



## Embedded Graphics

A glimpse through the steering wheel

Interior

# Hello....

## **Nishin K Vasu**

- › Core Discipline Software – Graphics – Instrumentation Division
- › Working since 07/2011 for Continental
- › Masters in Embedded System
- › Expert – Embedded Graphics
- › Interests – Puzzles, Shows, Learning, Cricket ..
- › Today, will try to explain what is Embedded Graphics, its concepts and what we do with respect to it

# Embedded Graphics- The Etymology

To simply put it

Combination of two terms → Embedded Systems + Computer Graphics

Let us take a peek at each of these terms

# Embedded Systems

- › Are also computers
- › Although are not for general use, but purpose built for their application
- › Designed to perform a set of dedicated tasks
- › Have constraints on processing power, memory, interfaces, battery ...
- › Examples include Home Appliances, Factory Automation, Pacemakers, MP3/ video players, Mobile phones... Distinction is getting blurry
- › Is at the heart of getting things smarter and connected (IoT)



Image Source: Google

# Computer Graphics

- › Technically can be defined as the process/ concept behind representing data on a screen / display
- › Needless to explain much, since it is part of our daily life through our phones, computers ...
- › Extensively used in simulations, Games, Movies, VR, AR ...

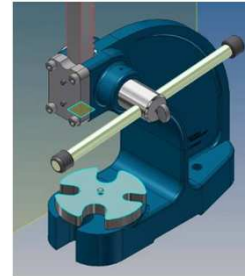


Image Source: Google  
Copyright: Marvel, Oculus

# Embedded Graphics



More on these lines ☺ -

- Not this level of sophistication!! (not yet at least)



Embedded Graphics – concept behind displaying information on the screen of an Embedded System

Source: YouTube  
CopyRight: Hyundai, CD PROJEKT RED

# Embedded Graphics – Evolution

- › For decades, User interface to an Embedded system was prominently mechanical buttons & some LEDs
- › Displays slowly crept in with Segmented LCDs (calculator-like), evolving into full color displays of varying size
- › At present, evolved to a level where mechanical components are quite comfortably shown on a display (e.g. Smart Watches)

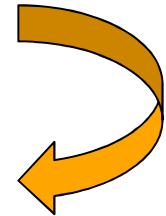
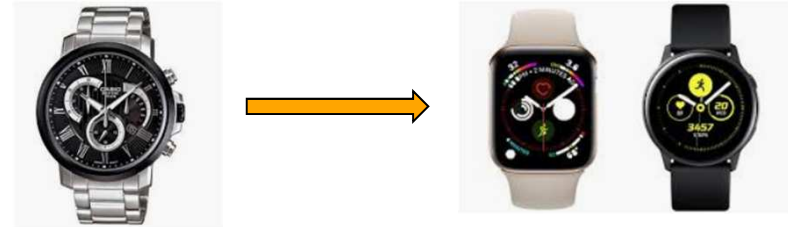
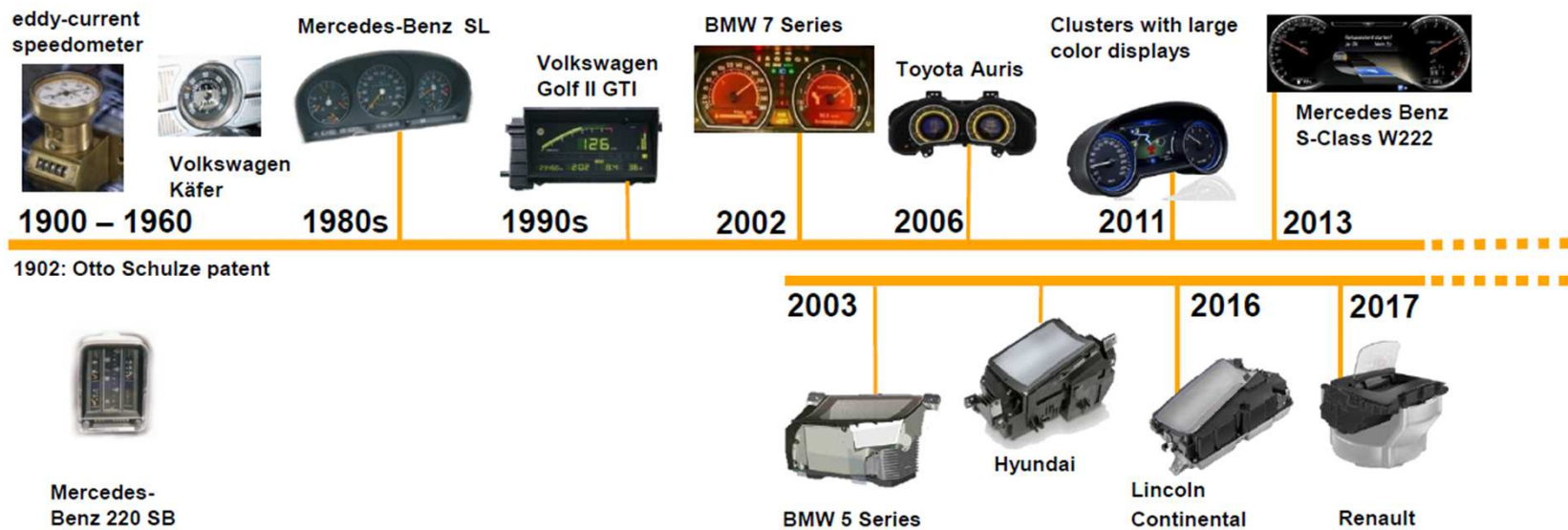


Image Source: Google  
Copyright: Apple, HP, CASIO

# Glimpse of the evolution through the Steering Wheel





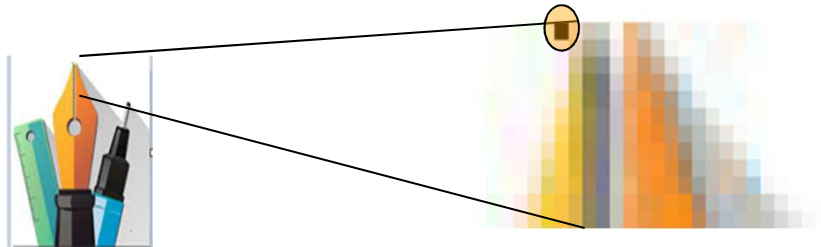
# Glimpse of the evolution through the Steering Wheel...



## Now for some Basics

### › Pixels (Picture Element)

- › Represents a single element / location of a digital image on a digital display or in memory
- › In memory combine to form the framebuffer / texture/ image data; follow a co-ordinate system and represented by certain no of bits / bytes
- › On a display depending on their size and their quantity define the quality of the display



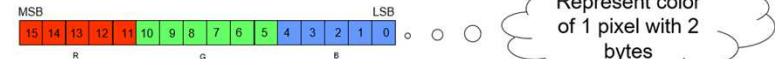
## Some Basics...

### › Bits Per Pixel & Color Format

- › Color Depth; Number of Bits used to represent one pixel
- › Color Format; usually combinations of Red, Green and Blue components with varying BPP – (RGB) + Transparency
- › Higher the BPP means better quality, but memory & processor intensive

Direct/True color formats:

RGB565 bits:



Represent color of 1 pixel with 2 bytes

Values for color components Red/Green/Blue directly given, range 0..31 for Red/Blue; range 0..63 for Green

RGBA8888 bits:



Represent color of 1 pixel with 4 bytes

Values for color components Red/Green/Blue directly given, range 0..255 for Red/Green/Blue; also range 0..255 for Alpha (Transparency per Pixel)



Source : Wikipedia

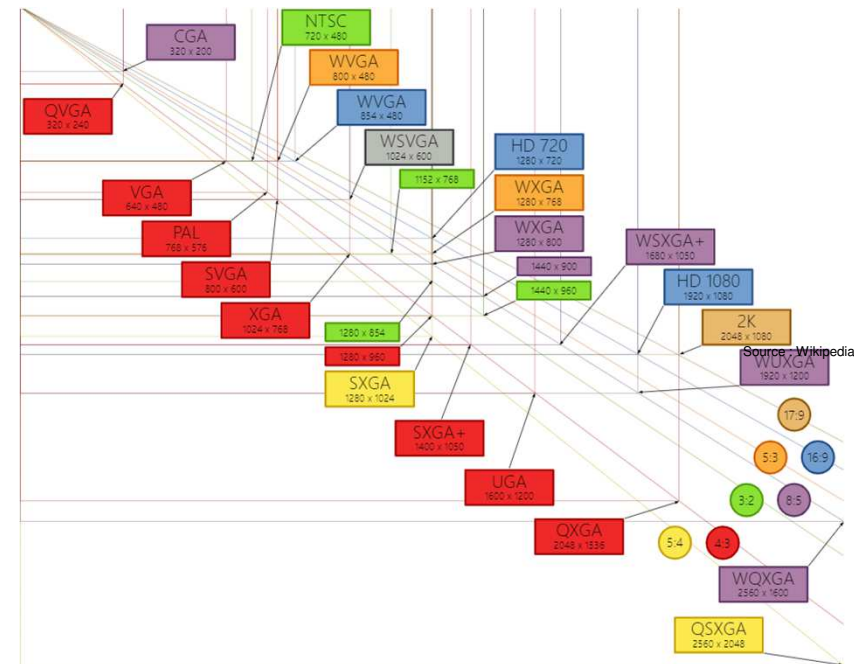
## Some More...

### > Resolution

- > No of Horizontal pixels x No of Vertical Pixels
- > Higher resolution - better quality, intensive processing

### > Framebuffer

- > Memory space where the final image(s) of the display is stored and retrieved
- > Size is a factor of the resolution, color format and bit-depth
- > Usually exists as double or triple buffered



## More...

- › Video Timing Signals
- › Blitting / Blending / Transparency
- › Compression
- › Color Look Up Table (CLUT)
- › Double / Triple buffering
- › Frame Rate
- › Refresh Rate
- › And .....

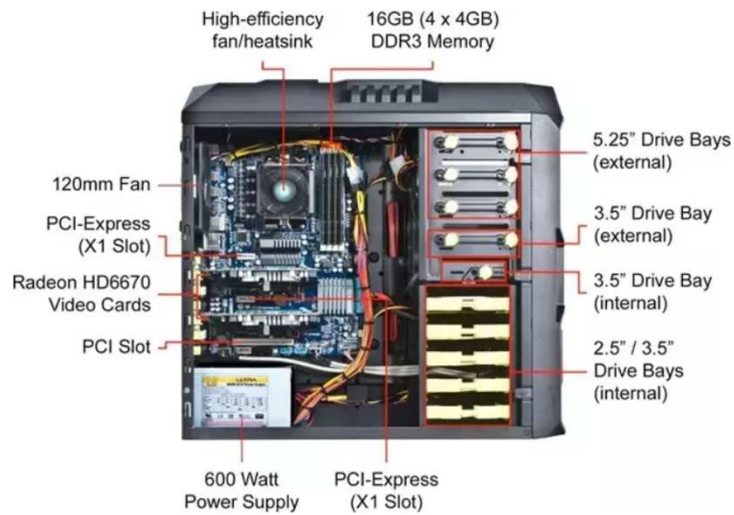
## A look inside

- › Take a closer look inside the cluster
- › Deal with 3 versions we saw earlier
  - › Monochrome – Gray-Scale
  - › Color display (3.5” – 7” display) + Mechanical pointers
  - › FDC – Fully Digital Cluster / Fully Programmable Cluster

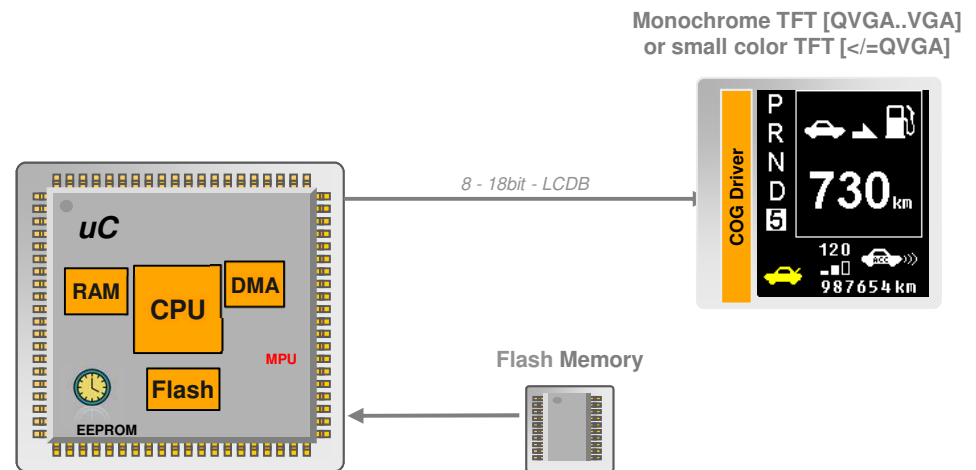


Image Source: WWW  
Copyright: VW

## The Monochrome / Gray-Scale Kind



Source: Quora; (Original Unknown)



# The Hybrid Kind



## Hybrid Instrument Cluster

Analog Meets Digital

Color TFT Display  
[QVGA..WVGA]





## The Full Digital Cluster Kind

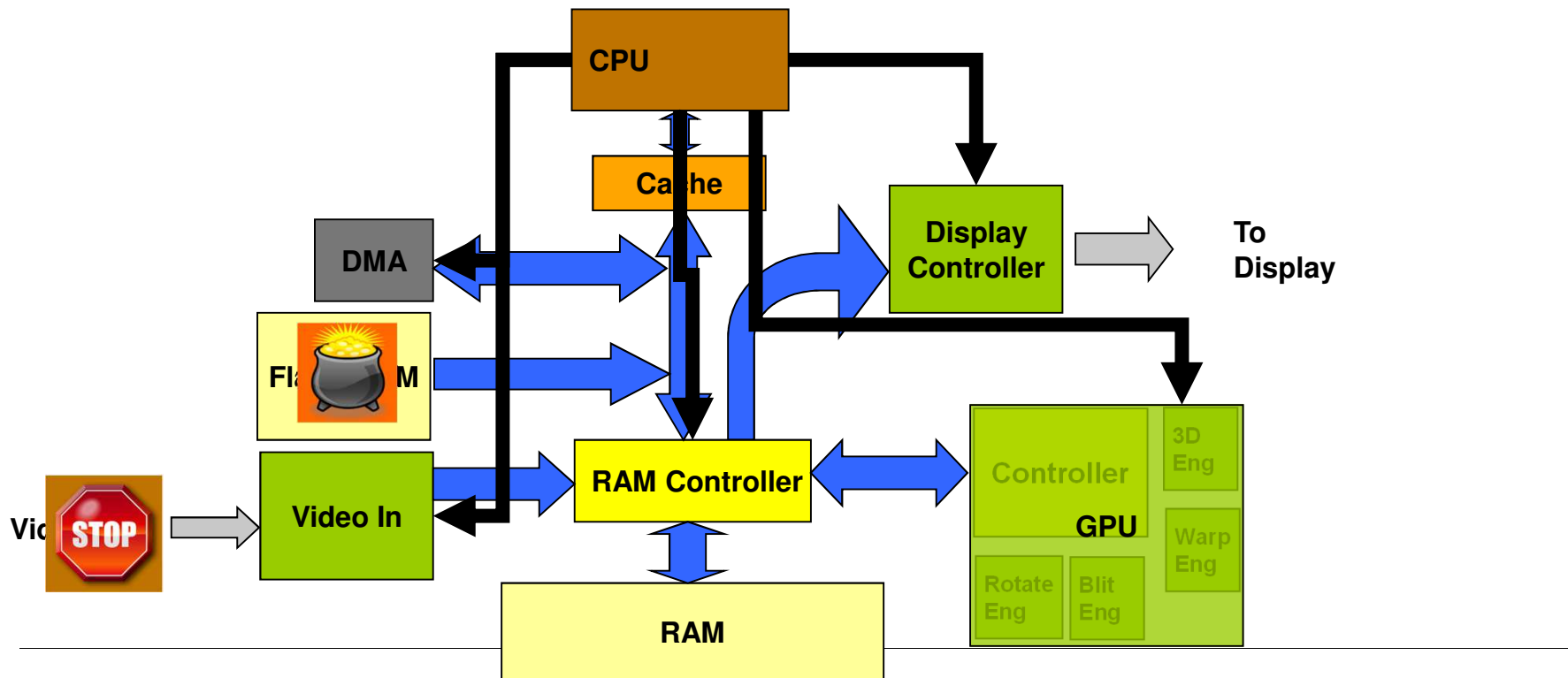


st. Video RAM

AND Flash

lash

# Typical Graphics Hardware Configuration / Data Flow



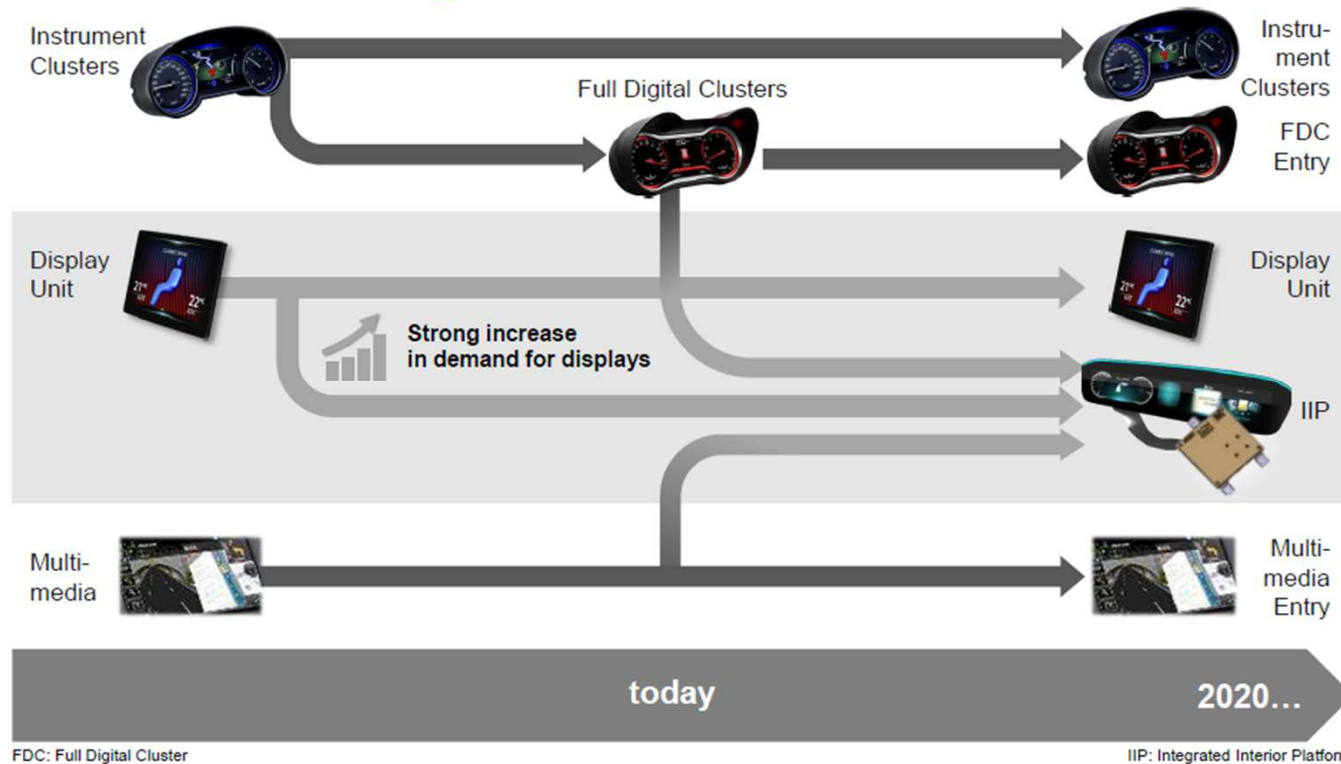
# Challenges

- › Memory – There is never enough of it!!
- › And when there is memory, then there are bottlenecks
- › The need to be as good as your tablet / phone
- › But hey, cannot afford a reset button here – A Safety Critical System
- › Quick Startup Times – Time by which cluster turns on
- › The more good and detailed they seem, the more complex they get; (contributed mainly through 3<sup>rd</sup> party framework, complex OS for e.g.: showing various Languages, 3D-Support etc.)



TeaPot Model Source: 3dexport.com  
OpenGL and Vulkan are Khronos trademark

## Despite the Challenges it continues to evolve...



## And evolve by..

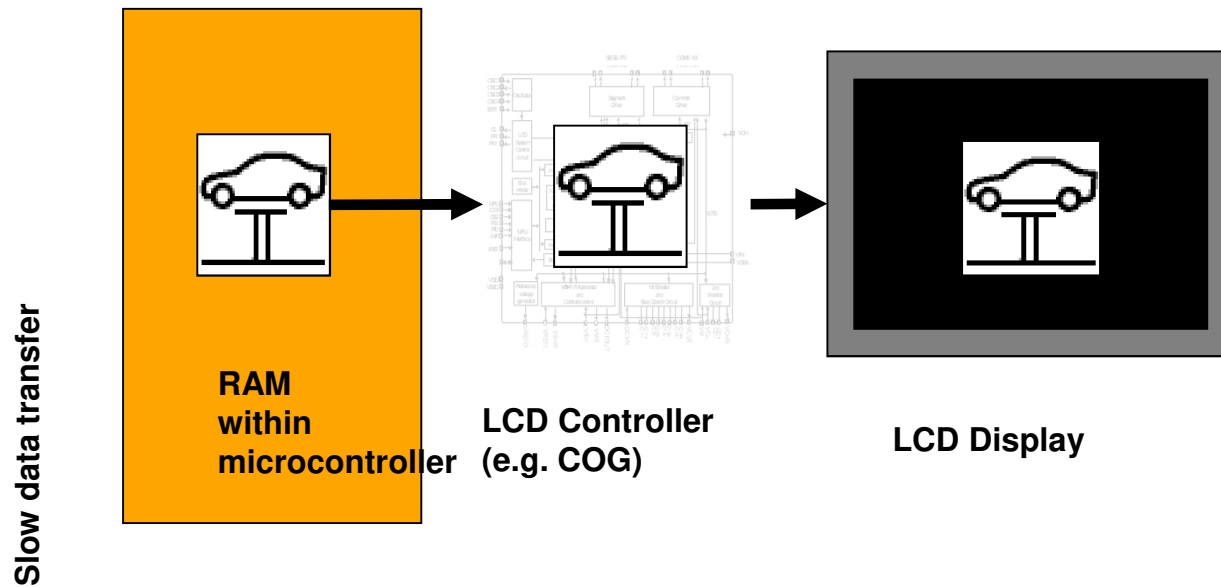
- › Facing the challenge – by learning, by adapting and by adopting



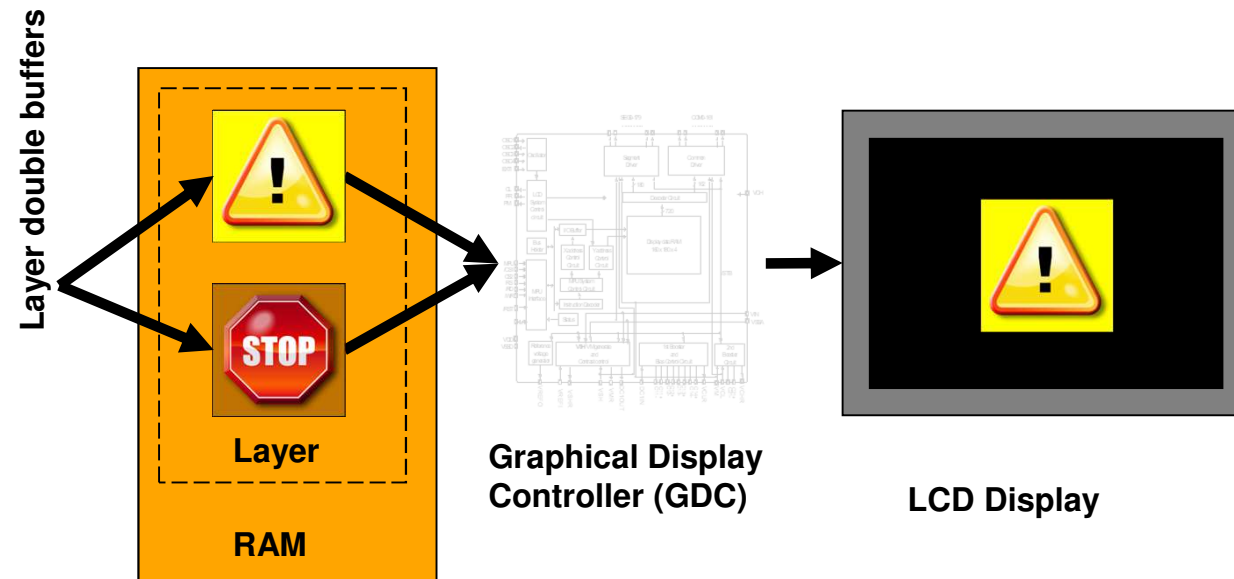
**Thank you for your  
attention!**

# Driving Black and White Display by COG driver

Transfer image to LCD controller memory



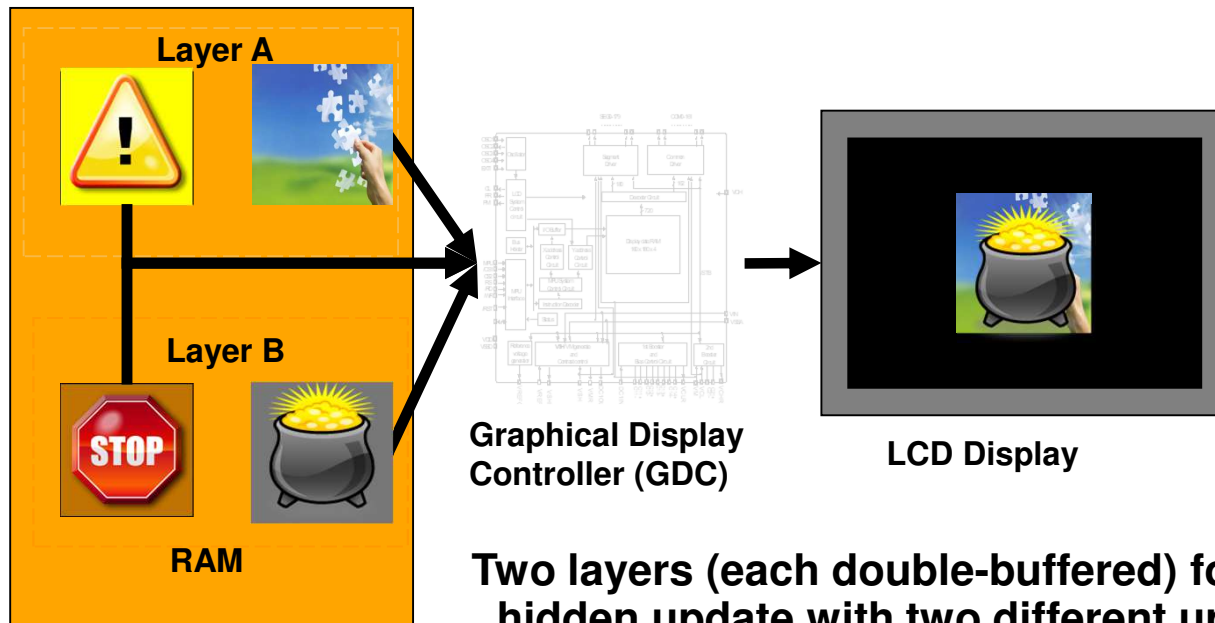
# Driving Color Display by Graphics Accelerator – single layer



**Double Buffering hides the update time completely**



# Driving Color Display by Graphics Accelerator – double layer



**Two layers (each double-buffered) for hidden update with two different update rates**

**– need 4 buffers, top normally with alpha**

## GA – BLiTer

- › Derived from “BLock Transfer” = the transfer of (rectangular blocks of) data within memory
- › BLiTer is a Coprocessor (= special DMA + blend unit) that performs BLT operations to image data independent of the CPU
- › Can perform different *Rendering* operations
- ▶ Three basic HW operations of the BLiTer

