Energy Budgeting for CubeSats with an Integrated FPGA

Scott Sterling Arnold, Ryan Nuzzaci, and Ann Gordon-Ross
Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611
Also with the NSF Center for High Performance Reconfigurable Computing (CHREC) at the University of Florida
http://www.ann.ece.ufl.edu; scottarnold@ufl.edu, nuzz87@gmail.com, ann@ece.ufl.edu

Abstract*—CubeSats are a simple, low-cost option for developing quickly-deployable satellites, however, the tradeoff for these benefits is a small physical size, which restricts the CubeSat's solar panels' size and thus the available power budget and stored energy reserves. These power/energy limitations restrict the CubeSat's functionality and data processing capabilities, which makes leveraging CubeSats for compute-intensive missions challenging. Additionally, increasing sensor capabilities due to technological advances further compounds this functionality limitation, enabling sensors to gather significantly more data than a satellite's limited downlink bandwidth can accommodate. The influx in sensed data, which is particularly high for image-processing applications, introduces a pressing need for high-performance on-board data processing, which preprocesses and/or compresses the data before transmission. FPGAs have been incorporated into state-of-the-art satellites to provide high-performance on-board data processing, while simultaneously reducing the satellites' data processing energy consumption. However, even though FPGAs can provide these capabilities in full-scale satellites, a CubeSat's limited power budget makes integration of FPGAs into CubeSats a challenging task. For example, the commonly used Virtex4QV Radiation Tolerant FPGA family's average power consumption ranges from 1.25 to 12.5 Watts, whereas the CubeSat's power budget ranges from 2 to 8 Watts, with the smallest, cheapest CubeSat systems at the lower end of this range. Therefore, in order to successfully integrate FPGAs into CubeSats, the components' power consumptions must be clearly budgeted with respect to the CubeSat's specific functionalities and orbital pattern, which dictates the available power and stored energy reserves. In this paper, we present two detailed energy reserve budgeting case studies for FPGA-based CubeSats with respect to stored energy reserves for image compression and processing using a Canny edge detector. CubeSat designers can leverage this energy reserve budget with the application-specific components' power consumptions for applications such as hyper-spectral imaging (HSI), ground motion target indication (GMTI), and star tracking to quickly determine maximum payload operational time with respect to specific orbital patterns and mission requirements.

TABLE OF CONTENTS

1. INTRODUCTION

CubeSats—or cube satellites—are small satellites categorized by a standard size and weight [21] and are divided into three different unit sizes based on physical volume: the 1U (one unit) CubeSat is 10x10x10 cm, the 2U (two unit) CubeSat is 10x10x20 cm, and the 3U (three unit) CubeSat is 10x10x30 cm. This size restricts the CubeSat's upper weight limit at 1 kg per unit-size (e.g., 3 kg for a 3U CubeSat). These size and weight restrictions limit the available payload for batteries and external solar panels, which limits the CubeSat's power consumption and therefore imposes a strict power budget. A CubeSat's *power budget* defines the maximum combined power consumption of all of the CubeSat's constituting payload components based on the total power production of the CubeSat's power producing components (e.g., solar panels). For a CubeSat's mission to be successful, designers must consider and adhere to the power budget during the CubeSat's design and construction process.

To assist designers in CubeSat mission design, CubeSat construction guidelines [20] impose SWAP (size, weight, and power), shape, and other restrictions that, when adhered to, offer several benefits over larger satellites in terms of cost, development time, and payload modularity. These benefits make CubeSats more amenable to academia, small companies, and countries without fully-funded space programs. A CubeSat's small size reduces launch costs [10] and a standardized shape reduces the development cost and time. Since a standardized shape imposes inherent design restrictions, which restrict complexity and reduce design options, CubeSats have short development times (the time

978-1-4577-0557-1/12/$26.00 ©2012 IEEE

from mission inception to CubeSat construction), typically ranging from six to twelve months [21]. Additionally, CubeSats leverage commercial-off-the-shelf (COTS) components, which are low cost and decrease development time by eliminating the need for lengthy custom component design and fabrication. However, the inherent risks of using COTS components offset these low-cost benefits are somewhat offset by the, such as high susceptibility to upset-inducting radiation (e.g., single event upsets (SEUs) [5]) and a shorter lifespan (usually orbit-dependent and on the order of 6 to 18 months).

Since mission design often begins before the payloads' components are determined and CubeSat construction begins, payload modularity enables mission design to commence relatively independent from CubeSat construction, allowing payloads to be easily integrated later in the construction process. However, payload modularity introduces power budgeting challenges since the payload's constituting components may not be determined until late in the mission design process, which may restrict the payload's power and data processing capabilities.

A CubeSat's power budget is composed of several key *subsystems*, these subsystem's constituting components, and the payload power. The CubeSat's subsystems include attitude determination and control (ADCS or ACS), electrical power and supply (EPS), communication, and control and data handling (C&DH). The ADCS can vary from passive magnetic control, which requires little to no power, to high power methods such as gyroscopic control based on horizon sensors and mechanical actuators using star-tracking. The ADCS choice depends on the mission's functionality requirements, and the ADCS influences the payload options based on these requirements (e.g., camera directionality, antennae position, etc.).

The EPS directly dictates the solar panels' generated and stored power, which defines the CubeSat's maximum power budget. Complex, foldout solar panels and higher quality batteries can increase the maximum power budget, However, physical CubeSat size restricts the area available for solar panels. Typically, the 1U, 2U, and 3U CubeSats' maximum power budgets range from 1 to 2.5 Watts, 2 to 5 Watts, and 7 to 20 Watts, respectively [10].

The communication subsystem, one of the largest power consuming subsystems, is critical for establishing and maintaining both uplink and downlink speeds and communication reliability. The communication bandwidth and reliability levels are directly proportional to the communication subsystem's allocated power. However, due to limited power, CubeSats typically have low communication bandwidth and reliability.

These communication limitations make the C&DH, which is responsible for on-board data processing and handling, critical for supporting compute-intensive mission functionality, such as image processing and/or missions with large data sets. Without data processing, the communication subsystem must provide sufficient resources to transmit the complete, unprocessed data. Data processing, using components such as microprocessors, reduces the data set size, which reduces the total amount of data transmitted and reduces communication requirements. Satellites typically employ microprocessors for this data processing, but microprocessors may not provide sufficient processing speed for highly compute-intensive missions, such as image-processing missions that pre-process/compress/coalesce multiple, large data images before transmitting the complete image.

Table 1: CP1 power budget estimation [27].

Subsystem	Power Usage
ADCS (sun-sensor, no control)	0
CMOS camera payload	70 mW
C&DH (ARM7 core)	110 mW
Color camera payload	80 mW
Communication	267 mw
Power Profile	
Total power consumed	527 mW
Solar panel-generated power	726 mW
Power differential	+190 mW

Large satellites often leverage specialized processors, such as field-programmable gate arrays (FPGAs), for compute-intensive applications such as hyperspectral imaging (HSI) [15] and synthetic aperture radar (SAR) [14]. For HSI, a multispectral camera delivers image data cubes ranging in size from hundreds of Megabytes to Gigabytes [15]. Jacobs et al. [15] show that a Virtex-4 SX35 FPGA achieves a 15x speedup as compared to a PowerPC 7455 for an HSI data processing algorithm. For SAR, Jacobs et al. [14] show that a Virtex-4 SX55 FPGA achieves a 6x speedup for azimuthal data processing as compared to a G4 PowerPC.

Even though FPGAs provide high performance data processing, typical FPGA power consumption ranges from 5 to 10 Watts for Virtex and Stratix FPGAs [2], which may exceed a CubeSat's power budget. Table 1 depicts the CP1 CubeSat's—Cal Poly's first developed CubeSat—power budget estimation for each subsystem and the power profile, which includes the total power consumed by the subsystems, the power generated by the solar panels, and the power differential, which is the difference between the power consumed and the power generated. The CP1 had no attitude control, contained two payloads (a sun-sensor and a color picture camera), and had a low-power communication subsystem [27]. The positive power differential shows that while in sunlight, the solar panels captured more power than the CP1 required, thus the excess power could be stored in the batteries for use during eclipse time. Even though demand for high-performance embedded computing has realized lower-power FPGAs that are more amenable to CubeSats, such as the Xilinx Spartan [8], Altera Cyclone [29], and Xilinx Artix families [19] with power consumptions ranging from 1 to 2.5 Watts, these FPGAs

may still require a large percentage of a CubeSat's power budget.

In order to aid designers in leveraging low-power FPGAs in CubeSat missions, we propose an *energy reserve budget*, which equates the total energy produced by the EPS to the total energy consumed by the CubeSat during a single orbit around the Earth. The energy reserve budget leverages multiple power budgets, which are associated with different modes of operation, referred to as *power modes*. All missions should have at least two basic power modes: a *power-storing mode* and an *overpower mode*. In the power-storing mode, the total power consumption is less than the power produced by the solar panels so that the batteries can store reserved energy. In the overpower mode, total power consumption can exceed the solar panels' generated power by leveraging the batteries' stored energy.

Systems with large payload power consumption, such as those including FPGAs, may have several variations of these basic power modes. For example, a system with both high power data processing and high communication requirements may have three power modes: a power-storing mode, a processing-overpower mode, and a communication-overpower mode. More complex CubeSats with multiple payload operations may require a specific overpower mode for each operation. Additionally, due to orbital variations, multiple power modes may be required to compensate for varying power availability.

Since the orbital pattern dictates the amount of sunlight a CubeSat receives and the elevation dictates the amount of time, if any, a CubeSat spends in the Earth's eclipse, these factors primarily determine the amount of available power and the amount of storable energy. Even though the exact orbital pattern and elevation may not be known at design time (CubeSats are often secondary payloads on to-be-determined launch vehicles), the designer can anticipate the orbital pattern to create an approximate energy reserve budget early in the design process, which enables the designer to determine if a launch opportunity is a valid candidate based on the mission's required functionality. Additionally, the energy reserve budget allows designers to more effectively leverage CubeSat resources with respect to predicted power availability, thereby maximizing overpower mode operation and increasing CubeSat efficacy.

In this paper, we propose energy reserve budgets for 1U and 3U case studies based on common orbital patterns and elevations to determine the maximum percentage of orbital time available for high-power FPGA processing (i.e., time spent in a processing-overpower mode). This effective processing time allows designers to most effectively choose the maximum processing resources with respect to the processing resources' power consumption. The energy reserve budget also determines what types of compute-intensive functionalities are possible on CubeSats of all unit sizes. Using Xilinx ISE and ISIM tools, we calculate the power consumption for several low-power FPGAs using a Canny edge detector (Canny filter) to approximate how much power the FPGA requires for a multistage image-processing filter. Based on the energy reserve budgets from these case studies, the components' power consumptions, and the specific orbital pattern and elevation,, results analysis shows that low-power FPGAs can be incorporated into a CubeSat's C&DH, and the energy reserve budget assists designers in effectively leveraging power resources to maximize the payload processing capabilities .

2. Background And Related Work

In this section, we summarize the goals and motivations behind CubeSat missions and discuss several past CubeSat missions. We discuss how CubeSats have become increasingly popular for demonstrating novel technologies in space and how CubeSats have provided low-cost opportunities for academia, small businesses, and governments to perform space-based experiments. Since CubeSats are a relatively new platform—the first proposed CubeSat was in 1999—we outline notable failures and lessons learned from these failures. Lastly, we discuss processing speed up and energy savings potential afforded by FPGAs for image-processing space systems.

2-1. CubeSat Missions

The California Polytechnic (Cal Poly) University Multidisciplinary Space Technology Laboratory and Stanford's Space Systems Development Laboratory initially conceptualized CubeSats as an effective way to provide students with hands-on experience with developing, launching, and operating a spacecraft [10]. The CubeSat standard was developed to provide basic physical features and safety requirements for CubeSat designers. The standard specified size, weight, and shape requirements (Section 1) and imposed additional restrictions such as no pyrotechnics/explosives and no interference with other CubeSats being deployed from the same launch vehicle.

Cal Poly also designed a standardized deployment capsule called the P-POD [25], which could hold any combination of 1U, 2U, or 3U CubeSats totaling up to three units in size (i.e., three 1Us, one 2U and 1U, or one 3U). P-PODs were designed to lower the inherent risk of CubeSats by providing protection from debris during launch, ensuring that early activation of CubeSats did not occur, and ensuring that antennas and booms were not prematurely deployed. Alternative deployment capsules, such as the X-POD and T-POD [18], perform the same function as the P-POD but are less popular due to the X- and T-PODs' lower carrying capacities and because Cal Poly, who organized the majority of early CubeSat launches, required the use of P-PODs. As a result, 90% of all CubeSats are launched from P-PODs [18].

CubeSat construction guidelines, though strict with respect to SWAP and shape, are amenable to a wide range of missions including: sensor deployment to retrieve telemetric data such as the QuakeSat that used a magnetometer to measure the Earth's magnetic field [17]; testing COTS

components in a space-based environment such as the CUTE 1.7 APD II that used a personal data assistant (PDA) as the primary C&DH subsystem [11]; and advancing CubeSat technology through experimentation such as the ION that was equipped with an electrical propulsion subsystem [13].

Due to the CubeSat's small size, light weight, and COTS components, CubeSats are often a risky platform, with only two-fifths of CubeSat launches successfully meet the mission's goals [10]. CubeSat failures can occur due to launch vehicle failures, such as the DNEPR rocket explosion that destroyed fourteen CubeSats [9], and subsystem malfunctioning, such as the AAU CubeSat that experienced a short-circuited radio shortly after launch that drained the entire battery [1]. However, since the majority of CubeSat failures cause the CubeSat to never establish radio contact, the failure reason cannot be diagnosed.

2-2. Space-based FPGA Image Processing

The FPGA's ability to perform computations and operations in circuit-emulating hardware as compared to processors performing computations in software is essential to understanding the speedups that are afforded from executing parallelizable applications on FPGAs. Even though FPGAs typically operate at lower frequencies than processors, FPGAs can attain performance gains that are magnitudes faster.

Since previous work has shown that image-processing applications are amenable to FPGA speedup [14][15], FPGAs are suitable for image-processing space missions, which typically capture large and/or many images and slow/limited downlink bandwidth from satellites to ground stations makes transmitting such large data sets difficult, if not impossible. FPGAs have been employed in image-processing missions to compress/coalesce image data before transmission to reduce downlink requirements. Specialized applications with data-filtering algorithms can benefit from significant speedups that, with regards to overall power consumption, can reduce the total energy consumption by using FPGAs as compared to processors. For example, Kovac et al. [16] used a fully-pipelined VLSI (very large scale integration) architecture to perform JPEG compression at a rate of 100 million pixels per second, which is fast enough to compress thirty 1024x1024 pixel frames in one second. Jacobs et al. [15] showed that an FPGA implementation of SAR afforded a 6x speedup and a 50% reduction in energy consumption as compared to a software implementation.

3. ORBITAL PATTERNS

Since a CubeSat's orbital pattern dictates the amount of sunlight a CubeSat experiences, it is critical to consider how the orbit influences the amount of available power during mission design. CubeSats are typically placed in low Earth orbit (LEO), which ranges from 160 Km to 2000 Km in altitude [23] and offers a variety of mission options for scientific applications such as experimentation and Earth science and imaging [7]. This altitude range defines LEO, rather than the specific orbital pattern, which defines the particular path a satellite follows during orbit. Above LEO is medium Earth orbit (MEO), which extends outward to 35,786 Km, and is followed by high Earth orbit (HEO) for altitudes greater than 35,786 Km.

Orbital patterns are defined by the orbit's inclination and eccentricity. The inclination establishes the angle at which a satellite orbits the Earth with respect to the equator, is dictated by the CubeSat's deployment from the launch vehicle, and dictates the satellite's Earth geographical position at a given point in time. Figure 1 shows the orbit of a satellite at a 75^o inclination, which is why the ground-tracking view shows the satellite taking an apparent north and south motion as it traverses around the globe. Satellite inclination is always quantified between 0^o and 180^o, with a 0^o heading indicating eastward movement in orbit and a 180^o heading indicating westward movement.

The eccentricity establishes the deviation an orbit takes from a perfect circle around the Earth based on a two-dimensional plane. Orbit eccentricity divides orbits into two categories: closed and open. Closed orbits, including circular and elliptical orbits, periodically repeat the same shaped path around the Earth. Closed orbits steadily decay and eventually end when the satellite loses velocity and plummets to Earth or when the satellite is moved into the Junk orbit (an orbit considered not operationally useful due to the orbit's high altitude). Open orbits, including hyperbolic and parabolic orbits, eventually escape the Earth's gravitational pull by achieving enough energy to reach escape velocity after a single orbit. Since satellites in open orbits only orbit the Earth once, our work focuses on closed orbits.

A sun-synchronous orbit is typically a circular, retro-grade orbit at a 98^o inclination and an altitude between 600 and 800 km. Sun-synchronous orbits are popular orbits for CubeSat's for two reasons: 1) sun-synchronous orbits typically experience near-constant sun illumination and thus provide high power; and 2) the CubeSat passes over the same Earth geographical position at the same time every day. Figure 2 depicts a sun-synchronous orbit that never enters the Earth's eclipse by keeping the orbital plane close to orthogonal to the sun's direction.

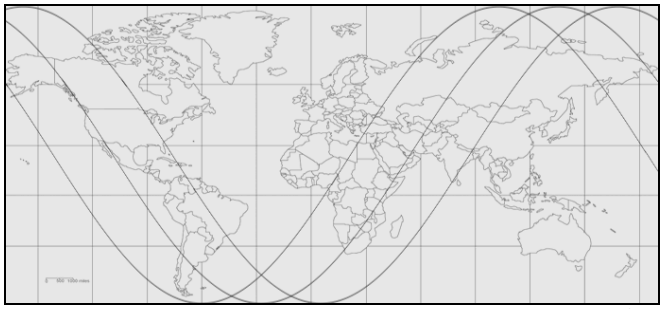

Figure 1: Ground-tracking of a circular orbit at a 75^o inclination

For all circular orbits other than sun-synchronous, the following equation determines the amount of time the satellite spends in the Earth's eclipse [23]:

$$t_s = \left\{ \cos^{-1}\left(\frac{\sqrt{1-R^2}}{\cos\beta} \right) \right\} \frac{\tau}{\pi} \quad (1)$$

where,

R = radius ratio = r_{eq} / r_{sat}

r_{eq} = equatorial radius of the Earth

r_{sat} = geocentric radius of the satellite = $r_{eq} + h_{sat}$

h_{sat} = altitude of the satellite

β = Sun-orbit-plane angle

τ = orbital period of the satellite = $2\pi\sqrt{r_{sat}^3 / \mu}$

μ = gravitational constant of the Earth

The Sun-orbit-plane angle β—or Beta angle—is the angle between the geocentric unit position vector to the sun and the satellite's orbit plane. The worst-case β is 0^o, which indicates that the satellite is spending the maximum amount of time in the Earth's eclipse as possible for a given altitude and the best-case β is $\pm 90^o$, which indicates that the orbit never enters eclipse:

$$\beta = \sin^{-1}\left(\mathbf{r}_{sun} \bullet \mathbf{h}_{sat} \right) \quad (2)$$

where r_{sun} is the geocentric unit position vector of the Sun and h_{sat} is the unit angular momentum vector of the satellite:

$$\mathbf{r}_{sun} = \begin{Bmatrix} \cos\delta_{sun}\cos\alpha_{sun} \\ \cos\delta_{sun}\sin\alpha_{sun} \\ \sin\delta_{sun} \end{Bmatrix} \quad (3) \qquad \mathbf{h}_{sat} = \begin{Bmatrix} \sin\Omega\sin i \\ -\cos\Omega\sin i \\ \cos i \end{Bmatrix} \quad (4)$$

where α_{sun} is the geocentric, equatorial right ascension of the Sun, δ_{sun} is the geocentric, equatorial right declination of the Sun, Ω is the right ascension of the ascending node (satallite) (RAAN), and i is the satellite's inclination [23]. α_{sun} and δ_{sun} are characteristic of the Sun's relative position to the Earth and are a function of the time of year and the inclination and RAAN are characteristics of the satellite [23]. The satellite's orbital period can be determined using a gravitational constant μ = 398,600.4418 km^3/s^2 and the satellite's altitude.

4. Power Consumption Evaluation and Case Studies

In this section, we describe our methodology for evaluating device usage (toggle rate, block RAM (BRAM) writes and reads, etc.) and component power consumption using a sample multistage image filter (Canny filter) for several Xilinx Spartan-3 devices and a Virtex-4 device. The device usage allows power consumption estimators provided by Altera [3] and Xilinx [30] to extrapolate our FPGA power consumption to inspect devices that were not physically simulated. We also present two case studies based on existing 1U and 3U CubeSat designs that will be used to evaluate of our energy reserve budgets in Section 5.

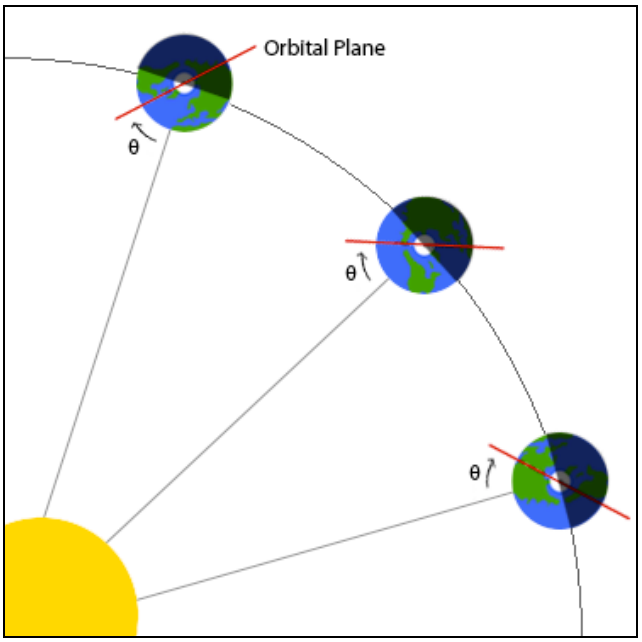

Figure 2: Sun-synchronous orbit showing constant sun illumination

4-1. Experimental Setup

We evaluated the power consumption for five low-power Xilinx Spartan-3 devices (XC3S400A, XC3S400AN, XC3S700A, XC3S700AN, and XC3S400), which represent the smallest Spartan-3 family with enough resources for the Canny filter (for power consumption reasons, we assume a designer would select the smallest device based on the mission resource requirements) and these devices allow us to evaluate device usage trends within the Spartan-3 family. We also evaluated one Virtex-4 device (XC4VFX12) for comparison with a non-low-power FPGA. We created a post-place and route simulation model in ISE [31] and a vcd file in ISIM [31]. The vcd file recorded the image filtering process at the bit-flip level and served as input into Xpower (Xilinx ISE Webpack [31]) to calculate the dynamic and static power for each of the devices, as well as the power consumed by the FPGA fabric resources (logic, BRAM, etc.). From these power consumption results, we obtained the device usage details and power consumption of the FPGA while executing a Canny filter, which comprises several image filter algorithms in series to process an image before edge detection takes place.

Our experiments evaluate an in-house VHDL Canny filter, which is a standard image-processing filter for locating and highlighting edges in an image. Image-processing filters traverse images pixel-by-pixel and perform several

(a) **(b)**

Figure 3: a) The original Wall Street bull image used for Canny filter processing. b) The output image after Canny filter processing.

operations on the pixels to create a modified version of the image. Filtering applications are highly parallelizable, which makes these applications highly amenable to FPGA processing speedups as compared to microprocessors. Figure 3 (b) depicts an example of a Canny filter's output after processing an image of the Wall Street bull in Figure 3 (a). We refer the reader to [12] for additional Canny filter details.

A Canny filter's front-end processing is a Gaussian filter and a Sobel filter in sequence, which obtains the first derivatives of the image to remove high-frequency components and determines the image's intensity gradient. Based on results obtained from Xpower averaged over all Spartan-3 devices, the Canny filter's static power consumption ranges between 40 and 55 mW, the dynamic power is approximately twice that of the static power, and the average resource utilization is approximately 2,500 logic slices and 18 BRAM slices.

The 1024x768 JPEG of the Wall Street bull was serialized using a python script before being input to the FPGA. We note that this serialization could have been done on the FPGA but is not necessary for our case studies since we are effectively simulating an incoming data stream from a camera. The Canny filter processed the serialized image data and stored the output into a FIFO (first-in-first-out) buffer, which was then compressed by the FPGA and transformed to a viewable JPEG image.

4-2. Power Results

Table 2 depicts the dynamic, static, and total power for our evaluated devices while performing the Canny filter operation and Table 3 depicts the associated resource requirements (look-up tables (LUTs) shift registers (shift regs), flip-flops (FFs), clock fanout, input/output pins (I/O), BRAMs, distributed clock manager (DCM), digital signal processors (DSPs), and multiplication units (MULT)). Since the results revealed that the resource requirements were device-independent for the Spartan-3 devices, Table 3 lists only a single column for the Spartan-3 devices. The results revealed only a 4.6% increase in the logic resource requirements (LUTs, shift registers, FFs) when comparing the Spartan-3 devices to the Virtex-4 device.

4-3. Case Studies

Due to the 1U CubeSat's highly constrained size and power budget, the energy reserve budget is most beneficial and

Table 2: FPGA power consumption for the Canny filter

	Power Consumption (mW)		
Device	**Dynamic**	**Static**	**Total**
Spartan-3			
XC3S400A	97.52	40.11	137.63
XC3S700A	103.34	46.40	149.74
XC3S400AN	98.66	48.62	147.28
XC3S700AN	110.35	55.68	166.03
XC3S400	104.19	95.29	199.48
Virtex-4			
XC4VFX12	161.00	199.10	360.10

Table 3: Per-family FPGA resource requirements

Resources	Spartan-3	Virtex-4
Logic (LUTs)	1528	1634
Logic (Shift regs)	46	44
Logic (FFs)	1055	1071
Clock fanout	1119	1156
I/O (input pins)	11	11
I/O (output pins)	12	12
BRAM	18	18
DCM	Single	Single
DSP	0	0
MULT	0	0

Table 4: Power budget for the M-Cubed CubeSat

Subsystem	Description	Minimum Power (mW)	Maximum Power (mW)
ADCS	Passive Magnetic ACS	0	0
C&DH	Atmel AVR32 7002	-216	-292.5
Uplink	144 MHz, .33-m dipole	-616.5	-616.5
Downlink	437 MHz, .5-m monopole	0	-1000
Primary Payload	CMOS camera	0	-250
Secondary Payload	COVE board	0	-5088.5
Board	I/O power and loss	-227.5	-1254
EPS	4 Polymer Li-ion Batteries	-960	+4091
EPS	6 Solar Panels	+2010	+2010

challenging for these CubeSats. Furthermore, since 1U CubeSats cost less than larger CubeSats, previous work shows that 1U CubeSats are the most popular to construct. Therefore, our first case study focuses on the 1U CubeSat, which is based on the M-Cubed CubeSat [4].

Table 4 shows the minimum and maximum power consumptions in mW for the M-Cubed's subsystems [4], where positive (+) numbers indicate that a subsystem is producing power and negative (-) numbers indicate that a subsystem is consuming power. The M-Cubed's secondary payload contained an on-board verification experiment (COVE) and required a high-power Virtex-5QV Single-event Immune Reconfigurable FPGA [24]. Since this payload was the highest power subsystem, the M-Cubed designers carefully considered the overall system's power consumption during mission design. In order to provide enough power for the secondary payload, the system included four Polymer Li-ion batteries, which each stored 3.89 W·h. These batteries were chosen by the M-Cubed designers to provide a larger discharge rate as compared to other higher capacity batteries such that there was enough power to run the secondary payload [4].

The 1U case study contains a single payload and has four power modes: 1) a power-storing mode; 2) a communication-overpower mode for data transmission to a ground station; 3) a communication-uplink-overpower mode for receiving ground station transmissions; and 4) a processing-overpower mode for high-performance FPGA payload processing. In Section 5, we demonstrate how the energy reserve budget, which consists of these power modes, can be used to calculate the orbital-pattern-dependent amount of time a CubeSat can spend in each power mode.

Since energy reserve budgeting is useful for any platform size, we also evaluate an energy reserve budget case study for a larger 3U CubeSat design based on the QuakeSat, which was the first 3U launched into space in 2003[17]. Even though the QuakeSat was short-lived due to a quickly degenerating orbit, the QuakeSat's design could have easily sustained a much longer mission and is therefore a good evaluation choice for a sample 3U CubeSat.

Table 5 shows the minimum and maximum power consumptions in mW for the QuakeSat's subsystems, where positive (+) numbers indicate that a subsystem is producing power and negative (-) numbers indicate that a subsystem is consuming power. Since the QuakeSat's mission dictated that the payload would be active for a significant portion of the orbital time to collect data, the QuakeSat was built using a very conservative power budget [6]. Due to the high power production provided by the twelve solar panels (14 Watts), the QuakeSat easily supported the constant power draw of 3.6 Watts for the CPU (C&DH), uplink, and board I/O for continual operation [17]. Prior to launch, the designers estimated that the maximum power consumption of 12.6 Watts (when all components were operating) would not exceed the average power production of the solar panels when in sunlight.

Table 5: Power budget for the QuakeSat CubeSat

Subsystem	Description	Minimum Power (mW)	Maximum Power (mW)
ADCS	Passive Magnetic ACS	0	0
C&DH	Prometheus CPU	-2500	-2500
Uplink	Tek-net 9600 Baud-Rate Solid-State transmitter	-750	-750
Downlink		-1000	-1400
Primary Payload	Magnetometer	-600	-2200
Board	I/O power and loss	-850	-850
EPS	2 Li-ion Batteries	-8800	+11700
EPS	12 Solar Panels	+14000	+14000

The 3U case study focuses on CubeSat resource optimization, the addition of fault-tolerant networking methodologies, and efficient usage of the available power. Whereas the 1U case study focuses on working within tight power constraints the 3U case study has more power resources and can thus leverage multiple FPGAs in a COTS distributed C&DH network. Due to the larger number of FPGAs and thus increased cumulative payload processing time as compared to the 1U case study, we incorporate fault-tolerant methodologies to ensure high system availability since the COTS FPGAs are not radiation hardened. Using multiple FPGAs allows implementation of a dependable multiprocessor (DM) fault-tolerance methodology, which ensures that FPGA failure does not cause entire system failure. This fault-tolerant operation leverages a *central* processor, which monitors the payload FPGAs to detect FPGA failures [26]. Detailed DM operation is beyond the scope of this paper and we refer the interested reader to [26] for additional information.

The 3U case study contains three payloads and the energy reserve budget requires two iterations to exemplify how the energy reserve budget can indicate an overly conservative CubeSat design with respect to power consumption. The first iteration pinpoints subsystems where increasing the operational time would be advantageous to meeting the mission's goals. The first iteration leverages five power modes: 1) a power-storing mode; 2) a communication-overpower mode for data transmission to a ground station; 3) a processing-overpower-P1 mode for using a single FPGA for payload processing; 4) a processing-overpower-P2 mode for using two FPGAs for payload processing; and 5) a processing-overpower-Pall mode for using all on-board FPGAs for payload processing. The second iteration refines the first iteration's output to further optimize the power usage.

5. ENERGY RESERVE BUDGET

In this section, we use the power results from our Canny filter (Section 4-2) and the 1U and 3U CubeSat designs (Section 4-3) to create three energy reserve budgets: two budgets for the 1U CubeSat (single-payload in Section 5-2) and one budget for a 3U CubeSat (triple-payload in Section 5-3). The energy reserve budgets' power modes are given as total power consumed. Using the eclipse time equations (Section 3), we calculate the total energy produced during a single orbit, which, when combined with the energy reserve budget, enables designers to calculate the percentage of time the system can spend in each power mode.

5-1. Energy Reserve Budget Development

Developing the energy reserve budget is an iterative process, which becomes more refined and concrete throughout the mission design process. Since early design stages have little information about the exact launch and deployment details, designers must estimate these details in order to create an initial energy reserve budget and construct power modes based on anticipated orbital patterns. Even though inexact in the early design stages, the energy reserve budget is beneficial at the beginning of the design process by enabling designers to better quantify the maximum payload power available as a function of both orbital time and the usable energy produced during a single orbit. By analyzing the orbital pattern and the energy reserve budget, the designer can calculate the payload(s) operational time(s) and compare these times to the mission requirements. If the mission requirements are not met, the designer may redesign the subsystems to achieve lower power consumption or create another iteration of the energy reserve budget to improve the payload(s) operational time(s). One potential side effect of using an energy reserve budget is that early design stages may require additional time due to the additional analysis time required to create the energy reserve budget. However, this additional time is recovered by quickening component selection and CubeSat construction.

We first establish the process of determining an energy reserve budget for a system with two power modes (processing-overpower and power-storing), and then generalize the process for systems with an arbitrary number of power modes. The first step in developing the energy reserve budget is to quantify the payload's power consumption for the overpower mode $P_{payload}$ (the payload is fully operational) and the power-storing mode P_{store} (the payload is completely powered off). These modes' power consumptions are combined with an assumed orbit to determine the CubeSat's eclipse time t_s. During the eclipse time, the solar panels do not generate any power, which decreases the overall energy stored in a single orbit. The following equation calculates the total energy $J_{produced}$ available during a single orbit:

$$J_{produced} = P_{panels} * (\tau - t_s) \tag{5}$$

where P_{panels} is the average power produced by the solar panels (this data can be obtained from the solar panels' datasheets or via physical testing) while in sunlight and τ is the orbital period. Using $J_{produced}$, the following equation calculates the operational time t_p available for the processing-overpower mode during a single orbit.

$$t_p = \frac{J_{produced} + \tau * P_{store}}{P_{store} - P_{payload}} \tag{6}$$

where P_{store} and $P_{payload}$ are the power consumptions during the power-storing and processing-overpower modes, respectively.

In order to generalize these equations to consider a system with an arbitrary number of power modes, assumptions must be made about the required *operational times* (required time spent in each power mode based on the mission's requirements) of all of the power modes except for one power mode. For example, if a communication-overpower mode is added to the energy reserve budget for the previous two-power-mode example, the designer must estimate the

new communication-overpower mode's required operational time and power consumption in order to calculate the new processing-overpower mode's operational time t_p. Similarly, for a system with n power modes, the designer must estimate each power mode's operational time and power consumption in order to calculate the new processing-overpower mode's operational time t_p. The following equation, which is a generalization of Equation (6), calculates a power mode's operational time t_p, for a system with n power modes:

$$t_p = \frac{J_{produced} + P_{stored}(\tau + t_1 + t_2 + \cdots + t_n) + P_1 t_1 + P_2 t_2 + \cdots + P_n P_n}{P_{store} - P_{payload}} \quad (7)$$

where P_n and t_n are the power consumption and operational time during a single orbit for power mode n, respectively. In this example, the communications-overpower mode's required downlink operational time is based on the mission's requirement and must be estimated by the designer before t_p can be calculated (7).

If the mission's requirements dictate that a particular power mode requires the maximum operational time available—a *maximized mode*—Equation (7) can be further generalized to calculate the maximum operational time τ_A by holding all other power modes' operational times constant, except one power mode—the *donor mode*. For example, if a mission requires an operational time of at least three minutes per orbital period for the processor-overpower mode and the communication-overpower mode is the maximized mode, the processor-overpower mode's operational time is set to a constant of three minutes, the power-storing mode is the donor-mode, and τ_A is calculated as:

$$t_A = \frac{J_{produced} + P_B(\tau + t_1 + t_2 + \cdots + t_n) + P_1 t_1 + P_2 t_2 + \cdots + P_n P_n}{P_B - P_A} \quad (8)$$

where P_A and P_B are the power consumptions of the maximized and donor modes, respectively.

5-2. Single-Payload 1U Case Study

For the single-payload 1U case study, we use the M-Cubed subsystems (except for the payload, which we replace with a lower power camera and used the FPGA power estimation from our power results) and the subsystems' estimated power usage from Table 4. In our energy reserve budgets, the single payload's subsystems are a camera and an FPGA. The camera is the same OmniVision 2655 CMOS image sensor used on the M-Cubed, which consumes 250 mW during operation. The FPGA is the Spartan-3 XC3A400T, the lowest power consumption device from our power results. Using these subsystems, we assess the energy reserve budget for the Canny filter (Section 4-1).

We evaluate the energy reserve budgets for ten orbital patterns. Table 6 shows the orbital patterns' altitude, RAAN and inclination, which determine the Beta range, τ (orbital period), and t_s (eclipse times) for the given Beta ranges. The t_s (low), t_s (high), and t_s (avg) columns specify the lowest, highest, and average eclipse times, respectively, for a satellite over a single year. The values in the table are based on an in-house orbital evaluation tool, which uses the equations in Section 3.

The first energy reserve budget iteration requires several assumptions about the mission's required uplink and downlink operational times. The communication subsystem's transmission and reception frequencies determine the maximum distance at which the communication signal can be reliably received, with lower frequencies traveling further. Since our communication subsystem is based on the M-Cubed, we assume that the communication subsystem requires the same communication time per orbit, which is five minutes of downlink time and ten minutes of uplink time. Even though not every orbit will require communication with the ground station, many orbital patterns allow for multiple communication opportunities per day. Therefore, it is necessary to evaluate our energy reserve budget using the maximum available uplink and downlink operational times to calculate the available processing-overpower operational time.

The power modes' functionalities are defined based on the mission requirements, and for this case study, we assume the following mission requirements. In the power-storing

Table 6: Orbital patterns

Orbit	Altitude	Inclination	RAAN	β range	τ	t_s (low)	t_s (high)	t_s (avg)
Equatorial	300 km	0	0	-23°:23°	90.5	35.8	36.6	36.2
Inclination 22.5	400 km	22.5	30	-43°:45°	92.6	31.6	36.1	35.2
Inclination 45	500 km	45	60	-61°:65°	94.6	14.5	35.7	32.7
Inclination 67.5	600 km	67.5	90	-88°:85°	96.7	0	35.5	26.1
Inclination 90 (polar)	300 km	90	60	36°:60°	90.5	27.0	34.4	32.3
Inclination 112.5	400 km	112.5	30	-88°:88°	92.6	0	36.1	29.7
Inclination 135	500 km	135	0	-67°:63°	94.6	8.1	35.8	32.6
Inclination 157.5	600 km	157.5	0	-46°:46°	96.7	29.2	35.5	34.3
Inclination 180	800 km	180	0	-23°:23°	100.9	33.6	35.1	34.4
Sun-synchronous polar	300 km	90	0	71°:79°	100.9	0	0	0
Sun-synchronous nonpolar	400 km	98	0	-75°:82°	92.6	0	0	0

Epoch time of all Orbits is considered to be January 1st, 2011

Table 7: Energy reserve budget for the single-payload 1U case study

	Power Mode's Power Consumptions (mW)			
Subsystem	**Power-Storing**	**Communication-overpower**	**Uplink-overpower**	**Processing-overpower**
ADCS	0	0	0	0
C&DH	216	292.5	216	292.5
Uplink	0	616.5	616.5	0
Downlink	0	1000	0	0
Payload	0	0	0	387.5
Board	960	1254	960	1532
Total	**1176**	**3163**	**1792.5**	**2212**

mode, only the C&DH subsystem is active to enable switching between power modes. This power mode switching process is designer-defined and can be triggered by either external events (e.g., transitioning to eclipse) or an internal schedule. In the communications-overpower mode, the uplink and downlink subsystems are active to allow for the required uplink and downlink operational times and in the uplink-overpower mode, only the uplink subsystem is active. In the processing-overpower mode, the communication subsystems are turned off and all power is used by the CD&H.

Table 7 depicts the energy reserve budget with one power-storing mode and three overpower modes (two for communications (communication- and uplink-overpower) and one for payload operation (processing-overpower)), the power modes' power consumptions per subsystem, and the power modes' total power consumptions. Based on the EPS subsystem's solar panel power production given in Table 5, the power-storing mode will store energy over a single orbit, which is required to evaluate the operational times of all the other higher power consumption modes in the energy reserve budget. Since the uplink subsystem consumes relatively little power, power is also stored in the batteries during the communication-uplink-overpower mode.

We evaluate our energy reserve budget using the uplink and downlink operational time requirements of ten and five minutes, respectively. Five minutes are spent in the communications-overpower mode, which accounts for the entire five minutes of the required downlink operational time and five minutes of the required uplink operational time, leaving five minutes of operational time for the uplink-overpower mode to account for the remaining required uplink operational time. Table 8 shows the processing-overpower mode operational times t_p given the worst-, best-, and average-case eclipse times (low, high, and avg, respectively) based on the required communications-overpower and uplink-overpower modes' operational times. These results reveal that during eight of the ten orbital patterns, the processing-overpower mode operation time t_p is negative for some orbits, which means that these orbital patterns are not suitable for this CubeSat's mission requirements. Furthermore, the negative operational times indicate an excess power consumption due to the other power modes, which reveals that either the uplink-overpower or communication-overpower mode is using too much energy to be sustained over that orbital pattern.

Table 8: 1U payload operational time in minutes during a single orbit

Orbit	**t_p (low)**	**t_p (high)**	**t_p (avg)**
Equatorial	-31.87	-33.42	-32.65
Inclination 22.5	-22.03	-30.76	-29.02
Inclination 45	12.75	-28.38	-22.56
Inclination 67.5	42.58	-26.30	-8.06
Inclination 90 (polar)	-14.80	-29.15	-25.08
Inclination 112.5	39.28	-30.76	-18.35
Inclination 135	25.17	-28.57	-22.36
Inclination 157.5	-14.08	-26.30	-23.97
Inclination 180	-19.23	-22.14	-20.78
Sun-synchronous polar	45.96	45.96	45.96
Sun-synchronous nonpolar	39.28	39.28	39.28

Given this energy reserve budget analysis, designers may either accept this energy reserve budget, which limits their mission to two orbital patterns or the designer may modify the mission's requirements or subsystems to allow this CubeSat to be amenable to more orbital patterns, and therefore more launch opportunities. Modifications to the mission's requirements include decreasing the power modes' required operational times or power consumptions or by completely replacing/redesigning subsystems. Modifications to the Cubesat can be quickly evaluated using the existing energy reserve budget and the payload operational time can be recalculated. Additionally, the designer may replace, redesign, or add new subsystems, however, these modifications are more drastic since even modification to a single component may affect the CubeSat's capabilities and adversely affect the overall system. For example, using a lower power downlink subsystem may reduce the downlink communication resources, which in turn may require more efficient on-board data compression/preprocessing. After changes are applied and new t_p values can be calculated and evaluated and further changes to the CubeSat may be assessed.

5-3. Triple-Payload 3U Case Study

For the triple-payload 3U case study, we use the QuakeSat subsystem design with the Magnetometer payload replaced by the same camera used in the 1U case study (the

Table 9: Energy reserve budget for the triple-payload 3U case study – first iteration

	Power Mode's Power Consumptions (mW)				
Subsystem	**Power-Storing**	**Communication-overpower**	**Processing-overpower-P1**	**Processing-overpower-P2**	**Processing-overpower-Pall**
ADCS	0	0	0	0	0
C&DH	2500	2500	2500	2500	2500
Uplink	750	750	750	750	750
Downlink	0	1400	0	0	0
P (Cam)	0	0	250	250	250
P (FPGAs)	0	0	137.6	275.2	1651.2
Board	350	350	350	350	350
Total	**3600**	**5000**	**3987.6**	**4125**	**5501.2**

OmniVision 2655 CMOS image sensor) and combine twelve low-power Spartan-3 XC3A400Ts for a distributed data processing network. This distributed data processing network combines the processing power of the COTS FPGAs with the dependability and fault-tolerance of a DM network [26] using the Prometheus CPU as the central processor. Similarly to the 1U case study, all of the FPGAs execute the Canny filter (Section 4-1) and have the same power consumption (Section 4-2). This case study's goal is to maximize the payload operational time for payload-overpower-Pall, since this mode provides the maximum processing capability using all of the FPGAs.

Table 9 depicts the first iteration of the energy reserve budget for the 3U case study. Unlike the 1U case study in which the power resources were more constrained, the uplink is continually powered on for all power modes, but similarly to the 1U case study, all downlink communication occurs in the communications-overpower mode.

We set the communication-overpower mode operational time to ten minutes for downlink, which enables a data transfer rate similarly to that of the original QuakeSat (3MB/day). Since it is preferable to operate in the highest processing-capable power mode where all twelve FPGAs are active (i.e., the processing-overpower-Pall mode), the first energy budget iteration allocates no operational time to the processing-overpower-P1 or the processing-overpower-P2 modes. If the processing-overpower-Pall mode's operational time given this energy reserve budget indicates that the mission's requirements cannot be met, then the designer can consider using a lower processing-capable and thus lower power, power mode (i.e., processing-overpower-P1 or processing-overpower-P2).

Table 10: Processor-overpower-Pall mode's operational time in minutes during a single orbit for the triple-payload 3U case study – first iteration

Orbit	**t_p (low)**	**t_p (high)**	**t_p (avg)**
Equatorial	186.20	180.31	183.25
Inclination 22.5	228.61	195.48	202.10
Inclination 45	365.47	209.36	231.45
Inclination 67.5	483.74	222.32	291.54
Inclination 90 (polar)	251.00	196.51	211.97
Inclination 112.5	461.31	195.48	242.60
Inclination 135	412.60	208.63	232.19
Inclination 157.5	268.71	222.32	231.16
Inclination 180	259.29	248.24	253.40
Sun-synchronous polar	506.71	506.71	506.71
Sun-synchronous nonpolar	461.31	461.31	461.31

Table 10 shows the processor-overpower-Pall mode's operational times for each orbit given the assumed communication-overpower mode's operational time and the energy reserve budget (Table 9). The operational times reveal that this energy reserve budget enables the CubeSat to operate in the highest processing-capable power mode (i.e., processor-overpower-Pall) for ~180 minutes in the worst-case orbit (the equatorial orbit), which is 2x greater than the total orbital period for this orbit. This excess operational time indicates that this energy reserve budget is too conservative and additional operational time can be allocated to other power modes and still enable the processor-overpower-Pall mode to operate for 100% of the orbital period.

Similarly to the 1U case study, the designer may either accept the energy reserve budget as is, allowing the CubeSat to operate successfully in any orbit, or the designer can create another iteration of the energy reserve budget to optimize the CubeSat's capabilities such that the CubeSat will leverage the available power resources more efficiently. Since the QuakeSat designers reported that the communications downlink was less efficient than desired [6], the second energy reserve budget iteration focuses on increasing the downlink operational time while not compromising the amount of data produced by the payload.

In order to leverage the available power resources more efficiently and increase the downlink time, the second iteration of the energy reserve budget could either increase the other subsystems' operational times or modify the communication-overpower mode operational time.

Table 11: Energy reserve budget for a triple-payload 3U case study – second iteration

	Power Mode's Power Consumptions (mW)			
Subsystem	**Power-Storing**	**Processing-overpower-P1**	**Processing-overpower-P2**	**Processing-overpower-Pall**
ADCS	0	0	0	0
C&DH	2500	2500	2500	2500
Uplink	750	750	750	750
Downlink	0	1400	1400	1400
P (Cam)	0	250	250	250
P (FPGAs)	0	137.6	275.2	1651.2
Board	350	350	350	350
Total	**3600**	**5387.6**	**5525**	**6901.2**

However, since the maximum operational time available to the payload-overpower-Pall mode per orbit is the orbital period minus the operational time used by all other modes, increasing the communication-overpower mode's operational time is counterproductive to the goal of maximizing the payload-overpower-Pall mode's operational time. Alternatively, reducing the communication-overpower mode's operational time would increase the payload-overpower-Pall mode's operational time, but would reduce the communication-overpower mode's operational time and thus would reduce the amount of transmitted data. Given these tradeoffs, the designer's best option to improve the power resource utilization and to increase the downlink time is to modify the power modes' subsystem power usage.

Table 11 depicts the energy reserve budget's second iteration with the goal of maximizing the CubeSat's performance by leveraging the downlink subsystem during the payload-overpower-Pall mode and increasing available operational time for the payload-overpower-Pall mode. This second iteration includes two modifications: 1) the communications-overpower mode's operational time is reduced to zero and is removed from the energy reserve budget, which increases the available operational time for the payload-overpower modes by ten minutes; and 2) since there is sufficient available power, the downlink is active during all modes to meet data transfer requirements, except in the power-storing mode, where the downlink is powered off to ensure minimal system usage and maximum energy storage. These changes increase the available downlink time, increase payload operational time, and enable complete control from a ground station during all overpower modes since both the uplink and downlink are always active (except in the power-storing mode). Since an analysis indicates that there is no need to reduce the payload's processing capabilities (i.e., use the payload-overpower-P1/P2 modes) the energy reserve budget only requires two power modes: the payload-overpower-Pall mode and the power-storing mode.

Table 12: Processor-overpower-Pall mode's operational time in minutes during a single orbit for the triple-payload 3U case study – second iteration

Orbit	**t_p (low)**	**t_p (high)**	**t_p (avg)**
Equatorial	133.28	129.89	131.59
Inclination 22.5	157.71	138.63	142.45
Inclination 45	236.53	146.63	159.35
Inclination 67.5	304.64	154.09	193.95
Inclination 90 (polar)	170.60	139.22	148.13
Inclination 112.5	291.72	138.63	165.77
Inclination 135	263.67	146.20	159.77
Inclination 157.5	180.81	154.09	159.18
Inclination 180	175.38	169.02	171.99
Sun-synchronous polar	317.87	317.87	317.87
Sun-synchronous nonpolar	291.72	291.72	291.72

Table 12 depicts the payload-overpower-Pall mode's operational time assuming the energy budget in Table 11. Similarly to the first iteration, the results indicate that even though the CubeSat has all subsystems operational in the processor-overpower-Pall mode, this mode's operational time for all orbits is still greater than the corresponding orbit's orbital period. Depending on the mission requirements, a designer may consider this design complete, in which case the satellite could be launched into any circular LEO orbit and operate in payload-overpower-Pall mode 100% of the time. Alternatively, since the operational times still indicate excess operational time (i.e., the payload-overpower-Pall mode's operational time is still greater than the orbital period), the designer could perform additional energy reserve budget iterations to further improve the subsystems or the energy reserve budget to increase the CubeSat's processing capabilities. For example, a more accurate and increased controlling-capable ADCS could be added if more accurate directional control were desired, more powerful FPGAs could be used to increase the payload processing capabilities, the Tek-net could be replaced with a more robust (higher power consumption) radio to improve communication reliability, or additional payloads or FPGAs could be added to maximize the processing capability given the size and weight restrictions.

6. CONCLUSIONS

Given the low-cost and fast design time afforded by CubeSats, CubeSats are a popular platform for academia, small companies, and countries without fully-funded space programs. However, given the CubeSat's tight design constraints, leveraging high-performance payload processing components such as FPGAs is challenging. In this paper, we proposed an energy reserve budget and associated power modes to aid designers in most effectively selecting CubeSat components and appropriate orbital patterns based on a mission's requirements. The energy reserve budget enables quick calculation of the maximum available operational time for a CubeSat's payloads based on the solar panels' power production and the batteries' energy reserves. Using this information, designers can quickly select appropriate launch opportunities or redesign their system to meet a mission's requirements based on a launch opportunity's orbital pattern.

We evaluate our energy reserve budget using a one unit (1U) and a three unit (3U) CubeSat for an image-processing mission using a Canny filter. Based on our selected FPGA device and ten sample orbital patterns, the energy reserve budget revealed that only two of these orbital patterns could meet the mission requirements. Using these results, a designer could either refine the mission's requirements or change subsystem components to make their design more amenable to more orbital patterns. Our future work includes evaluating the Cyclone and Artix families, which show promising results based on power estimations.

7. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation (ECCS-0901706 and the I/UCRC Program EEC-0642422). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. The authors gratefully acknowledge vendor equipment and tools provided by Xilinx that helped make this work possible.

8. REFERENCES

[1] L. Alminde, M. Bisgaad, D. Vinter, T. Viscor, and K. Z. Østergard, "The AAU-CubeSat student satellite project: architectural overview and lessons learned," in *Proceedings of the 16th IFAC Symposium on Automatic Control in Aerospace*, 2004

[2] Altera inc., "Stratix III FPGAs vs. Xilinx virtex-5 Devices Architecture and Perfomance Comparison", White Paper, October 2007

[3] Altera inc, *Altera Powerplay Early power Estimators, copywrite* 2005-2011

[4] D.L. Bekker, T.A. Werne, T.O. Wilson, P.J. Pingree, K. Dontchev, M. Heywood, et al, "A CubeSat design to validate the Virtex-5 FPGA for spaceborne image processing," *Aerospace Conference, 2010 IEEE*, Big Sky, MT, 6-13 March 2010

[5] P. Bernardi, M. Sonza Reorda, L. Sterpone, M. Violante,"On the evaluation of SEU sensitiveness in SRAM-based FPGAs," in *IEEE IOLTS Proceedings*, 10., 2004, Los Alamitos : IEEE Computer Society, p.115-120., 2004

[6] T. Bleier, P. Clarke, J Cutler, L. DeMartini, C. Dunson, et al, QuakeSat Lessons Learned: Notes from the Development of a Triple CubeSat, White Paper, June 4, 2003

[7] H. Bock, U. Hugentobler, T.A. Springer, G. Beutler, "Efficient precise orbit determination of LEO satellites using GPS, Advances in Space Research," Volume 30, Issue 2, July 2002, Pages 295-300, ISSN 0273-1177, 10.1016/S0273-1177(02)00298-3.

[8] D. Harris, Electronic Design. "If only the original Spartans could have thrived on so little power," February 27, 2008

[9] L. David (2006). CubeSat losses spur new development [Online] available at http://www.cnn.com/

[10] H. Heidt, C Turner, R. Twiggs, "CubeSat: A new generation of picosatellite for education and industry low-cost space experimentation." in *Proceedings of the 14th Annual AIAA/USU Conference on Small Satellites*, SSC00-V-5, Logan, UT, Aug. 2000

[11] H. Ashida, K. Fujihashi, S. Inagawa, Y. Miura, K. Omagari, et al, "Design of Tokyo Tech nano-satellite Cute-1.7+APD II and its operation," in *Acta Astronautica*, Volume 66, Issues 9-10, May-June 2010, Pages 1412-1424,ISSN 0094-5765, 10.1016/j.actaastro.2009.10.035.

[12] H.S. Neoh, A. Hazanchuk, "Adaptive Edge Detection for Real-Time Video Processing using FPGAs," *GSPx 2004 Conference*, 2004.

[13] H. Ding, L. Arber, L. Sha, M. Caccamo, "The Dependency Management Framework: A Case Study of the ION CubeSat," in *18th Euromicro Conference on Real-Time Systems* (ECRTS'06), pp. 52-64, 2006

[14] A. Jacobs, G. Cieslewski, C. Reardon, and A. George, "Multiparadigm Computing for Space-Based Synthetic Aperture Radar," in *Proceedings of 2008 International Conference on Engineering of Reconfigurable Systems and Algorithms (ERSA)*, Las Vegas, NV, July 14-17, 2008.

[15] A. Jacobs, C. Conger, A.D. George, "Multiparadigm Space Processing for Hyperspectral Imaging," in

Aerospace Conference, 2008 IEEE, Big Sky, MT, March 2008

[16] M. Kovac, N. Ranganathan, "JAGUAR: a fully pipelined VLSI architecture for JPEG image compression standard," in *Proceedings of the IEEE*, vol.83, no.2, pp.247-258, Feb 1995

[17] M. Long, A. Lorenz, G. Rodgers, E. Tapio, G. Tran, et al, "A CubeSat Derived Design for a Unique Academic Research Mission in Earthquake Signature Detection," in *16th Annual AIAA/USU Conference on Small Satellites*, Utah, USA, 2002.

[18] *Mike's list of CubeSat Satellite Missions* [Online]. Available at http://mtech.dk/thomsen/space/cubesat.php

[19] K. Morris, "Veni! Vidi! Virtex! (and Kintex and Artix Too)," In *FPGA Journal* June 21, 2010

[20] I. Nason, M. Creedon, J. Puig-Suari, "CubeSat Design Specifications Document," Revision V, pp. 1-6. Available at http://ssdl.stanford.edu/cubesat, Nov. 2001

[21] I. Nason, J. Puig-Suari, R. Twiggs, "Development of a family of picosatellite deployers based on the CubeSat standard," in *Aerospace Conference Proceedings, 2002. IEEE*, vol.1, pp. 1-457- 1-464 vol.1, 2002

[22] OmniVision, "Datasheet: Preliminary Specifications 1/5" CMOS (2 megapixel) image sensor with OmniPixel3-HS technology", available at http://www.trulydisplays.com/ccm/specs/2.0M%20Sensor%20OV2655%20Spec.pdf

[23] C. Eagle, *Shadow Conditions of the Earth in Circular Orbits* [Online]. Available at http://www.cdeagle.com/html/omnum.html

[24] P. Pingree, T. Werne, D. Bekker, T. Wilson, J. Cutler, M. Heywood, "The Prototype Development Phase of the CubeSat On-board processing Validation Experiment (COVE)," in *IEEE Proc. 2011 Aerospace Conference*, Big Sky, MT, 2011

[25] J. Puig-Suari, C. Turner, W. Ahlgren, "Development of the standard CubeSat deployer and a CubeSat class PicoSatellite," in *Aerospace Conference, 2001, IEEE Proceedings.* Vol.1, pp.1/347-1/353 vol.1, 2001

[26] J. Samson, J. Ramos, I. Troxel, R. Subramaniyan, A. Jacobs, et al, "High-Performance, Dependable Multiprocessor," in *Proc. of IEEE/AIAA Aerospace*, Big Sky, MT, Mar 4-11, 2006

[27] J. Schaffner, "The Electronic System Design, Analysis, Integration, and Construction of the Cal Poly State University CP1 CubeSat" in *16th AIAA/USU on Small Satellites Conference*, Logan, UT, October 2002, pp. 1-2

[28] A. Toorian, K. Diaz, S. Lee, "The CubeSat Approach to Space Access," in *Aerospace Conference, 2008 IEEE*, pp.1-14, 1-8 March 2008

[29] Xilinx Inc. Spartan-3 vs. Cyclone I Performance Analysis, White Paper, May 24th 2005

[30] Xilinx Inc. *7-series Power Estimator.* copywrite 2011

[31] Xilinx Inc. *Xilinx ISE Design Suite 13.2*. copywrite 2011

9. BIOGRAPHY

Scott Sterling Arnold received his Bachelor of Science in Electrical Engineering in 2011 from the University of Florida. Currently he is a PhD student at the University of Florida and is a student member of the NSF Center for High Performance Reconfigurable Computing (CHREC). During his undergraduate studies, he has worked for GE Appliances. His interests include CubeSat design, embedded systems design, computer architecture, reconfigurable and adaptive computing, and space computing.

Ryan Nuzzaci is a 2nd year Ph.D. student in Computer Engineering at the University of Florida. He is the lead hardware development engineer for the Small Sat Group and is currently designing the payload board for INTA's OPTOS II CubeSat. He received his B.S. and M.S. in Computer Engineering from the University of Florida in Dec. 2009 and Dec. 2011, respectively. Following graduation, he plans on pursuing a career in embedded systems design engineering. He is an active chair for UF IEEE.

Ann Gordon-Ross (M'00) received her B.S and Ph.D. degrees in Computer Science and Engineering from the University of California, Riverside (USA) in 2000 and 2007, respectively. She is currently an Assistant Professor of ECE at the University of Florida and is a member of the NSF Center for High Performance Reconfigurable Computing (CHREC). She is also faculty advisor for the Women in Electrical and Computer Engineering (WECE) and the Phi Sigma Rho National Society for Women in Engineering and Engineering Technology. She received her CAREER award from the National Science Foundation in 2010 and Best Paper awards at the Great Lakes Symposium on VLSI (GLSVLSI) in 2010 and the IARIA International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM) in 2010. Her research interests include embedded systems, computer architecture, low-power design, reconfigurable computing, dynamic optimizations, hardware design, real-time systems, and multi-core platforms.