

Serial Roulette Controller

5º Relatório do Projeto *Roulette Game*

Trabalho realizado por:

Gustavo Costa | Nº 52808

Ian Frunze | Nº 52867

Rafael Pereira | Nº 52880

Turma: **LEIC24D**

Docentes: **David Velez e Rui Duarte**

Licenciatura em Engenharia Informática e de Computadores

Laboratório de Informática e de Computadores (LIC)

2024 / 2025 - Semestre de Verão

Conteúdo

1	Introdução	3
2	Objetivo do Módulo SRC	4
3	Estrutura Interna	5
3.1	<i>Serial Receiver</i>	5
3.2	<i>Roulette Dispatcher</i>	5
3.3	Interligação entre os blocos	6
4	Protocolo de Comunicação	7
5	Testes e Simulação	8
5.1	Testes ao <i>Serial Receiver</i> (SRC)	8
5.2	Testes ao <i>Roulette Dispatcher</i>	8
6	Conclusão	10

Lista de Figuras

1	Módulo <i>Serial Roulette Controller</i>	5
2	Protocolo de comunicação com o módulo <i>Serial Roulette Controller</i> (SRC)	7
3	Módulo <i>Serial Roulette Controller</i> <i>simulação</i>	8
4	Módulo <i>Serial Roulette Controller</i> <i>simulação</i>	9

Lista de Códigos

1 Introdução

No contexto do projeto *Roulette Game*, o módulo *Serial Roulette Controller* (SRC) assume um papel crítico na camada de *hardware*, atuando como intermediário entre a lógica do jogo (implementada em *software*) e o *Roulette Display* — o componente visual central que exibe o sorteio da roleta, resultados numéricos, animações dinâmicas e créditos ganhos pelo jogador. Integrado numa arquitetura híbrida (*hardware-software*), o SRC é responsável por garantir que os comandos gerados pelo módulo *Control* (escrito em *Kotlin*) sejam traduzidos em ações visuais precisas e sincronizadas no mostrador físico da roleta.

O sistema global do jogo depende de uma comunicação eficiente entre módulos, e o SRC destaca-se pela sua capacidade de operar com um protocolo de comunicação série. Este protocolo permite a transmissão de mensagens estruturadas em 8 bits de dados e 1 bit de paridade ímpar, onde os primeiros 3 bits definem o comando (ex.: atualizar valor, iniciar animação, desativar *display*) e os restantes 5 bits especificam os dados associados (ex.: número sorteado, créditos). A implementação deste protocolo assegura não apenas a modularidade do sistema, mas também a robustez contra erros de transmissão, graças à verificação de paridade.

O SRC é composto por dois blocos principais:

- *Serial Receiver*: Responsável por receber e validar tramas, sincronizando os dados com o sinal de clock (SCLK) e detetando possíveis erros, via paridade.
- *Roulette Dispatcher*: Encarrega-se de interpretar os comandos recebidos e acionar os circuitos do *Roulette Display* como controlar *displays* de 7 segmentos, ativar LEDs para animações, etc.

Não só pelo nome mas também pelas suas funcionalidades, este módulo assemelha-se muito ao módulo *Serial LCD Controller* (SLCDC). Enquanto o SLCDC foca-se na exibição estática de texto no LCD (com tramas de 5 bits + paridade), o SRC lida com visualizações dinâmicas e interativas no *Roulette Display*, que consequentemente suporta comandos mais complexos (3 bits de comando + 5 bits de dados) e animações em tempo real.

Além disso, o SRC requer uma coordenação mais exigente com o *Control* (*software*) devido à natureza dinâmica do sorteio, enquanto o SLCDC opera com atualizações menos frequentes e mais previsíveis.

2 Objetivo do Módulo SRC

O principal objetivo do módulo *Serial Roulette Controller* (SRC) é servir como interface de comunicação entre o *software* de controlo e o mostrador da roleta, recebendo comandos em formato série e traduzindo-os em ações visuais no *Roulette Display*.

Este mostrador exibe os resultados do sorteio de forma clara para o jogador, sendo capaz de realizar algumas animações perante os *displays* de 7 segmentos, mostrar números sorteados, apresentar os créditos ganhos e até desativar ou limpar o visor conforme necessário.

De forma mais técnica, o SRC atua como um decodificador e expedidor de mensagens série enviadas pelo *software*, controlando diretamente os circuitos responsáveis pela visualização dos dados no mostrador físico da roleta (*display* da DE10-Lite *Expansion Board*).

O objetivo central do módulo *Serial Roulette Controller* (SRC) é funcionar como um tradutor entre o *software* de controlo e o *hardware* do *Roulette Display*, assegurando que as instruções do jogo se materializem em ações visuais coerentes e atraentes para o jogador. Mais especificamente, o SRC visa:

- Receber e processar comandos série enviados pelo módulo *Control*, convertendo-os em sinais compatíveis com o *Roulette Display*.
- Garantir sincronização temporal, assegurando que animações como efeitos de rotação e atualizações de valores ocorram sem atrasos perceptíveis, essencial para a imersão do jogador.
- Implementar mecanismos de erro, como a verificação de paridade ímpar, para evitar que dados corrompidos afetem algum elemento que influencie a integridade visual do jogo.
- Suportar modos operacionais, incluindo: O Modo Jogo, onde exibe os números sorteados e créditos ganhos; O Modo Manutenção, como teste de funcionalidades do *display*, como por exemplo, ativar todos os segmentos; O Modo Desligar, que limita-se a limpar o visor, e por consequência, a desativar todos os LEDs. Todas estes modos irão ser implementados posteriormente.

Tecnicamente, o SRC atua como um controlador de baixo nível, que descreve comandos (ex.: "mostrar o número 5") em sinais específicos para os circuitos da DE10-Lite *Expansion Board*. A sua eficiência é fundamental para a experiência do utilizador, pois determina não apenas a precisão das informações exibidas, mas também o impacto visual do jogo — um ponto decisivo para a credibilidade do sistema como um todo.

3 Estrutura Interna

Como já mencionado anteriormente, o módulo SRC é composto por dois blocos principais: *Serial Receiver* e *Roulette Dispatcher*.

Cada bloco possui uma função distinta, mas interligada: o *Serial Receiver* trata da receção, organização e validação da trama de dados. Já *Roulette Dispatcher* interpreta os comandos válidos e atua diretamente sobre o *display*.

Segue-se um esquema completo do módulo SRC:

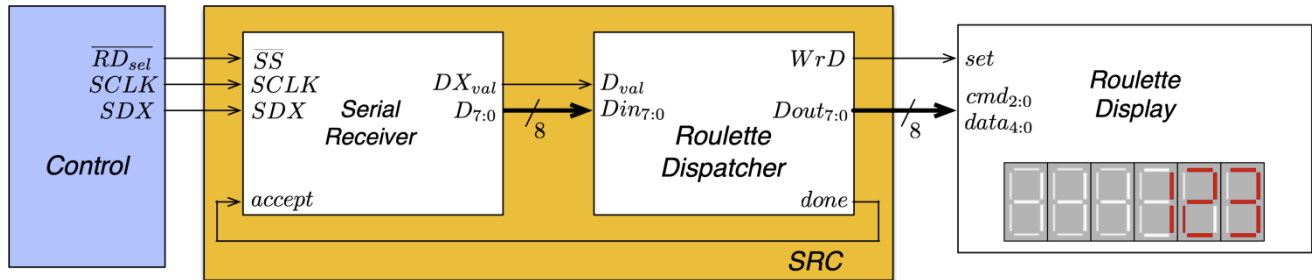


Figura 1: Módulo *Serial Roulette Controller*

3.1 *Serial Receiver*

O *Serial Receiver* é o primeiro bloco do módulo SRC e é responsável por receber a trama de bits em série, enviada pelo *software* (*Control*), validar a integridade dessa trama e disponibilizá-la para o bloco seguinte.

Este bloco transforma sinais de comunicação em série (bit a bit) num conjunto de dados legíveis e prontos para processamento interno. Ele é particularmente sensível ao sincronismo com o sinal de clock (SCLK) e à deteção do início da transmissão (através da transição no sinal *RDsel* [*active-low*]).

Este bloco inclui:

- Lógica de deteção de início da trama (transição em *LDsel*).
- Registo de deslocamento (*Shift Register*) para acumular os bits.
- Contador de bits (*Counter*) para garantir que a trama completa é recebida. Diferente do módulo SLCDC, agora a trama é de 9 bits.
- Bloco de verificação de paridade (*Parity Check*), que assegura a validade da transmissão antes de a processar.

A importância deste bloco reside no facto de garantir a fiabilidade da comunicação e evitar que comandos incorretos ou incompletos cheguem ao *Dispatcher*, o que poderia comprometer a coerência visual da interface para com o jogador.

Este bloco tem como base uma máquina de estados (ASM) idêntica à do *Serial Receiver* do módulo *Serial LCD Controller*.

3.2 *Roulette Dispatcher*

O *Roulette Dispatcher* é o segundo e último bloco do módulo SRC. A sua função é executar os comandos recebidos pelo *Serial Receiver* e controlar diretamente o *display*.

Depois de uma trama válida ser sinalizada com *Dval* pelo *Serial Receiver*, o *Dispatcher* extrai o comando e os dados, interpreta o significado da instrução e aciona o *hardware* do *display* conforme o comando recebido.

As ações que este bloco pode tomar incluem:

- Mostrar um número ou letra no display (*setValue*).

- Iniciar ou parar uma animação de sorteio.
- Ligar ou desligar o mostrador, limpar o conteúdo do visor, entre outras.

Este bloco traduz comandos lógicos em sinais físicos de controlo, ativando os respetivos circuitos no *hardware* do *Roulette Game*. Ao terminar a execução de cada comando, o *Dispatcher* ativa o sinal **done** para indicar ao *Serial Receiver* que está pronto para tratar uma nova trama.

É neste bloco que se materializa a interface visual com o jogador, sendo assim responsável pela parte mais visível da resposta do sistema no jogo da Roleta.

3.3 Interligação entre os blocos

Estes dois blocos estão fortemente conectados por sinais internos, com destaque para:

- D_{val} — sinal enviado do *Serial Receiver* para o *Dispatcher* a indicar que os dados são válidos e prontos para interpretação;
- **done** — sinal enviado pelo *Dispatcher* para o *Serial Receiver*, a confirmar que o comando anterior foi tratado e o sistema está pronto para receber a próxima trama.

Este mecanismo de sincronização garante um fluxo de dados controlado, sem sobrecargas ou colisões, o que assegura que cada comando é processado, completamente, antes de se iniciar o seguinte.

4 Protocolo de Comunicação

A comunicação com o SRC é feita por uma interface série, utilizando um protocolo de 9 bits, conforme a figura que se segue:

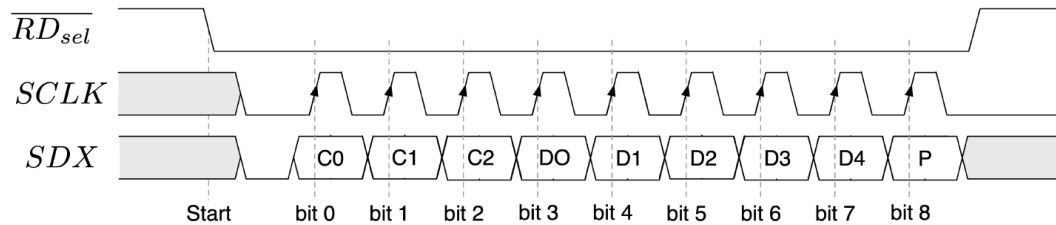


Figura 2: Protocolo de comunicação com o módulo *Serial Roulette Controller* (SRC)

A estrutura da trama é a seguinte:

- Bits 0 a 2 ($cmd_{2:0}$): Definem o comando a executar sobre o display.
- Bits 3 a 7 ($data_{4:0}$): Representam o dado associado ao comando, como o valor a mostrar.
- Bit 8: Bit de paridade ímpar, usado para validar a integridade da trama.

Os comandos possíveis estão organizados numa tabela (Tabela 1 do enunciado), onde:

- O comando $cmd = 000$ significa atualizar o dígito 0.
- O comando $cmd = 001$ significa atualizar o dígito 1.
- O comando $cmd = 101$ significa atualizar o dígito 5.
- O comando $cmd = 110$ significa atualizar o *display*.
- O comando $cmd = 111$ com o último bit - $data[0]$ - a '0' significa ligar o *display*.
- O comando $cmd = 111$ com o último bit - $data[0]$ - a '1' significa desligar o *display*.

A transmissão inicia-se com uma transição em RD_{sel} , seguida da transmissão dos bits sincronizados com $SCLK$.

Este protocolo permite ao *software* enviar instruções complexas com apenas alguns bits, mantendo a linha de comunicação rápida.

5 Testes e Simulação

Neste capítulo são relatados os testes realizados aos dois sub-blocos principais do módulo *Serial Roulette Controller* (SRC): o *Serial Receiver* e o *Roulette Dispatcher*. Os testes tiveram como principal objetivo garantir a fiabilidade da comunicação série entre *software* e o mostrador da rolet (*display*).

5.1 Testes ao *Serial Receiver* (SRC)

O *Serial Receiver* é responsável pela receção de tramas enviadas em série pelo *software*, contendo comandos e dados destinados ao mostrador da roleta. No contexto do módulo SRC, este bloco recebe uma trama composta por 8 bits de dados e 1 bit de paridade ímpar, totalizando 9 bits por transmissão, sendo a principal diferença para o bloco *Serial Receiver* pertencente ao módulo *Serial LCD Controller*, onde a trama é de apenas 6 bits.

Como objetivos do teste temos:

- Verificar a correta sincronização com o *clock* (SCLK).
- Confirmar a correta reconstrução da trama em paralelo, a partir dos dados em série em SDX.
- Validar a verificação de paridade ímpar, assegurando que apenas tramas corretas são aceites.
- Garantir a emissão do sinal D_{val} quando uma trama válida é completamente recebida.

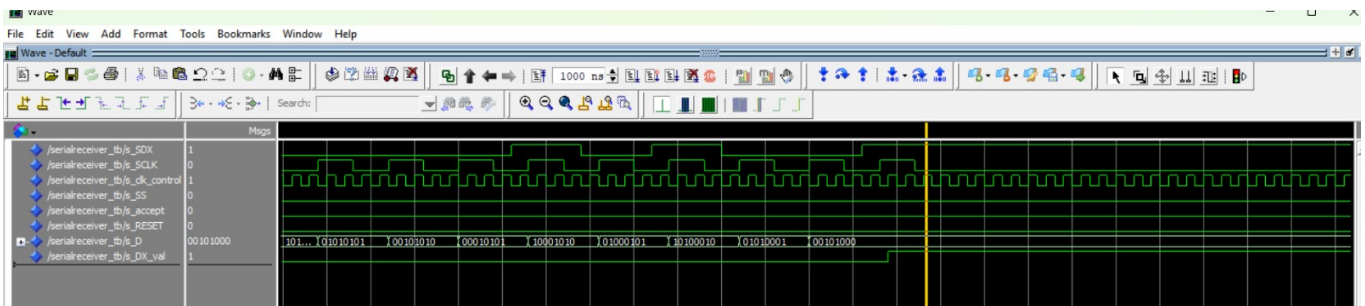


Figura 3: Módulo *Serial Roulette Controller* simulação

Ao observar o resultado da simulação, foi possível verificar que o sinal SS (*start*) ativa corretamente o início da receção, com uma transição de '1' para '0'. A cada flanco ascendente de SCLK, um novo bit é introduzido no *Shift Register*, até que os 9 bits sejam recebidos.

A paridade é validada de forma eficaz - o sinal D_{val} só é ativado quando o número total de bits a '1' na trama é ímpar.

Em caso de erro de paridade, a trama é ignorada, e D_{val} não é ativado, o que garante que não haja passagem de tramas não desejadas na comunicação.

Este módulo retorna automaticamente ao estado de espera após o sinal *accept* ser recebido.

Concluindo, o *Serial Receiver* demonstrou um comportamento estável e fiável durante a simulação. A receção, verificação e disponibilização das tramas estão em conformidade com o protocolo especificado, assegurando que apenas dados válidos são entregues ao bloco seguinte.

5.2 Testes ao *Roulette Dispatcher*

O *Roulette Dispatcher* é o bloco responsável por interpretar e executar os comandos recebidos a partir de uma trama válida, atuando diretamente sobre o *display*. Utiliza os 3 bits de comando e os 5 bits de dados, provenientes do *Serial Receiver* para decidir as operações corretas.

Os objetivos deste teste são:

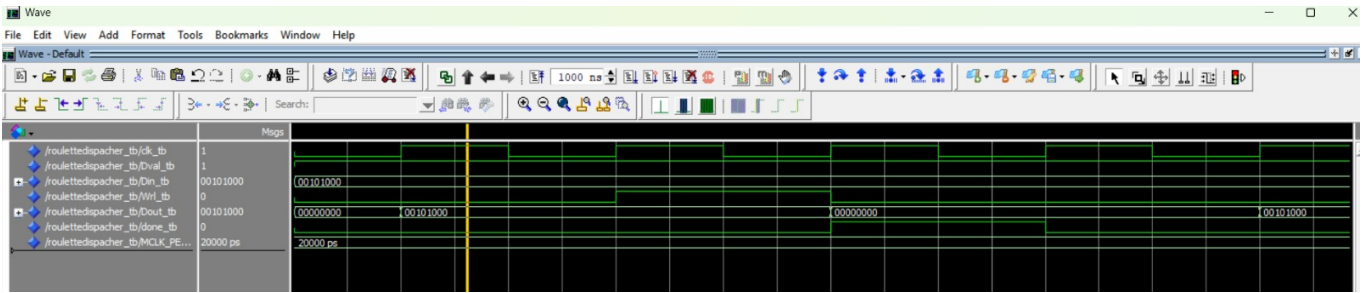


Figura 4: Módulo *Serial Roulette Controller* simulação

- Garantir que comandos recebidos são corretamente decodificados e aplicados ao mostrador.
- Confirmar a correta ativação do sinal **done** após o comando ser executado, libertando o canal de receção.

O *Roulette Dispatcher* mostrou-se eficaz na sua função de decodificação e atuação sobre o *display*. Comportou-se conforme esperado para diferentes tipos de comando, e a sua integração com o *Serial Receiver* revelou-se funcional e estável. A simulação confirmou que o sub-bloco cumpre os requisitos estabelecidos para a sua função no sistema.

6 Conclusão

O *Serial Roulette Controller* (SRC) é um componente essencial do sistema do jogo da Roleta. A sua arquitetura modular, composta pelo *Serial Receiver* e o *Roulette Dispatcher*, permite a receção, validação e execução de comandos série de forma eficaz. Este módulo é o elo entre o software de controlo e a saída visual, que proporciona uma experiência fluida e responsiva ao utilizador.

A utilização de um protocolo de comunicação série com verificação de paridade, aliada à separação de responsabilidades entre receção e execução de comandos, resulta num sistema altamente fiável.

Sem o SRC, o mostrador da roleta (*display*) não teria capacidade de receber e interpretar os resultados do jogo, tornando este módulo absolutamente indispensável para a apresentação visual do jogo e a interação com o jogador.