

Aula 16

- A interface RS-232C
- Estrutura das tramas
- Codificação dos sinais
- Sincronização de relógio
- Tolerância na frequência dos relógios do emissor e do recetor

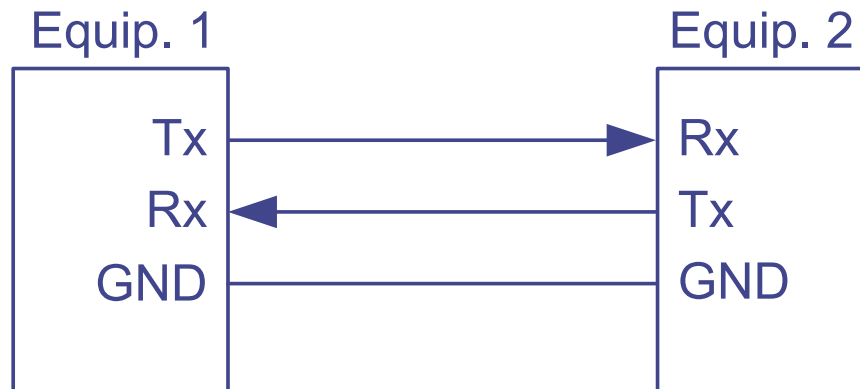
José Luís Azevedo, Bernardo Cunha, Tomás O. Silva, P. Bartolomeu

Introdução

- RS-232C – Standard (1969) para comunicação série assíncrona entre um Equipamento Terminal de Dados (DTE, e.g. computador) e um Equipamento de Comunicação de Dados (DCE, e.g. Modem)
- Permite comunicação bidirecional, *full-duplex*
- Transmissão orientada ao byte
- Conheceu uma grande utilização, que se estendeu muito para além do seu objetivo inicial (ligar DTEs a modems)
- Com o aparecimento do USB os computadores deixaram de disponibilizar comunicação RS-232C
- Por ser um modo de comunicação série muito fácil de implementar e de programar continua a ser muito usado em microcontroladores
- Apareceram no mercado conversores USB/RS-232C que permitem a ligação a PCs de equipamentos que implementam RS-232C

Sinalização

- Na sua forma mais simples, a implementação da norma RS-232C requer apenas a utilização de 2 linhas de sinalização e uma linha de massa



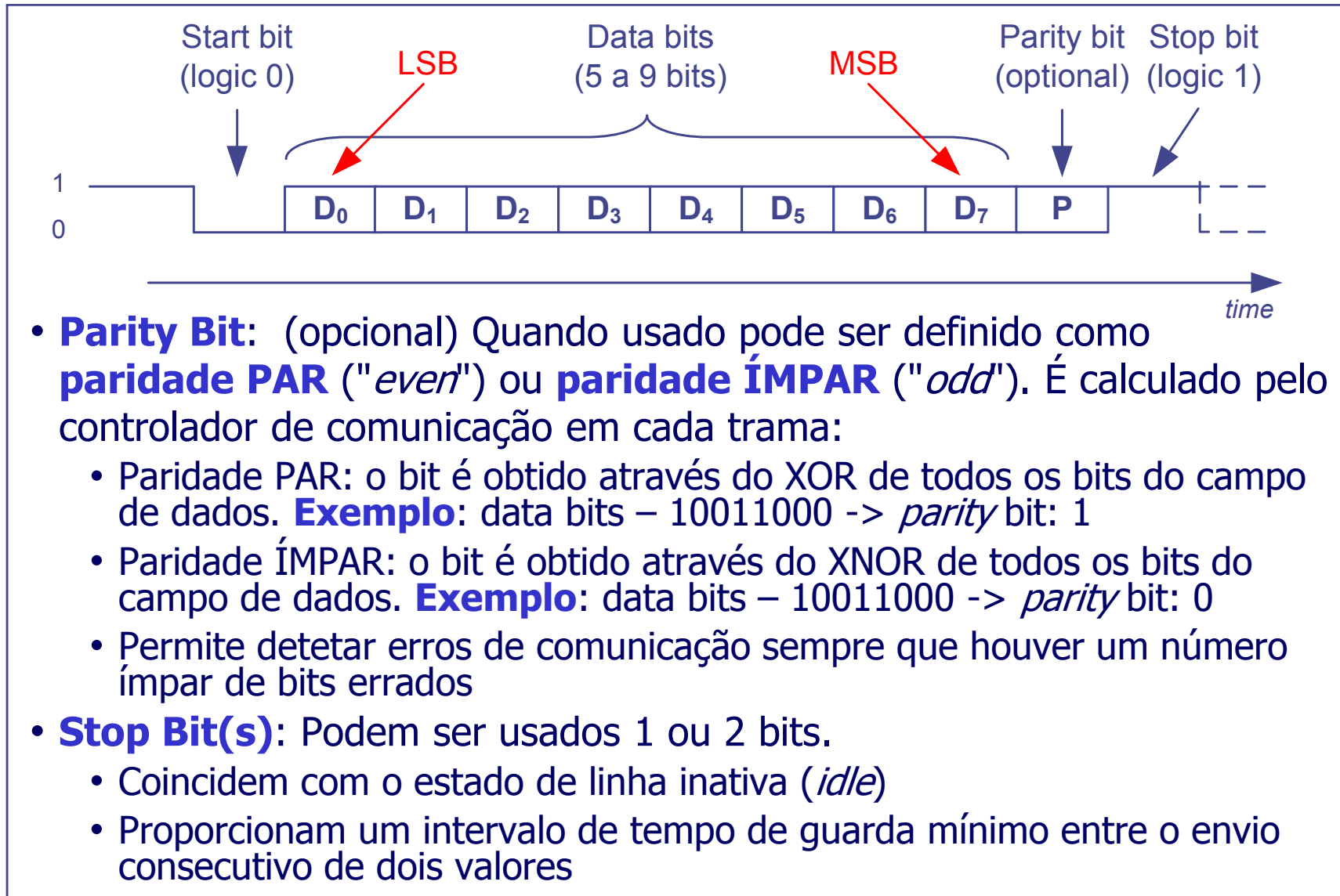
- Podem ser usadas linhas adicionais para protocolar a troca de informação entre os dois equipamentos (*handshake*)
 - RTS (*Request to send*)
 - CTS (*Clear to send*)
 - DTR (*Data terminal ready*)
 - DSR (*Data set ready*)

A norma original definia um total de 12 sinais (sendo 9 apenas para *handshaking*!)

Alguns problemas da norma RS-232C

- A nível físico (na linha de comunicação) os níveis lógicos são codificados com tensões simétricas (por exemplo +10V e -10V)
- Consumo de energia elevado
- Sinalização *single-ended*
 - Sinal é diferença entre tensão num fio e *common ground* (0V)
 - Baixa imunidade ao ruído
 - Impõe limitação na velocidade / distância
- Apenas suporta ligações ponto-a-ponto (implementações multi-ponto não standard)
- A norma era suficientemente vaga para permitir implementações proprietárias que, na prática, dificultavam ou mesmo impossibilitavam a interligação entre equipamentos de fabricantes diferentes

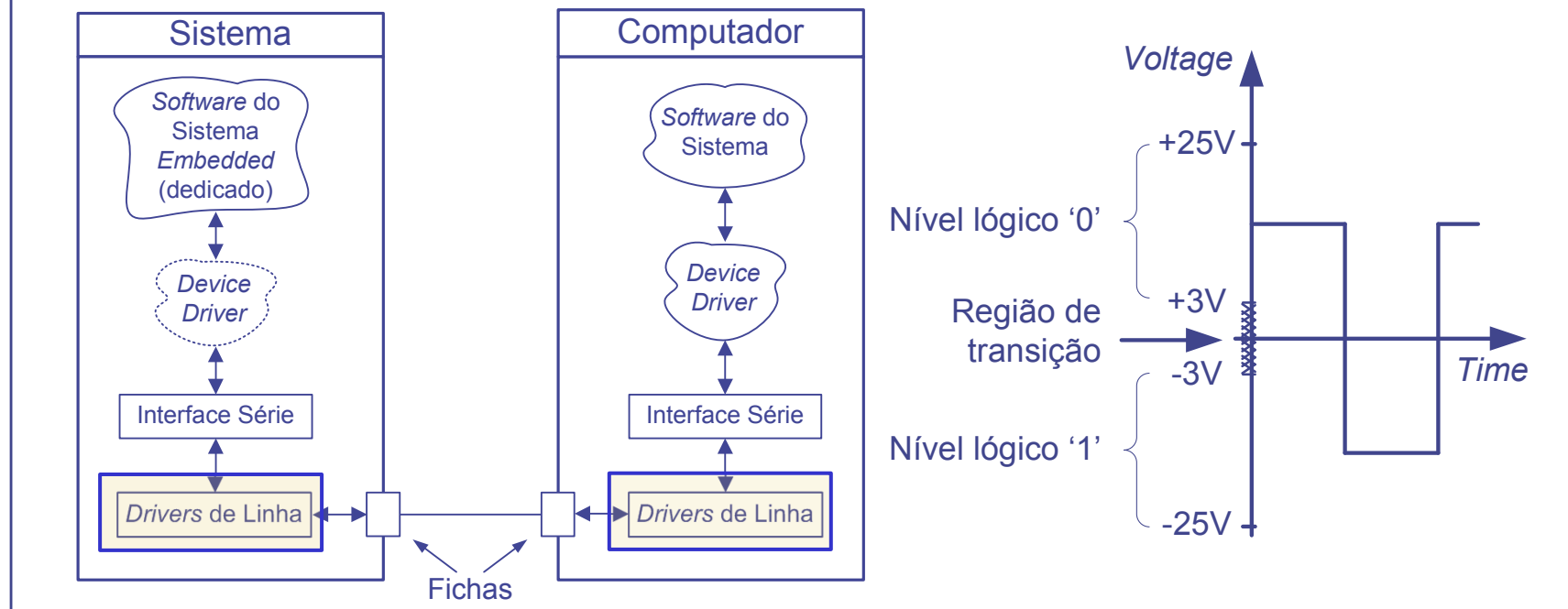
Estrutura de uma trama RS-232C



- **Parity Bit:** (opcional) Quando usado pode ser definido como **paridade PAR** ("even") ou **paridade ÍMPAR** ("odd"). É calculado pelo controlador de comunicação em cada trama:
 - Paridade PAR: o bit é obtido através do XOR de todos os bits do campo de dados. **Exemplo:** data bits – 10011000 -> *parity* bit: 1
 - Paridade ÍMPAR: o bit é obtido através do XNOR de todos os bits do campo de dados. **Exemplo:** data bits – 10011000 -> *parity* bit: 0
 - Permite detetar erros de comunicação sempre que houver um número ímpar de bits errados
- **Stop Bit(s):** Podem ser usados 1 ou 2 bits.
 - Coincidem com o estado de linha inativa (*idle*)
 - Proporcionam um intervalo de tempo de guarda mínimo entre o envio consecutivo de dois valores

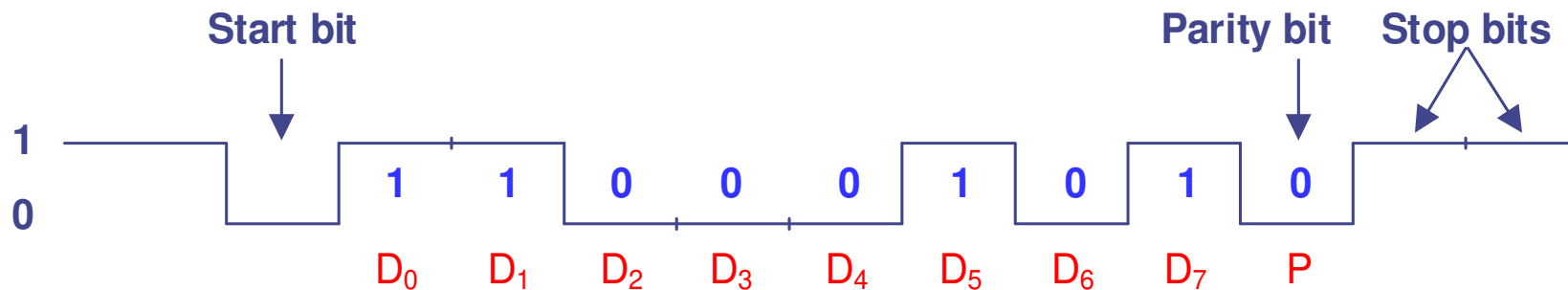
Camada física – codificação dos níveis lógicos

- Numa ligação física RS-232C os bits da trama são codificados em NRZ-L (*Non Return to Zero - Level*)
 - Nível lógico 1: codificado com uma tensão negativa (na gama -3V a -25V)
 - Nível lógico 0: codificado com uma tensão positiva (na gama +3V a +25V)
- A codificação e descodificação da trama com estes níveis de tensão é assegurada por circuitos eletrónicos designados por **drivers de linha**

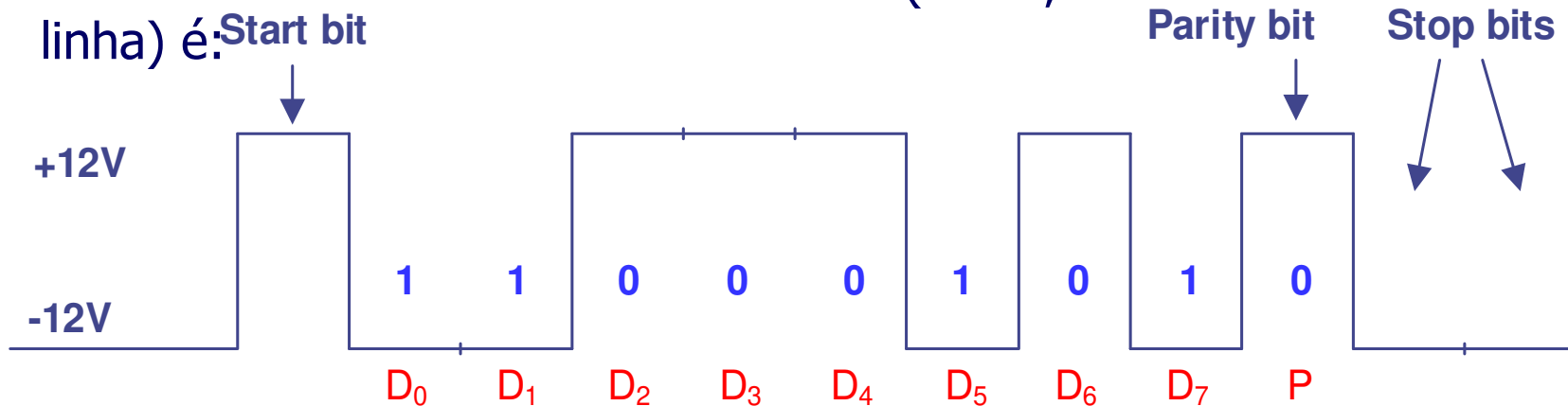


Exemplo: 8 bits de dados, 2 stop bits, paridade par

- A trama gerada pelo controlador de comunicação série RS-232C (e.g. UART-Universal asynchronous receiver/transmitter) para transmitir o valor 0xA3 é:



- A trama codificada em níveis RS-232C (isto é, à saída do driver de linha) é:



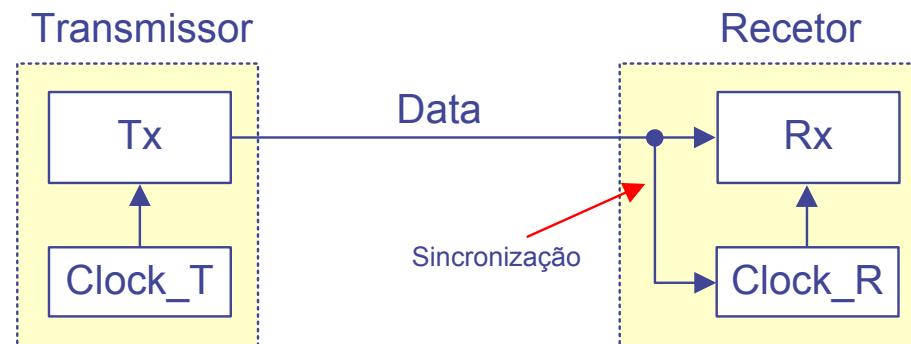
Baudrate (taxa de transmissão)

- O ***baudrate*** é, genericamente, o número de símbolos transmitidos por segundo. A cada símbolo pode corresponder um ou mais bits de dados
- A taxa de transmissão de dados bruta (*gross bit rate*) corresponde ao número de bits transmitidos por segundo (bps) (o *baudrate* não deve ser confundido com *gross bit rate*)
- No caso do RS-232C a cada símbolo está associado um único bit, logo o *baudrate* e o *gross bit rate* coincidem
- Exemplos comuns de *baudrates* em RS-232C [bps]: 600, 1200, 4800, 9600, 19200, 38400, 57600, 115200, 230400
- No exemplo anterior o número total de bits a serem transmitidos é 12
 - 1 start bit, 8 bits de dados, 1 bit de paridade, 2 stop bits
 - considerando um *baudrate* de 57600 bps a transmissão completa de uma trama demora $\sim 208 \mu\text{s}$ ($12 / 57600$)
 - o bit rate líquido é $(8 * 57600) / 12$, i.e., 38400 bps
 - o byte rate é: $38400 / 8 = 4800$ bytes/s

Receção de dados

- Sincronização de relógio: **relógio implícito**

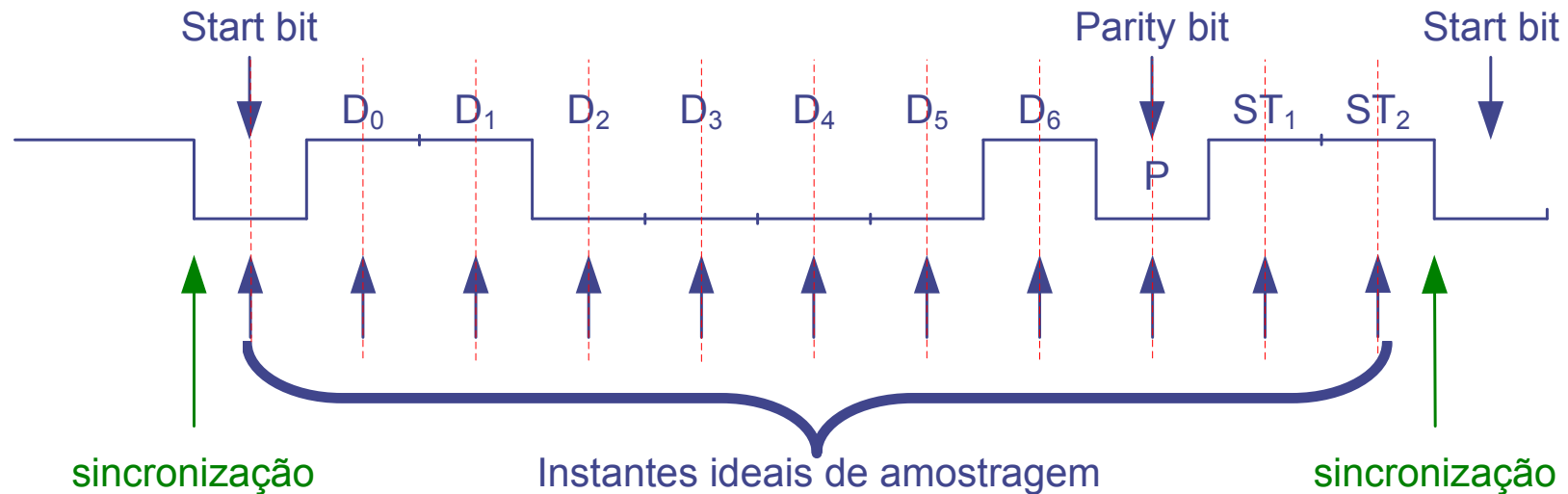
- Comunicação assíncrona (i.e. não há transmissão do relógio)
- O transmissor e o recetor têm relógios locais (independentes)



- A sincronização do processo de receção de dados é assegurada **no início da receção de cada nova trama** (sinalizado pelo **start bit**: transição de "1" para "0" na linha após um período de inatividade, por exemplo depois da receção completa de uma trama)
- Este método deve ser robusto, dentro de certos limites, a diferenças de frequência entre os relógios do transmissor e do recetor
 - Imprecisão na geração do relógio
 - Constante de divisão dos timers (que geram o relógio) não inteiras

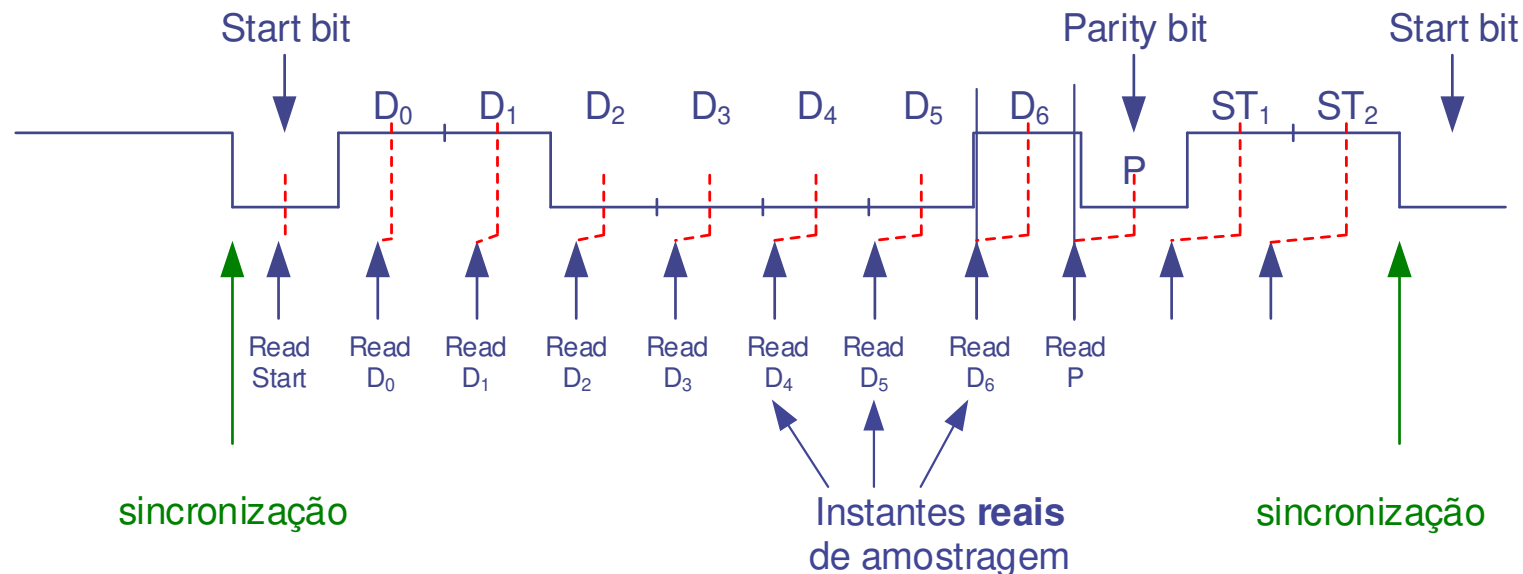
Receção de dados

- Para que a comunicação se processe corretamente, o transmissor e o recetor têm que estar **configurados com os mesmos parâmetros**:
 - Estrutura da trama: **nº de bits de dados, tipo de paridade, número de stop bits**
 - **Baudrate** (relógios com a mesma frequência)
- O recetor deve sincronizar-se pelo flanco negativo (transição do nível lógico "1" para o nível lógico "0") da linha (Start bit) e, idealmente, **fazer as leituras a meio do intervalo reservado a cada bit**
- Exemplo da receção do valor 0x43: estrutura da trama 7, 0, 2 (7 data bits, odd parity, 2 stop bits)



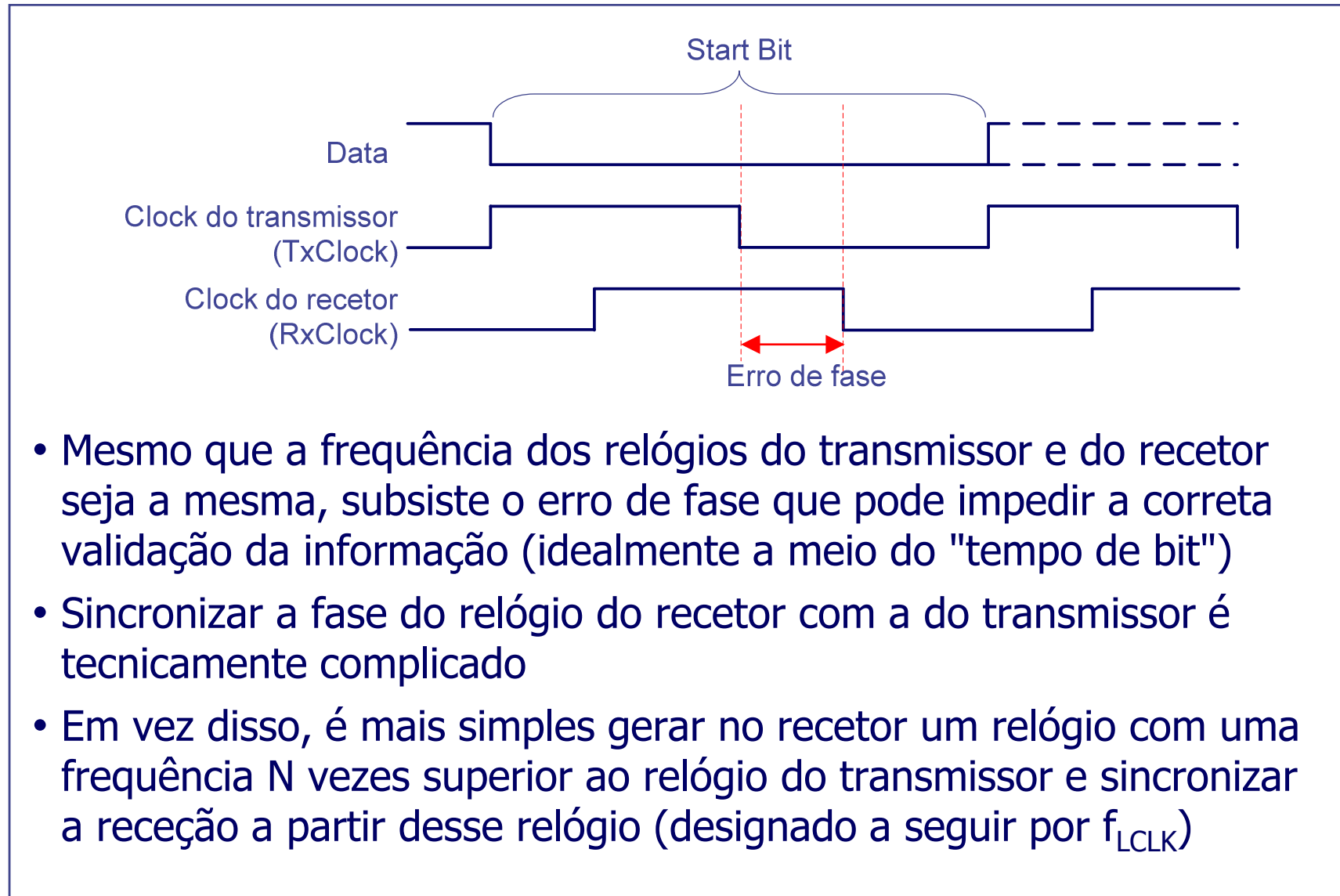
Sincronização

- Entre instantes de sincronização o desvio dos relógios depende da estabilidade/precisão dos relógios do transmissor e do recetor
- Exemplo em que a receção não é corretamente efetuada devido a um desvio da frequência dos relógios do transmissor e do recetor



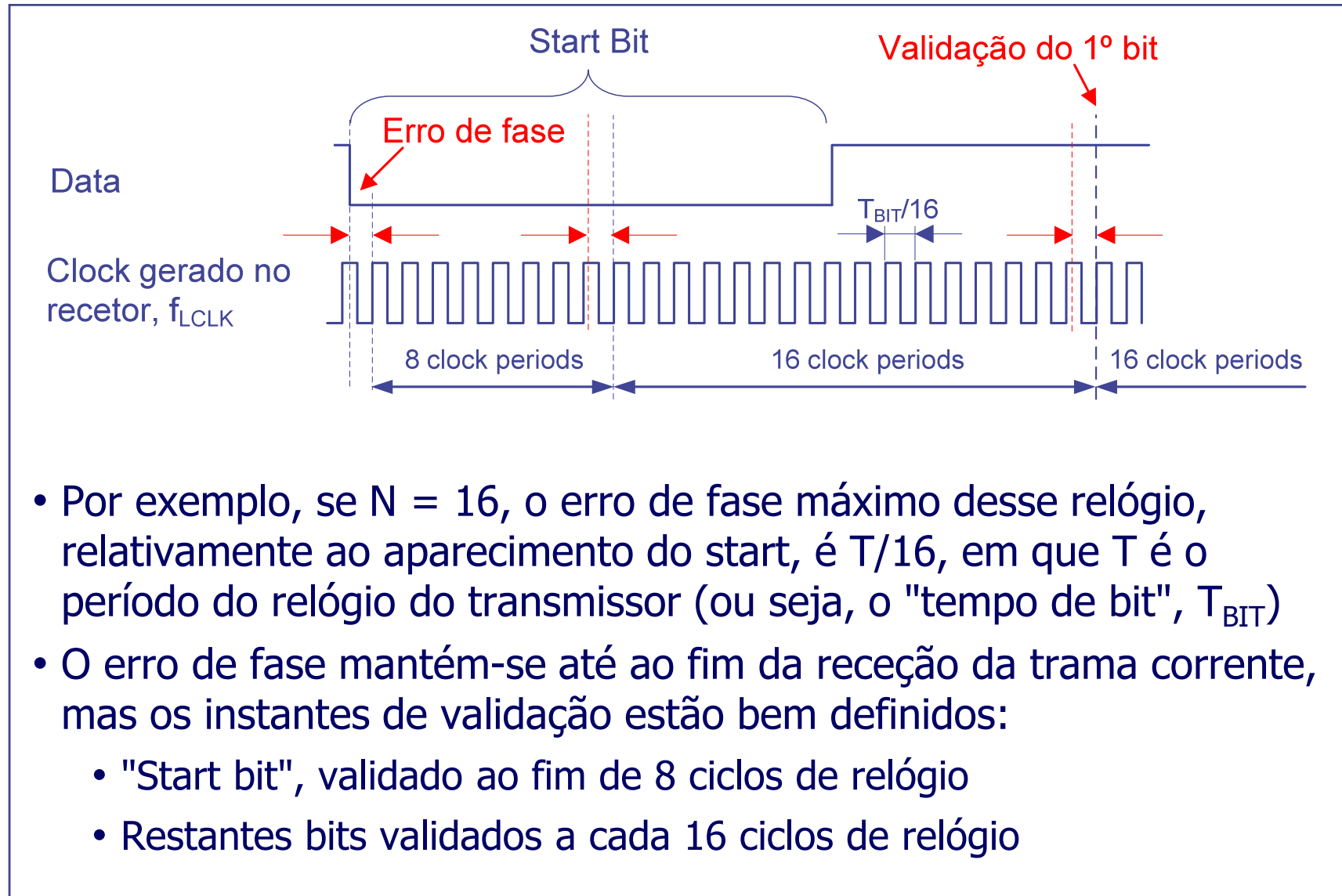
- Neste exemplo o recetor detetaria dois erros:
 - **Erro de paridade** (se o bit D₆ for detetado como 1): o bit de paridade devia ser 0 e é lido como 1
 - **Erro de framing**: é detetado o nível lógico 0 no instante em que era esperado um stop bit (nível lógico 1)

Sincronização



- Mesmo que a frequência dos relógios do transmissor e do recetor seja a mesma, subsiste o erro de fase que pode impedir a correta validação da informação (idealmente a meio do "tempo de bit")
- Sincronizar a fase do relógio do recetor com a do transmissor é tecnicamente complicado
- Em vez disso, é mais simples gerar no recetor um relógio com uma frequência N vezes superior ao relógio do transmissor e sincronizar a receção a partir desse relógio (designado a seguir por f_{LCLK})

Sincronização



Sincronização

- O relógio local (**LCLK**) deverá então ter, idealmente, uma frequência igual a:

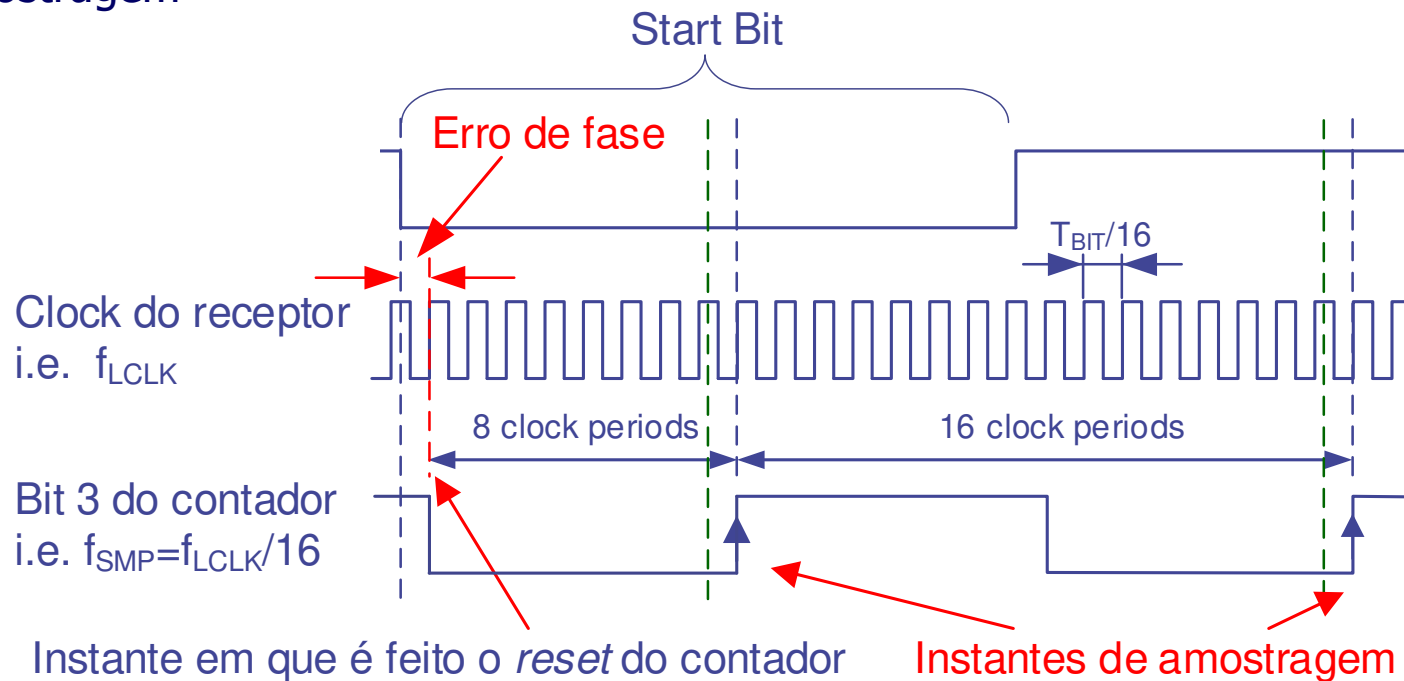
$$f_{LCLK} = N * f_{TCLK}$$

em que $f_{TCLK} = 1/T_{BIT}$ é a frequência do relógio de transmissão

- N é normalmente designado por **fator de sobreamostragem**
- Valores típicos de N: 4, 16, 64
- Esse relógio não é sincronizado com o sinal da linha, logo impõe um erro de fase (que é sempre inferior a um período, $\Delta_1 < T_{LCLK}$)
- Utilizando um relógio com N=16 o erro de fase máximo é **$T_{BIT}/16$** . Para N=64, o erro de fase máximo será **$T_{BIT}/64$**
- A utilização de fatores de sobreamostragem elevados nem sempre é possível (frequência da fonte de relógio disponível, constante de divisão do timer)

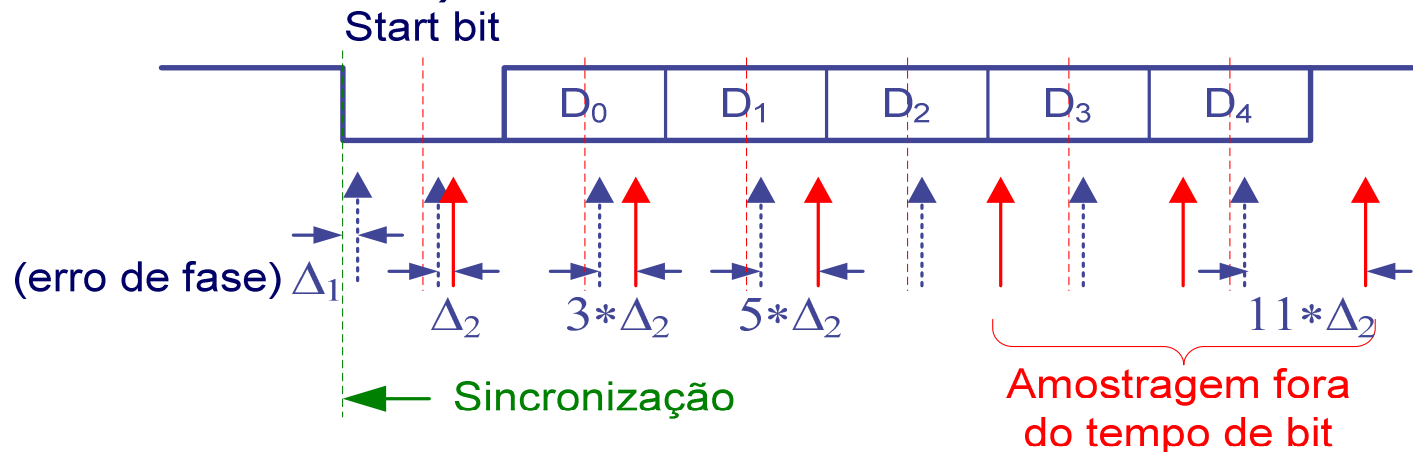
Sincronização – exemplo de implementação

- Considerando um fator de sobreamostragem (N) de 16, o tempo de bit equivale a 16 períodos do relógio local ($16T_{LCLK}$). O erro de fase máximo é $T_{BIT}/16$
- f_{SMP} é a frequência usada pelo recetor para a amostragem de cada bit da trama
- Usando um contador de 4 bits (com o relógio f_{LCLK}) como divisor de frequência por 16 (i.e. $f_0=f_{SMP}=f_{LCLK}/16$) a sincronização é trivial: basta fazer o reset (síncrono) desse contador quando é detetado o start bit
- As transições ascendentes do bit 3 do contador (MSBit) definem os instantes de amostragem



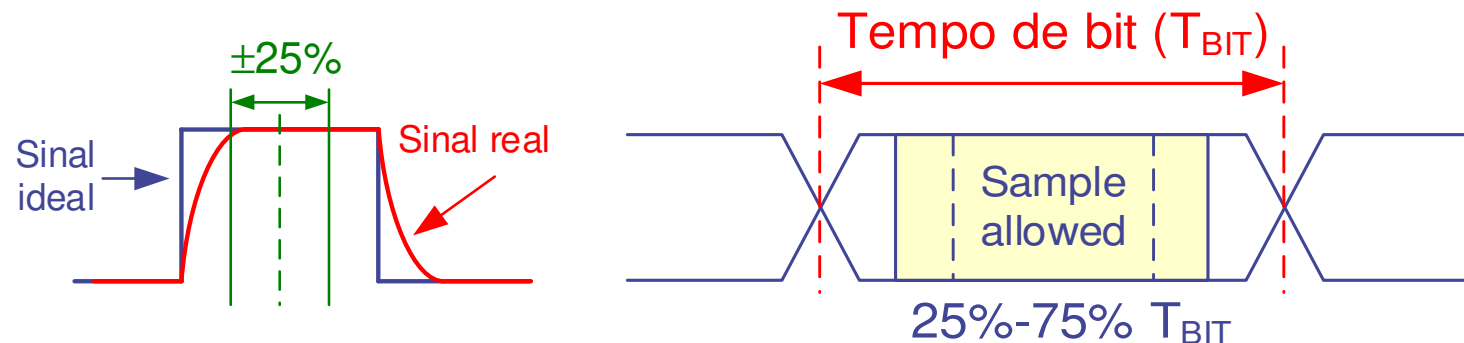
Sincronização – erro do instante de amostragem

- Os erros nos instantes de amostragem podem ter duas causas distintas:
 - **Erro de fase** (Δ_1): erro cometido ao determinar o instante inicial de sincronização
 - **Erro provocado por desvio de frequência** (Δ_2): a frequência dos relógios do transmissor e do recetor não são exatamente iguais (e.g. tolerância dos cristais de quartzo dos osciladores, constantes de divisão dos timers). Este erro é cumulativo e proporcional ao comprimento da trama (no caso da figura abaixo, no bit D_4 o erro é $11 * \Delta_2$)
- O efeito cumulativo destas fontes de erro pode originar erros na receção (como vimos no slide 11)



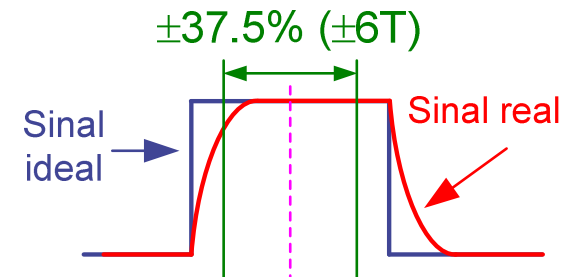
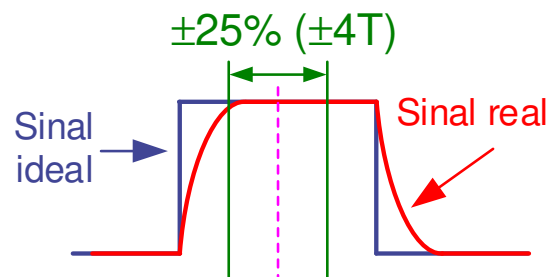
Máximo desvio de frequência entre emissor e recetor

- É comum considerar-se como zona segura de amostragem do bit:
 - **Pior caso** (cabos longos com efeito capacitivo pronunciado, velocidades de transmissão elevadas, ...): **$\pm 25\%$** do tempo de bit, em torno do instante ideal de amostragem, i.e., a meio
 - **Caso ideal** (cabos curtos e de acordo com as especificações, velocidades moderadas, ...): **$\pm 37,5\%$** do tempo de bit, em torno do instante ideal de amostragem



Máximo desvio de frequência entre emissor e recetor

- Considerando um fator de sobreamostragem (N) de 16, o tempo de bit equivale a 16 períodos do relógio local ($T_{\text{BIT}} = 16T_{\text{LCLK}}$)
- Assim, o desvio máximo admitido no instante de amostragem de um bit é de $\pm 4T_{\text{LCLK}}$ (pior caso, $\pm 0.25 \cdot 16$ ciclos) a $\pm 6T_{\text{LCLK}}$ (caso ideal, $\pm 0.375 \cdot 16$ ciclos)

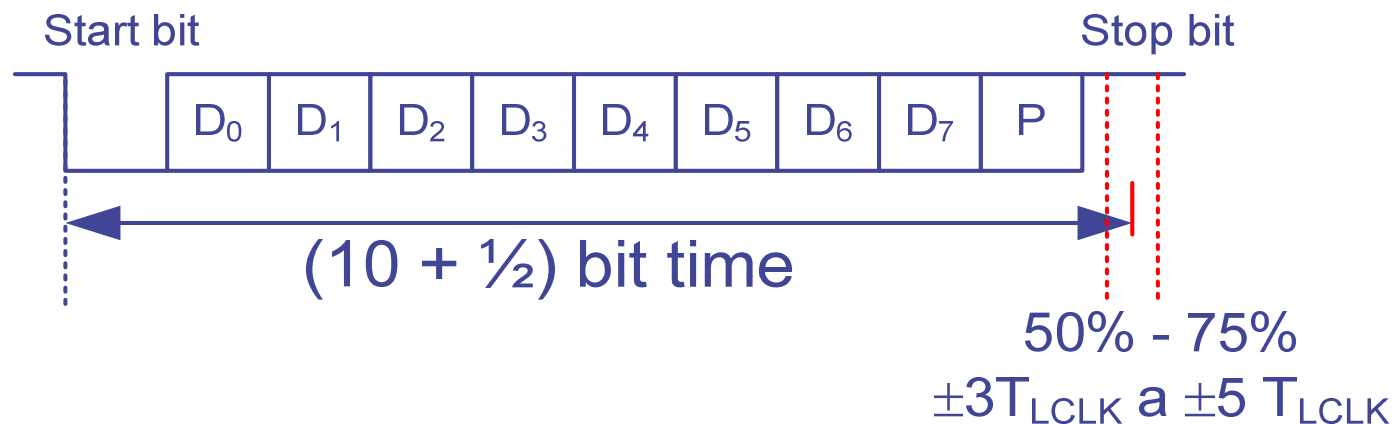


- Como há um erro intrínseco máximo de $1T_{\text{LCLK}}$ devido ao erro de fase, então o desvio máximo aceitável, resultante da diferença de frequência dos relógios, é de $\pm 3T_{\text{LCLK}}$ a $\pm 5T_{\text{LCLK}}$ (em cada instante de amostragem)

Esta correção do erro de fase para a obtenção do desvio máximo aceitável é conservativa. Porquê?

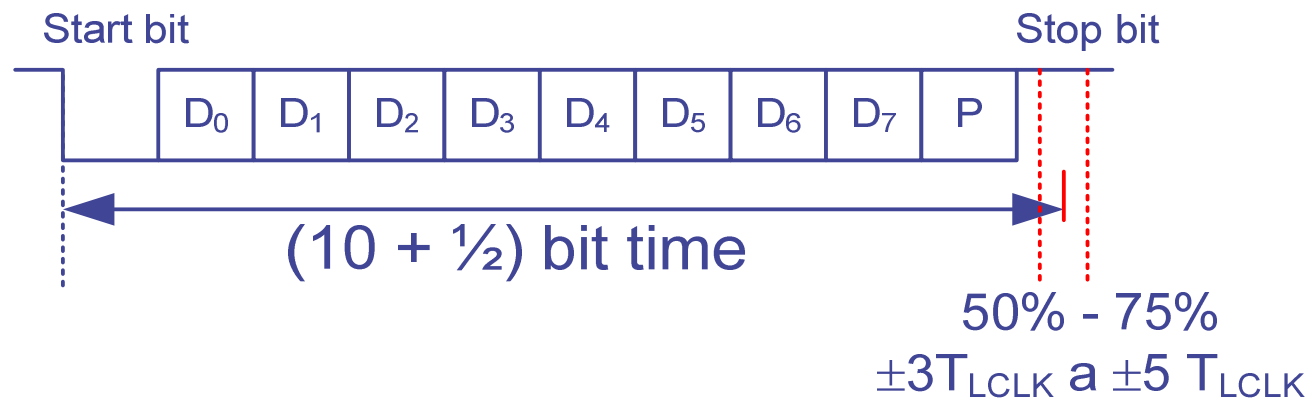
Máximo desvio de frequência entre emissor e recetor

- Ao contrário do erro de fase, o erro provocado por desvio de frequência é cumulativo e diretamente proporcional ao comprimento da trama
- Por outro lado é necessário garantir que o último bit da trama (independentemente do comprimento da trama) é sempre amostrado dentro da zona segura ($\pm 3T_{LCLK}$ a $\pm 5T_{LCLK}$, relativamente ao instante ideal de amostragem)



Máximo desvio de frequência entre emissor e recetor

- Consideremos então o caso em que pretendemos amostrar uma das tramas mais longas (start bit, 8 data bits, bit de paridade e 1 stop bit).

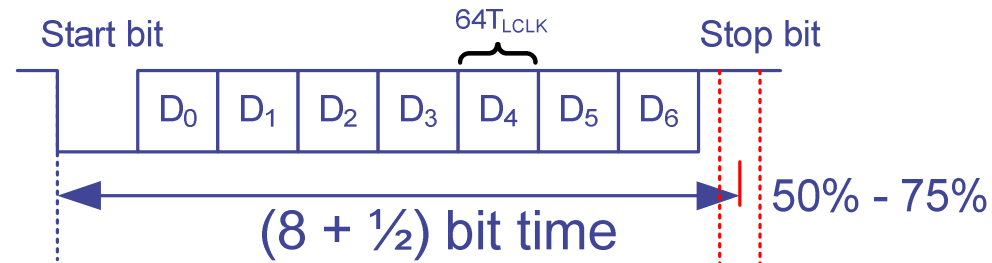


- Continuando a admitir que o relógio local tem um fator de sobreamostragem de $N=16$, para amostrar esta trama serão então necessários $10.5 \cdot 16 = 168$ períodos do relógio local
- Assim, a máxima discrepância que poderá ser tolerada entre os relógios do transmissor e do recetor é $\Delta T = \pm 3/168 \approx \pm 1.8\%$ (pior caso) a $\Delta T = \pm 5/168 \approx \pm 3.0\%$ (caso "ideal")

Máximo desvio de frequência entre emissor e recetor

- Como vimos, a máxima discrepância que poderá ser tolerada entre os relógios do transmissor e do recetor é $\Delta T = \pm 3/168 \approx \pm 1.8\%$ (pior caso) a $\Delta T = \pm 5/168 \approx \pm 3.0\%$ (caso "ideal") para as condições indicadas.
- **Exemplo:** vamos assumir que a taxa de transmissão é de 115200 bps, 8 data bits, parity bit, 1 stop bit e N=16. Para os dois casos-limite, para que a comunicação se processe sem erros, qual deve ser a gama de frequência do relógio do recetor?
- $T_{LCLK} = \frac{1}{(16*115200)} \gg f_{LCLK} = 1.843.200 \text{ Hz}$ (valor ideal)
- $T_{LCLK} = \frac{(1 \pm 0.018)}{(16*115200)} \gg f_{LCLK} \in [1.810.609, 1.876.986] \text{ Hz}$ (pior caso)
- $T_{LCLK} = \frac{(1 \pm 0.03)}{(16*115200)} \gg f_{LCLK} \in [1.798.515, 1.900.206] \text{ Hz}$ (melhor caso)
- **Exercício:** Admita que a configuração da comunicação é 38400 bps, 7 bits sem paridade, 1 stop bit e N = 64. Calcule o valor de frequência ideal no recetor e os intervalos admissíveis dessa frequência para os casos limite (tente resolver sozinho. A resolução está na versão em PDF)

Exercício: resolução



- Intervalo de validação (melhor caso): $\pm \left(\left(\frac{75\%}{2} * 64 \right) - 1 \right) = \pm 23T_{LCLK}$
- Intervalo de validação (pior caso): $\pm \left(\left(\frac{50\%}{2} * 64 \right) - 1 \right) = \pm 15T_{LCLK}$
- Nº de períodos de relógio para amostrar a trama = $8.5 * 64 = 544$
- Máxima discrepância temporal (melhor caso): $\frac{\pm 23}{544} \approx \pm 4.3\%$
- Máxima discrepância temporal (pior caso): $\frac{\pm 15}{544} \approx \pm 2.76\%$
- $T_{LCLK} = \frac{1}{(64 * 38400)} \gg f_{LCLK} = 2.457.600 \text{ Hz}$ (valor ideal)
- $T_{LCLK} = \frac{(1 \pm 0.0276)}{(64 * 38400)} \gg f_{LCLK} \in [2.391.592, 2.527.355] \text{ Hz}$ (pior caso)
- $T_{LCLK} = \frac{(1 \pm 0.043)}{(64 * 38400)} \gg f_{LCLK} \in [2.356.280, 2.568.025] \text{ Hz}$ (melhor caso)