

# OVERVIEW DE SOFTWARES OPEN SOURCE PARA DESIGN DE CHIP

Descubra as ferramentas Open Source disponíveis para Design de Chips e conheça os programas de incentivo e apoio ao redor do Mundo.

**13 DEZEMBRO 2023 | 19H30**

**INSCREVA-SE**



**VASCO CÔRTE-REAL**

ENG. ELETRÔNICO - TCO DA  
TOUCAN-ELAB



Patrocinado por



**MOUSER**  
**ELECTRONICS**

# Motivação

Até 2019, realizar um design completo de um chip não comercial, só seria possível em Universidades e Laboratórios de Microeletrônica.

Nesta data já existiam as ferramentas necessárias para as etapas de Design de Chip utilizando Software Livre, porém eram iniciativas isoladas em cada Centro de Laboratórios com seus próprios arquivos de implementação de transistores para o processo de manufatura e sem uma padronização.

Estes arquivos de implementação para o processo fabril chamado PDK de Manufatura nas grandes Foundries, eram comercializados somente para grandes empresas.

# Motivação

Em março de 2019, realizou-se em Paris, França, a primeira Free Silicon Conference já com objetivos de divulgar os trabalhos de Design de Chips com Software Open Source.

**FSiC2019**  
Free Silicon Conference

Pouco meses depois da realização desta conferência, através de um Projeto do Google, a Foundrie Skywater lançou o primeiro PDK (130nm) Open Source, seguido de outros dois PDKs Open Source da IHP GmbH (130 nm BICMOS) e Global Foundries (180 nm)



# Nossa Motivação

Este webinar tem como objetivo maior, o de informar, inspirar e iniciar uma comunidade que esteja motivada em entrar na área de design de Circuitos Integrados, utilizando Software livre e participar das iniciativas a nível mundial de estímulo a manufaturas de Chips.

# Agenda

- Os Semicondutores através das décadas
- Fluxo de Design ( Design Flow ) de Circuitos Integrados
- Open Source Software Tools
- Electronic Design Automation ( EDA ) and Open Source Iniciatives
- Ambiente de Desenvolvimento
- AI no Design de Circuitos Integrados



TOUCAN  
eLAB



# Os Semicondutores através das Décadas





# O Semicondutor através das décadas

**1950s**

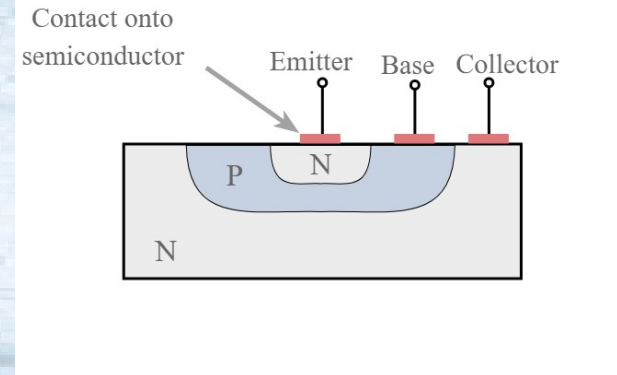
Silicon  
Transistor



**1**

**Transistor**

1959



A tecnologia planar abriu as portas para a fabricação em massa de circuitos integrados complexos e permanece a principal tecnologia usada até hoje.

<https://www.computerhistory.org/siliconengine/timeline/>



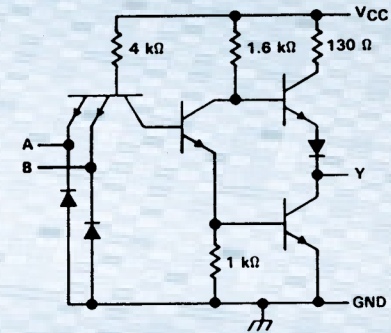
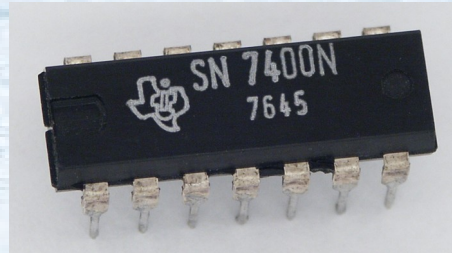
# O Semicondutor através das décadas

**1960s**

TTL  
Quad Gate



**16**  
**Transistors**



Schematic (each gate) '00

Frank Wanlass da Fairchild publicou a idéia do circuito MOS-complementar (CMOS).

Hoje a Tecnologia CMOS é aplicada na grande maioria de CI's de alta densidade.

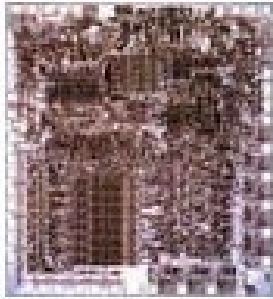
<https://www.computerhistory.org/siliconengine/timeline/>



# O Semicondutor através das décadas

**1970s**

**8-bit  
Microprocessor**



**4500  
Transistors**



**1972  
Intel 8008**

**1974**

**Intel 8080**



LSI NMOS de 6 microns

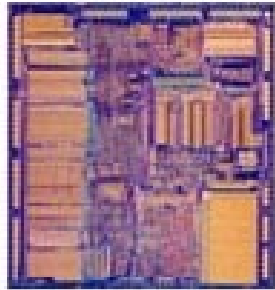
<https://www.computerhistory.org/siliconengine/timeline/>



# O Semicondutor através das décadas

**1980s**

32-bit  
Microprocessor



**275,000  
Transistors**

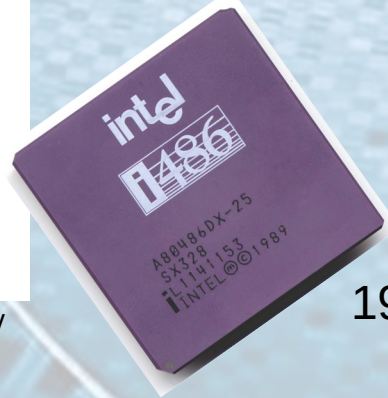
1985



Intel 80386DX

- Primeiro processador de 32 bits da Intel.
- Tecnologia CMOS com porta de silício, dimensão mínima de 1.5µm.
- 10 níveis de máscara, 1 camada de polysilício e 2 camadas de metal.
- 275.000 transistores.
- Freqüência de "clock" de 33MHz.
- Tamanho da pastilha : 104mm<sup>2</sup>.

1989



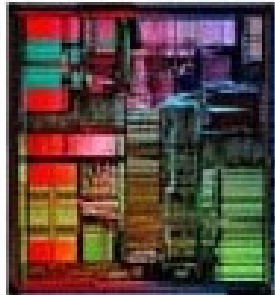
<https://www.computerhistory.org/siliconengine/timeline/>



# O Semicondutor através das décadas

**1990s**

32-bit  
Microprocessor



**3,100,000  
Transistors**

<https://www.computerhistory.org/siliconengine/timeline/>



1993

Intel Pentium

- Tecnologia BiCMOS com porta de silício, largura mínima de  $0.8\mu\text{m}$
- 18 níveis de máscara, 1 camada de polysilício, 3 camadas de metal
- 3,1 milhão transistores
- frequência de "clock" de 66MHz
- Tamanho da pastilha de  $264\text{mm}^2$ .

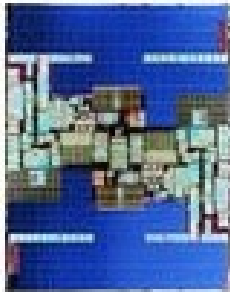
A indústria de semicondutores ultrapassa a casa dos US\$100 bilhões.

1994

# O Semicondutor através das décadas

**2000s**

64-bit  
Microprocessor



**592,000,000  
Transistors**



2004

Intel Itanium 2 Processor ( 9 Mb Cache )

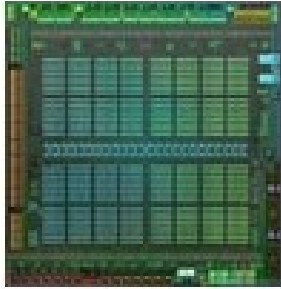
<https://www.computerhistory.org/siliconengine/timeline/>



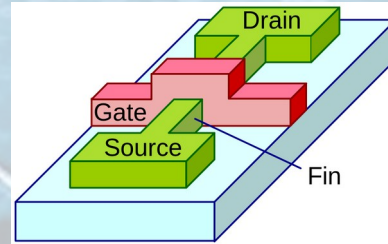
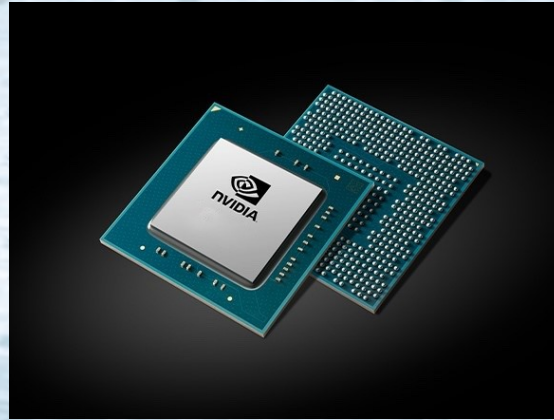
# O Semicondutor através das décadas

**2010s**

3072-Core  
GPU



8,000,000,000  
Transistors



Início da aplicação da Tecnologia FinFET em Chips comerciais de 22 nm

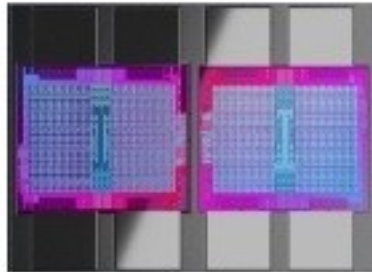
<https://www.computerhistory.org/siliconengine/timeline/>



# O Semicondutor através das décadas

**2020s**

AI  
GPU



**153,000,000,000  
Transistors**

<https://www.computerhistory.org/siliconengine/timeline/>

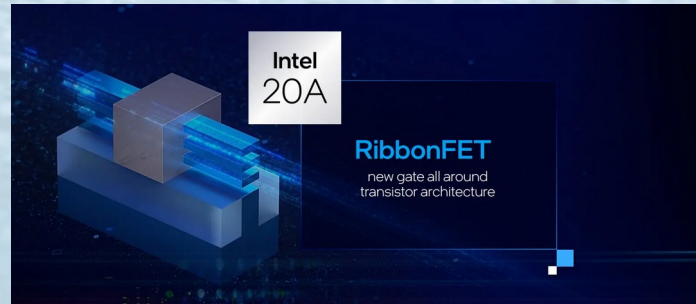
Durante a 34ª conferência



(2022)

O CEO da Intel Pat Gelsinger, afirmou que:

“chips terão 1 trilhão de transistores até 2030”.



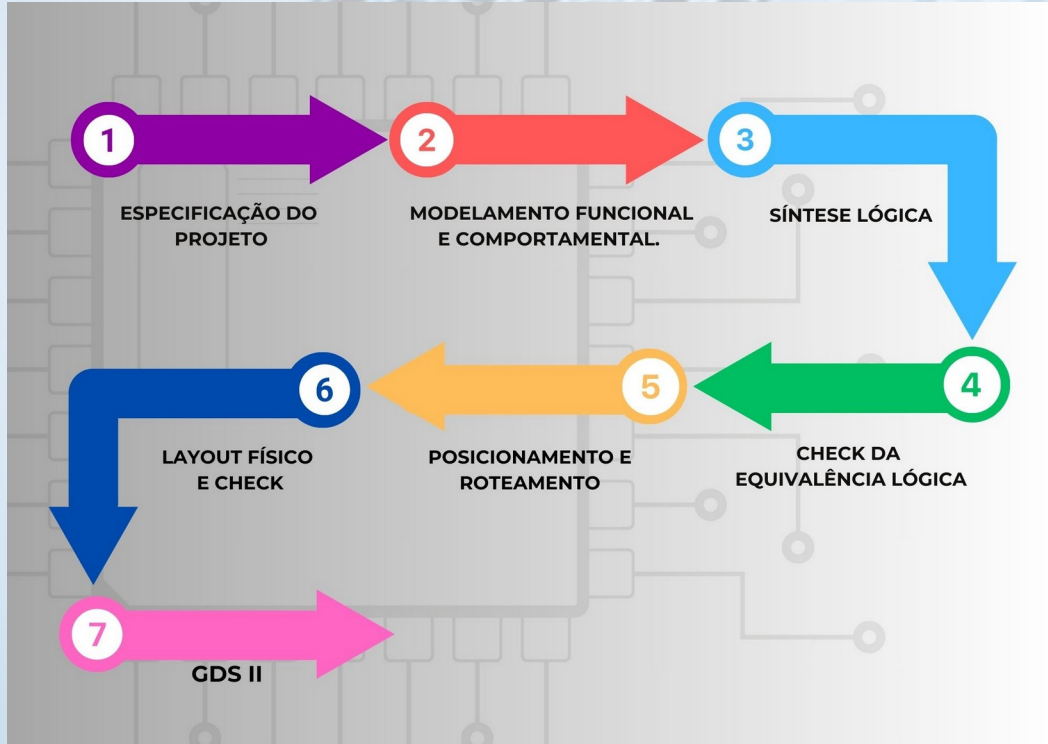
2024

Início da aplicação em  
chips comerciais da  
Tecnologia

RibbonFET

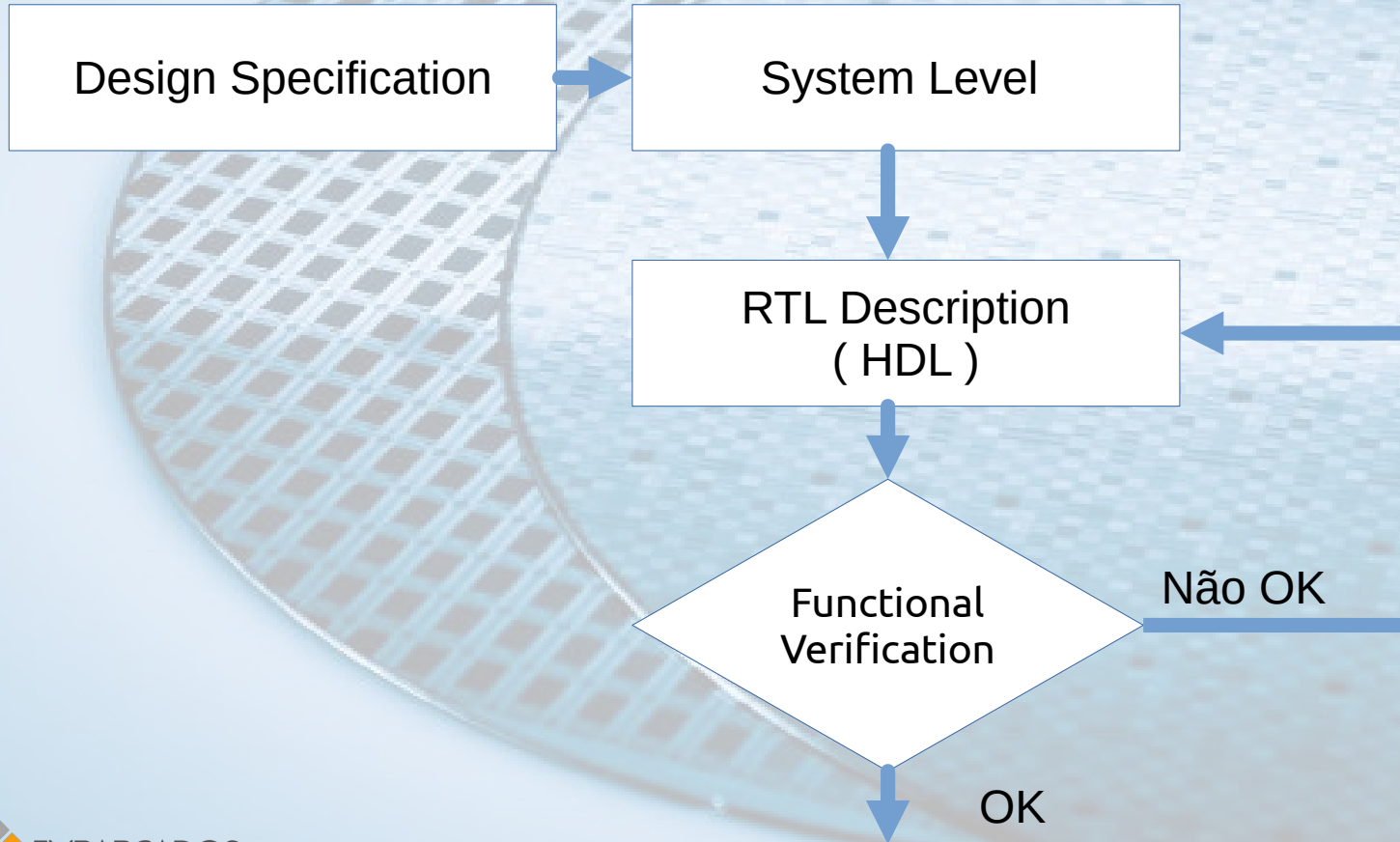
# Fluxo de Design ( Design Flow ) de Circuitos Integrados

# Fluxo de Design de um CI de Aplicação Especifica - ASIC

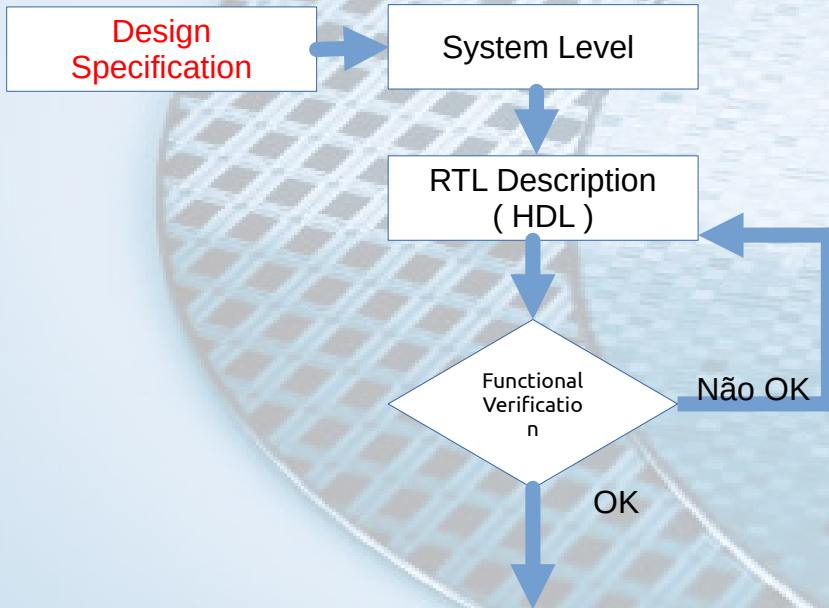


É um conjunto de ferramentas e métodos usados para transformar uma linguagem comportamental de alto nível em um circuito físico que se tornará um código de configuração para um **FPGA** ou um layout de um processo de fabricação de um **ASIC**. O Processo completo de Design pode ser dividido em duas partes:

- Design de Front-End
- Design de Back-End



# Design Specification



Esta é a fase em que o engenheiro define recursos, microarquitetura, funcionalidades (interface hardware/software), especificações (Tempo, Área, Potência, Velocidade) com diretrizes de projeto do ASIC.

Um dos métodos para executarmos nosso Projeto é decompor por meio de Níveis de Abstração.

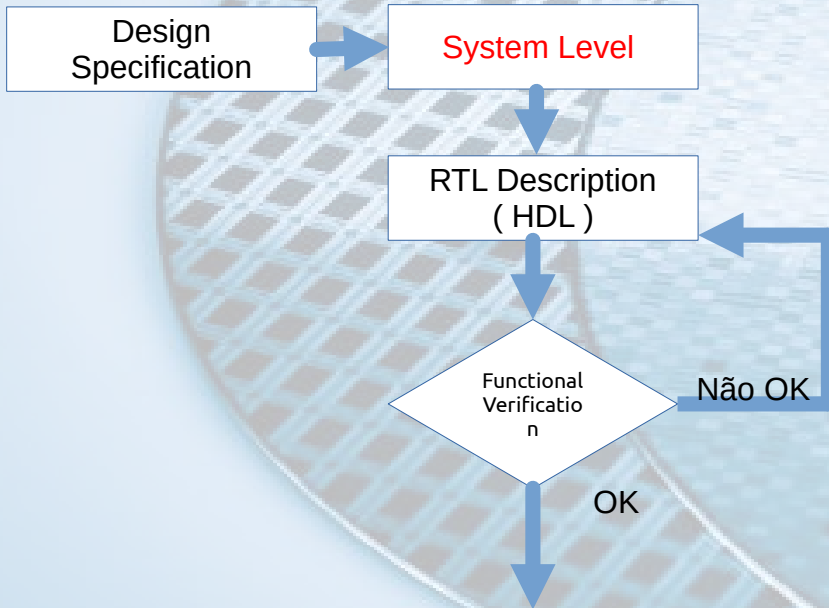
# Níveis de abstração

Os circuitos digitais podem ser representados em diferentes níveis de abstração.

No fluxo do Projeto, iremos procurar encontrar uma representação funcionalmente equivalente de abstração de nível superior em um nível de abstração inferior.

Quando isso é feito automaticamente por meio de software, usamos o termo **síntese**.

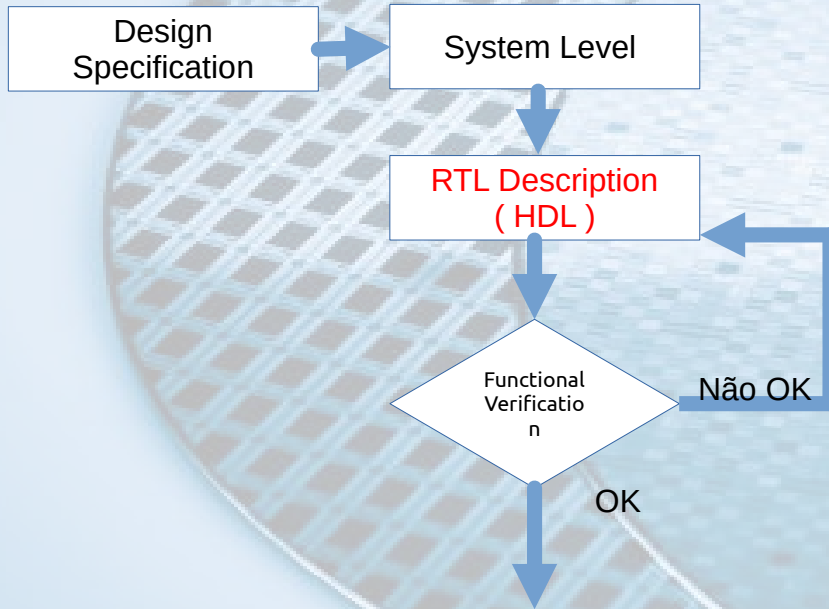
A síntese é a conversão automática de uma representação de alto nível de um circuito em uma representação de baixo nível funcionalmente equivalente.



A abstração de nível de sistema, analisa apenas seus maiores blocos de construção, como CPUs e núcleos de computação.

O circuito é geralmente descrito usando linguagens de programação tradicionais como C/C++ ,Matlab ou SystemC para simulação no nível do sistema.

O padrão IEEE 1685-2009 define o formato de arquivo IP-XACT que pode ser usado para representar projetos no nível do sistema e blocos de construção tipo IP-Block.



A abstração de alto nível de um sistema (às vezes chamada de nível algorítmico) também é frequentemente representada por linguagens de programação tradicionais, mas com um conjunto reduzido de recursos.

Existem ferramentas para sintetizar código de alto nível (geralmente na forma de código C/C++/SystemC com metadados adicionais) em código HDL comportamental (geralmente na forma de código Verilog ou VHDL).

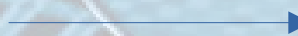
# Linguagem de descrição de Hardware HDL ( Hardware Description Language )

É uma linguagem de alto nível usada para descrever a estrutura e o comportamento de circuitos eletrônicos e, mais comumente, de circuitos lógicos digitais.

As HDLs foram criados para **implementar abstração em nível de transferência de registro**, um modelo de fluxo de dados e temporização de um circuito.

Existem hoje 3 Linguagens de Descrição de Hardware: VHDL, Verilog e Chisel.

Exemplo de um  
Fluxo de dados  
em VHDL



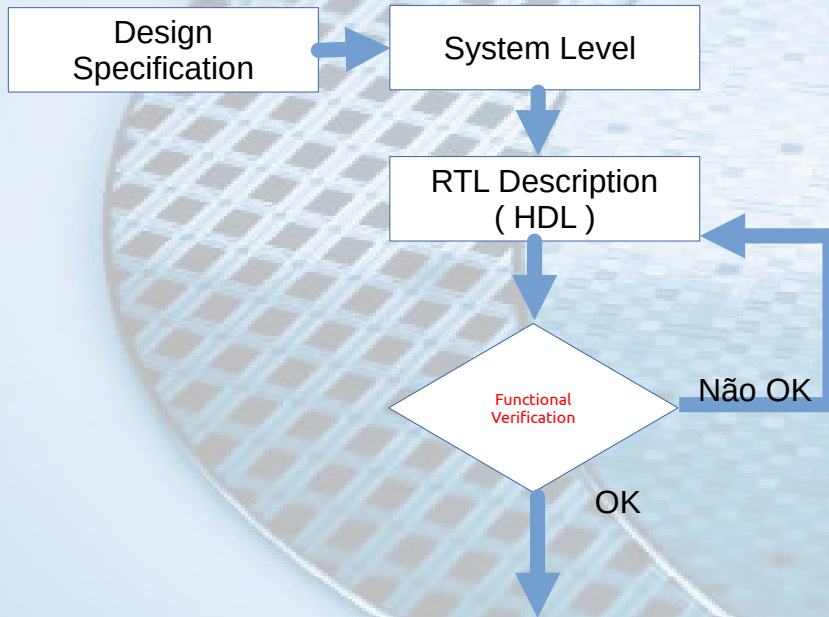
```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY not1 IS
    PORT(
        a : IN  STD_LOGIC;
        b : OUT STD_LOGIC;
    );
END not1;
ARCHITECTURE behavioral OF not1 IS
BEGIN
    b <= NOT a;
END behavioral;
```

# Register Transfer Level - RTL

É uma abstração de Projeto que modela um circuito digital síncrono em termos do fluxo de sinais entre registros de hardware e das operações lógicas realizadas nesses sinais.

Podem ser incorporadas células, Funções e Registradores no arquivo gerado para compor a entrada da próxima etapa que é a **Síntese Lógica**.

Mais adiante veremos que esta etapa é realizada pelo Software Yosys



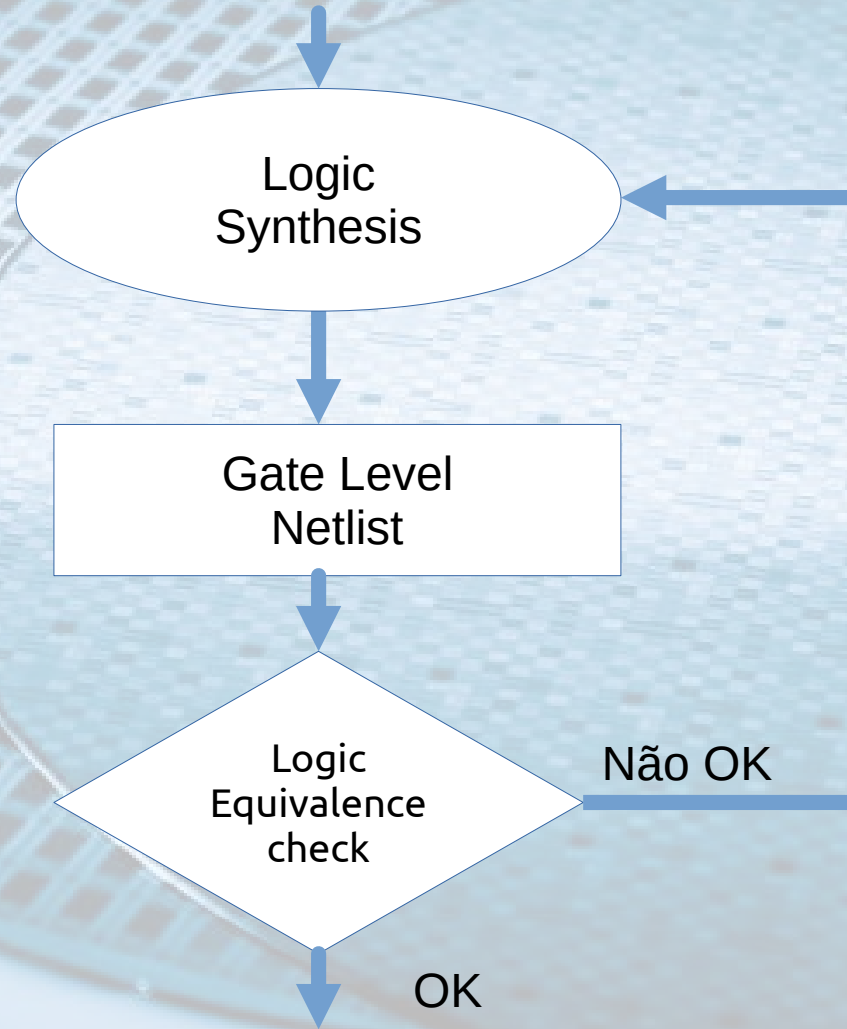
No nível de abstração comportamental, podemos verificar o Comportamento do circuito através de um software de modelagem Comportamental.

Também podemos simular o código e verificar a funcionalidade do Circuito.

Uma vez que o Comportamento e a funcionalidade do código esteja correta e verificada e se nenhum bug for encontrado, este código RTL está indo para o próximo estágio que é o Back-end Design

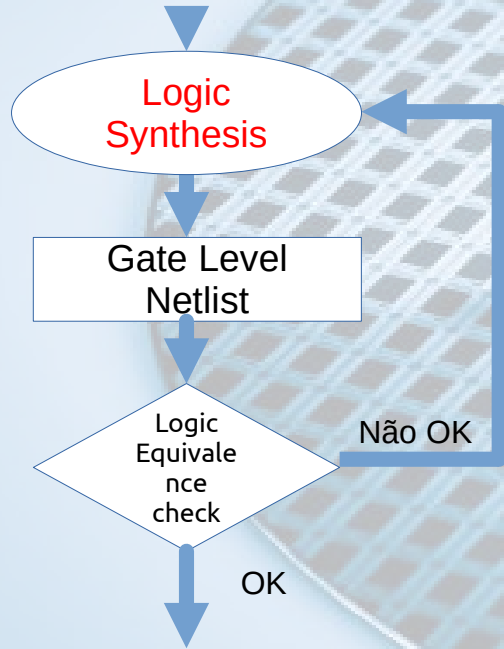
Back-end Design

Logic Design



# Síntese Lógica ( Logic Synthesis )

É o Processo de transformação por meio de um software, de uma **especificação abstrata** do comportamento de um circuito, geralmente em nível de Transferência de Registro ( RTL ), em uma implementação de Portas Lógicas.



Hardware  
Description  
Language

HDL

Logic  
Synthesis

Netlist

FPGA

ASIC

VHDL  
Verilog  
Chisel

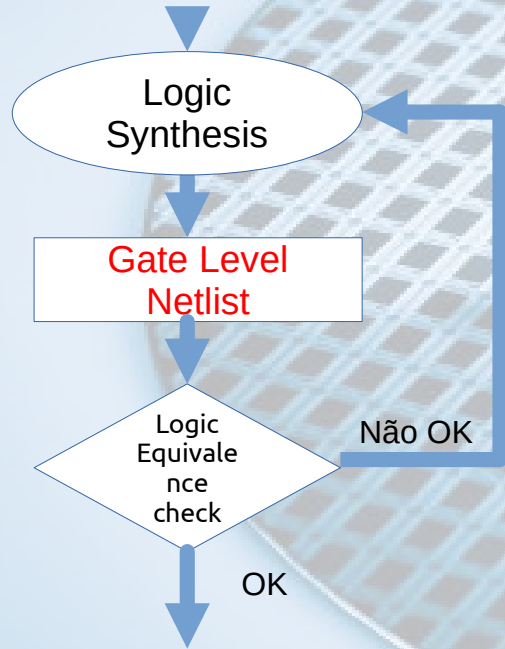
# Nível de Porta Lógica

No nível da porta lógica, o projeto é representado por uma netlist que usa apenas células de um pequeno número de células de bit único, como portas lógicas básicas (AND, OR, NOT, XOR, etc.) e registradores (geralmente Flip-Flop tipo D).

Existem vários formatos de netlist que podem ser usados neste nível, por ex. o Electronic Design Interchange Format (EDIF), mas para facilitar a simulação, muitas vezes é usada uma netlist HDL.

Este último é um arquivo HDL (Verilog ou VHDL) que utiliza apenas as construções de linguagem mais básicas para instanciação e conexão de células.

Veremos mais adiante que nesta etapa de síntese Lógica é usado o Software ABC

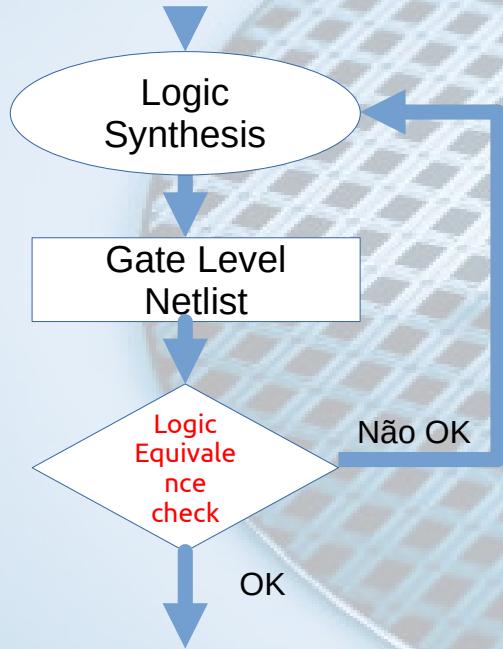


Também é conhecido como verificação formal.

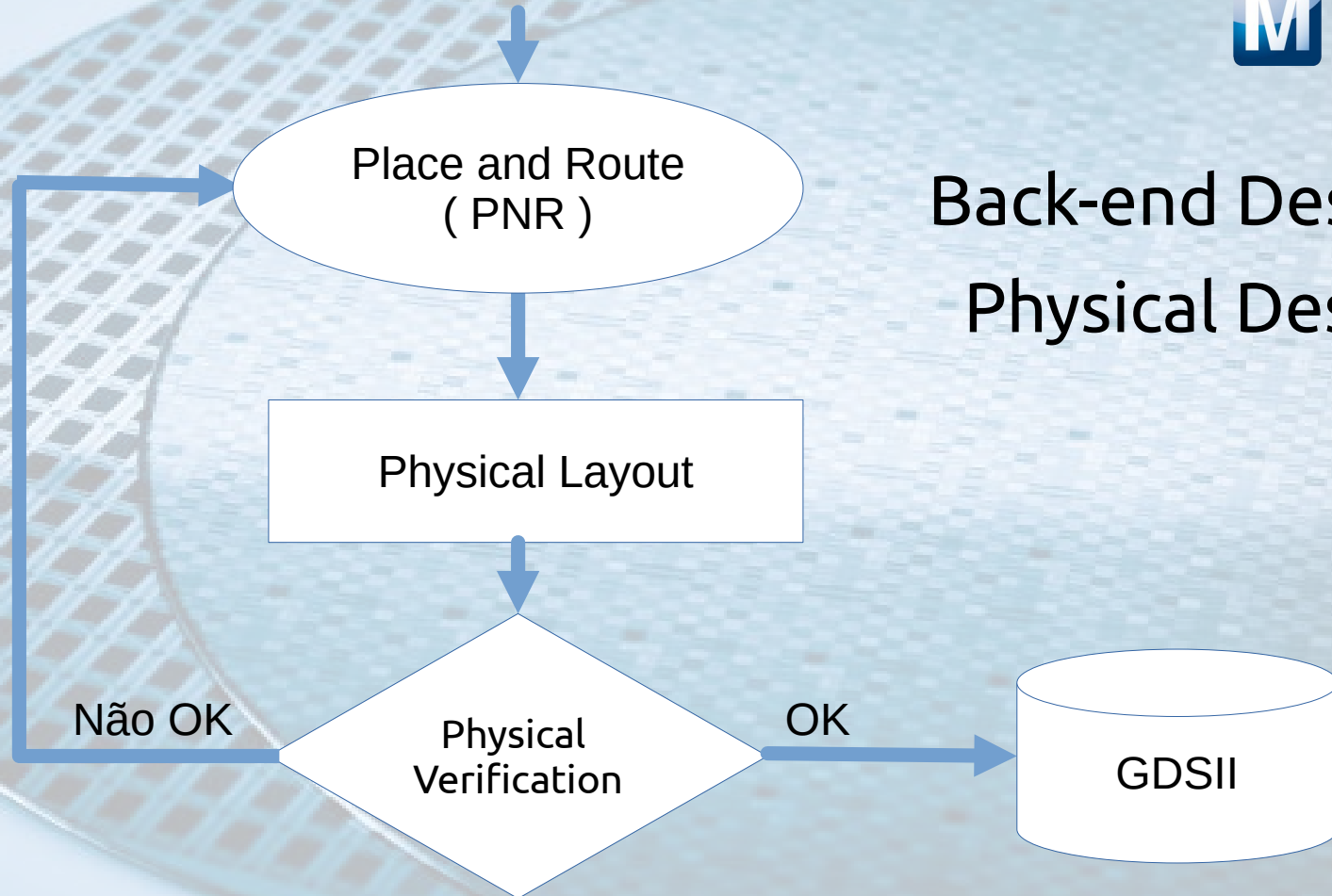
A verificação Formal utiliza métodos matemáticos para comparar a lógica projetada com a que está sendo verificada em relação a um detalhe lógico ou uma estrutura de referência.

Em contraste com o check por meio de simulação, a verificação formal não requer vetores de entrada.

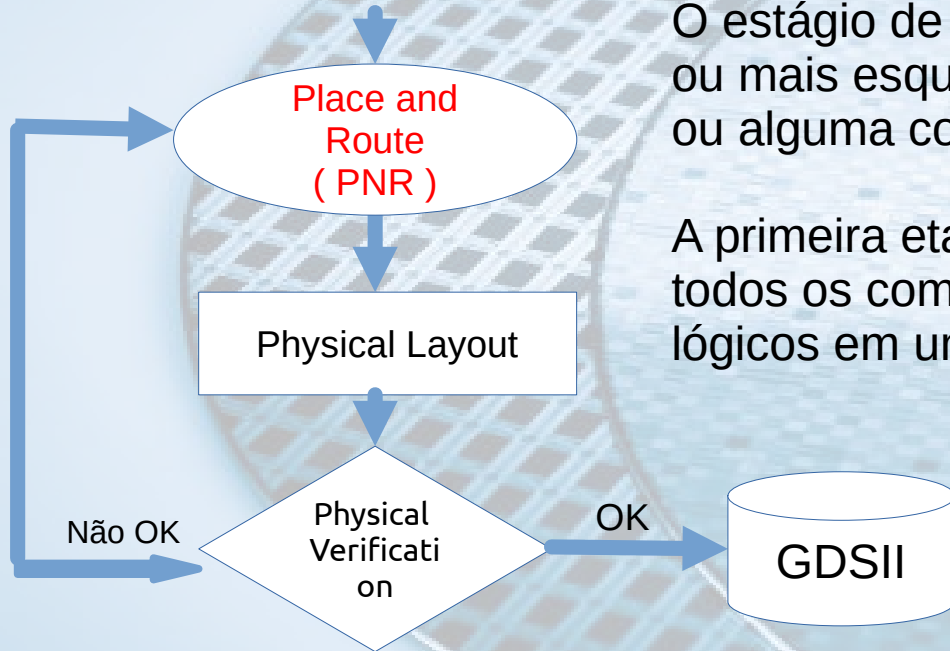
Como verificação formal pensa apenas em funções lógicas em meio a correlações, é independente das propriedades físicas do plano, por exemplo, layout e tempo.



# Back-end Design Physical Design



# Posicionamento e Roteamento ( Place and Route – PnD )



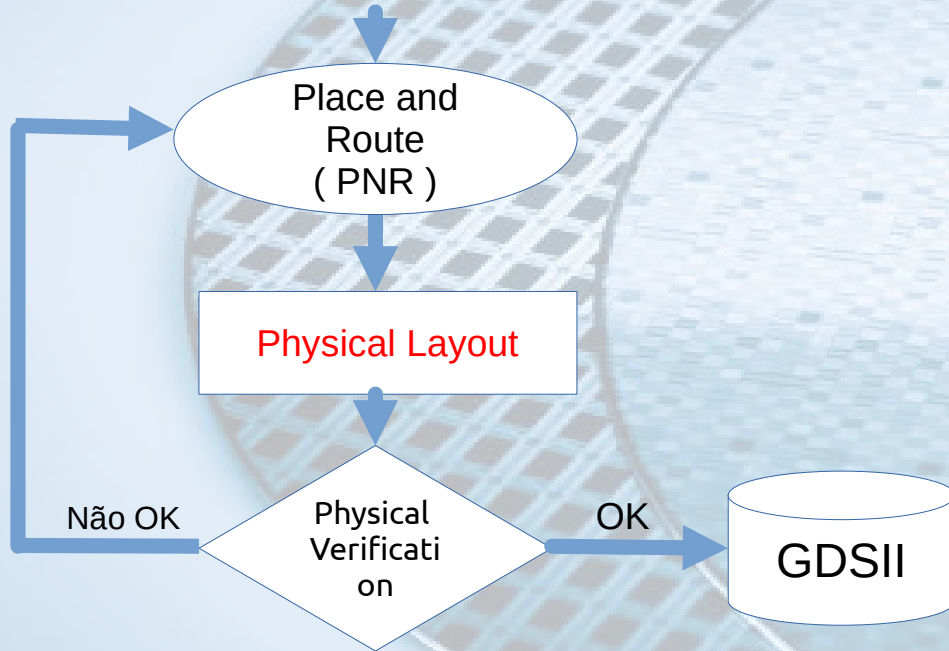
O estágio de Posicionamento e Roteamento começa com um ou mais esquemas, arquivos HDL ou núcleos IP pré-roteados, ou alguma combinação de todos os três.

A primeira etapa, posicionamento, envolve decidir onde colocar todos os componentes eletrônicos, circuitos e elementos lógicos em uma área limitada de espaço.

É seguido pelo roteamento, que decide o design exato de todos as conexões necessários para conectar os componentes posicionados na primeira etapa, seguindo as regras e limitações do processo de fabricação.

No final do Processo, se produz um layout que é automaticamente convertido em uma máscara no formato GDS II padrão ou OASIS.

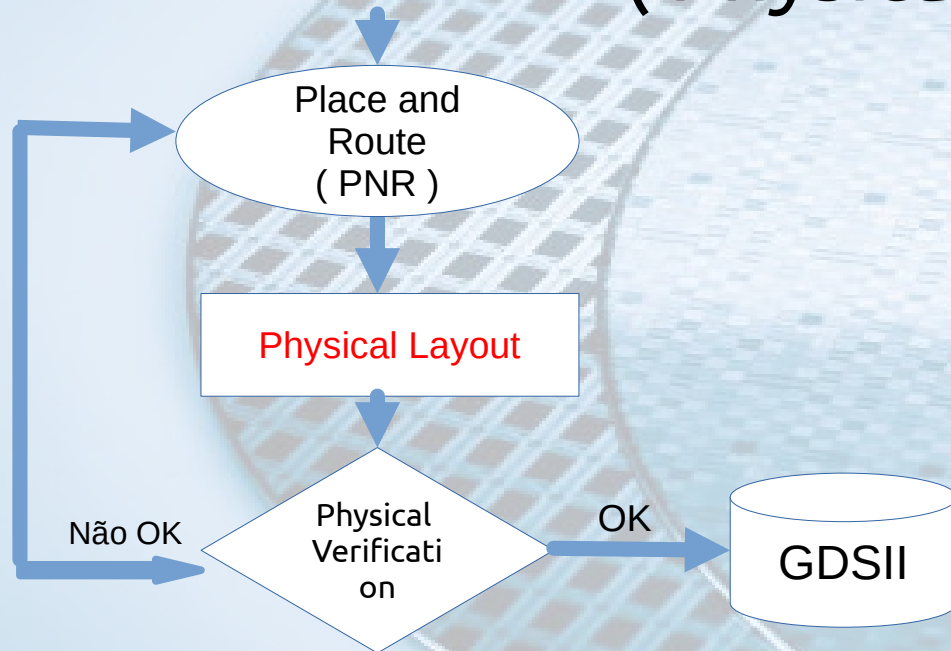
# Nível Porta Física ( Physical Gate Level ) 1



No nível da porta física, apenas são usadas portas que estão fisicamente disponíveis na arquitetura alvo.

Em alguns casos, podem ser apenas portas NAND, NOR e NOT, bem como registros do tipo D. Em outros casos, isso pode incluir células que são mais complexas do que as células usadas no nível da porta lógica (por exemplo, meio-somadores completos).

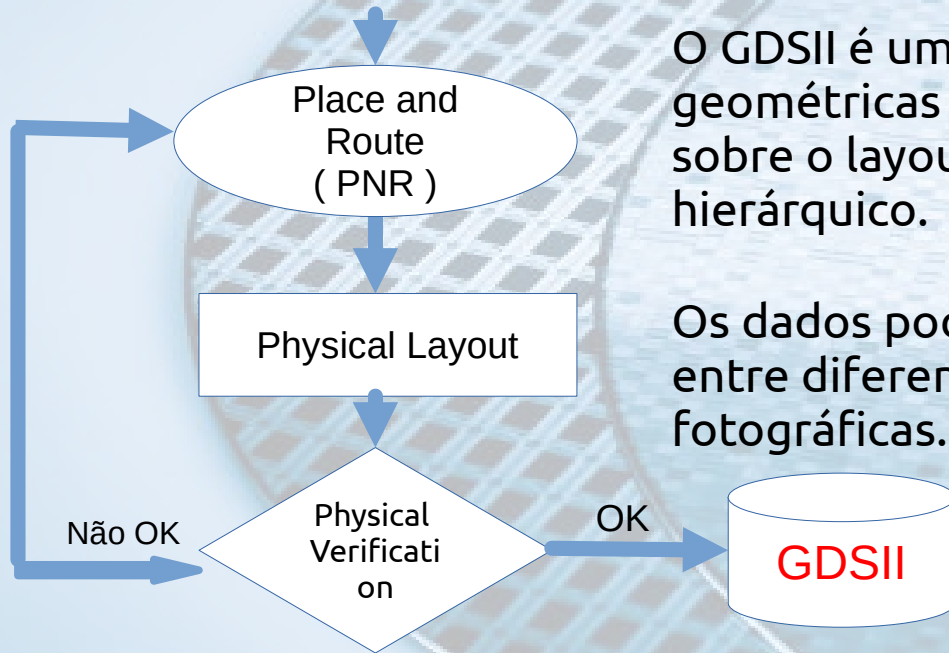
# Nível Porta Física ( Physical Gate Level ) 2



No caso de um projeto baseado em FPGA, a representação do nível da porta física é uma netlist de LUTs com registradores de saída opcionais, pois estes são os blocos de construção básicos das células lógicas do FPGA.

Para a cadeia de ferramentas de síntese esta abstração é geralmente o nível mais baixo. No caso de um projeto baseado em ASIC, a biblioteca de células pode conter informações adicionais sobre como as células físicas são mapeadas para comutadores individuais (transistores).

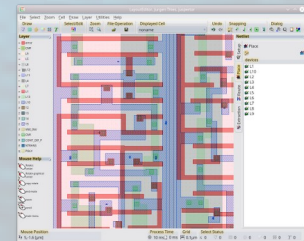
# ( Graphic Data Stream Information Interchange



O GDSII é um formato de arquivo binário que representa formas geométricas planas, rótulos de texto e outras informações sobre o layout de um Circuito Integrado em formato hierárquico.

Os dados podem ser usados na transferência de artes finais entre diferentes ferramentas ou na criação de máscaras fotográficas.

Em 2008, muitos fornecedores de software EDA mudaram para o formato de fluxo OASIS, que substituiu o GDSII.





TOUCAN  
eLAB



# Open Source Software Tools

## Front-end

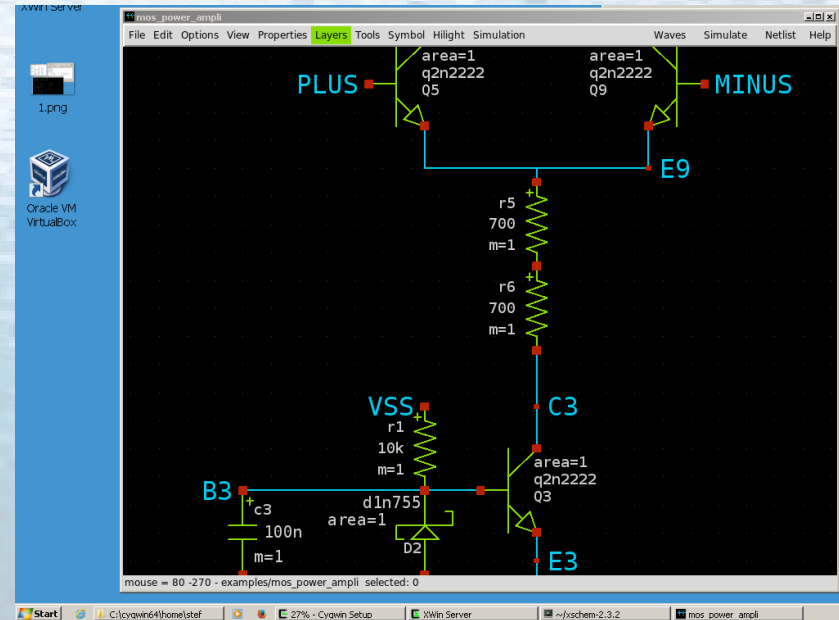


Dentre vários Open Source Softwares disponíveis , destacamos:

- XSCHEM: Schematic Capture and Netlisting EDA tool
- Ícarus Verilog : Implementation of the Verilog HDL
- NGSPICE : Mixed Mode- Mixed Level Circuit Simulator
- YOSYS : Logic Synthesis Suite
- ABC : A System for Sequential Synthesis and Verification

## Schematic circuit editor for VLSI and Mixed mode circuit simulation

Xschem é um programa de captura esquemática que permite criar uma representação hierárquica de circuitos com uma abordagem top down.



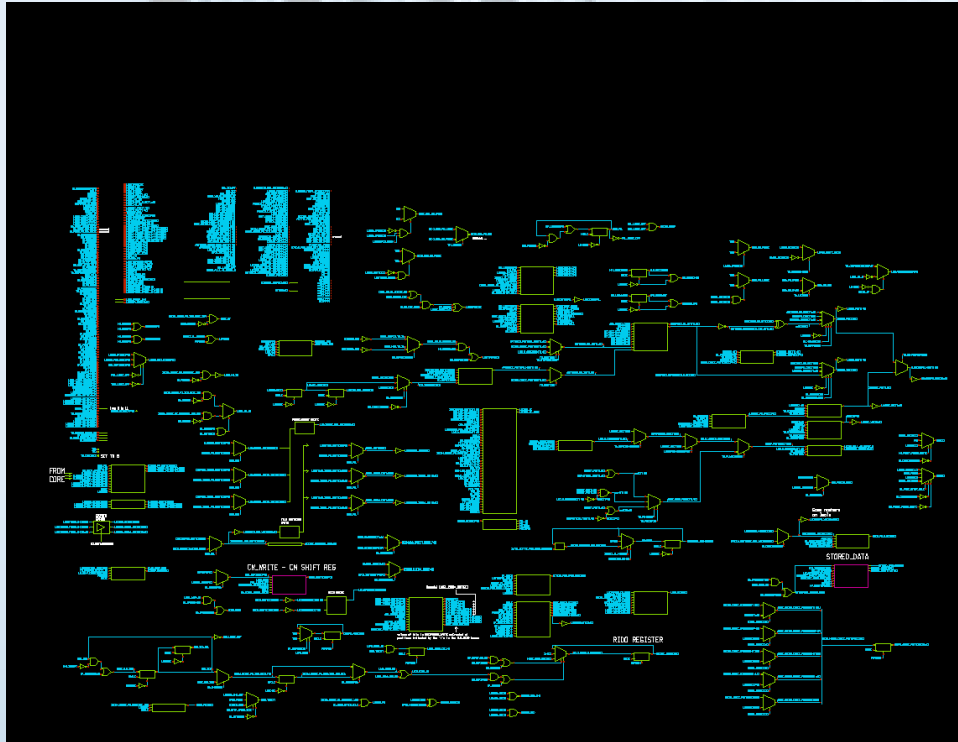


TOUCAN  
eLAB

# XSCHEM



Schematic circuit editor for VLSI and Mixed mode circuit simulation.



Pode ser descrito em termos de blocos de construção mais simples. Focando nas interconexões, hierarquia e propriedades de um sistema.

Uma netlist VHDL, Verilog ou Spice pode ser gerada a partir do esquemático desenhado, permitindo a simulação do circuito.

## Schematic circuit editor for VLSI and Mixed mode circuit simulation

Os esquemas podem ser impressos nos formatos SVG, PNG, PDF.

XSCHEM roda em Linux ou outros semelhantes ao Unix com servidor Xorg e em Windows com a camada Cygwi

Pode ser encontrado na WEB em:



<https://sourceforge.net/projects/xschem/>



<https://github.com/StefanSchippers/xschem>



TOUCAN  
eLAB



# Ícarus Verilog



Icarus Verilog é uma implementação do compilador de linguagem de descrição de hardware Verilog que gera netlists no formato desejado (EDIF). Ele suporta as versões 1995, 2001 e 2005 do padrão, partes do SystemVerilog e algumas extensões.

Icarus Verilog está disponível para Linux, FreeBSD, OpenSolaris, AIX, Microsoft Windows e Mac OS X. Lançado sob a Licença Pública Geral GNU, Icarus Verilog é um software livre.



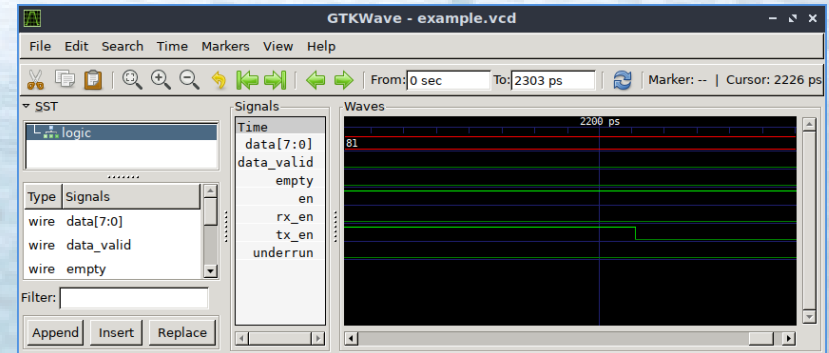
# Ícarus Verilog

O Icarus é composto por um compilador Verilog (incluindo um pré-processador Verilog) com suporte para backends de plug-ins e uma máquina virtual que simula o design.

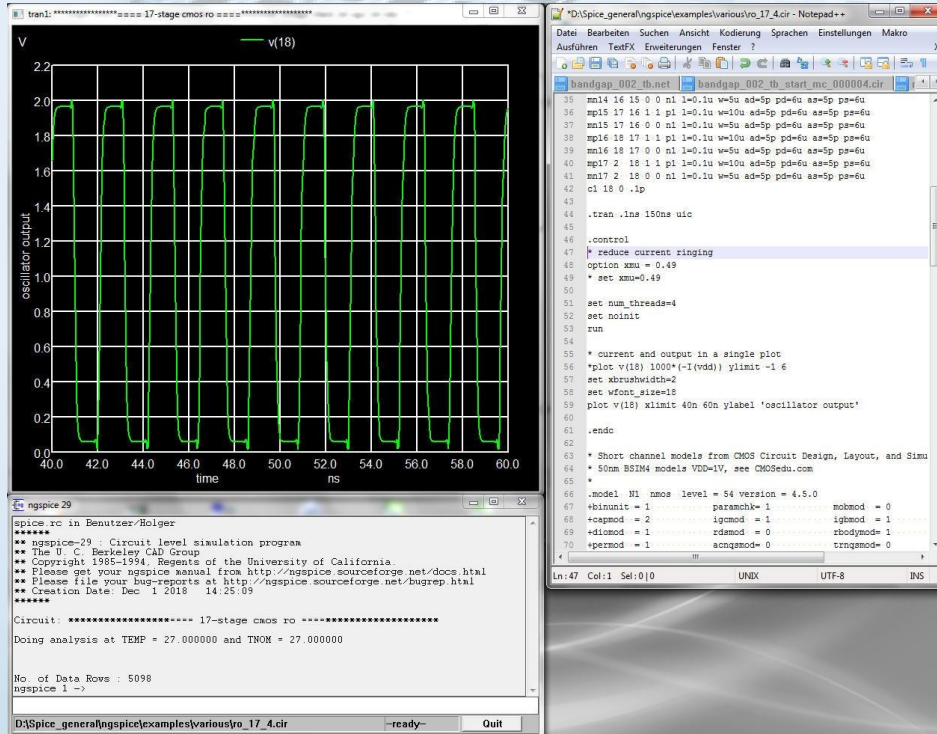
O Icarus possui uma interface para o GTKWave.

Ele é um visualizador de formas de onda VCD baseado na biblioteca GTK.

Pode ser encontrado na WEB em:



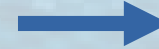
<https://github.com/steveicarus/iverilog>



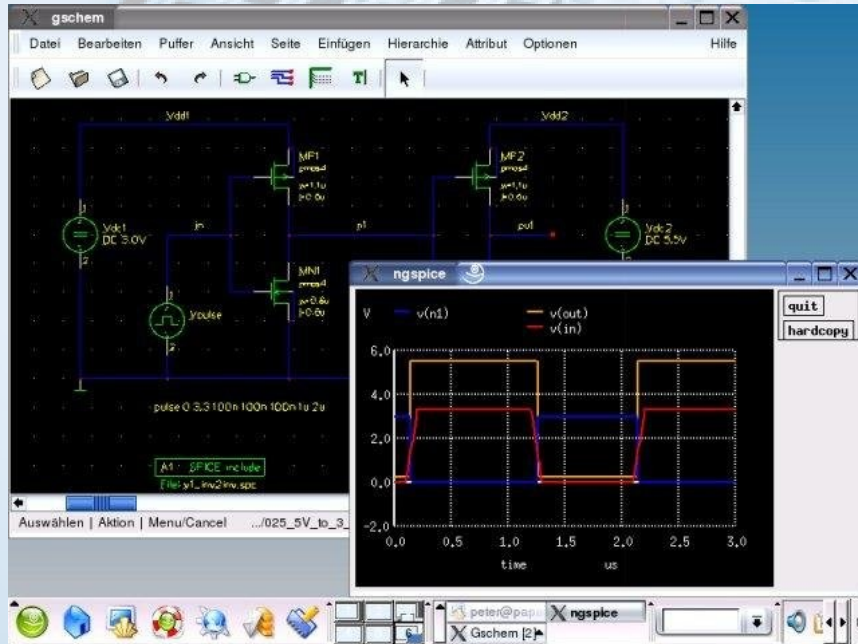
NGSPICE é um simulador spice open source para circuitos elétricos e eletrônicos.

A entrada dos circuitos é feita como uma netlist, e a saída é um ou mais gráficos de correntes, tensões e outras grandezas elétricas.

XSCHEM



Os resultados também poderão ser salvos em um arquivo de dados.



Oferece diversos modelos de dispositivos para elementos ativos, passivos, analógicos e digitais. Os parâmetros do modelo são fornecidos por bibliotecas, pelos fabricantes de dispositivos semicondutores ou por Foundries.

Pode ser encontrado na WEB em:

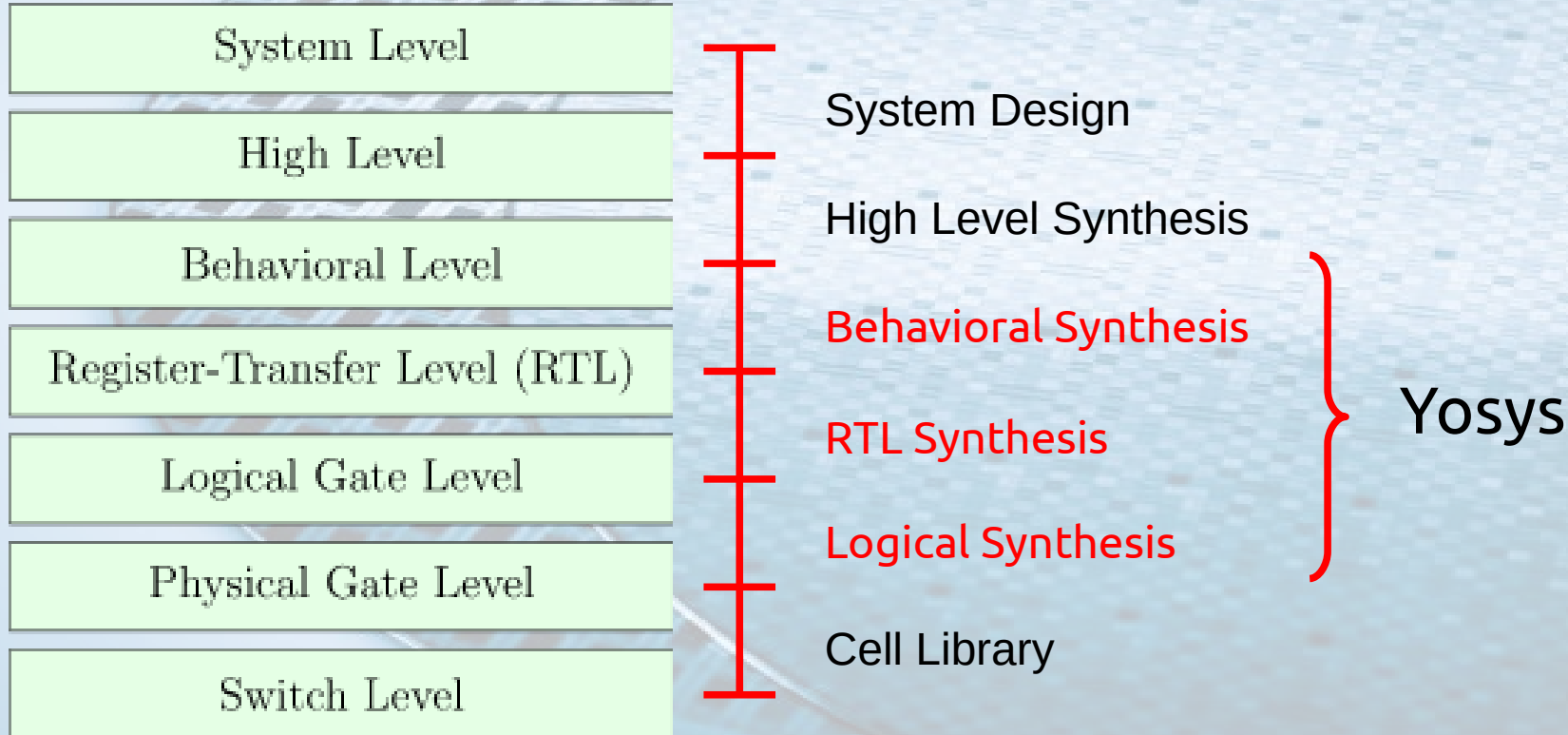


<https://sourceforge.net/projects/ngspice/>

Yosys é uma estrutura para síntese Verilog RTL fornecendo um conjunto básico de algoritmos de síntese para vários domínios de aplicação.

Recursos selecionados e aplicações típicas:

- Processa praticamente qualquer projeto Verilog-2005 sintetizável
- Convertendo Verilog para BLIF/EDIF/BTOR/SMT-LIB/RTL simples Verilog/etc.
- Métodos formais integrados para verificar propriedades e equivalência
- Mapeamento para bibliotecas de células padrão ASIC (em Liberty File Format)
- Mapeamento para FPGAs : Xilinx Série 7 e Lattice iCE40 e ECP5
- Fundação e/ou front-end para fluxos personalizados





```
# read design
read_verilog mydesign.v

# elaborate design hierarchy
hierarchy -check -top mytop

# the high-level stuff
proc; opt; fsm; opt; memory; opt

# mapping to internal cell library
techmap; opt

# mapping flip-flops to mycells.lib
dfflibmap -liberty mycells.lib

# mapping logic to mycells.lib
abc -liberty mycells.lib

# cleanup
Clean

# write synthesized design
write_verilog synth.v
```

Yosys é controlado usando scripts de síntese.

O script de síntese Yosys lê um design (com o módulo superior mytop) do arquivo verilog mydesign.v

sintetiza-o em uma netlist de nível de porta usando a biblioteca de células no arquivo Liberty mycells.lib

grava os resultados sintetizados como netlist Verilog para synth.v:



## Simple Example Verilog Design (Counter.v)

```

module counter (clk, rst, en, count);
  input clk, rst, en;
  output reg [3:0] count;

  always @(posedge clk)
    if (rst)
      count <= 4'd0;
    else if (en)
      count <= count + 4'd1;
endmodule

```

↓ Yosys

```

# read design
read_verilog counter.v
hierarchy -check

```

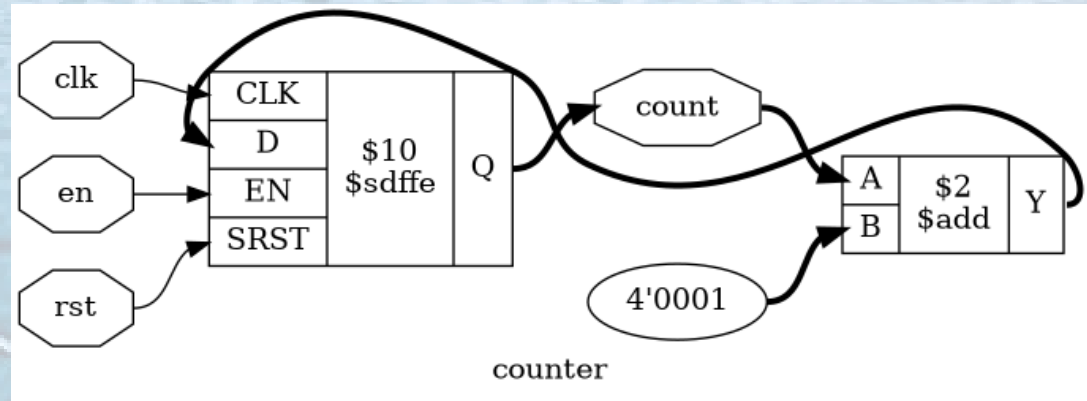
```

# high-level synthesis
proc; opt; fsm; opt; memory; opt

```

Exemplo de um simples script de síntese Yosys e a saída dos comandos "show" para os projeto sintetizados.

O comando "show" está usando o software GraphViz para gerar esquemas.



RTL Netlist with Coarse-Grain Cell Matches.



TOUCAN  
eLAB

# ABC



## A System for Sequential Synthesis and Verification

É um sistema de software para síntese e verificação de circuitos lógicos sequenciais binários que aparecem em projetos de hardware síncronos.

ABC combina otimização lógica escalável baseada em And-Inverter Graphs (AIGs), mapeamento de tecnologia baseada em DAG com atraso ideal para tabelas de consulta e células padrão e algoritmos inovadores para síntese e verificação sequencial.

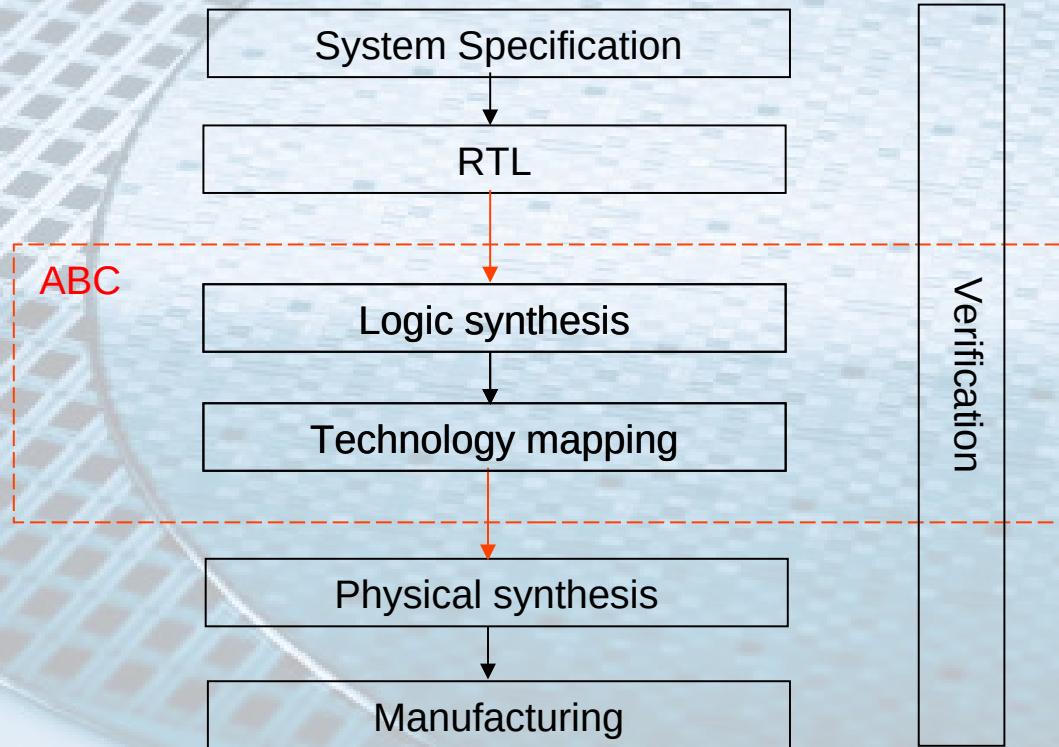
O código ABC mais recente pode ser encontrado em :



**GitHub**

<https://github.com/berkeley-abc/abc>

# ABC – Logic Synthesis from Gate-Level



# Open Source Software Tools

## Back-end

# Open Source Softwares Back-end

Dentre vários Open Source Softwares disponíveis , destacamos:

- **Magic VLSI** : Editor de layout VLSI, extração e ferramenta DRC.
- **Netgen** : Ferramenta de comparação de netlist de circuitos (LVS) e de conversão de netlist.
- **XCircuit** : Ferramenta de desenho de circuito e captura esquemática.
- **GrayWolf** : Utilitário usado para realizar o floor planing e o Placement
- **Qrouter** : Ferramenta para gerar camadas metálicas e vias para conectar fisicamente uma netlist
- **IRSIM** : Simulador de circuito digital em nível de switch.
- **OpenTimer** : Ferramenta de análise de temporização estática (STA)
- **Open PDKs** : Instalador PDK para ferramentas Open Sorce EDA
- **Klayout** : Visualizador de Arquivos GDS e OASIS

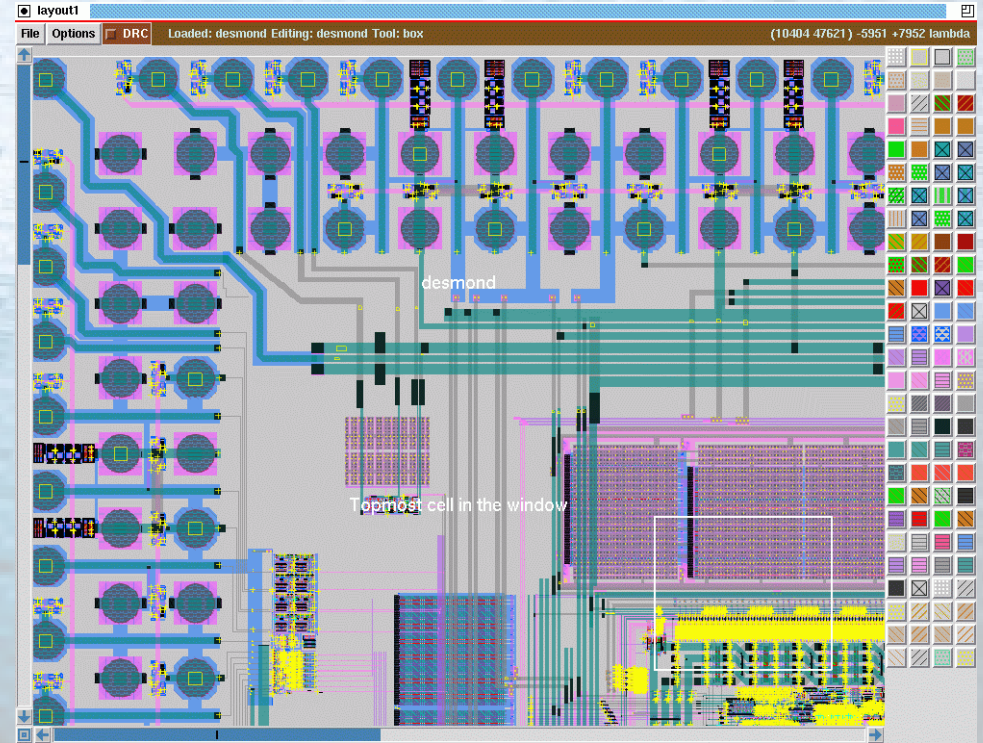
Magic é uma ferramenta de layout VLSI, escrita na década de 1980 em Berkeley por John Ousterhout.

É baseado na linguagem de programação Tcl ( Tool Command Language ) criada pelo próprio autor de Magic.

O código Magic mais recente pode ser encontrado em :



<https://github.com/RTimothyEdwards/magic>

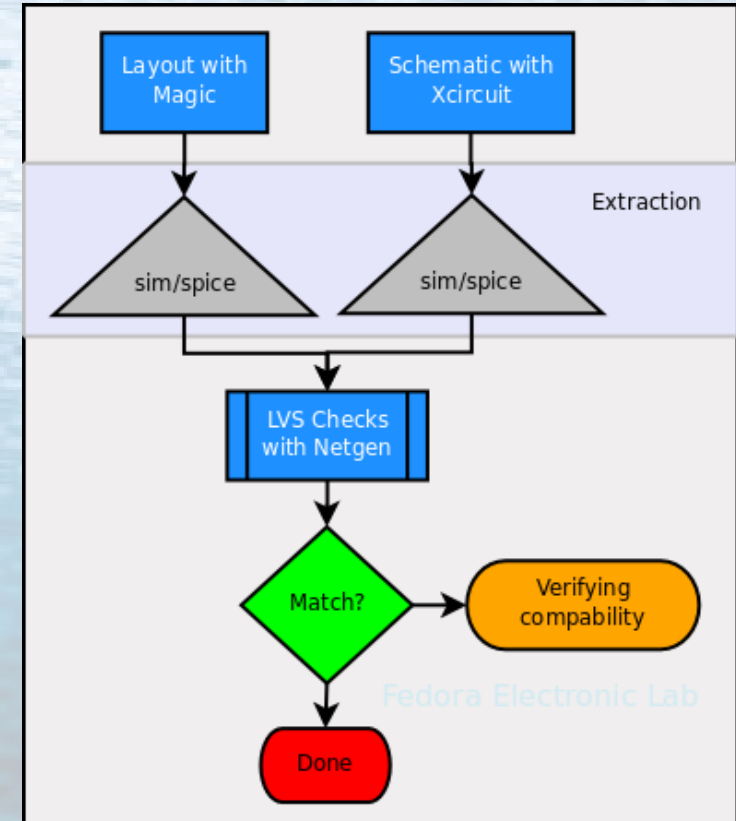




## netlist comparison (LVS) and format manipulation

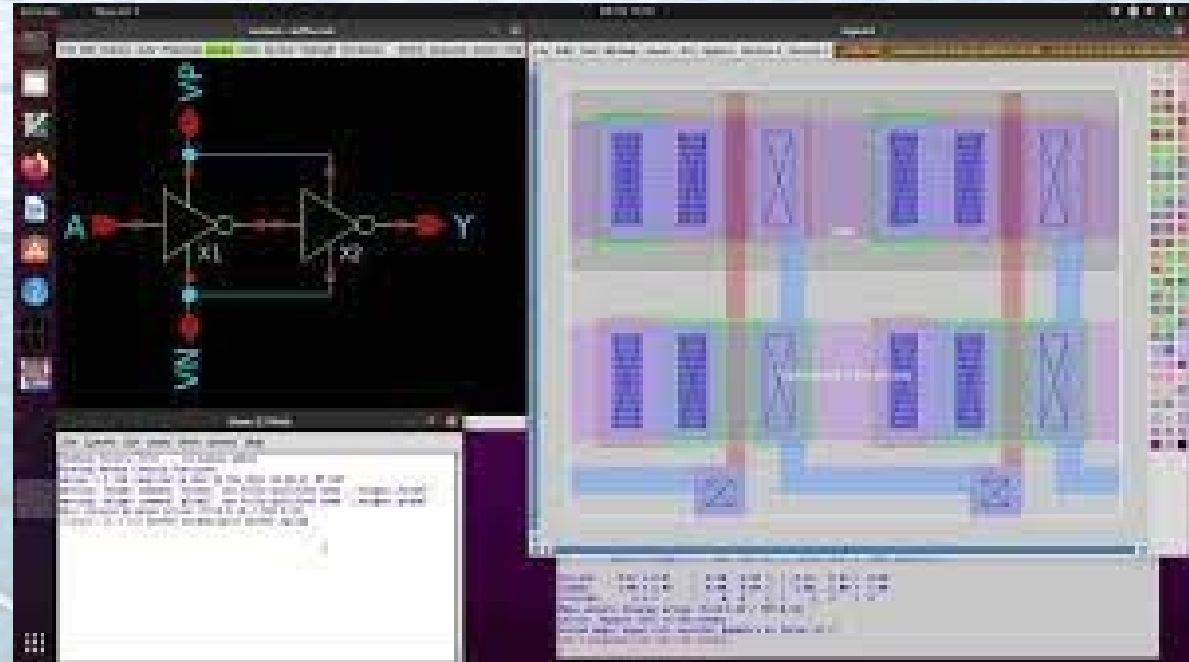
Netgen é uma ferramenta para comparação de netlists, processo conhecido como **LVS**, que significa "**Layout vs. Schematic**".

Este é um passo importante no fluxo do projeto do circuito integrado, garantindo que a geometria definida corresponda ao circuito esperado.

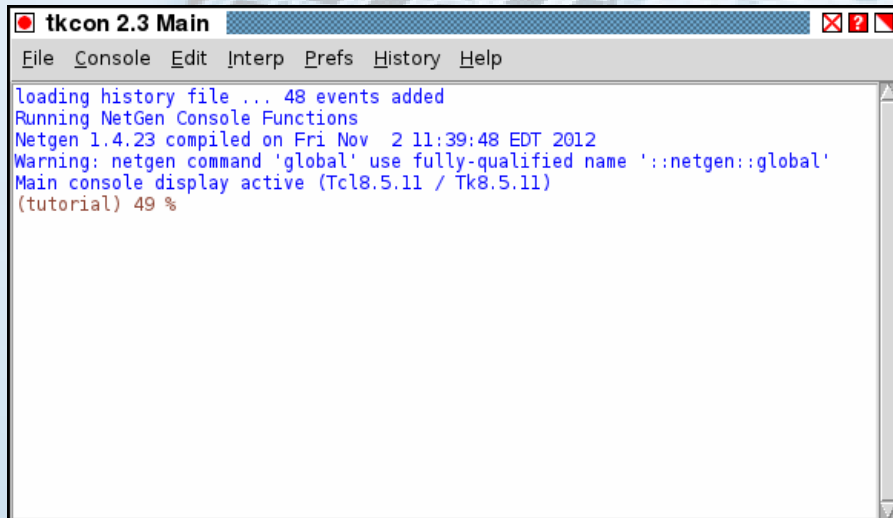


## netlist comparison (LVS) and format manipulation

Circuitos muito pequenos podem contornar esta etapa confirmando a operação do circuito por meio de extração e simulação, porem o LVS pode ser feito muito mais rápido que a simulação e fornece feedback que torna mais fácil encontrar um erro do que uma simulação.



## netlist comparison (LVS) and format manipulation



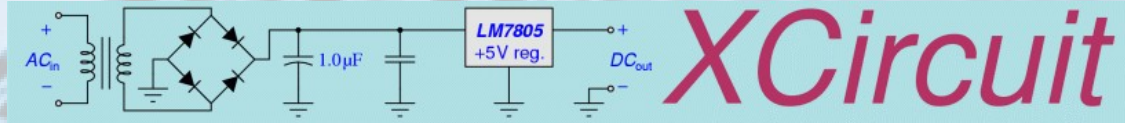
```
tkcon 2.3 Main
File Console Edit Interp Prefs History Help
loading history file ... 48 events added
Running NetGen Console Functions
Netgen 1.4.23 compiled on Fri Nov  2 11:39:48 EDT 2012
Warning: netgen command 'global' use fully-qualified name '::netgen::global'
Main console display active (Tcl8.5.11 / Tk8.5.11)
(tutorial) 49 %
```

A maior necessidade de LVS está em grandes circuitos analógicos ou de sinais mistos que não podem ser simulados em tempo razoável.

O código Netgen mais recente pode ser encontrado em :



<https://github.com/RTimothyEdwards/netgen>

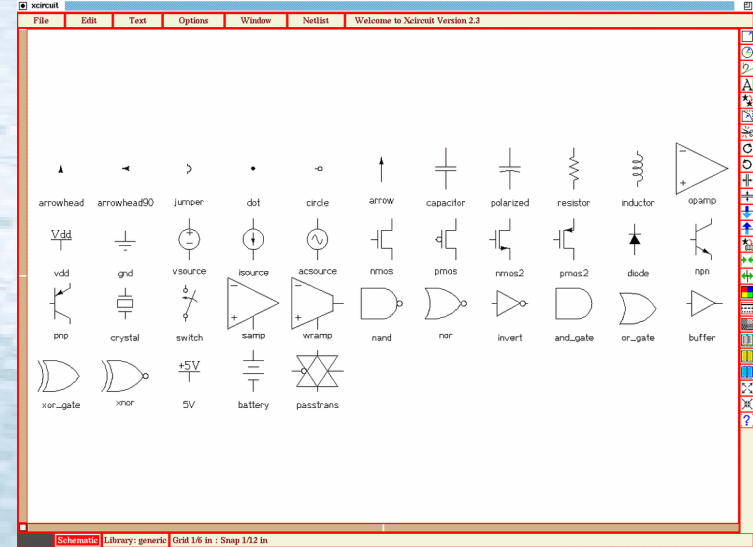
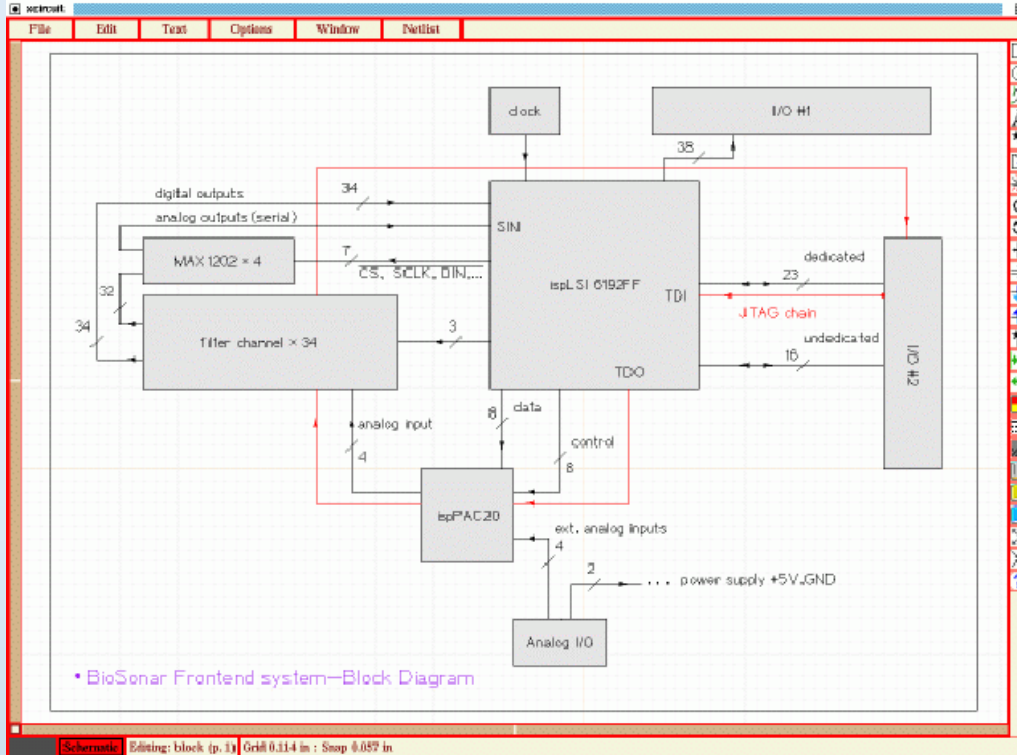
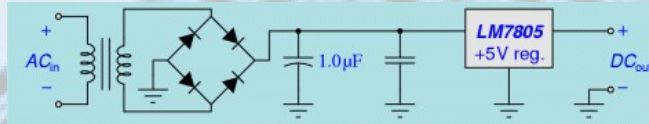


O XCircuit é flexível para ser usado como um programa genérico para desenhar praticamente qualquer coisa.

O resultado de um programa de captura esquemática geralmente não é adequado para publicação, pois muitas vezes nem é legível ou não pode ser redimensionado.

O XCircuit possibilita desenhar diagramas esquemáticos de circuitos com qualidade publicável e figuras relacionadas, e produzir netlists dos circuitos por meio de captura esquemática.

O XCircuit considera os circuitos como inerentemente hierárquicos e grava tanto a saída PostScript hierárquica quanto as netlists SPICE hierárquica



Pode ser encontrado na WEB em:  
<http://opencircuitdesign.com/xcircuit/>

É um utilitário usado para realizar o floor planing e o Placement de projetos digitais VLSI.

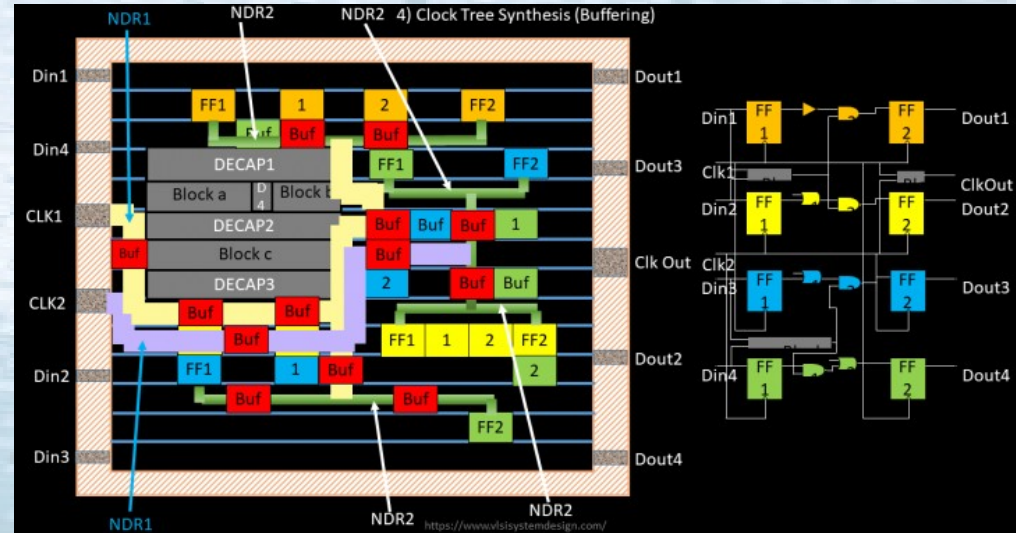
Foi desenvolvido (sob o nome "TimberWolf") na Universidade de Yale e distribuído como código aberto por um tempo até se tornar comercial.

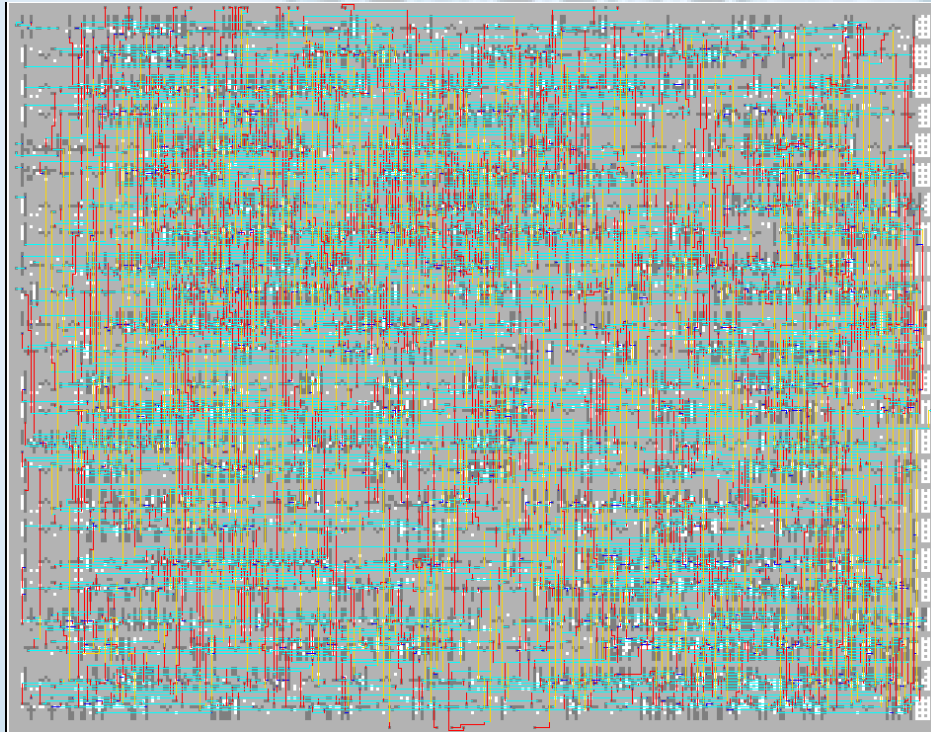
A última versão de código aberto desta ferramenta não realiza roteamento detalhado, mas é uma ferramenta de posicionamento de nível profissional.

É baseado em código do início dos anos 90 e é um dos blocos de construção do Fluxo de Design Digital Open Source qflow.

Ver na WEB : 

// [github.com/rubund/grafwolf](https://github.com/rubund/grafwolf)



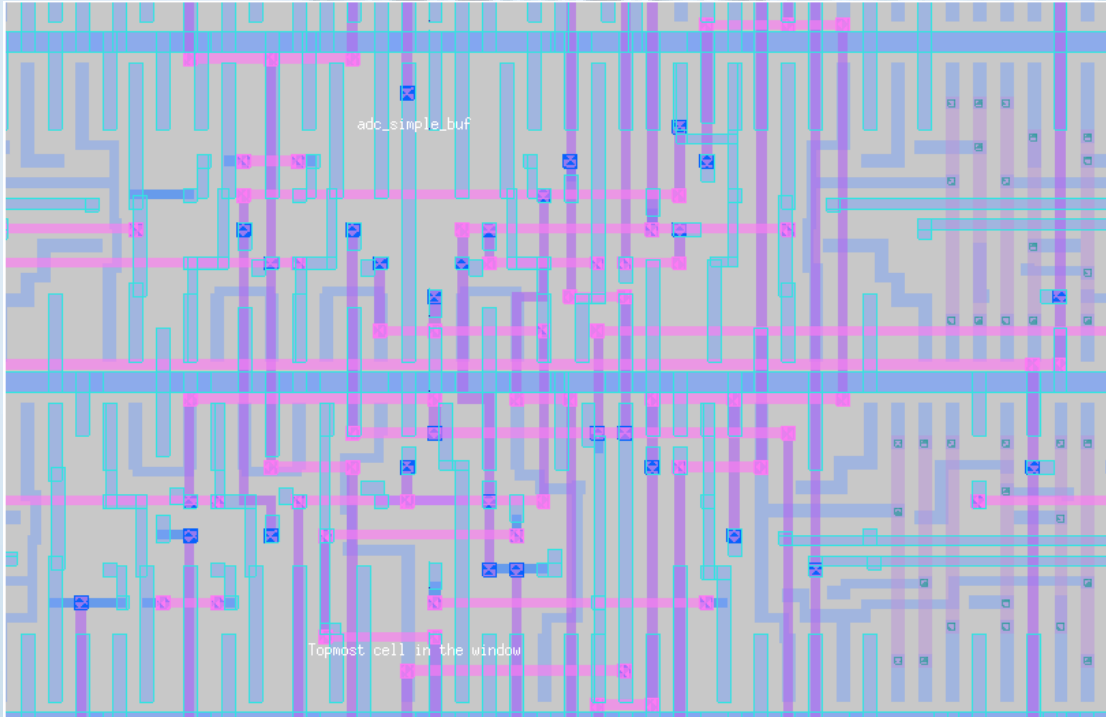


Graphic visualization of routing in qrouter 1.2. The design routed is i2c\_master\_top from Open Cores.

Qrouter é uma ferramenta para gerar camadas metálicas e vias para conectar fisicamente uma netlist em uma tecnologia de fabricação VLSI.

É um roteador "over-the-cell" ou roteador "sea-of-gates".

Inicia com uma descrição de células padrão agrupadas com espaçamento mínimo, e coloca rotas metálicas sobre as células padrão.



O Qrouter usa os formatos padrão aberto LEF e DEF como entrada e saída de arquivos.

Ele pega as definições de células de um arquivo LEF e analisa a geometria de cada célula para determinar pontos de contato e obstruções de rota.

Em seguida, ele lê o posicionamento da célula, o posicionamento dos pinos e a netlist de um arquivo DEF, executa a rota detalhada e grava um arquivo DEF anotado como saída.

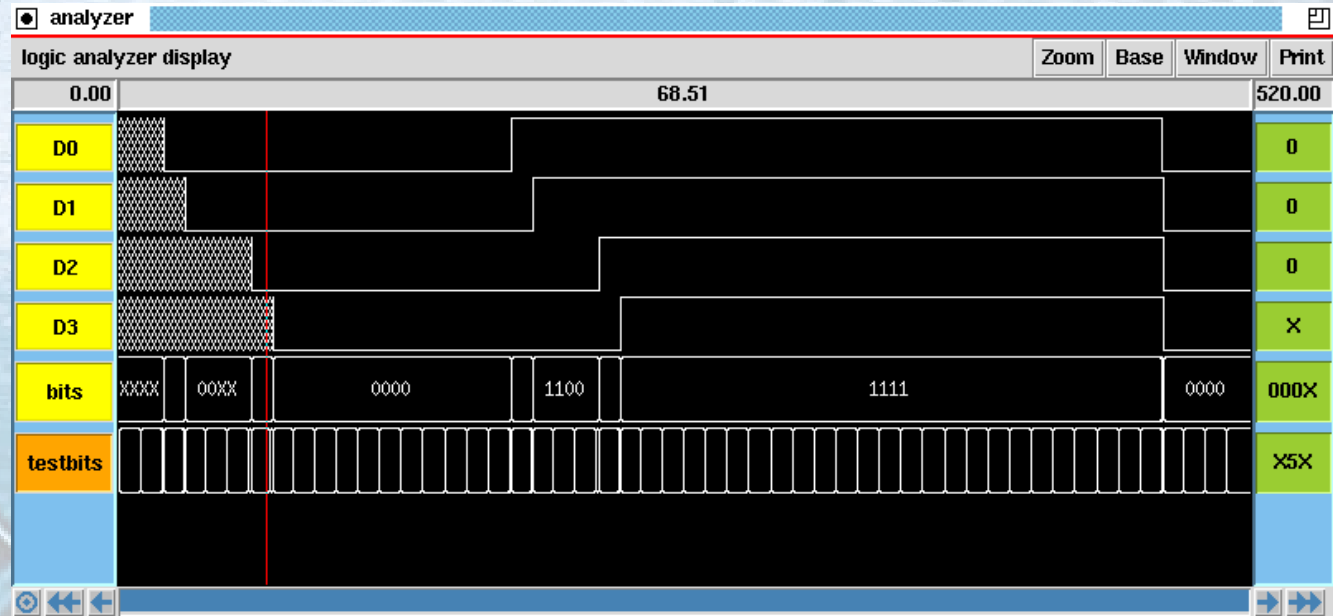
Pode ser encontrado na WEB em:

<http://opencircuitdesign.com/qrouter/>

IRSIM é uma ferramenta para simulação de circuitos digitais.

É um simulador de “nível de switch”; isto é, trata os transistores como chaves ideais.

Os valores extraídos de capacitância e resistência concentrada são usados para tornar a mudança um pouco mais realista do que o ideal, usando as constantes de tempo RC para prever o tempo relativo dos eventos.



```
tkcon 2.3 Main
File Console Edit Interp Prefs History Help
loading history file ... 48 events added
Running IRSIM Console Functions
Warning: irsim command 'time' use fully-qualified name '::irsim::time'
Warning: irsim command 'clear' use fully-qualified name '::irsim::clear'
Warning: irsim command 'q' use fully-qualified name '::irsim::q'
Warning: irsim command 'exit' use fully-qualified name '::irsim::exit'
Starting irsim under Tcl interpreter
IRSIM 9.7.29 compiled on Fri Oct 27 21:18:20 PDT 2006
Warning: Aliasing nodes 'VDD' and 'Vdd'
config9229c: Ignoring lumped-resistance ('R' construct)
Warning: Aliasing nodes 'GND' and 'Gnd'

Read config9229c lambda:0.01u format:MIT
202 nodes; transistors: n-channel=200 p-channel=200
parallel txtors:none
Main console display active (Tcl8.4.8 / Tk8.4.8)
(layout) 49 %
```

The IRSIM command console in the Tcl/Tk



Pode ser encontrado na WEB em:

<http://opencircuitdesign.com/irsim/>

O IRSIM foi desenvolvido em Stanford, enquanto o Magic foi desenvolvido em Berkeley.

Partes do Magic foram desenvolvidas especialmente para uso com IRSIM, permitindo que o IRSIM execute uma simulação em "background", enquanto exibe informações sobre os valores dos sinais diretamente no layout VLSI.



# OpenTimer

OpenTimer é uma ferramenta de análise de temporização estática (STA) para ajudar os projetistas de IC a verificar rapidamente o tempo do circuito.

Foi desenvolvido em usando C++ para oferecer suporte eficiente à temporização paralela e incremental.

Os principais recursos são:

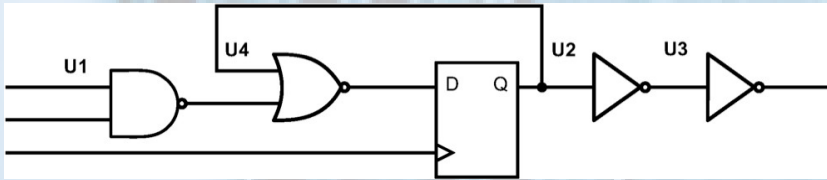
- Suporte ao formato padrão da indústria (.lib, .v, .spef, .sdc)
- Análise de tempo baseada em gráfico e caminho
- Temporização incremental paralela para fechamento rápido da temporização

Pode ser encontrado na WEB em:  <https://github.com/OpenTimer/OpenTimer>



# OpenTimer

```
ot> cd example/simple
ot> read_celllib osu018_stdcells.lib
ot> read_verilog simple.v
ot> read_sdc simple.sdc
```



Report the timing to show the most critical path.

```
ot> report_timing # report the most critical path
Startpoint : inp1
Endpoint   : f1:D
Analysis type : min
```

Type	Delay	Time	Dir	Description
port	0.000	0.000	fall	<b>inp1</b>
pin	0.000	0.000	fall	u1:A (NAND2X1)
pin	2.786	2.786	rise	u1:Y (NAND2X1)
pin	0.000	2.786	rise	u4:A (NOR2X1)
pin	0.181	2.967	fall	u4:Y (NOR2X1)
pin	0.000	2.967	fall	<b>f1:D (DFFNEGX1)</b>
arrival		2.967		data arrival time
related pin	25.000	25.000	fall	f1:CLK (DFFNEGX1)
constraint	1.518	26.518		library hold_falling
required		26.518		data required time
slack		-23.551		VIOLATED

O caminho crítico se origina da entrada primária **inp1** que alimenta o pino de dados **f1:D** do flip-flop **DFFNEGX1**



## PDK Installer for open-source tools

Os PDKs das Silicon Foundry não são padronizados impossibilitando uma configuração de arquivos finais para a manufatura. Além disso, as Silicon Foundry priorizam os usuários de ferramentas EDA comerciais ignorando os entusiastas de ferramentas EDA de código aberto.

Open\_pdks visa mitigar o problema definindo um layout padrão de arquivos e diretórios para formatos padrão abertos conhecidos (por exemplo, SPICE, verilog, liberty, LEF, etc.) e para várias ferramentas EDA de código aberto (por exemplo, magic, netgen, OpenROAD, klayout) usando um sistema Makefile e vários scripts de conversão para garantir que, para qualquer processo, todos os arquivos necessários para todas as ferramentas EDA possam ser encontrados em locais previsíveis.



## PDK Installer for open-source tools

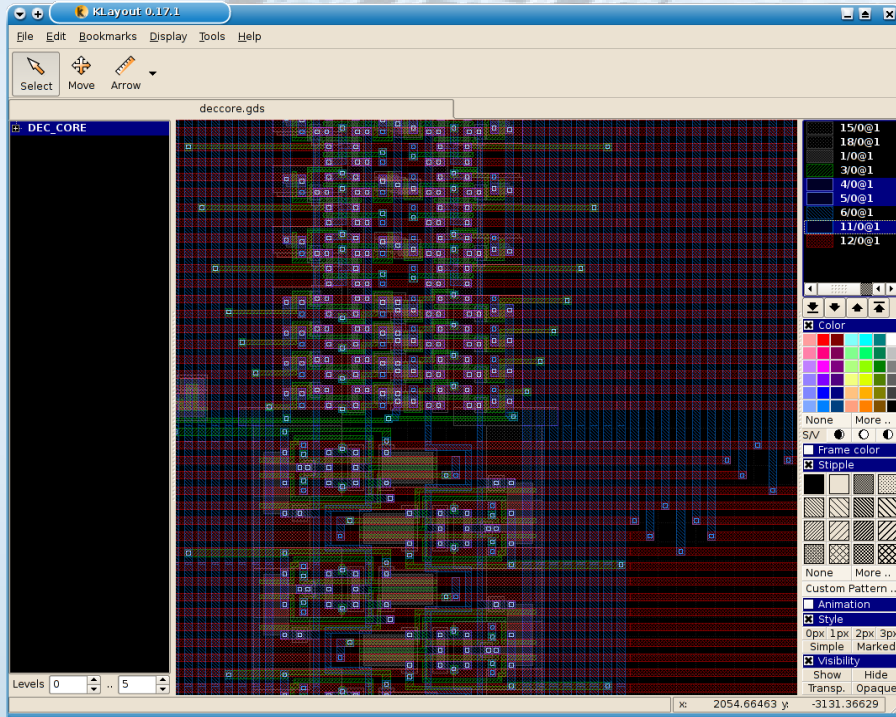
Os scripts pretendem ser o mais genéricos possível para permitir fácil adaptação a novas ferramentas, formatos e fundições. Onde os dados de fundição são intratáveis, arquivos de instalação personalizados podem ser adicionados para substituir ou anotar os dados do fornecedor, conforme necessário.

Open\_PDKs é distribuído com arquivos que suportam a descrição do processo aberto Google/SkyWater sky130 [github.com:google/skywater-pdk](https://github.com/google/skywater-pdk).

Open\_PDKs criará um ambiente para usar o processo SkyWater sky130 com ferramentas EDA de código aberto e fluxos de ferramentas como magic, qflow, openlane, netgen, klayout, etc.

Pode ser encontrado na WEB em:  [https://github.com/RTimothyEdwards/open\\_pdk](https://github.com/RTimothyEdwards/open_pdk)

## Visualizador de Arquivos GDS e OASIS



Suporte aos formatos de arquivo GDS e OASIS com descompactação automática de formatos compatíveis com zlib

Suporte total de 64 bits no Linux

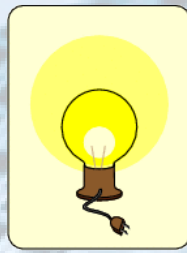
Ambiente de desenvolvimento integrado para scripts Ruby e Python

Pode ser encontrado na WEB em:

<https://www.klayout.de/>



[//github.com/KLayout/klayout](https://github.com/KLayout/klayout)



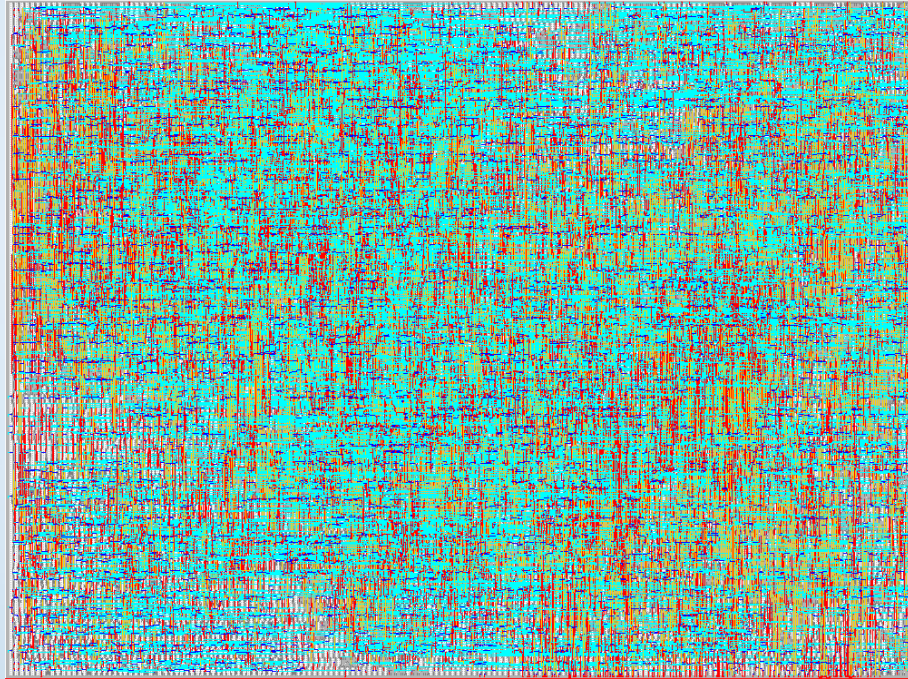
# Open Circuit Design

É um Repositório construído por Tim Edwards que reúne as ferramentas EDA (Electronic Design Automation) Open Source descritas nos slides anteriores.

Ele também apresenta um software Open Source chamado Qflow que é um fluxo completo de Design de síntese digital e bibliotecas de células padrão também Open Source.

O Qflow utiliza os Software:

- Yosys : Framework para Síntese Verilog-RTL
- Graywolf : Placement de projetos digitais VLSI .
- Qrouter : Roteador de detalhes over-the-cell (sea-of-gates).
- Magic : Editor de layout VLSI, extração e ferramenta DRC.
- Netgen : Ferramenta de comparação de netlist de circuitos (LVS) e de conversão de netlist.



The OpenMSP430 microprocessor from OpenCores, placed and routed by qflow.

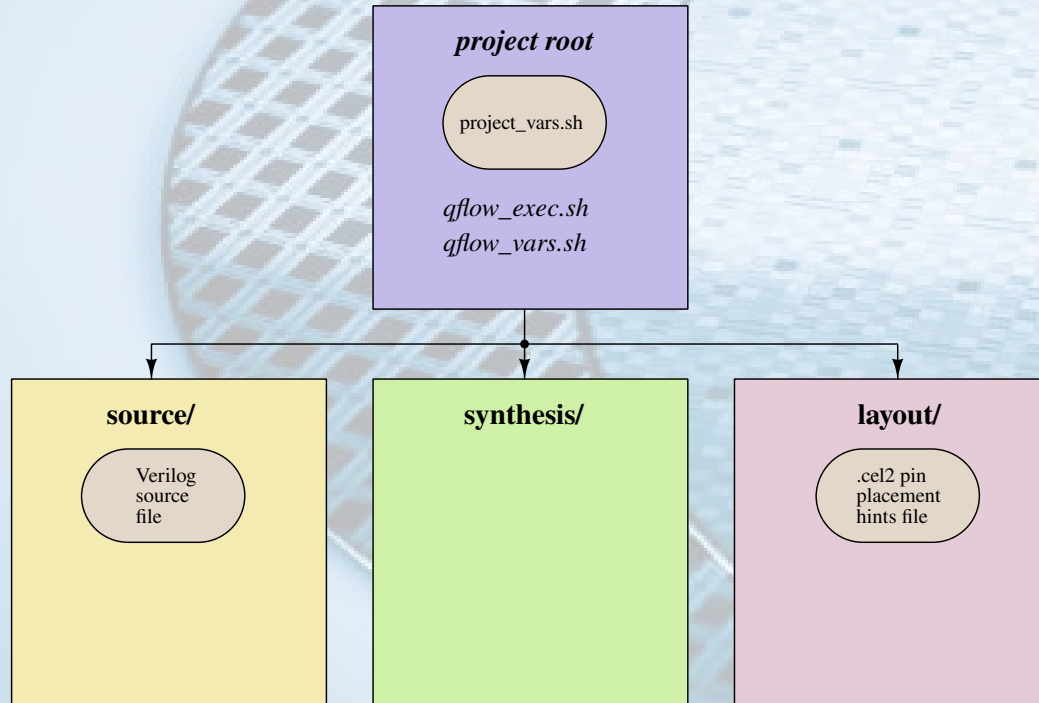
Qflow é um fluxo de síntese digital reunindo um conjunto de ferramentas e métodos usados para transformar um projeto de circuito escrito em uma linguagem comportamental de alto nível, como verilog ou VHDL, em um circuito físico.

O código de configuração gerado pode ser para um FPGA ou um layout de processo de fabricação de um ASIC.

Na WEB em:



<http://www.opencircuitdesign.com/qflow/index.html>



O primeiro passo para sintetizar um circuito é ter um diretório de projeto que será o espaço de trabalho do qflow.

O diretório do **Projeto** tem três subdiretórios, chamados:

**Fonte**: Contém o código-fonte do verilog (extensão .v)

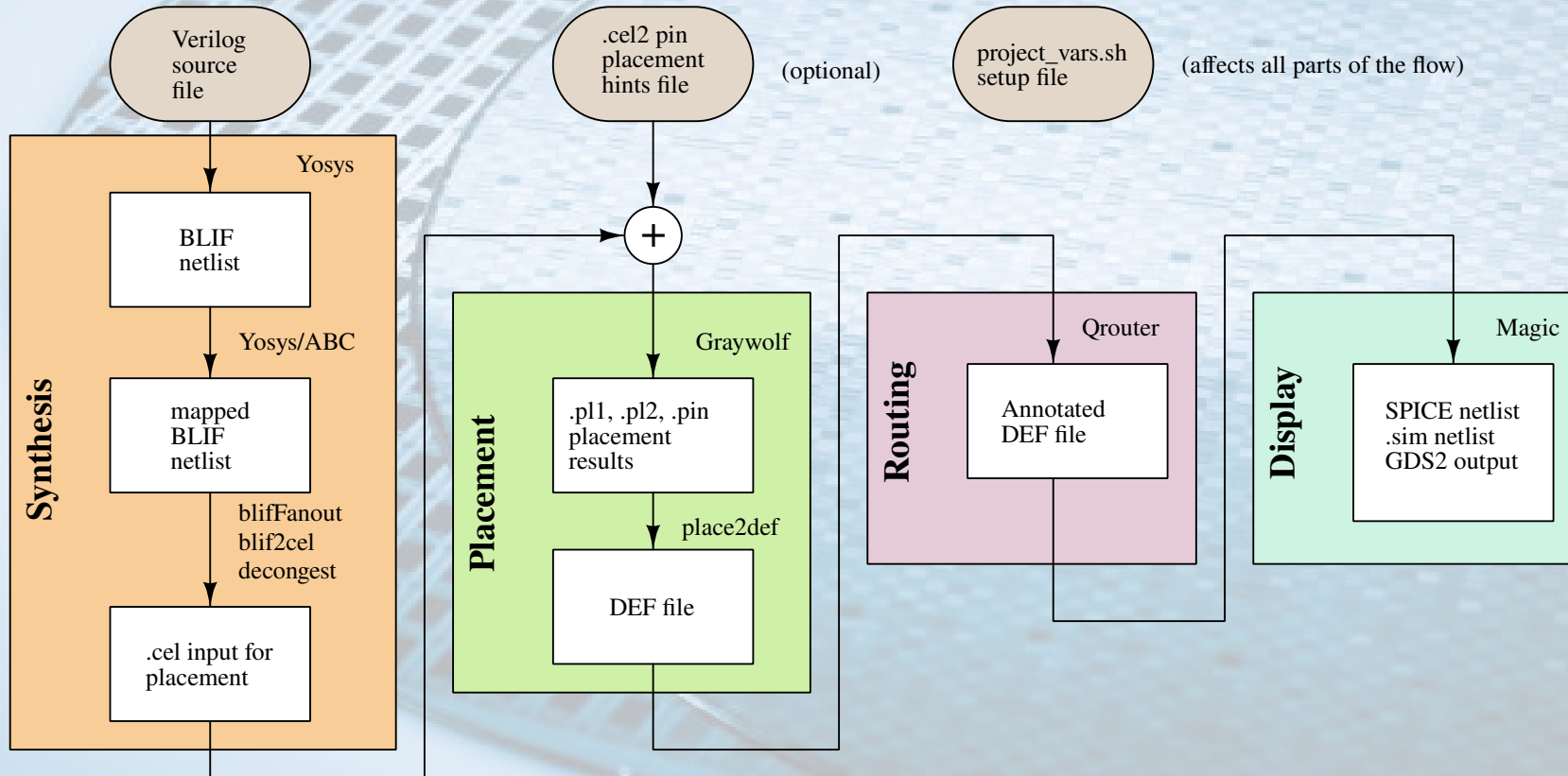
**Síntese**: Contém arquivos de trabalho e saída verilog RTL

**Layout**: Contém arquivos de trabalho e saída de arquivo DEF



# Qflow

An Open-Source Digital Synthesis Flow



# Electronic Design Automation ( EDA ) and Open Source Initiatives

# Automação de Projetos Eletrônicos EDA ( Electronic Design Automation )

O EDA foi usado pela primeira vez na década de 1960 na forma de programas simples para automatizar a localização de um número muito pequeno de blocos em uma placa de circuito.

Com advento do circuito integrado, criou-se a necessidade de um software que pudesse reduzir o número total de portas otimizando o Circuito.

As ferramentas de software atuais consideram também efeitos como atrasos de sinal e acoplamento capacitivo entre conexões adjacentes.

As líderes de Mercado hoje:

 SYNOPSIS® cādence® SIEMENS **Mentor**®  
A Siemens Business



## Electronics Resurgence Initiative

Iniciada em 2018, a iniciativa visa tanto as capacidades de segurança nacional como a competitividade e sustentabilidade econômica comercial.

Esses programas enfatizam parcerias voltadas para o futuro com a indústria dos EUA, a base industrial de defesa e pesquisadores universitários.

Em 2023, a DARPA expandiu o foco da ERI com o anúncio da ERI 2.0 buscando reinventar a fabricação nacional de microeletrônica.



**OPENROAD:  
FOUNDATIONS AND  
REALIZATION OF OPEN  
ACCESSIBLE DESIGN**

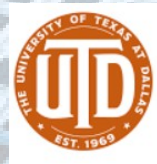


**ERI** ELECTRONICS  
RESURGENCE INITIATIVE  
**SUMMIT**  
2018 | SAN FRANCISCO, CA | JULY 23-25

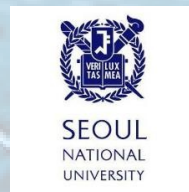
Na ERI 2018 o Projeto OpenROAD foi apresentado pelo Prof. Andrew B. Kahng representando toda equipe da Universidade da California.

















# Varias Instituições participam do Desenvolvimento do Projeto OpenROAD

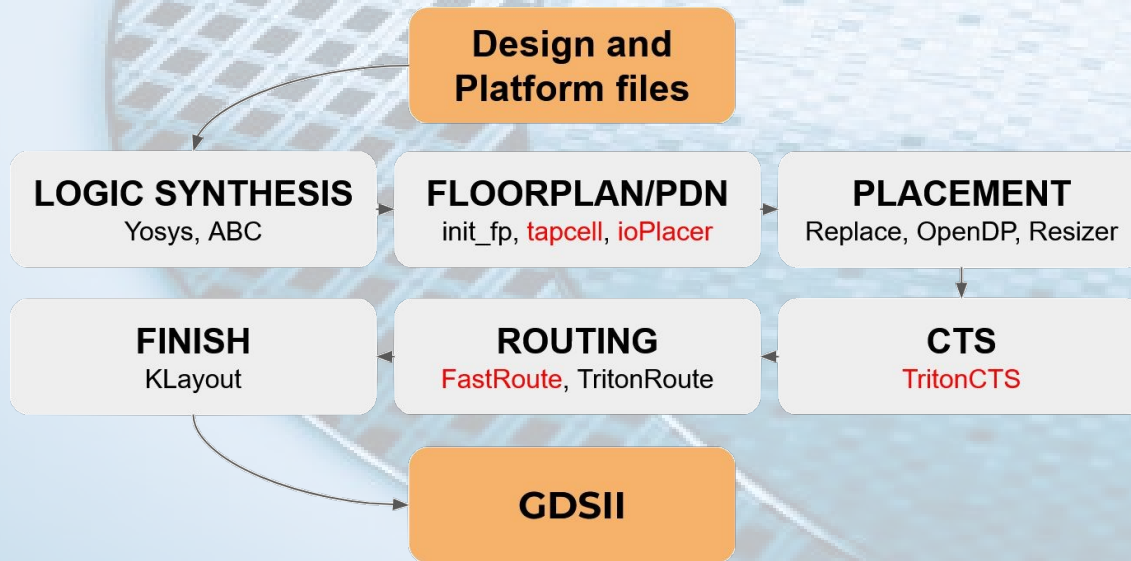


Além das universidades Norte Americanas , ainda outras instituições mundiais se engajaram dando suporte



Common Infrastructure	✓	Databases / Processing	
	✓	Cloud Infrastructure	
	✓	Timing Analysis	
	✓	Parasitic Extraction	
	✓	Readers + Writers	
Layout Generators	✓	Power and SI Analysis	
	✓	Logic Synthesis	
	✓	Floorplanning	
	✓	Placement	 
	✓	Clock Tree Synthesis	
	✓	Detailed Routing	 
Design	✓	Layout Finishing	 
		SoC Design Advisors	

As ferramentas destacadas em vermelho são desenvolvidas e mantidas pela equipe brasileira que dá suporte para quatro ferramentas no fluxo: nas etapas de Floorplan, Clock Tree Synthesis ( CTS ) e Routing.



Mateus Fogaça  
Isadora Oliveira  
Eder Monteiro  
Prof. Ricardo Reis  
Prof. Paulo Butzen



Marcelo Danigno



# OpenROAD Project

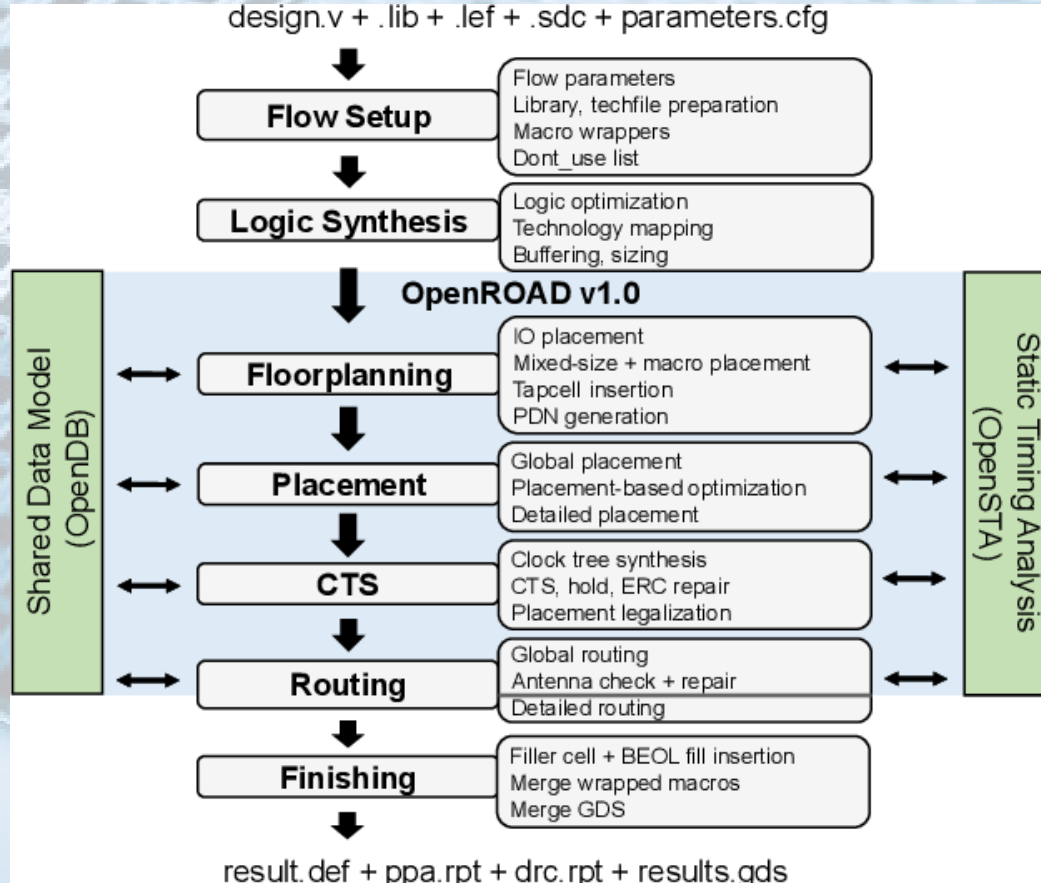
O projeto OpenROAD é um projeto sem fins lucrativos, financiado pela DARPA e patrocinado pelo Google, comprometido com a criação de ferramentas e fluxos inovadores e de baixo custo para automação de design eletrônico (EDA) para design de IC.

OpenROAD fornece um fluxo RTL-GDSII autônomo, sem intervenção humana, para design ASIC, estimativa de QoR e implementação física para uma variedade de tecnologias acima de 12 nm.

Ver na WEB em : <https://theopenroadproject.org/community/>

# The OpenROAD Project RTL to GDSII < 24 horas

Front-end



?



# Build Custom Silicon with

## PDK – Process Design Kit

Quem define o PDK é a Foundry

Nele, será definida a Tecnologia ( 180nm ou 130nm por exemplo ), comprimento do Gate, capacitância no Mosfet, etc.

Ver na WEB em: <https://developers.google.com/silicon?hl=pt-br>



**FOSS 130nm Production PDK**  
[github.com/google/skywater-pdk](https://github.com/google/skywater-pdk)



**FOSS 180nm Production PDK**  
[github.com/google/gf180mcu-pdk](https://github.com/google/gf180mcu-pdk)



TOUCAN  
eLAB



O módulo do OpenRoad é usado em outras aplicações, como :

OpenLane



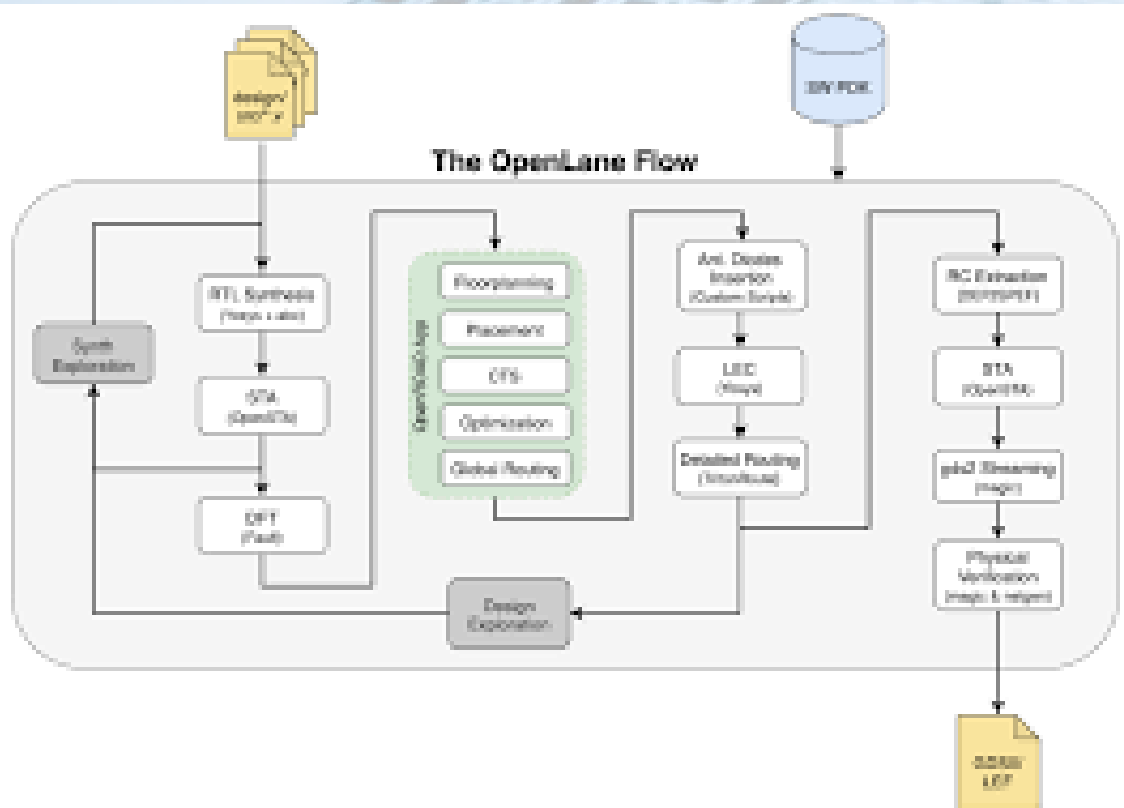
SiliconCompiler

efabless.com

zer|o


OpenFASOC  
para fluxo de Design com sinais mistos





OpenLane é um fluxo RTL para GDSII automatizado baseado em vários componentes, incluindo OpenROAD, Yosys, Magic, Netgen, CVC, SPEF-Extractor, KLayout e vários scripts personalizados para exploração e otimização de design.

O fluxo executa todas as etapas de implementação do ASIC, desde RTL até GDSII.

Na WEB em:  **GitHub**

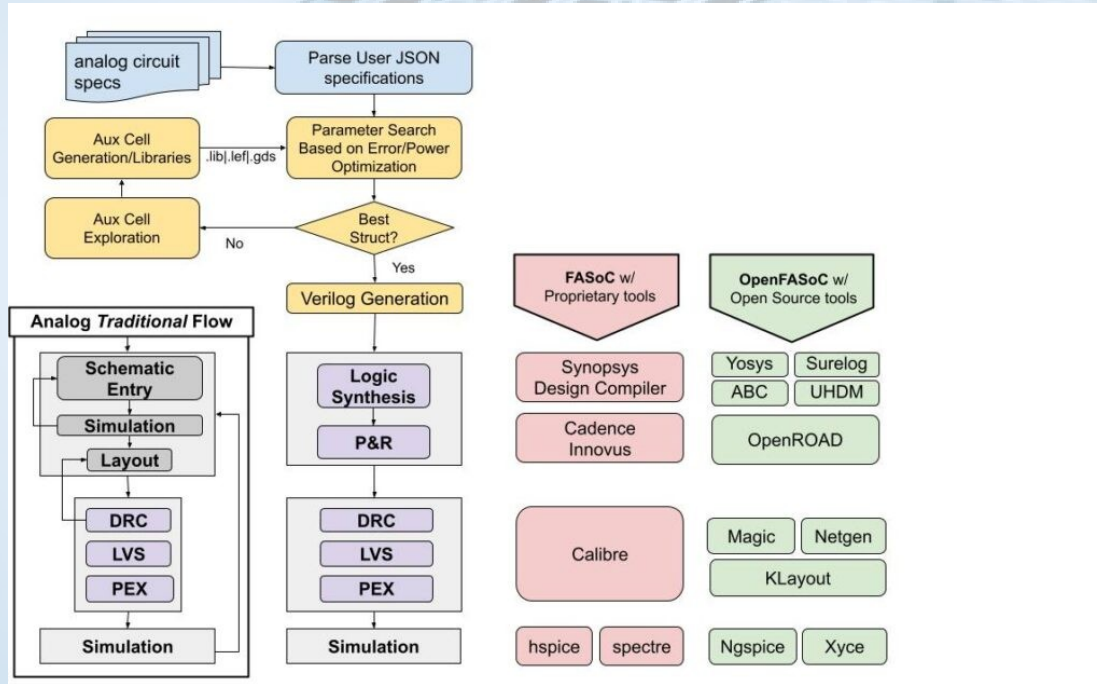
<https://github.com/The-OpenROAD-Project/OpenLane>

SiliconCompiler é um sistema modular de construção de hardware ("make for Silicon")

Na WEB em:  <https://github.com/siliconcompiler/siliconcompiler#supported-technologies>

Tecnologias aplicadas:


Tipo	Suportado
Languages	C, Verilog, SV, VHDL, Chisel, Migen/Amaranth, Bluespec
Simulation	Verilator, Icarus, GHDL
Synthesis	Yosys, Vivado, Synopsys, Cadence
ASIC APR	OpenRoad, Synopsys, Cadence
FPGA APR	VPR, nextpnr, Vivado
Layout Viewer	Klayout, OpenRoad, Cadence, Synopsys
DRC/LVS	Magic, Synopsys, Siemens
PDKs	sky130, asap7, freepdk45, gf12lp, gf22fdx, intel16



OpenFASOC está focado na geração analógica automatizada de código aberto, desde a especificação do usuário até GDSII.

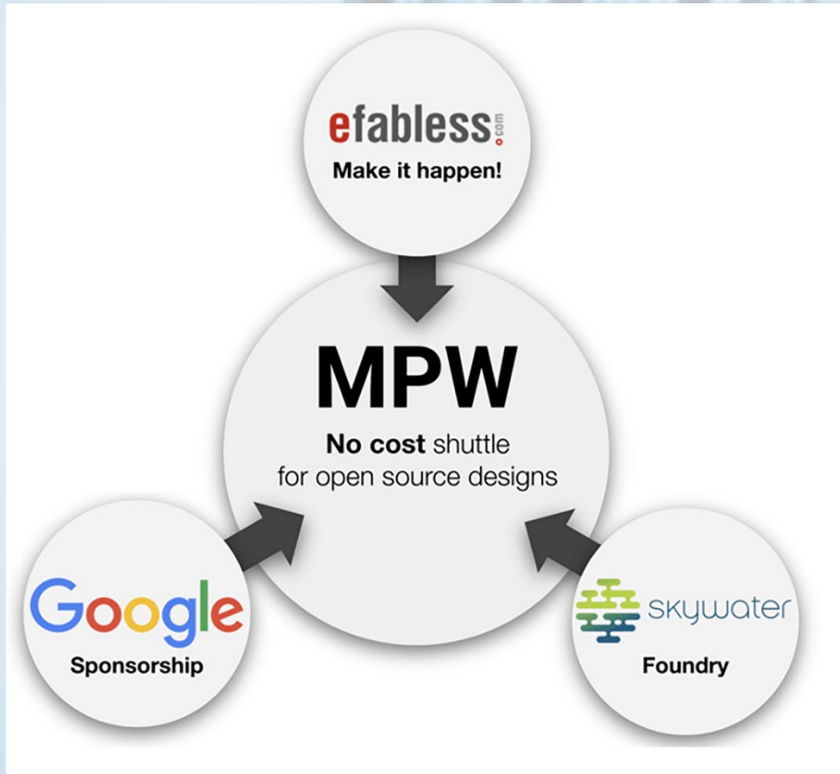
O Projeto é liderado por uma equipe de pesquisadores da Universidade de Michigan e é inspirado no FASoC, que se baseia em ferramentas proprietárias.

Todas as ferramentas são Open Source.

Na WEB em:  [GitHub https://github.com/idea-fasoc/OpenFASOC](https://github.com/idea-fasoc/OpenFASOC)

<https://theopenroadproject.org/automated-soc-mixed-signal-design-using-openroad-and-openfasoc/>

# Make Your Own Chips for Free



MPW - Multi-project Wafer

O Google fez uma parceria com GlobalFoundries, SkyWater Technology e Efabless para fornecer conjuntos de ferramentas e PDKs.

Assim, todos os desenvolvedores podem criar designs de silício manufacturados.

A cada dois meses, você poderá enviar seus designs de código aberto para inclusão no programa de traslado OpenMPW e ter a chance de fabricá-los sem custo.

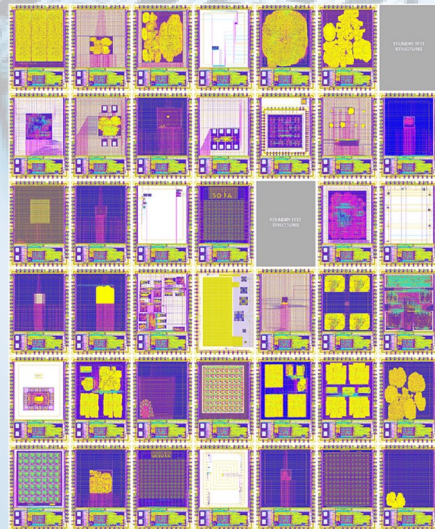




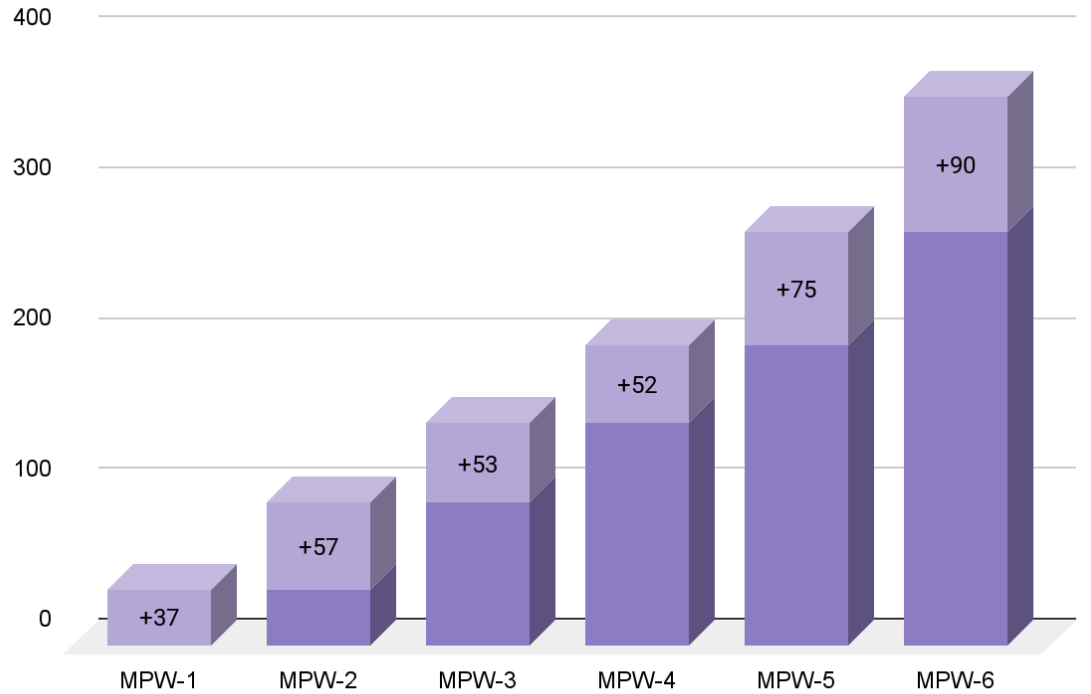
TOUCAN  
eLAB



# efabless.com



### Total number of Open MPW designs



↑ 2022



## Argentina – Brasil – Colômbia - Chile

### Type

Open MPW  
MPW-6



### Status

Closed



on 06/07/22 17:00:00 PDT

### Process

SKY130A  
Skywater 130nm



### Participants

78



+ 38 New Users

### Projects

90



31 Categories

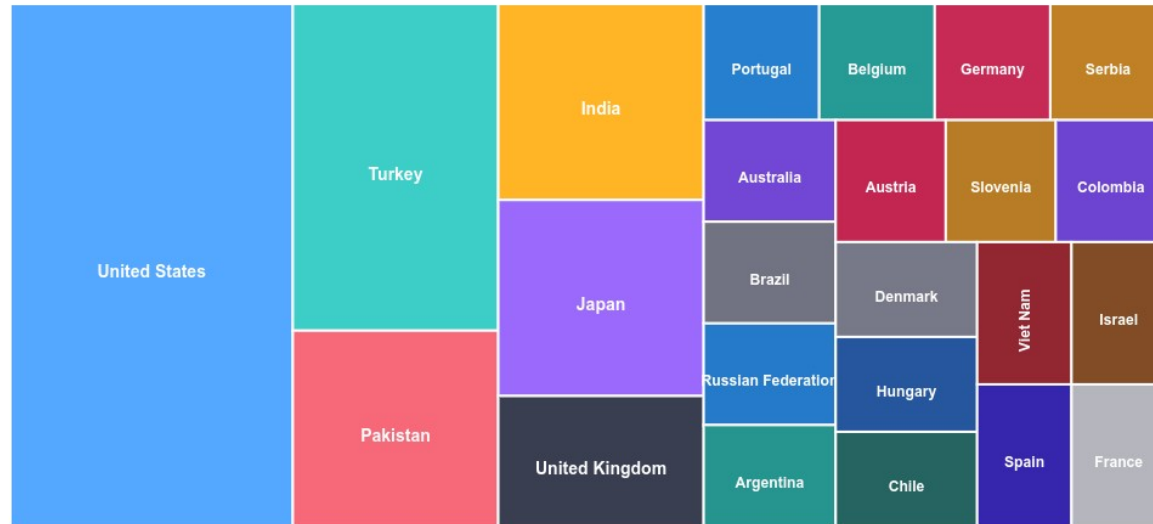
### Capacity

90 / 40



225.0 % Oversubscribed

### Community Diversity (24)





TOUCAN  
eLAB

## Multi-Project Wafer Service que cobram pelo Serviço



Os multi projetos de Chips (MPC) e os Multi projetos de Wafers (MPW) permitem que os clientes compartilhem o custo de fabricação de máscaras e wafers entre vários projetos.



THE **MOSIS** SERVICE

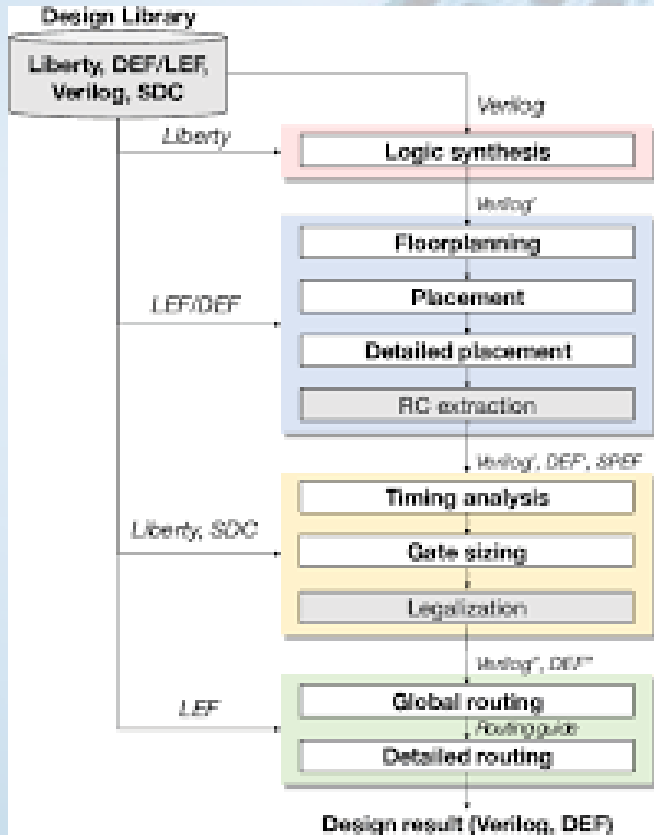
**efabless**.com



It's not just an MPW

**Muse Semiconductor**





## DATC - Design Automation Technical Committee

Sua missão é fornecer um fórum para discussão de questões estratégicas e atuais em automação de design, incluindo a coordenação de esforços para produzir fluxos de design e casos de teste disponíveis publicamente e organizar workshops, reuniões, atividades e publicações sobre tópicos relacionados.

RDF - Robust Design Flow - fluxo de design acadêmico de código aberto da síntese lógica ( RDF 2018 )

July 2019

RDF 2019 = RDF 2018 + OpenROAD + outras ferramentas ficou sendo um superset do OpenRoad

<https://ieee-ceda.org/technical-committee/datc/publications>

## Contest ( Desafios ):

IEEE - SSCS PICO Program

## Projetos Educacionais :

Zero to ASIC

Tiny Tapeout

VSD ( VLSI System Design )

## Hospedagem e Worgroups:

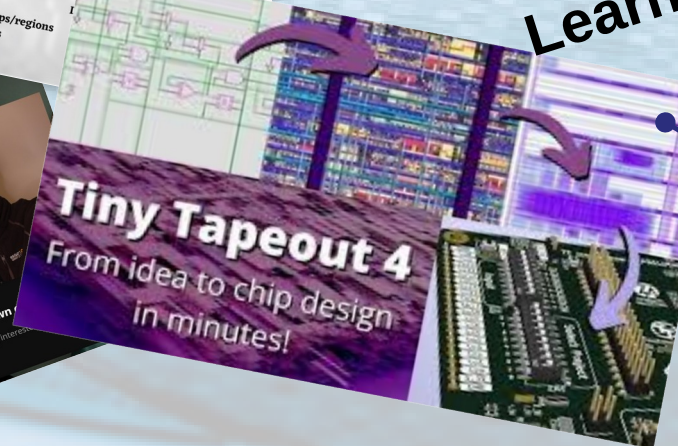
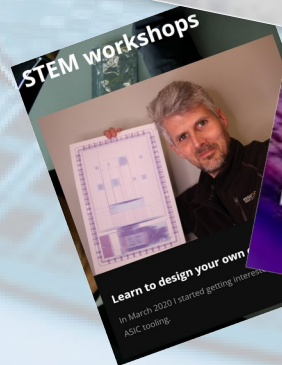
Alliance Chip

## Comunidade e Inovação :

The FOOSi Foundation



Learn From VSD-IAT





# Democratizing IC Design: The SSCS PICO Program

Através do seu programa Platform for IC Design Outreach (PICO), o SSCS está trabalhando com a comunidade de código aberto em rápido crescimento para ajudar a acelerar a construção do ecossistema necessário.

O programa fornecerá oportunidades de educação, orientação e colaboração, bem como corridas de fabricação patrocinadas.





## Algumas participações Latino Americanas desse Programa:

2021



Argentine School of Micro-Nanoelectronics, Buenos Aires, Argentina,  
Collection of EAMTA 2021 student projects

[https://github.com/eamta/caravel\\_eamta\\_2021](https://github.com/eamta/caravel_eamta_2021)



U. Federal de Santa Catarina, Brazil,  
Ultra-low-power analog front-end for bio signals, Chip 3

<https://platform.efabless.com/projects/474>

2022



Leonardo Amorese Gallo Gomes  
60 GHz demonstrator chip

[https://github.com/LeoGalloGomes/sky130\\_mm-wave\\_demonstrator](https://github.com/LeoGalloGomes/sky130_mm-wave_demonstrator)



Maximiliano Cerda Universidad del Bio Bio  
Mix-Pix - A mixed signal circuit for smart imaging

<https://github.com/MaxRCC/MixPix.git>



Criado por Matthew Venn, Engenheiro Eletrônico e Divulgador de Ciência e Tecnologia.

Seu objetivo é criar experiência de Aprendizagem para uma comunidade em todo mundo.

Em seu site [//zerotoasiccourse.com](https://zerotoasiccourse.com) Venn faz seu chamado:

“Aprenda a projetar seu próprio ASIC e fabricá-lo! Graças ao novo Process Development Kit de código aberto do **Google e Skywater** e às ferramentas **OpenLane ASIC da Efabless**, agora temos a oportunidade de nos envolver neste campo emocionante sem assinar NDAs ou pagar uma fortuna por licenças de ferramentas.”



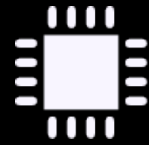


“TinyTapeout é um projeto educacional que torna mais fácil e barato do que nunca fabricar seus designs digitais em um chip real! “

Passos:

- 1 - Estude o Guia de Design.
- 2 - Crie seu próprio design com o modelo Wokwi ou, para usuários avançados, um HDL.
- 3 - Participe do grupo no Discord para tirar duvidas.
- 4 - Envie seu Projeto





VSD oferece treinamento entre eles:

- Fluxo de back-end vlsi,
- Síntese Digital e verificação RTL,
- Planejamento e design de SoC,
- Design de IP,
- Automação CAD/EDA
- RISC-V,
- Inteligência de máquina em EDA/CAD,



Plataforma de treinamento  
construída por especialistas e  
adequada aos requisitos do  
designer.

<https://www.vlssystemdesign.com/vsdopen2023/>

A CHIPS Alliance é uma organização que desenvolve e hospeda código de hardware de código aberto (núcleos IP) de alta qualidade, IP de interconexão (protocolos físicos e lógicos) e ferramentas de desenvolvimento de software de código aberto para design, verificação e muito mais.

Fornece um ambiente colaborativo sem barreiras, para reduzir o custo de desenvolvimento de IP e ferramentas para desenvolvimento de hardware.

### Alguns Projetos:



Conjunto de ferramentas Grátis e de código aberto para dispositivos FPGA



Suporte a linguagem de construção de hardware Chisel e projetos relacionados

FASoC

Síntese de SoC totalmente autônoma usando circuitos analógicos sintetizáveis baseados em células customizadas.

A Fundação FOSSi existe para promover e proteger o movimento dos chips de silício de código aberto.

Incentiva ativamente o crescimento da comunidade e ajuda a manter o espírito aberto do movimento, através de eventos, programas educacionais e grupos de trabalho.

Com uma associação internacional de especialistas da academia e da indústria, apoia novas iniciativas e colaborações de código aberto – oferecendo aconselhamento gratuito a governos e decisores políticos, empresas, acadêmicos e amadores.

Como organização sem fins lucrativos, a fundação é independente de quaisquer interesses comerciais e atua como administradora no apoio a projetos de código aberto que beneficiam amplamente a comunidade de silício de código aberto.

FOSSi é um acrônimo para Silício Livre e de Código Aberto.

The premier open source silicon conference

2023 September 15th to 17th in Munich, Germany

ORConf é uma conferência anual para designers de semicondutores de código aberto, desenvolvedores de ferramentas EDA de código aberto e a comunidade.

Todos os anos, os participantes são apresentados com uma variedade cada vez mais impressionante de apresentações de todos os cantos do espaço de hardware de código aberto.

<https://orconf.org/>





TOUCAN  
eLAB



# Ambiente de Desenvolvimento



Sistema Operacional : Linux



Maquina Virtual : Virtual Box



Contêiner : Docker



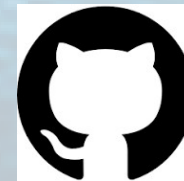
Cloud : AWS



Silicon – Notebook : Jupyter , Colab



Versionamento : Git, Github



Banco de Dados : OpenDB

# OpenDB

OpenDB é um banco de dados de design para oferecer suporte a ferramentas para design de chips físicos.

Foi originalmente desenvolvido pela Athena Design Systems. Nefelus, Inc adquiriu os direitos do código e o abriu em 2019 para apoiar o projeto DARPA OpenROAD.

A estrutura do OpenDB é baseada nos formatos de arquivo de texto da Cadence Design Systems LEF (biblioteca) e DEF (design) versão 5.6.

OpenDB suporta um formato de arquivo binário para salvar e carregar o design muito mais rápido do que usar LEF e DEF.



TOUCAN  
eLAB



# AI no Design de Circuitos Integrados





O ChatGPT é uma ferramenta baseada em Inteligência Artificial (IA), pública, que permite ao usuário fazer perguntas e receber respostas textuais, da mesma forma que um aplicativo de conversa, como o Whatsapp.

As suas respostas são baseadas em linguagem natural, ou seja, ele gera uma conversa natural com o usuário, como se fosse uma pessoa, e em diversos idiomas, inclusive o português.

Os tipos de resposta que ele gera podem ser resumos, artigos, listas, recomendações e até códigos de linguagens de programação.



Bard é um **chatbot** desenvolvido pelo **Google**, baseado na família de modelos de linguagem **LaMDA**. Foi criado como uma resposta direta ao **ChatGPT** da **OpenAI**, e foi lançado em uma capacidade limitada em março de 2023

## Exemplo simples de interação:

--> **CODIFICAR EM VERILOG UM SOMADOR DE 4 BITS**

Claro, aqui está a codificação em Verilog para um somador de 4 bits:

...

```
module full_adder (  
    input wire A, B, Cin,  
    output wire S,  
    output wire Cout  
);  
  
    assign S = A ^ B ^ Cin;  
    assign Cout = (A & B) | (B & Cin) | (Cin & A);  
  
endmodule
```

```
module four_bit_adder (  
    input wire [3:0] A, B,  
    output wire [3:0] S,  
    output wire Cout  
);
```

```
    full_adder FA0 (A[0], B[0], 1'b0, S[0], Cout);  
    full_adder FA1 (A[1], B[1], Cout, S[1], Cout);  
    full_adder FA2 (A[2], B[2], Cout, S[2], Cout);  
    full_adder FA3 (A[3], B[3], Cout, S[3], Cout);
```

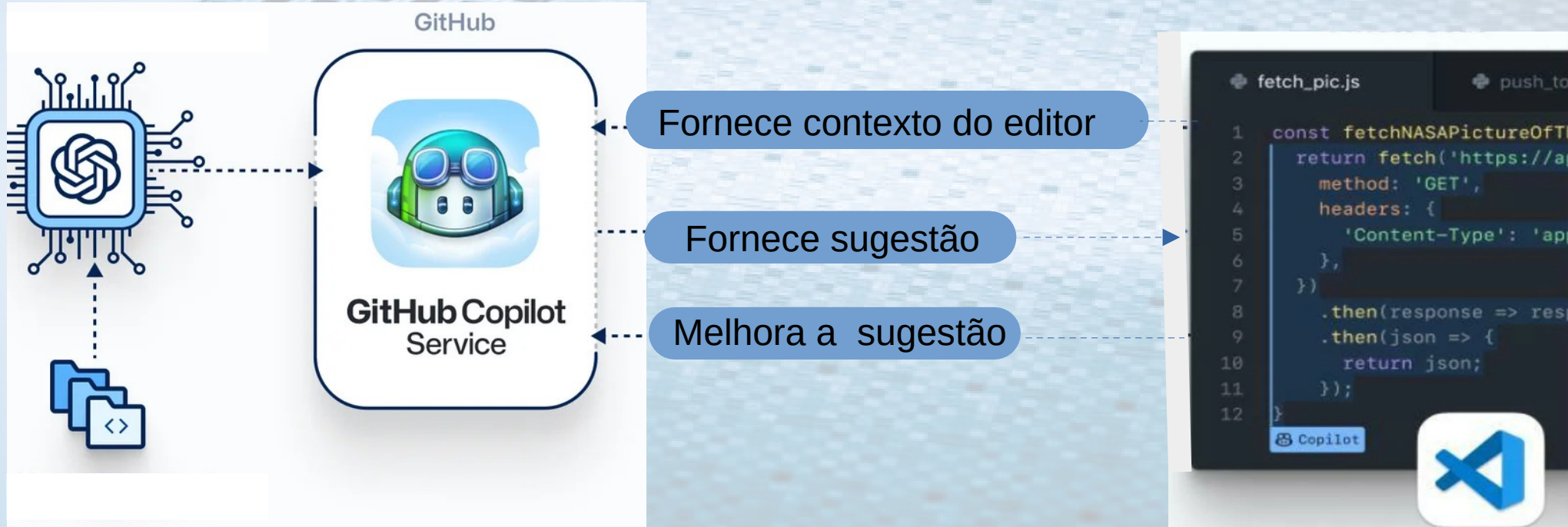
```
endmodule`
```

Este código define um módulo chamado `full\_adder` que tem três entradas: `A`, `B` e `Cin`. Ele também tem duas saídas: `S` e `Cout`. O módulo `full\_adder` calcula a soma dos seus dois operandos, `A` e `B`, e também calcula o carry-out. O carry-out é então passado para o próximo módulo `full\_adder` na cadeia.

O módulo `four\_bit\_adder` usa quatro módulos `full\_adder` para adicionar dois números de 4 bits. Ele também calcula o carry-out da adição. O carry-out é então passado para a saída `Cout`.

Espero que isto ajude!

O GitHub Copilot é uma ferramenta de IA desenvolvida pelo GitHub em conjunto com a OpenAI,



Copilot auxiliar usuários de ambientes de desenvolvimento integrados como Visual Studio Code, Visual Studio, Neovim e JetBrains por meio de sugestão de código para preenchimento automático.



TOUCAN  
eLAB



# Dúvidas?

[contato@toucan-elab.com.br](mailto:contato@toucan-elab.com.br)

## NAMASTÊ



# OBRIGADO!



Patrocinado por



[www.embarcados.com.br](http://www.embarcados.com.br)



[linkedin.com/embarcados](https://linkedin.com/embarcados)



[@portalembarcados](https://instagram.com/portalembarcados)



[youtube/Embarcados TV](https://youtube.com/EmbarcadosTV)