

**INSTITUTO FEDERAL DE MINAS GERAIS
BACHARELADO EM ENGENHARIA ELÉTRICA
RENNER RIBEIRO BRANDÃO**

**PROJETO DE *LAYOUT* DE UM CIRCUITO INTEGRADO CMOS CONTADOR
SÍNCRONO DE QUATRO *BITS* UTILIZANDO FERRAMENTAS EDUCACIONAIS**

**FORMIGA – MG
2017**

RENNER RIBEIRO BRANDÃO

**PROJETO DE *LAYOUT* DE UM CIRCUITO INTEGRADO CMOS CONTADOR
SÍNCRONO DE QUATRO *BITS* UTILIZANDO FERRAMENTAS EDUCACIONAIS**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica do Instituto Federal de Minas Gerais, como requisito para obtenção do título de bacharel em Engenharia Elétrica.

Orientador: Rafael Vinícius Tayette da Nobrega

FORMIGA – MG

2017

RENNER RIBEIRO BRANDÃO

**PROJETO DE *LAYOUT* DE UM CIRCUITO INTEGRADO CMOS CONTADOR
SÍNCRONO DE QUATRO *BITS* UTILIZANDO FERRAMENTAS EDUCACIONAIS**

Trabalho de Conclusão de Curso apresentado
ao Curso de Engenharia Elétrica do Instituto
Federal de Minas Gerais como requisito para
obtenção do título de bacharel em Engenharia
Elétrica.

Avaliado em: 21 de junho de 2017.

Nota: _____

BANCA EXAMINADORA

Prof. 01 – MSc. Rafael Vinícius Tayette da Nobrega

Prof. 02 – Dr. Otávio de Souza Martins Gomes

Prof. 03 – Dr. Ulysses Rondina Duarte

Dedico este trabalho a Deus, e a todos que contribuíram direta ou indiretamente em minha formação acadêmica.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela saúde e pela força que tive para superar as dificuldades encontradas.

Aos meus pais, que me deram amor, incentivo e apoio incondicional.

Aos professores, por proporcionarem aprendizado de qualidade durante todo o curso.

Ao meu orientador Rafael Vinícius, por todo o tempo que dedicou à minha orientação no decorrer de minha formação.

Aos amigos e colegas, que fizeram parte desta caminhada e compartilharam os momentos vividos no decorrer do curso, tornando-os mais felizes.

E enfim, a todos que contribuíram direta ou indiretamente para a realização deste trabalho, muito obrigado!

RESUMO

Atualmente observa-se o quanto a utilização dos circuitos integrados é importante para o desenvolvimento de tecnologia, eles não só mantêm o crescimento da indústria eletrônica, como fazem parte do cotidiano do ser humano, estando presentes em praticamente todos os aparelhos eletrônicos utilizados no dia-a-dia. O fato que permitiu o desenvolvimento dos circuitos integrados foi o domínio e a manipulação dos materiais semicondutores, o que levou à criação do transistor, dispositivo chave para a integração de componentes em circuito integrado. A indústria eletrônica concentra seus esforços em estar sempre diminuindo a escala dos transistores a serem integrados, isso permite que uma quantidade maior de componentes possam ser inseridos em um mesmo circuito integrado. Porém as dificuldades em se reduzir a escala ainda mais, mostram que os esforços não devem ser direcionados apenas à redução dos componentes, mas também à etapa de desenvolvimento de *layouts* destes dispositivos. Baseado nesse cenário, e levando em conta a escassez de mão de obra especializada nesta área no Brasil, decidiu-se nesta monografia desenvolver o *layout* de um circuito integrado CMOS conhecido como contador síncrono de quatro bits, utilizando o software LASI (educacional), contribuindo assim com a disseminação de conhecimentos pertinentes ao projeto de circuitos integrados. Adicionalmente, apresentou-se também as simulações do circuito proposto, realizadas por meio da plataforma SPICE (LTSPICE IV), garantindo o funcionamento pleno do *layout* desenvolvido. Realizou-se também a verificação de sensibilidade à temperatura do circuito e algumas outras simulações a fim de definir parâmetros de operação do circuito. Por fim, o conjunto das simulações realizadas forneceu a validação do funcionamento do circuito desenvolvido.

Palavras chave: Circuitos Integrados, Microeletrônica, LASI, *Layout*, *Standard cell*.

ABSTRACT

Currently, it has been observed how the use of integrated circuits is important for technology development, it not only keeps the growth of the electronics industry, but it's also a part of human daily life, being present in practically all electronic devices used in the routine. The fact that allowed the development of integrated circuits was the domain and the manipulation of semiconductor materials, which led to the creation of the transistor, a key device for the integration of components into an integrated circuit. The electronics industry focuses its efforts on always reducing the scale of the transistors to be integrated, this allows a greater amount of components to be inserted in the same integrated circuit. However, the difficulties of further scale reducing, show that the efforts must be directed not only to the reduction of components scale, but also to the layouts development stage of these devices. Based on this scenario and considering the lack of skilled labor in this area in Brazil, it was decided in this monograph to develop the layout of an integrated circuit CMOS, known as a synchronous 4-bit counter that uses LASI (educational software), as a way to contribute to the dissemination of relevant knowledge to the Integrated Circuits design area. Farther, the simulations of the proposed circuit, performed through SPICE platform (LTSPICE IV) are also shown, ensuring the full functioning of the layout developed. Finally, the circuit temperature sensitivity check and some other simulations were carried out in order to define circuit operation parameters.

Keywords: Integrated Circuits, microelectronics, LASI, layout, standard cell.

LISTA DE FIGURAS

FIGURA 1 – FOTOGRAFIA DO PRIMEIRO TRANSISTOR..	21
FIGURA 2 – FOTOGRAFIA DO PRIMEIRO CIRCUITO INTEGRADO	22
FIGURA 3 - MODELO DE BANDAS DOS TRÊS TIPOS DE MATERIAIS..	26
FIGURA 4 - LIGAÇÕES DE UM SEMICONDUTOR INTRÍNSECO.....	27
FIGURA 5 - LACUNAS EM UM SEMICONDUTOR EXTRÍNSECO TIPO P.....	28
FIGURA 6 - ELÉTRONS EM UM SEMICONDUTOR EXTRÍNSECO TIPO N.....	29
FIGURA 7 – SIMBOLOGIAS (A) NMOS E (B) PMOS	30
FIGURA 8 – CONSTRUÇÃO FÍSICA DO MOSFET..	31
FIGURA 9 – GRÁFICO DAS REGIÕES DE OPERAÇÃO DE UM MOSFET.....	32
FIGURA 10 – A) POLARIZAÇÃO B) FORMAÇÃO DO CANAL C) LIBERAÇÃO DE ELÉTRONS.....	33
FIGURA 11 – MOSFET COMO RESISTÊNCIA VARIÁVEL.	33
FIGURA 12 – PASSOS PARA OBTENÇÃO DO WAFER DE SILÍCIO.....	35
FIGURA 13 – PROCESSO DE GRAVAÇÃO DA MÁSCARA COM O LAYOUT.....	36
FIGURA 14 – OBTENÇÃO DO LAYOUT GRAVADO NO SILÍCIO..	36
FIGURA 15 – OBTENÇÃO DE UM CANAL N NO WAFER TIPO P.....	37
FIGURA 16 – FLUXOGRAMA DO PROJETO DE UM CI	43
FIGURA 17 – SÍMBOLO DA PORTA INVERSORA	45
FIGURA 18 – CIRCUITO INVERSOR CMOS	46
FIGURA 19 – SIMBOLOGIA NAND	47
FIGURA 20 – CIRCUITO ESQUEMÁTICO SPICE DA FUNÇÃO NAND	47
FIGURA 21 – ESQUEMÁTICO SPICE DA NAND DE TRÊS ENTRADAS	48
FIGURA 22 – SIMBOLOGIA NAND DE TRÊS ENTRADAS	48
FIGURA 23 – CIRCUITO LÓGICO E SIMBOLOGIA DO FLIP-FLOP JK	50
FIGURA 24 – ESQUEMÁTICO DE UM CONTADOR SÍNCRONO DE 4 BITS	51
FIGURA 25 – CONTAGEM BINÁRIA DO CONTADOR SÍNCRONO	51
FIGURA 26 – REGRAS DE PROJETO DE 1 A 48	53
FIGURA 27 – TELA DE CONFIGURAÇÃO DO DRC	54
FIGURA 28 – TELA DE CONFIGURAÇÃO DO LASI CKT	56
FIGURA 29 – NMOS DESENVOLVIDO EM LASI.....	58
FIGURA 30 – TELA FINAL DO DRC	59
FIGURA 31 – PMOS DESENVOLVIDO EM LASI.....	60

FIGURA 32 - INVERSOR DESENVOLVIDO NO LASI	62
FIGURA 33 - NAND DE DUAS ENTRADAS DESENVOLVIDA NO LASI	63
FIGURA 34 - NAND DE TRÊS ENTRADAS DESENVOLVIDA NO LASI	64
FIGURA 35 – NAND <i>FULL CUSTOM</i> À ESQUERDA E <i>STANDARD CELL</i> À DIREITA	65
FIGURA 36 – <i>FLIP-FLOP JK</i> DESENVOLVIDO NO LASI.	66
FIGURA 37 – <i>LAYOUT</i> DO CONTADOR DESENVOLVIDO NO LASI	67
FIGURA 38 – TELA FINAL DO DRC DO CONTADOR	68
FIGURA 39 – TELA DE COMPILAÇÃO DO CKT	69
FIGURA 40 – SINAIS DE ENTRADA E SAÍDA DO CIRCUITO	70
FIGURA 41 – SIMULAÇÃO DE SENSIBILIDADE À TEMPERATURA COM 30°C - VERDE, 55°C - AZUL E 80°C - VERMELHO	73
FIGURA 42 – <i>LAYOUT</i> DO CONTADOR PROJETADO	81

LISTA DE TABELAS

TABELA 1 – TABELA VERDADE DO BLOCO INVERSOR	46
TABELA 2 – TABELA VERDADE DA FUNÇÃO NAND DE DUAS ENTRADAS	47
TABELA 3 – TABELA VERDADE DA NAND DE TRÊS ENTRADAS	49
TABELA 4 – TABELA VERDADE DO FLIP-FLOP TIPO T	50
TABELA 5 – TEMPOS DE SUBIDA E DESCIDA DO CONTADOR	71
TABELA 6 – FREQUÊNCIA DOS ESTÁGIOS DO CONTADOR	72
TABELA 7 - FAIXA DE OPERAÇÃO GARANTIDA	74
TABELA 8 - CARACTERÍSTICAS ACIMA DA FAIXA NORMAL DE TEMPERATURA	74

LISTA DE ABREVIATURAS E SIGLAS

CMOS – *Complementary Metal-Oxide-Semiconductor.*

CI – *Circuito Integrado.*

LASI – *Layout Software for Individuals.*

ASIC - *Application-Specific Integrated Circuit.*

FPGA - *Field-Programmable Gate Array.*

MOSFET – *Metal-Oxide-Semiconductor Field Effect Transistor.*

FET - *Field Effect Transistor.*

CAD – *Computer Aided Design.*

ROM – *Read Only Memory.*

PLA – *Programmable Logic Array.*

RAM – *Random Access Memory.*

VHDL – *Very High Speed Hardware Description Language.*

PPM – *Partes Por Milhão.*

VLSI – *Very Large Scale Integrated.*

IEEE – *Institute of Electrical and Electronic Engineers.*

SPICE – *Simulation Program with Integrated Circuit Emphasis.*

MEMS – *Micro Electro Mechanical Systems.*

DRC – *Design Rules Checking.*

CKT – *Circuit.*

SOI – *Silicon On Insulator.*

SUMÁRIO

1. INTRODUÇÃO	14
1.1 CARACTERIZAÇÃO DO PROBLEMA	15
1.2 OBJETIVOS	16
1.3 JUSTIFICATIVA	17
1.4 ORGANIZAÇÃO DO TEXTO	17
2. FUNDAMENTAÇÃO TEÓRICA	19
2.1 HISTÓRICO DO SURGIMENTO DOS CIRCUITOS INTEGRADOS.....	19
2.1.1 A DESCOBERTA DO TRANSISTOR	19
2.1.2 O DESENVOLVIMENTO DOS CIRCUITOS INTEGRADOS	22
2.1.3 EVOLUÇÃO DOS PROCESSADORES E DO PROJETO DE CIs.....	23
2.2 MATERIAIS SEMICONDUTORES	26
2.2.1 DOPAGEM DOS SEMICONDUTORES	27
2.3 O MOSFET.....	29
2.4 A TECNOLOGIA CMOS	34
2.5 ETAPA DE <i>DESIGN</i> (PROJETO) DOS CIs.....	37
2.5.1 LINGUAGEM DE DESCRIÇÃO DE HARDWARE	38
2.5.2 PLATAFORMA SPICE.....	38
2.5.3 LASI – LAYOUT SYSTEM FOR INDIVIDUALS.....	40
2.6 O CONTADOR	41
3. METODOLOGIA.....	43
3.1 DESENVOLVIMENTO DOS LAYOUTS	44
3.2 VERIFICAÇÃO DRC	52
3.3 EXTRAÇÃO DA NETLIST (LASI CKT)	54
4. RESULTADOS E DISCUSSÕES	57
4.1 DESENVOLVIMENTO DOS LAYOUTS	57
4.1.1 TRANSISTOR NMOS.....	58
4.1.2 TRANSISTOR PMOS.....	60
4.1.3 PORTA NOT (INVERSORA)	61
4.1.4 PORTA NAND (NÃO-E) DE DUAS ENTRADAS	63
4.1.5 PORTA NAND (NÃO-E) DE TRÊS ENTRADAS.....	64

4.1.6	<i>FLIP-FLOP JK</i>	66
4.1.7	<i>CONTADOR SÍNCRONO DE QUATRO BITS</i>	67
5.	CONCLUSÕES	75
6.	TRABALHOS FUTUROS	77
7.	REFERÊNCIAS BIBLIOGRÁFICAS	78
8.	ANEXO I	81
9.	ANEXO II	82

1. INTRODUÇÃO

Os circuitos integrados são considerados a descoberta tecnológica mais significativa e importante do século XX, pois a descoberta e evolução destes dispositivos permitem a expansão tecnológica na área da eletrônica à velocidade que esta expansão se dá atualmente (KANG, 2003).

O conceito de circuito integrado (CI) foi proposto em 1958 por Jack Kilby, quando desenvolveu o primeiro dispositivo desta forma, vindo a receber o prêmio Nobel mais tarde no ano de 2000 (CHEAH, 2008).

A principal razão para a grande evolução da indústria eletrônica é o avanço no desempenho dos circuitos integrados, enquanto os custos de fabricação se mantêm fixos ou às vezes diminuem; além do fato destes dispositivos se tornarem cada vez menores, ocupando menos espaço e permitindo maior portabilidade nas aplicações dos mesmos, o que permite que os circuitos que utilizam os mesmos se tornem menores, mais confiáveis e dissipem menos potência.

Como a utilização dos circuitos integrados também reduz a quantidade de componentes em um circuito eletrônico, os CIs também aumentam a confiabilidade de um circuito, pois a confiabilidade de um circuito está diretamente relacionada à quantidade de componentes neste circuito, porque quanto mais componentes há em um circuito maior a chance de falha do mesmo, além do fato de que componentes fixos (como os componentes internos do CI) são mais confiáveis do que componentes variáveis (discretos) (KANG, 2003).

Porém, o advento dos circuitos integrados se deve à descoberta e à evolução de outro componente eletrônico, o transistor. Estes dispositivos foram descobertos e evoluíram a partir do domínio da utilização dos materiais semicondutores, pois os transistores trouxeram grande avanço para a indústria eletrônica que viu a possibilidade de substituir a válvula eletrônica (dispositivo eletrônico que dominava a indústria eletrônica na época, por permitir o controle de corrente) que era grande, frágil, dissipava altas potências e produzia muito calor (CHEAH, 2008).

Na próxima seção encontra-se a descrição da caracterização do problema no qual esta monografia se baseia.

1.1 CARACTERIZAÇÃO DO PROBLEMA

Cada vez mais fica explícita a importância da indústria eletrônica na economia mundial, considerando que os componentes e equipamentos eletrônicos estão presentes em praticamente todos os setores. Mas no Brasil, a indústria eletrônica depende de acordos externos, importações e do uso de produtos padronizados para operar. Pois o desenvolvimento e produção de componentes eletrônicos são restritos a produtos que não requerem alta complexidade tecnológica e conseqüentemente possuem baixo valor agregado (LOUREIRO, 2011). Além disso, a indústria brasileira não consegue destaque no que diz respeito ao uso da microeletrônica, pois o conhecimento na área e inclusive a capacidade industrial são escassos. Devido a isso, as empresas nacionais optam pela importação de componentes prontos, sem que haja uma diferenciação ou mesmo capacidade de inovação, com relação aos competidores globais do setor.

O desenvolvimento de um circuito integrado, ou mesmo a compra de um já existente no mercado é uma das etapas constituintes do processo de criação de um novo produto eletrônico, e este é um investimento que deve ser bem planejado pelas empresas, pois está relacionado a gastos e ao desempenho final do dispositivo, portanto, esta etapa está diretamente relacionada ao custo benefício do produto, além disso, cada alternativa apresenta suas vantagens e complicações. Por exemplo, a compra de componentes prontos apresenta menores custos, mas não permite uma grande diferenciação no que diz respeito a inovação tecnológica e desempenho visto que o produto utiliza componentes padrões utilizados também por outras companhias. Outro exemplo é a negociação com empresas estrangeiras (devido ao fato de não haver boas opções nesta cadeia produtiva no Brasil) da manufatura de um ASIC (*Application-Specific Integrated Circuit* – Circuito Integrado de Aplicação Específica), que é um circuito que desempenha uma função específica dentro de sua aplicação, que proporciona assim um melhor desempenho do produto final, porém se apresenta como uma opção de custo elevado. Um último exemplo, que vale ser citado é a utilização de um FPGA (*Field-Programmable Gate Array*), que um circuito integrado padronizado, que pode ser programado para diferentes aplicações, mas que se

restringe a aplicações de baixo volume por apresentar alto custo unitário e elevado consumo de energia (LOUREIRO, 2011).

Surge então uma nova alternativa, devido ao fato de o governo ter identificado a carência nacional nessa área, e investir através do Ministério da Ciência e Tecnologia em um programa de formação de projetistas de circuitos integrados chamado CI-Brasil. Esta nova alternativa permite que as empresas brasileiras negociem o projeto do circuito integrado aqui no Brasil, porém a manufatura do mesmo ainda tem de ser realizada no exterior, o que apresenta uma redução de custos com relação à importação total do projeto, mas que faz com que a operacionalização seja complicada devido ao alto nível de interação necessário entre projetistas e equipe de produção (LOUREIRO, 2011).

Como se pode observar o mercado brasileiro ainda é frágil, mas os investimentos iniciais para formação de projetistas ajudam a criar um ambiente atrativo para empresas e além disso, o governo também tem uma linha de investimento no fomento à indústria e ao desenvolvimento local deste setor (LOUREIRO, 2011).

Baseado neste cenário e em todas as questões levantadas, no trabalho de conclusão de curso desenvolveu-se o *layout* de um contador, visando a formação de mão de obra qualificada e a disseminação do ensino da área de projeto de CIs.

1.2 OBJETIVOS

O objetivo geral desta monografia é o desenvolvimento do projeto de *layout* de um circuito integrado CMOS contador síncrono de quatro *bits* utilizando ferramentas educacionais, bem como a realização de todas as simulações necessárias para garantir a funcionalidade do *layout* desenvolvido.

Para que o objetivo geral fosse consolidado, outros objetivos específicos foram alcançados. Os objetivos específicos são:

- Realizar um estudo sobre os materiais semicondutores;
- Realizar um estudo sobre o funcionamento dos MOSFETs e CMOS;

- Realizar um estudo sobre o estado da arte dos circuitos integrados;
- Gerar o conhecimento e habilidades em projeto de CIs;
- Disseminar da formação de mão de obra qualificada na área;
- Disseminar do uso dos *softwares* educacionais;
- Desenvolver bibliotecas de *standard cells* (CIs) para a construção final do circuito;
- Implementar o *layout* do contador, bem como, otimizar o mesmo;
- Analisar o comportamento do contador em função das condições de temperatura, sinais aplicados e técnicas empregadas.

1.3 JUSTIFICATIVA

O desenvolvimento do *layout* proposto do contador síncrono de quatro *bits*, se justifica pelo fato de ser uma estrutura com complexidade relativamente alta, uma vez que tais estruturas são utilizadas na construção de processadores e componentes eletroeletrônicos, tais como, computadores, *smartphones*, microcontroladores, etc. Para a implementação de qualquer *layout* de circuito integrado é necessário a utilização de softwares que na maioria das vezes são ferramentas caríssimas (CADENCE TOOLS, 2017)(PSPICE TOOL, 2017), porém, nesse trabalho foram utilizados *softwares* gratuitos educacionais, LASI e LTSPICE IV, para a confecção do *layout* proposto. Dessa forma, esta monografia contribuirá também para disseminar conhecimentos relativos ao projeto de *layout* de CIs, ajudando indiretamente na formação de mão-de-obra qualificada na área, ressaltando que todo o projeto foi desenvolvido com ferramentas computacionais gratuitas.

1.4 ORGANIZAÇÃO DO TEXTO

Essa monografia é organizada em 7 capítulos, sendo que no Capítulo 1 apresenta-se o projeto, expondo uma breve contextualização e apresentando a problemática vislumbrada, assim como os objetivos geral e específicos. O Capítulo 2

apresenta os conceitos teóricos necessários para a compreensão desse trabalho por meio de uma revisão da literatura relacionada à área de circuitos integrados, abordando conceitos físico-químicos nos quais se baseia o funcionamento destes dispositivos, os processos de construção dos mesmos e até os *softwares* utilizados para projetá-los. Já o Capítulo 3, apresenta a metodologia utilizada para confecção dos *layouts* propostos. Os resultados obtidos e as discussões dos mesmos são apresentados no Capítulo 4, assim como as possíveis análises considerando a proposição do trabalho. No Capítulo 5, são apresentadas as conclusões do projeto proposto. Algumas sugestões para trabalhos futuros são citadas no Capítulo 6, e por fim, as referências bibliográficas utilizadas são dispostas no Capítulo 7.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentado o levantamento bibliográfico visando a compreensão do estado-da-arte dos circuitos integrados, bem como explicitar sua fundamentação em conceitos teóricos.

2.1 HISTÓRICO DO SURGIMENTO DOS CIRCUITOS INTEGRADOS

O desenvolvimento dos circuitos integrados deve-se ao surgimento do transistor (TRANSfer-varISTOR), dispositivo semicondutor fundamental para a integração de outros dispositivos na confecção dos projetos de CIs. O aperfeiçoamento dos CIs só é possível devido a evolução das técnicas de construção dos materiais semicondutores que compõem os transistores e os demais dispositivos de integração. Dessa forma, na próxima subseção são apresentados conceitos pertinentes a descoberta do transistor.

2.1.1 A DESCOBERTA DO TRANSISTOR

Na primeira metade do século XX, a indústria eletrônica era dominada pelas válvulas eletrônicas, também conhecidas como válvulas termiônicas. Estas válvulas utilizavam do efeito, hoje conhecido como efeito termiônico, que se dá no vácuo, através do aquecimento de um eletrodo com polaridade negativa (permite que seus elétrons fiquem quase livres), quando uma placa com polaridade positiva é inserida no sistema os elétrons tendem a fluir do eletrodo para a placa (*University of Denver*, 2002).

Como as válvulas eram os dispositivos mais utilizados na época, o primeiro computador eletrônico da história, o ENIAC (*Electronic Numerical Integrator Analyzer and Computer* – Computador Integrador Numérico Eletrônico), era composto por 18.000 destas, e surgiu em 1946 (REIS, 2008). Mas em função da baixa durabilidade

e da instabilidade das válvulas, os pesquisadores da época concentraram seus esforços na descoberta de novos dispositivos, tanto que em 1926 Lilienfeld desenvolveu o conceito de um transistor de efeito de campo (FET – *field-effect-transistor*), vindo a receber as patentes referentes ao conceito, porém naquela época não havia quem fosse capaz de construir fisicamente tal dispositivo (EDGAR, 1926).

Por volta de 1936, os pesquisadores estavam com o sentimento de que a válvula não seria a última solução para os eletrônicos e se iniciou um grupo de pesquisa na empresa *Bell Laboratories*, tal grupo veio a descobrir em 1940 por meio do processo de dopagem, os cristais de silício tipo N e tipo P. Este foi o contexto da indústria eletrônica antes da segunda guerra mundial (CHEAH, 2008).

A guerra, porém, interrompeu parte da evolução das pesquisas, pois os laboratórios precisaram dedicar ao máximo os esforços para desenvolver equipamentos que auxiliassem o país na guerra, portanto os pesquisadores foram espalhados. De volta as pesquisas em 1946 o grupo, auxiliado por ideias de outros pesquisadores decidiram por focar seus trabalhos em dois elementos, o silício e o germânio. A partir de então Shockley reviveu a ideia de desenvolver um componente a base de efeito de campo, os trabalhos progrediram nesta direção, porém era necessário reduzir as interferências causadas pelos efeitos das superfícies de contatos dos dispositivos até então desenvolvidos (CHEAH, 2008).

Superados os obstáculos, em novembro de 1947, nos laboratórios da empresa de telefonia *Bell Laboratories*, desenvolveram o primeiro transistor funcional, conforme mostra a Figura 1. A descoberta foi sigilosa até que em junho de 1948, desenvolveu-se o primeiro protótipo de circuito, obtendo-se amplificação da voz por meio do novo dispositivo. Comparando este às válvulas, estes são menores, mais duráveis, mais confiáveis, consomem menos energia e produzem menos calor, portanto um substituto para as válvulas (CHEAH, 2008).

Na montagem da Figura 1, observa-se a base (por isso se usa base para transistores até os dias atuais), e os outros dois terminais, coletor e emissor. Pela primeira vez, com esta configuração, foi possível se observar um ganho (amplificação de sinal controlada pela corrente de base). Deste ponto em diante, os cientistas estavam prontos para melhorarem cada vez mais o dispositivo criado, pois estavam entendendo suas peculiaridades, o transistor foi, então, aprimorado e miniaturizado

para transistor bipolar em 1950 (primeiro transistor de junção NPN) (BRINKMAN, *et al*, 1997). Em 1952, foi desenvolvido por Ian Ross e George Dacey um dispositivo unipolar, tal dispositivo foi o precursor do atual FET.

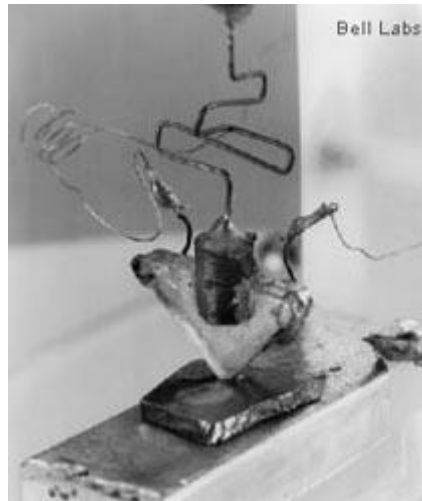


Figura 1 – Fotografia do primeiro transistor. Figura extraída de: (GUIZZO, 2000).

Já em 1955 a Sony, obtendo a licença do novo dispositivo desenvolvido na Bell Labs, desenvolveu o primeiro rádio transistor, permitindo então que todas as pessoas obtivessem o aparelho em suas residências e tornando industrial (produção em larga escala) o novo dispositivo (BRINKMAN, *et al*, 1997).

A invenção do transistor revolucionou a eletrônica naquele contexto, e trouxe muito progresso tanto no que diz respeito à eletrônica (substituindo as válvulas), quanto ao estudo e entendimento dos materiais. Pois a partir da descoberta deste componente se entendeu boa parte do comportamento dos metais, isolantes e principalmente dos semicondutores, bem como se dominou o processo de dopagem e as tecnologias para produção dos cristais tão utilizados na atualidade. A invenção do transistor também permitiu a evolução dos circuitos integrados, o que permite na atualidade a construção de aparelhos cada vez mais versáteis e rápidos, tornando-se úteis aos seres humanos.

É importante lembrar que estes componentes continuam em constante evolução, e que com o passar do tempo se tornam cada vez mais usuais.

Na próxima subseção são apresentados conceitos referentes aos circuitos integrados, que são dispositivos que evoluíram a partir dos transistores e são a base dos objetivos deste trabalho.

2.1.2 O DESENVOLVIMENTO DOS CIRCUITOS INTEGRADOS

A descoberta do transistor estimulou os engenheiros a desenvolverem variados tipos de circuitos discretos (circuitos que possuem componentes eletrônicos individuais), que se mostravam cada vez maiores e mais complexas de componentes. O grande problema destes circuitos era que os componentes ainda tinham de ser interconectados, por meio de cabos e soldas, o que gastava tempo e custava caro. A partir disso, Jack Kilby, no laboratório da empresa *Texas Instruments*, desenvolveu em 1958 o primeiro circuito integrado conforme mostra a Figura 2, que continha apenas um transistor e alguns outros elementos com interligações metálicas em um pedaço de germânio, mas que prometia revolucionar, mais uma vez, a indústria eletrônica (CHEAH, 2008).



Figura 2 – Fotografia do primeiro circuito integrado. Figura extraída de: (TEXAS INSTRUMENTS).

Em 1959, os futuros fundadores da Intel Jean Hoerni e Robert Noyce, desenvolveram nos laboratórios da *Fairchild Semiconductor* um novo processo chamado tecnologia planar, que consistia em difundir várias camadas sobre a superfície de um *wafer* de silício para construir um transistor, deixando uma camada de óxido entre as junções para proteção. Este novo processo permitiu então o início da fabricação dos CIs, pois eliminava de vez as conexões metálicas e as soldas.

Houve resistência para a adoção deste novo método de produção, porém no fim de 1960, cerca de 90% de todos os novos produtos eletrônicos utilizavam circuitos integrados. E a partir de então seu uso cresce todos os dias (CHEAH, 2008).

As tamanhas implicações das descobertas de Kilby e dos fundadores da Intel não poderiam ser imaginadas naquela época, porém nos dias atuais, a maioria dos dispositivos eletrônicos disponíveis no mercado não poderiam ser desenvolvidos sem a utilização dos circuitos integrados.

Na próxima subseção apresenta-se um breve histórico dos processos de projeto de circuitos integrados.

2.1.3 EVOLUÇÃO DOS PROCESSADORES E DO PROJETO DE CIs

O desenvolvimento dos primeiros circuitos integrados permitiu uma integração cada vez maior de componentes em um mesmo *chip*, o que culminou no desenvolvimento de microprocessadores e também exigiu melhorias na etapa de projeto dos CIs.

Então, em 1964, surgiu a família de portas lógicas denominada TTL (Lógica Transistor - Transistor). Em 1971 a Intel lança o primeiro microprocessador denominado 4004, e em seguida em 1974 lança o Intel 8080 (um microprocessador de 8 bits), que encontrou concorrência do M6800 da Motorola no mesmo ano e do Z80 da Zilog em 1976 que possuíam respectivamente 4.800 e 8.200 transistores integrados em suas versões iniciais. Hoje em dia existem microprocessadores que apresentam arquiteturas de 64 e até 128 bits (REIS, 2008).

Com o desenvolvimento e evolução destas tecnologias de fabricação dos circuitos integrados, atualmente se permite a implantação física de centenas de milhões de transistores em uma única pastilha de silício, ou seja, em um único CI.

Até o fim de 1970, a etapa de projeto destes CIs permaneceu praticamente a mesma (projeto quase manual, que utilizava apenas do apoio de ferramentas CAD básicas – *Computer Aided Design* – Desenho Auxiliado por Computador), porém com o crescente aumento na capacidade de integração (número de transistores por CI),

surgiu a necessidade de se automatizar também o projeto dos CIs, o que inicialmente teve de ser efetuado na etapa de *layout* dos mesmos, pois o “desenho” do *layout* era feito manualmente.

Já no final dos anos 70, quando se atingiu a marca de 20.000 transistores em um *chip*, apareceram os primeiros circuitos que apresentavam blocos funcionais gerados automaticamente. Este fato acarretou uma mudança na estrutura dos blocos funcionais utilizados, pois estes eram gerados manualmente e utilizavam-se apenas de um mapeamento por lógica aleatória, porém a automação desta lógica aleatória seria tão complexa que a solução utilizada foi a criação de blocos regulares (do tipo ROM – *Read Only Memory* ou Memória Apenas de Leitura - e/ou PLA - *Programmable Logic Array* ou Matriz Lógica Programável), nesta época então surgiram os microprocessadores com blocos de controle microprogramados (REIS, 2008).

Utilizou-se por um bom tempo os blocos microprogramados, porém ao fim da década de 1980, os blocos de controle de processadores integrados passaram a utilizar bibliotecas de células pré-desenhadas chamadas *standard cells* (tecnologia que é utilizada até os dias atuais). Tal mudança acarretou a melhoria do desempenho do CI, uma vez que blocos em ordem aleatória são mais rápidos do que blocos ROM/PLA (REIS, 2008). Porém a tecnologia *standard cells* também tem seus problemas, que são a área perdida com roteamento e a limitação do número de células existentes em uma biblioteca e do dimensionamento dos componentes de uma célula, além dos custos de manutenção da biblioteca devido à evolução das tecnologias vigentes.

Houve então um aumento na utilização das memórias *cache* (RAM) no interior dos microprocessadores, o que é determinante no aumento do desempenho dos mesmos.

A partir disso, buscou-se o desenvolvimento de ferramentas CAD, que permitissem a concepção de circuitos nos níveis funcional e comportamental. Hoje a especificação de um circuito ou sistema é efetuada utilizando-se linguagens de descrição de hardware (VHDL – *Very High Speed Hardware Description Language* - ou VERILOG – Linguagem de Descrição de Hardware), o que permite obter o *layout* através da análise da descrição comportamental do sistema. Portanto há uma busca contínua na direção do projeto conjunto de *hardware* e *software* de um sistema, e esta

evolução contínua é chamada de *hardware-software codesign*. Amplia-se então os esforços no desenvolvimento de linguagens e ferramentas que permitam esta especificação a nível de sistema (REIS, 2008).

Busca-se então um processo automatizado para gerar o *layout* que se aproxime o máximo do processo manual de concepção, que consiga processar o circuito de forma funcional e de forma comportamental antes de gerar o *layout*.

Há também, uma concentração de esforços na busca por mecanismos para sintetizar o sistema como um todo, tanto na parte estrutural como na parte lógica. Em *hardware-software codesign* a implementação de um algoritmo pode ser feita tanto em *software* quanto em *hardware*, e para definir quais equações serão implementadas em um ou em outro, é feita uma análise da descrição hierárquica a fim de identificar onde será implementado cada módulo.

Ao saber quais módulos serão implementados em *hardware*, é realizada uma descrição comportamental do sistema a ser integrado e em seguida é realizada uma etapa de síntese comportamental. Esta síntese identifica as equações da parte operativa do sistema e as equações de controle do mesmo, a partir disso sintetiza-se a estrutura, atrelando as operações da parte operativa às suas respectivas estruturas, alocando-as da melhor forma possível. É feita também uma previsão do sequenciamento destas operações, ou seja, em que momento (ciclo de relógio) cada uma delas será executada.

Após a síntese comportamental o sistema está pronto para a síntese lógica, onde serão definidas quais portas serão utilizadas na implementação física (agora as equações lógicas serão sofrerão uma otimização em função do mapeamento físico do *layout*). Estas etapas de síntese são efetuadas antes da etapa de síntese de *layout*, pois finalizado o *layout* o circuito integrado estará pronto para ser reproduzido pela *foundry* (fábrica que irá reproduzir os circuitos integrados) (REIS, 2008).

A próxima subseção introduz os materiais semicondutores e as tecnologias utilizadas para que os mesmos trabalhem da maneira esperada dentro dos dispositivos, pois os mesmos desempenham papel fundamental para o funcionamento dos CIs.

2.2 MATERIAIS SEMICONDUTORES

Conforme visto nas seções anteriores, a descoberta dos materiais semicondutores fomentou a base para que se desenvolvessem os componentes necessários à evolução da indústria eletrônica, portanto é fundamental conhecer o comportamento dos materiais semicondutores, visando a compreensão funcional dos transistores e CIs. Neste tópico são expostas algumas características importantes deste material.

Para a compreensão dos materiais semicondutores é importante a comparação entre o preenchimento da camada de valência por elétrons dos três tipos de materiais (condutores, semicondutores e isolantes). Os condutores possuem a banda de condução e a banda de valência em sobreposição, de modo que os elétrons já preenchem a banda de condução, fazendo com que estes tenham predisposição para conduzir corrente elétrica. Os isolantes possuem a banda de condução e a banda de valência separadas e é necessária uma grande quantidade de energia para levar um elétron de uma para a outra, isso faz com que este material se oponha à passagem de corrente elétrica. Já os semicondutores que são materiais que podem se comportar como ambos (condutores e isolantes), possuem as bandas separadas, porém uma pequena quantidade de energia é necessária para que este tipo de material passe para um estado de condução de corrente (STRYHALSKI, 2005).

As configurações descritas e as diferenças entre os três tipos de materiais são justificadas pela estrutura formada pelos mesmos após a ligação e pelo modelo de bandas conforme mostra a Figura 3.

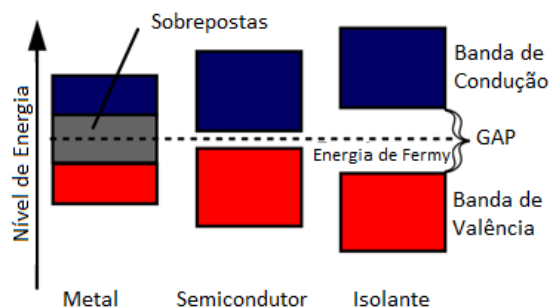


Figura 3 - Modelo de bandas dos três tipos de materiais. Figura adaptada de (FONTANA, 2011).

A Figura 3 apresenta a organização segundo o modelo de bandas de energia dos diferentes tipos de materiais, onde é possível ver que há um *gap* de energia, nos semicondutores e nos isolantes, entre a banda de valência (banda formada pela sobreposição de vários átomos com a mesma quantidade de elétrons na camada de valência onde ficam os elétrons no em condições normais na estrutura do material) e a banda de condução (banda para a qual os elétrons saltam quando recebem energia suficiente para se desprenderem da banda de valência), o que torna mais difícil a condução nos mesmos. Devido ao *gap* nos semicondutores ser menor do que nos isolantes, a quantidade de energia necessária para que os elétrons de valência superem o *gap* e passem à banda de condução é pequena, o que faz com que os semicondutores sejam utilizados em situações onde se deseja controlar a condução de corrente. Mas estes não são utilizados em sua forma pura, pois para que sejam empregados nestas aplicações, necessita-se realizar um processo conhecido como dopagem, que intensifica as propriedades condutoras e de controle de condução deste material (STRYHALSKI, 2005).

A próxima subseção mostra como se dá o processo de dopagem.

2.2.1 DOPAGEM DOS SEMICONDUTORES

No seu estado fundamental, o semicondutor intrínseco (alta pureza) realiza quatro ligações covalentes. A Figura 4 mostra o modelo de compartilhamento de elétrons formado por um semicondutor como o Silício (semicondutor mais utilizado) (MALVINO, 2015).

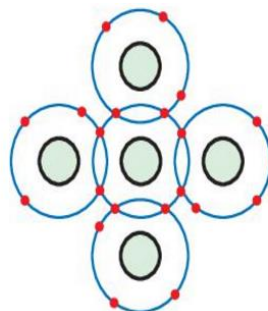


Figura 4 - Ligações de um semicondutor intrínseco. Figura extraída de: (MALVINO, 2015).

Devido ao fato de possuir quatro elétrons de valência e formar quatro ligações covalentes, os materiais semicondutores praticamente não possuem elétrons livres, o que faz com que seja difícil fornecer energia aos mesmos para alcançar um estado de condução. Para resolver esse problema, é necessário recorrer ao processo de dopagem, que é um processo de introdução de elementos adicionais (impurezas) na estrutura cristalina do semicondutor intrínseco. Tais impurezas são inseridas em baixas quantidades (da ordem de PPM – partes por milhão) e são compostas por elementos com três ou cinco elétrons de valência, o que respectivamente reduz ou aumenta a disponibilidade de elétrons livres (MALVINO, 2015).

Posteriormente ao processo de dopagem o semicondutor passa a ser extrínseco, além disso, a dopagem com elementos trivalentes cria um material tipo P (com excesso de lacunas – ou falta de elétrons livres) e a dopagem com elementos penta valentes cria um material tipo N (com excesso de elétrons livres).

2.2.1.1 Obtenção do material tipo P

Ao se dopar o semicondutor com um elemento que possua três elétrons de valência (Alumínio, Índio, Boro ou Gálio), obtemos um material do tipo P, este material possui excesso de lacunas, onde uma lacuna é a falta de um elétron na ligação individual (covalente) de um átomo do semicondutor com um átomo da impureza, é o buraco deixado pela falta de um elétron. A Figura 5 ilustra as lacunas de um material semicondutor tipo P (MALVINO, 2015).

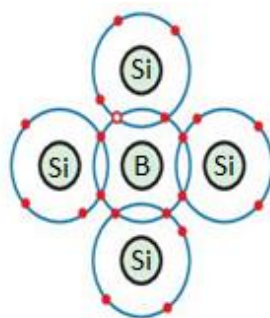


Figura 5 - Lacunas em um semicondutor extrínseco tipo P. Figura adaptada de: (MALVINO, 2015).

2.2.1.2 Obtenção do material tipo N

Ao se dopar o semicondutor com um elemento que possua cinco elétrons de valência (Antimônio, Fósforo ou Arsênio), obtemos um material do tipo N, este material possui excesso de elétrons livres, estes elétrons são aqueles que sobram nas ligações individuais (covalentes) dos átomos do semicondutor com os átomos da impureza, pois como o material tende a se estabilizar com oito elétrons de valência, realiza quatro ligações covalentes o que faz com que um elétron fique de fora das ligações. A Figura 6 ilustra os elétrons livres em um material semicondutor tipo N (MALVINO, 2015).

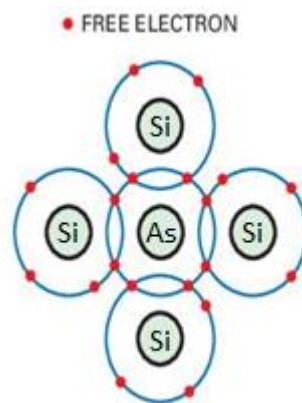


Figura 6 - Elétrons em um semicondutor extrínseco tipo N. Figura adaptada de: (MALVINO, 2015).

Com os materiais tipo P e tipo N, se torna possível a construção de dispositivos que atuam no controle de corrente como os transistores MOSFETs. A próxima subseção mostra aspectos construtivos e teóricos a respeito dos mesmos.

2.3 O MOSFET

O transistor MOSFET (*Metal Oxide Semiconductor Field-Effect-Transistor*) ou simplesmente MOS ainda é com certeza um dos carros chefes da área de projeto de

circuitos integrados digitais atualmente. O principal benefício na utilização dos mesmos em sistemas digitais é sua característica de funcionar muito bem como chave e introduzir poucos efeitos parasitas nos circuitos. Outras vantagens importantes são sua densidade de integração combinada com o processo de fabricação relativamente simples (planar), o que gera economia. Sua simbologia é mostrada na Figura 7, com suas duas variações NMOS (a) e PMOS (b) e com abordagem de 4 (esquerda) e de 3 terminais (direita) para ambos.

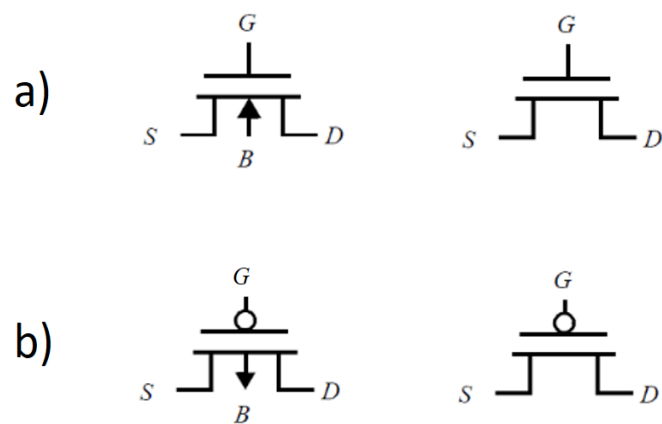


Figura 7 – Simbologias (a) NMOS e (b) PMOS. Figura extraída de: (RABAEY, et al, 2002).

Conforme visto, dois tipos de MOSFETs podem ser identificados, os MOSFETs canal N e os canal P (conforme processo de dopagem, composto por materiais tipo N e tipo P), também conhecidos respectivamente como NMOS e PMOS. Estes são dispositivos de quatro terminais chamados de *source* (fonte), *drain* (dreno), *gate* (porta) e *bulk* (substrato). E o funcionamento do mesmo consiste basicamente na aplicação de uma tensão no *gate* para se controlar o fluxo de corrente entre os pinos *source* e *drain*. O *bulk* ou *body* (corpo) é o quarto terminal do dispositivo e sua função é secundária, pois ele serve para moldar as características e parâmetros do dispositivo (RABAEY, et al, 2002). A construção do MOSFET sobre o *chip* de silício pode ser visualizada na Figura 8.

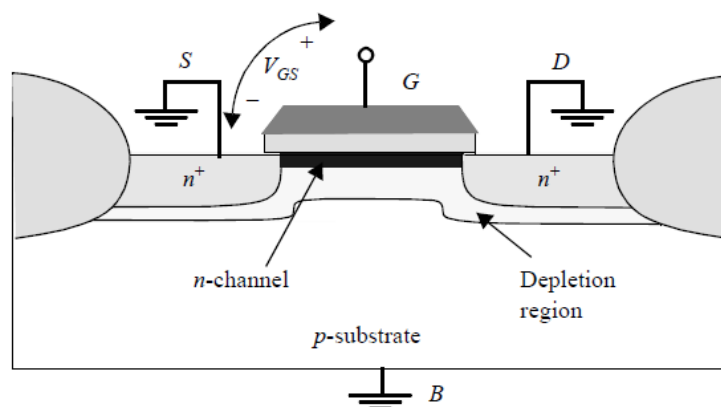


Figura 8 – Construção física do MOSFET. Figura extraída de: (RABAEY, *et al*, 2002).

No nível mais básico o transistor pode ser considerado uma chave (interruptor), pois quando a tensão aplicada no *gate* for maior que a tensão chamada tensão de limiar (V_{TH}), um canal de condução será formado entre a fonte e o dreno, e caso haja uma diferença de potencial entre estes dois últimos, haverá fluxo de corrente. A condutividade do canal será modulada pela tensão de *gate*, pois quanto maior a diferença de potencial entre *gate* e fonte, menor será a resistência do canal e maior será a corrente (RABAEY, *et al*, 2002).

A operação de um MOSFET se dá basicamente em três regiões distintas, a região de sublimiar (ou corte), a região de triodo e a região de saturação (ou *pinch off*). A região de sublimiar ocorre quando a tensão V_{GS} (Tensão *Gate-Source*) ainda não atingiu o valor mínimo necessário para que ocorra a criação do canal V_{TH} (Tensão de limiar), ou seja, dentro desta região, circulam pelo dispositivo apenas correntes de fuga, pois o mesmo ainda não está apto a operar efetivamente no controle de corrente. A região de triodo ocorre quando a tensão V_{GS} é maior do que a tensão de limiar, e nesse caso a corrente que flui entre fonte e dreno é controlada pelo aumento ou diminuição de V_{DS} (Tensão *Drain-Source*), que é uma tensão aplicada entre estes terminais. É também na região de triodo que fica evidente a principal característica dos MOSFETs, o fato de serem fontes de correntes controladas por tensão, pois ao observar a curva, percebe-se facilmente a diminuição da resistência do dispositivo de acordo com o aumento de V_{GS} , ou seja, é onde ocorre o controle efetivo de corrente ocasionado pela alteração da resistência do canal. Já a região de saturação, ocorre a partir do ponto na curva em que V_{DS} não mais ocasiona aumento na corrente que flui

pelo dispositivo, pois neste ponto o canal se encontra estrangulado (BAKER, 2010). A Figura 9 exibe a curva de operação de um MOSFET.

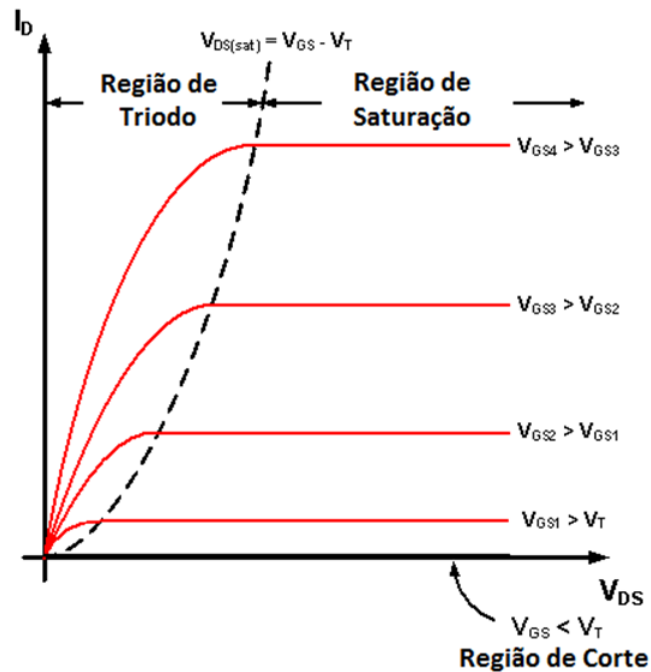


Figura 9 – Gráfico das regiões de operação de um MOSFET. Figura extraída de: (FERREIRA, 2012).

O funcionamento do MOSFET é mais complexo se comparado aos transistores de junção, e vários são os efeitos que influenciam na operação do mesmo, e a principal forma de se entender o funcionamento do MOSFET é estudando sua capacidade de funcionamento como um resistor variável.

O canal de condução formado entre a fonte e o dreno, pode ser visto como um resistor. Além do mais, desde que a densidade de elétrons que fluem pelo canal deve aumentar com o aumento da tensão de *gate* (V_G) esta resistência muda seu valor, permitindo maior passagem de elétrons. A Figura 10, mostra fisicamente como ocorre a formação do canal.

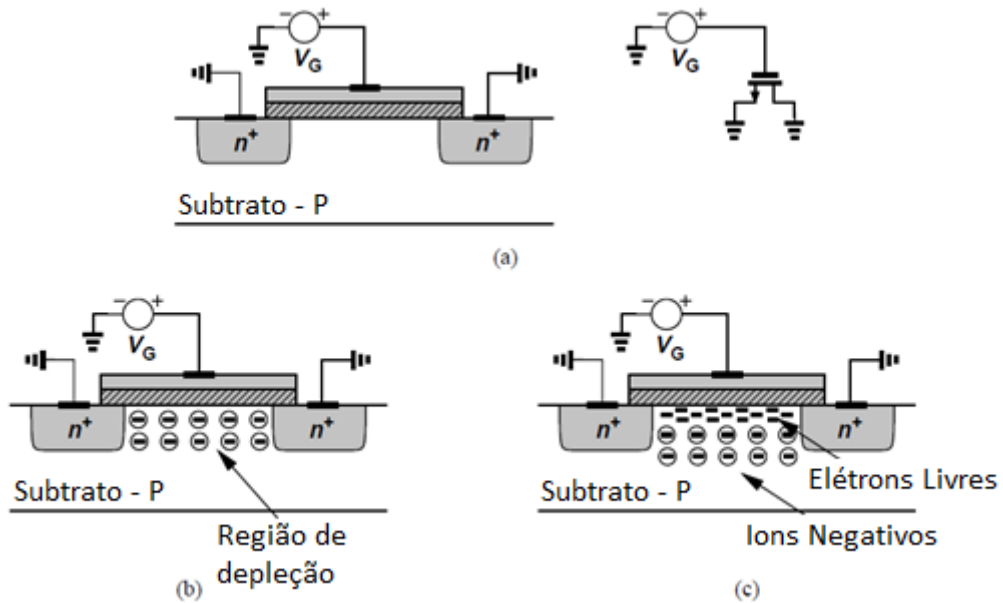


Figura 10 – a) Polarização b) Formação do canal c) Liberação de elétrons. Figura adaptada de: (BAKER, 2010).

Na Figura 10 (a) fica claro como deve ser a polarização para que o canal seja formado, em 10 (b) as tensões foram aplicadas e o canal começa a ser formado, e em 10 (c) houve a formação do canal com elétrons livres, porém ainda não há fluxo de elétrons porque os terminais fonte e dreno se encontram no mesmo potencial, a partir de então aumenta-se a tensão V_{DS} para que se inicie a condução de corrente (desde que V_{GS} seja maior que V_{TH}). A Figura 11 ilustra o MOSFET como resistência controlada por tensão, que é a característica que torna os MOSFETs extremamente usuais em aplicações analógicas e digitais.

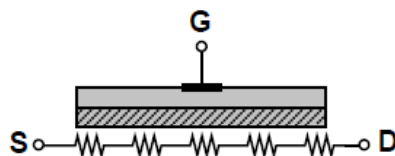


Figura 11 – MOSFET como resistência variável. Figura extraída de: (BAKER, 2010).

Além da tensão aplicada no *gate*, outros parâmetros de construção do transistor afetam a resistência entre os terminais fonte e dreno, e entre eles estão a largura e

comprimento do canal, bem como a espessura das camadas semicondutoras que constituem o canal. Todos estes parâmetros ficam a encargo do projetista para serem definidos (BAKER, 2010).

Os MOSFETs são utilizados dentro de outras tecnologias padrão para construção de CIs, o que é o caso da tecnologia CMOS. A próxima seção mostra as especificidades e aspectos teóricos que tornam a tecnologia CMOS usual nesta área.

2.4 A TECNOLOGIA CMOS

A tecnologia CMOS (*Complementary Metal-Oxide-Semiconductor*) foi proposta por Frank Wanlass em 1963 como uma tecnologia para construção de circuitos integrados. Esta tecnologia é utilizada na fabricação de microprocessadores, microcontroladores e em outras lógicas de circuitos digitais (BAKER, 2010).

Esta é empregada no processo de produção de CIs VLSI – *Very Large Scale Integrated* – termo geralmente associado a *chips* contendo milhares ou milhões de MOSFETs integrados e nos CIs ULSI - *Ultra Large Scale Integrated* – termo associado a *chips* com bilhões de transistores.

Esta tecnologia se tornou a mais utilizada na fabricação dos circuitos integrados por ser uma tecnologia confiável, manufaturável, garantir que os dispositivos consumam pouca energia e sejam de baixo custo (produção em larga escala), além disso, o mais importante é que os dispositivos podem mudar de escala, ou seja, a tecnologia permite que se reduzam as dimensões dos transistores garantindo que com o passar do tempo se agrupe uma quantidade cada vez maior de componentes em um mesmo espaço físico. Tal observação diz respeito à Lei de Moore, que disse em 1965 que a cada 18 – 24 meses se conseguiria dobrar a quantidade de transistores integrados em um mesmo espaço (BAKER, 2010).

O fato que permitiu que os transistores CMOS dominassem a produção de CIs foi a característica de que estes não consomem potência, ou seja, não há corrente elétrica circulando quando estes estão no estado estático, sendo assim estes só consomem energia na transição de um estado ao outro. No início não justificava

utilizar o CMOS, porém com o aumento da densidade e da aplicação dos circuitos integrados, o consumo de potência foi se tornando quase proibido, e então passou a se utilizar em larga escala a tecnologia CMOS. Além disso, a utilização dos CMOS fez com que se reduzisse a preocupação com resfriamento, pois uma vez que o consumo de potência seria menor, menor também seria o aquecimento (SWART, 2001).

O processo de produção CMOS se inicia com o material, que é o *wafer* de silício, que é cortado de um lingote, polido e dopado para que se inicie o processo de produção, a Figura 12 mostra o material (silício granulado) e as etapas até que se obtenha o silício preparado.

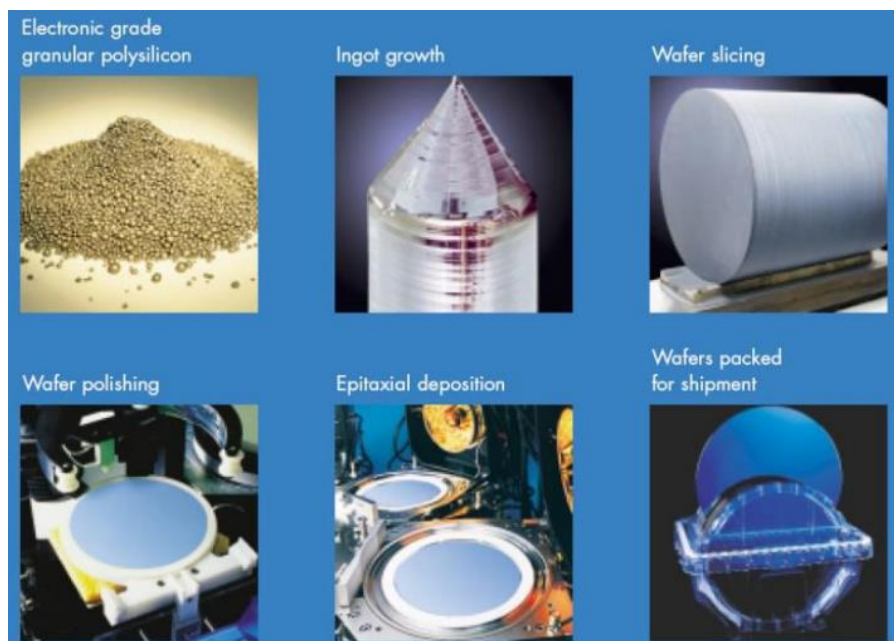


Figura 12 – Passos para obtenção do *wafer* de silício. Figura extraída de: (GOMES, 2014).

Para se gravar o *layout* proposto no silício, a próxima etapa, é a inserção da camada de óxido protetor, seguida da deposição da camada de *photoresist* (material fotossensível). Após isso realiza-se a gravação da máscara no *photoresist*, expõe-se o conjunto à luz ultravioleta para que a máscara marque o óxido e realiza-se a dopagem da região desejada formando o componente desejado (BAKER, 2010).

A Figura 13 mostra como é realizada a gravação da máscara no *photoresist*.

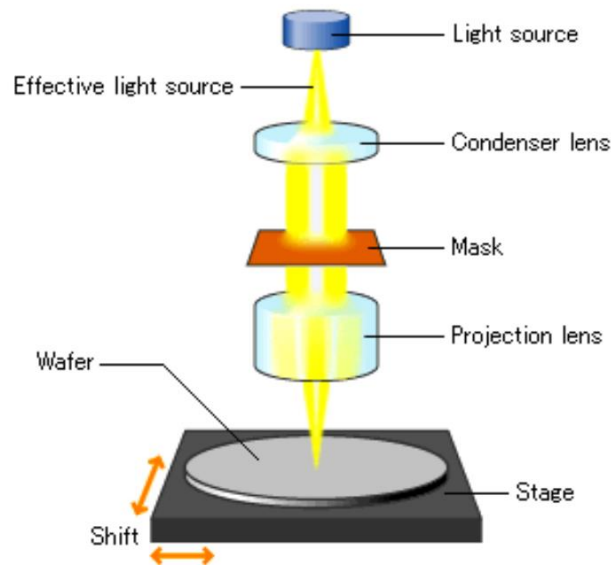


Figura 13 – Processo de gravação da máscara com o *layout*. Figura extraída de: (GOMES, 2014).

A Figura 14 mostra o passo-a-passo do processo de obtenção do *layout* gravado no *chip* sendo a camada superior o *photoresist*, a camada intermediária o óxido protetor e a camada inferior o *wafer* de silício.

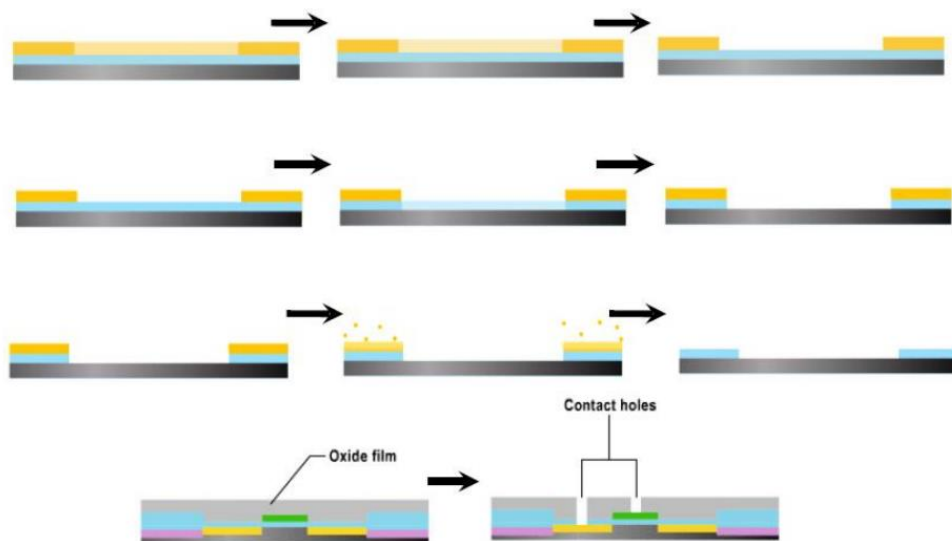


Figura 14 – Obtenção do *layout* gravado no silício. Figura extraída de: (GOMES, 2014).

Em seguida, a partir da obtenção do *wafer* coberto em algumas regiões com o óxido protetor, inicia-se o processo de dopagem das regiões desejadas, o que permite se obter transistores tipo n ou p. A Figura 15 ilustra como ocorre a obtenção do canal no *wafer*.

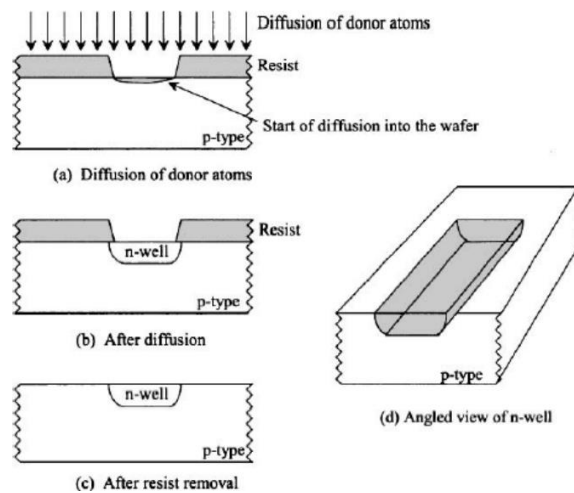


Figura 15 – Obtenção de um canal N no *wafer* tipo P. Figura extraída de: (BAKER, 2010).

Ao se obter o canal, pode-se decidir se o mesmo será utilizado dentro do circuito como uma resistência, pois é possível se mensurar a resistência que o canal irá representar no circuito (BAKER, 2010).

A demonstração de como a gravação é feita, é um ponto importante para se visualizar a etapa de confecção das máscaras, pois a gravação das máscaras é feita no *wafer* em várias etapas respeitando as camadas do *layout* do circuito integrado. Com isso, a próxima seção apresenta os métodos para confecção do *layout* do CI.

2.5 ETAPA DE *DESIGN* (PROJETO) DOS CIS

Nesta seção são apresentados os *softwares* utilizados na etapa de *design* e projeto dos circuitos integrados atualmente, bem como suas especificações e principais funcionalidades. É importante conhecê-los para que se saiba quais aspectos físicos devem ser abordados e conhecidos antes do início do projeto.

2.5.1 LINGUAGEM DE DESCRIÇÃO DE *HARDWARE*

Uma linguagem de descrição de *hardware* (HDL) permite que circuitos digitais sejam descritos por meio de sentenças, de tal forma que eles possam ser simulados e sintetizados. Um sistema descrito em HDL pode ser implementado tanto em um dispositivo programável FPGA (*Field Programmable Gate Array*) quanto em um dispositivo ASIC (*Application Specific Integrated Circuit*), sendo o processo deste último mais complexo. (CASILLO, 2010).

As linguagens de descrição de *hardware* se parecem muito às linguagens de programação, mas diferente destas são orientadas à descrição das estruturas e do comportamento do *hardware*. A descrição estrutural descreve a interconexão entre os componentes dos circuitos, e a descrição comportamental descreve o funcionamento destes componentes. Uma vantagem das HDLs em relação ao diagrama esquemático é que elas podem representar diretamente equações booleanas, tabelas verdade e operações complexas (MIDORIKAWA, 2007).

As linguagens de descrição de *hardware* mais utilizadas atualmente são o VHDL e o Verilog. Estas duas são linguagens padrões do IEEE (*Institute of Electrical and Electronic Engineers* - Instituto de Engenheiros Eletricistas e Eletrônicos), facilitando o desenvolvimento com diferentes sistemas (CASILLO, 2010).

As linguagens de descrição de *hardware* valem aqui como documentação, pois não foram utilizadas diretamente neste trabalho, elas são uma etapa crucial para produção de estruturas de circuitos integrados com grande densidade de transistores, visto que sua funcionalidade de síntese lógica reduz a quantidade de estruturas, economiza em espaço, em custos e mão de obra por permitir e reduzir redundâncias desnecessárias no projeto manual, porém não elimina a necessidade do projetista.

2.5.2 PLATAFORMA SPICE

O simulador utilizado para verificação da funcionalidade do *layout* projetado foi o LTSPICE IV (versão 4.231), *software* gratuito baseado na plataforma SPICE - *Simulation Program with Integrated Circuit Emphasis* - (Programa de Simulação com

Ênfase em Circuitos Integrados), que é um simulador de circuitos com propósito geral (utilizado em *Windows* ou *Linux* – neste último com restrições), que analisa circuitos em diversas configurações (corrente contínua, alternada, regimes transitórios, entre outros). Esta plataforma é muito empregada no projeto de circuitos tanto em nível macro (componentes discretos em eletrônica analógica) quanto em nível micro (circuitos integrados), pois devido a sua versatilidade, permite verificar a integridade de um circuito e prever o comportamento do mesmo (WARWICK, 2009).

Ao contrário de projetos realizados em nível de componentes discretos, os projetos de circuitos integrados devem ser simulados, e, além disso, suas propriedades e comportamentos devem ser estudados antes da fabricação. Pois, devido aos altos custos de máscaras de fotolitografia e outros pré-requisitos de fabricação tornam essencial a simulação do comportamento do circuito, antes que o circuito chegue a sua fase de produção, pois mesmo que o circuito tenha sucesso no funcionamento com componentes discretos, podem ocorrer falhas a nível micro de produção. Portanto, é necessário garantir a perfeição do *layout* antes dele ser construído. Devido a esta necessidade, existe um programa, que representa a maneira padrão de mercado de verificar parâmetros de operação do circuito a nível de transistor (micro ou nano) antes de acometer o mesmo à fabricação, este programa é o SPICE (NAGEL, 1973), (NAGEL, 1975), (STEER, 2007).

As falhas invisíveis a nível de projetos com componentes discretos, que podem vir a aparecer a nível micro são falhas devido à inserção de resistências ou capacitâncias parasitas a partir do processo de deposição de camadas, na construção de um circuito integrado. Estes componentes parasitas podem frequentemente ser estimados usando a simulação SPICE com mais precisão. Além disso, os projetistas podem obter mais informações sobre o circuito tanto a nível macro quanto micro. Por exemplo, o quanto o desempenho do circuito é afetado por tolerâncias de fabricação de componentes. Nesses casos, é comum o uso de SPICE para realizar simulações de Monte Carlo e se analisar o efeito de variações dos componentes sobre o desempenho do circuito, uma tarefa que é impraticável usando cálculos à mão para um circuito de qualquer complexidade apreciável.

Programas de simulação de circuitos de plataforma SPICE e derivados são os mais proeminentes, pois a partir de uma *netlist* textual (forma de programação SPICE

direcionada à descrição da disposição dos componentes no circuito) descrevendo os elementos do circuito (transistores, resistores, capacitores, etc.) e suas conexões, este gera as análises gráficas e numéricas necessárias para se avaliar o circuito (WARWICK, 2009). Desta forma, o SPICE foi utilizado, para gerar análises referentes aos circuitos desenvolvidos, a partir da *netlist* obtida com a compilação dos *layouts* gerados.

2.5.3 LASI – LAYOUT SYSTEM FOR INDIVIDUALS

LASI é um *software* de *layout* e *design* de circuitos integrados. É uma ferramenta versátil que pode ser utilizada para CIs, MEMS (*Micro Electro Mechanical Systems*), componentes discretos, esquemáticos, placas de circuito impresso e documentação de desenhos de projetos. Esta plataforma se destina principalmente ao uso educacional, pois sua licença é gratuita. Vale ressaltar que o LASI não substitui um *software* comercial.

Para se utilizar o LASI, não basta conhecer apenas o *software*, é necessário conhecer bem a tecnologia a qual será empregada na construção de seu dispositivo, saber sobre a física envolvida, e além disso ter uma boa noção de circuitos.

Este programa é principalmente composto por um editor de desenhos, onde são criados e organizados polígonos que correspondem às regiões que compõem os dispositivos em um CI, ele permite que os arquivos sejam exportados em várias extensões, para as mais variadas utilidades. E além do editor de desenhos, ele permite que uma *netlist* SPICE seja extraída tanto do esquemático quanto do *layout*. E também compila as *netlists* SPICE, porém não realiza simulações (LASI 7, 2017).

As principais funcionalidades e complementos fornecidos pelo LASI são a criação do *layout* em si (em plataforma gratuita), a possibilidade de, por meio de complementos externos, gerar material para simulações e análises de erro que são os casos dos complementos DRC (*Design Rules Check*) e do CKT (*Circuit*) que serão descritos nas etapas de fluxo de projeto.

O LASI (versão 7.0.90) foi utilizado então na construção do *layout* do contador proposto, por isso, a próxima seção introduz aspectos importantes da estrutura do mesmo.

2.6 O CONTADOR

O contador é uma combinação lógica de portas que implementam as funções lógicas conhecidas para desempenhar sua função. Ele utiliza basicamente de portas NAND (existem outras formas de implementá-lo), que são dispostas para criar um dispositivo chamado *flip-flop*, que é um registrador, que funciona mantendo ou alterando seu sinal de saída anterior de acordo com os sinais em suas entradas de dados no momento em que um pulso que o habilita é aplicado em sua entrada. O contador é uma configuração de *flip-flops* em cascata, ou seja, registradores, que juntamente com uma lógica digital em suas entradas de dados implementa a contagem binária (TOCCI, 2003).

São várias as aplicações de um contador em eletrônica digital na atualidade, sendo elas em microprocessadores, memórias (como dispositivos internos e também nos circuitos que fazem a seleção do bloco de memória a ser utilizado), microcontroladores, como contadores de pulsos (dependendo da configuração do usuário), como contadores de tempo (nos *timers* internos dos microcontroladores) entre outras (TOCCI, 2003).

Existem dois tipos de contadores, os assíncronos e os síncronos. Os assíncronos são utilizados quando não se necessita de grande precisão no tempo da contagem e a única preocupação é com o valor da contagem, pois estes carregam atrasos ocasionados por diversos fatores e um deles é que o sinal de *clock* (comando de atuação) é aplicado somente ao primeiro *flip-flop* do conjunto. Já os síncronos são mais precisos com os tempos e também com a contagem, pois todos os *flip-flops* recebem o *clock* ao mesmo tempo (TOCCI, 2003). Os detalhes relacionados à sua construção são mostrados no próximo capítulo.

O fato dos componentes citados serem de uso constante no cotidiano do ser humano atualmente, mostra que a escolha pelo desenvolvimento do contador se dá por este ser uma estrutura de uso disseminado e fundamental para o funcionamento das mais diversas aplicações dentro dos complexos circuito que envolvem os dispositivos modernos.

De posse dos conceitos apresentados neste capítulo, no próximo, são apresentados os passos desenvolvidos na metodologia deste trabalho para o desenvolvimento do *layout* do contador síncrono. Desta forma apresenta-se desde as bases das simulações pré-projeto a um fluxograma que representa o fluxo de projeto de um circuito integrado no geral.

3. METODOLOGIA

Neste capítulo é apresentada a metodologia das atividades desenvolvidas, tanto na etapa CAD do projeto, quanto nas etapas de simulação. É mostrado também o fluxo de um projeto de *layout* de circuitos integrados, considerando as etapas que devem ser seguidas para obtenção de uma estrutura funcional ao fim do processo. O fluxograma com as etapas do projeto é mostrado na Figura 16.

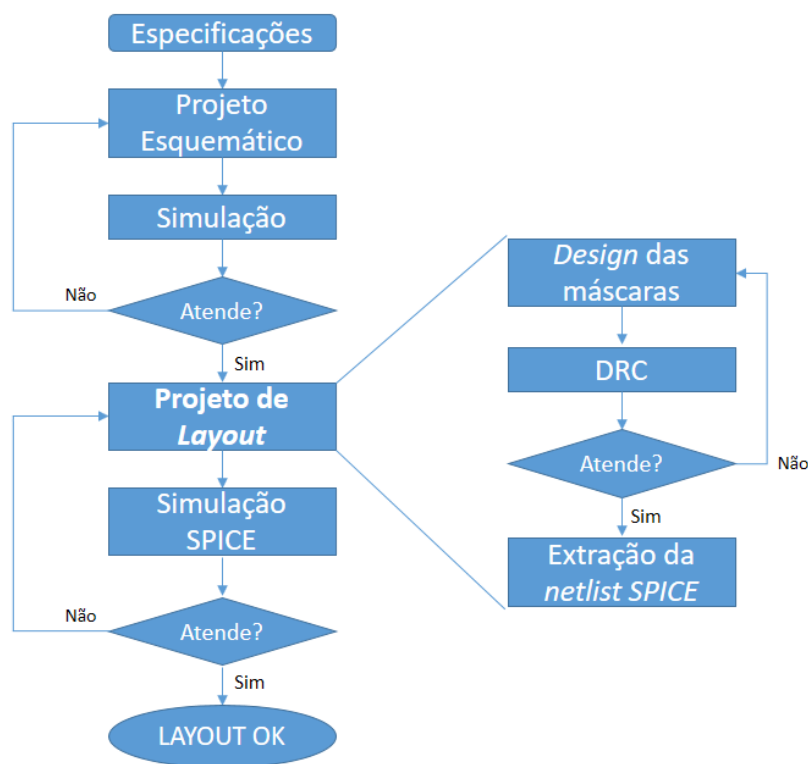


Figura 16 – Fluxograma do projeto de um CI (Fonte: Autor).

Conforme apresentado na Figura 16, o projeto se inicia com o recebimento das especificações do circuito a ser desenvolvido (funcionalidade, tecnologia a ser empregada, dimensões, tempos a serem respeitados, entre outros). A partir das especificações realiza-se a primeira montagem em simulador de circuitos, no caso o LTSPICE IV, onde é desenvolvido o esquemático, para verificar em simulação se o circuito cumpre os requisitos de funcionalidade propostos (dentro da tecnologia considerada pelo projetista). Caso o circuito não atenda aos pré-requisitos, são

realizados ajustes no esquemático até que o mesmo passe a atender os mesmos. A partir deste momento, inicia-se o projeto do *layout*, sendo que a primeira parte desta etapa consiste em desenhar, em *software* CAD, no caso o LASI, as trilhas das máscaras (dentro das variadas camadas que compõem um circuito integrado), respeitando as regras de projeto, e identificando cada uma das trilhas de acordo com os nós do circuito. Em seguida é feita a verificação DRC, que verifica se o *design* das máscaras viola alguma regra de projeto, pois mesmo estando de posse das regras no momento do projeto, alguma pode passar despercebida (por serem muitas), principalmente se for um projeto de maior complexidade. Caso a verificação DRC encontre algum erro, é necessário voltar ao *design* e corrigi-lo até que não haja mais erros de projeto, quando esta verificação ocasionar em retorno positivo, o circuito está pronto para que se realize a extração da *netlist* (código de programação, que pode ser compilado em plataforma SPICE, que define as características do circuito a nível micro), que possibilita a análise de funcionamento do circuito e a detecção de possíveis erros que podem estar contidos no mesmo. A partir da *netlist*, realiza-se então a simulação no LTSPICE IV, obtendo dados de respostas do circuito aos impulsos de entrada, tempos de resposta, características de funcionamento do mesmo, ruídos e possíveis distorções no sinal. Caso a simulação seja satisfatória o *layout* é considerado utilizável, caso contrário, deve se retornar à etapa do projeto de *layout*, realizar ajustes no *design* das máscaras e realizar as verificações pertinentes para validar o *layout*.

Basicamente o fluxo de projeto de um CI segue os passos descritos, porém cada uma das etapas exige grande concentração e muito conhecimento por parte do projetista.

Na próxima seção são mostrados os passos para desenvolvimento dos *layouts*.

3.1 DESENVOLVIMENTO DOS LAYOUTS

Após entender o funcionamento dos transistores MOS, é necessário saber como se dá a implementação de portas e funções lógicas a partir de transistores, para que se possibilite o início da etapa prática em si. Utilizou-se então o *software* LTSPICE

IV para simulação do funcionamento da função lógica, a fim de verificar que a função em desenvolvimento realmente atende aos pré-requisitos de respostas desejadas em função dos níveis lógicos aplicados. Os *layouts* desenvolvidos inicialmente foram dos transistores NMOS e PMOS (unidades básicas para construção das funções lógicas), em seguida foram desenvolvidas as portas referentes às funções NOT (inversor), NAND (não e) de duas entradas, NAND (não-e) de três entradas e a partir destas, montou-se o *flip-Flop* JK (registrador) e a montagem final do contador de quatro *bits*. Mas antes da montagem do *layout* em si, foi necessário que se buscasse em bibliografias relevantes as formas de implementação das funções descritas dentro da tecnologia CMOS (tecnologia explorada neste trabalho) as quais foram todas simuladas em funcionamento no *software* LTSPICE IV (BAKER, 2010) (KANG, 2003).

Conforme visto anteriormente, as formas de implementação das funções lógicas foram estudadas e são definidas em (REIS, 2008). Os símbolos eletrônicos destas funções lógicas, bem como sua montagem interna (a nível de transistor) serão demonstrados em figuras. O símbolo é utilizado em eletrônica digital para se reduzir os desenhos e levar ao entendimento apenas da função lógica desejada, e não do que ocorre por trás do circuito. O primeiro circuito a ser desenvolvido é o inversor (NOT), o qual possui seu símbolo lógico mostrado na Figura 17 (feito no LTSPICE IV).

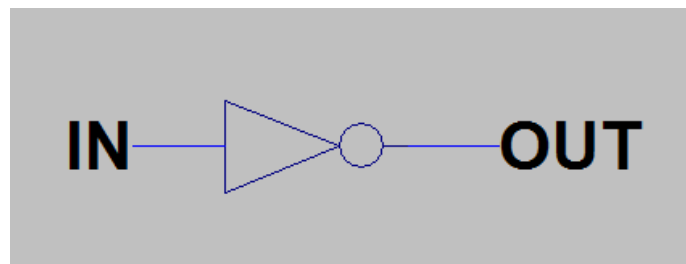


Figura 17 – Símbolo da porta inversora (Fonte: Autor).

Além da simbologia, vale demonstrar o circuito CMOS por trás da função lógica, pois uma vez entendida a função lógica ela pode ser demonstrada apenas por sua simbologia digital. A Figura 18 mostra o circuito lógico desenvolvido no LTSPICE IV que realiza a função NOT.

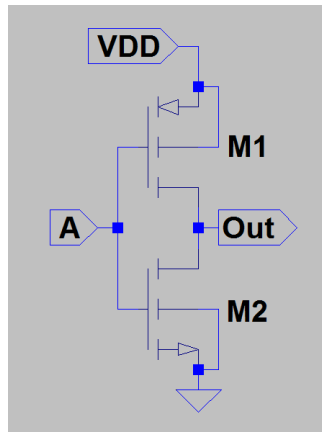


Figura 18 – Circuito inversor CMOS (Fonte: Autor).

Como pode ser visto na Figura 18, o inversor é composto por duas redes de transistores MOSFETs, uma *pull-up* (MOSFET PMOS) para gerar a saída em nível alto e outra *pull-down* (MOSFET NMOS) para gerar a saída em nível baixo. Ao se aplicar um sinal na entrada A, caso o nível do sinal seja alto, se ativa o NMOS e gera saída em nível baixo, caso o nível seja baixo se ativa o PMOS e gera saída nível alto. O funcionamento do circuito acima pode ser descrito pela Tabela 1 (verificado em simulação).

Tabela 1 – Tabela verdade do bloco Inversor (Fonte: Autor)

A	Out
0	1
1	0

De posse desses dados é possível iniciar a confecção do *layout* do bloco inversor, porém é importante simular todos os blocos a serem desenvolvidos antes de iniciar o projeto no LASI.

O segundo bloco desenvolvido no LTSPICE IV foi a porta NAND de duas entradas, cujo símbolo eletrônico é mostrado na Figura 19.

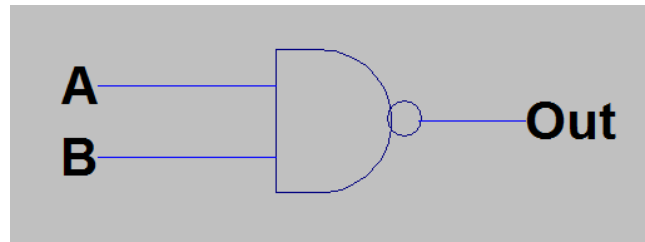


Figura 19 – Simbologia NAND (Fonte: Autor).

A função NAND também possui seu circuito interno composto por duas redes de transistores, porém com quatro destes sendo dois PMOS e dois NMOS. O esquemático é mostrado na Figura 20.

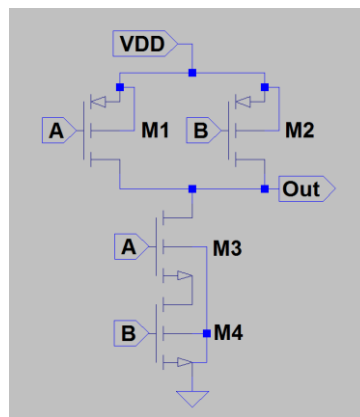


Figura 20 – Circuito esquemático SPICE da função NAND (Fonte: Autor).

A descrição do funcionamento do circuito se dá de acordo com o que mostra a Tabela 2.

Tabela 2 – Tabela verdade da função NAND de duas entradas (Fonte: Autor).

A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

Desenvolveu-se então a porta NAND com três entradas, que utiliza a mesma abordagem e simbologia da de duas entradas, porém com uma entrada a mais, ou seja, ela também é composta pelas redes *pull-up* e *pull-down*, porém com um transistor PMOS a mais em paralelo na *pull-up* e um NMOS a mais em série na *pull-down*. A Figura 21 mostra o esquemático da NAND de três entradas.

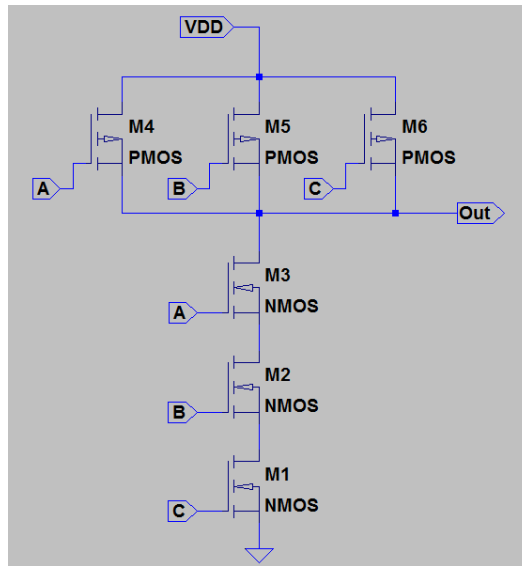


Figura 21 – Esquemático SPICE da NAND de três entradas (Fonte: Autor).

A Figura 22 mostra o símbolo utilizado para demonstrar a NAND de três entradas em um circuito lógico.

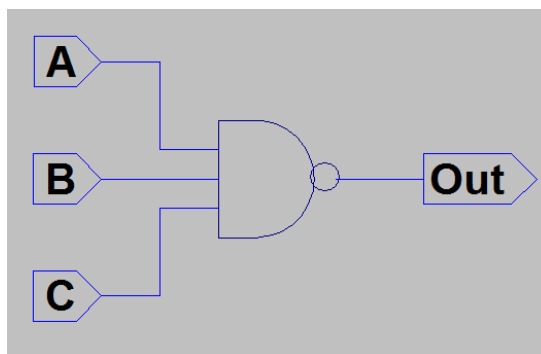


Figura 22 – Simbologia NAND de três entradas (Fonte: Autor).

Seguindo a mesma abordagem, a tabela verdade que representa o funcionamento do circuito da Figura 21, só apresenta nível lógico baixo na saída quando todas as entradas foram nível alto. A Tabela 3 demonstra a resposta da NAND de três entradas.

Tabela 3 – Tabela verdade da NAND de três entradas (Fonte: Autor).

A	B	C	Out
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Após a implementação em SPICE das funções lógicas, iniciou-se o desenvolvimento do *flip-flop* JK (unidade principal de um contador) que utiliza as funções lógicas construídas para ser implementado, sendo elas duas portas NAND de três entradas e duas portas NAND de duas entradas em uma configuração definida como *latch* (TOCCI, 2003).

A Figura 23 mostra o circuito lógico digital de um *flip-flop* JK construída no LTSPICE IV, bem como sua simbologia em lógica digital.

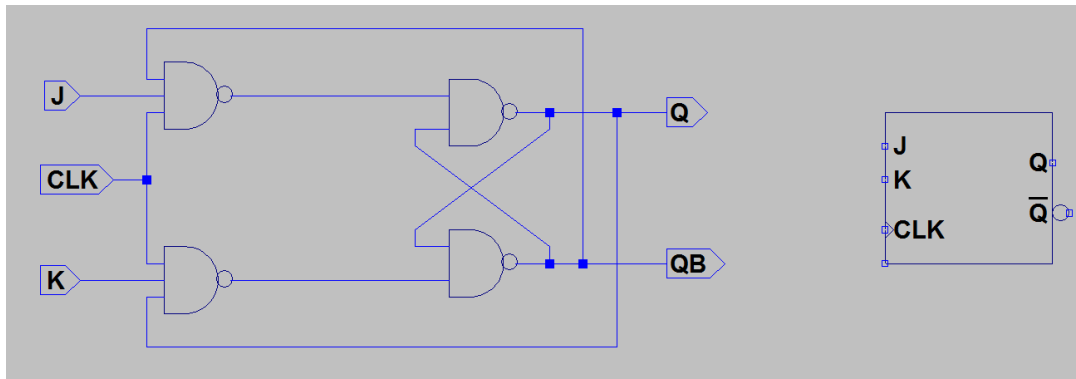


Figura 23 – Circuito lógico e simbologia do *flip-flop* JK (Fonte: Autor).

O flip-flop JK foi utilizado com suas entradas J e K interligadas diretamente uma à outra, formando assim um *flip-flop* tipo T (*Toggle*), pois nesta configuração este se torna mais útil para a construção de um contador (comuta a saída sempre que o nível lógico alto é aplicado nas entradas JK, quando do próximo pulso de *clock*). Portanto a Tabela 4 mostra o funcionamento do circuito quando em simulação.

Tabela 4 – Tabela verdade do flip-flop tipo T (Fonte: Autor).

Q	JK (T)	Q (CLK+1)
0	0	0
0	1	1
1	0	1
1	1	0

Após verificação de funcionamento do *flip-flop*, desenvolveu-se por fim o contador síncrono de quatro *bits*. Que é construído utilizando-se quatro unidades do *flip-flop* JK, uma porta NAND de duas entradas, uma porta NAND de três entradas e duas portas inversoras (TOCCI, 2003). O circuito lógico de um contador é apresentado na Figura 24.

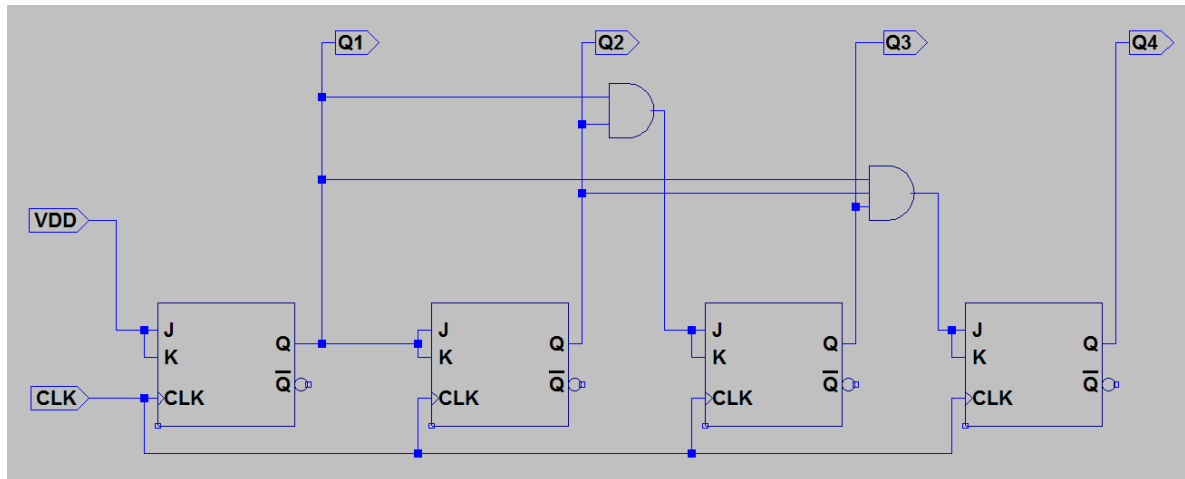


Figura 24 – Esquemático de um contador síncrono de 4 bits (Fonte: Autor).

Conforme dito, verifica-se que as entradas J e K ficam interligadas para implementação do flip-flop tipo T e as saídas de Q1 a Q4 representam as saídas que realizam a contagem binária no contador, sendo Q4 o *bit* mais significativo. O circuito funciona basicamente com o primeiro *flip-flop* comutando em todos os pulsos de *clock*, o segundo com metade desta frequência, o terceiro com metade da frequência do segundo e o quarto com metade da frequência do terceiro, gerando assim uma relação de contagem binária de quatro *bits* através da divisão das frequências anteriores. A Figura 25 demonstra como ocorre o funcionamento do contador a cada pulso de *clock*.

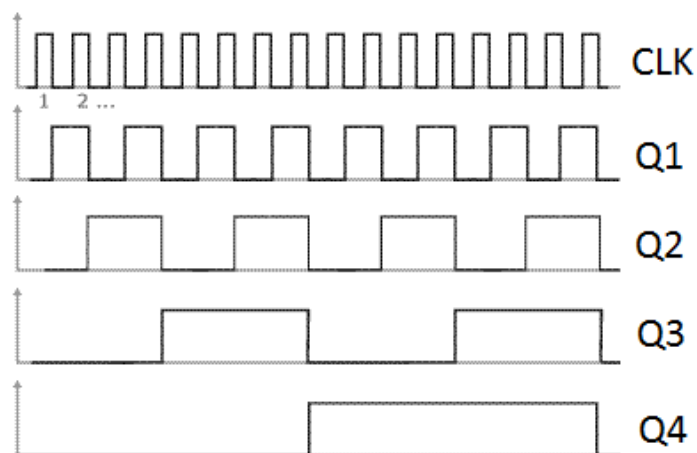


Figura 25 – Contagem binária do contador síncrono (TOCCI, 2003 - adaptada).

A partir desta etapa, obteve-se a base necessária para a confecção dos *layouts* finais das estruturas propostas, que serão apresentados como resultados, uma vez que estes são os produtos finais gerados neste trabalho. Nesta parte, utilizou-se o *software* LASI para desenvolver o *design* das máscaras que são necessárias para o desenvolvimento de várias funções lógicas descritas. Tais máscaras se apresentam no *layout* em conjunto (formando o dispositivo), mas no momento da gravação no *chip* (fabricação em si), elas são tratadas de maneira isolada.

Após a confecção dos *layouts* é necessário realizar simulações que verificam os mesmos. Na próxima seção é apresentada a metodologia para a realização destas simulações.

3.2 VERIFICAÇÃO DRC

Ao finalizar os *layouts*, é necessário garantir que nenhuma regra de projeto tenha sido violada, e esta etapa é realizada através da simulação da checagem de regras de projeto (*Design Rules Checking*), que é um utilitário do LASI, e deve ser realizada sempre ao final dos *layouts*. Para se verificar os erros de projeto, é necessário saber qual a tecnologia empregada, e obter o arquivo contendo as regras de projeto referentes à cada camada utilizada no projeto. No caso utilizou-se o conjunto conhecido pelo nome *MOSIS RULES* disponível gratuitamente em (CMOSEDU, 2017).

Rodando este complemento, o *layout* desenvolvido é mapeado, e cada uma das regras de projeto é verificada, caso haja alguma violação, o complemento gera um *log* de erros, com o mapa de localização do erro. Além disso, é informada qual regra foi violada, para que o usuário possa corrigir e realizar a verificação novamente até que não haja mais erros. Esta etapa pode ser um pouco demorada, o que depende da máquina utilizada no projeto (capacidade de processamento) e também da complexidade do circuito integrado a ser desenvolvido.

Dentro das *Design Rules*, existem limitações de tamanhos e espaçamentos entre elementos de diferentes *layers* (e até dentro de um mesmo *layer*). E na utilização

de cada camada no LASI se faz necessário consultá-las. Porém são tantas as regras, que se torna necessário a realização de uma simulação que verifique se todas foram atendidas, de forma a garantir que o projeto seja funcional. E como esta verificação deve ser feita após a confecção dos *layouts*, este se torna o passo anterior à etapa de simulações, que garante funcionalidade do projeto dentro da tecnologia empregada.

O conjunto de regras utilizado (dentro dos que foram pertinentes ao projeto), numerado de acordo com o DRC e as *MOSIS RULES* pode ser visualizado na Figura 26.

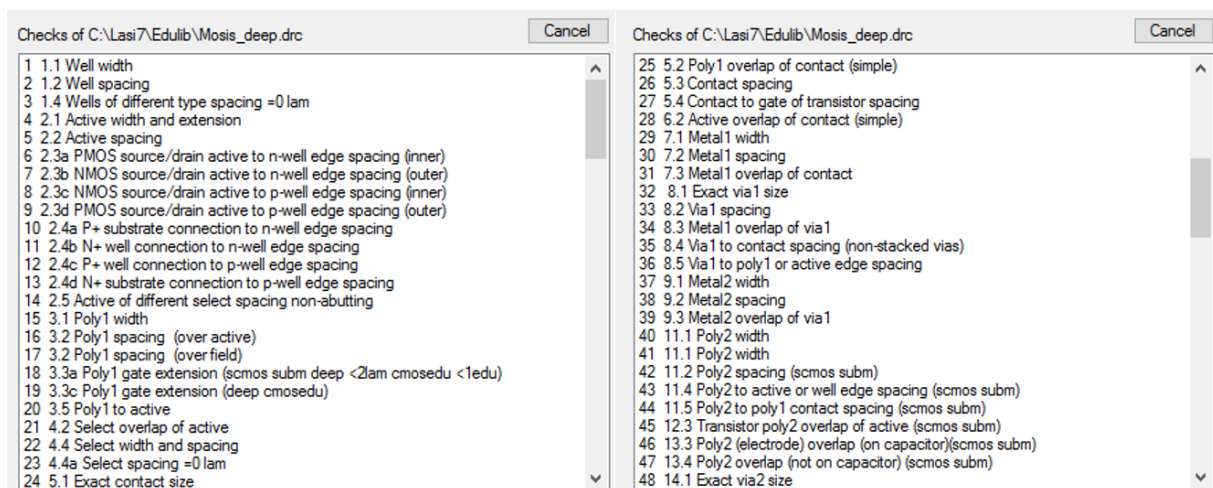


Figura 26 – Regras de projeto de 1 a 48 (Fonte: Autor).

Conforme visto na Figura 26, são várias as regras de projeto (117 no total), mas cabe ao projetista saber quais são realmente pertinentes ao seu projeto, para resumir o tempo computacional para simulação DRC, uma vez que esta simulação é uma parte que exige bastante do computador. Com isso, no projeto das portas lógicas e estruturas desenvolvidas utilizou-se as regras de 1 a 31. A tela de configuração do complemento DRC pode ser vista na Figura 27 e é a mesma para todos os *layouts*.

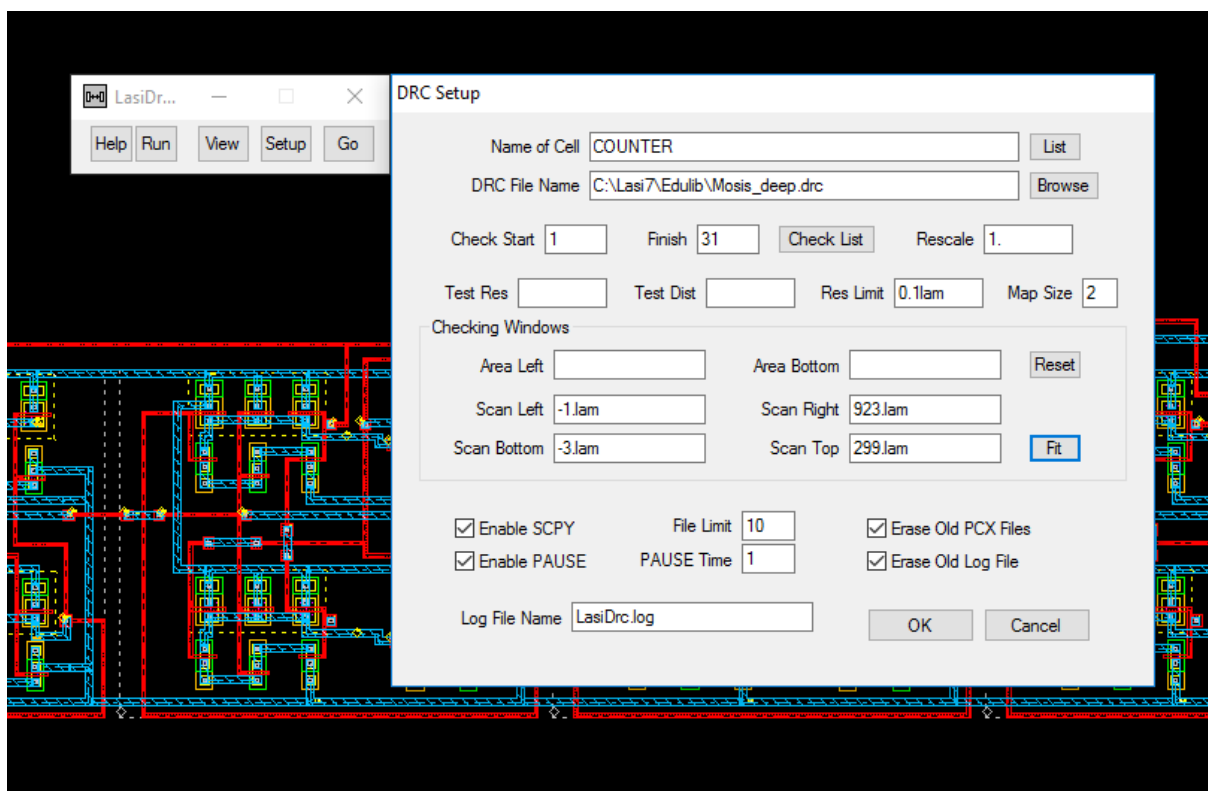


Figura 27 – Tela de configuração do DRC (Fonte: Autor).

Além das regras listadas, o arquivo contém um arquivo CAD, para que o projetista possa conferir visualmente a que distância corresponde uma determinada regra de projeto, sendo que neste arquivo, as distâncias referentes à cada regra em cada *layer* são definidas por exemplos práticos.

Após a execução do DRC, há outro complemento, chamado LASI CKT, que deve ser empregado no processo, este extrai os parâmetros do circuito para que se realize a simulação a nível micro, este último complemento, juntamente com o DRC realizam as simulações pós-*layout*. A próxima seção mostra a metodologia para uso do mesmo.

3.3 EXTRAÇÃO DA *NETLIST* (LASI CKT)

Apesar de garantir a funcionalidade do circuito, o DRC não permite a visualização do funcionamento do mesmo, portanto o utilitário do LASI que permite a

simulação do circuito desenvolvido se chama LASI CKT (LASI *circuit*). O mesmo permite extrair a *netlist* SPICE a partir do *layout* ou do esquemático desenvolvido, e a partir desta e do *software* baseado em plataforma SPICE (no caso o LTSPICE IV), é possível realizar a simulação e análise do circuito para verificar o comportamento físico do mesmo em momentos de tensões (níveis lógicos) aplicadas e de circulação de correntes no mesmo. Esta aplicação permite verificar se o circuito, após finalizado, atende aos requisitos de tempos de resposta, perdas por dissipação de potência, sensibilidade à temperatura, quedas de tensão e conformidade da onda de saída.

Este é um complemento de complexidade extremamente elevada, visto que para operá-lo é necessário saber como o próprio complemento realiza a compilação do circuito desenvolvido para gerar a *netlist* e as vezes são necessários vários ajustes no circuito para que o mesmo funcione corretamente, como nomear e indicar corretamente o tipo de cada nó, pois na hora de compilar podem ocorrer vários tipos de erros ocasionados pela rotulação incorreta dos nós. Portanto, além de conhecer bem a tecnologia é importante também conhecer a fundo o *software*.

É necessário também fornecer a este complemento dois arquivos que são fundamentais para que a *netlist* gerada apresente resultados. São eles o *header* e o *footer*. O *header* é o arquivo que informa quais estímulos de entrada serão aplicados aos pontos do circuito em questão e o *footer* é o arquivo que informa quais os modelos dos dispositivos utilizados no circuito. A tela de configuração do complemento CKT pode ser visualizada na Figura 28.

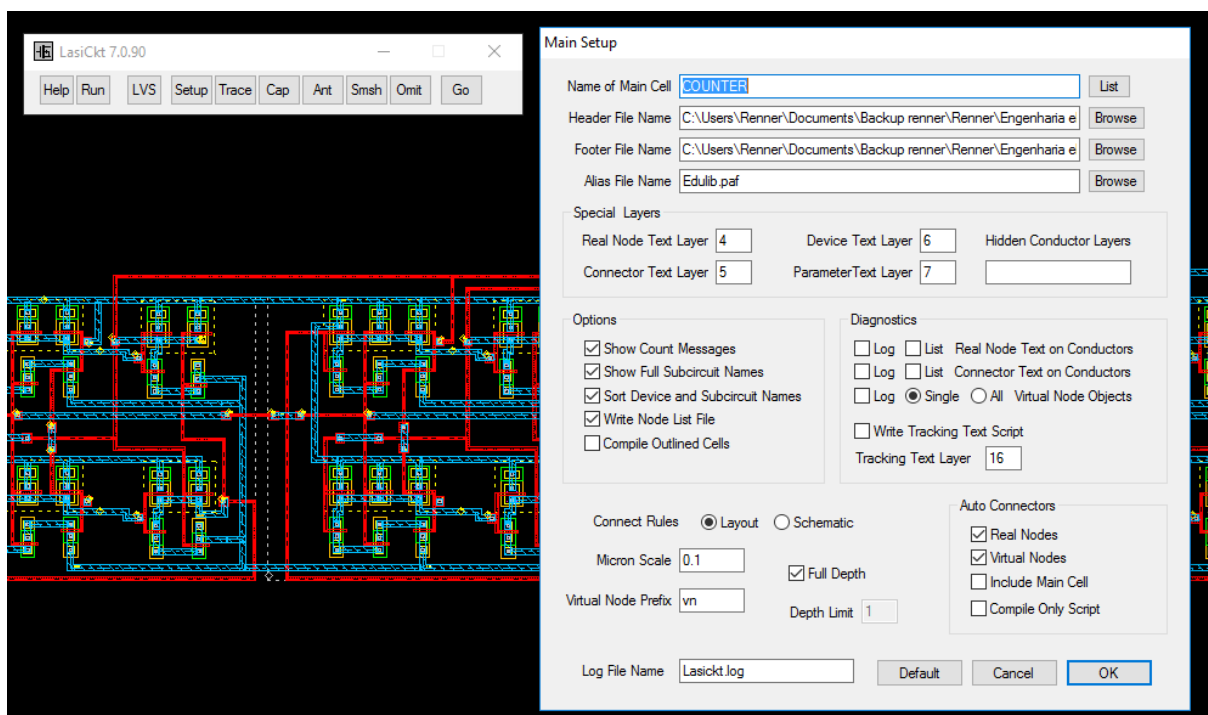


Figura 28 – Tela de configuração do LASI CKT (Fonte: Autor).

A extração da *netlist* é a última etapa realizada diretamente no LASI, mas esta etapa deve ser feita e refeita, após as devidas correções, até que o complemento não gere erros ou *warnings*, pois similar ao que ocorre no DRC erros na geração da *netlist* podem gerar circuitos incorretos invalidando assim as simulações pós-*layout*. Ao fim da simulação o complemento salva na pasta de *layouts* do LASI um arquivo com extensão que pode ser lida e simulada por um *software* de plataforma SPICE. Após geração do arquivo, as simulações e análises foram feitas no LTSPICE IV, mas não por meio do esquemático conforme ocorreu nas simulações pré-*layout*, mas através da programação (*netlist*) SPICE gerada pelo LASI.

A partir da metodologia apresentada, gerou-se a biblioteca com os *layouts* das células base, para então gerar o *layout* do contador proposto. Os resultados obtidos a partir do desenvolvimento do *design* dos *layouts*, bem como as simulações são apresentados no próximo capítulo.

4. RESULTADOS E DISCUSSÕES

Neste capítulo são apresentados os resultados obtidos pela aplicação da metodologia desenvolvida, bem como as discussões pertinentes aos mesmos.

4.1 DESENVOLVIMENTO DOS LAYOUTS

Ao finalizar a etapa de simulação do esquemático, as atividades foram voltadas para o projeto dos *layouts*, onde iniciou-se o emprego do *software* educacional LASI, que é empregado na área de projetos de circuitos integrados.

Existem duas tecnologias possíveis para se desenvolver um projeto de *layout* com o *software* utilizado, uma chamada *standard cell*, onde se desenvolve células padrão que são utilizadas nas construções subsequentes (agilizando o processo, porém, deixando o projetista preso às células criadas), e outra chamada *full custom*, onde todo o projeto é desenvolvido de maneira customizada, reduzindo-se a área gasta, porém perdendo em agilidade uma vez que todo o circuito é desenvolvido de maneira personalizada. Cada uma possui suas vantagens e desvantagens, e cabe ao projetista verificar qual delas é mais vantajosa para o projeto em questão. Adotou-se a abordagem *standard cell* (células padrões), devido ao maior suporte oferecido pela ferramenta utilizada para esta tecnologia.

Utilizou-se então o LASI, para o desenvolvimento dos *layouts*, pois o LASI é a ferramenta CAD específica para projeto de circuitos integrados, uma vez que suas camadas correspondem às camadas de materiais utilizados no processo de produção dos circuitos integrados. Este último programa também permite a criação de células, numa estrutura de hierarquia (*ranks*), nas quais as células menores podem ser utilizadas dentro do projeto das subsequentes.

A primeira célula desenvolvida foi a do transistor NMOS, que é apresentada na próxima subseção.

4.1.1 TRANSISTOR NMOS

As primeiras implementações foram os transistores básicos, começando pelo transistor NMOS. A Figura 29 mostra a célula do transistor NMOS projetado.

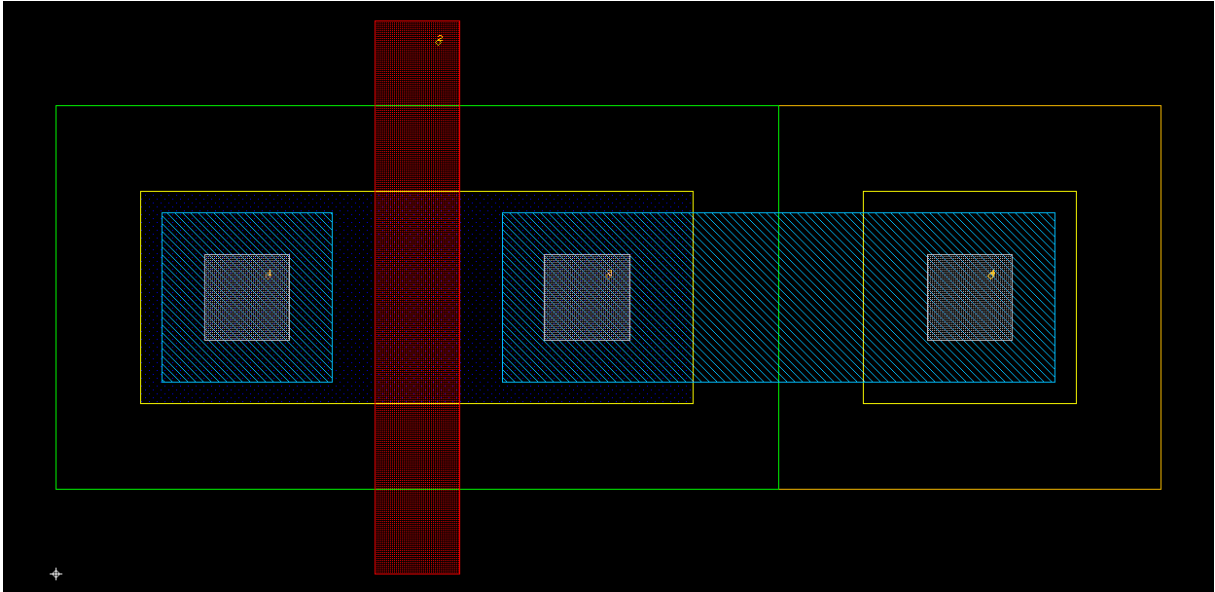


Figura 29 – NMOS desenvolvido em LASI. (Fonte: Autor).

A Figura 29 mostra o *layout* do NMOS desenvolvido no LASI, com os quatro terminais em destaque (pontos amarelos), onde ressalta-se a presença do substrato (bulk), que representa o quarto terminal do transistor. As camadas internas (em amarelo) representam as regiões ativas do dispositivo e as externas representam as camadas seletoras (que definem o tipo – p ou n – da região ativa em questão), a camada em vermelho representa a deposição do poli-silício referente ao terminal de *gate* (que cria o canal quando em ação de campo elétrico).

A etapa de criação dos transistores é uma parte que exige grande carga teórica do projetista, pois além de ter conhecimento do funcionamento do dispositivo em construção, este deve conhecer também a forma de construção de um dispositivo dentro da tecnologia empregada e além disso, deve ficar atento às regras de projeto. Cada polígono criado em CAD representa uma região do dispositivo final, que pode ocasionar em mau funcionamento do dispositivo final no caso de não conhecimento

ou desrespeito de algum detalhe técnico referente à construção ou mesmo à tecnologia.

O NMOS foi desenvolvido com um L (*Length* - Comprimento) de 1,6 μm e W (*Width* - Largura) de 8,0 μm afim de atender os requisitos de velocidade de resposta, pois a proposta inicial era de que o W fosse de 5,0 μm , porém o dispositivo não respondeu bem na frequência de *clock* aplicada.

Após o desenvolvimento do *layout* realizou-se a simulação DRC e correções de *design* até que o complemento não retornasse nenhum *log* de erro de acordo com as regras utilizadas, conforme mostra a Figura 30.

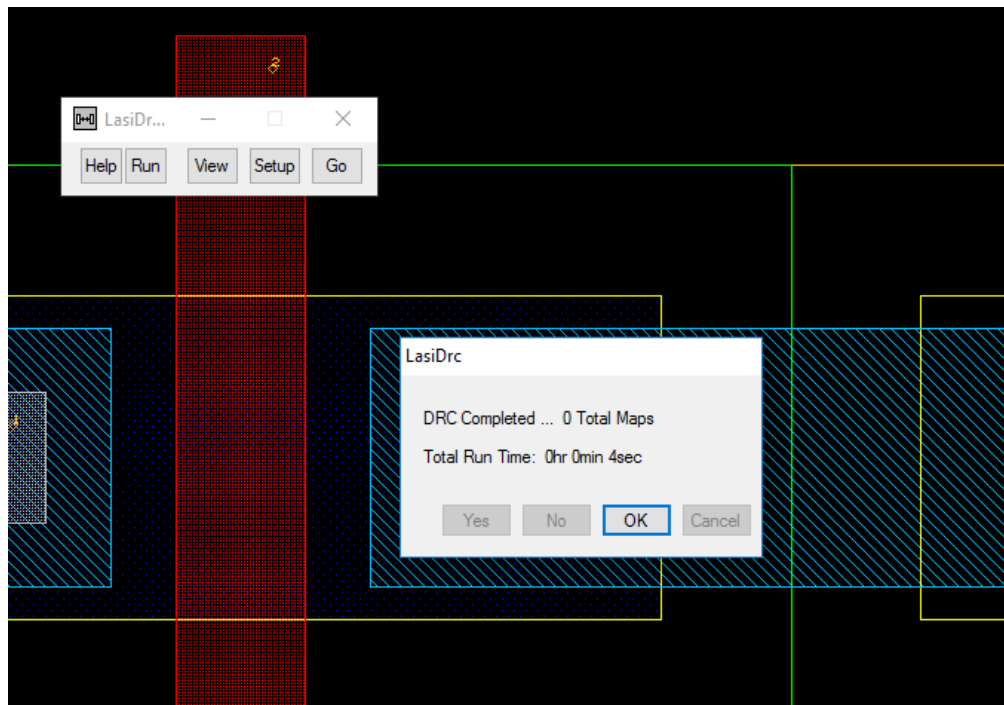


Figura 30 – Tela final do DRC (Fonte: Autor).

Além da quantidade de erros encontrados, o programa informa o tempo gasto para rodar o complemento DRC. Verifica-se que para o transistor NMOS o tempo gasto foi de quatro segundos, porém este tempo aumenta de acordo com o aumento da complexidade do circuito desenvolvido, o que poderá ser visto na tela final do DRC das estruturas maiores. Como a tela é praticamente igual para todas as estruturas (o que muda é só o tempo), optou-se por apresentar as telas do NMOS e do contador

(primeiro e último dos *layouts* desenvolvidos), afim de mostrar a diferença de tempo gasto com o aumento da área útil e da quantidade de *layers* do circuito desenvolvido.

Ao finalizar o projeto do NMOS, iniciou-se o projeto do PMOS, para criação das duas células base da tecnologia CMOS. O projeto PMOS é mostrado na próxima subseção.

4.1.2 TRANSISTOR PMOS

A segunda implementação foi o transistor PMOS, pois estes dois primeiros são a base para a construção das funções propostas. A Figura 31 mostra a célula do transistor PMOS projetado.

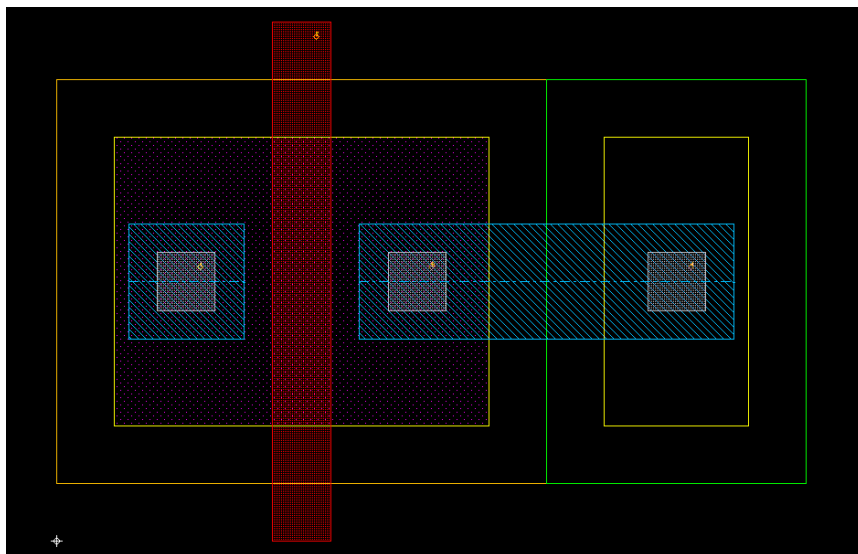


Figura 31 – PMOS desenvolvido em LASI. (Fonte: Autor).

A Figura 31 mostra o *layout* do PMOS desenvolvido no LASI, com os quatro terminais em destaque, similar ao NMOS. As camadas internas (em amarelo) representam as regiões ativas do dispositivo e as externas representam as camadas seletoras (que definem o tipo – P ou N – da região ativa em questão), a camada em vermelho representa a deposição do poli-silício referente ao terminal de *gate* (que cria o canal quando em ação de campo elétrico).

Ambos os dispositivos (NMOS e PMOS) foram utilizados como unidades básicas (*standard cells*) para a criação de outras funções lógicas que serão descritas. O PMOS foi desenvolvido com um L de 1,6 μm e W de 16,0 μm afim de atender os requisitos de velocidade de resposta, pois a proposta inicial era de que o W fosse de 10,0 μm , porém o dispositivo não respondeu bem na frequência de *clock* aplicada.

Um dos erros cometidos no projeto ocorreu nesta etapa, onde ignorou-se uma especificidade da tecnologia na qual o W do PMOS deve ser o dobro do W do NMOS, pois a velocidade dos portadores de um e outro são diferentes e influenciam na resposta do circuito. O erro ocorreu devido à grande quantidade de detalhes que devem ser considerados, com isso, o projeto inicial foi realizado utilizando-se o mesmo W para ambos os transistores, o que gerou mau funcionamento no dispositivo final, acarretando a necessidade de recomeçar o projeto para correção do erro. Ao recomeçar, consultou-se a literatura eliminando assim as causas do erro.

Após o projeto dos dispositivos base da tecnologia CMOS, inicia-se o projeto das funções lógicas. A próxima subseção apresenta o projeto da função NOT, que é uma das células base utilizadas no contador.

4.1.3 PORTA NOT (INVERSORA)

Após o desenvolvimento dos transistores, iniciou-se a implementação das funções lógicas mais simples (que serão utilizadas no contador), onde a primeira desenvolvida foi a NOT (inversora).

Conforme citado anteriormente, o SPICE permitiu realizar as simulações de verificação de funcionamento dos circuitos em questão antes do desenvolvimento do *layout* em si, sendo uma ferramenta extremamente usual no desenvolvimento das funções lógicas. Após a verificação de funcionamento pela aplicação de níveis lógicos no circuito, iniciou-se a confecção do *layout*. A Figura 32 mostra o *layout* desenvolvido no LASI para porta inversora (função lógica NOT).

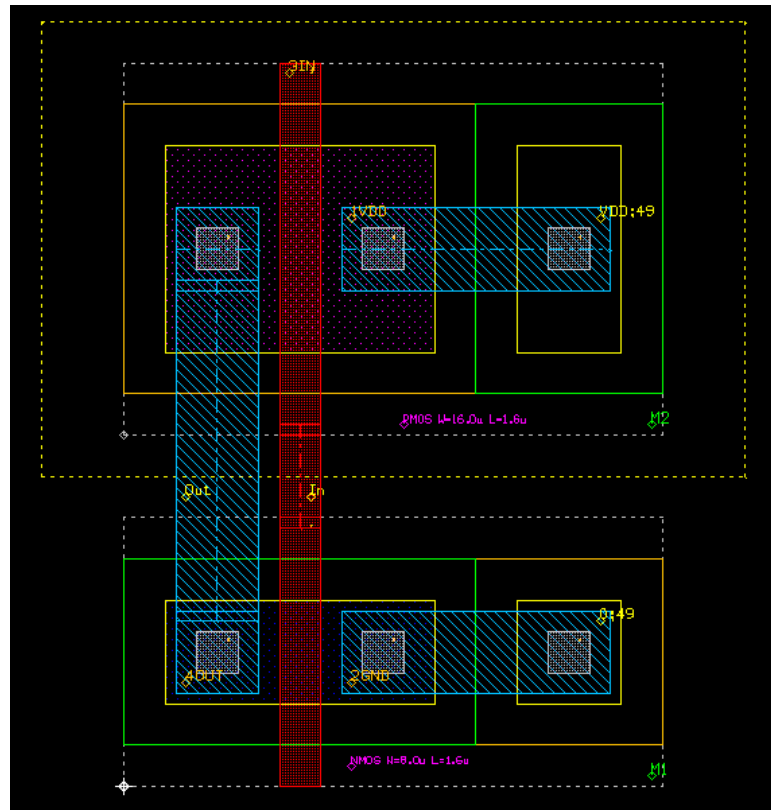


Figura 32 - Inversor desenvolvido no LASI (Fonte: Autor).

No *layout* do inversor da Figura 32, é possível verificar as junções de um transistor PMOS e um NMOS para implementação da função NOT, os terminais são interligados com uma camada de metal (em azul), o *gate* com poli-silício e os terminais de entrada e saída do dispositivo são indicados por textos no desenho. É possível ressaltar também a presença de uma camada extra envolvendo o transistor PMOS (camada *N-well* – camada amarelo pontilhado), esta camada é necessária uma vez que como o substrato consiste no *chip* em si (material geralmente tipo P), este permite a criação de transistores NMOS sem maiores problemas (uma vez que o substrato NMOS é tipo P), já para criação do PMOS é necessário que o substrato se apresente do tipo N, o que é feito com a camada *N-well* (que representa um poço do tipo N).

A próxima seção mostra o projeto da função NAND de duas entradas, que é outra célula base utilizada para construção do *layout* do contador.

4.1.4 PORTA NAND (NÃO-E) DE DUAS ENTRADAS

Após o desenvolvimento do inversor, implementou-se a porta NAND. A Figura 33 mostra o *layout* desenvolvido no LASI para a porta não-e (função lógica NAND).

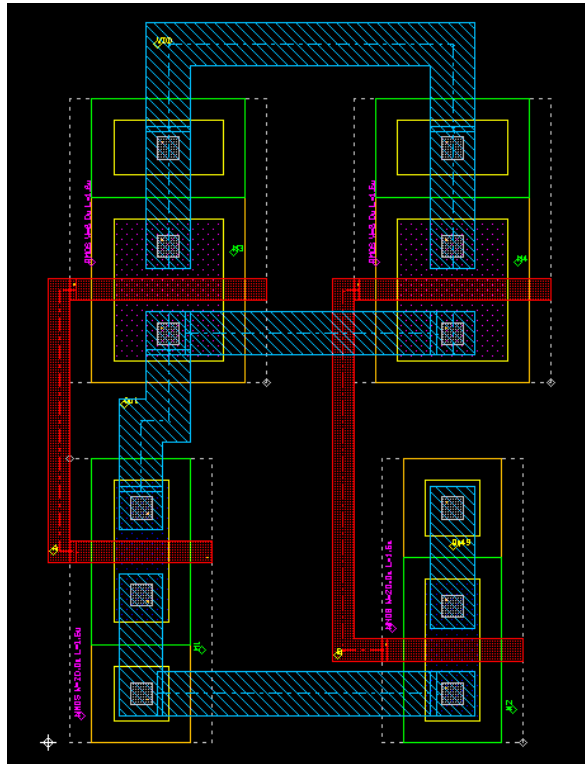


Figura 33 - NAND de duas entradas desenvolvida no LASI (Fonte: Autor).

No *layout* da porta NAND da Figura 33, verifica-se a utilização da tecnologia *standard cell*, tecnologia para a qual o LASI se torna mais prático, devido à facilidade de importar as células já criadas e utilizá-las no projeto. Além disso, verifica-se que a estrutura desenvolvida é composta por dois transistores PMOS e dois NMOS.

A próxima subseção mostra o projeto de outra célula base utilizada no projeto, a NAND de três entradas, que é bem similar à NAND de duas entradas, porém com o acréscimo de dois transistores.

4.1.5 PORTA NAND (NÃO-E) DE TRÊS ENTRADAS

Após o desenvolvimento da função lógica NAND de duas entradas, iniciou-se o desenvolvimento da função lógica NAND, porém com três entradas. A Figura 34 mostra o *layout* desenvolvido no LASI para a porta NAND de três entradas.

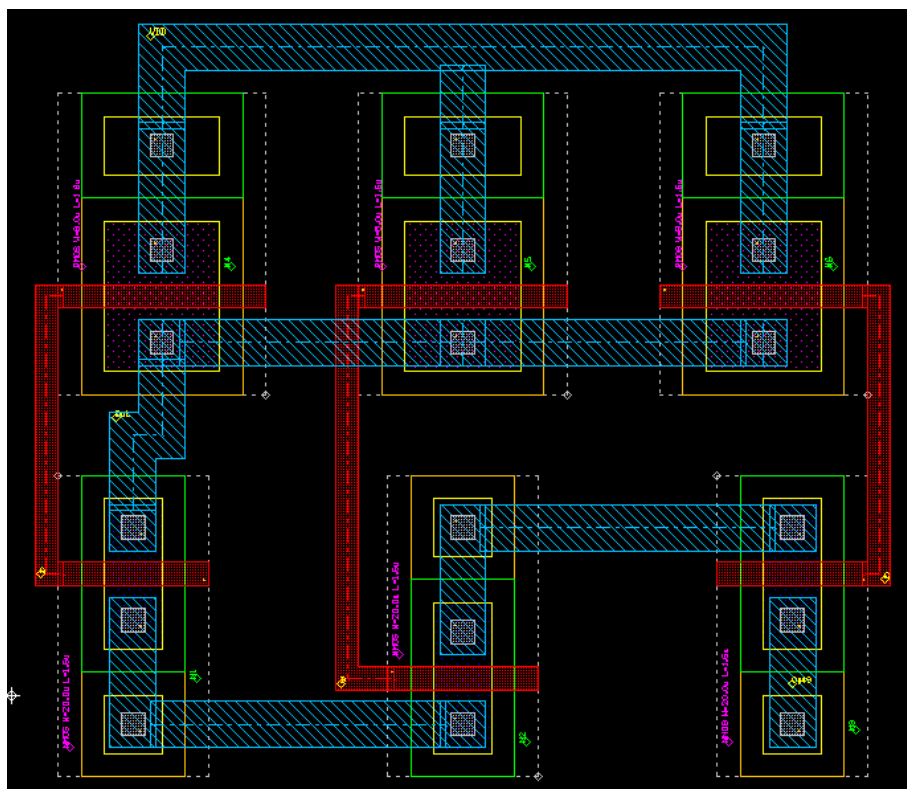


Figura 34 - NAND de três entradas desenvolvida no LASI (Fonte: Autor).

No *layout* da Figura 34, verifica-se que na implementação da porta NAND de três entradas, apenas acrescenta-se um transistor em série na rede *pull-down* e um em paralelo na rede *pull-up* (o que também pode ser visto pelo esquemático).

Para fins de comparação, desenvolveu-se uma porta NAND de três entradas utilizando-se também a tecnologia *full custom*, a fim de verificar a redução da área gasta no chip com uma e outra tecnologia. A Figura 35 mostra as duas portas NAND nas duas diferentes tecnologias lado a lado.

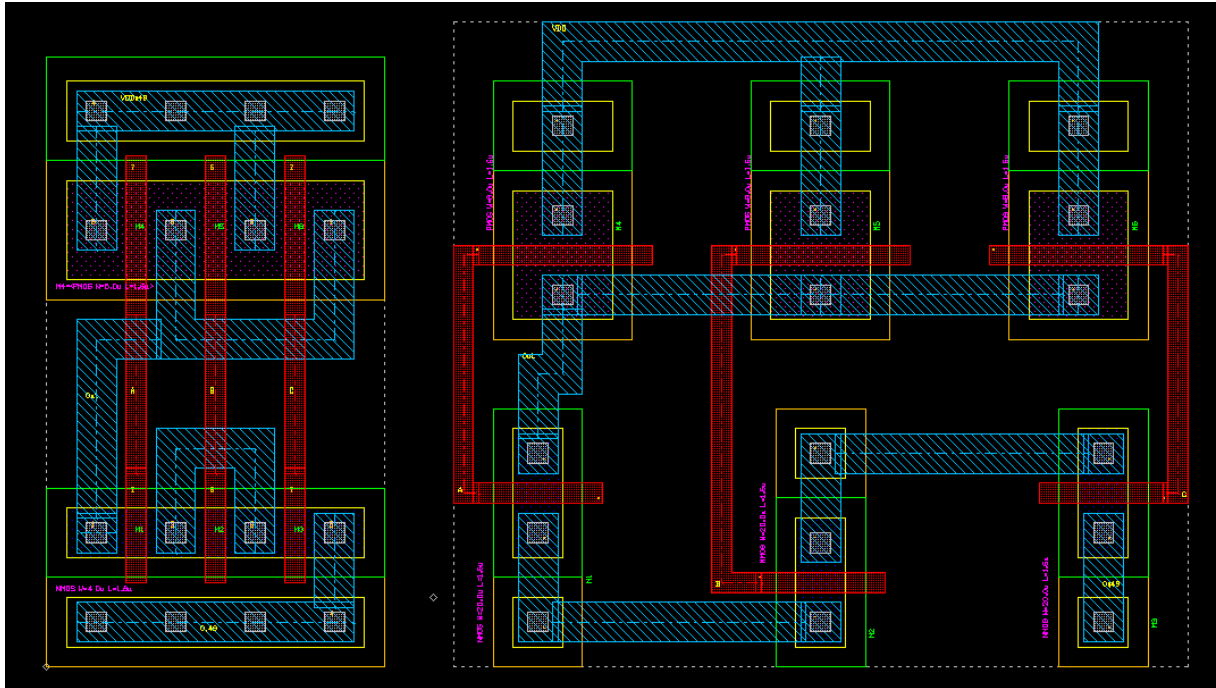


Figura 35 – NAND *full custom* à esquerda e *standard cell* à direita (Fonte: Autor).

Conforme pode ser visto, há uma grande redução na área gasta pela célula na tecnologia *full custom*, da ordem de 56,5% para a estrutura desenhada, porém existe um grande aumento no tempo de execução do *layout*, visto que no *standard cell* utiliza-se os dois transistores projetados e realiza-se apenas a interligação dos mesmos, e na *full custom* os próprios transistores já são construídos e interligados diretamente, acarretando maior tempo de projeto. Além disso, um projeto *full custom* deve ter seu esquemático também desenvolvido no LASI, para que o compilador faça a relação e separação de cada dispositivo do circuito na hora de compilar a *netlist*, o que acaba se tornando um trabalho adicional, visto que a *netlist* pode ser extraída diretamente do *layout* na outra tecnologia, pois os dispositivos são bem definidos.

Todas as estruturas projetadas são utilizadas no projeto do *layout* final do contador, porém são também empregadas no projeto de uma outra sub célula que compõe a estrutura final, o *flip-flop JK*. A próxima subseção apresenta o projeto deste *flip-flop*.

4.1.6 FLIP-FLOP JK

Após o desenvolvimento das portas NAND, iniciou-se o desenvolvimento do *flip-flop* JK. A Figura 36 mostra o *layout* desenvolvido no LASI para o *flip-flop* JK.

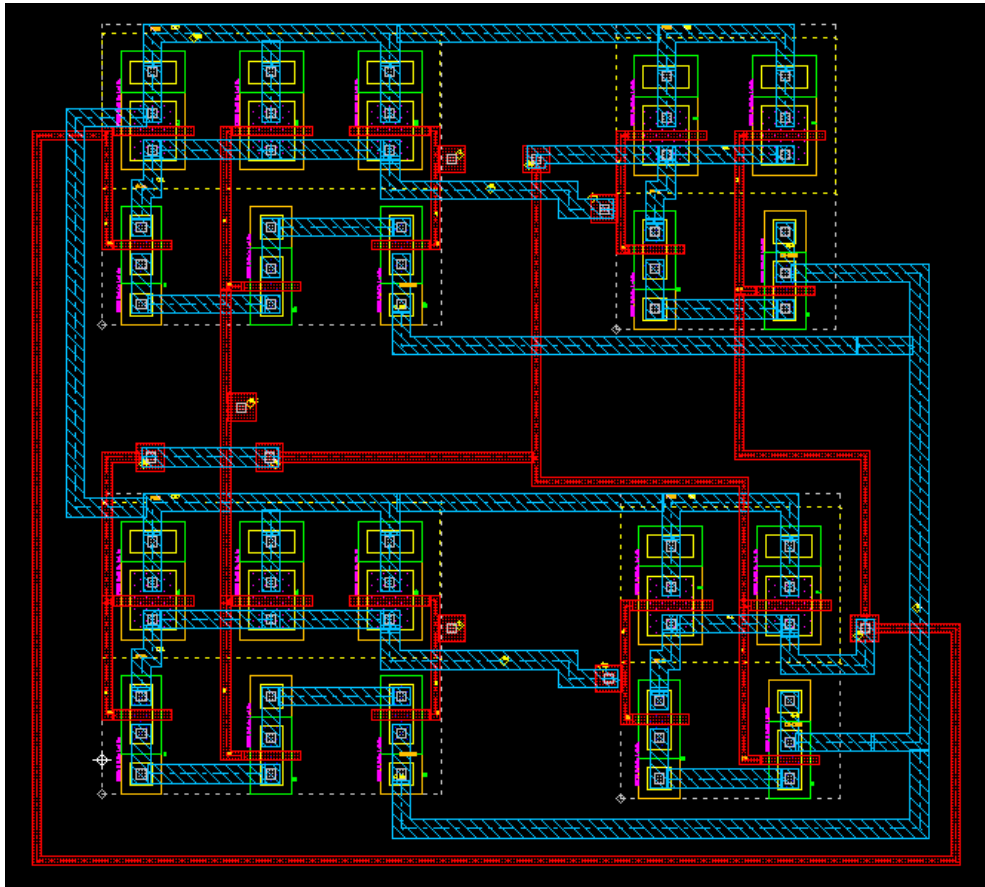


Figura 36 – *Flip-flop* JK desenvolvido no LASI. (Fonte: Autor).

No *layout* do *flip-flop* da Figura 36, verifica-se que o mesmo é composto por vinte transistores ao todo, sendo dez deles PMOS e dez NMOS, que compõem duas portas NAND de três entradas e duas portas NAND de duas entradas, além disso o *flip-flop* desenvolvido conta com as entradas de *clock* (CLK), as entradas independentes J e K e as entradas que são compostas pela realimentação das saídas. O dispositivo conta também com as saídas Q (saída comum) e QB (saída invertida), bem como com a alimentação própria do circuito VDD (cinco volts) e GND (zero volts).

As interligações entre os terminais são realizadas por meio dos *layers* de polissilício e metal, pois os mesmos podem se cruzar sem problemas devido ao fato de serem máscaras diferentes, ou seja, são separados por isolante.

Com o projeto do *flip-flop*, finalizou-se o projeto de todas as células que são utilizadas no *layout* final. A próxima subseção apresenta então, o projeto do *layout* do contador síncrono de quatro *bits*.

4.1.7 CONTADOR SÍNCRONO DE QUATRO BITS

Após o desenvolvimento do *flip-flop* JK, iniciou-se a confecção do *layout* final, objetivo deste trabalho. O mesmo foi desenvolvido utilizando-se quatro *flip-flops* JK, duas portas NAND (uma de duas e outra de três entradas) e dois inversores. A Figura 37 mostra o *layout* desenvolvido no LASI para o contador. Devido à restrição de tamanho, esta mesma figura pode ser visualizada com maior nível de detalhes, em orientação paisagem, na Figura 42 no ANEXO I.

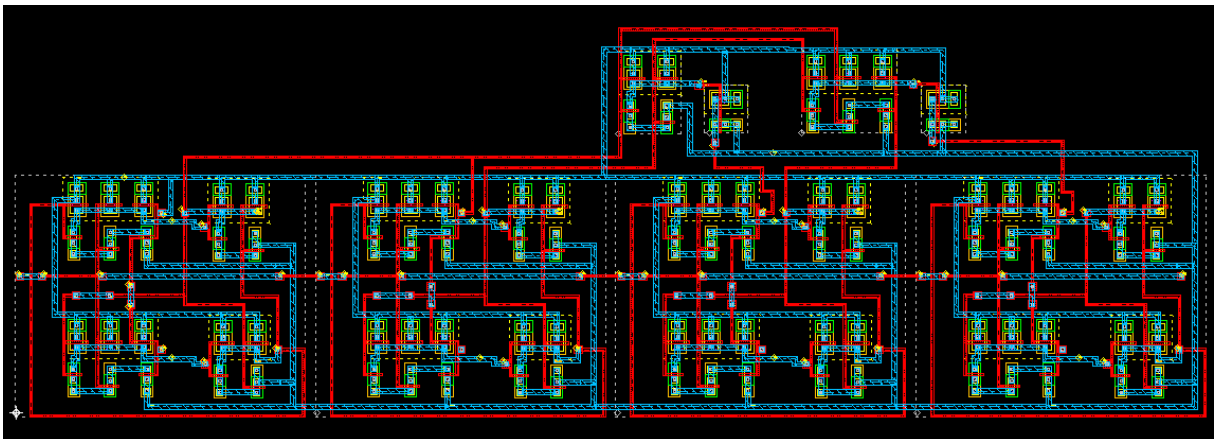


Figura 37 – *Layout* do contador desenvolvido no LASI (Fonte: Autor).

No *layout* do contador da Figura 37, verifica-se que os *flip-flops* do mesmo possui as entradas J e K curto-circuitadas conforme dito anteriormente, verifica-se também que o mesmo conta com uma entrada de *clock* e que as entradas J e K do

primeiro *flip-flop* da cadeia (menos significativo) ficam conectadas à alimentação do circuito.

Para fins de comparação é apresentado na Figura 38 a tela final do complemento DRC para o layout do contador. Mesmo que a simulação DRC tenha apresentado erros por várias vezes, apresentou-se apenas a tela da última vez em que o DRC foi simulado (sem erros), pois todos os erros anteriores foram corrigidos e a simulação foi novamente realizada.

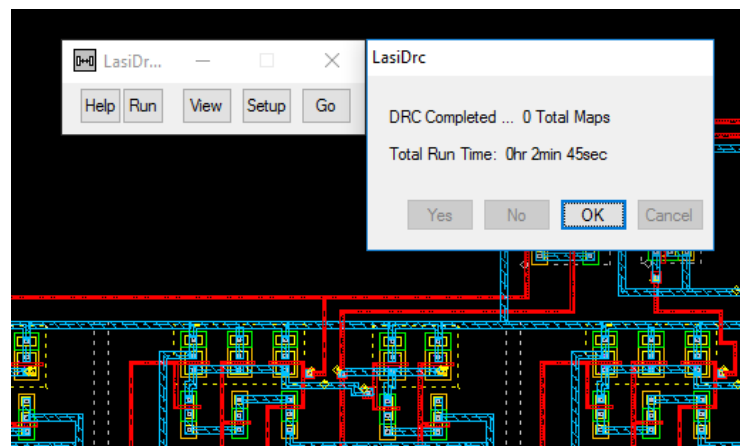


Figura 38 – Tela final do DRC do contador (Fonte: Autor).

Na Figura 38 é possível verificar que o tempo de DRC do contador foi de dois minutos e quarenta e cinco segundos, tempo bem maior se comparado ao DRC do NMOS, que foi de quatro segundos. Tal diferença é justificada pelo aumento da área utilizada pelo circuito e também pelo uso de mais *layers*.

Após o DRC, realizou-se a extração da *netlist*. Ao compilar a extração da *netlist* SPICE no LASI CKT, após a correção de todos os *warnings* e erros gerados, obteve-se o resultado conforme mostra a Figura 39.

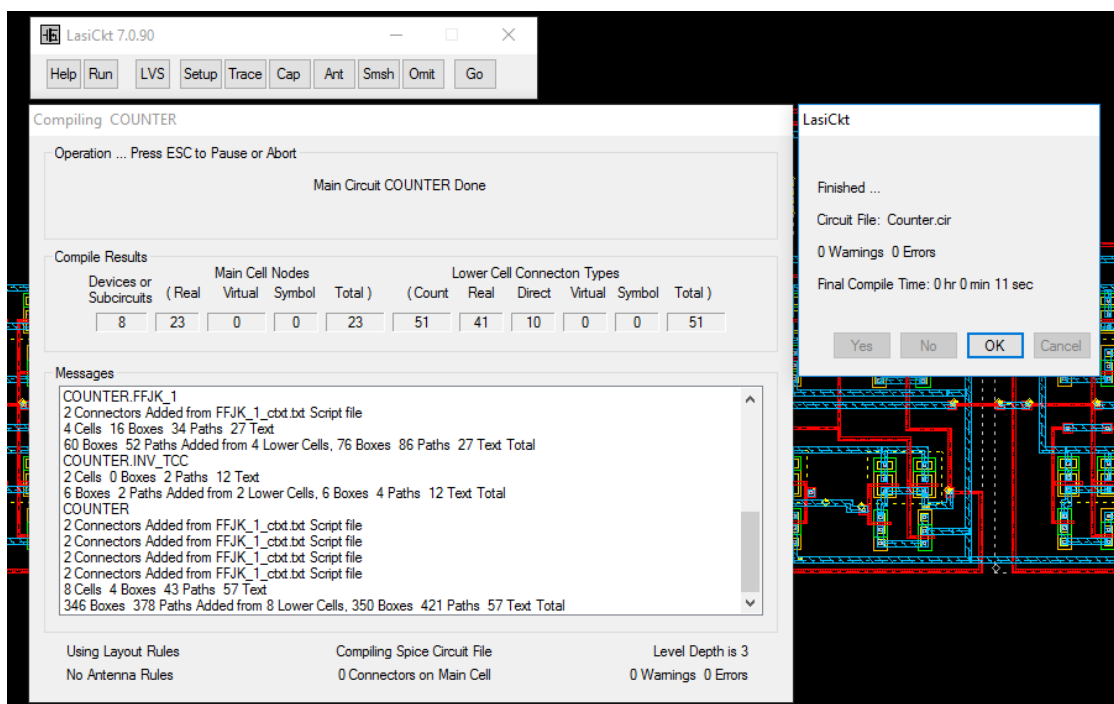


Figura 39 – Tela de compilação do CKT (Fonte: Autor).

Vale ressaltar que o tempo de execução foi baixo desta última vez, devido ao fato de não terem sido encontrados erros, pois na decorrência de um erro o tempo de execução do complemento aumenta consideravelmente. A *netlist* gerada pode ser visualizada no ANEXO II deste trabalho, pois devido à grande quantidade de linhas do código gerado, torna-se inviável a exposição do mesmo por meio de imagens. Na *netlist* é possível verificar os impulsos aos quais o circuito foi submetido, os subcircuitos e circuitos que contêm as conexões detalhadas do circuito, as capacitâncias parasitas que são geradas devido à passagem de uma trilha de um *layer* próxima a outra e que influenciam diretamente no funcionamento do circuito caso assumam valores indesejados (ocasionam acúmulos de carga e campos elétricos em regiões que podem afetar o desempenho do dispositivo desenvolvido), e os modelos SPICE utilizados para descrever os componentes a serem simulados.

A partir da *netlist* é possível realizar todas as simulações no circuito em questão. Inicialmente realizou-se a simulação de funcionamento do circuito, para verificar se o mesmo cumpre sua funcionalidade a nível micro, utilizou-se então um modelo SPICE *level 3* de transistor, modelo que leva em consideração uma maior quantidade de detalhes físicos, inclusive com efeitos quânticos, garantindo que a

simulação seja a mais próxima possível de um componente real, se afastando da consideração de componentes ideais, conforme mostrado em (BAKER, 2010).

Após a geração da *netlist*, ressalta-se o quanto é necessário entender bem o processo de produção da tecnologia CMOS, pois as especificidades exigidas pelo programa na etapa de *layout* (pela visualização das regras de projeto) e na etapa de extração da *netlist* (pelo entendimento da conexão das máscaras), fazem com que o LASI demonstre seu papel como um *software* educacional eficiente, ressaltando que tal ferramenta computacional é gratuita.

Na Figura 40 pode se observar o sinal de *clock* aplicado ao circuito e os sinais de cada um dos estágios do contador até chegar ao *bit* mais significativo. Vale ressaltar que, como o SPICE é um programa fiel ao comportamento do circuito, ele leva em consideração também o regime transitório inicial do circuito, fazendo com que na parte inicial do gráfico haja informação irrelevante para a análise do comportamento do circuito a nível de funcionamento.

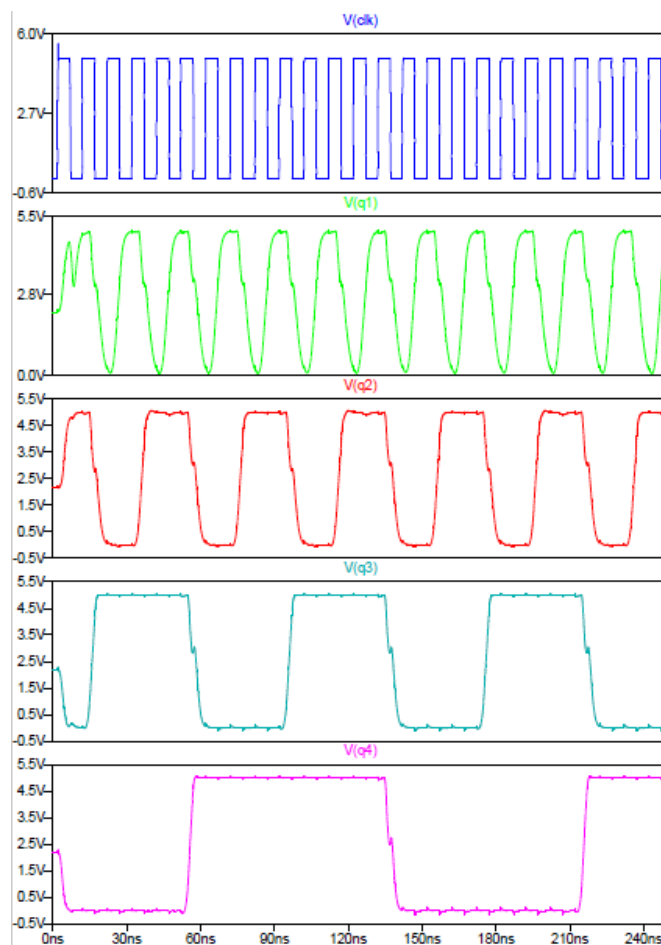


Figura 40 – Sinais de entrada e saída do circuito (Fonte: Autor).

Observando-se a Figura 40 pode se observar que o circuito cumpre bem seu papel na divisão de frequência e, portanto, como contador binário. Porém o circuito apresenta diferentes tempos de subida e descida dentro de um mesmo estágio, e além disso apresenta diferença também nos tempos de subida e descida entre estágios. A Tabela 5 apresenta os tempos de subida e descida extraídos da simulação.

Tabela 5 – Tempos de subida e descida do contador (Fonte: Autor).

TEMPOS		
Estágio	Subida (ns)	Descida (ns)
Q1	4,0	5,7
Q2	3,2	4,8
Q3	2,9	4,5
Q4	2,8	4,2

Conforme pode ser visto na Tabela 5, os tempos de subida e descida de um mesmo estágio são diferentes, isto pode ser justificado pelo fato de que, para que o circuito comute de nível alto para nível baixo, ele depende basicamente do desligamento da rede *pull-up* (composta por transistores PMOS) e do ligamento da rede *pull-down*, e como a velocidade dos portadores dos dispositivos PMOS é menor do que dos NMOS, ocorre esta diferença nos tempos de subida e descida. Além disso, verifica-se que os tempos do primeiro (Q1) estágio são maiores do que dos outros, isto ocorre porque como este é o primeiro estágio que recebe o *clock* ele tem menos tempo para que sua estrutura cristalina realize a recombinação eletrônica e se recupere completamente de uma comutação, por isso pode se perceber melhores tempos nos estágios mais distantes da frequência aplicada. O tempo de recombinação e a velocidade dos portadores dos PMOS juntos justificam o fato de a onda formada pelo primeiro estágio do contador não conseguir alcançar um nível baixo estável (apesar deste ser suficiente para o funcionamento correto do dispositivo), pois devido ao elevado tempo de descida e ao baixo tempo de descanso (para recombinação)

assim que a onda atinge nível baixo, outro pulso de *clock* já ocasionou a subida para nível alto.

Apesar das diferenças nos tempos, é possível ver que o circuito funcionou bem, conforme um contador digital deve funcionar, visto que a frequência do quarto estágio é dezesseis vezes menor do que a frequência de *clock* aplicada. Os valores das frequências referentes à operação do circuito podem ser observados na Tabela 6.

Tabela 6 – Frequência dos estágios do contador (Fonte: Autor).

Estágio	Frequência (MHz)
<i>Clock</i>	100
Q1	50
Q2	25
Q3	12,5
Q4	6,25

Na Tabela 6, observa-se os valores das frequências de cada estágio, retiradas da simulação, que mostram uma grande precisão com relação à divisão de frequências, o que é uma característica e motivo principal da utilização de contadores síncronos, pois estes não carregam atrasos relacionados aos tempos do circuito para a frequência como os contadores assíncronos. Com isso verificou-se que o circuito respeitou corretamente a divisão da frequência por dois de cada um dos estágios anteriores no contador.

Outra análise que se mostra importante é a de sensibilidade do circuito à temperatura, uma vez que ao dissipar potência os circuitos eletrônicos se aquecem, principalmente quando são exigidos a comutar em altas frequências, e este aumento de temperatura pode conduzir o circuito a um mau funcionamento e até mesmo à queima (no caso de altas temperaturas capazes de danificar a integridade física do circuito). A Figura 41 apresenta a simulação da operação do circuito, com todos os estágios da contagem, em determinadas temperaturas, sendo 30°C (verde), 55°C (azul) e 80°C (vermelho).

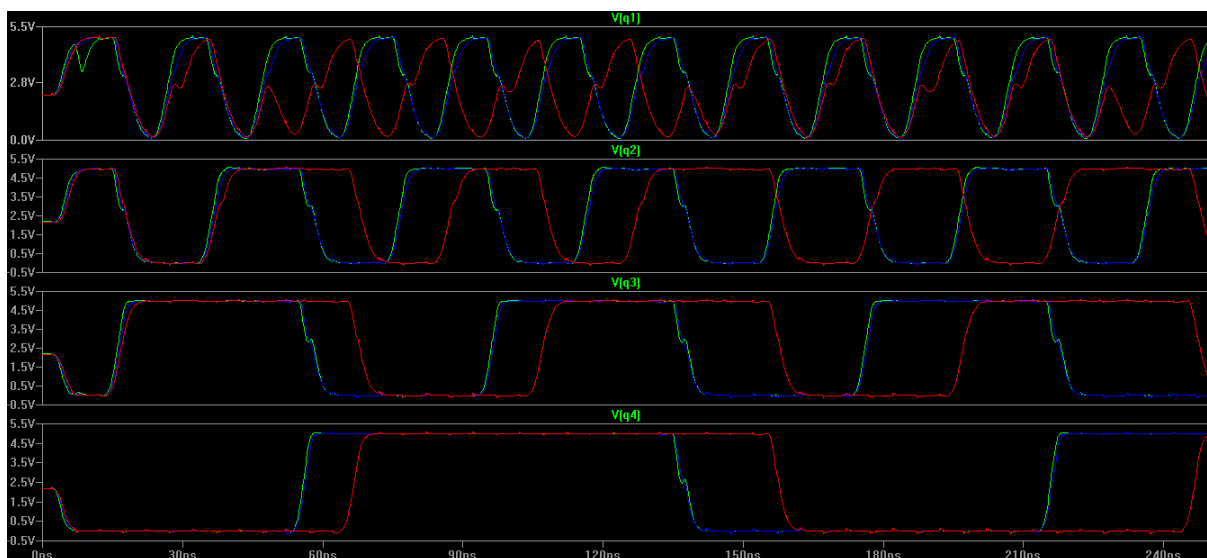


Figura 41 – Simulação de sensibilidade à temperatura com 30°C - verde, 55°C - azul e 80°C - vermelho (Fonte: Autor).

Verifica-se que no primeiro passo de temperatura de 30°C (próximo à temperatura ambiente), o circuito apresenta funcionamento normal. No segundo passo de 55°C, as formas de onda do circuito se comportam um pouco deslocadas para a direita, porém não apresenta grandes alterações com relação ao funcionamento normal. Já no último passo de 80°C, a forma de onda do primeiro estágio de contagem já se apresenta totalmente deformada o que gera atrasos na contagem dos próximos estágios e altera completamente a frequência dos estágios de contagem, portanto este circuito não deve ser submetido a temperaturas desta magnitude. Outras simulações de temperatura foram realizadas, com mais passos de temperatura, culminando na temperatura máxima de operação do contador de 75°C.

Outras simulações foram realizadas (tais como sensibilidade a tensões aplicadas, sensibilidade a frequências, e testes padrões para definir parâmetros do circuito), afim de coletar dados para gerar o *datasheet* parcial (folha de dados com as características elétricas) do circuito integrado desenvolvido. O *datasheet* desenvolvido é parcial pois após a etapa de projeto do *layout* o projeto segue para uma outra etapa (também de complexidade elevada), geralmente com outra equipe, que diz respeito ao encapsulamento do circuito integrado que também gera dados que são dispostos no *datasheet*, sendo que tal atividade não faz parte do escopo desta monografia de trabalho de conclusão de curso.

Os parâmetros do *datasheet* desenvolvido são separados em duas tabelas. A Tabela 7 apresenta os dados de tensões, temperaturas e correntes permitidas dentro da faixa de operação garantida do dispositivo.

Tabela 7 - Faixa de operação garantida (Fonte: Autor).

SIGLA	Parâmetro	Min	Típico	Máx	Unidade
V _{CC}	Tensão da fonte	4,5	5,0	6,0	V
T _A	Faixa de temperatura	0	30	75	°C
I _{OH}	Corrente de saída - Alto			-30	mA
I _{OL}	Corrente de saída - Baixo			25	mA

Na Tabela 8 são apresentados os dados característicos do circuito operando acima da faixa normal de temperatura.

Tabela 8 - Características acima da faixa normal de temperatura (Fonte: Autor).

SIGLA	Parâm	Limites			Unid.	Condições de teste
		Min	Tip	Máx		
V _{IH}	V Entr. Alta	3,7	4,0	-	V	-
V _{IL}	V Entr. Baixa	-	-	0,7	V	-
V _{OH}	V Saída Alta	4,2	4,5	-	V	V _{CC} =MIN, I _{OH} =MAX, V _{IN} =V _{IH}
V _{OL}	V Saída Baixa	-	0,18	-	V	V _{CC} =MIN, V _{IN} =V _{IL}
I _{IH}	I Entr. Alta	-	-	50	uA	V _{CC} =MAX, V _{IN} =4,2V
I _{IL}	I Entr. Baixa	-	-	-200	uA	V _{CC} =MAX, V _{IN} =0,7V
I _{OS}	I Curto Saída	-10	-	-90	mA	V _{CC} =MAX
I _{CC}	I Curto Entr.	-	-	28	mA	V _{CC} =MAX
t _{PLH}	Delay CLK-Q	3,8	4,08	4,5	ns	V _{CC} =5V, C _L =0,4pF
T _{PHL}	Delay CLK-Q	4,5	5,35	6,1	ns	V _{CC} =5V, C _L =0,4pF

Ao verificar os parâmetros finais do circuito, finalizou-se os resultados, uma vez que aspectos referentes ao comportamento do circuito desenvolvido podem ser entendidos pelos parâmetros das Tabelas 7 e 8. Com isso, o próximo capítulo apresenta as conclusões obtidas com a realização deste trabalho.

5. CONCLUSÕES

Com o desenvolvimento deste trabalho de conclusão de curso foi possível adquirir conhecimentos pertinentes a uma área pouco explorada em nosso país. A maior dificuldade no desenvolvimento do trabalho foi a falta de documentação da ferramenta educacional LASI, o que consumiu grande parte do esforço na obtenção do *layout* final proposto nessa monografia.

A tarefa de confeccionar um *layout* de circuito integrado tornou-se bastante complexa devido à vasta gama de conceitos físicos necessários para a integralização do processo. Desse modo, o projetista precisa aplicar tais conceitos levando em conta a tecnologia a ser utilizada, bem como, deter uma visão geral antecipada do *layout* a ser desenvolvido.

De um modo geral, os objetivos iniciais do trabalho foram alcançados satisfatoriamente, sendo que os *layouts* propostos foram desenvolvidos e validados através de simulações. Neste trabalho foram utilizadas apenas ferramentas gratuitas, o LASI e o LTSPICE IV. Ressaltando que o LASI foi o *software* base para o desenvolvimento do *layout* e o LTSPICE IV foi o *software* utilizado na validação do circuito integrado proposto. Vale a pena mencionar também que ambas ferramentas computacionais cumpriram muito bem seus papéis de *softwares* educacionais.

A partir das estruturas e do funcionamento por camadas do LASI, iniciou-se o desenvolvimento pelas células mais básicas (como transistores e inversores), para então passar ao desenvolvimento de células mais avançadas. Nesta etapa também houve grande dificuldade, pois por mais que a etapa de estudos tenha sido sólida, fazer a integração da construção em *software* dos dispositivos e da construção física dos mesmos requer a consideração de muitos fatores que as vezes ocasionaram grandes erros nas células geradas, sendo necessário por várias vezes recomeçar a construção das mesmas do início. Os complementos do LASI (DRC e CKT) fazem com que este programa seja ainda mais usual, e garantindo a visualização das potencialidades e possíveis falhas dos circuitos desenvolvidos.

Após a superação das dificuldades encontradas e correção dos erros ocorridos, o contador desenvolvido apresentou bom desempenho trabalhando com uma frequência de *clock* relativamente elevada, o que foi confirmado pelas simulações

apresentadas, que foram realizadas por meio do *software* LTSPICE IV, além disso validou-se também o desempenho do circuito trabalhando em diferentes temperaturas, obtendo a máxima temperatura de operação do circuito. Com o desenvolvimento do trabalho concluiu-se que a área de projeto de circuitos integrados é restrita devido também ao fato de exigir que o projetista se supere a cada obstáculo, encontrando dentro da complexa literatura da área da microeletrônica (ou nano) a solução para os problemas encontrados.

6. TRABALHOS FUTUROS

Apesar de não as utilizar neste projeto, ressalta-se mais uma vez a importância das HDLs no projeto de circuitos de maior complexidade. O motivo pelo qual não se utilizou a síntese lógica, é o fato de que o LASI é uma ferramenta que tem seu uso facilitado para utilização com a tecnologia *standard cell*, e caso a síntese lógica fosse implementada, seria mais usual se implementar a tecnologia *full custom*, e nesse caso se complicaria a etapa de projeto devido ao aumento do tempo gasto e aumento na complexidade das simulações finais. Porém as potencialidades do projeto customizado foram mostradas, e, portanto, fica como proposta de trabalho futuro o desenvolvimento de circuitos mais complexos, para que as HDLs sejam realmente implementadas como parte fundamental do ciclo de projeto.

Além da utilização da HDL e da tecnologia *full custom* em circuitos mais complexos, fica também como proposta de trabalho futuro a implementação e exploração da tecnologia SOI, que é a principal candidata a substituir a tecnologia CMOS empregada atualmente, conforme é mostrado em (RABAEY, *et al*, 2002). Tal tecnologia se chama SOI (*Silicon On Insulator* – Silício sobre Isolante) e tem a característica de utilizar do processo de produção basicamente inverso ao da atual tecnologia CMOS.

7. REFERÊNCIAS BIBLIOGRÁFICAS

BAKER, R. Jacob. **CMOS Circuit Design, Layout and Simulation**. Wiley - IEEE Press, Third Edition, 2010.

BRINKMAN, William F; *et al.* **A History of the Invention of the Transistor and Where It Will Lead Us**. IEEE Journal of Solid-state Circuits. Vol. 32, nº 12. December 1997.

CADENCE TOOLS. **Cadence Verification Suite for System Design and Verification**. Disponível em: < <https://www.cadence.com/>> Acesso em: 27 mai. 2017.

CASILLO, Leonardo Augusto. **Minicurso: Tópicos em VHDL**. Escolha Potiguar de Computação e suas Aplicações, 2010.

CHEAH, Kok Wai. **History of Integrated Circuit (IC)**. Department of Physics, Hong Kong Baptist University, 2008.

CHIQUITO, Adenilson J.; LANCIOTTI, Francesco Jr. **O Transistor, 50 Anos**. Departamento de Física da Universidade Federal de São Carlos. Revista Brasileira de Ensino de Física vol. 20, no. 4, Dezembro, 1998.

CMOSEDU. **MOSIS RULES – Set of Design Rules**. Disponível em: <<http://cmosedu.com/cmos1/winlasi/lasiproj/scells.htm>>. Acesso em: 15 mar. 2017.

EDGAR, Lilienfeld Julius. **Method and apparatus for controlling electric currents**. U.S. Patent n. 1745175 A. 1926.

FERREIRA. E. C. **Transistores de Efeito de Campo (FETS)**. Departamento de Eletrônica e Microeletrônica – Unicamp, 2012.

FONTANA, Eduardo. **Estrutura de bandas dos condutores, isolantes e semicondutores**. UFPE, 2011. Disponível em: <<https://www.ufpe.br/fontana/Eletromagnetismo1/EletromagnetismoWebPart01/mag1cap4.htm>>. Acesso em 07 de fevereiro de 2017.

GOMES, Otávio de Souza Martins. **Notas do Curso de Extensão - Linguagens de Descrição de Hardware**. Instituto Federal de Minas Gerais - Campus Formiga, 2014.

GUIZZO, Erico Marui. **O microchip: pequena invenção grande revolução.** Laboratório de Sistemas Integráveis - Escola Politécnica da Universidade de São Paulo. 2000. Disponível em: <<http://www.lsi.usp.br/~chip/index.html>>. Acesso em: 03 jan. 2015.

KANG, Sung – Mo; LABLEBICI, Yusuf. **CMOS Digital Integrated Circuits – Analysis and Design.** WCB McGraw-Hill, Third Edition, 2003.

LASI 7. **What is LASI?** Disponível em: <<http://lasihome.com/index.htm>>. Acesso em: 03 jan 2015.

LOUREIRO, I. F. K. **REAL OPTIONS APPLICATION ON INTEGRATED CIRCUITS. SECTOR.** PUC. Rio de Janeiro, 2011. Disponível em: <http://www.maxwell.lambda.ele.pucRio.br/Busca_etds.php?strSecao=resultado&nrSeq=16851@1>. Acesso em: 12 jan. 2017.

MAGNO, Afrânio Júnior. **Circuito Integrado: Amplificador Operacional.** Trabalho de Conclusão de Curso, Universidade São Francisco, Campinas, 2007.

MALVINO, Albert Paul. **Electronic Principles.** McGraw-Hill, 2015.

MEHL, Ewaldo Luiz de Mattos. **Do Transistor a Microprocessador.** Disponível em: <http://stoa.usp.br/kblane/files/827/4714/historia_transistor.pdf>. Acesso em: 03 jan. 2015.

MIDORIKAWA, Edison T. **Uma Introdução às Linguagens de Descrição de Hardware.** Escola Politécnica da Universidade de São Paulo, 2007.

NAGEL, Laurence W. **SPICE2: A Computer Program to Simulate Semiconductor Circuits.** Memorandum No. ERL-M520, University of California, Berkeley, May. 1975.

NAGEL, Laurence W.; PEDERSON, Donald O. **SPICE (Simulation Program with Integrated Circuit Emphasis).** Memorandum No. ERL-M382, University of California, Berkeley, Apr. 1973.

PSPICE TOOL. **PSPICE Advanced Analysis.** <<http://www.pspice.com/>> Acesso em: 27 mai. 2017.

RABAEY, Jan M.; CHANDRAKASAN, Anantha P.; NIKOLIC, Borivoje. **Digital integrated circuits**. Englewood Cliffs: Prentice hall, 2002.

REIS, Ricardo Augusto da Luz. **Concepção de Circuitos Integrados**. 2ª Ed. Porto Alegre: Bookman. Instituto de Informática da UFRGS, 2008. v. 7.

STEER, Michael B. **SPICE: User's guide and reference**. Manual, Edition 1.3, July 2, 2007.

STRYHALSKI, Joel. **Materiais semicondutores: estrutura e dopagem**. UNESC, 2005. Disponível em: <<http://www.mundofisico.joinville.udesc.br/index.php?idSecao=110&idSubSecao=&idTexto=49>>. Acesso em 01 de março de 2015.

SWART, Jacobus Willibrordus. **Evolução de Microeletrônica a Micro-Sistemas**. CCS e FEEC - UNICAMP, 2001.

TEXAS INSTRUMENTS. **The chip that Jack Kilby Built – Texas Instruments**. Disponível em: <<http://www.ti.com/corp/docs/kilbyctr/jackbuilt.shtml> >. Acesso em: 23/10/2016.

TOCCI, Ronald J.; WIDMER, Neal S.; MOSS, Gregory L. **Sistemas digitais: princípios e aplicações**. Prentice Hall, 2003.

UNIVERSITY OF DENVER. **History and Theory of Thermionic or Vacuum Tubes**. Disponível em: <<http://mysite.du.edu/~etuttle/electron/elect27.htm#Theory>>. Acesso em: 01 abr. 2017.

WARWICK, Colin. "**Everything you always wanted to know about SPICE* (*But were afraid to ask)**". EMC Journal (Nutwood UK Limited) (82): 27--29. ISSN 1748-9253, May 2009.

8.

ANEXO I

O *layout* do contador projetado é apresentado na Figura 42.

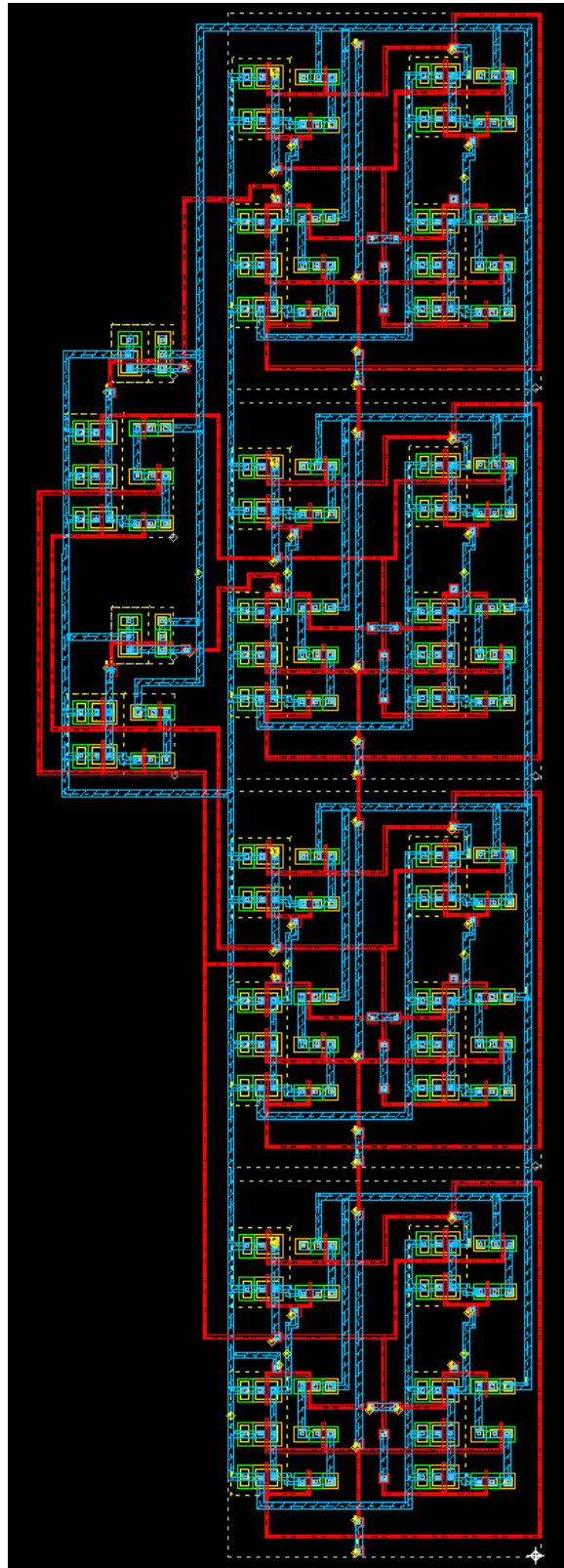


Figura 42 – Layout do contador projetado (Fonte: Autor).

9.

ANEXO II

Neste ANEXO, encontra-se a *netlist* SPICE, gerada a partir do *layout* do contador projetado no LASI e utilizada para as simulações a nível micro.

```
*** Spice Circuit File of COUNTER - LasiCkt 7.0.90 03/21/17
21:18:29
```

```
*Note: Read Text with Fixed Pitch Font
```

```
* Start of C:\Users\Renner\Tcc\Simulações\Counter\Header.txt
```

```
*.OPTIONS LIST NODE POST
```

```
.PRINT TRAN V(Q1) V(Q2) V(Q3) V(Q4) V(CLK)
```

```
* Opções
```

```
*.STEP TEMP 30 80 25 (análise de temperatura - tirar comentário)
```

```
* Tipo de análise
```

```
.TRAN 0.05n 250n
```

```
* Estímulos
```

```
Vsupp VDD 0 DC 5
```

```
C_IN VDD 0 0.9u
```

```
Vin1 CLK 0 PULSE(0 4 2n 100p 100p 5n 10n)
```

```
C_IN2 CLK 0 0.9u
```

```
CLOAD1 Q1 0 0.4p
```

```
CLOAD2 Q2 0 0.4p
```

```
CLOAD3 Q3 0 0.4p
```

```
CLOAD4 Q4 0 0.4p
```

```
* End of C:\Users\Renner\Tcc\Simulações\Counter\Header.txt
```

*** COUNTER.FFJK_1.NAND_3IN_1 ***

.SUBCKT NAND_3IN_1 VDD 0 A B C Out

M1 Out A vn2 vn2 NMOS W=8.0u L=1.6u
M2 vn2 B vn4 vn4 NMOS W=8.0u L=1.6u
M3 vn4 C 0 0 NMOS W=8.0u L=1.6u
M4 Out A VDD VDD PMOS W=16.0u L=1.6u
M5 Out B VDD VDD PMOS W=16.0u L=1.6u
M6 Out C VDD VDD PMOS W=16.0u L=1.6u

* Node to Gnd Parasitic Caps

C_A A 0 0.055975fF
C_B B 0 0.078975fF
C_C C 0 0.055975fF
C_Out Out 0 0.6664335fF
C_VDD VDD 0 0.721764fF
C_vn2 vn2 0 0.2665235fF
C_vn4 vn4 0 0.2665235fF

* Node to Node Parasitic Caps

C_B_Out B Out 0.0891268fF

.ENDS

*** COUNTER.FFJK_1.NAND_2IN_1 ***

.SUBCKT NAND_2IN_1 VDD 0 A B Out

M1 Out A vn2 vn2 NMOS W=8.0u L=1.6u
M2 vn2 B 0 0 NMOS W=8.0u L=1.6u

M3 Out A VDD VDD PMOS W=16.0u L=1.6u

M4 Out B VDD VDD PMOS W=16.0u L=1.6u

* Node to Gnd Parasitic Caps

C_A A 0 0.055975fF

C_B B 0 0.078975fF

C_Out Out 0 0.4638995fF

C_VDD VDD 0 0.418376fF

C_vn2 vn2 0 0.2665235fF

* Node to Node Parasitic Caps

C_B_Out B Out 0.0891268fF

.ENDS

*** COUNTER.FFJK_1 ***

.SUBCKT FFJK_1 VDD 0 CLK J QN QB SN31D SN31U

X_1_NAND_3IN_1 VDD 0 QN CLK J SN31D NAND_3IN_1

X_2_NAND_3IN_1 VDD 0 QB CLK J SN31U NAND_3IN_1

X_3_NAND_2IN_1 VDD 0 SN31U QB QN NAND_2IN_1

X_4_NAND_2IN_1 VDD 0 SN31D QN QB NAND_2IN_1

* Node to Gnd Parasitic Caps

C_CLK CLK 0 1.895671fF

C_J J 0 0.619093fF

C_QB QB 0 1.557319fF

C_QN QN 0 1.213563fF

C_SN31D SN31D 0 0.670493fF

C_SN31U SN31U 0 0.661293fF

C_VDD VDD 0 2.2621995fF

* Node to Node Parasitic Caps

C_CLK_J CLK J 0.0891268fF

C_CLK_QB CLK QB 0.1781136fF

C_CLK_QN CLK QN 0.1779736fF

C_CLK_VDD CLK VDD 0.0891136fF

C_J_QN J QN 0.0889736fF

C_QB_VDD QB VDD 0.0891136fF

C_QN_SN31U QN SN31U 0.0892932fF

C_QN_VDD QN VDD 0.08928fF

.ENDS

*** COUNTER.INV_TCC ***

.SUBCKT INV_TCC VDD 0 In Out

M1 Out In 0 0 NMOS W=8.0u L=1.6u

M2 Out In VDD VDD PMOS W=16.0u L=1.6u

* Node to Gnd Parasitic Caps

C_In In 0 0.00998fF

C_Out Out 0 0.183788fF

* Node to Node Parasitic Caps

.ENDS

*** COUNTER ***

* COUNTER

```

X_1_FFJK_1 VDD 0 CLK VDD Q1 QB1 S2 S1 FFJK_1
X_2_FFJK_1 VDD 0 CLK Q1 Q2 QB2 S4 S3 FFJK_1
X_3_FFJK_1 VDD 0 CLK J3 Q3 QB3 S6 S5 FFJK_1
X_4_FFJK_1 VDD 0 CLK J4 Q4 QB4 S8 S7 FFJK_1
X_5_NAND_3IN_1 VDD 0 Q2 Q1 Q3 OutN31 NAND_3IN_1
X_6_NAND_2IN_1 VDD 0 Q1 Q2 OutN21 NAND_2IN_1
X_7_INV_TCC VDD 0 OutN21 J3 INV_TCC
X_8_INV_TCC VDD 0 OutN31 J4 INV_TCC

```

* Node to Gnd Parasitic Caps

```

C_CLK CLK 0 0.158205fF
C_J3 J3 0 0.409368fF
C_J4 J4 0 0.528183fF
C_OutN21 OutN21 0 0.41789fF
C_OutN31 OutN31 0 0.4100945fF
C_Q1 Q1 0 1.409495fF
C_Q2 Q2 0 0.746635fF
C_Q3 Q3 0 0.32213fF
C_VDD VDD 0 6.390804fF
C_S1 S1 0 0.6509pF
C_S2 S2 0 0.652pF
C_S3 S3 0 0.601pF
C_S4 S4 0 0.6031pF
C_S5 S5 0 0.60011pF
C_S6 S6 0 0.60123pF
C_S7 S7 0 0.6008pF
C_S8 S8 0 0.60105pF

```

* Node to Node Parasitic Caps

C_Q1_VDD Q1 VDD 0.1781004fF

C_Q2_VDD Q2 VDD 0.1782536fF

* Start of C:\Users\Renner\Tcc\Simulações\Counter\Footer.txt

* Modelos MOS para um processo 1.2um (n-well)

* obtidos a partir de informação disponível em

* <http://www.mosis.org/Technical/Testdata/old-test-data.html>

.MODEL nmos NMOS

+ LEVEL=3 PHI=0.70 TOX=1.3E-08 XJ=0.20U TPG=1

+ VTO=0.8 DELTA=2.5E-01 LD=4.0E-11 KP=1.88E-04

+ UO=545 THETA=2.5E-01 RSH=2.1E+01 GAMMA=0.62

+ NSUB=1.4E+17 NFS=7.1E+11 VMAX=1.9E+05 ETA=2.2E-02

+ KAPPA=9.7E-02 CGDO=3.7E-10 CGSO=3.7E-10 CGBO=4.0E-10

+ CJ=5.4E-04 MJ=0.6 CJSW=1.5E-10 MJSW=0.3 PB=0.99

*

.MODEL pmos PMOS

+ LEVEL=3 PHI=0.70 TOX=1.3E-08 XJ=0.20U TPG=-1

+ VTO=-0.9 DELTA=2.5E-01 LD=6.7E-11 KP=4.45E-05

+ UO=130 THETA=1.8E-01 RSH=3.4E+00 GAMMA=0.52

+ NSUB=9.8E+16 NFS=6.5E+11 VMAX=3.1E+05 ETA=1.8E-02

+ KAPPA=6.3E+00 CGDO=3.7E-10 CGSO=3.7E-10 CGBO=4.3E-10

+ CJ=9.3E-04 MJ=0.5 CJSW=1.5E-10 MJSW=0.3 PB=0.95

* End of C:\Users\Renner\Tcc\Simulações\Counter\Footer.txt

.END