



# Understanding Burst Mode Flash Memory Devices

## Application Note

Current AMD flash memory products operate with random access times ranging from 45 ns to 150 ns, though the most common applications are those using random access times of about 70 to 90 ns. The faster access times of 45 ns to 55 ns are only available today in lower density devices from 1 to 4 Mbits.

New applications increasingly need higher speed access, greater density, and lower voltages. However, higher density and lower voltage tend to reduce performance in a standard random access memory architecture. To achieve faster access times, architectural approaches such *burst mode* have been developed. This application note applies to the following AMD devices: Am29BL162C (16Mb) and Am29BL802C (8 Mb).

## WHY USE A BURST MODE PART?

burst mode devices offer improvements in system speed and performance by reducing sequential read access times. The burst read capability offers an average access time reduction of *more than 65 percent* for an eight-word sequential read at 40 MHz. In the normal asynchronous mode, the read access time is 65 ns per byte. For a burst mode operation, the initial access time is also 65 ns, followed by sequential byte transfers of 18 ns each.

## REQUIRED CONTROL PINS FOR BURST OPERATIONS

AMD burst mode devices require four extra control pins to operate.

### Load burst Address (LBA#)

LBA# indicates that the valid address is present on the address inputs.

*LBA# Low* at the rising edge of the clock latches the address on the address inputs into the burst mode flash device. Data becomes available  $t_{IACC}$  after the rising edge of the same clock that latches the address.

*LBA# High* indicates that the address is not valid.

### Burst Address Advance (BAA#)

BAA# increments the address during the burst mode operation.

*BAA# Low* enables the burst mode flash device to read from the next word when gated with the rising edge of

the clock. Data becomes available  $t_{BACC}$  ns of burst access time after the rising edge of the clock.

*BAA# High* prevents the rising edge of the clock from advancing the data to the next word output. The output data remains unchanged.

### End of Burst Indicator (IND#)

*IND# Low* indicates when the last word in the burst sequence is at the data outputs.

### Clock (CLK)

Clock input that can be tied to the system or microprocessor clock and provides the fundamental timing and internal operating frequency.

CLK latches input addresses in conjunction with LBA# input and increments the burst address with the BAA# input.

This implementation allows easy interface with minimal glue logic to a wide range of microprocessors/microcontrollers for high performance read operations.

## HOW DOES BURST MODE READ WORK?

AMD burst mode devices have two different read modes: random read and burst mode read.

### Random Read (Non-Burst Mode Read)

Random read is an asynchronous operation, and is how data is normally read from a standard flash memory device. A valid address must be placed on the address lines, and both CE# and OE# must be driven to  $V_{IL}$ . The valid data will be available on the data bus after a delay of  $t_{ACC}$ .

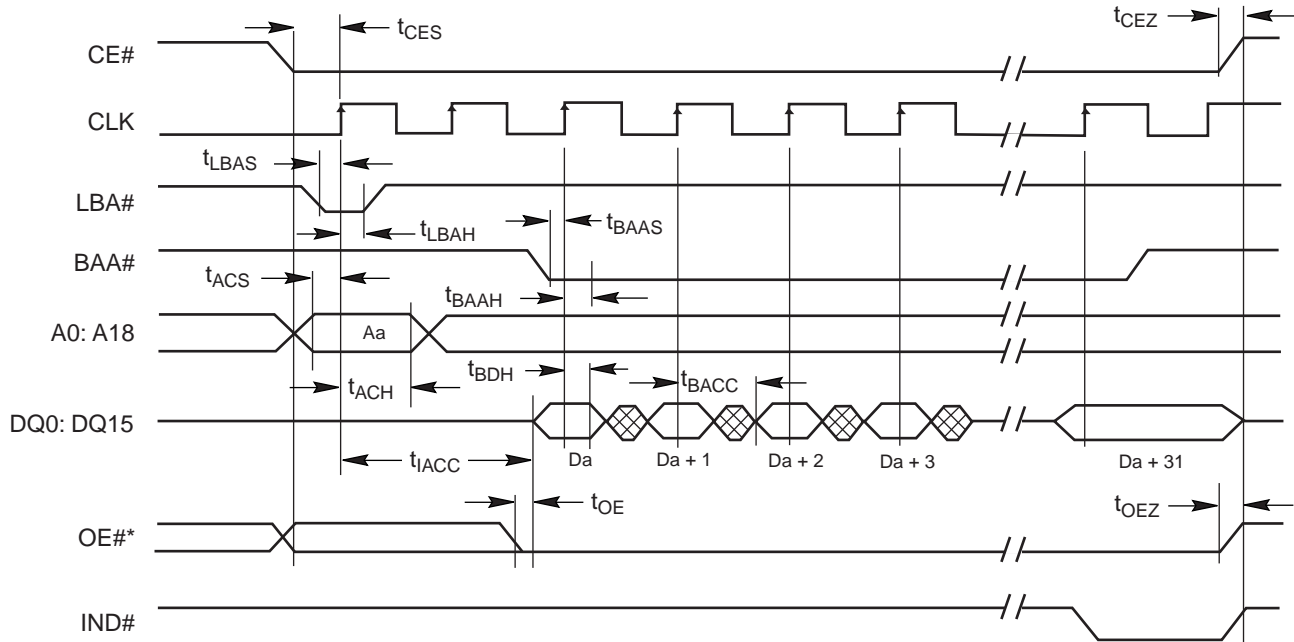
### Burst Mode Read

Burst mode read is a synchronous operation that is tied to the rising edge of a clock. The microprocessor/microcontroller supplies *only* the initial address to the device. In the linear mode, the device delivers a continuous sequential word stream starting at the specified word and wraps around when the end of the internal 5 bit address counter is reached (11111). For example, if the initial address is xxx0h, the data is 0-1-2-3...28-29-30-31-0-1...; if the initial address is xxxx2h, the data order is 2-3-4-5...28-29-30-31-0-1-2-

3... Data is repeated if more than 32 clocks are supplied.

All subsequent addresses are automatically generated by the device at the rising edge of subsequent clock cycles by the assertion of the BAA# signal. Figure 1 shows the timing diagram of a burst mode read operation.

cles by the assertion of the BAA# signal. Figure 1 shows the timing diagram of a burst mode read operation.



**Figure 1. Timing Diagram For Burst Mode Read Operation**

**Note:** Timing diagram is based on the Am29BL162C datasheet. For more information on this device, refer to this document, which is available at <http://www.amd.com/products/nvd/techdocs/22142.pdf>.

In Figure 1, notice how the various signals must act to achieve a burst mode read. After the first address (labeled “Aa”) is stabilized, the LBA# signal is driven low at the first rising clock edge. This loads the first burst address into the device. The LBA# signal is then driven high and the address lines are “don’t care” for the remainder of the burst mode read sequence.

After an access time of  $t_{ACC}$ , the data from the first address is available, and the device may now begin bursting the data out. The address is incremented by driving BAA# low (this is why the address lines are “don’t care”). The data is now available  $t_{BACC}$  after the previous read. At the end of the burst sequence, IND# goes low to indicate that the data is the last data in the sequence.

**Is There A Way To Suspend The Burst Mode Operation?**

AMD burst mode devices are capable of Burst Suspend and Burst Resume operations. The device enters the Burst Suspend mode when BAA# is deasserted (taken high). This means that the device will hold the data that was being presented at the outputs when the device was put into Burst Suspend operation. The presentation of data on the system data bus is independent

of the OE# signal. The burst operation can be resumed by asserting BAA#. Asserting BAA# on the rising edge of the CLK will increment the counter and present the next subsequent data at the outputs after the specified  $t_{BACC}$  time.

**Terminating A Burst Mode Read**

There are two ways to terminate a burst mode read operation.

1. Taking the RESET# pin low will reset the device, and it will default into reading array data in asynchronous mode.
2. Writing the Burst Disable command sequence will exit the device out of burst mode operation.
  1. Activating the LBA# (Low) will terminate the previous burst read cycle and start a new burst read cycle with the address that is currently valid.

**Is The Burst Data Always Available  $t_{BACC}$  After The Previous Read?**

While in burst mode operation, data will always be available  $t_{BACC}$  after the previous read. However, this data may not always be read from the next address. Recall that BAA# must be held low to increment the ad-

dress being read. Otherwise, the data being output will be the same as the previously output data.

For example, if the host system drives BAA# high for one clock cycle, the device outputs the data from the same address for two cycles. This may be desirable in certain situations where the processor did not successfully read the data from the device.

### Example: Comparing A Burst Mode Device To A Standard Device

By comparing a burst mode device and a standard device, the true advantage in speed can clearly be seen. Assume that System A uses a burst mode flash memory device, and that System B uses a comparable standard flash memory device. Both systems have the same processor, and both flash devices have the same random access time. Table 1 shows how much time is required by both systems to read eight consecutive bytes from memory.

**Table 1. Chart Comparing Read Times Between burst and Standard Devices**

	<b>System A: Am29BL162C</b>	<b>System B: Comparable standard device</b>
Byte 0	65 ns ( $t_{ACC}$ )	65 ns ( $t_{ACC}$ )
Byte 1	18 ns ( $t_{ACC}$ )	65 ns
Byte 2	18 ns	65 ns
Byte 3	18 ns	65 ns
Byte 4	18 ns	65 ns
Byte 5	18 ns	65 ns
Byte 6	18 ns	65 ns
Byte 7	18 ns	65 ns
<b>Total time</b>	<b>191 ns</b>	<b>520 ns</b>

Both systems require a  $t_{ACC}$  of 65 ns to do the first random access read. However, System A can now do sequential burst reads in 18 ns, whereas System B still requires 65 ns to complete each and every read command. By adding all the access times together, you can see that System A, using a burst mode memory device, can read almost 3 times as fast as System B.

### Is The Speed Improvement Always Constant?

Although the previous example shows that a burst mode memory device can be dramatically faster than a standard flash, the improvement is not always con-

stant. The more the system reads in burst mode, the faster the average access time. The less the system reads in burst mode, the slower the average access time. In order to obtain optimum performance from a burst mode device, the system's program should be structured such that the device can perform as many sequential reads as possible.

### What About Program And Erase Operations?

Program and Erase commands work on one word/byte or sector at a time, just as they do in standard flash memories.

## What Are The Requirements On A System Using A Burst Mode Flash?

The system must provide the three extra control pins: LBA#, BAA#, and IND#, as well as a clock to synchronize the burst reads. The system must also recognize how long the burst sequences are. For the Am29BL162C, the burst sequence is 32 words long and is used to support microprocessors that implement an instruction pre-fetch queue, and to support large data transfers during system configuration. The AMD burst mode devices are designed to work easily with processors that support burst mode data. Examples of burst-compatible processors include:

- *Motorola PowerPC family* (example: MPC555)
- *Motorola 68000 series* (example: Excalibur)

- Siemens Tri-Core Architecture
- Hitachi SH-4

## SUMMARY

burst mode memory devices have been developed to increase performance of flash memories. By reducing to groups of sequential words, a burst mode memory device can significantly increase system performance. Using a burst mode memory device can improve performance while holding or reducing cost. But, the degree of performance improvement will depend on system software taking advantage of the burst address boundaries. This is done by locating code branches and data sequences at the beginning of the burst group boundaries, so that as many words as possible can be read in burst mode.

### Trademarks

Copyright © 2000 Advanced Micro Devices, Inc. All rights reserved.

AMD, the AMD logo, and combinations thereof are registered trademarks of Advanced Micro Devices, Inc.

Expressflash is a trademark of Advanced Micro Devices, Inc.