

# Type or Die

Uma odisséia nas palavras

Projeto de MC613 - **Laboratório de Circuitos Digitais**

Membros:

Gabriel Pellegrino da Silva	- 172358
Diogo Teles Sant'Anna	- 169966

# Introdução

O projeto será um jogo de digitação. Ao longo do jogo, palavras aleatórias descem pela tela e o jogador deve digitar as palavras para que elas sumam da tela. O jogo termina quando alguma palavra atinge o limite inferior da tela.

A tela do jogo será composta basicamente por palavras descendo na tela. O jogador então poderá digitar alguma letra para iniciar a digitar alguma das palavras presentes na tela.

Se a letra que o jogador inserir não for a letra inicial de nenhuma das palavras na tela, isso é contabilizado com um erro de digitação, mas se a letra que o jogador digitar for a letra inicial de alguma das palavras visíveis, o jogador então começa a digitar essa palavra e precisa terminar de escrever essa palavra antes de começar a escrever outra.

Enquanto escreve uma palavra, se a letra inserida não corresponder à seguinte letra da palavra, isso é contabilizado como erro de digitação. Agora, quando o jogador termina de escrever uma palavra, ele novamente é livre para começar a escrever a palavra que quiser. Para o caso particular em que a letra digitada pelo jogador é a letra inicial de mais de uma das palavras dispostas na tela, o jogo automaticamente seleciona a palavra mais próxima do inferior da tela.

Ao final de cada partida é indicada a pontuação do jogador baseado no número de letras digitadas corretamente e no número de erros. O jogador apenas perde o jogo quando alguma palavra atinge o limite inferior da tela, mas à medida que o jogador digita letras diferentes das que são esperadas no momento, esses erros são contabilizados e mostrados



Figura 1: Diagrama de fluxo

## Brainstorm do diagrama de blocos:

// Converte o Clock padrão da placa para o nosso clock padrão

### **Clock\_parser**

**in:**

Clock\_50: std\_logic

**out:**

Clock: std\_logic

// Responsável por processar a tecla apertada e verificar se é uma das teclas que  
 // estávamos esperando em um tabela de palavras existentes. E também remover as letras  
 // pressionadas pelo usuário da tabela. De modo que Screen\_Processor só imprima a tabela

### **Typing\_Processor**

**in:**

PrEn : std\_logic // '0' if no key pressed and '1' if key pressed

new\_word\_id : natural // Index for this new word

NwEn : std\_logic // '0' if there's no word to add and else '1'

```

    key_pressed: string    // Key pressed from State_Machine
    new_word: string       // A new word to be inserted on table

out:

    allocated : std_logic  // '0' if no word is being destroyed and '1' if a word is being
                           // destroyed
    right_letter: std_logic // '1' if key_pressed corresponds to a valid letter in a word
                           // depending on allocated value, else '0'
    miss: std_logic        // '0' if allocated and pressed wrong letter of word or
                           // not allocated and pressed a letter that doesn't start a word in
                           // the table
    id_table: natural      // index of current allocated word in table (-1 if no word
                           // allocated). To be used by other word tables

signal:

    min_table_id : natural    // Minimum valid position in word's table
    max_table_id : natural    // Maximum valid position in word's table
    id_table_cur : natural    // Current table index for our allocated word
    words_table : std_logic_vector of strings // Words table

```

// Responsável por atual

### **Screen\_Processor**

**in:**

```

    new_word_id : natural // Index for this new word
    new_word: string      // A new word to be inserted on table
    NwEn : std_logic      // '0' if there's no word to add and else '1'
    game_lost: std_logic  // '0' if the game is running and '1' if the game is over
    stage_end: std_logic  // '0' if we're in a stage and '1' if the stage has ended
    go_down: std_logic    // '0' if letters are OK and '1' if they need to down 1 pixel
    new_word: string      // A new word to be added to our table of words
    id_table: natural      // index of current allocated word in table (-1 if no word
                           // allocated). We need this to print this word in a different color
    right_type: std_logic  // Foi inserida uma letra corretamente
    miss_type: std_logic   // Foi digitado uma letra errada (voto deixar pra depois)

```

**out:**

```

    parsed_screen: // A representação final da tela, a ser enviada para State_Machine

```

**signal:**

```

    stage: natural // A fase em que estamos no jogo
    words_table : std_logic_vector of strings // Words table
    min_table_id : natural    // Minimum valid position in word's table
    max_table_id : natural    // Maximum valid position in word's table
    position_table : std_logic_vector

```

// Entidade responsável por gerar as novas palavras. Ela recebe o clock da máquina de  
// estados, instancia os contadores e fornece as palavras.

### **Word\_Generator**

**in:**

```

    generate: std_logic

```

```

out:
    new_word: string      // A new word to be inserted on the tables
    word_id : natural     // Index for this new word
signal:
    word_id : signal      // Incremented at each new word and % (BIG_NUMBER)
// Conta de 1 até N e ativa a finished toda vez que terminar
Counter
generic:
    N: std_logic_vector   // Or a natural
in:
    clock: std_logic
out:
    finished: std_logic

Random Gen:
in:

out:

```

## Descrição Funcional:

Utilizaremos um monitor e um teclado como periféricos. Todo o resto será sincronizado por uma FPGA distribuída para os alunos da disciplina. Basicamente, o jogo administra os inputs do usuário pelo teclado e mostra no monitor um acompanhamento do estado atual das palavras na tela.

Estados da nossa máquina finita de estados (utilizaremos apenas uma):

### Início

Estado para o início do jogo. Será apenas uma tela inicial amigável com o nome do jogo na qual o jogador deve pressionar uma tecla start para iniciar o jogo (passar para o estado nova fase)

### Nova fase

Estado responsável por inicializar as variáveis que diferenciam uma fase de outra, como a quantidade de palavras e velocidade das mesmas. Pula diretamente para o estado “não começou nenhuma palavra”.

Atualiza tela

Estado responsável pela atualização da tela. Ele deve atualizar a posição das palavras de tempos em tempos e também criar novas palavras na tela. Também é responsável por verificar se o jogador perdeu e, nesse caso, pular para o estado “Perdeu”. Exceto por essa relação citada, o estado “atualiza tela” sempre retorna instantaneamente para o estado de onde o sistema veio (o estado anterior).

## Não começou nenhuma palavra

Enquanto nesse estado o jogador ainda não começou a escrever nenhuma palavra, e para qualquer tecla digitada se tentará iniciar a escrita de uma palavra(pular para o estado “preso a uma palavra”). Caso a letra não inicie nenhuma das palavras disponíveis na tela, o jogador será penalizado e continuará nesse estado. Esse estado também escuta um timer que de tempos em tempos direciona o sistema para o estado “atualiza a tela”.

## Preso a uma palavra

Nesse estado o jogador já iniciou a escrever uma palavra e portanto o sistema espera que o usuário digite a letra seguinte da palavra que está escrevendo. Se o usuário digitou a letra correta, verifica-se se é a última letra da palavra, então pula-se para o estado “Destrói palavra” ou para o “Destrói letra”, caso contrário, o usuário tem uma penalização na pontuação e continua nesse estado. Esse estado escuta um timer que de tempos em tempos direciona o sistema para o estado “atualiza a tela”. Um detalhe importante desse estado é alterar alguma propriedade da palavra que estamos presos, a fim de facilitar a sua identificação entre o restante das palavras.

## Destrói letra

Esse estado é responsável pela destruição das letras de uma palavra. As maneiras de entrar nesse estado são: não tendo começado uma palavra ou no meio de uma palavra. Mas de ambas as maneiras esse estado irá remover uma letra de uma palavra. Esse estado sempre retorna instantaneamente para o estado “preso a uma palavra”.

## Destrói palavra

Esse estado é responsável por contabilizar a pontuação relacionada à destruição de uma palavra, bem como liberar o usuário para destruir novas palavras. Além disso, esse estado irá verificar se uma fase acabou e se tiver acabado passa para o estado de Nova Fase.

## Perdeu

Estado responsável pelo término de uma partida. Esse estado ocorre quando uma letra chegar à parte inferior da tela. Nesse estado o usuário poderá com um input do teclado ir para o estado de início ou manter-se nessa tela com o placar da sua partida.

