Image Sensor Matrix High Speed Simulation

Z. Feng, V. Viswanathan, D. Navarro, and I. O’Connor

Abstract—This paper presents a new high speed simulation methodology to solve the long simulation time problem of CMOS image sensor matrix. Generally, for integrating the pixel matrix in SOC and simulating the system performance, designers try to model the pixel in various modeling languages such as VHDL-AMS, SystemC or Matlab. We introduce a new alternative method based on model spice in cadence design platform to achieve accuracy and reduce simulation time. The simulation results indicate that the pixel output voltage maximum error is at 0.7812% and time consumption reduces from 2.2 days to 13 minutes achieving about 240X speed-up for the 256x256 pixel matrix.

Keywords—CMOS image sensor, high speed simulation, image sensor matrix simulation.

I. INTRODUCTION

As reviewed by E. R. Fossum and M. Bigas in the papers [1][2], owing to its linear output response of the incident light intensity, the 3T-APS is the popular and basic pixel architecture in image sensors. A spice model has been reported by T. Reiner [3] studied mainly on the capacitance on the sensitive node and pixel transfer function. A detailed analysis has been given by A. El Gamal [4] on sensor dynamic range, system SNR, and several methods for improving the DR. The APS and PPS pixel has been modeled in VHDL-AMS in order to predict the chip behavior before the fabrication [5], but we can find that this model is not able to be as accurate as the spice model and did not take the noise and parasitic capacitor into account. Another APS VHDL model take the sense node nonlinearity into consideration and with the other functional block, but it suffers from internal errors when the imager matrix size becomes too big, the simulation time consumption is also an unavoidable problem [6]. The spice simulator based on numerical analysis, such as Spectre, Hspice, solves the circuit equation by iterative method, such as Newton’s method, Newton-Raphson method. To be specific, for the imager resolution 256x256, the tool has to solve equations with respect to 196608 mosfet transistors and 65536 photodiodes suppose that the pixel is a 3T-APS. For simulating the whole sensor matrix, the simulator has to solve 130K equations [10]. For solving this big time and calculation consumption problem, we propose a high level simulation in Cadence design platform to simulate the imager on matrix level and check the sensor matrix performance in early stages of design time.

We introduce the image sensor pixel models and basic signal processing procedure in the section II. In section III, we will discuss the fast simulation method and present a comparison result. Finally, in section IV we conclude and provide the future perspectives.

II. IMAGE SENSOR MODEL AND SIGNAL ANALYSIS

As shown in Fig 1, the 3T-APS consists of three NMOS transistors and a photodiode. Typical signal readout procedure can be roughly divided into three phases, such as reset, integration and readout. During the reset operation, the sense node full well capacitor Cpd which comprises the photodiode inner capacitor, reset transistor MN1 source capacitor, and the MN2 gate parasitic capacitor will be reset to a voltage Vpd. The charges will be accumulated on the sense nodes and it can gain maximum charge Qmax which depends on the full well capacitance of the photodiode. After resetting, the capacitor Cpd will be discharged by the photocurrent (Iph) and dark current (Idark), the former is proportional to the light intensity during the integration time and Idark is the leakage current flowing through the photodiode when no photons enter the image pixel, the main part of the total dark current is coming from the depletion of the photodiode edge at the surface [7]. As shown in Fig 2, the slope of discharge curve is determined by the sum of Iph and Idark and it also determine by the Cpd. The bigger the Iph is, the faster the Cpd discharges. For example, using three different light intensities as input to the simulation produces three different Iph giving three discharge curves with different slopes, as shown in Fig 2. Meanwhile, the discharge voltage is buffered by inner source follower MN2. The access transistor MN3 passes this voltage to column bus according to the readout timing. The column voltage will be sampled twice by the Correlated Double Sampling (CDS) circuit, which is used to reduce reset noise and fixed pattern noise [8]. The first sample happens in the reset period and the second sample happens during integration. The final output voltage can be expressed by equation (1):

\[ V_{OS} = V_{CDS-H} - V_{CDS-S} \]  

where, \( V_{CDS-H} \) is the output voltage in reset phase and \( V_{CDS-S} \) is the output voltage sampled during the integration. The differential voltage \( V_{OS} \) is the final signal passed to Analog to Digital converter (ADC) for generating the image numeric data.
III. HIGH SPEED SIMULATION METHOD AND IMPLEMENTATION

The spice simulator is commonly used to simulate the circuit behavior. Imager matrix is built up from pixel blocks that are repeated thousands or millions of times. The number of the circuit element in such a matrix is too huge for the common spice simulator. The reason why the Spectre simulator suffers from memory and runtime problem for huge pixel matrix is that the nets and terminals in the matrix form a very complicated equation with lots of variables according to Kirchhoff’s Current Law (KCL) and Kirchhoff’s Voltage Law (KVL) principle. It costs lot of time to solve the equations and it consumes lot of memory to execute the calculation. Currently designers simulate pixel matrix in a very limited size, such as 10x10 or 20x20 before the fabrication owing to these difficulties. In order to give high level performance estimation, we present our high level and high speed approach. The main idea in this work is to use one pixel instead of the whole matrix in the simulation with the help of the generated database to map the input photocurrent and output voltage. We use 3T-APS as a standard cell as shown in the Fig 3, in order to obtain an correlated output voltage database \( V[i][j] \) we perform a parametric simulation with the photocurrent parameter \( I_{ph}[i][j] \) which is the input signal of the pixel with address row \( i \) and column \( j \) in the pixel matrix. In this way, we convert the photocurrent values to its correlated output voltage database and we form a Look Up Table (LUT) by these data shown as the blocks marked \( I_{ph}[i][j] \) and \( V[i][j] \) in Fig 3.

| TABLE I  
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Photocurrent (A)</td>
<td>Output voltage (Volts)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( a: {I_{ph}_00} )</td>
<td>( V_a ) ( V_{a1} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( b: {I_{ph}_01} )</td>
<td>( V_b ) ( V_{b1} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( c: {I_{ph}_02} )</td>
<td>( V_c ) ( V_{c2} )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x )</td>
<td>( y )</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We assume that the photocurrent is a random unknown value as \( x \) and there is a threshold photocurrent in each range such as \( (a+b)/2 \) in region \( [a \ b] \). As shown in Fig 4, since the
point m1 has a bigger photocurrent value than the threshold \((a+b)/2\), so the correlated output CDS voltage \(V_{\text{m1}}\) is closer to \(V_a\) than \(V_b\) according to the circuit behavior presented by dashed line which stands for the output voltage increasing linearly along with the photocurrent. Every output voltage corresponds to an intensity data value in the grey level image, for example, \(V_a\) is converted to 165 in the numeric intensity data in the image and \(V_b\) to 164. Owing to the \(V_{\text{m1}}\) is closed to \(V_a\) and there is no available integer value between 164 and 165 in the image. So the voltage \(V_{\text{m1}}\) could be evaluated to \(V_a\) and \(V_{\text{m2}}\) could be \(V_b\) if the simulation results is shown as an grey level image. So in this way as shown in Fig 3, every photocurrent \(I(i,j)\) value in the scene will be matched in the generated database and every correlated output voltage \(V(i,j)\) will be estimated.

We have implemented this LUT function in our own image sensor matrix simulation tool box shown in Fig 5. This tool can help to investigate how the pixel performance affects the output image quality [9]. Two levels simulation are available for checking the pixel performance, user can investigate the single pixel character by performing the normal simulation and he can also perform the matrix level simulation which can be used to investigate the high level performance. For example, user can investigate the affection of pixel temperature changes on the final output image intensity data. Further more, the column capacitance has been taken into account in the big matrix simulation. With the help of this high speed approach, 130K equations needed to be solved by spice simulator for simulating the pixel matrix 256x256 have been reduced to 4.8K in our new approach and simulation time has been greatly saved. The time performance is shown in Fig 6. The time consumption of Spectre classical simulation increases exponentially with matrix row numbers, whereas it increases slowly after the initialization which takes 12 minutes to form the LUT database in the new fast simulation method.

![Image sensor simulation tool box in Cadence ADE](image)

**Fig. 5** Image sensor simulation tool box in Cadence ADE

Table II is used to show the time consumption comparison. For the matrix in small size, such as 4x4, 8x8 and 12x12 pixels, the classical simulation time consumption is 7 minutes, but the simulation time is 8 hours when the matrix size reach 100x100. In contrast, the initialization time for the fast simulation is around 12 minutes and the mapping function can be finished very quickly, so this fast simulation method is quite useful for the simulation when the matrix size is bigger than 12x12. The mapping function consumes about 1 second to map the whole matrix output of size 1024x768.

<table>
<thead>
<tr>
<th>Matrix size</th>
<th>Classical (min)</th>
<th>Initialization(min)</th>
<th>Mapping(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4x4</td>
<td>3.7</td>
<td>11.57</td>
<td>23.6</td>
</tr>
<tr>
<td>8x8</td>
<td>7.5</td>
<td>12.01</td>
<td>34.6</td>
</tr>
<tr>
<td>12x12</td>
<td>179.23</td>
<td>12.21</td>
<td>39.5</td>
</tr>
<tr>
<td>256</td>
<td>3465</td>
<td>12.31</td>
<td>91.4</td>
</tr>
<tr>
<td>512</td>
<td>3653</td>
<td></td>
<td>483</td>
</tr>
</tbody>
</table>

As it is a high level estimation method, validation of the results with a reference simulation is performed. The validation is based on AMS 0.35um technology and 3T-APS model with the fixed integration time. It is performed by simulating 256x256 classical simulations. The following image generation method [9] is used to convert the output voltage into gray level images.

\[
grey_{\text{relative}} = 255 \times \frac{V_{\text{os}}}{V_{\text{max}} - V_{\text{min}}} \]  

\[
grey_{\text{raw}} = 255 \times \frac{V_{\text{os}}}{V_{\text{CC}}} \]  

\[
grey_{\text{abs}} = 255 \times \frac{V_{\text{os}}}{V_{\text{rst}}} \]  

where, the \(V_{\text{os}}\) is the CDS output voltage, \(V_{\text{max}}\) and \(V_{\text{min}}\) is the maximum and minimum value of the pixel matrix \(V_{\text{os}}\) values, respectively. The results of classical Spectre simulation and the new approach are shown in Fig 7.
The Table IV shows the worst case of the simulation results and correlated intensity data. \(V_{\text{res}}\) is the output voltage at the reset stage which is equal to \(V_{\text{CDS-H}}\). \(V_{\text{sample}}\) is the output voltage after integration which is equal to \(V_{\text{CDS-S}}\). The maximum error of intensity data is 2 and it comes from the absolute image generation method shown in equation (4), using this equation does not correct the error in \(V_{\text{res}}\). It is about 0.78% (2/256) of data range 256 in grey level. The mean value which stands for the whole matrix simulation average error in intensity data is quite small and is about 0.1681% of the output data range.

**IV. CONCLUSION AND DISCUSSION**

This paper presents an alternative way of image sensor matrix simulation. This method is technology independent and the accurate spice model could be used to simulate the image sensor matrix instead of using a high level language model, such as VHDL-AMS model. The result has proved strongly that our new fast simulation approach can be used for accelerating the CMOS image sensor pixel matrix simulation. Due to the time consumption has been reduced largely, this makes the CMOS image sensor simulation more efficient and in turn it is possible to check the system performance in a very short time. This new improvement makes realizing the whole image generation and processing image data possible in Cadence. More cases should be simulated to verify and prove this high level and high speed simulation approach. The future work will be focused on the aspects such as: variability, technologies and pixel architectures. The pixel matrix FPN will be investigated and analyzed.

**REFERENCES**


