RAIL: Resilient Analog Instance Language/Library Enabled Process-Portable Mixed Signal Circuit

Documentation - I
RAIL Library

Chixiao Chen, Phil Huang, Ailing Piao, Helen Huang, C.-J. Richard Shi

May 10, 2019

NOTICE

Contents

Chapter 1

Introduction

1.1 Scope

This document defines the resilient analog instance library including features, functionalities, characteristics, and expressions. The purpose of this databook is to describe the basic set of electronic circuits and systems. The library is designed to provide a constraint-driven and technology-independent flow for the circuit designer. Cells in this library are designed for the metal routing resources; place and route tools can use the layers of upper metals. Designers can refer to this databook for cell availability, functionality, expression, and constraints.

Chapter 2

Logic Gates Data Sheet

2.1 Switches

2.1.1 CMOS Transmission Gate

The Transmission Gate is the most common switch used in AMS designs. After careful investigation into standard cells, we found that the unit CMOS transmission gate can be created by truncating the `MUX2` cell.

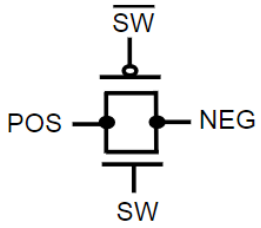

Figure 2.1: Transmission Gate.

- **Cell Name**
  TGAT
- **Pin Definition**
  Digital Input: `SW`
  Analog In/Out: `POS, NEG`
- **Analog Character**
  Maximum/minimum on resistance ($R_{on,max}$, $R_{on,min}$), off-resistance (R_{off})
  Parasitic capacitance at positive and negative nodes (C_{pos}, C_{neg})
- **Verilog-A Expression**
  ```
  analog I(POS,NEG) <+ SW ? (V(POS,NEG)/Ronmax):(V(POS,NEG)/Roff);
  ```
- **RAIL Expression**
  ```
  rail POS <+ SW ? NEG : hiZ @ Ron = 100;
  ```
- **Digital Fan-in**: 1x unit logic gate
- **Comments**
  The RAIL library composer generates a unit cell transgate, while the RAIL synthesizer will generate the number of unit cells that should be used according to the constraints.

2.1.2 NMOS/PMOS Single Transistor Switch

Single transistor switches are also very common in AMS designs and serve multiple purposes. For example, an NMOS switch can serve as the core transistor for the bootstrapped switch. Though they are quite straightforward, these switches are presented in their own category. The symbol of NMOS/PMOS single transistor switch is shown in Figure 2.2.

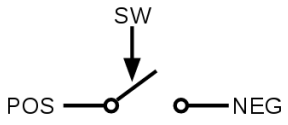

Figure 2.2: Single transistor switch.

- **Cell Name**
  PSW1, NSW1
- **Pin Definition**
  Digital Input: `SW`
  Analog In/Out: `POS, NEG`
- **Analog Character**
  Maximum/minimum on resistance ($R_{on,max}, R_{on,min}$), off-resistance (R_{off})
  Parasitic capacitance at positive and negative nodes (C_{pos}, C_{neg})
- **Verilog-A Expression**
  ```
  analog I(POS,NEG) <+ SW ? (V(POS,NEG)/Ronmax):(V(POS,NEG)/Roff);
  ```
- **RAIL Expression**
  ```
  rail POS <+ SW ? NEG : hiZ @ Ron = 100;
  ```
- **Digital Fan-in**: 1x unit logic gate
- **Comments**
  The RAIL library composer generates a unit cell single transistor switch, while the RAIL synthesizer will generate the number of unit cells that should be used according to the constraints.

2.1.3 Pull-up Analog MUX2

There are many cases where a node is either set to an analog signal or a logic high. We define these gates as pull-up analog MUX2. The logic high selection is normally used in reset/pre-charge cases. The analog signal selection is implemented by a typical transmission gate. Figure 2.3 shows the symbol for a pull-up analog MUX2.

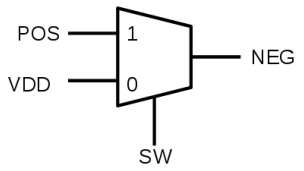

Figure 2.3: Pull-up analog MUX2.

- **Cell Name**
  MX2U
- **Pin Definition**
  Digital Input: `SW`
  Analog In/Out: `POS, VDD, NEG`
- **Analog Character**
  Maximum/minimum on resistance for the analog signal path ($R_{on,max}$, $R_{on,min}$),
  On-resistance for the logic high path ($R_{on,vdd}$),
  Off-resistance (R_{off}),
  Parasitic capacitance at analog input and output nodes (C_{in}, C_{out})
- **Verilog-A Expression**
```
analog I(AIN,NEG) <+ SW ?(V(POS,NEG)/Ronmax):V(POS,NEG)/Roff;
analog I(VDD,NEG) <+ (!SW)?(V(VDD,NEG)/Ronvdd):V()/Roff;
```
- **RAIL Expression**
```
rail NEG <+ SW ? POS : VDD @ Ron = 100;
```
- **Digital Fan-in**: 1x unit logic gate
- **Comments**
  The RAIL library composer generates a unit cell pull-up analog MUX2, while the RAIL synthesizer will generate the number of unit cells that should be used according to the constraints.

2.1.4 Pull-down Analog MUX2

There are many cases where a node is either set to an analog signal or a logic low/ground. We define these gates as pull-down analog MUX2. The logic low selection is normally used in reset/pre-charge cases. The analog signal selection is implemented by a typical transmission gate. Figure 2.4 shows the symbol for a pull-down analog MUX2.

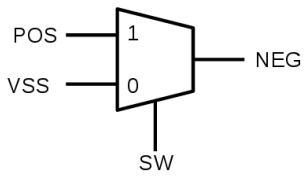

Figure 2.4: Pull-down analog MUX2.

- **Cell Name**
  MX2D
- **Pin Definition**
  Digital Input: `SW`
  Analog In/Out: `POS, VSS, NEG`
- **Analog Character**
  Maximum/minimum on resistance for the analog signal path ($R_{on,max}, R_{on,min}$),
  On-resistance for the logic low path ($R_{on,vdd}$),
  Off-resistance (R_{off}),
  Parasitic capacitance at analog input and output nodes (C_{in}, C_{out})
- **Verilog-A Expression**

```
analog I(POS,NEG) <+ SW ?(V(POS,NEG)/Ronmax):V(POS,NEG)/Roff;
analog I(VSS,NEG) <+ (!SW)?(V(VSS,NEG)/Ronvdd):V()/Roff;
```

- **RAIL Expression**

```
rail NEG <+ SW ? POS : VSS @ Ron = 100 ;
```

- **Digital Fan-in**: 1x unit logic gate
- **Comments**
  The RAIL library composer generates a unit cell transgate, while the RAIL synthesizer will generate the number of unit cells that should be used according to the constraints.

2.1.5 Tri-state Buffer

Tri-state buffers are quite useful for dynamic settling circuits, such as a switch driver for cap array. Its output is either logic high, logic low, or high impedance. The standard cell library provide a reference design, known as `BUFT`. However, sometimes this may not meet the requirements of analog design (see the next section for constraints).

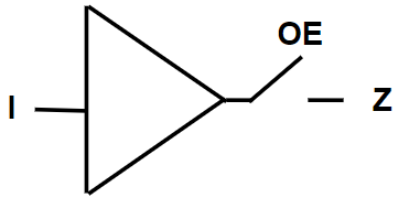

Figure 2.5: Tri-State Buffer.

- **Cell Name**
  TBUF
- **Pin Definition**
  Digital Input: `OE, I`
  Analog In/Out: `Z`
- **Analog Character**
  On-resistance for the logic high/low path (R_{on}), off-resistance (R_{off})
  (Note: for simplicity, both resistances are combined),
  Parasitic capacitance at output nodes (C_{load})
- **Verilog-A Expression**

```
analog I(VDD,Z) <+ (OE&I)?(V(VDD,Z)/Ron):V()/Roff;
analog I(VSS,Z) <+ (OE&!I)?(V(VSS,Z)/Ron):V()/Roff;
```

- **RAIL Expression**

```
rail Z <+ OE ? (I ? VDD : VSS) : hiZ @ Ron = 100;
```

- **Digital Fan-in**: 1x unit logic gate
- **Comments**
  The RAIL library composer generates a unit cell buffer-tri-state, while the RAIL synthesizer will generate the number of unit cells that should be used according to the constraints.

2.1.6 Switch for Driver

Typical tri-state buffers may not meet the requirements for charge domain applications. In other cases, it may be difficult to synchronize the enable signal and data input (especially in asynchronous design). The RAIL library's solution is uses a straightforward method that contains one PMOS and one NMOS. These switches serve as drivers for wireline transceivers. A current-mode push-pull driver can be created by connecting the switch with current sources. Alternatively, if voltage sources are connected to the switch, a voltage-mode driver is created. Figure 2.6 shows the symbol for the Switch for Driver.

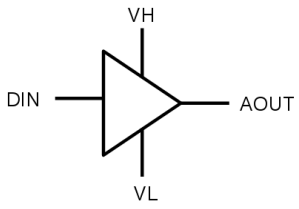

Figure 2.6: Switch for Driver.

- **Cell Name**
  SWFD
- **Pin Definition**
  Digital Input: `DIN`
  Analog In/Out: `VH, VL, AOUT`
- **Analog Character**
  On-resistance for the logic high/low path (R_{on}), off-resistance (R_{off})
  (Note: for simplicity, both resistances are combined),
  Parasitic capacitance at output nodes (C_{load})
- **Verilog-A Expression**
  ```
  analog I(DIN,AOUT) <+ (DIN)?(V(VH,AOUT)/Ron1):V(VL,AOUT)/Ron0;
  ```
- **RAIL Expression**
  ```
  rail AOUT <+ DIN ? VH :VL @ Ron = 100;
  ```
- **Digital Fan-in**: 1x unit logic gate
- **Comments**
  The RAIL library composer generates a unit cell switch, while the RAIL synthesizer generates the number of unit cells that should be used according to the constraints.

2.1.7 LDO Power switch with RO unit

When designing a digital LDO, it is necessary to have a power switch/transistor to control the regulated voltage. To this end, we modify the tri-state buffer to form a power switch. Note that the regulated voltage needs to be sensed or converted into digital codes. In this case, a ring-oscillator is desired. As a result, the extra NOR gate in the tri-state buffer is replaced with an output voltage supplied RO cell.

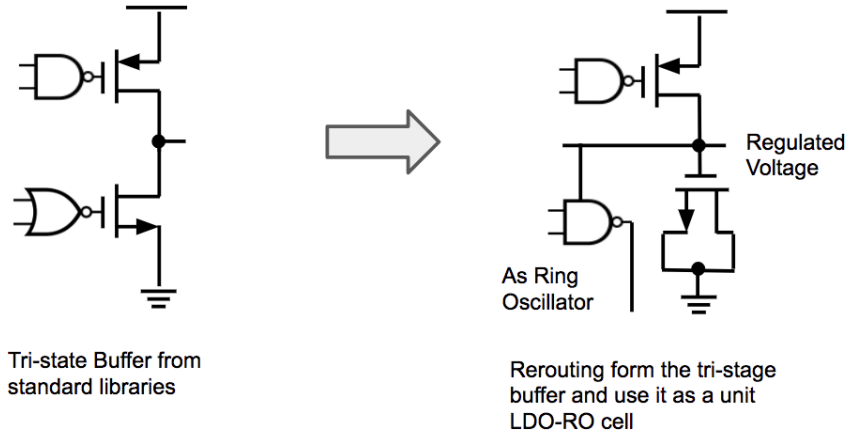

Figure 2.7: Change from the tri-state buffer topology to the LDO power switch

- **Cell Name**
  PXRO
- **Pin Definition**
  Digital Input: `EN, D, A, RST`
  Analog In/Out: `VO`
- **Analog Character**
  On-resistance when the dropout votlage is 50mV ($R_{on,50}$),
  load current (I_{load}),
  gate delay under certain power supply (t_d), have a built-in function with `VOUT`
  Parasitic capacitance at output nodes (C_{load})
- **Digital Fan-in**: 1x unit logic gate

2.2 Delay Sensitive Gates and Differential Pairs

2.2.1 CMOS Delay Cells

Normally, a digital standard cell library provides a variety of delay cells used for timing closure. These cells are also useful when synthesizing a digital controlled delay line (DCDL). When building a RAIL library, it is highly recommended to re-use these cells.

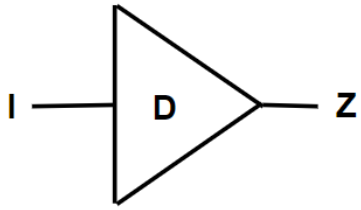

Figure 2.8: Delay Cell

- **Cell Name**
  DLY1
- **Pin Definition**
  Digital Input: `I`
  Digital Output: `Z`
- **Analog Character**
  Output: Rising Delay (t_{l2h}), falling Delay (t_{h2l})
  Input: Supply Voltage (V_{DD})
- **Verilog Expression**
  `rail assign Z = (r#(`t_{l2h}`) f#(`t_{h2l}`))I @ VDD=1;`
- **Electrical Coefficient**
  the first order and second order supply coefficient and bias ($\alpha_1,\ \alpha_2,\ \alpha_0$),
  the unit logic load delay (β),
  the rising and falling mismatch factor (γ).
- **Built-in function**

$$t_{l2h} = (\alpha_2 V_{DD}^2 + \alpha_1 V_{DD} + \alpha_0)\beta \tag{2.1}$$

$$t_{h2l} = \gamma\, t_{l2h} \tag{2.2}$$

- **Digital Fan-in**: 1x unit logic gate
- **Comments**
  Note that for the RAIL synthesizer, all the delay cells are load by unit logic gates (INVD1). The RAIL library composer normally picks up a DEL_01/02/04 of 3x/10x/20x unit buffer delay.

2.2.2 Voltage Controlled Discharge Cell

A significant portion of the digitizing AMS design is the conversion of traditional voltage domain analog signals into pulse width presented time-domain signals. Therefore, a voltage to time converter is important. Figure 2.9 shows a typical voltage controlled discharge cell. When `CLK` is low, `Vout` is precharged to power supply. When `CLK` rises, `Vout` is discharged via the voltage modulated transitor. The lower `Vin`, the slower `Vout` falls.

This circuit is widely used in Ring Oscillator designs and can serve as the dynamic preamplifier of latch-based comparators. In addition, with an extra discharge path (without preset transistor), it is transformed into a half circuit for a time-difference amplifier.

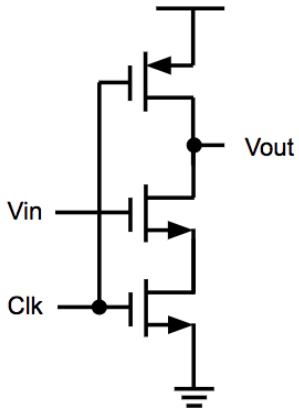

Figure 2.9: Voltage controlled discharge path.

- **Cell Name**
  VCDC, VCDP
- **Pin Definition**
  Digital Input: `CLK`
  Analog In/Out: `CKOUT, VIN`
- **Analog Character**
  Slope during discharging in the region of interest($\Delta V/\Delta t$),
  Coefficients between the slope and `Vin` ($\alpha_0, \alpha_1, \alpha_2$).
  The overall delay (normally including load, t_o) is computed using power supply
- **Built-in Function**

$$\frac{\Delta\, V}{\Delta\, t} = \alpha_0 + \frac{\alpha_2}{V_{in} + \alpha_1} \tag{2.3}$$

$$t_d = \frac{V_{DD}}{2\,\Delta\, V/\Delta\, t} + t_o \tag{2.4}$$

- **RAIL/Verilog Expression**
  `rail assign Vout = f#(`t_d`) !(CLK) @ t2v=20p;`
- **Digital Fan-in**: 1x unit logic gate

2.2.3 * Static Common-Source Amplifier Half Cell

The voltage controlled discharge cell has the same topology as typical common source amplifiers. (Recall the five-transistor amplifier common in analog textbooks.) Interestingly, this same circuitry can be used in simple static amplifier design if biased properly. The only difference is the additional node, `Vtail`, used as analog I/O. It is used for AC ground (tail point) / resistive degeneration connections.

Though the cell can work in traditional analog designs, we want to emphasize that the RAIL project's primary focus is digitized AMS design.

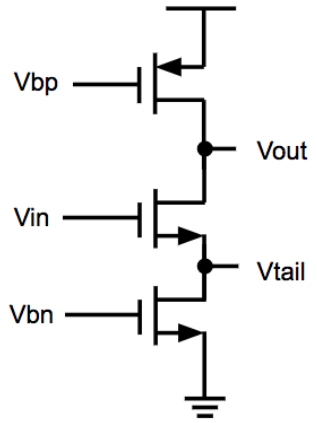

Figure 2.10: Common Source Amp.

- **Cell Name**
  RAIL_HCSAP
- **Pin Definition**
  Analog In/Out: `Vin, Vbp, Vbn, Vout, Vtail`
- **Analog Character**
  Static current (I_d),
  operation point at `Vout` (V_{ODC})
  Small Signal conductance (g_m),
  Small signal output impedance (r_o).
- **RAIL Expression**
  `rail Vout(pos,neg) <+- Vin(pos,neg) * A @ A=10;`
  (<+− means small signal, differential analog expression RAIL)
- **Built-in Function**

$$A = g_m(r_o//R_{\text{load}}) \tag{2.5}$$

- **Digital Fan-in**: 1x unit logic gate
- **Comments**
  In a future version, we may consider extending the amplifier part with a bandwidth or pole/zero optimize specification. These frequency characteristics are overlooked in the current version. Besides, we also suggest that the gate length could be extended (not minimum length) to better match if allowed.
- **Reference Testbench**
  Cell Name: `tb_rail_amp`, View Name: `sim_dc`

2.3 Custom Defined Block

2.3.1 Custom Class: Channel Connect Block

Most of the analog circuits can be divided into cascaded branches containing different numbers of pmos and nmos in series or parallel connections.

Let the numbers of pmos be x and the numbers of nmos be y, we define Px, Ny to represent the transistors in series connection and xP, yN to represent the transistoors in parallel connection.

As shown in Figure 2.11, P1N2 is a circuit with one pmos transistor and two nmos transistors in series, and P3N2 contains three pmos transistors in series and two nmos transistors in series. While P12N is a circuit with one pmos transistor and two nmos transistors in parallel.

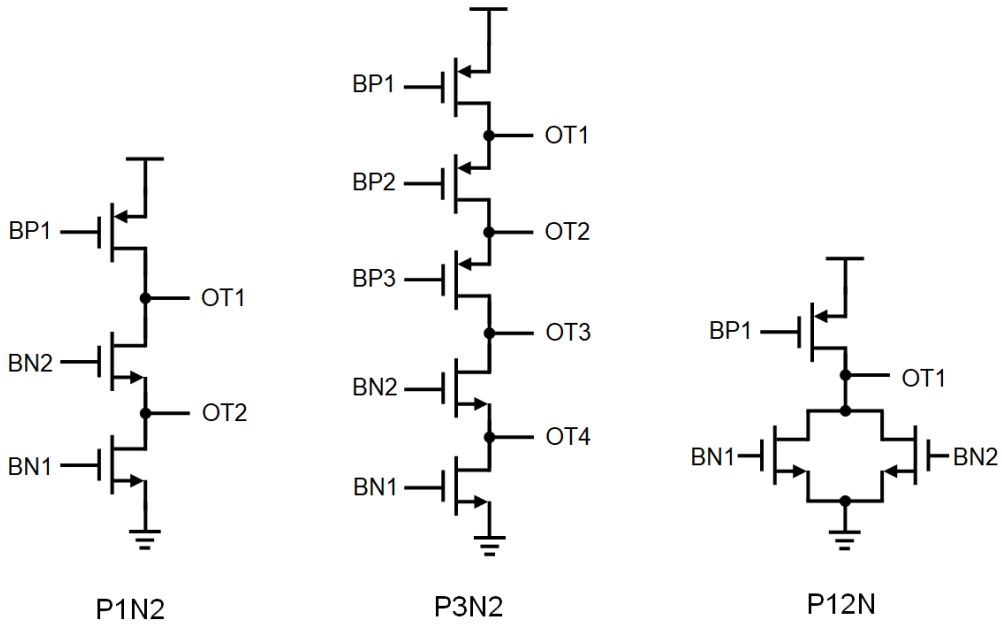

Figure 2.11: Channel Connect Block PxNy, x,y can be numbers from 1 to any number required.

- **Cell Name**
  PxNy, PxyN, xPNy, xPyN
- **Pin Definition**
  Digital/Analog Input: `BPx, BNy`, x and y can be numbers from 1 to any number required
  Analog Out: `OTn`, n can be a number from 1 to (x+y-1)
- **Commens:**
  In the future RAIL version users can define the numbers of transistors based on the needs to generate the required cells.

2.4 Miscellaneous

2.4.1 Custom Class: Bootstrapped Auxiliary Cell

In some cases, there may be circuit cells that cannot be presented by any existing topology in the common RAIL library. To address this, we have provided newly defined entries for developers. Figure 2.12, a bootstrapped sampling circuit, is shown here as an example.

- **Cell Name**
  CDC1
- **Pin Definition**
  Digital Input: `A2`
  Analog In/Out: `VO, CAP, A1`
- **Commens:**
  The cell might be replaced by a C^2MOS cell in the future RAIL version.

Note that for the custom-define-class (CDC), we do not provide any characters or testbench to analyze its electrical characters. But designers can force the number of CDC cells tracking any one defined cell.

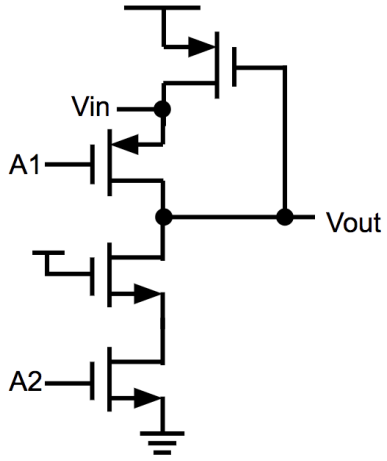

Figure 2.12: Auxiliary cell in boot-strapped switch.

Chapter 3

Passive Pcell Data Sheets

The RAIL project requires passive devices with robust programmability. Most passive devices have sufficient Pcell programmability in the Cadence/Virtuoso environment. However, three custom cells not supported in the PDK are defined in this section.

3.1 Configurable MOS Cap Load

In the case of comparator/delay-line calibration, a varactor-like mos cap load is desired. The key idea of this configuration is that biasing source/drain node at logic high or low results in different loading capacitance. The detailed circuit is shown in Figure 3.1.

- **Cell Name**
  PCAP, NCAP
- **Pin Definition**
  Digital input: `CT`
  Analog In/Out: `Vin`
- **Analog Character**
  Load capacitance when CT is high or low (C_{ld1}, C_{ld0})
- **Digital Fan-in**: 1x unit logic gate
- **Reference Testbench**
  Cell Name: `tb_moscap`
  View Name: `sim_ac`

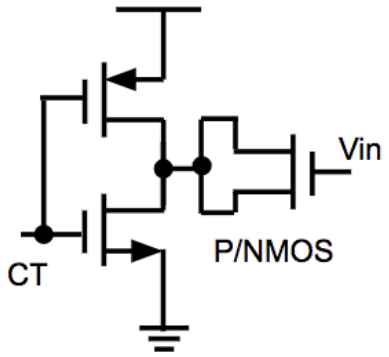

Figure 3.1: Configurable MOS cap.

3.2 Custom Unit MOM Capacitor I – Top/Bottom Imbalance

In SAR ADC designs, it is desirable to have an imbalanced unit capacitor. The SAR ADC can tolerate the large parasitic capacitance's in the bottom place (low impedance mode); however, the top plate (high impedance node) is more sensitive to this issue. To solve this problem, we provide a capacitor array with an asymmetrical 3D structure. One example is shown in 3.2. The top plate is surrounded by the all bottom, except the routing direction.

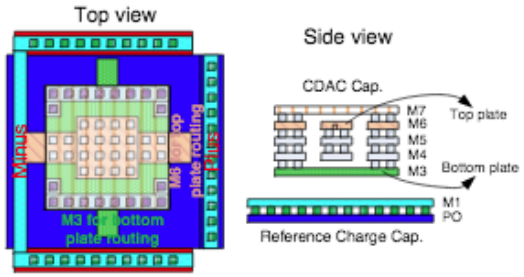

Figure 3.2: Custom unit cap - type I.

- **Cell Name**
  MOM1
- **Pin Definition**
  Analog In/Out: `POS, NEG`
- **Analog Character**
  The *parasitic* capacitance between nodes, (C_{unit}, C_{top}, C_{btm}), the last two characterize the capacitance between top/bottom place and the general ground.
- **Verilog-A Description**
  `analog I(pos,neg) <+` $C_{unit} \times ddt$`(V(pos,neg))`
- **Parametric Scalling factor**
  The unit cap could be set to within a range of (0.5, 2) fF, by changing the width of top plate. The poly used for decoupling is optional.

3.3 Custom Unit MOM Capacitor II – Top/Bottom Balance

Another type of unit capacitor is the fully symmetrical one. One example is illustrated in Figure 3.3. It is similar to the MOM capacitors provided by the PDK, only with less capacitance. These capacitors are more desired in capacitor bridges in sensors. Note that the routing for these cells should be done carefully.

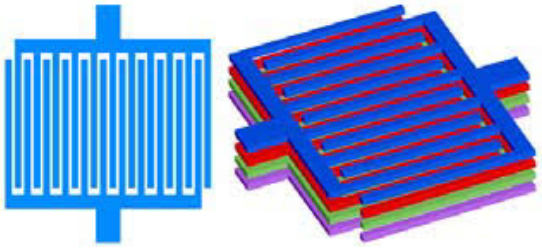

Figure 3.3: Custom unit cap - type II.

- **Cell Name**
  MOM2
- **Pin Definition**
  Analog In/Out: `POS, NEG`
- **Analog Character**
  The *parasitic* capacitance between nodes, (C_{unit}, C_{top}, C_{btm}), the last two characterize the capacitance between top/bottom place and the general ground.
- **Verilog-A Description**
  `analog I(pos,neg) <+` $C_{unit} \times ddt$`(V(pos,neg))`
- **Parametric Scaling factor**
  The unit cap could be set to within a range of (0.5, 2) fF, by changing the finger number of top/bottom pairs. The poly used for decoupling is optional.

For the application guide, the RAIL definition of the unit capacitor array is defined as
`rail POS <+ (-C-) NEG @ C=Cu*N, Cu=1f, N=128, Type=MOM_I ;`