



Módulo 11: Métodos de codificación física de señales

Contenidos del módulo 11

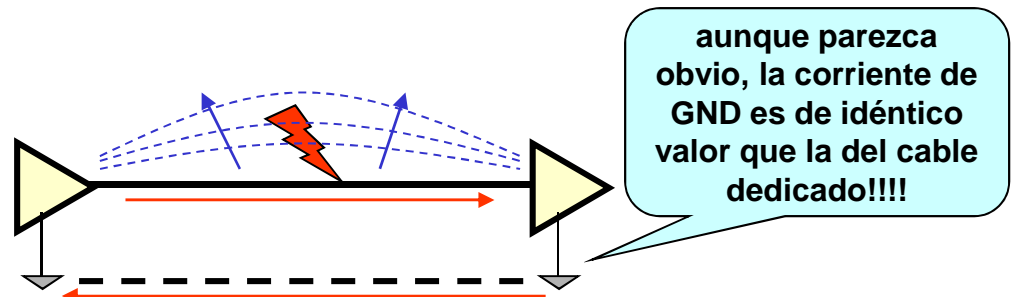
- Comunicación single-ended y diferencial.
 - Ventajas y desventajas.
 - Normas estandarizadas.
- Distintas maneras de incorporar la señal de reloj a los datos:
 - Manchester
 - bit-stuffing
 - AMI
 - HDB3
 - 8B10B.

Interfases “single-ended”

Características generales

Una interfase se dice *single-ended* cuando la información se transmite sobre un único cable dedicado, empleándose como cable de retorno una conexión común (usualmente GND). El tipo de señalización empleada para transmitir los datos depende de múltiples factores:

- ❑ distancia
- ❑ velocidad de los datos (tasa de bits)
- ❑ métodos de modulación.
- ❑ tasa de error aceptable
- ❑ tecnología disponible para la fabricación de receptores y transmisores
- ❑ tipo de interferencias a evitar (tanto el ingreso de interferencia externa como la emisión de interferencia electromagnética)



A priori podrían distinguirse varios grandes grupos

- ❑ de largo alcance: Coaxial (troncal E1 en telefonía, por ejemplo), RS232, RS485
- ❑ simples, de corto alcance y velocidad media: TTL, CMOS, PECL, LVTTTL, LVCMOS
- ❑ de corto alcance y alta velocidad: GTL, GTL+, AGP, HSTL, SSTL
- ❑ de propósitos especiales: buses I2C, CAN, LIN, USB, etc..

Interfases “single-ended” de largo alcance

❑ **Coaxial**: el uso de cable coaxiales para el transporte de datos presenta dos grandes ventajas:

- ❑ El excelente blindaje, inherente a la malla del coaxial puesta a GND
- ❑ El importante ancho de banda, que permite el multiplexado de información por división de frecuencia (FDM)

Su desventaja es el costo y en el caso de uso de FDM los sistemas complejos de modulación y demodulación. Pese a ello era el método de uso habitual en troncales digitales de telefonía (E1/T1) usando modulación AMI o HDB3, aunque ha sido reemplazado por enlaces wireless o de fibra óptica.

❑ **RS232**: definido por la norma EIA-232D, surgió en los 1960s para establecer enlaces entre sistemas terminales (DTE: Data Terminal Equipment) y centrales (DCE: Data Communication Equipment) de comunicaciones, a distancias de hasta 15 metros y velocidades menores a 20kbps. Emplea señales bipolares NRZ-L ($>+3V$ para ON, y $<-3V$ para OFF). Además de las básicas TXD y RXD la norma define otras señales de control de las que habitualmente se usa un subset (DTR, DCR, RTS, CTS, RI, DCD).

Interfases “single-ended” de velocidad media

Típicamente usadas dentro de sistemas y PCBs, y por ello de corto alcance. Su velocidad intermedia hace que el importante largo eléctrico de sus eventos haga posible (a veces) ignorar el efecto de línea de transmisión y obviar el uso de resistencias de terminación.

- ❑ **TTL (transistor-transistor logic):** desarrollada a inicios de los 70s en base a transistores NPN, opera a 5V, con salidas PushPull asimétricas e impedancias de entrada no despreciables. Hoy sólo se usa la variante LS-TTL (Low Power Schottky)
- ❑ **LVTTTL (low-voltage TTL):** variante de TTL para operar en 3.3V (definida en JEDEC JESD 8-A), con niveles de salida compatibles con TTL de 5V.
- ❑ **CMOS (complementary metal-oxide semiconductor) :** una primer familia (CD4000) con $V_{CC}=3:18V$ que evolucionó a HCMOS (High Speed CMOS). Con salida PushPull simétrica y muy elevada impedancia de entrada.
- ❑ **LVC MOS (low-voltage CMOS):** CMOS para $V_{CC}=3.3V$ (JEDEC JESD 8-A).
- ❑ **2.5-V I/O:** equivalente a LVC MOS para $V_{CC}=2.5V$ (JEDEC JESD 8-5).
- ❑ **1.8-V I/O:** equivalente a LVC MOS para $V_{CC}=1.8V$ (JEDEC JESD 8-7).
- ❑ **PECL (Positive Emitter Coupled Logic):** usa BJT NPN en zona activa, lo que mejora la velocidad de conmutación con menores excursiones de voltaje.

Interfases “single-ended” de alta velocidad

Estas interfases están pensadas para operar con las tecnologías modernas y a velocidad por arriba de 100MHz, y por ello surgen los siguientes cambios:

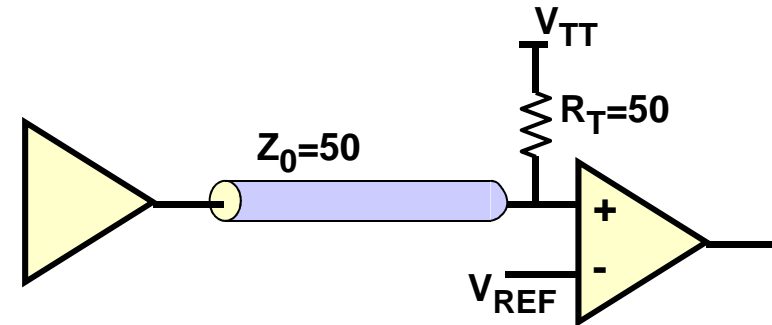
- ☐ Voltajes de operación de 3.3V, 2.5V, 1.8V, 1.5V y 1.2V
- ☐ Menor amplitud de excursión entre los valores lógicos del “1” y del “0”.
- ☐ Consideración del efecto de línea de transmisión y por ello uso de resistencias de terminación.
- ☐ Especialización en función del tipo de uso, ya sea buses internos, bancos de memoria o enlaces de datos de altísima velocidad

Entre **algunas** de las múltiples interfases existentes hoy tenemos:

- ☐ **CTT (center tap terminated)**
- ☐ **SSTL-2 Class I/II (Stub-Series Terminated Logic)**
- ☐ **SSTL-3 Class I & II (Stub-Series Terminated Logic)**
- ☐ **HSTL Class I/II (HSTL: High Speed Transceiver Logic)**
- ☐ **GTL+ (Gunning Transceiver Logic)**
- ☐ **AGP (Advanced Graphics Port)**

Interfases “single-ended” de alta velocidad - caso 1

- ❑ En este caso un buffer excita una línea de impedancia característica $Z_0=50$ ohms.
- ❑ En el destino se coloca una resistencia de terminación R_T que junto a la del receptor presentan un valor Z_0 . El voltaje al que se conecta R_T depende del tipo de salida (si es open collector o psuh-pull)
- ❑ La señal es comparada con una referencia V_{REF} para definir su valor lógico
- ❑ Entre las interfases de este tipo tenemos:
 - ❑ **HSTL Class I (High Speed Transceiver Logic)**
 - ❑ **CTT (Center Tap Terminated):** que es un superset de LVTTTL y LVCMOS



Nombre	Voltaje de operación	V_{TT}	V_{REF}	Data Rate (Mbps)
HSTL Class I	1.5	0.75	0.75	150-250
CTT	3.3	1.5	1.5	

HSTL y CTT están especificados en los estándares JEDEC JESD 8-6 y 8-4, respectivamente

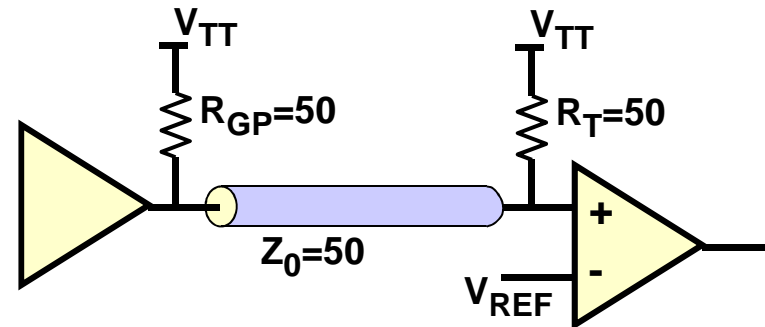
Interfases “single-ended” de alta velocidad - caso 2

❑ A diferencia del caso 1, acá se coloca también una resistencia de terminación en el generador para minimizar las reflexiones

❑ Entre las interfases de este tipo tenemos:

❑ **HSTL Class II (High Speed Transceiver Logic)**

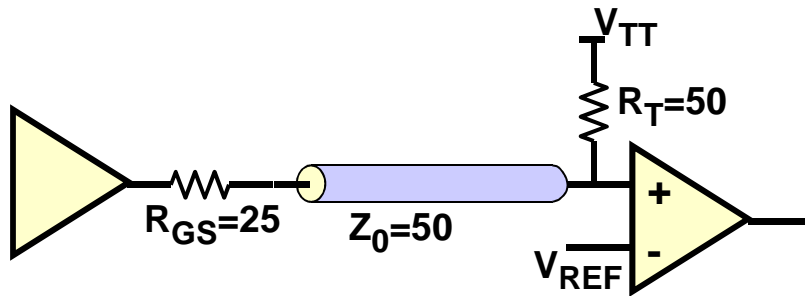
❑ **GTL y GTL+ (Gunning Transceiver Logic)**, con salida tipo open-collector usado por Intel para los Pentium Pro y en algunos backplanes



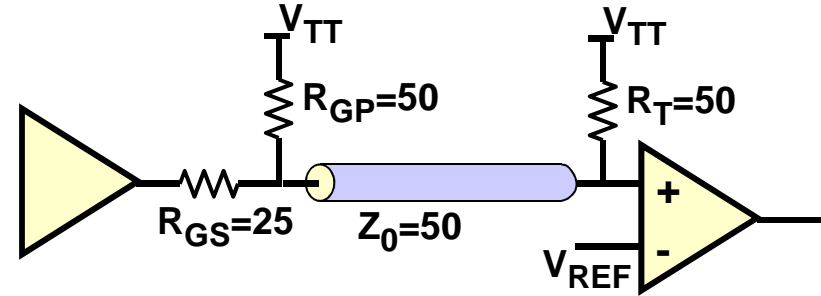
Nombre	Voltaje de operación	V_{TT}	V_{REF}	Data Rate (Mbps)
GTL	3.0	1.2	0.8	100
GTL+	3.0	1.5	1.0	133-200
HSTL Class II	1.5	0.75	0.75	150-250

HSTL está especificado en el estándar JEDEC JESD 8-6 respectivamente

Interfases “single-ended” de alta velocidad - casos 3 y 4



CLASS I



CLASS II

- ❑ Acá se coloca una resistencia en serie con el generador para minimizar las reflexiones y en ciertos casos una en paralelo

en muchas FPGAs modernas las resistencias están en el chip y se conectan al elegir la norma de operación

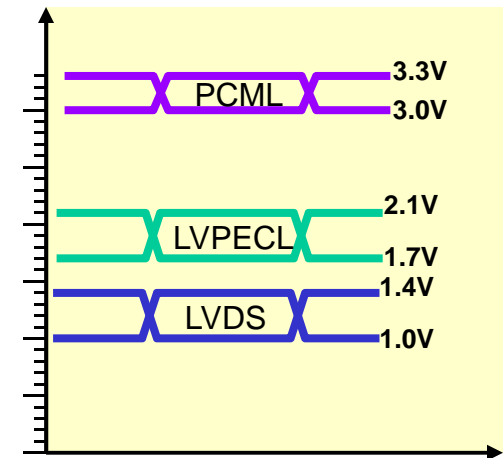


- ❑ Entre las interfases de este tipo tenemos las **SSTL (Serie Stub Terminated Logic)** y son usadas en interfases a memoria

Nombre	Voltaje de operación	$V_{TT} = V_{REF}$	JEDEC
SSTL-3 Class I/II	3.3	1.5	JESD 8-8
SSTL-2 Class I/II	2.5	1.25	JESD 8-9A
SSTL-18 Class I/II	1.8	0.9	JC42.3

Interfases diferenciales estándar

- Para enlaces de velocidades bajas y medias encontramos:
 - **RS485**: puede interconectar hasta 32 nodos en forma bidireccional
 - **RS422**: similar a RS485, solo que para enlaces punto a punto
- En el caso de interfases de muy alta velocidad tenemos
 - **LVDS** (Low Voltage Differential Signaling)
 - **LVPECL** (Low voltage Positive Emitter Coupled Logic, similar a LVDS)
 - **CML** (Current-Mode logic)
 - **Differential SSTL-2 Class I/II, Differential SSTL-3 Class I/II, Differential SSTL-18 Class I/II**
 - **Differential HSTL**

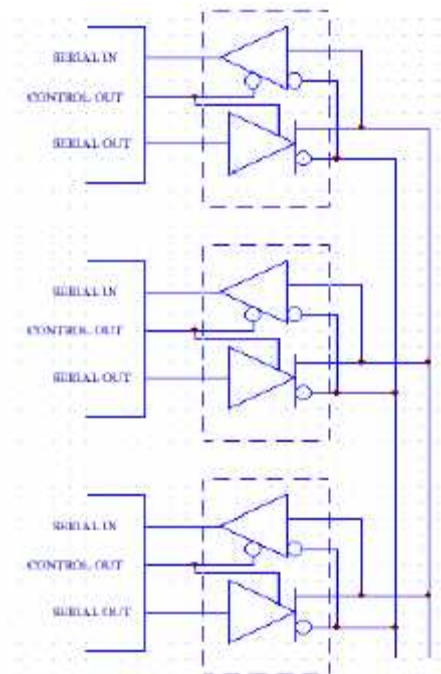


Interfases diferenciales de baja velocidad: RS485

Interfaz diferencial, con alimentación única de +5V, acoplamiento directo

- Permite la conexión multipunto de hasta 32 estaciones
- Velocidad máxima de 10 Mbit/s (a 12 metros), con una longitud máxima de alcance de 1200 metros (a 100 kbit/s)
- Adaptación de 120 ohms en cada extremo del bus
- Sensibilidad diferencial mínima de 200mV
- Existen transceivers aislados ópticamente para mejorar la susceptibilidad de la conexión (P.Ej: ADM2587)

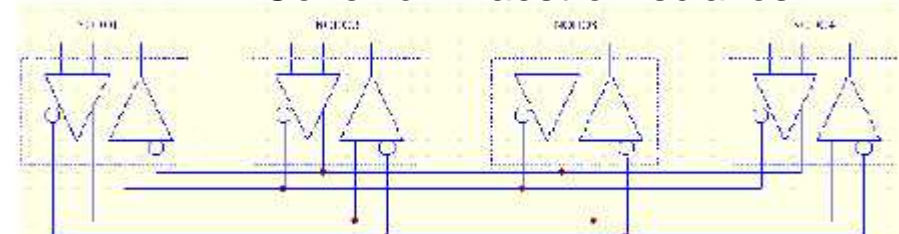
Conexión Bus



Conexión Full-Duplex

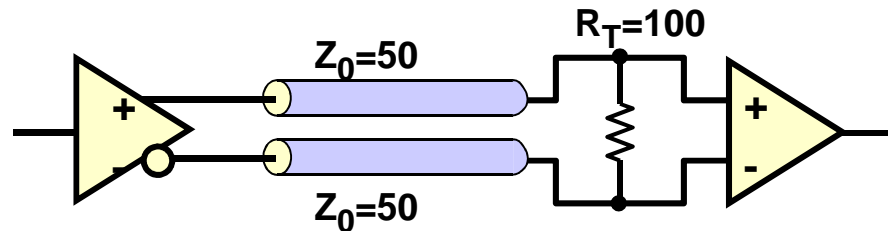


Conexión Maestro-Eslavos



<http://www.i-micro.com/pdf/articulos/rs-485.pdf>

Interfaces diferenciales de alta velocidad: **LVDS**

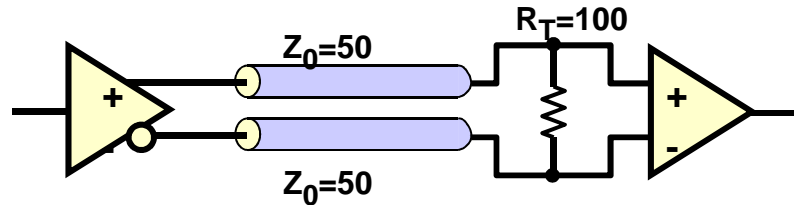


LVDS (Low Voltage Differential Standard) es muy usada para transferir datos a muy alta velocidad con bajo consumo de potencia. Está definida mediante las normas IEEE 1596.3 SCI-LVDS para velocidades de hasta 250 Mbps y ANSI/TIA/EIA-644 para hasta 655 Mbps

- ❑ La norma LVDS requiere una alimentación de 3.3-V y una resistencia de terminación de 100 ohms a la entrada del receptor (para las señales diferenciales, esta resistencia se comporta como dos resistencias de 50 ohms polarizadas al voltaje de modo común)
- ❑ El voltaje diferencial de salida LVDS es de mas de 350mV, con un valor de modo común de 1.25V.
- ❑ En una entrada LVDS la amplitud diferencial debe superar los 100mV, y la tensión de modo común puede valer entre 0.25V y 2.25V.
- ❑ Las líneas de transmisión deben ser de $Z_0=50$ ohms, y es deseable que corran próximas entre sí para homogeneizar el crosstalk de modo común, y que sean de igual longitud para minimizar el skew en el tiempo de arribo

Multi-Drop LVDS with Virtex-E FPGAs, XILINX Application Note XAPP231

Interfaces diferenciales de alta velocidad: variantes de LVDS



Mientras LVDS especifica un voltaje de salida de 350mV con un Vcom de 1.25V y sobre un par balanceado de 50 ohms con terminación de 100 ohms, la norma *RSDS* (por *Reduced Swing Differential Signaling*) define una amplitud de salida de 200 mV (400 mV_{pap}) con igual voltaje de modo común, y la norma *SLVS-400* (*scalable-low-voltage-signaling*) la baja a 200mV (400 mV_{pap}) con un modo común de 200mV, donde la corriente del transmitter y la impedancia tiene mayores holguras.

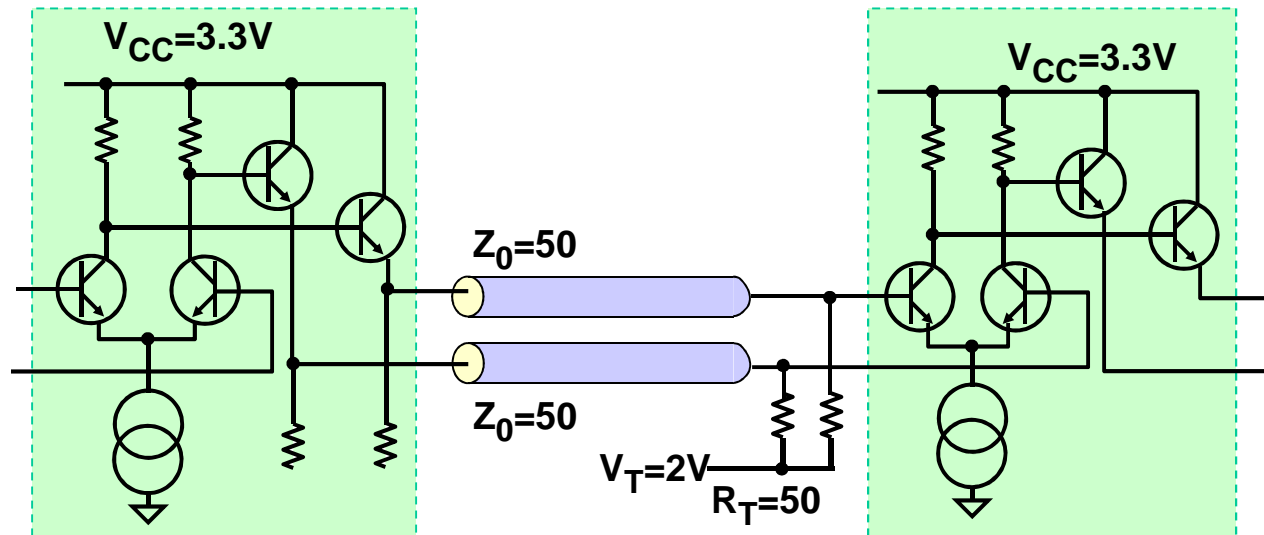
Para evaluar las diferencias, un enlace LVDS de 6-Gbps consume aproximadamente 250 mW, mientras que uno SLVS configurado para 800 Mbps consume cerca de 15mW. Así, 8 enlaces SLVS a 800Mbps proveen 6,4Gbps de transferencia de datos con sólo 120mW de consumo, la mitad del caso LVDS.

- http://www.edn.com/article/518242-Implementing_an_SLVS_transceiver.php
- “Scalable Low-Voltage Signaling (SLVS) standard for 400 mV, JESD8- 13”, JEDEC Solid State Technology Association, October 2001.

Interfases de alta velocidad: Differential SSTL/HSTL

- ❑ La interfase “**Differential SSTL-2 Class I/II**” (sin/con resistencia *stub* en el transceiver) corresponde al uso diferencial de **SSTL-2 Class I/II** en aplicaciones de alta velocidad como es la interfase a memorias de doble velocidad de acceso (*DDR: Double Data Rate*). Tal como **SSTL-2** single ended, opera en el rango de los 2,5V, con idénticos valores de impedancias, y voltajes. Para operar en 3.3V y en 1,8V existen las variaciones **Differential SSTL-3 Class I/II** y **Differential SSTL-18 Class I/II**.
- ❑ La interfase “**Differential HSTL I/O**” se usa en aplicaciones que operan con tensiones de 0 a 1,5V con elevadísimas velocidades, tal como sucede en el manejo de memorias de régimen de acceso cuádruple (*QDR: quad data rate*). Sus especificaciones son como las de HSTL single ended, con idénticos valores de impedancias y voltajes

Interfaces diferenciales de alta velocidad: **LVPECL**

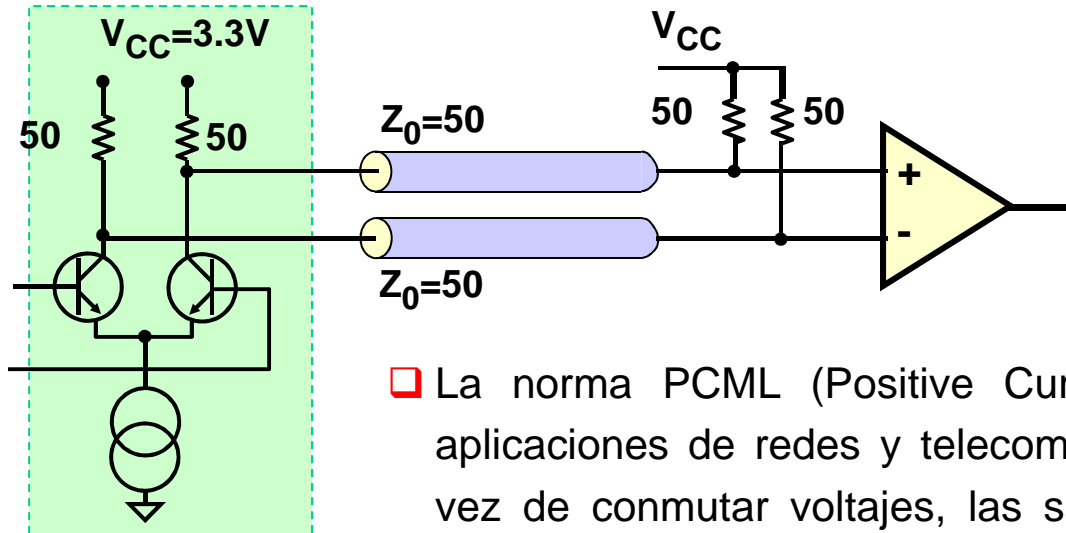


LVPECL es una interfase de alta velocidad con características propias de los BJT:

- ❑ requiere $V_{CC}=3,3V$ y resistencias de terminación de 50 ohms en las entradas del receptor, polarizadas a un punto común de 2V. Cada resistencia soporta un flujo constante de corriente de polarización de 14 mA
- ❑ El voltaje diferencial de salida es de más de 595mV, con un valor $V_{COM}=2V$.
- ❑ En una entrada la amplitud diferencial debe superar los 300mV

Usada en aplicaciones de gráficos, telecomunicaciones, comunicación de datos y distribución de clock.

Interfaces diferenciales de alta velocidad: PCML



- ☐ La norma PCML (Positive Current Mode Logic) es usada en aplicaciones de redes y telecomunicaciones. Opera en 3.3V y en vez de conmutar voltajes, las salidas PCML emplean un circuito diferencial usado como llave de corriente
- ☐ Las resistencias de terminación en las entradas están conectadas a VCC, con lo que la tensión VCOM es sólo 200 mV menor a VCC (mayor que LVDS y PECL) y el voltaje diferencial de 400mV.
- ☐ Tiene mejor performance y consume menos energía que LVPECL, porque sus excursiones más reducidas de voltaje posibilitan menor tiempo de conmutación y menos consumo de potencia.

Codificación de señales de banda base sin componente DC

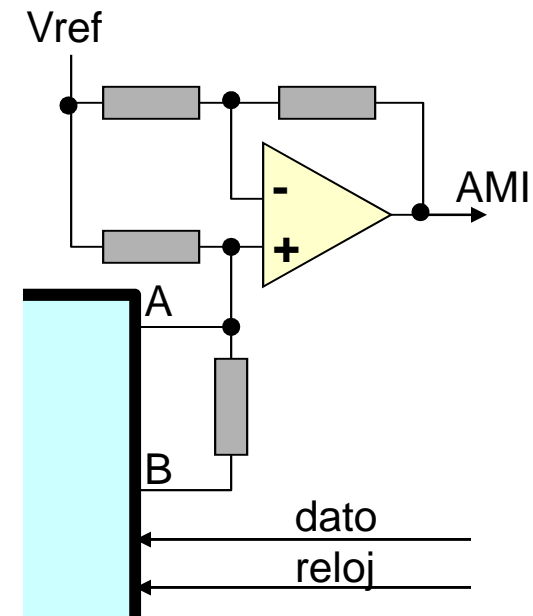
- AMI
- HDB3
- MANCHESTER
- 8B10B

Codificación de señales de banda base con y sin componente DC

- Al transmitir una señal de datos suele ser conveniente modificarla:
 - Agregarle información de reloj, de modo que ese reloj pueda luego ser recuperado por el receptor. De ese modo por un único vínculo se transmiten los datos y su temporización
 - Tratar que el valor medio de la información transmitida sea nulo, para poder usar métodos de acoplamiento magnético (transformadores) o capacitivos. Esta necesidad de evitar la componente DC puede ser conveniente cuando los sistemas están físicamente alejados, para evitar corrientes parásitas, e incluso como medida de seguridad, para bloquear la circulación de voltajes o corrientes destructivas para personas o cosas.
 - Y *—de ser posible—* agregar mecanismos para control y sincronismo
- Estas necesidades definen distintos métodos de codificación, aptos para distintas aplicaciones.

Codificación tri-nivel con componente DC nula: **AMI**

- ❑ **AMI (Alternate Mark Inversion)** es una forma de envío de datos y reloj por un único canal serie sincrónico usando pulsos bipolares
- ❑ Usa una señalización de tres niveles:
 - ❑ un pulso positivo (+) o negativo (-) para representar un "1"
 - ❑ la ausencia de pulso (0) para un representar un "0"
- ❑ Para mantener un valor medio nulo, sucesivos bits en "1" (o **Marks**) son transmitidos mediante pulsos de polaridad invertida en forma alternada (de ahí AMI)
- ❑ El patrón generado para transmitir la secuencia binaria **"1000011"** será entonces **"+ 0 0 0 0 - +"**; Una larga secuencia de "0"s genera el inconveniente de que la ausencia de transiciones no permite asegurar la sincronización de los relojes transmisor y receptor



A	B	AMI
OPEN	OPEN	+Vref
OPEN	GND	GND
GND	X	-Vref

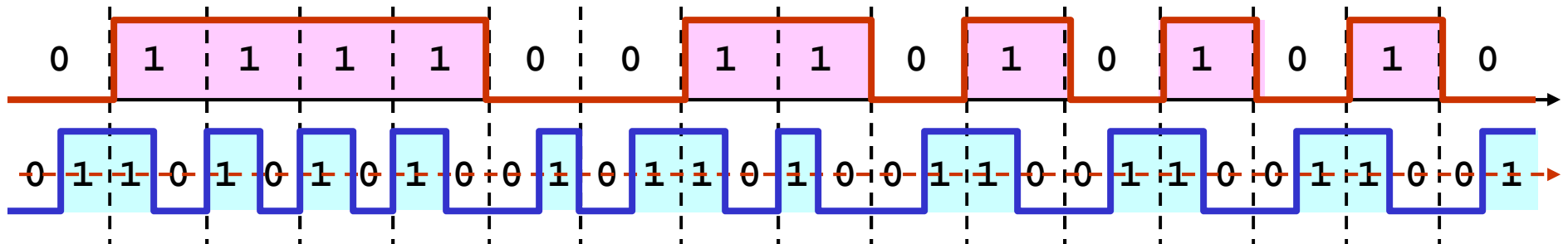
Diseñar el circuito en VHDL y simularlo
Usar ambos flancos del reloj para que los pulsos duren exactamente 1/2 ciclo

Codificación tri-nivel con componente DC nula: **HDB3**

- ❑ **HDB3 (High Density Binary level 3)** envía datos a través de un canal serie síncrono usando pulsos bipolares (+) o (-) o nulos (0) con reglas de codificación similares a AMI
- ❑ A partir de una señal codificada en **AMI**, en **HDB3** se agregan “violaciones de código” (V) y eventuales pulsos de balance (B) siempre que aparecen 4 ceros sucesivos en la entrada, los que son reemplazados por **000V** o **B00V** según corresponda. Esto garantiza la existencia de pulsos en largas secuencias de ceros, y facilita la recuperación de reloj
- ❑ La violación de código (**V**) se realiza reemplazando cada cuarto cero por un pulso no nulo de polaridad opuesta a la violación previa
- ❑ Y el eventual pulso **B**, de igual polaridad a **V**, se agrega si el pulso inmediato anterior a **B** es de polaridad opuesta a la de **V** (sinó, no se podría saber si V es una violación o un “1”).

1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1	<i>datos binarios</i>
+ - + 0 0 0 0 0 - + 0 0 0 0 0 0 0 0 0 -	<i>señal AMI</i>
+ - + 0 0 0 V 0 - + B 0 0 V B 0 0 V 0 -	<i>HDB3 con V y B</i>
+ - + 0 0 0 + 0 - + - 0 0 - + 0 0 + 0 -	<i>HDB3 final</i>
M M M 0 0 0 V 0 M M B 0 0 V B 0 0 V 0 M	<i>HDB3 usando M, B y V</i>

Codificación bi-nivel con valor medio fijo: **MANCHESTER**



La codificación **MANCHESTER** combina datos y reloj en un canal de datos según la siguiente regla:

- ❑ Un cero es transmitido asegurando un flanco positivo en el centro del tiempo de bit
- ❑ Un uno es transmitido asegurando un flanco negativo en el centro del tiempo de bit
- ❑ Según cuál sea el bit previo y el actual, al inicio del tiempo de bit puede haber otra transición

Como SIEMPRE hay una transición en el medio de cada bit, se logran las siguientes ventajas

- ❑ el valor medio de la señal transmitido es 0.5, no importa el valor de los datos
- ❑ hay una alta ocurrencia de transiciones, lo que hace muy fácil la recuperación del reloj
- ❑ Este método es usado en la norma **ETHERNET**, donde el acoplamiento es magnético

En casos donde puede perderse la polaridad se usa **MANCHESTER DIFERENCIAL** donde el dato se codifica en base a los bits MANCHESTER actual y previo: 0=iguales (00 o 11), 1=distintos (01 o 10)

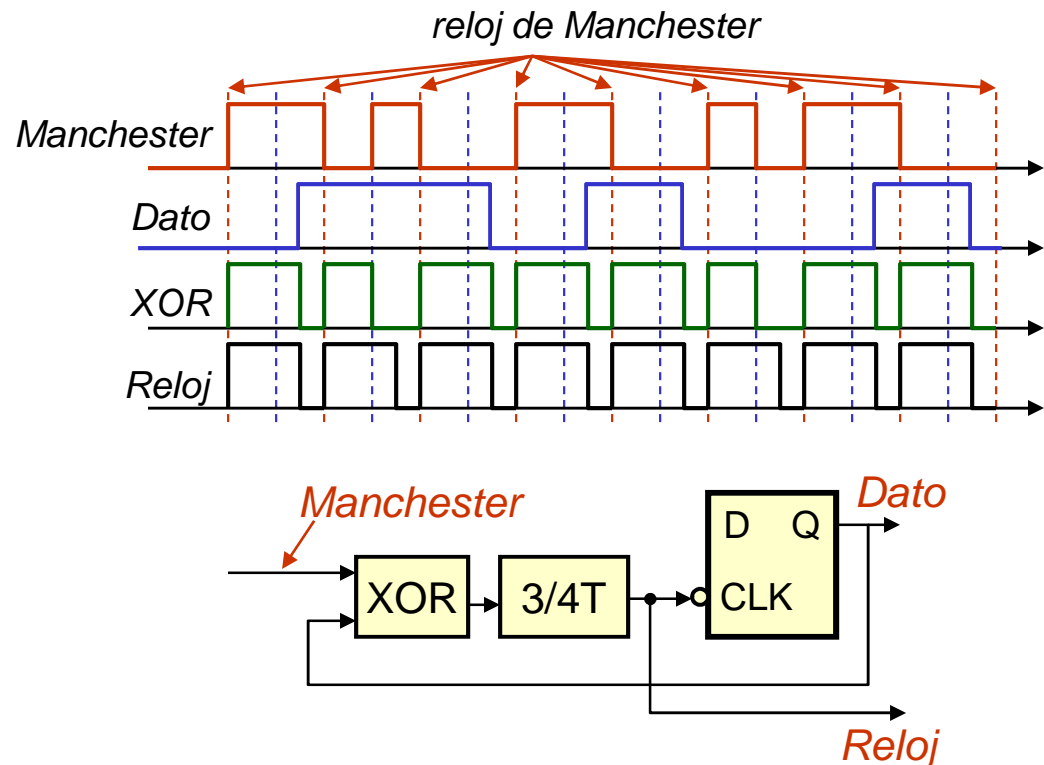
El espectro se corre hacia frecuencias altas, por lo que no conviene para tasas de bits muy elevadas

Decodificación MANCHESTER

La decodificación **MANCHESTER** permite recuperar los datos y el reloj, y de usarse Manchester diferencial esta decodificación es tolerante a la inversión de la señal, aunque la tasa de errores de la decodificación MANCHESTER diferencial es el doble que la del MANCHESTER estándar

La idea es:

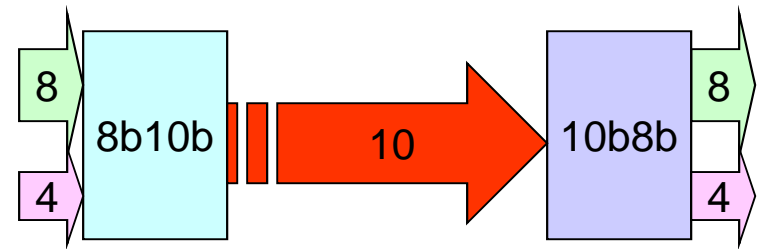
- Si el dato es 1 corresponde a un flanco – Manchester
- Si el dato es 0 corresponde a un flanco + Manchester
- El XOR del manchester y el dato tiene un flanco + en el momento del clock
- Si a partir del flanco + del clock se copia el valor Manchester luego de $\frac{3}{4}$ Tbit se tiene el dato decodificado



Codificación y decodificación con valor medio fijo **8b10b**

Para mantener un valor medio fijo y no elevar demasiado el espectro de frecuencias surge **8B10B**, que codifica datos de 8 bits en palabras de 10bits mediante el agregado de bits redundantes de modo de:

- Mejorar las características de transmisión
- Mejorar la detección de errores
- Discriminar entre datos y símbolos de control
- Facilitar la recuperación de reloj y los procesos de sincronización de bits y de palabras



Es un método estandarizado según los siguientes documentos:

- Institute of Electrical and Electronics Engineers, IEEE 802.3z, Media Access Control (MAC) Parameters, Physical Layer, Repeater and Management Parameters for 1000 Mb/s Operation, 1998, paragraphs 36.2.4.1 to 36.2.4.6
- American National Standards Institute, ANSI X3.230, Fibre Channel Physical and Signaling Interface (FC-PH), 1994
- International Telecommunication Union, ITU-T Recommendation G.7041, Generic Framing Procedure, October 2001

Ventajas de la codificación y decodificación 8b10b

8b10b es usado por Fiberchannel, Gigabit Ethernet, 10Gigabit Ethernet, Infiniband, y PCI-Express, y básicamente funciona del siguiente modo:

- ❑ Con los 1024 posibles códigos de 10 bits se forman dos conjuntos de 256 códigos para los datos y un conjunto de 12 códigos para control (*K-characters*). Los restantes son códigos inválidos
- ❑ Cada uno de los 256 posibles datos a transmitir es representado por dos posibles códigos de 10 bits, uno donde la relación de unos y ceros es 6/4 o 5/5, y otro donde vale 4/6 o 5/5. A la hora de transmitir cada dato se opta por aquel código que permite minimizar el valor medio de la señal
- ❑ Estos códigos presentan a su vez no más de 5 ceros o 5 unos consecutivos (y ese caso son llamados *comma character*), facilitando la recuperación del reloj y de trama

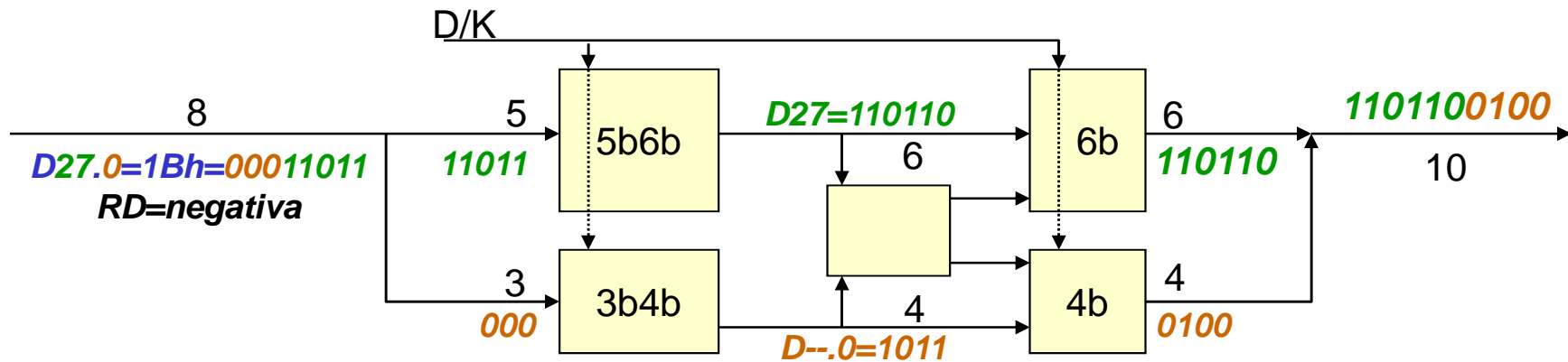
Los dos posibles códigos para cada dato permiten mantener un desbalance de valor medio nulo de unos y ceros en enlaces acoplados en alterna (ese desbalance se llama *Running Disparity* y es el transmisor quien trata de mantenerlo alrededor de cero).

Este método de codificación permite detectar una amplia cantidad de errores:

- ❑ patrones con relación unos/ceros distinta a 6/4, 5/5, o 4/6
- ❑ tramas de más de 5 unos o ceros consecutivos
- ❑ facilita la detección de errores simples, dobles y triples mediante CRC

Codificación 8b10b

Un encoder 8b10b puede realizarse con tablas LUT descomponiendo los 8 bits del dato en un campo de 5 bit (que crecen a 6 al codificarse, tabla 5b6b, con no más de 4 unos o 4 ceros) y uno de 3 bit (que pasan a 4 al codificarse, tabla 3b4b, con no más de 3 unos o 3 ceros).



La nomenclatura usada para representar un valor de entrada es **DXX.Y** para datos y **KXX.Y** para caracteres de control, donde **XX** representa los bits 4:0 y **Y** los bits 7:5. Por ejemplo:

- ❑ D00.0 significa el dato 00h, codificado como 100111_0100 (5/5) o 011000_1011 (5/5)
- ❑ D01.0 significa el dato 01h, codificado como 011101_0100 (5/5) o 100010_1011 (5/5)
- ❑ D00.1 significa el dato 20h, codificado como 100111_1001 (6/4) o 011000_1001 (4/6)
- ❑ D10.2 el dato 4Ah, es codificado como 010101_0101 (5/5)
- ❑ D21.5 el dato B5h, es codificado como 101010_1010 (5/5)

Estos dos ultimos casos son los caracteres con mayor número de transiciones!

Codificación 8b10b

Encoder 5 a 6 y 3 a 4

Dato	Bits	RD-	RD+	Dato	Bits	RD-	RD+
D0	00000	100111	011000	D16	10000	011011	100100
D1	00001	011101	100010	D17	10001	100011	011100
D2	00010	101101	010010	D18	10010	010011	101100
D3	00011	110001	001110	D19	10011	110010	001101
D4	00100	110101	001010	D20	10100	001011	110100
D5	00101	101001	010110	D21	10101	101010	010101
D6	00110	011001	100110	D22	10110	011010	100101
D7	00111	111000	000111	D23	10111	111010	000101
D8	01000	111001	000110	D24	11000	110011	001100
D9	01001	100101	011010	D25	11001	100110	011001
D10	01010	010101	101010	D26	11010	010110	101001
D11	01011	110100	001011	D27	11011	110110	001001
D12	01100	001101	110010	D28	11100	001110	110001
D13	01101	101100	010011	D29	11101	101110	010001
D14	01110	011100	100011	D30	11110	011110	100001
D15	01111	010111	101000	D31	11111	101011	010100

Dato	Bits	RD-	RD+
D--.0	000	1011	0100
D--.1	001	1001	0110
D--.2	010	0101	1010
D--.3	011	1100	0011
D--.4	100	1101	0010
D--.5	101	1010	0101
D--.6	110	0110	1001
D--.7	111	1110/0111	0001/1000

*De venir con disparidad negativa, para codificar D17.7=11110001 (F1h) si se usa **100011_1110** se crearía un comma character. En ese caso se usa 100011_0111. Esto pasa porque es el único de la tabla que tiene 3 ceros o 3 unos seguidos*

Codificación 8b10b

Ejercicios

A transmitir RD entrada +	0x3C	0x0F	0xF0	0x5A	0xA5	0x88	0x6F	0xFA	0x4D	0xF1
Dato previo codificado										
Formato 8b10b										
Campo 5b6b										
RD intermedia										
Campo 3b4b										
A transmitir										
RD Saliente										

Codificación 8b10b los K-Characters

En cuanto a los caracteres de control, ellos son:

K-ch	Bits	RD-	RD+
K23	10111	111010	000101
K27	11011	110110	001001
K28	11100	001111	110000
K29	11101	101110	010001
K30	11110	011110	100001

Dato	Bits	RD-	RD+
K--.0	00	1011	0100
K--.1	001	0110	1001
K--.2	010	1010	0101
K--.3	011	1100	0011
K--.4	100	1101	0010
K--.5	101	0101	1010
K--.6	110	1001	0110
K--.7	111	0111	1000

- ❑ **K.28.0:** 38h , se codifica 001111_0100 (5/5), es SKP en PCI-Express
- ❑ **K.28.1:** 3Ch, se codifica **001111_1**001 (6/4) *comma pattern* , es FTS en PCI-Express
- ❑ **K.28.2:** 3Ah, se codifica 001111_0101 (6/4), es SDP en PCI-Express
- ❑ **K.28.3:** 3Eh, se codifica 001111_0011 (6/4), es IDL en PCI-Express
- ❑ **K.28.4:** 39h, se codifica 001111_0010 (5/5)
- ❑ **K.28.5:** 3Dh, se codifica **001111_1**010 (6/4) *comma pattern* (COM en PCI-Express)
- ❑ **K.28.6:** 3Bh, se codifica 001111_0110 (6/4)
- ❑ **K.28.7:** 3Fh, se codifica **001111_1**000 (5/5) *comma pattern*
- ❑ **K.23.7:** EFh, se codifica 111010_1000 (5/5). Es PAD en PCI-Express
- ❑ **K.27.7:** DFh, se codifica 110110_1000 (5/5), es STP en PCI-Express
- ❑ **K.29.7:** BFh, se codifica 101110_1000 (5/5), es END en PCI-Express
- ❑ **K.30.7:** 7Fh, se codifica 011110_1000 (5/5), es EDB en PCI-Express