Creapyx: An Innovative Pixel Evaluation Platform

CMOS Image Sensors for High Performance Applications Workshop

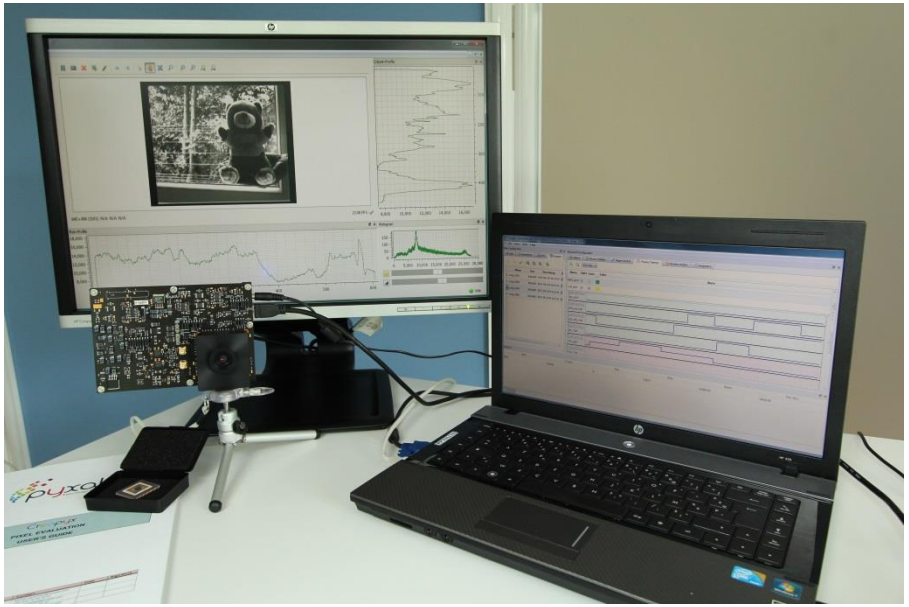
November 19, 2015

J. Michelot, P. Monsinjon, J. Caranana, A. Menard, M. Dubois, A. Lesire, C. Bouvier, S. Cohet, S. Caranhac

Contact: julien.michelot@pyxalis.com

Presentation Outline

- Motivations
- Flexibility
- Full environment overview
- Characterization strategy
- Application example
- Conclusion



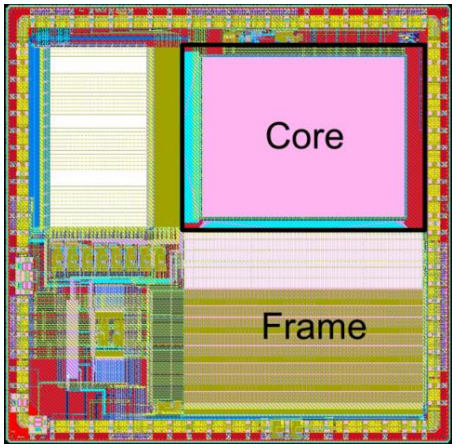
Full Creapyx pixel evaluation environment

Motivations

- Custom image sensor products often need custom pixel design
  - Need to assess the pixel performances in the early stages of the product design
  - Need to design a dedicated pixel test-chip and test environment (PCBs, software,…) => Costly and time consuming
- Image sensor foundries do not perform extensive pixel characterization
  - Need to perform home-made pixel characterization
  - Need of the deeper understanding of particular phenomenon
- Bright ideas require silicon
  - Perform pixel R&D (not everything can be simulated)
  - Assess new pixel architectures, new technology goodies
  - Perform demonstrations (proof of concepts)
- Assess an image sensor technology
  - Get the best performances out of a technology

Motivations

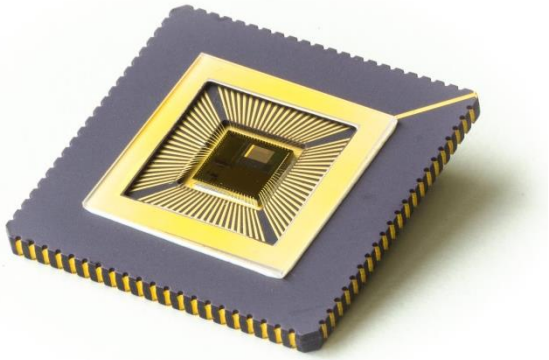
- Creapyx basic idea is to design one pixel evaluation kit (test-chip and test environment) once and for all
  - Decrease costs and design time
    - Re-use the same test-chip for all purposes
    - Only pixel array design is needed
  - Increase characterization throughput
    - No need to debug the pixel test environment
    - Run standard characterization routines
    - Perform sweeps to find the best pixel operation point
    - Use easy to handle packages/sockets (CLCC)



Test-chip layout

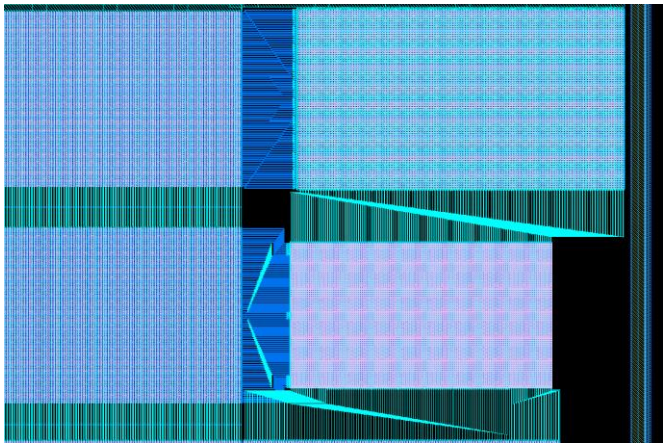
Main challenge:

Make the pixel evaluation environment as much flexible as possible

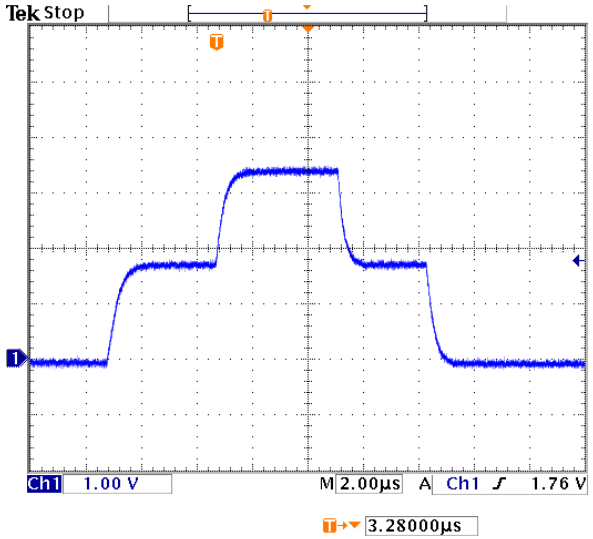

Packaged test-chip

Flexibility

- No predefined pixel pitch
  - Automated adaptation cell were developed
  - Fixed matrix surface (~ 7,7 mm²)
  - Maximal resolution is VGA (640 x 480)
- Up to 8 fully programmable pixel control signals per line
  - High/intermediate/low signal voltages are individually programmable
  - Slew rates can be individually controlled
  - Signals can be observed (using an oscilloscope) for debug purposes
- Highly flexible pixel readout sequencer
  - Supports all kind of readout methods (rolling shutter, global shutter, multiple readout, …)



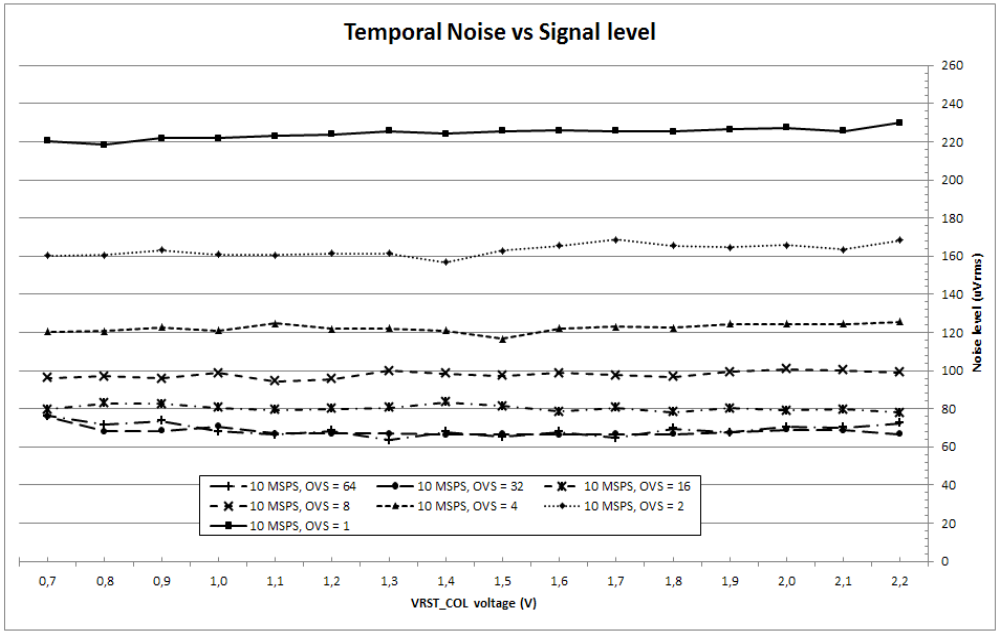
Pixel pitch adaptation cell



Observed pixel control signal

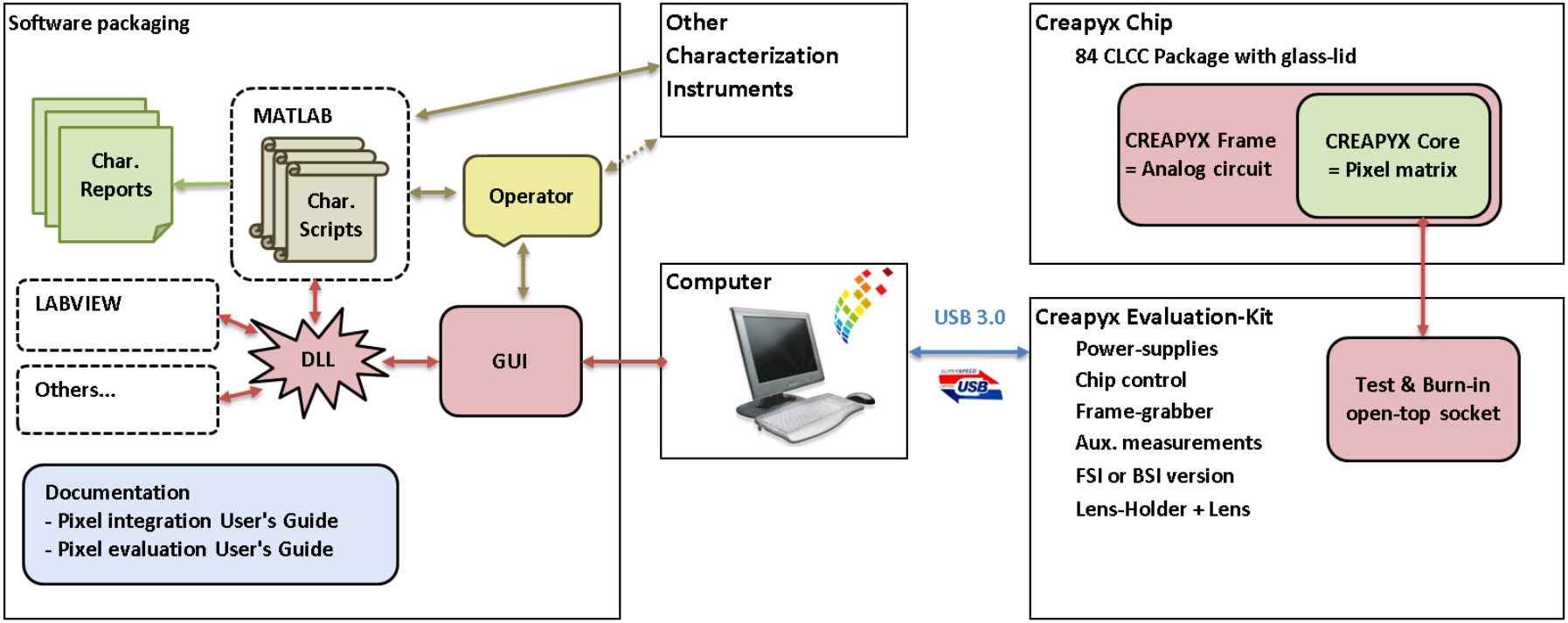
Flexibility

- Pixel readout can be performed at different readout noise performance using specific correlated multiple readout technique
  - Low noise readout (~ 220 µV rms) can be used to perform measurement at high frame rates (up to 25 fps)
  - Ultra low noise readout (~ 70 µV rms) can be used to perform pixel noise measurements at low frame rates



- Other goodies
  - Matrix power supply dynamic switching
  - Column bandwidth modulation
  - PMOS pixel readout
  - X, Y addressing
  - Custom image processing
  - …

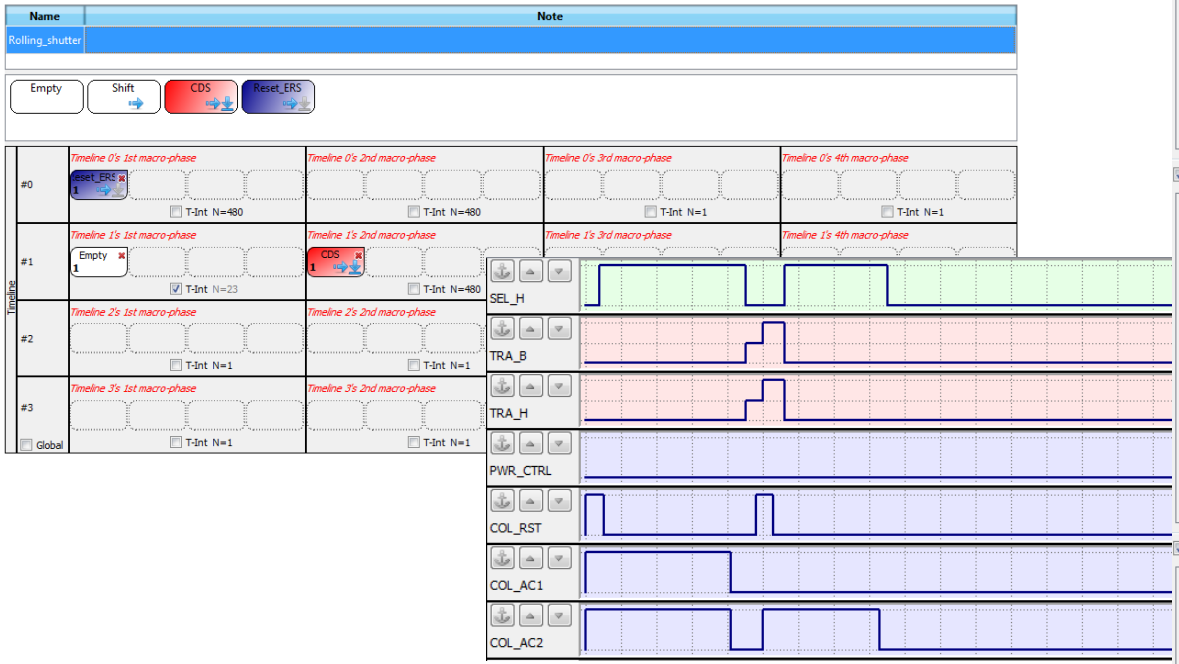
Full Environment Overview



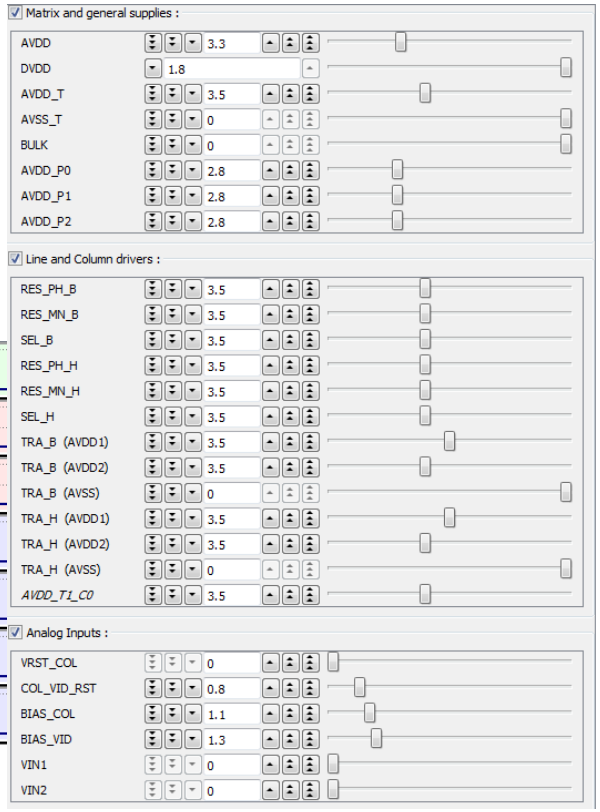
- Full pixel evaluation platform can be run from a laptop using simple USB connection
  - Also suited for demonstrations purposes

Characterization strategy

- Use Graphical User Interface (GUI) to setup pixel operating point
  - User friendly interface
  - Thanks to real time pixel timing/biasing control
  - Through real time pixel statistics



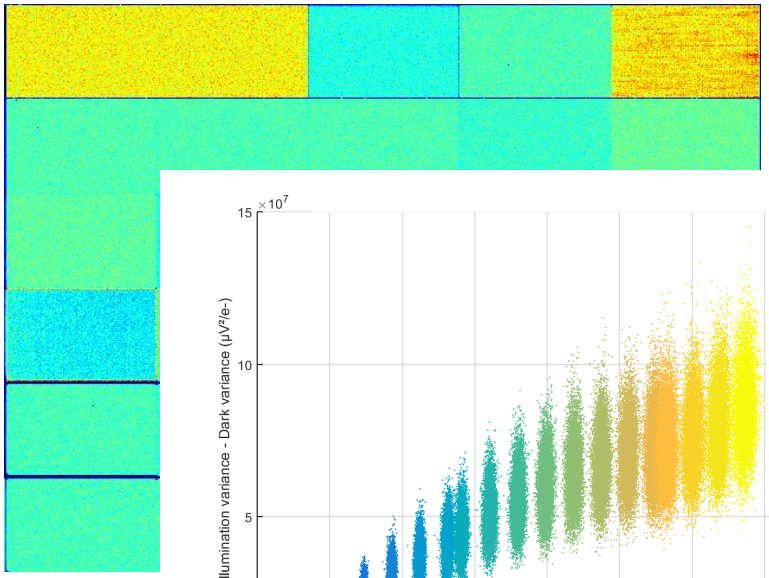
Pixel timing setup panels

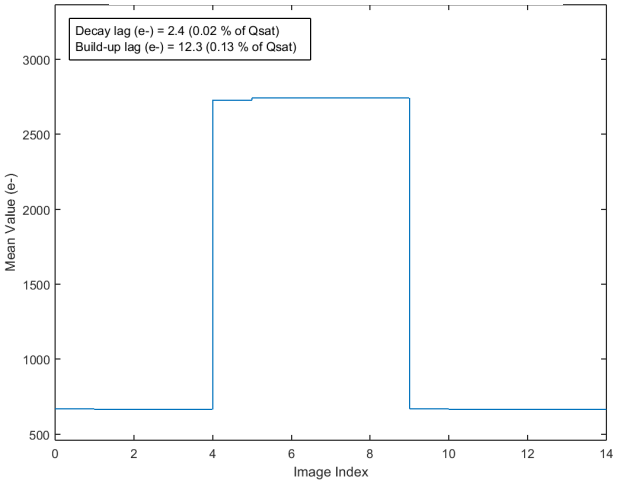


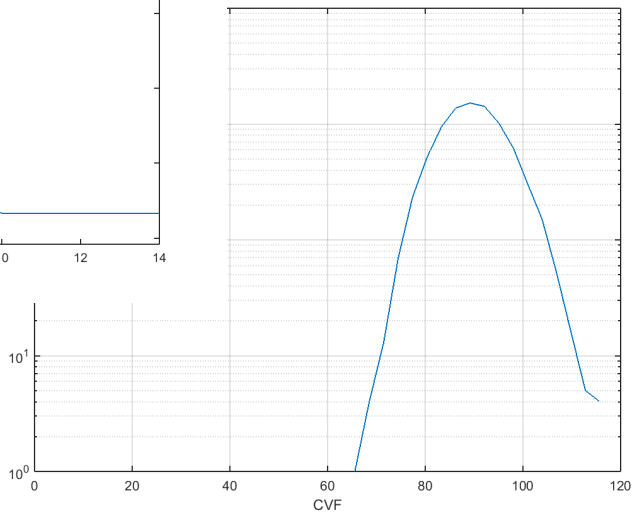
Pixel voltage biasing control panel

Characterization strategy

- Run Matlab or Labview automated characterization routines
  - Thanks to the DLL interface with GUI
  - Interface with other instruments are possible (Shutter, light source, photometer,…)
  - Spend less time on characterization, spend more time on understanding

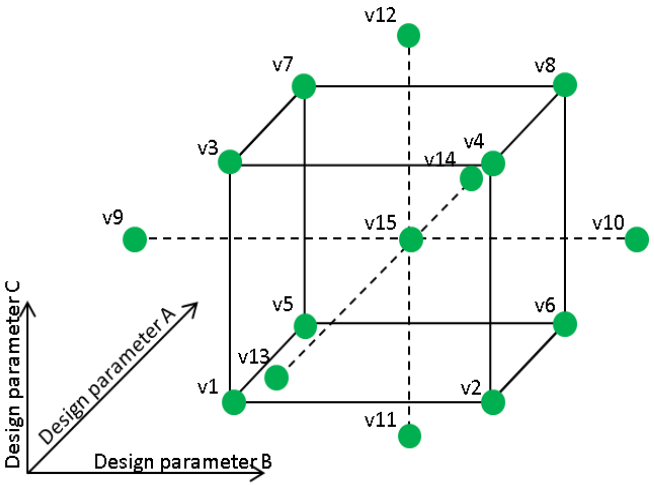




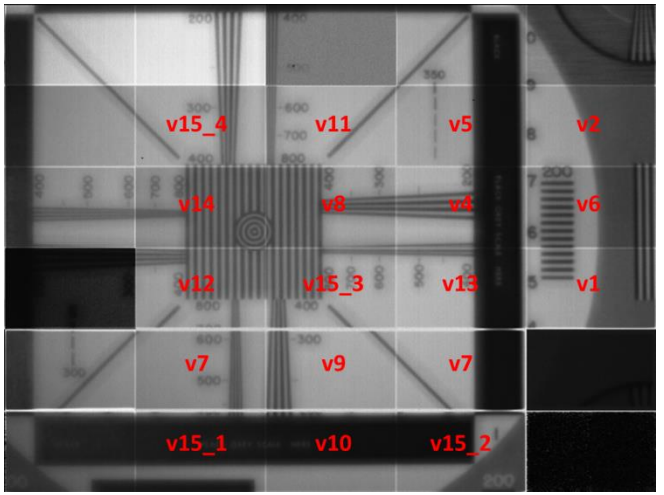


Application example

- Design of experiment (DOE) to find out technology best operating point
  - Use of a 3 factor central-composite design of experiment
    - DOE was realized thanks to a dedicated DOE software
    - 15 different pixel layouts have been embedded on CREAPYX
    - Center point (v15) has been duplicated 4 times to obtain the design pure error
  - DOE is done on small pitch global shutter pixels to find out the best achievable parasitic light sensitivity



3D representation of the design space

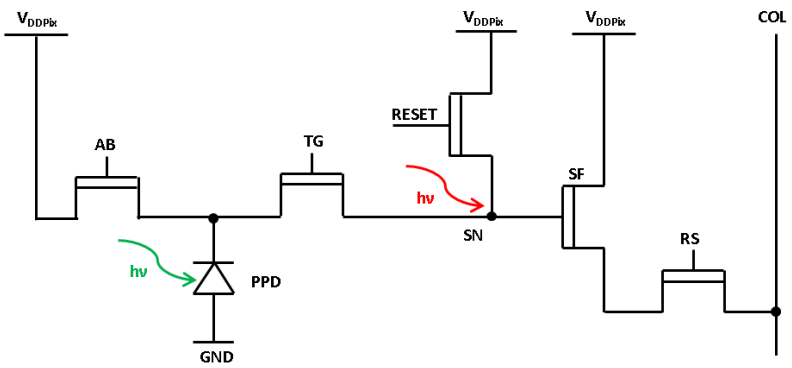


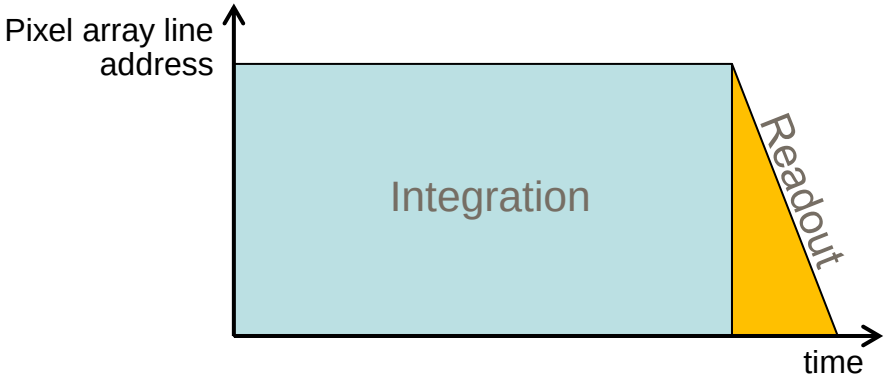
Focused image

Application example

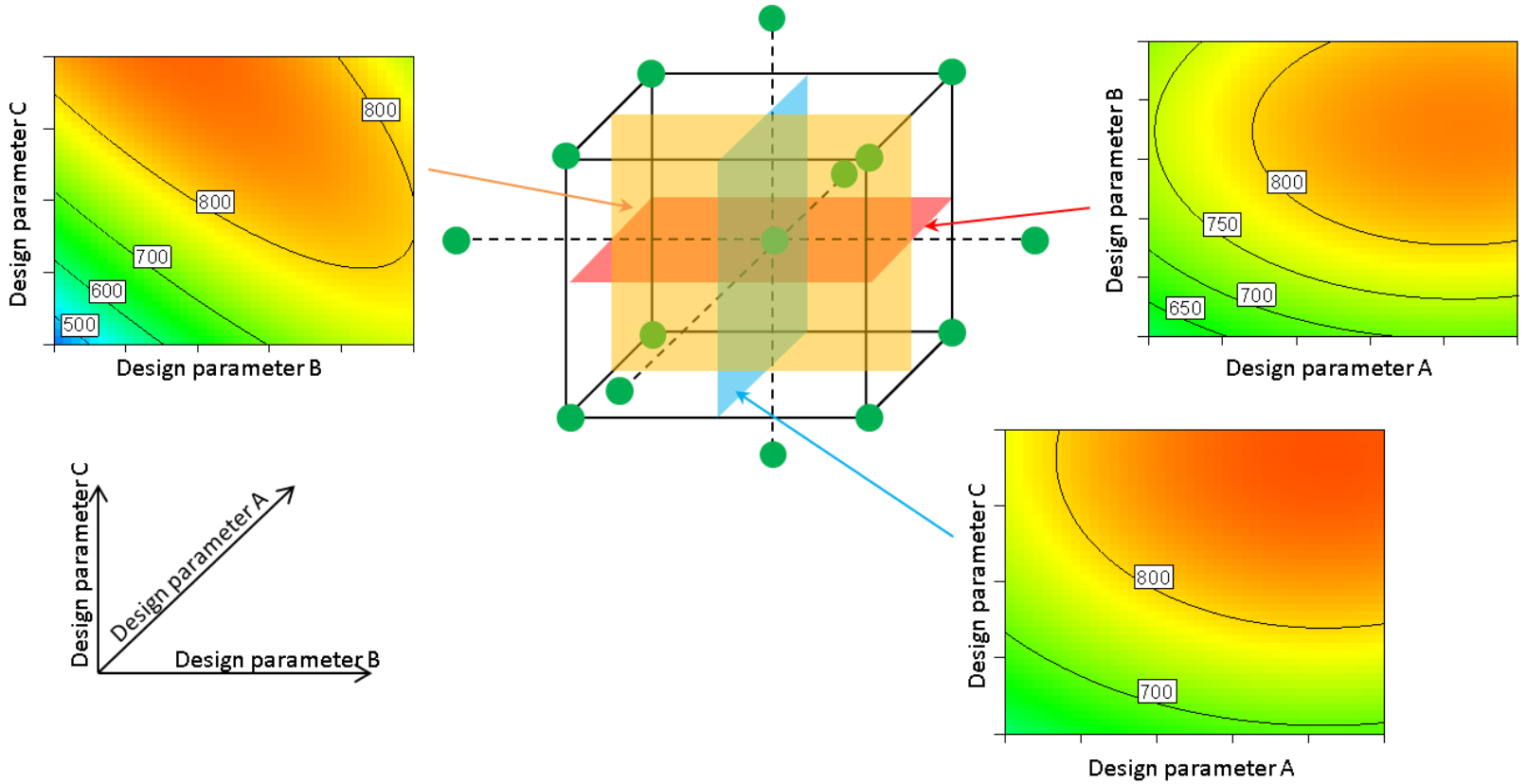
- Parasitic light sensitivity (PLS)
  - Is an undesired phenomenon in global shutter pixel architectures
  - Is problematic since integrated signal has to spend some time in the sensing node (SN) before being read out
  - Is defined as the ratio of the sensitivities of the pinned photodiode (PPD) and the sensitivity of the sensing node (SN)

$$\text{PLS} = \frac{PPD\ sensitivity}{SN\ sensitivity}$$



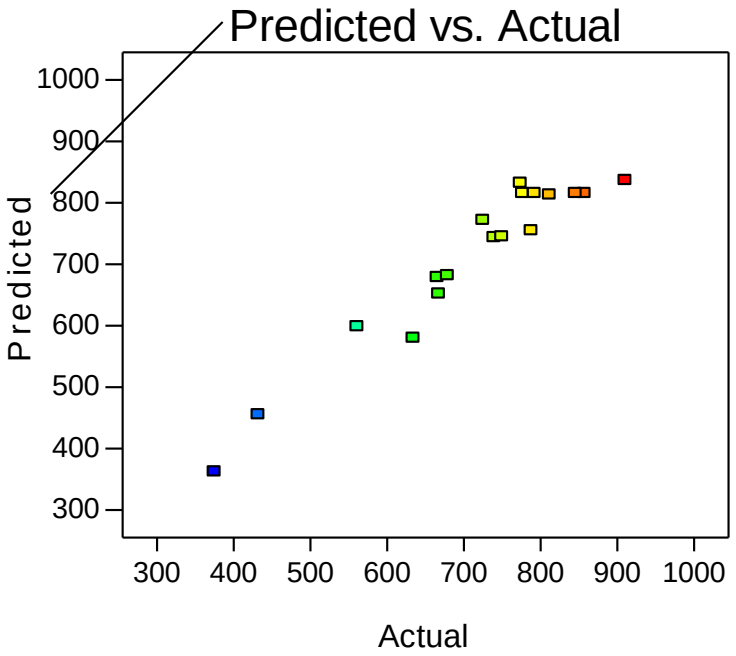


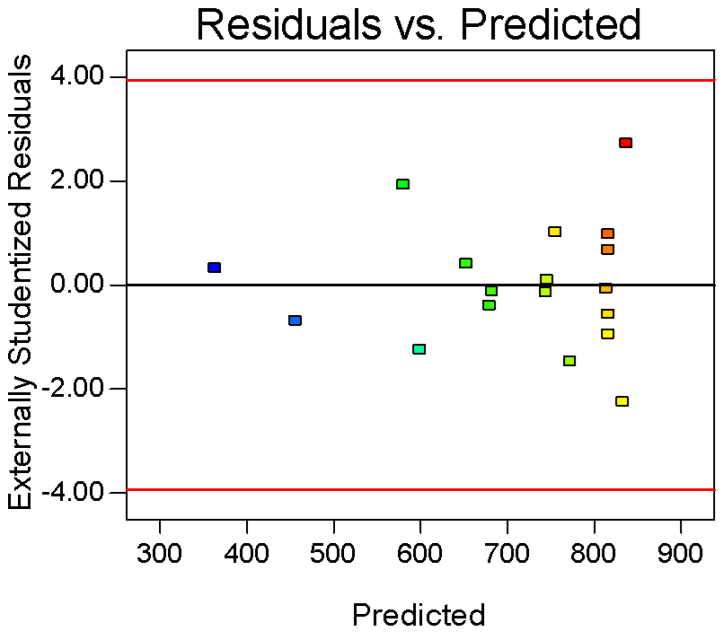
Application example



- Design parameters are linked to how to shield pixel sensing node (metal layers and implantation schemes)
- As can be seen from the extracted model, PLS shows a local extremum for a given triplet of design parameters (A, B and C)

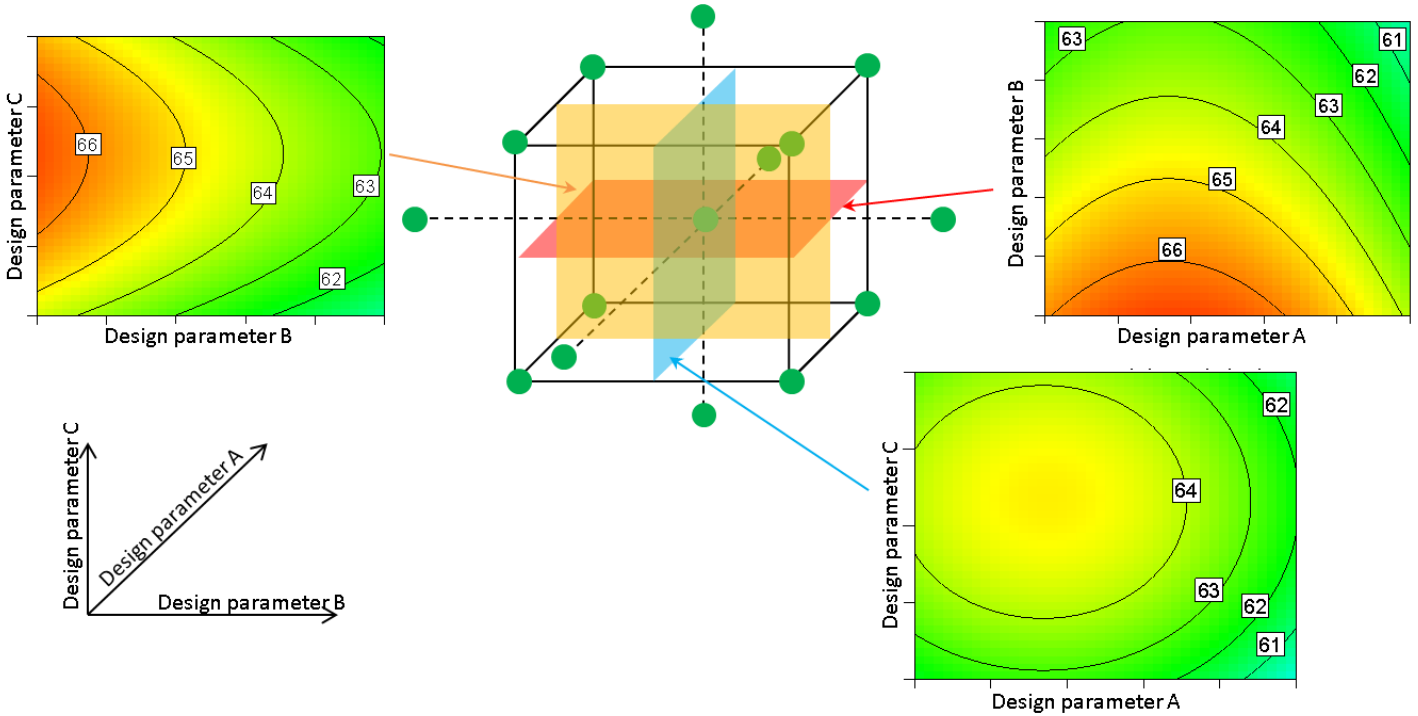
Application example





- It is possible to assess how well the extracted model fits to the actual data
- Since several center points (v15) have been introduced in the DOE it is also possible to extract a pure error (due to inaccurate measurements, lack of statistics or position in the pixel array)
- Here the standard deviation to predicted point to the model has been evaluated to 47.24
- A confidence interval may also be computed for a given model point

Application example



- As expected QE measurements show antagonist behavior vs PLS
  - The higher the PLS the lower the QE
- Depending on the target application it is possible to give priority either PLS or QE
- It is then possible to give the expected value for both parameters (PLS and QE) with their respective confidence intervals

Conclusions

- Creapyx evaluation kit is a unique pixel development platform
  - Its motto is quite simple: Flexibility
  - User friendly pixel setup (thanks to graphical user interface)
  - Re-use already validated characterization environment (test cards, pixel peripheral circuitry, measurement scripts,…)
  - Speed up characterization and focus on pixel development
- Creapyx can be used for lots of purposes
  - Drive pixel R&D
  - Secure projects
  - Get the best out of the technology
  - Demonstration
- Coupled to other tools (e.g. Design of experiment software) it can becomes very powerful

Conclusions

Creapyx

Thank you for your attention