

## Módulos PBL

No escopo deste treinamento, os pesquisadores deverão desenvolver um conjunto de soluções Hardware/Software para um mesmo problema – um jogo Genius. As soluções para o problema proposto serão utilizadas para guiar o processor de aprendizagem (PBL - *Problem-Based Learning*) dos pesquisadores no âmbito do treinamento. Nesse contexto, o mesmo problema deverá ser solucionado através de diferentes abordagens.

As estratégias solicitadas são: PBL01, desenvolvimento de uma solução em RTL com FSM *hardcoded*; PBL02, implementação do mesmo jogo em linguagem de máquina (assembly) RISC-V execução em simulador; PBL03, desenvolver verificar um processador RISC-V simplificado para rodar o mesmo jogo simulado anteriormente; PBL04, sintetizar e realizar a análise estática de tempo (STA) para o RTL e prototipar os resultados em FPGA.

Ao final de cada módulo PBL, cada participante apresentará os resultados das soluções para o problema em 10 minutos.

Obs.: Os PBL01, 02 e 04 serão implementados individualmente. O PBL03 será implementado em grupo, através de dois times, onde cada time vai implementar seu processador e verificar o processador desenvolvido pelo outro time.

### Calendário Proposto

Módulos PBL					
Sigla	Disciplina	C.H.T.	Turnos	Responsável	Data
PBL01	Desenvolvimento de uma solução em RTL com FSM	24	6 x 4h	Auto Aprendizado	29/04/25 05/05/25
PBL02	Desenvolvimento da solução em linguagem de máquina RISC-V	32	8 x 4h	Auto Aprendizado	06/05/25 09/05/25
PBL03	Desenvolvimento/Verificação UVM processador RISC-V ISA limitado	176	44 x 4h	Auto Aprendizado	12/05/25 13/06/25
PBL04	Síntese, STA e Prototipagem do Design	24	6 x 4h	Auto Aprendizado	13/06/25 18/06/25

## Problema: Sistema Embarcado

**Tema:** Modelagem, implementação e validação de um sistema hardware/software para um jogo Genius.

### **Problema:**

Coprojeto hardware/software é um termo que se refere ao projeto concorrente dos componentes de hardware e de software de um sistema, de forma a atender a funcionalidade pretendida e os requisitos de projeto, geralmente em curto prazo de tempo. Há uma grande variedade de componentes envolvidos, os quais podem ser utilizados de diferentes formas, proporcionando várias alternativas de projeto, as quais deverão ser testadas e avaliadas. Há várias metodologias de coprojeto, contudo, basicamente, o que se procura fazer é o particionamento adequado das funções do sistema em componentes de hardware ou de software, com base em restrições como, por exemplo, custo, desempenho, consumo de energia, potência dissipada e grau de programabilidade. Em linhas gerais, funções do sistema que necessitam de alta velocidade e baixo consumo são mapeadas para hardware (o qual será desenvolvido através de HDLs como SystemVerilog), ao passo que funções que apresentam maiores complexidade e necessidade de retrabalho são associadas para execução em processadores pré-existentes (uso de programação).

Neste contexto, é solicitada à sua equipe o desenvolvimento de uma aplicação embarcada cujos componentes (funções) poderão ser implementados tanto em hardware (através de Systemverilog) quanto em software (através de linguagem de máquina para ISA RISC-V). Para efeito de comparação, serão desenvolvidos uma versão completamente codificada em hardware, em RTL (PBL01), e uma versão em software baixo nível, codificada em *assembly*, para o RISC-V (PBL02). A versão em linguagem de máquina deverá ser testada em um simulador RISC-V.

A equipe também terá que desenvolver um processador RISC-V simplificado em RTL (Systemverilog) e verificá-lo funcionalmente com UVM (PBL03). O processador desenvolvido deverá suportar instruções necessárias para executar o software desenvolvido no PBL02. Além disso, como última etapa do processo de aprendizagem baseada em problema, cada integrante da equipe deverá sintetizar o RTL verificado para FPGA e realizar análise de tempo estática (STA) para um clock de 200 MHz, assim como prototipá-lo no mesmo FPGA destino.

### **Descrição do Sistema:**

O sistema embarcado consiste na construção do protótipo de um jogo que remonta à infância de muitos: trata-se do jogo Genius, lançado pela Brinquedos Estrela no início dos anos 1980 (veja figura 1).

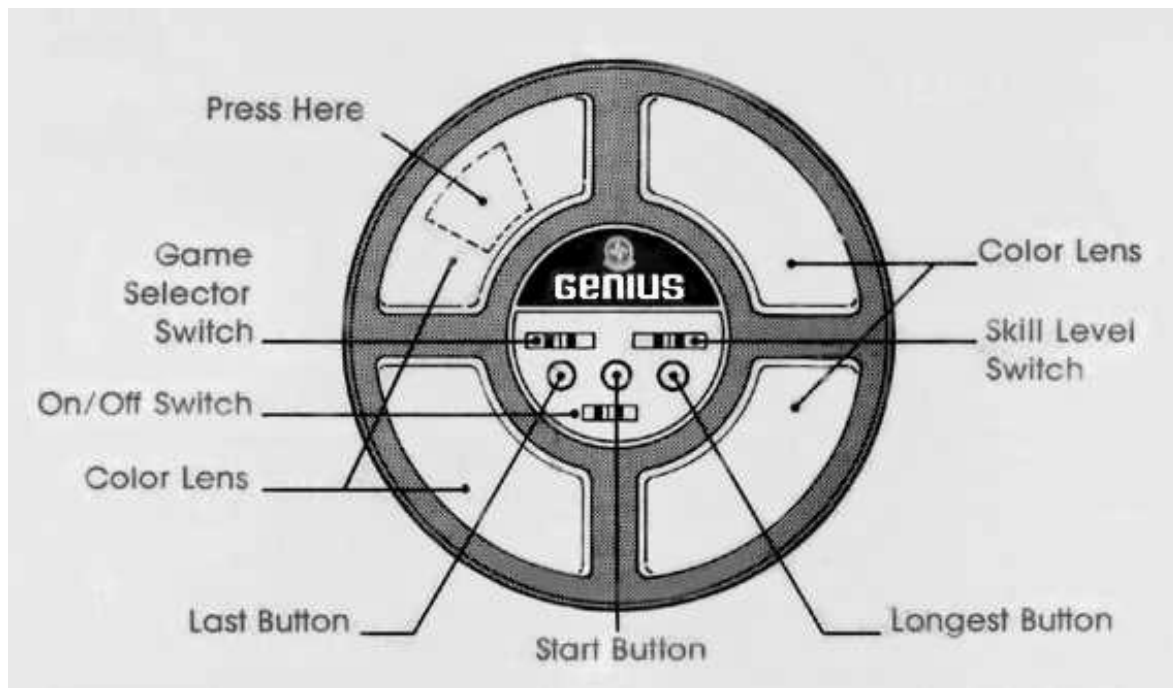


Fig. 1: Jogo Genius.

Na figura 1, observa-se 4 grandes lentes, cada qual associada a uma cor/sinal sonoro. Durante uma partida, o aparelho acende luzes situadas abaixo de tais lentes em uma dada sequência – tal sequência deverá ser reproduzida pelo jogador, teclando as lentes em sua parte central. Caso acerte a sequência, o aparelho a repete novamente, acrescentando mais uma luz (o tamanho da sequência de luzes varia entre 1 e 32, conforme o grau de dificuldade); caso falhe, o jogador perde a partida. Há outros modos de jogo, os quais se baseiam no mesmo princípio de funcionamento.

O protótipo solicitado incorpora algumas modificações ao projeto original. Em relação à funcionalidade, o protótipo deverá atender às seguintes especificações:

- 3 níveis de dificuldade:
  - **Fácil** (sequência de 8 luzes)
  - **Médio** (16 luzes)
  - **Difícil** (32 luzes)
- 2 níveis de velocidade:
  - **Lento** (2s entre o acionamento de uma luz e outra)
  - **Rápido** (1s)
- 2 modos de jogo:
  - **Siga**: jogo no qual o aparelho gera a sequência
  - **Mando eu**: jogo no qual o usuário gera a sequência (após o aparelho reproduzir a sequência entrada pelo usuário, o usuário deverá reproduzi-la, acrescentando uma nova luz) – modo apropriado para 2 jogadores
- Existência de um placar de pontuação: tal pontuação deverá levar em conta os níveis de dificuldade e de velocidade

- *Limitação de recursos de armazenamento: não poderá haver o armazenamento prévio de sequências em memória, sendo as mesmas aleatórias (PRNG)*

Por outro lado, as principais características físicas do protótipo são:

- *uso da placa FPGA*
- *interface gráfica com GPIO/LEDs*
- *uso de GPIO/ Botão como meio de entrada de comandos, um associado a cada LED*

Em relação à interface de operação, as seguintes especificações deverão ser respeitadas:

- GPIO/chaves deslizantes para escolha das seleções (dificuldade, velocidade e modo de jogo).*
- GPIO/Botão PARTIDA, com sinalização em LED durante 1s indicando que o botão foi pressionado;*
- GPIO/LED centrais acessas, quando do acionamento das respectivas GPIO/ Botão (lentes coloridas). A área central de cada lente colorida (GPIO/LED) deverá brilhar quando do acionamento da mesma. Esta área corresponde à região "Press Here" da figura 1;*
- Dígitos do placar de pontuação em GPIO para um display BCD de 7 segmentos.*

#### **Entrega Final e Demonstração:**

PBL01	Desenvolvimento de uma solução em RTL com FSM	05/05/25
PBL02	Desenvolvimento da solução em linguagem de máquina RISC-V	09/05/25
PBL03	Desenvolvimento/Verificar processador RISC-V ISA limitado	13/06/25
PBL04	Síntese, STA e Prototipagem do Design	18/06/25

#### **Produtos:**

Discussão em grupo e desenvolvimento individual para os PBLs 01, 02 e 04. Organizados em duas equipes para o desenvolvimento do PBL03.

Deverá ser utilizado a ferramenta Vivado para o desenvolvimento e verificação do sistema. As pastas de projeto deverão seguir o template <https://github.com/nelsonafn/verif-env-template>. Para simulação do PBL02, deverá ser usada RISC-V Assembly QEMU ([Web view](#)).

Deverá haver um relatório/apresentação contendo a descrição das implementações desenvolvidas, as formas de validação utilizadas, e o aprendizado adquirido por cada membro.