

Universidade Federal de Minas Gerais

Programa Microeletrônica 2B (Frente Acadêmica)
Orientado pelo Prof. Ricardo Duarte

Projeto de uma interface Inter-Integrated Circuit (I²C) utilizando tecnologia de 180nm

Vinícius Guimarães Silva

8 de Agosto de 2023

INTRODUÇÃO

O barramento I²C caracteriza-se por ser uma interface de comunicação serial de dois fios. Nesse sentido, o I²C permite que múltiplos dispositivos, em uma relação mestre-ajudante, troquem informações de forma eficiente e síncrona. Para tal, o I²C, vale-se de uma linha de dados serial (SDA) e uma linha de clock serial (SCL) que viabilizam diferentes modos de transferência de dados, como leitura, escrita, envio de comandos e confirmação de recebimento [1]. Apesar das vantagens associadas ao I²C, é importante ressaltar que o número de dispositivos em um único barramento é limitado pela capacitância deste. [2]

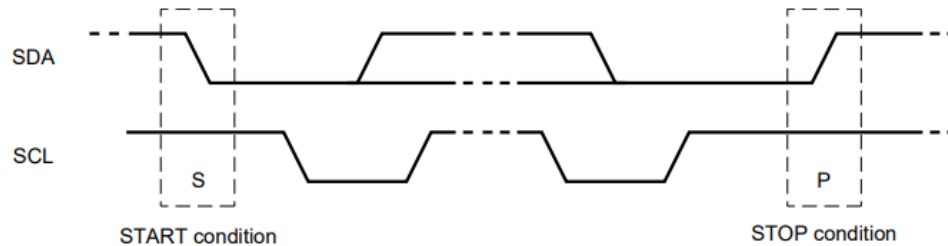
Ainda, nesse sentido, o I²C endereça unicamente cada um dos dispositivos do barramento de modo que as relações controlador/alvo existam o tempo todo com o controlador operando como receptor ou transmissor. Além do mais, em um sistema com múltiplos controladores, faz-se necessário incluir as funções detecção de colisão e arbitragem para prevenir o corrompimento dos dados se dois ou mais controladores iniciarem a transmissão de dados simultaneamente.

FUNCIONAMENTO

Diante das funções definidas anteriormente, faz-se necessário a descrever o funcionamento dos protocolos que contribuem com a eficiência e confiabilidade da comunicação entre os diversos componentes do sistema. Nesse sentido, destaca-se que os protocolos, a serem descritos, desempenham um papel fundamental na garantia da troca de informações de forma organizada e padronizada, possibilitando que o microcontrolador ou dispositivo mestre se comunique com os componentes periféricos, como sensores, atuadores e outros dispositivos. Dessa forma, destaca-se mandatoriamente para um barramento I²C com um único controlador as funções, a saber: (i) Condição de início; (ii) Condição de finalização; (iii) *Acknowledge* e (iv) Endereçamento de 7-bits. No que tange à aplicação de dois ou mais controladores em um barramento I²C, para satisfazer condições de concorrência, adiciona-se mandatoriamente as funções, quais sejam: (i) Sincronização e (ii) Arbitragem. Salienta-se que existem funções opcionais que se aplicam em ambos os modos: (i) Alongamento do *clock*; (ii) Endereçamento de 10-bits; (iii) Byte de iniciação, (iv) *Multicast*.

Primeiramente, na interface serial I²C define-se que para início e término de toda comunicação, deve-se satisfazer uma (i) condição de início e uma (ii) condição de finalização, respectivamente. Desse modo, tem-se que para uma transição de nível lógico alto para nível lógico baixo na linha de SDA durante um nível lógico alto na linha de SCL a definição da condição de *Start*. De modo semelhante, tem-se que para uma transição de nível lógico baixo para nível lógico alto na linha de SDA durante um nível lógico alto de SCL a definição da condição de *Stop*. Salienta-se que ambas as condições de *Start* e de *Stop*, são geradas por um controlador. Ademais, ressalta-se que uma vez instaurada a condição de *Start* o barramento I²C se encontrará em estado ocupado e só será considerado livre após um tempo depois de uma condição de *Stop*. A Figura 1, informa as condições de início e de finalização conforme descrição.

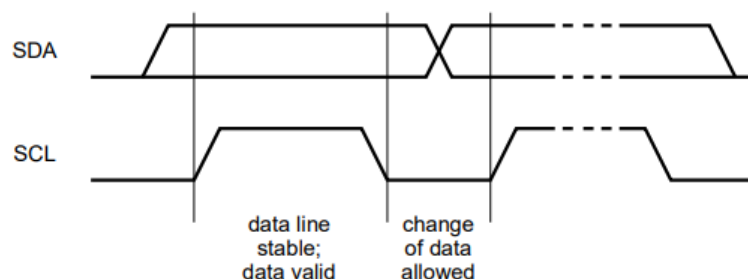
Figura 1: Condições de *Start* e *Stop* nas linhas SDA e SCL do barramento I²C.



Fonte: NXP SEMICONDUCTORS. I2C Bus Specification and User Manual, 2020 [S.l.] [2]

Em seguida, após a instauração e sucesso da condição de início, a linha serial de dados, SDA, estará apta a transmissão dos dados, para tanto é necessário especificar a forma de como os dados são transmitidos. Desse modo, especifica-se primeiramente a condição para validade dos dados na linha SDA, para tal, ressalta-se que o nível lógico do dado na linha serial de dados deve ser estável durante um nível lógico alto da linha de clock, SCL. Portanto, a mudança dos dados ocorre durante o nível lógico baixo da linha de clock. Com a validação da transmissão de dados, cada pulso de clock é gerado para cada bit transmitido. A Figura 2 mostra a transferência de um bit no barramento I²C.

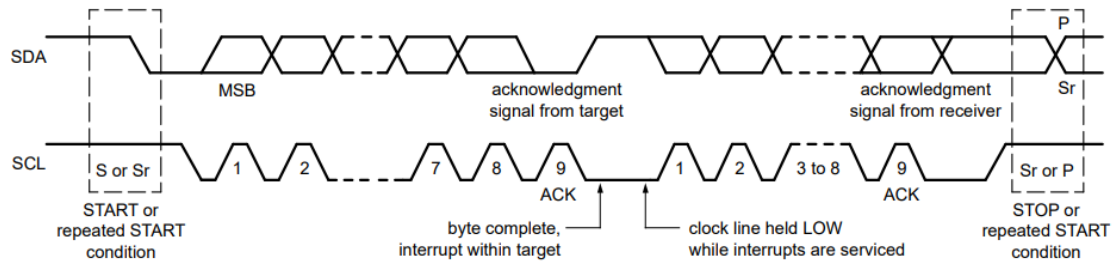
Figura 2: Linhas de SDA e SCL na transferência de 1 bit.



Fonte: NXP SEMICONDUCTORS. I2C Bus Specification and User Manual, 2020 [S.l.] [2]

Ainda em conformidade com a especificação do formato dos dados na SDA, especifica-se que todo byte transmitido na linha SDA deve ser de oito bits, acompanhado em seguida de um bit de reconhecimento (iii) *Acknowledge* (ACK). Ademais, o número de bytes durante uma transmissão é ilimitado e todo dado é transmitido segundo a representatividade do bit mais significativo (MSB). Dessa forma, para exemplificar uma transferência em um barramento I²C, observa-se a Figura 3, na qual tem-se a condição de partida para início da comunicação, seguida das transferências ilimitadas de bytes com os devidos ACKs e por fim tem-se a condição de finalização. Nota-se que é possível segurar a linha de clock em um nível lógico baixo caso o alvo da comunicação não possa receber ou transmitir um byte completo.

Figura 3: Linhas de SDA e SCL na transferência de dados.

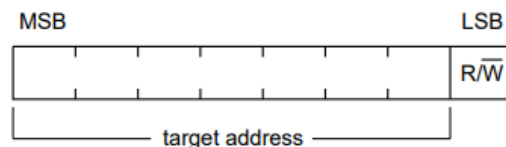


Fonte: NXP SEMICONDUCTORS. I2C Bus Specification and User Manual, 2020 [S.1.] [2]

Sequencialmente, a função do ACK, que sempre segue os bytes enviados, diz respeito a um bit de reconhecimento para que o receptor informe ao transmissor que o byte foi recebido com sucesso e outro byte possa ser enviado. O sinal de ACK, desse modo, define-se como um bit a ser transmitido em um nível lógico baixo após os 8 bits relacionados ao byte enviado. Semelhantemente, tem-se o *Not Acknowledge* (NACK) que caracteriza-se por um nível lógico alto ao invés de baixo após o envio de um byte. O NACK, diferentemente do ACK, ocorre em condições de erro, quais sejam: (i) Não há um endereçamento condizente com o receptor no barramento; (ii) O receptor está inapto. (iii) O receptor recebe dados não conhecidos durante a transferência; (iv) O receptor está sobrecarregado.

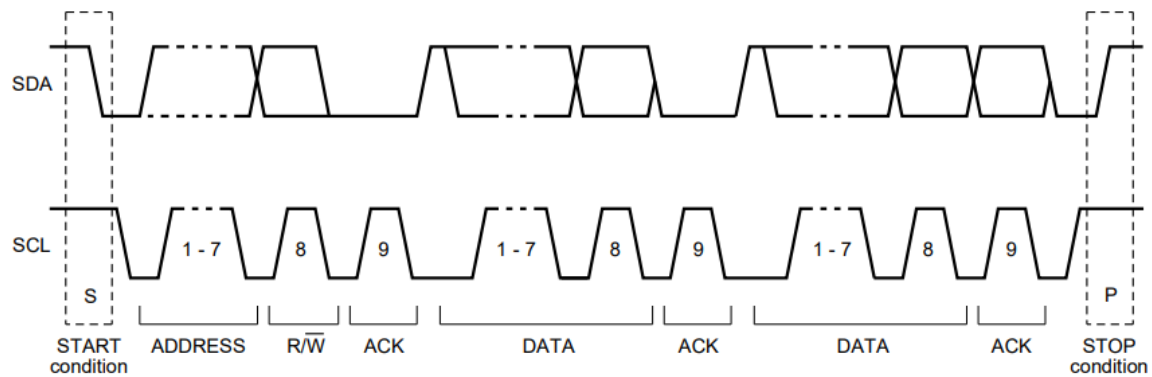
Por fim, diante das descrições apresentadas, descreve-se o papel do primeiro byte enviado. Nesse sentido, o primeiro byte enviado contém informações que dizem respeito ao endereçamento (7 bits — alvos da comunicação) e habilitação de escrita (1 bit — *Read/Write*), conforme a Figura 4. Vale ressaltar que, um nível lógico baixo no bit de R/W indica escrita e o contrário leitura. Desse modo uma comunicação completa, em termos das linhas SDA e SCL, desde a condição de *Start*, seguida pelo primeiro byte, transmissão de dados e finalização da transmissão com a condição de *Stop* pode ser visualizada conforme a Figura 5.

Figura 4: Primeiro byte enviado.



Fonte: NXP SEMICONDUCTORS. I2C Bus Specification and User Manual, 2020 [S.1.] [2]

Figura 5: Transferência completa.



Fonte: NXP SEMICONDUCTORS. I2C Bus Specification and User Manual, 2020 [S.l.] [2]

REPOSITÓRIOS

Em conformidade com o desenvolvimento da interface serial I²C em RTL, buscou-se repositórios livres que poderiam embasar o encaminhamento do projeto. Dessa forma, apresenta-se a Tabela 1 com informações e referências dos repositórios relevantes.

Ref.	DESCRIÇÃO
[3]	Projeto do NEORV32, confiável, atualizado. Entretanto, não tem suporte para múltiplos controladores na mesma linha SCL e SDA.
[4]	Possui documentação da implementação [5], confiável e suporte a múltiplos controladores. Última atualização em 2018.

REFERÊNCIAS

- [1] AFZAL, Sal. I2C Primer: What is I2C? (Part 1). Analog Devices [S.l.]: Disponível em: <https://www.analog.com/en/technical-articles/i2c-primer-what-is-i2c-part-1.htm> l. Acesso em: 11 jul. 2023.
- [2] NXP SEMICONDUCTORS. I2C Bus Specification and User Manual, 2020 [S.l.]: Disponível em: <https://www.nxp.com/docs/en/user-guide/UM10204.pdf> . Acesso em: 11 jul. 2023.
- [3] Nolting, S., & All the Awesome Contributors. (2022). The NEORV32 RISC-V Processor [Computer software]. Disponível em: <https://doi.org/10.5281/zenodo.7030070>. Acesso em: 7 ago. 2023.
- [4] HERVEILLE, Richard. I2C controller core, 2001-2018 [S.l.]: Disponível em: <https://opencores.org/projects/i2c?do=projects&download=i2c> . Acesso em: 7 ago. 2023.
- [5] HERVEILLE, Richard. I²C - Master Core Specification, 2003 [S.l.]: Disponível em: https://opencores.org/websvn/filedetails?repname=i2c&path=%2Fi2c%2Ftrunk%2Fdoc%2Fi2c_specs.pdf . Acesso em: 7 ago. 2023.