SSC21-WKII-07

Design Simulation and Testing of a Custom Co-Processor for Cubesatellites in LEO

Justin Heimerl, Allen Spain, Alexandria Lin, David Cotten
The University of Georgia Small Satellite Research Laboratory
1510 Cedar St, Athens, GA 30602, 770-314-6040
jph83405@uga.edu

ABSTRACT

An arms race in the last decade between computing performance and power-reduction has led to accelerated progress in both hardware and software, provisioning fast, efficient algorithms and small, capable, low-power hardware devices. Meanwhile, small satellites have become increasingly important in defense and commercial missions, due to their low resource consumption in terms of power, space, and cost. Improvements in computer vision algorithms and commercial off-the-shelf (COTS) edge computing introduce a means of addressing challenges posed for spacecraft in the realm of autonomy and perception. The University of Georgia's Small Satellite Research Laboratory has been working to address these obstacles in developing an interface to merge Graphics Processing Unit (GPU) computations into the standard PC104+ satellite stack. This paper presents the Core GPU Interface (CORGI), a hardware solution which integrates the NVIDIA Jetson TX2/TX2i module into the cube satellite stack. The CORGI can be used as a standalone flight computer or as a co-processor with a designated onboard computer (OBC). This board will fly on the Multi-view Onboard Computational Imager (MOCI), a 6U satellite scheduled for launch into low earth orbit (LEO) in 2022, and will serve NASA's efforts to demonstrate autonomy and high-performance computing on small satellites. The CORGI provides input/output capability for command and telemetry and development interfaces for increased usability, specifically DisplayPort and two USB 3.0 type-A interfaces. Additionally, a UART umbilical provides an interface between an off-PCB computer through the PC104+ stack, enabling the TX2i to be externally triggered for science data handoff by a radiation-tolerant onboard computer. The CORGI is designed as a payload processor, and thus utilizes the TX2i module's USB 3.0 hub to connect one or more satellite imagers over USB. The CORGI provides other standard development access points, including general-purpose input-output (GPIOs), from both the designated onboard computer and TX2i module. Standard headers also provide a means of testing the TX2i line voltages for discharge monitoring and power status. An SD card provides flash memory for logging science data and telemetry. This paper presents 1) PCB design and manufacturing specifications to assist teams in designing hardware for space applications. 2) Spice simulations to validate power management circuit design and evaluation for relevant cases. 3) Electrical test results demonstrate the nominal operation of both power management and regulation circuitry and the validation of correct discharge behavior during a power outage. 4) We present tests that demonstrate CORGI's operation and performance characteristics under load.

INTRODUCTION

Background

NASA's 2015 Roadmap has identified current computing solutions' inability to provide real-time or near real-time data analysis.[5] Computing platforms capable of real-time image and video processing and autonomous decision making and mobility would advance this effort. Deep learning and artificial intelligence could be critical in space applications by leaving less decision-making up to the ground operations team. Others have echoed this sentiment, such as the Air Force Research Laboratory (AFRL), in collaboration with NASA to build next-generation flight computing systems for dynamic and harsh environments.[12] The Defense Advanced Research Projects Agency (DARPA) has moved towards capable small spacecraft in their F6 initiative for autonomous communications and collaboration.[15]

Traditional terrestrial applications rely on TCP/IP connected hardware devices with inexpensive and widely available COTS GPUs. Bridging the gap between terrestrial and space-based applications is crucial to furthering space-based exploration and

autonomy and will require the systematic validation of higher-performance computers in space environments. Both NASA and the Air Force have taken part in encouraging universities to take part in small satellite technology development through programs like the NASA Undergraduate Student Instrument Project (USIP) and the Air Force Research Lab's (AFRL) University Nanosatellite Program (UNP). USIP is described as an Educational Flight Opportunity (EFO), encouraging universities to develop payload technology, and thus assisting in expanding the collective understanding of COTS components performance in space.

Figure 1: The Multi-view Onboard Computational Imager's Core GPU Interface Board

In the past twenty years, the number of launches for nanosatellites (1-10kg) and microsatellites (10-100kg) per year is enabled by the smallsat development process due to their small size, power, and relatively short mission timelines.[11] This combination of improved computing for spacecraft and the increasing utility of small spacecraft introduces new scientific research and educational opportunities in the domain of small satellite compatible High-Performance spaceflight computing (HPSC) platforms.

System-on-Chip (SOC) computers have been identified as a promising computing solution – integrating CPUs for sequential processing tasks, such as IO and command and data handling, and GPUs or FPGAs for handling high volume parallel processing tasks. Commonly used SOCs for spacecraft are the SmartFusion 2 SoC, combining an ARM M3 and the Xilinx UltraScale+ MPSoC. NASA's NEPP program has also been working to prove NVIDIA SoC tolerance in space environments.[19] In industry, NVIDIA Jetson computers are being used but their performance in LEO remains an open question.

Other custom CubeSatellite hardware solutions have emerged such as PyCubed, which deploys an ARM M4 MCU running MicroPython, and provides a well-documented hardware and software development platform for educational projects and general purpose CubeSatellite Missions.[13] Other efforts to create a flexible platform include NASAs PhoneSat, which used low-cost mobile processors, such as the Nexus One and Nexus S for data handling, attitude determination, communication, and imaging.[16] Efforts of merging a high level of usability with high-performance computing in a small form factor remains an ongoing engineering and research effort for university programs.

Computing platforms are several generations behind the state-of-the-art acting and are the limiting factor in overall spacecraft design, limiting sensing resolution and speed. The CORGI was designed to provide a technology demonstration of small, flexible, and computationally-capable hardware platforms for Earth Observation in Low Earth Orbit.

Project Overview

Small satellite data gathering capabilities have advanced significantly in recent years, but the capability of transmitting large amounts of data back to the ground for further processing and analysis has not necessarily kept up. One solution to this bottleneck is by performing this post-processing onboard the satellite in situ before transmitting back the downsized results/data products. GPUs are well-suited to be implemented for this solution, due to their powerful parallel-processing nature that can carry out heavy computations more efficiently and quickly than standard CPUs or OBCs on smallsats. However, putting accelerated computing platforms in space presents a challenge; the vast majority of GPUs are not designed for space, much less have a tested, space-rated carrier board that can easily integrate into a CubeSat. The CORGI is one such attempt at overcoming this challenge that conforms to the PC104+ form factor – popular for CubeSats – and exposes important power and communication peripherals that aim to facilitate development, integration, and testing of a modular GPU with little flight heritage (the Jetson TX2i) in a satellite vehicle, either as a full-fledged OBC or a co-processor for offloaded computations.

The TX2i's internal power management integrated circuitry (PMIC) is sensitive to changes in input power. Consequently, on-board power regulation is included to provide clean power to the TX2i module, Overvoltage Protection (OVP), Undervoltage Protection (UVP), and limit the supply voltage to the specified voltage and current limits. In the

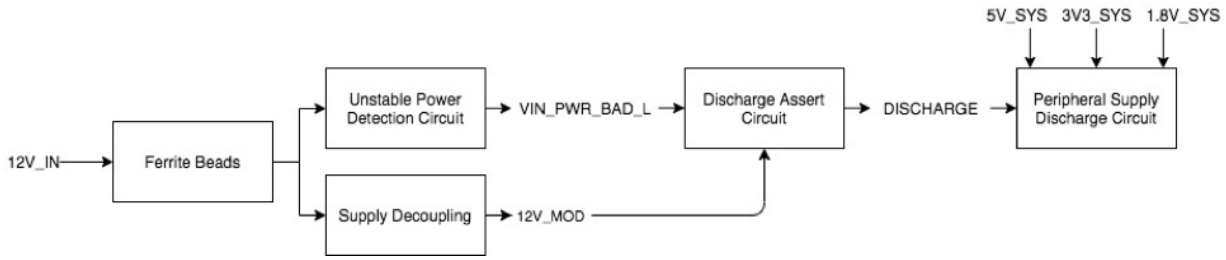

Figure 2: CORGI power management block diagram

design of CORGI, careful consideration is given to the board layout to meet NVIDIA design specifications. Impedance matched differential pairs are used for high speed signalling to mitigate crosstalk. Additionally, termination resistors between differential pairs reduce common mode noise, and those differential paired signal transitions are kept to a minimum in order to avoid pair length mismatch. The CORGI PCB is an 8-layer stackup which consists of three ground planes to provide a solid ground connection and improve signal quality.[14] The power planes circumscribe the dedicated rail voltages for the purpose of layout efficiency and to supply stable power to the CORGI components and the TX2i module. Efficient layout planning can prevent excessive via creation and reduce overhead costs at the board house.

The TX2i module requires clean and stable supply voltages, as well as dedicated management circuitry to communicate the carrier's power state, provide power to peripheral circuitry, and boot into the operating system. To minimize trips to the board house, the power up/down and discharge circuits were simulated in Analog Devices' circuit simulator LTSpice. The power up/down circuits were modeled to assess power mode logic sent from the CORGI to the TX2i. The power discharge circuit was simulated to validate each voltage rail's complete and gradual discharge from its nominal voltage to ground. The discharge circuit's dampened discharge prevents high currents from being drawn from the satellite battery in the event of a power outage or when the TX2i is instructed to transition to a power-saving mode by the onboard computer. Electrical acceptance testing of the CORGI is carried out which incorporates a basic electrical validation testing suite including: visual inspection, trace continuity tests, nominal voltage tests, and probing of flags sent from the CORGI carrier board to the TX2i. Once electrically validated the final component, the TX2i module, can be integrated onto the CORGI by way of an 8x50 SAMTEC connector.

Once integrated, UART two way communication was validated in brief, followed by serial communication benchmark performance testing. To demonstrate CORGI's communication capability, file transfer benchmarks were used, whereby three large data files were sent over UART to an external computer.

MOCI utilized the Jetson TX2i for accelerated computing making use of the 256 core NVIDIA Pascal GPU for image reconstruction and object detection. Mission-specific computing demands were tested using the nineteen layer convolutional neural network VGG-19. GPU tests were run while integrated on the CORGI, and resultant power and temperature traces were recorded for the duration of the task. Complete integration into the flatsat is still underway, but integration with the MOCI satellite battery pack and OBC were successfully demonstrated.

SYSTEM DESIGN

Power

The TX2i module requires power handling circuitry on the custom board to function correctly. The custom board must provide proper decoupling, noise immunity, and discharge circuitry for the system to be fully reliable. First and foremost, a clean (low ripple) 12V signal must be applied to the module. Our system creates this signal by utilizing a large power plane on the board and approximately 360 µF of bypass capacitance from the rail to a continuous ground plane. The 12V signal comes from either an external power supply or Clyde Space 40Whr batteries. We also include MPZ1608S300ATAH0 ferrite beads in series with the 12V input in order to provide a lossy medium for high frequency noise that may couple into the 12V bus from radio frequency signals, clocks, high speed digital data lines, or various other mediums.

The two circuit outputs are `VIN_PWR_BAD_L` and `12V_MOD`. The supply `12V_MOD` is a clean 12V signal, while `VIN_PWR_BAD_L` is a signal required by the

TX2i that it uses to determine if unstable power is present at the input. If present, the device will not power on.

The other power circuitry required by the TX2i is a voltage discharge circuit for peripheral power buses (5V, 3V3, and 1.8V) so that the carrier board gracefully powers down when the module is turned off or the 12V power signal is removed. Both the voltage input circuit and the discharge circuit were simulated to ensure intended behavior. The results can be seen in section Simulation Results. In addition to the simulation, they were also physically verified on assembled hardware. The results of that test can be seen in section in the Results Section.

When power is removed, or invalid power is detected, the discharge net asserts, causing a controlled power down on all peripheral rails. The peripheral rails are used to carry out various helper functions, such as logic shifting, voltage regulation, current monitoring, or power switching. A block diagram of the input power handling sequence can be seen in the figure 2.

Peripheral rails are `5V_SYS`, `3V3_SYS`, and `1.8V_SYS`. `5V_SYS` and `3V3_SYS` are created by load switching the 5V and 3V3 rails generated by the spacecraft Electrical Power System (EPS) on the PC104+ header. The control signal is generated by the TX2i, and asserts when the module is powered on, indicating that the carrier board can begin power sequencing. The `1.8V_SYS` rail is created by a TPS54318RT switch mode buck converter.

Communications

Currently, the system allows access to one of the TX2i UART busses via the PC104 header, and two of the UART busses via a debug header. The UART bus on the PC104 header is how data is transferred between the Clyde Space On Board Computer (OBC) and the TX2i. The baud rate is 115200 with an 8N1 data format. UART allows simple interaction between the OBC, which lacks any high speed ports such as USB, and the TX2i. Further revisions of the system will allow access to the TX2i SPI interface, allowing for faster file transfer speeds.

Peripherals

The final revision of the board will contain the following peripherals, with appropriate support circuitry:

- 2x USB C 3.0
- 1x HDMI
- 1x SD Card
- 1x SPI Interface (through PC104+)
- 1x UART Interface (through PC104+)

For the MOCI satellite, the dual USB ports allow access to the dual imager payload, allowing for full utilization of the GPU to carry out Structure from Motion (SfM) in orbit.[9] For other missions, however, USB could be used to interact with science instruments, external processors, or communication systems.

When implementing USB, or other high speed differential protocols, it is recommended to obtain equipment to produce an eye diagram of the system. This allows verification of signal and trace integrity. See[3] for information on how to produce the eye diagram. Unfortunately, this equipment tends to be prohibitively expensive for university teams, so a best practices approach can also be taken when designing high speed signal traces (see Board Design).

HDMI should be selected de facto when designing a carrier board for the TX2i. While a similar hardware design methodology is used for both HDMI and DisplayPort (See High Speed Signaling Schematic Guidelines), the factory Ubuntu distribution from NVIDIA has HDMI enabled in the device tree, and DisplayPort disabled. The authors have found it challenging to modify the device tree to enable DisplayPort in current JetPack (NVIDIA SDK) versions due to a lack of documentation on the Linux system.

An SD card slot is included to allow ease of modifying Linux parameters, as many applications require or benefit from a custom/lightweight Linux distribution, such as Petalinux or Yocto. The SD slot also allows for mass storage for applications that require access to large amounts of data such as traditional machine learning algorithms or image processing applications.

Layout

USB, HDMI, and DisplayPort are all examples of differential paired high speed buses with data rates in the range of many gigabits/sec. These high speed digital signals require special attention from both a schematic and board layout perspective in order to achieve proper signal integrity and meet rise time requirements. Some general guidelines for these signals can be followed however. The most typical set of components to include in a schematic are AC coupling capacitors with low ESR (Equivalent Series Resistance) and ESL (Equivalent Series Inductance),

common mode rejection filters, and ESD (Electrostatic Discharge) protection, typically in the form of transient voltage suppression (TVS) diodes. AC coupling capacitors can serve various roles such as removal of DC offset or coupling between different logic levels. The lowest possible ESR and ESL values should be selected, as the capacitor does act as an impedance discontinuity on the transmission line and can cause reflections, degrading signal integrity. The CORGI utilizes 0402 X7R capacitors for all AC coupling, if possible smaller capacitors should be used. Capacitors should be placed as close as physically possible to the signal source in order to minimize reflections. Common mode chokes can be utilized to improve EMI (Electromagnetic Interference) measurements by filtering out high frequency noise. They do, however, degrade signal integrity by their inductive nature, and are by default not utilized in this system. If EMI is observed to be an issue then they can be integrated as a solution.[6] ESD protection should use very low capacitance TVS diodes. The CORGI uses the RCLAMP524PA, which has a typical capacitance of .3pF. Unlike common mode chokes, the protection provided by TVS diodes is well worth the small loss in signal integrity in most cases.

Certain care should be taken when laying out a board containing various power nets and high speed RF or digital signals. The first concern when doing board layout of this sort should be signal integrity as signal integrity is entirely dependent on correct layout and stackup practices. For reference, the CORGI uses 1 oz copper on external layers, .5oz copper on internal layers, FR4 dielectric with a reported dielectric constant ϵ_r of 3.9 - 4.9 (at DC to 10GHz), a dielectric thickness of 0.184 mm between external to internal layers and .20 mm between internal layers. The PCB stackup is as follows:

Table 1: CORGI PCB Stackup

Layer #	Purpose
1	High Speed Signals
2	GND
3	Low Speed Signals
4	GND
5	POWER
6	Low Speed Signals
7	GND
8	High Speed Signals

To provide clean power to the TX2i, as well as peripheral circuitry large, copper pours are utilized on layer 5, with different sections of the board containing different voltage domains. All circuitry that utilizes a domain is placed above or near the pour. This allows vias directly to the plane and keeps power traces short, minimizing series inductance (loop area) and resistance. Special care is taken to provide a solid reference plane for high speed signals. When routing high speed signals the reference plane (where return current flows) should be solid copper with no breaks or discontinuities. High speed return current will take the path of least impedance, and therefore inductance, and this is typically along the primary current carrying trace. A break in the ground plane will be seen as a large inductive discontinuity to the signal, and will greatly degrade signal integrity.[14] CORGI does utilize up to two vias per trace on some of the USB signals. However, this is unlikely to cause any real signal integrity issue. All vias used on the CORGI are drilled with a hole size of 12 mil and an annular ring diameter of 28 mil. Equation 1 below can be used to approximate parasitic capacitance of one via.

$$C = 1.41 * \epsilon_r * T * D_1/(D_2 - D_1) \quad (1)$$

Where T is the PCB thickness, D_1 is the hole size, and D_2 is the diameter of the annular ring. Plugging in $\epsilon_r = 4.3$ for FR-4, $T = 1.6mm$, $D_1 = 12$ mil, and $D_2 = 28mil$ gives a capacitance of approximately $0.3pF$, the USB trace is a microstrip transmission line with $Z_o = 45\Omega$ The 10-90 % rise time delay for a step signal can be computed by Equation 2.

$$T_{10-90} = 2.2 * C * (Z_o/2) = 14.8ps. \quad (2)$$

USB 3.0 specifies a maximum delay on compliant transmission media of $200ps$,[4] indicating that a small number of vias are very unlikely to cause signal degradation to the point where the bus is unreliable. All differential signals must be treated as a transmission line with impedance requirements that must be met. As CORGI routes all differential pairs on external layers all traces can be considered microstrip transmission lines and calculating single ended and differential impedances can simply be done by utilizing an online calculator such as the one in.[1] Alternatively, a finite element field solver may be utilized, but are generally prohibitively expensive for university teams.

Radiation Mitigation

Single Event Effect (SEE) testing has been carried out on the NVIDIA TX2 in.[16] It was found that power cycling the device rectified all issues across most runs. Unfortunately, due to the modular nature of the TX2i, carrier board designers have no control over internals such as flash memory or other

possible sources of functional interrupts, so it is challenging to identify and solve root causes of failure from a hardware perspective. Several software mitigation techniques can be implemented to help address this problem, such as watchdog timers, triple modular redundancy (TMR), and error-correcting code (ECC). DRAM ECC is a feature specific to the TX2i model, and includes SBE (Single Bit Error) correction and DBE (Double Bit Error) detection in the Safety Engine firmware (SCE-FW). In the case of the MOCI mission, TMR is implemented by modifying the bootloader to store three copies of the TX2i's operating system images, which are susceptible to radiation damage, and its hashes – when booting the module, each image's hash is calculated to determine if an image is corrupted. Additionally, the satellite's OBC, which is itself radiation-hardened, acts as a watchdog to detect anomalous events occurring on the TX2i and attempts to correct the fault through hard commanding/shutdown of the module.[9]

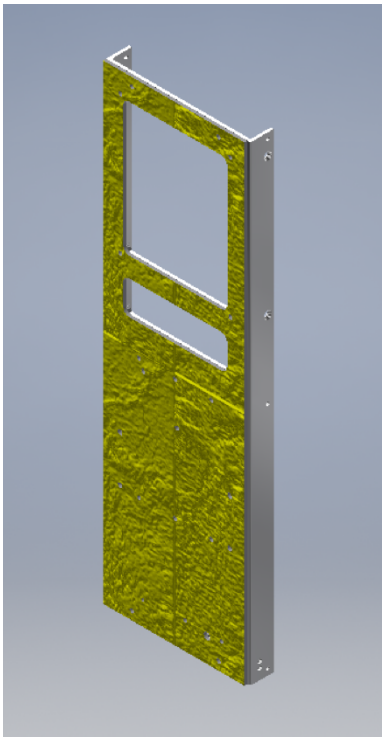

Figure 3: Foil Radiation Shielding

Table 2: Orbital Parameters for MOCI

Orbit	Heliosynchronous
Apogee	500 km
Perigee	500 km
Inclination	97.4 degrees

To the best of the authors' knowledge, Total Ionizing Dose (TID) data has not been published on either the TX2 or the TX2i, however testing has been carried out on various other GPU systems including the Jetson Nano[17] as well as a set of NVIDIA 600 series chips.[15] All of the NVIDIA 600 series chipset tested were able to withstand up to 6krad without total failure, only requiring a reset, and the Jetson Nano was found to be likely to survive past 20krad. While these results can not quantifiably be extended to the TX2i it is not unreasonable to believe the TX2i will survive at least a short mission ($< 1 year$) in LEO with TID levels likely being on the level of approximately 20 krad as computed by SPENVIS models (See SPENVIS output) for MOCI's orbit (see table below). Radiation shielding is provided to the CORGI by utilizing the Dunmore Aerospace Sat Kit. A full summary of the characteristics can be found in,[8] but the important details are that the total shield thickness is 0.1mm and is 99.9% aluminum. Note that this foil lies on the frame of the spacecraft (see figure showing this). Additional protection is offered by the physical casing of the TX2i.

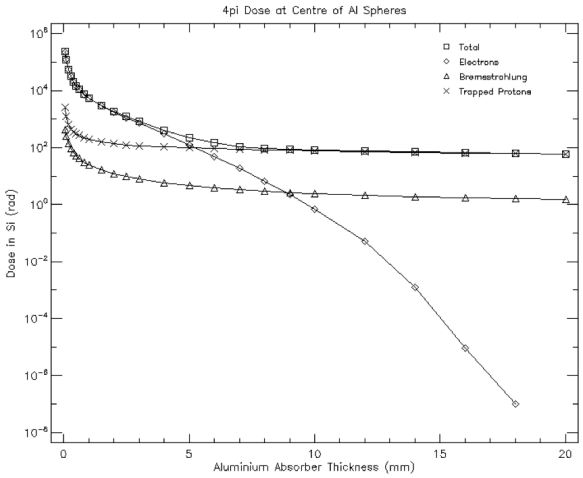

Figure 4: SPENVIS radiation test results

Simulation Results

Simulation results show the circuitry utilized on the board provides appropriate discharge timings. The results of the simulations of the power discharging circuit implemented on the board can be seen in the figure below.

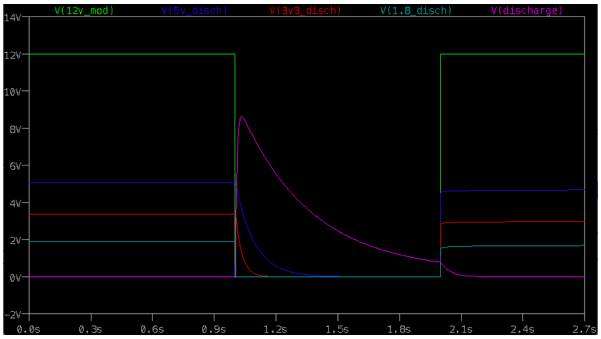

Figure 5: Simulated rail discharge

It can be seen at $t = 1s$ the `12V_MOD` signal is removed, simulating a loss of power case. The discharge signal then asserts, causing the `5V_SYS`, `3V3_SYS`, and `1.8V_SYS` nets to discharge. The simulated results can be seen to match the theoretical time constants of each net, shown in Table 3, to within 10%. R denotes the resistance of the current limiting resistor in series with the $R_{ds_{on}}$ (1Ω) of the NTR4003NT1G NMOS device used to trigger the rail discharge (See schematic in section above). C denotes the total bypass capacitance on the rail. Trace RC characteristics are negligible and are not included.

Table 3: Simulated time constants for each rail

Net	R Ω	C μF	τ ms
1.8V_SYS	37 Ω	3	0.1
3V3_SYS	48	660	31.6
5V_SYS	101	904	91.3

At $t = 2s$ power is reapplied to the system, causing the grounding of the discharge signal. Note that for the purposes of this simulation the input resistance to the capacitive load of all power nets is simulated as 0Ω. This causes instant charging of each power net. In reality the charge time of each net will depend on the current drive capabilities of the spacecraft EPS.

The unstable power case also needs to be handled by the board as detailed in NVIDIA's TX2 OEM design guide. Results of the circuit simulation can be seen in the figure below. At $t = 1s$ the `12v_mod` signal drops to 70% of the original value, or 8.4 volts. It can be seen that this falling edge triggers the `vin_pwr_bad_l` net, indicating to the module and the discharge circuitry that unstable power has been detected.

Discharge circuitry was tested by removing power and observing the decay of the `5V_SYS` and `3V3_SYS` rails on a Rigol 1054Z Oscilloscope. A faulty 1.8V regulator circuit was causing high frequency noise to spill into all power rails when active, so it was not tested in this revision. An image of the scope capture can be seen below.

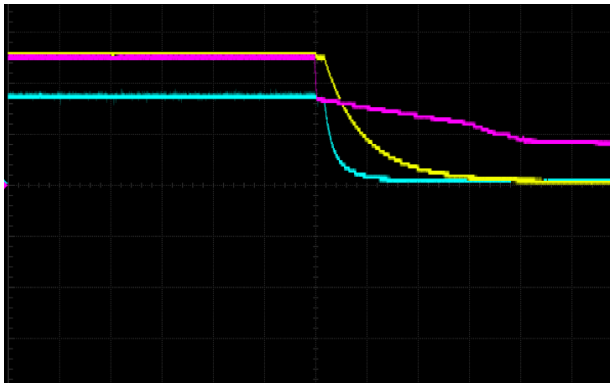

Figure 6: 5V and 3.3V Rail Discharge

Channel 1 reads the `5V_SYS` rail, channel 2 reads the `3V3_SYS` rail, and channel 3 shows the `12V_MOD` rail. The time scale is $100ms/division$. The time constants can be seen to be about 180ms for the `5V_SYS` rail and 80ms for the `3V3_SYS` rail. The controlled discharge of the `12V_MOD` rail can also be seen here, which decays to 37% of its original value (4.44 volts) at approximately $t = 400ms$. The image below shows the behavior of the `VIN_PWR_BAD_L` net when power is applied, then removed.

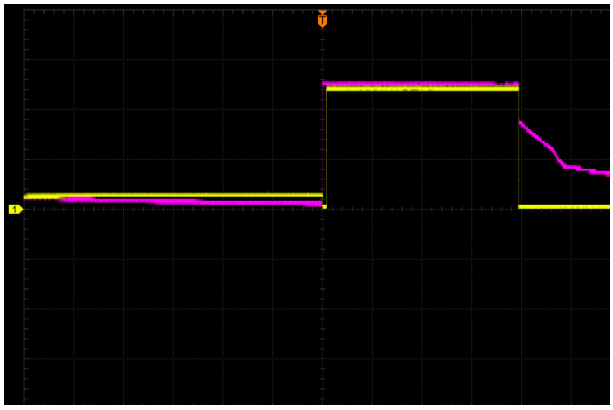

Figure 7: 12V Rail Discharge

Channel 1 monitors the `VIN_PWR_BAD_L` net while channel 3 monitors the `12V_MOD` net. The results show that when power is removed the `VIN_PWR_BAD_L` net goes active, and signals to the TX2i module that faulty power has been detected. Also of interest is the total power draw of the board. While the TX2i is idle it draws on average 5000-6500mW. Under high computational load (see Section GPU Test Results) the system draws approximately 9000mW. Peripheral circuitry at this time has negligible current draw, with microwatt level power draw on the 5V and 3V3 rails.

Communications Test Results

The UART bus was tested with the standard configuration discussed in the "Communications" subsection (115200 baud, 8N1). Three large files from the Canterbury Corpus collection were written over serial to a PC using a serial-to-USB cable five times per file and their total times to be sent entirely were measured.[10] The file transfer was performed without using a particular file transfer protocol in order to measure the raw bit rate as closely as possible, although processing overhead could not be eliminated entirely.The table below shows the files transferred and their sizes, average time taken to transfer, and standard deviation.

Table 4: Style Specifications

File	**Size** *bytes*	**Average time** *sec*	σ *sec*
"alice29.txt" Canterbury Corpus	152089	13.41	0.00452
"plrabn12.txt" Canterbury Corpus	481861	41.33	0.00798
"pi.txt" Miscellaneous Corpus	1000000	1:24.98	0.01892

With a baud rate of 115200, 8 data bits, and 2 bits for the start and end of the frames, the times calculated above are close to their theoretical times; any extra time can be attributed to processing times and potentially the use of the USB converter device connected to the PC. This demonstrates the correct functionality of CORGI's UART interface. However, UART in general may be more ill-suited to larger files (i.e. "pi.txt", which is 1MB large, took nearly 90 seconds across all runs). This could potentially be mitigated by changing the clock source of the UART in software and configuring it accordingly to reliably use a much larger baud rate. Alternatively, using a faster peripheral on the TX2i, such as SPI or USB, is another solution.

GPU Test Results

Several benchmark tests were run on CORGI to characterize the computational performance, power consumption, and thermals of the integrated NVIDIA Pascal GPU while integrated into the carrier board. Three different tests were run: the VGG-19 Jetson benchmark developed by the NVIDIA team themselves as part of the "jetson benchmarks" package, the CUDA bandwidth test included with the CUDA 10.0 kit, and Ville Timonen's `gpu_burn` utility ran in tandem with the Linux "stress" tool.[18] The VGG-19 benchmark was configured with the TX2i placed in maximum throughput mode, recommended clock frequency of 1.122 GHz and workspace size of 1024, and was run a total of three times. The CUDA bandwidth test was configured with pinned memory and a memory transfer size of 33554432 bytes. The `gpu_burn` utility was executed with the following command to run for 3600 seconds: `./gpu_burn -d 3600`. The stress utility was executed as `./stress --cpu 8 --vm 2`. NVIDIA's tegrastats utility was used to log to a file for all test runs in order to retrieve thermal and power draw metrics on the power rails. The results for the VGG-19 model runs are presented in the table below. All metrics are averages across each run. The average frames per second shown per run very closely approximate the frames per second given by NVIDIA (29 fps) when running this particular test with a similar configuration.[2] The integrated GPU on CORGI shows similar performance, indicating that it can sustain heavy computational loads under maximum throughput without throttling.

Table 5: VGG-19 benchmark results A

Run #	FPS	VDD_SYS_GPU (W)	VDD_SYS_SOC (W)	VDD_IN (W)
1	27	3.032	0.912	9.051
2	28	3.032	0.913	9.048
3	25	3.198	0.912	9.044

Table 6: VGG-19 benchmark results B

Run #	FPS	VDD_SYS_CPU (W)	VDD_SYS_DDR (W)
1	27	2.284	1.318
1	28	2.302	1.318
1	25	2.285	1.319

The results for the CUDA bandwidth test are shown below. The test took 2.1 minutes to execute fully. While not necessarily a full-fledged benchmark, the test allowed verification of the compute and bandwidth numbers to ensure that they were in range for the TX2i and did not suffer loss. The acryonym D2H used in Table 7 is communication from the device to the host machine, H2D is host to device, and D2D is device to device.

Table 7: BandwidthTest benchmark results

Run #	H2D BW (MB/s)	D2H BW (MB/s)	D2D (MB/s)
1	15407.1	15527.1	23063.8
2	15501.9	15523.4	23840.1
3	15456.3	15525.3	23100.9

Lastly, the characterization of the TX2i state during the tandem `gpu_burn`/stress test are shown. All CPU cores were at 100% utilization for nearly the entire duration of the test. The peak temperatures of the GPU and Tboard was 68.5 ° Celsius and 65 ° Celsius respectively, which is within the overall operating range of the TX2i module (between -40 and 85 ° Celsius). Peak power consumption on the `VDD_IN` line was 12.28W, although this appears to be an anomaly as the TX2i showed nominal thermals and power draw on other lines otherwise. 9.1W was the second highest instantaneous power draw as logged by tegrastats. This is within a reasonable range given that both the CPU and GPU were at full utilization, albeit at room operating temperature. Additional tests must be performed in a thermal vacuum chamber to further verify the performance of integrated CORGI under actual LEO conditions.

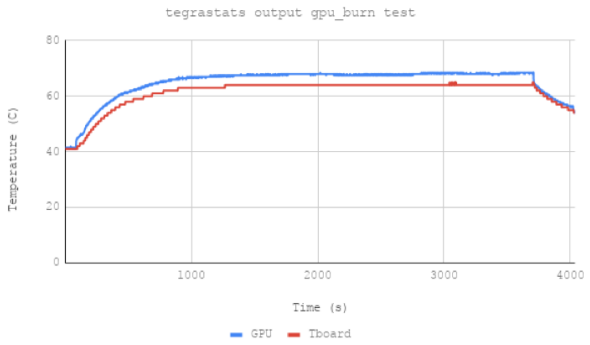

Figure 8: GPU and Tboard temps over gpu burn/stress test

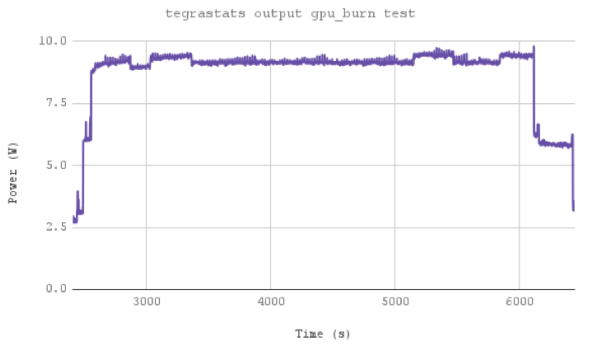

Figure 9: Power consumption during gpu burn/stress test

FlatSat Integration

The PCB was built utilizing a full turnkey process by Advanced Assembly. The board is approximately 90x96x1.6mm unpopulated and 90x96x24mm fully populated. The image below shows the board with the TX2i integrated. A faulty 1.8V regulator on the current revision of the board means the TX2i is not able to interface directly with other satellite components via the PC104+ header. Despite this, the system was able to be powered and Linux booted from Clyde Space 40Whr batteries, and communication was established with a Clyde Space OBC over UART via external logic shifters. This all implies the system is able to be fully integrated into a spacecraft or flatsat upon a final revision. An image of the test setup can be seen in the figure below.

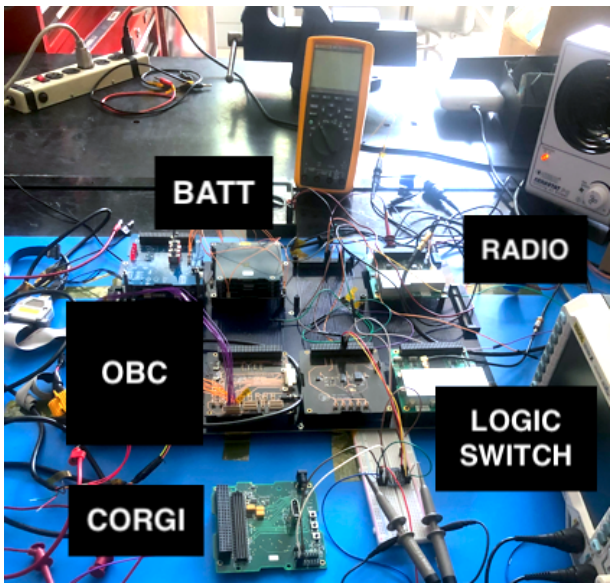

Figure 10: FlatSat Integration

CONCLUSION

Electrical testing of the CORGI exhibited expected onboard voltages and correct power-down sequencing. Additionally, the TX2i module managed to boot into the OS and exhibit nominal GPU performance, and communication between the CORGI, with the TX2i integrated, and an external processor was verified. The CORGI board is now at TRL4, serving as an engineering unit-level component, and will be updated to meet final mission requirements and accommodate payload and flight software. The CORGI design is intended to provide a demonstration of a small usable hardware platform for high performance computing.[7]

Acknowledgments

Thank you to the MOCI Payload Engineers Jackson Parker and Caleb Adams for the many CORGI interrogations, and GPU processing insight. Thank you to Austin Kinkade for his collaboration on the CORGI design. Thank you to Evan Lengle and Cate Davis for their circuit simulation work, and the rest of the Small Satellite Research Laboratory Hardware team for your creative work towards space-bound hardware solutions. Finally, thank you to Chris Heistand, Jeff Boye, Sean Martin, and the Johns Hopkins DART Team for consulting with us on this project.

References

[1] *Edge Coupled Microstrip Impedance Calculator*. All About Circuits. URL: `https://www.allaboutcircuits.com/tools/edge-coupled-microstrip-impedance-calculator`.

[2] *Jetson Benchmarks*. NVIDIA. URL: `https://developer.nvidia.com/embedded/jetson-benchmarks`.

[3] *USB 3.0 Electrical Compliance Methodology, Revision 0.5*. USB Implementers Forum, 1993.

[4] *Universal Serial Bus 3.0 Connectors and Cable Assembly Compliance Document*. USB Implementers Forum, 2011.

[5] *NASA Technology Roadmaps*. National Aeronautics and Space Administration, 2015.

[6] *Appropriate Data Line Common Mode Choke Selection*. Signal Integrity Journal, 2016.

[7] *OEM Product Design Guide*. NVIDIA, 2019. URL: `https://developer.nvidia.com/embedded/downloads`.

[8] C. Adams, A. Spain, J. Parker, M. Hevert, J. Roach, and D Cotten. *Towards an integrated GPU accelerated SoC as a flight computer for small satellites.* 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2019.

[9] Caleb Adams. *High Performance Computation with Small Satellites and Small Satellite Swarms*. Thesis. University of Georgia, 2020.

[10] Tim Bell. *The Canterbury Corpus*. University of Canterbury, New Zealand, 1997. URL: `http://www.data-compression.info/Corpora/CanterburyCorpus/`.

[11] National Research Council. *The Role of Small Satellites in NASA and NOAA Earth Observation Programs*. Washington, DC: The National Academies Press.

[12] Richard Doyle, Raphael Some, Wesley Powell, Gabriel Mounce, Montgomery Goforth, Stephen Horan, and Michael Lowry. *High performance spaceflight computing (hpsc) next-generation space processor (ngsp)*. Proceedings of the Workshop on Spacecraft Flight Software, 2013.

[13] Maximillian Holliday, Andrea Ramirez, Connor Settle, Tane Tatum, Debbie Senesky, and Zachary Manchester. *PyCubed: An Open-Source, Radiation-Tested CubeSat Platform Programmable Entirely in Python*. NVIDIA, 2019.

[14] Howard Johnson. *A handbook of Black Magic*. Pearson, 1993.

[15] George Salazar and Glen Steele. *Commerical Off the Shelf (COTS) Graphics Processing Board (GBP) Radiation Test Evaluation Report*. National Aeronautics and Space Administration, 2013.

[16] Doug Sinclair and Jonathan Dyer. *Radiation Effects and COTS Parts in Small Satellites*. Proceedings of 27th Annual AIAA/USU Conference on Small Satellites, 2013.

[17] W. S. Slater, N. P. Tiwari, T. M. Lovelly, and J. K. Mee. *Total Ionizing Dose Radiation Testing of NVIDIA Jetson Nano GPUs*. 2020 IEEE High Performance Extreme Computing Conference (HPEC), 2020.

[18] Ville Timonen. *Multi-GPU Stress Test*. 2012. URL: `http://wili.cc/blog/gpu-burn.html`.

[19] E.J Wyras. *Proton Testing of an NVIDIA Jetson TX2*. National Aeronautics and Space Administration, 2019.