rfTextController

© 2022 Robert Finch

## Description

This is a text or tile mode video display controller that supports color intended for use with a 64 or 32-bit bus. The controller uses several internal dual ported r/w memories to store text, text attributes and character bitmaps. The display memory is sixty-four bits wide. Up to 8192 different simultaneous characters may be displayed along with 21-bit background and foreground colors (RGB777 format). The use of internal dual ported memories means that the text controller does not consume any memory bandwidth from the processor. The text controller may also be used as a tile graphics controller via the programmable character set.

The controller is programmable using only seven registers. Default values are established that should provide a reasonable display for 800x600 VGA mode.

The core respects byte lane selects and partial updates of registers are possible. This makes it possible for the core to have an optional 32-bit bus slave interface.

The core is selected via a circuit select input.

|  |  |
| --- | --- |
| Address | Description |
| $00000 to $0FFFF | text screen and attribute memory area, currently the controller only supports a 64kB memory, enough for two 80x50 screens |
| $10000 to $1FEFF | character bitmap memory, only byte addressable (LDB, STB) number of chars depends on char size |
| $1FF00 to $1FFFF | Text controller register area |

Text and Attribute Memory Layout

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 63 60 | 59 58 | 57 37 | 36 16 | 15 13 | 12 0 |
| Plane4 | ~ | Fore Color21 | Back Color21 | ~ | Char code13 |

## Clocks

The text video display controller uses two clocks, a bus timing clock (clk\_i) and a video timing clock (dot\_clk\_i), which can be completely independent.

The core synchronizes the display relative to externally supplied horizontal and vertical synchronization signals.

## Register Description

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Reg No. | Bit | R/W |  | Default Value |  |
| 00 | 7 – 0 | RW | Number of character columns | 64 |  |
| 15 - 8 | RW | Number of character row | 32 |  |
| 19 - 16 | RW | Character output delay | 7 |  |
| 31 – 20 | - | These bits are reserved |  |  |
| 43 – 32 | RW | Window left | 3956 |  |
| 47-44 | - | These bits are reserved |  |  |
| 59-48 | RW | Window top | 4058 |  |
| 63 – 60 | - | These bits are reserved |  |  |
| 08 | 4 – 0 | RW | Maximum scanline for character -1 | 7 |  |
| 7 - 5 | - | Reserved |  |  |
| 11 – 8 | RW | Pixel size width -1 (dot clocks) | 0 |  |
| 15 – 12 | RW | Pixel size height -1 (scan lines) | 0 |  |
| 24 | RW | Reset state (auto resets to zero) | 0 |  |
| 32 | RW | Controller enable | 1 |  |
| 40 | RW | Multi-color mode enable | 0 |  |
| 52 – 48 | RW | Y scroll | 0 |  |
| 60 – 56 | RW | X scroll | 0 |  |
| Other | - | reserved |  |  |
| 10 | 30 – 0 | RW | Color for transparent color ZRGB 4-9-9-9 | 511 |  |
| 62 – 32 | RW | Border color ZRGB 4-9-9-9 | FFBF2020h |  |
| 18 | 30 – 0 | RW | Tile color 1 (multi-color mode) | 0 |  |
| 62 - 32 | RW | Tile color 2 (multi-color mode) | 0 |  |
| 20 | 4 – 0 | RW | Cursor end | 31 |  |
| 7 – 5 | RW | Blink Control  |  |  | | --- | --- | | 00 | No blink | | 01 | No display | | 10 | 1/16 field rate | | 11 | 1/32 field rate | | 7 |  |
| 12 - 8 | RW | Cursor start | 0 |  |
| 15 - 14 | RW | Cursor image type | 0 |  |
| 47 - 32 | RW | Cursor location in memory | 3 |  |
| Other | - | reserved |  |  |
| 28 | 15 – 0 | RW | start address – index into display memory | 0 |  |
| Other | - | reserved |  |  |
| 30 | 15 – 0 | RW | Font address in char bitmap memory | 0 |  |
| 63 -32 | RW | “LOCK” or “UNLK” font locking | “LOCK” |  |
| Other | - | reserved |  |  |

## Graphics

The core may be used as a low-resolution graphics controller via the programmable character set. The characters can be programmed for block graphics. For instance, each character could be a two-by-two grid of pixels. Sixteen different characters would be required to represent all the different combinations. It is also possible to program characters to a three-by-three grid of pixels using 512 programmable characters to represent every possible combination of on/off pixels. The default resolution is 64x32 or (768x576 pixels).

Graphics and text may be intermixed by allocating part of the programmable character set for a graphic array. For instance, using 256 programmable characters a 128x128 bitmapped display can be created.

## Fonts

Multiple fonts can be loaded into the character bitmap memory. The controller supports a 64kB font memory. Which font is selected is determined by the contents of the font address register. The font memory may be locked so that it is not inadvertently changed by an errant program. The number of character glyphs that may be stored depends on the size of characters. An 8x8 glyph will use eight bytes of memory, meaning 8192 different characters can be supported. The default pre-loaded font is 12x18 requiring 36 bytes of memory for each character, therefore only 1820 characters of this size can be supported. Fonts with character glyphs up to 64x64 pixels can be used. Horizontally, glyphs are blocked into a size of 8,16,32,or 64 bits. Vertically, glyphs are a multiple of the horizontal size. A 47x56 glyph must be mapped into a 64x56 array of bytes.

## Multi-color Mode

If multi-color mode is enabled, pixels are combined into pairs to select one of four colors, the foreground color, background color, tile color 1 or tile color2. Each character or tile may then display pixels in one of the four colors.

## Output Planes

A four-bit output plane number may be supplied as part of the character attributes. The plane number controls the display priority when multiple video display devices are present in the video pipeline. Higher numbered planes will appear in front of lower numbered ones.

## Smooth Scrolling

Scrolling the screen in a smooth fashion is supported with the x and y scroll registers which allow the screen to be scrolled pixel by pixel.

## Power On Screen Randomizer

The controller features an automatic screen randomizer that causes random characters to be displayed when the controller is reset. Video display memory is loaded with random values. This is a visual aid that the controller is working properly.

## Display Input / Output Bus

The controller inputs 40-bit ZRGB data and outputs 40-bit ZRGB video data. ZRGB is RGB data with plane number indicator bits tacked on. Four bits are reserved for the plane number and 12 bits are reserved for each RGB color component.

## Core Parameters

|  |  |  |
| --- | --- | --- |
| **Name** | **Default Value** | **Description** |
| COLS | 64 | default number of columns of text |
| ROWS | 32 | default number of rows of text |
| BUSWID | 64 | Slave bus width may be 64 or 32 |

## Module Interface Description

rfTextController

module rfTextController(rst\_i, clk\_i, cs\_i, cyc\_i, stb\_i, ack\_o, wr\_i, sel\_i, adr\_i, dat\_i, dat\_o, dot\_clk\_i, hsync\_i, vsync\_i, blank\_i, border\_i, zrgb\_i, zrgb\_o, xonoff\_i);

|  |  |
| --- | --- |
| **System** | **Description** |
| rst\_i | This signal is normally connected to the system reset signal. It resets the text controller interface forcing it to the reset state. |
| clk\_i | This is usually connected to the system clock and is used as a base timing clock for I/O operations. |
| Slave Port | |
| cs\_i | circuit select input - selects the core for a read or write operation. |
| cyc\_i | indicates that a valid bus cycle is taking place. The core will not respond to the bus unless this signal is active. |
| stb\_i | This strobe signal also indicates that a valid bus cycle is taking place |
| ack\_o | This signal indicates that the core has processed the bus transaction (it is the logical and of cyc\_i and stb\_i). |
| wr\_i | This signal is used to signify a write operation to the text controller. |
| sel\_i | These are byte lane selects Either 8 for 64 bit interface or 4 for 32-bit interface. |
| adr\_i | This seventeen bit address bus is used to address one of text controllers's registers or internal memory. (Registers are described above). Registers respond to the address range $1DFxx |
| dat\_i | This is the 64 or 32 bit data input bus to the text controller. |
| dat\_o | This is the 64 or 32 bit data output bus from the text controller. |
| Video Ports | |
| dot\_clk\_i | This input is the video clock input. Pixel timing is derived from it. |
| hsync\_i | Horizontal sync. This input signal signals the start/end of a video scanline (end-of-line) |
| vsync\_i | Vertical sync. This input signal indicates the end of the video frame. |
| blank\_i | This input signal indicates that the display should be blanked. It is active during the video blanking period. |
| border\_i | This input signal indicates that a border area is active. |
| zrgb\_i | This 40-bit input bus can be connected to an external RGB input. (The text controller may display on top of the external input). |
| zrgb\_o | This output signal bus contains the 40-bit RGB display data. |
| **Parameters** |  |
| ROWS | Use this parameter to specify the default number of text rows. |
| COLS | Use this parameter to specify the default number of text columns. |
| BUSWID | Specify the slave bus width using this parameter |

# WISHBONE Compatibility Datasheet

The text controller core may be directly interfaced to a WISHBONE compatible bus.

|  |  |  |
| --- | --- | --- |
| WISHBONE Datasheet  WISHBONE SoC Architecture Specification, Revision B.3 | | |
|  |  | |
| Description: | Specifications: | |
| General Description: | rfTextController - Text mode video display controller | |
| Supported Cycles: | SLAVE, READ / WRITE  SLAVE, BLOCK READ / WRITE  SLAVE, RMW | |
| Data port, size:  Data port, granularity:  Data port, maximum operand size:  Data transfer ordering:  Data transfer sequencing | 64 or 32 bit (configurable)  8 bit byte lane selects  64 or 32 bit  Little Endian  any (undefined) | |
| Clock frequency constraints: | None, uses separate independent slave bus and video clocks. | |
| Supported signal list and cross reference to equivalent WISHBONE signals | Signal Name:  ack\_o  sel\_i(7:0)  adr\_i(16:0)  clk\_i  dat\_i(63:0)  dat\_o(63:0)  cyc\_i  stb\_i  wr\_i  rst\_i | WISHBONE Equiv.  ACK\_O  SEL\_I()  ADR\_I()  CLK\_I  DAT\_I()  DAT\_O()  CYC\_I  STB\_I  WE\_I  RST\_I |
| Special Requirements: | external sync generator | |