

Comunicación Serie

Alejandro Furfaro

26 de septiembre de 2023

- 1 **Introducción**
 - Motivación
 - Primeros estándares
- 2 **SPI**
 - Introducción
 - Funcionamiento
 - Implementación SPI en el SoC Sitara AM3358 (BBB)
- 3 **I2C**
 - Introducción
 - Funcionamiento
 - Implementación I2C en el SoC Sitara AM3358 (BBB)
- 4 **Otros protocolos serie**
 - Controller Area Network (CAN)
 - 1-Wire

Temario

- 1 **Introducción**
 - Motivación
 - Primeros estándares
- 2 SPI
 - Introducción
 - Funcionamiento
 - Implementación SPI en el SoC Sitara AM3358 (BBB)
- 3 I2C
 - Introducción
 - Funcionamiento
 - Implementación I2C en el SoC Sitara AM3358 (BBB)
- 4 **Otros protocolos serie**
 - Controller Area Network (CAN)
 - 1-Wire

Interfaces de un computador

- Desde el inicio las interfaces para periféricos se diseñaron en base a dos tipos diferentes.

Interfaces de un computador

- Desde el inicio las interfaces para periféricos se diseñaron en base a dos tipos diferentes.
- **Interfaz Paralelo**: La información se transmite por un conjunto de líneas enviándose varios bits en paralelo

Interfaces de un computador

- Desde el inicio las interfaces para periféricos se diseñaron en base a dos tipos diferentes.
- **Interfaz Paralelo**: La información se transmite por un conjunto de líneas enviándose varios bits en paralelo
- **Interfaz Serie**: Los datos se transfieren a razón de un bit a la vez por una única línea (o dos si la comunicación es bidireccional y Full duplex).

Interfaces de un computador

- Desde el inicio las interfaces para periféricos se diseñaron en base a dos tipos diferentes.
- **Interfaz Paralelo**: La información se transmite por un conjunto de líneas enviándose varios bits en paralelo
- **Interfaz Serie**: Los datos se transfieren a razón de un bit a la vez por una única línea (o dos si la comunicación es bidireccional y Full duplex).

A pesar de que se resigna velocidad de transmisión frente a la interfaz paralelo, las interfaces serie son siempre preferidas ya que se ahorra terminales, hecho que siempre pesa en la simplificación de los diseños. Por tal motivo se han desarrollado incesantemente nuevos protocolos serie, marcando una clara evolución tecnológica en este tipo de interfaces

Interfaces serie de baja velocidad

Interfaces serie de baja velocidad

- Todas las comunicaciones (serie o paralelo) requieren técnicas de sincronismo para establecer la misma frecuencia de intercambio de datos en ambos extremos de la comunicación.

Interfaces serie de baja velocidad

- Todas las comunicaciones (serie o paralelo) requieren técnicas de sincronismo para establecer la misma frecuencia de intercambio de datos en ambos extremos de la comunicación.
- Una posibilidad es incluir una línea adicional para enviar por allí el reloj de transmisión.

Interfaces serie de baja velocidad

- Todas las comunicaciones (serie o paralelo) requieren técnicas de sincronismo para establecer la misma frecuencia de intercambio de datos en ambos extremos de la comunicación.
- Una posibilidad es incluir una línea adicional para enviar por allí el reloj de transmisión.
- Esto nos obliga a asegurar la integridad del pulso rectangular de reloj en destino. Objetivo contrapuesto con la distancia a la que se encuentra. Se usa comunicación sincrónica en general en distancias cortas

Interfaces serie de baja velocidad

- Todas las comunicaciones (serie o paralelo) requieren técnicas de sincronismo para establecer la misma frecuencia de intercambio de datos en ambos extremos de la comunicación.
- Una posibilidad es incluir una línea adicional para enviar por allí el reloj de transmisión.
- Esto nos obliga a asegurar la integridad del pulso rectangular de reloj en destino. Objetivo contrapuesto con la distancia a la que se encuentra. Se usa comunicación sincrónica en general en distancias cortas
- La otra posibilidad es manejar el sincronismo dentro de la trama de bits que se transmite. A pesar de que el sincronismo existe, ésta variante de comunicación serie se denomina asincrónica, mas que nada por la ausencia de una línea de reloj física.

Interfaces serie de baja velocidad

“The great thing about standards is there are so many to choose from.”

Interfaces serie de baja velocidad

Las principales interfaces son las que listamos a continuación:

Interfaces serie de baja velocidad

Las principales interfaces son las que listamos a continuación:

Interfaces serie de baja velocidad

Las principales interfaces son las que listamos a continuación:

- RS-232

Interfaces serie de baja velocidad

Las principales interfaces son las que listamos a continuación:

- RS-232
- RS-422

Interfaces serie de baja velocidad

Las principales interfaces son las que listamos a continuación:

- RS-232
- RS-422
- RS-485

Interfaces serie de baja velocidad

Las principales interfaces son las que listamos a continuación:

- RS-232
- RS-422
- RS-485
- I²C

Interfaces serie de baja velocidad

Las principales interfaces son las que listamos a continuación:

- RS-232
- RS-422
- RS-485
- I²C
- SPI

Interfaces serie de baja velocidad

Las principales interfaces son las que listamos a continuación:

- RS-232
- RS-422
- RS-485
- I²C
- SPI
- CAN

Interfaces serie de baja velocidad

Las principales interfaces son las que listamos a continuación:

- RS-232
- RS-422
- RS-485
- I²C
- SPI
- CAN
- 1-Wire

Temario

1 Introducción

- Motivación
- **Primeros estándares**

2 SPI

- Introducción
- Funcionamiento
- Implementación SPI en el SoC Sitara AM3358 (BBB)

3 I2C

- Introducción
- Funcionamiento
- Implementación I2C en el SoC Sitara AM3358 (BBB)

4 Otros protocolos serie

- Controller Area Network (CAN)
- 1-Wire

RS-232

- Pensada para velocidades de transmisión máxima menores de 20 kbit s, llegó a dar 115 200 bit/second. Con drivers (típicamente MAX3225E, o similar), alcanzan velocidades de 1 Mbit s

RS-232

- Pensada para velocidades de transmisión máxima menores de 20 kbit s, llegó a dar 115 200 bit/second. Con drivers (típicamente MAX3225E, o similar), alcanzan velocidades de 1 Mbit s
- Se implementa en las archiconocidas UART (Unidad Asíncrona Receptora Transmisora).

RS-232

- Pensada para velocidades de transmisión máxima menores de 20 kbit s, llegó a dar 115 200 bit/second. Con drivers (típicamente MAX3225E, o similar), alcanzan velocidades de 1 Mbit s
- Se implementa en las archiconocidas UART (Unidad Asíncrona Receptora Transmisora).
- Si bien es sumamente antigua sigue existiendo en cualquier sistema embebido. De hecho el SoC SITARA 3358 dispone de seis UARTs compatibles con un clásico 16C750.

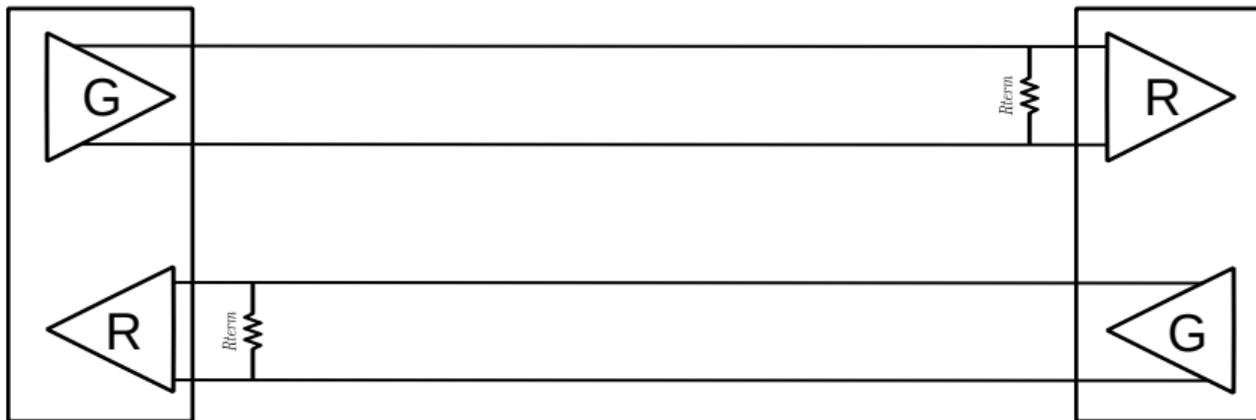
RS-232

- Pensada para velocidades de transmisión máxima menores de 20 kbit/s, llegó a dar 115 200 bit/second. Con drivers (típicamente MAX3225E, o similar), alcanzan velocidades de 1 Mbit/s
- Se implementa en las archiconocidas UART (Unidad Asíncrona Receptora Transmisora).
- Si bien es sumamente antigua sigue existiendo en cualquier sistema embebido. De hecho el SoC SITARA 3358 dispone de seis UARTs compatibles con un clásico 16C750.
- Indispensable para conectar una consola de inicio a un sistema Linux (Heredado de UNIX). Linux asume que las UARTs tiene conectadas terminales “bobas”.

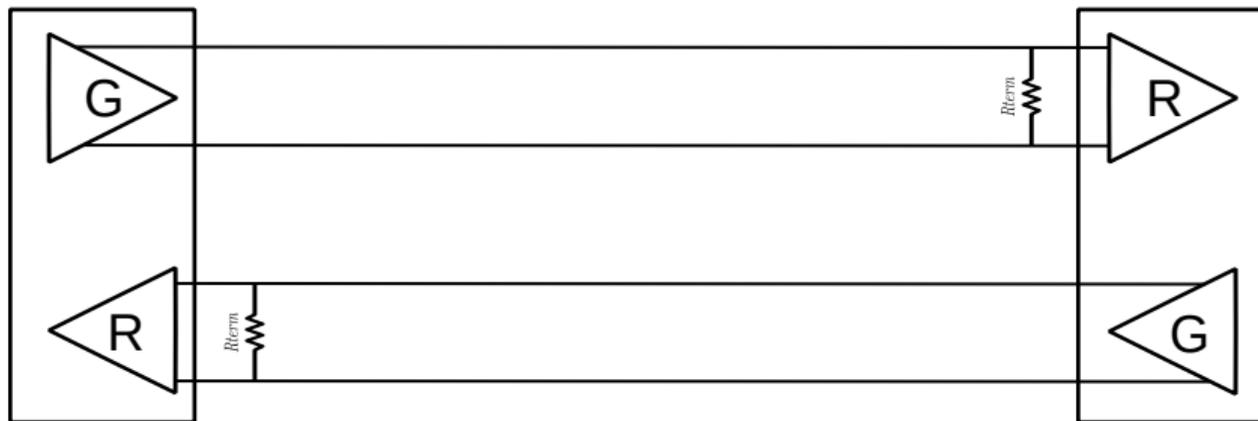
RS-232

- Pensada para velocidades de transmisión máxima menores de 20 kbit s, llegó a dar 115 200 bit/second. Con drivers (típicamente MAX3225E, o similar), alcanzan velocidades de 1 Mbit s
- Se implementa en las archiconocidas UART (Unidad Asíncrona Receptora Transmisora).
- Si bien es sumamente antigua sigue existiendo en cualquier sistema embebido. De hecho el SoC SITARA 3358 dispone de seis UARTs compatibles con un clásico 16C750.
- Indispensable para conectar una consola de inicio a un sistema Linux (Heredado de UNIX). Linux asume que las UARTs tiene conectadas terminales “bobas”.
- Principales limitaciones: relativamente baja velocidad, limitada cantidad de bits que puede transmitir (9 a lo sumo), imposibilidad de múltiples master o múltiples slaves (es solo punto a punto), y escasa tolerancia a variabilidad del baud rate entre extremos.

RS-422

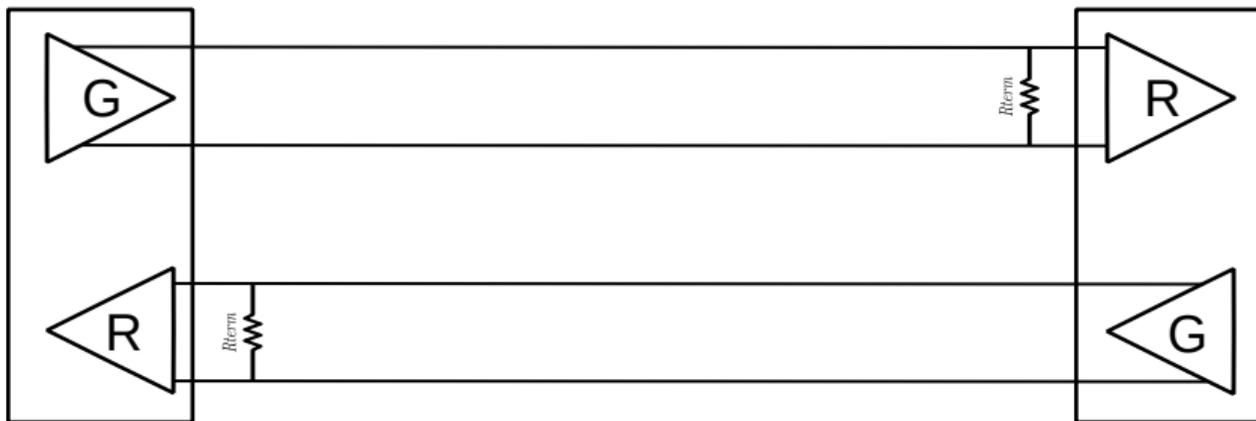


RS-422



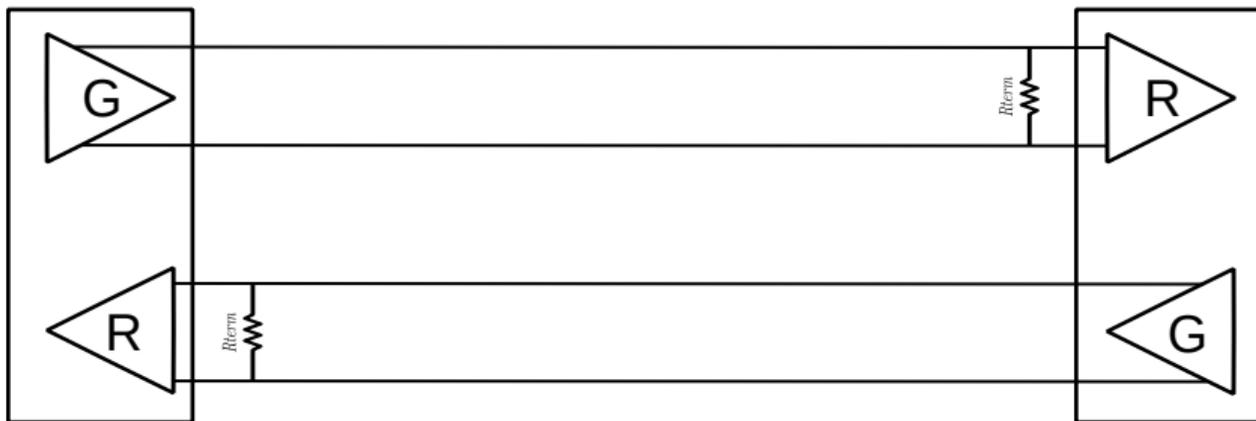
- Posterior a RS-232, busca aumentar la velocidad alrededor de 10 Mbit/s, para distancias máximas de alrededor de 1200 m.

RS-422



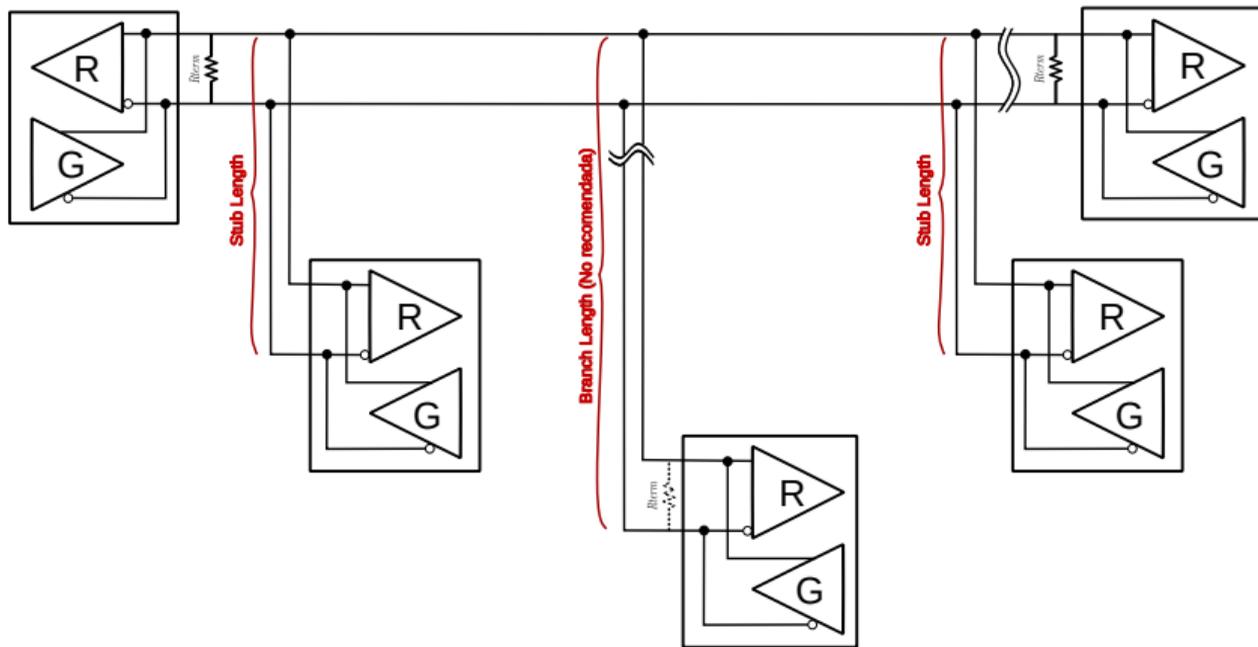
- Posterior a RS-232, busca aumentar la velocidad alrededor de 10 Mbit/s, para distancias máximas de alrededor de 1200 m.
- Utiliza señales diferenciales en ambos sentidos para lograr mayor inmunidad al ruido en la transmisión, y alcanzar mayores distancias y velocidad libre de errores.

RS-422

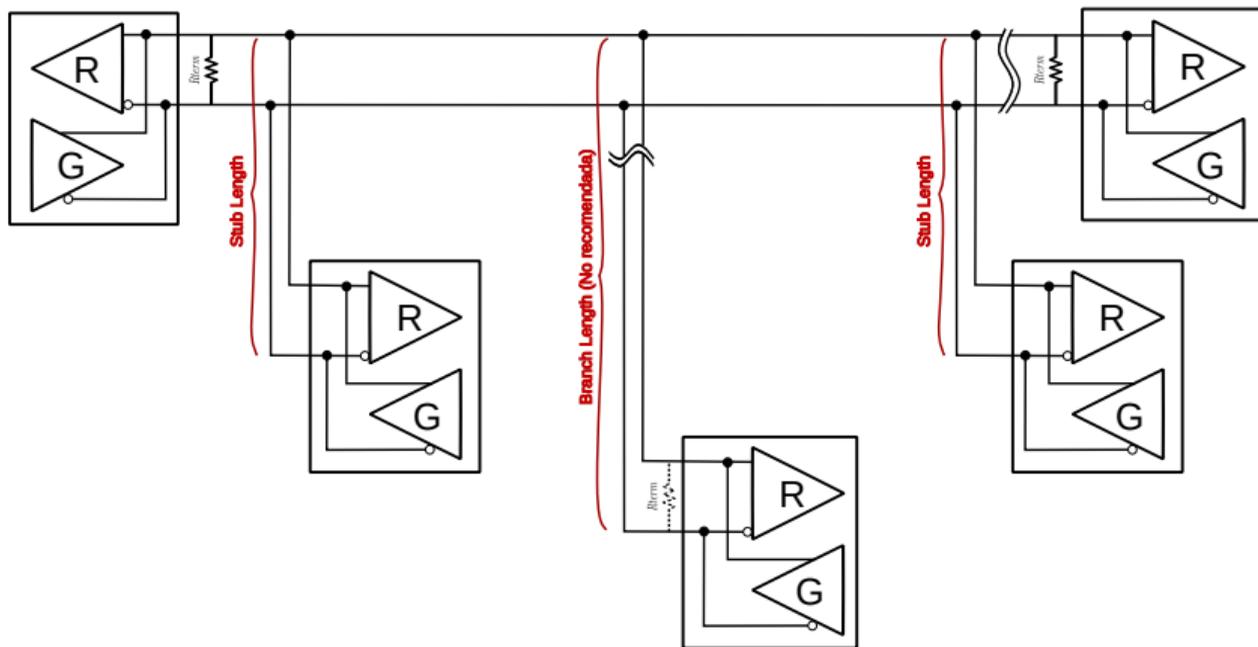


- Posterior a RS-232, busca aumentar la velocidad alrededor de 10 Mbit/s, para distancias máximas de alrededor de 1200 m.
- Utiliza señales diferenciales en ambos sentidos para lograr mayor inmunidad al ruido en la transmisión, y alcanzar mayores distancias y velocidad libre de errores.
- Terminó teniendo como principal aplicación la de mecanismo de transporte de extensores RS-232.

RS-485

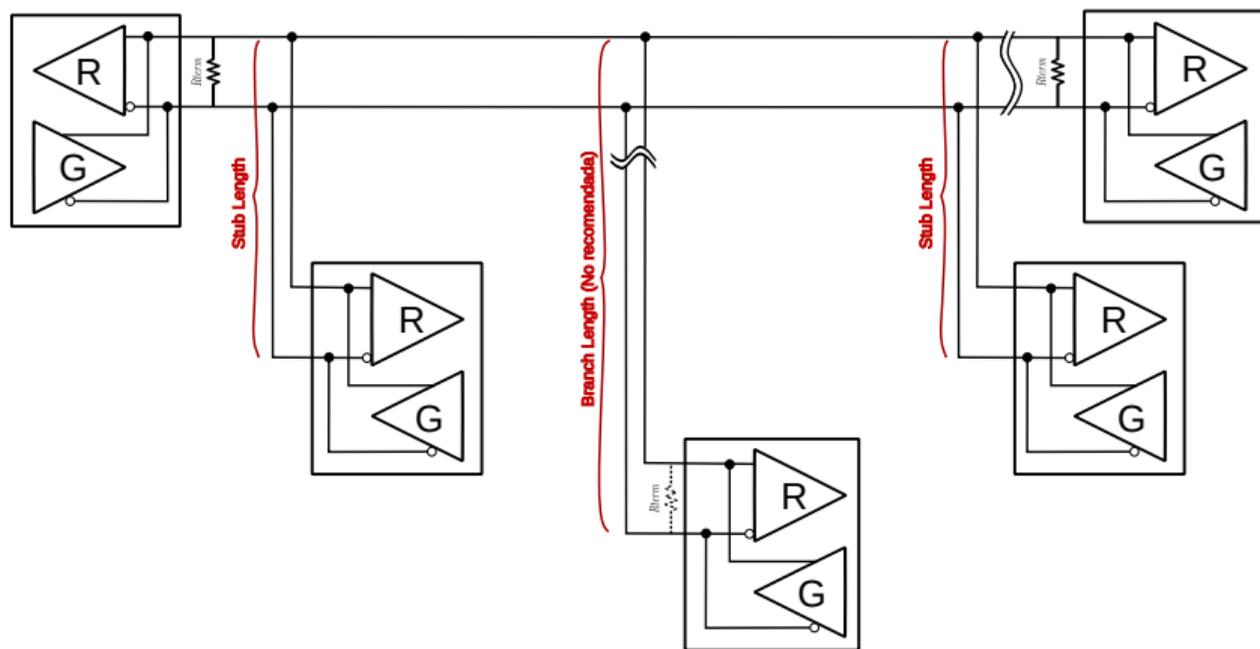


RS-485



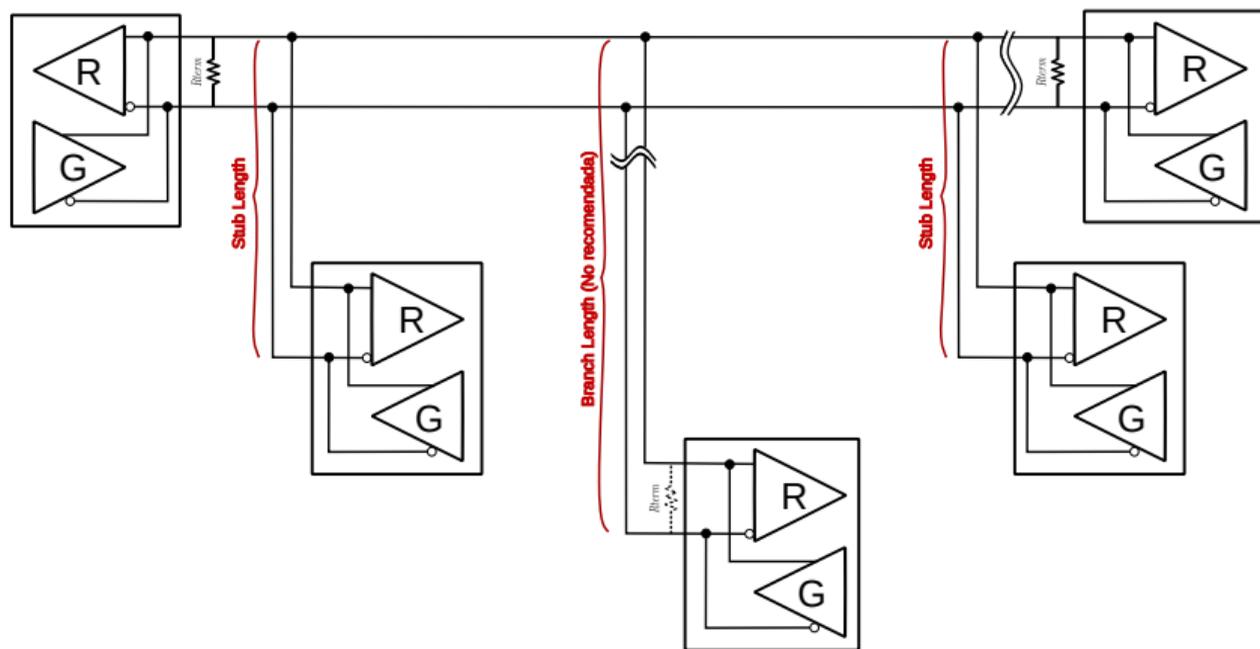
- Señal diferencial como la de RS-422. Transmisión multipunto. Hasta 10 Mbit s.

RS-485



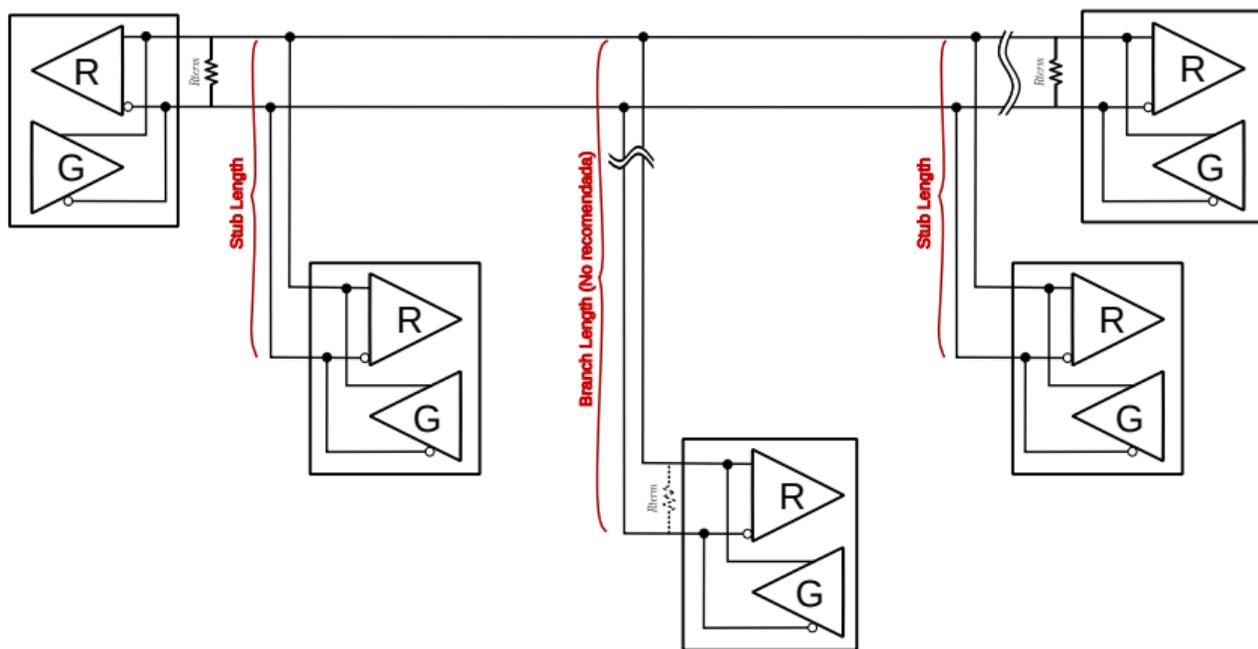
- Señal diferencial como la de RS-422. Trasmisión multipunto. Hasta 10 Mbit s.
- Los transmisores tiene salida Three-state. Pueden desactivarse permitiendo tratar al par trenzado como un bus multipunto.

RS-485



- Los nodos no se conectan en una suerte de red estrella, debido a la dificultad de obtener impedancias de terminación que eviten reflexiones de señal.

RS-485



- Los nodos no se conectan en una suerte de red estrella, debido a la dificultad de obtener impedancias de terminación que eviten reflexiones de señal.
- Los nodos se conectan a un bus serie con conexiones cortas.

RS-485

RS-485

- La resistencia de terminación es la resistencia característica del par trenzado de Cu, alrededor de 120Ω .

RS-485

- La resistencia de terminación es la resistencia característica del par trenzado de Cu, alrededor de $120\ \Omega$.
- Necesita además una resistencia de polarización en cada hilo para asegurar los niveles de referencia de tensión adecuados que permitan a la señal diferencial rechazar ruido de manera eficaz.

RS-485

- La resistencia de terminación es la resistencia característica del par trenzado de Cu, alrededor de 120Ω .
- Necesita además una resistencia de polarización en cada hilo para asegurar los niveles de referencia de tensión adecuados que permitan a la señal diferencial rechazar ruido de manera eficaz.
- Al igual que RS-422 con un par de líneas diferenciales es capaz de soportar transmisiones full duplex.

RS-485

- La resistencia de terminación es la resistencia característica del par trenzado de Cu, alrededor de 120Ω .
- Necesita además una resistencia de polarización en cada hilo para asegurar los niveles de referencia de tensión adecuados que permitan a la señal diferencial rechazar ruido de manera eficaz.
- Al igual que RS-422 con un par de líneas diferenciales es capaz de soportar transmisiones full duplex.
- Sin embargo, para la mayoría de las aplicaciones al comportarse como un bus masterizado, ésto no es necesario.

RS-485

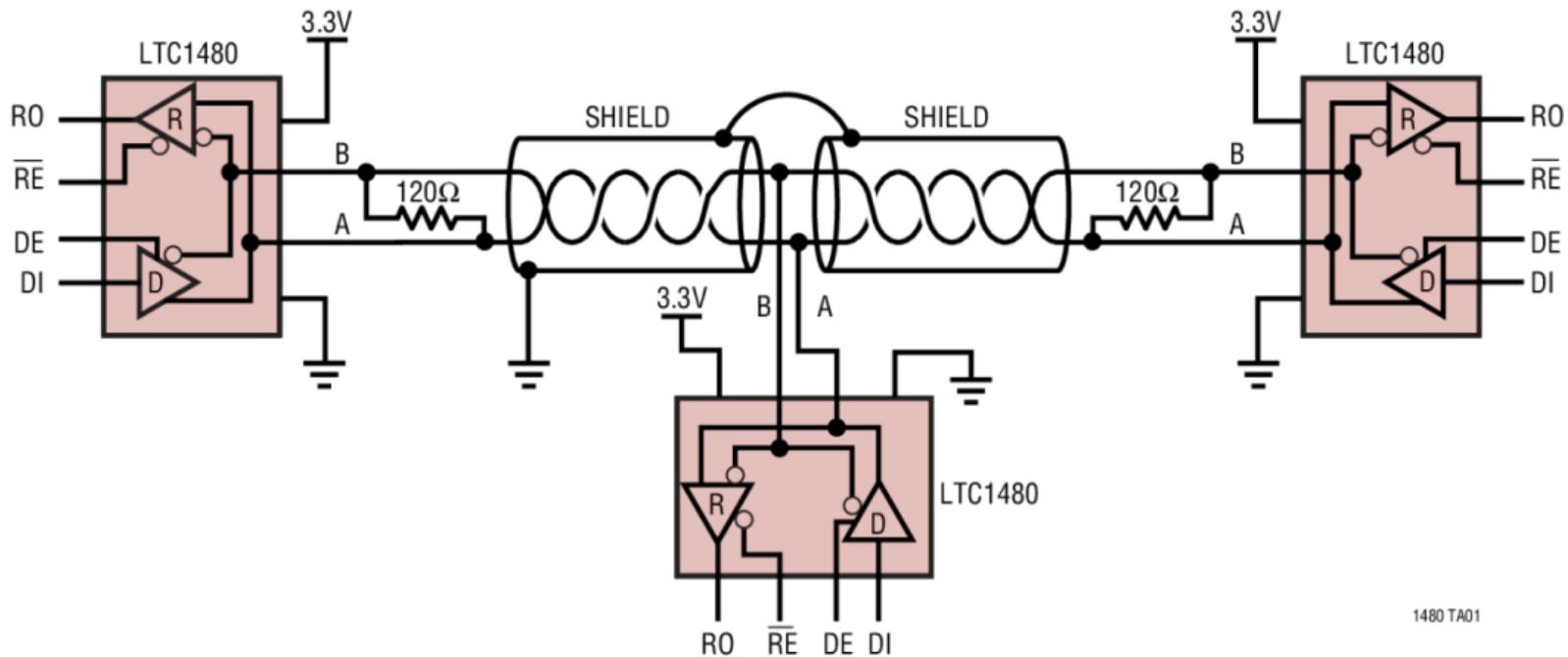
- La resistencia de terminación es la resistencia característica del par trenzado de Cu, alrededor de 120Ω .
- Necesita además una resistencia de polarización en cada hilo para asegurar los niveles de referencia de tensión adecuados que permitan a la señal diferencial rechazar ruido de manera eficaz.
- Al igual que RS-422 con un par de líneas diferenciales es capaz de soportar transmisiones full duplex.
- Sin embargo, para la mayoría de las aplicaciones al comportarse como un bus masterizado, ésto no es necesario.
- Actualmente tiene un importante rango de aplicaciones en sistemas de automatización industrial.

RS-485

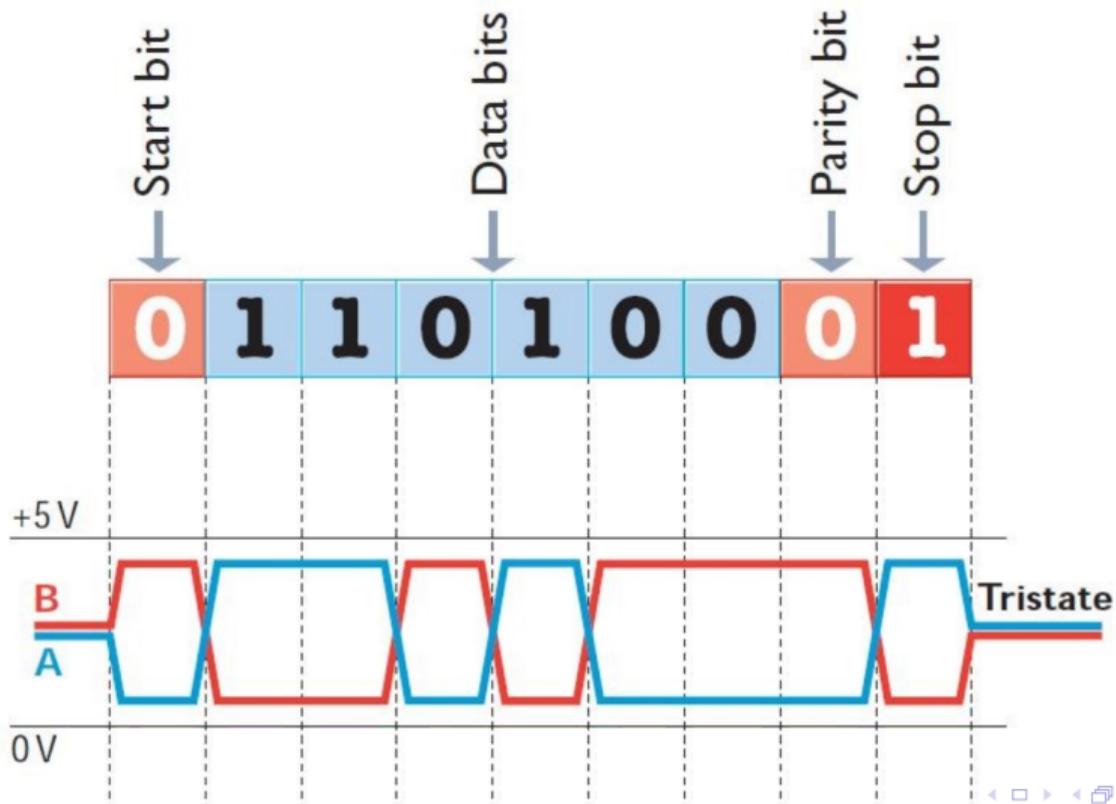
- La resistencia de terminación es la resistencia característica del par trenzado de Cu, alrededor de 120Ω .
- Necesita además una resistencia de polarización en cada hilo para asegurar los niveles de referencia de tensión adecuados que permitan a la señal diferencial rechazar ruido de manera eficaz.
- Al igual que RS-422 con un par de líneas diferenciales es capaz de soportar transmisiones full duplex.
- Sin embargo, para la mayoría de las aplicaciones al comportarse como un bus masterizado, ésto no es necesario.
- Actualmente tiene un importante rango de aplicaciones en sistemas de automatización industrial.
- Es la capa física de numerosos protocolos de automatización, normalizados y hasta patentados, incluso Modbus y Profibus, que no requieren altas velocidades pero sí en cambio requieren alta inmunidad al ruido, empleando mínimo hardware.

Red RS-485 de 3.3 V

3.3V RS485 Network



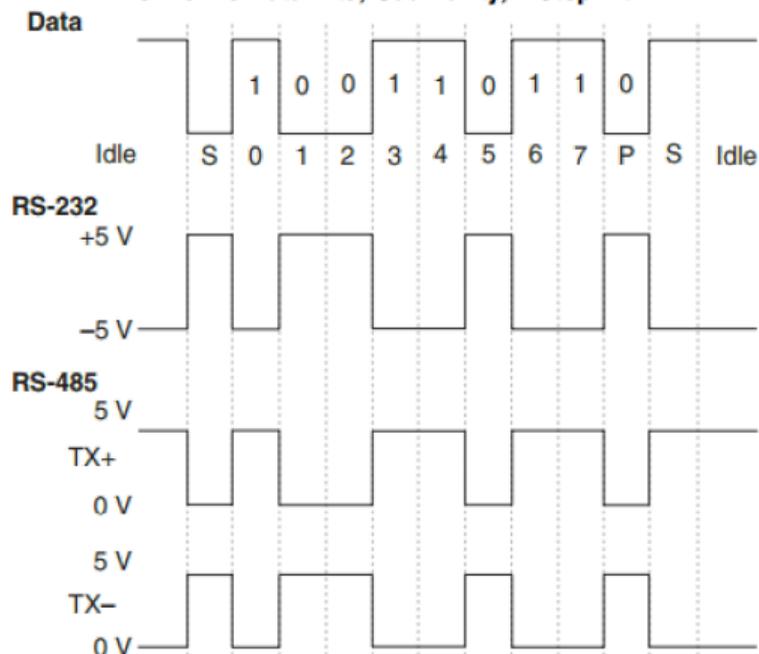
Frame RS-485



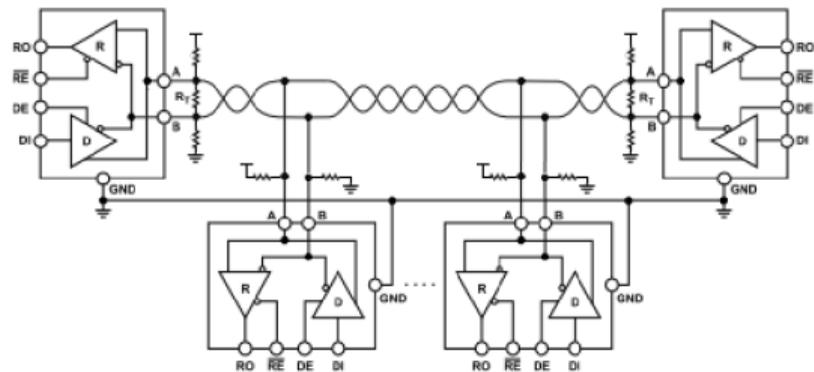
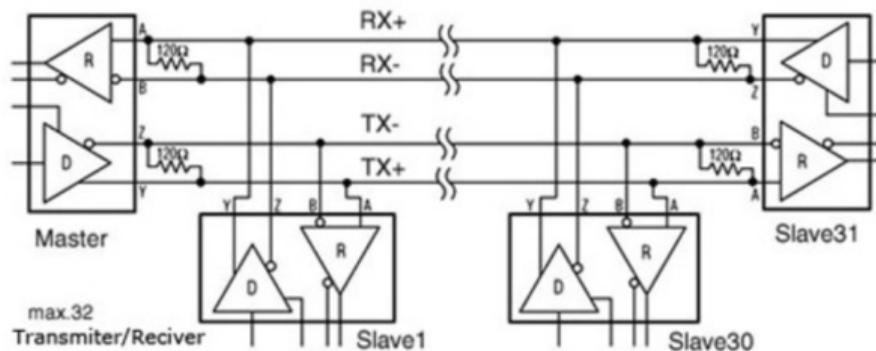
Comparación RS-485 vs. RS232/422

UART Data Frame Example

0xD9—8 Data Bits, Odd Parity, 1 Stop Bit



Voltages are for illustration only. Actual voltage levels may vary.



Temario

- 1 Introducción
 - Motivación
 - Primeros estándares
- 2 **SPI**
 - **Introducción**
 - Funcionamiento
 - Implementación SPI en el SoC Sitara AM3358 (BBB)
- 3 I2C
 - Introducción
 - Funcionamiento
 - Implementación I2C en el SoC Sitara AM3358 (BBB)
- 4 Otros protocolos serie
 - Controller Area Network (CAN)
 - 1-Wire

SPI. Características principales

- Es uno de los protocolos mas utilizados para la interconexión en sistemas embebidos entre la CPU y sus dispositivos periféricos.

SPI. Características principales

- Es uno de los protocolos mas utilizados para la interconexión en sistemas embebidos entre la CPU y sus dispositivos periféricos.
- Aplicaciones típicas: Lectores RFID, módulos SD, e Interfaces WiFi 2,4 GHz.

SPI. Características principales

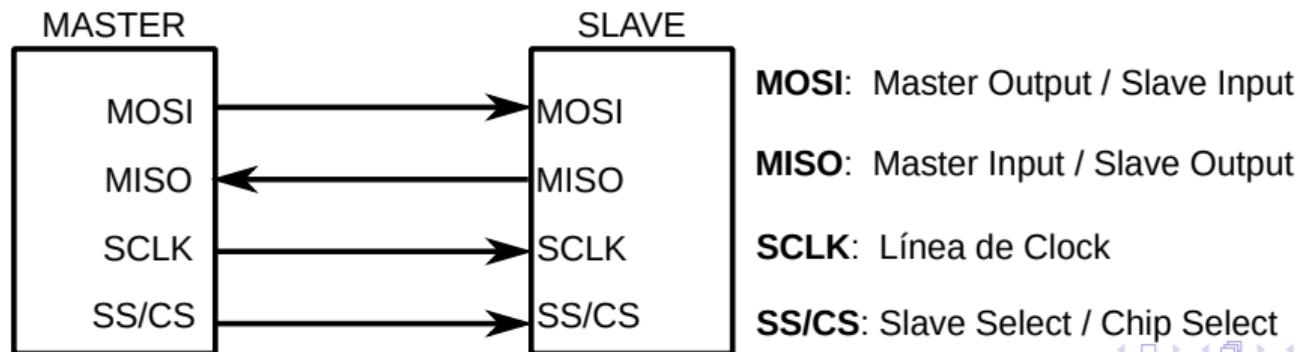
- Es uno de los protocolos mas utilizados para la interconexión en sistemas embebidos entre la CPU y sus dispositivos periféricos.
- Aplicaciones típicas: Lectores RFID, módulos SD, e Interfaces WiFi 2,4 GHz.
- Principal fortaleza: permite transmitir un stream de cualquier cantidad de bits sin interrupción. Las UART como es sabido, e I2C como veremos envían paquetes de bits con bits de start y de stop para delimitar el paquete transmitido.

SPI. Características principales

- Es uno de los protocolos mas utilizados para la interconexión en sistemas embebidos entre la CPU y sus dispositivos periféricos.
- Aplicaciones típicas: Lectores RFID, módulos SD, e Interfaces WiFi 2,4 GHz.
- Principal fortaleza: permite transmitir un stream de cualquier cantidad de bits sin interrupción. Las UART como es sabido, e I2C como veremos envían paquetes de bits con bits de start y de stop para delimitar el paquete transmitido.
- Operan en configuración Master (típicamente la CPU) Slave (el dispositivo: Sensor, display, memoria, etc.)

SPI. Características principales

- Es uno de los protocolos mas utilizados para la interconexión en sistemas embebidos entre la CPU y sus dispositivos periféricos.
- Aplicaciones típicas: Lectores RFID, módulos SD, e Interfaces WiFi 2,4 GHz.
- Principal fortaleza: permite transmitir un stream de cualquier cantidad de bits sin interrupción. Las UART como es sabido, e I2C como veremos envían paquetes de bits con bits de start y de stop para delimitar el paquete transmitido.
- Operan en configuración Master (típicamente la CPU) Slave (el dispositivo: Sensor, display, memoria, etc.)



SPI. Características principales

- Cantidad de alambres: 4

SPI. Características principales

- Cantidad de alambres: 4
- Ancho de banda máximo: 10 Mbit s

SPI. Características principales

- Cantidad de alambres: 4
- Ancho de banda máximo: 10 Mbit s
- Comunicación sincrónica (**SCLK**).

SPI. Características principales

- Cantidad de alambres: 4
- Ancho de banda máximo: 10 Mbit s
- Comunicación sincrónica (**SCLK**).
- Cantidad de Dispositivos Master: 1

SPI. Características principales

- Cantidad de alambres: 4
- Ancho de banda máximo: 10 Mbit s
- Comunicación sincrónica (**SCLK**).
- Cantidad de Dispositivos Master: 1
- Cantidad de Slaves: En teoría ilimitada. En la práctica depende de la carga capacitiva de las entradas conectadas al bus que a la larga impedirán la conexión de slaves adicionales. De todos modos es sumamente alta.

Temario

- 1 Introducción
 - Motivación
 - Primeros estándares
- 2 SPI
 - Introducción
 - **Funcionamiento**
 - Implementación SPI en el SoC Sitara AM3358 (BBB)
- 3 I2C
 - Introducción
 - Funcionamiento
 - Implementación I2C en el SoC Sitara AM3358 (BBB)
- 4 Otros protocolos serie
 - Controller Area Network (CAN)
 - 1-Wire

Señal de **CLOCK**

- Las comunicaciones en SPI están siempre configuradas por el Master, de modo que es éste quien configura el clock.

Señal de **CLOCK**

- Las comunicaciones en SPI están siempre configuradas por el Master, de modo que es éste quien configura el clock.
- La transmisión es a razón de un bit por ciclo de clock, de modo que **SCLK**, determina la velocidad de transmisión.

Señal de **CLOCK**

- Las comunicaciones en SPI están siempre configuradas por el Master, de modo que es éste quien configura el clock.
- La transmisión es a razón de un bit por ciclo de clock, de modo que **SCLK**, determina la velocidad de transmisión.
- Tiene dos propiedades: Polaridad y Fase. Ambas determinan cuando se pone un dato en la línea de transmisión y cuando se muestrea la entrada.

Señal de **CLOCK**

- Las comunicaciones en SPI están siempre configuradas por el Master, de modo que es éste quien configura el clock.
- La transmisión es a razón de un bit por ciclo de clock, de modo que **SCLK**, determina la velocidad de transmisión.
- Tiene dos propiedades: Polaridad y Fase. Ambas determinan cuando se pone un dato en la línea de transmisión y cuando se muestrea la entrada.
- El Master configura la polaridad del clock para determinar si se trabaja con lógica positiva (high) o negativa (low). Tiene que ver con el duty cycle del clock. En el de lógica positiva el estado activo es cuando la señal de clock es alta.

Señal de **CLOCK**

- Las comunicaciones en SPI están siempre configuradas por el Master, de modo que es éste quien configura el clock.
- La transmisión es a razón de un bit por ciclo de clock, de modo que **SCLK**, determina la velocidad de transmisión.
- Tiene dos propiedades: Polaridad y Fase. Ambas determinan cuando se pone un dato en la línea de transmisión y cuando se muestrea la entrada.
- El Master configura la polaridad del clock para determinar si se trabaja con lógica positiva (high) o negativa (low). Tiene que ver con el duty cycle del clock. En el de lógica positiva el estado activo es cuando la señal de clock es alta.
- La fase determina si se muestrea en el primer o segundo flanco independientemente de si éste es ascendente o descendente.

Señal de Slave Select

- El Master selecciona con qué Slave quiere hablar configurando la línea CS/SS del esclavo en un nivel de tensión baja.

Señal de Slave Select

- El Master selecciona con qué Slave quiere hablar configurando la línea CS/SS del esclavo en un nivel de tensión baja.
- En el estado inactivo, sin transmisión, la línea SS se mantiene a un nivel de tensión alto.

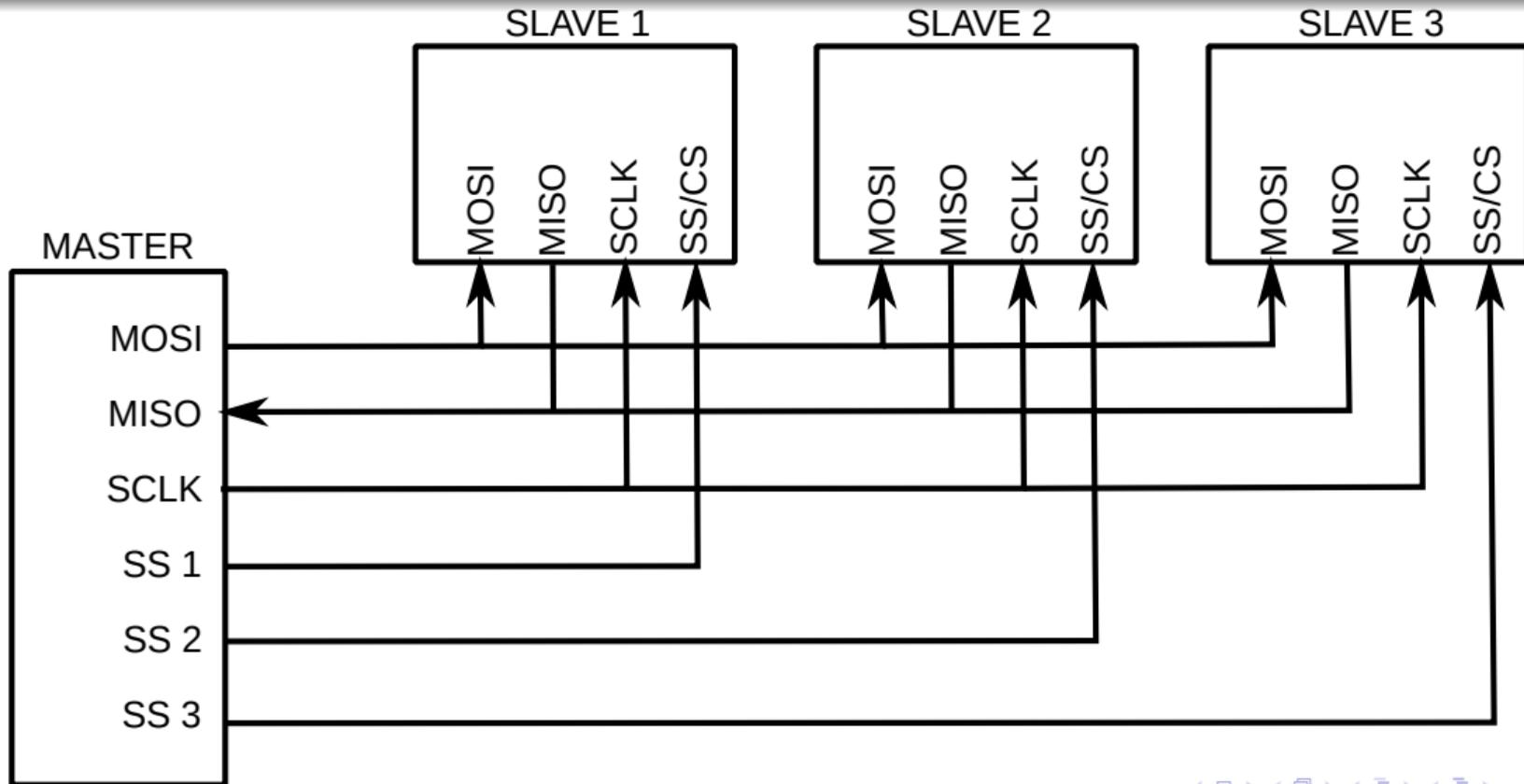
Señal de Slave Select

- El Master selecciona con qué Slave quiere hablar configurando la línea CS/SS del esclavo en un nivel de tensión baja.
- En el estado inactivo, sin transmisión, la línea SS se mantiene a un nivel de tensión alto.
- Es posible que haya varios pines CS/SS disponibles en el Master, lo que permite conectar varios Slaves en paralelo.

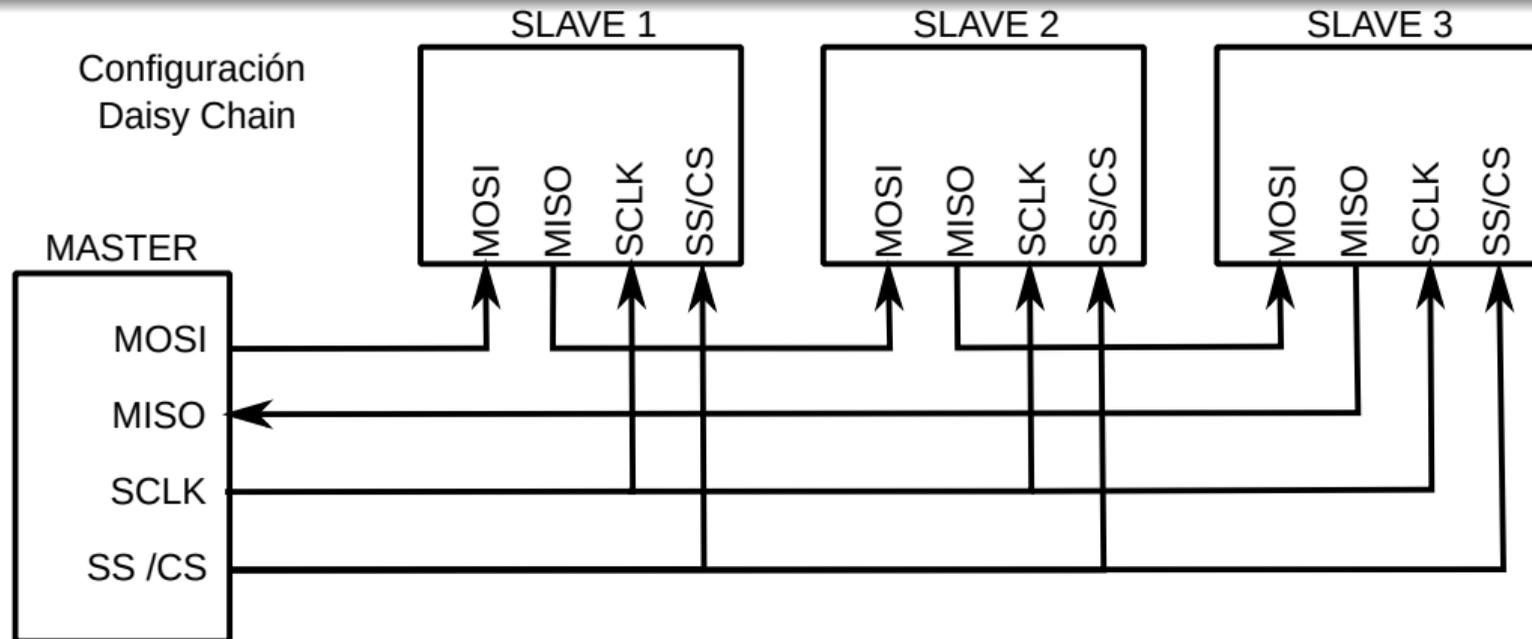
Señal de Slave Select

- El Master selecciona con qué Slave quiere hablar configurando la línea CS/SS del esclavo en un nivel de tensión baja.
- En el estado inactivo, sin transmisión, la línea SS se mantiene a un nivel de tensión alto.
- Es posible que haya varios pines CS/SS disponibles en el Master, lo que permite conectar varios Slaves en paralelo.
- Si solo hay un pin CS/SS, se pueden conectar varios Slaves al Master mediante una conexión encadenada

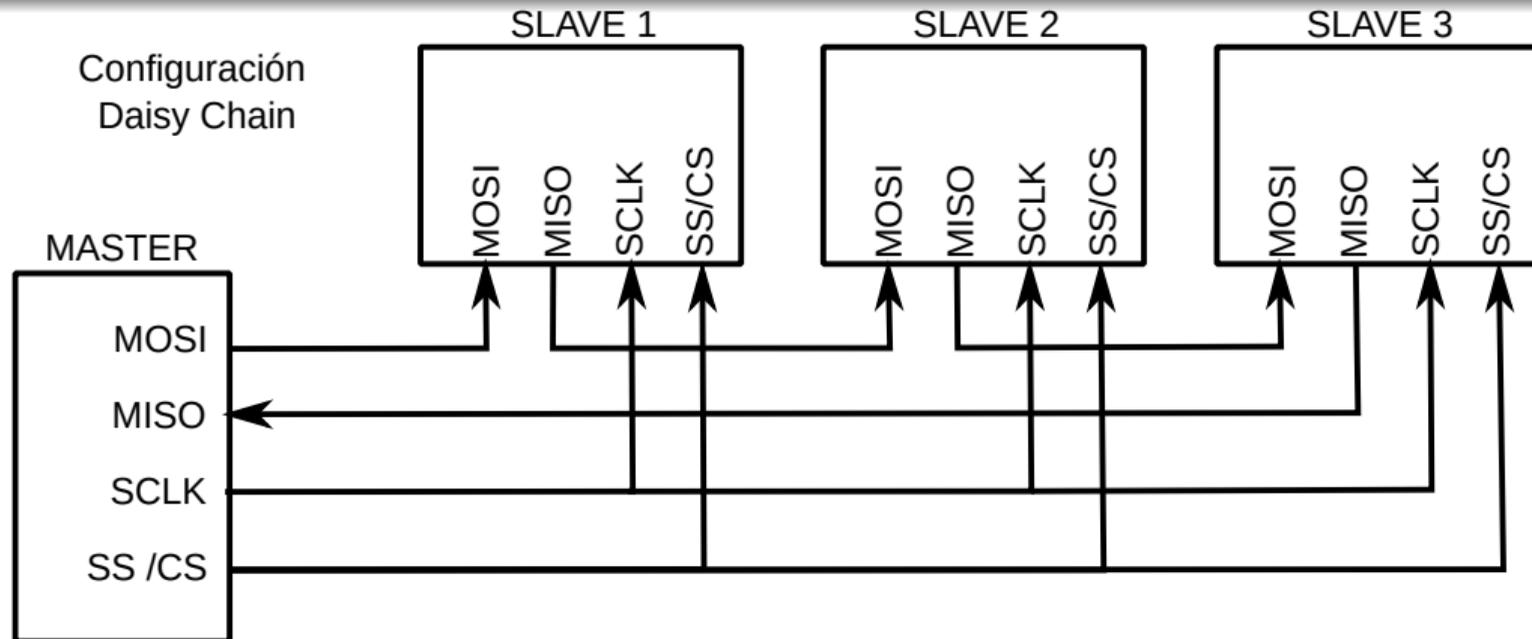
Señal de Configuraciones Multi Slave



Señal de Configuraciones Multi Slave

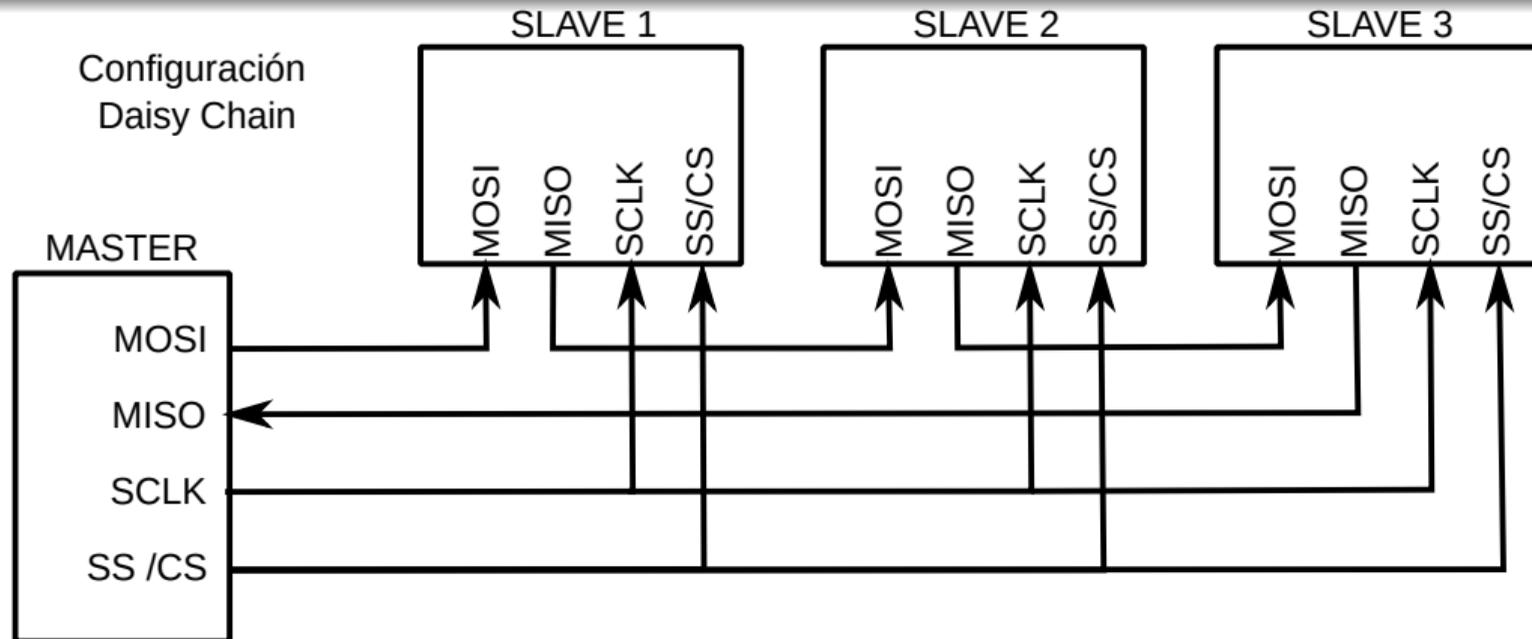


Señal de Configuraciones Multi Slave



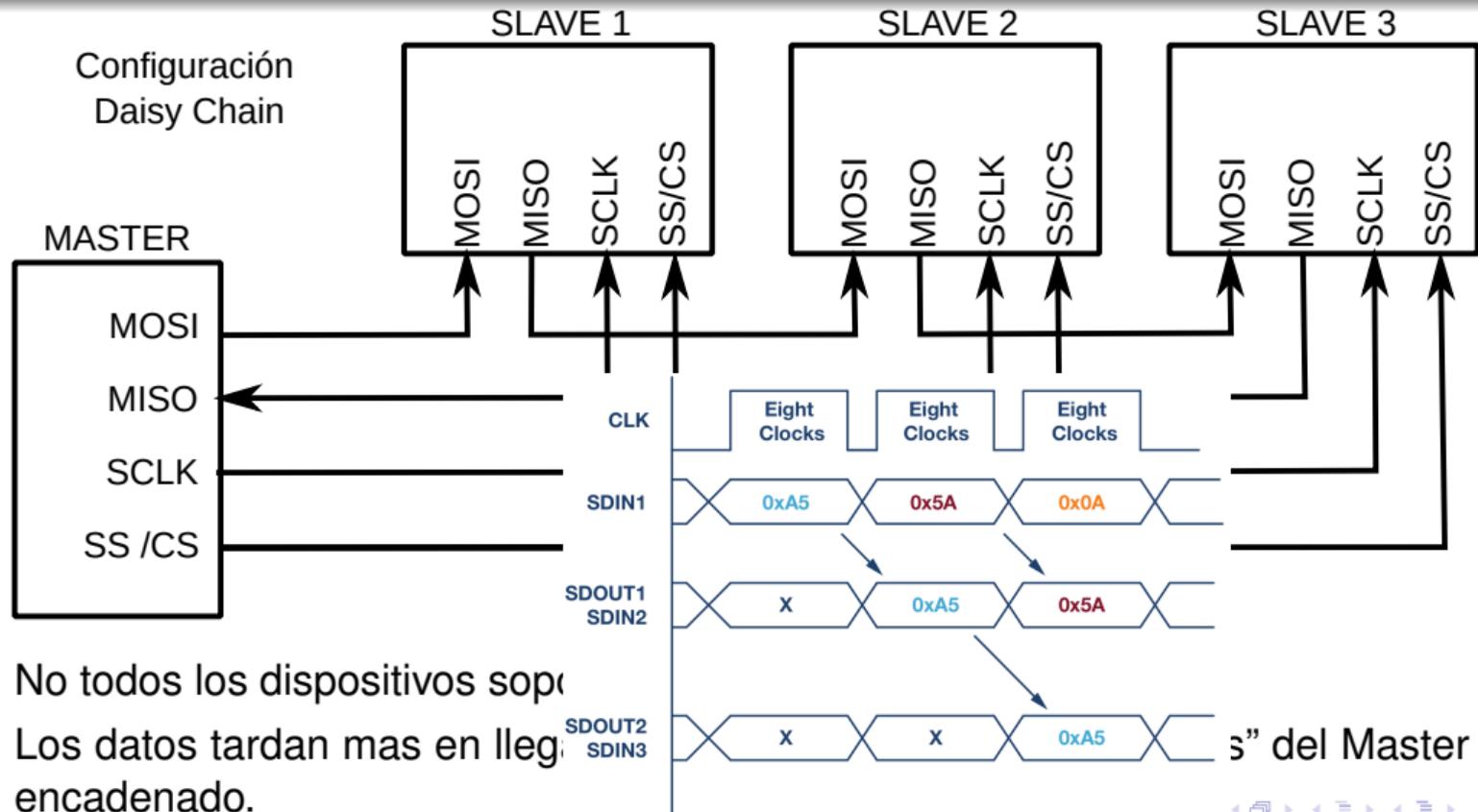
- No todos los dispositivos soportan este método.

Señal de Configuraciones Multi Slave



- No todos los dispositivos soportan este método.
- Los datos tardan mas en llegar a un dispositivo cuanto mas “lejos” del Master está encadenado.

Señal de Configuraciones Multi Slave



- No todos los dispositivos soportan esta configuración.
- Los datos tardan mas en llegar al último dispositivo encadenado.

Señal de Señales MOSI y MISO

- El Master envía datos al esclavo bit a bit, a través de la línea **MOSI**.

Señal de Señales MOSI y MISO

- El Master envía datos al esclavo bit a bit, a través de la línea **MOSI**.
- El Slave recibe los datos enviados desde el MASTER en su pin **MOSI**.

Señal de Señales MOSI y MISO

- El Master envía datos al esclavo bit a bit, a través de la línea **MOSI**.
- El Slave recibe los datos enviados desde el MASTER en su pin **MOSI**.
- Los datos enviados desde el Master al Slave generalmente se envían con el bit más significativo primero.

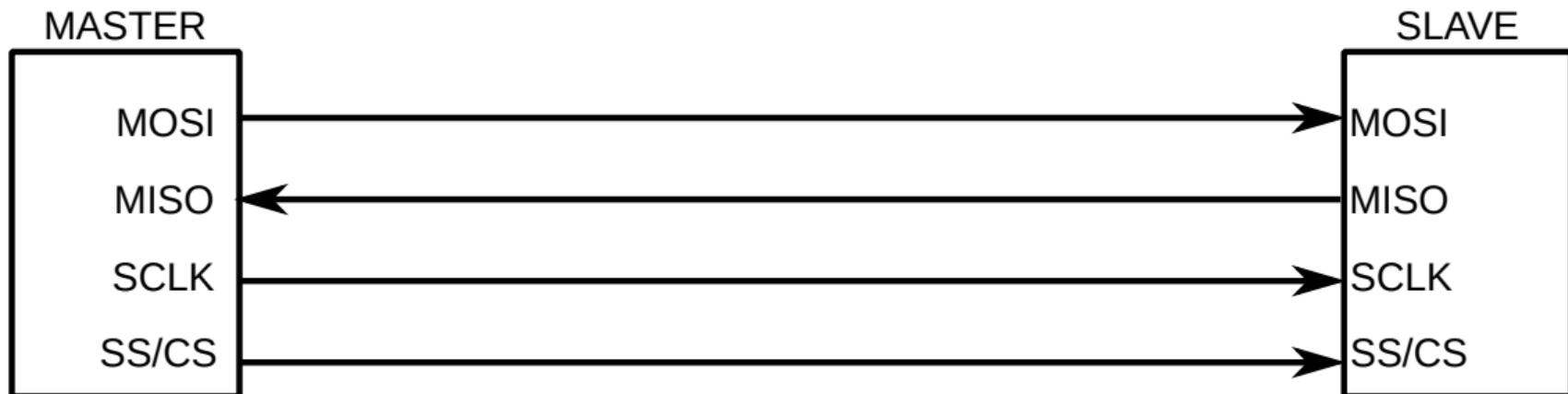
Señal de Señales MOSI y MISO

- El Master envía datos al esclavo bit a bit, a través de la línea **MOSI**.
- El Slave recibe los datos enviados desde el MASTER en su pin **MOSI**.
- Los datos enviados desde el Master al Slave generalmente se envían con el bit más significativo primero.
- El Slave también puede enviar datos al Master a través de la línea **MISO**.

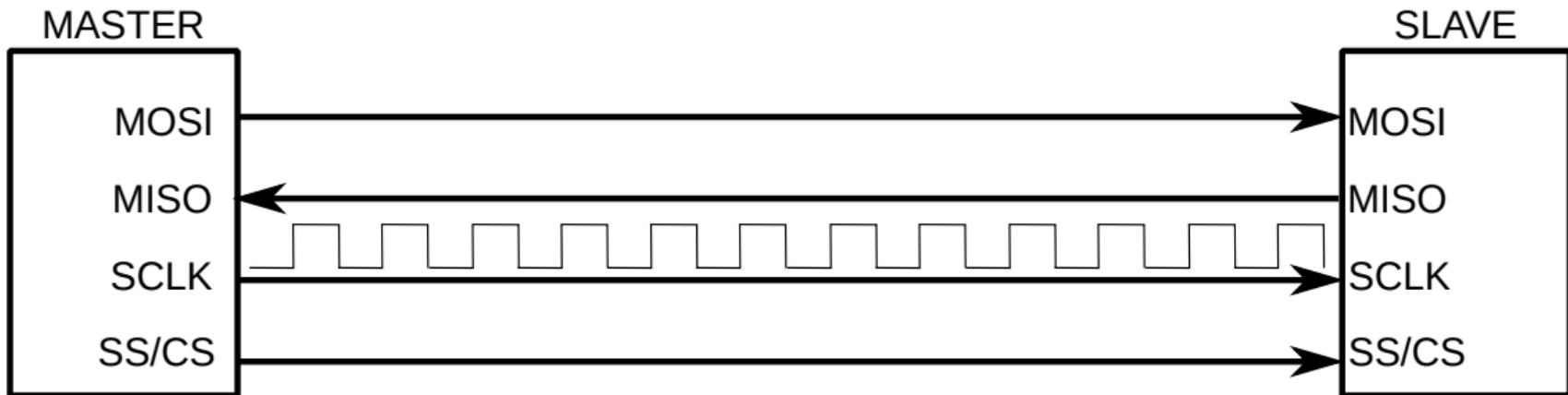
Señal de Señales MOSI y MISO

- El Master envía datos al esclavo bit a bit, a través de la línea **MOSI**.
- El Slave recibe los datos enviados desde el MASTER en su pin **MOSI**.
- Los datos enviados desde el Master al Slave generalmente se envían con el bit más significativo primero.
- El Slave también puede enviar datos al Master a través de la línea **MISO**.
- Los datos enviados desde el Slave al Master generalmente se envían con el bit menos significativo primero.

SPI. Operación

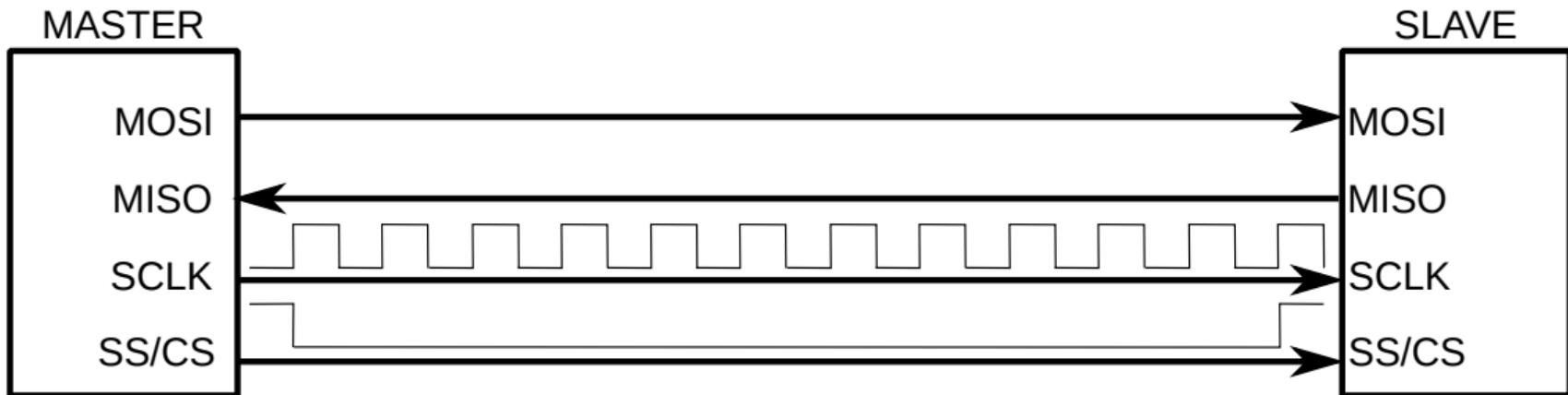


SPI. Operación



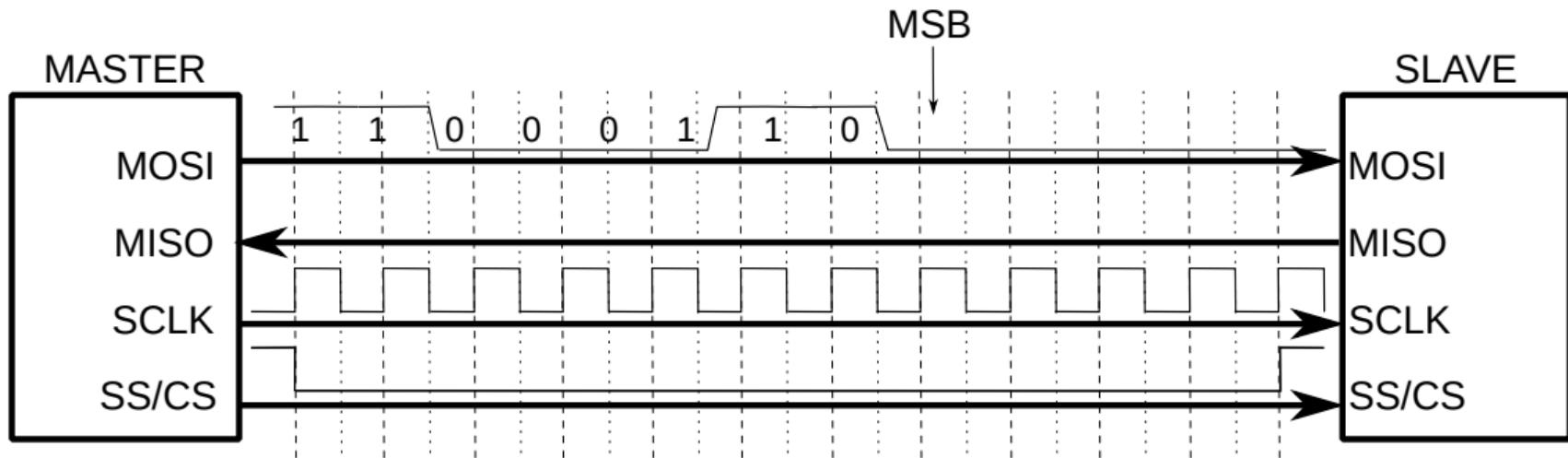
El Master inicia la operación poniendo la señal **SCLK** pra generar el sincronismo entre ambas partes

SPI. Operación



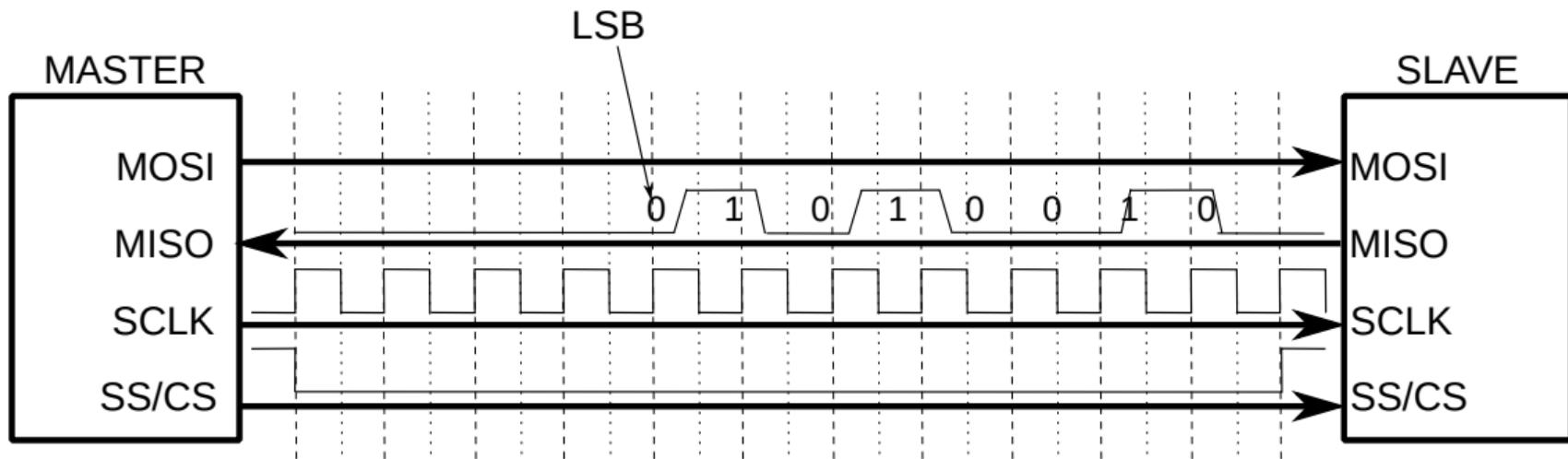
El Master selecciona al Slave mediante la señal **SS** correspondiente

SPI. Operación



El Master envía al Slave seleccionado una trama de n bits (en el gráfico hay 8, pero es ilimitado.) por a línea **MOSI**. El Bit Mas significativo se transmite en primer lugar y luego el resto consecutivamente de acuerdo su peso en la trama. En general los bits se validan en el flanco ascendente de **SCLK**.

SPI. Operación



Si se requiere respuesta el Slave la envía por **MISO** empezando con el Bit menos significativo de la trama detrás del cual viajará el resto en orden de su peso creciente.

Operación según SCLK Poliarity y Phase

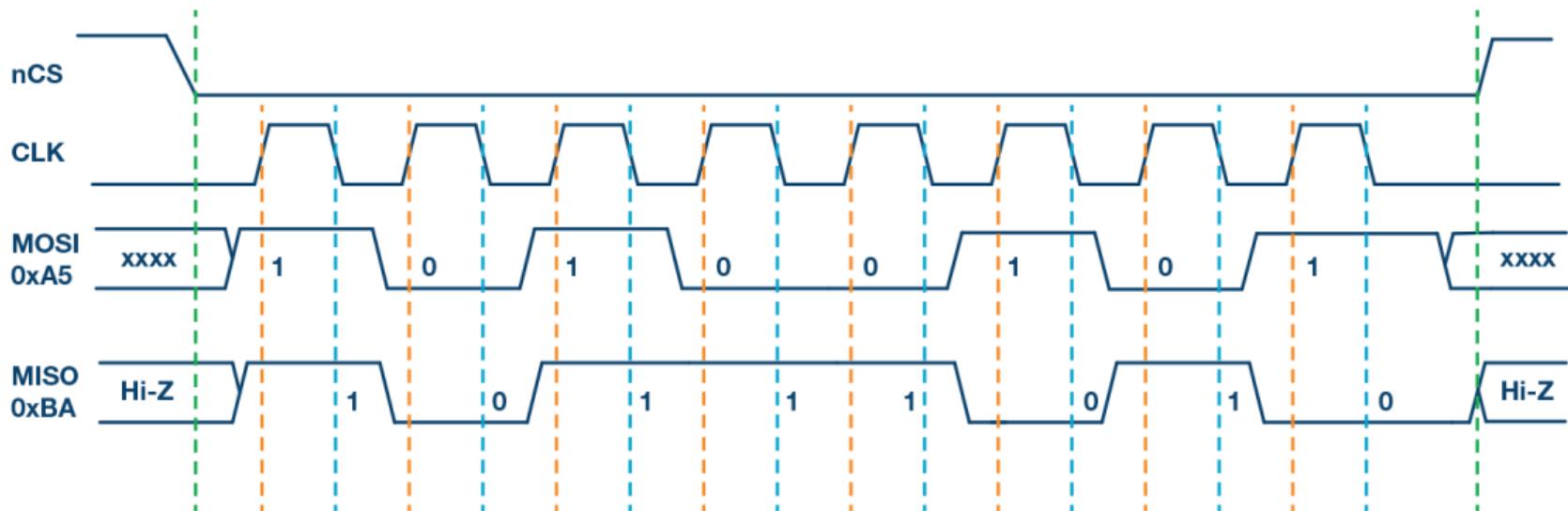
- **CPOL = 0, CPHA = 0** (Generalmente conocido como Modo 0)

Operación según SCLK Poliarity y Phase

- **CPOL = 0, CPHA = 0** (Generalmente conocido como Modo 0)
- CLK idle state = low, Muestrea datos en el flanco ascendente, y desplaza datos en el flanco descendente.

Operación según SCLK Polarity y Phase

- **CPOL = 0, CPHA = 0** (Generalmente conocido como Modo 0)
- CLK idle state = low, Muestra datos en el flanco ascendente, y desplaza datos en el flanco descendente.



Operación según SCLK Poliarity y Phase

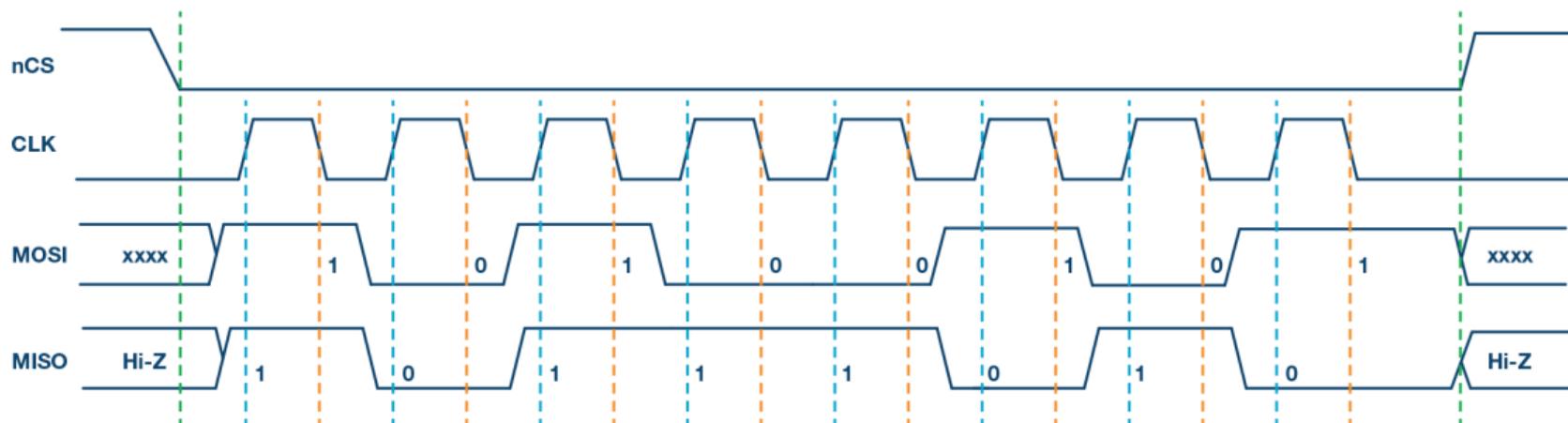
- **CPOL = 0, CPHA = 1** (Generalmente conocido como Modo 1)

Operación según SCLK Poliarity y Phase

- **CPOL** = 0, **CPHA** = 1 (Generalmente conocido como Modo 1)
- CLK idle state = low, Muestrea datos en el flanco descendente, y desplaza datos en el flanco ascendente.

Operación según SCLK Polarity y Phase

- **CPOL = 0, CPHA = 1** (Generalmente conocido como Modo 1)
- CLK idle state = low, Muestra datos en el flanco descendente, y desplaza datos en el flanco ascendente.



Operación según SCLK Poliarity y Phase

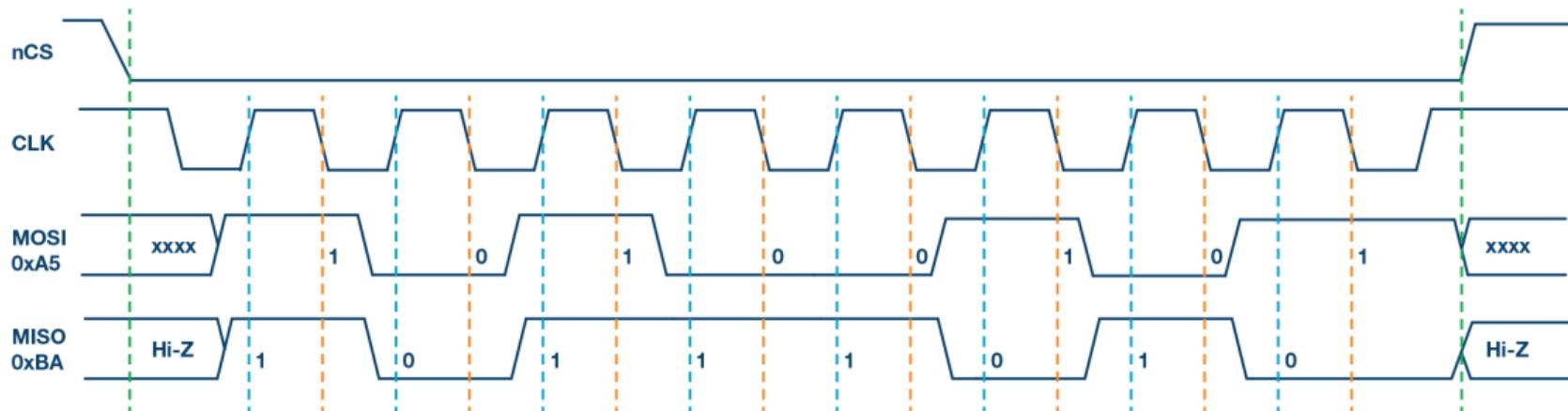
- **CPOL = 1, CPHA = 1** (Generalmente conocido como Modo 2)

Operación según SCLK Poliarity y Phase

- **CPOL = 1, CPHA = 1** (Generalmente conocido como Modo 2)
- CLK idle state = high, Muestrea datos en el flanco descendente, y desplaza datos en el flanco ascendente.

Operación según SCLK Polarity y Phase

- **CPOL = 1, CPHA = 1** (Generalmente conocido como Modo 2)
- CLK idle state = high, Muestrea datos en el flanco descendente, y desplaza datos en el flanco ascendente.



Operación según SCLK Poliarity y Phase

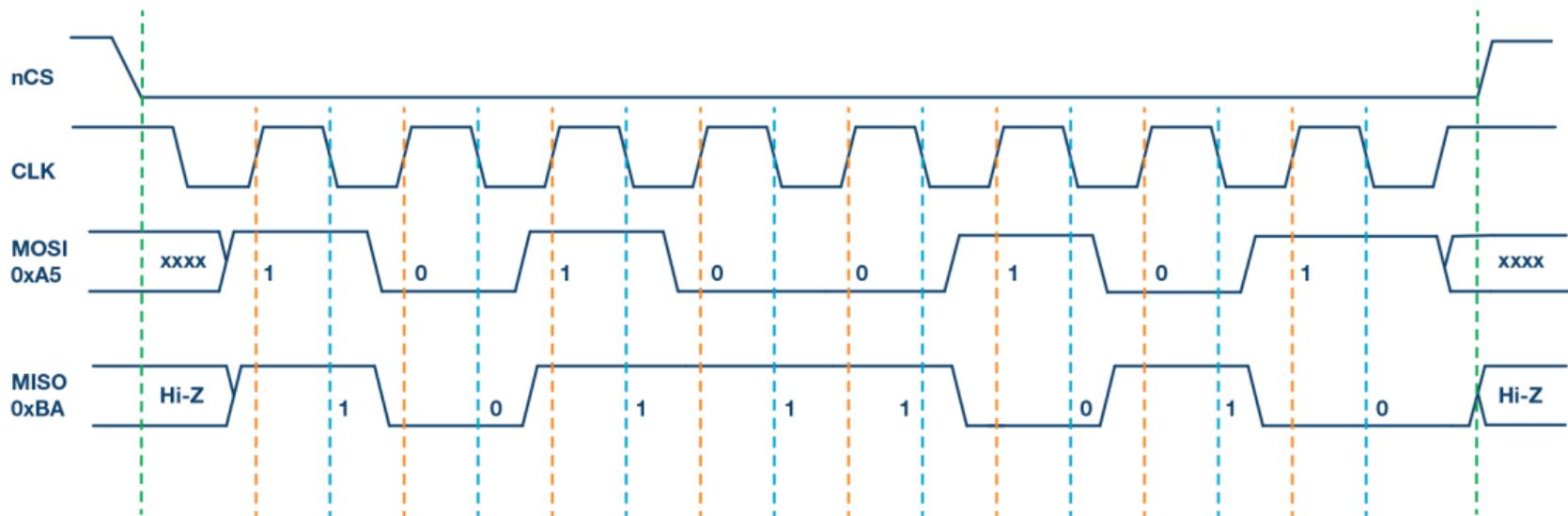
- **CPOL = 1, CPHA = 0** (Generalmente conocido como Modo 3)

Operación según SCLK Poliarity y Phase

- **CPOL** = 1, **CPHA** = 0 (Generalmente conocido como Modo 3)
- CLK idle state = high, Muestrea datos en el flanco ascendente, y desplaza datos en el flanco descendente.

Operación según SCLK Poliarity y Phase

- **CPOL = 1, CPHA = 0** (Generalmente conocido como Modo 3)
- CLK idle state = high, Muestra datos en el flanco ascendente, y desplaza datos en el flanco descendente.



SCLK Polarity y Phase. Conclusiones

- Cada vez que se seleccione un dispositivo SPI deben chequearse los valores de polaridad y fase de clock con que trabaja a fin de ajustar adecuadamente la programación con el Master.

SCLK Polarity y Phase. Conclusiones

- Cada vez que se seleccione un dispositivo SPI deben chequearse los valores de polaridad y fase de clock con que trabaja a fin de ajustar adecuadamente la programación con el Master.
- No nos tomamos el Modo al pie de la letra ya que al no ser SPI un estándar, de acuerdo al fabricante se suelen encontrar dispositivos o controladores, cuyos números de modo que no responden a los que figuran en los slides anteriores.

Temario

- 1 Introducción
 - Motivación
 - Primeros estándares
- 2 **SPI**
 - Introducción
 - Funcionamiento
 - **Implementación SPI en el SoC Sitara AM3358 (BBB)**
- 3 I2C
 - Introducción
 - Funcionamiento
 - Implementación I2C en el SoC Sitara AM3358 (BBB)
- 4 Otros protocolos serie
 - Controller Area Network (CAN)
 - 1-Wire

Implementación

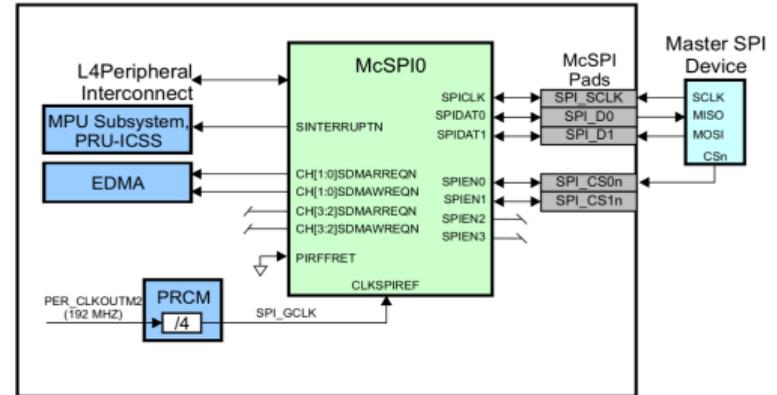
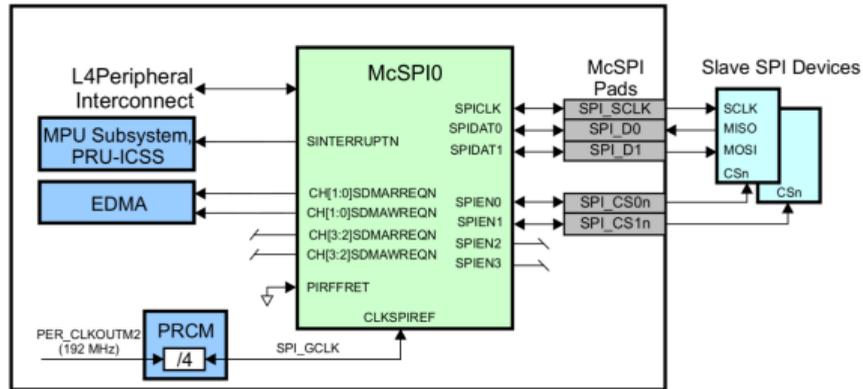
- El Sitara AM3358 posee dos implementaciones de un SPI Multi Canal, Master / Slave (McSPI)

Implementación

- El Sitara AM3358 posee dos implementaciones de un SPI Multi Canal, Master / Slave (McSPI)
- El McSPI es un controlador capaz de interactuar con hasta 4 Slaves o un Master externo

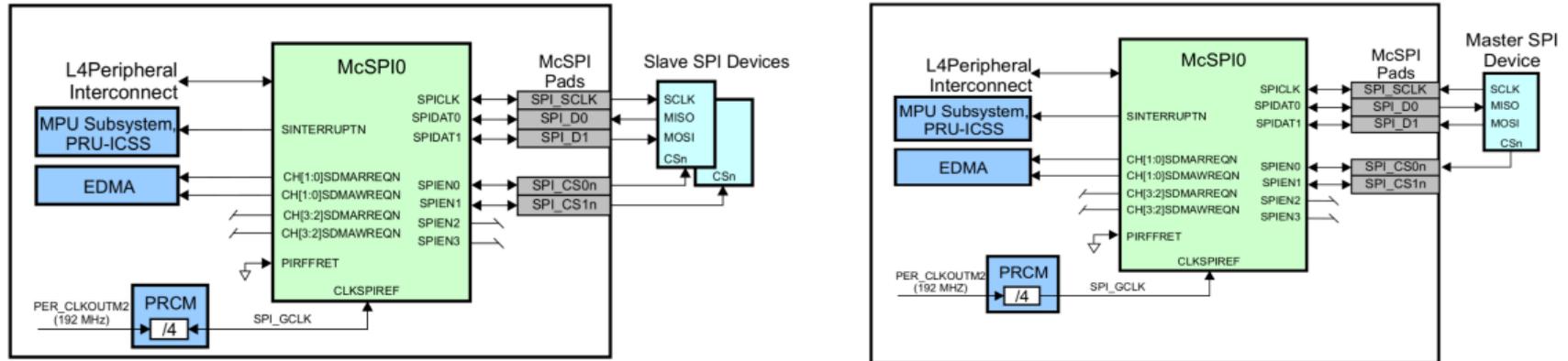
Implementación

- El Sitara AM3358 posee dos implementaciones de un SPI Multi Canal, Master / Slave (McSPI)
- El McSPI es un controlador capaz de interactuar con hasta 4 Slaves o un Master externo



Implementación

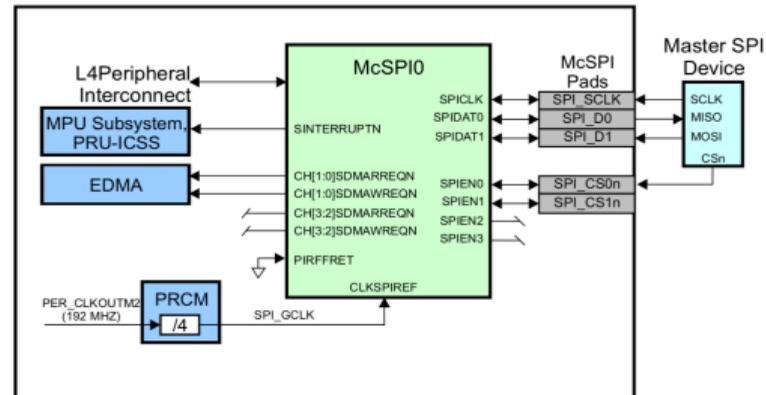
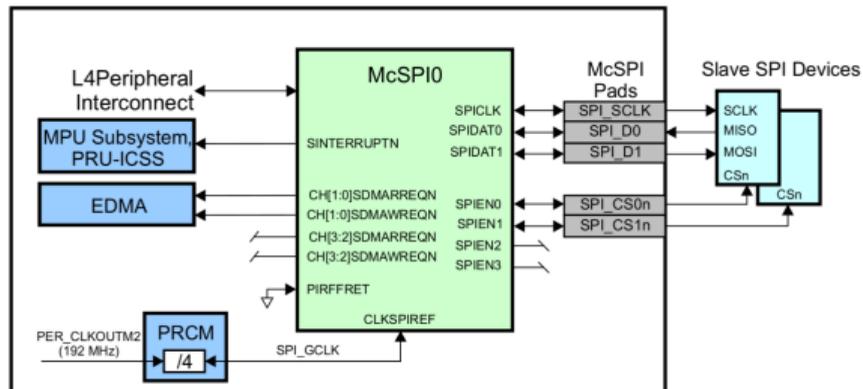
- El Sitara AM3358 posee dos implementaciones de un SPI Multi Canal, Master / Slave (McSPI)
- El McSPI es un controlador capaz de interactuar con hasta 4 Slaves o un Master externo



- Los Chip Select 2 y 3 para Slaves no han sido conectados al pinout.

Implementación

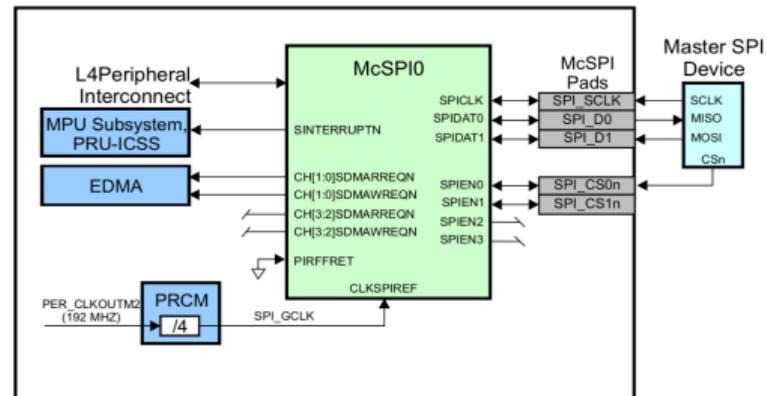
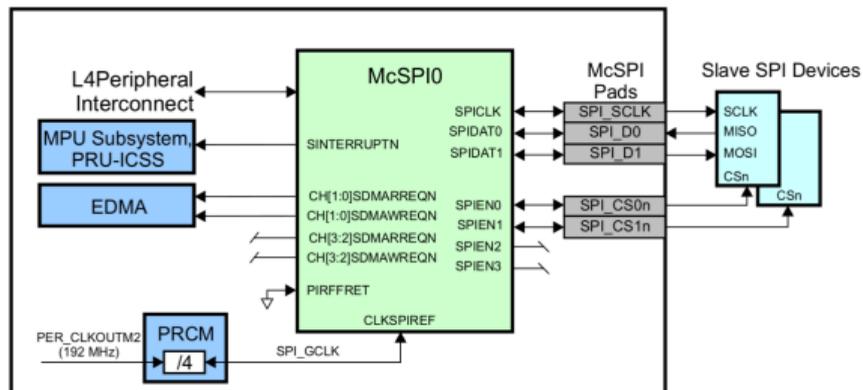
- El Sitara AM3358 posee dos implementaciones de un SPI Multi Canal, Master / Slave (McSPI)
- El McSPI es un controlador capaz de interactuar con hasta 4 Slaves o un Master externo



- Los Chip Select 2 y 3 para Slaves no han sido conectados al pinout.
- Tampoco El Slave Mode Wakeup

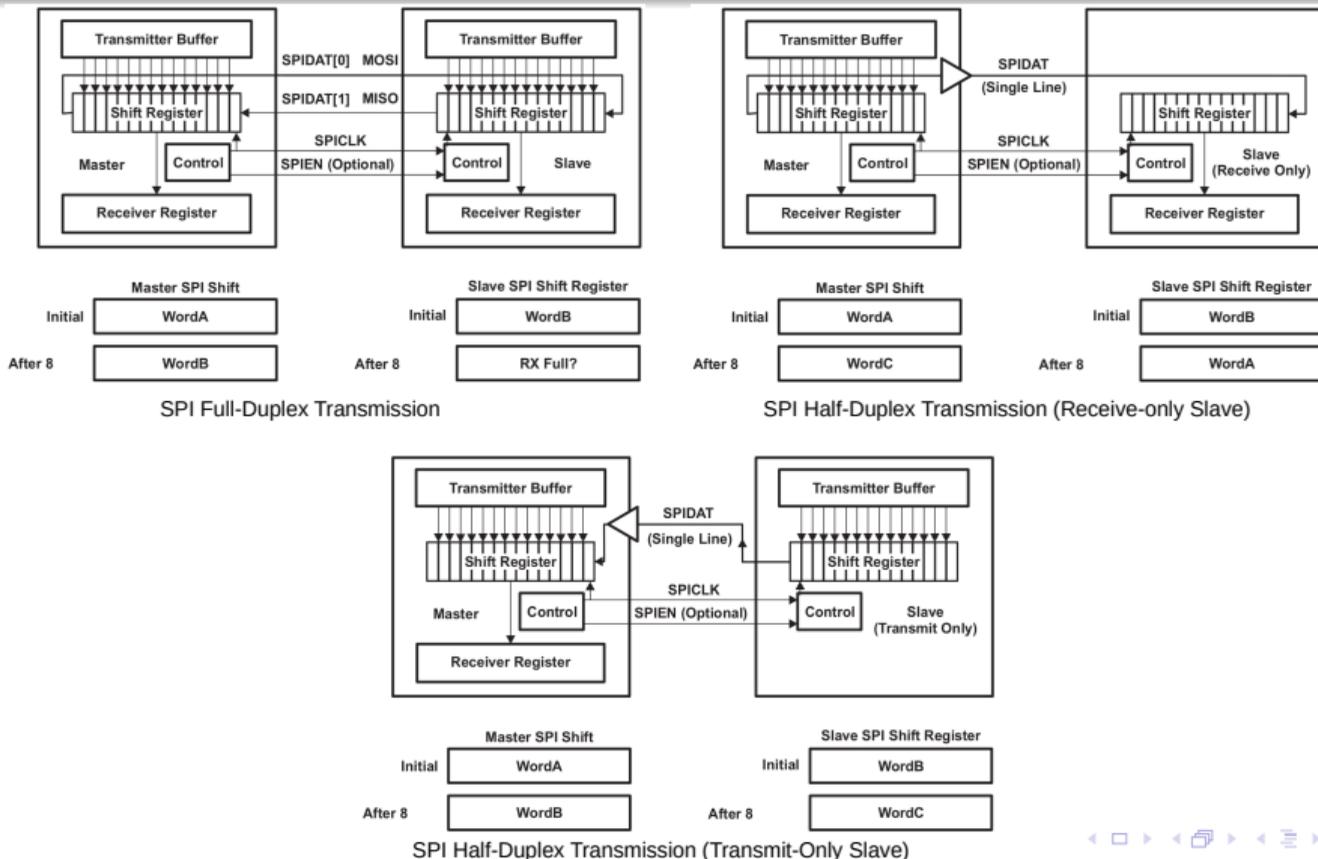
Implementación

- El Sitara AM3358 posee dos implementaciones de un SPI Multi Canal, Master / Slave (McSPI)
- El McSPI es un controlador capaz de interactuar con hasta 4 Slaves o un Master externo



- Los Chip Select 2 y 3 para Slaves no han sido conectados al pinout.
- Tampoco El Slave Mode Wakeup
- No esta sintetizado el Modo Retención durante Power Down

AM3358: Transmisión SPI



AM3358: Formatos de transferencia

- Tamaño de palabra programable: Rango 4 a 32 bits.

AM3358: Formatos de transferencia

- Tamaño de palabra programable: Rango 4 a 32 bits.
- Generación Programable de SPI Enable. Si se habilita solo un Master y un Slave pueden conectarse al Bus

AM3358: Formatos de transferencia

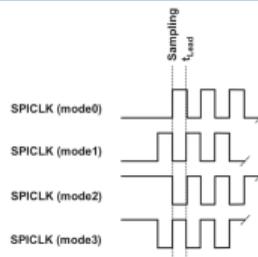
- Tamaño de palabra programable: Rango 4 a 32 bits.
- Generación Programable de SPI Enable. Si se habilita solo un Master y un Slave pueden conectarse al Bus
- SPI Enable programable, tanto en polaridad (Activo alto o activo bajo), como en envío automático o manual. Permite que dos palabras consecutivas a Slaves diferentes se envíen con diferente polaridad

AM3358: Formatos de transferencia

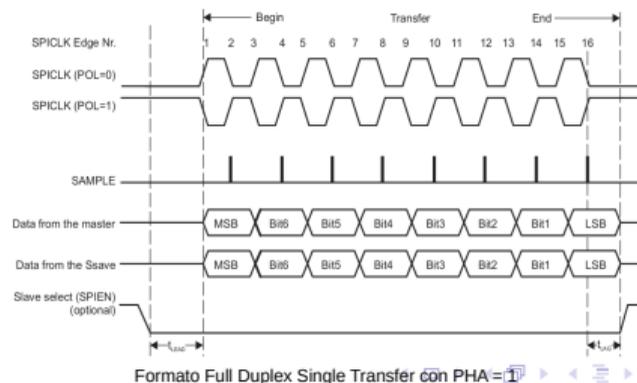
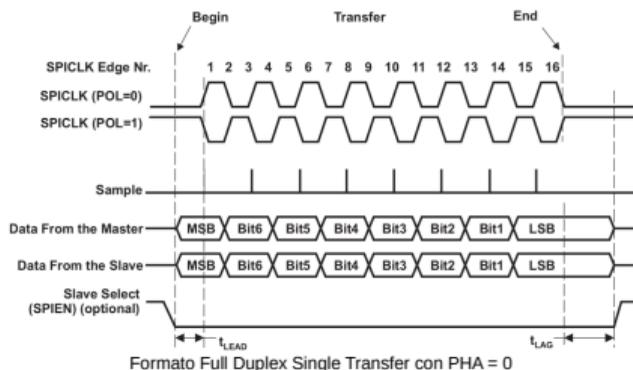
- Tamaño de palabra programable: Rango 4 a 32 bits.
- Generación Programable de SPI Enable. Si se habilita solo un Master y un Slave pueden conectarse al Bus
- SPI Enable programable, tanto en polaridad (Activo alto o activo bajo), como en envío automático o manual. Permite que dos palabras consecutivas a Slaves diferentes se envíen con diferente polaridad
- SPI CLOCK Programable. Polaridad Fase y frecuencia (ésta última solo en modo Master).

AM3358: Formatos de transferencia

Polarity (POL)	Phase (PHA)	Modo SPI	Comentarios
0	0	mode0	SPICLK activo alto. Sampling flanco ascendente.
0	1	mode1	SPICLK activo alto. Sampling flanco descendente.
1	0	mode2	SPICLK activo bajo. Sampling flanco descendente.
1	1	mode3	SPICLK activo bajo. Sampling flanco ascendente.



Combinaciones de Fase y Polaridad



AM3358: Recursos adicionales

- Registros de datos Rx/Tx por cada canal (1 word de profundidad).

AM3358: Recursos adicionales

- Registros de datos Rx/Tx por cada canal (1 word de profundidad).
- FIFO para Acceso Multiples word en un mismo canal.

AM3358: Recursos adicionales

- Registros de datos Rx/Tx por cada canal (1 word de profundidad).
- FIFO para Acceso Múltiple word en un mismo canal.
- Dos DMA requests por canal, con una línea de interrupción.

AM3358: Recursos adicionales

- Registros de datos Rx/Tx por cada canal (1 word de profundidad).
- FIFO para Acceso Multiples word en un mismo canal.
- Dos DMA requests por canal, con una línea de interrupción.
- Línea de interrupción única para múltiples eventos fuente.

AM3358: Recursos adicionales

- Registros de datos Rx/Tx por cada canal (1 word de profundidad).
- FIFO para Acceso Multiples word en un mismo canal.
- Dos DMA requests por canal, con una línea de interrupción.
- Línea de interrupción única para múltiples eventos fuente.
- El enlace serie soporta Full duplex / Half duplex, en operación Multi-channel Master o Single channel Slave, Operación shift programable de 1 a 32 bit en transmisión o recepción, y Longitud de palabra programable de 4 a 32 bits.

AM3358: Recursos adicionales

- Registros de datos Rx/Tx por cada canal (1 word de profundidad).
- FIFO para Acceso Multiples word en un mismo canal.
- Dos DMA requests por canal, con una línea de interrupción.
- Línea de interrupción única para múltiples eventos fuente.
- El enlace serie soporta Full duplex / Half duplex, en operación Multi-channel Master o Single channel Slave, Operación shift programable de 1 a 32 bit en transmisión o recepción, y Longitud de palabra programable de 4 a 32 bits.
- Hasta 4 canales SPI (Solo disponibles en Pinout canales 0 y 1)

AM3358: Recursos adicionales

- Registros de datos Rx/Tx por cada canal (1 word de profundidad).
- FIFO para Acceso Multiples word en un mismo canal.
- Dos DMA requests por canal, con una línea de interrupción.
- Línea de interrupción única para múltiples eventos fuente.
- El enlace serie soporta Full duplex / Half duplex, en operación Multi-channel Master o Single channel Slave, Operación shift programable de 1 a 32 bit en transmisión o recepción, y Longitud de palabra programable de 4 a 32 bits.
- Hasta 4 canales SPI (Solo disponibles en Pinout canales 0 y 1)
- Asignación de slots de Transmisión / Recepción de palabras basado en arbitración del tipo round robin.

AM3358: Recursos adicionales

- Registros de datos Rx/Tx por cada canal (1 word de profundidad).
- FIFO para Acceso Múltiples word en un mismo canal.
- Dos DMA requests por canal, con una línea de interrupción.
- Línea de interrupción única para múltiples eventos fuente.
- El enlace serie soporta Full duplex / Half duplex, en operación Multi-channel Master o Single channel Slave, Operación shift programable de 1 a 32 bit en transmisión o recepción, y Longitud de palabra programable de 4 a 32 bits.
- Hasta 4 canales SPI (Solo disponibles en Pinout canales 0 y 1)
- Asignación de slots de Transmisión / Recepción de palabras basado en arbitración del tipo round robin.
- Soporte para generación de Clock Programmable al Master (a partir de una entrada de clock funcional fija de 48-MHz), y phase, polarity programables por Chip Select.

AM3358: Recursos adicionales

Lecturas y Documentación adicional:

- Introduction to SPI Interface. By Piyu Dhaker. Analog Devices.
- AM335x and AMIC110 Sitara™ Processors Technical Reference Manual. Chapter 24 McSPI. Texas Instruments

Temario

- 1 Introducción
 - Motivación
 - Primeros estándares
- 2 SPI
 - Introducción
 - Funcionamiento
 - Implementación SPI en el SoC Sitara AM3358 (BBB)
- 3 **I2C**
 - **Introducción**
 - Funcionamiento
 - Implementación I2C en el SoC Sitara AM3358 (BBB)
- 4 Otros protocolos serie
 - Controller Area Network (CAN)
 - 1-Wire

I2C. Características principales

- Es uno de los protocolos mas utilizados para la interconexión en sistemas embebidos entre la CPU y sus dispositivos periféricos.

I2C. Características principales

- Es uno de los protocolos mas utilizados para la interconexión en sistemas embebidos entre la CPU y sus dispositivos periféricos.
- Displays OLED, sensores barométricos de presión, giroscopios / acelerómetros, son casos comunes de interconexión por I2C.

I2C. Características principales

- Es uno de los protocolos mas utilizados para la interconexión en sistemas embebidos entre la CPU y sus dispositivos periféricos.
- Displays OLED, sensores barométricos de presión, giroscopios / acelerómetros, son casos comunes de interconexión por I2C.
- Es particularmente útil cuando una memoria o un LCD deben ser accedidos por varias CPUs en forma concurrente.

I2C. Características principales

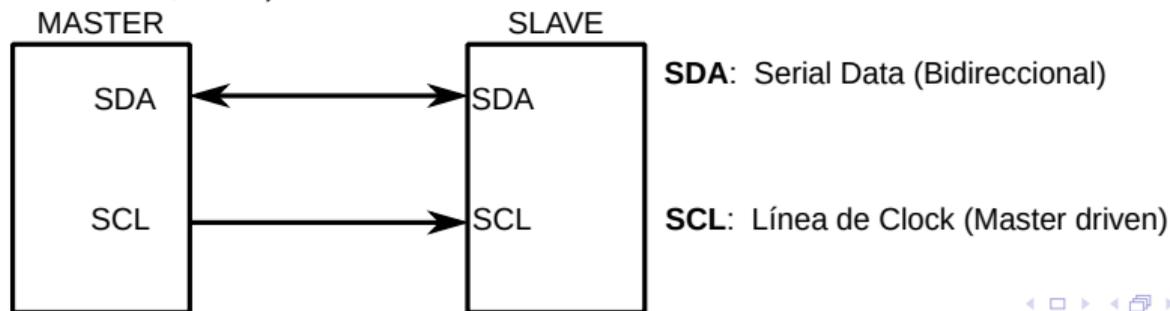
- Es uno de los protocolos mas utilizados para la interconexión en sistemas embebidos entre la CPU y sus dispositivos periféricos.
- Displays OLED, sensores barométricos de presión, giroscopios / acelerómetros, son casos comunes de interconexión por I2C.
- Es particularmente útil cuando una memoria o un LCD deben ser accedidos por varias CPUs en forma concurrente.
- Como las UART la información se transmite por una única línea.

I2C. Características principales

- Es uno de los protocolos mas utilizados para la interconexión en sistemas embebidos entre la CPU y sus dispositivos periféricos.
- Displays OLED, sensores barométricos de presión, giroscopios / acelerómetros, son casos comunes de interconexión por I2C.
- Es particularmente útil cuando una memoria o un LCD deben ser accedidos por varias CPUs en forma concurrente.
- Como las UART la información se transmite por una única línea.
- Operan en configuración Master (típicamente la CPU) Slave (el dispositivo: Sensor, display, memoria, etc.)

I2C. Características principales

- Es uno de los protocolos mas utilizados para la interconexión en sistemas embebidos entre la CPU y sus dispositivos periféricos.
- Displays OLED, sensores barométricos de presión, giroscopios / acelerómetros, son casos comunes de interconexión por I2C.
- Es particularmente útil cuando una memoria o un LCD deben ser accedidos por varias CPUs en forma concurrente.
- Como las UART la información se transmite por una única línea.
- Operan en configuración Master (típicamente la CPU) Slave (el dispositivo: Sensor, display, memoria, etc.)



I2C. Características principales

- Cantidad de alambres: 2

I2C. Características principales

- Cantidad de alambres: 2
- Ancho de banda máximo:

I2C. Características principales

- Cantidad de alambres: 2
- Ancho de banda máximo:
 - Modo Standard: 100 kbit s

I2C. Características principales

- Cantidad de alambres: 2
- Ancho de banda máximo:
 - Modo Standard: 100 kbit s
 - Fast Mode 400 kbit s,

I2C. Características principales

- Cantidad de alambres: 2
- Ancho de banda máximo:
 - Modo Standard: 100 kbit s
 - Fast Mode 400 kbit s,
 - High Speed Mode 3,4 Mbit s

I2C. Características principales

- Cantidad de alambres: 2
- Ancho de banda máximo:
 - Modo Standard: 100 kbit s
 - Fast Mode 400 kbit s,
 - High Speed Mode 3,4 Mbit s
 - Ultra Fast Mode 5 Mbit s

I2C. Características principales

- Cantidad de alambres: 2
- Ancho de banda máximo:
 - Modo Standard: 100 kbit s
 - Fast Mode 400 kbit s,
 - High Speed Mode 3,4 Mbit s
 - Ultra Fast Mode 5 Mbit s
- Comunicación sincrónica (**SCL**).

I2C. Características principales

- Cantidad de alambres: 2
- Ancho de banda máximo:
 - Modo Standard: 100 kbit s
 - Fast Mode 400 kbit s,
 - High Speed Mode 3,4 Mbit s
 - Ultra Fast Mode 5 Mbit s
- Comunicación sincrónica (**SCL**).
- Cantidad de Dispositivos Master: Ilimitada

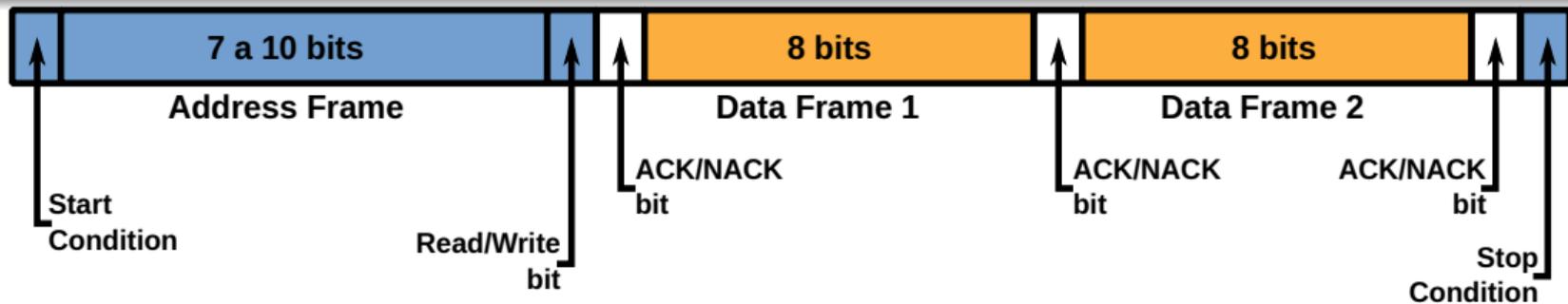
I2C. Características principales

- Cantidad de alambres: 2
- Ancho de banda máximo:
 - Modo Standard: 100 kbit s
 - Fast Mode 400 kbit s,
 - High Speed Mode 3,4 Mbit s
 - Ultra Fast Mode 5 Mbit s
- Comunicación sincrónica (**SCL**).
- Cantidad de Dispositivos Master: Ilimitada
- Cantidad de Slaves: 1008

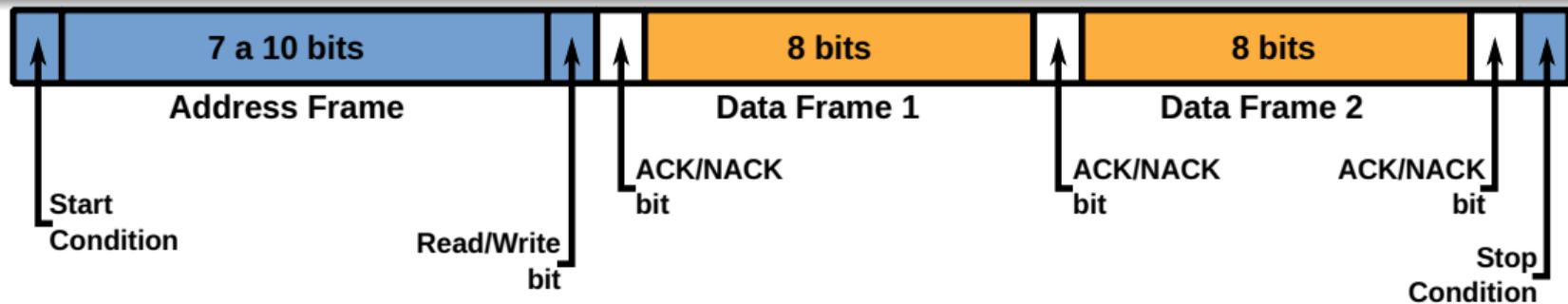
Temario

- 1 Introducción
 - Motivación
 - Primeros estándares
- 2 SPI
 - Introducción
 - Funcionamiento
 - Implementación SPI en el SoC Sitara AM3358 (BBB)
- 3 I2C**
 - Introducción
 - Funcionamiento**
 - Implementación I2C en el SoC Sitara AM3358 (BBB)
- 4 Otros protocolos serie
 - Controller Area Network (CAN)
 - 1-Wire

I2C. Formato del mensaje

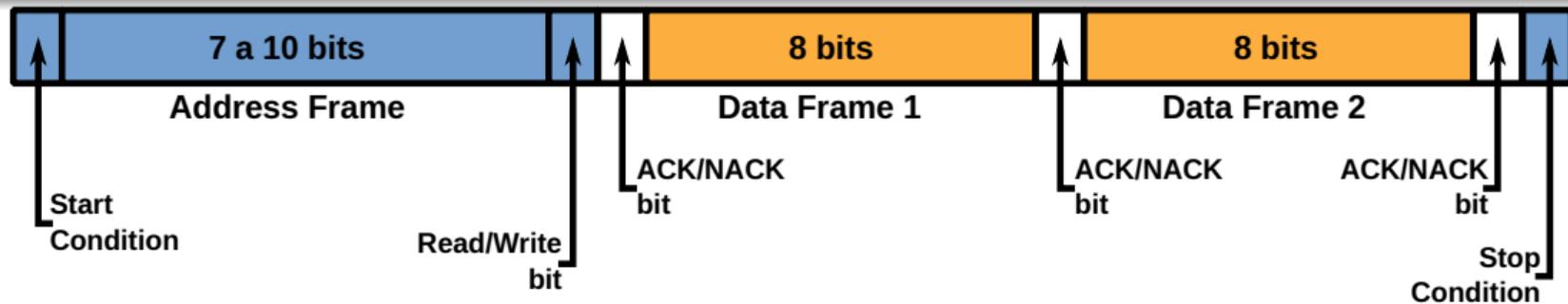


I2C. Formato del mensaje



Start Condition: Se produce cuando la línea **SDA** que se encuentra normalmente en un nivel de tensión alto cae a un nivel de tensión bajo, antes que la línea **SCL** ha pasado de Estado alto a Estado bajo

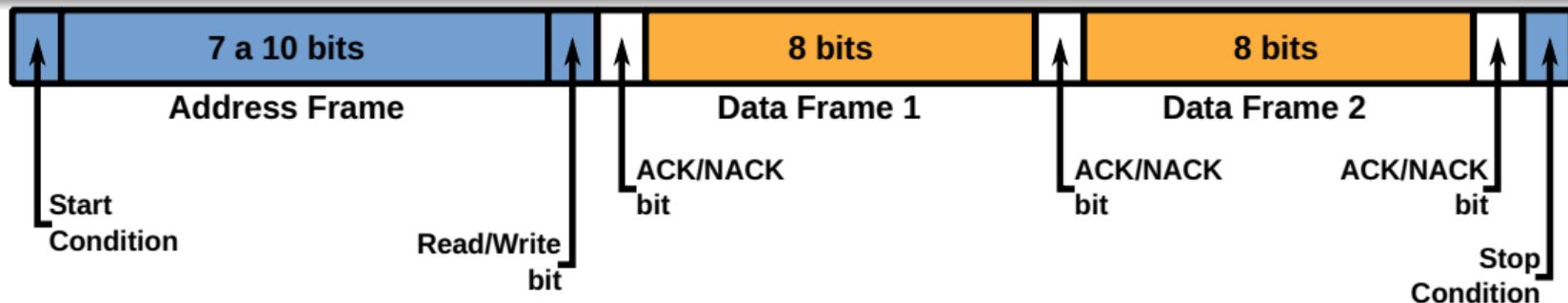
I2C. Formato del mensaje



Start Condition: Se produce cuando la línea **SDA** que se encuentra normalmente en un nivel de tensión alto cae a un nivel de tensión bajo, antes que la línea **SCL** ha pasado de Estado alto a Estado bajo

Stop Condition: Se produce cuando la línea **SDA** pasa a un nivel de tensión alto desde un nivel de tensión bajo, luego de que la línea **SCL** ha pasado de Estado bajo a Estado alto

I2C. Formato del mensaje

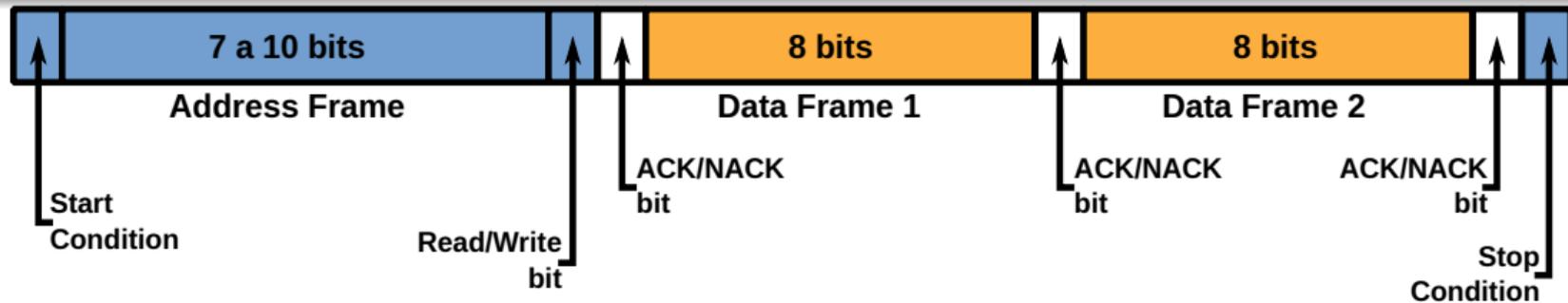


Start Condition: Se produce cuando la línea **SDA** que se encuentra normalmente en un nivel de tensión alto cae a un nivel de tensión bajo, antes que la línea **SCL** ha pasado de Estado alto a Estado bajo

Stop Condition: Se produce cuando la línea **SDA** pasa a un nivel de tensión alto desde un nivel de tensión bajo, luego de que la línea **SCL** ha pasado de Estado bajo a Estado alto.

Address Frame: Este número de 7 a 10 bits identifica unívocamente al slave con el que el Master que inicia la transmisión requiere comunicarse.

I2C. Formato del mensaje



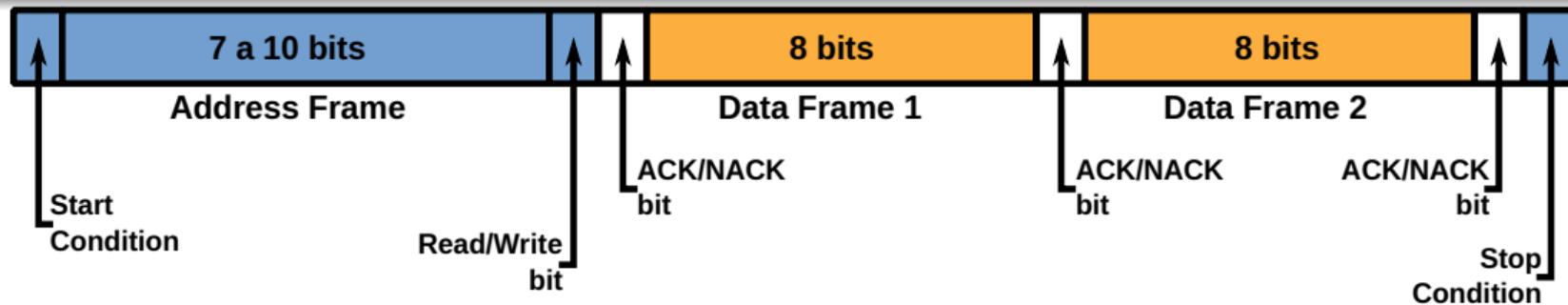
Start Condition: Se produce cuando la línea **SDA** que se encuentra normalmente en un nivel de tensión alto cae a un nivel de tensión bajo, antes que la línea **SCL** ha pasado de Estado alto a Estado bajo

Stop Condition: Se produce cuando la línea **SDA** pasa a un nivel de tensión alto desde un nivel de tensión bajo, luego de que la línea **SCL** ha pasado de Estado bajo a Estado alto.

Address Frame: Este número de 7 a 10 bits identifica unívocamente al slave con el que el Master que inicia la transmisión requiere comunicarse.

Read/Write bit: Especifica si el master envía datos a slave (estado bajo) o si requiere datos del slave (estado alto).

I2C. Formato del mensaje



Start Condition: Se produce cuando la línea **SDA** que se encuentra normalmente en un nivel de tensión alto cae a un nivel de tensión bajo, antes que la línea **SCL** ha pasado de Estado alto a Estado bajo

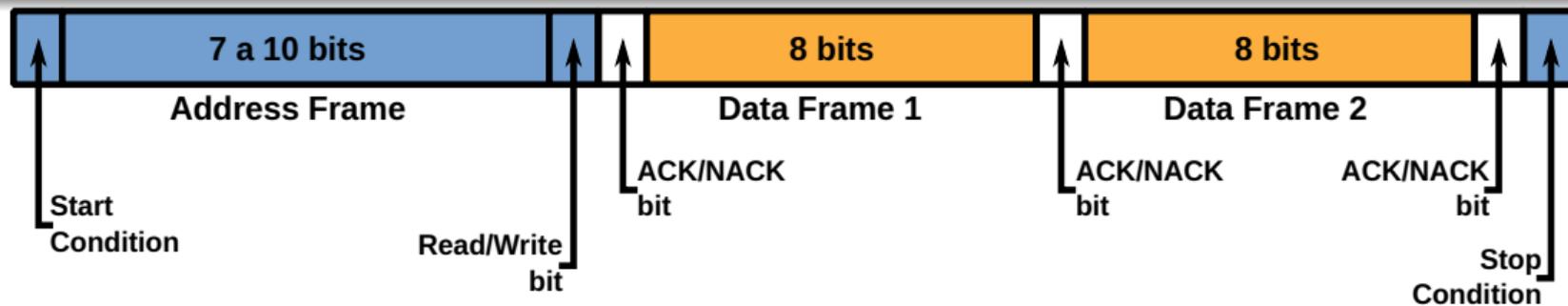
Stop Condition: Se produce cuando la línea **SDA** pasa a un nivel de tensión alto desde un nivel de tensión bajo, luego de que la línea **SCL** ha pasado de Estado bajo a Estado alto.

Address Frame: Este número de 7 a 10 bits identifica unívocamente al slave con el que el Master que inicia la transmisión requiere comunicarse.

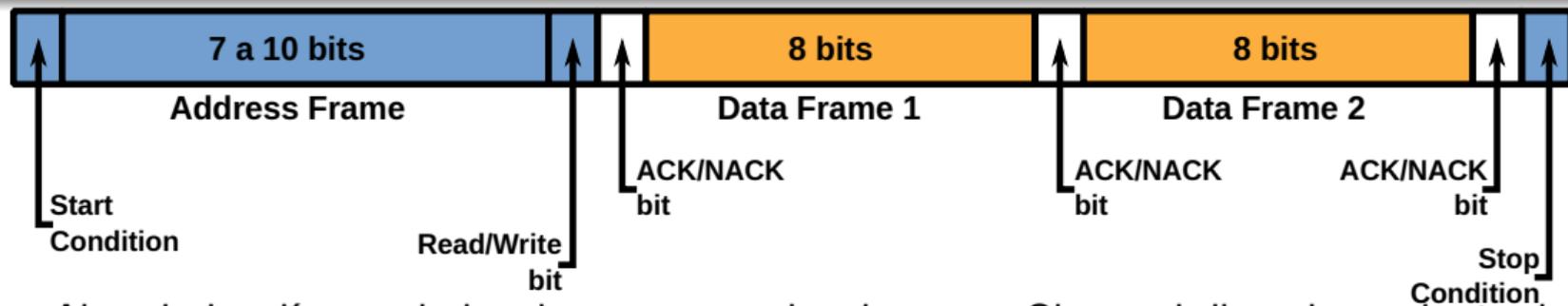
Read/Write bit: Especifica si el master envía datos a slave (estado bajo) o si requiere datos del slave (estado alto).

ACK/NACK bit: Cada frame es seguido de un bit de ACK o NACK que indica si el frame de datos se recibió en forma exitosa (ACK) o no (NACK).

Direcccionamiento

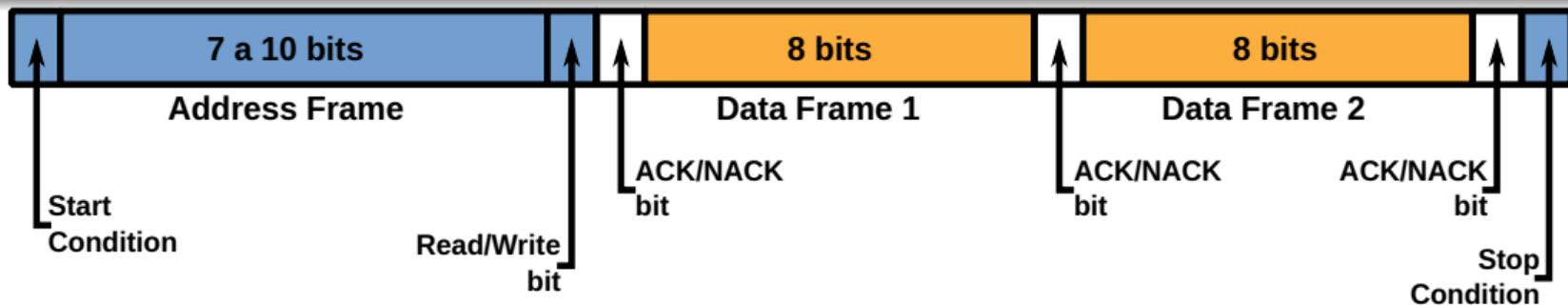


Direcccionamiento

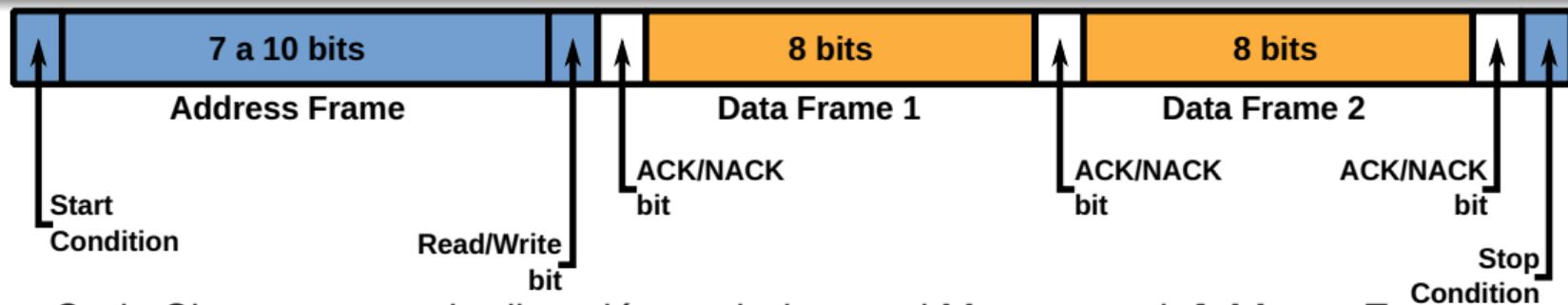


- Al no haber líneas de hardware para seleccionar un Slave, el direccionamiento debe necesariamente incluirse en la trama de comunicación.
- Dentro de la trama del mensaje, luego del bit de Start, se incluye el **Address Frame** para direccionar al Slave con el que el Master que inicia la transacción necesita comunicarse
- El mensaje es recibido por todos los Slaves conectados por la línea **SDA** con el Master.

Direcccionamiento

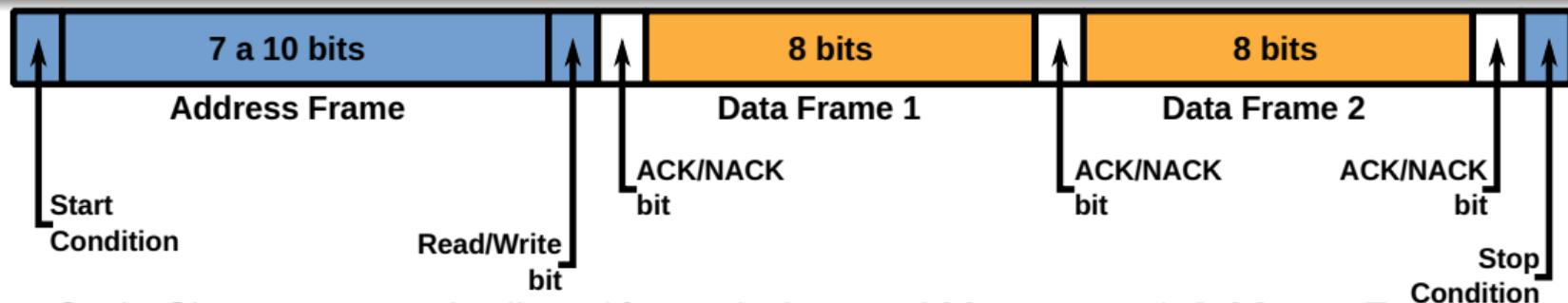


Direcccionamiento



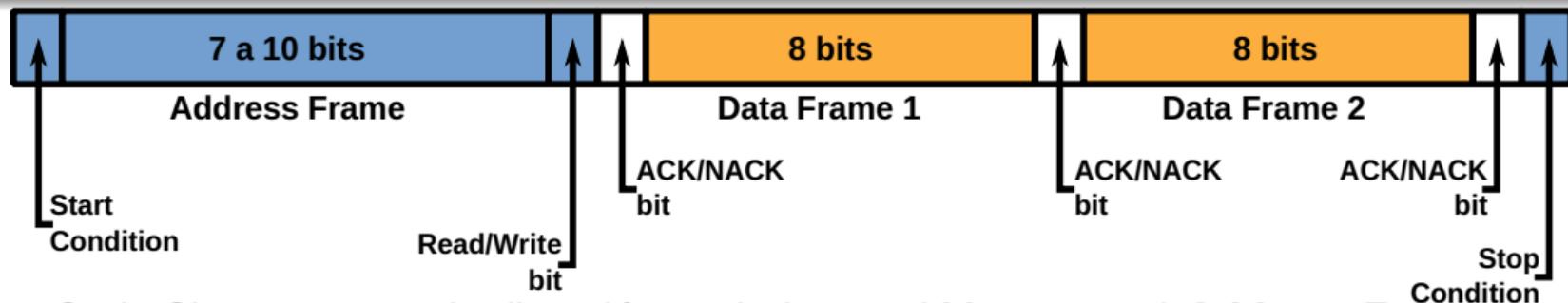
- Cada Slave compara la dirección enviada por el Master en el **Address Frame** con su propia dirección.

Direcccionamiento



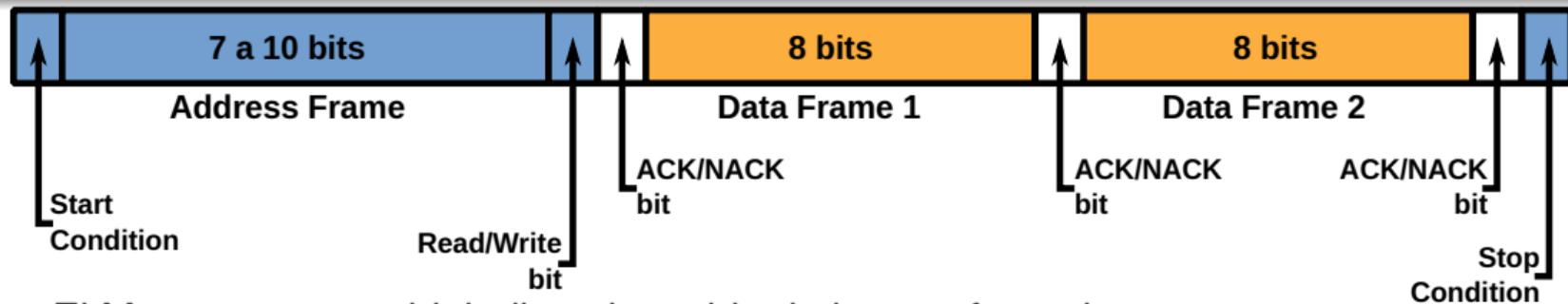
- Cada Slave compara la dirección enviada por el Master en el **Address Frame** con su propia dirección.
- Solo el Slave cuya dirección programada coincida, con el **Address Frame**, responderá con un bit de ACK (nivel de tensión bajo).

Direcccionamiento



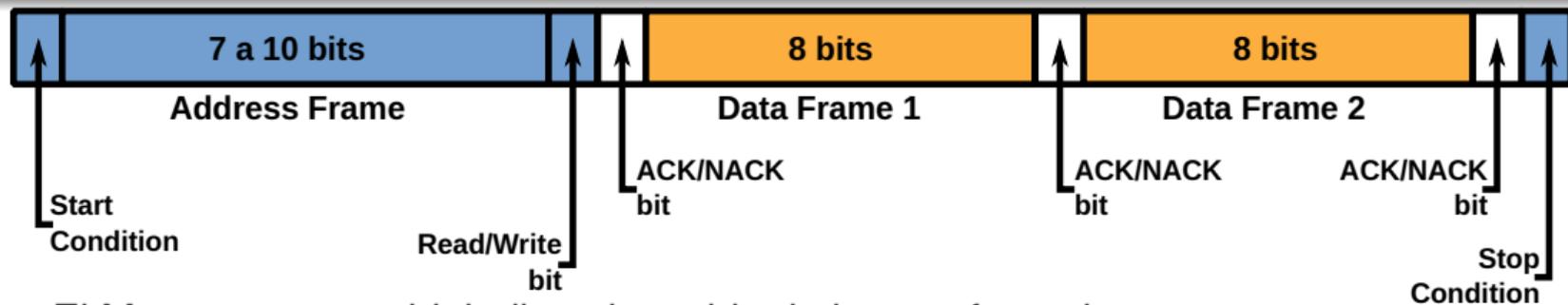
- Cada Slave compara la dirección enviada por el Master en el **Address Frame** con su propia dirección.
- Solo el Slave cuya dirección programada coincida, con el **Address Frame**, responderá con un bit de ACK (nivel de tensión bajo).
- El resto no hace nada.

Read / Write Bit



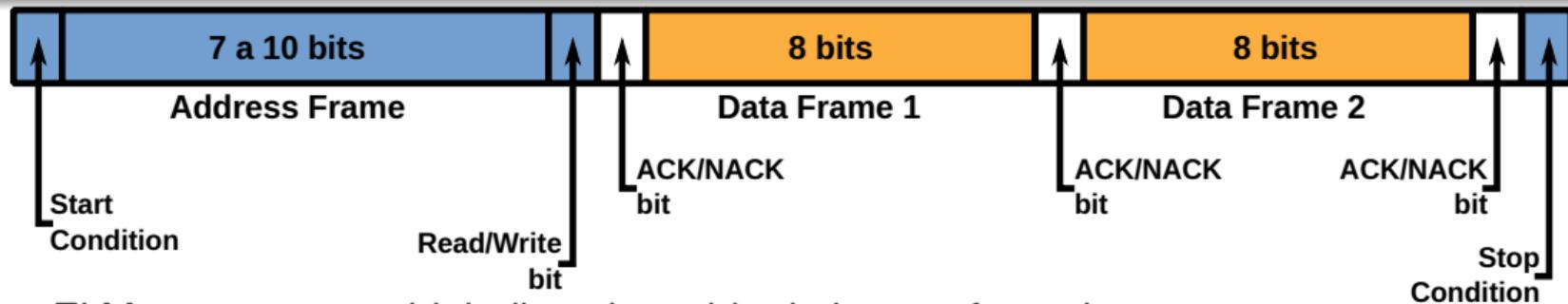
- El Master con este bit indica el sentido de la transferencia.

Read / Write Bit



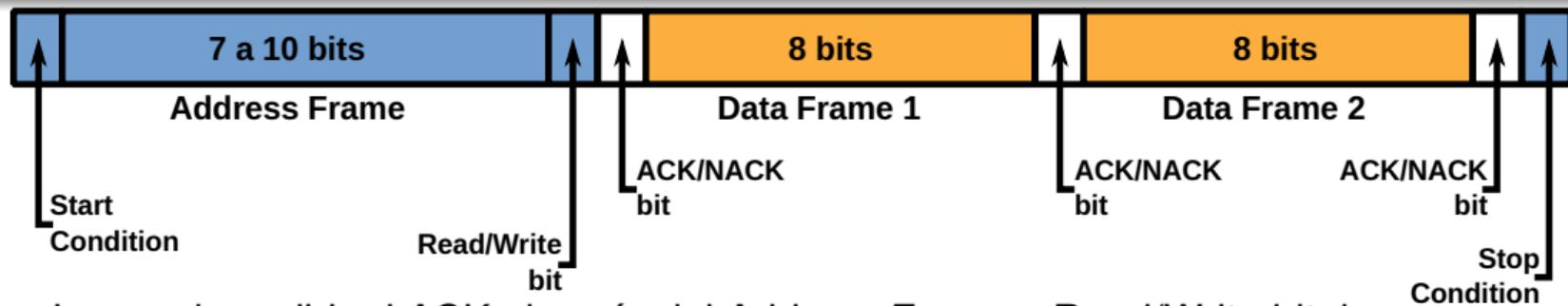
- El Master con este bit indica el sentido de la transferencia.
- Read corresponde a un nivel de tensión bajo.

Read / Write Bit



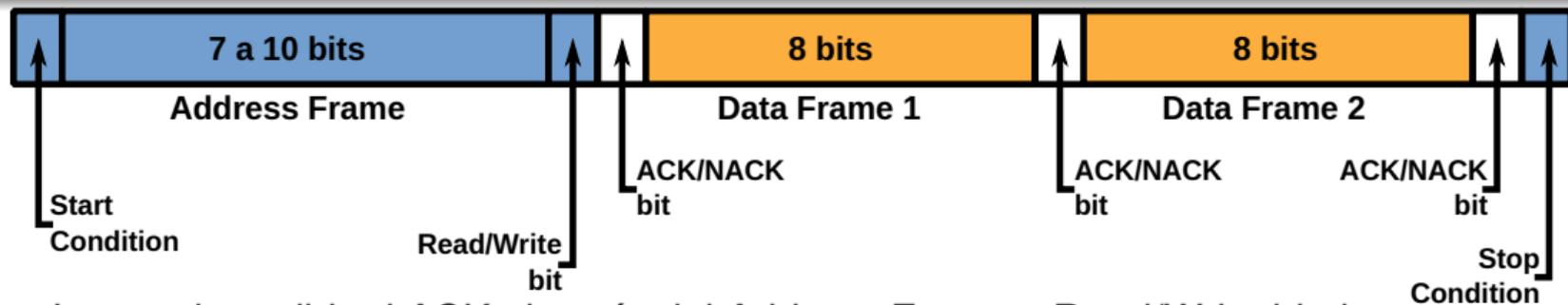
- El Master con este bit indica el sentido de la transferencia.
- Read corresponde a un nivel de tensión bajo.
- Write corresponde a un nivel de tensión alto.

El Data Frame



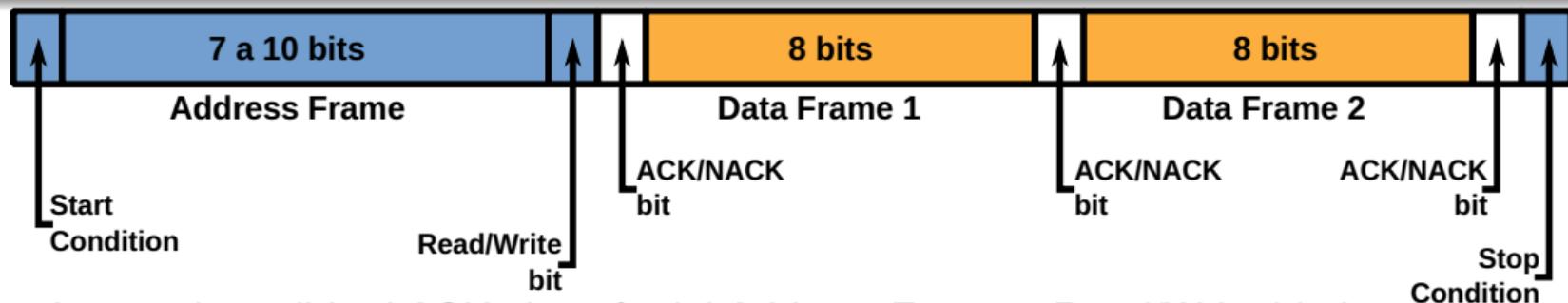
- Luego de recibir el ACK al envío del Address Frame y Read/Write bit, hay que empezar a transmitir.

El Data Frame



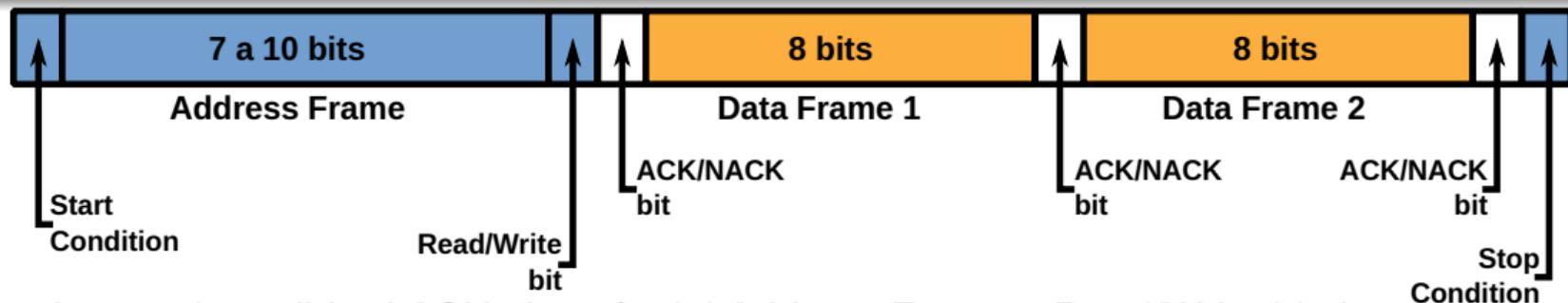
- Luego de recibir el ACK al envío del Address Frame y Read/Write bit, hay que empezar a transmitir.
- El data frame es una palabra de 8 bits (MSB primero) enviado por el transmisor, seguido de un ACK/NACK enviado por el receptor.

El Data Frame



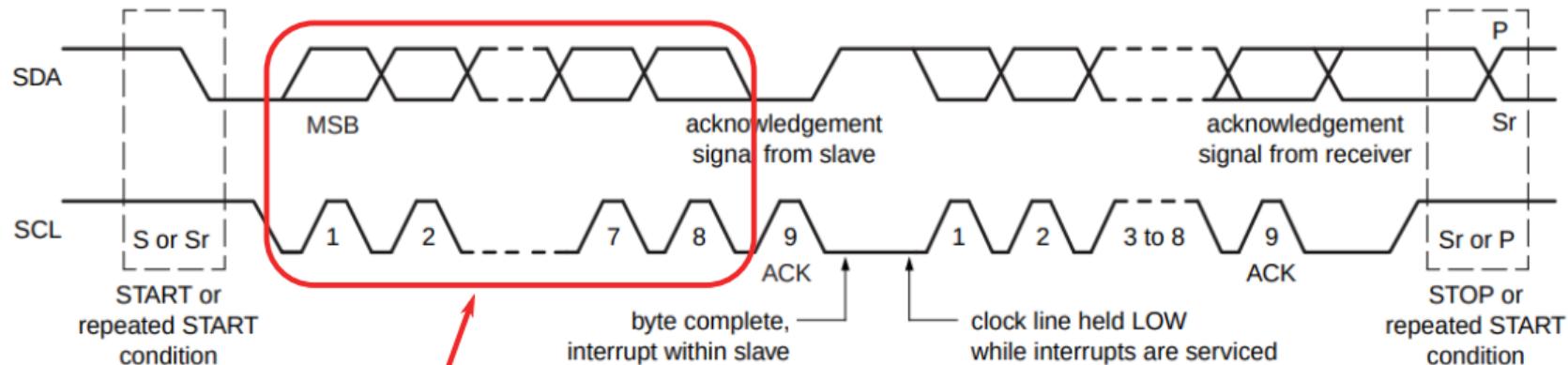
- Luego de recibir el ACK al envío del Address Frame y Read/Write bit, hay que empezar a transmitir.
- El data frame es una palabra de 8 bits (MSB primero) enviado por el transmisor, seguido de un ACK/NACK enviado por el receptor.
- Una vez finalizado el envío de todos los Data Frames requeridos para la transferencia, el Master envía el bit de Stop y finaliza la transmisión, llevando la línea **SDA** al estado alto de manera permanente.

El Data Frame



- Luego de recibir el ACK al envío del Address Frame y Read/Write bit, hay que empezar a transmitir.
- El data frame es una palabra de 8 bits (MSB primero) enviado por el transmisor, seguido de un ACK/NACK enviado por el receptor.
- Una vez finalizado el envío de todos los Data Frames requeridos para la transferencia, el Master envía el bit de Stop y finaliza la transmisión, llevando la línea **SDA** al estado alto de manera permanente.
- El estado de stop se asume luego del próximo cambio de estado bajo a estado alto de la línea **SCL** que también quedará en estado alto.

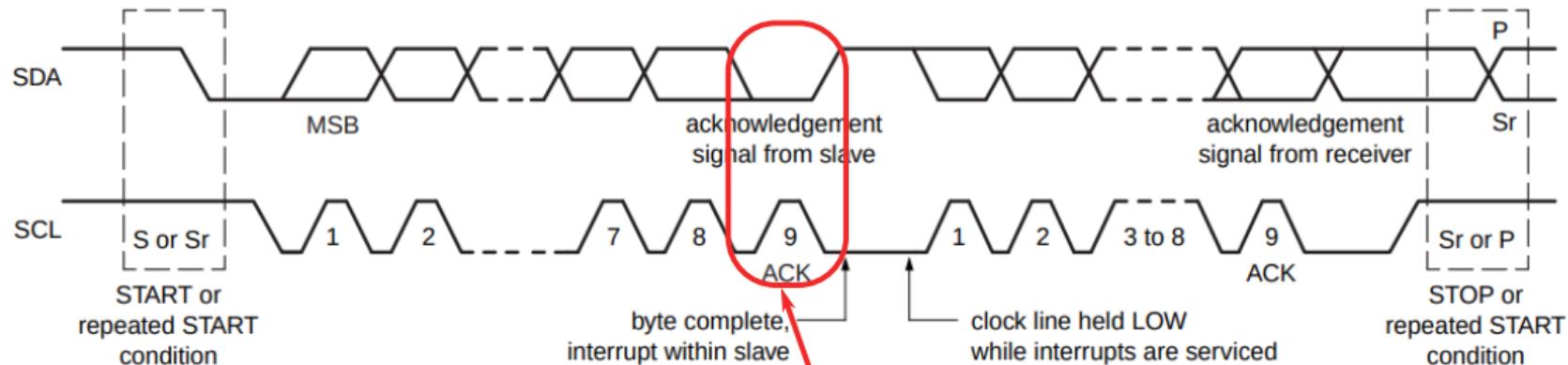
I2C. Trama de una comunicación



002aac861

El master envía a cada Slave el **Address Frame** de 7 o 10 bits con la dirección del Slave con el que quiere comunicarse, junto con el **Read/Write bit**

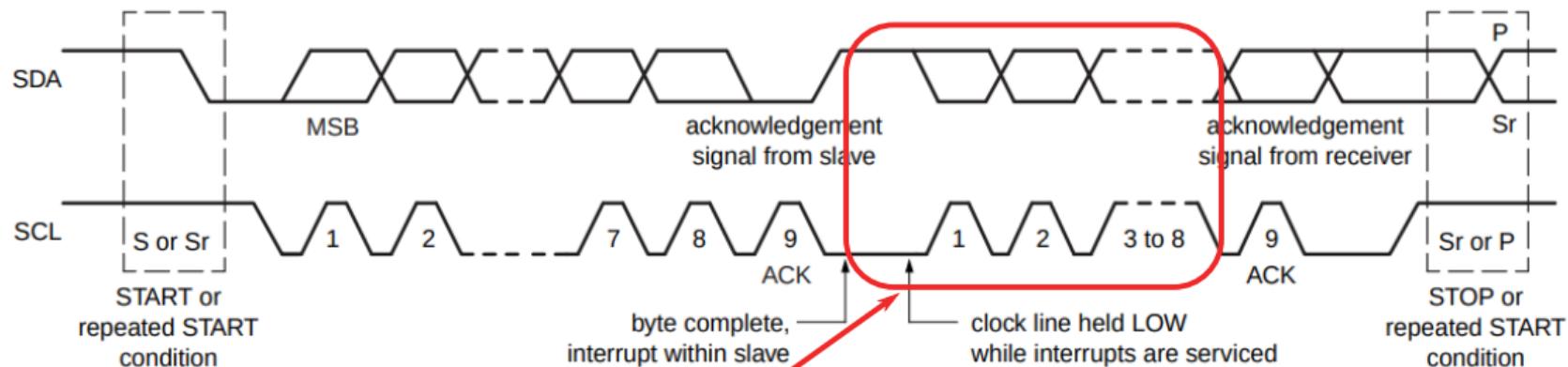
I2C. Trama de una comunicación



002aac861

Cada Slave compara la dirección enviada desde el Master con su propia dirección. Si coincide el Slave retorna un bit de **ACK** llevando la línea **SDA** a un nivel de tensión bajo durante un bit. Si la dirección enviada por el Master no coincide con la dirección del Slave, éste deja la línea **SDA** en un nivel de tensión alto.

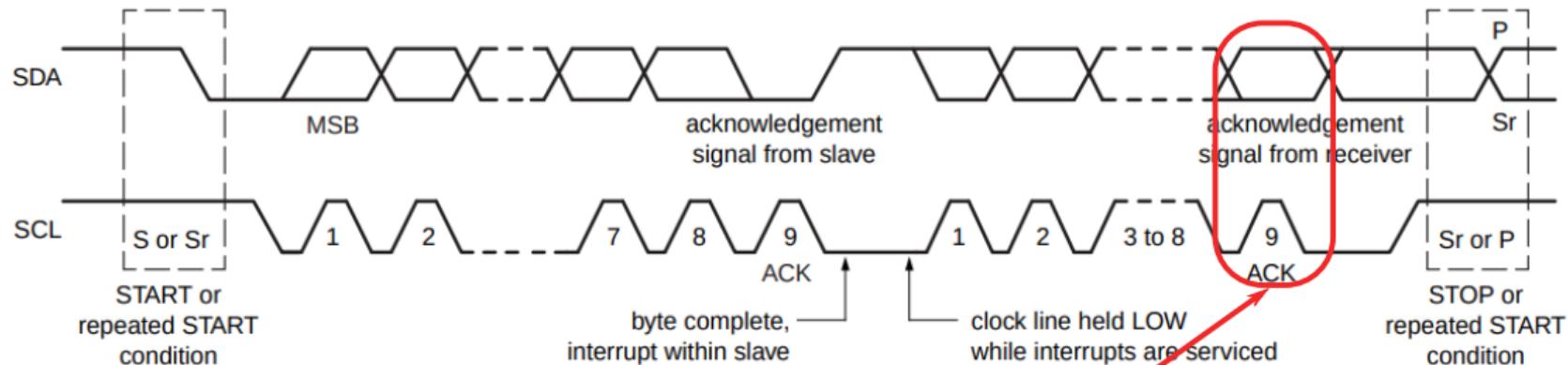
I2C. Trama de una comunicación



002aac861

El Master envía o recibe Data Frames

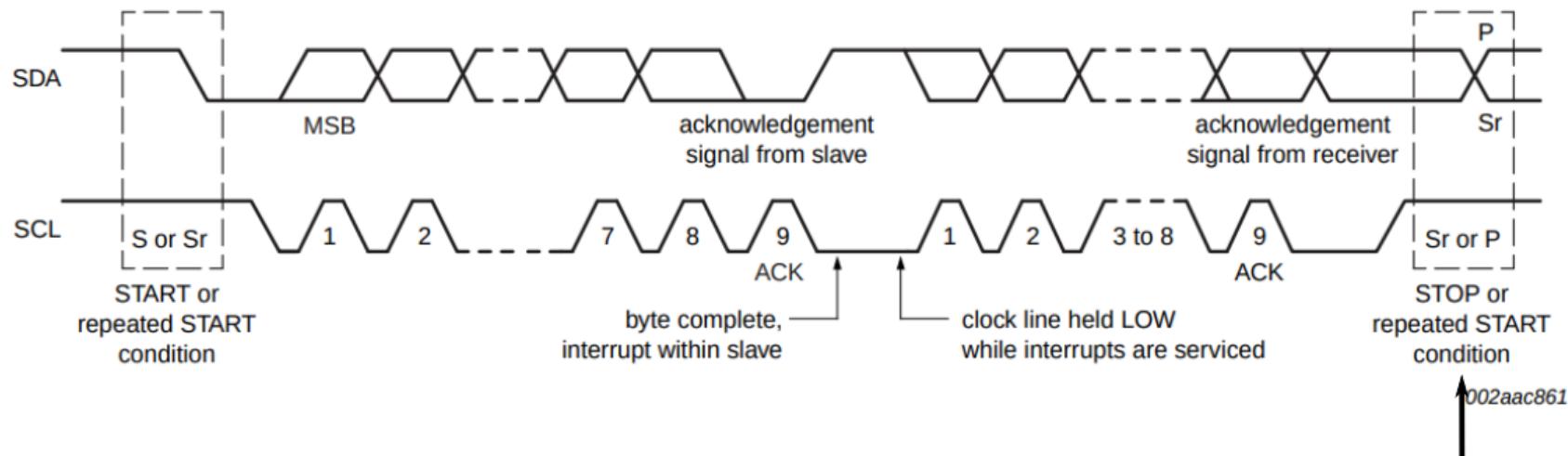
I2C. Trama de una comunicación



002aac861

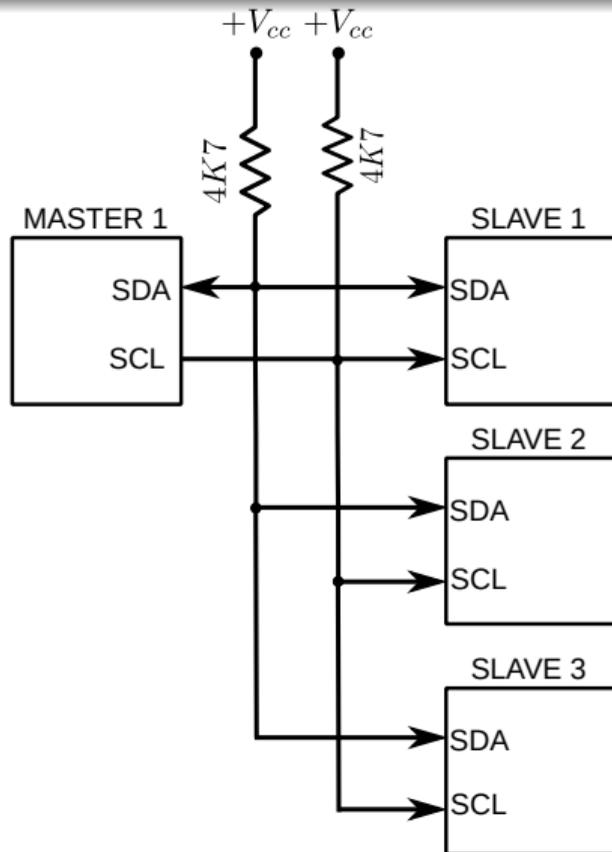
Luego de transferir cada **Data Frame**, el dispositivo receptor retorna un bit de **ACK** al transmisor para indicarle la recepción correcta del frame.

I2C. Trama de una comunicación

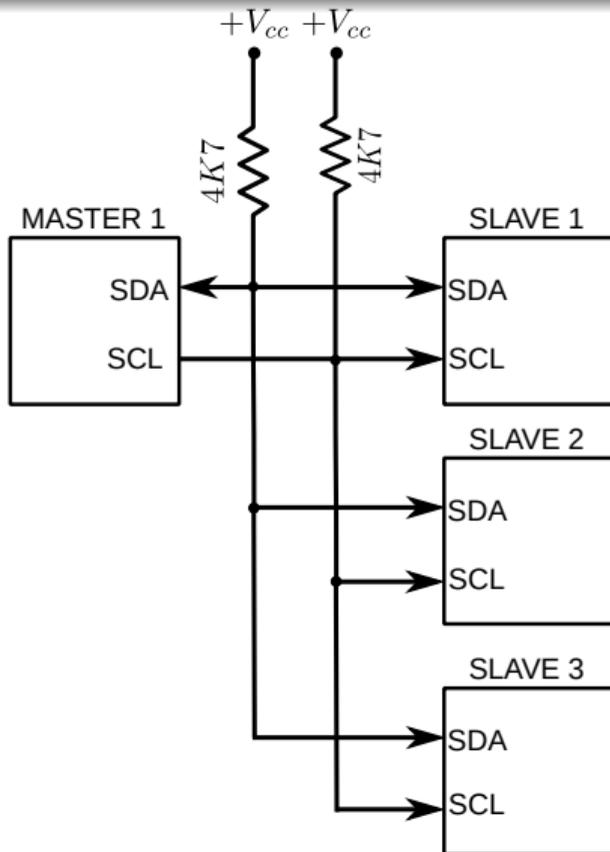


El Master envía el **Stop Condition** al Slave para terminar la transmisión conmutando la línea **SCL** a un estado de tensión alto antes de conmutar la línea **SDA** a un nivel de tensión alto.

I2C. Single Master Multi Slave

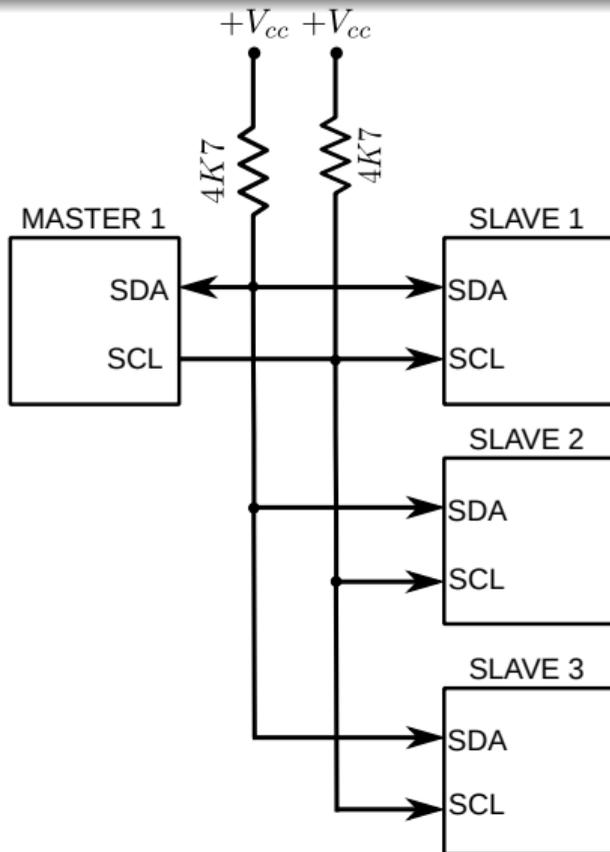


I2C. Single Master Multi Slave



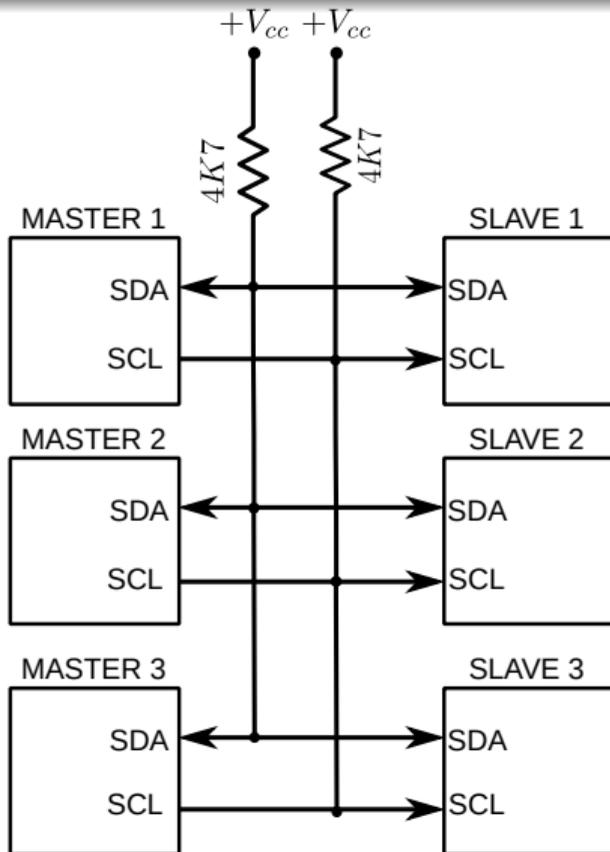
- Si se utiliza un esquema de direcciones de 7 bits es posible conectar a un único Master, $2^7 = 128$ Slaves.

I2C. Single Master Multi Slave

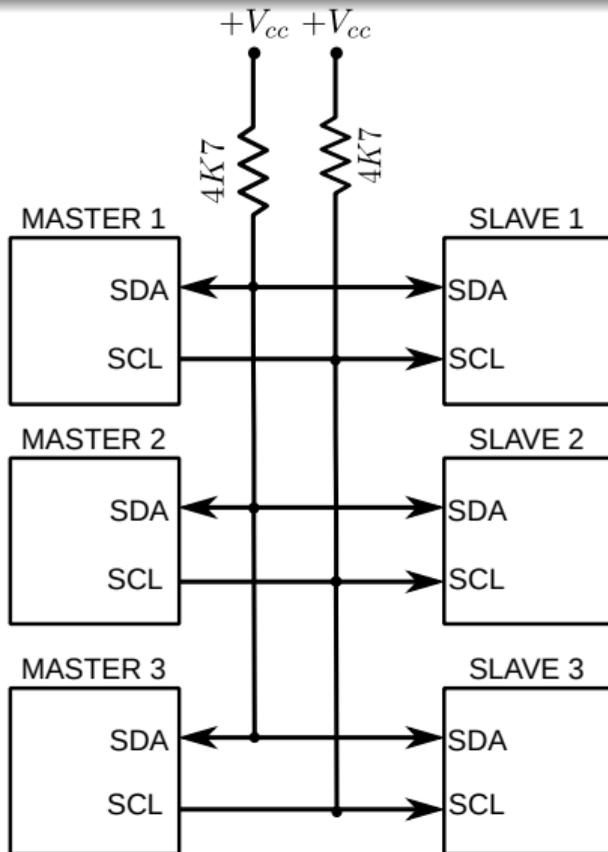


- Si se utiliza un esquema de direcciones de 7 bits es posible conectar a un único Master, $2^7 = 128$ Slaves.
- Si se utiliza un esquema de direcciones de 10 bits es posible conectar a un único Master, $2^{10} = 1024$ Slaves.

I2C. Multi Master Multi Slave

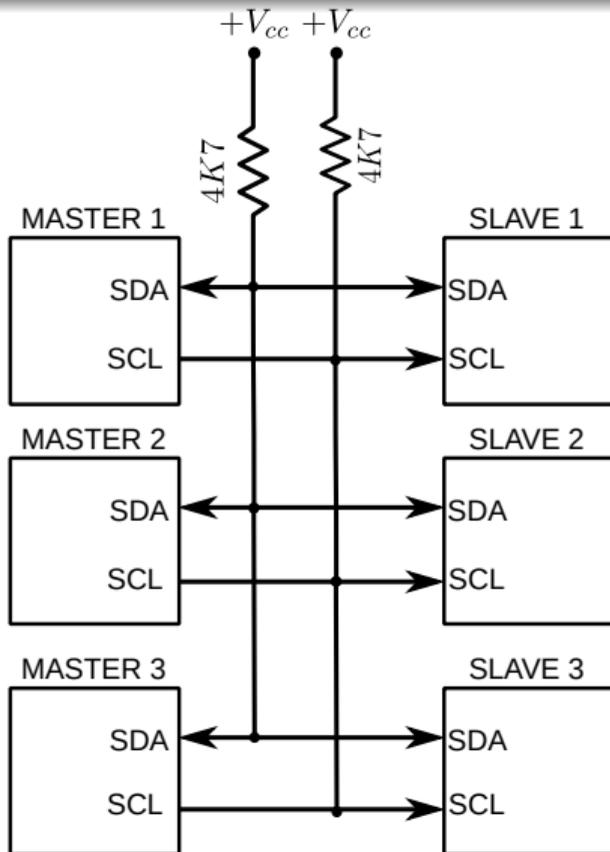


I2C. Multi Master Multi Slave



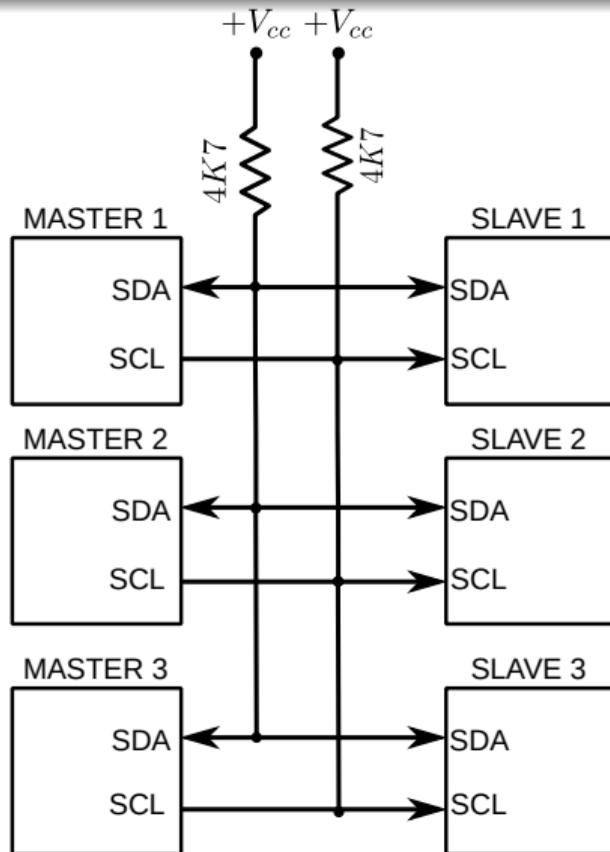
- El problema de usar múltiples Masters en el mismo sistema es cuando dos Masters tratan de iniciar una transferencia de datos sobre la misma línea **SDA**.

I2C. Multi Master Multi Slave



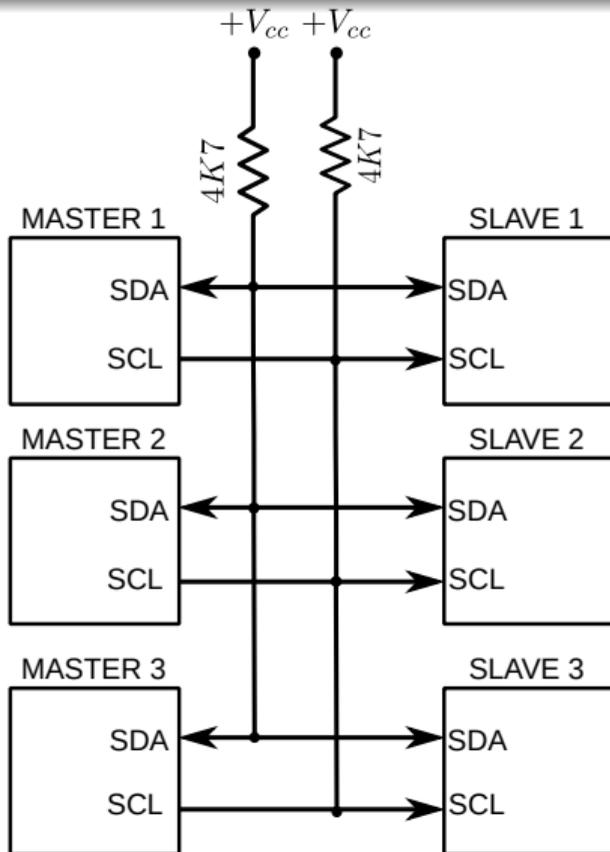
- El problema de usar múltiples Masters en el mismo sistema es cuando dos Masters tratan de iniciar una transferencia de datos sobre la misma línea **SDA**.
- Para resolver el escenario de acceso simultáneo, cada Master tiene que poder detectar si la línea **SDA** está en estado alto o bajo antes de transmitir.

I2C. Multi Master Multi Slave



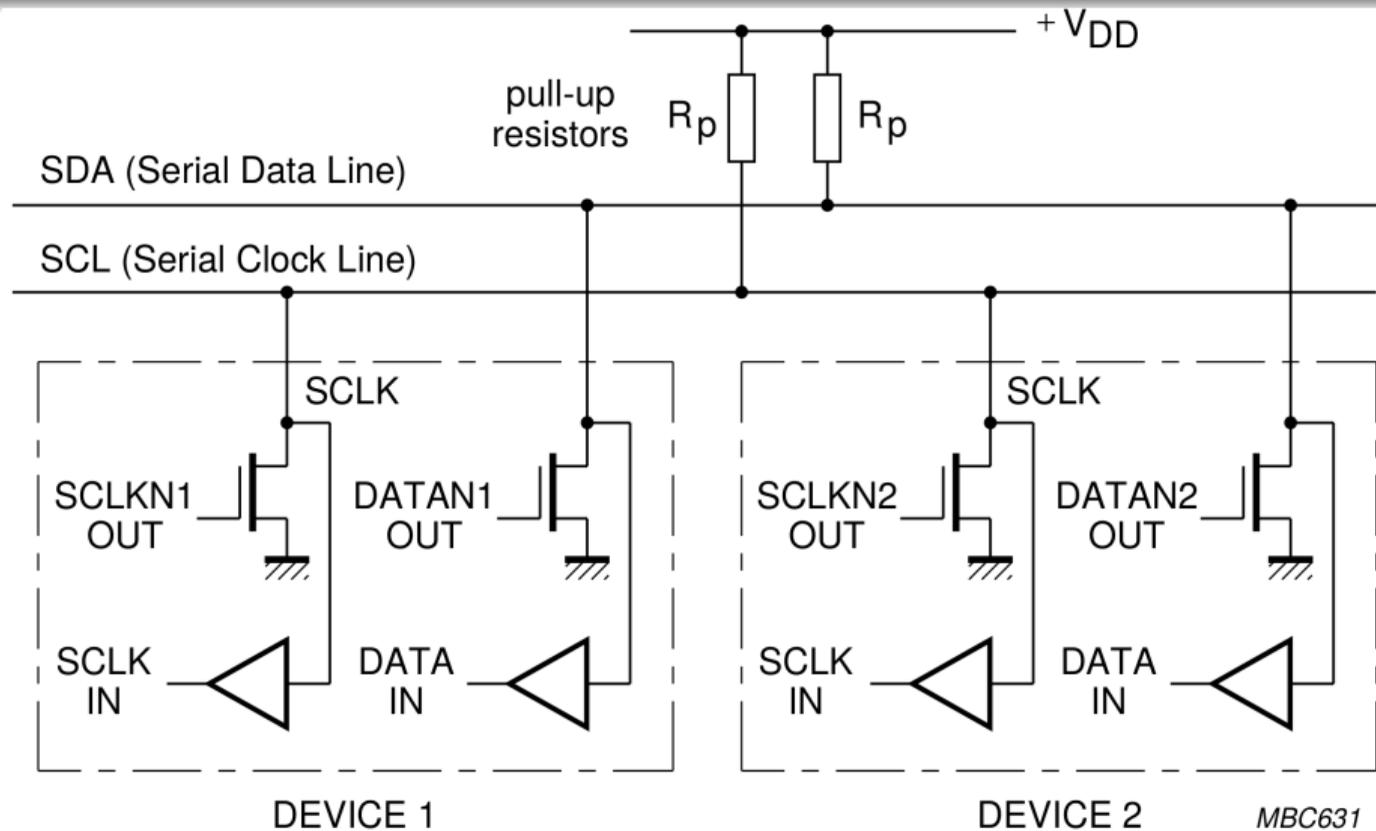
- El problema de usar múltiples Masters en el mismo sistema es cuando dos Masters tratan de iniciar una transferencia de datos sobre la misma línea **SDA**.
- Para resolver el escenario de acceso simultáneo, cada Master tiene que poder detectar si la línea **SDA** está en estado alto o bajo antes de transmitir.
- Si **SDA** está baja, significa que otro Master está en poder del Bus. El Master debe esperar.

I2C. Multi Master Multi Slave



- El problema de usar múltiples Masters en el mismo sistema es cuando dos Masters tratan de iniciar una transferencia de datos sobre la misma línea **SDA**.
- Para resolver el escenario de acceso simultáneo, cada Master tiene que poder detectar si la línea **SDA** está en estado alto o bajo antes de transmitir.
- Si **SDA** está baja, significa que otro Master está en poder del Bus. El Master debe esperar.
- Si la línea **SDA** está en estado alto, puede transmitir el mensaje.

Sensado de SDA. Configuración Wired-AND



MBC631

I2C Multimaster: Sincronización de Clock

I2C Multimaster: Sincronización de Clock

- Todos los Masters generan su propio reloj en la línea **SCL** para transferir mensajes en el bus I2C.

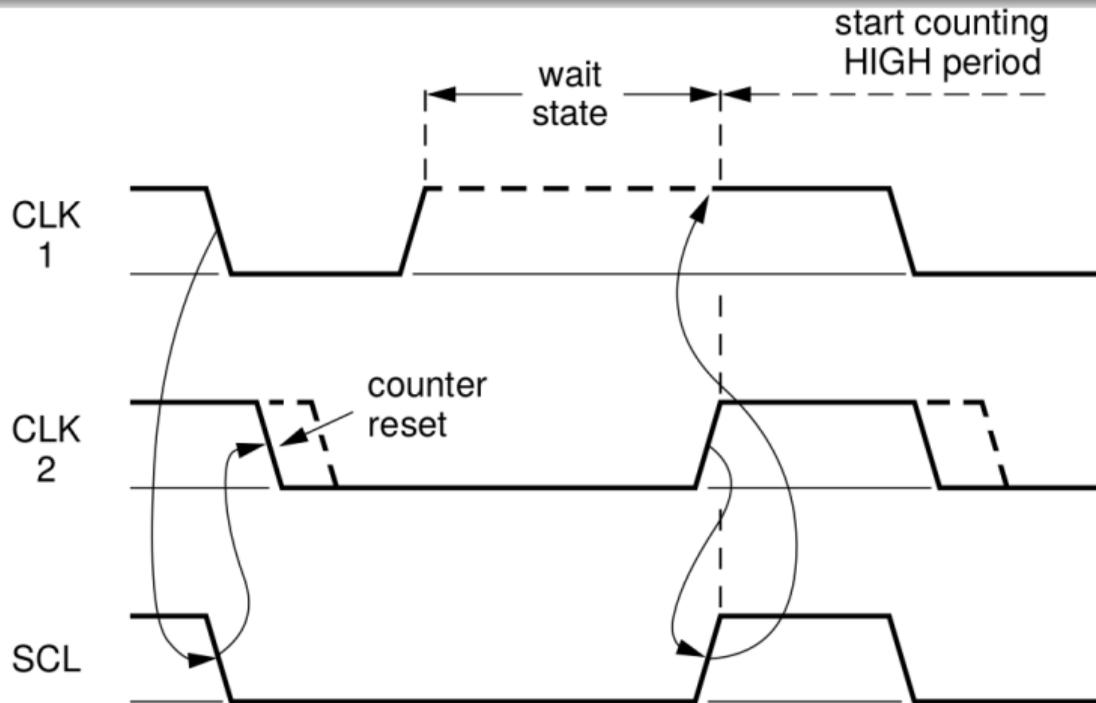
I2C Multimaster: Sincronización de Clock

- Todos los Masters generan su propio reloj en la línea **SCL** para transferir mensajes en el bus I2C.
- Los datos son válidos solo durante el período alto del clock.

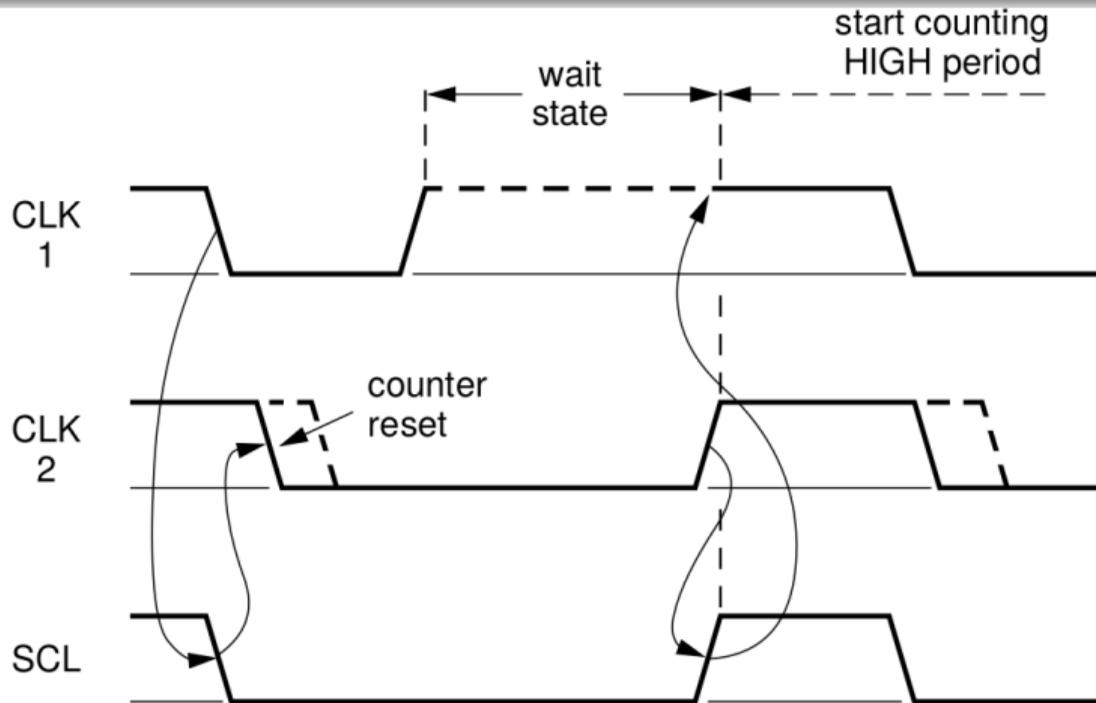
I2C Multimaster: Sincronización de Clock

- Todos los Masters generan su propio reloj en la línea **SCL** para transferir mensajes en el bus I2C.
- Los datos son válidos solo durante el período alto del clock.
- La sincronización del reloj se realiza mediante la conexión Wired-AND de la línea **SCL**.

I2C Multimaster: Sincronización de Clock

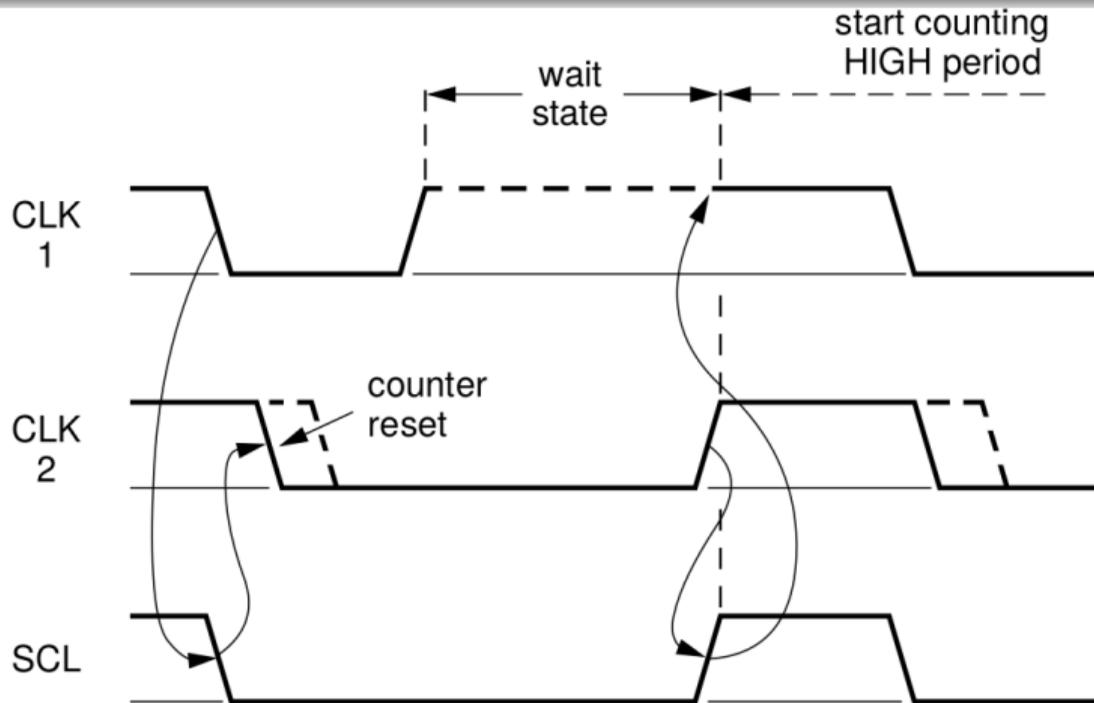


I2C Multimaster: Sincronización de Clock



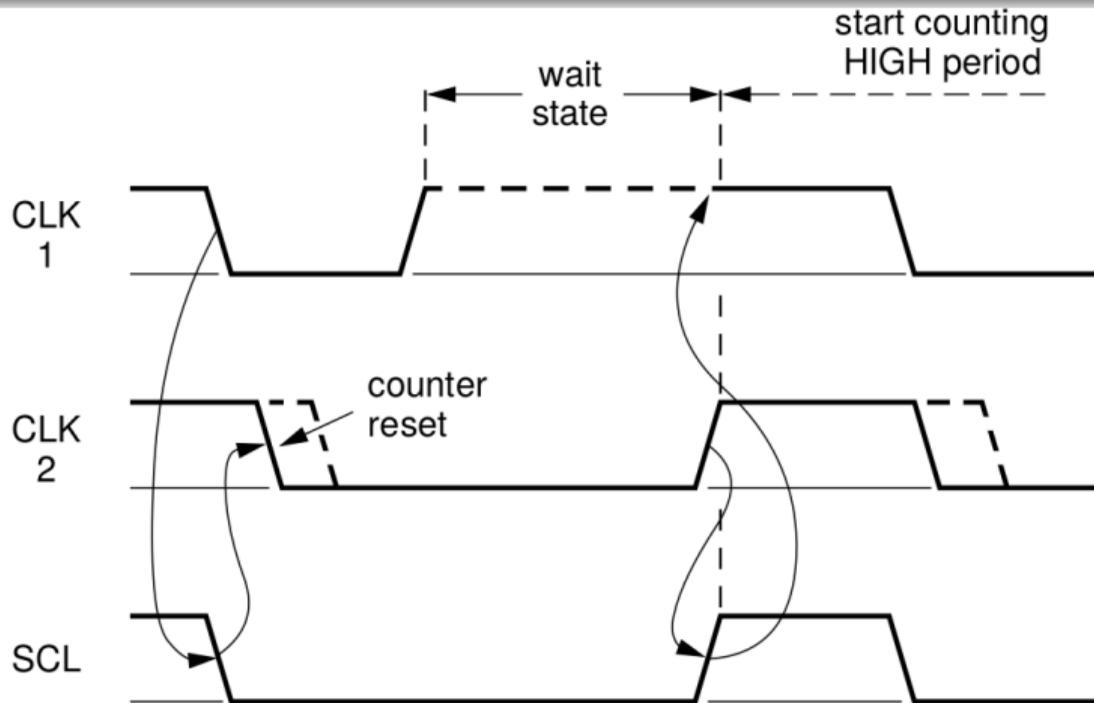
Cuando un Master realiza una transición del estado de tensión alto a un estado de tensión bajo, los dispositivos en cuestión comienzan a contar su período Bajo.

I2C Multimaster: Sincronización de Clock



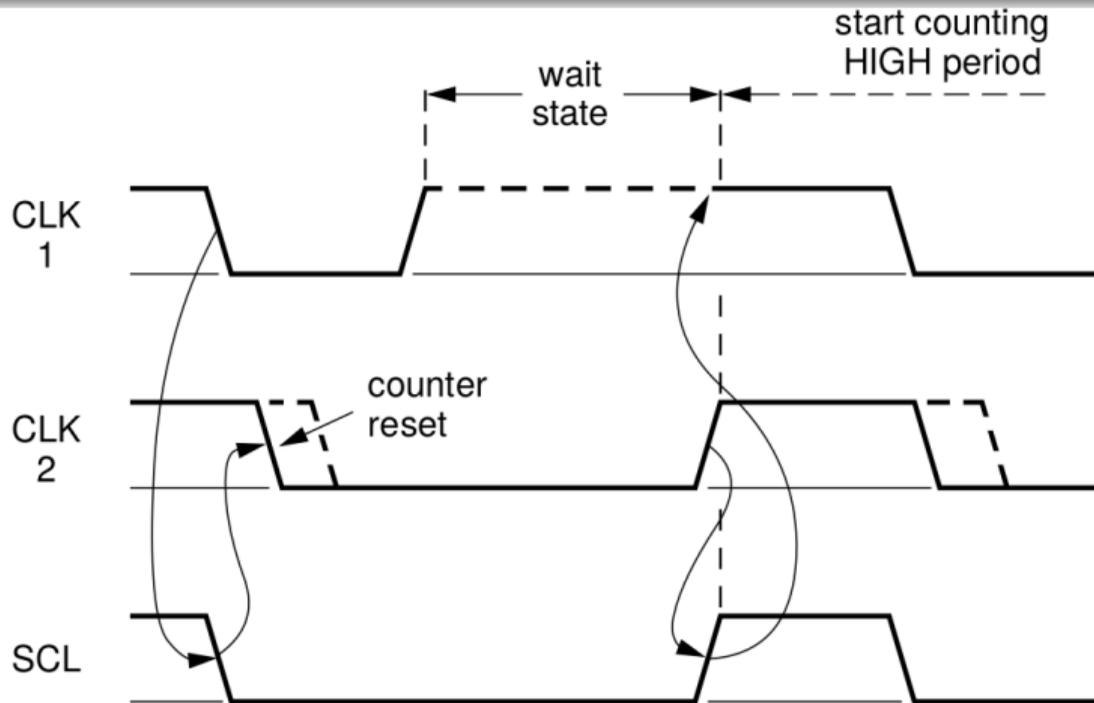
Cuando este reloj genere la transición nuevamente al estado de tensión alto, si otro reloj está todavía dentro de su período bajo, no cambiará el estado de la línea **SCL**.

I2C Multimaster: Sincronización de Clock



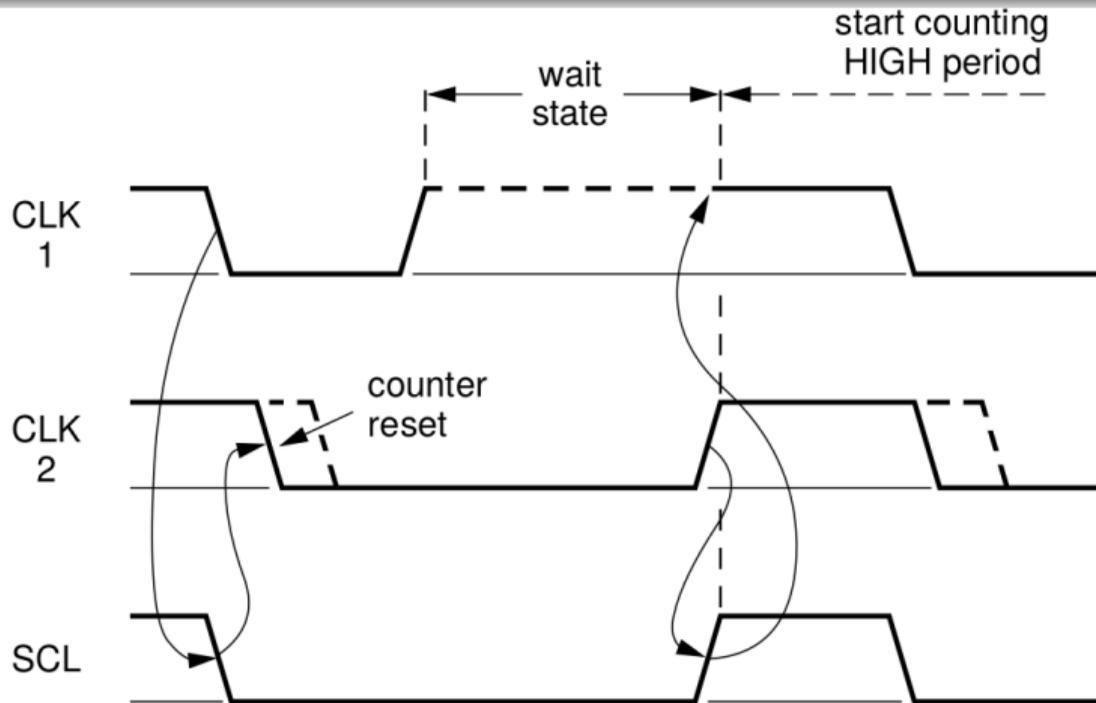
La línea **SCL** del bus se mantendrá en estado bajo debido al dispositivo con el período Bajo más largo.

I2C Multimaster: Sincronización de Clock



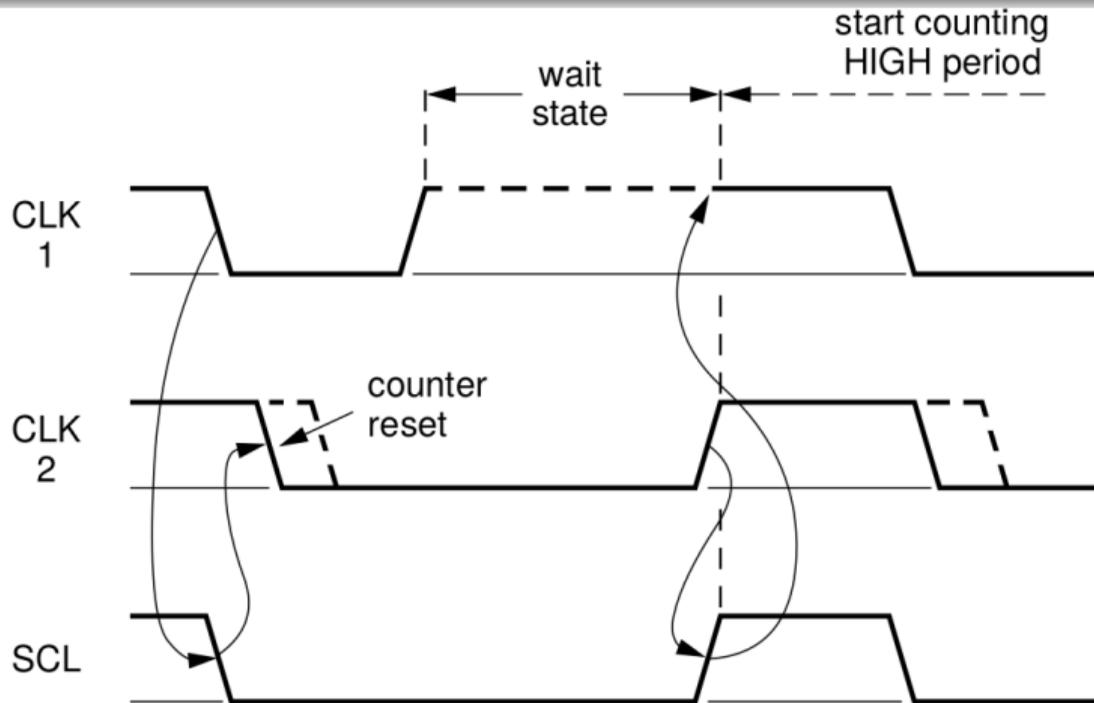
Los dispositivos con períodos Bajos más cortos entran en un Wait State alto durante este tiempo.

I2C Multimaster: Sincronización de Clock



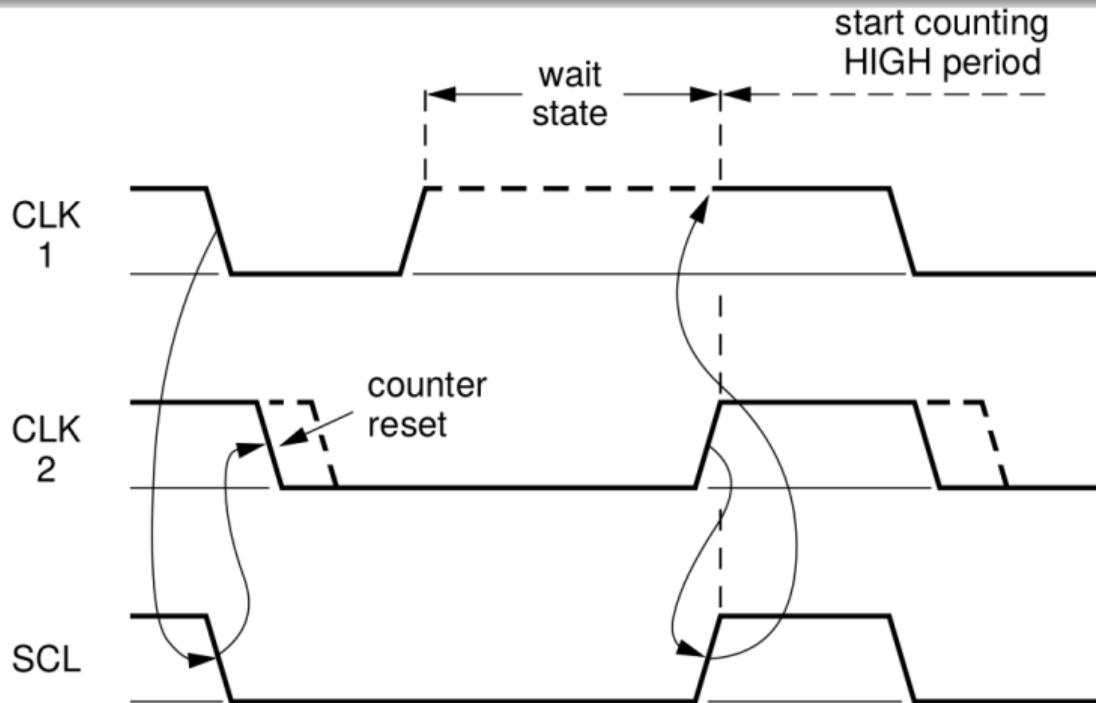
Cuando todos los dispositivos hayan contado este período bajo, la línea del reloj se liberará y pasará a alto.

I2C Multimaster: Sincronización de Clock



Entonces no habrá diferencia entre los relojes de los dispositivos y el estado de la línea **SCL**.

I2C Multimaster: Sincronización de Clock



Todos los dispositivos comenzarán a contar su período alto. El primer dispositivo en completar su período alto volverá a llevar la línea **SCL** a bajo.

I2C Multimaster: Arbitración

I2C Multimaster: Arbitración

- Un Master puede iniciar una transferencia solo si el bus está libre. Dos o más maestros pueden generar una Start Condition dentro del tiempo mínimo de retención ($t_{HD;STA}$) que da como resultado una Start Condition definida para el bus.

I2C Multimaster: Arbitración

- Un Master puede iniciar una transferencia solo si el bus está libre. Dos o más maestros pueden generar una Start Condition dentro del tiempo mínimo de retención ($t_{HD;STA}$) que da como resultado una Start Condition definida para el bus.
- El arbitraje se lleva a cabo en la línea **SDA**, mientras que la línea **SCL** está en el nivel alto, de tal manera que el Master que transmite un nivel alto, mientras otro Master está transmitiendo un nivel Bajo apagará su etapa de salida de datos porque el nivel en el bus no corresponde a su propio nivel.

I2C Multimaster: Arbitración

- Un Master puede iniciar una transferencia solo si el bus está libre. Dos o más maestros pueden generar una Start Condition dentro del tiempo mínimo de retención ($t_{HD;STA}$) que da como resultado una Start Condition definida para el bus.
- El arbitraje se lleva a cabo en la línea **SDA**, mientras que la línea **SCL** está en el nivel alto, de tal manera que el Master que transmite un nivel alto, mientras otro Master está transmitiendo un nivel Bajo apagará su etapa de salida de datos porque el nivel en el bus no corresponde a su propio nivel.
- El arbitraje continuará en diversos momentos.

I2C Multimaster: Arbitración

- Un Master puede iniciar una transferencia solo si el bus está libre. Dos o más maestros pueden generar una Start Condition dentro del tiempo mínimo de retención ($t_{HD;STA}$) que da como resultado una Start Condition definida para el bus.
- El arbitraje se lleva a cabo en la línea **SDA**, mientras que la línea **SCL** está en el nivel alto, de tal manera que el Master que transmite un nivel alto, mientras otro Master está transmitiendo un nivel Bajo apagará su etapa de salida de datos porque el nivel en el bus no corresponde a su propio nivel.
- El arbitraje continuará en diversos momentos.
- La primera etapa es la comparación de los bits de dirección.

I2C Multimaster: Arbitración

- Un Master puede iniciar una transferencia solo si el bus está libre. Dos o más maestros pueden generar una Start Condition dentro del tiempo mínimo de retención ($t_{HD;STA}$) que da como resultado una Start Condition definida para el bus.
- El arbitraje se lleva a cabo en la línea **SDA**, mientras que la línea **SCL** está en el nivel alto, de tal manera que el Master que transmite un nivel alto, mientras otro Master está transmitiendo un nivel Bajo apagará su etapa de salida de datos porque el nivel en el bus no corresponde a su propio nivel.
- El arbitraje continuará en diversos momentos.
- La primera etapa es la comparación de los bits de dirección.
- Si cada uno de los Masters está tratando de direccionar al mismo dispositivo, el arbitraje continúa con la comparación de los bits de datos si son Master-transmit, o bits de reconocimiento si son Master-receive.

I2C Multimaster: Arbitración

- Debido a que la dirección y la información de datos en el bus I2C está determinada por el Master ganador, no se pierde ninguna información durante el proceso de arbitraje.

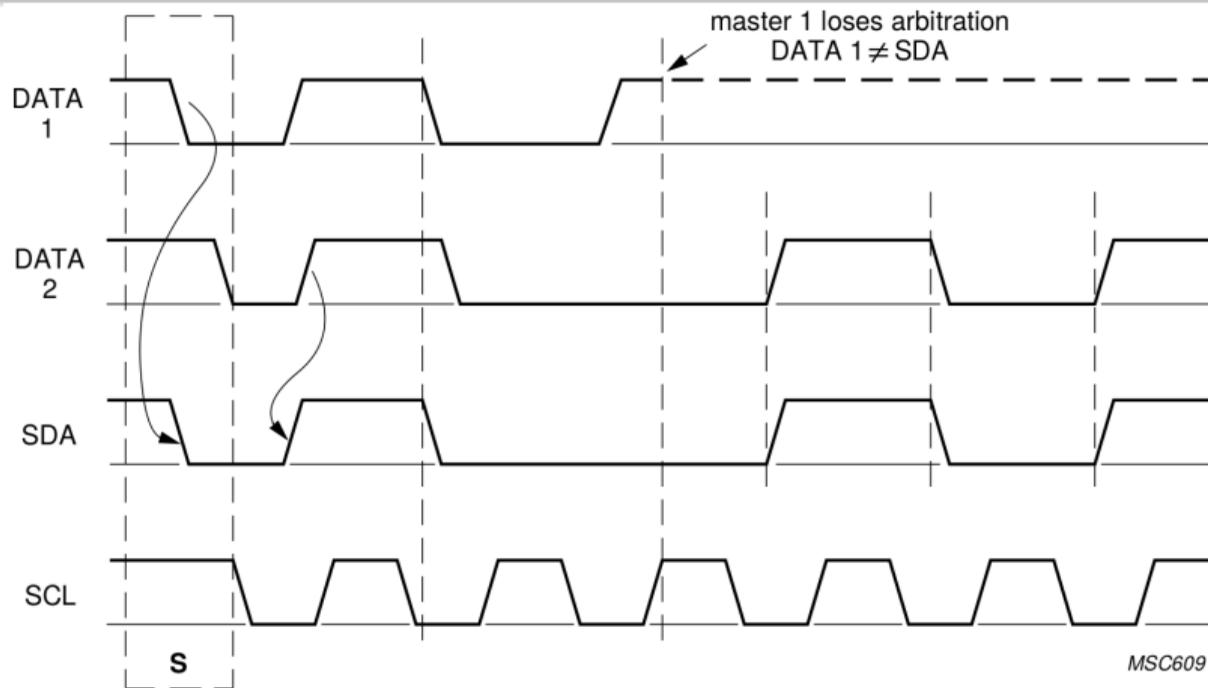
I2C Multimaster: Arbitración

- Debido a que la dirección y la información de datos en el bus I2C está determinada por el Master ganador, no se pierde ninguna información durante el proceso de arbitraje.
- Como un Master en modo Hs (High Speed) tiene un Master code único de 8 bits, siempre finalizará el arbitraje durante el primer byte.

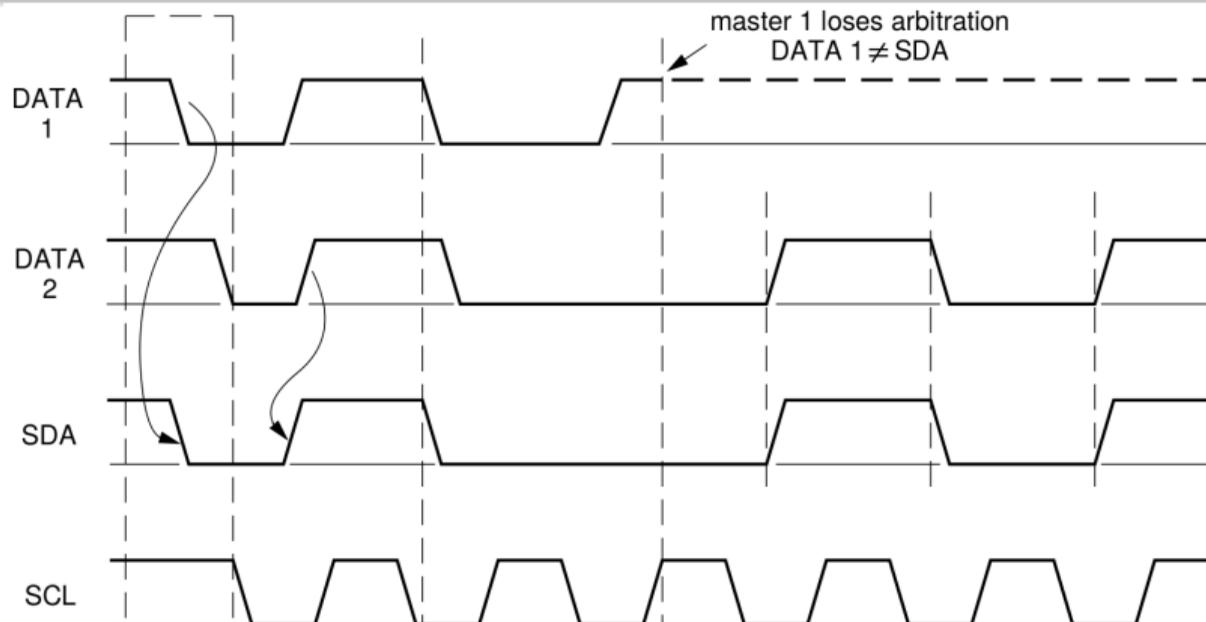
I2C Multimaster: Arbitración

- Debido a que la dirección y la información de datos en el bus I2C está determinada por el Master ganador, no se pierde ninguna información durante el proceso de arbitraje.
- Como un Master en modo Hs (High Speed) tiene un Master code único de 8 bits, siempre finalizará el arbitraje durante el primer byte.
- Si un Master también incorpora una función de Slave y pierde el arbitraje durante la etapa de direccionamiento, asume que el Master ganador esté tratando de abordarlo. Por lo tanto, el Master perdedor debe cambiar inmediatamente a su modo Slave.

I2C Multimaster: Arbitración

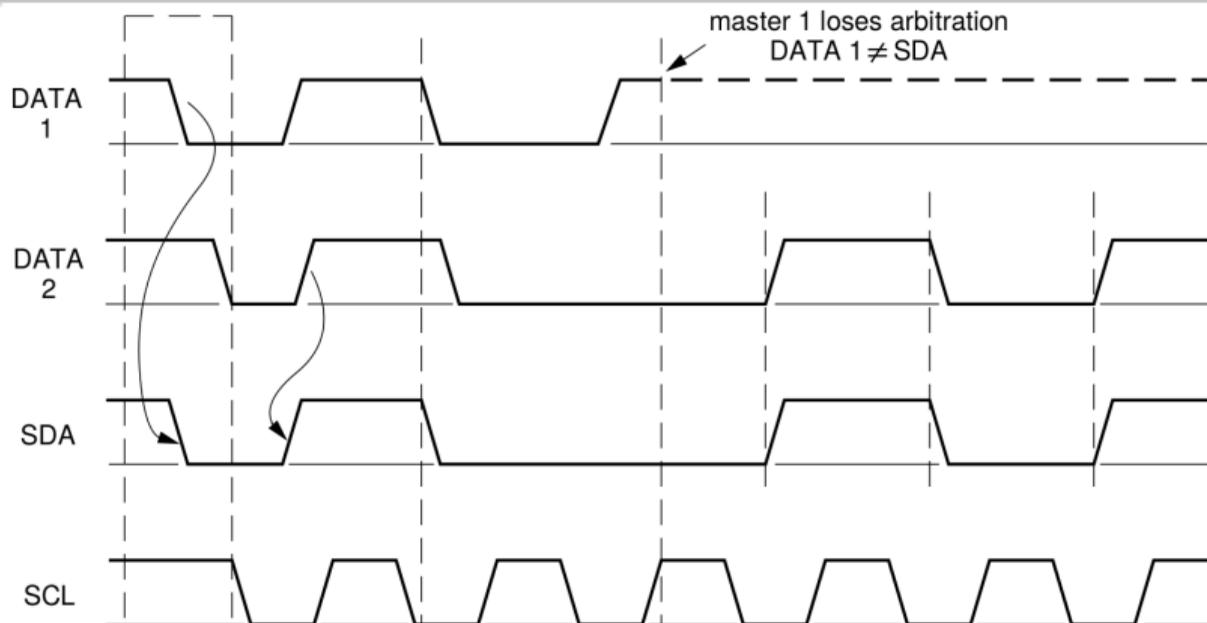


I2C Multimaster: Arbitración



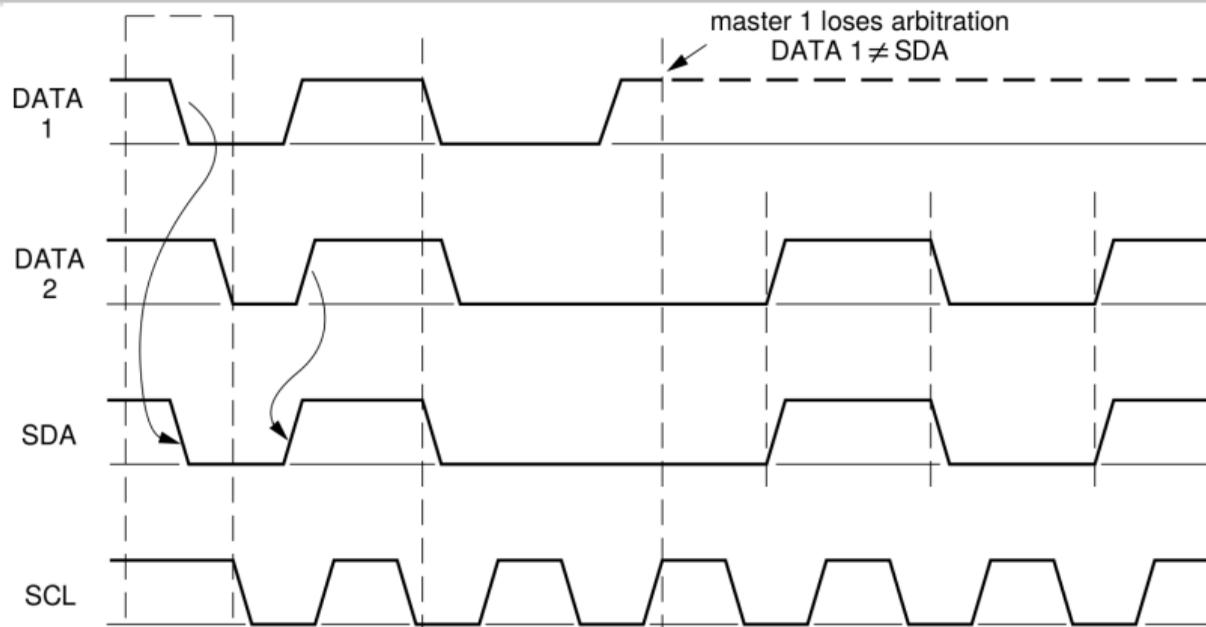
En el momento en que hay una diferencia entre el nivel de datos internos del Master que genera **DATOS 1** y el nivel real en la línea **SDA**, su salida de datos se apaga, lo que significa que se lleva el bus a un nivel de salida alto. Esto no afectará la transferencia de datos iniciada por el Master ganador.

I2C Multimaster: Arbitración



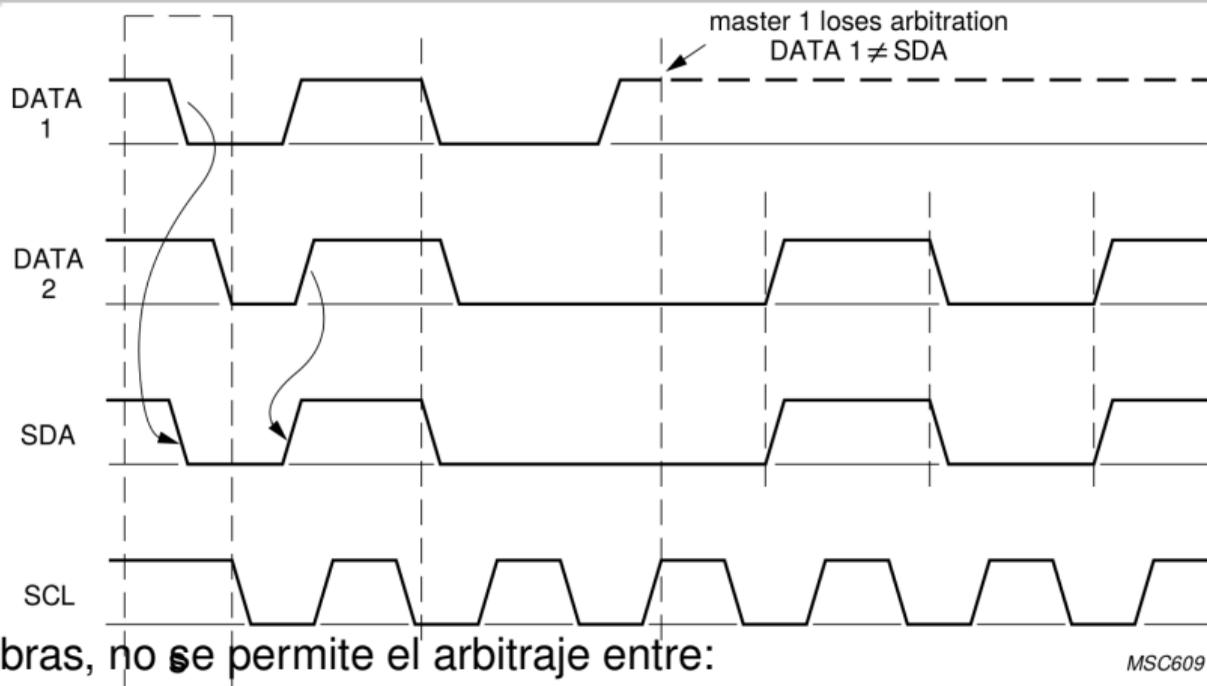
Dado que el control del bus I2C se decide únicamente sobre la dirección o el Master Code y los datos enviados por Masters competidores, no hay un Master central ni ningún orden de prioridad en el bus.

I2C Multimaster: Arbitración

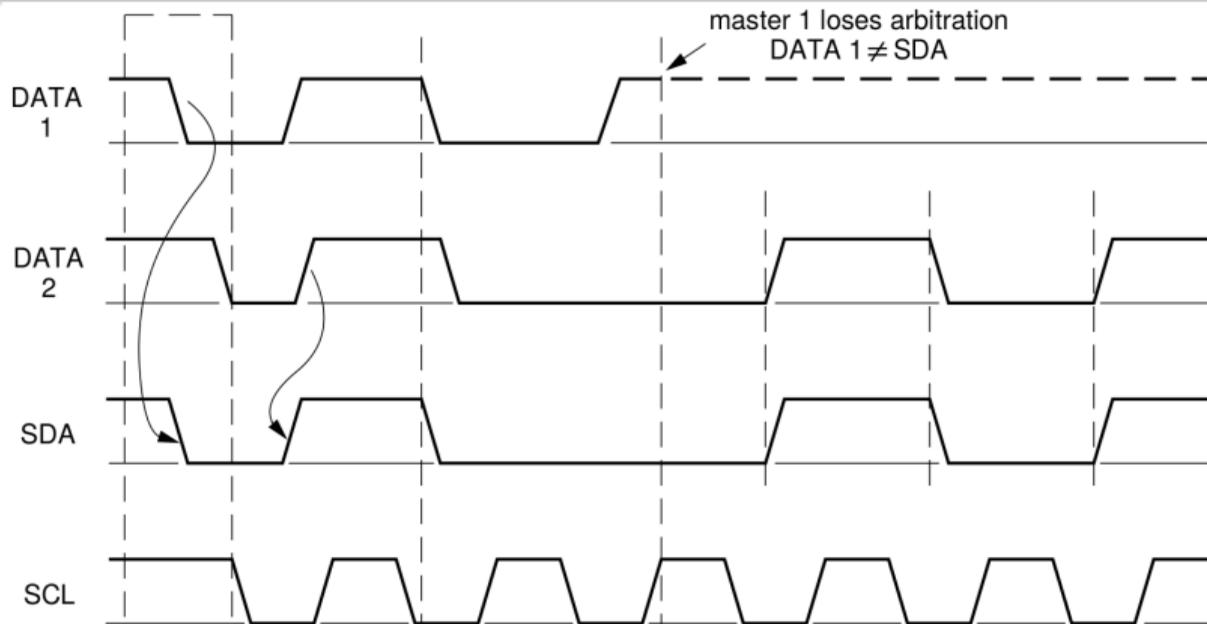


Si durante una transferencia, el procedimiento de arbitraje aún está en curso cuando se transmite una Start Condition repetida o una Stop Condition al bus, los Masters involucrados deben enviar esta Start o Stop Condition en la misma posición establecida en el formato del frame.

I2C Multimaster: Arbitración



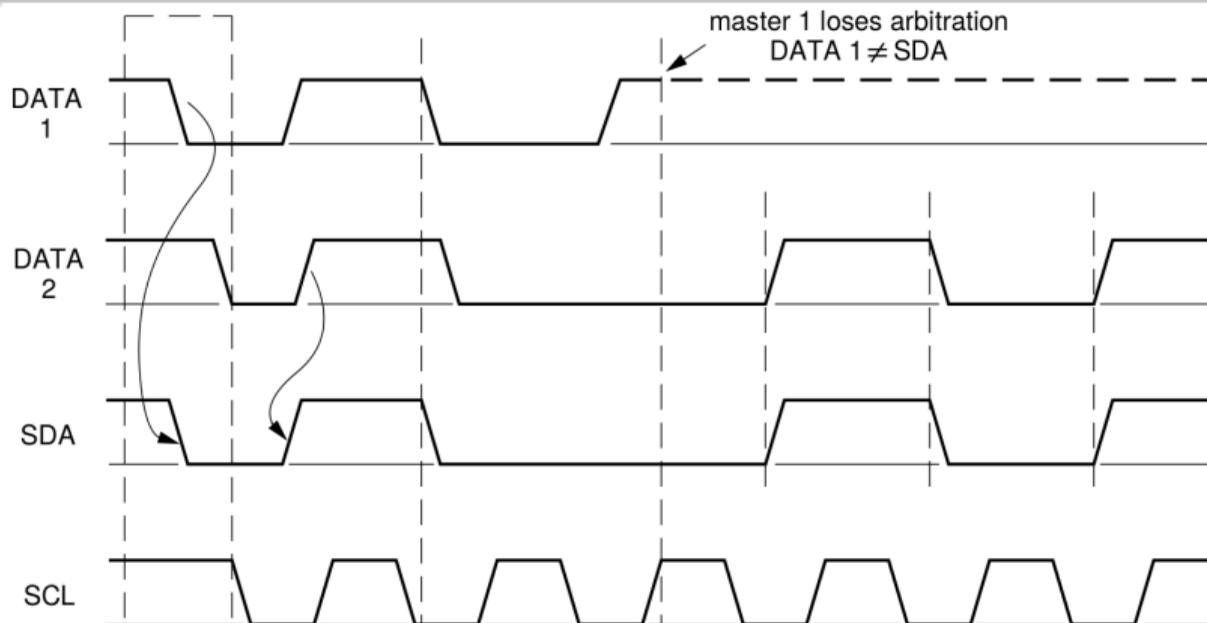
I2C Multimaster: Arbitración



MSC609

En otras palabras, no se permite el arbitraje entre:
✓ Una Start Condition repetida y un data bit.

I2C Multimaster: Arbitración

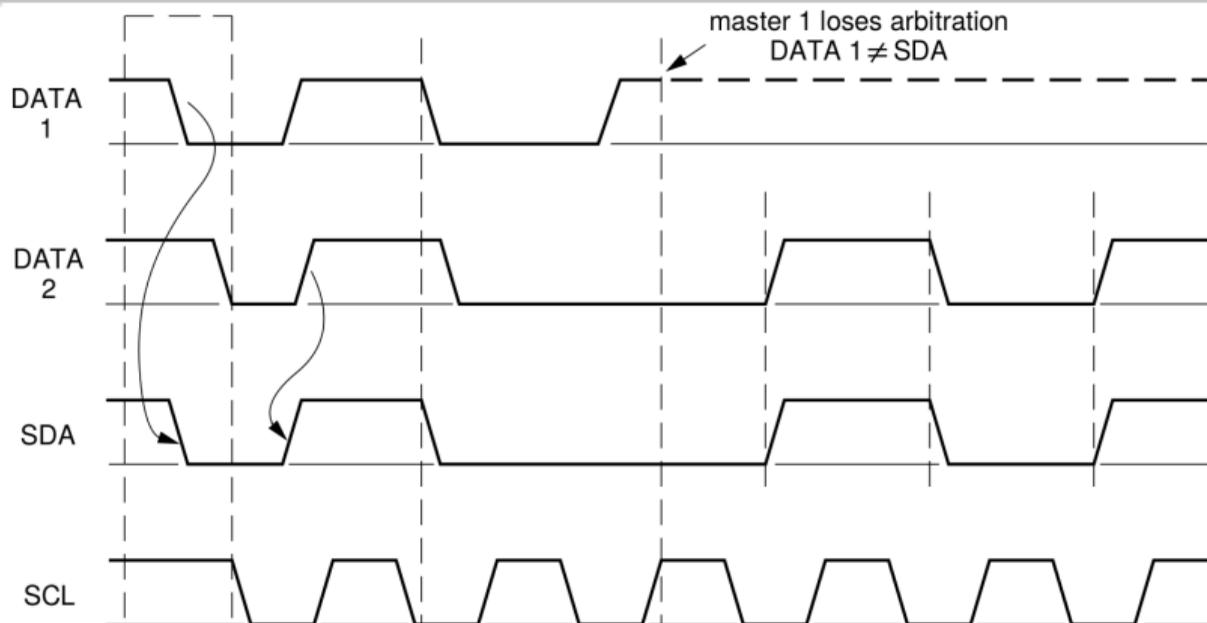


MSC609

En otras palabras, no se permite el arbitraje entre:

- ✓ Una Start Condition repetida y un data bit.
- ✓ Una Stop Condition y un data bit.

I2C Multimaster: Arbitración



MSC609

En otras palabras, no se permite el arbitraje entre:

- ✓ Una Start Condition repetida y un data bit.
- ✓ Una Stop Condition y un data bit.
- ✓ Una Start Condition repetida y un Stop Condition.

I2C - Conclusiones

Ventajas

I2C - Conclusiones

Ventajas

- Solo dos alambres.

I2C - Conclusiones

Ventajas

- Solo dos alambres.
- Soporta Múltiples Masters y Slaves

I2C - Conclusiones

Ventajas

- Solo dos alambres.
- Soporta Múltiples Masters y Slaves
- Bit ACK/NACK, asegura confirmación de recepción exitosa.

I2C - Conclusiones

Ventajas

- Solo dos alambres.
- Soporta Múltiples Masters y Slaves
- Bit ACK/NACK, asegura confirmación de recepción exitosa.
- Hardware mas simple que el de una UART

I2C - Conclusiones

Ventajas

- Solo dos alambres.
- Soporta Múltiples Masters y Slaves
- Bit ACK/NACK, asegura confirmación de recepción exitosa.
- Hardware mas simple que el de una UART
- Protocolo ampliamente utilizado.

I2C - Conclusiones

Ventajas

- Solo dos alambres.
- Soporta Múltiples Masters y Slaves
- Bit ACK/NACK, asegura confirmación de recepción exitosa.
- Hardware mas simple que el de una UART
- Protocolo ampliamente utilizado.

Desventajas

I2C - Conclusiones

Ventajas

- Solo dos alambres.
- Soporta Múltiples Masters y Slaves
- Bit ACK/NACK, asegura confirmación de recepción exitosa.
- Hardware mas simple que el de una UART
- Protocolo ampliamente utilizado.

Desventajas

- Menor velocidad respecto de SPI.

I2C - Conclusiones

Ventajas

- Solo dos alambres.
- Soporta Múltiples Masters y Slaves
- Bit ACK/NACK, asegura confirmación de recepción exitosa.
- Hardware mas simple que el de una UART
- Protocolo ampliamente utilizado.

Desventajas

- Menor velocidad respecto de SPI.
- Tamaño de datos limitado a 8 bits.

I2C - Conclusiones

Ventajas

- Solo dos alambres.
- Soporta Múltiples Masters y Slaves
- Bit ACK/NACK, asegura confirmación de recepción exitosa.
- Hardware mas simple que el de una UART
- Protocolo ampliamente utilizado.

Desventajas

- Menor velocidad respecto de SPI.
- Tamaño de datos limitado a 8 bits.
- Hardware de implementación mas complicado que SPI.

Temario

- 1 Introducción
 - Motivación
 - Primeros estándares
- 2 SPI
 - Introducción
 - Funcionamiento
 - Implementación SPI en el SoC Sitara AM3358 (BBB)
- 3 **I2C**
 - Introducción
 - Funcionamiento
 - **Implementación I2C en el SoC Sitara AM3358 (BBB)**
- 4 Otros protocolos serie
 - Controller Area Network (CAN)
 - 1-Wire

Implementación

Tres instancias I2C en modo Multimaster

Implementación

Tres instancias I2C en modo Multimaster

- Compatible con la especificación I2C version 2.1 de Philips.

Implementación

Tres instancias I2C en modo Multimaster

- Compatible con la especificación I2C version 2.1 de Philips.
- Suporta Modo standard (hasta 100 kbit s) y Modo fast (hasta 400 kbit s).

Implementación

Tres instancias I2C en modo Multimaster

- Compatible con la especificación I2C version 2.1 de Philips.
- Soporta Modo standard (hasta 100 kbit s) y Modo fast (hasta 400 kbit s).
- Modo Multimaster transmitter/Slave receiver.

Implementación

Tres instancias I2C en modo Multimaster

- Compatible con la especificación I2C version 2.1 de Philips.
- Soporta Modo standard (hasta 100 kbit s) y Modo fast (hasta 400 kbit s).
- Modo Multimaster transmitter/Slave receiver.
- Modo Multimaster receiver/Slave transmitter.

Implementación

Tres instancias I2C en modo Multimaster

- Compatible con la especificación I2C version 2.1 de Philips.
- Soporta Modo standard (hasta 100 kbit s) y Modo fast (hasta 400 kbit s).
- Modo Multimaster transmitter/Slave receiver.
- Modo Multimaster receiver/Slave transmitter.
- Modos combinados Master transmit/receive y receive/transmit.

Implementación

Tres instancias I2C en modo Multimaster

- Compatible con la especificación I2C version 2.1 de Philips.
- Soporta Modo standard (hasta 100 kbit s) y Modo fast (hasta 400 kbit s).
- Modo Multimaster transmitter/Slave receiver.
- Modo Multimaster receiver/Slave transmitter.
- Modos combinados Master transmit/receive y receive/transmit.
- Modos de direccionamiento de dispositivos de 7 bit y 10 bit.

Implementación

Tres instancias I2C en modo Multimaster

- Compatible con la especificación I2C version 2.1 de Philips.
- Soporta Modo standard (hasta 100 kbit s) y Modo fast (hasta 400 kbit s).
- Modo Multimaster transmitter/Slave receiver.
- Modo Multimaster receiver/Slave transmitter.
- Modos combinados Master transmit/receive y receive/transmit.
- Modos de direccionamiento de dispositivos de 7 bit y 10 bit.
- FIFO interna de 32 B para bufferear lecturas y escrituras en cada módulo.

Implementación

Tres instancias I2C en modo Multimaster

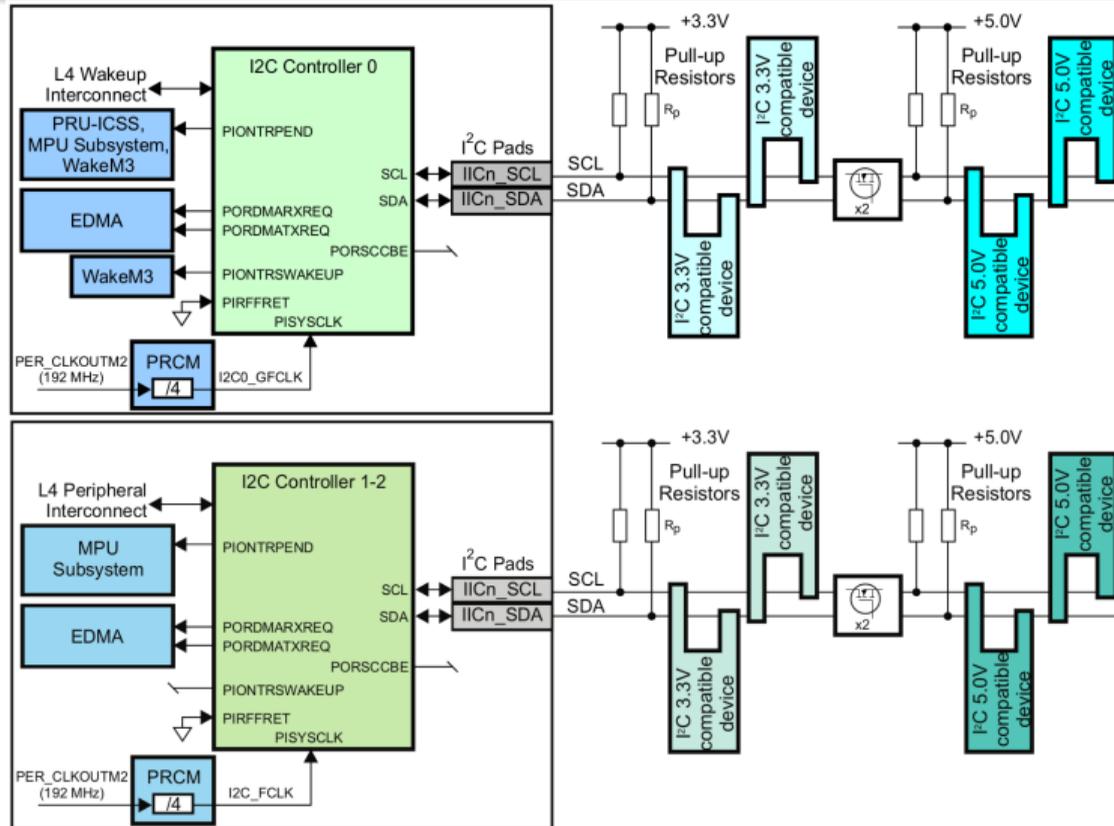
- Compatible con la especificación I2C version 2.1 de Philips.
- Soporta Modo standard (hasta 100 kbit s) y Modo fast (hasta 400 kbit s).
- Modo Multimaster transmitter/Slave receiver.
- Modo Multimaster receiver/Slave transmitter.
- Modos combinados Master transmit/receive y receive/transmit.
- Modos de direccionamiento de dispositivos de 7 bit y 10 bit.
- FIFO interna de 32 B para bufferear lecturas y escrituras en cada módulo.
- Generation de clock programable.

Implementación

Tres instancias I2C en modo Multimaster

- Compatible con la especificación I2C version 2.1 de Philips.
- Soporta Modo standard (hasta 100 kbit s) y Modo fast (hasta 400 kbit s).
- Modo Multimaster transmitter/Slave receiver.
- Modo Multimaster receiver/Slave transmitter.
- Modos combinados Master transmit/receive y receive/transmit.
- Modos de direccionamiento de dispositivos de 7 bit y 10 bit.
- FIFO interna de 32 B para bufferear lecturas y escrituras en cada módulo.
- Generation de clock programable.
- Dos canales de DMA, una línea de interrupción.

Implementación



I2C: Recursos adicionales

Lecturas y Documentación adicional:

- The I2C Specification. Version 2.1. Philips.
- I2C-bus specification and user manual. NXP.
- AM335x and AMIC110 Sitara™ Processors Technical Reference Manual. Chapter 21 I2C. Texas Instruments

Temario

- 1 Introducción
 - Motivación
 - Primeros estándares
- 2 SPI
 - Introducción
 - Funcionamiento
 - Implementación SPI en el SoC Sitara AM3358 (BBB)
- 3 I2C
 - Introducción
 - Funcionamiento
 - Implementación I2C en el SoC Sitara AM3358 (BBB)
- 4 **Otros protocolos serie**
 - **Controller Area Network (CAN)**
 - 1-Wire

CAN. Características principales

- Topología bus diferencial (símil RS-485).

CAN. Características principales

- Topología bus diferencial (símil RS-485).
- Hasta 1 Mibit s.

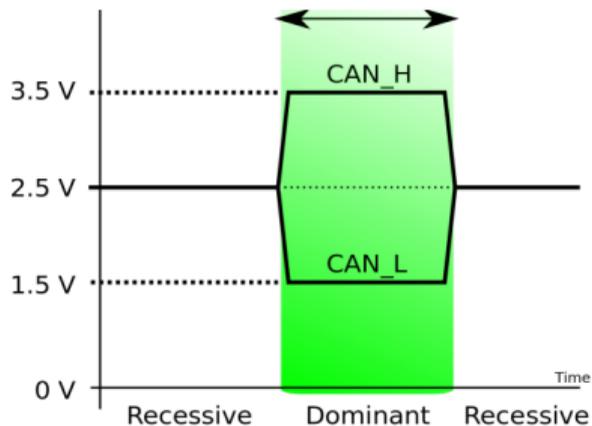
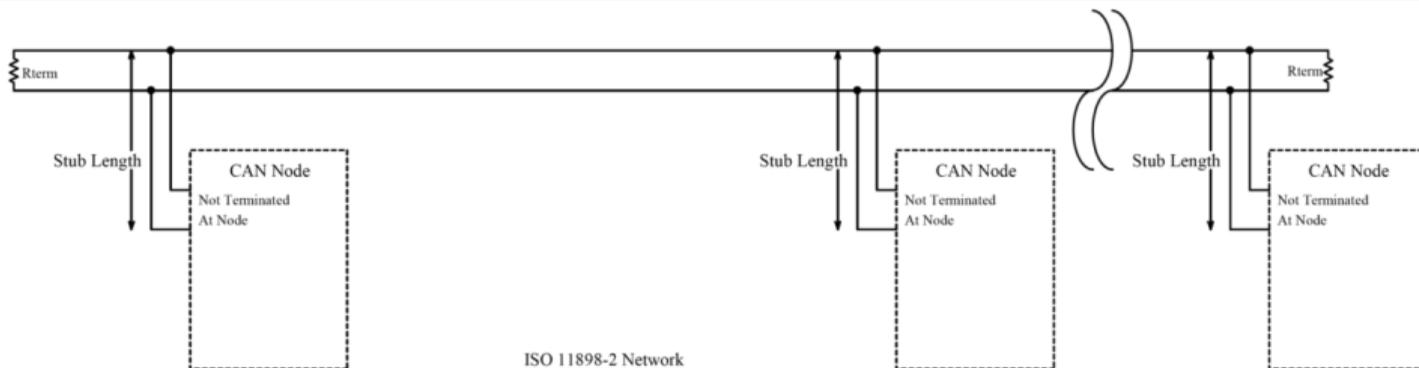
CAN. Características principales

- Topología bus diferencial (símil RS-485).
- Hasta 1 Mibit s.
- Alta inmunidad al ruido.

CAN. Características principales

- Topología bus diferencial (símil RS-485).
- Hasta 1 Mibit s.
- Alta inmunidad al ruido.
- Muy usado en entornos industriales, automotivo, aerospace.

CAN. Topología. Señales



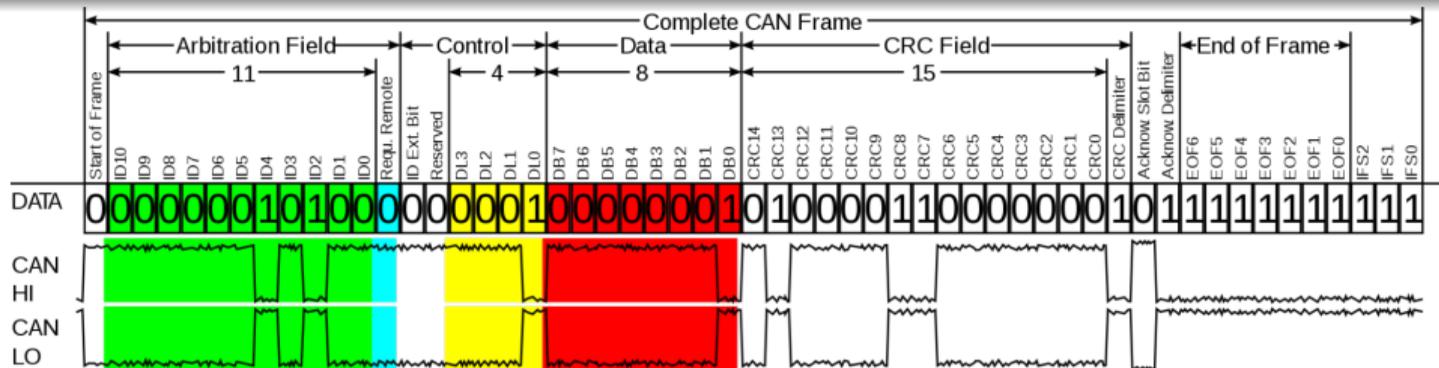
Este bus utiliza señales diferenciales Wired-AND.

*Las señales, CAN high (**CANH**) y CAN low (**CANL**) se llevan a un estado "dominante" cuando $CANH > CANL$, o a un estado "recesivo" con $CANH \leq CANL$.*

*Un bit de datos 0 corresponde al estado **dominante**.
Un bit de datos de 1 a un estado **recesivo**.*

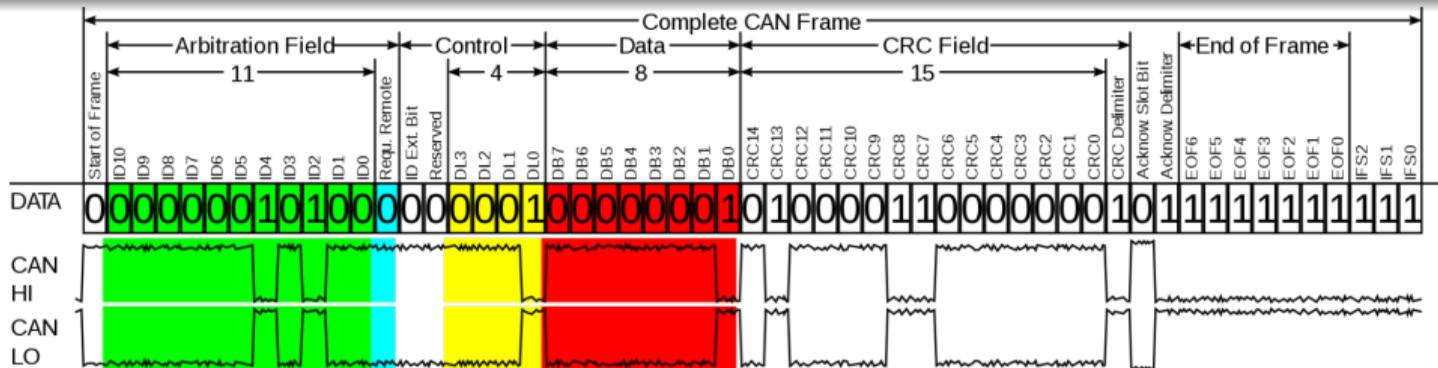
La convención Wired-AND, permite otorgar mayor prioridad en el bus a los nodos con números de identificación más bajos.

CAN. Formato de un Frame

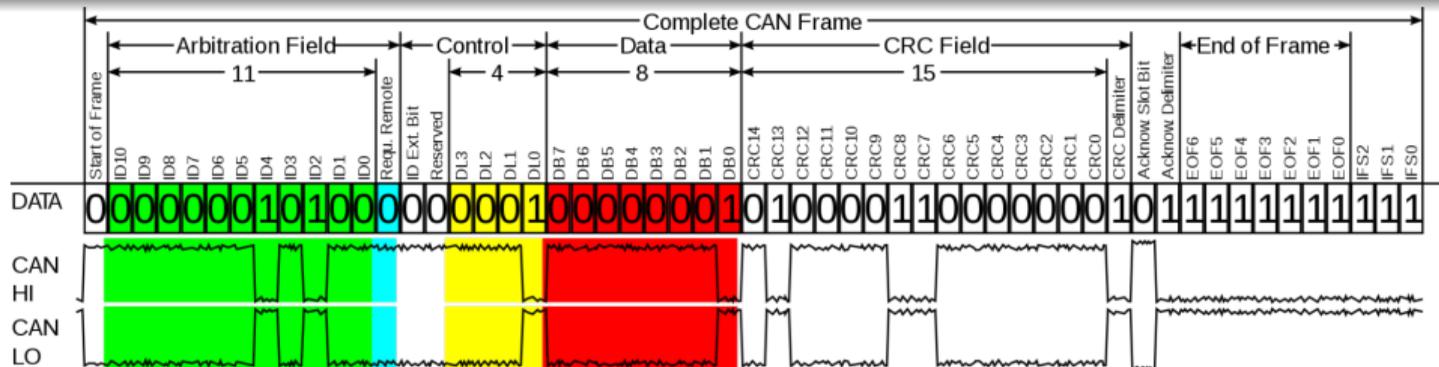


Start-of-frame: 1 bit. Indica el inicio de transmisión de un frame.

CAN. Formato de un Frame

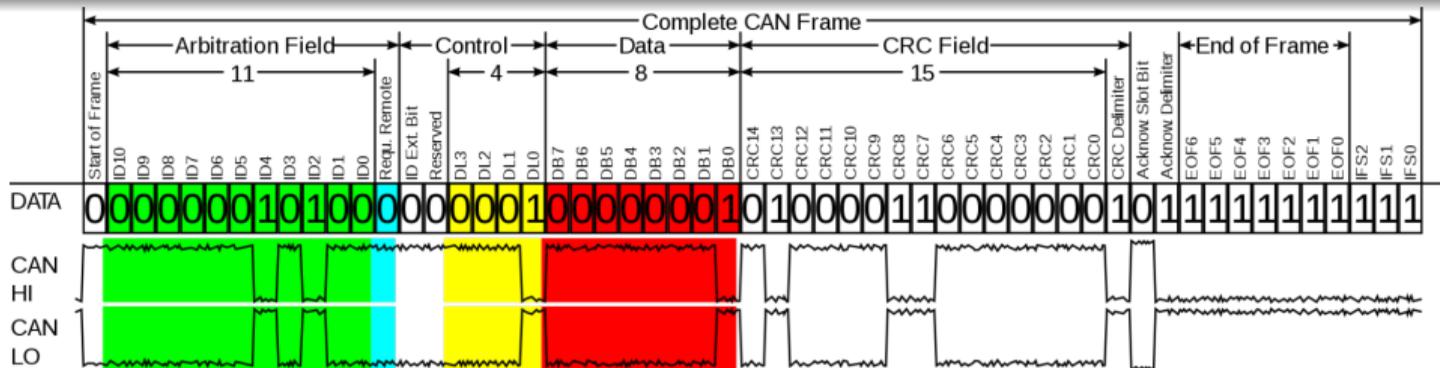


CAN. Formato de un Frame



Data length code (DLC) (amarillo): 4 bit. Número de bytes de datos (0 a 8 B).

CAN. Formato de un Frame

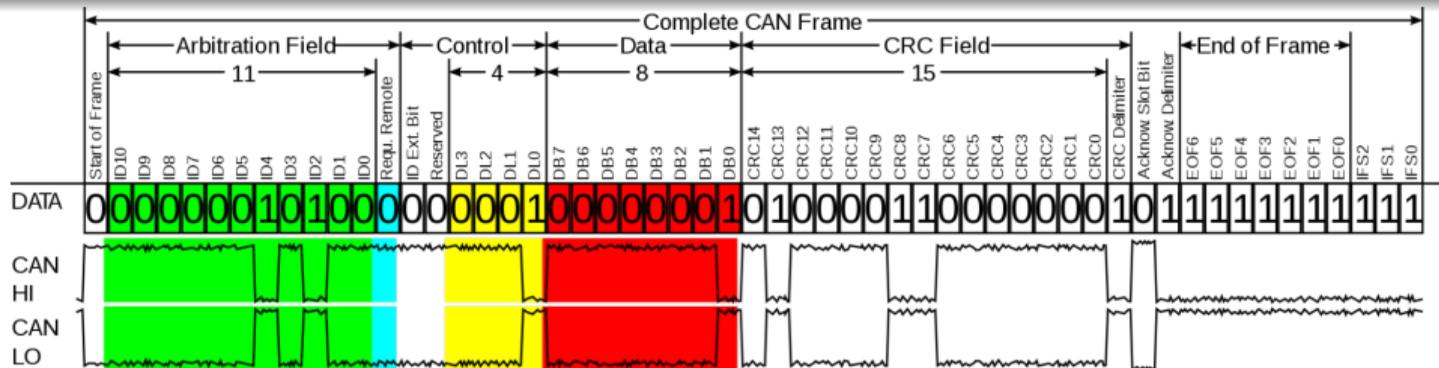


Data length code (DLC) (amarillo): 4 bit. Número de bytes de datos (0 a 8 B).

Data field (rojo): 0 bit a 64 bit. Datos a transmitir (longitud en bits especificada en DLC).

CRC: 15 bit. Cyclic redundancy check.

CAN. Formato de un Frame



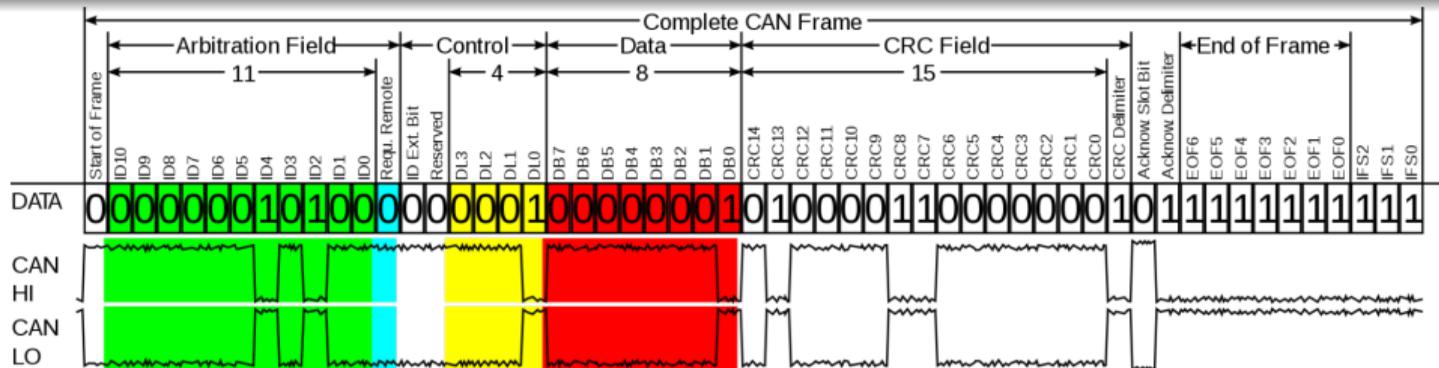
Data length code (DLC) (amarillo): 4 bit. Número de bytes de datos (0 a 8 B).

Data field (rojo): 0 bit a 64 bit. Datos a transmitir (longitud en bits especificada en DLC).

CRC: 15 bit. Cyclic redundancy check.

CRC delimiter: 1 bit. Debe ser recesivo (1).

CAN. Formato de un Frame



Data length code (DLC) (amarillo): 4 bit. Número de bytes de datos (0 a 8 B).

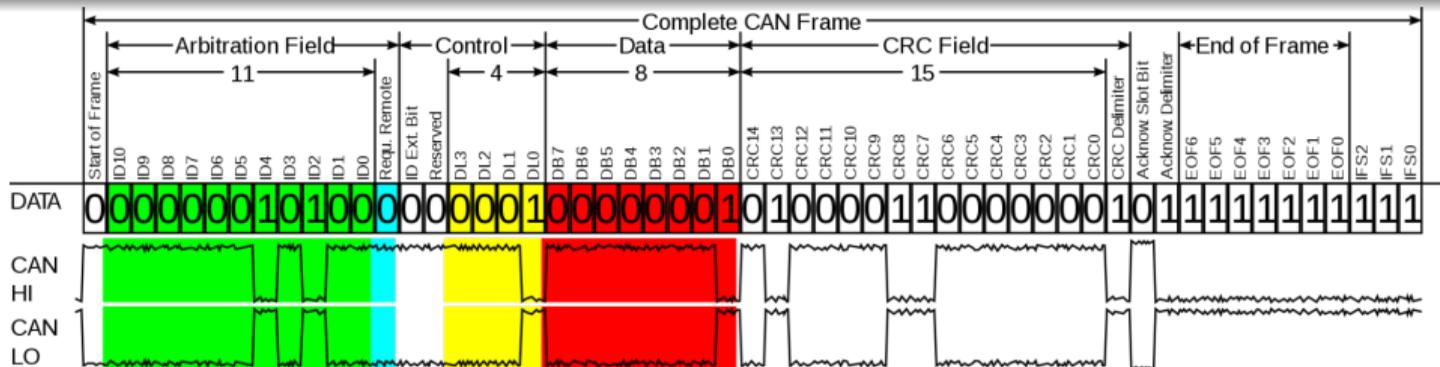
Data field (rojo): 0 bit a 64 bit. Datos a transmitir (longitud en bits especificada en DLC).

CRC: 15 bit. Cyclic redundancy check.

CRC delimiter: 1 bit. Debe ser recesivo (1).

ACK Slot: 1 bit. El transmisor envía recesivo (1) y el receptor dominante (0).

CAN. Formato de un Frame



Data length code (DLC) (amarillo): 4 bit. Número de bytes de datos (0 a 8 B).

Data field (rojo): 0 bit a 64 bit. Datos a transmitir (longitud en bits especificada en DLC).

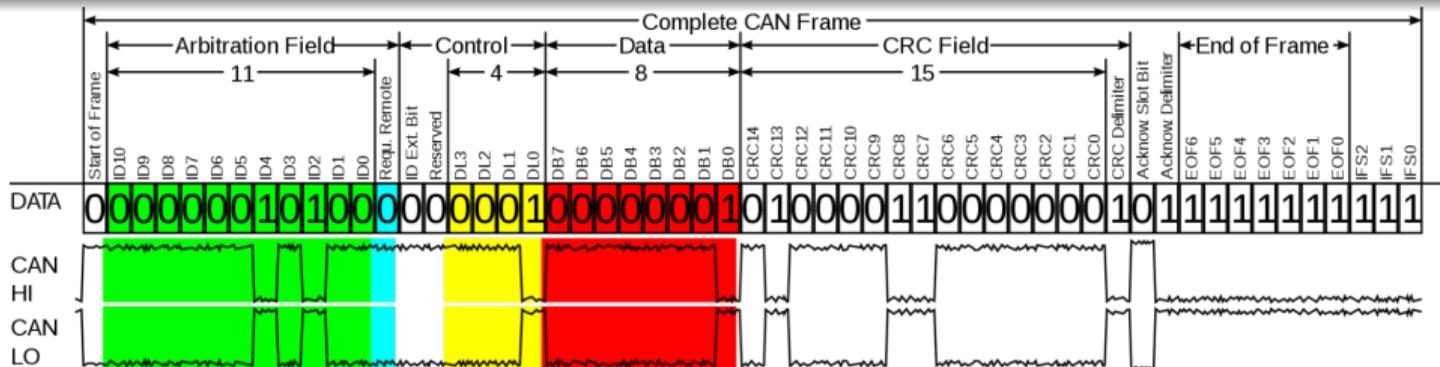
CRC: 15 bit. Cyclic redundancy check.

CRC delimiter: 1 bit. Debe ser recesivo (1).

ACK Slot: 1 bit. El transmisor envía recesivo (1) y el receptor dominante (0).

ACK delimiter: 1 bit. Debe ser recesivo (1).

CAN. Formato de un Frame



Data length code (DLC) (amarillo): 4 bit. Número de bytes de datos (0 a 8 B).

Data field (rojo): 0 bit a 64 bit. Datos a transmitir (longitud en bits especificada en DLC).

CRC: 15 bit. Cyclic redundancy check.

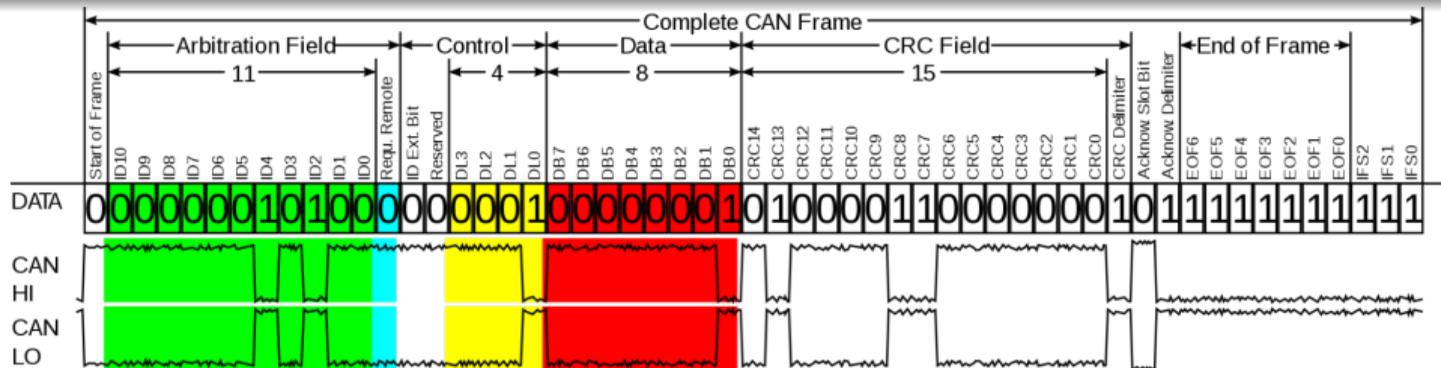
CRC delimiter: 1 bit. Debe ser recesivo (1).

ACK Slot: 1 bit. El transmisor envía recesivo (1) y el receptor dominante (0).

ACK delimiter: 1 bit. Debe ser recesivo (1).

End-of-frame (EOF): 7 bit. Debe ser recesivo (1).

CAN. Formato de un Frame



Data length code (DLC) (amarillo): 4 bit. Número de bytes de datos (0 a 8 B).

Data field (rojo): 0 bit a 64 bit. Datos a transmitir (longitud en bits especificada en DLC).

CRC: 15 bit. Cyclic redundancy check.

CRC delimiter: 1 bit. Debe ser recesivo (1).

ACK Slot: 1 bit. El transmisor envía recesivo (1) y el receptor dominante (0).

ACK delimiter: 1 bit. Debe ser recesivo (1).

End-of-frame (EOF): 7 bit. Debe ser recesivo (1).

Inter-frame spacing (IFS): 3 bit. Debe ser recesivo (1).

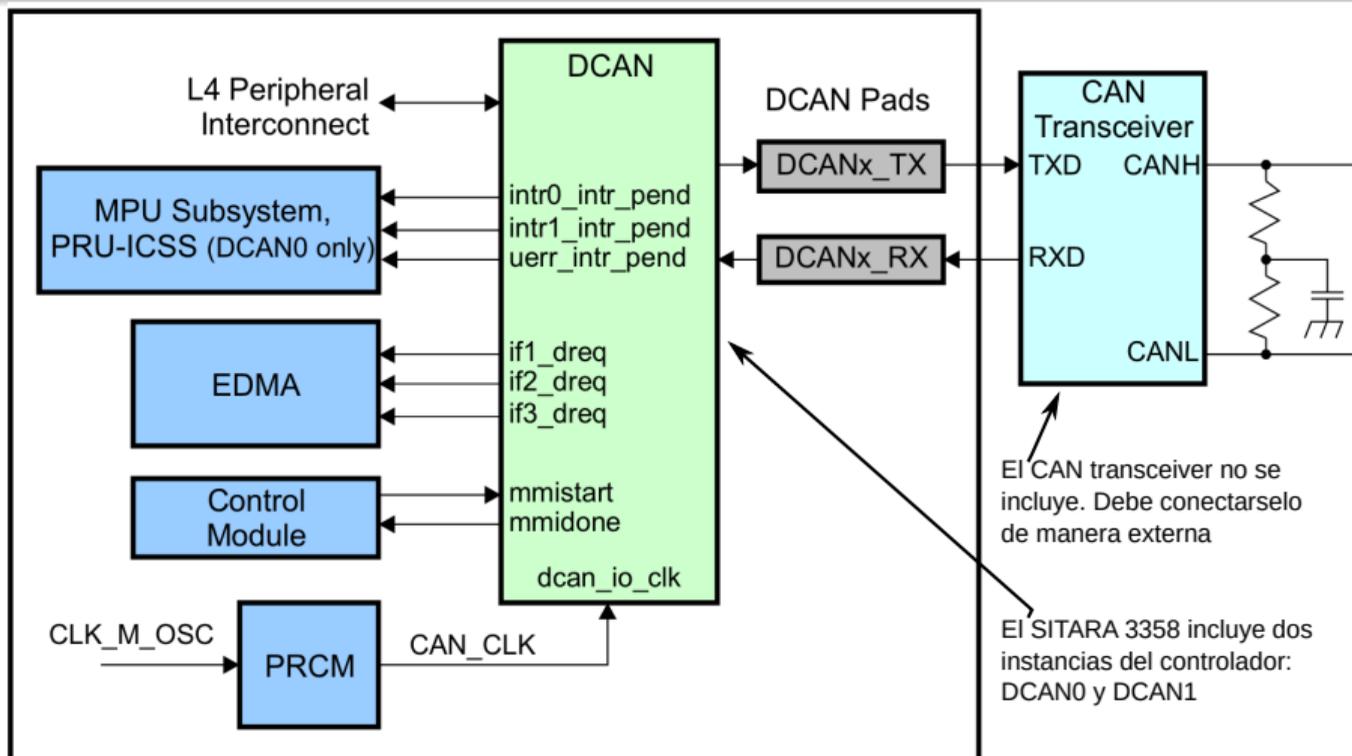
Implementación CAN en el SoC Sitara AM3358 (BBB)

- Soporta protocolo CAN version 2.0 part A, B (ISO 11898-1)
- Bit rates e hasta 1 Mibit s⁻¹
- Dual clock source
- 16, 32, 64 o 128 Objetos de mensaje (Instanciados como 64 en este dispositivo)
- Máscara de identificación Individual pra cada Objeto de Mensaje.
- Modo FIFO Programable para Objetos de Mensaje.
- Modo loop-back Programable para operación self-test.
- Modo Suspend para soporte de debug.

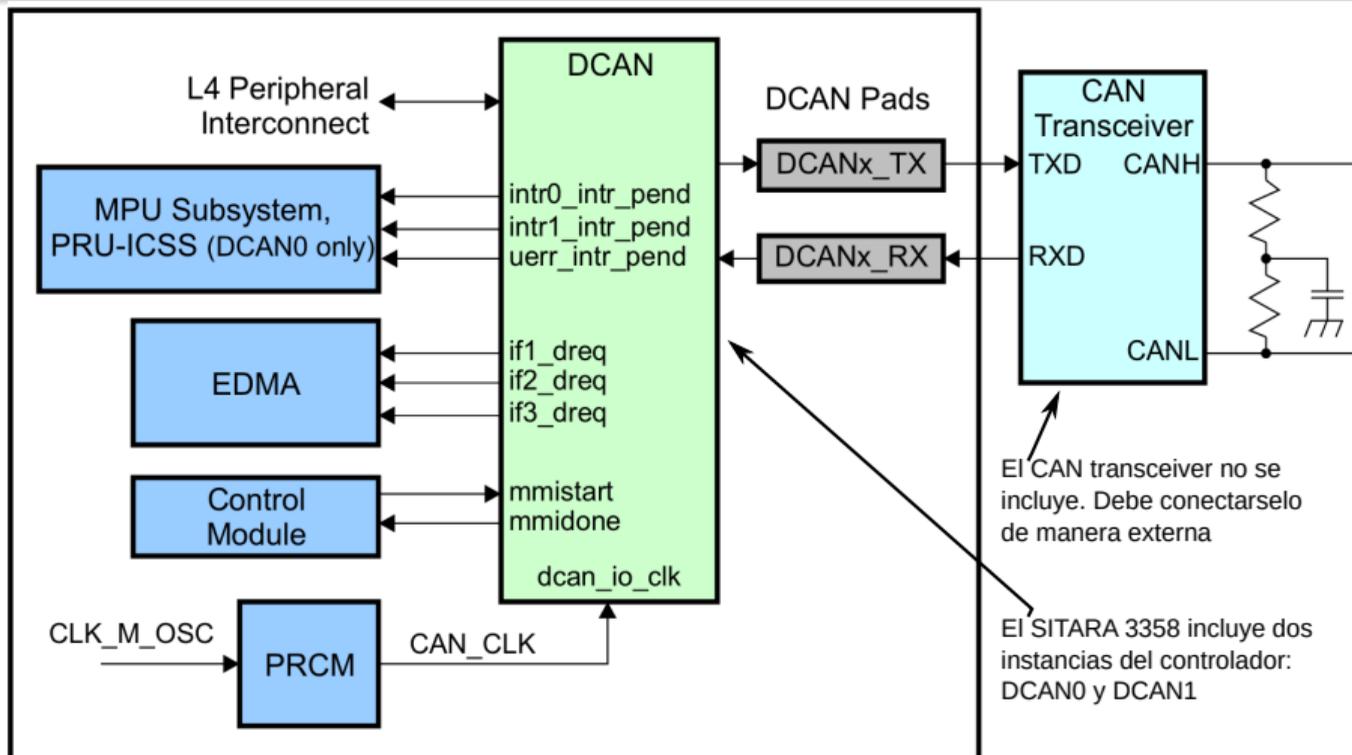
Implementación CAN en el SoC Sitara AM3358 (BBB)

- Software module reset.
- Timer 32-bit programable para activación de Bus Automática luego del estado Bus-Off.
- Mecanismo de Chequeo de Paridad de la RAM de mensajes.
- Acceso Directo a la RAM de mensajes durante el modo test.
- Terminales CAN Rx / Tx configurables como terminales de E/S de propósito general.
- Dos líneas de interrupción, mas línea de interrupción adicional de error de paridad.
- Inicialización de RAM.
- Soporte a DMA.

Implementación CAN en el SoC Sitara AM3358 (BBB)

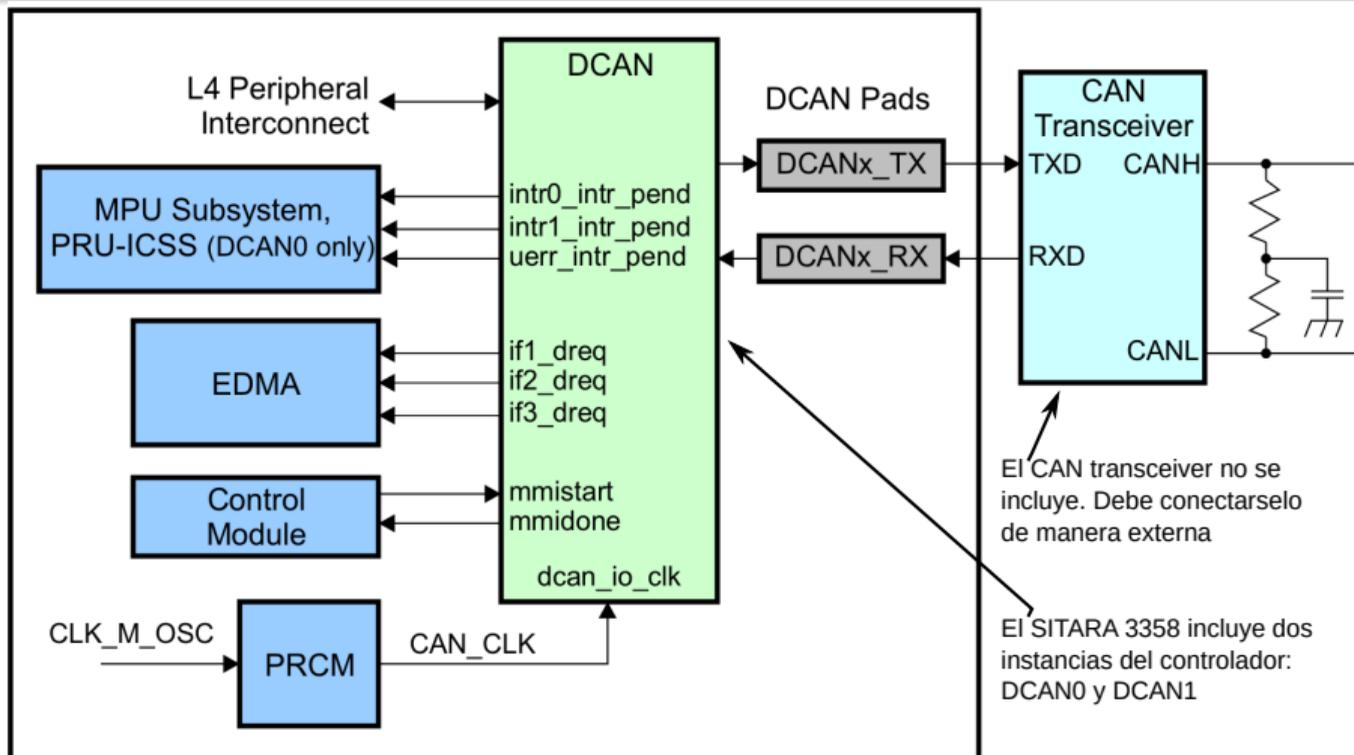


Implementación CAN en el SoC Sitara AM3358 (BBB)



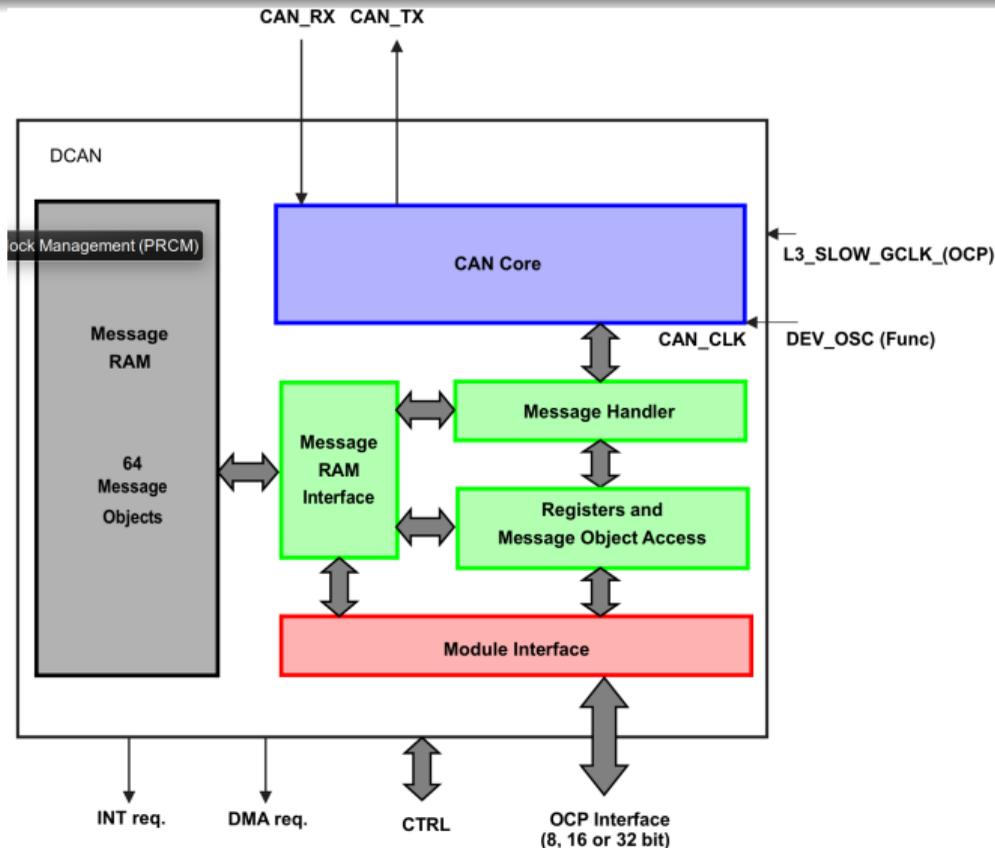
Compatible con la especificación del protocolo CAN 2.0B. La IP del controlador DCAN integrado pertenece a Bosch.

Implementación CAN en el SoC Sitara AM3358 (BBB)

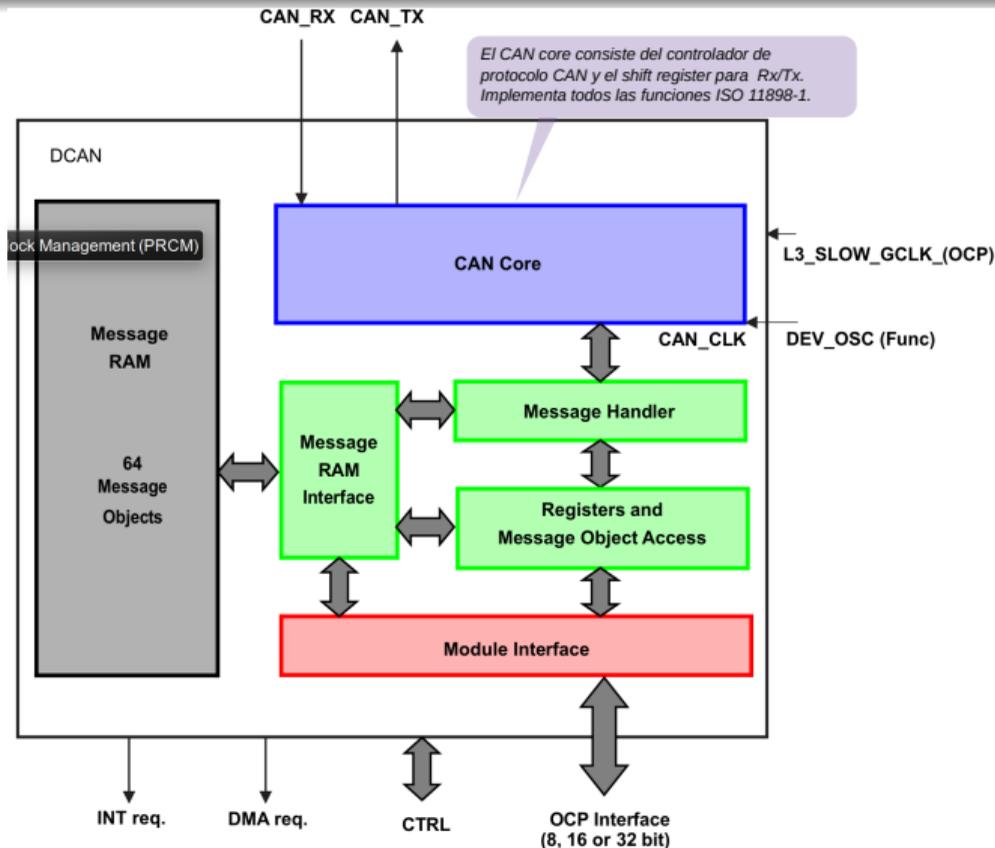


Una opción para agregar un transceiver: [SN65HVD23x 3.3-V CAN Bus Transceivers](#)

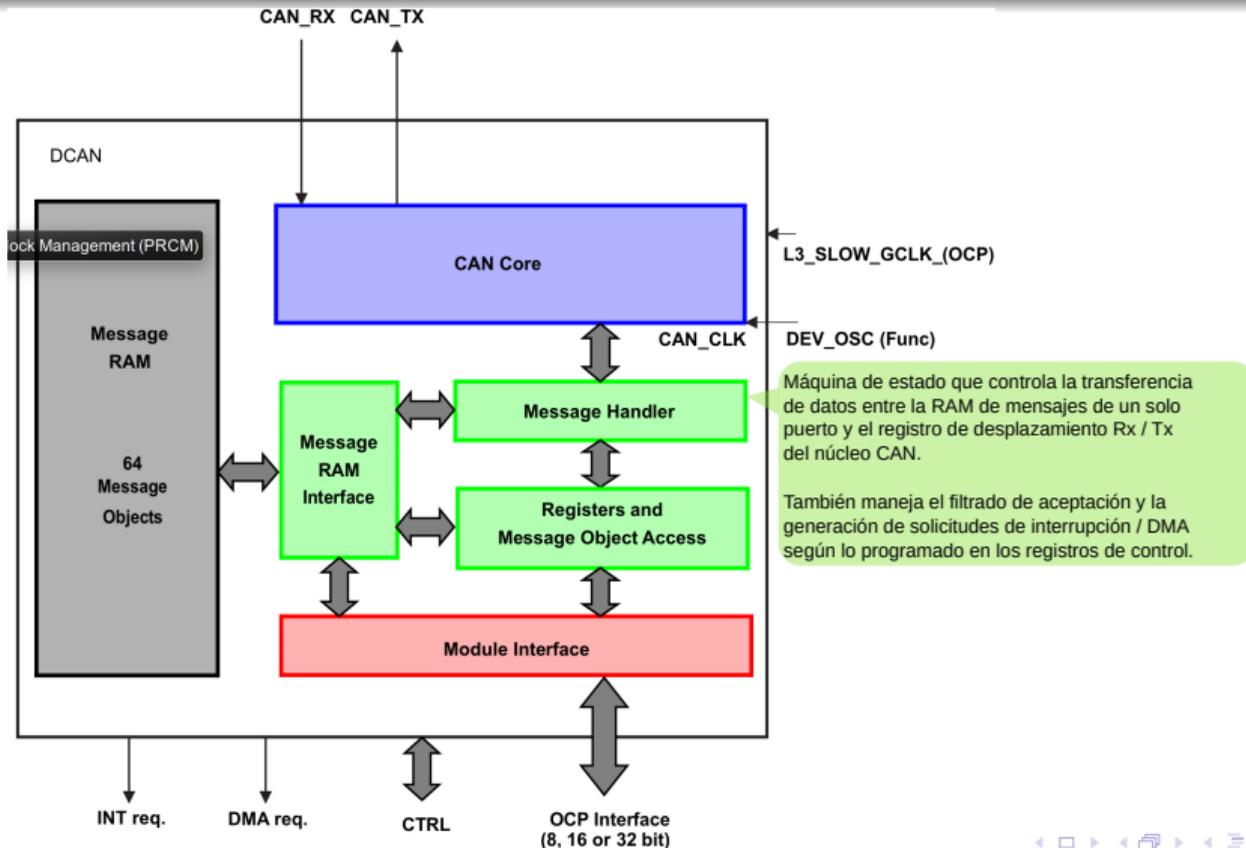
Implementación CAN en el SoC Sitara AM3358 (BBB)



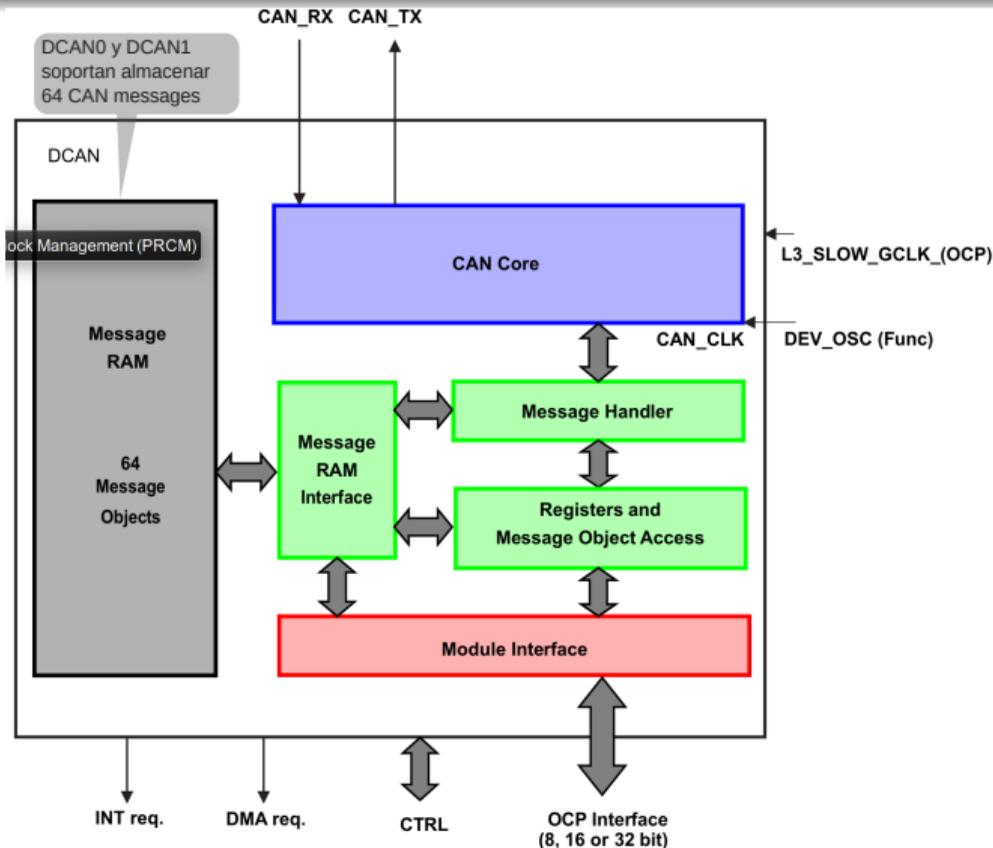
Implementación CAN en el SoC Sitara AM3358 (BBB)



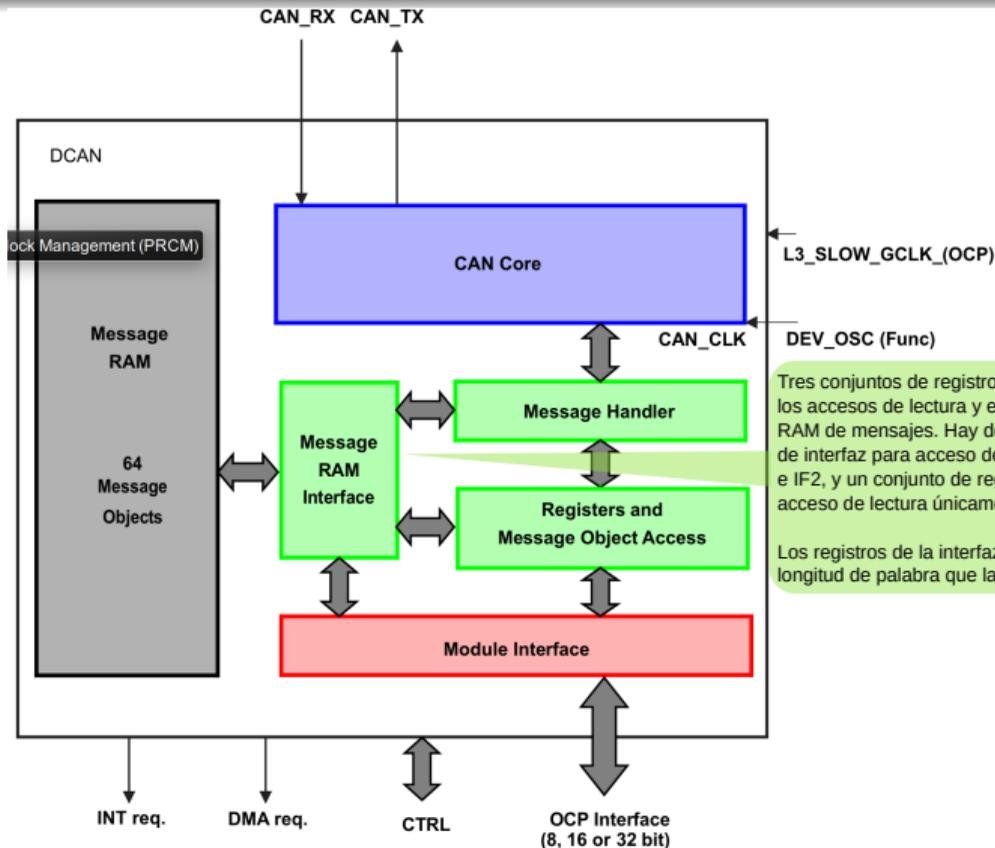
Implementación CAN en el SoC Sitara AM3358 (BBB)



Implementación CAN en el SoC Sitara AM3358 (BBB)



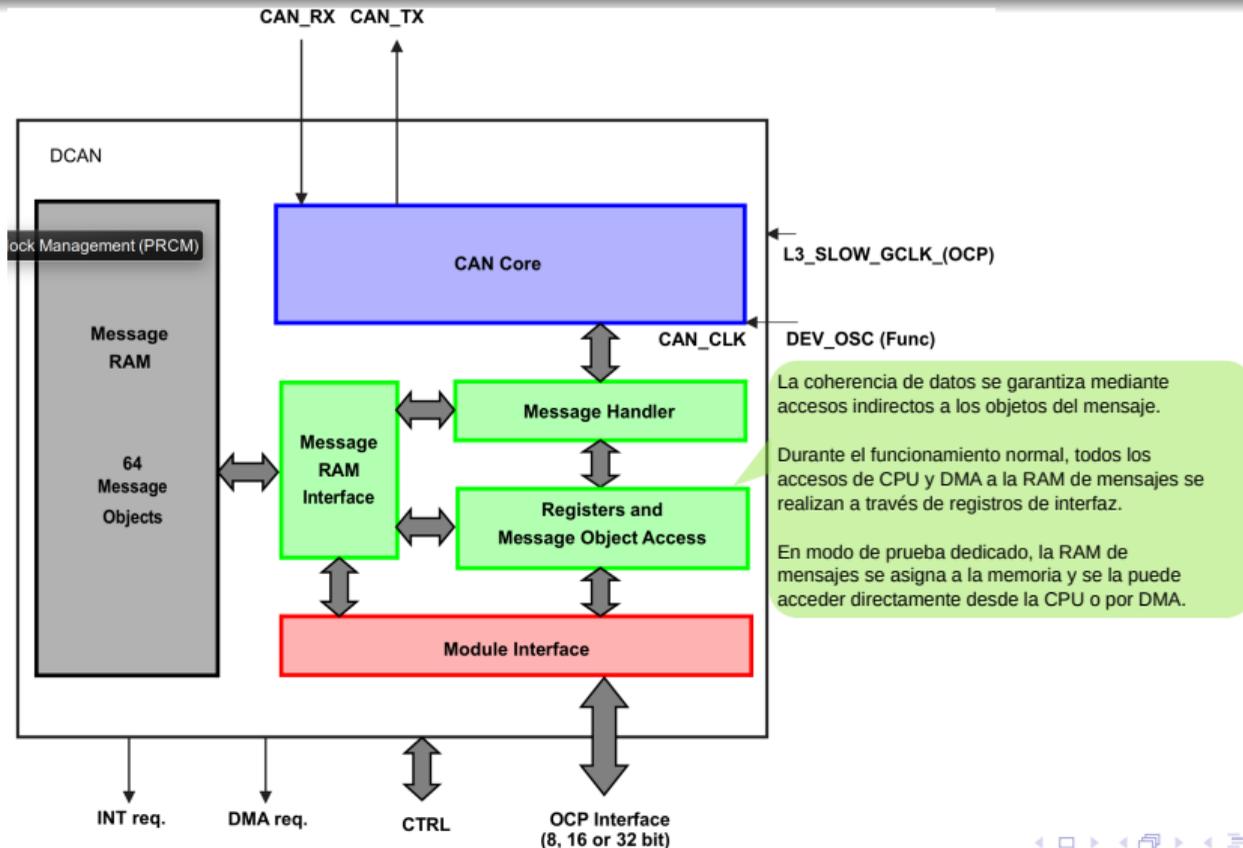
Implementación CAN en el SoC Sitara AM3358 (BBB)



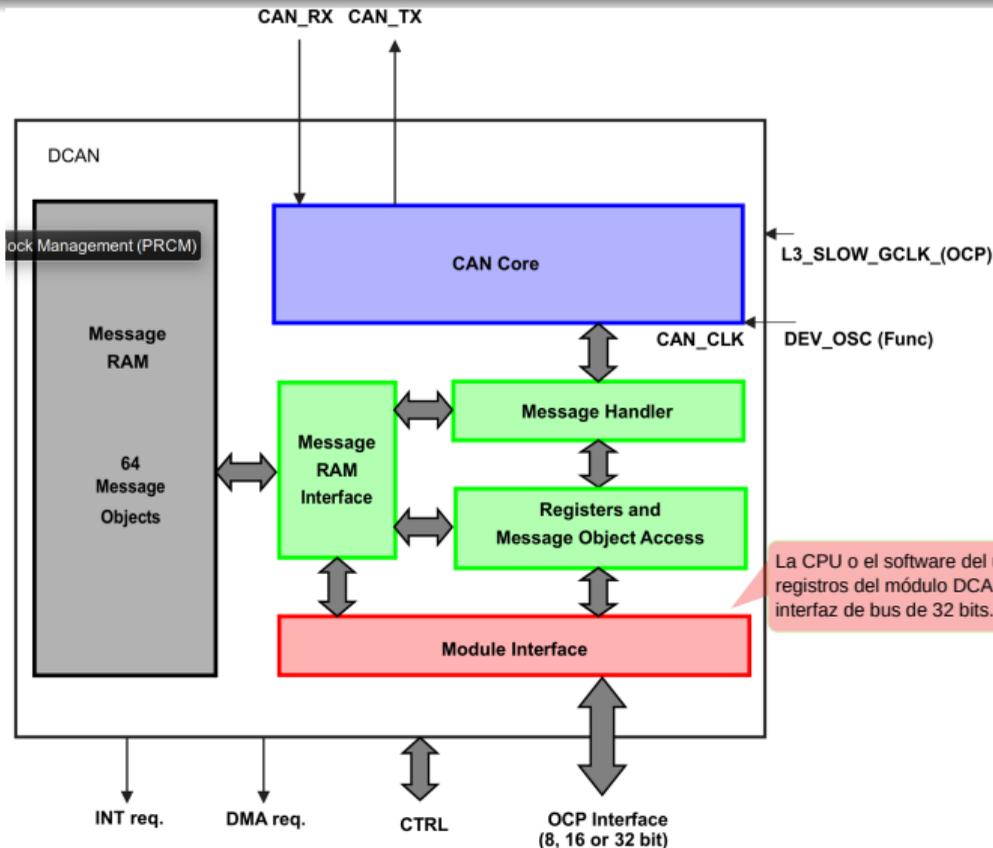
Tres conjuntos de registros de interfaz controlan los accesos de lectura y escritura de la CPU a la RAM de mensajes. Hay dos conjuntos de registros de interfaz para acceso de lectura y escritura, IF1 e IF2, y un conjunto de registros de interfaz para acceso de lectura únicamente, IF3.

Los registros de la interfaz tienen la misma longitud de palabra que la RAM de mensajes.

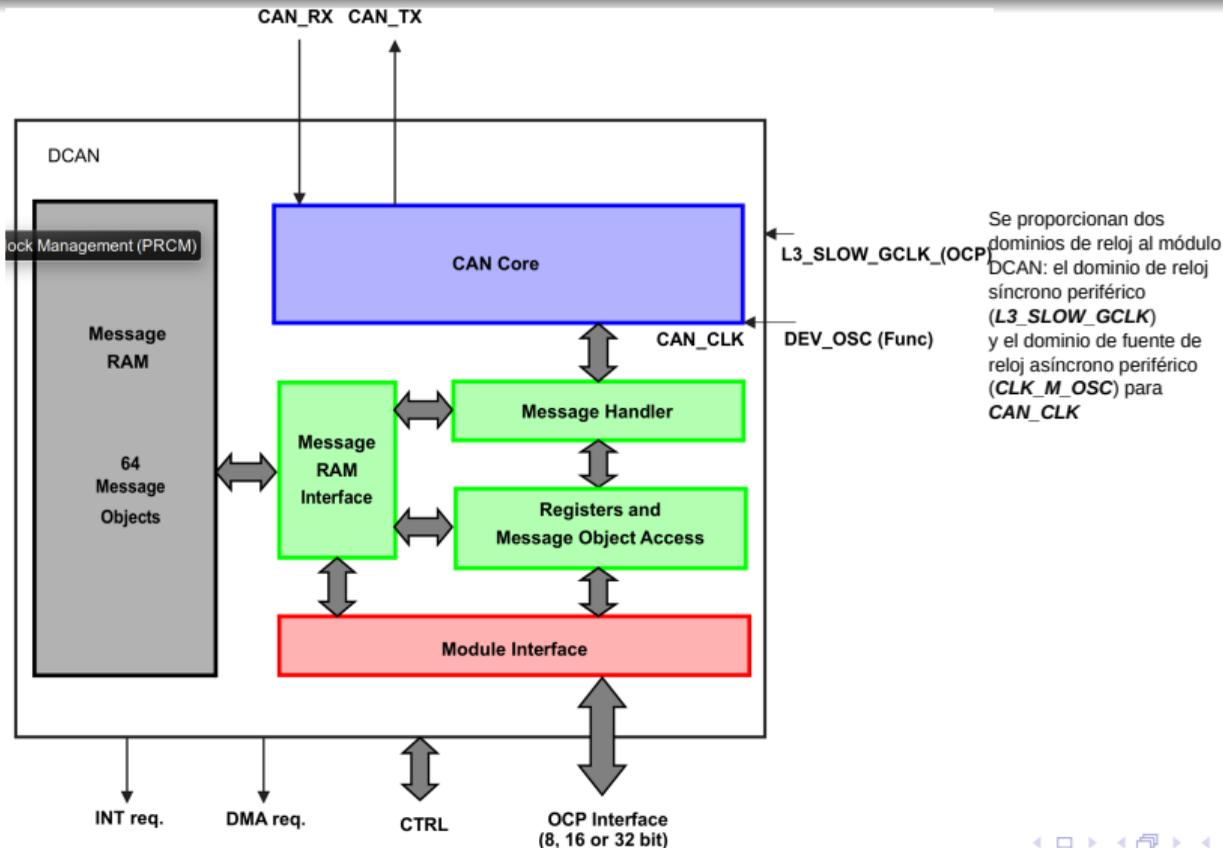
Implementación CAN en el SoC Sitara AM3358 (BBB)



Implementación CAN en el SoC Sitara AM3358 (BBB)



Implementación CAN en el SoC Sitara AM3358 (BBB)



CAN: Recursos adicionales

- ISO 11898-1:2015, Part 1: Data link layer and physical signalling
- ISO 11898-2:2003, Part 2: High-speed medium access unit
- ISO 11898-3:2006, Part 3: Low-speed, fault-tolerant, medium-dependent interface
- ISO 11898-4:2004, Part 4: Time-triggered communication
- ISO 11898-5:2007, Part 5: High-speed medium access unit with low-power mode
- ISO 11898-6:2013: Part 6: High-speed medium access unit with selective wake-up functionality
- ISO 16845:2004, Conformance test plan

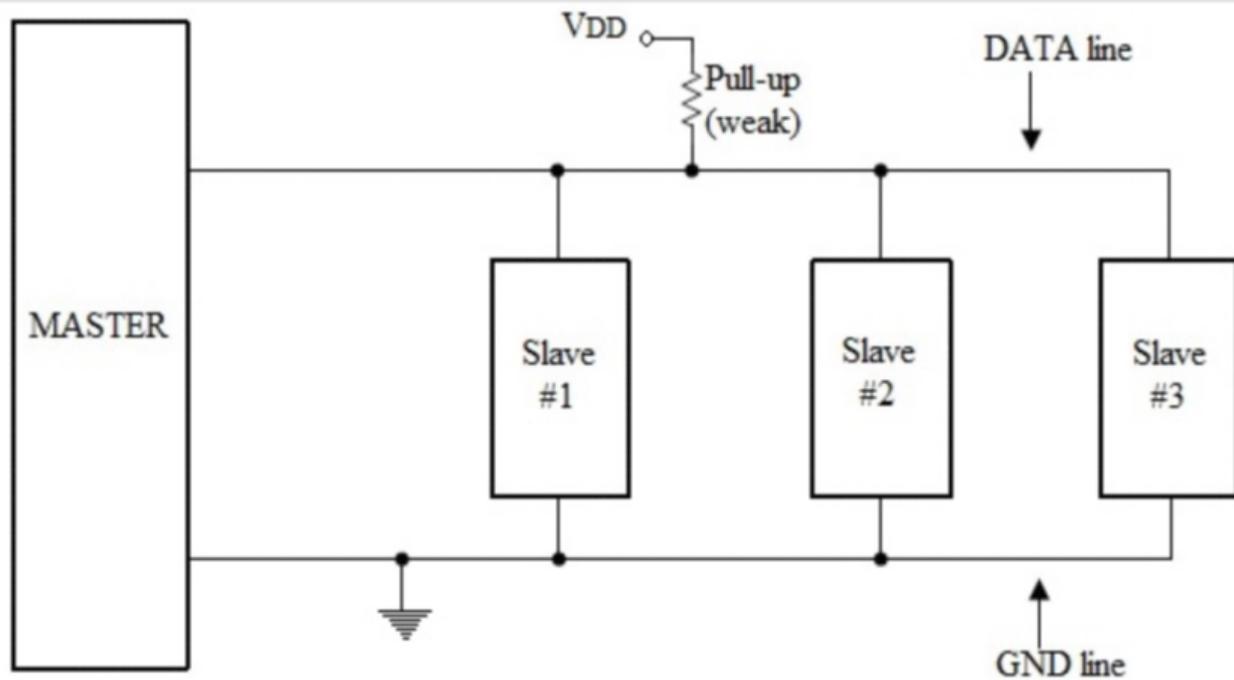
Temario

- 1 Introducción
 - Motivación
 - Primeros estándares
- 2 SPI
 - Introducción
 - Funcionamiento
 - Implementación SPI en el SoC Sitara AM3358 (BBB)
- 3 I2C
 - Introducción
 - Funcionamiento
 - Implementación I2C en el SoC Sitara AM3358 (BBB)
- 4 **Otros protocolos serie**
 - Controller Area Network (CAN)
 - **1-Wire**

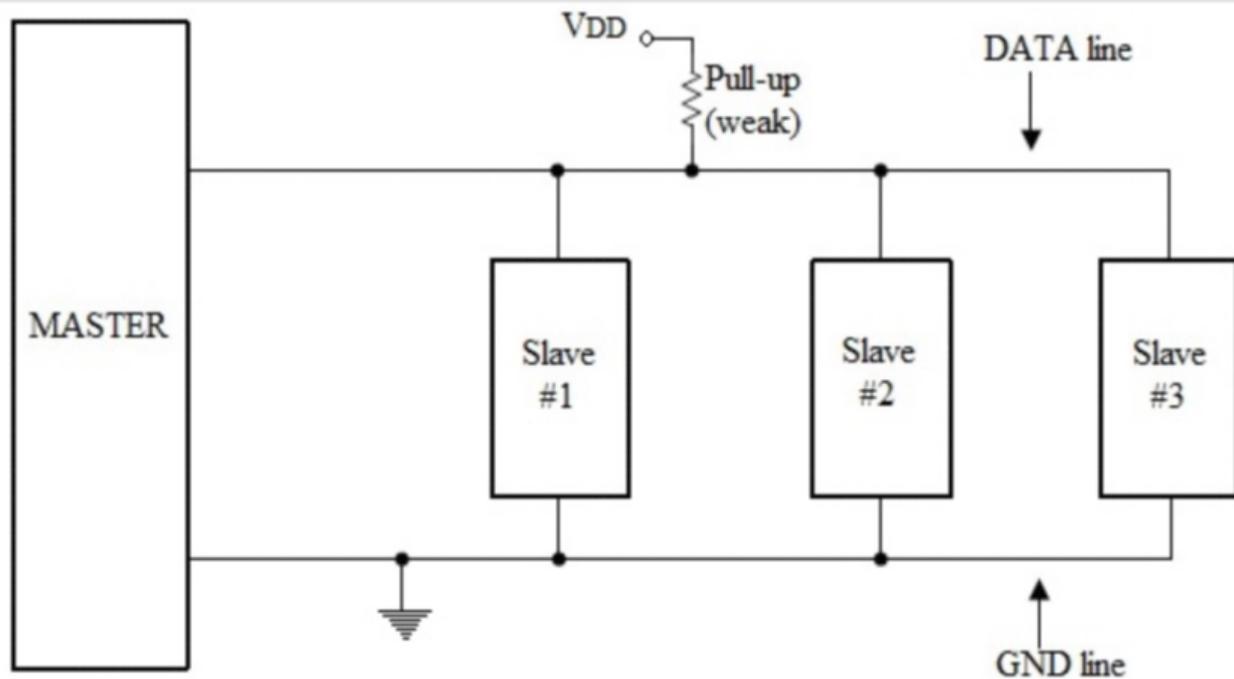
Características Principales

- Inspirado en I2C.
- Alcanza velocidades menores pero amplía el rango
- Requiere una sola señal para datos + alimentación. Esto se logra incluyendo en cada dispositivo un capacitor de 800 pF, que se encarga de almacenar energía durante los períodos inactivos de la línea.
- Se resuelve con dos alambres: Señal/Alimentación y tierra
- Comunicación bidireccional.
- Multi-slave. Un solo Master (a.k.a. **MicroLAN**)
- Ampliamente utilizado en sensores de temperatura, EEPROMs, y llaves de identificación.
- Presente en cables de conexión de Laptops como Dell y Mac. Envía por este hilo características de la fuente, para que el mother board acepte o no el cargador si no es compatible.

Topología de Bus

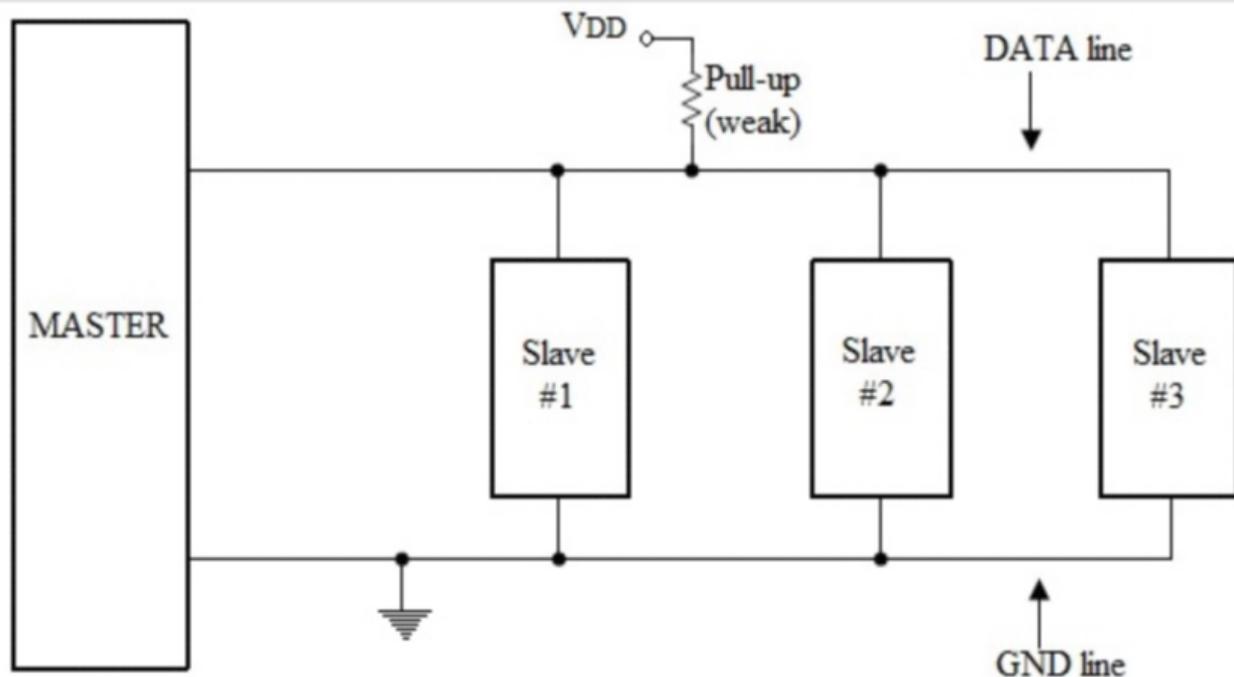


Topología de Bus



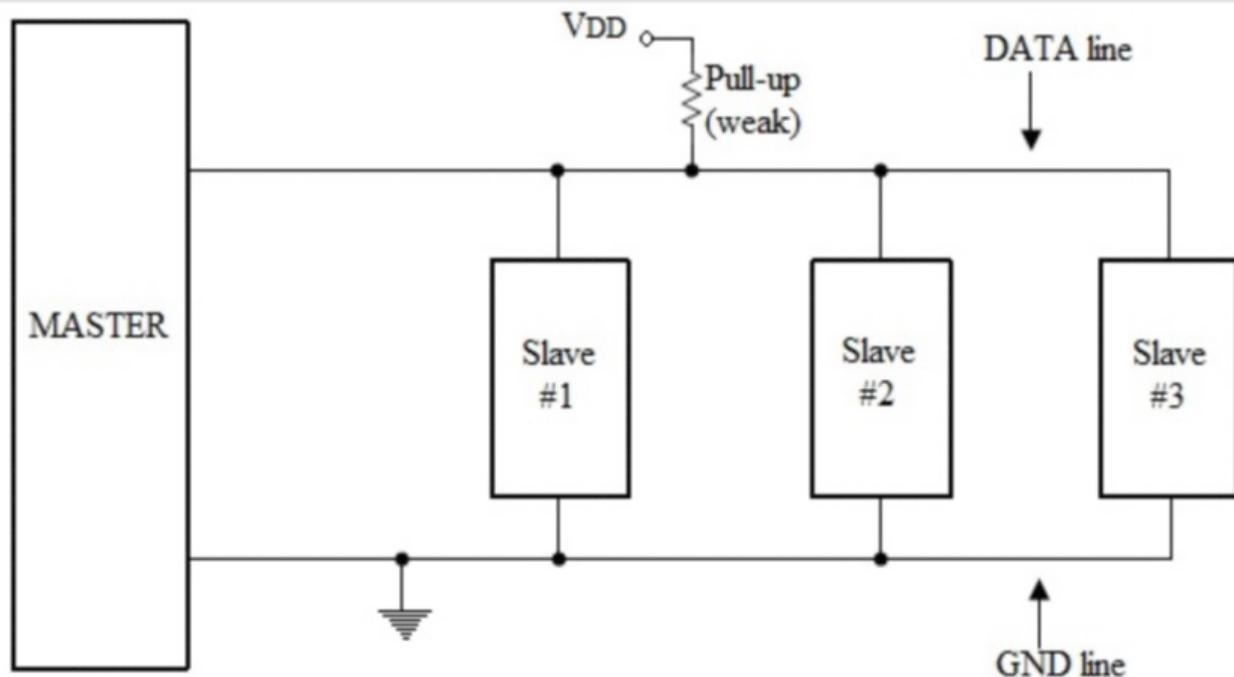
El Master es una PC, un SoC embedded, o un microcontrolador. Debe disponer de fuente de alimentación

Topología de Bus



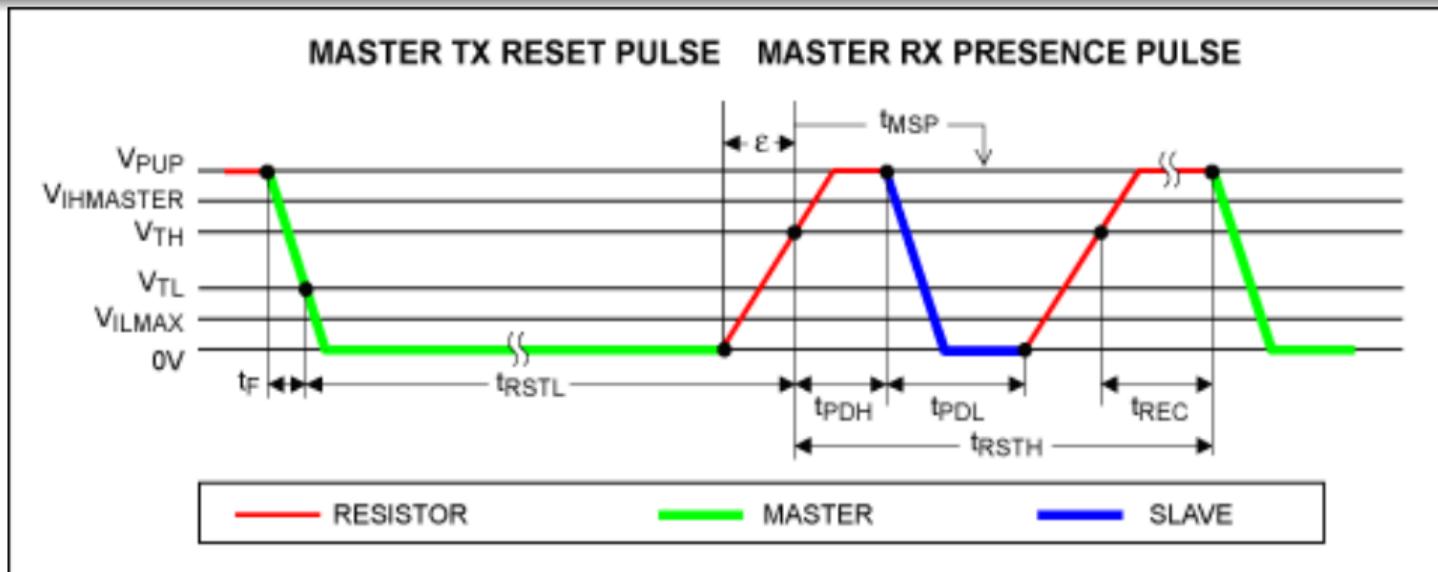
El Master inicia las transacciones lo cual minimiza la probabilidad de colisiones. No obstante el Master puede detectar colisiones y en tal caso reintenta la comunicación.

Topología de Bus

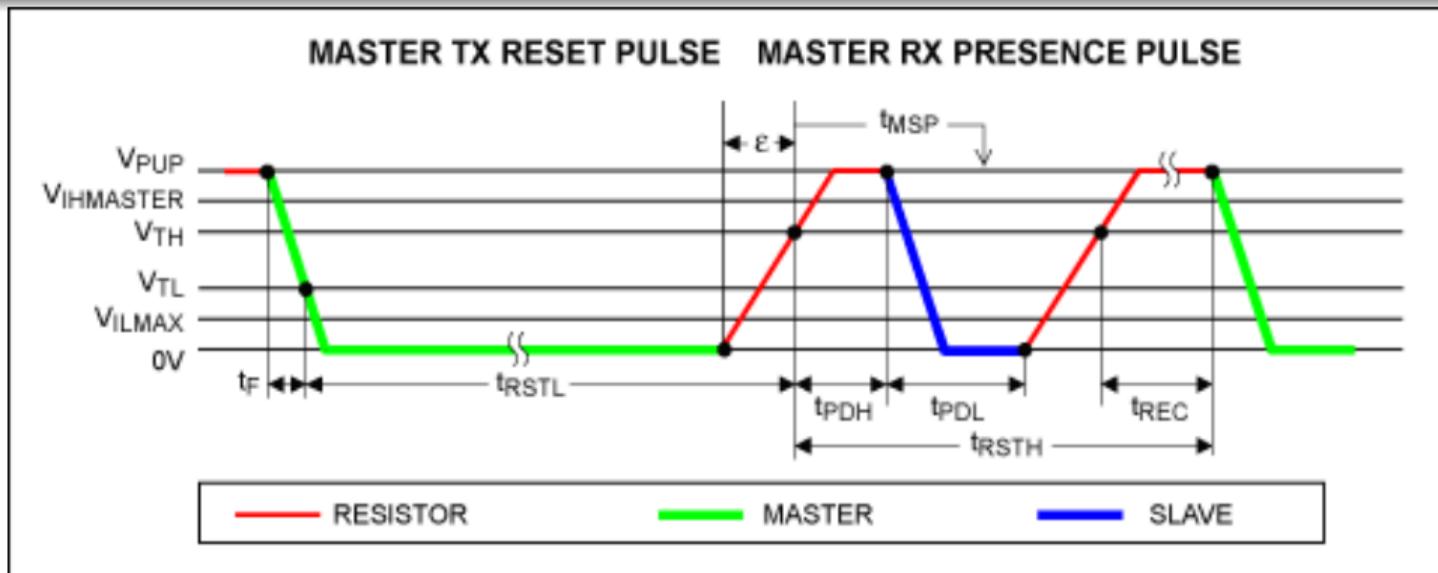


Una comunicación inicia cuando el Master o el Slave pone muy brevemente la línea a tierra. Esto genera corriente por el pull up que drena por el MOSFET open drain de la entrada

Timing de Bus

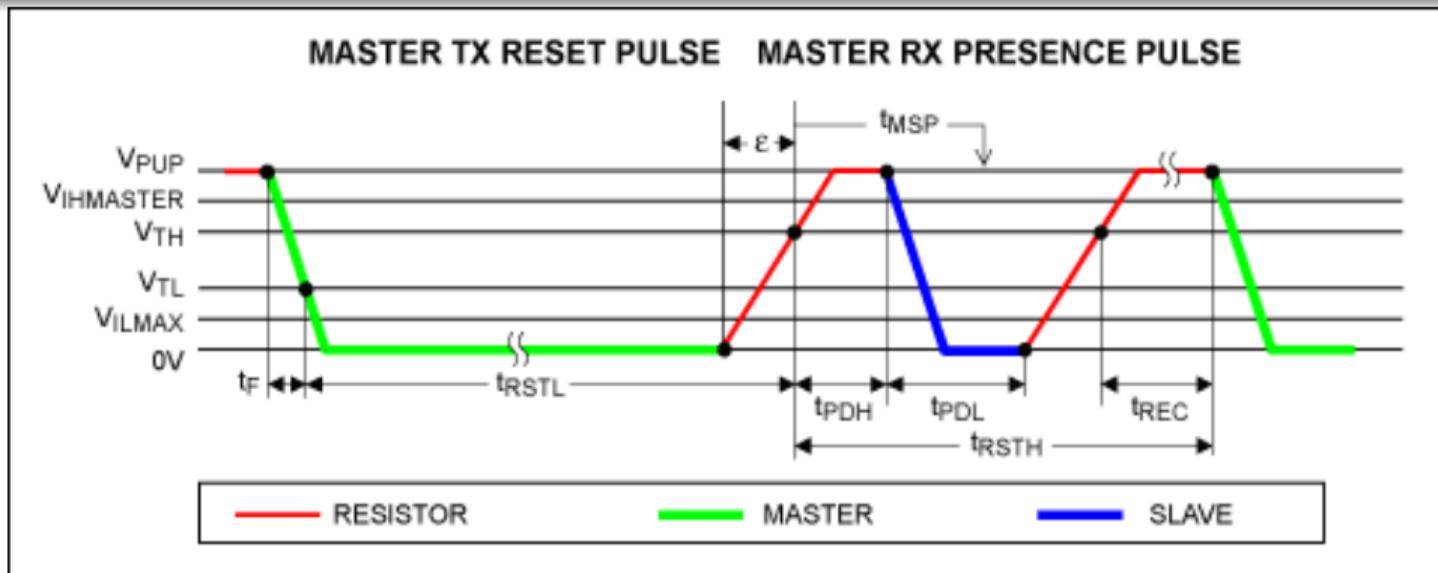


Timing de Bus



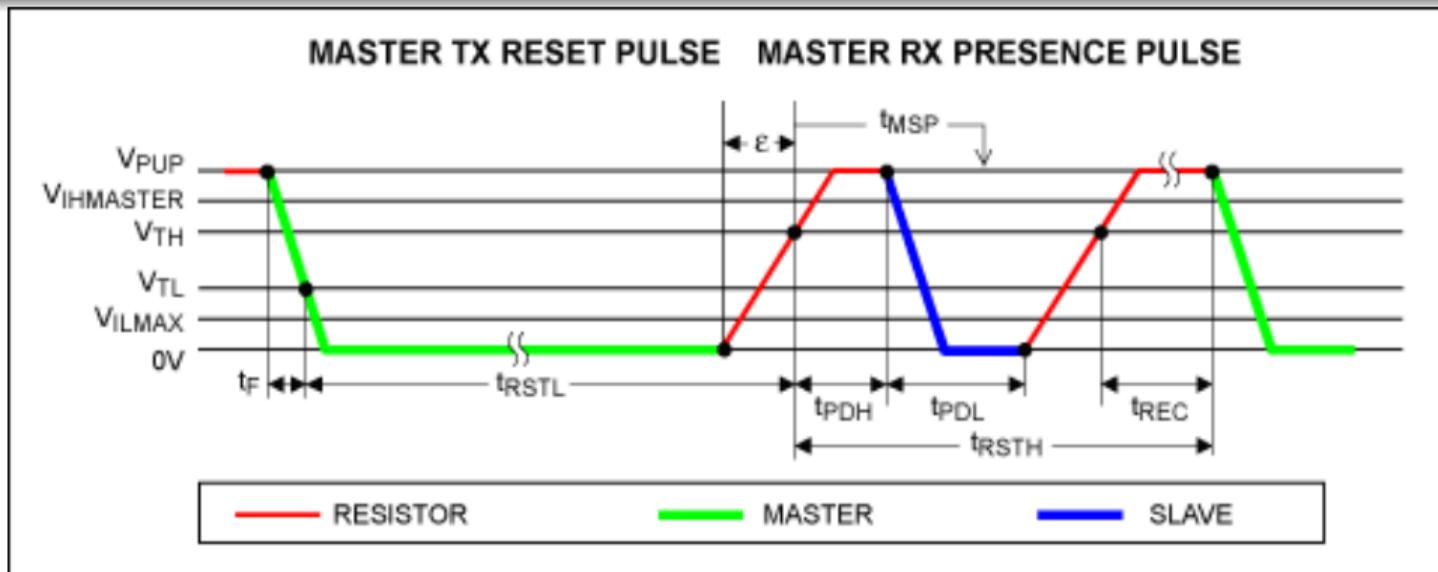
La comunicación 1-Wire comienza con un ciclo de Reset/ Detección de Presencia. Para pasar de inactivo a activo, la tensión en el bus 1-Wire debe caer desde V_{PUP} por debajo del umbral, V_{TL} .

Timing de Bus



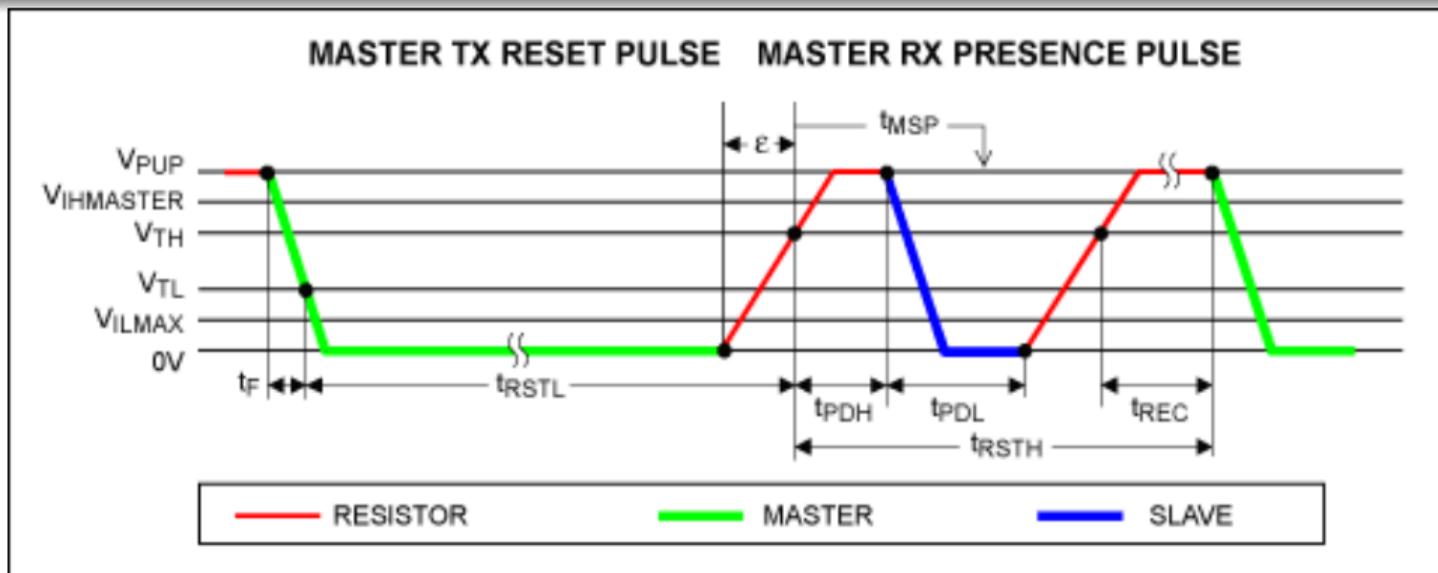
Para pasar de activo a inactivo, la tensión debe aumentar desde V_{ILMAX} más allá del umbral, V_{TH} . El tiempo que tarda en lograr este aumento (" ϵ "), y su duración depende tanto de la resistencia pullup, R_{PUP} , utilizada, como de la capacitancia de la red 1-Wire conectada. La tensión V_{ILMAX} es relevante para el Slave a la hora de determinar un nivel lógico, no para desencadenar ningún evento.

Timing de Bus



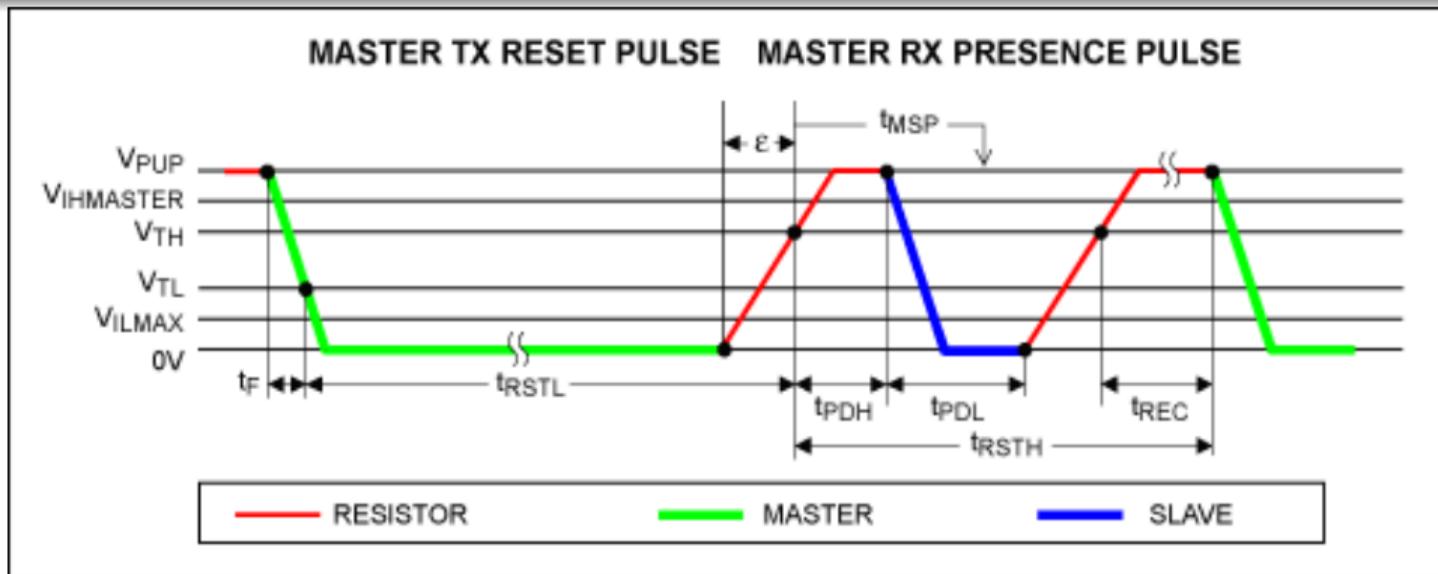
Si el Master usa el control de slew-rate en el flanco descendente, debe llevar a un nivel bajo la línea para que $t_{RSTL} + t_F$ compense el flanco.

Timing de Bus



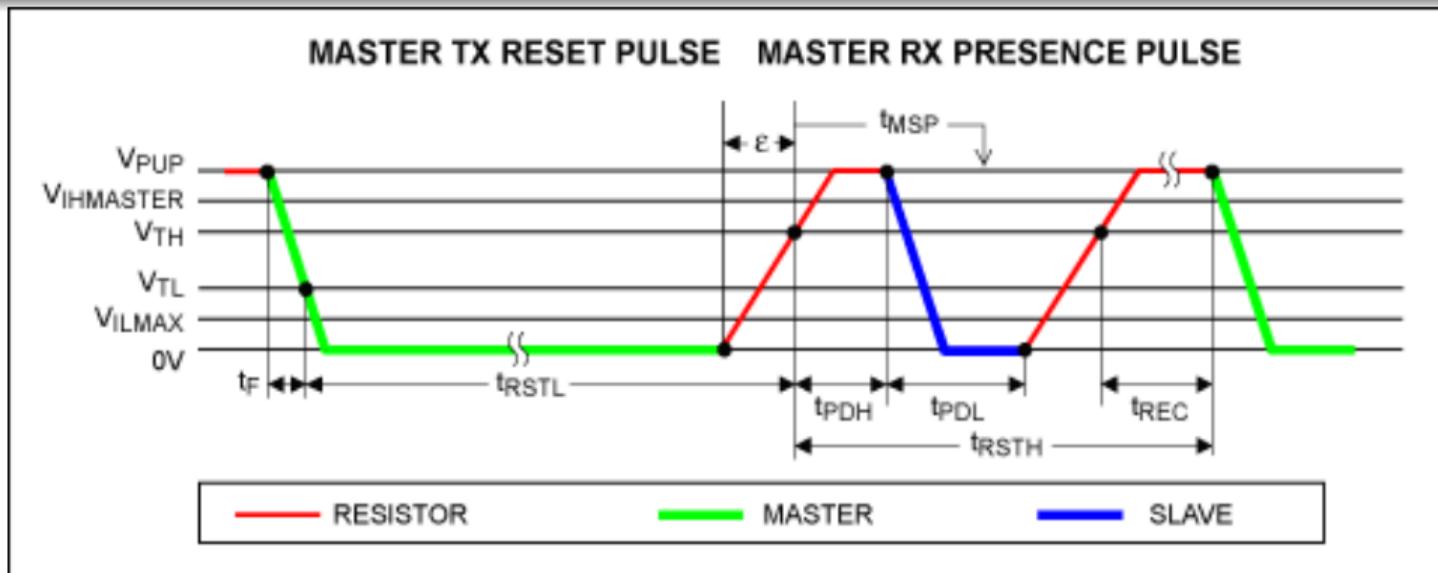
Una duración de t_{RSTL} de $480 \mu s$ o más, saca al dispositivo del modo Overdrive, y lo pone en velocidad estándar. Si un Slave está en modo Overdrive y t_{RSTL} no supera los $80 \mu s$, permanece en modo Overdrive. Si un Slave está en modo Overdrive y t_{RSTL} está entre $80 \mu s$ y $480 \mu s$, se reiniciará pero a una velocidad de comunicación indeterminada.

Timing de Bus



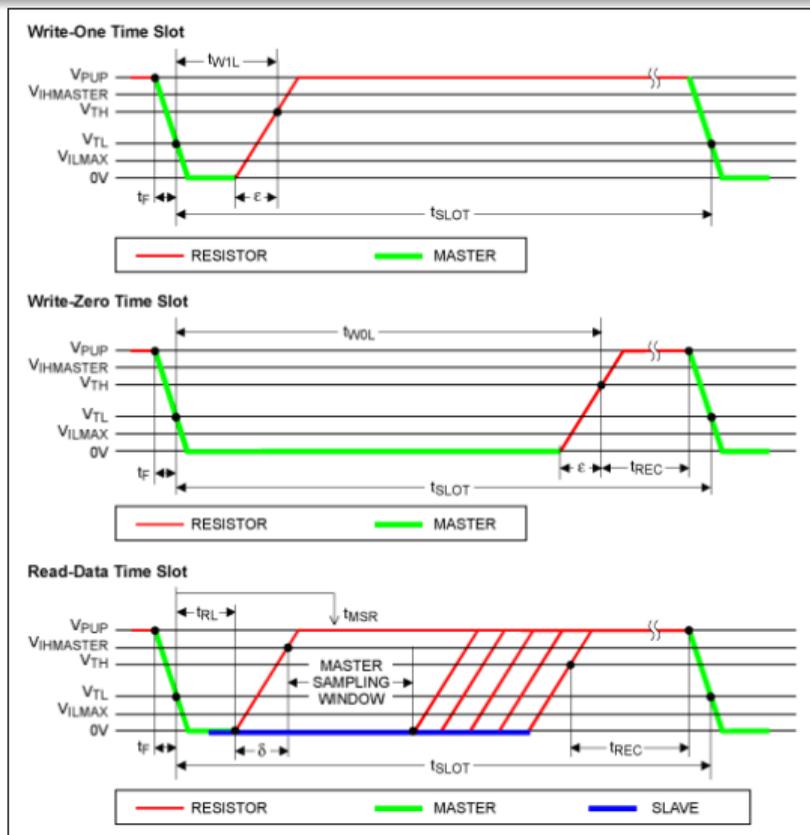
Una vez que el Master libera la línea, entra en modo de recepción. Ahora el bus 1-Wire se lleva a V_{PUP} a través de la resistencia de pullup. Cuando se cruza el umbral, V_{TH} , el Slave espera t_{PDH} y luego transmite un pulso de presencia tirando de la línea hacia abajo durante t_{PDL} .

Timing de Bus



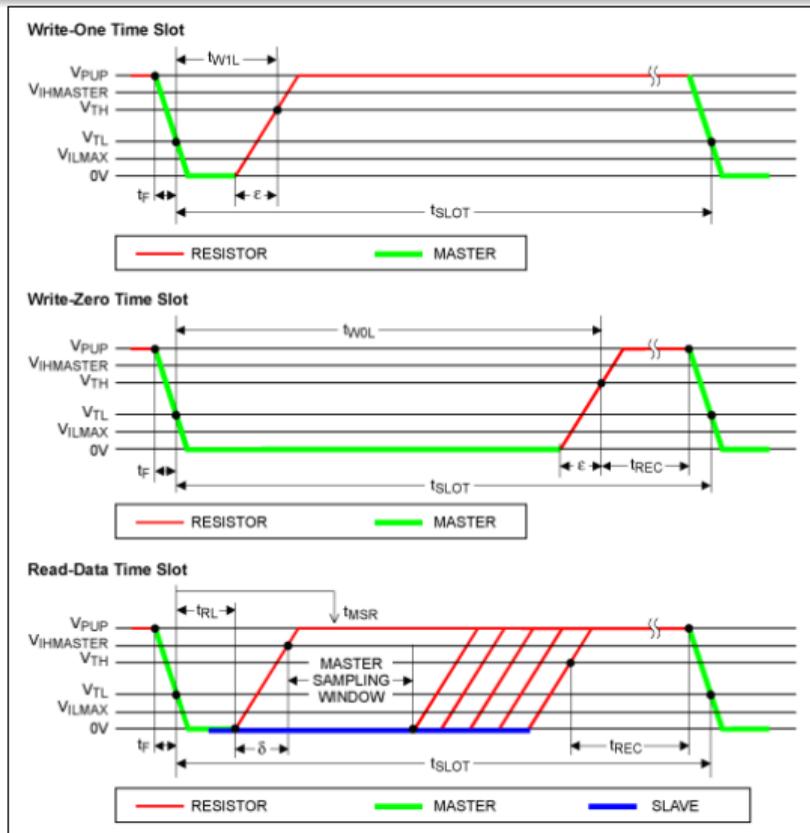
Para detectar un pulso de presencia, el Master debe probar el estado lógico de la línea 1-Wire en t_{MSP} . La ventana t_{RSTH} debe ser al menos la suma de t_{PDHMAX} , t_{PDLMAX} y t_{RECMIN} . Inmediatamente después de que caduca t_{RSTH} , el Slave está listo para la comunicación de datos.

Timing de Bus



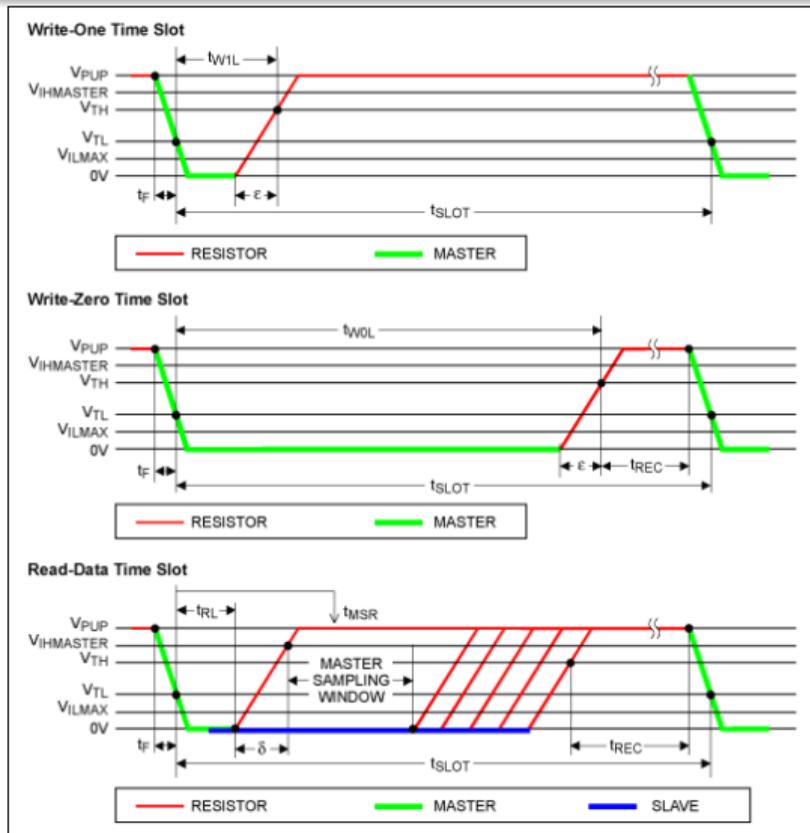
Para un intervalo de tiempo de escritura uno, la tensión en la línea de datos debe haber cruzado el umbral V_{TH} antes de que expire el tiempo bajo de escritura uno, t_{W1LMAX} .

Timing de Bus



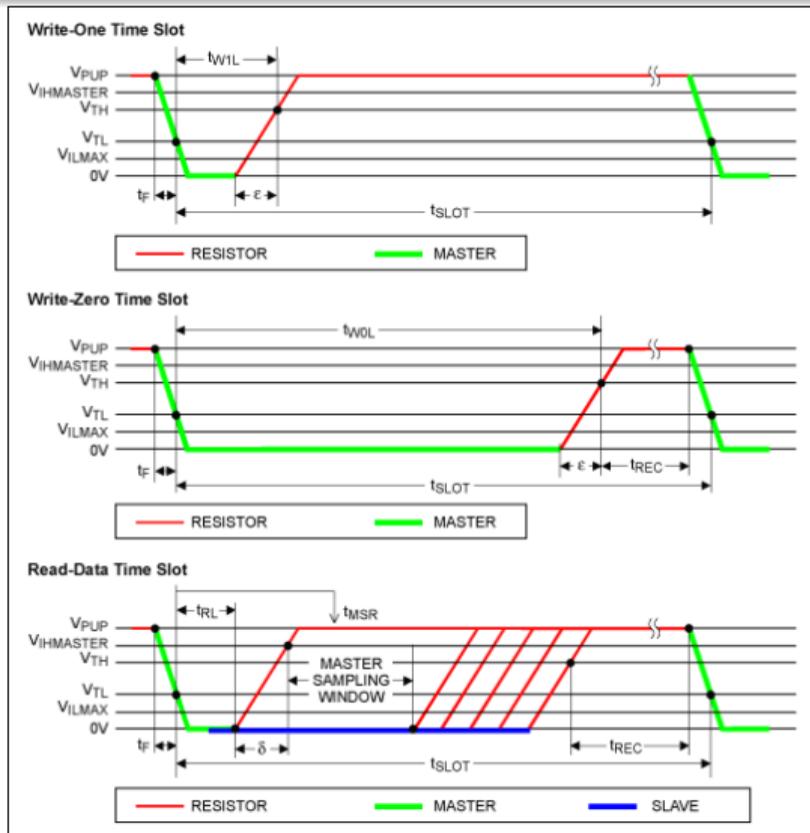
Un intervalo de tiempo de lectura de datos comienza como un intervalo de tiempo de escritura única. La tensión en la línea de datos debe permanecer por debajo de V_{TL} hasta que expire el tiempo de lectura baja, t_{RL} .

Timing de Bus



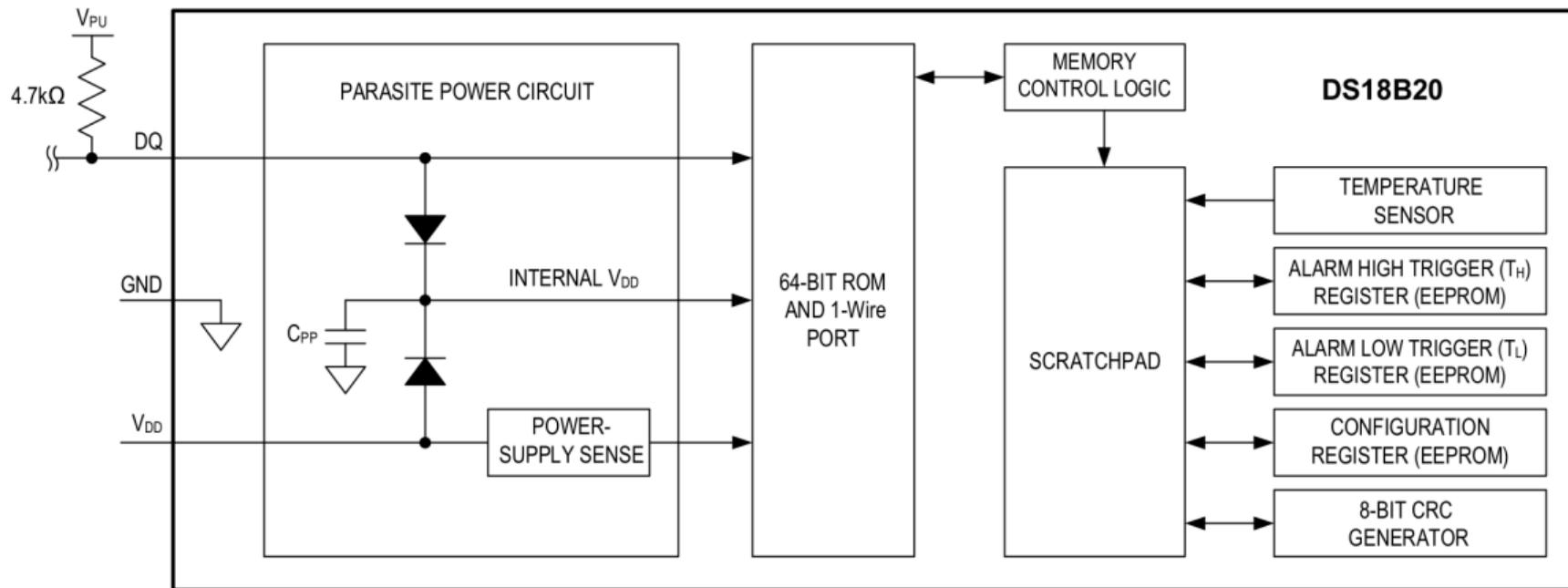
Durante la ventana t_{RL} , cuando responde con un 0, un Slave comienza a bajar la línea de datos; su generador de tiempo interno determina cuándo termina este pulldown y la tensión comienza a subir de nuevo.

Timing de Bus



Cuando responde con un 1, un Slave no mantiene baja la línea de datos y la tensión comienza a subir tan pronto como termina t_{rL} . La suma de $t_{rL} + \delta$ (tiempo de subida) en un lado y el generador de temporización interno del Slave en el otro lado definen la ventana de muestreo del Master, t_{MSRMIN} a t_{MSRMAX} , en la que el Master debe realizar una lectura desde la línea de datos, luego de lo cual debe esperar hasta que caduque t_{SLOT} .

Ejemplo: DS18B20



1-Wire: Recursos adicionales

Lecturas y Documentación adicional:

- APPLICATION NOTE 74. Reading and Writing 1-Wire Devices Through Serial Interfaces. Maxim Integrated part of Analog Devices
- TUTORIALS 148: Guidelines for Reliable Long Line 1-Wire Networks. Maxim Integrated part of Analog Devices
- Reference Design of a 1-Wire® Bidirectional Voltage-Level Translator for 1.8V to 5V. Maxim Integrated part of Analog Devices