

ULA

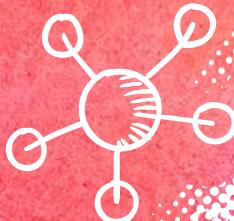
Unidade Lógica Aritmética

Introdução

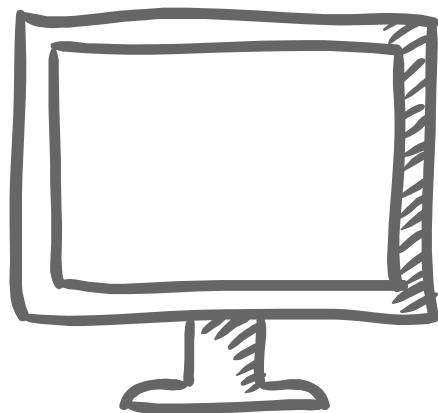
A **ULA** é uma peça fundamental da unidade central de processamento (UCP), e até dos mais simples microprocessadores. É na verdade, uma "*grande calculadora eletrônica*"



A ULA executa as principais operações lógicas e aritméticas do computador. Ela soma, subtrai, divide, determina se um número é positivo ou negativo ou se é zero. Além de executar funções aritméticas, uma ULA deve ser capaz de determinar se uma quantidade é menor ou maior que outra e quando quantidades são iguais.



1.
Entradas
e saídas.





ALU I/O

Nome	Tipo	Largura	Função
ALU-A	Entrada	8 bits	Operando A
ALU-B	Entrada	8 bits	Operando B
ALU-OP	Entrada	3 bits	Operação
ALU-OUT	Saída	8 bits	Resultado
FLAGS	Saída	4 bits	Flags de Resultado

Operações

Código	Operação
000	A + B
001	A + 1
010	not A
011	nop
100	passa B
101	A or B
110	A and B
111	passa A



Resumo das saídas

BIT	FLAG
bit (3)	carry
bit (2)	overflow
bit (1)	negativo
bit (0)	zero

Zero: C tem valor 0 em todos os bits;

Negativo: C tem valor 1 no bit mais significativo;

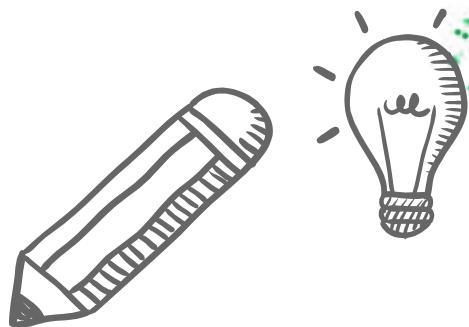
Carry: carry_out da soma A+B ou A+1 tem valor 1.

Overflow: A & B são positivos (0 no bit mais significativo) e C é negativo (1 no bit mais significativo);

2.

Construção da ULA

- × Etapas iniciais
- × Como ocorreu o processo
- × Principais dificuldades



Etapas



Definição das portas utilizadas e implementação

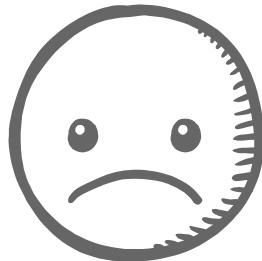
Desenvolvimento elétrico das portas

Testes das portas e montagem da ULA (parte elétrica)

Layout das portas utilizadas

Testes DRC e LVS.
Montagem da ULA (Layout)

Dificuldades



Servidor

Algumas falhas quanto ao acesso ao servidor. Seja remotamente (homework) ou na própria universidade.

Tempo

Administrar as fases do projeto de acordo com as semanas e as aulas disponíveis.

Projeto

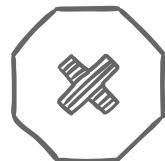
Entender a lógica e ideia do trabalho proposto

Metodologia

Não adotar uma metodologia para o andamento do projeto, estabelecendo metas e prazos da equipe.

Dimensões

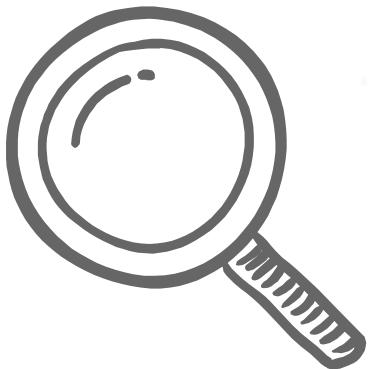
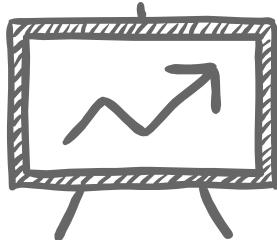
Conseguir chegar a uma montagem padrão de layout para montarmos a ULA sem perdas de espaço no CI.

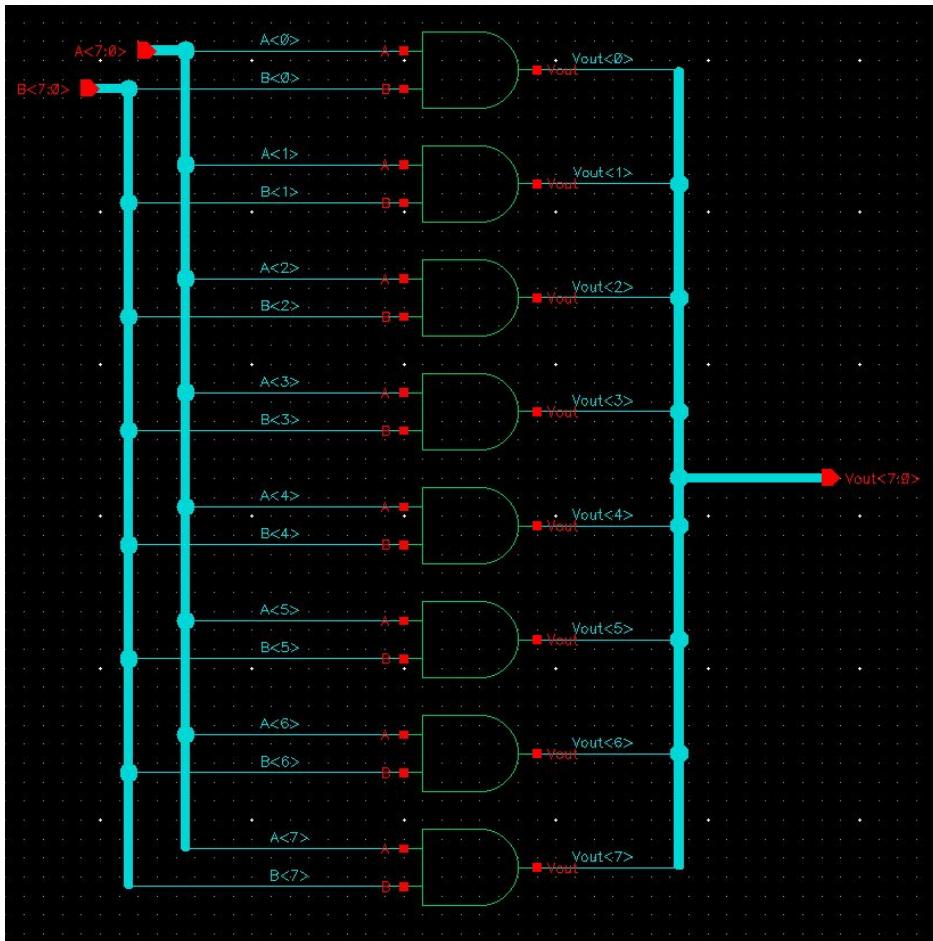


3.

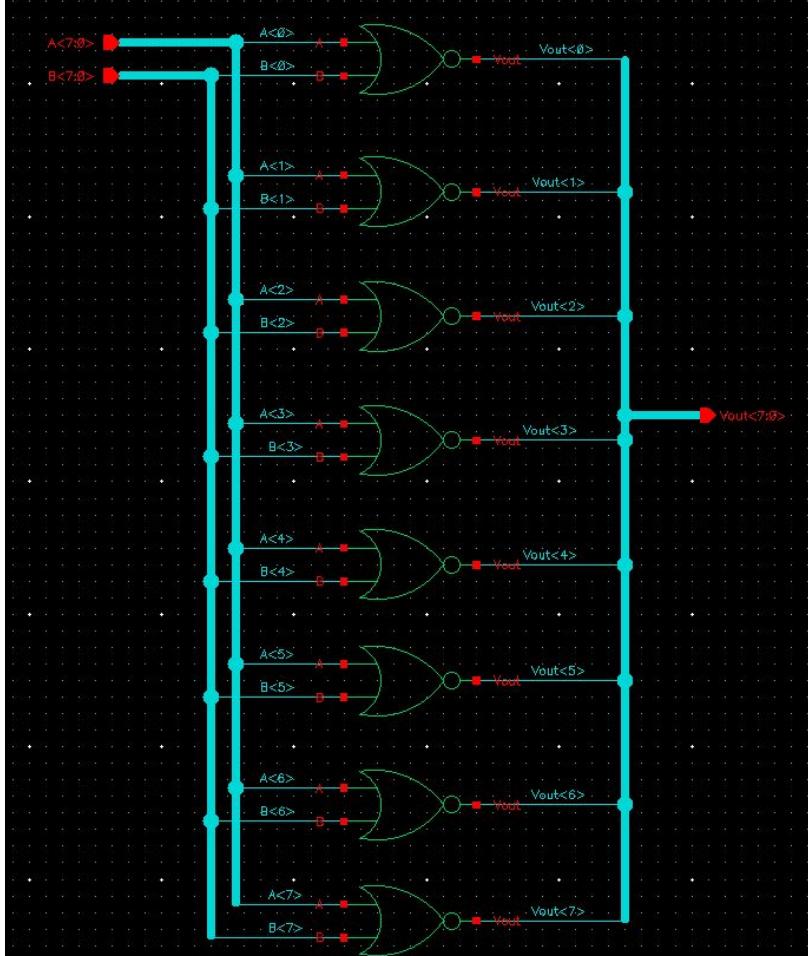
Resultados

- × Esquemáticos
- × Layouts



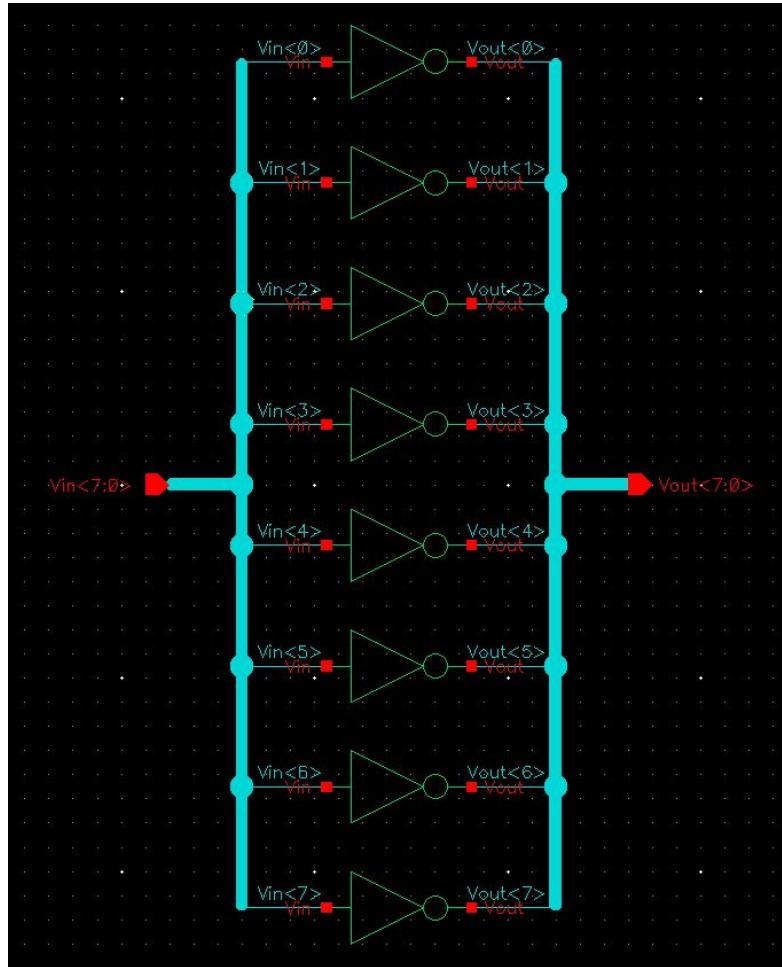


Porta AND
(8 bits)



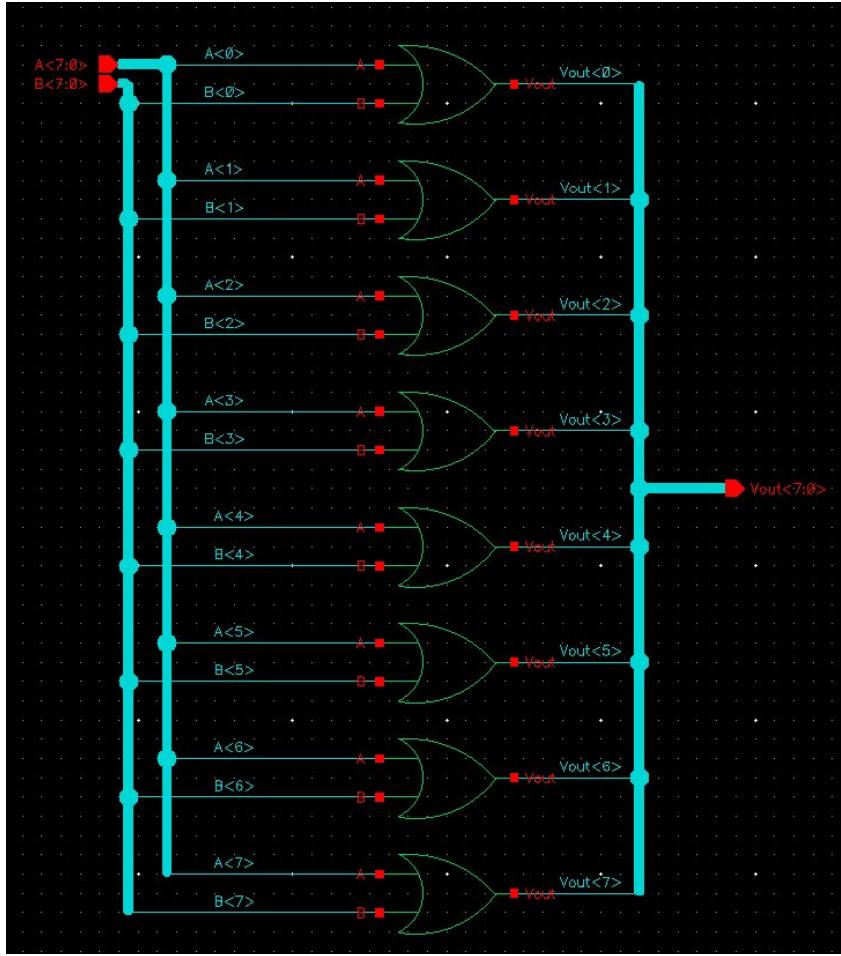
Porta NOR (8 bits)





Porta NOT
(8 bits)

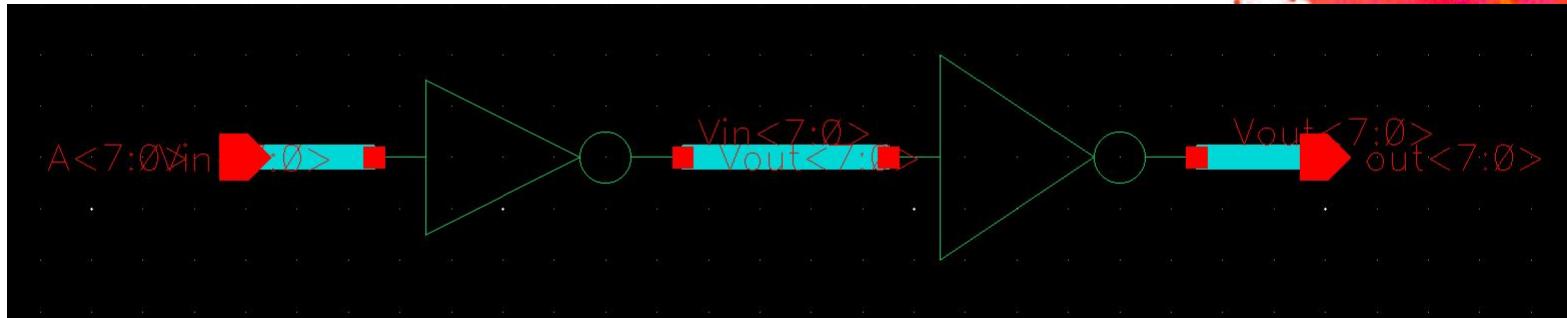




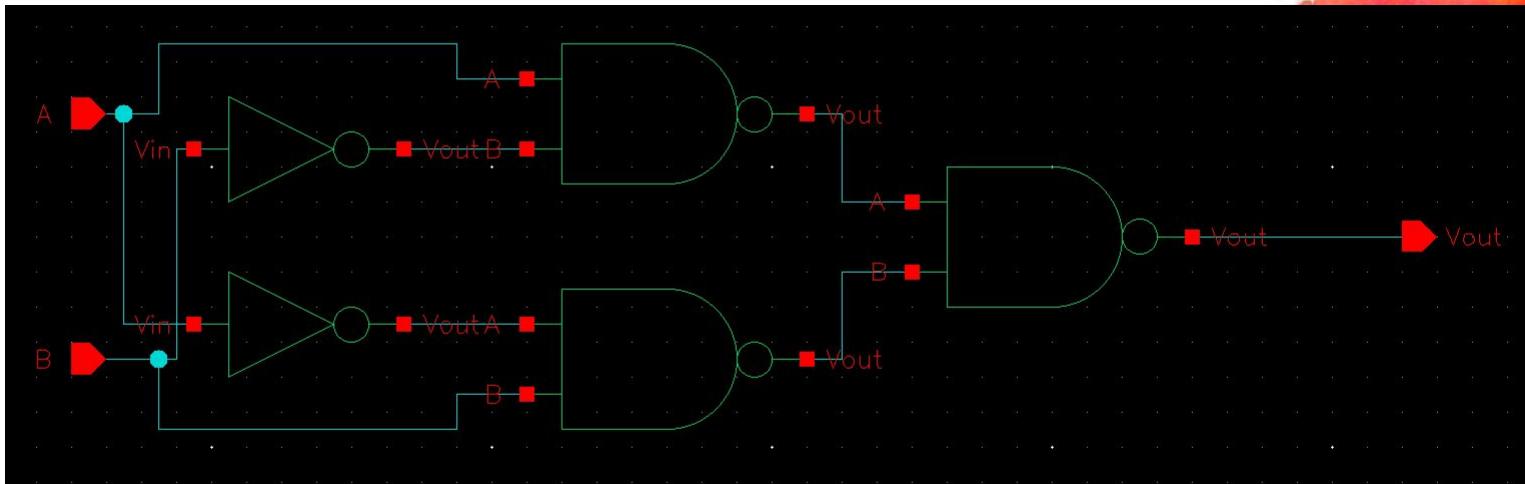
Porta OR (8 bits)



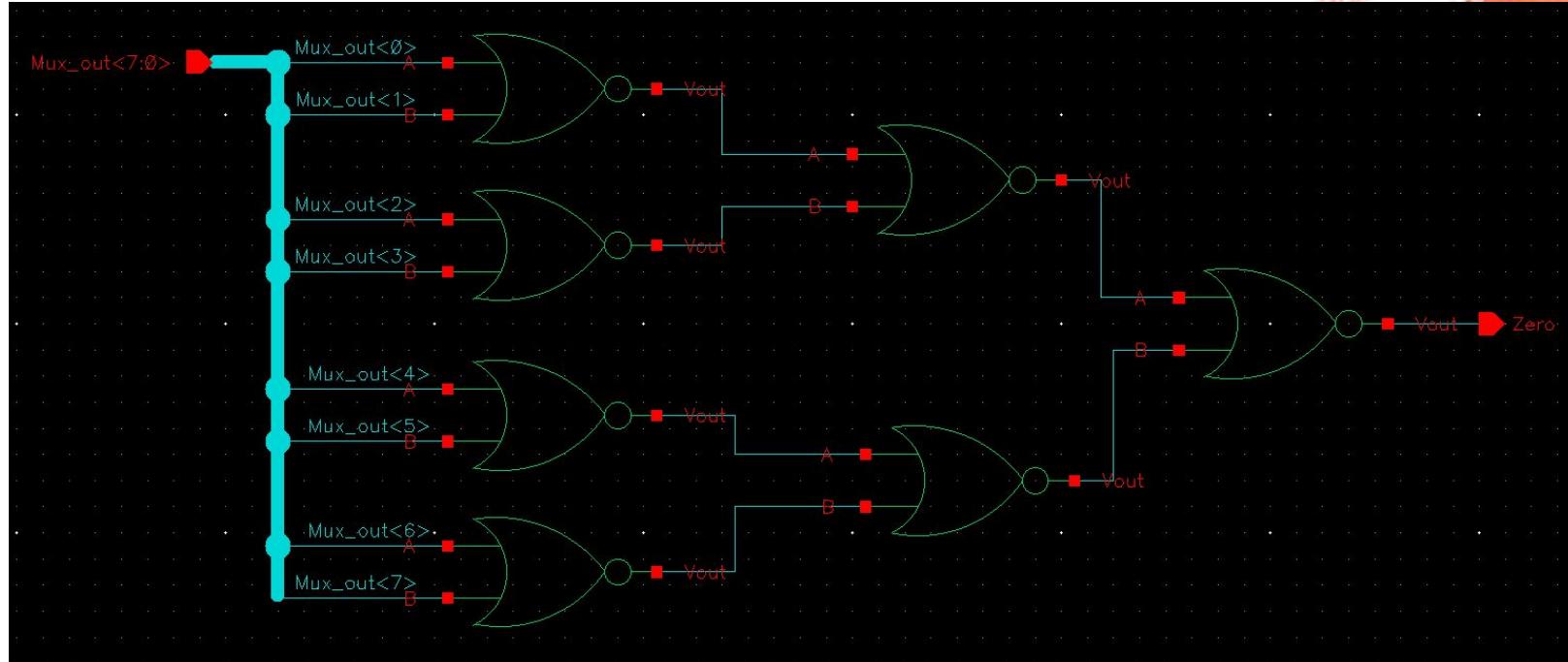
Buffer (8 bits)



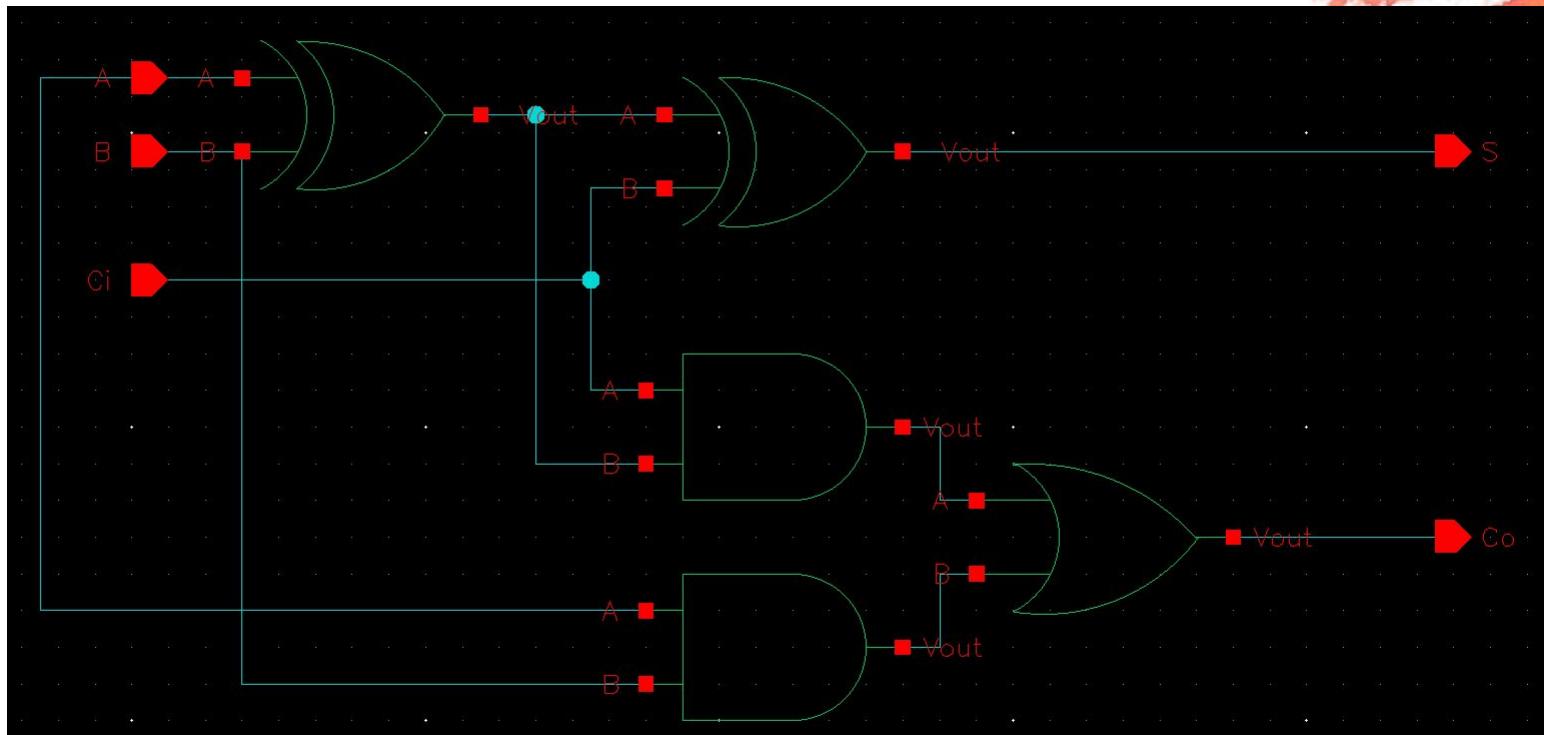
Porta XOR (1 bit)

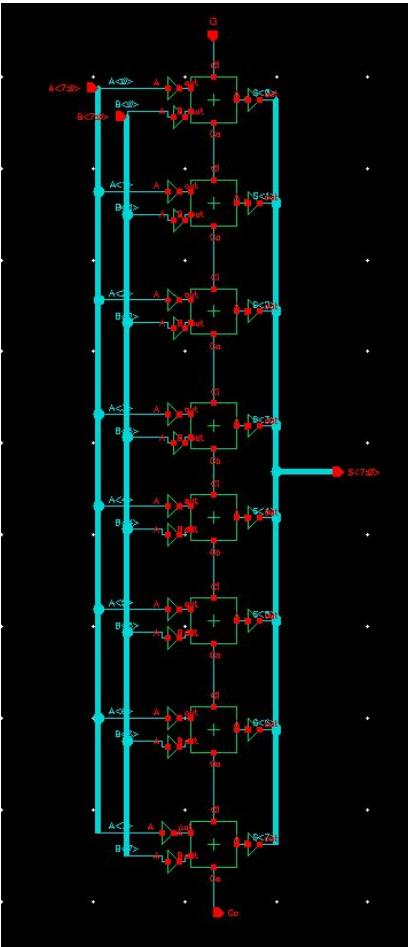


Porta NOR 8 entradas(bit a bit)



Somador (1 bit)

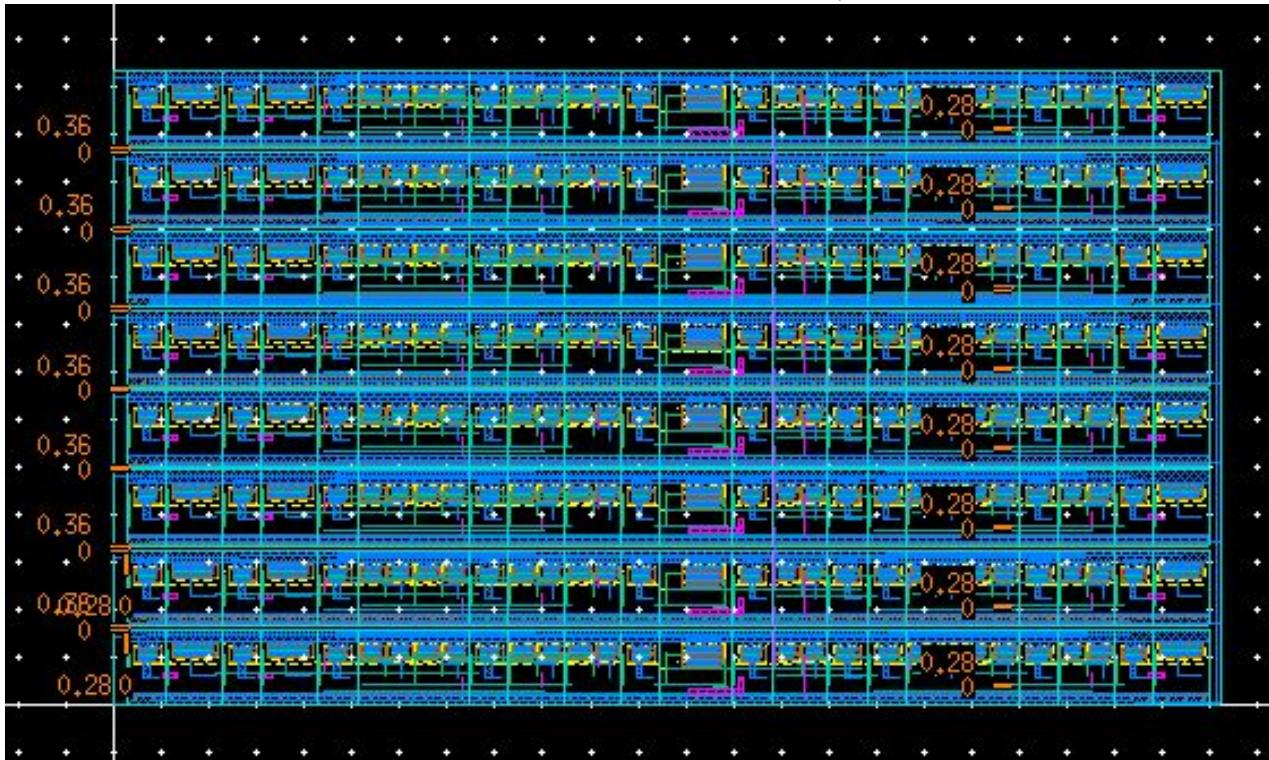


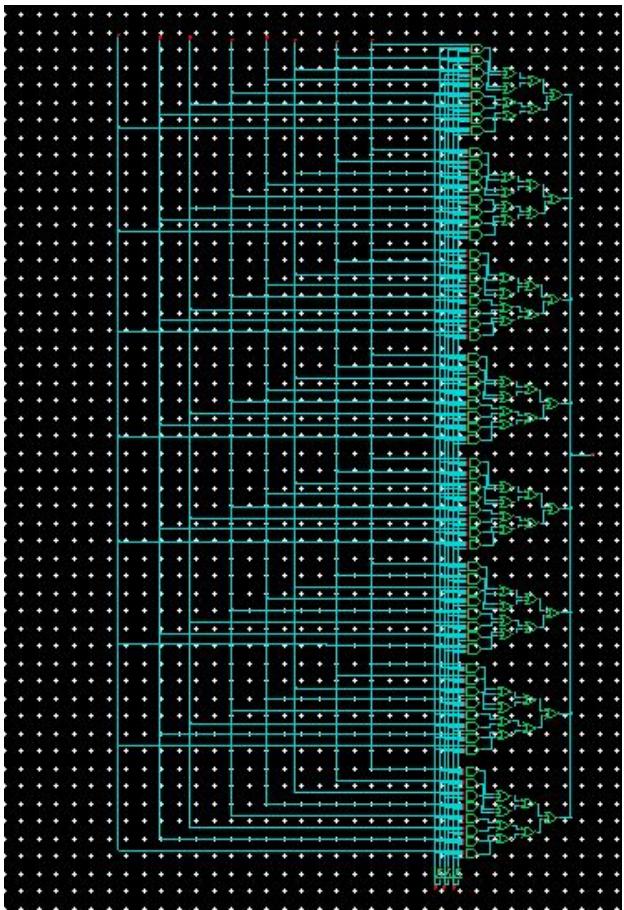


Somador (8 bits)



Somador (8 bits) - Layout

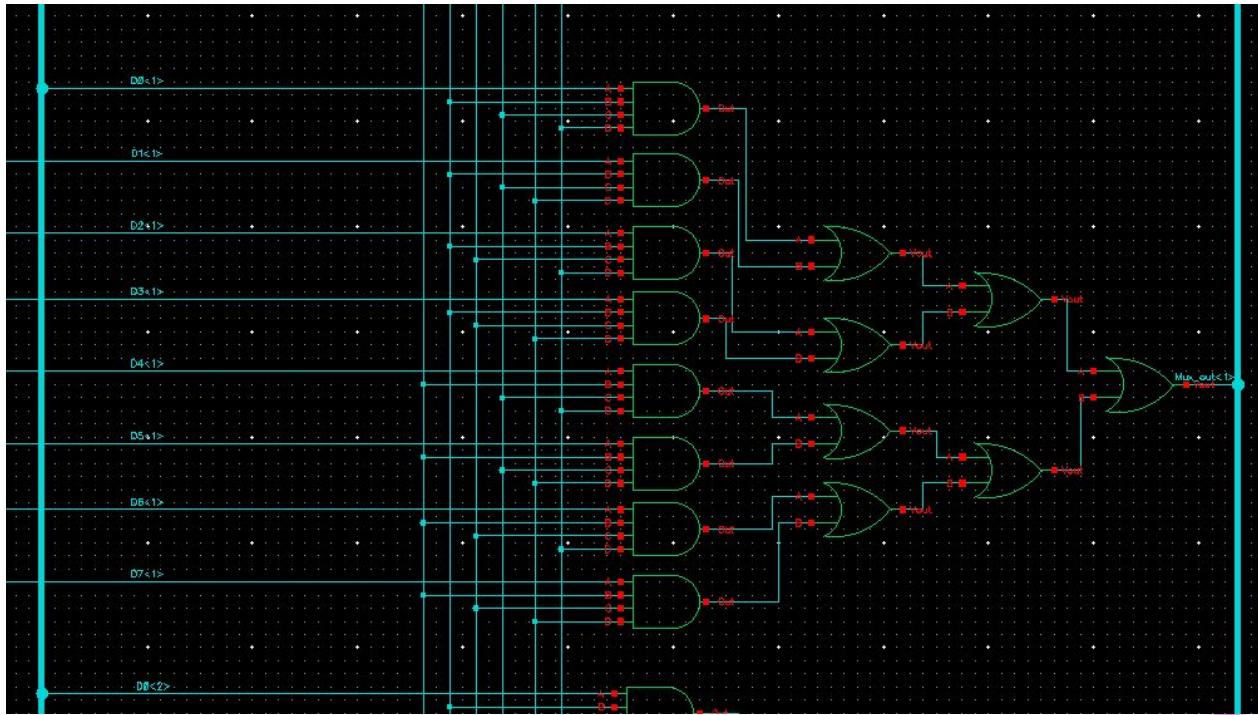




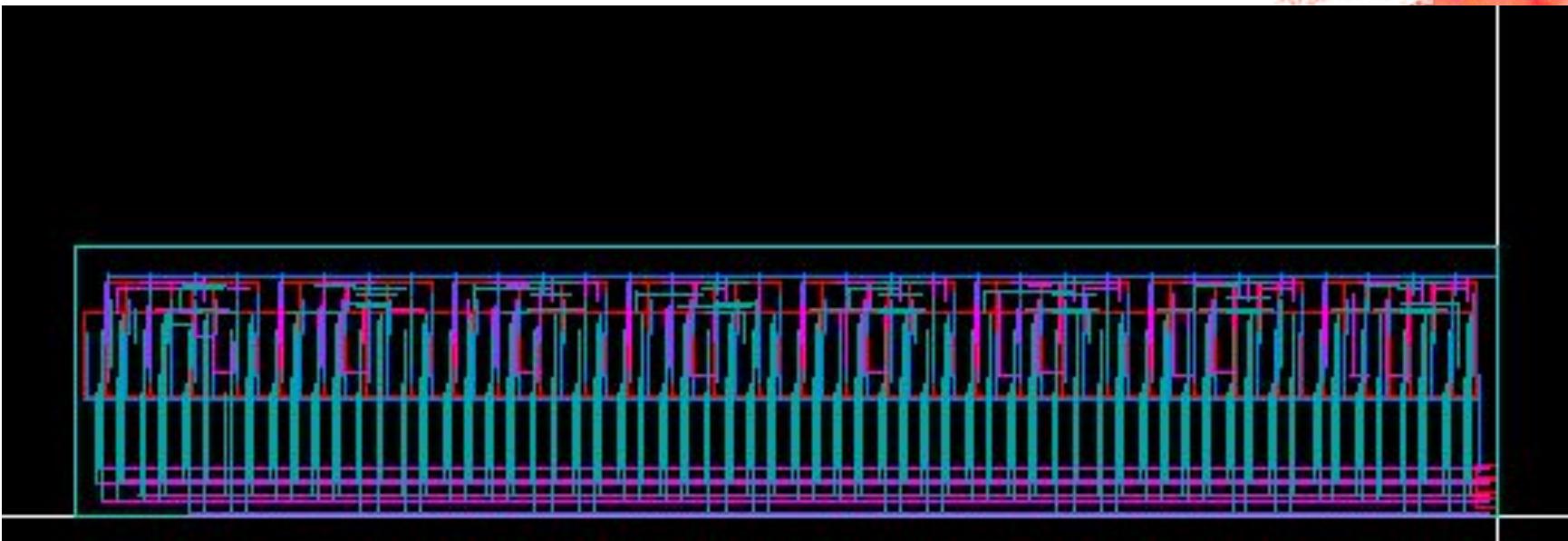
Multiplexador
(1 bit)



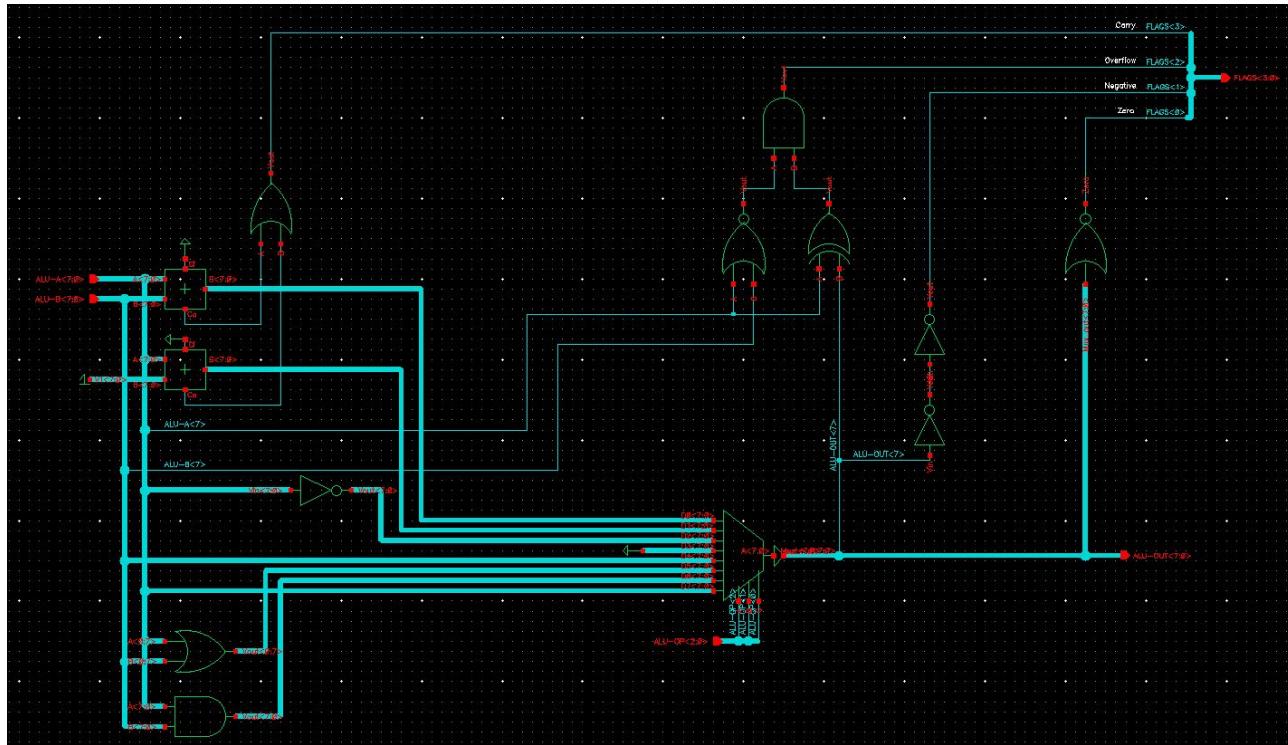
[ZOOM] Multiplexador(8 bits)



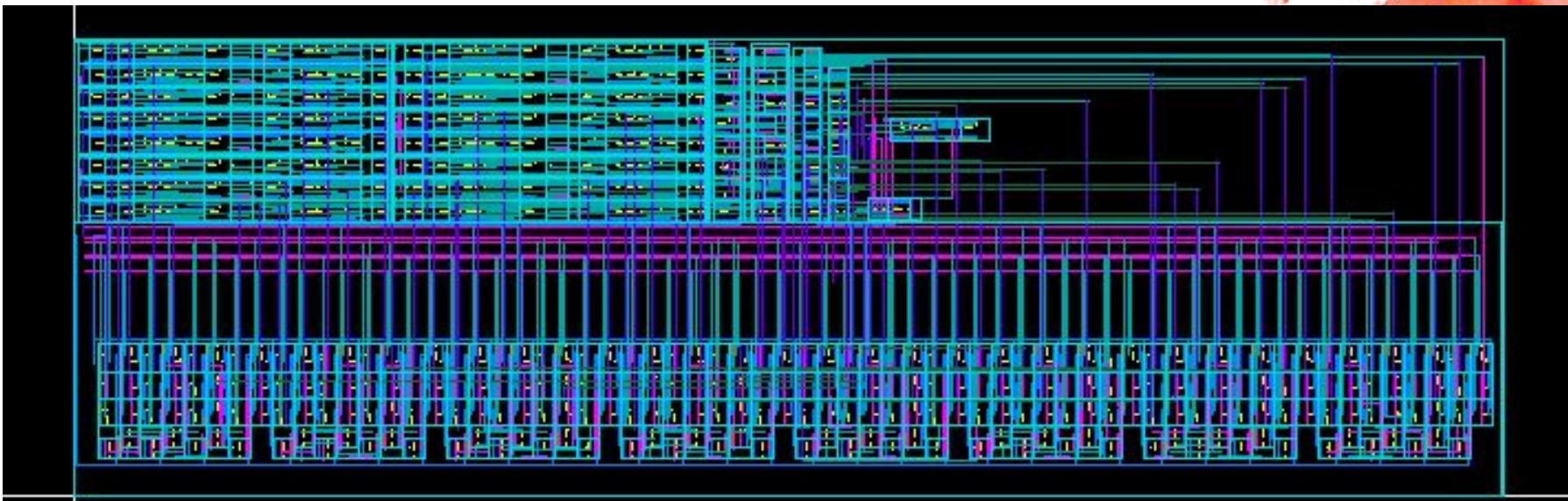
Multiplexador (8bits) - Layout



ULA (esquemático completo)



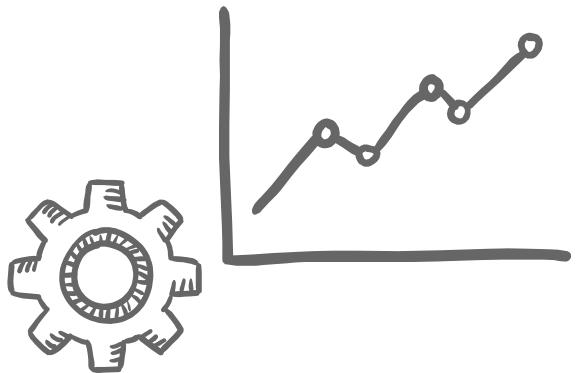
ALU Final



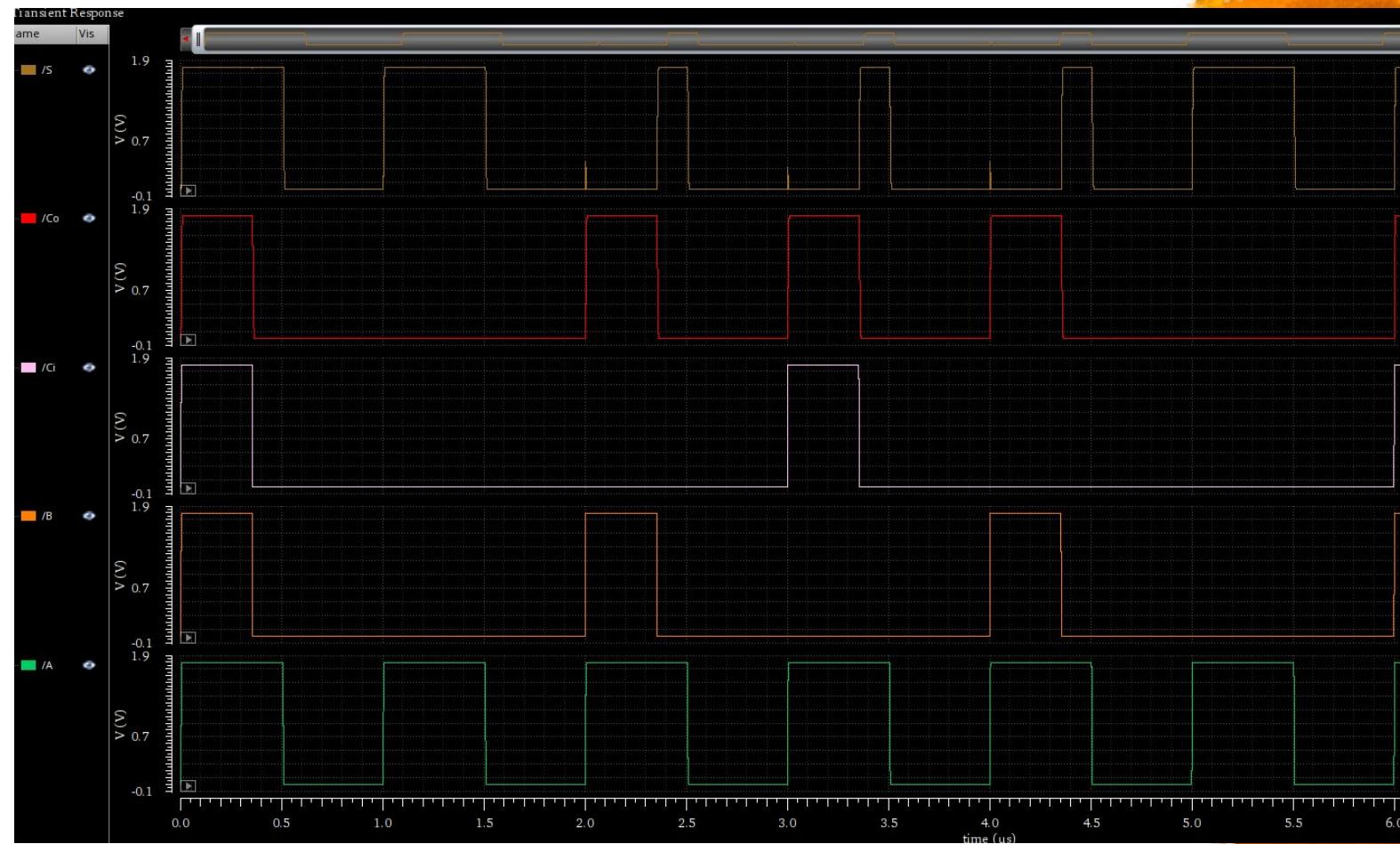
4.

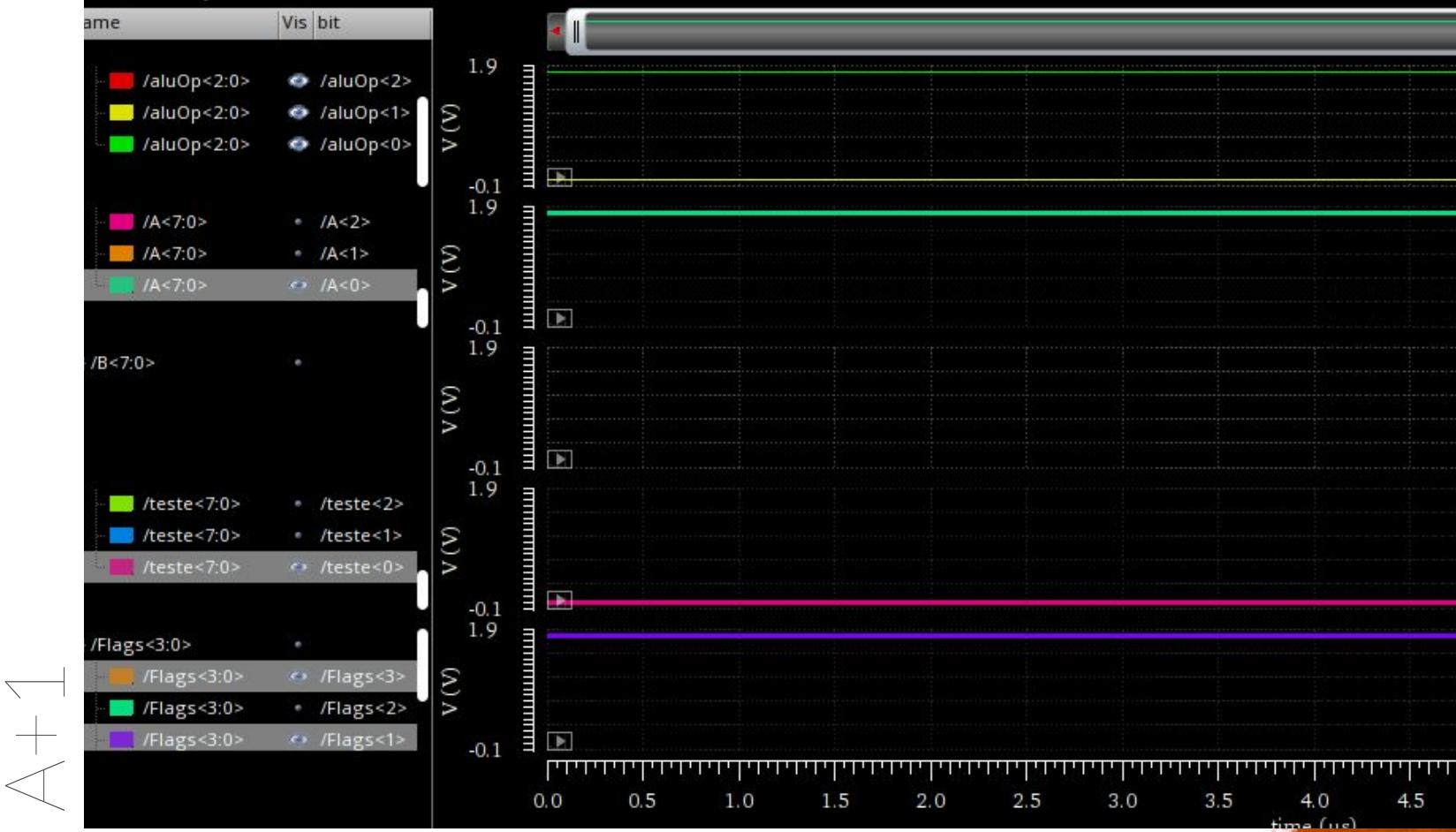
Simulações

- ✗ Somador
- ✗ $A+1$
- ✗ $A+B$
- ✗ A and B
- ✗ A or B
- ✗ $\text{not } A$
- ✗ Passa A
- ✗ Passa B



SOMADOR

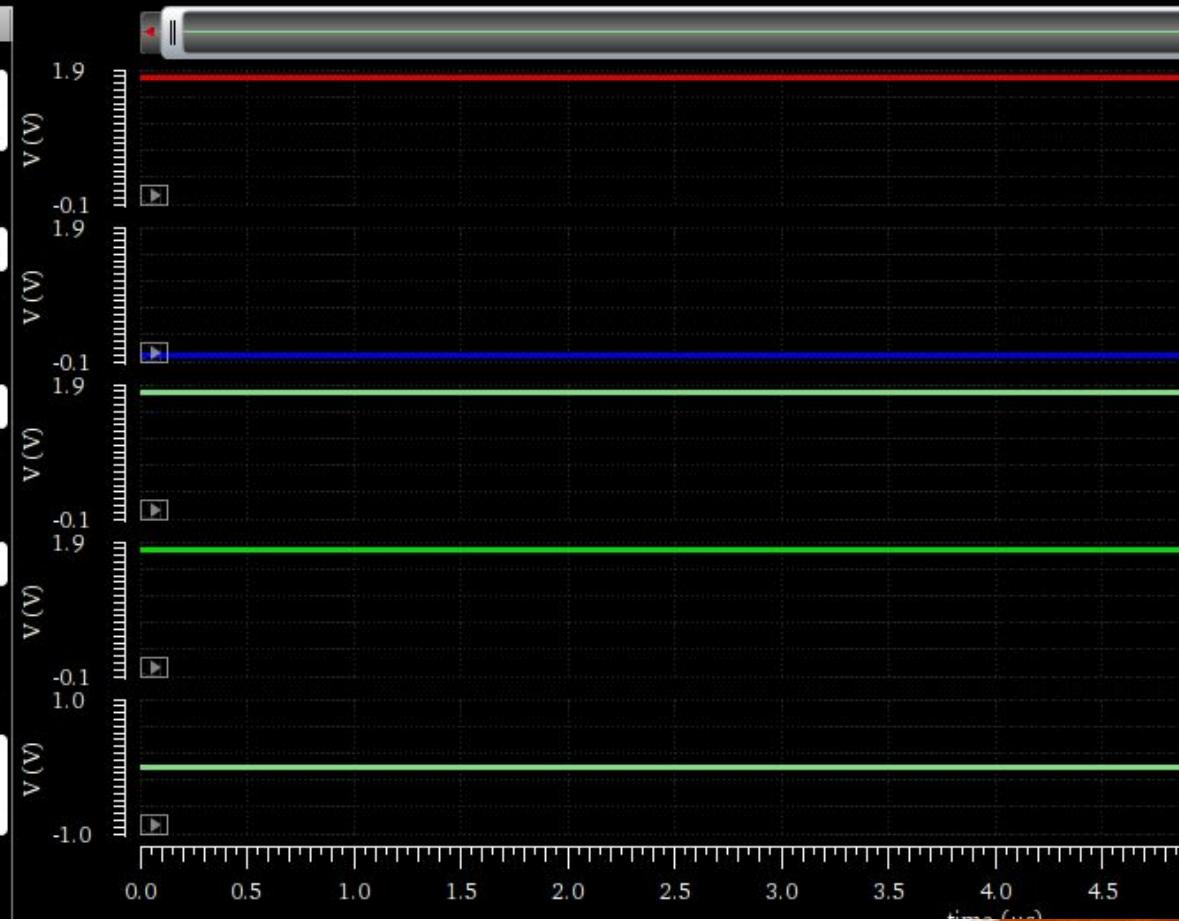




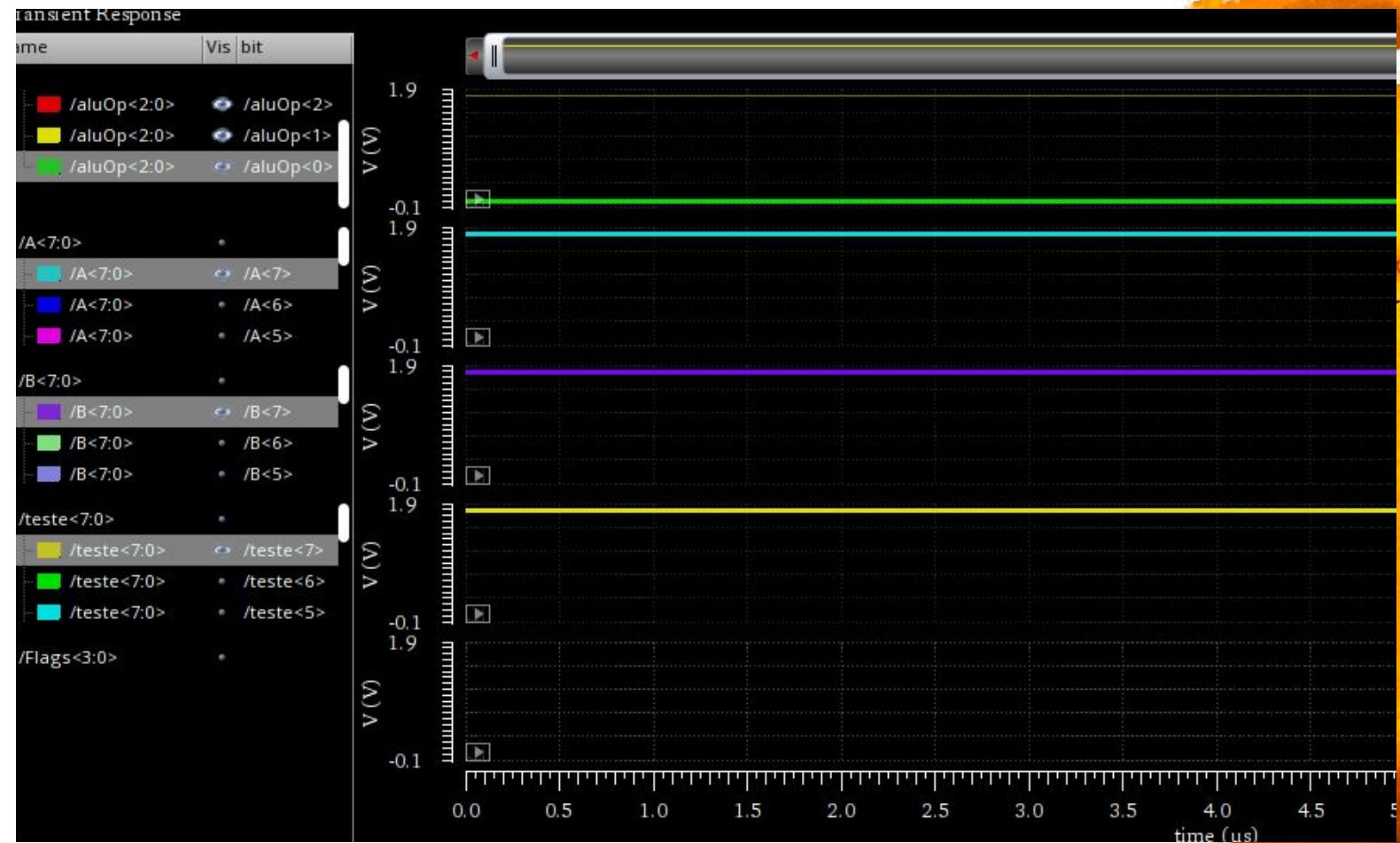
B
+
A

Transient Response

Name	Vis	bit
/Flags<3:0>	*	
■ /Flags<3:0>	● /Flags<3>	
■ /Flags<3:0>	● /Flags<2>	
■ /Flags<3:0>	● /Flags<1>	
/teste<7:0>	*	
■ /teste<7:0>	● /teste<7>	
■ /teste<7:0>	● /teste<6>	
■ /teste<7:0>	● /teste<5>	
/B<7:0>	*	
■ /B<7:0>	● /B<7>	
■ /B<7:0>	● /B<6>	
■ /B<7:0>	● /B<5>	
/A<7:0>	*	
■ /A<7:0>	● /A<7>	
■ /A<7:0>	● /A<6>	
■ /A<7:0>	● /A<5>	
/aluOp<2:0>	*	
■ /aluOp<2:0>	● /aluOp<2>	
■ /aluOp<2:0>	● /aluOp<1>	
■ /aluOp<2:0>	● /aluOp<0>	

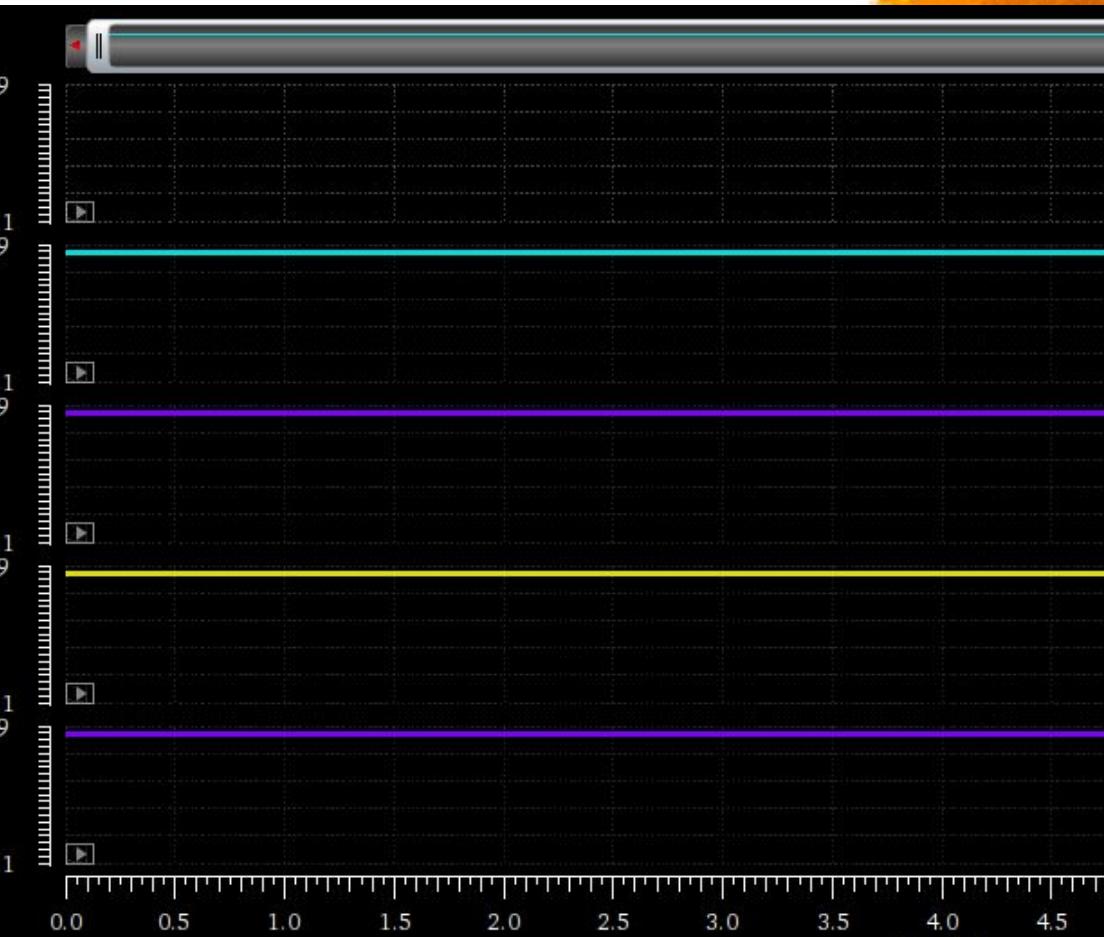


A And
B

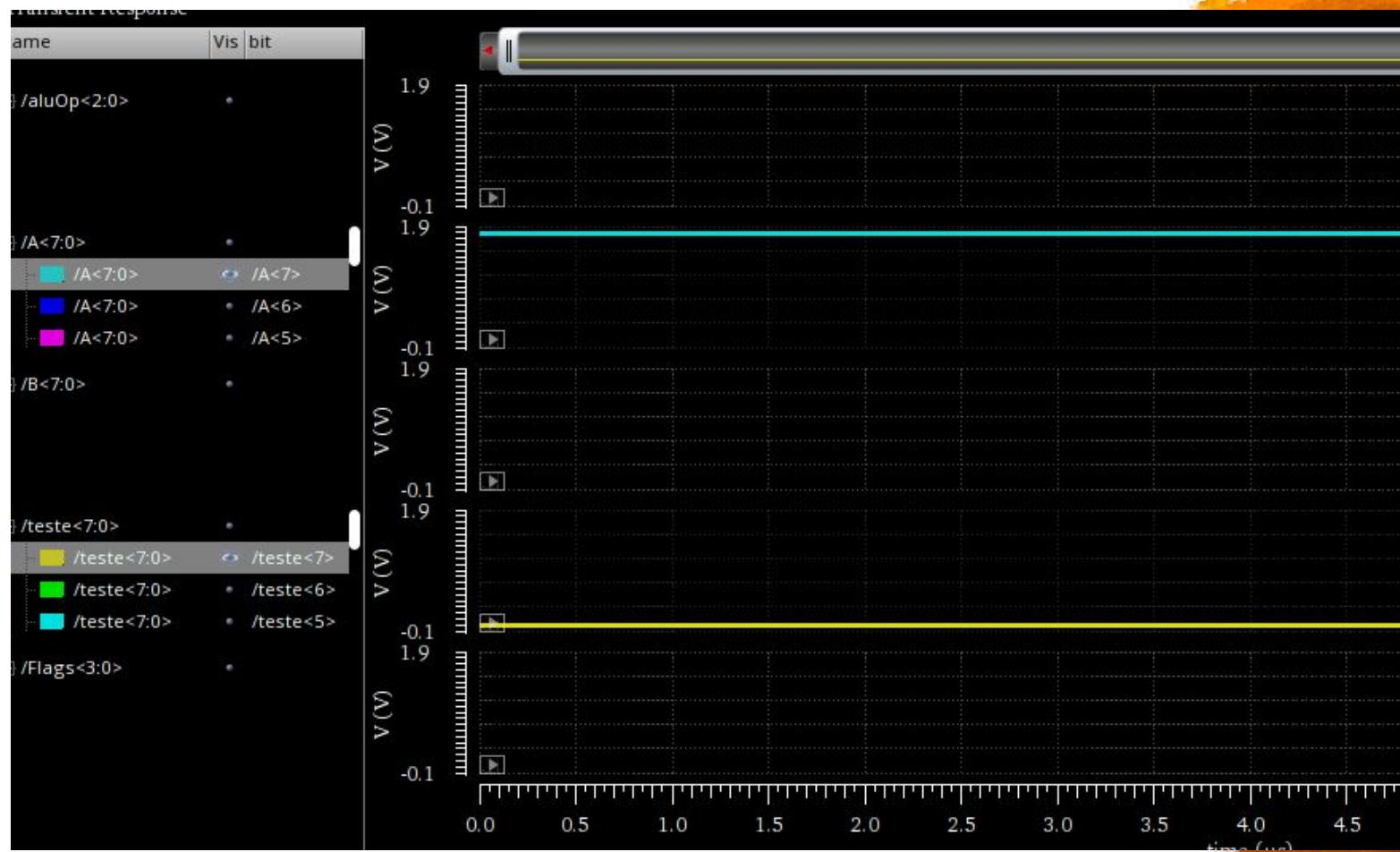


A O / B

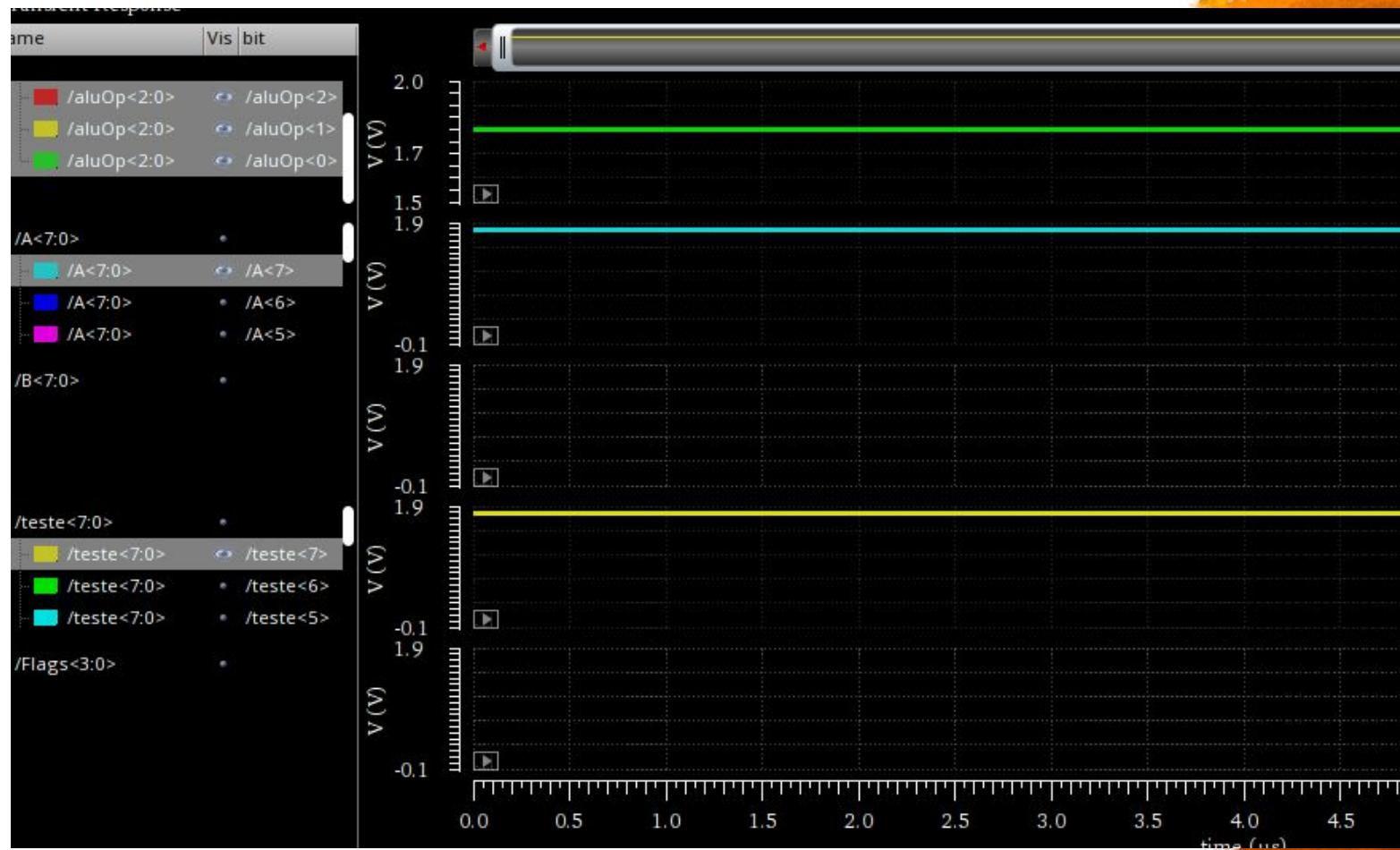
Name	Vis	bit
/aluOp<2:0>	*	
/A<7:0>	*	
/A<7:0>	*	/A<7>
/A<7:0>	*	/A<6>
/A<7:0>	*	/A<5>
/B<7:0>	*	
/B<7:0>	*	/B<7>
/B<7:0>	*	/B<6>
/B<7:0>	*	/B<5>
/teste<7:0>	*	
/teste<7:0>	*	/teste<7>
/teste<7:0>	*	/teste<6>
/teste<7:0>	*	/teste<5>
/Flags<3:0>	*	/Flags<3>
/Flags<3:0>	*	/Flags<2>
/Flags<3:0>	*	/Flags<1>
/Flags<3:0>	*	/Flags<0>



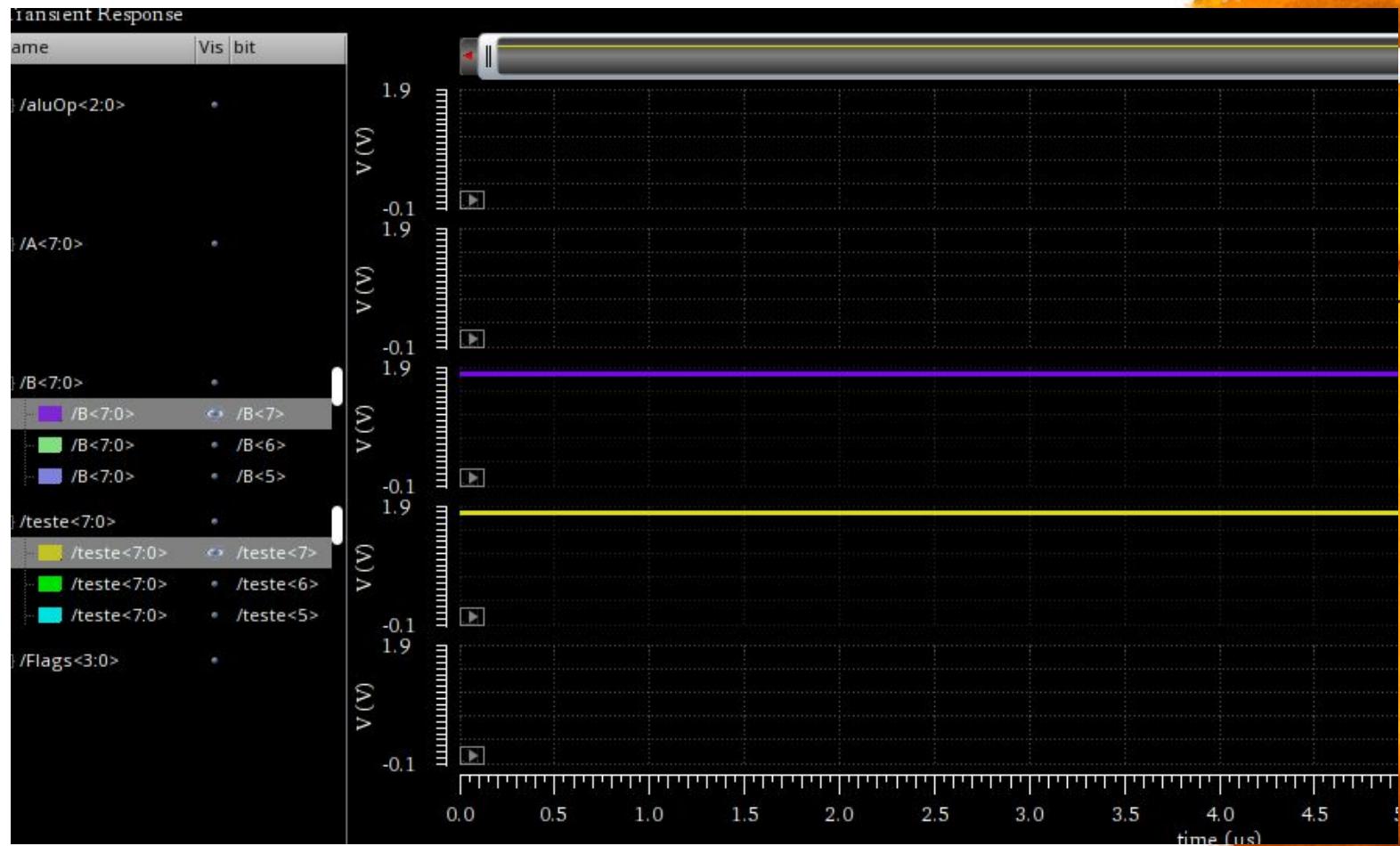
NOTA



A
S
S
A



B
A
S
S
A





OBRIGADO!



Giuliano Bohn Benedeti Becker | criação@fejers.org.br

Keli Tauana Prass Ruppenthal | kelitauana@gmail.com

Victor Dallagnol Bento | victor.bento@ecomp.ufsm.br