TI Developer Conference
March 7-9, 2007 • Dallas, TX

Minds in Motion

Using the DSP in the Dual-Core DaVinci as a Graphics Render Engine

Soft graphic accelerator

Bastian Zuehlke

TES™ Electronic Solutions

Content

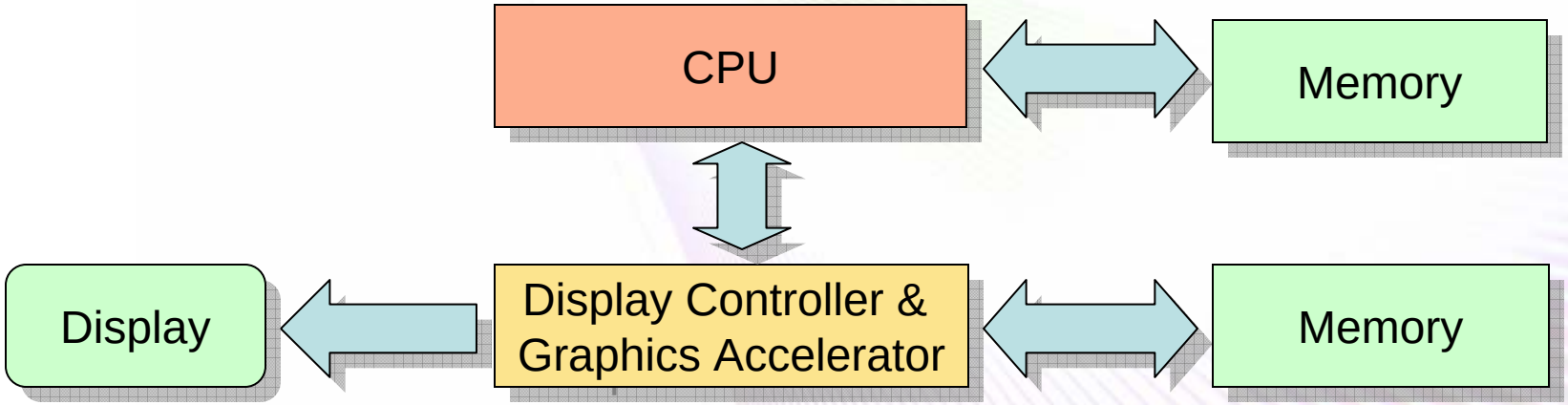
- Brief introduction into graphics
  - HW/SW System architecture
  - Graphic primitives, color formats, etc.
  - Antialiasing, blending, etc.
  - Render pipeline, rasterization, pixel pipeline
- Dynamic code generation
- Rasterization: Triangle
  - Classic / Explicit approach
  - Implicit approach
- Performance numbers
- Use Case: PMP Gui

Lead into graphics

- Examples

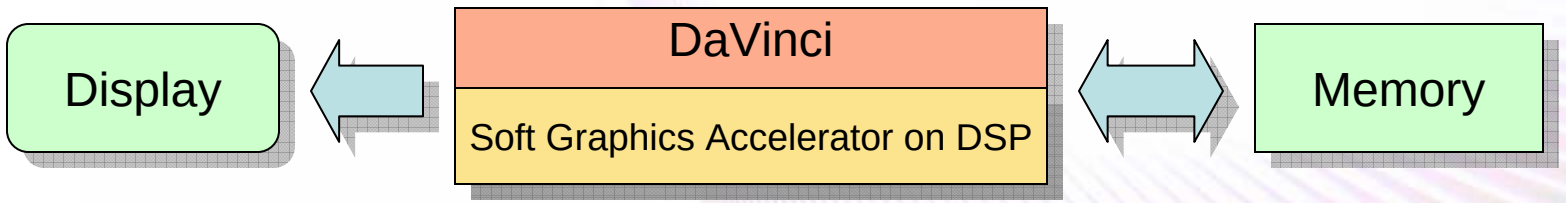
Lead into graphics

- Classic HW System Architecture
  - 4 Chip Solution



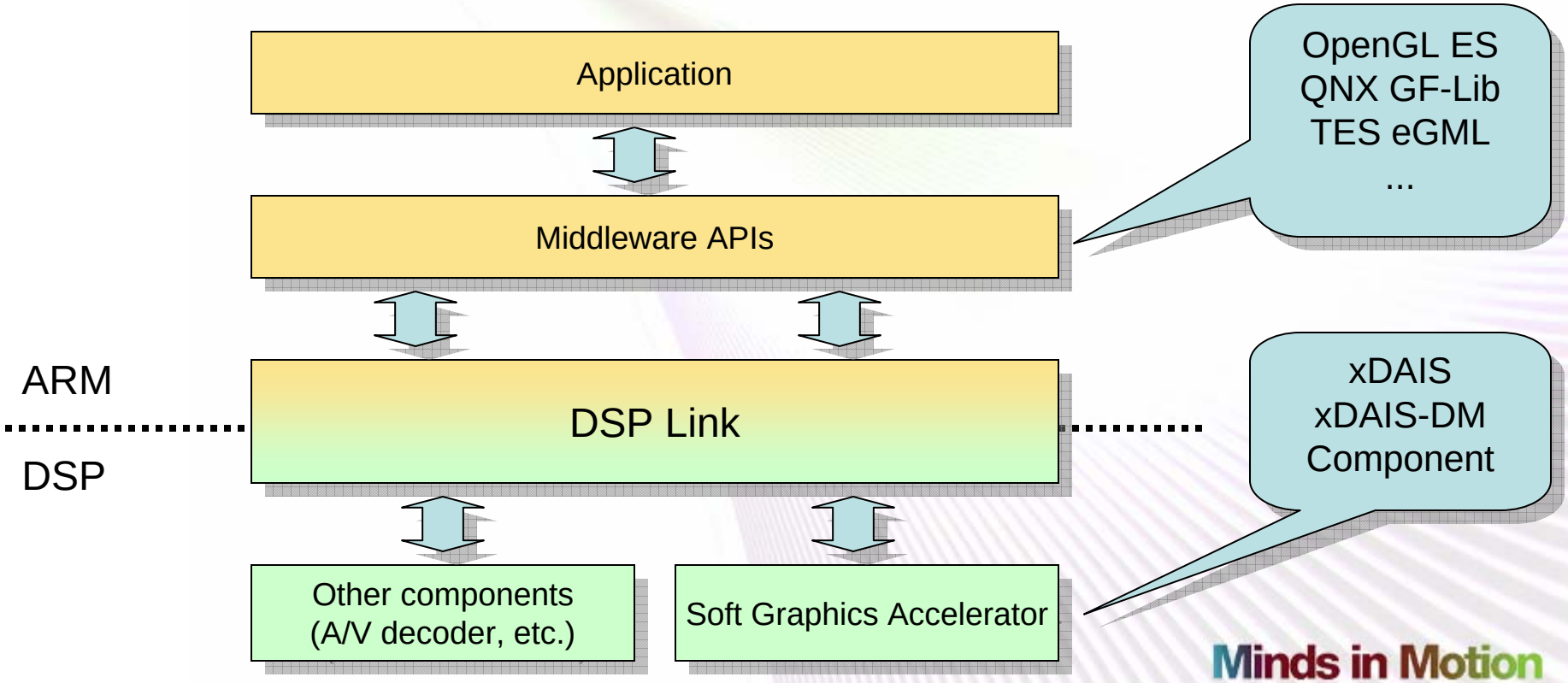
Lead into graphics

- DaVinci HW System Architecture
  - 2 Chip Solution
  - Easily adaptable to new requirements



Lead into graphics

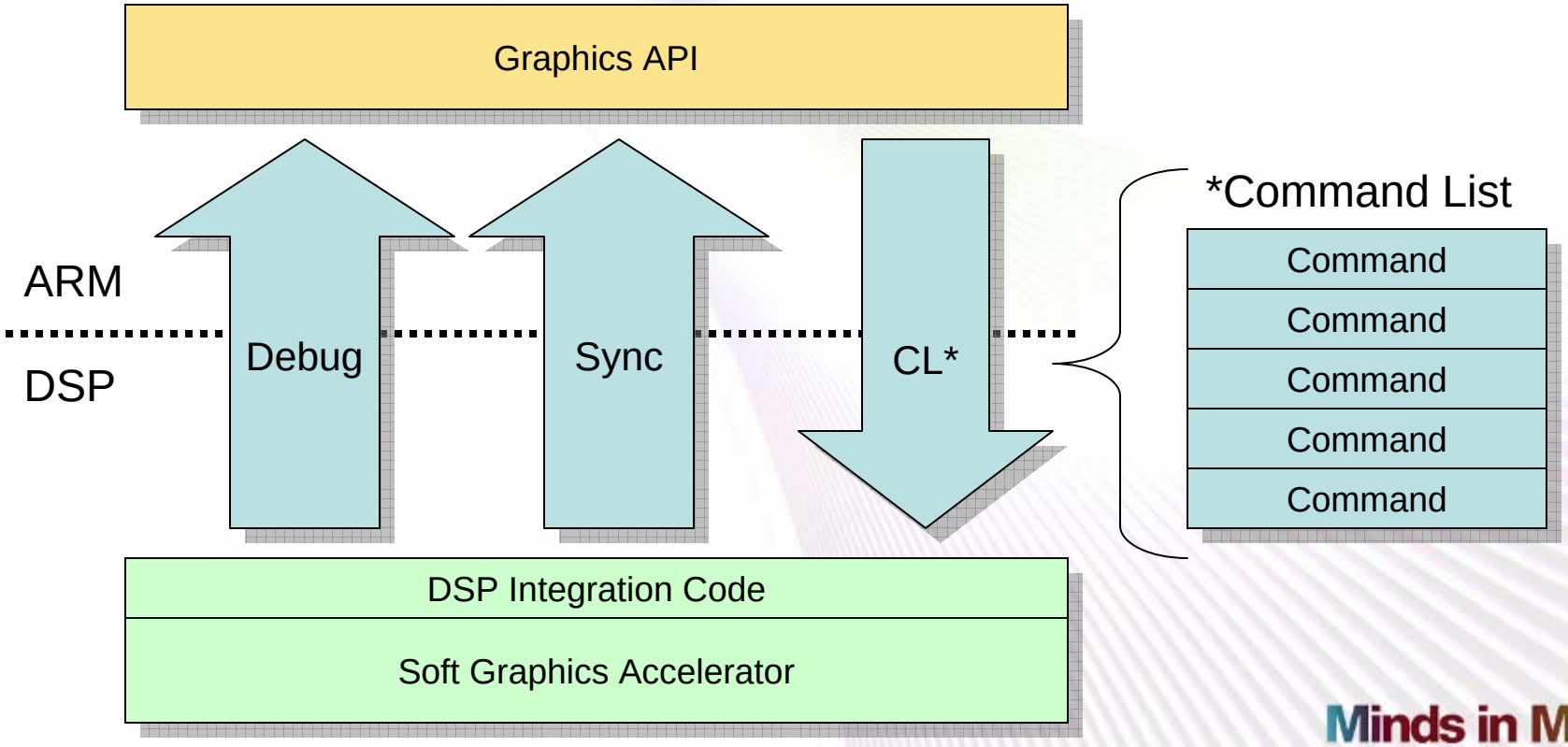
- DaVinci SW System Architecture



Minds in Motion

Lead into graphics

- Connection to Soft Graphics Accelerator



Lead into graphics

- Graphic primitives
  - Blit* operations
  - Lines
  - Triangles

*Block image transfer

Lead into graphics

- Color Formats
  - 32 Bit: ARGB8888, RGB888
  - 24 Bit: RGB888
  - 16 Bit: RGB565, ARGB1555, ARGB4444
  - 8 Bit: CLUT*

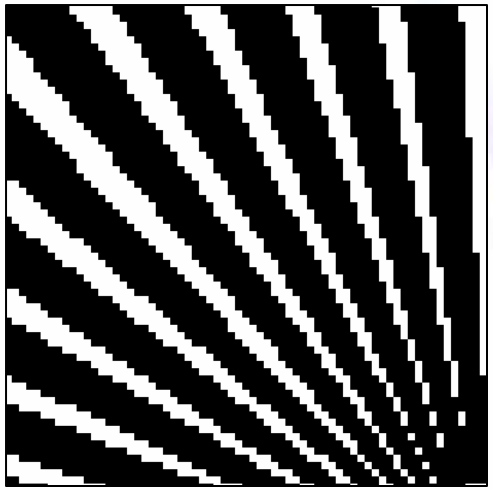
*Color Look Up Table

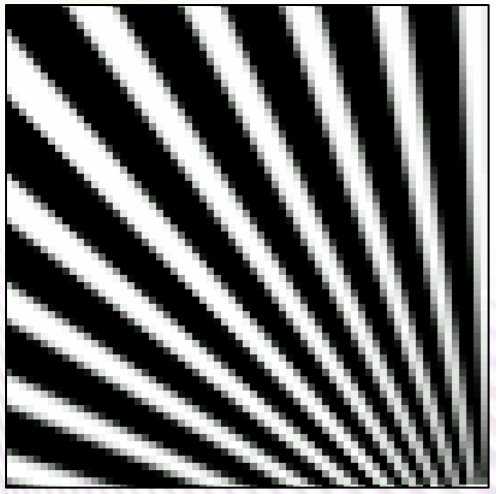
Lead into graphics

- Display Resolutions
  - QVGA: 320x240, WQVGA: 400x240
    - Mobil devices
    - In-Car-Nav-Radios
  - VGA: 640x480, WVGA: 800x480
    - NG PDAs, PNDs
    - In-Car-Multimedia-Systems

Lead into graphics

- Antialiasing
  - Post Filtering (Supersampling)
  - Pre Filtering



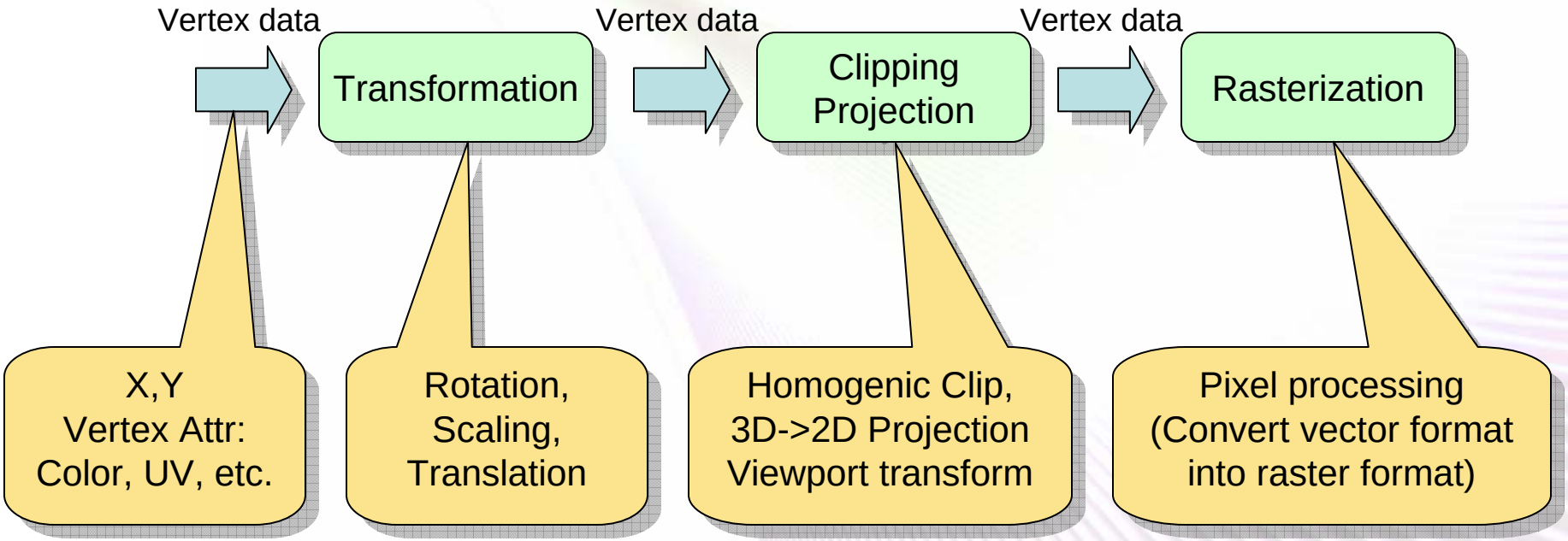


Lead into graphics

- Blendmodes
  - Blending combines the source color (Cs) with destination (framebuffer) color (Cd).
  - Examples:
    - Copy: C = Cs
    - Add: C = Cs + Cd
    - Multiply: C = Cs*Cd
    - Blend: C = Cs*As + Cd*(1-As)

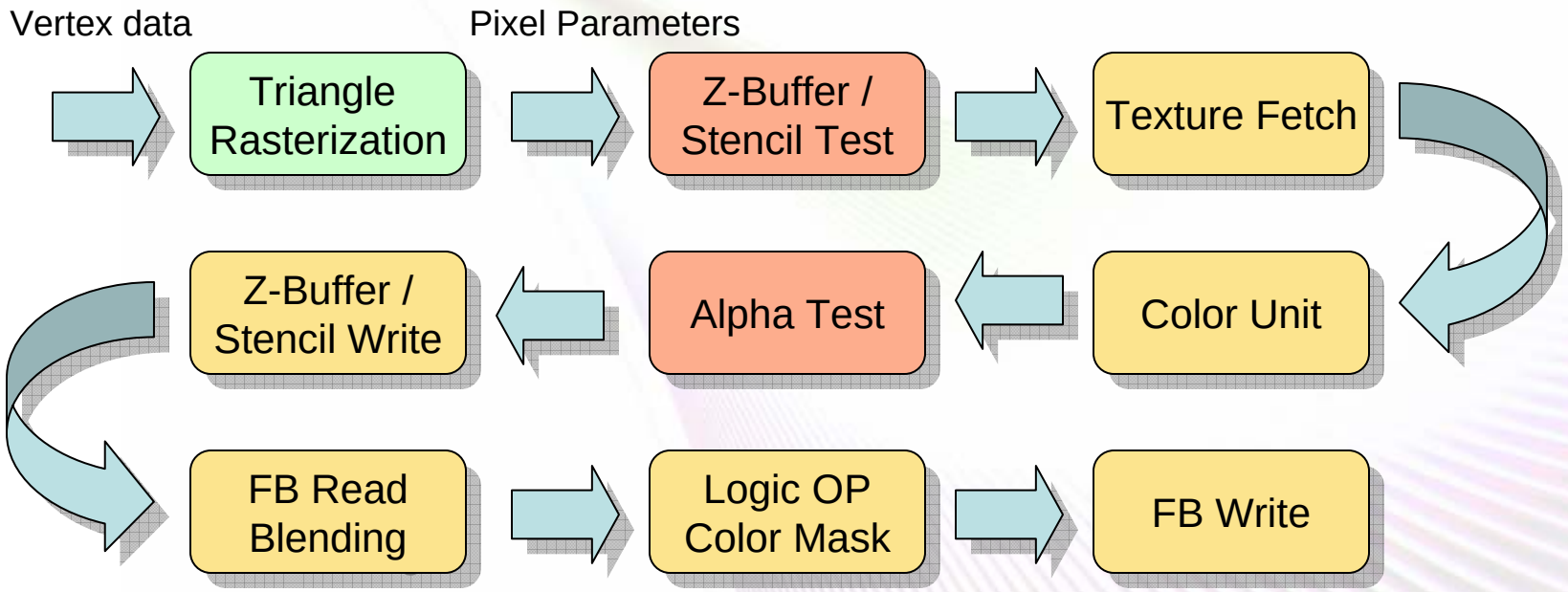
Lead into graphics

- Rendering-Pipeline



Lead into graphics

- Rasterization / Pixel Pipeline



Dynamic Code Generation

- Motivation
  - Pixel processing is the *Hot Spot*
  - Every cycle counts
  - Special code for each pixel process combination would give optimal performance

Dynamic Code Generation

- Combinations for solid fill Triangles
  - Framebuffer formats
    - RGB565, RGB888 (x2)
  - Blendmodes
    - Color Blendmodes (x4)
    - Alpha Blendmodes (x3)
  - Antialiasing
    - On/Off per Edge (x4)

Combinations
96

Dynamic Code Generation

- Combinations for solid fill triangles
  - Alpha Test (x9)
  - Stencil Test (x324)
  - Z-Buffer Test (x10)

Combinations

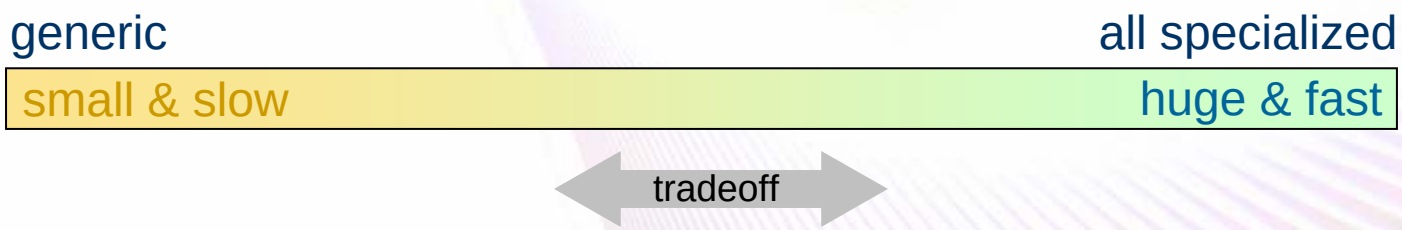
2799360

Dynamic Code Generation

- How to handle this many combinations?
  - Generic Code
    - Using a lot of *if* and *switch* statements does the job
    - It‘s easy to write and perfect to maintain
    - But makes pixel processing terribly slow
  - Specialized Code
    - Can get best performance in theory
    - Needs clever coding for automatic generation at compile time
    - But makes the memory footprint / compile time explode

Dynamic Code Generation

- Hybrid approach
  - Use C++ templates for codepath generation
  - Only small blocks are specialized
  - Function pointers used to call specialized code blocks

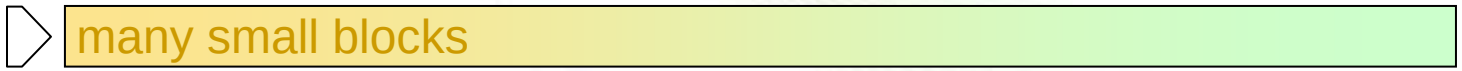


Dynamic Code Generation

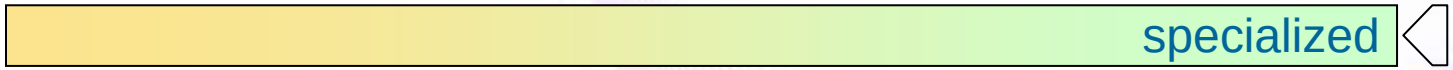
- Hybrid approach - disadvantages
  - New features push combinations much further
  - Find optimal balancing is difficult
  - DSP does not like code with a lot of jumps
  - Huge code size of library

Dynamic Code Generation

- Dyn. Codegen approach
  - Using many very small code blocks



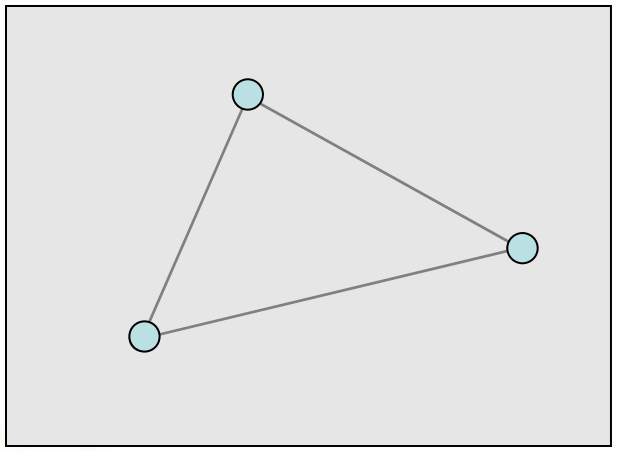
  - Build a specialized code at runtime

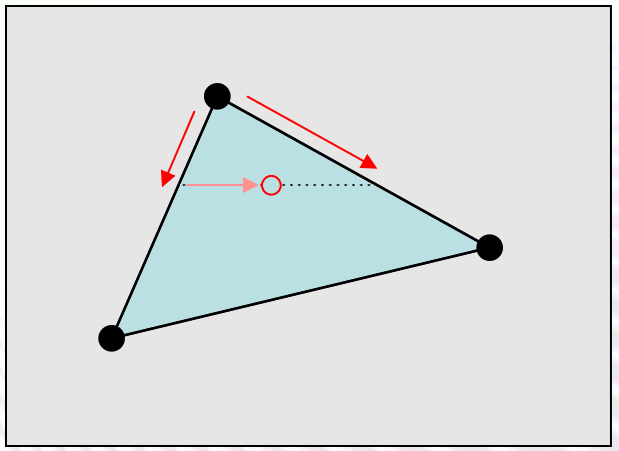


=> Result: Small code size and high speed

Rasterization

- Classic Approach
  - Geometry specified by vertices
  - Rasterization by interpolation
  - Suitable for GPP



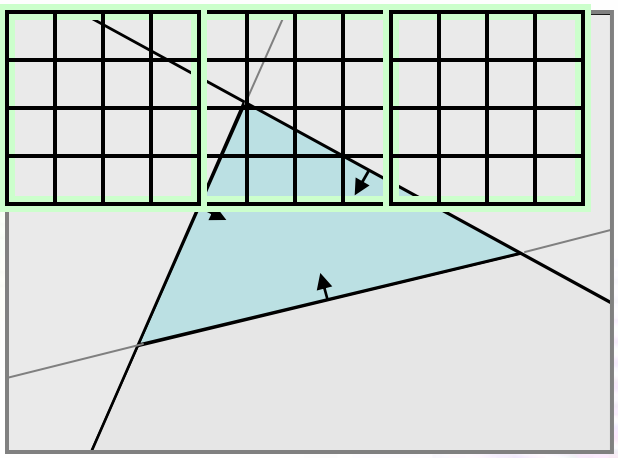


Rasterization

- Classic Approach – Disadvantages
  - Interpolation generates point samples
    - Difficult to antialias
  - Clipping explicit required
    - Causes artifacts at the border (jumping edges)

Rasterization

- Implicit approach
  - Splitting the plane into two halfs by an edge
  - Forming a region
  - Pixel inside if positive side of all edges
  - Allows blockwise rendering



Rasterization

- Implicit Approach – Advantages
  - Process any number of pixels in parallel
  - Clipping and tiling trivial because of implicit rendering
  - Pre Filtered Antialiasing becomes easy

Rasterization

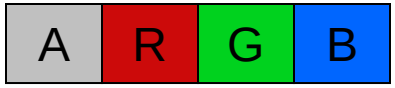
- Why suited for DSP?
  - Evaluate distance to each edge for every pixel
  - Only one addition/edge if done incrementally
  - DSP calculates 2 pixels of a triangle in one cycle
  - Setup is using only dot products, no division

Rasterization

- Why suited for DSP?
  - Pixels are examined blockwise
  - An entire 8x8 pixel block can be tested to be inside/outside the triangle with just 3 cmp's
  - DSP can fill 'fully inside' 8x8 blocks very fast issuing two 64 bit stores/cycle

Rasterization

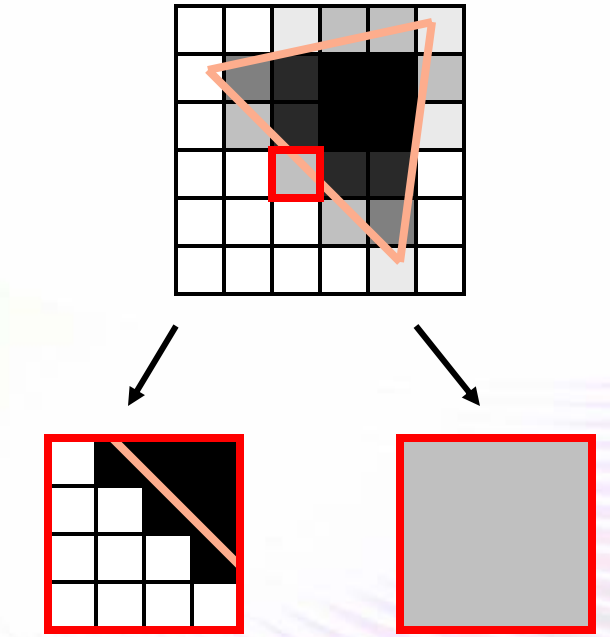
- Why suited for DSP?
  - Blending has to be performed on 4 color channels in parallel



  - DSP can operate on packed 8bit quantities directly

Rasterization

- Direct edge antialiasing
  - Requires pixel coverage (PC) value (Range: 0-1)
  - Blend source and test color with pixel coverage C=Cs*PC+Cd(1-PC)
  - PC calculation
    - Multi-sampling (number of hits/number of samples)
    - Distance field (distance to edge)



Performance numbers

- System description
  - DSP: c64p
  - DSP Clock: 567 [MHz]
  - Memory Clock: 187 [MHz]
  - Memory Type: DDR

Performance numbers

- Fill (RGB565)
  - 640x200 -> 1.22 ms -> 98.1 mpix/s
- Small Line (RGB565)
  - Len: 6, Width: 1, AA=Off -> 412000 lines/s
  - Len: 6, Width: 1, AA=On -> 333000 lines/s
- Small Triangle (RGB565)
  - (10,10)(13,10)(10,13) -> 210000 triangles/s

Performance numbers

- Blit Copy (RGB565->RGB565)
  - 800x480 -> 5.1 ms -> 74.5 mpix/s
- Blit Const Alpha (RGB565->RGB565)
  - 800x480 -> 8.8 ms -> 43.6 mpix/s
- Blit Pixel Alpha (RGB565+A8,RGB565)
  - 800x480 -> 9.3 ms -> 41.2 mpix/s
- Blit Color (A8,RGB565)
  - 800x480 -> 6.6 ms -> 58.1 mpix/s

Use Case: PMP Gui

- Requirements:
  - RGB565, 480x240
  - 100ms refresh (10 FPS)
  - Blitoperations:
    - 480x240 Background (Blit Copy)
    - 4x64x64 Animated Icons (Blit Pixel Alpha)
    - 64x32x32 Text / 1 Color Icons (Blit Color)

Use Case: PMP Gui

- Performance Requirements:
  - Blit copy
    - 1.53 mpix/s -> 1.8 mpix/s -> 22.5 ms
  - Blit Pixel Alpha
    - 0.164 mpix/s -> 0.2 mpix/s -> 5 ms
  - Blit Color
    - 0.656 mpix/s -> 0.8 mpix/s -> 15 ms

=> 42.5 ms ~ 5% DSP Load ~ 30 MHz

TI Developer Conference
March 7-9, 2007 • Dallas, TX

Using the DSP in the Dual-Core DaVinci as a Graphics Render Engine

Soft graphic accelerator

More information:

- **Meeting room #3184**
- **www.tesbv.com**

Minds in Motion

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
Low Power Wireless	www.ti.com/lpw	Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265