

AULA 25-11-2021

A avaliação prática é feita individualmente!!!

Vamos ter aulas que faltam: 2, 9, 16 de dezembro e 6 de janeiro

13 de janeiro é a avaliação prática da disciplina (vai ser oral -> vai perguntar tudo o que mandou fazer)

Vamos começar hoje o trabalho prático

Hoje vamos começar com o S-Edit (para casa, entender melhor o L-Edit e acabar o exercício 2 e fazer o 3 -> pedir ajuda ao Miguel)

Tem que se colocar o start dentro do design <- tannerEDA <- Documents

t-spice -> permite simular os circuitos

w-edit -> podemos ver a forma de onda dos sinais

S-Edit

- Pasta start deve estar nos documents
- Vamos abrir o ficheiro start.tanner
- Ao abrir e der erros, então quer dizer que a pasta não está bem colocada
- Abrir sem erros é o ideal
- Testbench vai ter vários sinais, no qual ele foi buscar às bibliotecas, sendo uma delas a pasta start -> quando quiser testar tenho que ir a esta célula
- Podemos ver a propriedades dos componentes em view -> propriets
- Para ver a fonte de tensão podemos carregar 2 vezes em cima dela (aqui só dá o mínimo) ou ir a properties e dá me tudo (o que é melhor)
- As fontes estão ao lado nas bibliotecas (na spice-elements -> que tem 9 tipos diferentes de fontes)
- A fonte bit-strem que vai girar uma série de bits como, por exemplo: 10101110 (ver imagem)

- Pattern -> padrão e podemos definir o padrão que queremos (é o que está em cima)
- RT -> tempo de subida ou rise time (de passar de 5V para 9V por exemplo)
- FT -> tempo de descida
- Ground -> Terra (estão todos na biblioteca em Misc)
- É bom ter as bibliotecas fechadas para não colocar elementos lá dentro
- Para fazer unlock da biblioteca, carrego com o botão direito em cima dela e lá aparece para editar
- Como se liga os componentes -> usam-se os fios que estão a vermelho em cima e vou andando e ligando os componentes (botão do lado direito libertar) diz "wire" -> pedir ao Carlos
- Net label -> Carrego e coloco aonde quero (só dá para colocar em cima de fios) e digo o nome que quero (sem espaços) -> colocamos nos sítios em que queremos ver o sinal (se não, não dá)
- Também podemos ir com uma ponta e ver o sinal no ponto em que colocamos essa ponta em cima
- Print Voltage -> fazer copy paste do que o professor fez e serve (coloca-se em cima do netlabel) para mostrar o valor da tensão do netlabel
- Biblioteca Misc
- Para ir buscar o Ground, vou à Biblioteca e carrego no lado esquerdo do rato e arrasto para o lado preto -> Como ir buscar componentes???
- Vamos fazer um inversor para começar:
Precisamos de um n-mosfet e um p-mosfet às bibliotecas Devices->Generic_250nm e vamos buscar os componentes pretendidos 2.5V e vamos colocar com o botão esquerdo (biblioteca logigate -> de portas lógicas, se não houver uma gate temos que a criar e colocá-la nesta biblioteca)
Cada extremidade dos componentes tem uma bolinha vermelha que serve para ligar a outros componentes.

Cuidado a copiar e colar porque podemos sobrepor dois componentes!!!!

Vamos colocar o n-mosfet no espaço em baixo direitinho e ligado

Vamos buscar o p-mosfet, carregamos no esquerdo e deixamos no espaço preto em cima ligado também

A fonte tem um pattern que é 1010....

Aquele tracinho é quase como um jumper -> liga à fonte que está de lado

Variável global -> diz que entre Vdd e ground é 5V, por exemplo

Para simular, depois de ter o circuito pronto, vamos fazer o seguinte:

Basta carregar no botão verde em cima!!!

Se eu colocar ao lado nas ferramentas vou ver as propriedades da simulação (tenho que colocar o certinho nas dois primeiros quadradinhos)

Fazendo a simulação, ele vai correr o t-spice para fazer a simulação e vai chamar automaticamente o w-edit para mostrar os sinais

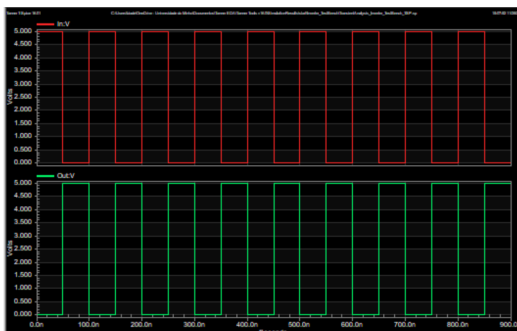
No w-edit:

Para ver os dois sinais separados temos que fazer o seguinte: chart -> expand traces (verificamos logo que temos um inversor)

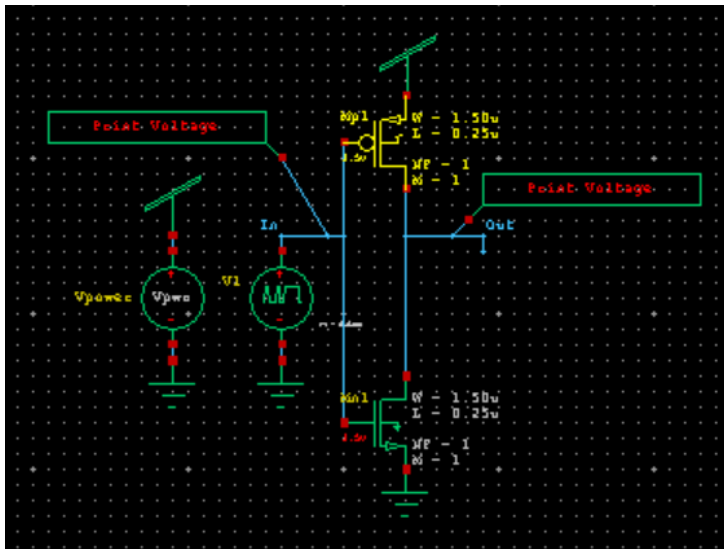
EXERCÍCIOS:

➔ Simular o inversor sozinho

Resultado obtido é: (pelo w-edit)



O circuito que fica é:



➔ Vamos fazer uma NAND

A NAND é uma gate que faz o seguinte: (por exemplo)

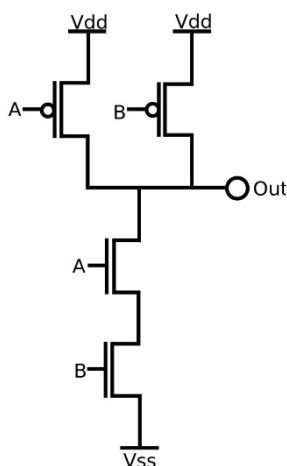
1 1 -> 0

1 0 -> 1

0 1 -> 1

0 0 -> 1

Basicamente o circuito de um NAND tem que ser dado por isto:

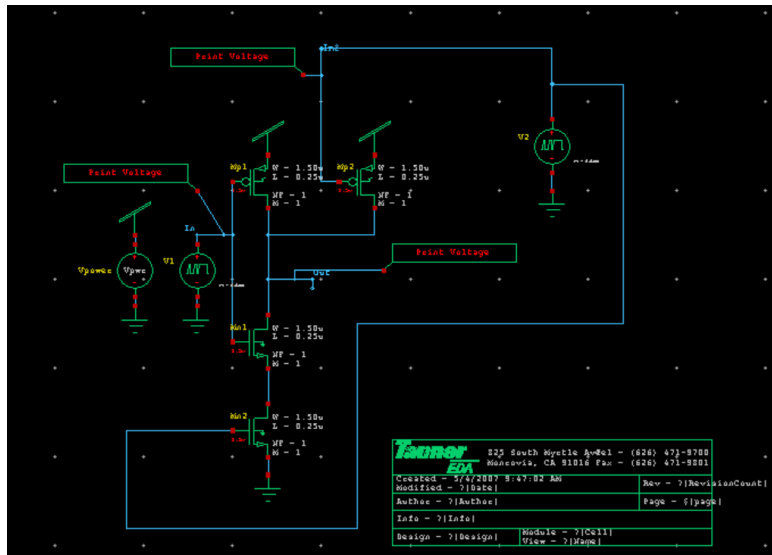


Quem tiver uma bolinha é o P-Mosfet

Quem não tiver bolinha é um N-Mosfet

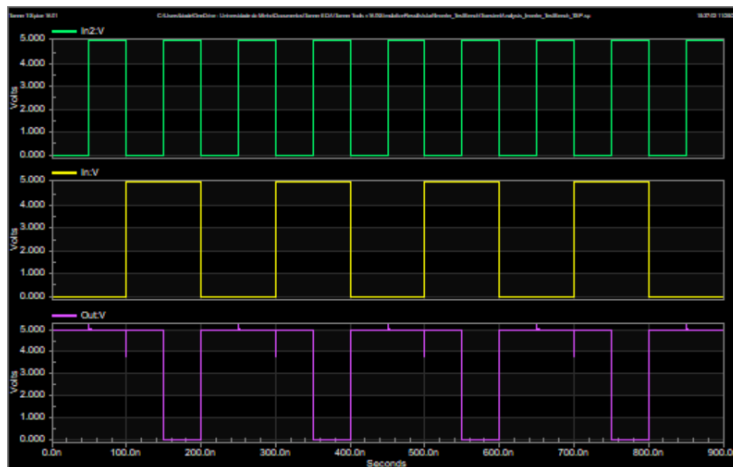
Basicamente, para fazermos uma NAND temos que fazer os 2 P-MOSFETs em paralelo e colocar os 2 N-MOSFETs em série

Basicamente no programa vamos colocar isto:



NOTA: ter em atenção que nas fontes dos bits temos que mudar o pattern das 2 para poder fazer a tabela

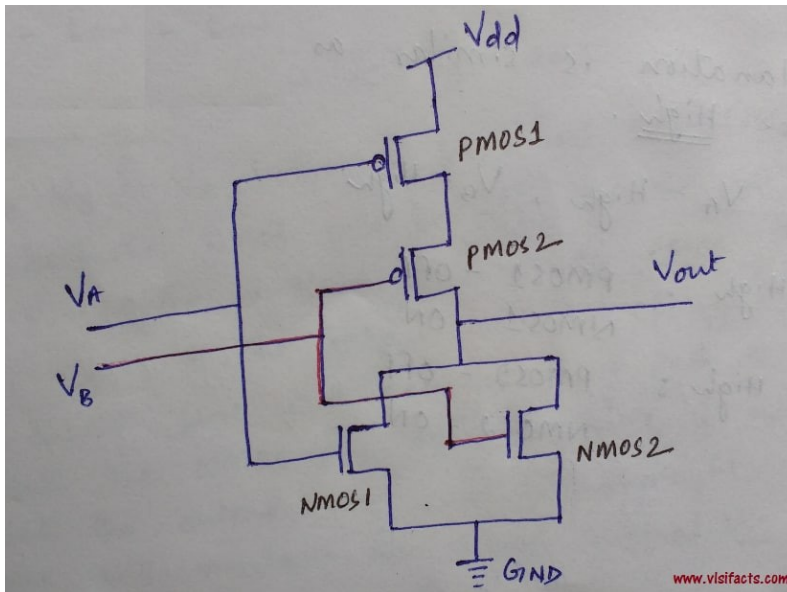
No w-edit vamos ter o seguinte:



Como resultado vamos ter isto:

➔ Vamos fazer uma NOR

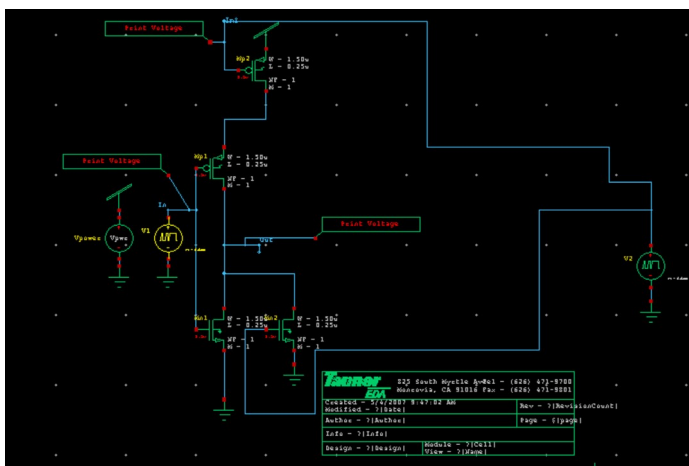
Aqui vamos ter os 2 P-MOSFETs em série e os 2 N-MOSFETs em paralelo:



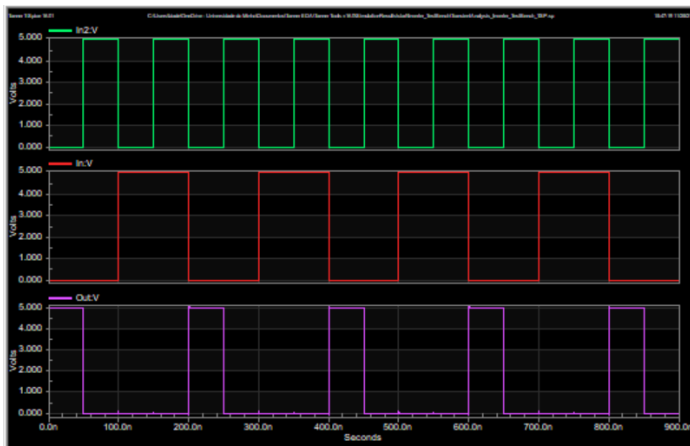
Na qual a tabela correspondente é a seguinte:

Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

O circuito que temos que montar é o seguinte:



O que vamos obter é o seguinte (no w-edit):



Aula do dia 02-12-2021

- Ir ao s-edit
- Abrir o documento start
- Queremos fazer um P-Latch e coloca-lo na biblioteca start

CB -> não/invertido clock

C -> clock

Para fazer isto fazemos:

- Buscar o Flip-flop D na biblioteca genérica 250 nm
- DFFC } Lado direito do rato e carregar em copy cell
(vamos usar este para depois modificar)

Copy to:

Escolher start

PLat -> mudar nome

Meter certo no Overwrite

- No start, carrego e faço open view
- Temos 2 schematic e o símbolo (apagar o que eu não quero do símbolo)
 - No schematic (0), vou ter que mudar o nome e mudar o circuito grande

O circuito mais pequeno é um BUFFER (não muda o sinal, apenas lhe dá muita potência)

- No schematic (2) temos que preservar o

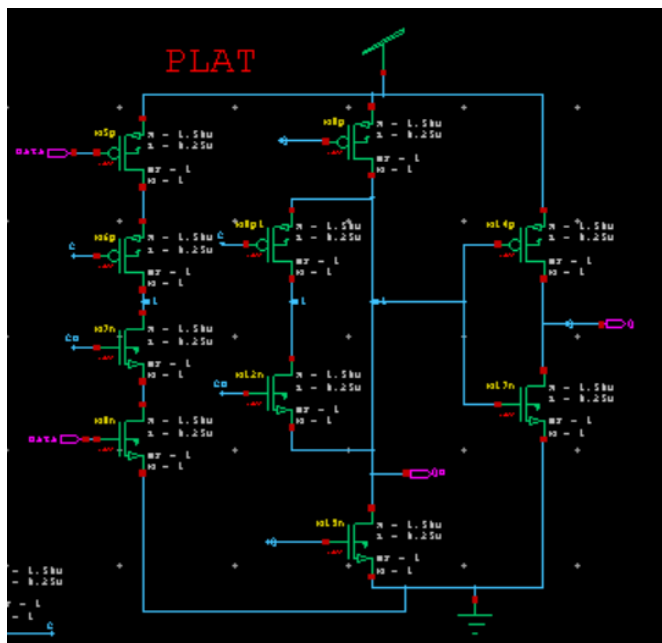
QB e o Q -> selecionar e copiar para a página 0, para termos saídas lá (depois podemos apagar a página 2)
cell > delete page > escrever 2

!!!! -> Apagar netlabels (=jumpers) todos que já existem na página 2

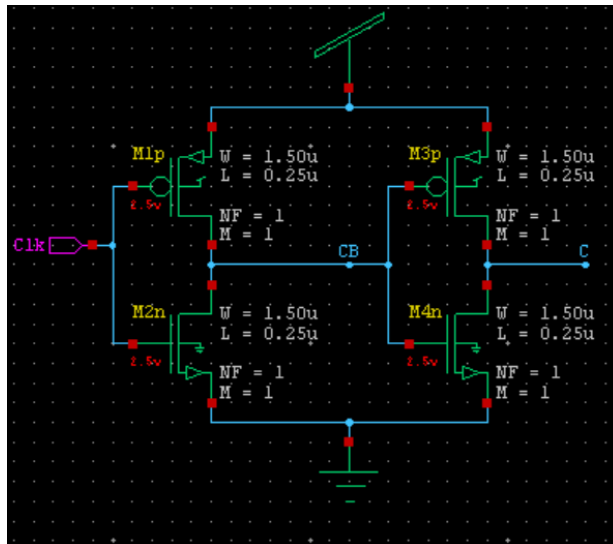
Interface -> diz o que é que são as coisas

Diz que o Q é de saída, ou seja, unidirecional por isso, não podemos usar aquilo roxo quando colocamos o Q dentro do circuito, temos que usar um netlabel ou ligar fios à saída (para podermos usar a cena roxa temos que ter o Q definido no interface como bdirecional).

Basicamente, o P-Latch vai ficar da seguinte forma:

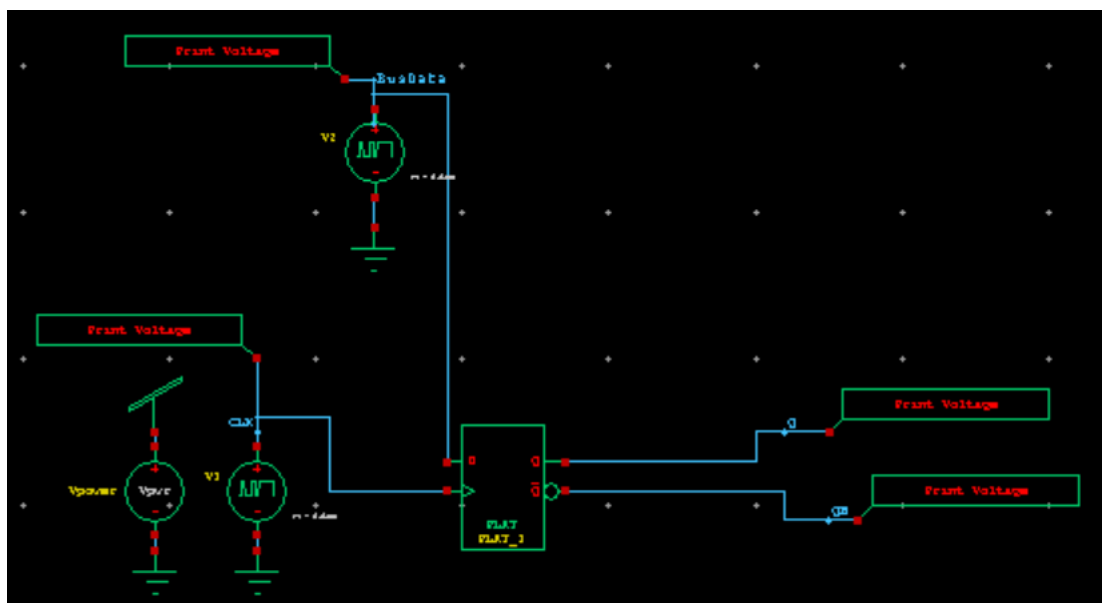


O buffer fica igual (não se mexe):



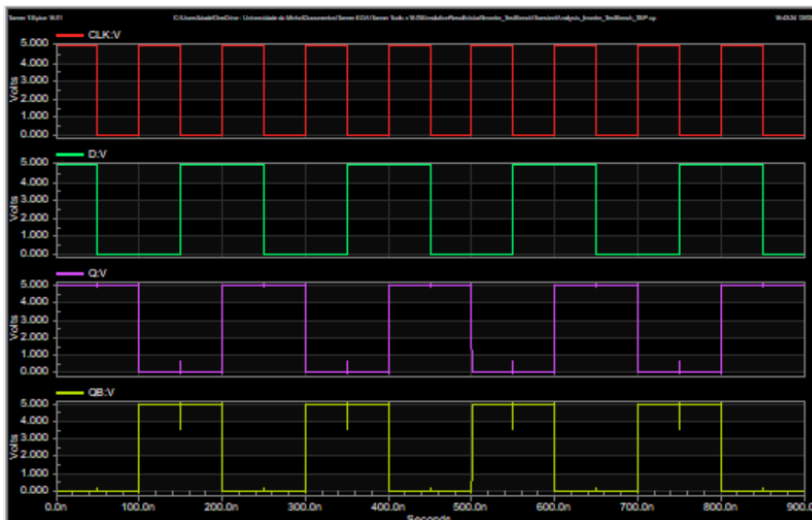
Para fazer a simulação temos que fazer o seguinte:

!!!! -> não esquecer antes de apagar a página desnecessária (página 2, neste caso)



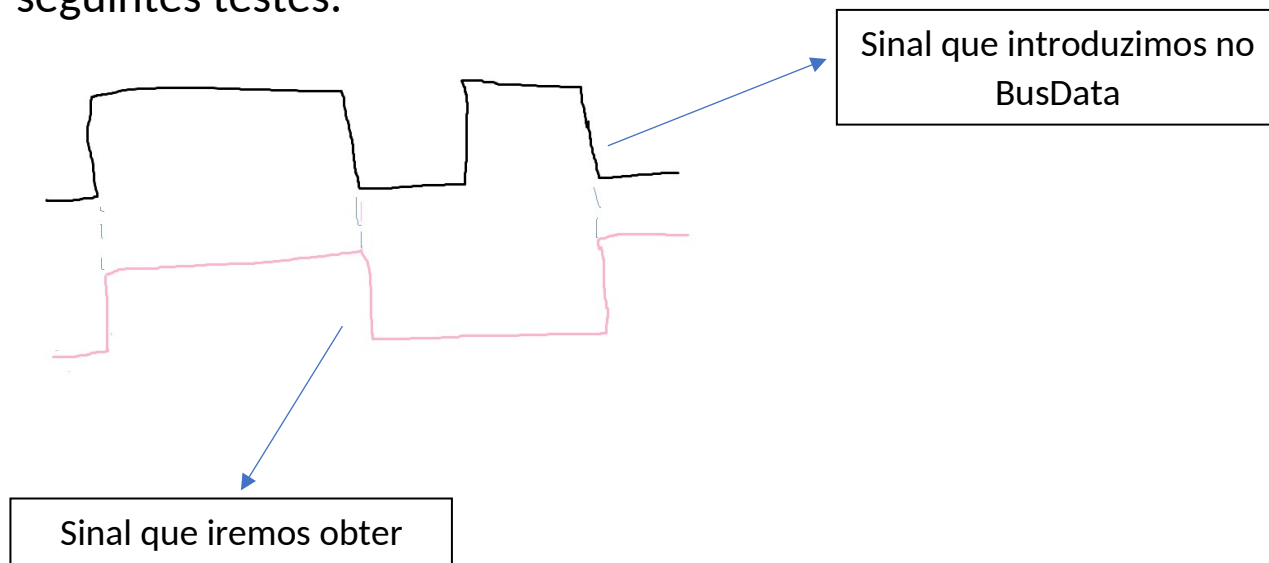
Temos que ir ao inverter_test e fazer este circuito e testá-lo.

Basicamente, a resposta que temos que obter para um certo BusData vai ser o seguinte:

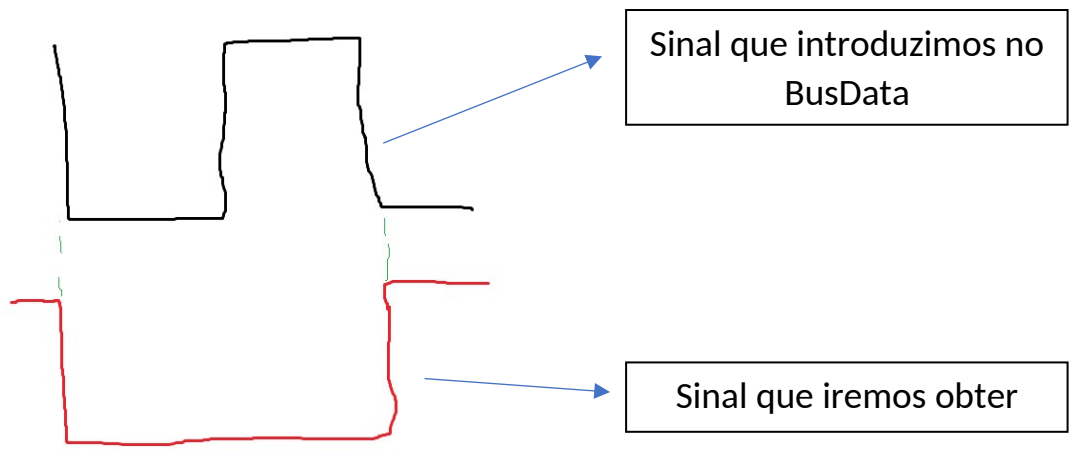


NOTA: tenho que ver com atenção como funciona este latch

- Para P-Latch funcionar, podemos utilizar os seguintes testes:



ou podemos usar o start bit:

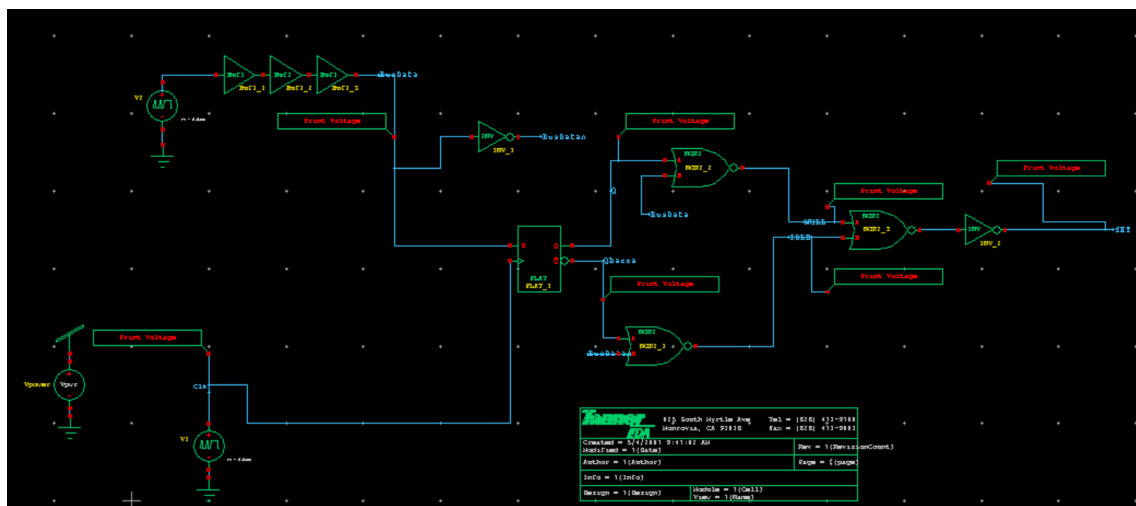


DESCODIFICADOR DE MANCHESTER

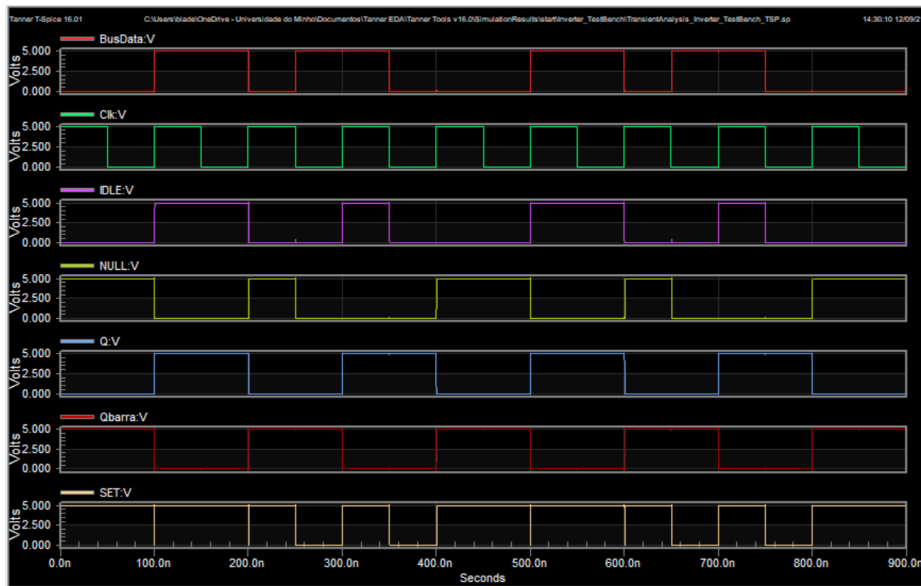
- Para ir buscar portas lógicas -> generic 250 nm portas lógicas
- Usamos o Buf1 , NOR2 e para fazer o OR uso uma NOR2 seguida de um inversor
- NAND -> usamos o NAND2C e ao out ligamos o no connection (NC), que se encontra na biblioteca MISC
- Para usar um flip-flop, temos que ir generic logic gate e tenho vários. Pego no DFFC e para ele funcionar normal (D flip-flop) simplesmente ligo ao Clr um buffer, que por sua vez tem que estar ligado ao Vdd.

!!!!!! -> Nunca ligar diretamente o Vdd ao Clr (perguntar porquê)

O circuito que se tem de montar vai ficar assim:



E o resultado vai ser este:

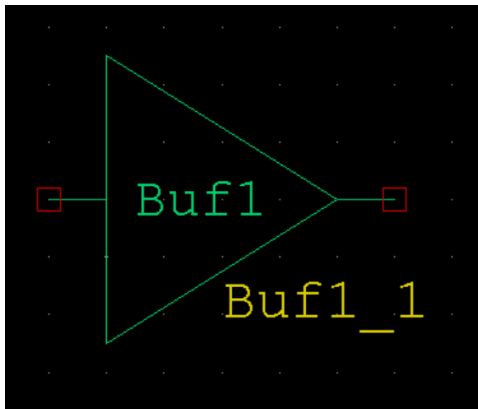


CONSELHO: sempre que tiver um sinal de clock, devemos colocar um buffer a seguir (para aumentar a potência do sinal).

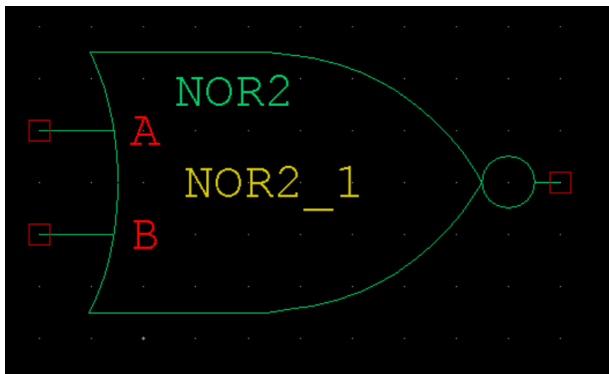
Vamos ter que aumentar a potência do BusData também (colocar no mínimo 3 buffers de seguida para aguentar o sinal). Se não colocarmos, o nosso BusData tornar-se-á mais lento pela perda de fluência no decodificador.

Os Componentes utilizados são os seguintes:

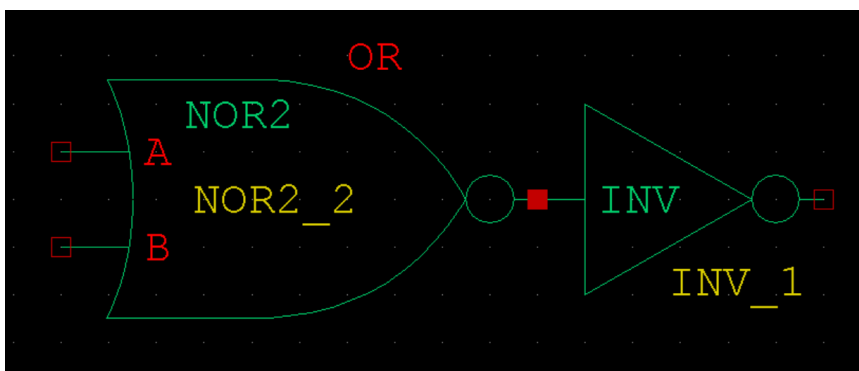
✓ Buffer



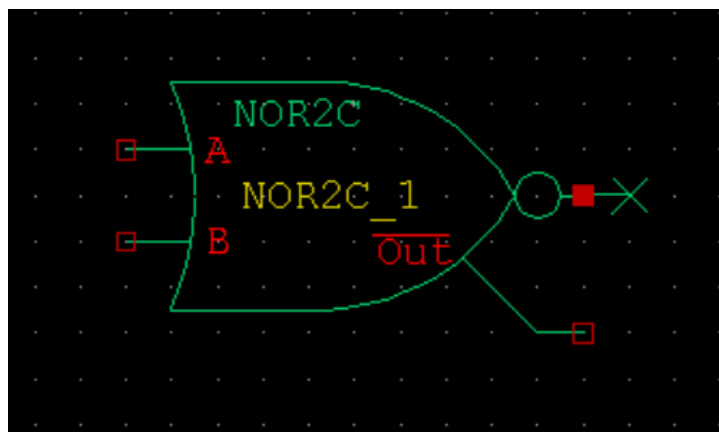
✓ NOR



✓ OR



em vez de usarmos um NOR seguido de um inversor como na imagem, podemos usar um NOR2C, na qual colocamos aquilo do X (no connection) na saída principal e usamos a saída de baixo:



Aula 09-12-2021

- Ir ao s-edit
- Abrir o start
- E abrir o decodificador de manchester que fizemos
- Ter em atenção que temos que colocar no busdata vários buffers porque como vamos fazer um circuito muito grande, ele pode não ter capacidade suficiente para aguentar isso tudo e por isso devemos aumentar a potência do sinal (para não termos sinais errados) -> no clock também mas aqui colocar sempre o clock a sair da mesma origem
- Estamos a querer colocar o set em vários flip-flops, ou seja, o sinal pode também não aguentar, portanto devemos colocar no mínimo 2 buffers
- Set up -> spice simulation
- É preciso 1000n segundos para conseguir fazer a simulação e de seguida o maximum ... devemos ter 50n segundos (mas não é critico, ou seja, não há necessidade de mudar)
- BusData -> propriedades -> colocar o pattern :
001111011010101011

(10 deve-se ao facto de que como o slave vai enviar 10 (acknowledge bit) e nós para não dar erro mandamos também um 1 em código Manchester)

(01 é o start bit que o master vai enviar)

(1010 é o A_0 e o A_1 respetivamente)

(10 deve-se ao facto de que como o slave vai enviar os dados e nós para não dar erro mandamos também um 1 em código Manchester)

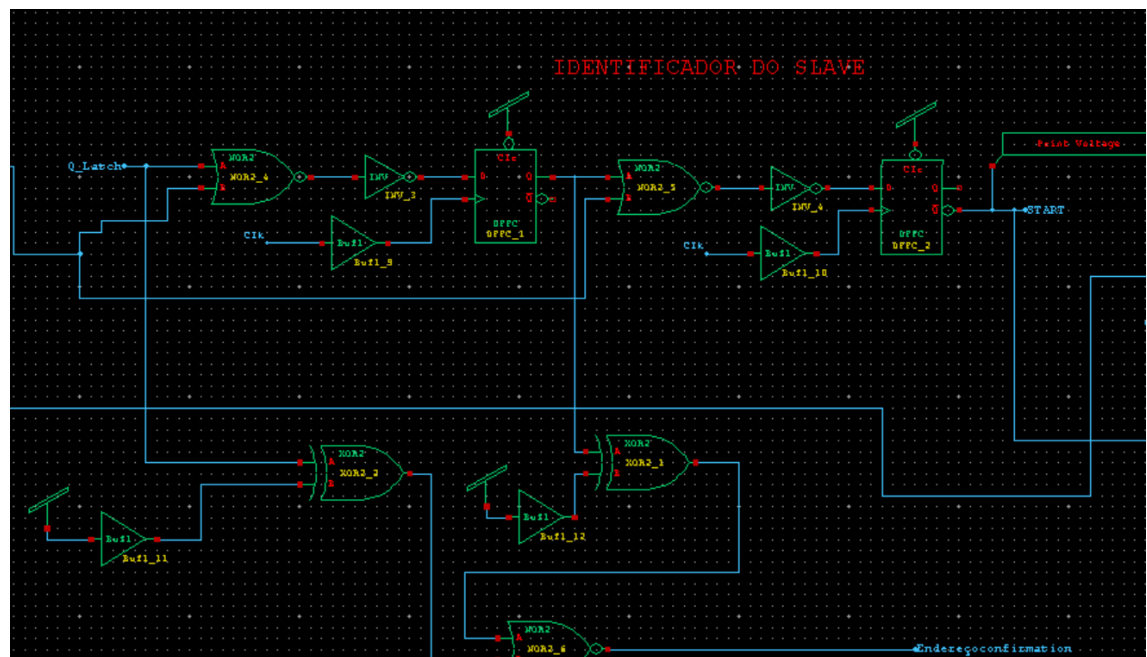
!!!! -> Algumas coisas não consegui ouvir... Pedir ao Miguel

- Cuidado -> antes de usares as XOR ver se elas funcionam (se não funcionarem, usar um XORN e depois um inversor)

NOTA: para rodar um componente é seleccionar o mesmo e carregar na tecla R

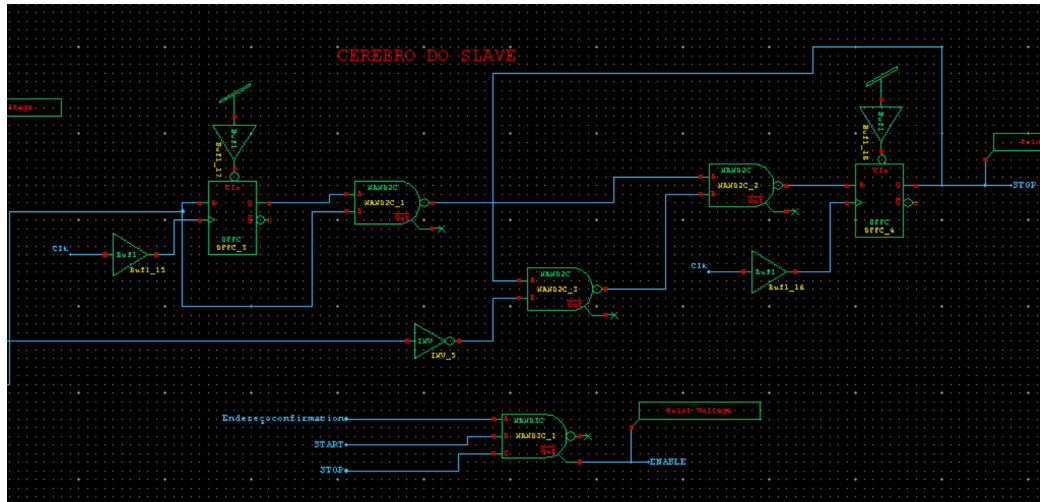
- Ter em atenção para não se ligar Vdd diretamente ao clr do DFFC. Isto deve-se ao facto de que Vdd é analógico e DFFC é digital e por isso a simulação vai dar erro, logo a seguir a Vdd tem que se colocar um buffer.

Basicamente, vamos ter que o conversor série-paralelo vai ser da seguinte forma:



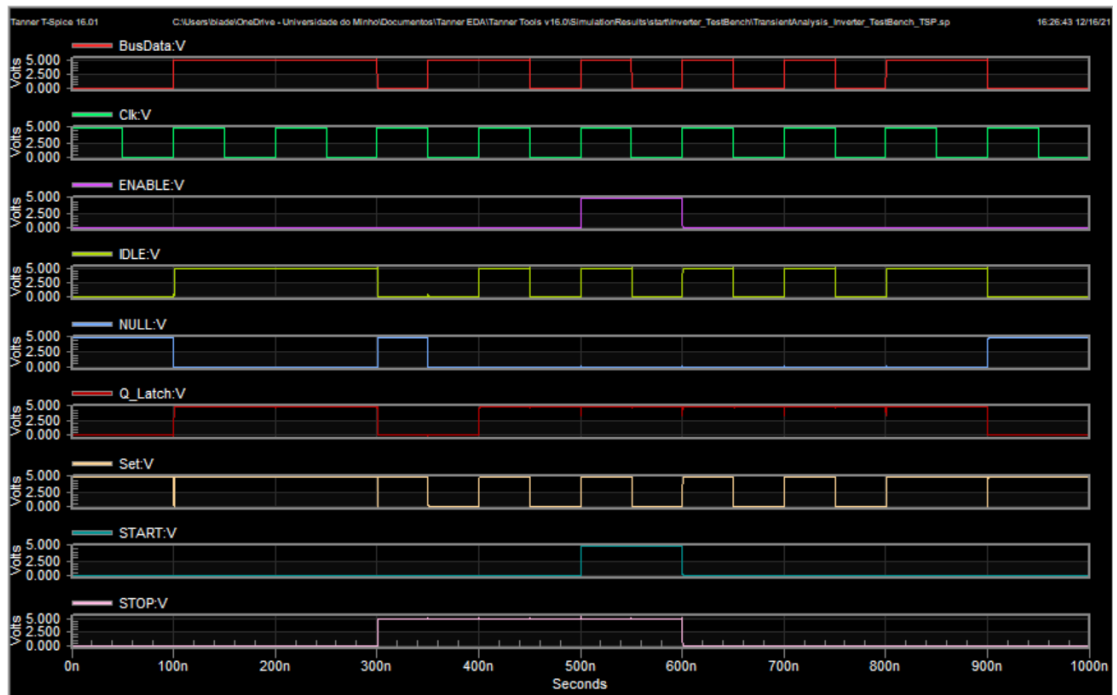
!!!! -> não entendi o porquê mas supostamente não dá para testar isto diretamente, temos que aplicar o bloco a seguir para poder testar.

A seguir, vamos ter que implementar o “cérebro” do Slave:



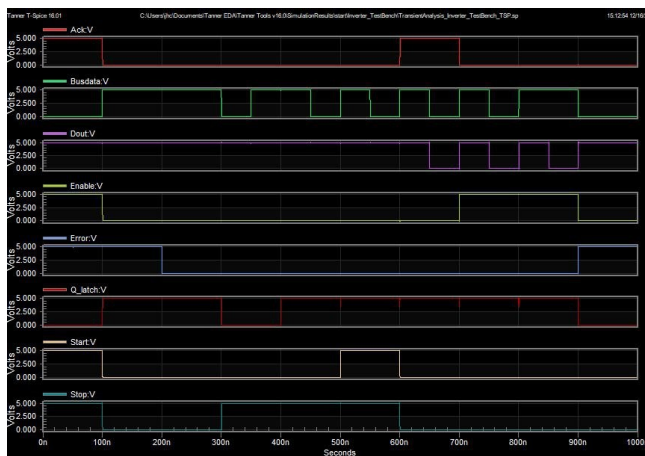
O resultado obtido terá que ser o seguinte:

!!!! -> ter em atenção que os resultados obtidos de 0-100n não devem ser tidos em conta

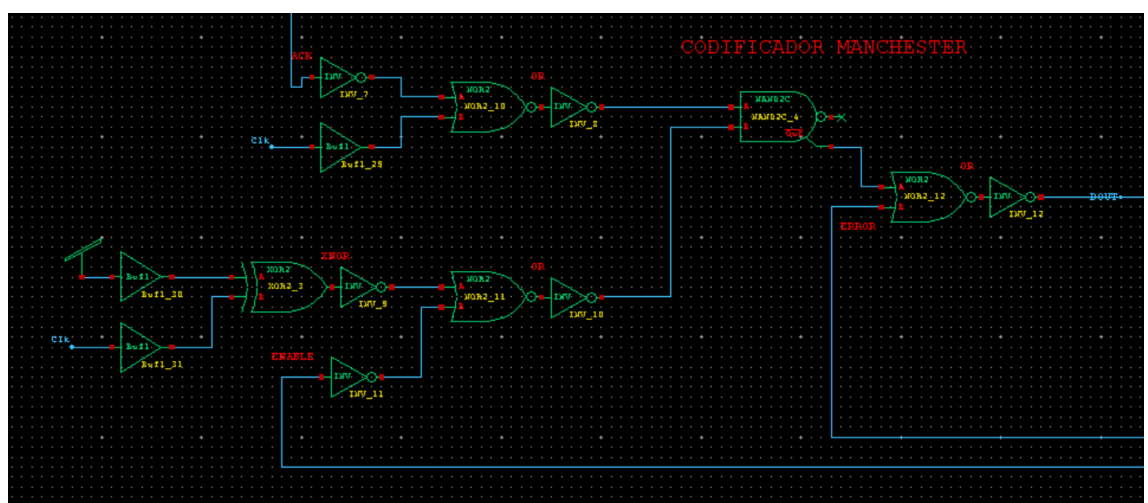
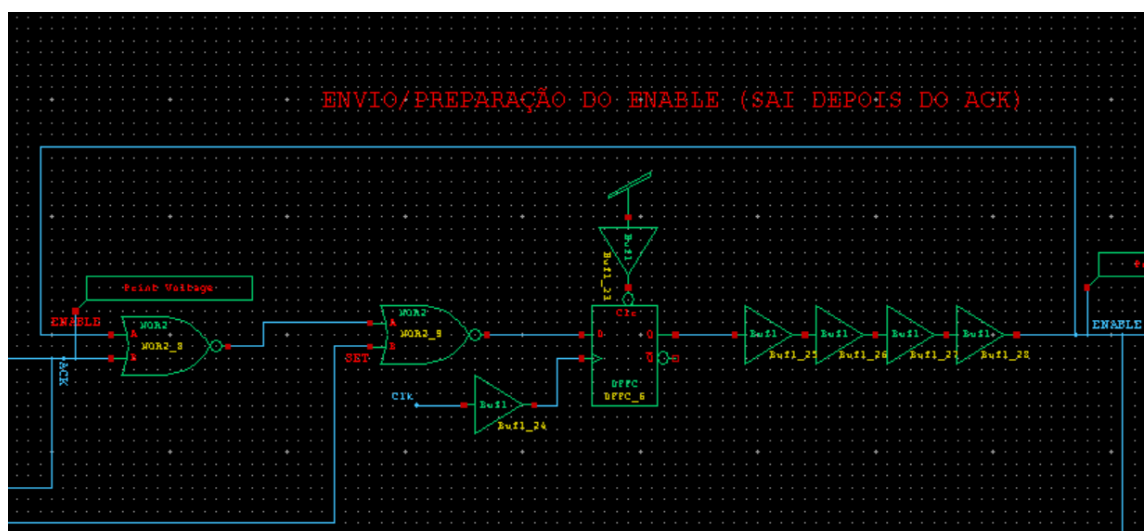
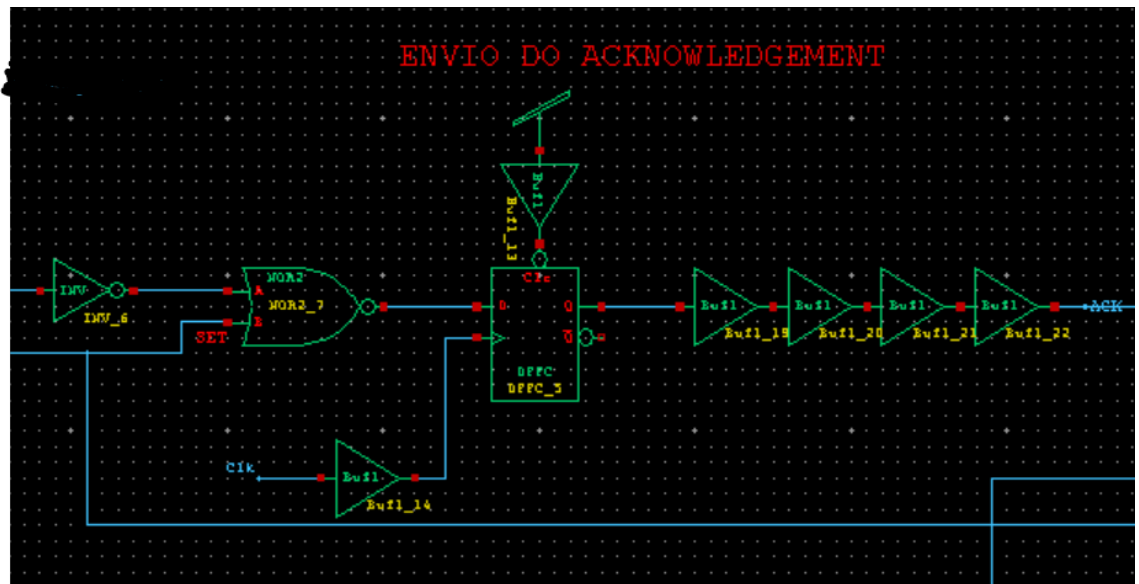


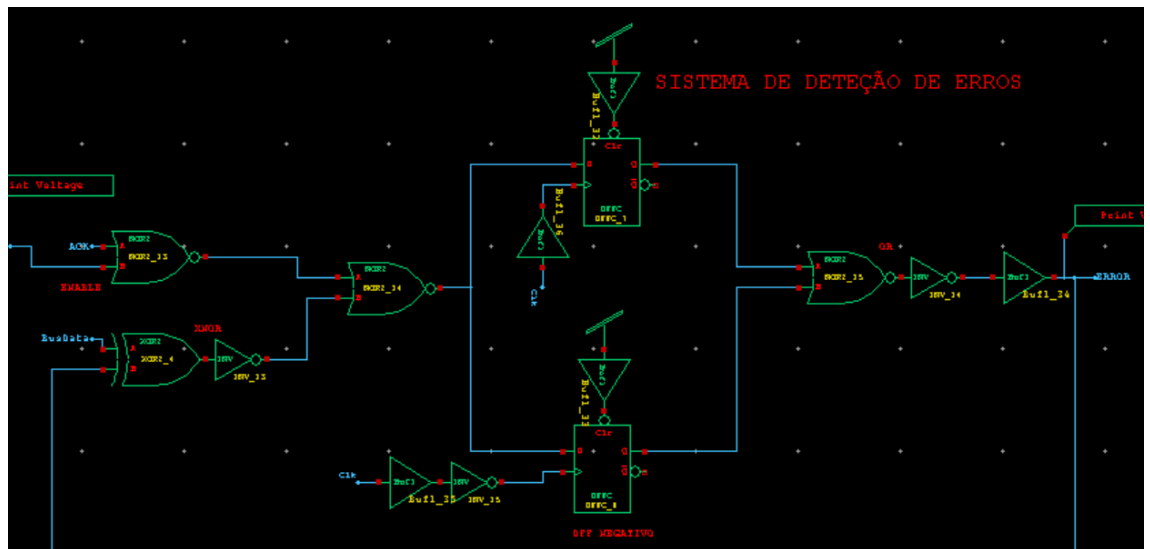
Aula 16-12-2021

- Os sinais que temos mostrar na avaliação são os seguintes: (ver melhor esta sequência)
 - BusData
 - Clock (não é necessário colocar)
 - ACK -> ao mesmo tempo que gerar o start tem que gerar o ack (vai começar com 11 mas isso é irrelevante)
 - Dout ->
 - ENABLE -> é o terceiro e os dados a partir daqui vão começar a sair, é igual ao busdata até o ack = 1, e aí vai começar a enviar os dados (101010 -> porque foi o que nós decidimos)
 - START -> o segundo a ver é este (tem que sair no Q negado)
 - STOP ->
 - Q_Latch -> é a primeira coisa que temos que ver se está direitinho
 - ERROR
- Vai dar o erro porque é uma saída e depois vamos dar uma entrada (cagar nisto)
- O resultado que temos que obter no final é este (o gráfico do professor:

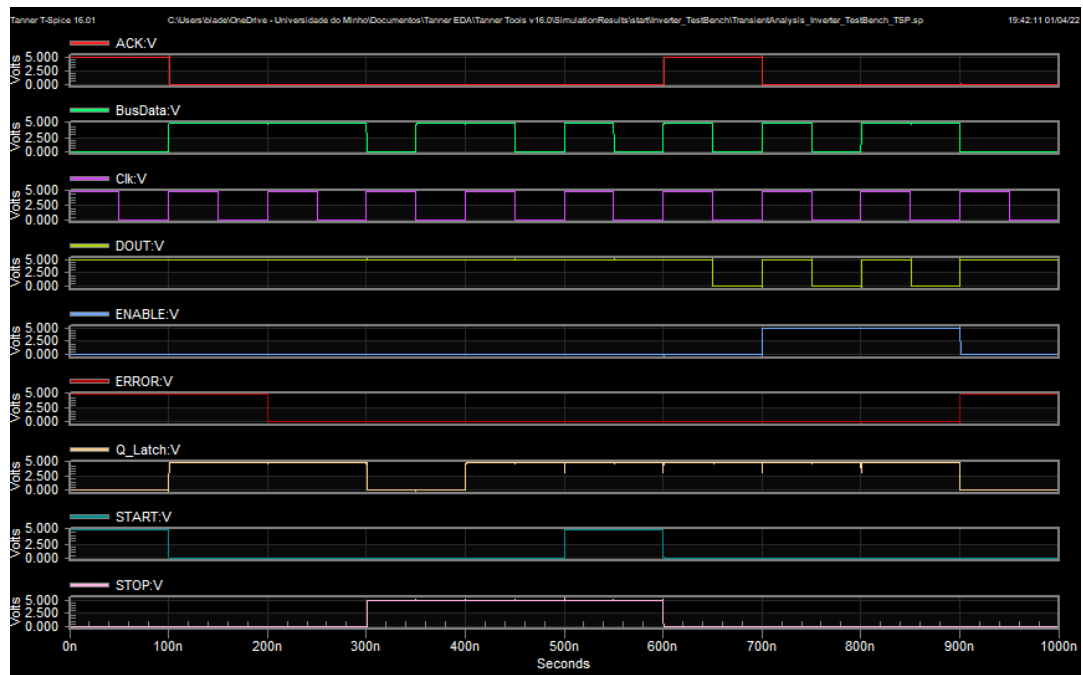


- Os circuitos dados pelo professor esta aula são os seguintes:





O gráfico que obtive foi o seguinte:



NOTA: ter em atenção que tudo o que está no intervalo de 0n a 100n segundos não importa para nada, ignorar este intervalo