**RELATÓRIO FINAL SOBRE FUNCIONAMENTO DO BLOCO DETECTOR DE FASE DE UM PLL E IMPLEMENTAÇÃO EM LAYOUT**

**Santa Maria, 08 de julho de 2019**

Giuliano Bohn Benedeti Becker[[1]](#footnote-0)

Keli Tauana Prass Ruppenthal[[2]](#footnote-1)

Victor Dallagnol Bento[[3]](#footnote-2)

***Resumo: Neste relatório trataremos do desenvolvimento dos layouts referentes aos componentes do bloco do detector de fase. As análises serão baseadas em implementações do bloco em questão feitas na ferramenta Virtuoso/Cadence.***

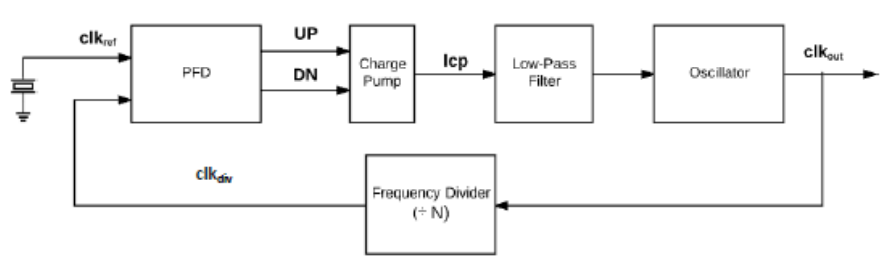
*Palavras-chave: detector de fase, up, down, layout.*

**1. Introdução**

O objetivo por trás deste relatório é, primeiramente, o funcionamento de um PLL (*Phase Locked Loop*), o qual pode ser definido como um dos mais importantes de todos os circuitos que atualmente encontramos em aplicações eletrônicas de todos os tipos. O PLL trabalha basicamente com frequências, da mesma forma que um amplificador operacional trabalha com tensões, o que demonstra a grande importância deste na eletrônica. Além disso, o relatório tratará ainda das implementações do bloco em nível de layout, com todos os seus componentes devidamente construídos para que possa ser feita uma análise mais completa do real funcionamento do PLL.

Como já mencionado antes, existem inúmeras aplicações dos PLLs, como exemplos temos os receptores de AM e FM, modems, sintetizadores de frequências, telefones sem fio,telefones celulares, instrumentos digitais e analógicos, ou seja, o PLL se encontra em uma infinidade de aplicações onde o valor de frequência esteja presente e seja relevante.

Porém, o foco deste relatório é na análise e construção de layout para um dos blocos responsáveis pelo funcionamento do PLL, o detector de fase (*Phase Detector*). O PD, como pode ser chamado, tem como principal função controlar a carga/descarga do capacitor no próximo bloco (*Charge Pump*). Os sinais UP/DN (*Up* e *Down*) são responsáveis por essa comunicação entre os blocos, sendo responsáveis basicamente por controlar a frequência. Veremos a seguir como ocorre de fato esta análise de frequências e a sua implicação. Na **Figura 1** observa-se o esquemático de um circuito PLL.

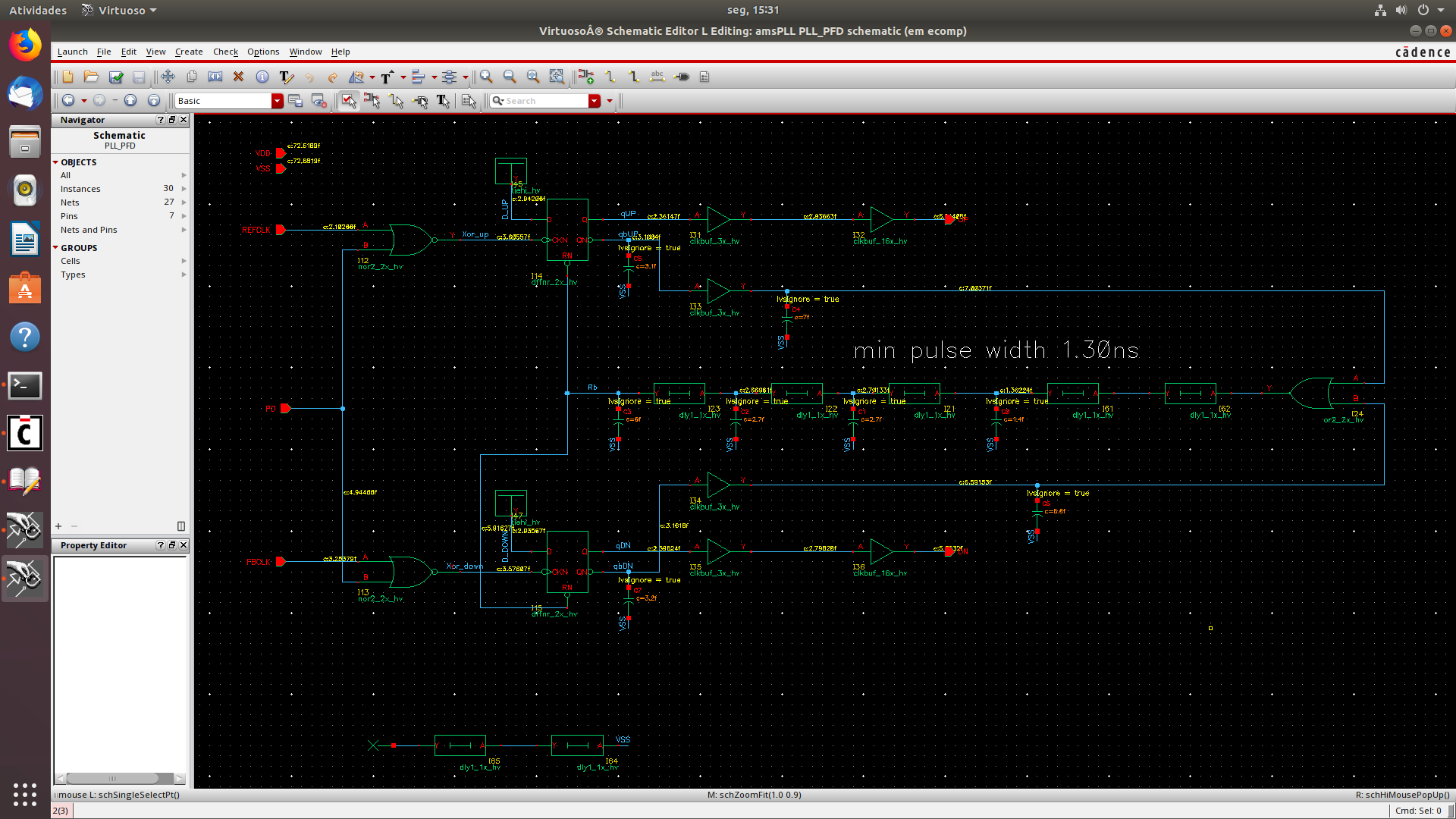


**Figura 1:** Esquemático geral do PLL implementado.

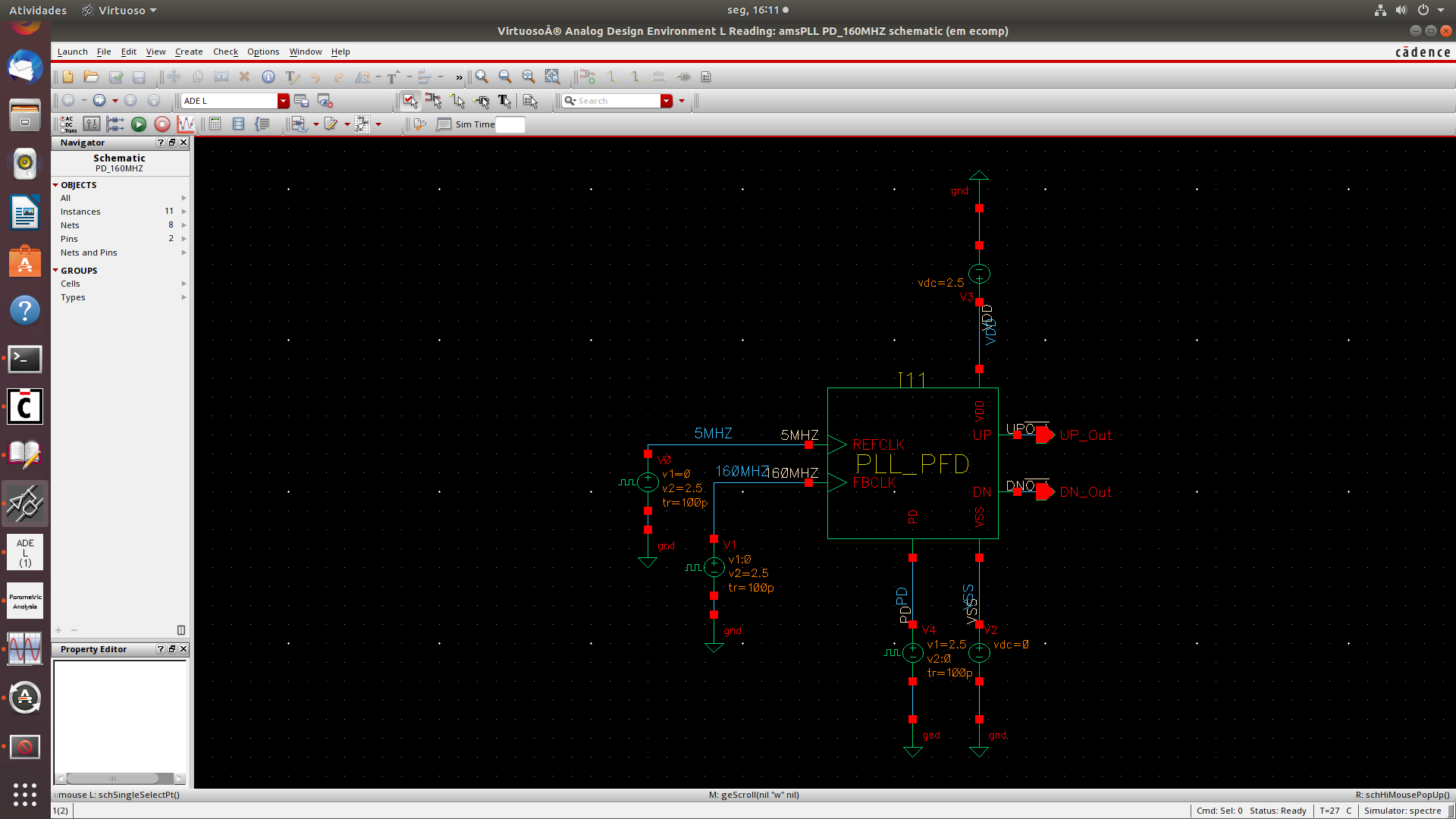
**2. Desenvolvimento Teórico**

De modo geral, o detector de fase é responsável por comparar o erro existente entre as duas frequências de entrada e sinalizar para o próximo bloco se a frequência deve ser aumentada ou diminuída. Caso a frequência de feedback seja menor do que a de referência (neste caso 5MHz), o circuito irá gerar impulsos de UP (up), para aumentar a tensão de controle que vai para o Charge Pump. De modo inverso, se a frequência de feedback vinda do VCO for maior do que a frequência de referência, o circuito PD irá detectar esta diferença e irá gerar mais pulsos de DN (down). O detector de fase pode ser utilizado em qualquer lugar em que seja necessário detectar a fase entre dois sinais.

A **Figura 2** demonstra o esquemático do circuito que será feito o layout, enquanto que a **Figura 3** mostra o bloco externamente. Nota-se que a maioria dos componentes são de tamanhos diferentes, sendo necessário a divisão do grupo para que o processo pudesse ficar pronto a tempo. Dessa forma, o Latch (por se tratar de um componente maior) teve dois integrantes envolvidos, enquanto que …….



**Figura 2**: Esquemático completo do circuito PD.

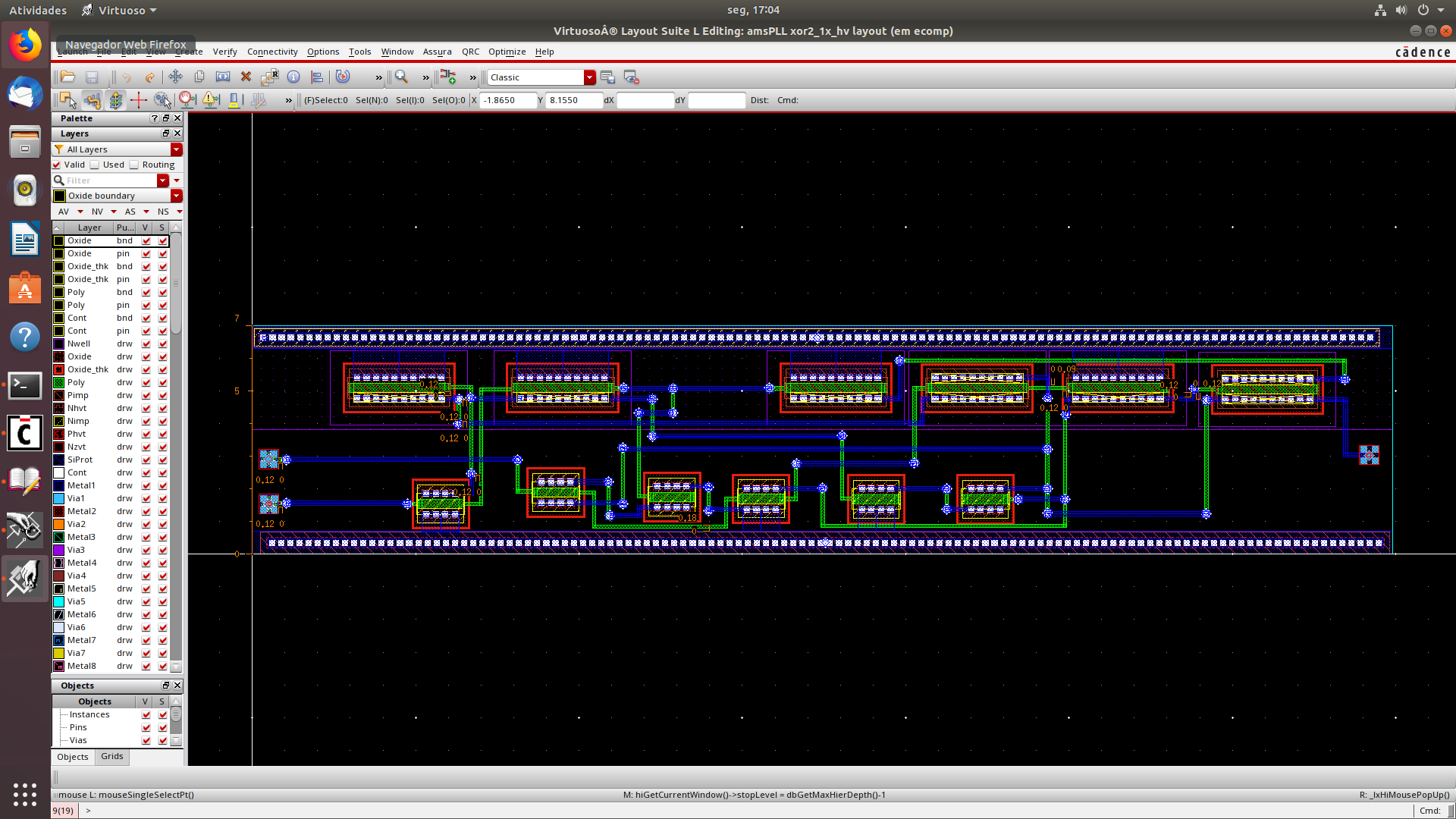


**Figura 3:** Bloco PD construído em nível de esquemático.

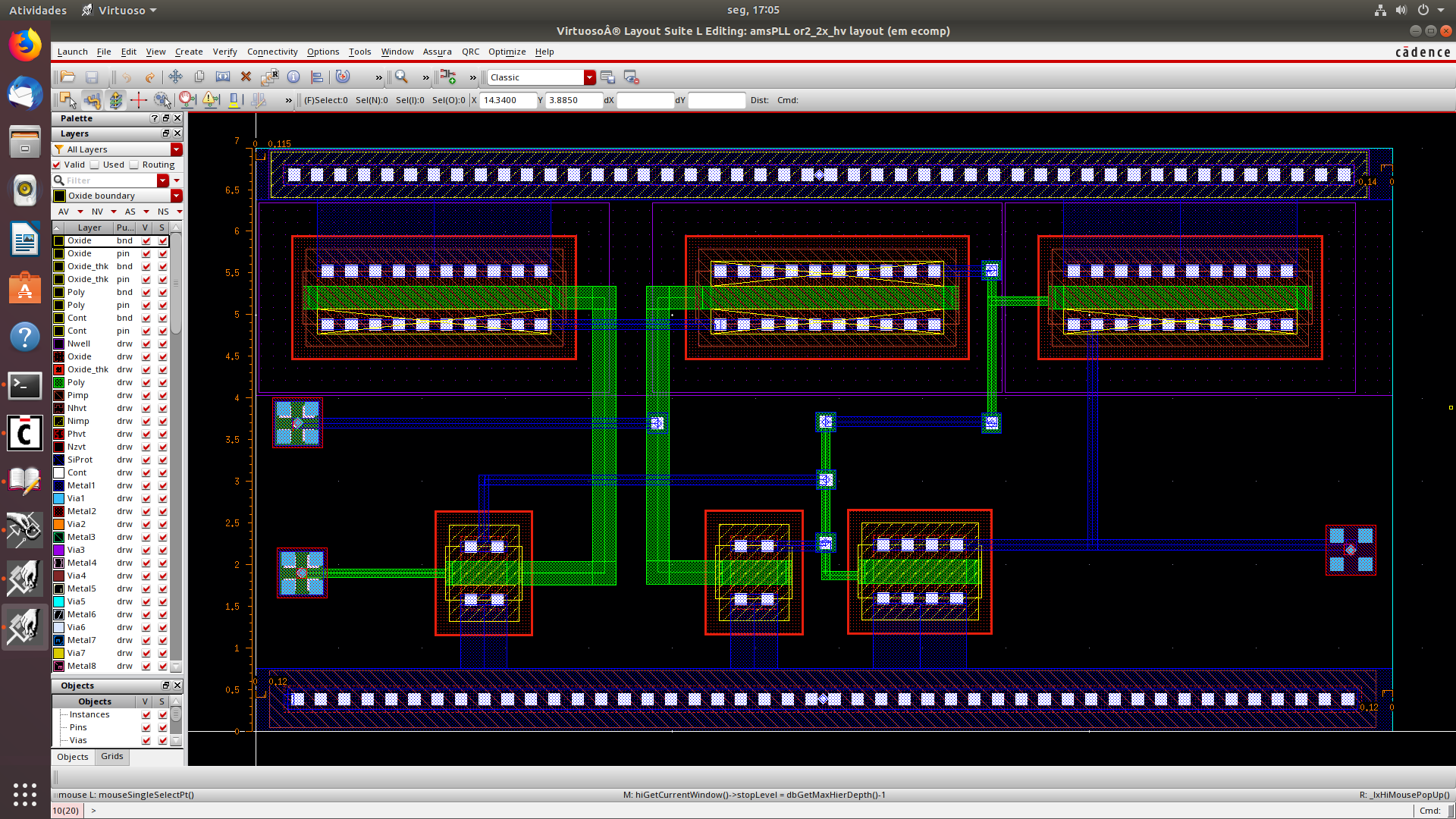
**3. Desenvolvimento Prático**

Nesta seção, trataremos basicamente da implementação dos layouts na ferramenta Virtuoso. As partes que serão mostradas a seguir são referentes a todos os layouts construídos pelo grupo. No servidor, o caminho com a pasta do projeto implementado é */home/cci2018/ktpruppenthal/Proj\_PLL/PLL\_PFD*.

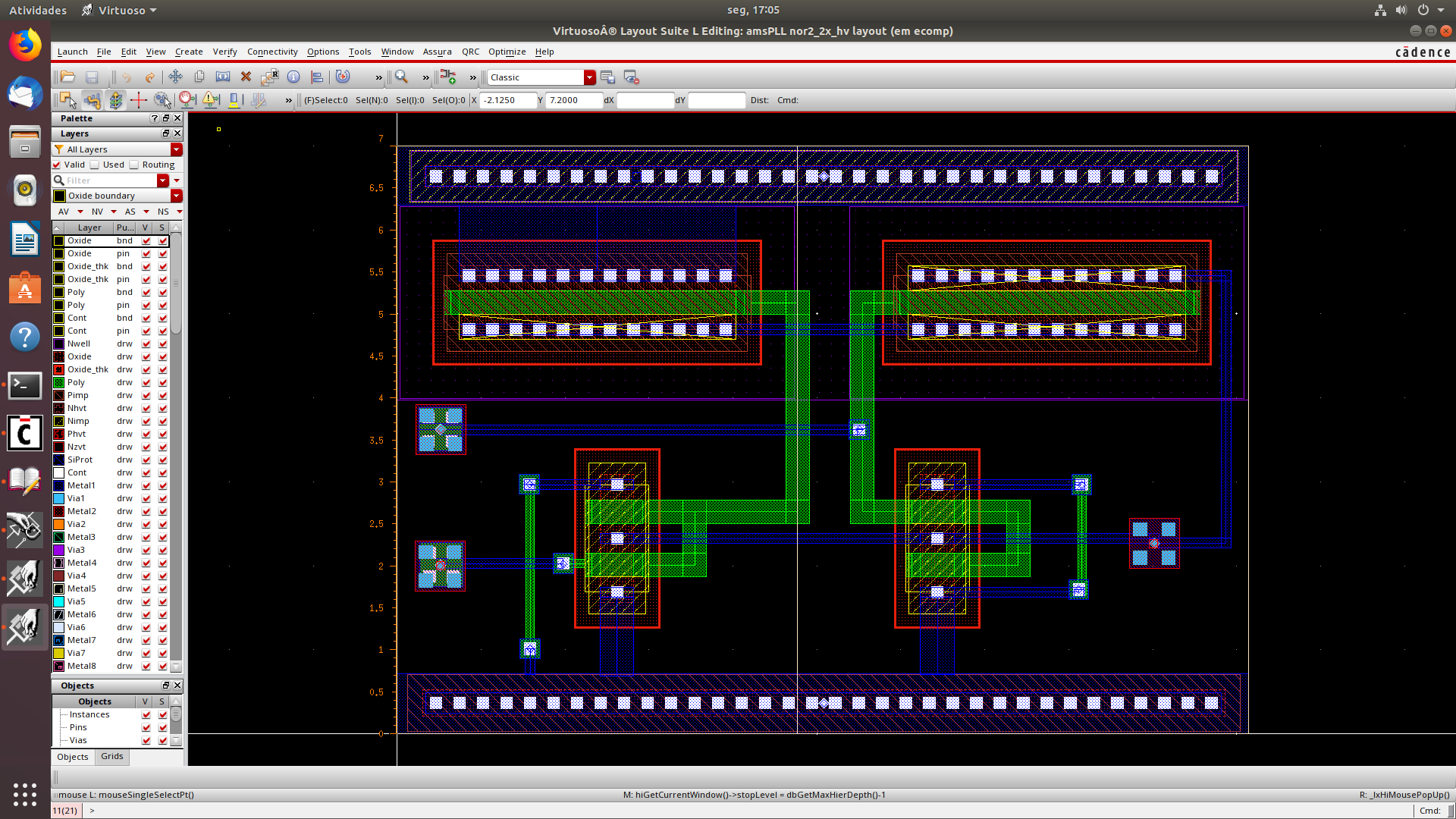
Com toda a parte elétrica funcionando corretamente, o desenvolvimento dos layouts foi dividido em grupos para que fosse concluído no período de tempo estipulado. Dessa forma, portas mais simples como **OR**, **NOR**, **XOR** (apesar de não ser utilizada de fato) e **NOT** foram desenvolvidas por um integrante, enquanto que o **Latch** e os **Buffers** foram desenvolvidos pelos dois integrantes restantes. As portas **NOT** e **Buffer** possuem mais de um tipo de modelo, utilizando transistores diferentes. Abaixo, estão representados os layouts de todas as portas mencionadas. O processo de design baseou-se em padronizar as medidas dos contatos feitos em metal 1 na parte superior e inferior do componente, com 0,7 micras de altura. Ademais, limitou-se a altura do layout total em 7 micras, para que todos os componentes tivessem a mesma dimensão e utilizou-se, na horizontal, predominantemente metal 1, enquanto na vertical utilizou-se polissílico.



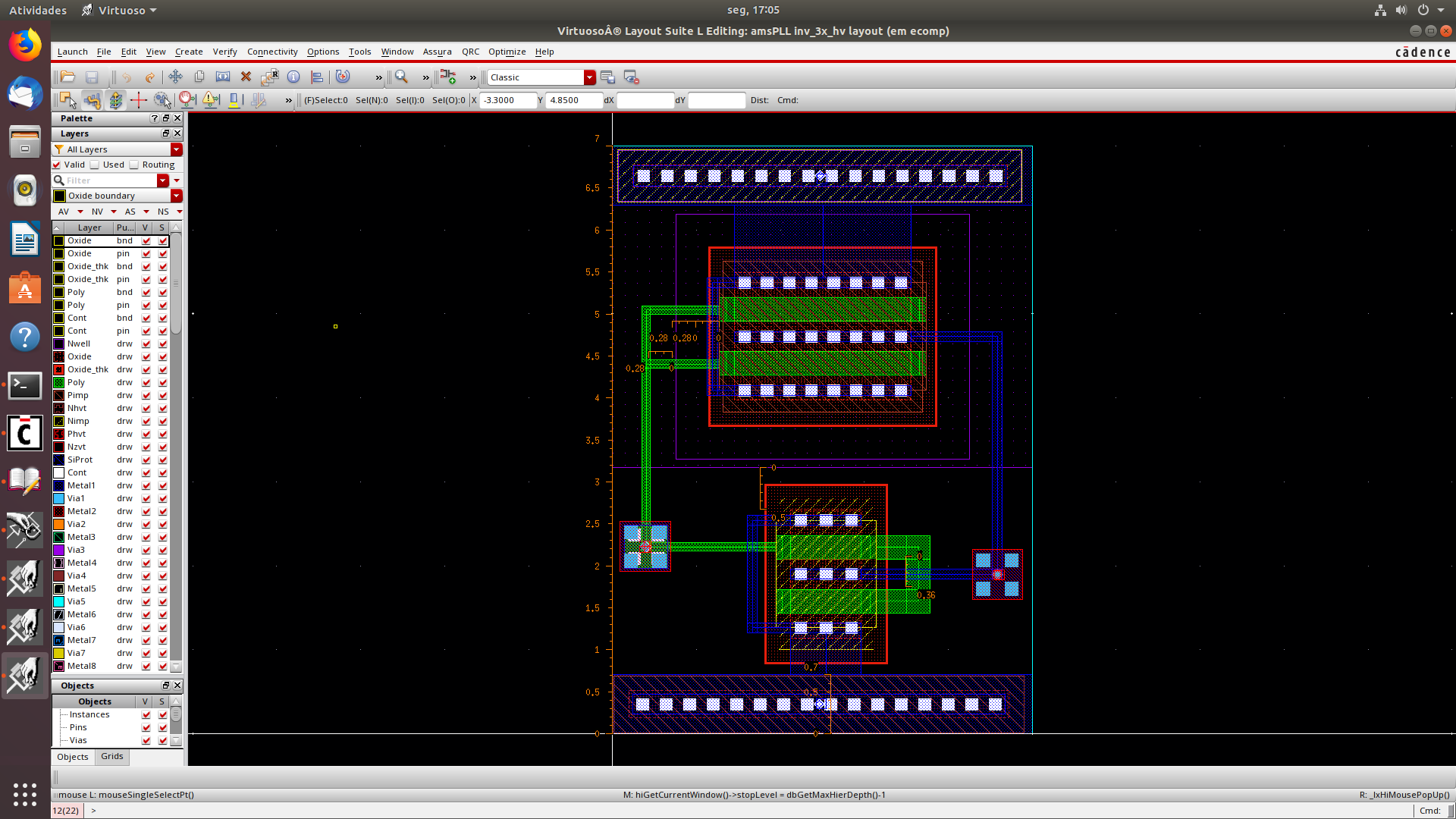
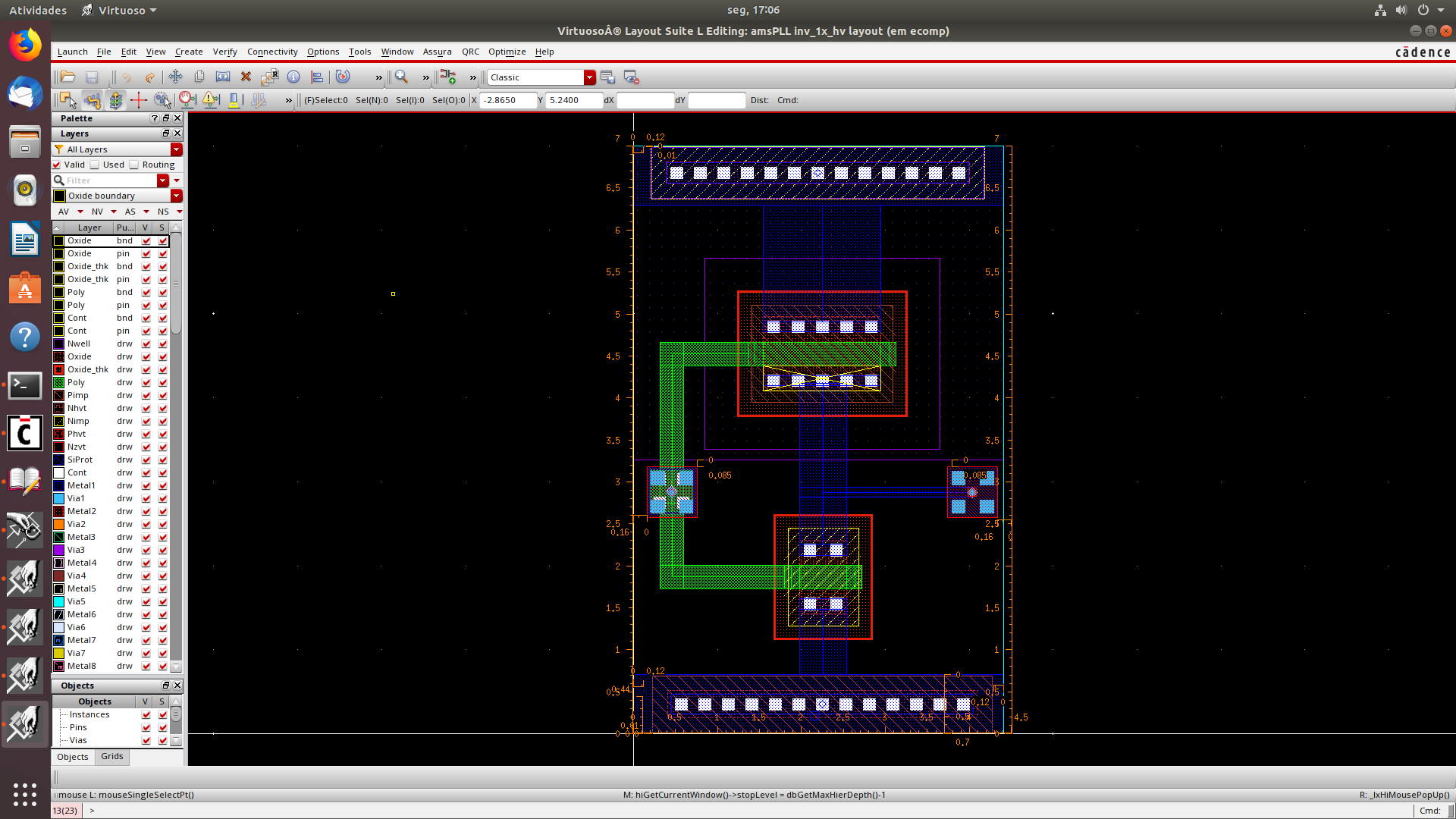
**Figura 4:** Layout da porta XOR.



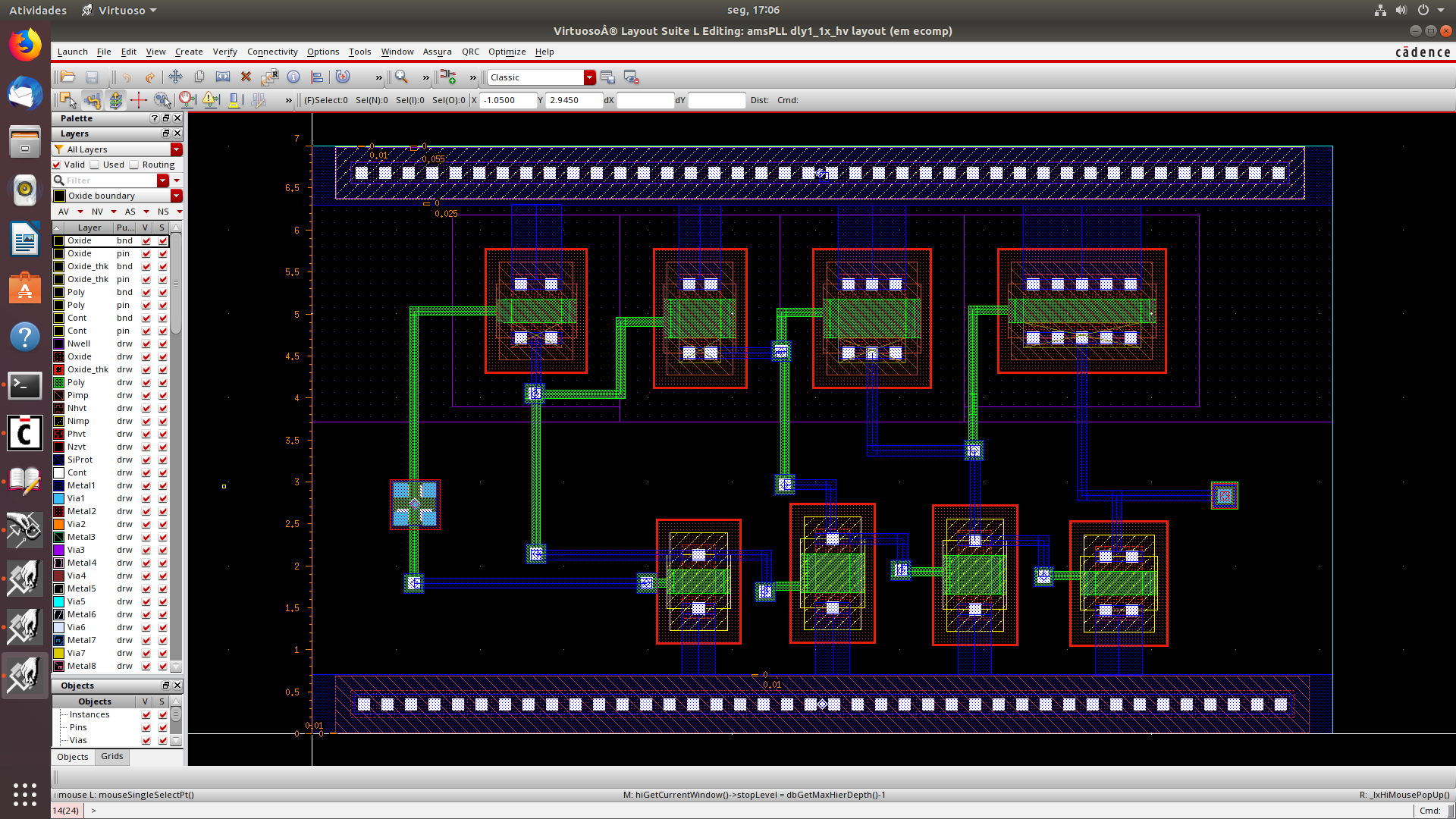
**Figura 5:** Layout da porta OR.



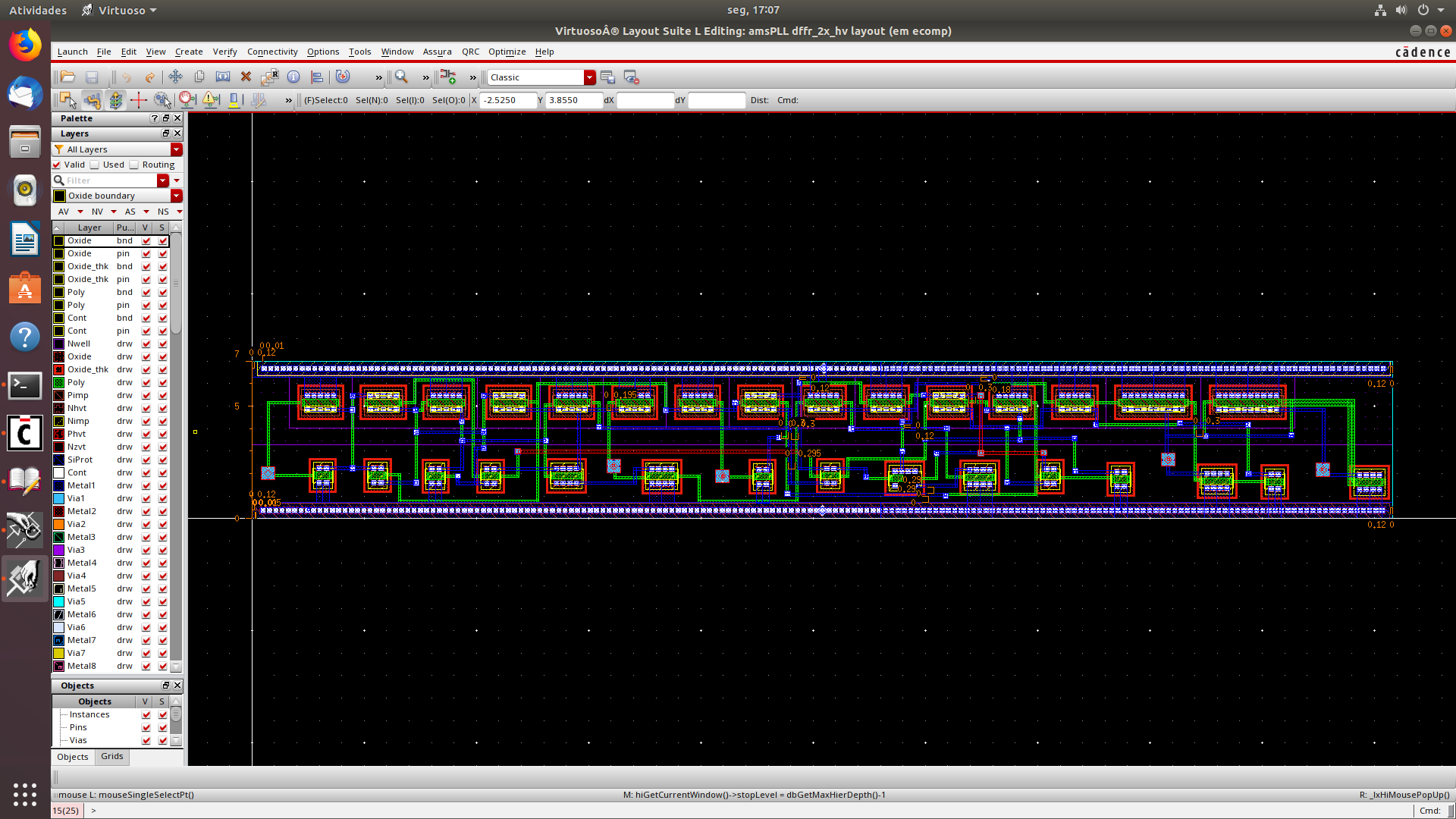
**Figura 6:** Layout da porta NOR.



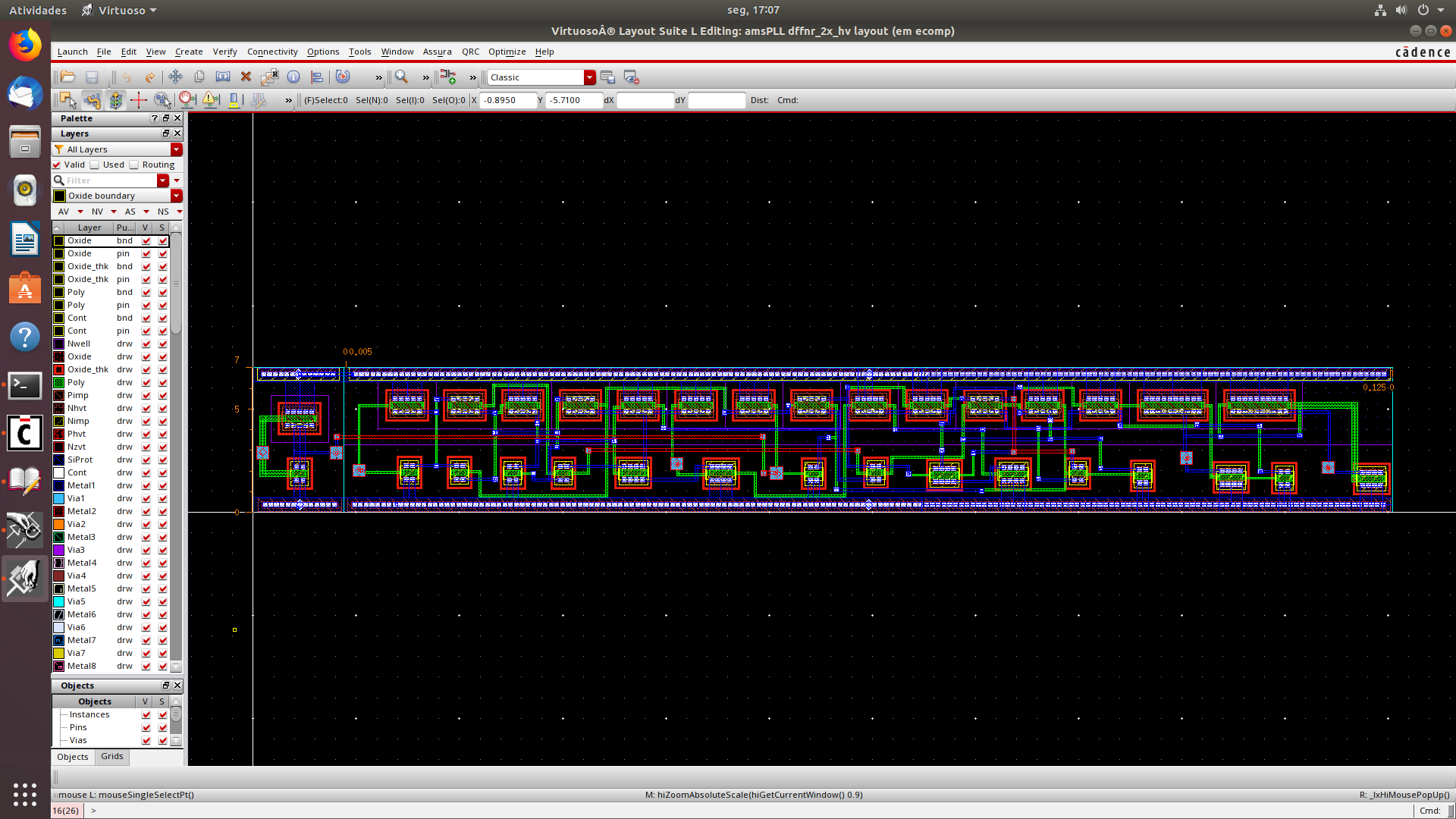
**Figura 7:** Layout dos Inversores (NOT) tipo 1 e 2.



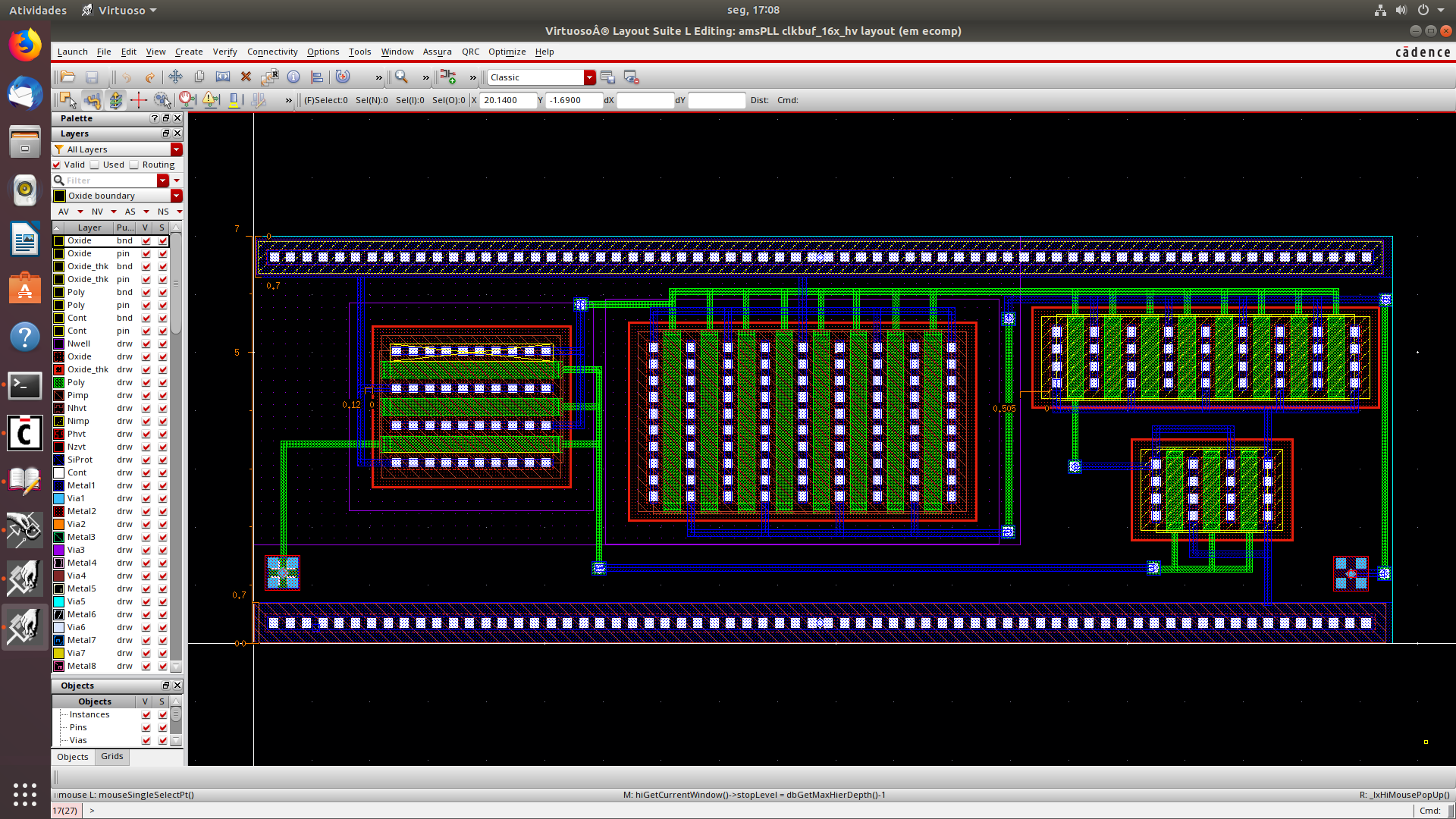
**Figura 8:** Layout do Buffer tipo 1.



**Figura 9:** Layout do Latch.



**Figura 10:** Layout do Latch + Inversor.

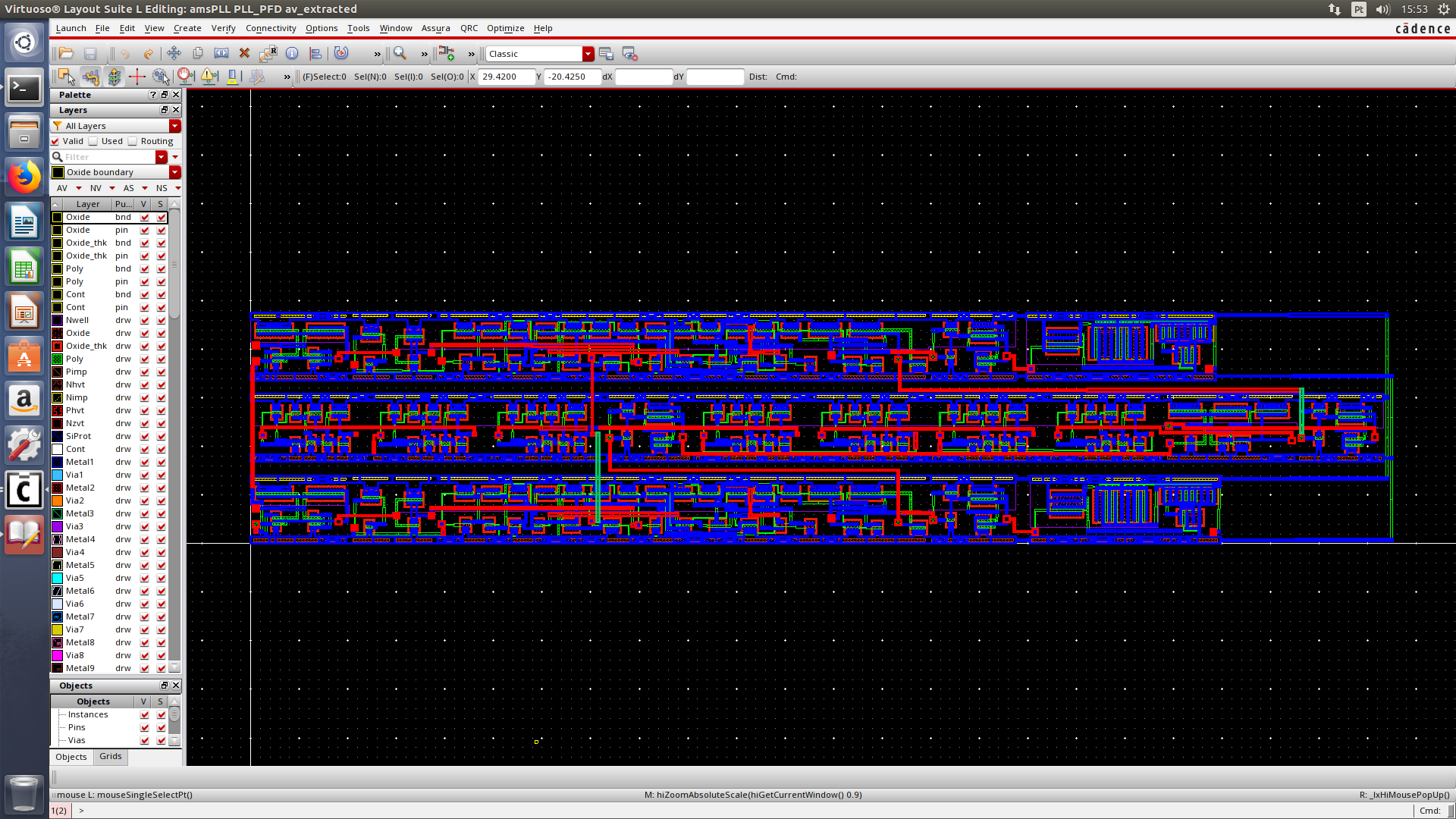


**Figura 11:** Layout do Buffer tipo 2.



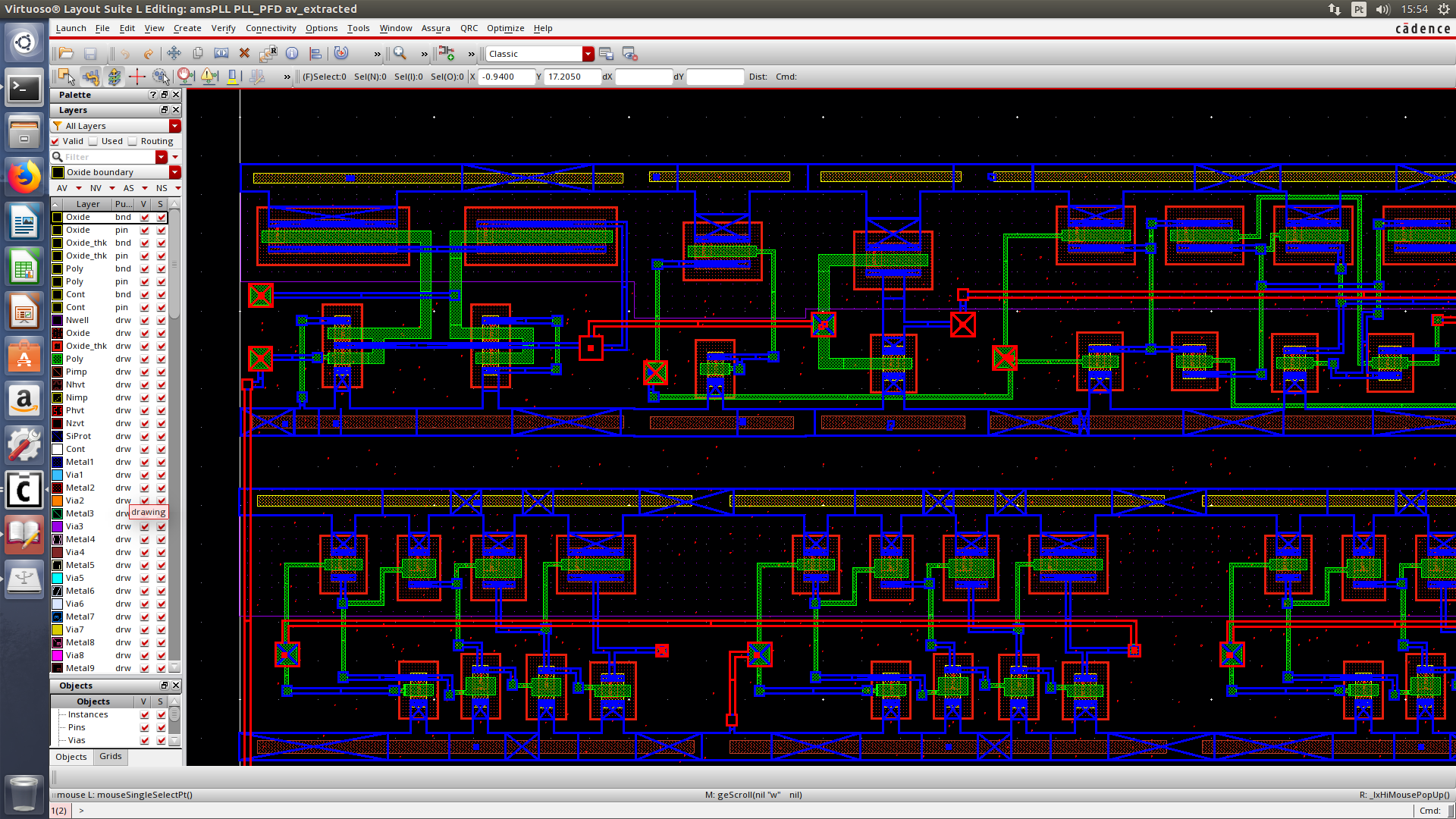
**Figura 12:** Layout final do detector de fase completo.

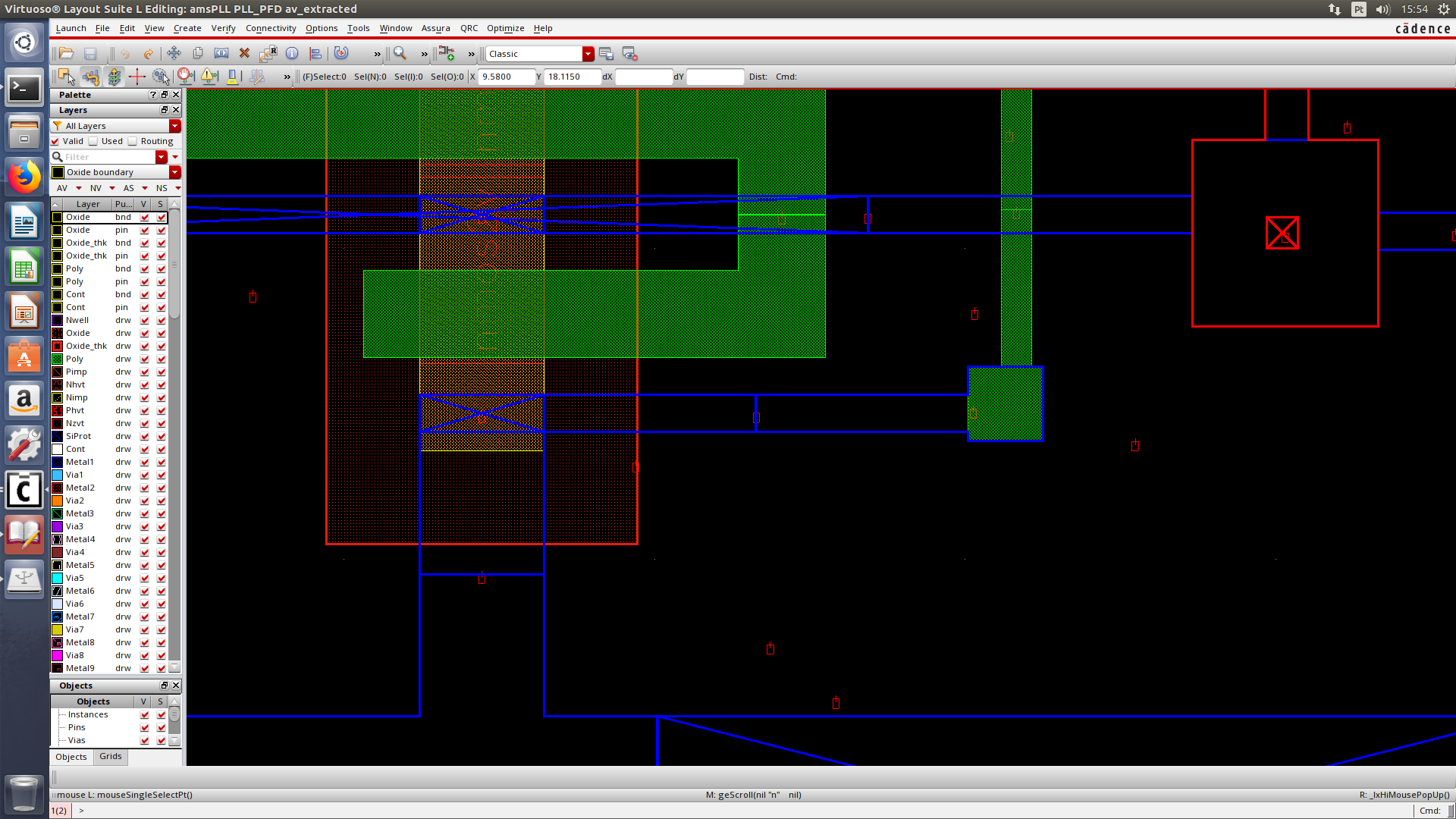
Feito isso, aplicou-se os testes DRC e LVS, para que fosse possível corrigir erros detectados pela ferramenta. Após ajustar as falhas detectadas, foi necessário fazer uma simulação diferente, que consistia em extrair um novo layout final a partir do layout da **Figura 12**. Esta nova versão considerará capacitâncias e resistências parasitas, o que aproxima mais o circuito desenvolvido com a realidade de implementação do mesmo. A seguir encontram-se imagens do circuito obtido.

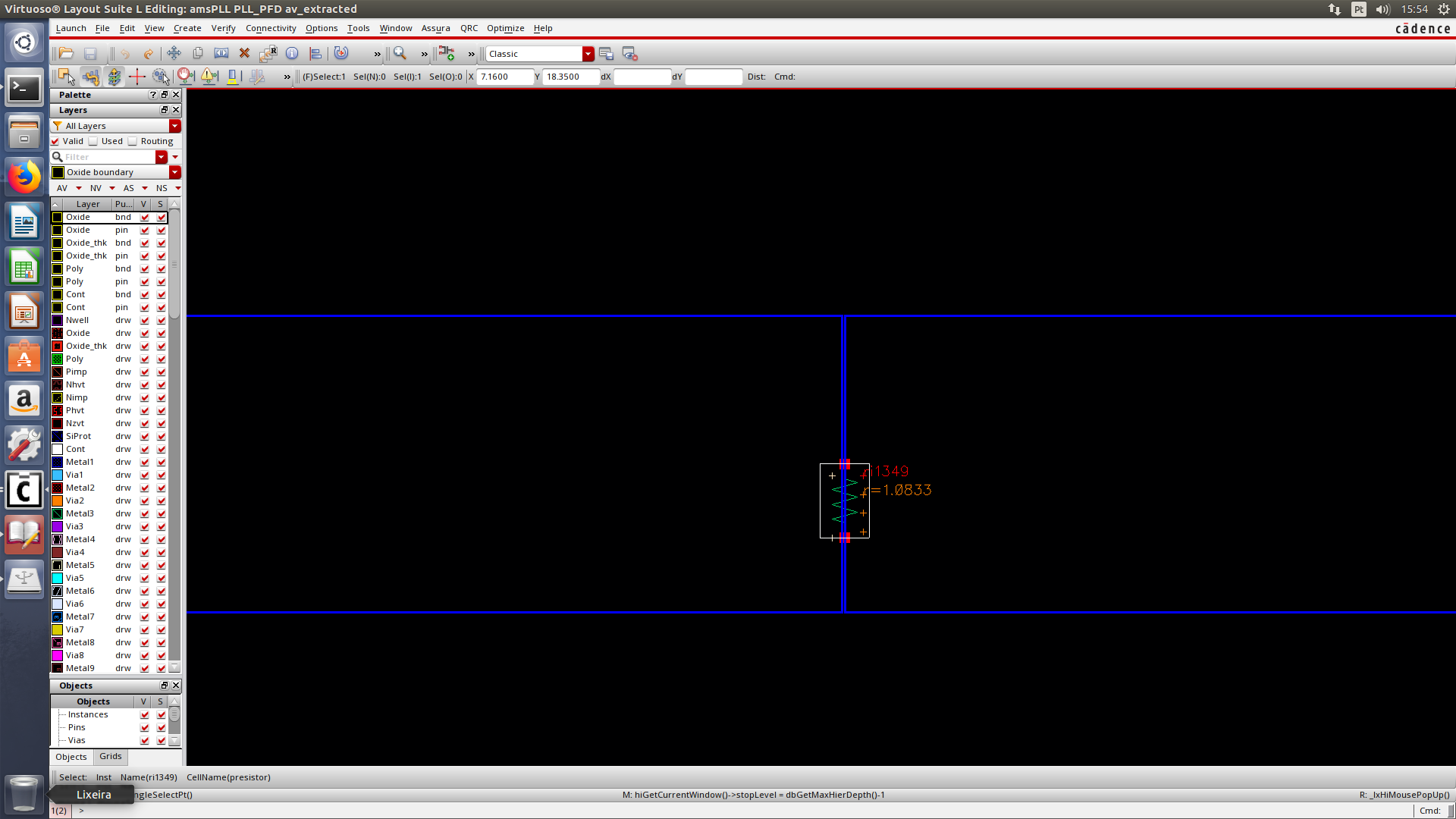


**Figura 13:** Layout final extraído do detector de fase completo.

Abaixo nota-se em mais detalhes o que a ferramenta gerou: pequenos componentes resistivos e capacitivos dispostos por todo o projeto. Isso garante que a funcionalidade do circuito seja vista de uma perspectiva real, com comportamentos mais verídicos.

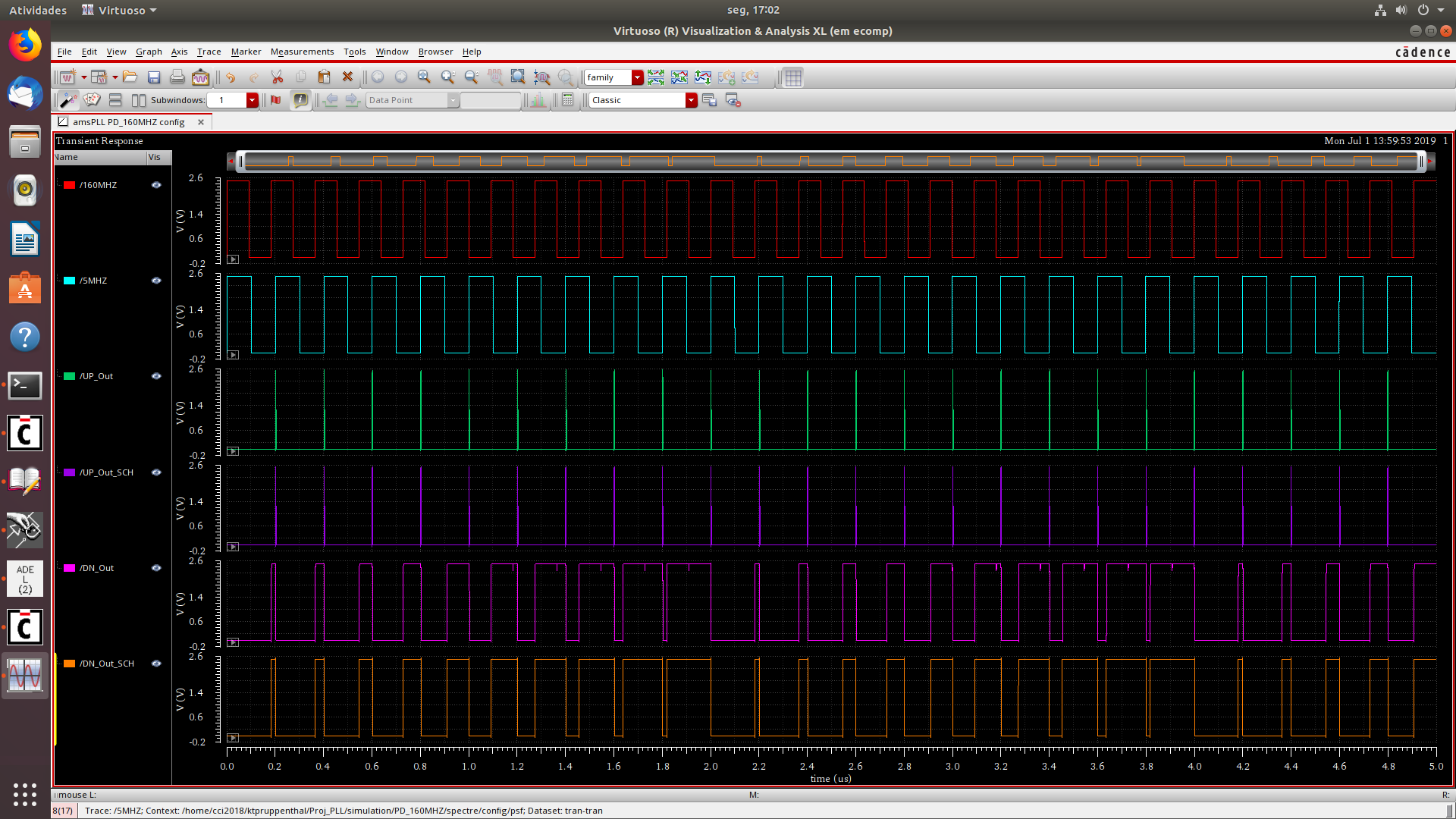




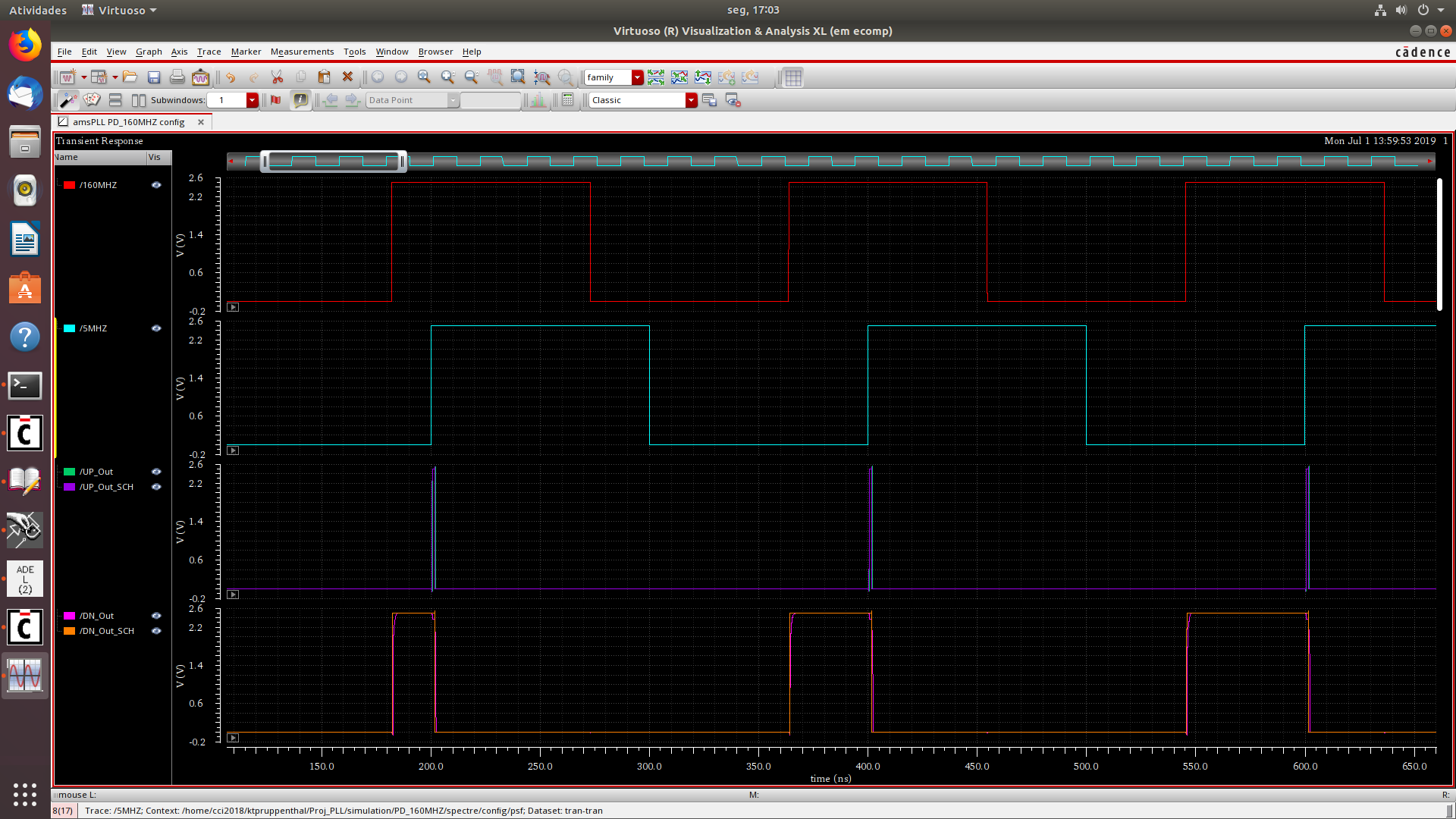


**Figura 14:** Detalhes do Layout final extraído do detector de fase completo.

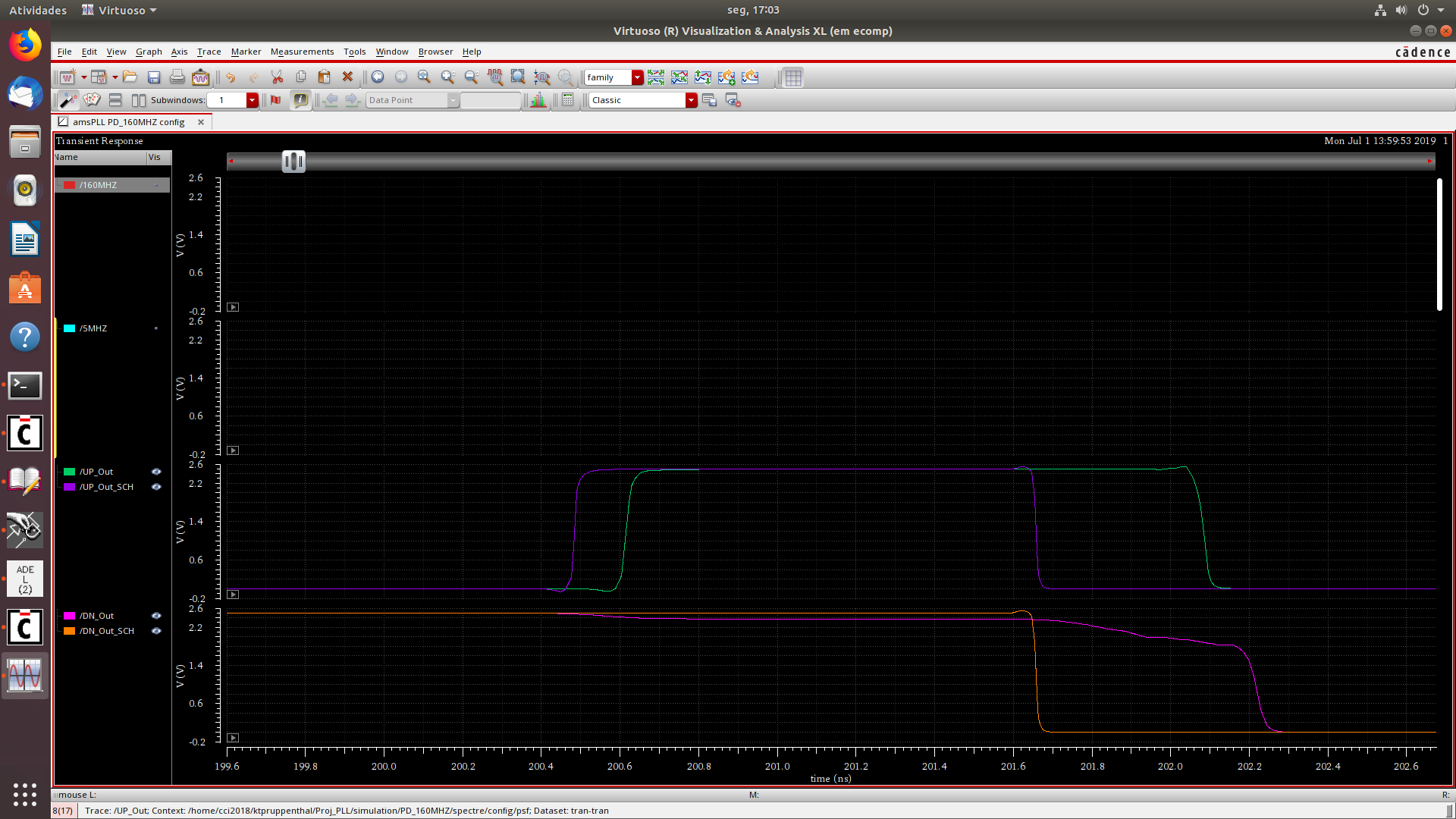
A partir do circuito extraído, fez-se novamente a simulação de sinais de entrada e saída do detector de fase, para que fosse verificada a interferência das capacitâncias e resistências parasitas no circuito. Os resultados podem ser verificados abaixo.



**Figura 15:** Simulação do Layout final extraído do detector de fase completo (1).



**Figura 16:** Simulação do Layout final extraído do detector de fase completo (2).



**Figura 17:** Simulação do Layout final extraído do detector de fase completo (3).

Como pode ser visto na **Figura 15** (e depois de forma mais próxima nas **Figuras 16** e **17**), após o processo de extração do layout, capacitância e resistência parasitas foram adicionadas automaticamente ao circuito, para assim deixá-lo mais próximo do seu real funcionamento. Isso traz os resultados vistos nas simulações: há uma perda da qualidade do sinal, quando comparamos o resultado do circuito original e a saída do circuito pós implementação em layout. Nota-se, principalmente, um atraso da saída do esquemático oriundo do layout, o que pode, mais uma vez, ser explicado pelos elementos parasitas inseridos no circuito após extração do layout pela ferramenta. Estes elementos geram uma latência maior na propagação da tensão, o que pode ser visto claramente na **Figura 17**.

**4. Conclusão**

Como foi verificado, a funcionalidade do bloco detector de fase baseia-se na verificação das duas frequências de entrada: a de referência (5MHz) e a de feedback, que vem do VCO. Com a simulação paramétrica, pudemos observar que o circuito opera exatamente como o esperado, ou seja, quando a frequência de feedback era menor do que 5 MHz, o bloco respondia dando mais pulsos de UP, caso contrário, eram gerados mais pulsos de DN. Quando a frequência de 5MHz era atingida, os pulsos eram iguais.

Desse modo, entendemos que o bloco detector de fase é extremamente importante para o bom funcionamento do PLL, uma vez que a constante comparação para a verificação de quando a frequência desejada é atingida, é o que garante a precisão e confiabilidade de um circuito desse tipo. Sua operabilidade está diretamente ligada com o resultado deste bloco, sendo que a interconexão e dependência de todos os estágios do PLL necessitam de resultados precisos em todos os blocos do mesmo.

Ademais, foi possível notar que após o processo de extração do layout capacitância e resistências parasitas foram adicionadas automaticamente ao circuito, para assim deixá-lo mais verossímil. Resistências e capacitâncias parasitas são elementos indesejados em qualquer circuito elétrico/eletrônico, mas que na sua forma real, todos possuem.

A capacitância parasita se origina da proximidade de dois condutores elétricos, carregados e com uma diferença de potencial entre eles, podendo-se dizer que esta situação cria um capacitor. Já as resistências parasitas podem ser encontradas nas “emendas” dos metais, nas ligações entre metais.

Capacitâncias parasitas tem grande influência negativa sobre a relação sinal-ruído do circuito, o que prejudica bastante seu funcionamento e cria a necessidade de correção deste problema, aplicando meios de minimizar a capacitância parasita, como, por exemplo, manter os terminais dos componentes o mais curto possível, diminuindo assim o número de ligações com “emendas” minimizando também a resistência parasita.

**Referências**

[1] “*Como funciona o PLL*”. Disponível em :<<https://www.newtoncbraga.com.br/index.php/como-funciona/6528-como-funciona-o-pll-art1063>>. Acesso em: 24 maio 2019.

[2] Material disponibilizado pelo professor.

1. Acadêmico(a) do curso de Engenharia de Computação da Universidade Federal de Santa Maria - UFSM, matrícula: 201322332, e-mail: giuliano.benedeti@ecomp.ufsm.br [↑](#footnote-ref-0)
2. Acadêmico(a) do curso de Engenharia de Computação da Universidade Federal de Santa Maria - UFSM, matrícula: 201520603, e-mail: kelitauana@gmail.com [↑](#footnote-ref-1)
3. Acadêmico (a) do curso de Engenharia de Computação da Universidade Federal De Santa Maria- UFSM, matrícula: 201520835, e-mail: [victor.bento@ecomp.ufsm.br](mailto:victor.bento@ecomp.ufsm.br) [↑](#footnote-ref-2)