

**UNIVERSIDADE FEDERAL DA BAHIA - UFBA**  
**ESCOLA POLITÉCNICA / DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO**  
**2025.1 – ENGG57 – LABORATÓRIO INTEGRADO IV-A – PROF. WAGNER L. A. DE OLIVEIRA**

Este Laboratório Integrado será formado por um produto constituído por componentes de hardware e software. A nota final da disciplina (N) será formada por 3 componentes, com ponderação ajustável:

- Nota Produto (NP), com peso de 70%;
- Nota Relatório (NR), com peso de 10%;
- Nota Participação Individual (NI), com peso de 20%.

A nota NI será aferida através da participação do estudante nas sessões tutoriais e no cumprimento de metas dos projetos de software e de hardware. Caso a nota NP ou a nota NI seja inferior a 5,0 (cinco vírgula zero), haverá a aplicação de uma avaliação escrita individual que substituirá a nota final N.

#### **Agenda Prevista de Aulas**

<b>Data</b>	<b>Atividade</b>
25/03	Apresentação da Disciplina e do Problema Semestral
01/04	Sessão tutorial / desenvolvimento de atividades/tarefas pelos estudantes
08/04	Sessão tutorial / desenvolvimento de atividades/tarefas pelos estudantes
15/04	Sessão tutorial / desenvolvimento de atividades/tarefas pelos estudantes
22/04	Sessão tutorial / desenvolvimento de atividades/tarefas pelos estudantes
29/04	Entrega Parcial
06/05	Sessão tutorial / desenvolvimento de atividades/tarefas pelos estudantes
13/05	Sessão tutorial / desenvolvimento de atividades/tarefas pelos estudantes
20/05	Poli-ACTA'25
27/05	Sessão tutorial / desenvolvimento de atividades/tarefas pelos estudantes
03/06	Sessão tutorial / desenvolvimento de atividades/tarefas pelos estudantes
10/06	Entrega Parcial
17/06	Sessão tutorial / desenvolvimento de atividades/tarefas pelos estudantes
24/06	Sessão tutorial / desenvolvimento de atividades/tarefas pelos estudantes
01/07	Sessão tutorial / desenvolvimento de atividades/tarefas pelos estudantes
08/07	Sessão tutorial / desenvolvimento de atividades/tarefas pelos estudantes
15/07	Entrega Final e Demonstração
22/07	Poli-ACTA'25 Avaliação individual caso NP/NI inferior a 5,0

# Plataforma para Jogos em FPGA

## Tema:

Noções básicas de operacionalização das partes internas de um sistema processador de gráficos 2-D e de seu interfaceamento com dispositivos de memória, monitor e gamepad.

## Especificação:

O primeiro jogo para computadores foi criado em 1958 nos Estados Unidos, no laboratório de pesquisas militares **Brookhaven National Laboratory**. O programa se chamava **Tênis para Dois** e era uma simulação bem simplificada do esporte. Um ponto piscando representava a bola e dois jogadores controlavam barras que serviam de raquetes. O jogo nunca saiu do laboratório. Desde então, a indústria do videogame evoluiu a passos largos, devendo atingir receitas próximas a 200 bilhões de dólares em 2024.

Atenta a tais números, a empresa para a qual você trabalha, visando o lançamento de uma plataforma de hardware própria, de baixo custo, voltada para jogos com temas regionais e educacionais, solicitou à sua equipe o desenvolvimento de uma plataforma para jogos 2-D simples, baseada em FPGA, com o intuito de introduzi-los neste universo e desenvolver tecnologia na área.

Tal plataforma deverá ser implementada a partir de diferentes módulos, descritos na sequência, na placa **DE2-115 da Intel**, como um ou mais componentes do ambiente **Platform Designer**, visando sua expansão futura.

## **Módulo 1: Processador Gráfico – Módulo de Exibição de Imagens de Fundo**

Este módulo do processador gráfico consiste em um exibidor de imagens de fundo em formato gráfico (por exemplo, bitmap), com as seguintes características:

- Possibilidade de trabalhar com imagens *True Color*, para monitores com resolução mínima VGA;
- Possibilidade de carregar imagens externamente, usando recursos da placa, através do **Platform Designer**;
- Considerando-se que uma tela VGA corresponda a um quadro, a imagem armazenada poderá ter o tamanho de até 9 quadros: disposição horizontal (1x9), disposição vertical (9x1) e disposição igualitária (3x3);
- Possibilidade de posicionar a imagem em qualquer coluna, quando da disposição horizontal (sem extrapolar os limites de exibição);
- Possibilidade de posicionar a imagem em qualquer linha, quando da disposição vertical (sem extrapolar os limites de exibição);
- Possibilidade de posicionar a imagem em qualquer coluna e linha, quando da disposição igualitária (sem extrapolar os limites de exibição).

## Módulo 2: Processador Gráfico – Módulo de Processamento de *Sprites*

Este módulo deverá ter as seguintes características:

- Possibilidade de se trabalhar com até 32 padrões de 16x16 pixels (programáveis pelo processador principal), como elementos móveis (*sprites*), simultaneamente na tela, em diferentes níveis de sobreposição, com cores *True Color* e transparência;
- Possibilidade de sobrepor, ao menos, 4 *sprites*;
- Identificar colisões entre *sprites*; e
- Disponibilizar, para o processador principal, as seguintes informações (para cada *sprite*):
  - o perímetro da sobreposição do *sprite* com a imagem de fundo, para cada uma de suas laterais;
  - para cada lateral, os valores máximo, mínimo e médio dos componentes RGB das cores dos pontos identificados na sobreposição;
  - comparação dos valores máximo e mínimo dos componentes RGB com valores limítrofes fornecidos pelo processador principal (através de programação); e
  - comparação do valor médio dos componentes RGB com um intervalo de referência fornecido pelo processador principal (através de programação), para identificação de tons de cores da área sobreposta para cada lateral.

## Módulo 3: Processador para Jogos

O processador otimizado para jogos deverá ter as seguintes características:

- Instruções que permitam trabalhar com imagens *True Color* armazenadas na SDRAM, conforme descrito na especificação do **Módulo de Exibição de Imagens de Fundo**. Tais instruções deverão garantir as seguintes possibilidades:
  - posicionar a imagem em qualquer coluna, quando da disposição horizontal (sem extrapolar os limites de exibição)
  - posicionar a imagem em qualquer linha, quando da disposição vertical (sem extrapolar os limites de exibição)
  - posicionar a imagem em qualquer coluna e linha, quando da disposição igualitária (sem extrapolar os limites de exibição)
- Instruções que permitam trabalhar com até 32 padrões de 16x16 pixels, como elementos móveis (*sprites*), simultaneamente na tela, em diferentes níveis de sobreposição, com um código de cor reservado para transparência;
- Interfacear um ou mais dispositivos de memória da placa DE2-115, através do **Platform Designer**, para permitir a leitura do programa e a configuração do **Módulo de Processamento de *Sprites***, mediante esquema de arbitragem que dê prioridade ao fluxo de exibição VGA do **Módulo de Processamento de *Sprites***;
- Capacidade máxima de endereçamento: 32k x 32;
- Instruções e operações internas em 32 bits;
- Mínimo de 32 registradores de propósito geral: R0 – R31 (R0 e R1 auxiliares);
- Modos de endereçamento:
  - imediato
  - base-deslocamento
  - a registrador

- Possibilidade de comunicação com dispositivos de E/S externos (por exemplo, entradas de gamepads e saída de áudio);
- Conjunto de instruções: vide tabela 1, mais as instruções para manipulação de *sprites* e *imagens*.

**Tabela 1. Conjunto de instruções para o processador proposto.**

Tipo	Instrução	Operandos	Descrição
Transferência de Dados	LW	$R_D, I_{16}(R_B)$	$R_D \leftarrow [I_{16}+R_B] \mid R_D \leftarrow [R_B] \mid R_D \leftarrow I_{16}$
	SW	$R_S, I_{16}(R_B)$	$[I_{16}+R_B] \leftarrow R_S \mid [R_B] \leftarrow R_S$
	MOV	$R_D, R_S$	$R_D \leftarrow R_S$
Aritmética	ADD	$R_D, R_S$	$R_D \leftarrow R_D + R_S$
	SUB	$R_D, R_S$	$R_D \leftarrow R_D - R_S$
	MUL	$R_D, R_S$	$R_D \leftarrow R_D * R_S$
	DIV	$R_D, R_S$	$R_D \leftarrow R_D / R_S$
Lógica	AND	$R_D, R_S$	$R_D \leftarrow R_D \text{ AND } R_S$
	OR	$R_D, R_S$	$R_D \leftarrow R_D \text{ OR } R_S$
	SHL	$R_D, R_S$	$R_D \leftarrow R_D \ll R_S$
	SHR	$R_D, R_S$	$R_D \leftarrow R_D \gg R_S$
	CMP	$R_D, R_S$	compara $R_D$ e $R_S$ , configurando RFlags
	NOT	$R_D$	$R_D \leftarrow \text{NOT } R_D$
Transferência de Controle	JR	$R$	desvio incondicional para o endereço $[R]$
	JPC	$I_{26}$	desvio incondicional relativo a PC atualizado
	BRFL	$R, I_7, M_7$	desvio para o endereço $[R]$ se $RFlags[i] = I_7[i]$ , para todo $i = 1$ na máscara $M_7$
	CALL	$R$	chamada de sub-rotina no endereço $[R]$
	RET	Nenhum	retorno de sub-rotina
	NOP	Nenhum	nenhuma operação

Obs.:  $[]$  = conteúdo de memória (de dados ou de instruções, a depender da instrução executada)  
 $R_0 = 0$  e  $R_1 = 1$  somente podem ser usados para leitura  
RFlags = vetor de 7 bits com componentes [overflow; above; equal; below; between; collision; error]  
 $I_7$  = vetor de 7 bits a ser comparado com RFlags, mediante máscara  $M_7$   
 $M_7$  = vetor de 7 bits (componente = 1 ativa comparação; componente = 0 desativa comparação)

- As instruções a seguir, para manipulação de *sprites* e *imagens*, poderão ser implementadas diretamente no processador ou serem suportadas via tradutor/compilador):
  - **SPRITE\_LEVEL** (*sprite\_level*, *id\_sprite*)  
para definição do shape associado a um nível (obs.: *id\_sprite* = 0 corresponde à ausência de sprite)
  - **SPRITE\_POS** (*sprite\_level*, *row*, *col*)  
para posicionamento do sprite associado a um nível
  - **SPRITE\_COLLISION\_BG** (*sprite\_level*, *side*, *R*, *G*, *B*,  
*selector\_min\_max\_mean*, *register\_min*, *register\_max*)  
para configuração do flag *between*: caso o valor mínimo/máximo/médio (selecionado por *selector\_min\_max\_mean*) dos componentes de cor selecionados (*R*, *G*, *B*) dos pontos sobrepostos por uma das bordas do sprite associado a um nível (borda definida por *side*) esteja dentro dos limites dados pelos parâmetros *register\_min* e *register\_max*, o flag *between* será 1 – caso contrário, será 0.

- **SPRITE\_COLLISION\_SP** (*sprite\_level\_1, sprite\_level\_2*)  
*para configuração do flag collision: caso sprite\_level\_2 pertença à lista de colisões de sprite\_level\_1, o flag collision será 1 – caso contrário, será 0.*
  - **PUT\_IMAGE** (*row, col*)  
*para posicionar a imagem de fundo na área de exibição*
  - **WAIT\_VSYNC**  
*para suspender o processador, até a ativação do sinal VSYNC pela interface VGA, visando sincronismo com cada tela exibida*
- Obs.: Outras poderão ser definidas pela equipe.

### **Produtos:**

Sua equipe deverá entregar 2 produtos:

1. O protótipo da **Plataforma para Jogos em FPGA**, como um ou mais componentes do **Platform Designer**, com um exemplo de jogo teste; e
2. Um relatório técnico, em formato de artigo IEEE, com no máximo 15 páginas, contendo a descrição do protótipo (descrição geral, partes constituintes e sua correlação, estudos realizados e detalhes relevantes para a compreensão), conclusões (problemas encontrados, abrangência da implementação, possíveis melhorias, participação individual e aprendizado adquirido por cada membro) e referências bibliográficas.

As entregas parciais serão realizadas ao longo do semestre, conforme indicado no calendário, para averiguação do desempenho, consistindo em apresentações simples no laboratório. A entrega final dos produtos deverá ser realizada em duas partes: um relatório técnico, acompanhado da pasta de projeto Quartus (link do Google Drive compartilhado com o e-mail [oliveira.wagner@ufba.br](mailto:oliveira.wagner@ufba.br)), e a apresentação do jogo, na data limite indicada no calendário.