TECHNICAL NOTE

REGULAR, REAL-TIME AND SPLIT READ TRANSFERS

INTRODUCTION

As described in TN-42-02 "Designing with the MT42C4256/8128 VRAM," READ TRANSFER functions on VRAMs are used to perform internal transfers of data from the DRAM array to the Serial-Access-Memory (SAM) portion of the VRAM. In graphics systems, the DRAM array is used to store the data representing pixels to be displayed on the screen. This pixel data is manipulated via the DRAM I/O port, and system performance is optimized by maximizing the bandwidth of the DRAM port that is available for this manipulation. Toward that end, VRAMs include a SAM to handle the task of providing the stream of pixels to the display circuitry. The SAM acts as a buffer between the display memory (DRAM array) and the monitor (see Figure 1).

Display data is transferred, a portion at a time, to the SAM buffer. From there it is clocked out to the display circuitry which drives the monitor. To maintain a continuous flow of data to the monitor, the subsequent portion of data must be transferred to the SAM before the buffer becomes "empty."

This note describes the various forms of read-transfer functions and how each is used to feed data to the SAM buffer in different situations.

READ TRANSFERS

The vast majority of VRAMs available on the market, including all Micron VRAMs, provide a SAM buffer that is equal in size to one row of the DRAM array for that particular VRAM. For example, 2 Meg VRAMs include a DRAM array consisting of 512 rows; each row contains 512 columns of 8-bit-wide locations. Accordingly, the SAM consists of 512 8-bit locations. In "regular" READ TRANSFERs (the most fundamental form) one complete row of data from the DRAM array is copied to the SAM, thereby overwriting the entire contents of the SAM. READ TRANSFERs are used when each row in the DRAM array (and hence, the SAM) contains the pixel data for an integral number of scan lines on the monitor. In this case, the buffer will become empty at the end of a scan line and can be reloaded during the horizontal blanking time of the monitor. The READ TRANSFER cycle can easily be completed within the blanking time; and therefore the specific timing is not critical.

REAL-TIME READ TRANSFERS

In configurations where part of a scan line is stored in one row of the DRAM array and the remainder is stored in another row, the SAM buffer will become "empty" in the middle of a scan line, and a midline transfer will be required to reload the SAM. Since a read transfer overwrites the entire SAM, this transfer cannot take place until after the data from the last location has been clocked out (read). However, the transfer must also be completed prior to the next serial clock edge. (The serial clock cannot be delayed or disabled while in the middle of a scan-line, or artifacts will appear on the screen.) In other words, the actual transfer of data must take place within one serial clock cycle (the cycle between the positive edge for the last piece of "old" data and the positive edge for the first piece of "new" data). Since a transfer cycle typically spans several serial clock cycles, it is necessary to know where within the transfer cycle the data is actually transferred, and to make sure that that portion of the transfer cycle falls within the window defined between the two specific clock edges. The REAL-TIME READ TRANSFER is specified for this purpose.

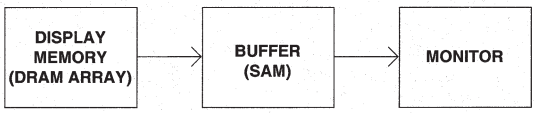

Figure 1
SAM BUFFER BETWEEN DISPLAY MEMORY AND MONITOR

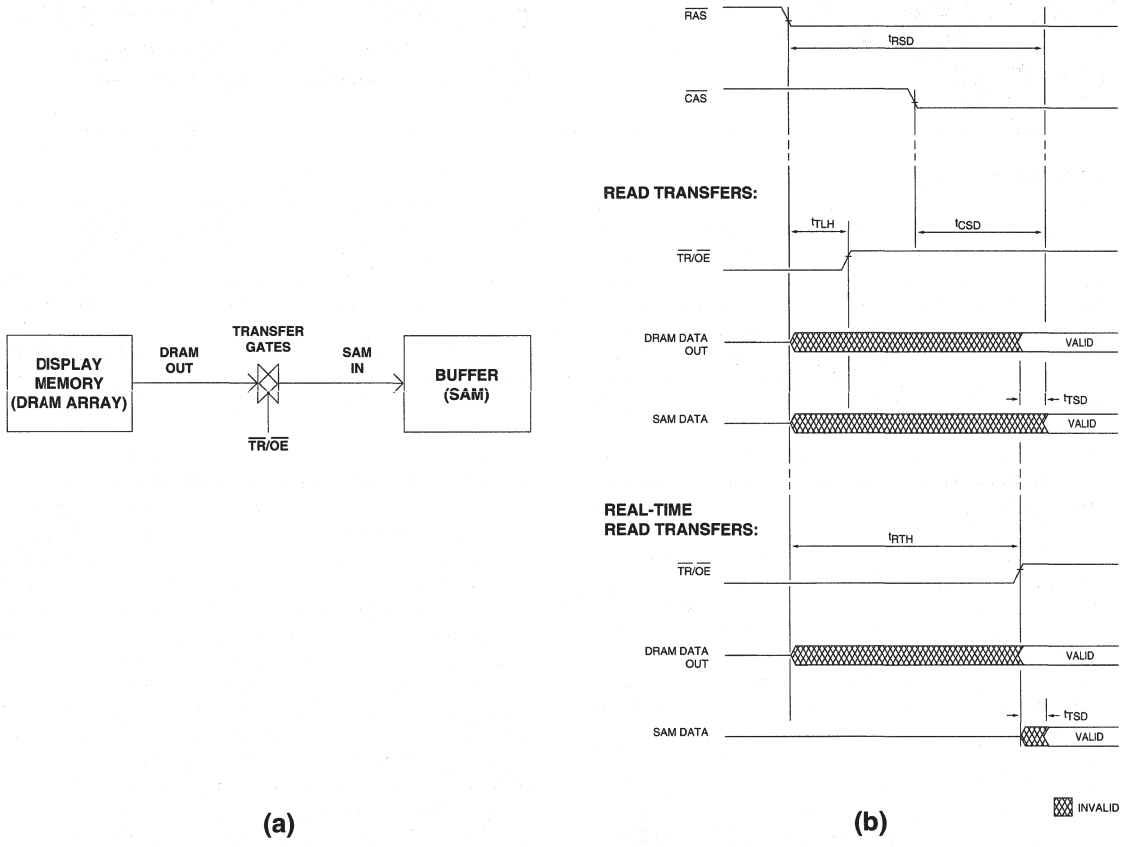

Figure 2
(a) CONTROLLING TRANSFERS BETWEEN DRAM AND SAM
(b) RELATED TIMING

Returning for a moment to "regular" READ TRANSFERs, $\overline{TR}/\overline{OE}$ goes HIGH "early" in the transfer cycle (sometime after ^{t}TLH is met, but before ^{t}RTH is met). When $\overline{TR}/\overline{OE}$ goes HIGH early in the cycle, the transfer gates between the DRAM and the SAM open before valid data has propagated completely through the DRAM circuitry (see Figure 2). So, new data reaches the SAM following the propagation delays through the DRAM circuitry and the transfer gates (^{t}RSD, ^{t}CSD and ^{t}TSD). The exact time at which new data arrives at the SAM is not specified (when $\overline{TR}/\overline{OE}$ goes HIGH before ^{t}RTH is met, the resulting transfer is sometimes referred to as "self-timed"). However, the maximum time is specified and is therefore the minimum time that should be met before supplying a positive edge on the serial clock input.

In contrast, in a REAL-TIME READ TRANSFER cycle $\overline{TR}/\overline{OE}$ goes HIGH "late" (after ^{t}RTH has been met). This way, the DRAM data is guaranteed to be at the transfer gates when they are opened and the only propagation delay is through the transfer gates (^{t}TSD), which thereby provide a shorter period of invalid SAM data. However, it is still a formidable task to fit this window within the desired clock cycle, and for this reason, SPLIT READ TRANSFERs were created.

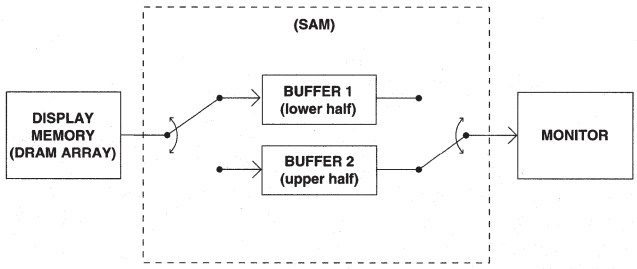

Figure 3
SPLIT BUFFER ARCHITECTURE FOR SPLIT READ TRANSFERS

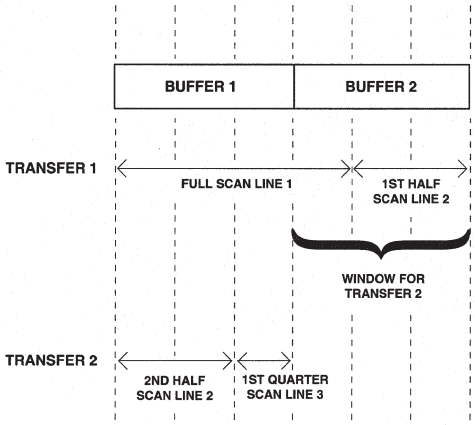

Figure 4
RELAXED TIMING CONSTRAINTS FOR SPLIT READ TRANSFERS

SPLIT READ TRANSFERS

When using SPLIT READ TRANSFERs, the SAM is effectively split into two buffers, each being half the length of the complete SAM. This provides double-buffered transfers between the DRAM and the SAM, with each buffer alternately functioning as the input buffer and the output buffer (data can be transferred into one buffer without corrupting the data being read out of the other buffer; see Figure 3).

This architecture calls for twice the number of transfer cycles, but provides less restrictive timing relative to REAL-TIME READ TRANSFERs. As an example of a case where midline transfers are required, consider a system configuration where one row in the DRAM array contains 1.5 scan lines of the display. A VRAM is always initialized for SPLIT READ TRANSFERs by executing a regular READ TRANSFER cycle first (see Transfer 1 in Figure 4). Here an entire row of the DRAM is transferred to the entire SAM. If the system continued using full READ TRANSFERS, a REAL-TIME READ TRANSFER would be required halfway through the second scan line of the display. However, with SPLIT READ TRANSFERS, the data for the remainder of the second scan line can be loaded in advance in buffer 1, thereby eliminating the seam that would otherwise exist. Specifically, one half of the next row in DRAM can be transferred to the SAM (Transfer 2 in Figure 4) any time after the SAM address counter passes from buffer 1 to buffer 2. This window is clearly much larger than the window for REAL-TIME READ TRANSFERs.

Then, after the SAM address counter wraps from the end of buffer 2 to the beginning of buffer 1, another half-row of data from the DRAM can be transferred to buffer 2. This process is continued for seamless operation of the SAM.

SUMMARY

Several types of read transfer cycles are available to transfer data from the DRAM array to the SAM portion of VRAMs. Each is suited to certain situations.

Regular READ TRANSFERs are used when each row in the DRAM array contains data for an integral number of display scan lines (or to initialize a VRAM prior to using SPLIT READ TRANSFERs). REAL-TIME READ TRANSFERs are used when midline transfers are needed (i.e. when a DRAM row contains a nonintegral number of scan lines). SPLIT READ TRANSFERs are also used when midline transfers are needed. The use of SPLIT READ TRANSFERs increases the number of transfer cycles executed, but avoids the tight timing requirements of REAL-TIME READ TRANSFERs.