

PCS3335 - Laboratório Digital A - Experiência 6

por Bruno de Carvalho Albertini

28/03/2024

A experiência 6 continua a interface serial, adicionando a recepção.

Introdução

A recepção serial é um pouco mais complexa pois é assíncrona, ou seja, o receptor não possui um sinal em fase com o *clock* que foi utilizado pelo transmissor.

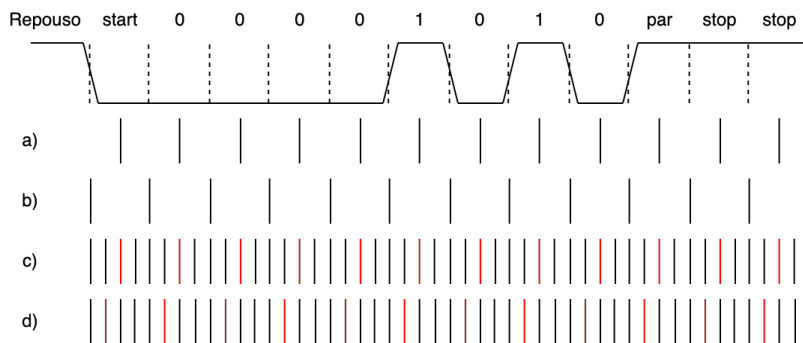


Figura 1: Recepção serial.

Na Figura 1 podemos ver o fluxo de uma recepção serial. Na amostragem mostrada no item (a), todos os bits foram amostrados exatamente no meio do ciclo de *clock* serial. Essa seria uma amostragem perfeita, mas inviável sem o sincronismo de *clock*. A amostragem (b) é o oposto, ou seja, muito ruim, pois todas as amostragens foram feitas no início (ou no final) do ciclo de *clock*. Isso torna impossível determinar se o valor amostrado é o correto pois não respeitamos os tempos de *setup/hold* de possíveis elementos de memória envolvidos na amostragem.

As amostragens (c) e (d) são as factíveis pois são feitas em $3/4$ e $1/4$ do ciclo de *clock*, respectivamente, garantindo com certa segurança que não amostramos muito perto das bordas. No entanto, para serem viáveis, o seu receptor precisa trabalhar em uma frequência quatro vezes maior que a frequência de transmissão esperada.

Experiência 5

Seu objetivo nesta experiência é receber os bits enviados pelo computador e mostrá-los nos *displays*.

Planejamento

Refaça o seu contador (ou instancie outro) para gerar um sinal de *clock* com quatro vezes a frequência esperada de transmissão. O sinal deve ser derivado do *clock* de 50MHz presente na placa.

Se seguiu a dica de usar 4800bps, será de 19200bps

```
entity serial_in is
  generic(
    POLARITY: boolean := TRUE;
    WIDTH: natural := 8;
    PARITY: natural := 1;
    CLOCK_MUL: positive := 4
  );
  port (
    clock, reset, start, serial_data: in bit;
    done, parity_bit: out bit;
    parallel_data: out bit_vector(WIDTH-1 downto 0)
  );
end serial_in;
```

Figura 2: Entidade para a experiência 5

Na listagem da Figura 2, podemos ver a entidade do módulo de recepção serial. Os parâmetros são similares ao transmissor: polaridade (TRUE significa que os bits de controle, incluindo o repouso, são altos e a transmissão de um bit 1 é também alta, FALSE significa o contrário), WIDTH é o tamanho da palavra em bits, PARITY (0 é par, 1 é ímpar), e multiplicador de *clock*. Para este oferecimento, usaremos sempre polaridade positiva (TRUE), 802 (8 bits de dados, paridade ímpar e dois *stop* bits).

As entradas e saídas do módulo são o *clock* serial (e.g. 19200Hz), um sinal de *reset* ativo alto, um sinal que indica o início da recepção (*start*) e um sinal que indica o fim da recepção (*done*). Há ainda uma saída paralela de dados, uma entrada serial de dados e uma saída com a paridade recebida. Ao levantar o sinal de início de recepção, o módulo aguarda o *start* bit na linha serial, recebe os 8 bits, a paridade e os *stop* bits, quando levanta o sinal de *done* e permanece assim até um novo sinal de *start* ou *reset*. Os dados recebidos sempre estão disponíveis na saída paralela, porém só corresponder ao dado recebido quando o *done* é alto.

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

Preparação para montagem

Faça a mesma montagem da experiência 5. Adicione o receptor serial desta experiência. Ligue a entrada serial em um pino próximo fisicamente do pino da saída serial que escolheu para a experiência 5. O *reset* é compartilhado com transmissor, porém o *start* é um novo pino na GPIO. O *done* deve ir para um LED da placa e o último caractere recebido (saída paralela do receptor) deve ser mostrado em um *display* de sete segmentos. Todos os demais mapeamentos de entrada e saída da experiência 5 devem ser mantidos.

Execução

A execução é similar à experiência 5, porém desta vez você deve enviar um caractere a partir da Analog Discovery, receber através do seu receptor serial e mostrá-lo no *display*. Você pode usar um *jumper* entre a saída serial do transmissor (experiência 5) e a entrada serial do receptor (esta experiência), no entanto antes de fazer esta ligação consulte o seu professor.

Esta experiência não tem desafio.

Dicas

Use um registrador de deslocamento para a entrada serial e uma máquina de estados para a recepção. O receptor deve trabalhar em uma frequência maior que a recepção. Apesar de estar livre para utilizar outras soluções, recomendamos a solução com quatro vezes, amostrado em 3/4 do ciclo de *clock*. Para isso, use um contador que, a partir do recebimento do *start* bit, conte quatro ciclos de *clock* repetidamente, amostrando o sinal no terceiro elemento de contagem. Não se esqueça de amostrar o bit de paridade e colocá-la na saída. Os *stop* bits não precisam ser amostrados (nem o *start* bit), mas devem ser levados em consideração para a sinalização do protocolo (*start* e *done*).

O *reset* não é garantido a cada recepção e o *start* pode ser mantido em alto para permitir recepções consecutivas. A saída paralela não é amostrada com o *done* baixo, mas deve ser a saída correta quando o *done* estiver alto.

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)

O teste adequado garante que sua solução funciona.

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

☐ 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