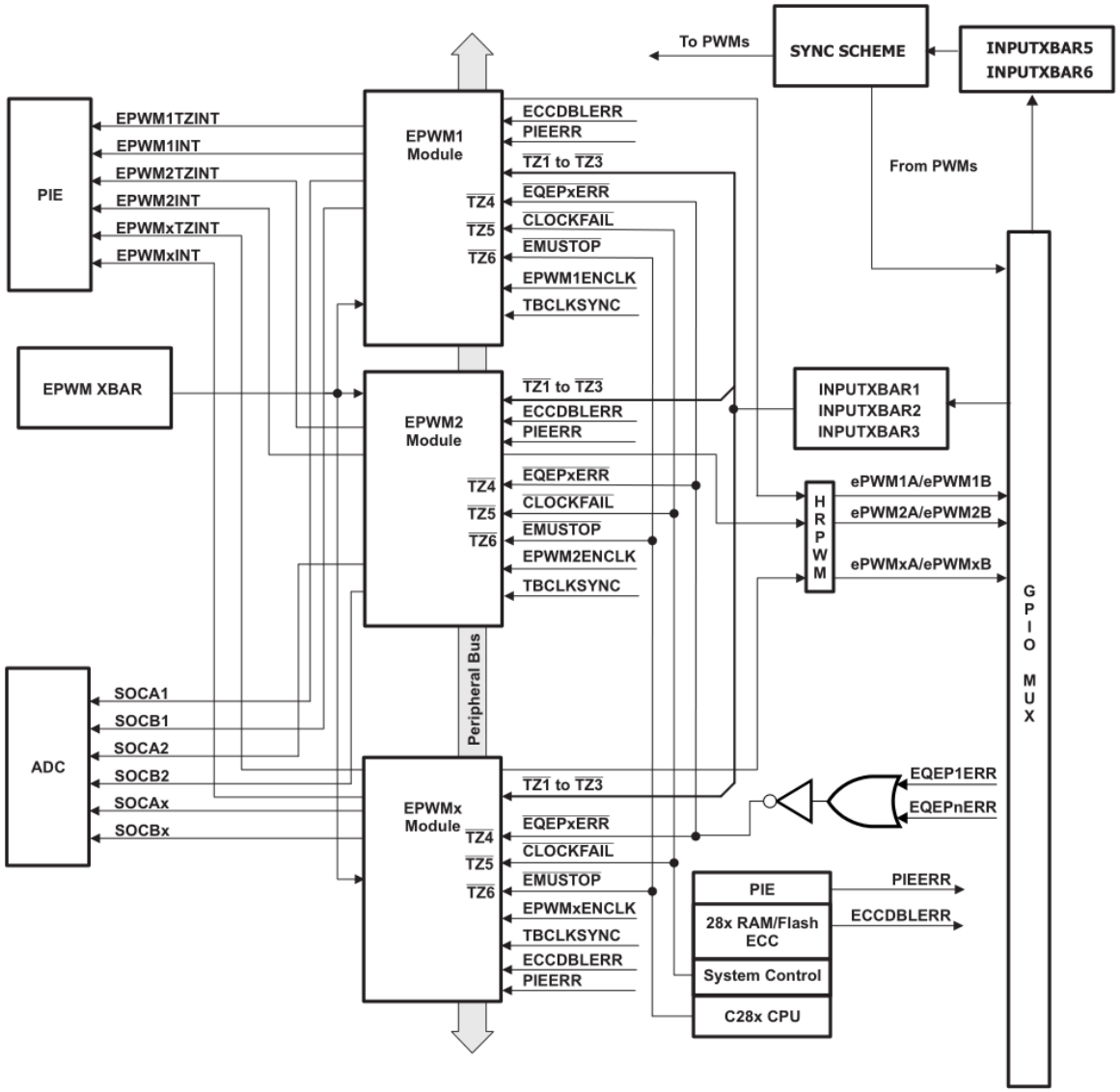


Figura 7: Diagrama de bloques del módulo PWM, from [1, page 1865].

3.2.2. Principio de operación de la PWM

A grandes rasgos, la PWM cuenta con un contador llamado Time-Base-Counter (TBCTR), el que cuenta hasta un cierto valor llamado Time-Base-Period-Register (TBPRD) y el tiempo que tarda en llegar desde 0 hasta TBPRD define el periodo de la PWM. La velocidad que el TBCTR cuenta desde un valor anterior al siguiente depende de la frecuencia del módulo ePWM (EPWMCLK). El Modo de cuenta de la PWM se observa en la figura 8. Existen 4 modos de cuenta, los que se configuran con el registro TBCTL.bit.CTRMODE, en el modo de cuenta de subida, TBCTR va desde 0 hasta TBPRD, en el modo bajada, TBCTR va desde TBPRD hasta llegar a 0, y en el modo de subida-bajada cuenta desde 0 hasta TBPRD y luego desciende hasta 0 en el mismo periodo de tiempo, en el modo de cuenta Freeze, el contador se queda detenido y no realiza acción alguna (útil cuando se quiere dejar apagado el PWM).

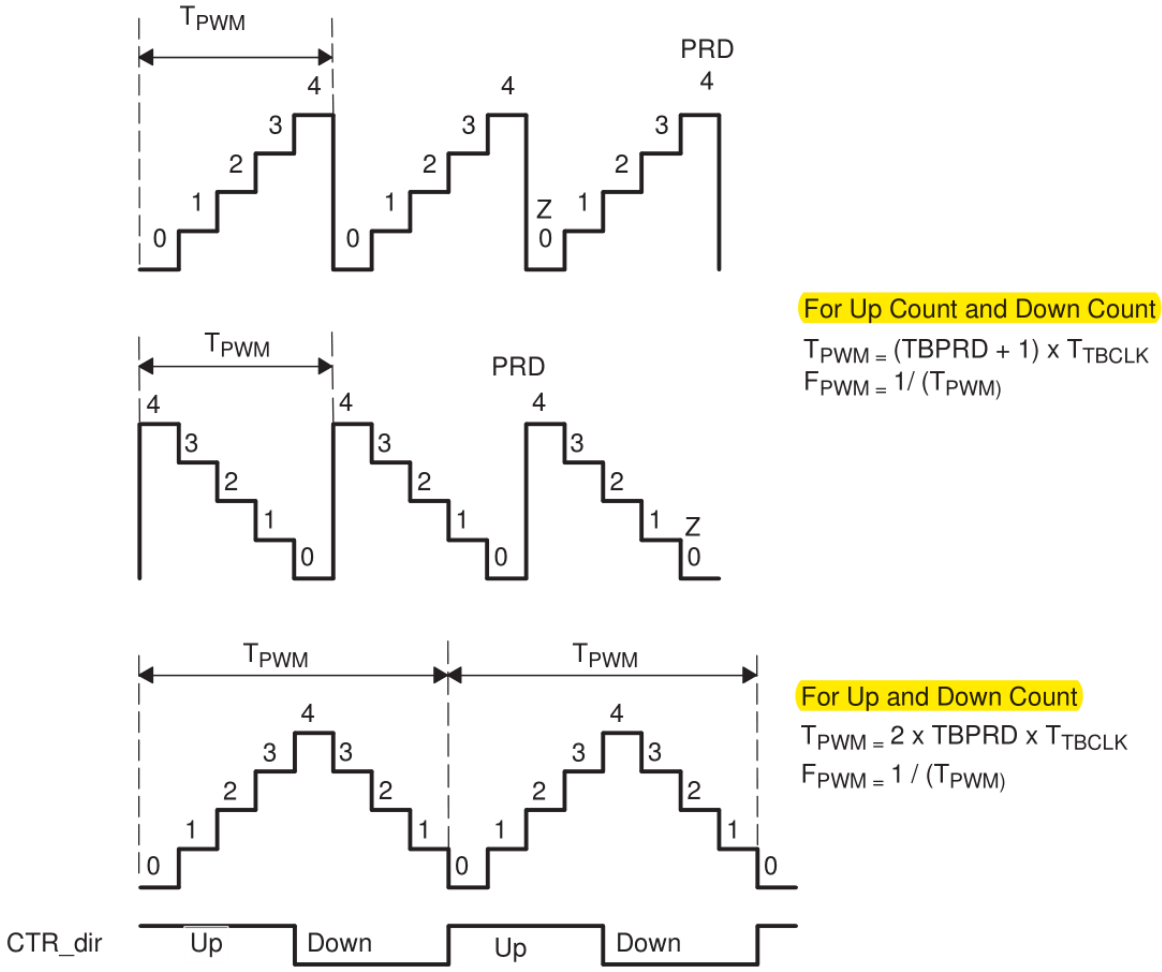


Figura 8: Modo de cuenta: Subida, Bajada, subida-bajada, from [1, page 1874].

3.2.3. Frecuencia de la PWM

La frecuencia de la PWM generada depende del modo de cuenta que se esté utilizando (registro TBCTL.bit.CTRMODE):

Modo subida o bajada

$$T_{PWM} = (TBPRD + 1) \cdot T_{TBCLK}$$

Modo subida-bajada

$$T_{PWM} = 2 \cdot TBPRD \cdot T_{TBCLK}$$

Para ser más precisos, la ecuación queda como:

$$F_{PWM} = \frac{F_{SYSCLKOUT}}{4 \cdot TBPRD \cdot HSPCLKDIV \cdot CLKDIV}$$

donde:

- $F_{SYSCLKOUT}$ es la frecuencia del sistema (200MHz en caso de no haber cambiado nada).

- TBPRD es el registro que se configura para definir la frecuencia de la PWM.
- HSPCLKDIV Registro divisor del clock de la PWM.
- CLKDIV Registro divisor del clock de la PWM.

3.2.4. Comparador

El módulo ePWM tiene un submódulo Counter-Compare. Este valor se compara periódicamente con los registros counter-compare A (CMPA), counter-compare B (CMPB), counter-compare C (CMPC) y counter-compare D (CMPD). Es posible generar eventos cuando el contador base se iguala a algún registro de comparación (por ejemplo, TBCTR = CMPA). En los sistemas de control, el ciclo de trabajo se configura utilizando CMPA y CMPB.

En la figura 9 se observa los valores de comparación de CMPA y CMPB y los eventos que se generan cuando TBCTR = CMPA y TBCTR = CMPB.

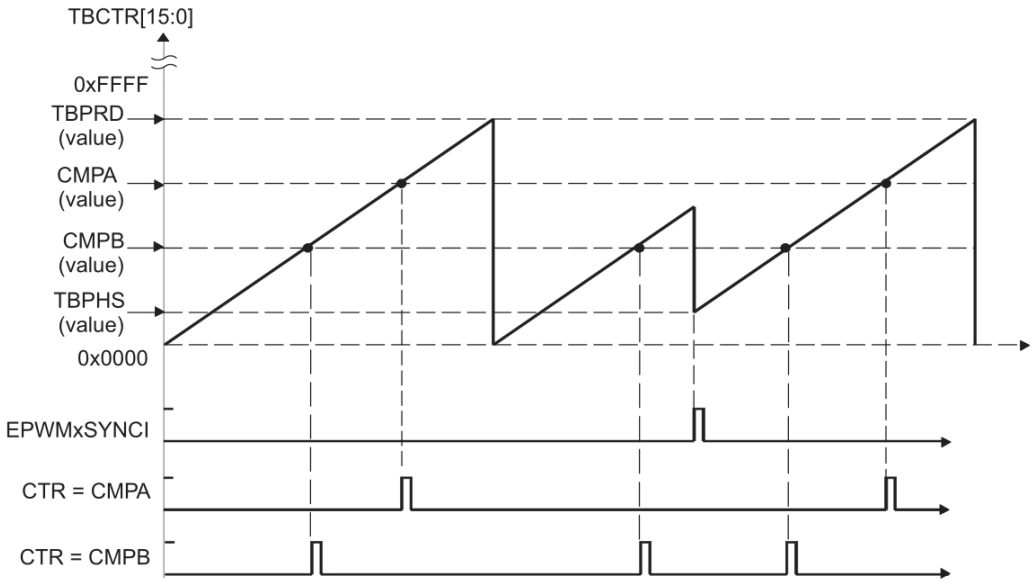


Figura 9: Counter-Compare Event Waveforms in Up-Count Mode, from [1, page 1886].

Si definimos la salida de la PWM en 1 cuando TBCTR = CMPA en subida (CAU = SET) y 0 cuando TBCTR = CMPA en bajada (CAD = CLEAR), para el modo de cuenta subida-bajada, observamos como cambia el ciclo de trabajo en la figura 10 definiendo el valor de CMPA.

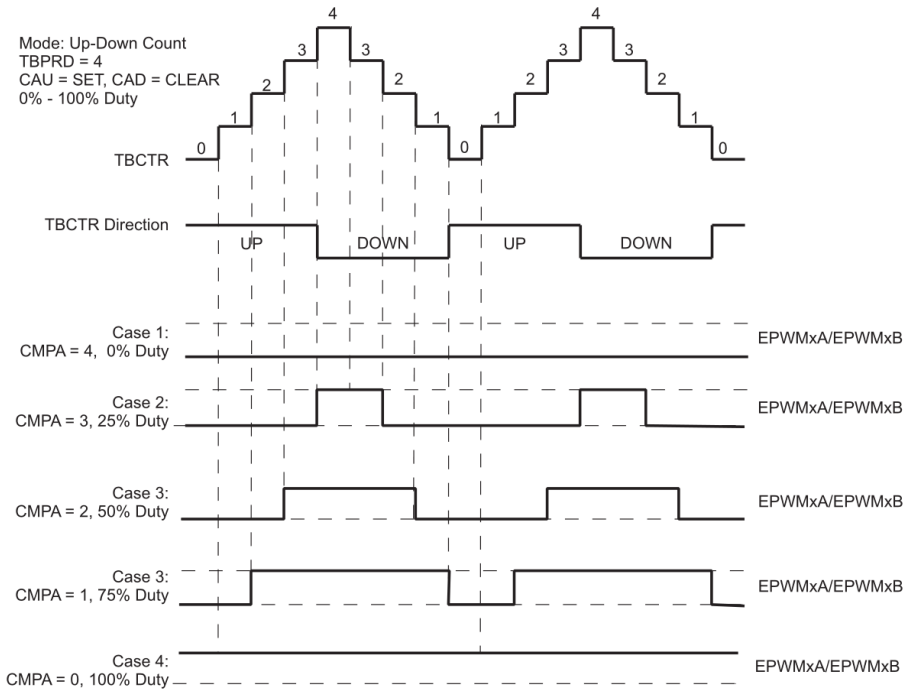


Figura 10: Up-Down-Count Mode Symmetrical Waveform, from [1, page 1896].

3.2.5. Eventos e Interrupciones

Nuestro MCU posee 12 módulos ePWM en total, los que pueden hacer trigger hasta 24 interrupciones y cada módulo puede hacer trigger a uno o varios SOC para activar la conversión de los ADC de forma periódica.

Con el registro ETSEL (Event Trigger Selection Register) podemos configurar los bits correspondientes a la generación de eventos, como:

- **INTSEL** ePWM Interrupt Selection Options, nos permite definir cuándo se genera un evento (por ejemplo, se pueden definir eventos cuando TBCTR sea igual 0, a TBPRD, CMPA, CMPB, etc).
- **INTEN** Enable ePWM Interrupt Generation, permite activar la generación de interrupciones.
- **SOCASEL** ePWMxSOCA Selection Options, determina cuándo el pulso SOC será generado (por ejemplo, cuando TBCTR sea igual a 0, a TBPRD, CMPA, CMPB, etc).
- **SOCAEN** Enable the ADC Start of Conversion A, permite generar el pulso de activación del SOC A.

El registro ETPS (Event Trigger Pre-Scale Register) nos permite configurar el número de eventos para generar interrupciones, SOC, etc:

- **INTPRD** ePWM Interrupt Period Select, determina cuántos eventos ETSEL[INTSEL] tienen que ocurrir antes de que se genere la interrupción.

- **SOCAPRD** ePWM ADC Start-of-Conversion A Event (EPWMxSOCA) Period Select, determina cuántos eventos ETSEL[SOCASEL] deben ocurrir antes que el pulso EPWMxSOCA sea generado.

3.2.6. Dead-Band generator

El módulo ePWM también tiene un submódulo DB-Generator, el cual nos permite retardar el encendido o apagado de las salidas del ePWM. Las principales funciones del submódulo son:

- Generar pares de señales (EPWMxA y EPWMxB) con una relación de banda muerta a partir de solo una entrada
- Programar los pares para:
  - Active high (AH)
  - Active low (AL)
  - Active high complementary (AHC)
  - Active low complementary (ALC)
- Agregar retardo a los cantos de subida (RES)
- Agregar retardo a los cantos de bajada (FED)

La gráfica que resume el funcionamiento del módulo dead-band generator se observa en la figura 11

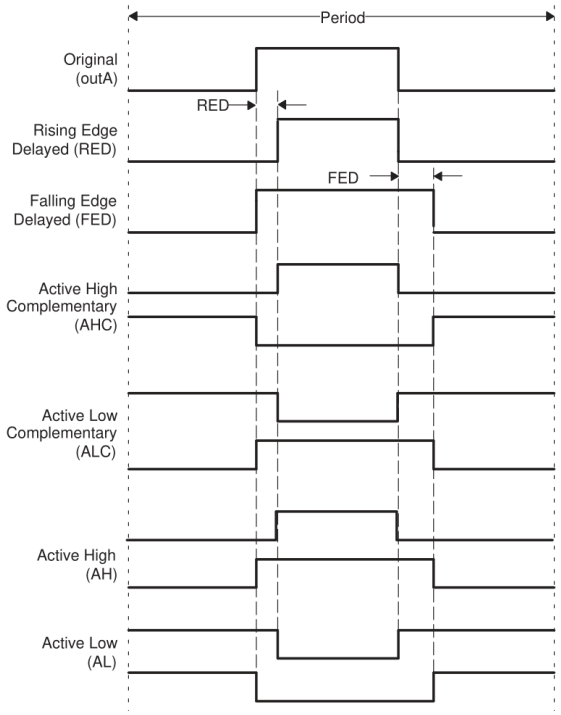


Figura 11: Dead-Band Waveforms for Typical Cases (0 % <Duty <100 %), from [1, page 1906].

Los registros a configurar son: DBCTL (Dead-Band Generator Control Register)

- **OUT_MODE** Dead-Band Output Mode Control, con estos bits se activa el FED, RED o ambos.
- **POLSEL** Polarity Select Control, permite seleccionar entre modos AHC, ALC, AH y AL.
- **IN_MODE** Dead-Band Input Mode Control, permite seleccionar la fuente (puede ser ePWMxA o ePWMxB) de los retardos tanto para el FED como el RED.

DBRED Dead-Band Generator Rising Edge Delay register, define el valor del canto de bajada.
DBFED Dead-Band Generator Falling Edge Delay register, define el valor del canto de subida.

3.2.6.1. Cálculo de RED y FED

Para encontrar estos valores se utiliza la siguiente ecuación:

$$FED = DBFED \cdot T_{TBCLK}/2$$

Recordando que:

$$T_{TBCLK} = 1/F_{TBCLK} = \frac{HSPCLKDIV \cdot CLKDIV}{F_{EPWMCLK}} = \frac{2 \cdot HSPCLKDIV \cdot CLKDIV}{F_{SYSCLKOUT}}$$

Luego:

$$DBFED = \frac{FED \cdot F_{SYSCLKOUT}}{HSPCLKDIV \cdot CLKDIV}$$

Por ejemplo, si queremos un Dead Band de 500[ns], y tenemos que el ciclo del reloj es de 200[Mhz], HSPCLKDIV = 2 y CLKDIV = 1, nuestro registro DBFED a configurar debe ser igual a 50.
RED se calcula de la misma forma:

$$RED = DBRED \cdot T_{TBCLK}/2$$

Referencias

[1] Texas Instruments. *TMS320F2837xD Dual-Core Microcontrollers, Technical Reference Manual*. Texas Instruments, December, 2013.