



Overview

With the LF3312's flexible memory address architecture, data can be sequentially stored in memory and then accessed using a completely reordered address. An application such as creating the 'vertical flip', or upside-down image, of a frame of video is a good example of this addressing flexibility.

The memory would accept the pixel data as typical raster scanned video (with horizontal resolution H and vertical resolution V).

P0,0 P0,1 P0,2 ... P0,H-1 P1,0 P1,1 P1,2 ...

Through real-time address manipulation we can easily rearrange the order we read the stored data - resulting in the 'vertical mirror' or upside-down version of the frame of video. The common raster scan method starts reading the first line (line 0), scans left to right and then increments to the next line. Instead, we can start reading the last line (line V-1), scan from left to right and then decrement the line address.

The resulting pixel order would look like

PV-1,0 PV-1,1 ... PV-1,H-1 PV-2,0 PV-2,1 ...

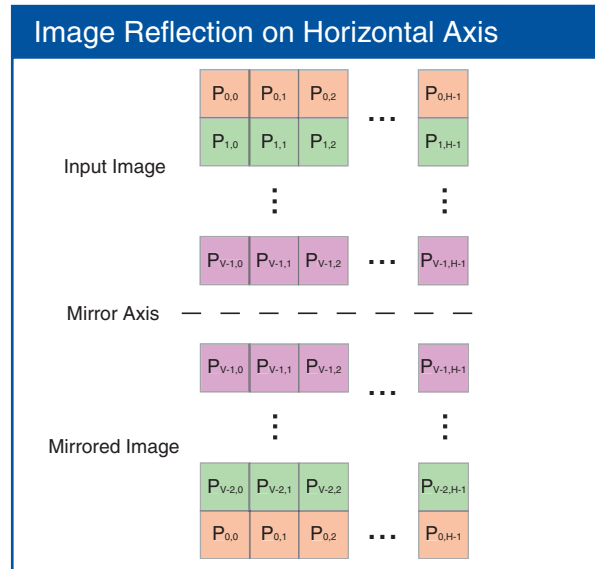
Address Manipulation

For this particular application, two frames of video are stored in two 'memory regions'. While the current frame is being written (memory region 1), the previously stored frame (memory region 0) can be read out with the rearranged addressing sequence. Writing and reading must "ping-pong" between memory regions 0 and 1, with the writes and reads never coinciding on the same region. The write pointer should be reset on the boundary of even-numbered frames. That is, the write pointer should be reset on every other Frame sync signal (F-sync) starting at 0 (frames 0, 2, 4, etc). This results in storing the two most recent frames in memory at all times - without requiring the write pointer to be explicitly forced back and forth between memory regions 0 and 1. Using this method, only the magnitude of the vertical address differentiates video frame 0 or 1. The first frame of video is stored in lines 0 to V-1. The second frame of video is stored in lines V to 2V-1. The read pointer follows the vertically-reversed raster scan addressing sequence described above, but must sometimes be offset by a full frame in reference to the write pointer (address_sequence + frame_depth). For example, when the latest frame is being written to memory region 0, the read address sequence must be accessing memory region 1. This buffering system should be synchronized to begin reading the first pixel of the last line of the previous frame just as we begin writing the latest frame's first pixel. Devices can be cascaded easily for depth expansion. Please refer to the Cascading Application Note.

The input pixel data will be automatically written into memory as sequential data. That is, the write address is incremented sequentially like a FIFO with no external addressing needed.

The read address reordering however does require the use of external addressing. The read address can be forced to a new address every cycle, or 'jumped' to the start of a new line only once per line. In the simpler case of 'jumping' the read pointer, we can force the read pointer to the start of a line and then release the pointer, letting it automatically resume a natural sequential sequence. The LF3312 offers a 24bit external address port for applications such as this that require Random Access addressing. Of course, not all of the 24bits of address need to be dynamically controlled. Depending on the resolution, much of the MSB address bits can be tied off and ignored.

For image manipulation applications, the LF3312 is capable of internally remapping its internal linear address space into a X/Y Cartesian coordinate system. We can access image data in terms of line/column or horizontal/vertical addresses. This greatly simplifies the address controller.

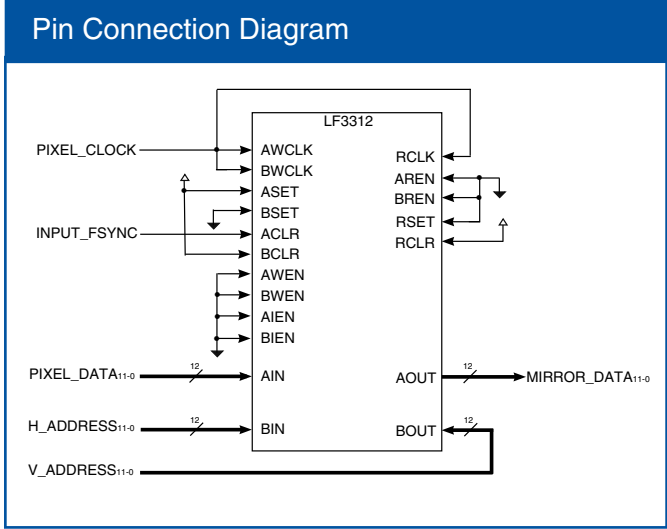




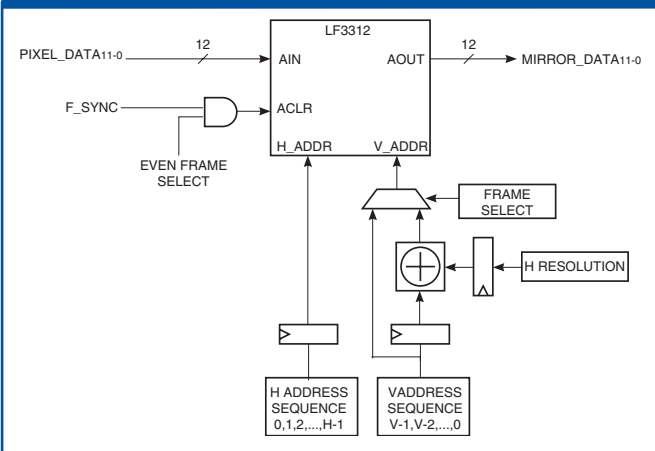
Implementation Details

The LF3312 must be placed in single channel random access mode (Register 8[2:0] = 001). Single channel allows us to use the full depth of the memory and it allows us to cascade devices for greater depth, and it frees up the BIN/BOU ports - which act as the 24bit external address port. The Random Access mode invalidates any empty/full FIFO considerations and allows us to manipulate the write/read pointers as we wish.

In order for the device to remap a Horizontal/Vertical address into a linear address, the horizontal resolution (line-length) must be provided. This is programmed using the 16bit ROWLENGTH control register (Registers 0 & 1). For example, for a line-length of 1280 pixels, we would load ROWLENGTH as 500hex.



Address Generators - Full-time Address Forcing



The BIN port is used as the Horizontal address and the BOU port is the Vertical address. The Cartesian coordinates are internally mapped to the memory's linear address space as:

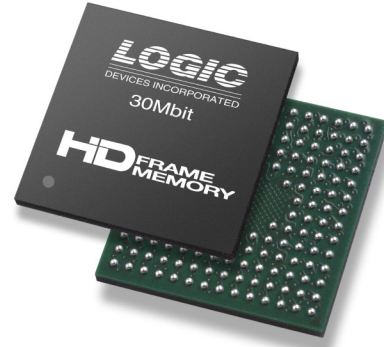
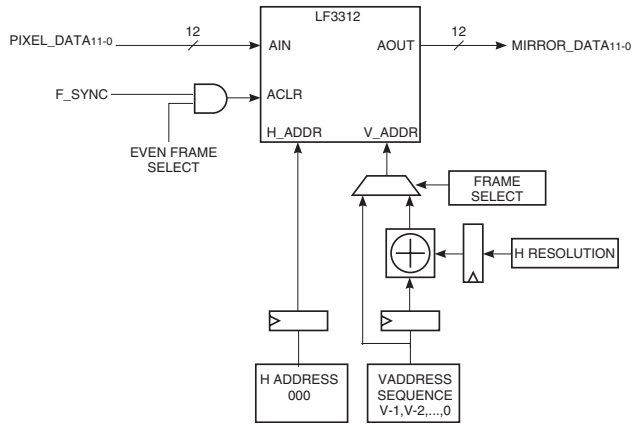
$$\text{Internal memory address} = (\text{BOU} \times \text{ROWLENGTH}) + \text{BIN}$$

In order to select the external BIN/BOU 24bit address port as the source of the Read-address, BSET=0 and BCLR=1. Bringing RSET LOW forces the read-pointer to an external address. RSET can be programmed to be level or negative edge triggered. In order to continuously force the read pointer to an external address, RSET must be programmed to be level sensitive - and kept LOW. In order to release the address to let it continue a sequential sequence, RSET can be programmed to be either negative edge or level sensitive (in which case it must be brought LOW only for one cycle).

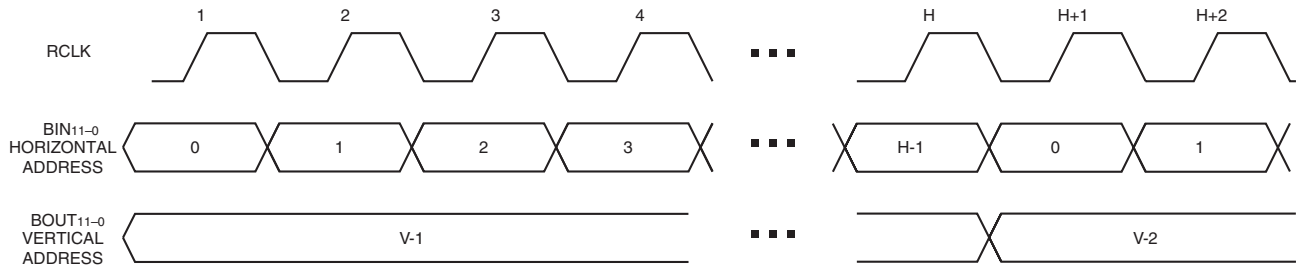
As mentioned earlier, an external frame-sync signal must reset the write pointer at the start of every other input frame. This maintains a constant address reference (the first pixel of Line 0 Frame 0 is always defined as address 0). The ACLR pin resets the write pointer when brought LOW. NOTE: ACLR can be programmed to be level or negative edge triggered. reference (the first pixel of Line 0 is always defined as address 0). The ACLR pin resets the write pointer when brought LOW. NOTE: ACLR can be programmed to be level or negative edge triggered.



Address Generator - Force & Release Method

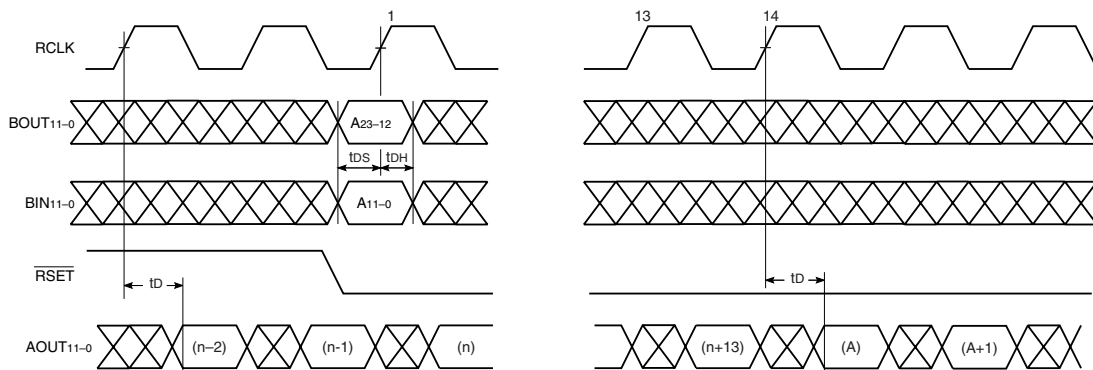


Re-odering of the Read Address to Implement "Mirror Image"



- * Rising edges 1 through H latch the address of the pixels of Line V-1 in normal raster order
- * Rising edges H+1 through 2H latch the address of the pixels of Line V-2 in normal raster order, and so-on
- * The contents of memory addresses supplied by BIN/BOUT appear on the AOUT port 14 rising edges of RCLK after the address is latched

Re-odering of the Read Address using Force & Release Method



AOE = LOW BOE = HIGH AREN = BREN = LOW BSET = LOW BCLR = HIGH OPMODE[2:0] = 001 MARK_SEL (Register 9[3]) = 1

NOTE: RSET programmed to be falling edge sensitive

NOTE: It takes 14 rising edges of RCLK upon setting/jumping the Read pointer (to the 24bit Address "A" on BIN/BOUT) for the contents of location A to be dumped onto AOUT