

Máquina de Calcular com Memória

Departamento de Eletrónica, Telecomunicações e
Informática

Laboratório de Sistemas Digitais

Grupo 1, Turma P9
Luís Costa, Vasco Sousa
(92996, 93049)

2018-2019

INTRODUÇÃO

Para este projeto foi implementada uma máquina de calcular com memória com capacidade de fazer cinco operações com operandos de até 8 bits, em complemento para dois, ou seja $[-128, 127]$ em decimal.

O utilizador pode definir dois operandos e de seguida escolher a operação a realizar. Estão disponíveis as operações de adição, subtração, multiplicação, divisão e raiz quadrada.

Após esta etapa o utilizador pode visualizar no display de sete segmentos o resultado final.

De acordo com o que foi requisitado para projeto, este divide-se em três fases diferentes.

- Fase 1 - Implementação das operações de soma, subtração, multiplicação e divisão (inteira).
- Fase 2 - Implementação da operação da raiz quadrada (Inteira).
- Fase 3 - Implementação da memória da máquina de calcular.

Neste projeto não foi implementado no display de 7 segmentos, a memória da raiz em SlideShow, esta é apresentada incorretamente, reforçando a ideia que o calculo da raiz está totalmente funcional. Além disto, para a complicação deste programa, remova a promoção de critical errors.

ARQUITECTURA DO SISTEMA

Calculator (*Top Level Entity*)

Módulo responsável pelo controlo do sistema, utilizando a máquina de estados principal, que faz gestão dos operandos introduzidos pelo utilizador e qual a operação a realizar pelo mesmo.

Debouncer

Módulo que corrige os problemas mecânicos dos botões através da limpeza do ruído do sinal, utilizado para que seja possível o correto funcionamento destes.

BinTo7Segments

Módulo responsável por converter um número binário em BCD e posteriormente apresentar no display de 7 segmentos o valor em decimal com sinal.

Scroll

Módulo responsável pela transição dos displays de 7 segmentos, apresentando a informação guardada na memória (RAM).

Save to Ram

Módulo responsável pela escrita e leitura da memória RAM dos operandos, operação e resultado, das últimas três operações realizadas pelo utilizador, com o respetivo sinal.

Este módulo utiliza um incremento calculado no Calculator(Top level), para guardar nas posições de 0 a 19 os operandos, com o respetivo sinal, operador, resultado, sinal do resultado e overflow, já descodificados também neste módulo para os 7 segment decoders, visto que a informação introduzida na RAM já são os valores descodificados.

A leitura também ocorre neste módulo, devolve para o componente Scroll sequencialmente os valores de todas as posições da RAM utilizadas.

Para a descodificação recorremos ao load do descodificador de binário para 7 segments, que depois de calcular todos os valores para os segmentos, salva sequencialmente na RAM.

Clock Divider

Módulo responsável pela divisão do sinal de clock. Este possibilita o aumento ou redução da frequência de um determinado clock.

Add core/Sub core

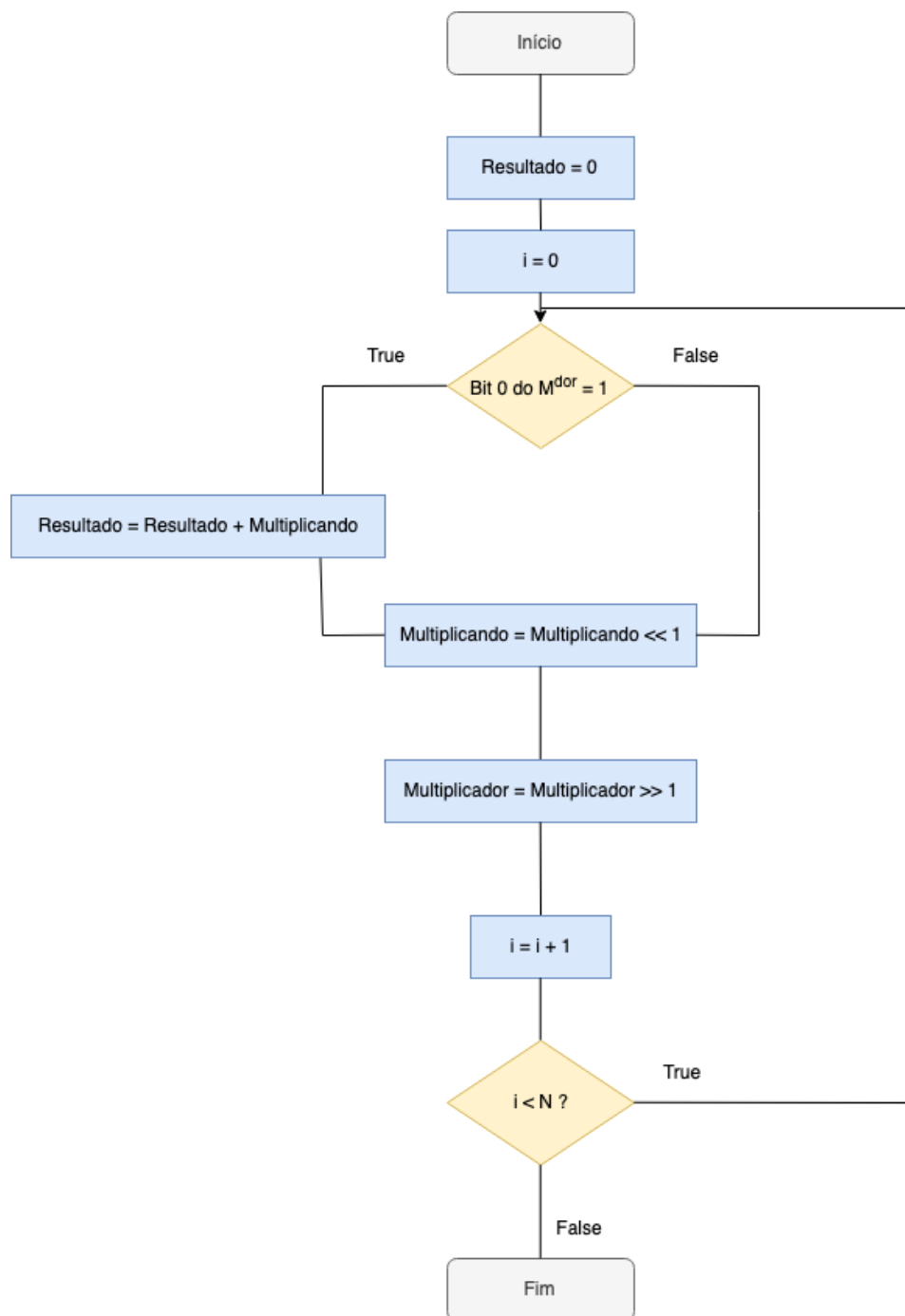
Módulos responsáveis por realizar as operações de soma e subtração, respetivamente. Ambos fazem uso do pacote *numeric_std* da biblioteca *IEEE*. Para além do resultado, também calcula, o sinal do mesmo, e o sinal de overflow.

IterMultCore

Módulo responsável por fazer a multiplicação de uma forma iterativa.

Foi utilizado o algoritmo fornecido pela UC, para o cálculo do resultado, sem sinal, sendo este calculado no bloco CheckSigned.

Para a implementação foi utilizado 2 registos e um acumulador, para o multiplicando, multiplicador e resultado respetivamente.



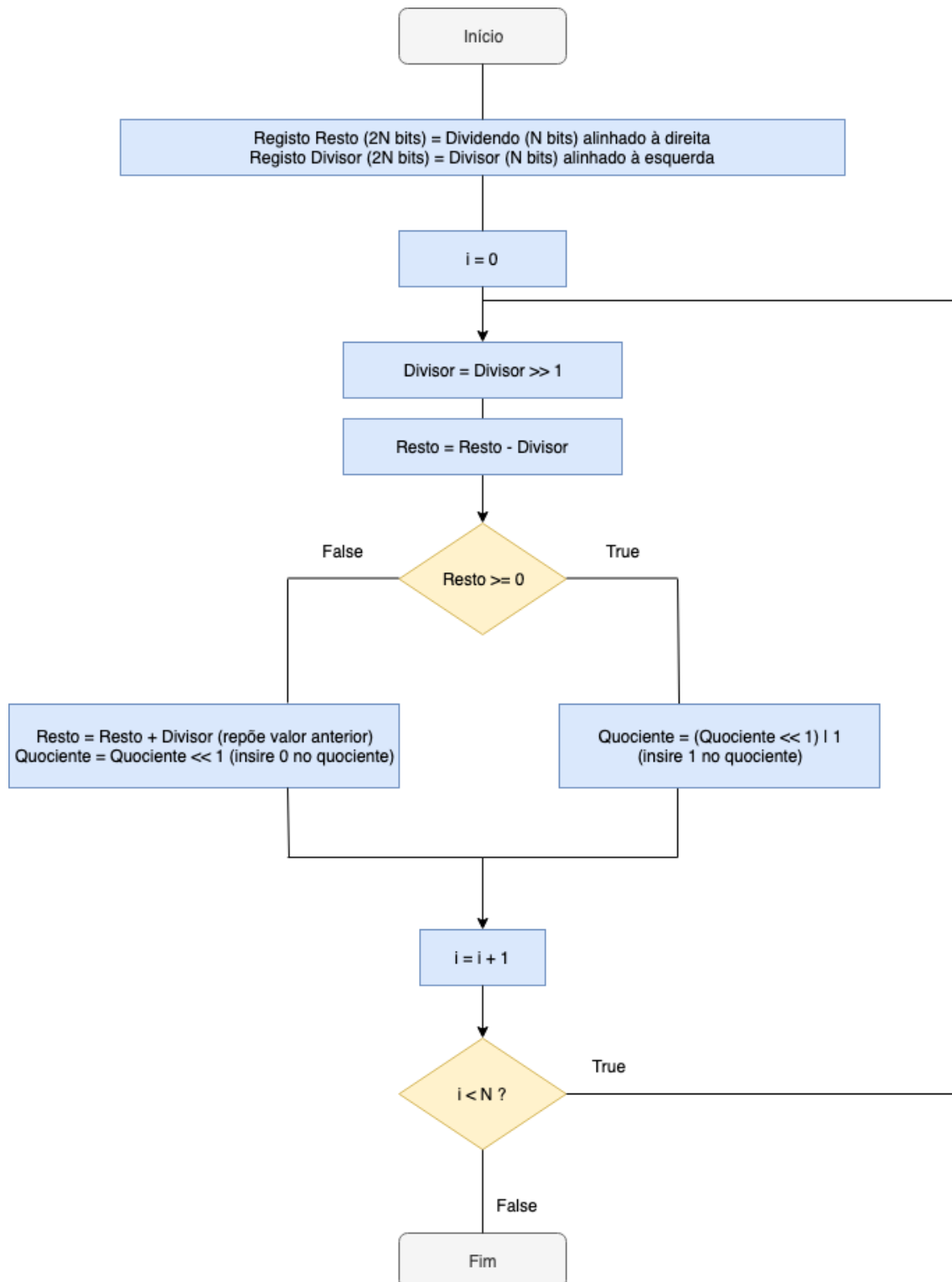
IterDivCore

Módulo responsável por fazer a divisão de uma forma iterativa.

Como na multiplicação, foi utilizado o algoritmo fornecido pela UC para a divisão inteira, para o cálculo do resultado, sem sinal, sendo este calculado no bloco CheckSigned.

Para a implementação foi utilizado 2 registos e um acumulador, para o Divisor, Quociente e resto respetivamente.

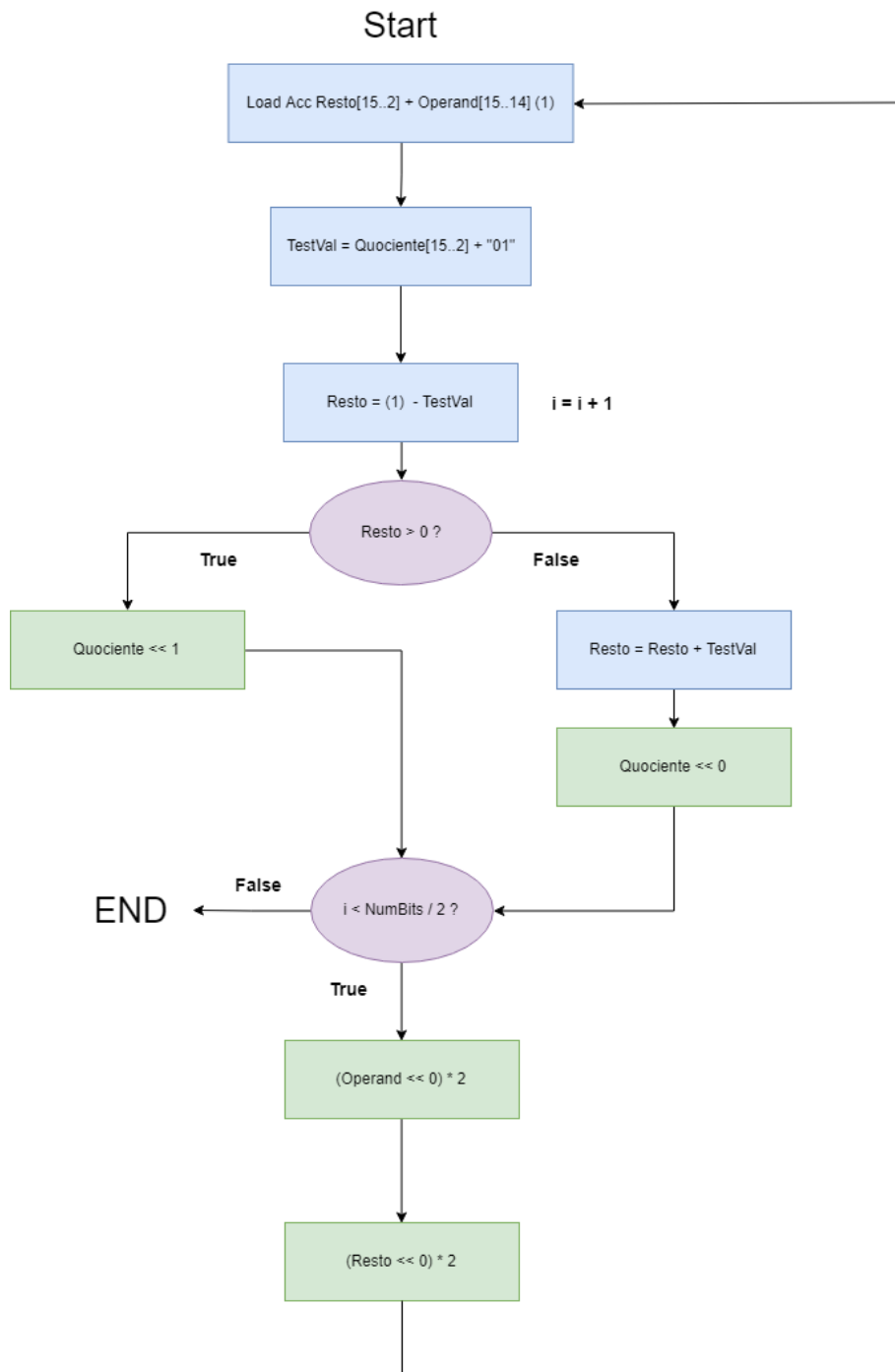
Saliento que o overflow por divisão por 0 nunca ocorre, pois o próprio sistema bloqueia o cálculo do resultado, quando um dos operandos é 0.



IterSqrtCore

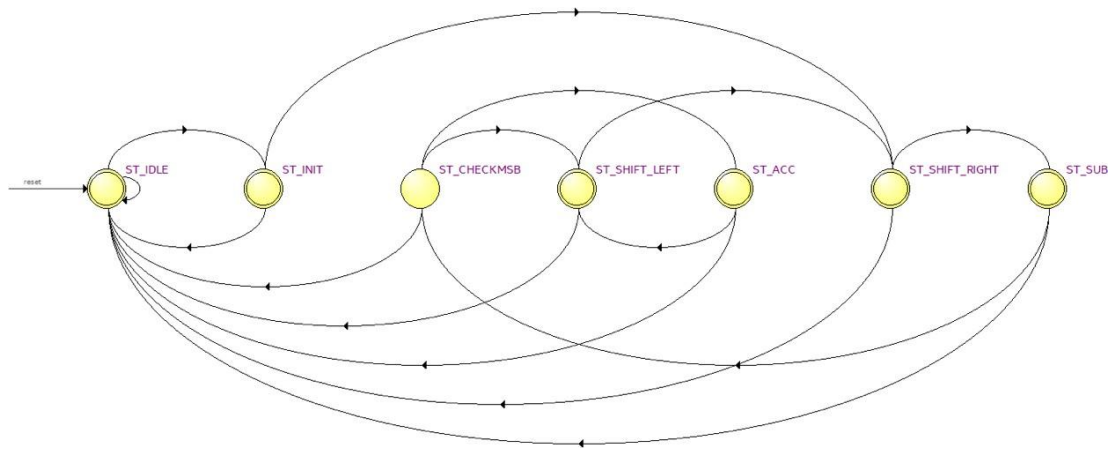
Módulo responsável pelo cálculo da raiz quadrada de um operando de 16 bits através da utilização de um algoritmo conhecido como : “Non Restoring Square Root Algorithm”.

Este algoritmo encontra-se descrito no diagrama abaixo, sendo que tanto os valores introduzidos pelo utilizador como o resultado se encontram unsigned.

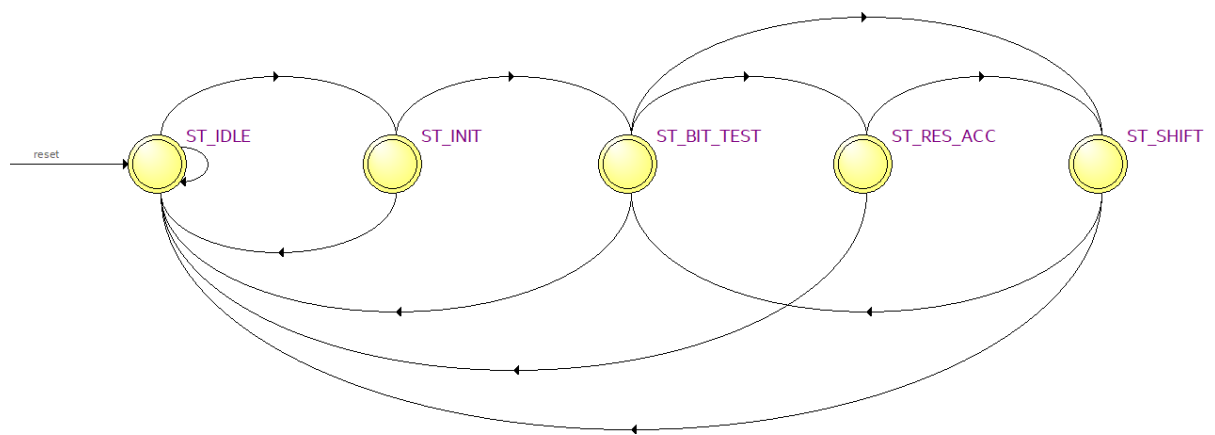


MÁQUINAS DE ESTADO

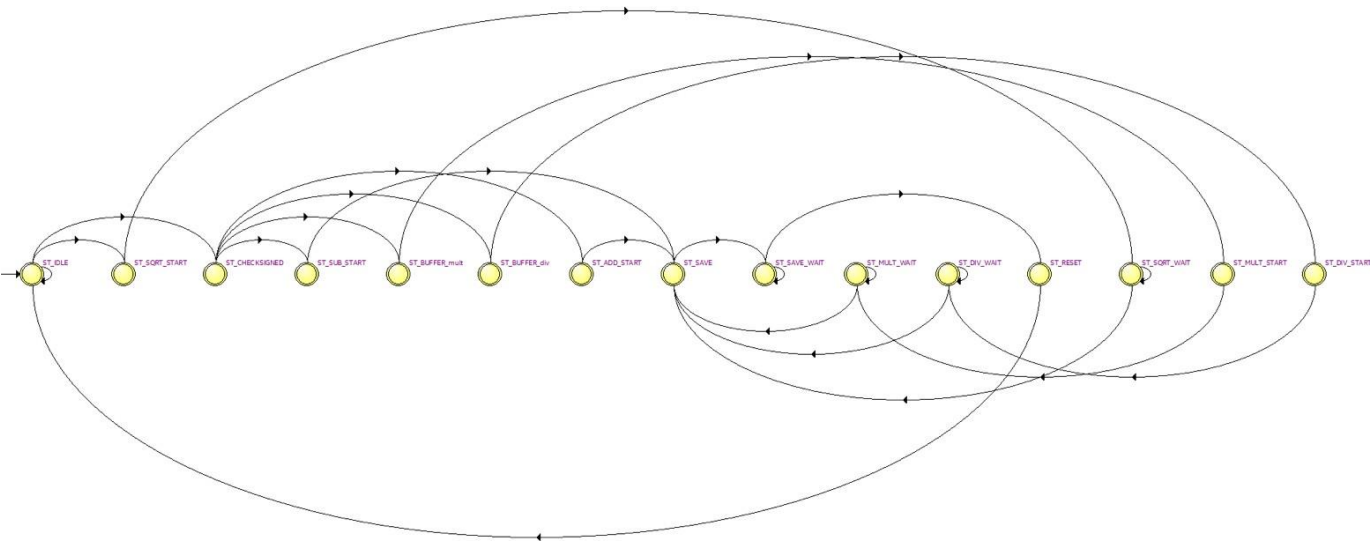
Divisão



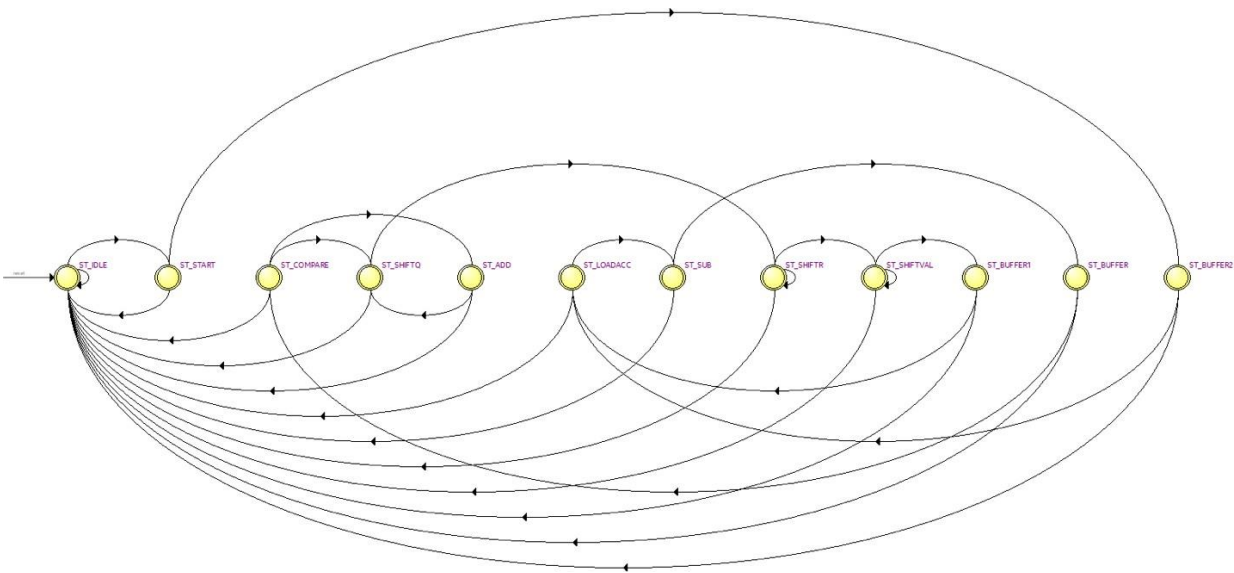
Multiplicação



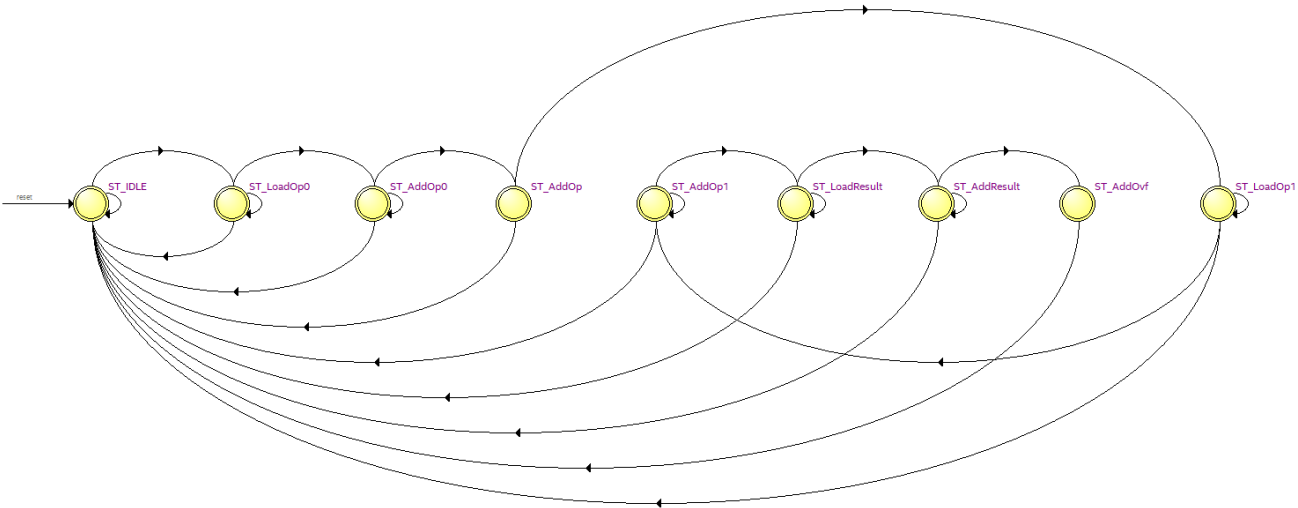
Calculadora (Top Level)



Raiz quadrada

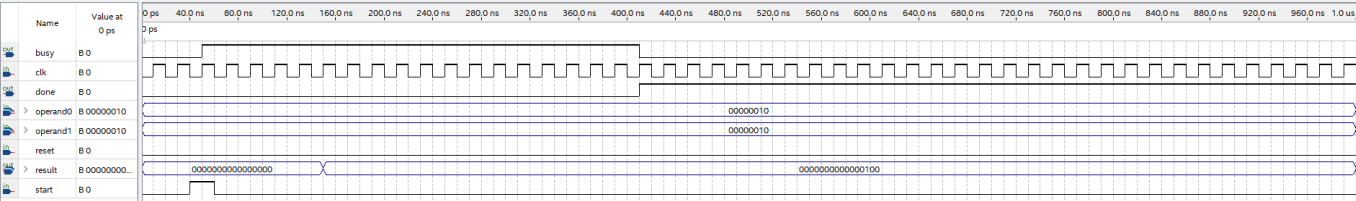


SaveToRam

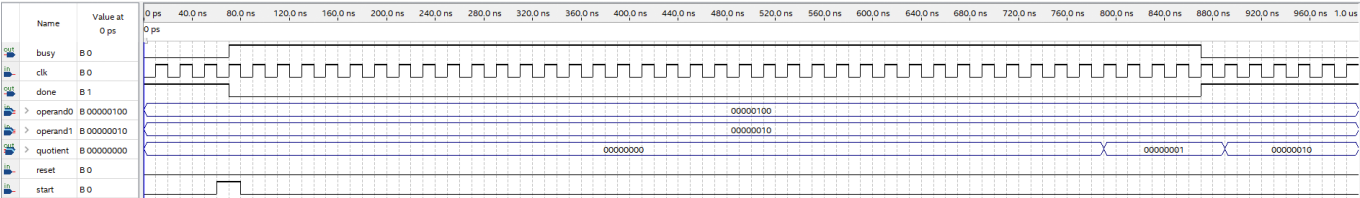


VALIDAÇÃO

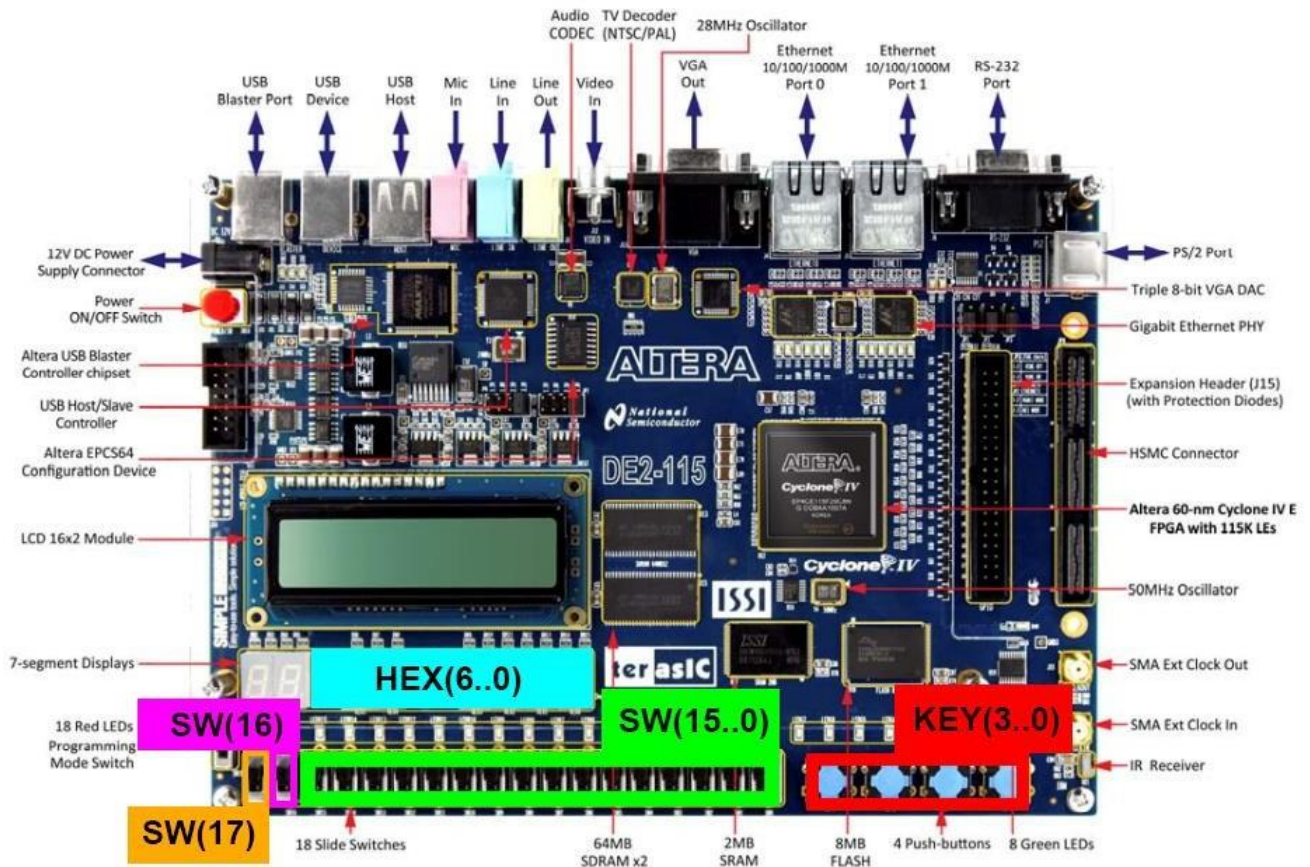
Multiplicação



Divisão



MANUAL DE UTILIZADOR



SW(15..0) – Permite selecionar os operandos, SW(15..8) permite selecionar o primeiro operando (8 bits), SW(7..0) permite selecionar o segundo operando, no caso da raiz, SW(15..0) é o operando final.

KEY(3..0) – Permite selecionar as operações adição [KEY(0)], subtração [KEY(1)], multiplicação [KEY(2)], divisão [KEY(3)].

SW(16) – Permite selecionar a operação da raiz quadrada, calculada por cada flip do SW(16).

SW(17) – Mostra as últimas três operações realizadas, no seguinte formato :
Operand0 _ Operação _ Operand1 _ Resultado _ Overflow(1bit)

HEX(6..0) – Mostra o resultado da operação em decimal ou slide das ultimas 3 operações

CONCLUSÃO

É importante salientar que o projeto tem algumas diferenças relativamente ao que estava descrito na milestone intermédia pois, à medida que o projeto foi desenvolvido, foi verificado que seriam necessários alguns módulos extra. Em segundo lugar, em relação à divisão do trabalho, podemos definir que cada um dos elementos do grupo trabalhou de forma semelhante neste projeto, pelo que Luís Costa tem 55% de influência no trabalho final e Vasco Sousa 45%. Concluindo, este trabalho permitiu aprofundar o conhecimento sobre a programação em VHDL e arquiteturas de máquinas de estados.

AGRADECIMENTOS

É deixado o agradecimento aos autores dos slides das aulas teórico-práticas pois foram utilizados os algoritmos da divisão e multiplicação presentes nos mesmos.