SPAD CHIP

P. Fischer

Preface

- Grading of the ‘VLSI design’ will be based on a practical chip design.

- This includes
  - Schematic design
  - Analogue simulations
  - Mixed mode simulation
  - Layout with DRC and LVS

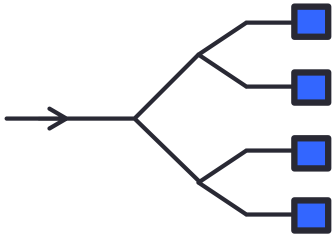
- There will be an opportunity to submit a chip on a real run of the IMS 350nm technology in spring.

- I therefore suggest to make a common design effort (in teams of 2) to produce a full SPAD readout chip.

- These slides document the goal of the design

Goal

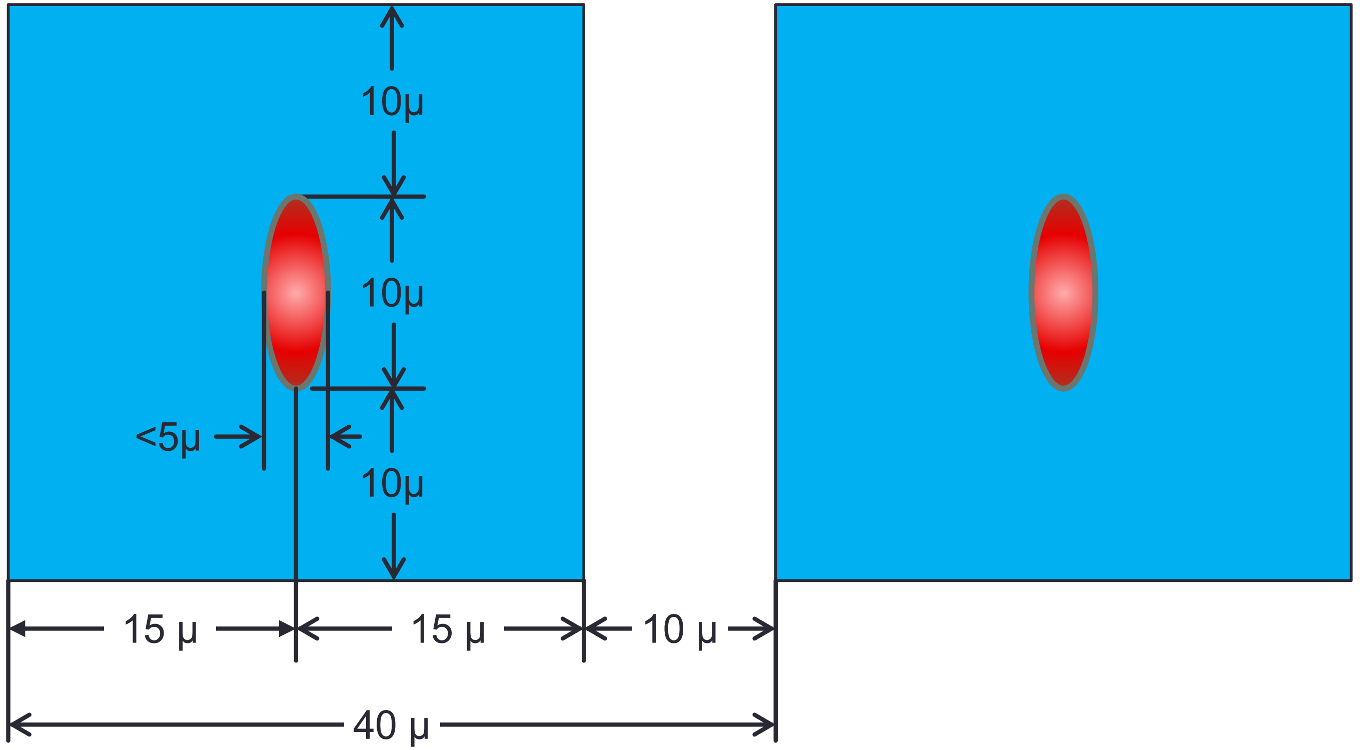
- We want to solve a real world problem as encountered by a KIP group:
- Photon bursts at known times propagate in an optical fiber.
  We want to measure the number of photons in each burst.
  (Bursts are spaced by ~100ns)
- Concept: dilute the photon number by sending photons to multiple fibers so that at most 1 photon is left per fiber. Observe each fiber output with one SPAD.



- Preliminary discussions with the users indicate that the SPADs should be arranged in a regular X-Y-grid with 40µm pitch (=repetition).

(SPAD Array Geometry)

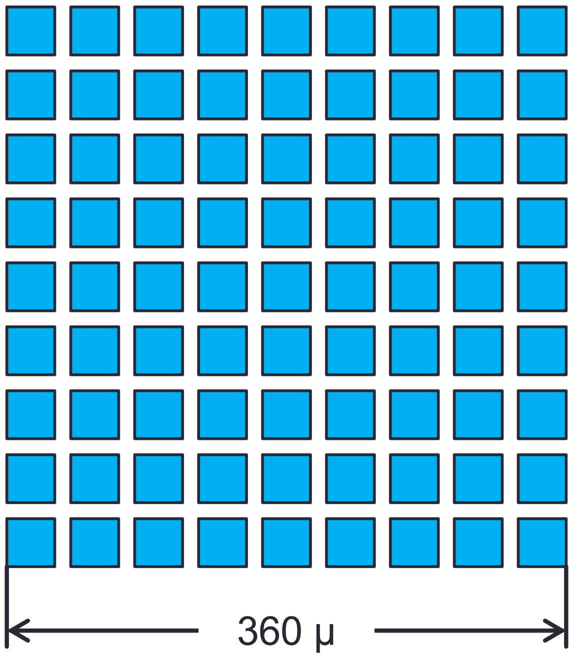
- Lateral fiber facet is oval of < 5 × 10 µm^2
- Pitch of splitters is ~40 µm. Can be adjusted arbitrarily by design.
- Leave ~10 µm lateral space for alignment tolerance
- Keep structure symmetric in x and y (assembly freedom, free space concept)
- -> use SPADs of 30 × 30 µm^2 in 40 µm pitch

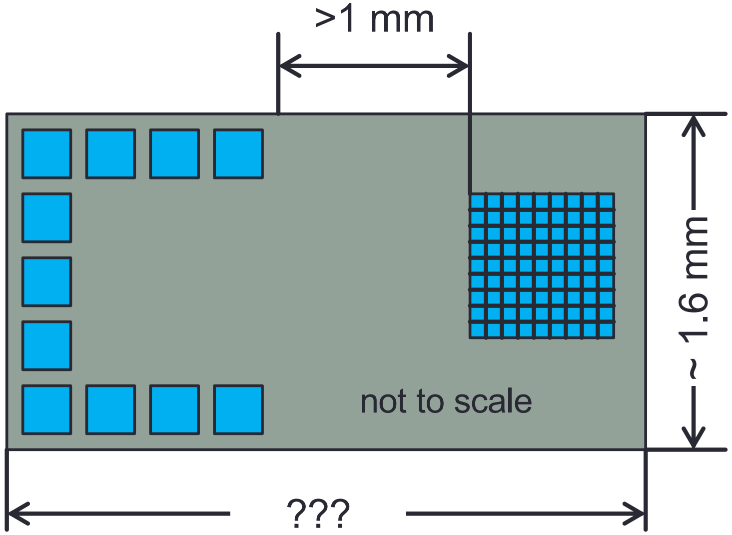


SPAD Array and Chip Size

- Number of SPADs not well defined.
- Odd number preferred to have a clear center SPAD
- Chose 9 x 9 (by chance, this is good for binary adder: 9 x 9 = 81 = 3 x 27)

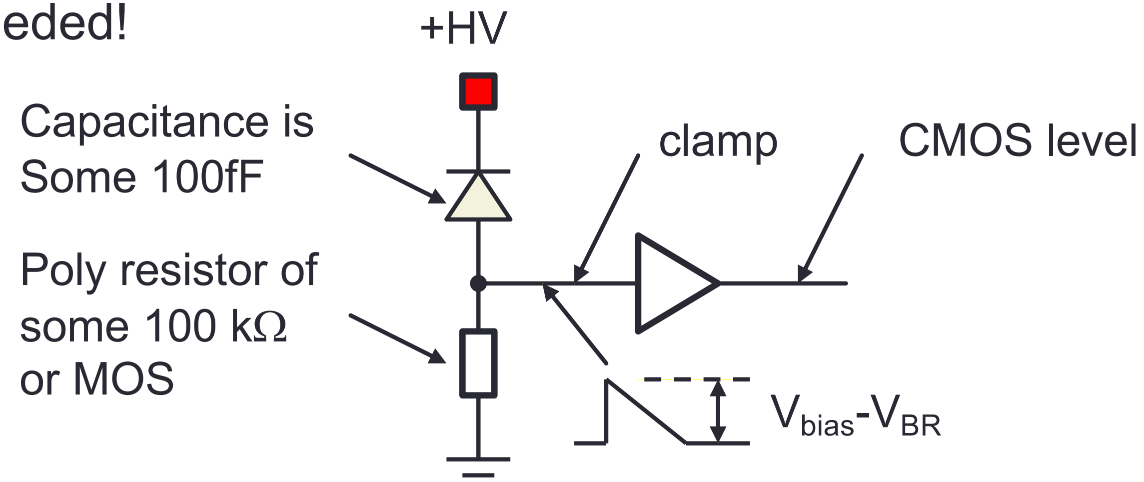
- Chip needs ~20 Bond pads. Our pitch is 168 µm.
- Decide on geometry so that fibers can be easily mounted to SPADs
- Keep chip as small as possible to save space





SPAD Readout

- SPADs are reverse biassed with positive voltage of ~ +30V
  When $V_{bias} > V_{BR}$ = 'Break down voltage', photons can trigger an avalanche so that voltage drops back down to V_{BR}.
    - Note: Bond Pad for V_{bias} may ***not*** have any input protection diodes!!!
    - Note: In order not to destroy following CMOS components, a clamp diode is needed!



- An 'amplifier' is used to produce a CMOS level:
    - No static consumption
    - Low input threshold
    - Fast for rising signal at input

Readout

- We provide the chip with a short GATE signal 
- All *rising* edges *during* that GATE shall be counted

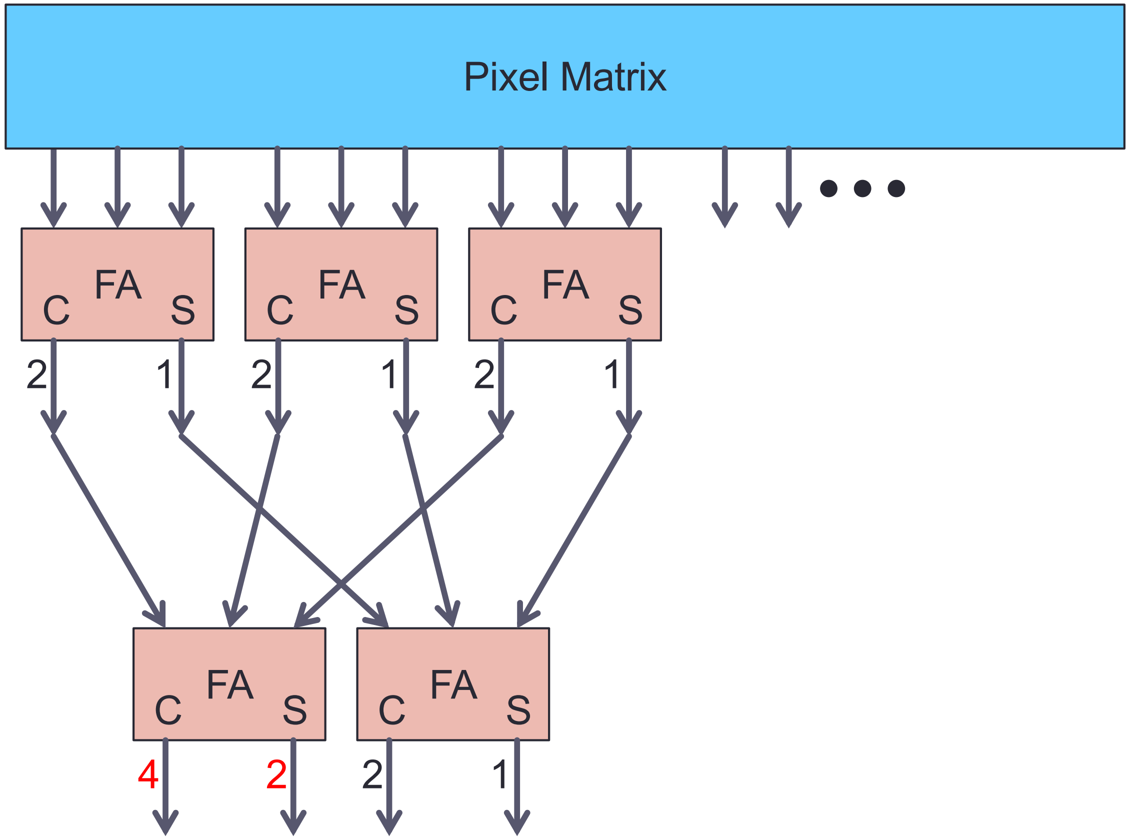
- We best set flipflops with the rising edge signals (in each of the 9 x 9 channels)
- We then ‘count’ the number of set flipflops by adding them up.

- Operation should be pipelined, i.e. we can send the next GATE while we add up the previous pattern.

- Chip output is a binary (7 bit) number which occurs with some latency after each GATE

Adder

- One solution: Add up bits using a tree of full adders



Further Blocks

- We have a PCELL for the SPADs
    - Gap between active areas is ~7um

- We can use digital cells from the IMS library, or design our own!

- I suggest to have no buffers in the input cells, so that we do not have to bother about drive strength (but maybe about RC effects!)

- We need a fast output pad to drive the count in a load of, say, 5 pF, in, say, 10 ns.
    - Make sure that pad has no short circuit current in the push-pull driver
    - May use a lower high level to gain speed.

Initial work program

- Think about input circuitry per pixel
- Get acquainted to full adder
- Try out a small adder tree, get a feeling for speed.
  - Use realistic fan-out / fan-in and an estimate for wiring caps.
- Make an overall chip hierarchy
- Count pads