Trabalho de Organização e Arquitetura de Computadores I

Matthias Oliveira de Nunes

FACIN – Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

Porto Alegre – RS – Brazil

matthiasnunes@gmail.com

Abstract. This meta-paper describes a simple 5 bit arithmetic and logic unity using structured VHDL.

Resumo. Este meta-artigo descreve uma simples unidade lógica e aritmética de 5 bits usando VHDL estruturado.

1. Introdução

Foi implementado em VHDL uma ULA que faz 8 operações: Subtração, soma, multiplicação por 2, divisão por 2, complemento, não-ou lógico, ou exclusivo lógico, não-e lógico.

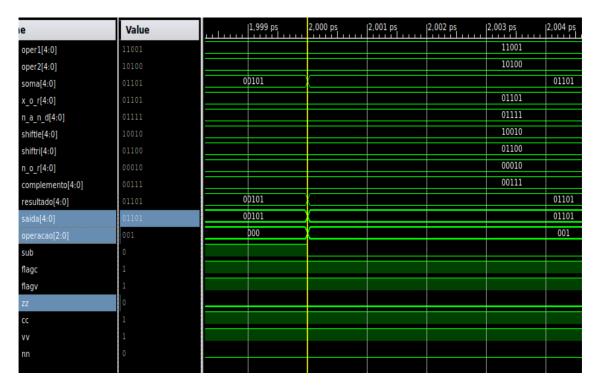
2. Formas de Onda

2.1 Subtração

ne	Value	999 ps	1,000 ps	1,001 ps	1,002 ps	1,003 ps	1,004 ps
oper1[4:0]	11001	00000					11001
oper2[4:0]	10100	00000	X				10100
soma[4:0]	00101	00000					00101
x_o_r[4:0]	01101	00000					01101
n_a_n_d[4:0]	01111	11111					01111
shiftle[4:0]	10010	00000					10010
shiftri[4:0]	01100	00000					01100
n_o_r[4:0]	00010	11111					00010
complemento[4:0]	00111	00000					00111
resultado[4:0]	00101	00000					00101
saida[4:0]	00101	00000					00101
operacao[2:0]	000					000	
sub	1						
flagc	1						
flagv	1						
zz	0						
сс	1						
VV	1						
nn	0						

Coloquei antes os operandos como "00000" para mostrar que a flag "zz" estáva indicando que a saída era "0". Após isso os operandos recebem os valores "11001", "10100" e o resultado muda para "00101".

2.2 Soma



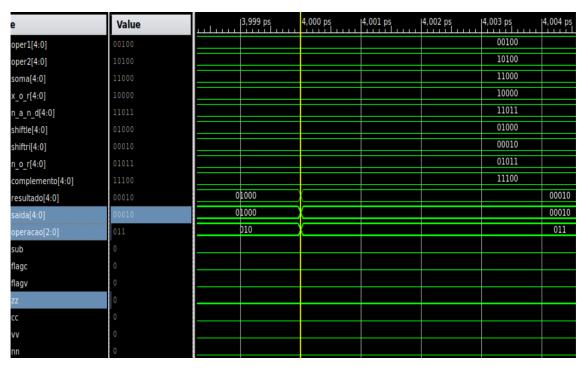
A operação muda para "001", indicando uma soma entre oper1 e oper2. A saída muda para "01101". Os flags "cc" e "vv" ativaram indicando que teve um carry e um overflow.

2.3 Multiplicação por 2

ne .	Value	2,999 ps	3,000 ps	3,001 ps	3,002 ps	3,003 ps	3,004 ps
oper1[4:0]	00100	11001	X				00100
oper2[4:0]	10100					10100	
soma[4:0]	11000	01101					11000
x_o_r[4:0]	10000	01101					10000
n_a_n_d[4:0]	11011	01111					11011
shiftle[4:0]	01000	10010	X				01000
shiftri[4:0]	00010	01100	X				00010
hiftri[4:0]	01011	00010	X				01011
complemento[4:0]	11100	00111	X				11100
resultado[4:0]	01000	01101					01000
saida[4:0]	01000	01101					01000
operacao[2:0]	010	001					010
sub	0						
flagc	0						
flagv	0						
ZZ	0						
СС	0						
vv	0						
nn	0						

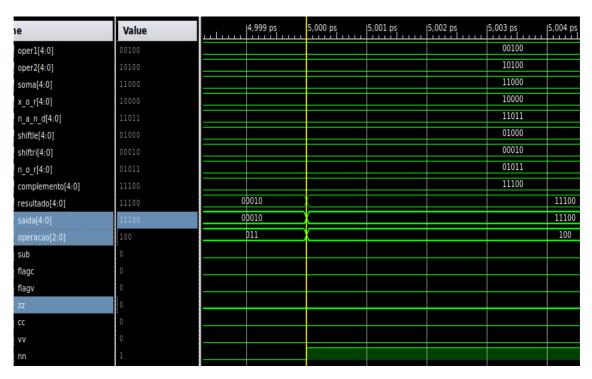
O oper1 mudou para "001", a operação mudou para "010" e a saída para "0100" já que foi realizado um shift para a esquerda.

2.4. Divisão por 2



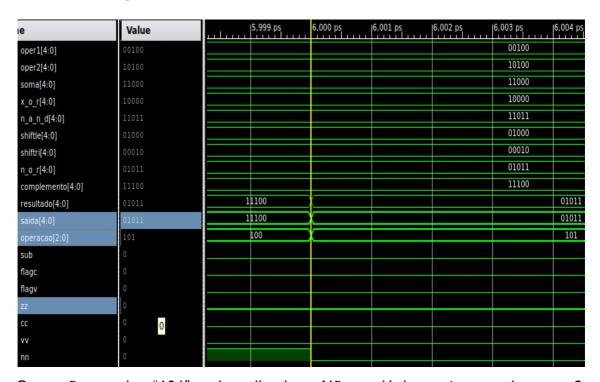
Operação recebe "011" e a saída muda para "00010" já que o oper1 sofreu um shift para a direita.

2.5 Complemento



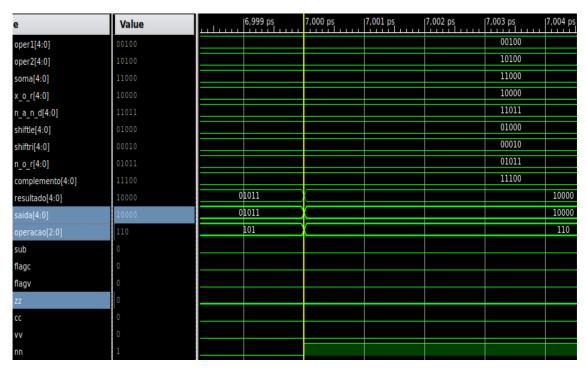
Operação recebe "100" e faz o complemento de 2 do oper1 tendo como saída "11100". O flag nn ativa pois poderia ser considerado um número negativo.

2.6 Não-ou lógico



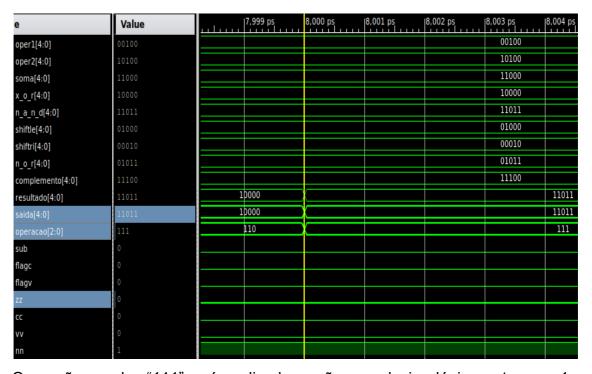
Operação recebe "101" e é realizado o Não-ou lógico entre oper1 e oper2, resultando na saída "01011".

2.7 Ou exclusivo lógico



Operação recebe "110" e é realizado o ou exclusivo lógico entre oper1 e oper2, resultando na saída "10000".

2.8 Não-E lógico



Operação recebe "111" e é realizado o não-e exclusivo lógico entre oper1 e

oper2, resultando na saída "11011".

3.0 Bibliografia utilizada

d'Amore, Roberto "VHDL Descrição e Síntese de Circuitos Digitais".