

Projeto em VHDL de uma ULA de 32 bits

Daniel L. Murta¹, Hugo S. Almeida², Ricardo P. de Andrade³, Vitor Hugo F. de Paula⁴

¹Instituto de Ciências Exatas e de Informática – Pontifícia Universidade Católica de Minas Gerais
Caixa Postal 1.686 –CEP 30535.901 - Belo Horizonte - Minas Gerais – Brazil

{daniel.murta,hugo.almeida,flavio,ricardo.andrade.1280935,vhfpaula}@sga.pucminas.br

Abstract. *The Logica and Arithmetic Unit (ULA), which is a digital circuit, fundamental to the central processing unit (CPU), that performs arithmetic and boolean operations, thus being essential for computing, since computers and electronic equipment in short is used to perform all and any computational activity.*

For this reason this article proposes the development of a 32-bit ULA, being implemented in the VHDL description language

Resumo. *A Unidade Logica e Aritmetica(ULA), que é um circuito digital,fundamental da unidade central de processamento (UCP), que realiza operações de aritméticas e booleanas, assim sendo essencial para a computação, pois o computadores e equipamentos eletrônicos em suma utiliza-se para a realização de toda e qualquer atividade computacional.*

Por este motivo este artigo propõe o desenvolvimento de uma ULA de 32 bits, sendo está implementada na linguagem de descrição VHDL

1. Introdução

Este artigo tem como objetivo a aplicação de todos os conhecimentos adquiridos sobre a linguagem VHDL e a forma como os circuitos computacionais funcionam, desta forma, utilizamos como objetivo de prática a criação de uma ULA de 32 bits, sendo que cada uma destas contém cinco tipos de respostas. Contudo, este artigo vai abordar com uma maior especificidade cada um dos pontos, deste a parte teórica que utilizamos até os resultados proporcionados pela interface que compilou os códigos em VHDL.

2. Trabalhos Correlatos

Nesta seção, trabalhos correlatos que abordam propostas de ULA's

Em [Torok and Cappelatti 2011], os autores propõem um criação de ULA de 4 bits, na qual é dividida em dois blocos: um lógico e outro aritmético possuindo a operação de and, or(lógico) e adição e subtração(aritmético). Assim eles constroem a aplicação em vhdL passo a passo

Por sua vez em [Sassi 2013] propõe que pelo fato de que normalmente, a ULA é o componente de maior consumo em um processador, o que a torna alvo de diversos estudos sobre técnicas para redução de consumo. Este trabalho apresenta um resumo sobre consumo de potência em circuitos digitais CMOS e as principais técnicas para sua redução, assim como os fundamentos para o projeto de ULAs

Por fim em [Nascimento et al. 2006] propõe uma ULA reversível (ULA-R) básica, que pode ser utilizada tanto para construir computadores clássicos reversíveis quanto

computadores quânticos. Descrevemos os principais módulos da unidade reversível e discutimos as implicações de algumas aplicações quânticas em seu funcionamento

3. Proposta de Arquitetura para ULA

A ULA de 32 bits tem 1 ULA cabeça de 1 bit, 30 ulas corpo de 1 bit, e uma ula rodapé de 1 bit. Cada ula de 1 bit é constituída de um multiplexador 2x1, um multiplexador 4x1, uma operação or, uma operação and, e um somador completo, que é constituído de 2 meio somadores. Contudo, a ULA rodapé contém um elemento a mais que é o detector de overflow, logo abaixo podemos ver exemplos visuais das ulas e seus componentes.

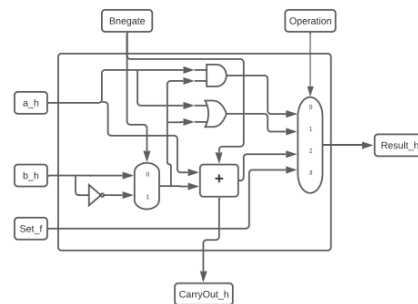


Figura 1. Exemplo visual da ULA HEAD.

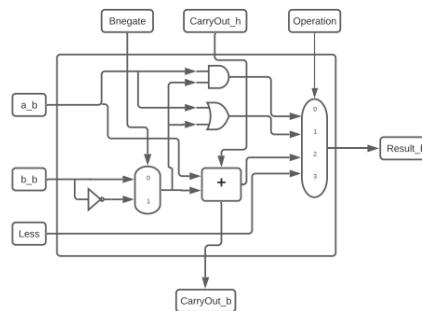


Figura 2. Exemplo visual da ULA BODY.

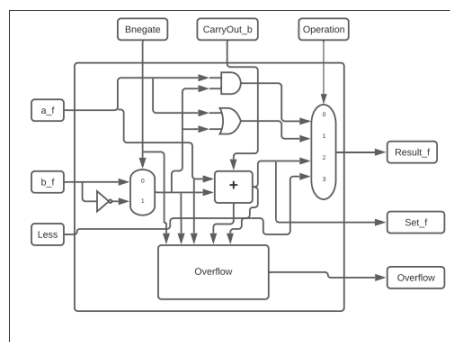


Figura 3. Exemplo visual da ULA FOOTER.

Primeiramente uma ula cabeça pega um valor Bn que sai do decodificador 3x3, joga no multiplexador 2x1, que pode inverter o valor de B, caso o Bn seja igual a 1.

Depois pega a saída do multiplexador 2x1 e o valor A e faz as operações de soma, OR e AND.

O meio somador executa uma operação soma a and not b com um or not a and b e um carry a and b, o somador completo faz essa soma, e leva o resultado da primeira soma para ser o a da segunda, executando uma operação or com o primeiro carry e o segundo carry. O somador faz subtração e adição dependendo do valor de Bn. Os resultados são lançados em um multiplexador 4x1 que decodifica 00 no AND previo, 01 no OR, 10 na soma e o resto no less.

O carry out da ula cabeça é passado para ser o carry in da a ula corpo, enquanto o Less da ula cabeça era o resultado da soma da ula rodapé. A ula corpo e rodapé são semelhantes a ula cabeça, com as diferenças faladas anteriormente, e o fato que a ula rodapé performa um detector de overflow.

Overflow ocorre com respeito ao tamanho do tipo de dados que deve acomodar o resultado. Overflow indica que o resultado foi muito grande ou muito pequeno para caber no tipo de dados. Quando dois números signed são adicionados, overflow é detectado se ambos os operandos forem positivos e o resultado negativo ou ambos os operandos forem negativos e os resultados positivos quando dois números unsigned são adicionados overflow acontece se existe um carry out do bit mais a esquerda

Adição de um bit $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$ e $1 + 1 = 10$.

Última linha indica que temos um output de carry, ou seja, uma quantidade de um bit não pode acomodar(1+1) portanto um tipo de dados maior é requerido para 1+1 suceder

4. Avaliação dos Resultados

Após a criação dos códigos e a compilação feita por meio da plataforma EDA Plauground obtemos os resultados esperados, em todas as operações, além disso conseguimos resultados visuais tendo em vista que a plataforma disponibiliza esse tipo de resultado para cada uma ULA presente neste trabalho, os mesmos podem ser vistos logo abaixo.

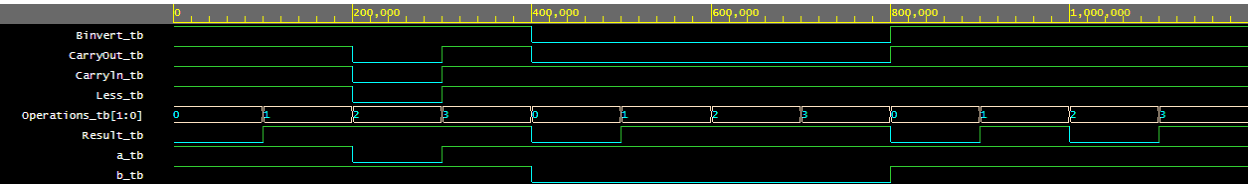


Figura 4. Resultado da ULA HEAD.

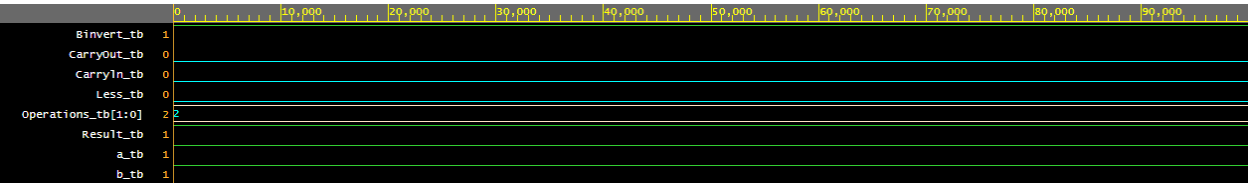


Figura 5. Resultado da ULA BODY.



Figura 6. Resultado da ULA FOOTER.

Além disso nós, após juntarmos todas as ULAs em um único design para montar desta forma a ULA de 32 bits, obtemos o resultado mostrado abaixo, é importante ressaltar que esse resultado também foi retirado da plataforma EDA Playground.

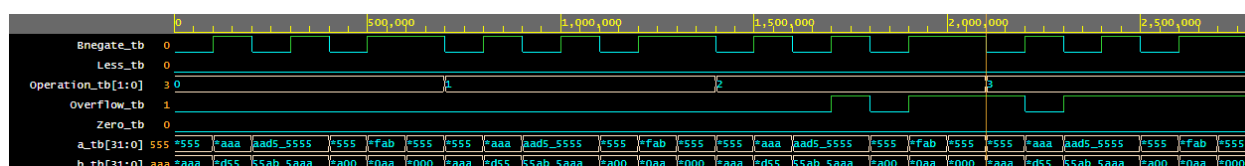


Figura 7. Testbench da entidade global constituída pelas 32 ULA's de 1bit.

4.1. Metodologia de Avaliação

Como metodologia de avaliação calculamos aritmeticamente os resultados das operações e realizamos um testbench para ver se o resultado na prática correspondia ao esperado. Desta forma, podemos perceber logo abaixo que os resultados presentes na tabela "Resultados Calculados Aritmeticamente" está batendo com os resultados obtidos na prática que estão na tabela "Resultados", mostrando assim que os nossos códigos estão satisfazendo a necessidade da ULA proposta.

Resultados Calculados Aritmeticamente

INPUT	AND	OR	SOM/SUB	LET	CARRY/SOM	CARRY/SUB
a = 0 / b = 0	0	0	0	0	0	0
a = 1 / b = 0	0	1	1	0	0	0
a = 0 / b = 1	0	1	1	0	0	1
a = 1 / b = 1	1	1	0	0	1	0

Resultados Obtidos

INPUT	AND	OR	SOM/SUB	LET	CARRY/SOM	CARRY/SUB
a = 0 / b = 0	0	0	0	0	0	0
a = 1 / b = 0	0	1	1	0	0	0
a = 0 / b = 1	0	1	1	0	0	1
a = 1 / b = 1	1	1	0	0	1	0

5. Conclusões

Foram obtidos com sucesso os resultados esperados na simulação de um circuito computacional com uma ULA de 32 bits, apesar de notarmos uma dificuldade tanto no entendimento das funcionalidades quanto na implementação dos códigos.

Referências

- Nascimento, A. L., Kowada, L. A. B., and de Oliveira, W. R. (2006). *Uma Unidade Lógica e Aritmética Reversível L*. Universidade de Pernambuco (UPE) – Recife, 2006.
- Sassi, A. B. (2013). *Projeto de uma ULA de inteiros e de baixo consumo em tecnologia CMOS*. São Carlos, 2013, São Carlos - SP.
- Torok, D. L. and Cappelatti, E. A. (2011). *PRATICANDO VHDL*. Novo Hamburgo : Feevale, 2010., Novo Hamburgo - Rio Grande do Sul.