

Descrição geral do sistema

Desenvolver um sistema composto por duas máquinas de estados, que implemente um sistema de comunicação padrão I²C, no qual uma das máquinas é Master na comunicação e a outra é Slave. O objetivo é “escrever” um número binário nas chaves deslizantes. A máquina Master envia este número através do barramento I²C e a máquina Slave envia o número recebido para o display de sete segmentos.

A Figura 1 apresenta o formato dos sinais de um barramento I²C. Ele é composto por dois sinais elétricos apenas: um sinal de relógio **SCL** e **SDA**, um sinal para envio e recepção de dados SDA, e pode trabalhar nas taxas de 100 Kbit/segundo ou 400 Kbit/segundo (neste trabalho será apenas de 100 Kbps). Também na Figura 1, é mostrado um barramento I²C com um processador atuando como Master e três dispositivos Slave (A, B e C).

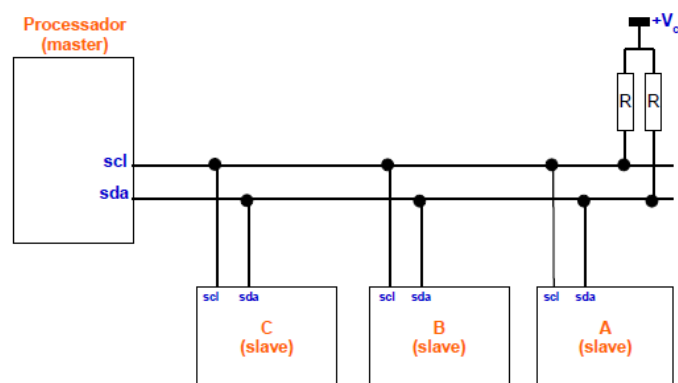


Figura 1: Barramento I²C com três dispositivos escravos (Slave).

Os dados são enviados e recebidos pelo sinal SDA, na forma de bytes. Isso implica em dizer que os dispositivos conectados à uma rede I²C devem deixar este sinal em alta impedância, para não prejudicar o envio de dados pelo Master, por exemplo. Em repouso os sinais SCL e SDA ficam em nível lógico um. Por este motivo são colocados dois resistores de pull-up (para a alimentação +VCC), de modo a garantir que em alta impedância os sinais estejam neste nível lógico.

A Figura 2 mostra a transferência de um Byte no barramento I²C. Existe duas áreas na cor cinza naquela figura, no lado mais a esquerda e no lado mais a direita da figura. O sinal mais à esquerda recebe o nome de START e o sinal mais a direita recebe o nome de STOP. Estes pontos indicam o início e o final de uma transmissão, respectivamente, para todos os dispositivos conectados na rede I²C. O START acontece quando o Master deixa o sinal SDA em nível lógico zero e faz a transição de nível lógico do sinal SCL de um para zero. O final da transmissão (STOP) é indicado pelo processador quando ele deixa o sinal SCL em nível lógico um e faz a transição do sinal SDA de zero para um, como mostra a Figura 2.

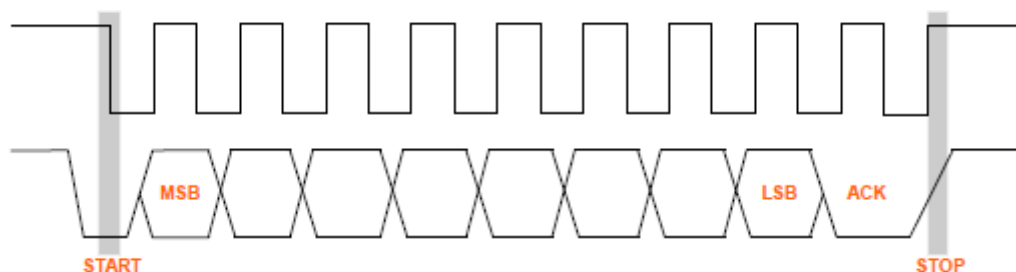


Figura 2: Sinais SCL e SDA do barramento I²C.

A Figura 3 mostra três exemplos de como é feita a transferência de dados no barramento I²C. Cada Slave conectado na rede tem um endereço de 7 bits, o que permite ter um total de até 128 dispositivos. O bit menos significativo do endereço é usado para informar aos dispositivos Slave se a operação é de escrita (0) ou de leitura (1). Note ainda que nos três exemplos cada transmissão iniciou com um START (S na figura) e finalizou com um STOP (P na figura). No primeiro exemplo da figura, o Master escreve dois para o Slave endereçado. No segundo, o Master requisita uma leitura do Slave endereçado. No terceiro, o Master escreve um byte para um dispositivo Slave, e em seguida envia outro START para iniciar a leitura de um Slave.

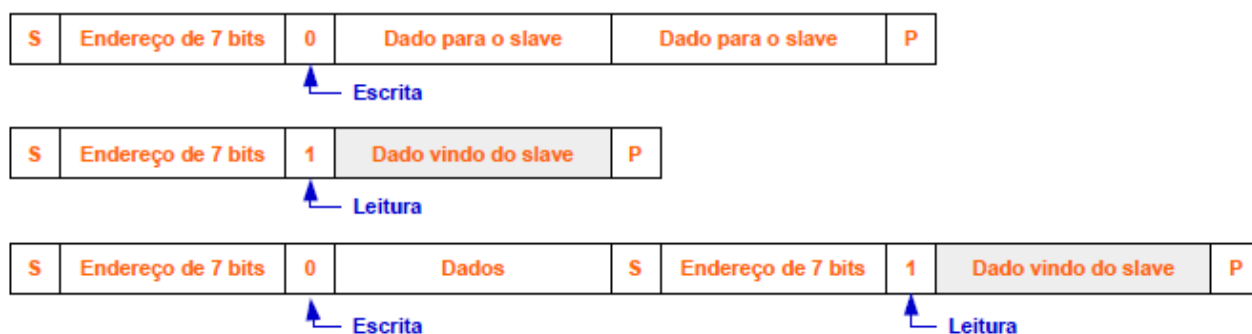


Figura 3: Exemplos de comunicação num barramento I²C.

Abaixo é mostrado na Figura 4 o kit de desenvolvimento DE10-Lite a ser utilizado no trabalho. O barramento I²C deverá operar na frequência de 100 Kbps.

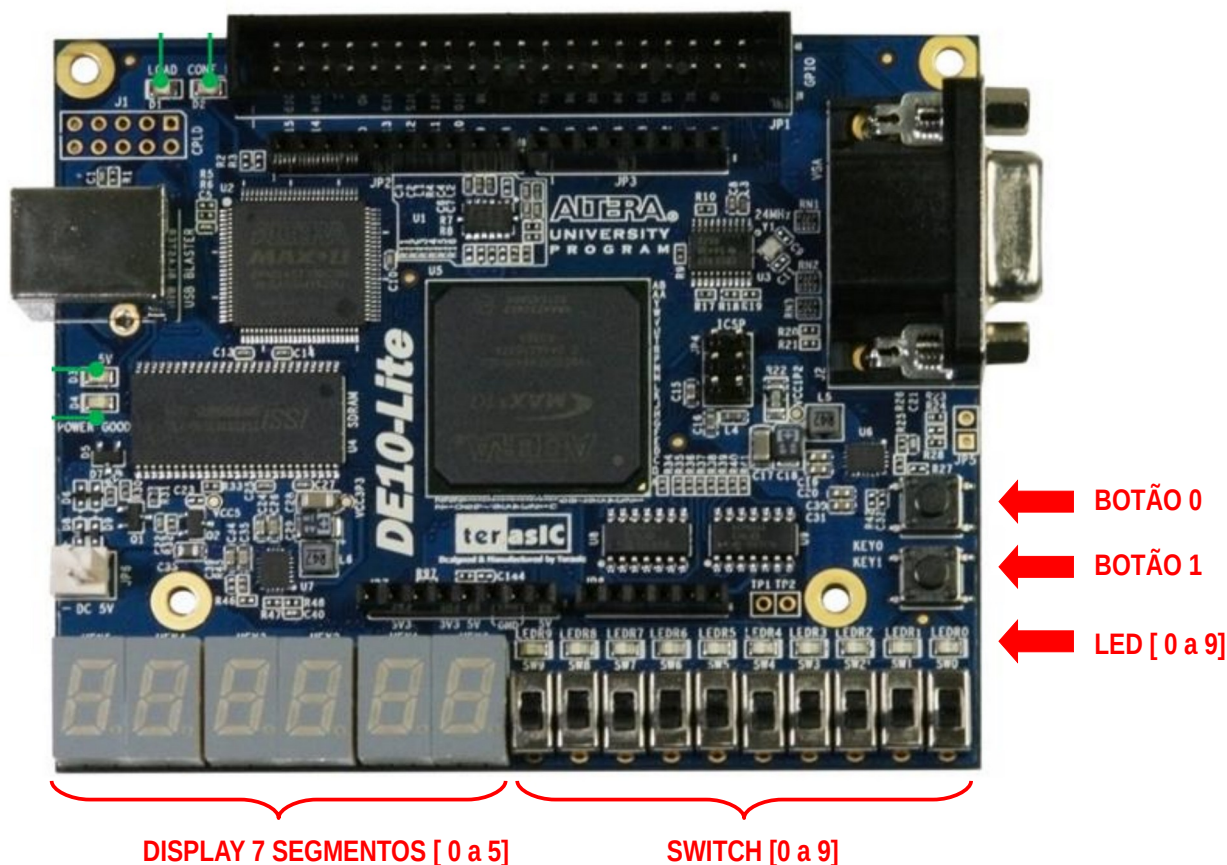


Figura 4: Kit de desenvolvimento DE0 usado no desenvolvimento do trabalho.

O projeto deverá consistir de dois circuitos, sendo uma Master e outro Slave. O Master deve enviar um dado escrito nas chaves SW0-SW7 para a máquina Slave. O botão BOTÃO 1 servirá para indicar que o dado (previamente definido nos SWITCHs de 0 a 7) deve ser transmitido pelo Master através da serial I²C. O BOTÃO 0 será usado como Reset geral do sistema. O dispositivo Slave deve mostrar o valor recebido em um display de sete segmentos. Além disso, a máquina que controla o dispositivo Slave deverá retornar o valor lido previamente ao receber um comando de leitura. Neste caso o Master irá escrever no display 7 segmentos o valor recebido nesta leitura. A Figura 5 mostra a macro arquitetura do projeto, utilizando diagrama de blocos funcionais.

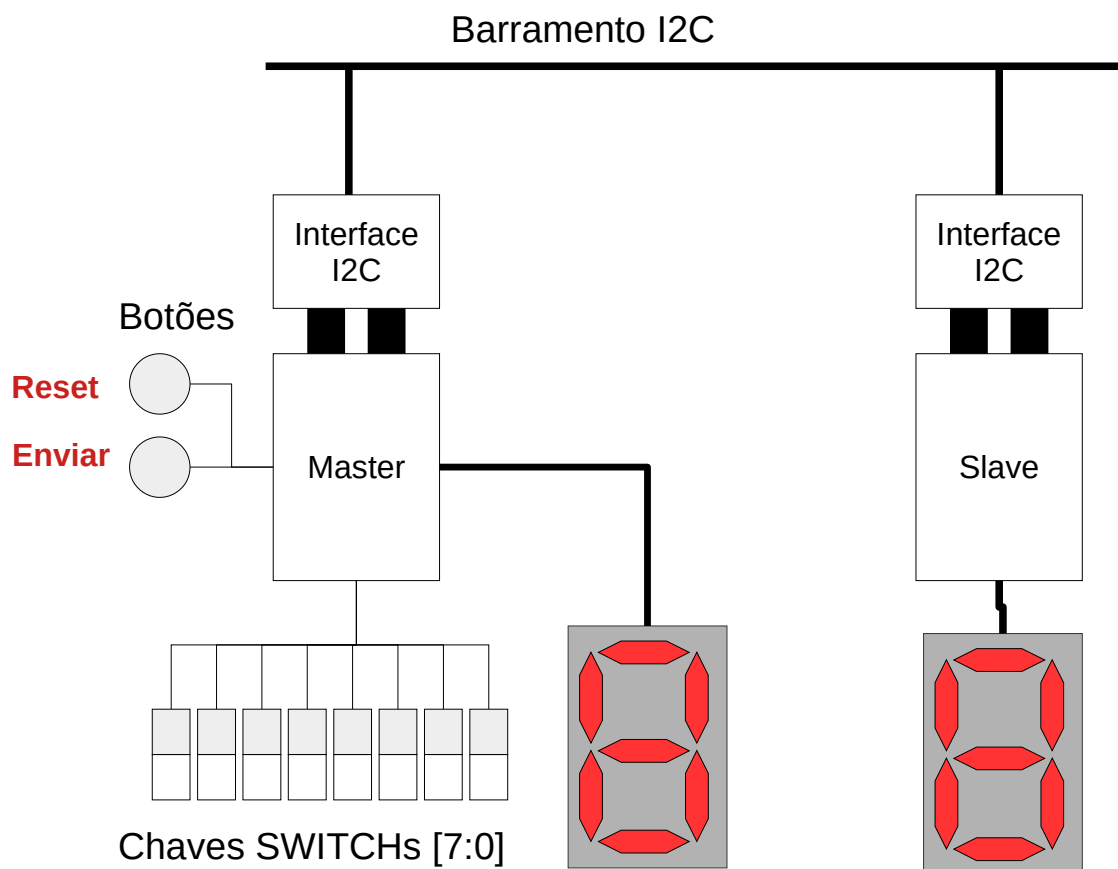


Figura 5: Macro arquitetura do projeto I2C.

Cr terios de avalia  o

- 2 pontos pela entrega de um diagrama em blocos funcionais (desde que esteja coerente com o c digo VHDL implementado).
- 2 pontos pela entrega de TestBenchs (funcionando) para cada componente projetado.
- 6 pontos a serem dados durante a apresenta  o e defesa do projeto.