

UNIVERSIDADE DE SÃO PAULO

Escola de Engenharia de São Carlos
SEL0621 - Projetos de Circuitos Integrados Digitais I
Prof. Dr. João Pereira do Carmo

Projeto 6

Davi Diório Mendes	7546989
Nivaldo Henrique Bondança	7143909



9 de agosto de 2014

Lista de Figuras

1	Porta lógica <i>CMOS</i> . Implementa a função $\overline{AB+C}$	p. 5
---	--	------

Lista de Tabelas

Códigos Fontes

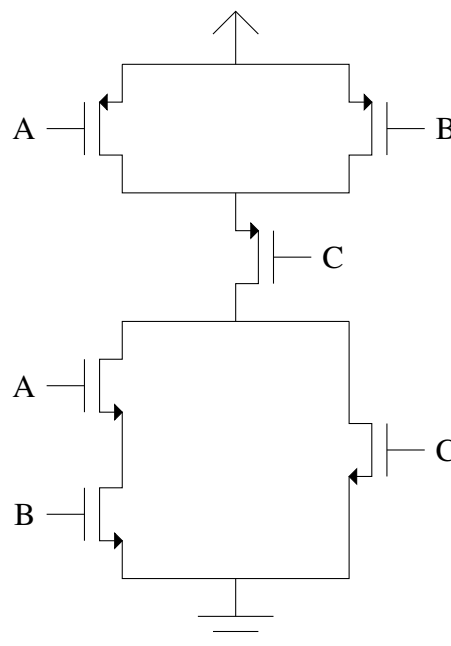
Nesta experiência é dada continuidade ao que foi ensinado no semestre anterior. Será visto aqui como se extrai o *netlist* para simulação já a partir do esquemático, permitindo a verificação da sua funcionalidade. Também será visto como fazer a comparação entre *layout* e esquemático. Por fim é introduzida a simulação do tipo Monte Carlo que possibilita analisar o comportamento dos circuitos com as variações dos parâmetros dos transistores.

1. *Considere a porta lógica *CMOS* estática que implementa a função lógica $\overline{(ab + c)}$. Tendo o transistor *NMOS* de menor dimensão um $W = 2\mu m$ e $L = 0,35\mu m$, determine as dimensões de todos transistores de forma que:

- i (atraso de propagação na descida com $ABC = "110"$) = (atraso de propagação na descida com $ABC = "001"$) = (pior atraso de propagação na subida);
- ii Todos transistores *PMOS* tenham as mesmas dimensões.

(deixe indicado os valores usados)

Figura 1: Porta lógica *CMOS*. Implementa a função $\overline{AB + C}$.



Seja o tempo de propagação dado pela seguinte relação:

$$t_p = \frac{1,6 \cdot C_L}{\mu \cdot \left(\frac{W}{L}\right) \cdot C_{ox} \cdot V_{DD}} \quad (1)$$

No caso 110, teremos os transistores M_4 e M_5 abertos, realizando a descida do sinal. Para o cálculo tempo de propagação na descida, teremos um transistor equivalente com as seguintes

dimensões:

$$W_{eq} = W_{M_4, M_5} \quad (2)$$

$$L_{eq} = 2 \cdot L \quad (3)$$

Por outro lado, no caso 001, somente o transistor M_6 abre. Desta forma as dimensões equivalentes são as mesmas dimensões do transistor M_6 :

$$W_{eq} = W_{M_6} \quad (4)$$

$$L_{eq} = L \quad (5)$$

Desta forma temos:

$$t_{pHL(110)} = \frac{1,6 \cdot C_L}{\mu_N \cdot \left(\frac{W_{M_4, M_5}}{2 \cdot L} \right) \cdot C_{ox} \cdot V_{DD}} \quad (6)$$

$$t_{pHL(001)} = \frac{1,6 \cdot C_L}{\mu_N \cdot \left(\frac{W_{M_6}}{L} \right) \cdot C_{ox} \cdot V_{DD}} \quad (7)$$

e, ao igualarmos (6) e (7), eliminaremos todas as variáveis não abordadas até o momento, obtendo a relação:

$$\frac{W_{M_4, M_5}}{2 \cdot L} = \frac{W_{M_6}}{L} \quad (8)$$

$$\Rightarrow W_{M_4, M_5} = 2 \cdot W_{M_6} \quad (9)$$

Como $W_{M_6} < W_{M_4, M_5} \Rightarrow W_{M_6} = 2\mu m$. Logo:

$$W_{M_4, M_5} = 2 \cdot 2\mu m = 4\mu m \quad (10)$$

$$\Rightarrow W_{M_4} = W_{M_5} = 4\mu m \quad (11)$$

Para o análise dos transistores *PMOS*, tomamos como os piores casos $ABC = 101$ ou $ABC = 011$. Os transistores equivalente — utilizado no cálculo do tempo de propagação — de ambos os casos possuem as mesmas dimensões, mostradas a seguir:

$$W_{eq} = W_P \quad (12)$$

$$L_{eq} = 2 \cdot L \quad (13)$$

Desta forma, podemos modelar o pior atraso de propagação na subida como:

$$t_{pLH(101|011)} = \frac{1,6 \cdot C_L}{\mu_P \cdot \left(\frac{W_P}{2 \cdot L}\right) \cdot C_{ox} \cdot V_{DD}} \quad (14)$$

Igualando as equações (7) e (14), analogamente ao que foi feito para os transistores *NMOS*, podemos calcular W_P :

$$W_P = \frac{\mu_N}{\mu_P} \cdot W_{M_6} \quad (15)$$

$$\Rightarrow W_P = 12,70 \mu m \quad (16)$$

para $\mu_N = XX cm^2/Vs$ e $\mu_P = XX cm^2/Vs$.

2. Faça o circuito esquemático da porta *CMOS* e gere seu símbolo. Faça todas as verificações necessárias no esquemático e no símbolo não deixando nenhum erro ou *warning*. Não esqueça de ligar o *bulk* dos transistores.

3. Crie para o esquemático um *designview* point para poder gerar arquivos para simulação e comparar com *layout*. Para isso execute o comando **Hit-Kit Utilites – Create Viewpoint** na parte superior da janela do *Design Architecture*. No menu que aparece coloque o nome de seu arquivo esquemático no *Design Path*. Deixe a opção *device*.

4. Agora gere um arquivo *netlist* para o *ELDO*. Para isso entre no *Simulation* mode (coluna esquerda da janela do *Design Architecture*, último comando). No menu selecione o *vpt_c35b4_device*, que foi gerado no item anterior, e dê *OK*.

Obs.: A partir da nova janela podemos retornar à anterior executando novamente o último comando da coluna à esquerda.

5. *Gere o *netlist* executando o comando apropriado na coluna à esquerda. Com outro comando nessa coluna, o *ASCII Results*, verifique os resultados na opção *view netlist*. Acrescente o *netlist* ao relatório.

6. *Como são calculadas as áreas e perímetros de dreno e *source* no circuito extraído pelo esquemático (relação usada)?

7. Observe que no *netlist* não aparecem as capacitâncias parasitas que são geradas quando é feita a extração do *layout*. Prepare um arquivo para simulação e, com os parâmetros típicos, $V_{DD} = 3V$, determine o atraso de propagação na subida *versus* capacitância de carga (entrada com onda quadrada com *rise/fall time* pequenos (1% do período, por exemplo)). Escolha os sinais de entrada de forma a obter a pior situação, ao menos cinco valores para a capacitância

de carga, e escolha o período dos sinais de forma a obter resultados corretos. Faça o mesmo para o atraso de propagação na descida *versus* capacitância de carga.

8. *Apresente os gráficos da questão anterior e copie os comandos de medida e sinais de entrada que usou no *ELDO*.

9. Faça agora o *layout* da porta (utilize o *designview point* gerado para geração). No *layout* deve-se tomar cuidado com:

- área total do circuito;
- o uso correto dos metais e poli como camadas de conexão;
- a posição dos *ports* de entrada e saída.

Faça a verificação com o *DRC (CALIBRE)* e elimine todos os erros.

Obs.: Veja as opções do **Route – ARoute Commands – Setup – Display** para melhor fazer o roteamento manual.

10. Utilize o comando **Connectivity – port – Add to Port** para ampliar as áreas dos *ports*. Para isso selecione o *shape* que deseja acrescentar a um *port* e então execute o comando.

11. *Como se pode acrescentar aos *ports* V_{DD} e V_{SS} as regiões de *source* dos transistores sem transformarmos os transistores em *flatten*?

12. Uma vez acrescentadas aos *ports* todas as regiões desejadas, faça nova verificação com o *DRC*.

13. Vamos agora fazer a última verificação do circuito: comparação entre o *layout* e o esquemático que o gerou (*Layout vs. Schematic* ou simplesmente *LVS*). Para isso, dentro do *ICStation* feche o circuito lógico associado ao *layout* (**File – Logic – close**, menu superior). Execute então os comandos *IcTrace(M)* e *LVS* (menu à direita). No menu que aparece complete o *source name* (nome do *netlist* para comparar que esta no ... logic.view/name/.../vpt_c35b4_device) e na opção *Abort on Supply Error*, deixe *NO*. Verifique a função das outras opções que estão disponíveis.

14. Ao dar o *OK*, é feita a comparação entre os *netlists* extraídos do *layout* e do esquemático. Para ver se há ou não erros execute o comando ***IcTrace(M) – LVS – Report – LVS***. Verifique também as outras opções no menu de *Report*. Caso não houver erros aparecerá a "*carinha feliz*" se houver, serão fornecidas informações sobre os erros.

15. Para determinar onde estão os erros pode ser usado o comando ***IcTrace(M) – discrepancies***. A opção *first* aí mostra o primeiro erro; a opção *next*, o próximo. Caso tenha tido algum

erro tente achá-lo com esses comandos; se não teve erros, modifique o *layout* (apague alguma conexão) para poder praticar (caso não pratique na aula poderá ter surpresas na prova).

Obs.: o comando *discreps* deixa selecionada uma ou mais regiões do circuito. Essas regiões selecionadas são deselecionadas apenas pelo comando ***IcTrace(M) – unshow – all*** . Algo similar acontece com o *DRC*. Obs.: O comando ***IcTrace(M) – netlist*** também serve, como o *PEX (CALIBRE)*, para gerar *netlists* para o *ELDO*.

16. *Uma vez feitas as verificações com *DRC* e *LVS*, caso não tenha sido encontrado nenhum erro, o *layout* estará pronto para uso. Agora, extraia o circuito de simulação a partir do *layout* (opção C+CC) e repita as simulações feitas no item 7. Apresente os gráficos com resultados (gere uma figura do *layout* e inclua no trabalho).

17. *Para as curvas atraso de propagação na subida e descida *versus* carga, geradas a partir do *layout*, calcule as inclinações e o pontos de cruzamento com o eixo Y (eixo de tempo).

18. *Comente as diferenças entre os resultados encontrados nas questões 8 e 16/17? Dê as razões para elas.

19. *Faça um inversor com $W_N = 2\mu m$ e $L_N = 0,35\mu m$. Faça o esquemático, símbolo e *layout*. Passe as verificações no esquemático e símbolo. O *layout* deve ser feito com cuidado para ter área pequena, utilização correta de metais/poli e *ports* de tamanho conveniente. Passe o *DRC* no *layout* e faça o *LVS* deixando a célula pronta para uso. Acrescente ao relatório o *layout* feito.

20. A partir das duas células desenhadas, monte o esquemático de uma nova célula que executa a função lógica $(ab + c)$. Gere o seu símbolo e faça todas as verificações necessárias.

Obs.: antes de realizar o item 20, deve-se acrescentar aos símbolos anteriores a propriedade *phy_comp* com a posição do *layout* de cada célula.

21. Gere a partir do esquemático o arquivo para simulação com o *ELDO* (acrescente o *netlist* ao relatório). Simule com os parâmetros típicos, $V_{DD} = 3V$, e determine o atraso de propagação na subida *versus* capacitância de carga (entrada com onda quadrada com *rise/fall time* igual a 1%). Escolha os sinais de entrada de forma a obter a pior situação, ao menos cinco valores para a capacitância de carga e escolha o período dos sinais de forma a obter resultados corretos. Faça o mesmo para o atraso de propagação na descida *versus* capacitância de carga.

22. *Desenhe os gráficos da questão anterior e copie os comandos de medida e sinais de entrada que usou no *ELDO*.

23. Faça o *layout* final da célula. Utilize nas linhas de metal que ligam o V_{DD} e o V_{SS} largura

sempre superior ou igual a $1,0\mu m$. Par isso veja e utilize o comando **Route – ARoutre NEt Classe – Edit** que permite especificar as características de conexão de qualquer sinal. Utilize a opção *New/Edit* para fornecer as características desejadas e a opção *Assign* para associá-las a um sinal (e apenas um). Coloque os sinais de V_{DD} e o V_{SS} com metais de $1,0\mu m$.

Obs.: Veja que quando se esta executando o comando **Route – Iroute Commands – Run**, a tecla *w* pode ser utilizada para alterar a largura da linha desenhada.

24. *Termine layout da célula, passe o *DRC* e faça o *LVS*. Gere uma figura do *layout* mostrando todos os níveis e inclua no trabalho.

25. *Agora extraia o circuito de simulação a partir do *layout* (opção *C+CC*) e repita as simulações feitas no item 22. Apresente gráficos e tabelas com os resultados.

26. *Para as curvas tempo de propagação na subida e descida geradas a partir do *layout*, calcule as inclinações e os pontos de cruzamento com o eixo *Y* (eixo de tempo).

27. *Gere novamente os tempos de propagação na subida e descida utilizando agora os comandos (faça os ajustes necessários para seu circuito)

```

1 Va  a 0 3V
2 Vb  b 0 0
3 Vc  c 0 pulse (0 3 0 1p 1p 2n 4n)
4 . tran 1n 40n 0n 1p
5 .meas tran delayF trig v(c) val=1.5 fall=6 targ v(out) val
   =1.5 fall=6
6 .meas tran delayR trig v(c) val=1.5 rise=6 targ v(out) val
   =1.5 rise=6
7 Cl out 0 30fF

```

28. *Vamos realizar agora a *simulação de Monte Carlo*. Nesta simulação são realizadas, na verdade, várias simulações com parâmetros diferentes e podemos conhecer o comportamento do circuito para diversas condições de fabricação. Utilizando os comandos abaixo realize *Monte Carlo* (faça os ajustes necessários para seu circuito, não coloque no arquivo o modelo do transistor). Forneça os gráficos da tensão em *c* e na saída (valor típico e piores casos) e os gráficos do número de saídas *versus delayF* e *delayR*.

```

1 Va  a 0 3V
2 Vb  b 0 0
3 Vc  c 0 pulse (0 3 0 1p 1p 2n 4n)
4 . tran 1n 30n 0n 10p

```

```
5 .meas tran delayF trig v(c) val=1.5 fall=5 targ v(out) val  
   =1.5 fall=5  
6 .meas tran delayR trig v(c) val=1.5 rise=5 targ v(out) val  
   =1.5 rise=5  
7 Cl out 0 30fF  
8  
9 .option SST_MTHREAD=1  
10 * MONTE CARLO  
11 .MC 100 NBBINS=20  
12 .INCLUDE /local/tools/dkit/ams_3.70_mgc/eldo/c35/profile.opt  
13 .LIB /local/tools/dkit/ams_3.70_mgc/eldo/c35/wc53.lib mc
```

29. *Utilize o comando **Report – Windows** do *ICStation* para determinar o tamanho da célula (coloque o tamanho de sua célula no relatório).

manual do Mentor-ELDO: /local/tools/mentor/shared/pdfdocs/eldo_ur.pdf

modelos dos transistores: /local/tools/dkit/ams_3.70_mgc/eldo/c35.