



Atividade 6

Objetivo

O objetivo dessa atividade será desenvolver um jogo simples na placa de desenvolvimento, utilizando o vídeo, os botões, os displays de 7 segmentos e a saída de vídeo. Faça um jogo ao estilo pong.

Introdução

Seu jogo deve ser de usuário único, controlado pelos botões da placa, com um placar no display numérico e a tela na saída VGA.

Detalhes

O jogo de pong é um jogo simples no qual um jogador controla uma barra que se move para a direita e esquerda na base da tela. Uma bola se move na tela e o objetivo do jogo é fazer com que a bola não toque na base da tela. Se a bola tocar na base da tela, o jogo acaba e o placar é mostrado no display de 7 segmentos.

Você deve utilizar dois botões, para mover para direita e esquerda a barra. A bola deve bater e retornar nas 3 bordas (direita, esquerda e superior) em ângulo compatível com o da entrada. A barra horizontal da base da tela deve ser capaz de rebater a bola.

O ângulo de reflexão da bola deve ser alterado de acordo com a posição em que a bola toca na barra. Se a bola tocar no centro da barra, ela deve ser refletida normalmente. Se a bola tocar na extremidade da barra, o ângulo deve ser mais inclinado para a direita ou esquerda, conforme o canto em que bateu.

MC613 - Laboratório de Circuitos Digitais

deve terminar quando a bola tocar na base da tela. O placar deve ser mostrado no display de 7 segmentos.

Seu display de 7 segmentos deve ser dividido em duas partes, a parte direita deve mostrar o placar atual e a parte esquerda deve mostrar o placar máximo atingido. O placar máximo deve ser armazenado em um registrador e só deve ser atualizado se o placar atual for maior que o placar máximo. Em nenhum dos dois números (máximo e placar atual) você deve mostrar os zeros à esquerda. Ambos devem ser mostrados em decimal na tela.

A bola deve aumentar a velocidade à medida em que o placar aumenta. Cuidado com a jogabilidade para não tornar o jogo impossível.

Como gerenciar a tela

Quando a tela começa a ficar mais complicada, com mais componentes a serem mostrados, é importante pensar nas alternativas de controlar o que está sendo gerado nela. Para isso, algumas alternativas são interessantes e podem ser consideradas. Vamos falar sobre duas aqui: 'framebuffer' e 'sprite'. Você não precisa implementar as duas mas pode ser mais fácil com elas.

Framebuffer

O framebuffer é uma técnica que consiste em armazenar a imagem inteira que será mostrada na tela em um buffer de memória. O seu circuito geral só escreve nesse buffer e seu controlador de vídeo só lê os dados do buffer. Isso facilita o controle de temporização, evitando sincronismos extras, ao mesmo tempo em que permite que você tenha controle total do que está sendo mostrado na tela.

MC613 - Laboratório de Circuitos Digitais

vídeo que passa a ser responsável por ler sequencialmente a cada ponto que precisa ser mostrado na tela. Além dessa leitura, essa memória também tem os sinais de escrita, que são externados do controlador de vídeo para o circuito principal. Se você conseguir fazer uma memória 'dual-port', capaz de realizar uma leitura e uma escrita simultaneamente, você pode fazer o controlador de vídeo e o circuito principal compartilharem a mesma memória sem dificuldades. Se sua memória for 'single-port', então você terá que fazer um controle de acesso para que o controlador de vídeo e o circuito principal não acessem a memória ao mesmo tempo. Nesse último caso, o controlador de vídeo deve ter sempre a prioridade máxima.

Sprite

O sprite é uma técnica que consiste em ter um circuito que gera uma imagem específica e que é capaz de ser movida na tela. O controlador de vídeo é responsável por mover esse sprite na tela, sem se preocupar com o que está sendo mostrado. O circuito principal só precisa se preocupar em gerar o sprite e atualizar sua posição.

Um sprite é uma imagem pequena, que funciona como se fosse um mini framebuffer, juntamente com as coordenadas x e y onde ele deve ser mostrado na tela. Um controlador de vídeo pode ter capacidade de múltiplos sprites distintos e também múltiplas instâncias (coordenadas) para cada sprite. Dessa forma, você pode apenas enviar as coordenadas x e y de cada um deles e seu controlador de vídeo se encarrega de mostrar na tela.

Um problema que surge com framebuffer e sprite é a necessidade de consolidar os múltiplos valores para os pontos que ficam na tela. Suponha uma configuração com framebuffer e 2 sprites. O sprite, por estar em um lugar da tela, precisará ser sobreposto ao valor do framebuffer. Existem formas simples

MC613 - Laboratório de Circuitos Digitais

Cada sprite mostra um ou exclusivo dos dois e uma terceira pode ser realizada uma lógica específica para cada caso, ou sprite. O mesmo problema acontece quando você tem 2 sprites compartilhando um pedaço da tela, o que nos coloca com potencialmente 3 valores para o mesmo ponto (1 do framebuffer + 2 dos sprites).

O que fazer

Tarefa 1

Implemente a tela de fundo e a barra horizontal que deve se mover para a direita e esquerda de acordo com os botões. A barra não pode ultrapassar as bordas da tela. O arquivo de alto nível deve ser o ***top1.v***.

Tarefa 2

Implemente a bola que se move na tela e rebata nas bordas e na barra. A bola deve se mover em ângulo de acordo com a posição em que bateu na barra. Implemente o fim de jogo. Você deve utilizar um outro botão para iniciar o jogo. O arquivo de alto nível deve ser o ***top2.v***.

Tarefa 3

Implemente o placar do jogo. O placar deve ser incrementado a cada vez que a bola tocar na barra. O placar máximo deve ser atualizado se o placar atual for maior que o placar máximo. O arquivo de alto nível deve ser o ***top3.v***.

MC613 - Laboratório de Circuitos Digitais

Implemente a velocidade da bola. A bola deve aumentar a velocidade à medida em que o placar aumenta. O arquivo de alto nível deve ser o ***top4.v***.

Entrega

O código deve ser entregue no GitHub Classroom. O código deve ser entregue até o dia 01/04 até as 15h.

← Previous
Atividade 5

Next →
Atividade 7