4. Full-Custom Analog Design Methodology

Francesc Serra Graells
francesc.serra.graells@uab.cat
Departament de Microelec\"tronica i Sistemes Elec\"tronics
Universitat Aut\"onoma de Barcelona

paco.serra@imb-cnm.csic.es
Integrated Circuits and Systems
IMB-CNM(CSIC)
1. Device Sizing
2. Process and Mismatching Simulation
3. The Art of Analog Layout
4. Physical Verification
5. Parasitics Extraction
6. DFM Techniques
1 Device Sizing

2 Process and Mismatching Simulation

3 The Art of Analog Layout

4 Physical Verification

5 Parasitics Extraction

6 DFM Techniques
Full-Custom Analog IC Design Flow

From circuit idea to layout masks...

EDA-assisted methodology

Schematic flow

Physical flow

- System-level schematic entry
- Architecture HDL simulation
- Block HDL specification
- Circuit-level schematic entry
- Automatic circuit optimization
- PCell-based layout entry
- Design rule checker
- Layout versus schematic
- Parasitics extraction
- Post-layout simulation
- Tape-out

Mask making
Wafer processing
Screening
Dicing
Packaging
Full-Custom Analog IC Design Flow

From circuit **idea to layout** masks...

**EDA-assisted methodology**

- From circuit idea to layout masks...
- **Device sizing** is an intermediate key point!

EDA-assisted methodology:

- System-level schematic entry
- Architecture HDL simulation
- Block HDL specification
- Circuit-level schematic entry
- Automatic circuit optimization

M × \( \frac{W}{L} \)

- PCell-based layout entry
- Design rule checker
- Layout versus schematic
- Parasitics extraction
- Post-layout simulation
- Tape-out

Schematic flow

Physical flow

Design of Analog and Mixed Integrated Circuits and Systems

F. Serra Graells
Full-Custom Schematic Design

Architecture selection according to system and application specifications:

- Signal processing performance
- Communications standards
- Power constraints
- Testability requirements
- ... and many more!

![Diagram](attachment:image.png)
Full-Custom Schematic Design

- System modeling through any hardware description language (HDL)

```c
void cm_zinteg2lim(ARGS) {
    inp = INPUT(inp);   /* Retrieving input values */
    clk = INPUT_STATE(clk);
    pos_edge = PARAM(pos_edge);  /* Retrieving parameters */
    out_max = PARAM(out_max);
    ...
    switch (ANALYSIS) {
        case TRANSIENT:
            if ((*clk_mem==ONE)&&(clk==ZERO)) { /* Neg. clk edge */
                if (pos_edge==FALSE)
                    action = SAMPLING_INTEGRATION;
            } else {
                if ((*clk_mem==ZERO)&&(clk==ONE)) { /* Pos. clk edge */
                    if (pos_edge==TRUE)
                        action = SAMPLING_INTEGRATION;
                } else { /* No clock edge */
                    action = HOLDING;
                }
            }
            ...
            switch (action) {
                case SAMPLING_INTEGRATION:
                    *inp_mem = inp;
                    out = *out_mem+*inp_mem;
                    if (out<out_min) { out = out_min; } /* Limiter */
                    if (out>out_max) { out = out_max; }
                    *out_mem = out;
                    break;
                case HOLDING:
                    out = *out_mem;
            }
    }
}
```

- Architecture description
- Simplified modeling
- Second order effects are neglected

*E.g. XSpice*
Full-Custom Schematic Design

- **HDL numerical simulation of the full system using event-based engines**
  - Architecture evaluation
  - Simulation **speed-up** by orders of magnitude
  - **Ideal** response (maximum possible performance)

```
.dsub dsm_arch vin dclk dout
asumin [%v(vin) %v(vdac)] %v(verr) msumin
.model msumin usummer(sign=[1.0 -1.0])
aki1 %v(verr) %v(vint1in) mki1
.model mki1 kgain(k=0.3)
azint1 %v(vint1in) %d(dclk) %v(vint1out) mzint1
.model mzint1 zinteg2lim(pos_edge=0 out_ic=0.0
+ out_min=-5.0 out_max=5.0)
aki2 %v(vint1out) %v(vint2in) mki2
.model mki2 kgain(k=0.7)
azint2 %v(vint2in) %d(dclk) %v(vint2out) mzint2
.model mzint2 zinteg2lim(pos_edge=0 out_ic=0.0
+ out_min=-5.0 out_max=5.0)
akff %v(vint1out) %v(vkffout) mkff
.model mkff kgain(k=2.0)
asumout [%v(vint2out) %v(vkffout)] %v(vin)
+ %v(vquantin) msumout
.model msumout usummer(sign=[1.0 1.0 1.0])
agquant %v(vquantin) %d(~dclk) %d(dout) mquant
.model mquant quant21sh(inp_th=0.0 out_ic=0 pos_edge=0
+ t_rise=1e-9 t_fall=1e-9)
adac %d(dout) %v(vdac) mdac
.model mdac dac2lsym(out_level=2.0)
.ends
```

- System-level schematic entry
- Architecture HDL simulation
- Block HDL specification
- Circuit-level schematic entry
- Automatic circuit optimization

**e.g. SpiceOpus**
4. Full-Custom Analog Design Methodology

Full-Custom Schematic Design

► Choosing most suitable **circuit technique** according to:

- **System-level schematic entry**
- **Architecture HDL simulation**
- **Block HDL specification**
- **Circuit-level schematic entry**
- **Automatic circuit optimization**

- **CMOS** technology options
- Circuit sensitivity against process, supply and temperature (**PVT**)
- **IC operation** conditions (calibration, testability...)
- **External** components available

![Diagram]

- Choosing suitable circuit technique according to:

  - **System-level schematic entry**
  - **Architecture HDL simulation**
  - **Block HDL specification**
  - **Circuit-level schematic entry**
  - **Automatic circuit optimization**

- **CMOS** technology options
- Circuit sensitivity against process, supply and temperature (**PVT**)
- **IC operation** conditions (calibration, testability...)
- **External** components available

- **System-level schematic entry**
- **Architecture HDL simulation**
- **Block HDL specification**
- **Circuit-level schematic entry**
- **Automatic circuit optimization**

- **CMOS** technology options
- Circuit sensitivity against process, supply and temperature (**PVT**)
- **IC operation** conditions (calibration, testability...)
- **External** components available

- **System-level schematic entry**
- **Architecture HDL simulation**
- **Block HDL specification**
- **Circuit-level schematic entry**
- **Automatic circuit optimization**

- **CMOS** technology options
- Circuit sensitivity against process, supply and temperature (**PVT**)
- **IC operation** conditions (calibration, testability...)
- **External** components available
**Full-Custom Schematic Design**

- **Splitting** system design problem into independent **blocks**:
  - Modeling **second order** effects due to limited block performance
  - Electrical simulation of each **block** is feasible
  - Transistor-level **final** simulation of overall system is still required!

---

**NAME_TABLE**:

<table>
<thead>
<tr>
<th>Spice_Model_Name:</th>
<th>opamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>C Function_Name:</td>
<td>cm_opamp</td>
</tr>
<tr>
<td>Description:</td>
<td>&quot;OpAmp macro&quot;</td>
</tr>
</tbody>
</table>

**PORT_TABLE**:

<table>
<thead>
<tr>
<th>Port_Name:</th>
<th>inp</th>
<th>inn</th>
<th>out</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>&quot;pos. input&quot;</td>
<td>&quot;neg. input&quot;</td>
<td>&quot;output&quot;</td>
</tr>
<tr>
<td>Direction:</td>
<td>in</td>
<td>in</td>
<td>out</td>
</tr>
<tr>
<td>Default_Type:</td>
<td>v</td>
<td>v</td>
<td>v</td>
</tr>
</tbody>
</table>

**PARAMETER_TABLE**:

<table>
<thead>
<tr>
<th>Parameter_Name:</th>
<th>G</th>
<th>GBW</th>
<th>SR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>&quot;DC OL gain&quot;</td>
<td>&quot;GBW&quot;</td>
<td>&quot;slew-rate&quot;</td>
</tr>
<tr>
<td>Data_Type:</td>
<td>real</td>
<td>real</td>
<td>real</td>
</tr>
<tr>
<td>Default_Value:</td>
<td>1000</td>
<td>1e6</td>
<td>1e6</td>
</tr>
<tr>
<td>Limits:</td>
<td>[0 -]</td>
<td>[0 -]</td>
<td>[0 -]</td>
</tr>
</tbody>
</table>

**PARAMETER_TABLE**:

<table>
<thead>
<tr>
<th>Parameter_Name:</th>
<th>out_min</th>
<th>out_max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description:</td>
<td>&quot;lower out limit&quot;</td>
<td>&quot;upper out limit&quot;</td>
</tr>
<tr>
<td>Data_Type:</td>
<td>real</td>
<td>real</td>
</tr>
<tr>
<td>Default_Value:</td>
<td>0</td>
<td>5.0</td>
</tr>
<tr>
<td>Limits:</td>
<td>[0 5.0]</td>
<td>[0 5.0]</td>
</tr>
</tbody>
</table>

---

**System-level schematic entry**

**Architecture HDL simulation**

**Block HDL specification**

**Circuit-level schematic entry**

**Automatic circuit optimization**

---

**Opamp** macro model
Full-Custom Schematic Design

Choosing the particular circuit topology for each system block:

- Separated block tests
- Target specs from previous step
  e.g. \{G, GBW, SR\}
- Fast and accurate electrical simulations
- Several topologies can be easily investigated
- Inter-block coupled effects not covered!

e.g. fully-differential cascode and folded OpAmp
Full-Custom Schematic Design

Device size optimization at block level:

e.g. OpAmp automatic optimization

```
optimize
  optimize 0  @m1:xopamp[w] low 6u high 120u initial 32u
  optimize 1  @m6:xopamp[m] low 1 high 10 initial 8
  optimize 3  @ccomp:xopamp[w] low 25u high 250u
...
```

```
  analysis 25 ac dec 50 10 10e6
  analysis 26 let gmag=20*log10(mag(v(vout)))
  analysis 27 let gph=phase(v(vout))
  analysis 28 cursor c right gmag 0
  analysis 29 let gbw=abs(frequency[%c])/1e6
  analysis 30 let pm=180+gph[%c]
...
```

```
  analysis 46 tran ln 5u
  analysis 47 cursor c right vout 2.1
  analysis 48 let t1=time[%c]
  analysis 49 cursor c right vout 2.9
  analysis 50 let t2=time[%c]
  analysis 51 let srpos=0.8/(t2-t1)*1e-6
...
```

```
  implicit 0 op2.pd lt 1.5
  implicit 1 op2.area lt 0.025
  implicit 4 ac2.pm gt 60
  implicit 5 tran2.srpos gt 12
  cost 1/tran2.srneg+1/tran2.srpos+abs(60-ac2.pm)
  method genetic elitism yes maxgen 1000
```
MOSFET Sizing Analog Guidelines

- **Transistor (also other devices) matching ratios:**
  
  \[
  M \times \frac{W}{L}
  \]

- **Device output impedance:**
  
  \[
  \lambda \propto \frac{1}{L}
  \]

- **MOS transconductance:**
  
  \[
  M \frac{W}{L} \uparrow \quad g_{m,s} \uparrow
  \]

- **Bandwidth, technology mismatching and flicker noise:**
  
  \[
  C_{ox} = \frac{\varepsilon_{ox}}{t_{ox}} MWL
  \]
  
  \[
  \sigma (\Delta P) \approx \frac{A_P}{\sqrt{MWL}}
  \]
  
  \[
  \frac{dv_{n,fk}^2}{df} = \frac{K_{fk}}{MWL f}
  \]

- **Physical layout considerations:**
  
  - \( M \) integer design for an overall MW/L
  - Overall \( M \) and W/L design for a given absolute L value
  - Non-minimum channel length (L) above design rules for a given W/L
  - Multiplicity (M) design for a common channel aspect ratio (W/L)

- **Matching ratios:**
  
  - \( M1 \)
  - \( M2 \)
  - \( M1a \)
  - \( M1b \)
  - \( M2a \)
  - \( M2b \)

  - e.g. M even values for common centroid layout techniques
1. Device Sizing

2. Process and Mismatching Simulation

3. The Art of Analog Layout

4. Physical Verification

5. Parasitics Extraction

6. DFM Techniques
Process Simulation

**Global** deviations of model parameters:

- Same change in all devices of the ASIC
- Worse-case corner analysis: (t) yp, (f)ast, (s)low...

ASIC fabrication pass/fail!

---

### CNM26 process corners

```plaintext
* CNM26 process corners

.lib common
.model cmn25modn nmos LEVEL = 2
+ TDX = 380E-10 VTO = {vton} NSUB = 2.64E16 UO = {uon}
+ UCRIT = 1E4 UEXP = 6.8E-2 NFS = 7.11E11
+ DELTA = 2.20 RS = 93.8 LD = 9.13E-7 XJ = 8.24E-8
+ VMAX = 5.96E4 NEFF = 1.48 CJ = 3.50E-4 MJ = .40
+ CJSW = 5.95E-10 MJSW = .29 PB = .65
+ AF = 1.33 KF = 1e-29
.model cmn25modp pmos LEVEL = 2
+ TDX = 380E-10 VTO = {vtop} NSUB = 1.36E16 UO = {uop}
+ UCRIT = 1E4 UEXP = 1.16E-1 NFS = 6.62E11
+ DELTA = 1.82 RS = 134.9 LD = 8.10E-7 XJ = 2.78E-9
+ VMAX = 1.20E5 NEFF = 6.67E-2 CJ = 3.82E-4 MJ = .35
+ CJSW = 7.38E-10 MJSW = .39 PB = .56
+ AF = 1.33 KF = 1e-29
.model cmn25cpoly c CJ= 4.227E-4 CJSW=0.0
.endl

.lib tt
.param vton=.942
.param vtop=-1.139
.param uon=648
.param uop=213
.lib 'cnm25proc.lib' common
.endl

.lib ss
.param vton=1.1
.param vtop=-1.3
.param uon=415
.param uop=131
.lib 'cnm25proc.lib' common
.endl

.lib ff
.param vton=0.7
.param vtop=-0.9
.param uon=881
.param uop=295
.lib 'cnm25proc.lib' common
.endl
```

---

Same change in all devices of the ASIC
Process Simulation

▶ **Global** deviations of model parameters:

- Same change in all devices of the ASIC
- Worse-case corner analysis: (t)yp, (f)ast, (s)low...
- Combined corners: process/voltage/temperature (PVT)

\[ p(\text{param}) \]

ASIC fabrication pass/fail!

- **Montecarlo** statistical analysis:

\[ p(\text{param}) \]

ASIC fabrication yield!

---

* CNM26 process corners

```verbatim
.lib common
.model cnm25modn nmos LEVEL = 2
+ TDX = 380E-10 VTO = \{vton\} NSUB = 2.64E16 UO = \{uon\}
+ UCRIT = 1E4 UEXP = 6.86E-2 NFS = 7.11E11
+ DELTA = 2.20 RS = 93.8 LD = 9.13E-7 XJ = 8.24E-8
+ VMAX = 5.96E4 NEFF = 1.48 CJ = 3.50E-4 MJ = .40
+ CJSW = 5.95E-10 MJSW = .29 PB = .65
+ AF =1.33 KF =1e-29
.model cnm25modp pmos LEVEL = 2
+ TDX = 380E-10 VTO = \{vtop\} NSUB = 1.36E16 UO = \{uop\}
+ UCRIT = 1E4 UEXP = 1.16E-1 NFS = 6.62E11
+ DELTA = 1.82 RS = 134.9 LD = 8.10E-7 XJ = 2.78E-9
+ VMAX = 1.20E5 NEFF = 6.67E-2 CJ = 3.82E-4 MJ = .35
+ CJSW = 7.38E-10 MJSW = .39 PB = .56
+ AF =1.33 KF =1e-29
.model cnm25cpoly c CJ= 4.227E-4 CJSW=0.0
.endl
.lib tt
.param vton=.942
.param vtop=-1.139
.param uon=648
.param uop=213
.lib 'cnm25proc.lib' common
.endl
.lib ss
.param vton=1.1
.param vtop=1.3
.param uon=415
.param uop=131
.lib 'cnm25proc.lib' common
.endl
.lib ff
.param vton=0.7
.param vtop=-0.9
.param uon=881
.param uop=295
.lib 'cnm25proc.lib' common
.endl
```

Same change in all devices of the ASIC.

Combined corners: process/voltage/temperature (PVT).

Montecarlo statistical analysis:

Monte Carlo statistical analysis:

ASIC fabrication pass/fail!

ASIC fabrication yield!
**Mismatching Simulation**

 ► **Local** deviations of model parameters:

- Different change for **each device** of the ASIC
- **Pelgrom Law**
- **Montecarlo** statistical analysis:

![Diagram showing local deviations and yield!](image)

```
.param avto=30e-9
.param rauo=5e-8
.param rac=7.3e-8

.subckt cnm25match.lib

.param avto=30e-9
.param rauo=5e-8
.param rac=7.3e-8

.subckt cnm25modn d g s b param: w=3u l=3u + ad=0 as=0 pd=0 ps=0 m=1
.param vton=.942+randgauss(avto/sqrt(m*w*1))
.param uon=648*(1+randgauss(rauo/sqrt(m*w*1)))
.model modnlocal nmos level=2 vto={vton} uo={uon} tox=...

.param vton=.942+randgauss(avto/sqrt(m*w*1))
.param uon=648*(1+randgauss(rauo/sqrt(m*w*1)))
.model modnlocal nmos level=2 vto={vton} uo={uon} tox=...

.ends

.subckt cnm25cpoly t b param: w=30u l=30u m=1
.param cj=4.227E-4*(1+randgauss(rac/sqrt(m*w*1)))
.model cpolylocal c cj={cj} cjsw=0.0
.cpip t b cpolylocal w={w} l={l} m={m}
.ends
```
1. Device Sizing

2. Process and Mismatching Simulation

3. The Art of Analog Layout

4. Physical Verification

5. Parasitics Extraction

6. DFM Techniques
General Matching Rules

- **Unitary** elements

  - e.g. 1 : 2 ratio

  - Play with *multiplicity* only
  - Same ratio for *second order* effects also (e.g. area and perimeter ratio in caps)
  - **Larger area** for the overall array
General Matching Rules

- **Unitary** elements

- **Large area** devices

  - Technology local **granularity** (e.g. distribution of dopants)
  - **Pelgrom Law** $\propto \frac{1}{\sqrt{W L}}$
General Matching Rules

- **Unitary** elements
  - e.g. 1 : 1 ratio

- Large **area** devices

- Minimum **distance**
  - Technology global **drifts** (e.g. gate oxide thickness slope)
  - **Design rule** spacing limits
General Matching Rules

➤ **Unitary** elements

➤ **Large area** devices

➤ **Minimum distance**

➤ **Same orientation**

- **Anisotropic** materials
  (e.g. wafer crystal lattice orientation)

- Longer **routing** may be required...
General Matching Rules

- **Unitary** elements
- Large **area** devices
- Minimum **distance**
- Same **orientation**
- Same **surround**

![Diagram showing matching rules]

- **Inter-device** second order effects (e.g. parasitic RLC)
- **Larger area** for the overall array

Example ratios: 1:1:1
General Matching Rules

- **Unitary** elements
- Large **area** devices
- Minimum **distance**
- Same **orientation**
- Same **surround**
- Same **symmetry**

- **Differential** circuits (common mode interference)
- Complex **floorplan**

Interference source (thermal drift, radio coupling, power ripple, mechanical stress)
General Matching Rules

- **Unitary** elements
- **Large area** devices
- **Minimum distance**
- **Same orientation**
- **Same surround**
- **Same symmetry**

- Compensation of **linear** gradients (and non-linear at short distance)
- **Longer routing** is usually required...
Common Centroid Arrays

Difficulty to achieve for large and multiple groups of unitary elements

Unitary device element (M=1)

- E.g. A : B : C : D ratios
  8  4  2  2
Common Centroid Arrays

Difficult to achieve for **large** and **multiple groups** of unitary elements

Centroid as a **center-of-mass** concept, but with area weights...

\[
\sum_{i=1}^{M} A_i (C_i - C_0) = 0
\]

Centroid coordinates

\[
C_0 = \frac{\sum_{i=1}^{M} A_i C_i}{\sum_{i=1}^{M} A_i}
\]

If all elements are of **same area**:

\[
C_0 \equiv \frac{1}{M} \sum_{i=1}^{M} C_i
\]
Common Centroid Arrays

- Difficult to achieve for **large** and **multiple groups** of unitary elements

- Centroid as a **center-of-mass** concept, but with area weights...

\[
\sum_{i=1}^{M} A_i (C_i - C_0) = 0
\]

Centroid coordinates

\[
C_0 = \frac{\sum_{i=1}^{M} A_i C_i}{\sum_{i=1}^{M} A_i}
\]

If all elements are of **same area**:

\[
C_0 \equiv \frac{1}{M} \sum_{i=1}^{M} C_i
\]

Center-of-area is the **average** of elements positions...
Common Centroid Arrays

▼ Difficult to achieve for large and multiple groups of unitary elements

► Centroid as a center-of-mass concept, but with area weights...

▲ Centroid-based golden rules:

■ Coincidence of all centroids

Not a common centroid arrangement!
Common Centroid Arrays

- Difficult to achieve for large and multiple groups of unitary elements

- Centroid as a center-of-mass concept, but with area weights...

- **Centroid**-based golden rules:
  - Coincidence of all centroids
  - Symmetry for X and Y axes
Common Centroid Arrays

▼ Difficult to achieve for large and multiple groups of unitary elements

► Centroid as a center-of-mass concept, but with area weights...

▲ Centroid-based golden rules:

- **Coincidence** of all centroids
- **Symmetry** for X and Y axes
- **Dispersion** of groups as uniformly as possible

<table>
<thead>
<tr>
<th>A</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>B</th>
<th>A</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>B</td>
<td>D</td>
<td>C</td>
<td>B</td>
<td>A</td>
<td>A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>D</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>A</td>
<td>D</td>
<td>A</td>
<td>C</td>
<td>A</td>
<td>B</td>
<td>A</td>
</tr>
</tbody>
</table>
Common Centroid Arrays

- Difficult to achieve for large and multiple groups of unitary elements

- Centroid as a center-of-mass concept, but with area weights...

- Centroid-based golden rules:
  - Coincidence of all centroids
  - Symmetry for X and Y axes
  - Dispersion of groups as uniformly as possible
  - Compactness of overall array to ideal square shape

- e.g. A : B : C : D ratios 8 4 2 2

- A B A C A D A B
- B A D A C A B A

- A B A B
- C A D A
- A D A C
- B A B A
PCell-Based Layout

▲ Unitary elements

▲ Regular geometry

▲ Design rule compliant

e.g. CNM25 NMOSFET parameterized cell (PCell)
Decoupling Guidelines

- **Signal integrity** between analog, digital, RF... domains
Decoupling Guidelines

► **Signal integrity** between analog, digital, RF... domains

► Avoiding signal coupling through **power rails**

*guard rings against substrate noise*

*analog weak signal*

*digital strong signal*

*ground shield*

*analog*

*digital**

*diff.*

*comm.*

*power rail*
Decoupling Guidelines

- **Signal integrity** between analog, digital, RF... domains

- Avoiding signal coupling through **power rails**

- Investing peripheral free area for on-chip **decoupling capacitors**
OpAmp Layout Examples

Let's evaluate other students work....
OpAmp Layout Examples

Let's evaluate other students work....
1. Device Sizing

2. Process and Mismatching Simulation

3. The Art of Analog Layout

4. Physical Verification

5. Parasitics Extraction

6. DFM Techniques
Geometrical Rules

- Basic 2D concepts

  - e.g. MOSFET gate
  - e.g. MiM capacitor
  - e.g. signal routing
  - e.g. serpentine resistor
Geometrical Rules

- Technology rules set

e.g. 2.5um 2P2M CMOS (CNM25)
Geometrical Rules

Technology rules set

Front end of line (FEOL)

Back end of line (BEOL)

<table>
<thead>
<tr>
<th>rule</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>Design grid is 0.25um x 0.25um</td>
</tr>
<tr>
<td>1.1</td>
<td>N-well width &gt;= 8um</td>
</tr>
<tr>
<td>1.2</td>
<td>N-well spacing and notch &gt;= 8um</td>
</tr>
<tr>
<td>2.1</td>
<td>GASAD width &gt;= 2um</td>
</tr>
<tr>
<td>2.2</td>
<td>GASAD spacing and notch &gt;= 4um</td>
</tr>
<tr>
<td>2.3</td>
<td>N-well enclosure of P-plus active &gt;= 5um</td>
</tr>
<tr>
<td>2.4</td>
<td>N-well spacing to N-plus active &gt;= 5um</td>
</tr>
<tr>
<td>3.1</td>
<td>Poly0 width &gt;= 2.5um</td>
</tr>
<tr>
<td>3.2</td>
<td>Poly0 spacing and notch &gt;= 6um</td>
</tr>
<tr>
<td>3.3</td>
<td>Poly0 spacing to GASAD &gt;= 6um</td>
</tr>
<tr>
<td>4.1.a</td>
<td>Poly1 width inside GASAD &gt;= 3um</td>
</tr>
<tr>
<td>4.1.b</td>
<td>Poly1 width outside GASAD &gt;= 2.5um</td>
</tr>
<tr>
<td>4.2</td>
<td>Poly1 spacing and notch &gt;= 3um</td>
</tr>
<tr>
<td>4.3</td>
<td>GASAD extension of Poly1 &gt;= 3um</td>
</tr>
<tr>
<td>4.4</td>
<td>Poly1 extension of GASAD &gt;= 2.5um</td>
</tr>
<tr>
<td>4.5</td>
<td>Poly1 spacing to GASAD &gt;= 1.25um</td>
</tr>
<tr>
<td>4.6</td>
<td>Poly0 enclosure of Poly1 &gt;= 3um</td>
</tr>
<tr>
<td>5.1</td>
<td>N-plus enclosure of GASAD &gt;= 2.5um</td>
</tr>
<tr>
<td>5.2</td>
<td>N-plus spacing to P-plus active &gt;= 2.5um</td>
</tr>
<tr>
<td>5.3</td>
<td>N-plus spacing to Poly1 inside P-plus active &gt;= 2um</td>
</tr>
<tr>
<td>5.4</td>
<td>N-plus extension of Poly1 inside N-plus active &gt;= 1.5um</td>
</tr>
<tr>
<td>5.5</td>
<td>N-plus width &gt;= 2.5um</td>
</tr>
<tr>
<td>5.6</td>
<td>N-plus spacing and notch &gt;= 2.5um</td>
</tr>
<tr>
<td>6.1</td>
<td>Exact contact size = 2.5um x 2.5um</td>
</tr>
<tr>
<td>6.2</td>
<td>Contact spacing &gt;= 3um</td>
</tr>
<tr>
<td>6.3</td>
<td>GASAD enclosure of Contact &gt;= 1um</td>
</tr>
<tr>
<td>6.4</td>
<td>Poly1 enclosure of Contact &gt;= 1.25um</td>
</tr>
<tr>
<td>6.5</td>
<td>Poly1 Contact spacing to GASAD &gt;= 2.5um</td>
</tr>
<tr>
<td>6.6</td>
<td>Contact spacing to Poly1 inside GASAD &gt;= 2um</td>
</tr>
<tr>
<td>6.9</td>
<td>Poly0 enclosure of Contact &gt;= 4um</td>
</tr>
<tr>
<td>6.10</td>
<td>Contact spacing to Poly1 &amp; Poly0 &gt;= 4um</td>
</tr>
<tr>
<td>7.1</td>
<td>Metal1 width &gt;= 2.5um</td>
</tr>
<tr>
<td>7.2</td>
<td>Metal1 spacing and notch &gt;= 3um</td>
</tr>
<tr>
<td>7.3</td>
<td>Metal1 enclosure of Contact &gt;= 1.25um</td>
</tr>
<tr>
<td>8.1</td>
<td>Exact via size = 3um x 3um</td>
</tr>
<tr>
<td>8.2</td>
<td>Via spacing &gt;= 3.5um</td>
</tr>
<tr>
<td>8.3</td>
<td>Metal1 enclosure of Via &gt;= 1.25um</td>
</tr>
<tr>
<td>8.4</td>
<td>Via spacing to Contact &gt;= 2.5um</td>
</tr>
<tr>
<td>8.5</td>
<td>Via spacing to Poly1 &gt;= 2.5um</td>
</tr>
<tr>
<td>9.1</td>
<td>Metal2 width &gt;= 3.5um</td>
</tr>
<tr>
<td>9.2</td>
<td>Metal2 spacing and notch &gt;= 3.5um</td>
</tr>
<tr>
<td>9.3</td>
<td>Metal2 enclosure of Via &gt;= 1.25um</td>
</tr>
<tr>
<td>10.1</td>
<td>Exact passivation window size = 100um x 100um</td>
</tr>
</tbody>
</table>
Design Rule Checker

- **Programming** a rules set...

```python
...  
active = geomGetShapes("GASAD", "drawing")
polygate = geomGetShapes("POLY1", "drawing")
polycap = geomGetShapes("POLY0", "drawing")
gate = geomAnd(polygate, active)
cpoly = geomAnd(polygate, polycap)
geomOffGrid(polygate, 0.25, 1, "0.0. Design grid is ...
geomWidth(gate, 3, "4.1.a. Poly1 width inside GASAD >= ...
geomSpace(polygate, 3, diffnet, "4.2. Poly1 spacing...
geomNotch(polygate, 3, "4.2. Poly1 notch >= 3um")
geomExtension(polygate, active, 2.5, "4.4. Poly1 ext...
geomEnclose(polycap, cpoly, 3, "4.6. Poly0 enclosure...
...
```

e.g. 2.5um 2P2M CMOS (CNM25)
Design Rule Checker

- Using assisted DRC tools

- e.g. glade and CNM25

- Report on total DRC error count
1. Device Sizing

2. Process and Mismatching Simulation

3. The Art of Analog Layout

4. Physical Verification

5. Parasitics Extraction

6. DFM Techniques
Motivation

- **Planar** technology parasitics

(2D layout view)

\[ R_{\text{shunt}} \]

\[ 13.5\rho \]

\[ \rho[\Omega/\square] \]
Motivation

- **Planar** technology parasitics

(2D layout view)  
(2D cross section view)

\[ R_{shunt} \]

\[ 13.5\rho \]

\[ \rho [\Omega/\square] \]
Motivation

- **Planar** technology parasitics
- ▼ **R** and **L** require node **splitting**!

(2D layout view)  
(2D cross section view)  
(3D view)  

\[ R_{shunt} \]
\[ 13.5 \rho \]
\[ \rho [\Omega/\square] \]

\[ C_{fringe} \]
\[ C_{overlap} \]
\[ C_{sidewall} \]

\[ L_{shunt} \]
\[ K_{coup} \]
Extraction Tools

**Programming** a simple rules set for parasitic overlap capacitance only...

```python
... geomLabel(polygate, "POLY1", "pin", 1) geomLabel(polygate, "POLY1", "net", 0) geomConnect([
    [cont, ndiff, pdiff, polygate, polycap, metal1],
    [via12, metal1, metal2]... ] ) extractMOS("cnm25modn", ngate, polygate, ndiff, pwell) extractParasitic3(pdiff, metal2, cmetal2diff, 0,
    [metal1, polygate, polycap]) ...
```

e.g. 2.5um 2P2M CMOS (**CNM25**)
Extraction Tools

- **Programming** a simple rules set for parasitic overlap capacitance only...

- Using assisted extraction tools...

```
.SUBCKT opamp vinn vinp vout vdd vss ibias
MM0 vdd ibias vdd vdd cnm25modp w=1.2e-05 l=6e-06 as=-...
MM1 vdd ibias vout vdd cnm25modp w=1.2e-05 l=6e-06 as=-...
Cc0 vinter vout cnm25cpoly w=6.42928e-05 l=0.000156207
MM8 vout ibias vdd vdd cnm25modp w=1.2e-05 l=6e-06 as=-...
...
CP1 vinter vss C=3.8582e-13
CP2 vout ibias C=3.33692e-15
CP3 vinp vss C=1.85938e-15
CP4 vout vcomm C=2.0918e-15
... .ENDS
```

e.g. glade and CNM25
1 Device Sizing

2 Process and Mismatching Simulation

3 The Art of Analog Layout

4 Physical Verification

5 Parasitics Extraction

6 DFM Techniques
Going Into Production

- From few IC prototypes to small and medium series

- Design for manufacturing (DFM) layout rules to increase manufacturing yield

- Some examples:
  - Dummy filling
Going Into Production

- From few IC prototypes to small and medium series
- Design for manufacturing (DFM) layout rules to increase manufacturing yield
- Some examples:
  - Dummy filling

Using few shapes also simplifies optical proximity correction (OPC)
Going Into Production

- From few IC prototypes to small and medium series
- Design for manufacturing (DFM) layout rules to increase manufacturing yield
- Some examples:
  - Dummy filling
  - Antenna reduction

charge stored during metal patterning can break thin gate oxide:

\[
\frac{A_{\text{met}}}{A_{\text{gate}}} < \text{max}
\]

discharging path
Going Into Production

- From few IC prototypes to small and medium series

- Design for manufacturing (DFM) layout rules to increase manufacturing yield

- Some examples:
  - Dummy filling
  - Antenna reduction
  - Metal slotting
  - Multiple contacts
  - Extra guard rings

large metal area = mechanical stress issues

stress-relief holes