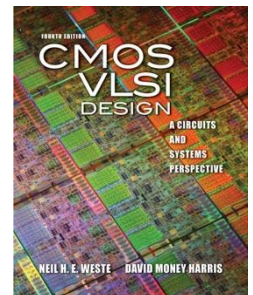
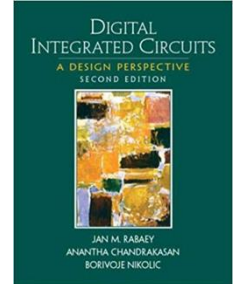


Microeletrônica

Aula #11 → Projeto Standard-Cells

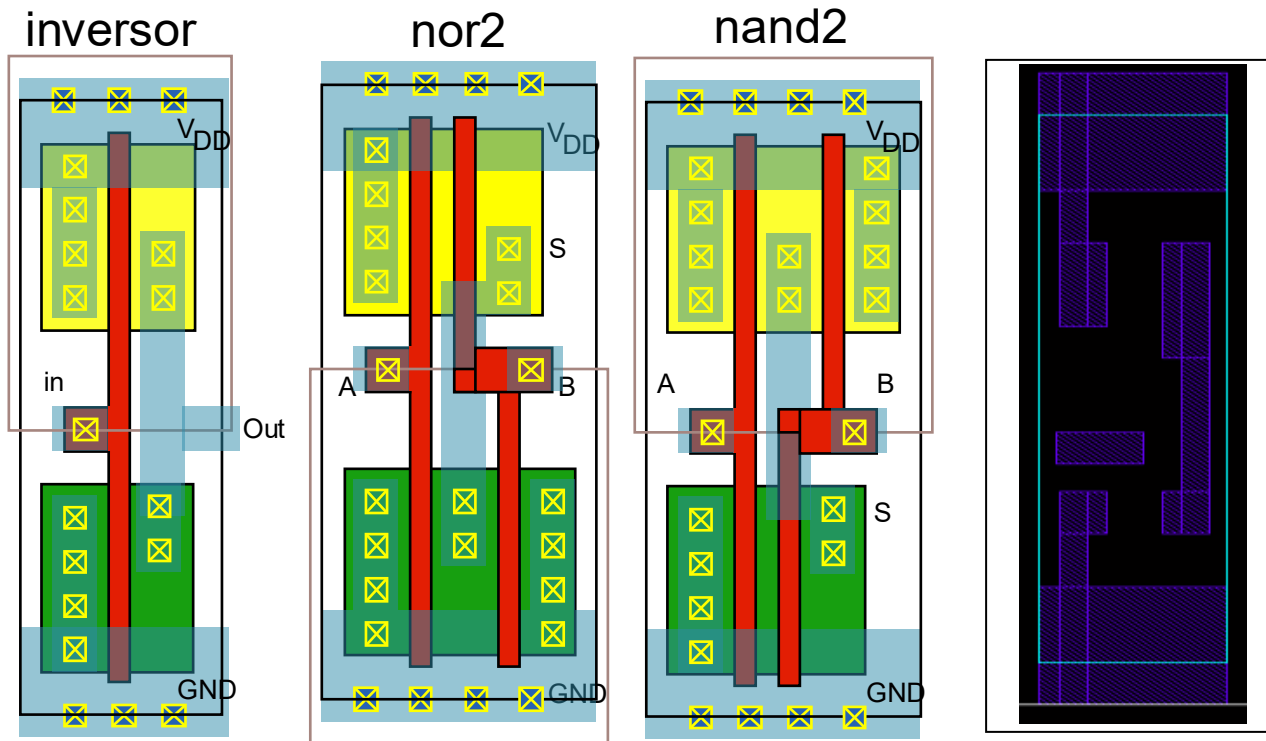
- Professor: Fernando Gehm Moraes
- Livro texto:
 - Digital Integrated Circuits a Design Perspective - Rabaey
 - C MOS VLSI Design - Weste



10/junho/2026

Standard-cells

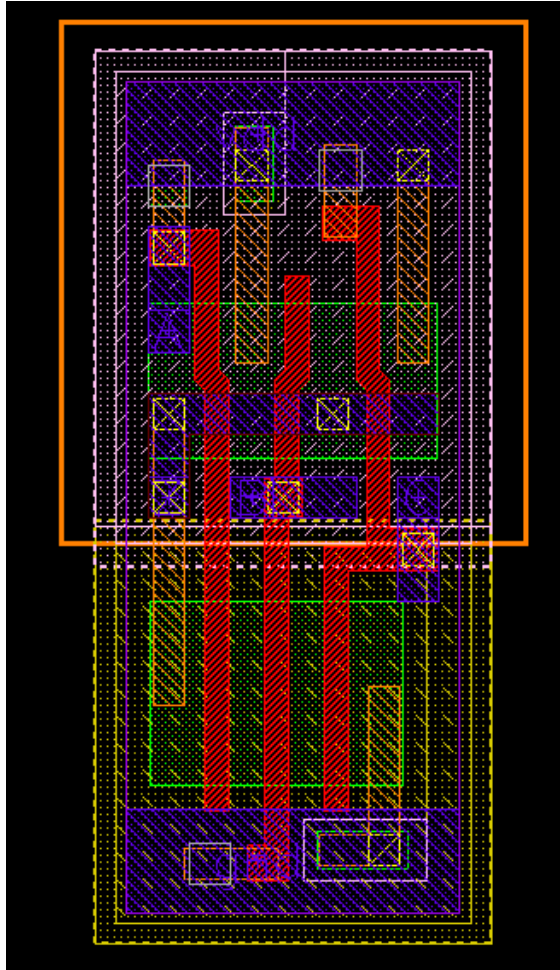
- Reduz a complexidade para o projeto
 - *full custom* só no nível de células
- Layout de um circuito é montado usando células disponíveis em uma biblioteca de células
- Todas as camadas são fabricadas na *foundry*



"Vista"
LEF

Library
Exchange
Format

Standard Cells



3-input NAND cell
(from ST Microelectronics)

Biblioteca de células

- Diversos layouts para cada tipo de célula
- Cada layout prevê uma certa carga capacitiva máxima (*logic-effort*)
- Todas as células têm a **mesma altura** e são **pré-caracterizadas**
- Conexão da alimentação por justaposição

Path	1.2V - 125°C	1.6V - 40°C
$In1-t_{pLH}$	$0.073+7.98C+0.317T$	$0.020+2.73C+0.253T$
$In1-t_{pHL}$	$0.069+8.43C+0.364T$	$0.018+2.14C+0.292T$
$In2-t_{pLH}$	$0.101+7.97C+0.318T$	$0.026+2.38C+0.255T$
$In2-t_{pHL}$	$0.097+8.42C+0.325T$	$0.023+2.14C+0.269T$
$In3-t_{pLH}$	$0.120+8.00C+0.318T$	$0.031+2.37C+0.258T$
$In3-t_{pHL}$	$0.110+8.41C+0.280T$	$0.027+2.15C+0.223T$

C = Load capacitance T = input rise/fall time

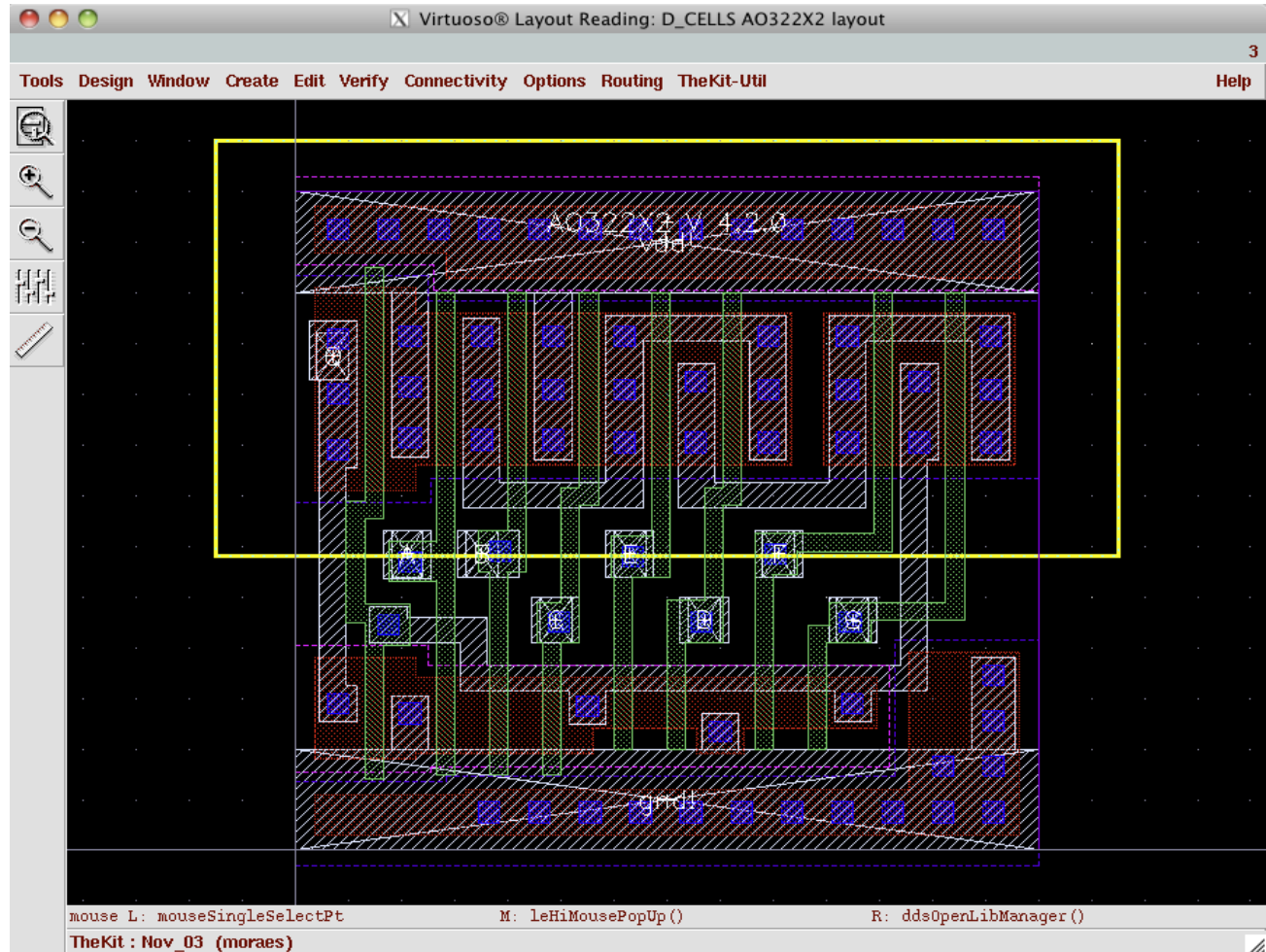
Vista: LIB

Standard Cells

Diversos layouts para cada tipo de célula
- diferentes *drive strengths*

```
HS65_GL_IVX2  
HS65_GL_IVX4  
HS65_GL_IVX7  
HS65_GL_IVX9  
HS65_GL_IVX13  
HS65_GL_IVX18  
HS65_GL_IVX22  
HS65_GL_IVX27  
HS65_GL_IVX31  
HS65_GL_IVX35  
HS65_GL_IVX40  
HS65_GL_IVX44  
HS65_GL_IVX49  
HS65_GL_IVX53  
HS65_GL_IVX62  
HS65_GL_IVX71  
HS65_GL_IVX106  
HS65_GL_IVX142  
HS65_GL_IVX213  
HS65_GL_IVX284
```

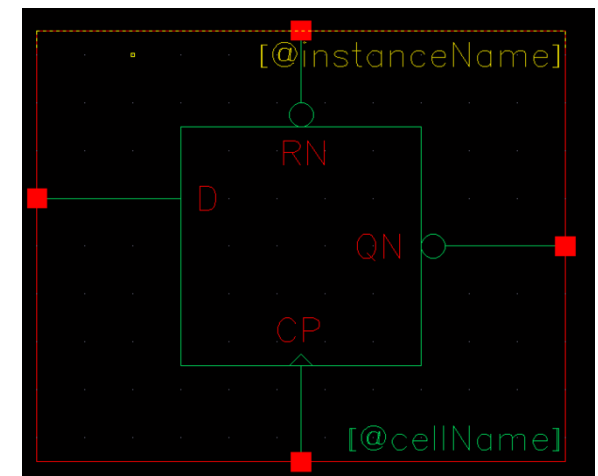
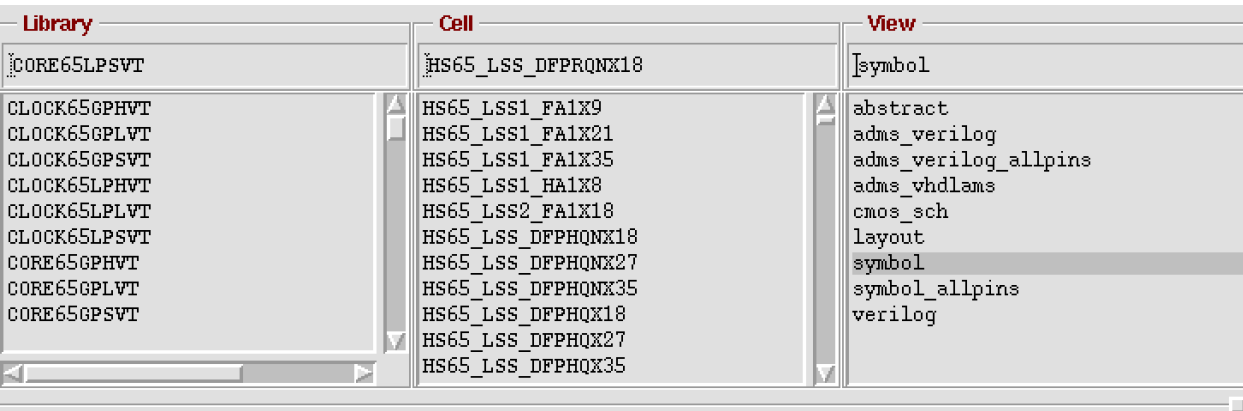
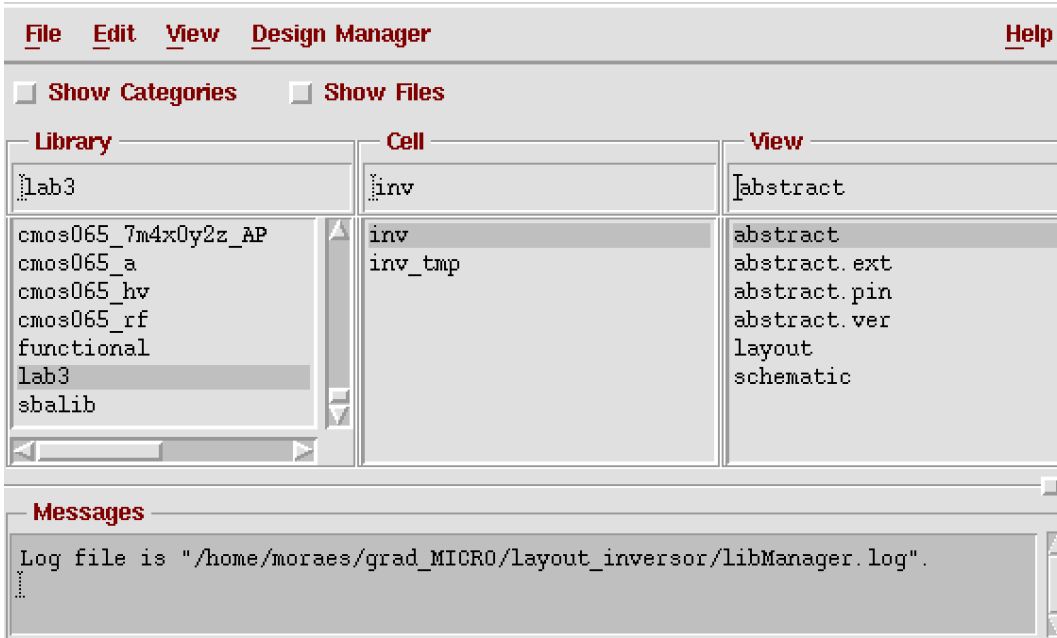
```
HS65_GL_NAND2X2  
HS65_GL_NAND2X4  
HS65_GL_NAND2X5  
HS65_GL_NAND2X7  
HS65_GL_NAND2X11  
HS65_GL_NAND2X14  
HS65_GL_NAND2X21  
HS65_GL_NAND2X29  
HS65_GL_NAND2X43  
HS65_GL_NAND2X57
```



Standard Cells

– “Vistas” de uma célula

- LEF (abstract)
- LIB
- layout
- symbol
- esquemático
- verilog
- ...



Standard Cells

Automatização corresponde a montar o layout

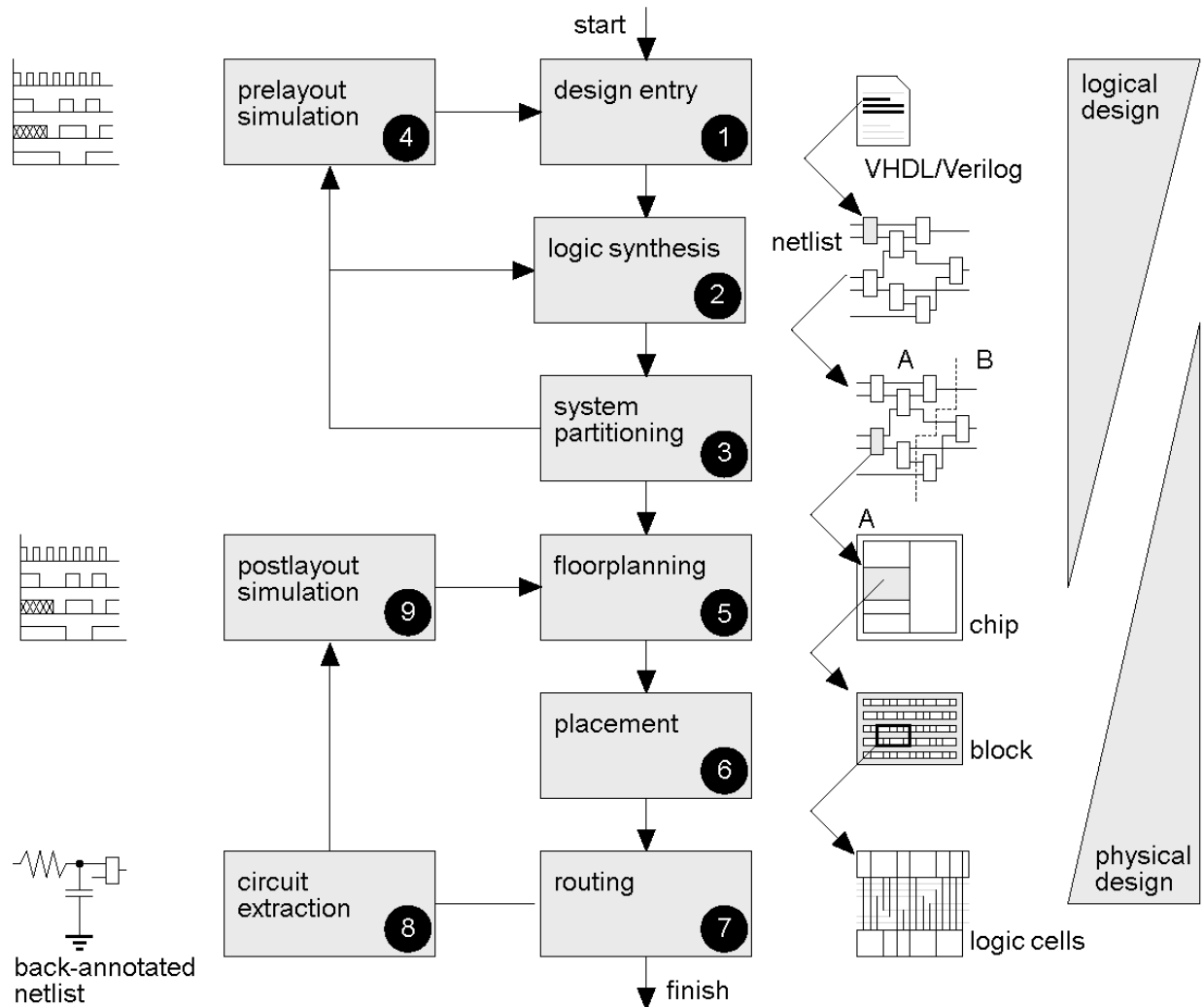
- **Particionamento**
 - **Posicionamento**
 - **Roteamento**
- Etapas da síntese física**

Muito usado na indústria por permitir prever

- **Área**
- **Potência**
- **Atraso**

Etapas do Fluxo de Projeto – visão geral

- 1. Design entry** - Using a hardware description language (HDL) or schematic entry
- 2. Logic synthesis** - Produces a netlist - logic cells and their connections
- 3. System partitioning** - Divide a large system into ASIC-sized pieces
- 4. Prelayout simulation** - Check to see if the design functions correctly
- 5. Floorplanning** - Arrange the blocks of the netlist on the chip
- 6. Placement** - Decide the locations of cells in a block
- 7. Routing** - Make the connections between cells and blocks
- 8. Extraction** - Determine the resistance and capacitance of the interconnect
- 9. Postlayout simulation** - Check to see the design still works with the added loads of the interconnect



(1) Design entry

VHDL, Verilog

“think hardware!”:

mesmo que esquemático

seja coisa do passado,

reconhecer no HDL

estruturas como FSMs,

decoders, mux,

registradores, é **necessário**

```
process( Bus2IP_Clk ) is
begin
    if Bus2IP_Clk'event and Bus2IP_Clk = '1' then
        if Bus2IP_Reset = '1' then
            slv_reg(0) <= (others => '0');
            slv_reg(1) <= (others => '0');
            slv_reg(2) <= (others => '0');
            slv_reg(3) <= (others => '0');
            slv_reg(4) <= (others => '0');
            slv_reg(5) <= (others => '0');
            slv_reg(6) <= (others => '0');
            slv_reg(7) <= (others => '0');
            slv_reg(8) <= (others => '0');
            slv_reg(9) <= (others => '0');
        elsif reseta_bit_slv_reg0='1' then
            slv_reg(9) <= (others => '0');
        else
            case Bus2IP_WrCE(0 to 14) is
                when "10000000000000" => slv_reg(0) <= Bus2IP_Data;
                when "01000000000000" => slv_reg(1) <= Bus2IP_Data;
                when "00100000000000" => slv_reg(2) <= Bus2IP_Data;
                when "00010000000000" => slv_reg(3) <= Bus2IP_Data;
                when "00001000000000" => slv_reg(4) <= Bus2IP_Data;
                when "00000100000000" => slv_reg(5) <= Bus2IP_Data;
                when "00000010000000" => slv_reg(6) <= Bus2IP_Data;
                when "00000001000000" => slv_reg(7) <= Bus2IP_Data;
                when "00000000100000" => slv_reg(8) <= Bus2IP_Data;
                when "00000000010000" => slv_reg(9) <= Bus2IP_Data;
                when others => null;
            end case;
        end if;
    end if;
end process;
```


Exemplo de simulador: xrun (CADENCE), questa (SIEMENS)

The image displays the Cadence SimVision simulation environment. The top-left window is the 'Console - SimVision' window, showing the command-line interface for the ncsim simulator. The commands entered are:

```
ncsim>  
ncsim>  
ncsim>  
ncsim>  
ncsim>  
ncsim> reset  
Loaded snapshot work.user_logic_tb.tb_architecture  
ncsim>  
ncsim> run 12 us
```

The bottom-right window is the 'Waveform 2 - SimVision' window, showing a digital waveform. The waveform is titled 'Baseline = 11,197,981,874fs' and 'Cursor-Baseline = 661,668,562fs'. The time scale is set to 12,000,000,000fs + 0. The waveform shows a clock signal (Bus2IP_Clk) and several data signals (sly_reg, sly_ip2bus_data, reseta_bit_sly_reg0, pixel, endx, endy, EA, Bus2IP_WCE, Bus2IP_Reset, Bus2IP_Clk). The Bus2IP_Clk signal is highlighted in yellow in the signal list. The waveform shows a clock signal (Bus2IP_Clk) and several data signals (sly_reg, sly_ip2bus_data, reseta_bit_sly_reg0, pixel, endx, endy, EA, Bus2IP_WCE, Bus2IP_Reset, Bus2IP_Clk). The Bus2IP_Clk signal is highlighted in yellow in the signal list. The waveform shows a clock signal (Bus2IP_Clk) and several data signals (sly_reg, sly_ip2bus_data, reseta_bit_sly_reg0, pixel, endx, endy, EA, Bus2IP_WCE, Bus2IP_Reset, Bus2IP_Clk). The Bus2IP_Clk signal is highlighted in yellow in the signal list.

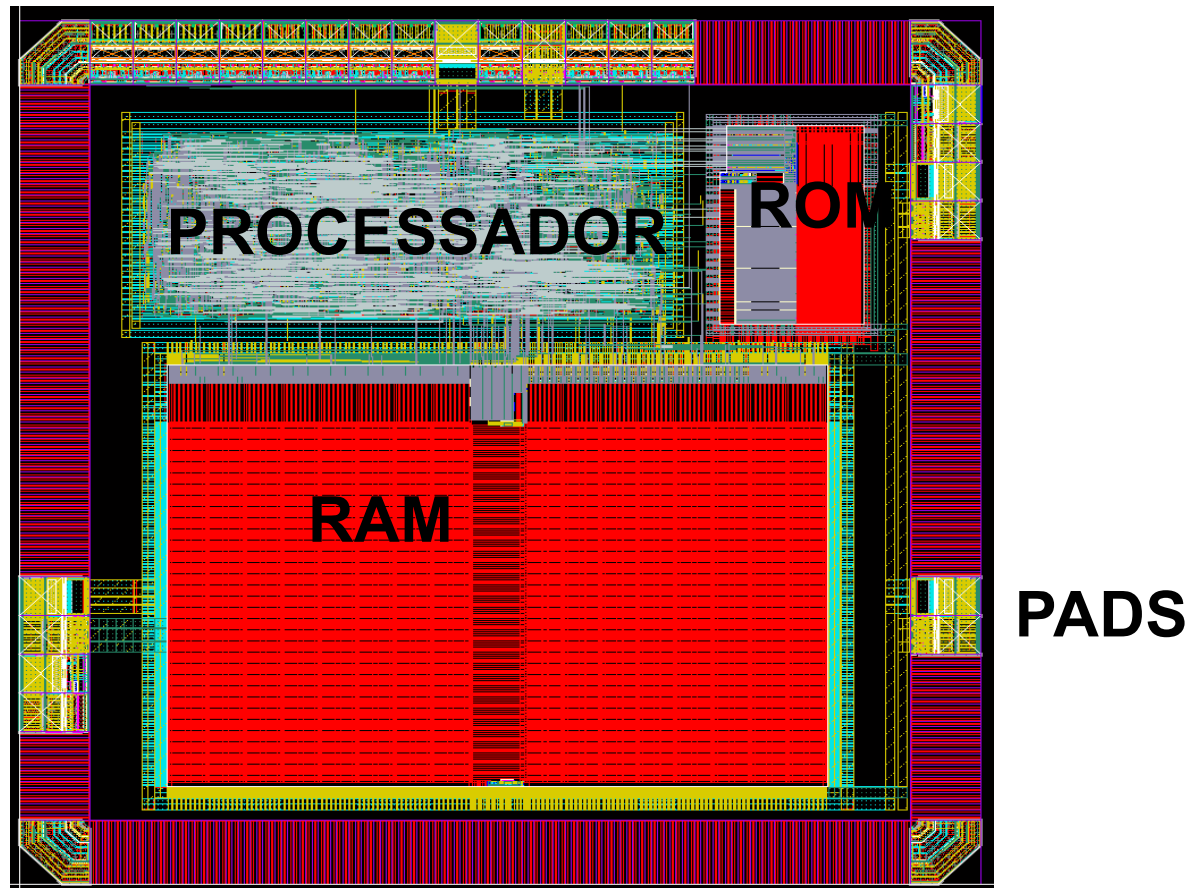
The waveform viewer shows a time axis from 0 to 12,000,000,000fs. The signals are plotted as digital waveforms. The Bus2IP_Clk signal is a periodic square wave. The other signals show various data patterns, including a 15-bit bus (sly_reg) and a 15-bit bus (sly_ip2bus_data). The Bus2IP_Reset signal is a single pulse. The Bus2IP_WCE signal is a periodic square wave. The EA signal is a periodic square wave. The endy signal is a periodic square wave. The endx signal is a periodic square wave. The pixel signal is a periodic square wave. The reseta_bit_sly_reg0 signal is a periodic square wave. The sly_ip2bus_data signal shows a sequence of hex values: 00, 02, 01, 02, 0A, 00. The sly_reg signal shows a sequence of hex values: 8C, D, E, E, REP, 0000.

(2) Particionamento e *floorplanning*

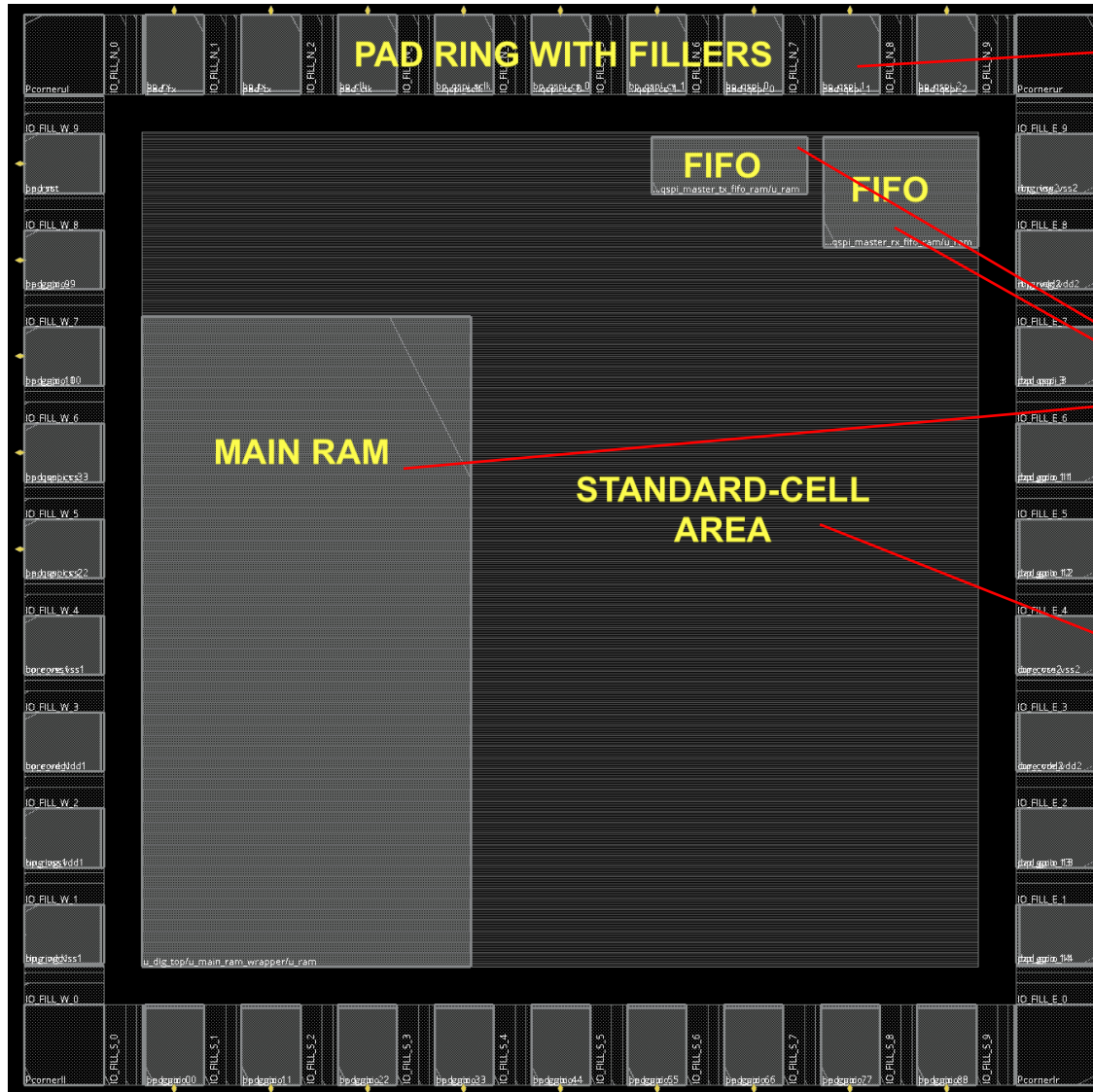
Divide o circuito em módulos

Floorplanning:

- Define a forma dos blocos standard cells e o posicionamento dos blocos “fixos”



Exemplo



PAD

- Célula de E/S

Blocos de memória:

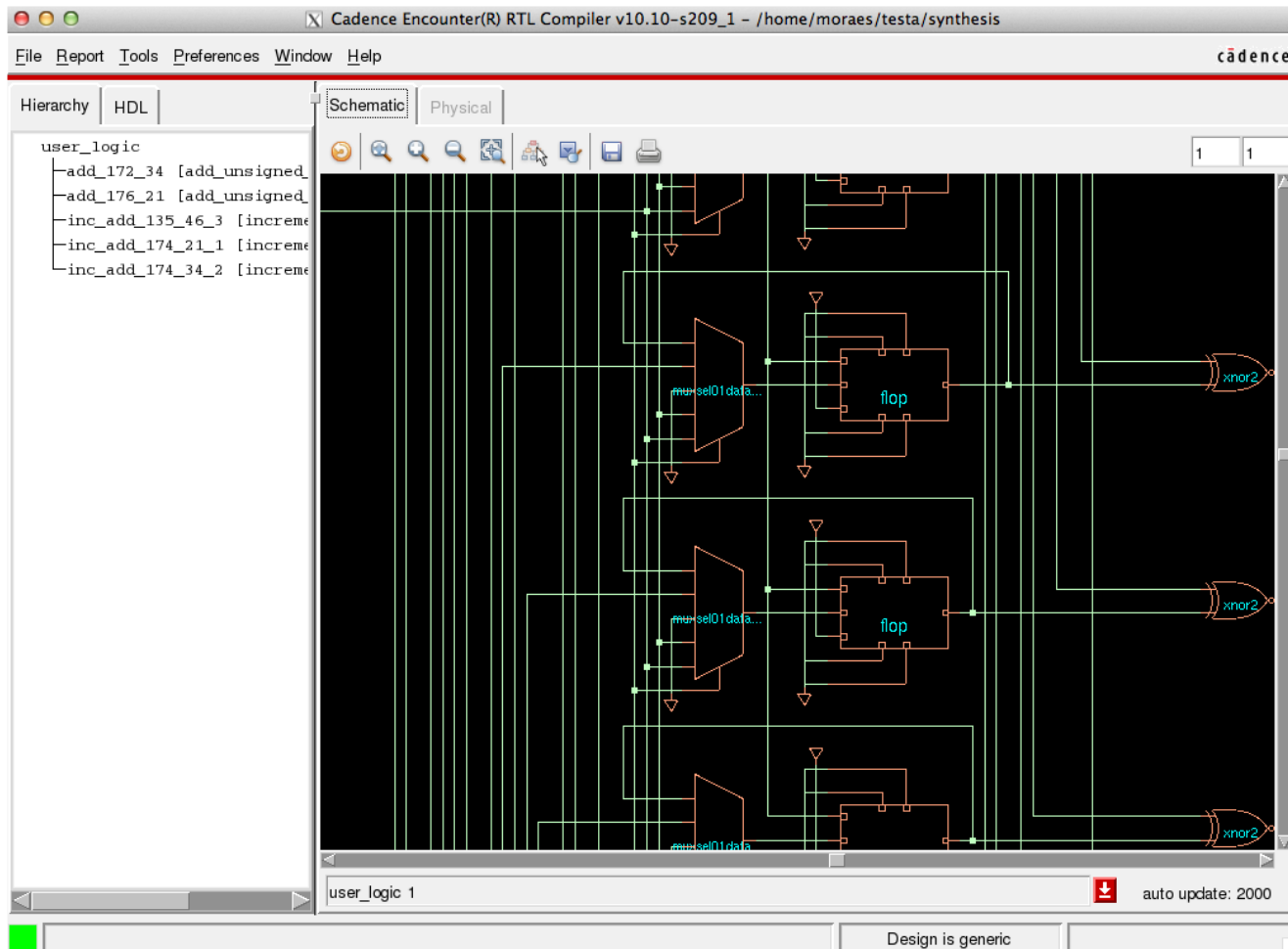
- Gerados automaticamente por ferramentas da tecnologia alvo

Standard-cells

- Gerados por síntese lógica

(3) Síntese Lógica

- Transforma o VHDL/Verilog em um netlist mapeado para uma dada tecnologia
- Importante nesta etapa: **restrições de projeto**
- Exemplo de ferramenta: **Genus** (Cadence)
- Utilização por **script**



```
## 1) load synthesis configuration, read description and elaborate design
```

```
include ./constraint/load.tcl
```

```
read_hdl -sv nanoCPU.sv
```

```
elaborate NanoCPU
```

```
set_db [current_design] .dft_dont_scan true
```

**Le configurações da tecnologia,
HDL e verifica código**

```
## 2) read constraints
```

```
read_sdc ./constraint/restrictions.sdc
```

Restrições de temporização

```
## 3) synthesize to mapped
```

```
syn_generic
```

```
syn_map
```

```
syn_opt
```

**Síntese para portas lógicas, portas da biblioteca de
células e otimização**

```
## 4) reports
```

```
report_area > nano.txt
```

```
report_gates >> nano.txt
```

```
report_timing >> nano.txt
```

```
report_power -unit mW >> nano.txt
```

Relatórios área, portas lógicas e power

```
## 5) build physical synthesis environment
```

```
write_netlist [current_design] > nano.v
```

```
write_db -common -all_root_attributes nanoCPU_genus.db
```

```
exit
```

Exporta para Innovvus

load .tcl – principais comandos

```
set_db syn_generic_effort high  
set_db syn_map_effort high  
set_db syn_opt_effort high
```

Esforço das etapas de síntese

```
#keep hierarchy  
set_db auto_ungroup none
```

Mantém hierarquia

```
#=====
```

```
# Load libraries
```

Carrega os LIB e LEF (é o que define a tecnologia)

```
#=====
```

```
#Set liberty
```

```
set_db library "...C28SOI_SC_12_CORE_LR_ss28_0.90V_125C.lib ..."
```

```
#set LEF
```

```
set_db lef_library "sites.lef ... technology.12T.lef ... C28SOI_SC_12_CORE_LR_soc.lef"
```

```
#Set captable
```

```
set_db cap_table_file "... FuncRCmax.captable ..."
```

Capacitâncias parasitas dos fios

```
#Set PLE
```

```
set_db interconnect_mode ple
```

**Modo de estimativa de
capacitância dos fios**

restrictions.sdc – principais comandos (Synopsys Design Constraints)

```
set sdc_version 2.0
```

```
set_units -capacitance pF -time ns
```

```
# 500 MHz freq
```

```
create_clock -name {ck} -period 2.0 [get_ports {ck}]
```

Freq. de clock

```
set_false_path -from [get_ports {rst}]
```

Ignora o reset na análise de timing

```
## INPUTS
```

```
set_input_delay -clock ck -max 0.03 [all_inputs]
```

```
# Output delay
```

```
set_output_delay -clock ck 0.03 [all_outputs]
```

```
# Output pins should support to drive a load of an inverter
```

```
set_load 0.000570 [all_outputs]
```

Ao ler as restrições, deve-se verificar se as mesmas foram atendidas

```
@genus:root:> read_sdc ./constraint/restrictions.sdc
```

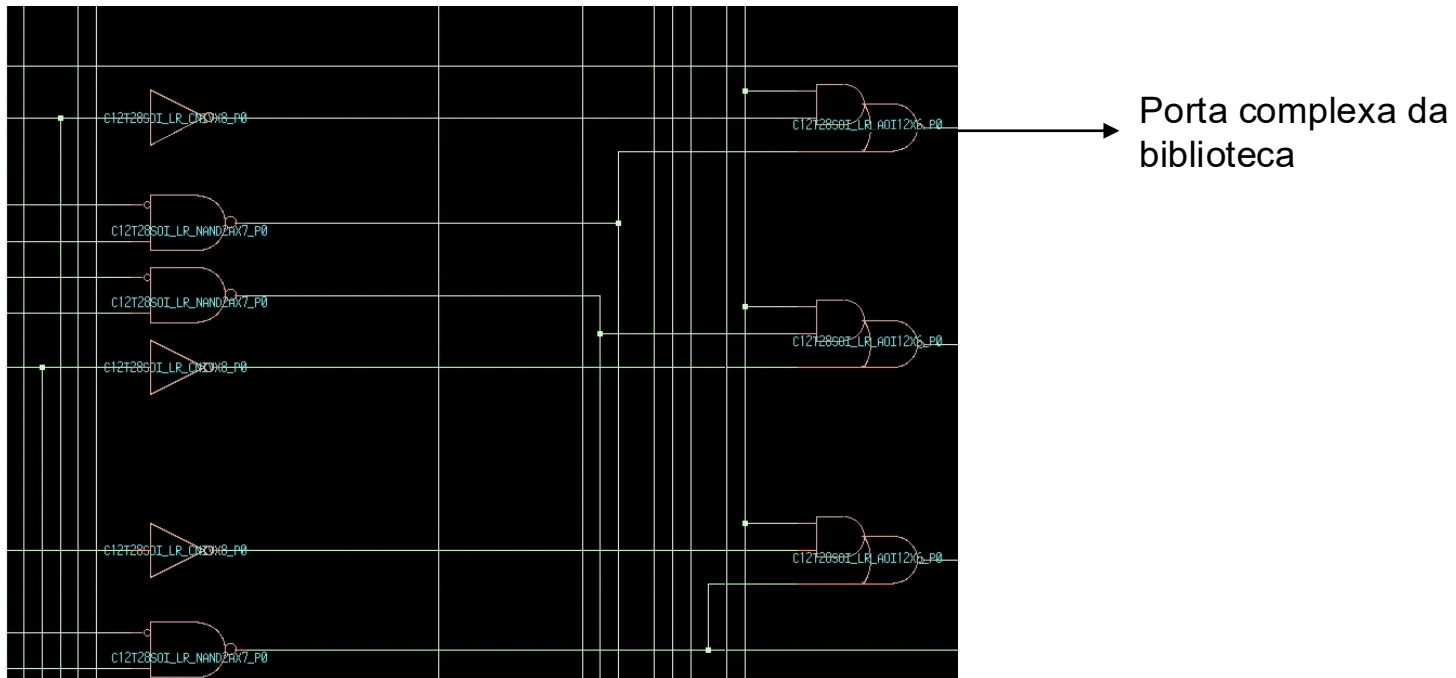
```
  : Statistics for commands executed by read_sdc:
```

```
"all_inputs"      - successful  1, failed  0 (runtime 0.00)
"all_outputs"    - successful  2, failed  0 (runtime 0.00)
"create_clock"   - successful  1, failed  0 (runtime 0.01)
"get_ports"      - successful  2, failed  0 (runtime 0.01)
"set_false_path" - successful  1, failed  0 (runtime 0.00)
"set_input_delay" - successful  1, failed  0 (runtime 0.01)
"set_load"       - successful  1, failed  0 (runtime 0.00)
"set_output_delay" - successful  1, failed  0 (runtime 0.00)
"set_units"     - successful  1, failed  0 (runtime 0.00)
```

```
read_sdc completed in 00:00:00 (hh:mm:ss)
```

Comandos de síntese

- **syn_generic** - síntese genérica da descrição RTL
- **syn_map** - mapeamento da lógica para as células da biblioteca
- **syn_opt** - otimização do circuito mapeado visando atender às restrições de área, temporização e potência.



Relatório de área

```
=====
```

Instance	Module	Cell-Count	Cell-Area	Net-Area	Total-Area
NanoCPU	NA	732	1100.294	457.726	1558.021
R_IR	Reg16bit_98	15	62.179	0.000	62.179
R_PC	Reg16bit_97	9	35.578	0.000	35.578
genblk1_0.reg_inst	Reg16bit	17	70.829	0.000	70.829
genblk1_1.reg_inst	Reg16bit_101	17	70.829	0.000	70.829
genblk1_2.reg_inst	Reg16bit_100	17	70.829	0.000	70.829
genblk1_3.reg_inst	Reg16bit_99	17	70.829	0.000	70.829

Relatório de atraso

Path 1: **MET** (6 ps) Setup Check with Pin genblk1_3.reg_inst/Q_reg_15/CP->TI

Group: ck

Startpoint: (R) R_IR/Q_reg_0/CP

Clock: (R) ck

Endpoint: (F) genblk1_3.reg_inst/Q_reg_15/TI

Clock: (R) ck

	Capture	Launch
Clock Edge:+	2000	0
Src Latency:+	0	0
Net Latency:+	0 (I)	0 (I)
Arrival:=	2000	0
Setup:-	198	
Required Time:=	1802	
Launch Clock:-	0	
Data Path:-	1796	
Slack:=	6	

Relatório de power

- Componentes do power: leakage, interno, chaveamento

```
@genus:root: 14> report_power -unit mW
Info      : Joules engine is used. [RPT-16]
          : Joules engine is being used for the command report_power.
Instance: /NanoCPU
Power Unit: mW
PDB Frames: /stim#0/frame#0
```

Category	Leakage	Internal	Switching	Total	Row%
memory	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
register	2.28975e-03	1.93080e-01	1.80885e-02	2.13459e-01	31.13%
latch	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
logic	6.20109e-03	1.70096e-01	2.63214e-01	4.39511e-01	64.09%
bbox	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
clock	0.00000e+00	0.00000e+00	3.28050e-02	3.28050e-02	4.78%
pad	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
pm	0.00000e+00	0.00000e+00	0.00000e+00	0.00000e+00	0.00%
Subtotal	8.49084e-03	3.63176e-01	3.14107e-01	6.85774e-01	100.00%
Percentage	1.24%	52.96%	45.80%	100.00%	100.00%

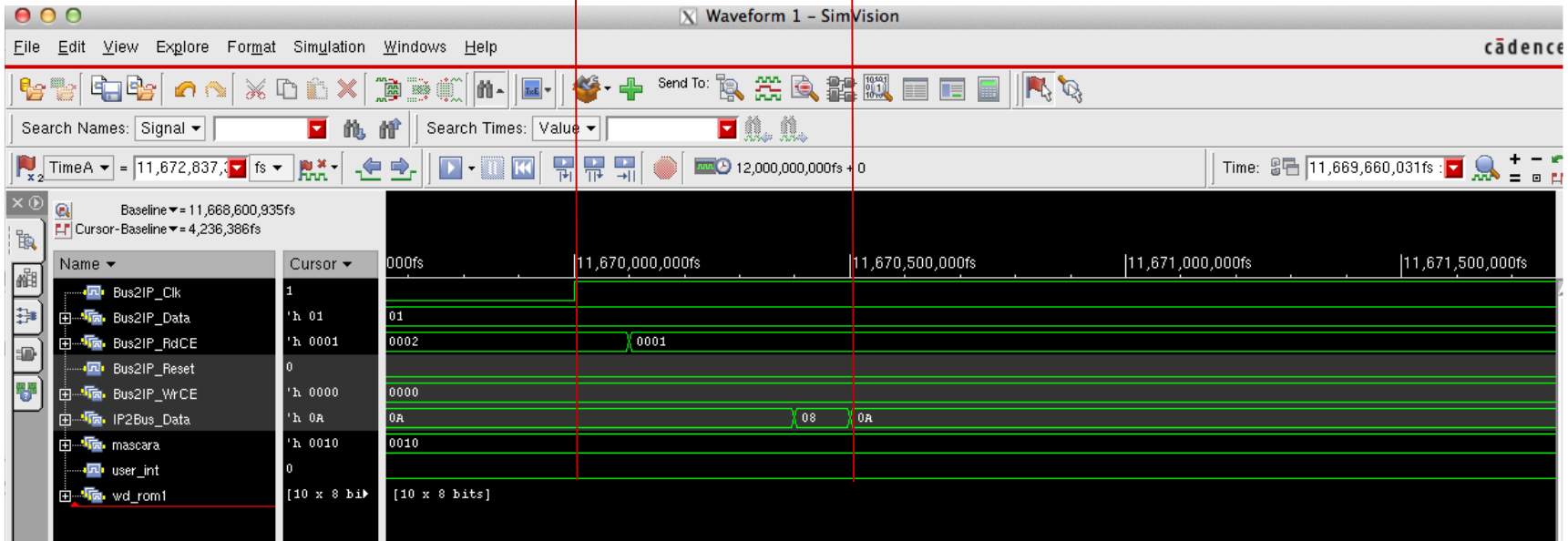
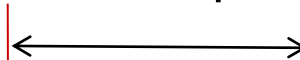
(4) Simulação pós-síntese lógica

Simula o netlist com a descrição das portas lógicas

Exemplo de script:

```
-smartorder -work work -V93 -top user_logic_tb -gui -access +rw  
/soft64/design-kits/stm/65nm-cmos065_536/CORE65GPSVT_5.1/behaviour/verilog/CORE65GPSVT.v  
/soft64/design-kits/stm/65nm-cmos065_536/CLOCK65GPSVT_3.1/behaviour/verilog/CLOCK65GPSVT.v  
../../synthesis/layout/busca_padrao.v  
../../tb/tb_padrao.vhd
```

Atraso de porta lógica

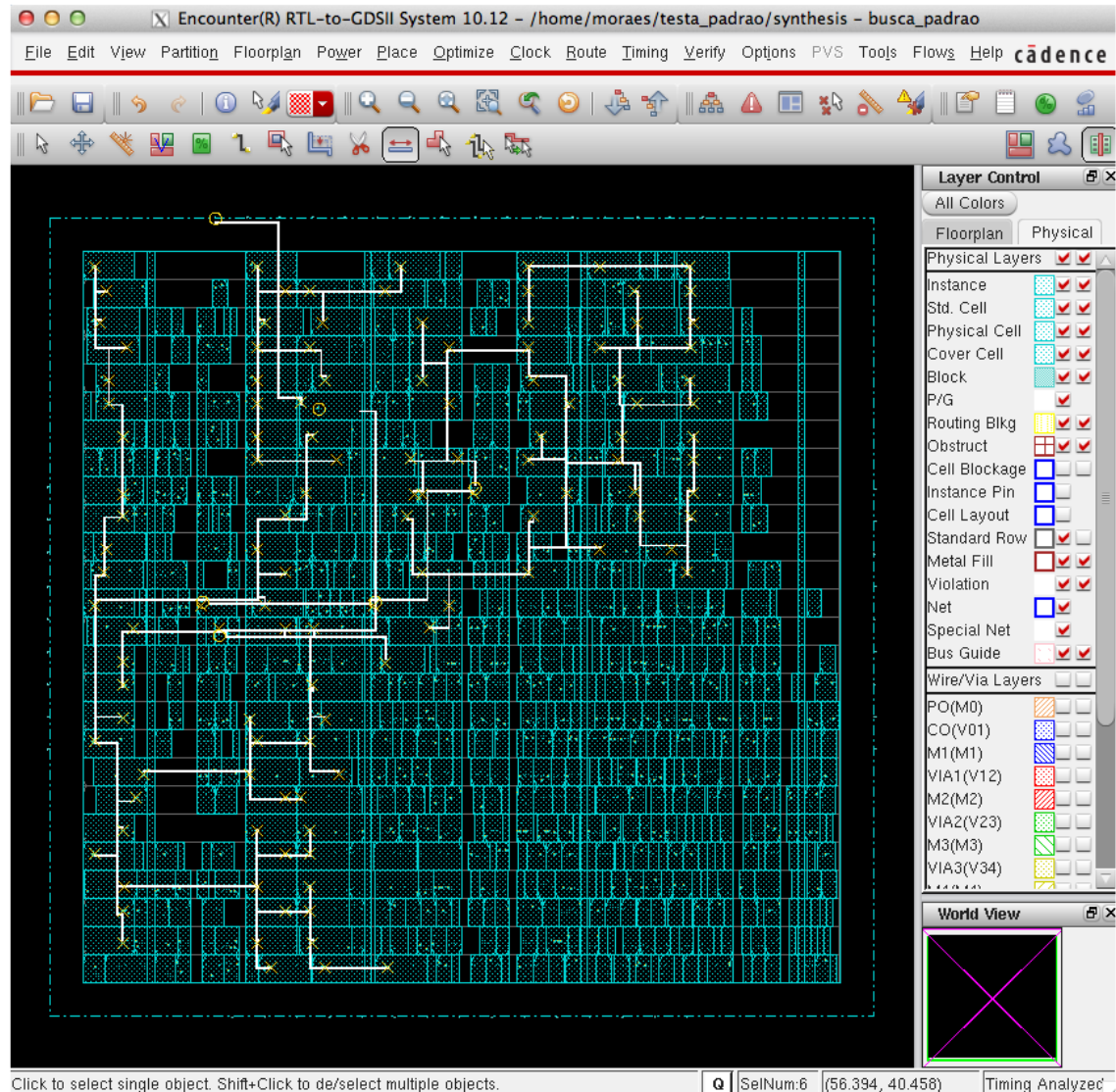


(5) Posicionamento e Roteamento

Posicionamento das células

Etapas importantes neste processo:

- alimentação
- árvore de clock (em destaque)

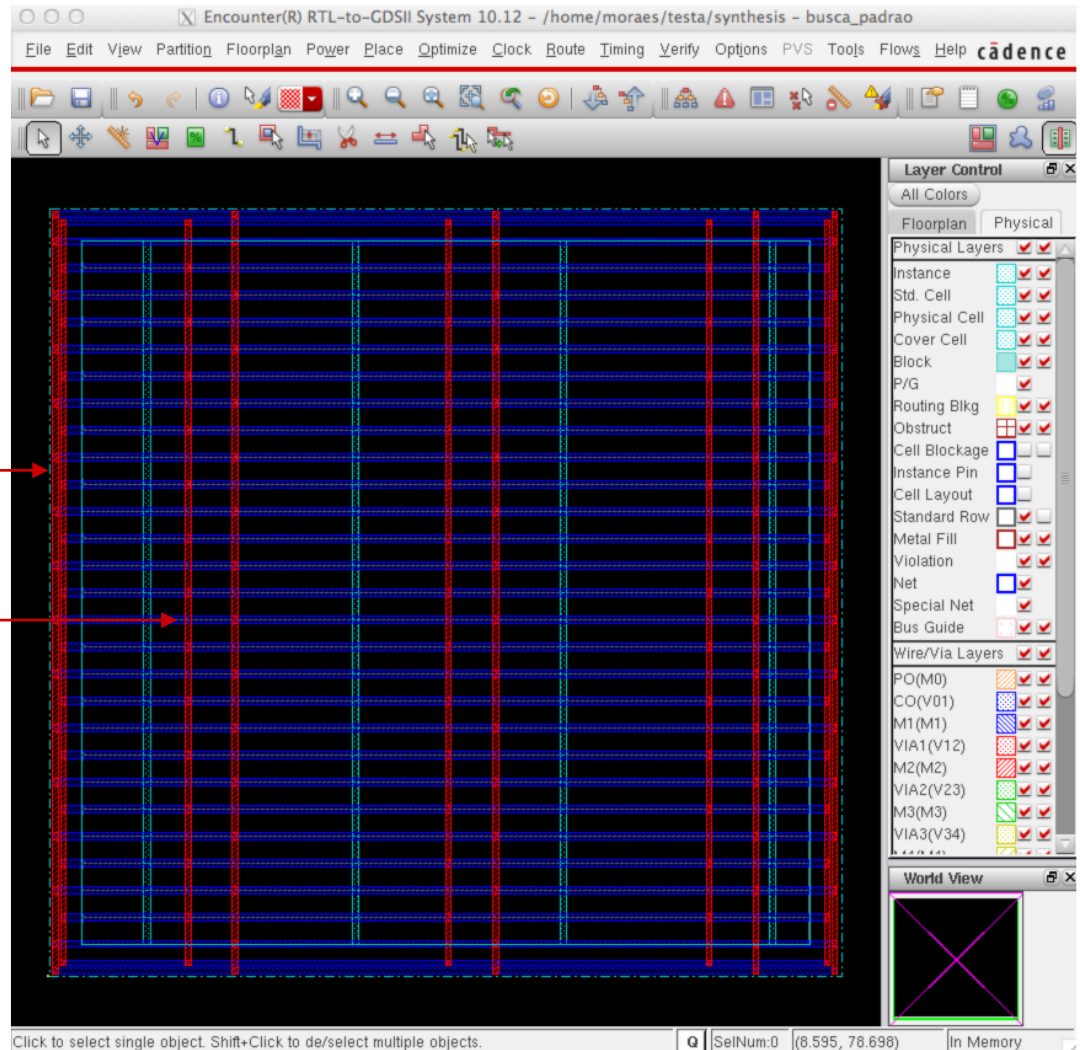


Definição do anel de alimentação e stripes

- **Stripes**: evitam ruídos nas linhas de alimentação (IR drop)

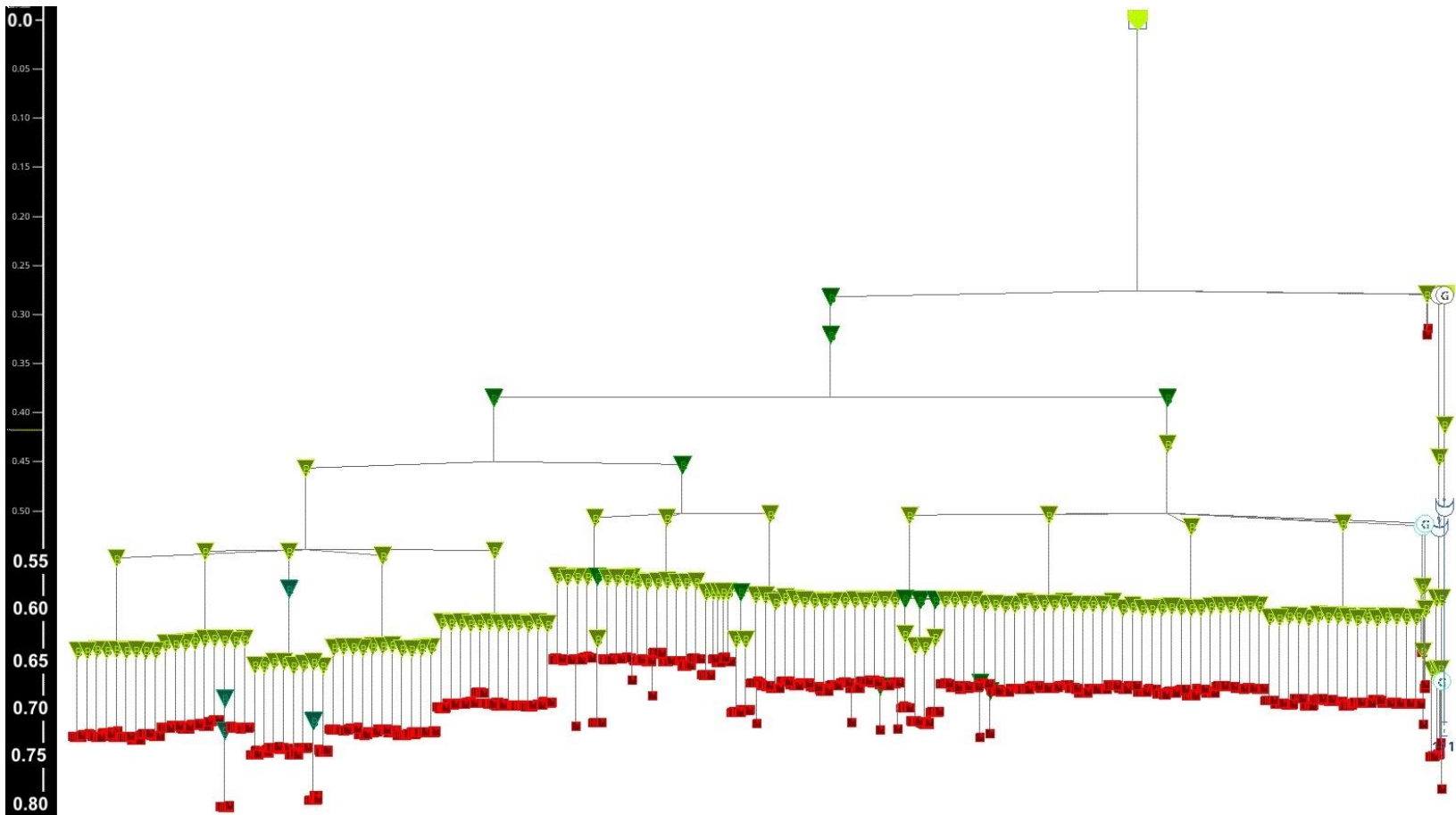
Anel de alimentação

stripes

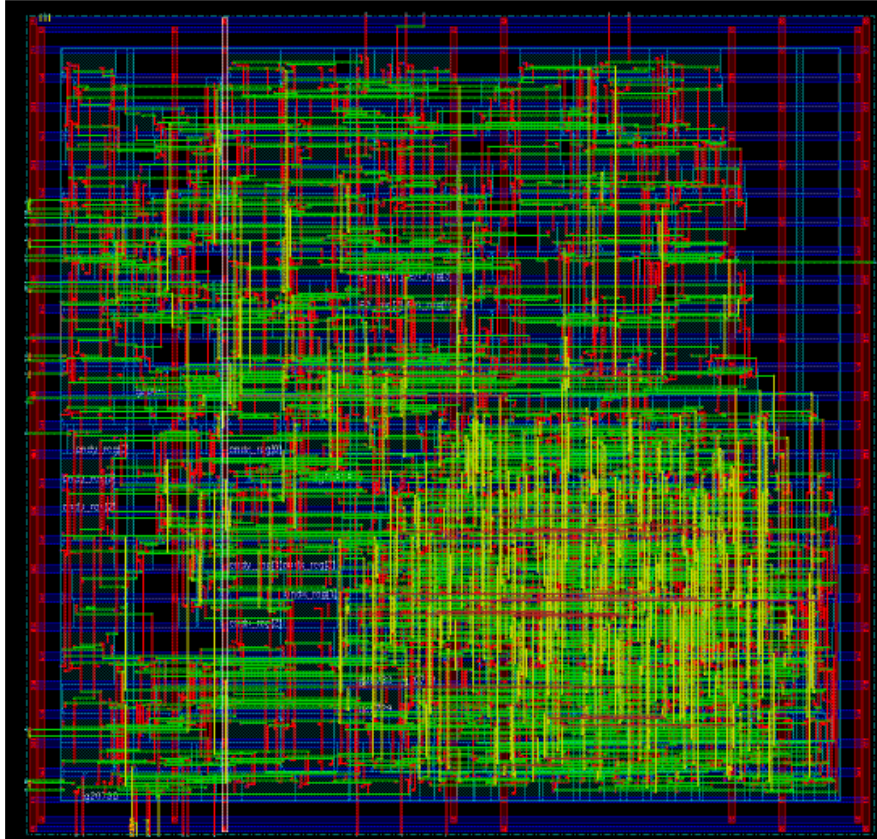


Exemplo de análise de timing do clock

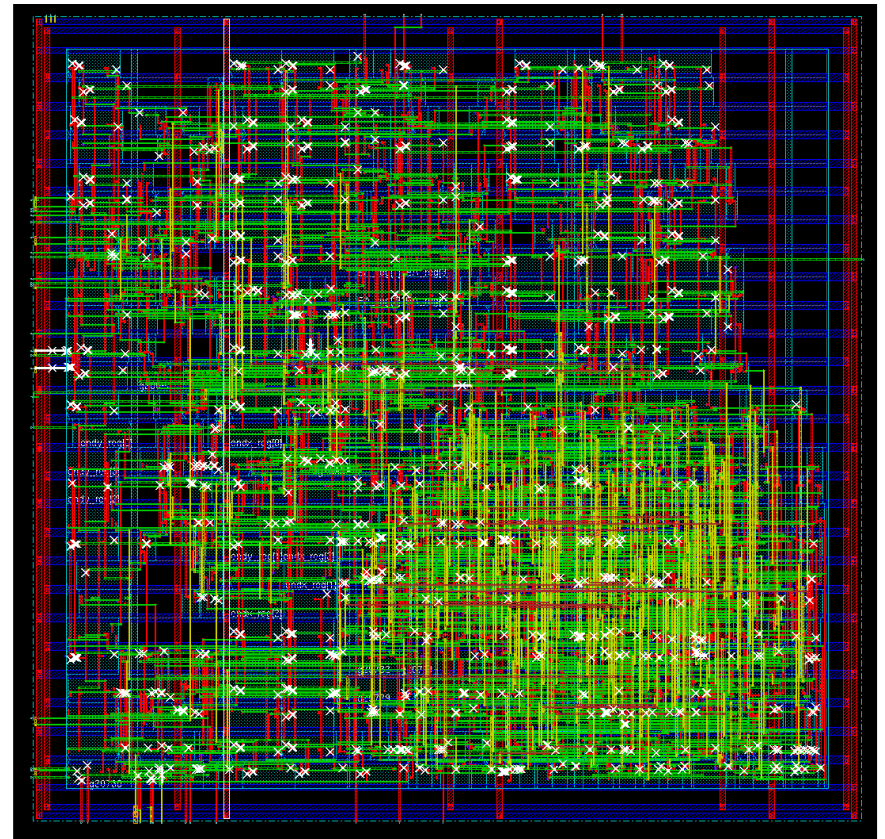
- O ponto de entrada é o PAD do clock, e as extremidades os buffers que alimentam os flip-flops
- Objetivo desta etapa: minimizar o *skew*



Roteamento

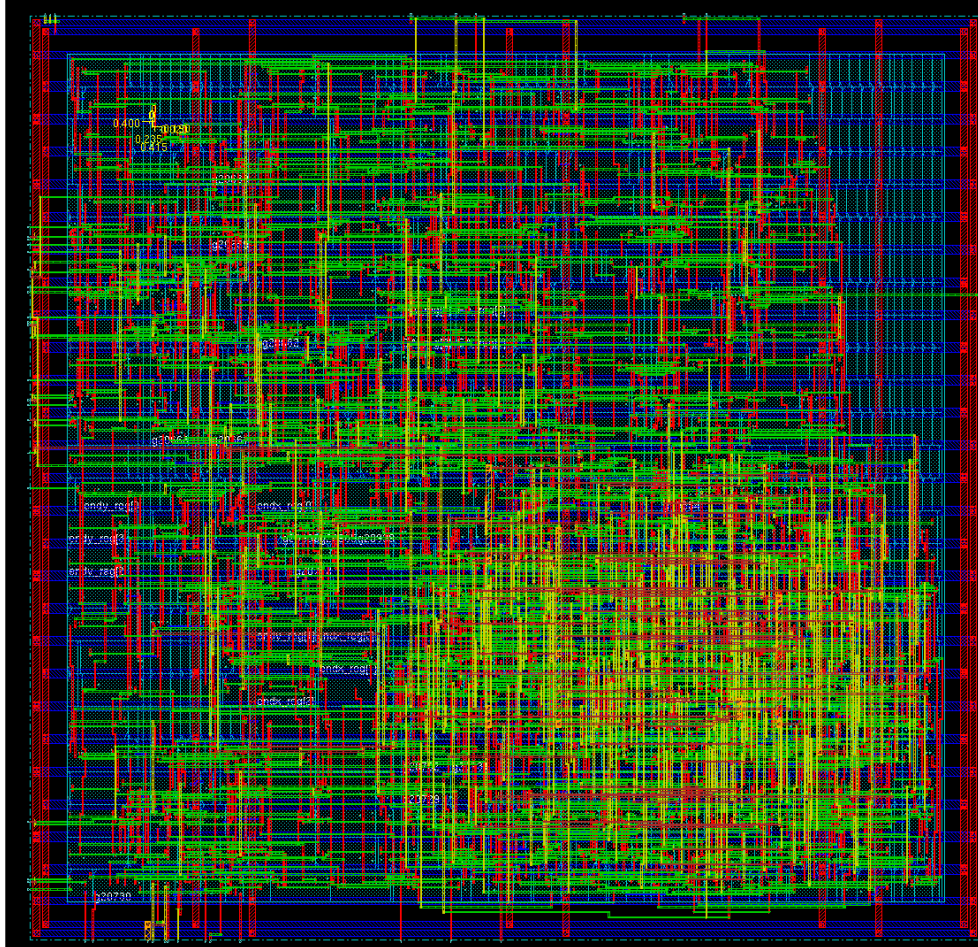


Roteamento inicial



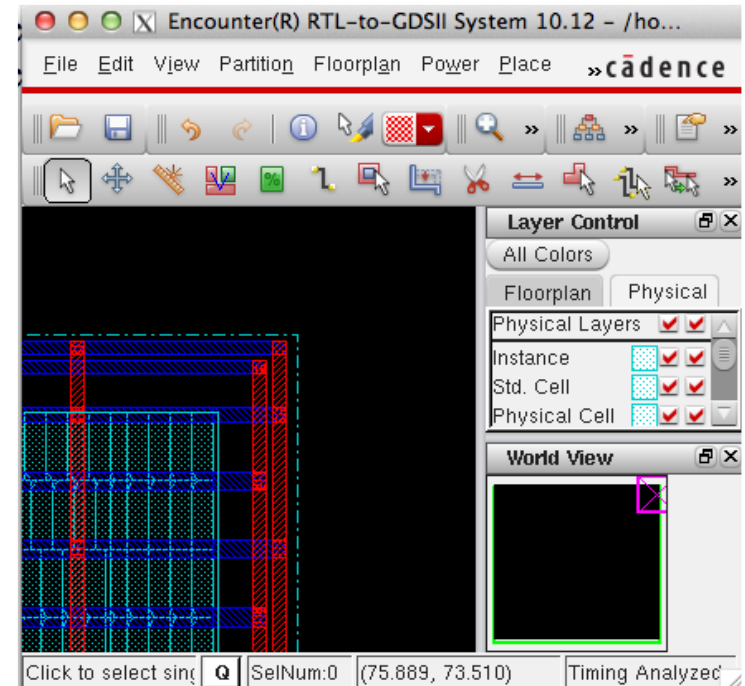
Verificação de DRC e timing

Roteamento



Detalhes:

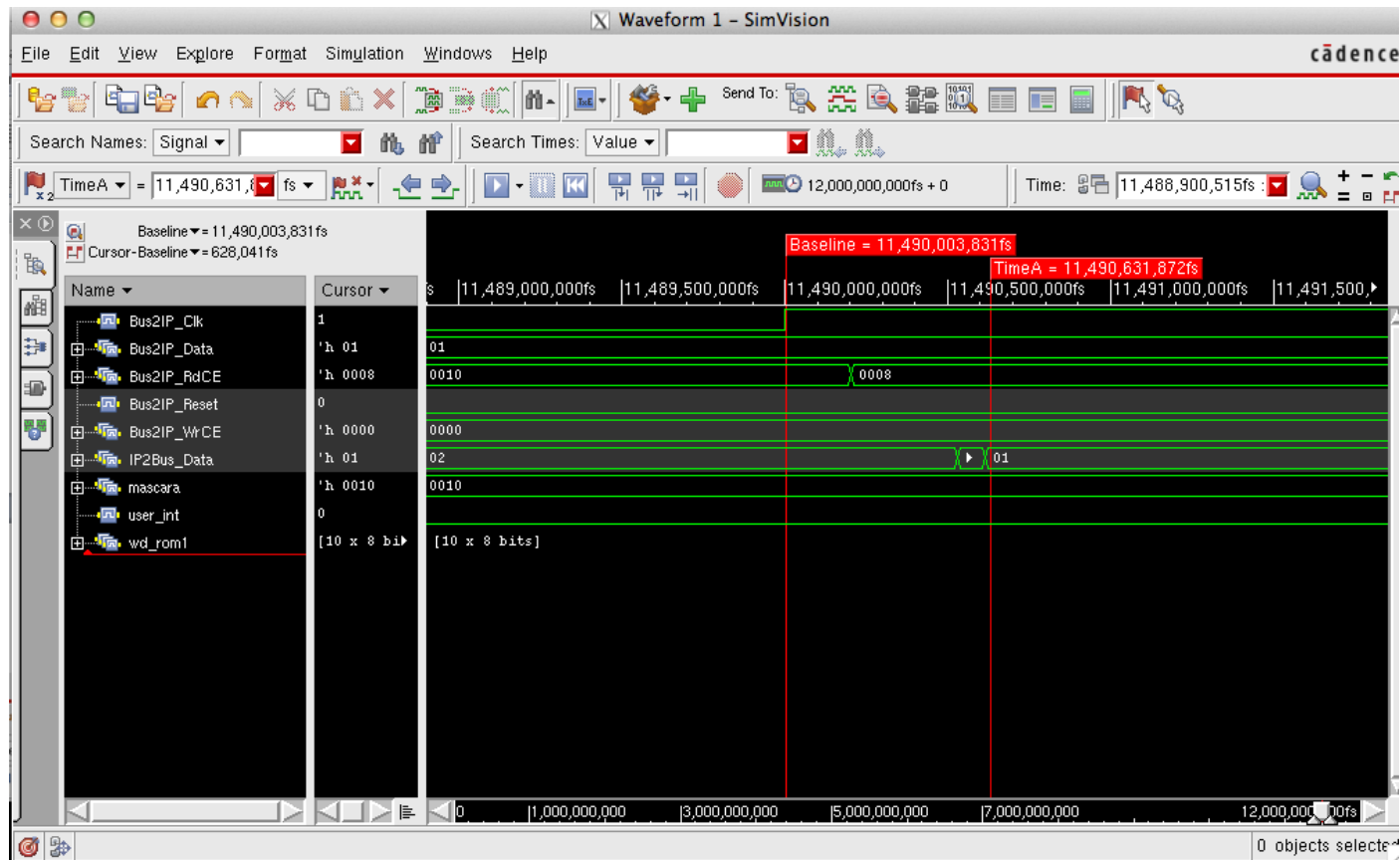
- filler cells (preenchem espaços vazios)
- alimentação



Roteamento final

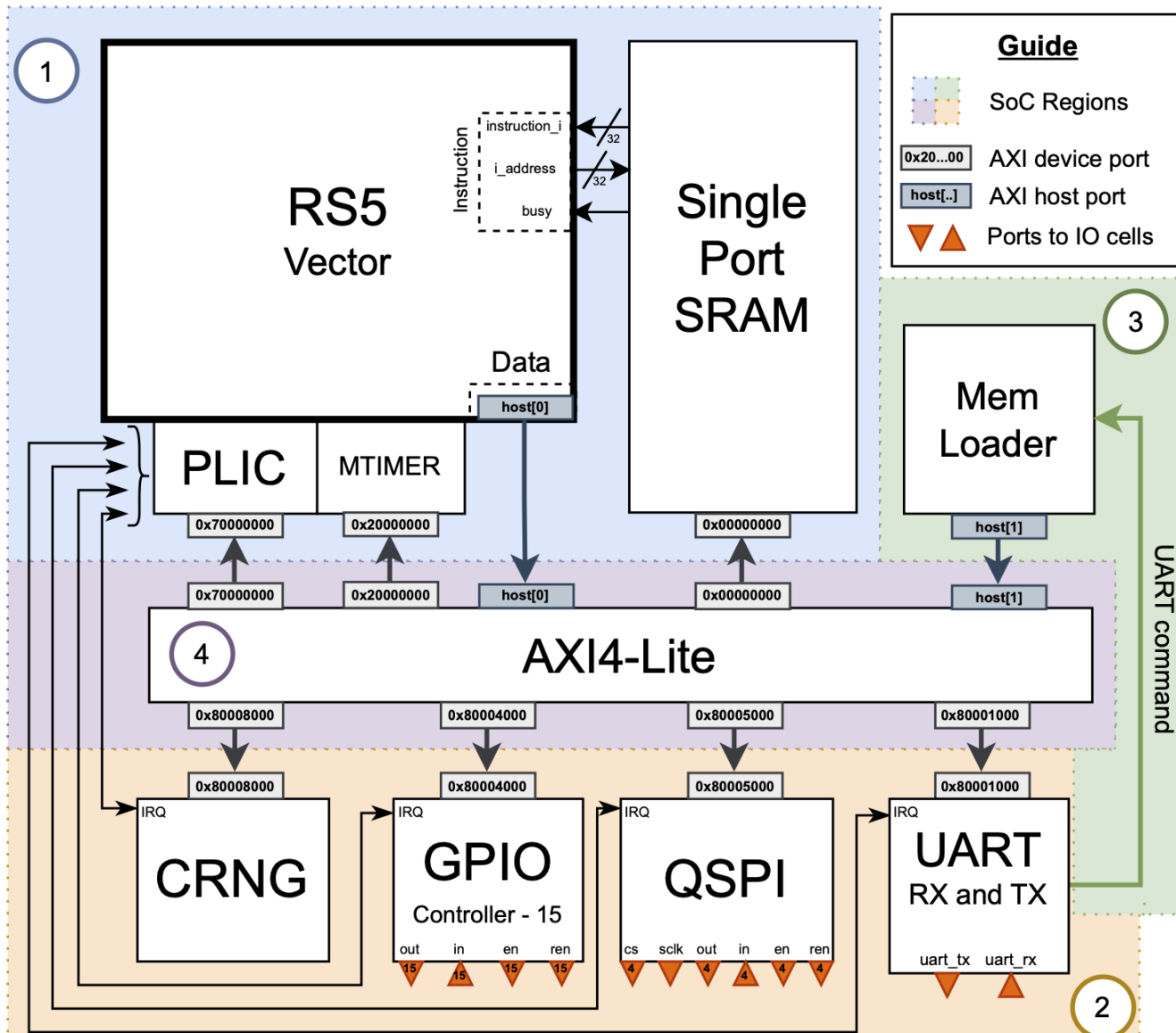
(6) Extração de parasitas e simulação pós-layout

- Arquivo com as capacitâncias de roteamento
 - Formato SDF
- Permite obter uma estimativa precisa do atraso do circuito



EXEMPLO DE PROJETO COMPLETO

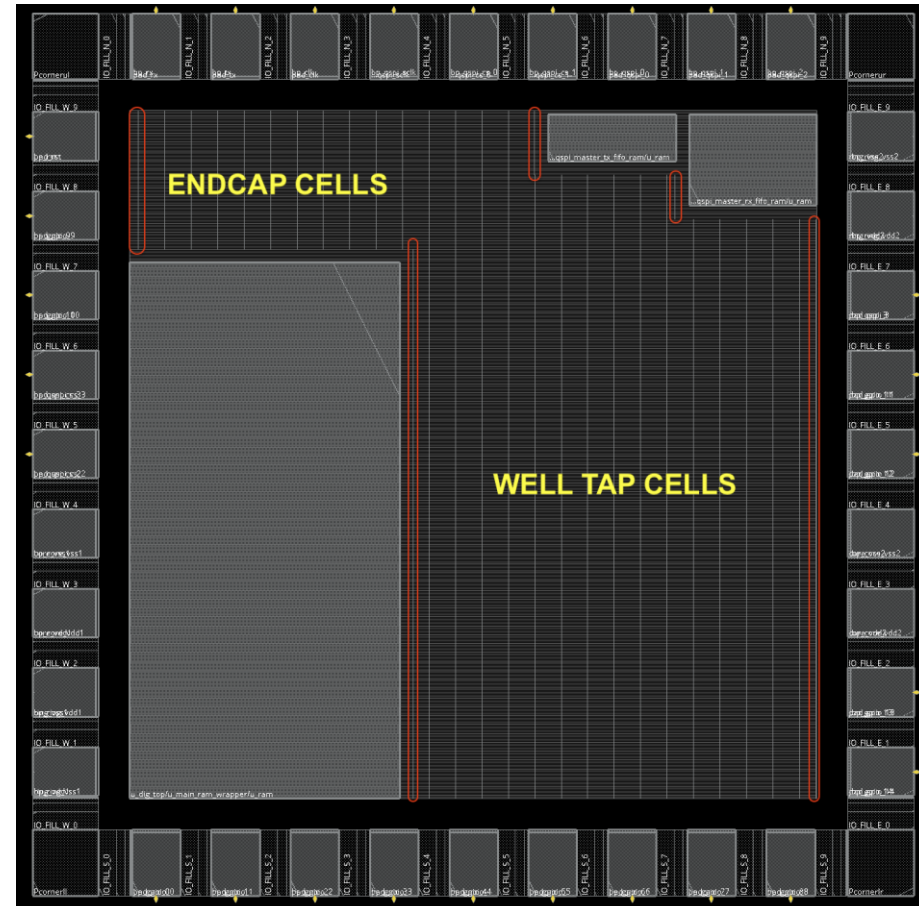
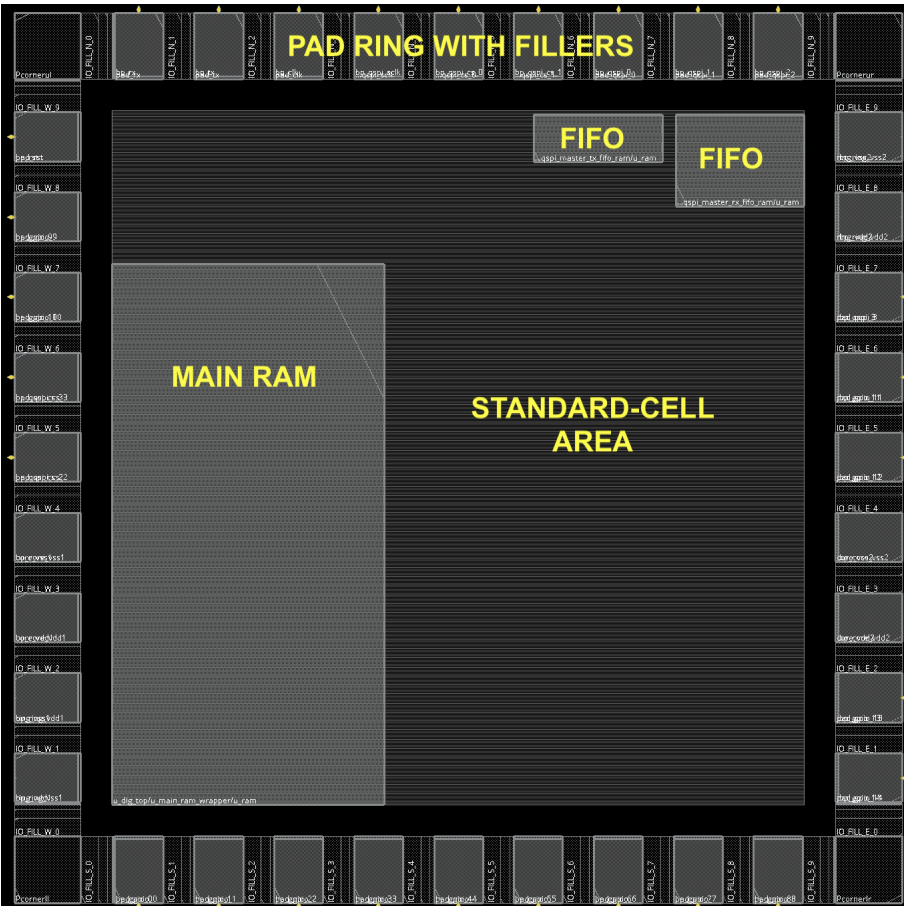
SoC com processador RISC-V RS5 vetorial – 65nm TSMC



Floorplanning inicial com área de PADs, memórias, std-cells

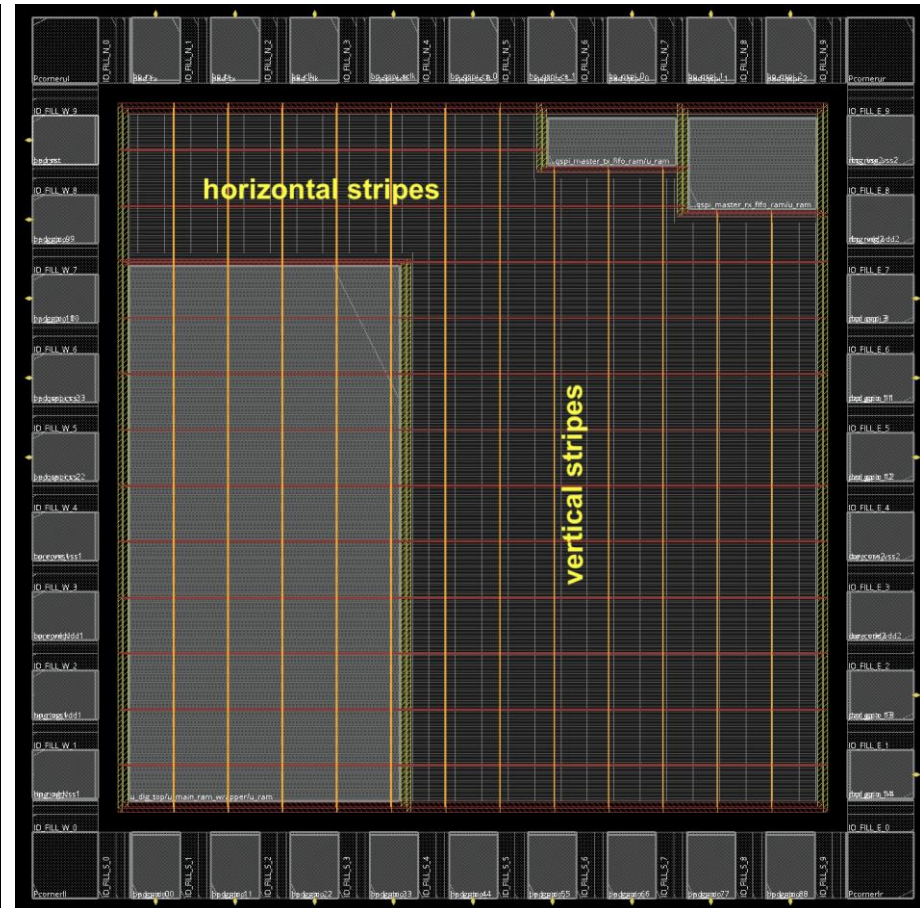
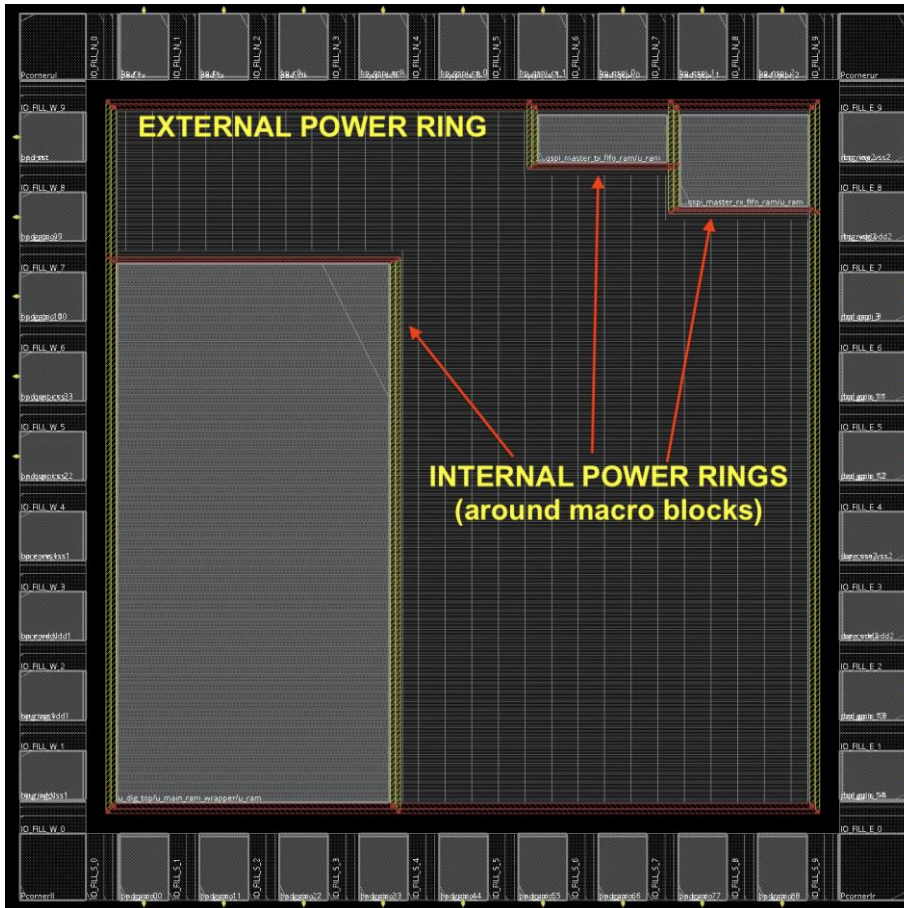
Inserção de células físicas:

- ENDCAPs – finalizam linhas de células
- WELL TAPS – polarização de substrato



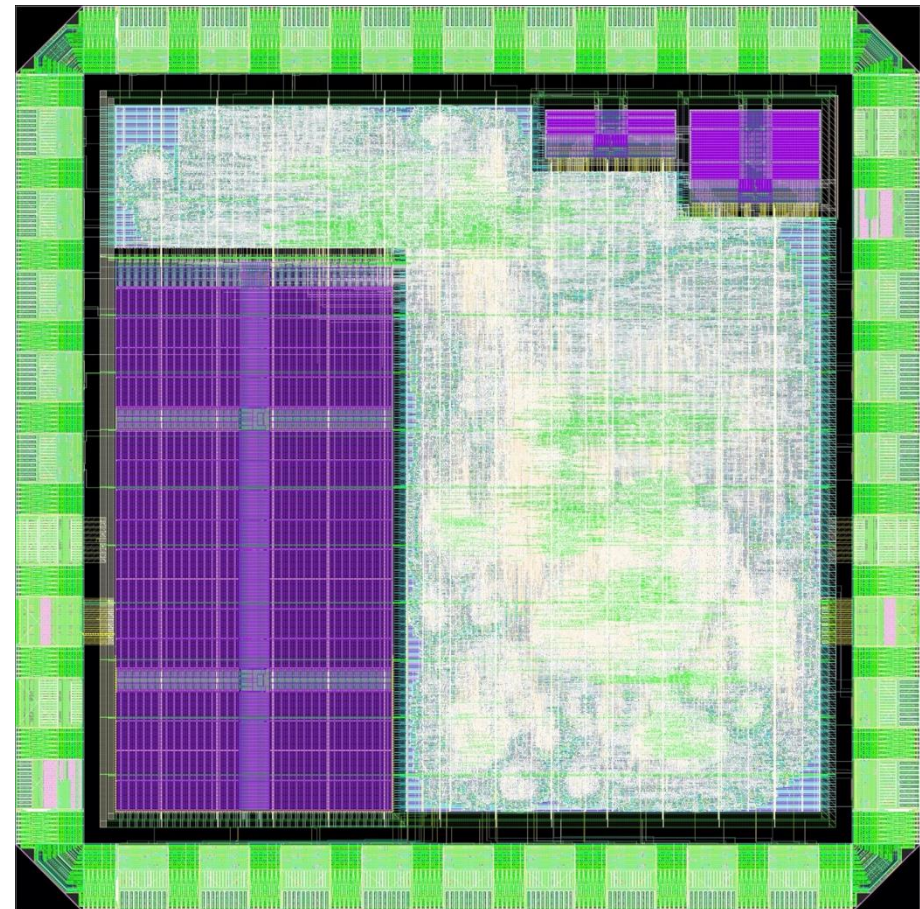
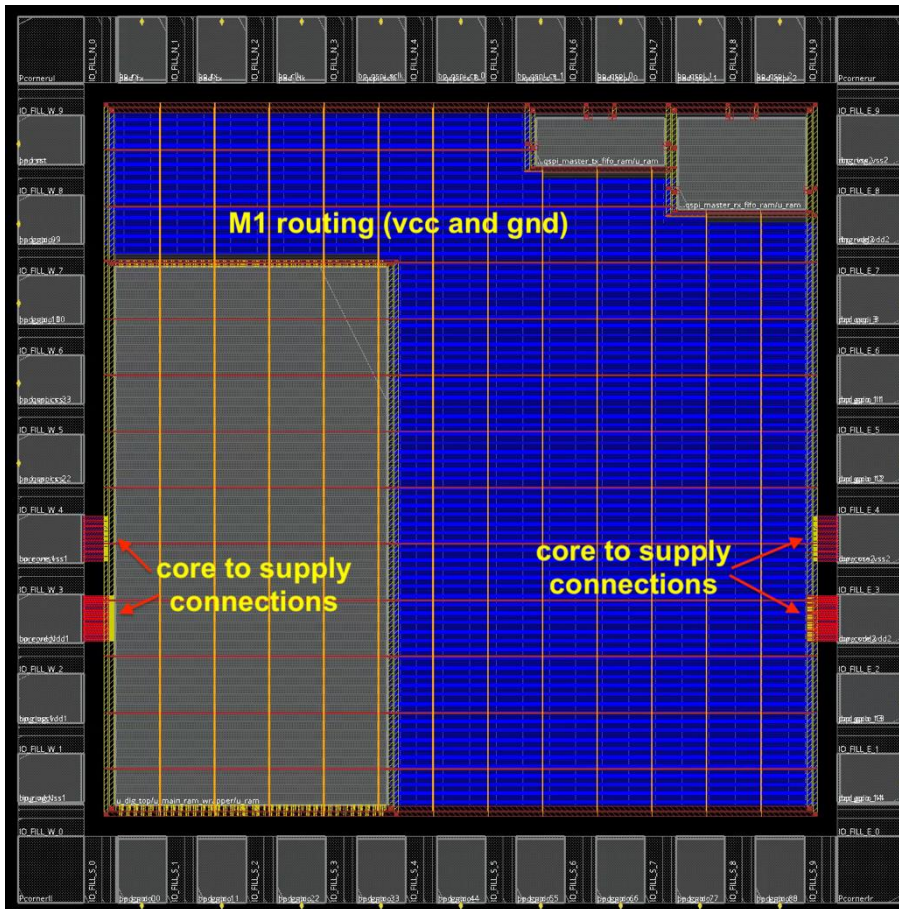
Anel de alimentação ao redor de cada bloco e ao redor do circuito

Inserção de stripes verticais e horizontais



Roteamento de alimentação – nas standard cells e do ring externo de alimentação aos PADs

Síntese lógica e síntese física das standard-cells e roteamento entre os blocos e com os PADs



Executar signoff e exportar GDSII (formato do layout)

- análise de temporização em todos os corners (mais importante que a simulação)

```
-----
time_design_signoff Summary
-----
```

Setup mode	all	default	reg2cgate	reg2reg
WNS (ns):	0.001	6.166	1.149	0.001
TNS (ns):	0.000	0.000	0.000	0.000
Violating Paths:	0	0	0	0

analysis_view_lp08v_125c_capwst_slowest				
	0.001	6.166	1.149	0.001
	0.000	0.000	0.000	0.000
	0	0	0	0

analysis_view_lp20v_25c_captyp_nominal				
	1.528	6.764	3.961	1.528
	0.000	0.000	0.000	0.000
	0	0	0	0

analysis_view_lp32v_m40c_capbst_fastest				
	2.289	7.099	5.357	2.289
	0.000	0.000	0.000	0.000
	0	0	0	0

TNS – total negative slack
Setup OK

slowest corner (1.08 V, 125°)

typical corner (1.20 V, 25°)

fastest corner (1.32 V, -40°)

```
-----
```

DRVs	Signal nets		Clock nets	
	Nr nets (terms)	Worst Vio	Nr nets (terms)	Worst Vio
max_cap	0 (0)	0	0 (0)	0
max_tran	0 (0)	0	0 (0)	0
max_fanout	1 (1)	-33	0 (0)	0

```
-----
```

```
-----
```

Hold mode	all	default	reg2cgate	reg2reg
WNS (ns):	0.100	N/A	0.551	0.100
TNS (ns):	0.000	N/A	0.000	0.000
Violating Paths:	0	0	0	0

analysis_view_lp08v_125c_capwst_slowest				
	0.293	N/A	1.667	0.293
	0.000	N/A	0.000	0.000
	0	0	0	0

analysis_view_lp20v_25c_captyp_nominal				
	0.166	N/A	0.930	0.166
	0.000	N/A	0.000	0.000
	0	0	0	0

analysis_view_lp32v_m40c_capbst_fastest				
	0.100	N/A	0.551	0.100
	0.000	N/A	0.000	0.000
	0	0	0	0

```
-----
```

Hold OK

DRC (Design Rule Check)

- corretude do layout
- 3 DRCs executados: retângulos, antenas e wirebond

--- DRC SUMMARY

```
---  
...  
RULECHECK OD.DN.xx..... TOTAL Result Count = 6  
RULECHECK PO.DN.2 ..... TOTAL Result Count = 9  
RULECHECK Mx.DN.1 ..... TOTAL Result Count = 21  
RULECHECK DMx|.R.1 ..... TOTAL Result Count = 7  
RULECHECK ESD.22g ..... TOTAL Result Count = 30  
...  
TOTAL CPU Time: 718  
TOTAL REAL Time: 38  
TOTAL Original Layer Geometries: 11114549 (86000152)  
TOTAL DRC RuleChecks Executed: 1808  
TOTAL DRC Results Generated: 73 (883)
```

Erros de densidade – DN/DM -
(corrigidos no final do processo), ou
erros que são aceitos pela fábrica
(ESD.22.g)

86 milhões de retângulos

--- ANTENNA SUMMARY

```
---  
TOTAL CPU Time: 56  
TOTAL REAL Time: 6  
TOTAL Original Layer Geometries: 8388351 (57794164)  
TOTAL DRC RuleChecks Executed: 25  
TOTAL DRC Results Generated: 0 (0)
```

Verificação de antenas - verifica se as
interconexões conectadas aos gates dos transistores
podem acumular carga durante a fabricação, evitando
danos ao óxido de gate.

--- WIREBOND SUMMARY

```
---  
TOTAL CPU Time: 43  
TOTAL REAL Time: 6  
TOTAL Original Layer Geometries: 2699486 (46221642)  
TOTAL DRC RuleChecks Executed: 205  
TOTAL DRC Results Generated: 0 (0)
```

Wirebond verifica se os PADs e as regiões próximas à
borda do chip atendem às regras físicas necessárias para
a conexão por fios de bonding, garantindo espaçamento,
área de contato e afastamento adequados para o
encapsulamento.

2,75 milhões de transistores (todo o circuito, com as memórias e PADs)

LVS (Layout versus Schematic) – garante que o layout seja equivalente HDL

```


*****
OVERALL COMPARISON RESULTS
*****

```

```

          #          #####
          #          #          #
#      #          #      CORRECT      #
#  #          #          #
#          #####

```



Warning: Unbalanced smashed mosfets were matched.
Warning: Ambiguity points were found and resolved arbitrarily.

NUMBERS OF OBJECTS AFTER TRANSFORMATION

```

-----

```

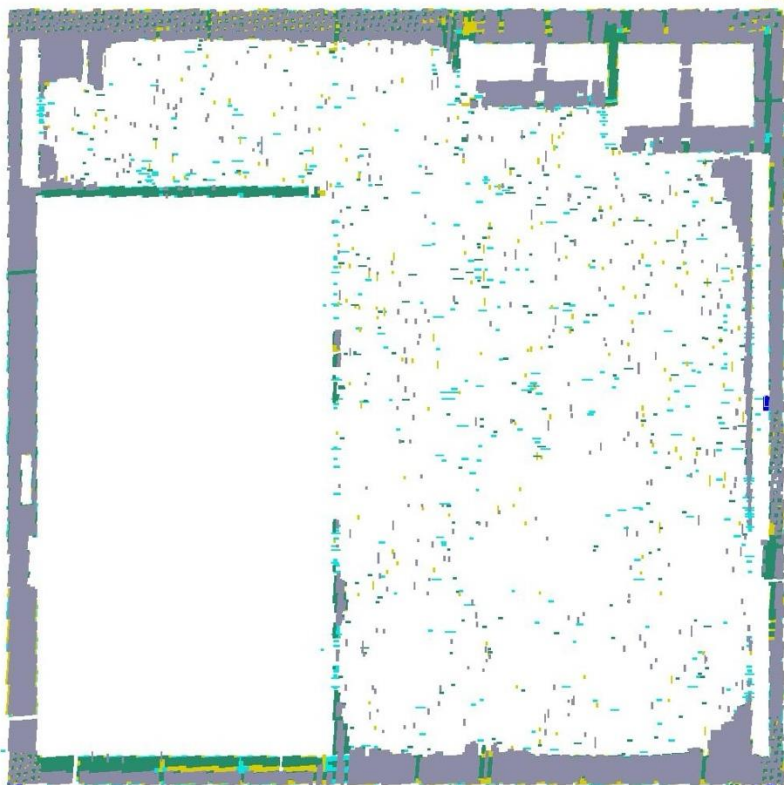
	Layout	Source	Component Type
	-----	-----	-----
Ports:	33	33	
Nets:	1103843	1103843	
Instances:	1658204	1658204	MN (4 pins)
	1098423	1098423	MP (4 pins)
	146	146	rppolywo (2 pins)
	429	429	D (2 pins)
	-----	-----	
Total Inst:	2757202	2757202	

2,75 milhões de transistores (todo o circuito, com as memórias e PADs)

Para fabricar? Garantir que todas as camadas tenham densidade uniforme sobre circuito.

A última etapa do processo de projeto é a inserção automática de dummy retângulos sobre o circuito

Dummy poly e difusão



Dummy metais

