

## PSI-3452- Projeto de Circuitos Integrados Digitais e Analógicos

### Projeto (Parte 2): Edição de Leiaute do DAC e Simulações Parciais

#### 1 – Os blocos do projeto.

Após o dimensionamento realizado na Semana 1 de projeto, com a confirmação pelas simulações, os grupos devem passar ao projeto do leiaute das diversas partes do circuito DAC. A minimização da área, isto é a compactação do leiaute, é fundamental no desenvolvimento de qualquer leiaute e deve ser uma preocupação constante neste projeto (será um dos aspectos levados em conta na avaliação). Use todos os seus conhecimentos adquiridos no semestre para o desenvolvimento do projeto.

A simulação do circuito extraído a partir do leiaute deve ser feita para certificar-se da correção de seus desenvolvimentos. Apesar de ser desejável uma separação de atividades (responsabilidades) entre os componentes de um grupo, todos devem estar acompanhando e com conhecimento de todos os desenvolvimentos em paralelo.

Replicamos na Figura 1 o diagrama apresentado no documento de especificação do DAC de 8 bits para facilitar a referência.

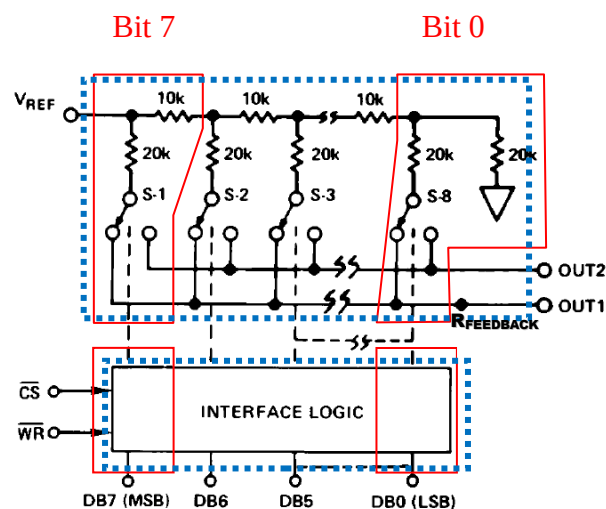


Figura 2. Célula de um bit da rede R/2R e chave.

Do ponto de vista técnico, o circuito divide-se em duas partes (pontilhado azul):

- A rede R/2R com as chaves (deve ser de 4 bits), visto na parte superior da figura;
- O módulo lógico (deve ser de 4 bits), na parte inferior.

Para cada uma das partes acima, os grupos deverão realizar o projeto de forma modular, isto é, desenvolver os leiautes dos blocos individuais (de cada bit, por exemplo) e integrá-las, depois, gerando o módulo completo.

#### 2 – A rede R/2R e as Chaves

Projetar e desenhar o leiaute da célula R/2R e Chave (transistores N de passagem) por bit. Deve-se utilizar os valores obtidos na Semana 1 de projeto. Veja pela Figura 2 os portos de uma célula de um bit R/2R com uma chave.

Deve-se observar que as entradas BIT\_IN e ^BIT\_IN a sinais gerados no módulo digital, correspondendo aos bits (DB0 a DB3). Os dois sinais devem chegar nos gates do par de transistores na forma de um no nível lógico de DBx e outro na forma complementada.

Observe-se que o nó GND é necessário para a polarização do substrato dos transistores de passagem. Observe também que a célula apresenta um estilo conhecido como *bit-slice*; com o devido posicionamento de linhas de metal na horizontal, de lado a lado, células poderão simplesmente ser justapostas para criarem um circuito de vários bits.

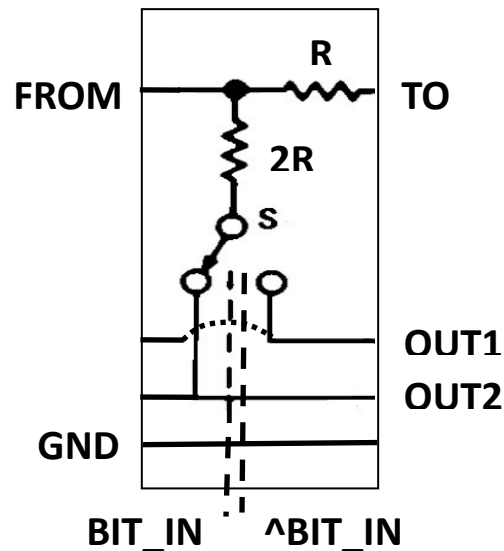


Figura 2. Célula de um bit da rede R/2R e chave.

Sequência de tarefas:

- 1) Construir o leiaute da célula com a geração dos resistores R e 2R, mais dois transistores das chaves;
- 2) Extrair o circuito (**No-RC, sem parasitários**) para confirmar valores dos componentes.
- 3) Caso (é provável) o valor do R extraído for diferente do requerido (projetado), por proporcionalidade, calcule e o novo valor a ser utilizado no gerador de resistores do IC Station e crie uma célula modificada. Repita o passo 2) até que o valor de R extraído seja o correto.
- 4) Juntar as células e denominar os nós do bloco completo para simulação (semelhante ao realizado para a Parte 1, da semana anterior);
- 5) Extrair o circuito completo (a sugestão é fazer **inicialmente No-RC** pra verificar funcionalmente apenas - futuramente para medidas de atraso, deve-se usar o **RC...**)
- 6) Simular com o Eldo o circuito completo de maneira similar à simulação final feita na Semana 1.
- 7) Verificar e reportar a correção pelo EzWave.

### 3 – Módulo Lógico

O módulo lógico consiste de um banco de latches com dois sinais de controle,  $\overline{\text{WR}}$  (active-low write) e  $\overline{\text{CS}}$  (active-low chip-select), além de inversores na saída pra cada latch. Denominaremos o circuito de latch com inversor como **latch\_completo** para diferenciá-lo do latch simples tradicional. Este é um registrador ativado por clock, sensível a nível (diferente de um flip-flop, sensível à borda do clock). Usaremos o termo simples **latch** para denominá-lo.

A Figura 3 apresenta a funcionalidade do módulo digital em relação aos sinais de controle para os latches\_completos. Resumindo, somente quando ambos os sinais  $\overline{\text{WR}}$  e  $\overline{\text{CS}}$  estiverem ativos (em 'L'), a saída do banco de latches assumirá o valor na entrada. É então necessário desenvolver uma lógica para adaptar os sinais de entrada  $\overline{\text{WR}}$  e  $\overline{\text{CS}}$  para o sinal de **clock** do **latch**, pois este servirá como **um enable**.

MODE SELECTION TABLE

$\overline{\text{CS}}$	$\overline{\text{WR}}$	Mode	DAC Response
L	L	Write	DAC responds to data bus (DB0-DB7) inputs.
H	X	Hold	Data bus (DB0-DB7) is Locked Out:
X	H	Hold	DAC holds last data present when $\overline{\text{WR}}$ or $\overline{\text{CS}}$ assumed HIGH state.

L = Low State, H = High State, X = Don't Care.

Figura 3. Tabela com a funcionalidade do módulo digital

Como o **latch tradicional** é controlado apenas pelo sinal de clock, cabe ao grupo obter uma lógica adicional (bloco **adaptador**) e incluí-la ao módulo digital, que habilitará o **latch** via o sinal de **clock**, para que funcione de acordo com a especificação da tabela. Esta lógica adicional pode ter o seu leiaute realizada e posicionada de várias maneiras (construído como NORs, inversores, etc..., buscadas na biblioteca de células). A Figura 4 (b) apresenta uma possível solução que permite uma futura justaposição direta, mas fica a critério dos grupos apresentarem uma satisfatória.

Convém lembrar que a entrada D estará conectada a um sinal digital DBn de entrada do DAC. A saída Q, por sua vez estará conectada a um inversor (este deve ser qualquer um dos projetados durante o semestre pelos integrantes do grupo, porém a escolha deve ser reportada). Assim o sinal Q e o seu complemento serão saídas do bloco digital e seguirão diretamente às portas dos transistores de passagem (chave) do módulo R/2R.

Como trata-se de um módulo digital de 4 bits independentes, os grupos deverão desenvolver o leiaute de uma célula de um latch\_completo de um bit, para depois conectar todos entre si.

Existe na biblioteca de células TSMC 0,35 $\mu\text{m}$  o leiaute pronto de um latch tradicional. Trata-se de uma célula-padrão com a geometria de seus portos mostrada na Figura 4 (a). A justaposição de células faz com que as linhas de VDD e GND (em Metal 1) fiquem interconectadas enquanto as entradas e saídas. Além do sinal de clock, são alcançáveis por linhas verticais (vias prontas para uso com Metal 2).

É requisito para esta etapa:

- a lógica adicional do **adaptador** deve usar as mesmas dimensões dos transistores simulados na Semanal1.
- minimização de área pela compactação das células..

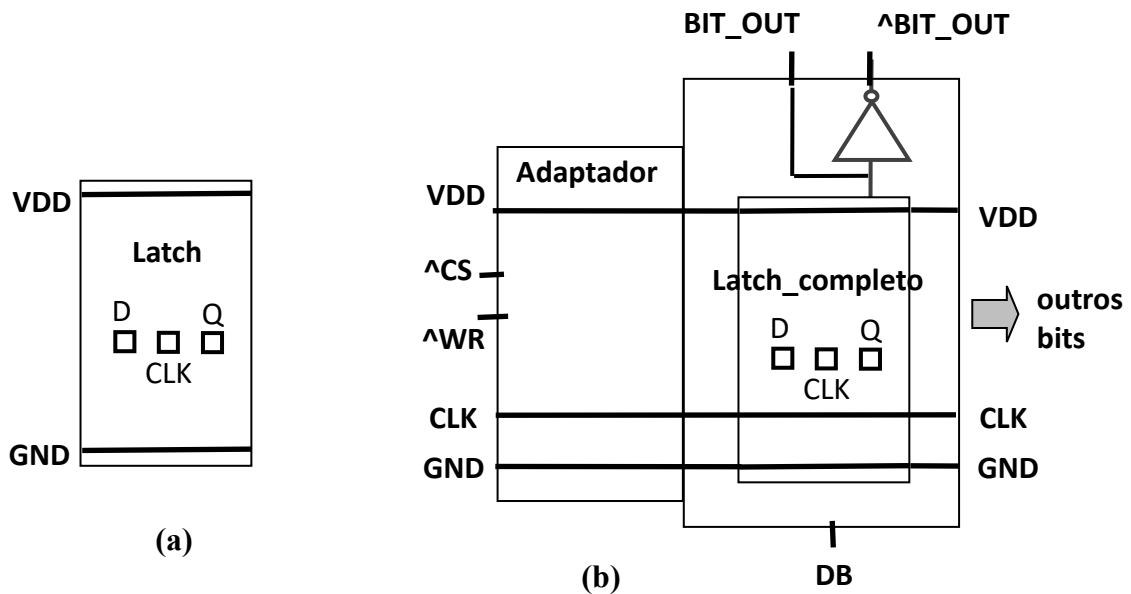


Figura 4: (a) Representação do leiaute de latch da biblioteca TSMC035; (b) Representação de leiaute possível de latch\_completo+adaptador.

Observe que a célula apresentará um estilo *bit-slice* similar ao do bloco R/2R com o devido posicionamento de linhas de metal na horizontal, de lado a lado. As células poderão simplesmente ser justapostas para criarem um circuito de vários bits.

Sequência de tarefas:

- 1) Projetar e apresentar o esquema lógico de um latch\_completo a partir de um latch simples, com a inclusão do inversor;
- 2) Abrir um projeto de leiaute e incluir a célula de latch (com clock) da biblioteca ADK e a célula do inversor;
- 3) Extrair o circuito elétrico do **latch\_completo de 1 bit (No-RC)** para confirmação da funcionalidade;
- 4) Simular este latch\_completo com o Eldo, para confirmação da funcionalidade;
- 5) Verificar e reportar a correção pelo EzWave.
- 6) Juntar as células formando o módulo lógico, incluindo-se a lógica de adaptação CS, WR para CLK.;
- 7) Extrair o módulo digital completo (**No-RC**) para confirmação da funcionalidade;
- 8) Simular com o Eldo o módulo digital completo para confirmação da funcionalidade;
- 9) Verificar e reportar a correção pelo EzWave.