

Projeto de uma interface UART (Universal asynchronous receiver/transmitter) utilizando tecnologia de 180nm

Guilherme Astolfo Rigacci

Agosto 2023

Universidade Federal de Minas Gerais

Programa Microeletrônica 2B (Frente acadêmica)

Orientado pelo Prof. Ricardo Duarte

1 Introdução

O barramento UART (Universal Asynchronous Receiver-Transmitter) é uma interface de comunicação serial amplamente utilizada para a transferência de dados entre dispositivos eletrônicos. Essa ampla utilização se dá pelo fato de ser uma interface assíncrona, dessa maneira, permitindo a comunicação de plataformas com frequências de operação distintas e dando do controle da taxa de transmissão para o projetista ou o usuário.

O UART é frequentemente empregado para estabelecer conexões de comunicação entre microcontroladores, sensores, módulos sem fio, periféricos e outros dispositivos eletrônicos. Ele transmite dados bit a bit, geralmente em formato de caracteres, utilizando uma combinação de bits de dados, bits de paridade (opcional) e bits de parada para delimitar os dados transmitidos[Bal19]

Os dispositivos conectados ao barramento UART devem concordar com as configurações de taxa de transmissão (baud rate), número de bits de dados, bits de paridade e bits de parada para garantir uma comunicação bem-sucedida[Bal19]. A simplicidade e a ampla adoção do barramento UART o tornaram uma escolha popular para aplicações que requerem uma comunicação serial confiável e direta. No entanto, ele não é adequado para transferência de grandes volumes de dados devido à sua natureza serial[new], que pode ser mais lenta em comparação com interfaces mais rápidas, como USB ou Ethernet.

Além disso, é possível diferenciar a interface em 3 tipos [roh]:

1. Simplex : o dado é enviado de forma unilateral.
2. Half-duplex : o dado é enviado de forma bilateral ,porém somente um pier pode transmitir por vez
3. Full-duplex : o dado é enviado de forma bilateral e ambos os pieres podem transmitir ao mesmo tempo

2 Funcionamento

A porta UART (Universal Asynchronous Receiver-Transmitter) é uma interface de comunicação serial que permite a troca de dados entre dispositivos eletrônicos. Ela funciona de maneira relativamente simples, transmitindo e recebendo dados em formato serial assíncrono, ou seja, sem a necessidade de um clock compartilhado entre os dispositivos para sincronização. Aqui está uma descrição do funcionamento básico da porta UART:

Transmissão de Dados:

1. Os dados a serem transmitidos são geralmente organizados em caracteres, que são sequências de bits que representam informações legíveis.
2. Antes de enviar um caractere, o transmissor UART aguarda até que a linha de transmissão esteja inativa (no nível lógico alto) para indicar que nenhum caractere está sendo enviado.
3. O caractere a ser transmitido é então dividido em bits individuais, começando pelo bit menos significativo (LSB).
4. Cada bit é enviado sequencialmente através da linha de transmissão. Isso é feito alterando o nível lógico da linha para representar o valor do bit (0 ou 1). A velocidade desse processo é determinada pela taxa de transmissão (baud rate) configurada.

Bits de Paridade (Opcional):

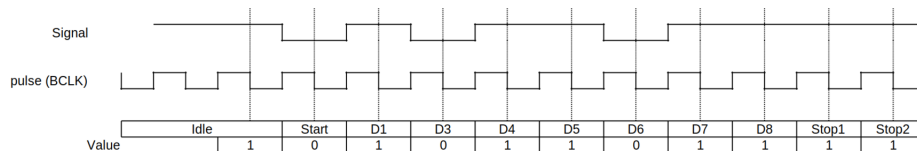
1. Um bit de paridade pode ser adicionado ao caractere transmitido para verificar a integridade dos dados durante a transmissão. Existem diferentes tipos de bits de paridade, como par, ímpar ou nenhum.
2. Se o bit de paridade estiver configurado como "par", por exemplo, o transmissor irá definir o bit de paridade de modo que a soma total dos bits de dados seja par. Isso pode ajudar a detectar erros de transmissão.

Bits de Parada:

1. Após a transmissão dos bits de dados (e, se necessário, do bit de paridade), um ou mais bits de parada são adicionados. Esses bits de parada indicam o final do caractere transmitido.
2. Geralmente, são usados um ou dois bits de parada. Eles são definidos como nível lógico alto para indicar que a linha de transmissão está inativa novamente.

Recepção de Dados:

1. No lado do receptor, a linha de recepção é monitorada em busca de mudanças nos níveis lógicos. Quando uma transição do nível alto para o nível baixo é detectada (indicando o início de um caractere), o receptor começa a contar o tempo para determinar quando ler os bits seguintes.
2. Os bits são lidos sequencialmente no tempo apropriado, de acordo com a taxa de transmissão configurada. Se um bit de paridade estiver sendo usado, o receptor pode verificar se a soma dos bits de dados coincide com o bit de paridade recebido.



Em resumo, a porta UART possibilita a comunicação serial assíncrona entre dispositivos por meio da transmissão e recepção de caracteres organizados em bits, com a adição de bits de paridade e de parada para garantir a integridade e a sincronização dos dados transmitidos.

3 Repositórios auxiliares

Como embasamento do projeto, foram procurados repositórios de código que corroborem com o funcionamento RTL aqui desejado.

1. stnolting/neorv32 : [Nol23]
2. pabennett/uart : [Ben16]

References

- [Bal19] Alexandro Baldassin. “COMUNICAÇÃO SERIAL (UART)”. In: (2019). URL: <http://www1.rc.unesp.br/igce/demac/alex/disciplinas/MicroII/EMA864315-Serial.pdf>.

- [Ben16] Peter Bennett. In: (2016). URL: <https://github.com/pabennett/uart>.
- [new] newton-braga. “Como funcionam as UARTs (TEL006)”. In: (). URL: <https://newtoncbraga.com.br/index.php/telecom-artigos/1709->.
- [Nol23] Stephan Nolting. In: (2023). URL: https://github.com/stnolting/neorv32/blob/main/rtl/core/neorv32_uart.vhd.
- [roh] rohde-schwarz. “Understanding UART”. In: (). URL: https://www.rohde-schwarz.com/us/products/test-and-measurement/essentials-test-equipment/digital-oscilloscopes/understanding-uart_254524.html.