

Relatório da Tarefa 2: Microwind 3.

Aluno: Mairon Schneider Cardoso.

Data: 23/06/2021

Número de matrícula: 17102515.

1 Introdução

A tecnologia *MOSFET* possibilita a construção eficiente dos circuitos digitais. Os circuitos digitais são circuitos que tem apenas dois níveis de tensão, sendo zero e um, diferentemente dos analógicos, onde qualquer tensão pode ser admitida. Os circuitos digitais são extremamente importante justamente por realizarem as operações lógicas através da região de operação denominada de região de corte, funcionando como uma chave. Nesse contexto, a construção de portas lógicas a partir do leiaute é um importante passo como engenheiro uma vez que, é possível quantificar os custos para produção de determinada porta lógica tendo em vista o espaço ocupado, as margens de consumo, assim como a dissipação de tensão em todos os terminais, possibilitando com que o projetista consiga criar uma porta lógica ainda mais eficiente observando os *trade-offs*.

2 Metodologia

O estudo deste trabalho será fundamentado em ideias e pressupostos teóricos que apresentam significativa importância na construção de dispositivos eletrônicos a partir do design das portas lógicas, isso só é possível por meio de um *software* que permite a construção coerente das portas lógicas. Além disso, através do leiaute obtido, é possível analisar a resposta obtida através do uso da ferramenta *Spice* (assim como o *Microwind*) para determinar nuncias dos circuitos requisitados no escopo do trabalho.

3 Resultados e Discussões

A construção da primeira porta lógica F_1 se dá primeiramente pela análise da disposição dos transistores, tal observação é possível através do *Switch-craft*, isso também nos permite garantir um caminho ideal para conceber uma porta lógica que consuma o menor espaço possível (figura 1). A metodologia para desenvolvimento do leiaute é baseada na concepção das redes de transistores *pull-up* e *pull-down*, através da ferramenta *Microwind*, é possível construir o leiaute da porta lógica F_1 (figura 2), que tem a equação descrita na equação 1. A construção desta porta lógica utiliza-se de apenas metal 1 na concepção, o que garante uma altura de 11 nas dimensões de projeto, além disso, como observada a técnica *logic effort* nos diz que a dimensão do poço da rede *pull-up* será igual a $100 \cdot 1,5 = 0.10U$ e a dimensão da rede *pull-down* terá um poço igual a $4 \cdot 100 \cdot 1 = 0.40U$.

$$F_1 = (a * d * b * c) \quad (1)$$

A validação do porta lógica F_1 se dá pelo meio de dois processos (aqui é empregado dois métodos para conferir a correta concepção), simulando através do próprio *software* usado para conceber a porta lógica (figura 3), removendo as informações de *delay* através do próprio *software*. Ao obter as informações quanto a resposta da porta lógica com diferentes entradas, conseguimos cumprir a correta resposta que deveríamos garantir, entretanto, para verificar a veracidade, foi utilizado a simulação via *spice* e, através da simulação, conseguimos observar o comportamento semelhante das respostas (figura 4).

A construção da primeira porta lógica F_2 se dá primeiramente pela análise da disposição dos transistores, tal observação é possível através do *Switch-craft*, isso também nos permite garantir um

caminho ideal para conceber uma porta lógica que consuma o menor espaço possível, nesse caso, indentificando o caminho de Euler (figura 5). A metodologia para desenvolvimento do leiaute é baseada na concepção das redes de transistores *pull-up* e *pull-down*, através da ferramenta *Microwind*, é possível construir o leiaute da porta lógica F_2 (figura 6), que tem a equação descrita na equação 2. A construção desta porta lógica utiliza-se de metal 1 e metal 2 na concepção, o que garante uma altura de 16 nas dimensões de projeto, além disso, como observada a técnica *logic effort* nos diz que a dimensão do poço da rede *pull-up* será igual a $100 \cdot 3 \cdot 1,5 = 0,45U$ e a dimensão da rede *pull-down* terá um poço igual a $100 \cdot 4 \cdot 1 = 0,40U$. A construção das portas lógicas negadas é feita ligando um metal 2 entre os conectores passando por um inversor, e saindo para os seguintes transistores (que seguem o caminho de Euler).

$$F_2 = (!a*!b*c) + (!a * b * c) + (a*!b * c) + (a * b*!c) \quad (2)$$

A validação do porta lógica F_2 , simulando através do próprio *software* usado para conceber a porta lógica (figura 7), removendo as informações de *delay* através do próprio *software*. Ao obter as informações quanto a resposta da porta lógica com diferentes entradas, conseguimos cumprir a correta resposta que deveríamos garantir, entretanto, para verificar a veracidade, foi utilizado a simulação via *spice* e, através da simulação, conseguimos observar o comportamento semelhante das respostas (figura 8).

A construção da primeira porta lógica F_3 se dá primeiramente pela análise da disposição dos transistores, tal observação é possível através do *Switch-craft*, isso também nos permite garantir um caminho ideal para conceber uma porta lógica que consuma o menor espaço possível (figura 9). A metodologia para desenvolvimento do leiaute é baseada na concepção das redes de transistores *pull-up* e *pull-down*, através da ferramenta *Microwind*, é possível construir o leiaute da porta lógica F_3 (figura 10), que tem a equação descrita na equação 3. A construção desta porta lógica utiliza-se de apenas metal 1 na concepção, o que garante uma altura de 9 nas dimensões de projeto, além disso, como observada a técnica *logic effort* nos diz que a dimensão do poço da rede *pull-up* será igual a $100 \cdot 1,5 \cdot 2 = 0,28U$ e a dimensão da rede *pull-down* terá um poço igual a $1 \cdot 100 \cdot 1 = 0,20U$.

$$F_3 = (a + b) \quad (3)$$

A validação do porta lógica F_3 , simulando através do próprio *software* usado para conceber a porta lógica (figura 11), removendo as informações de *delay* através do próprio *software*. Ao obter as informações quanto a resposta da porta lógica com diferentes entradas, conseguimos cumprir a correta resposta que deveríamos garantir, entretanto, para verificar a veracidade, foi utilizado a simulação via *spice* e, através da simulação, conseguimos observar o comportamento semelhante das respostas (figura 12).

3.1 Concepção do circuito completo.

A construção da porta lógica que implementa a função descrita na equação 4, aproveita-se do método de inversão da porta lógica F_1 verticalmente, o que diminuiu a altura do circuito completo em 1, justamente por compartilhar o mesmo terra entre os circuitos (figura 13). É possível conferir a validação do circuito completo analisando a simulação (figura 14) e através da simulação via *spice* é possível garantir o mesmo comportamento do circuito (figura 15), onde, através das tabelas verdades do circuito, é possível verificar que o circuito implementado garante a operação requerida com uma altura total de 26.

$$F_3 = F_1 + F_2 \quad (4)$$

4 Anexos

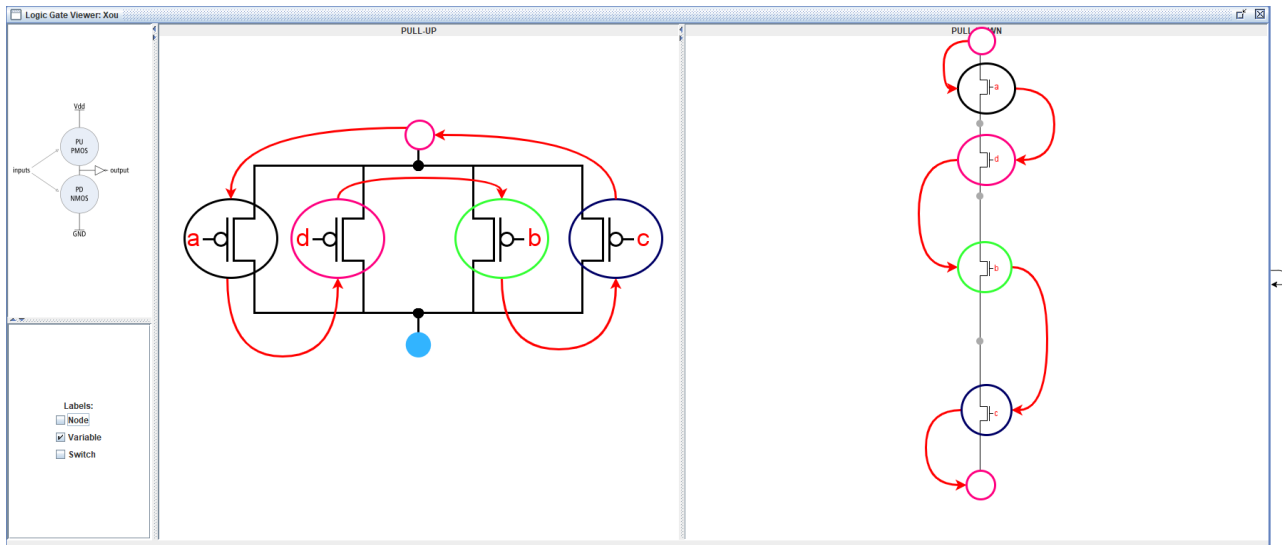


Figura 1: Caminhos das ligações da porta lógica F_1 .

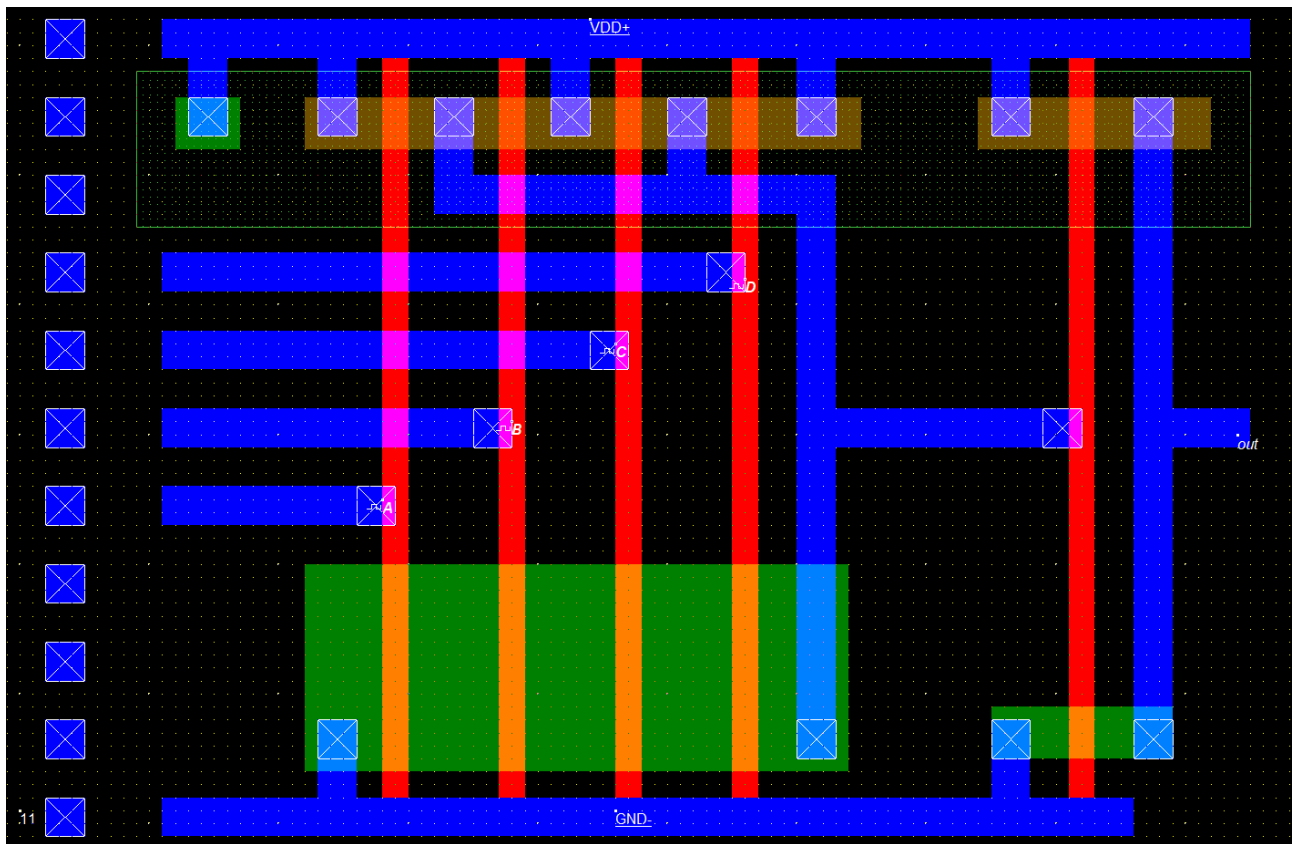


Figura 2: Porta lógica F_1 .

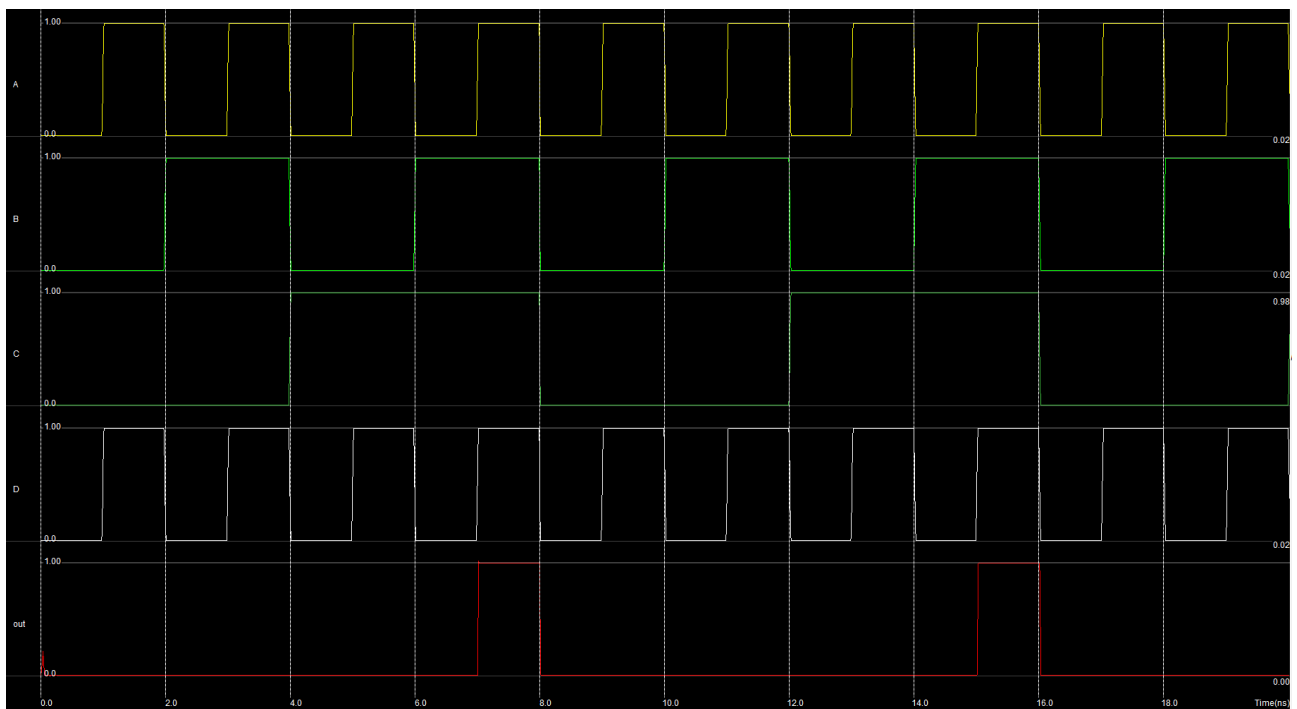


Figura 3: Simulação da porta F_1 através do *Microwind*.

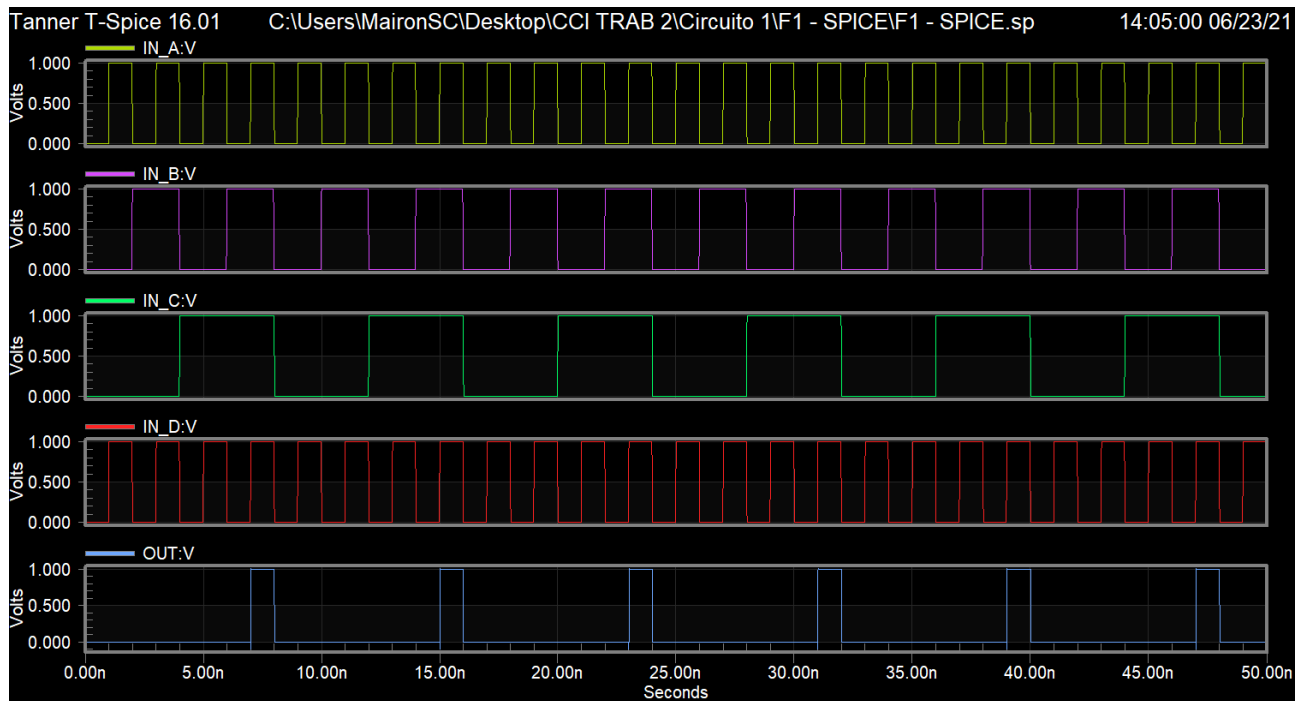


Figura 4: Simulação da porta F_1 através do *Spice* (*Tanner EDA Tools*).

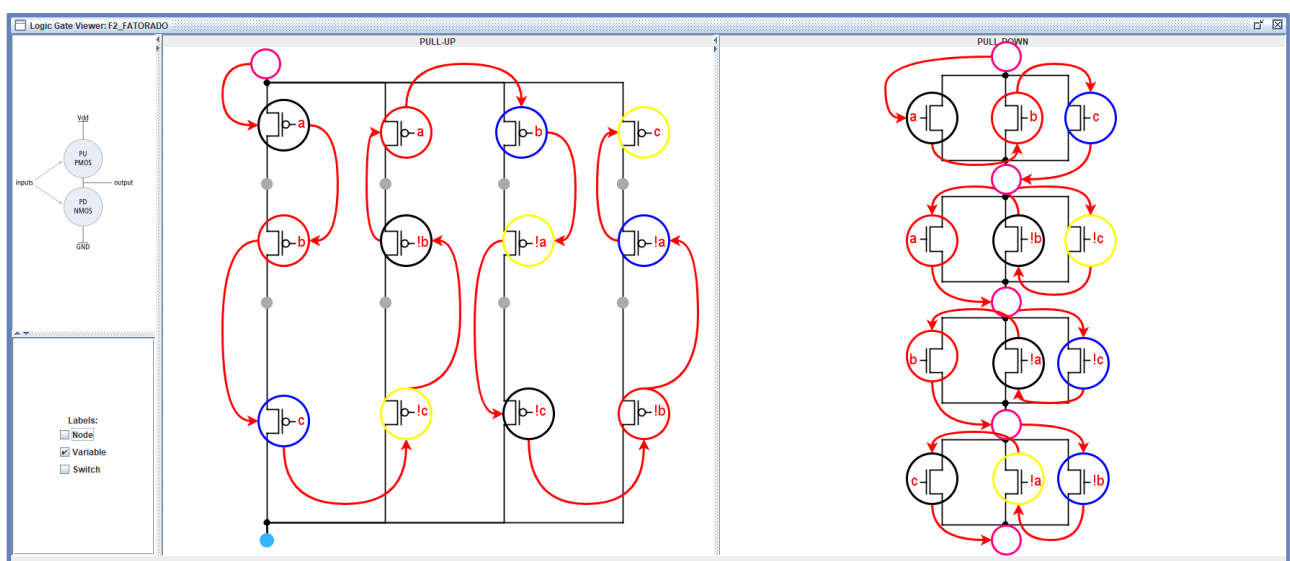


Figura 5: Caminhos das ligações da porta lógica F_2 .

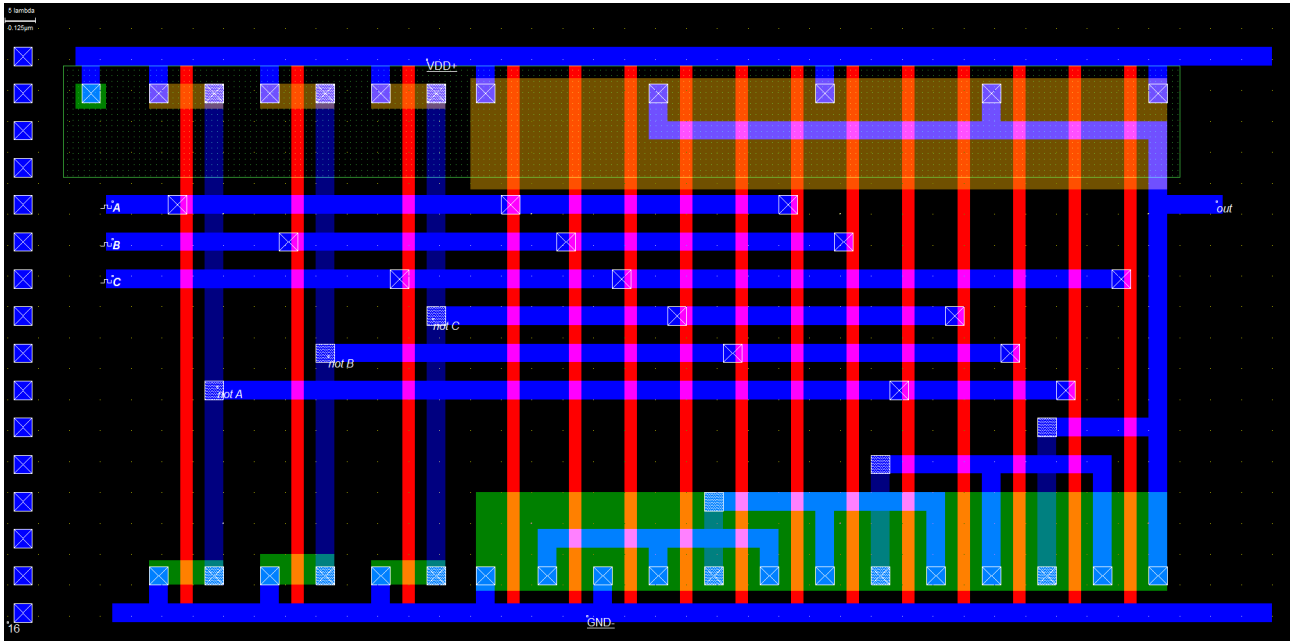


Figura 6: Porta lógica F_2 .

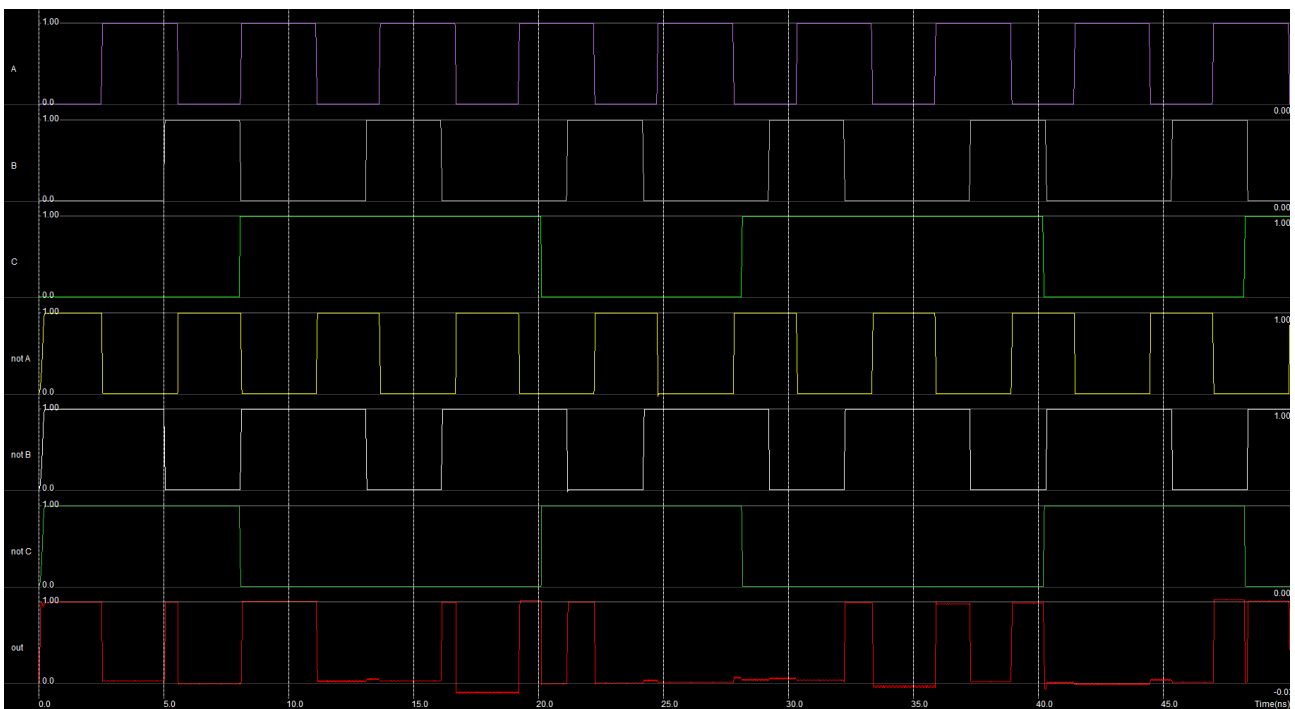


Figura 7: Simulação da porta F_2 através do *Microwind*.

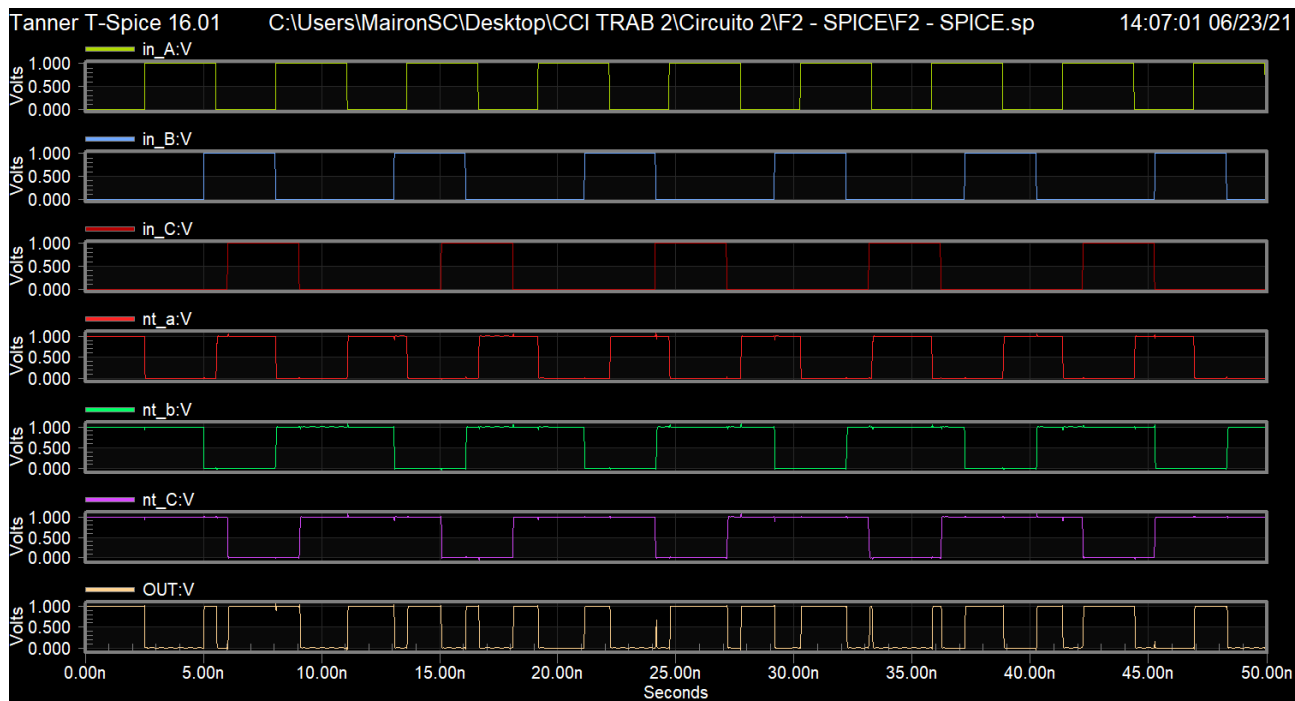


Figura 8: Simulação da porta F_2 através do *Spice* (*Tanner EDA Tools*).

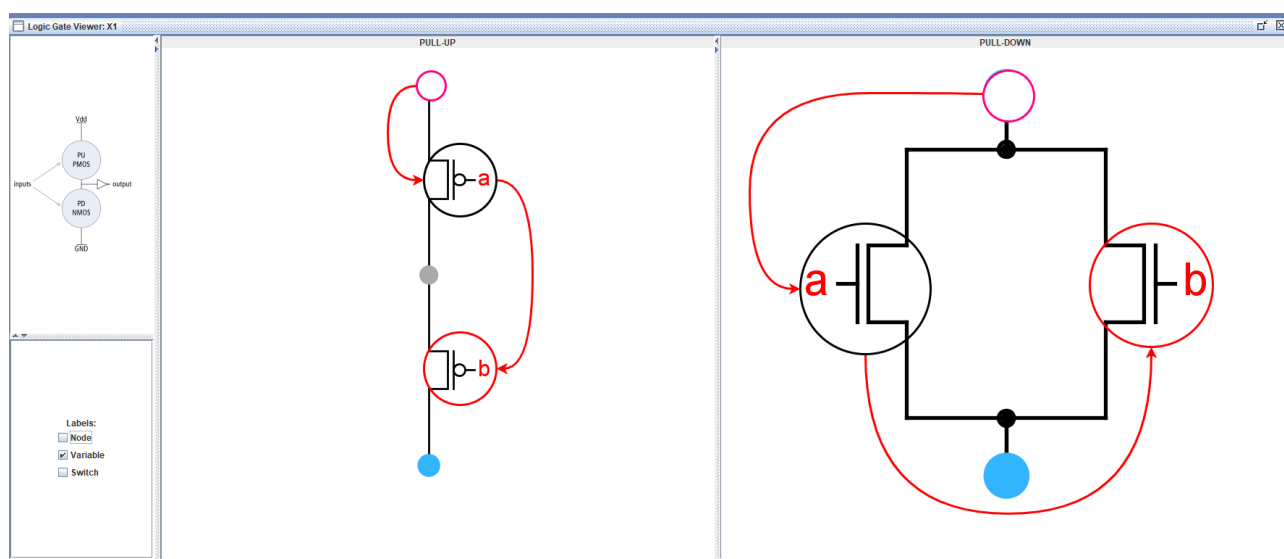


Figura 9: Caminhos das ligações da porta lógica F_3 .

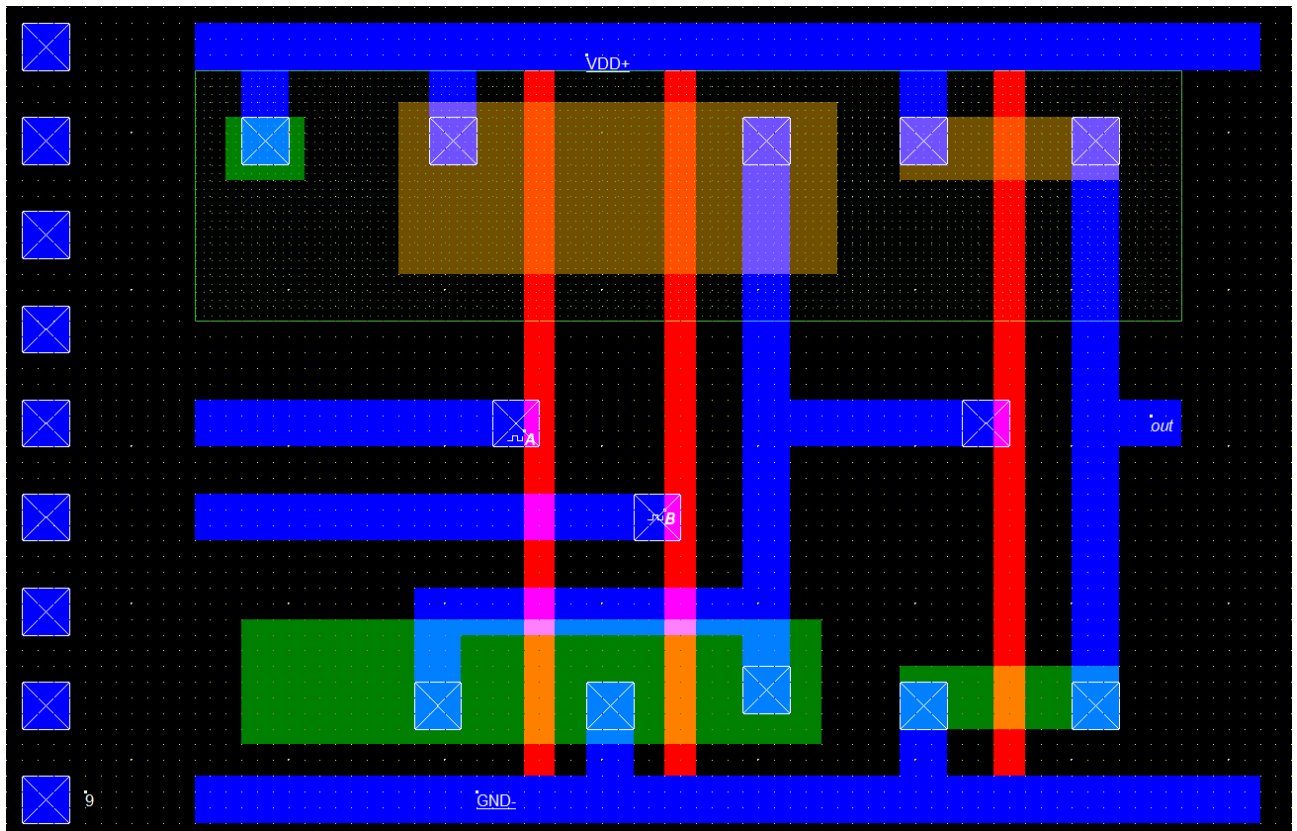


Figura 10: Porta lógica F_3 .

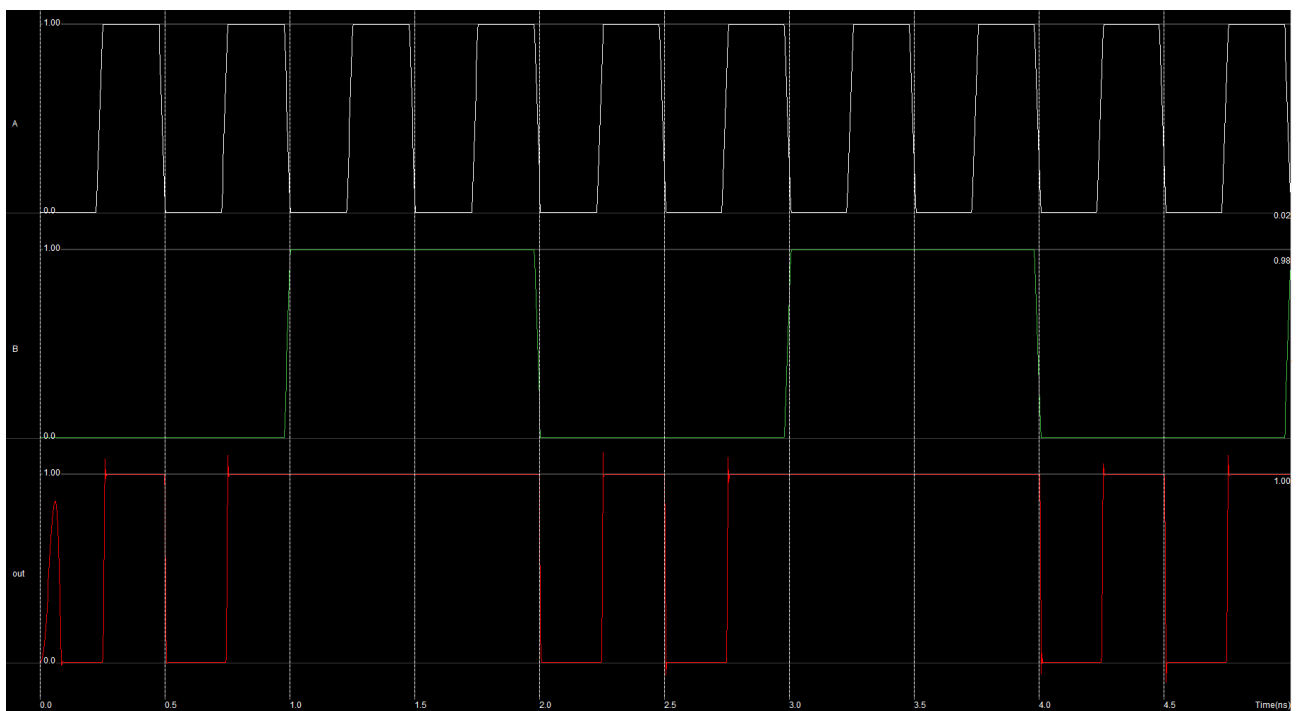


Figura 11: Simulação da porta F_3 através do *Microwind*.

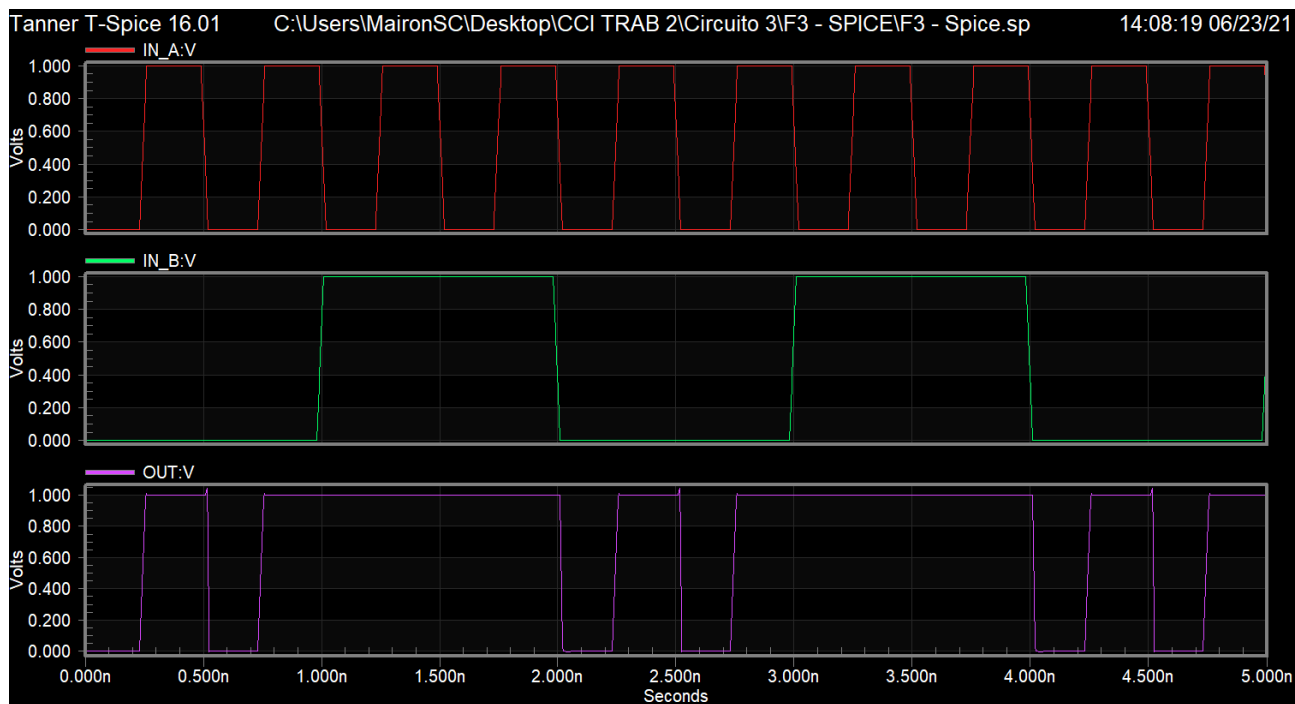


Figura 12: Simulação da porta F_3 através do *Spice* (*Tanner EDA Tools*).

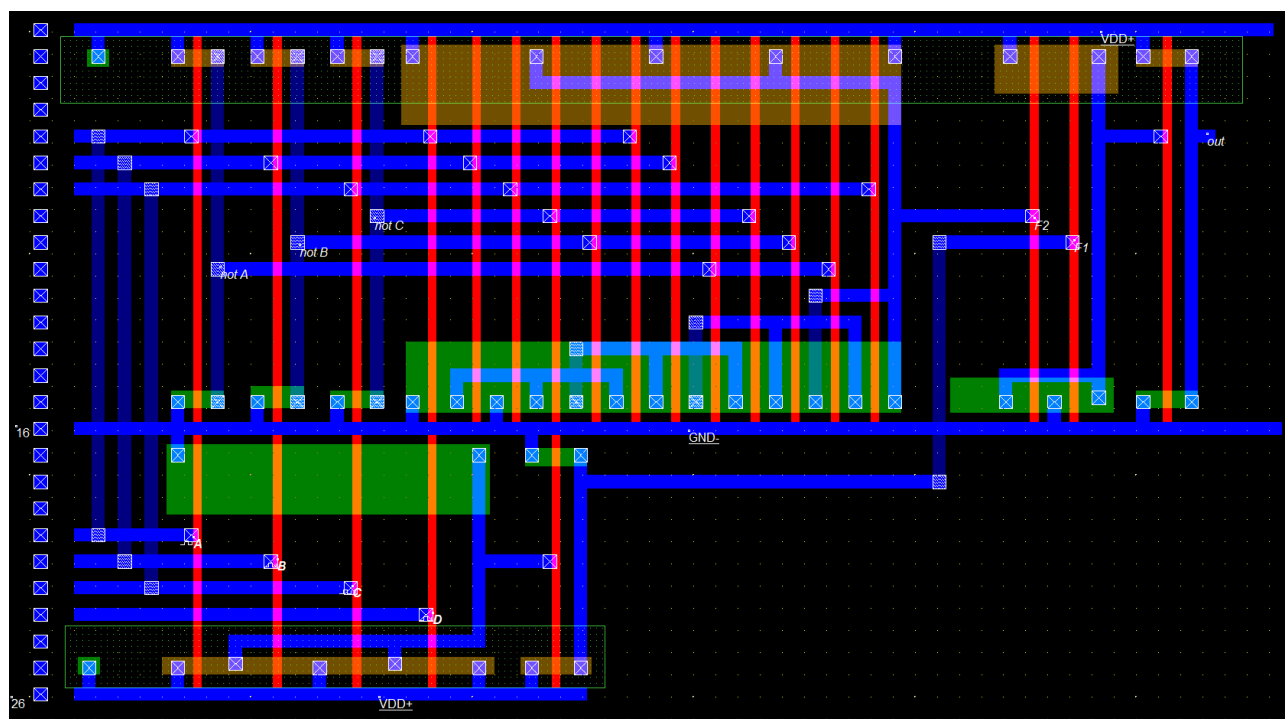


Figura 13: Porta lógica F_3 , recebendo entrada as outras duas portas lógicas F_1 e F_2 .

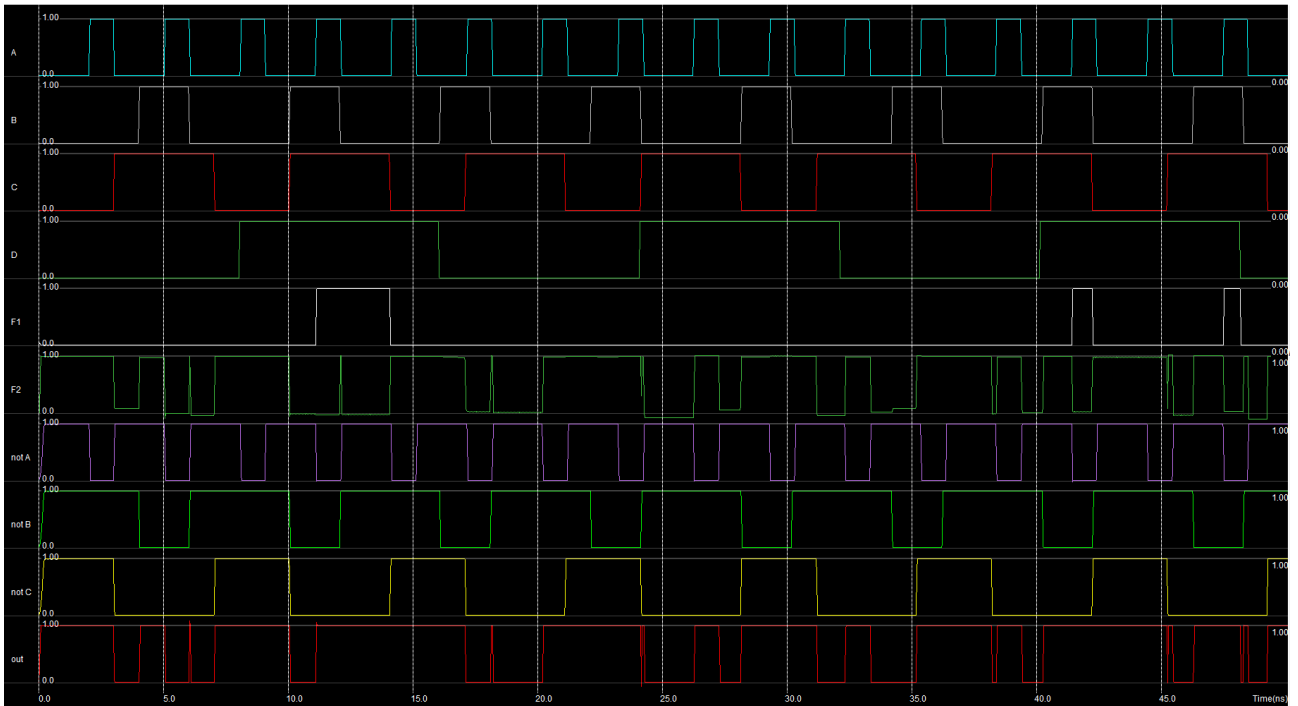


Figura 14: Simulação da porta F_3 através do *Microwind*, recebendo entrada as outras duas portas lógicas F_1 e F_2 .

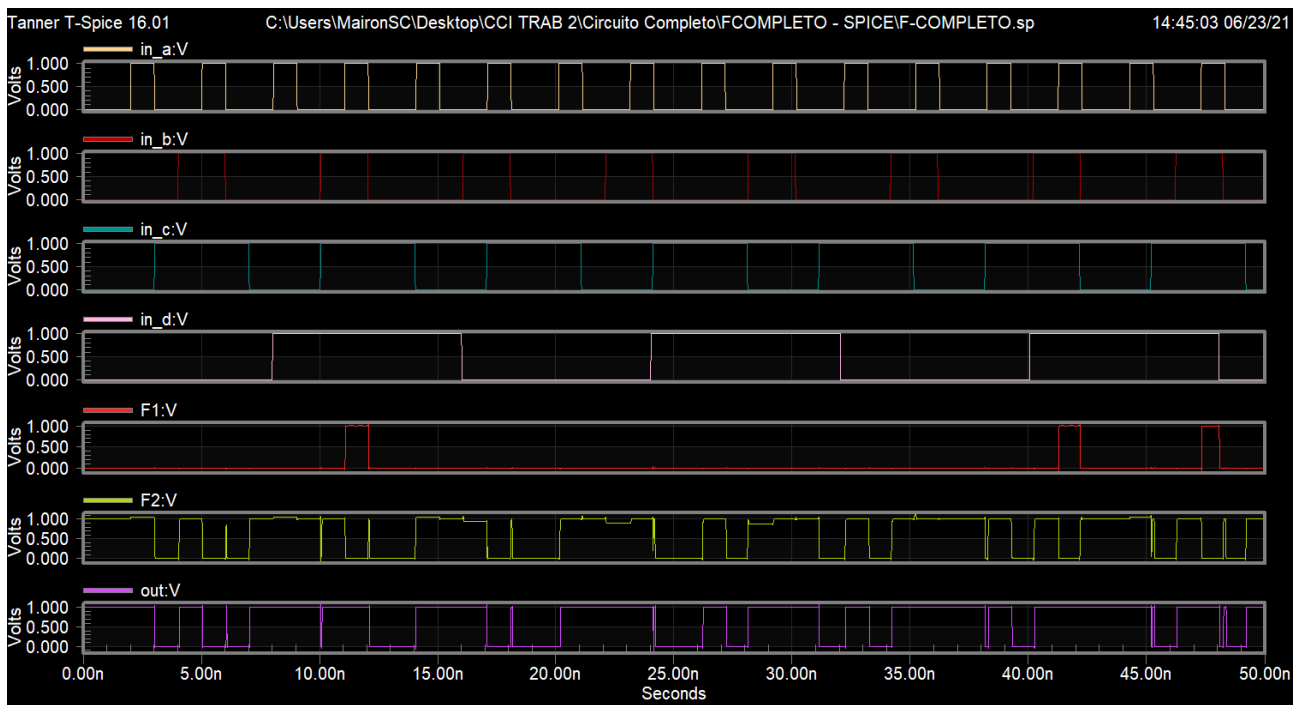


Figura 15: Simulação da porta F_3 através do *Spice (Tanner EDA Tools)*, recebendo entrada as outras duas portas lógicas F_1 e F_2 .