

Bus I2C (<https://www.i2c-bus.org>)

□ Índice de la presentación

- Introducción: Historia
- Criterios de Uso
- Características
- Ejemplo de Uso
- Introducción al uso en prácticas

También conocido genéricamente como
“**Two Wire Interface**” (**TWI**)
(<https://www.i2c-bus.org/twi-bus/>)



Actualizado: 1/03/2020

Algo de Historia: origen del bus I2C....

- ❑ Fue desarrollado a principios de los años 80 por Philips Semiconductors
- ❑ Lo denominó bus “Inter-Integrated Circuits” → IIC → I2C
- ❑ Su nombre es el acrónimo de la explicación literal de su propósito original
- ❑ La motivación de partida original fue crear una manera sencilla y barata de comunicar la CPU y los periféricos que empezaban a montarse dentro de los Televisores de forma creciente.

Algo de Historia: ¿cuál ha sido su evolución?

□ Los modos de transferencia de datos del bus I2C

- **Modo Estándar,** **Standard-mode (Sm)** hasta 100 kBits/Seg.
- **Modo Rápido,** **Fast-mode (Fm)** hasta 400kbits/Seg.
- **Modo Rápido plus,** **Fast-mode-plus (Fm+)** hasta a 1 Mbits/Seg.
- **Modo Alta velocidad,** **High-Speed-mode (Hs-mode)** hasta 3,4 Mbits/Seg.

Algo de Historia: ¿cuál ha sido su evolución?

- ❑ El modo High-speed (Hs-mode) ofrece una solución ideal para los sistemas de tecnología mixta, donde las altas velocidades y la variedad de voltajes (5 V, 3 V o menor) son comúnmente usados (Combinado con una función de desplazamiento del nivel de voltaje)
- ❑ El modo Hs es compatible con todos los sistemas existentes del bus I2C, incluyendo el estándar original (S-mode) y el modo Fast (F-mode)
- ❑ **Sistemas de distintas velocidades pueden coexistir**
 - todos los dispositivos en modo Hs se comunicarán bidireccionalmente a toda velocidad mientras se utilicen los modos F/S (Fast/Standard) de velocidades con las partes más lentas del sistema.

Introducción al Bus I2C

□ Índice de la presentación

- Introducción: Historia
- Criterios de Uso
- Características
- Ejemplo de Uso
- Introducción al uso en prácticas

Criterios para Utilizar I²C

- ❑ Transferencias “lentas”: El sistema que utiliza este Bus no debe requerir una alta tasa de transferencia de datos
- ❑ Cuando la restricciones de coste de conexión entre los varios dispositivos dentro del sistema es un factor importante.
- ❑ Cuando es necesario conectar múltiples elementos: Un sistema consiste en al menos un Controlador y varios sistemas periféricos como memorias, display, sensor de temperatura, ... Etc.

Ejemplos: Equipos con el bus I2C...

- ❑ Los procesadores de señal en televisores (LA7610, TA1223, DTC810,...)
- ❑ Memorias 24Cxx
- ❑ Codificadores de video de reproductores de DVD (SAA 7128, TC 90A32F,...)
- ❑ Preamplificadores de video en monitores (KB 2502)
- ❑ Etc...

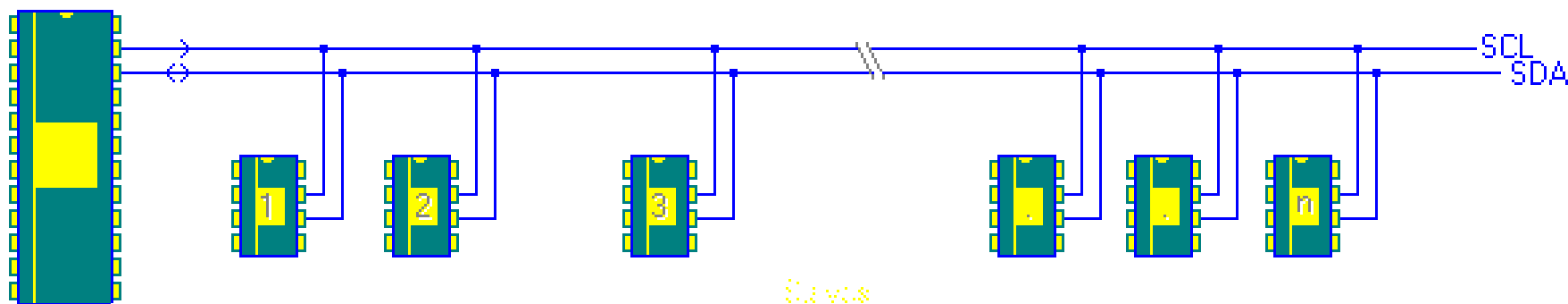
Introducción al Bus I2C

□ Índice de la presentación

- Introducción: Historia
- Criterios de Uso
- Características
- Ejemplo de Uso
- Introducción al uso en prácticas

Características del bus I2C...

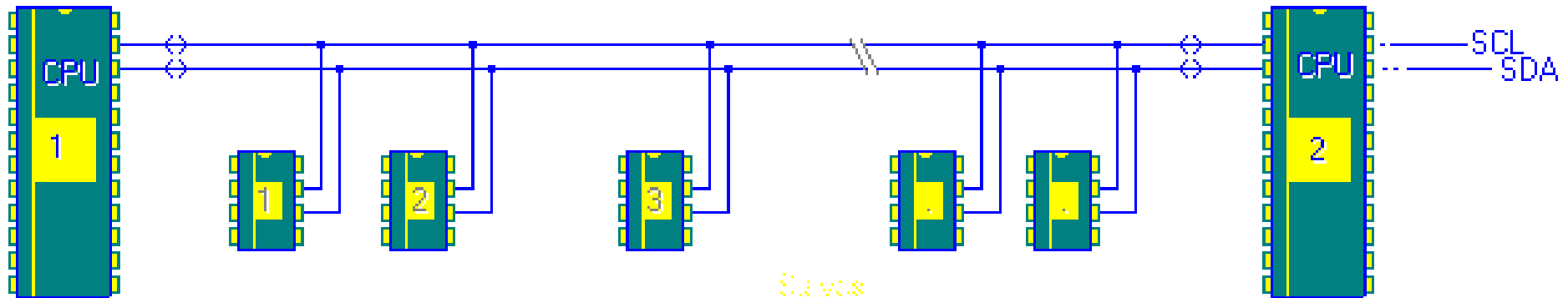
- ❑ Se necesitan solamente dos líneas, la de **datos (SDA)** y la de **reloj (SCL)**.



- ❑ Cada dispositivo conectado al bus tiene un **código de dirección** seleccionable mediante software.
- ❑ Existe permanentemente una **relación Máster / Slave** entre el controlador y los dispositivos conectados.
- ❑ El **protocolo de transferencia** de datos y direcciones posibilita diseñar sistemas completamente **definidos por software**.
- ❑ Los datos y direcciones se transmiten con palabras de 8 bits.

Características del bus I2C...

- ❑ El bus **permite la conexión de varios Masters**, ya que el protocolo incluye un **mecanismo detector de colisiones**.

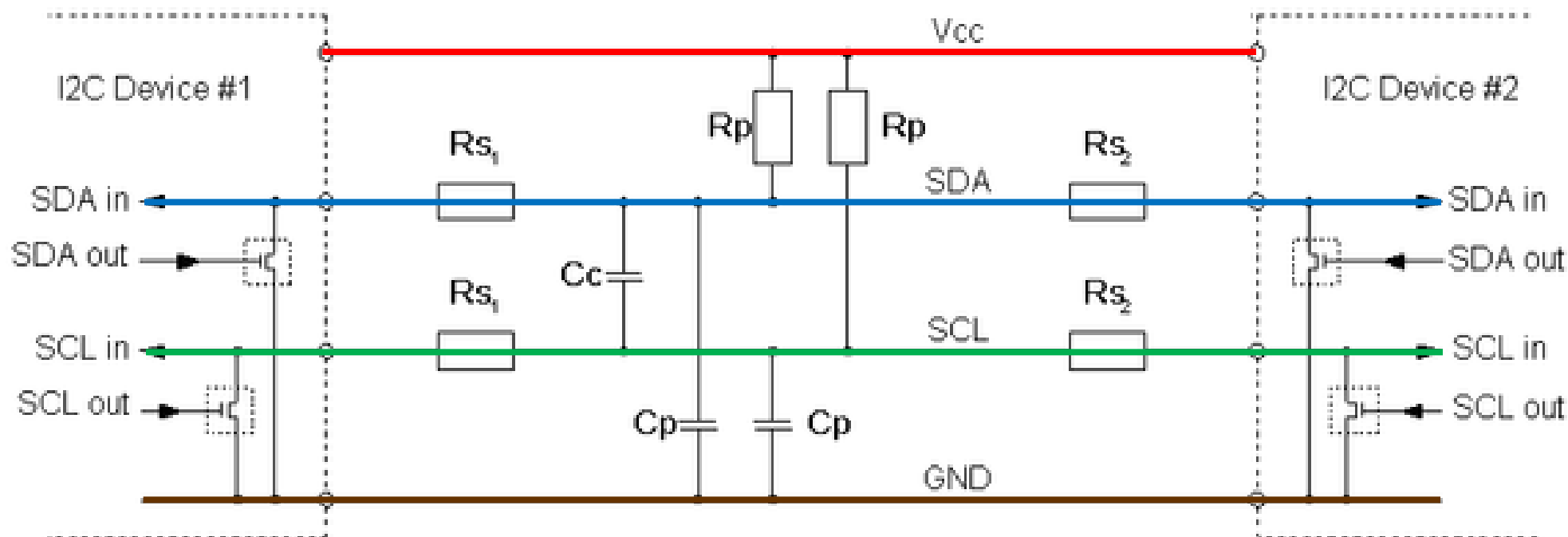


- La cantidad de dispositivos que se pueden conectar al bus está limitada, solamente, por la **máxima capacidad permitida de 400 pF**.

- El **protocolo de transferencia** de datos y direcciones posibilita diseñar sistemas completamente **definidos por software**.

- ❑ La eficacia del sistema depende de la correcta selección de la naturaleza de los dispositivos y de la interconexión de la estructura del bus.

Conexionado:



Introducción al Bus I2C

□ Índice de la presentación

- Introducción: Historia
- Criterios de Uso
- Características
- Ejemplo de Uso
- Introducción al uso en prácticas

Terminología básica del Bus I2C

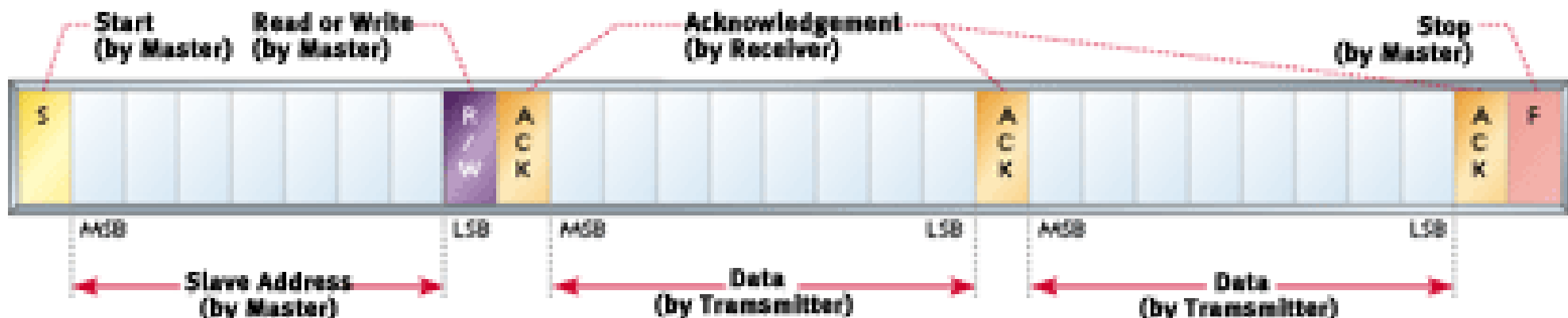
<u>Términos</u>	<u>Descripción</u>
Transmisor	El dispositivo que envía datos al Bus
Receptor	El dispositivo que recibe datos desde el Bus
Master (Maestro)	El dispositivo que inicia una transferencia, genera las señales del reloj y termina un envío de datos
Slave (Esclavo)	El dispositivo direccionado por un master
Multi-Master	Más de un master puede controlar el bus al mismo tiempo sin corrupción de los mensajes
Arbitraje	Procedimiento que gestiona el uso del Bus, tal que solo uno de los peticionarios es el que toma el control, evitando la corrupción de mensajes. Procedimiento distribuido
Sincronización	Procedimiento para sincronizar las señales del reloj de dos o mas dispositivos

Instrucciones en el I2C Bus

- ❑ Para realizar una operación de acceso a un esclavo sobre el Bus I2C solo son necesarios seis simples pasos, (enviar o recibir información):
1. Un bit de Inicio (Start)
 2. Direccionamiento (7-bit o 10-bits)
 3. Un bit de R/W que define si el esclavo es transmisor o receptor
 4. Un bit de reconocimiento
 5. Mensaje dividido en bytes (8 bits + 1 bit de reconocimiento)
 6. Un bit de finalización (Stop)
-

Formato

- ❑ El Máster envía la **condición de Start** y un **código de dirección de un esclavo**
 - 7 bits de dirección + un bit de R/W (W=0→escritura; R=1→Lectura)
- ❑ Una transferencia de datos siempre acaba con una **condición de Stop** generado por el máster
 - sin embargo, si un máster todavía desea direccionar a otro(s) esclavo(s), puede generar repetidamente condiciones de Start y sin generar primero la condición de stop.

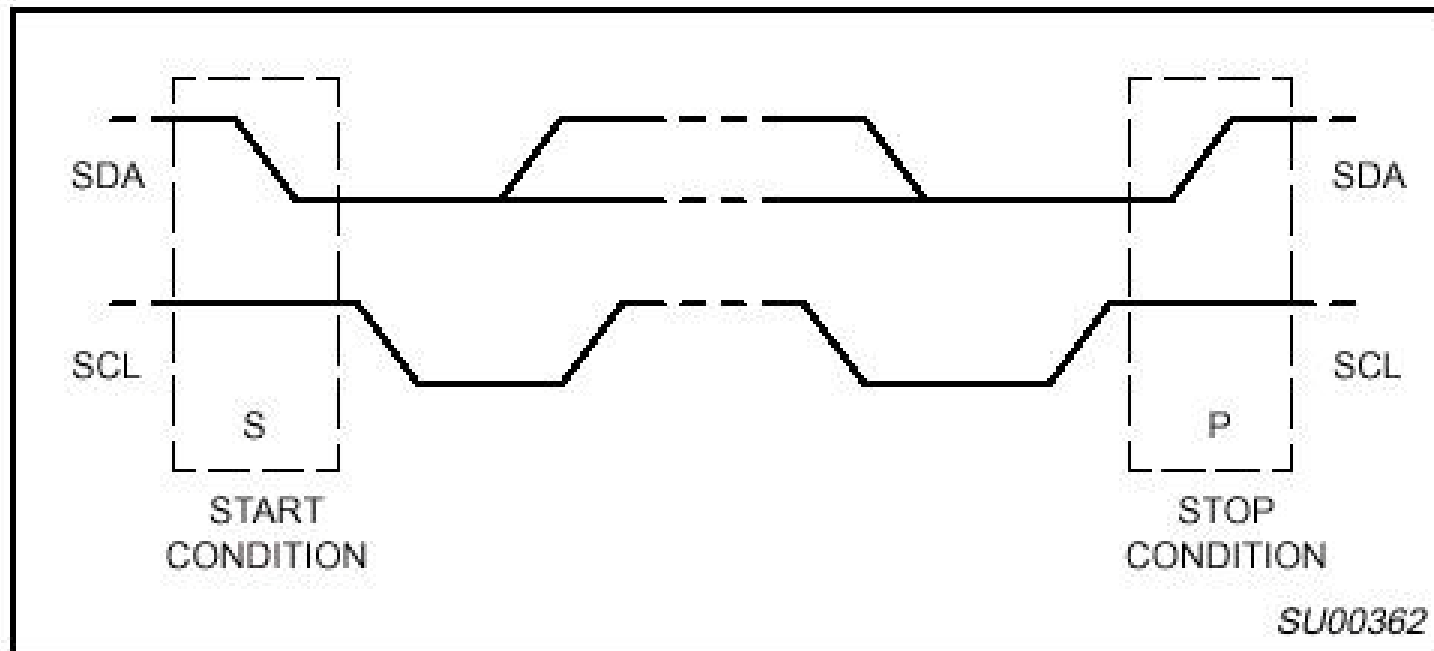


Condiciones de Inicio (Start) y Parada (Stop)

Dentro del proceso de transferencia de datos en el bus I2C hay dos situaciones básicas que son el **inicio (start)** y el **final (Stop)** de toda transferencia de datos. Estas son:

- **INICIO (START)** - Una transición de "1" a "0" (caída) en la línea de datos (SDA) mientras la línea del reloj (SCL) esta a "1".
 - **PARADA (STOP)** - Una transición de "0" a "1" (ascenso) en la línea de datos (SDA) mientras la línea de reloj (SCL) esta a "1"
-

Condiciones de Inicio (Start) y Parada (Stop)

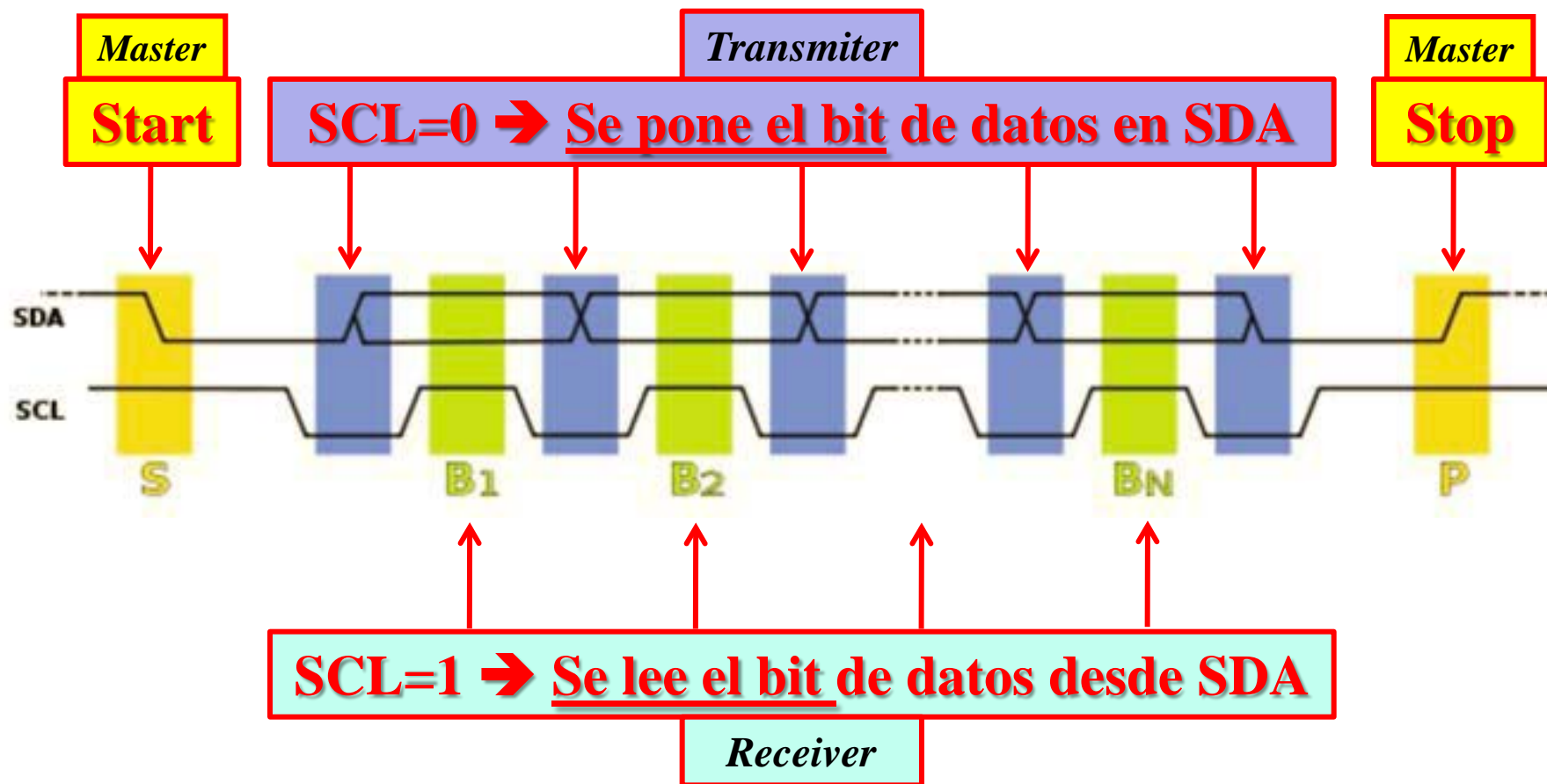


Las condiciones de **Inicio y Stop** son siempre generadas por el **Máster**.

- El Bus I2C se considera ocupado después de la condición de Inicio.
- El Bus se considera libre de nuevo después de un cierto tiempo tras la condición de Stop.

Transfiriendo datos

- El numero de bytes que se envíen a la línea SDA no tiene restricción.



Procedimiento general de escritura:



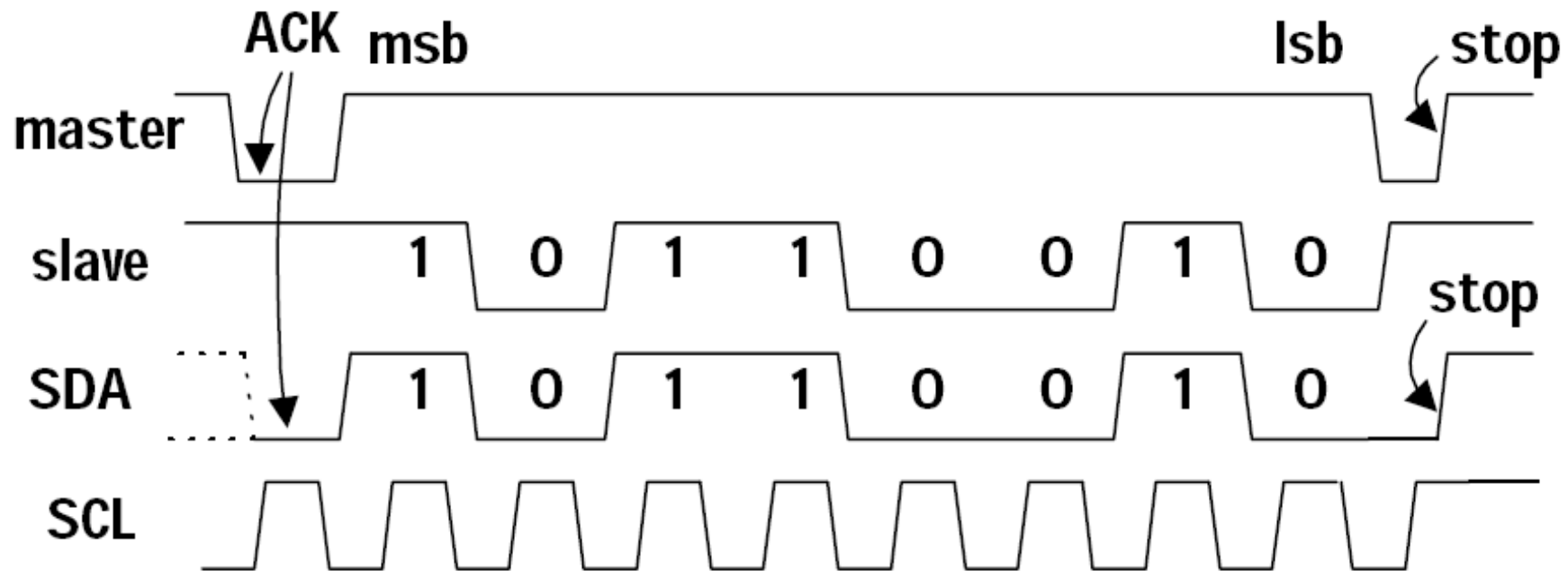
Procedimiento general de lectura:



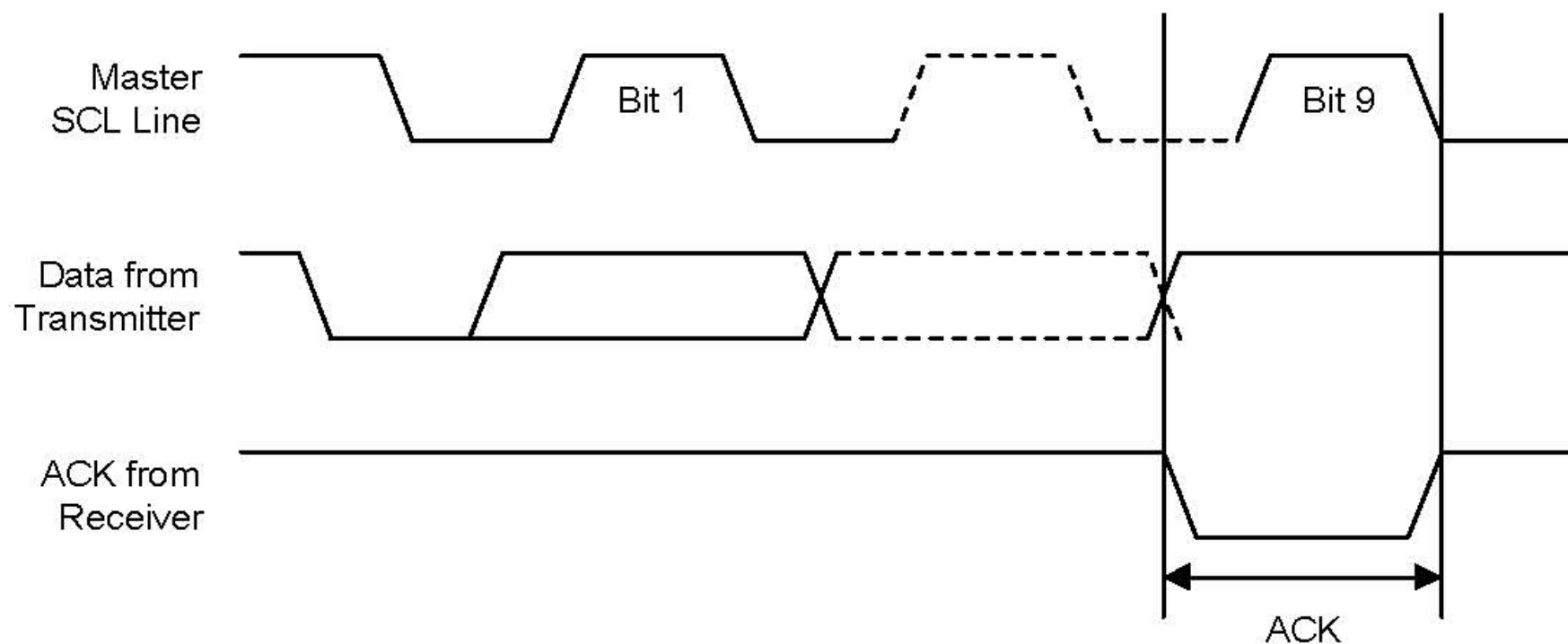
Manejo de las señales....

La del maestro, la del esclavo y el “and” en el bus

- ❑ Ejemplo: El *Maestro* está leyendo y el *Esclavo* transmitiendo
- ❑ SDA = AND de las dos señales maestro-esclavo



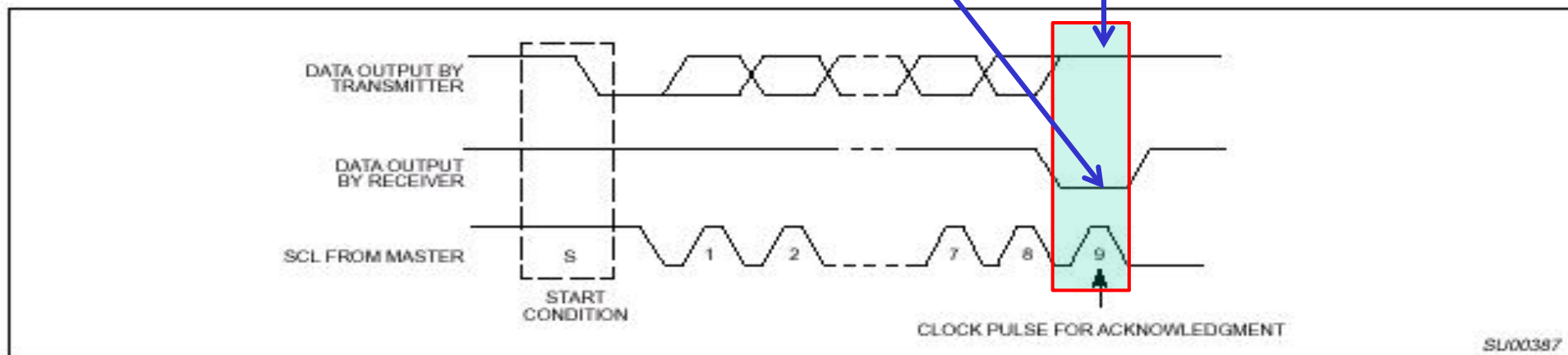
Manejo de las señales en general....



Bibliografía básica:

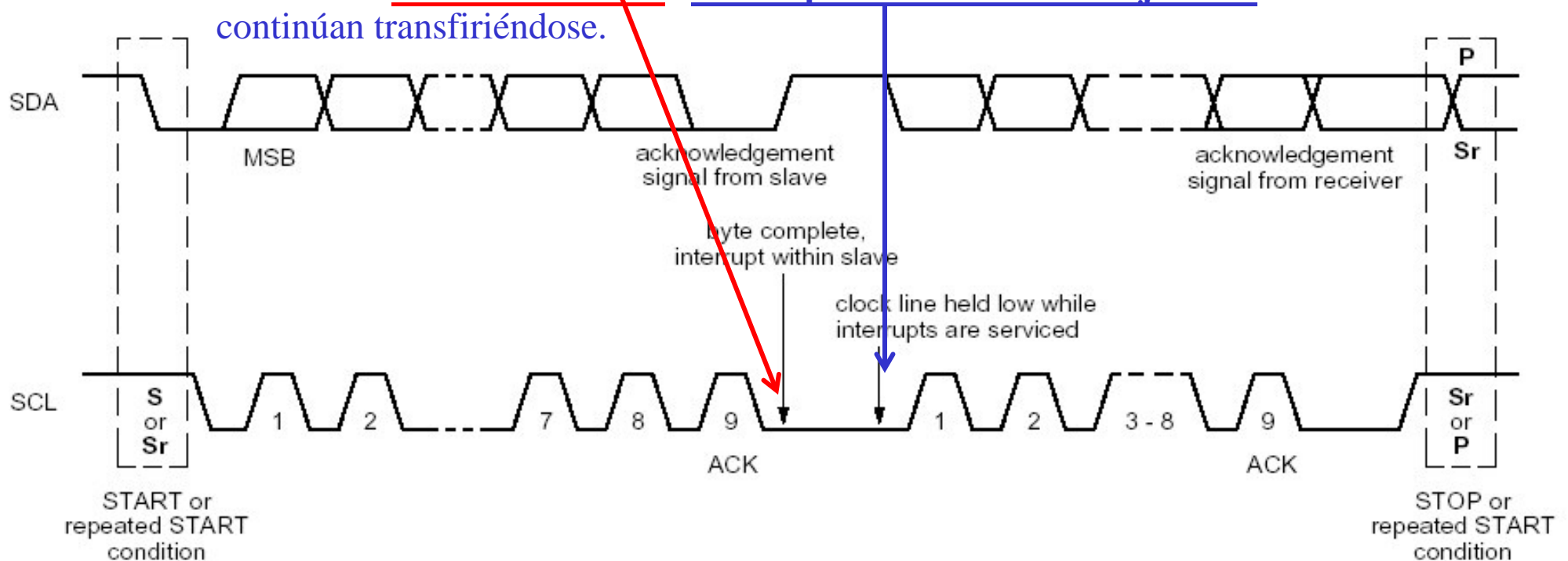
Reconocimiento (Ack)

- ❑ El bit de reconocimiento es obligatorio en la transferencia de datos.
- ❑ El pulso de reloj correspondiente al bit de reconocimiento (ACK) es generado por el Master
 - El Transmisor desbloquea la línea SDA → pone un "1" durante el pulso de reconocimiento (para poder “escuchar” al receptor)
 - El receptor responde → debe poner a "0" la línea SDA durante el pulso ACK de modo que siga siendo "0" durante el tiempo que el master tiene SCL a "1" (durante el ACK = ciclo 9).



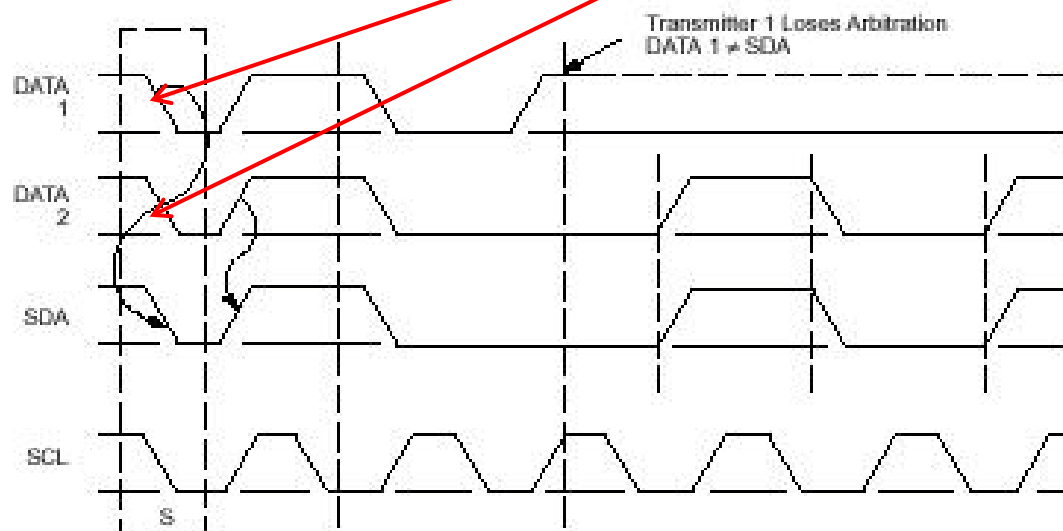
Transfiriendo datos (clock stretching)

- ❑ Si un dispositivo esclavo no está listo para recibir o transmitir un byte de datos....
 - Pone la línea SCL a "0" para forzar al Master a entrar y permanecer en un **estado de espera**.
 - El Master debe monitorizar continuamente si cuando pone un '1', hay un nivel '1' en la línea SCL, leyéndola. Si se lee un '0' ➔ **esperar** monitorizando
 - Cuando el esclavo esté listo: desbloquea la línea de reloj SCL ➔ Los datos continúan transfiriéndose.

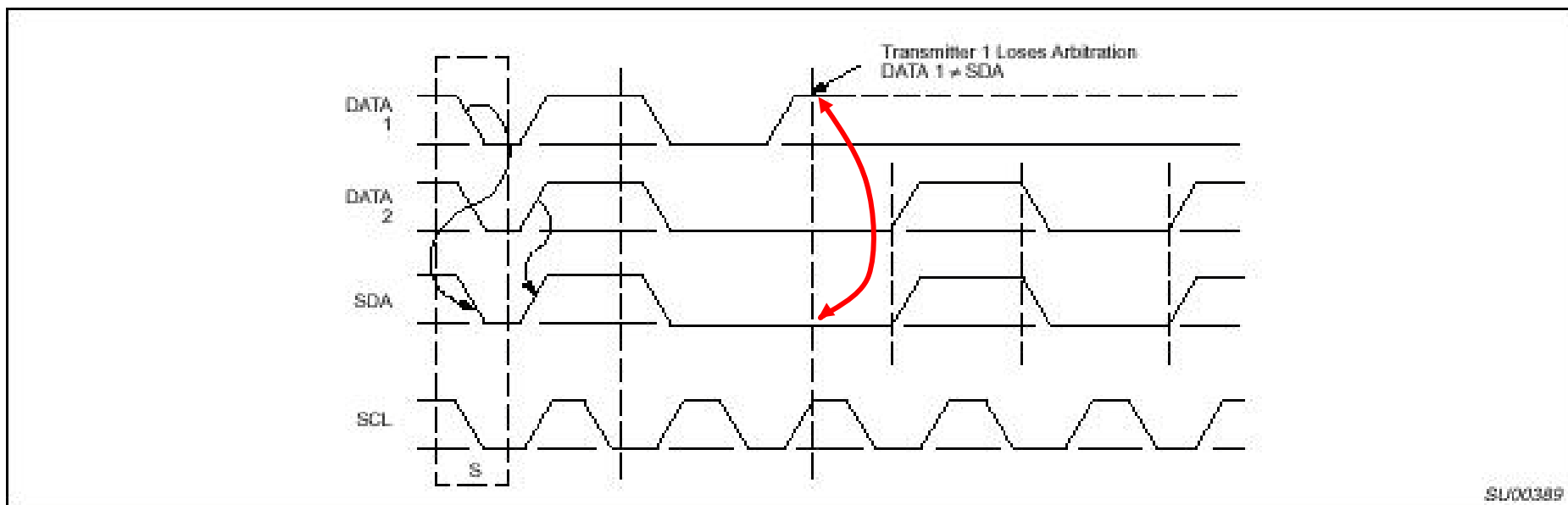


Arbitraje

- ❑ Un master puede iniciar la transmisión solo si el bus esta libre → SCL,SDA =1,1
- ❑ Puede darse el caso que dos o más masters generen simultáneamente una condición de Inicio en el bus lo que da como resultado una condición de Inicio general →



Procedimiento de Arbitraje entre dos Master



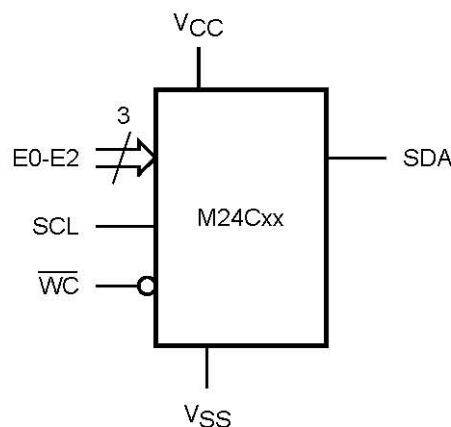
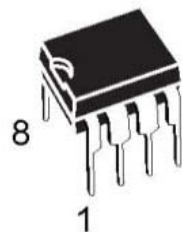
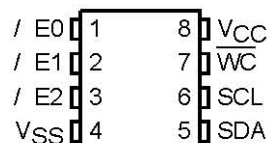
- ❑ Cada Master debe comprobar siempre si el bit de datos que transmite junto a su pulso de reloj, coincide con el nivel lógico en la línea de datos SDA.
- ❑ El sistema de arbitraje actúa sobre la línea de datos SDA...
 - mientras la línea SCL esta a nivel "1", el master que transmite un nivel "1" en SDA, pierde el arbitraje a favor del master que envía un nivel "0" a la línea de datos SDA.
- ❑ El que ha "cedido" el uso del bus deberá permanecer en esta situación hasta que se detecte la condición de Stop generada por el master que se hizo cargo del Bus.

Introducción al Bus I2C

□ Índice de la presentación

- Introducción: Historia
- Criterios de Uso
- Características
- Ejemplo de Uso
- Introducción al uso en prácticas

El dispositivo => M24C01: 1kbit = 128 Bytes



AI02033

Table 1. Signal Names

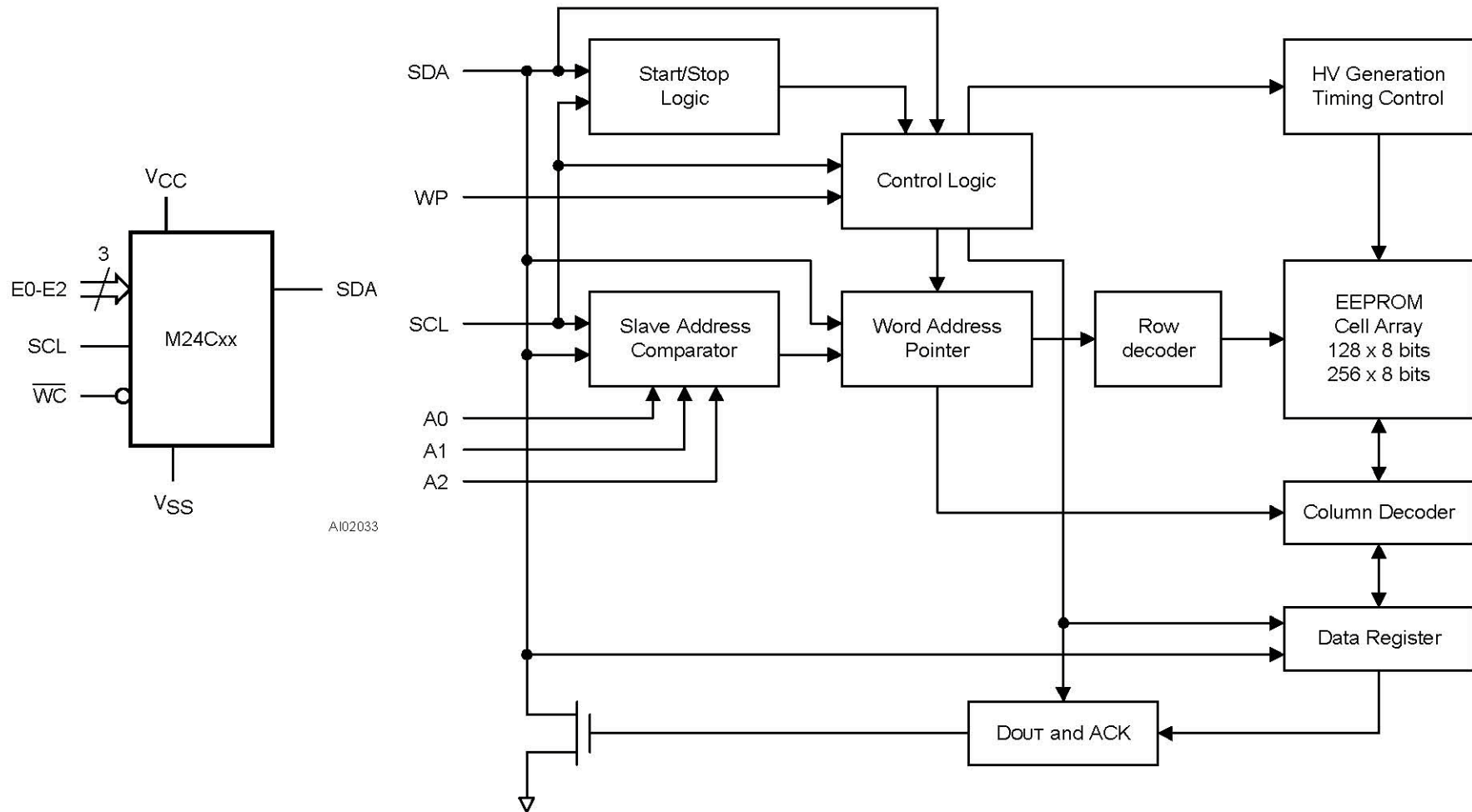
E0, E1, E2	Chip Enable
SDA	Serial Data
SCL	Serial Clock
\overline{WC}	Write Control
V _{CC}	Supply Voltage
V _{SS}	Ground

Table 3. Operating Modes

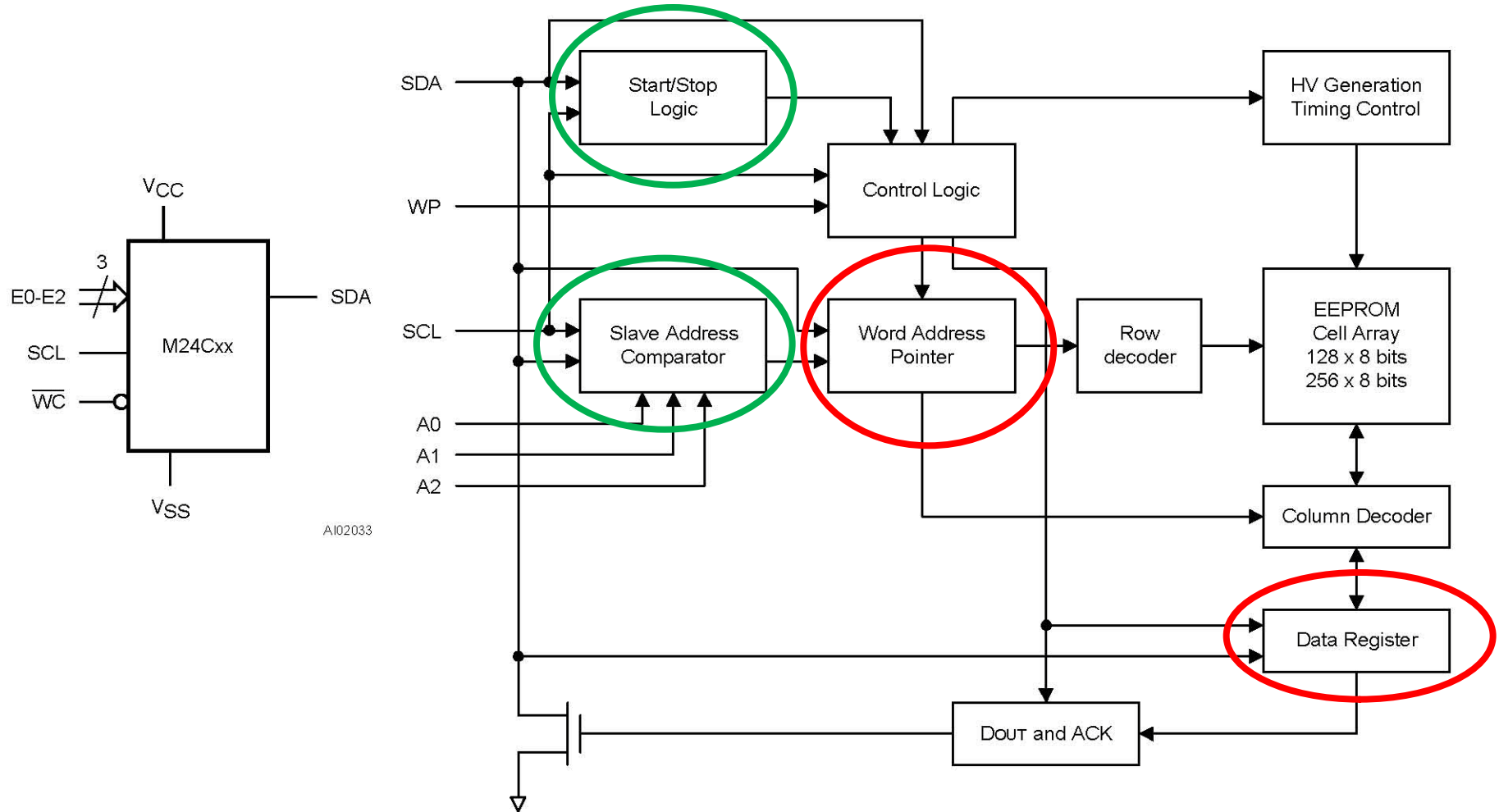
Mode	R \overline{W} bit	\overline{WC} 1	Bytes	Initial Sequence
Current Address Read	1	X	1	START, Device Select, R \overline{W} = 1
Random Address Read	0	X	1	START, Device Select, R \overline{W} = 0, Address
	1	X		reSTART, Device Select, R \overline{W} = 1
Sequential Read	1	X	≥ 1	Similar to Current or Random Address Read
Byte Write	0	V _{IL}	1	START, Device Select, R \overline{W} = 0
Page Write	0	V _{IL}	≤ 16	START, Device Select, R \overline{W} = 0

Note: 1. X = V_{IH} or V_{IL}.

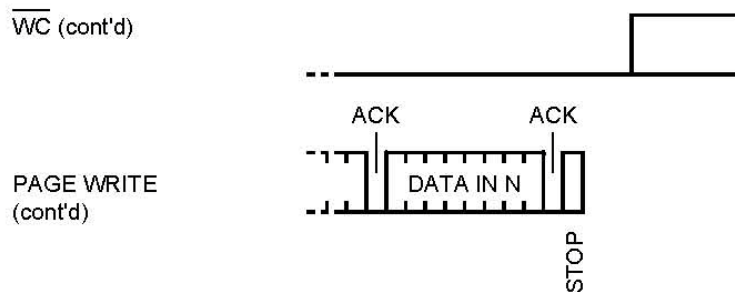
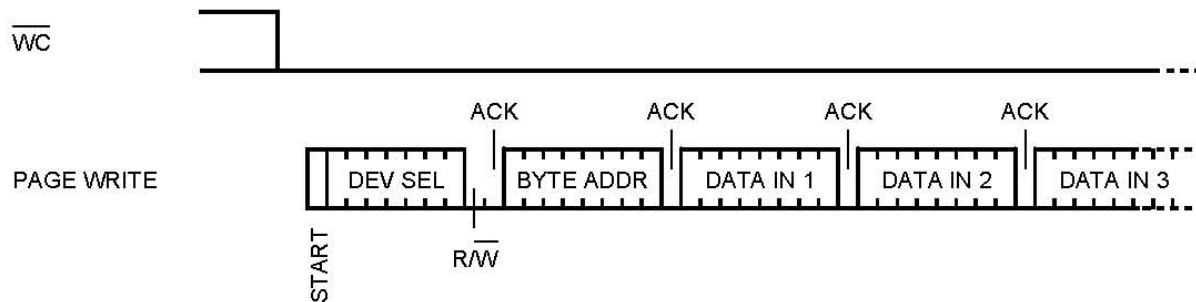
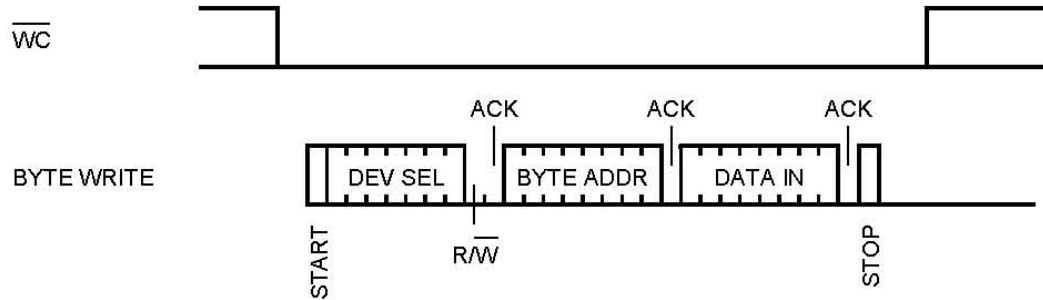
Estudio del dispositivo => M24C01: 1kbit = 128 Bytes



Estudio del dispositivo => M24C01: 1kbit = 128 Bytes



Operaciones de escritura en el M24C01:



□ Dos modos de escritura:

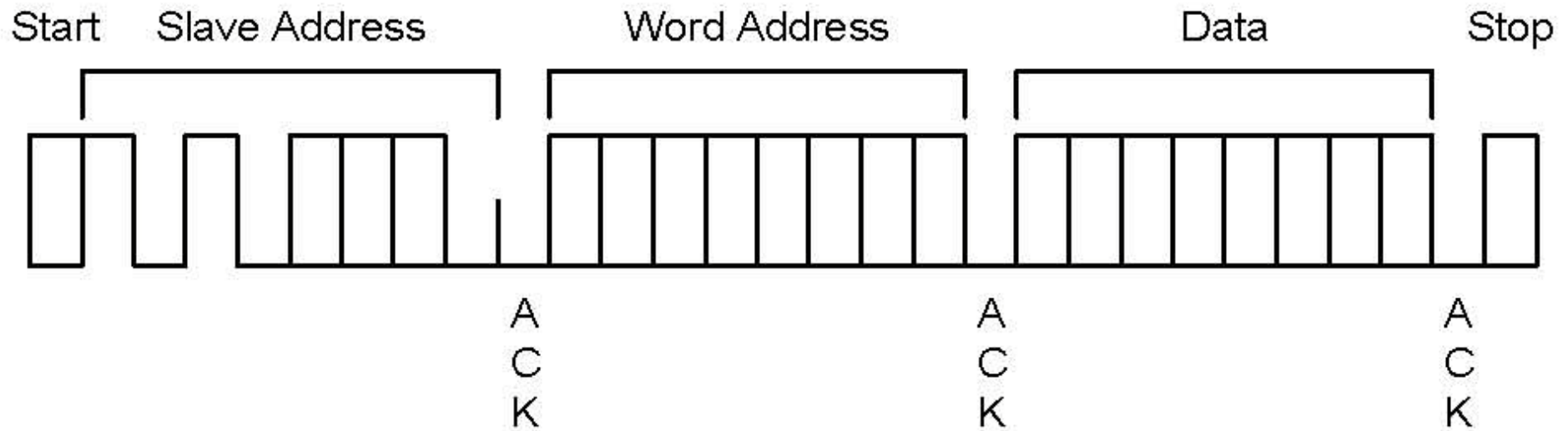
- **Byte-Write** = almaceno solo un byte (data-in) en una posición (byte-addr)
- **Page-Write** = Almaceno una secuencia de bytes a partir de una dirección (byte-addr)
 - **Máximo = 16 bytes**

□ Escritura muy lenta:

- ¡10 milisegundos!
- Hay que esperar a que esté disponible
- Señalado con el ACK

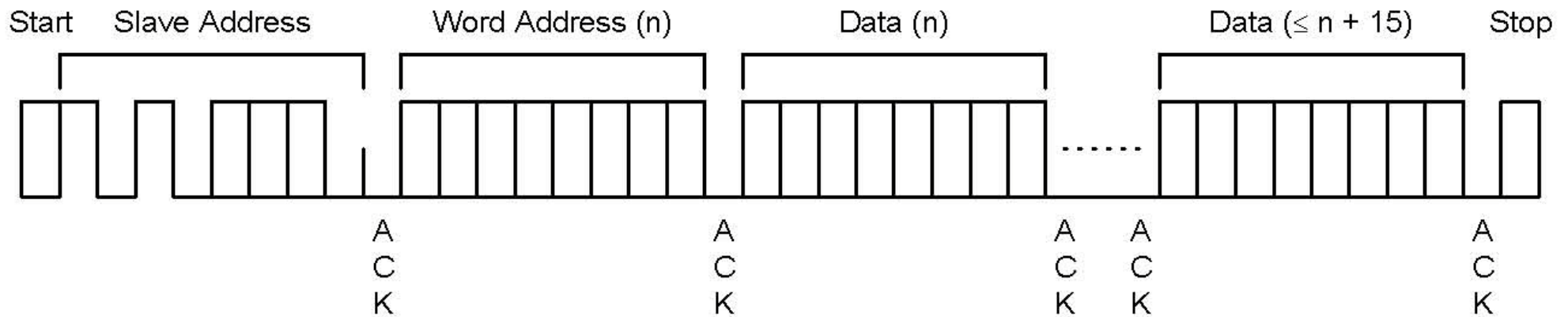
AI02804B

Detalle de “Byte-Write”:



- ❑ En la siguiente operación habrá que esperar a que el ACK nos indique que terminó la operación interna de escritura

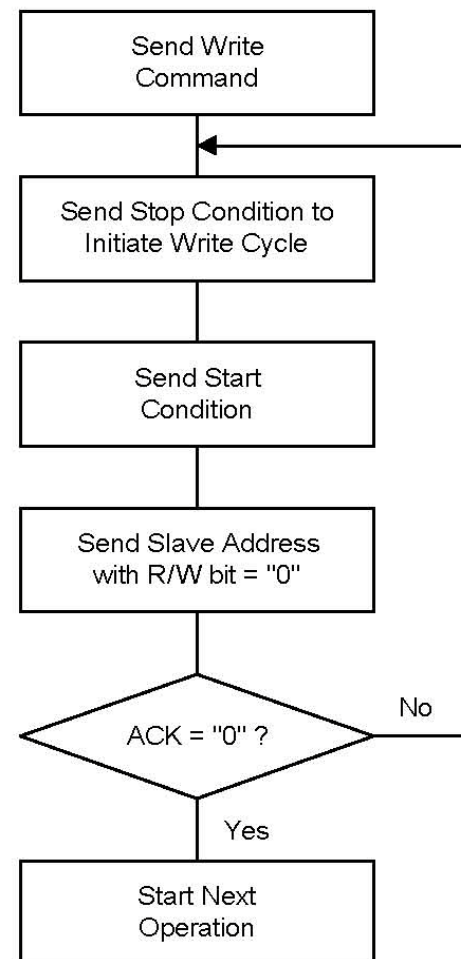
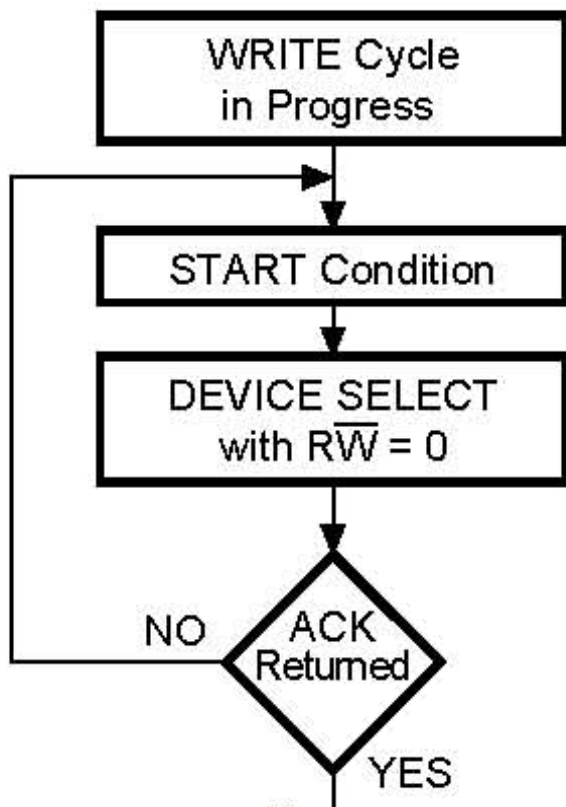
Detalle de “Page-Write”:



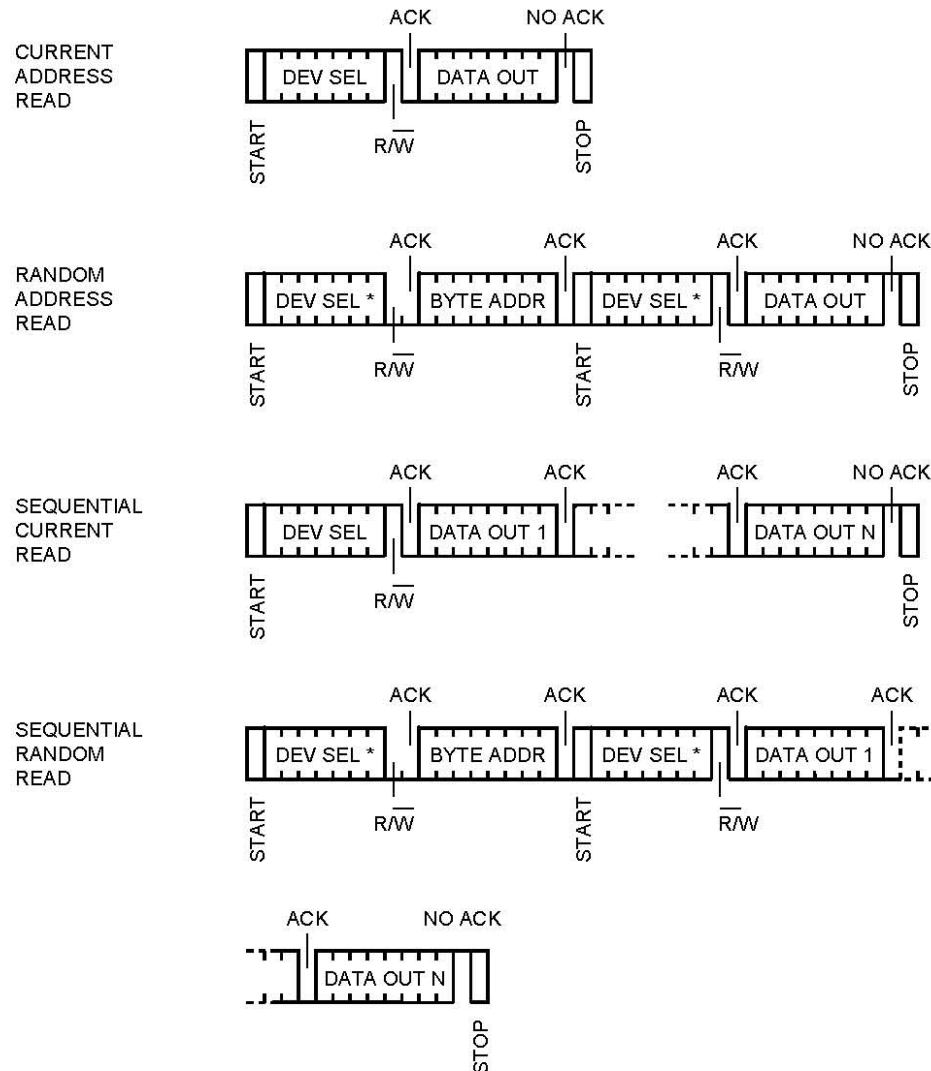
- ❑ **!!!Máximo = 16 datos!!!** (algunos chips = 8)
- ❑ En la siguiente operación habrá que esperar a que el ACK nos indique que terminó la operación interna de escritura

Bibliografía básica:

Algoritmo de espera:

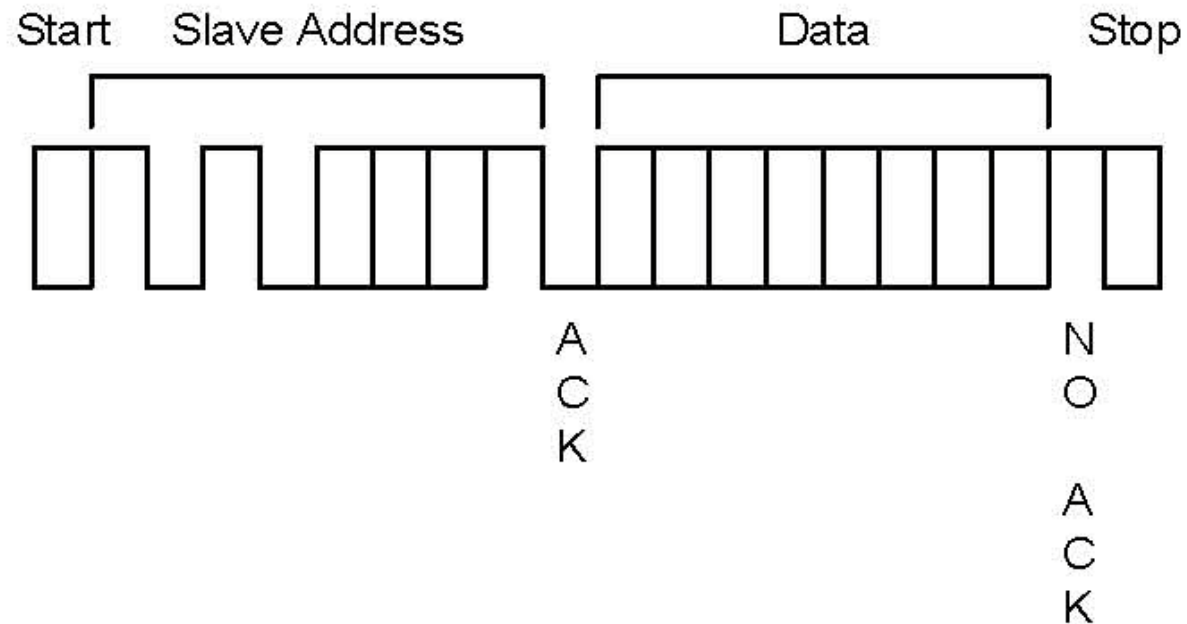


Operaciones de lectura en el M24C01:

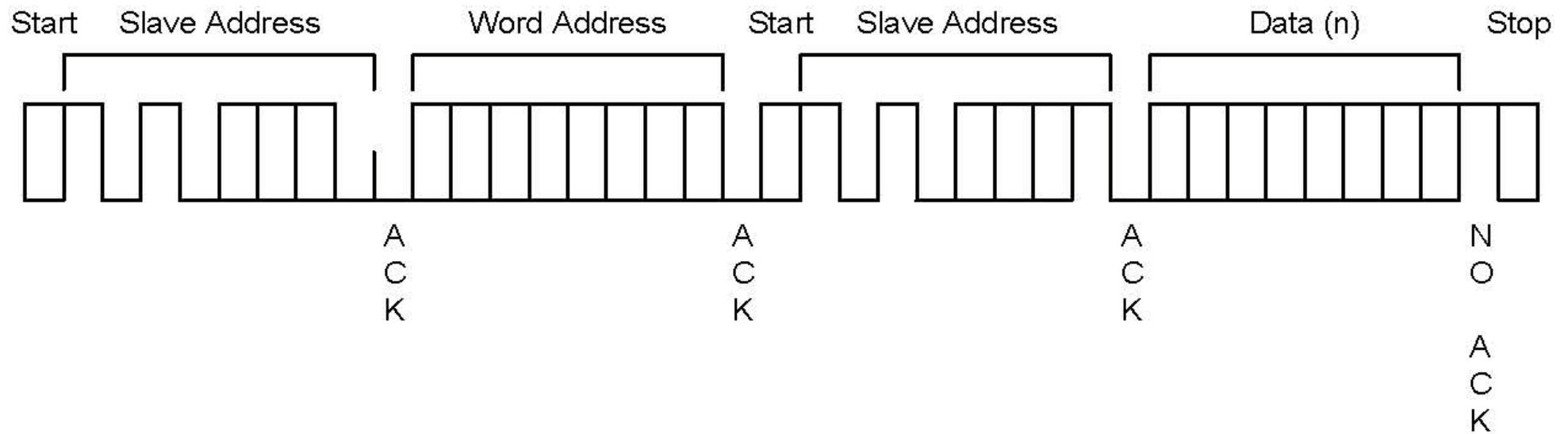


□ Tres modos de escritura:

- **Current-Addr.-Read** = acceso a la posición a la que apunta el contador interno
- **Random-Addr.-Read** = acceso solo a un byte (data-in) en la posición (byte-addr)
- **Sequential-Addr.-Read** =
 - **Random** → Lectura de una secuencia de bytes a partir de una dirección (byte-addr)
 - **Current** → Lectura de una secuencia de bytes a partir de la dirección del contador interno
 - **Si llegamos a la dirección máxima y seguimos leyendo se continua desde la posición cero**

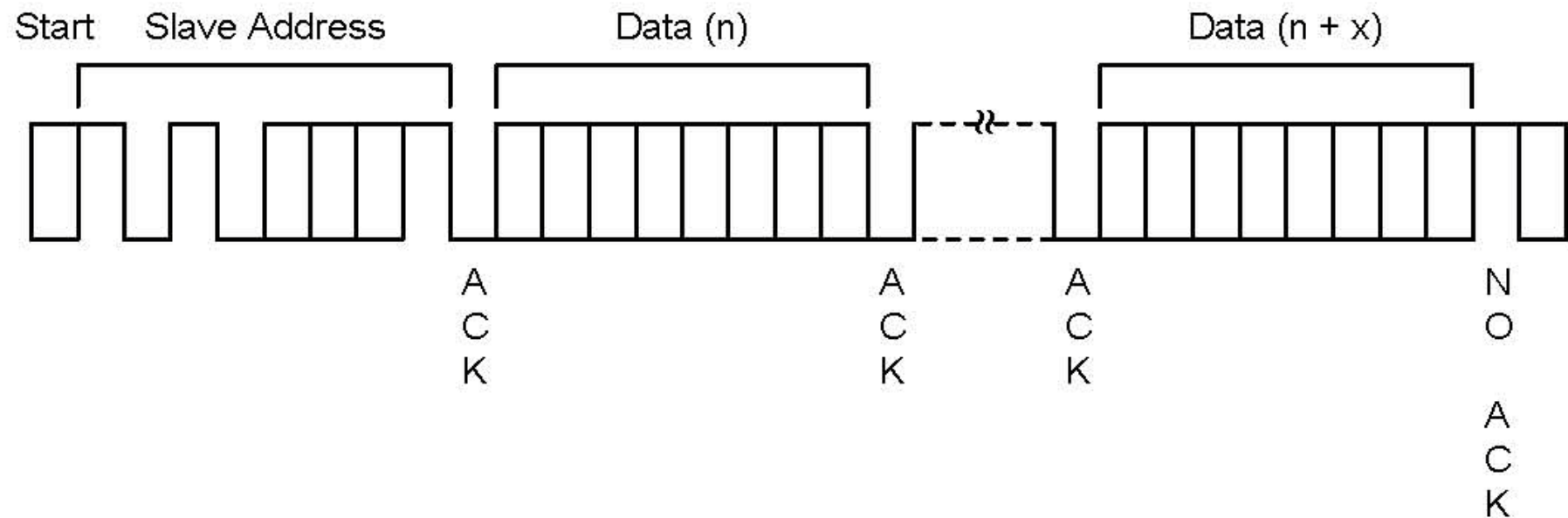


Bibliografía básica:



Bibliografía básica:

Detalle de “Sequential Read Operation”:



Bibliografía básica:

Fin de la presentación

¿Preguntas?