Interfaz I^2C

Transmisión de Datos

Ing. Luis Di Pinto (ldipinto@herrera.unt.edu.ar)

http://www.microprocesadores.unt.edu.ar/transmision/

Repaso Esquema General (2da parte)

- En esta segunda parte de la materia, vamos a enfocarnos en un conjunto de interfaces seriales.
  - Algunas de las más usadas y populares.



Temas que veremos

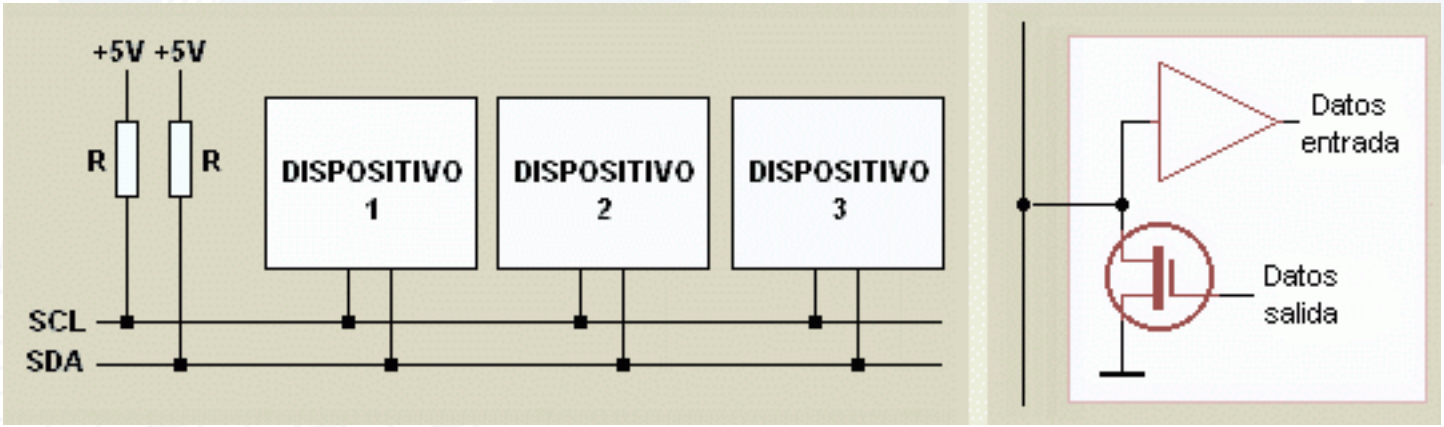
- Características generales.
- Esquema de conexión.
- Protocolo de comunicación.
- Esquemas de direccionamiento.
- *Clock Stretching*.
- Arbitraje del bus.
- Comparación con SPI.

Introducción

- ***Inter-Integrated Circuit***
  - IIC, I2C o I^2C.
  - Es una interfaz ampliamente difundida.
- Protocolo desarrollado por Philips en 1982 que usa un bus **sincrónico**.
- Cada dispositivo tiene **una dirección única**.
- La comunicación se efectúa entre un controlador (master) y un periférico (slave).
  - Es half-dúplex.
- Define un sistema de **arbitraje** que permite que más de un dispositivo pueda ser controlador del bus (en diferentes momentos).

Características del bus I^2C

- Utiliza **dos líneas** con salidas a colector abierto, cada una con una resistencia de pull-up externa conectada a Vcc.
  - SCL: *Serial Clock Line*
  - SDA: *Serial Data Line*
- Esto define una estructura de bus que permite la conexión en paralelo de una **cantidad indefinida** de dispositivos.



La conexión en detalle

- Los valores de las resistencias de pull-up no son críticos y pueden variar entre 1,8 KΩ y 47 KΩ.
  - Una resistencia baja aumenta el consumo pero mejora la inmunidad al ruido y los tiempos de los flancos.
  - El valor mas común es de 10 KΩ.
- La cantidad máxima de dispositivos que se pueden conectar al bus está dada por la suma de las capacidades de entrada de los dispositivos.
  - La capacidad total del bus incluidos los dispositivos y las líneas de conexión no deben exceder los 400 pF.
  - En la práctica, son pocos metros.

Funcionamiento general

- Un controlador (Master) inicia una transferencia destinada a un periférico en particular (Slave).
- **El periférico es seleccionado a partir de una dirección única** que el controlador transmite por el bus.
  - Normalmente, de 7 bits. Puede ser también de 10 bits, pero es poco común.
- El controlador indica tambien si la operación será una lectura (‘1’) o una escritura (‘0’).
- El controlador suministra siempre la señal de reloj, tanto para lecturas como para escrituras.

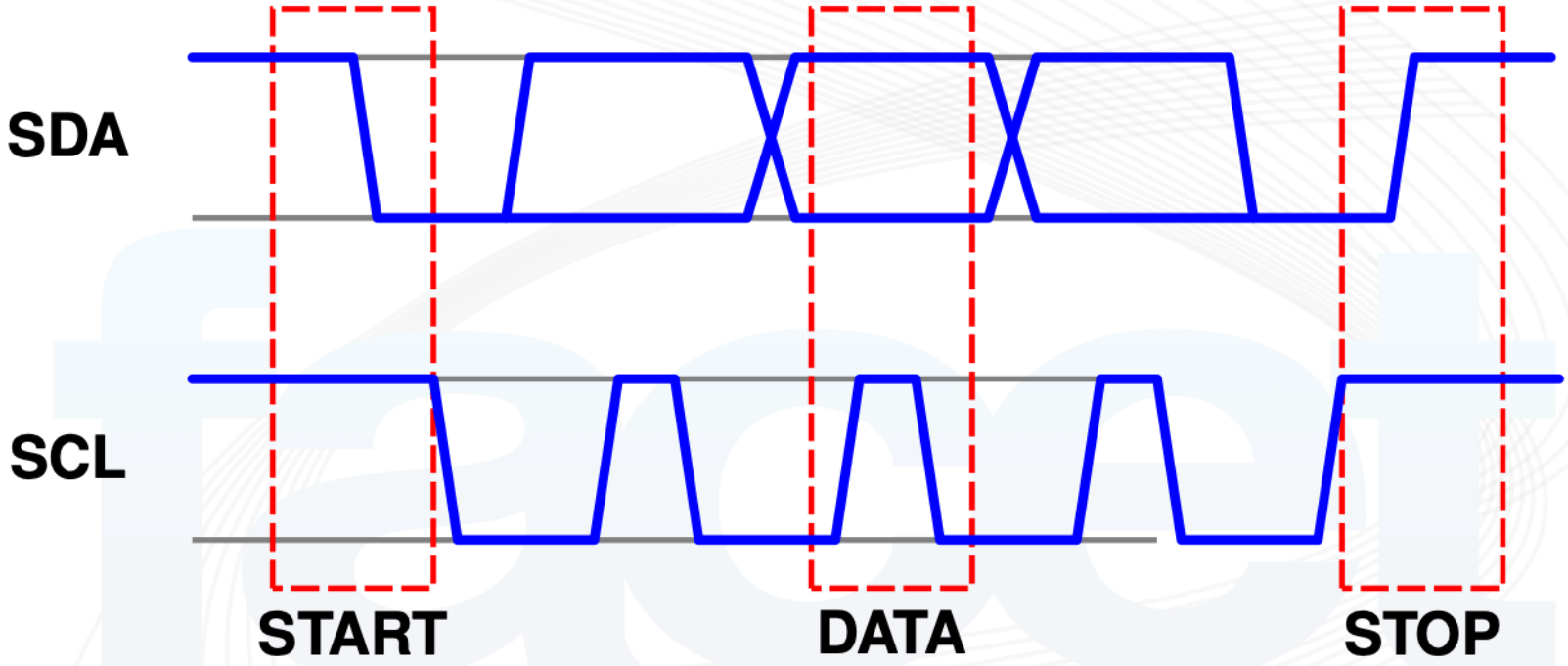
Funcionamiento general

- La línea SDA transmite los bits de datos, comenzando por el más significativo.
- La línea SCL transporta la señal de reloj.
- La transferencia se realiza por bytes y puede ser tan larga como se desee.
- El controlador también termina la transacción, ya sea una lectura o una escritura.

Funcionamiento detallado

- Mientras el bus está libre las lineas SDA y SCL se mantienen en el nivel lógico alto.
- Con el bus libre, cualquier controlador puede iniciar una transferencia generando una condición de START.
  - La condición de START se obtiene al bajar la línea SDA mientras se mantiene SCL en alto.
- Para terminar una transferencia un controlador debe generar una señal de STOP.
  - La condición de STOP se obtiene al subir la línea SDA mientras SCL se mantiene en alto.
- Los cambios de bit en la línea SDA se realizan siempre con el reloj SCL en bajo.
  - Cuando SCL es alto, los datos son válidos.
  - Las condiciones de START y de STOP se distinguen porque no cumplen esta condición.

Diagrama de tiempos



- Nótese que las condiciones de START y STOP no son bits, sino un esquema de señalización diferente.

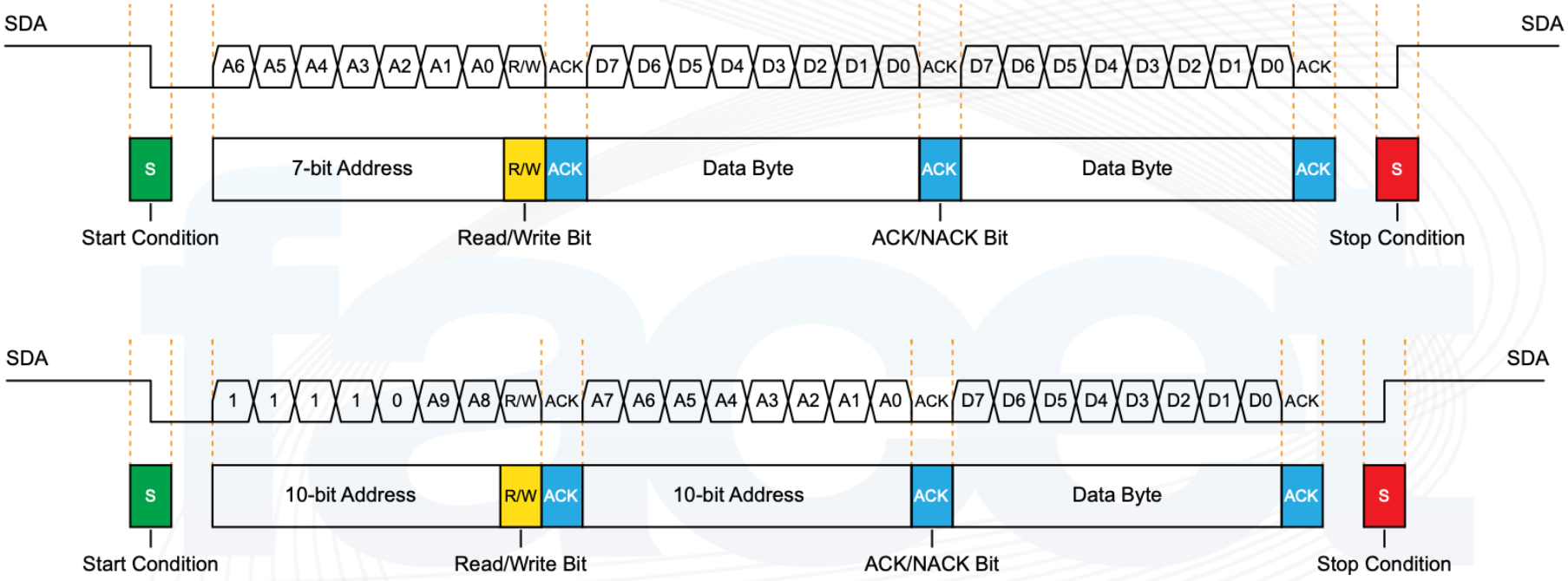
Reconocimiento del dato recibido

- Los datos transmitidos por la línea SDA son siempre bytes (8 bits).
  - Pueden ser múltiples bytes.
- Al final de cada byte el receptor debe generar un bit de confirmación.
- Para esto, el transmisor libera la línea SDA, la que recupera un valor lógico alto.
- El receptor debe llevar la línea SDA a un valor bajo durante el noveno bit.

Direccionamiento del periférico

- Después de la condición de START el controlador transmite una cabecera con la dirección única del periférico.
- Esta cabecera puede ser de 1 o 2 bytes según las direcciones sean 7 o 10 bits.
  - Si las direcciones son de 10 bits, se comienza la cabecera con los bits 11110, (*Address Extension)*
  - Por lo tanto, ninguna dirección de 7 bits puede comenzar con 11110.
- La cabecera también incluye 1 bit para indicar si la transferencia es una lectura o una escritura.

Direccionamiento del periférico



Escrituras de datos

- Si la operación es una escritura, el Controlador envía los bytes de datos después de la cabecera.
- El periférico debe generar el bit de reconocimiento después de cada byte recibido.
- La transferencia termina por decisión del Controlador o porque el periférico no envía el reconocimiento del byte recibido.

Lecturas de datos

- Si la operación es una lectura, el Controlador continua generando pulsos de reloj para que el periférico envíe los datos.
- El Controlador debe generar el bit de reconocimiento después de cada byte recibido.
- Para el último byte no genera un bit de reconocimiento, lo que informa al dispositivo que se completó la lectura de datos.

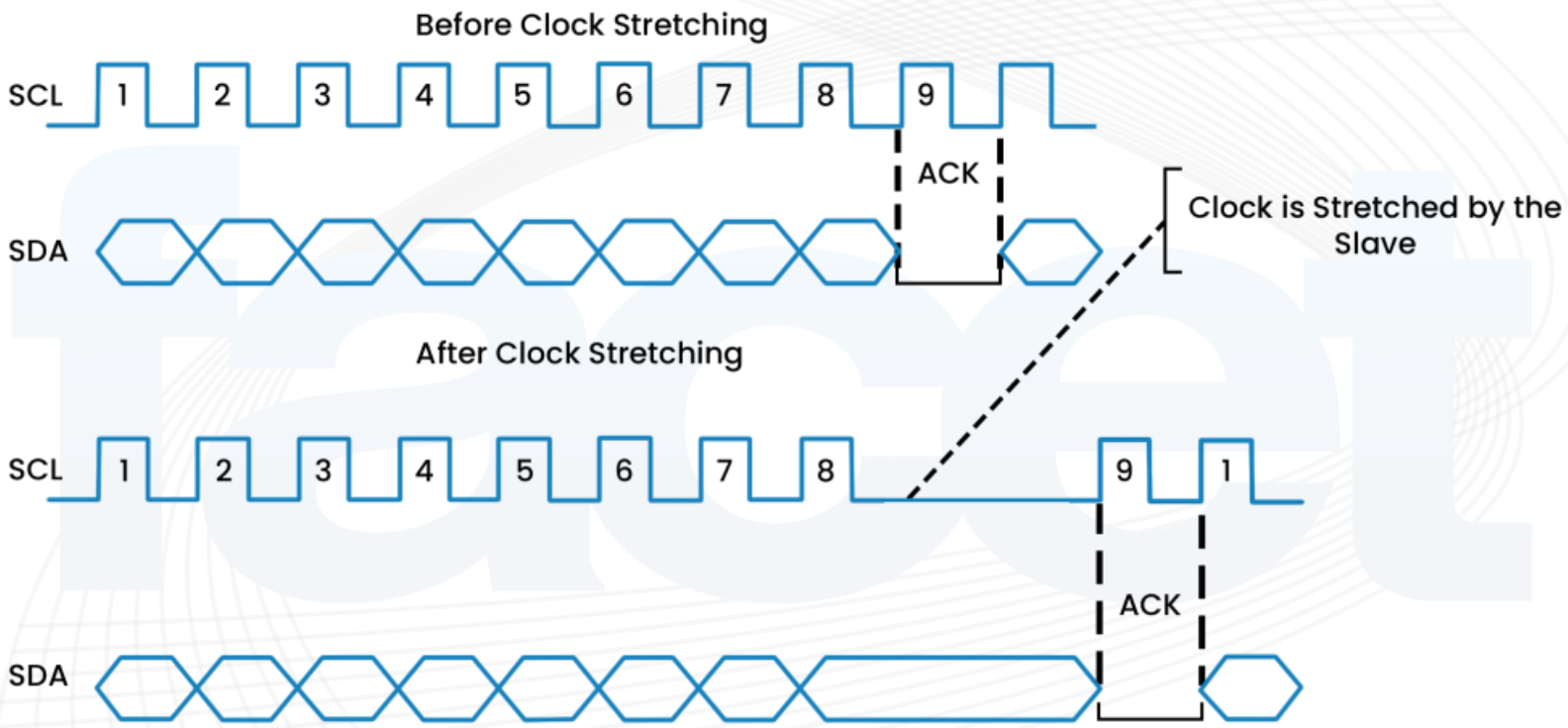
Eficiencia de la comunicación

- Supongamos que un Controlador desea escribir 32 bits en un dispositivo, y que se usa direccionamiento de 7 bits.
- **¿Cuántos bits realmente se deben enviar?**
- **¿Cuánto es el *overhead* de la comunicación?**
- **¿Cuánto sería la eficiencia de la comunicación?**

Sincronización de reloj

- En una operación de lectura es el Controlador el que genera la señal de reloj.
- Podría suceder que el periférico no esté listo para enviar los datos.
- Para resolver esto, el Controlador revisa el valor de la linea SCL después de liberarla para que vuelva al estado alto.
- El periférico puede retener la linea SCL en nivel bajo y entonces el controlador espera hasta que el periférico la libere, para así poder leer el dato.
- Esto se conoce como *Clock Stretching*.
  - Única vez que el periférico maneja SCL.

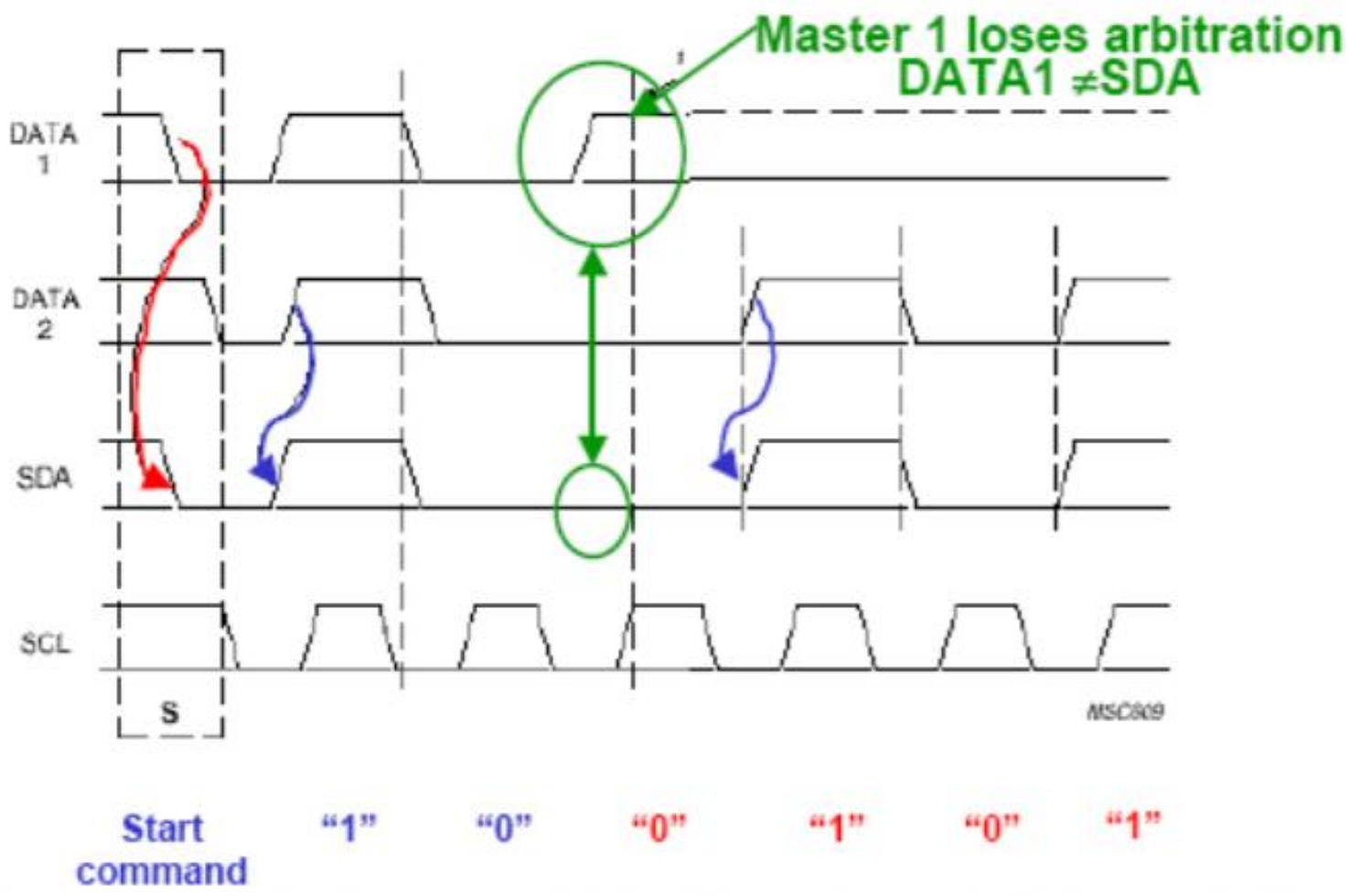
Sincronización de reloj



Arbitraje del Bus

- Un Controlador monitorea en el bus las condiciones de START y de STOP, para no iniciar una transferencia si otro Controlador está utilizando el bus.
- Pero puede suceder que dos Controladores intenten iniciar una transferencia simultáneamente.
- El arbitraje se realiza sobre la línea SDA cuando la línea SCL está en alto.
  - El controlador que quiere transmitir un '1' pierde el arbitraje y retira el pedido.

Arbitraje del Bus



Velocidades de operación

- Existe cinco opciones de velocidad:
  - Modo Estándar (Sm): 100 Kbps.
  - Modo Rápido (Fm): 400 Kbps.
  - Modo Rápido Plus (Fm+): 1 Mbps.
  - Modo Alta Velocidad (Hs-mode): 3,4 Mbps.
  - Modo Ultra Rápido (UFm): 5 Mbps.
    - Incompatible con los anteriores.
    - Básicamente, de sólo escritura.

Limitaciones y desventajas

- El hecho de ser un bus de colector abierto, que requiere de resistencias de pull-up, hace que el consumo de energía sea alto.
- El direccionamiento es acotado.
- El rango de velocidades es acotado.
- Si un dispositivo se rompe y mantiene una de las líneas en bajo, el bus queda inutilizable.

Comparativa vs SPI

Característica	SPI	I2C
Reloj	Sincrónico	Sincrónico
Modo de transmisión	Full Duplex	Half Duplex
Pines	4	2
Conexión	Directa o encadenada	Bus compartido
Dispositivos	Múltiples	Hasta 128
Masters	1	Múltiples
Velocidad	Alta	Más bajas
Introduce overhead?	No	Si
Control de flujo	No	Sí
Arbitraje	No	Sí

Resumen

- Interfaz I2C: estándar de facto para comunicaciones de corta distancia.
  - Comunicación sincrónica, half dúplex, multipunto, de tipo Master-Slave.
  - Solamente dos líneas: SDA y SCL.
  - Simple y de bajo costo.
  - Bus único, con direccionamiento de 7 bits.
  - Cinco velocidades de operación.
- Protocolo simple, con control de flujo
  - Genera overhead.
- Permite arbitraje del bus.

Agradecimientos

- Las diapositivas de este tema fueron basadas en las realizadas por el Ing. Nicolás Majorel Padilla.