

Towards a Qualification Data Set: Expanded SEE Data on the P2020 Processor

Steven M. Guertin, *Member, IEEE*, and Sergeh Vartanian

Abstract— Earlier P2020 SEE data are compared and expanded to a recent die revision, significantly increasing samples tested by protons by five devices, and by heavy ions by five devices. Earlier tested SEE types are found to be fairly similar in register, L1 cache, L2 cache, and CPU crashes. New test methods give SEE performance for the flash memory controller, watchdog circuit, and a built-in Ethernet port on the P2020 processor. Results from heavy ion and proton tests are presented, with data separated over a large number of specific error types and test programs.

I. INTRODUCTION

The P2020 processor is used in several space programs, and is available from Space Micro in the Proton 400k-L flight computer [1]. The processor uses an architecture that is very similar to the RAD750 [2] – a radiation hardened specialty processor for space, but is a much newer and more powerful 45 nm dual core device that is built for the commercial market. Similar testing has previously been reported [3], but in the current case the increase in data is significant and the data are now taken across multiple die revisions of the P2020. Thus the present data may be combined with the earlier data to create a more complete data set, and enable programs to make a more compelling argument for qualification of these devices.

We tested five Freescale (now Qualcomm) P2020RDB-PCA boards for proton single-event effects (SEE). We also tested five different P2020RDB-PCA boards for heavy ion SEE. Testing was performed over five test trips including the Tri-University Meson Facility (TRIUMF), Massachusetts General Hospital (MGH), Lawrence Berkeley National Lab (LBNL), and Texas A&M University (TAMU).

This workshop proceeds as follows. We discuss the test setup in section II. We follow this with proton results in section III, then heavy ion results in section IV. The

Manuscript received August 23, 2017. The research in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Reference herein to any specific commercial product, process, or service by trade name, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology. This work was supported by NASA flight projects.

Steven M. Guertin is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109 (USA), phone: 818-393-6895, e-mail: steven.m.guertin@jpl.nasa.gov.

Sergeh Vartanian is with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109 (USA), phone: 818-354-1390, e-mail: sergeh.vartanian@jpl.nasa.gov.

© 2017 California Institute of Technology. Government sponsorship acknowledged.

workshop conclusion then follows in section V.

II. TEST SETUP

A. Test Boards and Support Equipment

Testing was performed using the P2020RDB-PCA. The test boards were used to provide bias to the DUT because earlier work showed no significant risk of power-related issues when irradiating the processor. Each P2020RDB-PCA was connected to a test interface computer that provided two universal asynchronous receiver transmitter (UART) ports. Each central processing unit (CPU) was provided its own UART for direct communication of results during testing.

For proton testing there was no special preparation. But for heavy ion testing it was necessary to remove the copper heatspreader over the DUT. Our approach was to mill down the DUT to expose the epoxy underneath. For clarification of linear energy transfer values, we further removed the filler epoxy inside the heatspreader by using acid. Initial efforts were somewhat unreliable, and we determined that it was necessary to attempt to remove the epoxy before any particular board (DUT) was powered for a length of time and the epoxy hardened.

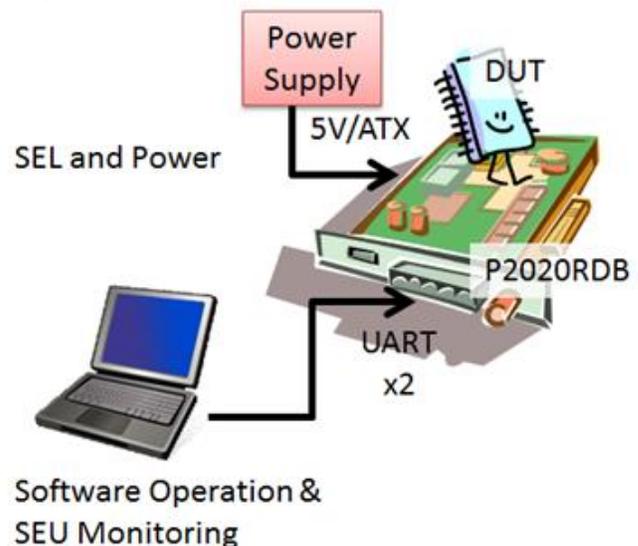


Figure 1: Schematic of the test system. Note that the actual power supply was the P2020RDB-PCA power unit. The system was not monitored for SEL, and none was expected.

Testing was performed on the P2020RDB-PCA using a custom hardware setup that allowed for a second computer to interact with the P2020, with one UART (universal asynchronous receiver transmitter) port assigned to each of

the e500 cores in the P2020. The arrangement of test hardware is given in Fig. 1.

B. Event Types

We have established the following as event types that can be identified by our test software and hardware during SEE exposure:

- 1) *Register SBU – a single bit changes in a processor register*
- 2) *Register MBU – a register changes its value completely –note that all observed register MBUs were consistent with complete value changes, rather than two- or three- bit errors*
- 3) *L1 invalidates – an L1 cache line (with parity protection disabled) is lost. We believe this occurs when the disabled parity protection system encounters a parity error and interprets it as a cache miss.*
- 4) *L1 SBU – this is a reported parity error when parity is enabled in the L1 cache. This causes a detected loss of data.*
- 5) *L1 parity invalidations – this is when a parity protected L1 cache loses a valid line of data with no other warning. This causes silent data corruption (SDC).*
- 6) *L2 SBU – a bit error observed in L2 data (we test L2 with EDAC disabled, and as regular memory – so that tag errors are not tested)*
- 7) *External memory errors*
- 8) *Watchdog – this is a general error in the watchdog system (incorrect watchdog timeout, incorrect change in watchdog status)*
- 9) *Ethernet packet error – this is when a packet is received by, or transmitted from, the DUT with an error in it.*
- 10) *Flash Memory – errors reading or writing flash memory using external debugging tools*
- 11) *Crashes – this is when a CPU core fails with no other indication of problems. Sometimes an exception occurs before a crash, but in those cases it is expected that a crash will not occur (to be clear, if the exception is not properly handled, then the processor may crash after the exception, but this is a software problem rather than an inherent hardware SEE sensitivity.*
- 12) *Strange Events – All other event types that could result in undetectable erroneous operation are individually indicated. Usually these are actually register SBUs that impact how a given test runs, but in some cases the behaviors may indicate a different problem. We are not able to differentiate anything other than a potential SBU sensitivity.*

C. Test Algorithms and Methods

In this section provide information on the test algorithms and methods used to test for the event types indicated in the previous section.

Register tests were performed by writing a known value to a target register or set of registers, then the algorithm waits a given period, after which the registers should be read to

observe SEEs and then the test repeats. This test is performed by writing values to registers before the beam is turned on; the beam is then turned on and fluence is collected on the DUT while it holds static data; and finally, after the beam is turned off, the stored values can be read to determine errors. This testing collects SBUs and MBUs in registers. Register testing was also performed using an external debugger that could directly read and write the registers, with essentially the same sort of test approach, however the results are only provided in a qualitative format.

L1 cache testing was performed in three ways. First, when parity protection is disabled and the cache is tested by loading a known value and having the cache sit idle during exposure. Second, when parity protection is enabled and the test algorithm is identical (cache is inactive during exposure). And, third, by actively using the cache during irradiation – by constantly writing and reading from the cache.

When parity protection is disabled, cache lines are invalidated when a parity event is detected. (This is an interpretation of the events we observe – specifically, we observe no actual bit errors in the caches when parity is disabled. Instead, we observe cache lines invalidated with a cross section that is consistent with the sum of the cross sections for all the bits in a given cache line). We also observed exceptions when parity protection was enabled (but these exceptions caused the test program to restart the test loop, and undercounts events). With the parity protection enabled, these events are expected to be capturable by the operating system (OS) – and if the OS is configured for write-through caches (a mode that is almost never used), these events will be silently repaired and the user will be unaffected. We also observed invalidations when the parity protection was enabled, and we expect these would cause bad data to get into a running program with no notification to the user or operating system. These cache tests are performed in the same manner as the register tests. Namely, the given memory range is loaded with a known data pattern, then it is left to idle while the beam is applied to the DUT, periodically, the system is told to wake up and check the cache for errors, then refresh the data in the cache.

The L2 cache was tested in cache as RAM mode, with the built-in error detection and correction (EDAC) disabled, so that bit errors could be observed. This mode disables the cache tag information, which limits the collection of cache state error data, however, we interpret the cache state bits to be similar in SEE sensitivity to the data bits. The L2 cache testing is performed with the same type of algorithm as the tests for the L1 cache. Namely, the cache is loaded with a known value; the software is then put in an idle mode; the DUT is irradiated; and after irradiation is complete, the DUT is interrogated to determine what SEEs were collected.

External memory interfaces were tested with one test program that performs writing and reading of external memory constantly during irradiation. No external memory errors were ever observed during this test, but the crash sensitivity of the processor increased significantly (see below). This test is activated and allowed to run

continuously during exposure. If any events occur, exceptions are logged and can details be retrieved provided the test software does not crash. If a crash occurs, it almost always requires immediately stopping the beam exposure and performing algorithms to attempt to recover the DUT. In practice, it is usually necessary to at least reset the DUT, however we usually power-cycled the DUT after crashes.

The watchdog system was tested in a few different configurations, observing different versions of machine checks related to time-outs on the watchdog system. If a timeout or full watchdog error is erroneously observed, it is an error. The test algorithm for this testing involved configuring the timeouts for two levels of watchdog machine checks. We then observed if the configuration/control of the watchdog indicated proper transitions of watchdog state during irradiation. Any unexpected or missed watchdog exceptions were recorded.

The Ethernet system was tested by configuring a port to send UDP packets which were picked up by a second computer and checked for incorrect data. Ethernet was tested at two speeds – 237 kbps and 44 Mbps. Any errors in UDP packets were recorded as errors. Lost UDP packets were ignored because they never exceeded 1% of packets, which was indistinguishable from rate observed without radiation present. Ethernet testing was performed by allowing U-Boot to send UDP packets and then hijacking the Ethernet and DMA settings used in order to repeatedly send duplicate packets. The Ethernet test ran continuously during exposure, and any significant delay in data transfer would be noticeable.

Flash memory was tested by using an external debugger to perform flash writing or reading during exposure. Prior to use, a known test file was uploaded to the flash memory. Reading during exposure was checked for errors by comparing the read file to the source file. Similarly, after writing during exposure, we read back, after exposure, to see if the writing resulted in any errors. These tests were entirely depended up on the operation of the BDI3000 debugger which was used to control the Flash controller on the P2020.

Crashes are a general condition where a CPU stops interacting with the test system. In this work, a crash is possible during any other test, and what it actually means is context-dependent. We collect crash data from every test algorithm described here. The actual cause of a crash is not known (this is somewhat different than other test groups may define, because this P2020 testing is the result of a significant amount of software development, and if we know what is causing the crash we design the software to not be vulnerable to that sensitivity – within reason). We expect these, and a large percentage of the other errors described here, would result in single event functional interrupts (SEFIs) in real systems, but we do not try to separate out SEFIs in this work because the results would depend heavily on the exact test program used. In fact, for some of the data reported here, crashes are split into two types – those occurring during external memory testing, and those occurring in every other test. When actively stressing external memory, crashes are more than an order of magnitude more common.

Strange events are tested in a similar way to crashes. All test algorithms were used to collect strange events. In general, we do not try to distinguish strange events for the purposes of developing SEE rates. Instead, we indicate the qualitative behavior during a strange event and indicate if it is consistent with a predicted vulnerability of the code (for example, loops used to control execution duration can easily be altered by a bit flip in a significant bit, causing a dramatic increase or decrease in loop execution duration).

III. PROTON RESULTS

A. Register SBU

The sensitivity of the registers to SBUs is shown in Figure 2. As with most of the results in this section, the data are presented with boards presented side-by-side, and include results from earlier work [3] for comparison.

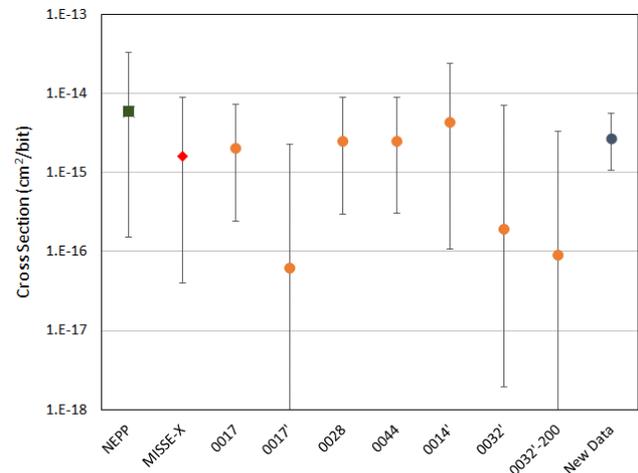


Figure 2: Register SBU cross sections for P2020 all test boards and old data. Note that all five new boards are presented, with ' indicating testing at MGH. Other points were taken at TRIUMF. "0032'-200" was taken at MGH with 200 MeV protons. All others were taken at 100 MeV. The NEPP and MISSE-X points come from earlier work which was previously reported in [3].

B. Register MBU

The sensitivity of the registers to proton-induced MBUs is shown in Figure 3.

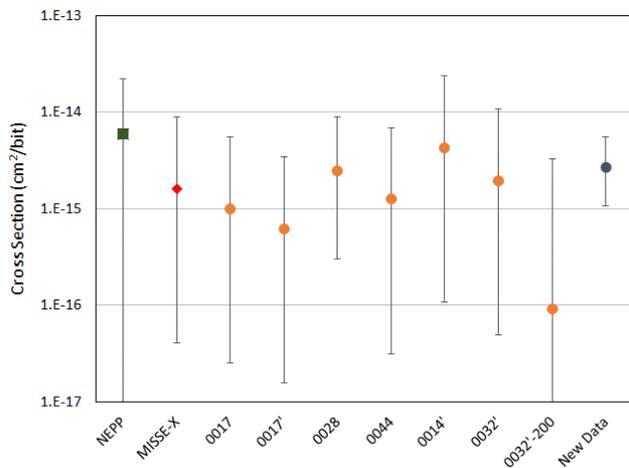


Figure 3: Register MBU cross sections for P2020 all test boards and old data. See the caption in Fig. 2 for more information on the different test boards and conditions.

C. L1 Cache Invalidations – Parity Disabled

The sensitivity of the L1 cache to proton-induced line invalidations when parity protection is not enabled is shown in Figure 4.

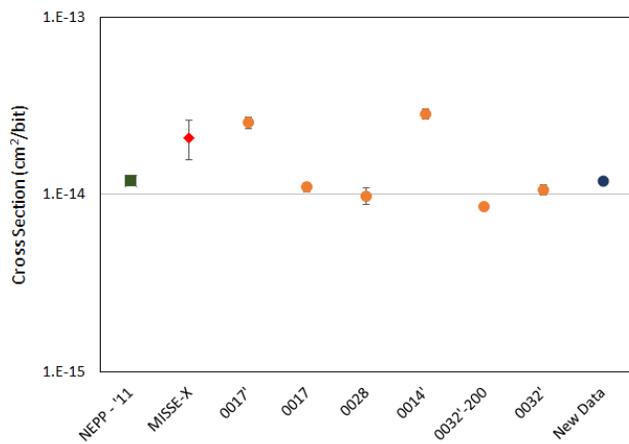


Figure 4: Sensitivity of P2020 to cache line invalidations when parity is not enabled. See Fig.2 for information about the different boards and test conditions.

D. L1 SBUs – Parity Enabled

The sensitivity of the L1 cache invalidations and parity SBU events are compared in Fig. 5.

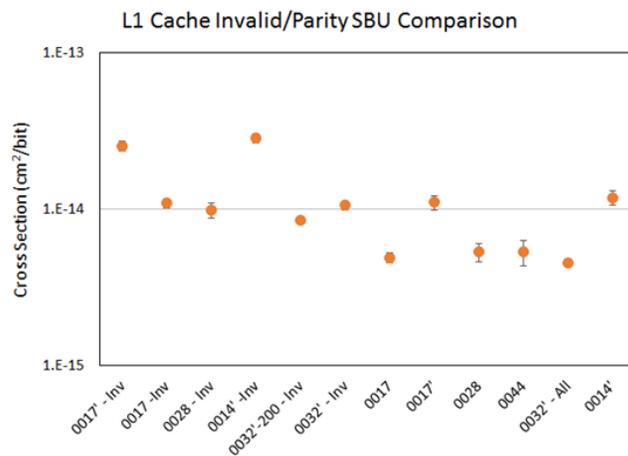


Figure 5: The L1 Cache SBU cross section, as invalidations (left points) and parity events (right points). Prime marks, ', indicate data taken at MGH. Board 0017 data from MGH at multiple energies combined in the parity SBU data analysis.

E. L1 Cache Invalidations – Parity Enabled

The sensitivity of the L1 cache to proton-induced line invalidations when parity protection is enabled is shown in 6. These events would lead to undetected erroneous operation of a real system.

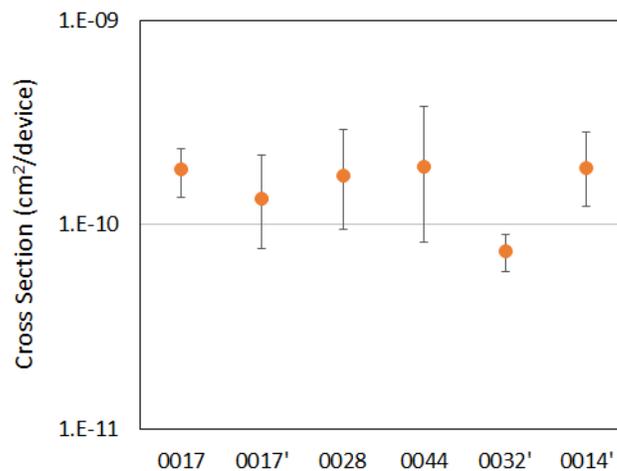


Figure 6: Sensitivity of L1 cache to line invalidations when parity is enabled. See Fig. 2 for information about the different boards and test conditions.

F. L2 Cache SBUs

The cross sections for bit upsets on each of the five test boards are compared, both by facility (primed points are from MGH, unprimed from TRIUMF), and to the earlier NEPP and MISSE-X data in Fig. 7.

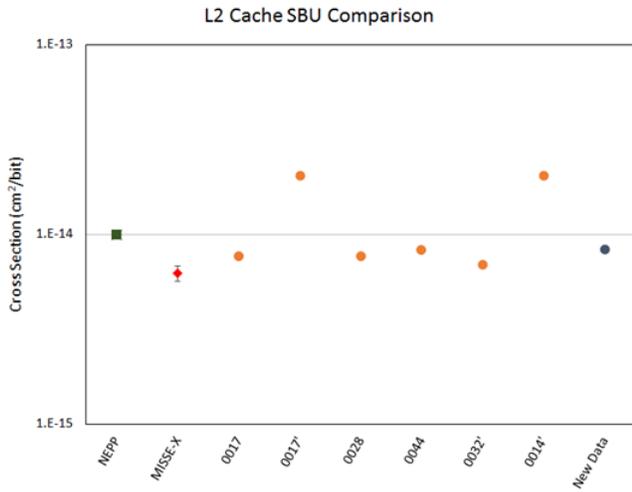


Figure 7: The cross section for bit upsets in the L2 cache is shown for the three test boards, compared to the earlier NEPP and MISSE-X results. Note that for most data points, the error bars are smaller than the data points. Data from MGH is indicated with a prime mark, '. MGH data from 0017 combine 100 and 200 MeV results.

G. CPU Crashes

After the events listed above, a high occurrence event type is for a CPU core to crash, resulting in no communication from that CPU core until it is reset or power-cycled (we routinely power-cycled, but usually reset is sufficient to restart). For this error type, we observed a significant difference depending on whether or not we were running the test program that performs external memory reads and writes during exposure. The sensitivity for these events to protons is presented in Figure 8. Note the number of test cases and the clustering of the results. The memory and non-memory tests are then combined to provide the “total-“ points. This indicates the memory tests are 50-100x more prone to crashes than other test types. We believe these are more indicative of an actual system with high memory usage (such as a standard operating system).

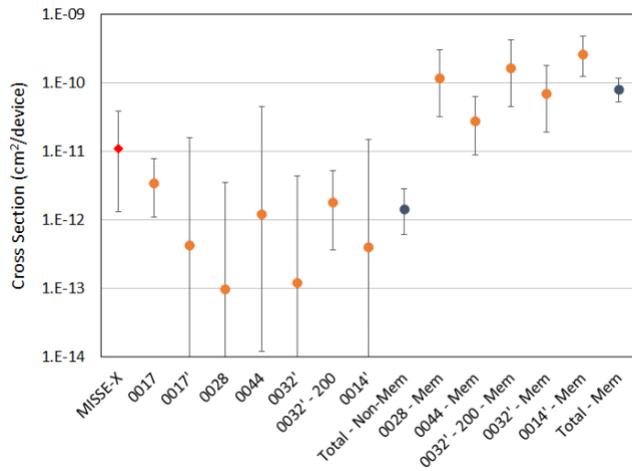


Figure 8: Sensitivity of P2020 DUTs to crashes. In this case some of the data were taken while running small test programs with minimal external memory requirements. These are the Non-mem points. The remaining data were taken while running an off-chip memory test. The off-chip access dramatically increases the device crash sensitivity.

H. Strange Events

The final workshop will also provide data on the sensitivity of the P2020 to strange events, which are very similar to crashes, but have other symptoms than crashes. The primary quality of these events that we wish to highlight is that they are believed to cause incorrect behavior without being identifiable by any error protection.

IV. HEAVY ION RESULTS

A. Register Sensitivity

Heavy ion register sensitivity was measured and is shown below in Fig. 9. Note that the large error bars (spanning four orders of magnitude) are actually situations with 0 counts that could not be cleanly shown on this figure. Three new boards' (0033, A1, and A3) data re compared to old data taken for the NEPP program.

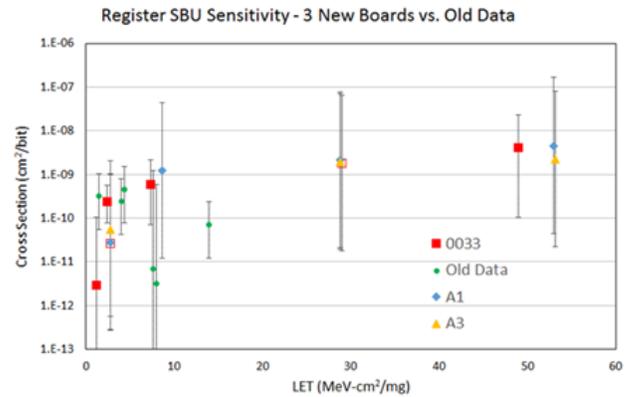


Figure 9: Register SBU sensitivity for P2020 is shown, comparing three new boards to old data. Note that the large error bars (four orders of magnitude) reflect conditions with no counts (0 events).

B. L1 Cache Sensitivity

L1 cache sensitivity is shown in Fig. 10. Note this also compares L2 cache sensitivity, and block errors, which are the same as line invalidations with parity enabled. Block errors are plotted as device cross section, while L1 and L2 bit sensitivity is plotted as per-bit cross section.

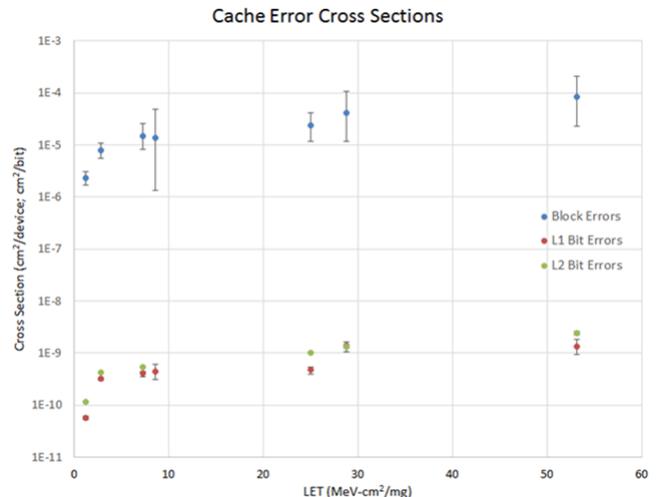


Figure 10: Comparison of L1 and L2 bit sensitivity and L1 block errors (line invalidations).

C. L2 Cache Sensitivity

The sensitivity of the L2 cache bits during this testing is presented in Figure 11, shown separately from the previous figure. Also in the figure is a comparison to the earlier reported data [3].

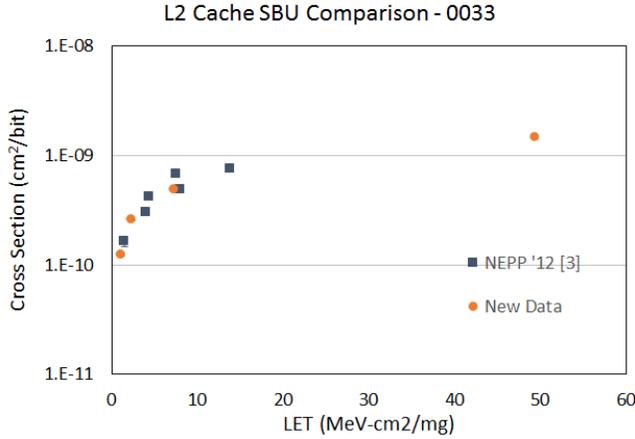


Figure 11: L2 Cache SBU sensitivity to heavy ions for new data and old data. The results are essentially identical. Note that error bars are smaller than the plotting symbols.

D. Flash Sensitivity

No events were observed during testing of the Flash memory controller (keep in mind that only the P2020 processor was irradiated). The limiting cross section for Flash memory sensitivity is given in Fig. 12, below. All data points represent only limiting cross section.

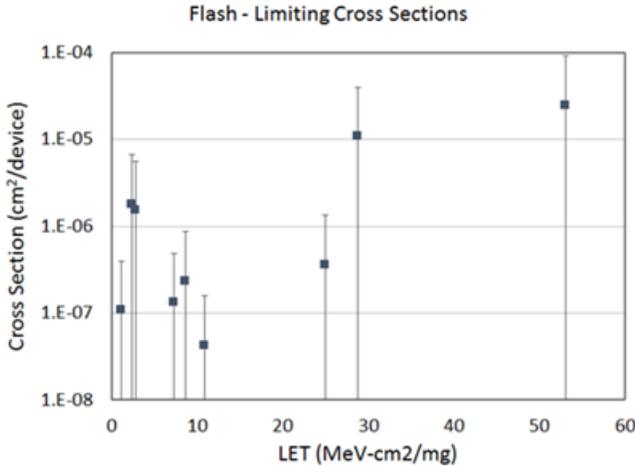


Figure 12: The limiting cross section for possible Flash controller sensitivity (no events were seen to affect the Flash memory).

E. Watchdog Sensitivity

Watchdog testing showed no discernable watchdog events. The data taken show a limiting cross section of less than 1×10^{-5} cm²/device.

F. Ethernet Sensitivity

During testing, no errors were observed during Ethernet packet transfers at any speed. We did lose approximately 1-3% of all packets that were sent, however this number did not significantly change with and without the beam. Packet loss

is a standard behavior of Ethernet protocols. In a TCP/IP system, packet loss is masked by the protocol, which automatically requests resending of dropped packets. We used UDP, which provides packets as-transmitted, and lost packets are simply lost.

The primary limitation of the Ethernet results was the signal to noise ratio. In this case, the noise was the crash rate of the test devices. Ethernet errors were significantly below the crash sensitivity. This is shown in Fig. 13, which indicates, in the data points, the cross section for device crashes. The region below the data points is the limiting cross section for packet corruption (of any kind), normalized to packets sent, which can be observed to be on the order of 100,000s of packets sent during testing, with no errors. Although it is possible to improve detection of the signal (lost packets) by increasing the total amount of testing performed, it is unclear how to do this in a useful way, because increasing the beam flux actually increases the noise while reducing the signal. Ideally, we would collect data over a very long time with increased overall fluence, but not flux. However, this is not practical.

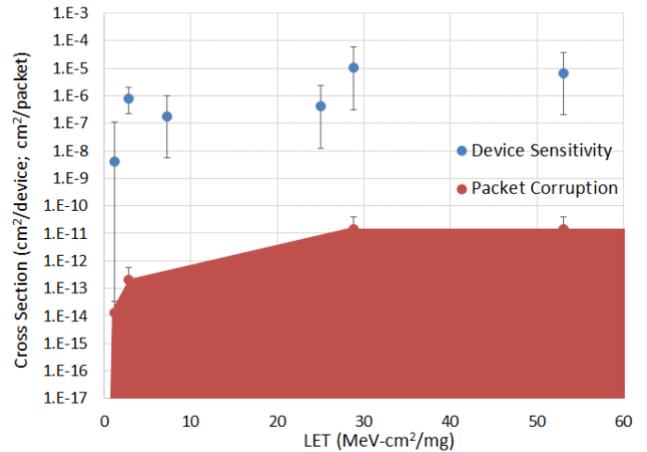


Figure 13: The crash sensitivity of the P2020 is compared to the limiting cross section for packet corruption. This is somewhat complicated by the fact that the cross section per packet should be corrected by the packet up time.

G. Other Results

Heavy ion results will also be presented in the full workshop for registers, L1 cache, watchdog timers, crashes, and rare events.

V. CONCLUSION

Although the P2020 processor has undergone a revision between previously reported testing and the present testing, the SEE sensitivity is essentially unchanged. The present data set increases the amount of data available to the point that a more viable qualification approach may be taken, though the present data is not a qualification dataset.

The heavy ion results for this effort are summarized below, with highest tested LET, and cross section at highest LET. Note that the onset for all events was observed to be 1 MeV-cm²/mg, with most events having a clear onset, or being

statistically limited at all LETs so that no good onset could be established.

1. Register SBU: 1.10×10^{-9} cm²/bit
2. Register MBU: 1.10×10^{-9} cm²/bit
3. L1 Cache Invalidates: 1.51×10^{-9} cm²/bit
4. L1 Cache Parity Events: 2.23×10^{-9} cm²/bit
5. L1 Cache Invalid w/Parity: 9.75×10^{-5} cm²/device
6. L2 SBU: 3.10×10^{-9} cm²/bit
7. Flash 1×10^{-5} cm²/device
8. Watchdog 1×10^{-5} cm²/device
9. Ethernet 1×10^{-11} cm²/packet
10. Crash Sensitivity (estimated) 10^{-5} - 10^{-3} cm²/device
11. Strange Events (estimated) 10^{-5} cm²/device

The proton results for this effort are summarized below (cross sections do not significantly change between 50 and 200 MeV):

1. Register SBU: 2.69×10^{-15} cm²/bit
2. Register MBU: 2.69×10^{-15} cm²/bit
3. L1 Cache Invalidates: 1.19×10^{-14} cm²/bit
4. L1 Cache Parity Events: 1.19×10^{-15} cm²/bit
5. L1 Cache Invalid w/Parity: 1.09×10^{-10} cm²/device
6. L2 SBU: 8.35×10^{-15} cm²/bit
7. Crash Sensitivity (regular): 1.41×10^{-12} cm²/device
8. Crash Sensitivity (memory): 7.90×10^{-11} cm²/device
9. Strange Events: 2.60×10^{-12} cm²/device

VI. ACKNOWLEDGMENT

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. This work was supported by the NASA Electronic Parts and Packaging (NEPP) program. The authors would like to thank Jeff George and Rocky Koga of The Aerospace Corporation for support with beam availability and discussions. Government sponsorship is acknowledged.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

VII. REFERENCES

- [1] "Proton 400k-L™ Single Board Computer," <http://www.spacemicro.com/assets/datasheets/digital/slices/proton400k.pdf>, (accessed 1/2017).
- [2] BAE Systems "Processors" brochure, <http://www.baesystems.com/en-us/download-en-us/20151109141738/1434554723043>, (accessed 1/2017).
- [3] S. M. Guertin, M. Amrbar, "SEE Test Results for P2020 and P5020 Freescale Processors", Radiation Data Workshop, IEEE, 2014