Systematic Methodology of Modeling and Design Space Exploration for CMOS Image Sensors

Tianrui Ma, *Member, IEEE,* Zhe Gao, *Member, IEEE,* Zhe Chen, *Member, IEEE,* Ramakrishna Kakarala, *Senior Member, IEEE,* Charles (Qingwei) Shan, Weidong Cao, *Member, IEEE,* Xuan Zhang, *Senior Member, IEEE*

***Abstract*—CMOS Image Sensors (CIS) are integral to both human and computer vision tasks, necessitating continuous improvements in key performance metrics such as latency, power, and noise. Despite experienced designers being able to make informed design decisions, novice designers and system architects face challenges due to the complex and expansive design space of CIS. This paper introduces a systematic methodology that elucidates the trade-offs among CIS performance metrics and enables efficient design space exploration. Specifically, we propose a first-principle-based CIS modeling method. By exposing low-level circuit parameters, our modeling method explicitly reveals the impacts of design changes on high-level metrics. Based on the modeling method, we propose a design space exploration process that swiftly evaluates and identifies the optimal CIS design, capable of exploring over 10^9 designs in under a minute without the need for time-consuming SPICE simulations. Our approach is validated through a case study and comparisons with real-world designs, demonstrating its practical utility in guiding early-stage CIS design.**

***Index Terms*—CMOS image sensors, Integrated circuit modeling, Design space exploration.**

I. Introduction

CMOS Image Sensors (CIS) play a vital role in converting signals from external world to visual data for either human vision or computer vision tasks. Throughout years huge efforts have been made to improve CIS performance metrics [1], [2], among which three of them are the most important: latency, power, and noise. Other metrics, such as dynamic range, signal-to-noise ratio, and sensitivity, are also related to them. Experienced CIS designers can make fast and deliberate decisions to satisfy the design requirements, however, there lacks a systematic design methodology for novice designers and system architects who are not familiar with CIS details.

The fundamental reason behind this is CIS design involves a large and complicated design space. First, the design space spans over circuit level and architecture level. Circuit-level design parameters, such as transistor sizing, capacitance/resistance, bias current, and circuit topology, interactively affect architecture-level design parameters such as CIS floorplanning and CIS timing. Second, the design space crosses analog domain and digital domain. Not only is the whole CIS a mixed-signal case with analog processing circuits before Analog-to-Digital Conversion (ADC) and digital processing circuits after ADC, but there are also mixed-signal blocks (e.g., ADC and phase-locked loop circuit) where analog and digital signals are interleaved.

Critically, such a design space makes it implicit to investigate the trade-offs among CIS performance metrics. Without clear trade-offs, systematic performance optimization is more challenging. For example, a designer wants to reduce the CIS power by decreasing the bias current of the source follower in each pixel; however, the source follower might thus have high latency and the ADC needs to be faster to guarantee the frame rate requirement, which in turn increases the CIS power. Even though one can list all the design cases, validating each case in SPICE simulation is extremely time-consuming.

In this paper we propose a modeling method that systematically and explicitly investigates the CIS performance trade-offs in latency, power, and noise. Based on the modeling method, we further propose an efficient Design Space Exploration (DSE) process to obtain the optimal CIS design given a specified target. Since the modeling equations are derived from first principles and the SPICE simulations are not needed, the DSE process searches over 10^9 designs in less than a minute. The proposed modeling method and DSE process can be a useful tool to guide system designers towards their goals at CIS on early design stage.

Modeling Method. Our modeling method expresses CIS performance metrics with tunable circuit parameters – low-level circuit *knobs* – and first-principled equations, thus the systematic effect of changing a knob becomes explicit. Our modeling objects cover typical in-sensor circuits and generally apply to common CIS design cases. Specifically, we consider two common CIS architectures: CIS with column-level ADC and CIS with pixel-level ADC. At this point, we only support the modeling of Single-Slope (SS) ADC as this is the most commonly adopted ADC in commercial CIS products [3], because it has a good balance between speed, resolution, and power consumption and a relatively simple design which makes pixel-level integration easier.

DSE Process and Optimization. The DSE process is to sweep over the low-level circuit knobs exposed by the modeling equations, and evaluate the performance of the design under each set of knobs. The optimization is done by choosing the set of knobs that gives the best performance under the given optimization target. We use a case study to

This work was supported in part by NSF Awards #1942900 and #2416375.

T. Ma is with the Department of Electrical & Systems Engineering; Washington University in St. Louis, St. Louis, MO, 63130 USA, email: {tianrui.ma@wustl.edu}. Part of this work is done by T. Ma in his internship at OmniVision Technologies.

Z. Gao, Z. Chen, R. Kakarala, and C. Shan are with OmniVision Technologies, Santa Clara, CA, 95054 USA.

W. Cao is with the Department of Electrical & Computer Engineering; The George Washington University, Washington, DC, 20052 USA, email: {weidong.cao@gwu.edu}.

X. Zhang is with the College of Engineering; Northeastern University, Boston, MA, 02115 USA, e-mail: {xuan.zhang@northeastern.edu}.

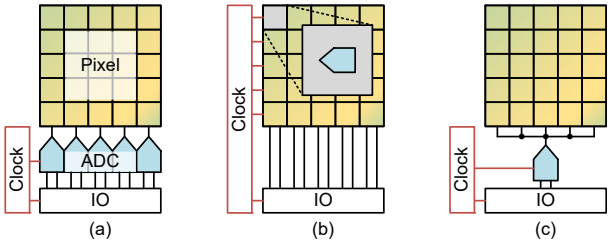


Fig. 1. Typical CIS architectures: (a) CIS with column-level ADC, (b) CIS with pixel-level ADC, and (c) CIS with chip-level ADC.

demonstrate the DSE process, and explain the rationale behind the performance trend. We discuss two optimization targets: power-optimized design and power-noise-product-optimized design.

Validation Study. To show the practicability of our method in real design cases, we compare the optimized design with the actual design from two chips – a scientific paper and a commercial product – and elaborate how the proposed methodology helps.

The remaining part of this paper is organized as follows: Sec. II introduces background and related works; Sec. III presents our modeling method for the CIS with column-level ADC; Sec. IV presents our modeling method for the CIS with pixel-level and chip-level ADC; Sec. V analyzes the DSE process inspired by the modeling method; Sec. VI demonstrates the validation with real designs; and Sec. VII is conclusion and future work.

II. Background and Related Works

In this section we first give a brief introduction to CIS, including basic operations and common architectures (II-A). Then we review the state-of-the-art modeling methods for CIS (II-B). Finally we review the classical modeling methods for analog/mixed-signal circuits, which many components in CIS belong to (II-C).

A. CMOS Image Sensors

CIS are the most popular visual sensing devices in industrial applications, due to their low cost, small footprint, and seamless integration with on-chip CMOS computation circuits [4]. Numerous CIS designs have been developed with novelties from in-sensor components to sensor architecture; nonetheless, they share the commonality in main components: 2D pixel array (with its addressing circuits), ADC, clock generator, digital processing circuits, and Input/Output (I/O) interface. The light signal is acquired by the pixel array, read out by the ADC, processed by the digital processing circuits, and finally sent off-chip by the I/O interface, under the clock sequence from the clock generator. Among these CIS designs, SS ADC is widely adopted, and its key circuits are a comparator, a counter, and a ramp signal generator.

CIS designs are usually categorized by the placement of ADCs in the sensor, as shown in Fig. 1, which include CIS with column-level ADC [5], CIS with pixel-level ADC [6], and CIS with chip-level ADC [7]. They represent different trade-offs among speed, power, and area chosen by the designers. The chip-level design places the only one ADC beside the pixel array which reads out the pixels sequentially; this ADC consumes the longest readout time which is proportional to the number of pixels, but its layout design is the least constrained thus complex circuits with flexible sizing can be used to gain better performance. The column-level design has column-parallel (or column-sharing) ADCs which read out the pixels in column-wise parallelism; this ADC consumes moderate readout time which is proportional to the number of pixel rows, but the width of its layout is limited by the pixel pitch. The pixel-level design places an ADC in every pixel to enable pixel-parallel readout; this ADC consumes the shortest readout time, but its layout has to be small enough for high pixel fill factor thus its performance is degraded.

B. CMOS Image Sensor Models

Researchers are finding ways to characterize and model the performance of CIS. As the era of visual embodied artificial intelligence comes where computations are integrated into CIS, such models become more urgent in assisting designers to evaluate the performance of the entire visual systems. However, previously proposed CIS modeling methods are either over-simplified or limited to single metric evaluation.

For example, Meta [8] proposes a power estimation framework for visual computation systems on AR/VR devices. They estimate the CIS power by coarsely dividing it into only four stages (sensing, readout, idle, and communication) with a constant power number for each stage; thus their model is unclear on how different CIS configurations affect the systematic power. CamJ [9] proposes an energy modeling framework for general computational CIS. It models the energy based on circuit structures and considers latency constraints; however, it emphasizes on the exploration of in-sensor computation architectures without considering the details of CIS-specific circuits. Gow et al. [10] present a MATLAB-based tool that simulates the impact of various noise sources on CIS and Fossum [11] explores the noise performance of quanta CIS using Poisson arrival statistics. Callens et al. [12] analyze the multi-metric trade-offs between speed, power, and noise under different readout architectures for 3D-stacked CIS; however, they limit the scope on the readout circuits only, and they do not propose a modeling method but instead characterizing the trade-offs with the data from published papers.

To our best knowledge, our proposed modeling method is the first of this kind to quantitatively and systematically discuss the latency-power-noise trade-offs in CIS and explicitly connect the trade-offs with low-level circuit details.

C. Analog/Mixed-Signal Circuit Models

At the circuit level, CIS is an Analog/Mixed-Signal (AMS) system. The most accurate AMS model is given by SPICE simulations; however, they require a huge amount of fundamental physics parameters and run slowly, especially as the size and complexity of the circuit scale up. Thus researchers are simplifying the AMS circuits with first-principle-based approximation models, e.g., the latency model [13], the power model [14], the noise model [15] as well as the model that emulates signal waves [16].

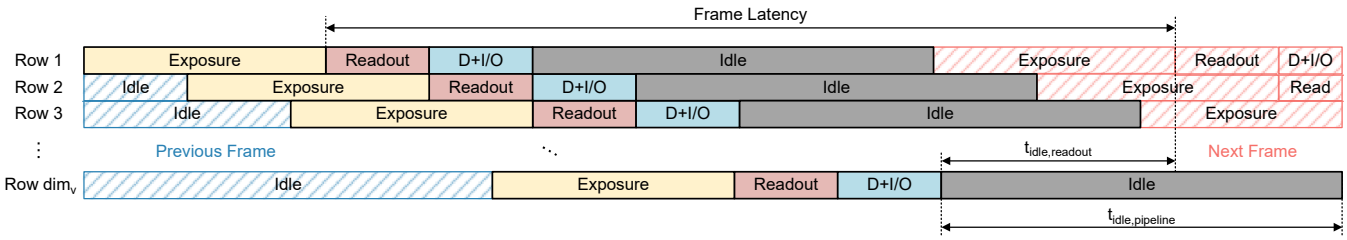


Fig. 2. Timing pipeline of CIS with column-level ADC, where "Readout" is pipelined with "D (digital processing) + I/O". dim_v is the vertical dimension of the pixel array – the number of pixels in one column. Note that the "Readout" and "D+I/O" stages can each have different durations.

These circuit models are combined to build up the model for complex AMS systems. For example, Lauwers et al. [17] propose a power modeling framework for ADCs and analog-continuous time filters without knowing circuit details. Their framework targets simpler AMS systems, and lacking circuit parameters in the model makes it less inspirational for designers to optimize circuits. Assche et al. [18] propose a modeling framework for ElectroEncephaloGram (EEG) sensors. Their system includes analog amplifier, ADC, digital logics/memory/processor, and transmission. With the consideration of speed and noise, their framework finds the power-optimal EEG sensor design; however, the latency and noise of the circuits are considered by high-level parameters, such as Gain-BandWidth-product (GBW) and Signal-to-Noise-Ratio (SNR), and the constraints among building blocks are not discussed. Our proposed modeling method is based on the similar first-principled paradigm but targets a more complex AMS system – the entire image sensor, with reasonable approximations to make the modeling parameters accessible to system designers.

III. Modeling CIS With Column-Level ADC

In Sec. III and Sec. IV we discuss our modeling method for CIS with different ADC placement. We start from discussing the CIS with column-level ADC (Sec. III), as this is the most common architecture for rolling shutter CIS and it well demonstrates our modeling principle. The modeling method for CIS with pixel-level ADC is similar to the column-level one with minor modifications and is discussed later (Sec. IV). The modeling method for CIS with chip-level ADC is also briefly mentioned in Appendix (Apdx. B).

A. Basic Assumptions

Operation Timing. CIS with column-level ADC typically adopts rolling shutter exposure [19]: the sensor converts and reads out light signal row by row. As shown in Fig. 2, there are five operation stages: exposure, readout (ADC), digital processing, I/O, and idle. The start of Row 1's exposure stage and the end of Row 1's idle stage mark the start and the end of the frame.

Note that the exposure stage of different rows can be overlapped, however, the readout stage cannot because there is only one row of ADCs shared by all rows of pixels. Within the frame latency, the ADCs are consecutively activated for each row and remain idle until the readout of Row 1 at the next frame.

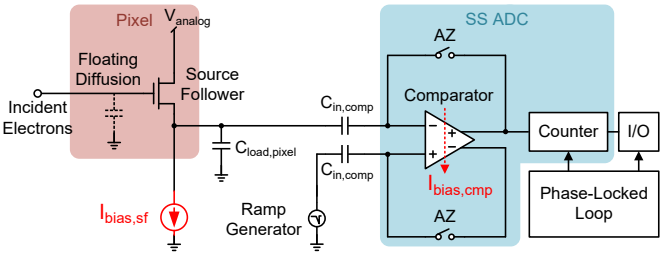


Fig. 3. Abstracted signal path of the modeled CIS, which is commonly adopted in previous works [20].

With the consecutive readout, the digital processing stage and the I/O stage are hidden behind the readout stage to reduce latency. There are two common ways of pipelining: (1) (Fig. 2) the digital processing and the I/O share the same stage and it is pipelined with the readout stage; in this way, the latency sum of the digital processing and the I/O cannot exceed the readout latency; and (2) the digital processing and the I/O take two stages and they are pipelined with the readout stage; in this way, neither the digital processing latency nor the I/O latency can exceed the readout latency (*Constraint-A*).

After each row is transmitted off-chip through the I/O, the row goes through an idle stage before starting exposure for the next frame. Given the frame latency and the readout latency, the exposure length is adjusted by tuning the idle latency $t_{idle,pipeline}$.

Signal Path. The signal path in our model starts from pixel and ends at I/O. The abstracted signal path is shown in Fig. 3: the input light is acquired as electrons on the in-pixel Floating Diffusion (FD) node; the electrons are read out through the in-pixel Source Follower (SF) as voltage; the voltage is received and converted to digital pixel value by the SS ADC, which consists of one comparator, two Auto-Zero (AZ) switches, and one counter, through two input capacitors and one ramp generator; the digital value is processed by an on-chip digital processor (if there is any) and finally sent off-chip through the I/O. Note that the ADC is column-parallel while a single ramp generator and a single I/O are shared by all ADCs. A phase-locked loop provides clocks for the counter in SS ADC and the driver in I/O.

Hardware Abstraction. Modeling the circuits with detailed physical parameters gives the most accurate results, but it is impractical and makes the model no difference to SPICE simulations. Thus, our model tries to find a good balance between accuracy and modeling efficiency – using the minimum parameters to achieve reasonable modeling accuracy. Our

model is indeed aware of transistor-level parameters, unlike the coarse function-level model in Meta's framework [8]; however, for each transistor we only ask for the basic parameters that define the large-signal current and small-signal conductances.

We further reduce the parameters by simplifying the current equations. Specifically we assume three inversion levels for each transistor: strong, moderate, and weak inversion. For the transistors in strong inversion, we use classical squared-law current equations [21]; for the transistors in weak inversion, we use subthreshold current equations [22]; and for the transistors in moderate inversion, we use g_m/I_d method [23] to directly estimate current from g_m (transconductance) without having any other physical parameters.

For the small-signal conductances we consider transconductance $g_m = \partial i_d / \partial v_{gs}$, output conductance (caused by channel-length modulation effect) $g_d = \partial i_d / \partial v_{ds}$, and body conductance (caused by body effect) $g_{mb} = \partial i_d / \partial v_{sb}$. Although the ratios between these conductances depend on process nodes, without loss of generality, we assume $g_m = 100 g_d = 10 g_{mb}$ for straightforward numerical calculations.

With these assumptions, we show our model in details for latency, power, and noise in Sec. III-B, Sec. III-C, and Sec. III-D, respectively.

B. Latency Model

In Fig. 2, the frame latency is given by the sum of exposure length, readout latency, digital processing and I/O latency, and idle latency. A clearer way to calculate the frame latency, due to the pipelining nature, is to mark the start and the end of the frame by the start of Row 1's readout stage at current frame and the start of Row 1's readout stage at next frame. Thus the frame latency t_{frame} is given by:

$$t_{frame} = \frac{1}{\text{FrameRate}} \approx \text{dim}_v \times t_{readout} + t_{idle,readout} \tag{1}$$

where $t_{readout}$ is the readout latency and $t_{idle,readout}$ is the idle latency between the readout of Row dim_v at current frame and the readout of Row 1 at next frame. Since usually dim_v is large, we ignore the digital processing and I/O latency for simplicity. $t_{idle,readout}$ can then be specified by the frame's duty cycle:

$$\text{DutyCycle} = \frac{\text{dim}_v \times t_{readout}}{t_{frame}} \times 100\% \tag{2}$$

Designers commonly allocate a specific portion of the frame time for readout, leaving remaining time for digital processing, communication, and timing synchronization. The duty cycle is the fraction of total frame time actively used for readout. Specifying a duty cycle is crucial as it influences power, noise, and overall system timing.

$t_{readout}$ contains a series of consecutive operations from pixel to ADC. Our latency model adopts a typical readout operation sequence [20]. It consists of three stages: the electron transfer through in-pixel SF (t_{sf}), the auto-zero of ADC's comparator (t_{az}), and the counting of ADC's counter (t_{count}). Modern CIS adopt Correlated Double Sampling (CDS) to suppress offset, flicker (1/f), and reset noise. To perform CDS, two pixel readouts are required: (1) *reset level*: sampled immediately after pixel reset, containing the reset voltage plus noise; and (2) *signal level*: sampled after exposure, containing the decreased reset voltage (because photocharge discharges the sense node) plus approximately the same noise. Subtracting the two samples cancels the common mode noise, generating an estimate of the true signal. Therefore, both electron transfer and ADC counting are performed twice per pixel. $t_{readout}$ is thus expanded as:

$$\begin{aligned} t_{readout} &= t_{sf} + t_{az} + t_{count} \\ &= \left(t_{sf,reset} + t_{sf,signal}\right) + t_{az} + \left(t_{count,reset} + t_{count,signal}\right) \end{aligned} \tag{3}$$

Eqn. (2) and Eqn. (3) reveal an important constraint: given a frame rate and a duty cycle, the sum of t_{sf}, t_{az}, and t_{count} is a constant (*Constraint-B1*). In other words, with a faster SF and a faster ADC's comparator, the ADC's counter must be slowed down to meet the constraint. Eqn. (3) contains the latency from both analog and digital circuits and we deal with them differently for accurate modeling.

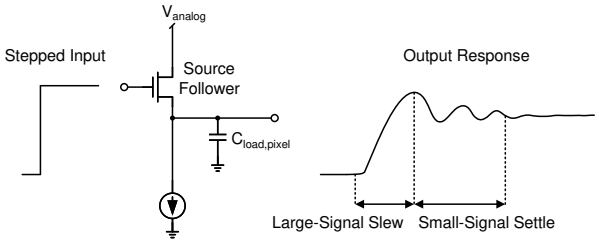


Fig. 4. Analog latency is expressed by large-signal slew latency and small-signal settling latency. The SF is used as an example.

Analog Latency. For the analog latency ($t_{sf,reset}$, $t_{sf,signal}$, and t_{az}), they are modeled as the sum of large-signal slew latency and small-signal settling latency. This is because these analog circuits are discrete-time systems where the inputs are sampled voltages and the outputs are the step response to the corresponding inputs, as shown in Fig. 4. Specifically:

$$\begin{aligned} t_{sf,reset} &= \frac{|V_{sf0} - V_{sf,reset}|}{\text{SlewRate}_{sf}} + k_{settle} \times \tau_{settle,sf} \\ t_{sf,signal} &= \frac{|V_{sf,reset} - V_{sf,signal}|}{\text{SlewRate}_{sf}} + k_{settle} \times \tau_{settle,sf} \\ t_{az} &= \frac{|V_{in,cmp0} - V_{out,cmp0}|}{\text{SlewRate}_{cmp}} + k_{settle} \times \tau_{settle,cmp} \end{aligned} \tag{4}$$

where subscript "sf0/comp0" represents the SF/comparator's initial voltage at the beginning of the operation; k_{settle} is a factor describing how stable the small-signal settling is – typically $k_{settle} = 5$ gives a stable settling; and τ is the small-signal time constant.

Based on large-signal models, SlewRate_{sf} and SlewRate_{cmp} are derived as:

$$\begin{aligned} \text{SlewRate}_{sf} &= \frac{I_{bias,sf}}{C_{load,pixel}} = \frac{I_{bias,sf}}{\text{dim}_v \times C_{pixel} + C_{in,cmp}} \\ \text{SlewRate}_{cmp} &= \frac{I_{bias,cmp}}{C_{in,cmp} + C_{load,cmp}} \end{aligned} \tag{5}$$

where I_{bias} is the static bias current; the SF drives a column of pixels with each pixel having an internal capacitance C_{pixel} and the input capacitance of the comparator $C_{in,cmp}$; and the comparator drives both $C_{in,cmp}$ and its output capacitance

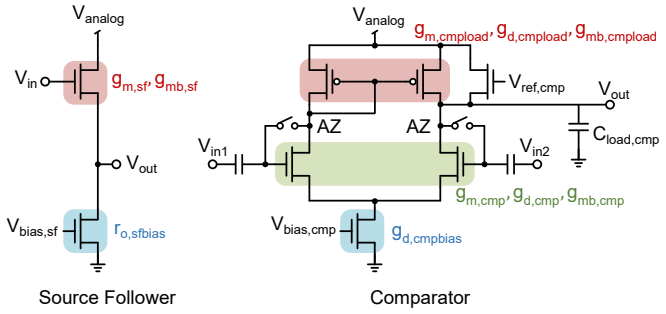


Fig. 5. The schematic of an SF and a comparator [24]. The color shadow indicates the small-signal conductances used in our model.

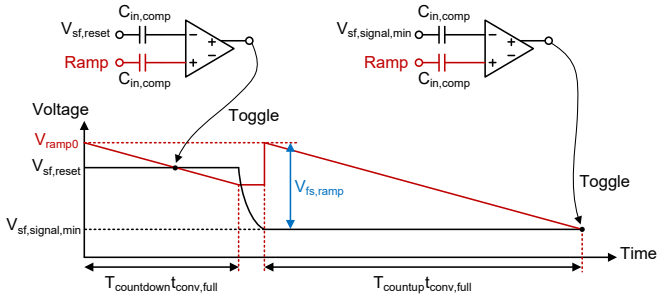


Fig. 6. Comparator behaviour during ADC counting. The ramp signal (red) decreases from V_{ramp0} for two pre-allocated counting periods that guarantees the intersections with both pixel readouts $V_{sf,reset}$ and $V_{sf,signal}$ (black). Since $V_{sf,reset} > V_{sf,signal}$, the counting period for $V_{sf,reset}$ (count down) is shorter than that for $V_{sf,signal}$ (count up). The ramp swing $V_{fs,ramp}$ is set by the minimum signal level $V_{sf,signal,min}$, corresponding to the longest exposure.

$C_{load,cmp}$ in AZ operation. Based on small-signal models, $\tau_{settle,sf}$ and $\tau_{settle,comp}$ are derived as:

$$\tau_{settle,sf} = \frac{1}{2\pi \times f_{p,sf}} = \frac{C_{load,pixel} + C_{gs,sf}}{g_{m,sf}} \approx \frac{C_{load,pixel}}{g_{m,sf}}$$
$$\tau_{settle,cmp} = \frac{1}{2\pi \times f_{p,cmp}} = \frac{r_{o,cmp} \times \left(C_{in,cmp} + C_{load,cmp}\right)}{1 + A_{dc,cmp}} \tag{6}$$

where f_p is the frequency at the circuit's principle pole; and $r_{o,cmp}$ and $A_{dc,cmp}$ are the comparator's output resistance and DC gain, respectively (see Apdx. C for derivation). The SF and the comparator used in our model adopt the structures in Fig. 5, in which the required small-signal conductances are marked in colors.

Digital Latency. For the digital latency ($t_{count,reset}$ and $t_{count,signal}$), they are modeled as the product of the number of clock cycles and the clock period. Theoretically, $t_{count,reset}$ and $t_{count,signal}$ represent the time from when the SS ADC begins counting (and the ramp signal at its input starts falling) to the moment the ramp intersects with the corresponding pixel readout. However, determining this exact intersection in real time complicates the digital control logic. Therefore, in line with common CIS design practice, these durations are pre-allocated to be sufficiently long to ensure the intersection occurs. Specifically, they are set as $T_{countdown}$ and $T_{countup}$ times the ADC's full-resolution conversion latency $t_{conv,full}$:

$$\begin{aligned} t_{count,reset} &= T_{countdown} \times t_{conv,full} \\ &= T_{countdown} \times 2^{reso_{adc}} \times t_{clk,count} \\ t_{count,signal} &= T_{countup} \times t_{conv,full} \\ &= T_{countup} \times 2^{reso_{adc}} \times t_{clk,count} \end{aligned} \tag{7}$$

where "countdown" means the counter counts towards smaller value during the ramp falling while "countup" means the opposite – by first counting downwards and then counting upwards, the pixel's reset level is subtracted from its signal level, and the CDS is realized; $reso_{adc}$ is ADC resolution; and $t_{clk,count}$ is counting clock period.

Since typically $T_{countdown} < T_{countup}$, the ADC's counting latency brings a contraint on the minimum slope of the ramp signal (*Constraint-B2*):

$$Slope_{ramp,min} = \frac{V_{fs,ramp}}{t_{count,signal}} = \frac{V_{ramp0} - V_{sf,signal,min}}{t_{count,signal}} \tag{8}$$

where V_{ramp0} is the voltage level that the ramp signal is reset to at the beginning of each counting. Any ramp signal with an absolute slope smaller than it would make the ramp signal never cross the pixel output V_{sf}, thus ADC would not performed properly. The expression of the ramp slope is related to the ramp generator's circuit schematic and we will discuss in Sec. III-C.

C. Power Model

The CIS power P_{frame} is expressed by the total energy consumption within the frame latency. As shown in Fig. 2, within the frame latency t_{frame} there is a period where dim_v rows are read out, processed, and sent off-chip sequentially, causing non-idle energy consumption; and there is also a period where the whole CIS is not working, causing idle energy consumption:

$$P_{frame} = \frac{E_{row} \times dim_v + E_{idle}}{t_{frame}} \tag{9}$$

The energy per row E_{row} comes from the operations at the readout, the digital processing, and the I/O stages:

$$\begin{aligned} E_{row} = P_{readout} \times t_{readout} + P_{digital} \times t_{digital} + P_{io} \times t_{io} \\ + P_{pll} \times t_{pll} + P_{periph} \times t_{readout} \end{aligned} \tag{10}$$

where specifically, the readout power $P_{readout}$ comes from the SF, the ADC, and the ramp generator; the digital processing power $P_{digital}$ comes from the dynamic switching in digital processing and the static leakage; the I/O power P_{io} comes from the physical layer module and the dynamic switching at capacitive loads; the clock generator (PLL) power P_{pll} comes from generating the VCO's clock, the ADC's counting clock, and the I/O clock; the peripheral power P_{periph} comes from the peripheral circuits whose components we do not explicitly consider. Thus:

$$\begin{aligned} P_{readout} &= (P_{sf} + P_{adc}) \times dim_h + P_{ramp} \\ P_{digital} &= P_{digital,dyn} + P_{digital,stat} \\ P_{io} &= P_{PHY} + P_{io,dyn} \\ P_{pll} &= P_{pll,vco} + P_{pll,adc} + P_{pll,io} \end{aligned} \tag{11}$$

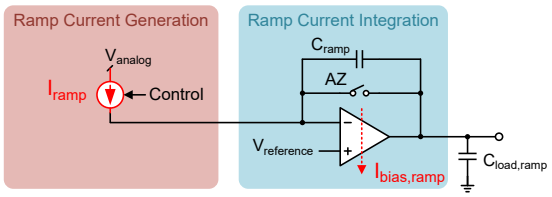


Fig. 7. The simplified block diagram of a continuous-time integrator-based ramp generator [25].

The idle energy E_{idle} comes from the leakage power $\tilde{P}$ of the circuits in each stage when the stage is not active:

$$\begin{aligned} E_{idle} = {} & \tilde{P}_{readout} \times (t_{frame} - dim_v \times t_{readout}) \\ & + P_{digital,stat} \times \left(t_{frame} - dim_v \times t_{digital}\right) \\ & + \tilde{P}_{io} \times (t_{frame} - dim_v \times t_{io}) \\ & + \left(\tilde{P}_{pll} + \tilde{P}_{periph}\right) \times t_{idle,readout} \end{aligned} \tag{12}$$

We assume the circuits expressed in $P_{readout}$, $P_{digital}$, and P_{io} are immediately idle and consume leakage power when they are not active, while the PLL circuits and the peripheral circuits are not idle until the end of last row in the frame.

Modeling Principle. Before delving into the details of the power model, we reiterate our modeling principle. Eqn. (11) shows that, similar to the latency model, the power components also contain both analog and digital parts. For analog power, we derive the bias current (I_{bias}) based on first principles and multiply it with the analog supply voltage; while for digital power, we either directly define the averaged power, or calculate the energy from the product of the energy per operation and the operation counts, as conventional digital simulators do [9]. Additionally, for the analog circuits with bias current, we consider their leakage current ($\tilde{I}_{bias}$) during inactive periods for leakage power estimation. In the following power model details, for simplicity we only highlight the bias current equations for each component and leave the power equations to Apdx. C.

Source Follower. The SF's power comes from its bias current $I_{bias,sf}$. We use squared-law current equation to determine $I_{bias,sf}$, because the SF is typically in strong inversion:

$$I_{bias,sf} = \frac{1}{2} kp_n \left(\frac{W}{L}\right)_{sf} V_{ov,sf}^2, \quad g_{m,sf} = kp_n \left(\frac{W}{L}\right)_{sf} V_{ov,sf} \tag{13}$$

where kp is the transistor's process parameter (μC_{ox}), (W/L) is the transistor's aspect ratio, and $V_{ov,sf}$ is the SF's overdrive voltage. To guarantee the squared-law current equation, $V_{ov,sf}$ is constrained by the classical saturation region condition (*Constraint-C1*) [21].

ADC. ADC's power comes from the energy consumption of the comparator and the counter within the readout time. For the comparator, it only needs to be turned on during the AZ period and the reset/signal quantization period:

$$E_{cmp} = V_{analog} \times I_{bias,cmp} \times \left(t_{az} + t_{reset,quant} + t_{signal,quant}\right) \tag{14}$$

Different to $I_{bias,sf}$, we use $\frac{g_m}{I_d}$ method [23] to quickly determine $I_{bias,cmp}$, otherwise the squared-law equations would be

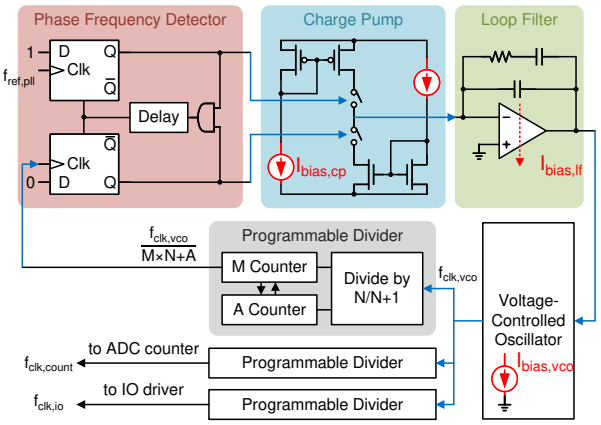


Fig. 8. The simplified block diagram of a PLL [26].

complicated for the comparator. Observing Fig. 5, $I_{bias,cmp}$ is:

$$I_{bias,cmp} = \frac{g_{m,cmpbias}}{\left(\frac{g_m}{I_d}\right)_{cmpbias}} = 2\frac{g_{m,cmpload}}{\left(\frac{g_m}{I_d}\right)_{cmpload}} = 2\frac{g_{m,cmp}}{\left(\frac{g_m}{I_d}\right)_{cmp}} \tag{15}$$

By assuming a constant $\frac{g_m}{I_d}$ ratio for each transistor, we connect the bias current to each transistor's transconductance.

For the counter, it consumes dynamic energy by toggling the digit:

$$E_{counter} = C_{counter} \times V_{digital}^2 \times NumFlip_{adc} \tag{16}$$

where $C_{counter}$ is the effective capacitance of the counter and $NumFlip_{adc}$ is the number of toggled digit in the counter. Different counter circuit results in different $C_{counter}$ and different coding scheme results in different $NumFlip_{adc}$.

Ramp Generator. There are two common types of ramp generators: continuous-time integrator-based ramp generator [25] (CT-Ramp) and discrete-time current-steering DAC-based ramp generator [27] (DT-Ramp). Both types have the similar power model and we only discuss the continuous-time type here for brevity. As shown in Fig. 7, the power comes from its static currents: I_{ramp} is the internal current of the CT-Ramp to achieve desired ramp slope; and $I_{bias,ramp}$ represents the bias current of the amplifier which integrates I_{ramp} to voltage output.

The ramp slope is thus $Slope_{ramp} = \frac{I_{ramp}}{C_{ramp}}$. Since $I_{bias,ramp}$ drives both the integrator's feedback capacitor C_{ramp} and the input capacitors of all comparators $C_{load,ramp}$, it should be large enough to guarantee that the maximum discharging rate on $C_{load,ramp}$ is faster than the ramp slope (*Constraint-C2*):

$$\frac{I_{bias,ramp}}{C_{ramp} + C_{load,ramp}} \geq Slope_{ramp} \geq Slope_{ramp,min} \tag{17}$$

Formula (17) and Eqn. (8) consider the constraint on the ramp generator from both systematic power and systematic latency perspectives.

Phase-Locked Loop. We consider a typical PLL circuit in Fig. 8. It includes a Phase Frequency Detector (PFD), a Charge Pump (CP), a Loop Filter (LF), a Voltage-Controlled Oscillator (VCO), and three programmable dividers to generate three independent clocks on-chip: PLL's main clock ($f_{clk,pll}$), ADC's counting clock ($f_{clk,count}$), and I/O clock ($f_{clk,io}$). Each

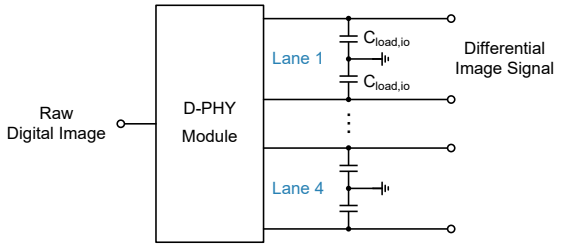


Fig. 9. The simplified block diagram of an I/O interface.

programmable divider includes a pre-divider (with ratio N) and two counters (with ratios M and A) and achieves a total dividing ratio of $M \times N + A$. These ratios need to be positive integers (*Constraint-C3*). The total power of the PLL is thus given by:

$$P_{pll} = P_{pfd} + P_{cp} + P_{lf} + P_{vco} + \Sigma_{n=1}^{3} P_{divider,n} \tag{18}$$

Among them, P_{pfd} and $P_{divider}$ are dynamic digital power while P_{cp}, P_{lf}, and P_{vco} are analog power.

The bias current of the charge pump ($I_{bias,cp}$) and the loop filter ($I_{bias,lf}$) has to be large enough to drive their respective internal capacitive loads within the VCO's clock period (*Constraint-C4*):

$$\begin{aligned} I_{bias,cp} &= C_{load,cp} \times V_{analog} \times f_{clk,vco} \\ I_{bias,lf} &= C_{load,lf} \times V_{analog} \times f_{clk,vco} \end{aligned} \tag{19}$$

I/O Interface. The I/O modulates and transmits the raw digital image (ADC codes) to the external endpoints under MIPI CSI-2 protocol. The image transmission needs to go through the physical layer (PHY), and MIPI alliance has developed dedicated PHY standards for the communication between the CIS and the external endpoints [28]. Among these standards, D-PHY and C-PHY are widely used.

As shown in Fig. 9, we simplify the I/O as a D-PHY module with differential output capacitive loads, since D-PHY uses differential signaling. Thus the power of I/O is given by:

$$\begin{aligned} P_{io} &= P_{PHY} + P_{io,dyn} \\ &= P_{PHY} + Num_{lane} \times \left(2C_{load,io} \times V_{io}^2 \times f_{clk,io}\right) \end{aligned} \tag{20}$$

where P_{PHY} is the power of D-PHY module, $C_{load,io}$ is the load capacitance, V_{io} is the range of the output signal, and Num_{lane} is the number of lanes in the I/O and a typical number is 4.

D-PHY has two modes [29]: low-power mode and high-speed mode. Our model automatically chooses the mode according to the I/O speed $f_{clk,io}$, and further chooses the corresponding P_{PHY} and V_{io} (*Constraint-C5*): when $f_{clk,io} \leq$ 10MHz, the I/O is at the low-power mode with V_{io} in [0, 1.2 V]; otherwise, the I/O is at the high-speed mode with V_{io} in [0.1 V, 0.3 V].

Peripherals. P_{periph} represents the power of all components that drain currents during the readout period $t_{readout}$ and that are not explicitly considered in our model. These components may include additional analog amplifiers that enhance signal in low light, auxiliary interfaces, or integrated IP cores (either analog or digital). Their currents are lumped into one term I_{periph}, which is independently pre-defined by designers and does not participate in systematic trade-offs. Designers can set a relatively large I_{periph} to provide a design margin for on-chip current management.

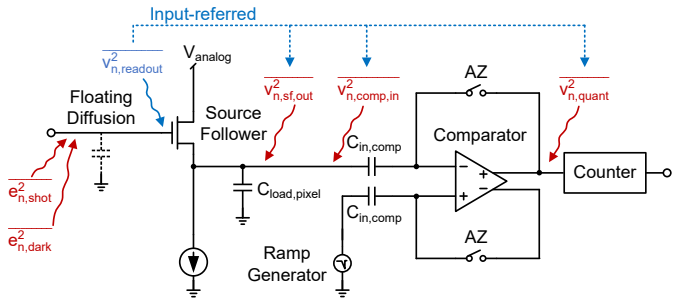


Fig. 10. Noise sources in the signal path. The read noise $\overline{v^2_{n,readout}}$ is an input-referred noise from the SF's gate terminal, including the noise from SF, comparator, and quantization. The subscript "n" means noise.

Digital Processing. Simple digital image preprocessing, including noise cancellation, demosaicking, and white balancing, is performed in the CIS. Since there exist mature digital modeling methods, we do not model the digital part in details but instead ask designers to directly provide its dynamic power $P_{digital,dyn}$ and static power $P_{digital,stat}$.

D. Noise Model

Our noise model only considers shot noise ($\overline{e_{n,shot}}$), dark current noise ($\overline{e_{n,dark}}$), and thermal noise in the readout signal path (or read noise, $\overline{e_{n,read}}$), as these are the main noise sources. For other noise sources, random telegraph noise has minimal effect on the CIS overall noise performance [30], and 1/f noise and reset noise are largely cancelled by CDS and AZ operations [31]. As shown in Fig. 10, the total input-referred noise is:

$$\overline{e_{n,total}} = \sqrt{\overline{e^2_{n,shot}} + \overline{e^2_{n,dark}} + \overline{e^2_{n,read}}} \tag{21}$$

where $\overline{e^2_{n,shot}} = e_{sig}$ as the shot noise follows Poisson distribution [31].

To specify the read noise, a signal path from FD to ADC is set up and the noise and the gain for each component are derived in Fig. 10. At the FD, the input-referred read noise is converted from electrons to voltages:

$$\sqrt{\overline{v^2_{n,read}}} = \overline{e_{n,read}} \times CG \tag{22}$$

where CG is conversion gain.

$\overline{v^2_{n,read}}$ is further expanded as:

$$\overline{v^2_{n,read}} = \frac{2}{A_{sf}^2}\left(\overline{v^2_{n,sf,out}} + \overline{v^2_{n,cmp,in}} + \overline{v^2_{n,quant}}\right) \tag{23}$$

where A_{sf} is the SF's gain (see Apdx. C for derivation), $\overline{v^2_{n,sf,out}}$ is the SF's output-referred noise, $\overline{v^2_{n,cmp,in}}$ is the comparator's input-referred noise, and $\overline{v^2_{n,quant}}$ is the ADC's quantization noise. Note that the read noise is multiplied with 2 to model the increased uncertainty caused by double sampling.

The SF's output-referred noise is derived from its small-signal model:

$$\overline{v^2_{n,sf,out}} = \gamma \frac{A_{sf}^2\left(\frac{1}{g_{m,sf}} + \frac{g_{m,sfbias}}{g^2_{m,sf}}\right)}{1/g_{m,sf} \| 1/g_{mb,sf} \| r_{o,sf} \| r_{o,sfbias}} \times \frac{kT}{C_{load,pixel}} \tag{24}$$

where k is Boltzmann constant, T is temperature, and γ is noise factor.

The comparator's input-referred noise is derived in a previous work [24]:

$$\overline{v_{n,cmp,in}^2} = 8\sqrt{2}kT\gamma n_e \sqrt{\frac{Slope_{ramp,min}}{2g_{m,cmp}V_{ref,cmp}C_{load,cmp}}} \tag{25}$$

where n_e is the effective number of transistors, and $V_{ref,cmp}$ is the reference voltage of the comparator.

The ADC's quantization noise is:

$$\overline{v_{n,quant}^2} = \frac{V_{lsb}^2}{12} = \frac{1}{12}\left(\frac{V_{fs,ramp}}{2^{reso_{adc}}-1}\right)^2 \tag{26}$$

Note that these noise numbers are integrated noise power (unit: V^2) rather than noise power density (unit: V^2/Hz). With the noise model, system-level parameters such as dynamic range (DR) and input signal-to-noise ratio (SNR_{in}) can be expressed: $DR = 20\log_{10}\frac{FWC}{\sqrt{e_{n,dark}^2+e_{n,readout}^2}}$, and $SNR_{in} = 20\log_{10}\frac{e_{sig}}{\sqrt{e_{n,shot}^2+e_{n,dark}^2+e_{n,readout}^2}}$, where FWC is full-well capacity.

E. Model Discussion and Summary

The full parameter list required by the proposed modeling method can be found in Apdx. A. All the other parameters are derived from these parameters.

Extensibility. Although this paper focuses on the SS ADCs, the similar modeling equations can be extended to the shared components (e.g., comparators and bias-current-driven analog blocks) in other ADCs such as Successive Approximation Register (SAR) and Delta-Sigma. And for those architecture-specific components in other ADCs, they can still be modeled using the same first-principled analysis method. Moreover, the proposed method can accommodate more complex shutter schemes beyond standard rolling or global shutter by adjusting the timing diagrams and constraints.

Corner Cases. We focus on a first-principled modeling approach rather than a full SPICE-level simulation, so precise Process-Voltage-Temperature (PVT) variation analysis lies beyond our scope. However, designers can estimate corner cases by extracting key transistor-level parameters our models need (e.g., g_m, g_d, V_{th}) under a few extreme conditions (e.g., slow-slow at –40°C, –10% supply vs. fast-fast at +120°C, +10% supply) and applying our algorithm iteratively. Although this omits some second-order effects, it remains a practical solution for early design exploration.

Observing Fig. 3, we find that the PLL does not directly affect the signal but serves only as a clock provider. Additionally, examining the modeling equations from Sec. III-B to Sec. III-D, we see that the digital processing and I/O are highly modularized and do not impact systematic latency and noise. Consequently, different optimization targets are considered for the on-chip components in Sec. V. We combine the SF, ADC's comparator, ADC's counter, ramp generator, and peripherals to investigate the latency-power-noise trade-off, while the PLL, digital processing, and I/O are grouped to focus solely on power minimization.

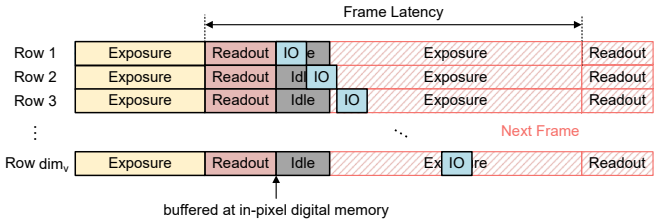


Fig. 11. Timing pipeline of CIS with pixel-level ADC. The digital processing stage is neglected. The stage length is not scaled.

IV. Modeling CIS with Pixel-Level ADC

To avoid redundant description, in this section we only highlight key differences in the models between the CIS with column-level ADC and the CIS with pixel-level ADC.

Operation Timing. CIS with pixel-level ADC typically adopts global shutter exposure to make the best use of the pixel parallelism. Fig. 11 shows its timing diagram, where the digital processing stage is neglected for simplicity. The exposure and the readout of all pixel rows happen concurrently, and the digital pixel values are transmitted through I/O row by row since the only I/O is shared by all the pixels. During the I/O transmission, after a period of idle stage, the exposure of the next frame starts; this exposure does not affect the digital pixels that have not been transmitted out, because they are temporarily buffered in the per-pixel digital memory, which is a typical feature in CIS with pixel-level ADC.

The latency of the I/O stage is constrained by the readout latency and the frame rate (*Constraint-A*):

$$dim_v \times t_{io} \leq t_{frame} - t_{readout} \tag{27}$$

Note that there can also be two ways of pipelining as the CIS with column-level ADC if the digital processing stage is considered.

Analog Loading. The ramp generator now drives $dim_v \times dim_h$ comparators rather than dim_h comparators in the CIS with column-level ADC.

Digital Memory. Since the pixel array is exposed and quantized simultaneously but sent off-chip row by row, in-pixel digital memory that buffers the ADC outputs is needed. Tools such as CACTI [32] can be used to estimate dynamic power and leakage power of the digital memory.

Transistor Sizing. Due to limited area, the in-pixel comparator adopts smaller sizing and thus works in the sub-threshold region. Rather than squared-law equations and g_m/I_d method, we use sub-threshold current equations to estimate currents and transconductances [22].

V. Optimization with Design Space Exploration

In this section, we show how the proposed modeling method facilitates CIS design optimization by efficient DSE. First, we discuss the optimization methodology (V-A). Then, we elaborate the DSE process step by step (V-B). Finally, we use a toy example to illustrate the DSE process and analyze the optimization results (V-C).

A. Optimization Methodology

Optimization Scenario. Our optimization solves the following design scenario: under the given CIS architecture (i.e., the placement of ADCs) and the peripherals static current I_{periph}, how the designers should *configure* the image sensor to make its performance optimal. The configuration theoretically can be changing any parameters in the modeling equations in Sec. III and Sec. IV, however, we determine four knobs which can be conveniently tuned by the designers. They include one systematic knob – sensor duty cycle, and three low-level circuit knobs – SF bias current $I_{bias,sf}$, comparator's bias current $I_{bias,cmp}$, and I/O's clock frequency $f_{clk,io}$. We choose the set of knobs that gives the optimal systematic CIS performance by sweeping each knob in its own practical range.

Optimization Targets. Two optimization targets are considered: to minimize CIS power (`Pow-opt`) and to minimize CIS power-noise-product (`PowNoi-opt`). We consider two Figures-of-Merit (FoMs) for the evaluation of the two targets [33], [34]:

$$\mathrm{FoM}_1 = \frac{P_{frame}}{\mathrm{FrameRate} \times \mathrm{dim_h} \times \mathrm{dim_v}} \times 10^{12} \quad [\mathrm{pJ/pixel}]$$
$$\mathrm{FoM}_2 = \frac{P_{frame} \times \overline{e_{n,read}}}{\mathrm{FrameRate} \times \mathrm{dim_h} \times \mathrm{dim_v}} \times 10^{12} \quad [\mathrm{pJ/pixel} \cdot e^-] \tag{28}$$

Note that we do not intend to minimize the CIS latency (i.e., maximize the frame rate) because the frame rate is a constant constraint pre-defined by the designers.

B. Two-Step Design Space Exploration

As mentioned in Sec. III-E, we adopt a two-step DSE approach to deal with mixed trade-offs. For those components in the latency-power-noise trade-off, we perform the first DSE step where the optimization target could be either `Pow-opt` or `PowNoi-opt`; while for the remaining components, we carry out the second DSE step with the optimization target fixed to `Pow-opt`. The sensor's total power is then the sum of the optimal power values obtained from both DSE steps.

Algo. 1 shows the pseudo-code of our DSE algorithm. Designers are required to provide the CIS architecture, the optimization target (for the first DSE), the searching range of each knob, and the modeling parameters other than the knobs. The algorithm then outputs the optimized component-wise latency/power/noise breakdown and FoMs with corresponding knobs under the given settings.

We explain the algorithm step by step. First, we define the CIS timing diagram with *Constraint-A* and assign the modeling parameters to each component.

Second, we perform the first DSE: for each combination of $I_{bias,sf}$ and $I_{bias,cmp}$, we compute the latency of SF and comparator with *Constraint-C1*; we compute the ADC's counting latency with *Constraint-B1* and *Constraint-C3*; we compute the ramp generator's bias current with *Constraint-B2* and *Constraint-C2*; and we compute the noise, the power, and the leakage power of each component. With all the power components, we compute the sum of them ($power_1$), and plot both $power_1$ and the product of $power_1$ and the read noise $\overline{e_{n,read}}$. Each plot is a 3D surface with $I_{bias,sf}$ and $I_{bias,cmp}$ being the independent variables. We search for the minimum of the 3D surface ($power_{1,opt}$) and find the corresponding $\{I_{bias,sf}, I_{bias,cmp}\}$ combination.

Third, we perform the second DSE: with each $f_{clk,io}$ from its range and the $f_{clk,count}$ obtained from the first DSE, we compute the bias current and the power of PLL with *Constraint-C3* and *Constraint-C4*; we compute the power of

Algorithm 1: Two-Step DSE

Input:
CIS architecture (FrameRate; ADC placement; digital processing pipeline type; I_{periph}),
Optimization target ('Pow-opt', 'PowNoi-opt'),
Knobs Sweeping range (DutyCycle, $I_{bias,sf}$, $I_{bias,cmp}$, $f_{clk,io}$),
Other modeling parameters (Apdx. A).
Output:
Component-wise latency/power/noise breakdown, FoMs.

1 **for** DutyCycle *in range* **do**
2 Define CIS timing diagram under *Constraint-A*;
3 Assign modeling parameters to components;
4 // first optimization
5 **for** $I_{bias,sf}$ *in range* **do**
6 **for** $I_{bias,cmp}$ *in range* **do**
7 Compute analog latency t_{sf} and t_{az} by Eqn. (4) under *Constraint-C1*;
8 Compute digital latency t_{count} by Eqn. (7) under *Constraint-B1&C3*;
9 Compute ramp generator's bias current $I_{bias,ramp}$ (minimally required) under *Constraint-B2&C2*;
10 Compute noise by Eqn. (21);
11 Compute $I_{bias,sf}$ and $I_{bias,ramp}$ (minimally required) by Eqn. (13) and Formula (17), compute $I_{ramp} = C_{ramp}\mathrm{Slope}_{ramp}$, compute E_{cmp}, $E_{counter}$ by Eqn. (14) to (16), read I_{periph} from user input, and compute power P_{sf}, P_{adc}, P_{ramp}, P_{periph} by Apdx. C;
12 Compute leakage currents by Eqn. (29) with the pre-defined Ratio_{leak}, and compute leakage power $P_{leak1} = \tilde{P}_{sf} + \tilde{P}_{cmp} + \tilde{P}_{ramp} + \tilde{P}_{periph}$ by substituting these currents into the corresponding power expressions;
13 **end**
14 **end**
15 Plot graph of $power_1 = P_{sf} + P_{cmp} + P_{counter} + P_{ramp} + P_{periph} + P_{leak1}$;
16 Plot graph of $power_1 \times noise = power_1 \times \overline{e_{n,read}}$;
17 Search for the optimal design under the optimization target and obtain the optimal $\{I_{bias,sf}, I_{bias,cmp}\}$;
18 // second optimization
19 **for** $f_{clk,io}$ *in range* **do**
20 Compute PLL power P_{pll} by Eqn. (18) under *Constraint-C3&C4*;
21 Compute IO power P_{io} by Eqn. (20) under *Constraint-C5*;
22 Compute digital power $P_{digital}$ by Eqn. (11);
23 Compute $\tilde{P}_{pll}$ (same procedure as Line 12), read $\tilde{P}_{digital}$ and $\tilde{P}_{PHY}$ from user input, compute leakage power $P_{leak2} = \tilde{P}_{pll} + \tilde{P}_{digital} + \tilde{P}_{PHY}$;
24 **end**
25 Compute $power_2 = P_{pll} + P_{digital} + P_{io} + P_{leak2}$;
26 Search for the power-minimized design and obtain the optimal $f_{clk,io}$;
27 Compute total power = $power_{1,opt} + power_{2,opt}$, FoM_1, and FoM_2;
28 **end**

TABLE I
KEY MODELING PARAMETERS IN THE CASE STUDY.

V_{analog} / $V_{digital}$	2.8 V / 1.2 V
Frame Resolution	600 × 500
Frame Rate	30 fps
Pixel Output Range	1.2 V to 2.4 V
$T_{countdown}$ / $T_{countup}$	0.5 / 1.5
$\overline{e_{n,dark}}$ / FWC / CG	30 e^- / 18600 e^- / 150 μV/e^-
$Ratio_{leak}$	0.2

I/O with *Constraint-C5*; we compute the power of digital processing and peripheral circuits; and we compute the leakage power of each component. With all the power components, we compute and plot the sum of them ($power_2$). This plot is a 2D curve with $f_{clk,io}$ being the independent variable. We search for the minimum of the 2D curve ($power_{2,opt}$) and find the corresponding $f_{clk,io}$.

Finally, we compute the total power and FoMs with $power_{1,opt}$ and $power_{2,opt}$, and we repeat the process for every desired I_{periph}.

C. Case Study

We use a toy example to show the DSE process.

CIS Configuration. We consider a CIS with column-level 10-bit ADCs, 600 × 500 pixel array, and 30 fps frame rate. No digital processing is included for simplicity. Some modeling parameters are estimated as follows: for the parameters of the transistors, we use a teaching-purposed datasheet [35]; for the parameters of the comparator, we use the design from a recent paper [24]; for the parameters of the VCO, we use a classical design [36] where $I_{bias,vco}$ = 2 mA; for the power of the D-PHY module P_{PHY}, we use a product from Texas Instruments [37] which consumes 150 mW in high-speed mode, 14 mW in low-power mode, and 0.75 mW in idle. To estimate the leakage power in Eqn. (12), we simply apply a unified ratio term to all the bias currents as the leakage currents:

$$I_{leak} = Ratio_{leak} \times I_{bias} \tag{29}$$

For example, $Ratio_{leak}$ = 0.2 means that all the leakage currents are 20% of their corresponding bias currents. Some of the key modeling parameters are defined in Tbl. I. During the DSE, I_{periph} is chosen from 4 μA to 2048 μA, the duty cycle is swept from 1% to 100%, $I_{bias,sf}$ and $I_{bias,cmp}$ are both swept from 10 nA to 10 μA, and $f_{clk,io}$ is swept from 3 MHz to 3 GHz.

First DSE. First, we only perform the first DSE and illustrate the latency-power-noise trade-off. Assuming I_{periph} = 100 μA and duty cycle is 15%, we plot $power_1$ under `Pow-opt` target in Fig. 12(a) and $power_1 \times \overline{e_{n,read}}$ under `PowNoi-opt` target in Fig. 12(b). The plots use the latency of SF and comparator as the x/y-axis rather than the current of SF and comparator for better visualization. The black star indicates the minimum point (optimal design) in each design space and beside the plot there shows the set of {$I_{bias,sf}$, $I_{bias,cmp}$} at the black star.

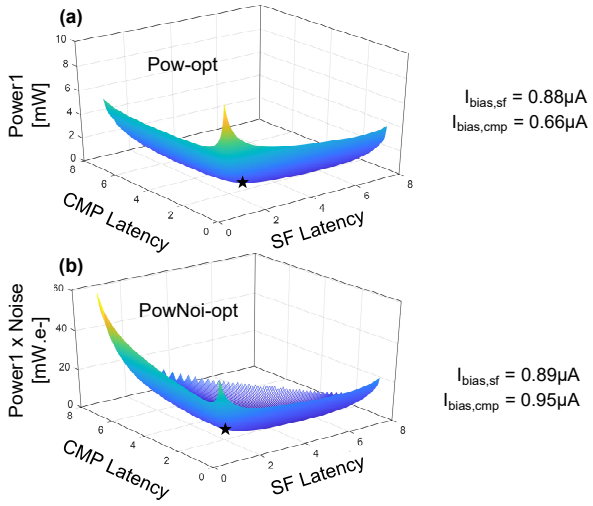


Fig. 12. (a) The plot of $power_1$ under `Pow-opt` target. (b) The plot of $power_1 \times \overline{e_{n,read}}$ under `PowNoi-opt` target. I_{periph} is fixed to 100 μA and duty cycle is fixed to 15%.

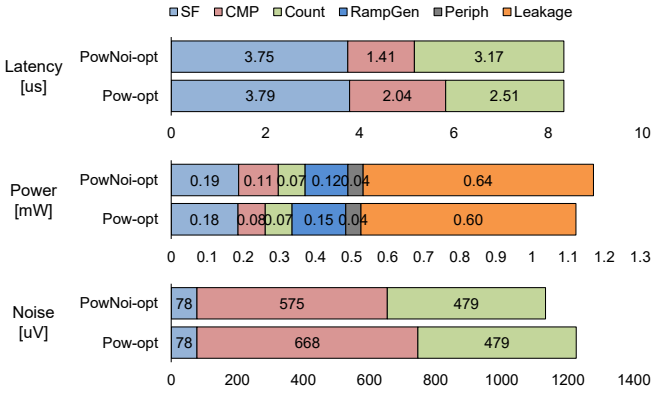


Fig. 13. The metric comparison between `Pow-opt` design and `PowNoi-opt` design as starred in Fig. 12. Note that in the noise plot the total noise is not the sum but the root-sum-square of the three components.

Fig. 13 compares the latency/power/noise breakdown of the two optimal designs in Fig. 12. We observe that `PowNoi-opt` design tends to increase $I_{bias,cmp}$ (from 0.66 μA to 0.95 μA) for lower comparator's noise (from 668 μV to 575 μV); and because of that, the comparator's power is increased (from 0.08 mW to 0.11 mW). Although the ADC's counter is slower to get smaller $I_{bias,ramp}$ and lower ramp generator's power (from 0.15 mW to 0.12 mW), the total CIS power still increases a bit.

Fig. 12 and Fig. 13 show the optimization effect of tuning two low-level circuit knobs; however, the systematic knob – duty cycle – also affects the optimization result. To illustrate the effect of duty cycle, we compare the trend of both FoMs (where only $power_1$ is taken into account) under `Pow-opt` target in Fig. 14, by sweeping over duty cycle (from 5% to 90%) for each desired I_{periph}. The color scale is made for each column, with green-yellow-red meaning the values from the lowest to the highest within the column. Although intuitively the duty cycle should be small for lower active power, Fig. 14 indicates that there exists a preferable range of duty cycles (green parts) to obtain the lowest FoMs. This

FoM1

Peripheral Current [uA]

Duty Cycle	4	8	16	32	64	128	256	512	1024	2048
5%	260	261	261	262	265	269	279	298	336	413
10%	151	151	152	153	156	162	173	195	240	329
15%	115	115	116	118	121	127	140	166	217	319
20%	97	98	99	101	104	111	126	154	212	326
25%	87	87	88	90	94	102	118	150	214	341
30%	80	81	82	84	88	97	115	150	220	360
35%	75	76	77	79	84	94	113	151	227	380
40%	72	72	73	76	81	92	112	154	237	402
45%	69	69	71	74	79	90	113	157	246	425
50%	67	67	69	72	78	90	114	161	257	448
55%	65	66	67	70	77	89	115	166	268	472
60%	63	64	66	69	76	90	117	171	279	496
65%	62	63	65	68	76	90	119	176	291	520
70%	61	62	64	68	75	90	121	181	302	544
75%	60	61	63	67	75	91	123	187	314	569
80%	60	61	63	67	75	92	125	192	326	594
85%	59	60	62	67	75	93	128	198	338	619
90%	58	60	62	67	76	94	131	204	350	644

FoM2

Peripheral Current [uA]

Duty Cycle	4	8	16	32	64	128	256	512	1024	2048
5%	1218	1219	1222	1228	1239	1261	1306	1396	1574	1932
10%	704	706	709	716	729	755	807	911	1119	1536
15%	537	539	542	550	565	594	654	773	1011	1486
20%	454	456	460	469	486	519	586	719	987	1522
25%	405	407	412	421	439	476	551	699	995	1589
30%	372	374	379	390	410	451	532	695	1020	1671
35%	349	352	357	368	391	435	524	701	1056	1766
40%	331	334	340	352	376	424	520	712	1096	1864
45%	319	322	329	342	368	419	523	730	1144	1972
50%	309	312	319	333	361	416	527	748	1191	2078
55%	300	304	311	326	355	415	533	769	1241	2185
60%	294	298	306	322	353	416	542	793	1296	2302
65%	287	291	300	316	349	415	548	812	1342	2401
70%	283	288	296	314	349	419	559	839	1399	2520
75%	279	284	293	311	348	422	569	863	1452	2630
80%	276	281	290	310	348	426	580	890	1508	2746
85%	273	279	289	309	350	431	593	918	1567	2866
90%	270	275	286	307	349	434	603	941	1617	2969

Fig. 14. The trend of FoM_1 [pJ/pixel] and FoM_2 [pJ/pixel · e^-] under `Pow-opt` target during the first DSE in the case study. $power_2$ (digital processing, PLL, and I/O) is excluded. The trend is similar under `PowNoi-opt` target.

FoM1

Peripheral Current [uA]

Duty Cycle	4	8	16	32	64	128	256	512	1024	2048
3%	1.66	1.66	1.66	1.66	1.67	1.67	1.68	1.70	1.73	1.80
4%	1.58	1.58	1.58	1.58	1.58	1.59	1.60	1.62	1.65	1.73
5%	1.56	1.56	1.56	1.56	1.56	1.57	1.58	1.60	1.64	1.71
6%	1.56	1.56	1.56	1.56	1.56	1.57	1.58	1.60	1.64	1.72
7%	1.57	1.57	1.57	1.57	1.57	1.58	1.59	1.61	1.65	1.73
8%	1.59	1.59	1.59	1.59	1.59	1.60	1.61	1.63	1.67	1.75
9%	1.61	1.61	1.61	1.61	1.61	1.62	1.63	1.65	1.69	1.78
10%	1.63	1.63	1.63	1.63	1.64	1.64	1.65	1.68	1.72	1.81
11%	1.66	1.66	1.66	1.66	1.66	1.67	1.68	1.70	1.75	1.84
12%	1.69	1.69	1.69	1.69	1.69	1.70	1.71	1.73	1.78	1.87
13%	1.72	1.72	1.72	1.72	1.72	1.73	1.74	1.76	1.81	1.91
14%	1.75	1.75	1.75	1.75	1.75	1.76	1.77	1.80	1.85	1.94
15%	1.78	1.78	1.78	1.78	1.78	1.79	1.80	1.83	1.88	1.98
16%	1.81	1.81	1.81	1.81	1.82	1.82	1.84	1.86	1.91	2.02
17%	1.84	1.84	1.84	1.84	1.85	1.85	1.87	1.89	1.95	2.06
18%	1.87	1.88	1.88	1.88	1.88	1.89	1.90	1.93	1.98	2.09
19%	1.91	1.91	1.91	1.91	1.91	1.92	1.94	1.96	2.02	2.13
20%	1.94	1.94	1.94	1.94	1.95	1.96	1.97	2.00	2.06	2.17

FoM2

Peripheral Current [uA]

Duty Cycle	4	8	16	32	64	128	256	512	1024	2048
3%	7.55	7.55	7.55	7.56	7.57	7.59	7.63	7.71	7.87	8.19
4%	7.36	7.36	7.37	7.37	7.38	7.40	7.45	7.53	7.71	8.05
5%	7.30	7.30	7.30	7.31	7.32	7.34	7.38	7.47	7.65	8.01
6%	7.29	7.29	7.29	7.30	7.31	7.33	7.38	7.47	7.66	8.03
7%	7.34	7.34	7.34	7.35	7.36	7.38	7.43	7.53	7.72	8.10
8%	7.41	7.41	7.42	7.42	7.44	7.46	7.51	7.61	7.81	8.20
9%	7.51	7.51	7.51	7.52	7.53	7.56	7.61	7.71	7.91	8.32
10%	7.62	7.62	7.62	7.63	7.64	7.67	7.72	7.83	8.04	8.45
11%	7.74	7.74	7.74	7.75	7.77	7.79	7.85	7.95	8.17	8.59
12%	7.87	7.87	7.87	7.88	7.89	7.92	7.98	8.09	8.31	8.75
13%	8.01	8.01	8.01	8.02	8.04	8.06	8.12	8.23	8.46	8.91
14%	8.14	8.14	8.15	8.15	8.17	8.20	8.26	8.37	8.60	9.07
15%	8.29	8.30	8.30	8.31	8.32	8.35	8.41	8.53	8.77	9.24
16%	8.43	8.43	8.43	8.44	8.46	8.49	8.55	8.67	8.91	9.40
17%	8.58	8.59	8.59	8.60	8.61	8.64	8.71	8.83	9.08	9.58
18%	8.74	8.74	8.74	8.75	8.77	8.80	8.86	8.99	9.25	9.76
19%	8.90	8.90	8.90	8.91	8.93	8.96	9.02	9.16	9.42	9.94
20%	9.05	9.05	9.06	9.07	9.08	9.12	9.18	9.32	9.59	10.12

Fig. 15. The trend of FoM_1 [nJ/pixel] and FoM_2 [nJ/pixel · e^-] under `Pow-opt` target during the full two-step DSE in the case study. The trend is similar under `PowNoi-opt` target.

is the consequence of the trade-off between active power and leakage power. For example, this range is 25% to 30% for I_{periph} = 512 µA under `Pow-opt` target. We observe the similar trend when the optimization target is `PowNoi-opt`, and the plot is not shown in Fig. 14 for brevity.

Second DSE. Second, we perform the full two-step DSE and only look at the second DSE, where the last low-level circuit knob $f_{clk,io}$ is tuned to minimize $power_2$. We observe that $power_2$ linearly increases with duty cycle, and the optimal $f_{clk,io}$ usually takes the fastest value under the constraints. This is because in $power_2$ the dynamic power only depends on the number of bits transmitted and does not change with duty cycle; while the static power is larger than the dynamic power and proportional to duty cycle. Thus the optimization tends to quickly finish the transmission and shut down the PLL and the I/O as soon as possible.

Two-Step DSE. Finally, we perform the full two-step DSE and compare the trend of FoMs (with total CIS power) under `Pow-opt` target in Fig. 15, by sweeping over duty cycle (from 3% to 20%) for each desired I_{periph}. Although the area of the optimal duty cycles is not as dispersed as that in Fig. 14, it still shows that the smallest duty cycle is not the best choice for the lowest FoMs. In this case study, the preferred duty cycle is 5% or 6%. Again, similar trend is observed under `PowNoi-opt` target and is not shown in Fig. 15 for brevity.

VI. Validation with Chip Results

We validate our optimization results with two published CIS works and analyze the observations. Note that it is impractical

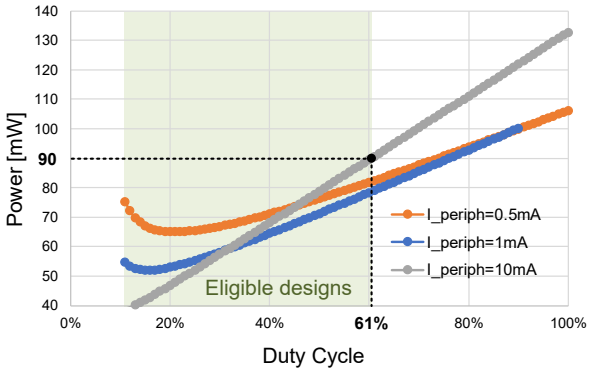


Fig. 16. Power-DutyCycle trade-off under different I_{periph} in OV2740.

TABLE II
Comparison between the actual design and one optimized design in CIS with column-leve ADC validation.

	$I_{bias,sf}$	$I_{bias,cmp}$	$Ratio_{leak}$	Power	Noise
Actual Design	N/A	N/A	210 µA*	< 90 mW	N/A
`Pow-opt` Design	2.93 µA	1.44 µA	0.42	65 mW	4.78 e^-

to get complete modeling parameters from the published works, however, we estimate those missing parameters with our best guess. Calculating the accurate number is not the purpose of this validation; rather, we aim to see the actual design is indeed included in the design space supported by our modeling method, and there still exists improvement opportunities unveiled by our optimization method, thereby achieving the goal of assisting real sensor design.

A. Validating CIS with Column-Level ADC

As the validation case for CIS with column-level ADC, we choose OV2740 [38], a commercial CIS product from OmniVision. OV2740 is an ultra low-power 1080p/60fps high definition image sensor for front-facing camera applications, such as smartphones and digital still cameras. It consumes less than 90 mW active power and 210 µA standby current. The sensor's diagram can be found on its product brief.

To achieve the required power efficiency, we use the `Pow-opt` target while configuring our modeling parameters to be as similar as the ones disclosed in OV2740's product brief. Since the actual I_{periph} in OV2740 is unknown, we try three I_{periph} values – 0.5 mA, 1 mA, and 10 mA – and adjust the leakage current ratio $Ratio_{leak}$ corresponding to each of them based on the sensor's standby current. For example, we set $Ratio_{leak}$ to 0.42 when I_{periph} = 0.5 mA, which generates a leakage current $\tilde{I}_{periph}$ = 210 µA. Note that $Ratio_{leak}$ also applies to other bias currents and thus the total leakage current is larger, but it does not affect our discussion. Besides, the static power of PLL is doubled because two PLLs are implemented in OV2740.

The trade-off between sensor power and sensor duty cycle for the three I_{periph} values is plotted in Fig. 16. With the target of consuming less than 90 mW power, the duty cycle needs to be less than 61%; and the duty cycle cannot be smaller than 10% otherwise there is no sufficient time to finish quantization

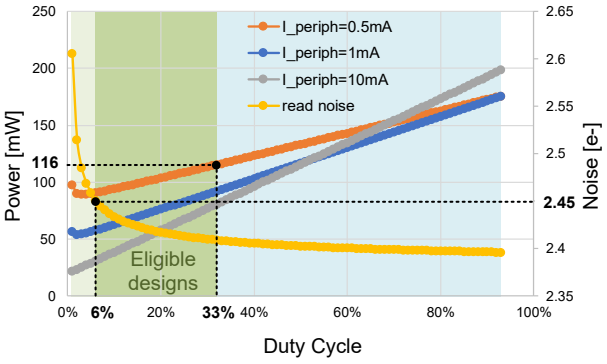


Fig. 17. Power-DutyCycle trade-off and Power-Noise trade-off under different I_{periph} in Samsung sensor.

TABLE III
COMPARISON BETWEEN THE ACTUAL DESIGN AND ONE OPTIMIZED DESIGN IN CIS WITH PIXEL-LEVEL ADC VALIDATION.

	$I_{bias,sf}$	$I_{bias,cmp}$	$Ratio_{leak}$	Power	Noise
Actual Design	50 nA	20 nA	N/A	116.2 mW	2.45 e^-
`PowNoi-opt` Design	10 nA	10 nA	0.21	57.9 mW	2.45 e^-

within the given frame rate (*Constraint-B1*). Therefore, the data points within the light green region are eligible designs.

Although all the eligible designs satisfy the requirement, carefully choosing the duty cycle gives the minimal power consumption. We pick the optimal design at I_{periph} = 0.5 mA and DutyCycle = 21% and compare against the actual OV2740 design in Tbl. II. Since the product brief does not disclose design details, we can only compare the power. The `Pow-opt` design consumes approximately 25 mW less power, one of the reasons of which is that we do not model the variable gain stage between pixel and ADC, the ISP, and the Serial Camera Control Bus interface in OV2740. Nonetheless, knowing the margin helps system architects better allocate power resources on the chip.

B. Validating CIS with Pixel-Level ADC

As the validation case for CIS with pixel-level ADC, we choose a JSSC work from Samsung [39]. The Samsung sensor is an ultra low-noise 2-Mega-pixel 30fps image sensor. It consumes 116.2 mW power with 2.45 e^- random noise. The sensor's diagram can be found in its paper.

To achieve both the required power efficiency and noise performance, we use the `PowNoi-opt` target while configuring our modeling parameters to be as similar as the ones in the Samsung sensor. To reduce read noise, the sensor adopts circuit techniques including sub-threshold SF and comparator, increased ramp slope, and small AZ capacitors, all of which are supported by our models and can be easily configured.

The trade-off between sensor power and sensor duty cycle and the trade-off between sensor noise and sensor duty cycle for different I_{periph} values are plotted in Fig. 17. With the target of consuming less than 116 mW power, the duty cycle needs to be less than 33% (light green region); and with the target of generating less than 2.45 e^- input-referred read noise, duty cycle is required to be larger than 6% (light blue region).

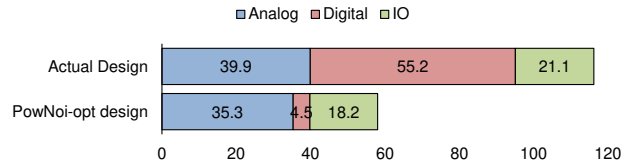


Fig. 18. Power breakdown comparison between the actual design and the `PowNoi-opt` design.

Therefore, the data points within the overlapped dark green shadowed region are eligible designs.

We pick one of those eligible designs (I_{periph} = 1 mA and DutyCycle = 6%) and compare against the actual Samsung sensor design in Tbl. III. The actual design uses extremely small bias currents at SF and comparator and our `PowNoi-opt` design indicates the same optimization direction. Under the same noise performance, the `PowNoi-opt` design consumes approximately 60 mW less power, and part of the reasons comes from not modeling the per-pixel positive-feedback amplifier, the per-pixel SRAM, and the ISP in the Samsung sensor. Fig. 18 shows the power breakdown comparison between the two designs. We observe that the analog power and the I/O power of the two are similar while the digital power has significant difference, which indicates the design margin for the SRAM and the ISP.

VII. Conclusion and Future Work

In this paper we propose a systematic modeling and optimization method that facilitates early-stage CIS design. The proposed modeling method covers common CIS architectures and makes fast performance estimations while only requiring less-detailed modeling parameters. The proposed optimization method utilizes the design knobs, which are exposed by the modeling and can be conveniently tuned in actual sensor hardware, and searches the optimal design under desired targets. The optimization process is validated with two real designs, showing reasonable approximations to real measurements and exhibiting insightful opportunities for further performance improvement.

As future work, we will make our modeling more comprehensive and make our optimization more systematic. Specifically, we will add the programmable analog gain stage between pixel and ADC, support more ADC architectures, include the clock for digital processing, and consider complete noise sources (random telegraph noise and 1/f noise). Besides, we will include area constraints and sensor chip floorplanning in the DSE and optimization.

References

[1] A. El Gamal and H. Eltoukhy, "Cmos image sensors," *IEEE Circuits and Devices Magazine*, vol. 21, no. 3, pp. 6–20, 2005.
[2] A. J. Theuwissen, "Cmos image sensors: State-of-the-art," *Solid-State Electronics*, vol. 52, no. 9, pp. 1401–1406, 2008.
[3] M. Kwon and B. Murmann, "A new figure of merit equation for analog-to-digital converters in cmos image sensors," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2018, pp. 1–5.
[4] J. Ohta, *Smart CMOS image sensors and applications*. CRC press, 2020.
[5] M. Kwon, S. Lim, H. Lee, I.-S. Ha, M.-Y. Kim, I.-J. Seo, S. Lee, Y. Choi, K. Kim, H. Lee *et al.*, "A low-power 65/14nm stacked cmos image sensor," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2020, pp. 1–4.

[6] C. Liu, L. Bainbridge, A. Berkovich, S. Chen, W. Gao, T.-H. Tsai, K. Mori, R. Ikeno, M. Uno, T. Isozaki *et al.*, "A 4.6 μm, 512× 512, ultra-low power stacked digital pixel sensor with triple quantization and 127db dynamic range," in *2020 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2020, pp. 16–1.

[7] M. J. Loinaz, K. J. Singh, A. J. Blanksby, D. A. Inglis, K. Azadet, and B. D. Ackland, "A 200-mw, 3.3-v, cmos color camera ic producing 352/spl times/288 24-b video at 30 frames/s," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 12, pp. 2092–2103, 1998.

[8] J. Gomez, S. Patel, S. S. Sarwar, Z. Li, R. Capoccia, Z. Wang, R. Pinkham, A. Berkovich, T.-H. Tsai, B. De Salvo *et al.*, "Distributed on-sensor compute system for ar/vr devices: A semi-analytical simulation framework for power estimation," *arXiv preprint arXiv:2203.07474*, 2022.

[9] T. Ma, Y. Feng, X. Zhang, and Y. Zhu, "Camj: Enabling system-level energy modeling and architectural exploration for in-sensor visual computing," in *Proceedings of the 50th Annual International Symposium on Computer Architecture*, 2023, pp. 1–14.

[10] R. D. Gow, D. Renshaw, K. Findlater, L. Grant, S. J. McLeod, J. Hart, and R. L. Nicol, "A comprehensive tool for modeling cmos image-sensor-noise performance," *IEEE Transactions on Electron Devices*, vol. 54, no. 6, pp. 1321–1329, 2007.

[11] E. R. Fossum, "Modeling the performance of single-bit and multi-bit quanta image sensors," *IEEE Journal of the Electron Devices Society*, vol. 1, no. 9, pp. 166–174, 2013.

[12] N. Callens and G. G. Gielen, "Analysis and comparison of readout architectures and analog-to-digital converters for 3d-stacked cmos image sensors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 8, pp. 3117–3130, 2021.

[13] Y.-H. Chang and A. T. Yang, "Analytic macromodeling and simulation of tightly-coupled mixed analog-digital circuits," in *1992 IEEE/ACM International Conference on Computer-Aided Design*. IEEE, 1992, pp. 244–247.

[14] C. Svensson and J. J. Wikner, "Power consumption of analog circuits: a tutorial," *Analog Integrated Circuits and Signal Processing*, vol. 65, pp. 171–184, 2010.

[15] A. J. Scholten, L. F. Tiemeijer, R. Van Langevelde, R. J. Havens, A. T. Zegers-van Duijnhoven, and V. C. Venezia, "Noise modeling for rf cmos circuit simulation," *IEEE Transactions on Electron Devices*, vol. 50, no. 3, pp. 618–632, 2003.

[16] "xmodel: Empower SystemVerilog with Event-Driven Analog Models ," https://www.scianalog.com/xmodel/.

[17] E. Lauwers and G. Gielen, "Power estimation methods for analog circuits for architectural exploration of integrated systems," *IEEE transactions on very large scale integration (VLSI) systems*, vol. 10, no. 2, pp. 155–162, 2002.

[18] J. Van Assche, R. Helsen, and G. Gielen, "Efficsense: an architectural pathfinding framework for energy-constrained sensor applications," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 136–141.

[19] "Hamamatsu Photonics Technical note: CMOS linear image sensors ," https://www.hamamatsu.com/content/dam/hamamatsu-photonics/sites/documents/99_SALES_LIBRARY/ssd/cmos_linear_kmpd9017e.pdf.

[20] Y. Nitta, Y. Muramatsu, K. Amano, T. Toyama, K. Mishina, A. Suzuki, T. Taura, A. Kato, M. Kikuchi, Y. Yasui *et al.*, "High-speed digital double sampling with analog cds on column parallel adc architecture for low-noise active pixel sensor," in *2006 IEEE International Solid State Circuits Conference-Digest of Technical Papers*. IEEE, 2006, pp. 2024–2031.

[21] B. Razavi, *Design of Analog CMOS Integrated Circuits*, 1st ed. USA: McGraw-Hill, Inc., 2000.

[22] A. Suess, M. Guo, R. Jiang, X. Mou, Q. Huang, W. Yang, and S. Chen, "Physical modeling and parameter extraction for event-based vision sensors," in *Int'l. Image Sensor Workshop (IISW)(IISW, Edinburgh, 2023)*, vol. 5, 2023.

[23] P. G. Jespers and B. Murmann, *Systematic design of analog CMOS circuits*. Cambridge University Press, 2017.

[24] A. Matsuzawa, A. M. Alonso, L. Yu, M. Sung, and M. Miyahara, "A partial-multi-conversion single-slope adc with response-linearized rdac," *Signal*, vol. 10, no. 1, pp. 10–2, 2021.

[25] P.-S. Chou, C.-H. Chang, M. M. Mhala, C. C.-M. Liu, C. Y.-P. Chao, C.-Y. Huang, H. Tu, T. Wu, S.-F. Yeh, S. Takahashi *et al.*, "A 1.1 μm-pitch 13.5 mpixel 3d-stacked cmos image sensor featuring 230fps full-high-definition and 514fps high-definition videos by reading 2 or 3 rows simultaneously using a column-switching matrix," in *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*. IEEE, 2018, pp. 88–90.

[26] K. Shu and E. Sánchez-Sinencio, *CMOS PLL synthesizers: analysis and design*. Springer Science & Business Media, 2006, vol. 783.

[27] S.-F. Yeh, K.-Y. Chou, H.-Y. Tu, C. Y.-P. Chao, and F.-L. Hsueh, "A 0.66 e rms- temporal-readout-noise 3-d-stacked cmos image sensor with conditional correlated multiple sampling technique," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 2, pp. 527–537, 2017.

[28] "MIPI D – PHY^{SM}," https://www.mipi.org/specifications/d-phy#.

[29] "Understanding and Performing MIPI® D-PHY Physical Layer, CSI and DSI Protocol Layer Testing ," https://download.tek.com/document/61W_25772_0_HR.pdf.

[30] C. Y.-P. Chao, H. Tu, T. Wu, K.-Y. Chou, S.-F. Yeh, and F.-L. Hsueh, "Cmos image sensor random telegraph noise time constant extraction from correlated to uncorrelated double sampling," *IEEE Journal of the Electron Devices Society*, vol. 5, no. 1, pp. 79–89, 2016.

[31] A. Boukhayma, *Ultra low noise CMOS image sensors*. Springer Cham, 2018.

[32] R. Balasubramonian, A. B. Kahng, N. Muralimanohar, A. Shafiee, and V. Srinivas, "Cacti 7: New tools for interconnect exploration in innovative off-chip memories," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 14, no. 2, pp. 1–25, 2017.

[33] J. Choi, J. Shin, D. Kang, and D.-S. Park, "Always-on cmos image sensor for mobile and wearable devices," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 1, pp. 130–140, 2015.

[34] J. Lee, H. Park, B. Song, K. Kim, J. Eom, K. Kim, and J. Burm, "High frame-rate vga cmos image sensor using non-memory capacitor two-step single-slope adcs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 9, pp. 2147–2155, 2015.

[35] "65nm CMOS Process Data Sheet for the Analog IC Design Course," https://www.eit.lth.se/fileadmin/eit/courses/etin30/65nm_CMOS_Process_Data_Sheet.pdf.

[36] S. Levantino, C. Samori, A. Bonfanti, S. L. Gierkink, A. L. Lacaita, and V. Boccuzzi, "Frequency dependence on bias current in 5 ghz cmos vcos: Impact on tuning range and flicker noise upconversion," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 8, pp. 1003–1011, 2002.

[37] "SNx5DPHY440SS CSI-2/DSI DPHY Re-timer," https://www.ti.com/lit/ds/symlink/sn65dphy440ss.pdf?ts=1717907634720&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FSN65DPHY440SS.

[38] "1/6" CMOS 1080p (1920 x 1080) HD Image Sensor with PureCel® Technology ," https://www.ovt.com/products/ov2740/.

[39] M.-W. Seo, M. Chu, H.-Y. Jung, S. Kim, J. Song, D. Bae, S. Lee, J. Lee, S.-Y. Kim, J. Lee *et al.*, "2.45 e-rms low-random-noise, 598.5 mw low-power, and 1.2 kfps high-speed 2-mp global shutter cmos image sensor with pixel-level adc and memory," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 4, pp. 1125–1137, 2022.

[40] S. Hamami, L. Fleshel, and O. Yadid-Pecht, "Cmos aps imager employing 3.3 v 12 bit 6.3 ms/s pipelined adc," in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512)*, vol. 4. IEEE, 2004, pp. IV–960.

APPENDIX A
LIST OF MODELING PARAMETERS

The list of minimum-required modeling parameters is in Tbl. IV.

APPENDIX B
MODELING CIS WITH CHIP-LEVEL ADC

CIS with chip-level ADC were popular during late 1990s and early 2000s [7], [40] and have been gradually replaced by CIS with column/pixel-level ADC, due to their high speed requirement on ADCs and wideband thermal noise. Nonetheless, we provide a typical timing diagram and necessary modeling modifications to make our modeling scope comprehensive.

Operation Timing. CIS with chip-level ADC typically adopts rolling shutter exposure. As shown in Fig. 19, the timing diagram is similar to the CIS with column-level ADC, except that the readout and the I/O transmission are performed interchangeably. Again, the digital processing stage is neglected for simplicity.

TABLE IV
LIST OF MODELING PARAMETERS.

Component	Parameter
General	V_{analog}, $V_{digital}$, DutyCycle, FrameRate, $Ratio_{leak}$, dim_h, dim_v
Transistor	V_{th}, kp, $V_{sat,const}$
Pixel	$\overline{e_{n,dark}}$, e_{sig}, FWC, CG, C_{pixel}
Source Follower	$V_{sf,signal,min}$, $I_{bias,sf}$, $g_{m,sf}$, $g_{mb,sf}$, $r_{o,sfbias}$, $\left(\frac{W}{L}\right)_{sf}$, $\left(\frac{W}{L}\right)_{sfbias}$
ADC Comparator	$V_{ref,cmp}$, $I_{bias,cmp}$, $\left(g_m/I_d\right)_{cmp}$, $\left(g_m/I_d\right)_{cmpbias}$, $\left(g_m/I_d\right)_{cmpload}$, $g_{m,cmp}$, $g_{d,cmp}$, $g_{mb,cmp}$, $g_{m,cmpload}$, $g_{d,cmpload}$, $g_{mb,cmpload}$, $g_{d,cmpbias}$, $C_{in,cmp}$, $C_{load,cmp}$, $T_{countdown}$, $T_{countup}$, n_e
ADC Counter	$reso_{adc}$, $C_{counter}$, $NumFlip_{adc}$
Ramp Generator	V_{ramp0}, I_{ramp}, $I_{bias,ramp}$, C_{ramp}
PLL	I_{biascp}, $I_{bias,lf}$, $I_{bias,vco}$, C_{pfd}, $C_{divider,prescale}$, $C_{divider,count}$, $C_{load,cp}$, $C_{load,lf}$, $f_{ref,pll}$, $f_{clk,vco}$, M, N, A
I/O	Num_{lane}, $C_{load,io}$, $f_{clk,io}$, P_{PHY}
Digital Processing	$P_{digital,dyn}$, $P_{digital,stat}$
Peripherals	I_{periph}

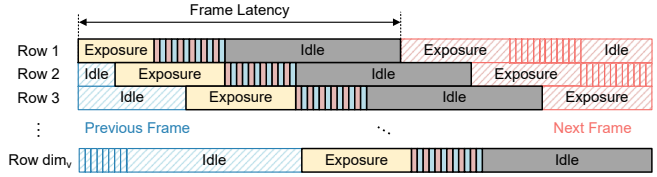


Fig. 19. Timing pipeline of CIS with chip-level ADC. The digital processing stage is neglected. The readout stage and the I/O stage are performed interchangeably. The stage length is not scaled.

ADC Type. Due to the high requirement on ADC speed, faster ADC such as flash ADC and pipelined ADC are usually used instead of SS ADC. Yet if the SS ADC is still used, the ramp generator now drives only one comparator rather than dim_h comparators in the CIS with column-level ADC.

APPENDIX C
SUPPLEMENT OF MODELING EQUATIONS

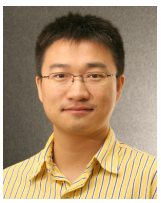
Power equations:

- SF: $P_{sf} = V_{analog} \times I_{bias,sf}$
- ADC: $P_{adc} = E_{adc}/t_{readout} = \left(E_{cmp} + E_{counter}\right)/t_{readout}$
- Ramp generator: $P_{ramp} = V_{analog} \times \left(I_{ramp} + I_{bias,ramp}\right)$
- PLL: $P_{pfd} = C_{pfd}V_{digital}^2 f_{clk,vco}/(M \times N + A)$, $P_{divider} = C_{divider,prescale}V_{digital}^2 f_{clk,vco} + C_{divider,count}V_{digital}^2 f_{clk,vco}/N$, $P_{cp} = V_{analog} \times I_{bias,cp}$, $P_{lf} = V_{analog} \times I_{bias,lf}$, $P_{vco} = V_{analog} \times I_{bias,vco}$
- Peripherals: $P_{periph} = V_{analog} \times I_{periph}$

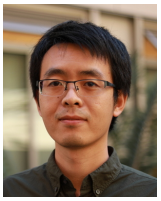
Other equations:

- SF gain: $A_{sf} = \frac{1}{g_{mb,sf}} \| r_{o,sf} \| r_{o,sfbias} \Big/ \left(\frac{1}{g_{mb,sf}} \| r_{o,sf} \| r_{o,sfbias} + \frac{1}{g_{m,sf}} \right)$
- Comparator DC gain: $A_{dc,cmp} = V_{analog}/\left(V_{fs,ramp}/2^{reso_{adc}}\right)$
- Comparator output resistance: $r_{o,cmp} \approx \left[\left(1 + \frac{g_{m,cmp} - g_{mb,cmp}}{g_{d,cmpbias}}\right)/g_{d,cmp}\right] \| \frac{1}{g_{m,cmpload} + g_{mb,cmpload} + g_{d,cmpload}}$

Tianrui Ma Tianrui Ma (S'22, M'24) received the B.S. (Hons) degree in Electrical Engineering from Beihang University, Beijing, China, in 2019, and the M.S. degree and the Ph.D. degree in Electrical Engineering from Washington University in St. Louis, St. Louis, MO, USA, in 2023 and 2024, respectively. He is currently a Research Assistant Professor at Institute of Computing Technology, Chinese Academy of Sciences. Dr. Ma is a recipient of ISLPED 2022 Best Paper Award.



Zhe Gao Zhe Gao received the B.S. degree in Electrical Engineering from Tianjin University in 2007, and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Rochester, Rochester, NY, USA, in 2009 and 2013, respectively. He is currently a R&D Manager with OMNIVISION, Santa Clara, CA, USA. His current interests include CMOS image sensor and related system design.



Zhe Chen Zhe Chen (M'16) received the B.S. degree in Electrical Engineering from Tsinghua University in 2011, and the Ph.D. degree from the Institute of Semiconductors, Chinese Academy of Sciences in 2016. After that, he joined the University of California, Los Angeles as a postdoctoral scholar. He received the Best Paper Award at the IEEE/ACM ISLPED in 2018. In 2022, he joined the OmniVision Technologies Inc. as a Senior Digital Designer.

Ramakrishna Kakarala Dr. Ramakrishna Kakarala (SM '01) is a Director of Algorithm Development at Omnivision Technologies.

Charles (Qingwei) Shan Qingwei Shan earned his Bachelor's and Master's degrees in Electrical Engineering from Tsinghua University in Beijing, China, in 1985 and 1988, respectively. He currently serves as the Senior Vice President of Design Engineering at OMNIVISION in Santa Clara, CA, USA. His professional interests focus on CMOS image sensor technology and project development.

Weidong Cao Dr. Weidong Cao (S'16, M'22) is an Assistant Professor with the Department of Electrical and Computer Engineering, The George Washington University, Washington, DC, USA. Before that, he was a Principal Engineer with TSMC Corporate Research, San Jose, CA, USA. His research interests focus on VLSI Design, Computer Architecture, EDA, and Quantum Computing. Dr. Cao is the recipient of IEEE/ACM ISLPED Best Paper Award in 2022 and his work has also been nominated for Best Paper Award at DATE 2019 and DAC 2022.

Xuan Zhang Dr. Xuan 'Silvia' Zhang (S'08, M'15, SM'22) is an Associate Professor in the Electrical and Computer Engineering Department at Northeastern University. She works across the fields of integrated circuits/VLSI design, computer architecture, and electronic design automation. Dr. Zhang is an IEEE Women in Engineering (WiE) Distinguished Lecturer for 2023-2024, IEEE Circuits and Systems Society (CAS) Distinguished Lecturer for 2022-2023, and the recipient of NSF CAREER Award in 2020. She currently serves as the Associate Editor-in-Chief at IEEE Transactions on Circuits and Systems I (TCAS-I) and Associate Editor at IEEE Transactions on Computer-Aided Designs (TCAD). Her work has received numerous best paper awards and nominations including ISLPED Best Paper Award in 2022, AsianHOST Best Paper Award in 2020, DATE Best Paper Award in 2019, and nomination for Best Paper Awards at DAC 2022, ASP-DAC 2021, MLCAD 2020, DATE 2019, and DAC 2017.