

Politechnika Warszawska

WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH



Instytut Systemów Elektronicznych

Praca dyplomowa magisterska

na kierunku Elektronika
w specjalności Mikrosystemy i Systemy Elektroniczne

Wielokanałowy moduł sterujący pułapką jonową w standardzie
MicroTCA.4

Tomasz Przywózki

Numer albumu 252871

promotor
dr inż. Grzegorz Henryk Kasproicz

WARSZAWA 2020

Streszczenie

Wielokanałowy moduł sterujący pułapką jonową w standardzie MicroTCA.4

W ramach niniejszej pracy zaprojektowany został układ „Shuttler”, sterujący w czasie rzeczywistym 18 kanałami sygnałów analogowych, przeznaczonych do kontroli elektrod pułapki jonowej wykorzystywanej w eksperymentach fizyki kwantowej. Stosowane przetworniki zapewniają wysoką precyzję działania w tej dziedzinie, dzięki typowym wartościom nieliniowości DNL oraz INL na poziomie równym bądź mniejszym niż ± 1 LSB. Shuttler dedykowany jest do działania w kracie MicroTCA.4, jako jeden z elementów systemu sprzętowego Sinara, wspierany przez system kontrolny ARTIQ (Advanced Real-Time Infrastructure for Quantum physics, pol. Zaawansowana Infrastruktura Czasu Rzeczywistego dla Fizyki Kwantowej). Moduł w formacie AMC zaprojektowany został tak, aby wraz z pozostałymi modułami projektu Sinara stanowił system sterowania pułapkami jonowymi (zawierającymi kilkaset elektrod) zoptymalizowany pod kątem kosztu jego zestawienia. Opracowano trzy koncepcje niestandardowego wykorzystania połączeń pomiędzy układem FPGA a przetwornikiem o magistrali równoległej w standardzie LVDS, pozwalające na znaczne zwiększenie ilości kanałów przypadających na pojedynczy moduł. W ramach pracy dokonano symulacji zaproponowanych metod, a następnie ich oceny. Najbardziej obiecujące rozwiązanie – sterowanie odbiorników LVDS linią asymetryczną – zostało następnie przetestowane z wykorzystaniem płyty demonstracyjnej przetwornika. Dokonane pomiary wykazały poprawność działania interfejsu, który ostatecznie został zaimplementowany w projekcie modułu Shuttler. Badania te pozwoliły na trzykrotne zwiększenie ilości przetworników stosowanych w pojedynczym module oraz na ponad dwukrotne zmniejszenie kosztów zestawienia przykładowego 1000-kanałowego systemu sterowania.

Słowa kluczowe: pułapka jonowa, FPGA, MicroTCA, AMC, przetworniki cyfrowo-analogowe, ARTIQ, Sinara, system sterujący czasu rzeczywistego

Abstract

Multi-channel ion trap control module in MicroTCA.4 standard

In this master thesis project, the „Shuttler” module was designed. It is a driver module of 18 channels of analog signals for real-time control of electrodes in ion traps used in quantum physics experiments. The converters which are used in project provide unusual precision in this quantum field, thanks to typical DNL and INL nonlinearity values equal to or less than ± 1 LSB. Shuttler was designed to work in MicroTCA crate as a one of subsystem of hardware system Sinara, that is supported by control system ARTIQ (Advanced Real-Time Infrastructure for Quantum physics). The main criteria during designing process was to build ion traps control system (with several hundred electrodes) optimized for the cost of set-up. Three methods of non-standard links between FPGA and converters with LVDS parallel receivers were presented, allowing significant increase the number of analog channels per module. These methods have been simulated and the most optimal one has been chosen. The most promising method of driving LVDS receiver with asymmetric signal was tested with the usage of converter demo board. Thanks to the research it was possible to increase threefold the number of digital-to-analog converters used on a single Shuttler module and to reduce the cost of setting up an example 1000 channels control system more than 50%.

Keywords: ion trap, FPGA, MicroTCA, AMC, digital-to-analog converters, ARTIQ, Sinara, real-time control system



Warszawa, 28.10.2019
miejsowość i data
place and date

Tomasz Mirosław Przywózki
imię i nazwisko studenta
name and surname of the student
252871
numer albumu
student record book number
Elektronika
kierunek studiów
field of study

OŚWIADCZENIE *DECLARATION*

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.

Under the penalty of perjury, I hereby certify that I wrote my diploma thesis on my own, under the guidance of the thesis supervisor.

Jednocześnie oświadczam, że:
I also declare that:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
- *this diploma thesis does not constitute infringement of copyright following the act of 4 February 1994 on copyright and related rights (Journal of Acts of 2006 no. 90, item 631 with further amendments) or personal rights protected under the civil law,*
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- *the diploma thesis does not contain data or information acquired in an illegal way,*
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów lub tytułów zawodowych,
- *the diploma thesis has never been the basis of any other official proceedings leading to the award of diplomas or professional degrees,*
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- *all information included in the diploma thesis, derived from printed and electronic sources, has been documented with relevant references in the literature section,*
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.
- *I am aware of the regulations at Warsaw University of Technology on management of copyright and related rights, industrial property rights and commercialisation.*



Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płyce kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

I certify that the content of the printed version of the diploma thesis, the content of the electronic version of the diploma thesis (on a CD) and the content of the diploma thesis in the Archive of Diploma Theses (APD module) of the USOS system are identical.

.....
czytelny podpis studenta
legible signature of the student

Spis treści

1. Wstęp	9
1.1. Technologie fizycznych implementacji informacji kwantowej	9
1.2. Pułapka jonowa	10
1.3. Przegląd istniejących systemów	13
2. Geneza, cel oraz założenia pracy	19
2.1. Geneza	19
2.2. Cele pracy	19
2.3. Założenia techniczne	19
3. Koncepcja architektury i dobór komponentów	21
3.1. Dobór środowiska sprzętowego	22
3.2. Koncepcja systemu	23
3.3. Przetwornik cyfrowo-analogowy dla modułu Shuttler	24
3.4. Pierwsze koncepcje modułu Shuttler	25
3.5. Analiza kosztów dla 1000 kanałowego systemu	27
4. Koncepcje metod optymalizacji	31
4.1. Analiza układu odbiornika	31
4.2. Analiza układu nadajnika	32
4.3. Koncepcja połączenia równoległego	33
4.4. Koncepcja sterowania odbiornika LVDS linią asymetryczną	35
5. Symulacje badanych metod	37
5.1. Połączenie równoległe	37
5.2. Sterowanie LVDS linią asymetryczną	40
5.3. Wnioski	42
6. Testy i pomiary koncepcji asymetrycznej transmisji danych	45
6.1. Wykorzystywany sprzęt	45
6.2. Układ pomiarowy	49
6.3. Pomiary i wnioski	51
6.3.1. Pomiar 1. – badanie wpływu przetwornika LTC2000 na składową stałą napięcia linii sygnałowej	52
6.3.2. Pomiar 2. – badanie zaproponowanego interfejsu dla częstotliwości 100 MHz	53
6.3.3. Oszacowanie maksymalnej częstotliwości pracy badanego interfejsu	58
6.4. Podsumowanie badań	59
7. Finalny moduł Shuttler	60
7.1. Dobór układu FPGA	60
7.2. Realizacja poszczególnych bloków funkcjonalnych	61
7.2.1. Połączenia FPGA	62
7.2.2. Blok MMC	65
7.2.3. Przetworniki C/A oraz tor analogowy	66
7.2.4. Dystrybucja sygnału zegarowego	67

7.2.5. USB	69
7.2.6. Magistrala I ² C	70
7.2.7. Układy zasilania	71
7.3. Realizacja i symulacje projektu obwodu drukowanego	73
7.3.1. Połączenia linii danych przetworników	75
7.3.2. Linie gigabitowe	75
7.3.3. Sygnały zegarowe	76
7.3.4. Wyrównanie balansu miedzi	78
7.4. Analiza i porównanie kosztów całego systemu	78
8. Wnioski	80
9. Lista skrótów	83
A. Schematy elektryczne płyty Shuttler	89
B. Kody użyte w projekcie	126
B.1. Projekt FPGA	126
B.2. Kod mikrokontrolera do translacji UART - SPI	128

1. Wstęp

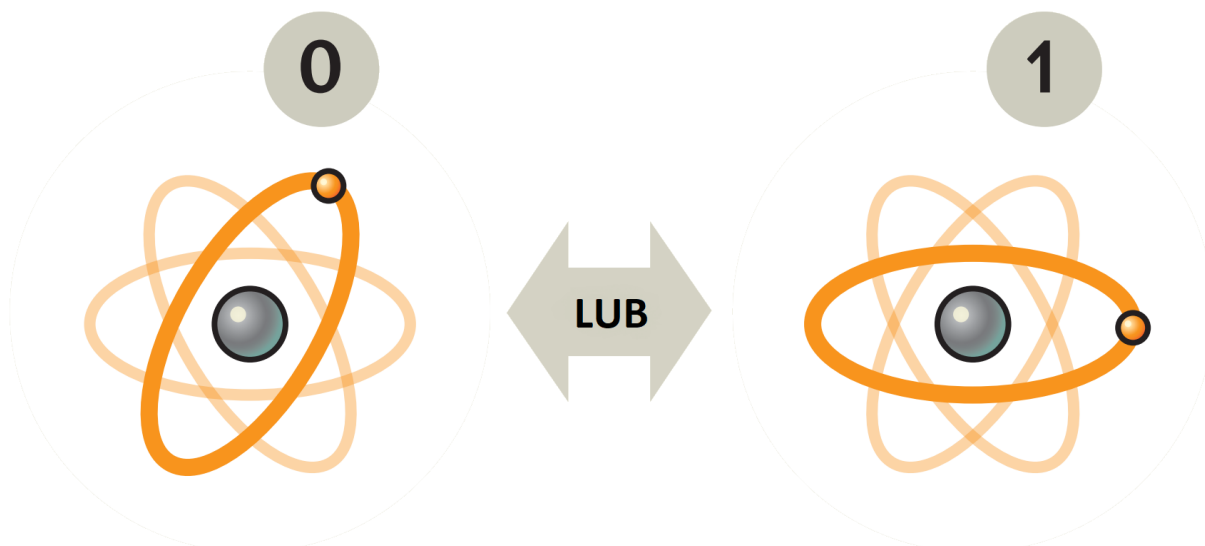
Obserwując obecny postęp technologiczny w szeroko rozumianej dziedzinie informatyki, można zauważyć tendencję do rozwoju technologii kwantowej. Wiele spośród znanych firm, dysponujących wysokimi budżetami, poświęca coraz więcej czasu i zasobów na badanie tej technologii, aby doprowadzić do skonstruowania w pełni funkcjonalnego komputera kwantowego. Przykładem takich firm są: Microsoft [35], Google [17], Intel [20], IBM czy CERN [7]. Trudno się temu dziwić – postęp w zwiększaniu mocy obliczeniowej sprzętu opartego na półprzewodnikach znacząco zwolnił, ponieważ proces miniaturyzacji powoli osiąga swoje granice. Już dziś produkowane są układy scalone wykonane w technologii 5 nm [38], gdzie dla porównania jeden z większych atomów – cez – posiada średnicę około 500 pm.

Technologia kwantowa nie jest jednak tylko kolejnym etapem w rozwoju możliwości obliczeniowych ludzkości. Potencjalne zastosowania sugerują raczej, że w przypadku udanej implementacji komputerów kwantowych dojdzie do prawdziwego skoku technologicznego. Już teraz wymienia się wiele zastosowań, a nawet opracowuje kwantowe algorytmy, które pozwolą na rozwiązanie problemów nieosiągalnych dla obecnych komputerów. Przykładem jest algorytm Shora [34], który umożliwi szybką faktoryzację, a co za tym idzie – złamanie części z obecnie stosowanych algorytmów kryptograficznych. Z tego powodu już teraz powstają prace opisujące rozwiązania tego potencjalnego problemu [5]. Uznaje się także, że komputery kwantowe pozwolą na symulacje skomplikowanych układów chemicznych, osiągnięcie jeszcze większej precyzji wzorców zegarowych czy też utworzenie różnego rodzaju czujników, dokonujących pomiarów na poziomie pojedynczych atomów lub cząsteczek [27]. Jest to zatem technologia, która ma szansę w przyszłości całkowicie zrewolucjonizować obraz świata, podobnie jak już wcześniej do takiej rewolucji doprowadziło pojawienie się klasycznych komputerów.

1.1. Technologie fizycznych implementacji informacji kwantowej

Podstawową różnicą pomiędzy komputerem klasycznym a kwantowym jest sposób definiowania najmniejszej jednostki informacji. Klasyczny bit przyjmuje wartość logiczną 0 bądź 1, która zazwyczaj reprezentowana jest przez odpowiednią wartość napięcia. Kubit – będący najmniejszą i niepodzielną jednostką informacji kwantowej – może przyjmować wiele różnych modeli fizycznych. Przykładem takiego modelu jest pozycja elektronu wokół jądra atomowego (przedstawiona na rys. 1.1). W tym przypadku informacja zostaje zakodowana przez umieszczenie elektronu na konkretnej powłoce. Szczególną właściwością kubitów jest fakt, że dzięki kwantowej superpozycji elektron może znajdować się w dwóch stanach jednocześnie. W konsekwencji nic nie stoi na przeszkodzie, aby informacja przybierała wartość logiczną 0 i 1 w tym samym czasie, pozwalając na testowanie wielu rozwiązań danego problemu symultanicznie. Drugą właściwością jest możliwość osiągnięcia stanu splątanego. Splątanie polega na utworzeniu połączenia pomiędzy dwoma lub większą ilością kubitów, które powoduje, że działanie podjęte na jednym z nich wpływa także na pozostałe, niezależnie od dzielących ich odległości. Gdyby wprowadzić wiele kubitów w stan splątania, możliwe byłoby zrównoleglenie obliczeń na niespotykaną dotąd skalę.

Istnieje kilka sposobów fizycznej implementacji kubitów. Jedną z najbardziej obiecujących – pod względem osiągniętych dokładności bramek – jest technika wykorzystująca obwody nadprzewodzące. Wartości 0 oraz 1 mogą być kodowane jako obecność lub brak fotonu z zakresu mikro-



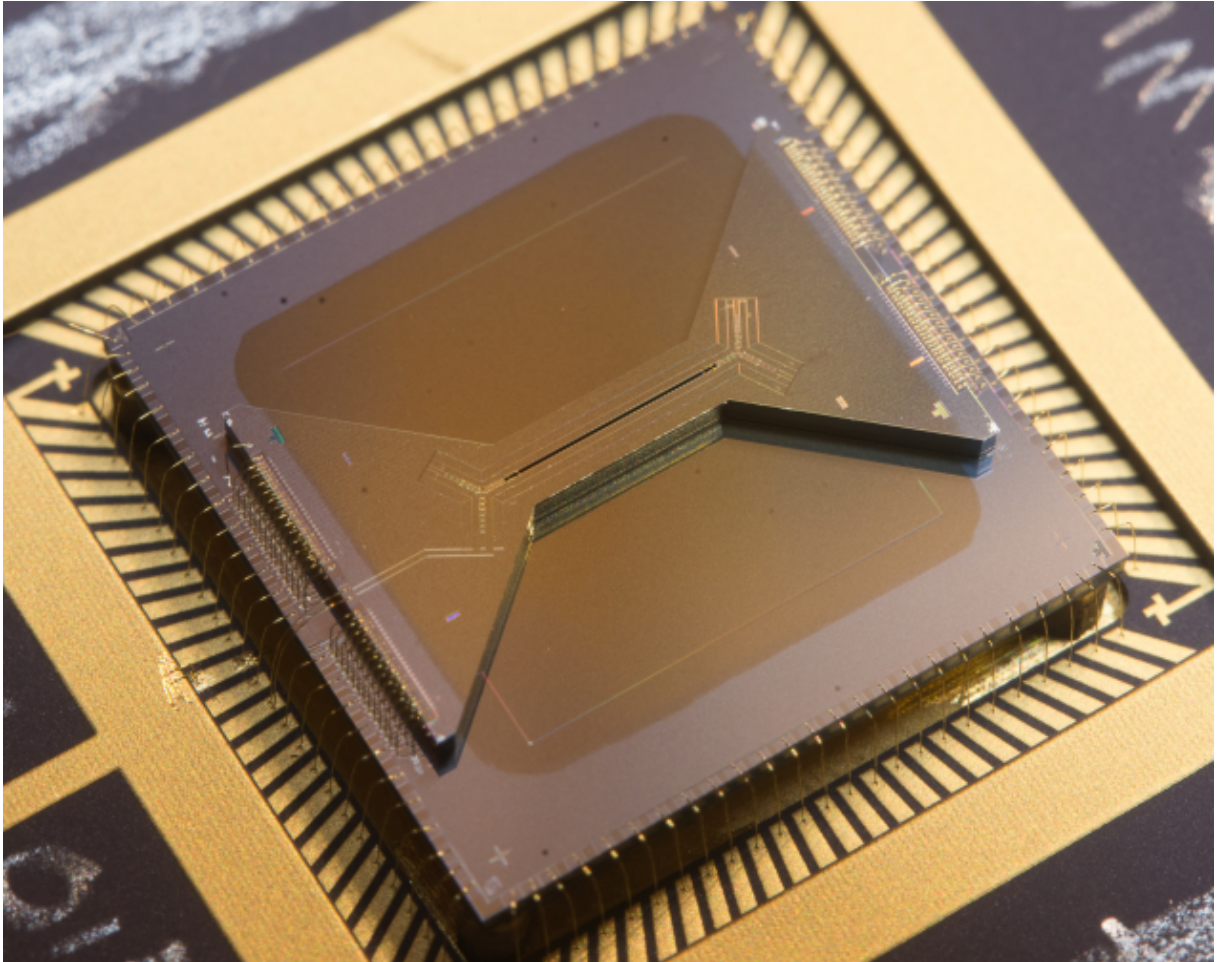
Rysunek 1.1: Przykład fizycznego modelu kubitów na podstawie orbity elektronu [27].

falowego albo jako oscylacje prądu elektrycznego w obwodzie. Drugą metodą jest wykorzystanie zjonizowanego atomu uwięzionego w pułapce i wspomnianego stanu jego elektronu. Przetwarzanie informacji w tej metodzie jest około 1000 razy wolniejsze, niż w przypadku doświadczeń przeprowadzanych na bramkach bazujących na obwodach nadprzewodzących [32]. Jest ona jednak równie przyszłościowa ze względu na dużo większe możliwości miniaturyzacji docelowego urządzenia.

Przed konstrukcją w pełni działającego komputera kwantowego stoją trzy główne problemy. Pierwszym jest kwestia skalowania systemu. Zwykle powiększanie badanych struktur prowadzi do tego, że układ traci swoje kwantowe właściwości. Przyjęto zatem kierunek tworzenia niezależnych modułów, które wciąż wykorzystywałyby superpozycję oraz splątanie kwantowe, jednocześnie przysyłając informacje do innych modułów klasycznymi metodami np. przez emisję fotonu. Drugim problemem jest samo utrzymanie superpozycji i stanu splątania. Prowadzone są liczne badania, mające na celu uzyskanie jak najdłuższego czasu koherencji – czyli czasu, w którym możliwe jest ustalenie stanu kubitów, a następnie przeprowadzanie na nich operacji, zanim na skutek oddziaływania z otoczeniem utracą swoje właściwości. Ostatnim problemem jest zapewnienie dokładności działania bramek kwantowych. Ze względu na samą konstrukcję bramki oraz stosowane narzędzia kontrolno-pomiarowe idealny stan kwantowy nie pokrywa się w pełni z faktycznym stanem wyjściowym wytwarzanym przez bramkę. W praktyce, osiągając odpowiednio dużą dokładność (poprzez optymalizację konstrukcji bramki oraz zwiększanie precyzji narzędzi kontrolnych), możliwe będzie uzyskanie w pełni sprawnego komputera przy zastosowaniu kwantowej korekcji błędów [32].

1.2. Pułapka jonowa

Pierwotnym zadaniem pułapki jonowej w badaniach kwantowych było uwięzienie jonu w miejscu, umożliwiając jego schładzanie za pomocą wiązki laserowej oraz przeprowadzanie dalszych manipulacji. Obecnie stosowane są dużo bardziej skomplikowane struktury, mające na celu utrzymanie większej ilości jonów w rzędzie, a także umożliwienie zamiany ich pozycji. Przykładem takiej pułapki (wykorzystywanej przez naukowców, we współpracy z którymi powstał niniejszy projekt) jest pułapka HOA 2.0, która zaprezentowana została na rys. 1.2. Widoczna po środku szczelina to przestrzeń,



Rysunek 1.2: Pułapka HOA 2.0 wraz z wyprowadzeniami dla sterowania elektrodami [23].

gdzie jony więzione są w rzędzie mikropułapek i poddawane dalszym badaniom. Co istotne, jony te mogą poruszać się tylko w jednym kierunku – wzdłuż wspomnianej szczeliny, aż do rozgałęzień widocznych na jej końcu. Tam z kolei dochodzi do zamiany pozycji atomów poprzez ich transport do odpowiednich linii rozgałęzienia, a następnie przywrócenie do centralnej strefy w odwrotnej kolejności. Wzdłuż całej długości linii (oraz linii pomocniczych rozgałęzienia), po których poruszają się atomy, rozmieszczone są elektrody kontrolne oraz wspólna elektroda sygnału Radio Frequency (RF). Schemat ich rozmieszczenia dla środkowej części pułapki ukazany został na rys. 1.3.

Cały układ posiada elektrody dla 94 napięć kontrolnych (które w przypadku braku ruchu jonów są sterowane napięciami stałymi – dlatego nazywane są elektrodami Direct Current (DC)) oraz jedną wspólną dla całego obszaru pułapki na sygnał RF. Wymagana częstotliwość (o wartości rzędu dziesiątek MHz) zależy od doboru atomu oraz oczekiwanej częstotliwości oscylacji jonu w pułapce.

Po spuławkowaniu i schłodzeniu atomów możliwe jest wykonywanie na nich różnych operacji logicznych oraz ich splątanie [39]. Wykorzystując światło lasera o odpowiedniej częstotliwości i świecąc odpowiedni okres czasu, osiąga się wzbudzenie jonu, a więc zmianę poziomu energetycznego elektronu na wyższy. Stan ten ma jednak skończony czas życia, dlatego wszelkie operacje kwantowe muszą zostać wykonane w ściśle określonym czasie. Powtarzając ten sam proces na jonie już wzbudzonym, doprowadza się do emisji wymuszonej i tym samym realizuje kwantową bramkę



Rysunek 1.3: Schemat rozmieszczenia elektrod w centralnej części pułapki HOA. Elektrody rozmieszczone są co 70 μm , zaś szerokość szczeliny pomiędzy elektrodami, gdzie łąpane są jony, wynosi 60 μm [23].

NOT. Zmniejszając czas naświetlania, można osiągnąć superpozycję stanu podstawowego i wzbu-
dzonego.

W celu stworzenia w pełni funkcjonalnego komputera kwantowego konieczne jest wykonywanie
operacji na większej ilości kubitów jednocześnie. Dzięki temu, że jony w pułapce umieszczone są w
jednej linii w bliskich odległościach, wzajemnie oddziałują na siebie elektrostatycznie i dlatego moż-
liwe jest pobudzenie do drgań pojedynczego jonu wraz ze wszystkimi sąsiednimi. Co istotne, ope-
racje zmiany poziomu energetycznego oraz obecności drgań mogą być są od siebie zależne dzięki
odpowiedniej manipulacji częstotliwości wiązki laserowej. Przykładowo można wykonać operację
wzbudzenia drgań, która zadziała wyłącznie w przypadku, gdy jon będzie w stanie podstawowym
lub operację wzbudzenia jonu wyłącznie w przypadku obecności drgań.

Wykorzystywanie określenia „napięcia stałego” na elektrodach jest konsekwencją stosowania
tego pojęcia w kontekście pierwotnych pułapek liniowych, gdzie elektrody miały za zadanie wyłącz-
nie uniemożliwić jonom ucieczkę wzdłuż osi. Opisana wyżej zasada dokonywania wielokubitowych
operacji kwantowych dotyczy jonów znajdujących się w sąsiednich mikropułapkach, dzięki czemu
mogą one na siebie oddziaływać. Jeśli jednak dany jest rząd kubitów w pułapce 1. 2. oraz 3.
i potrzeba wykonać operację na kubitach 1. i 2., a następnie 1. i 3. konieczne jest zastosowanie
następującej procedury uwzględniającej transport tych jonów, a więc zmianę napięć na elektrodach:

- oddalenie jonu 3. na odległość, przy której nie będą na niego oddziaływały drgania jonu 1.
oraz 2.,
- dokonanie operacji na jonach 1. i 2.,
- transport jonu 1. do jednej z linii rozgałęzień oraz 2. do drugiej linii,
- transport powrotny jonu 1. i umieszczenie go w mikropułapce sąsiadującej z jonem 3,
- wykonanie operacji na jonach 1. i 3..

Z kolei wszystkie te operacje muszą zostać wykonane z uwzględnieniem warunków czasowych –
skończonego czasu trwania superpozycji (m. in. ze względu na naturalny proces powrotu jonu do

stanu podstawowego) oraz ewentualnych splątań między kubitami. W przypadku dalszego skalowania układów kwantowych, zestawianych ze sobą będzie coraz więcej tego typu pułapek. Zaistnieje zatem konieczność dodatkowego przesyłania informacji pomiędzy nimi, co także zwiększy rygory czasowe i wymusi zastosowanie szybkich przetworników cyfrowo-analogowych.

Docelowym rozwiązaniem problemu sterowania pułapkami jonowymi będzie produkcja układu Application-Specific Integrated Circuit (ASIC), łączącego w sobie pułapkę oraz przetworniki. Powstały już pierwsze projekty pośrednie, gdzie pułapka jonowa umieszczona została bezpośrednio na małym obwodzie drukowanym, zawierającym dwa przetworniki cyfrowo-analogowe, obsługujące wspólnie 80 kanałów [18]. Celem tych prac było jednak badanie zysków, związanych ze skróceniem długości linii doprowadzeń sygnału do elektrod, a nie stanowienie gotowego narzędzia dla eksperymentów nad kubitami w postaci uwięzionych jonów. Z kolei zaprojektowanie układu typu ASIC jest bardzo czasochłonne i niezwykle kosztowne, zwłaszcza że mowa jest tu o produkcie, który nie będzie produkowany na masową skalę. Konieczne jest zatem wcześniejsze wykonanie wielu badań, aby określić wszystkie potrzebne funkcjonalności i opracować odpowiednie algorytmy, obsługujące wielomodułową i wielokubitową pułapkę.

1.3. Przegląd istniejących systemów

Sterowanie wieloma niezależnymi urządzeniami stało się podejściem niepraktycznym, utrudniającym kwestię skalowalności całego systemu, a także synchronizacji poszczególnych funkcjonalności, co z kolei miało swoje konsekwencje w pogorszeniu parametrów badanych bramek kwantowych. Naukowcy dokładają coraz większych starań, aby stosowany sprzęt elektroniczny stanowił zwarty system sterowania wszystkich podsystemów. Poza generacją sygnału o częstotliwościach RF oraz DC dla pułapek jonowych konieczne jest także zapewnienie:

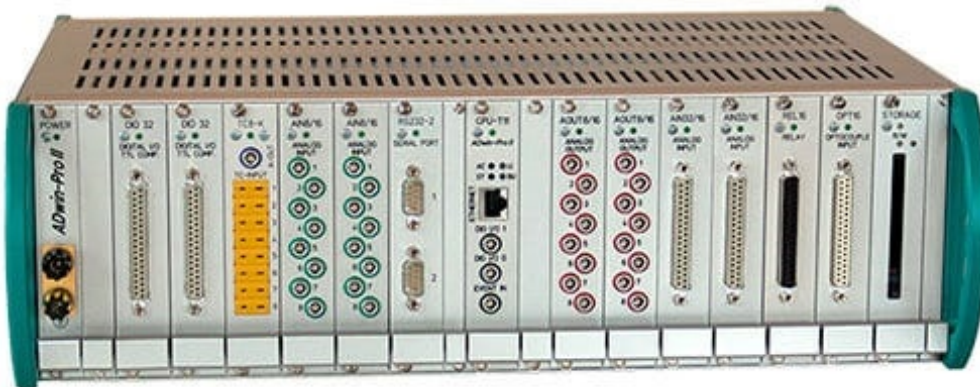
- sterowania laserów, odpowiedzialnych za chłodzenie jonów oraz wprowadzanie w stany splątania;
- sterowania modulatorami akustooptycznymi, zmieniającymi częstotliwości światła laserowego;
- kontroli zasilania;
- możliwości odczytu stanów jonów przy wykorzystaniu zjawiska fotoemisji.

W praktyce jednak oczekiwania pod względem poszczególnych parametrów układów są bardzo różne w zależności od przeprowadzanych doświadczeń. Tworzonych jest zatem wiele oddzielnych systemów, zazwyczaj niekompatybilnych ze sobą, co utrudnia wymianę osiągniętej wiedzy – brak odpowiedniego sprzętu powoduje, że często niemożliwe jest odtworzenie danego doświadczenia przez inny podmiot. Poniżej przedstawiono kilka przykładów systemów, stanowiących potencjalne kompleksowe rozwiązanie sprzętowo-programowe dla doświadczeń kwantowych.

ADwin-PRO-II

Propozycja firmy CAS DataLoggers [8] to uniwersalny system przetwarzania danych i kontroli w czasie rzeczywistym. Stosowany jest w doświadczeniach kwantowych z wykorzystaniem zimnych atomów w pułapce magnetycznooptycznej. Prezentuje w pełni modułowe podejście, gdzie za poszczególne funkcjonalności odpowiadają dedykowane karty, przeznaczone do umieszczenia w autorskiej

kracie. Wyposażono je także we własne procesory, umożliwiające samodzielną pracę. Przykład zestawionego sprzętu przedstawiony został na rys. 1.4.



Rysunek 1.4: Przykładowa kasetka wyposażona w moduły z rodziny ADwin-PRO-II [8].

System posiada jednak dwie zasadnicze wady. Po pierwsze jest to projekt w całości komercyjny, co znacząco utrudnia dostosowywanie systemu do rosnących potrzeb naukowców, a także wymaga wykorzystania autorskiego oprogramowania, nie będącego rozwiązaniem na otwartej licencji. Drugim problemem jest kwestia głównej jednostki obliczeniowej, którą jest układ Zynq firmy Xilinx bądź procesor ADSP TS101 (w zależności od wybranego modułu zarządzającego krata). Wszystkie pozostałe podsystemy posiadają wyłącznie proste procesory. Zdecydowanie ogranicza to możliwości obliczeniowe całego systemu. Problemem jest także mała przepustowość magistrali pomiędzy jednostką zarządzającą a dostępnymi modułami przetworników, wynosząca 200 MB/s.

Yggdrasil

Yggdrasil to nazwa oprogramowania utworzonego we współpracy kilku włoskich uniwersytetów i instytutów [31]. Autorzy opracowali równolegle dedykowany system sprzętowy. Jego cechą charakterystyczną jest dostosowanie do sterowania hybrydowymi doświadczeniami – łączącymi przykładowo metodę uwięzionych jonów i kwantowego gazu neutralnych atomów. Na system składają się dwa typy modułów. Pierwszy – nadrzędny – odpowiada za komunikację z użytkownikiem, obsługę do 6 modułów typu podrzędnego oraz połączenia z innymi modułami nadrzędnymi (w celu utworzenia struktury drzewa w przypadku systemu, wymagającego większej ilości zasobów).

Z punktu widzenia niniejszej pracy największą wadą tej koncepcji jest architektura modułu podrzędnego. Jest to bowiem jeden rodzaj płyty, zapewniający wszystkie potrzebne funkcjonalności, co znacząco ogranicza elastyczność całego systemu. Moduł ten zawiera:

- 2 układy Direct Digital Synthesis (DDS) AD9910 z wyjściem o częstotliwości w zakresie 0,1-400 MHz,
- 2 przetworniki analogowo-cyfrowe o częstotliwości próbkowania do 100 MHz i rozdzielczości 16 bitów,
- 2 przetworniki cyfrowo-analogowe o częstotliwości próbkowania do 800 MHz i rozdzielczości 16 bitów,

- 2 przetworniki analogowo-cyfrowe o częstotliwości próbkowania około 100 kHz i rozdzielczości 18 bitów,
- 2 przetworniki cyfrowo-analogowe o częstotliwości próbkowania około 100 kHz i rozdzielczości 18 bitów,
- układ Field Programmable Gate Array (FPGA) Cyclone V, sterujący wspomnianymi przetwornikami oraz gwarantujący dodatkowe cyfrowe sygnały wejścia/wyjścia.

Skalowalność systemu opartego o takie moduły jest bardzo nieefektywna. Już sama próba zestawienia eksperymentu z wykorzystaniem pułapki jonowej, umożliwiającej transport jonów, powoduje, że zapotrzebowanie na kanały przetworników cyfrowo-analogowych wzrasta niewspółmiernie do pozostałych podzespołów.

Moduły National Instruments

Firma National Instruments stworzyła LabView – graficzne środowisko programistyczne, cechujące się wyjątkową prostotą i intuicyjnością. Firma zapewnia stałe wsparcie m. in. w postaci gotowych modułów sterowników, obsługujących całą komunikację z urządzeniami. Dzięki temu można w łatwy sposób zestawić oraz opisać różnego rodzaju programy z wykorzystaniem zewnętrznego sprzętu pomiarowego czy kontrolnego, jak np. zasilacze lub generatory. Aby zapewnić kompleksowe rozwiązania, National Instruments wprowadziło do swojej oferty wiele dedykowanych dla swojego środowiska kart Peripheral Component Interconnect Express (PCIe) o różnorodnej funkcjonalności w zakresie m. in. akwizycji danych i kontroli. Fizycy kwantowi często sięgają po te rozwiązania, które z ich perspektywy są bardzo wygodne, proste w obsłudze i umożliwiają zminimalizowanie czasu potrzebnego do zestawienia i uruchomienia odpowiedniego sprzętu. Nie dziwi zatem, że informacja o wykorzystaniu kart NI oraz środowiska LabView pojawia się w wielu publikacjach [29][1][4].

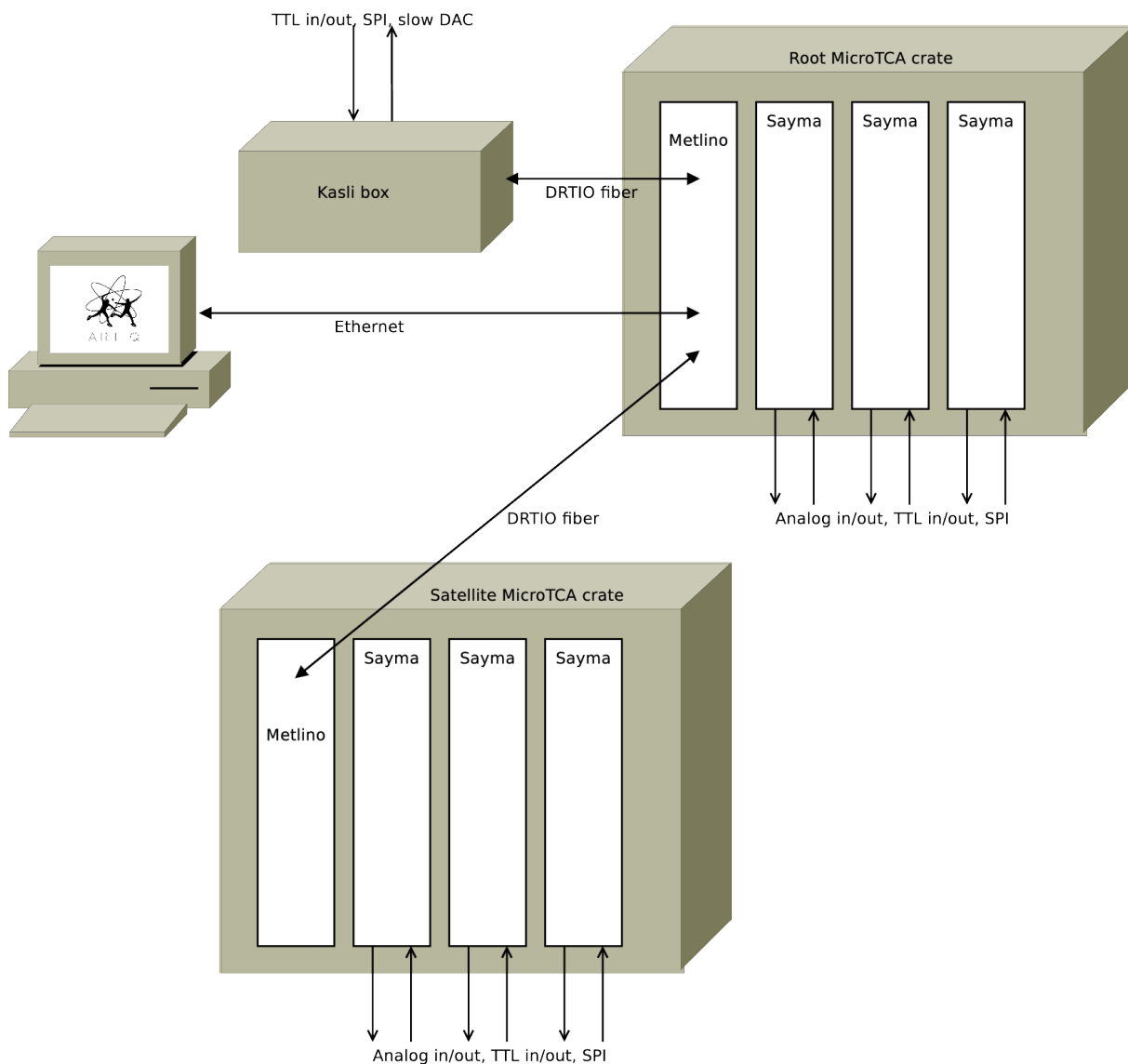
Niezależnie jednak od funkcjonalności i parametrów kart, oferowanych przez firmę, cały system posiada dwie zasadnicze wady. Po pierwsze programy napisane w środowisku LabView charakteryzują się dużą latencją. Dla wielu rodzajów badań (między innymi dla przedstawionych w podanych publikacjach) nie stanowi to problemu. Jednak w szybko rozwijającej się technologii kwantowej opieranie się wyłącznie na takich rozwiązaniach programowo-sprzętowych nie jest już wystarczające. Po drugie należy pamiętać, że LabView to środowisko komercyjne – niskopoziomowy kod nie jest dostępny, zaś bez zakupu odpowiedniej licencji nie można odczytać programów nawet do analizy.

ARTIQ oraz Sinara

Advanced Real-Time Infrastructure for Quantum physics (ARTIQ) to projekt systemu sterowania i kontroli doświadczeń kwantowych, oparty na otwartej licencji, będący próbą utworzenia standardyzowanego środowiska [22]. Projekt zapoczątkowała firma M-Labs we współpracy z National Institute of Standards and Technology (NIST). Wykorzystuje język wysokiego poziomu, bazujący na Pythonie, do opisywania skomplikowanych eksperymentów. Kod jest kompilowany i wykonywany na układach FPGA, gwarantując synchronizację czasową działań na poziomie nanosekund oraz opóźnienie poniżej mikrosekundy. Komunikacja z komputerem PC jest ustanawiana przy wykorzystaniu właśnie języka Python.

Cechy te gwarantują użyteczność projektu ARTIQ w praktycznie wszystkich laboratoriach wykorzystujących układy FPGA – otwarta licencja pozwala na prosty przepływ informacji i modyfikowanie już istniejących doświadczeń przez różne zespoły. Wraz z rozwojem badań pojawiła się jednak potrzeba coraz większej skalowalności, a co za tym idzie – modularnego podejścia do sprzętu. Istniejące urządzenia, tworzone bezpośrednio przez fizyków, stały się elementem ograniczającym możliwości systemu ARTIQ. Tak zrodziła się idea powstania bliźniaczego projektu Sinara – sprzętowego systemu kontroli w pełni kompatybilnego z oprogramowaniem ARTIQ. Założeniem ponownie było użycie otwartej licencji dla wszystkich podprojektów, umożliwiające łatwe zestawienie identycznych systemów, a także elastyczność modułów, pozwalająca na konfigurację ich parametrów w szerokim zakresie.

Schemat blokowy systemu został przedstawiony na rys. 1.5. Składa się z dwóch równoległych zestawów opartych na różnych systemach modułowych. Pierwszy, przeznaczony dla krat MicroTCA, zawiera moduły w formacie Advanced Mezzanine Cards (AMC), generujące precyzyjne sygnały o wysokiej częstotliwości. Równolegle wykorzystywana może być kasetka Eurocard z kartami w forma-



Rysunek 1.5: Schemat blokowy pełnego systemu Sinara [33].

cie 3U, stanowiąca gotowe rozwiązanie dla mniej wymagających doświadczeń kwantowych, gwarantująca jednocześnie dużo niższą cenę. Użytkownik komunikuje się poprzez połączenie Ethernet z głównym modułem zarządzającym (Metlino – jeśli wykorzystywana jest krata MicroTCA lub Kasli – w przypadku wykorzystania wyłącznie kraty o tej samej nazwie), który stanowi multiplexer danych dla wszystkich podsystemów. Generuje on także sygnał synchronizujący przesyłany płytą tylną (wewnątrz MicroTCA) lub przewodami pomiędzy kolejnymi kratami, umożliwiając rozszerzenie systemu w topologii drzewa. Dzięki użyciu autorskiego protokołu Distributed Real-Time Input/Output (DRTIO), zapewniono deterministyczną i niską latencję, a także synchronizację na poziomie 1 ns oraz dystrybucję czasu.

Obecnie projekt Sinara posiada około 18 gotowych modułów: płyty zarządzające systemem (Metlino i Kasli), posiadające wielokanałowe przetworniki do 1 MS/s (Sampler oraz Zotino), 4 kanały 1 GS/s DDS (Urukul), oraz generator przebiegów arbitralnych (Sayma – 8 kanałów 1,2 GS/s przetworników cyfrowo-analogowych oraz 8 kanałów 125 MS/s przetworników analogowo-cyfrowych). W fazie prototypowania jest kolejnych kilkanaście modułów.

Przedstawiona koncepcja z otwartą licencją na część sprzętową i kody źródłowe oprogramowania zaspokaja różne potrzeby naukowców, umożliwiając zestawienie zarówno prostych i tanich systemów, jak i wysoce wyspecjalizowanych, o precyzji do tej pory nieosiągalnej w tego typu doświadczeniach, stanowiących kolejny krok w kierunku zwiększenia niezawodności bramek kwantowych.

Podsumowanie

Spośród wymienionych systemów szczególnie perspektywiczne są te powstałe jako dedykowany sprzęt dla doświadczeń kwantowych, a więc Yggdrasil oraz Sinara. Projekty te posiadają otwartą licencję na dedykowane oprogramowanie, co jest szczególnie istotne w kontekście możliwości powielania eksperymentów przez inne instytucje – a co za tym idzie, stanowią potencjalnie najkorzystniejsze rozwiązanie dla nowo powstających grup badawczych. Zaletą projektu Sinara jest posiadanie dedykowanych dla konkretnych funkcjonalności modułów, kompatybilnych ze sobą w ramach całego systemu sterowania. Gwarantuje to wyższą elastyczność i prostotę skalowania eksperymentów w porównaniu do stosowania wielofunkcyjnego modułu Yggdrasil.

Z punktu widzenia niniejszej pracy najistotniejszy jest jednak fakt, że żaden z przedstawionych systemów nie stanowi rozwiązania problemu precyzyjnego, szybkiego (rzędu 50-100 MS/s), równoległego sterowania pułapkami jonowymi – parametry oferowanych modułów z wyjściowymi kanałami analogowymi przedstawione zostały w tabeli 1.1.

Zdecydowana większość modułów posiadających większą liczbę kanałów (konieczną do efektywnego skalowania przy sterowaniu dużą ilością pułapek jonowych) posiada częstotliwość generacji próbek na poziomie 1 MS/s lub mniejszą, a więc niewystarczającą do zastosowania w wielokubitowym komputerze kwantowym. Pozostałe moduły z szybszymi przetwornikami posiadają z kolei małą liczbę kanałów (PRO-II-Aout-1/16, Slave-Board). Dodatkowo moduły takie jak Yggdrasil Slave-Board oraz Sayma zapewniają szereg dodatkowych funkcjonalności, niezwiązanych bezpośrednio ze sterowaniem pułapkami jonowymi, co sprawia, że skalowanie systemu poprzez dodawanie kolejnych kart byłoby rozwiązaniem wysoce nieoptymalnym pod względem ekonomicznym. Także precyzja stosowanych przetworników nie spełnia oczekiwanych wartości +/- 1 LSB Differential Non-linearity (DNL).

Tabela 1.1: Przedstawienie dostępnych modułów z wyjściami analogowymi, stosowanych w przedstawionych systemach sprzętowych, wraz z parametrami stosowanych na nich przetworników cyfrowo-analogowych.

System	Moduł	Ilość Kanałów	Rozdzielczość	fs(max.) [MS/s]	DNL [LSB]	Zależność temp. błędu wzmacnienia [ppm/K]
ADwin-PRO-II	Pro II AOut-8/16	8	16	0,3	1	-
ADwin-PRO-II	PRO-II-Aout-1/16	1	16	50	-	-
Yggdrasil	Slave-Board	2	18	0,1	0,75	0,018
		2	16	800	2,1	100
National Instruments	PXIe-7867 ¹	18	16	1	0,5	12,6
Sinara	Sayma	8	16	1200	4,3	114
Sinara	Zotino	32	16	1	1	- ²

¹W przypadku modułów firmy National Instruments przytoczono dane przykładowej płyty zawierającej największą liczbę wyjściowych kanałów analogowych

²Producent stosowanych przetworników podaje wyłącznie parametr zależności temperaturowej zawierającej w sobie dryft liniowości, błędu przesunięcia zera i wzmacnienia: 5 ppm * FSR/K

2. Geneza, cel oraz założenia pracy

2.1. Geneza

Motywacją dla tej pracy była chęć opracowania, dedykowanego dla rozbudowanych eksperymentów kwantowych, modułu na otwartej licencji, rozszerzającego funkcjonalność projektu Sinara o możliwość precyzyjnego sterowania pułapkami jonowymi. Stanowił on odpowiedź na rosnące zapotrzebowanie na urządzenie łączące szybkości działania przetworników cyfrowo-analogowych, precyzję generowanych sygnałów analogowych oraz możliwość skalowania systemu dla złożonych eksperymentów. Dotychczas stosowany sprzęt nie gwarantował spełnienia wszystkich tych cech równocześnie, co zostało wykazane w poprzednim rozdziale. Brak odpowiednich narzędzi widoczny był także w przytaczanych wcześniej publikacjach, które dotyczyły osiągania coraz większej precyzji układów zawierających małą liczbę bramek kwantowych lub badań nad wielokubitowymi pułapkami wyłącznie pod kątem optymalizacji technicznych rozwiązań – nie zaś przeprowadzania bardziej skomplikowanych obliczeń kwantowych z wykorzystaniem pułapek jonowych.

Podstawową funkcją projektowanego układu miało być sterowanie nie tylko efektywnym łapaniem jonów, ale przede wszystkim ich szybkim transportem w trakcie przeprowadzania skomplikowanych obliczeń kwantowych. Stanowić miał także narzędzie pozwalające na opracowywanie nowych algorytmów sterowania modularnymi pułapkami jonowymi, co w przyszłości umożliwiłoby optymalizację projektu dedykowanego układu ASIC.

Głównym założeniem całego procesu projektowego było kierowanie się optymalizacją kosztu zestawienia całego systemu ze względu na przewidywany wysoki koszt jego produkcji. Miało to związek z koniecznością stosowania precyzyjnych układów przy jednoczesnym zachowaniu wszystkich wymaganych parametrów technicznych. Stosowana w dalszej części pracy nazwa modułu „Shuttler” pochodzi od używanego w anglojęzycznych publikacjach słowa „shuttling”, opisującego wahadłowy transport jonów w pułapce liniowej, które poruszają się „tam i z powrotem”.

2.2. Cele pracy

Celem niniejszej pracy było opracowanie koncepcji metody sterowania pułapkami jonowymi, pozwalającej na przeprowadzanie doświadczeń kwantowych na dużą skalę (tj. przy zestawieniu pułapek wymagających około 1000 kanałów), gwarantującej jednocześnie wysoką precyzję generowanych sygnałów analogowych oraz ich szybkość. Miało to pozwolić na efektywny czasowo transport jonów, zwiększający możliwą do wykonania ilość operacji w trakcie ograniczonego czasu życia poszczególnych stanów kwantowych. Kolejnym krokiem było zaprojektowanie w postaci gotowych schematów logicznych oraz projektu obwodu drukowanego modułu Shuttler, pozwalającego wraz istniejącymi już komponentami na realizację systemu zgodnego z opracowaną koncepcją i zoptymalizowanego pod względem kosztów jego zestawienia.

2.3. Założenia techniczne

Ze względu na bezcelowość tworzenia wyłącznie niezależnego urządzenia, podstawowym założeniem było dostosowanie płyty Shuttler do pracy jako jeden z modułów projektu Sinara, wraz ze

wsparciem dla części programowej ARTIQ. Wybrane przetworniki cyfrowo-analogowe musiały spełniać ściśle określone wymagania:

- 16 bitów rozdzielczości,
- błąd nieliniowości różniczkowej maksymalnie +/- 1 LSB,
- błąd nieliniowości całkowitej maksymalnie +/- 1 LSB,
- częstotliwość próbkowania co najmniej 50 MHz,
- możliwie niska zależność temperaturowa.

Założeniem było zaimplementowanie możliwie jak największej liczby kanałów wybranych przetworników w celu zminimalizowania kosztu zestawienia potencjalnego 1000-kanałowego systemu sterowania pułapkami jonowymi. Ponieważ liczba kanałów przypadających na moduł stanowiła kluczowy czynnik decydujący o końcowej cenie systemu, założono także możliwość zastosowania wszelkich niestandardowych rozwiązań technicznych, pozwalających na zwiększenie tej liczby pod warunkiem zapewnienia niezawodności działania całego modułu. W ramach zachowania odpowiedniej precyzji sterowanego sygnału konieczne było zapewnienie także deterministycznej latencji pojawiania się zmian sygnału analogowego oraz rozdzielczości czasowej całego układu na poziomie 1 ns. Użytkownik musiał mieć także możliwość wykorzystania urządzenia Shuttler bez posiadania innych modułów projektu Sinara, aby opracowany moduł stanowił użyteczne narzędzie także w eksperymentach na małą skalę.

3. Koncepcja architektury i dobór komponentów

Pierwszy krok polegał na określeniu uszczegółowionych wymagań oraz koniecznych do zastosowania elementów, których wykorzystanie było konsekwencją otrzymanych założeń technicznych. Funkcjonowanie modułu Shuttler jako składowej projektu Sinara wymuszało przyjęcie jednego z dwóch formatów mechanicznych, zgodnych z ogólną architekturą przedstawioną w rozdziale 1.3:

1. karty w formacie Eurocard Extension Modules (EEM) [15], przeznaczonej do pracy w tzw. Kasli Box, a więc sterowanej przez moduł Kasli,
2. karty w formacie AMC z opcją rozszerzenia o moduł Rear Transition Module (RTM), pracującej w kracie MicroTCA.4.

Kompatybilność z częścią programową ARTIQ wymagała obecności układu FPGA wyposażonego w pamięć Random-Access Memory (RAM). Jego funkcją było dokonywanie odpowiednich obliczeń w celu wyznaczenia wartości generowanych próbek.

Konsekwencją wspomnianych oczekiwań kompatybilności oraz synchronizacji urządzeń w obrębie systemu była konieczność zapewnienia połączenia sygnałów DRTIO do nadajników i odbiorników gigabitowych układu FPGA, a także stosowanie układu tłumiącego jitter odzyskanego sygnału zegarowego (z sygnału DRTIO), bądź precyzyjnego źródła zegarowego (w tym wypadku konieczna byłaby także znajomość różnicy faz względem sygnału źródłowego). W systemie Sinara sygnał DRTIO doprowadzany jest na kilka sposobów:

- bezpośrednie połączenie kablowe poprzez złącze Small Form-factor Pluggable (SFP) lub Quad SFP (QSFP) do urządzeń zarządzających daną kratą: Metlino bądź Kasli,
- w kracie MicroTCA przez połączenia na płycie tylnej (źródłem jest moduł zarządzający kratą).

Połączenie kablowe w podstawowej konfiguracji przewidziane zostało wyłącznie dla modułów zarządzających w celu włączenia danej kraty do systemu. Ponieważ jednak założeniem modułu Shuttler była możliwość uruchomienia go jako niezależnego urządzenia, konieczne było umieszczenie złącza SFP bądź QSFP. Drugie takie złącze należało uznać za wysoce zalecane, ponieważ umożliwiłoby to potencjalne zestawienie kilku modułów poza kratą przy jednoczesnym zapewnieniu synchronizacji.

Przyjęto także założenie, że w trakcie procesu projektowego należy w możliwie dużym stopniu wykorzystywać gotowe sprzętowe bloki funkcjonalne oraz konkretne układy scalone zawarte na już istniejących modułach projektu Sinara. Takie podejście pozwoliło na ustandaryzowanie części rozwiązań sprzętowych w obrębie projektu Sinara, czego dalszym efektem będzie zminimalizowanie czasu potrzebnego na opracowanie oprogramowania i plików projektowych dla układów FPGA przez pozostałych członków zespołu, a także zmniejszenie kosztów produkcji pełnych systemów kontroli doświadczonych kwantowych.

Na podstawie powyższych wymagań określono zadania projektowe, prowadzące do zrealizowania celu całej pracy. Opracowano następujący plan działania:

1. Dobór środowiska sprzętowego oferowanego przez projekt Sinara.
2. Opracowanie koncepcji połączeń pomiędzy projektowanym modułem Shuttler a pozostałą częścią systemu Sinara w celu stworzenia ogólnej architektury systemu sterowania pułapkami jonowymi.
3. Dobór przetwornika cyfrowo-analogowego, będącego kluczowym elementem modułu Shuttler.

4. Opracowanie wstępnej architektury modułu Shuttler wraz z wyceną potencjalnego 1000-kanalowego systemu.
5. Próba opracowania koncepcji niestandardowych połączeń pomiędzy układem FPGA a przetwornikami wraz z fizyczną weryfikacją (w celu zmniejszenia kosztu systemu poprzez zwiększenie ilości kanałów przypadających na pojedynczy moduł).
6. Opracowanie ostatecznej wersji modułu Shuttler w postaci schematów elektrycznych i projektu obwodu drukowanego oraz wycena zestawienia 1000-kanalowego systemu sterowania pułapkami jonowymi.
7. Ocena osiągniętej optymalizacji kosztu systemu.

Plan opracowania niestandardowego połączenia pomiędzy przetwornikami a układem FPGA wynikał z analizy dokonanej przez fizyków w trakcie określania wymaganych parametrów – wstępnie wyselekcjonowane układy cechowały się równoległym interfejsem, wymagającym wykorzystania wielu wyprowadzeń FPGA. Z tego względu od początku zakładano możliwość wystąpienia problemu polegającego na uzyskaniu stosunkowo małej liczby kanałów w pojedynczym module, co negatywnie wpłynęłoby na koszt 1000-kanalowego systemu.

3.1. Dobór środowiska sprzętowego

Już w pierwotnym założeniu projektu Sinara krata MicroTCA miała być narzędziem służącym do zestawienia modułów o wyższych wymaganiach technicznych, dotyczących szybkości pracy urządzeń, a także oczekiwanej precyzji. Dlatego teoretycznie oczywistym wyborem było przyjęcie formatu AMC z możliwością wykorzystania rozszerzenia RTM. Gwarantowałoby to uproszczenie połączeń rozważanego w pracy 1000-kanalowego systemu sterowania (dzięki obecności płyty tylnej kraty zamiast połączeń kablowych między modułami) oraz zwiększenie przepustowości przesyłanych danych (kwestia parametrów modułów Metlino oraz Kasli). Sama krata MicroTCA to jednak znaczący koszt (rzędu 30 tys. zł), dlatego przed podjęciem ostatecznej decyzji konieczne było wykazanie, że nie istnieje możliwość wykorzystania dużo tańszego środowiska Kasli Box wraz z kartami EEM.

Koncepcja modułów EEM zakładała tworzenie stosunkowo prostych urządzeń, zasilanych i sterowanych przez moduł zarządzający Kasli. Dostępne interfejsy dla połączenia EEM oraz Kasli to wyłącznie I2C oraz 8 linii różnicowych w standardzie Low-Voltage Differential Signaling (LVDS). Już wstępna analiza fizyków definiujących wymagania dla przetworników cyfrowo-analogowych modułu Shuttler wykazała, że konieczne będzie stosowanie przetwornika o równoległym interfejsie. Konieczne byłoby zatem stosowanie układów FPGA mających stanowić deserializer dla poszczególnych kanałów, zaś za obliczenia odpowiadałby Artix-7 znajdujący się na module Kasli – zgodnie jednak z ustaleniami z fizykami układ ten jest wyposażony w zbyt małą ilość pamięci RAM oraz wewnętrznych zasobów, aby rozwiązanie to mogło sprawdzić się w praktyce. Drugą opcją mogłoby być założenie, że układ FPGA na module Shuttler EEM bezpośrednio odpowiadałby za dokonywanie odpowiednich obliczeń dla kanałów znajdujących się na tej samej karcie. W tym wypadku jednak, konieczna byłaby synchronizacja tego układu z resztą systemu poprzez doprowadzenie sygnału DRTIO. Interfejsy z modulem nadrzędnym Kasli na to nie pozwalają, a więc skutkowałoby to w praktyce koniecznością stosowania wyprowadzenia przez panel zewnętrzny. W ten sposób osiągnięta zostałaby pełna niezależność płyty Shuttler EEM – moduł nadrzędny Kasli stałby się zbędny, jego wydajność prądowa dla rozszerzeń wynosząca w sumie maksymalnie 5 A dla głównego zasilania

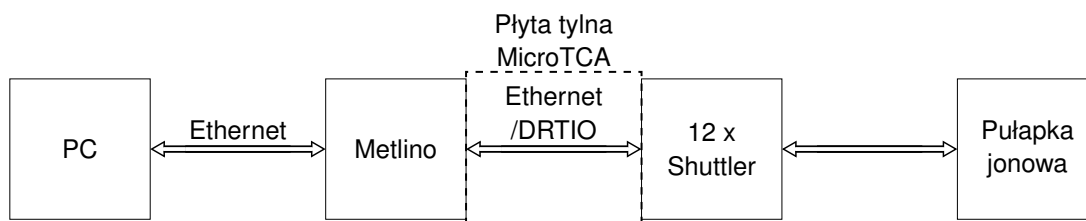
12 V z pewnością okazałaby się niewystarczająca dla 8 modułów Shuttler wyposażonych w szybkie przetworniki oraz stosunkowo duże układy FPGA. Z kolei dane i rozkazy byłyby przesyłane bezpośrednio sygnałem DRTIO zapewniającym jednocześnie synchronizację. Dlatego ostatecznie można było odrzucić opcję wykorzystania "Kasli Box" z dwóch powodów. Po pierwsze taka architektura nie stanowiłaby części projektu Sinara – byłaby wyłącznie niezależnym urządzeniem, bądź początkiem opracowywania nowego systemu, co zaprzecza celom niniejszej pracy. Po drugie takie moduły nie pozwalałyby na proste skalowanie, a więc nie stanowiłyby rozwiązania problemu zestawienia 1000-kanalowego systemu sterowania pułapkami jonowymi. Dlatego też, pomimo znaczących kosztów, konieczne było podjęcie decyzji o zaprojektowaniu modułów do pracy w kracie MicroTCA.

3.2. Koncepcja systemu

Kolejnym krokiem projektowym było opracowanie schematu blokowego skalowalnej architektury pozwalającej na zestawienie 1000-kanalowego systemu sterowania pułapkami jonowymi. Ze względu na chęć rozszerzenia funkcjonalności projektu Sinara, należało wykorzystać istniejące już moduły, dodając do nich projektowany moduł Shuttler. W standardowym systemie wykorzystującym kraty MicroTCA użytkownik poprzez interfejs Ethernet steruje nadrzędnym modułem Metlino. Względem niego synchronizowane są wszystkie pozostałe urządzenia. Jego funkcjonalność zakłada zarządzanie daną kratą MicroTCA oraz skalowanie systemu poprzez sygnały DRTIO połączone kablami do podrzędnych modułów Metlino (zgodnie z rys. 1.5). Poprzez płytę tylną kraty do płyt AMC realizowane są interfejsy DRTIO oraz Ethernet. Były to więc dwa możliwe źródła danych dla projektowanego modułu Shuttler.

Połączenie sygnałów analogowych do obwodów pułapki zrealizowane zostało poprzez wyprowadzenie linii przez panel przedni i doprowadzenie kablem. Była to jedyna możliwość zestawienia układów poprzedzających elektrody z systemem zawartym w kracie MicroTCA. Konieczne było zastosowanie przewodów o kontrolowanej impedancji i gwarantujących możliwie niskie szумы oraz przesłuchy, jednak sam dobór konkretnych złączy na płycie był zależny od osiągniętej liczby kanałów oraz ewentualnie wymaganych wymiarów (w przypadku potrzeby zaoszczędzenia miejsca na obwodzie drukowanym).

Docelowe połączenia w obrębie systemu (z pominięciem skalowania poprzez zastosowanie kolejnych krat MicroTCA, które zostało ukazane na rys. 1.5) przedstawione zostały na rys. 3.1.



Rysunek 3.1: Schemat połączeń między poszczególnymi składowymi docelowego systemu.

Największy skalowalny blok rozważanego 1000-kanalowego systemu składał się zatem z kraty MicroTCA, która przy pełnym wyposażeniu gwarantowała zasilanie dla wszystkich modułów, chłodzenie oraz płytę tylną do połączeń między poszczególnymi podzespołami. Zawierała 12 miejsc na karty w formacie AMC – przeznaczone na moduły Shuttler. Każda krata posiadała także moduł zarządzający Metlino, będący dystrybutorem danych, rozkazów oraz sygnału synchronizującego.

Pozwalał on także na dodanie kolejnych krat – wymagane było wyłącznie połączenie dwóch Metlino przy wykorzystaniu SFP. Liczba koniecznych do zastosowania krat była ściśle związana z osiągniętą liczbą kanałów na pojedynczym module Shuttler – w związku z tym na obecnym etapie projektowym opracowana została tylko ogólna architektura systemu, wciąż nieznany pozostawał jego fizyczny rozmiar.

3.3. Przetwornik cyfrowo-analogowy dla modułu Shuttler

Dobór przetwornika cyfrowo-analogowego był kluczową decyzją konieczną do podjęcia przed opracowaniem wstępnej architektury modułu. Wynikało to z faktu postawienia wysokich wymagań dotyczących jego parametrów. Choć jednym z celów projektowych była optymalizacja kosztów, płyta Shuttler musiała spełniać postawione wymogi bez względu na cenę, aby stanowić narzędzie umożliwiające poszerzenie zakresu przeprowadzanych eksperymentów kwantowych. Dlatego też konieczne było podejście polegające na dostosowaniu architektury do konkretnego przetwornika. Dokonano przeglądu dostępnych układów, wykorzystując do filtracji wyników parametry: rozdzielczość 16 bitów, DNL mniejsze bądź równe w wartości bezwzględnej 1 LSB oraz częstotliwość próbkowania co najmniej 50 MHz. Otrzymane wyniki przedstawiono w tabeli 3.1

Tabela 3.1: Parametry dostępnych przetworników cyfrowo-analogowych, które spełniały podstawowe warunki projektowe.

Model	fs(max.) [MS/s]	Liczba kanałów	Moc [W]	DNL [LSB]	INL [LSB]	Zależność temp. błędu wzmocnienia [ppm/K]	Koszt [\$]	Interfejs
AD9726	400	1	0,6 (400 MHz)	±0,5	±1	10	58	LVDS
LTC1668	50	1	0,2 (25 MHz)	±1	-	30	15	CMOS
LTC2000	2500	1	0,4 (250 MHz)	±0,5	±1	5	96	LVDS
MAX5895	500	2	0,83 (250 MHz)	±1	±3	110	13	CMOS
MAX5898	500	2	0,83 (250 MHz)	±1	±3	110	13	LVDS
MAX5888	500	1	0,18 (250 MHz)	±1,3	±2,6	50	45	LVDS

Podane wartości DNL oraz Integral Nonlinearity (INL) są wartościami typowymi deklarowanymi przez producenta. Źródła danych: [10], [36], [11], [24], [25].

Pierwszy z wymienionych przetworników – AD9726 – był już wykorzystywany w doświadczeniach kwantowych przez współpracujących w projekcie fizyków i spełniał wszystkie wymagane przez nich kryteria związane z precyzją działania. W zestawieniu charakteryzuje się on wyjątkowo niską zależnością temperaturową, z którą równać się może wyłącznie układ LTC2000, ale posiada jednocześnie dużo niższą cenę. Niestety producent zaznacza na swojej stronie [9], że układ ten jest niezalecany dla nowych projektów, ze względu na planowane zakończenie jego produkcji.

Drugim z kolei rozważanym przetwornikiem był układ LTC1668. Posiada on minimalnie gorsze parametry od wspomnianego AD9726, jednak cechuje się dużo niższą ceną i interfejsem przesyłu danych w standardzie CMOS, co było zaletą w kontekście osiągnięcia możliwie dużej liczby kanałów na pojedynczym module. Niewątpliwą wadę stanowiła maksymalna częstotliwość próbkowania wynosząca 50 MHz. Zawierała się ona w wymaganym zakresie, jednak należało pamiętać o ciągle rosnących oczekiwaniach. Dowodem jest fakt, że już w trakcie badań nad niniejszym projektem

zgłosił się instytut zainteresowany modulem Shuttler, o ile będzie on pozwalał na próbkowanie z częstotliwością 250 MHz.

Zaprezentowane układy firmy Maxim Integrated zostały odrzucone ze względu na zbyt dużą zależność temperaturową. Obawiano się, że problemy związane z nagrzewaniem się całej płyty wraz z otoczeniem mogą drastycznie wpłynąć na precyzję działania przetworników, która jest kwestią kluczową w badaniach kwantowych.

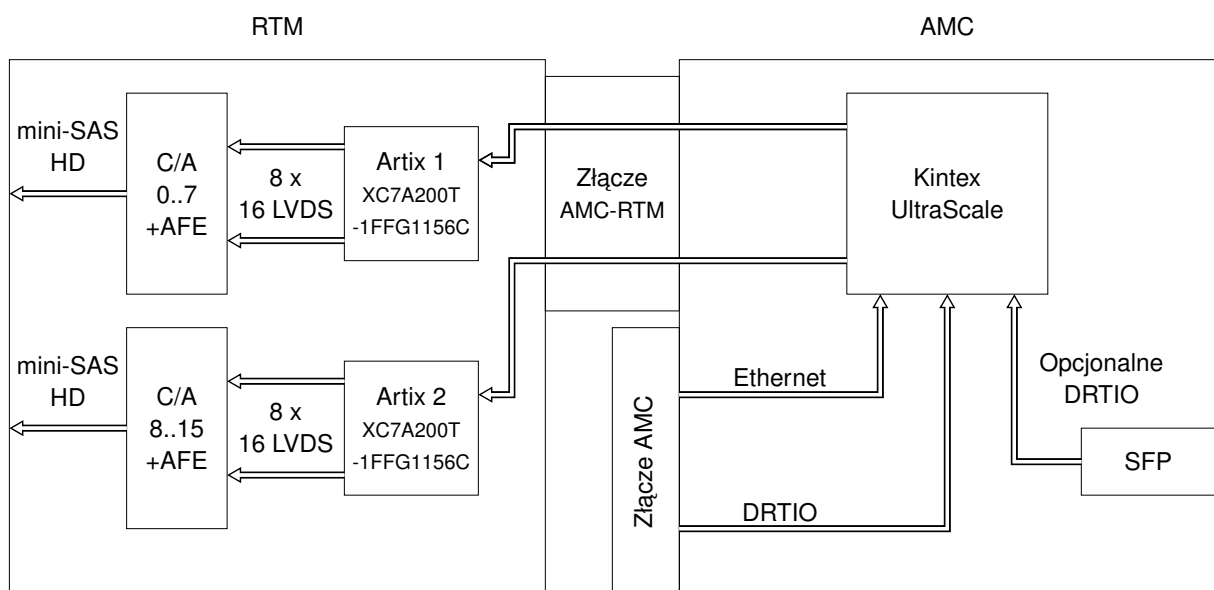
Ostateczny wybór padł na przetwornik cyfrowo-analogowy LTC2000 w wersji 16 bitowej. Posiada on najlepsze parametry (w tym 5 ppm/K zmiany błędu wzmocnienia, nieosiągalne przez inne układy), jednocześnie gwarantując szeroki zakres częstotliwości próbkowania. Wartość maksymalna wynosi 2,5 GS/s, zatem wykorzystanie tego układu do generacji sygnału z częstotliwością próbkowania rzędu kilkuset MHz może wydawać się rozwiązaniem nieoptymalnym. W praktyce pozostałe przetworniki nie były jednak w stanie połączyć wymagań dotyczących precyzji oraz szybkości działania, które w przypadku większości urządzeń wzajemnie się wykluczają.

3.4. Pierwsze koncepcje modułu Shuttler

Głównym problemem w doborze odpowiedniej architektury modułu Shuttler była potrzeba zawarcia jak największej liczby kanałów przetworników cyfrowo-analogowych przy jednoczesnej konieczności wykorzystywania przetwornika o równoległym interfejsie. Dodatkowo interfejs ten korzystał z linii różnicowych (w standardzie LVDS), które dla każdego bitu wymagają dwóch wyprowadzeń układu sterującego. Opracowane zostały dwie ogólne koncepcje schematów blokowych modułu:

Wersja z modułami AMC i RTM

Pierwsza koncepcja zakładała wykorzystanie zarówno modułu AMC, jak i karty rozszerzenia RTM, możliwej do implementacji w systemie pracującym w kracie MicroTCA.4. Ogólny schemat blokowy tej koncepcji przedstawiony został na rys. 3.2. Zakładała ona obecność 3 układów FPGA. Pierwszy



Rysunek 3.2: Schemat blokowy modułów Shuttler wykorzystujących format kart AMC + RTM.

to układ Kintex UltraScale, dokonujący wszelkich obliczeń związanych z generacją odpowiednich sygnałów dla pułapki jonowej, a także za komunikację z resztą systemu. Kolejne dwa układy to Artix-7, których rolą byłoby stanowienie deserializerów danych, pochodzących z pierwszego FPGA i ich rozsyłanie do poszczególnych przetworników cyfrowo-analogowych. W ramach modułu AMC można było wykorzystać istniejącą płytę Sayma AMC, gdyż posiada wystarczająco duży układ FPGA i jest w pełni kompatybilna z resztą urządzeń projektu Sinara. Płyta RTM (poza wspomnianym rozdzielaniem danych pomiędzy poszczególne kanały) generowałaby wtedy także precyzyjny sygnał zegarowy (na podstawie sygnału doprowadzonego przez płytę tylną RF dla modułów RTM lub dostarczonego z zewnątrz) oraz sterowałaby sygnałem analogowym. W przypadku wykorzystania układów Artix-7 w wersji z największą możliwą ilością wyprowadzeń, możliwe było uzyskanie wyniku 16 kanałów na pojedynczym slocie zajmowanym w kasecie.

Przedstawiona koncepcja architektury AMC+RTM została jednak odrzucona. Głównym powodem była obawa, czy przy takim przepływie danych uda się zachować ściśle deterministyczną latencję, która była wymagana do sterowania pułapkami jonowymi. Wadą był także sam wzrost latencji.

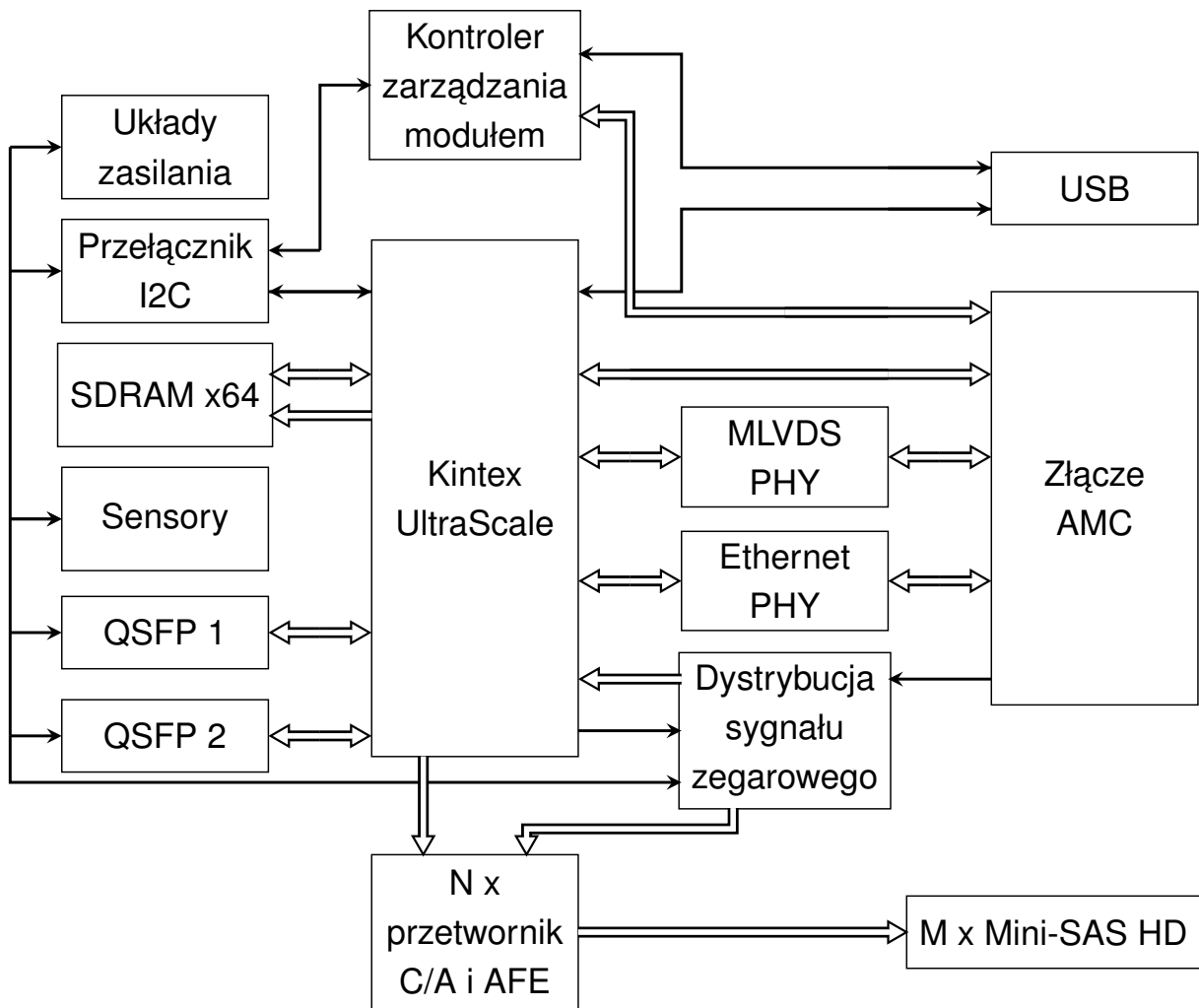
Moduł Shuttler w wersji AMC

Po odrzuceniu pierwszej koncepcji, postawiono nowe wymaganie, aby dane były przekazywane do przetworników bezpośrednio przez układ FPGA generujący cyfrowe próbki. Z tego względu konieczne było podjęcie decyzji o wykorzystaniu wyłącznie karty AMC. Połączenie modułów AMC i RTM posiadało bowiem zbyt mało linii, aby umożliwić sterowanie z karty AMC przetwornikami, znajdującymi się na karcie RTM, wykorzystując jednocześnie w maksymalnym stopniu dostępne wyprowadzenia układu FPGA. Umieszczenie z kolei przetworników na obydwu płytach także nie stanowiło rozsądnego rozwiązania, gdyż utrudniłoby zaprojektowanie obwodów drukowanych z zachowaniem odpowiednich zależności czasowych pomiędzy przetwornikami, a także wymuszałyby wyprowadzenie sygnałów analogowych poprzez złącza z obydwu stron kasety.

Koncepcja samodzielnego modułu AMC została zatwierdzona przez fizyków, w związku z czym opracowano bardziej szczegółowy schemat blokowy, który przedstawiony został na rys. 3.3. Użycie większości poszczególnych bloków funkcjonalnych było związane z wymaganiami dotyczącymi kompatybilności z modułami projektu Sinara oraz ARTIQ i zapewniało prawidłowe funkcjonowanie modułu w kracie MicroTCA. Złącze AMC stanowiło połączenie całego modułu z płytą tylną kraty, umożliwiające komunikację z głównym modułem zarządzającym kratą, sąsiednimi modułami oraz dystrybucję sygnału zegarowego.

Blok Module Management Control (MMC) stanowił mikroprocesor odpowiadający za komunikację z modułem MicroTCA Carrier Hub (MCH) w celach identyfikacji modułu oraz jego zarządzania (uruchamianie zasilania, kontrola temperatury itp.). Powszechnym wymaganiem dla modułów mogących potencjalnie pracować poza kratą MicroTCA była obecność złącza SFP – bądź jak w wypadku płyty Shuttler – QSFP w celu dystrybucji sygnału DRTIO. Był on konieczny, aby zapewnić synchronizację wszystkich podsystemów na poziomie nanosekund.

Połączenie Universal Serial Bus (USB) zapewniało zestawienie kanałów do diagnostyki płyty. Pozostałe bloki związane były z główną funkcjonalnością modułu Shuttler. Double Data Rate (DDR)3 Synchronous Dynamic Random-Access Memory (SDRAM) dla układu FPGA umożliwiała przeprowadzanie obliczeń koniecznych do generowania prawidłowych sygnałów analogowych. Ze względu



Rysunek 3.3: Schemat blokowy modułów Shuttler AMC.

na oczekiwaną synchronizację wymagane było umieszczenie bloku tłumiącego jitter sygnału zegarowego oraz precyzyjnych układów dystrybuujących oczyszczony sygnał zegarowy do przetworników i samego FPGA. Blok przetworników miał za zadanie generować sygnały do prostych torów analogowych, składających się ze wzmacniaczy, jednocześnie translujących sygnał prądowy na sygnał napięciowy. Dalej przekazywane były one do złącza Mini Serial Attached Small Computer Systems Interface High Density (Mini-SAS HD) i wyprowadzane na zewnątrz kasety. Taka koncepcja toru analogowego wymagała stosowania dalszych obwodów po stronie układów pułapki jonowej. Niemożliwe było bowiem generowanie na module Shuttler sygnału bezpośrednio podłączonego do elektrod – każdy model pułapki jonowej wymaga stosowania innych wartości napięć, dlatego konieczne było zastosowanie uniwersalnego podejścia.

3.5. Analiza kosztów dla 1000 kanałowego systemu

Jednym z głównych celów niniejszej pracy było dokonanie badań, umożliwiających potencjalne zmniejszenie rozmiarów i kosztów zestawienia wielokanałowego systemu sterowania pułapkami jonowymi z możliwością transportu jonów. Aby móc w przyszłości dokonać jakiegokolwiek oceny jakościowej tych badań, konieczna była analiza kosztów potencjalnego systemu, wykorzystującego

moduł Shuttler o specyfikacji zgodnej z zaproponowaną architekturą, wykorzystującą standardowe połączenia pomiędzy układem FPGA a przetwornikami LTC2000. Dla potrzeb analizy przyjęto, że rozpatrywany system zawierać powinien około 1000 kanałów. Następnie dokonano obliczeń kosztów samego sprzętu, gdzie w skład systemu wchodziły kraty MicroTCA (wyposażone w układy zasilania oraz system chłodzenia), moduły zarządzające kratą Metlino (po jednym na kratę) oraz moduły Shuttler (po 12 sztuk na kratę). Ich ceny (zakupu elementów, produkcji płyty i montażu) przedstawiono w tabeli 3.2.

Tabela 3.2: Zestawienie kosztów elementów składowych systemu sterowania pułapkami jonowymi

Element systemu	Orientacyjny koszt [tys. zł]
Krata MicroTCA	34,2
Metlino MCH	8,5
Sayma AMC*	8,5

*Koszt modułu Sayma AMC podany został jako wartość wyjściowa do szacunkowych obliczeń kosztu modułu Shuttler.

Shuttler był jedynym modułem, który nie został jeszcze w pełni opracowany, a więc nie posiadał dokładnych danych, dotyczących kosztów jego produkcji. Dlatego jako bazę przyjęto koszt produkcji płyty Sayma AMC, posiadającej wiele podobnych bloków funkcjonalnych (związanych z wymaganiami projektu Sinara). Różniły się one przede wszystkim elementami konfigurującymi poszczególne układy scalone i dostosowującymi je do konkretnych zadań czy też poziomów napięciowych. Następnie kwotę tę skorygowano, odejmując koszty zakupu bardziej znaczących cenowo elementów, których nie będzie na płycie Shuttler (są to kwoty, związane z zakupem np. złączy dla FPGA Mezzanine Card (FMC), RTM czy większej ilości SDRAM) i dodając cenę układów charakterystycznych dla projektowanego modułu, a więc przetworników czy wzmacniaczy analogowych. Należy podkreślić, że obliczona kwota była wartością wyłącznie orientacyjną. Zmienną pozostawała cena układu FPGA. Preferowany model to KU040 w największej obudowie (ze względu na jego wykorzystywanie w innych projektach i związaną z tym korzystną ceną i znajomością samego układu), o prędkości -1 i temperaturowym zakresie C (0-85 °C). Możliwe było wykorzystanie większych modeli, gdyby mogło to skutkować znaczącym zmniejszeniem kosztów produkcji całego systemu. Zmiana mogła dotyczyć jedynie wykorzystania większego układu FPGA z rodziny Kintex UltraScale – zastosowanie innej rodziny FPGA wygenerowałoby bowiem dużo większe koszty zarówno finansowe, jak i czasowe potrzebne na rozwój oprogramowania. Opracowywany system sterowania pułapkami jonowymi miał stanowić formę przejściową, umożliwiającą dalsze badania, prowadzące do utworzenia dedykowanych układów ASIC. Rozwiązanie zatem musiało być z jednej strony bezkompromisowe pod względem osiąganego precyzji, a z drugiej musiało gwarantować możliwie krótki czas realizacji i wdrożenia w eksperymentach kwantowych.

Dobór układu FPGA implikował liczbę kanałów w obrębie pojedynczego modułu. Obsługa jednego przetwornika, zgodnie z jego specyfikacją, wymagała wykorzystania 34 wyprowadzeń wejścia/wyjścia (17 par różnicowych). Ze względu na rygory czasowe nie wskazano było wykorzystywanie wyprowadzeń pochodzących z różnych banków (każdy posiada 52 wyprowadzenia). Dlatego też liczba banków dostępnych w danym układzie pomniejszona o 4 banki (wykorzystywane

na potrzeby obsługi pamięci oraz sterowania poszczególnych podzespołów) odpowiadała końcowej liczbie kanałów. Niemożliwe było potencjalne zwiększenie tej liczby poprzez zastosowanie dwóch układów FPGA. Przestrzeń zajmowana przez Kintex UltraScale (nawet dla najmniejszego preferowanego układu z obudową FFVA1156) wraz z wymaganą do obliczeń pamięcią oraz podwojoną liczbą przetworników byłaby zbyt duża. W tabeli kosztów uwzględniono jedynie układy o najniższej cenie dla danej liczby banków wyprowadzeń, za najmniejszy możliwy układ przyjmując preferowany XCKU040-1FFVA1156C. Wzięto także pod uwagę skalę ilości kupowanych układów FPGA oraz przetworników. Przykładowo cena za FPGA może spaść do 55% wartości pojedynczego układu przy zakupie minimum 100 sztuk. Ostateczne wyniki obliczeń całkowitych kosztów przykładowego systemu 1000-kanałowego przedstawione zostały w tabeli 3.3.

Tabela 3.3: Koszt zakupu i produkcji sprzętu dla systemu sterowania pułapkami jonowymi, opartego o moduł Shuttler zgodnego z pierwotną koncepcją architektury w zależności od zastosowanego układu FPGA

Układ FPGA	Liczba kanałów w module	Koszt modułu Shuttler [tys. zł]	Liczba krat MicroTCA	Koszt systemu [tys. zł]	Liczba kanałów w systemie	Cena za kanał [zł]
XCKU040-1FFVA1156	6	10,5	14	2 366,5	1008	2 347
XCKU060-1FFVA1517	8	13,4	11	2 249,5	1056	2 130
XCKU085-1FLVB1760	9	16,3	10	2 387,9	1080	2 211
XCKU115-1FLVA2104*	12	24,9	7	2 396,2	1008	2 377

W tabeli koszt pełnego systemu zakłada pełne wyposażenie krat MicroTCA. Przyjęty koszt układów FPGA wynosi 55% ceny jednostkowej podawanej przez stronę Octopart [30].

*koszt dla XCKU115-1FLVA2104 ustalony został na poziomie 70% ceny podawanej przy zakupie pojedynczej sztuki, ze względu na nieprzekroczenie ilości 100 sztuk.

Najlepszym cenowo wariantem okazał się ten z wykorzystaniem układu XCKU060-1FFVA1517. Jego koszt wyniósłby 2,25 mln zł, składałby się z 11 krat i posiadałby 1056 kanałów. System z preferowanym układem XCKU040-1FFVA1156 kosztowałby o ponad 100 tys. zł więcej i wykorzystywałby dodatkowe 3 kraty. Należy jednak pamiętać, że układ ten jest wykorzystywany w innych projektach Sinara i w końcowym rozrachunku istniałaby szansa uzyskania jeszcze lepszej ceny samego FPGA. Droższe warianty to koszt większy o kolejno 138,4 tys. zł oraz 146,7 tys. zł. Warto podkreślić jednak, że ostatnia konfiguracja pozwalałaby na użycie aż 4 krat mniej, co znacząco zmniejszyłoby objętość całego systemu. Należy pamiętać, że przeprowadzone obliczenia były szacunkowe – zastosowanie większego układu FPGA mogłoby wiązać się z koniecznością wykorzystania większej ilości warstw obwodu drukowanego, co zwiększyłoby koszty produkcji modułu Shuttler.

Powyższe wyniki wykazały, że niezależnie od zastosowanego układu FPGA koszt jednego kanału to ponad 2,1 tys. zł (około 555 \$). Dotychczas systemy sterowania pułapkami jonowymi stosowane przez fizyków współpracujących przy projekcie Sinara oraz zewnętrznych instytutów zainteresowanych modułem Shuttler generowały koszt na poziomie 300\$ (około 1 166 zł) za pojedynczy kanał. Na tym etapie projekt Shuttler zapewniał większą precyzję i łatwość skalowania, przez co gotowi byli zaakceptować koszt nieznacznie przekraczający tę kwotę, lecz prawie dwukrotny wzrost stanowił barierę skutecznie zniechęcającą do inwestycji.

Wymagania systemu nie pozwalały jednak na podjęcie kroków optymalizujących cenę poprzez modyfikację stosowanej architektury. Wykorzystanie środowiska MicroTCA było konieczne ze względu na kompatybilność z resztą podsystemów projektu. Jego zmiana zaprzeczyłaby idei standaryzacji, która jest jednym z głównych celów przyświecających powstawaniu kolejnych modułów w projekcie Sinara. Z kolei stosowanie wyłącznie modułu AMC zostało wymuszone przez wymagane

parametry i właściwości (jak np. deterministyczna latencja). Jedynym standardowym działaniem mogłoby być wykorzystanie dodatkowych układów nadajników LVDS, sterowanych liniami asymetrycznymi. W teorii umożliwiłoby to sterowanie do 3 przetworników z jednego banku wyprowadzeń. Dokładna analiza wykazała jednak, że byłoby to rozwiązanie dające niewielkie korzyści. Podstawowym problemem był fakt, że w przypadku mnożenia dostępnej liczby przetworników na pojedynczej płycie, kryterium ograniczającym stałaby się dostępna przestrzeń na obwodzie drukowanym. Szacowana maksymalna liczba możliwych do umieszczenia przetworników LTC2000-16 to około 20 sztuk. Przegląd nadajników LVDS wykazał, że dostępny układ o największej gęstości wyprowadzeń to SN75LVDS387 z 16 nadajnikami mogącymi obsłużyć linie danych dla pojedynczego przetwornika. Ilość miejsca zajmowana przez ten układ była jednak porównywalna do tej, wykorzystywanej przez sam przetwornik. W związku z tym maksymalna potencjalna liczba kanałów została by około dwukrotnie zmniejszona z 20 do 10 sztuk. Nie było możliwości umieszczenia tych układów po drugiej stronie obwodu drukowanego, ponieważ było to miejsce zarezerwowane na tory analogowe. Oznacza to, że dla rozpatrywanego wariantu z największym układem FPGA metoda ta zmniejszyłaby wręcz dostępną ilość kanałów. W przypadku wariantu z najmniejszym FPGA (a więc teoretycznie dającym największe pole do optymalizacji) liczba dodatkowych przetworników oscylowałaby w okolicach 4 sztuk.

Dodatkowym problemem była kwestia poziomu sygnałów logicznych. Układ SN75LVDS387 na wejściu wykorzystuje standard Low-Voltage Transistor–Transistor Logic (LVTTTL) – nieobsługiwany przez część banków wyprowadzeń układów UltraScale. Konieczne byłoby zatem stosowanie dodatkowych układów translacji poziomów logicznych lub innych układów nadajników LVDS (o mniejszej liczbie kanałów). Każda z tych opcji wiązała się z dalszym pogłębianiem problemu dostępnej powierzchni, a w przypadku translatorów poziomów także z kwestią zachowania odpowiednich zależności czasowych i niskiej latencji. Konieczne zatem było podjęcie niestandardowych kroków i poszukiwanie nieszablonowych rozwiązań w celu znaczącego obniżenia ceny całego systemu.

4. Konceptcje metod optymalizacji

W przedstawionej sytuacji niezmiennymi elementami modułu, stanowiącymi barierę w osiągniętej liczbie kanałów na pojedynczej płycie, były:

- układ FPGA z rodziny Kintex UltraScale,
- przetwornik cyfrowo-analogowy LTC2000-16.

Problemem był fakt, że przetwornik posiada interfejs równoległy z liniami LVDS, a więc, jak zostało już wspomniane, wymaga podłączenia aż 34 linii. Z kolei układy FPGA Kintex UltraScale standardowo posiadają 52 wyprowadzenia (dla uściślenia, występują modele posiadające pojedyncze banki o zmniejszonej ilości wyprowadzeń), z czego tylko 48 może służyć jako linie różnicowe. Byłoby to nieoptymalne rozwiązanie w kontekście wykorzystania dostępnych wyprowadzeń. Dlatego podjęto próbę opracowania niestandardowej metody połączeń pomiędzy FPGA a przetwornikiem. Gdyby umożliwić podłączenie większej ilości kanałów do banku, koszty całego systemu zostałyby znacząco zmniejszone dzięki ograniczeniu liczby wykorzystywanych krat MicroTCA oraz modułów MCH Metlino. W pierwszej kolejności dokonano zatem analizy wspomnianych układów.

4.1. Analiza układu odbiornika

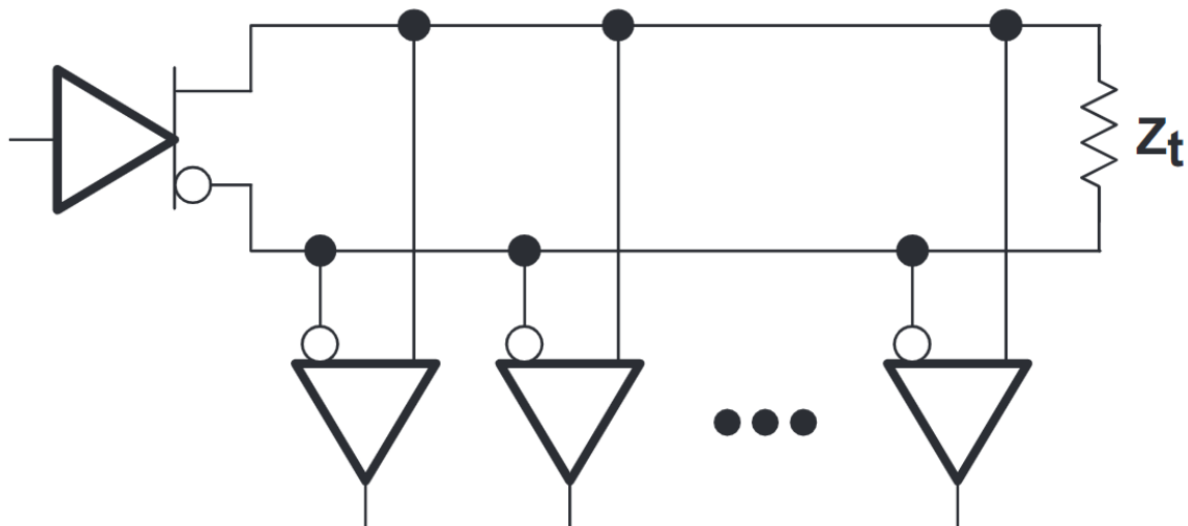
Na rys. 4.1 przedstawione zostały parametry sygnału wejściowego wymaganego przez układ odbiornika LTC2000. Niezależnie od badanej metody poziomy logiczne musiały zostać zachowane, aby w poprawny sposób przekazać cyfrowe dane do przetwornika.

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
LVDS Inputs (DCKIP, DCKIN, DAP/N, DBP/N)						
	Differential Input Voltage Range	●	±0.2		±0.6	V
	Common-Mode Voltage Range	●	0.4		1.8	V
	Differential Input Impedance	●	95	120	145	Ω
	Maximum Data Rate	●			1250	Mbps
	LVDS Clock Frequency	●	25		625	MHz

Rysunek 4.1: Parametry odbiorników LVDS przetwornika LTC2000 [11].

Dostarczony sygnał musiał generować napięcie różnicowe pomiędzy wyprowadzeniami wejściowymi przetwornika na poziomie minimum $\pm 0,2$ V. Dopuszczalne napięcie wspólne dla linii różnicowej było elastyczne – zakres 0,4 V do 1,8 V oznacza praktycznie cały zakres napięcia zasilania wynoszącego właśnie 1,8 V.

Problematyczną kwestią była różnicowa impedancja wejściowa. Wartość 120 Ω oznaczała, że zaimplementowana została typowa wewnętrzna terminacja dla standardu LVDS. Znacząco ograniczyło to możliwości alternatywnego połączenia pomiędzy układami, np. uniemożliwiło to wykorzystanie topologii analogicznej do tej stosowanej w przypadku standardu Multipoint Low-Voltage Differential Signaling (MLVDS) – pozwalającej na proste równoległe połączenie kilku odbiorników do wspólnej magistrali (przedstawione na rys. 4.2). Wymaga ona bowiem zastosowania pojedynczej zewnętrznej terminacji oraz odbiorników o jak największej impedancji wejściowej (w celu niezaburzenia impedancji linii różnicowej). Próba równoległego połączenia układów wiązałaby się także z koniecznością stosowania dodatkowych mechanizmów synchronizujących.



Rysunek 4.2: Konfiguracja MLVDS z jednym nadajnikiem i wieloma odbiornikami.

4.2. Analiza układu nadajnika

Zgodnie z wymaganiami dane dla przetworników cyfrowo-analogowych sterujących pułapką jonową miały być generowane przez układ z rodziny Kintex UltraScale. FPGA tego typu dysponują wyprowadzeniami wejścia/wyjścia podzielonymi na dwa typy banków: high-range (HR) oraz high-performance (HP). Istotną informacją jest fakt, że pojedynczy bank zawiera 52 wyprowadzenia, w tym 48 wyprowadzeń może być skonfigurowanych w maksymalnie 24 pary różnicowe. Jak zostało już wspomniane, zalecane jest, aby pojedynczy przetwornik byłysterowany przez wyprowadzenia pochodzące z tego samego banku, dla zachowania odpowiednich zależności czasowych. Typy banków różnią się zaś przede wszystkim wspieranymi wartościami zasilania dla swoich buforów, a co za tym idzie, możliwymi do zaimplementowania standardami poziomów logicznych. Ponieważ, niezależnie od doboru modelu i jego obudowy, do obsługi przetworników zostałyby wykorzystane zarówno banki HR i HP, konieczne było opracowanie koncepcji połączeń możliwej do implementacji dla każdego z tych typów, aby zminimalizować różnice w przebiegach czasowych i opóźnieniach generowanych sygnałów. Z tego względu, zgodnie z tabelą udostępnioną przez firmę Xilinx i przedstawioną na rys. 4.3, do rozpatrzenia zostały wyłącznie standardy wspierane przy zasilaniu o wartości 1,8 V, 1,5 V, 1,35 V oraz 1,2 V. Standardy wspólne dla obydwu typów banków to:

Feature	HP I/O Banks	HR I/O Banks
3.3V I/O standards ⁽¹⁾	N/A	Supported
2.5V I/O standards ⁽¹⁾	N/A	Supported
1.8V I/O standards ⁽¹⁾	Supported	Supported
1.5V I/O standards ⁽¹⁾	Supported	Supported
1.35V I/O standards ⁽¹⁾	Supported	Supported
1.2V I/O standards ⁽¹⁾	Supported	Supported
1.0V POD I/O standard	Supported	N/A

Rysunek 4.3: Wspierane wartości zasilania dla poszczególnych typów banków wyprowadzeń dla układów FPGA UltraScale [42].

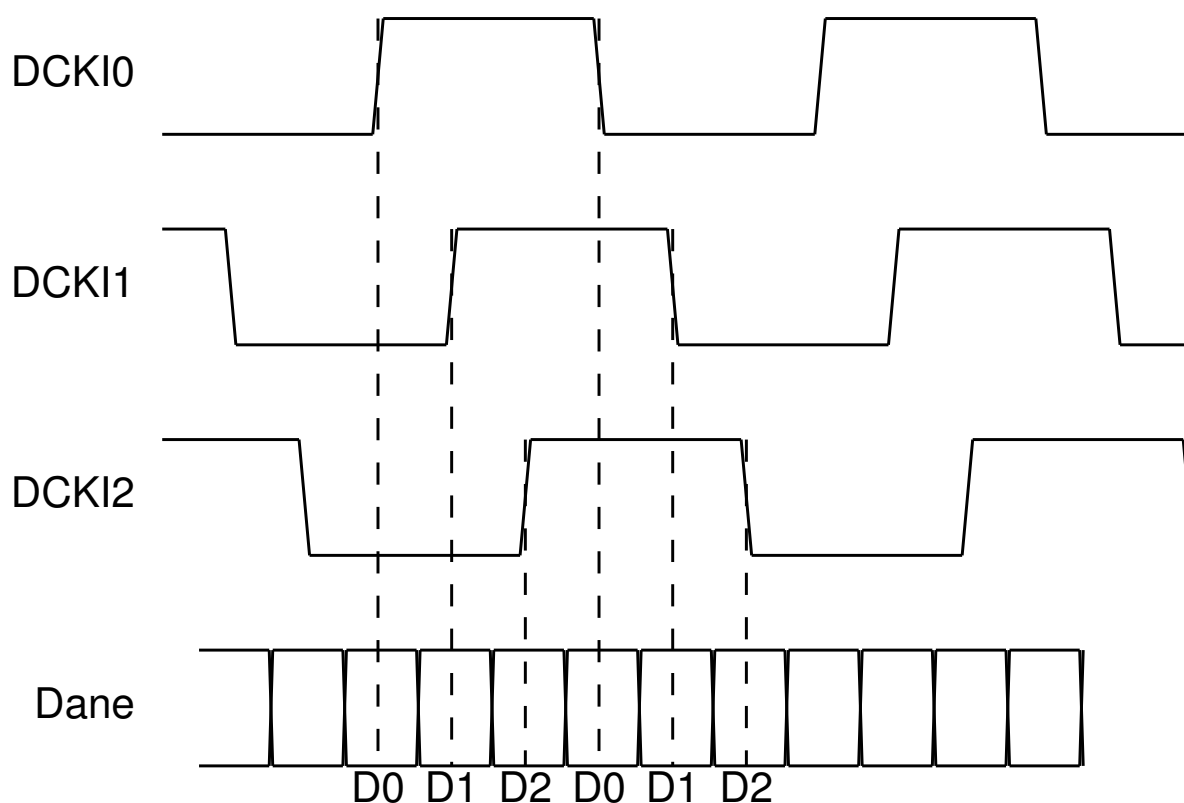
- LVCMOS (18; 15; 135; 12),
- HTSL (I; I_18 oraz ich różnicowe odpowiedniki),
- SSTL (18_I; 15; 135; 12 oraz ich różnicowe odpowiedniki),
- HSUL_12 i różnicowy HSUL_12.

Standardy te różnią się gwarantowanymi poziomami logicznymi, koniecznością obecności terminacji oraz wykorzystaniem napięcia odniesienia. Elastyczność układu odbiornika – przede wszystkim szeroki zakres napięcia wspólnego – sprawił, że praktycznie każdy ze standardów potencjalnie mógł zostać wykorzystany przy opracowywaniu koncepcji nietypowego interfejsu pomiędzy układami FPGA i przetwornikami C/A LTC2000. Należy jednak podkreślić, że celem niniejszej pracy nie było znalezienie optymalnego standardu dla zaproponowanych w kolejnych podrozdziałach metod. Konieczne było wyłącznie zweryfikowanie, czy dana konfiguracja zapewni niezawodną transmisję z zadaną szybkością w dowolnym ze standardów.

4.3. Koncepcja połączenia równoległego

Pierwsza koncepcja zakładała próbę implementacji połączenia równoległego. Linie różnicowe, przesyłające dany bit danych, rozdzielałyby się na dwie lub trzy dodatkowe pary różnicowe. Należy od razu podkreślić konsekwencje takiego rozwiązania – częstotliwość sygnału danych dla danej częstotliwości próbkowania musiałaby wzrosnąć wprost proporcjonalnie do stopnia zrównoleglenia połączeń. Ograniczyłoby to maksymalną częstotliwość zmiany sygnału analogowego. Każdy przetwornik, w obrębie pojedynczego banku wyprowadzeń, musiałby posiadać dedykowane połączenie pary różnicowej z sygnałem zegarowym taktującym odbiór danych, zaś po stronie układu FPGA konieczne byłoby zapewnienie odpowiedniego przesunięcia fazowego pomiędzy takimi sygnałami. Sytuacja ta dla podłączenia trzech przetworników została zobrazowana na rys. 4.4. W takim przypadku doszłoby do dodatkowego problemu synchronizacji. Częstotliwość sygnału zegarowego, taktującego generację próbek, ze względu na wewnętrzne obwody synchronizujące musi być dwukrotnością częstotliwości sygnału zegarowego danych. Spowodowałoby to, że część próbek byłaby generowana z opóźnieniem jednego taktu. Konieczne byłoby zatem opóźnienie pozostałej części próbek wewnątrz układu FPGA.

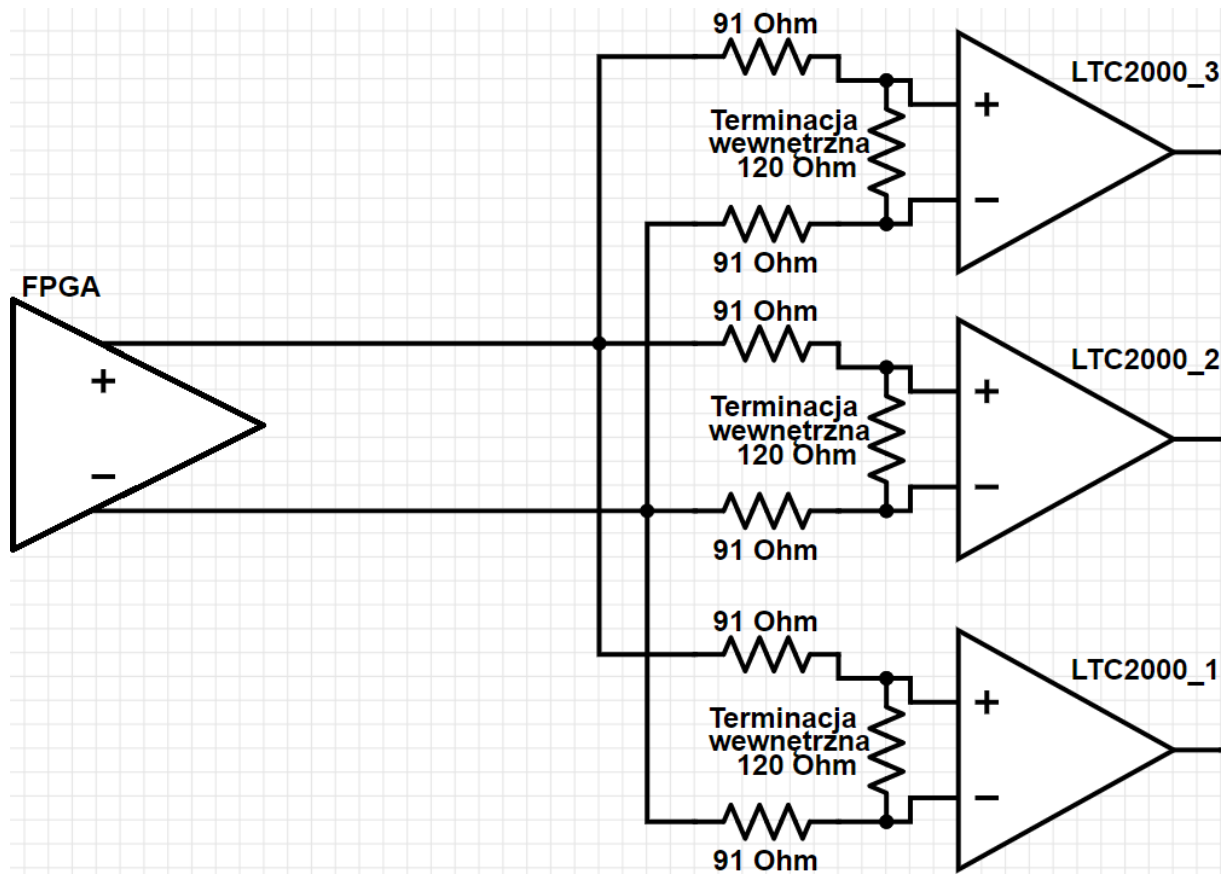
Wyzwaniem było także uwzględnienie dopasowania poszczególnych linii ze względu na wspomnianą wcześniej wewnętrzną terminację układu LTC2000. Konieczne było wykorzystanie topologii gwiazdy. Występuje ona zazwyczaj w dwóch wariantach, różniących się miejscem rozdzielenia linii – na początku lub na końcu linii transmisyjnej. Zaproponowano zatem dwie koncepcje – pierwsza polega na rozdzieleniu sygnału przy nadajniku i poprowadzeniu go dwiema lub trzema parami linii różnicowych o impedancji charakterystycznej 120Ω (najlepiej dopasowaną do odbiornika) do przetworników. Metoda ta jest absolutnie najprostszą z możliwych – nie wymaga użycia ani jednego dodatkowego elementu. Jedyną nieciągłość impedancji znajduje się bezpośrednio przy nadajniku, zatem nie ma dużego wpływu na integralność sygnałową. Jej wadą jest zwiększenie pojemności linii ze względu na prowadzenie trzech oddzielnych ścieżek. Jedyną nietypową kwestią wykorzystania tej metody pozostaje potrzeba ciągłego nadawania danych do wszystkich odbiorników równocześnie oraz zachowania pełnej synchronizacji przetworników (na poziomie próbkowania, a nie odbioru danych, która w takim wypadku jest niemożliwa). Kolejną wadą (poza wspomnianą koniecznością zwiększenia częstotliwości nadawanego sygnału) jest konieczność wyprowadzenia większej ilości



Rysunek 4.4: Diagramy czasowe sygnałów zegarowych dla trzech przetworników ze wspólną linią danych. Oznaczenia D0, D1, D2 opisują dane, które powinny zostać odebrane przez kolejne przetworniki o numerze 0, 1 oraz 2.

linii. W przypadku podwójnego zrównoleglenia konieczne byłoby wyprowadzenie 68 linii z banku zawierającego 52 wyprowadzenia (pod warunkiem niewykorzystania pozostałych wyprowadzeń). W modułach zawierających duże układy w obudowie Ball Grid Array (BGA) (jak Kintex UltraScale) liczba warstw obwodu drukowanego związana jest z ilością linii wyprowadzanych spod tych układów. Zatem, niezależnie od wybranego modelu FPGA, konieczne byłoby zastosowanie większej ilości warstw, co zwiększyłoby koszty produkcji. Dla potrójnego zrównoleglenia liczba linii rośnie do 102, co praktycznie podwoiłoby tę ilość.

Druga koncepcja zakładała rozdzielenie wspólnej linii przy odbiornikach. Przedstawiona została na rys. 4.5. W tym wariantcie przetworniki połączone równolegle należałoby umieścić blisko siebie, ponieważ każdy fragment ścieżki po rozdzieleniu linii stanowiłby coraz większe zaburzenie ciągłości impedancji. Aby zapewnić odpowiednie dopasowanie na końcu takiej linii, konieczne byłoby dodanie rezystorów szeregowo do wyprowadzeń odbiornika. Każdy odbiornik musiałby bowiem wykazać rezystancję wejściową 300Ω , aby równoległe połączenie trzech identycznych układów dało dopasowanie do linii o impedancji charakterystycznej 100Ω . Rozwiązanie to uprościłoby wyprowadzenie linii danych spod układu FPGA, jednak posiadało dosyć znaczącą wadę. Obudowa przetwornika LTC2000 ma wymiary 15 mm na 9 mm, zatem rozdzielenie 16 linii danych oraz ich doprowadzenie do tych stosunkowo dużych układów w praktyce oznaczałoby zastosowanie linii doprowadzeń rzędu długości 2 cm. Miałyby one zatem wpływ na kwestie integralności sygnałowej, co wymagało weryfikacji w symulacji. W przypadku zrównoleglenia dwóch układów fizycznie możliwe byłoby uzyskanie długości doprowadzeń rzędu 1 cm.



Rysunek 4.5: Schemat układu sterowania trzech odbiorników LVDS przetwornika LTC2000 metodą rozdzielania ścieżek przy odbiornikach.

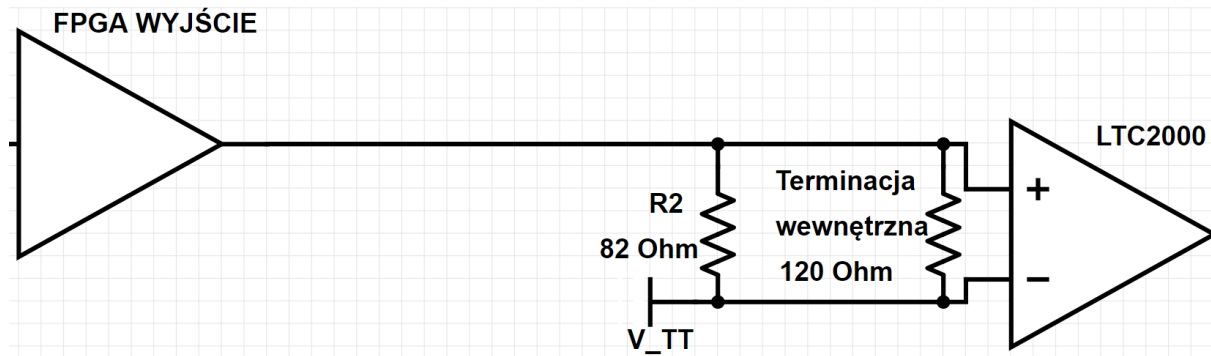
4.4. Koncepcja sterowania odbiornika LVDS linią asymetryczną

Inna koncepcja opierała się na pomysłu redukcji wymaganych połączeń pomiędzy układem FPGA a pojedynczym przetwornikiem cyfrowo-analogowym, aby umożliwić obsłużenie większej ilości połączeń typu punkt-punkt zamiast ich zrównoleglania. Najprostszą metodą byłoby przesyłanie sygnału wyłącznie przez linię asymetryczną. Dla przypomnienia, w standardowym połączeniu równoległym LVDS wymagane były 34 linie na pojedynczy układ LTC2000-16. Gdyby dane przesyłać asymetrycznie, liczba ta zmniejszyłaby się do zaledwie 18 lub do 17, gdyby tak samo przesyłać sygnał zegarowy. Szczególnie drugi wariant byłby bardzo korzystny, ponieważ pojedynczy bank mógłby pomieścić połączenia dla trzech przetworników (51 wykorzystanych wyprowadzeń na 52 dostępne).

Należało zatem opracować metodę podłączenia sygnału przesyłanego linią asymetryczną tak, aby umożliwić prawidłowy odczyt odbiornika opisanego w podrozdziale 4.1. Pierwszym krokiem było dodanie napięcia odniesienia V_{TT} i podłączenie go do wejścia N odbiornika („P” z ang. *positive* – oznacza sygnał, którego wartość logiczna interpretowana asymetrycznie odpowiada wartości logicznej całej pary różnicowej oraz „N” z ang. *negative* – oznacza linie z sygnałem logicznym stanowiącym negację wartości na linii P). Jego wartość musiała być dostosowana do standardu poziomów logicznych, który byłby zastosowany na asymetrycznej linii danych i znajdować się w połowie zakresu przyjmowanych poziomów napięć. W zależności od stanu logicznego sygnał asymetryczny posiadałby większe bądź mniejsze napięcie względem wartości napięcia odniesienia – a więc musiałoby być ono w stanie zarówno dostarczyć, jak i odebrać prąd przepływający przez rezystancję

wewnętrzna odbiornika. Z tego powodu należało wykorzystać regulator napięcia terminującego, który posiada wszystkie potrzebne właściwości.

Konieczne było także zapewnienie odpowiedniej terminacji. W przypadku linii asymetrycznej standardowo stosowaną wartością jest $50\ \Omega$. Należało jednak pamiętać o wewnętrznej terminacji LVDS odbiornika. Dodano zatem rezystor o wartości $82\ \Omega$ pomiędzy linie wejściowe odbiornika, dzięki czemu poprzez równoległe połączenie z rezystancją wewnętrzną zapewniona została prawidłowa terminacja. Schemat ideowy całego połączenia przedstawiony został na rysunku 4.6.



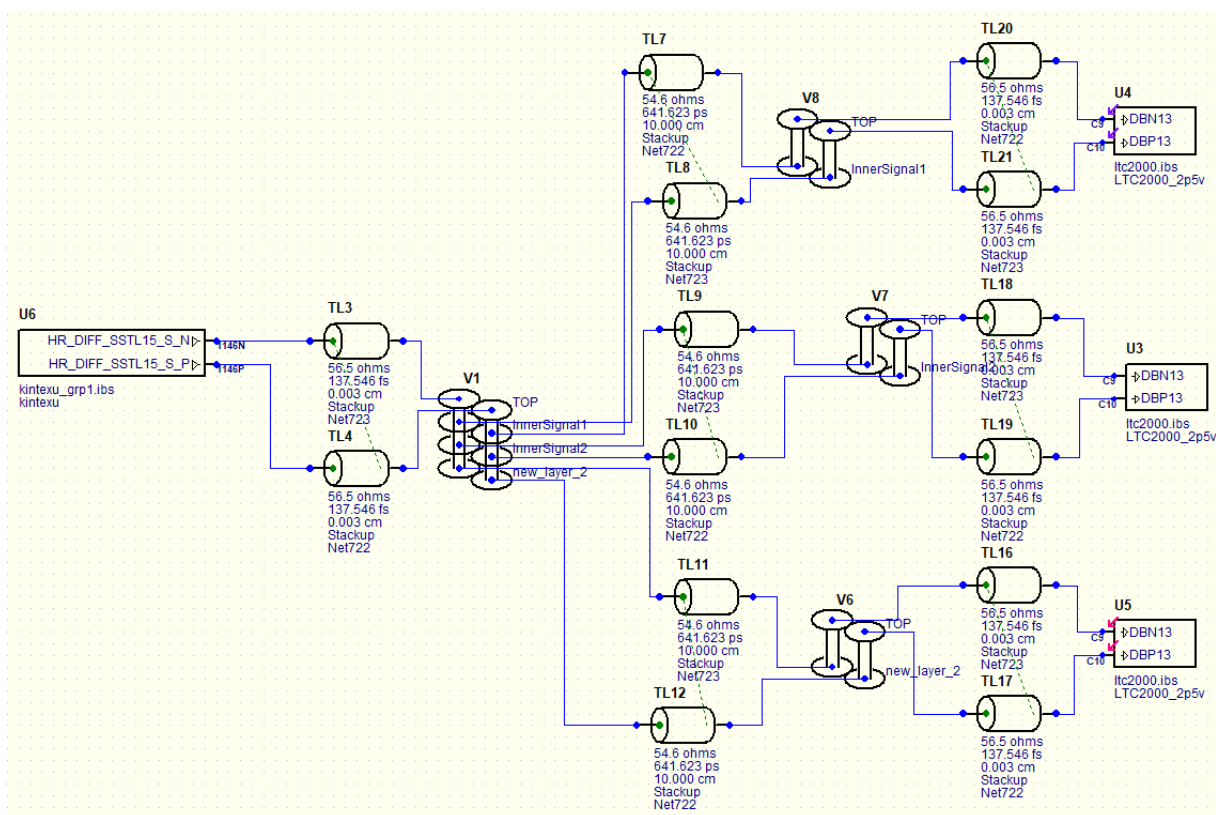
Rysunek 4.6: Schemat układu sterowania odbiornika LVDS przetwornika LTC2000 linią asymetryczną.

5. Symulacje badanych metod

Wszystkie symulacje zostały wykonane w oprogramowaniu HyperLynx z wykorzystaniem dodatkowych plików Input/output Buffer Information Specification (IBIS) dostarczonych przez producentów – firmę Xilinx oraz Analog Devices – odpowiednio dla układu FPGA oraz przetwornika cyfrowo-analogowego LTC2000. Pomimo faktu, że pierwsze symulacje miały na celu wyłącznie sprawdzenie, czy zaproponowane metody mają jakąkolwiek szansę na fizyczną realizację, to skonfigurowane zostały one tak, aby możliwie jak najdokładniej odzwierciedlić warunki związane z połączeniem tych układów na docelowej płycie. W związku z tym każdy schemat uwzględnia przelotki znajdujące się zaraz obok wyprowadzeń poszczególnych układów oraz linie długie dla sygnałów o długości 10 cm (jest to wartość orientacyjna, dobrze oddająca rząd wielkości możliwych docelowych połączeń na płycie w formacie AMC). Wprowadzono także dane dotyczące konfiguracji warstw płyty, pobrane z projektu Sayma AMC – ponieważ z dużym prawdopodobieństwem ta sama konfiguracja będzie wykorzystywana w płycie Shuttler.

5.1. Połączenie równoległe

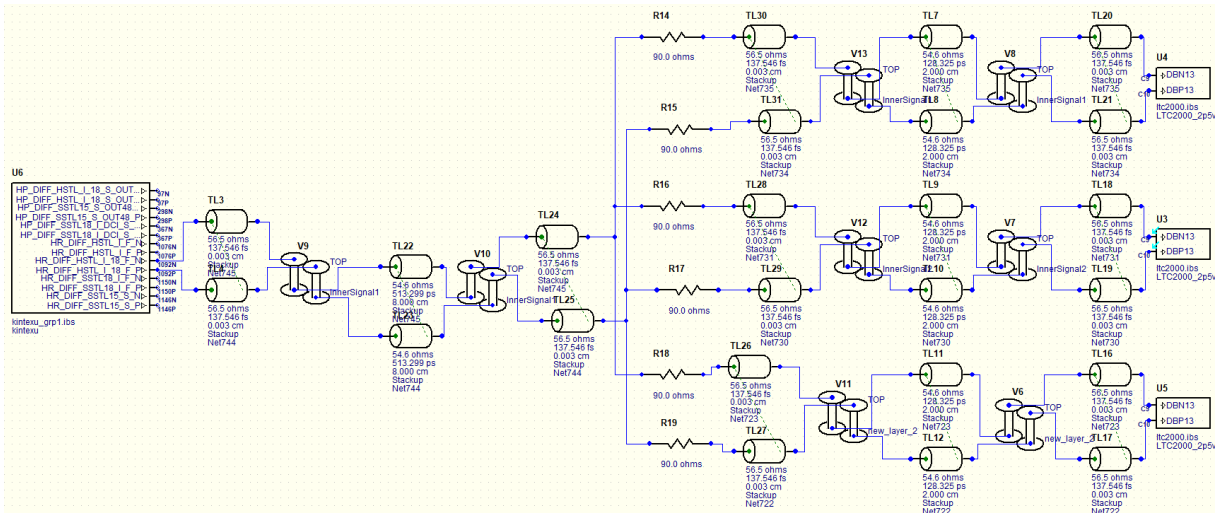
Schematy symulowanych układów (zrównoleglenia trzech przetworników) zostały przedstawione kolejno na rys. 5.1 dla koncepcji rozdzielania linii przy nadajniku oraz na rys. 5.2 dla koncepcji rozdzielania linii przy odbiornikach. W pierwszym przypadku sygnał rozdzielony został w przelotce i dla



Rysunek 5.1: Schemat symulacji sterowania trzech odbiorników LVDS przetwornika LTC2000 metodą rozdzielania ścieżek przy nadajniku.

każdego z układów poprowadzona została para różnicowa na innej warstwie obwodu drukowanego.

Linia długa o długości 10 cm kończyła się przelotką do warstwy górnej, gdzie sygnał różnicowy był bezpośrednio podłączony do wyprowadzeń LTC2000. Widoczne na schemacie dodatkowe linie długie związane były z koniecznością podłączenia sygnału z przelotki właśnie do modelu takiej linii, skonfigurowanej na minimalną długość. Drugi przypadek był dużo bardziej skomplikowany. Sygnał z



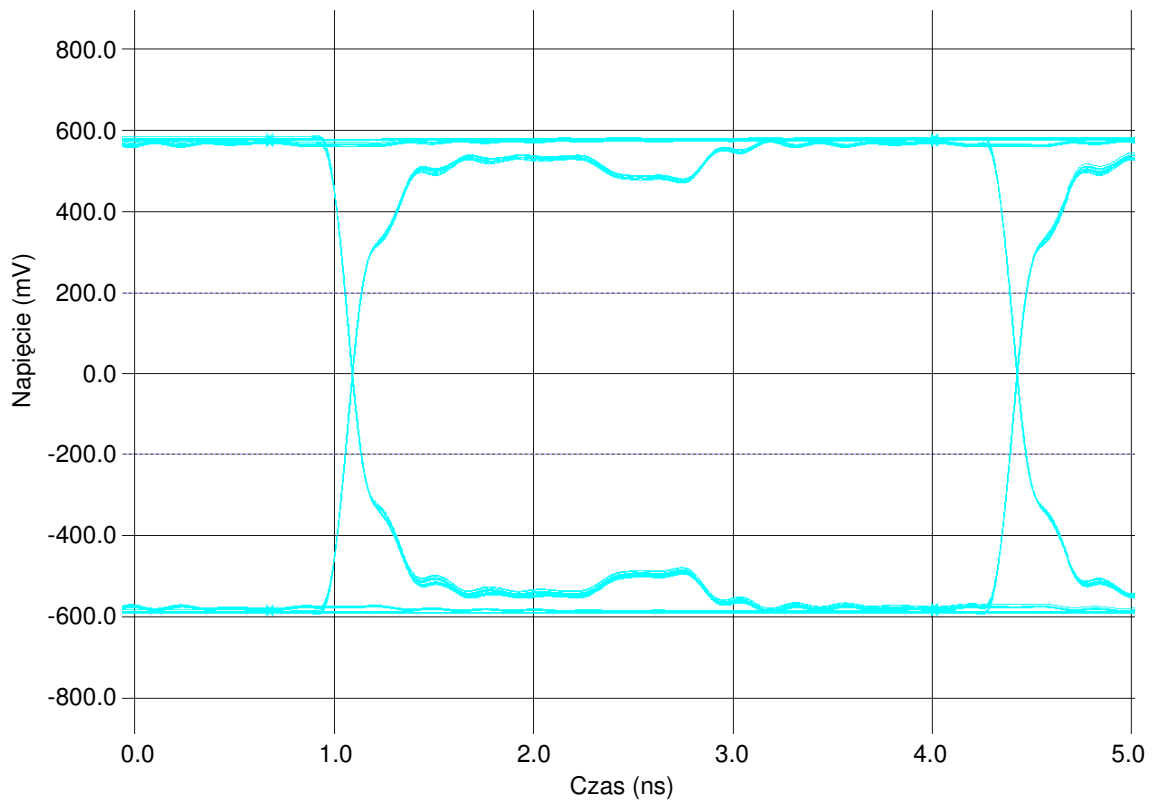
Rysunek 5.2: Schemat symulacji sterowania trzech odbiorników LVDS przetwornika LTC2000 metodą rozdzielania ścieżek przy odbiornikach.

układu FPGA prowadzony był przez przelotkę, a następnie 8 cm linią długą po warstwie wewnętrznej w okolice przetworników. Tam został wyprowadzony na wierzchnią warstwę, gdzie przy wykorzystaniu rezystorów doszło do rozdzielania sygnału. Ponieważ przetwornik LTC2000 posiada obudowę typu BGA konieczne było powrócenie z rozdzielonymi sygnałami na warstwę wewnętrzną. Niemożliwe było bowiem doprowadzenie sygnału po warstwie wierzchniej do wszystkich wyprowadzeń.

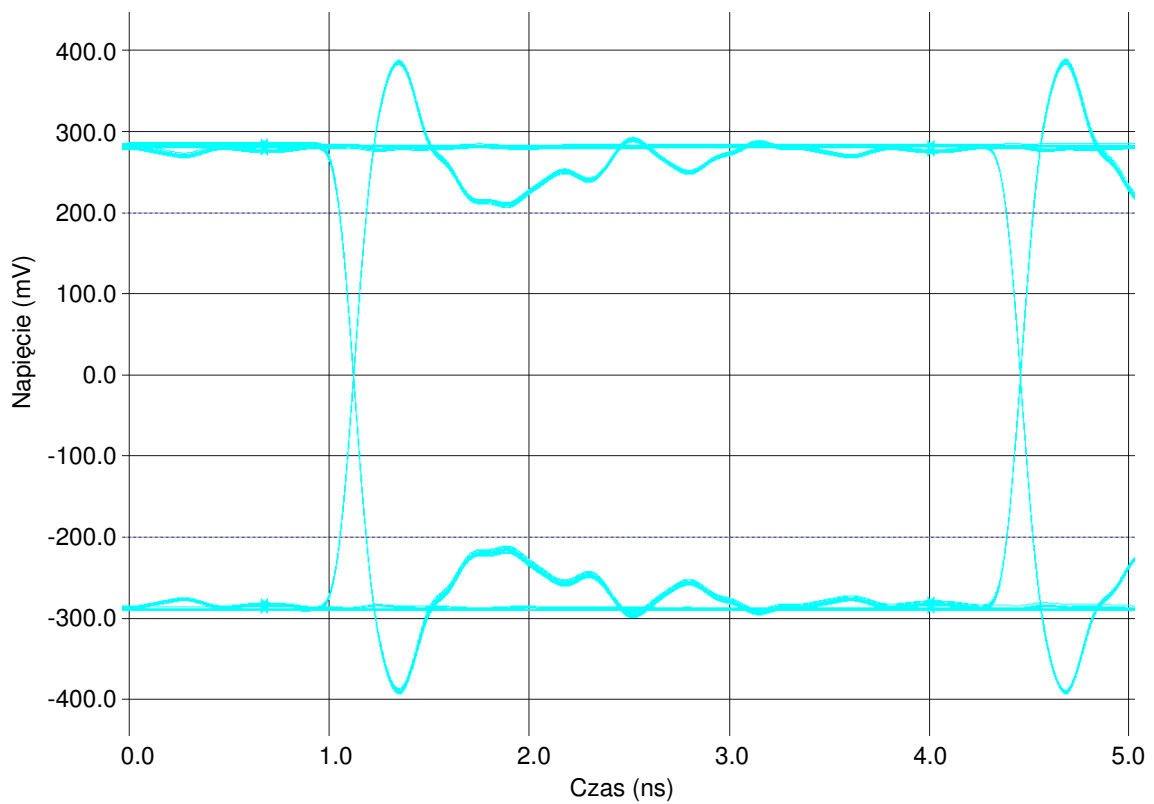
Symulacje wykonano dla transferu 300 Mb/s (100 Mb/s dla każdego z przetworników). Diagramy oka sygnału różnicowego odbieranego przez przetworniki cyfrowo-analogowe przedstawione zostały na rys. 5.3 i 5.4. Osiągnięte wyniki symulacji rozdzielania linii przy nadajniku w standardzie HR DIFF SSTL15 S pokazują, że sygnał zachowuje wszystkie parametry konieczne do jego poprawnego zinterpretowania przez układ odbiornika. Poziomy logiczne są bliskie wartości +/- 600 mV (przy +/- 200 mV wymaganym). Widoczny jest za to efekt nieznacznego niedopasowania impedancji w postaci początkowego zmniejszenia amplitudy.

Nieco inaczej wygląda sytuacja w przypadku symulacji z rozdzielaniem linii przy odbiornikach. Wprowadzone rezystory, mające na celu dopasowanie impedancji, tworzą wraz z terminacją wewnętrzną dzielnik napięciowy obniżający amplitudę sygnału różnicowego. Dla wyprowadzeń układu FPGA HR DIFF HSTL I 18 osiągnięto największą amplitudę, która wraz z widocznymi zniekształceniami nieznacznie przekracza (o około 10 mV) wymagane napięcie dla poprawnego odczytu poziomu logicznego. Była to wartość na tyle niska, że nie możliwe było przyjęcie założenia, że odbiorniki bezbłędnie będą mogły odczytywać wartości logiczne. Należało pamiętać o tym, że w docelowym układzie pojawią się dodatkowe zakłócenia w postaci np. przesłuchów z sąsiednich linii.

Dokonano zatem dodatkowej symulacji równoległego połączenia z rozdzielaniem linii przy odbiornikach dla dwóch układów LTC2000. Większość ustawień pozostała bez zmian. Konieczna była korekta wartości transferu sygnału na 200 Mb/s (w celu osiągnięcia 100 Mb/s dla każdego przetwornika) oraz wartości rezystorów szeregowych, zmienionych na 40 Ω. W tym wypadku bo-

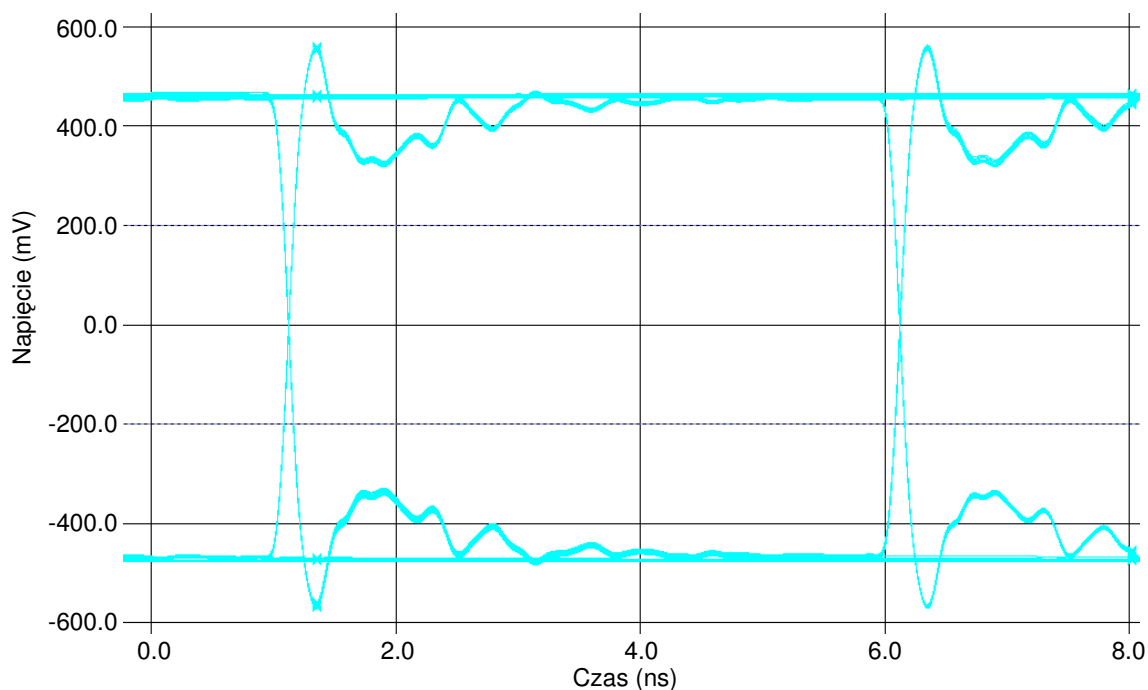


Rysunek 5.3: Diagram oka sygnału różnicowego wejściowego odbiornika LTC2000 dla transferu danych 300 Mb/s (100 Mb/s dla każdego z odbiorników) w konfiguracji zrównoleglenia trzech połączeń przy nadajniku.



Rysunek 5.4: Diagram oka sygnału różnicowego wejściowego odbiornika LTC2000 dla transferu danych 300 Mb/s (100 Mb/s dla każdego z odbiorników) w konfiguracji zrównoleglenia trzech połączeń przy odbiornikach.

wiem każdy przetwornik dla zachowania dopasowania musiał stanowić obciążenie 200Ω . Wyniki symulacji przedstawiono na rys. 5.5. Dowodzą one, że metoda ta mogłaby znaleźć zastosowanie wyłącznie przy próbie podwojenia ilości stosowanych przetworników. Znacząco wzrosła amplituda



Rysunek 5.5: Diagram oka sygnału różnicowego wejściowego odbiornika LTC2000 dla transferu danych 200 Mb/s (100 Mb/s dla każdego z odbiorników) w konfiguracji zrównoleglenia dwóch połączeń przy odbiornikach.

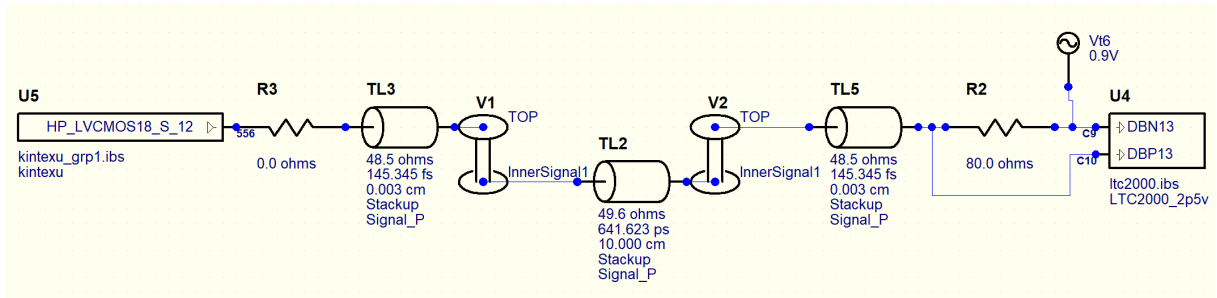
sygnału różnicowego. Uwzględniając zniekształcenia sygnału, amplituda sygnału pozostawała o 110 mV większa niż wymagany próg 200 mV i gwarantowała poprawne działanie całego interfejsu.

5.2. Sterowanie LVDS linią asymetryczną

Schemat symulowanego układu przedstawiony został na rys. 5.6. Składa się on z pojedynczego nadajnika i odbiornika, gdzie pomiędzy wejścia odbiornika podłączony został rezystor 82Ω zgodnie z zaproponowaną koncepcją. W ramach symulacji napięcia terminującego podłączono idealne źródło napięciowe o wartości 0,9 V. Zśród możliwych modeli nadajnika wybrano HP LVC MOS18 S 12 – nadajnik typu LVC MOS w banku wyprowadzeń typu HP zasilany z napięcia 1,8 V w konfiguracji prądu sterującego 12 mA (maksymalna wartość dla banków typu HP) i najmniej stromym zboczem przy zmianie stanu logicznego – co ogranicza problemy z integralnością sygnałową. Wybór LVC MOS18 podyktowany był doбором standardu występującego zarówno w bankach typu HR, jak i HP o największej deklarowanej wydajności prądowej i amplitudzie.

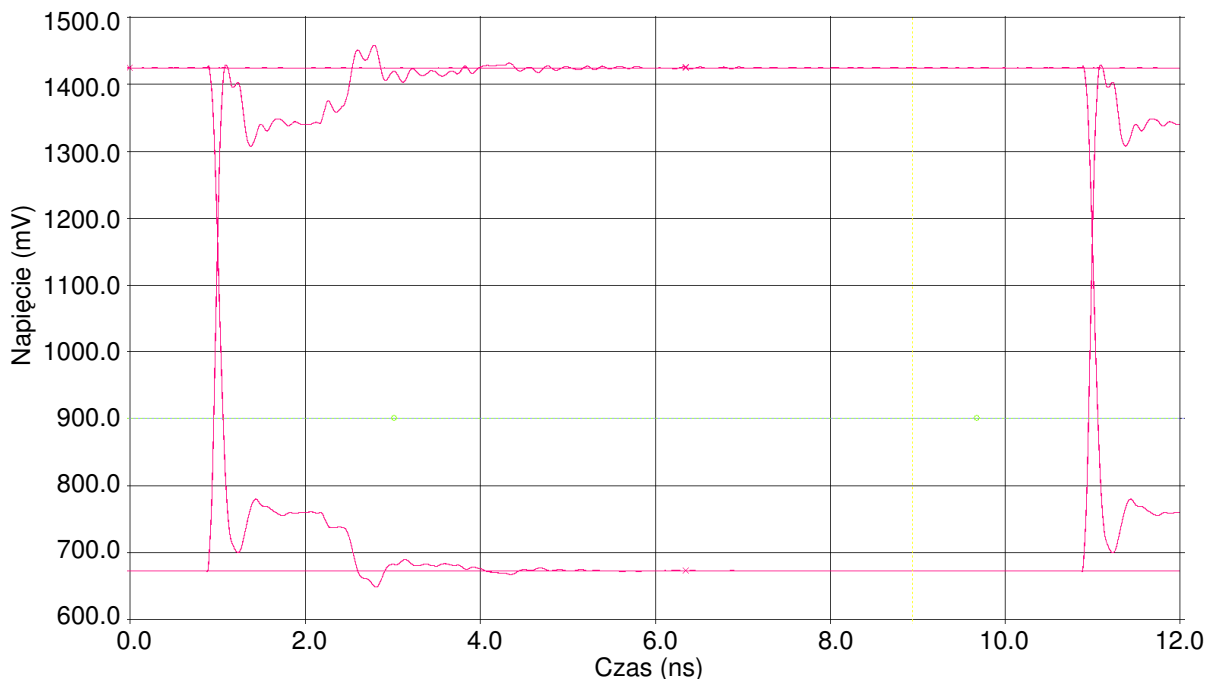
Symulacji diagramu oka dokonano dla częstotliwości sygnału 100 MHz. Wyniki ukazujące napięcie na sterowanym wejściu odbiornika LVDS przedstawiono na rys. 5.7.

Wysokość przedstawionego diagramu oka wynosi około 751 mV, przy oczekiwanym około 1,15 V. Na co jednak szczególnie należy zwrócić uwagę, to wartość średnia napięcia, wokół którego oscyluje sygnał. Wynosi ona około 1,05 V, nie zaś oczekiwane 0,9 V. Zgodnie z dokumentacją techniczną dotyczącą poziomów logicznych układów z rodziny Kintex UltraScale [40] górny poziom logiczny powinien wynosić co najmniej 1,35 V, zaś poziom logicznego zera powinien wynosić maksymalnie

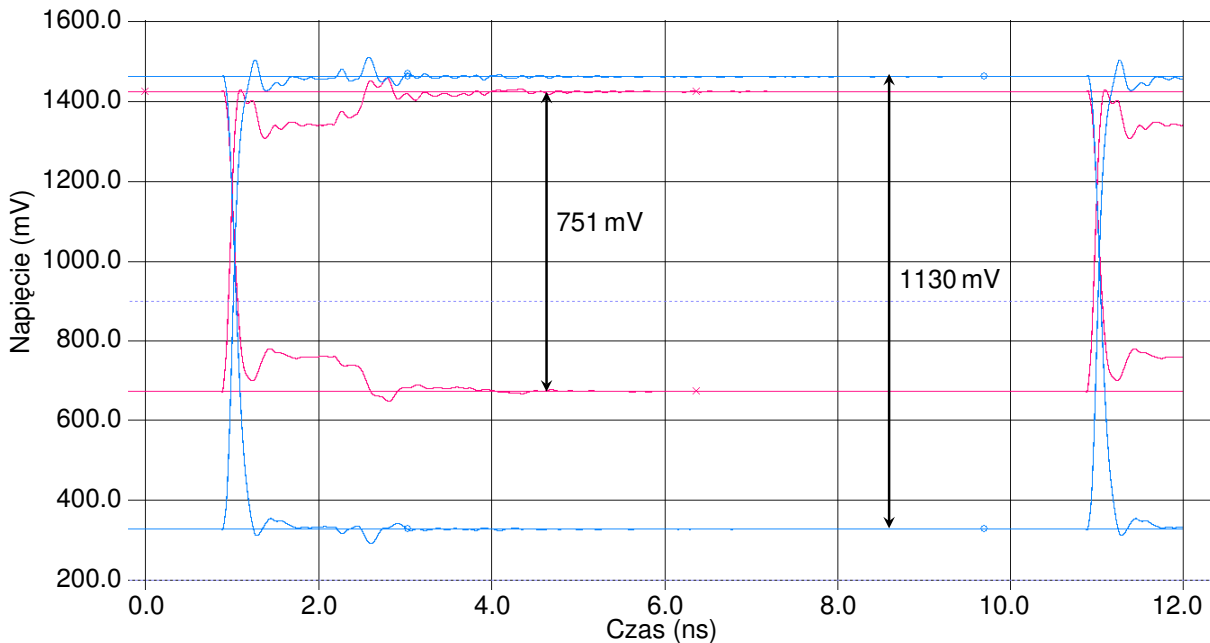


Rysunek 5.6: Schemat symulacji sterowania odbiornika LVDS przetwornika LTC2000 linią asymetryczną.

0,45 V. Z przeprowadzonej analizy wynika, że poziom logiczny zera nie został zachowany i nie ma to związku z niewystarczającą wydajnością prądową wyprowadzenia układu FPGA. W związku z wystąpieniem wartości sprzecznych z teoretycznie spodziewanymi postanowiono wykonać dodatkowe symulacje, mające na celu wyznaczenie elementu badanego układu, który odpowiada za ten stan rzeczy. W pierwszej kolejności zbadano wpływ modelu IBIS przetwornika cyfrowo-analogowego. Usunięto go ze schematów, a w jego miejsce wstawiono rezystancję 120Ω , mającą za zadanie symulować impedancję wejściową odbiornika. Wyniki tak przeprowadzonej symulacji porównano z wcześniej uzyskanymi wynikami i przedstawiono na rys. 5.8. Wysokość diagramu oka wynosi $1,13 \text{ V}$, zaś wartość średnia wynosi około $0,9 \text{ V}$. Sygnał odbierany przez przetwornik posiada zatem amplitudę $\pm 565 \text{ mV}$ i spełnia wymagane parametry.



Rysunek 5.7: Diagram oka symulacji sterowania odbiornika LVDS linią asymetryczną LVCMOS18 z modelem IBIS odbiornika. Transfer danych 100 Mb/s .



Rysunek 5.8: Diagram oka symulacji sterowania odbiornika LVDS linią asymetryczną LVCMOS18 z modelem IBIS odbiornika (czerwony) oraz z modelowaniem odbiornika w postaci pojedynczego rezystora (niebieski). Transfer danych 100 Mb/s.

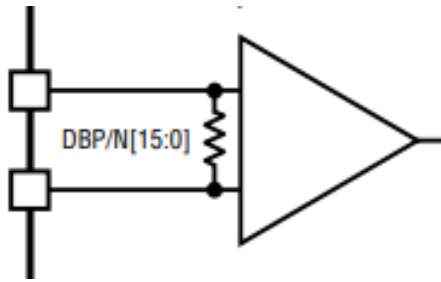
5.3. Wnioski

W wyniku przeprowadzonych symulacji sterowania linią asymetryczną wykazano nietypowe właściwości modelu IBIS odbiornika cyfrowego LTC2000. Wystąpiły dwie różnice w porównaniu z modelowaniem odbiornika poprzez pojedynczy rezystor:

- przesunięcie składowej stałej o około 150 mV,
- zmniejszenie wartości amplitudy sygnału prostokątnego o 379 mV.

Sygnał różnicowy (uwzględniając, że na wyprowadzeniu N napięcie przyjmuje wartość 0,9 V) osiągał skrajne wartości 511 mV oraz -219 mV, co oznaczało, że mogłyby pojawić się problemy przy odczycie logicznej wartości '0'. Oczywiście układ można by zmodyfikować, dopasowując napięcie odniesienia V_{TT} tak, aby jego wartość odpowiadała składowej stałej napięcia na linii sygnałowej, co doprowadziłoby do wyrównania tych wartości na obydwu wyprowadzeniach. Byłoby to jednak ryzykowne i nieprofesjonalne podejście, ponieważ stanowiłoby próbę zniwelowania wymienionych efektów, bez znajomości przyczyny ich wystąpienia. Z dużym prawdopodobieństwem wygenerowałyby to wyłącznie kolejne problemy i błędy.

Teoretycznie symulacje zdyskwalifikowały metodę sterowania odbiorników układu LTC2000 przez linie asymetryczne LVCMOS18. Należało jednak podejść krytycznie do osiągniętych wyników. Na początku opracowywania koncepcji niestandardowego interfejsu przyjęto założenie, że wewnętrzna terminacja LTC2000 składa się z rezystora podłączonego do linii pary różnicowej, co byłoby standardowym rozwiązaniem dla linii LVDS. Sugerowane było to także przez producenta w dokumentacji, co przedstawiono na rys. 5.9. Wyniki symulacji ukazały jednak zmianę składowej stałej napięcia występowanego przez FPGA sygnału, co oznaczało, że obwody wejściowe przetwornika miały wpływ na tę wartość. Efekt taki mógł zostać osiągnięty przez zastosowanie przykładowo terminacji Thevenina. Byłoby to rozwiązanie co najmniej nieoczywiste, ale teoretycznie nie niemożliwe. Znaczne



Rysunek 5.9: Fragment uproszczonego schematu blokowego przetwornika LTC2000 przedstawiający odbiornik danych cyfrowych [11].

zmniejszenie wartości międzyszczytowej sygnału świadczyło jednak o dużo mniejszej impedancji wejściowej układu, niż wynika to z danych podanych przez producenta. Ta sprzeczność pozwoliła na wysunięcie wniosku, że z dużym prawdopodobieństwem dostarczone modele IBIS zawierały błędy, które z kolei spowodowały nieprawidłowe wyniki symulacji, zaś modelowanie z wykorzystaniem pojedynczego rezystora mogło pozwolić na otrzymanie prawidłowych wyników. Profilaktycznie dokonano także weryfikacji, czy dany model IBIS generuje identyczne efekty w przypadku metod połączenia równoległego. Wyniki były zgodne z oczekiwaniami. Po wymianie modelu na rezystor amplituda sygnału różnicowego w metodzie z rozdzielaniem linii przy dwóch odbiornikach wzrosła o ponad 700 mV i oscylowała wokół wartości 0 V (ponieważ efekty związane ze składową stałą napięcia nakładały się jednakowo na obydwie linie). Przedstawiony problem został zgłoszony do producenta, który to, ze względu na skrajnie niestandardowe i niezalecane wykorzystanie przetwornika, zalecił jedynie osobistą weryfikację na fizycznym sprzęcie.

W takim wypadku konieczne było podjęcie wstępnej decyzji dotyczącej stosowanej metody połączeń, aby móc opracować fizyczną realizację sprzętu, pozwalającą na wykonanie odpowiednich testów weryfikujących hipotezę dotyczącą błędnych modeli IBIS. W praktyce najbardziej obiecującą koncepcją było sterowanie linią asymetryczną, dlatego to ona została wybrana do wykonania testów na fizycznym sprzęcie. W przypadku potwierdzenia hipotezy stanowiła ona najlepsze rozwiązanie, niezależnie od wyników symulacji połączeń równoległych z wykorzystaniem modelu rezystora. Przemawiały za tym następujące argumenty:

- Metoda ta gwarantowała najlepsze dopasowanie linii długiej, a więc najmniejsze zniekształcenia sygnału.
- W przeciwieństwie do metod równoległych – nie wymagała stosowania kilkukrotnie większych ilości transferowanych bitów względem częstotliwości próbkowania, a co za tym idzie nie wprowadzała dodatkowych opóźnień.
- Jako jedyna nie wprowadzała konieczności stosowania dodatkowych programowych mechanizmów synchronizacji wyjść analogowych przetworników.
- Pozwalała na zastosowanie stosunkowo prostych połączeń w obwodzie drukowanym (najmniejsza ilość docelowych linii ze wszystkich wariantów), ułatwiając tym samym zachowanie poprawnych zależności czasowych, związanych z czasem propagacji sygnału w ścieżce.
- Gwarantowała efektywne trzykrotne zwiększenie liczby przetworników przy niezauważalnych zmianach w kosztach produkcji modułu Shuttler (pomijany w tym argumentie jest koszt zakupu dodatkowych przetworników, ponieważ ich liczba dla całego systemu – około 1000 sztuk – jest stała).

Do ostatniego z wymienionych punktów należy dodać, że metoda równoległego sterowania z rozdzielaniem linii przy odbiornikach teoretycznie mogłaby pozwolić nawet na czterokrotne zwiększenie liczby przetworników bez zwiększania kosztów produkcji (zakładając, że amplituda sygnału w symulacjach z wykorzystaniem modelu rezystora okazałaby się zgodna z rzeczywistością). W praktyce nie miało to jednak znaczenia – dla najmniejszego rozważanego układu FPGA dałoby to wynik 24 kanałów w module Shuttler, których płyta w formacie AMC nie byłaby w stanie zmieścić na swojej powierzchni obwodu drukowanego. Dlatego metoda ta w porównaniu z wybraną nie miała żadnych znaczących zalet.

6. Testy i pomiary koncepcji asymetrycznej transmisji danych

W pierwotnych planach testy i pomiary z fizycznym sprzętem miały ostatecznie dowieść, że zaproponowane rozwiązanie modułu Shuttler, pozwalające na optymalizację kosztów całego systemu, będzie działać niezawodnie. W praktyce okazało się, że konieczne było przeprowadzenie dodatkowych badań. Nowe cele postawione przed rozpoczęciem testów przedstawiały się następująco:

1. Weryfikacja, czy efekty uzyskane w symulacji (dotyczące przesunięcia składowej stałej napięcia oraz obniżenia spodziewanej wartości amplitudy sygnału prostokątnego) z wykorzystaniem potencjalnie wadliwego modelu IBIS przetwornika LTC2000 są zgodne z rzeczywistością.
2. W przypadku dowodu na różnice pomiędzy wynikami symulacji a wynikami pomiarów: weryfikacja z wykorzystaniem rzeczywistego układu, czy zastosowanie przedstawionej koncepcji połączeń pomiędzy układem FPGA a przetwornikiem analogowo-cyfrowym LTC2000 umożliwi prawidłowe i w pełni poprawne przekazywanie danych i działanie przetwornika w zakresie oczekiwanych częstotliwości.
3. Przy pozytywnych wynikach testów – oszacowanie maksymalnej częstotliwości, dla której zaproponowany interfejs będzie działał bezbłędnie.

6.1. Wykorzystywany sprzęt

DC2303A-A

DC2303A-A to moduł demonstracyjny dla przetwornika cyfrowo-analogowego LTC2000-16. Płyta posiada format FMC, dzięki czemu łatwo jest jąysterować przy wykorzystaniu modułów posiadających złącza na takie właśnie rozszerzenie. Przez złącze poprowadzono linie dla sygnałów danych, sygnału zegarowego danych oraz linie konfiguracyjne Serial Peripheral Interface (SPI). Płyta posiada także 4 złącza SubMiniature version A (SMA). Dwa z nich służą do wyprowadzenia analogowego różnicowego sygnału wyjściowego generowanego przez przetwornik, jeden stanowi miejsce podłączenia źródła sygnału zegarowego taktującego generację kolejnych próbek, zaś ostatni stanowi opcjonalne podłączenie dla sygnału diagnostycznego, którego funkcja jest konfigurowalna przez użytkownika.

Zastosowanie modułu demonstracyjnego zamiast własnego projektu dedykowanej płyty z zaimplementowanymi rozwiązaniami, przedstawionymi w niniejszej pracy, zostało podyktowane opóźniającym się finansowaniem na badania i produkcje modułu Shuttler. Z kolei płyta DC2303A-A została wypożyczona od jednego z podmiotów uczestniczących w projekcie, pod warunkiem nie wprowadzania nieodwracalnych zmian.

DC2303A-A został zaprojektowany do standardowego wykorzystania układu LTC2000 – posiada interfejs LVDS. Konieczne było zatem wprowadzenie zmian pozwalających na odwzorowanie zaproponowanego rozwiązania interfejsu. Na podstawie analizy projektu obwodu drukowanego, udostępnionego przez producenta[3], okazało się, że dla każdej linii danych występowało dokładnie jedno miejsce, w którym istniała fizyczna możliwość jakiegokolwiek ingerencji w badane linie sygnałowe, bez większego ryzyka uszkodzenia całego modułu. Były to punkty przelotek – łączących warstwę spodnią, na której zostało umieszczone złącze FMC, z wewnętrzną warstwą – przez które zostały poprowadzone ścieżki. Nie były to przelotki typu „zagrzebanego”, a więc na wierzchniej warstwie,

pod warstwą laminatu, wyprowadzone zostały fragmenty metalizacji. Wspomniane punkty znajdowały się w odległości około 3 cm od samego układu przetwornika. Była to sytuacja niefortunna ze względu na fakt, że terminacja powinna zostać umieszczona jak najbliżej punktu końcowego, jakim jest przetwornik. Niestety przelotki, które doprowadzają sygnał z warstwy wewnętrznej do warstwy wierzchniej, gdzie znajdują się już bezpośrednio wyprowadzenia LTC2000, to przelotkami typu „zagrzebanego”, w związku z czym nie było żadnego fragmentu metalizacji linii sygnałowej po drugiej stronie płyty. Sam zaś przetwornik posiada obudowę typu BGA, dlatego zarówno przelotka, jak i wyprowadzenie znajdowały się pod układem i jest niemożliwe było doprowadzenie jakichkolwiek dodatkowych połączeń. Wprowadzenie modyfikacji w tym nieoptymalnym położeniu na obwodzie drukowanym było koniecznością. Zmieniło to jednak cel stawiany przed pomiarami. Przy badaniach układu posiadającego tego typu niskiej jakości terminacje, przypuszczalnie niemożliwe byłoby zrealizowanie w pełni punktu 2. i 3. opisanego w celach pomiarów. Potencjalnie nieudana próba wysterowania odpowiednich poziomów logicznych układu FPGA nie oznaczałaby, że przedstawiona koncepcja nie ma szans realizacji na wymaganym sprzęcie. Mogłaby ona być także związana ze zbyt dużymi zniekształceniami sygnału, spowodowanymi zastosowaną terminacją. Z tego samego powodu sens utraciło doświadczalne wyznaczenie częstotliwości granicznej, dla której interfejs działa poprawnie. Dlatego zdefiniowano dwa dodatkowe cele, których realizacja uzależniona jest od istnienia błędów w plikach IBIS. Przedstawiały się one następująco:

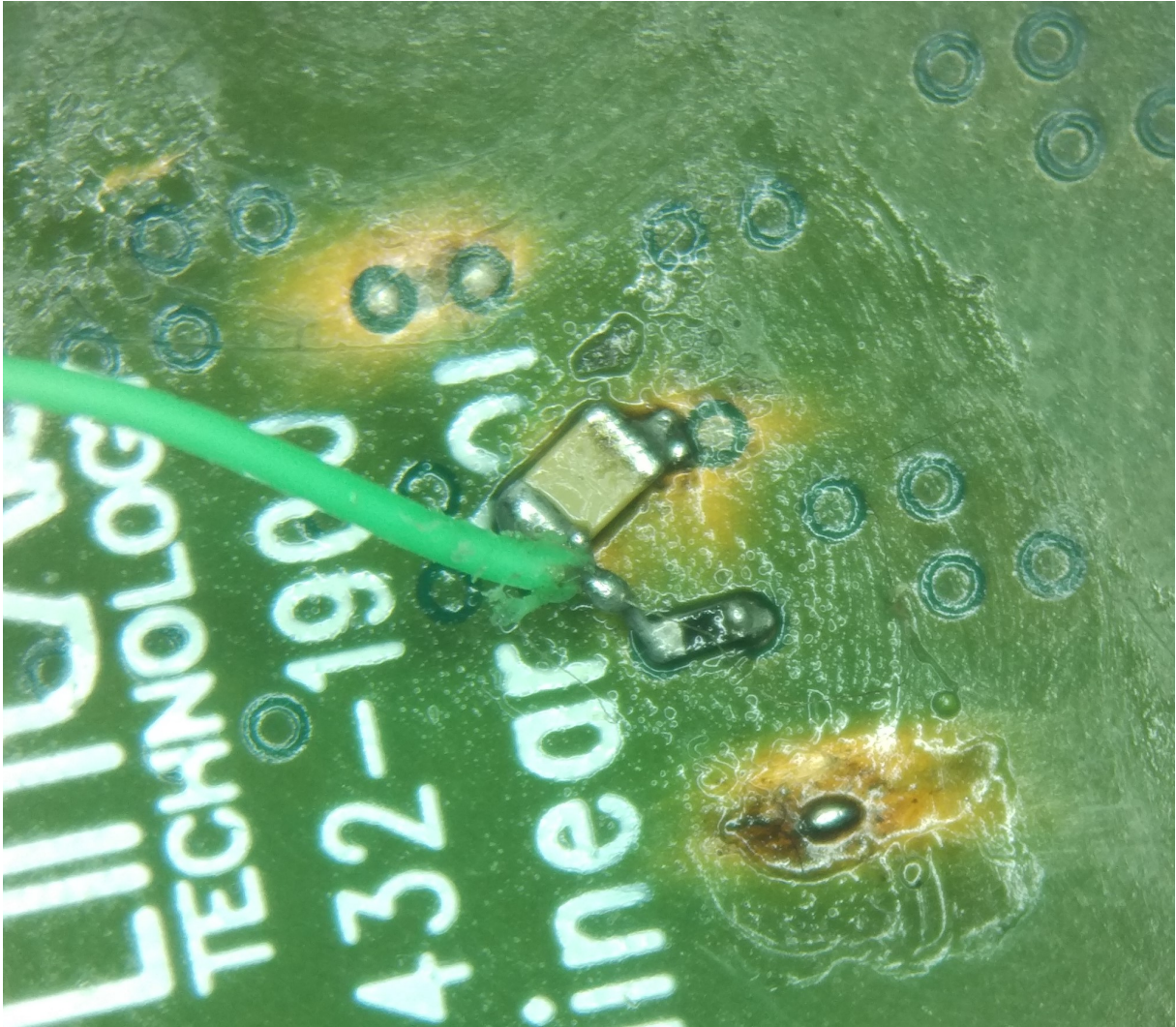
1. Analiza oscylogramów sygnału danych otrzymanych dla wyższych częstotliwości rzędu 100 MHz oraz porównanie wyników z rezultatami osiągniętymi w symulacji przy modelowaniu przetwornika pojedynczym rezystorem – wymagało to dodatkowo zamodelowania badanego układu i wykonania nowych symulacji.
2. W przypadku zgodności wyników symulacji i pomiarów – wyznaczenie częstotliwości granicznej zaproponowanego interfejsu na podstawie symulacji zamodelowanego układu docelowego (projektu obwodu drukowanego płyty Shuttle) z zastosowaniem modelowania przetwornika jako pojedynczego rezystora.

W ramach implementacji badanego interfejsu wykonano następujące czynności:

1. Usunięto warstwę lakieru izolującego w celu odsłonięcia metalizacji pary różnicowej, którą doprowadzany jest cyfrowy sygnał sterujący wartością bitu 14. przetwornika (metalizacja dla bitów 15. i 16. była dodatkowo pokryta warstwą opisową).
2. Usunięto warstwę lakieru izolującego w celu odsłonięcia fragmentu metalizacji płaszczyzny masy w najbliższym sąsiedztwie odsłoniętej pary różnicowej.
3. Wlutowano rezystor o rezystancji 82Ω pomiędzy linie pary różnicowej.
4. Dolutowano do linii P pary różnicowej przewód w celu jego późniejszego podłączenia do napięcia terminującego V_{TT} .
5. Dolutowano kondensator o pojemności 100 nF pomiędzy linię P pary różnicowej i odsłoniętą masę w celu zapewnienia odsprężenia napięcia terminującego i jednoczesnego zwarcia do masy linii P z punktu widzenia szybkich sygnałów przesyłanych linią N pary różnicowej.

Wprowadzone zmiany zostały zaprezentowane na rys. 6.1.

Ze względu na brak możliwości wykorzystania interfejsu SPI poprzez złącze FMC (co zostanie wyjaśnione w kolejnym punkcie) dolutowano także przewody do punktów testowych SPI w celu umożliwienia konfiguracji przetwornika cyfrowo-analogowego przez inne urządzenie. Płyta posia-



Rysunek 6.1: Zdjęcie z mikroskopu modyfikacji na płycie DC2303A-A.

dała dwie możliwe opcje zasilania: bezpośrednie podłączenie napięcia 5 V lub wykorzystanie zasilania 12 V pochodzącego z płyty bazowej, do której jest podłączana karta FMC. Zastosowano drugi wariant, aby ograniczyć ilości przewodów w układzie pomiarowym (kosztem większych strat mocy, które były w tym wypadku bez znaczenia), co wiązało się z koniecznością dolutowania koralka ferrytowego L6 (oznaczenie zgodne ze schematami producenta). Należy podkreślić, że zgodnie z wymaganiami wszystkie wprowadzone zmiany nie uniemożliwiają powrotu płyty do pierwotnego stanu. Wyjątkiem jest tu fragment zdartego lakieru, jednak w żaden sposób nie wpływa on na funkcjonowanie modułu – po ręcznym lakierowaniu stanowiłby jedynie zmianę wizualną.

Moduł Sayma AMC

Moduł Sayma AMC jest jednym z głównych modułów projektu Sinara [21]. Wraz ze swoją siostrzaną płytą Sayma RTM odpowiadają za generację i odczyt przebiegów o dużej częstotliwości przy wykorzystaniu 8 kanałów 16-bitowych przetworników C/A pracujących do częstotliwości 1,2 GS/s i 8 kanałów przetworników A/C 16 bitowych z częstotliwością próbkowania 125 MS/s.

Z punktu widzenia pomiarów najistotniejszą cechą płyty Sayma AMC jest posiadanie złącza FMC High-pin count (HPC) (choć fizycznie podłączone zostały linie przewidziane wyłącznie dla rozmiaru Low-pin count (LPC)) oraz układu FPGA Kintex UltraScale XCKU040-1FFVA1156C sterującego płytą. Był to preferowany model do zastosowania w module Shuttler, zaś testy z jego wykorzystaniem pozwoliły na zweryfikowanie potencjalnego działania dla całej rodziny układów Kintex UltraScale.

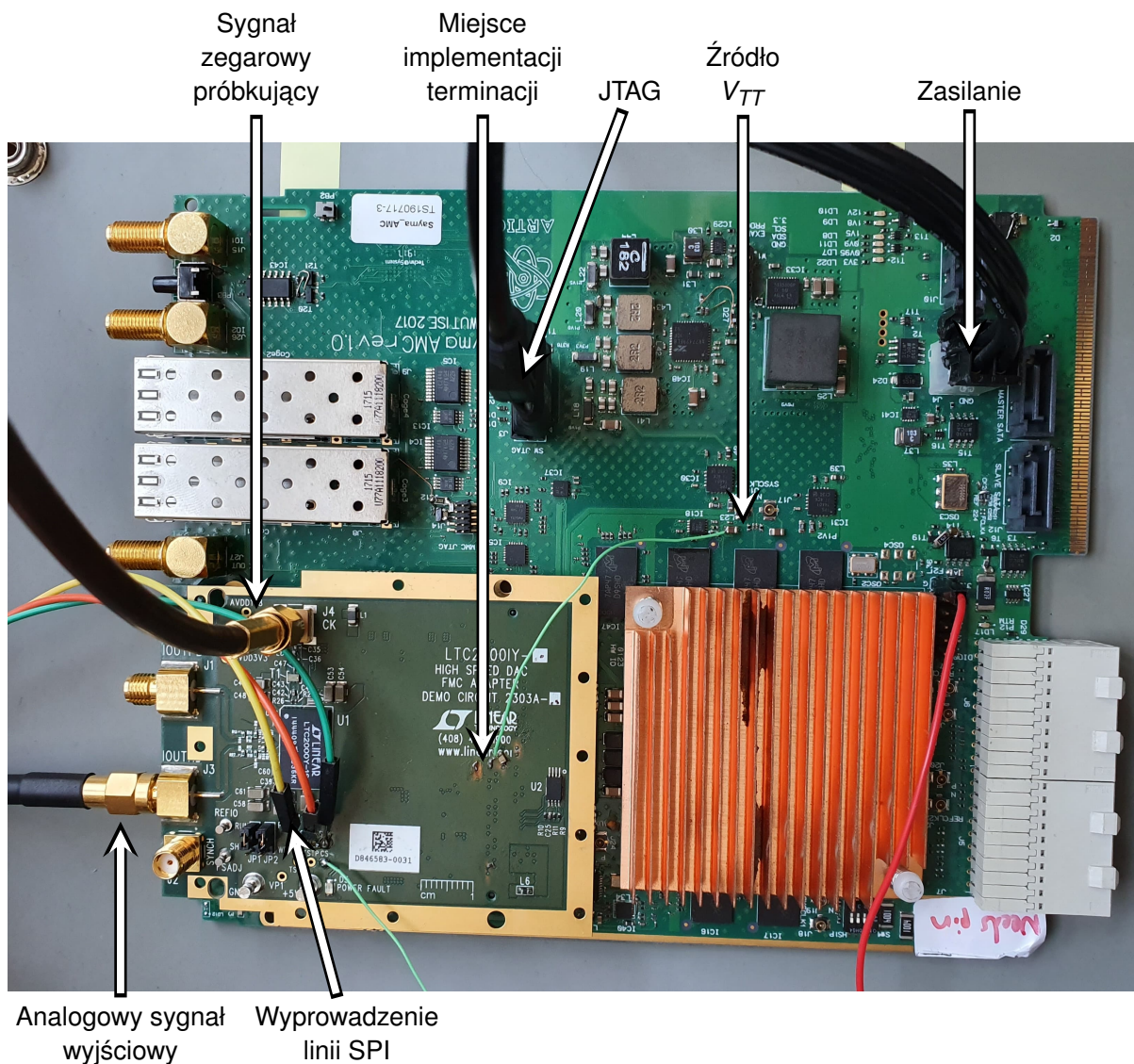
Brak połączeń typowych dla złącz FMC HPC spowodował, że niemożliwa była konfiguracja przetwornika poprzez interfejs SPI, przewidziany przez producenta płyty demonstracyjnej. Dlatego konieczne było wprowadzenie przewodowego połączenia do punktów testowych na module przetwornika, opisanych w poprzednim punkcie. W konsekwencji Sayma AMC sterowała przetwornikiem LTC2000, zaś do jego konfiguracji potrzebne było wykorzystanie innego urządzenia.

Płyta Sayma AMC była także źródłem napięcia terminującego badanego interfejsu. Posiada ona pamięć SDRAM DDR3 wraz z regulatorem terminującym „sink/source” TPS51200DRCT. Tego typu pamięć pracuje przy napięciu zasilania 1,5 V i wymaga stosowania napięcia terminującego V_{TT} o wartości równej połowie wartości napięcia zasilania, a więc 0,75 V. Ponieważ badany interfejs wymagał, aby wartość ta wynosiła 0,9 V, konieczna była zmiana napięcia referencyjnego doprowadzonego do regulatora. Jest ono generowane przez dzielnik rezystorowy podłączony do linii zasilania pamięci i składa się z dwóch rezystorów po 10 k Ω każdy. Zmiana rezystora podłączonego do masy na wartość 15 k Ω pozwoliła na generację oczekiwanego napięcia 0,9 V. Dodatkowo dolutowano przewód doprowadzający napięcie V_{TT} do badanego interfejsu. Moduł Sayma AMC wraz z modułem demonstracyjnym DC2303A-A stanowiły całość badanego układu i zostały przedstawione na rys. 6.2.

Na zdjęciu ukazane zostały także wszystkie konieczne połączenia do urządzeń zewnętrznych, a więc: przewody SMA dla sygnału wyjściowego z przetwornika oraz dla wejściowego sygnału zegarowego taktującego generację kolejnych próbek, podłączony programator Joint Test Action Group (JTAG), przewód zasilania oraz kolorowe przewody dla SPI.

Pozostały sprzęt

- Generator sygnału mikrofalowego i RF SMB100A – wykorzystany jako źródło sygnału zegarowego taktującego próbkowanie przetwornika,
- zasilacz laboratoryjny DF1723003TC – generujący napięcie zasilania +12 V dla całego badanego układu,
- zestaw uruchomieniowy ZL27ARM z procesorem STM32F103 – wykorzystany do konfiguracji przetwornika cyfrowo-analogowego poprzez interfejs SPI,
- konwerter USB-Universal asynchronous receiver-transmitter (UART) z układem FT232RL – umożliwiający użytkownikowi komunikację z procesorem STM32,
- oscyloskop MSO 4304,
- pasywna sonda P6139A firmy Tektronix pasmo 500 MHz,
- multimetr uniwersalny UT71E firmy UNI-T.



Rysunek 6.2: Płyta Sayma AMC wraz z zamontowanym modulem DC2303A-A.

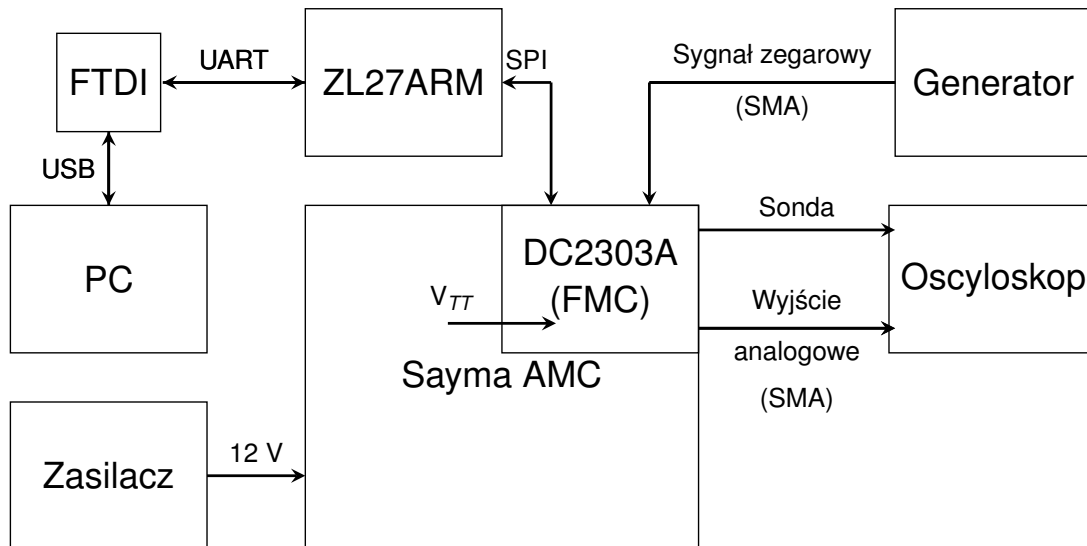
6.2. Układ pomiarowy

Po wykonaniu wszelkich modyfikacji został złożony układ pomiarowy, którego schemat przedstawiono na rys. 6.3, zaś fizyczne zestawienie na rys. 6.4.

Procesor STM32 na płycie ZL27ARM został zaprogramowany tak, aby stanowić konwerter UART-SPI i umożliwić użytkownikowi modyfikowanie rejestrów przetwornika LTC2000 w trakcie testów bez konieczności przeprogramowywania procesora. Kod programu dla procesora został umieszczony w załączniku B.2.

Plan pomiarów przedstawiał się następująco:

1. Pomiar natężenia prądu na przewodzie napięcia terminującego z wykorzystaniem miernika uniwersalnego. Miało to na celu weryfikację, czy obwody wejściowe przetwornika cyfrowo-analogowego wpływają na składową stałą napięcia na liniach danych zgodnie z symulacjami z wykorzystaniem modelu IBIS.



Rysunek 6.3: Schemat połączeń układu pomiarowego pomiędzy wykorzystywanymi modułami i sprzętem.

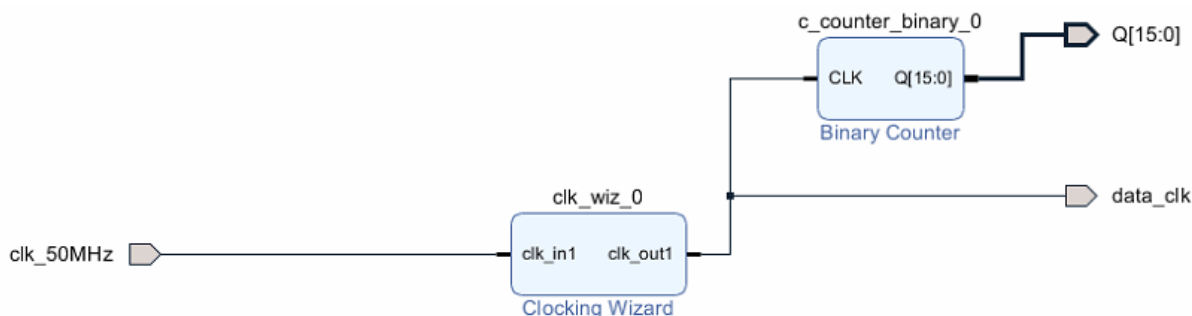


Rysunek 6.4: Zestawienie stanowiska pomiarowego – niewidoczny pozostaje wyłącznie komputer PC.

2. Pomiar poprzez sondę sygnału wejściowego dochodzącego do przetwornika (mierzony na dodatkowym rezystorze terminującym) oraz napięcia terminującego dla częstotliwości 100 MHz.
3. Równoległy do pomiaru sondą pomiar sygnału wyjściowego przetwornika, który został podłączony kablem koncentrycznym o impedancji 50Ω ze złączami SMA bezpośrednio do oscyloskopu w celu weryfikacji poprawnego działania przetwornika.

6.3. Pomiary i wnioski

Całość konfiguracji układu FPGA składała się z kodu VHASIC (Very High Speed Integrated Circuit) Hardware Description Language (VHDL), zamieszczonego w dodatku B.1 oraz projektu blokowego przedstawionego na rys. 6.5. Widoczne na schemacie źródło sygnału zegarowego posiada często-



Rysunek 6.5: Część blokowa projektu wsadu FPGA wykorzystywanego w trakcie pomiarów.

ciwość 50 MHz. Licznik 16-sto bitowy służy do generacji przykładowego sygnału piłokształtnego, który może zostać dostarczony do przetwornika. Zdefiniowano 17 par połączeń pomiędzy FPGA a przetwornikiem – 16 związanych z poszczególnymi bitami generowanej próbki oraz 1 para sygnału zegarowego dla przesyłanych bitów. Wszystkim połączeniom przypisano standard LVDS z wyjątkiem połączeń bitu 14. – w tym przypadku zastosowano standard LVCMOS18, użyty w trakcie symulacji. W trakcie wszystkich pomiarów stosowano ten sam kod oraz schemat blokowy z dwoma wyjątkami:

- zmieniano częstotliwości sygnału zegarowego generowanych przez układ FPGA,
- porty wyjściowe danych były podłączane do wyjść licznika lub ustawiane w stan wysokiej impedancji.

W związku z tym każdy opis poszczególnych pomiarów zawiera informację wyłącznie o zmieniających się elementach konfiguracji. Wszystkie pomiary wykonane zostały przy w pełni działającym przetworniku. Procedura jego uruchomienia (bez wykorzystywania dodatkowej funkcjonalności) wymagała dokonania kilku wpisów do rejestrów poprzez interfejs SPI i przedstawiała się następująco:

- wpis 0x01 do rejestru 0x01, generujący programowy reset urządzenia,
- wpis 0x01 do rejestru 0x03, uruchamiający proces odbioru sygnału zegarowego taktującego przesył danych,
- wpis 0x06 do rejestru 0x04, uruchamiający proces odbioru cyfrowego sygnału danych w trybie jednoportowym,
- po czasie minimum 1 ms (czas wymagany na zakończenie synchronizacji i inicjalizacji) – wpis 0x0E do rejestru 0x04, włączający analogowe wyjście przetwornika.

6.3.1. Pomiar 1. – badanie wpływu przetwornika LTC2000 na składową stałą napięcia linii sygnałowej

Ustawienia

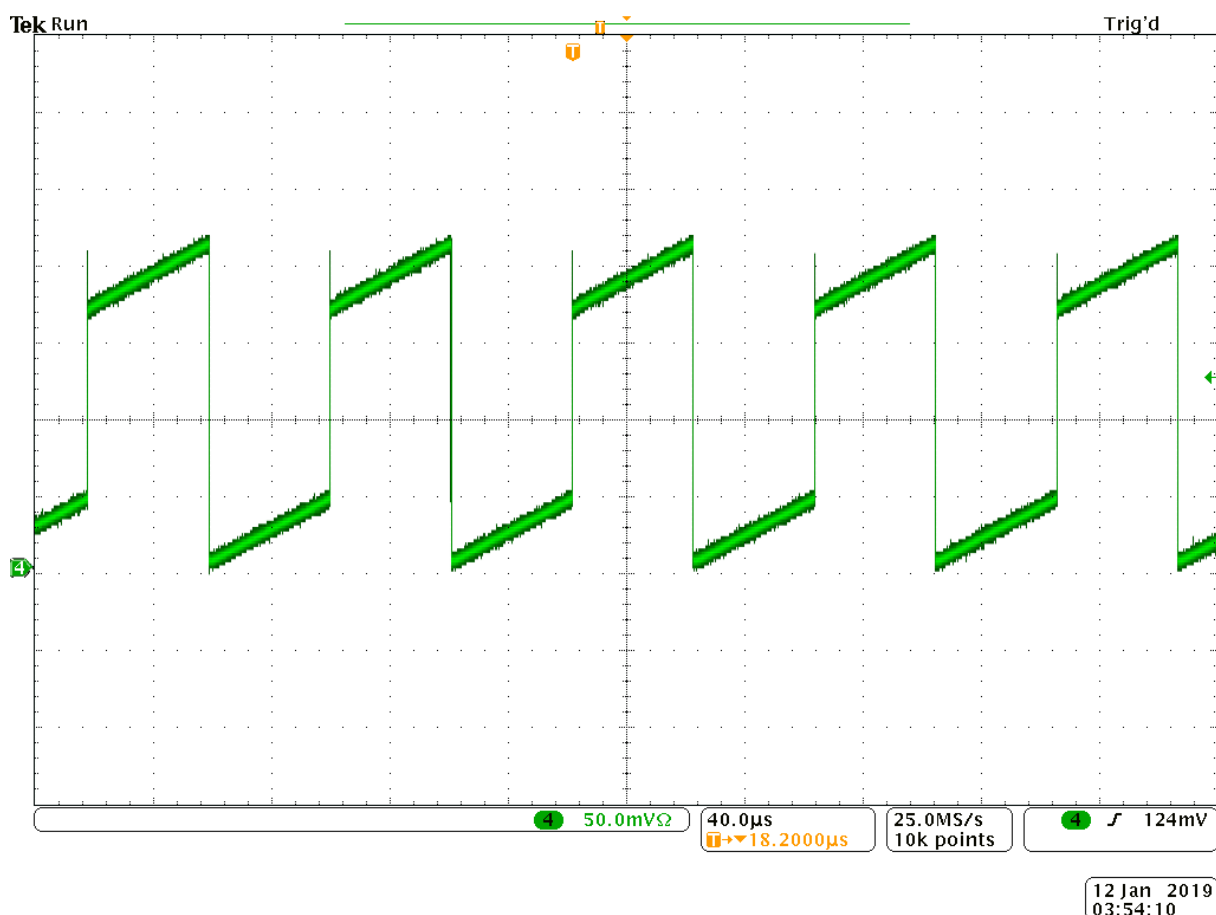
W celu poprawnego zmierzenia wpływu przetwornika LTC2000 na wejściowe linie danych, konieczne było wyeliminowanie potencjalnego wpływu układu FPGA. Dlatego jego wyprowadzenia podłączone do badanych linii (na których podawany jest bit 14. próbki) zostały skonfigurowane w stan wysokiej impedancji. Tak samo skonfigurowano linie do sterowania bitem 15., zaś do linii sterującej 16. bitem (najbardziej znaczącym) przypisano 14. bit licznika. Dzięki temu spodziewanym sygnałem wyjściowym przetwornika był sygnał piłokształtny z pojedynczym przeskokiem w połowie okresu narastania. Z kolei do pozostałych linii danych pomiędzy FPGA a przetwornikiem przypisano odpowiadające bity licznika. Częstotliwość sygnału zegarowego próbkującego ustawiono na 400 MHz (300 MHz to wartość minimalna rekomendowana przez producenta płyty demonstracyjnej ze względu na znajdujący się na płycie symetryzator konwertujący asymetryczny sygnał zegarowy na sygnał różnicowy). Częstotliwości wewnętrznego sygnału zegarowego ustawiono na wartości 200 MHz. Przewód łączący linię P badanego interfejsu z napięciem terminującym V_{TT} został chwilowo wylutowany – połączenie w tym pomiarze zapewnione zostało przez miernik uniwersalny w konfiguracji amperomierza. Głównym celem generacji sygnału przez przetwornik cyfrowo-analogowy było upewnienie się, że układy odbiorników sygnałów cyfrowych będą reagować tak, jak w standardowym trybie pracy w przypadku realizacji zaproponowanego rozwiązania. Dodatkowo pozwoliło to zweryfikować, czy dostarczony układ nie był uszkodzony.

Wyniki

Wskazanie amperomierza, mierzącego natężenie prądu pomiędzy regulatorem napięcia V_{TT} a badaną terminacją przetwornika, wynosiło 0 mA przy wykorzystaniu najmniejszego zakresu pomiarowego urządzenia. Jednocześnie za pomocą sondy oscyloskopowej zweryfikowano wartość napięcia na badanych liniach pary różnicowej – średnia wartość napięcia wynosiła 905 mV na obydwu liniach. Potwierdzono także prawidłowe działanie samego przetwornika przez generację sygnału piłokształtnego z oczekiwanym przeskokiem. Oscylogram sygnału wyjściowego przedstawiony został na rys. 6.6.

Wnioski

Wyniki pierwszego pomiaru, dotyczącego poboru prądu ze źródła napięcia terminującego V_{TT} , dowodzi, że musi istnieć błąd w modelach IBIS dla przetwornika LTC2000. Gdyby zgodnie z wynikami symulacji układ ten próbował zmienić wartość składowej stałej napięcia na liniach danych na 1,05 V, musiałoby dać to efekt w postaci przepływu prądu do lub z układu generującego napięcie V_{TT} . Wyniki pozwoliły na wysnucie hipotezy, że modelowanie cyfrowych obwodów wejściowych przetwornika jako pojedynczego rezystora o rezystancji 120 Ω stanowi prawidłowe postępowanie w przypadku próby symulacji zachowania tego układu.



Rysunek 6.6: Oscylogram analogowego sygnału wyjściowego z przetwornika wykonany w ramach pomiaru 1..

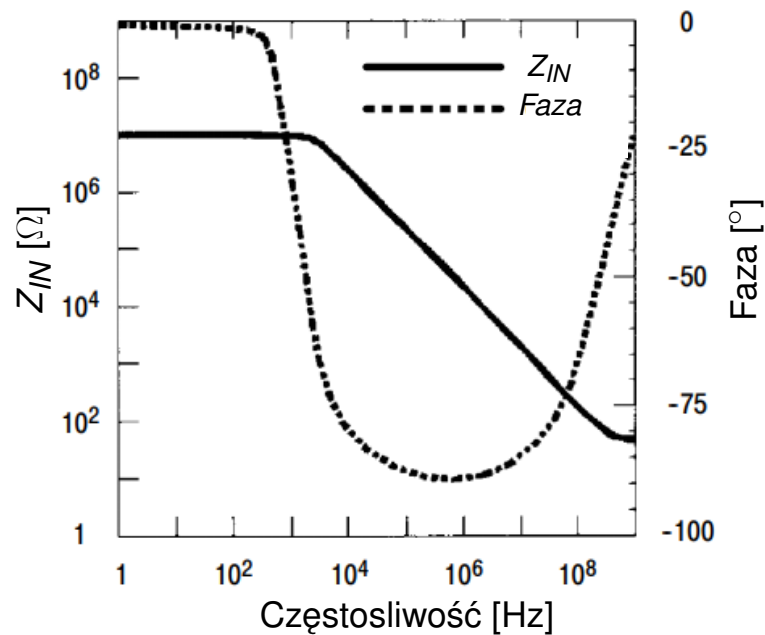
6.3.2. Pomiar 2. – badanie zaproponowanego interfejsu dla częstotliwości 100 MHz

Ustawienia

W przypadku opisywanego właśnie pomiaru istotne było wyłącznie badanie zaproponowanego interfejsu w czasie standardowej pracy. Częstotliwość wewnętrznego sygnału zegarowego została ustawiona na 200 MHz i jednocześnie sygnał ten wyprowadzono jako sygnał taktujący przesył danych. Wszystkie wyprowadzenia danych LVDS zostały skonfigurowane w tryb wysokiej impedancji, zaś do badanej linii podłączono najmniej znaczący bit licznika. Oznacza to przesyłanie cyfrowo sygnału prostokątnego o częstotliwości 100 MHz i wypełnieniu 50%. Zgodnie ze specyfikacją przetwornika częstotliwość sygnału zegarowego taktującego wyjścia analogowe ustalona została na 400 MHz.

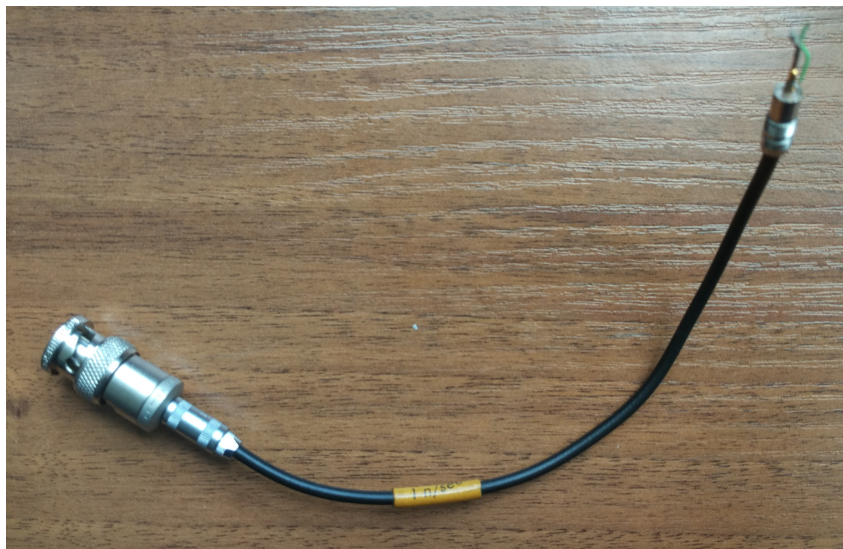
Wyniki pierwszych pomiarów tak skonfigurowanego układu widoczne są na rys. 6.10. Widoczny sygnał przypominający typowy sygnał sinusoidalny o częstotliwości 100 MHz świadczy o dokonaniu niepoprawnego pomiaru. Wszelkie nieoptymalne rozwiązania zastosowane w badanym układzie, opisane we wcześniejszych podrozdziałach, nie mogłyby bowiem odpowiadać za taki kształt sygnału. Problemem była stosowana sonda oscyloskopowa. Jej pojemność sprawiła, że czas narastania sygnału był zbyt duży, aby móc w pełni odwzorować mierzony sygnał. Dodatkowo wraz ze wzrostem częstotliwości drastycznie spada impedancja wejściowa sondy, co pokazane zostało na

rysunku 6.7. Dla badanego w tym pomiarze sygnału wynosiła ona około 100Ω , co oznacza, że znacząco wpływała na parametry obwodu terminującego.



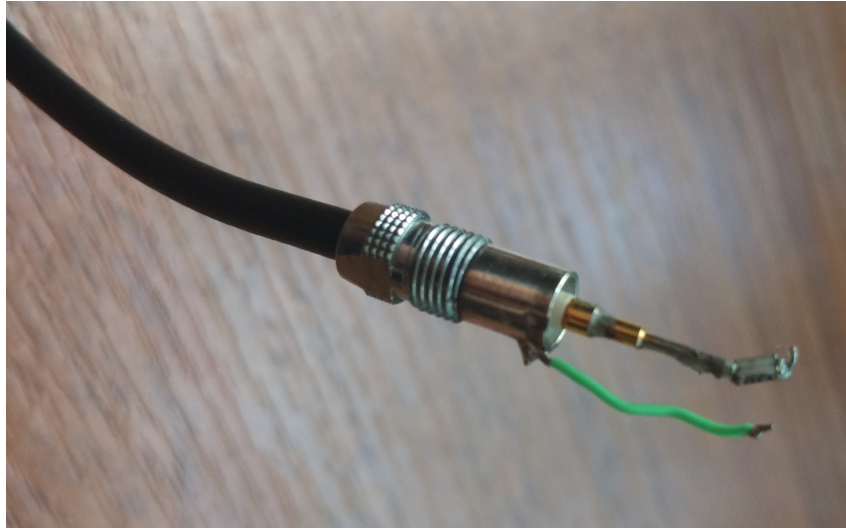
Rysunek 6.7: Wykres zależności impedancji i przesunięcia fazowego w zależności od częstotliwości sygnału sondy oscyloskopowej P6139A [37]

Ze względu na brak dostępu do lepszej jakości sondy, skonstruowano własną sondę widoczną na rys. 6.8 i 6.9. Wykorzystano w tym celu krótki odcinek przewodu koncentrycznego o impedancji 50Ω , zakończonego złączem typu LEMO, przeznaczonego do przesyłania szybkich sygnałów. Na dru-



Rysunek 6.8: Konstrukcja utworzonej sondy oscyloskopowej.

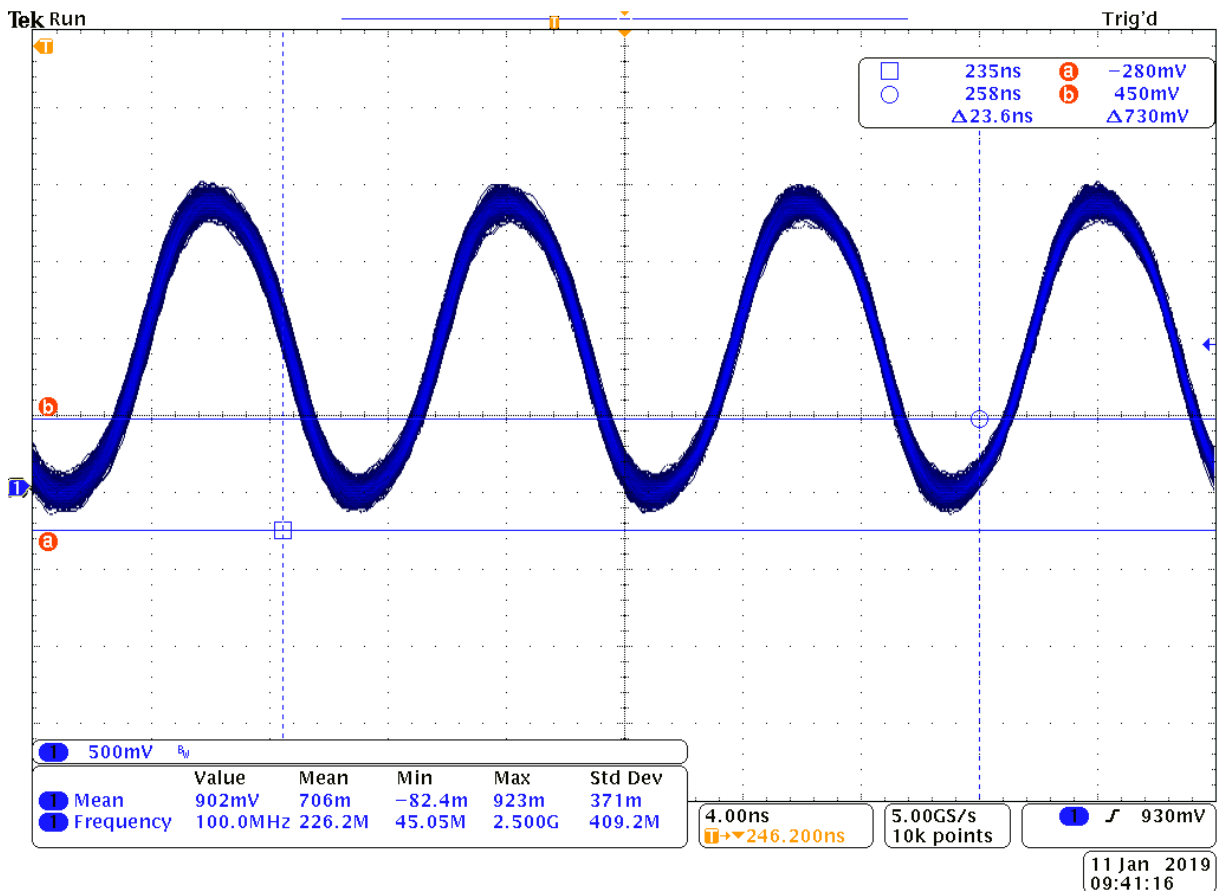
gim końca dolutowano przewód masy o długości około 1 cm, zaś do części sygnałowej równoległe do siebie dwa rezystory o rezystancji 910Ω każdy, tworząc rezystancję 455Ω . Uwzględniając konfigurację kanału wejściowego oscyloskopu, ustawioną na impedancję wejściową 50Ω , osiągnięto typowe dla sondy 10-krotne tłumienie sygnału. Równoległe włączenie jej rezystancji wejściowej



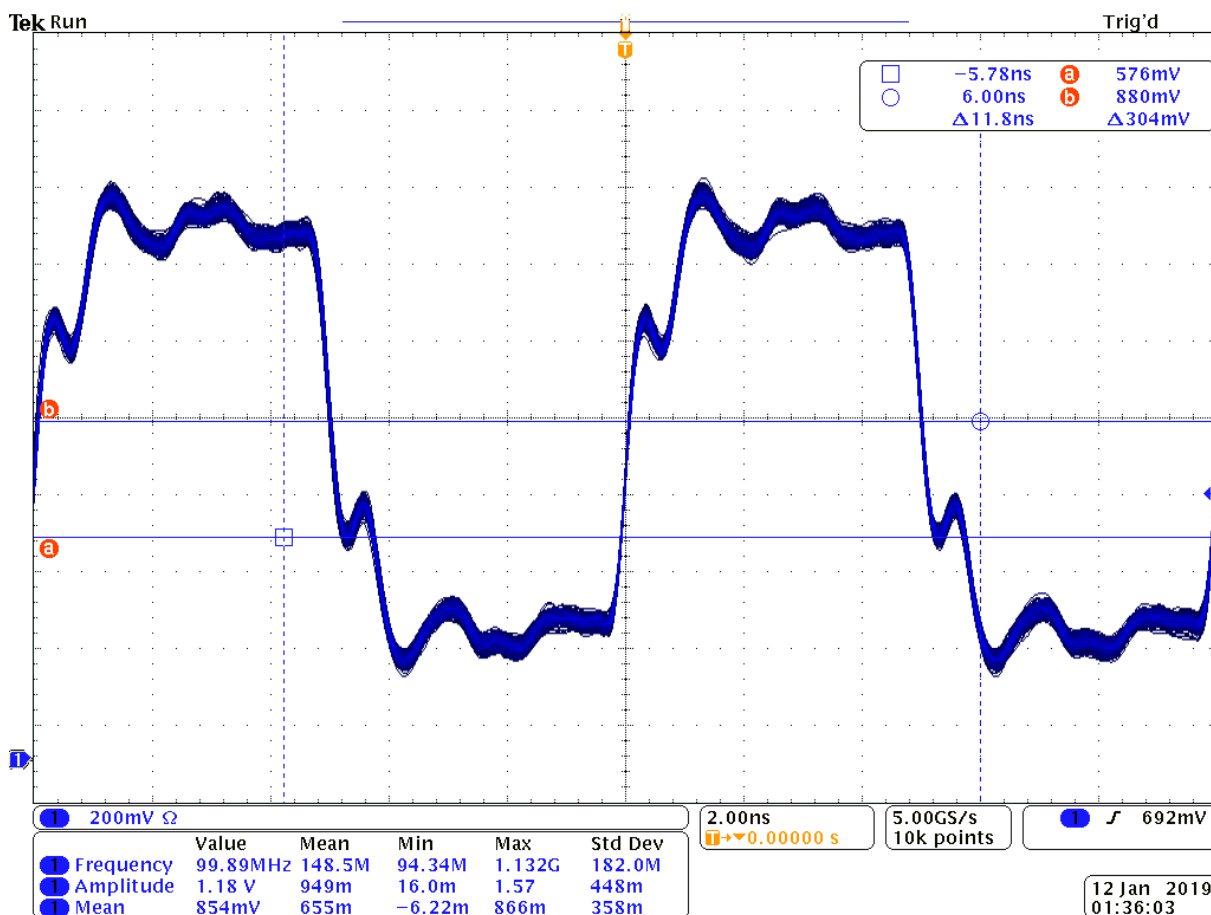
Rysunek 6.9: Konstrukcja utworzonej sondy oscyloskopowej – zbliżenie na końcówkę pomiarową.

do mierzonego obwodu zmniejszyło rezystancję obciążenia o około 5Ω . Wyniki nowego pomiaru przedstawione zostały na rys. 6.11. Dodatkowo sprawdzono sygnał wyjściowy przetwornika – przedstawiony na rys. 6.12.

Wyniki



Rysunek 6.10: Oscylogram linii danych – generacja sygnału prostokątnego 100 MHz, pomiar sondą P6139A.

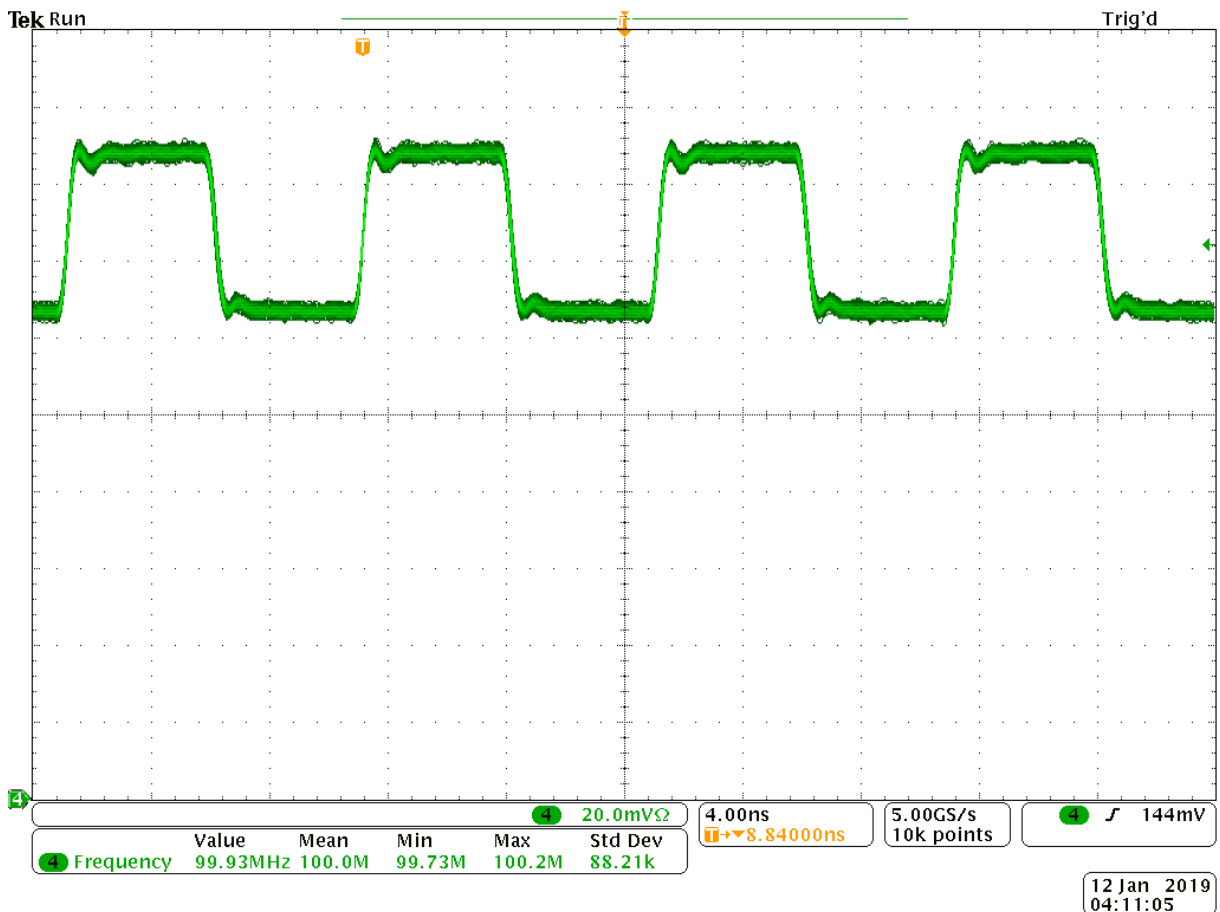


Rysunek 6.11: Oscylogram linii danych – generacja sygnału prostokątnego 100 MHz, pomiar sondą własną.

Wnioski

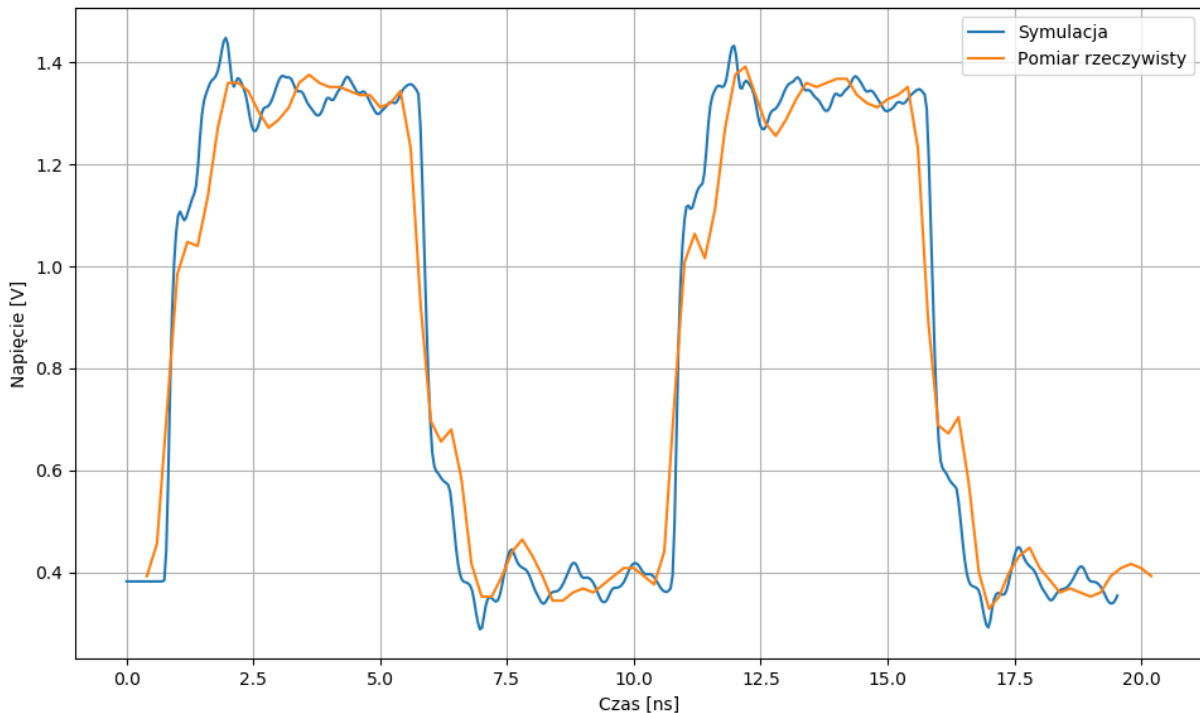
W kontekście działania przetwornika cyfrowo-analogowego wyniki były zadowalające. Sygnał zachowywał odpowiednie poziomy logiczne wymagane przez odbiornik, zaś sam odbiornik generował prawidłowy sygnał analogowy. Widoczne zniekształcenia nie osiągały dużych wartości i objawiały się przede wszystkim przy zmianie stanu logicznego w postaci niemonotonicznych zboczy. Jest to zachowanie charakterystyczne dla zastosowania zbyt niskiej impedancji terminującej. Należy pamiętać, że badany układ nie został stworzony jako dedykowany do tego typu sterowania. Przewodzona linia, która w docelowym układzie powinna posiadać impedancję charakterystyczną 50Ω , jest jedną z linii pary różnicowej. Projektowana była zatem do osiągnięcia impedancji różnicowej o wartości 100Ω . Dwie ścieżki o impedancji własnej 50Ω dają impedancję różnicową o wartości 100Ω wyłącznie w przypadku ich nieskończonego oddalenia od siebie. Zbliżanie ścieżek (co jest zabiegiem pożądanym w celu osiągnięcia sprzężenia i równomiernego wpływu zakłóceń na obydwie linie) powoduje zmniejszanie impedancji różnicowej. Dlatego też z punktu widzenia sygnału na pojedynczej linii (tak jak w badanym przypadku) impedancja charakterystyczna wynosiła około 60Ω i zaterminowana została impedancją około 45Ω (po uwzględnieniu wpływu sondy pomiarowej o rezystancji wejściowej 500Ω).

Przeprowadzone badania pozwoliły na wysunięcie wniosku, że zaproponowana metoda niestandardowego połączenia układu FPGA oraz przetwornika cyfrowo-analogowego z interfejsem LVDS



Rysunek 6.12: Sygnał wyjściowy z przetwornika cyfrowo-analogowego w trakcie pomiaru 2..

mogła zostać zaimplementowana w projektowanym module Shuttler, zapewniając niezawodne działanie w zakresie oczekiwanych przez naukowców częstotliwości. Jak zostało już jednak wspomniane, wymagania te rosną wraz z poziomem zaawansowania badań nad technologią kwantową, dlatego wartościowe było oszacowanie maksymalnej wartości częstotliwości, dla jakiej opracowana metoda będzie miała zastosowanie. Ponieważ dostępny sprzęt nie pozwalał na uzyskanie wiarygodnych wyników dla większej częstotliwości (drastycznie zwiększające się zniekształcenia związane z niedoskonałością sprzętu pomiarowego oraz niedoskonałością badanego układu) postanowiono oprzeć się na wynikach symulacyjnych. Należało jednak w pierwszej kolejności wykazać, że modelowanie odbiornika w postaci pojedynczego rezystora było prawidłowym rozwiązaniem. W tym celu wykonano dodatkowe symulacje. Badany układ został zamodelowany przy wykorzystaniu programu Altium Designer, za pomocą którego wyeksportowano pliki modeli dla programu HyperLynx na podstawie plików projektowych obwodów drukowanych płyty Sayma AMC oraz DC2303A-A. Następnie na podstawie danych modeli płyt wygenerowano symulowany układ w postaci schematów elektrycznych. Tam wprowadzono zaimplementowane zmiany – dodano rezystor i kondensator przy odpowiednich przelotkach oraz podłączono napięcie terminujące, symulując przewód jako linię długą. Układ odbiornika zastąpiono pojedynczym rezystorem. Uwzględniono także obecność sondy pomiarowej w postaci rezystora podłączonego do punktu pomiarowego i masy. Otrzymane wyniki rzeczywistego pomiaru oraz wykonanych dodatkowych symulacji zestawiono ze sobą i przedstawiono na rys. 6.13.

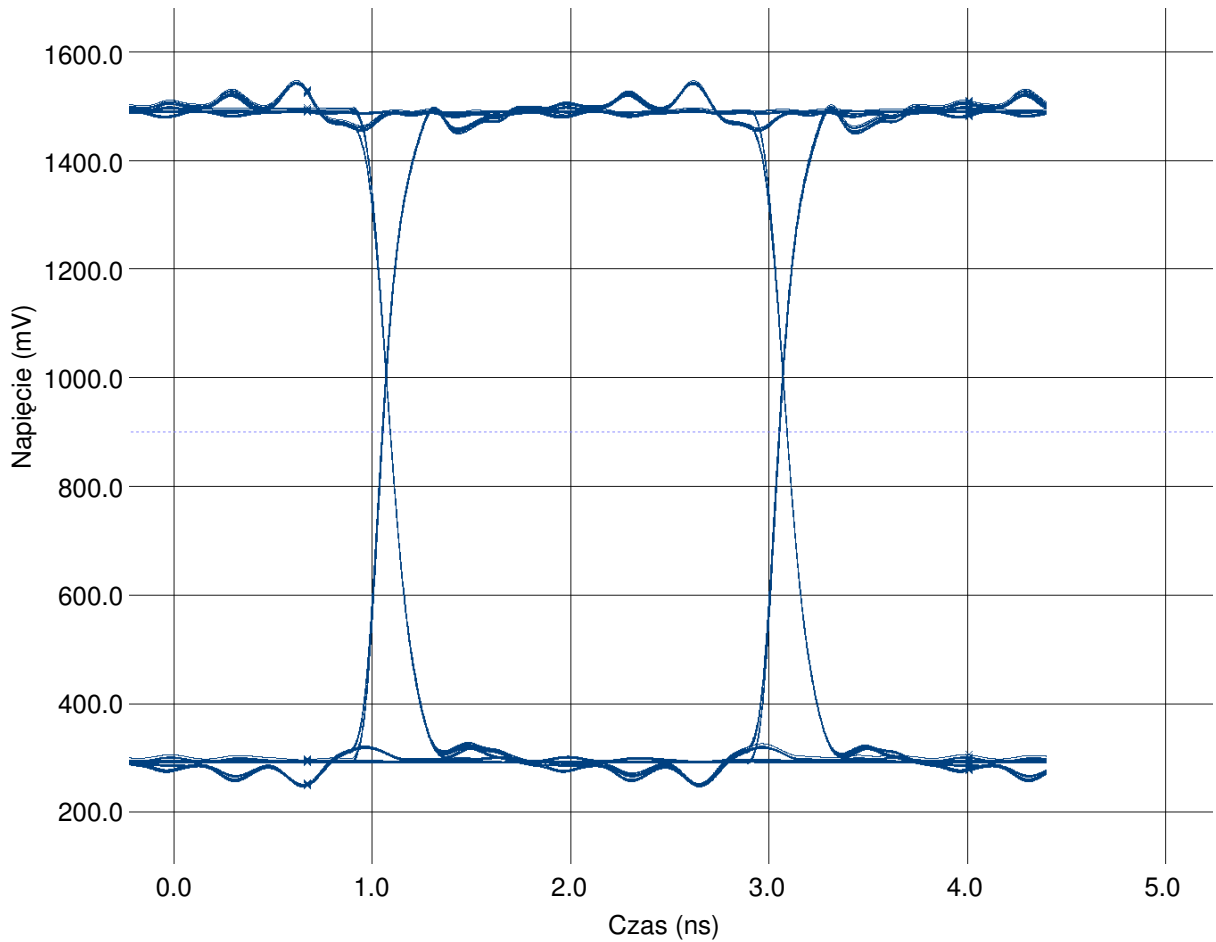


Rysunek 6.13: Porównanie wyników pomiaru napięcia na linii sygnałowej mierzonego na dodatkowym rezystorze terminującym dla sygnału prostokątnego o częstotliwości 100 MHz.

Uzyskane wyniki są do siebie zbliżone – różnią się przede wszystkim łagodniejszym zboczem przy zmianie stanu logicznego dla pomiaru rzeczywistego, co może być efektem ograniczonego pasma narzędzi pomiarowych. Wyciągnięto wniosek, że zastosowanie modelu odbiornika LVDS przetwornika cyfrowo-analogowego w postaci pojedynczego rezystora było działaniem pozwalającym na uzyskanie wiarygodnych wyników symulacji.

6.3.3. Oszacowanie maksymalnej częstotliwości pracy badanego interfejsu

W celu dokonania jak najbardziej precyzyjnych symulacji wykonano prosty „projekt” obwodu drukowanego w formacie AMC. Rozmieszczono wstępnie układ FPGA oraz przetwornik w najdalszym możliwym punkcie płyty Shuttle oraz zaimportowano konfigurację warstw z projektu Sayma AMC (która w przypadku zastosowania preferowanego układu FPGA pozostanie bez zmian). Dzięki temu możliwe było połączenie przetwornika z układem FPGA ścieżką, której długość z dość dobrym przybliżeniem odpowiadała docelowej długości linii długiej. W tym wypadku zaimplementowano finalny projekt obwodu drukowanego dla elementów wejścia danych przetwornika LTC2000, przez co nie należało się spodziewać obecności dotychczas obserwowanych zniekształceń. Wyniki symulacji zostały przedstawione na rys. 6.14. Widoczny diagram oka nie wykazał żadnych znaczących zniekształceń sygnału. Oznacza to, że impedancja rzeczywiście została poprawnie dopasowana i jedynym ograniczeniem dla docelowego modułu będą częstotliwości osiągnane przez układ nadajnika przy zastosowanym standardzie logicznym.



Rysunek 6.14: Symulacja diagramu oka dla badanego interfejsu przy transferze 500 Mb/s w zamodelowanym module Shuttler.

6.4. Podsumowanie badań

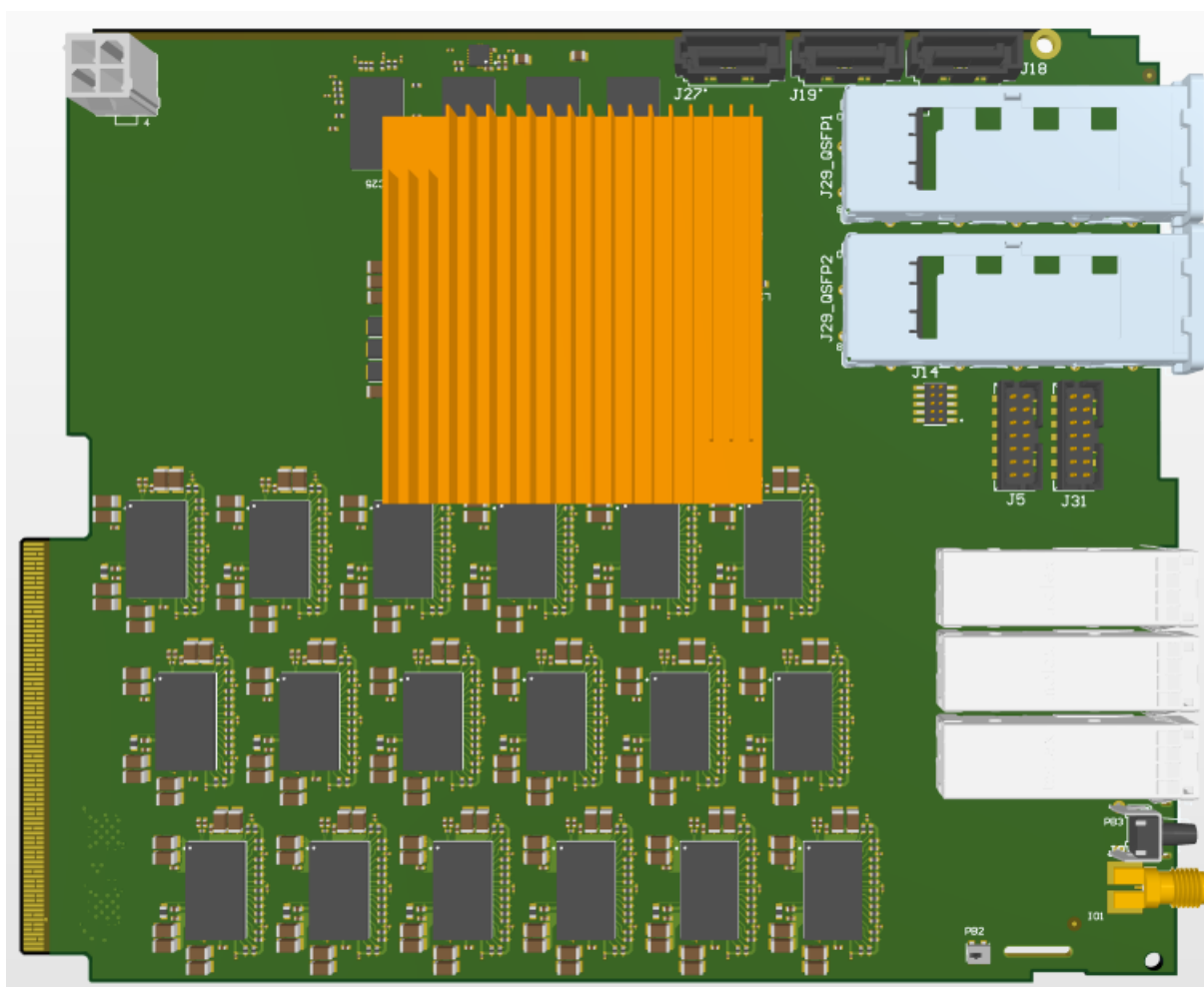
W trakcie badań dowiedziono, że pierwsze symulacje wykonane z wykorzystaniem modelu IBIS przetwornika LTC2000 dają wyniki niezgodne z rzeczywistym działaniem układu. Wykazano także, że za błąd w symulacji odpowiadają właśnie wspomniane modele. Dalsze pomiary udowodniły, że zaproponowana koncepcja przesyłu sygnału linią asymetryczną do odbiorników LVDS może znaleźć zastosowanie w docelowym module Shuttler i będzie z całą pewnością działać w wymaganym przez naukowców zakresie częstotliwości (rzędu 100 MHz). Wykazano także, że zamodelowanie układu wejściowego przetwornika za pomocą pojedynczego rezystora o wartości 120 Ω pozwala na uzyskanie wyników symulacji zbieżnych z rzeczywistymi. Dzięki temu możliwa była teoretyczna weryfikacja działania połączenia dla wyższych częstotliwości.

Na podstawie wniosków dotyczących wszystkich trzech zaproponowanych metod połączenia (przedstawionych w ramach podsumowania pierwszych symulacji w rozdziale 5.3) oraz wyników pomiarów podjęto decyzję o wykorzystaniu w finalnym projekcie modułu Shuttler metody sterowania odbiornika LVDS linią asymetryczną.

7. Finalny moduł Shuttler

7.1. Dobór układu FPGA

Symulacje oraz badania wykazały, że możliwe było podwojenie bądź potrojenie dostępnej liczby kanałów na pojedynczym module. Na tym etapie projektowym jedynym ograniczeniem okazała się wielkość dostępnej powierzchni na obwodzie drukowanym. Z tego względu postanowiono wykonać wstępne rozmieszczenie największych elementów, a więc: złączy, pamięci SDRAM, przetworników oraz układu FPGA (najmniejszego spośród rozważanych). Wynik przedstawiony został na rys. 7.1. Przedstawiony na nim został także wstępnie zaprojektowany obwód drukowany całego bloku przetwornika dla metody sterowania linią asymetryczną wraz z wymaganymi rezystorami i kondensatorami, gdyż miały one istotny wpływ na całkowitą zajmowaną powierzchnię.



Rysunek 7.1: Wstępne rozmieszczenie największych elementów modułu Shuttlet na płycie w formacie AMC przy wykorzystaniu FPGA XCKU040-1FFVA1156C.

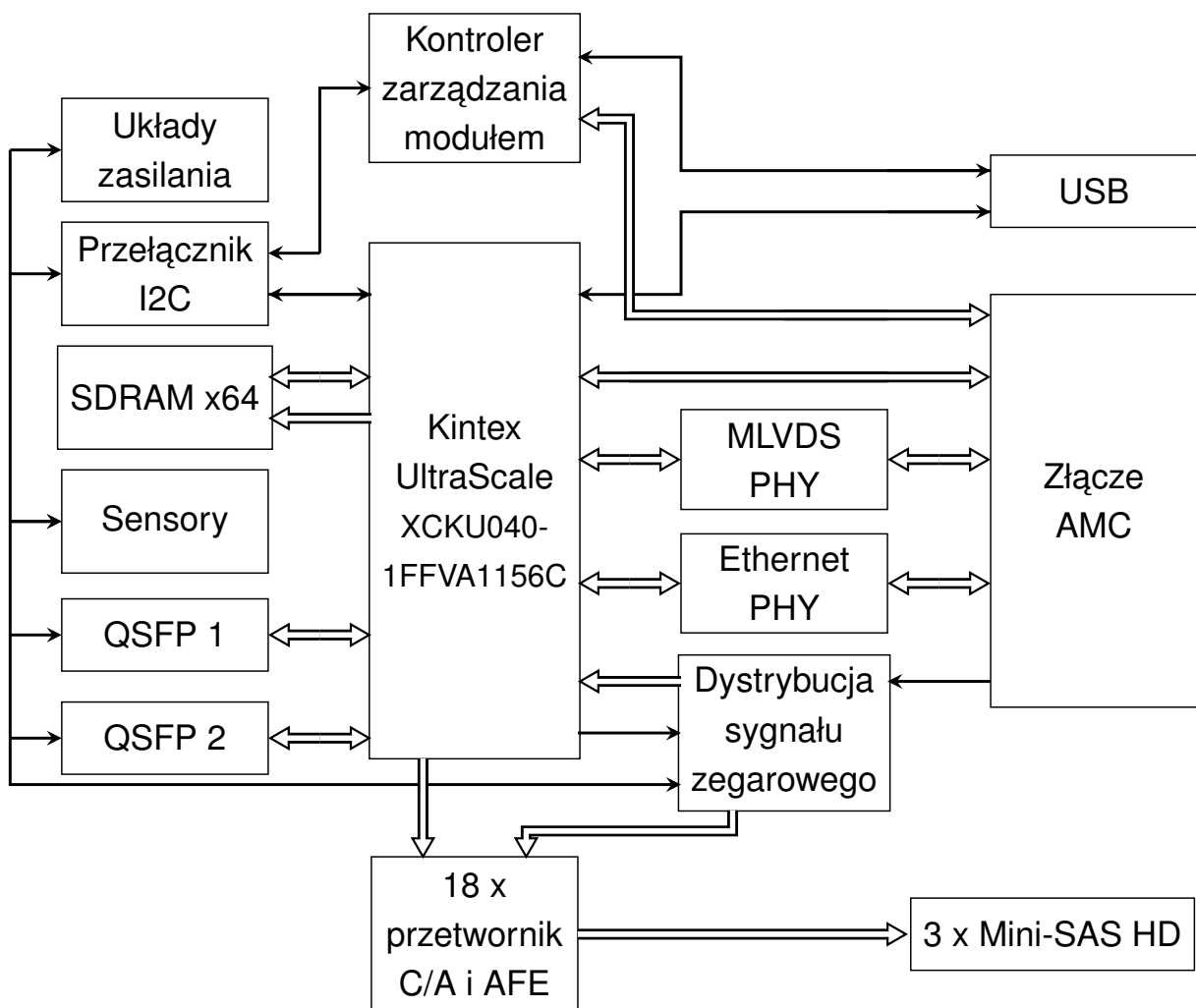
Rozmieszczone zostało 18 przetworników. Była to maksymalna wartość możliwa do uzyskania przy pełnej dbałości o jakość generowanego sygnału analogowego. Warstwa od spodu zajęta została przez torry analogowe. Przestrzeń pozostawiona w górnym lewym rogu przeznaczona była na układy zasilania. W końcowym projekcie być może fizycznie udałoby się dołożyć w tym miejscu dodatkowy przetwornik, jednak zdecydowanie nie byłoby to wskazane ze względu na bliską obecność

źródeł zakłóceń w postaci przełączanych cewek przetwornic. Z kolei część przestrzeni pozostawiona z prawej strony będzie posiadała ograniczone chłodzenie ze względu na blokowanie swobodnego przepływu powietrza przez klatki QSFP. Zatem potencjalnie umieszczone tam dodatkowe przetworniki obciążone byłyby większym błędem związanym z dużymi zmianami temperaturowymi.

Opracowana metoda pozwalała na uzyskanie maksymalnej liczby 18 kanałów dla dowolnego z rozważanych modeli układu FPGA. Wybrano zatem model XCKU040-1FFVA1156C ze względu na jego najniższą cenę. Dodatkowo był to układ preferowany przez fizyków uczestniczących w projekcie Sinara.

7.2. Realizacja poszczególnych bloków funkcjonalnych

Ogólny schemat blokowy modułu przedstawiony został na rys. 7.2. W porównaniu z jego pierwotną wersją sprecyzowaniu uległ model układu FPGA oraz liczba stosowanych przetworników C/A i złącz Mini-SAS HD potrzebnych do wyprowadzenia sygnałów analogowych na zewnątrz płyty. Schematy projektowe, których realizacja będzie omawiana w kolejnych punktach, zostały wykonane w programie Altium Designer i przedstawione w dodatku A.

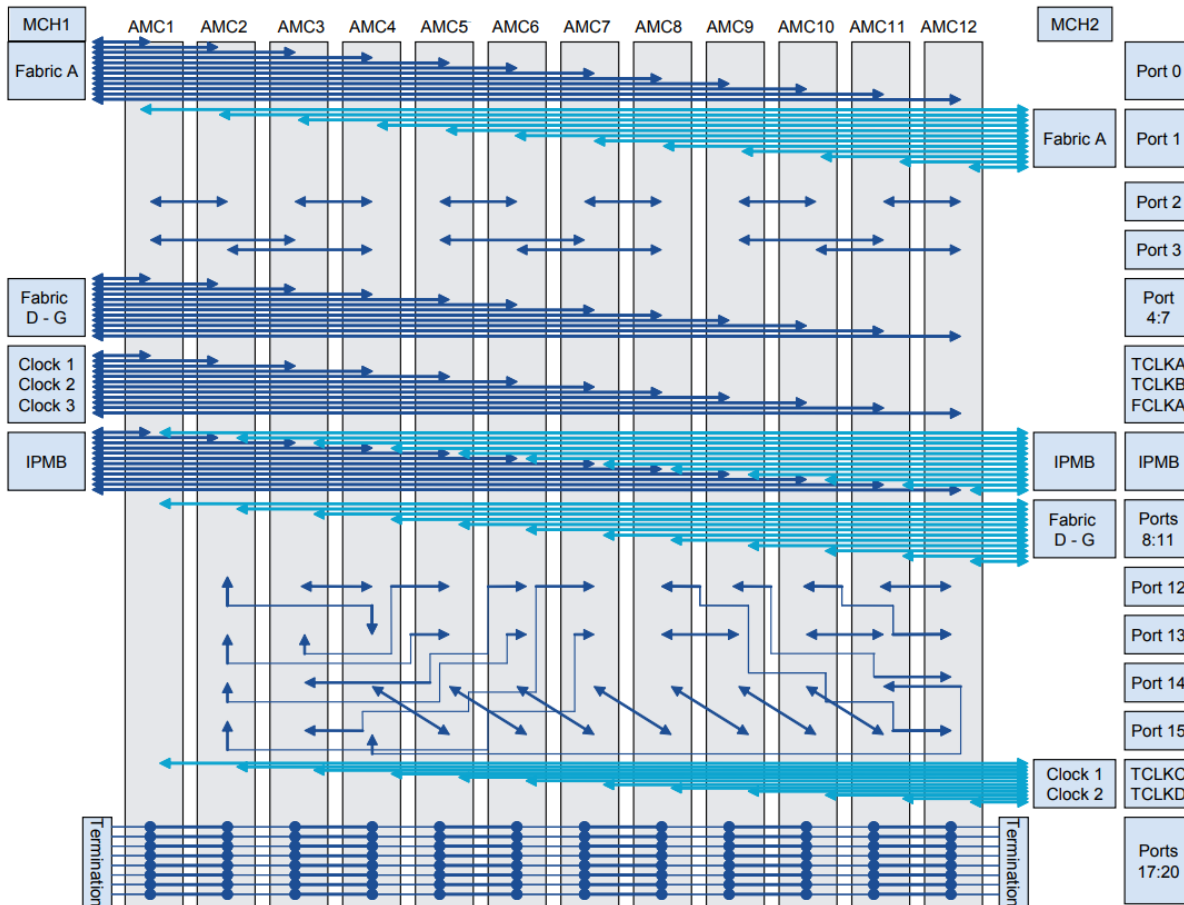


Rysunek 7.2: Finalny schemat blokowy modułu Shuttler AMC

7.2.1. Połączenia FPGA

Połączenia AMC

Połączenie do płyty tylnej kraty MicroTCA miało na celu nie tylko zapewnić niezbędną komunikację z modułem zarządzającym Metlino zgodnie z zapotrzebowaniem systemu Sinara, ale także gwarantować potencjalną możliwość komunikacji z pozostałymi podzespołami obecnymi w kracie MicroTCA, wykorzystując dostępne połączenia widoczne na rys. 7.3.



Rysunek 7.3: Połączenia na płycie tylnej kraty MicroTCA z 12 kartami AMC [19].

Ze względu na wykorzystanie standardu MicroTCA wszystkie moduły będące częścią projektu Sinara musiały posiadać identyczny schemat wykorzystania połączeń. I tak kolejno są to:

- port 0: Ethernet dostarczony z płyty Metlino,
- port 4-7: komunikacja DRTIO z płytą Metlino,
- port 12-15: komunikacja z sąsiednimi modułami typu punkt-punkt,
- porty 17-20: wspólna magistrala MLVDS dla wszystkich modułów AMC w kracie,
- sygnały zegarowe TCLKA oraz FCLKA z płyty Metlino – potencjalne źródło sygnału zegarowego dla układu dystrybucji sygnału zegarowego oraz gigabitowych nadajników i odbiorników,

- Intelligent Platform Management Bus (IPMB) – magistrala do komunikacji pomiędzy Metlino zarządzającym kratą oraz kontrolerem modułu AMC,
- niewidoczne na rysunku linie: JTAG umożliwiający zdalne programowanie modułu w kracie oraz linie adresowe informujące o pozycji zajmowanej wewnątrz kraty przez moduł AMC.

Kluczowym interfejsem służącym do przesyłania danych w obrębie całego systemu są linie DRTIO. Podłączone zostały bezpośrednio do nadajników i odbiorników gigabitowych GTH. Alternatywnym źródłem sygnału DRTIO są dostępne złącza QSFP.

Drugą metodą przesyłu danych do Shuttlera jest wykorzystanie sieci Ethernet. W domyślnym systemie połączenie zapewniać będzie port 0 złącza krawędziowego AMC. Ze względu jednak na możliwość utworzenia innych konfiguracji blok Ethernetu został opisany w osobnym podrozdziale.

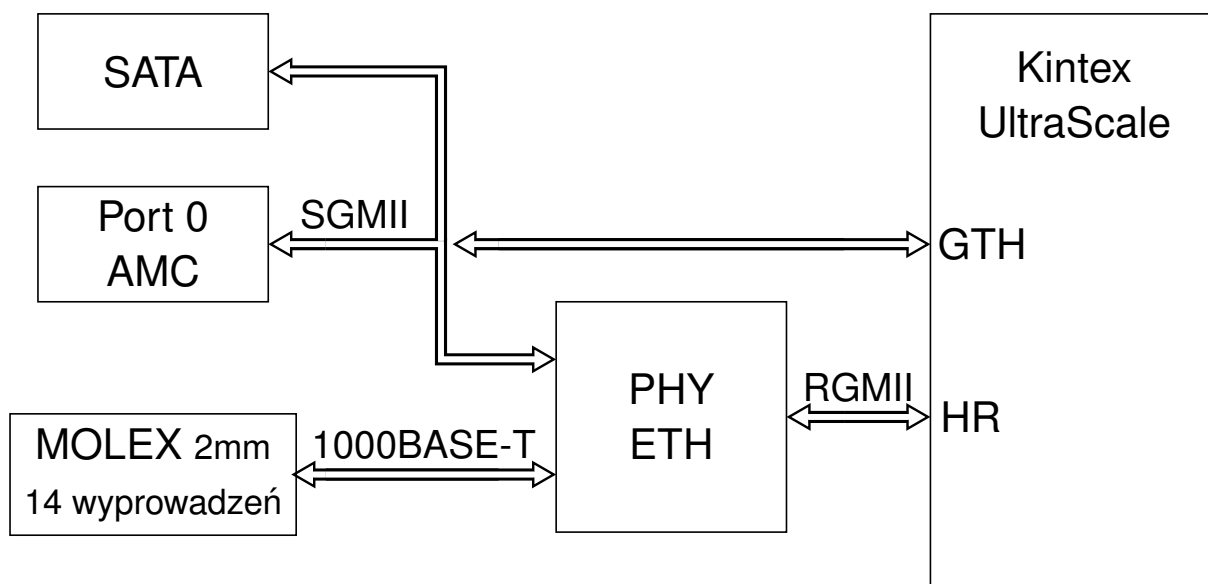
Wspólna magistrala MLVDS wymagała użycia dodatkowych układów, ponieważ nie jest to standard obsługiwany przez układy Kintex UltraScale. MLVDS wymaga, aby dołączone układy odbiorcze znajdowały się jak najbliżej głównej pary linii różnicowej. W przypadku MicroTCA linie te prowadzone są przez płytę tylną kraty. Zatem każdy fragment ścieżki poprowadzony na module AMC powoduje zwiększenie zaburzenia ciągłości impedancji. Konieczne było odebranie i nadawanie sygnału w jak najbliższych odległościach od złącza. Wykorzystano dwa nadajniki-odbiorniki MLVDS SN65MLVD040RGZT. Ich wybór podyktowany był powszechnym użyciem we wszystkich modułach AMC projektu Sinara. Układy te do obsługi pojedynczej magistrali wymagają dwóch linii asymetrycznych – linii danych oraz kierunku przepływu danych. Podłączone zostały do wolnych wyprowadzeń w bankach obsługujących pamięć SDRAM. Ponieważ bank ten zasilany jest napięciem 1,5 V, konieczne było zastosowanie dodatkowo przetwornicy napięciowej TXB0108PW (SN65MLVD040RGZT pracuje z logiką 3,3 V, zaś wolny bank HR nie posiadał już wolnych wyprowadzeń).

Porty 17-20, gwarantujące komunikację typu punkt-punkt z 4 najbliższymi modułami AMC, zostały podłączone do nadajników i odbiorników GTH. Pozostałe trzy wolne porty GTH podłączone zostały do złącz Serial Advanced Technology Attachment (SATA) – umożliwiając alternatywną komunikację w przypadku pracy karty poza kratą MicroTCA lub przy braku modułu Metlino.

Ethernet

Jak już zostało wspomniane, opcjonalnym interfejsem komunikacyjnym jest Ethernet. Schemat połączeń zestawionych na module Shuttler przedstawiony został na rys. 7.4. Pierwotną koncepcją podłączenia układu FPGA było wykorzystanie SGMII, wyprowadzonego z portu 0 złącza krawędziowego AMC lub złącza SATA (dla pracy poza kratą) i doprowadzenie go do nadajników i odbiorników gigabitowych GTH. To fizycznie najprostsze rozwiązanie generowało jednak pewne problemy przy powstawaniu systemu ARTIQ. Z tego powodu wszystkie moduły AMC projektowane w ramach projektu Sinara wyposażane są w dodatkowy układ 88E1512-XX-NNP2I000, zapewniający obsługę sprzętowej warstwy PHY Ethernet.

Szczególnie istotną cechą 88E1512-XX-NNP2I000 jest możliwość pracy w trybie SGMII-RGMII, który pozwolił na zmianę problematycznego interfejsu. Konfiguracja ta jest obecnie domyślna dla wszystkich powstałych modułów AMC. Zapewniona została możliwość zestawienia pierwotnego bezpośredniego połączenia poprzez odpowiednie przelutowanie kondensatorów zmieniających ścieżkę dalszej propagacji sygnału. Sam RGMII obsługiwany jest w układzie FPGA przez bank wyprowa-



Rysunek 7.4: Schemat blokowy połączeń interfejsu Ethernet.

dzeń typu HR ze względu na konieczność stosowania standardu logicznego LVCMOS25 lub LVCMOS33.

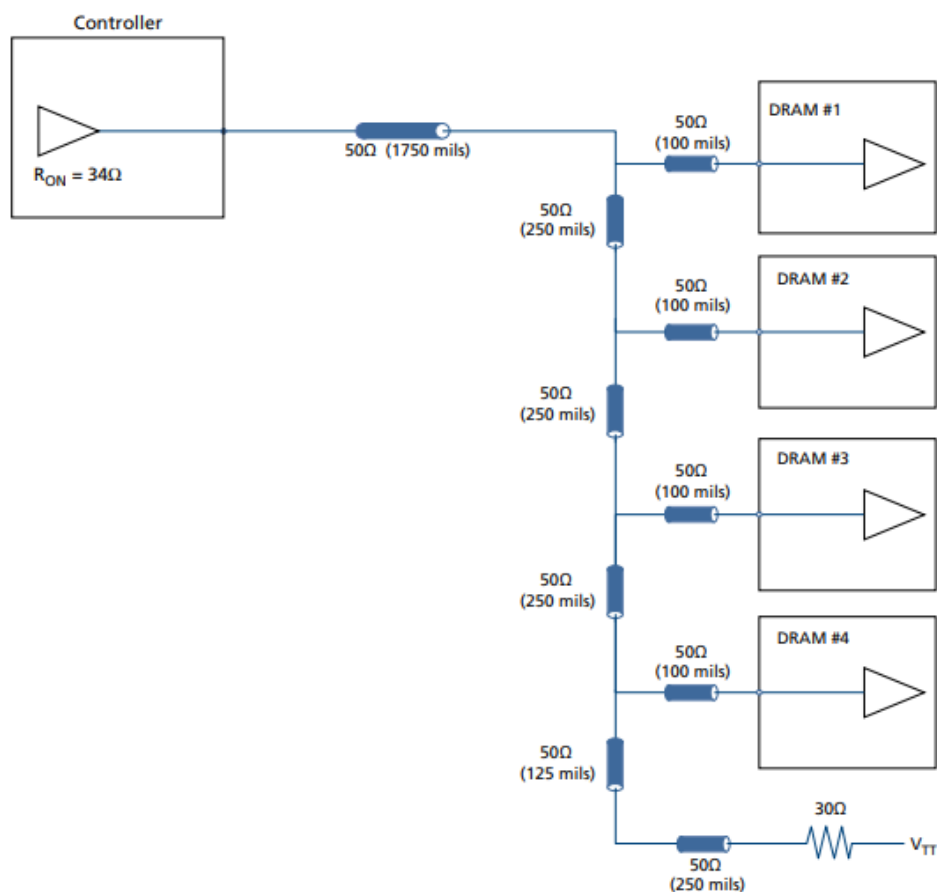
Do układu 88E1512-XX-NNP2I000 podłączono także złącze umożliwiające ustanowienie zewnętrznego połączenia w trybie 1000BASE-T w celu ułatwienia testowania modułów oraz nowych funkcjonalności. Jeśli na dalszym etapie rozwoju części programowej projektu Sinara oraz ARTIQ zostanie osiągnięte rozwiązanie gwarantujące bezproblemową pracę dla bezpośredniego SGMII, to układ 88E1512-XX-NNP2I000 automatycznie przestanie być montowany w nowo produkowanych płytach, a przy ewentualnej kolejnej rewizji – zostanie na stałe usunięty z projektu.

Moduł Shuttler posiada dwa złącza QSFP, których wyprowadzenia połączone zostały z nadajnikami i odbiornikami GTH – zatem także mogą być wykorzystane przez użytkownika w celu podłączenia płyty do sieci Ethernet.

SDRAM

Zgodnie z wymaganiami do przeprowadzania obliczeń konieczne było wyposażenie układu FPGA w pamięć SDRAM DDR3. Zastosowano cztery układy MT41K256M16TW-107:P kompatybilne z rozwiązaniem z płyty Sayma. Gwarantują one łącznie 16 Gb pamięci oraz transfer danych 1866 MT/s. W teorii są to układy DDR3L, a więc pracujące z napięciem 1,35 V. Dostosowane są one jednak także do wartości zasilania 1,5 V. Ponieważ stosowane odpowiedniki na wcześniejszych rewizjach innych podsystemów Sinara stanowiły standardowe rozwiązanie DDR3 dla napięcia 1,5 V, wszystkie obecnie stosowane układy pamięci zasilane są taką samą wartością napięcia dla zachowania jednolitości rozwiązań – także na module Shuttler.

Linie adresowe oraz kontrolne są współdzielone przez wszystkie urządzenia pamięci. W kwestii podłączenia układów zdecydowano się na zastosowanie architektury typu „fly-by”, której przykład przedstawiony został na rys. 7.5. Taka konfiguracja wymaga terminacji na końcu linii, za wszystkimi układami, do napięcia terminującego o wartości równej połowie wartości napięcia zasilania. Jest ono generowane przez regulator TPS51200DRCT – identyczny zastosowany został do generacji napię-



Rysunek 7.5: Przykład architektury Fly-By dla linii adresowych i kontrolnych [26]

cia terminującego dla linii w połączeniu między Kintex UltraScale a przetwornikami C/A. Pozostałe linie podłączone punkt-punkt wykorzystują wewnętrzne terminacje układów, które są przełączane w zależności od kierunku transmisji. Nie wymagały one zatem żadnych dodatkowych rezystorów.

Konieczne było także zapewnienie prawidłowego podłączenia linii do wyprowadzeń układu FPGA zgodnie z wymaganiami producenta [41]. Wyprowadzenia w każdym banku podzielone są na grupy po 13 wyprowadzeń każda. Wewnątrz nich każde wyprowadzenie ma swoje własne oznaczenia N0 do N12. I tak linie DQS musiały być podłączone w danej grupie do wyprowadzeń N6 oraz N7. Odpowiadające danemu DQS linie danych DQ podłączono w tej samej grupie do dowolnych wyprowadzeń z wyjątkiem N1 oraz N12. DM z kolei, także umieszczone w odpowiadającej grupie, przypisano do wyprowadzenia N0. Wszystkie linie kontrolne musiały być zawarte w jednym banku (cała pamięć obsługiwana jest przez 3 banki wyprowadzeń).

7.2.2. Blok MMC

Blok MMC to przede wszystkim mikrokontroler LPC1776FET180,551 zarządzający modułem w kra- cie zgodnie ze standardem MicroTCA [2]. Ponieważ jego funkcjonalność jest ta sama niezależnie od przeznaczenia samego modułu, schemat połączeń prawie w całości został zaimportowany z wcze- śniejszych projektów wewnątrz systemu Sinara – ze względu na fakt posiadania gotowych kodów przeznaczonych właśnie dla tego procesora i danych połączeń. MMC komunikuje się z modułem MCH zarządzającym całą kratą w celu otrzymania głównego napięcia 12 V (samemu będąc zasilo-

nym z dedykowanego napięcia 3,3 V, doprowadzonego przez płytę tylną kraty MicroTCA). Po włożeniu karty do systemu użytkownik jest informowany o jej zmieniającym się statusie wewnątrz kraty poprzez niebieski Light Emitting Diode (LED). Mikrokontroler sprawdza także temperaturę poprzez komunikację z zastosowanymi czujnikami: 4x LM75AImm/NOPB (temperatury w różnych miejscach obwodu drukowanego) oraz MAX6642ATT90+T (temperatura FPGA) i zapisuje dane diagnostyczne w pamięci Electrically Erasable Programmable Read-Only Memory (EEPROM) AT24MAC402-XHM. Do komunikacji z układem FPGA zapewniono dedykowany SPI.

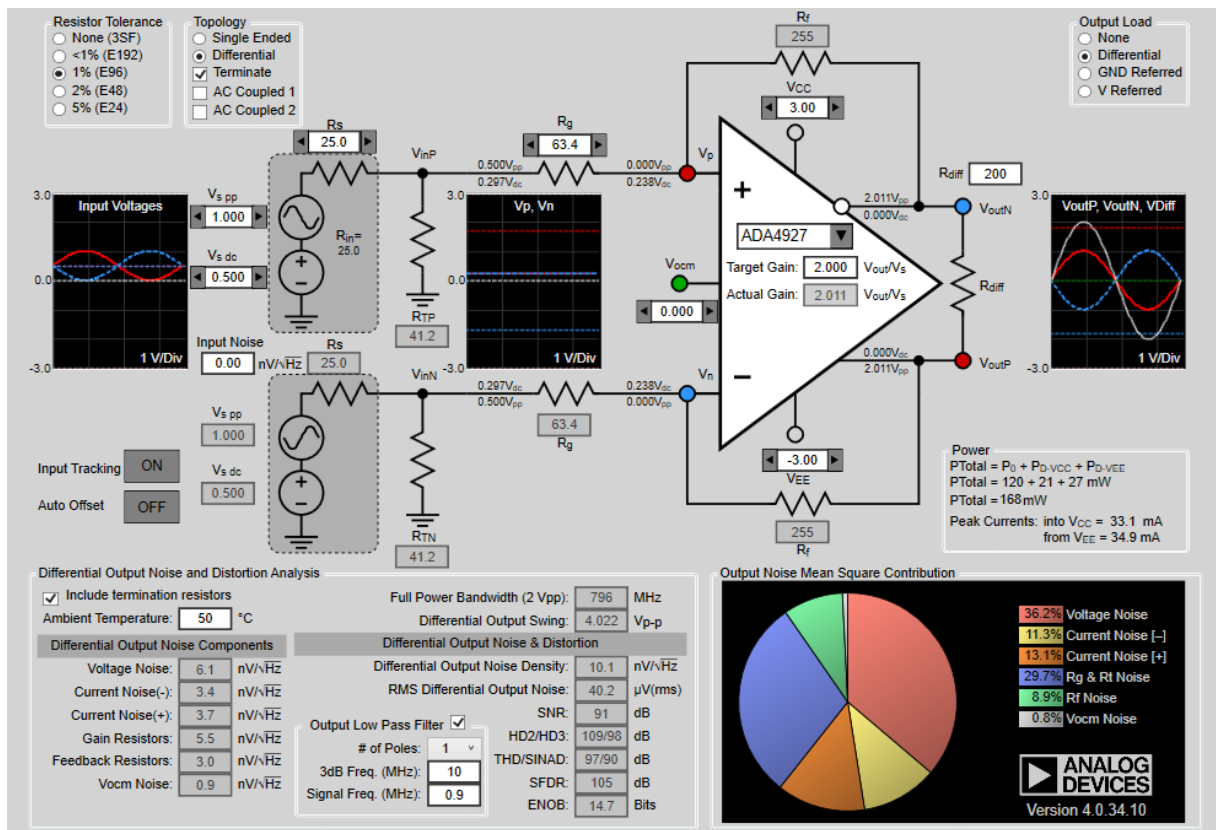
Mikrokontroler odpowiada za uruchomienie każdej przetwornicy obecnej na module oraz za kontrolę ich wyprowadzeń, sygnalizujących osiągnięcie odpowiedniego napięcia wyjściowego. Umożliwia to implementację kontrolowanej sekwencji uruchamiania poszczególnych poziomów napięciowych.

7.2.3. Przetworniki C/A oraz tor analogowy

Linie danych oraz linia sygnału zegarowego dla danych pomiędzy FPGA a przetwornikami C/A zostały podłączone zgodnie ze zbadaną metodą sterowania odbiorników LVDS linią asymetryczną. Niewykorzystane wyprowadzenia portu wejściowego A układów zostały zgodnie z zaleceniem producenta zwarte do linii masy modułu. Zastosowano także kondensatory odprężające po jednym na każde 4 rezystory terminujące. Zapewnia to stabilność napięcia terminującego. Jest to powszechna praktyka w terminacji linii SDRAM, bazującej na podobnym schemacie.

Każdy przetwornik LTC2000 do rozpoczęcia pracy wymaga skonfigurowania poprzez interfejs SPI. Dla 18 układów wymagałoby to wykorzystania dużej liczby wyprowadzeń, które były niedostępne w układzie FPGA. Zdecydowano zatem dołączyć te interfejsy do głównego interfejsu Inter-Integrated Circuit (I^2C). Wykorzystano w tym celu układy SC18IS602B, będące konwerterami I^2C na SPI. Każdy może obsługiwać do 4 urządzeń podrzędnych. Ponieważ nie pozwalają one na translacje poziomów napięciowych, konieczne było podłączenie linii zasilającej obwody SPI przetworników do napięcia o wartości 3,3 V.

Schemat toru analogowego dla sygnału wyjściowego przetwornika LTC2000 został zaimplementowany zgodnie z wytycznymi naukowców zamawiających moduł Shuttler. Podstawowym zadaniem było wzmocnienie sygnału różnicowego do amplitudy o wartości 2 V z wykorzystaniem niskoszumnego wzmacniacza różnicowego ADA4927. W celu doboru odpowiednich wartości rezystorów wykorzystano oprogramowanie ADI DiffAmpCalc, dostarczone przez producenta tego układu scalonego [13] – otrzymane wyniki przedstawiono w formie zrzutu ekranu programu na rys. 7.6. Konfiguracja sygnału wejściowego dla wzmacniacza została opracowana w programie z uwzględnieniem kilku czynników. Po pierwsze producent LTC2000 zastrzega, aby wartość napięcia międzyszczytowego różnicowego sygnału wyjściowego przetwornika nie przekraczała 2 V. Wyższe dodatnie napięcia zaburzyłyby liniowość przetwornika, zaś niższe ujemne są ograniczane przez diody zabezpieczające. Po drugie zalecane jest także stosowanie niskich wartości amplitudy oraz składowych stałych napięć w celu uzyskania jak najmniejszych zniekształceń. W samej dokumentacji LTC2000 widoczne jest, że większość parametrów przetwornika podana została dla wartości obciążenia $12,5 \Omega$ na każdym z wyjść, co przy konfiguracji maksymalnego prądu sterującego 40 mA (na obydwu wyprowadzenia) daje maksymalną wartość międzyszczytową wyjściowego napięcia różnicowego 1 V. Takie same warunki były oczekiwane przez fizyków. Dlatego przy wyprowadzeniach zastosowano rezystory po 50Ω podłączone do masy, które wraz z identyczną wewnętrzną rezystancją obciążającą dały wartość



Rysunek 7.6: Zrzut ekranu programu ADI DiffAmpCalc dla skonfigurowanego toru analogowego

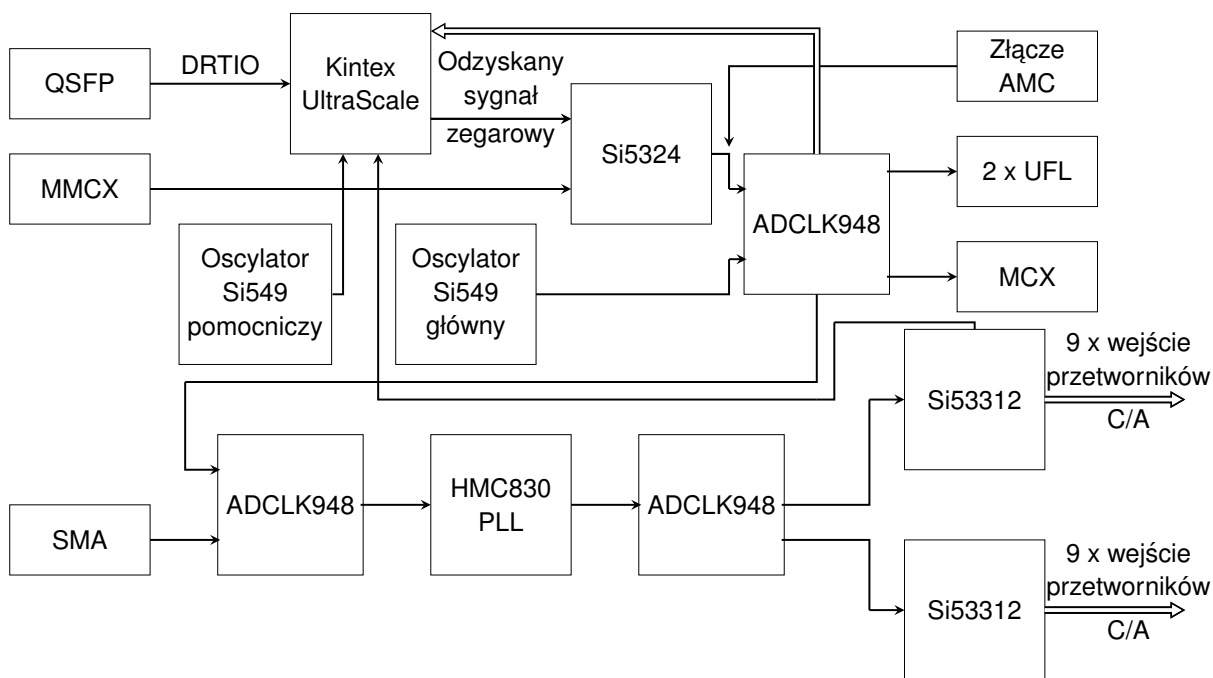
równoległego połączenia 25Ω . Takie źródła prądowe (którym są wyjścia przetwornika) o prądzie maksymalnym 40 mA przekształcono na źródła napięciowe o wartości międzyszczytowej napięcia 1 V , składowej stałej $0,5 \text{ V}$ i rezystancji szeregowej 25Ω . Program dopasował impedancję wejściową toru analogowego, dzięki czemu otrzymano oczekiwane obciążenie całkowite $12,5 \Omega$.

Sygnal analogowy ze wzmacniacza skierowany został do jednego z trzech złączy Mini-SAS HD, gwarantujących ekranowanie każdej pary różnicowej. Dalsze przetwarzanie analogowe następować będzie w bezpośrednim sąsiedztwie pułapki – wzmocnienie, filtracja i konwersja sygnału różnicowego na sygnał asymetryczny. Konieczne do uzyskania parametry mogą znacząco różnić się w zależności od stosowanej pułapki jonowej, dlatego projekt tej części przetwarzania musi być zaprojektowany bezpośrednio przez fizyków przeprowadzających dany eksperyment.

7.2.4. Dystrybucja sygnału zegarowego

Jedną z podstawowych cech projektu Sinara jest możliwość synchronizacji każdego modułu z dokładnością rzędu 1 ns , wykorzystując autorski protokół DRTIO. Jednak samo odebranie sygnału zegarowego przez układ FPGA nie gwarantuje zachowania należytej precyzji. Konieczne było zastosowanie układu tłumiącego jitter. Schemat blokowy układu dystrybucji sygnału zegarowego przedstawiony został na rys. 7.7.

We wszystkich modułach składających się na projekt Sinara do tłumienia jitteru stosowany jest układ Si5324. Posiada on dwa wyjścia w standardzie Low-Voltage Positive Emitter-Coupled Logic (LVPECL), jednak wykorzystane zostało wyłącznie jedno z nich. Wiązało się to z problemem od-



Rysunek 7.7: Schemat bloku dystrybucji sygnału zegarowego w module Shuttler.

krytym w trakcie opracowywania wsadu dla FPGA na płycie Sayma. Zauważono, że układ Si5324 przy uruchomieniu lub po resecie generuje niedeterministyczną latencję, niezależnie dla obydwu wyjść, jednak o stałej wartości w trakcie nieprzerwanego działania. Biorąc pod uwagę stawiane wszystkim modułom wymagania synchronizacyjne, była to sytuacja niedopuszczalna. W praktyce postanowiono już wtedy na niezmiennianie modelu układu, ponieważ pozostałe jego parametry były satysfakcjonujące. Problem ten rozwiązano w pętli pomiędzy układem FPGA a Si5324 – Kintex UltraScale, wykorzystując wewnętrzną Phase Locked Loop (PLL), dostosowującą fazę odzyskanego sygnału zegarowego, aby w odpowiedni sposób przesunąć fazę sygnału zegarowego wychodzącego z układu tłumiącego jitter [21]. Ponieważ jednak niemożliwe było wyrównanie opóźnień dla obydwu wyjść, postanowiono pozostawić jedno z nich niewykorzystane. Takie samo rozwiązanie sprzętowe zostało zastosowane w projekcie modułu Shuttler.

Wspomniane dostosowywanie fazy powodowało jednak niewielką degradację jej stabilności. Z tego powodu dodano alternatywny sposób synchronizacji, będący nieznacznie zmienioną formą rozwiązania projektu White Rabbit Project (WR). Metoda ta bazuje na wykorzystaniu dwóch identycznych precyzyjnych oscylatorów o programowalnej częstotliwości. Faza odzyskanego sygnału zegarowego jest porównywana z fazą pochodząca z oscylatora głównego (skonfigurowanego na tę samą częstotliwość) poprzez ich przepuszczenie przez przerzutniki D, taktowane sygnałem pochodzącym z oscylatora pomocniczego (o minimalnie zmienionej częstotliwości)[28]. W ten sposób różnica fazy jest cyfrowo rozciągana w czasie i możliwe jest jej precyzyjne wyliczenie poprzez zliczanie taktów sygnału zegarowego, w których przerzutniki dawały różne wartości sygnałów wyjściowych. Ze względu na znane problemy w układach Kintex UltraScale zastosowano fizyczne układy przerzutników, a nie ich implementację w FPGA. To rozwiązanie, opracowane przez projektantów dla wcześniejszych modułów projektu Sinara [21], zostało wykorzystane także na płycie Shuttler. Zmianie uległy jedynie elementy odpowiadające za dostosowanie poziomów logicznych oscylatorów i przerzutników do odpowiednich banków wyprowadzeń FPGA. Do czasu zakończenia badań nad osiąganymi poszczególnych sposobów synchronizacji, obydwa z nich są implementowane na wszyst-

kich powstających modułach. W późniejszym etapie rozwiązanie dające gorsze wyniki nie będzie montowane, a w przypadku powstania nowych wersji modułów zostanie także usunięte z projektu obwodu drukowanego.

Podstawowym źródłem sygnału zegarowego w obrębie całego modułu Shuttleer może być:

- sygnał odzyskany ze strumienia danych DRTIO przez FPGA – dostarczony przez płytę tylną kraty MicroTCA lub z zewnątrz przez złącze QSFP bądź,
- zewnętrzne różnicowe źródło sygnału zegarowego dołączone do dwóch złącz Micro-miniature Coaxial (MMCX),
- sygnał zegarowy generowany przez główny oscylator Si549,
- sygnał zegarowy pochodzący z modułu MCH rozproszony poprzez płytę tylną do złącza krawędziowego AMC. Jest to jedyny sygnał, do którego użycia wymagane jest wprowadzenie sprzętowych zmian w postaci przelutowania kondensatorów zapewniających sprzężenie Alternating Current (AC). Jego zastosowanie uniemożliwia wykorzystanie sygnału pochodzącego z układu SI5324.

Dwa pierwsze źródła podłączone zostały do wspomnianego układu tłumiącego jitter. W dalszej kolejności sygnał zegarowy dystrybuowany jest przez układy ADCLK948, które umożliwiają wybór jednego z dwóch źródeł sygnału i przekazują go do 8 równoległych linii różnicowych w standardzie LVPECL. Wybór tego układu także podyktowany był wykorzystywaniem go w projekcie Sinara. W pierwszej kolejności zsynchronizowany z resztą systemu sygnał przekazany zostaje do układu FPGA – do banków GTH, będących gigabitowymi nadajnikami/odbiornikami, banku obsługującego SDRAM i sterowanie innymi podzespołami. Umożliwiono także jego wyprowadzenie na zewnątrz modułu jako sygnał różnicowy poprzez 2 złącza U.FL bądź w linii asymetrycznej poprzez złącze Micro Coaxial Connector (MCX). Dalsze obwody dystrybucji sygnału zegarowego związane są z bezpośrednim sterowaniem przetworników cyfrowo-analogowych. Pierwszy układ rozdzielający sygnał umożliwia dobór sygnału źródłowego, którym będzie wcześniej omawiany sygnał, wykorzystywany przez resztę układów na płycie lub zewnętrzny sygnał dostarczony przez użytkownika na złącze SMA. Drugie rozwiązanie umożliwia prostą konfigurację modułu do pracy w eksperymentach, gdzie nie będzie wymagana tak precyzyjna synchronizacja z resztą systemu.

Układ HMC830 z zewnętrzną PLL generuje sygnał zegarowy o docelowej częstotliwości taktującej przetworniki. Jest to kolejny układ powszechnie wykorzystywany w projekcie Sinara. Gotowa jest zatem cała część programowa do jego obsługi. Konieczne było odpowiednie zaprojektowanie pętli fazowej. W tym celu wykorzystano oprogramowanie udostępnione przez producenta ADI: simPLL [14], pozwalające w prosty sposób obliczyć wartości elementów składowych filtra pętli. Sygnał następnie przez kolejne układy rozdzielające jest kierowany do wszystkich przetworników oraz do układu FPGA.

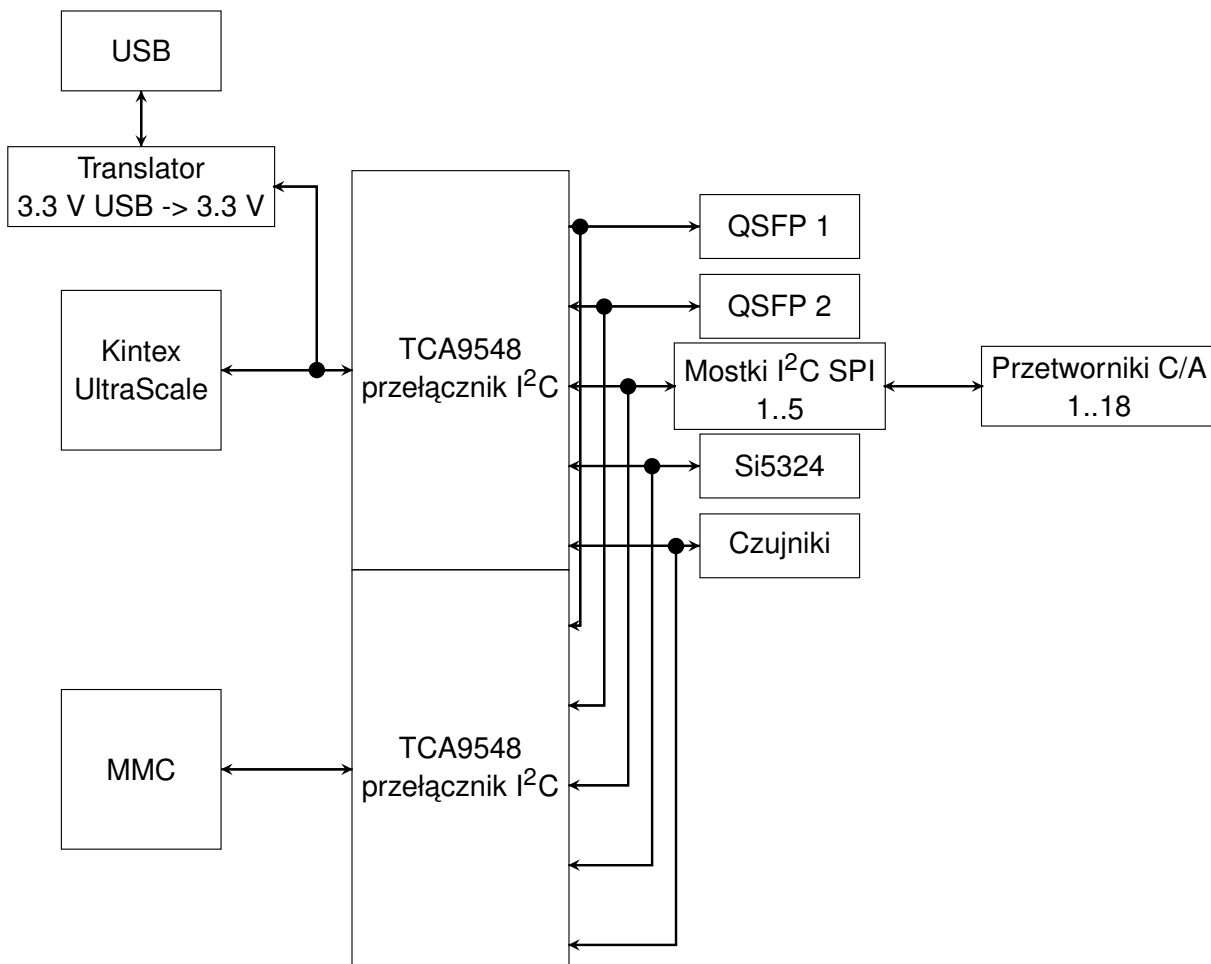
7.2.5. USB

Zgodnie ze stosowaną we wszystkich modułach projektu Sinara praktyką, także i w tym wypadku umożliwiono podłączenie komputera użytkownika do modułu poprzez USB. Wykorzystany układ FT4232H pozwolił na zestawienie czterech niezależnych 8-bitowych magistral szeregowych lub równoległych. Implementacja konkretnych interfejsów była podyktowana opracowanymi w zespole skryptami obsługującymi inne moduły. Była to możliwość programowania układów poprzez JTAG,

kanały diagnostyczne dla FPGA i mikrokontrolera wykorzystujące UART oraz możliwość komunikacji z urządzeniami diagnostycznymi i konfigurowania podrzędnych układów scalonych wykorzystując interfejs I²C.

7.2.6. Magistrala I²C

Główną magistralą komunikacyjną, wykorzystywaną do konfiguracji i odczytów danych przez układy nadrzędne (mikrokontroler oraz FPGA) została magistrala I²C. Zastosowano dwa przełączniki I²C TCA9548ARGER w celu odseparowania urządzeń zasilanych z generowanego na module napięcia o wartości 3,3 V od tych, uruchamianych poprzez zasilanie 3,3 V dedykowane dla MMC (np. czujniki temperatury). Pozwoliło to także na znaczące uproszczenie prowadzonych linii w obwodzie drukowanym. Same przełączniki łączą urządzenia nadrzędne z dowolnymi z 8 kanałów urządzeń podrzędnych. Pełny schemat połączeń przedstawiony został na rys. 7.8.



Rysunek 7.8: Schemat połączeń magistrali I²C na płycie Shuttler.

Zastosowano dwa takie układy, każdy dedykowany dla innego urządzenia nadrzędnego: FPGA oraz mikrokontrolera (są zasilane z oddzielnych źródeł 3,3 V). Równolegle do linii obsługiwanych przez FPGA dołączono sygnał pochodzący z USB. Będzie on wykorzystywany wyłącznie na wczesnym etapie uruchamiania prototypów, aby w możliwie najprostszy sposób zweryfikować komunika-

cję ze wszystkimi urządzeniami podrzędnymi – docelowo nie będzie wykorzystywany w działającym systemie. Kolejne kanały gwarantują:

- połączenia z pamięcią EEPROM i układem zarządzania modułów QSFP,
- możliwość konfiguracji przetworników C/A (pośrednio przez konwertery I²C na SPI),
- możliwość konfiguracji układu Si5324,
- możliwość konfiguracji alternatywnego czterokanałowego modułu zasilającego EXAR,
- odczyt danych z diagnostycznych czujników temperaturowych.

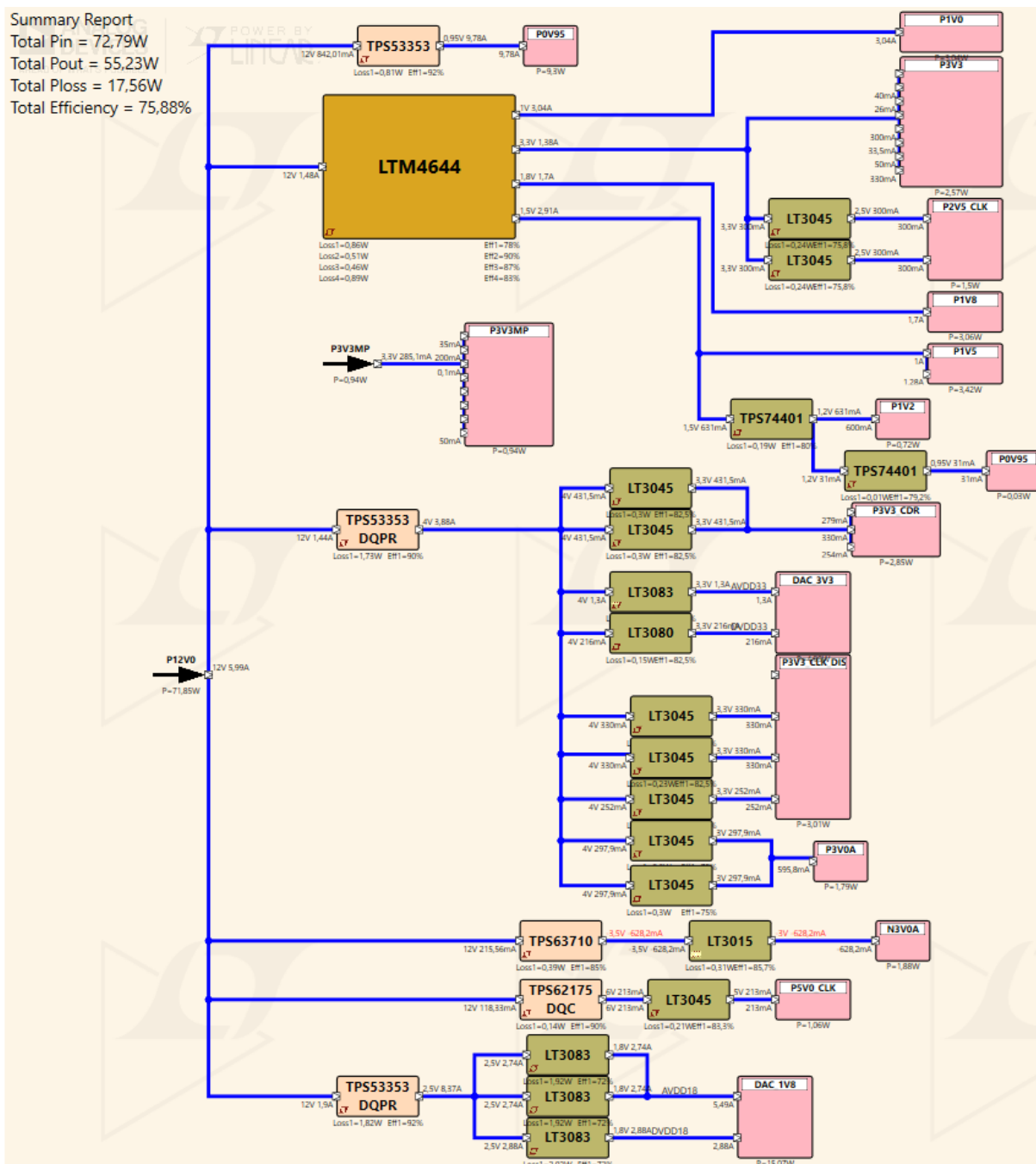
7.2.7. Układy zasilania

W celu poprawnego zaprojektowania układów zasilania konieczne było określenie zapotrzebowania prądowego wszystkich obecnych układów w najgorszym możliwym przypadku (np. przy maksymalnej częstotliwości dla układów sygnałów zegarowych). Posłużono się do tego dokumentacjami technicznymi, a dla układu Kintex UltraScale wykorzystano arkusz wspomagający oszacowanie poboru mocy FPGA udostępniony przez producenta [43]. Układy podzielono w zależności od wymaganego poziomu napięcia, zaś w tych grupach dokonano kolejnego podziału na układy zwykle (zasilane bezpośrednio z przetwornicy obniżającej napięcie z poziomu 12 V, będącego głównym zasilaniem dostarczonym do modułu AMC) oraz układy wrażliwe (których źródło napięcia powinno być generowane przez stabilizatory napięcia typu Low Dropout (LDO), cechujące się dużo niższym poziomem szumów względem przetwornic obniżających napięcie, kosztem większych strat mocy).

Przy przeprowadzaniu obliczeń posłużono się oprogramowaniem firmy Analog Devices LTpowerCAD [12]. Stanowi ono zestaw narzędzi pozwalających np. na skonfigurowanie przetwornicy obniżającej napięcie poprzez dobór odpowiednich wartości elementów, a także zaplanowanie schematu zasilania dla całego modułu. Po wprowadzeniu danych do poszczególnych bloków symbolizujących zasilane układy oraz ich połączeniu otrzymano gotowy diagram przepływu prądu. Następnie program umożliwiał automatyczne obliczenie zapotrzebowania prądowego na poszczególnych węzłach z uwzględnieniem strat mocy w przetwornicach i regulatorach LDO. Przy obliczaniu najgorszego wariantu przyjęto, że przetworniki C/A pracować będą z częstotliwością próbkowania 1 GS/s. Była to wartość dużo większa od tej, która zostanie zastosowana przy pierwszych eksperymentach (będzie ona wynosiła 50-200 MS/s). Dysponując jednak tak szybkimi przetwornikami, projektowanie płyty dla relatywnie niskich częstotliwości byłoby marnotrawieniem jej potencjału.

Widok finalnego diagramu dla modułu Shuttler wraz z obliczeniami pobieranej mocy przedstawiony został na rys. 7.9. Wynik w najbardziej pesymistycznym przypadku wyniósł 72,79 W. Zgodnie ze standardem MicroTCA [2] przypadająca na pojedynczy moduł AMC+RTM moc dostarczona wraz zasilaniem +12 V wynosi 80 W, zatem projekt spełnił wymagania. W tabeli 7.1 przedstawiono otrzymane wyniki wymaganej wydajności prądowej dla poszczególnych przetwornic. Warto podkreślić, że w praktyce moduł będzie wymagać mniejszej mocy, ponieważ dane do obliczeń były brane na podstawie deklarowanej w dokumentacjach maksymalnej możliwej wartości poboru prądu poszczególnych urządzeń, nie zaś wartości typowej. Dodatkowo stosowanie niższej częstotliwości próbkowania niż 1 GS/s także wpłynie na spadek wymaganej mocy.

Wprowadzona została możliwość dostarczenia zasilania dla całego modułu z zewnątrz, podłączając napięcie 12 V. Pozwala to na uruchomienie płyty Shuttler jako niezależnego urządzenia poza



Rysunek 7.9: Zrzut ekranu programu LTpowerCad dla kompletnego systemu zasilania płyty Shuttler.

kratą MicroTCA oraz ułatwia jego testowanie. Zasilanie dla banków wyprowadzeń FPGA, przesyłających sygnał do przetworników C/A, doprowadzone zostało poprzez koralek ferrytowy, dzięki czemu po jego wylutowaniu będzie istniała opcja doprowadzenia zasilania o innej wartości. Rozwiązanie to ułatwi w przyszłości dokonanie dalszych badań, mających na celu wyznaczenie optymalnego standardu logicznego do sterowania odbiorników LTC2000, gwarantującego najmniejsze straty mocy przy niezawodnym działaniu samych przetworników. Uniknięto w ten sposób konieczności projektowania i ponoszenia kosztów produkcji dedykowanego modułu badawczego.

Zaimplementowana została opcja wykorzystania alternatywnego czterokanałowego układu zasilającego, generującego napięcia dla układu FPGA. We wcześniejszych modułach projektu Sinara

Tabela 7.1: Wykorzystane przetwornice wraz z parametrami oraz generowane przez nie napięcia

Układ	Maksymalna wydajność prądowa [A]	Generowane napięcie [V]	Wymagana wydajność prądowa [A]	Zasilane układy
LTM4644	4	1	3,04	FPGA
	4	1,5	2,97	FPGA / SDRAM
	4	1,8	1,7	FPGA
	4	3,3	1,38	FPGA / mostki I2C -> SPI / QSFP bufory MLVDS / PHY Ethernet
TPS53353	20	0,95	9,78	FPGA
TPS53353	20	4	3,88	Układy dystrybucji sygnału zegarowego oraz toru analogowego (+LDO odpowiednio do 3,3 V i 3 V)
TPS53353	20	2,5	8,37	Przetworniki C/A (+LDO do 1,8 V)
TPS62175	0,5	6	0,213	HMC830 (+LDO do 5 V)
TPS63710	-1	-3,5	-0,628	Tor analogowy (+LDO do -3,0 V)

Tabela nie uwzględnia napięć generowanych przez układy typu LDO, jednak ich obecność została wspomniana w opisie zasilanych układów.

używany był programowalny XRP7724ILB-F. Jego główną zaletą była możliwość zdalnej zmiany parametrów pracy, która pozwalała na wprowadzanie poprawek wymaganych w związku ze starzeniem się poszczególnych elementów bez konieczności ingerencji mechanicznej. W trakcie testów wystąpił jednak problem z zanikającym napięciem w tym układzie po kilku dniach ciągłej pracy. Wszystkie nowe projekty wykorzystują zatem w jego miejsce LTM4644, umożliwiając jednocześnie montaż nakładki zawierającej wcześniej stosowane rozwiązanie.

Wszystkie przetwornice obniżające napięcie sterowane są przez mikrokontroler, który je włącza, a także odczytuje linie informujące o ustanowieniu oczekiwanej wartości napięcia. Umożliwia to implementację sekwencji uruchamiania zasilania dla układu FPGA w ściśle określonej kolejności [40]. Zgodnie z instrukcją sekwencja przedstawia się następująco:

1. $V_{CCINT}/V_{CCINTIO}$ oraz V_{CCBRAM} – 0,95 V (wewnętrzne napięcie zasilania oraz bloku RAM),
2. $V_{CCAUX}/V_{CCAUXIO}$ – 1,8 V (zasilanie pomocnicze),
3. V_{CCO} – zasilanie dla wszystkich banków wyprowadzeń ogólnego przeznaczenia zgodnie z przewidzianymi w projekcie napięciami.

W przypadku wyprowadzeń dla nadajników i odbiorników gigabitowych GTH:

1. V_{CCINT} – 0,95 V,
2. $V_{MGTAVCC}$ – 1 V (zasilanie analogowych obwodów nadajników i odbiorników GTH),
3. $V_{MGTAVTT}$ – 1,2 V (zasilanie analogowych obwodów terminujących GTH).

7.3. Realizacja i symulacje projektu obwodu drukowanego

Proces projektowania obwodu drukowanego w programie Altium Designer rozpoczęto od doboru konfiguracji warstw. Ich parametry związane były z możliwościami firmy produkującej oraz koniecznością zdefiniowania reguł dla ścieżek różnicowych o kontrolowanej impedancji 100 Ω , które fizycz-

nie pomieszczą się pomiędzy przelotkami umożliwiającymi wyprowadzenie sygnałów spod układów w obudowie BGA. Liczba takich sygnałów implikuje także konieczną do zastosowania ilość warstw. Ponieważ układ FPGA XCKU040-1FFVA1156C wykorzystany został także na płycie Sayma AMC, postanowiono wykorzystać zaimplementowaną tam konfigurację warstw w projekcie Shuttler (przedstawioną na rys. 7.10). Składa się ona z 6 warstw sygnałowych oraz 6 warstw zasilania.

#	Name	Material	Weight	Thickness	Dk
	Top Overlay				
	Top Solder	Soder_Mask		0,01mm	4,4
1	L1		1oz	0,04mm	
	Dielectric 1	FR408HR		0,13mm	3,7
2	P2		1/2oz	0,018mm	
	Dielectric2	FR408HR		0,1mm	3,7
3	L3		1/2oz	0,018mm	
	Dielectric3	FR408HR		0,13mm	3,7
4	P4		1/2oz	0,018mm	
	Dielectric4	FR408HR		0,1mm	3,7
5	L5		1/2oz	0,018mm	
	Dielectric5	FR408HR		0,13mm	3,7
6	P6		1/2oz	0,018mm	
	Dielectric6	FR408HR		0,1mm	3,7
7	P7		1/2oz	0,018mm	
	Dielectric7	FR408HR		0,13mm	3,7
8	L8		1/2oz	0,018mm	
	Dielectric8	FR408HR		0,1mm	3,7
9	P9		1/2oz	0,018mm	
	Dielectric9	FR408HR		0,13mm	3,7
10	L10		1/2oz	0,018mm	
	Dielectric10	FR408HR		0,1mm	3,7
11	P11		1/2oz	0,018mm	
	Dielectric11	FR408HR		0,13mm	3,7
12	L12		1oz	0,04mm	
	Bottom Solder	Solder Resist		0,01mm	4,4
	Bottom Overlay				

Rysunek 7.10: Zrzut ekranu przedstawiający konfigurację warstw obwodu drukowanego zastosowaną w projekcie Shuttler.

Zdefiniowano następnie profile ścieżek o kontrolowanej impedancji dla szybkich sygnałów:

- 25 Ω – dla sygnałów analogowych prowadzonych jako niezależne linie na krótkim odcinku od przetwornika do wzmacniacza,
- 43 Ω – dla sygnałów asymetrycznych SDRAM,
- 50 Ω – dla linii danych przetworników oraz sygnałów zegarowych prowadzonych linią asymetryczną,
- 88 Ω linia różnicowa – dla sygnałów różnicowych SDRAM,
- 90 Ω linia różnicowa – dla sygnału różnicowego USB,

- 100 Ω linia różnicowa – dla pozostałych sygnałów różnicowych.

Powstały dwa oddzielne profile dla linii różnicowych 100 Ω . Pierwszy dedykowany był dla najszybszych sygnałów gigabitowych m.in. linii SGMII, QSFP, DRTIO. Zawierał on szersze ścieżki (0,132 mm na warstwach zewnętrznych i 0,079 na warstwach wewnętrznych), a w konsekwencji także większy odstęp pomiędzy liniami pary różnicowej (0,123 mm na warstwach zewnętrznych i 0,132 mm na warstwach wewnętrznych). Takie podejście generowało mniejszy stopień sprzężenia, co pozwoliło z kolei uzyskać mniej drastyczny efekt nieciągłości impedancji, kiedy to linie muszą zostać rozdzielone, aby doprowadzić je do odpowiednich wyprowadzeń złącz. Drugi profil dedykowany był dla sygnałów zegarowych – o większym stopniu sprzężenia.

Wszystkie symulacje opisane w dalszych podrozdziałach wykonane zostały w programie HyperLynx z wykorzystaniem modeli IBIS dostarczanych przez producentów danych układów scalonych. Sam model płyty do symulacji wyeksportowany został bezpośrednio w programie Altium Designer. W poniższych podrozdziałach nie został opisany blok SDRAM. Wynika to z faktu, że połączenia dla tego bloku funkcjonalnego zostały zaimportowane z projektu Sayma AMC, który wykorzystuje te same modele układów scalonych oraz konfiguracji warstw. Moduł ten został już wyprodukowany oraz przetestowany, nie było zatem potrzeby dokonywania symulacji.

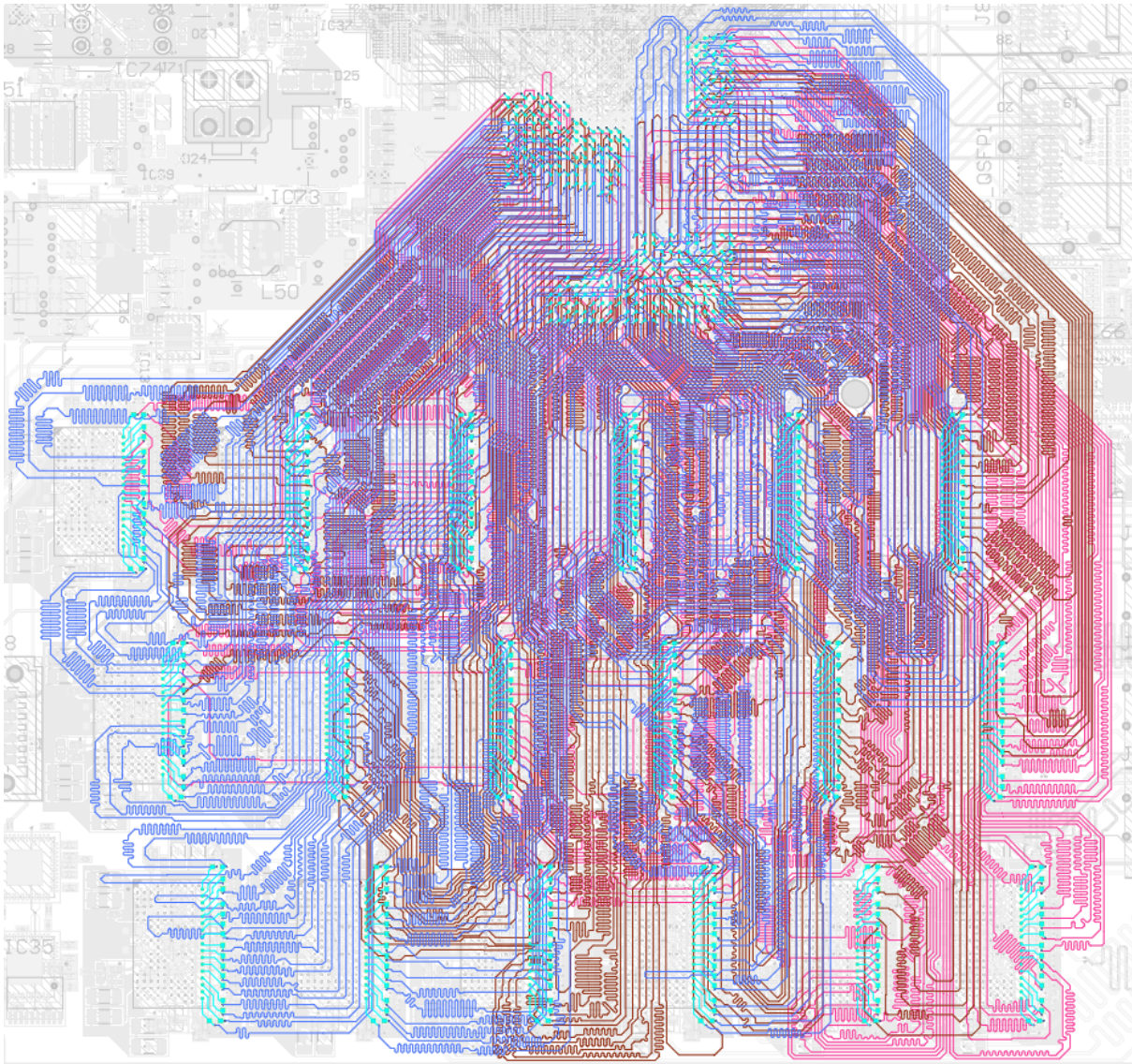
7.3.1. Połączenia linii danych przetworników

Jednym z największych wyzwań było zapewnienie połączeń badanego w niniejszej pracy interfejsu. Aby zapewnić synchronizację, konieczne było zachowanie stałej długości linii wszystkich sygnałów danych oraz sygnału zegarowego taktującego te dane. Ze względu na obecność aż 18 przetworników, różnica w linii prostej pomiędzy najmniej i najbardziej oddalonym od FPGA układem wynosiła aż 7 cm. Konieczne było także wykorzystanie maksymalnie 3 warstw sygnałowych, aby wciąż możliwe było doprowadzenie sygnałów sterujących, zegarowych taktujących próbkowanie oraz wyprowadzenie sygnałów analogowych. Ostateczne połączenia ukazane zostały na rys. 7.11.

Przy dopasowaniu długości ścieżek przyjęto margines błędu 5 mm, co przy założeniu czasu propagacji 15 cm/ns w materiale FR4 daje około 33 ps rozrzutu czasu propagacji. Dla częstotliwości próbkowania 1 GHz stanowi to około 3% okresu sygnału. W celu ograniczenia przesłuchów stosowano regułę odstępów pomiędzy kolejnymi liniami 0,3 mm (za wyjątkiem obszaru bezpośrednio pod układem FPGA). Ponieważ jednak część z nich prowadzona była równolegle przez stosunkowo długie odległości, wykonano symulacje z uwzględnieniem sprzężenia linii. Przykładowy diagram oka zaprezentowano na rys. 7.12. Wyniki dowiodły, że zastosowany odstęp był wystarczający – na żadnej z linii nie zaobserwowano znaczących przesłuchów.

7.3.2. Linie gigabitowe

Liniami wymagającymi najwięcej uwagi podczas projektowania obwodu drukowanego były te, związane interfejsami takimi jak QSFP, SATA, DRTIO czy SGMII. Linie prowadzone były w miarę możliwości na wierzchnich warstwach obwodu drukowanego, co minimalizowało straty związane z propagacją sygnału w ścieżce. Jak w przypadku wszystkich szybkich sygnałów zadbaną o ciągłość płaszczyzny masy na warstwie referencyjnej, a w przypadku konieczności zmiany warstwy prowadzenia ścieżki – także o obecność przelotki masy w bezpośrednim sąsiedztwie miejsca zmiany warstwy. Za-

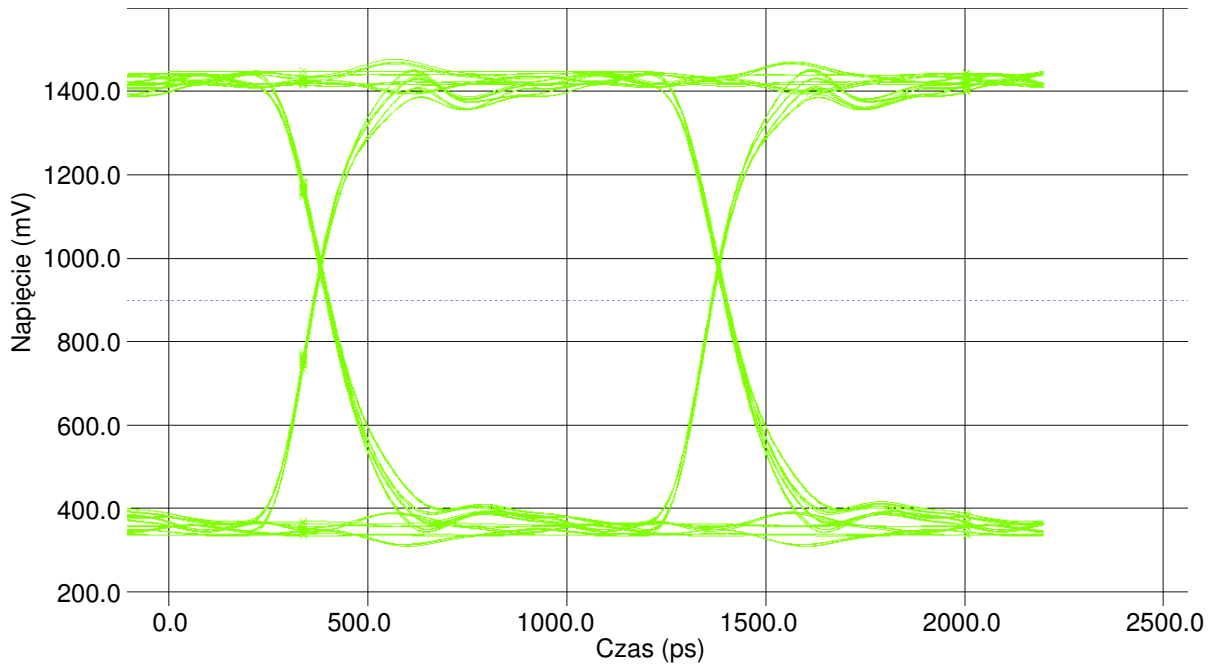


Rysunek 7.11: Połączenia linii sygnału danych oraz sygnału zegarowego danych wszystkich przetworników.

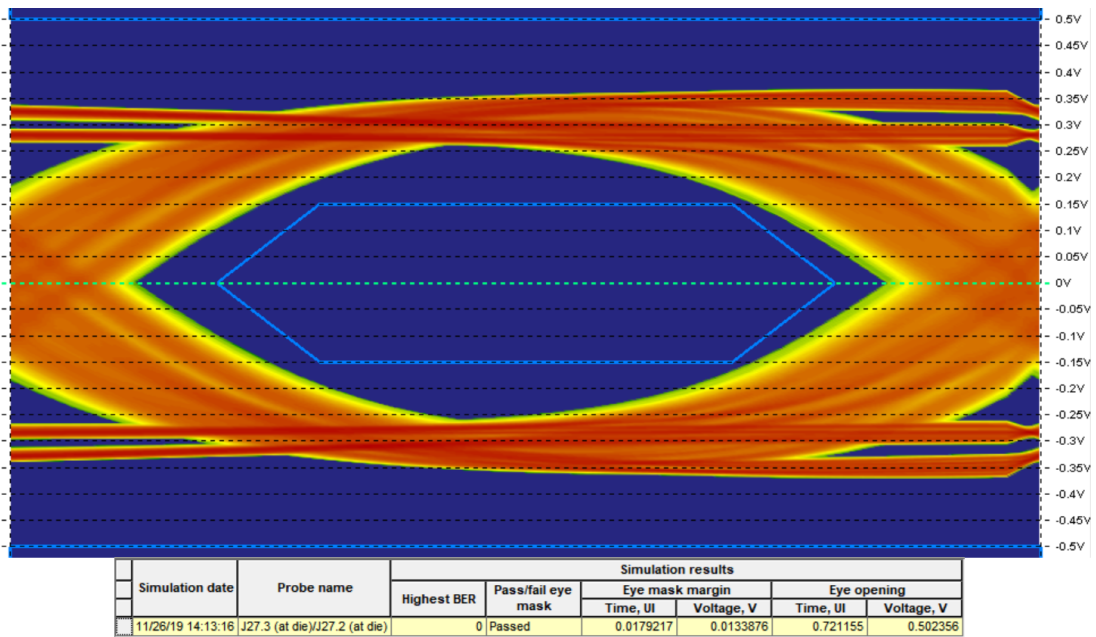
dbano także o zminimalizowanie wszelkich nieciągłości impedancji. Pola lutownicze kondensatorów oraz złączy ze względu na swoje wymiary posiadały znacząco niższą impedancję niż prowadzona ścieżka. Dlatego utworzono wycięcia w warstwie referencyjnej masy pod tymi polami, zmniejszając ich pojemność, a w konsekwencji zwiększając impedancję. W symulacjach wykorzystano ciągi pseudolosowe przesyłanych bitów oraz przepustowość linii odpowiednio 16 Gb/s (QSFP), 6 Gb/s (SATA, DRTIO), 1,25 Gb/s (SGMII). Użyte zostały modele IBIS Algorithmic Modeling Interface (IBIS-AMI) dostarczone przez firmę Xilinx. Te same modele zostały przypisane do wyprowadzeń złączy, zbadano zatem wyłącznie propagację sygnału w obrębie modułu Shuttler. Przykładowy wynik symulacji analizatora kanału IBIS-AMI przedstawiony został na rys. 7.13.

7.3.3. Sygnały zegarowe

Sygnały zegarowe podzielone zostały na dwie grupy konieczne do synchronizacji. Pierwsza z nich obejmowała sygnały zegarowe, o domyślnej częstotliwości 125 MHz, doprowadzone do układu

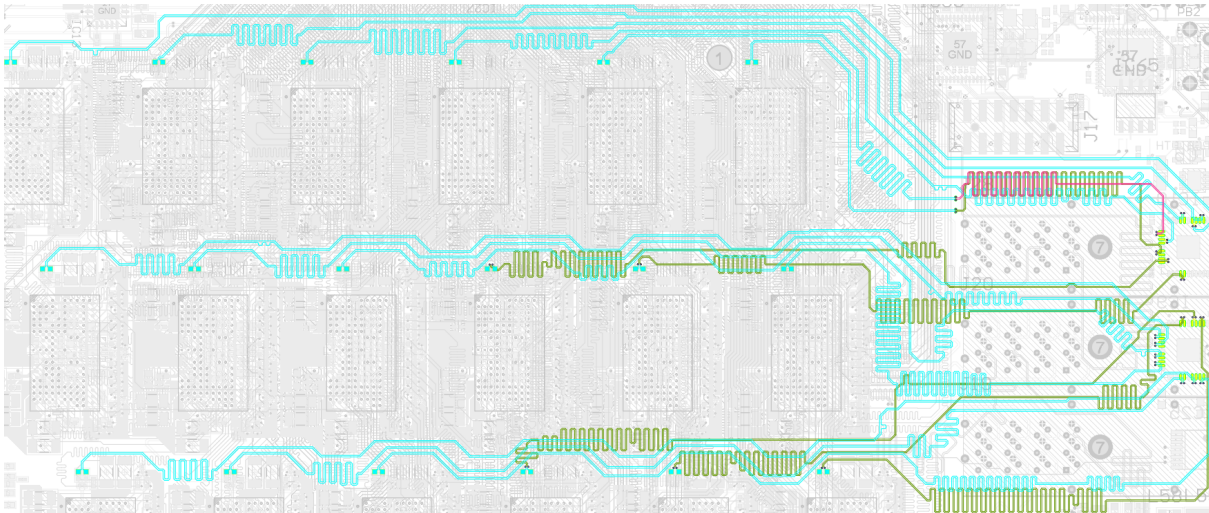


Rysunek 7.12: Symulacja diagramu oka dla linii danych przetwornika przy transferze 1 Gb/s z uwzględnieniem wpływu sąsiednich linii.

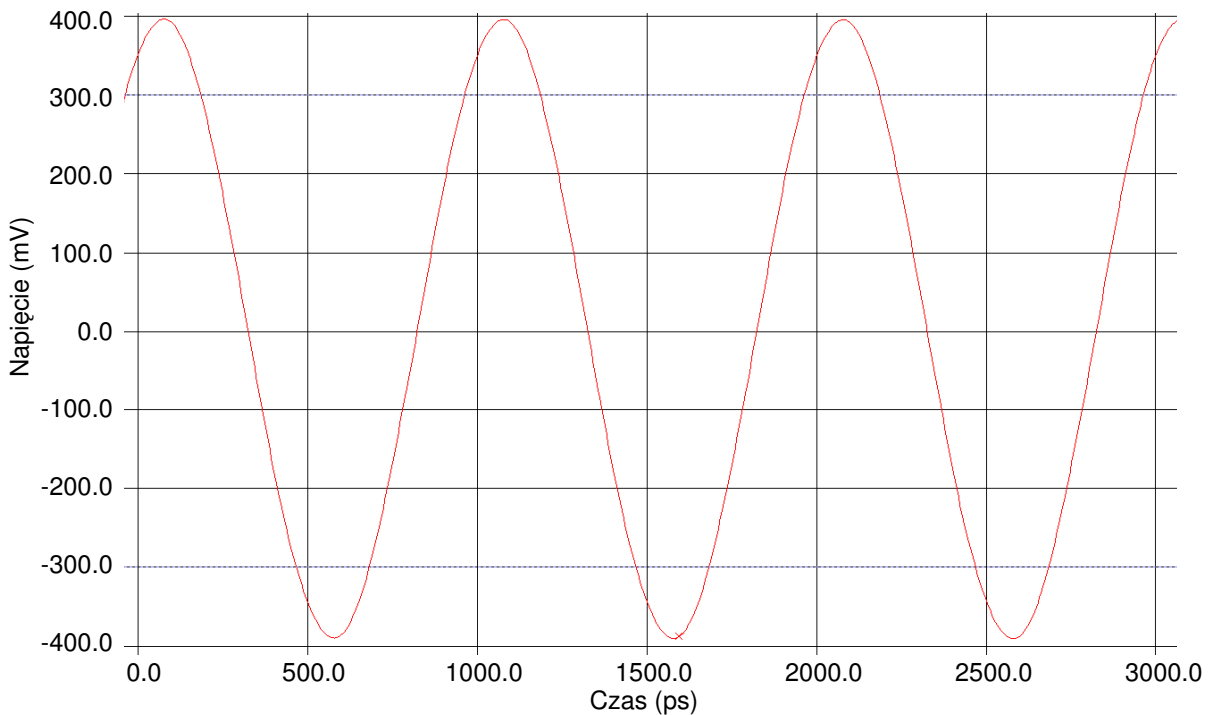


Rysunek 7.13: Diagram oka symulacji linii różnicowej QSFP przy przepustowości 16 Gb/s.

FPGA, pochodzące z pierwszego układu rozdzielającego ADCLK948. Wyrównanie długości ścieżek w tym wypadku zapewniało synchronizację wewnętrznej logiki Kintex UltraScale z resztą systemu. W ramach drugiej grupy należało zadbać o możliwie jak najlepszą synchronizację sygnału taktującego próbkowanie przetworników (do 1 GHz), aby każdy kanał zmieniał swoją wartość napięcia sterującego pułapką jonową w tym samym momencie. Było to zadanie trudniejsze ze względu na odległości pomiędzy skrajnie rozmieszczonymi przetwornikami. Połączenia przedstawione zostały na rys. 7.14. Przykładowa symulacja przedstawiona została na rys. 7.15.



Rysunek 7.14: Połączenia linii sygnałów zegarowych dla przetworników C/A.



Rysunek 7.15: Symulacja sygnału zegarowego taktującego przetwornik C/A dla częstotliwości 1 GHz.

7.3.4. Wyrównanie balansu miedzi

Pozostawiona przestrzeń na warstwach sygnałowych została wypełniona kwadratowymi obszarami miedzi. Gwarantuje to równomierne trawienie w obrębie całego obwodu drukowanego. Przykład zastosowanego balansowania miedzi pokazany został na rys. 7.16.

7.4. Analiza i porównanie kosztów całego systemu

Na potrzeby prezentacji osiągniętych wyników przyjęto następujące oznaczenia: Shuttler v.1 to moduł zgodny z pierwotną koncepcją sterowania przetworników z wykorzystaniem standardu wypro-



Rysunek 7.16: Przykład zaimplementowanego balansu miedzi.

wadzeń LVDS, z kolei Shuttler v.2 to zaprojektowany finalny moduł z zaimplementowaną metodą sterowania odbiorników przetwornika LTC2000 przy wykorzystaniu asymetrycznych linii w standardzie LVCMOS18. W tabeli 7.2 podsumowano koszt zestawienia rozważanego wielokanałowego systemu sterowania pałapkami jonowymi. W celu oceny dokonanej optymalizacji osiągnięty rezultat porównano wyłącznie z wariantem modułu Shuttler v.1, cechującym się najniższą ceną za pojedynczy kanał. Pomimo faktu, że zestawienie 1000 kanałów w ostatecznej wersji projektu nie wymaga użycia ponad 100 sztuk układu FPGA, to cena jednostkowa XCKU040-1FFVA1156C uwzględniona w obliczeniach wciąż wynosiła 55% ceny detalicznej, ponieważ zapewniona jest ona przez wykorzystywanie tego modelu w innych modułach będących częścią projektu Sinara. Dzięki zastosowaniu

Tabela 7.2: Zestawienie kosztów zakupu i produkcji sprzętu dla systemu sterowania pałapkami jonowymi opartego o moduły Shuttler v.1 oraz Shuttler v.2

Moduł	Liczba kanałów w module	Koszt Shuttler [tys. zł]	Liczba krat MicroTCA	Koszt systemu [tys. zł]	Liczba kanałów w systemie	Cena za kanał [zł]
Shuttler v.1 XCKU060-1FFVA1517	8	13,4	11	2 249,5	1056	2 130
Shuttler v.2 XCKU040-1FFVA1156C	18	14,8	5	1 102,0	1080	1 020
Różnica			6	1 147,5		1 110
Procentowo			54,5%	51,0%		52,1%

W tabeli koszt pełnego systemu zakłada pełne wyposażenie minimalnej liczby krat MicroTCA, dzięki której przekroczona zostanie wartość 1000 kanałów, w maksymalną liczbę kart AMC. Przyjęty koszt układów FPGA wynosi 55% ceny jednostkowej podawanej przez stronę Octopart [30].

opracowanej metody połączeń możliwe było obniżenie ceny za pojedynczy kanał o ponad połowę – wyniosła ona 1 110 zł (około 266 \$). Była to więc wartość niższa od zadeklarowanej przez fizyków uśrednionej kwoty w obecnie stosowanych systemach (wykorzystujących mniej precyzyjne układy). Co także istotne, zredukowano wymaganą liczbę krat MicroTCA o 6 sztuk. Wymiary przykładowej kraty mogącej pomieścić 12 kart AMC to 397,25 mm wysokości, 482,60 mm szerokości i 373,3 mm głębokości [16]. Mowa zatem o zmniejszeniu objętości całego systemu o około 430 dm³.

8. Wnioski

W ramach pracy opracowano ogólną koncepcję metody sterowania pułapkami jonowymi, będącą rozwiązaniem kompatybilnym z projektem Sinara. Przyjęta architektura pozwala na rozszerzenie funkcjonalności już wykorzystywanych systemów (zawierających kraty MicroTCA oraz moduły Metlino)[6], bez konieczności zestawiania osobnego sprzętu oraz opracowywania niezależnej warstwy programowej, wymagającej dodatkowych interfejsów do komunikacji i synchronizacji z pozostałymi podzespołami. Dzięki temu procedura dodawania nowego sprzętu odbędzie się minimalnym kosztem czasowym oraz finansowym. Samo stosowanie kraty zapewnia także ustandaryzowany sposób dostarczania zasilania dla poszczególnych układów oraz wydajny system chłodzenia, pozwalający na uzyskanie na pojedynczym module mocy rzędu 80 W.

Zastosowany schemat systemu oraz zaprojektowany moduł Shuttler pozwalają na wygodne skalowanie, polegające wyłącznie na dodawaniu kolejnych kart Shuttler do kraty MicroTCA. W przypadku wyczerpania zasobów pojedynczej kraty (zapewniającej zestawienie 216 kanałów), użytkownik może wykorzystać właściwości systemu Sinara do prostego podłączenia kolejnych krat - wymagane jest jedynie ustanowienie połączenia kablowego do złącz SFP dwóch kart Metlino. Sprawia to, że potencjalne rozbudowywanie systemu ograniczone jest wyłącznie do budżetu danego projektu oraz możliwej do wykorzystania przestrzeni zajmowanej przez kolejne kraty. Jednocześnie dzięki zastosowaniu przetwornika LTC2000, posiadającego parametry błędów nieliniowości całkowitej oraz różniczkowej na poziomie +/- 1 LSB oraz częstotliwości pracy do ponad 1 GS/s, generowany może być sygnał analogowy z wysoką precyzją i szybkością. Prowadzić to będzie nie tylko do zwiększenia liczby operacji jakie użytkownik będzie mógł przeprowadzić w trakcie skończonego czasu życia danych stanów kwantowych, ale także zwiększenia stosowanej liczby kubitów, wraz z którą rośnie wymagana liczba operacji transportu poszczególnych jonów. Żaden z dotychczas stosowanych systemów nie łączył wszystkich tych właściwości.

Uzyskanie tak wydajnego skalowania możliwe było dzięki przeprowadzonym badaniom nad niestandardowym interfejsem pomiędzy układem FPGA a przetwornikiem LTC2000. Analiza wyników badań pozwoliła na wyznaczenie optymalnego rozwiązania w postaci sterowania odbiornika LVDS linią asymetryczną, zaś przeprowadzone doświadczenia z wykorzystaniem sprzętu udowodniły słuszność teoretycznej koncepcji i możliwość jej fizycznej realizacji. Dzięki temu uzyskano trzykrotne zwiększenie liczby kanałów przypadających na pojedynczy bank wyprowadzeń układu FPGA. To z kolei w konsekwencji doprowadziło do ponad dwukrotnego zmniejszenia kosztu zestawienia potencjalnego 1000-kanałowego systemu sterowania pułapkami jonowymi. Cena za pojedynczy kanał przed badaniami wynosiła około 2 130 zł, zaś została zredukowana do kwoty około 1 020 zł. O skali dokonanej optymalizacji świadczy fakt, że osiągnięty koszt zestawienia pojedynczego kanału w proponowanym systemie jest mniejszy od zadeklarowanego przez fizyków kosztu ponoszonego w obecnie stosowanych systemach przy jednoczesnym wzroście precyzji stosowanych komponentów.

Obecność układu FPGA wraz z pamięcią DDR3 zapewniła kompatybilność z warstwą programową ARTIQ. Dzięki temu możliwe jest uzyskanie synchronizacji modułu Shuttler z resztą systemu z dokładnością do 1 ns, poprzez wykorzystanie sygnałów DRTIO oraz układu tłumiącego jitter sygnału zegarowego. Jest to parametr wyróżniający cały projekt Sinara z pośród innych rozwiązań dostępnych na rynku. Zastosowanie architektury płyty z pojedynczym układem FPGA (bez układów deserializujących) generującym wartości próbek analogowych pozwoliło na zminimalizowanie latencji (jednostka obliczeniowa i sterująca przetworniki C/A jest tym samym urządzeniem).

Obecność prostego toru analogowego na wyjściu przetworników zapewnia możliwość stosowania modułu Shuttler przy dowolnych pułapkach jonowych. Fizykom przeprowadzającym doświadczenia pozostaje konstrukcja toru samej pułapki, filtrującego sygnał zgodnie z oczekiwaniami oraz dostosowującego poziomy napięciowe do danego sprzętu – tak jak w dotychczas stosowanych rozwiązaniach.

Sam moduł wciąż może być stosowany jako niezależne urządzenie, bez utraty swoich parametrów, dzięki czemu w małej skali możliwe jest uniknięcie ponoszenia kosztu zestawienia kraty MicroTCA wraz z modułem zarządzającym. Jest to efektem uwzględnienia dodatkowego złącza zasilania oraz dwóch złącz QSFP, które pozwalają na dostarczenie sygnału DRTIO poprzez połączenie kablowe i dodawanie kolejnych modułów, tworząc połączenia typu "daisy chain".

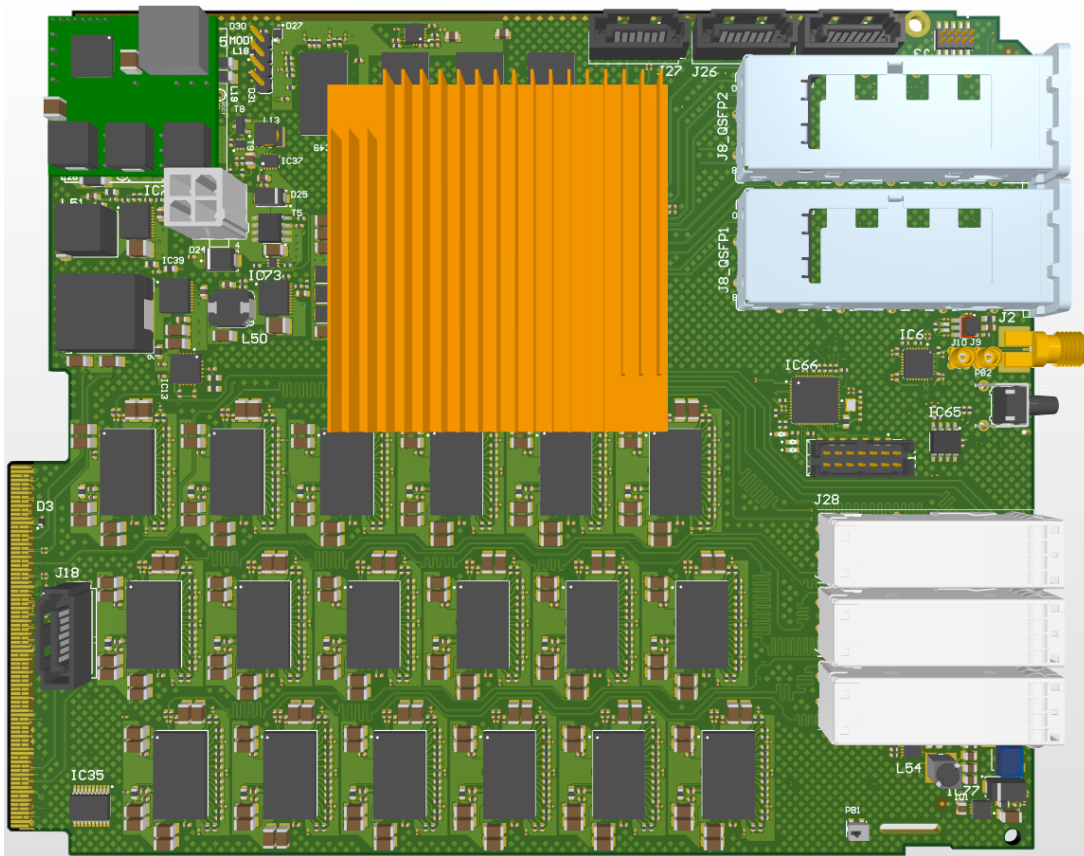
Podsumowanie parametrów i cech modułu Shuttler:

- Układ FPGA Kintex UltraScale XCKU040-1FFVA1156C wraz z 16 Gb pamięci DDR3.
- 18 kanałów przetworników C/A LTC2000-16 pracujących z konfigurowalną częstotliwością generacji próbek z zakresu 50 - 1000 MHz.
- Sygnały wyjściowe wyprowadzone poprzez 3 złącza typu Mini-SAS HD.
- Latencja generacji sygnału analogowego dla częstotliwości próbkowania 200 MS/s wynosi około 41,5 ns (od momentu wysterowania poziomów logicznych przez układ FPGA).
- Wymagana maksymalna moc 72,8 W przy generacji próbek z częstotliwością 1 GS/s.
- Dostępne złącza w przypadku pracy poza kratą MicroTCA (stanowiąca alternatywne źródła danych i sygnałów zegarowych): 2 x QSFP, 3 x SATA, 1 x MCX, 2 x MMCX, 1 x SMA.

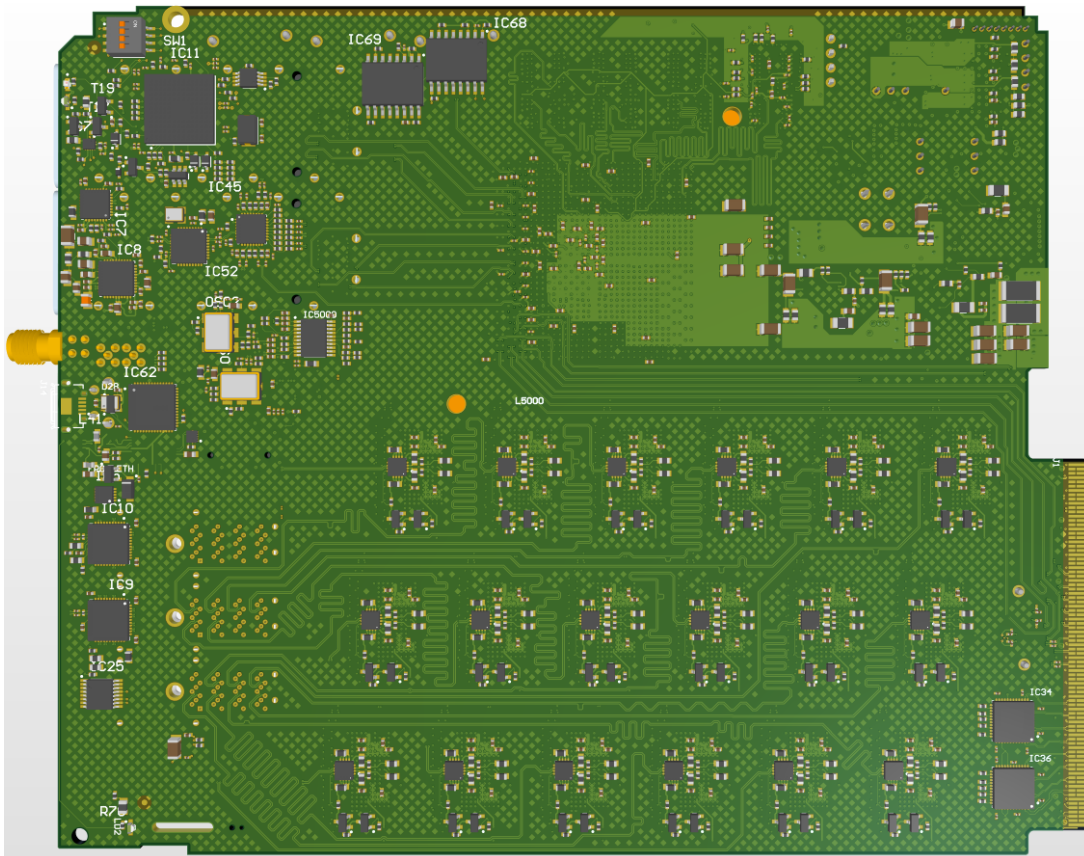
Widok 3D gotowego projektu płyty został przedstawiony na rys. 8.1 oraz 8.2.

Projekt otrzymał finansowanie w ramach wniosku NCBiR o nazwie NESTER na produkcję prototypowych modułów w 2021 roku. Przyszłymi zadaniami autora pracy będzie ich uruchomienie i znalezienie oraz zdiagnozowanie wszelkich potencjalnych błędów w warstwie sprzętowej modułu. Równolegle zespół odpowiedzialny za oprogramowanie rozpocznie opracowywanie nowej wersji ARTIQ w pełni wspierającej płytę Shuttler.

W ramach dalszego rozwoju przewidziane jest przeprowadzenie badań nad sprzętem, mające na celu wyznaczenie optymalnego standardu logicznego dla wyprowadzeń FPGA, przesyłających dane do przetworników. Pozwoli to na niewielkie ograniczenie wydzielanej mocy, co pozytywnie wpłynie na stabilność parametrów LTC2000. Planowane są także badania pod kątem realnego wpływu temperatury na działanie sieci przetworników – zmierzone zostaną rozrzuty wartości sygnałów analogowych, powstające w zależności od położenia układów na płycie, a także sprawdzona zostanie zasadność potencjalnego wykorzystania wspólnego radiatora, obejmującego wszystkie układy LTC2000.



Rysunek 8.1: Wizualizacja 3D modułu – widok warstwy górnej.



Rysunek 8.2: Wizualizacja 3D modułu – widok warstwy dolnej.

9. Lista skrótów

- Advanced Mezzanine Cards (AMC)** Obwód drukowany zgodny ze specyfikacją określoną przez PCI Industrial Computers Manufacturers Group [2], przeznaczony do pracy w kratach typu TCA. 16, 21–23, 25–29, 37, 44, 47–49, 57, 58, 60–63, 69, 71, 74, 75, 79, 85
- Advanced Real-Time Infrastructure for Quantum physics (ARTIQ)** Oprogramowanie dla infrastruktury doświadczeń kwantowych pracujące w czasie rzeczywistym. 15, 16, 20, 21, 26, 63, 64, 80, 81
- Alternating Current (AC)** Oznaczenie wykorzystywane do rozróżnienia typów zasilania bądź sygnałów – w tym wypadku cechujących się prądem okresowo zmiennym. 69
- Application-Specific Integrated Circuit (ASIC)** Specjalizowany układ scalony zaprojektowany do pełnienia określonych funkcji. 13, 19, 28
- Ball Grid Array (BGA)** Rodzaj obudowy do montażu powierzchniowego z wyprowadzeniami sferycznymi ułożonymi w siatkę. 34, 38, 46, 74
- Differential Nonlinearity (DNL)** Nieliniowość różniczkowa. 17, 24
- Direct Current (DC)** Oznaczenie wykorzystywane do rozróżnienia typów zasilania bądź sygnałów – w tym wypadku cechujących się prądem stałym. 11, 13
- Direct Digital Synthesis (DDS)** Bezpośrednia synteza cyfrowa. 14, 17
- Distributed Real-Time Input/Output (DRTIO)** System transferu czasu i danych do synchronizacji i kontroli w czasie rzeczywistym podsystemów przez urządzenie nadrzędne. 17, 21–23, 26, 62, 63, 67, 69, 75, 76, 80, 81
- Double Data Rate (DDR)** Podwójna przepustowość danych – cecha układów taktujących dane przy obydwu zboczach sygnału zegarowego. 26, 48, 64, 80, 81
- Electrically Erasable Programmable Read-Only Memory (EEPROM)** Rodzaj pamięci nieulotnej, która wbrew nazwie może być kasowana i ponownie zapisywana. Proces kasowania odbywa się wyłącznie przy użyciu prądu elektrycznego. 66, 71
- Eurocard Extension Modules (EEM)** Wewnętrzny standard projektu Sinara modułów rozszerzeń kontrolowanych przez ARTIQ, przewidzianych do pracy w "Kasli Box". 21, 22
- Field Programmable Gate Array (FPGA)** Rodzaj programowalnego układu logicznego 15, 16, 21, 22, 25–35, 37, 38, 41, 42, 44–46, 48, 51, 52, 56, 58, 60, 61, 63–75, 77, 79–81, 83
- FPGA Mezzanine Card (FMC)** Standard obwodów drukowanych stanowiących moduły rozszerzeń połączonych do układów FPGA. 28, 45–48
- high-performance (HP)** Typ banków wyprowadzeń wejścia wyjścia stosowany w układach FPGA firmy Xilinx charakteryzujący się niższymi napięciami pracy i większymi osiągalnymi prędkościami sygnałów. 32, 40
- High-pin count (HPC)** Typ złącza FMC posiadający 400 wyprowadzeń 48
- high-range (HR)** Typ banków wyprowadzeń wejścia wyjścia stosowany w układach FPGA firmy Xilinx charakteryzujący się szerokim zakresem napięciowym pracy. 32, 38, 40, 63, 64

- IBIS Algorithmic Modeling Interface (IBIS-AMI)** Rozwinięcie modeli IBIS wykorzystywane dla gigabitowych wyprowadzeń wejścia/wyjścia. 76
- Input/output Buffer Information Specification (IBIS)** Model behawioralny opisujący właściwości elektryczne cyfrowych wejść i wyjść urządzenia poprzez dane napięcia/natężenia prądu bez ujawniania zastrzeżonych informacji o budowie układu. 37, 41–43, 45, 46, 49, 52, 59, 75, 84
- Integral Nonlinearity (INL)** Nieliniowość całkowita. 24
- Intelligent Platform Management Bus (IPMB)** Magistrala do zarządzania kratą MicroTCA przez moduł MCH. 63, 84
- Inter-Integrated Circuit (I²C)** Prosta dwuliniowa magistrala szeregowo wykorzystująca adresowanie układów podrzędnych. 66, 70, 71
- Joint Test Action Group (JTAG)** Standard definiujący protokół pozwalający na programowanie i uruchamianie układów programowalnych. Jest on także wykorzystywany do testowania połączeń w obwodach drukowanych drukowanych. 48, 63, 69
- Light Emitting Diode (LED)** Dioda elektroluminescencyjna 66
- Low Dropout (LDO)** Typ regulatora napięcia cechujący się możliwością pracy przy niewielkim spadku pomiędzy napięciem wejściowym a wyjściowym 71, 73
- Low-pin count (LPC)** Typ złącza FMC posiadający 160 wyprowadzeń 48
- Low-Voltage Differential Signaling (LVDS)** Standard specyfikujący niskomocowy różnicowy protokół komunikacyjny. 22, 25, 30, 31, 35–38, 40–42, 45, 51, 53, 56, 58, 59, 66, 79, 80, 84
- Low-Voltage Positive Emitter-Coupled Logic (LVPECL)** Niskonapięciowa logika oparta na szybkich tranzystorach bipolarnych. Zgodnie z dosłownym tłumaczeniem posiada dwa tranzystory wejściowe o zwartych emiterach. 67, 69
- Low-Voltage Transistor–Transistor Logic (LVTTTL)** Niskonapięciowa wersja standardu poziomów logicznych układów scalonych, bazujący na tranzystorach bipolarnych. 30
- Micro Coaxial Connector (MCX)** Rodzaj miniaturowego złącza RF, cechującego się zapadkowym mechanizmem blokującym 69, 84
- Micro-miniature Coaxial (MMCX)** Zminiaturyzowana wersja złącza MCX 69
- MicroTCA Carrier Hub (MCH)** Główna płyta zarządzająca kratą MicroTCA - kontroluje system zasilania, chłodzenia i zapewnia komunikację z wszystkimi modułami wewnątrz kasety. 26, 28, 31, 65, 69, 84
- Mini Serial Attached Small Computer Systems Interface High Density (Mini-SAS HD)** Rodzaj interfejsu komunikacyjnego. 27, 61, 67, 81
- Module Management Control (MMC)** Kontroler zarządzający modulem i połączony interfejsem IPMB z modulem zarządzającym kratą MicroTCA. 26, 65, 70
- Multipoint Low-Voltage Differential Signaling (MLVDS)** Rozwinięcie standardu LVDS pozwalające na zestawienie połączenia pomiędzy większą liczbą urządzeń. 31, 32, 62, 63, 73
- National Institute of Standards and Technology (NIST)** Amerykański Narodowy Instytut Standaryzacji i Technologii będący odpowiednikiem polskiego Głównego Urzędu Miar 15

- Peripheral Component Interconnect Express (PCIe)** Standard połączenia typu punkt punkt charakteryzujący się dużą przepustowością danych. Typowo kojarzony z połączeniami na płytach głównych komputerów umożliwiającymi na dodanie kart rozszerzeń takich jak karty graficzne, sieciowe itp.. 15
- Phase Locked Loop (PLL)** Pętla synchronizacji fazy. 68, 69
- Quad SFP (QSFP)** Poczwórna wersja SFP 21, 26, 61, 63, 64, 69, 71, 73, 75–77, 81
- Radio Frequency (RF)** Częstotliwość radiowa – częstotliwości z zakresu od kilkudziesięciu kHz do 300 GHz. 11, 13, 26, 48, 84
- Random-Access Memory (RAM)** Pamięć o dostępie swobodnym. 21, 22, 73
- Rear Transition Module (RTM)** Karta stanowiąca rozszerzenie powierzchni oraz panelu na połączenia zewnętrzne - w niniejszej pracy dla modułów AMC 21, 22, 25, 26, 28, 47, 71
- Serial Advanced Technology Attachment (SATA)** Rodzaj szeregowej magistrali komputerowej. 63, 75, 76, 81
- Serial Peripheral Interface (SPI)** Interfejs komunikacyjny, zapewniający synchroniczną, szeregową komunikację pomiędzy urządzeniami. 45, 46, 48, 49, 51, 66, 71
- Small Form-factor Pluggable (SFP)** Nadajnik/odbiornik o dużej przepustowości z opcją zestawienia połączenia w już uruchomionym systemie 21, 24, 26, 80, 85
- SubMiniature version A (SMA)** Złącze współosiowe o impedancji 50Ω dla sygnałów do 18 GHz. 45, 48, 50, 69, 81
- Synchronous Dynamic Random-Access Memory (SDRAM)** Rodzaj dynamicznej pamięci, która jest odczytywana synchronicznie do zegara podłączonej magistrali 26, 28, 48, 60, 63, 64, 66, 69, 73–75
- Universal asynchronous receiver-transmitter (UART)** Interfejs komunikacyjny, umożliwiający szeregową komunikację dwóch urządzeń 48, 49, 70
- Universal Serial Bus (USB)** Komputerowy szeregowy interfejs komunikacyjny, umożliwiający podłączenie wielu różnych urządzeń 26, 48, 69, 70, 74
- VHSIC (Very High Speed Integrated Circuit) Hardware Description Language (VHDL)** Język opisu sprzętu. 51
- White Rabbit Project (WR)** Projekt powstały we współpracy wielu instytucji badawczych (m.in. CERNu czy GSI) mający na celu opracowanie technologii pozwalającej na synchronizację urządzeń z dokładnością sub-nanosekundową. 68

Bibliografia

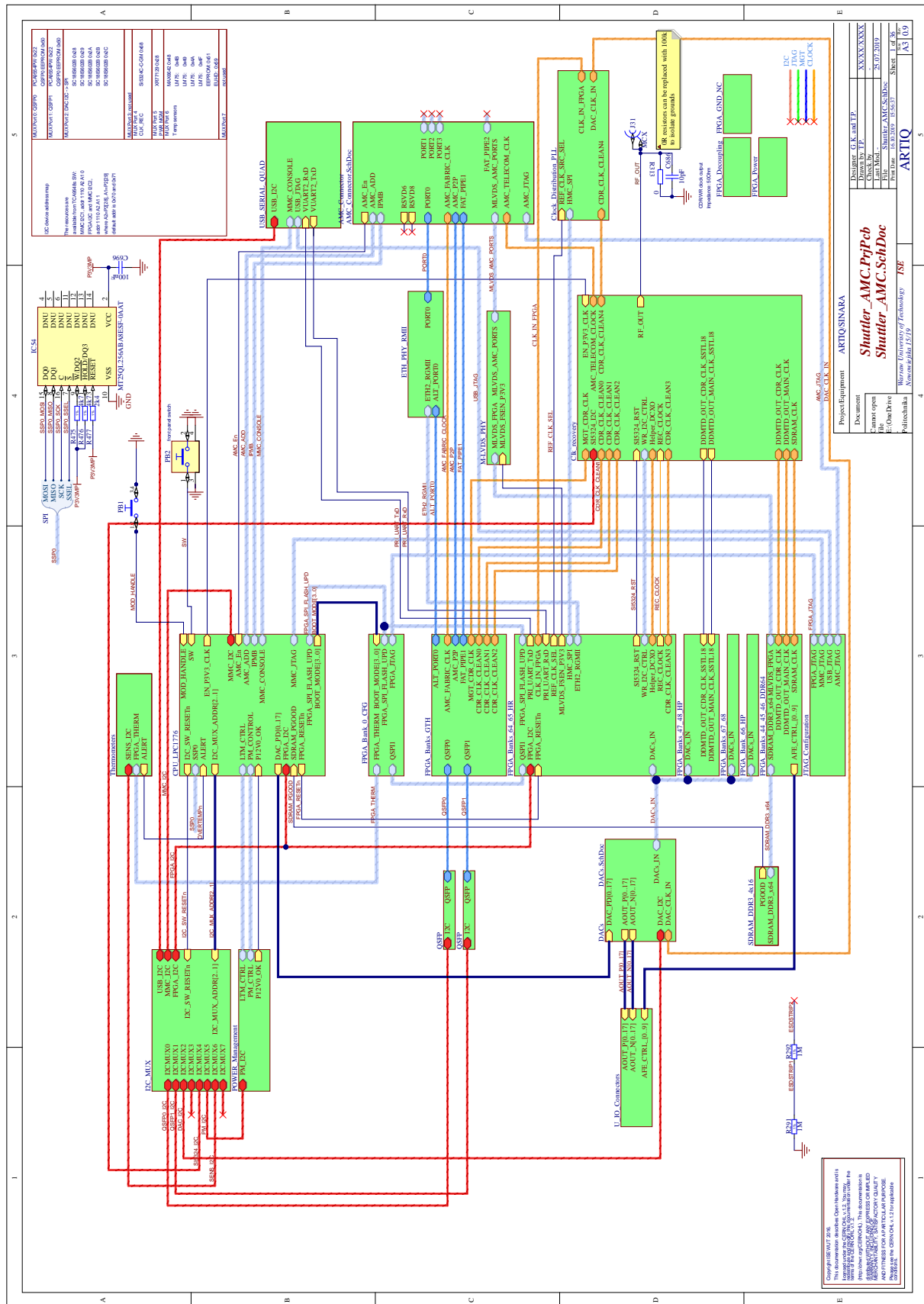
- [1] David Thomas Charles Allcock, *Surface-electrode ion traps for scalable quantum computing*, praca doktorska, Oxford University, UK, 2011.
- [2] *AMC.0 R2.0 Short Form Specification*, Rev. 2.0, PICMG, grud. 2006, URL: https://www.picmg.org/wp-content/uploads/AMC.0_R2.0_Short_Form.pdf (termin wizyty 25.05.2019).
- [3] Analog Devices, *DC2303A-A - LTC2000-16 Demo Board (FMC)*, URL: <https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/dc2303a-a.html#eb-overview> (termin wizyty 25.09.2019).
- [4] Jan Benhelm, *Precision spectroscopy and quantum information processing with trapped calcium ions*, na, 2008.
- [5] Daniel Bernstein, *Introduction to post-quantum cryptography*, sty. 2009, strony 1–14, DOI: 10.1007/978-3-540-88702-7_1.
- [6] Joseph Britton, *ARTIQ/Sinara Survey results*, 21 wrz. 2018, URL: <https://ssl.serverraum.org/lists-archive/artiq/2019-January/001248.html> (termin wizyty 31.01.2019).
- [7] Federico Carminati, *CERN, IBM Collaborate on Quantum Computing*, 12 mar. 2018, URL: <https://www.ibm.com/blogs/research/2019/03/cern-ibm-quantum/> (termin wizyty 10.10.2019).
- [8] CAS DataLoggers, *ADwin-PRO-II*, URL: <https://www.dataloggerinc.com/product/adwin-pro-ii-modular-real-time-data-acquisition-system/> (termin wizyty 11.10.2019).
- [9] Analog Devices, *AD9726 - przegląd produktu*, URL: <https://www.analog.com/en/products/ad9726.html#product-overview> (termin wizyty 25.11.2018).
- [10] Analog Devices, *AD9726 16-bit, 400 MSPS Digital-to-Analog Converter datasheet*, URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD9726.pdf> (termin wizyty 28.08.2019).
- [11] Analog Devices, *LTC2000 Datasheet*, URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/2000fb.pdf> (termin wizyty 11.10.2019).
- [12] Analog Devices, *LTpowerCAD and LTpowerPlanner*, URL: <https://www.analog.com/en/design-center/ltpowercad.html> (termin wizyty 25.11.2018).
- [13] Analog Devices, *Program ADI DiffAmpCalc*, URL: <https://www.analog.com/en/design-center/interactive-design-tools/adi-diffampcalc.html> (termin wizyty 25.11.2018).
- [14] Analog Devices, *Program ADIsimPLL*, URL: https://form.analog.com/form_pages/rfcomms/adisimpll.aspx (termin wizyty 25.11.2018).
- [15] *Eurocard Extension Modules (v1.0)*, URL: <https://github.com/sinara-hw/meta/wiki/EEM> (termin wizyty 25.11.2019).
- [16] Schroff GmbH, *MTCA.4 Shelf User's Manual*, URL: <https://schroff.nvent.com/wcsstore/ExtendedSitesCatalogAssetStore/Attachment/SchroffAttachments/Documents/63972-328.pdf> (termin wizyty 30.10.2019).
- [17] Google, *A Preview of Bristlecone, Google's New Quantum Processor*, 5 mar. 2018, URL: <https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html> (termin wizyty 10.10.2019).
- [18] Nicholas D. Guise i inni, *In-vacuum active electronics for microfabricated ion traps*, „Review of Scientific Instruments” 2014 85.nr. 6, strona 063101, DOI: 10.1063/1.4879136, URL: <http://dx.doi.org/10.1063/1.4879136>.

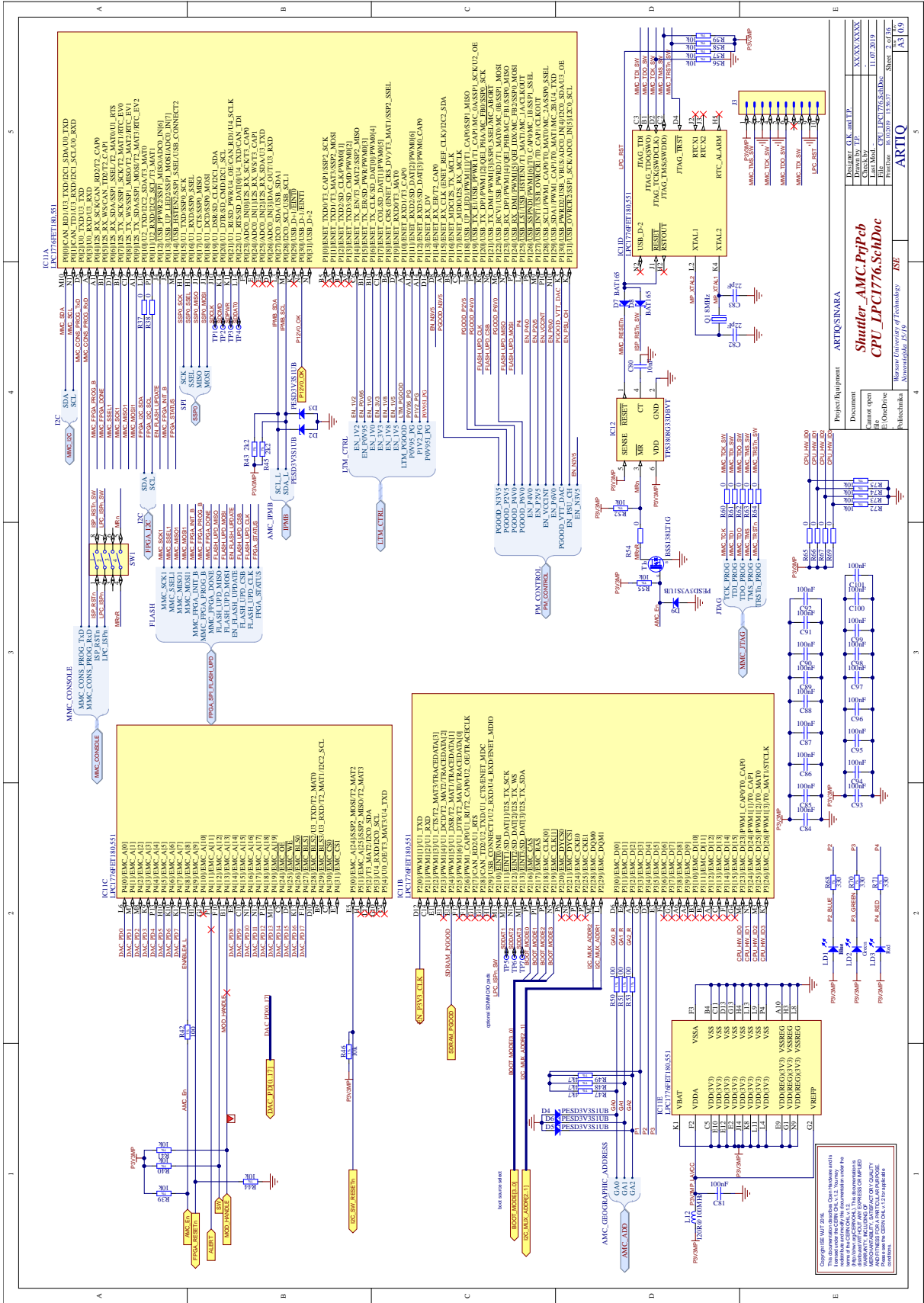
- [19] Elma Electronic Inc., *MicroTCA.4 System Platform, 12 Slot Crate - Datasheet*, URL: <https://www.elma.com/-/media/product-files/region-americas/systems-solutions/chassis-platforms/19-rackmount/type-11/datasheets/us-sp-mtca4-12-slot-d.ashx?la=en&hash=A285D81349B32B357B30186009B5D7E1DCAFD5A2> (termin wizyty 25. 11. 2018).
- [20] Intel, *The Future of Quantum Computing is Counted in Qubits*, 2 maj. 2018, URL: <https://newsroom.intel.com/news/future-quantum-computing-counted-qubits/#gs.bk748r> (termin wizyty 10. 10. 2019).
- [21] Grzegorz Kasprowicz, *Repozytorium: Sayma AMC*, URL: https://github.com/sinara-hw/Sayma_AMC (termin wizyty 25. 11. 2018).
- [22] M-Labs, *Strona projektu Artiq*, URL: <https://m-labs.hk/experiment-control/artiq/> (termin wizyty 25. 11. 2018).
- [23] Peter Lukas Wilhelm Maunz, *High Optical Access Trap 2.0*. Sty. 2016, DOI: 10.2172/1237003.
- [24] Maxim, *16-Bit, 500MSPS Interpolating and Modulating Dual DAC with CMOS Inputs - datasheet*, URL: <https://datasheets.maximintegrated.com/en/ds/MAX5895.pdf> (termin wizyty 28. 08. 2019).
- [25] Maxim, *3.3V, 16-Bit, 500MSPS High Dynamic Performance DAC with Differential LVDS Inputs - datasheet*, URL: <https://datasheets.maximintegrated.com/en/ds/MAX5888.pdf> (termin wizyty 28. 08. 2019).
- [26] Micron, *TN-41-13: DDR3 Point-to-Point Design Support*, sierp. 2013, URL: https://www.micron.com/-/media/client/global/documents/products/technical-note/dram/tn4113_ddr3_point_to_point_design.pdf (termin wizyty 28. 08. 2019).
- [27] Christopher R. Monroe, Robert J. Schoelkopf i Mikhail D. Lukin, *Technology: Quantum Connections*, „Scientific American” 2016 314.nr. 5, strony 50–57, DOI: <https://doi.org/10.1038/scientificamerican0516-50>, URL: <http://www.nature.com/scientificamerican/journal/v314/n5/full/scientificamerican0516-50.html>.
- [28] P Moreira i I Darwazeh, *Digital femtosecond time difference circuit for CERN's timing system*, URL: <http://www.ee.ucl.ac.uk/lcs/previous/LCS2011/LCS1136.pdf> (termin wizyty 30. 10. 2019).
- [29] Altaf H Nizamani i inni, *Computer-based frequency drift control of multiple LASERS in real-time*, „Int. J. Sci. Eng. Res” 2013 4, strona 3038.
- [30] Octopart, *Octopart - strona główna*, URL: <https://octopart.com/> (termin wizyty 25. 11. 2018).
- [31] Elia Perego i inni, *A scalable hardware and software control apparatus for experiments with hybrid quantum systems*, „Review of Scientific Instruments” 2018 89.nr. 11, strona 113116, DOI: 10.1063/1.5049120, URL: <http://dx.doi.org/10.1063/1.5049120>.
- [32] John Preskill, *Quantum Computing in the NISQ era and beyond*, „Quantum” 2018 2, strona 79, DOI: 10.22331/q-2018-08-06-79, URL: <http://dx.doi.org/10.22331/q-2018-08-06-79>.
- [33] *Repozytorium projektu Sinara*, URL: <https://github.com/sinara-hw/sinara/wiki> (termin wizyty 25. 11. 2018).
- [34] Peter W. Shor, *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, „SIAM Journal on Computing” 1997 26.nr. 5, 1484–1509, DOI: 10.1137/S0097539795293172, URL: <http://dx.doi.org/10.1137/S0097539795293172>.
- [35] Microsoft Quantum Team, *Quantum team shows how the best quantum tools are getting even better*, 16 maj. 2018, URL: <https://cloudblogs.microsoft.com/quantum/2019/05/16/>

build-2019-quantum-team-shows-how-the-best-quantum-tools-are-getting-even-better/ (termin wizyty 10. 10. 2019).

- [36] Linear Technology, *LTC1666/LTC1667/LTC1668 12-Bit, 14-Bit, 16-Bit, 50 Msps DACs - data-sheet*, URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/166678f.pdf> (termin wizyty 28. 08. 2019).
- [37] Tektronix, *P6139A 10X Passive Probe*, URL: https://www.csus.edu/indiv/t/tatror/eee_117_lab/3017_Test_Equipment/Tektronix%20P6139A%20Probe.pdf (termin wizyty 28. 08. 2019).
- [38] TSMC, *TSMC and OIP Ecosystem Partners Deliver Industry's First Complete Design Infrastructure for 5nm Process Technology*, 3 kw. 2019, URL: <https://www.tsmc.com/tsmcdotcom/PRListingNewsAction.do?action=detail&language=E&newsid=THPGWQTHTH> (termin wizyty 10. 10. 2019).
- [39] dr hab. Rafał Demkowicz-Dobrzański prof. UW, *Komputery kwantowe – od Feynmana do Google'a*, „DELTA” 2017 12 (523) 2017.
- [40] Xilinx, *Kintex UltraScale FPGAs Data Sheet: DC and AC Switching Characteristics*, 21 maj. 2019, URL: https://www.xilinx.com/support/documentation/data_sheets/ds892-kintex-ultrascale-data-sheet.pdf (termin wizyty 28. 08. 2019).
- [41] Xilinx, *UltraScale Architecture-Based FPGAs Memory IP v1.4*, 30 paź. 2019, URL: https://www.xilinx.com/support/documentation/ip_documentation/ultrascale_memory_ip/v1_4/pg150-ultrascale-memory-ip.pdf (termin wizyty 01. 11. 2019).
- [42] Xilinx, *UltraScale Architecture SelectIO Resources. User Guide*, 28 sierp. 2019, URL: https://www.xilinx.com/support/documentation/user_guides/ug571-ultrascale-selectio.pdf (termin wizyty 28. 09. 2019).
- [43] Xilinx, *Xilinx Power Estimator (XPE) - UltraScale*, 20 grud. 2017, URL: https://www.xilinx.com/member/forms/download/power-estimator-license.html?cid=d247ed6b-6836-46a4-bf71-8ae59222fc27&filename=UltraScale_XPE_2017_4.zip (termin wizyty 25. 11. 2018).

A. Schematy elektryczne płyty Shuttler





Section	Pin	Function
C1A	P4060	MMC_CONSOLE
	P4061	MMC_CONSOLE
	P4062	MMC_CONSOLE
	P4063	MMC_CONSOLE
	P4064	MMC_CONSOLE
	P4065	MMC_CONSOLE
	P4066	MMC_CONSOLE
	P4067	MMC_CONSOLE
	P4068	MMC_CONSOLE
	P4069	MMC_CONSOLE
C1B	P4070	MMC_CONSOLE
	P4071	MMC_CONSOLE
	P4072	MMC_CONSOLE
	P4073	MMC_CONSOLE
	P4074	MMC_CONSOLE
	P4075	MMC_CONSOLE
	P4076	MMC_CONSOLE
	P4077	MMC_CONSOLE
	P4078	MMC_CONSOLE
	P4079	MMC_CONSOLE
C1C	P4080	MMC_CONSOLE
	P4081	MMC_CONSOLE
	P4082	MMC_CONSOLE
	P4083	MMC_CONSOLE
	P4084	MMC_CONSOLE
	P4085	MMC_CONSOLE
	P4086	MMC_CONSOLE
	P4087	MMC_CONSOLE
	P4088	MMC_CONSOLE
	P4089	MMC_CONSOLE
C1D	P4090	MMC_CONSOLE
	P4091	MMC_CONSOLE
	P4092	MMC_CONSOLE
	P4093	MMC_CONSOLE
	P4094	MMC_CONSOLE
	P4095	MMC_CONSOLE
	P4096	MMC_CONSOLE
	P4097	MMC_CONSOLE
	P4098	MMC_CONSOLE
	P4099	MMC_CONSOLE
C1E	P4100	MMC_CONSOLE
	P4101	MMC_CONSOLE
	P4102	MMC_CONSOLE
	P4103	MMC_CONSOLE
	P4104	MMC_CONSOLE
	P4105	MMC_CONSOLE
	P4106	MMC_CONSOLE
	P4107	MMC_CONSOLE
	P4108	MMC_CONSOLE
	P4109	MMC_CONSOLE

Section	Pin	Function
C1A	P4110	MMC_CONSOLE
	P4111	MMC_CONSOLE
	P4112	MMC_CONSOLE
	P4113	MMC_CONSOLE
	P4114	MMC_CONSOLE
	P4115	MMC_CONSOLE
	P4116	MMC_CONSOLE
	P4117	MMC_CONSOLE
	P4118	MMC_CONSOLE
	P4119	MMC_CONSOLE
C1B	P4120	MMC_CONSOLE
	P4121	MMC_CONSOLE
	P4122	MMC_CONSOLE
	P4123	MMC_CONSOLE
	P4124	MMC_CONSOLE
	P4125	MMC_CONSOLE
	P4126	MMC_CONSOLE
	P4127	MMC_CONSOLE
	P4128	MMC_CONSOLE
	P4129	MMC_CONSOLE
C1C	P4130	MMC_CONSOLE
	P4131	MMC_CONSOLE
	P4132	MMC_CONSOLE
	P4133	MMC_CONSOLE
	P4134	MMC_CONSOLE
	P4135	MMC_CONSOLE
	P4136	MMC_CONSOLE
	P4137	MMC_CONSOLE
	P4138	MMC_CONSOLE
	P4139	MMC_CONSOLE
C1D	P4140	MMC_CONSOLE
	P4141	MMC_CONSOLE
	P4142	MMC_CONSOLE
	P4143	MMC_CONSOLE
	P4144	MMC_CONSOLE
	P4145	MMC_CONSOLE
	P4146	MMC_CONSOLE
	P4147	MMC_CONSOLE
	P4148	MMC_CONSOLE
	P4149	MMC_CONSOLE
C1E	P4150	MMC_CONSOLE
	P4151	MMC_CONSOLE
	P4152	MMC_CONSOLE
	P4153	MMC_CONSOLE
	P4154	MMC_CONSOLE
	P4155	MMC_CONSOLE
	P4156	MMC_CONSOLE
	P4157	MMC_CONSOLE
	P4158	MMC_CONSOLE
	P4159	MMC_CONSOLE

Section	Pin	Function
C1A	P4160	MMC_CONSOLE
	P4161	MMC_CONSOLE
	P4162	MMC_CONSOLE
	P4163	MMC_CONSOLE
	P4164	MMC_CONSOLE
	P4165	MMC_CONSOLE
	P4166	MMC_CONSOLE
	P4167	MMC_CONSOLE
	P4168	MMC_CONSOLE
	P4169	MMC_CONSOLE
C1B	P4170	MMC_CONSOLE
	P4171	MMC_CONSOLE
	P4172	MMC_CONSOLE
	P4173	MMC_CONSOLE
	P4174	MMC_CONSOLE
	P4175	MMC_CONSOLE
	P4176	MMC_CONSOLE
	P4177	MMC_CONSOLE
	P4178	MMC_CONSOLE
	P4179	MMC_CONSOLE
C1C	P4180	MMC_CONSOLE
	P4181	MMC_CONSOLE
	P4182	MMC_CONSOLE
	P4183	MMC_CONSOLE
	P4184	MMC_CONSOLE
	P4185	MMC_CONSOLE
	P4186	MMC_CONSOLE
	P4187	MMC_CONSOLE
	P4188	MMC_CONSOLE
	P4189	MMC_CONSOLE
C1D	P4190	MMC_CONSOLE
	P4191	MMC_CONSOLE
	P4192	MMC_CONSOLE
	P4193	MMC_CONSOLE
	P4194	MMC_CONSOLE
	P4195	MMC_CONSOLE
	P4196	MMC_CONSOLE
	P4197	MMC_CONSOLE
	P4198	MMC_CONSOLE
	P4199	MMC_CONSOLE
C1E	P4200	MMC_CONSOLE
	P4201	MMC_CONSOLE
	P4202	MMC_CONSOLE
	P4203	MMC_CONSOLE
	P4204	MMC_CONSOLE
	P4205	MMC_CONSOLE
	P4206	MMC_CONSOLE
	P4207	MMC_CONSOLE
	P4208	MMC_CONSOLE
	P4209	MMC_CONSOLE

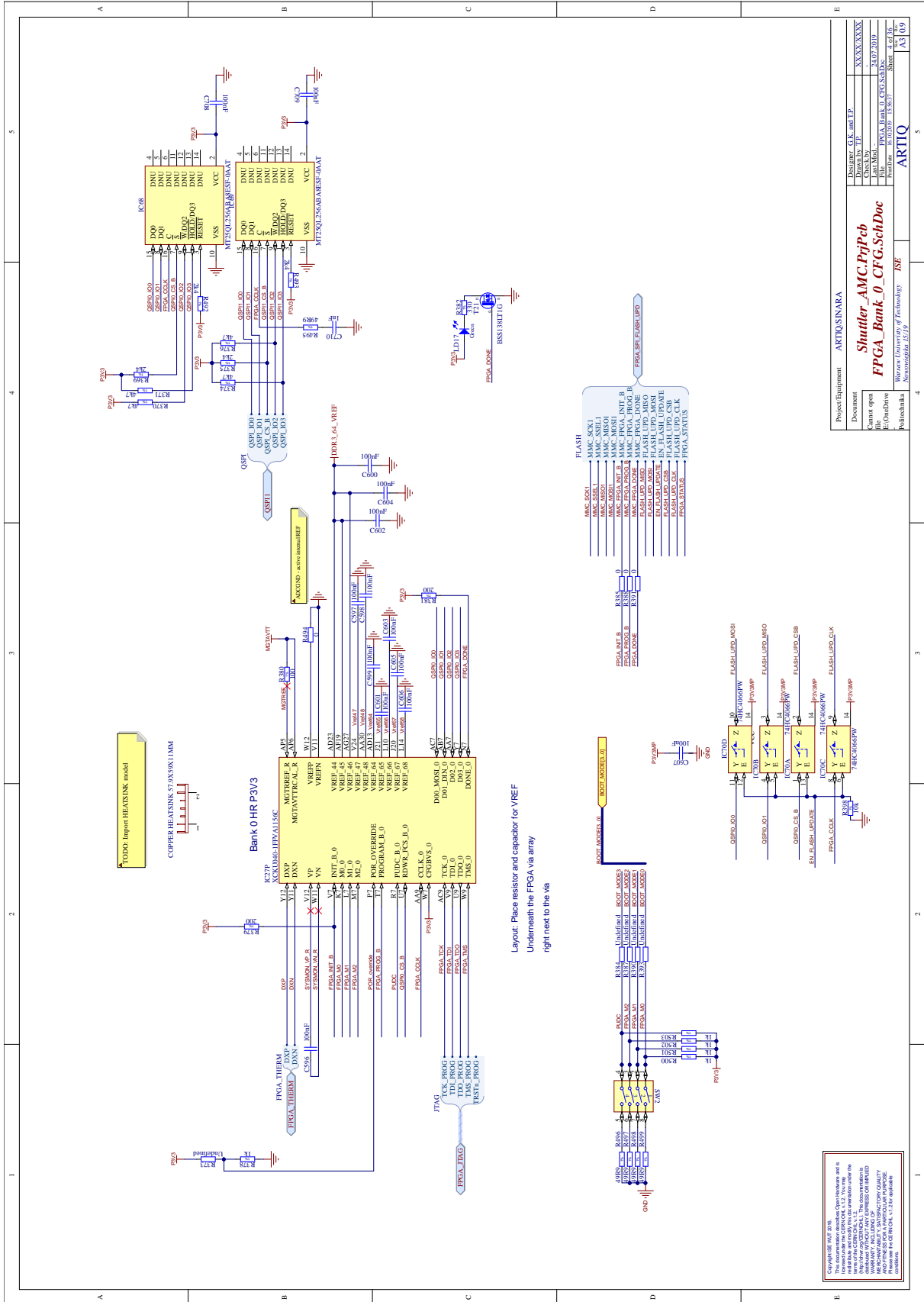
Section	Pin	Function
C1A	P4210	MMC_CONSOLE
	P4211	MMC_CONSOLE
	P4212	MMC_CONSOLE
	P4213	MMC_CONSOLE
	P4214	MMC_CONSOLE
	P4215	MMC_CONSOLE
	P4216	MMC_CONSOLE
	P4217	MMC_CONSOLE
	P4218	MMC_CONSOLE
	P4219	MMC_CONSOLE
C1B	P4220	MMC_CONSOLE
	P4221	MMC_CONSOLE
	P4222	MMC_CONSOLE
	P4223	MMC_CONSOLE
	P4224	MMC_CONSOLE
	P4225	MMC_CONSOLE
	P4226	MMC_CONSOLE
	P4227	MMC_CONSOLE
	P4228	MMC_CONSOLE
	P4229	MMC_CONSOLE
C1C	P4230	MMC_CONSOLE
	P4231	MMC_CONSOLE
	P4232	MMC_CONSOLE
	P4233	MMC_CONSOLE
	P4234	MMC_CONSOLE
	P4235	MMC_CONSOLE
	P4236	MMC_CONSOLE
	P4237	MMC_CONSOLE
	P4238	MMC_CONSOLE
	P4239	MMC_CONSOLE
C1D	P4240	MMC_CONSOLE
	P4241	MMC_CONSOLE
	P4242	MMC_CONSOLE
	P4243	MMC_CONSOLE
	P4244	MMC_CONSOLE
	P4245	MMC_CONSOLE
	P4246	MMC_CONSOLE
	P4247	MMC_CONSOLE
	P4248	MMC_CONSOLE
	P4249	MMC_CONSOLE
C1E	P4250	MMC_CONSOLE
	P4251	MMC_CONSOLE
	P4252	MMC_CONSOLE
	P4253	MMC_CONSOLE
	P4254	MMC_CONSOLE
	P4255	MMC_CONSOLE
	P4256	MMC_CONSOLE
	P4257	MMC_CONSOLE
	P4258	MMC_CONSOLE
	P4259	MMC_CONSOLE

Section	Pin	Function
C1A	P4260	MMC_CONSOLE
	P4261	MMC_CONSOLE
	P4262	MMC_CONSOLE
	P4263	MMC_CONSOLE
	P4264	MMC_CONSOLE
	P4265	MMC_CONSOLE
	P4266	MMC_CONSOLE
	P4267	MMC_CONSOLE
	P4268	MMC_CONSOLE
	P4269	MMC_CONSOLE
C1B	P4270	MMC_CONSOLE
	P4271	MMC_CONSOLE
	P4272	MMC_CONSOLE
	P4273	MMC_CONSOLE
	P4274	MMC_CONSOLE
	P4275	MMC_CONSOLE
	P4276	MMC_CONSOLE
	P4277	MMC_CONSOLE
	P4278	MMC_CONSOLE
	P4279	MMC_CONSOLE
C1C	P4280	MMC_CONSOLE
	P4281	MMC_CONSOLE
	P4282	MMC_CONSOLE
	P4283	MMC_CONSOLE
	P4284	MMC_CONSOLE
	P4285	MMC_CONSOLE
	P4286	MMC_CONSOLE
	P4287	MMC_CONSOLE
	P4288	MMC_CONSOLE
	P4289	MMC_CONSOLE
C1D	P4290	MMC_CONSOLE
	P4291	MMC_CONSOLE
	P4292	MMC_CONSOLE
	P4293	MMC_CONSOLE
	P4294	MMC_CONSOLE
	P4295	MMC_CONSOLE
	P4296	MMC_CONSOLE
	P4297	MMC_CONSOLE
	P4298	MMC_CONSOLE
	P4299	MMC_CONSOLE
C1E	P4300	MMC_CONSOLE
	P4301	MMC_CONSOLE
	P4302	MMC_CONSOLE
	P4303	MMC_CONSOLE
	P4304	MMC_CONSOLE
	P4305	MMC_CONSOLE
	P4306	MMC_CONSOLE
	P4307	MMC_CONSOLE
	P4308	MMC_CONSOLE
	P4309	MMC_CONSOLE

Section	Pin	Function
C1A	P4310	MMC_CONSOLE
	P4311	MMC_CONSOLE
	P4312	MMC_CONSOLE
	P4313	MMC_CONSOLE
	P4314	MMC_CONSOLE
	P4315	MMC_CONSOLE
	P4316	MMC_CONSOLE
	P4317	MMC_CONSOLE
	P4318	MMC_CONSOLE
	P4319	MMC_CONSOLE
C1B	P4320	MMC_CONSOLE
	P4321	MMC_CONSOLE
	P4322	MMC_CONSOLE
	P4323	MMC_CONSOLE
	P4324	MMC_CONSOLE
	P4325	MMC_CONSOLE
	P4326	MMC_CONSOLE
	P4327	MMC_CONSOLE
	P4328	MMC_CONSOLE
	P4329	MMC_CONSOLE
C1C	P4330	MMC_CONSOLE
	P4331	MMC_CONSOLE
	P4332	MMC_CONSOLE
	P4333	MMC_CONSOLE
	P4334	MMC_CONSOLE
	P4335	MMC_CONSOLE
	P4336	MMC_CONSOLE
	P4337	MMC_CONSOLE
	P4338	MMC_CONSOLE
	P4339	MMC_CONSOLE
C1D	P4340	MMC_CONSOLE
	P4341	MMC_CONSOLE
	P4342	MMC_CONSOLE
	P4343	MMC_CONSOLE
	P4344	MMC_CONSOLE
	P4345	MMC_CONSOLE
	P4346	MMC_CONSOLE
	P4347	MMC_CONSOLE
	P4348	MMC_CONSOLE
	P4349	MMC_CONSOLE
C1E	P4350	MMC_CONSOLE
	P4351	MMC_CONSOLE
	P4352	MMC_CONSOLE
	P4353	MMC_CONSOLE
	P4354	MMC_CONSOLE
	P4355	MMC_CONSOLE
	P4356	MMC_CONSOLE
	P4357	MMC_CONSOLE
	P4358	MMC_CONSOLE
	P4359	MMC_CONSOLE

Section	Pin	Function
C1A	P4360	MMC_CONSOLE
	P4361	MMC_CONSOLE
	P4362	MMC_CONSOLE
	P4363	MMC_CONSOLE
	P4364	MMC_CONSOLE
	P4365	MMC_CONSOLE
	P4366	MMC_CONSOLE
	P4367	MMC_CONSOLE
	P4368	MMC_CONSOLE
	P4369	MMC_CONSOLE
C1B	P4370	MMC_CONSOLE
	P4371	MMC_CONSOLE
	P4372	MMC_CONSOLE
	P4373	MMC_CONSOLE
	P4374	MMC_CONSOLE
	P4375	MMC_CONSOLE
	P4376	MMC_CONSOLE
	P4377	MMC_CONSOLE
	P4378	MMC_CONSOLE
	P4379	MMC_CONSOLE
C1C	P4380	MMC_CONSOLE
	P4381	MMC_CONSOLE
	P4382	MMC_CONSOLE
	P4383	MMC_CONSOLE
	P4384	MMC_CONSOLE
	P4385	MMC_CONSOLE
	P4386	MMC_CONSOLE
	P4387	MMC_CONSOLE
	P4388	MMC_CONSOLE
	P4389	MMC_CONSOLE
C1D	P4390	MMC_CONSOLE
	P4391	MMC_CONSOLE
	P4392	MMC_CONSOLE
	P4393	MMC_CONSOLE
	P4394	MMC_CONSOLE
	P4395	MMC_CONSOLE
	P4396	MMC_CONSOLE
	P4397	MMC_CONSOLE
	P4398	MMC_CONSOLE
	P4399	MMC_CONSOLE
C1E	P4400	MMC_CONSOLE
	P4401	MMC_CONSOLE
	P4402	MMC_CONSOLE
	P4403	MMC_CONSOLE
	P4404	MMC_CONSOLE
	P4405	MMC_CONSOLE
	P4406	MMC_CONSOLE
	P4407	MMC_CONSOLE
	P4408	MMC_CONSOLE
	P4409	MMC_CONSOLE

Section	Pin	Function
C1A	P4410	MMC_CONSOLE
	P4411	MMC_CONSOLE
	P4412	MMC_CONSOLE
	P4413	MMC_CONSOLE
	P4414	MMC_CONSOLE
	P4415	MMC_CONSOLE
	P4416	MMC_CONSOLE
	P4417	MMC_CONSOLE
	P4418	MMC_CONSOLE
	P4419	MMC_CONSOLE
C1B	P4420	MMC_CONSOLE
	P4421	MMC_CONSOLE
	P4422	MMC_CONSOLE
	P4423	MMC_CONSOLE
	P4424	MMC_CONSOLE
	P4425	MMC_CONSOLE
	P4426	MMC_CONSOLE
	P4427	MMC_CONSOLE
	P4428	MMC_CONSOLE
	P4429	MMC_CONSOLE
C1C	P4430	MMC_CONSOLE
	P4431	MMC_CONSOLE
	P4432	MMC_CONSOLE
	P4433	MMC_CONSOLE
	P4434	MMC_CONSOLE



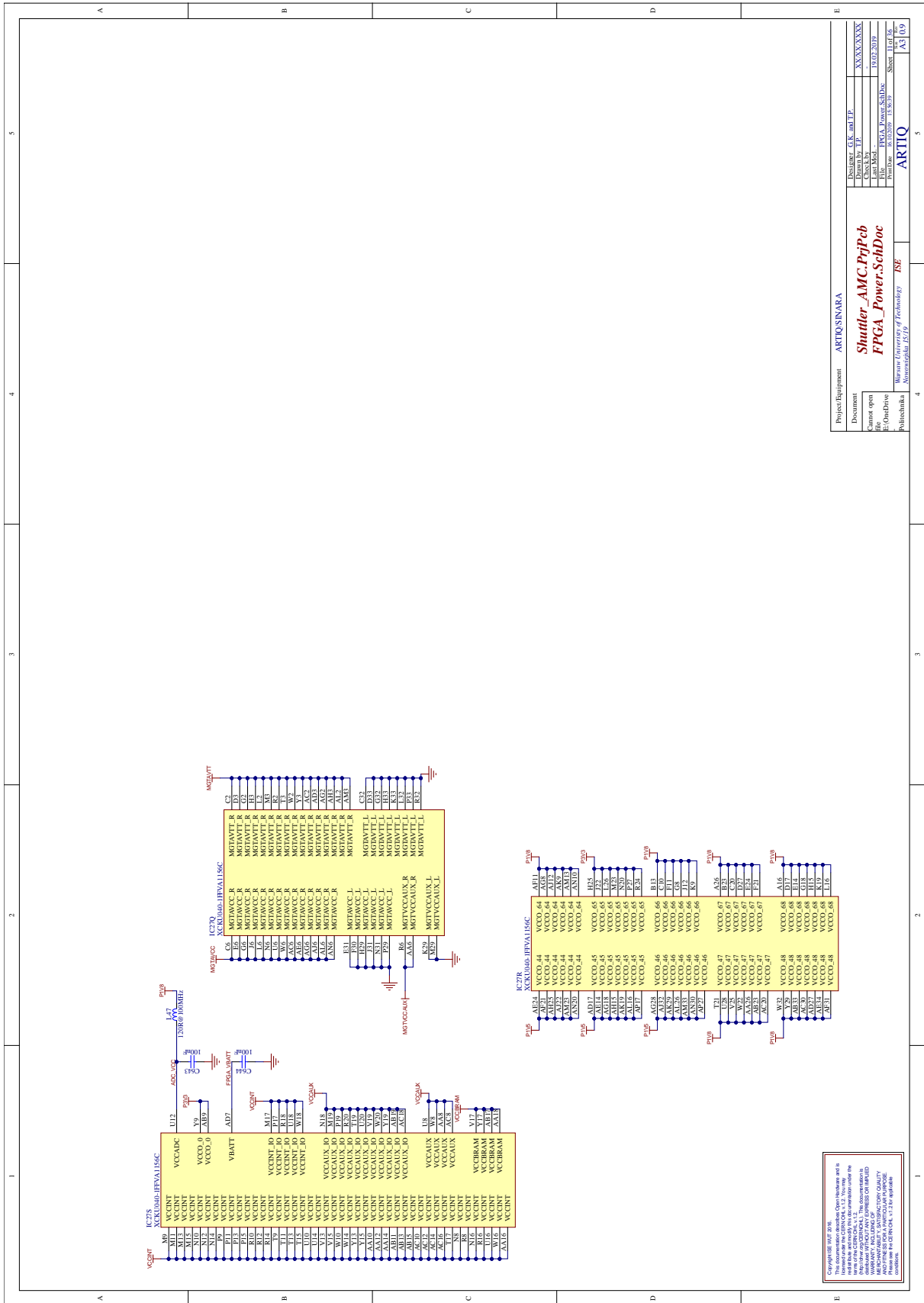
Layout: Please resistor and capacitor for VREF
Underneath the FPGA via array
right next to the via

Bank 68 HP PIV8

IC271
XKCRUB01HPVA156C

BANK 68

18	IO.L24P.TU.N0.66	DACS IN DACS IN	DACS IN DACS IN
19	IO.L24P.TU.N1.67		
20	IO.L24P.TU.N2.68		
21	IO.L24P.TU.N3.69		
22	IO.L24P.TU.N4.70		
23	IO.L24P.TU.N5.71		
24	IO.L24P.TU.N6.72		
25	IO.L24P.TU.N7.73		
26	IO.L24P.TU.N8.74		
27	IO.L24P.TU.N9.75		
28	IO.L24P.TU.N0.76		
29	IO.L24P.TU.N1.77		
30	IO.L24P.TU.N2.78		
31	IO.L24P.TU.N3.79		
32	IO.L24P.TU.N4.80		
33	IO.L24P.TU.N5.81		
34	IO.L24P.TU.N6.82		
35	IO.L24P.TU.N7.83		
36	IO.L24P.TU.N8.84		
37	IO.L24P.TU.N9.85		
38	IO.L24P.TU.N0.86		
39	IO.L24P.TU.N1.87		
40	IO.L24P.TU.N2.88		
41	IO.L24P.TU.N3.89		
42	IO.L24P.TU.N4.90		
43	IO.L24P.TU.N5.91		
44	IO.L24P.TU.N6.92		
45	IO.L24P.TU.N7.93		
46	IO.L24P.TU.N8.94		
47	IO.L24P.TU.N9.95		
48	IO.L24P.TU.N0.96		
49	IO.L24P.TU.N1.97		
50	IO.L24P.TU.N2.98		
51	IO.L24P.TU.N3.99		
52	IO.L24P.TU.N4.00		
53	IO.L24P.TU.N5.01		
54	IO.L24P.TU.N6.02		
55	IO.L24P.TU.N7.03		
56	IO.L24P.TU.N8.04		
57	IO.L24P.TU.N9.05		
58	IO.L24P.TU.N0.06		
59	IO.L24P.TU.N1.07		
60	IO.L24P.TU.N2.08		
61	IO.L24P.TU.N3.09		
62	IO.L24P.TU.N4.10		
63	IO.L24P.TU.N5.11		
64	IO.L24P.TU.N6.12		
65	IO.L24P.TU.N7.13		
66	IO.L24P.TU.N8.14		
67	IO.L24P.TU.N9.15		
68	IO.L24P.TU.N0.16		
69	IO.L24P.TU.N1.17		
70	IO.L24P.TU.N2.18		
71	IO.L24P.TU.N3.19		
72	IO.L24P.TU.N4.20		
73	IO.L24P.TU.N5.21		
74	IO.L24P.TU.N6.22		
75	IO.L24P.TU.N7.23		
76	IO.L24P.TU.N8.24		
77	IO.L24P.TU.N9.25		
78	IO.L24P.TU.N0.26		
79	IO.L24P.TU.N1.27		
80	IO.L24P.TU.N2.28		
81	IO.L24P.TU.N3.29		
82	IO.L24P.TU.N4.30		
83	IO.L24P.TU.N5.31		
84	IO.L24P.TU.N6.32		
85	IO.L24P.TU.N7.33		
86	IO.L24P.TU.N8.34		
87	IO.L24P.TU.N9.35		
88	IO.L24P.TU.N0.36		
89	IO.L24P.TU.N1.37		
90	IO.L24P.TU.N2.38		
91	IO.L24P.TU.N3.39		
92	IO.L24P.TU.N4.40		
93	IO.L24P.TU.N5.41		
94	IO.L24P.TU.N6.42		
95	IO.L24P.TU.N7.43		
96	IO.L24P.TU.N8.44		
97	IO.L24P.TU.N9.45		
98	IO.L24P.TU.N0.46		
99	IO.L24P.TU.N1.47		
100	IO.L24P.TU.N2.48		
101	IO.L24P.TU.N3.49		
102	IO.L24P.TU.N4.50		
103	IO.L24P.TU.N5.51		
104	IO.L24P.TU.N6.52		
105	IO.L24P.TU.N7.53		
106	IO.L24P.TU.N8.54		
107	IO.L24P.TU.N9.55		
108	IO.L24P.TU.N0.56		
109	IO.L24P.TU.N1.57		
110	IO.L24P.TU.N2.58		
111	IO.L24P.TU.N3.59		
112	IO.L24P.TU.N4.60		
113	IO.L24P.TU.N5.61		
114	IO.L24P.TU.N6.62		
115	IO.L24P.TU.N7.63		
116	IO.L24P.TU.N8.64		
117	IO.L24P.TU.N9.65		
118	IO.L24P.TU.N0.66		
119	IO.L24P.TU.N1.67		
120	IO.L24P.TU.N2.68		
121	IO.L24P.TU.N3.69		
122	IO.L24P.TU.N4.70		
123	IO.L24P.TU.N5.71		
124	IO.L24P.TU.N6.72		
125	IO.L24P.TU.N7.73		
126	IO.L24P.TU.N8.74		
127	IO.L24P.TU.N9.75		
128	IO.L24P.TU.N0.76		
129	IO.L24P.TU.N1.77		
130	IO.L24P.TU.N2.78		
131	IO.L24P.TU.N3.79		
132	IO.L24P.TU.N4.80		
133	IO.L24P.TU.N5.81		
134	IO.L24P.TU.N6.82		
135	IO.L24P.TU.N7.83		
136	IO.L24P.TU.N8.84		
137	IO.L24P.TU.N9.85		
138	IO.L24P.TU.N0.86		
139	IO.L24P.TU.N1.87		
140	IO.L24P.TU.N2.88		
141	IO.L24P.TU.N3.89		
142	IO.L24P.TU.N4.90		
143	IO.L24P.TU.N5.91		
144	IO.L24P.TU.N6.92		
145	IO.L24P.TU.N7.93		
146	IO.L24P.TU.N8.94		
147	IO.L24P.TU.N9.95		
148	IO.L24P.TU.N0.96		
149	IO.L24P.TU.N1.97		
150	IO.L24P.TU.N2.98		
151	IO.L24P.TU.N3.99		
152	IO.L24P.TU.N4.00		
153	IO.L24P.TU.N5.01		
154	IO.L24P.TU.N6.02		
155	IO.L24P.TU.N7.03		
156	IO.L24P.TU.N8.04		
157	IO.L24P.TU.N9.05		
158	IO.L24P.TU.N0.06		
159	IO.L24P.TU.N1.07		
160	IO.L24P.TU.N2.08		
161	IO.L24P.TU.N3.09		
162	IO.L24P.TU.N4.10		
163	IO.L24P.TU.N5.11		
164	IO.L24P.TU.N6.12		
165	IO.L24P.TU.N7.13		
166	IO.L24P.TU.N8.14		
167	IO.L24P.TU.N9.15		
168	IO.L24P.TU.N0.16		
169	IO.L24P.TU.N1.17		
170	IO.L24P.TU.N2.18		
171	IO.L24P.TU.N3.19		
172	IO.L24P.TU.N4.20		
173	IO.L24P.TU.N5.21		
174	IO.L24P.TU.N6.22		
175	IO.L24P.TU.N7.23		
176	IO.L24P.TU.N8.24		
177	IO.L24P.TU.N9.25		
178	IO.L24P.TU.N0.26		
179	IO.L24P.TU.N1.27		
180	IO.L24P.TU.N2.28		
181	IO.L24P.TU.N3.29		
182	IO.L24P.TU.N4.30		
183	IO.L24P.TU.N5.31		
184	IO.L24P.TU.N6.32		
185	IO.L24P.TU.N7.33		
186	IO.L24P.TU.N8.34		
187	IO.L24P.TU.N9.35		
188	IO.L24P.TU.N0.36		
189	IO.L24P.TU.N1.37		
190	IO.L24P.TU.N2.38		
191	IO.L24P.TU.N3.39		
192	IO.L24P.TU.N4.40		
193	IO.L24P.TU.N5.41		
194	IO.L24P.TU.N6.42		
195	IO.L24P.TU.N7.43		
196	IO.L24P.TU.N8.44		
197	IO.L24P.TU.N9.45		
198	IO.L24P.TU.N0.46		
199	IO.L24P.TU.N1.47		
200	IO.L24P.TU.N2.48		
201	IO.L24P.TU.N3.49		
202	IO.L24P.TU.N4.50		
203	IO.L24P.TU.N5.51		
204	IO.L24P.TU.N6.52		
205	IO.L24P.TU.N7.53		
206	IO.L24P.TU.N8.54		
207	IO.L24P.TU.N9.55		
208	IO.L24P.TU.N0.56		
209	IO.L24P.TU.N1.57		
210	IO.L24P.TU.N2.58		
211	IO.L24P.TU.N3.59		
212	IO.L24P.TU.N4.60		
213	IO.L24P.TU.N5.61		
214	IO.L24P.TU.N6.62		
215	IO.L24P.TU.N7.63		
216	IO.L24P.TU.N8.64		
217	IO.L24P.TU.N9.65		
218	IO.L24P.TU.N0.66		
219	IO.L24P.TU.N1.67		
220	IO.L24P.TU.N2.68		
221	IO.L24P.TU.N3.69		
222	IO.L24P.TU.N4.70		
223	IO.L24P.TU.N5.71		
224	IO.L24P.TU.N6.72		
225	IO.L24P.TU.N7.73		
226	IO.L24P.TU.N8.74		
227	IO.L24P.TU.N9.75		
228	IO.L24P.TU.N0.76		
229	IO.L24P.TU.N1.77		
230	IO.L24P.TU.N2.78		
231	IO.L24P.TU.N3.79		
232	IO.L24P.TU.N4.80		
233	IO.L24P.TU.N5.81		
234	IO.L24P.TU.N6.82		
235	IO.L24P.TU.N7.83		
236	IO.L24P.TU.N8.84		
237	IO.L24P.TU.N9.85		
238	IO.L24P.TU.N0.86		
239	IO.L24P.TU.N1.87		
240	IO.L24P.TU.N2.88		
241	IO.L24P.TU.N3.89		
242	IO.L24P.TU.N4.90		
243	IO.L24P.TU.N5.91		
244	IO.L24P.TU.N6.92		
245	IO.L24P.TU.N7.93		
246	IO.L24P.TU.N8.94		
247	IO.L24P.TU.N9.95		
248	IO.L24P.TU.N0.96		
249	IO.L24P.TU.N1.97		
250	IO.L24P.TU.N2.98		
251	IO.L24P.TU.N3.99		
252	IO.L24P.TU.N4.00		
253	IO.L24P.TU.N5.01		
254	IO.L24P.TU.N6.02		
255	IO.L24P.TU.N7.03		
256	IO.L24P.TU.N8.04		
257	IO.L24P.TU.N9.05		
258	IO.L24P.TU.N0.06		
259	IO.L24P.TU.N1.07		
260	IO.L24P.TU.N2.08		
261	IO.L24P.TU.N3.09		
262	IO.L24P.TU.N4.10		
263	IO.L24P.TU.N5.11		
264	IO.L24P.TU.N6.12		
265	IO.L24P.TU.N7.13		
266	IO.L24P.TU.N8.14		
267	IO.L24P.TU.N9.15		
268	IO.L24P.TU.N0.16		
269	IO.L24P.TU.N1.17		
270	IO.L24P.TU.N2.18		
271	IO.L24P.TU.N3.19		
272	IO.L24P.TU.N4.20		
273	IO.L24P.TU.N5.21		
274	IO.L24P.TU.N6.22		
275	IO.L24P.TU.N7.23		
276	IO.L24P.TU.N8.24		
277	IO.L24P.TU.N9.25		
278	IO.L24P.TU.N0.26		
279	IO.L24P.TU.N1.27		
280	IO.L24P.TU.N2.28		
281	IO.L24P.TU.N3.29		
282	IO.L24P.TU.N4.30		
283	IO.L24P.TU.N5.31		
284	IO.L24P.TU.N6.32		
285	IO.L24P.TU.N7.33		
286	IO.L24P.TU.N8.34		
287	IO.L24P.TU.N9.35		
288	IO.L24P.TU.N0.36		
289	IO.L24P.TU.N1.37		
290	IO.L24P.TU.N2.38		
291	IO.L24P.TU.N3.39		
292	IO.L24P.TU.N4.40		
293	IO.L24P.TU.N5.41		
294	IO.L24P.TU.N6.42		
295	IO.L24P.TU.N7.43		
296	IO.L24P.TU.N8.44		
297	IO.L24P.TU.N9.45		
298	IO.L24P.TU.N0.46		
299	IO.L24P.TU.N1.47		
300	IO.L24P.TU.N2.48		
301	IO.L24P.TU.N3.49		
302	IO.L24P.TU.N4.50		
303	IO.L24P.TU.N5.51		
304	IO.L24P.TU.N6.52		
305	IO.L24P.TU.N7.53		
306	IO.L24P.TU.N8.54		
307	IO.L24P.TU.N9.55		
308	IO.L24P.TU.N0.56		
309	IO.L24P.TU.N1.57		
310	IO.L24P.TU.N2.58		
311	IO.L24P.TU.N3.59		
312	IO.L24P.TU.N4.60		
313	IO.L24P.TU.N5.61		
314	IO.L24P.TU.N6.62		
315	IO.L24P.TU.N7.63		
316	IO.L24P.TU.N8.64		
317	IO.L24P.TU.N9.65		
318	IO.L24P.TU.N0.66		
319	IO.L24P.TU.N1.67		
320	IO.L24P.TU.N2.68		
321	IO.L24P.TU.N3.69		
322	IO.L24P.TU.N4.70		
323	IO.L24P.TU.N5.71		



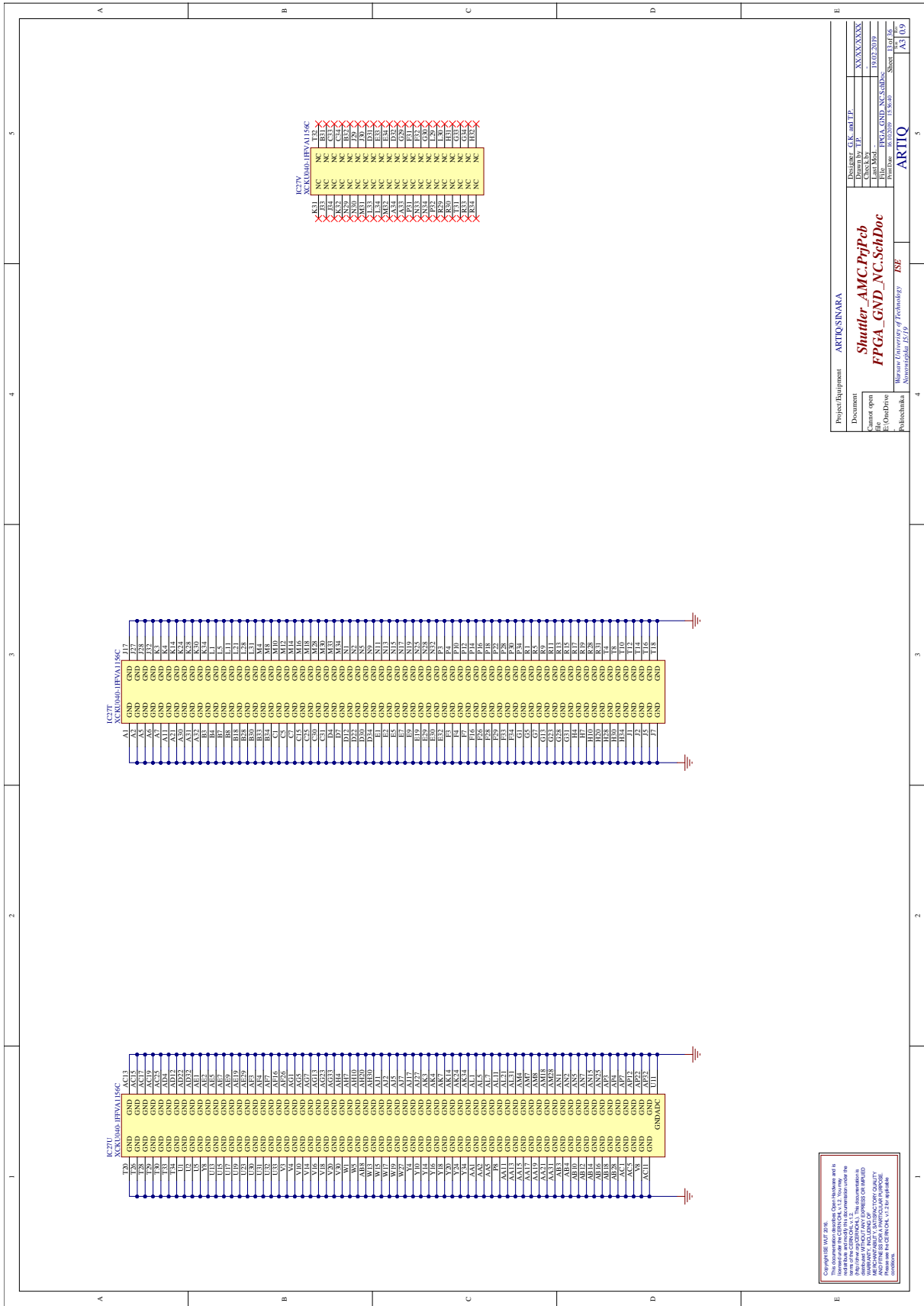
Copyright © 2018 by Shuttler AMC. All rights reserved.
 This document is the property of Shuttler AMC. It is intended for use only for the project and system for which it was prepared. It is not to be distributed, copied, or used for any other purpose without the written permission of Shuttler AMC. The information in this document is provided as a service to our customers and is not intended to constitute an offer of any financial product or service. Shuttler AMC is not responsible for any errors or omissions in this document. For more information, please contact us at support@shuttler.com.

Project Equipment: ARTIQ/SINARA

Document: Shuttler AMC Pripob
 FPGAs, Power, SchDoc

Author: G.K. and I.P.
 Date: 01/02/2019
 File: PPGA_Power_SchDoc
 Path: C:\Users\G.K.\Desktop\ARTIQ_SINARA\ARTIQ_SINARA\ARTIQ_SINARA_SchDoc\ARTIQ_SINARA_SchDoc_Sheet_117.dwg
 Sheet: 117 of 120

Project: ARTIQ



Copyright © 2016, Intel Corporation. All rights reserved. Intel, the Intel logo, and other marks contained herein are trademarks of Intel Corporation or its subsidiaries in the United States and other countries. This document contains confidential information and is intended for use only by the individual named in this document. If you are not the named individual you should not disseminate, distribute, or copy this document. For more information on Intel's privacy policy, see the Privacy Policy on Intel's website. Intel reserves the right to change specifications without notice. © 2016 Intel Corporation. All rights reserved.

Project Equipment: ARTIQSINARA

Document: XXXXX-XXXX

Revision: 1.0

Created: 10/02/2019

File: FPGA_GND_NC_SchDoc

Path: C:\Users\user\Documents\ARTIQSINARA\1.0\PCB\1.0_SchDoc

Sheet: 15 of 16

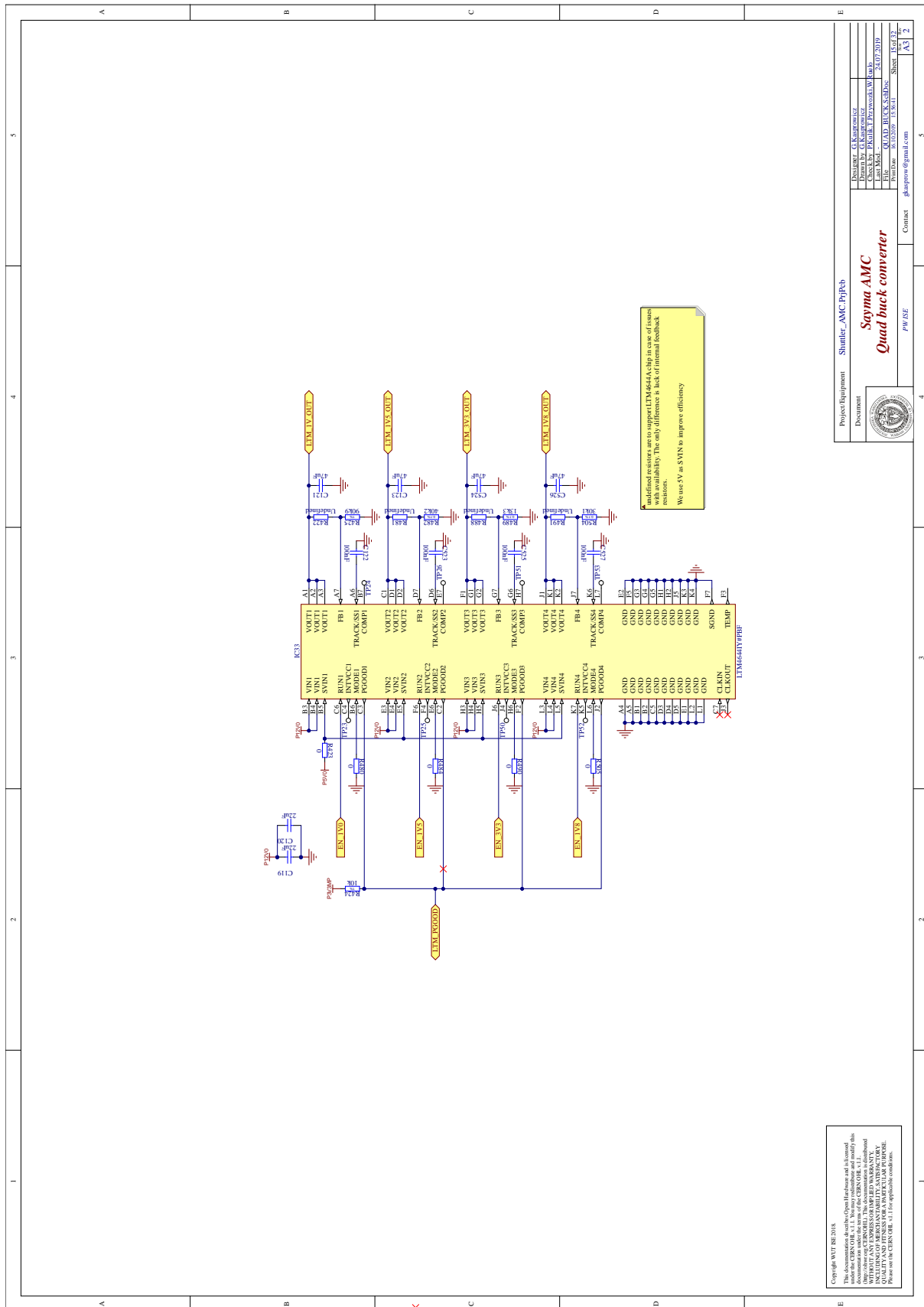
Photo: ARTIQ

Project: Shuttler AMC Pripob

Doc: FPGA_GND_NC_SchDoc

Author: Marwan University of Technology

Date: November 14, 2019



Modified resistors are to support LTM484A chip in case of issues with availability. The only difference is lack of internal feedback resistors.
We use 5V as SVMN to improve efficiency

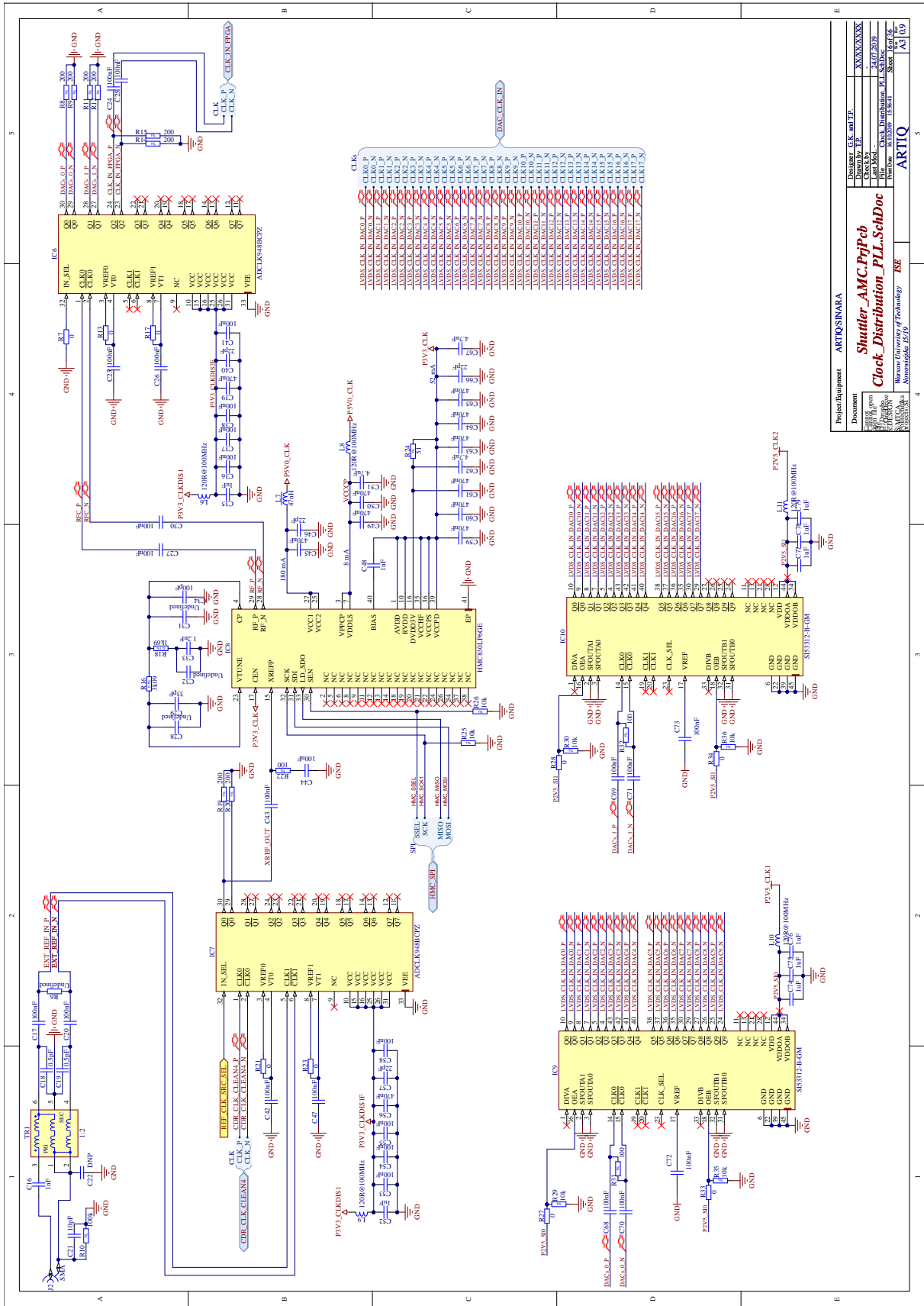
Copyright © 2016 Analog Devices, Inc. All rights reserved. This document is the property of Analog Devices, Inc. and its subsidiaries. It is intended for use only in conjunction with the products described herein. No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, without the prior written permission of Analog Devices, Inc.

Project/Equipment: **Shuller_AMC_PjPcb**

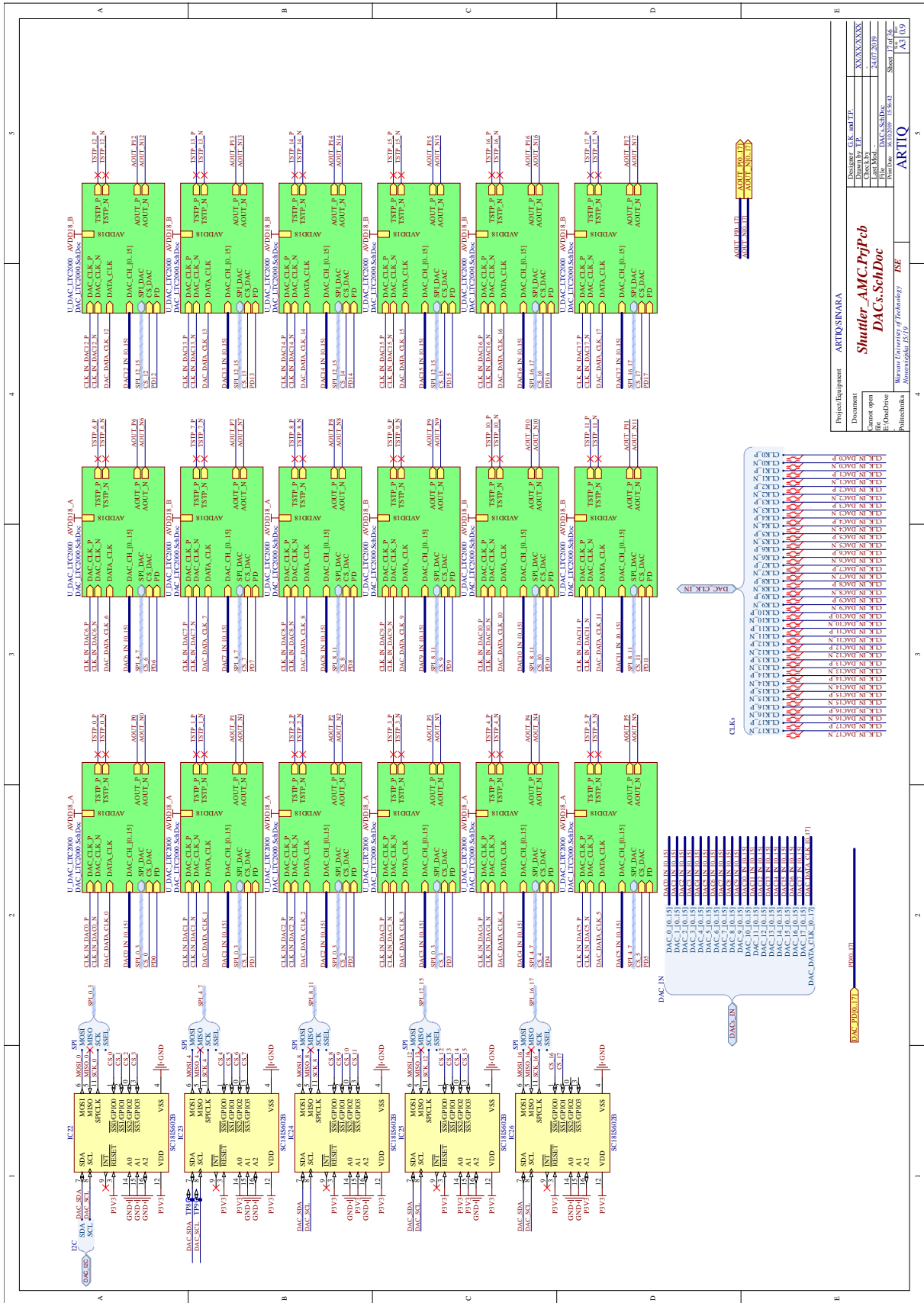
Document:

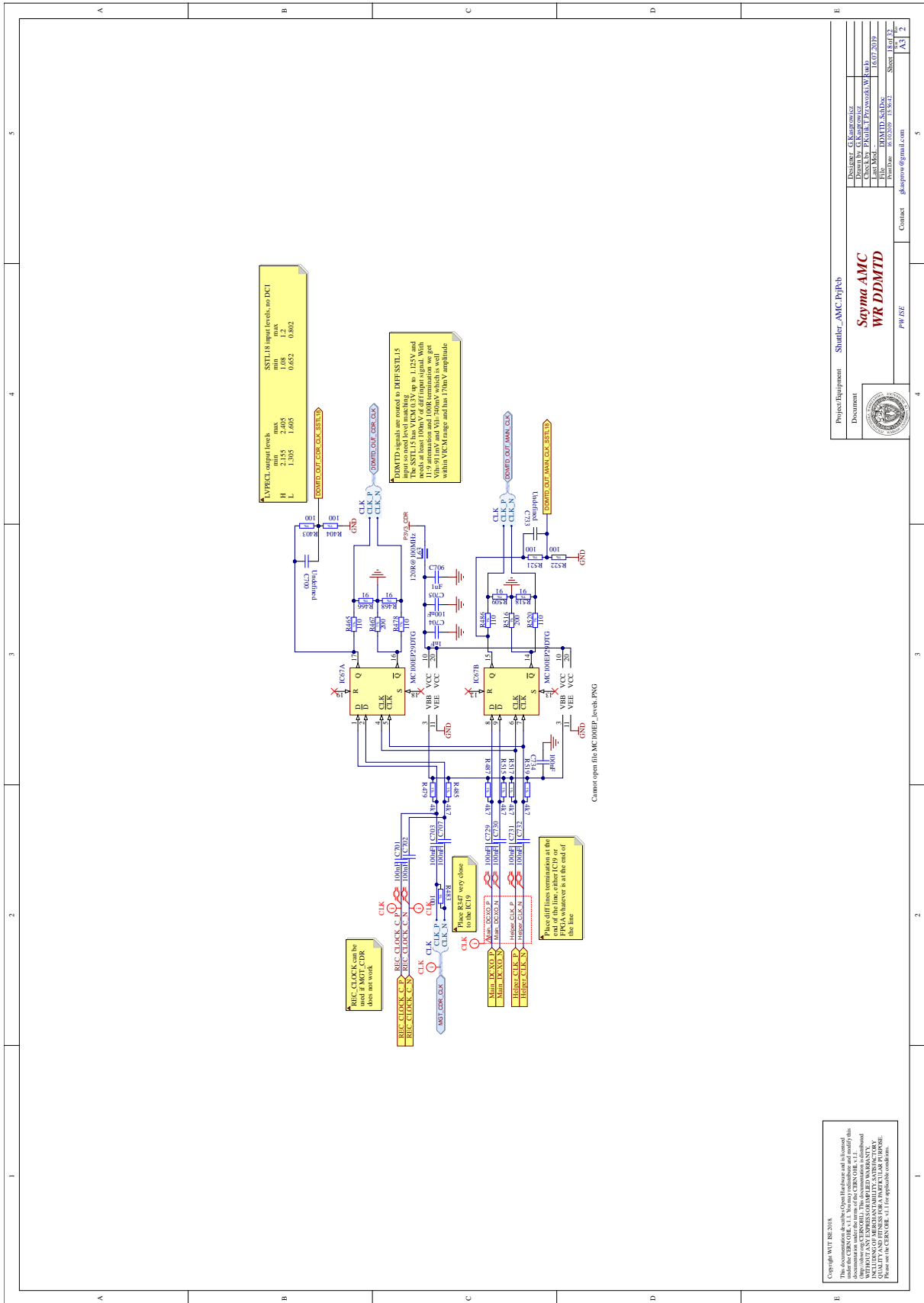
Sayma AMC
Quad buck converter

Drawn by: G.Karapınar
Checked by: G.Karapınar
Date: 24.07.2019
File: QUAD_BUCKS.HDoc
Title: QUAD_BUCKS.HDoc
Sheet: 1 of 3
Contact: gkarapin@analog.com



Project/Equipment	ARTIQ/SINARA
Document	XXXXXXXXXX
Drawn by	TP
Checked by	TP
Rev. Mod.	24.07.2019
File	Clock_Distribution_PLL_SchDoc
Version	1.0
Author	TP
Sheet	1 of 1
Project	ARTIQ
Organization	ISE
Warwick University of Technology	15/19
Sheet	1 of 1
Page	143 of 149





Copyright © 2018 STMicroelectronics
 This document contains confidential information of STMicroelectronics. It is intended for internal use only. No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of STMicroelectronics.

Project Equipment: **Shuller_AMC_PjPcb**

Document: **Sayma AMC WR DDMTD**

Author: G. Karpowicz

Drawn by: G. Karpowicz

Rev: 1.0

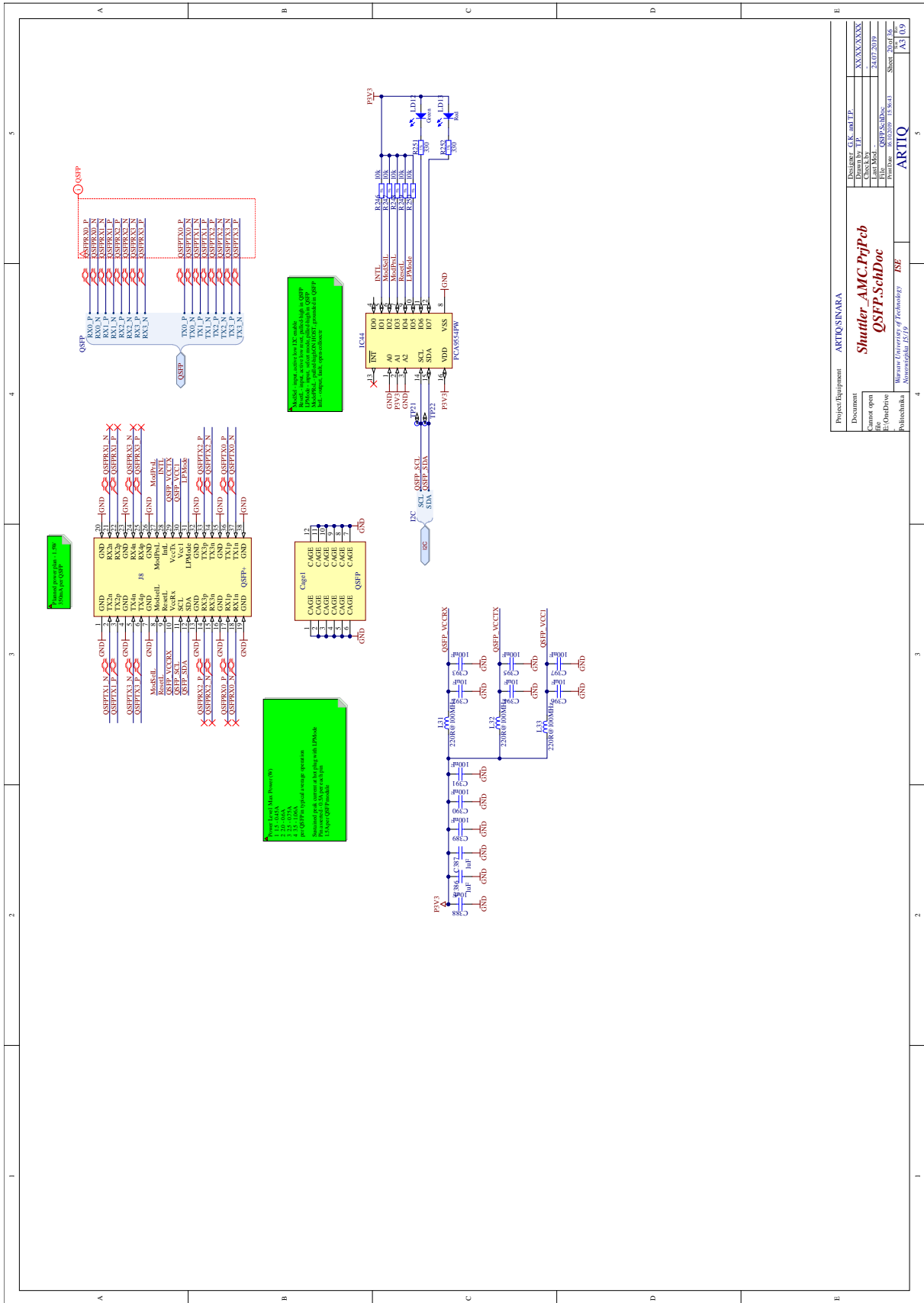
Rev Date: 16/07/2019

File: DDMTD_SchDoc

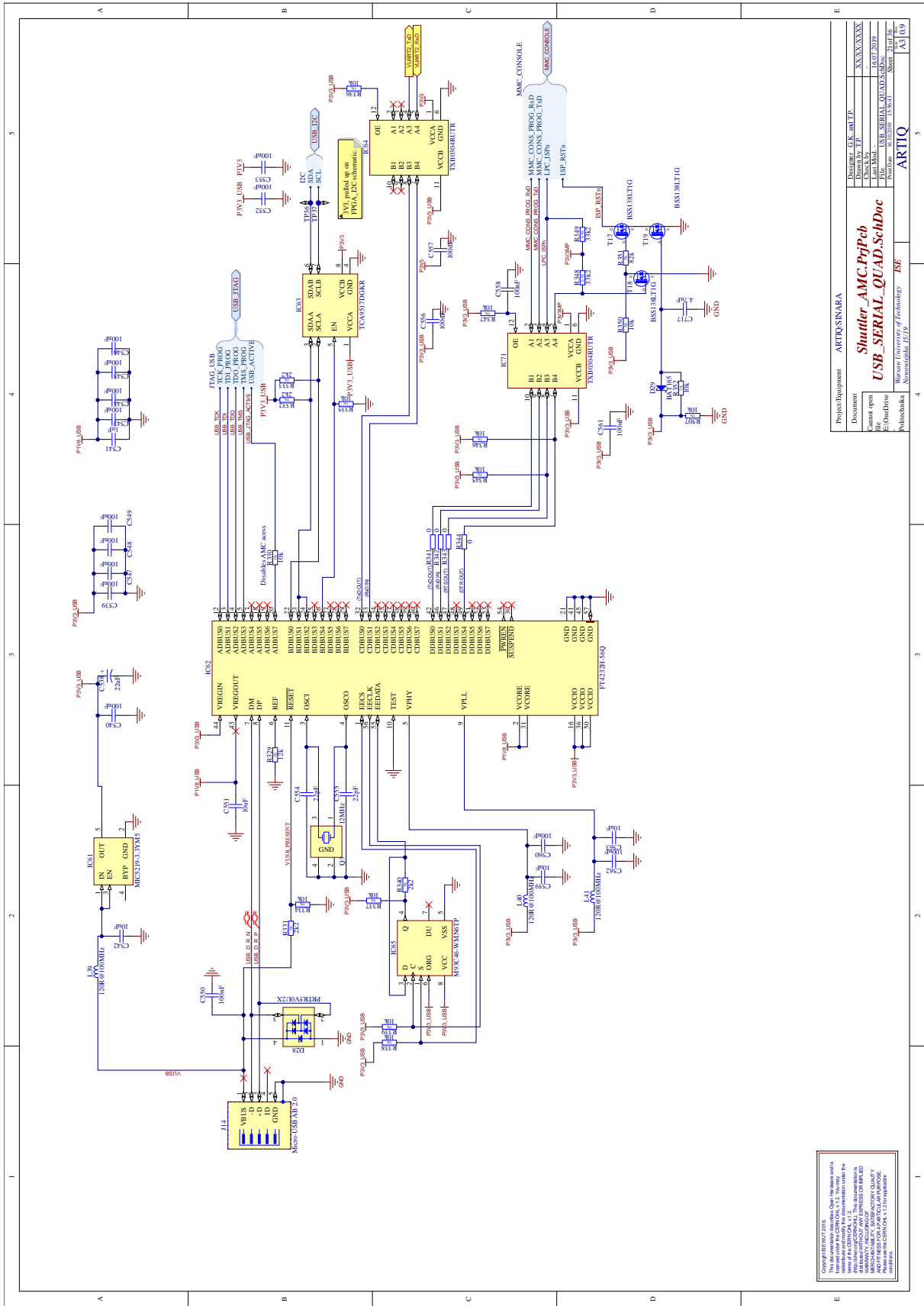
Print Date: 16/07/2019 11:52:11

Sheet: 05/03

Contact: g.karpowicz@st.com



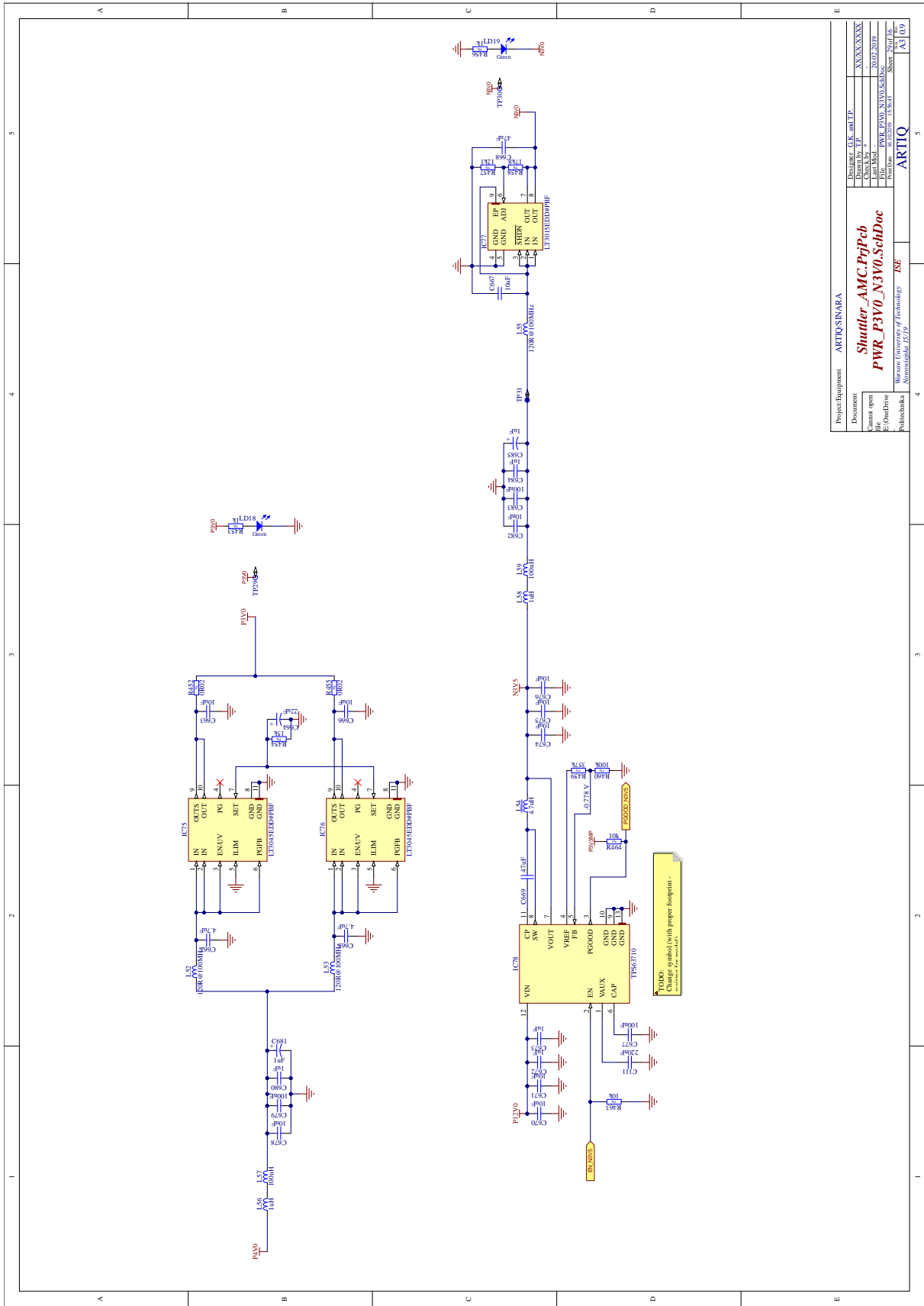
Project/Equipment: ARTIQ/SINARA		Designer: G.K. and LP	XXXX-XXXX
Document: QSPFP.SchDoc		Drawing: LP	24.07.2019
Cannot open: E:\QspDrive		File: QSPFP.SchDoc	Sheet: 04 of 04
Photo: 13.04.19		Author: ARTIQ	743 019
Photobank: November 1919		ISE	



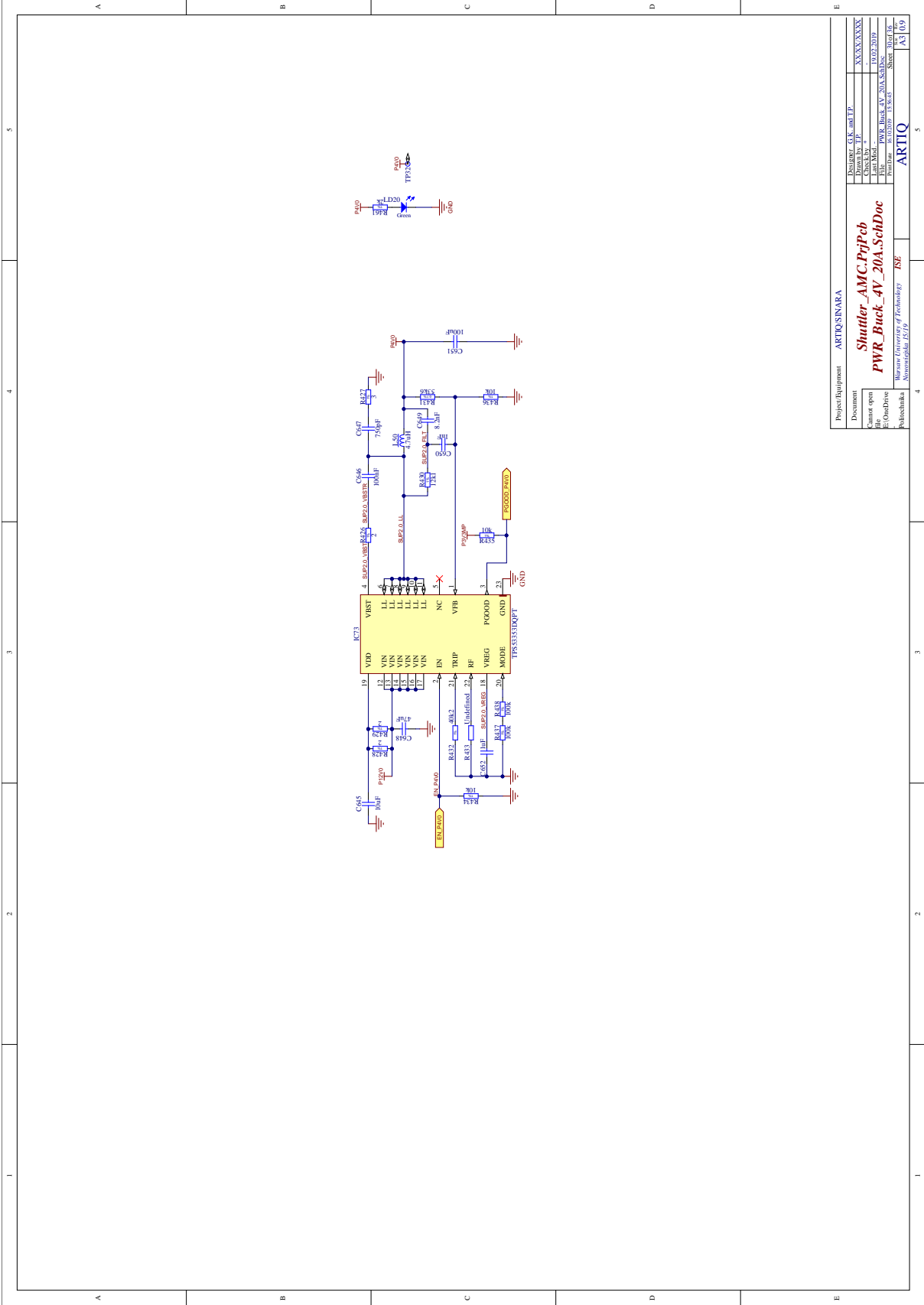
COMPONENTS LIST
 The components listed in this document are the property of the company and are not to be used in any other project without the written permission of the company.
 THE COMPANY MAKES NO WARRANTY, REPRESENTATION OR GUARANTEE OF ANY KIND, EXPRESS OR IMPLIED, REGARDING THE QUALITY, PERFORMANCE, RELIABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE OF ANY COMPONENTS LISTED IN THIS DOCUMENT.
 Please refer to the component manufacturer's data sheets for more information.

Project Equipment	ARTIOSINARA
Document	XXXXXXXXXX
Drawn by	XXXXXXXXXX
Checked by	XXXXXXXXXX
Date	10/07/2019
File	USB_SERIAL_QUAD.SchDoc
Path	XXXXXXXXXX
Sheet	1 of 1
Sheet	1 of 1

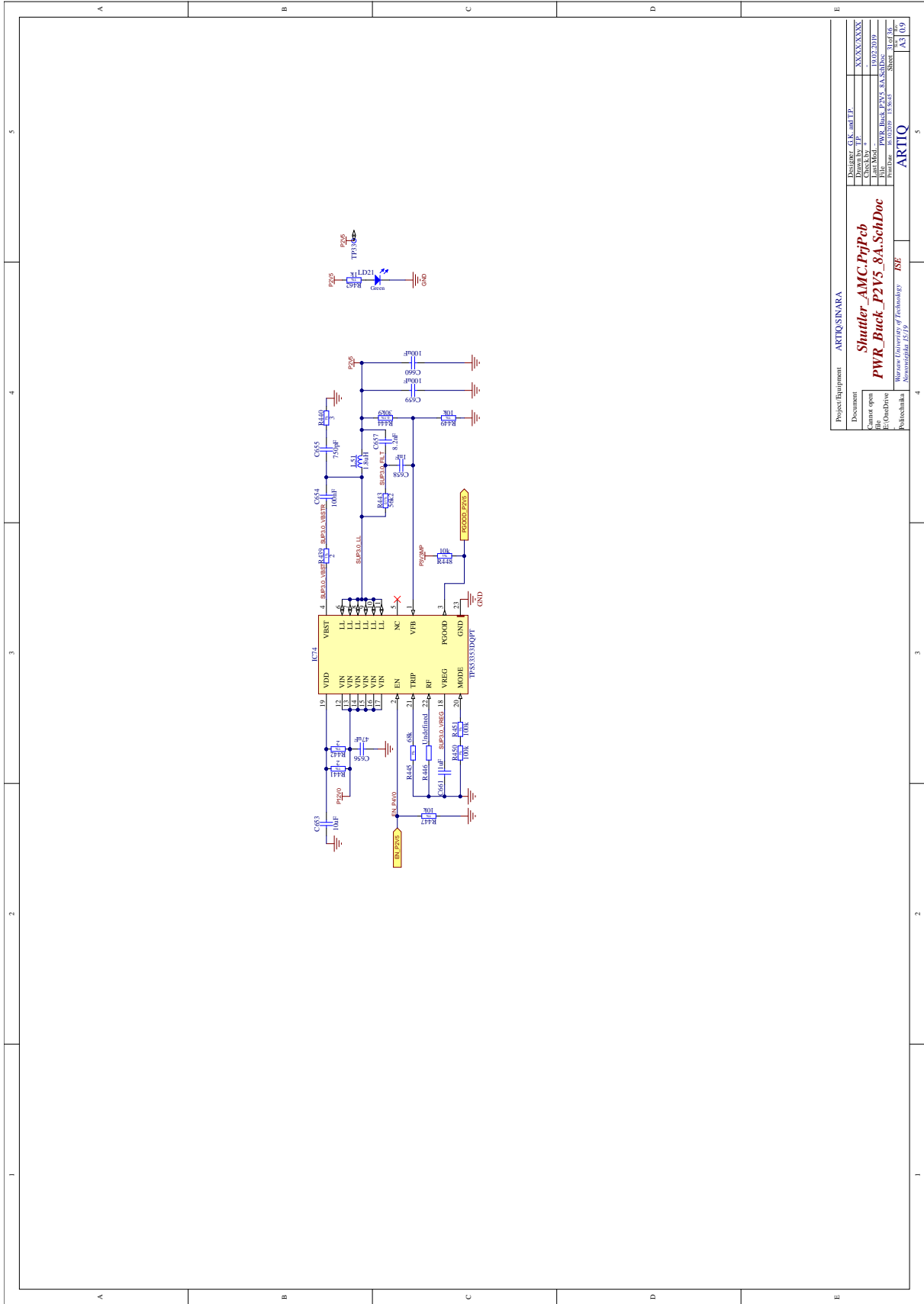
Shuttler AMC ProjPcb
 USB_SERIAL_QUAD.SchDoc
 Warsaw University of Technology
 Institute of Electronics and Information Technology
 Nowotki 15/19



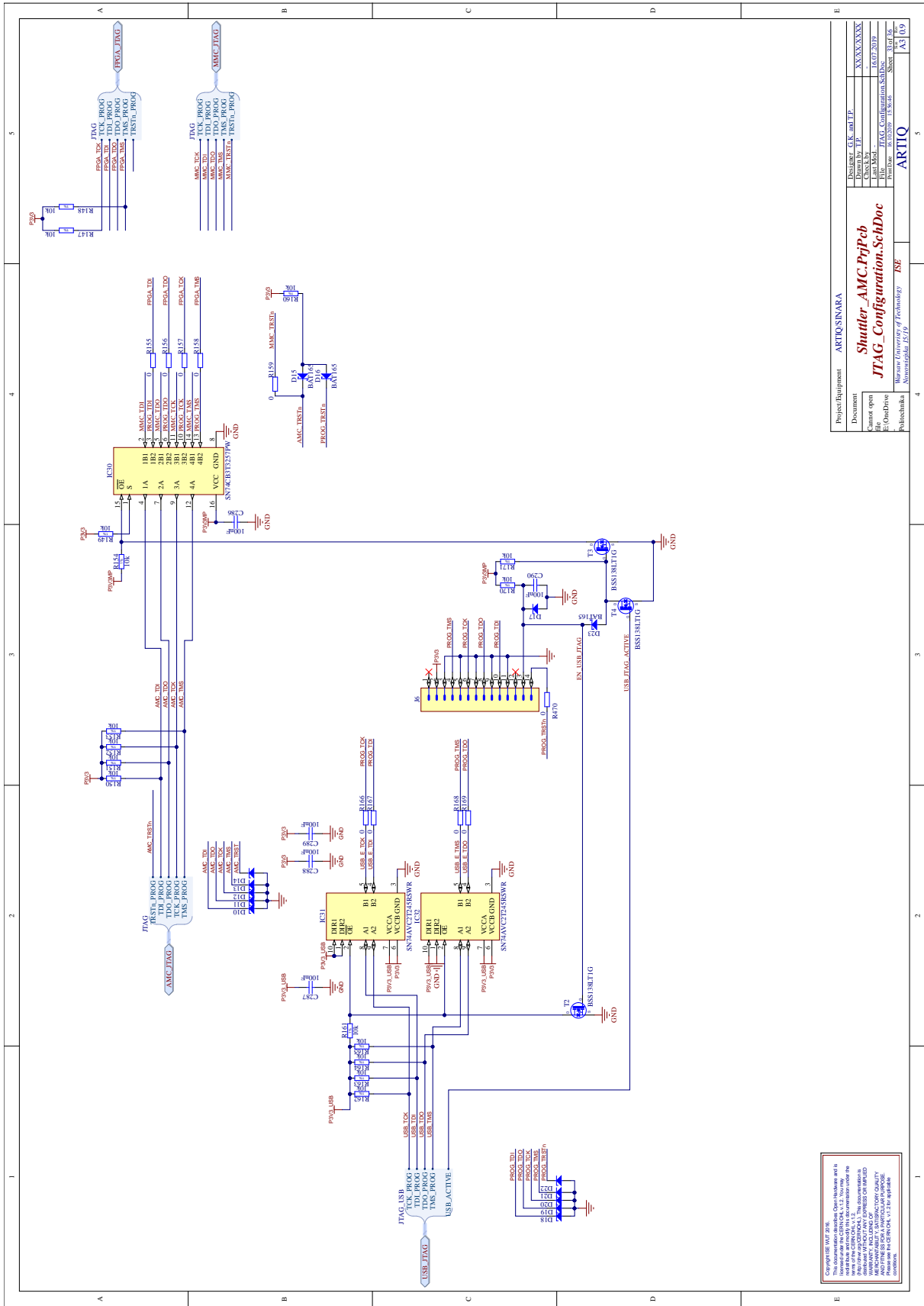
Project Equipment - ARTIQSINARA	
Document	Designer: G.K. and LP
Cannot open	Drawn by: LP
E:\QuadDrive	File: PWR_P3V0_N3V0_SchDoc
Photocahala	Project: 13249
November 15, 2019	Sheet: 02 of 02
ARTIQ	



Project/Equipment		ARTIQ/SINARA	
Document	Designer	G.K. and LP	XXXXXX
Cannot open	Drawing	LP	XXXXXX
E:\Coudrive	Cont Model		10/02/2019
Photochiba	File	PWR_Buck_AV_20A_SchDoc	Sheet 05
	Project	1.1 year	Sheet 05
	Photochiba	Marwan University of Technology	Sheet 05
		Amman 1979	Page 09



Project Equipment		ARTIQSINARA	
Document	Designer: G.K. and LP	Sheet	XXXX/XXXX
Cannot open	Drawn by: LP	File	14/07/2019
E:\C:\QadDrive	Cam Mod -	Printed	1/1
E:\C:\QadDrive	PWR_Buck_P2V5_8A.SchDoc	Sheet	1/1
Photochika	Version: 1.0.0.0	Sheet	1/1
Photocchika	Moscow University of Technology	Sheet	1/1
Novosibirsk, 1979	ISE	ARTIQ	14/07/2019



Project Equipment	ARTIQ SoC
Document	XXXXXXXXXX
Drawn by	XXXXXXXXXX
Checked by	XXXXXXXXXX
File	JTAG_Configuration_SchDoc
Printed	6/07/2019
Printed by	XXXXXXXXXX
Sheet	122
Total	129

Project Equipment: ARTIQ SoC

Document: XXXXXXXXXXXX

Drawn by: XXXXXXXXXXXX

Checked by: XXXXXXXXXXXX

File: JTAG_Configuration_SchDoc

Printed: 6/07/2019

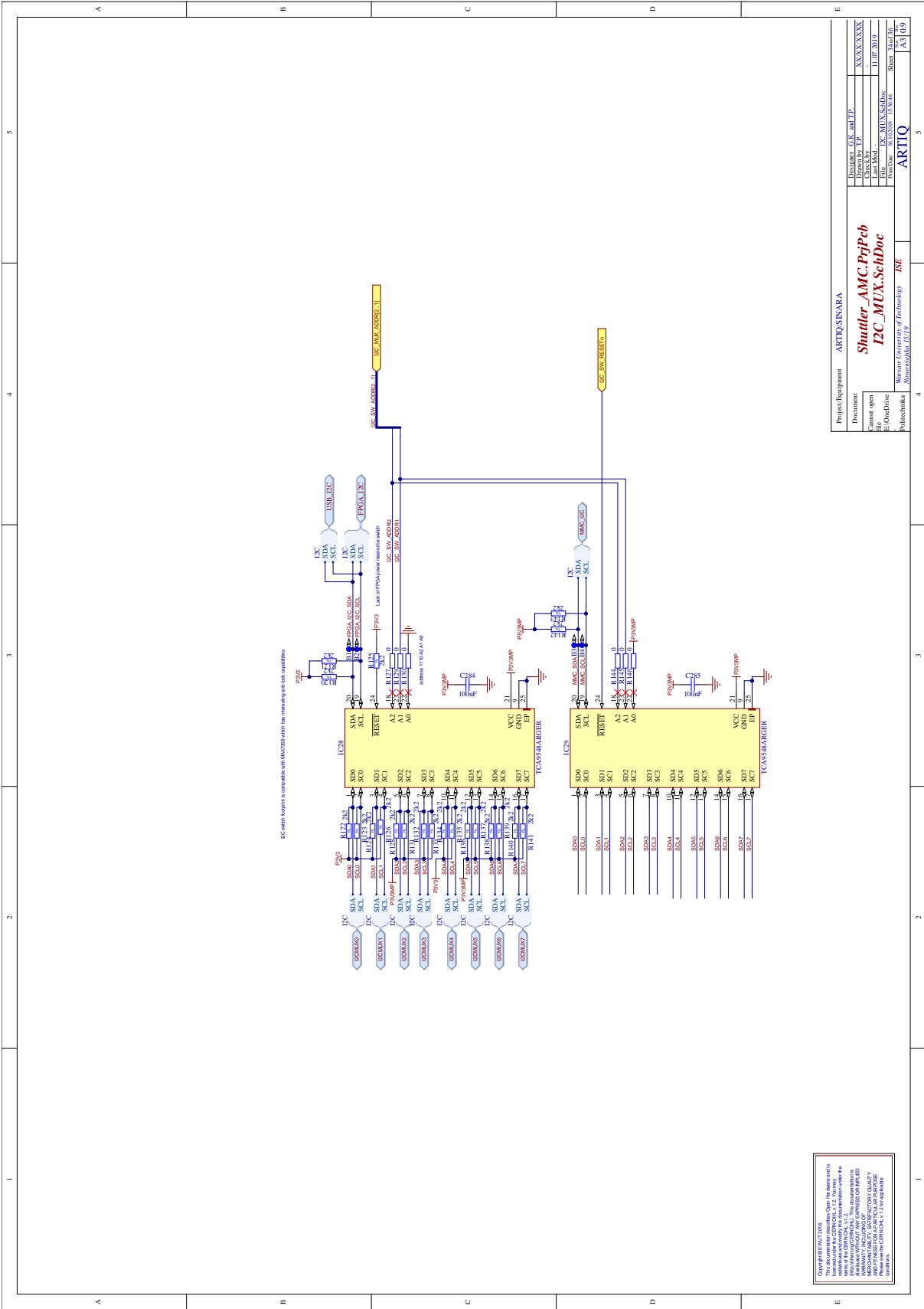
Printed by: XXXXXXXXXXXX

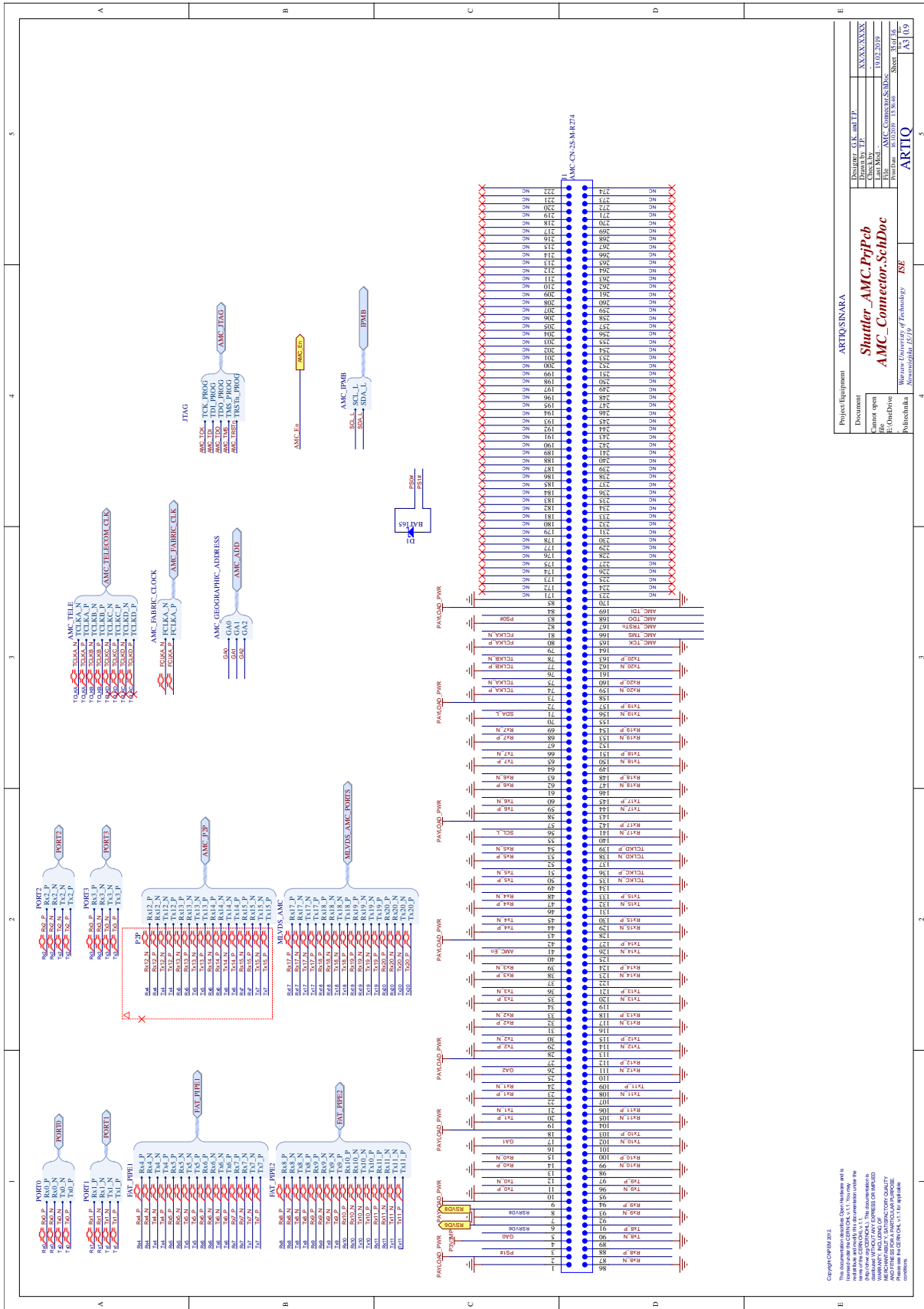
Sheet: 122

Total: 129

Shutter AMC PripCh
JTAG Configuration SchDoc
 Marwan University of Technology
 Al-Qadisiyah
 November 15, 2019

Copyright © 2019
 This document describes Open Hardware and is
 not a trade secret. It is intended for use under the
 terms of the Creative Commons Attribution-NonCommercial-ShareAlike license.
 All rights reserved. This document is licensed under the
 Creative Commons Attribution-NonCommercial-ShareAlike license.
 WWW.OPENHW.COM
 WWW.CC-BY-NC-SA.COM
 WWW.CC-BY-NC-SA.COM
 WWW.CC-BY-NC-SA.COM





B. Kody użyte w projekcie

B.1. Projekt FPGA

Listing 1: Kod głównego bloku (najwyższego w hierarchii).

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.std_logic_unsigned.all;
library UNISIM;
use UNISIM.VCOMPONENTS.ALL;

entity top is
  port (
    db_p : inout STD_LOGIC_VECTOR( 14 downto 0);
    db_n : inout STD_LOGIC_VECTOR( 14 downto 0);
    data_p : in STD_LOGIC;
    data_n : out STD_LOGIC;
    dckip : out STD_LOGIC;
    dckin : out STD_LOGIC;
    clk_50m_i : in STD_LOGIC
  );
end top;

architecture Behavioral of top is

  component design_1_wrapper is
    port (
      Q : out STD_LOGIC_VECTOR ( 15 downto 0 );
      clk_100_MHz : out STD_LOGIC;
      clk_50MHz : in STD_LOGIC
    );
  end component design_1_wrapper;

  component bufory is
    port (
      Bp : out STD_LOGIC_VECTOR( 14 downto 0);
      Bn : out STD_LOGIC_VECTOR( 14 downto 0);
      B : in STD_LOGIC_VECTOR( 15 downto 0)
    );
  end component bufory;

  signal data_clk : STD_LOGIC;
  signal licznik : STD_LOGIC_VECTOR ( 15 downto 0);

begin

  — dla pomiaru pierwszego B => licznik
  bufory_i: component bufory
    port map(
      Bp => db_p,
      Bn => db_n,
      B => "0000000000000000"
    );

  clock_wiz : component design_1_wrapper
    port map(
      clk_50MHz => clk_50m_i,
      data_clk=> data_clk,
      Q => licznik
    );

  B0 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => dckip,
```

```

    OB => dckin ,
    I => (data_clk)
);

data_n <= licznik(0); -- dla pomiaru pierwszego 'Z'

end Behavioral;

```

Listing 2: Kod buforów wyjść różnicowych

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.numeric_std.all;
library UNISIM;
use UNISIM.VCOMPONENTS.ALL;
entity bufory is
    port (
        Bp : out STD_LOGIC_VECTOR(14 downto 0);
        Bn : out STD_LOGIC_VECTOR(14 downto 0);
        B : in STD_LOGIC_VECTOR (15 downto 0)
    );
end bufory;

architecture STRUCTURE of bufory is

begin

B0 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(0) ,
    OB => Bn(0) ,
    I => (B(0))
);

B1 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(1) ,
    OB => Bn(1) ,
    I => (B(1))
);

B2 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(2) ,
    OB => Bn(2) ,
    I => (B(2))
);

B3 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(3) ,
    OB => Bn(3) ,
    I => (B(3))
);

B4 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(4) ,
    OB => Bn(4) ,
    I => (B(4))
);

B5 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(5) ,
    OB => Bn(5) ,
    I => (B(5))
);

B6 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(6) ,
    OB => Bn(6) ,

```

```

    I => (B(6))
);

B7 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(7),
    OB => Bn(7),
    I => (B(7))
);

B8 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(8),
    OB => Bn(8),
    I => (B(8))
);

B9 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(9),
    OB => Bn(9),
    I => (B(9))
);

B10 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(10),
    OB => Bn(10),
    I => (B(10))
);

B11 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(11),
    OB => Bn(11),
    I => (B(11))
);

B12 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(12),
    OB => Bn(12),
    I => (B(12))
);

B13 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(13),
    OB => Bn(13),
    I => ('Z')
);

B14 : OBUFDS generic map(IOSTANDARD => "LVDS") port map(
    O => Bp(14),
    OB => Bn(14),
    I => (B(13))
);

end STRUCTURE;

```

B.2. Kod mikrokontrolera do translacji UART - SPI

Listing 3: Mostek UART-SPI mikrokontrolera STM32

```

#include "stm32f10x.h"
#include "stdio.h"

void GPIO_Config(void);
void RCC_Config(void);
void NVIC_Config(void);
void USART_Config(void);
void SPI_Config(void);

```



```

void uart_send_char(char c);
void uart_send_string(const char* s);
uint16_t transfer_16b_SPI1_Slave(uint16_t outByte);
uint8_t get_char(char c);
char hex_to_int(uint8_t liczba);
void send_hex(uint8_t liczba);

int main(void)
{
    volatile unsigned long int i;
    unsigned int stanDiod, stanGPIOB;
    uint16_t spi_buff = 0x0000;
    uint16_t recv_buff = 0x0000;
    uint8_t uart_buff = 0x00;
    int counter = 0;

    RCC_Config();
    GPIO_Config();
    NVIC_Config();
    SPI_Config();
    USART_Config();
    GPIO_SetBits(GPIOA, GPIO_Pin_4);

    while (1) {
        if (USART_GetFlagStatus(USART1, USART_FLAG_RXNE) != RESET) {
            stanDiod=USART_ReceiveData(USART1);           // (1) First sign
            uart_send_char(stanDiod);                     // (2) Echo
            stanDiod = get_char(stanDiod);                 // (3) Convert to hex value
            if (stanDiod <=9){                            // (4) First char has to be number
                spi_buff = stanDiod << 12;                // (4.1) 4 hex numbers in buff -
                first one
                while(USART_GetFlagStatus(USART1, USART_FLAG_RXNE)==RESET);
                stanDiod=USART_ReceiveData(USART1);       // (1) Second sign
                uart_send_char(stanDiod);                 // (2) echo
                stanDiod = get_char(stanDiod);            // (3) Convert to hex value
                spi_buff = spi_buff | (stanDiod <<8);     // (4) 4 hex numbers in buff - second
                one
                while(USART_GetFlagStatus(USART1, USART_FLAG_RXNE)==RESET);
                stanDiod=USART_ReceiveData(USART1);
                uart_send_char(stanDiod);
                stanDiod = get_char(stanDiod);
                spi_buff = spi_buff | (stanDiod << 4);
                while(USART_GetFlagStatus(USART1, USART_FLAG_RXNE)==RESET);
                stanDiod=USART_ReceiveData(USART1);
                uart_send_char(stanDiod);
                stanDiod = get_char(stanDiod);
                spi_buff = spi_buff | stanDiod;
                uart_send_string("\r\nOdpowiedz_SPI:\r\n");
                recv_buff = transfer_16b_SPI1_Slave(spi_buff); // SPI transfer
                send_hex((recv_buff >> 8) & 0xFF);           // send back response
                send_hex(recv_buff & 0xFF);
                uart_send_string("\r\n");
            }
        }
    };
    return 0;
}

char hex_to_int(uint8_t i){
    if (i <=9)
        return i + 48;
    else{
        switch(i){
            case 0x0A:
                return 'A';
            case 0x0B:

```

```

        return 'B';
    case 0x0C:
        return 'C';
    case 0x0D:
        return 'D';
    case 0x0E:
        return 'E';
    case 0x0F:
        return 'F';
    default:
        return 'z';
    }
}
}

void send_hex(uint8_t liczba){
    uint8_t first = (liczba >> 4) & 0x0F;
    uint8_t second = liczba & 0x0F;
    uart_send_char(hex_to_int(first));
    uart_send_char(hex_to_int(second));
}

uint8_t get_char(char c){
    if ((c-48) <= 9){
        return c-48;
    }
    else{
        switch(c){
            case 0x41:
                return 0x0A;
            case 0x42:
                return 0x0B;
            case 0x43:
                return 0x0C;
            case 0x44:
                return 0x0D;
            case 0x45:
                return 0x0E;
            case 0x46:
                return 0x0F;
            default:
                return 0;
        }
    }
}

void uart_send_char(char c){
    while (USART_GetFlagStatus(USART1, USART_FLAG_TXE) == RESET);
    USART_SendData(USART1, c);
}

void uart_send_string(const char* s){
    while(*s)
        uart_send_char(*s++);
}

uint16_t transfer_16b_SPI1_Slave(uint16_t outByte){
    while (!SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_TXE));
    GPIO_ResetBits(GPIOA, GPIO_Pin_4);
    SPI_I2S_SendData(SPI1, outByte);
    while (!SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_RXNE));
    GPIO_SetBits(GPIOA, GPIO_Pin_4);
    // send
}

```

```

    while (!SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_RXNE));
    return SPI_I2S_ReceiveData(SPI1);           // read received
}

void RCC_Config(void)
{
    ErrorStatus HSEStartUpStatus;

    RCC_DeInit();                             //Reset RCC
    RCC_HSEConfig(RCC_HSE_ON);
    HSEStartUpStatus = RCC_WaitForHSEStartUp();
    if (HSEStartUpStatus == SUCCESS)
    {
        FLASH_PrefetchBufferCmd(FLASH_PrefetchBuffer_Enable); //
        FLASH_SetLatency(FLASH_Latency_2); //
                                                // 0:<24MHz; 1:24~48MHz; 2:>48
                                                //      MHz
        RCC_HCLKConfig(RCC_SYSCLK_Div1);       // HCLK=SYSCLK
        RCC_PCLK2Config(RCC_HCLK_Div1);       // PCLK2=HCLK
        RCC_PCLK1Config(RCC_HCLK_Div2);       // PCLK1=HCLK/2
        RCC_PLLConfig(RCC_PLLSource_HSE_Div1, RCC_PLLMul_9); // 8MHz * 9 = 72 MHz
        RCC_PLLCmd(ENABLE);
        while (RCC_GetFlagStatus(RCC_FLAG_PLLRDY) == RESET);
        RCC_SYSCLKConfig(RCC_SYSCLKSource_PLLCLK);
        while (RCC_GetSYSCLKSource() != 0x08);

        RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE);
        RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1, ENABLE);
        RCC_APB2PeriphClockCmd(RCC_APB2Periph_SPI1, ENABLE);
        RCC_APB2PeriphClockCmd(RCC_APB2Periph_AFIO, ENABLE);

    } else {
    }
}

void NVIC_Config(void)
{
#ifdef VECT_TAB_RAM
    NVIC_SetVectorTable(NVIC_VectTab_RAM, 0x0);
#else // VECT_TAB_FLASH
    NVIC_SetVectorTable(NVIC_VectTab_FLASH, 0x0);
#endif
}

void GPIO_Config(void)
{
    GPIO_InitTypeDef GPIO_InitStructure;

    //PA9 – Tx
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    //PA10 – Rx
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    // SPI1 : SCK | MISO | MOSI
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5 | GPIO_Pin_6 | GPIO_Pin_7;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
}

```

```

GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
GPIO_Init(GPIOA, &GPIO_InitStructure);

//SPI1 : SS
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
GPIO_Init(GPIOA, &GPIO_InitStructure);
}

void USART_Config(void)
{
    USART_InitTypeDef  USART_InitStructure;

    USART_InitStructure.USART_BaudRate = 115200;
    USART_InitStructure.USART_WordLength = USART_WordLength_8b;
    USART_InitStructure.USART_StopBits = USART_StopBits_1;
    USART_InitStructure.USART_Parity = USART_Parity_No;
    USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
    USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
    USART_Init(USART1, &USART_InitStructure);

    USART_Cmd(USART1, ENABLE);
}

void SPI_Config(void)
{
    SPI_InitTypeDef  SPI_InitStructure;

    SPI_InitStructure.SPI_Direction = SPI_Direction_2Lines_FullDuplex;
    SPI_InitStructure.SPI_Mode = SPI_Mode_Master;
    SPI_InitStructure.SPI_DataSize = SPI_DataSize_16b;
    SPI_InitStructure.SPI_CPOL = SPI_CPOL_Low;
    SPI_InitStructure.SPI_CPHA = SPI_CPHA_1Edge;
    SPI_InitStructure.SPI_NSS = SPI_NSS_Soft;
    SPI_InitStructure.SPI_BaudRatePrescaler = SPI_BaudRatePrescaler_256; // 36MHz
    //256=140.625kHz
    SPI_InitStructure.SPI_FirstBit = SPI_FirstBit_MSB;
    SPI_InitStructure.SPI_CRCPolynomial = 7;
    SPI_Init(SPI1, &SPI_InitStructure);
    SPI_SSOutputCmd(SPI1, ENABLE);

    SPI_Cmd(SPI1, ENABLE);
}

```