

# PCS3335 - Laboratório Digital A - Experiência 5

por Bruno de Carvalho Albertini

28/03/2024

A experiência 5 é uma pausa no desenvolvimento do SHA256 para introduzirmos uma interface serial.

## Introdução

Você deve ter um componente que, em 64 ciclos, gera um *hash* equivalente ao algoritmo padronizado SHA256. Até o momento, usamos as chaves para compor um valor de sintético de 512b. No entanto, os valores para um *hash* real não são a replicação de um valor de 8b e sim os 512b para os quais deseja-se calcular o *hash*. Ao invés de usarmos 512 chaves, o que é inviável, vamos transmitir e receber os 512b do computador para a placa. Esta experiência envolve somente a transmissão serial, porém sugerimos que veja o próximo enunciado pois, no caso da serial, o desafio está na recepção e não na transmissão. Recomendamos que resolva as duas experiências concomitantemente.

## Experiência 5

Seu objetivo nesta experiência é enviar 8b pela serial e recebê-los no computador.

### Planejamento

A transmissão serial é relativamente simples, mas envolve alguns cuidados. Na Figura 1 podemos ver a transmissão do valor em binário 01010000. Observe a forma de onda com cuidado antes de prosseguir.

No modelo que usaremos neste oferecimento, há algumas definições que precisamos seguir. A linha de transmissão serial é diferente da linha de recepção, portanto há um fio exclusivo para transmissão e outro para a recepção, sem necessidade de multiplexação ou de lidar com alta impedância.

A frequência de transmissão é fixa e o *clock* disponível na placa é de 50MHz. Para a serial, usaremos um divisor de *clock*, o que pode ser feito com o contador que você desenvolveu na experiência 4. Você pode usar qualquer frequência de transmissão que quiser, porém recomendamos 4800 bits por segundo. Você deve dividir o *clock* da placa e gerar o *clock* serial, ou seja, um sinal periódico, com *duty cycle* de 50%,

[3 pontos]

*tri-state*

Se você está avançado, use um PLL.

Você sabe a diferença de *baud rate* e bits/s?

que seja na frequência desejada (e.g. 4800Hz). Caso opte por outra frequência, tente usar uma que seja padronizada pois as interfaces seriais com o computador são padronizadas e, portanto, esperam uma frequência padronizada.

O protocolo serial usado neste experimento consiste no seguinte: a linha em repouso fica sempre em 1. Antes de começar há o que chamamos de *start bit*, ou seja, a linha vai para zero por um ciclo da serial (um período de *clock* serial). Em seguida, transmite-se os 8 bits, começando pelo menos significativo até o mais significativo. Por último, transmite-se a paridade ímpar dos 8 bits transmitidos, seguida de dois *stop bits*. Os *stop bits* são uma garantia de que a linha ficará no estado de repouso por pelo menos dois ciclos de *clock* serial antes da próxima transmissão.

Você pode usar mais de 2 *stop bits* quando for para a experiência, mas nunca menos.

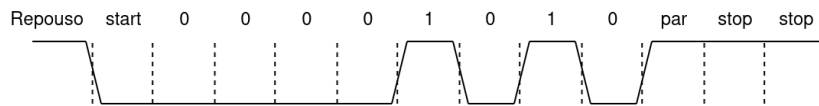


Figura 1: Transmissão serial.

Na Figura 1 podemos ver a transmissão da palavra 01010000. A linha tracejada corresponde aos ciclos de *clock* da serial (um ciclo completo começando com a borda de subida e terminando imediatamente antes da borda de subida do próximo ciclo). Começa-se com o *start bit*, seguida dos 8b do LSB para o MSB, seguida da paridade e dos dois *stop bits*.

### Planejamento

```
entity serial_out is
  generic (
    POLARITY: boolean := TRUE;
    WIDTH: natural := 7;
    PARITY: natural := 1;
    STOP_BITS: natural := 1
  );
  port (
    clock, reset, tx_go: in bit;
    tx_done: out bit;
    data: in bit_vector(WIDTH-1 downto 0);
    serial_o: out bit
  );
end serial_out;
```

Figura 2: Entidade para a experiência 5

Na listagem da Figura 2, e podemos ver a entidade do módulo de saída serial. Os parâmetros são a polaridade (TRUE significa que o

Neste oferecimento sempre usaremos transmissão 802 em polaridade TRUE.

*start bit* é baixo, a transmissão de um bit um é alta e de um bit zero é baixa, FALSE significa o contrário), WIDTH é o tamanho da palavra em bits, PARITY (0 é par, 1 é ímpar), e o número de *stop bits*.

As entradas e saídas do módulo são o *clock* serial (e.g. 4800Hz), um sinal de *reset* ativo alto, um sinal que indica o início da transmissão (*tx\_go*) e um sinal que indica o fim da transmissão (*tx\_done*). Há ainda uma entrada paralela de dados e uma saída serial de dados. Ao levantar o sinal de início de transmissão, o módulo amostra a entrada paralela de dados, abaixa o sinal de fim de transmissão, realiza a transmissão serial dos dados amostrados paralelamente conforme descrita acima (*start*, dados, paridade e *stops*), colocando a saída serial na saída de dados homônima, e levanta o sinal de fim de transmissão quando o ciclo de transmissão terminar, mantendo-o ativo até que outro início de transmissão seja ordenado ou na presença de um *reset*.

Você deve planejar e codificar este módulo, e enviá-lo para o juiz.

### Preparação para montagem

Antes de ir para o laboratório, você precisa preparar sua pinagem adequadamente. Os 8b de entrada de dados devem ser ligados nas chaves. O *reset* e o *tx\_go* devem ser ligados na GPIO, assim como a saída serial. O sinalizador de fim de transmissão deve ser ligado em um LED. Um dos *display* deve mostrar o caractere ASCII correspondente a entrada paralela, que também deve ser replicada nos LEDs. Os demais *displays* devem permanecer apagados.

Veja o manual da placa FPGA no site do laboratório.

### Execução

Você fará ligações físicas na placa, o que exige certo cuidado. É possível ligar esta serial em um computador usando diretamente a interface serial (ou um conversor USB-serial), mas por segurança, usaremos um equipamento de depuração chamado Analog Discovery. Pare neste momento e veja os manuais do dispositivo para Analog Discovery e Analog Discovery 2.

Clique nos links para abrir o manual.

Você deve ligar ao menos três pinos digitais e o sinal de referência (terra) na GPIO da placa. O primeiro é a saída serial do seu módulo e os demais são as entradas de *reset* e *tx\_go*. Para a saída serial, use uma aba do software da Analog Discovery (Waveforms) para lógica digital, configurando a entrada como uma UART com os parâmetros corretos. Na Figura 3 podemos ver uma captura desta tela para a transmissão contínua do caractere ASCII "B".

As entradas devem ser ligadas na aba *Static I/O* e configuradas como botões.

Esta experiência não tem desafio.

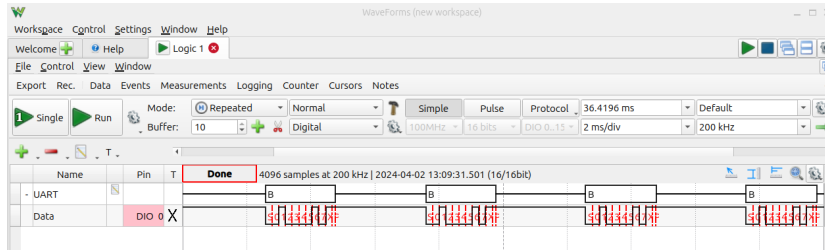


Figura 3: Tela da Analog Discovery com a serial.

## Checklist

- ☐ Atividades pré-aula:
  - ☐ Implementar o módulo serial em VHDL
  - ☐ Implementar um *testbench*
  - ☐ Enviar sua solução para o juiz (individual)
  - ☐ Planejar os testes a serem realizados no laboratório (dupla)
- ☐ Opcionalmente montar o projeto no Quartus
- ☐ Atividades durante a aula:
  - ☐ Montar o projeto no Quartus (se não montou)
  - ☐ Atribuir os pinos corretamente (se não atribuiu)
  - ☐ Sintetizar o projeto e configurar a placa
  - ☐ Ligar a Analog Discovery
  - ☐ Executar os testes planejados e corrigir eventuais erros
  - ☐ Escrever o relatório da experiência (dupla)

O teste adequado garante que sua solução funciona.

Todos os planejamentos devem ser enviados até 24h antes da sua próxima aula.

Atenção: não ligue a Analog Discovery antes de configurar a placa.  
Em caso de dúvidas, chame seu professor ou um técnico.