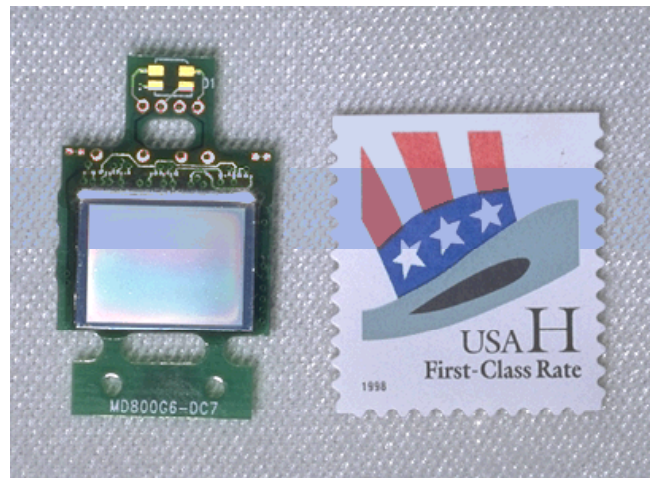
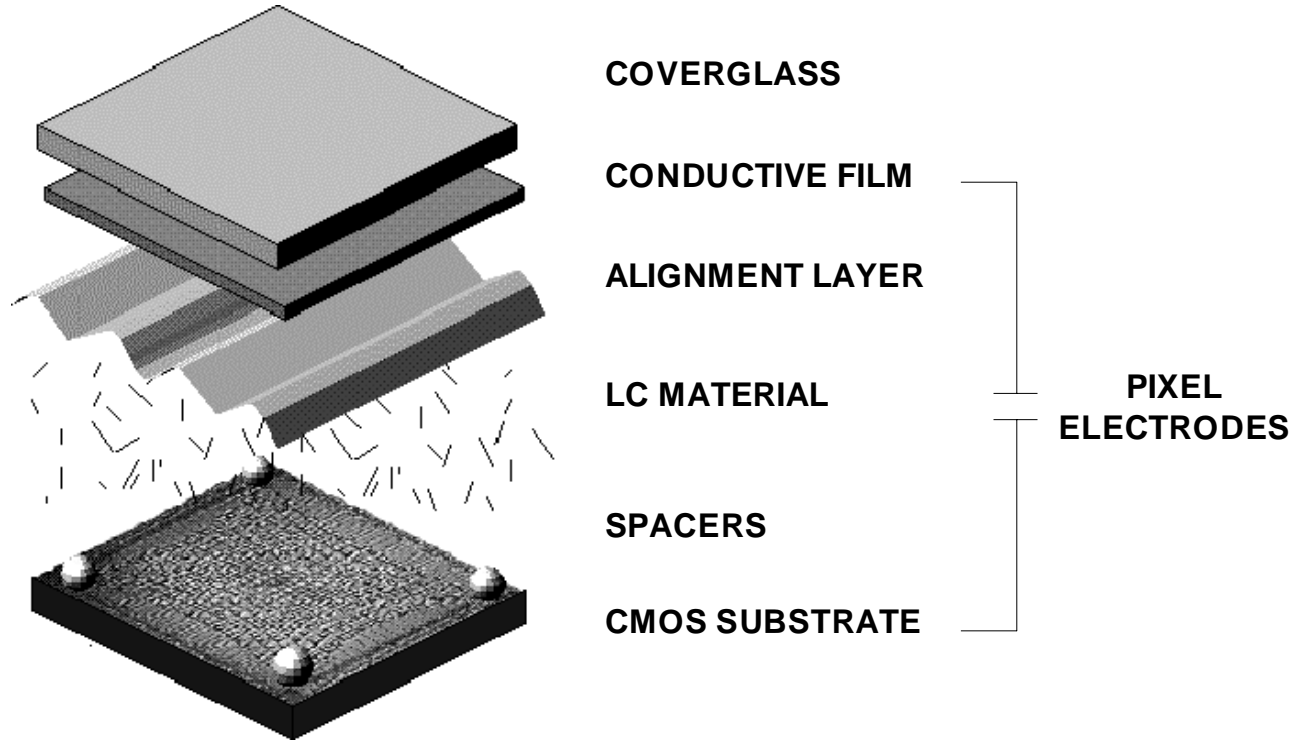


**A Field-Sequential-Color
1024 _ 768 Liquid-Crystal-on-
Silicon Display**

Michael Bolotski and Jean-Jacques Drolet

The Micro Display Corporation
San Pablo, CA 94806
drolet@microdisplay.com

What is a microdisplay?





Technology background

Application areas:

Telecommunications
Body-worn
Projection

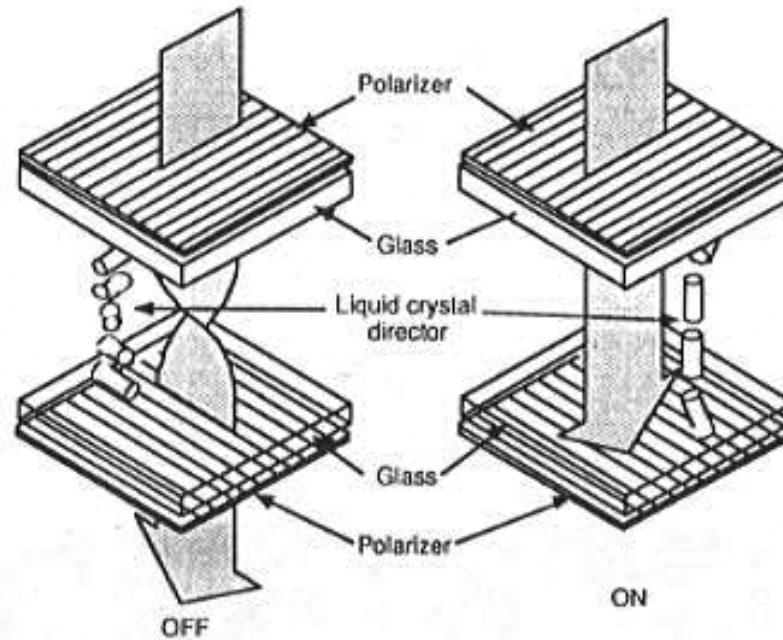
Microelectronics:

Use mature CMOS technology
Integration
Rapid development cycle

Manufacturing:

Foundry production of CMOS wafers
In-house LC assembly and packaging

Twisted-nematic liquid crystal cell



No voltage

- Helix
- Polarization modified
- Bright state

High voltage

- LC molecules vertical
- Polarization unaffected
- Dark state



Electro-optic response

Twisted Nematic LC

Normally white

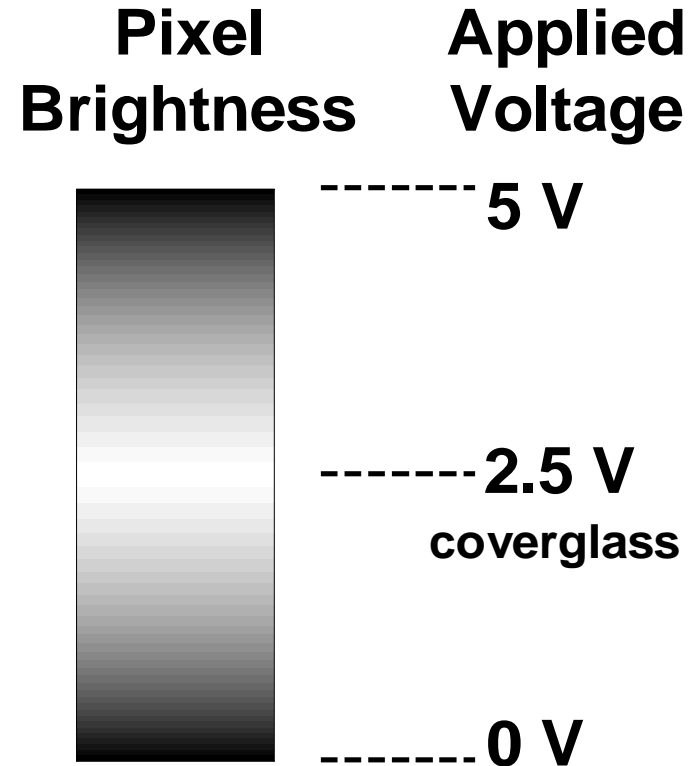
Darker with greater electric field
(+ or -)

Requires AC drive

Prevent ionic impurities from
collecting on electrode

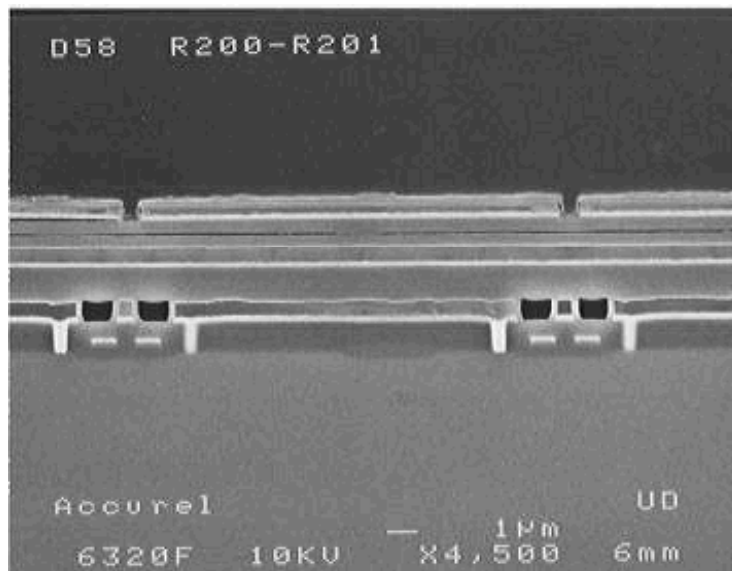
Method: Inversion

Drive pixel positive then
negative



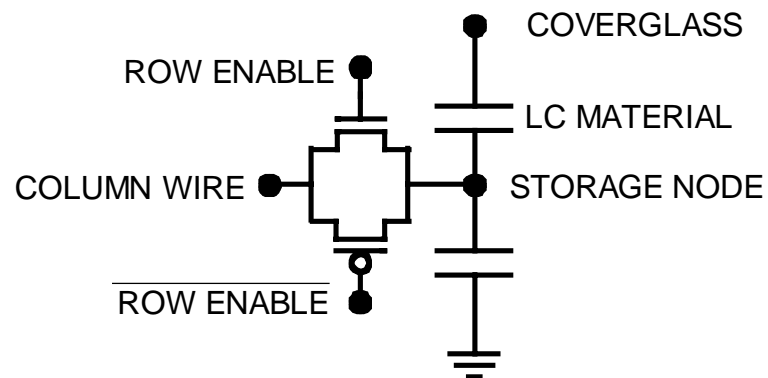
CMOS Process Enhancements

- Surface reflectivity enhancement
- Flatness
- Fill-factor enhancement
- Internal light capture



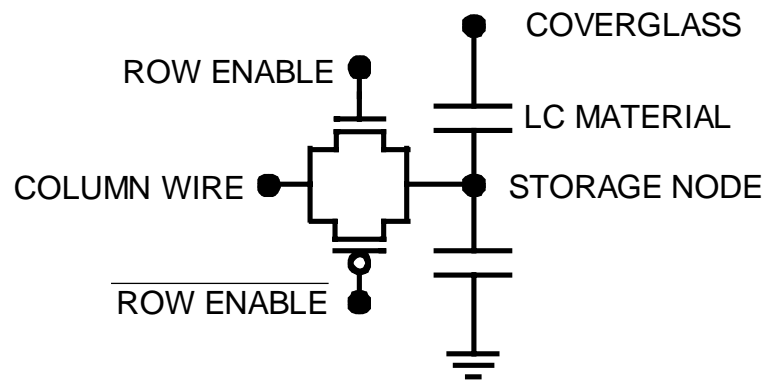
Pixel: circuit-design goals

- **Sample-and-hold**
- **Sample rail-to-rail voltages: CMOS T-gate**
- **Minimize droop rate: high capacitance (∇V)**
 - source/drain junctions
 - inter-metal capacitors
 - liquid crystal
- **Minimize clock feedthrough: CMOS T-gate**

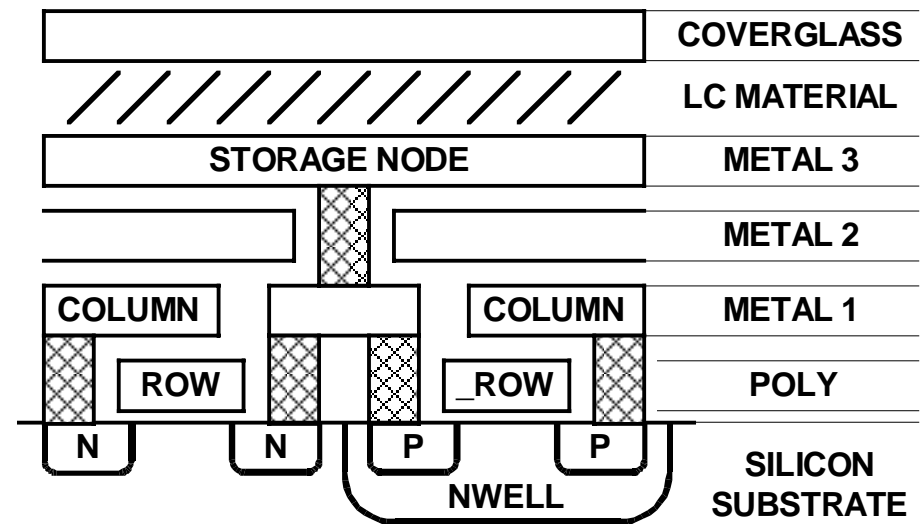


Pixel structure

Schematic



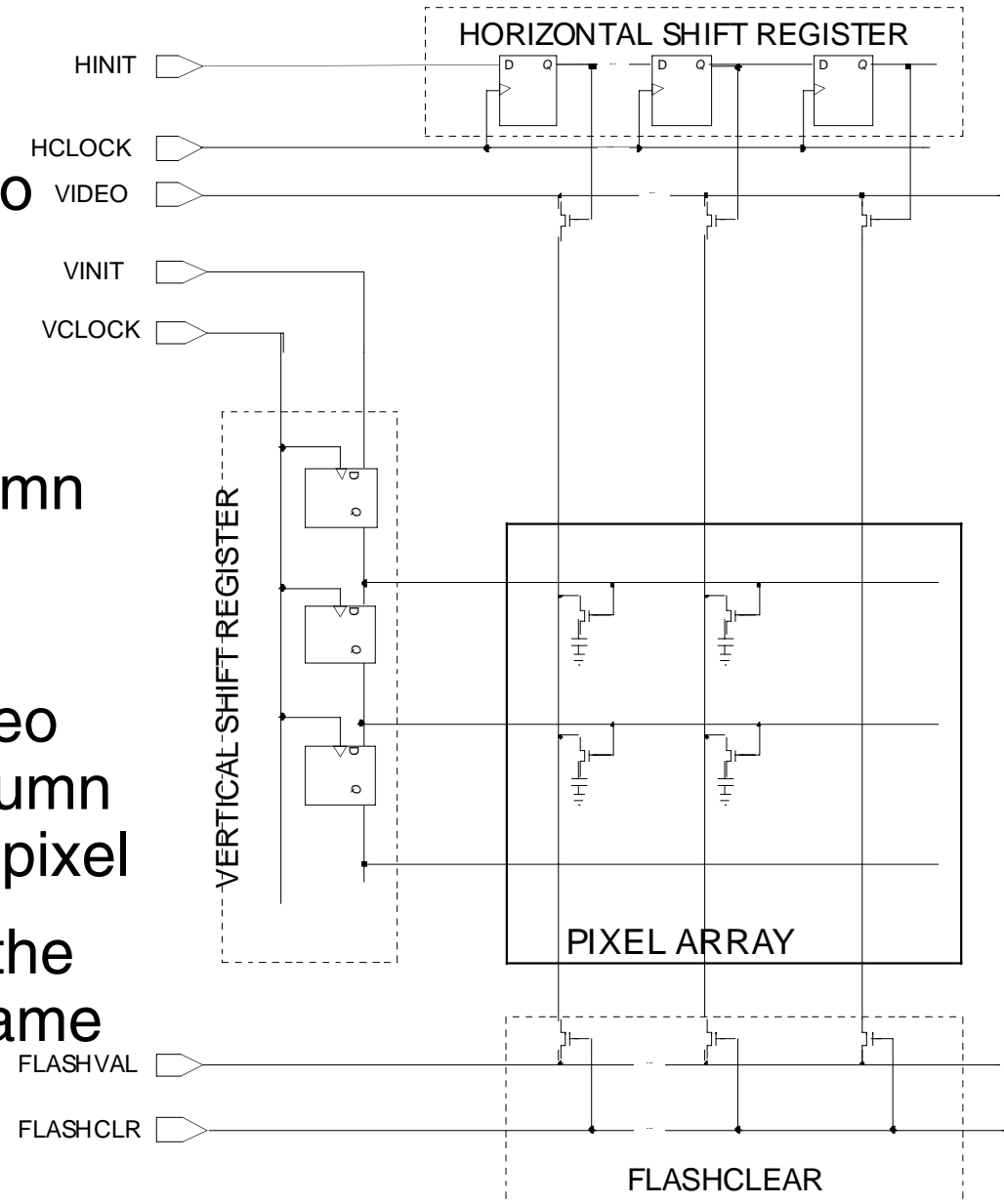
Cross Section



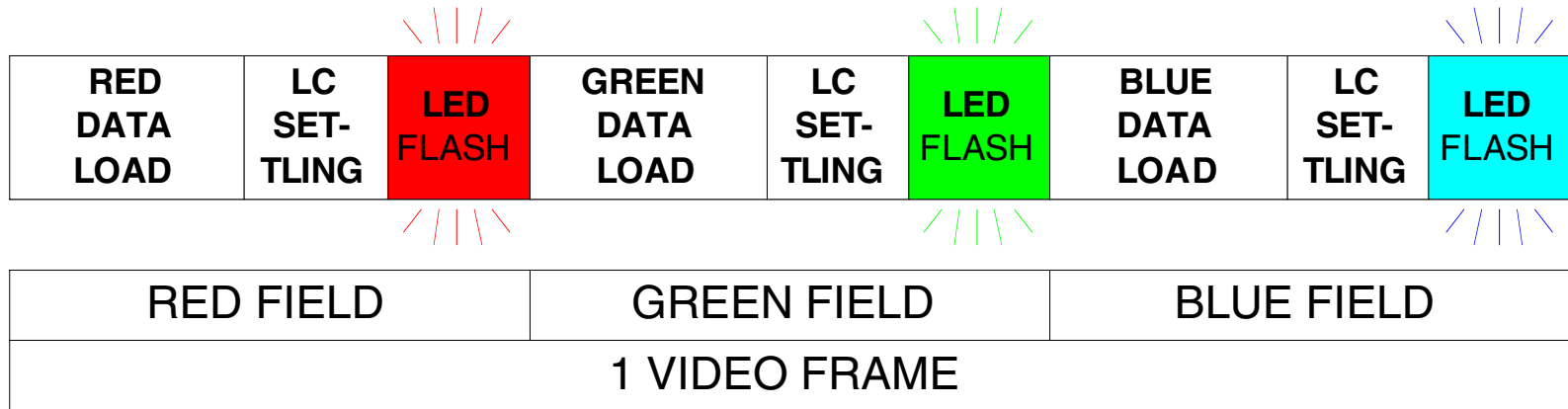
Pass-gate connects storage node to column wire

MicroMonitor architecture

- Column wires
 - Connection to video wire controlled by horizontal register
- Pixels
 - Connection to column wire controlled by vertical register
- Analog value on video wire sampled by column and currently active pixel
- Flashclear charges the entire array to the same voltage

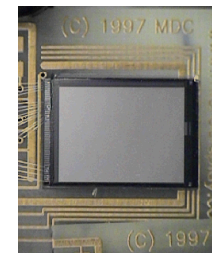
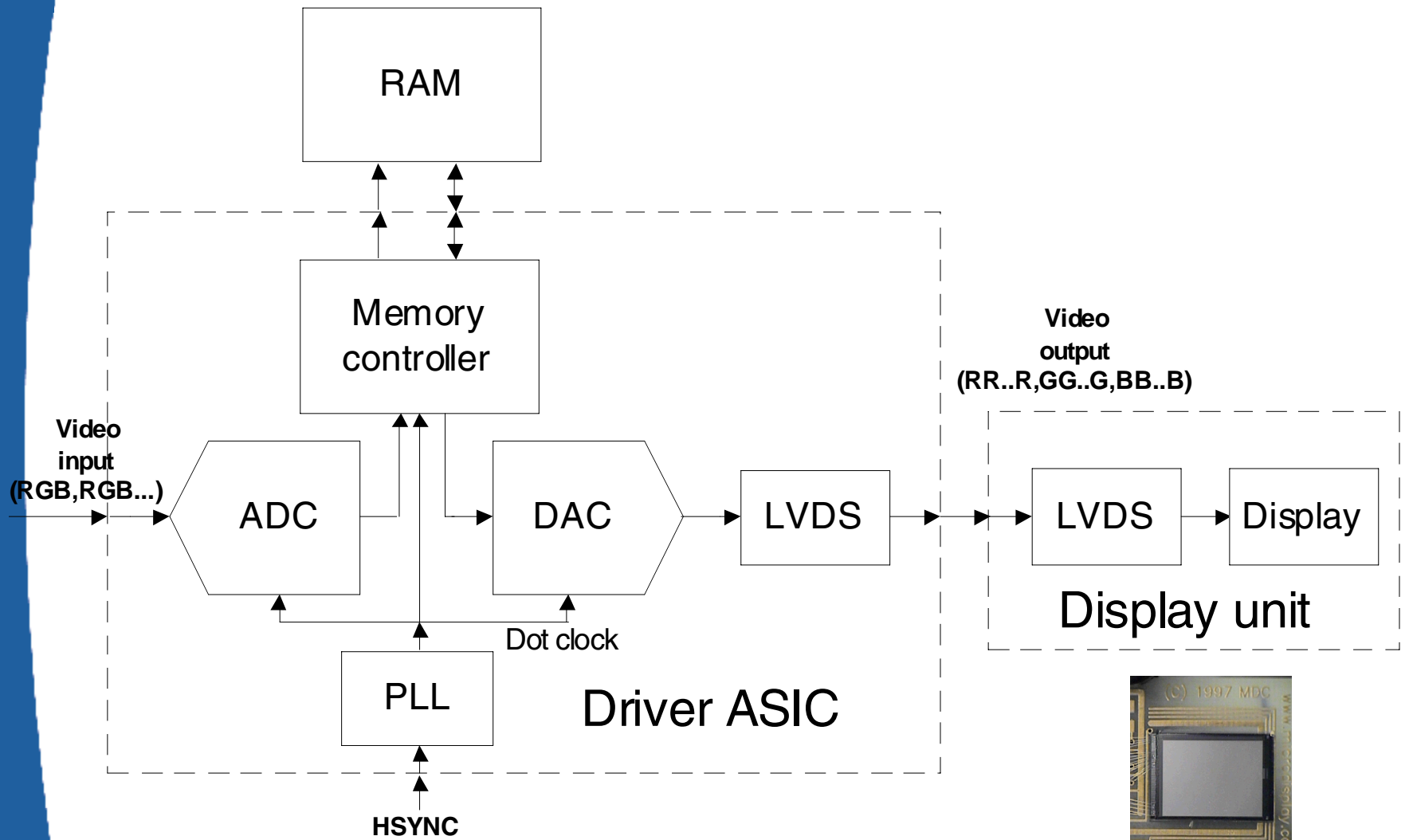


Field-sequential color



- Sequential display of red, green and blue fields
- Display previous frame's data while new data is buffered \Rightarrow memory
- Need to load pixel array quickly (max. LED flash time, LC settling time)

System architecture





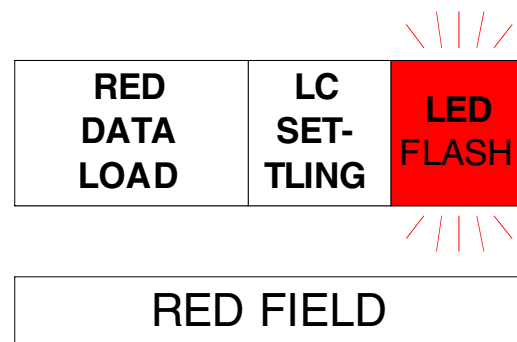
Timing requirements

- **Exceed fusion frequency, avoid flicker**
 - 72 Hz frame rate
 - 216 Hz field rate (4.6 ms)
- **Need ample time for LC settling, LED flash**
- **Only ~2.3 ms left for loading 786,432 pixels**
- **3 ns/ pixel for addressing + settling to within 7 mV**
- **12 ns/ pixel with 4X parallelism**

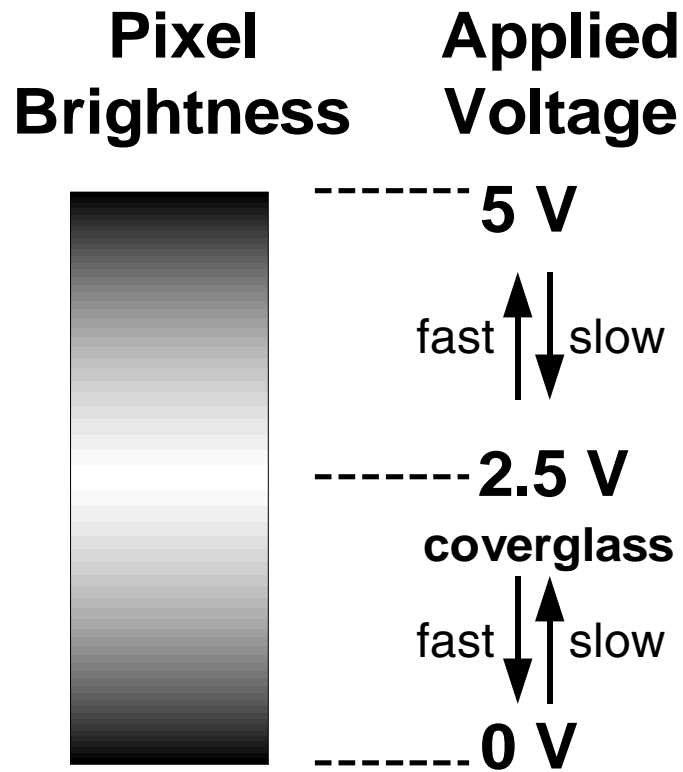
Flashclear: motivations

Speeding LC transition advantageous in many ways

- **More complete LC switching improves brightness and contrast**
- **Higher illumination duty cycle improves brightness**
- **Relaxes load-time requirement**



Flashclear: principle of operation



- Start some LC transitions before data is loaded on pixels
- Give slow LC transition a head start
- Simultaneously precharge all pixels to same voltage



Rapid Prototyping Design Cycle

Global Wires

Pixel Designs

Manual LeafCell Editing

DisCo compilation

Verification



Global Wires 1: Analysis

MathCAD/ Maple symbolic analysis

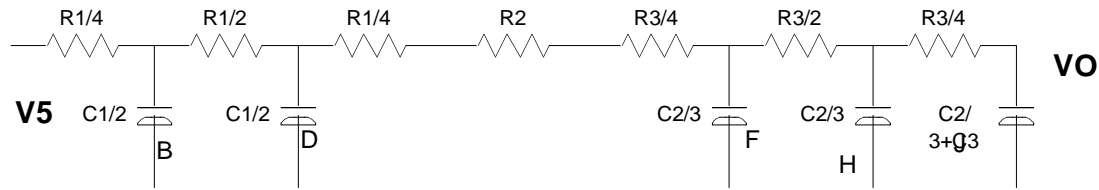
Detailed parasitic equations

Laplace equations for distributed wires

Symbolic solutions

Parameter Optimization (Wire R, Wire C, switch size)

Global Wires 2: Example



Specify:

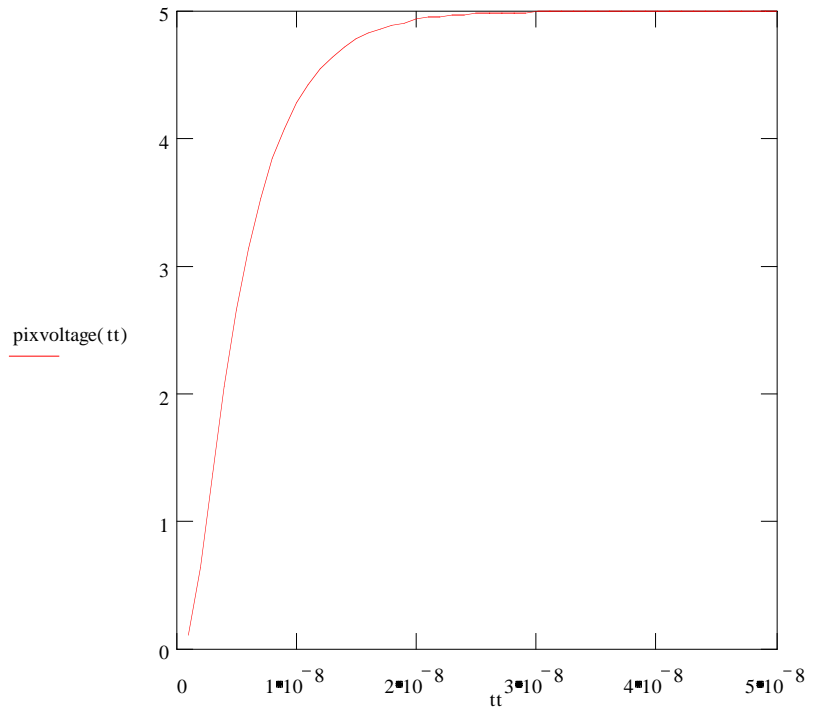
resolution = 640 x 480
 pixel size = 17.1 microns
 video wire width = 10.2 microns
 video wire layout = metal 1 over substrate
 LCD voltage swing = 1.8 V
 etc:

Get:

Pixel charging time (display speed)
 Power dissipation

SettleTime · sec = 28.206 ns

VideoPowerDiss = 5.09 mW





Global Wires 3: Detailed Simulation

Hspice transistor level simulation of distributed network

Improve simulation accuracy to within 3% compared to quick lumped approximation.



Pixel Issues

Photocurrent leakage (reverse bias into substrate)

Leakage (pass gate)

Coupling to column wire

Coupling to neighbor pixel

Storage capacitance maximization

Yield optimization



Pixel Design Process

Fully manual design

Non-aggressive design rules

Detailed parasitic extraction

Hspice simulation of coupling issues

Fabricated test structures



Leaf Cells

All cells peripheral to pixel array

Example: vertical/horizontal shift registers, guard rings, column switches

Most cells must be pitch-matched to pixel size

Edits required for functionality or process migration, not for resolution or pixel size changes.



Manual LeafCell Editing

Hand-tuned cells, full custom layout

Designed for abutment, non-overlapping

Feedthrough methodology

Relatively few complex cells



Automatic LeafCell Pitch-Matching

Base cell designed to minimum pitch

Right side designed to be extended to match pixel pitch

Automatic stretching by Display Compiler

Stretched cell is NOT yield optimized (but is tiny fraction of total area)



Now What?

Have pixel layout

Have global wire parameters

Modified leaf-cells to match pixel pitch, wire sizes

Now: painstaking task of assembling a chip?



Display Compiler

Constrained geometry specification

“place block A here, aligned with top of block B”

Abstracts leaf cells as bounding boxes and I/O pins

Built on Scheme programming language, allowing embedded code fragments



Display Compiler example

```
(build-a-cell "chip"  
  `((dc "display_core"           :at (0 0))  
    (pwc "pad_wiring_channel"    :anchor dc      :where left   :flush bot)  
    (te "top_edge"               :anchor pwc   :where above :flush left)  
    (be "bottom_edge"           :anchor pwc   :where below :flush left)  
    (re "right_edge"            :anchor te    :where right :flush top)  
    (ps "pad_stack"             :anchor pwc   :where left  :flush bot  
      :offset (0 ,(- *bot_edge:m2_width*))  
    )  
  )  
)
```



DisCo:Vertical Shift Register

```
(define vertical_shift (build-a-cell "vertical_shift"
  ;; instance list
  `((vb "vshift_bot" :at (0 0))
    (vs "vertical_slice" :anchor vb :where above :flush right :array-y ,*pixels-y* )
    (vt "vshift_top" :anchor vs :where above :flush right))

  ;; shape list
  (append`
    ;; the following two are one on top of other along left side of vshift unit
    ("metal2" ,*vshift:powerstrip* ,*MAX_DIMEN* :name vddstrip
      :anchor vs :where left :flush bot :clips ((vt top 0)))
    ("metal1" ,*vshift:powerstrip* ,*MAX_DIMEN* :name gndstrip
      :anchor vddstrip :where left-in :flush bot :clips ((vddstrip all 0)))
    ;; LC fill channel
    ("glass" ,*MAX_DIMEN* ,*LC_fill_channel* :at (0 ,*vshift:LC_channel_yoffset* )
      :clips ((frame_left right ,(- 0 *fill_channel_area_sep* *frame_less_sep* )))))

  ;;propagate up pins
  (propagate-pin "ClockBuf" "metal2" 'vt)
  (propagate-pin "gnd!" "metal2" 'vt)
  (propagate-pin "vdd!" "metal2" 'vt)
  ))
```

Vertical Resolution

LCFill Hole Size

Power Distribution Width



DisCo Features

Parameterized to pixel size, resolution

Local geometric relationships usually do not change

Considerable design re-use

One chip usually described by 20-30 pages

Disco description serves as documentation and design log

One happy family

4 custom designs in one reticle

- Resolutions: QVGA, SVGA, XGA
- Pixel Sizes: 5x15, 10x10, 12.5x12.5 microns
- Diagonals: .3", .5", .7"
- Pixel Types: Reflective and Diffractive
- Interfaces: 4-bank analog FSC, 3-bank spatial
- Chip Spec: August 24, 1998
- Operational: November 11, 1998
- 4 designs verified in 2.5 months

