Outline

• **System Level Overview**
• Single Pixel
• Pixel Array
• Post-Processing
System-Level Overview

4 major blocks:

1. Pixel Array: light sensors (photodiodes + active device)
2. Control Logic: facilitates the pixel array sense/readout (counter, decoder, mux)
3. Programmable-Gain Amplifier: amplifies a sensed voltage after it is read out
4. ADC
System-Level Overview (2)

(very high-level block diagram)
Outline

• System Level Overview
• **Single Pixel**
  – Architecture
  – Operation
  – Design Issues
  – Simulation Results
• Pixel Array
• Post-Processing
Single Pixel (Schematic)

- N+/psub photodiode
- PMOS reset switch (for signal range)
- NMOS source-follower
- NMOS row-select device/switch
- Output to shared column bus
Single Pixel Operation - Reset

- Reset switch closed.
- Internal node’s parasitic capacitance pre-charged to $V_{dd}$ (diode is reverse-biased)
- Incident light on diode causes photocurrent $I_{photo}$
Single Pixel Operation - Exposure

- Open the reset switch
- Photocurrent will discharge the intermediate node, proportional to amount of incident light.
- Voltage at output of source-follower will follow.
Single Pixel Operation - Recap

- At end of exposure period, sample the voltage at output onto column capacitor
- Repeat if desired for next image to be taken.
Source-Follower Design

- Since ADC speed is 100ksamp/s, design for source-follower bandwidth > 100kHz
- Shooting for about 50uW power consumption of 128x128 array
  - At Vdd=1V, this is 50uA/128 = 0.4uA bias current.
  - Biasing is simplistic: gigantic resistor and diode-connected NMOS
Pass Transistor Sizing

- Reset (PMOS) device: chosen to minimize leakage (sized based on simulations)
- Row-Select (NMOS) device: chosen to minimize on-resistance (for performance)
Single Pixel Simulation Model

- Photodiode modeled as current source in parallel with reverse-biased diode
- Assuming junction cap. of 1.5 fF/um², diode capacitance ≈ 40fF
- Photocurrent is estimated as 1-10pA (derivation?)
- Assume Row-Select is high for the single pixel
Single Pixel Simulation

• Start by running DC/AC/transient simulations of source-follower alone; Verified bandwidth, bias points, gain, noise

• Added the switches (reset, row-select), the simulated photodiode, and column S/H capacitor and verified with transient simulation.

• Some Results:
  – Pixel output = 410mV during reset (DC)
  – Source-follower total integrated noise (SPICE .noise) < 200uV rms (does this matter?)
• Hand-calculated worst-case total integrated noise (Tian, Fowler, and El Gamal, 2001) rms:
  224uV (reset); 800uV (integration);
  100uV (readout); **total = 1.3mV rms**
Simulated noise: TBD, once I learn how to run noise analysis in Spectre
• Dark current (with 0A simulated photocurrent) = 1pA
  (equivalent to about 26.5W/m² light power…1/20 of sunlight’s intensity)
Outline

• System Level Overview
• Single Pixel
• **Pixel Array**
  – Column Circuitry
  – Rolling-Shutter Architecture
  – Control Logic/Circuits
  – Simulation Results
• Post-Processing
Column Circuitry

- Active load (provides bias current)
- Column S/H capacitors (sized at 1pF for now) will store one row’s read-out data at a time and hold it there while it is post-processed
- Buffer before sending signal to analog mux??? (since it will eventually have to drive PGA)
Rolling-Shutter

• All this means is that the pixels are exposed (the reset switch is opened) one row at a time, from top to bottom, in order.
• While one row is post-processed (by ADC), other rows are in varying stages of reset/exposure.
• Contrast to “snap-shot” mode where all pixels are exposed at the same time (though readout may be row-by-row).
• (cite some papers)
Rolling Shutter Relative Row Timing

rowsel[m]

rowsel[m+1] = reset[m]*

rowsel[m+2] = reset[m+1]*

(A) (B) (C)

X Y

For 128 rows at 100kSamp/s, period must be at least 1.28ms

A-B: row m’s exposure
B: row (m-1)’s values stored on column caps, are processed by ADC; row m’s active pixels are connected to the column buses
X: row m’s values get sampled onto column caps
C: row m gets reset, but its “measurements” are still stored on column caps and are processed by the ADC
Y: row (m+1)’s values get sampled onto column caps
Control Logic

• Inputs:
  – External clock
  – Reset signal (or equivalently, a “start” signal)

• Outputs:
  – Reset/Row-Select Signals for each row of the pixel array
  – Analog MUX addressing
  – Column-Sample signal
  – PGA control signals/clocks
  – ADC control signals/clocks

These control the (pixel) order of readout in the array
Control Logic – Implementation

- Row Selection: 7-bit ripple counter and 128-bit decoder
- Column Selection: 7-bit ripple counter feeding analog mux address
- Column-Sample Signal
- PGA/ADC control/clocks
Array Simulation

- So far, simulated an 8x8 array plus control logic (transient)
Array Simulation – in pictures
Outline

• System Level Overview
• Single Pixel
• Pixel Array
• Post-Processing
  – Programmable-Gain Amplifier
  – ADC
Programmable-Gain Amplifier

- Insert after analog mux….
- Used to boost the pixel output voltage and drive the ADC
- Pixel’s output voltage < 410mV; so set gain=2 to use more of the full-scale voltage (Vdd=1V)
- Control signals?
ADC

- Using Mike’s ADC
- Main issue will be generating control signals (clocks, etc.) on-chip; but it’s been done before.