



Politechnika Warszawska

Wydział Elektroniki i Technik Informacyjnych
Instytut Mikroelektroniki i Optoelektroniki

inż. Michał Gąska

Nr albumu: 218774

Praca magisterska

Moduł komputera pokładowego do satelity CubeSat z funkcją Star Tracker

Praca wykonana pod kierunkiem:

**Dr inż. Grzegorza Kasprówicza
Instytut Systemów Elektronicznych
Politechnika Warszawska**

WARSZAWA 2016

Moduł komputera pokładowego do satelity CubeSat z funkcją Star Tracker

W pracy magisterskiej przedstawiono konstrukcję *Modulu komputera pokładowego*. We wstępnie zaprezentowano budowę satelitów typu CubeSat oraz porównano dostępne na rynku komputery pokładowe. Następnie scharakteryzowano zagrożenia dla elementów elektronicznych przebywających w przestrzeni kosmicznej oraz przedstawiono rozwiązania zwiększające niezawodność pracy systemu. W kolejnych rozdziałach opisano koncepcję oraz realizację sprzętową zaprojektowanego urządzenia. Przedstawiono podstawowe oprogramowania testujące wykonanego moduł komputera pokładowego. W końcowej części zaprezentowano efekty działania skonstruowanego urządzenia oraz nakreślono dalszy kierunek prac.

Słowa kluczowe

CubeSat, FPGA, VHDL, Atmega, OBC, SDRAM, TMS570, Cortex-R4, Altera Quartus II, Mentor Graphics Hyper Lynx, Star Tracker, ADC, PCB, Signal Integrity

On-board computer module for CubeSat satellite with Star Tracker camera

In this master thesis *on-board computer* for a CubeSat satellite is described. In the introduction architecture of CubeSat and basics modules were presented. Commercially available CubeSat on-board computer were compared. Risks for the electronic components due to radiation and mitigation techniques were described. Then conception of the design and physical realization are given. The basic software and firmware for debugging were also depicted. In the summation plans for further development are drafted.

Keywords

CubeSat, FPGA, VHDL, Atmega, OBC, SDRAM, TMS570, Cortex-R4, Altera Quartus II, Mentor Graphics Hyper Lynx, Star Tracker, ADC, PCB, Signal Integrity

Pragnę złożyć serdeczne podziękowania:

Rodzicom za cierpliwość,

Mgr inż. Eweliny Użarowskiej za wiarę i motywację,

Prof. dr hab. inż. Krzysztofowi Poźniakowi za cenne uwagi,

Dr inż. Grzegorzowi Kasprowiczowi za pomoc w realizacji pracy magisterskiej,

Koleżankom i kolegom, za atmosferę i współpracę w laboratorium

PERG/ELHEP ISE PW

Spis treści

1 Wstęp	7
1.1 Satelity	7
1.2 Satelity typu CubeSat	9
1.3 Budowa nano satelitów	10
1.3.1 Komputer pokładowy	11
1.3.2 System zasilania	12
1.3.3 Komunikacja	13
1.3.4 System kontroli położenia	13
1.3.5 System napędowy	17
1.3.6 Mechanika	18
1.3.7 System kontroli termicznej	20
1.4 Podsumowanie	21
2 Przestrzeń kosmiczna	23
2.1 Próżnia	23
2.2 Wibracje	23
2.3 Promieniowanie	24
2.3.1 Single Event Upset	25
2.3.2 Single Event Latch-up	26
2.3.3 Single Event Gate Rupture	26
2.3.4 Single Event Burnout	26
2.3.5 Single Event Functional Interrupt	26
2.3.6 Single Event Transient	27
2.3.7 Total Ionizing Dose	28
2.4 Temperatura	29
2.5 Wpływ promieniowania na elementy COTS	30
2.5.1 Pamięci DRAM	30
2.5.2 Pamięć Flash	30
2.5.3 Mikroprocesory	31
2.5.4 Układy FPGA	31
2.6 Techniki zwiększające odporność na promieniowanie	33
2.6.1 Rozwiązania stosowane na poziomie architektury	34
2.6.2 Rozwiązanie stosowane na poziomie komponentu	35
3 Geneza, cel i założenia pracy	39
4 Koncepcja konstrukcji modułu komputera pokładowego	41
4.1 <i>Komputer pokładowy</i>	41

4.2	Podstawowe zadania komputera pokładowego	42
4.2.1	Obsługa modułów pokładowych satelity	42
4.2.2	Obsługa misji	43
4.2.3	Autodiagnostyka	43
4.2.4	Nawigowanie w kosmosie	44
4.3	Koncepcja konstrukcji	44
4.4	Plan realizacji	47
4.5	Podsumowanie	47
5	Realizacja modułu komputera pokładowego	49
5.1	Podsystemy komputera pokładowego	49
5.1.1	Główna jednostka obliczeniowa	51
5.1.2	Układ nadzorujący	54
5.1.3	System zasilania	55
5.1.4	Star Tracker	59
5.2	Projekt wielowarstwowego obwodu drukowanego PCB	65
5.3	Podsumowanie	66
6	Oprogramowanie modułu komputera pokładowego	67
6.1	Oprogramowanie mikrokontrolera ATmega128	67
6.1.1	Koncepcja oprogramowania	68
6.1.2	Realizacja oprogramowania	71
6.2	Oprogramowanie układu FPGA Cyclone III	76
6.2.1	System NIOS II	76
6.3	Oprogramowanie mikrokontrolerów TMS570	83
7	Uruchomienie i testowanie	85
7.1	Badanie generacji napięć zasilających	85
7.2	Uruchamianie mikrokontrolera ATmega128A	86
7.2.1	Badanie czasu reakcji na zabezpieczenia systemu zasilania	86
7.3	Uruchamianie układu FPGA	88
7.3.1	Testy poprawności pracy pamięci SDRAM	88
7.4	Uruchamianie układów TMS570	89
8	Wnioski końcowe	91
A	Dodatki	93
A.1	Płyta CD	93
A.2	Wielowarstwowy obwód drukowany	94
A.3	Interfejs SpaceWire oraz CubeSat Space Protocol	96
A.3.1	SpaceWire	96
A.3.2	CubeSat Space Protocol	96
A.4	Zaprojektowana płytka CubeSat Motherboard	98
A.5	Symulacje SI	99
Bibliografia		101

Rozdział 1

Wstęp

W niniejszym rozdziale krótko omówiono współczesne satelity a w szczególności skupiono się na CubeSat'ach. Scharakteryzowano konstrukcję nano-satelitów i przedstawiono podstawowe komponenty z jakich są zbudowane. Dużą uwagę poświęcono modułowi komputera pokładowego. W podsumowaniu porównano systemy komputerowe dostępne na rynku pod względem architektury oraz ceny. Ponadto, wyciągnięto wnioski dotyczące możliwości rozbudowy, funkcjonalności oraz łatwości implementacji różnego rodzaju algorytmów.

1.1. Satelity

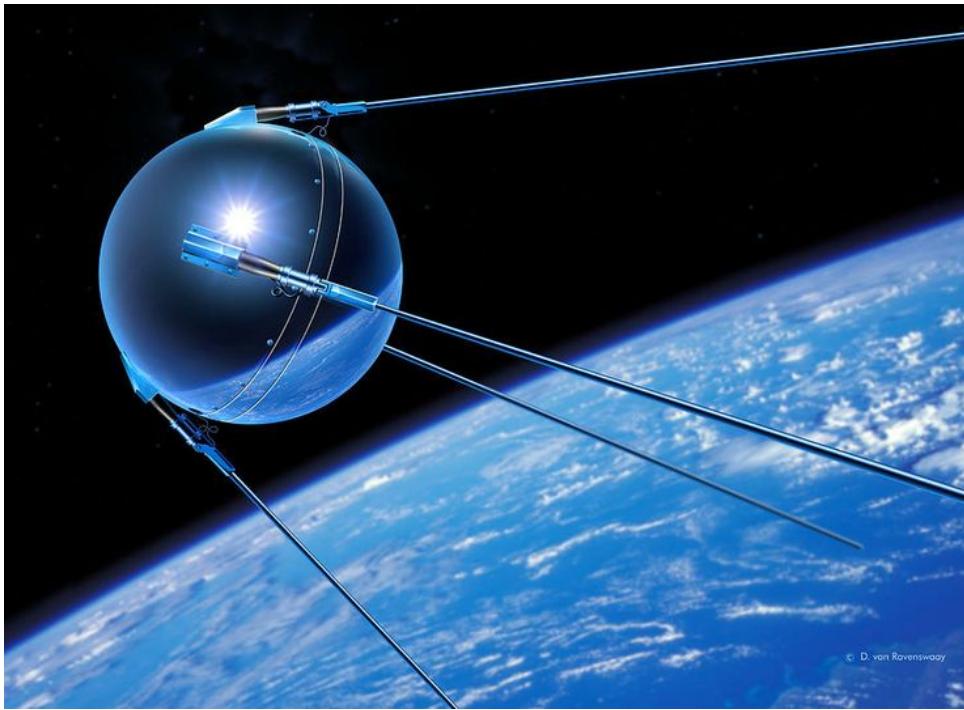
Pierwszym na świecie sztucznym satelitą, zbudowanym przez człowieka, był Sputnik 1 [1] (rys. 1.1), wystrzelony przez Związek Radziecki w roku 1957. Od tamtego czasu, setki satelitów zostało umieszczone na orbicie okołoziemskiej. Niektóre z nich, szczególnie Międzynarodowa Stacja Kosmiczna, zostały wystrzelone w częściach i złożone na orbicie. Ponad 40 państw buduje swoje sztuczne satelity mimo iż tylko 10 krajów posiada możliwości wynoszenia satelitów na orbitę [3]. Obecnie, na orbicie okołoziemskiej znajduje się kilkaset czynnych satelitów oraz tysiące nieużywanych już satelitów i ich fragmentów okrążających Ziemię jako kosmiczny złom. Ludzkość wystrzeliła w przestrzeń kosmiczną kilka sond które zostały umieszczone na orbitach innych ciał niebieskich stając się sztucznymi satelitami dla Słońca, Księżyca, Wenus, Marsa, Jowisza, Saturna, planetoid Westa oraz Eros, planety karłowatej Ceres [3].

W trakcie pisania niniejszej pracy Europejska Agencja Kosmiczna wystrzeliła w kierunku Marsa orbiter rozpoczynając tym samym misję badawczą ExoMars [4] mającej na celu znalezienie oznak życia na czerwonej planecie (rys. 1.2).

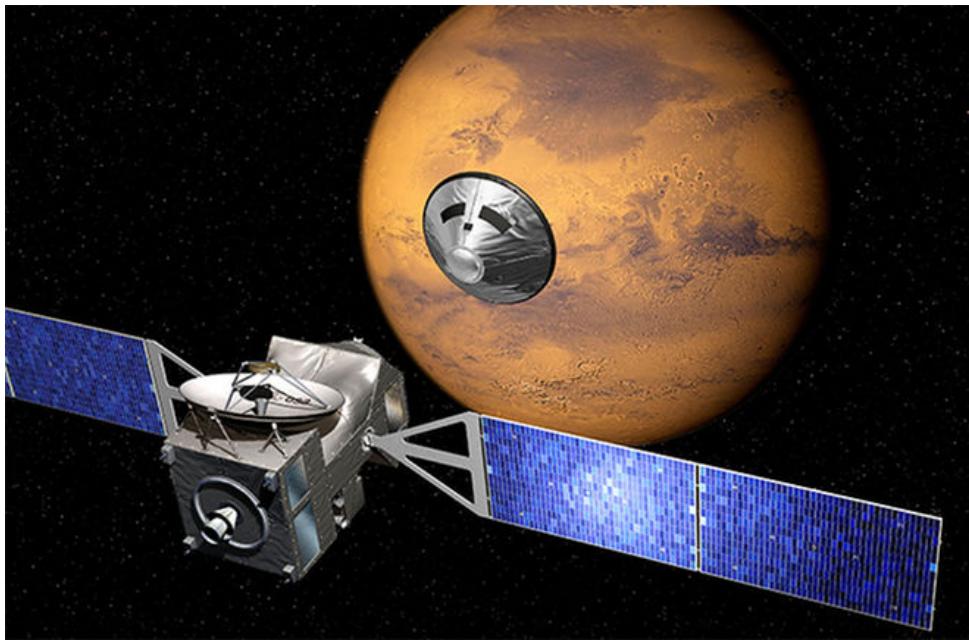
Satelity wykorzystywane są do wielu różnych celów. Najczęstsze typy [3] to:

- wojskowe i cywilne satelity obserwujące Ziemię,
- satelity komunikacyjne,
- satelity nawigacyjne,
- satelity pogodowe,
- oraz satelity badawcze.

Stacja kosmiczna i pojazdy kosmiczne znajdujące się na orbicie to także satelity. Orbity satelitów są bardzo zróżnicowane i zależą od przeznaczenia satelity. Orbity dzielimy na niską



Rysunek 1.1: Model satelity *Sputnik 1* produkcji ZSRR [2]

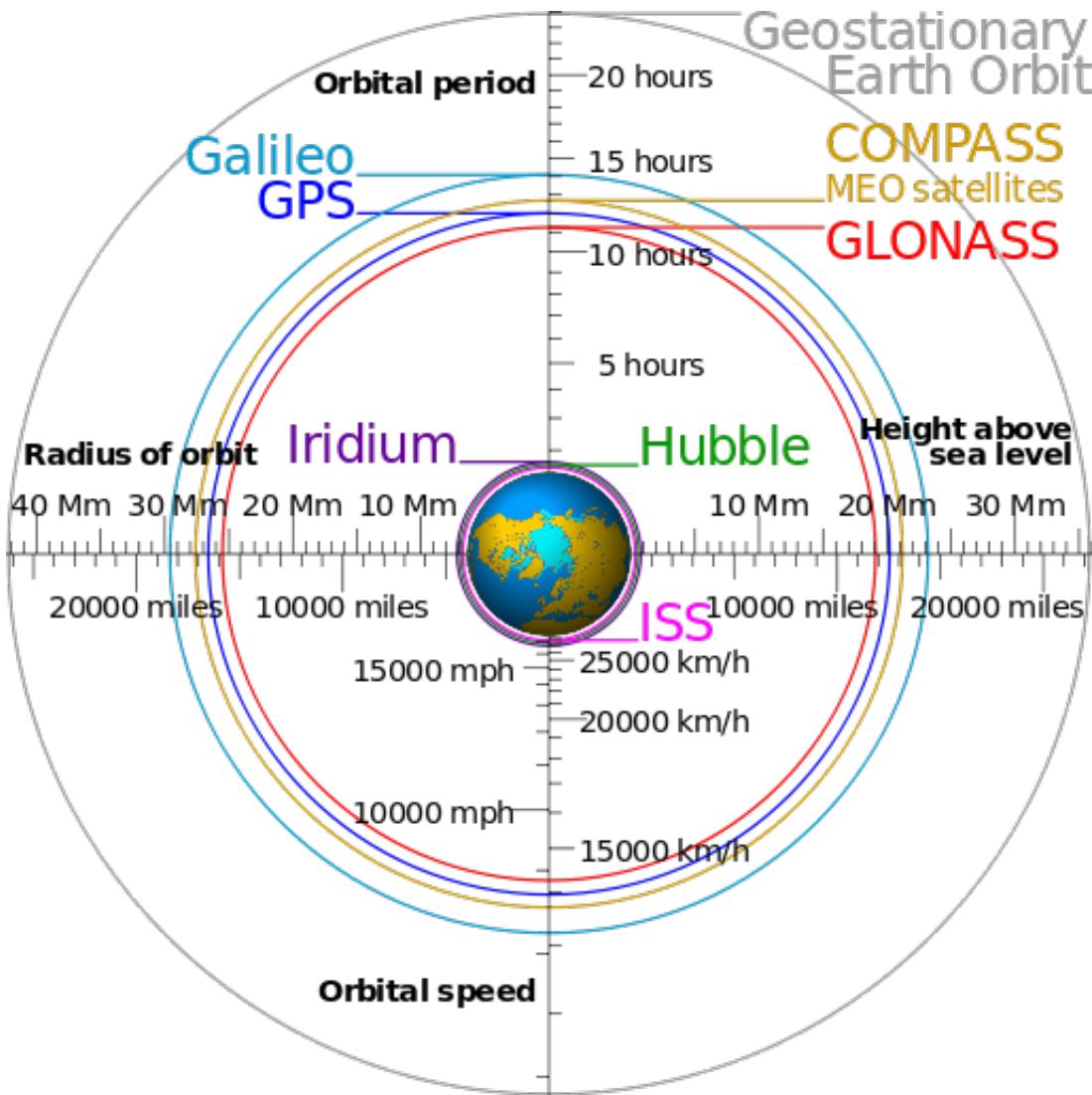


Rysunek 1.2: Model orbitera misji *ExoMars 2016* [4]

orbitę okołoziemską (ang. LEO), średnią orbitę okołoziemską (ang. MEO), orbitę geostacjonarną oraz wysoką orbitę okołoziemską. Rysunek (1.3) przedstawia orbity okołoziemskie.

W przestrzeni kosmicznej wystrzelonych zostało około 6600 satelitów. Szacuje się że na orbicie pozostało około 3600 z nich [6] z czego tylko 1000 nadal funkcjonuje. Reszta satelitów okres użyteczności ma już dawno za sobą i są częścią kosmicznych śmieci. Szacunkowo 500 satelitów znajduje się na niskiej orbicie okołoziemskiej, 50 na średniej orbicie okołoziemskiej a pozostałe na orbicie geostacjonarnej [6].

Satelity wynoszone są na orbitę na pokładzie rakiety nośnej. W przeważającej większości rakiety wystrzeliwane są z platform startowych umieszczonych na ziemi. W niewielu przypadkach satelity wystrzeliwane są z morza (z pokładu łodzi podwodnych lub z mobilnych



Rysunek 1.3: Orbity okoziemskie [5]

platform morskich) bądź też z pokładu samolotu [7].

Satelity zazwyczaj są samodzielnymi systemami kontrolowanymi przez wbudowany komputer pokładowy. Ich podsystemy przystosowane są do wykonywania wielu zadań, takich jak generowanie energii, telemetria, komunikacja ze stacją naziemną, sterowanie temperaturą podzespołów, kontrolowanie położenia i korekcja orbity.

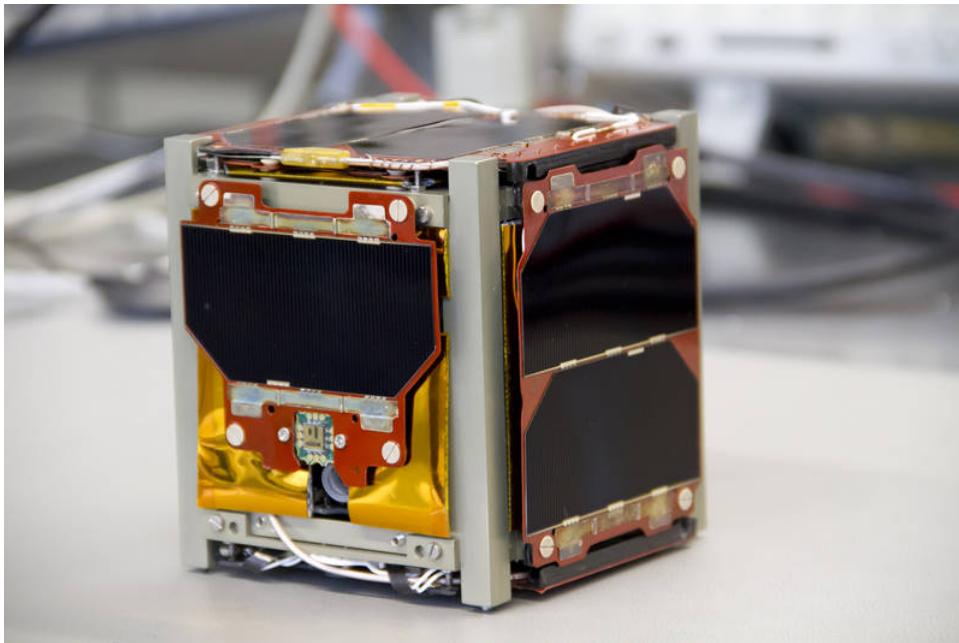
Ze względu na masę satelity możemy podzielić na [8]

- mini-satelity (100 - 500 kg),
- mikro-satelity (10 - 100 kg),
- nano-satelity (1 - 10 kg),
- piko-satelity (0.1 - 1 kg),
- femto-satelity (0.01 - 0.1 kg).

1.2. Satelity typu CubeSat

CubeSat (rys. 1.4) to rodzaj zminiaturyzowanej satelity (nano-satelity) przeznaczonej do badań kosmosu składającej się z wielokrotności sześciadanów (1U) o rozmiarach 10cm x 10cm

x 11.35cm i masie jednostkowej nie przekraczającej 1.33kg [8]. Struktura CubeSatów jak i elektroniczne systemy pokładowe zazwyczaj wykonywane są z elementów dostępnych komercyjnie (ang. COTS – Commercial Off-The-Shelf). Nano satelity najczęściej umieszczane są na orbicie przez specjalne wyrzutniki znajdujące się na Międzynarodowej Stacji Kosmicznej lub stanowią drugorzędny ładunek rakiet nośnych.



Rysunek 1.4: Zdjęcie satelity *CubeSat First-MOVE* [9]

W roku 1999, *California Polytechnic State University* i *Stanford University* opracowały pierwszą specyfikację CubeSatów [10] w celu promowania i rozwijania umiejętności niezbędnych do projektowania, wytwarzania i testowania małych satelitów przeznaczonych do umieszczenia na niskiej orbicie okoziemskiej. Satelity te miały być przystosowane do wykonywania szeregu funkcji badawczych i rozwijania nowych technologii kosmicznych. Uniwersytety produowały w produkcji nano satelitów do roku 2013, kiedy to ponad połowa CubeSatów nie miała akademickiego pochodzenia. Już w 2014 roku większość nowo wdrożonych małych satelitów przeznaczona była do zastosowań komercyjnych lub amatorskich. Obecnie technologia umożliwiająca budowę CubeSatów stała się powszechnie dostępna.

Zazwyczaj stosowane są do eksperymentów, które mogą być miniaturyzowane lub służą celom takim jak obserwacja Ziemi. Wiele CubeSatów wykorzystywanych jest do testowania technologii kosmicznych przeznaczonych do większych satelitów. Nano satelity doskonale nadają się także dla eksperymentów które są kwestionowane przez środowisko naukowe ponieważ niskie koszty uzasadniają większe ryzyko doświadczenia.

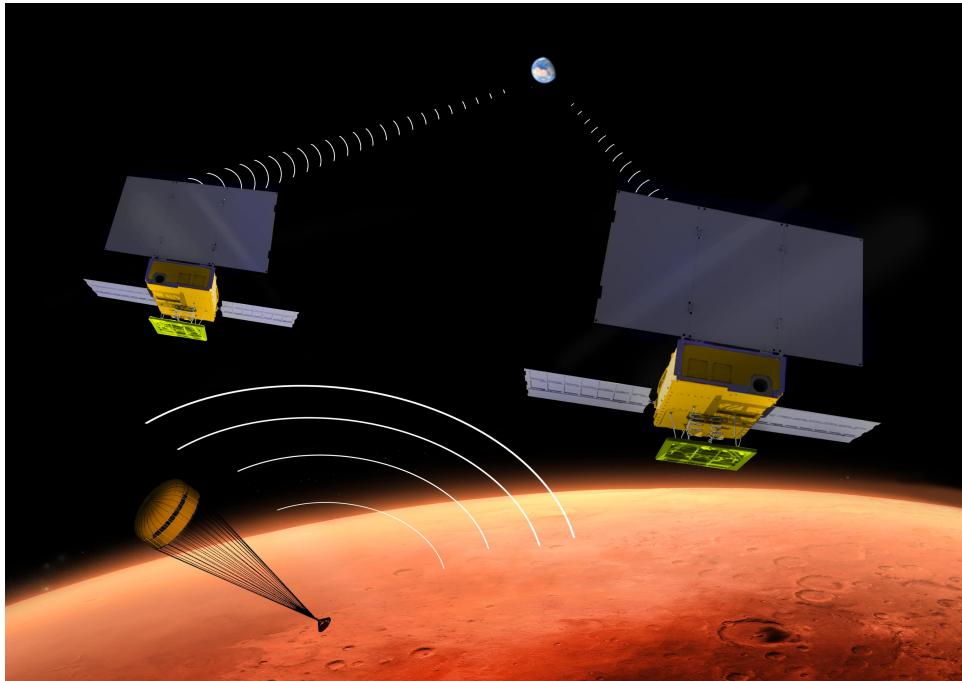
1.3. Budowa nano satelitów

Oficjalne specyfikacja CubeSatów . Głównym powodem miniaturyzacji satelitów jest znacząco niższy koszt wyniesienia na orbitę. W jednej rakiecie nośnej zmieści się wile małych CubeSatów , ponadto wypełnią one wolną przestrzeń obok głównego ładunku co radykalnie obniża cenę. Ujednolicenie wymiarów satelity i konstrukcji wyrzutnika umożliwia szybkie dostosowanie ładowni rakiety nośnej od aktualnego zapotrzebowania.

Standardowy CubeSat posiada wymiary 10 cm x 10 cm x 11.35 cm aby zapewnić 1 decy-

metr przestrzeni użytkowej. Więcej informacji na temat standardowych wymiarów znajduje się w podrozdziale 1.3.6. W ostatnich latach zaczęto budować CubeSat'y o rozmiarach wykraczających poza oficjalną specyfikację. Bardziej popularne stały się satelity o wielkości 6U (12 cm 24 cm 36 cm) i 12U (24 cm 24 cm 36 cm). Większe wymiary pociągnęły za sobą znaczący wzrost możliwości wykraczający poza obszar akademicki a otwierający drogę do zaawansowanych badań naukowych i zastosowań militarnych.

W 2018 roku NASA planuje wystrzelić w przestrzeń kosmiczną dwa CubeSat'y 6U rozpoczynając misję *Mars Cube One* [11] (rys. 1.5). Zadaniem satelitów będzie orbitowanie wokół Marsa i przesyłanie na Ziemię danych pochodzących z łazika znajdującego się na powierzchni czerwonej planety. Będą to pierwsze CubeSat'y w historii pracujące poza ziemską orbitą.



Rysunek 1.5: Model satelitów misji *Mars Cube One*[11]

1.3.1. Komputer pokładowy

Tak jak duże satelity, CubSat'y często dysponują kilkoma komputerami pokładowymi obsługującymi równolegle wiele zadań takich jak kontrola położenia w przestrzeni kosmicznej, zarządzanie energią, sterowanie ładunkiem oraz czuwanie nad poprawną pracą całego systemu. Moduł kontroli położenia, bazujący na elementach COTS, zazwyczaj posiada swój własny komputer zarządzający podobnie jak to jest w przypadku systemu zarządzania energią. Ładunek satelity (np. kamera), aby użytecznie pełnił swoją funkcję, musi być w stanie komunikować się z głównym komputerem co czasami wymaga zastosowania dodatkowego mikrokontrolera. Zazwyczaj spowodowane to jest ograniczeniami sprzętowymi (np. niewystarczającej ilość interfejsów zewnętrznych) głównej jednostki obliczeniowej. Ponadto, nadzędny komputer mógłby nie poradzić sobie na przykład z jednocześnie akwizycją danych z sensora obrazu i wykonywaniem zadań związanych z misją satelity. W konsekwencji spowodowałoby to jego przeciążenie i wprowadziło duże opóźnienia w realizacji innych celów.

Mimo to, główny komputer (rys. 1.6) może być wykorzystany do wykonywania zadań związanych z ładunkiem satelity, które mogą obejmować przetwarzanie obrazu, analizę danych i kompresję danych. Typowe zadanie nadziednego komputera obejmują delegowanie

zadań do innych komputerów, kontrolę położenia satelity, wykonywanie obliczeń niezbędnych do wykonania manewrów orbitalnych, pracę zgodnie z harmonogramem misji oraz zarządzanie temperaturą. Moduł komputera pokładowego jest bardzo wrażliwy na promieniowanie kosmiczne dlatego należy stosować odpowiednie techniki zwiększące niezawodność pracy w środowisku radiacyjnym, np. pamięć RAM wyposażoną w system kodowania korekcyjnego (ang. ECC RAM). W niektórych satelitach stosuje się冗余度 redundancy polegającą na implementacji kilku takich samych komputerów pokładowych w celu zmniejszenia ryzyka niepowodzenia misji.



Rysunek 1.6: Moduł komputera pokładowego *A712D* [12] firmy *GOMSPACE* [13]

1.3.2. System zasilania

CubeSat'y wykorzystują ogniwa słoneczne do zamiany światła słonecznego w energię elektryczną która magazynowana jest w lądowych bateriach litowo-jonowych. Akumulatory zapewniają ciągłość zasilania podczas gdy panele słoneczne nie są oświetlone a także w porach szczytu obciążenia. Satelity tego typu posiadają ograniczoną powierzchnię na montaż ogniw słonecznych na swoich ścianach zewnętrznych. Ponad to, muszą efektywnie współdzielić powierzchnię ścian bocznych z innymi elementami satelity, takimi jak anteny, czujniki optyczne, obiektywy kamer, systemy napędowe. Akumulatory litowo-jonowe cechują się wysokim stosunkiem magazynowanej energii do masy przez co znakomicie nadają się do wykorzystania na pokładzie orbitera kosmicznego w którym masa komponentów jest krytyczna. Proces ładowania i rozładowywania baterii jest obsługiwany przez dedykowany system elektroniczny zwany EPS (ang. Electrical Power System). Czasami akumulatory wyposażone są w grzejnik, wykonany zazwyczaj z drutu oporowego, przeznaczony do zapobiegania obniżenia się ich temperatury poniżej krytycznej wartości (około 0°C [8]) ponieważ może to skutkować uszkodzeniem baterii a nawet awarią satelity. Misje wymagające większego zapotrzebowania na energię elektryczną wykorzystują system nawigacyjny w celu osiągnięcia najbardziej efektywnego położenia paneli słonecznych w kierunku słońca, ponadto możliwe jest sterowanie ekspozycją samych modułów słonecznych. Najnowsze innowacje wprowadzają dodatkowe sprężyny które

rozkładają panele słoneczne jak tylko satelita zostaje uwolniony a także wyposażają moduły słoneczne w mechanizm noża termicznego który rozkłada je na rozkaz operatora satelity. CubeSat'y muszą mieć wyłączone zasilanie na czas startu rakiety nośnej w celu zapobiegania pracy w ładowni P-POD'a i w związku z tym wyposażane są w mechaniczny przełącznik odcinający zasilanie. Wyłącznik jest aktywny na czas przebywania satelity w P-POD'zie a w chwili opuszczania ładowni zasilanie zostaje włączone.



Rysunek 1.7: Moduł systemu zasilania *NanoPower P31u* [14] firmy *GOMSPACE*

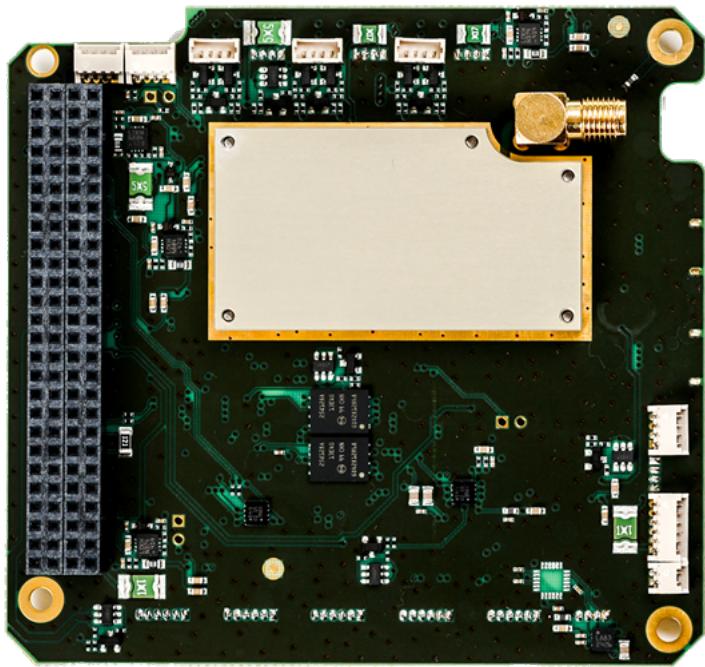
1.3.3. Komunikacja

Niska cena CubeSat'ów umożliwiła uzyskanie dostępu do przestrzeni kosmicznej małym instytucjom i organizacjom jednak zakres dostępnej mocy wyjściowej nadajników radiowych dla wszystkich beneficjentów jest ograniczony do około 2 W. Nano satelity wykorzystują systemy komunikacji radiowej na pasma VHF, UHF, L-, S-, C- i X-band [8]. Dla transmisji UHF i VHF najczęściej stosowane są anteny helikalne lub cztery anteny monopolowe rozkładane przy pomocy mechanizmu sprężynowego. Rysunek 1.8 przedstawia przykładowy moduł komunikacyjny.

Z powodu koziołkowania i niskiej mocy nadawczej, komunikacja radiowa stanowi wyzwanie dla inżynierów. Wiele CubeSat'ów używa anten dookólnych lub anten dipolowych dostępnych komercyjnie. Dla bardziej wymagających potrzeb, stosuje się anteny o wysokim wzmacnieniu jednak ich proces rozkładania i nakierowania jest wysoce skomplikowany.

1.3.4. System kontroli położenia

Kontrola położenia satelitów CubeSat polega na użyciu miniaturowej wersji dostępnej technologii bez znaczącego pogorszenia wydajności. Satelita, po uwolnieniu z ładownika, zaczyna koziołkować ze względu na działające na niego asymetryczne siły wypychające i obijanie się o inne CubeSat'y. Niektóre satelity mogą normalnie pracować nie zważając na koziołkowanie ale te które wymagają skierowania w określonym kierunku lub nie mogą pracować bezpiecznie podczas wirowania, muszą zostać ustabilizowane. Systemy które określają i kontrolują położenie (ang. ADCS - Attitude Determination and Control System), często wyposażone są w:

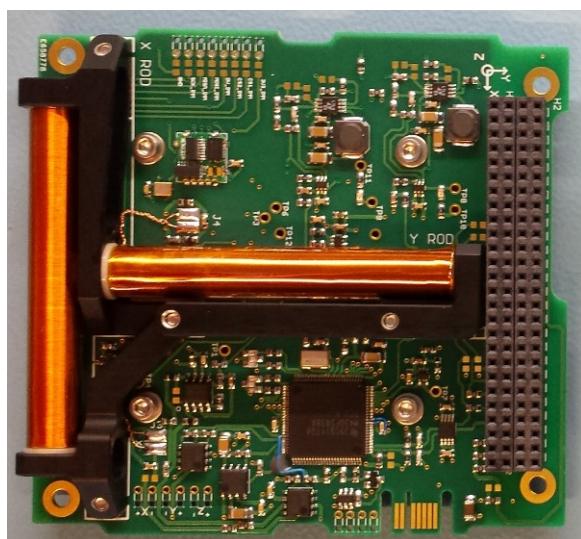


Rysunek 1.8: Moduł komunikacyjny *SatBus 3C0* [15] firmy *NanoAvionics* [16]

- koła reakcyjne (rys. 1.10),
- cewki indukcyjne (ang. magnetorquer)((rys. 1.9)),
- silniki odrzutowe (ang. thruster).

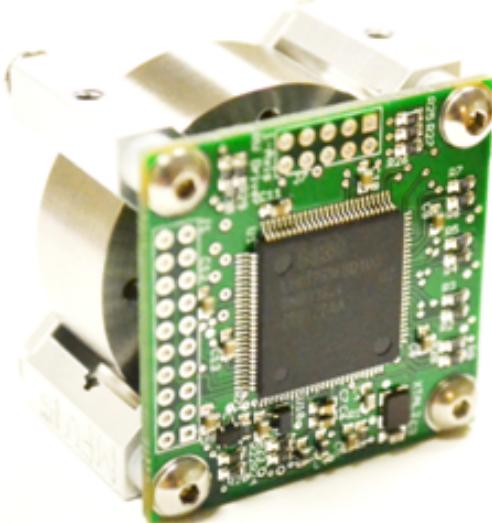
Systemy które określają położenie satelity w przestrzeni kosmicznej często bazują na:

- czujnikach śledzących gwiazdy (ang. star tracker),
- czujnikach położenia względem Słońca,
- czujnikach położenia względem Ziemi,
- czujnikach prędkości kątowej,
- odbiornikach sygnału GPS



Rysunek 1.9: Moduł z cewkami indukcyjnymi *MagneTorQuer* [17] produkowany przez *ISIS - Innovative Solutions In Space BV* [18]

Zazwyczaj stosuje się kombinacje tych czujników w celu zwiększenia funkcjonalności i złagodzenia ich wad. Koła reakcyjne są powszechnie stosowane ze względu na ich zdolności do osiągania relatywnie dużych momentów w stosunku do pobieranej energii. Jednak użytko-



Rysunek 1.10: Koło reakcyjne *MAI-400* [19] firmy *Maryland Aerospace* [20]

ność koła reakcyjnego jest ograniczona do limitowanej prędkości obrotowej. Działanie koła reakcyjnego może zostać wzmocnione poprzez użycie silnika rakietowego albo cewki indukcyjnej. Silnik rakietowy dostarcza dużego ciągu jednak jest nieefektywny w miniaturowych systemach napędowych ponieważ bardzo szybko zużywa całe dostępne paliwo. Stosowane w niemal wszystkich CubeSat'ach są cewki indukcyjne w których przepływający prąd elektryczny wytwarza pole magnetyczne które przeciwstawia się ziemskiemu i generuje moment hamujący ruch obrotowy. W satelitach, które wymagają tylko stabilizacji kątowej, nie stosuje się systemu określającego położenie. Wykorzystuje się tylko czujniki prędkości kątowej lub elektroniczne żyroskopy.

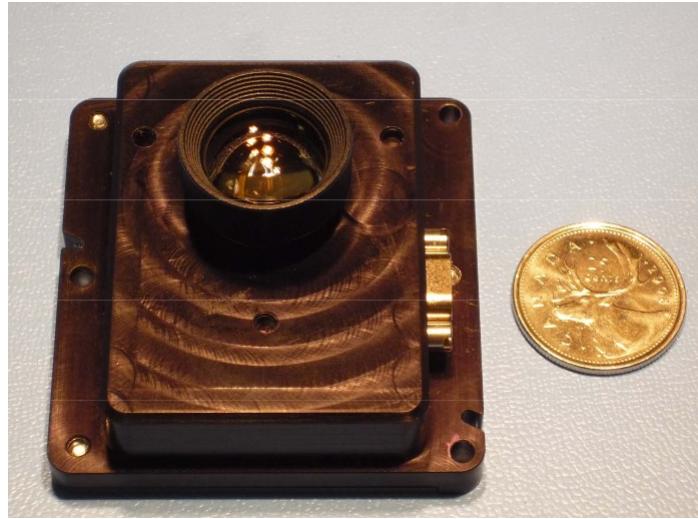
Określanie dokładnego położenia satelity w przestrzeni kosmicznej jest konieczne w przypadku obserwacji Ziemi, manewrów orbitalnych, maksymalizowania oświetlenia paneli słonecznych oraz niektórych instrumentów naukowych. Dokładność wyznaczania kierunku może zostać osiągnięta poprzez wykrywanie Ziemi i jej horyzontu, Słońca a także konkretnych gwiazd.

Star Tracker

Star Tracker [34] jest jednym z najdokładniejszych urządzeń przeznaczonych do wyznaczania orientacji w przestrzeni kosmicznej. Czujnik ten jest kamerą której jedynym zadaniem jest obserwowanie i wyznaczanie pozycji gwiazd na obserwowanej sferze niebieskiej. Star Tracker automatycznie oblicza położenia na podstawie obserwacji i identyfikacji gwiazd będących w jego polu widzenia.

Nowoczesne Star Tracker'y są urządzeniami cechującymi się wysoką czułością, niskim zapotrzebowaniem na energię oraz niewielką masą. Przykładowo, sensor ST-16 [21] (rys. 5.4) waży zaledwie 90g (waga bez przesłony), dostarcza informacje o położeniu z częstotliwością 10 Hz i cechuje się dokładnością poniżej 7 sekund kątowych.

Działanie czujników gwiazd może zostać zakłócone przez światło słoneczne odbite od satelity lub wskutek smugi rozchodzących się gazów pochodzących z silników napędowych. Dodatkowo, urządzenie to podatne jest na różnego rodzaju zakłócenia optyczne (aberracja sferyczna, aberracja chromatyczna). Istnieje również wiele potencjalnych źródeł nieprawidło-



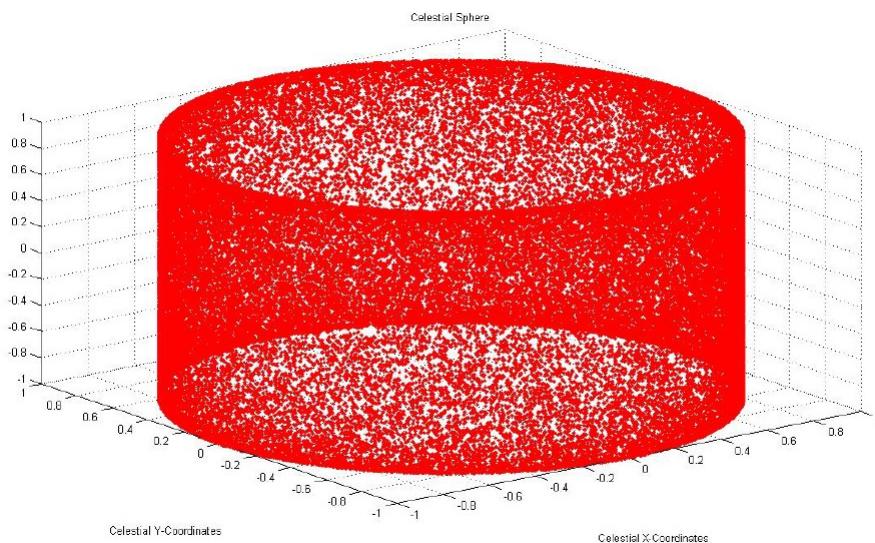
Rysunek 1.11: Star tracker *ST-16* [21]

wości powodujących zaburzanie pracy algorytmu identyfikującego gwiazdy (planety, komety, supernowe, inne pobliskie satelity).

Star Tracker powinien być umieszczony z dala do układu napędowego satelity aby zapobiec oślepianiu kamery przez gazy wylotowe. Ważne jest dodanie do obiektywu sensora obrazu przesłony w celu ograniczenia ilości światła oświetlającego czujnik.

Katalog gwiazd

Istnieje około 57 gwiazd nawigacyjnych powszechnie przeznaczonych do określania położenia. Jednakże, dla bardziej wymagających misji, wykorzystuje się znacznie więcej skatalogowanych ciał niebieskich. Katalog gwiazd stosowany do dokładnego wyznaczania orientacji, uzyskuje się z precyzyjnej dokumentacji sfery niebieskiej. Przykładowa baza danych o nazwie *Hipparcos* [23], zawierająca informacje o ponad stu tysiącach gwiazd udostępniona została przez Europejską Agencję Kosmiczną. Połowa katalogu *Hipparcos* została użyta do zilustrowania (rys. 1.12) sfery niebieskiej widocznej przez satelity przebywające na orbicie.



Rysunek 1.12: Ilustracja połowy katalogu *Hipparcos* w przestrzeni dwuwymiarowej [22].

1.3.5. System napędowy

W ostatnich latach w dziedzinie napędu satelitów CubeSat nastąpił szybki postęp, szczególnie w technologiach takich jak: zimny gaz, napęd chemiczny, napęd elektryczny oraz żągiel słoneczny [35]. Największym wyzwaniem jest zmniejszenie ryzyka awarii napędu podczas wynoszenia satelity w przestrzeń kosmiczną przy jednoczesnym zachowaniu wymaganej funkcjonalności. Komponenty i metody powszechnie stosowane w dużych satelitach w przypadku CubeSat'ów są zabronione lub ograniczone przez specyfikacje która zezwala na magazynowanie 100 Wh energii reakcji chemicznej oraz minimalizuje użycie materiałów niebezpiecznych [8].

Ograniczenie te stanowią wielkie wyzwanie dla satelitarnych systemów napędowych gdyż typowe silniki napędowe wykorzystują kombinację wysokiego ciśnienia, dużej gęstości energii oraz materiałów niebezpiecznych. Poza ograniczeniami nałożonymi przez usługodawcę wynoszenia satelitów, należy wziąć pod uwagę różne wyzwania technologiczne związane ze skomplikowaną mechaniką. Przykładem może być technologia wektorowego kierowania ciągu gdzie w przypadku CubeSat'ów należy zastosować kilka dysz rozmieszczonych asymetrycznie.

Silniki na zimny gaz

Silnik strumieniowy oparty o zimny gaz zazwyczaj przechowuje obojętny gaz, na przykład azot, w ciśnieniowym zbiorniku i uwalnia go przez dyszę w celu wytworzenia ciągu. Do operacji wyzwolenia potrzebny jest zaledwie jeden zawór co sprawia że zimny gaz jest najprostszą użyteczną technologią napędową. Takie silniki są bardzo bezpieczne ponieważ stosowane gazy nie muszą być lotne bądź źrące, jednak niektóre systemy wykorzystują niebezpieczne związki, takie jak dwutlenek siarki [24]. Możliwość użycia obojętnych gazów jest bardzo korzystna dla nano satelitów ponieważ ograniczone jest stosowanie materiałów niebezpiecznych na ich pokładzie. Niestety, silniki te osiągają niską wydajność co uniemożliwia wytworzenie silnych impulsów manewrujących nawet w CubeSat'ach o niskiej masie. Z tego powodu nie są stosowane jako główny układ napędowy i w tej dziedzinie ustępują miejsca wydajniejszym systemom o tylko niewiele większej złożoności. Częściej widuje się je jako silniki przeznaczone do korekcji położenia.

Silniki chemiczne

Chemiczne systemy napędowe wykorzystują reakcję chemiczną do produkcji gazu wylotowego o wysokim ciśnieniu i temperaturze. Paliwo użyte do reakcji chemicznej może być jednoskładnikowe lub dwuskładnikowe (tlen i paliwo). Zaletami jednoskładnikowych systemów jest wysoka niezawodność, niskie zapotrzebowanie na energię i stosunkowo mała złożoność przy zachowaniu dużej siły ciągu wyjściowego. Dzięki swojej prostocie, silniki te mogą być bardzo małe co sprawia iż znalazły zastosowanie na pokładach CubeSat'ów. Opracowane zostały także, bardzo wydajne systemy napędowe [25] oparte na hydrazynie [26]. Jednak z uwagi na to iż, hydrazyna jest substancją bardzo niebezpieczną, stosowanie jej na pokładzie CubeSat'a wymaga specjalnego pozwolenia. Znacznie bezpieczniejszym paliwem niewymagającym zezwoleń jest AF-M315 (azotan hydroksyloaminy) [27].

Napędy elektryczne

Elektryczne systemy napędowe wykorzystują energię elektryczną do przyspieszania materiału pędnego do dużej prędkości. Silniki te mogą być dostatecznie małe aby znalazły zastosowanie w nano satelitach. Technologie napędu elektrycznego przystosowane do użycia w CubeSat'ach to:

- silniki Halla [28],
- silniki jonowe [29],
- impulsowe silniki plazmowe (rys.1.13) [30],
- silniki „electrospray” [31],
- resistojet [32].



Rysunek 1.13: Impulsowy silnik plazmowy *BmP-220* [33] firmy *Busek*

Wysoka efektywność związana z napędem elektrycznym może umożliwić nano satelitom podróż do Marsa (np. wspomniana już misja MarCO). Wadą elektrycznych systemów napędowych jest wysokie zapotrzebowanie na energię co stwarza konieczność wykorzystania obszerniejszych paneli słonecznych, skomplikowanego systemu zarządzania i dystrybucji mocy oraz większych baterii. Ponadto, wiele technologii napędu elektrycznego wymaga stosowania zbiornika ciśnieniowego do przechowywania materiału pędnego.

Żagiel słoneczny

Żagiel słoneczny jest typem napędu kosmicznego wykorzystującym ciśnienie promieniowania słonecznego do rozprzędzenia ultra cienkiego lustra do wysokich prędkości bez pomocy materiału pędnego. Generowana siła ciągu jest proporcjonalna do powierzchni żagla. Niska masa sprawia że doskonale nadają się do zastosowań w nano satelitach. Jednakże, moduł żagla słonecznego w porównaniu do CubeSata jest dość dużym elementem i wymaga implementacji skomplikowanego mechanizmu rozkładającego który może ulec awarii.

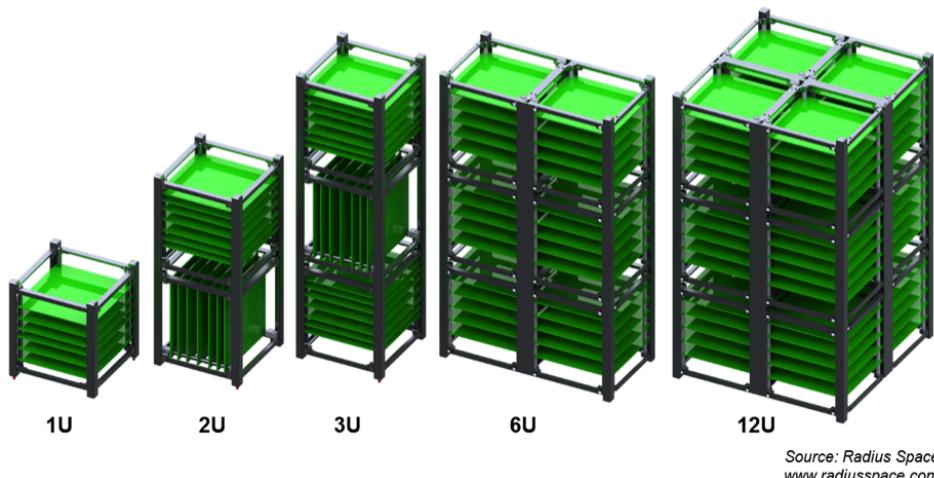
1.3.6. Mechanika

Ilość połączonych ze sobą jednostek podstawowych (1U) klasyfikuje wielkość CubeSat'a i zgodnie ze specyfikacją [10], satelita jest skalowalny tylko wzduż jednej osi do standar-dowych wymiarów 0.5U, 1U, 1.5U, 2U i 3U. Urządzenia o rozmiarach 6U, 12U i większe

powstają przez dołączanie jednostki podstawowej wzdułż wszystkich osi (rys. 1.15). Materiały wykorzystane do wytwarzania struktury nano-satelity muszą posiadać taki sam współczynnik rozszerzalności termicznej co surowiec z którego wykonany jest moduł wyrzutnika aby zapobiec zakleszczeniu. W szczególności, dozwolone są cztery materiały będące stopami aluminium: 7075, 6061, 5005 i 5052 [36]. Aluminium stosowane do wykonania elementów satelity które mają bezpośredni kontakt z P-POD'em [37] musi być dodatkowo anodowane w celu zapobiegania tzw. zimnemu zespawaniu. Poza tym, należy wziąć pod uwagę że nie wszystkie materiały mogą być wykorzystywane w próżni. Konstrukcje mechaniczne CubeSatów często wyposażone są w amortyzatory na każdym końcu, wykonane z miękkiej gumy, aby zmniejszyć oddziaływanie na pozostałe satelity umieszczone w wyrzutniku. Rysunek 1.14 przedstawia strukturę mechaniczną satelity wielkości 1U.



Rysunek 1.14: Struktura mechaniczna firmy *Pumpkin Inc.* [39] CubeSat'a wielkości 1U [38]



Rysunek 1.15: Porównanie wielkości CubeSat'ów [40]

Odstępstwa, od maksymalnych wymiarów dozwolonych przez standardową specyfikację, to najwyżej 6.5 mm z każdej strony. Wszystkie odstępstwa nie mogą kolidować z szynami prowadzącymi struktury rozmieszczającej i przeważnie, ta wolna przestrzeń zajmowana jest przez anteny i panele słoneczne. W aktualnej rewizji specyfikacji CubeSatów (rewizja numer 13 [10]) została zdefiniowana dodatkowa wolna przestrzeń dla satelitów o wielkości 3U. Ta dodatkowa pojemność została uzyskana poprzez zagospodarowanie wolnego miejsca w

pobliżu mechanizmu sprężyny w podajniku P-POD Mk III [41] (rys. 1.16). CubeSat'y 3U wykorzystujące tą wolną przestrzeń oznaczone są jako 3U+ i mogą mieć dodatkowy ładunek nie większy od cylindra umieszczonego w centralnym punkcie jednego z końców satelity. Cylindryczne przestrzenie ma maksymalną średnicę równą 6.4 cm i wysokość nie przekraczającą 3.6 cm. Ponadto masa satelity wielkości 3U+ musi być taka sama jak w przypadku 3U czyli 4 kg. Systemy napędowe oraz anteny są najpopularniejszymi elementami wykorzystującymi tę dodatkową przestrzeń.



Rysunek 1.16: Podajnik *P-POD Mk III*

Struktury CubeSat'ów nie mają takiej samej wytrzymałości mechanicznej, jak to jest w przypadku dużych satelitów, ponieważ dodatkową wytrzymałość niezbędną podczas startu zapewnia im konstrukcja P-POD'a. Mimo to, niektóre CubeSaty poddaje się testom wibracyjnym i analizom wytrzymałościowym w celu upewnienia się że elementy niewsparte przez P-POD nie ulegną uszkodzeniu podczas wszystkich etapów wynoszenia satelity. Mimo że, CubeSat'ów nie poddaje się tak restrykcyjnej analizie jak duże satelity, usterki mechaniczne zdarzają się bardzo rzadko.

1.3.7. System kontroli termicznej

Różne elementy CubeSat'ów posiadają inne akceptowalne zakresy temperatur, po których przekroczeniu, mogą stać się czasowo lub trwale niezdolne do użytku. Satelity na orbicie są podgrzewane energią promieniowania emitowanego przez Słońce, Ziemię, światła słonecznego odbitego od Ziemi, jak również ciepłem generowanym przez inne komponenty orbitera. CubeSat'y mogą także zostać schłodzone zarówno poprzez wypromieniowanie ciepła w przestrzeń kosmiczną jak i przez promieniowanie cieplne Ziemi pod warunkiem iż powierzchnia planety jest chłodniejsza do satelity. Przyjmuje się, że wszystkie radiacyjne źródła ciepła i proces rozpraszania są stałe i bardzo przewidywalne, tak długo, jak znana jest orbita orbitera i godziny zaćmienia.

W celu zapewnienia komponentom satelity temperatury nie przekraczającej dozwolonego zakresu, stosuje się wielowarstwową izolację oraz elementy grzejne. Inne techniki cieplne sto-

sowane w CubeSat'ach obejmują specyficzne rozmieszczenie komponentów bazujące na oczekiwanej wydzielonej mocy cieplnej. Symulacje i analiza modelu cieplnego statku kosmicznego jest bardzo ważnym elementem decydującym o wyborze odpowiedniej techniki zarządzania energią cieplną. CubeSat'y, o specjalnych wymaganiach temperaturowych dotyczących pewnych elementów pokładowych (np. mechanizmy), mogą zostać przetestowane przed startem w termicznej komorze próżniowej. Takie testy pozwalają na osiągnięcie większego stopnia pewności powodzenia misji niż w przypadku dużych satelitów ponieważ, CubeSat'y są na tyle małe że mogą zmieścić się w całości w komorze próżniowo termicznej. Czujniki temperatury są zwykle umieszczone przy różnych krytycznych komponentach aby mogły być podjęte odpowiednie działania gdy temperatura tych elementów zbliży się do niebezpiecznej wartości. Działania te mogą polegać na przemieszczeniu satelity w przestrzeni kosmicznej tak aby uniknąć lub wystawić, daną część, na bezpośrednie działanie promieniowania cieplnego w celu ogrzania bądź schłodzenia jej.

1.4. Podsumowanie

W tab. 1.1 zestawiono dostępne komercyjnie komputery pokładowe do satelitów CubeSat. Większość z nich oparta jest o nowoczesny mikrokontroler Cortex jednak żaden z tych układów nie jest fabrycznie przystosowany do aplikacji gdzie kluczowy nacisk postawiono na bezpieczeństwo wykonywanego programu. Ponadto, systemy te nie zawierają redundantnych kontrolerów przez co, w przypadku awarii głównego procesora, cały satelita staje się bezużyteczny.

Tylko jeden komercyjnie dostępny komputer pokładowy integruje w jednym systemie układ FPGA i mikrokontroler. W urządzeniu *Cube Computer*, układ programowalny przeznaczony jest jedynie do zabezpieczenia pamięci techniką EDAC. Brak możliwości implementacji dodatkowych algorytmów, takich jak kompresor JPEG lub analiza obrazu, ogranicza funkcjonalność komputera pokładowego. W przypadku bardziej złożonych urządzeń pokładowych satelity, wymagających szybkiego przesyłania i analizy danych, sam mikrokontroler może nie być w stanie zapewnić dostatecznej mocy obliczeniowej z uwagi na ograniczenia sprzętowe. Z drugiej strony, system zawierający tylko układ FPGA wymaga zdrożenia softwarowego procesora do obsługi podsystemów satelity. Dlatego preferowana architektura komputera pokładowego powinna integrować układ programowalny oraz mikrokontroler.

Żaden z analizowanych urządzeń nie posiada wbudowanego interfejsu kamery. Czujnik obrazu pozwala na realizację Star Trackera oraz na obserwację praktycznie dowolnego obszaru kosmosu.

Biorąc pod uwagę ograniczenia sprzętowe i wysoką cenę komercyjnych komputerów pokładowych oraz zagrożenia opisane w rozdziale 4, przystąpiono do opracowania nowatorskiej konstrukcji którą zaprezentowano w dalszej części pracy.

	ISIS On Board Computer [42]	NanoMind A712D[43]	Cube Computer[44]	Nanosatellite On Board Computer - GPS Bundle[45]	MAI-SS Space Sextant [46] (moduł Star Tracker)	GOS CubeSat OBC[47]	SatBus 1C0[48]
Architektura procesora	ARM9	ARM7	ARM Cortex-M3	software ARM Cortex-M3	bd	Cortex M4	Cortex M4
Częstotliwoœ Redundantne procesory	400 MHz	40 MHz	48 MHz	100 MHz	bd	do 180 MHz	do 168 MHz
Układ FPGA	brak	brak	tak	tak	bd	brak	brak
Pamięć RAM	64MB	2MB	2 x 1MB	8MB	bd	1MB	192KB
Pamięć ROM	2x 8GB	4MB + 4MB	2 GB	4GB	bd	8GB	2x 16MB
Pamięć FRAM	256KB	brak	brak	brak	bd	brak	brak
Zegar RTC	2x	bd	bd	tak	bd	tak	tak
Interfejs CAN	brak	1x	1x	1x	1x	1x	1x
Interfejs I2C	1x	1x	2x	2x	bd	1x	2x
Star Tracker	brak	brak	brak	brak	tak	brak	brak
Cena	4300 Euro	4750 Euro	4500 Euro	10300 USD	32500 USD	bd	bd

Tablica 1.1: Porównanie dostępnych komputerów pokładowych (bd - brak danych).

Rozdział 2

Przestrzeń kosmiczna

W niniejszym rozdziale przedstawiono warunki środowiskowe panujące w przestrzeni kosmicznej. Scharakteryzowano zagrożenia jakim musi przeciwstawić się satelita w trakcie wynoszenia w kosmos a także podczas przebywania na orbicie. Ponadto, zaprezentowano techniki służące do redukcji wpływu szkodliwych efektów wywołanych promieniowaniem na systemy elektroniczne.

2.1. Próżnia

Sztuczne satelity poruszają się po niskiej orbicie okołoziemskiej [51] (ang. low Earth orbit - LEO) znajdującej się pomiędzy powierzchnią Ziemi a Pasami Van Allena [52], znajduje się na wysokości od 350 km do 1200 km [49]. Na takiej wysokości, ciśnienie atmosferyczne może zostać pominięte i zastąpione próżnią. W związku z tym, wszystkie komponenty satelity muszą przeciwstawić się próżni. Z próżnią wiążą się dwa zagrożenia:

- odgazowanie,
- odkształcenia mechaniczne spowodowane oddziaływaniem próżni.

2.2. Vibracje

Ładunek rakiety nośnej będzie poddany wibracjom i przeciążeniom podczas startu. Maksymalna amplituda jak i częstotliwość wibracji zostały przedstawione na rysunku 2.1 (na podstawie rakiety nośnej VEGA[50]).

DIRECTION	LONGITUDINAL		LATERAL	
Frequency Band (Hz)	5 - 45	45 - 100	5 - 25	25 - 100
Sine Amplitude (g)	≤ 0.8	≤ 1.0	≤ 0.8	≤ 0.5

Rysunek 2.1: Amplituda i częstotliwość wibracji według dokumentacji rakiety VEGA [50]

Efekty związane z przyspieszeniem podczas startu nie mogą być lekceważone. Ładunek rakiety VEGA musi być w stanie wytrzymać przeciążenie do 15g [50] nawet, jeśli w rzeczywistości będzie ono niższe.

Szczególną uwagę należy zwrócić na mocowanie cięższych komponentów. Ponad to, elementy elektroniczne mogą zostać oderwane wskutek zginania płytka PCB pod wpływem drgań

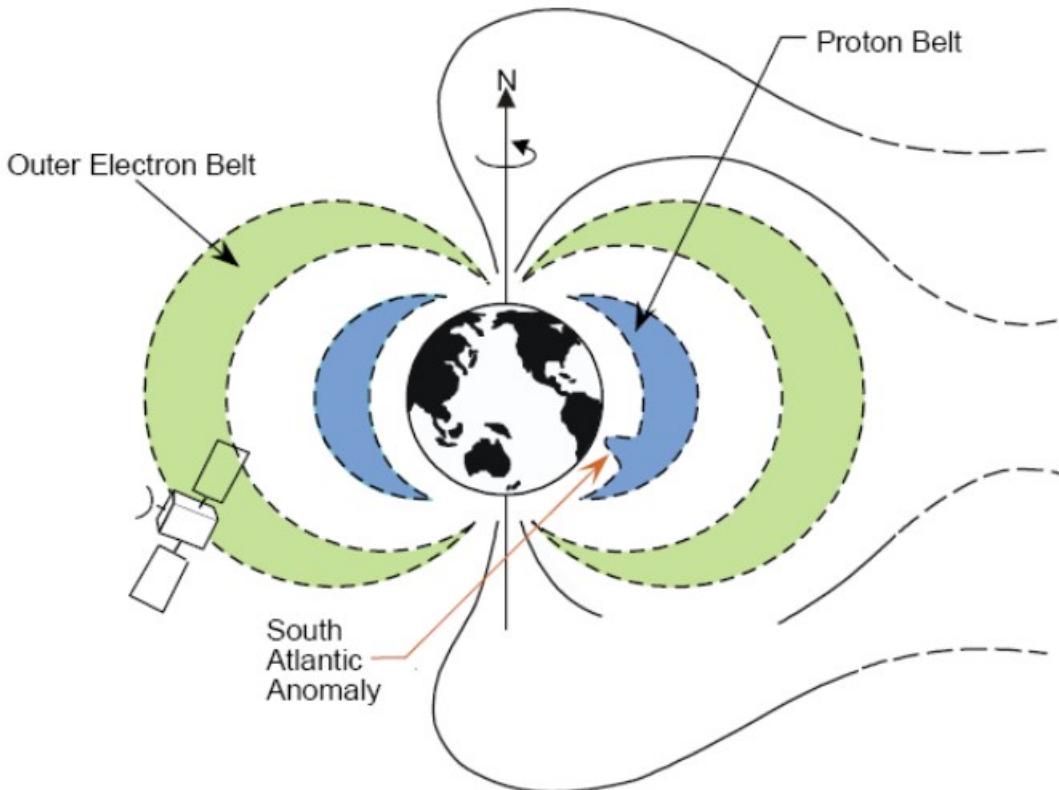
i przyspieszeń.

2.3. Promieniowanie

Naładowane cząstki wiatru słonecznego, głównie elektrony i proton, są wychwytywane przez ziemskie pole magnetyczne i umieszczone w pułapkach w których poruszają się po trajektorii w kształcie zbliżonym do helis. Helisy te nazywane są pasami Van Allen'a. Możemy wyróżnić dwa pasy:

- pas wewnętrzny – znajduje się pomiędzy 1000 km a 15000 km, zawiera wysokie stężenie energetycznych protonów o energii powyżej 100 MeV i elektrony o energii w zakresie od kilkuset keV [52],
- pas zewnętrzny – rozciąga się od 50000 km, składa się głównie z wysokoenergetycznych elektronów o energiach od 0.1 MeV do 10 MeV [52].

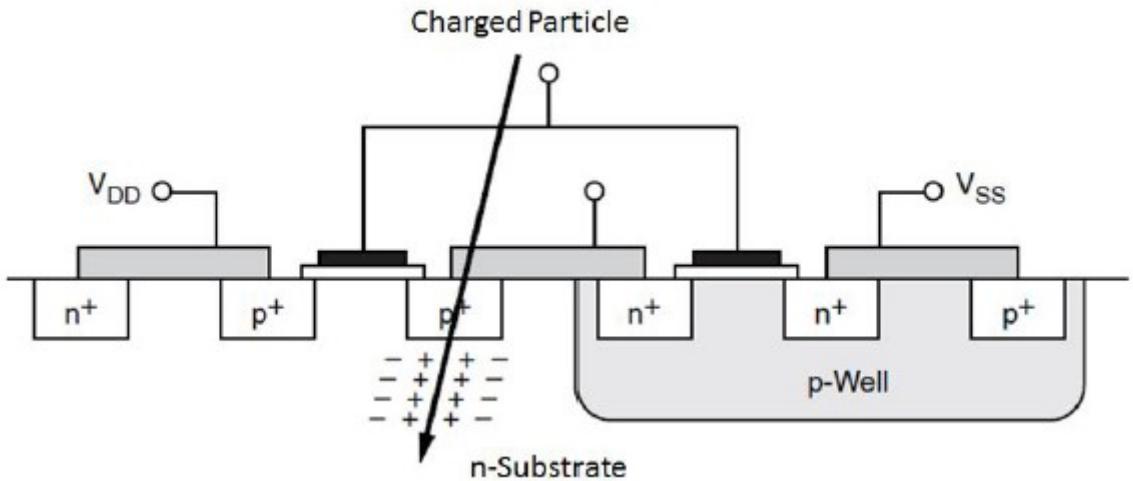
Dla satelitów umieszczonych na niskiej orbicie okołoziemskiej występuje utrudnienie związane z anomalią południowoatlantycką (rys. 2.2). W okolicach południowego Atlantyku wewnętrzny pas Van Allen'a zbliża się do Ziemi na odległość kilkuset kilometrów (200 km).



Rysunek 2.2: Pasy Van Allen'a oraz anomalia południowoatlantycka [53]

Jednym z głównych wymogów jaki stawia się przed komputerem pokładowym satelity jest to aby bezawaryjnie wykonywał powierzone mu zadania. Na satelitę umieszczonego na niskiej orbicie okołoziemskiej oddziałuje jonizujące promieniowanie kosmiczne. Co za tym idzie, wszystkie moduły pokładowe poddawane są działaniu destrukcyjnego promieniowania. Naładowane cząstki bombardując urządzenia elektroniczne zakłócają ich pracę. Wywołują szkodliwe zjawiska w strukturach krzemowych układów scalonych. Efekty te nazywają się między innymi *Single Event Effects* (SEE) [54, 72, 56]. Rysunek 2.3 reprezentuje proces gro-

madzenia ładunku przez naładowaną cząstkę podczas przelatywania przez fizyczną strukturę tranzystora.

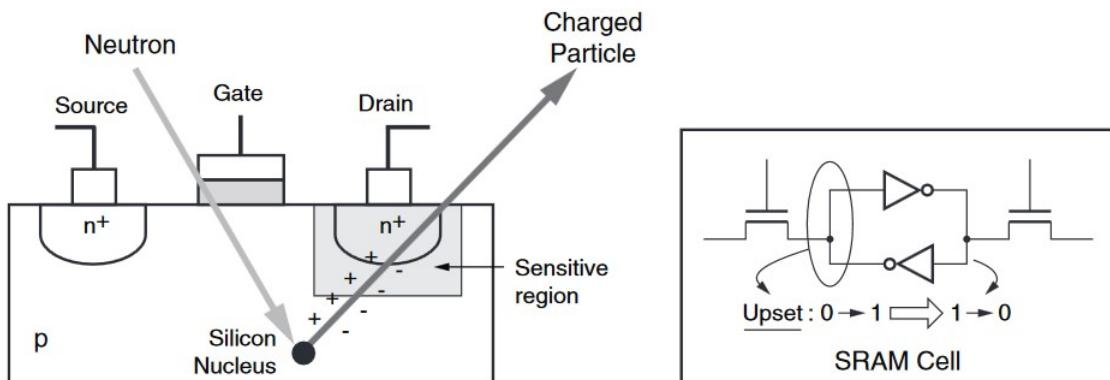


Rysunek 2.3: Ładunek gromadzony przez naładowaną cząstkę w podłożu tranzystora [57].

Cieżkie jony (promieniowanie kosmiczne i słoneczne) zazwyczaj są przyczyną SEE ponieważ powodują gromadzenie ładunku. Efekt ten jest mierzony przez *liniowy współczynnik przenoszenia energii* (ang. *LET - Linear Energy Transfer*) który odpowiada energii traconej przez jon na jednostkę długości znormalizowanej do gęstości konkretnego materiału [58]. Jednostką LET jest $\text{MeV}^*\text{mg}/\text{cm}^2$. Nie każdy ciężki jon przelatujący przez strukturę odda wystarczającą ilość energii aby wywołać zaburzenie ponieważ uderzające cząstki mają różną drogę przelotu przez obszar otaczający wrażliwy punkt.

2.3.1. Single Event Upset

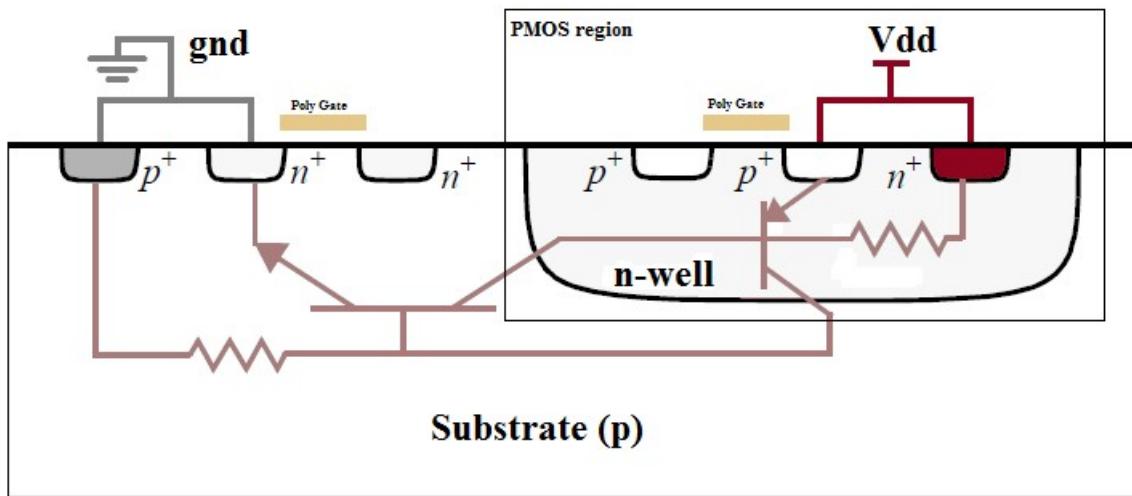
SEU - single event upset (SEU) jest to zmiana stanu spowodowana przez uderzenie pojedynczej naładowanej cząstki (jon, elektron, foton) w wrażliwy punkt układu mikroelektronicznego takiego jak mikrokontroler, pamięć półprzewodnikowa, tranzystor mocy [60, 56]. Zmiana stanu jest wynikiem powstania wolnego ładunku (elektrony lub dziury) wytworzzonego przez jonizację bezpośrednio w, lub w bliskiej odległości od komórki logicznej (np. komórki pamięci) (rys. 2.4).



Rysunek 2.4: Zmiana przechowywanej wartości w komórce pamięci wywołana przez SEU [59]

2.3.2. Single Event Latch-up

SEL - single event latch-up jest to rodzaj zwarcia które może powstać w układach scalonych. Dokładniej mówiąc, jest to przypadkowe wytworzenia niskoimpedancyjnej ścieżki pomiędzy szynami zasilającymi w obwodzie tranzystora MOSFET, wzbudzając pasożytniczą strukturę (rys. 2.5) która zakłóca funkcjonowanie przyrządu a nawet może spowodować jego nieodwracalne uszkodzenie w skutek przeciążenia prądowego [63]. Tylko wyłączenie zasilania przywróci normalne funkcjonowanie układu. Przyczyną powstawania single event latch-up jest SEU, typowo są to ciężkie jony lub protony pochodzące z rozbłysków słonecznych lub promieniowania kosmicznego [64]. Wytworzony pasożytniczy element, zazwyczaj jest odpowiadkiem tyristora, struktury PNPN składającej się z tranzystorów PNP i NPN ułożonych obok siebie. Podczas SEL, oba tranzystory zaczynają przewodzić i wzajemnie trzymają się w nasyceniu dopóki struktura krzemowa nie ulegnie uszkodzeniu.



Rysunek 2.5: Pasożytniczy tyristor powstały w strukturze tranzystor MOS [64]

2.3.3. Single Event Gate Rupture

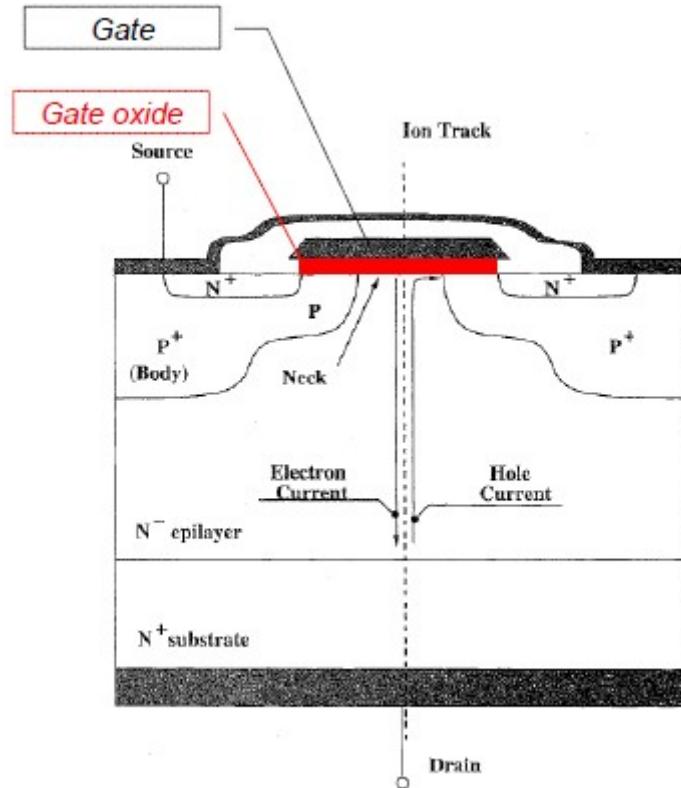
SEGR - single event gate rupture jest to nieodwracalne zniszczenie (pęknięcie) dielektryka w bramce tranzystora MOSFET spowodowane uderzeniami ciężkich, wysokoenergetycznych jonów w niektóre fragmenty struktury półprzewodnikowej (rys. 2.6) [65]. Na ten efekt szczególnie narażone są tranzystory mocy.

2.3.4. Single Event Burnout

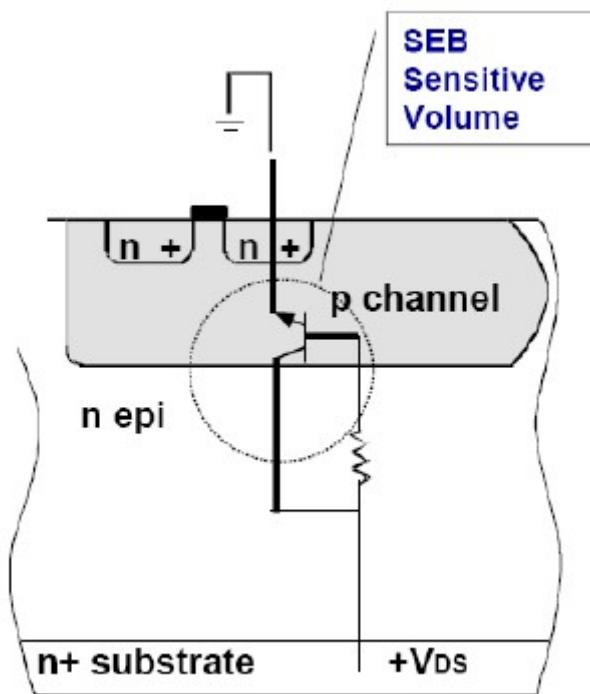
SEB – single event burnout efekt wywoływany jest przez wysokoenergetyczne jony przelatujące przez strukturę tranzystora. Polega na wytworzeniu pasożytniczego tranzystora bipolarnego który zwiera źródło drenem (rys. 2.7) [65].

2.3.5. Single Event Functional Interrupt

SEFI – single event functional interrupt – jest to błąd który powoduje reset układu scalonego, jego zawieszenie albo inne nieprawidłowe działanie [66]. W przypadku pamięci może wywołać przekłamanie w komórce z danymi. SEFI nie jest destruktywne dla komponentu elektronicznego i nie wymaga resetu zasilania.



Rysunek 2.6: Uderzenie jonu spowodowało zwężenie kanału tranzystor MOS [65]

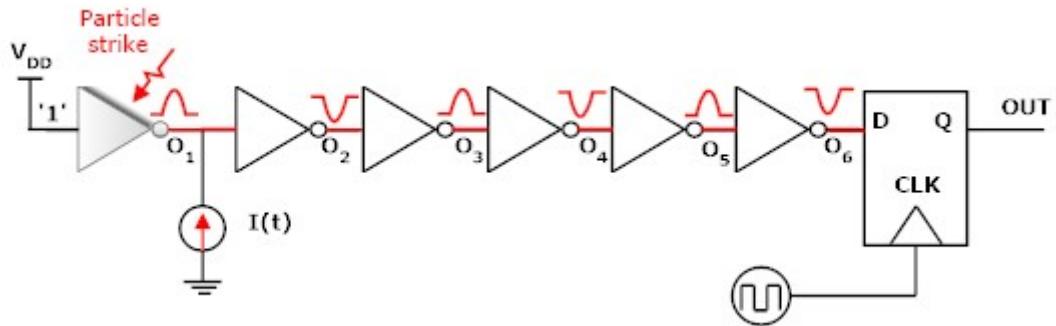


Rysunek 2.7: Paszytniczy tranzystor bipolarny wywołany efektem SEB [65]

2.3.6. Single Event Transient

SET – single event transient – jest to nieoczekiwana zmiana napięcia w układach logicznych wywoływaną przez wysokoenergetyczne cząstki przelatujące przez strukturę półprzewodnikową [62, 61]. Efekt SET może powodować zmianę stanu logicznego (rys. 2.8) i nie jest

destruktywny dla układu scalonego.

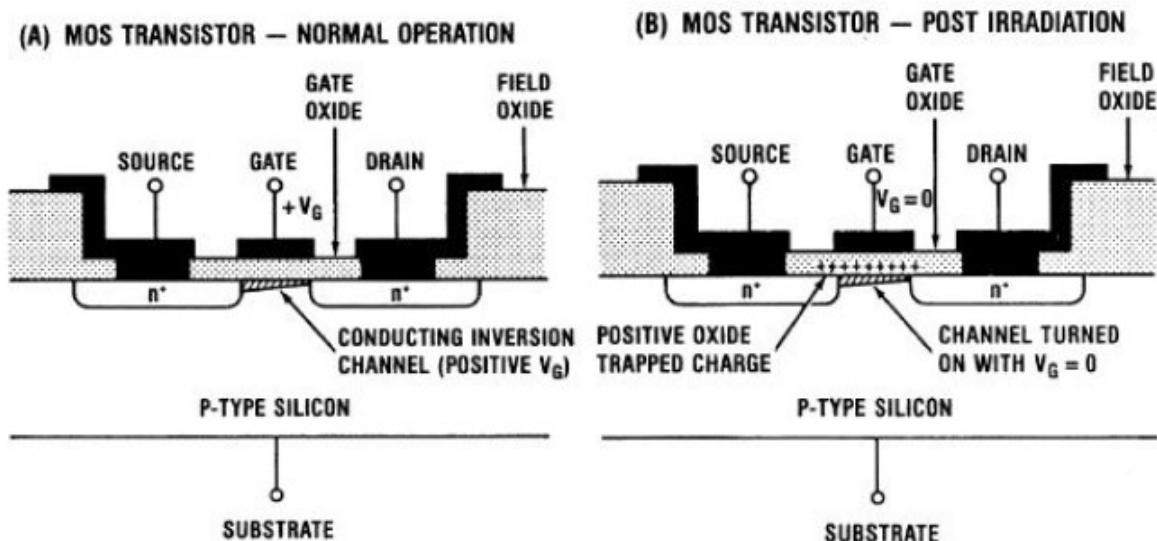


Rysunek 2.8: Zmiana stanu układu logicznego składającego się z kilku inwerterów [67]

2.3.7. Total Ionizing Dose

Są to skumulowane, długotrwałe szkody powstałe w wyniku jonizacji. Jonizacja jest to proces dodawania bądź usuwania elektronów z atomu pod wpływem zderzeń z ciężkimi jonomi, protonami a w mniejszym stopniu z promieniowaniem kosmicznym [69]. Pochłanianą dawkę można wyrazić jako lokalnie absorbowaną energię w wyniku jonizacji na jednostkę masy. Jednostką SI absorbowanej dawki jest **Grej**, jednak wciąż używa się **rad** (ang. radiation absorbed dose).

TID powoduje w komponentach CMOS powstawanie nowych par elektron-dziura i gromadzenie ładunku (np. w warstwie tlenku) (rys. 2.9). W konsekwencji [69, 70] może ulec zmianie prąd upływu, przebiegi czasowe, mogą zostać przesunięte napięcia progowe a nawet możliwe są błędy funkcjonalne.

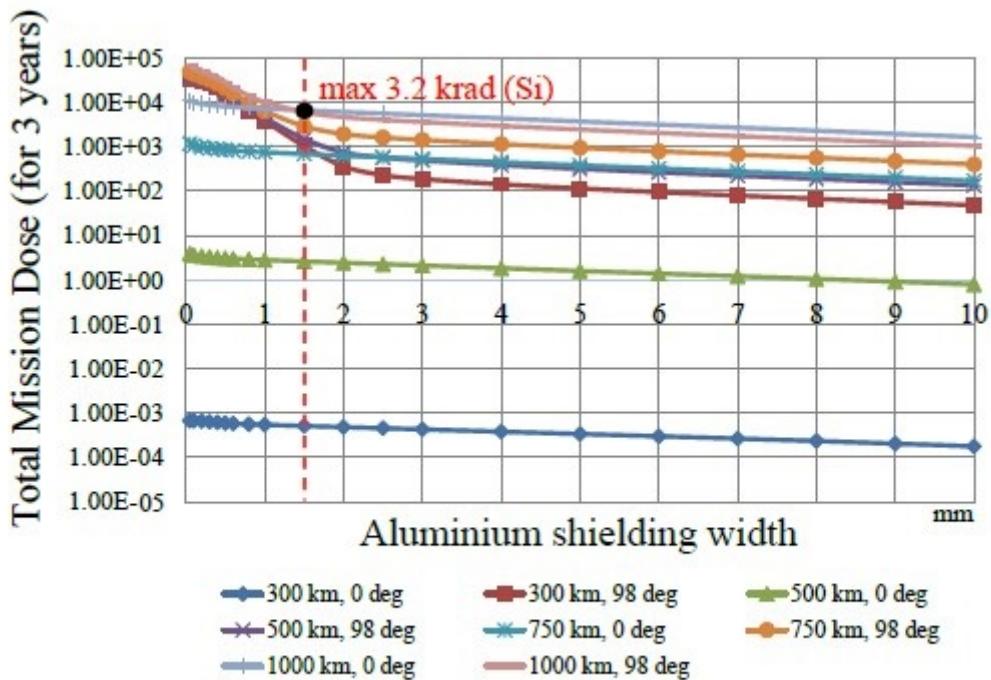


Rysunek 2.9: Gromadzenie się ładunku w tlenku bramkowym w N-kanalowym tranzystorze MOS: (a) normalna pracac (b) po napromieniowaniu [68]

Na ogólny, TID przyczynia się do pogorszenia się funkcjonalności układu na przestrzeni czasu [71]. Niektóre szkodliwe efekty powodowane przez TID mogą ulec samodzielnemu wygaszeniu jednak czas na to potrzebny zazwyczaj jest większy niż czas misji. Odpowiedni dobór

elementów elektronicznych i odpowiednie osłony metalowe może zapobiec krytycznej awarii spowodowanej TID [72].

Nawiązując do przykładowych wyników symulacji [przypis do publikacji] wykonanych w oprogramowaniu ESA SPENVIS [73] udostępnionym przez Europejską Agencję Kosmiczną, oszacowana została wartość TID dla trzyletniej misji kosmicznej. Jest to typowa długości misji satelity CubeSat. Różne możliwe scenariusze (zależne od wysokości, nachylenia, i grubości osłony) zostały zaprezentowane na rysunku 2.10.



Rysunek 2.10: Wpływ ekranowanie na TID [74]

Z symulacji wynika iż już aluminiowa osłona o grubości 1.5 mm znaczaco zmniejsza TID. Według wykresu, w najgorszym przypadku TID dla trzyletniej misji na wysokości pomiędzy 300 km a 750 km odpowiada wartości 3.2 krad. Poprawność tych symulacji potwierdza model empiryczny APEXRAD [75] udostępniony przez NASA (National Aeronautics and Space Administration).

Nie da się całkowicie wyeliminować powyższych zjawisk. Należy wziąć pod uwagę fakt że mogą one wystąpić w każdej chwili pracy systemu. Dlatego projektując architekturę systemu zastosowano pewne rozwiązania pozwalające ograniczyć szkody wyrządzane przez powyższe zjawiska.

2.4. Temperatura

Temperatura satelity w przestrzeni kosmicznej, kiedy urządzenia elektroniczna są włączone, w znacznej mierze zależy będzie od systemu kontroli termicznej. Nie w każdym miejscu CubeSat'a temperatury będą takie same. Symulacje termiczne dają pewne wyobrażenie na temat rozkładu temperatur w satelicie. W niniejszej pracy posłużono się symulacją wykonaną na potrzeby misji satelity PW-Sat 2 [76] z której wynika, iż temperatura satelity będzie mieścić się w zakresie od -20°C do +100°C.

Temperatury przy wystrzeliwaniu satelity

Startująca rakieta nośna przecina kilka warstw atmosfery cechujących się różnymi temperaturami. Podczas startu, we wnętrzu rakiety Vega, satelita CubeSat musi znieść temperatury od -40°C do $+80^{\circ}\text{C}$ [50]. Komponenty komputera pokładowego (przy wyłączonym zasilaniu) muszą być w stanie znieść te skrajne temperatury panujące w ładowni rakiety. Elementy elektroniczne o temperaturze pracy od -40°C do $+100^{\circ}\text{C}$ nie powinny mieć problemu z przetrwaniem lotu.

2.5. Wpływ promieniowania na elementy COTS

Nowoczesne komputery pokładowe zbudowane są z wielu różnych komponentów COTS. Tak więc wymagane jest oszacowanie wpływu promieniowania na powszechnie dostępne elementy elektroniczne. W dalszej części przedstawione jest oszacowanie wpływu promieniowania na poszczególne klasy układów scalonych.

2.5.1. Pamięci DRAM

Używając pamięci SDR RAM i DDR RAM w aplikacjach kosmicznych należy rozwiązać dwa główne problemy [80]. Pierwszym z nich jest tolerancja na zmianę stanu wielu bitów spowodowane przez SEU. Jon, uderzający w układ pamięci może wywołać zmianę stanu logicznego pojedynczego lub wielu bitów. W konsekwencji, spowoduje to przekłamanie całego słowa bitowego lub nawet, modyfikację wielu słów w zależności od architektury układu. Drugim problemem jest tolerancja na SEFI. Wysokoenergetyczny jon, uderzający w krzemową strukturę pamięci może spowodować usterkę całego układu.

W rezultacie pamięć DRAM wykazuje tolerancje wobec oczekiwanej, całkowitej dawki promieniowania podczas misji. [Przypis] Odnosząc się do źródeł, nowoczesne pamięci DRAM nie są podatne na SEL w czasie trwania misji małych satelitów CubeSat. Jednakże, układy ta są bardzo wrażliwe na SEU i SEFI.

2.5.2. Pamięć Flash

Pamięć typu Flash jest to pamięć nieulotna, elektronicznie kasowalna i programowalna [81]. Podstawowym elementem przechowującym jest tranzystor który składa się z bramki sterującej położonej nad bramką pływającą (ang. Floating Gate - FG), źródła oraz drenu [82]. Wewnętrzna pompa ładunku generuje wysokie napięcie (wyższe od napięcia zasilającego) niezbędne do programowania i kasowania pamięci. Generator wysokiego napięcia jest elementem wrażliwym na promieniowanie [83]. Kolejnymi komponentami pamięci Flash, czulymi na radiację, są skomplikowane układy sterowania (Finite State Machines, bufora wyjściowe).

Pamięci COTS Flash są powszechnie wykorzystywane w aplikacjach komercyjnych i kosmicznych jako układy pamięci masowej ponieważ posiadają dużą pojemność, wysoki czas przechowywania danych oraz szybki interfejs [82, 84]. Nawiązując do eksperymentu [85], pamięć NOR Flash jest o wiele bardziej podatna na efekty SEU i SEFI niż pamięć typu NAND Flash. Może być to wytłumaczone względną prostotą układu sterowania pamięcią NOR w porównaniu do pamięci NAND.

Krytyczny poziom TID jest bezpośrednio związany ze zmniejszeniem zdolności retencji bramki pływającej spowodowanej zderzeniami z ciężkimi jonami [84].

Zazwyczaj SEFI powoduje błędy na dużym obszarze pamięci podczas procesu zapisy lub odczytu. Niektóre błędy SEFI mogą być skorygowane przez ponowny odczyt danych z pamięci, inne natomiast wymagają resetu zasilania lub nawet ponownego zapisu.

2.5.3. Mikroprocesory

Dla komercyjnych mikrokontrolerów, próg SEU i SEFI określony jest na poziomie od 0.2 do 9 MeV*cm²/mg [86]. Taka niska wartość powoduje iż, negatywne efekty związane z przebywaniem w środowisku radiacyjnym, mogą pojawić się od kilku razy na dziennie do jednego wystąpienia na rok.

Single Event Upset jest przyczyną błędów na liniach danych mikrokontrolera (data error) a Single Event Functional Interrupt skutkuje zawieszaniem się układu [87]. MPU składa się między innymi z rejestrów General-Purpose Register (GPR) i z Special-Purpose Register (SPR). Konsekwencje uszkodzenia rejestrów są różne i zależą od ich typu. SEU może powodować błędy w działaniu Program Counter (PC) co skutkuje nieprawidłową kolejnością wykonywania instrukcji. Z kolei SEU w GPR prowadzi do uszkodzenia przechowywanych danych i błędne działanie wykonywanego algorytmu.

Chociaż prawdopodobieństwo usterki w skutek wystąpienia SEE w MPU jest mała w porównaniu z np. pamięcią RAM, powstało wiele technik ograniczających błędy ze względu na duże znaczenie mikroprocesora w systemach sterowania satelitów. Ponieważ większość obszaru MPU zajmuje pamięć podręczna, powinny być zastosowane odpowiednie metody zapobiegania błędem danych (np. ECC).

2.5.4. Układy FPGA

FPGA są to reprogramowalne przez użytkownika układy scalone składające się z tysięcy lub nawet milionów bloków logicznych [71]. Każdy blok może wykonywać dowolną funkcję logiczną. Łatwa rekonfiguracja, wysoka wydajność, niska cena i relatywnie niski pobór mocy są poszukiwanymi właściwościami niezbędnymi do opracowania innowacyjnych systemów kosmicznych [88].

Łatwa rekonfiguracja pozwala inżynierom na opracowanie optymalnej konfiguracji a nawet, umożliwia wprowadzenie zmian w systemie już podczas trwania misji kosmicznej [71, 89]. Ponadto układ FPGA zwiększa elastyczność i adaptacyjność komputera pokładowego poprzez wykonywanie skomplikowanych algorytmów oraz umożliwienia podłączenia szerokiej gamy urządzeń peryferyjnych.

Mechanizm szybkiej i łatwej rekonfiguracji układu programowanego oparty jest na komórkach pamięci ulotnej (SRAM) bądź nieulotnej (Flash) w których przechowywane są dane konfiguracyjne. Wyróżniamy także układy FPGA które mogą być programowane tylko raz (Anti-fuse FPGA). Te dwa sposoby konfiguracji dzielą układy FPGA na trzy grupy:

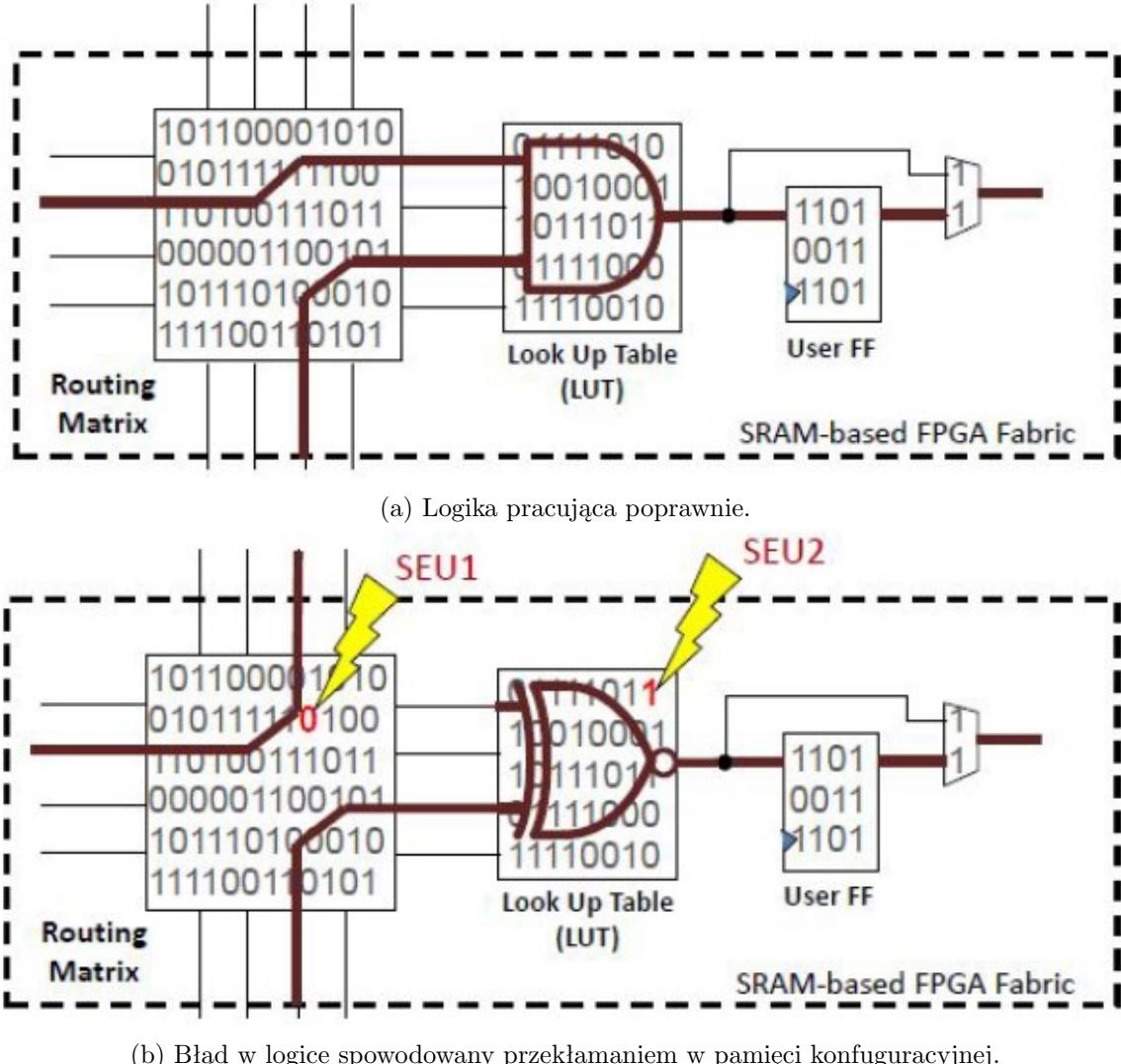
- bazujące na pamięci SRAM,
- bazujące na pamięci Flash,
- Anti-fuse FPGA.

SRAM FPGA

Układy FPGA oparte na ulotnej pamięci konfiguracyjnej zapewniają wysoką szybkość rekonfiguracji i nieograniczoną ilość cykli konfiguracyjnych [90]. Z drugiej strony, taka pamięć

jest podatna na efekty SEU, SET i SEFI [88, 89]. Kiedy w pamięci konfiguracyjnej wystąpi SEU, spowoduje zmianę konfiguracji logiki (rys. 2.11) [91]. Ponadto SET może przekształcić się w stały efekt SEU jeśli zmieni dane konfiguracyjne [90]. Zmiany w danych konfiguracyjnych mogą doprowadzić do SEFI w całym układzie [88, 89]. Ponieważ, zarówno pamięć użytkownika jak i pamięć konfiguracyjna jest podatna na efekty SEU i SET, cały komputer pokładowy oparty na układzie FPGA będzie bardzo wrażliwy na promieniowanie.

Jednakże układy programowalne oparte na pamięci SRAM wytrzymują stosunkowo dużą dawkę TID do 100 – 200 krad [89]. Nowoczesny układ FPGA Virtex-5 wykonany w technologii CMOS 65 nm ma wartość TID równą 500 krad [62]. Popularne i niedrogie układy programowalne firmy Altera z serii Cyclone posiadają TID na poziomie 1 Mrad [77].



Rysunek 2.11: Efekt SEU w układzie FPGA opartym na pamięci konfiguracyjnej SRAM [78]

Flash FPGA

Układy FPGA bazujące na nieulotnej pamięci konfiguracyjnej typu Flash są mniej wrażliwe na SEU ponieważ bramka płynąca stosunkowo dobrze przeciwstawia się radiacji [88, 92]. Można przyjąć iż pamięć Flash jest odporna na SEE i tylko elementy logiczne układy programowalnego będą czułe na promieniowanie [61].

Bramki płynące (FG) bezpośrednio kontrolują programowalne połączenia wewnętrzne

i bloki logiczne w układzie FPGA [93]. Przełącznik oparty na FG składa się z dwóch tranzystorów NMOS a ich napięcie progowe określone jest przez zgromadzony ładunek [94]. Wyniki badań wykazały, iż indukowany ładunek może spowodować SET który czasowo zmieni prawidłową realizację połączeń wewnętrz układy FPGA [93]. Ponieważ każde programowalne połączenie jest potencjalnie wrażliwe na SET, niewykluczona jest tymczasowa zmiana w strukturze połączeń, wdrożonej funkcji logicznej a nawet zmiana stanu sygnału wyjściowego. W konsekwencji, Single Event Transition może doprowadzić do zmiany danych w komórkach pamięci [95]. Jednakże, zapisana konfiguracji układu FPGA w pamięci Flash nie ulegnie trwałemu uszkodzeniu poprzez efekty związane z promieniowaniem [88].

TID może prowadzić do różnic napięć progowych tranzystorów w komórce pamięci Flash co w konsekwencji blokuje jej programowalność [93, 94]. Jednak najbardziej szkodliwym efektem TID jest wadliwe działanie procesu programowania i kasowania układu spowodowane wrażliwością pompy ładunku na promieniowanie [84].

Podsumowując, układ programowalny oparty o pamięć konfiguracyjną typu Flash wykazuje bardzo dużą odporność na TID o wartości poniżej 20 krad(Si). Dla TID z zakresu od 20 krad(Si) do 40 krad (Si) powoli zaczynają wzrastać opóźnienia czasów propagacji i napięcia progowe tranzystorów. Większość układów doświadcza błędów funkcjonalnych dla TID powyżej 40 krad(Si) [96].

Anti-fuse FPGA

W przeciwieństwie do dwóch poprzednich typów układów programowalnych, Anti-fuse FPGA można programować tylko raz. W rezultacie, pamięć konfiguracyjna nie może zostać utracona pod wpływem działania efektów związanych z promieniowaniem [89].

Tolerancja na TID przykładowych Antifuse FPGA firmy Actel wynosi od 100 krad(Si) do 300 krad(Si) [79]. Czułość na promieniowanie układów programowalnych typu Antifuse określona jest tylko przez wrażliwość elementów logicznych.

Choć zarówno układy FPGA Antifuse jak i te oparte na pamięci Flash są podatne na SET to bardzo wysoka cena tych pierwszych sprawia że są one niedostępne dla małych satelitów.

2.6. Techniki zwiększające odporność na promieniowanie

Komputer pokładowy musi przeciwwieścić się szkodliwym efektom wywoływanym przez promieniowanie. Stosowane obecnie komputery, kwalifikowane do misji kosmicznych, bazują na niestandardowych układach w których zastosowano odpowiednie techniki zwiększające odporność na promieniowanie (np. zmodyfikowany proces produkcji). Jednak to podejście skutkuje obniżeniem wydajności, zwiększeniem zapotrzebowania na energię i przede wszystkim, znacząco podwyższa koszty [66]. Inne komputery pokładowe bazują na układach które były wcześniej wykorzystywane w misjach kosmicznych i dobrze zniosły przebywanie w środowisku radiacyjnym. Jednakże takie układy zazwyczaj wymagają większego zapotrzebowania na energię i są mniej wydajne niż nowoczesne elementy dostępne komercyjnie.

Wszystkie techniki redukcji szkodliwych efektów związanych z promieniowaniem można podzielić na 3 grupy [97]:

- rozwiązania polegające na zwiększaniu odporności układów scalonych,
- techniki związane z projektowaniem,

- techniki implementacji oprogramowania odpornego na błędy (ang. Software Implemented Fault Tolerance)[99]

Ponieważ komponenty odporne na promieniowanie są niedostępne dla małych satelitów, w rozdziale tym zostanie przedstawiony przegląd sprzętowych i programowych technik ograniczania wpływu promieniowania. Najbardziej popularne rozwiązania (na przykład redundancja sprzętowa lub czasowa) mogą być stosowane na różnych poziomach architektury (na przykład na poziomie komponentu lub na poziomie konfiguracji układu FPGA). Rozwiązania systemowe zwiększające odporność na błędy zostaną omówione w podrozdziale 3.6.1, natomiast podrozdział 3.6.2 skupi się na rozwiązaniach w obrębie komponentu.

2.6.1. Rozwiązania stosowane na poziomie architektury

Ekranowanie

Ekranowanie jest techniką zwiększającą odporność na promieniowania poprzez umieszczenie narażonego elementu zazwyczaj w metalowej osłonie bezpośrednio chroniącej go przed naładowanymi cząstками. Jedną z głównych wad takiego rozwiązania jest masa ekranu która znacząco zwiększa koszty wyniesienia w przestrzeń kosmiczną [98]. Ekranowanie może być wykonane na poziomie struktury mechanicznej satelity, na poziomie modułu lub w obrębie komponentu. Ekran może być skutecznym środkiem redukującym TID jednakże nie gwarantuje ochrony przed SEE [87].

Watchdog

Układ Watchdog może dostarczyć sygnał reset oraz wymusić cykl włączenia linii zasilających pojedynczemu komponentowi, podsystemowi lub całemu modułowi w przypadku wykrycia nieprawidłowej pracy lub zawieszenia się. Cykl zasilania i sygnał resetu są sprawdzonymi sposobami wychodzenia z SEFI [66, 100]. Wadą tych technik jest to, iż mikroprocesor, podsystem czy cały system nie są w stanie pracować podczas resetu.

Windowing

Częste błędy podczas komunikacji lub niemożność nawiązania poprawnej transmisji z danym komponentem może świadczyć o błędnym działaniu całego systemu. Układ nadzędny może zakończyć komunikację z błędnie działającym urządzeniem w celu zaprzestania marnowania zasobów [87]. Inną formą techniki Windowing [102] jest sytuacje w której zadanie realizowane przez RTOS [197] uznawane jest za wadliwe gdy jego wykonanie zajmuje więcej czasu niż przewidziano. Generalnie, technika Windowing zakłada iż, zadanie uznaje się za błędne jeśli jego wykonanie wykracza poza zdefiniowane ramy czasowe lub jeżeli zużywa zasoby sprzętowe których normalnie nie powinno.

Zapobieganie skutkom SEL

Szybkie wykrywanie efektu SEL jest konieczne do zapobiegania krytycznym uszkodzeniom bramek logicznych. Single Event Latch-up może być wychwycone poprzez monitorowanie poboru prądu układu scalonego [62]. W przypadku wykrycia zbyt wysokiego poboru energii, zasilanie danego komponentu lub podsystemu powinno zostać wyłączone a następnie włączone [87].

Jedną z przeszkód skutecznej detekcji SEL jest duża zależność poboru prądu przez logikę CMOS od aktualnego zapotrzebowania na przetwarzanie danych. Rozwiązaniem może być uruchomienie w układzie przetwarzania znanego algorytmu i próbkowanie ilości zużywanej energii.

N-modułowa redundancja

N-modułowa redundancja może być stosowana jako technika redukcji szkodliwych efektów w obrębie rdzenia procesora, pamięci, soft-procesora w FPGA, a nawet całych podsystemów. W tej metodzie należy wziąć pod uwagę dodatkowy narzut na wykorzystywany sprzęt. TMR (ang. Triple Modular Redundancy) jest najpopularniejszym typem redundancji. W większości przypadków, TMR zawiera mechanizm „głosowania” który decyduje o wyborze wyniku i zakłada iż, w danym czasie tylko jeden z modułów popełnił błąd [103, 56, 104].

Dwu-modułowa redundancja może być wykorzystana do określenia prawidłowego wyniku wykonywanej instrukcji. W tym przypadku wykonywanie instrukcji musi zostać powtórzone [100, 101].

Redundancja czasowa

Redundancja czasowa polega na ciągłym wykonywaniu tej samej operacji aż uzyska się poprawny wynik [66]. Wadą tej metody jest długotrwałe zajęcie zasobów w celu upewnienia się iż wynik nie został uszkodzony przez SEE. SET może zostać wykryty dzięki redundancji czasowej.

2.6.2. Rozwiązanie stosowane na poziomie komponentu

Mikroprocesory

Ochrona procesora przed skutkami SEFI może zostać osiągnięta przez reset układu. Sygnał resetujący może pochodzić z zewnętrznego źródła lub być wygenerowany przez wewnętrzny układ Watchdog [87].

Błędy w pamięci RAM usuwane są dzięki systemowi EDAC (ang. Error-Detection And Correction) który je identyfikuje i koryguje.

Techniki stosowane na poziomie oprogramowania mikroprocesora

Techniki redukcji skutków promieniowania związane z oprogramowaniem są nieodzowną częścią systemów satelitarnych. Implementowane są do wykrywania nieprawidłowości i ponownej inicjalizacji po wystąpieniu krytycznych błędów.

Wielokrotna i niezależna implementacja tych samych funkcji umożliwia wychwycić nieprawidłowości w realizowanych algorytmach [105]. Redundancja czasowa wykonywanych instrukcji, procedur i algorytmów pozwala wykryć i naprawić błędy związane z SEU w trakcie wykonywania programu [106].

Krytyczne dane niezbędne do przywrócenia poprzedniego stanu pracy programu po restartie powinny być przechowywane w pamięci nieulotnej, najlepiej w kilku kopiacach, aby zapewnić tak zwany „worm start”.

Aktualizacja oprogramowania

Reprogramowalność układów FPGA oraz zdolność aktualizacji oprogramowania mikrokontrolera można uznać za kluczową funkcjonalność pozwalającą na zwiększenie czasu życia satelitów przeznaczonych do długotrwałych misji [107].

Główny problemem w procesie aktualizacji oprogramowania jest całkowita utrata funkcjonalności systemu na czas zmiany programu oraz niewykluczone jest doprowadzenie do całkowej i nieodwracalnej awarii urządzenia w przypadku wystąpienia nieprzewidzianych błędów [107].

Nowa konfiguracja układu FPGA czy oprogramowanie mikrokontrolera musi zostać przesłane ze stacji naziemnej. Binarny plik konfiguracyjny nierzadko zajmuje nawet kilka meabajtów, stąd proces jego transmisji może zająć kilka okresów orbitalnych.

Aktualizacja wbudowanego programu stwarza duże możliwości dostosowania, optymalizacji czy naprawienia błędów oprogramowania urządzeń satelitarnych.

Integralność danych w pamięciach

Technika „memory scrubbing” polega na cyklicznym odczytywaniu komórek pamięci i w przypadku wykrycia uszkodzenia spowodowanego efektem SEU, naprawianiu wadliwych danych. Detekcja i korekcja uszkodzonych komórek może zostać zrealizowana tylko wtedy gdy alokowane są dodatkowe bajty [108]. Jednym z najpopularniejszych kodów wykrywania i naprawiania błędów jest kod Humminga [56].

Układy FPGA

Bazujące na pamięci SRAM

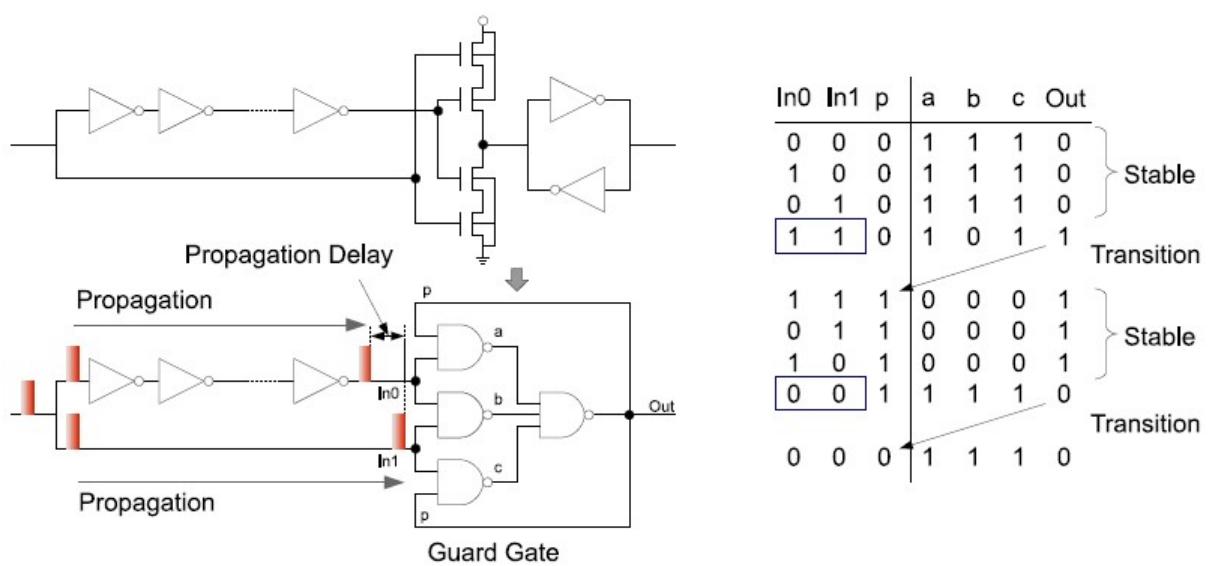
Układy FPGA bazujące na pamięci SRAM wykazują dostateczną tolerancję na TID. W celu przeciwdziałania efektom SEU wykorzystuje się sprzętową i czasową redundancję powielając projektowane struktury logiczne. Często stosowana jest technika potrójnego zwielokrotnienia (TMR), jednak wymaga ona ponad trzykrotnie większych zasobów sprzętowych gdyż należy uwzględnić mechanizm decydujący [109, 89, 100].

Z powodu wrażliwości układów SRAM FPGA na SEU, komputer pokładowy powinien być wyposażony w system „memory scrubbing” dla pamięci konfiguracyjnej. Czasami stosuje się dodatkowy, mały układ FPGA który realizuje funkcję „memory scrubbing” dla pamięci konfiguracyjnej większego układu programowalnego [89].

Bazujące na pamięci Flash

Pamięć konfiguracyjna typu Flash układu FPGA jest odporna na SEE. Jednak logika kombinacyjna jest podatna na SET i SEU.

Wrażliwość na efekty SET i SEU może zostać złagodzona poprzez redundancję czasową dostosowaną do szerokości indukowanego impulsu [54]. Ponieważ SET jest pojedynczym impulsem o bardzo krótkim czasie trwania który propaguje się w logice, może on zostać przefiltrowany przez elementy opóźniające. Filtr SET (rys. 2.12) może składać się z bloków opóźniających (np. połączonych szeregowo inwerterów) i elementu „Guard Gate”. Jednak głównym warunkiem poprawnej filtracji SET jest to, aby opóźnienie był dłuższe niż czas trwania impulsu [62].



Rysunek 2.12: Struktura logiczna przykładowego filtra SET [62]

Rozdział 3

Geneza, cel i założenia pracy

Oferowane na rynku komputery pokładowe do satelitów typu CubeSat nie są wyposażone w moduł nawigacyjny. Wymagane jest dołączenie zewnętrznego systemu wyznaczania orientacji co zajmuje cenne miejsce

Ponadto, komercyjne urządzenia nie są zbudowane w oparciu o nowoczesne, wydajne mikrokontrolery przystosowane do aplikacji lotniczych i kosmicznych. W tych układach wysoki nacisk postawiono na zapewnienie bezpieczeństwa i poprawności wykonywanych instrukcji.

W związku z powyższym, a także z powodu dynamicznie rozwijającego się rynku nano satelitów, przystąpiono do opracowania innowacyjnego komputera pokładowego bazującego na mikrokontrolerze o zwiększym bezpieczeństwie pracy. Dodatkowo, budowane urządzenie wyposażono w zintegrowany moduł nawigacyjny.

Celem niniejszej pracy magisterskiej była analiza stosowanych rozwiązań, opracowanie architektury zwiększającej odporność systemu na skutki promieniowania jonizującego, a następnie zaprojektowanie obwodu drukowanego, wykonanie prototypu komputera pokładowego oraz napisanie podstawowego oprogramowania diagnostycznego.

Ponadto, przeprowadzono badania czasu reakcji trzech różnych koncepcji zabezpieczeń systemu zasilania na wypadek wystąpienia przeciążenia prądowego.

System komputera pokładowego do satelity typu CubeSat zaprojektowano w oparciu o następujące założenia:

- praca jako moduł komputera pokładowego z systemem nawigacyjnym lub jako samodzielny system nawigacyjny,
- wykorzystanie łatwo dostępnych, standardowych, układów scalonych,
- architektura oparta o dwa niezależne główne mikrokontrolery,
- możliwość podłączenia dwóch modułów kamer.

Rozdział 4

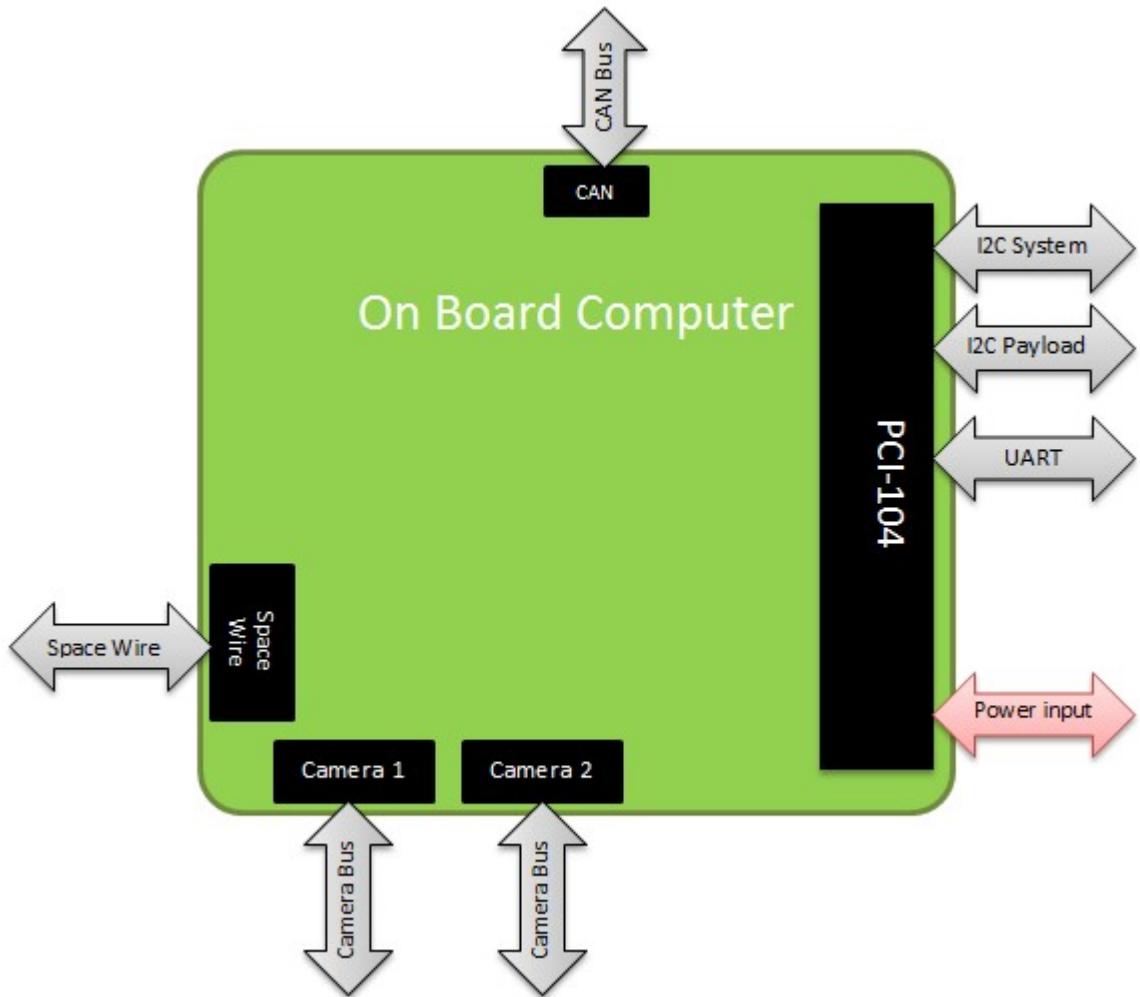
Koncepcja konstrukcji modułu komputera pokładowego

W niniejszym rozdziale opisano koncepcję konstrukcji komputera pokładowego do satelity typu CubeSat. Sporządzono schemat blokowy urządzenia oraz wymieniono interfejsy zewnętrzne. Opisano podstawowe zadania wykonywane przez projektowany system a także scharakteryzowano główne bloki funkcjonalne. Następnie zestawiono oprogramowanie niezbędne do zaprojektowania komputera.

4.1. *Komputer pokładowy*

Komputer pokładowy jest integralną częścią większego systemu na który składają się wszystkie moduły satelity. Komunikuje się on z nimi poprzez szereg interfejsów zewnętrznych. Rysunek 4.1 obrazuje rozmieszczenie interfejsów zewnętrznych. Interfejsy te wymieniono i opisano poniżej:

- **I2C System** - szeregowa, dwukierunkowa magistrala I2C przeznaczona do sterowania modułem zarządzającym zasilaniem a także do obsługi modułu komunikacyjnego,
- **I2C Payload** - szeregowa, dwukierunkowa magistrala I2C służąca do zarządzania ładunkiem satelity,
- **UART** - szeregowa, dwukierunkowa magistrala wykorzystana do debugowania uruchamianego oprogramowania. Możliwe jest wykorzystanie tego interfejsu do obsługi modułów satelity.
- **Power input** - wejście zasilania, przeznaczone jest dostarczenia odpowiedniego napięcia zasilającego komputer pokładowy,
- **CAN Bus** - szeregowa magistrala komunikacyjna, przeznaczona jest do przesyłania danych z prędkością do 1 Mbps,
- **SpaceWire** - interfejs wykorzystywany w urządzeniach kosmicznych, przeznaczony jest do szybkiego przesyłania danych pomiędzy modułami satelity. Maksymalna szybkość transmisji to 400 Mbps,
- **Camera 1** - interfejs modułu kamery przeznaczonej do nawigacji w przestrzeni kosmicznej,
- **Camera 2** - interfejs modułu kamery służącej do robienia zdjęć i nagrywania filmów.



Rysunek 4.1: Rozkład interfejsów zewnętrznych *modulu komputera pokładowego*

4.2. Podstawowe zadania *komputera pokładowego*

Podczas projektowania architektury systemu wzięto pod uwagę podstawowe zadania jakie ma realizować komputer pokładowy. Są to:

- obsługa modułów pokładowych satelity,
- nawigacja w kosmosie,
- autodiagnostyka,
- obsługa misji.

Zadania te zostały scharakteryzowane poniżej.

4.2.1. Obsługa modułów pokładowych satelity

W zależności od rozmiarów, nano-satelita może być wyposażony w od kilku do kilkunastu modułów pokładowych. Komputer pokładowy musi mieć możliwość dwukierunkowej komunikacji z każdym modułem. Do urządzeń pokładowych przesyłane będą dane konfiguracyjne i rozkazy a pobierany będzie status i dane pomiarowe. Do komunikacji wykorzystane będą, w zależności od modułu, różne interfejsy szeregowe. Większość urządzeń pokładowych stosowanych w CubeSat'ach komunikuje się poprzez magistralę I2C. Komputer dysponuje dwoma niezależnymi szynami I2C Payload i System. Magistrala Payload może być wykorzystana do zarządzania ładunkiem satelity który jest przedmiotem eksperymentu. Na przykład rozkładany żagiel deorbitujący [110] lub kamera obserwująca Ziemię. Natomiast I2C System

obsługuje moduł komunikacyjny oraz moduł zasilania które są jednymi z najważniejszych systemów CubeSat'a.

Komputer pokładowy powinien posiadać także interfejs zdolny do przesyłania dużej ilości danych, takich jak obraz z kamery wysokiej rozdzielczości lub szybka komunikacja z modułem nadawczo-odbiorczym. Dlatego też wyposażono go w magistralę CAN o przepustowości do 1 Mbps. Dzięki interfejsowi CAN możliwa jest przyszła implementacja protokołu *CubeSat Space Protocol* specjalnie zaprojektowanego do komunikacji w obrębie CubeSat'a.

Ponadto, komputer pokładowy posiada możliwość implementacji sieci komunikacyjnej *SpaceWire* wykorzystywanej na szeroką skalę w dużych satelitach. Przepustowość tej magistrali sięga 400 Mbps. Więcej informacji na temat *SpaceWire* i *CubeSat Space Protocol* znajduje się w dodatku A.3.

4.2.2. Obsługa misji

Zadaniem każdego satelity jest wykonanie misji szczegółowo zdefiniowanej już na Ziemi na etapie projektowania. Poszczególne zadania które wchodzą w skład misji muszą być wykonane w ścisłe określonym czasie. Odpowiedzialny jest za to komputer pokładowy. Pobiera on zadania ze spisu znajdującego się w wewnętrznej pamięci i kolejno je wykonuje w zaplanowanym czasie. Aktualny czas i datę dostarcza wbudowany zegar czasu rzeczywistego.

W czasie trwania misji zadania do wykonania mogą ulec zmianie. W celu dostosowania misji do nowych warunków komputer pokładowy musi mieć możliwość wykonania dodatkowych zadań pochodzących z centrum obsługi naziemnej. Ponadto, w krytycznych sytuacjach może zaistnieć potrzeba zmiany wewnętrznej listy zadań. W takich sytuacjach obsługa naziemna musi mieć możliwość zdalnej aktualizacji spisu zadań.

4.2.3. Autodiagnostyka

Niezmiernie ważnym elementem warunkującym poprawną pracę systemu jest przeprowadzanie autodiagnostyki. Polega ona na stałym monitorowaniu kluczowych parametrów satelity. Jednym z nich jest wartość pobieranego prądu przez poszczególne podsystemy. Jest to kluczowe z dwóch powodów. Po pierwsze, budżet mocy przypadający na komputer pokładowy jest ścisłe określony i nie może zostać przekroczyony. Może dojść do sytuacji w której wykonywane zadania zużywać będą cały dostępny zasób energii i rozpoczęcie realizacji następnego zadania spowodowałoby przekroczenie dozwolonego limitu mocy. W takie sytuacji główny mikrokontroler musi poczekać aż zwolnią się zasoby energetyczne i dopiero wtedy rozpocząć wykonywanie następnych zadań. Po drugie, stale monitorując pobór prądu przez poszczególne podsystemy, układ nadzorujący jest w stanie wykryć nagły wzrost pobieranej energii. Świadczyć to może o wystąpieniu SEE. W takiej sytuacji należy jak najszybciej wyłączyć zasilanie danej sekcji w celu uniknięcia trwałego uszkodzenia struktur półprzewodnikowych a także w celu zapobiegnięcia przekroczenia dozwolonego budżetu mocy.

Kolejnym parametrem wartym monitorowania jest napięcie na poszczególnych szynach zasilających. Nagły spadek napięcia bądź jego wzrost może świadczyć o uszkodzeniu. obsługa naziemna mając takie informacje może wykryć usterkę i w miarę możliwości, rozpocząć działania prowadzące do jej usunięcia lub zmiany harmonogramu zadań.

W przestrzeni kosmicznej nie ma atmosfery i w związku z tym ciepło wydzielane przez pracujące układy scalone nie jest rozpraszczone. Pomiar temperatury kluczowych elementów

komputera pokładowego stwarza możliwość wykrycia nadmiernego nagrzewania się jego podsystemów. Dzięki tej wiedzy układ nadzorujący będzie mógł podjąć decyzje o wyłączeniu grzejącego się podsystemu bądź o ograniczeniu wykonywanych przez niego zadań.

Pożądaną funkcjonalnością systemu autodiagnostyki jest umożliwienie obsłudze naziemnej zdalnej aktualizacji oprogramowania mikrokontrolerów komputera pokładowego. Dzięki temu inżynierowie odpowiedzialni za satelitę będą mogli całkowicie zmienić charakter misji w chwili gdy system będzie na orbicie. Ponadto w przypadku wystąpienia nieodwracalnej usterki, aktualizacja oprogramowania może okazać się jedynym rozwiązaniem pozwalającym satelicie na kontynuowanie swojej misji.

Satelita znajdujący się na niskiej orbicie okoziemskiej poddawany jest działaniu promieniowania jonizującego. Pomimo zastosowania licznych zabezpieczeń zapobiegającym SEE, nie należy wykluczyć zawieszenia się wszystkich podsystemów komputera pokładowego włącznie z układem nadzorującym. W takim przypadku zdalny reset systemu nadzorującego może przywrócić prawidłową pracę komputera pokładowego.

4.2.4. Nawigowanie w kosmosie

Aby satelita mógł poprawnie wykonywać misję musi wiedzieć gdzie się znajduje. Podsystem komputera pokładowego powinien dostarczyć informacji o położeniu satelity w przestrzeni kosmicznej. Jest to jedno z jego najważniejszych zadań. Wyznaczenie orientacji wymaga użycia znacznie więcej zasobów sprzętowych i większego zapotrzebowania na moc, niż inne zadania realizowane przez komputer pokładowy.

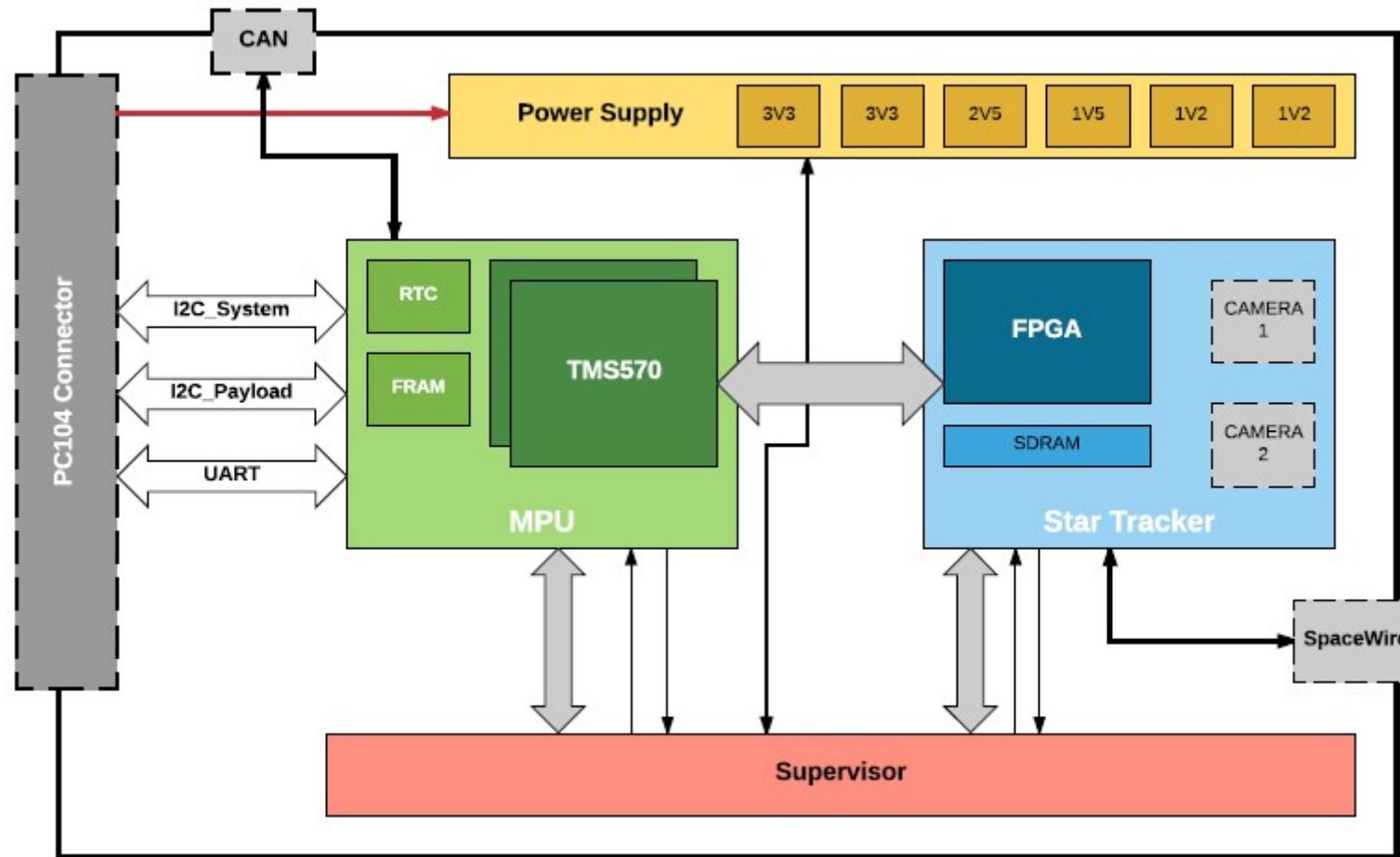
Do wyznaczania położenia satelity w przestrzeni kosmicznej zdecydowano się na wykorzystanie metody z użyciem Star Tracker'a ponieważ jest ona najdokładniejsza. Projektując komputer pokładowy zadbano o odpowiednie interfejsy i zasoby obliczeniowe aby aktualne położenie mogło być dostarczone 2 do 5 razy w przeciągu sekundy. Algorytm wyznaczania orientacji wykorzystuje dane z sensora optycznego oraz mapę gwiazd znajdująca się wewnętrznej pamięci NOR Flash komputera pokładowego.

4.3. Koncepcja konstrukcji

Na rysunku 4.2 przedstawiono schemat blokowy komputera pokładowego do satelity. W skład systemu wchodzą następujące jednostki:

- **Main microcontrollers** - dwa redundantne mikrokontrolery, które:
 - zapewniają komunikację z pozostałymi modułami satelity poprzez interfejsy I2C, UART, CAN,
 - odczytują z zewnętrznej pamięci Flash zaplanowane zadania i kolejno je wykonują,
 - aktualny czas uzyskują z zewnętrznego zegara RTC,
 - komunikują się z układem nadzorującym poprzez interfejs UART,
 - przechowują aktualny status misji i wykonanych zadań w pamięci zewnętrznej FRAM,
 - komunikują się z układem FPGA poprzez interfejs SPI; doczytują wyniki pracy wbudowanych algorytmów i mogą pobrać skompresowany obraz z kamery,
 - posiadają możliwość zmiany konfiguracji wewnętrznej układu FPGA,
- **FPGA** - układ programowalny, który:
 - dokonuje akwizycji danych z dwóch kamer,

- odebrany obraz umieszcza w zewnętrznej pamięci SDRAM,
 - wykonuje algorytm który na podstawie obrazu z jednej z kamer wyznacza orientacji satelity w przestrzeni kosmicznej,
 - obsługuje interfejs SpaceWire,
- **SDRAM** - pamięć typu DDR SDRAM, służy do tymczasowego przechowywania odebranych klatek obrazu z kamer Star Tracker i kamery dodatkowej,
 - **Supervisor** - mikrokontroler który pełni rolę układu nadzorującego. Jego zadania to:
 - wykrywanie zawieszenia się aktywnego mikrokontrolera głównego i w związku z tym odłączenie go od podsystemów komputera pokładowego, aktywowanie redundantnego mikrokontrolera głównego,
 - resetowanie mikrokontrolerów,
 - przekazywanie aktywnemu mikrokontrolerowi informacji o budżecie mocy,
 - pomiar napięcia na liniach zasilających,
 - odbieranie danych pomiarowych z układów mierzących pobór prądu,
 - w przypadku wykrycia przekroczenia dopuszczalnego napięcia lub poboru prądu, wyłączenie odpowiedniej przetwornicy,
 - włączanie lub wyłączenie modułów kamer,
 - pomiar temperatury w kluczowych miejscach płytka PCB,
 - przejmowania kontroli nad magistralą I2C podłączoną do modułu komunikacji w celu pobrania aktualnego oprogramowania wbudowanego przeznaczonego dla głównych mikrokontrolerów; aktualizacja oprogramowania układów TMS570,
 - komunikacja z układem FPGA poprzez interfejs UART,
 - tworzenie raportu o stanie komputera pokładowego,
 - **Power Supply** - podsystem odpowiadający za dostarczenie szeregu stabilnych napięć zasilających elektronikę komputera pokładowego, wyposażony został w specjalne układy scalone służące do pomiaru prądu na poszczególnych liniach zasilających,
 - **Camera 1** - moduł sensora obrazu Star Tracker wraz z obiektywem, dane z tej kamery są wykorzystywane przez algorytm wyznaczania orientacji,
 - **Camera 2** - dodatkowy moduł sensora obrazu przeznaczony do robienia zdjęć i nagrywania krótkich filmów,
 - **FRAM** - nieulotna pamięć ferroelektryczna przechowująca wrażliwe dane oprogramowania mikrokontrolerów,
 - **RTC** - zegar czasu rzeczywistego dostarczający aktualną datę i czas.



Rysunek 4.2: Schemat blokowy *modulu komputera pokładowego*

4.4. Plan realizacji

Dokładnie analizując założenia związane z projektem, przystąpiono do sporządzenia planu prac nad komputerem pokładowym do satelity typu CubeSat. Wyznaczono zadania:

- opracowanie architektury sprzętowej o zwiększonej odporności na promieniowanie jonizujące,
- określenie niezbędnych interfejsów zewnętrznych oraz tych mogących znaleźć zastosowanie w przyszłości,
- wybór głównych układów elektronicznych tj. FPGA, SDRAM, mikrokontrolery, pamięci nieulotne,
- wybór czujników obrazu,
- oszacowanie zapotrzebowania na energię elektryczną oraz wybór odpowiednich układów,
- zaprojektowanie sieci kondensatorów odsprzęgających dla układu FPGA oraz głównych mikrokontrolerów,
- symulacja integralności sygnałowej pamięci SDRAM,
- opracowanie testowej konfiguracji układu FPGA potwierdzającej syntezowalność projektu dla konkretnego układu programowanego,
- zaprojektowanie wielowarstwowego obwodu drukowanego z uwzględnieniem impedancji ścieżek,
- napisanie oprogramowania testującego mikrokontrolery,
- sporządzenie konfiguracji dla układu FPGA testującej pamięć SDRAM.

Wszystkie schematy elektryczne oraz projekt wielowarstwowego obwodu drukowanego zostały sporządzone w programie Altium Designer 14.1 [111]. Symulacje integralności sygnałowej wykonano w środowisku HyperLynx SI/PI [112] firmy Mentor Graphics. Oprogramowanie mikrokontrolerów głównych napisano w programie Code Composer Studio [114] firmy Texas Instruments [115]. Program sterujący układem nadzorującym przygotowano w aplikacji Atmel Studio 7 [116] firmy Atmel [117]. Konfiguracje układu programowanego sporządzono w programie Quartus II 13.0 [118] firmy Altera [119].

4.5. Podsumowanie

W projekcie komputera pokładowego najważniejsze było takie zaprojektowanie architektury i oprogramowania, aby system przeciwstawił się szkodliwym efektom spowodowanym oddziaływaniem promieniowania jonizującego. Warto także podkreślić, iż moduł może pracować jako samodzielna jednostka obliczeniowa oraz jako podrzędny system nawigacyjny, a także pełnić obie te funkcje jednocześnie. Dzięki zastosowaniu układu programowanego o znacznych zasobach sprzętowych, zaprojektowana platforma umożliwia implementacje różnych algorytmów użytkowych na pokładzie satelity typu CubeSat.

Rozdział 5

Realizacja modułu komputera pokładowego

W skład modułu komputera pokładowego wchodzą następujące podsystemy: *główna jednostka obliczeniowa, układ nadzorujący, sekcja zasilania, Star Tracker oraz interfejsów zewnętrznych*. W niniejszym rozdziale przedstawiono realizacje komputera pokładowego wykonanego w ramach pracy magisterskiej. Funkcje systemowe urządzenia przedstawiono poniżej:

- obsługa zewnętrznych magistral I2C oraz interfejsu UART,
- wykrywanie zawieszenia się głównego mikrokontrolera oraz resetowanie go w razie konieczności,
- pomiar prądu i napięcia na liniach zasilających,
- obsługa pamięci FRAM,
- obsługa pamięci SDRAM,
- akwizycja ramki z czujnika obrazu.

W celu zmniejszenia rozmiarów gotowego urządzenia, jednostkę komputera pokładowego i moduł Star Tracker zintegrowano na jednej płytce PCB. W zależności od aplikacji, istnieje możliwość selektywnego wykorzystania podsystemów. Dzięki temu urządzenie jest w stanie pracować tylko jako Star Tracker lub tylko jako komputer pokładowy.

5.1. Podsystemy komputera pokładowego

Jednym z głównych założeń satelitów typu CubeSat jest to aby wszystkie moduły wykonane były z komercyjnie dostępnych elementów elektronicznych. Dlatego też konstruując podsystemy komputera pokładowego wybrano nowoczesne układy oferowane przez globalnych dostawców. Warunkiem jaki postawiono wyszukiwanym elementom było udokumentowane badanie w środowisku radiacyjnym. Niestety, tylko niewielka ilość komponentów elektronicznych została odpowiednio przebadana ze względu na stosunkowo wysokie koszty takiej analizy. W związku z tym, kolejnym kryterium doboru układów było potwierdzone wykorzystanie na pokładzie innych CubeSat’ów. Korzystano także z oficjalnej listy elementów o rozszerzonych parametrach (ang. enhanced products)[120].

Układ	Funkcja	TID [krad]	LETh	Wykorzystanie na CubeSat'cie	Enhanced Products	Źródło
S	TMS570LS3137 [134]	Microcontroller	-	-	TAK	[120]
	MT46V64M8 [135]	SDRAM	100	40	-	[128]
	EP3C25Q240C8 [136]	FPGA	1000	26.6	-	[77]
	FM24CL04B [137]	FeRAM	1000	75	-	[127]
	N25Q128A [138]	NOR Flash	-	-	TAK	-
	TPS5430 [139]	Power	15-20	-	-	TAK
	MAX1951 [140]	Power	-	-	TAK	-
	TPS62160 [141]	Power	-	-	-	TAK
	Atmega128 [142]	Microcontroller	20	-	TAK	-
	SN74AVC8T245 [143]	Bus transceiver	30	-	-	TAK
	SN74CB3T3257 [144]	Multiplekser	-	-	-	TAK
	PCA9517 [145]	I2C switch	-	-	TAK	-
	DS3231 [146]	RTC	-	-	TAK	-
	INA209 [147]	Power monitor	-	-	TAK	-
	INA219 [148]	Power monitor	-	-	TAK	-
	AT45DB161B [149]	Flash	14	-	TAK	-
	TPS2553 [156]	Current limited switch	-	-	TAK	[157, 120]
	MT9D111 [150]	Image sensor	-	-	TAK	-

Tablica 5.1: Obsługiwane tryby konfiguracji układu FPGA (- brak danych)

Niestety, nie wszystkie układy spełniają założone wymagania co do LETH i TID. Ponadto nie w każdym przypadku udało się znaleźć publikacje naukową świadczącą o przeprowadzeniu odpowiedniego badania. W związku z tym, zastosowano dodatkowe rozwiązania konstrukcyjne minimalizujące szkodliwy wpływ efektów powodowanych przez promieniowanie jonizujące.

Wszystkie powyższe układy scalone posiadają obudowę inną niż BGA ponieważ stosowanie tego typu obudów w aplikacjach kosmicznych jest ryzykowne ze względu na wysokie przeciążenia i odkształcenia termiczne. Odkształcające się płytki PCB może doprowadzić do oderwania się kulki układu scalonego od padu lutowniczego.

Poniższe podrozdziały opisują rozwiązania sprzętowe podsystemów komputera pokładowego. Wszystkie schematy elektryczne znajdują się w dodatku A.1.

5.1.1. Główna jednostka obliczeniowa

Jako główny mikrokontroler komputera pokładowego użyto nowoczesnego układu TMS570LS3137 firmy Texas Instruments. W tabeli 5.2 wymieniono zasoby sprzętowe mikrokontrolera. Dzięki wydajnemu rdzeniowi i dużej ilości pamięci Flash i RAM, umożliwia implementacje zaawansowanych algorytmów użytkowych na satelicie CubeSat. Procesor ten jest wyposażony w szereg rozwiązań poprawiających bezpieczeństwo i niezawodność wykonywanego programu. Najważniejsze z nich to:

- dwa redundantne rdzenie Cortex-R4F pracujące w trybie „lockstep” (oba rdzenie wykonują te same instrukcje, wynik jest porównywany i jeśli się różni to instrukcja jest powtarzana),
- pamięci Flash, RAM i EEPROM wyposażone są w ECC,
- system Memory Protection,
- auto-diagnostyka procesora,
- system pozwalający na szybki test pamięci RAM podczas startu,
- wbudowany monitor zegara i napięcia,
- moduł sygnalizacji błędu wyposażony w zewnętrzny pin.

Ponadto, posiada on certyfikaty *ISO 26262 ASIL D* [151] i *IEC 61508 SIL 3* [152]. Wymienione rozwiązania i certyfikaty są bardzo istotne z punktu widzenia projektu ponieważ będą minimalizować szkodliwe efekty wywołane promieniowaniem jonizującym i warunkami środowiskowymi panującymi w przestrzeni kosmicznej.

Zasób	Ilość
Pamięć Flash	3 MB
Pamięć RAM	256 kB
Częstotliwość	160 Mhz
FlexRay	1
CAN Bus	3
SPI	3
I2C	1
UART	2
Kanały ADC	24

Tablica 5.2: Zasoby wykorzystanego układu TMS570LS3137 firmy Texas Instruments

Na rysunku 5.1 pokazano schemat blokowy głównej jednostki obliczeniowej komputera po-

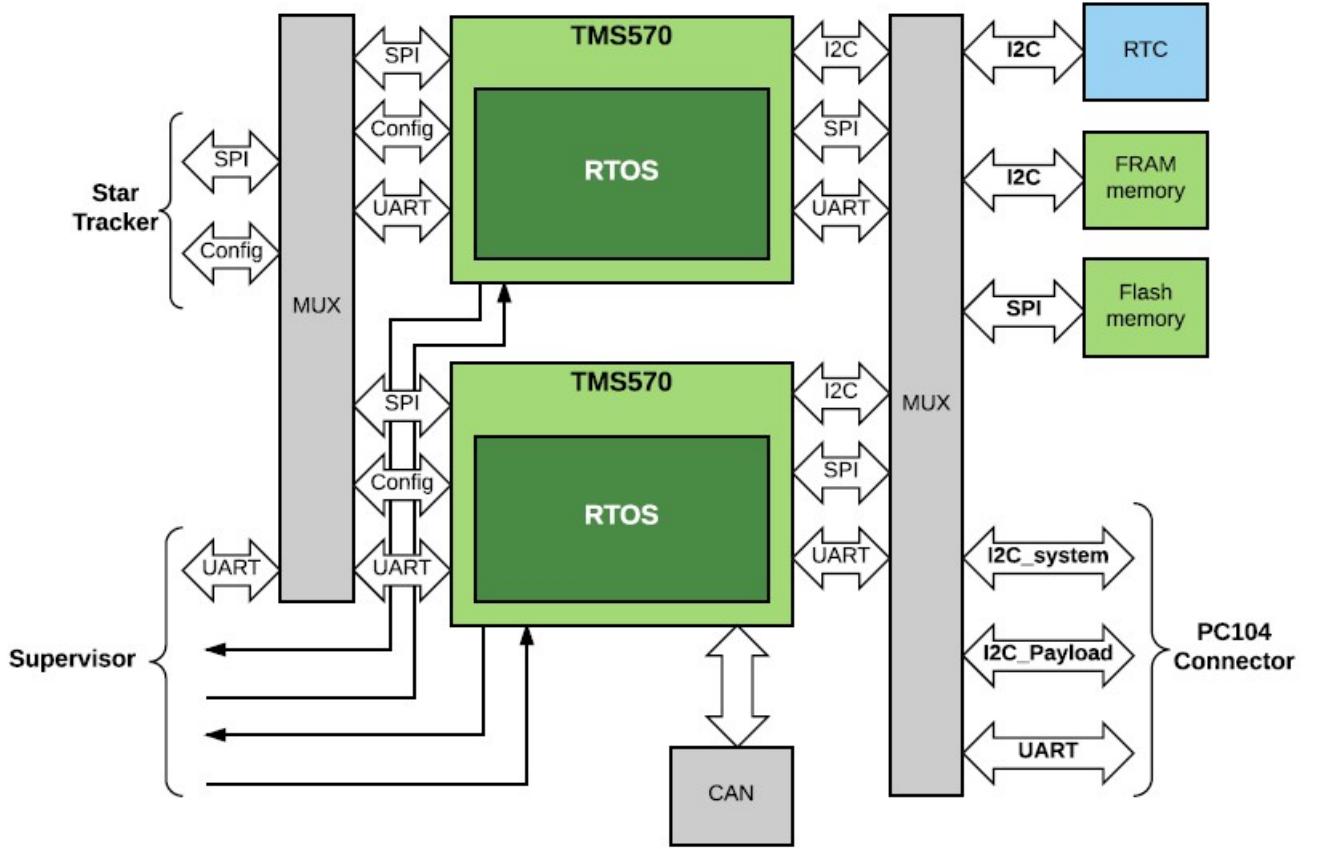
kładowego. Składa się on z dwóch redundantnych mikrokontrolerów TMS570LS3137. Każdy z nich może być zaprogramowany takim samym oprogramowaniem wbudowanym bądź też wgrane kody maszynowe mogą być różne. Taka architektura sprawia iż, wybierając aktywny mikrokontroler użytkownik może kompletnie zmieniać wykonywane funkcje i algorytmy. W przypadku takiego samego oprogramowania zyskuje się dodatkową redundancję w postaci zapasowego procesora.

W tabeli 5.3 widnieje lista sygnałów wejściowych i wyjściowych wykorzystanych do komunikacji i sterowania podsystemami komputera pokładowego. Pamięć NOR Flash podłączono do obu układów TMS570 poprzez interfejs SPI.

Nazwa sygnałów lub grupy sygnałów	Opis	Podsystem
SPI_NOR_FLASH	Interfejs SPI do pamięci NOR Flash	Główna jednostka obliczeniowa
SPI MCU_TO_FPGA	Interfejs SPI do komunikacji z układem FPGA	Star Tracker
SPI_FPGA_CONF	Interfejs SPI mogący służyć do konfigurowania układu FPGA	Star Tracker
I2C_SYSTEM	Sprzętowy interfejs I2C obsługujący magistrale System	Wszystkie podsystemy
I2C_BB_FRAM_RTC	Programowy interfejs I2C wykorzystany do obsługi pamięci FRAM i zegara RTC	Główna jednostka obliczeniowa
I2C_BB_PAYLOAD	Programowy interfejs I2C obsługujący magistrale Payload	Wszystkie podsystemy
UART MCU_TO_SUP	Interfejs UART wykorzystany do komunikacji z układem nadzorującym	Układ nadzorujący
MCU_BOOT_EN	Sygnal wprowadzający mikrokontroler w tryb programowania przez interfejs UART	Układ nadzorujący
MCU_RESET	Zewnętrzne sygnały resetu	Układ nadzorujący
MCU_CTRL_SIG	Sygnały świadczące o prawidłowej pracy mikrokontrolera	Układ nadzorujący
CAN_TRANSCEIVER	Interfejs do zewnętrznego kontrolera warstwy fizycznej sieci CAN	
JTAG	Interfejs programujący mikrokontrolery	

Tablica 5.3: Sygnały komunikacyjne *Głównej jednostki obliczeniowej*

Nie można łączyć dwóch nadrzędnych interfejsów SPI równolegle dlatego też, magistrale SPI mikrokontrolerów podłączone są do pamięci poprzez multipleksery. Wybranie aktywnego procesora powoduje przełączenie multiplekserów i dołączenie interfejsów do zewnętrznych peryferiów. W danej chwili tylko jeden mikrokontroler może korzystać z pamięci. W taki sam



Rysunek 5.1: Schemat blokowy głównej jednostki obliczeniowej

sposób przełączane są sygnały i magistrale SPI wykorzystywane do komunikacji z podsystemem Star Tracker oraz służące do konfiguracji układu programowalnego.

Mikrokontroler TMS570 wyposażony jest tylko w jeden sprzętowy interfejs I2C. Dlatego też, aby zapewnić wymaganą ilość niezależnych magistrów, zaimplementowano programowe interfejsy I2C_BB_Payload oraz I2C_BB_FRAM_RTC. I2C_BB_FRAM_RTC obsługuje pamięć FRAM oraz zegar czasu rzeczywistego. Zegar RTC wyposażono w zewnętrzny kondensator podtrzymujący napięcie. Prawdopodobna jest sytuacja w której satelita nie będzie oświetlany przez promienie słoneczne i w konsekwencji może nastąpić zanik napięcia zasilającego wszystkie moduły. Dlatego też, tak dobrano pojemność kondensatora podtrzymującego napięcia aby układ RTC zachował konfiguracje przez 2 godziny¹. Pojemność kondensatora obliczona korzystając z dedykowanego kalkulatora [153].

Do jednej magistrali I2C może być dołączonych kilka układów nadzorzących. Jednak należy wziąć pod uwagę iż, promieniowanie jonizujące jest w stanie nieodwracalnie uszkodzić układ podłączony do interfejsu I2C. W konsekwencji linie zegara i danych I2C mogą zostać trwale zwartych do masy blokując całą magistralę. Aby temu zapobiec, każde urządzenie podłączone jest do interfejsu I2C poprzez układ PCA9517. Jest to retransmiter magistrali wyposażony w wejście Enable służące do blokowania wyjść interfejsu I2C. W przypadku uszkodzenia układu podłączonego do I2C, układ nadzorujący odetnie niesprawne urządzenie od magistrali zmieniając stal logiczny na pinie Enable układu PCA9517.

Każdy z mikrokontrolerów komunikuje się z układem nadzorującym poprzez interfejs

¹Z symulacji przeprowadzonych przez zespół projektowy misji PW-Sat2 wynika iż, satelita nie będzie oświetlony przez Słońce przez maksymalnie 2 godziny [110].

UART. Tu także zastosowano multiplekser przełączający interfejs pomiędzy aktywnym mikrokontrolerem głównym a układem nadzorującym. Dzięki interfejsowi UART i sygnałowi MCU_BOOT_EN nadzorca ma możliwość przeprogramowania każdego mikrokontrolera głównego.

Poprawna praca układów TMS570 sygnalizowana jest poprzez cykliczną zmianę stanu na pinie podłączonym do układu nadzorującego. Ponadto, mikrokontroler TMS570 wyposażony jest w specjalne wyjście informujące o wystąpieniu błędu. Wyjście to także monitorowane jest przez układ nadzorujący. W przypadku niepojawienia się sygnału świadczącego o poprawnej pracy lub wystąpienia informacji o błędzie, mikrokontroler główny jest resetowany.

Interfejs CAN oraz niezbędne sygnały sterujące podłączone przez multiplekser do układu SN65HVD234 [154] będącego transceiver'em magistrali. Do linii wyjściowych magistrali CAN podłączono rezystor terminujący. Jest on niezbędny do poprawnej pracy magistrali.

Mikrokontrolery TMS570 połączone są w łańcuchu interfejsu JTAG. Umożliwia to programowanie i debugowanie obu układów poprzez jedno złącze JTAG. Manipulując montażem odpowiednich rezystorów możliwe jest rozpięcie łańcucha JTAG.

Interfejs	Tryb	Opis
ByteBlaster II Download Cable	JTAG [198]	Podczas prac rozwojowych możliwe jest wgranie konfiguracji do układu FPGA, poprzez interfejs JTAG, przy wykorzystaniu zewnętrznego programatora ByteBlaster II [198]
Configuration flash	AS [198]	Istnieje możliwość wgrania konfiguracji układu FPGA do zewnętrznej pamięci [198]. Podczas startu systemu, FPGA jest konfigurowane na podstawie ustawień zapisanych w pamięci flash.
Zewnętrzny mi- krokontroler	PS [198]	Układ FPGA konfigurowany jest przez zewnętrzny mikrokontroler przy wykorzystaniu interfejsu PS (ang. passive serial).

Tablica 5.4: Obsługiwane tryby konfiguracji układu FPGA

5.1.2. Układ nadzorujący

Mikrokontroler ATmega128 pełni rolę układu nadzorującego. Dzięki dużej ilości portów wejścia/wyjścia i niskiemu poborowi energii doskonale nadaje się do wykonywania prostych funkcji. Schemat blokowy podsystemu nadzorującego znajduje się na rysunku 5.2. Do układu, poprzez interfejs SPI dołączono zewnętrzną pamięć Flash pojemności 2 MB. Jej zadaniem jest przechowywanie oprogramowania wbudowanego przeznaczonego do aktualizacji głównych mikrokontrolerów. Układ nadzorujący komunikuje się z każdym z procesorów TMS570 dzięki przełączaniu interfejsu UART. Ponadto, odczytuje sygnały kontrolne i ustawia sygnały resetujące. W razie konieczności, nadzorca odcina niesprawne układy od magistrali I2C. Informacje o wadliwym układzie otrzymuje przez interfejs UART od głównego mikrokontrolera. Podsystem kontrolny może też nawiązać dwustronną komunikację z modułem Star Tracker. Wykorzystuje do tego celu kolejny interfejs UART. Ważnym zadaniem podsystemu nadzorującego jest sterowanie sekcją zasilania. Mikrokontroler ATmega128 może włączyć lub wyłączyć każdą przetwornice DC/DC na module komputera pokładowego. Wykorzystując magistralę I2C, nadzorca odbiera informacje o wartości napięcia i poborze prądu niezależnie

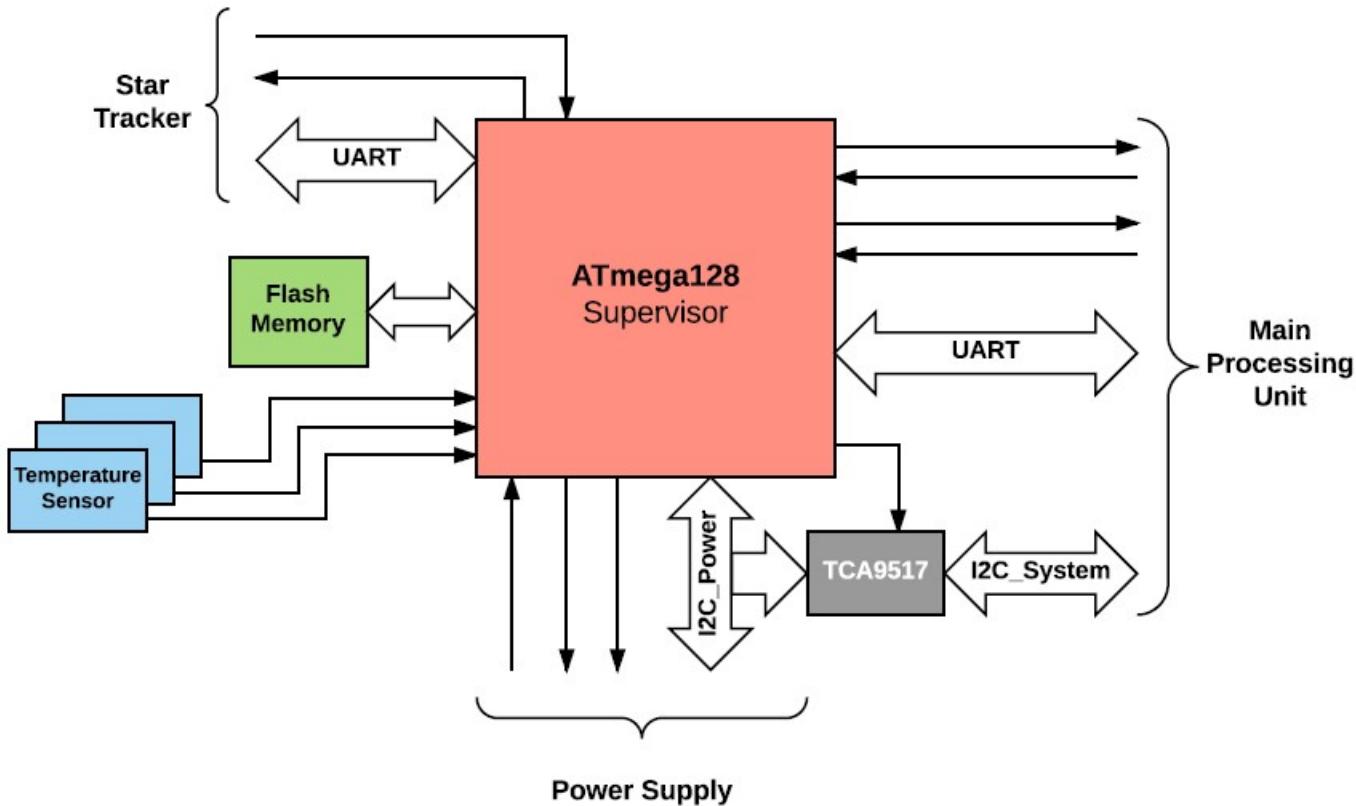
z każdej linii zasilającej. Dodatkowo, specjalne sygnały informują mikrokontroler ATmega128 o stanie głównej szyny zasilania. Sygnalizowane jest zbliżanie się do ustawionego limitu poboru prądu oraz przekroczenie wartości maksymalnej. Dzięki temu układ nadzorujący może szybko zareagować na nagły wzrost poboru energii. W trzech miejscach na płytce PCB komputera pokładowego umieszczono czujniki temperatury LM60 [155]. Ich zakres pomiarowy wynosi od -40°C do $+125^{\circ}\text{C}$. Wyjścia analogowe czujników podłączone do wejść przetwornika analogowo cyfrowego układu nadzorującego. Mikrokontroler ATmega128 programuje się poprzez dedykowany interfejs ISP.

Nazwa sygnału lub grupy sygnałów	Opis	Podsystem
SPI_FLASH	Interfejs SPI do pamięci Flash	Układ nadzorujący
UART MCU_TO_SUP	Interfejs UART wykorzystany do komunikacji z głównymi mikrokontrolerami	Główna jednostka obliczeniowa
MCU_CTRL_SIG	Sygnały świadczące o prawidłowej pracy głównych mikrokontrolerów	Główna jednostka obliczeniowa
MCU_RESET	Sygnały resetujące główne mikrokontrolery	Główna jednostka obliczeniowa
MCU_BOOT_EN	Sygnały wprowadzające główne mikrokontrolery w tryb programowania	Główna jednostka obliczeniowa
BUS_I2C_CTRL	Sygnały sterujące retransmiterami I2C	Główna jednostka obliczeniowa
MUX_CTRL	Sygnały sterujące multipleksерami	Główna jednostka obliczeniowa
UART_TO_FPGA	Interfejs UART wykorzystany do komunikacji z układem FPGA	Star Tracker
POWER_CTRL	Sygnały sterujące przetwornicami DC/DC	Sekcja zasilania
I2C_POWER	Interfejs I2C komunikujący się z układami mierzącymi napięcie i pobór prądu	Sekcja zasilania
POWER_WAR_SIG	Sygnały ostrzegające o przekraczaniu limitu zużycia energii	Sekcja zasilania
VOLT_MEAS_ADC	Dodatkowe sygnały	Sekcja zasilania
TEMP_MEAS_ADC	Sygnały analogowe z czujników temperatury	
ISP	Interfejs programujący mikrokontroler ATmega128	

Tablica 5.5: Sygnały układu nadzorującego

5.1.3. System zasilania

Schemat blokowy sekcji zasilania przedstawia rysunek 5.3. Składa się ona z przetwornic DC/DC oraz układów mierzących napięcie i pobór prądu. Doprowadzone do komputera po-



Rysunek 5.2: Schemat blokowy układu nadzorującego

kładowego napięcia zasilające może zmieniać się od 8V do nawet 16V. Mimo tego podsystem zasilający musi dostarczyć stabilnych napięć o wartościach 3.3V, 2.5V, 1.5V oraz 1.2V.

Układ nadzorujący zasilany jest z niezależnego źródła, aby mógł wyłączyć wszystkie przetwornice w razie konieczności. Dlatego zastosowano dodatkowy układ dostarczający tylko i wyłączenie napięcia dla podsystemu nadzorującego. Dodatkowo zastosowano układ TPS2553 aby chronić mikrokontroler ATmega128 przed skutkami SEL. W razie wzrostu pobieranego prądu powyżej ustawionego limitu (limit ten ustawiany jest na etapie projektowania), TPS2553 rozłączy linie zasilającą układ nadzorujący i włączy ją ponownie po ustalonym czasie. Ponadto, układ ten posiada wyjście *Fault* sygnalizujące wystąpienie przekroczenia wartości prądu oraz wejście *Enable* pozwalające na manualne włączanie lub wyłączenie napięcia wyjściowego.

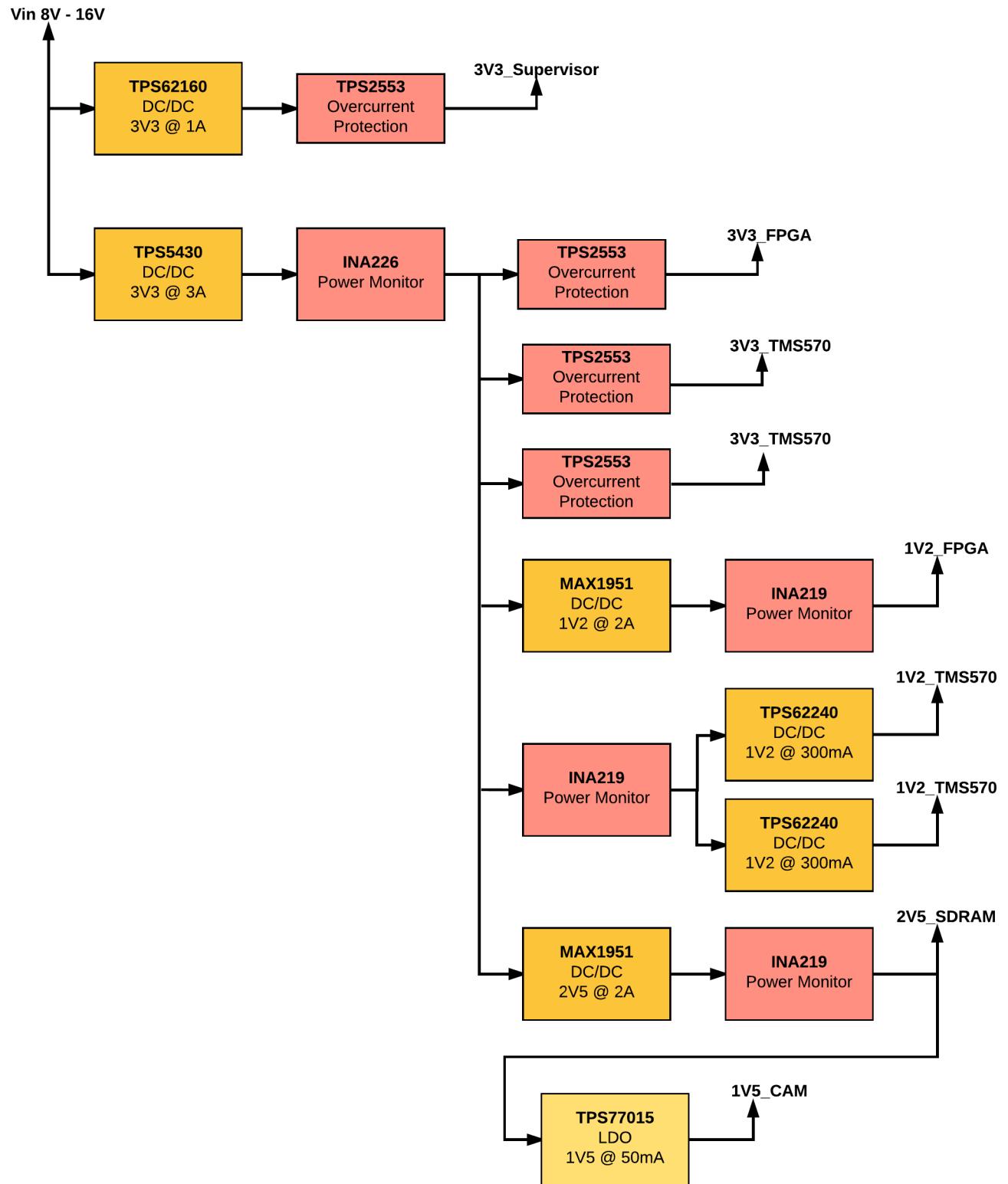
Przetwornica TPS5430 obniża napięcie wejściowe doprowadzone do komputera pokładowego do poziomu 3.3V. Aby zwiększyć efektywność, z tego napięcia zasilane są tez inne przetwornice. Do pomiaru prądu i napięcia na linii 3.3V zastosowano układ INA209 który dodatkowo wyposażony jest w wyjścia ostrzegawcze. Każdy mikrokontroler TMS570 zasilany jest napięciem 1.2V pochodzący z oddzielnych przetwornic DC/DC. Zabezpieczono je przed skutkami SEL poprzez włączenie w pętle napięcia zasilającego układu mierzącego pobór prądu. Z racji ograniczonej ilości miejsca na płytce zastosowano tylko jeden układ dostarczający informacje o pobieranym prądzie przez obie przetwornice. Napięcie 3.3V zasilające obwody wejścia wyjścia układów TMS570 także doprowadzono przez zabezpieczenie w postaci TPS2553. Dodatkowo, sterując odpowiednim pinem wejściowym TPS2553 możliwe jest selektywne wyłączenie zasilania każdemu z mikrokontrolerów.

Przetwornica MAX1951 dostarcza napięcia o wartości 1.2V układowi programowanemu. Układy INA219 mierzą napięcie i pobór prądu na liniach 1.2V_FPGA oraz 2.5V_DDR. W

zależności od wykorzystanych zasobów układu programowalnego pobiera on więcej lub mniej energii. W związku z tym odrzucono opcję zabezpieczenia układu FPGA przed skutkami SEE poprzez TPS2553 na rzecz bezpośredniego pomiaru prądu na linii zasilającej. Układ TPS2553 zastosowano do zabezpieczenia szyny 3.3V_FPGA ponieważ zasila ona tylko obwody wyjściowe. Napięcie 1.5V niezbędne jest do zasilania kamery numer 2.

Nazwa sygnału lub grupy sygnałów	Opis	Podsystem
POWER_CTRL	Sygnały sterujące przetwornicami DC/DC	Układ nadzorujący
I2C_POWER	Interfejs I2C komunikujący się z układami mierzącymi napięcie oraz pobór prądu	Układ nadzorujący
VOLT_MEAS_ADC	Dodatkowe sygnały analogowe przeznaczone do monitorowania napięcia	Układ nadzorujący
POWER_WAR_SIG	Sygnały ostrzegające o przekroczeniu limitu poboru prądu	Układ nadzorujący

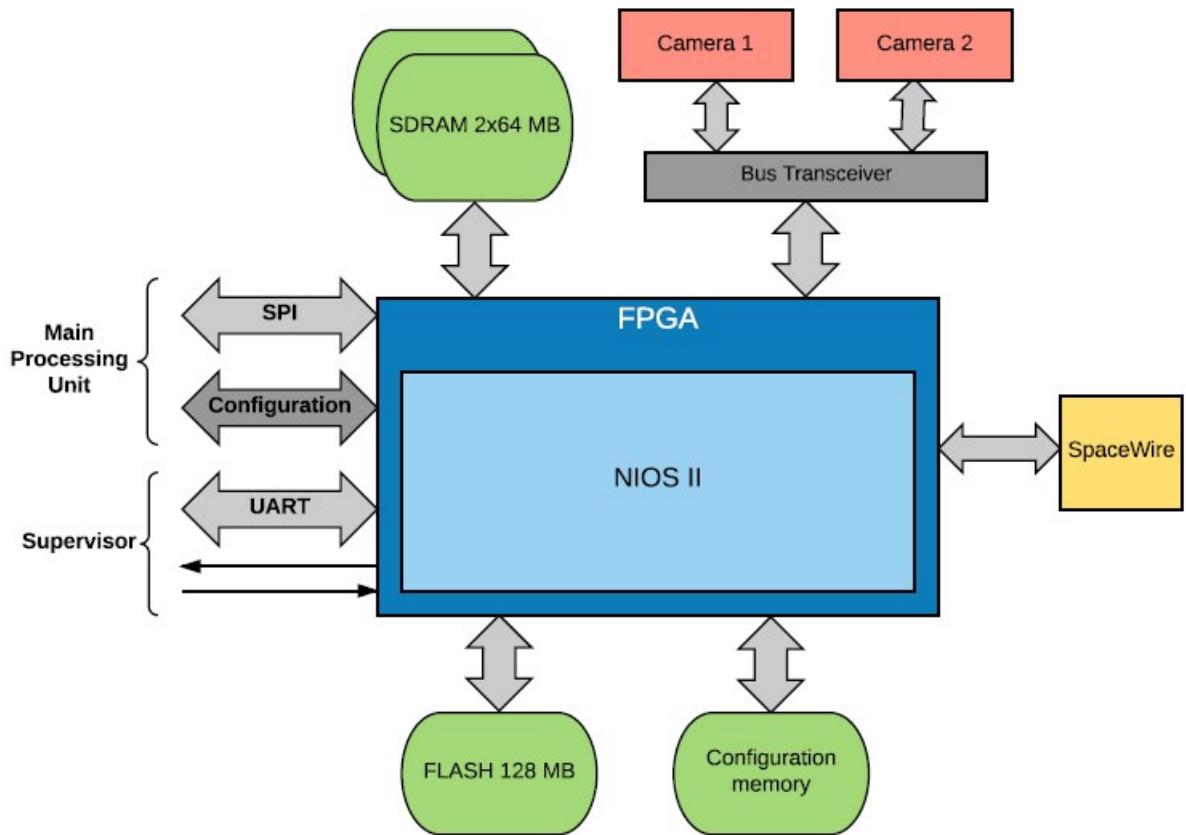
Tablica 5.6: Sygnały systemu zasilającego



Rysunek 5.3: Schemat blokowy systemu zasilania

5.1.4. Star Tracker

Podsystem Star Tracker'a możemy podzielić na układ programowalny, pamięć SDRAM i moduł sensora obrazu. Schemat blokowy podsystemu star Tracker pokazano na rysunku 5.4.



Rysunek 5.4: Schemat blokowy podsystemu Star Tracker

Układ FPGA

Spośród wielu dostępnych układów programowalnych wybrano EP3C25Q240 firmy Altera ponieważ zawiera względnie dużą ilość zasobów niezbędnych do uruchomienia prototypu oraz późniejszej implementacji różnych algorytmów. Ponadto EP3C25Q240 posiada obudowę QFP-240. Zasoby sprzętowe układu przedstawiono w tabeli 5.8.

Tabela 5.4 przedstawia dostępne tryby które umożliwiają wgranie konfiguracji do układu FPGA. W projekcie komputera pokładowego możliwe jest wybranie pomiędzy konfiguracją z wykorzystaniem zewnętrznej pamięci bądź przy użyciu mikrokontrolera głównego. Wyboru dokonuje się na etapie montażu urządzenia montując odpowiednie zwory. Poprawna konfiguracja układu zgłaszana jest mikrokontrolerowi głównemu. Do odpowiedniego wejścia doprowadzono sygnał zegarowy o częstotliwości 50 Mhz pochodzący z generatora kwarcowego. Układ programowalny ma dostęp do zewnętrznej pamięci NOR Flash o pojemności 16 MB przeznaczonej do przechowywania krótkich filmów i zdjęć wykonanych kamerą numer 2.

Cały jeden bank układu programowalnego wykorzystano do wyprowadzenia czterech par różnicowych przeznaczonych do realizacji interfejsu Space Wire. Interfejs ten wyprowadzono na dziesięcio-pinowe złącze PFC o rastrze styków 0.5mm.

Nazwa sygnału lub grupy sygnałów	Opis	Podsystem
CAM_1_BUS	Interfejs kamery 1	Star Tracker
CAM_2_BUS	Interfajs kamery 2	Star Tracker
SPI_MC_TO_FPGA	Interfejs SPI do komunikacji z głównym mikrokontrolerem	Główna jednostka obliczeniowa
UART_TO_FPGA	Interfaj UART komunikujący się z układem nadzorującym	Układ nadzorujący
SDRAM	Interfejs przeznaczony do obsługi pamięci SDRAM	Star Tracker
SPACE_WIRE	Pary różnicowe wykorzystane do realizacji interfejsu Space-Wire	
SPI_FLASH	Interfejs SPI obsługujący zewnętrzną pamięć Flash	Star Tracker
SPI_FPGA_CONF	Interfejs SPI mogący służyć do programowania układu FPGA	Główna jednostka obliczeniowa
JTAG	Interfejs programujący układ FPGA	

Tablica 5.7: Sygnały podsystemu Star Tracker

Zasób	Ilość
Elementy logiczne	24624
Blok M9K	66
Pamięć RAM	0.6 Mbit
Sprzętowe bloki mnożace	66
Pętle PLL	4
Linie wejścia / wyjścia	148
Pary różnicowe	43

Tablica 5.8: Zasoby sprzętowe układu EP3C25Q240.

Pamięć SDRAM

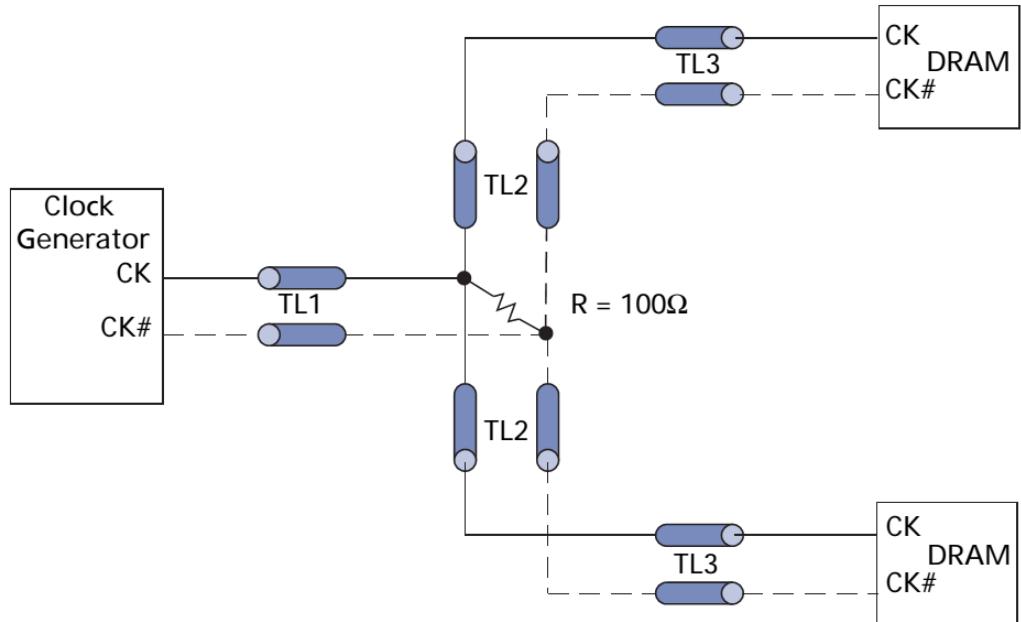
W celu tymczasowego przetrzymywania odebranych klatek obrazu wykorzystano równoległą pamięć DDR SDRAM MT46V64M8 firmy Micron [158]. Dzięki użyciu dwóch 64 MB układów uzyskano łączną pojemność 128 MB.

Układ *MT46V64M8* pracuje z 8-bitową szyną danych. Pamięć DDR podłączono do deodykowanych portów układu programowanego. Układ FPGA posiada dwie grupy 8-bitowe wewnętrzne kontrolera pamięci DDR, dlatego aby wykorzystać całą szerokość szyny danych podłączono dwa układu pamięci SDRAM. Linie komend (CMD), zegara (CK) i adresu (ADDR) są współdzielone. Natomiast linie danych (DQ), maski aktywnego banku(DM) oraz strobu danych są połączone punkt-punkt. W tabeli 5.9 przedstawiono sygnały pamięci SDRAM wraz z opisem.

W celu zapewnienia poprawnej komunikacji, przy wysokiej częstotliwości pracy interfejsu do pamięci DDR, zastosowano odpowiednią terminację. Wybrano terminacje szeregową (rys.

Sygnal	Typ	Interfejs elektryczny	Opis
DQ[8]	wejście/wyjście	SSTL-2 Class I	Wejście, wyjście danych
DQS[1]	wejście/wyjście	SSTL-2 Class I	Wejście, wyjście strobu danych
DQM[1]	wejście	SSTL-2 Class I	Wejście maski
BA[2]	wejście	SSTL-2 Class I	Adres banków
ADDR[13]	wejście	SSTL-2 Class I	Magistrala adresująca rzędy i kolumny pamięci
CKE	wejście	SSTL-2 Class I	Wejście włączające bądź wyłączające sygnał zegarowy
CK[2]	wejście	SSTL-2 Class I	Różnicowy sygnał zegarowy
WE[1], CAS[1], RAS[1]	wejście	SSTL-2 Class I	Wejścia komend. Przeznaczone do wybierania trybu pracy pamięci
CS[1]	wejście	SSTL-2 Class I	Wejście włączające bądź wyłączające dekoder komend

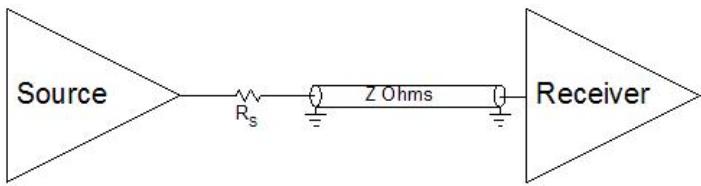
Tablica 5.9: Sygnały pamięci SDRAM *MT46V64M8* wraz z opisem i rodzajem interfejsu elektrycznego



Rysunek 5.5: Dystrybucja zegara do pamięci oraz sposób terminacji [161]

5.6) [159] ponieważ działa na zasadzie ograniczania prądu wprowadzanego do linii. Kiedy inne rodzaje terminacji rozpraszają dodatkową energię, terminacja szeregowa zużywa mniej energii, nawet niż w przypadku braku terminacji. Redukowanie zużywanej energii jest bardzo istotne w przypadku satelity zasilanego baterijnie.

Zegar różnicowy CK zterminowano przy pomocy rezystora 100ohm. Następnie rozdzielono linię różnicową i podłączono ją do dwóch układów pamięci DDR (rys. 5.5). Przy po-



Rysunek 5.6: Idea terminacji szeregowej [160]

Łączeniach DQS oraz DQ zastosowano zewnętrzny szeregowy rezystor terminujący. Wartości rezystorów terminujących dobrano na podstawie symulacji integralności sygnałowej. Linie komend i adresu także zeterminowano przy pomocy szeregowego rezystora którego wartość wyznaczono symulacyjnie.

Moduły kamer

Do układu programowalnego podłączono dwa moduły kamer. Pierwsza kamera wykorzystywana jest do obserwacji gwiazd w celu wyznaczenia orientacji w przestrzeni kosmicznej. Natomiast druga, dodatkowa kamera może posłużyć do nagrywania krótkich filmów oraz do wykonywania zdjęć.

Pierwsza kamera oparta jest na sensorze CMOS *MT9D111*. Zdecydowano się na ten układ ponieważ wyposażony jest w wbudowany kompresor JPEG oraz zużywa mało energii. Najważniejsze parametry sensora przedstawiono w tabeli 5.10. Wyposażony jest on w równoległy interfejs z 8-bitową linią danych. Sygnały interfejsu przedstawiono w tabeli 5.11. Sensor wyposażony jest również w interfejs I2C przeznaczony do konfigurowania podstawowych parametrów obrazu oraz wielu innych opcji układu (między innymi sposobu kodowania, preferencji kompresji JPEG).

Parametr	Wartość
Format	1/3.2 cala
Rozdzielcość	1600 x 1200 pikseli
Maksymalna ilość klatek	15 fps
Rozdzielcość ADC	10 bitów
Format danych wyjściowych	YCbCr, 565RGB, 555RGB, 444RGB, JPEG 4:2:2, JPEG 4:2:0
Dodatkowe zalety	Wbudowany kompresor JPEG

Tablica 5.10: Najważniejsze parametry sensora *MT9D111*

Moduł kamery zaprojektowany został na oddzielnej płycie PCB podłączanej do komputera pokładowego 30-pinową tasiemką PFC. Takie rozwiązanie pozwala na umieszczenie kamery w praktycznie dowolnym miejscu w satelicie. Płytkę sensora obrazu wyposażona jest w mocowanie CS przeznaczone do montażu obiektywu. Zastosowano obiektyw 1/3 cala o ogniskowej 16 mm i jasności F1.2. Zdjęcie gotowego modułu kamery znajduje się na rysunku 5.7.

Jako dodatkową kamerę zastosowano moduł *Li-OV9712-FF-65* [162] firmy *Leopard Imaging* [163]. Bazuje on na sensorze *OmniVision OV9712* [164]. Zdjęcie kamery pokazano na rysunku 5.8. Podstawowe parametry czujnika obrazu przedstawiono w tabeli 5.12.

Sygnały	Opis
Data[8]	Równoległa magistrala niosąca informacje o pikselu. Transmitowane dane zmieniają się w takt sygnału zegarowego PIXCLK.
Hsync	Pojawienie się tego sygnału oznacza że jedna linia pikseli została przesłana.
Vsync	Sygnal ten pojawia się gdy jedna cała ramka zostanie przetransmitowana.
PIXCLK	Sygnal zegarowy w takt którego przesyłane są kolejne piksele.

Tablica 5.11: Interfejs sensora obrazu



Rysunek 5.7: Moduł kamery *Star Tracker*

Kamera ta także posiada równoległy interfejs przeznaczony do przesyłania ramek obrazu oraz wyposażona jest w magistralę I2C służącą do konfigurowania parametrów czujnika. Dzięki małym wymiarom wynoszącym zaledwie 6.5 mm na 6.5 mm oraz niskiemu poborowi energii podczas pracy (około 0.1 W) doskonale nadaje się do umieszczenia na nano-satelicie.

Moduł posiada wbudowany obiektyw 1/4 cala o jasności F2.8 i zakresie ogniskowej od 60 cm do nieskończoności. Kamera ma zintegrowaną taśmę typu flex zakończoną 24-pinowym wtykiem FFC. Podłączona jest do złączna FH12-24S-0.5SH [165] znajdującego się na płycie komputera pokładowego.

Z racji ograniczonej ilości linii wejścia-wyjścia układu FPGA, aby odebrać dane z dwóch

sensorów obrazu postanowiono współdzielić sygnały danych (*DATA*) oraz synchronizacji poziomej (*Hsync*) i pionowej (*Vsync*). Do przełączania współdzielonych linii wykorzystano układy *SN74AVC8T245RHLR* firmy *Texas Instruments*. Sterując odpowiednimi liniami układów *SN74AVC8T245RHLR* można dołączyć lub odłączyć wybraną kamerę od układu programowalnego.

Parametr	Wartość
Format	1/4 cala
Rozdzielcość	1280 x 800 pikseli
Maksymalna ilość klatek	30 fps
Rozdzielcość ADC	10 bitów
Format danych wyjściowych	10-bit RAW RGB

Tablica 5.12: Najważniejsze parametry sensora *Li-OV9712-FF-65*



Rysunek 5.8: Moduł kamery *Li-OV9712-FF-65* [162]

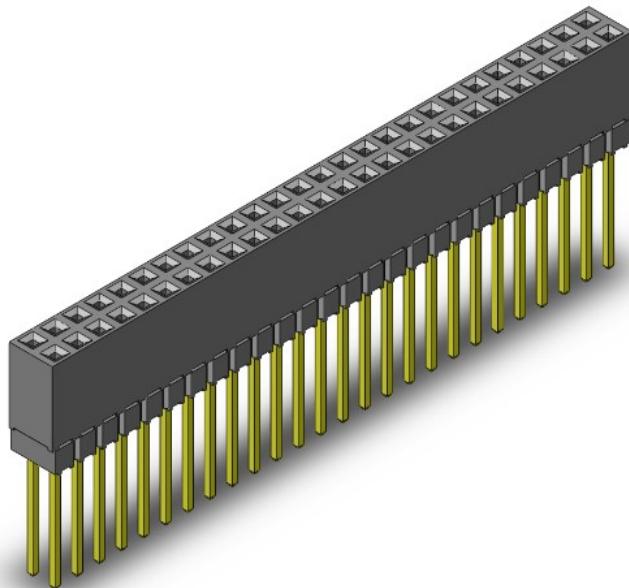
5.2. Projekt wielowarstwowego obwodu drukowanego PCB

W celu połączenia wszystkich układów elektronicznych, doprowadzenia napięć zasilających oraz zapewnienia poprawnej transmisji szybkich sygnałów zaprojektowano wielowarstwowy obwód drukowany. Płytkę PCB zaprojektowano w programie Altium Designer 14.1, następnie przeprowadzono symulacje integralności sygnałowej w programie HyperLynx firmy Mentor Graphics. W tabeli zaprezentowano stack-up zaprojektowanego obwodu drukowanego z opisem wszystkich warstw.

Nr warstwy	Rodzaj	Grubość [mm]	Opis
	TOP_SOLDER	0.0127	Soldermaska dla warstwy TOP
1	TOP	0.035	Warstwa sygnałowa Top
	Core	0.129	Warstwa technologiczna
2	Power	0.035	Warstwa Power
	PREPREG	0.986	Warstwa technologiczna
3	GND	0.035	Warstwa GND
	Core	0.129	Warstwa technologiczna
4	Bottom	0.035	Warstwa sygnałowa Bottom
	BOTTOM_SOLDER	0.0127	Soldermaska dla warstwy BOTTOM

Tablica 5.13: Opis wielowarstwowego obwodu drukowanego

Obwód drukowany został tak zaprojektowany aby jego wielkość i rozmieszczenie głównego złącza oraz otworów montażowych była zgodna ze standardem PC-104. Dlatego też, wielkość płytki PCB wynosi 92mm x 96mm. Jako główne złącze zastosowano dwa złącza ESQ-126-39-G-D [166] firmy Samtec [167] (rys. 5.9).



Rysunek 5.9: Złącze ESQ-126-39-G-D firmy *Samtec* [167]

5.3. Podsumowanie

Wykonany prototyp *Modułu komputera pokładowego* wyposażony został w szereg rozwiązań zwiększących odporność systemu na negatywne skutki oddziaływanego promieniowania jonizującego. Ponadto dzięki względnie dużej ilości zasobów sprzętowych istnieje możliwość implementacji dodatkowych rozwiązać programowych zwiększących niezawodność pracy urządzenia w przestrzeni kosmicznej. Wszystkie te rozwiązania zostały zebrane w tabeli 5.14.

	SEL	SEU	SET	SEFI	dodatkowe rozwiązania
AT-mega128	monitorowanie zasilania	wbudowany Watchdog	możliwość implementacji rozwiązać programowych	wbudowany Watchdog	
TMS570	monitorowanie zasilania	redundantne rdzenie, zewnętrzny układ nadzorujący	wbudowane pamięci z ECC	zewnętrzny układ nadzorujący	możliwość zdalnej aktualizacji oprogramowania, możliwość implementacji rozwiązać programowych
układ FPGA	monitorowanie zasilania	zewnętrzny układ nadzorujący	możliwość implementacji rozwiązać programowych	zewnętrzny układ nadzorujący	możliwość konfiguracji przez mikrokontroler TMS570
SDRAM	monitorowanie zasilania	kontroler pamięci wyposażony w EDAC	kontroler pamięci wyposażony w EDAC	kontroler pamięci wyposażony w EDAC	
pozostałe układy	monitorowanie zasilania	technika Windowing		technika Windowing	

Tablica 5.14: Zastosowane rozwiązania zmniejszające wpływ promieniowania na najważniejsze układy komputera pokładowego

Rozdział 6

Oprogramowanie modułu komputera pokładowego

W niniejszym rozdziale opisano oprogramowanie przygotowane dla *Modułu komputera pokładowego*. Omówiono firmware mikrokontrolera TMS570, układu nadzorującego oraz układu FPGA. Przedstawiono, układy peryferyjne, które wykorzystano oraz funkcje jakie pełnią. Następnie przedstawiono wygenerowane oprogramowanie dla układu FPGA Altera Cyclone III.

6.1. Oprogramowanie mikrokontrolera ATmega128

W skonstruowanym *Module komputera pokładowego* wykorzystano 8-bitowy mikrokontroler o architekturze RISC [193] ATmega128A-MU firmy Atmel. W tab. 6.1 zestawiono parametry użytego układu. W celu gęstszego rozmieszczenia elementów na płycie drukowanej wykorzystano obudowę 64QFN. Możliwe jest zaprogramowanie układu poprzez interfejs ISP, wyprowadzony na złączu kołkowym. Mikrokontroler pracuje z zewnętrznym oscylatorem zegarowym o częstotliwości 8MHz.

Parametr	Wartość
Częstotliwość	25MHz
Pamięć Flash	128kB
Pamięć SRAM	8kB
GPIO	47
Liczniki	4 po 16b
Układ Watchdog	1
Kontroler I2C	1
Kontrolery SPI	4
Przetwornik ADC	12b, 16kanałów

Tablica 6.1: Zasoby mikrokontrolera ATmega128A-MU

6.1.1. Koncepcja oprogramowania

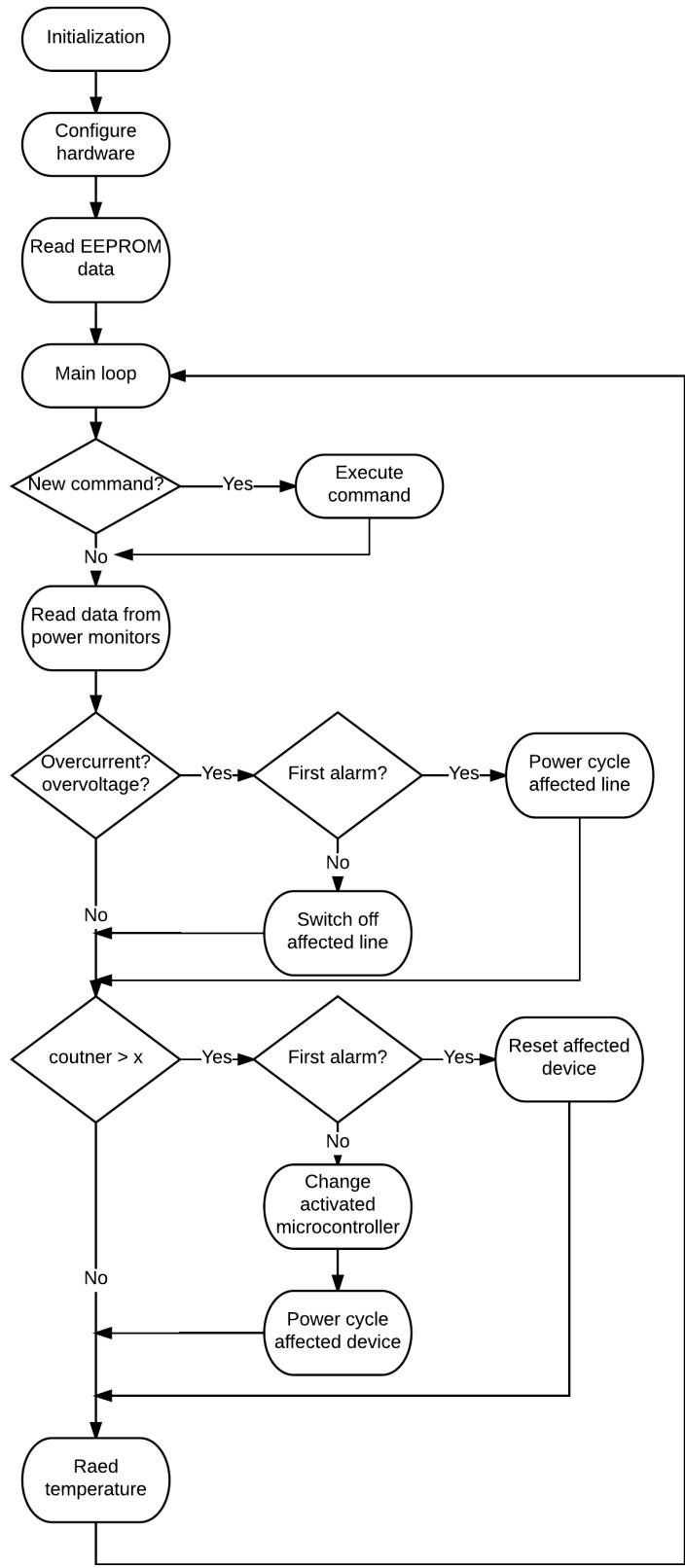
Na rys. 6.1 przedstawiono algorytm, który jest wykonywany przez mikrokontroler ATmega128A-MU. Po włączeniu zasilania odbywa się inicjalizacja, podczas której konfigurowane są cyfrowe wejścia-wyjścia oraz zastosowane układy peryferyjne. Zestawienie oraz opis użytych kontrolerów zaprezentowano w tab. 6.2. Następnie konfigurowane są układy mierzące napięcie i pobór prądu oraz włączane są przetwornice DC/DC. Kolejnym krokiem jest odczytanie z wewnętrznej pamięci EEPROM parametrów definiujących ograniczenia prądowe. W pętli głównej oprogramowanie sprawdza czy została odebrana nowa komenda, jeśli tak to następuje jej realizacja. Poprzez interfejs I²C odczytywane są dane pomiarowe ze wszystkich układów INA219 oraz z INA209. Jeśli wystąpiło przekroczenie napięcia lub dozwolonego poboru prądu na którejś linii zasilającej to dana przetwornica DC/DC jest resetowana poprzez jej wyłączenie i włączenie. W przypadku gdy sytuacja przekraczania limitów powtórzy się z rzędu ustaloną ilość razy, nastąpi całkowite wyłączenie przetwornicy. Ponowne włączenie zostanie wykonane po odebraniu odpowiedniej komendy.

Kontroler	Opis
GPIO (General purpose I/O)	Sterowanie przetwornicami DC/DC, sterowanie układami monitorującymi prąd TPS2553, ustawianie sygnałów kontrolujących multipleksery oraz układy odłączające urządzenia od magistrali I ² C, sterowanie sygnałami resetującymi, kontrola poprawności pracy mikrokontrolerów TMS570 oraz NIOS II
I²C	Komunikacja poprzez interfejs <i>I²C</i> z układami INA209, INA219 oraz poprzez układ PCA9517 z magistralą <i>I²C-System</i>
SPI	Komunikacja poprzez interfejs <i>SPI</i> z zewnętrzną pamięcią Flash AT45DB0161B
USART0 (Universal Synchronous and Asynchronous Receiver and Transmitter)	Komunikacja poprzez interfejs <i>USART</i> z układami TMS570
USART1 (Universal Synchronous and Asynchronous Receiver and Transmitter)	Komunikacja poprzez interfejs <i>USART</i> z układem FPGA Cyclone III
Watchdog	Kontrola poprawności pracy mikrokontrolera ATmega128A-MU
Timer/Counter 16-Bit Timer	Precyzyjny timer do generowania opóźnienia
External Interrupt	Kontroler przerwań zewnętrznych do którego podłączone są sygnały z mikrokontrolerów TMS570 oraz z systemu NIOS II
A/D 10-Bit Analog-to-Digital Converter	Digitalizacja napięcia z układów do pomiaru temperatury LM45

Tablica 6.2: Wykorzystane układy peryferyjne mikrokontrolera ATmega128A-MU

Każdy z mikrokontrolerów TMS570 generuje sygnał prostokątny o częstotliwości 100 Hz świadczący o jego poprawnej pracy. Sygnały te podłączone są do wejść przerwań zewnętrznych ATmega128A-MU. Funkcja obsługująca zewnętrzne przerwanie resetuje licznik przyporządkowany danemu mikrokontrolerowi. W przypadku zawieszenia się któregoś z TMS570, przerwanie nie wystąpi i w konsekwencji program główny układu nadzorującego wykryje iż licznik przekroczył zadeklarowaną wartość. Jeżeli zdarzenie to wystąpiło pierwszy raz to dany mikrokontroler zostanie zresetowany przez ATmega128A-MU. W przypadku gdy incydent pojawi się kolejny raz z rzędu, nastąpi przełączenie aktywnego układu TMS570. Nieodpowiadający mikrokontroler główny przejdzie procedurę cyklu linii zasilających.

Ostatnim krokiem wykonywanego algorytmu jest odczytanie temperatury z trzech czujników rozmieszczonych na płytce drukowanej.



Rysunek 6.1: Algorytm wykonywany przez mikrokontroler ATmega128A-MU

6.1.2. Realizacja oprogramowania

Oprogramowanie mikrokontrolera ATmega128A-MU napisano w języku C w aplikacji *Atmel Studio 7* firmy Atmel.

Interfejs USART1 przeznaczony jest do komunikacji z soft-procesorem wbudowanym w układ programowalny. Jednak w celu ułatwienia uruchamiania pisanej oprogramowania wbudowanego USART1 wykorzystano do komunikacji z terminalem UART komputera PC. Do tego celu zastosowano konwerter USB-UART i terminal portu szeregowego Termite [przy-pis].

Komunikacja z układami monitorującymi linie zasilania

Mikrokontroler ATmega128A-MU komunikuje się z układami mierzącymi prąd i napięcie poprzez interfejs I2C. Prędkość transmisji danych ustalono na 400kbps. W tabeli 6.3 znajduje się spis układów podłączonych do magistrali I2C wraz z ustawionym adresem.

Układ	Adres	Opis
INA209	0x40	Monitor linii zasilającej 3.3V
INA219	0x41	Monitor linii zasilającej 1.2V układu FPGA
INA219	0x44	Monitor linii zasilającej 1.2V mikrokontrolerów TMS570
INA219	0x46	Monitor linii zasilającej 2.5V

Tablica 6.3: Spis układów podłączonych do magistrali I2C mikrokontrolera ATmega128A-MU

Układ INA209 przeznaczony jest do monitorowania prądu oraz napięcia na linii zasilającej dostarczającej napięcia 3.3V. Używa banku 16-bitowych rejestrów do przechowywania ustawień konfiguracyjnych, wyników pomiarów, zakresów ostrzeżeń oraz informacji o statusie. Spis rejestrów wraz z adresami widnieje na rysunkach 6.2 i 6.3. Do układu INA209 dostępne jest bezpłatne oprogramowanie o nazwie INA209 EVM [168] firmy Texas Instruments ułatwiające wygenerowanie danych konfiguracyjnych i kalibracyjnych. Rysunek 6.4 przedstawia wygenerowane w programie INA209 EVM dane konfiguracyjne. Na potrzeby prototypu dokonano podstawowej konfiguracji układu INA209.

Układy INA219 przeznaczone są do monitorowania prądu i napięcia na liniach zasilających dostarczających napięcia 1.2V układowi FPGA oraz 2.5V pamięci SDRAM. Podobnie jak w przypadku INS209 układy INA219 wykorzystują 16-bitowe rejesty do których dostęp umożliwia interfejs I2C. Bank rejestrów przechowuje dane konfiguracyjne, kalibrujące a także wyniki pomiarów. Rysunek 6.5 przedstawia rejesty układowów INA219. Podstawowa konfiguracja umożliwia wykonywanie pomiarów tuż połączeniu zasilania. Jednak aby zwiększyć dokładność mierzonych wartości zmieniono podstawową konfigurację. Do generacji ustawień oraz danych kalibracyjnych posłużyły się programem INA219 EVM [169] dostarczonym przez firmę Texas Instruments. Dzięki odpowiednim ustawieniom uzyskano teoretyczną dokładność miernika prądu na poziomie 70uA/bit. Na rysunku 6.6 widoczna jest zakładka kalibracji pochodząca z programu INA219 EVM.

Maksymalne dozwolone wartości prądu i napięcia zdefiniowane są w pamięci EEPROM mikrokontrolera ATmega128. W przypadku wykrycia przekroczenia limitu na danej linii zasi-

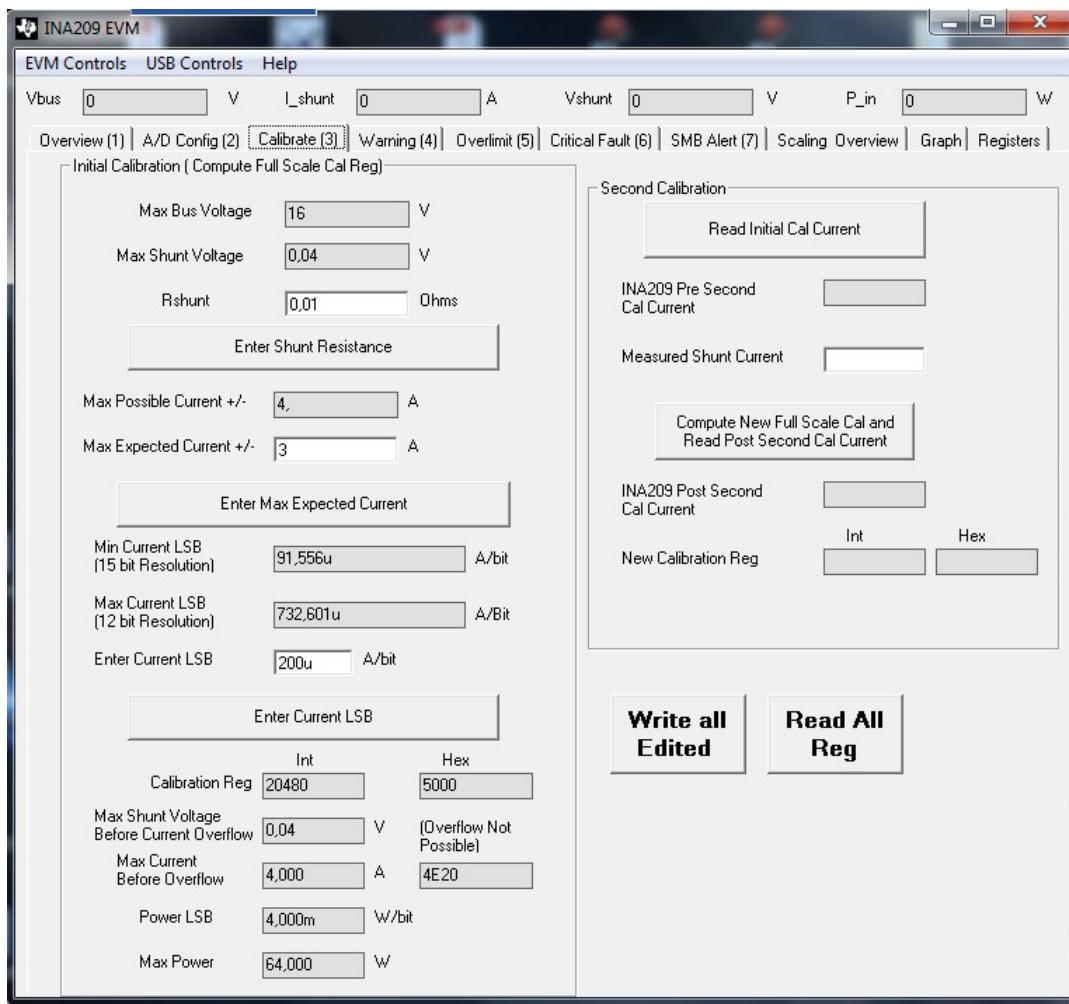
POINTER ADDRESS	REGISTER NAME	FUNCTION	POWER-ON RESET		TYPE ⁽¹⁾
			BINARY	HEX	
00	Configuration Register	All-register reset, settings for bus voltage range, PGA Gain, ADC resolution/averaging.	00111001 10011111	399F	R/W
01	Status Register	Status flags for warnings, over-/under-limits, conversion ready, math overflow, and SMBus Alert.	00000000 00000000	0000	R
02	SMBus Alert Mask/Enable Control Register	Enables/disables flags in the Status Register	00000000 00000000	0000	R/W
03	Shunt Voltage	Shunt voltage measurement data.	00000000 00000000	0000	R
04	Bus Voltage	Bus voltage measurement data.	00000000 00000000	0000	R
05	Power	Power measurement data.	00000000 00000000	0000	R
06	Current ⁽²⁾	Contains the value of the current flowing through the shunt resistor.	00000000 00000000	0000	R
07	Shunt Voltage Positive Peak	Contains most positive voltage reading of Shunt Voltage Register.	10000000 00000000	8000	R/W
08	Shunt Voltage Negative Peak	Contains most negative voltage reading of Shunt Voltage Register.	01111111 11111111	7FFF	R/W
09	Bus Voltage Maximum Peak	Contains highest voltage reading of Bus Voltage Register.	00000000 00000000	0000	R/W
0A	Bus Voltage Minimum Peak	Contains lowest voltage reading of Bus Voltage Register.	11111111 11111000	FFF8	R/W
0B	Power Peak	Contains highest power reading of Power Register.	00000000 00000000	0000	R/W
0C	Shunt Voltage Positive Warning	Warning watchdog register. Sets positive shunt voltage limit that triggers a warning flag in the Status Register, and activates Warning pin.	00000000 00000000	0000	R/W
0D	Shunt Voltage Negative Warning	Warning watchdog register. Sets negative shunt voltage limit that triggers a warning flag in the Status Register, and activates Warning pin.	00000000 00000000	0000	R/W
0E	Power Warning	Warning watchdog register. Sets power limit that triggers a warning flag in the Status Register, and activates Warning pin.	00000000 00000000	0000	R/W
0F	Bus Over-Voltage Warning	Warning watchdog register. Sets high Bus voltage limit that triggers a warning flag in the Status Register, and activates Warning pin. Also contains bits to set Warning pin polarity and latch feature.	00000000 00000000	0000	R/W

Rysunek 6.2: Spis rejestrów układu *INA209* [147]. Część 1.

10	Bus Under-Voltage Warning	Warning watchdog register. Sets low Bus voltage limit that triggers a warning flag in the Status Register and activates Warning pin.	00000000 00000000	0000	R/W
11	Power Over-Limit	Over-limit watchdog register. Sets power limit that triggers an over-limit flag in the Status Register, and activates the Overlimit pin.	00000000 00000000	0000	R/W
12	Bus Over-Voltage Over-Limit	Over-limit watchdog register. Sets Bus over-voltage limit that triggers an over-limit flag in the Status Register, and activates the Overlimit pin. Also contains bits to set Overlimit pin polarity and latch feature.	00000000 00000000	0000	R/W
13	Bus Under-Voltage Over-Limit	Over-limit watchdog register. Sets Bus under-voltage limit that triggers an over-limit flag in the Status Register, and activates the Overlimit pin.	00000000 00000000	0000	R/W
14	Critical DAC+ Register (Critical Shunt Positive Voltage)	Sets a positive limit for internal Critical DAC+. Contains bits for GPIO pin status and mode of operation, Critical Comparator latch feature and hysteresis.	00000000 00000000	0000	R/W
15	Critical DAC- Register (Critical Shunt Negative Voltage)	Sets a negative limit for internal Critical DAC-. Contains bits for Warning pin delay, and Critical Comparator output filter configuration.	00000000 00000000	0000	R/W
16	Calibration	Sets full-scale range and LSB of current and power measurements. Overall system calibration.	00000000 00000000	0000	R/W

Rysunek 6.3: Spis rejestrów układu *INA209* [147]. Część 2.

lajacej układ nadzorujący wyłączy odpowiednią przetwornice poprzez zmianę stanu na pinie Enable. Po chwili nastąpi próba ponownego uruchomienia przetwornicy.



Rysunek 6.4: Konfiguracja układu *INA209* w programie *INA209 EVM*

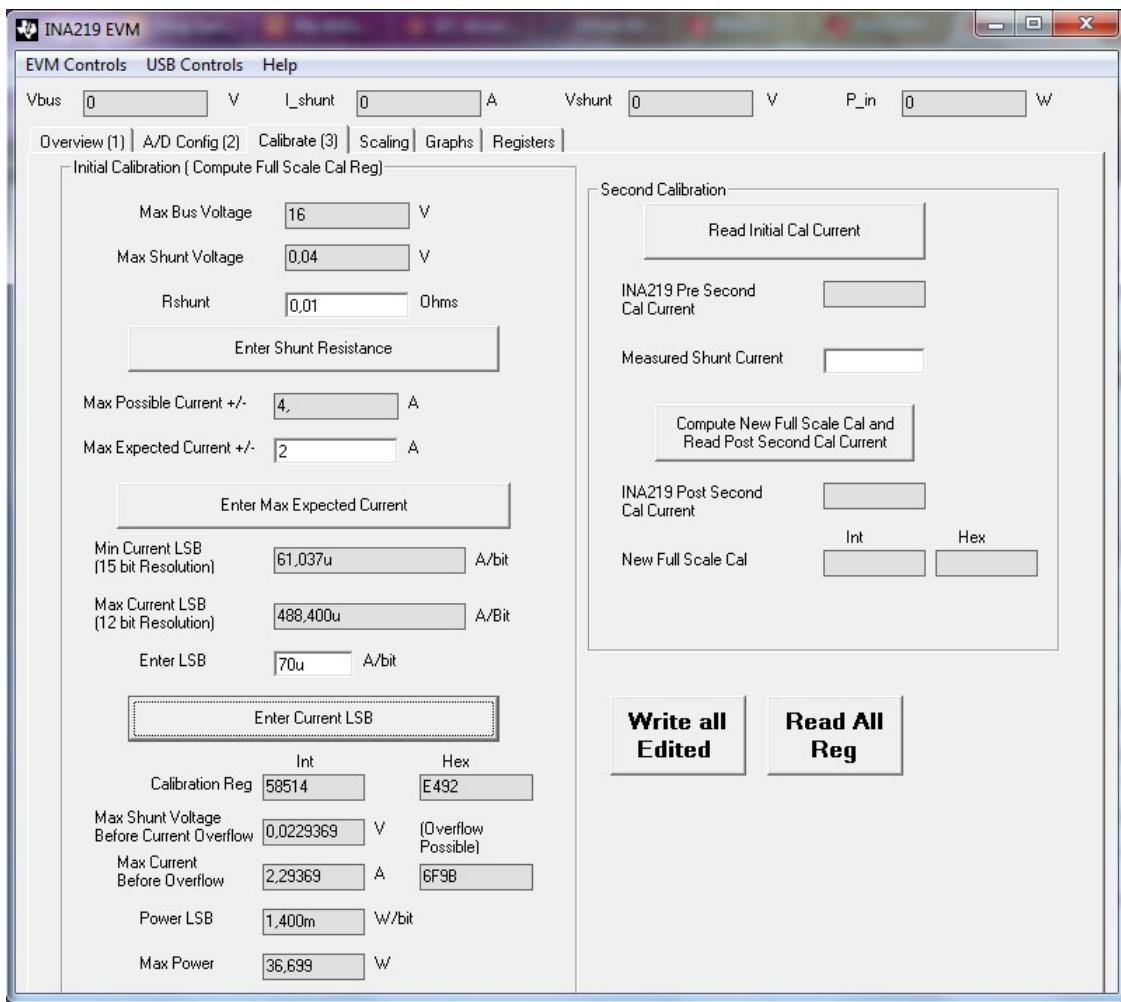
POINTER ADDRESS HEX	REGISTER NAME	FUNCTION	POWER-ON RESET		TYPE ⁽¹⁾
			BINARY	HEX	
00	Configuration	All-register reset, settings for bus voltage range, PGA Gain, ADC resolution/averaging.	00111001 10011111	399F	R/W
01	Shunt voltage	Shunt voltage measurement data.	Shunt voltage	—	R
02	Bus voltage	Bus voltage measurement data.	Bus voltage	—	R
03	Power ⁽²⁾	Power measurement data.	00000000 00000000	0000	R
04	Current ⁽²⁾	Contains the value of the current flowing through the shunt resistor.	00000000 00000000	0000	R
05	Calibration	Sets full-scale range and LSB of current and power measurements. Overall system calibration.	00000000 00000000	0000	R/W

Rysunek 6.5: Spis rejestrów układu *INA219* [148].

Sterowanie multiplekserami i przełącznikami interfejsu I2C

Oprogramowanie mikrokontrolera kontroluje multipleksery przełączające interfejsy głównych mikrokontrolerów poprzez sterowanie pinami *Output_Enable (OE)* oraz *Select (S)*. Rysunek 6.7 przedstawia stan wyjść multipleksera w zależności od poziomów logicznych sygnałów sterujących.

Program układu nadzorującego steruje pracą przełączników magistrali I2C zmieniając stan na linii *Enable* układów *PCA9517*. Rysunek 6.8 przedstawia stan układu *PCA9517* w zależności od wartości logicznej wejścia *Enable*.



Rysunek 6.6: Konfiguracja układu *INA219* w programie *INA219 EVM*

INPUTS		INPUT/OUTPUT	FUNCTION
OE	S	A	
L	L	B1	A port = B1 port
L	H	B2	A port = B2 port
H	X	Z	Disconnect

Rysunek 6.7: Sterowanie multiplekserem w zależności od stanu linii *INPUTS* [144]

INPUT EN	FUNCTION
L	Outputs disabled
H	SDAA = SDAB SCLA = SCLB

Rysunek 6.8: Funkcjonalność układu *PCA9517* w zależności od stanu linii *INPUT EN* [145]

Nadzorowanie mikrokontrolerów głównych oraz układu **FPGA**.

Mikrokontroler ATmega128 nadzoruje układy TMS570 i układ FPGA które cyklicznie generują sygnał prostokątne świadczące o poprawnej pracy. Sygnały te podłączone są do wejść przerwań zewnętrznych układu nadzorującego. Pojawienie się stanu wysokiego powo-

duje wywołanie przerwania i natychmiastowe przejście programu do funkcji obsługującej to przerwanie. Zadaniem funkcji obsługującej przerwanie jest zerowanie odpowiedniego licznika. Jeśli z jakiegoś powodu nie wystąpi sygnał przerwania, licznik nie zostanie zresetowany i licząc dalej przekroczy z góry ustaloną wartość. Będzie to świadczyć o zawieszeniu się układu nadzorowanego i w związku z tym mikrokontroler ATmega128 odłączy aktywny procesor TMS570 od interfejsów i ustawi sygnały resetujące. W przypadku układu programowalnego wystawi sygnał resetujący wewnętrzny soft-procesor. Dodatkowo ATmega128 odczytuje stan na specjalnych wyjściach nERROR mikrokontrolerów TMS570. Pojawienie się stanu wysokiego na nERROR świadczy o wystąpieniu błędu podczas pracy układu.

6.2. Oprogramowanie układu FPGA Cyclone III

Oprogramowania układu FPGA Altera EP3C25Q240 przygotowano w pakiecie Quartus II 13.0 Web Edition. Dzięki wykorzystaniu kreatorów dostępnych w środowisku Quartus II możliwe było generowanie bloków funkcjonalnych z określonymi parametrami co znacznie ułatwiło i przyśpieszyło implementację.

6.2.1. System NIOS II

W układzie programowalnym zaimplementowano 32-bitowy soft-procesor NIOS II [170] w wersji Fast. W tej wersji, procesor może pracować z częstotliwością 175Mhz [170] i używa 195 DMIPS [170]. Generator Qsys, będącym integralną częścią środowiska Quartus, pozwolił na opracowanie i wygenerowanie kompletnego mikroprocesora wraz z niezbędnymi peryferiami. Wygenerowano rdzeń wyposażony w następujące cechy:

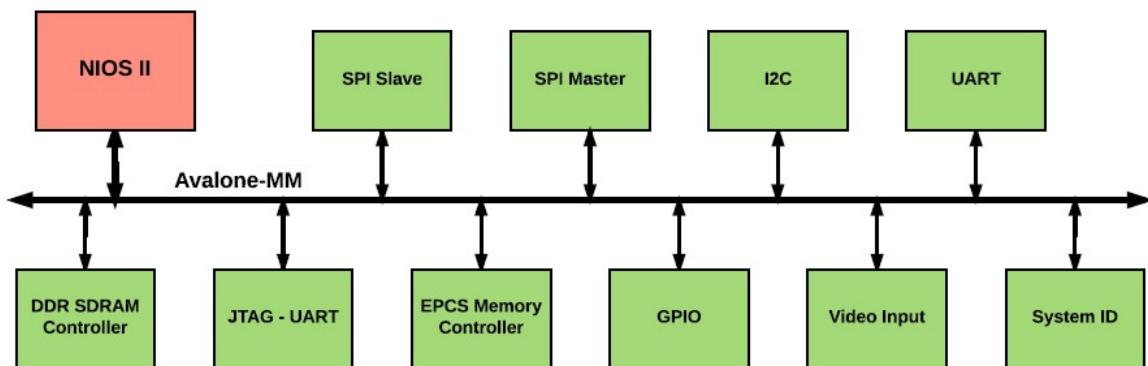
- memory protection unit (MPU),
- kontroler przerwań zewnętrznych,
- pamięć data cache o rozmiarach 2 kb,
- pamięć instruction cache o rozmiarach 4 kb,
- przestrzeń adresowa do 4 GB,
- 6-stopniowa potokowość,
- interfejs JTAG.

Do układu FPGA doprowadzono sygnał zegarowy o częstotliwości 50Mhz pochodzący z zewnętrznego generatora *CFPS-69IB* [171]. Zostaje on powielony do częstotliwości 100Mhz dzięki wbudowanej pętli PLL. Po powięleniu zostaje doprowadzony do procesora NIOS II i wszystkich jego urządzeń peryferyjnych.

Rysunek 6.9 przedstawia wygenerowany soft-procesor wraz z peryferiami dołączonymi do magistrali Avalon-MM [172]. Wykorzystując magistrale Avalon-MM dołączono do procesora szereg urządzeń peryferyjnych.

W generatorze Qsys dodano i skonfigurowano wszystkie urządzenia peryferyjne procesora NIOS II. Następnie podłączono linię zegarową, sygnał reset i magistralę Avalon-MM.

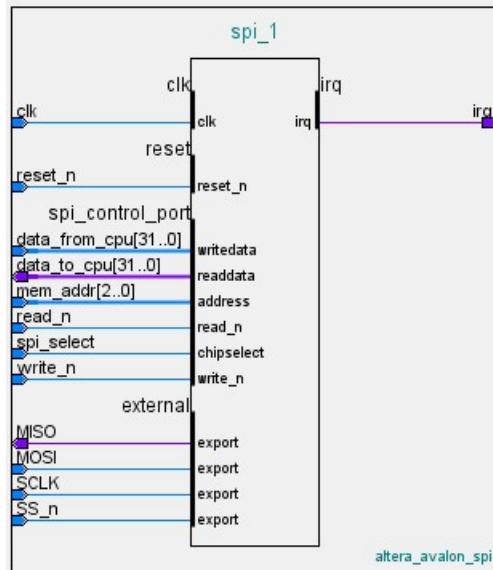
Wygenerowany procesor NIOS II wraz z blokami peryferyjnymi przeszedł poprawnie proces syntezы. Następnie wszystkie sygnały wejściowe i wyjściowe podłączono do fizycznych pinów zewnętrznych układu programowego wykorzystując narzędzie *Pin Planner*. Projekt wymagał kilku iteracji procesu syntezы oraz przyporządkowania linii wyjściowych aby poprawnie przeszedł komplikację.



Rysunek 6.9: Schemat blokowy systemu NIOS II wraz z dołączonymi peryferiami

Blok SPI Slave

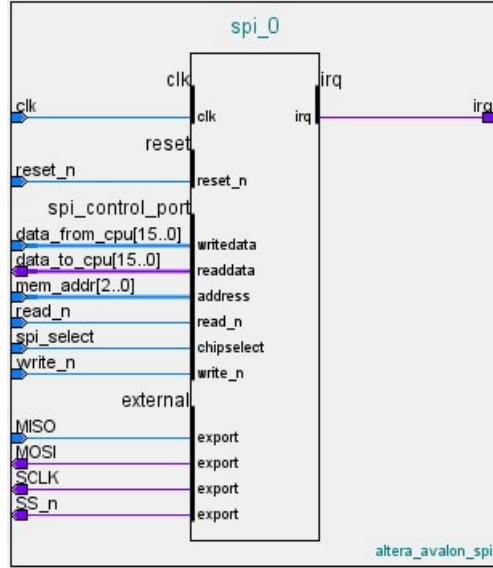
Obsługuje interfejs SPI Slave. Dzięki temu interfejsowi główny mikrokontroler ma możliwość odczytywania wyników pracy algorytmu Star Tracker oraz obrazu zarejestrowanego przez jedną z kamer. Rejestr danych ma długość 32-bitów. Rysunek 6.10 przedstawia blok wraz z sygnałami.



Rysunek 6.10: Blok interfejsu SPI Slave

Blok SPI Master

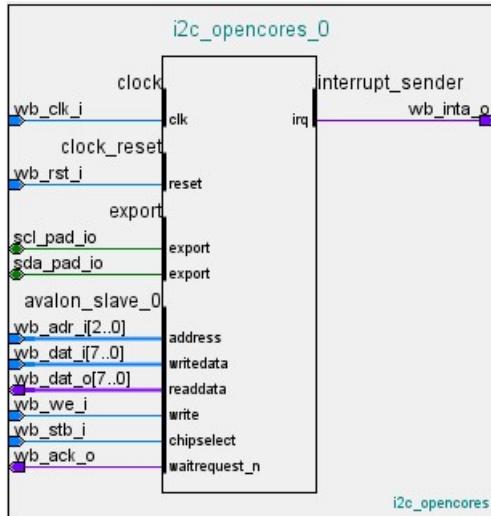
Obsługuje interfejs SPI Master przeznaczony do komunikacji z zewnętrzną pamięcią NOR Flash. Rejestr danych ma długość 32-bitów. Rysunek 6.11 przedstawia blok wraz z sygnałami.



Rysunek 6.11: Blok interfejsu SPI Master

Blok interfejsu I2C

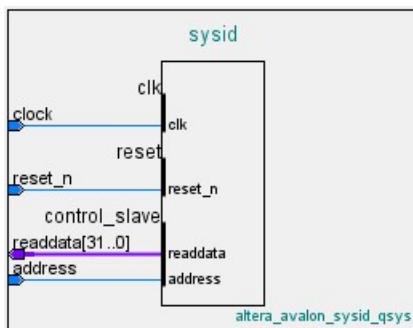
Skorzystano z darmowego IP Core interfejsu I2C [173]. IP Core obsługuje 3 prędkości magistrali: 100kbps, 400kbps i 3.5Mbps oraz 7-bitową lub 10-bitową długość adresu [174]. Rysunek 6.12 przedstawia blok interfejsu I2C wraz z nazwami sygnałów.



Rysunek 6.12: Blok interfejsu I2C

Blok System ID

Jest to prosty moduł przeznaczony tylko do odczytu i zawierający 32-bitowe słowo identyfikujące system NIOS II. Procesor wykorzystuje numer ID do weryfikacji czy wykonywany program został skompilowany pod aktualnie zaprogramowany plik konfiguracyjny. Jeżeli słowo identyfikacyjne wykonywanego programu nie zgadza się ze słowem sprzętowo zaprogramowanym, prawdopodobne jest że oprogramowanie nie będzie działać prawidłowo. Rysunek 6.13 przedstawia moduł System ID.



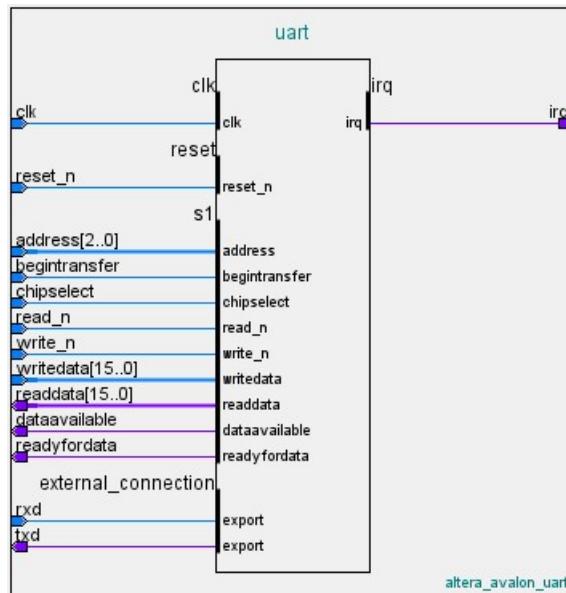
Rysunek 6.13: Blok System ID

Blok interfejsu UART

Obsługuje interfejs UART przeznaczony do komunikacji z układem nadzorującym. Parametry wygenerowanego bloku interfejsu szeregowego przedstawiono w tabeli 6.4. Na rysunku 6.14 widnieje moduł UART wraz z sygnałami wejściowymi i wyjściowymi.

Parametr	Wartość
Parity	NONE
Data bits	8
Stop bits	1
Boud Rate	115200 bps

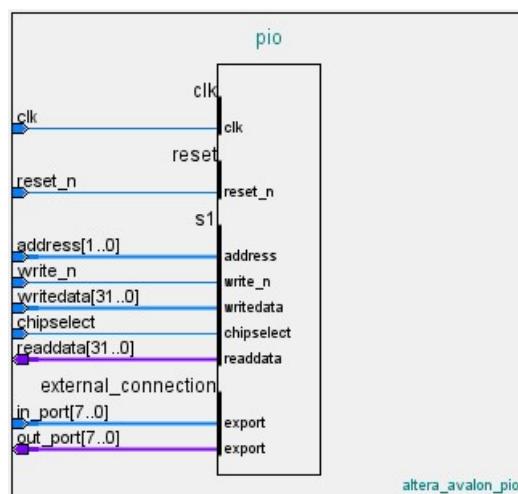
Tablica 6.4: Parametry interfejsu UART



Rysunek 6.14: Blok interfejsu UART

Blok GPIO

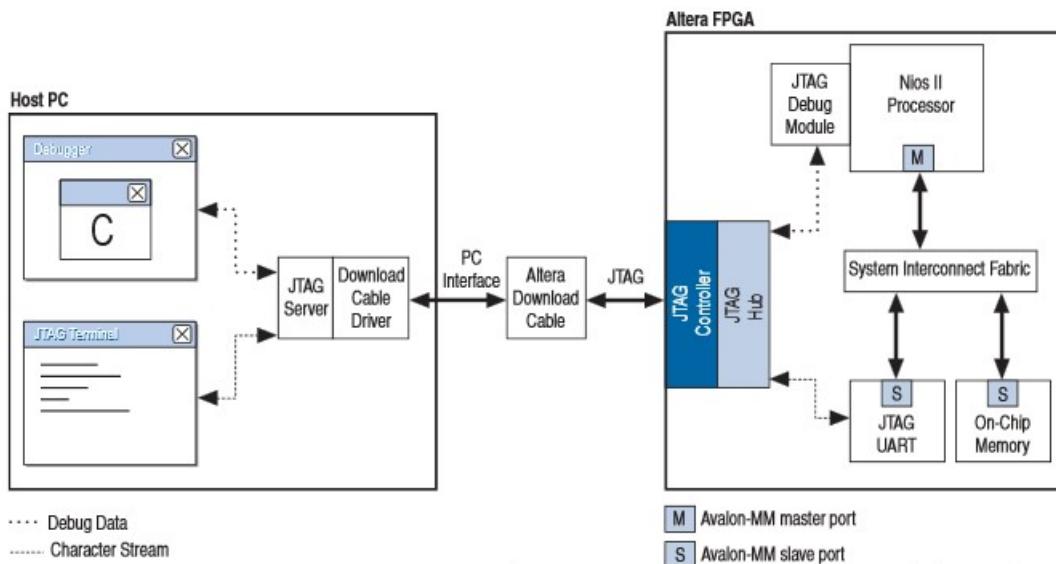
Blok ten obsługuje linie wejścia-wyjścia ogólnego przeznaczenia. Sygnały zostały podzielone na dwie 8-bitowe grupy. Pierwszą grupą są linie wejściowe, drugą wyjściowe. Sygnały te przeznaczone są do kontrolowania układów peryferyjnych. Na zdjęciu 6.15 przedstawiono blok GPIO.



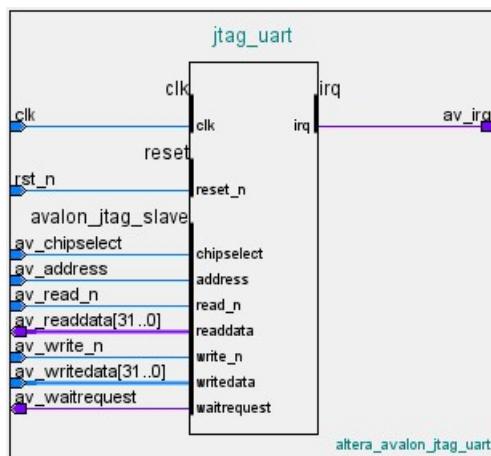
Rysunek 6.15: Blok linii wejścia-wyjścia

Blok JTAG – UART

Moduł ten przeznaczony jest do implementacji metody komunikacji systemu NIOS II z komputerem PC. Pozwala na wyeliminowanie dodatkowego portu RS232 oraz elementów elektronicznych mu towarzyszących. Blok JTAG-UART umożliwia dwustronną komunikację z terminalem RS232 komputera PC poprzez wbudowany w układ FPGA interfejs JTAG. Układ programowalny podłączony jest do komputera z wykorzystaniem kabla USB-Blaster. Idea ta przedstawiona jest na rysunku 6.16 [175]. Rozwiążanie to znacząco ułatwia inżynierowi uruchamianie aplikacji pisanej na system NIOS II. Na rysunku 6.17 widnieje moduł JTAG-UART wraz z sygnałami.



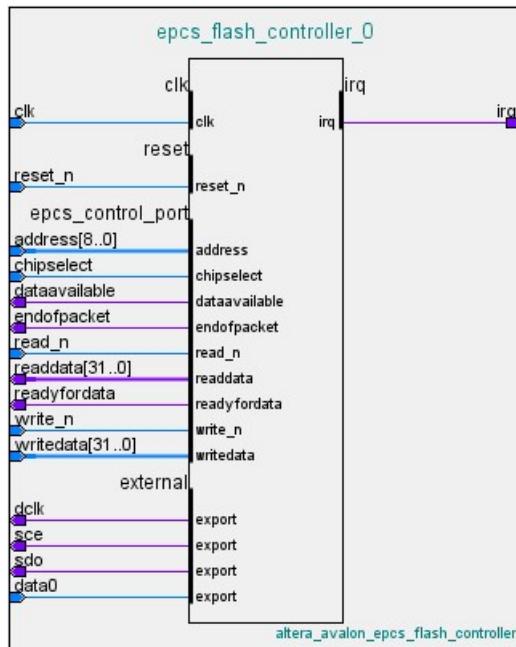
Rysunek 6.16: Koncepcja działania bloku JTAG-UART [175]



Rysunek 6.17: Blok modułu JTAG-UART

Blok EPICS Memory Controller

Moduł kontrolera EPICS (rys. 6.18) pozwala na podłączenie do procesora NIOS II zewnętrznej, nieulotnej pamięci Flash. Może ona być wykorzystana przez procesor do przechowywania danych programu równolegle z konfiguracją układu FPGA. Możliwe jest również bootowanie procesora z tej pamięci.



Rysunek 6.18: Blok kontrolera pamięci EPICS

Blok DDR SDRAM Controller

Moduł kontrolera pamięci DDR SDRAM. Wygenerowano go za pomocą funkcji DDR SDRAM Controller with ALTMEMPHY v13.0 z parametrami przedstawionymi na rysunku 6.19. Wykorzystany układ FPGA Altera Cyclone III EP3C25Q240C8N jest 8 klasy szybkości. Sygnał zegarowy CK/CK o częstotliwości 125 MHz generowany jest przez generator PLL z sygnału referencyjnego 100 MHz. Na rysunku 6.20 zaprezentowano ustawione parametry dla modułów pamięci SDRAM. 16-bitowy interfejs do pamięci podzielony jest na dwie grupy po 8 bitów każda. Układy pracują z tym samym sygnałem zegarowym i używają tego samego sygnału CKE. Wygenerowany kontroler posiada włączoną opcję Error Detection and Correction Logic. Na rysunku przedstawiono parametry czasowe wykorzystanych układów pamięci MT46V64M8TG-5B firmy Micron.

The screenshot shows the 'DDR SDRAM Controller with ALTMEMPHY' software interface. At the top, there is a logo for 'MegaCore' and navigation links for 'About' and 'Documentation'. Below the header, a breadcrumb navigation bar shows the current path: 'Memory Settings > PHY Settings > Board Settings > Controller Settings >'. The main content area is titled 'General Settings' and contains the following configuration parameters:

- Device family: Cyclone III
- Speed grade: 8
- PLL reference clock frequency: 100 MHz (10000 ps)
- Memory clock frequency: 125 MHz (8000 ps)
- Controller data rate: Full
- Local interface clock frequency: 125,0 MHz
- Local interface width: 16 bits

Below these settings, there are two sections: 'Show in 'Memory Presets' List' and 'Memory Presets'. The 'Show in 'Memory Presets' List' section contains filters for 'Parameter', 'Value', 'Memory vendor', 'Memory format', and 'Maximum memory frequency', with a 'Show All' button. The 'Memory Presets' section lists several JEDEC memory presets, with a 'Load Preset...' button. At the bottom, there is a 'Selected memory preset: JEDEC DDR-400 512Mb x8' field and a 'Modify parameters...' button. A descriptive text at the bottom states: 'Description: DDR SDRAM, 200.0MHz, 128MB, 16 bits wide, Discrete Device, CAS 3.0, 1 Chip Select'.

Rysunek 6.19: Parametry kontrolera pamięci DDR SDRAM

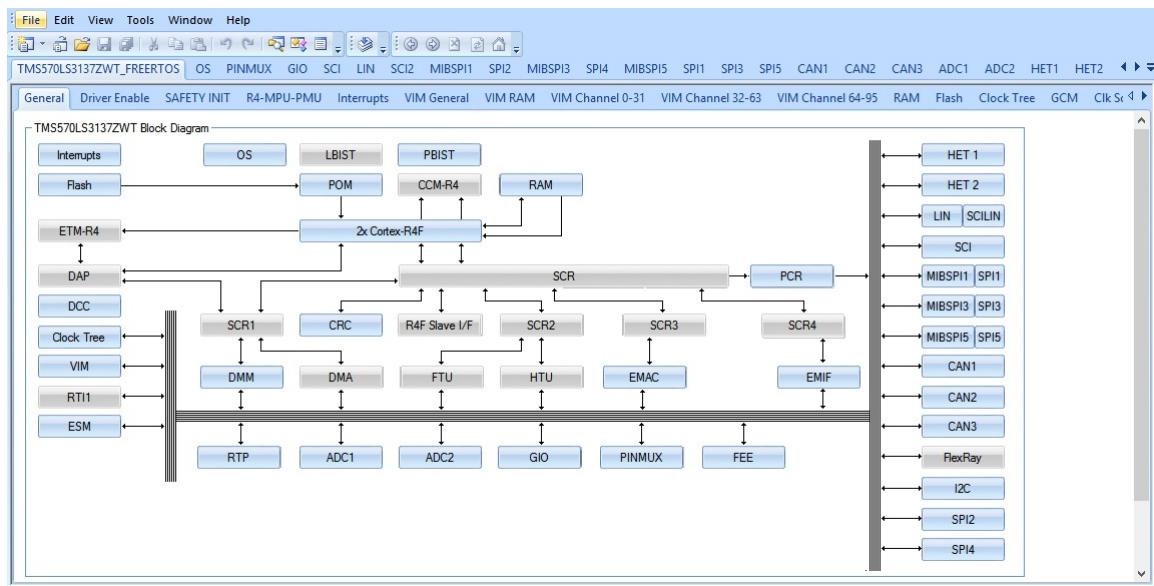
Parameters			
Parameter	Value	Units	
Active to read/write time (tRCD)	15.0	ns	
Precharge command period (tRP)	15.0	ns	
Refresh command interval (tREFI)	7.0	us	
Auto-refresh command interval (tRFC)	70.0	ns	
Write recovery time (tWR)	15.0	ns	
Write to read period (tWTR)	2	tCK	
DQ output access time from CK/CK# (tAC)	700	ps	
DQS output access time from CK/CK# (tDQSQ)	600	ps	
DQS-DQ skew for DQS and associated DQ signals (tDQSS)	400	ps	
First latching edge of DQS to associated clock edge, + or - (tDQSS)	0.28	tCK	
DQ and DM input hold time relative to DQS (tDH)	400	ps	
DQ and DM input setup time relative to DQS (tDS)	400	ps	
DQS falling edge hold time from CK (tDSH)	0.2	tCK	
DQS falling edge to CK setup time (tDSS)	0.2	tCK	
Address and command input hold time (tIH)	600	ps	
Address and command input setup time (tIS)	600	ps	
DQ hold skew factor (tQHS)	500	ps	
RAS to RAS delay time (tRRD)	10.0	ns	

Rysunek 6.20: Parametry kontrolera pamięci DDR SDRAM

6.3. Oprogramowanie mikrokontrolerów TMS570

Oprogramowanie układów TMS570 napisano w środowisku *Code Composer Studio 6.0.1* udostępnionym przez firmę Texas Instruments. Mikrokontrolery pracują pod kontrolą systemu operacyjnego czasu rzeczywistego. Zdecydowano się na system *FreeRTOS* [176] rozwijany przez *Real Time Engineers Ltd.* [177] ponieważ jest on darmowy i wspierany przez środowisko *Code Composer Studio*. Ponadto, kod napisany na *FreeRTOS'a* można łatwo przenieść na *SafeRTOS'a* [178] gdzie kluczowy nacisk postawiono na bezpieczeństwo. *SafeRTOS* firmy *Wittenstein* [179], zależności od wersji, może posiadać różne certyfikaty bezpieczeństwa w tym *DO178C* [180] przeznaczony dla przemysłu lotniczego i kosmicznego.

Przy pomocy programu *HALCoGen* [181] firmy *Texas Instruments* dokonano konfiguracji urządzeń peryferyjnych mikrokontrolera TMS570 i wygenerowano pliki źródłowe zawierające niezbędne ustawienia wraz z plikami systemu operacyjnego czasu rzeczywistego (rys. 6.21). W pliku *FreeRTOSConfig.h* ustalono podstawową konfigurację systemu operacyjnego. System taktowany jest zegarem TICK zliczającym impulsy procesora. Dlatego w pierwszej kolejności, w pliku konfiguracyjnym ustaliona została częstotliwość tego zegara. Następnie ustalono przydział pamięci RAM dla zadań systemu. Ponadto, zdefiniowane zostały priorytety wykonywanych zadań. Utworzone zostały zadania wykonywane przez system operacyjny takie jak obsługa interfejsu I2C oraz interfejsu UART.



Rysunek 6.21: Wycinek ekranu przedstawiający konfiguracje mikrokontrolera TMS570 wygenerowaną w programie *HALCoGen*

Rozdział 7

Uruchomienie i testowanie

W rozdziale opisano proces uruchamiania oraz testowania *Modułu komputera pokładowego*. Przedstawiono sekwencję włączania oraz wartości pomiarów napięć zasilających. Następnie przeprowadzono badanie systemu zabezpieczeń linii zasilających przed zbyt dużym poborem prądu. Skonfigurowano układ FPGA w celu weryfikacji poprawności pracy pamięci SDRAM. Następnie zademonstrowano podstawowy tryb działania mikrokontrolerów TMS570. Ponieważ większość najważniejszych interfejsów *Modułu komputera pokładowego* została podłączona do złącza PC104 zaprojektowano i wykonano dodatkowy moduł elektroniczny który ułatwił proces uruchamiania. Opis dodatkowego modułu *CubeSat Motherboard* znajduje się w dodatku A.4.

7.1. Badanie generacji napięć zasilających

Po podłączeniu zasilania 12V na *Module komputera pokładowego* najpierw zaczyna pracować przetwornica TPS62160. Generuje ona napięcie 3.3V, z którego jest zasilany mikrokontroler Atmega128. Pozostałe przetwornice energii są w tym czasie wyłączone. Mikrokontroler, po włączeniu oraz inicjalizacji, wywołuje funkcję, która po kolei uruchamia przetwornice DC/DC. Sekwencja we włączaniu napięć zasilających została zastosowana ze względu na zmniejszony pobór prądu z zewnętrznego źródła co w przypadku satelity zasilanej baterijnie ma znaczenie.

W celu sprawdzenia poprawności wartości generowanych napięć zasilających dla podsistemów komputera pokładowego wykonano pomiary, które zestawiono w tabeli 7.1. Po analizie pomiarów w tab. 7.1 stwierdzono, iż wartości napięć są poprawne i mieszczą się granicach zaczerpniętych z dokumentacji technicznych. Pomiary wykonano przy pomocy multimetru Fluke 289 firmy Fluke.

Nazwa	Nominalne napięcie [V]	Zmierzzone napięcie [V]
VCC_3V3_SUP	3.3	3.293
VCC_3V3	3.3	3.292
VCC_2V5	2.5	2.506
VCC_1V2_TMS_1	1.2	1.193
VCC_1V2_TMS_2	1.2	1.193
VCC_1V2_FPGA	1.2	1.205
VCC_1V5	1.5	1.494

Tablica 7.1: Wartości napięć zasilających modułu komputera pokładowego

7.2. Uruchamianie mikrokontrolera ATmega128A

Mikrokontroler ATmega128A zaprogramowano programatorem *Atmel-ICE* [182] firmy *Atmel*.

7.2.1. Badanie czasu reakcji na zabezpieczenia systemu zasilania

W celu przeciwdziałaniu trwałemu uszkodzeniu układów scalonych w skutek wystąpienia efektu Single Event Latchup, każda linia zasilająca została wyposażona w zabezpieczenie przeciwko nadmiernemu poborowi energii. W *Module komputera pokładowego* wykorzystano trzy koncepcje monitorowania prądu oraz odłączenia zasilania w przypadku przekroczenia zadeklarowanych wartości. Zastosowano następujące realizacje:

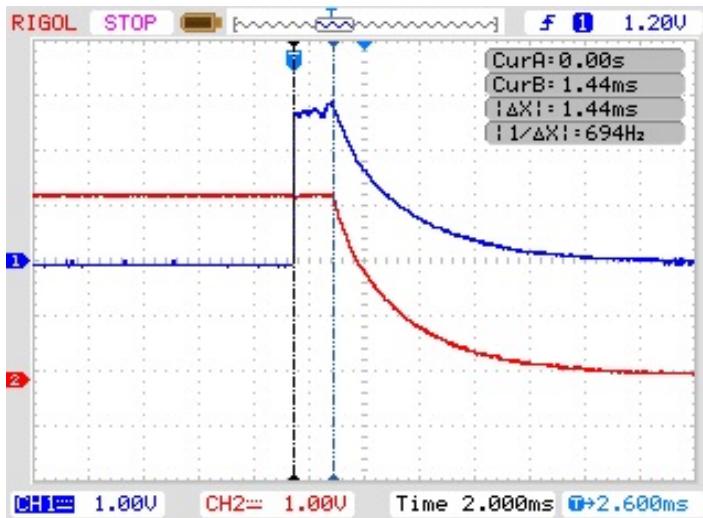
- **TPS2553** - układ ten mierzy pobór prądu i porównuje go wartością ustaloną przez zewnętrzny rezystor. Przekroczenie wyznaczonego limitu powoduje przerwanie obwodu dostarczającego napięcie wyjściowe.
- **INA219 + mikrokontroler** - w tym przypadku mikrokontroler, poprzez interfejs I2C, cyklicznie odczytuje prąd zmierzony przez układ INA219. Atmega128 porównuje odczyt z zapisaną w pamięci wewnętrznej wartością progową i w przypadku przekroczenia limitu wyłącza przetwornicę DC/DC.
- **INA209 + mikrokontroler** - układ INA209 mierzy napięcie na rezystorze bocznikującym, następnie wewnętrzny komparator porównuje je z ustalonym napięciem progowym i w przypadku przekroczenia limitu wystawia stan wysoki na wyjściu *Critical*. Linia ta podłączona jest do wejścia zewnętrznego przerwania mikrokontrolera. Pojawienie się na niej stanu wysokiego powoduje natychmiastowe przejście programu do funkcji obsługi przerwania w której wyłącza daną przetwornicę DC/DC.

Bardzo ważne jest aby jak najszybciej wykryć i zareagować na efekt Single Event Latchup ponieważ zwiększa to szanse na uniknięcie trwałego uszkodzenia struktur krzemowych układów scalonych. W związku z tym przeprowadzono badania czasu reakcji trzech zastosowanych koncepcji polegające na pomiarze czasu od momentu wystąpienia zwiększonego poboru prądu na linii zasilającej do chwili zaniku napięcia wyjściowego danej przetwornicy.

Proces zwiększenia konsumpcji prądu polegał na dołączeniu do szyny zasilającej rezystora mocy o wartości 5 omów. Pomiarów dokonano oscyloskopem DS1052E firmy *Rigol*. Jedną elektrodę rezystora dołączona do masy, do drugiej natomiast podłączono sondę oscylomkopu. Kolejną sondę podpięto do badanej linii zasilającej. Pobór prądu zwiększano poprzez podłączenie wolnej elektrody rezystora do dodatniej szyny zasilającej. Moment narastania napięcia na rezystorze świadczy o tym, iż przez układ zaczyna płynąć większy prąd.

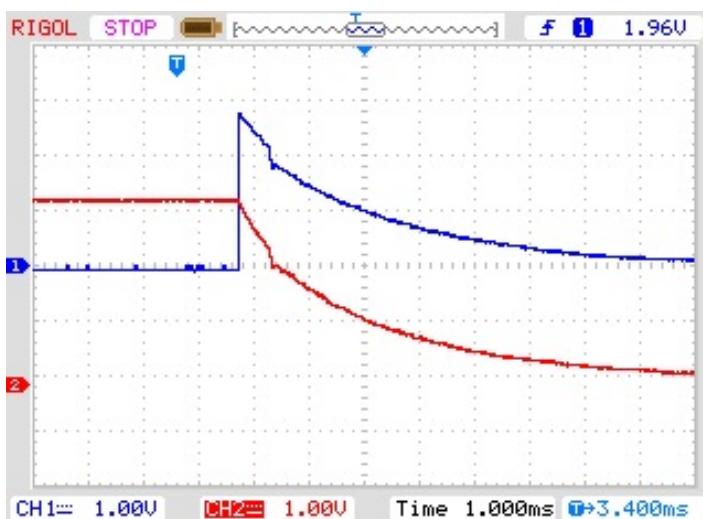
Rysunek 7.1 przedstawia wynik badania czasu reakcji zabezpieczenia w postaci układu INA219 i mikrokontrolera odczytującego pomiar prądu. Kolorem niebieskim oznaczono napięcie na dołączanym rezystorze a kolor czerwony odzwierciedla napięcie na linii zasilającej. Z oscylogramu można odczytać, iż czas liczony od momentu przekroczenia ustalonego limitu pobieranego prądu do wyłączenia przetwornicy wynosi w tym przypadku 1.44 ms.

Rysunek 7.2 przedstawia wynik badania czasu reakcji zabezpieczenia w postaci układu mikrokontrolera którego wejście zewnętrzne przerwania podłączono do wyjścia *Critical* układu INA209. Kolorem niebieskim oznaczono napięcie na dołączanym rezystorze a kolor czerwony odzwierciedla napięcie na linii zasilającej. Z oscylogramu można odczytać, iż układ



Rysunek 7.1: Oscylogram przedstawiający czas reakcji systemu zabezpieczeń

praktycznie natychmiast reaguje na przekroczenie limitu pobieranego prądu i wyłącza daną przetwornicę.



Rysunek 7.2: Oscylogram przedstawiający czas reakcji systemu zabezpieczeń

Niestety z powodu błędnie zaprojektowanego obwodu drukowanego nie udało się zbadać zabezpieczenia w postaci układu TPS2553. Jednak z dokumentacji technicznej [156] tego układu odczytano, iż jego czas reakcji na przekroczenie limitu pobieranego prądu wynosi 2 us.

Na oscylogramach widać też proces rozładowywania kondensatorów odsprzęgających linie zasilające. W kolejnej rewizji sprzętowej *modułu komputera pokładowego* zostanie dodany dodatkowy tranzystor mocy który w momencie zadziałania zabezpieczenia prądowego zewrze szynę zasilającą do masy i tym samym przyspieszy rozładowanie kondensatorów.

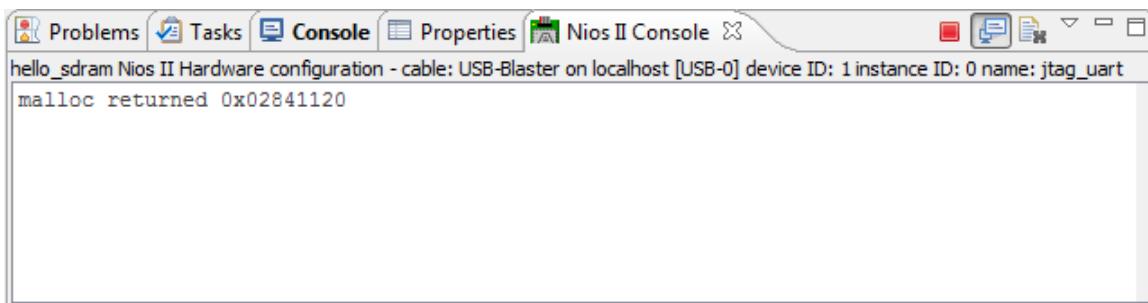
7.3. Uruchamianie układu FPGA

Układ programowalny skonfigurowano przy pomocy programatora *USB-Blaster II* [183] który podłączono do złącza *JTAG* płytki *komputera pokładowego*. Wykorzystując narzędzie *Programmer* wchodzące w skład oprogramowania *Quartus II 13.1* wgrano konfiguracje do układu EP3C25Q240. Następnie w dedykowanym kompilatorze procesora *NIOS II*, bazującym na środowisku *Eclipse*, napisano prosty program testujący pamięć SDRAM.

7.3.1. Testy poprawności pracy pamięci SDRAM

Pakiet *Quartus II 13.1* wyposażony jest w narzędzie pozwalające na wykonanie testów zewnętrznej pamięci SDRAM podłączonej do układu programowalnego. Niestety nie wspiera ono wykorzystanego układu EP3C25Q240. Dlatego też, w programie *NIOS II Eclipse* napisano program który w prosty sposób testuje dołączoną pamięć. Oprogramowanie wykorzystuje funkcję *malloc()* która dynamicznie alokuje pamięć. W przypadku gdy funkcja poprawnie zarezerwowała miejsce w pamięci zwraca wskaźnik na ten blok. Wynikiem funkcji jest zero kiedy podczas wykonywania *malloc()* wystąpił błąd.

W celu wizualizacji działania *malloc()* posłużyono się funkcją *printf()*. Metoda ta, wykorzystując blok *JTAG - UART*, wypisuje tekst na konsoli UART wbudowanej w środowisko *NIOS II Eclipse*. Funkcję alokującą pamięć wywołano z argumentem $1024 * 1024 * 100$, rezultat operacji widoczny jest na rysunku 7.3.



The screenshot shows the Eclipse IDE interface with the 'Nios II Console' tab selected. The console window displays the following text:

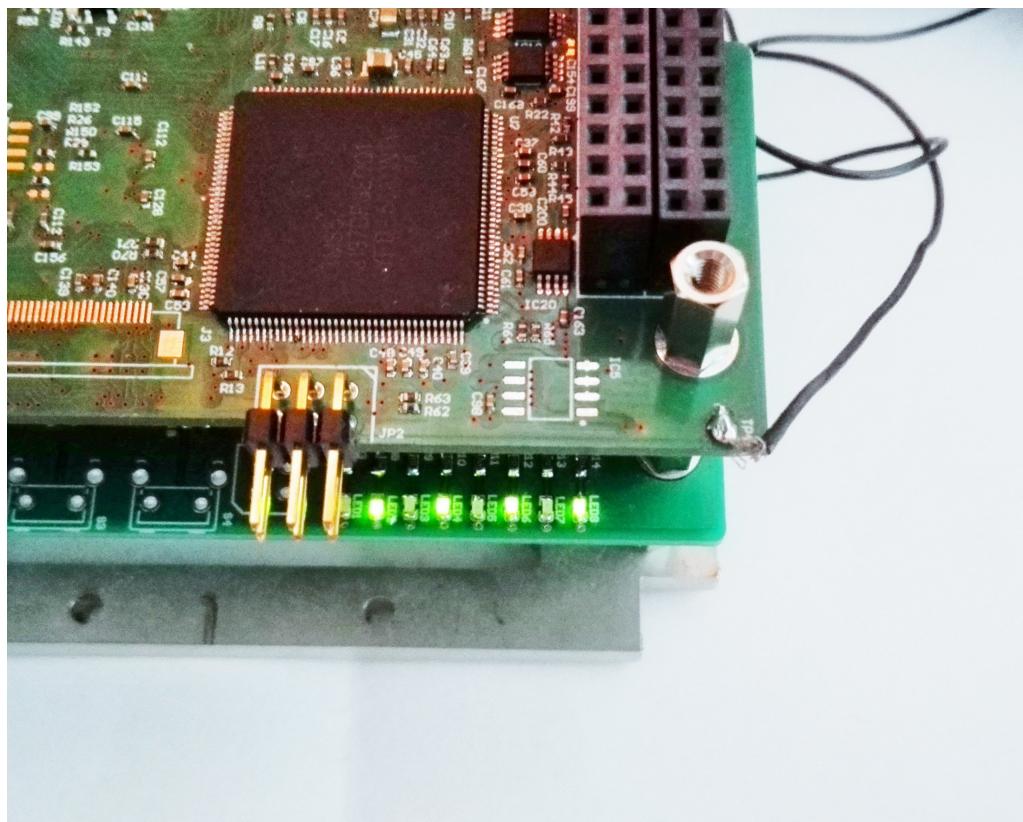
```
hello_sdram Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0 name: jtag_uart
malloc returned 0x02841120
```

Rysunek 7.3: Rezultat działania funkcji *malloc()*.

Funkcja zwróciła wskaźnik o wartość *0x02841120* co świadczy o tym, iż pamięć została poprawnie alokowana.

7.4. Uruchamianie układów TMS570

Układy TMS570LS3137 zostały zaprogramowane przy pomocy programatora J-Link [184] firmy SEGGER [185] który podłączono do odpowiedniego złącza JTAG znajdującego się na płytce *komputera pokładowego*. Program testujący, bazujący na systemie operacyjnym czasu rzeczywistego, komunikuje się poprzez interfejs I2C z układem PCA9555 [186] znajdującym się na module *CubeSat Motherboard*. Napisana aplikacja zapala i gasi diody LED w określonej sekwencji. Rysunek 7.4 przedstawia wysterowane diody LED.



Rysunek 7.4: Zapalone diody LED na płytce *CubeSat Motherboard*

Rozdział 8

Wnioski końcowe

W ramach pracy magisterskiej w pełni zrealizowano założone cele. Skonstruowano *Moduł komputera pokładowego do satelity CubeSat* z uwzględnieniem założeń opisanych w rozdziale 4. Zbudowane urządzenie umożliwia zarządzanie misją satelity, sterowanie modułami pokładowymi. Ponadto, wyposażone jest w szereg zabezpieczeń zwiększających czas bezawaryjnej pracy w przestrzeni kosmicznej.

Dzięki zaimplementowaniu równoległej architektury łączącej mikrokontrolery oraz układ FPGA umożliwiono pracę modułu w następujących trybach:

- komputer pokładowy,
- Star Tracker,
- komputer pokładowy i Star Tracker.

W tabeli 8.1 przedstawiono uzyskane parametry techniczne modułu komputera pokładowego.

Dalsze prace nad *Komputerem pokładowym* będą obejmować:

- rozwijanie oprogramowania układu FPGA w tym akwizycję danych z sensorów obrazu,
- implementację oprogramowania zarządzającego satelitą,
- implementację algorytmu Star Tracker oraz wybranego algorytmu kompresji obrazu,
- integrację z innymi modułami CubeSat przy współpracy z zespołem projektującym satelitę PW-Sat 2,
- wykonanie testów zastosowanych rozwiązań w środowisku radiacyjnym.

Zaprojektowany *Komputer pokładowy* umożliwia dołączenie różnego typu urządzeń elektronicznych oraz implementację szerokiej gamy algorytmów pozwalających na wykonanie praktycznie dowolnej misji satelity CubeSat. W toku przyszłego rozwoju systemu, powstaną dwie prace dyplomowe magisterskie. Ponadto, dalsze prace nad *Komputerem pokładowym* będą kontynuowane na studiach doktoranckich.

Wykonany moduł komputera pokładowego	
Architektura	Dwa niezależne mikrokontrolery z rdzeniem Cortex-R4F
Częstotliwość	180 MHz
Układ FPGA	Cyclone III EP3C25Q240
Pamięć RAM	128 MB
Pamięć Flash	16 MB + 2 MB
Pamięć FRAM	256 KB
Zegar RTC	TAK
Interfejs CAN	1x
Interfejs I2C	2x
Interfejs SpaceWire	możliwość implementacji
Interfejs sensora obrazu	2x (kamera Star Tracker oraz dodatkowa kamera)
Rozwiązania zwiększające odporność na promieniowanie	TAK

Tablica 8.1: Parametry techniczne komputera pokładowego

Dodatek A

Dodatki

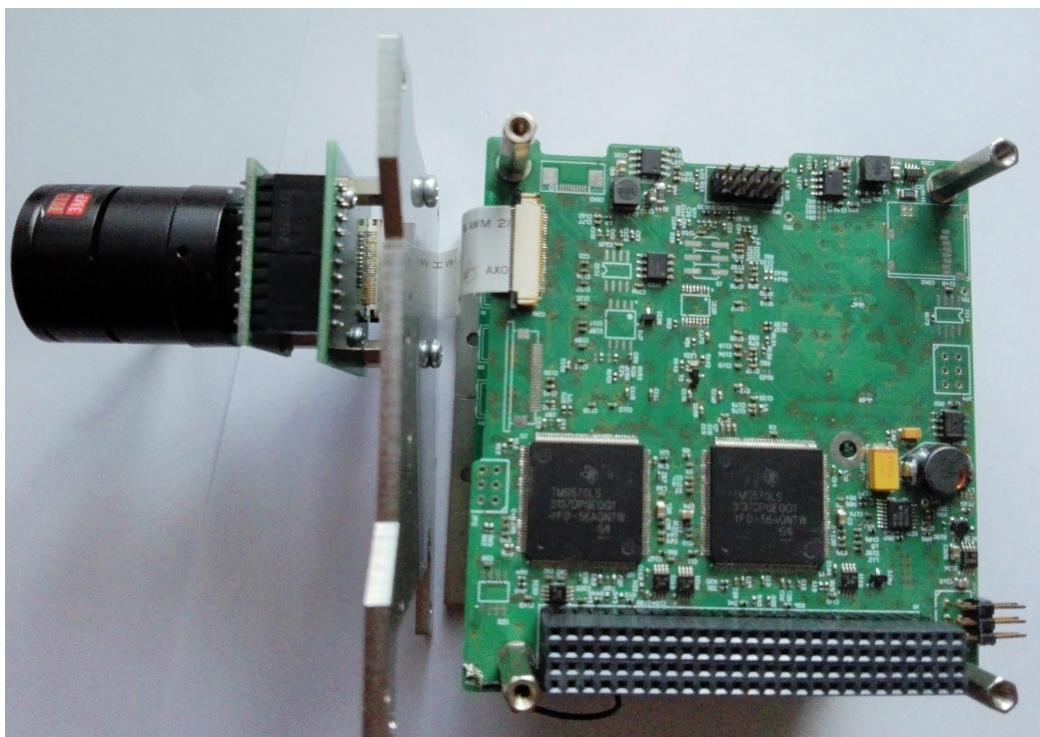
A.1. Płyta CD

Opis plików znajdujących się na płycie CD:

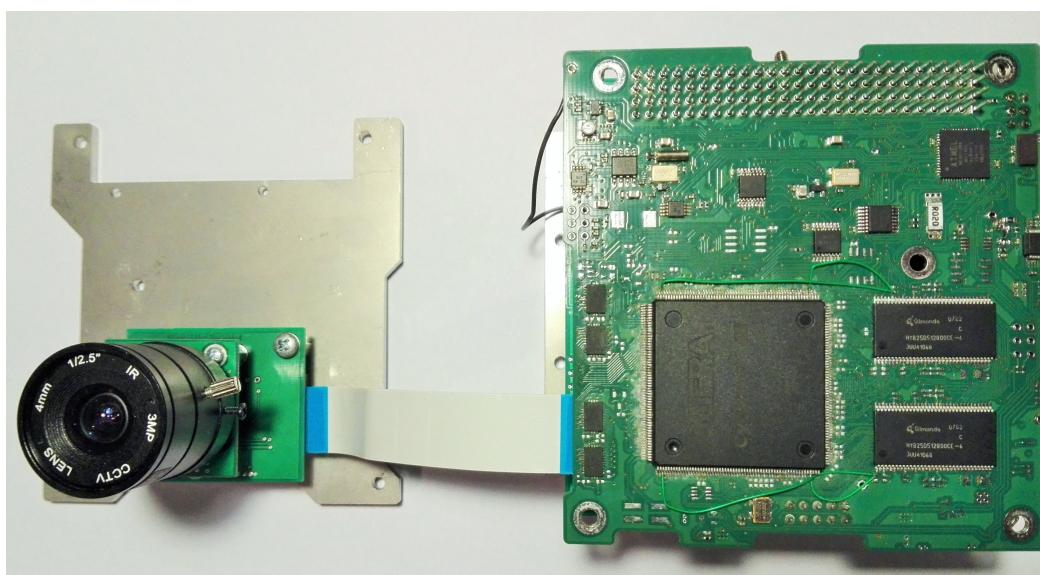
- **cubesat_computer.pdf** - schemat elektryczny *Komputera pokładowego*;
- **camera_module.pdf** - schemat modułu kamery Star Tracker;

A.2. Wielowarstwowy obwód drukowany

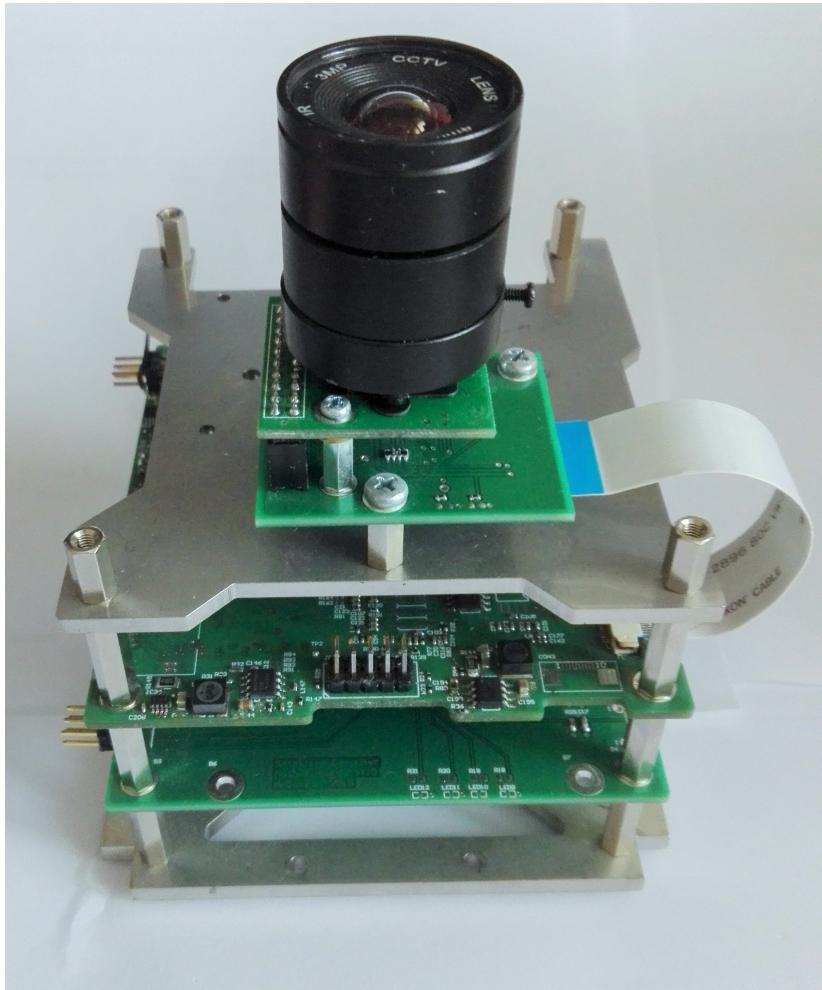
Poniższe fotografie przedstawiają zaprojektowany wielowarstwowy obwód drukowany dla *Komputera pokładowego*.



Rysunek A.1: Zaprojektowana płyta PCB (warstwa góra - *TOP*)



Rysunek A.2: Zaprojektowana płyta PCB (warstwa dolna - *BOTTOM*)



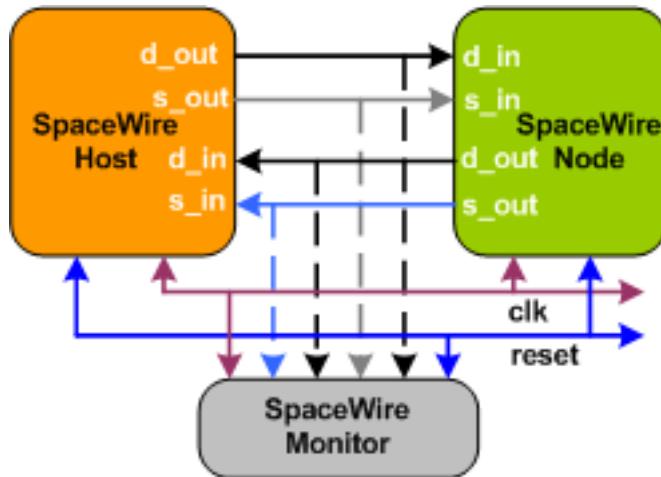
Rysunek A.3: Zdjęcie wykonanego *modułu komputera pokładowego z zamontowaną kamerą*

A.3. Interfejs SpaceWire oraz CubeSat Space Protocol

A.3.1. SpaceWire

SpaceWire jest siecią komunikacyjną przeznaczoną do zastosowań na pokładach satelitów bazującą na standardzie IEEE 1355 [190]. SpaceWire rozwijana jest przez Europejską Agencję Kosmiczną przy współpracy z międzynarodowymi agencjami kosmicznymi takimi jak NASA, JAXA oraz RKA.

Interfejs ten wykorzystuje komunikację kablową typu punkt - punkt aby zapewnić szybką komunikację pomiędzy modułami satelity. Specyfikacji SpaceWire definiuje warstwę fizyczną, elektryczną oraz warstwę protokołu. Magistrala ta może pracować z prędkościami od 2 Mbps do 400 Mbps na dystansie do 10 metrów [190].

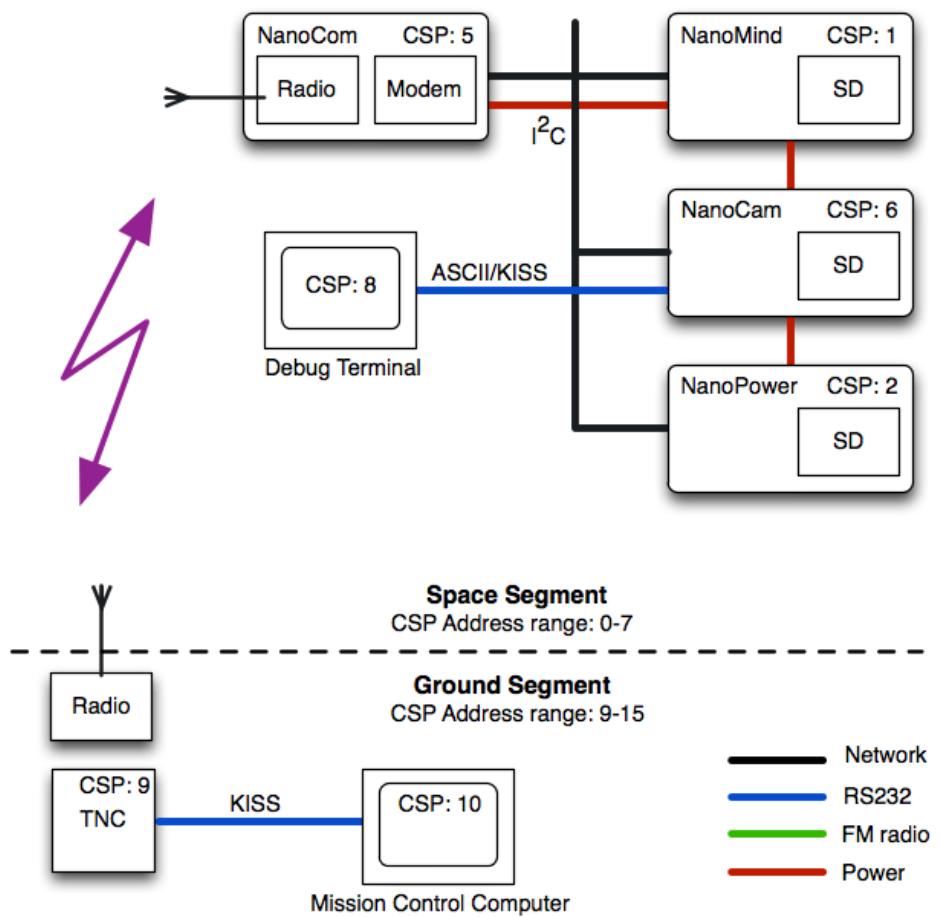


Rysunek A.4: Interfejs SpaceWire [191]

A.3.2. CubeSat Space Protocol

CubeSat Space Protocol (CSP) [194] jest nieskomplikowanym protokołem warstwy sieciowej zaprojektowanym dla satelitów CubeSat. Bazuje on na 32-bitowym nagłówku który zawiera informacje o warstwie sieci i warstwie transportowej. Implementacja protokołu została zaprojektowana z myślą o "małych systemach wbudowanych opartych o 8-bitowe oraz 32-bitowe mikrokontrolery AVR firmy Atmel. Została ona napisana w języku C dzięki czemu może zostać w łatwy sposób przeniesiona na inne platformy. Obecnie CubeSat Space Protocol posiada port na *FreeRTOS* i *POSIX*.

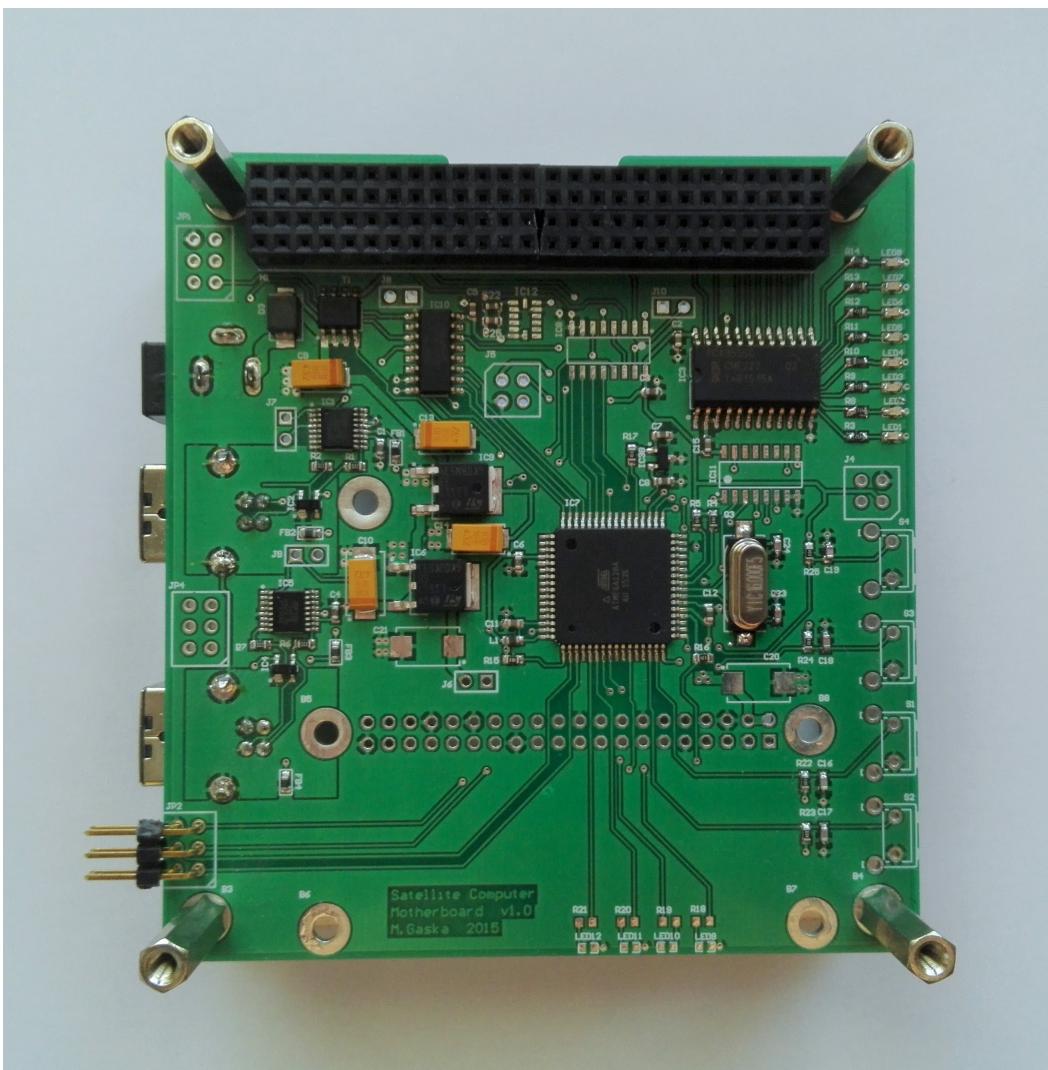
Protokół ten wspiera kilka interfejsów warstwy fizycznej. Od wersji 1.1 są to: CAN, I2C, RS-232 i TCP/IP. Dodanie nowych magistral wymaga jedynie zdefiniowania funkcji transmitemającej pakiet oraz metody przekazującej odebrane dane do stosu protokołu. Idea CubeSat Space Protocol została przedstawiona na rysunku A.6.



Rysunek A.5: Schemat modułów połączonych protokołem CSP [192]

A.4. Zaprojektowana płytka CubeSat Motherboard

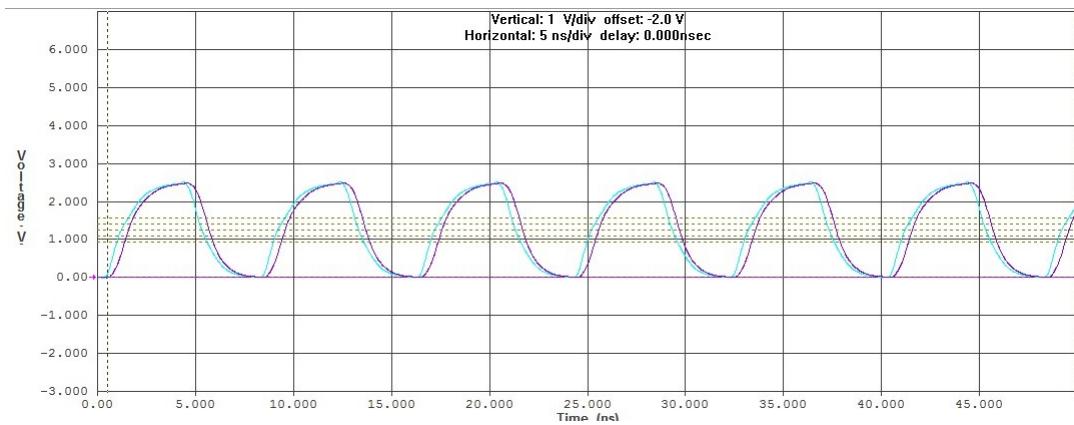
W celu ułatwienia uruchamiania i testowania *Modułu komputera pokładowego* wykonano dodatkową płytke *CubeSat Motherboard*. Wyposażona jest ona w dwa konwertery USB - UART FT230X [187] firmy *FTDI* [188] które poprzez multipleksery 74HC4051 [189] podłączone są do złącza PC104 tak aby umożliwić komunikację z interfejsami UART *Modułu komputera pokładowego*. Multiplekserami steruje mikrokontroler ATmega128. Płytkę *CubeSat Motherboard* posiada także ekspander portów wejścia/wyjścia PCA9555 dołączony do magistrali I2C. System *komputera pokładowego* dzięki czemu możliwe jest testowania transmisji. Do portów wejścia/wyjścia układu PCA9555 podłączono diody LED.



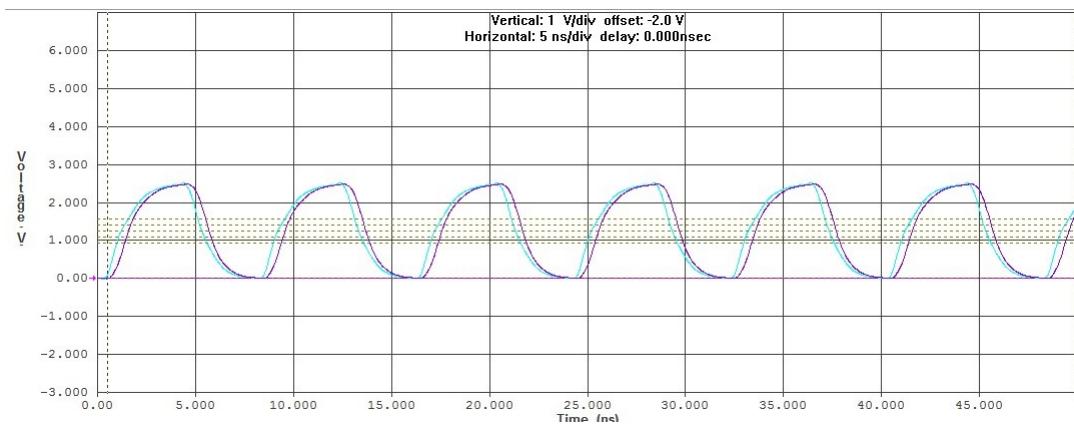
Rysunek A.6: Zdjęcie modułu *CubeSat Motherboard*

A.5. Symulacje SI

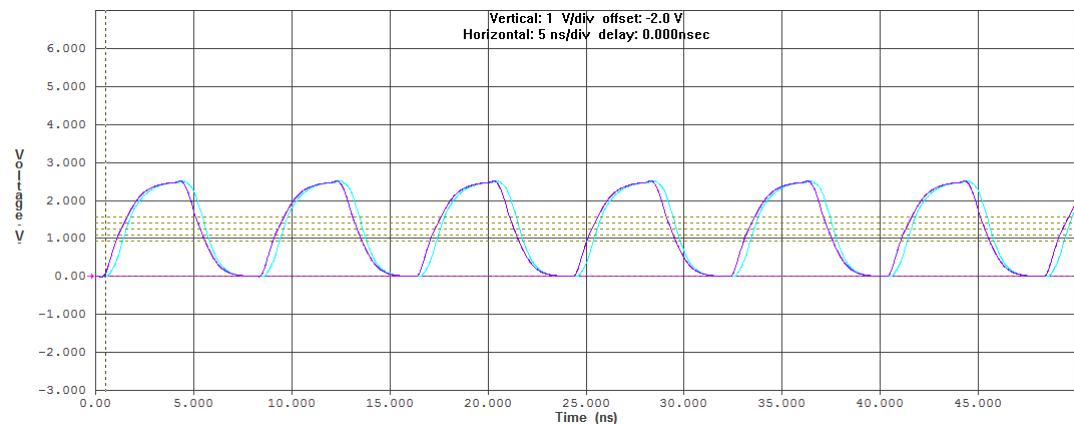
W niniejszym dodatku zaprezentowano wyniki przeprowadzonych badań integralności sygnałowej pamięci SDRAM. Symulacje wykonano w oparciu o oprogramowanie HyperLynx. Pliki IBIS dla wykorzystanych układów scalonych zostały dostarczone przez producentów [195, 196].



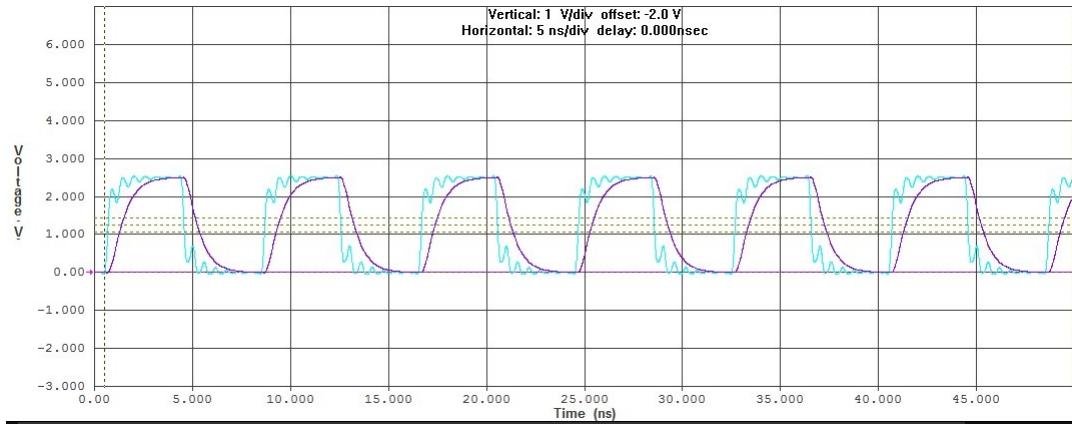
Rysunek A.7: Symulacja sygnału DQ w trybie WRITE



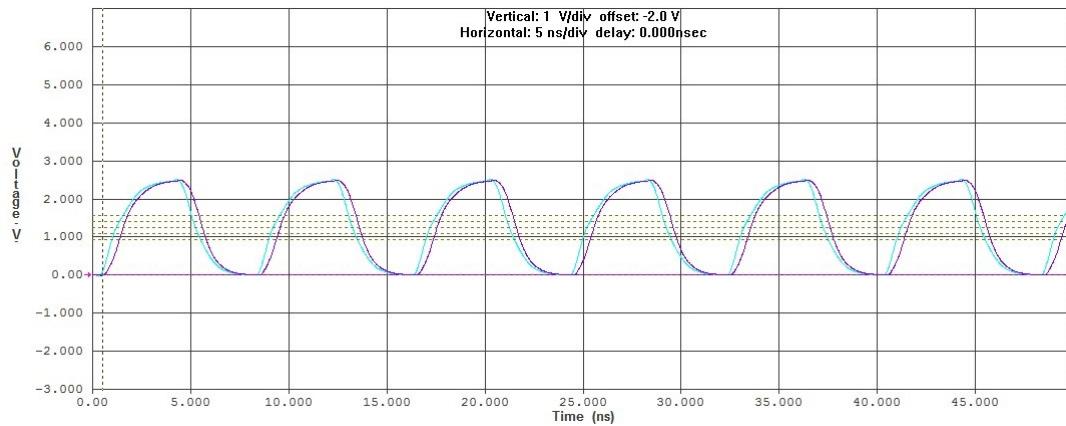
Rysunek A.8: Symulacja sygnału DQ w trybie READ



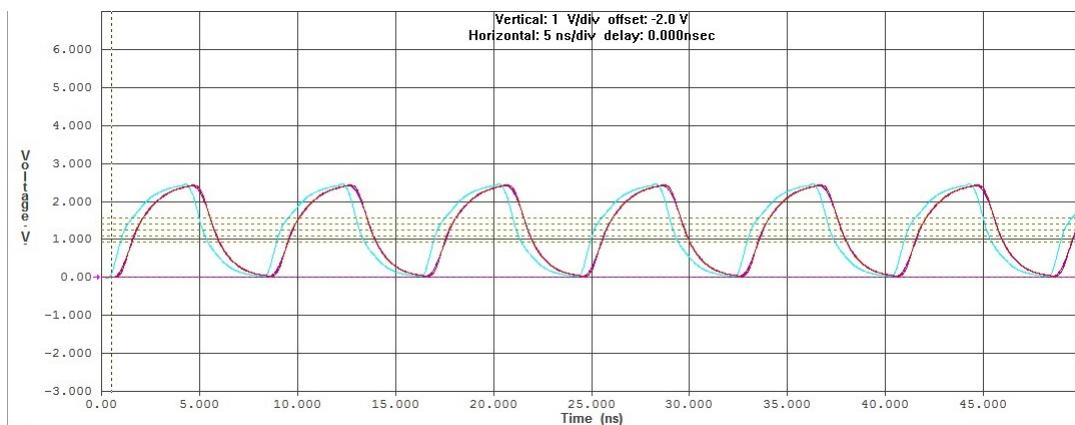
Rysunek A.9: Symulacja sygnału DQS w trybie WRITE



Rysunek A.10: Symulacja sygnału DQS w trybie READ



Rysunek A.11: Symulacja sygnału DM



Rysunek A.12: Symulacja sygnału WE

Bibliografia

- [1] Wikipedia - Sputnik 1, https://pl.wikipedia.org/wiki/Sputnik_1, dostęp 22.06.2016
- [2] Sputnik 1, <http://fineartamerica.com/featured/3-sputnik-1-satellite-detlev-van-ravenswaay.html>, dostęp 22.06.2016
- [3] Wikipedia - Satellite , <https://en.wikipedia.org/wiki/Satellite>, dostęp 22.06.2016
- [4] ESA - Robotic Exploration of Mars, <http://exploration.esa.int/mars/>, dostęp 22.06.2016
- [5] Comparison satellite navigation orbits , https://commons.wikimedia.org/wiki/File:Comparison_satellite_navigation_orbits.svg, dostęp 22.06.2016
- [6] How Many Satellites are in Space? - Universe Today , <http://www.universetoday.com/42198/how-many-satellites-in-space/>, dostęp 22.06.2016
- [7] H. Sohier, J. L. Farges and H. Piet-Lahanier, "Adaptation of the Morris method to multi-dimensional factors for air-launch-to-orbit separation," 2014 IEEE Aerospace Conference, Big Sky, MT, 2014, pp. 1-14.
- [8] Wikipedia - CubeSat , <https://en.wikipedia.org/wiki/CubeSat>, dostęp 22.06.2016
- [9] CubeSat First-MOVE, <https://www.tum.de/en/studies/studinews/issue-052012/show/article/30451/>, dostęp 22.06.2016
- [10] CubeSat Design Specification Rev. 13, http://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/56e9b62337013b6c063a655a/1458157095454/cds_rev13_final2.pdf, dostęp 22.06.2016
- [11] Mars Cube One (MarCO), <http://www.jpl.nasa.gov/cubesat/missions/marco.php>, dostęp 22.06.2016
- [12] NanoMind A712D, <http://gomspace.com/?p=products-a712c>, dostęp 22.06.2016
- [13] Gomspace - Cubesat and nano-satellite solutions , <http://gomspace.com/>, dostęp 22.06.2016
- [14] NanoPower P31u , <http://gomspace.com/?p=products-p31u>, dostęp 22.06.2016
- [15] NanoAvionics Communication Module, <http://n-avionics.com/command-service-modules>, dostęp 22.06.2016

- [16] NanoAvionics, <http://n-avionics.com/>, dostęp 22.06.2016
- [17] ISIS Magnetorquer Board, <http://www.isispace.nl/product/isis-magnetorquer-board/>, dostęp 22.06.2016
- [18] ISIS-Innovative Solutions In Space , <http://www.isispace.nl/>, dostęp 22.06.2016
- [19] MAI-400 Reaction Wheel, <http://maiaero.com/products/mai400alacarte/mai-400-reaction-wheel/>, dostęp 22.06.2016
- [20] Maryland Aerospace Inc, <http://maiaero.com/>, dostęp 22.06.2016
- [21] The Sinclair Interplanetary ST-16 Star Tracker, <http://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=3074&context=smallsat>, dostęp 22.06.2016
- [22] Development of star tracker system for accurate estimation of spacecraft attitude, <http://hdl.handle.net/10945/4335>, dostęp 22.06.2016
- [23] Hipparcos, <http://www.cosmos.esa.int/web/hipparcos>, dostęp 22.06.2016
- [24] CU Aerospace - Propulsion for Cubesats, <http://www.cuaerospace.com/Products/SmallSatellitePropulsion/TabId/195/ArtMID/737/ArticleID/169/Propulsion-for-Cubesats-PUC-%E2%80%93-A-Smart-Robust-Propulsion-System-for-CubeSats.aspx>, dostęp 22.06.2016
- [25] Aerojet Rocketdyne - CubeSat Modular Propulsion Systems, <http://www.rocket.com/cubesat>, dostęp 22.06.2016
- [26] Wikipedia - Hydrazyna, <https://pl.wikipedia.org/wiki/Hydrazyna>, dostęp 22.06.2016
- [27] Green Propellant Infusion Mission, https://en.wikipedia.org/wiki/Green_Propellant_Infusion_Mission, dostęp 22.06.2016
- [28] N. Turan, O. Korkmaz and M. Celik, "Investigation of the effect of hollow cathode neutralizer location on hall effect thruster efficiency,"*Recent Advances in Space Technologies (RAST)*, 2015 7th International Conference on, Istanbul, 2015, pp. 599-604.
- [29] B. Yavuz, E. Turkoz and M. Celik, "Prototype design and manufacturing method of an 8 cm diameter RF ion thruster,"*Recent Advances in Space Technologies (RAST)*, 2013 6th International Conference on, Istanbul, 2013, pp. 619-624.
- [30] R. Vasantharaj, D. Sureshbabu and D. Kumar, "Advanced pulsed plasma thruster for satellites orbit control,"*Recent Advances in Space Technology Services and Climate Change (RSTSCC)*, 2010, Chennai, 2010, pp. 111-113.
- [31] V. Hruby et al., "Colloid thrusters for the new millennium, ST7 DRS mission,"*Aerospace Conference, 2004. Proceedings. 2004 IEEE*, 2004, pp. 213 Vol.1.
- [32] M. Mihailovic, T. V. Mathew, J. F. Creemer, B. T. C. Zandbergen and P. M. Sarro, "MEMS silicon-based resistojet micro-thruster for attitude control of nano-satellites,"*2011 16th International Solid-State Sensors*, 2011, pp. 262-265.

- [33] PULSED PLASMA THRUSTERS, http://www.busek.com/technologies__ppt.htm, dostęp 22.06.2016
- [34] C. R. McBryde and E. G. Lightsey, A star tracker design for CubeSats, Aerospace Conference, 2012 IEEE, Big Sky, MT, 2012, pp. 1-14.
- [35] Solar Sail for spacecraft, <http://sail.planetary.org/>, dostęp 22.06.2016
- [36] SEBROS stopy aluminium, [http://www.sebros.eu/aluminium/aluminium-EN-AW-5005A-ISO:-AlMg1\(C\)-EN:-AW-AlMg1\(C\)-PN:-PA-43-DIN:---wnr:--/](http://www.sebros.eu/aluminium/aluminium-EN-AW-5005A-ISO:-AlMg1(C)-EN:-AW-AlMg1(C)-PN:-PA-43-DIN:---wnr:--/), dostęp 22.06.2016
- [37] CubeSat Concept and the Provision of Deployer Services, <http://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/cubesat-concept>, dostęp 22.06.2016
- [38] Cubesat mechanical structure, http://www.wow.com/wiki/CubeSat#cite_note-23, dostęp 22.06.2016
- [39] Pumpkin, Inc., <http://www.cubesatkit.com/>, dostęp 22.06.2016
- [40] Cubesat Size Comparison, <http://www.nsr.com/news-resources/the-bottom-line/mass-challenge-for-cubesats>, dostęp 22.06.2016
- [41] P-POD Mk III, https://fenix.tecnico.ulisboa.pt/downloadFile/395144736837/MScThesis-JFerreira_vFinal.pdf, dostęp 22.06.2016
- [42] ISIS On Board Computer, http://www.cubesatshop.com/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=119&category_id=8&option=com_virtuemart&Itemid=75&vmcchk=1&Itemid=75, dostęp 22.06.2016
- [43] NanoMind A712D, <http://gomspace.com/?p=products-a712c>, dostęp 22.06.2016
- [44] Cube Computer, http://www.cubesatshop.com/index.php?page=shop.product_details&flypage=flypage.tpl&product_id=106, dostęp 22.06.2016
- [45] Nanosatellite On Board Computer, <https://www.clyde.space/products/7-nanosatellite-on-board-computer-gps-bundle>, dostęp 22.06.2016
- [46] MAI-SS Space Sextant, http://www.cubesatshop.com/index.php?page=shop.product_details&product_id=130&flypage=flypage.tpl&pop=0&option=com_virtuemart&Itemid=65, dostęp 22.06.2016
- [47] GOS CubeSat OBC, <http://www.orbitalsystems.de/produkte/gos-cubesat-obc/?lang=en>, dostęp 22.06.2016
- [48] SatBus 1C0, <http://n-avionics.com/flight-computers>, dostęp 22.06.2016
- [49] Stephanie Galli, Mission Design for the CubeSat OUFTI-1, Master's Thesis, University of Liège, Liège, Belgium, 2008.
- [50] Edouard Perez, VEGA launch vehicle User's Manual, ARIANESPACE, March, 2006.

- [51] S. Cakaj, B. Kamo and K. Malaric, "The nodal regression simulation for low Earth Sun synchronized orbits," Telecommunications (ConTEL), 2013 12th International Conference on, Zagreb, 2013, pp. 59-64.
- [52] F. M. Smith, K. D. Smith and W. L. Brown, "Solar cells for communication satellites in the van allen belt," in Radio Engineers, Journal of the British Institution of, vol. 22, no. 2, pp. 161-169, August 1961.
- [53] Kurt Anderson. Low-cost, radiation-tolerant, on-board processing solution. IEEE Aerospace Conference, pages 1-8, March 2005.
- [54] N. Battezzati, F. Decuzzi, L. Sterpone, and M. Violante. Soft Errors in Flash-based FPGAs: Analysis methodologies and first results. International Conference on Field Programmable Logic and Applications, pages 723 - 724, August-September 2009.
- [55] J. L. Kaschmitter, D. L. Shaeer, and N. J. Colella. Operation of commercial R3000 processors in the LEO space environment. IEEE Transactions on Nuclear Science, 38(6):14151420, December 1991.
- [56] M.M.Ibrahim, A.M.Tobal, M.Y.E. Nahas, and M.K. Refai. FPGA-based On-Board Computer for LEO satellites. IEEE International Conference on Space Science and Communication (IconSpace), pages 314 - 319, July 2011.
- [57] Brendan Bridgford, Carl Carmichael, and Chen Wei Tseng. Single-Event Upset Mitigation Selection Guide. XAPP987 v1.0, Xilinx, March 2008.
- [58] Gregory R. Allen and Gary M. Swift. Single event effects test results for advanced field programmable gate arrays. IEEE Radiation Effects Data Workshop, pages 115 - 120, July 2006.
- [59] Jameel Hussein, Gary Swift. Mitigating Single-Event Upsets, May 2015.
- [60] Solar flare proton and heavy ion modeling for single event effects. NASA Preferred Reliability Practices, (Practice No. PD-EC-1105):1-3, June 1995.
- [61] J. J. Wang. Radiation effects in FPGAs. The 9th Workshop on Electronics for LHC Experiments, pages 1-10, September-October 2003.
- [62] Toshinori Kuwahara. FPGA-based Reconfigurable On-Board Computing System for Space Application. PhD thesis, the University of Stuttgart, October 2009.
- [63] N. Rezzak and J. J. Wang, "Single Event Latch-Up Hardening Using TCAD Simulations in 130 nm and 65 nm Embedded SRAM in Flash-Based FPGAs," in IEEE Transactions on Nuclear Science, vol. 62, no. 4, pp. 1599-1608, Aug. 2015.
- [64] Latch-up, <https://en.wikipedia.org/wiki/Latch-up>, dostęp 22.06.2016
- [65] F.Sturesson, Single Event Effects (SEE) Mechanism and Effects, June 2009.
- [66] D. R. Czajkowski, M. P. Pagey, P. K. Samudrala, M. Goksel and M. J. Viehman, "Low Power, High-Speed Radiation Hardened Computer amp; Flight Experiment," 2005 IEEE Aerospace Conference, Big Sky, MT, 2005, pp. 1-10.

- [67] Parthivkumar Prajapati, A COMPARISON OF RADIATION TOLERANCE OF DIFFERENT LOGIC STYLES, May 2011.
- [68] T. R. Oldham and F. B. McLean, "Total ionizing dose effects in MOS oxides and devices," in IEEE Transactions on Nuclear Science, vol. 50, no. 3, pp. 483-499, June 2003.
- [69] Marc Poizat, Total Ionizing Dose Mechanisms and Effects, June 2009.
- [70] Kenneth A. LaBel, Challenges for Electronics in the Vision for Space Exploration, NASA, October 2005
- [71] C. A. Hulme, H. H. Loomis, A. A. Ross and Rong Yuan, Configurable fault-tolerant processor (CFTP) for spacecraft onboard processing, Aerospace Conference, 2004. Proceedings. 2004 IEEE, 2004
- [72] J. L. Kaschmitter, D. L. Shaeffer, N. J. Colella, C. L. McKnett and P. G. Coakley, Operation of commercial R3000 processors in the Low Earth Orbit (LEO) space environment," in IEEE Transactions on Nuclear Science, vol. 38, no. 6, pp. 1415-1420, Dec 1991.
- [73] The official web-site of SPENVIS project., <https://www.spenvis.oma.be/>, dostęp 22.06.2016
- [74] Dmitry Burlyaev, System-level Fault-Tolerance Analysis of Small Satellite On-Board Computers, July 2012.
- [75] The official web-site of APEXRAD empirical model., <https://creme.isde.vanderbilt.edu/CREME-MC/help/apexrad>, dostęp 22.06.2016
- [76] Alan Budzynski, PW-Sat 2: PRELIMINARY DESIGN REVIEW - THERMAL CONTROL SYSTEM, August 2015.
- [77] S. L. Clark, K. Avery and R. Parker, "TID and SEE testing results of Altera Cyclone field programmable gate array, Radiation Effects Data Workshop, 2004 IEEE, Atlanta, GA, USA, 2004, pp. 88-90.
- [78] High-Reliability FPGA Designs, http://www.eetimes.com/author.asp?section_id=36&doc_id=1324287, dostęp 22.06.2016
- [79] R. Ichimiya et al., Radiation qualification of electronics components used for the ATLAS level-1 muon endcap trigger system, Nuclear Science Symposium Conference Record, 2004 IEEE, 2004, pp. 779-783 Vol. 2.
- [80] Gennady Ivanovich Zebrev, Igor Olegovich Ishutin, Rustem Galeyevich Useinov, and Vasily Sergeyevich Anashin. Methodology of soft error rate computation in modern microelectronics. IEEE Transactions on Nuclear Science, 57(6):3725-3733, December 2010.
- [81] Kathrin Peter. Data distribution algorithms for reliable parallel storage on Flash memories. Computer science research. Zuse Institute Berlin.
- [82] D.N. Nguyen, S. M. Guertin, and J. D. Patterson. Radiation tests on 2Gb NAND Flash memories. IEEE Radiation Effects Data Workshop, pages 121 - 125, July 2006.

- [83] D.N. Nguyen, S.M. Guertin, G.M. Swift, and A.H. Johnston. Radiation effects on advanced Flash memories. *IEEE Transactions on Nuclear Science*, 46(6):1744- 1750, December 1999.
- [84] G.Cellere et al. A model for TID effects on FG memory cells. *IEEE Transactions on Nuclear Science*, 51(6):3753 - 3758, December 2004.
- [85] Farokh Irom and Duc N. Nguyen. Single event effect characterization of high density commercial nand and nor nonvolatile flash memories. *IEEE TRANSACTIONS ON NUCLEAR SCIENCE*, pages 25472553, December 2007.
- [86] J.W. Howard Jr., M.A. Carts, R. Stattel, C.E. Rogers, T.L. Irwin, C. Dunsmore, J.A. Sciarini, and K.A. LaBel. Total dose and single event effects testing of the Intel Pentium III and AMD K7 microprocessor. *IEEE Radiation Effects Data Workshop*, pages 38 - 47, July 2001.
- [87] T.Takano, T. Yamada, K. Shutoh, and N. Kanekawa. In-orbit experiment on the fault-tolerant space computer aboard the satellite Hiten. *IEEE Transactions on Reliability*, 45:624 - 631, August 2002.
- [88] N. Battezzati, S. Gerardin, A. Manuzzato, A. Paccagnella, S. Rezgui, L. Sterpone, and M. Violante. On the evaluation of radiation-induced transient faults in Flash-based FPGAs. *The 14th IEEE International On-Line Testing Symposium*, pages 135 - 140, July 2008.
- [89] Ramin Roosta. A comparison of radiation-hard and radiation-tolerant FPGAs for space applications. *NASA Electronic Parts and Packaging Program*, (JPL D-31228), December 2004.
- [90] Dae-Soo Oh, Dong-Soo Kang, and Kyoung-Son Jhang. Design and implementation of a radiation tolerant On-Board Computer for science technology Satellite-3. *ASA/glses Conference on Adaptive Hardware and Systems*, pages 17 - 23, June 2010.
- [91] F.Lima, L.Carro, and R. Reis. Designing fault tolerant systems into SRAM-based FPGAs. *Design Automation Conference*, pages 650 - 655, June 2003.
- [92] Gregory R. Allen and Gary M. Swift. Single event effects test results for advanced field programmable gate arrays. *IEEE Radiation Effects Data Workshop*, pages 115 - 120, July 2006.
- [93] N.Battezzati, L.Sterpone, M.Violante, and F.Decuzzi. A new software tool for static analysis of SET sensitiveness in Flash-based FPGAs. *The 18th IEEE/IFIP VLSI System on Chip Conference (VLSI-SoC)*, pages 79 - 84, September 2010.
- [94] F.L. Kastensmidt, E.C.P. Fonseca, R.G. Vaz, O.L. Goncalez, R. Chipana, and G.I. Wirth. TID in Flash-based FPGA: Power supply-current rise and logic function mapping effects in propagation-delay degradation. *IEEE Transactions On Nuclear Science*, 58(4):1927 - 1934, August 2011.
- [95] S. Rezgui et al. New methodologies for SET characterization and mitigation in Flash-based FPGAs. *IEEE Transactions on Nuclear Science*, 54(6):2512-2524, December 2007.

- [96] Sana Rezgui. RT ProASIC3: New RT Flash-Based FPGAs. CMOS Emerging Technologies Workshop, Whistler, Canada, May 2010.
- [97] Andrew S. Keys and Michael D. Watson. Radiation hardened electronics for extreme environments. NASA Marshall Space Flight Center.
- [98] Kurt Anderson. Low-cost, radiation-tolerant, on-board processing solution. IEEE Aerospace Conference, pages 1-8, March 2005.
- [99] Arbi V. Karapetian, Raphael R. Some, and John J. Beahan. Radiation fault modeling and fault estimation for a COTS based space-borne supercomputer. Proceeding IEEE Aerospace Conference, 2002.
- [100] M.S. Reorda, M. Violante, C. Meinhardt, and R. Reis. An On-Board Data-Handling Computer for deep-space exploration built using Commercial-Off-the-Shelf SRAM-based FPGAs. The 24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, pages 254 - 262, October 2009.
- [101] Xinpeng Zhu and Wei Qin. Prototyping a fault-tolerant multiprocessor SoC with run-time fault recovery. The 43rd ACM/IEEE Design Automation Conference, pages 53 - 56, September 2006.
- [102] The official web-site of Green Hills Software company., http://www.ghs.com/index_main.html, dostęp 22.06.2016
- [103] Nobuyasu Kanekawa, Eishi H. Ibe, Ta kashi S uga, and Yu taka Uematsu. Dependability in Electronic Systems. Mitigation of Hardware Failures, Soft Errors, and Electro-Magnetic Disturbances. ISBN 978-1-4419-6714-5. Springer, 2011.
- [104] Hiroyuki Yashiro and Teruo Fujiwara. A high assurance on-line recovery technology for a Space On-board Computer. The 5th International Symposium on Autonomous Decentralized Systems, pages 47 - 56, August 2002.
- [105] L. Chen and A. Avizienis. N-version programming: A fault tolerance approach to reliability of software operation. FTCS-8, page 39, 1978.
- [106] Demid Borodin. Performance-Oriented Fault Tolerance in Computing Systems. PhD thesis, TU Delft, Delft, the Netherlands, 2010.
- [107] A.T. Tai, K.S. Tso, L. Alkalai, S.N. Chau, and W.H. Sanders. Low-cost error containment and recovery for onboard guarded software upgrading and beyond. IEEE Transactions on Computers, 51:121 - 137, February 2002.
- [108] Krzysztof Iniewski. Radiation Effects In Semiconductors. Taylor and Francis Group, 2011.
- [109] C.C. Yui et al. SEU mitigation testing of Xilinx Virtex II FPGAs. IEEE Radiation Effects Data Workshop, pages 92 - 97, July 2003.
- [110] Cel misja satelity PW-Sat 2, <http://pw-sat.pl/misja/>, dostęp 22.06.2016
- [111] Altium Designer, <http://www.altium.com/altium-designer/overview>, dostęp 22.06.2016

- [112] HyperLynx - PCB Analysis and verification tool, <https://www.mentor.com/pcb/hyperlynx/>, dostęp 22.06.2016
- [113] Mentor Graphics, <https://www.mentor.com/>, dostęp 22.06.2016
- [114] Code Composer Studio, <http://www.ti.com/tool/ccstudio>, dostęp 22.06.2016
- [115] Texas Instruments, <http://www.ti.com/>, dostęp 22.06.2016
- [116] Atmel Studio, <http://www.atmel.com/tools/atmelstudio.aspx>, dostęp 22.06.2016
- [117] Atmel Corporation, <http://www.atmel.com/>, dostęp 22.06.2016
- [118] Quartus II Web Edition, <https://www.altera.com/downloads/download-center.html>, dostęp 22.06.2016
- [119] FPGA, SoC and CPLD from Altera , <https://www.altera.com/>, dostęp 22.06.2016
- [120] TI Enhanced Products, <http://www.ti.com/lscds/ti/space-high-reliability/enhanced-products.page>, dostęp 22.06.2016
- [121] CPX: Design of a Standard Cubesat Software Bus, http://www.crn2.inpe.br/conasat1/projetos_cubesat/subsistemas/CDHS/CP1%20-%20CDHS%20-%20Design%20of%20a%20Standard%20Cubesat%20Software%20Bus.pdf, dostęp 22.06.2016
- [122] KUMU A'O CUBESAT: ELECTRICAL POWER SUBSYSTEM, http://www.spacegrant.hawaii.edu/reports/21_SP09/RYamura_SP09.pdf, dostęp 22.06.2016
- [123] Power System of the NTNU Test Satellite, http://nuts.cubesat.no/upload/2012/03/29/eps_project_jacobsen.pdf, dostęp 22.06.2016
- [124] ISTNanosat-1 Heart, https://fenix.tecnico.ulisboa.pt/downloadFile/395144736837/MScThesis-JFerreira_vFinal.pdf, dostęp 22.06.2016
- [125] Tullio's Preferred Part List - CERN, <https://radwg.web.cern.ch/radwg/Pages/Documents/TulliosPartList.pdf>, dostęp 22.06.2016
- [126] Seoul National University SNUSAT-1, <https://radwg.web.cern.ch/radwg/Pages/Documents/TulliosPartList.pdf>, dostęp 22.06.2016
- [127] David A. Kamp, Alan D. DeVilbiss, Gerald R. Haag, Kirk E. Russell and Gary F. Derbenwick, High Density Radiation Hardened FeRAMs on a 130 nm CMOS/FRAM Process, 2005
- [128] R. Ladbury et al., "Radiation Performance of 1 Gbit DDR SDRAMs Fabricated in the 90 nm CMOS Technology Node," 2006 IEEE Radiation Effects Data Workshop, Ponte Vedra, FL, 2006, pp. 126-130.
- [129] NanoAvionics Primary Flight Computer, http://n-avionics.com/wp-content/uploads/2015/04/SatBus1CO_data_sheet.pdf, dostęp 22.06.2016
- [130] D. J. Cochran et al., "Total Ionizing Dose and Displacement Damage Compendium of Candidate Spacecraft Electronics for NASA," 2009 IEEE Radiation Effects Data Workshop, Quebec City, QC, 2009, pp. 25-31.

- [131] Distributed Electrical Power System in Cubesat Applications, <http://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=2070&context=etd>, dostęp 22.06.2016
- [132] T. Rajkowska, R. Graczyka, C. Palaua, P. Orleański, Development of an onboard computer (OBC) for a CubeSat,
- [133] The Internal Data Bus in a Student Satellite, <http://www.diva-portal.org/smash/get/diva2:566379/FULLTEXT01.pdf>, dostęp 22.06.2016
- [134] TMS570LS3137, <http://www.ti.com/lit/ds/symlink/tms570ls3137.pdf>, dostęp 22.06.2016
- [135] MT46V64M8, https://www.google.pl/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0ahUKEwjswP7y38_NAhUEFywKHfypA0cQFggnMAI&url=https%3A%2F%2Fwww.micron.com%2F~%2Fmedia%2Fdocuments%2Fproducts%2Fdata-sheet%2Fdram%2Fddr1%2F512mb_ddr.pdf&usg=AFQjCNGcBJKdC0yzZS-0zu6Hrj_NILtqaw&sig2=FN9BbLVmkyxMMfN-10RsIQ&cad=rja, dostęp 22.06.2016
- [136] EP3C25Q240, https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyc3/cyclone3_handbook.pdf, dostęp 22.06.2016
- [137] FM24CL04B, <http://www.cypress.com/file/136466/download>, dostęp 22.06.2016
- [138] N25Q128A, https://www.google.pl/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwipv5TQ38_NAhWDFSwKHdAeCGkQFggdMAA&url=https%3A%2F%2Fwww.micron.com%2F~%2Fmedia%2Fdocuments%2Fproducts%2Fdata-sheet%2Fnor-flash%2Fserial-nor%2Fn25q%2Fn25q_128mb_1_8v_65nm.pdf&usg=AFQjCNG-wcEk606hvRJYcEU8zwAF5z2omg&sig2=RpE-_TW2bIGlxx_nDZW2Bw, dostęp 22.06.2016
- [139] TPS5430, <http://www.ti.com.cn/lit/ds/symlink/tps5430.pdf>, dostęp 22.06.2016
- [140] MAX1951, <http://pdfserv.maximintegrated.com/en/ds/MAX1951-MAX1952.pdf>, dostęp 22.06.2016
- [141] TPS62160, <http://www.ti.com/lit/ds/slvsam2d/slvsam2d.pdf>, dostęp 22.06.2016
- [142] Atmega128A, <http://www.atmel.com/images/doc2467.pdf>, dostęp 22.06.2016
- [143] SN74AVC8T245, <http://www.ti.com.cn/general/cn/docs/lit/getliterature.tsp?genericPartNumber=sn74avc8t245&fileType=pdf>, dostęp 22.06.2016
- [144] SN74CB3T3257, <http://www.ti.com.cn/general/cn/docs/lit/getliterature.tsp?genericPartNumber=sn74cb3t3257&fileType=pdf>, dostęp 22.06.2016
- [145] PCA9517, http://www.nxp.com/documents/data_sheet/PCA9517.pdf, dostęp 22.06.2016
- [146] DS3231, <https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>, dostęp 22.06.2016

- [147] INA209, <http://www.ti.com/lit/ds/symlink/ina209.pdf>, dostęp 22.06.2016
- [148] INA219, <http://www.ti.com/lit/ds/symlink/ina219.pdf>, dostęp 22.06.2016
- [149] AT45DB161B, <http://www.atmel.com/Images/doc2224.pdf>, dostęp 22.06.2016
- [150] MT9D111, http://www.dragonwake.com/download/camera/MT9D111/mt9d111_rev5.pdf, dostęp 22.06.2016
- [151] ISO 26262, <http://quality-one.com/iso-26262/>, dostęp 22.06.2016
- [152] A summary of the IEC 61508 Standard, http://www.win.tue.nl/~mvdbbrand/courses/sse/1213/iec61508_overview.pdf, dostęp 22.06.2016
- [153] Super Capacitor Calculator, <https://www.maximintegrated.com/en/design/tools/calculators/product-design/supercap.cfm>, dostęp 22.06.2016
- [154] SN65HVD234, <http://www.ti.com/lit/ds/symlink/sn65hvd234.pdf>, dostęp 22.06.2016
- [155] LM60, <http://www.ti.com/lit/ds/symlink/lm60.pdf>, dostęp 22.06.2016
- [156] TPS2553, <http://www.ti.com/lit/ds/symlink/tps2553-q1.pdf>, dostęp 22.06.2016
- [157] Wayne Stanton, Patrick Schulz, John Wise, Brett King, Intelligent Power Management System for a Nanosatellite
- [158] Micron Technology, <https://www.micron.com/>, dostęp 22.06.2016
- [159] Terminations for Advanced CMOS Logic, <https://www.fairchildsemi.com/application-notes/AN/AN-610.pdf>, dostęp 22.06.2016
- [160] Proper Termination of High-Speed Digital Signals , <http://www.marvintest.com/KnowledgeBase/KBArticle.aspx?ID=196>, dostęp 22.06.2016
- [161] DDR2 (Point-to-Point) Package Sizes and Layout Basics, <https://www.micron.com/~media/Documents/Products/Technical%20Note/DRAM/TN4720.pdf>, dostęp 22.06.2016
- [162] LI-OV9712-FF-65, <https://www.leopardimaging.com/LI-OV9712-FF.html>, dostęp 22.06.2016
- [163] Leopard Imaging, <https://www.leopardimaging.com/>, dostęp 22.06.2016
- [164] OmniVision OV9712, <http://www.ovt.com/products/sensor.php?id=29>, dostęp 22.06.2016
- [165] FH12-24S-0.5SH, <http://pl.mouser.com/ProductDetail/Hirose-Connector/FH12-24S-05SH55/?qs=Ux3WWAnHpjDGmuhYLgxkoQ%3D%3D>, dostęp 22.06.2016
- [166] ESQ-126-39-G-D, http://suddendocs.samtec.com/catalog_english/esq_th.pdf, dostęp 22.06.2016
- [167] Samtec, <https://www.samtec.com/>, dostęp 22.06.2016

- [168] INA209 EVM, <http://www.ti.com/product/INA209/toolssoftware>, dostęp 22.06.2016
- [169] INA219 EVM, <http://www.ti.com/product/INA219/toolssoftware>, dostęp 22.06.2016
- [170] Nios II Processor, <https://www.altera.com/products/processors/overview.html>, dostęp 22.06.2016
- [171] CFPS-69IB, <http://www.farnell.com/datasheets/2048860.pdf>, dostęp 22.06.2016
- [172] Avalon Interface Specifications - Altera, https://www.google.pl/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&cad=rja&uact=8&ved=0ahUKEwjoovoamh9DNAhUJiywKHeceAPsQFggdMAA&url=https%3A%2F%2Fwww.altera.com%2Fliterature%2Fmanual%2Fmn1_avalon_spec.pdf&usg=AFQjCNFzM91w945b5Pzdj9pNS23KomP13g&sig2=Tqnc9vByaD7cGRUGpC72eA, dostęp 22.06.2016
- [173] I2C controller core, <http://opencores.org/project,i2c,overview>, dostęp 22.06.2016
- [174] I2C - Master Core Specification, http://opencores.org/websvn,filedetails?repname=i2c&path=%2Fi2c%2Ftrunk%2Fdoc%2Fi2c_specs.pdf, dostęp 22.06.2016
- [175] Altera - JTAG UART Core, http://www.ece.mtu.edu/~saeid/multimedia/labs/Documentation/n2cpu_nii51009_JTAG_UART.pdf, dostęp 22.06.2016
- [176] FreeRTOS, <http://www.freertos.org/>, dostęp 22.06.2016
- [177] Real Time Engineers Ltd., <http://www.freertos.org/RTOS-contact-and-support.html>, dostęp 22.06.2016
- [178] SafeRTOS, <http://www.highintegritysystems.com/safertos/>, dostęp 22.06.2016
- [179] Wittenstein AG, <http://www.wittenstein.de/de-de/>, dostęp 22.06.2016
- [180] DO178C, <http://www.adacore.com/gnatpro-safety-critical/avionics/do178c/>, dostęp 22.06.2016
- [181] HALCoGen - Hardware Abstraction Layer Code Generator for Hercules MCUs , <http://www.ti.com/tool/halcogen>, dostęp 22.06.2016
- [182] Atmel-ICE, <http://www.atmel.com/tools/atmel-ice.aspx>, dostęp 22.06.2016
- [183] USB-Blaster II Download Cable, https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ug/ug_usb_blstr_ii_cable.pdf, dostęp 22.06.2016
- [184] Debug Probes - J-Link, <https://www.segger.com/jlink-debug-probes.html>, dostęp 22.06.2016
- [185] SEGGER - The Embedded Experts, <https://www.segger.com/>, dostęp 22.06.2016
- [186] PCA9555, <http://www.ti.com/product/PCA9555>, dostęp 22.06.2016

- [187] FT230X, http://www.ftdichip.com/Support/Documents/DataSheets/ICs/DS_FT230X.pdf, dostęp 22.06.2016
- [188] FTDI Chip, <http://www.ftdichip.com/>, dostęp 22.06.2016
- [189] 74HC4051, http://www.nxp.com/documents/data_sheet/74HC_HCT4051.pdf, dostęp 22.06.2016
- [190] SpaceWire Avionics Data Bus, http://www.interfacebus.com/SpaceWire_Avionics_Bus.html, dostęp 22.06.2016
- [191] SpaceWire Verification IP, <http://www.smart-dv.com/vip/spacewire.html>, dostęp 22.06.2016
- [192] Cubesat Space Protocol (CSP), <http://gomspace.com/index.php?p=products-software>, dostęp 22.06.2016
- [193] What is RISC? - Stanford Computer Science, <https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/whatis/index.html>, dostęp 22.06.2016
- [194] Wikipedia - Cubesat Space Protocol, https://en.wikipedia.org/wiki/Cubesat_Space_Protocol, dostęp 22.06.2016
- [195] EP3C25Q240 IBIS file, https://www.altera.com/support/support-resources/download/board-layout-test/ibis/ibs-ibis_index.html, dostęp 22.06.2016
- [196] MT46V64M8 IBIS file, <https://www.micron.com/partsdram/ddr-sdram/mt46v128m8p-75>, dostęp 22.06.2016
- [197] Real-time operating system, <http://www.freertos.org/>, dostęp 22.06.2016
- [198] Configuration, Design Security, and Remote System Upgrades in the Cyclone III Device Family, https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/hb/cyc3/cyc3_ciii51016.pdf, dostęp 22.06.2016