Proposal for IP block development for the Efabless 2024 Chipalooza challenge

IP Block name:

Current bias

Designer/Design Team:

Christoph Maier

Email address: *christoph.maier@ieee.org*

01 March 2024

Circuit description

In its simplest form, the bias generator will just be matched current sources/sinks. To get good matching and good 1/f noise performance, the matched transistors will be long and narrow [https://bitbucket.org/cmucsd/analogdesignlecture/downloads/lecture.pdf].

As compliance is not specified,
I will play with cascoded or self-cascoded topologies [https://ieeexplore.ieee.org/document/6122459] to get good power supply rejection ratio.

As size is not specified,
I will play with size to meet noise and matching specifications.

Circuit pinout

Pin name	Use	Value
avdd	analog power	3.3. . . 5.5V
dvdd	digital power	1.8V typ.
avss	analog ground	0V
dvss	digital ground	0V
ena[15:0]	current branch enable	[dvss. . . dvdd]
vbg	bandgap voltage	???
iref[15:0]	reference output current	Variable 50, 100, 200, 400, 600nA ???
poweron	*digital, master power on*	*[dvss. . . dvdd]*

Note any changes from the specification, such as if trim bits have been added.

Circuit architecture

Basically, matched current mirrors with dummy structures for matching [https://bitbucket.org/cmucsd/analogdesignlecture/downloads/lecture.pdf].

If the current mirrors need to be switched individually,
I'll probably use cascode transistors to double as switches [https://github.com/tatzelbrumm/Zeugnisse/blob/master/PhD.pdf], p.86

Probably, I'll design *two* current bias blocks,

- matched (cascoded) PMOS current sources
- matched (cascoded) NMOS current sinks

irrespective of whether they're actually going to be needed to meet any corporate suit's device specification
(i.e., I just leave them out if the *official* specs don't require one of them).

To generate common bias voltages for the matched cascoded current sources and/or sinks, I'll put in a replica bias block
[https://www.semanticscholar.org/paper/Ultra-High-Input-Impedance%2C-Low-Noise-Integrated-Chi-Maier/ab83669efb5f29a94e19b8e2c3f4801ab50ba3ea/figure/11]

External resources (if any) (all resources must be open source)

List any resources that come from on chip, such as a bandgap-referenced voltage or PTAT current, or from other IP blocks (such as a comparator used in an ADC). List any external resources needed for testbench circuits, including any digital control sequencing needed.

I have chosen the current bias block because it requires a minimum of external resources from circuit building blocks that do not share matched layout primitives (i.e., identically sized and oriented MOSFETs).

Every *divide et impera* specification requires interfacing logic to re-establish matching by digitally trimmed adaptive circuits.
I will implement them as necessary.

A bandgap voltage would be more effort than a PTAT [? . . . I generally know what birds do, but not their name] input current.
A digitally trimmed current mirror is less expensive to build than a digitally trimmed voltage-to-current converter.

Maybe I will cheat and design yet another of my favorite Oguey&Aebischer bias blocks
[https://www.semanticscholar.org/paper/Ultra-High-Input-Impedance%2C-Low-Noise-Integrated-Chi-Maier/ab83669efb5f29a94e19b8e2c3f4801ab50ba3ea/figure/10],
[https://github.com/MastellaM/sky130_TAC3/pull/3]
with matching layout primitives.

In that case, all I need would be

- a digital input to switch off the building block
  (i.e., pull all the negative biases to avss and all the positive biases to avdd)

- digital inputs to enable current sinks and sources,
  which might require some dvdd/dvss to avdd/avss buffering (i.e., inverters and level shifters)

Specification difficulties

List all specifications which may be difficult to attain, and what circuit design methods will be used to meet those specifications. Note where specifications will be affected by layout considerations, such as mismatch, crosstalk, and I-R drop.

This building block is essentially an **exercise in matched layout**.

There may be a tradeoff between compliance voltage and output impedance — to cascode or not to cascode, that is the question.

All specifications **not based on matching layout primitives used across building blocks** are difficult to attain.

Testbenches required for verifying circuit performance

List what testbenches are used for each of the electrical parameters to be tested, and briefly describe the testbench circuit setup and how it measures the specified parameter.

Answer for Mr. Potbox

This: https://github.com/tatzelbrumm/sky130_TAC3/blob/main/playground/test_OgueyAebischerBias.sch, with transient, ac, and Monte Carlo mismatch simulations added to any current outputs there may be in addition to the power rails.

Suitable answer for suits

Simulation test bench with voltage sources at all connections (outputs are current outputs)

- Transient and dc sources to all digital inputs and outputs.
- DC simulations for PVT corner characterization. Monitor supply currents.
- DC sweeps at current output terminals to characterize compliance.
- Transient simulations of power startup and power transients. In addition, resistors built from idealized controlled sources to characterize stability of output currents.
- AC sources to all inputs, for quantifiable simulation results like PSRR and load sensitivity.

- AC sources to all loads, for characterization of sensitivity of the ciruict to AC injection.

Connections required for standalone (breakout) implementation

Indicate how the circuit is to be connected for individual testing outside of the eventual SoC application, and where test points may need to be added to access internal states of the circuit. Note where pad capacitance, wirebond inductance, and wire resistance from pad to circuit may affect measurement, and how to mitigate. Note where the circuit may need to be placed as close as possible to a pad.

Design something along the lines of a 74HC595 [https://www.diodes.com/assets/Datasheets/74HC595.pdf] with as many bits as needed to shift in the digital control bits from inputs

pin
SCLK
DATAIN
LATCH
ENABLE

Maybe add a digital serial DATAOUT pin to test the shift register,
and a digital RESET input to ensure a defined initial state.

As many analog current outputs as feasible.
Wire and pad capacitance would be a serious problem unless the external amperemeters are designed to actively keep the voltage constant
(I think such a bird is called *transimpedance amplifier*).
Keep the wires from the current sources/sinks to the analog pads short and wide, use minimal if any pad protective structures.

Dedicated power pins, if I want to be *completely* independent of any other stuff on chip.

Test plan for standalone (breakout) implementation

Describe how the standalone circuit can be measured on a lab bench to verify that the circuit meets performance requirements for each specified electrical parameter.

- Serial data generator for digital control,
- Ultra low input impedance micro- or nanoamperemeters, either as obscenely expensive laboratory instruments, or self-built.
- Provide the possibility to (capacitively? By transformer??) inject AC into the power lines and the current outputs to characterize PSRR, output resistance and such.

General idea of this circuit

My goal for designing this *(deceptively)* simple *(looking)* circuit is to overcome the monumental obstacle of configuring a working design flow.

The design philosophy should be matching layout driven, with no intermediate building block specifications between matching layout elements.
Hell is other people's APIs.

As time left after obtaining a stable design flow allows,
I might extend the *"current bias"* block to include one or more reference circuits with startup and brown-out detection, a current DAC, and maybe even other blocks, just as long as no interface specification disrupts matching of basic layout elements and bias voltages that adapt to process and temperature variations.

I'll consider all sorts of building blocks useful for an on-chip potbox [http://opencircuitdesign.com/~tim/research/potbox/potbox.ps],
but I won't promise I'll implement them [https://fossbytes.com/linus-torvaldss-famous-email-first-linux-announcement/].