FUJITSU

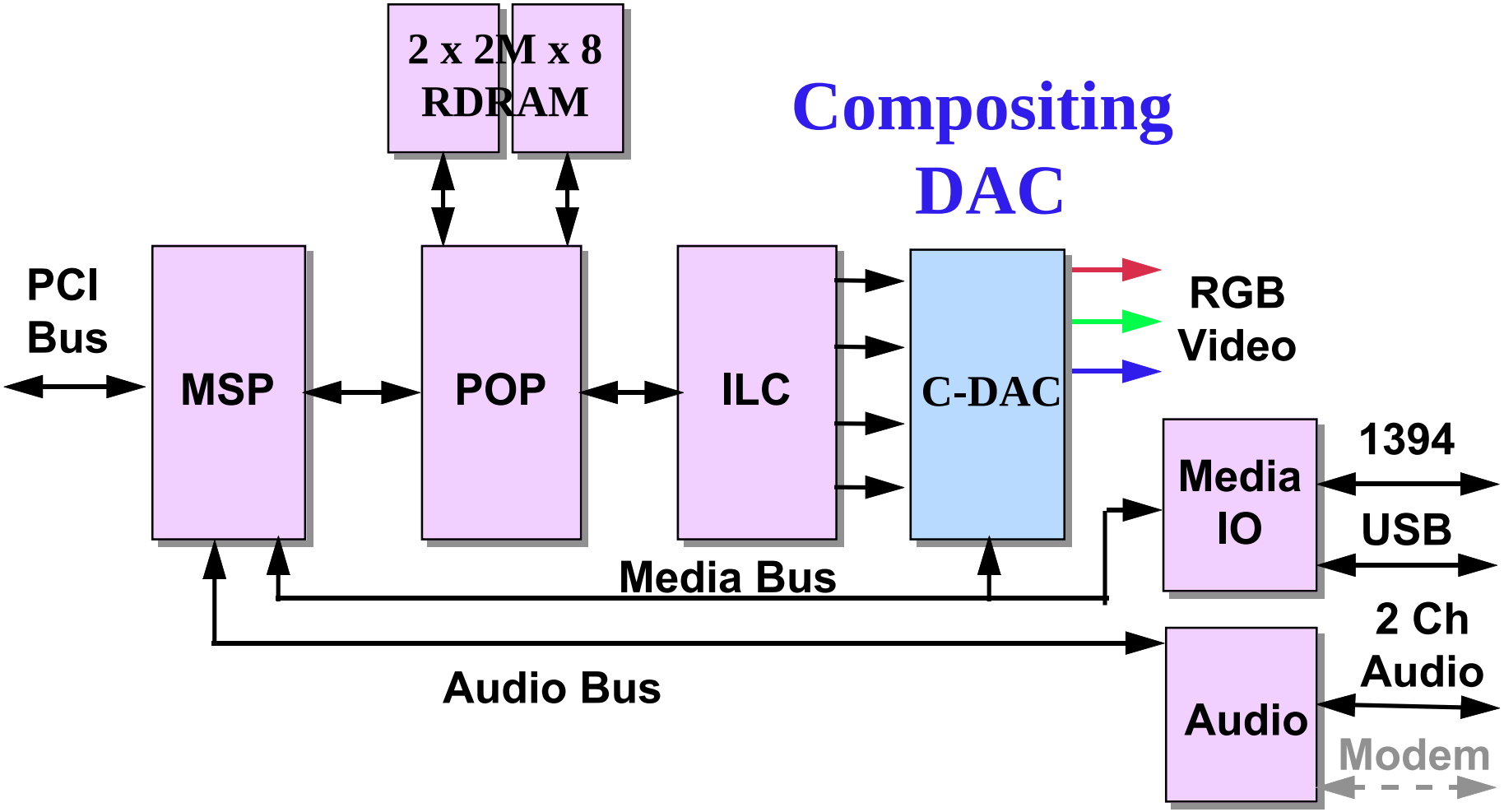
Custom VLSI for the Compositing DAC of the Touchstone Multimedia Accelerator

Introduction

- **Compositing DAC (C-DAC)**
  - One of five related ASIC Components
  - Advanced Multimedia Architecture
  - Hot Chips Code Name: Touchstone

FUJITSU

Touchstone Multimedia Architecture



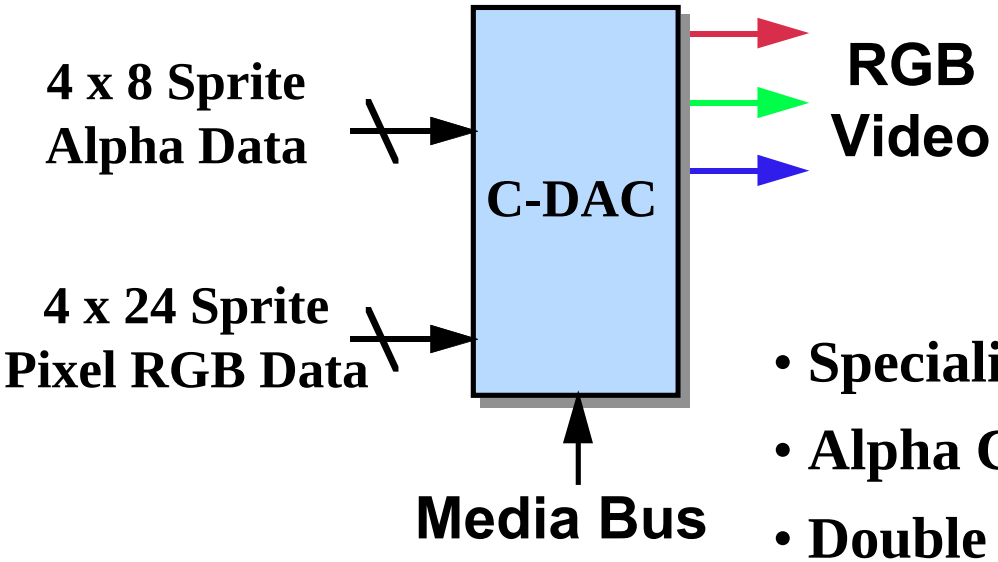
FUJITSU

C-DAC Chip Contribution

- **Touchstone Function and Performance**
  - 1344 x 1024 @ 75Hz
  - 24 bit true color at all resolutions
  - Optimized 3D Animation at 75 Hz - Combination of of Image Layer Animation and 3D Rendering
  - Scene Complexity of >20,000 Rendered Polygons
  - Very High Image Quality - anisotropic filtering, subpixel anti-aliasing, shadows, reflections, etc.
- **Cost Goals**
  - PCI Add-in Card for Less Than $ 250 BOM Cost
  - Compositing DAC Targeted Cost < $35

FUJITSU

Touchstone Compositing DAC



- Specialized Memory & Logic Device
- Alpha Compositing of Sprite Data
- Double Buffered Scanlines at 1344 x 32
- Processes 4 Pixels @ 32bpp @ 80MHz
- Compositing Rate: 320MPixels/sec
- Integrated DAC
- Standard RGB Output with Stereo

FUJITSU

Compositing DAC Vital Statistics

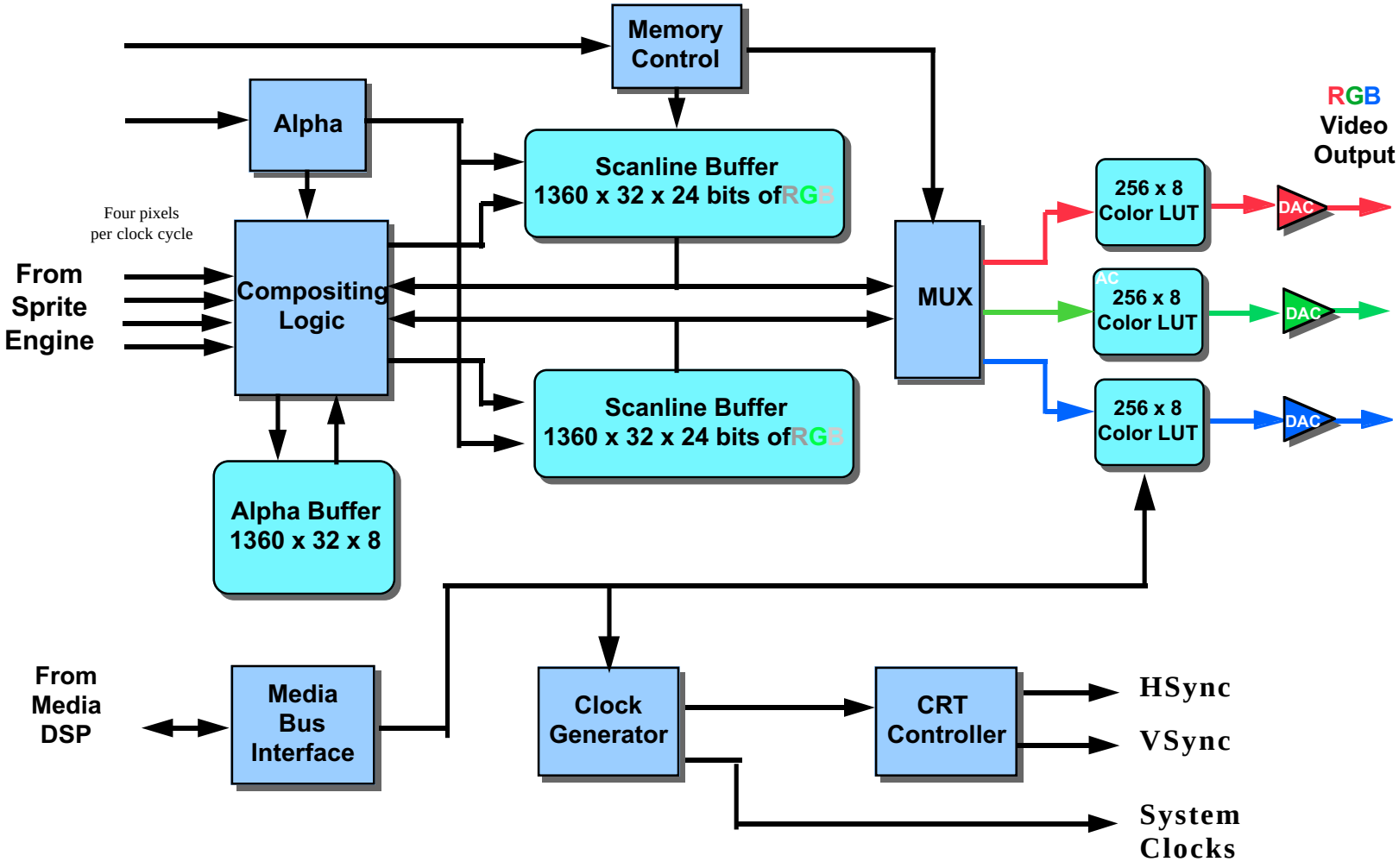
- **Package: SQFP**
- **# Pins: 304**
- **Power: 2.5 W**
- **Logic Transistors: ~ 350K**
- **Memory Transisitors: ~ 15M**
- **Process Technology: 0.35 Micron C-MOS**

FUJITSU

C-DAC Challenges

- **Potpourri of Circuitry**
  - Logic, Memory, and Analog
- **Price Point**
  - SRAM based initial design to minimize risk
  - Rapid DRAM evolution required to meet consumer product cost goals
- **Schedule**
  - First Engineering Samples in 12 months

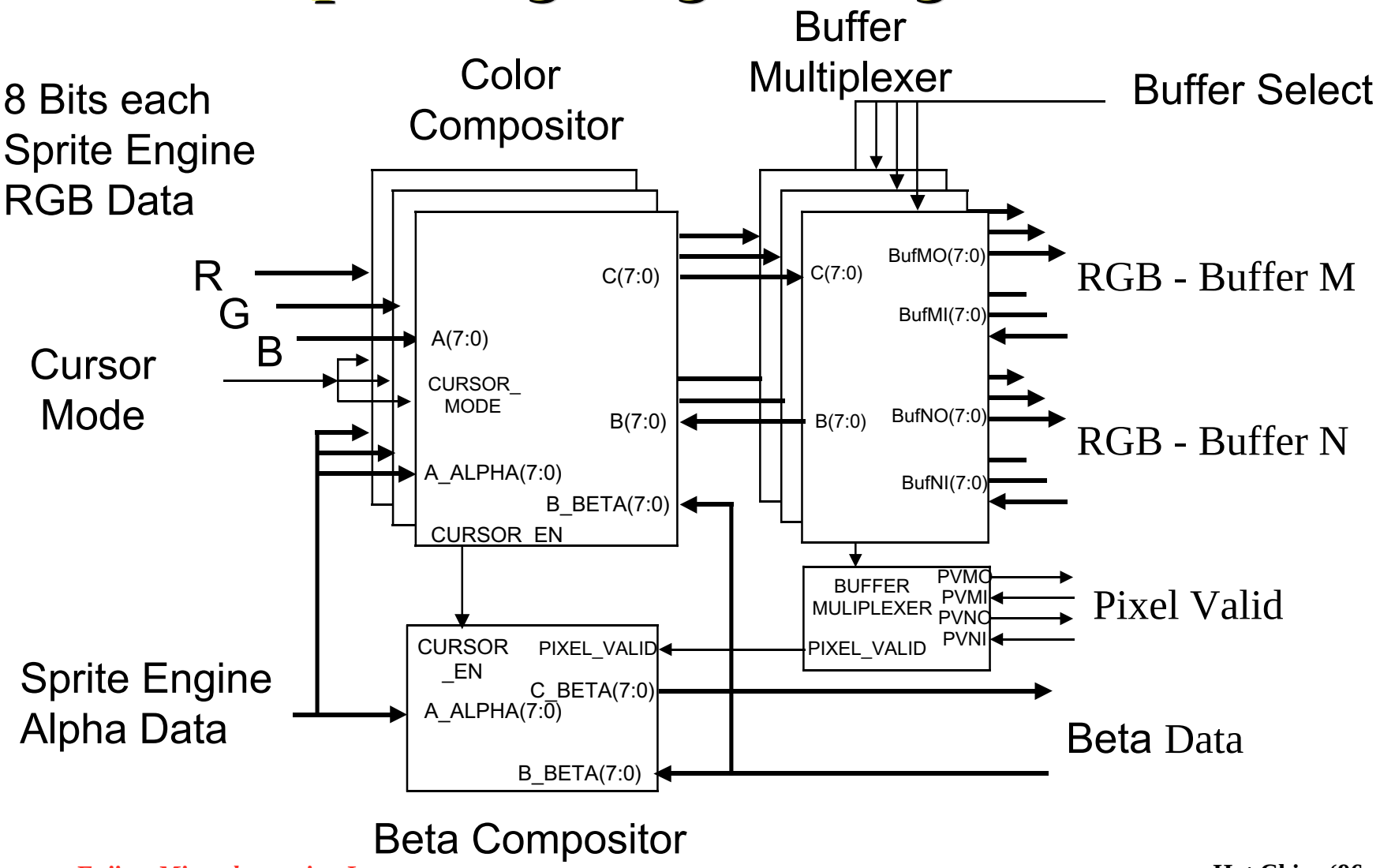
Compositing Buffer and DAC



Compositing DAC - Five Major Blocks

1. **Compositing Engine**
2. **Compositing (Color and Beta) Buffers**
3. **135 MHz LUT-DAC**
4. **16-bit Media Bus Interface**
5. **CRT Controller & Clock Generator**

Compositing Engine Logic



Compositor Logic

1. Normal Mode

The Compositor Logic Performs the functions:

$$C = B + B\_BETA * A$$

$$C\_BETA = B\_BETA* (1- A\_ALPHA)$$

-or-

$$C\_BETA = B\_BETA * A\_BETA$$

Compositor Logic

2. Cursor Mode

- **Incoming pixel is a cursor pixel.**
- **XORed with stored data to produce a cursor**
- **Stored alpha is unaltered**

The equation is:

If A_ALPHA = 0x01,

C = NOT(B)

C_BETA = B_BETA

ELSE

C = B + B_BETA * A

C_BETA = B_BETA * A_BETA

END IF

Compositor Logic

where:

A = Color of Incoming Pixel (From Sprite Engine)

A_ALPHA = Coverage (Alpha) of Incoming Pixel (from Sprite Engine)

B = Stored Color (from Buffer)

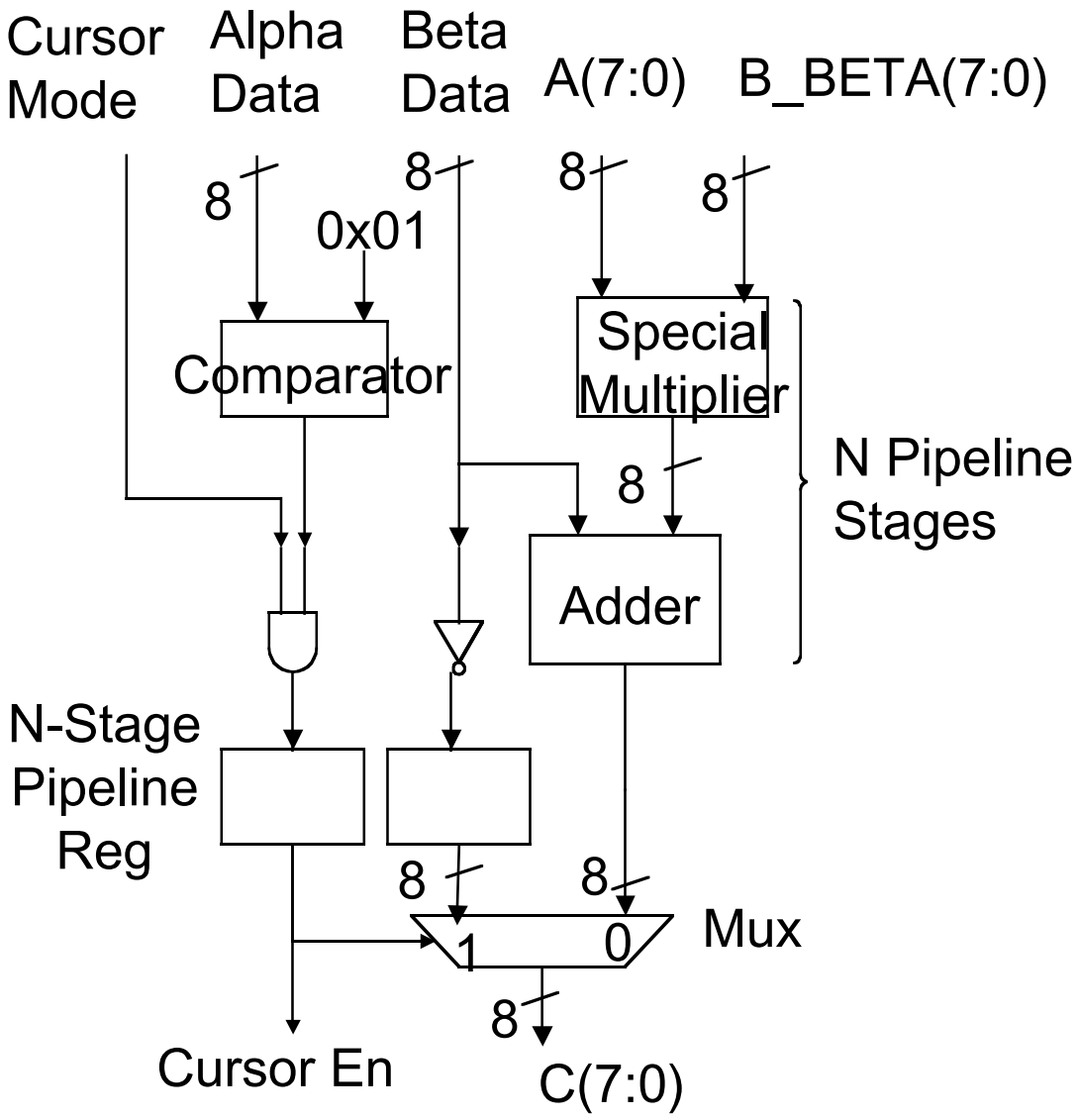
B_BETA = Stored Transparency (Beta from Buffer: Beta = 1-Alpha)

C = Resultant Pixel Color (new value to write back into buffer)

C_BETA = Resultant Pixel Transparency (New Beta to write back into buffer: Beta = 1 - Alpha)

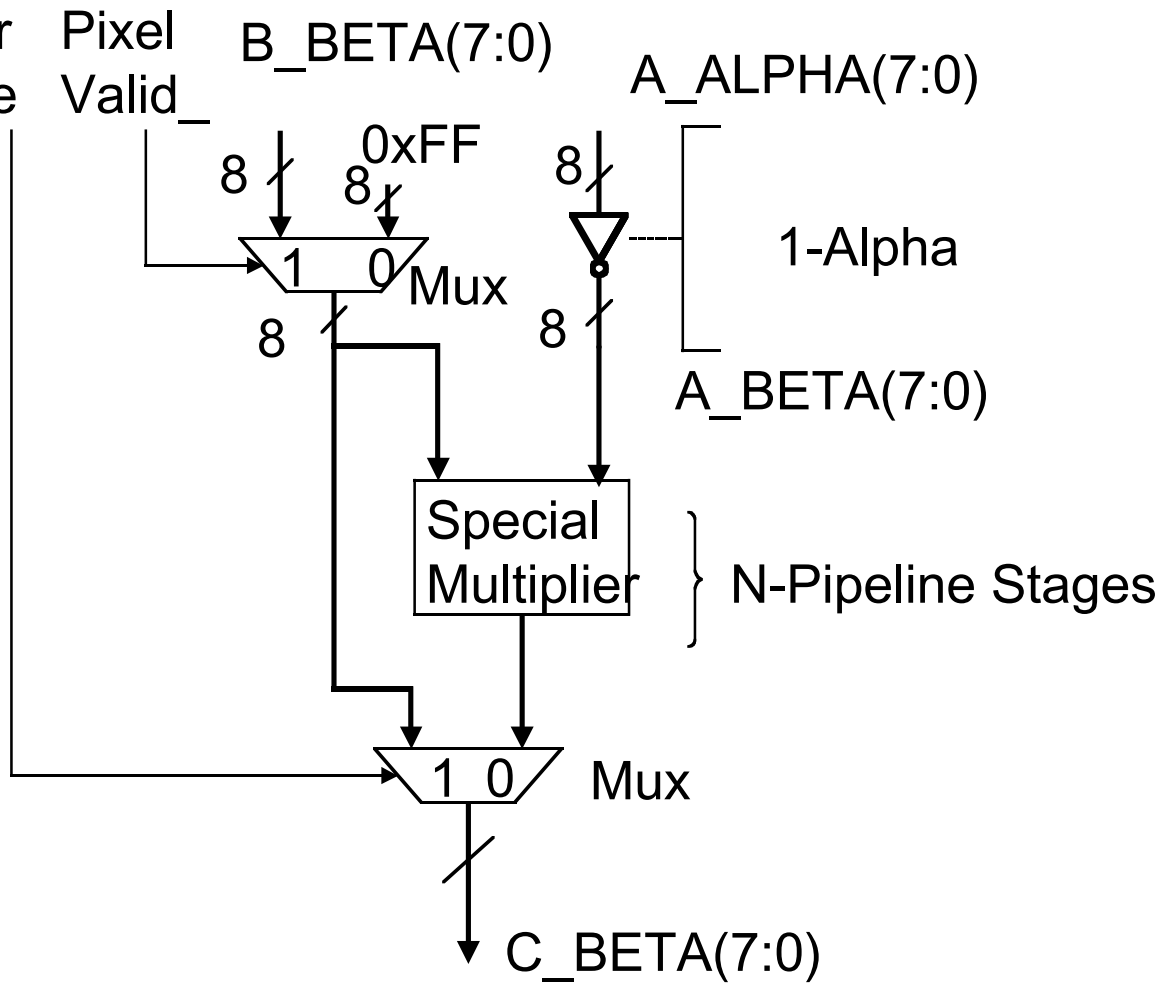
Color Composition Logic

- One Pixel, One Channel
- Repeat 12 Times -
  4 Pixels, 3 Channels



Beta Composition Logic

- One Pixel, One Channel
- Repeat 4 Times - for 4 Pixels



FUJITSU

Color Buffers

Ping-pong buffers for each of the 8-bit RGB Channels

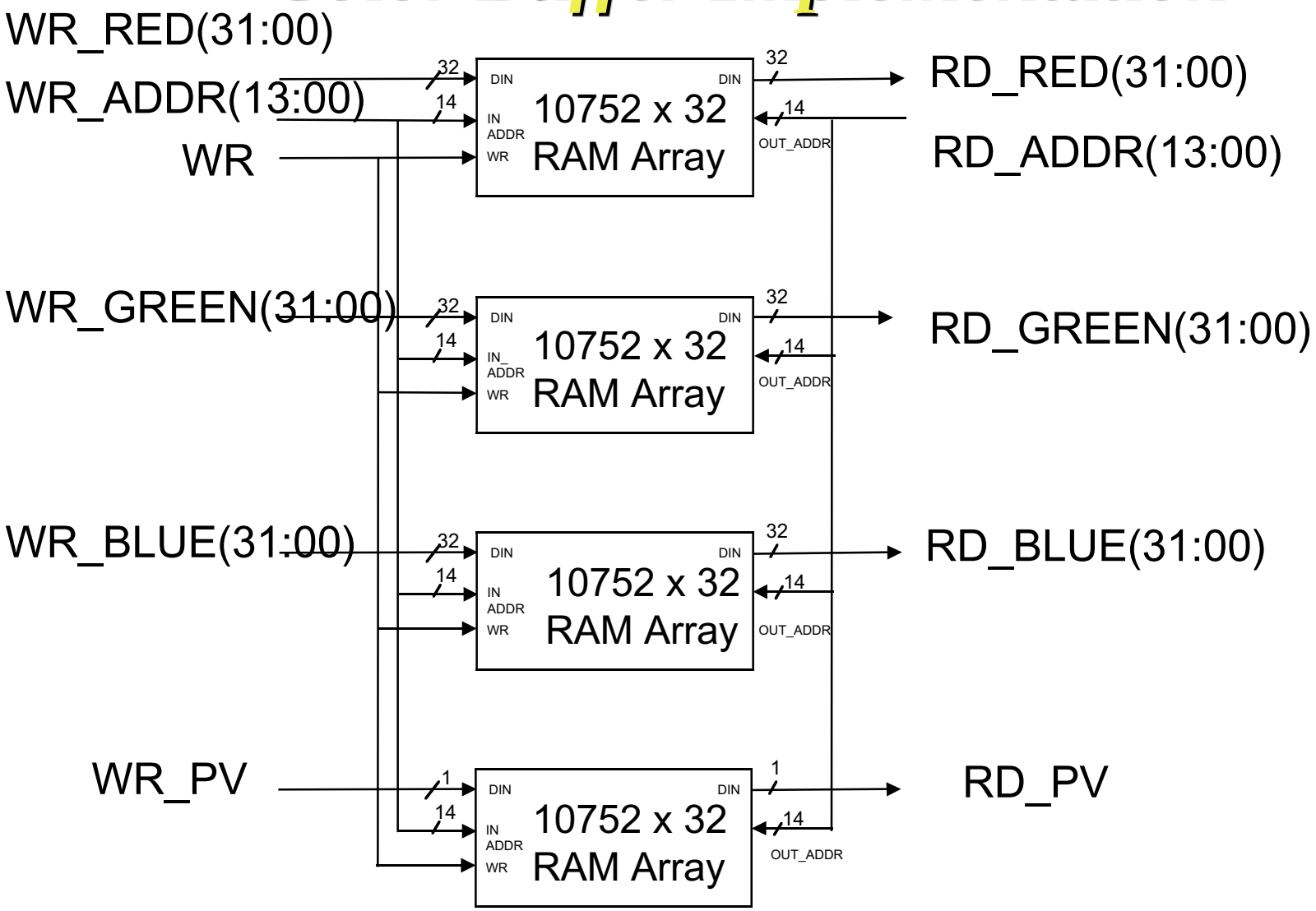
- Compositing in one of the Color Buffers
- Scanline-based read out of the other

- **Two separate 10752 x 32-bit Dual Ported RAM arrays**
  - Three instances of the Color Buffer for RGB
- **Two 10752 x 1-bit Dual Ported RAM arrays for Pixel Valid**
- **Arrays contain a read port and a write port**
  - Separate addresses generated for each port
- **Self refresh**

Challenge:

- Send pixels in an uninterrupted fashion to display
- Up to 1344 sequential row pixels MUST be read out at pixel clock

Color Buffer Implementation

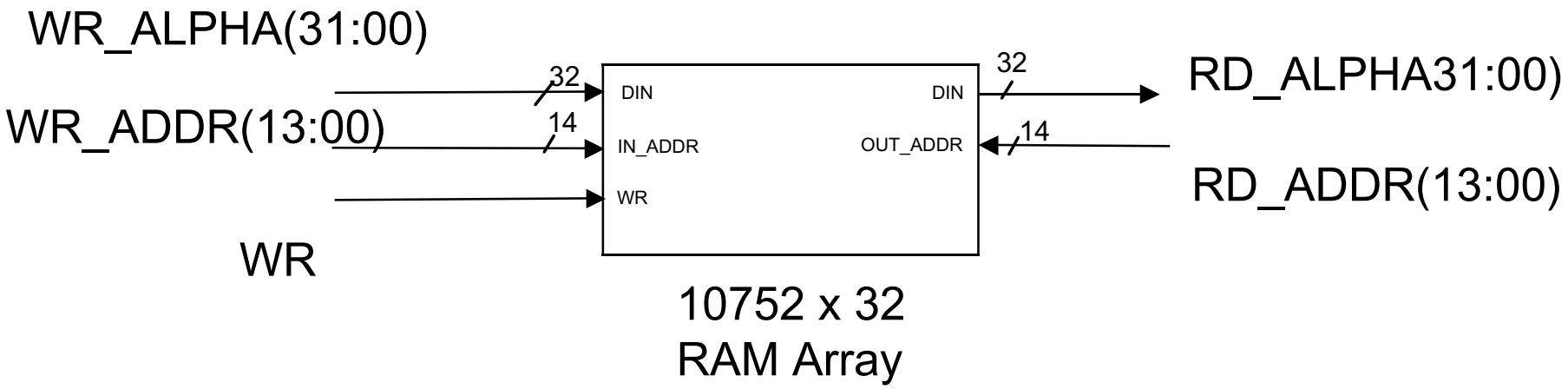


FUJITSU

Beta Buffer

- One 10752 x 32-bit Dual Ported RAM Array
- Self refresh
- Accessed Only during compositing
  - Single-buffered

Beta Buffer Implementation



Display & Compositor Addressing

Display Address Generator Logic

- Counters for row position and line position
- Circuitry for supporting the Line Sequential Stereo Mode
- Sprite Engine draws the left image in the stereo pair in the left half of the Compositing Buffer (LPC Count starting at pixel 0)
- Right image in the stereo pair in the right half of the Compositing Buffer (LPC Count starting at pixel 672)

• Compositor Address Generator

- Receives the compositor address from the Sprite Engine

FUJITSU

Address Control Logic

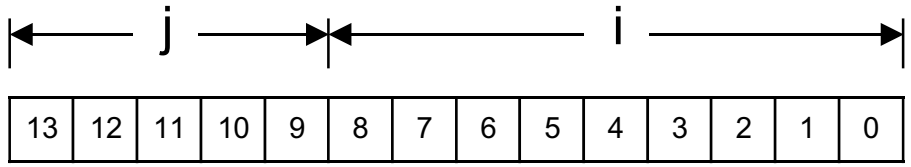
- **Performs Memory Addressing Function for Build and Display Buffers**
- **Compositing Addressing**
  - Address Broken into 2 Portions - I, J
  - One Address Per 4 Pixels in 1 x 4 Array

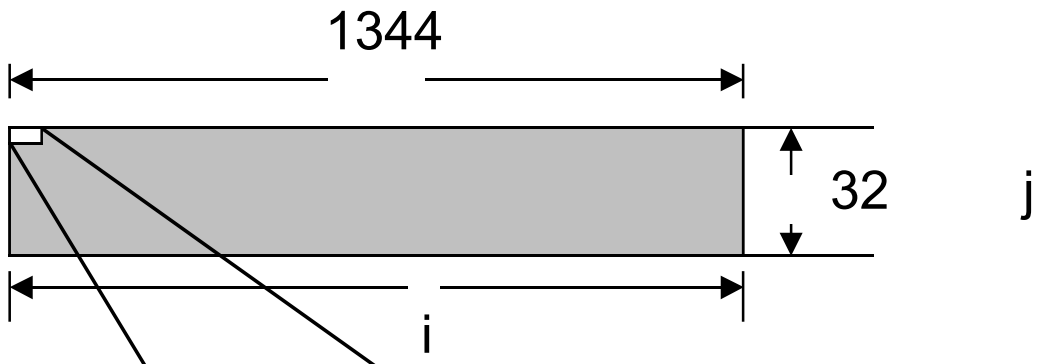
    Addresses (8:0) - I location

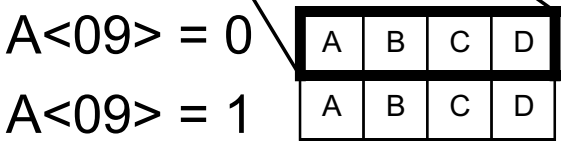
    Addresses (13:9) - J location

Address Control Logic

Normal Mode
Compositor and
Display Address







Compositing and
Display Addressing

(accesses the four different buffers simultaneously as a 1x4 array)

FUJITSU

Media DAC

- **Standard RGB Output**
  - 135 MHz DACs
  - 1344 x 1024 at 75 Hz
- **3 x 256 x 8 Color LUTs**
  - Loaded from Media DSP via Media Bus

FUJITSU

Media Bus

- 16 Bit Multimedia Bus @ 40 MHz
- PCI - like Protocol
- Interface for Device Initialization and Operational Control of the Compositing DAC
- Expandable for future high performance interface for Additional MM Devices

FUJITSU

Summary

- SRAM Based C-DAC Design Complete
- Large Chip for SRAM Design (~ 14mm x 14mm)
- Price Point Requirements Dictate DRAM Based Cell Design
- Next Step Design in Process