CS 535 Computer Graphics

Fuhua (Frank) Cheng

Department of Computer Science
College of Engineering
University of Kentucky

Rm 303, Davis Marksbury Bldg
329 Rose Street
Lexington, KY 40506-0633

"Graphics and artificial intelligence are inseparable, graphics needs AI, and AI needs graphics." - Jensen Huang

So you are in the right place if you are also interested in AI.

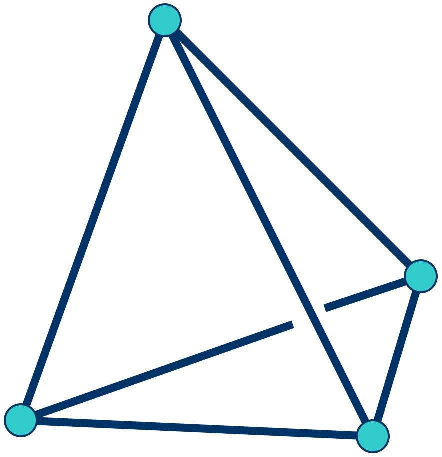
1. Introduction

1.1 Graphics Areas

- Modeling: building *specification of shape* and *appearance properties* that can be stored in computer
- Rendering: *creation of shaded images* from 3D computer models
- Animation: to create an *illusion of motion* through sequences of images

Modeling

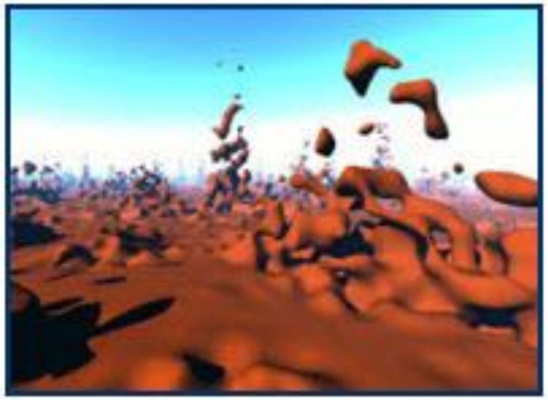
For example, how should the pyramid on the right be represented internally?



You need to record both geometric and topological Information:

Vertex Table + Edge Table

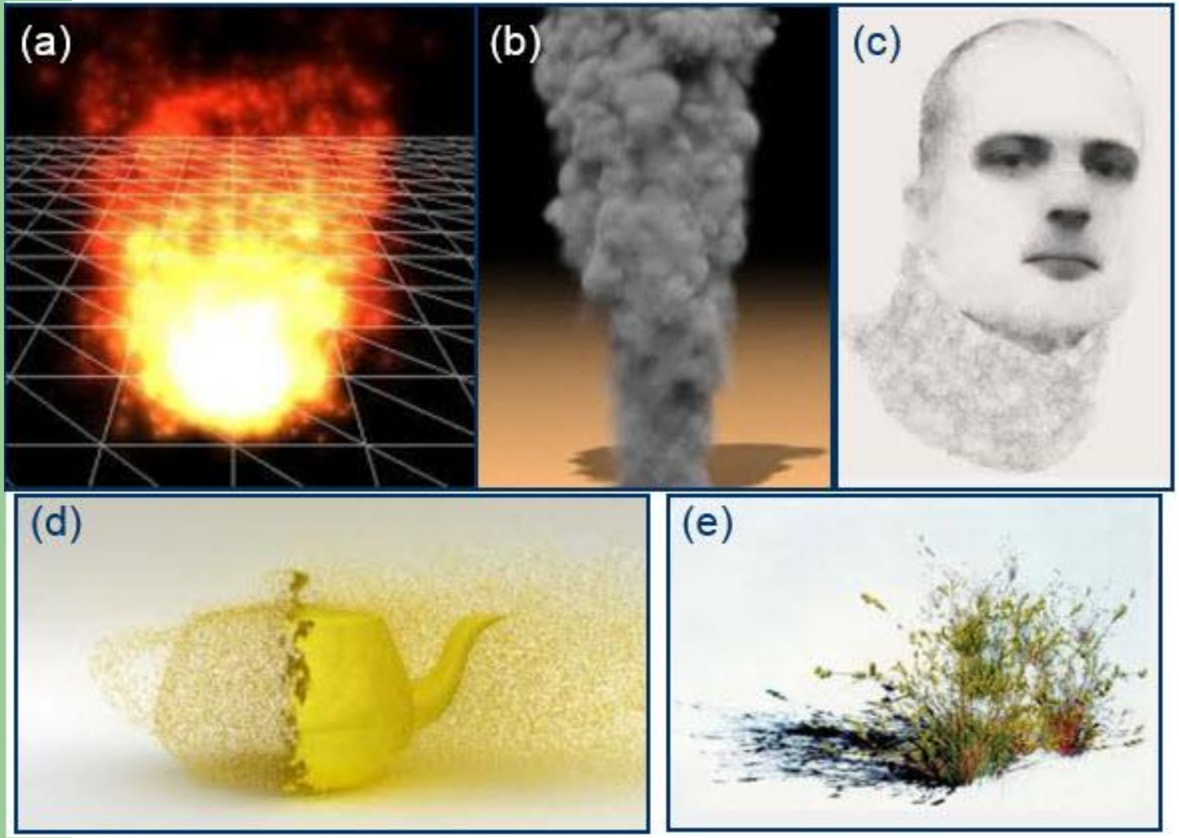
Modeling



or, these?

shape design/representation using: implicit surfaces, parametric surfaces, subdivision surfaces, …

Modeling



or, these?

particle system

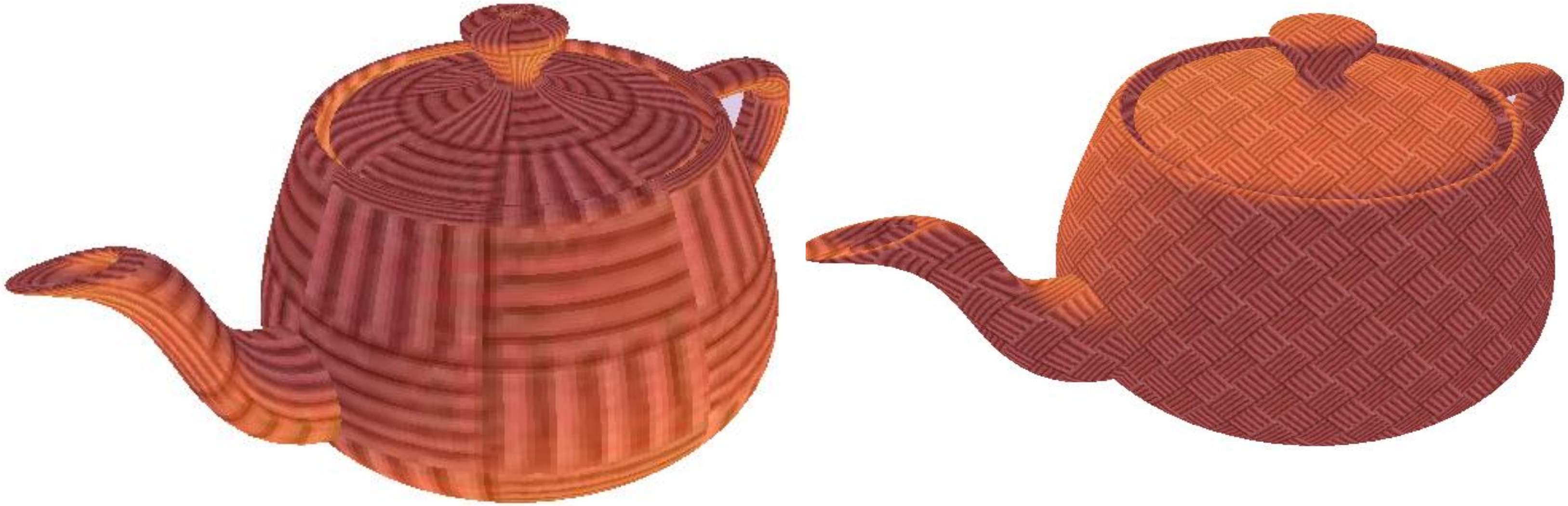
Modeling

Or, these?

Context free grammar (fractals, L-system)

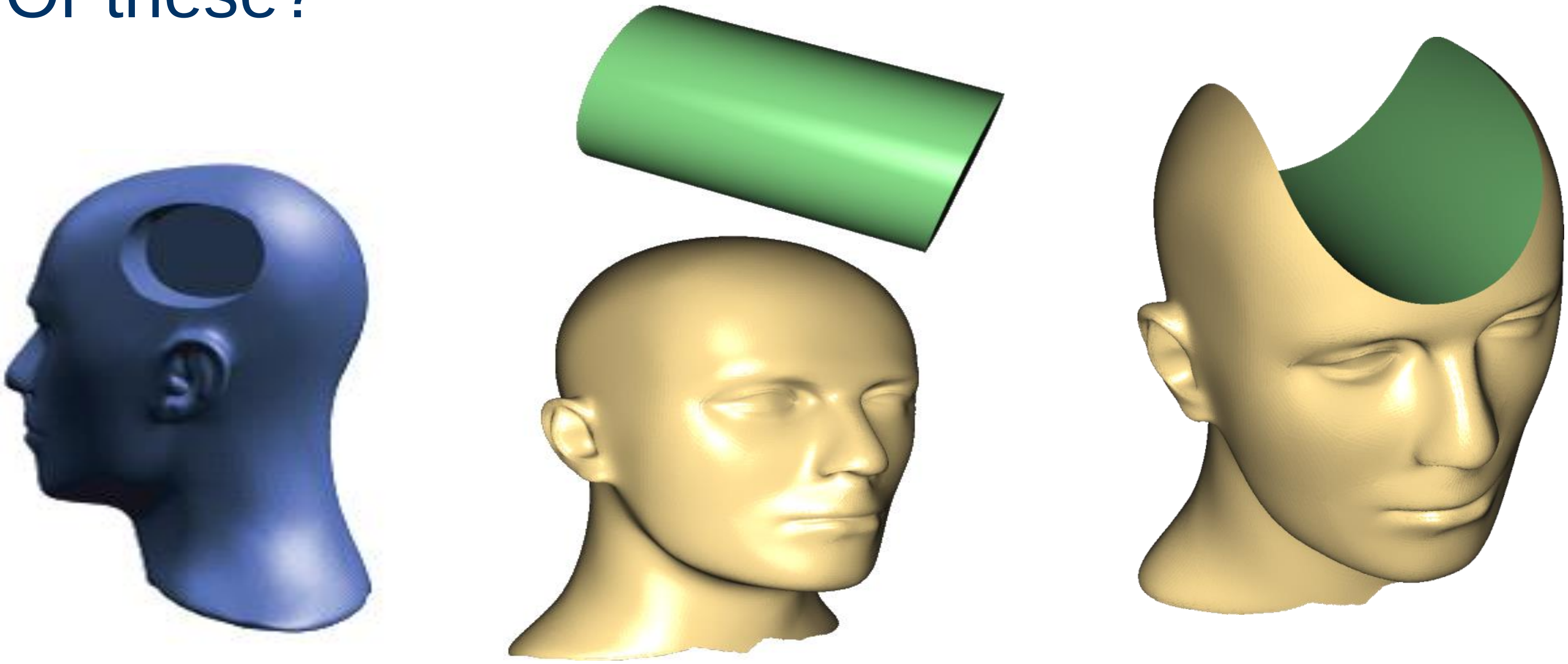
Rendering

How should images like these be generated?



Rendering

Or these?



Animation

How should an illusion of motion be generated?

Animation

Animation

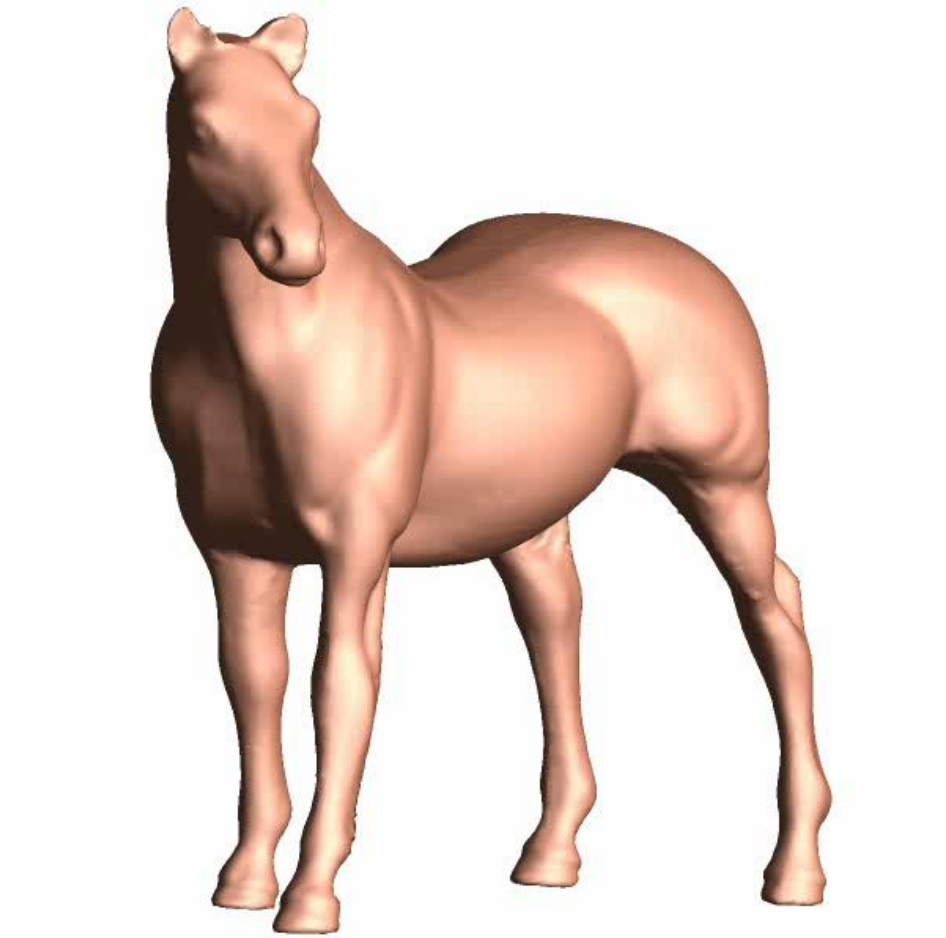
Dart shooting

One should keep a low profile, never show off.

Animation

- Fat horse animation

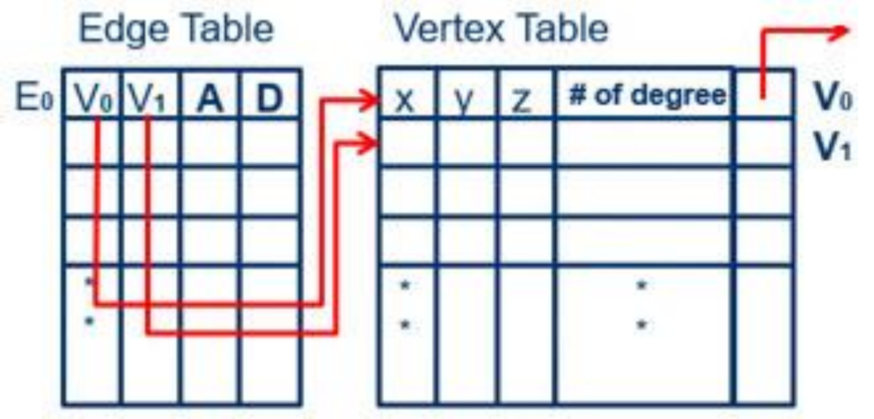
mplayerc.exe



Advantages

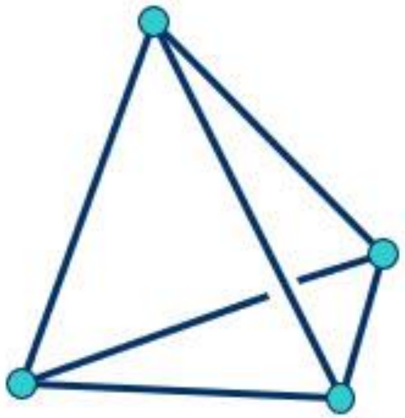
- **Quantitative description**

  - precise, not easy to be recognized



- **Pictorial description**

  - easy to be recognized (a picture is worth a thousand words)



History

Founded by the PhD thesis of Ivan D. Sutherland at MIT in 1963,

- A **line drawing system** with **data structures** for storing symbol hierarchies and **interaction techniques**

SIGGRAPH: important CG organization, formed in 1969

Website: *http://www.siggraph.org*

Computer Graphics, Computer Vision & Image Processing

(blending together more each year)

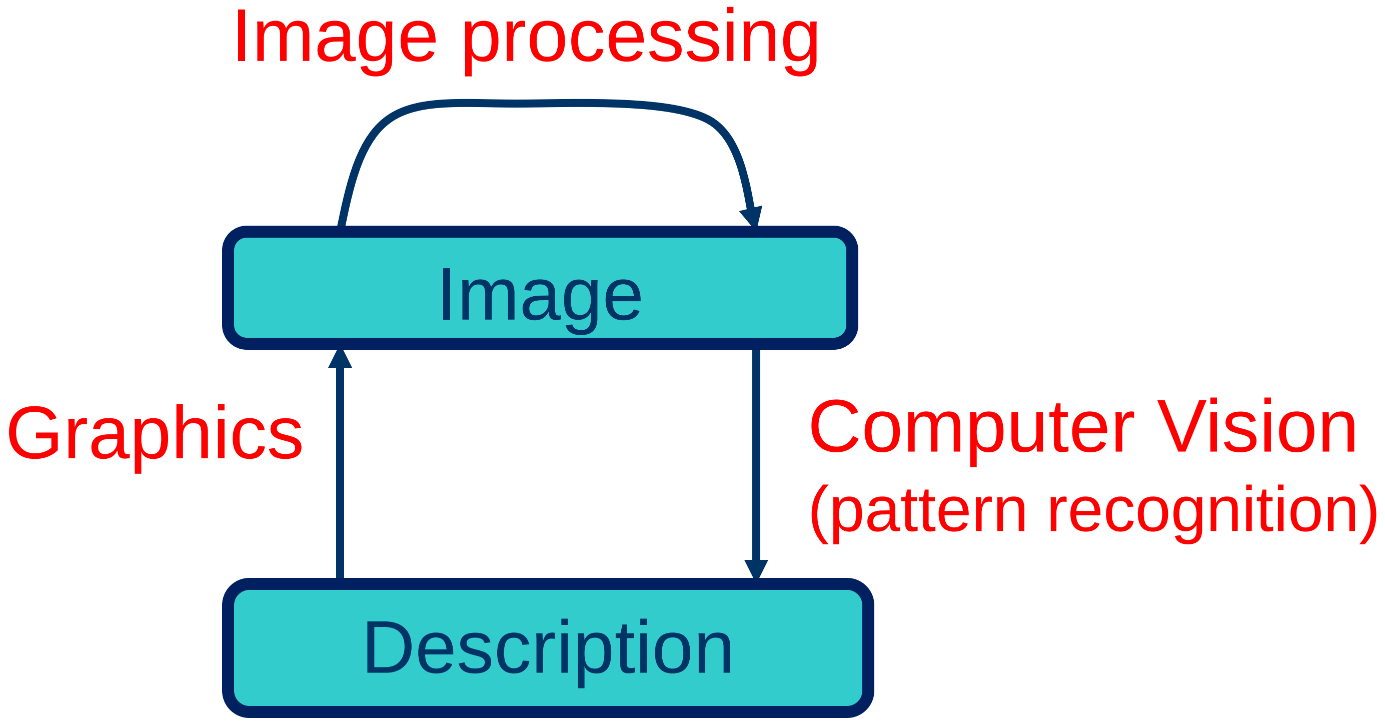


Image processing

Image

Graphics

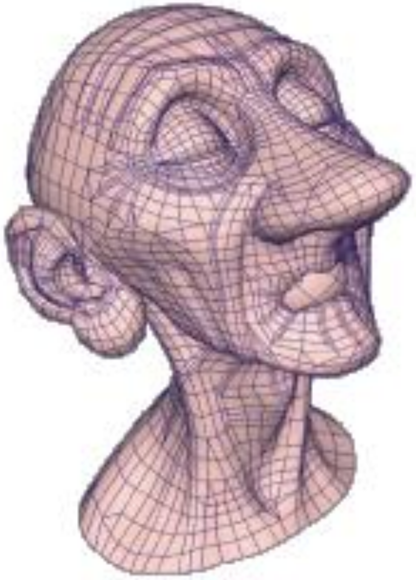
Computer Vision
(pattern recognition)

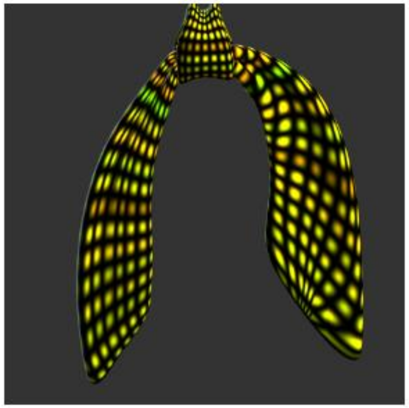
Description

1.2 Applications

- Art, Entertainment, and Publishing
  - Movie production, Animation, and Special Effects
  - Computer Games
  - Browsing on the World Wide Web
  - Slide, Book and Magazine Design

Examples





1.2 Applications

- **Process Control (Monitoring)**
  - **Status display** for refineries, power plants, computer networks from sensors attached to critical components
- **Simulation**

  - Flight simulation

  - Simulation of the movement of a robot

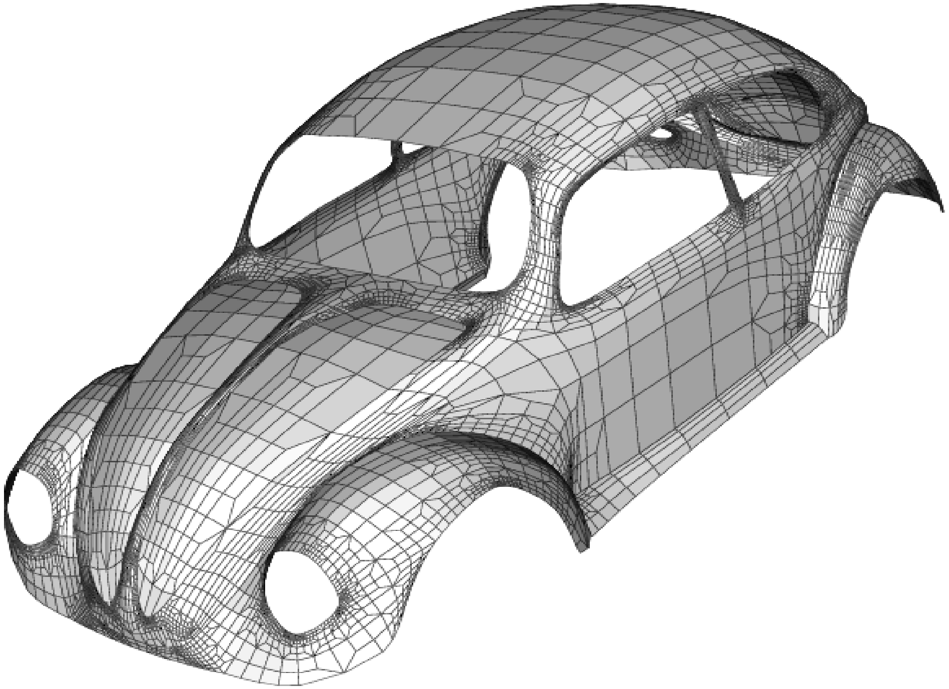
  - Simulation of 'virtual world'

Virtual World

1.2 Applications

- **Computer Aided Design (CAD)**
  - Computer Aided Mechanical Part Design (big market)
  - Computer Aided Architectural Design
  - Electrical Circuit (IC) Design (big market)

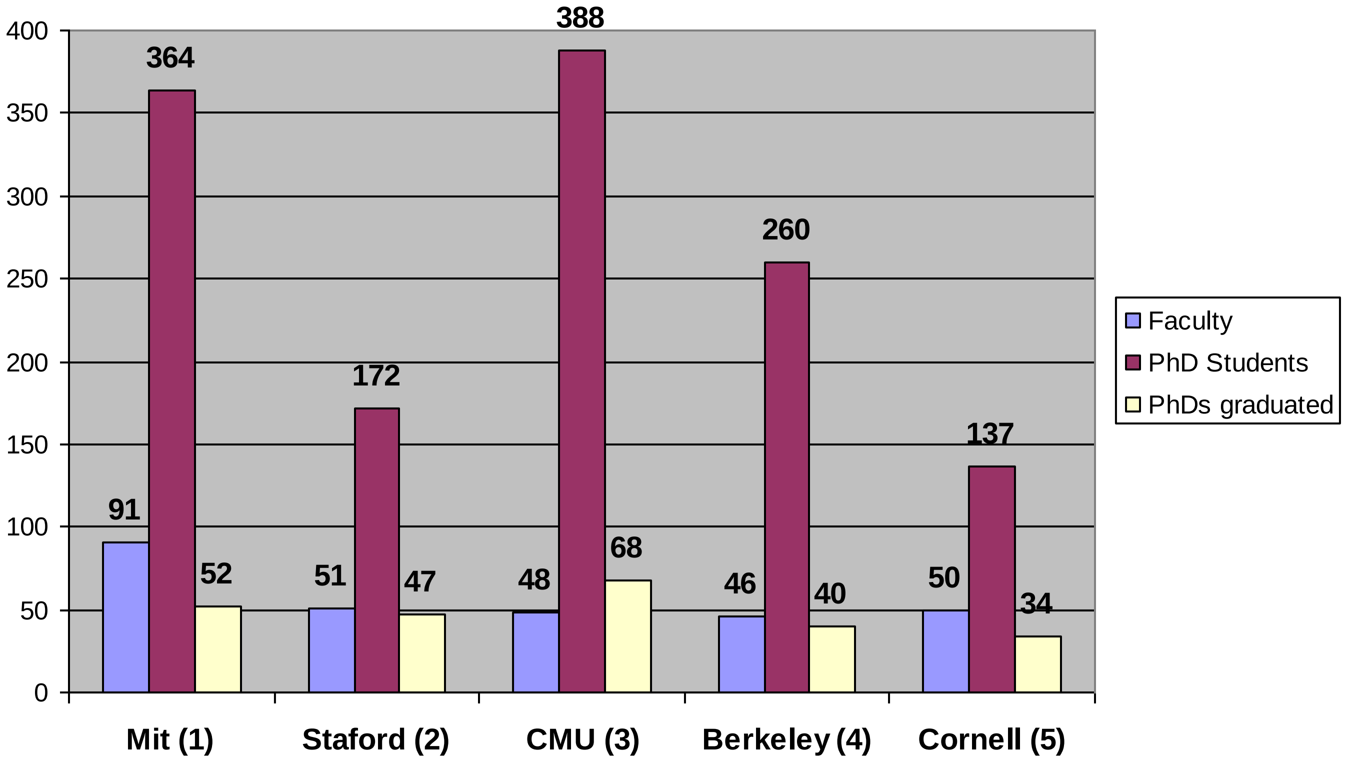
Examples of CAD



1.2 Applications

- **Scientific Analysis and Visualization**
  - Assist scientists in understanding measured data
  - Provide insight into complex mathematical ideas

Bar chart

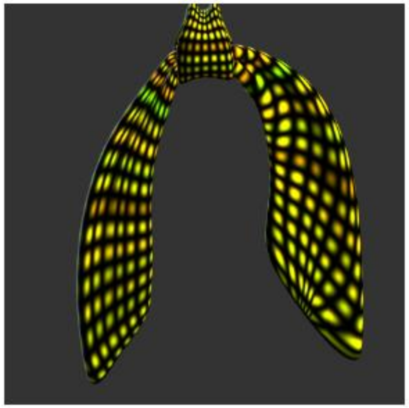

400
350
300
250
200
150
100
50
0
364
388
260
172
137
91
52
51
47
48
68
46
40
50
34
Mit (1)
Staford (2)
CMU (3)
Berkeley (4)
Cornell (5)
Faculty
PhD Students
PhDs graduated

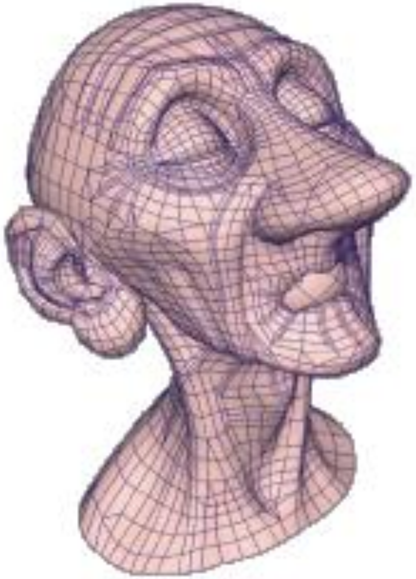
1.3 Elements of Pictures Created in Computer Graphics

- **Output Primitive**:

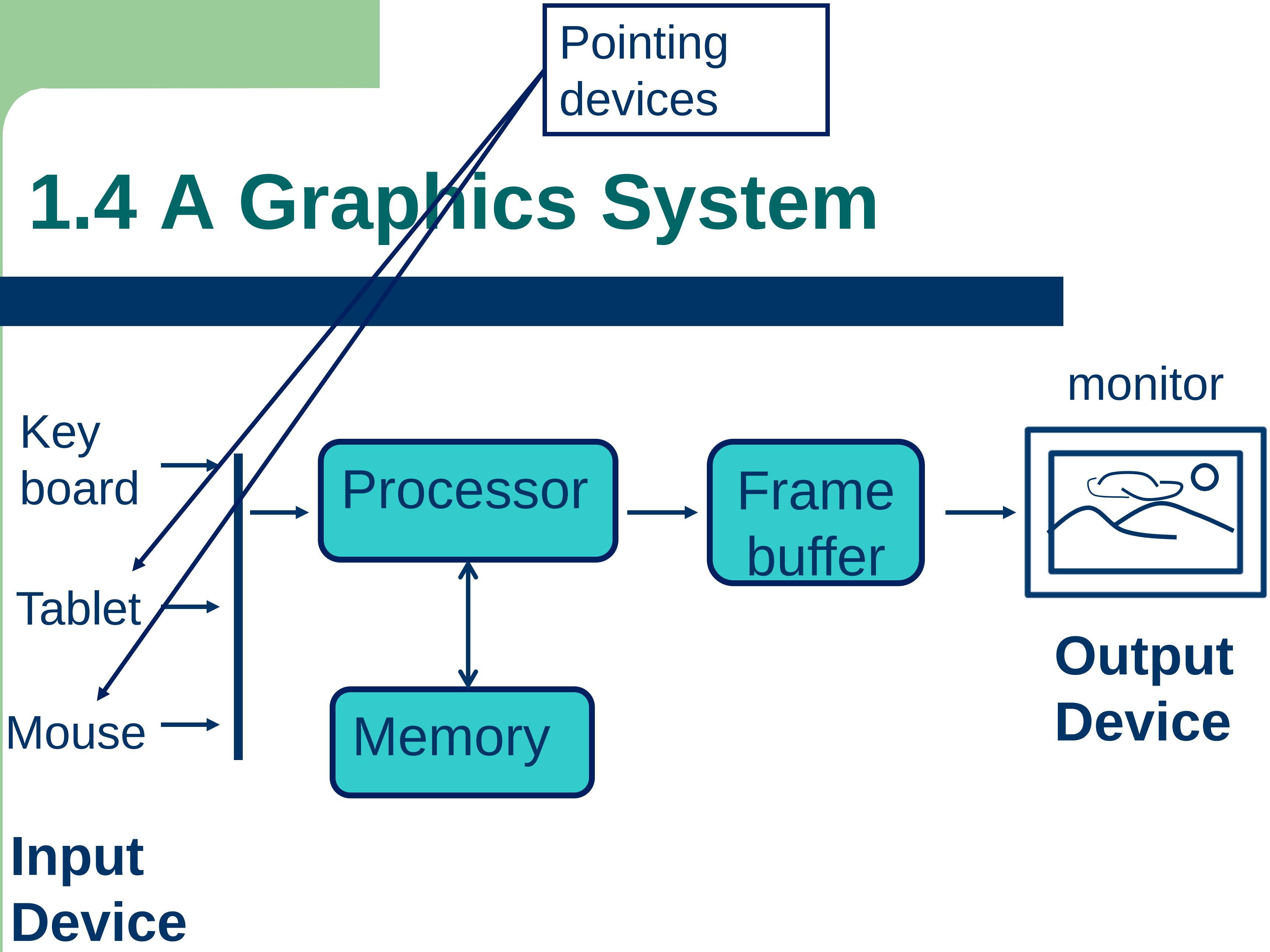
  - points
  - lines
  - triangles (filled regions)
  - text

Examples

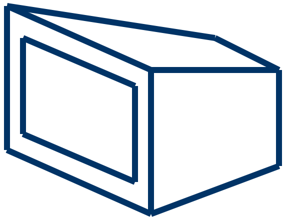




1.4 A Graphics System

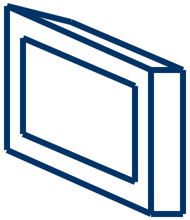


Output Devices (Video monitors)



Cathode Ray Tube (CRT)

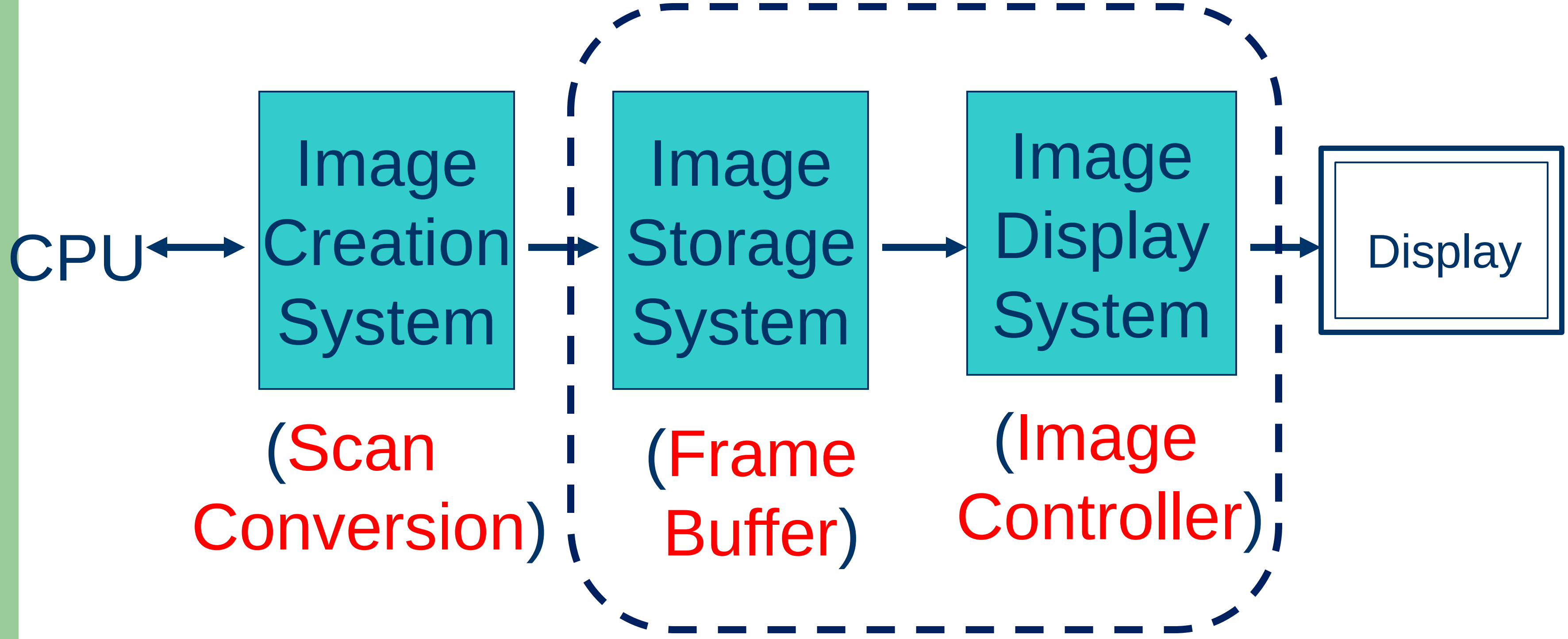
(Big and bulky, no longer used)



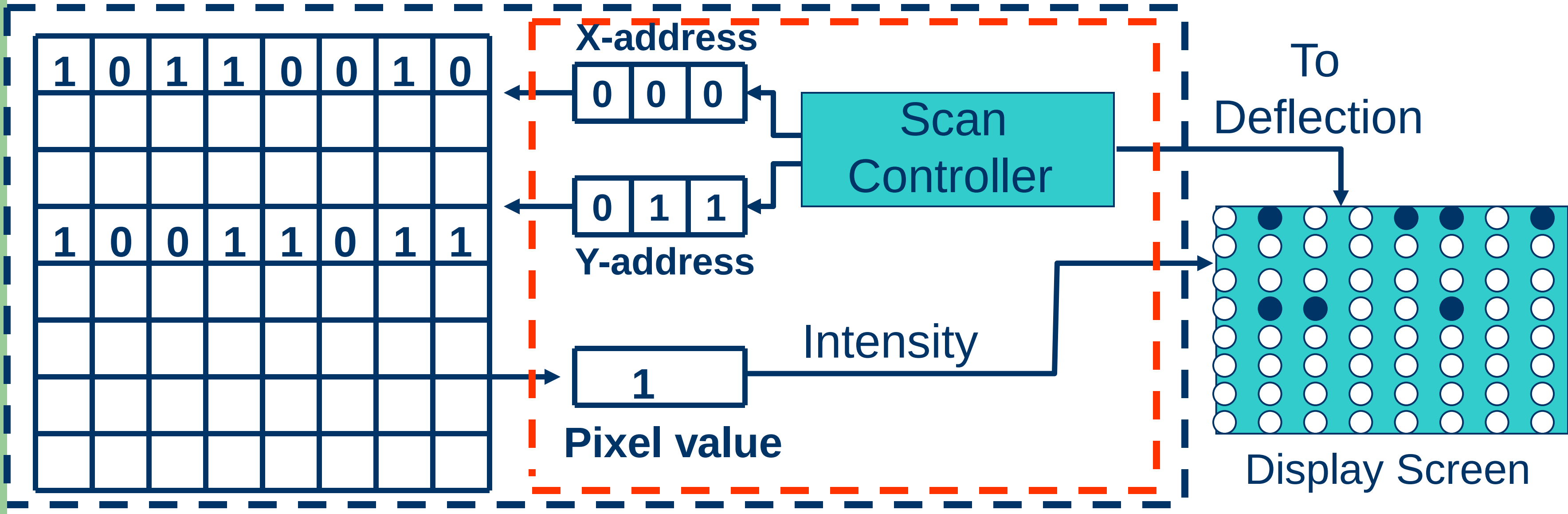
Liquid Crystal Display (LCD)

(Flat-panel display)

1.5 Display Processing Unit



Display Processing Unit (a simple two-color raster –scan system)



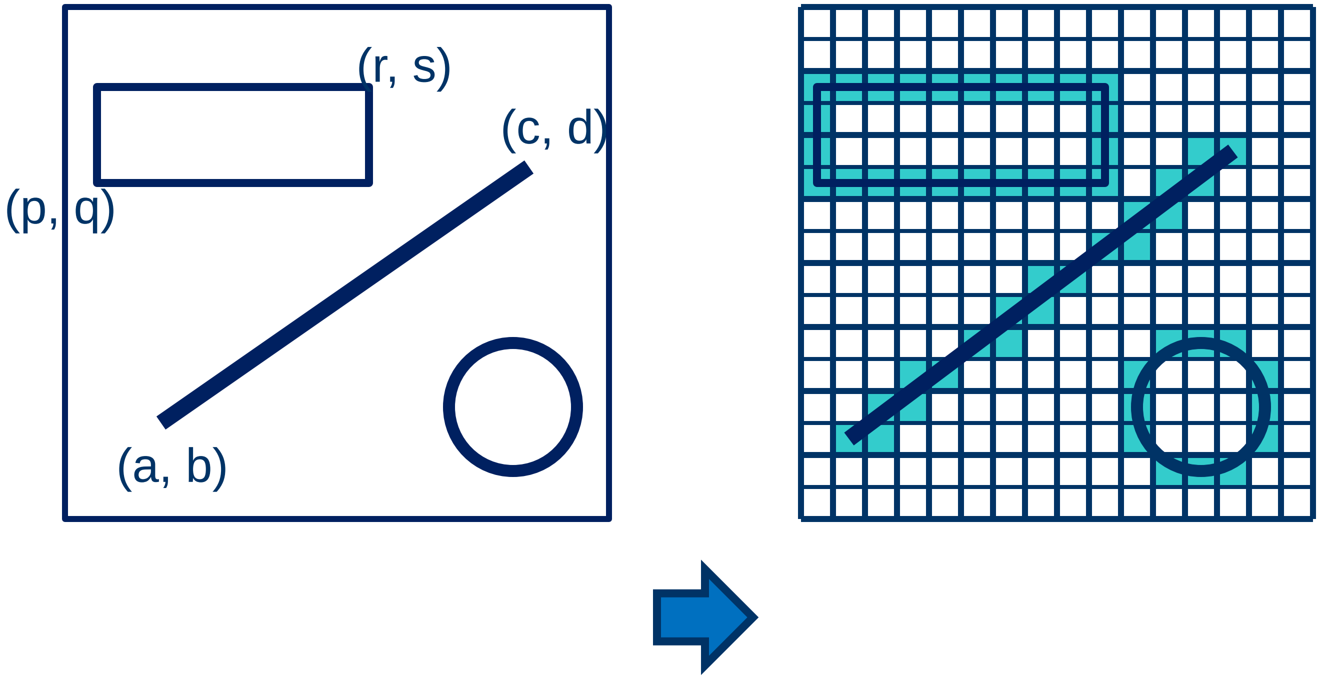
Frame Buffer

Image Controller (DPU)

1.6 Image Creation System

- Scan-converts abstract representation of an image into appropriate pixel values in the frame buffer

Scan Conversion



(r, s)

(c, d)

(p, q)

(a, b)

Frame Buffer

(Scan Conversion)

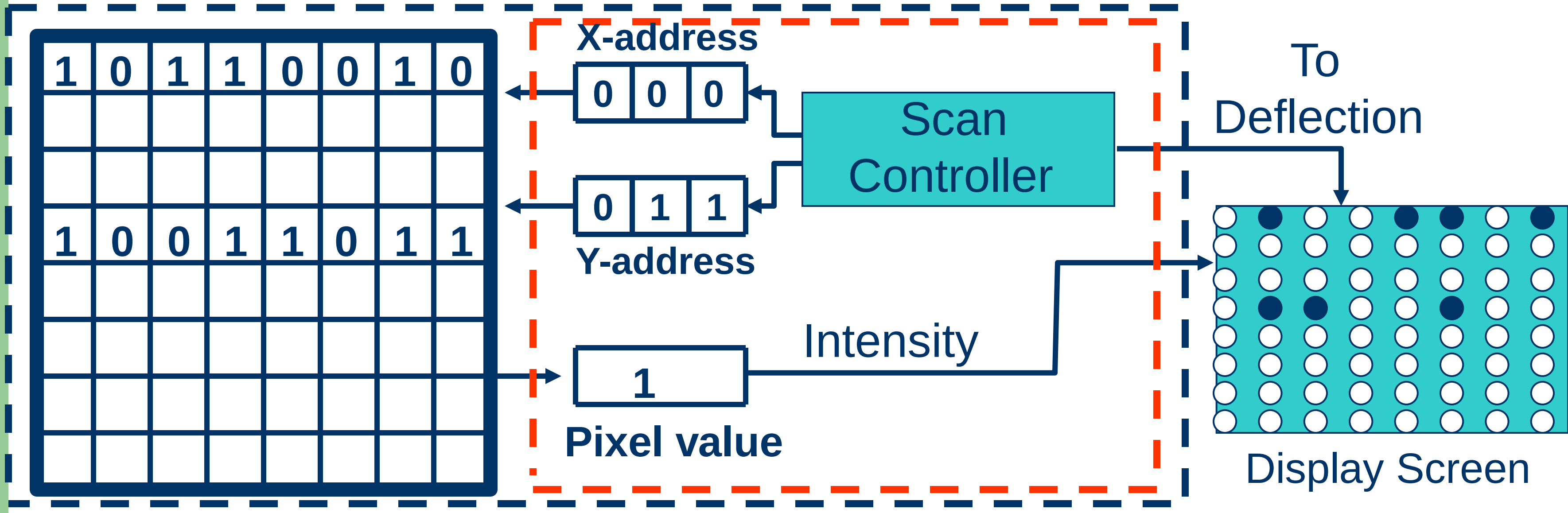
1.7 Image Storage System

(frame buffer, bitmap, color buffer)

- refresh memory arranged as a 2D array; each entry corresponds to a screen pixel

  (i.e., dimension of the frame buffer is the same as the resolution of the screen)
- each entry is composed of a number of bits; brightness and/or color value of each pixel of the screen is stored in corresponding entry in frame buffer
- implemented with solid state RAM

Image Storage System (a simple two-color raster –scan system)



1	0	1	1	0	0	1	0
1	0	0	1	1	0	1	1

Frame Buffer

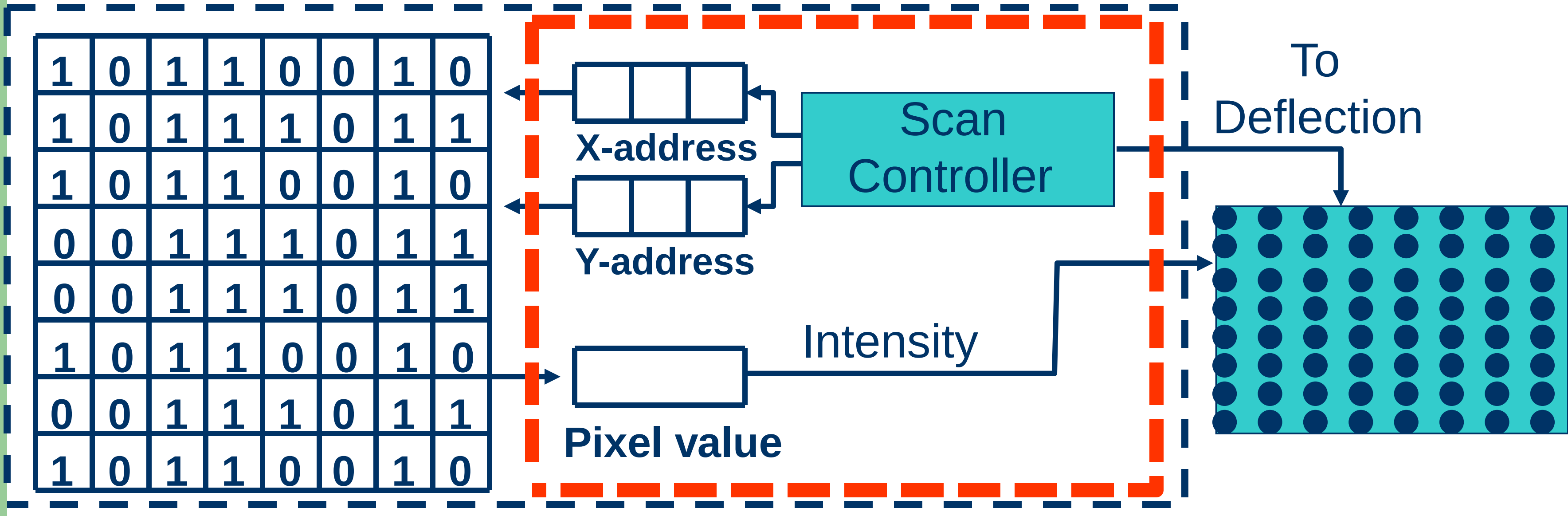
1.8 Image Display System

(video/image controller, DPU)

- cycle through frame buffer row by row,

  60 or 120 times/sec
- memory reference addresses are generated in synchronism with the raster scan; contents of the memory are used to control monitor beam's intensity
- changes in frame buffer is done during the 1.3 millisecond flyback (or, vertical retrace) time
- interlaced raster scan (to produce a picture whose effective refresh rate is closer to 120 than to 60 Hz).

Image Display System (a simple two-color raster –scan system)

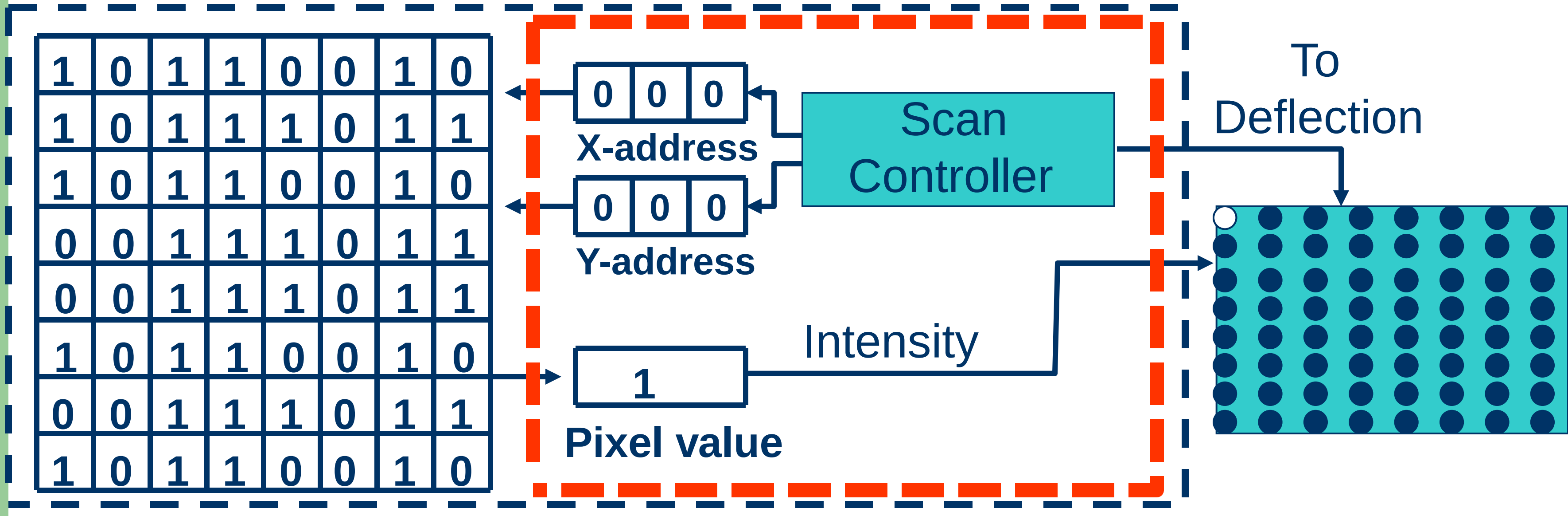


1	0	1	1	0	0	1	0
1	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0

Frame Buffer

Image Controller

Image Display System (a simple two-color raster –scan system)



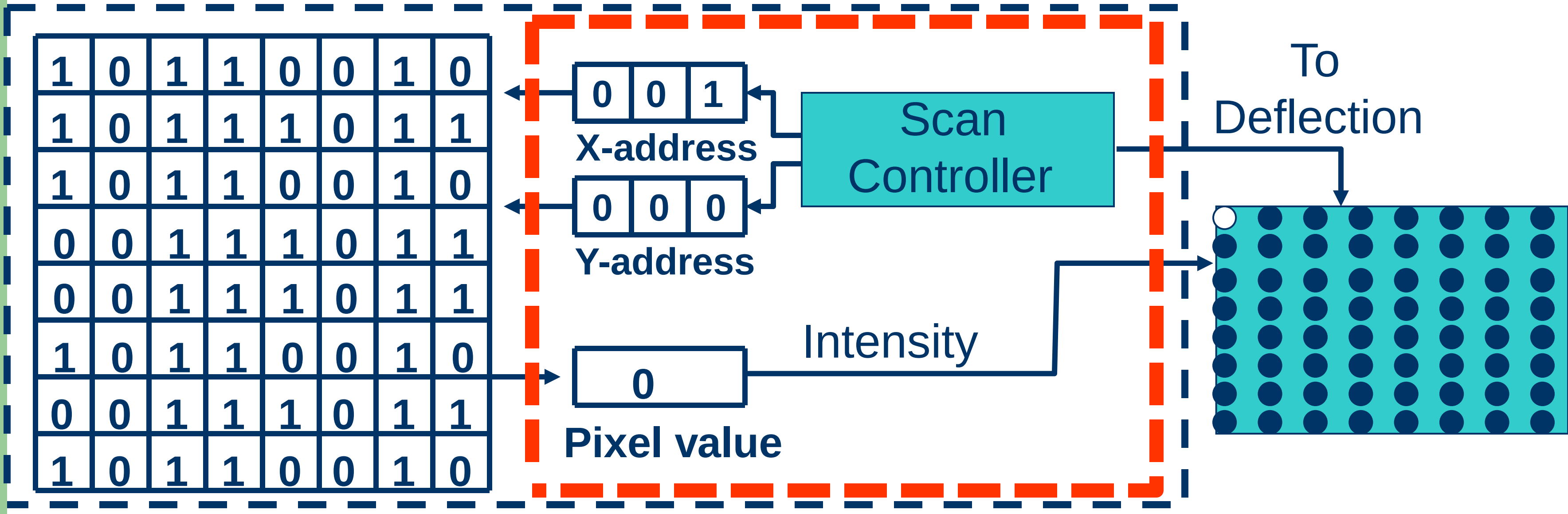
1	0	1	1	0	0	1	0
1	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0

Frame Buffer

Image Controller

First row, first pixel

Image Display System (a simple two-color raster –scan system)



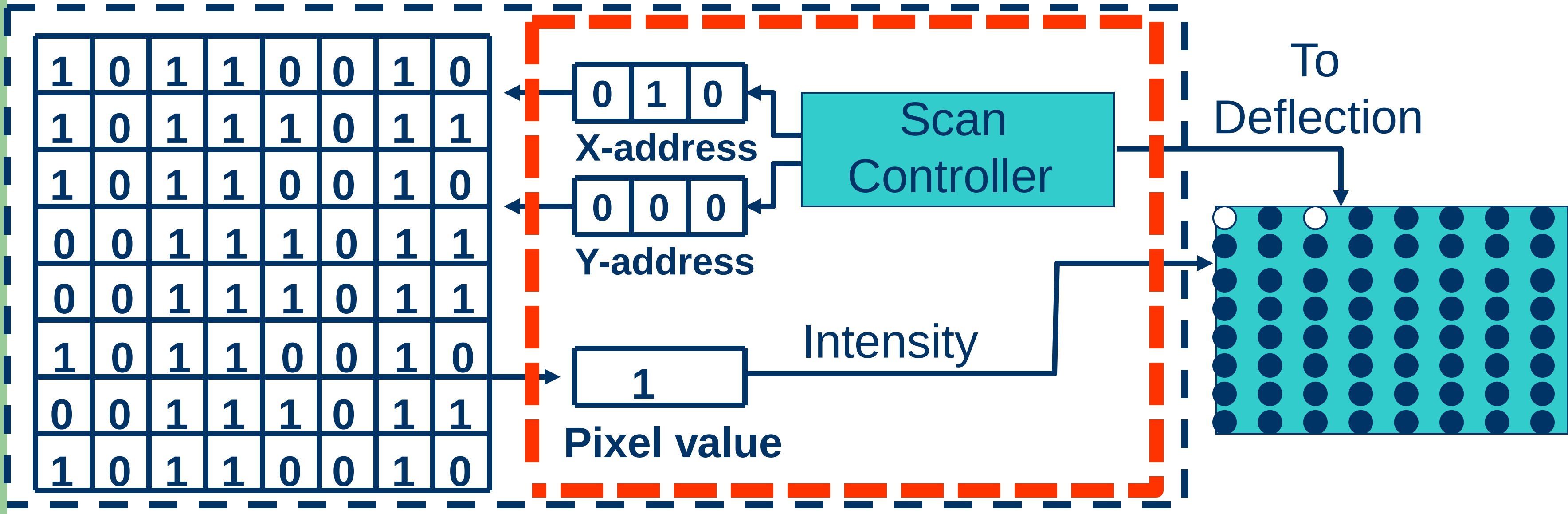
1	0	1	1	0	0	1	0
1	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0

Frame Buffer

Image Controller

First row, second pixel

Image Display System (a simple two-color raster –scan system)



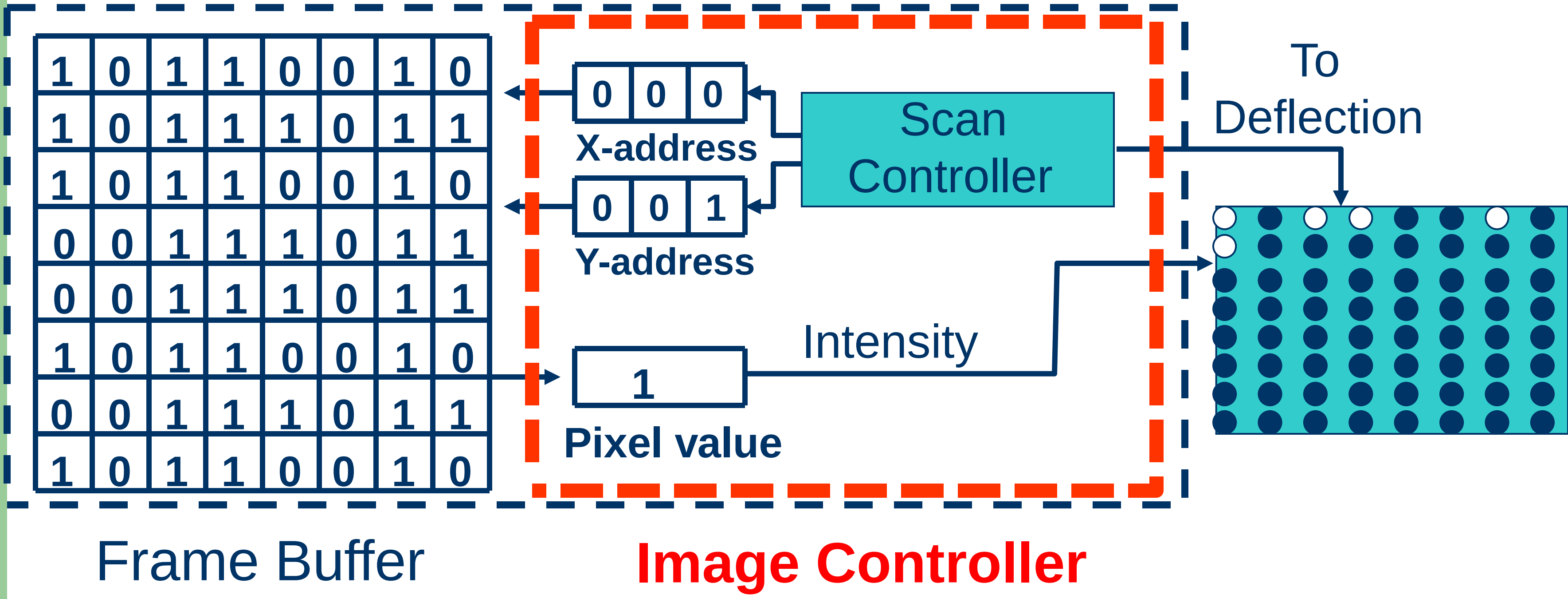
1	0	1	1	0	0	1	0
1	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0

Frame Buffer

Image Controller

First row, third pixel …

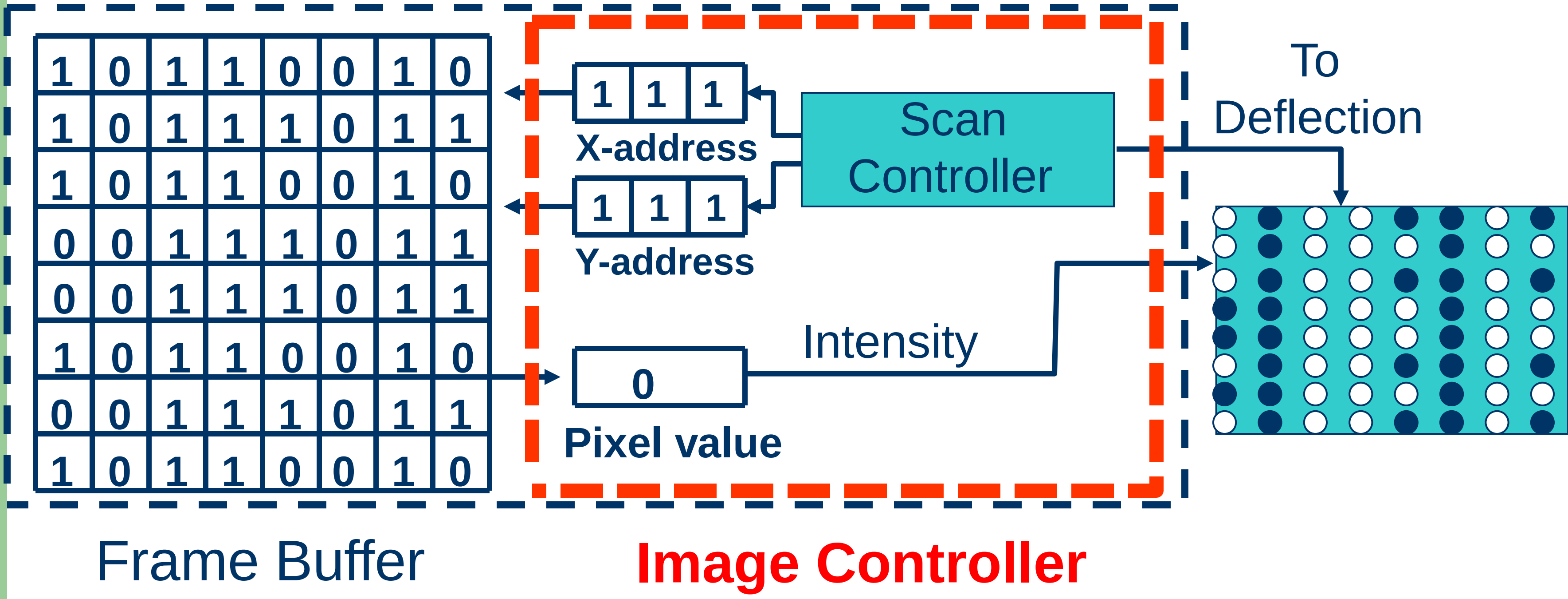
Image Display System (a simple two-color raster –scan system)



1	0	1	1	0	0	1	0
1	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0

Second row, first pixel …

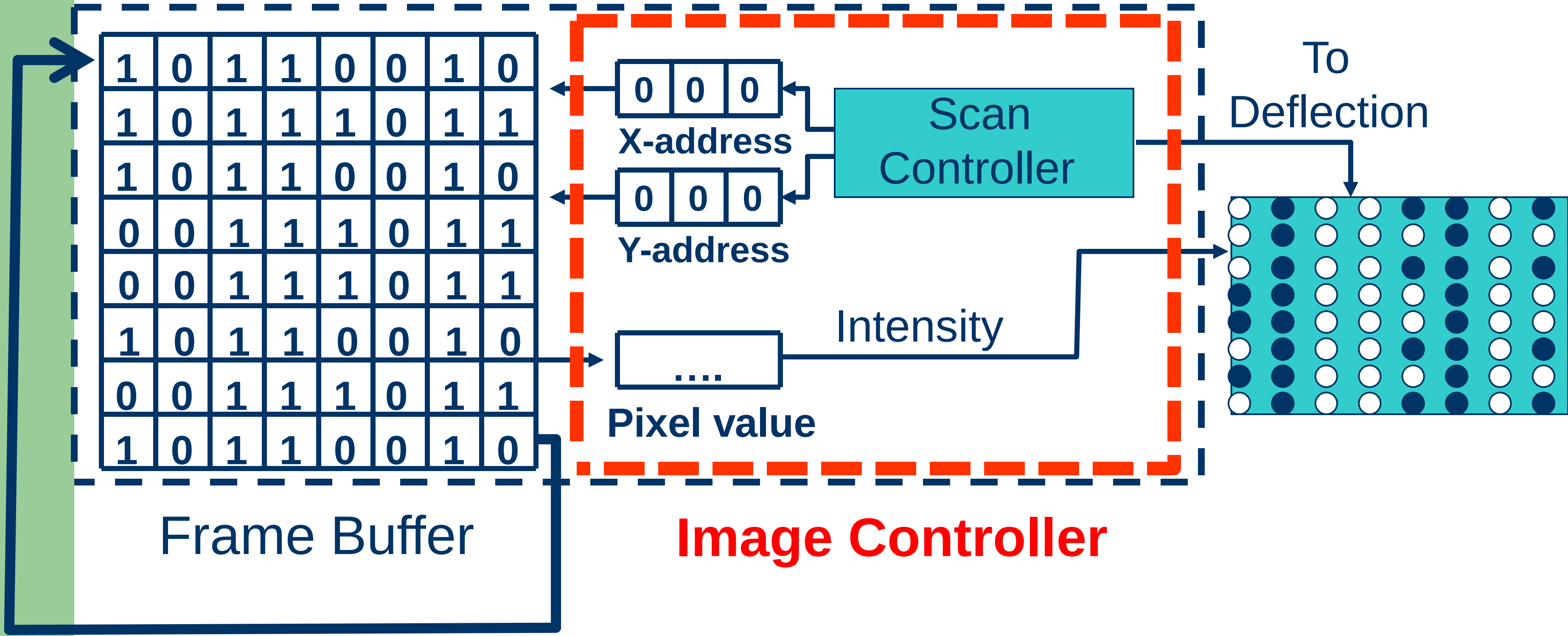
Image Display System (a simple two-color raster –scan system)



1	0	1	1	0	0	1	0
1	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0

Last row, last pixel (1/ 60 sec)

Image Display System (a simple two-color raster –scan system)

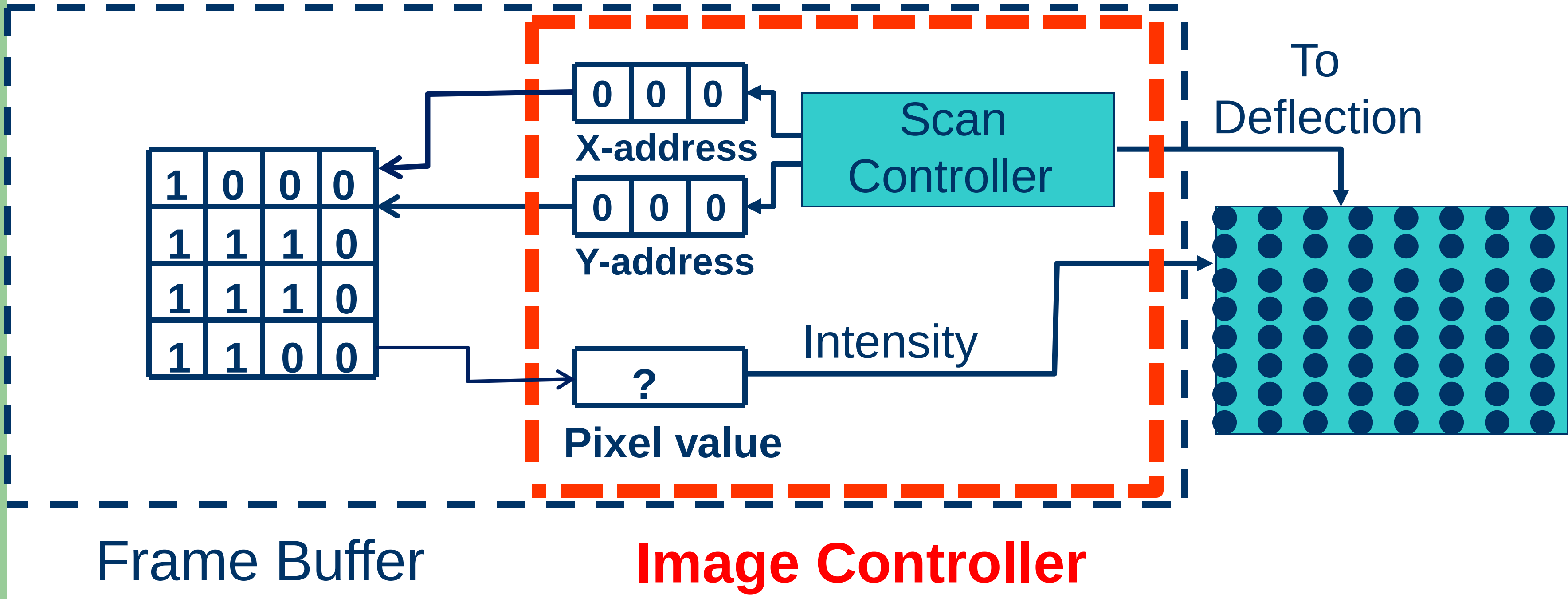


1	0	1	1	0	0	1	0
1	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0
0	0	1	1	1	0	1	1
1	0	1	1	0	0	1	0

Fly back (to do the next refresh cycle)

(1.3 millisecond; update frame buffer)

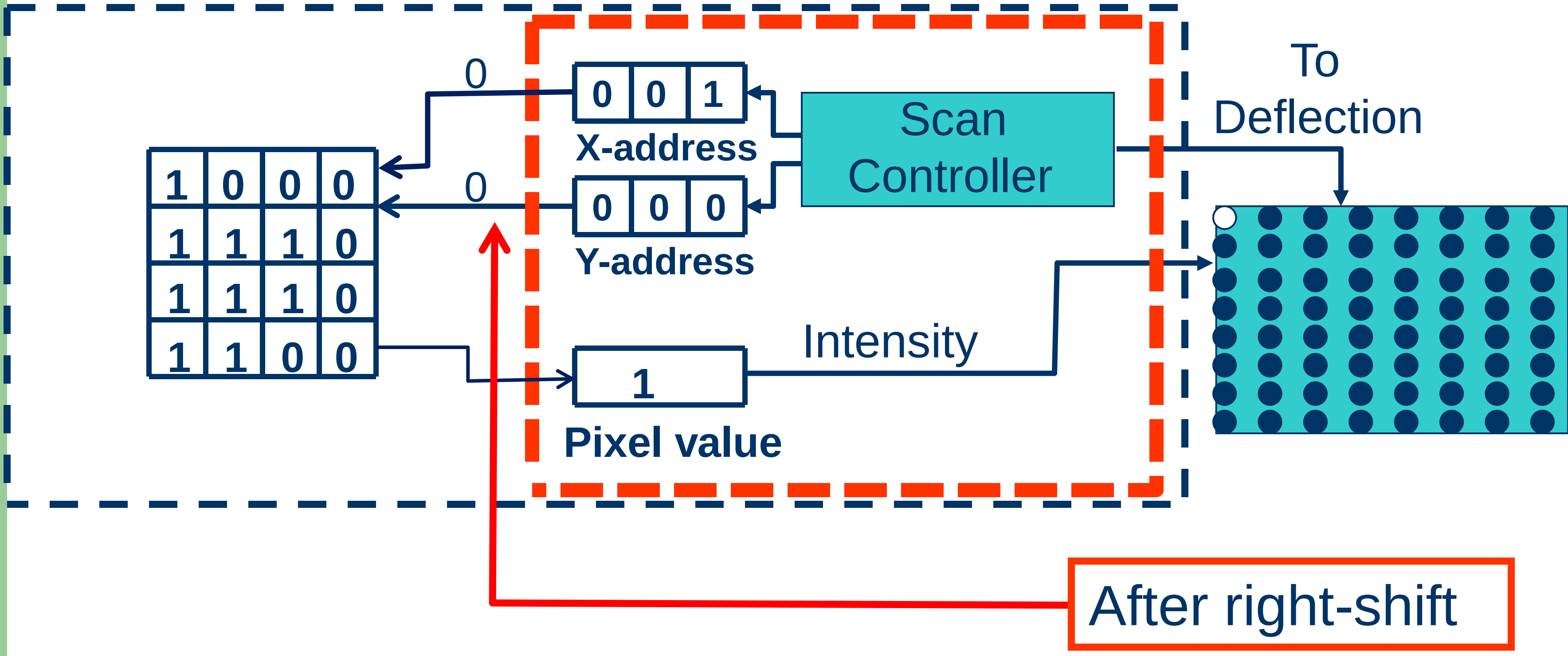
What if dimension of the frame buffer is different from resolution of the display surface?



1	0	0	0
1	1	1	0
1	1	1	0
1	1	0	0

Do a right-shift of the X- and the Y-registers before sending the indices to the Frame Buffer

What if dimension of the frame buffer is different from resolution of the display surface?

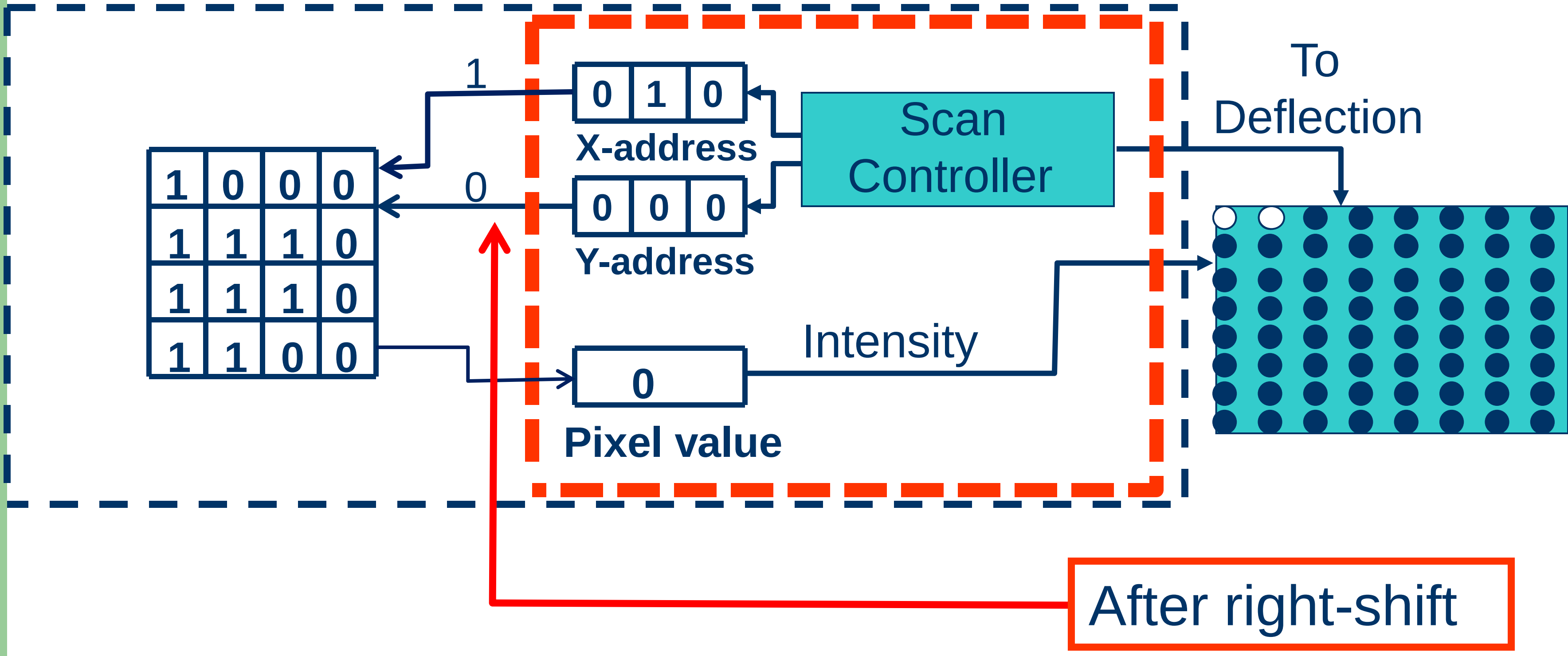


1	0	0	0
1	1	1	0
1	1	1	0
1	1	0	0

After right-shift

First row, first pixel, of the screen

What if dimension of the frame buffer is different from resolution of the display surface?

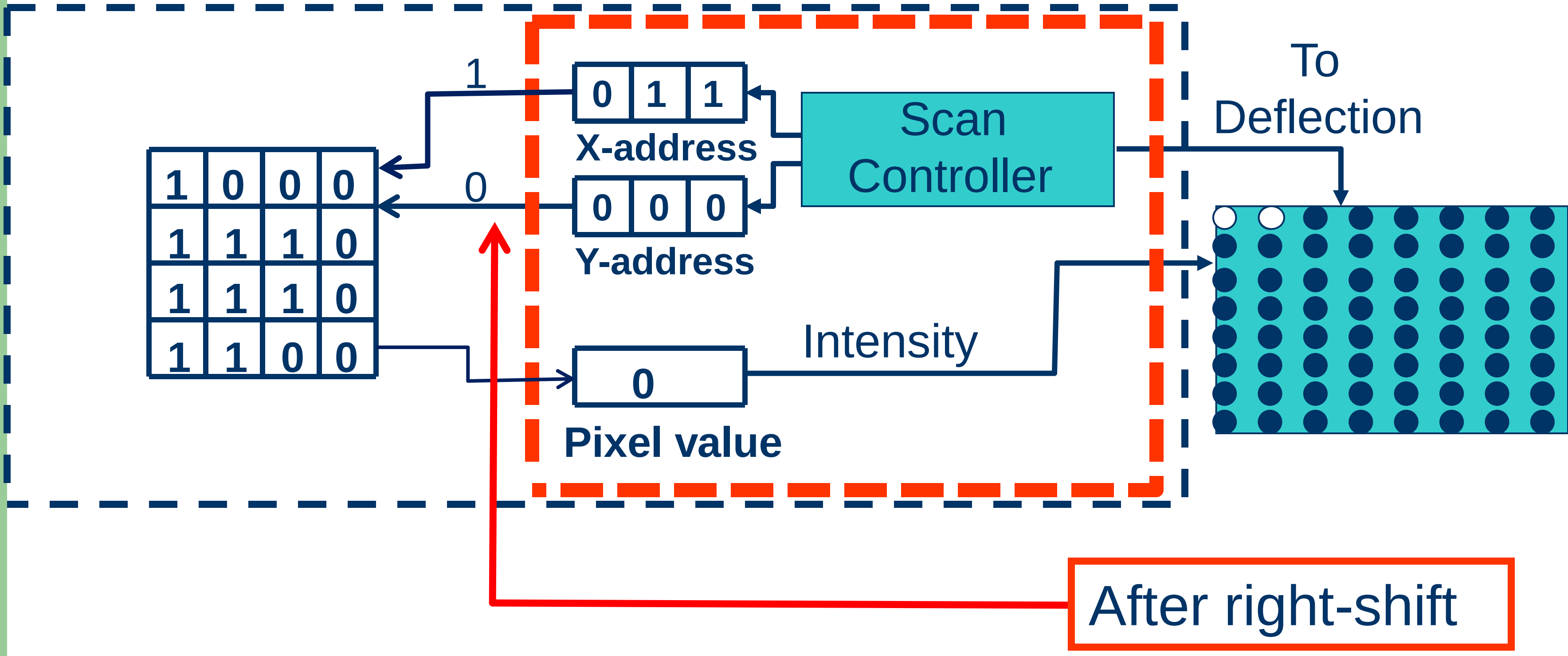


1	0	0	0
1	1	1	0
1	1	1	0
1	1	0	0

After right-shift

First row, third pixel, of the screen

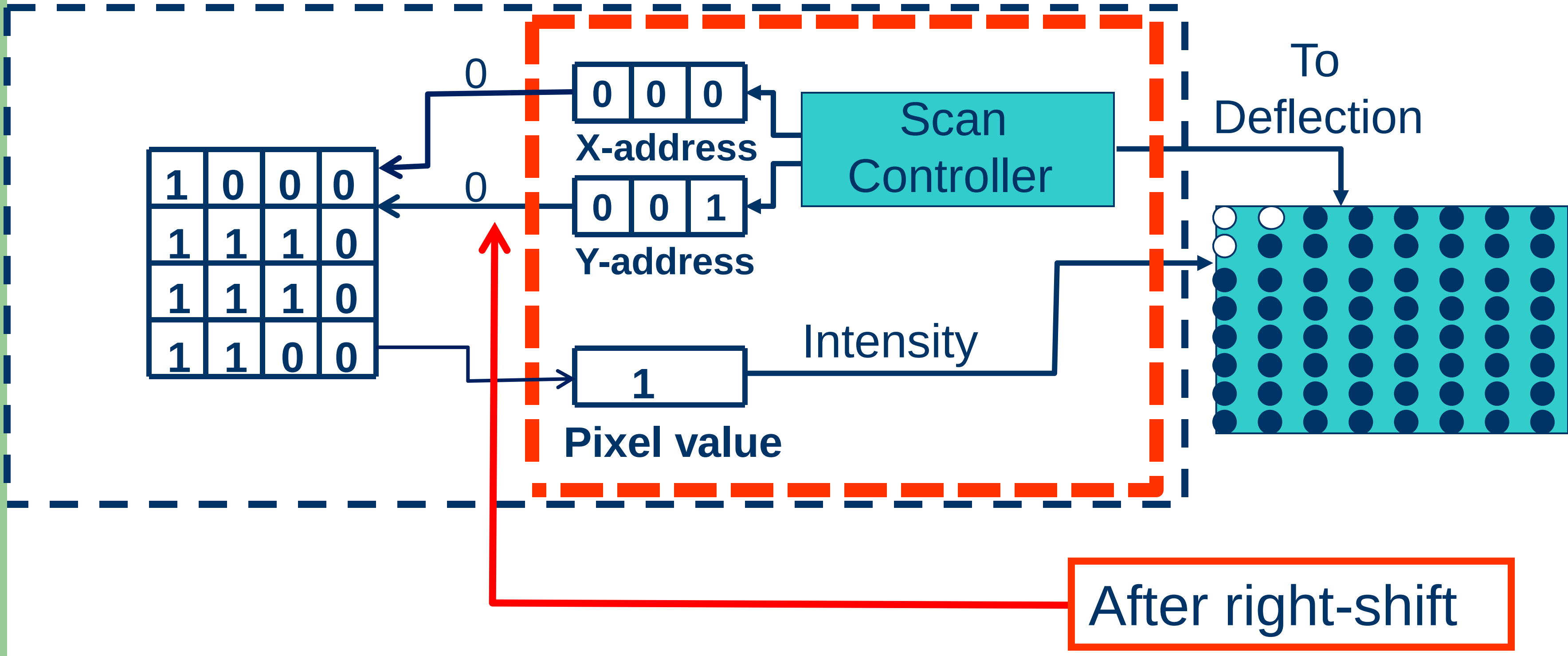
What if dimension of the frame buffer is different from resolution of the display surface?



First row, fourth pixel, of the screen

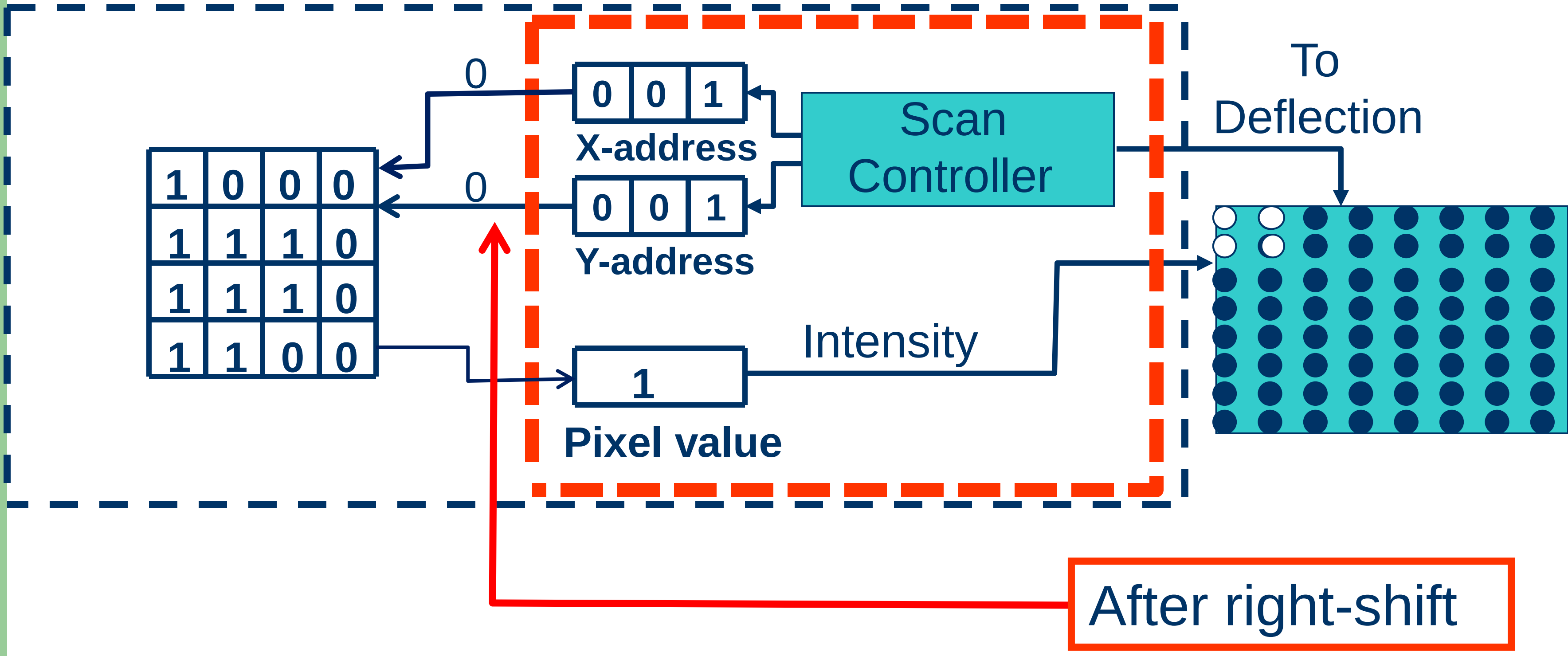
…

What if dimension of the frame buffer is different from resolution of the display surface?



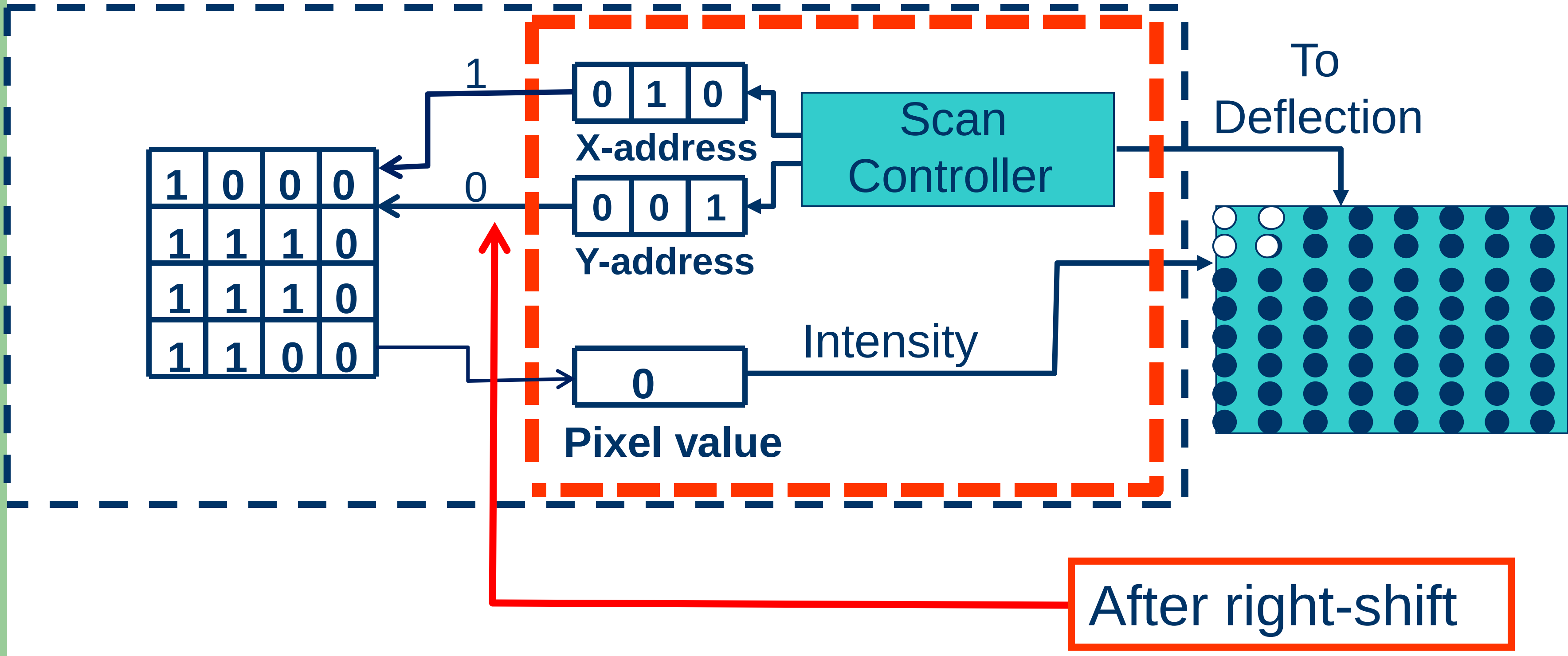
Second row, first pixel, of the screen

What if dimension of the frame buffer is different from resolution of the display surface?



Second row, 2nd pixel, of the screen

What if dimension of the frame buffer is different from resolution of the display surface?

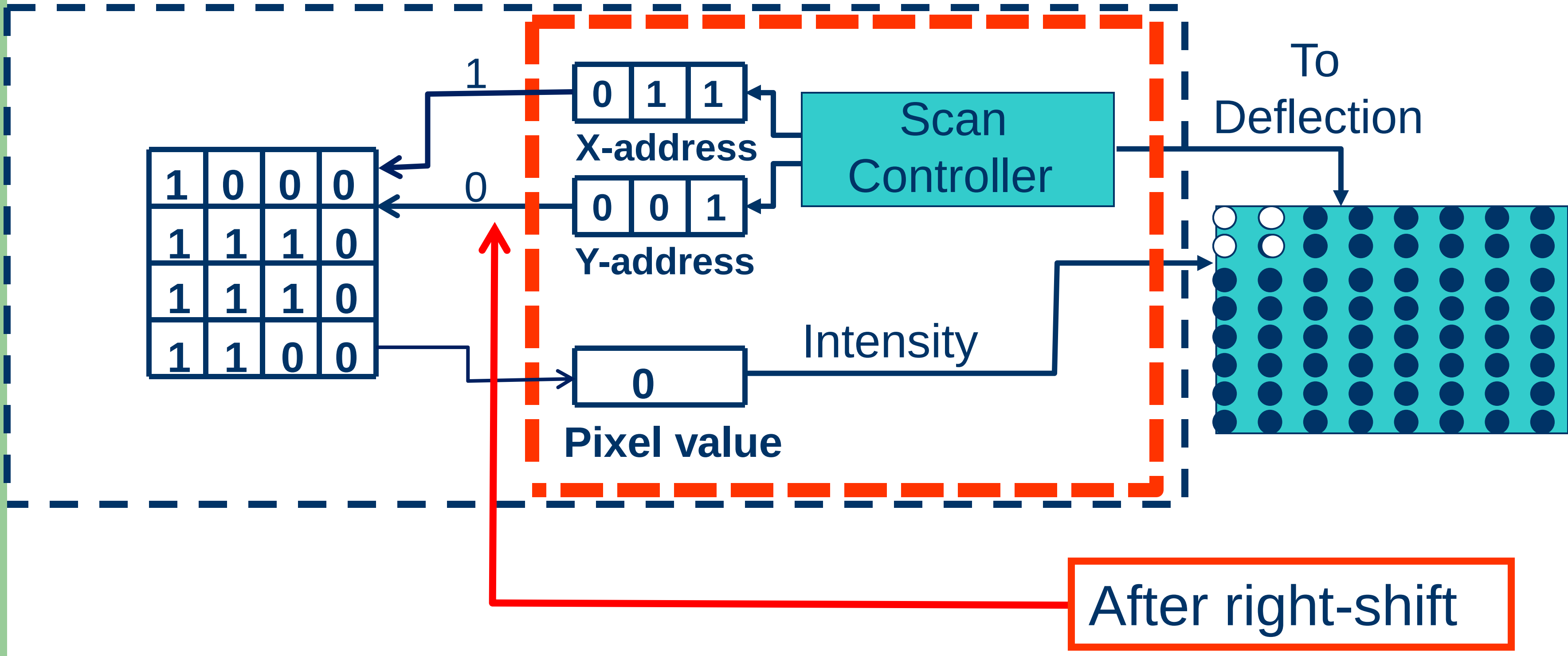


1	0	0	0
1	1	1	0
1	1	1	0
1	1	0	0

After right-shift

Second row, 3rd pixel, of the screen

What if dimension of the frame buffer is different from resolution of the display surface?



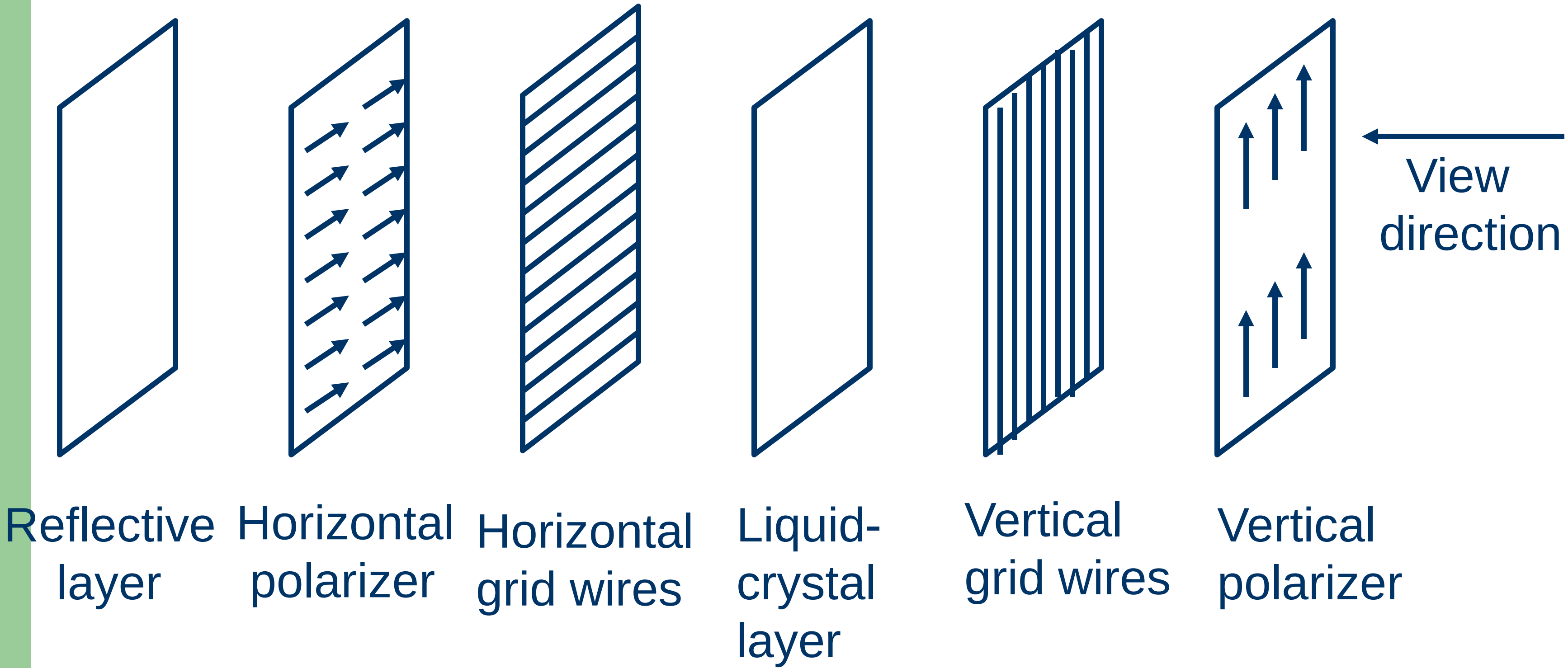
Second row, 4th pixel, of the screen

1.9 Flat-Panel Displays

- Liquid-crystal display (LCD)
- ~~Active matrix panel (AMP)~~
- ~~Plasma panel~~

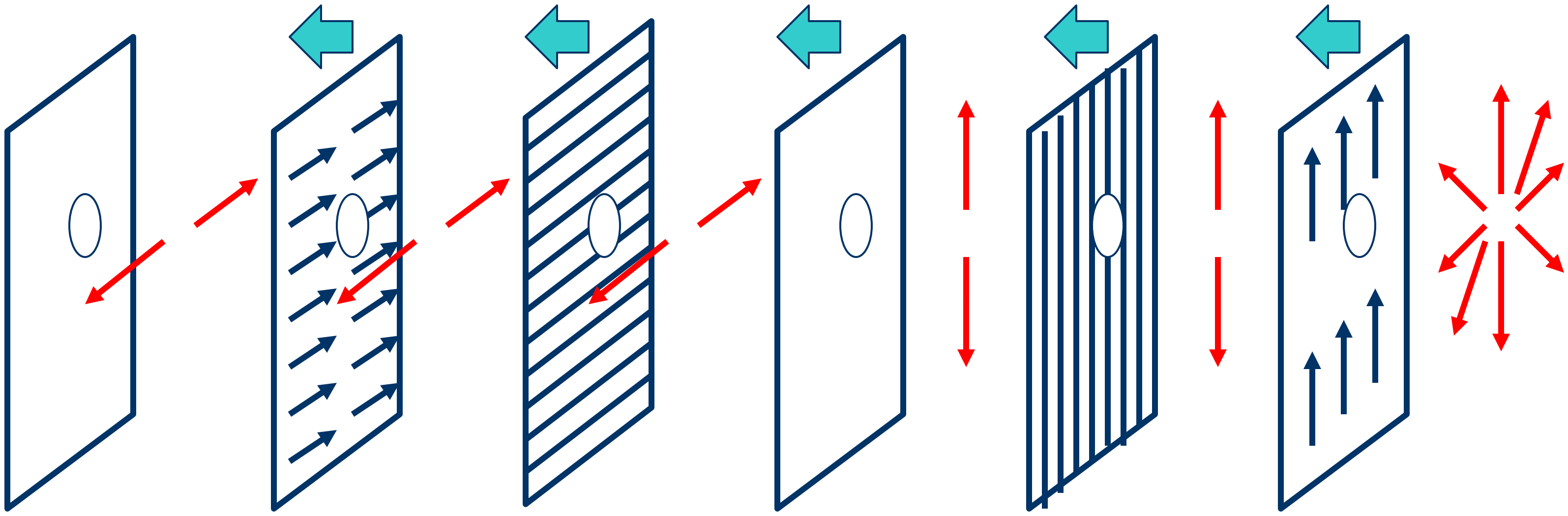
Without Backlight module

Reflective
Liquid-crystal display (LCD)

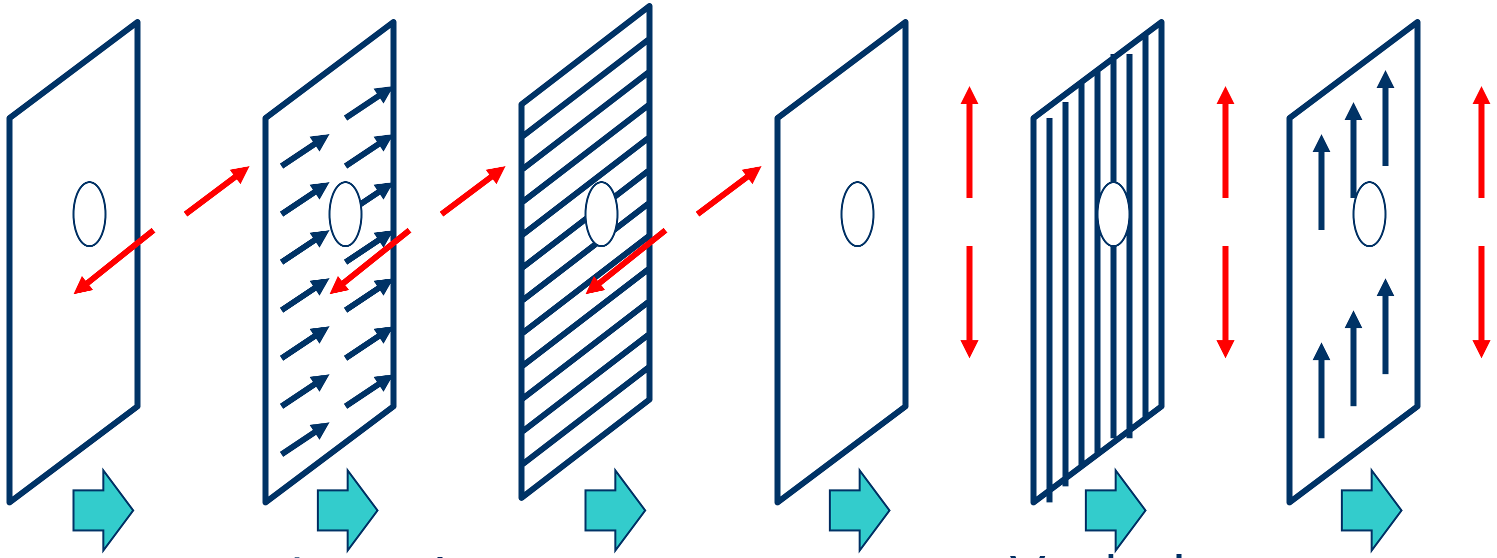


Reflective
Liquid-crystal display (LCD)

(with polarizing effect)



Reflective
Liquid-crystal display (LCD)



Reflective layer

Horizontal polarizer

Horizontal grid wires

Liquid-crystal layer

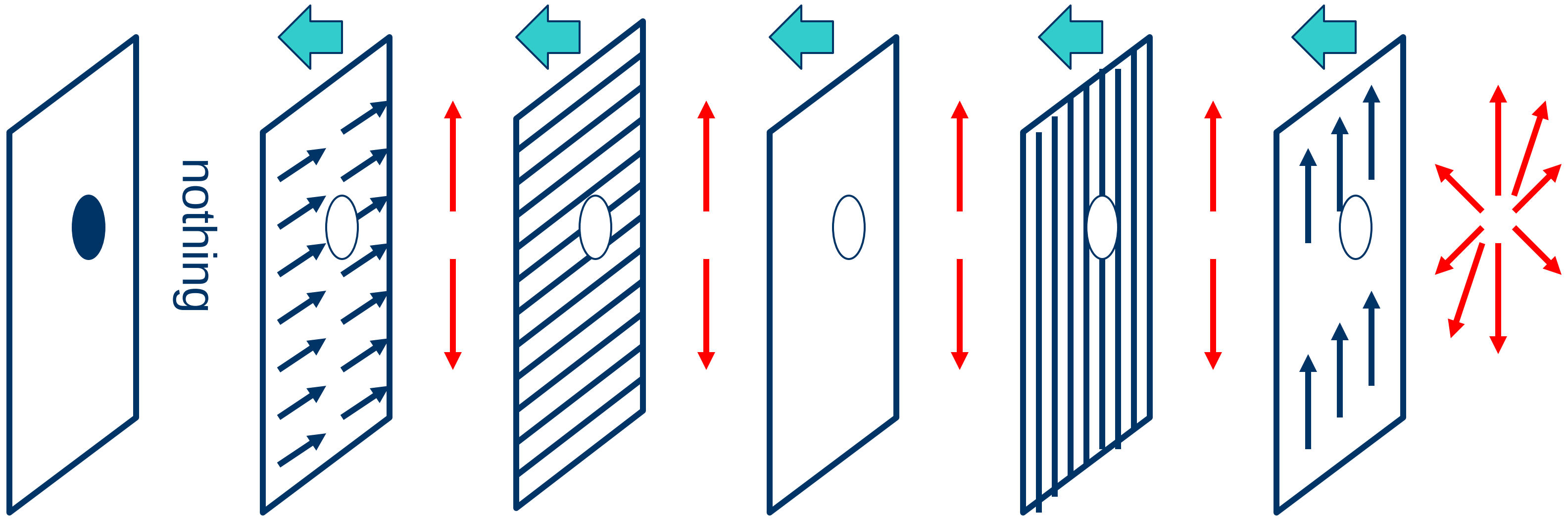
Vertical grid wires

Vertical polarizer

With polarizing effect

Reflective
Liquid-crystal display (LCD)

(without polarizing effect)



Reflective layer

Horizontal polarizer

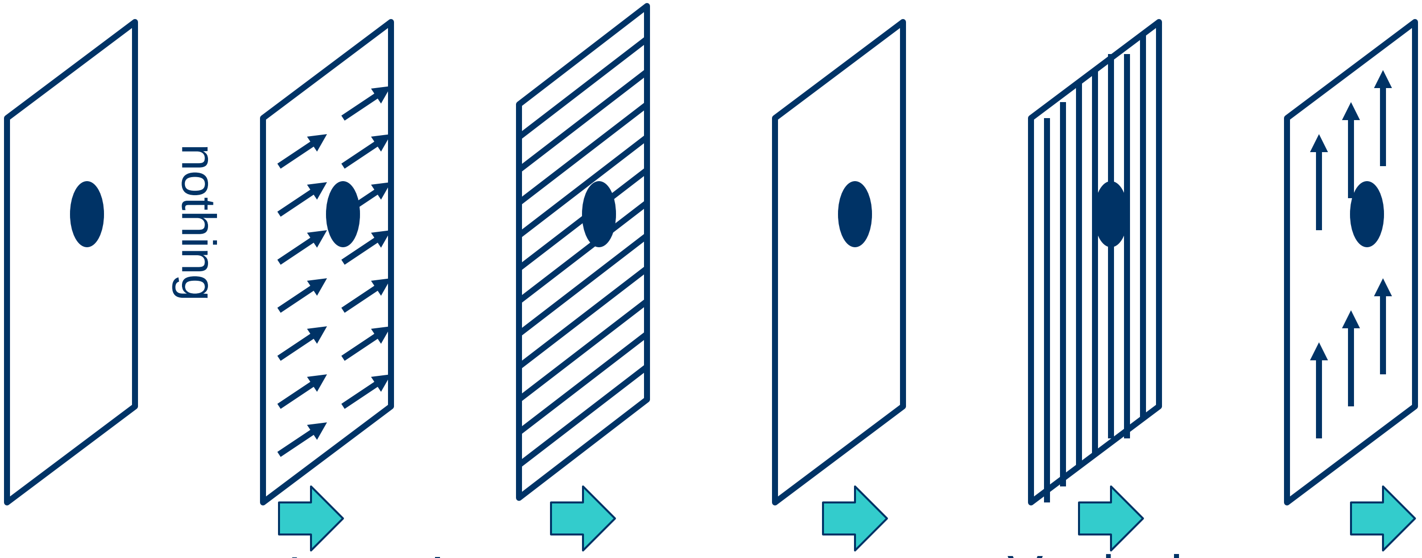
Horizontal grid wires

Liquid-crystal layer

Vertical grid wires

Vertical polarizer

Reflective
Liquid-crystal display (LCD)



nothing

Reflective layer

Horizontal polarizer

Horizontal grid wires

Liquid-crystal layer

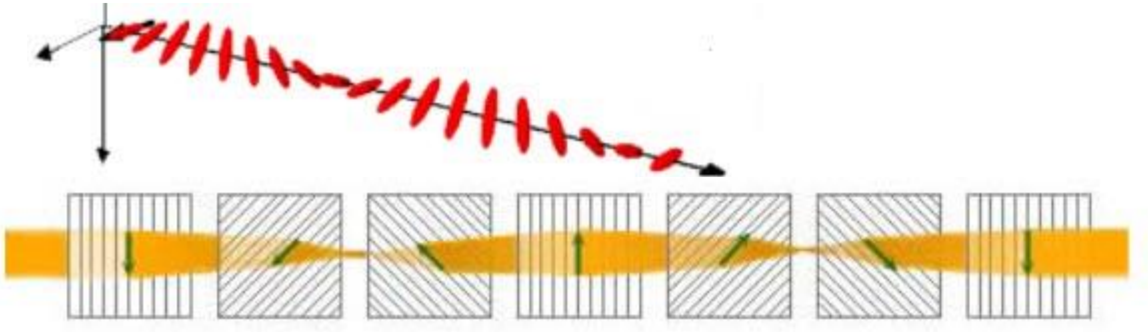
Vertical grid wires

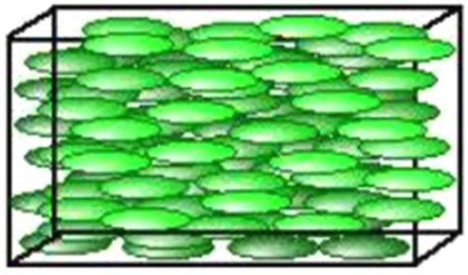
Vertical polarizer

Without polarizing effect

Reflective
Liquid-crystal display (LCD)

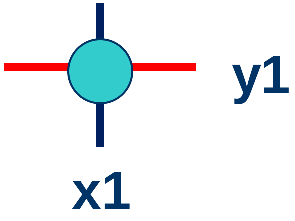
- Six layers (see the above figure)
- Liquid-crystal is made up of long crystalline molecules arranged in a spiral fashion
- Direction of polarization of polarized light passing through is rotated 90 degrees
- The crystals line up in the same direction when in an electric field, therefore no polarizing effect





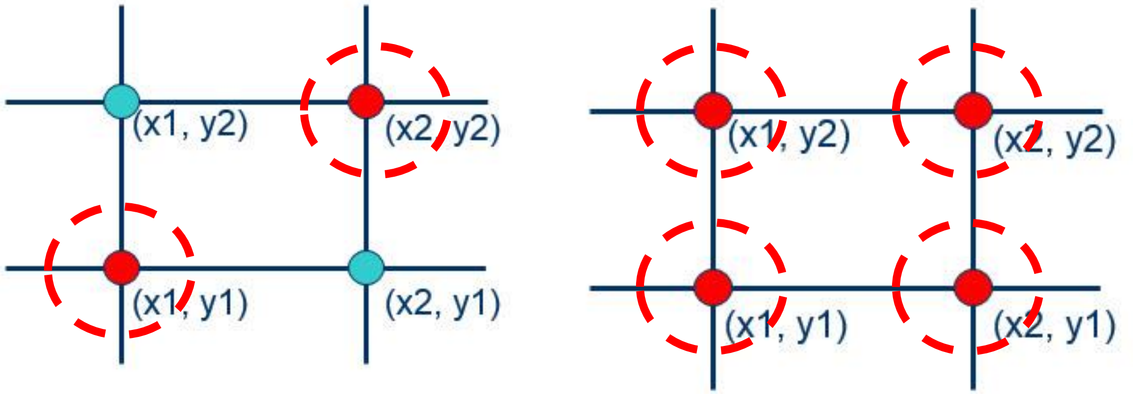
Reflective
Liquid-crystal display (LCD)

- In this case the light passing through the liquid-crystal layer will be absorbed by the rear polarizer, so the viewer sees a dark spot on the display
- To create a dark spot at (*x*1, *y*1), use *matrix addressing*: applying a negative voltage −*V* to the vertical grid wire *x*1 and a positive voltage +*V* to the horizontal grid wire *y*1 to create an electric field at (*x*1, *y*1).

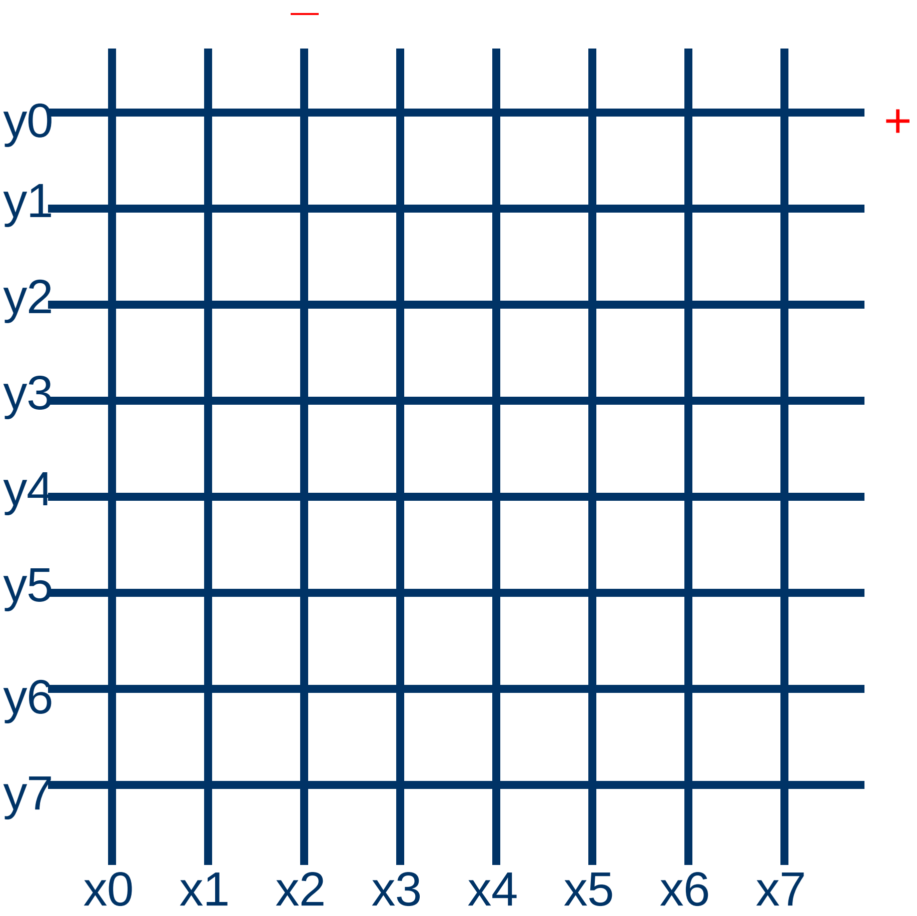


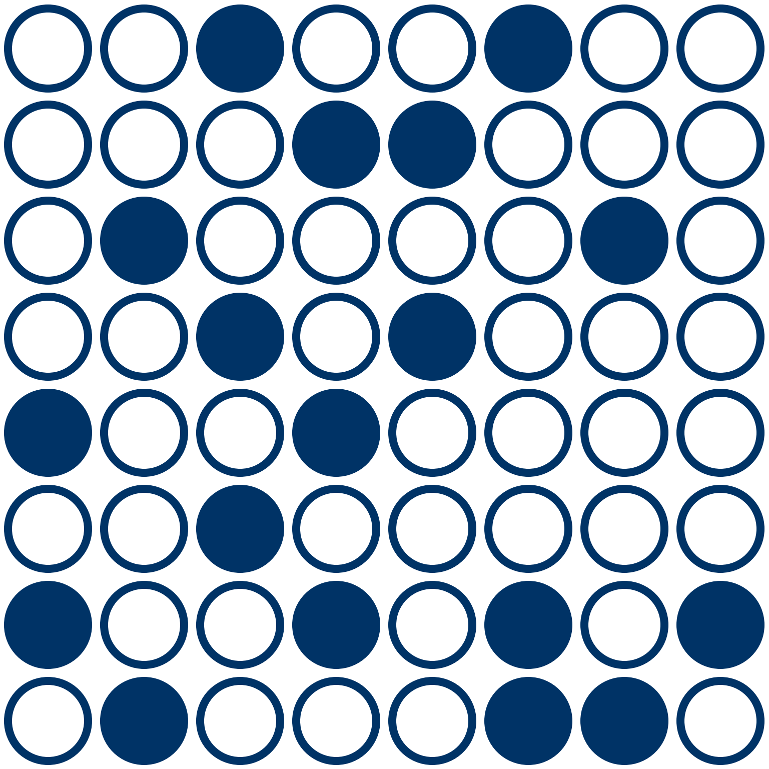
Reflective
Liquid-crystal display (LCD)

- To display dots at (*x*1, *y*1) and (*x*2, *y*2), cannot simply apply negative voltage to *x*1 and *x*2 and positive voltage to *y*1 and *y*2: that would cause dots to appear at (*x*1, *y*1), (*x*1, *y*2), (*x*2, *y*1) and (*x*2, *y*2). We have to activate them one at a time. The display is refreshed one row at a time.



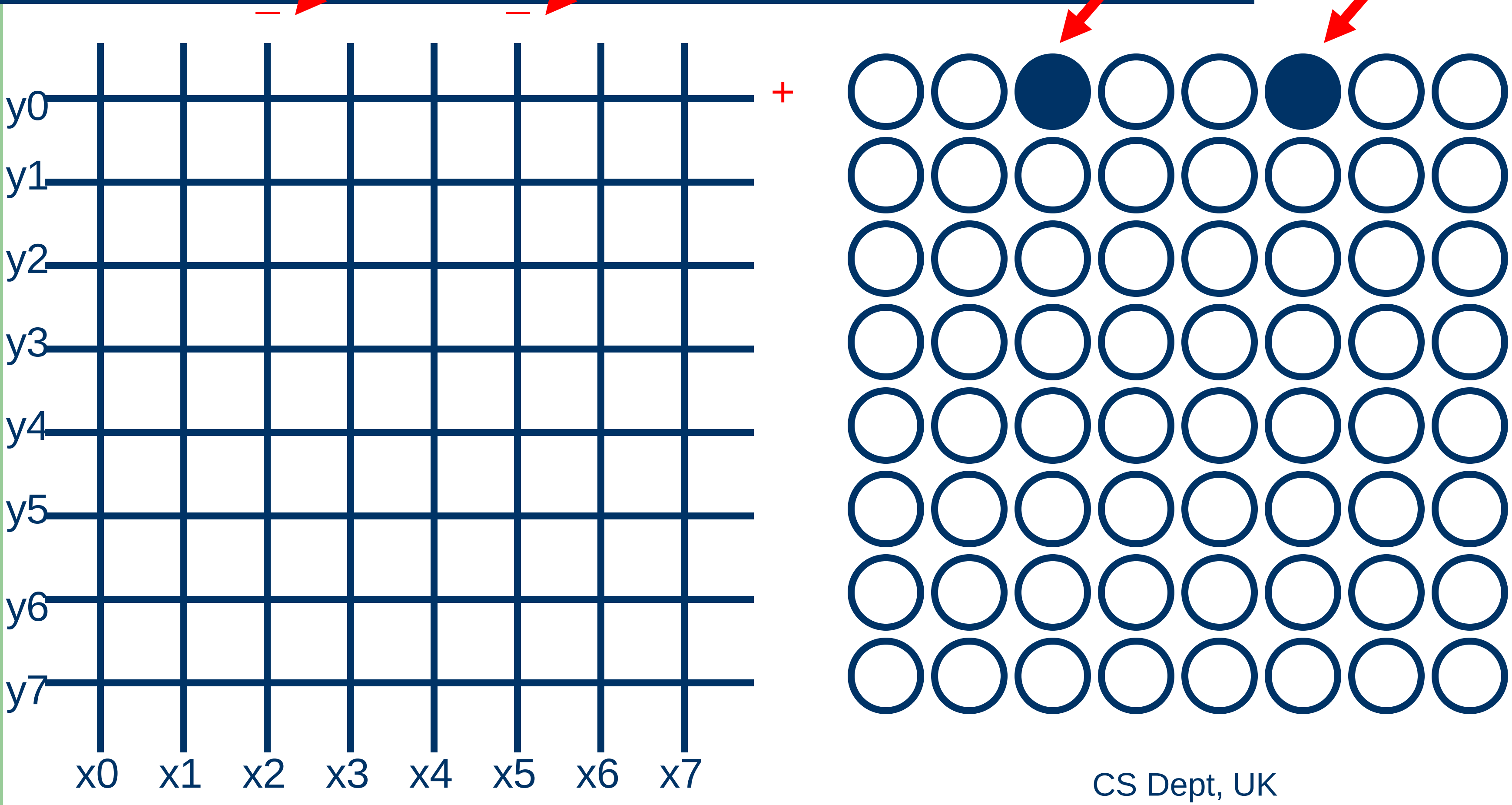
How is an image created on a reflective LCD?



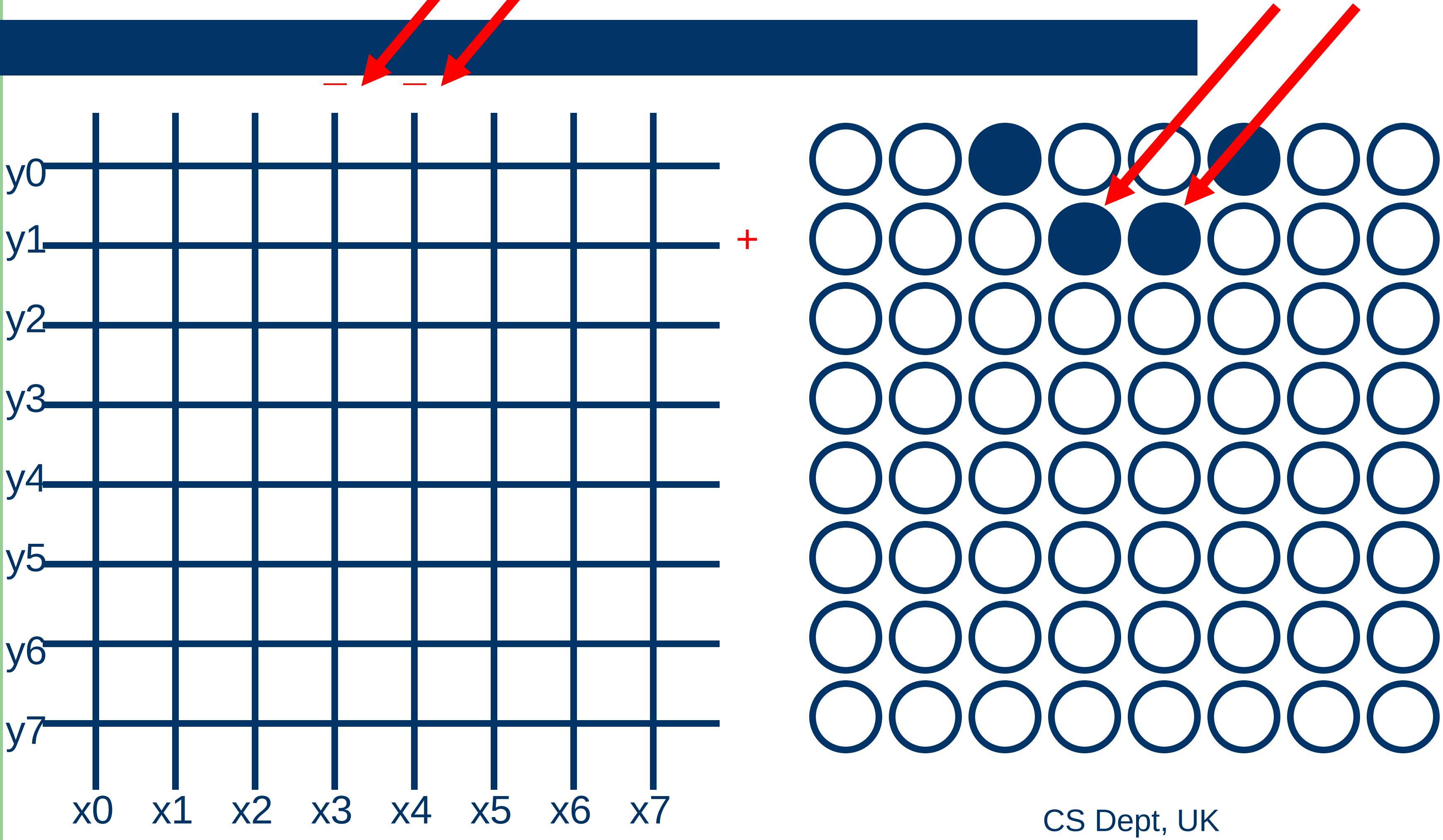


○	○	●	○	○	●	○	○
○	○	○	●	●	○	○	○
○	●	○	○	○	○	●	○
○	○	●	○	●	○	○	○
●	○	○	●	○	○	○	○
○	○	●	○	○	○	○	○
●	○	○	●	○	●	○	●
○	●	○	○	○	●	●	○

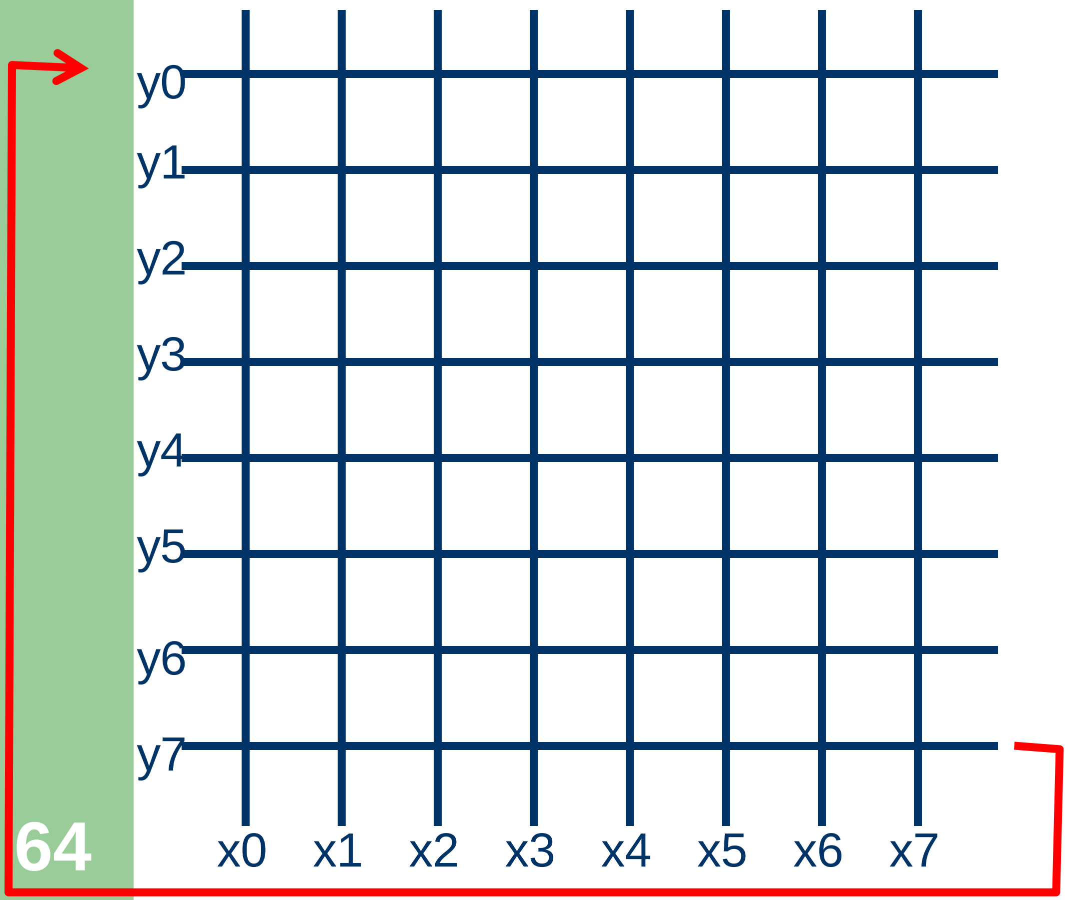
First row:



Second row:

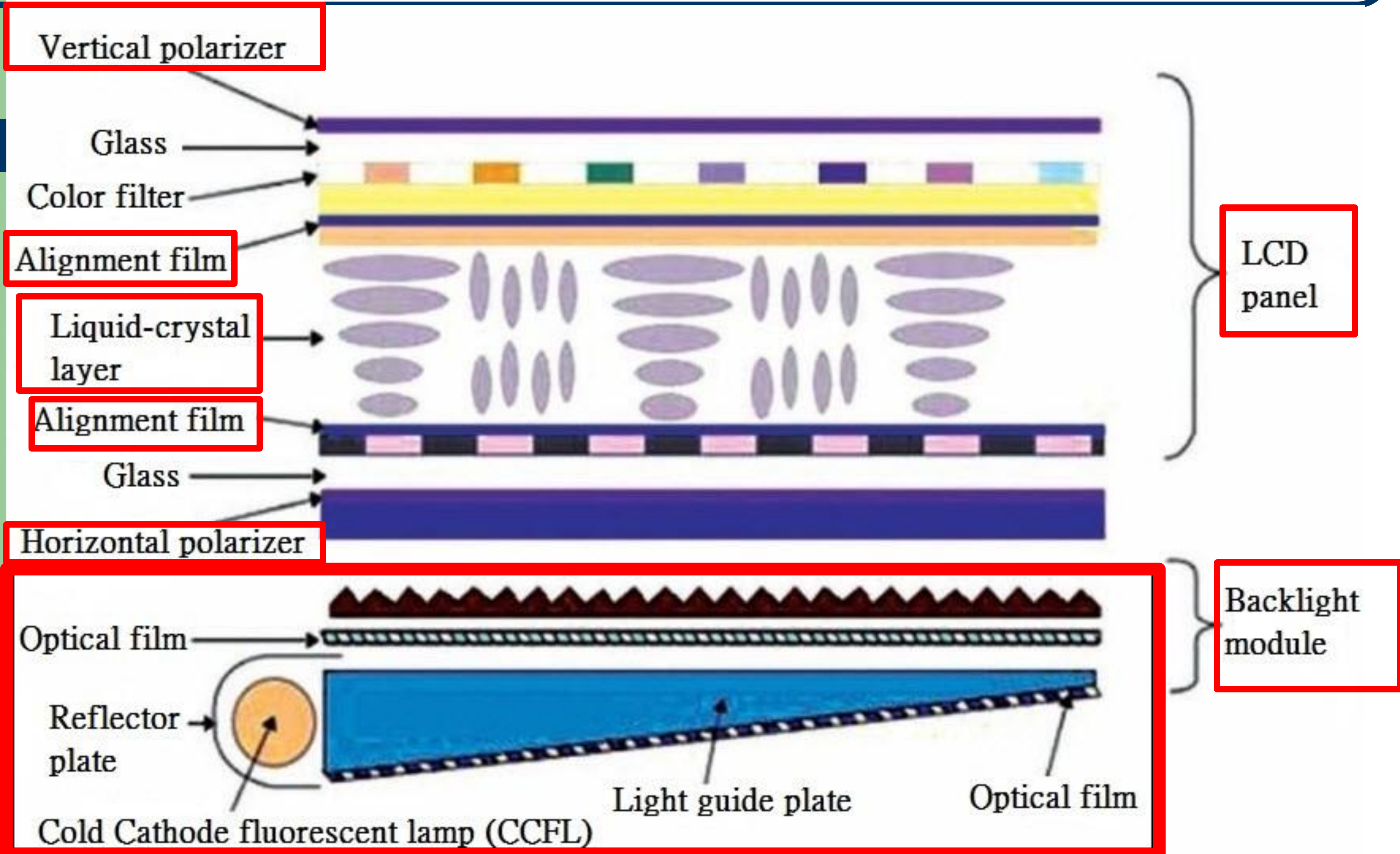


Last row:

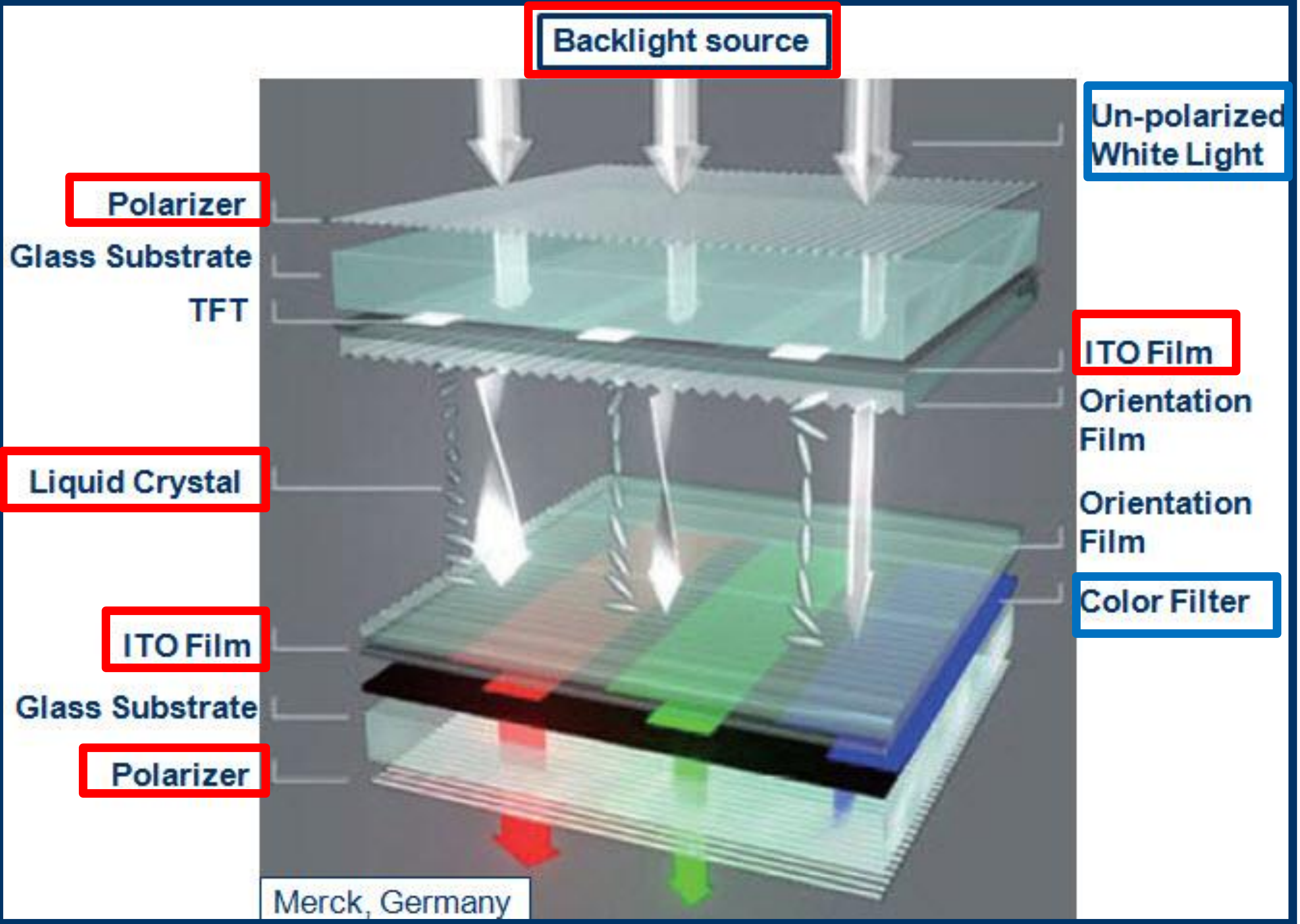


○	○	●	○	○	●	○	○
○	○	○	●	●	○	○	○
○	●	○	○	○	○	●	○
○	○	●	○	●	○	○	○
●	○	○	●	○	○	○	○
○	○	●	○	○	○	○	○
●	○	○	●	○	●	○	●
○	●	○	○	○	●	●	○

Transmissive LCD (single pixel)

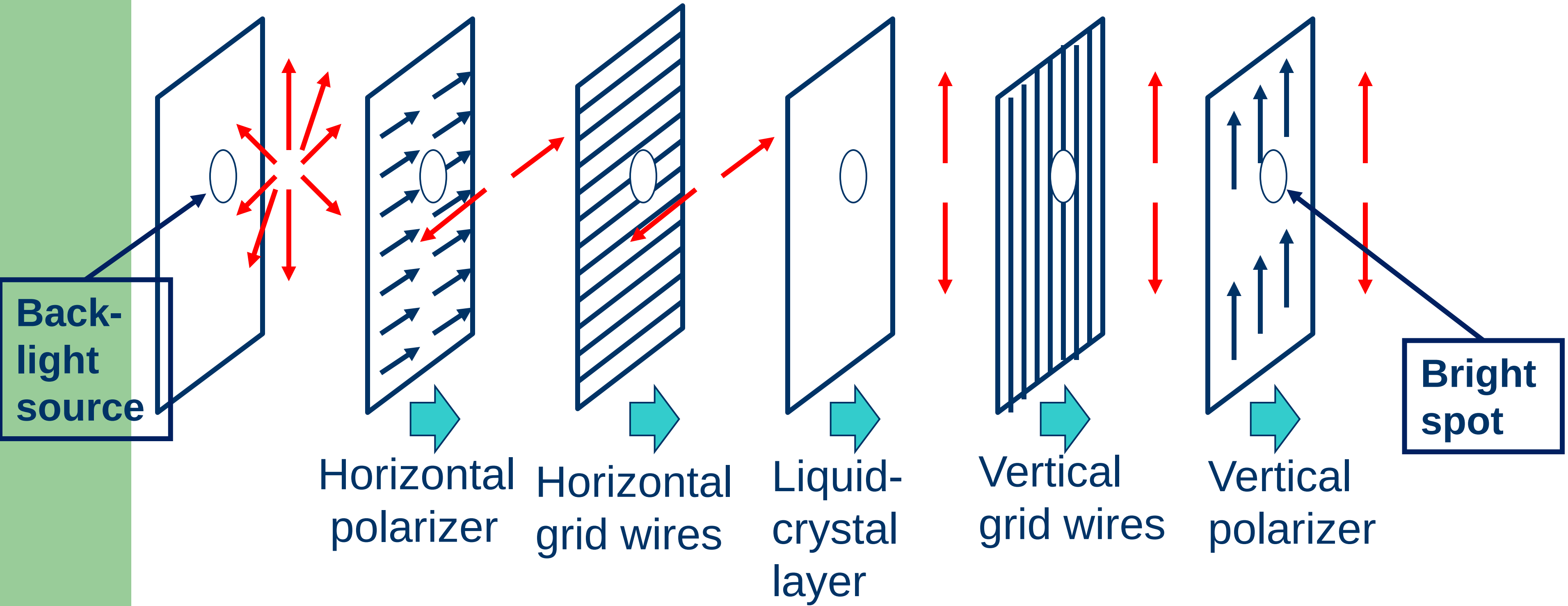


Color Transmissive LCD (single pixel w/ rgb sub)



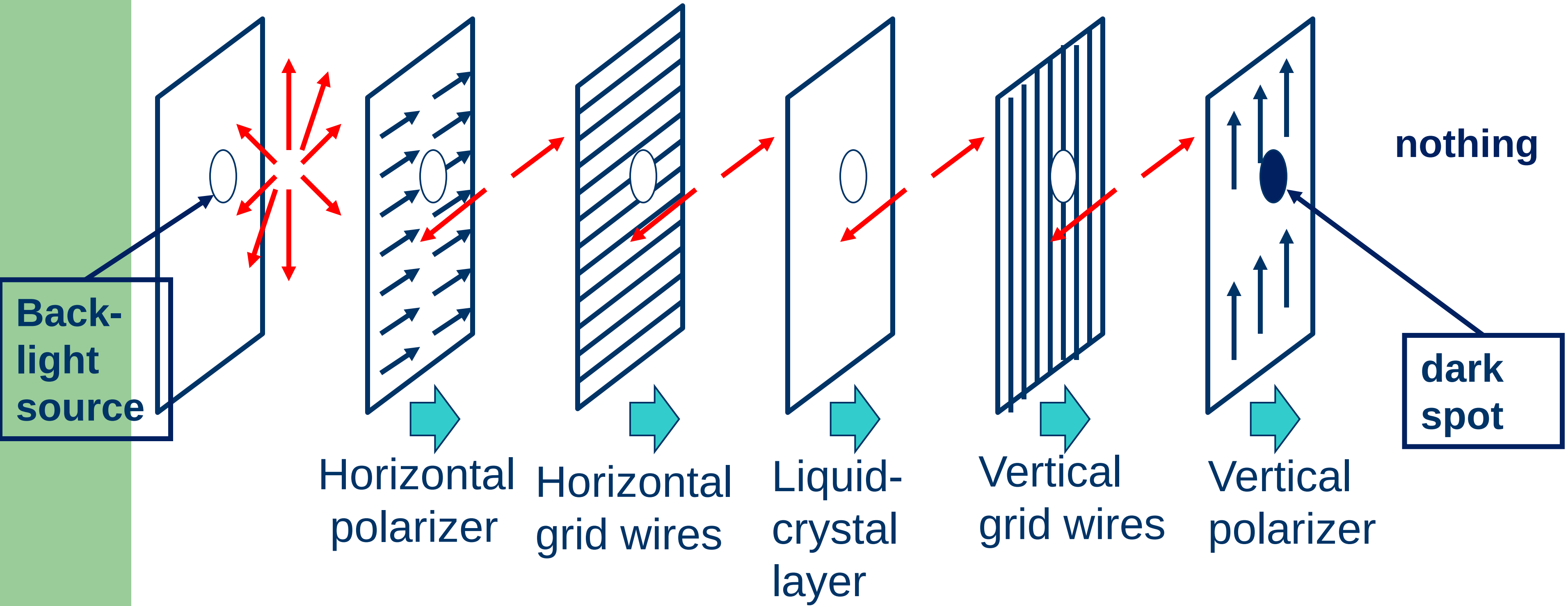
Transmissive Liquid-crystal display (LCD)

(with polarizing effect)



Transmissive Liquid-crystal display (LCD)

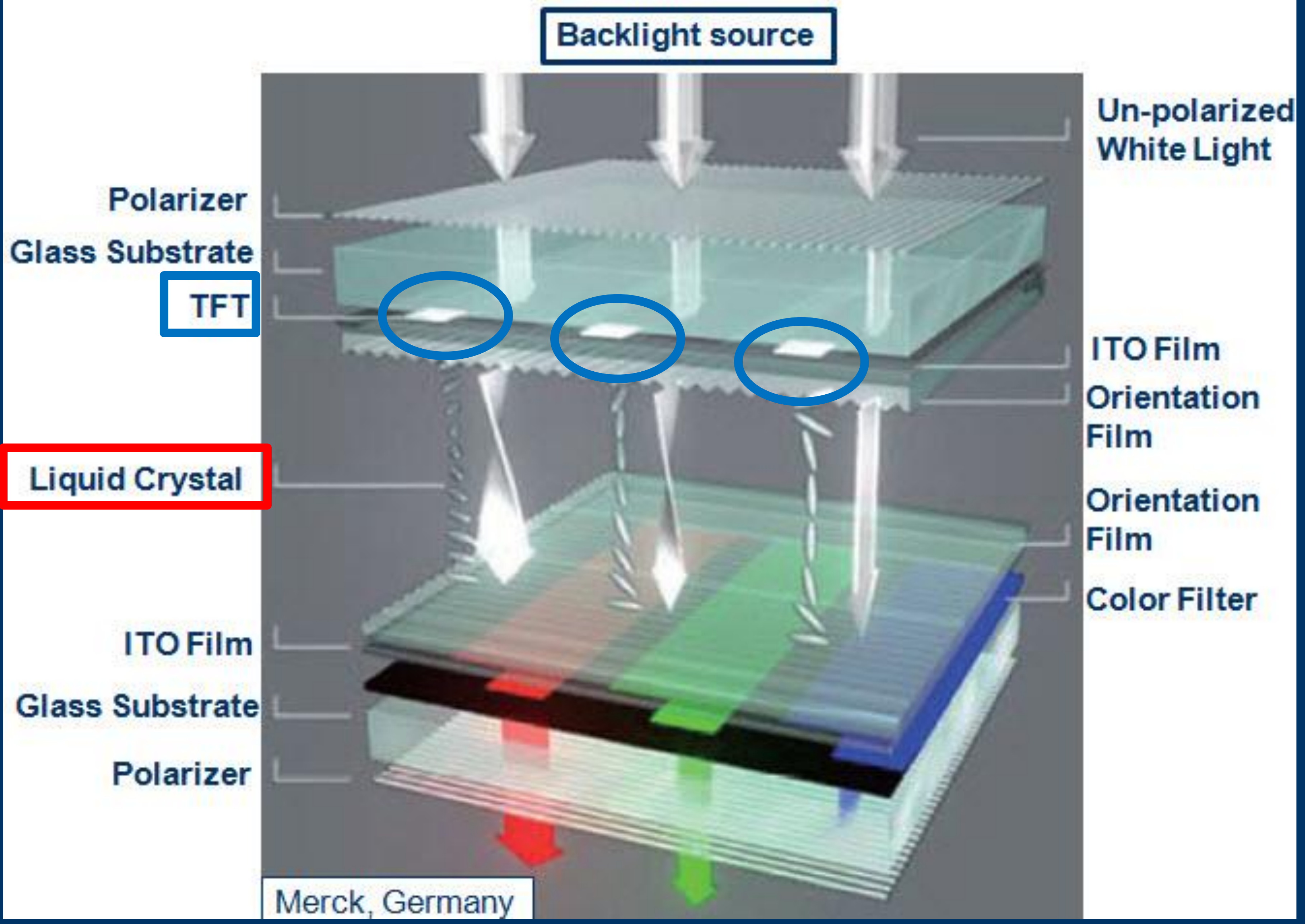
(without polarizing effect)



Active Matrix Panel (TFT LCD)

- LCD panel with a thin-film transistor (TFT) at each grid point
- Transistor can hold the cell in "adjusted" state until changed
- The display need not be refreshed and is brighter

Color Transmissive LCD (single pixel w/ rgb sub)



Merck, Germany

Plasma Panel

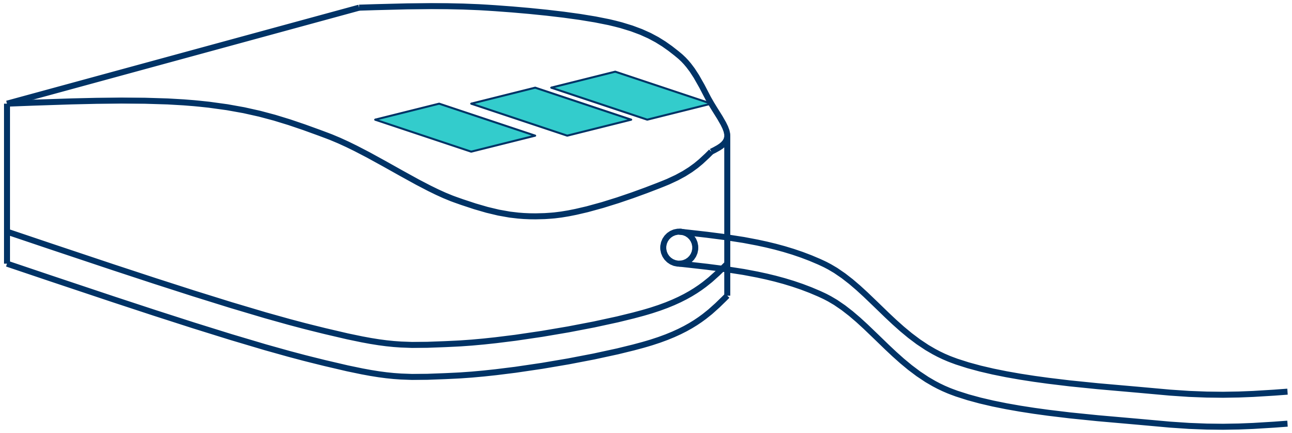
- Similar to the center part of the previous figure
- Array of tiny neon bulbs
- Need not be refreshed

1.10 Input Devices

- Logical Classes of devices and techniques

Logical Device	Function	**Physical device**
Keyboard	Input character string	**Alphnumeric keyboard**
Locator	Indicate a position and/or orientation	Tablet, mouse, joystick
Pick	**Select a displayed entity**	Light pen
Choice	Select from a set of actions or choices	PFK, mouse
Dial (Valuator)	Input an analog value (number)	Slidebar, potentiometer

Mouse – most commonly used

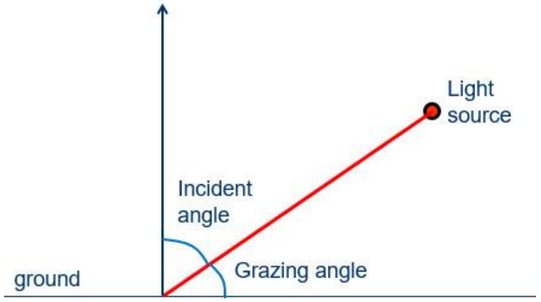


Mouse – most commonly used

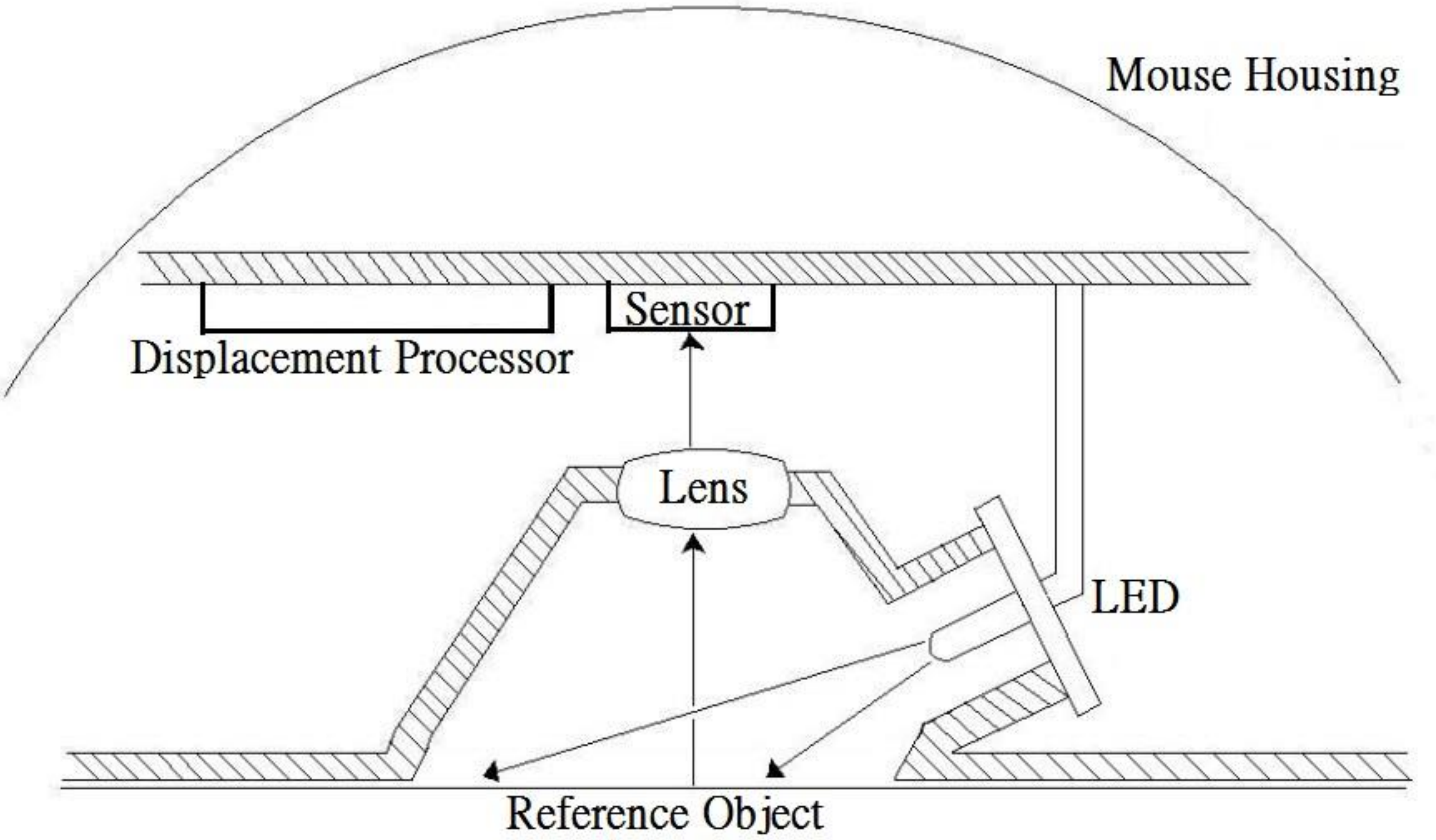
- using mechanical detector or optical detector to measure motion
- mechanical mice measure distance by turning a ball (at the bottom) and consequently a pair of encoders. The encoders measure motion in two directions.
- old optical mice measure distance traveled by counting lines on a special pad

Mouse – most commonly used

- modern surface-independent optical mice work by using an optoelectronic sensor (essentially, a tiny low-resolution video camera) to take successive images of the surface on which the mouse operates.
- the surface is lit at a *grazing angle* by a light emitting diode (LED).



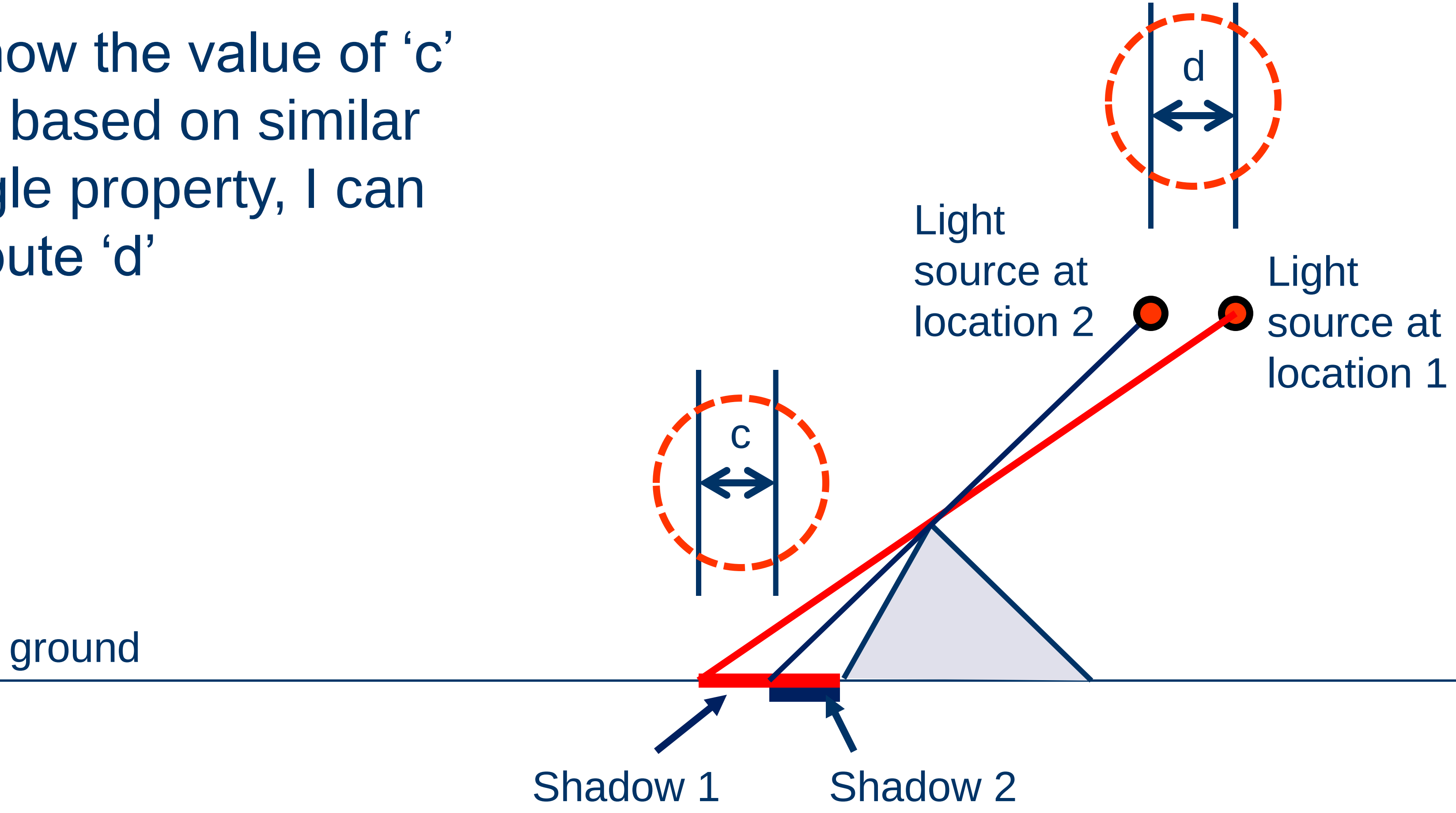
Mouse – most commonly used



Mouse – most commonly used

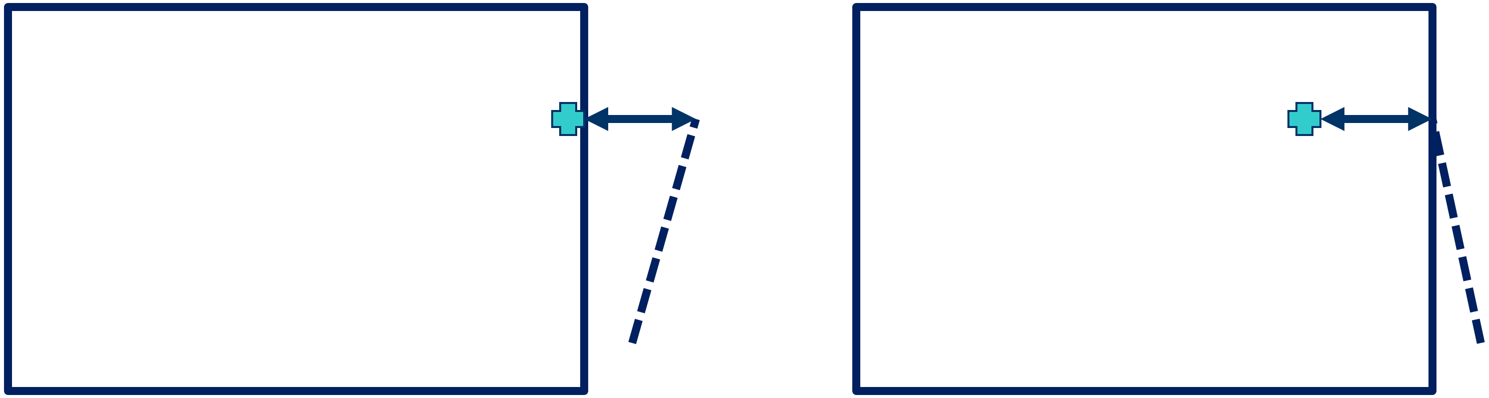
- the purpose is for the texture of the surface to cast shadows on the surface itself, like the situation of a hilly terrain lit at sunset.
- images taken of the surface are then compared to determine how far the mouse has moved.
- the displacement information is then sent to the computer to update the location of the mouse cursor.

If I know the value of ‘c’ then based on similar triangle property, I can compute ‘d’



Mouse – most commonly used

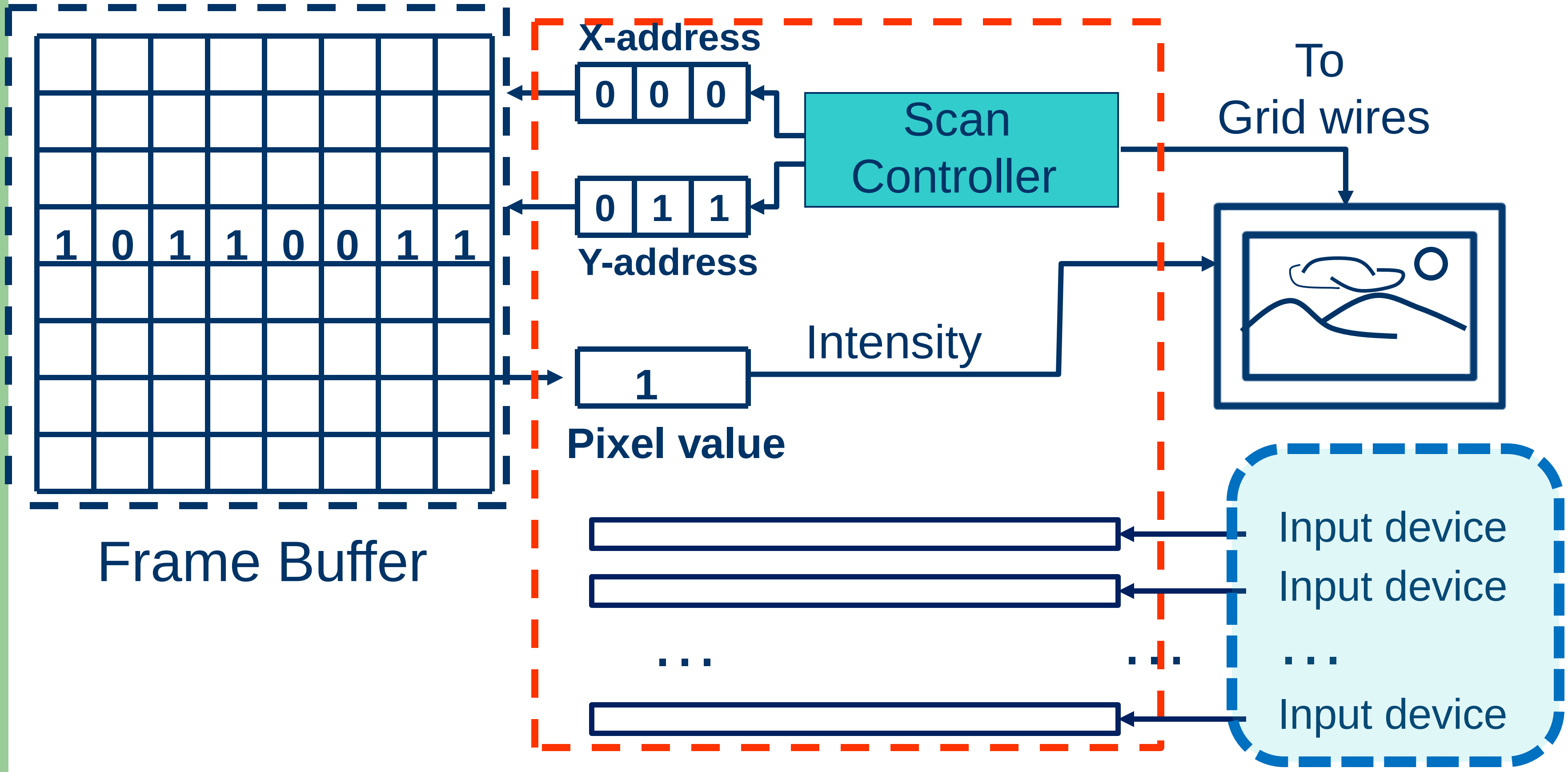
- a **relative device**, has no absolute origin, report only changes from their former position
- the application program can reposition the cursor anywhere on the screen



1.11 Input Modes

- Defined by the relationship between the measure process and the trigger
  - Measure: what the device returns to the user program
  - Trigger: a physical action on the device
- The display processing unit (DPU) contains a number of registers (buffers). Once initialized, input devices store appropriate values in these registers

Image Display System (DPU) (a simple two-color raster –scan system)



Input modes

- **Request mode:** application program requests input from a device, the graphics user interface returns control and the measure of the device only after the user has triggered the device
- Used with only one device at a time
- Application program cannot provide dynamic feedback, because application program does not regain control until the trigger action occurs

Input modes

- **Sample mode:** a single device is sampled, and the measure of the device is immediately returned. No trigger is needed
- Sequence of user inputs might be lost in sampling.

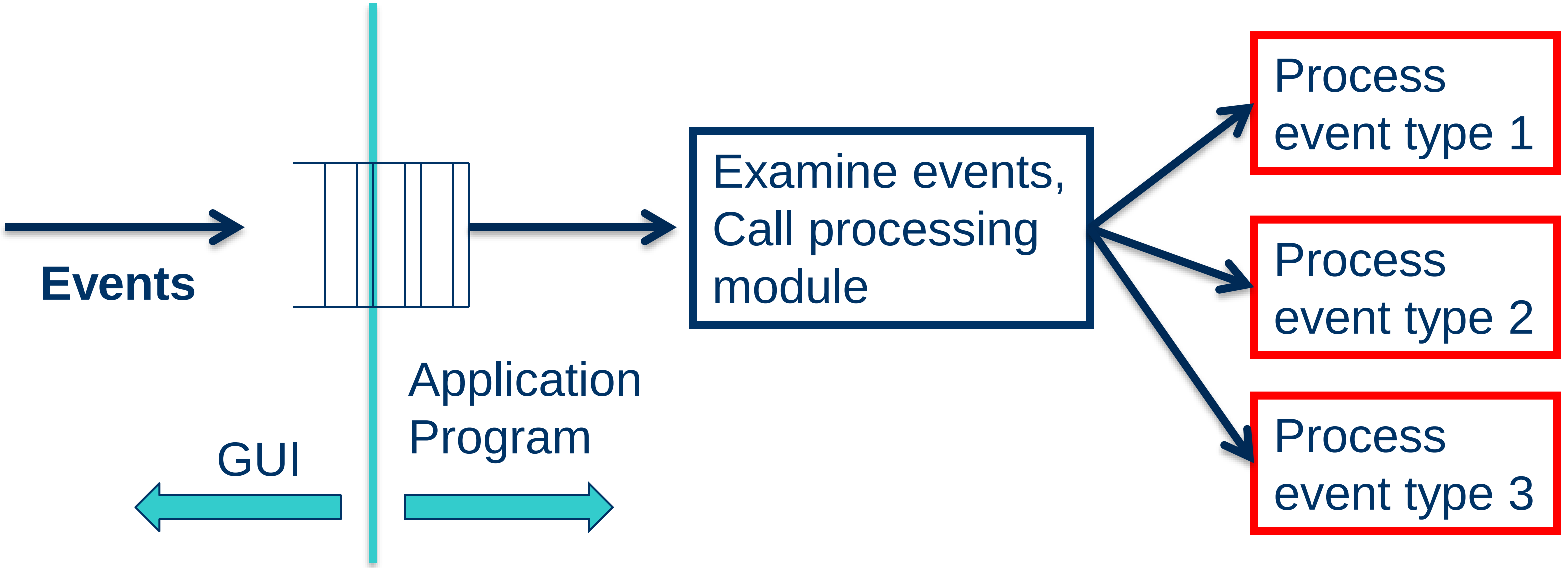
  Why?

- Well-suited for dynamic feedback from the application program

Input modes

- **Event mode:** when a device is triggered, the device measure with the identifier for the deivce is placed in an "event queue“ (but application program is not interrupted)
- Application program first enables all devices whose use is to be permitted
- Once enabled, a trigger action for any of them places an event report in an input queue, in order of occurrence

Input modes: event mode



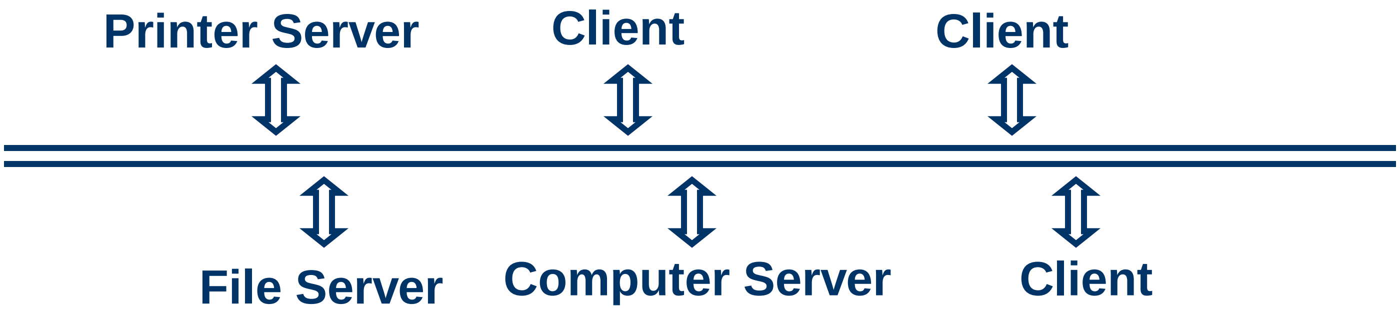
- more natural mode for systems with several independent processes and shared input devices
- typical for a graphical user interface (GUI) and is supported by the library.
- handles both hardware interrupts and software interrupts

1.12 Clients and Servers

- Primary motivation for the development of X Window System:

  "*do graphics over a network*“

- In a world of distributed computing and networks, building blocks are entities called "*server*"



Printer Server

Client

Client

File Server

Computer Server

Client

(Server: remote machine supporting client workstations)

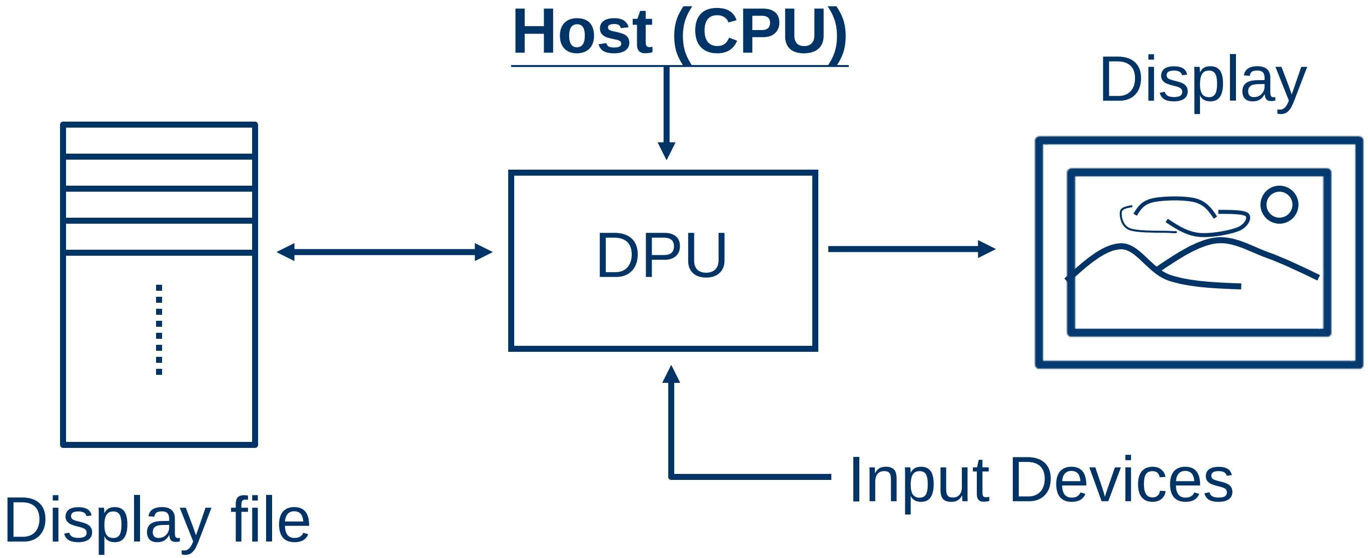
Clients and Servers

However, for X Window System & OpenGL

- **Server:** device that displays the graphics (machine in front of the user)
- **Client:** device that does computation (whatever machine running the application)

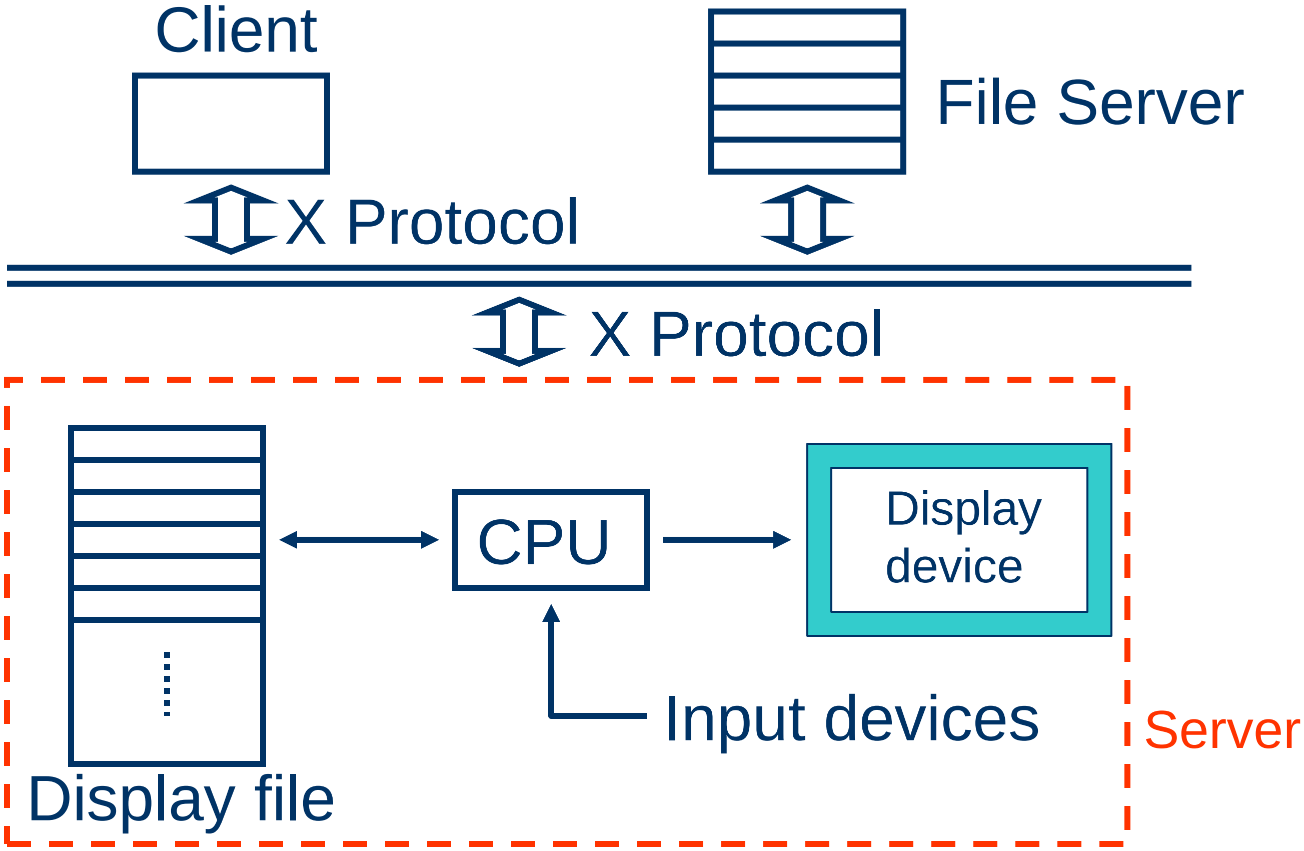
Concept of X Server

Then (Vector Display Device):



Concept of X Server

Now



End

Class website:

http://www.cs.uky.edu/~cheng/

http://www.cs.uky.edu/~cheng/cs535/CS535-HomePage-2022f.htm