



universidade de aveiro
theoria poiesis praxis

Relatório - Oferta de Produtos

Turma 6 - César Malhão N^o104279 e Eduardo Alves N^o104179
Universidade de Aveiro - Laboratórios de Sistemas Digitais

16 de junho de 2021

Conteúdo

Introdução	2
Manual do utilizador	2
Arquitetura	3
Implementação	3
Conclusão	5

Introdução

O objetivo deste projeto seria o de programar a *FPGA* de forma a que fosse possível interagir com uma máquina de oferecer produtos. O sistema iniciaria com uma mensagem de boas vindas, que estaria visível durante 6 segundos, passando depois para uma mensagem para escolher um refrigerante. De seguida o utilizador deveria selecionar uma bebida entre: fanta, sumo de limão e sumo de pêssago. Após a escolha da bebida o sistema iniciaria a preparação da bebida, que levaria 8s e durante este tempo a *FPGA* deveria mostrar o nome da bebida e acender o LEDR. Caso o utilizador entrasse no modo cubos de gelo, o tempo de preparação da bebida entraria em pausa podendo o utilizador escolher entre 0 a 4 cubos de gelo que deveriam aparecer na *FPGA*. Depois da escolha de cubos de gelo o tempo retornaria e quando terminasse a *FPGA* deveria mostrar o nome da bebida e acender o LEDG. Caso o utilizador opte por não entrar no modo cubos de gelo, após os 8 s de preparação da bebida a *FPGA* deveria mostrar o nome da bebida e acender o LEDG.

Manual do utilizador

O utilizador deverá interagir com a *FPGA* através dos cinco interruptores: *SW0* deverá poder reiniciar todo o sistema, *SW1* para selecionar a bebida fanta, *SW2* para selecionar a bebida sumo de limão, *SW3* para selecionar a bebida sumo de pêssago e *SW4* para, durante a preparação da bebida, entrar no modo de selecionar cubos de gelo. E através de dois botões: *KEY0* deverá poder diminuir o número de cubos de gelo e *KEY1* deverá poder aumentar o número de cubos de gelo, isto apenas quando o utilizador se encontrar no modo de seleção de cubos de gelo. Os oito *displays* hexadecimais deverão mostrar, da esquerda para a direita

(de HEX7 a HEX4), a quantidade de cubos de gelo e (de HEX3 a HEX0) o modo Start com a palavra HIII, ou o modo de escolher o refrigerante, com a palavra EREF, ou ainda o nome da bebida (FAN, SOLI, SPES) para as bebidas fanta, sumo de limão e sumo de pêsego, respectivamente.

Arquitetura

O diagrama de blocos do sistema, por ser demasiado extenso, encontra-se no anexo (ficheiro "Anexo 1 - Diagrama de Blocos"). Seguidamente iremos explicar passo a passo a função de cada entidade descrita nesse diagrama, da esquerda para a direita.

O sistema tem como entradas cinco interruptores ($SW(4)$, $SW(3)$, $SW(2)$, $SW(1)$ e $SW(0)$), dois interruptores ($KEY0$ e $KEY1$) e um relógio (*clock*). O relógio utilizado em todos os blocos foi associado ao *CLOCK50*.

Através do bloco *RegisterRef* que pudemos visualizar no diagrama de blocos do sistema, são associados os interruptores ($SW(3)$, $SW(2)$, $SW(1)$ e $SW(0)$) com as entradas *selGelo*, *sumoPes*, *sumoLim*, *fanta* e *reset*, respectivamente, que são utilizadas posteriormente em outros blocos.

Ligado a cada botão está um *debouncer*, para evitar possíveis transições indesejadas.

No diagrama de blocos temos a entidade *OferProd*, que tem como entradas: as saídas do *RegisterRef*, o *clock* e o *timeExp*. E como saídas: o *timeVal*, *timerEnable* e *newTime* para o bloco *timerFSM*, o *selWord* e *enable* para o *Bin7SegDecoder*, o *selModoGelo* para o *CubosGelo* e os *LEDR* e *LEDG*.

A entidade *timerFSM* permite que dados as entradas *timeVal*, *newTime* e *timerEnable* inicie uma contagem e no final da contagem retorne como saída o *timeExp*.

A saída do *debouncer*, ligado ao botão $KEY(0)$ e $KEY(1)$, serve para diminuir ou aumentar, respectivamente, o número de cubos de gelo.

No diagrama de blocos temos a entidade *CubosGelo*, que tem como entradas: a saída *selModoGelo* do bloco *OferProd*, as saídas dos *debouncers* e o *reset* e *clock*. E como saídas: o *selcubo* e *enable* para o *Bin7SegDecoder*.

Os *Bin7SegDecoders* terão com saída os *displays* hexadecimais (HEX7 a HEX0).

Implementação

Nesta secção, iremos explicar o que faz cada máquina de estados: *OferProd* e *CubosGelo*.

A entidade *OferProd* é uma máquina de estados com oito estados (ver "Anexo 2 - Máquina de Estados"). Por defeito começa no estado Start e as saídas *LEDG*, *LEDR*, *timeVal* e *selModoGelo* estão a '0' e *enable* a '1'. No primeiro estado (Start), as saídas *timerEnable* e *newTime* ficam a '1' e *selWord* e *timeVal* assumem valores de constantes. E o próximo estado será o StartDone.

No segundo estado (StartDone), a saída *timerEnable* fica a '1' e *newTime* fica a '0' e *selWord* assume valores de constantes. Depois, caso *timeExp* seja '1' o próximo estado será EREF, senão o próximo estado será StartDone.

No terceiro estado (EREF), a saída *timerEnable* fica a '1' e *newTime* fica a '0' e *selWord* e *timeVal* assumem valores de constantes. Depois, caso *fanta* seja '1' o próximo estado será PrepFan, caso *sumoLim* seja '1' o próximo estado será PrepSoli, caso *sumoPes* seja '1' o próximo estado será PrepSpes, senão o próximo estado será EREF.

No quarto estado (PrepFan), a saída *timerEnable* e *LEDR* fica a '1' e *newTime* fica a '0' e *selWord* assume valores de constantes. Depois, caso *selGelo* seja '1' o próximo estado será PrepFan e a saída *timerEnable* fica a '0' e *selModoGelo* fica a '1', caso *timeExp* seja '1' o próximo estado será DisFan, caso *fanta* seja '0' o próximo estado será EREF, senão o próximo estado será PrepFan.

No quinto estado (DisFan), a saída *timerEnable* e *LEDG* ficam a '1' e *newTime* fica a '0' e *selWord* assume valores de constantes. Depois, caso *fanta*, *sumoLim* e *sumoPes* sejam '0' o próximo estado será EREF, senão o próximo estado será DisFan.

No sexto estado (PrepSoli), a saída *timerEnable* e *LEDR* fica a '1' e *newTime* fica a '0' e *selWord* assume valores de constantes. Depois, caso *selGelo* seja '1' o próximo estado será PrepSoli e a saída *timerEnable* fica a '0' e *selModoGelo* fica a '1', caso *timeExp* seja '1' o próximo estado será DisSoli, caso *sumoLim* seja '0' o próximo estado será EREF, senão o próximo estado será PrepSoli.

No sétimo estado (DisSoli), a saída *timerEnable* e *LEDG* ficam a '1' e *newTime* fica a '0' e *selWord* assume valores de constantes. Depois, caso *fanta*, *sumoLim* e *sumoPes* sejam '0' o próximo estado será EREF, senão o próximo estado será DisSoli.

No oitavo estado (PrepSpes), a saída *timerEnable* e *LEDR* fica a '1' e *newTime* fica a '0' e *selWord* assume valores de constantes. Depois, caso *selGelo* seja '1' o próximo estado será PrepSpes e a saída *timerEnable* fica a '0' e *selModoGelo* fica a '1', caso *timeExp* seja '1' o próximo estado será DisSpes, caso *sumoPes* seja '0' o próximo estado será EREF, senão o próximo estado será PrepSpes.

No nono estado (DisSpes), a saída *timerEnable* e *LEDG* ficam a '1' e *newTime* fica a '0' e *selWord* assume valores de constantes. Depois, caso *fanta*, *sumoLim* e *sumoPes* sejam '0' o próximo estado será EREF, senão o próximo estado será DisSpes.

A entidade *CubosGelo* é outra máquina de estados, com seis estados (ver "Anexo 2 - Máquina de Estados"). Por defeito começa no estado INICIO e a saída *enable* está a "0000". No primeiro estado (INICIO), a saída *selcubo* assume valores de constantes. Depois, caso *selModoGelo* seja '1' o próximo estado será DOISCUBOS, senão o próximo estado será o INICIO.

No segundo estado (DOISCUBOS), a saída *enable* fica a "0011" e *selcubo* assume valores de constantes. Depois, caso *aumentarGelo* e *selModoGelo* sejam '1' o próximo estado será TRESCUBOS, caso *diminuirGelo* e *selModoGelo* sejam '1' o próximo estado será UMCUBO, caso *selModoGelo* seja '0' o próximo estado

será INICIO, senão o próximo estado será DOISCUBOS.

No terceiro estado (UMCUBO), a saída *enable* fica a "0001" e *selcubo* assume valores de constantes. Depois, caso *aumentarGelo* e *selModoGelo* sejam '1' o próximo estado será DOISCUBOS, caso *diminuirGelo* e *selModoGelo* sejam '1' o próximo estado será ZEROCUBOS, caso *selModoGelo* seja '0' o próximo estado será INICIO, senão o próximo estado será UMCUBO.

No quarto estado (ZEROCUBOS), a saída *enable* fica a "0000" e *selcubo* assume valores de constantes. Depois, caso *aumentarGelo* e *selModoGelo* sejam '1' o próximo estado será UMCUBO, caso *diminuirGelo* e *selModoGelo* sejam '1' o próximo estado será ZEROCUBOS, caso *selModoGelo* seja '0' o próximo estado será INICIO, senão o próximo estado será ZEROCUBOS.

No quinto estado (TRESCUBO), a saída *enable* fica a "0111" e *selcubo* assume valores de constantes. Depois, caso *aumentarGelo* e *selModoGelo* sejam '1' o próximo estado será QUATROCUBOS, caso *diminuirGelo* e *selModoGelo* sejam '1' o próximo estado será DOISCUBOS, caso *selModoGelo* seja '0' o próximo estado será INICIO, senão o próximo estado será TRESCUBOS.

No sexto estado (QUATROCUBOS), a saída *enable* fica a "1111" e *selcubo* assume valores de constantes. Depois, caso *aumentarGelo* e *selModoGelo* sejam '1' o próximo estado será QUATROCUBOS, caso *diminuirGelo* e *selModoGelo* sejam '1' o próximo estado será TRESCUBOS, caso *selModoGelo* seja '0' o próximo estado será INICIO, senão o próximo estado será QUATROCUBOS.

Conclusão

O projeto alcançou e foi de encontro aos objetivos definidos. Foi um projeto que estimulou o nosso interesse pela unidade curricular, principalmente na programação e no uso das *FPGA*'s. Optamos pelo projeto final número 3 (Máquina automática de oferta de Produtos (versão 3)) devido ao nosso interesse pela comunicação entre máquinas de estado, acompanhado também, pelo nosso interesse em um projeto capaz de nos desenvolver e progredir nos nossos estudos.

O maior desafio foi também o que nos mostrou interesse em primeiro lugar, a comunicação entre a máquina de estados principal e a máquina de estados usada para implementar os cubos de gelo. Foi nesta fase, onde utilizamos a maior parte do nosso tempo e que foi superada com planejamento e estudo dos guiões feitos ao longo do semestre. O projeto foi elaborado em devido tempo, cumprindo também a data imposta de entrega do projeto.