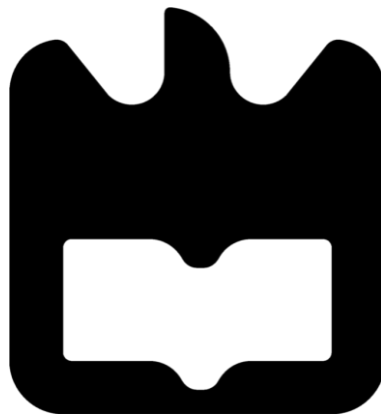


Relatório final

Gerador de números primos

HUGO LEAL (nº 93059), LUÍSA AMARAL (nº 93001)

Laboratório de Sistemas Digitais



Índice

Introdução	2
Arquitetura.....	2
Implementação	3
Validação	5
Conclusão	6
Divisão percentual do trabalho	7
Manual do utilizador	7

INTRODUÇÃO

Este projeto foi desenvolvido no âmbito da disciplina Laboratório de Sistemas Digitais, lecionada na Universidade de Aveiro a alunos do 1º ano dos cursos Engenharia de Computadores e Telemática e Engenharia Eletrónica e de Telecomunicações.

O sistema desenvolvido possui como objetivo detetar números primos num intervalo fechado de números inteiros. A dimensão deste intervalo é definida pelo utilizador, sendo que o valor mínimo que poderá ser escolhido é 0 e o máximo é 1023. Após a escolha do intervalo, o sistema percorre-o e identifica os números primos, armazenando-os numa memória RAM para, quando o processo de pesquisa estiver concluído, mostrá-los ao utilizador através dos *displays* de 7 segmentos (ou do *display* LCD).

Assim, são gerados e visualizados os números primos entre dois números inteiros escolhidos previamente, permitindo o sistema cumprir as funcionalidades pedidas.

ARQUITETURA

Através da abordagem faseada que foi adotada para realizar este projeto, obteve-se uma construção gradual dos vários componentes do sistema (como podemos ver na Figura 1). Cada fase desenvolvida corresponde a um projeto novo no *Quartus Prime*.

Inicialmente, foram implementados dois registos. Cada um deles guarda um extremo do intervalo. Deste modo, quando é escolhido o valor mínimo e após a validação deste através do botão SAVE, este é guardado no primeiro registo e o mesmo processo repete-se para o valor máximo, tendo o cuidado de verificar que este é maior que o mínimo. Posteriormente é varrido o intervalo escolhido, sendo estas ações controladas por uma máquina de estados. Após o registo dos extremos do intervalo, o utilizador desencadeia o início da pesquisa dos números primos através do botão *START*. Esta é realizada através de vários blocos que executam variadas operações e de uma máquina de estados que os coordena. Quando concluída, acende-se um LED verde. Assim, o utilizador toma conhecimento do fim da pesquisa e os valores terão sido armazenados na memória. Foram implementados blocos que permitem visualizar os valores nos *displays* de 7 segmentos, para além de ter sido também implementado um conversor de binário para BCD, adaptado para valores com 10 bits. Todos estes blocos foram implementados no projeto “Fase1”.

Foi criado o projeto “Fase2” para implementar a leitura dos números primos gerados no *display* LCD.

É importante referir que o sistema corre com um sinal de relógio de 25MHz (obtido através da divisão da frequência para metade do sinal de relógio de *CLOCK_50*) para evitar erros temporais, uma vez que o bloco *isPrime* contém lógica combinatória complexa.

O sistema possui duas máquinas de estados que o controlam. Estas permitem a validação dos limites do intervalo escolhido, o início do varrimento deste e o armazenamento dos primos na memória (*Control_Now* e o *IsPrime*).

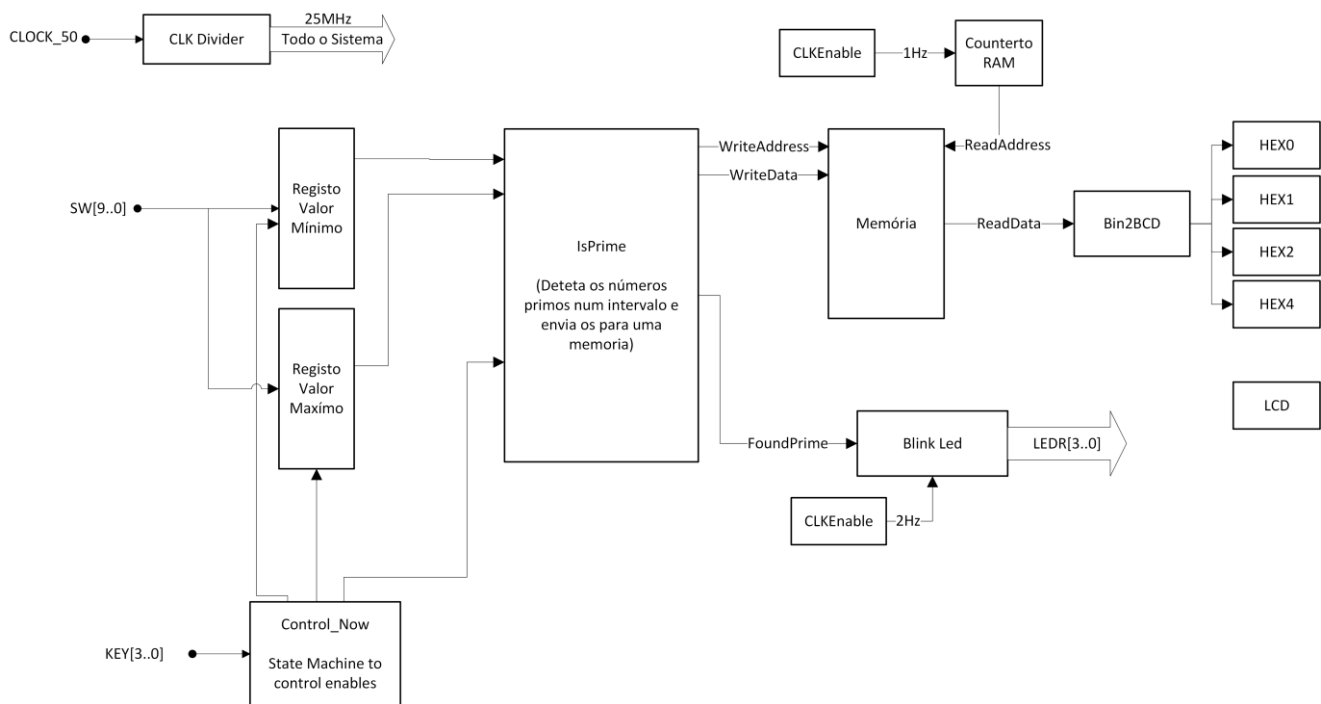


Figura 1 - Esquema dos blocos utilizados no projeto

IMPLEMENTAÇÃO

O sistema possui como *controlpath* uma máquina de estados que determina a validação dos valores mínimo e máximo do intervalo e o início da pesquisa dos números primos (Figura 2).

Nesta máquina de estados o sinal *SAVE* está ligado à KEY(1) e o *START* está ligado à KEY(2). Esta máquina é iniciada no estado *r_min*. Neste estado enquanto o *SAVE* não for '1' todos os *enables* dos blocos estão desligados. Quando o *SAVE* for '1' a máquina liga o *enable* do *LoadMin* e este valor está ligado ao registo do valor mínimo e o mesmo é guardado para depois ser utilizado pela máquina de estados *isPrime*.

O mesmo acontece para o estado *r_max*. Neste é ligado o *LoadMax* e o valor é gravado para o registo. Após isto a máquina avança para o estado *verify*. Caso o valor máximo definido anteriormente seja maior que o mínimo volta ao estado *r_max* para gravar um novo valor máximo válido. Após validação de ambos os extremos do intervalo a máquina liga o *enablePrime* que está ligado ao *enable* do Bloco *isPrime* que deteta se os valores são primos.

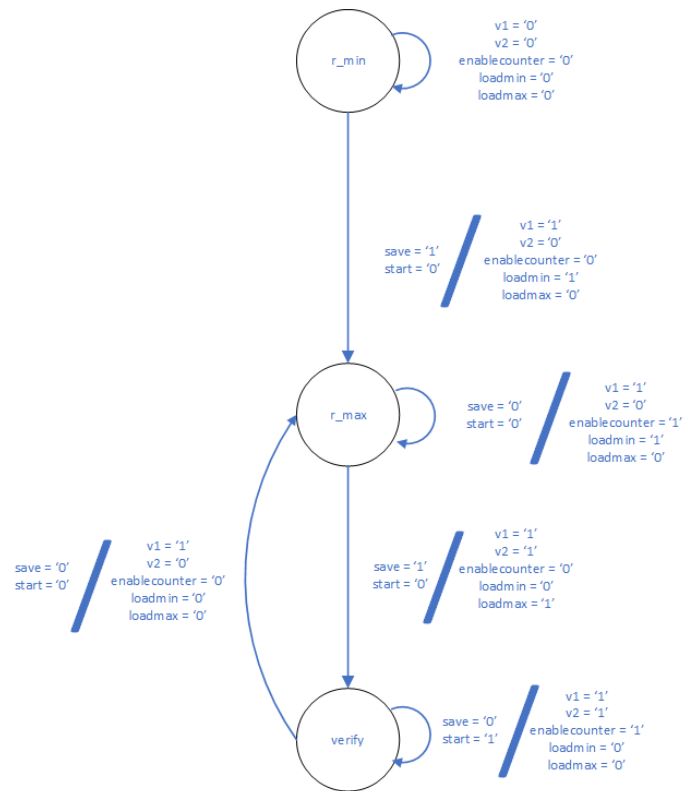


Figura 2 -Diagrama da FSM de controlo do sistema

Para testar a primalidade dos números no intervalo escolhido foi desenvolvida uma máquina de estados de Meally. Esta máquina começa no estado *s_init* em que todos os sinais de saída como *s_Done*, *s_FoundPrime*, *WriteEnable(s_svenable)*, passam a '0'. O valor mínimo é atribuído ao sinal que contém o valor que está a ser testado (*s_num*) e o valor máximo a um sinal (*s_valueMax*) que indica o último valor de teste. Caso *START* seja '1' a máquina avança para o estado *s_receiveValue*. Neste estado, se o valor for 2 ou 3 o mesmo é considerado primo (avança para o estado *s_isPrime*) se não, se for 1 ou 0, não é primo (avança para o estado *s_isnotPrime*), se for outro valor avança para o estado *s_checknum*. Aqui verifica se o valor é par, pois se for não é primo e assim avança para o estado *s_isnotPrime*, e caso não seja par testa o resto da divisão por todos os números inteiros desde 2 até metade do número (estado *s_test* incrementa o valor dos possíveis divisores, voltando ao estado *s_checknum* para realizar um novo teste de divisão cada vez que incrementa). Se o número não for primo entra no estado *s_isnotPrime*.

A partir deste estado volta ao estado *s_receiveValue* para testar o próximo número. Finalmente, se for encontrado um número primo, no estado *s_address* é devolvido um valor de endereço para guardar o valor primo na memória. À medida que vão sendo encontrados números primos o valor de endereço vai incrementando. A partir do estado *s_address* volta para o estado *s_receiveValue* para testar o próximo valor.

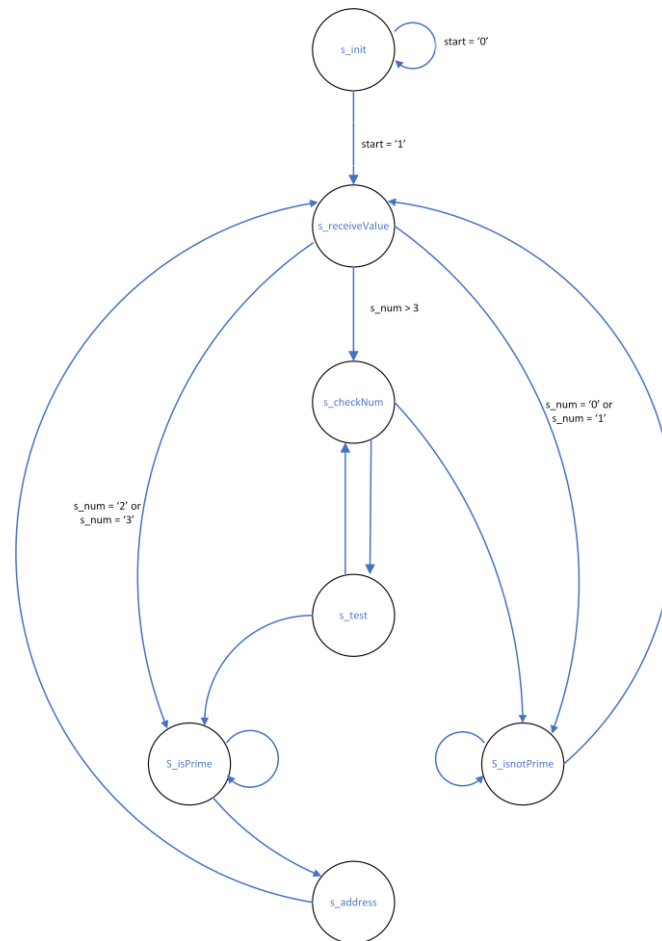


Figura 3 -Diagrama da FSM de detecção de números primos

O sistema é coordenado por:

- Um sinal de relógio (CLOCK_50)
- Dez *switches* (SW[9..0])
- Três *keys* (KEY[2..0])

E reage através dos seguintes componentes:

- Quatro LEDs vermelhos (LEDR[3..0])
- Três LEDs verdes (LEDG[2..0])
- *Displays* de 7 segmentos (HEX)
- *Display* LCD

VALIDAÇÃO

Ao longo do desenvolvimento do sistema e do processo de implementação dos blocos, foram realizadas vários *Testbenches* e *Waveforms* para verificar o funcionamento correto dos vários blocos implementados.

Foi desenvolvida, na fase 2, uma *Testbench* para simular o funcionamento da memória RAM, permitindo verificar se esta guardava os valores recebidos de forma correta. Em adição, foi desenvolvida uma *Testbench* para testar o projeto concluído.

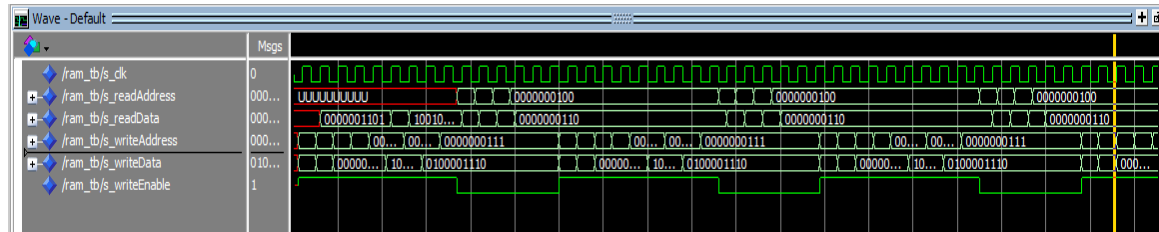


Figura 4 - *Testbench* que simula o funcionamento da memória

Foi também criado um *Waveform* para testar a identificação dos números primos.

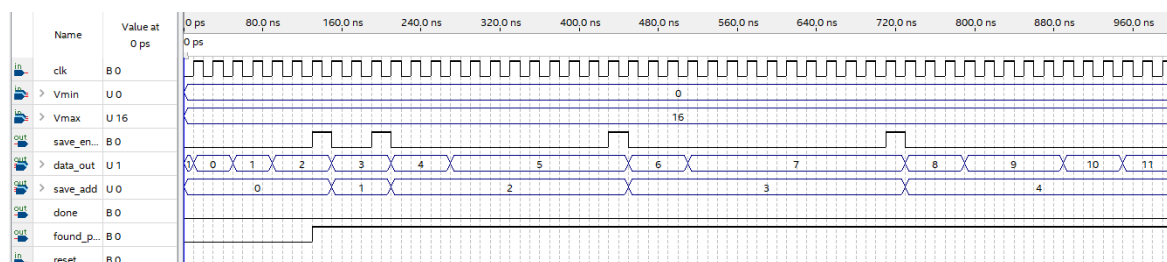


Figura 5 - *Waveform* que testa a identificação dos números primos

Assim, a validação do projeto foi efetuada através de *Testbenches* que simulam o funcionamento de vários blocos, através de *Waveforms* e através de vários testes na FPGA com diferentes intervalos de números (em que existia apenas um número primo, vários números primos ou nenhum).

CONCLUSÃO

O projeto realizado cumpre os critérios propostos pelo enunciado fornecido, tendo sido este usado como documento de orientação para organizar a implementação do mesmo.

Uma das principais conclusões retiradas foi a necessidade de um sinal de relógio com metade da frequência do CLOCK_50 (ou seja, 25 MHz) para possibilitar as operações dos blocos sem gerar conflitos temporais.

A máquina de estados de detecção de números primos (isPrime) terá sido o bloco que apresentou mais dificuldades no seu desenvolvimento. No entanto, a sua implementação foi bem sucedida e comprovada com auxílio de simulações e posteriormente verificada no funcionamento do sistema implementado na FPGA.

Este projeto permitiu-nos compreender melhor as características da linguagem VHDL e as suas possibilidades de implementação de sistemas. Através da aprendizagem da linguagem nas aulas e de pesquisa autónoma conseguimos elaborar o projeto na íntegra dentro do tempo limite, cumprindo os nossos objetivos de trabalho.

Desta forma, este trabalho permitiu-nos desenvolver um sistema que deteta números primos de forma rápida e eficaz, o que foi de encontro aos objetivos definidos no guião.

DIVISÃO PERCENTUAL DO TRABALHO

O trabalho foi distribuído de forma equitativa por ambos os elementos do grupo e foi desenvolvido de forma cooperativa por ambos. Assim, ambos colaboraram 50% para o trabalho.

MANUAL DO UTILIZADOR

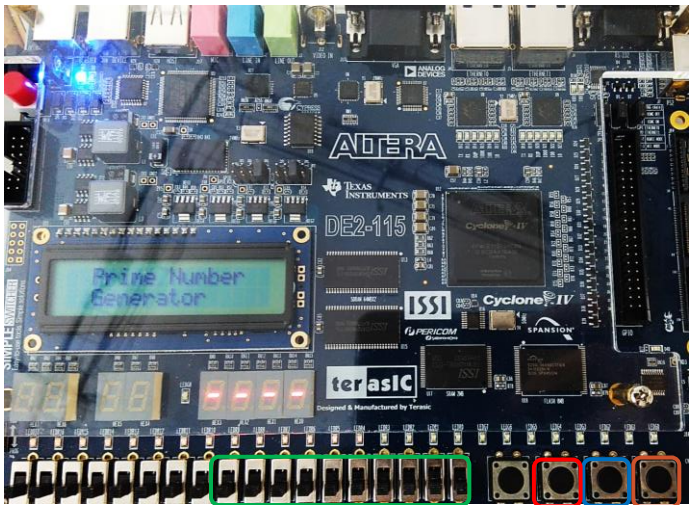
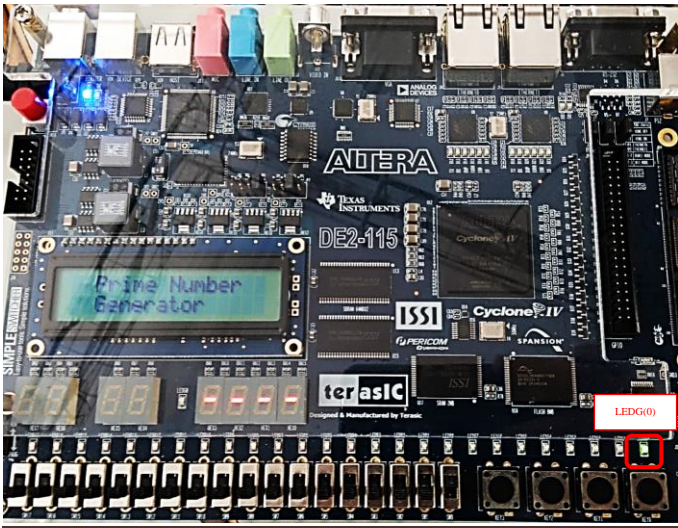
O sistema contém um **Master Reset** **KEY(0)** que reinicia o sistema.

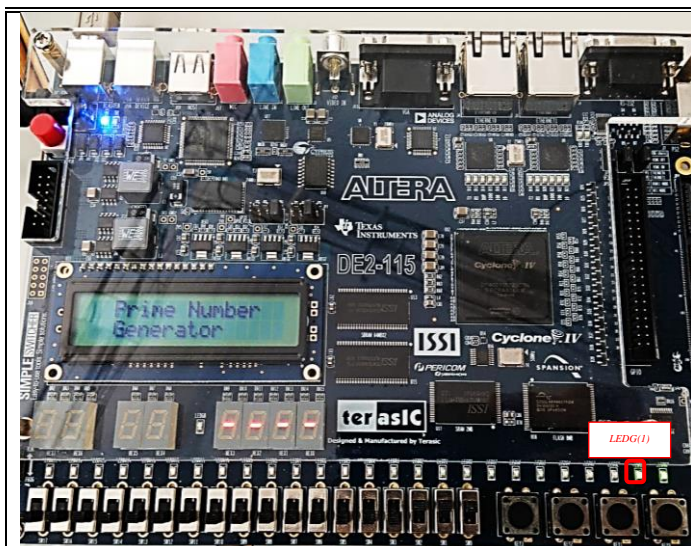
Para a introdução do intervalo no sistema:

- O menor valor é introduzido através dos **SW(9..0)**
- Para definir o valor introduzido premir o botão **KEY(1)**
- O LEDG(0) acende para mostrar que o valor é válido
- O maior valor é introduzido através dos **SW(9..0)**
- Para definir o valor introduzido premir o botão **KEY(1)**
- O LEDG(1) acende para mostrar que o valor é válido. Se este LED não acender é porque o valor máximo introduzido é inválido, pois é maior que o mínimo
- Para iniciar a pesquisa dos números primos (após premir o botão **KEY(2)**)
- Quando a pesquisa estiver concluída o **LEDG(2)** acende-se
- Se foi encontrado algum número primo, acende-se o **LEDR(0)**.

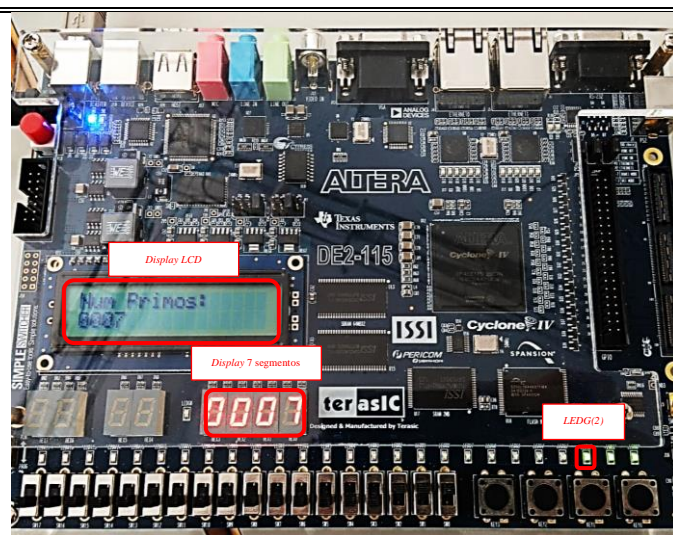
Se no intervalo escolhido não tiverem sido encontrados números primos os quatro LEDs vermelhos **LEDR(3..0)** irão piscar ao ritmo de 2Hz.

Após os valores terem sido armazenados estes vão ser apresentados nos *displays* de 7 segmentos (extra: *Display LCD*) de forma cíclica. Para parar este ciclo premir o (**Master Reset**) **KEY(0)**.

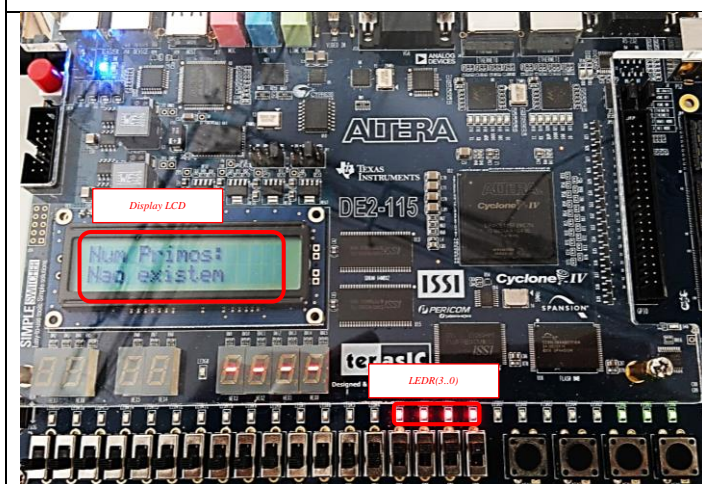
 <p>Inserir o valor Start Save Reset</p>	 <p>LEDG(0)</p>
<ol style="list-style-type: none">1) Iniciação da máquina ou após um <i>reset</i>.2) Inserir o valor mínimo nos SW[9..0].3) Após definir o valor mínimo clicar no <i>Save</i>.	<ol style="list-style-type: none">4) Após definir o valor mínimo o LEDG(0) acende5) Inserir o valor máximo nos SW[9..0].6) Após definir o valor máximo clicar no <i>Save</i>.



7) Enquanto o valor máximo não for válido ($max > min$) o LEDG(1) não acende e a máquina fica à espera que seja introduzido um valor válido.



8) Quando a pesquisa é concluída o LEDG(2) acende e os valores primos encontrados são visualizados nos Displays de 7 Segmentos e no LCD à frequência de 1Hz.



9) Se não forem encontrados números primos os LEDR(3..0) piscam à frequência de 2 Hz.