

Techniques to Improve Time-to-Insight Debugging High-speed Memory Failures

By

Jennie Grosslight, Agilent Technologies

jennie_grosslight@agilent.com

Intermittent memory failures can be perplexing to debug. While some teams achieve rapid resolution debugging memory failures, many teams struggle debugging intermittent failures.

This article outlines a debug methodology and introduces techniques applicable to memory failures in DDR and DDR2 systems (including the DRAM side of Fully Buffered DIMM). The basic debug methodology for determining the root cause of an intermittent memory failure has four phases.

First, determine if the failure is repeatable. Try to duplicate the conditions of the failure. Duplication of a failure often provides valuable insight into the nature of the problem.

Second, connect a logic analyzer to the memory bus with a probe or interposer to gain rapid insight into timing relationships of the entire DDR2 bus, parts-per-million errors, clock quality, and bus protocol.

Third, on the logic analyzer, run SW tests to provide insight into smallest data valid windows, protocol errors, refresh rates, and precharge intervals.

Fourth, conduct parametric measurements using a high-performance scope with high bandwidth probing.

Phase 1

Duplicating conditions can be challenging. Keep in mind that the root cause of the problem can come from a sub-system or applications that are not directly connected to memory. LAN access, power sequences of subsystems, entering and exiting sleep modes, and power cycles can be important to consider when evaluating memory failures.

Isolation of a problem during a specific test or set of conditions makes the problem easier to evaluate. With a repeatable failure, you have the ability to take multiple measurements under the conditions of the failure.

Details to consider include reviewing error logs and identifying what software was running at the time of the failure. Environmental variants can also impact system failures. What was the room temperature when the system failed? Check the airflow to system.

Hardware considerations are numerous.

Is the power to the system within specifications?

Has a system of this same design ever passed validation tests?
Do other systems fail or is this failure unit specific?
What are the revisions on the board, DIMM, processor, or other components of the failed system?
How does the failed system differ from working systems?
Have there been recent component changes in manufacturing?

If conditions are repeatable, run your tests under those conditions, if not chose a robust memory test and vary the test conditions, such as temperature and power supply limits, in a methodical manner.

Phase 2

A logic analyzer with a DDR probe, interposer, or direct attach provides rapid insight across the entire DDR bus. Logic analyzer systems offer state traces with protocol decode to translate commands for functional validation. Simultaneous to the state capture, there are 64k deep traces of high-resolution timing analysis across the entire DDR bus. High-resolution eye measurements on logic analyzers make it possible to identify parts per million errors.

Global markers (up to 1024) can be set automatically from search functions. The global markers track between waveform and listing windows to allow for different views of suspect areas.

Colorized filters on the logic analyzer can help visualize problems. Color filtering enables engineers to use pattern recognition while viewing waveforms to recognize areas that require further investigation.

Phase 3

Measurements from logic analyzer traces can be automated with an advanced customization environment like the Agilent B4606A on 16900 Series logic analyzer systems. Benefits of software analysis tools are additional insight and automation of repetitive tasks and measurements.

As an example, one automated measurement showed us that the average data valid window of data bursts on one system was 2.1ns. However, there were two data signals with average data valid windows of 500ps. Such a dramatic variation in data valid windows is a concern. Instead of looking at all data signals with a scope, we concentrated on the data signals with the smallest data valid windows.

Another example of an automated SW routine saving time is shown in figures 1, 2, and 3. Here the SW script on the logic analyzer was checking for functional errors in logic analyzer traces while running repetitive captures. Every time an error occurred, the trace was saved for review. Email notification of the error was optional.

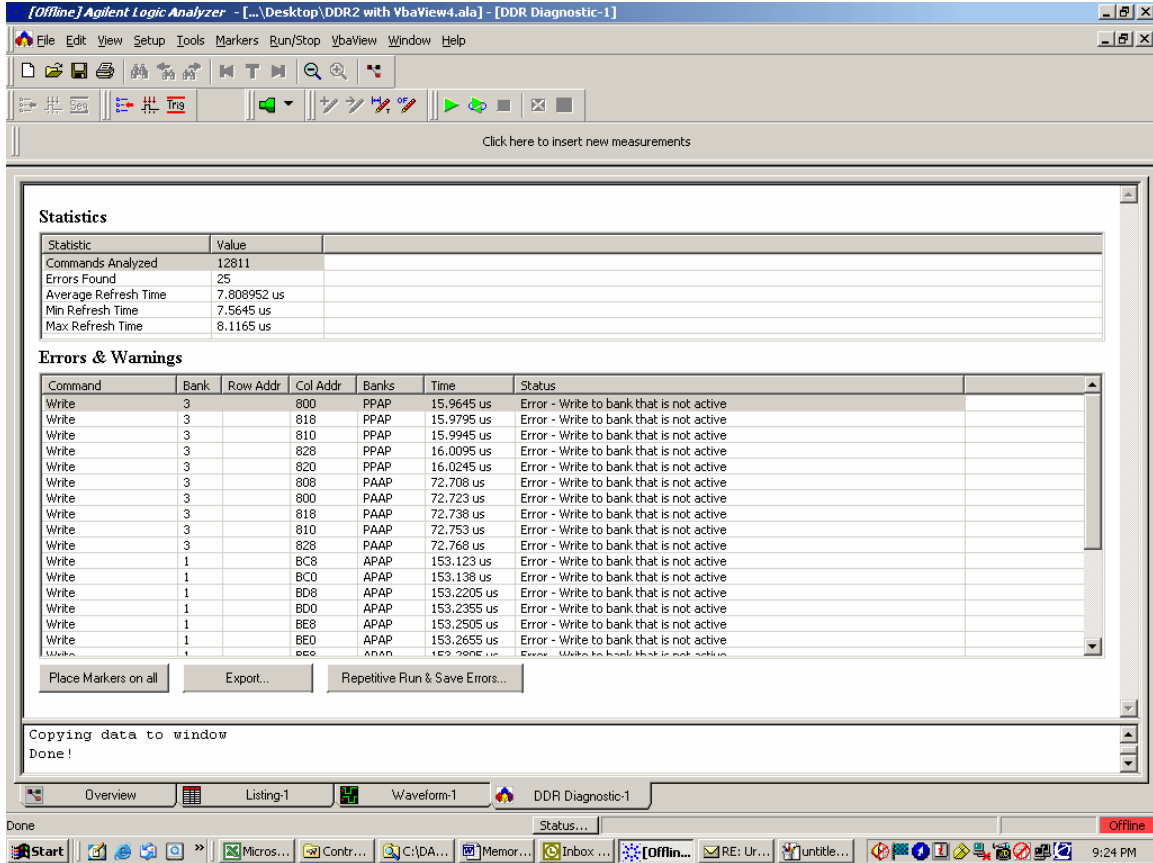


Figure 1: Error report provides summary and details.

The error report in figure 1 indicates that 12,811 commands were analyzed in the trace. There were 25 errors recorded. All errors were Writes to a bank that was not active. A marker was placed at each error. Click on any error and the global markers track in the listing and waveform windows to help you investigate the problem. Scrolling through the errors, we see that the errors were not bank specific.

Sample Number	Time	COMMAND	CommandClk	ADDR	DATA31-0	DATA63-32	READdata6
8494.7							
8495	15.9285 us		1	0 8000 0000	0000 0001	0000	0000
8496	15.9315 us		0	0 8000 0000	0000 0001	0000	0000
8497	15.9330 us		1	0 8000 0000	0000 0001	0000	0000
8498	15.9345 us	Precharge: Bank 3	0	0 8002 220A	0000 0001	0000	0000
8499	15.9360 us		1	0 B1B3 2FBA	0003 0C01	030B	030B
8500	15.9390 us		0	0 FFFD FFFF	030B 7EFF	030B	030B
8501	15.9405 us		1	0 FFF7 FFFF	030F 7EFF	030F	030F
8502	15.9420 us	Write	0	0 FFDF FFFF	030F 7EFF	030F	030F
8502.1		Bank = 2					
8502.2		Address = 398 3F0					
8502.3		Data = 030F7EFF DFFFFFFF CB = 0					
8502.4		Data = 030F7EFE 7FFFFFFF CB = 0					
8502.5		Data = 030F7EFB FFFFFFFD CB = 0					
8502.6		Data = 030F7EEF FFFFFFF7 CB = 0					
8502.7							
8503	15.9435 us		1	0 FF7F FFFF	020F 7EFF	020F	020F
8504	15.9465 us		0	0 FDFF FFFF	030F 7EFF	030F	030F
8505	15.9480 us		1	0 F7FF FFFF	030F 7EFF	030F	030F
8506	15.9495 us		0	0 DFFF FFFF	030F 7EFF	030F	030F
8507	15.9510 us		1	0 7FFF FFFF	030F 7EFE	030F	030F
8508	15.9540 us		0	0 FFFF FFFD	030F 7EFB	030F	030F
8509	15.9555 us		1	0 FFFF FFF7	030F 7EEF	030F	030F
8510	15.9570 us		0	0 FFFF FFDF	030F 7EBF	030F	030F
8511	15.9585 us		1	0 FFFF FF7F	030F 7EFF	030F	030F
8512	15.9615 us		0	0 FFFF FDFF	030F 7AFF	030F	030F
8513	15.9630 us		1	0 FFFF F7FF	030F 6EFF	030F	030F
8514	15.9645 us	Write	0	0 FFFF DFFF	030F 3EFF	030F	030F
8514.1		Bank = 3					
8514.2		Address = 3B8 800					
8514.3		Data = 030F3EFF FFFF7FFF CB = 0					
8514.4		Data = 030E3EFF FFFF7FFF CB = 0					
8514.5		Data = 030F7EFB FFFFFFFD CB = 0					
8514.6		Data = 030F7EEF FFFFFFF7 CB = 0					
8514.7							
8515	15.9660 us		1	0 FFFF 7FFF	030E 7EFF	030E	030E
8516	15.9690 us		0	0 FFFF 7FFF	030F 3EFF	030E	030E
8517	15.9705 us		1	0 FFFF 7FFF	030E 7EFF	030F	030F
8518	15.9720 us		0	0 FFFF 7FFF	030F 3EFF	030E	030E
8519	15.9735 us		1	0 FFFF 7FFF	030E 3EFF	030F	030F
8520	15.9765 us		0	0 FFFF FFFD	030F 7EFB	030F	030F

Figure 2: Listing displays Precharge of Bank 3 without activating Bank 3 prior to Write to Bank 3.

Viewing the listing, shown in figure 2, we see that Bank 3 was precharged prior to the Write without being activated. Further investigation of the errors using the listing and waveform windows showed that not all Writes were in error, errors occurred intermittently on all banks.

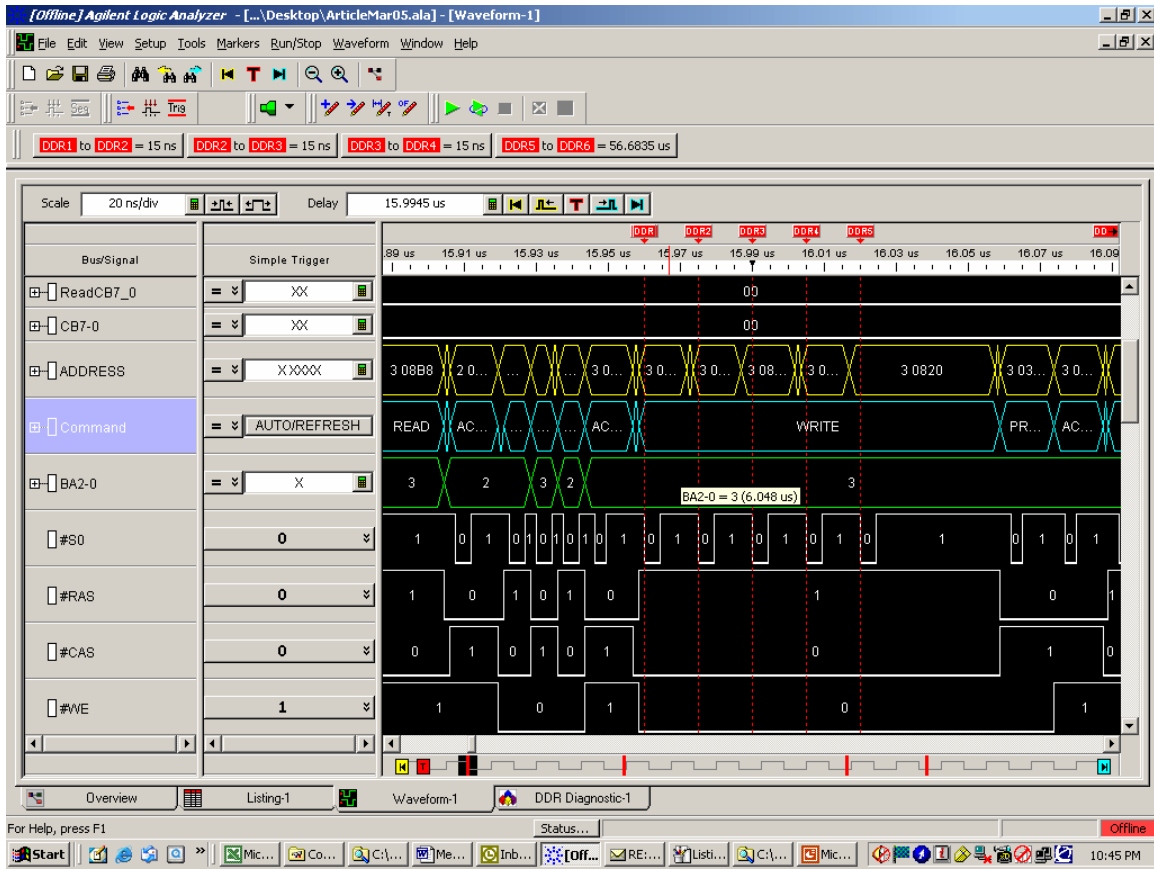


Figure 3: Observing the errors in the waveform helps to visualize the error patterns.

In figure 3, marker measurements in the waveform show a series of 5 evenly spaced errors followed by a long delay until the next error. The gray bar under the black waveform window indicates which section of the trace is being viewed (shown in black) and where the errors occur on the trace. Red markers are set at all errors.

Phase 4

Parametric measurements of high-speed signals require flat response, high bandwidth, and high sampling rate. Accurate parametric measurements are like a chain, where the lowest performance component in the measurement system will limit the bandwidth of the measurement. Choose your scope and probing combination to provide enough bandwidth to cover the 5th harmonic of your data rate for the most accurate parametric measurements. Sample rate will also impact your measurement accuracy. 20Gs/s is excellent for DDR and DDR2.

Characterization measurements include, Ts/Th, rise time, clock overshoot, frequency, and jitter.

Jitter analysis packages, eye measurements, eye masks, and eye unfolding software provide details into signal behavior.

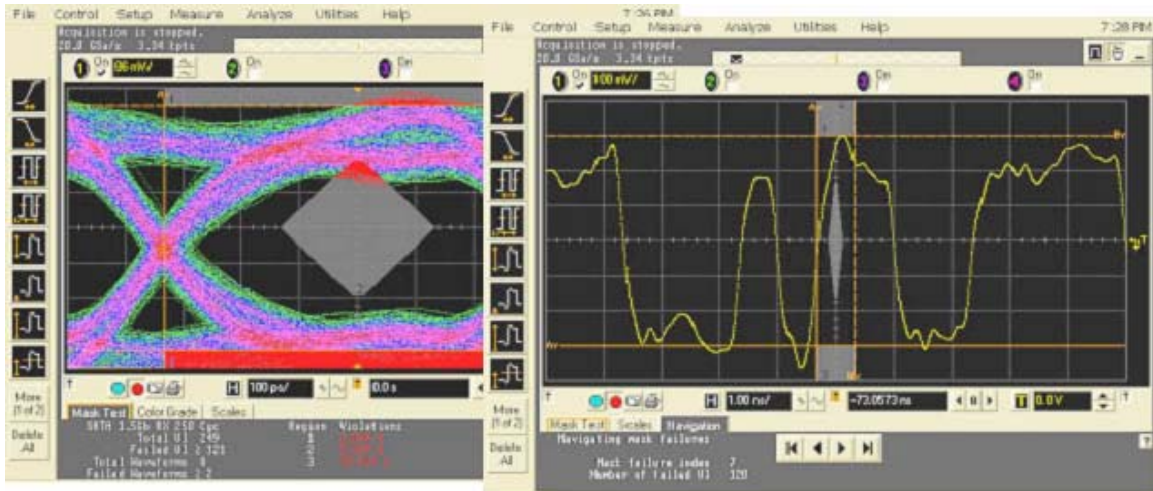


Figure 4: Eye mask identify violations in Clock, Data, or Strobe eyes. Eye unfolding provides the trace details surrounding a violation.

On the left side of figure 4 there is an eye mask with violations on the topside of the eye. Unfolding the eye as shown on the right of figure 4, displays the trace at the point(s) of failure. From the trace we can consider the data pattern preceding the failure and observe signal characteristics such as rise time, ringing, and overshoot.

Probe placement is critical to making accurate parametric measurements for signal characterization. Probe READ data and strobes at the memory controller. Probe WRITE data and strobes at the SDRAM.

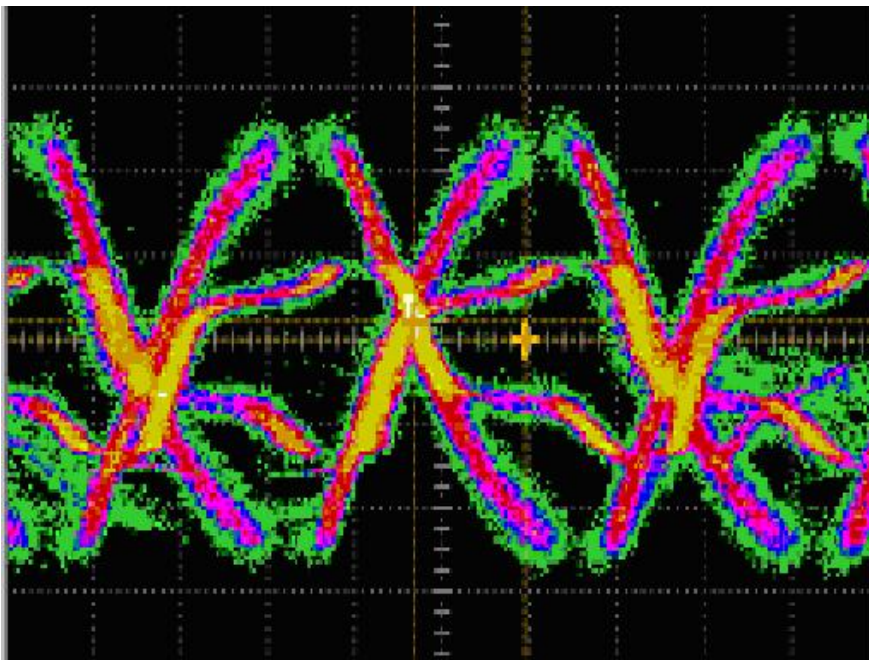


Figure 5: View of READ and WRITE strobes depends on probing location.

Figure 5 is an eye measurement of DQS0 relative to the rising and falling edges of DQS5 at $T=0$. The measurement was taken at an interposer in the DIMM slot and illustrates the importance of probe placement for parametric measurements. The eye for WRITE strobes is large and well shaped. The probe location on the interposer is close enough to the SDRAM that the signal is clear of reflections.

READ strobes are degraded from reflections at the interposer. The eye is adequate for relative measurements of pulse width as seen with a logic analyzer. However the position on the bus is inadequate for actual characterization of the READ traffic.

For an accurate view of the READ data, as seen by the memory controller, miniature scope probe tips need to be placed at the memory controller.

Many memory technology leaders validate and debug high-speed memory systems using the tools and techniques described in this article. Engineers who use time saving tools reap the rewards of faster debug and greater insight into system performance.

###

Author biography:

Jennie Grosslight graduated from the University of Colorado in Colorado Springs, Colorado with a Bachelors Degree in electrical engineering. She has 15 years of experience at Agilent Technologies (Hewlett-Packard) in the areas of system engineering, high-speed hardware design and validation, product marketing, application support, and project management. Today Jennie is a program development manager for logic analyzer

support of computer systems. She has one daughter and enjoys outdoor activities in her spare time.