Bt431

Monolithic CMOS 64 x 64 Pixel Cursor Generator

Distinguishing Features

- 64 x 64 Pixel User-Definable Cursor
- Full-Screen/Window Cross Hair Cursor
- Pixel Positioning of Cursors
- Supports Pixel Rates up to 175 MHz
- 1:1, 4:1, and 5:1 Output Multiplexing
- TTL-Compatible Inputs/Outputs
- Standard MPU Interface
- +5 V CMOS Monolithic Construction
- 24-pin 0.3" DIP or 28-pin PLCC Package
- Typical Power Dissipation: 450 mW

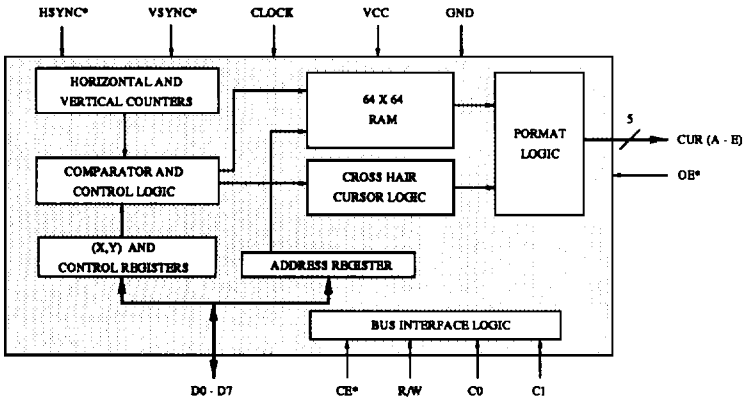
Applications

- High-Resolution Color Graphics
- Image Processing

Customer Benefits

- Reduces Component Count
- Reduces PCB Area Requirements
- Simplifies Cursor Implementation
- Allows Fast Cursor Movement
- Simplifies Software Interface

Functional Block Diagram



Product Description

The Bt431 cursor generator provides a 64 x 64 pixel user-definable cursor and a cross hair cursor for high-resolution, noninterlaced, monochrome or color graphics systems. The cross hair cursor may be implemented as a full-screen or full-window cross hair cursor. Both cursors may be displayed simultaneously, with logical OR and exclusive-OR operations supported. Either cursor may be moved off the top, bottom, left, or right side of the display without wrap-around.

The cursors may be positioned with pixel resolution, and may be individually enabled or disabled from being displayed. A standard MPU bus interface is supported, simplifying system design.

The Bt431 may be programmed to output cursor information for one, four, or five horizontally consecutive pixels, enabling it to be interfaced to either the multiplexed or nonmultiplexed overlay inputs of Brooktree RAMDACs.

The 5:1 output multiplex mode enables support of pixel rates up to 175 MHz.

Brooktree Corporation
9950 Barnes Canyon Rd.
San Diego, CA 92121
(619) 452-7580 • (800) VIDEO IC
TLX: 383 596 • FAX: (619) 452-1249
L431001 Rev. J

Brooktree®

Circuit Description

MPU Interface

As illustrated in the functional block diagram, the Bt431 supports a standard MPU bus interface, allowing the MPU direct access to the internal control registers and cursor RAM.

The MPU interface signals consist of D0–D7, CE*, R/W, C0, and C1. Table 1 illustrates the truth table for the control inputs, and Figure 1 illustrates the MPU read/write timing of the device.

Two 8-bit address registers (address register0 and address register1), cascaded to form a 16-bit address pointer register, are used to address the internal control registers and cursor RAM, as illustrated in Table 2. During read/write cycles to the cursor RAM, the 9 least significant bits of the address pointer register (ADDR0–ADDR8) are incremented following each read or write cycle to the cursor RAM. Thus, the MPU may load the address pointer register with the desired starting cursor RAM address, and burst load new cursor RAM data by writing up to 512 bytes of data to the device. Following a read or write cycle to RAM location $01FF, the address pointer register resets to $0000.

During accesses to the control registers, ADDR0–ADDR8 are incremented after any read or write cycle to a register. While accessing the control registers, the address pointer register will reset to $0000 only following a write cycle to location $01FF. The address register is not incremented when read or written to.

RAMDAC Interface

The Bt431 is designed to generate cursor information using the overlay input ports of Brooktree RAMDACs.

The Bt431 may be interfaced directly to RAMDACs with 4:1 or 5:1 multiplexed overlay ports, supporting display resolutions up to 1280 x 1024 pixels. In this instance, the CUR (A–E) outputs of the Bt431 would connect directly to the overlay inputs of the RAMDAC, and the CLOCK input of the Bt431 would typically be connected directly to the LD* or LDOUT pin of the RAMDAC. The Bt431 must be programmed to output either four or five horizontally consecutive pixels of cursor information each CLOCK cycle. This enables the Bt431 to output cursor information at an effective 175 MHz rate (using 5:1 mode).

To support RAMDACs with nonmultiplexed overlay inputs, the Bt431 may be programmed to output a single pixel of cursor information each CLOCK cycle. In this configuration, the CURA output of the Bt431 would connect directly to one of the overlay inputs of the RAMDAC. This configuration limits the cursor information to an effective 35 MHz rate. The CLOCK input of the Bt431 is typically connected directly to the CLOCK input of the RAMDAC.

The Bt431 may be configured for 4:1 or 5:1 output multiplexing, and an external shift register used (with appropriate control logic) to interface to RAMDACs whose input pixel rate is greater than 35 MHz. In this configuration, the CLOCK must be driven at 1/4 or 1/5 the pixel clock rate. Pixel rates up to 175 MHz may be supported using this technique.

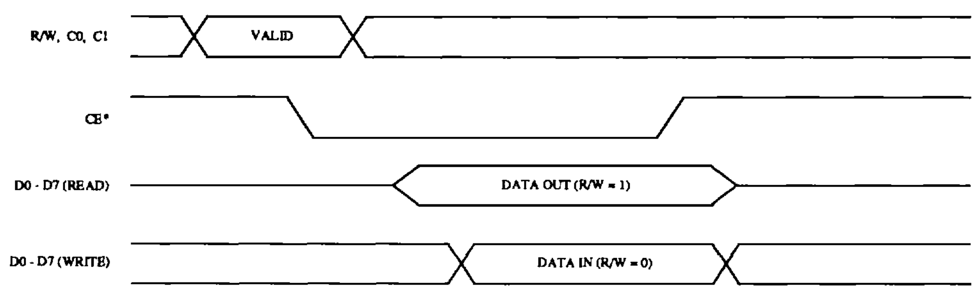


Figure 1. MPU Read/Write Timing.

Circuit Description (continued)

R/W	C1	C0	
0	0	0	write address register0
0	0	1	write address register1
0	1	0	write to RAM location specified by address pointer register
0	1	1	write to control register specified by address pointer register
1	0	0	read address register0
1	0	1	read address register1
1	1	0	read RAM location specified by address pointer register
1	1	1	read control register specified by address pointer register

Table 1. MPU Control Truth Table.

	Address Pointer Register (ADDR15–ADDR0)		
C0	Address Register1 (D7–D0)	Address Register0 (D7–D0)	Register/RAM location addressed
0	0000 0000	0000 0000	cursor RAM location $000
0	0000 0000	0000 0001	cursor RAM location $001
:	:	:	:
0	0000 0000	1111 1111	cursor RAM location $0FF
0	0000 0001	0000 0001	cursor RAM location $100
0	0000 0001	0000 0001	cursor RAM location $101
:	:	:	:
0	0000 0001	1111 1111	cursor RAM location $1FF
1	xxxx xxxx	xxxx 0000	command register
1	xxxx xxxx	xxxx 0001	cursor (x) low register
1	xxxx xxxx	xxxx 0010	cursor (x) high register
1	xxxx xxxx	xxxx 0011	cursor (y) low register
1	xxxx xxxx	xxxx 0100	cursor (y) high register
1	xxxx xxxx	xxxx 0101	window (x) low register
1	xxxx xxxx	xxxx 0110	window (x) high register
1	xxxx xxxx	xxxx 0111	window (y) low register
1	xxxx xxxx	xxxx 1000	window (y) high register
1	xxxx xxxx	xxxx 1001	window width low register
1	xxxx xxxx	xxxx 1010	window width high register
1	xxxx xxxx	xxxx 1011	window height low register
1	xxxx xxxx	xxxx 1100	window height high register

Table 2. Address Pointer Register.

Circuit Description (continued)

64 x 64 Cursor Positioning

When the cursor RAM is being displayed, the contents of the cursor RAM are output onto the CUR (A–E) outputs. A logical one in the cursor RAM results in a logical one being output onto the appropriate CUR (A–E) output during the appropriate clock cycle. The cursor pattern may be changed by changing the contents of the cursor RAM. (See Figure 2.)

The 64 x 64 cursor is centered about the value specified by the cursor (x,y) register. Thus, the cursor (x) register specifies the location of the 31st column of the 64 x 64 RAM (assuming the columns start with 0 for the left-most pixel and increment to 63). Similarly, the cursor (y) register specifies the location of the 31st row of the 64 x 64 RAM (assuming the rows start with 0 for the top-most pixel and increment to 63).

Note that the Bt431 expects (x) to increase going right, and (y) to increase going down, as seen on the display screen.

The cursor (x) position is relative to the first rising edge of CLOCK following the falling edge of HSYNC*. The software must take into account the internal pipeline delays, the amount of skew between the output cursor data and external pixel data, and whether 1:1, 4:1, or 5:1 output multiplexing is being done.

The cursor (y) position is relative to the first falling edge of HSYNC* that is at two or more clock cycles after the falling edge of VSYNC*. (See Figure 2.)

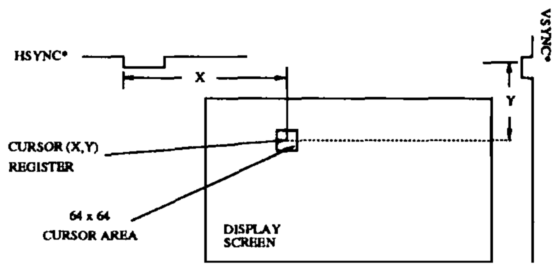


Figure 2. 64 x 64 Cursor Positioning.

Circuit Description (continued)

Cross Hair Cursor Positioning

Cursor positioning for the cross hair cursor is also done through the cursor (x,y) register. (See Figure 3.)

The intersection of the cross hair cursor is specified by the cursor (x,y) register. If the thickness of the cross hair cursor is greater than one pixel, the center of the intersection is the reference position.

During times that cross hair cursor information is to be displayed, a logical one is output onto the appropriate CUR (A–E) output during the appropriate clock cycle.

The cross hair cursor is limited to being displayed within the cross hair window, which is specified by the window (x,y), window width, and window height registers. Since the cursor (x,y) register must specify a point within the window boundaries, **it is the responsibility of the software to ensure that the cursor (x,y) register does not specify a point outside of the cross hair cursor window.**

If a full-screen cross hair cursor is desired, the window (x,y) registers should contain $0000 and the window width and height registers should contain $0FFF.

Again, the cursor (x) position is relative to the first rising edge of CLOCK following the falling edge of HSYNC*. The software must take into account the internal pipeline delays, the amount of skew between the output cursor data and the external pixel data, and whether 1:1, 4:1, or 5:1 output multiplexing is being done.

The cursor (y) position is relative to the first falling edge of HSYNC* that is two or more clock cycles after the falling edge of VSYNC*.

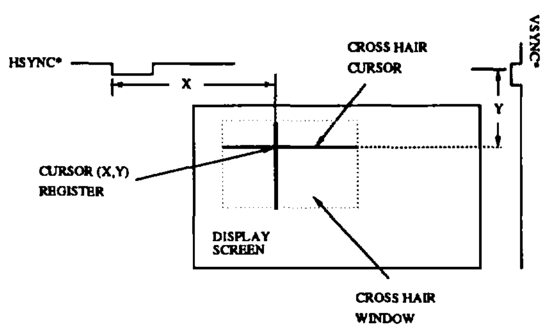


Figure 3. Cross Hair Cursor Positioning.

Circuit Description (continued)

Dual Cursor Positioning

Both the 64 x 64 cursor and the cross hair cursor may be enabled for display simultaneously, enabling the generation of custom cross hair cursors.

During the 64 x 64 pixel area in which the user-definable cursor would be displayed, the contents of the cursor RAM may be logically ORed or exclusive-ORed with the cross hair cursor information.

As previously mentioned, the cursor (x,y) register specifies the location of bit (31,31) of the cursor RAM. As the user-definable cursor contains an even number of pixels in the horizontal and vertical direction, there will be a one-pixel offset from being truly centered about the cross hair cursor.

Figure 4 illustrates displaying the dual cursors, and Figure 5 illustrates the video input/output timing of the Bt431.

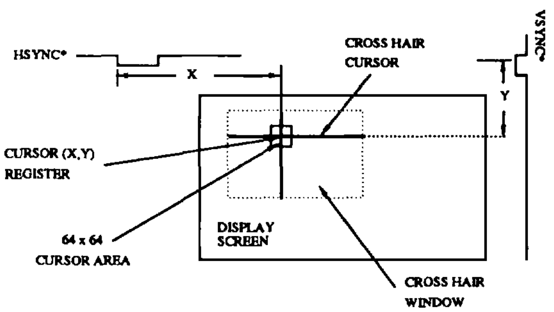


Figure 4. Dual Cursor Positioning.

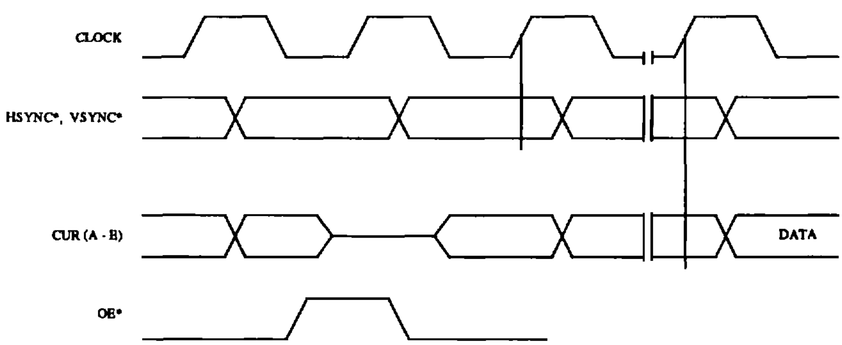


Figure 5. Video Input/Output Timing.

Internal Registers

Cursor (x,y) Register

These registers are used to specify the (x,y) coordinate of the center of the 64 x 64 pixel cursor window, or the intersection of the cross hair cursor. The cursor (x) register is made up of the cursor (x) low register (CXLR) and the cursor (x) high register (CXHR); the cursor (y) register is made up of the cursor (y) low register (CYLR) and the cursor (y) high register (CYHR). They are not initialized and may be written to or read by the MPU at any time.

CXLR and CXHR are cascaded to form a 12-bit cursor (x) register. Similarly, CYLR and CYHR are cascaded to form a 12-bit cursor (y) register. Bits D4–D7 of CXHR and CYHR are always a logical zero.

	Cursor (x) High (CXHR)				Cursor (x) Low (CXLR)							
Data Bit	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X Address	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

	Cursor (y) High (CYHR)				Cursor (y) Low (CYLR)							
Data Bit	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Y Address	Y11	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

The cursor (x) value to be written is calculated as follows:

Cx = desired display screen (x) position + D + H–P

where

P = 37 if 1:1 output multiplexing, 52 if 4:1 output multiplexing, 57 if 5:1 output multiplexing
D = skew (in pixels) between the output cursor data and external pixel data
H = number of pixels between the first rising edge of CLOCK following the falling edge of HSYNC* to active video.

The P value is 1/2 cursor RAM width + (internal pipeline delay in clock cycles * 1, 4, or 5 depending on multiplex selection)

Values from $0000 to $0FFF may be written into the cursor (x) register.

Internal Registers (continued)

The cursor (y) value to be written is calculated as follows:

Cy = desired display screen (y) position + V–32

where

V = number of scan lines from the first falling edge of HSYNC* that is two or more clock cycles after the falling edge of VSYNC* to active video.

Values from $0FC0 (–64) to $0FBF (+4031) may be loaded into the cursor (y) register. The negative values ($0FC0 to $0FFF) are used in situations where V < 32, and the cursor must be moved off the top of the screen.

The cursor (x,y) registers should be written to only during the vertical retrace interval. Note that a falling edge of VSYNC* should not occur between the time the MPU writes the first byte of (x,y) and the last (fourth) byte of (x,y) information. Otherwise, temporary "tearing" of the cursor may occur.

Internal Registers (continued)

Cursor RAM

This 64 x 64 RAM is used to define the pixel pattern within the 64 x 64 pixel cursor window. It is not initialized, and may be written to or read by the MPU at any time. As MPU accesses to the cursor RAM have priority over the cursor display process, the cursor RAM should not be accessed during the horizontal sync intervals to minimize contention of the cursor updating and displaying processes.

During MPU accesses to the cursor RAM, the address pointer register is used to address the cursor RAM, as illustrated below. Figure 6 illustrates the internal format of the cursor RAM, as it appears on the display screen.

Address Pointer Register Value	Address RAM Location
$0000	byte $000
$0001	byte $001
:	:
$01FF	byte $1FF

As shown below, bit D7 is the left-most pixel within a segment of eight pixels. This enables the software generation of cursor patterns without bit swapping to obtain the desired pattern.

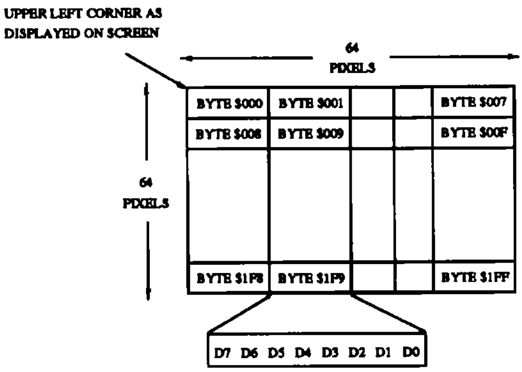


Figure 6. Cursor RAM as Displayed on the Screen.

Internal Registers (continued)

Window (x,y) Register

These registers are used to specify the (x,y) coordinate of the upper left corner of the cross hair cursor window. The window (x) register is made up of the window (x) low register (WXLR) and the window (x) high register (WXHR); the window (y) register is made up of the window (y) low register (WYLR) and the window (y) high register (WYHR). They are not initialized and may be written to or read by the MPU at any time.

WXLR and WXHR are cascaded to form a 12-bit window (x) register. Similarly, WYLR and WYHR are cascaded to form a 12-bit window (y) register. Bits D4–D7 of WXHR and WYHR are always a logical zero.

	Window (x) High (WXHR)				Window (x) Low (WXLR)							
Data Bit	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X Address	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

	Window (y) High (WYHR)				Window (y) Low (WYLR)							
Data Bit	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Y Address	Y11	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

The window (x) value to be written is calculated as follows:

Wx = desired display screen (x) position + D + H–P

where

P = 5 if 1:1 output multiplexing, 20 if 4:1 output multiplexing, 25 if 5:1 output multiplexing
D = skew (in pixels) between the output cursor data and external pixel data
H = number of pixels between the first rising edge of CLOCK following the falling edge of HSYNC* to active video.

The P value is the number of internal pipeline delays times 1, 4, or 5 depending on the multiplex selection.

The window (y) value to be written is calculated as follows:

Wy = desired display screen (y) position + V

where

V = number of scan lines from the first falling edge of HSYNC* that is two or more clock cycles after the falling edge of VSYNC* to active video.

Values from $0000 to $0FFF may be written to the window (x) and window (y) registers. A full-screen cross hair cursor is implemented by loading the window (x,y) registers with $0000 and the window width and height registers with $0FFF.

The window (x,y) registers should be written to only during the vertical retrace interval. Note that a falling edge of VSYNC* should not occur between the time the MPU writes the first byte of (x,y) and the last (fourth) byte of (x,y) information. Otherwise, temporary repositioning of the cross hair cursor may occur.

Internal Registers (continued)

Window Width and Height Registers

These registers are used to specify the width and height (in pixels) of the cross hair cursor window. The window width register is made up of the window width low register (WWLR) and the window width high register (WWHR); the window height register is made up of the window height low register (WHLR) and the window height high register (WHHR). They are not initialized and may be written to or read by the MPU at any time.

WWLR and WWHR are cascaded to form a 12-bit window width register. Similarly, WHLR and WHHR are cascaded to form a 12-bit window height register. Bits D4–D7 of WWHR and WHHR are always a logical zero.

	Window Width High (WWHR)				Window Width Low (WWLR)							
Data Bit	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
X Address	X11	X10	X9	X8	X7	X6	X5	X4	X3	X2	X1	X0

	Window Height High (WHHR)				Window Height Low (WHLR)							
Data Bit	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
Y Address	Y11	Y10	Y9	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0

The actual window width is 2, 8, or 10 pixels more than the value specified by the window width register, depending on whether 1:1, 4:1, or 5:1 output multiplexing is specified. The actual window height is 2 pixels more than the value specified by the window height register. Therefore, the minimum window width is 2, 8, or 10 pixels, for 1:1, 4:1, and 5:1 multiplexing, respectively, and the minimum window height is 2 pixels.

Values from $0000 to $0FFF may be written to the window width and height registers.

The window width and height registers should be written to only during the vertical retrace interval. Note that a falling edge of VSYNC* should not occur between the time the MPU writes the first byte and the last (fourth) byte of information. Otherwise, temporary "resizing" of the cross hair cursor may occur.

Internal Registers (continued)

Command Register

The command register is used to control various functions of the Bt431. It is not initialized, and may be written to or read by the MPU at any time.

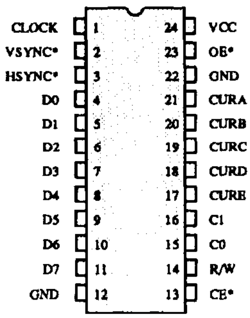
D7	Reserved. This bit should always be a logical zero.
D6	64 x 64 cursor enable. A logical one enables the contents of the cursor RAM to be output during times that user-definable cursor information is to be displayed. A logical zero disables the cursor RAM information from being output.
D5	Cross hair cursor enable. A logical one enables cross hair cursor information to be output. A logical zero disables the cross hair cursor information from being output.
D4	Cursor format control. If both the 64 x 64 cursor and the cross hair cursor are enabled for display, this bit specifies whether the contents of the cursor RAM are to be logically exclusive-ORed (logical zero) or ORed (logical one) with the cross hair cursor.
D3, D2	Multiplex control. These 2 bits specify whether 1, 4, or 5 bits of cursor information are output every clock cycle, as follows: (00) 1:1 multiplexing (01) 4:1 multiplexing (10) 5:1 multiplexing (11) reserved
D1, D0	Cross hair cursor thickness. These 2 bits specify whether the horizontal and vertical thickness of the cross hair is 1, 3, 5, or 7 pixels, as follows: (00) 1 pixel (01) 3 pixels (10) 5 pixels (11) 7 pixels The horizontal and vertical segments are centered about the value in the cursor (x,y) register.

Pin Descriptions

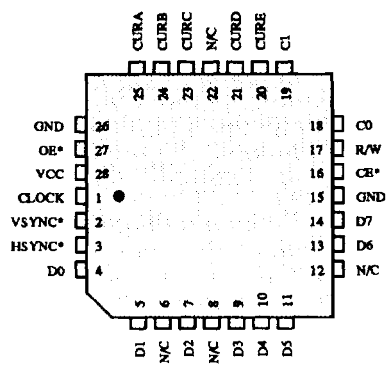
Pin Name	Description
VSYNC*	Vertical sync control input (TTL compatible). A logical zero indicates that the display is currently in the vertical sync interval. It is latched on the rising edge of CLOCK.
HSYNC*	Horizontal sync control input (TTL compatible). A logical zero indicates that the display is currently in the horizontal sync interval. It is latched on the rising edge of CLOCK.
CLOCK	Clock input (TTL compatible). The rising edge of CLOCK is used to latch the VSYNC* and HSYNC* inputs, and to output cursor information onto the CUR (A–E) outputs. It is recommended that the CLOCK input be driven by a dedicated TTL buffer. If programmed for 1:1 output multiplexing, CLOCK should be the pixel clock rate. When programmed for 4:1 or 5:1 output multiplexing, CLOCK should be 1/4 or 1/5 the pixel clock rate, respectively.
CUR (A–E)	Cursor outputs (TTL compatible). During the pixel times that cursor information is to be displayed, either cross hair cursor information or the contents of the cursor RAM are output onto these pins. If programmed for 4:1 output multiplexing, the CURE output will always be a logical zero. If programmed for 1:1 output multiplexing, the CURB, CURC, CURD, and CURE outputs will always be a logical zero. When programmed for 4:1 or 5:1 multiplexing, CURA corresponds to the left-most pixel, followed by CURB, etc., repeating every four or five pixels.
OE*	Output enable control input (TTL compatible). A logical one asynchronously three-states the CUR (A–E) outputs, and a logical zero asynchronously enables cursor data to be output on the cursor outputs.
R/W	Read/write control input (TTL compatible). A logical zero indicates that the MPU is writing data to the device and a logical one indicates that the MPU is reading data from the device. See Figure 1.
CE*	Chip enable control input (TTL compatible). This input must be a logical zero to enable data to be written to or read from the device. During write operations, data is internally latched on the rising edge of CE*. See Figure 1.
C0, C1	Control inputs (TTL compatible). These inputs specify the operation the MPU is performing. See Tables 1 and 2.
D0–D7	Data bus (TTL compatible). Data is transferred into and out of the device over this 8-bit bidirectional data bus. D0 is the least significant bit.
VCC	Power.
GND	Ground.

Pin Descriptions (continued)

24-pin DIP Package



28-pin Plastic J-Lead (PLCC) Package



Note: N/C pins may be left floating without affecting the performance of the Bt431.

ESD and Latchup Considerations

Correct ESD-sensitive handling procedures are required to prevent device damage, which can produce symptoms of catastrophic failure or erratic device behavior with somewhat "leaky" inputs.

All logic inputs should be held low until power to the device has settled to the specified tolerance. Avoid power decoupling networks with large time constants, which could delay VAA power to the device. Ferrite beads must only be used for analog power VAA decoupling. Inductors cause a time constant delay that induces latchup.

Latchup can be prevented by assuring that all VCC pins are at the same potential, and that the VCC supply voltage is applied before the signal pin voltages. The correct power-up sequence assures that any signal pin voltage will never exceed the power supply voltage by more than +0.5 V.

Application Information

Power-up Initialization

Following a power-up sequence, the Bt431 must be initialized. The following sequence is recommended:

1. Write $0000 to address pointer register
2. Do 13 write cycles to control registers
3. Write $0000 to address pointer register
4. Do 512 write cycles to the cursor RAM

Prior to the above sequence, the MPU may perform diagnostic checks on the device, such as checking that the RAM and control registers may be written to and read back.

Loading the Cursor RAM

When changing the cursor pattern, it is recommended that the following sequence be used to load the cursor RAM:

1. Write $0000 to address pointer register
2. Do 512 write cycles to the cursor RAM

Moving the Cursor

It is recommended that the following sequence be used to update the cursor (x,y) register:

1. Write $0001 to address pointer register
2. Read cursor (x) low
3. Read cursor (x) high
4. Read cursor (y) low
5. Read cursor (y) high
6. Calculate new (x,y) value
7. Write $0001 to address pointer register
8. Write new cursor (x) low
9. Write new cursor (x) high
10. Write new cursor (y) low
11. Write new cursor (y) high

The above sequence also applies to updating the window (x,y) register, except $0005 should be written to the address pointer register.

Changing the Window Size

To change the size of the cross hair window, it is recommended that the following sequence be used:

1. Write $0009 to address pointer register
2. Read window width low
3. Read window width high
4. Read window height low
5. Read window height high
6. Calculate new window width/height
7. Write $0009 to address pointer register
8. Write new window width low
9. Write new window width high
10. Write new window height low
11. Write new window height high

Using Multiple Devices

Multiple Bt431s may be used to generate more than one cursor, or to generate a multi-color cursor.

If using multiple devices to generate more than one cursor, the cursor outputs may be logically gated together, or each Bt431 may interface to a separate overlay input of the RAMDAC. If separate overlay inputs are used, the cursors will be automatically prioritized depending on which overlay is used for each cursor.

To generate a multi-color cursor (for example, using two Bt431s to generate a three-color cursor), each Bt431 must interface to a separate overlay input of the RAMDAC. Either a separate cursor (x,y) calculation for each Bt431 may be performed, or the same cursor (x,y) calculation used with the cursor information appropriately offset in the cursor RAM.

Interfacing to the Bt453 and Bt458

Figure 7 illustrates interfacing a single Bt431 to the Bt453 RAMDAC and Figure 8 illustrates interfacing to the Bt458 RAMDAC.

Interfacing to the Bt451, Bt454, Bt457, and Bt461/462 RAMDACs are similar to interfacing to the Bt458, due to the multiplexed overlay inputs of these devices. When interfacing to the Bt454, the CLOCK pin of the Bt431 should be connected to the LDOUT pin of the Bt454, and the Bt431 configured for 4:1 output multiplexing. Interfacing to the Bt450, Bt473, Bt475/477, Bt479, and Bt471/476/478 RAMDACs is similar to interfacing to the Bt453.

Application Information (continued)

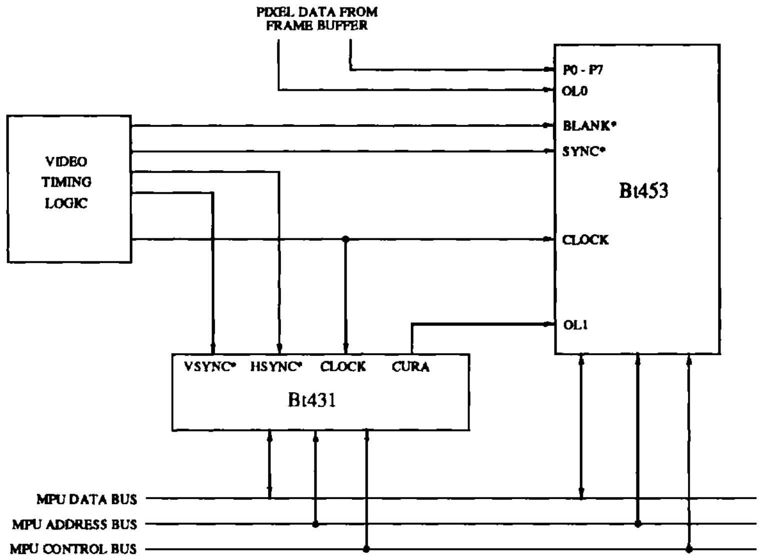


Figure 7. Interfacing to the Bt453.

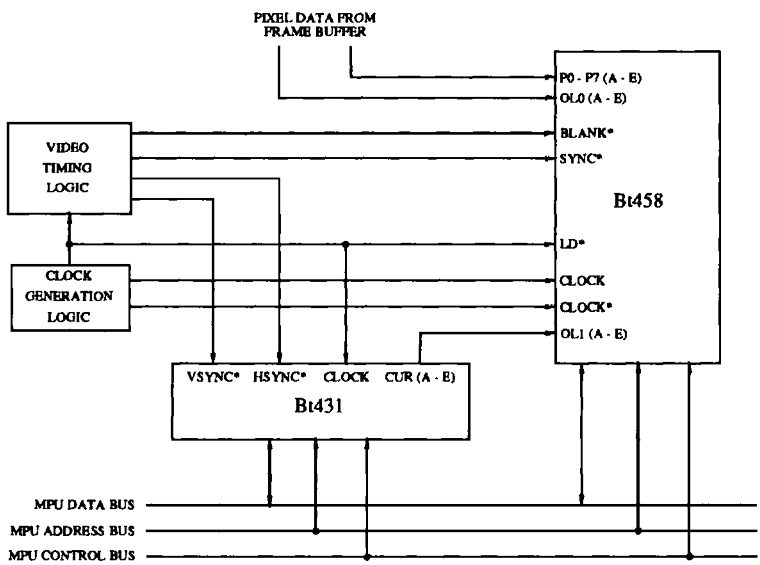


Figure 8. Interfacing to the Bt458.

Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Units
Power Supply	VCC	4.75	5.00	5.25	Volts
Ambient Operating Temperature	TA	0		+70	°C

Absolute Maximum Ratings

Parameter	Symbol	Min	Typ	Max	Units
VCC (measured to GND)				7.0	Volts
Voltage on Any Signal Pin*		GND–0.5		VCC + 0.5	Volts
Ambient Operating Temperature	TA	–55		+125	°C
Storage Temperature	TS	–65		+150	°C
Junction Temperature	TJ				
Ceramic Package				+175	°C
Plastic Package				+150	°C
Soldering Temperature (5 seconds, 1/4" from pin)	TSOL			260	°C
Vapor Phase Soldering (1 minute)	TVSOL			220	°C

Note: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

* This device employs high-impedance CMOS devices on all signal pins. It should be handled as an ESD-sensitive device. Voltage on any signal pin that exceeds the power supply voltage by more than +0.5 V can induce destructive latchup.

DC Characteristics

Parameter	Symbol	Min	Typ	Max	Units
Digital Inputs					
Input High Voltage	VIH	2.0		VCC + 0.5	Volts
Input Low Voltage	VIL	GND–0.5		0.8	Volts
Input High Current (Vin = 2.4 V)	IIH			1	µA
Input Low Current (Vin = 0.4 V)	IIL			-1	µA
Input Capacitance	CIN				pF
(f = 1 MHz, Vin = 2.4 V)			7		
Digital Outputs (D0–D7)					
Output High Voltage	VOH	2.4			Volts
(IOH = –400 µA)					
Output Low Voltage	VOL			0.4	Volts
(IOL = 3.2 mA)					
3-state Current	IOZ			10	µA
Output Capacitance	COUT		20		pF
Digital Outputs (CURA–CURE)					
Output High Voltage	VOH	2.4			Volts
(IOH = –400 µA)					
Output Low Voltage	VOL			0.4	Volts
(IOL = 1.6 mA)					
3-state Current	IOZ			10	µA
Output Capacitance	COUT		20		pF

Test conditions (unless otherwise specified): "Recommended Operating Conditions." Typical values are based on nominal temperature, i.e., room, and nominal voltage, i.e., 5 V.

AC Characteristics

Parameter	Symbol	Min	Typ	Max	Units
Clock Rate (per 1, 4, or 5 pixels)	Fmax			35	MHz
C0, C1, R/W Setup Time	1	10			ns
C0, C1, R/W Hold Time	2	15			ns
CE* Low Time	3	50			ns
CE* High Time	4	25			ns
CE* Asserted to Data Bus Driven	5	6			ns
CE* Asserted to Data Valid	6			100	ns
CE* Negated to Data Bus 3-Stated	7			15	ns
Write Data Setup Time	8	35			ns
Write Data Hold Time	9	4	2.5		ns
VSYNC*, HSYNC* Setup Time	10	10			ns
VSYNC*, HSYNC* Hold Time	11	5			ns
VSYNC*, HSYNC* Low Time		4			Clocks
VSYNC*, HSYNC* High Time		4			Clocks
Clock Cycle Time	12	28.6			ns
Clock Pulse Width High	13	10			ns
Clock Pulse Width Low	14	10			ns
Pipeline Delay	15			5	Clocks
Output Delay	16			20	ns
Three-State Disable Time	17			15	ns
Three-State Enable Time	18			15	ns
VCC Supply Current*	ICC			100	mA

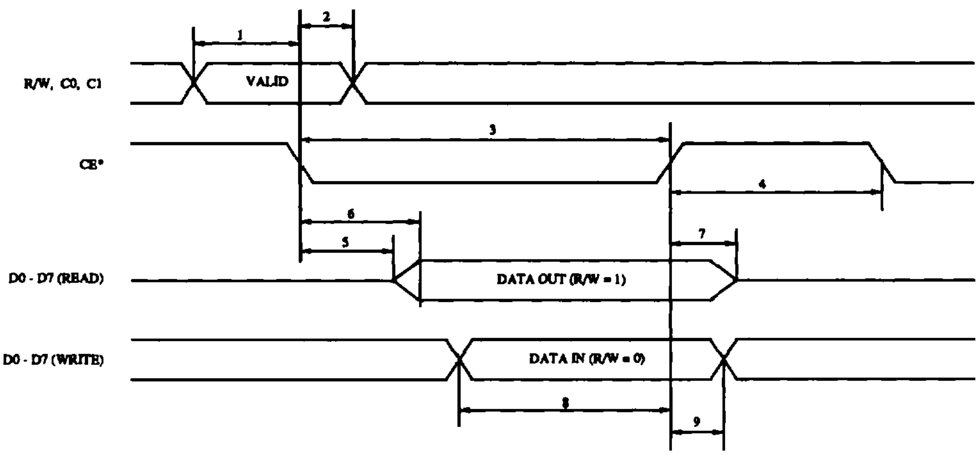
Test conditions (unless otherwise specified): "Recommended Operating Conditions." TTL input values are 0–3 V, with input rise/fall times ≤ 4 ns, measured between the 10% and 90% points. Timing reference points at 50% for inputs and outputs. CURA–CURE output load ≤ 10 pF, D0–D7 output load ≤ 130 pF. Typical values are based on nominal temperature, i.e., room, and nominal voltage, i.e., 5 V.

*At Fmax. ICC (typ) at VAA = 5.0 V. ICC (max) at VAA = 5.25 V.

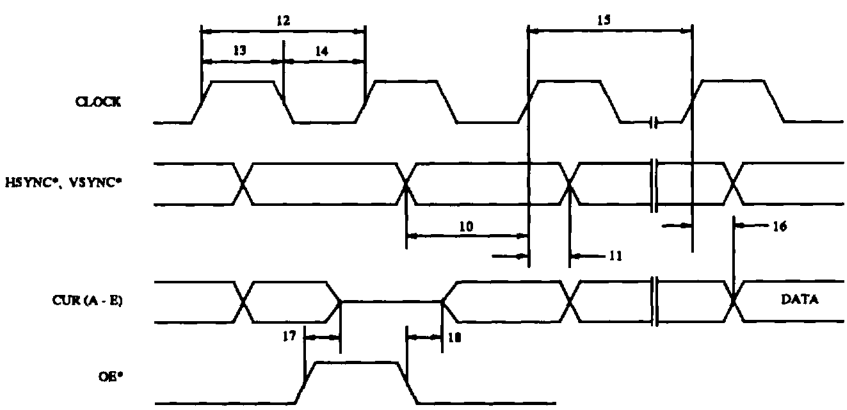
Ordering Information

Model Number	Package	Ambient Temperature Range
Bt431KC	24-pin 0.3" CERDIP	0° to +70° C
Bt431KPJ	28-pin Plastic J-Lead	0° to +70° C

Timing Waveforms



MPU Read/Write Timing.



Video Input/Output Timing.

Revision History

Datasheet Revision	*Change from Previous Revision*
G	Correct PLCC pinout.
H	Update Application Information Section to include interfacing to new RAMDACs.
I	Update three-state currents in DC section to be 10 µA.
J	Expanded ESD/Latchup information.