

Projeto Final de LSD

Universidade de Aveiro

Inês Moreira, Telmo Sauce



Guess nº2

Projeto Final de LSD

Departamento de Eletrónica, Telecomunicações e
Informática

Universidade de Aveiro

Inês Moreira, Telmo Sauce
(100084) inesfm@ua.pt, (104428) telmobelasauce@ua.pt

16 de junho de 2021

Índice

1	Introdução	1
2	Metodologia	2
2.1	Funcionamento das Teclas do Jogo	2
2.2	Máquinas De Estados	4
2.2.1	Máquina de Estados BCD	4
2.2.2	Máquina Principal	5
3	Conclusões	6

Capítulo 1

Introdução

O trabalho que nos foi atribuído consiste num jogo com o objetivo de mostrar na nossa FPGA o número que foi pensado por nós através de várias tentativas em erro. Número este que está contido entre 0 e 999.

Neste relatório temos como objetivo explicar como funciona cada tecla e demonstrar as máquinas de estado por de trás do jogo.

Capítulo 2

Metodologia

2.1 Funcionamento das Teclas do Jogo

Ao inicializarmos o jogo a máquina irá mostrar no display de 7 segmentos a mensagem "Guess_n5" durante 10s e de seguida mostra-nos "Grupo_n1". Quando esta última mensagem é mostrada a máquina divide automaticamente o nosso intervalo tendo como primeira referência o número 500.

A partir deste número 500 a máquina vai começar a tentar adivinhar o número que foi pensado por nós. Usamos as teclas (que vão ser explicadas já de seguida) para ajudar neste processo.

Temos 4 respostas possíveis: "Hi", "Lo", "==" e "_batota_":

1. "hi" caso o numero calculado seja maior que o pensado.
2. "lo" caso o numero calculado seja menor que o pensado.
3. "==" caso o numero calculado seja igual ao pensado.
4. "_batota_" caso o numero calculado não esteja dentro do intervalo ou a FPGA não possa mais calcular valores novos.

As teclas são:

- SW[0]- Funciona como um "reset global", isto é, o jogo volta ao estado inicial que é o start.
- KEY[3]- Inicializa o jogo, passa do estado "grupo" para o estado "Calc_mid".
- KEY[0]- O jogo quando se encontra no estado "Cheater" ou "Win" volta ao estado "grupo".
- KEY[1]- Tem várias funções:
 - Ajuda a máquina a ir igualando o número ao que foi pensado pelo jogador.

- Supondo que o número que foi pensado por nós é mais baixo que o número pensado pela máquina (no caso do jogo estar no início seria o 500) o KEY[1] muda para o estado "Calc_mid", onde é calculada a nova tentativa, dando a entender à máquina que o número é mais alto.
 - Serve também para chegar ao estado “Cheater” caso o jogador tenha pensado num numero exterior ao intervalo(0 a 999).
 - Caso estejamos no estado "Lo"onde a máquina pensou num número mais baixo do que aquele que o Jogador escolheu a tecla KEY[1] permite voltar para o estado "Calc_mid", para calcular a nova tentativa.
 - Atingindo o estado “equals” significa que o jogo está ganho e carregamos em KEY[1] passando para o estado "Win"que faz com que o display e os leds pisquem.
- KEY[2]- É um ciclo que vai do "Hi"para o "Lo", do "Lo"para o "equals"e do "equals"para o "Hi".

2.2 Máquinas De Estados

2.2.1 Máquina de Estados BCD

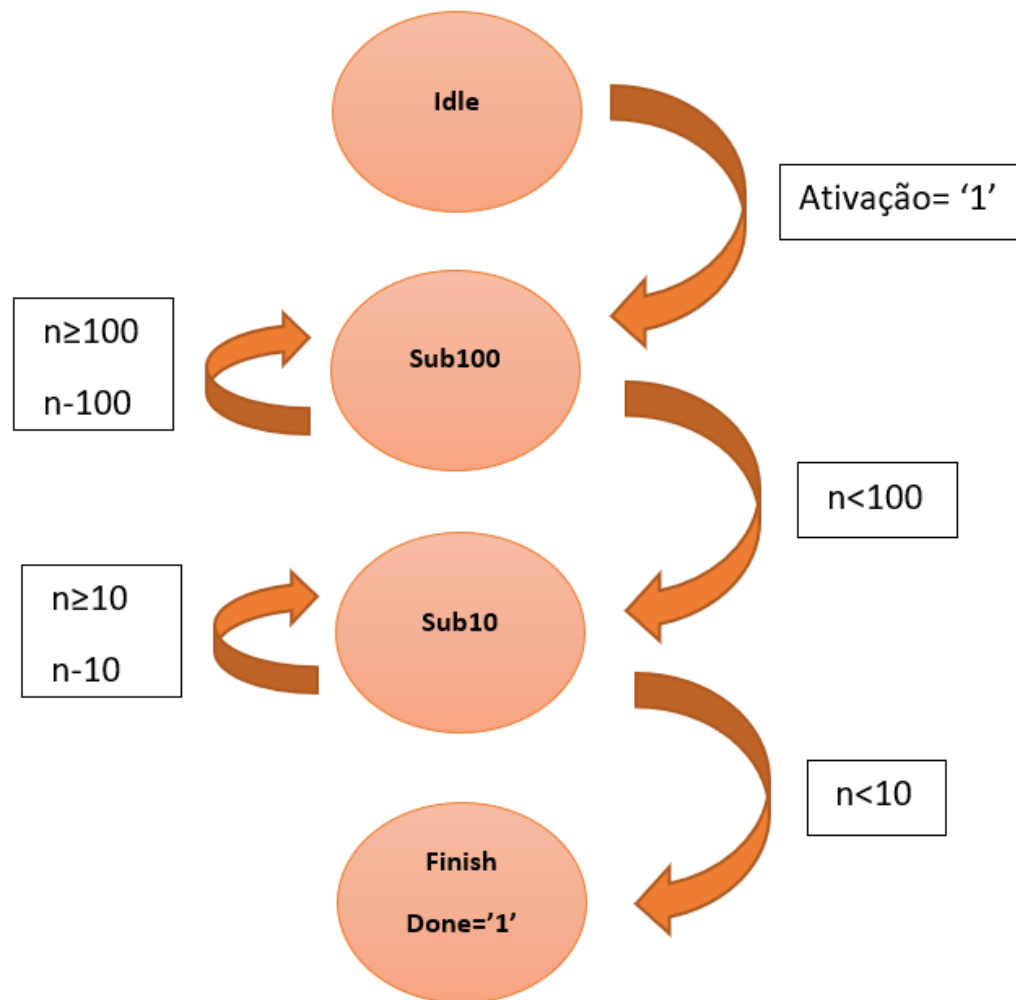


Figura 2.1: Máquina de Estados BCD

2.2.2 Máquina Principal

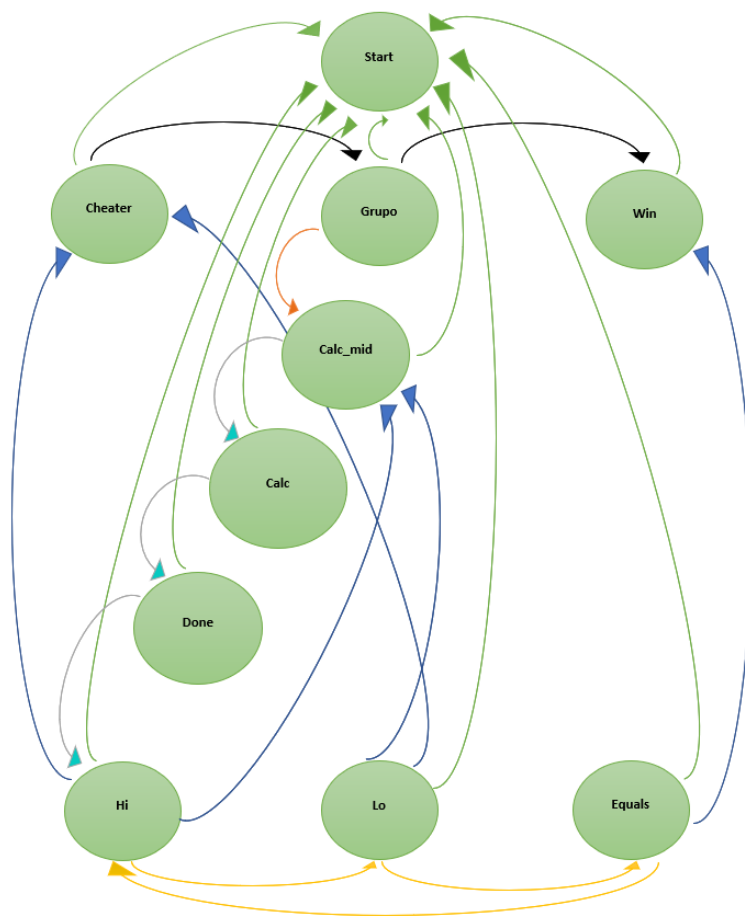


Figura 2.2: Máquina de Estados Principal

Para uma melhor organização do nosso esquema optamos por fazer as setas com cores e explicar a baixo ao que é que estas estão associadas:

- A cor verde está associada à tecla “SW[0]”
- A cor laranja está associada à tecla “KEY[3]”
- A cor preta está associada à tecla “KEY[0]”
- A cor azul está associada à tecla “KEY[1]”
- A cor amarela está associada à tecla “KEY[2]”
- A cor azul clara é passagem automática.

Capítulo 3

Conclusões

Por fim conseguimos realizar com sucesso as duas primeiras fases do trabalho que consistem no funcionamento total do jogo, incluindo a espera da conversão para BCD (Binary-coded decimal), e manipulação dos Leds e dos displays, fazendo com que estes pisquem.

Acrónimos

Binary-coded decimal BCD