

Módulo de comunicaciones serie síncrona: MSSP

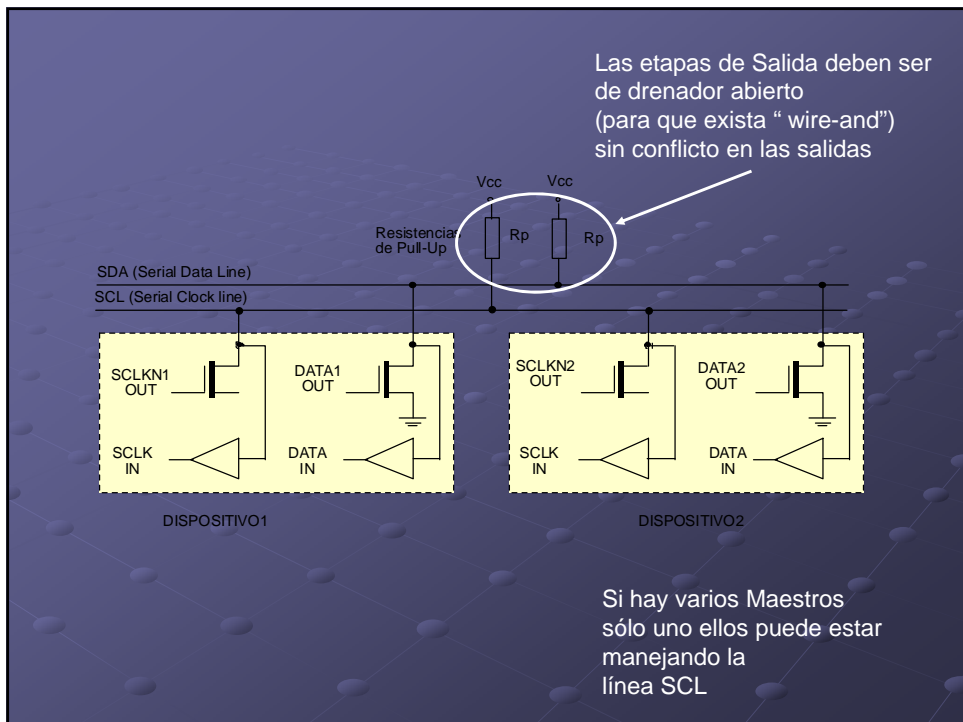
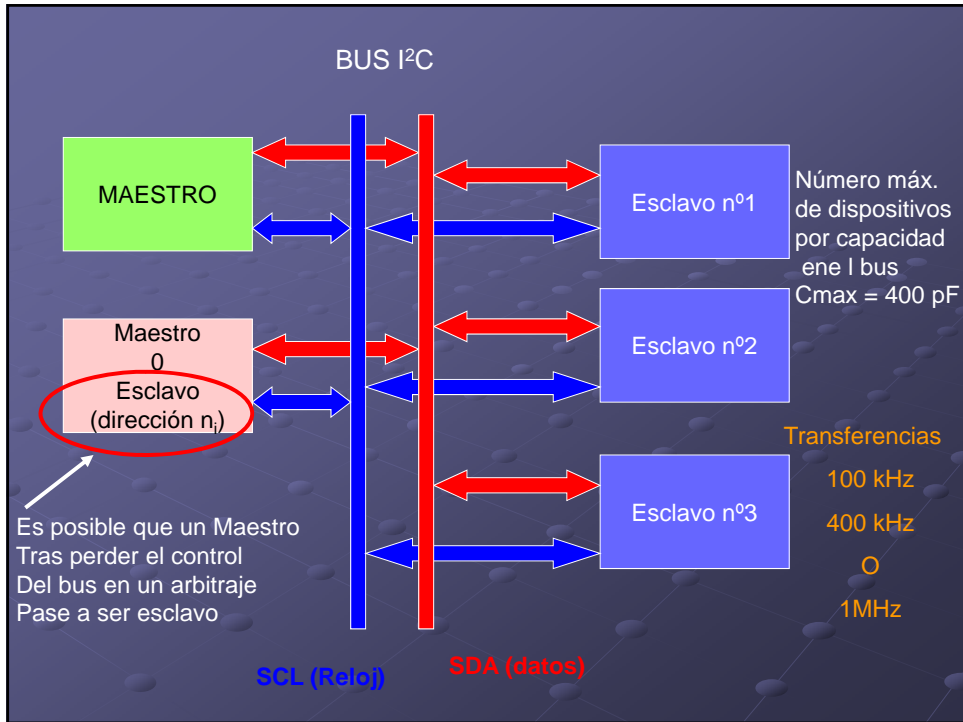
- **Puerto Serie Síncrono:** Interface de Comunicación Serie Síncrona Pensado para comunicación con otros Microcontroladores o Periféricos:
 - EEPROM serie (almacenamiento de datos no volátiles)
 - Registros de Desplazamiento (expansión de entradas y/o salidas)
 - Drivers de Displays (reducción de conexiones)
 - Conversores A/D (digitalización externa de señales)
 -
- **Modos de Operación posibles del módulo SSP:**
 - SPI (Serial Peripheral Interface): Interface de Periféricos Serie: SPI: Es una Marca Registrada de Motorola Corporation
 - I2C (Inter-Integrated Circuit): Entre Circuitos Integrados I2C: Es una Marca Registrada de Philips

Generalidades del BUS I²C

- Es un interface síncrono a 2 hilos pensado para la comunicación entre C.I. y desarrollado por Philips Corporation. Es un bus estándar para muchos equipos de consumo y telecomunicaciones
- Las dos líneas de comunicación corresponden a reloj (SCL) y datos (SDA).
- Las transferencias son por tanto semidúplex.
- En el bus debe existir al menos un dispositivo Maestro que genera la señal de reloj y uno o varios Esclavos que reciben tal señal SCL.
- El dispositivo Maestro es el único que tiene capacidad de iniciar la transferencia, decidir con quién se realiza, el sentido de la misma (envío o recepción del Maestro) y cuándo se finaliza

Generalidades del BUS I²C

- Cada dispositivo esclavo conectado al bus tiene asignada una dirección única codificada en 7 (ó en 10 bits de manera opcional).
- Se admite la presencia de varios Maestros en el bus (sistema multi-Maestro) con un arbitraje que asegura que en cada instante sólo hay uno dominante



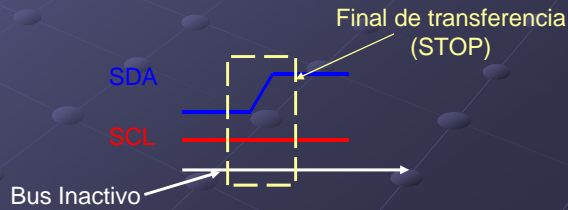
Estados en el Bus

- En el ESTADO INACTIVO, las líneas SDA y SCL se encuentran a "1": los transistores de las etapas de salida de todos los dispositivos se encuentran en corte
- El INICIO de transferencia se produce cuando aparece una transición de "1" a "0" en la línea SDA manteniéndose la línea de reloj SCL a "1".

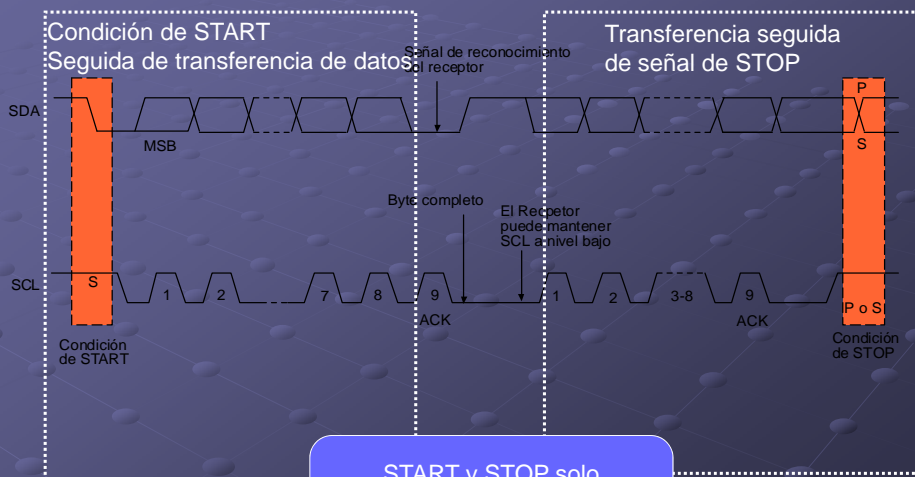


- El FIN de transferencia es por transición de "0" a "1" en SDA cuando SCL está a "1"

En una transmisión, SDA sólo puede cambiar cuando SCL="0" si no, se interpretaría como START ó STOP



Estado del Bus I²C



START y STOP solo pueden ser generadas por el Maestro

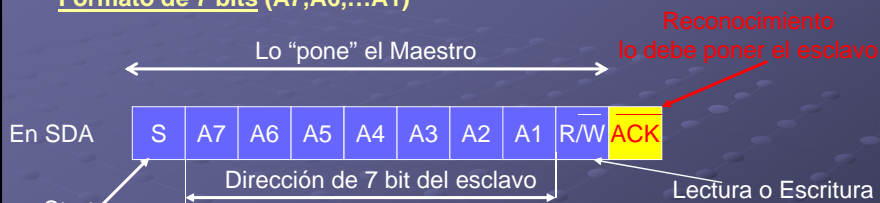
Comunicación con dispositivos

- Cuando un Maestro decide iniciar una transferencia, **envía primero la dirección del dispositivo** con el que quiere "hablar".
- Todos los dispositivos "escuchan" para determinar **si la dirección colocada en el bus es la suya**. Acompañando a la dirección aparece un **bit R/W** que indica al esclavo destinatario si el Maestro quiere "leer" (recibir si R/W=1) del Esclavo o si el Maestro quiere "escribir" (enviar si R/W=0) en el Esclavo.
- El Maestro y el Esclavo direccionado deben encontrarse siempre en **estados opuestos**:
 - Maestro emite / Esclavo recibe
 - Maestro recibe / Esclavo emite
- En cualquiera de los casos, el Maestro será el que **genere la señal de reloj en la línea SCL**.
- Si hay más de un Maestro en el bus, se debe producir un **arbitraje** cuando $SCL=1$: el Maestro que envía un "1" y detecta un "0" en la línea SDA "pierde" en el arbitraje y desactiva su salida de datos (aunque podría seguir generando flancos en la línea SCL hasta que finalizara el envío del byte donde "perdió" el control del bus)

Direccionamiento de dispositivos en el bus I²C

- Hay dos formatos de direccionamiento posibles (se elige uno):

Formato de 7 bits (A7,A6,...A1)

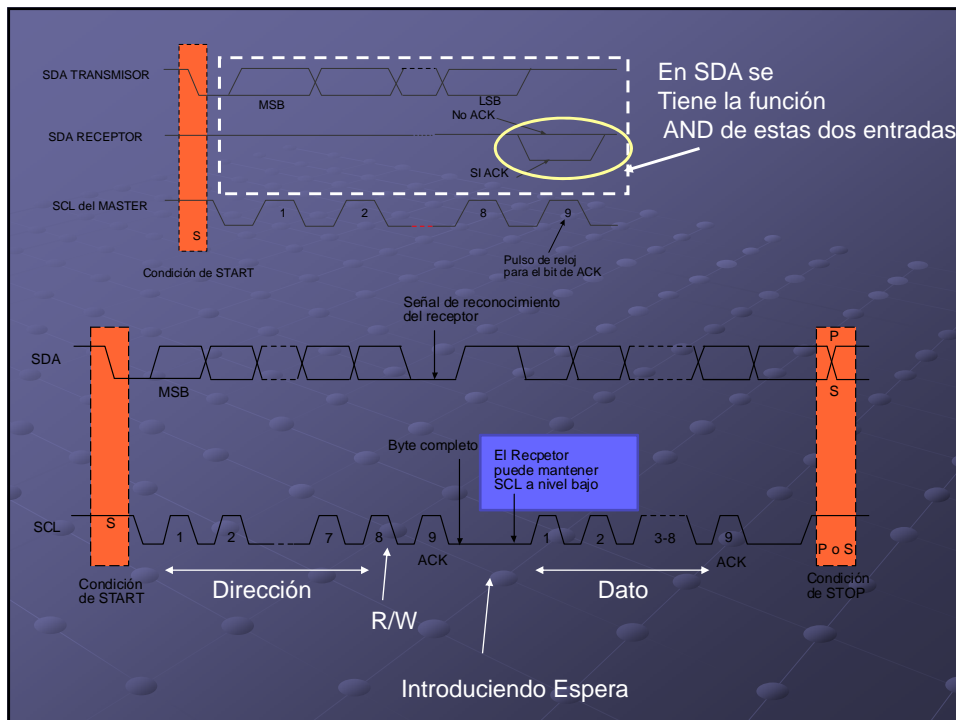


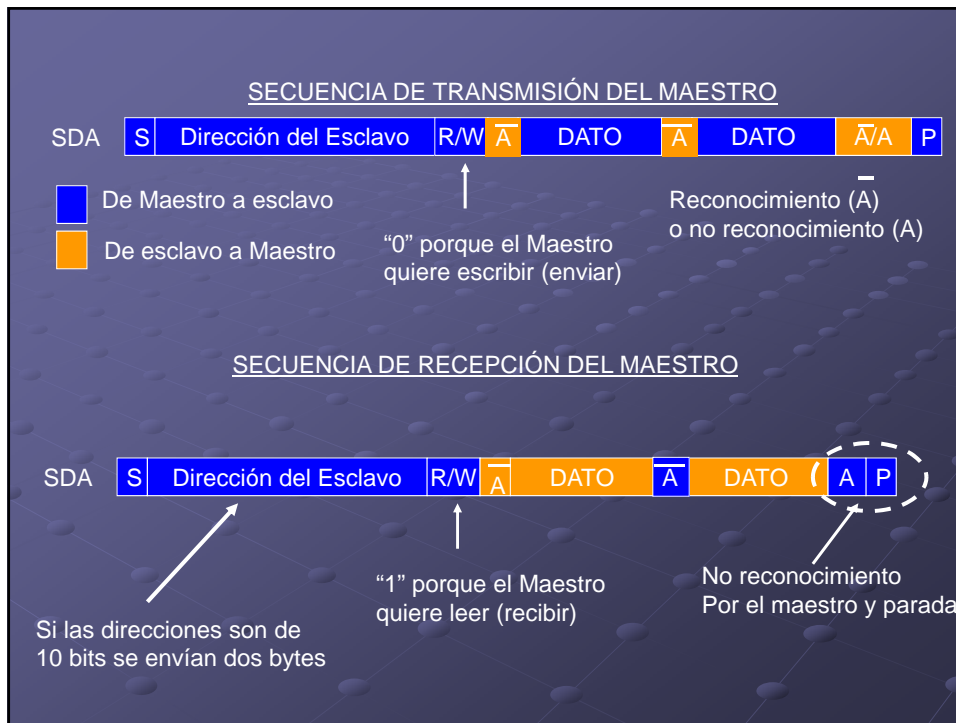
Formato de 10 bits (A9,A8,...A0)



Reconocimiento de transferencia ACK

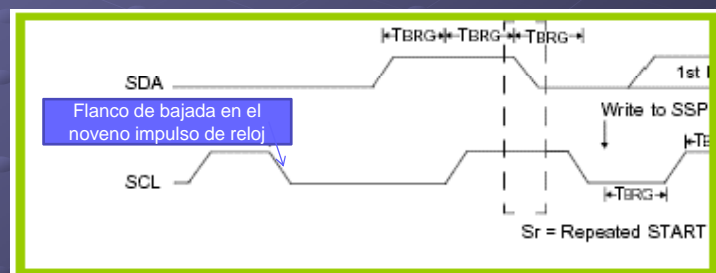
- Los datos se transmiten en paquetes de 8 bits (bytes), no hay límite en cuanto a número de bytes, pero **después de cada byte** se debe intercalar un **bit de reconocimiento (ACK)** por parte del dispositivo receptor
- Si es el Esclavo el receptor y **no genera el bit de reconocimiento** después de cada byte, el Maestro **debe abortar la transferencia generando un bit de fin (STOP)**.
- Dado que las etapas de salida son de drenador abierto, **los niveles lógicos "0" son dominantes** en las líneas, por tanto el **Esclavo debe dejar su salida SDA a "1"** para que el Maestro **pueda generar el STOP** (paso de "0" a "1" con SCL="1").
- Si el dispositivo **Maestro es el receptor**, genera un ACK tras cada byte recibido, permitiendo al Esclavo que continúe enviando bytes. Si el Maestro **decide finalizar la transferencia**, genera un bit de STOP en lugar de colocar un bit de reconocimiento
- Si el Esclavo necesita **retardar el envío del siguiente byte** (porque no lo tiene todavía disponible, p.e.), puede situar la línea SCL en estado bajo y forzar al Maestro a situarse en un estado de "Espera" puesto que no se podrían generar flancos en SCL





Repetición de START (Sr)

- Si un dispositivo Maestro no desea abandonar el bus tras una transferencia, en lugar de generar un bit de STOP y volver a generar el nuevo START, puede generar una **Repetición de START (Sr)**.
- La **Repetición de START** es idéntica a la condición de START (SDA pasa de 1 a 0 con SCL en estado alto) pero se produce tras un pulso de reconocimiento de bus (ACK) y no desde un estado inactivo

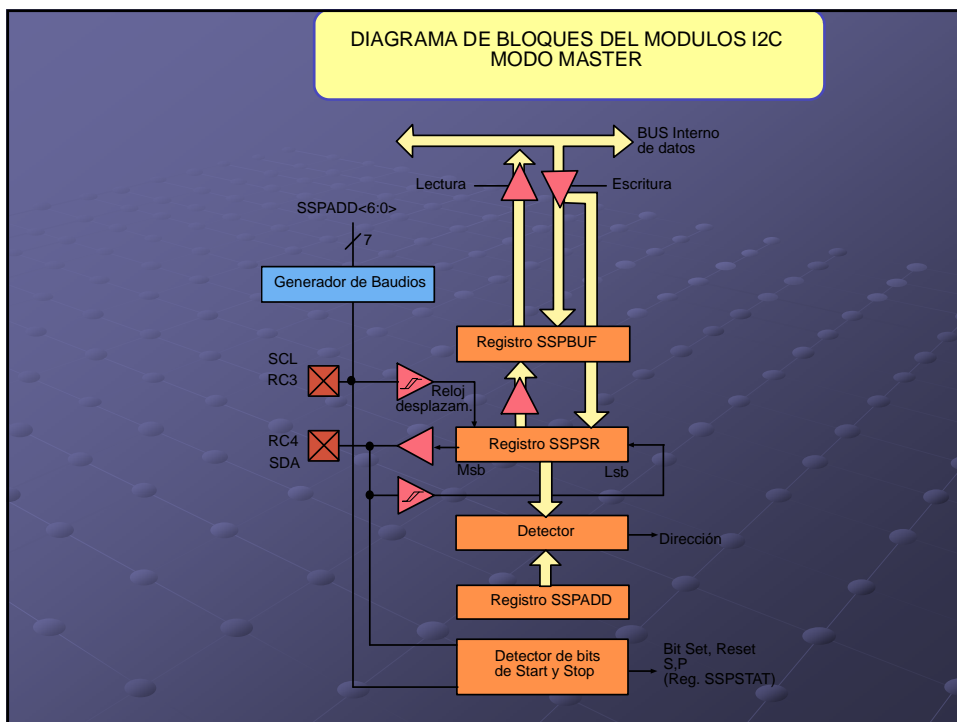
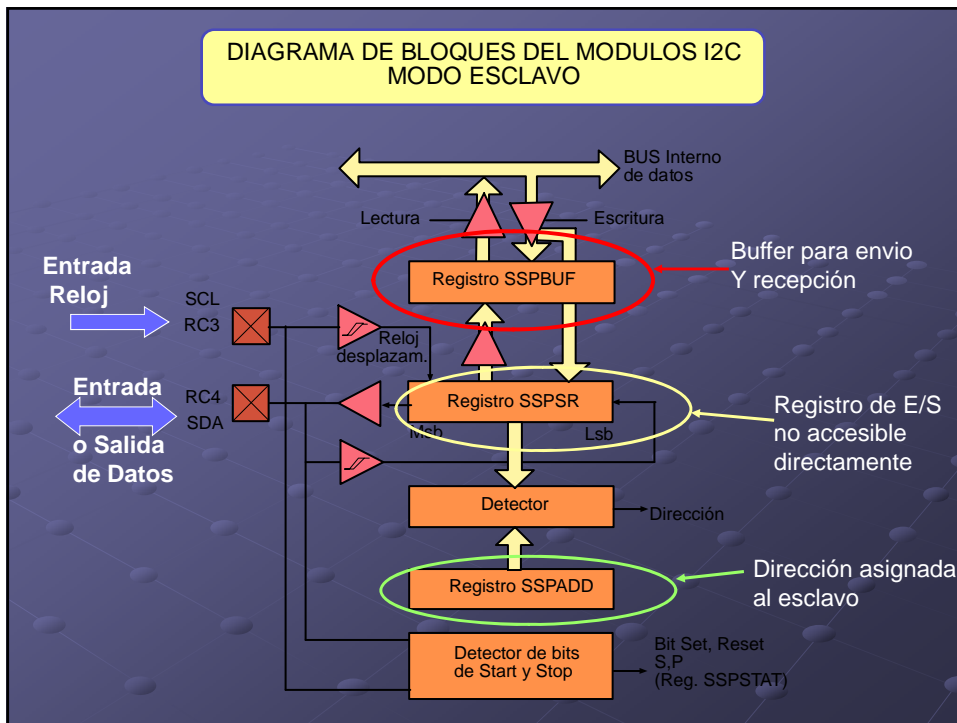


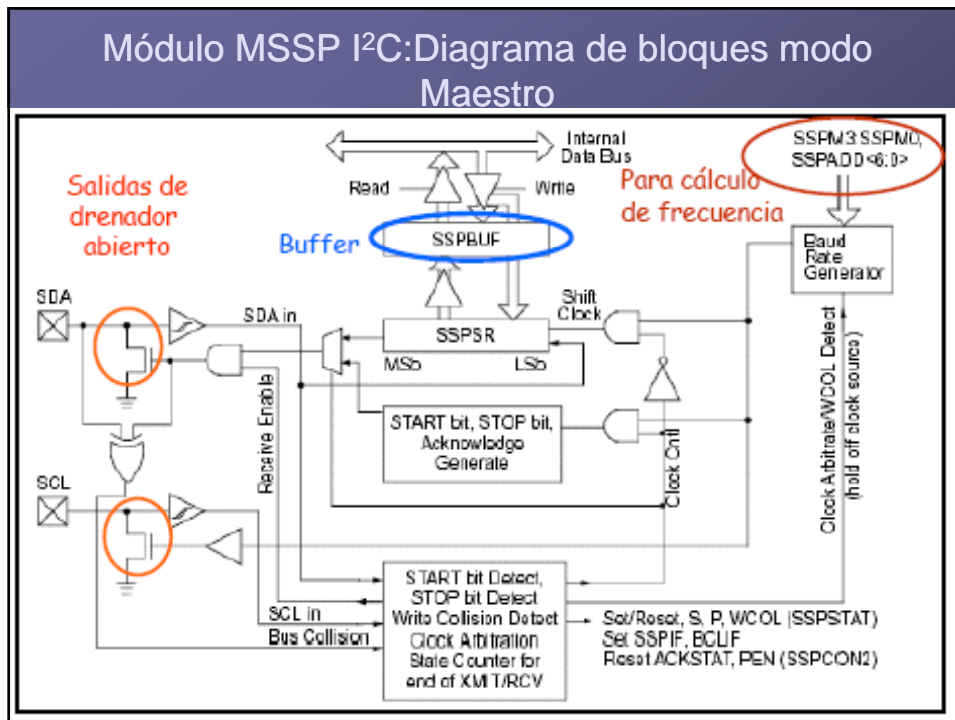
Interface I²C de los PIC16F87X

- **MSSP** Master Synchronous Serial Port, que para el bus I²C implementa:
 - Modo Esclavo: Completo por Hw
 - Modo Maestro: Completo por Hw

Módulo MSSP I²C

- Implementa **todas las funciones del Maestro** y del **Esclavo** en un bus I²C por hardware
- Permite detectar las condiciones START y STOP en el bus **por interrupción**.
- Permite emplear **direcciones de 7 ó de 10 bits** como Esclavo
- Se puede seleccionar uno de los siguientes modos de operación:
 - Esclavo I²C con **dirección de 7 bits**
 - Esclavo I²C con **dirección de 10 bits**
 - Maestro I²C con reloj SCL de **frecuencia = $F_{osc}/(4*(SSPADD+1))$**
- El módulo MSSP dispone de 6 registros asociados:
 - **SSPCON** Registro de Control
 - **SSPCON2** Registro de Control 2
 - **SSPSTAT** Registro de Estado
 - **SSPADD** Registro de Dirección
 - **SSPBUF** Buffer de Transmisión/Recepción
 - **SSPSR** Registro de Desplazamiento Serie (no accesible)





Módulo MSSP I²C: Modo Maestro

- En modo Maestro, las líneas SDA y SCL son manejadas por el hardware del módulo MSSP, no es necesario definir su dirección en TRISC
- Se dispone de circuitos lógicos **para detección de START (S)** y **STOP (P)** y su aparición queda reflejada en los bits S y P de SSPSTAT. Estos bits son de **sólo lectura** y quedan a cero tras un reset o cuando el módulo MSSP está desactivado.
- El **flag SSPIF que permite interrupciones** es sensible a los siguientes eventos:

- | | |
|--|---|
| <ul style="list-style-type: none"> Condición de START Transferencia de un byte START repetido | <ul style="list-style-type: none"> Condición de STOP Transmisión de ACK |
|--|---|

Módulo MSSP I2C: Modo Maestro

- En buses **multi-Maestro**, el control del mismo sólo puede intentarse cuando el bit P **está a 1 ó cuando el bus está inactivo** (S y P a "0"). En este caso, la línea SDA será monitorizada para el arbitraje y comprobar si el nivel presente coincide con el nivel esperado y propuesto por el propio Maestro, esta verificación la realiza el hardware y el resultado queda reflejado en el bit de **flag de colisiones BCLIF (PIR2<3>)** **si hay discrepancia** y podría generar interrupción (su máscara es BCLIE = PIE2<3>).

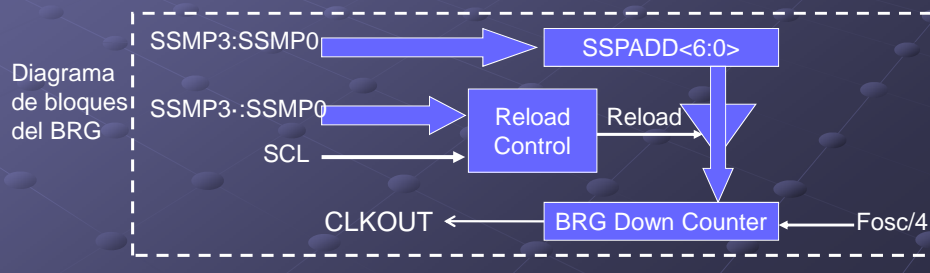
I2C Maestro: Generador de relación de Baudios (BRG)

- El **generador de relación de baudios (BRG)** empleado en el modo SPI es el utilizado para generar la frecuencia del reloj SCL. La frecuencia viene dada por la expresión:

$$F_{SCL} = \frac{F_{OSC}}{4 * (SSPADD + 1)}$$

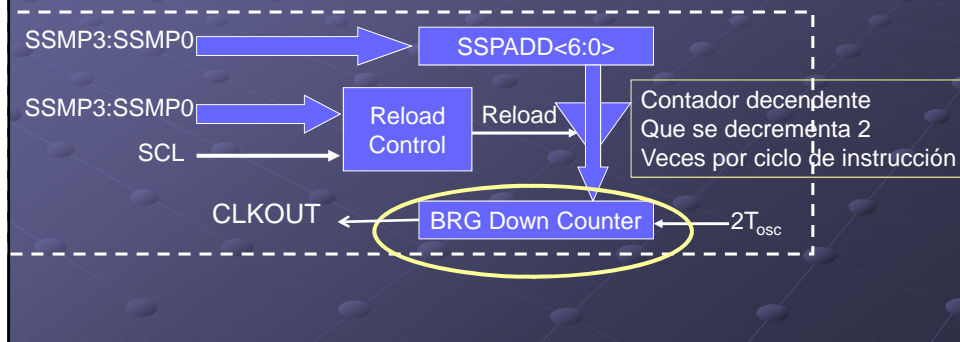
En I2C las Frecuencias estándar Son 100 KHz, 400KHz y 1MHz

- Se emplean los 7 bits menos significativos del registro SSPADD. En modo Maestro BRG se actualiza automáticamente con el cambio en SSPADD, pero en un bus multi-Maestro, se carga cuando SCL pasa de 0 a estado alto



I2C Maestro: Generador de relación de Baudios (BRG)

- El contador BRG se decrementa dos veces por ciclo de instrucción, y marca un tiempo T_{BGR} para establecer secuencias de STOP, START, etc



I²C MAESTRO: Operación

- El Maestro **generará los flancos en SCL** y las condiciones de **START** y **STOP**.
- Una transferencia **finaliza** con una condición de **STOP (P)** o con una **repetición de START (Sr)**, en este caso se trata también del inicio de la siguiente transferencia serie con lo que **el bus I2C no quedaría libre**.
- El modo Maestro **funciona poniendo a "1" ó "0" los bits apropiados del registro SSPCON** y habilitando el módulo con el bit `SSPEN`. Son posibles 6 acciones:
 - Establecer una **condición de START** en las líneas SDA y SCL (**bit SEN**)
 - Escribir un dato** en SSPBUF para iniciar una transmisión de dato o dirección
 - Configurar la entrada I2C para **recibir datos** (**bit RCEN**)
 - Generar un **pulso de reconocimiento (ACK)** (**bit ACKEN**)
 - Generar una **condición de STOP** en SDA y SCL (**bit PEN**)
 - Establecer una **repetición de START** en las mismas líneas (**bit RSEN**)
- Una acción generada por el Maestro **debe esperar a que finalice la anterior** (p.e. no se puede escribir el dato en SSPBUF hasta que se haya completado la condición START)

I²C MAESTRO: Operación

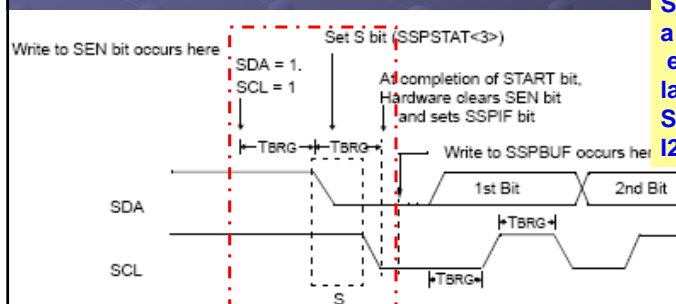
- Debe quedar claro que **las acciones que va ejecutando el Maestro son totalmente independientes unas de otras.**
- Como ejemplo: no es posible realizar **en un solo** paso la generación del bit de START y el envío de la dirección del Esclavo (éstas serían dos acciones independientes)
- Para realizar una transferencia completa por parte de un Maestro, emitiendo o recibiendo, será preciso **encadenar las diferentes acciones individuales** de la manera adecuada, interpretando los resultados obtenidos y los estados presentes en el bus de manera previa a la ejecución de la siguiente acción.
- Será necesario que el Maestro **compruebe que la acción** que él está "intentando" realizar, efectivamente se produce como verificación a su "dominio" sobre el bus
- Si el microcontrolador **no dispusiera de módulo MSSP** para realizar las acciones por hardware, debería ejecutar cada una de ellas mediante código interno (firmware). También es posible que aún disponiendo de módulo MSSP, se prescindiera de la implementación hardware y **se realice también en el firmware.**

I²C MAESTRO: Operación

- Ejemplo de Secuencia típica de **transmisión de un byte** por parte del Maestro:
 - a) Se **genera START** poniendo a 1 el bit de habilitación de START (SEN) del registro SSPCON2
 - b) Se esperará el tiempo necesario para **detectar START**, cuando se haya dado la condición, **SSPIF=1** (se debe poner a 0 por software)
 - c) Se **carga SSPBUF con la dirección** a enviar por el bus y el bit R/W=0 (dir. 7 bits)
 - d) Los bits de la dirección van saliendo por la línea SDA hasta completar 8 bits
 - e) Se "lee" el **bit de reconocimiento** (ACK) recibido del esclavo y se introduce ese bit en SSPCON2<6>
 - f) Al final del 9º flanco en SCL, se pone **SSPIF=1** (hay que resetearlo)
 - g) Se **cargan en SSPBUF los 8 bits del dato** a enviar
 - h) Los bits del dato van saliendo por SDA hasta completar la transmisión
 - i) Se "lee" el bit de reconocimiento (ACK) y se graba su valor en SSPCON<6>
 - j) Al final del 9º flanco en SCL se pone el flag **SSPIF** a 1 (a 0 por programa)
 - k) Se **genera una condición de STOP** poniendo a 1 el bit de habilitación de STOP (PEN) de SSPCON2
 - l) Una vez **detectada** la condición de **STOP**, se pone a 1 el flag **SSPIF**

Acción 1 del Maestro I²C: Generación del Start

- Para generar condición de **START**, el usuario debe poner a "1" el bit SEN (SSPCON2<6>). Si los pines SCL y SDA están a 1, el generador de relación de baudios (BRG) se carga con el contenido de SSPADD<6:0> empezando a decrementarse. Si al llegar al final de la cuenta (T_{BRG}) se muestrean SCL y SDA y ambos permanecen a uno, el pin SDA se pone en estado bajo (es por tanto condición de START).
- Si se detecta la condición generada, el bit S (SSPSTAT<3>) se pone a 1, se cargará a continuación de nuevo BRG y cuando haya transcurrido T_{BRG} , el bit SEN se pondrá a 0 por hardware. Si se intenta escribir en SSPBUF durante secuencia de START, entonces bit WCOL=1 y no cambia SSPBUF.



Si SCL y SDA no estuvieran a 1, se considera colisión en el BUS (BCLIF=1) se para la generación de START y se pasa a estado inactivo de I2C

Acción 1 del Maestro I²C: Generación del START

```

; Subrutina "I2C_EnviaStart" -----
; Esta subrutina envía una condición de Start o inicio.
;
I2C_EnviaStart
    bcf    PIR1,SSPIF           ;limpia SSPIF
    bsf    STATUS,RP0          ;Selecciona página 1
    bsf    SSPCON2,SEN        ;Activa secuencia de inicio
    bcf    STATUS,RP0          ;Selecciona página 0

Espera_Start
    btfss  PIR1,SSPIF         ;Fin de secuencia de inicio ??
    goto  Espera_Start        ;No, esperar
    return
  
```


Maestro I²C: Colisiones

- Resulta importante distinguir entre los flag **WCOL** y **BCLIF** ya que son indicadores de conceptos distintos de colisiones.
 - **WCOL**: es un indicador de "colisión" por acceso al registro **SSPBUF** debido al intento de escritura en un estado no válido del bus I²C si el dispositivo es Maestro o con intento de escritura en **SSPBUF** cuando todavía se está enviando un dato anterior
 - **BCLIF** es un indicador de "colisión" en el bus I²C (otro tipo por tanto), se produce cuando estando la línea SCL libre, y el estado esperado en SDA es el "1" pero el muestreo resulta ser el "0".

La colisión puede darse durante un **START**, un **STOP** o una **repetición de START**.

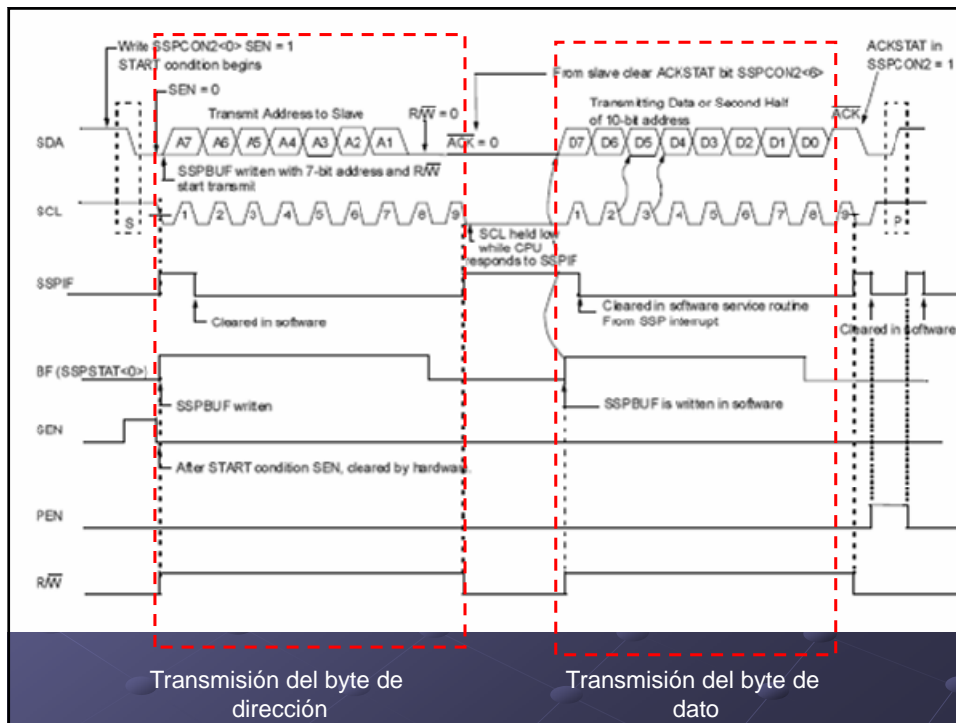
BCLIF puede tener asociada una interrupción.

Acción 2 del Maestro I²C: Transmisión de un byte

- El envío de un byte, sea dirección o dato se inicia con la escritura del mismo en el registro **SSPBUF**. Esto hará que el flag **BF** se ponga a 1 y que el BRG se ponga en marcha. Cada bit saldrá por SDA en los flancos de bajada de la línea SCL.
- Tras la salida del 8º bit por SDA, el flag **BF** se pone a 0 y el Maestro "libera" la línea SDA para que el Esclavo direccionado responda con el bit de reconocimiento (ACK).
- El estado **ACK** se carga en el bit **ACKSTAT** (=SSPCON2<6>) en el flanco de bajada del 9º pulso en SCL. Tras completar el dato, se da **SSPIF=1** y la generación de reloj se suspende hasta que se escriba un nuevo dato en **SSPBUF**, la línea SCL queda en estado bajo y SDA no cambia

Flags en Transmisión

- El flag **BF** (SSPSTAT<0>) es de solo lectura y se pone a 1 cuando se escribe un byte en **SSPBUF**, vuelve a 0 cuando hayan sido enviados los 8 bits
- El bit **WCOL** (SSPCON<7>) es de lectura/escritura y se pone a 1 cuando se intenta escribir en **SSPBUF** en el transcurso de un envío. Debe ponerse a 0 por software
- El bit **ACKSTAT** (SSPCON2<6>) es de lectura/escritura, se pone a 0 cuando el esclavo ha enviado un bit de reconocimiento (ACK) y a 1 si no lo ha enviado



Acción 2 del Maestro I²C: Transmisión de un byte

```
; Subrutina "I2C_EnviaByte" -----
; El microcontrolador maestro transmite un byte por el bus I2C, comenzando
; por el bit MSB. El byte a transmitir debe estar cargado previamente en el
; registro de trabajo W.
; De la subrutina ejecutada anteriormente I2C_EnviaStart o esta misma
; I2C_EnviaByte, la línea SCL se debe encontrar a nivel bajo al menos durante
; 5 µs.
```

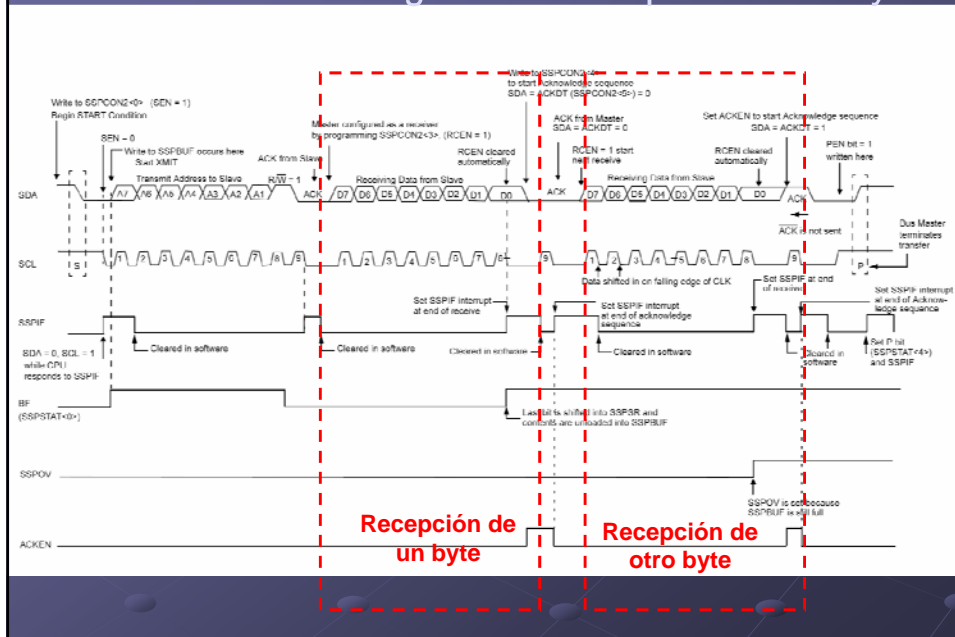
```
I2C_EnviaByte
    bcf    PIR1,SSPIF    ;Limpia SSPIF
    movwf SSPBUF        ;Byte a transmitir pasa al buffer de salida
Espera_Envia_Byte
    bitfs  PIR1,SSPIF    ;Recibido el bit /ACK ??
    goto  Espera_Envia_Byte ;No, esperar
    return
```

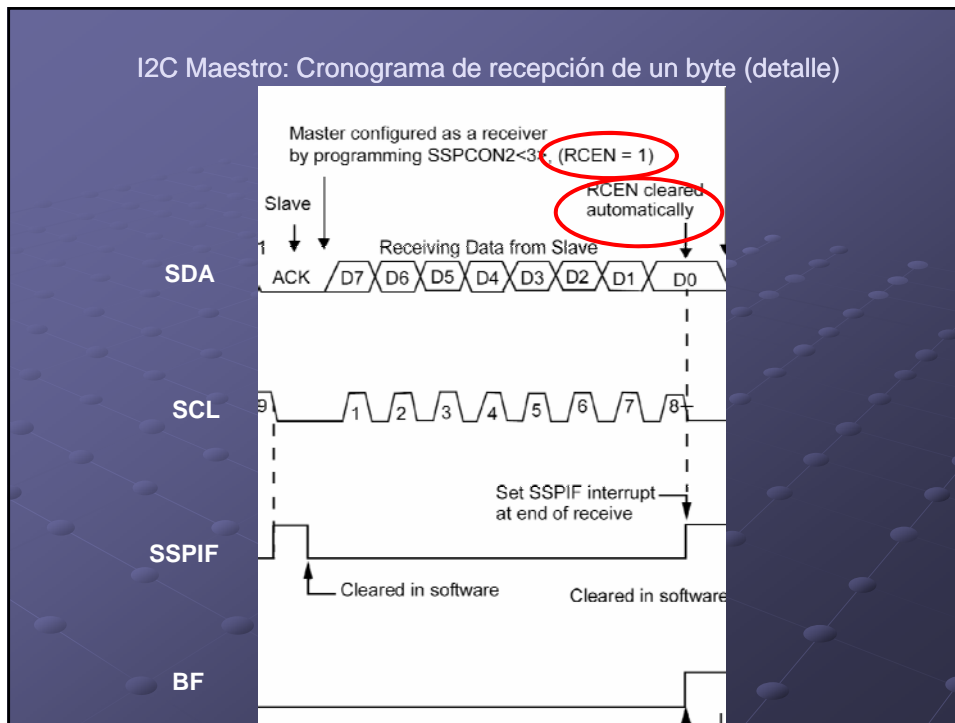
Acción 3 del Maestro I²C: Recepción de un byte

- La recepción de un byte se inicia con la puesta a "1" del bit de habilitación de la recepción RCEN (SSPCON2<3>). Tras esa acción aparecerán los flancos en la línea SCL e irán entrando los bits por la línea SDA.
- Quando se hayan completado 8 flancos en SCL, el bit RCEN se pondrá a 0 de manera automática, el contenido de SSPSR pasará a SSPBUF y flags BF y SSPIF se pondrán a 1. El reloj dejará de generar flancos, se detendrá y dejará la línea SCL a "0".
- El MSSP quedará ahora inactivo y a la espera de otra acción. Cuando se lea SSPBUF, el flag BF se pondrá a 0 por hardware. El Maestro podría enviar ahora el bit de reconocimiento al final de la recepción poniendo el bit ACKEN=1 (SSPCON2<4>), pero eso ya formaría parte de otra acción

- El flag BF (SSPSTAT<0>) en recepción se pone a 1 cuando se carga SSPBUF desde el registro SSPSR, se pondrá a 0 cuando se haya leído SSPBUF.
- El bit WCOL (SSPCON<7>) se pone a 1 cuando se intenta escribir en SSPBUF cuando una recepción está en curso (SSPBUF no cambia)
- El bit SSPOV (SSPCON<6>) en recepción se pone a 1 cuando se ha completado un dato en SSPSR y el flag BF está todavía a 1 por una recepción previa

I²C Maestro: Cronograma de recepción de un byte





Acción 3 del Maestro I²C: Recepción de un byte

```

; Subrutina "I2C_LeeByte" -----
; Lee un byte procedente del dispositivo I2C seleccionado y lo devuelve en W
; Seguidamente se genera y transmite el bit /ACK
I2C_LeeByte
    bcf    PIR1,SSPIF    ;Restaura el flag del módulo MSSP
    bsf    STATUS,RP0    ;Selecciona página 1
    bsf    SSPCON2,RCEN  ;Activa el modo receptor
    bcf    STATUS,RP0    ;Selecciona página 0
Espera_Lectura
    btfss  PIR1,SSPIF    ;Recibidos los 8 bits ??
    goto  Espera_Lectura ;No, esperar
    bcf    PIR1,SSPIF    ;Restaura el flag del módulo MSSP
    btfsc  ACK,0         ;Envia ACK??
    goto  I2C_EnviaNACK  ;No, Iniciar I2C_EnviaNACK
    bsf    STATUS,RP0    ;Selecciona página 1
    bcf    SSPCON2,ACKDT ;Pone bit ACK a "0"
    bsf    SSPCON2,ACKEN ;Activa la secuencia de generación del bit ACK
    bcf    STATUS,RP0    ;Selecciona página 0
Espera_ACK
    btfss  PIR1,SSPIF    ;Secuencia ACK finalizada ??
    goto  Espera_ACK     ;No, esperar
    movf  SSPBUF,W        ;Lee el byte recibido
    return

```

Acción 4 de Maestro I²C: Generación de NACK

```

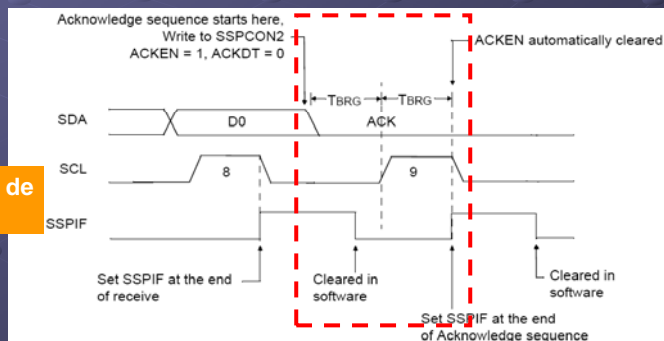
*****
,
I2C_EnviaNACK
    bsf STATUS,RP0      ;Selecciona Página 1
    bsf SSPCON2,ACKDT   ;Pone el bit ACK a "1"
    bsf SSPCON2,ACKEN   ;Activa la secuencia de generación del bit NACK
                        ;generación del bit NACK
    bcf STATUS,RP0      ;Selecciona la página 0
Espera_NACK
    btfss PIR1,SSPIF    ;Ha finalizado la secuencia NACK?
    goto  Espera_NACK
    movf  SSPBUF,W      ;Lee el byte recibido
    return

```

Acción 4 de Maestro I²C: Generación de ACK

- La secuencia de generación de un **pulso de reconocimiento (ACK)** se activa mediante el **bit de habilitación ACKEN** (SSPCON2<4>). Si este bit está a 1, el pin SCL queda retenido a 0 y el contenido del bit ACKDT (SSPCON2<5>) aparece en la línea SDA.
- Si se desea **generar pulso de reconocimiento**, se debe colocar **ACKDT=0**, si se desea **colocar no reconocimiento**, **ACKDT** se debe poner a 1.
- Los tiempos para **generar el pulso ACK**, se basan en el empleo del generador de baudios: tras un tiempo T_{BRG} la línea SCL se libera (vuelve a 1). Tras detectar el flanco de subida, permanece a 1 durante un tiempo T_{BRG} . Transcurrido ese tiempo SCL vuelve a 0, el bit **ACKEN se borra de manera automática** y el módulo MSSP pasa a inactivo.

Generación de ACK



Acción 4 de Maestro I²C: Generación de ACK

I2C_Envia_ACK

```

bcf    PIR1,SSPIF           ;Limpia SSPIF
bsf    STATUS,RP0
bcf    SSPCON2,ACKDT       ;Pone a "0" el bit ack
bsf    SSPCON2,ACKEN      ;Activa secuencia NACK
bcf    STATUS,RP0

```

Espera_ACK_1

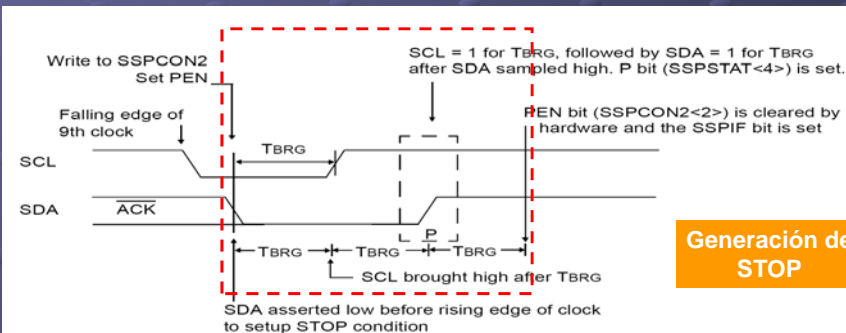
```

btfss  PIR1,SSPIF         ;Secuencia ACK finalizada ??
goto   Espera_ACK_1      ;No, esperar
return

```

Acción 5 de Maestro I²C: Generación de STOP

- La secuencia de STOP al final de una transmisión o recepción en la línea SDA se consigue poniendo a 1 el bit PEN (SSPCON2<2>).
- Al acabar una transmisión la línea SCL queda en estado bajo, si el bit PEN se pone a 1, la línea SDA se pondrá a 0, si se detecta esa situación, el BRG (generador de relación de baudios) contabilizará un tiempo igual a T_{BRG} y subirá SCL a 1 y volverá a contabilizar el mismo tiempo T_{BRG} para liberar la línea SDA (STOP).
- La condición será detectada por el bit P (SSPSTAT<4>) que se pone a 1, un tiempo TBRG posterior, el bit PEN volverá a 0 por el propio hardware.



Acción 5 de Maestro I²C: Generación de STOP

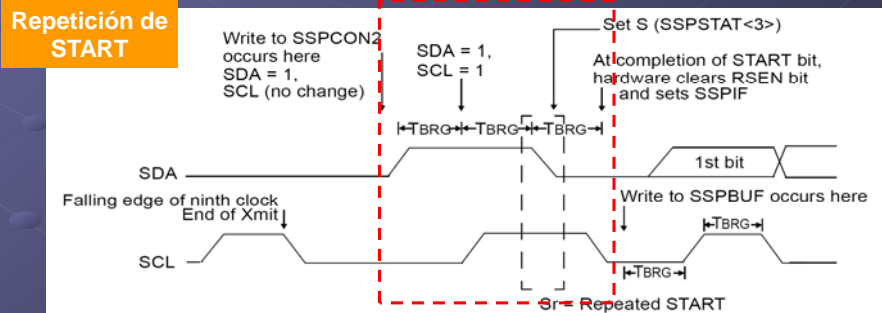
```

; Subrutina "I2C_EnviaStop" -----
; Esta subrutina envía un condición de Stop o parada.
;
I2C_EnviaStop
    bcf    PIR1,SSPIF           ;Limpia SSPIF
    bsf    STATUS,RP0          ;Selecciona página 1
    bsf    SSPCON2,PEN        ;Activa secuencia de stop
    bcf    STATUS,RP0
Espera_Stop
    btfss  PIR1,SSPIF         ;Fin de secuencia de Stop ??
    goto  Espera_Stop        ;No, esperar
    return

```

Acción 6 de Maestro I²C: Repetición de START

- La repetición de START se puede producir si el bus está desocupado y se activa el bit **RSEN** (SSPCON2<1>). En ese momento se libera SDA que pasa a 1, tras un tiempo T_{BRG} SCL se libera también. SDA y SCL deben mantenerse ambos a 1 durante un tiempo T_{BRG} tras el cual se "baja" SDA manteniendo SCL a 1. Esta situación debe permanecer durante un tiempo T_{BRG} equivalente (véase cronograma inferior).
- Transcurrida esta secuencia, el bit **RSEN** pasa a 0 de modo automático y deja la línea SDA a 0. Tan pronto como la condición de START se detecta en las líneas del bus, el bit **S** (SSPSTAT<3>) se pondrá a 1. El flag SSPIF se pondrá a 1 posteriormente una vez transcurrido un tiempo T_{BRG} .



Registro PIR1 (0Ch)

| | | | | | | | |
|-------|-------|-------|-------|--------------|--------|--------|--------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| PSPIF | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF |
| bit7 | | | | | | | bit 0 |

SSPIF: Flag de interrupción del Puerto Serie Sincrono (SSP)

En Modo SPI

Ha tenido lugar una Transmisión/Recepción

En Modo I2C Slave

Ha tenido lugar una Transmisión/Recepción

En Modo I2C Master

Ha tenido lugar una Transmisión / Recepción

La condición de salida iniciada se completó por el módulo de SSP.

La condición de parada inicio se completó por el módulo de SSP.

La condición de reinicialización se completo por el módulo SSP

Una condición de la salida se ha realizado mientras el módulo de SSP estaba en estado de espera (sistema de Multimaster).

Registro PIR2 (0Dh)

| | | | | | | | |
|------|-----------|-----|-------|--------------|-----|-----|--------|
| U-0 | R/W 0 | U-0 | R/W-0 | R/W-0 | U-0 | U-0 | R/W 0 |
| -- | Reservado | -- | EEIF | BCLIF | -- | -- | CCP2IF |
| bit7 | | | | | | | bit 0 |

BCLIF: Flag que indica la colisión en el bus SSP

1 = Se ha producido una colisión de bus SSP

0 = No se ha producido colisión en el bus SSP

Registro SSPSTAT (94h)

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|------------|-------|------|-----|-----|------|-----|-------|
| SMP | CKE | D/#A | P | S | R/#W | UA | BF |
| bit7 | | | | | | | bit 0 |

SMP : Fase de datos de entrada SPI

Modo Master

- 1 = Muestra datos de entrada al final del tiempo de salida
- 0 = Muestra datos de entrada a la mitad del tiempo de salida

Modo Esclavo SPI

SMP debe borrarse cuando el SPI se emplea en modo esclavo

Modo I²C Master o Esclavo

- 1 = Deshabilita control de variaciones para velocidad estándar (100KHz y 1MHz)
- 0 = Habilita control (400 KHz)

Registro SSPSTAT (94h)

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|-------|------------|------|-----|-----|------|-----|-------|
| SMP | CKE | D/#A | P | S | R/#W | UA | BF |
| bit7 | | | | | | | bit 0 |

CKE : Selección del flanco de reloj

Modo SPI

Con CKP = 0

- 1 = Dato transmitido en flanco ascendente de SCK
- 0 = Dato transmitido en flanco descendente de SCK

Con CKP = 1

- 1 = Dato transmitido en flanco descendente de SCK
- 0 = Dato transmitido en flanco ascendente de SCK

Modo I²C Master o Esclavo

- 1 = Dato transmitido en flanco descendente de SCK
- 0 = Dato transmitido en flanco ascendente de SCK

Registro SSPSTAT (94h)

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|-------|-------|------|-----|-----|------|-----|-------|
| SMP | CKE | D/#A | P | S | R/#W | UA | BF |
| bit7 | | | | | | | bit 0 |

D/A : Bit de Datos/Direcciones (solo en modo I²C)

1 = Indica el último byte recibido o transmitido es un dato

0 = Indica que el último byte recibido o transmitido es una dirección.

P: Bit de Stop (solo en modo I²C)

1 = Indica que se ha detectado un bit de Stop (este bit es "0" en RESET)

0 = Bit de Stop no detectado

S: Bit de Start (solo en modo I²C)

1 = Indica que se ha detectado un bit de Start

0 = Bit de Start no detectado

Registro SSPSTAT (94h)

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|-------|-------|------|-----|-----|------|-----|-------|
| SMP | CKE | D/#A | P | S | R/#W | UA | BF |
| bit7 | | | | | | | bit 0 |

D/A : Bit de Datos/Direcciones (solo en modo I²C)

1 = Indica el último byte recibido o transmitido es un dato

0 = Indica que el último byte recibido o transmitido es una dirección.

P: Bit de STOP (solo en modo I²C)

1 = Indica que se ha detectado un bit de Stop (este bit es "0" en RESET)

0 = Bit de Stop no detectado

S: Bit de START (solo en modo I²C)

1 = Indica que se ha detectado un bit de Start

0 = Bit de Start no detectado

Registro SSPSTAT (94h)

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|-------|-------|------|-----|-----|------|-----|-------|
| SMP | CKE | D/#A | P | S | R/#W | UA | BF |
| bit7 | | | | | | | bit 0 |

R/W : Bit de I (solo en modo Información de Lectura/Escritura (Solo en modo I²C))

I²C Modo Esclavo

1 = Escritura

0 = Lectura

I²C Modo Master

1 = Transmisión en progreso

0 = No hay transmisión en progreso

UA: Activación de dirección (solo en modo I²C 10 bits)

1 = Indica que es necesario actualizar la dirección en el registro SSPADD

0 = no es necesario actualizar la dirección

Registro SSPSTAT (94h)

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|-------|-------|------|-----|-----|------|-----|-------|
| SMP | CKE | D/#A | P | S | R/#W | UA | BF |
| bit7 | | | | | | | bit 0 |

BF : Bit de estado de buffer lleno

Receptor (Modo SPI e I²C)

1 = Recepción completa SSPBUF está lleno

0 = Recepción no completa el SSPBUF no está lleno

Transmisor (Modo I²C)

1 = Transmisión en progreso, SSPBUF está lleno (no incluye ACK y bit de Stop)

0 = Transmisión completa SSPBUF está vacío (no incluye ACK y bit de Stop)

Registro SSPCON (14h)

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|-------------|-------|-------|-----|-------|-------|-------|-------|
| WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit7 | | | | | | | bit 0 |

WCOL : Bit de detección de colisión

1 = El registro SSPBUF se ha escrito cuando hay una transmisión en proceso (se debe borrar por software)

0 = No hay colisión

Registro SSPCON (14h)

| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
|-------|--------------|-------|-----|-------|-------|-------|-------|
| WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit7 | | | | | | | bit 0 |

SSPOV: bit de detección de desbordamiento en recepción

En modo SPI

1 = Se recibe un nuevo byte cuando el registro SSPBUF aún mantiene los datos anteriores . En caso de desbordamiento, los datos del SSPSR serán falsos , solo puede ocurrir en modo esclavo. Se deberá leer SSPBUF, aún si solo se envían datos, para anular el rebose.

En modo Master el bit no se pone a "1" ya que en cada transmisión o recepción es inicializado por la escritura del registro SSPBU.

0 = No hay desbordamiento

En modo I2C

1 = Se recibe un nuevo byte cuando el registro SSPBUF aún mantiene los datos anteriores . En modo transmisión no tiene importancia, en cualquier caso debe borrarse por software

0 = No hay desbordamiento

Registro SSPCON (14h)

Pa

| | | | | | | | |
|-------|-------|-------|-----|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit7 | | | | | | | bit 0 |

SSPEN : Bit de habilitación del módulo SSP (synchronous Serial Port)

En modo SPI

1 = Habilita puerto serie y configura SCK, SDO y SDI como patillas del puerto serie.

0 = Deshabilita el puerto serie y configura estas patillas como puerto de Entrada/Salida

En modo I2C

1 = Habilita puerto serie y configura SDA y SDL como patillas del puerto serie.

0 = Deshabilita el puerto serie y configura estas patillas como puerto de Entrada/Salida

En ambos casos, se deberán configurar como entradas o salidas adecuadamente.

Registro SSPCON (14h)

Pa

| | | | | | | | |
|-------|-------|-------|-----|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit7 | | | | | | | bit 0 |

CKP : Bit de selección de polaridad del reloj

EN Modo SPI

1 = Reloj a nivel alto. Transmisión en flanco de bajada, en recepción de subida.

0 = Reloj en nivel bajo. Transmisión en flancos de subida por flancos de bajada.

EN Modo I2C Esclavo (en este modo no se usa como master)

Control de desbloqueo de SCR

1 = Habilita reloj

0 = Mantiene el reloj a nivel bajo

Registro SSPCON (14h)

| | | | | | | | |
|-------|-------|-------|-----|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 |
| WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 |
| bit7 | | | | | | | bit 0 |

SSPM0:SSPM0 Bit de selección del modo del SSP (Synchronous Serial Port)

- 0000: Modo Master del SPI, reloj = $Fosc/4$
- 0001: Modo Master del SPI, reloj = $Fosc/16$
- 0010: Modo Master del SPI, reloj = $Fosc/32$
- 0011: Modo Master del SPI, reloj = TMR2 salida/2
- 0100: Modo Master del SPI, reloj = patilla SCK. Deshabilitada patilla #SS.
- 0101: Modo Master del SPI, reloj = patilla SCK. Deshabilitada patilla #SS, se puede emplear I/O
- 0110: Modo Esclavo I2C, dirección de 7 bits
- 0111: Modo Esclavo I2C, dirección de 10 bits
- 1000: Modo Master I2C, Reloj = $Fosc/(4x(SSPADD+1))$.**
- 1011: Modo Esclavo I2C con soporte
- 1110: Modo Esclavo I2C, dirección de 7 bits con interrupciones Start y Stop habilitados
- 1111: Modo Esclavo I2C, dirección de 10 bits con interrupciones Start y Stop habilitadas

Registro SSPCON2 (91h)

| | | | | | | | |
|-------|----------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| GCEN | ASKSTART | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN |
| bit7 | | | | | | | bit 0 |

GCEN: Bit de HABILITACIÓN GENERAL

- 1: Habilita Interrupción general (Solo en modo esclavo I²C)**
- 0: Deshabilita dirección de llamada general**

Registro SSPCON2 (91h)

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|-------|-----------------|-------|-------|-------|-------|-------|-------|
| GCEN | ACKSTART | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN |
| bit7 | | | | | | | bit 0 |

ACKSTART: Bit de estado de reconocimiento (Solo en masters I²C)

Valor de Transmisión

1: Reconocimiento del esclavo no recibido

0: Reconocimiento del esclavo recibido

Valor de Recepción

1: No reconocimiento

0: Reconocimiento

Registro SSPCON2 (91h)

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|-------|----------|--------------|-------|-------|-------|-------|-------|
| GCEN | ACKSTART | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN |
| bit7 | | | | | | | bit 0 |

ACKDT: Bit de reconocimiento (solo en modo I²C)

Valor de Recepción

Valor transmitido cuando el usuario indica una secuencia de reconocimiento y final de una recepción

1: No reconocimiento (NACK)

0: Reconocimiento (ACK)

```
I2C_EnviaACK
bsf STATUS,RP0      ; El registro SSPCON2 está en el banco 1
bcf STATUS,RP1      ; nos situamos en ese banco primero
bcf SSPCON2,ACKDT   ; Cargamos 0 en ACKDT para sacarlo luego
bsf SSPCON2,ACKEN   ; Iniciamos la secuencia con ACKEN=1
S_ACK  btfsc SSPCON2,ACKEN ; Cuando se haya dado la condición de ACK
        ; se pondrá a 0
        goto S_ACK        ; Si no, seguimos esperando
bcf STATUS,RP0      ; Volvemos al banco 0
return              ; Salimos del subprograma
```

```
I2C_EnviaNoACK
bsf STATUS,RP0      ; El registro SSPCON2 está en el banco 1
bcf STATUS,RP1      ; nos situamos en ese banco primero
bsf SSPCON2,ACKDT   ; Cargamos 1 en ACKDT para sacarlo luego
bsf SSPCON2,ACKEN   ; Iniciamos la secuencia con ACKEN=1
S_NOACK btfsc SSPCON2,ACKEN ; Cuando se haya dado la condición de NOACK
        ; se pondrá a 0
        goto S_NOACK     ; Si no, seguimos esperando
bcf STATUS,RP0      ; Volvemos al banco 0
return              ; Salimos del subprograma
```


Registro SSPCON2 (91h)

| | | | | | | | |
|-------|----------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| GCEN | ACKSTART | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN |
| bit7 | | | | | | | bit 0 |

ACKEN: Bit de habilitación de secuencia de reconocimiento y final de una recepción (solo en modo I²C)

Modo de Recepción

- 1: Indica una secuencia de reconocimiento en las patillas SDA y SCL y transmite bit ACKDT. Automáticamente borrado por Hw.
- 0: Secuencia de reconocimiento en reposo.

Registro SSPCON2 (91h)

| | | | | | | | |
|-------|----------|-------|-------|-------|-------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| GCEN | ACKSTART | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN |
| bit7 | | | | | | | bit 0 |

RCEN: Bit de habilitación de recepción (solo en modo I²C)

- 1: Habilita recepción en modo I²C, se pone a cero cuando se finaliza.
- 0: Recepción en reposo

```

I2C_LeerByte
    bsf STATUS,RP0      ; Nos situamos en el banco 1 que es donde
    bcf STATUS,RP1      ; está el registro SSPCON2
    bsf SSPCON2,RCEN    ; Iniciamos la lectura con el bit RCEN, se pone a cero cuando se realiza
    nop

LLEGANDO
    btfsc SSPCON2,RCEN  ; Verificamos que se ha completado la recepción
    goto  LLEGANDO      ; si no seguiremos esperando mientras tanto
    bcf STATUS,RP0      ; Volvemos al banco 0
    movf SSPBUF,W        ; Recojo el dato de SSPBUF y dejo en W
    return
  
```

Registro SSPCON2 (91h)

| | | | | | | | |
|-------|----------|-------|-------|-------|------------|-------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| GCEN | ACKSTART | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN |
| bit7 | | | | | | | bit 0 |

PEN: Bit de habilitación de condición de Stop recepción (solo en modo I2C)

1: Indica una condición de Stop en SDA y SCL. Automáticamente borrado por Hw.

0: Condición de Stop en reposo

```

I2C_EnviaStop
    bsf STATUS,RP0      ; El registro SSPCON2 está en el banco 1
    bcf STATUS,RP1
    bsf SSPCON2,PEN    ; Activamos la generación de STOP, se pone a "0" cuando se genera
    nop
S_STOP
    btfsc SSPCON2,PEN  ; Cuando se haya dado la condición de STOP
    ; se pondrá a 0
    goto S_STOP        ; Si no, seguimos esperando
    bcf STATUS,RP0     ; Volvemos al banco 0
    return              ; Salimos del subprograma
  
```

Registro SSPCON2 (91h)

| | | | | | | | |
|-------|----------|-------|-------|-------|-------|-------------|-------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| GCEN | ACKSTART | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN |
| bit7 | | | | | | | bit 0 |

RSEN: Bit de habilitación repetir condición Start (solo en modo I2C)

1: Indica repetición de la condición de START en SDA y SCL. Automáticamente borrado por HW.

0: Condición de recepción de START en reposo

```

I2C_Envia_RESTART
    bsf STATUS,RP0      ; El registro SSPCON2 está en el banco 1
    bcf STATUS,RP1
    bsf SSPCON2,RSEN    ; Activamos la generación de RESTART, se pone a "0" cuando la genera
    nop
S_RESTART
    btfsc SSPCON2,RSEN  ; Cuando se haya dado la condición de RESTART
    ; se pondrá a 0
    goto S_RESTART     ; Si no, seguimos esperando
    bcf STATUS,RP0     ; Volvemos al banco 0
    return              ; Salimos del subprograma
  
```

Registro SSPCON2 (91h)

| | | | | | | | |
|-------|----------|-------|-------|-------|-------|-------|------------|
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| GCEN | ACKSTART | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN |
| bit7 | | | | | | | bit 0 |

SEN: Bit de habilitación de condición de START recepción (solo en modo I²C)

1: Indica una condición de START en SDA y SCL. Automáticamente borrado por Hw.

0: Condición de START en reposo

```

;I2C_EnviaStart
; Este subprograma genera un bit de START y lo comprueba,
; espera hasta que se produzca y aparezca tal condición en el bus
I2C_EnviaStart
    bsf    STATUS,RP0
    bsf    SSPCON2,SEN    ;Inicializa de START en SDA y SCL, se pone a cero cuando la genera
Espera_Start
    btfsc  SSPCON2,SEN    ;Se ha producido la condición de estar?
    goto  Espera_Start   ;$-1
    bcf    STATUS,RP0
    return
    
```

Registros y bits asociados al Modo I²C

| Address | Name | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Value on: POR, BOR | Value on: MCLR, WDT |
|----------------------|---------|--|---------|-------|-------|-------|--------|--------|--------|--------------------|---------------------|
| 0Bh, 8Bh, 10Bh, 18Bh | INTCON | GIE | PEIE | T0IE | INTE | RBIE | T0IF | INTF | RBIF | 0000 000x | 0000 000u |
| 0Ch | PIR1 | PSPIF ⁽¹⁾ | ADIF | RCIF | TXIF | SSPIF | CCP1IF | TMR2IF | TMR1IF | 0000 0000 | 0000 0000 |
| 8Ch | PIE1 | PSPIE ⁽¹⁾ | ADIE | RCIE | TXIE | SSPIE | CCP1IE | TMR2IE | TMR1IE | 0000 0000 | 0000 0000 |
| 0Dh | PIR2 | — | (2) | — | EEIF | BCLIF | — | — | CCP2IF | -x-0 0--0 | -x-0 0--0 |
| 8Dh | PIE2 | — | (2) | — | EEIE | BCLIE | — | — | CCP2IE | -x-0 0--0 | -x-0 0--0 |
| 13h | SSPBUF | Synchronous Serial Port Receive Buffer/Transmit Register | | | | | | | | xxxx xxxx | uuuu uuuu |
| 14h | SSPCON | WCOL | SSPOV | SSPEN | CKP | SSPM3 | SSPM2 | SSPM1 | SSPM0 | 0000 0000 | 0000 0000 |
| 91h | SSPCON2 | GCEN | ACKSTAT | ACKDT | ACKEN | RCEN | PEN | RSEN | SEN | 0000 0000 | 0000 0000 |
| 93h | SSPADD | I ² C Slave Address/Master Baud Rate Register | | | | | | | | 0000 0000 | 0000 0000 |
| 94h | SSPSTAT | SMP | CKE | D/Ā | P | S | R/W | UA | BF | 0000 0000 | 0000 0000 |

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the SSP in I²C mode.
Note 1: These bits are reserved on PIC16F873/876 devices; always maintain these bits clear.
Note 2: These bits are reserved on these devices; always maintain these bits clear.