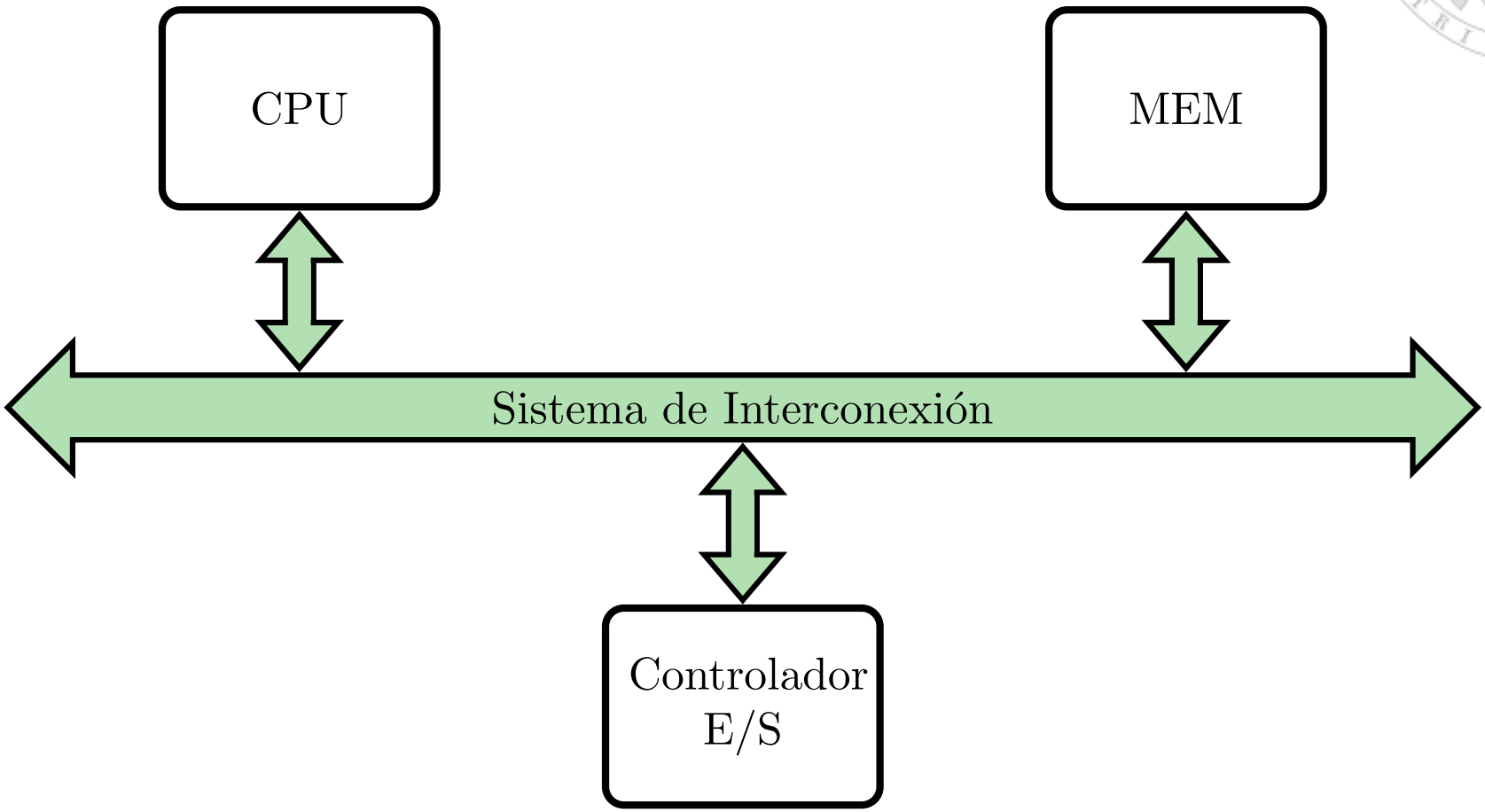
Estructura de Computadores

Curso
2017-2018

Sistema de Entrada/Salida
Interconexión (Buses)

Sistema de Interconexión

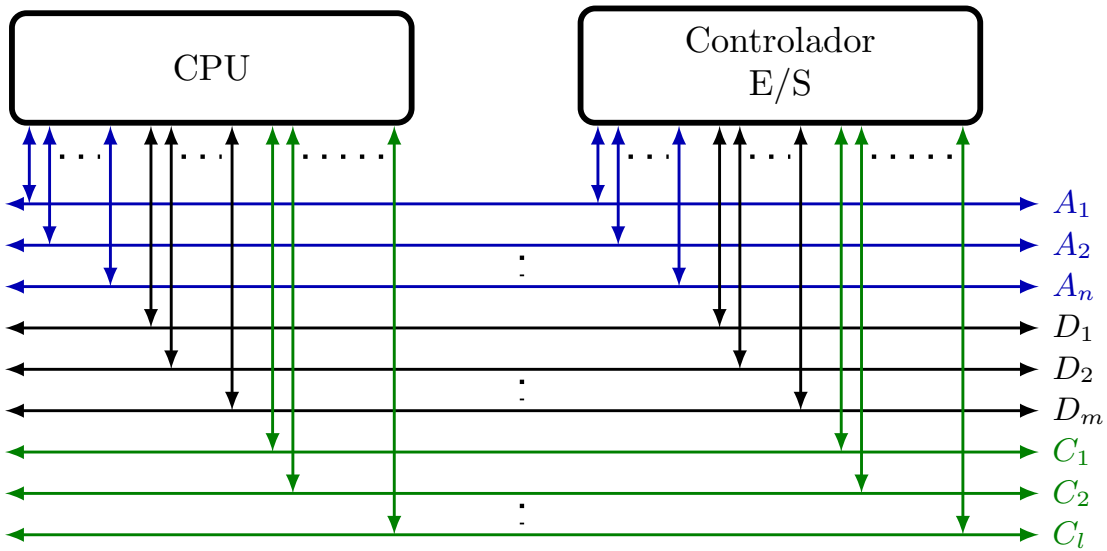


Agenda

Agenda

Comunicación tradicional en el computador

- Transmisión paralela
  - Ancho de banda elevado (en teoría...)
  - Constoso a grandes distancias
  - Frecuencia limitada por factores físicos
- Líneas compartidas, es necesario el arbitraje
  - **Dirección**: transmiten las direcciones
  - **Datos**: transmiten los datos
  - **Control**: gestión de la transferencia y arbitraje
    - En versiones síncronas una línea es el reloj



Operaciones de transferencia

- Dos roles:
  - Master: Dispositivo capaz de tomar el control del bus
  - Slave: resto
- Dos tipos/sentidos de transferencia
  - Escritura: Master $\rightarrow$ Slave
  - Lectura: Slave $\rightarrow$ Master
- Fases de una operación de bus
  1. Arbitraje (master toma control del bus)
  2. Direccionamiento del slave
  3. Especificación del tipo de operación (lectura/escritura)
  4. Transferencia del dato
  5. Finalización del ciclo de bus
- Sincronización
  - Síncrono
  - Semi-síncrono
  - Asíncrono

Arbitraje

Mecanismo para resolver los conflictos en el acceso al bus con másters

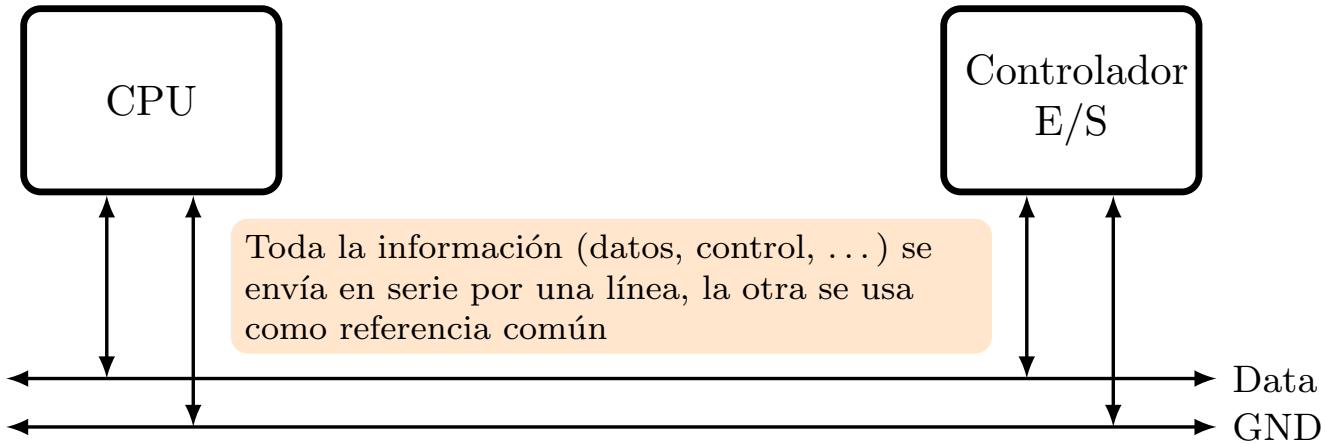
- Evita que dos módulos escriban simultáneamente en el bus
- Cada Master solicita permiso para poder tomar el control del bus

Existen dos clases de protocolos

- Centralizados
  - Existe un árbitro o máster principal que controla el acceso al bus
  - Ejemplo: En estrella, daisy chain de 2,3 y 4 hilos
- Distribuidos
  - Control repartido entre todos los másters conectados al bus
  - Ejemplo: códigos de identificación con prioridades

Agenda

Transmisión Serie



- Una línea principal de datos/control
- Líneas auxiliares (clk, gnd,. . .)
- Menos costosa que la E/S paralela
- Frecuencias mayores (sin *clock skew*)
  - A igualdad de frecuencia el ancho de banda es menor
  - Aumentar el ancho de banda $\rightarrow$ varias conexiones serie

Principales ámbitos de uso

- Redes de computadores
  - Ethernet
- Interfaces/buses para periféricos externos
  - DVI, HDMI, USB, ...
- Interfaces/buses para periféricos internos
  - SATA, PCI Express
- Interfaces/buses entre chips
  - SPI, I^2C, 1-Wire, CAN Bus, ...

Tipos de comunicación serie

- Asíncrona
  - Emisor y receptor no comparten la señal de reloj
    - Frecuencias no completamente idénticas
    - Relojes desfasados
  - Problemas de sincronización bit/carácter
  - Transmisión orientada a carácter
  - Ejemplo: UART (RS-232, ...)
- Síncrona
  - Emisor y receptor comparten señal de reloj
  - Transmisión orientada a bloque
  - Problema de sincronización byte/bloque
  - Ejemplos: I^2C, SPI, 1-Wire, ...

Agenda

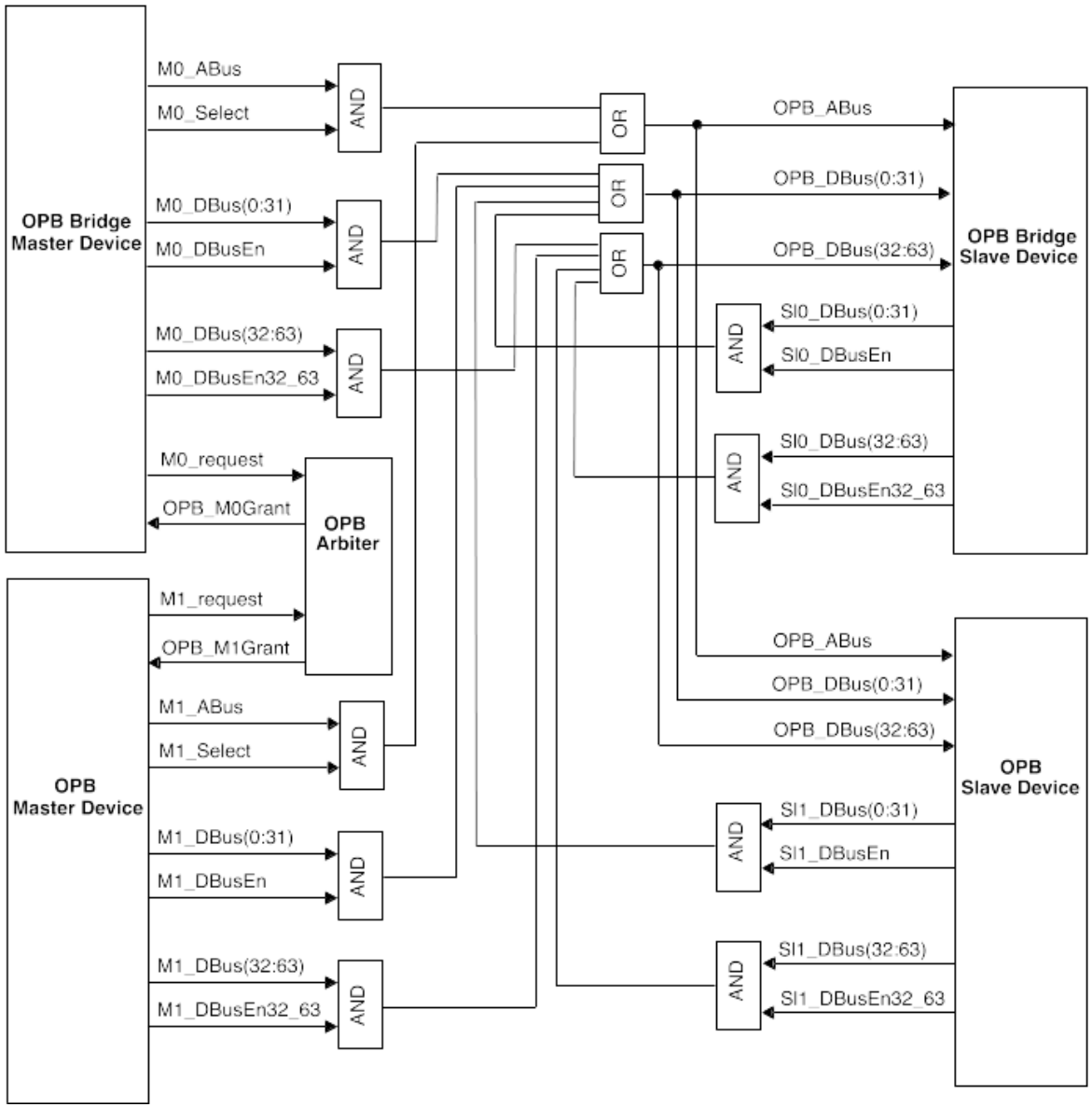
Open Peripheral Bus (OPB)

- IBM
  - `https://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/9A7AFA74DAD200D087256AB30005F0C8`
- Para conexión de dispositivos lentos en chip
- Versiones 32 y 64 bits
- Síncrono
- Big-endian:
  - Byte más significativo -¿ menor dirección
  - Bit0 más significativo, Bit31 menos significativo

OPB: nomenclatura

- Señales del bus hacia master o slave
  - `OPB_*`
- Señales de master n hacia árbitro o bus:
  - `Mn_*`
- Señales de slave n hacia bus:
  - `Sln_*`

Conexiones AND-OR: mux distribuido



Arbitraje

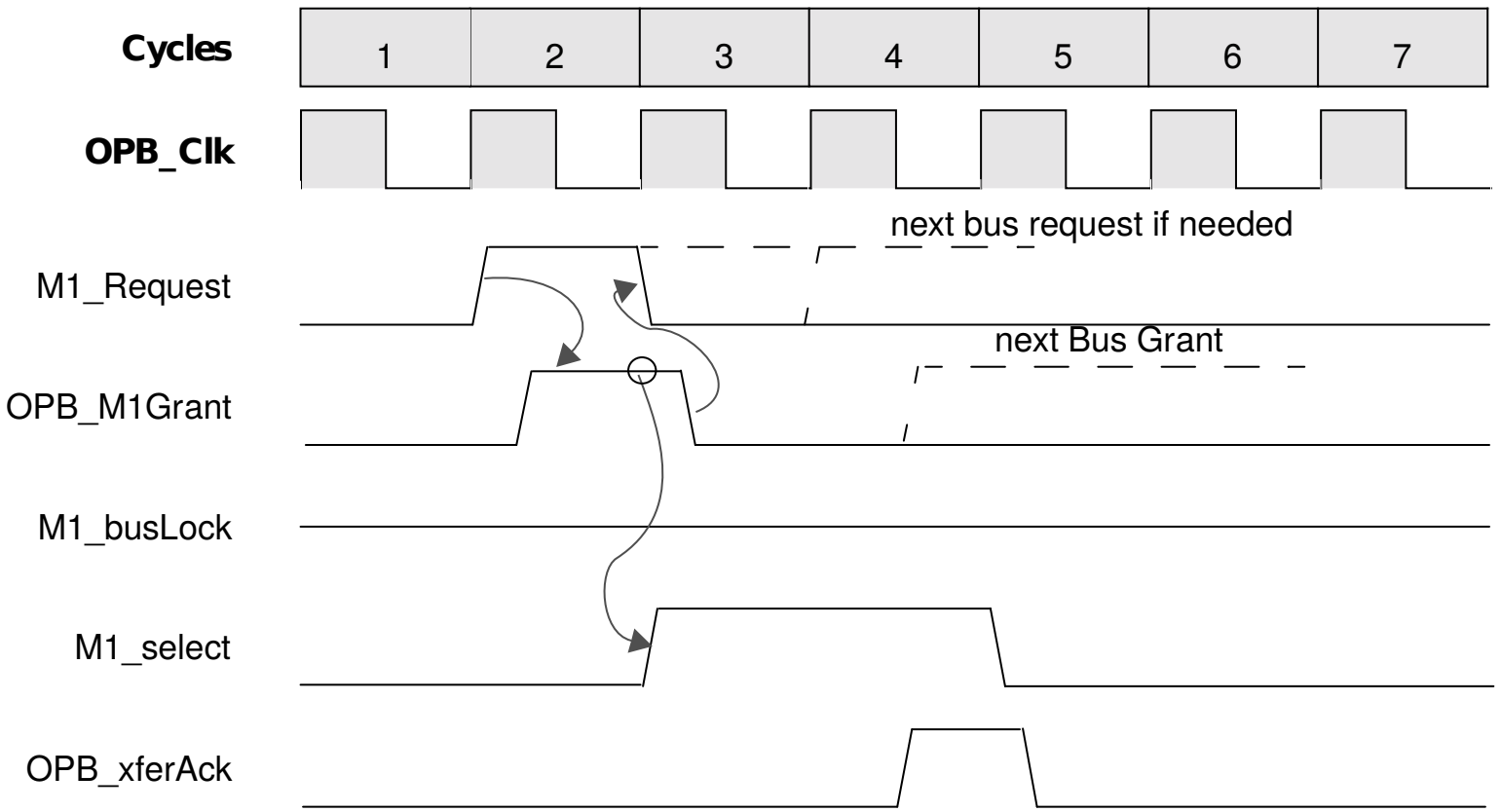
Arbitro centralizado

- Señal de request para master n: `Mn_req`
- Señal de grant a master n: `OPB_Mngrant`
- La señal se muestrea con el flanco de subida del clk
- Master inicia operación activando `Mn_select`

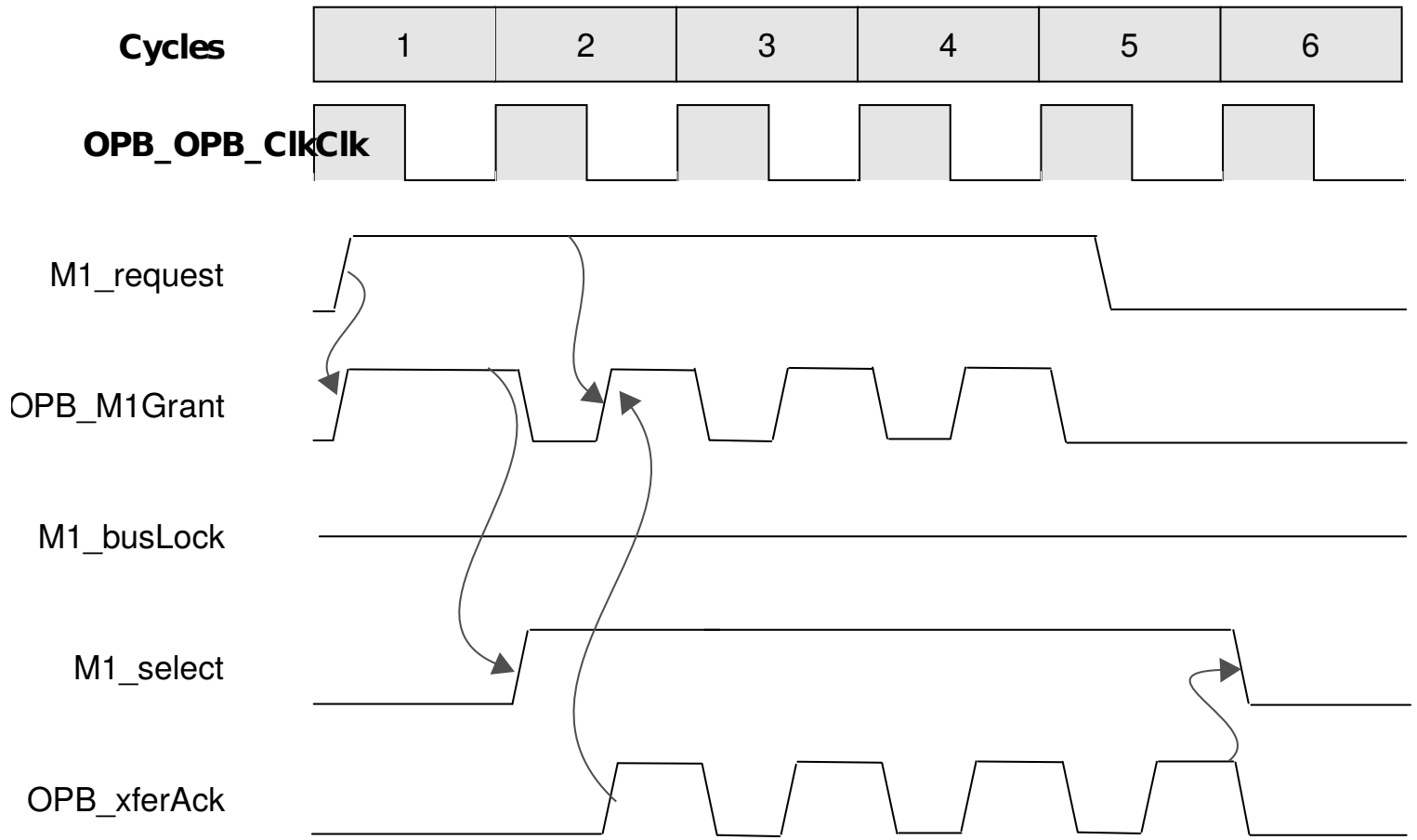
Ciclo de arbitraje válido:

- IDLE:
  - (`OPB_select` $==$ 0) y (`OPB_buslock` $==$ 0)
- Overlapped:
  - (`OPB_xferACK` $==$ 1) y (`OPB_buslock` $==$ 0)

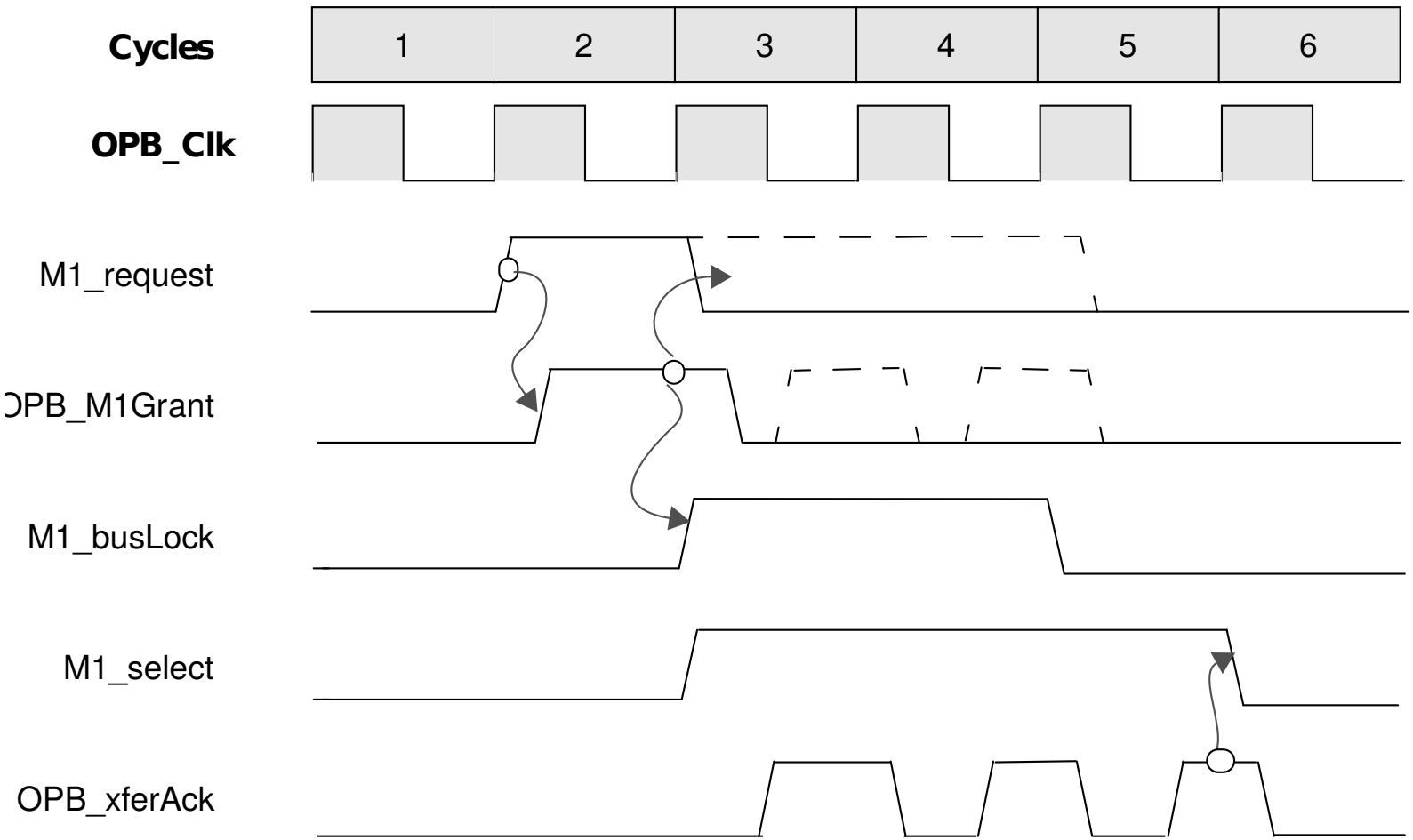
Ciclo de arbitraje básico



Petición continua de bus



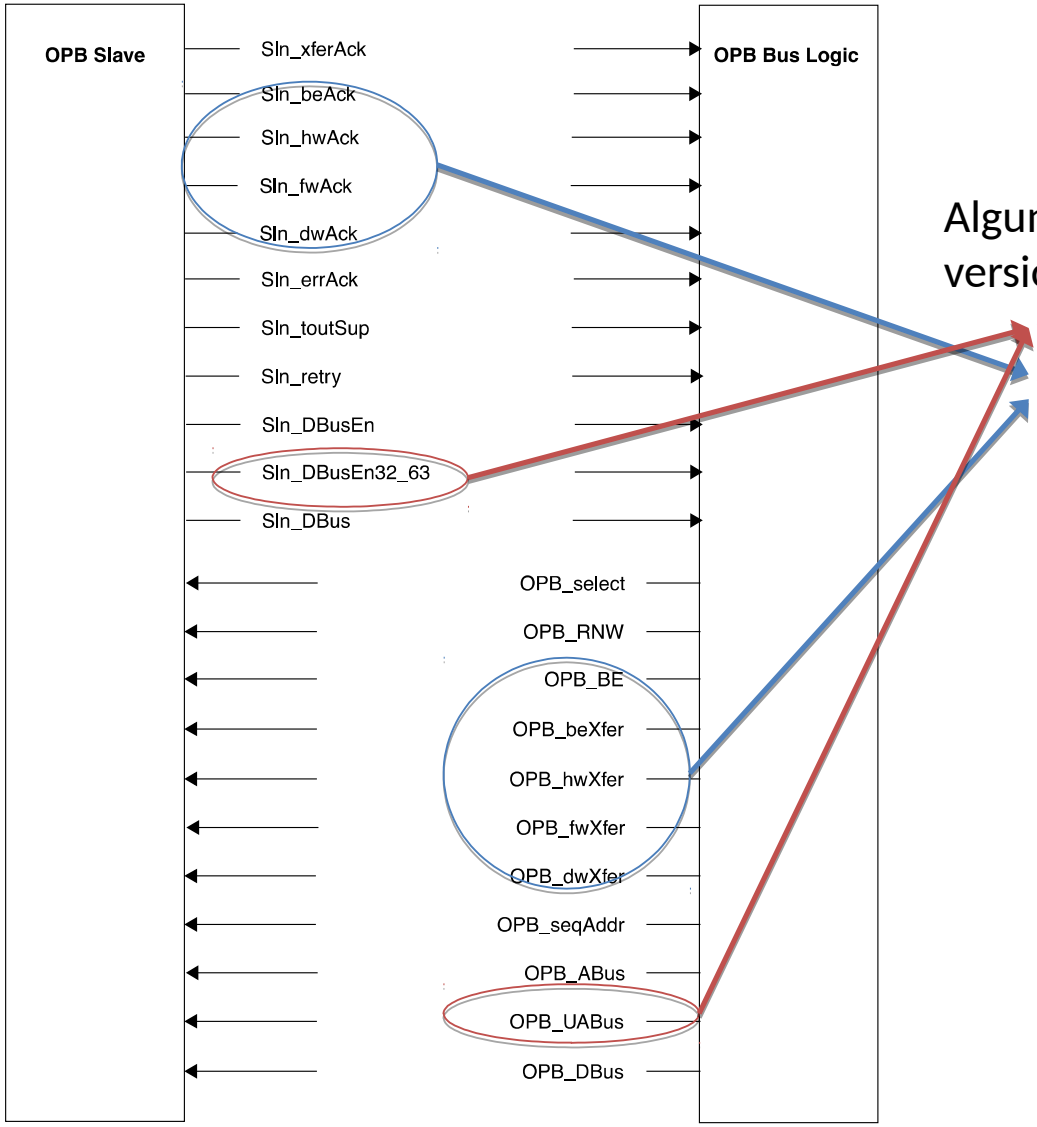
Bloqueo de bus



Dispositivo esclavo

- No toma el bus
  - No entra en ciclo de arbitraje
  - No puede iniciar transferencias
- Pendiente del bus *por si algún master inicia una operación con él*
  - `OPB_select` $==$ 1
  - `OPB_Abus` en el rango del dispositvo

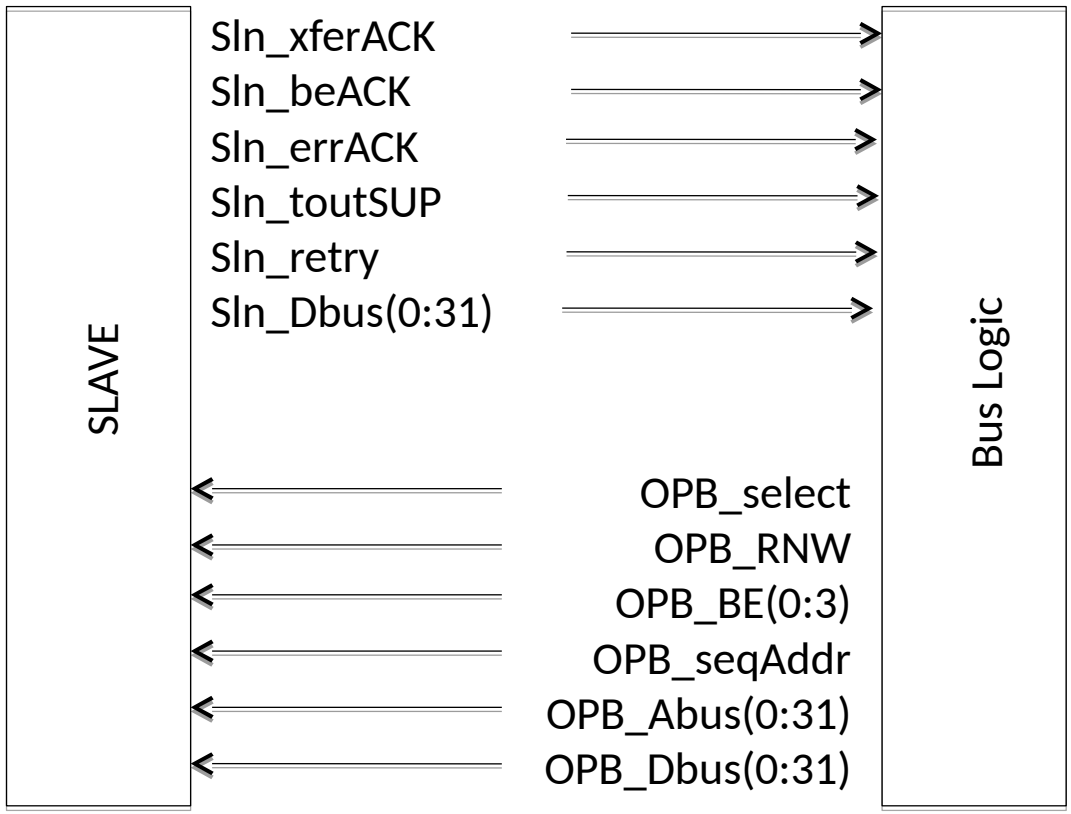
Interface esclavo



Algunas conexiones dependen de versión:

- 32 vs 64 bits
- Dynamic bus sizing y/o byte enable arch.

Interface esclavo 32 bits, byte enable



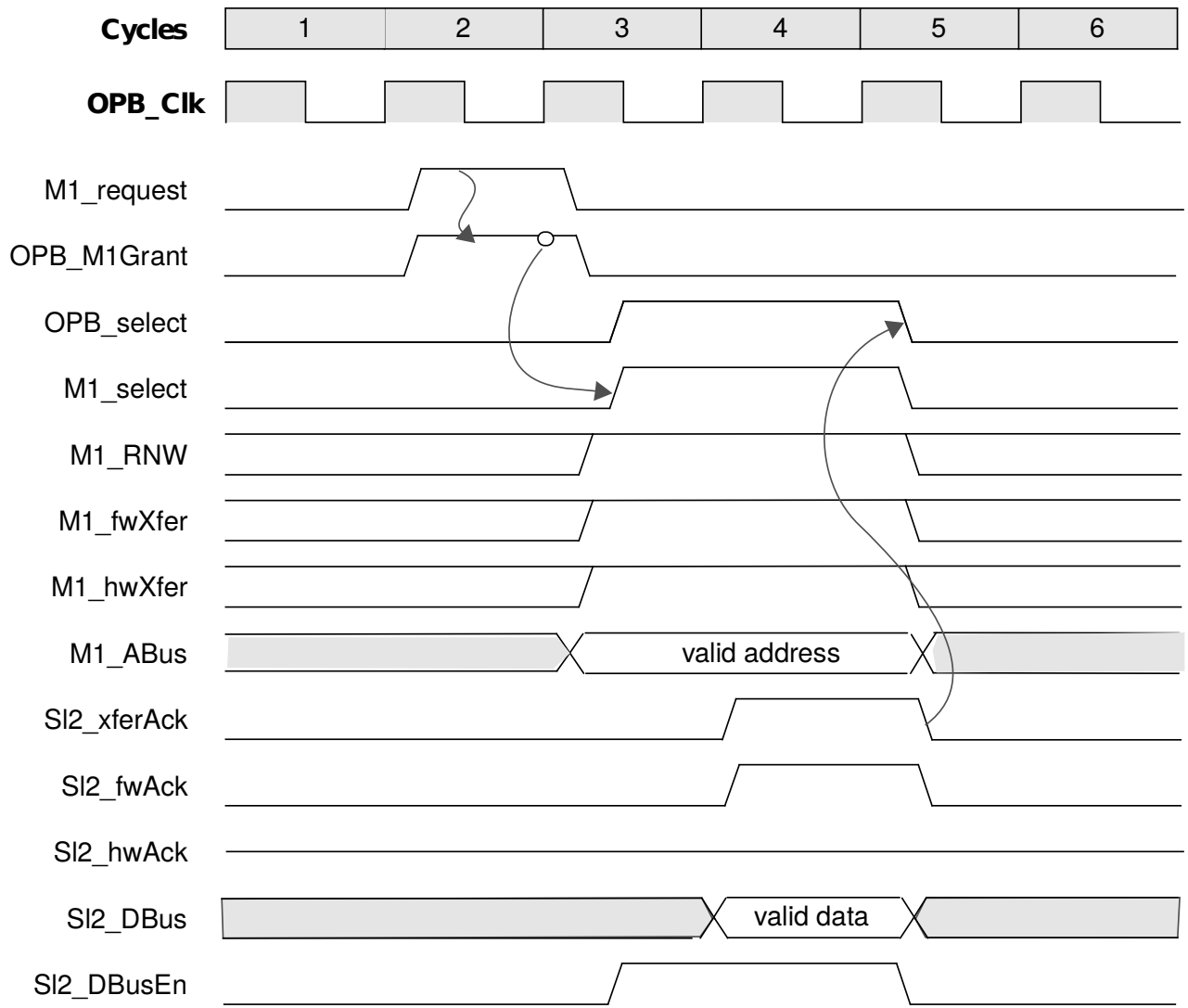
Señales Bus → Esclavo

- `OPB_select`: transferencia en curso
- `OPB_RNW`: a 1 es lectura, a 0 escritura
- `OPB_BE(0:3)`: bytes a transmitir
- `OPB_SeqAddr`: transferencia secuencial
  - Siguiente transferencia sobre siguiente dirección
  - Exige bloqueo de bus por parte del master
- `OPB_Abus(0:31)`: bus de direcciones
- `OPB_Dbus(0:31)`: bus de datos

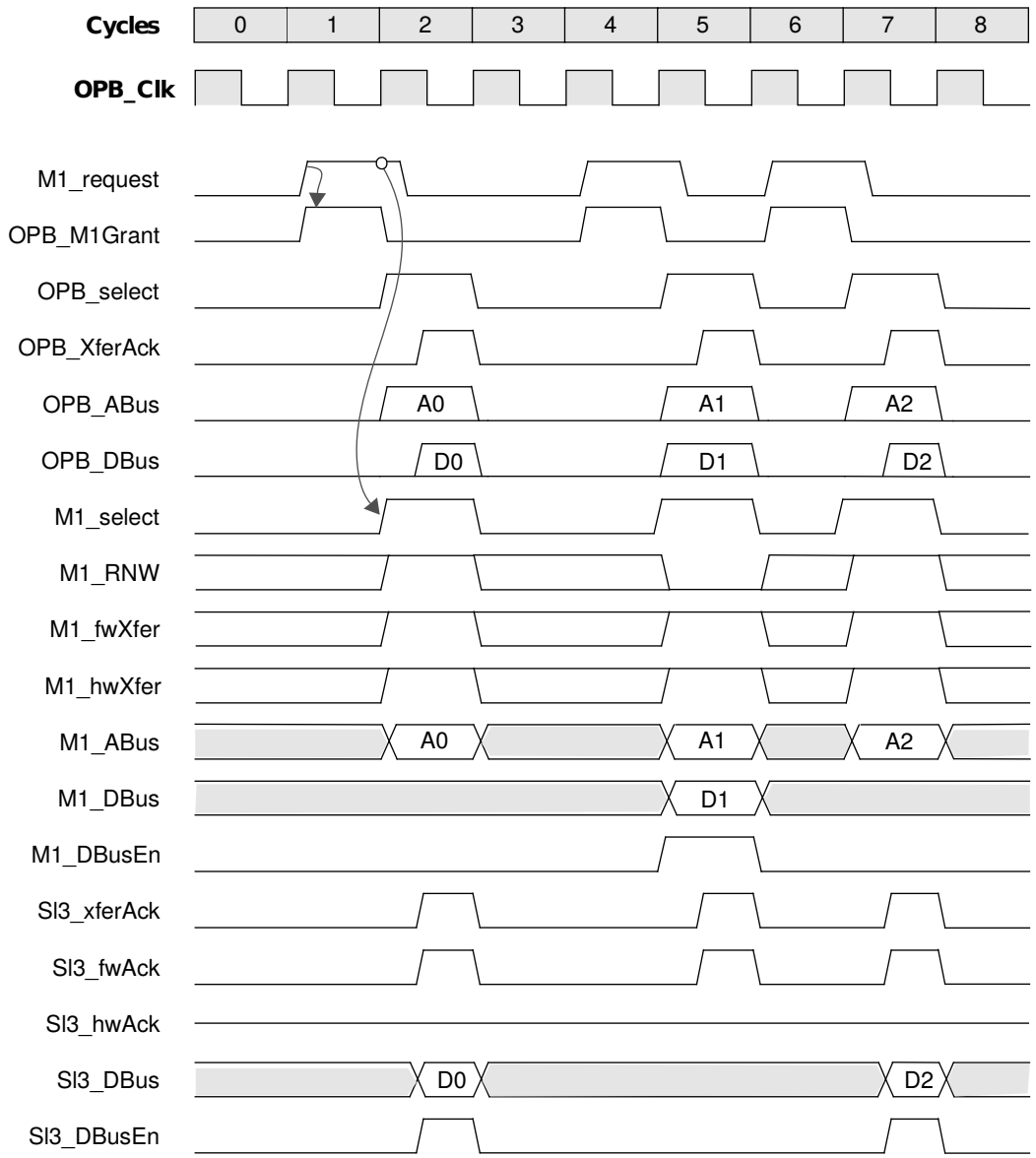
Señales Slave → Bus

- `Sln_xferACK`: transferencia completada
- `Sln_beACK`: reconocimiento transferencia be
- `Sln_errACK`: error en transferencia
- `Sln_toutSUP`: slave timeout suppress (opcional)
- `Sln_retry`: forzar fin de transferencia y re-arbitraje de bus
- `Sln_Dbus(0:31)`: entrada bus de datos

Transferencia Básica



Transferencia 1 ciclo

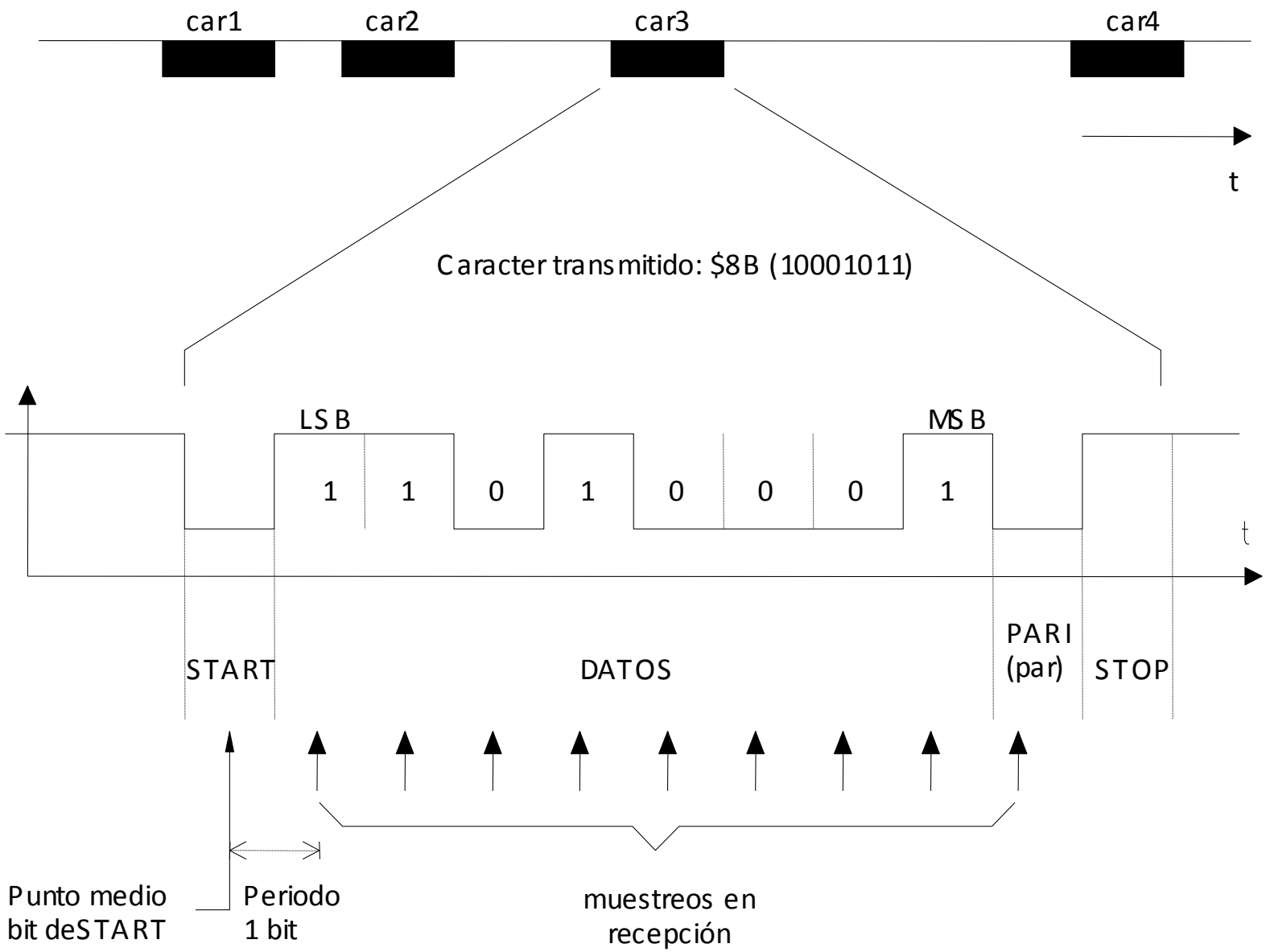


Agenda

UART

- Universal Asynchronous Receiver/Transmitter (UART)
  - Dispositivo que realiza la conversión paralelo/serie
  - Formato y velocidad variables
  - Uni-direccional, bidireccional (half- o full-duplex)
  - Usa transceivers externos para la generación de señales (+/-, diff)
- Se usa en varios puertos: RS-232, RS-422 y RS-485
  - Se requiere un driver MAX232 para adaptar niveles eléctricos

Tramado


car1
car2
car3
car4
t
Caracter transmitido: $8B (10001011)
LSB
MSB
1 1 0 1 0 0 0 1
t
START
DATOS
PARI (par)
STOP
Punto medio bit deSTART
Periodo 1 bit
muestreos en recepción

Agenda

Características

2 puertos de transmisión serie asíncrona bidireccionales

- Operan mediante encuesta (polling), interrupciones o DMA
- Permite usar buffers FIFO de 16 bytes (envío/recepción)
- Permite "handshake" (RS-232 por SW)
- Soporta IrDA 1.0

Configuración de los baudios

`UBRDIVn` (UART Baud Rate Division Register)

- Permite determinar la frecuencia de comunicación
- Fórmula para calcular el ratio

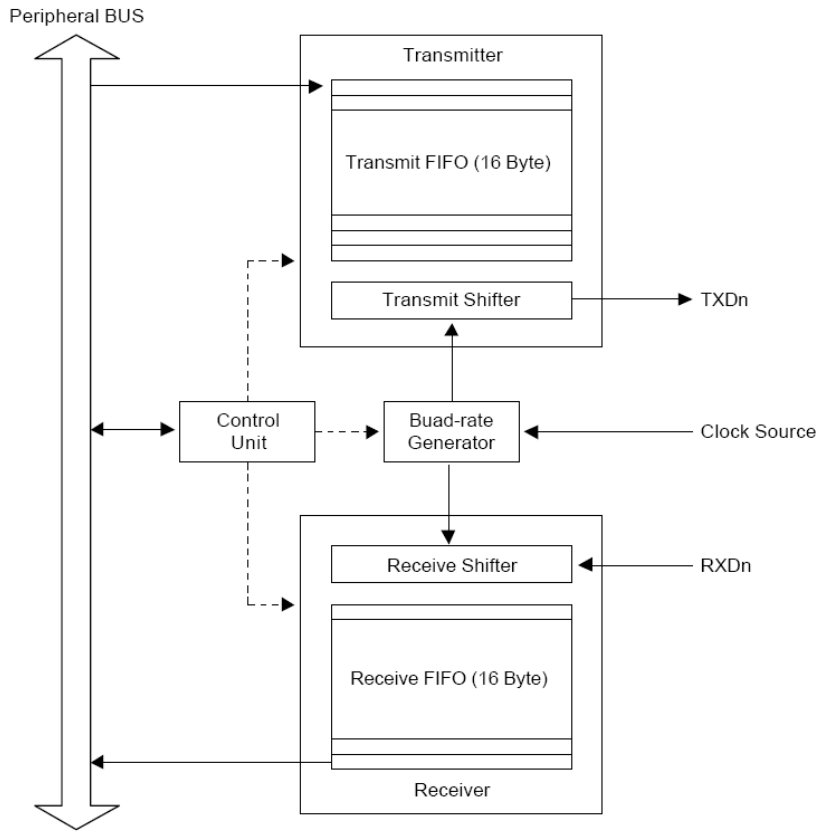
$$UBRDIVn = (round_off)(MCLK/(bpsx16)) - 1$$

- Ejemplo: ¿Valor de divisor para transmitir a 115200 bps, con frecuencia de reloj de 40 MHz?

$$\begin{aligned} UBRDIVn &= (int)(40000000/(115200x16) + 0,5) - 1 \\ UBRDIVn &= (int)(21,7 + 0,5) - 1 \\ UBRDIVn &= 22 - 1 = 21 \end{aligned}$$

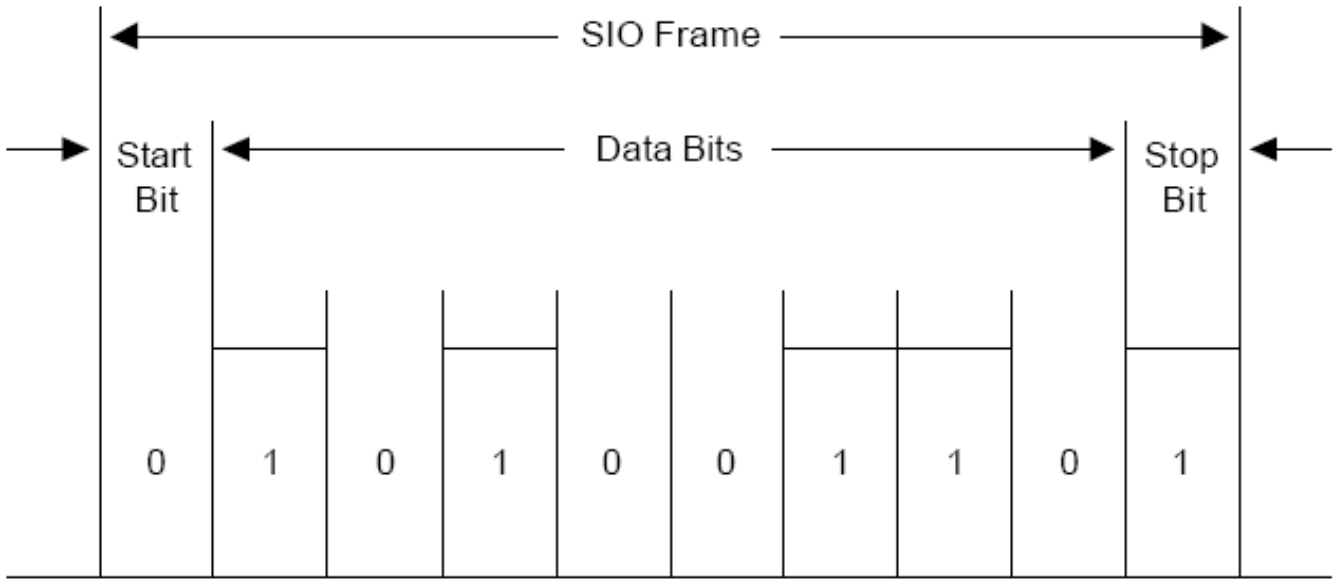
Estructura de cada canal

- Unidad de Control
- Buffers
- Unidad de generación de reloj
- Líneas envío/recepción



ULCONn (UART Line Control Register)

- Activar/desactivar modo IrDA
- Configurar paridad (none/even/odd)
- Determinar bits de stop (1/2)
- Longitud del carácter (5/6/7/8)



ULCONn (UART Line Control Register)

Register	Address	R/W	Description	Reset Value
ULCON0	0x01D00000	R/W	UART channel 0 line control register	0x00
ULCON1	0x01D04000	R/W	UART channel 1 line control register	0x00

ULCONn	Bit	Description	Initial State
Reserved	[7]		0
Infra-Red Mode	[6]	The Infra-Red mode determines whether or not to use the Infra-Red mode. 0 = Normal mode operation 1 = Infra-Red Tx/Rx mode	0
Parity Mode	[5:3]	The parity mode specifies how parity generation and checking are to be performed during UART transmit and receive operation. 0xx = No parity 100 = Odd parity 101 = Even parity 110 = Parity forced/checked as 1 111 = Parity forced/checked as 0	000
Number of stop bit	[2]	The number of stop bits specifies how many stop bits are to be used to signal end-of-frame. 0 = One stop bit per frame 1 = Two stop bit per frame	0
Word length	[1:0]	The word length indicates the number of data bits to be transmitted or received per frame. 00 = 5-bits 01 = 6-bits 10 = 7-bits 11 = 8-bits	00

Registro de control principal

`UCONn` (UART Control Register)

- Tipo de interrupción de envío/recepción (pulso/flanco)
- Activar interrupción por timeout
- Activar interrupción por error (break, frame, parity, overrun)
- Activar modo loop-back
- Mandar señal de break
- Determinar modo de transmisión para envío/recepción
  - Desactivado
  - Interrupción o encuesta
  - DMA

UCONn (UART Control Register)

Register	Address	R/W	Description	Reset Value
UCON0	0x01D00004	R/W	UART channel 0 control register	0x00
UCON1	0x01D04004	R/W	UART channel 1 control register	0x00

UCONn	Bit	Description	Initial State
Tx interrupt type	[9]	Interrupt request type 0 = Pulse (Interrupt is requested the instant Tx buffer becomes empty) 1 = Level (Interrupt is requested while Tx buffer is empty)	0
Rx interrupt type	[8]	Interrupt request type 0 = Pulse (Interrupt is requested the instant Rx buffer receives the data) 1 = Level (Interrupt is requested while Rx buffer is receiving data)	0
Rx time out enable	[7]	Enable/Disable Rx time out interrupt when UART FIFO is enabled. The interrupt is a receive interrupt. 0 = Disable 1 = Enable	0
Rx error status interrupt enable	[6]	This bit enables the UART to generate an interrupt if an exception, such as a break, frame error, parity error, or overrun error occurs during a receive operation. 0 = Do not generate receive error status interrupt 1 = Generate receive error status interrupt	0
Loop-back Mode	[5]	Setting loop-back bit to 1 causes the UART to enter the loop-back mode. This mode is provided for test purposes only. 0 = Normal operation 1 = Loop-back mode	0
Send Break Signal	[4]	Setting this bit causes the UART to send a break during 1 frame time. This bit is auto-cleared after sending the break signal. 0 = Normal transmit 1 = Send break signal	0
Transmit Mode	[3:2]	These two bits determine which function is currently able to write Tx data to the UART transmit holding register. 00 = Disable 01 = Interrupt request or polling mode 10 = BDMA0 request (Only for UART0) 11 = BDMA1 request (Only for UART1)	00
Receive Mode	[1:0]	These two bits determine which function is currently able to read data from UART receive buffer register. 00 = Disable, 01 = Interrupt request or polling mode 10 = BDMA0 request (Only for UART0) 11 = BDMA1 request (Only for UART1)	00

Registro de control de fifo y modem

`UFCONn` (UART FIFO Control Register)

- Habilitar FIFO
- Determinar el nivel de ocupación con el que se desencadenan las interrupciones de envío/recepción FIFO
- Borrar FIFO

`UMCONn` (UART Modem Control register)

- Habilitar control de flujo automático (AFC)
- Activar la señal RTS (Ready To Send) cuando el control de flujo es SW

Registros de estado de Tx/Rx

`UTRSTATn` (UART Tx/Rx Status Register)

- Determinar si el shifter de transmisión está vacio
- Determinar si el buffer de transmisión está vacio
- Deteminar si el buffer de recepción tiene datos listos

Register	Address	R/W	Description	Reset Value
UTRSTAT0	0x01D00010	R	UART channel 0 Tx/Rx status register	0x6
UTRSTAT1	0x01D04010	R	UART channel 1 Tx/Rx status register	0x6

UTRSTATn	Bit	Description	Initial State
Transmit shifter empty	[2]	This bit is automatically set to 1 when the transmit shift register has no valid data to transmit and the transmit shift register is empty. 0 = Not empty 1 = Transmit holding & shifter register empty	1
Transmit buffer empty	[1]	This bit is automatically set to 1 when the transmit buffer register does not contain valid data. 0 =The buffer register is not empty 1 = Empty If the UART uses the FIFO, users should check Tx FIFO Count bits and Tx FIFO Full bit in the UFSTAT register instead of this bit.	1
Receive buffer data ready	[0]	This bit is automatically set to 1 whenever the receive buffer register contains valid data, received over the RXDn port. 0 = Completely empty 1 = The buffer register has a received data If the UART uses the FIFO, users should check Rx FIFO Count bits in the UFSTAT register instead of this bit.	0

Otros registros de estado

`UERSTATn` (UART Error Status Register)

- Determinar el tipo de error que ha desencadenado una interrupción (break, frame, parity, overrun)

`UFSTATn` (UART FIFO Status Register)

- Determinar si el FIFO envío/recepción está lleno
- Determinar el nº de elementos en el FIFO

`UMSTATn` (UART Modem Status Register)

- Gestión de señal CTS (Clear To Send) en el control de flujo SW

Registros de Tx/Rx

`UTXHn/URXHn` (UART Transmit/Receive Holding Register)

- Registro en el que se escriben/leen los datos

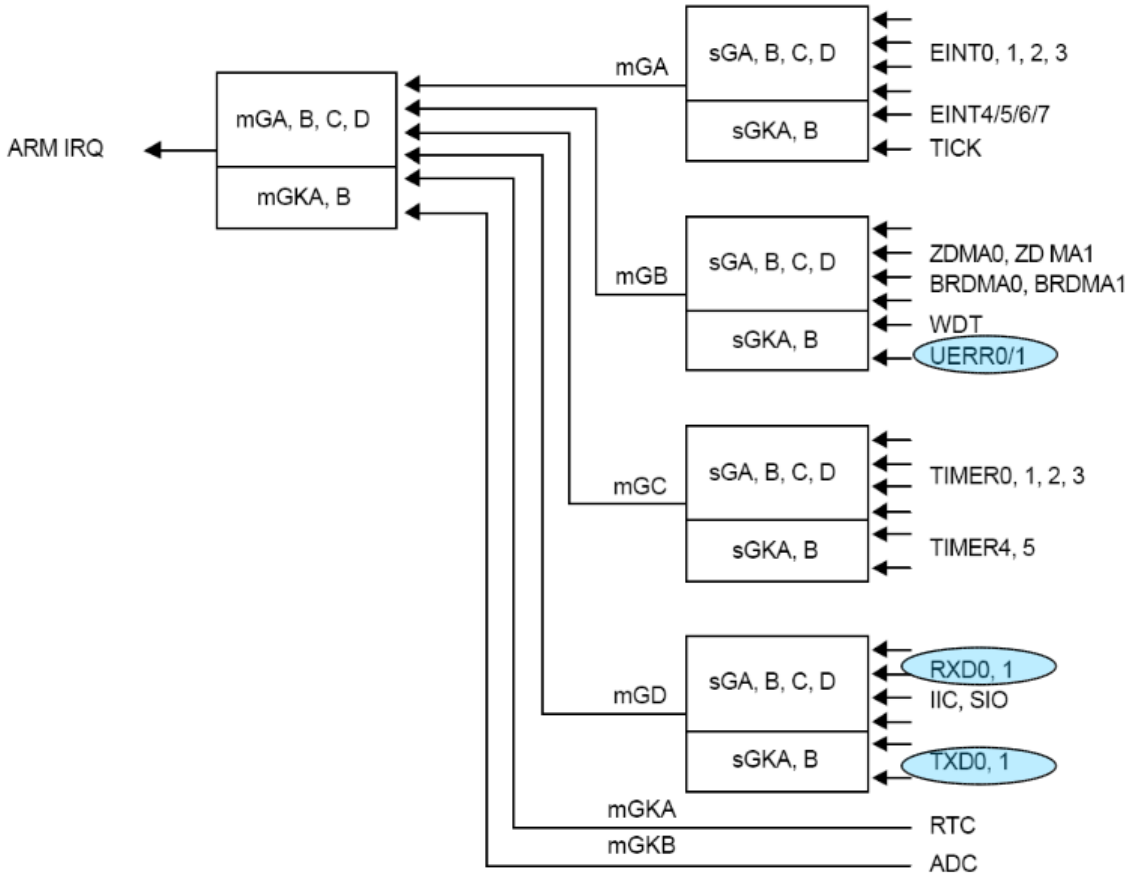
Register	Address	R/W	Description	Reset Value
UTXH0	0x01D00020(L) 0x01D00023(B)	W (by byte)	UART channel 0 transmit holding register	–
UTXH1	0x01D04020(L) 0x01D04023(B)	W (by byte)	UART channel 1 transmit holding register	–

Register	Address	R/W	Description	Reset Value
URXH0	0x01D00024(L) 0x01D00027(B)	R (by byte)	UART channel 0 receive buffer register	–
URXH1	0x01D04024(L) 0x01D04027(B)	R (by byte)	UART channel 1 receive buffer register	–

Interrupciones que genera

Interrupciones:

- 14 UERR0/1
- 7 URXD0
- 6 URXD1
- 3 UTXD0
- 2 UTXD1



Conexiones UART en la s3cev40

Conexión UART-DB9

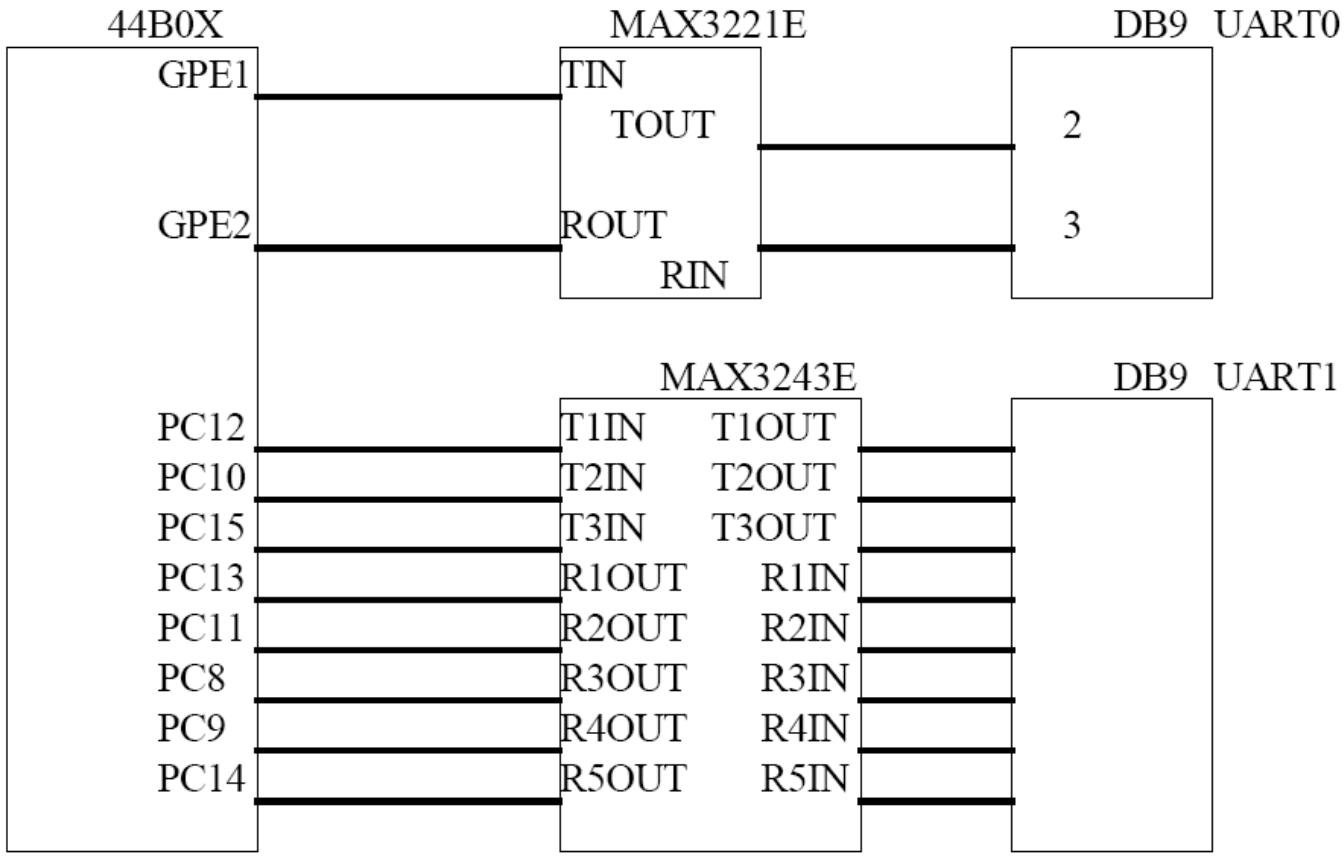


Figure 3-6 Serial Circuit