



Wydział Fizyki i Informatyki Stosowanej

Praca magisterska

Szymon Kulis

kierunek studiów: informatyka stosowana

specjalność: informatyka w nauce i technice

Projekt elektroniki front-end dla kalorymetru LumiCal dla ILC

Opiekun: dr hab. inż. Marek Idzik

Kraków, czerwiec 2008

Oświadczam, świadomy odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem osobiście i samodzielnie i nie korzystałem ze źródeł innych niż wymienione w pracy.

Kraków, 26 czerwca 2008

**Tematyka pracy magisterskiej i praktyki dyplomowej Szymona Kulisa,
studenta V roku studiów kierunku informatyka stosowana, specjalności
informatyka w nauce i technice**

Temat pracy magisterskiej: **Projekt elektroniki front-end dla kalorymetru
LumiCal dla ILC**

Opiekun pracy: dr hab. inż. Marek Idzik

Recenzenci pracy: dr inż. Tomasz Fiutowski

Miejsce praktyki dyplomowej: DESY, Zeuthen

Program pracy magisterskiej i praktyki dyplomowej

1. Omówienie realizacji pracy magisterskiej z opiekunem.
2. Zebranie i opracowanie literatury dotyczącej tematu pracy.
3. Praktyka dyplomowa:
 - projekt układu elektronicznego,
 - wykonanie oraz testy układu elektronicznego własnego projektu,
 - dyskusja i analiza wyników pomiarów,
 - sporządzenie sprawozdania z praktyki.
4. Kontynuacja prac związanych z tematem pracy magisterskiej.
5. Zebranie i opracowanie wyników pomiarów.
6. Analiza wyników pomiarów, ich omówienie i zatwierdzenie przez opiekuna.
7. Przygotowanie dokumentacji technicznej.
8. Opracowanie redakcyjne pracy.

Termin oddania w dziekanacie: 27 czerwca 2008

.....
(podpis kierownika katedry)

.....
(podpis opiekuna)

Merytoryczna ocena pracy przez opiekuna:

Końcowa ocena pracy przez opiekuna:

Data:

Podpis:

Merytoryczna ocena pracy przez recenzenta:

Końcowa ocena pracy przez recenzenta:

Data:

Podpis:

Skala ocen: (6.0 – celująca), 5.0 – bardzo dobra, 4.5 – plus dobra, 4.0 – dobra, 3.5 – plus dostateczna, 3.0 – dostateczna, 2.0 – niedostateczna

Spis treści

Wstęp	9
Rozdział 1. Międzynarodowy Liniowy Zderzacz ILC	11
1.1. Fizyka w ILC	11
1.2. Akcelerator	12
1.3. Detektory	13
1.3.1. Kalorymetr przedni i detektor świetlności	14
Rozdział 2. Spektrometryczny tor pomiarowy detektora LumiCal	17
2.1. Sensory promieniowania	18
2.2. Wzmacniacz ładunkoczuły	21
2.2.1. Parametry małosygnalowe	25
2.2.2. Analiza szumowa	27
2.3. Kształtowanie impulsów i filtracja szumów	29
2.3.1. Filtracja (CR)-(RC)	30
2.3.2. Filtracja z PZC	32
2.3.3. Filtracja nieciągła w czasie	33
2.4. Przetwornik ADC	35
2.4.1. Funkcja przenoszenia przetwornika	35
2.4.2. Szum kwantyzacji i S/N idealnego ADC	35
2.4.3. ADC o architekturze potokowej	37
2.4.4. Korekcja cyfrowa	38
Rozdział 3. Projekt i pomiary przedwzmacniacza	43
3.1. Architektura	43
3.2. Model małosygnalowy	44
3.3. Realizacja	45
3.4. Wyniki symulacji	46
3.4.1. Charakterystyki częstotliwościowe	46

3.4.2.	Odpowiedzi czasowe	47
3.4.3.	Wzmocnienie ładunkowe	48
3.4.4.	Stabilność	49
3.4.5.	Szumy	49
3.5.	Plan masek	52
3.6.	Stanowisko pomiarowe	53
3.7.	Wyniki pomiarów	57
Rozdział 4.	Pomiary przetwornika analogowo-cyfrowego	63
4.1.	Metodologia pomiarów przetworników analogowo-cyfrowych	63
4.1.1.	Metryki statyczne	63
4.1.2.	Metryki dynamiczne	65
4.2.	Implementacja potokowego przetwornika ADC	70
4.3.	Pomiary parametrów ADC	72
4.3.1.	Stanowisko pomiarowe	72
4.3.2.	Pomiary statyczne	76
4.3.3.	Pomiary dynamiczne	76
Rozdział 5.	Stanowisko testowe i system akwizycji danych	83
5.1.	Płytki testowa	84
5.2.	Szybki i wysoko-rozdzielczy system próbkowania sygnałów analogowych	89
5.3.	Szybki system akwizycji danych cyfrowych	90
5.3.1.	Szybka logika sterująca	92
5.3.2.	Mikrokontroler	93
5.3.3.	Komunikacja	96
5.4.	Oprogramowanie	97
Rozdział 6.	Podsumowanie	101
Dodatek A.	Dokumentacja płytki drukowanej	105
Dodatek B.	Model bloku szybkiej logiki	109
Dodatek C.	Oprogramowanie mikrokontrolera	111
C.1.	main.c	111
C.2.	daq.h	112
C.3.	daq.c	112
C.4.	uart.h	113
C.5.	uart.c	113
C.6.	utils.h	114

C.7. utils.c	114
Dodatek D. Biblioteki obsługi sprzętu	115
D.1. gpibdev.h	115
D.2. gpibdev.cpp	118
Bibliografia	125
Spis rysunków	129
Spis tablic	133

Wstęp

Najbliższa przyszłość może się okazać jednym z najbardziej owocnych okresów Fizyki dla zrozumienia podstawowych składników i praw natury. Być może wraz z uruchomieniem Wielkiego Zderzacza Hadronów LHC (ang. Large Hadron Collider) poznamy eksperymentalne przesłanki, które pomogą wytłumaczyć mechanizm uzyskiwania przez cząstki masy, a może nawet dowiemy się czegoś na temat pochodzenia ciemnej materii czy ciemnej energii. Prawie na pewno zostanie zmodyfikowany, rozszerzony czy zastąpiony bardziej ogólną teorią, tak dobrze dotychczas pracujący, model standardowy. Dla precyzyjnego zbadania nowych zjawisk niezbędny stanie się wtedy dużo prostszy niż proton-proton i pozwalający na precyzyjniejszą analizę system zderzeń. Będą nim z bardzo dużym prawdopodobieństwem kolizje elektron-pozyton. Dla dalszego rozwoju Fizyki Cząstek Elementarnych najistotniejsze stanie się wtedy pytanie o obszar energii, w którym LHC odkryje nowe zjawiska. Jeśli stanie się to w obszarze poniżej 1 TeV, to z graniczącym z pewnością przekonaniem podjęta zostanie decyzja o jak najszybszej budowie liniowego zderzacza elektronów i pozytonów ILC (ang. International Linear Collider). Jeśli nowe zjawiska odkryte zostaną w obszarze o energii kilku TeV, to bardziej prawdopodobna stanie się budowa liniowego akceleratora CLIC (ang. Compact Linear Collider). W tym jednak przypadku horyzont czasowy może się okazać zdecydowanie dłuższy, gdyż technologia akceleratora CLIC jest ciągle w bardzo wstępnym stadium rozwoju. Jednak nawet gdyby ta druga opcja się sprawdziła to budowa samego systemu detekcji produktów zderzeń w CLIC-u może się okazać bardzo podobna do tej z ILC. Dzieje się tak dlatego, gdyż w przeciwieństwie do całkowicie odmiennego sposobu akceleracji cząstek, koncepcja rejestracji produktów zderzeń jest w obu przypadkach bardzo podobna.

Od wielu lat trwają już badania i budowane są wstępne prototypy urządzeń, tak dla przyszłego akceleratora, jak i dla detektora ILC. W chwili obecnej liczba fizyków zaangażowanych w te badania wyraża się już w tysiącach osób, pracujących przy setkach uczelni i instytucji naukowych w różnych krajach, na (prawie) wszystkich kontynentach. Katedra Oddziaływań i Detekcji Cząstek AGH od lat jest zaangażowana w budowę jednego z detektorów przy ILC, a mianowicie detektora do pomiaru świetlności, czyli detektora LumiCal (ang. Luminosity Calorimeter). Jest to chyba jedyny detektor dla ILC, który prawie w całości ma być zbudowany przez

polskie grupy naukowców. Jednym z podstawowych zadań Katedry Oddziaływań i Detekcji Cząstek jest zaprojektowanie i zbudowanie systemu elektroniki odczytu detektora LumiCal.

Niniejsze opracowanie jest pierwszą pracą magisterską całkowicie poświęconą zagadnieniu odczytu sygnałów z detektora LumiCal. Ponieważ prace koncepcyjne i projektowe nad tym zagadnieniem trwają zaledwie dwa lata, znalazło to pełne odzwierciedlenie w zawartości pracy, opisującej projekty prototypowych układów elektroniki odczytu oraz pomiary pierwszych otrzymanych struktur testowych. Organizacyjnie praca została podzielona na pięć rozdziałów.

W pierwszym rozdziale poglądowo omówiony został eksperyment ILC. W omówieniu tym podano motywację fizyczną oraz przedstawiono najważniejsze dane dotyczące akceleratora i architektury systemu detekcji.

Drugi rozdział omawia spektrometryczny tor pomiarowy detektora LumiCal. W jego kolejnych podrozdziałach opisane zostały odpowiednio: sensory promieniowania jonizującego, wzmacniacze ładunkoczułe odczytujące sygnały z sensorów, układy kształtowania i filtracji szumów oraz przetworniki analogowo-cyfrowe zmieniające wyjściowy sygnał analogowy w postać cyfrową.

W trzecim rozdziale zaprezentowano prototypowy układ elektroniki front-end zaprojektowany przez autora pracy. W pierwszej kolejności opisany został projekt proponowanego przedwzmacniacza. Następnie pokazano wyniki przeprowadzonych symulacji elektrycznych oraz omówiono projekt układu masek technologicznych. W drugiej części tego rozdziału przedyskutowano wyniki pomiarów pierwszych prototypów zaprojektowanego przez autora wzmacniacza ładunkoczułego, wykonanego w submikronowej technologii AMS 0,35 μm (ang. Austrian Micro Systems).

Rozdział czwarty poświęcony jest testom prototypowych układów przetwornika analogowo-cyfrowego, wykonanym przez autora pracy. Na początku opisana została metodologia testowania przetworników ADC, a w szczególności opisano podstawowe sposoby pomiarów własności statycznych i dynamicznych przetworników. W drugiej części rozdziału przedstawiono wyniki przeprowadzonych pomiarów wraz z wpływającymi z nich wnioskami.

W rozdziale piątym opisano stanowisko testowe oraz układ akwizycji danych zaprojektowany i wykonany przez autora. Bez tego stanowiska, czyli odpowiedniej konfiguracji instrumentów pomiarowych, dedykowanych płytek testowych, sterującego pomiarami oprogramowania i systemu akwizycji danych, nie byłoby możliwe wykonanie pomiarów opisanych w rozdziale trzecim i czwartym. Integralną część rozdziału piątego stanowi opis bardzo szybkiego, dedykowanego systemu akwizycji danych, opartego na technologii programowalnych układów logicznych FPGA (ang. Field-Programmable Gate Array).

Pracę kończy podsumowanie, w którym uwypuklono najważniejsze osiągnięcia pracy oraz wyszczególniono pełny wkład autora.

Rozdział 1

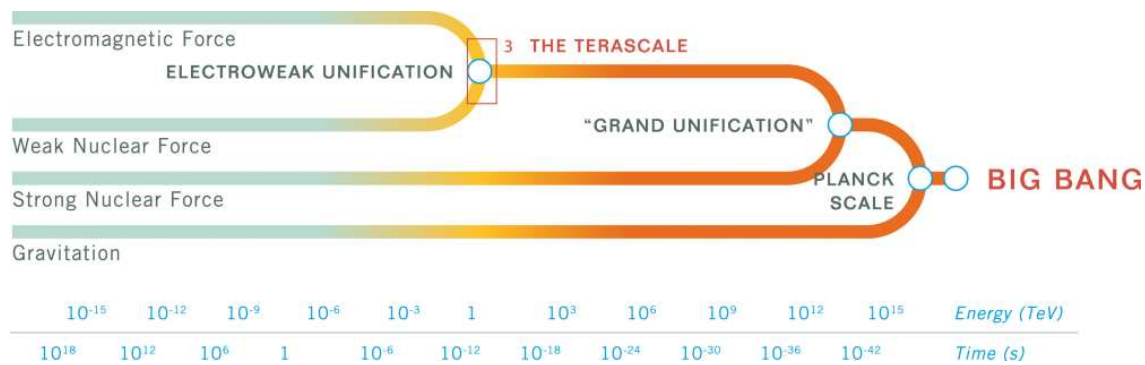
Międzynarodowy Liniowy Zderzacz ILC

1.1. Fizyka w ILC

Od zarania dziejów ludzie zadają pytania, tworzą teorię oraz przeprowadzają eksperymenty aby zrozumieć naturę otaczającego ich świata. Odkrycia te zmieniają nasze spojrzenie na świat, zmieniają nasze społeczeństwo oraz wpływają stymulująco na rozwój cywilizacji. Fizyka cząstek elementarnych jest dziedziną nauki pytającą o strukturę materii oraz oddziaływań pomiędzy jej składnikami. Podstawowym narzędziem badawczym są ogromne akceleratory cząstek, które zamieniają masę w energię i z powrotem energię w masę zgodnie z prawem $E = mc^2$. Wielkim triumfem fizyki dwudziestego wieku było odkrycie Modelu Standardowego. Różne eksperymenty określały składniki zwykłej materii, identyfikowały podstawowe cztery siły scalające materię. Szereg sukcesów prowadzi fizyków cząstek elementarnych do zadawania jeszcze dogłębszych pytań, takich jak: o istnienie jeszcze nie odkrytych praw natury, możliwość unifikacji wszystkich znanych sił, naturę ciemnej materii oraz ciemnej energii. Międzynarodowy Liniowy Zderzacz ILC ma za zadanie odegrać znaczącą rolę w tym procesie [ILC08].

Model Standardowy zawiera jeszcze jeden komponent, poza cząstkami i siłami, który nie został jeszcze zweryfikowany doświadczalnie - mechanizm Higgs'a tłumaczący masy cząstek. Teoria ta mówi, że pole Higgs'a penetruje wszechświat, nadając masę cząstkom elementarnym oraz rozbijając oddziaływania elektrosłabe na dwa: oddziaływania elektromagnetyczne oraz oddziaływania słabe (rys. 1.1). Cząstka Higgs'a nie może być stabilna w rejonie teraelektronowoltowych (ang. Tera-scale) energii, przez co koniecznym staje się wprowadzenie nowych teorii mogących opisać obserwowane zjawiska, takich jak teorie supersymetryczne lub inne teorie mówiące o dodatkowych wymiarach. Jeśli cząstka Higgs'a istnieje, konieczne będzie jej dokładne zbadanie, w przeciwnym wypadku konieczny będzie nowy mechanizm tłumaczący łamanie oddziaływania elektrosłabego oraz pochodzenie masy cząstek.

W ciągu najbliższych lat kilka eksperymentów w Europejskim Ośrodku Badań Jądrowych (fr. Conseil Européen pour la Recherche Nucléaire – CERN) pracujących na akceleratorze zwanym Wielkim Zderzaczem Hadronów LHC, zderzającym protony z antyprotonami, spojrzysz na region teraelektronowoltowych energii. Jeśli cząstka Higgs'a istnieje, prawie na pewno zostanie ona odnaleziona w LHC, a jej



Rysunek 1.1. Skala energetyczna pokazująca unifikację oddziaływań elektromagnetycznych i oddziaływań słabych w zakresie energii teraelektronowoltowych [ILC08]

masa zostanie zmierzona w eksperymentach ATLAS lub CMS [ATL00, CMS05]. Jeśli istnieje więcej cząstek Higgs'a eksperymenty na LHC mają bardzo dużą szansę ich zaobserwowania. Będzie jednak bardzo trudno dokonać precyzyjnych pomiarów spinu i parzystości, a co za tym idzie wyjaśnić prawa natury. Źródłem tych ograniczeń jest bardzo bogata struktura cząstek (protonów) zderzanych w tych eksperymentach, utrudniająca interpretację obserwowanych przypadków. Aby sprostać temu zadaniu zaproponowano budowę liniowego akceleratora ILC będącego zderzaczem elektronów i pozytonów. Ponieważ cząstki te nie mają struktury wewnętrznej (wedle obecnie posiadanych wyników eksperymentalnych i modelu standardowego), ich wzajemne zderzenia powinny być dużo bardziej przejrzyste, co pozwoli na bardzo precyzyjne pomiary interesujących wielkości. Precyzja ta niestety okupiona jest zakresem energii możliwych do zbadania. Dzięki tym pomiarom możliwa będzie weryfikacja czy cząstka Higgs'a jest zwykłym składnikiem Modelu Standardowego czy czymś bardziej skomplikowanym. W zderzacz ILC możliwe będzie również badanie wielu innych zjawisk. Szczegółowy opis wraz z obszerną listą referencji zamieszczono w pracy [ES07c].

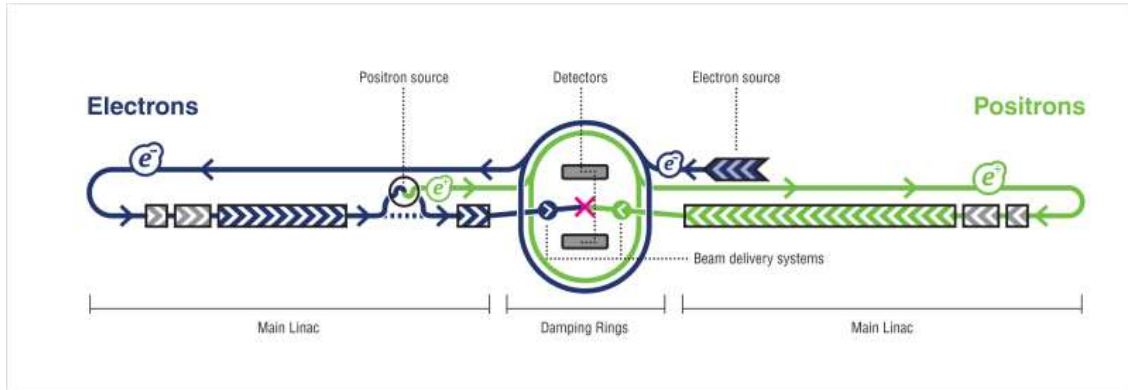
1.2. Akcelerator

Podstawowymi wymogami stawianymi akceleratorowi ILC są:

- energia zderzeń dochodząca do 500 GeV, z możliwością rozszerzenia do 1 TeV,
- świetlność na poziomie $2 \cdot 10^{34} \text{cm}^{-2} \text{s}^{-1}$, co daje scałkowaną świetlność w okresie pierwszych czterech lat operacji wynoszącą 500fb^{-1} ,
- możliwość zmiany energii (wykonywanie skanów) w zakresie od 200 do 500 GeV,
- polaryzacja wiązki elektronów na poziomie conajmniej 80%,
- stabilność energii na poziomie lepszym niż 0,1%,
- możliwość wykonywania eksperymentów elektron-elektron oraz foton-foton.

W pracy „International Linear Collider. Reference Design Report. Accelerator” [ES07a] została opisana proponowana architektura akceleratora, mająca spełnić te

wymagania. Na rysunku 1.2 przedstawiono schematycznie proponowane rozwiązanie. W tej koncepcji akcelerator bazuje na 1,3 GHz nadprzewodzących wnękach rezonansowych, pracujących z gradientem pola elektrycznego wynoszącym $31,5 \text{ MV/m}$. Elektrony będą zorganizowane w paczki zderzające się ze sobą co około 330 ns.



Rysunek 1.2. Schemat projektu eksperymentu ILC [ILC08]

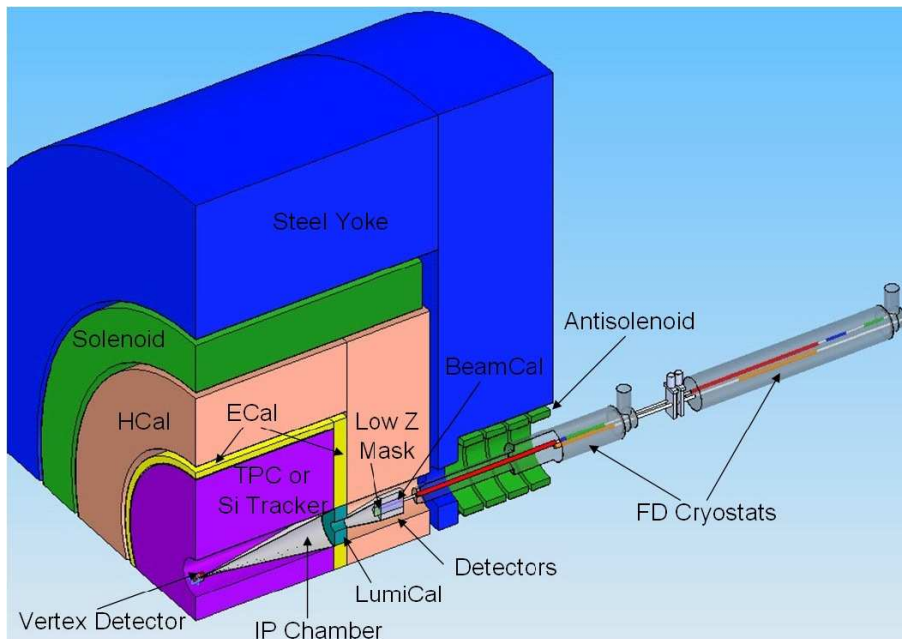
Zderzenia będą następowały przez około 1 ms (około 3000 zderzeń). Taka sekwencja będzie powtarzana 5 razy na sekundę. Długość akceleratora kształtować się będzie na poziomie 31 km (dla energii 500 GeV). Wiązka niskoenergetycznych elektronów przygotowywana będzie w akceleratorze kołowym o średnicy około 6,7 km. Tak przygotowana wiązka przyspieszana będzie w części liniowej akceleratora o długości 11 km z każdej strony.

1.3. Detektory

Kluczowym elementem każdego eksperymentu na akceleratorze są systemy detekcyjne będące w stanie obserwować punkt interakcji. Zadaniem detektorów jest precyzyjny pomiar trajektorii, masy oraz energii produktów powstałych w wyniku zderzenia cząstek pierwotnych. Na obecnym etapie rozwoju projektu prowadzone są prace nad kilkoma koncepcjami systemów detekcji, z których finalnie zostaną wybrane dwa i będą pracować w eksperymencie naprzemiennie (ang. pull-and-push scheme). Jedną z koncepcji detektora jest Koncepcja Wielkiego Detektora (ang. Large Detector Concept – LDC) nad którą pracują głównie grupy europejskich naukowców [LDC08]. Detektor ten składał się będzie, idąc od jego wnętrza, z:

- detektora wierzchołka (ang. micro-vertex detector),
- komory projekcji czasowej (ang. time projection chamber – TPC),
- wysoko-rozdzielczego zespołu kalorymetrów elektromagnetycznych i hadronowych (ang. ECal, HCal),
- komór mionowych.

Architektura taka ma zapewnić możliwość bardzo precyzyjnej rekonstrukcji torów i energii cząstek (ang. particle flow). Schematycznie rozmieszczenie poszcze-



Rysunek 1.3. Schemat ideowy detektora LDC [ILC08]

gólnych systemów detekcji zaprezentowano na rysunku 1.3. Ponieważ wspomniane wyżej główne systemy detekcji nie są przedmiotem niniejszej pracy, dla zainteresowanych, ich bardziej szczegółowy opis można znaleźć np. w pracy [ES07b].

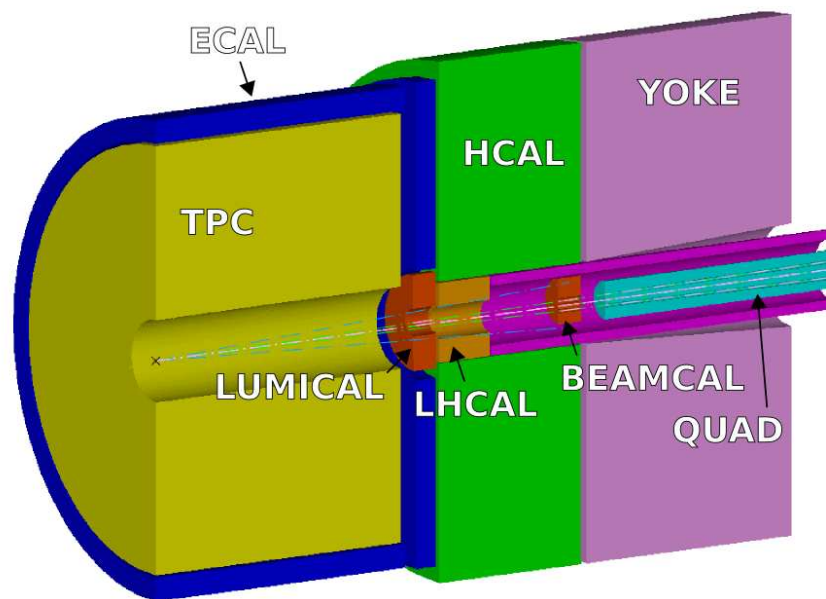
1.3.1. Kalorymetr przedni i detektor świetlności

Część systemu kalorymetrycznego stanowi tak zwana część przednia detektora (ang. forward region) obserwująca punkt interakcji pod bardzo małymi kątami [FCA08]. Zadaniem urządzeń pracujących w tym obszarze są:

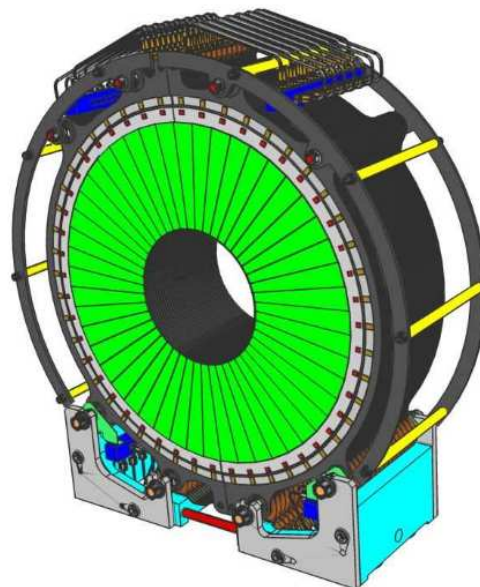
- pomiar świetlności (ang. luminosity),
- dostarczanie veta dwuelektronowego,
- dostarczanie informacji diagnostycznych na temat wiązki,
- zapewnianie hermetyczności całego systemu detekcyjnego.

W skład instrumentalizacji części przedniej wchodzi następujące detektory: detektor świetlności (ang. Luminosity Calorimeter – LumiCal), którego zadaniem jest precyzyjny pomiar częstości zdarzeń typu Bhabha, detektor wiązki (ang. Beam Calorimeter – BeamCal) oraz detektor fotonów (ang. Gamma Calorimeter – GamCal), których zadaniem jest dostarczanie informacji o chwilowej intensywności i aktualnej pozycji wiązki. Informacja ta wykorzystywana jest do korygowania nastawów systemu dostarczania wiązki, aby uzyskać możliwie dużą świetlność. Schematycznie rozmieszczenie detektorów w przedniej części zaprezentowano na rysunku 1.4.

Detektor LumiCal, będący głównym tematem tej pracy, jest kalorymetrem składającym się z aktywnych warstw krzemu oraz absorbcyjnych warstw wolframu. Detektor składa się z dwóch półbeczek, które zapięte będą na wiązce wychodzącej.



Rysunek 1.4. Schemat ideowy przedniej części detektora LDC (ang. forward region)
[FCA08]



Rysunek 1.5. Projekt mechaniczny detektora LumiCal [FCA08]

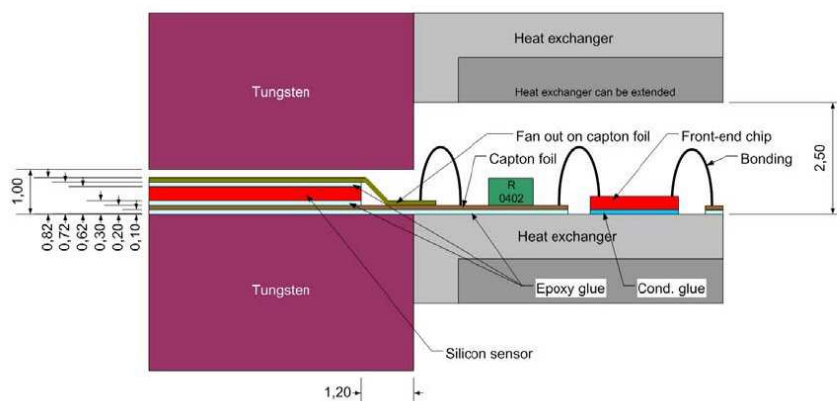
W obecnym układzie detektor LumiCal (rys. 1.5) umieszczony jest 2,27 m od punktu interakcji. Wewnętrzny promień detektora wynosi 80 mm, a zewnętrzny 350 mm. Wymiary takie określają kąty pod jakimi detektor spogląda na punkt

interakcji jako 35 do 153 mrad. Przekrój poprzeczny wzdłuż osi detektora LumiCal został zaprezentowany na rysunku 1.6. Wewnętrzna część każdej z warstw składa się z:

- 3,5 mm warstwy wolframu (co odpowiada jednej drodze radiacyjnej),
- 0,3 mm warstwy sensorów krzemowych,
- folii kaptonowej na której są ścieżki łączące pady sensora z elektroniką,
- kilku warstw kleju (łączącego poszczególne warstwy).

W kolejnych warstwach wolframu następują oddziaływania promieniowania elektromagnetycznego i leptonów, w wyniku których produkowane są kaskady niskoenergetycznych fotonów, elektronów i pozytonów (ang. electromagnetic shower), które są następnie absorbowane w warstwach sensorów krzemowych.

W zewnętrznej części detektora zarezerwowano miejsce na układy elektroniki front-end, złącza oraz system chłodzący. Jak można zauważyć przestrzeń dostępna



Rysunek 1.6. Przekrój poprzeczny wzdłuż osi detektora LumiCal [FCA08]

dla układów odczytowych jest bardzo mała (około 2,5 mm wysokości i kilkanaście milimetrów długości). Konsekwencje tak małej dostępnej przestrzeni omówiono w kolejnych rozdziałach. Na obecnym etapie prac, najbardziej prawdopodobną realizacją systemu chłodzącego jest system z chłodzeniem wodnym, z przepływem wymuszonym.

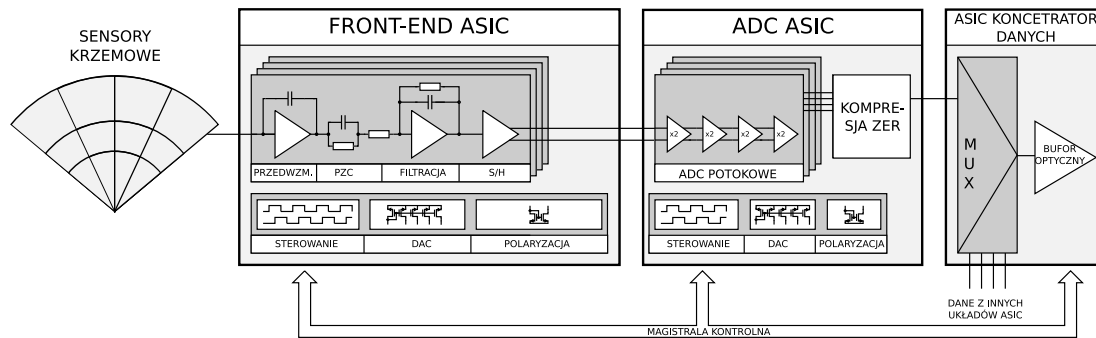
Rozdział 2

Spektrometryczny tor pomiarowy detektora LumiCal

Detektor LumiCal, omówiony poglądowo w poprzednim rozdziale, stanowi tylko jeden z systemów detekcji eksperymentu ILC. Aby cały eksperyment pracował efektywnie, każdy z jego systemów detekcji musi spełniać szereg wymagań nałożonych tak przez fizykę badanych procesów jak i całościową architekturę detektora ILC (uwzględniającą powiązania pomiędzy różnymi systemami detekcji). Z tego względu także detektor LumiCal, a w szczególności jego elektronika odczytu muszą spełniać szereg postawionych wymagań i dostosować się do wielorakich ograniczeń. Należy pamiętać, że odczyt krzemowych sensorów kalorymetru LumiCal wymaga specjalizowanych układów elektronicznych, które zdolne są mierzyć ładunek deponowany przez cząstki w pojedynczych segmentach, zwanych dalej padami sensora. Do najistotniejszych wymagań istotnych także z punktu widzenia elektroniki odczytu należą:

- ograniczona przestrzeń (patrz rys. 1.6). Ponieważ detektor będzie zawierał bardzo dużą liczbę kanałów (blisko 200 000), przestrzeń dostępna na odczyt pojedynczego kanału jest bardzo ograniczona. Układ elektroniczny wykonany z elementów dyskretnych nie zmieściłby się w wymaganej objętości detektora,
- wymagania szumowe. Ponieważ każda pojemność dołączona do wejścia przedwzmacniacza degraduje parametry pracy całego toru odczytowego (patrz rozdział 2.2.2) dlatego też połączenia sensor-przedwzmacniacz muszą być jak najkrótsze,
- minimalizacja wtórnych oddziaływań. Każdy dodany materiał jest źródłem wtórnych oddziaływań, co może zaburzać rekonstrukcję przypadków. W przypadku elektroniki odczytu takim problemem mogłoby być stosowanie układów z obudowami ceramicznymi elementów dyskretnych,
- minimalizacja masy detektora. Oszacowanie na chwilę obecną jest już bardzo duże (prawie 500 kg) [BDD⁺07],
- niezawodność pracy. W przypadku elektroniki wykonanie w postaci zintegrowanych układów scalonych jest dużo bardziej niezawodne niż skomplikowane systemy elementów dyskretnych. Dodatkową zaletą tworzenia układu scalonego jest możliwość bardzo dokładnego wymiarowania każdego z tranzystorów pod kątem uzyskania żądanych parametrów, podczas gdy w technologiach dyskretnych jesteśmy skazani na elementy poniekąd uniwersalne. Ponadto technologie scalone zapewniają wysoką jednorodność parametrów w obrębie jednego wafla krzemowego [HH05].

Po analizie powyższych wymagań stwierdzono [BDD⁺07], iż jedynym sposobem realizacji odczytu detektora LumiCal jest wykonanie elektroniki odczytu w postaci specjalizowanych układów scalonych ASIC (ang. Application Specific Integrated Circuit).



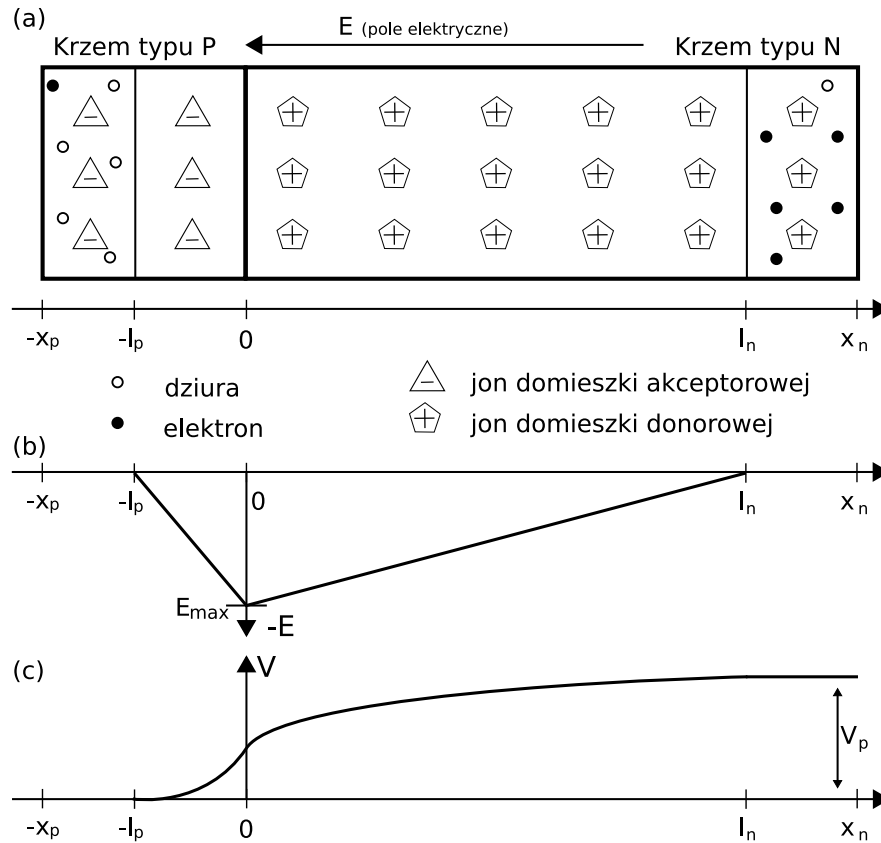
Rysunek 2.1. Schemat toru elektroniki odczytu

Zadaniem elektroniki front-end jest przetworzenie sygnału pochodzącego z sensora promieniowania, tj. wzmacnianie go oraz ukształtowanie w celu osiągnięcia założonego stosunku sygnału do szumu. Amplituda tak przetworzonego sygnału powinna zostać zapamiętana oraz wysłana do przetwornika analogowo-cyfrowego, skąd w postaci cyfrowej powinna trafić do systemu akwizycji danych. Architektura systemu realizującego takie zadanie została przedstawiona na rysunku 2.1. Na obecnym poziomie zaawansowania prac tylko elementy elektroniki front-end oraz ADC były przedmiotem badań. W dalszej części pracy autor skupi się na projekcie przedwzmacniacza, testach prototypowych układów scalonych oraz projekcie stanowiska pomiarowego wraz z systemem akwizycji danych, czym osobiście się zajmował.

2.1. Sensory promieniowania

Większość współcześnie stosowanych półprzewodnikowych sensorów promieniowania to zaporowo spolaryzowane złącza $p^+ - n$. Obszar aktywny sensora stanowi obszar ładunku przestrzennego, zwany również warstwą zubożoną (rys. 2.2). Zewnętrzne obwody polaryzujące generują odpowiednio silne pole elektryczne, które zapewnia zbieranie ładunku wyprodukowanego w objętości czynnej sensora, generując sygnał prądowy. Nośniki wygenerowane termicznie również dają kontrybucję do całkowitego prądu. Ten typ prądu, ciągłego w czasie, jest nazywany prądem upływu. Prąd ten w istotny sposób degraduje jakość pracy całego systemu detekcyjnego ze względu na zwiększenie szumów.

Tak ze względu na prądy upływu, jak również na doskonale opanowaną technologię produkcji, podstawowym półprzewodnikiem używanym do wytwarzania sensorów jest krzem.

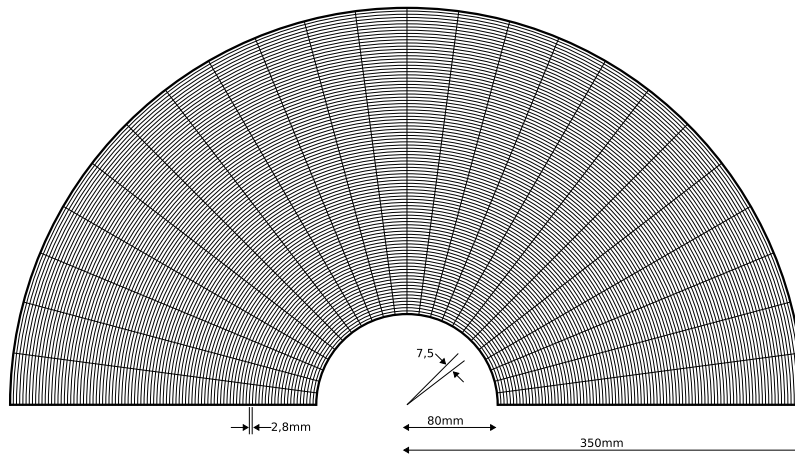


Rysunek 2.2. Model złącza p^+n (a) rozkład ładunku (b) rozkład pola elektrycznego (c) rozkład potencjału

W praktycznych zastosowaniach grubość sensora mieści się w przedziale od 100 do 500 μm . Granice te są determinowane przez możliwości obecnej technologii półprzewodnikowej. Warto zauważyć, że grube sensory zapewniają większą depozycję ładunku, mają dłuższe czasy zbierania oraz mniejszą pojemność w porównaniu do sensorów cienkich. Ponieważ spodziewane depozycje ładunku w detektorze LumiCal są dość znaczne, kompromisowym rozwiązaniem pomiędzy pojemnością, a ilością generowanego ładunku było zastosowanie standardowych sensorów o grubości 300 μm [BDD⁺07]. Symulacje fizyczne pokazały, iż maksymalny sygnał generowany przez wysokoenergetyczną kaskadę cząstek w pojedynczym padzie sensora będzie się mieścił poniżej 15 pC. Całkowity czas zbierania ładunku jest rzędu kilku nanosekund. Czas ten jest zależny od własności elektrycznych półprzewodnika, grubości sensora, jak również od przyłożonego napięcia polaryzującego. Przykładając napięcia znacznie większe niż napięcie pełnego zubożenia, można znacząco skrócić czas zbierania ładunku [Kor05]. Zbyt duże zwiększanie napięcia polaryzującego prowadzi jednak do wzrostu prądu upływu, a w rezultacie do zwiększenia szumów.

Dodatkową zaletą sensorów półprzewodnikowych jest to, iż wykonywane są w procesach technologicznych bardzo podobnych do tych, w których wytwarza się

układy scalone. Dzięki temu możliwe jest wykonywanie wielu sensorów o niemalże dowolnych kształtach, na jednym wafle krzemu (ang. silicon wafer). Studia optymalizacyjne wykazały, że dla detektora LumiCal optymalne rozmieszczenie padów powinno wyglądać jak na rysunku 2.3.



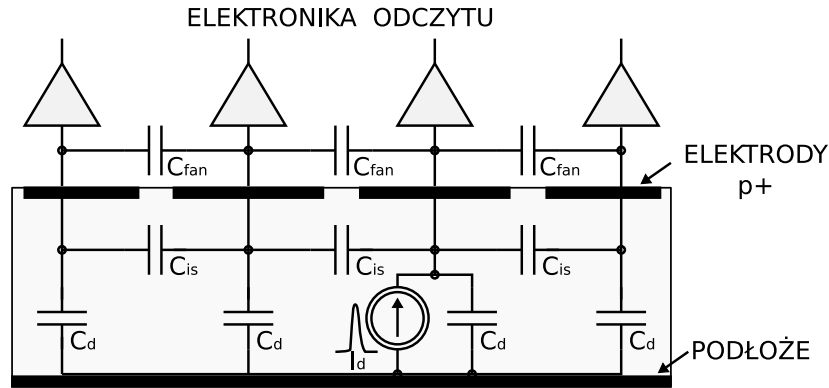
Rysunek 2.3. Proponowana segmentacja sensora krzemowego. Warstwa podzielona jest na 24 sektory, każdy z sektorów składa się z 96 padów sensora.

Pojedyncza warstwa składa się z 24 sektorów radialnych oraz 96 azymutalnych. Każdy pad odczytywany jest oddzielnie. Taka konfiguracja zapewnia dobrą przestrzenną i energetyczną zdolność rozdzielczą.

Pady podłączane są do elektroniki odczytu od strony domieszkowanej półprzewodnikiem typu p+, a do podłoża (ang. backplane) podłączone jest napięcie polaryzujące wspólne dla wszystkich padów. Istnieją dwa sposoby połączenia sensorów z elektroniką odczytu: stałoprądowe (DC) i zmiennoprądowe (AC). Sprzężenie stałoprądowe, charakteryzujące się bezpośrednim podłączeniem padu do wejścia elektroniki odczytu, wymusza na elektronice konieczność przyjęcia całego prądu upływu sensora i zapewnienia poprawnej pracy dla szerokiego zakresu tych prądów. Przy sprzężeniu typu AC, prądy upływu nie wpływają do elektroniki odczytu, dzięki zastosowaniu pojemności odsprzęgającej pomiędzy sensorem, a wejściem elektroniki. Rozwiązanie to wymaga jednak dodania rezystora o dużej wartości, polaryzującego stopień wejściowy, co okupione jest pojawieniem się w układzie kolejnej, długiej stałej czasowej. To z kolei, przy wysokich częstościach zliczeń, prowadzi do spiętrzenia się impulsów (ang. pile up) oraz zwiększenia całkowitych szumów systemu.

W detektorze LumiCal, ze względu na oczekiwane niewielkie prądy upływu ($\sim 1 \text{ nA/cm}^2$), oraz wysoką częstość sygnałów zdecydowano się na stałoprądowe (DC) sprzężenie sensorów z elektroniką.

Całkowita pojemność C_d padu odczytowego, pracującego przy napięciu pełnego zubożenia, jest pojemnością geometryczną wynikającą z rozmiarów, odległości okładek i stałej dielektrycznej półprzewodnika [Mar84]. Model wielopadowego sensora został zaprezentowany na rysunku 2.4.



Rysunek 2.4. Schemat zastępczy krzemowego sensora promieniowania wraz z elektroniką odczytu.

Źródło prądowe I_d symbolizuje impuls prądowy wygenerowany przez cząstkę jonizującą na zadanym padzie odczytowym. Pojemności C_{is} oraz C_{fan} to pojemności pomiędzy sąsiednimi padami. Pojemność C_{is} jest determinowana przez wzajemną geometrię padów podczas gdy C_{fan} wynika ze sprzężenia linii sygnałowych doprowadzających sygnał do elektroniki odczytu (ang. fanout). W pierwszym przybliżeniu założyć można, iż każdy tor elektroniki widzi pojemność własnego padu oraz sprzężenia do najbliższych sąsiadów.

Ze względu na geometrie padów, w detektorze LumiCal wartości pojemności mieszczą się w przedziale od 10 pF do 100 pF [BDD⁺07]. Pojemności międzypadowe wynikające z tzw. fanout-u mogą osiągać nawet 20 pF, co w istotny sposób zmienia sumaryczną pojemność na wejściu pojedynczego kanału. Tak duże wartości pojemności całkowite, jak również ich szeroki przedział zmienności, stawiają bardzo trudne wymagania dla elektroniki odczytu.

2.2. Wzmacniacz ładunkoczuły

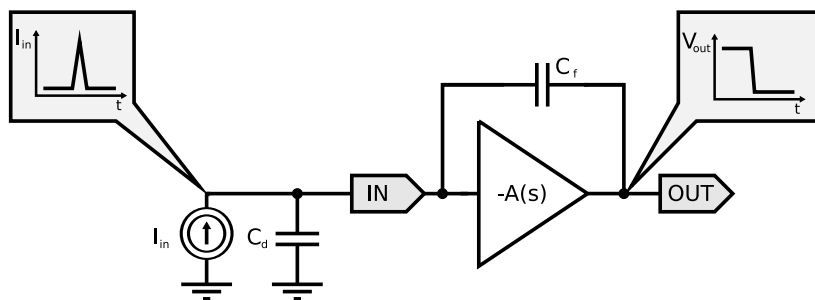
Opisana w poprzednim rozdziale architektura sensorów detektora LumiCal, jak również specyfika eksperymentu ILC, mają bezpośredni wpływ na wymagania dotyczące elektroniki odczytu. Najważniejsze wynikające stąd konsekwencje można wyrazić w postaci następującej specyfikacji:

- bardzo szeroki zakres pojemności wejściowych 10 – 100 pF pojedynczego kanału,
- bardzo szeroki zakres ładunków wejściowych, od 2 fC do około 15 pC, dla pojedynczego kanału,

- krótki czas pomiędzy zderzeniami wiązek oraz wysoka okupancja implikują konieczność przetwarzania jednego przypadku w czasie krótszym niż około 350 ns,
- niskie szумы wymuszone przez konieczność pomiaru cząstek minimalnie jonizujących (MIP), dla których stosunek sygnału do szumu (SNR) powinien wynosić co najmniej 10 (nawet dla największych pojemności wejściowych),
- mała dyssypacja mocy. Stabilność termiczna jest bardzo ważna ze względu na pozycjonowanie sensorów. Na obecnym etapie prac oszacowano, iż maksymalna moc rozpraszana przez jeden kanał nie powinna przekraczać 10 mW. Wymagania te mogą zostać zrelaksowane ze względu na charakter eksperymentu, pozwalający na wyłączenie elektroniki pomiędzy kolejnymi grupami zderzeń cząstek,
- odporność na zakłócenia pochodzące z bloków cyfrowych, jak również ze światła zewnętrznego.

Niestety większość z tych wymagań jest wzajemnie sprzecznych i znalezienie idealnego rozwiązania nie jest możliwe. Aby jak najlepiej spełnić wyżej wymienione wymagania, w szczególności niskoszumność oraz szeroki zakres pojemności wejściowych (tzw. ładunkoczułość), jako stopień wejściowy elektroniki odczytu zdecydowano się użyć wzmacniacza ładunkoczułego [Kor05, GM86].

Zadaniem przedwzmacniacza ładunkoczułego jest uformowanie odpowiedzi napięciowej $V(t)$ na wejściowy sygnał prądowy $I(t)$, o amplitudzie proporcjonalnej do ilości ładunku niesionego przez sygnał wejściowy. Podstawową konfigurację wzmacniacza ładunkoczułego przedstawiono na rysunku 2.5.



Rysunek 2.5. Schemat ideowy przedwzmacniacza ładunkoczułego

Układ przedwzmacniacza ładunkoczułego zawiera wzmacniacz odwracający (o funkcji przenoszenia $A(s)$) pracujący w trybie integratora, z pojemnością C_f w sprzężeniu zwrotnym, C_d symbolizuje tu pojemność sensora.

Funkcję przenoszenia takiego układu można zapisać jako:

$$V_{out}(s) = -\frac{I_{in}(s)}{C_f + \frac{C_d + C_f}{A(s)}}. \quad (2.1)$$

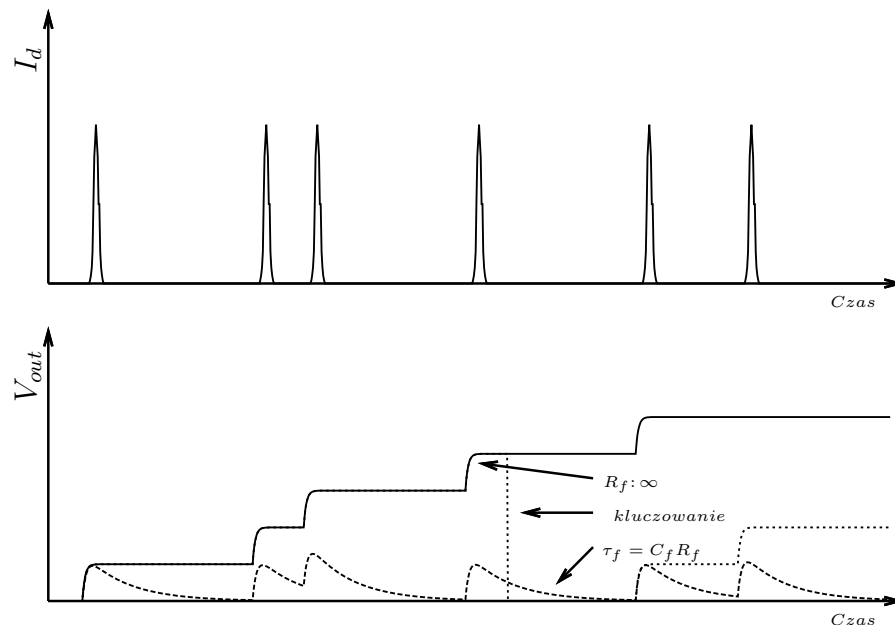
Zakładając, iż sygnał prądowy ma kształt delty Diraca, niosącej ładunek Q_{in} (co jest dobrym założeniem w wypadku gdy długość impulsu prądowego jest dużo

krótsza niż czas kształtowania układu) oraz, że wzmacniacz ma nieskończone wzmocnienie i pasmo przenoszenia, odpowiedź takiego układu jest skokiem jednostkowym o amplitudzie [Kor05]:

$$V_{out}(t) = -\frac{Q_{in}}{C_f}. \quad (2.2)$$

Po przetworzeniu sygnału ładunek zgromadzony na pojemności sprzężenia zwrotnego musi zostać rozładowany, gdyż w przeciwnym wypadku wyjście układu szybko się nasyci, osiągając potencjał zbliżony do jednej z linii zasilania (rys. 2.6). Rozładowanie wyjścia przedwzmacniacza może zostać zrealizowane w dwojaki sposób:

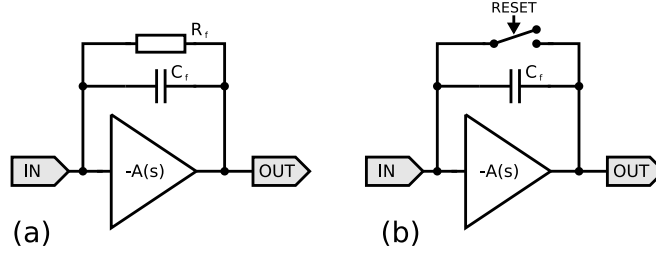
- ciągły, za pomocą rezystora (rys. 2.7a); wtedy impulsy na wyjściu układu znikają ze stałą czasową $\tau_f = R_f C_f$,
- nieciągły, za pomocą klucza (rys. 2.7b); dla odpowiednio małej rezystancji klucza, pojemność C_f rozładowuje się niemal natychmiast po przyjsciu sygnału zwierającego klucz.



Rysunek 2.6. Przykładowe odpowiedzi przedwzmacniacza dla różnych wartości rezystancji sprzężenia zwrotnego

Rozwiązanie z kluczem wymaga dodatkowej logiki podejmującej decyzję w którym momencie można wyzerować układ. Ponieważ układ ma być stosowany w eksperymencie synchronicznym (z dobrze określonymi czasami poszczególnych zdarzeń) wymaganie to jest proste do spełnienia. Rozwiązanie takie ma natomiast zalety w postaci zmniejszonych szumów oraz możliwości dynamicznej zmiany wzmocnienia co zostanie zaprezentowane w kolejnych rozdziałach.

Stosowane w praktyce wzmacniacze mają ograniczone pasmo przenoszenia i wzmocnienie, co oczywiście ma wpływ na impuls wyjściowy. W pierwszym przybliżeniu



Rysunek 2.7. Schematy rozładowywania pojemności sprzężenia zwrotnego wzmacniacza ładunkoczułego (a) rezystywna (b) kluczowana

wzmacniacz charakteryzuje się jednobiegunową funkcją przenoszenia, co można zapisać jako [Kor05]:

$$A(s) = \frac{A_0}{1 + s\tau_{p0}}, \quad (2.3)$$

gdzie A_0 jest wzmocnieniem dla bardzo małych częstotliwości, natomiast τ_{p0} jest stałą czasową bieguna dominującego wzmacniacza. Rozwiązując równania Kirchoffa dla takiego zastosowanego wzmacniacza otrzymujemy [Kor05]:

$$V_{out}(s) = \frac{I_{in}(s)Z_f(s)}{1 + \frac{1}{A(s)} + \frac{sC_dZ_f(s)}{A(s)}} \cong \frac{I_{in}(s)Z_f(s)}{1 + \frac{sC_dZ_f(s)}{A(s)}}, \quad (2.4)$$

gdzie $Z_f(s)$ jest impedancją sprzężenia zwrotnego wynoszącą:

$$Z_f(s) = \frac{R_f}{1 + s\tau_f}, \quad \tau_f = R_f C_f \quad (2.5)$$

Ogólne rozwiązanie równania 2.4 w dziedzinie czasu, w zależności od wartości elementów sprzężenia zwrotnego i pojemności wejściowej, można zapisać jako:

$$V_{out}(t) = \begin{cases} A \frac{e^{\left(\frac{t}{B}\right)} \sinh(t\sqrt{D})}{\sqrt{D}} & \text{dla } D > 0 \\ A e^{\left(\frac{t}{B}\right)} t & \text{dla } D = 0 \\ A \frac{e^{\left(\frac{t}{B}\right)} \sin(t\sqrt{-D})}{\sqrt{-D}} & \text{dla } D < 0 \end{cases}, \quad (2.6)$$

gdzie:

$$A = \frac{A_0}{C_d \tau_{p0}}, \quad B = \frac{2\tau_{p0}}{1 + \frac{C_d}{C_f} + A_0}, \quad D = \frac{(\tau_d + \tau_f A_0)^2 - 4\tau_d \tau_{p0} A_0}{(2\tau_d \tau_{p0})^2}, \quad \tau_d = R_f C_d. \quad (2.7)$$

Analiza powyższych formuł prowadzi do kilku istotnych wniosków. Przy czysto pojemnościowym sprzężeniu zwrotnym otrzymujemy:

$$V_{out}(t) = -Q_{in} \left(\frac{1 - e^{-2t/B}}{C_f + \frac{C_d}{A_0}} \right) \quad (2.8)$$

Na tym przykładzie bardzo ładnie widać wpływ wzmocnienia oraz pasma wzmacniacza na kształt impulsu. Zwiększając wzmocnienie wzmacniacza A_0 minimalizujemy stratę ładunku na pojemność sensora C_d , natomiast większe pasmo przyspiesza odpowiedź. Zmierzając z tymi wielkościami do nieskończoności otrzymujemy równanie 2.2.

Niestety, zwiększając pasmo oraz wzmocnienie można doprowadzić do zmiany znaku współczynnika D na ujemny, co doprowadzi do oscylujących odpowiedzi. Fakt ten nabiera znaczenia gdy w analizach uwzględnimy drugi biegun wzmacniacza (nie-dominujący). Ze względu na skomplikowane formuły matematyczne zrezygnowano ze szczegółowej analizy przypadku z dwoma biegunami. Oszacowanie wykonano w symulacjach (rys. 3.10).

Aby wzmocnienie ładunkowe układu nie zależało w sposób istotny od pojemności wejściowej (zdominowanej przez pojemność sensora oraz fanout), wzmacniacz musi się charakteryzować odpowiednim wzmocnieniem. Zaniedbując ograniczone pasmo wzmacniacza oraz rezystancję sprzężenia zwrotnego, wzmocnienie ładunkowe omawianego układu można zapisać jako (2.1):

$$k_q = \frac{V_{out}}{Q_{in}} = \frac{1}{C_f + \frac{C_d + C_f}{A_0}}. \quad (2.9)$$

Wynikającą stąd względną zmianę wzmocnienia wynikającą ze zmian C_d można zapisać jako:

$$\sigma_k \equiv \frac{dk_q}{k_q} = \frac{dC_d}{(C_f + C_d) + A_0 C_f}. \quad (2.10)$$

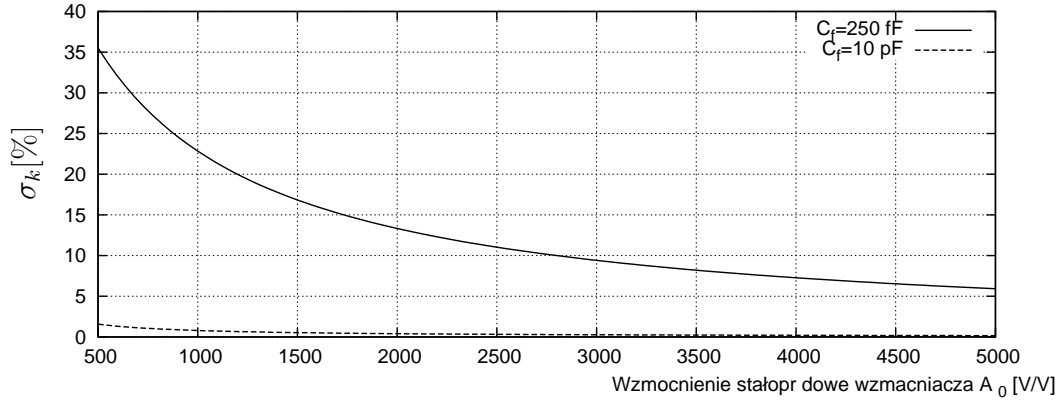
Widzimy, iż zmiany wzmocnienia układu będą tym większe im mniejsza pojemność sprzężenia zwrotnego i im mniejsze wzmocnienie wzmacniacza. Przykładowe zmiany wzmocnienia ładunkowego (wynikające z równania 2.10), dla pojemności spodziewanych w detektorze LumiCal zostały zaprezentowane na rysunku 2.8.

Z powyższego rysunku widać jak ogromny wpływ na zmiany wzmocnienia ładunkowego (przy możliwym do uzyskania w praktyce wzmocnieniu wzmacniacza) ma wartość pojemności sprzężenia zwrotnego.

2.2.1. Parametry małosygnalowe

Kolejnym bardzo ważnym parametrem jest impedancja wejściowa kanału odczytowego (przedwzmacniacza). Parametr ten jest istotny, gdyż determinuje on poziom przesłuchów pomiędzy kanałami (ang. crosstalk). Zjawisko przesłuchu ma znaczenie fundamentalne w systemach wielokanałowych, takich jak projektowany system odczytowy.

Impedancja wejściowa dana jest przez stosunek napięcia wejściowego do prądu wpływającego do wzmacniacza. Ponieważ stopniem wejściowym jest tranzystor polowy, w pierwszym przybliżeniu możemy przyjąć, że jego impedancja wynika tylko z pojemności bramki tranzystora wejściowego. Impedancję wejściową całego układu



Rysunek 2.8. Zależność rozrzutu wzmocnienia ładunkowego w funkcji wzmocnienia wzmacniacza pomiędzy pojemnością $C_d = 20$ pF, a $C_d = 100$ pF, dla różnych pojemności C_f

można wyznaczyć korzystając z klasycznej teorii ujemnego sprzężenia zwrotnego [GHLM01] jako:

$$Z_{in}(s) = \frac{1}{sC_d} \parallel \frac{Z_f(s)}{1 + A(s)} \cong \frac{R_f(1 + s\tau_{p0})}{A_0(1 + s\tau_f)}. \quad (2.11)$$

Moduł $|Z_{in}(\omega)|$ możemy wyznaczyć jako:

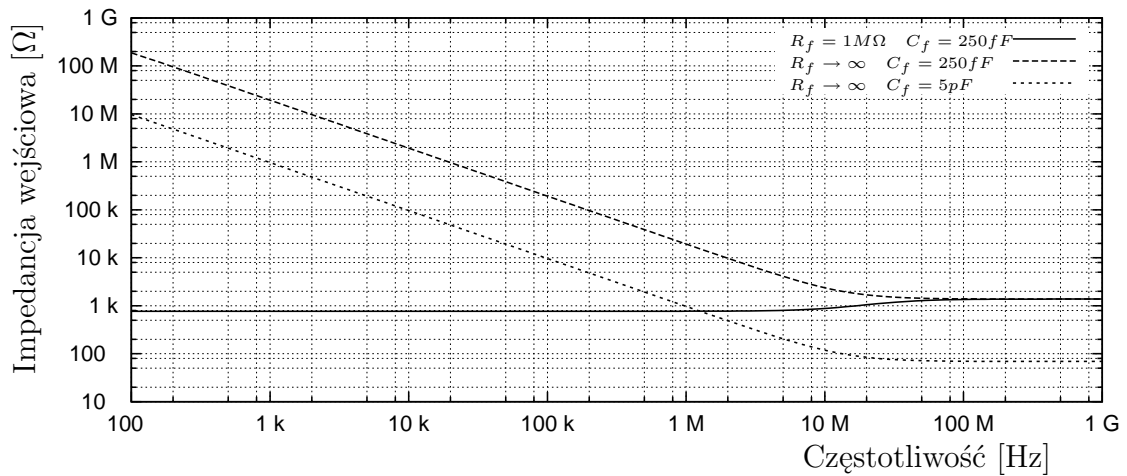
$$|Z_{in}(\omega)| = \frac{R_f \sqrt{1 + \omega^2 \tau_{p0}^2}}{A_0 \sqrt{1 + \omega^2 \tau_f^2}}. \quad (2.12)$$

Z powyższej zależności płyną następujące wnioski: w celu minimalizacji impedancji wejściowej należy maksymalizować wzmocnienie oraz pasmo. Zmniejszanie rezystancji sprzężenia zwrotnego zmniejsza również impedancje wejściową dla niskich częstotliwości. Wartości impedancji wejściowych dla kilku przykładowych zestawów parametrów zaprezentowano na rysunku 2.9.

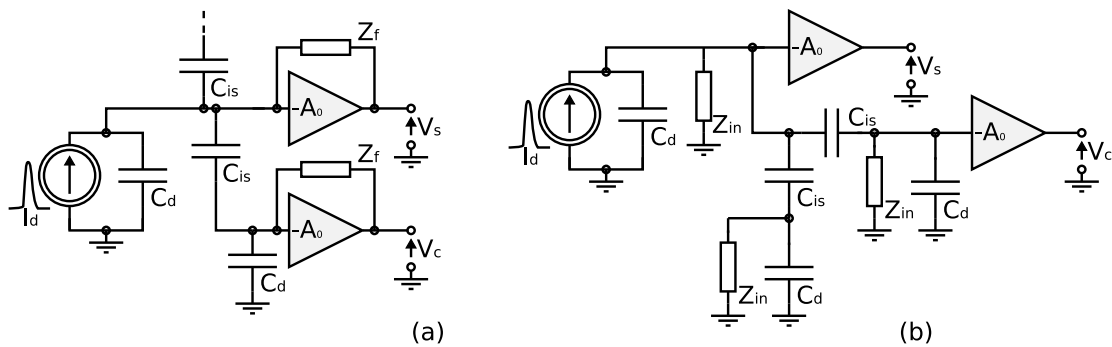
Aby oszacować przesłuchy wykorzystać można najprostszy model zawierający tylko pojemności geometryczne sensora do podłoża C_d oraz pojemności międzypadowe C_{is} (rysunek 2.10a). Wartości pojemności międzypadowych C_{is} wynikać będą z położenia padów względem siebie oraz fanout-u (przewodu łączącego sensor z elektroniką). W praktycznych realizacjach najistotniejsze są sprzężenia z najbliższymi sąsiadami. Korzystając ze schematu zastępczego zaprezentowanego na rysunku 2.10b można wyznaczyć względną wartość przesłuchu jako:

$$\left| \frac{v_{crosstalk}(\omega)}{v_{signal}(\omega)} \right| = \left| \frac{Z_{in}(\omega)}{Z_{in}(\omega) + \frac{1}{\omega C_{is}}} \right|. \quad (2.13)$$

Z przedstawionej zależności jasno wynika, iż w celu minimalizacji przesłuchów pomiędzy kanałami należy minimalizować C_{is} co, w wypadku określonej geometrii sen-



Rysunek 2.9. Zależność impedancji wejściowej od częstotliwości. Wzmocnienie wzmacniacza A_0 wynosi 1300 a $\tau_{p0} = 450$ ns

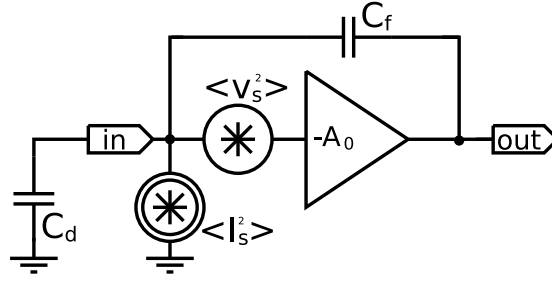


Rysunek 2.10. (a) Model wykorzystywany do obliczania przesłuchów wynikających z rozprygu ładunku pomiędzy sąsiednimi padami sensora oraz (b) schemat zastępczy

sora, zwykle nie jest możliwe. Przy ustalonej C_{is} najmniejsze przesłuchy otrzymamy minimalizując impedancję wejściową wzmacniacza.

2.2.2. Analiza szumowa

W zastosowaniach spektrometrycznych bardzo istotne są szумы. Wprowadzają one nieoznaczoność amplitudy impulsu wyjściowego, a tym samym wyznaczają minimalną amplitudę sygnałów możliwych do wykrycia. Szczegółowe rozważania dotyczące szumów w układach elektronicznych można znaleźć w pracach [Kor05, GM86, Gaj07]. W niniejszej pracy zostaną zaprezentowane tylko metody modelowania oraz kluczowe wnioski. Schemat szumowy dla rozważanej konfiguracji wzmacniacza ładunkoczułego został zaprezentowany na rysunku 2.11. W modelu tym można wyróżnić dwa źródła szumów, które zostały wyodrębnione ze struktury wzmacniacza.



Rysunek 2.11. Zastępczy schemat szumowy wzmacniacza ładunkoczułego

Posunięcie takie implikuje, iż wzmacniacz można traktować jako idealny, tj. pozbawiony źródeł szumów. Oba źródłom przypisać można gęstości widmowe mocy szumów, odpowiednio prądowemu:

$$\frac{d \langle I_p^2 \rangle}{df} = a, \quad (2.14)$$

oraz napięciowemu:

$$\frac{d \langle V_s^2 \rangle}{df} = b + \frac{A_f}{f}, \quad (2.15)$$

gdzie a i b są współczynnikami określającymi gęstość widmową szumów białych, natomiast A_f jest stałą charakteryzującą szum nadmiarowy (tzw. szum $1/f$). Wartości powyższych współczynników determinowane są wybraną konfiguracją układową, warunkami polaryzacji poszczególnych elementów oraz procesem technologicznym, w jakim dany układ jest wykonywany.

Korzystając z prawa przenoszenia szumów, można policzyć gęstość widmową szumów wyjściowych:

$$S_{out}(\omega) = S_{in}(\omega) |H(\omega)|^2, \quad (2.16)$$

Szum prądowy przetransmitowany na wyjście układu wzmacniacza obecny jest jako:

$$\frac{d \langle V_p^2 \rangle}{df} = a \left[\frac{A_0}{C_f(A_0 + 1) + C_d \omega} \right]^2 \simeq a \left[\frac{1}{\omega C_f} \right]^2. \quad (2.17)$$

Podobnie gęstość widmową szumu napięciowego na wyjściu można wyznaczyć jako:

$$\frac{d \langle V_s \rangle^2}{df} = \left[b + \frac{A_f}{f} \right] \left[\frac{A_0(C_f + C_d)}{C_f(A_0 + 1) + C_d} \right]^2 \simeq \left[b + \frac{A_f}{f} \right] \left[\frac{C_f + C_d}{C_f} \right]^2. \quad (2.18)$$

Całkowity szum na wyjściu przedwzmacniacza dany jest superpozycją szumu prądowego oraz napięciowego [Kor05]:

$$\frac{d \langle V_{pre} \rangle^2}{df} = \left[\frac{C_f + C_d}{C_f} \right]^2 \left[b + \frac{A_f}{f} + \frac{a}{\omega^2(C_f + C_d)^2} \right]. \quad (2.19)$$

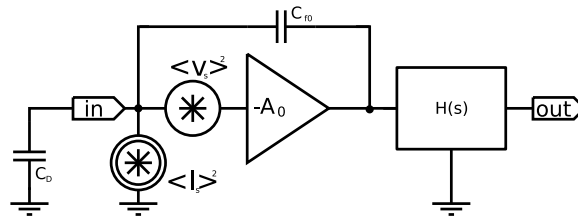
Wartość średniokwadratową napięcia szumów na wyjściu przedwzmacniacza można policzyć całkując je w pełnym paśmie częstotliwości, a mianowicie:

$$V_{prerms}^2 = \int_0^\infty \frac{d \langle V_{pre} \rangle^2}{df} df. \quad (2.20)$$

Jak można wywnioskować ze wzoru 2.20 całkowity szum na wyjściu idealnego przedwzmacniacza ładunkowego powinien dążyć do nieskończoności. Ponieważ w rzeczywistości obserwowany szum jest zawsze skończony można się domyślić, że ograniczenie górnej i dolnej części pasma będzie wynikało tak ze skończonego pasma wzmacniacza jak i istnienia elementów pasożytniczych, nie uwzględnionych w rozważanym schemacie.

2.3. Kształtowanie impulsów i filtracja szumów

W praktycznych zastosowaniach stosunek sygnału do szumu na wyjściu przedwzmacniacza jest niewystarczający i konieczna jest jego poprawa. W tym celu stosuje się odpowiednie układy filtrów, które z jednej strony zapewniają pożądany stosunek sygnału do szumu, a z drugiej nadają sygnałowi wyjściowemu odpowiedni kształt (należy tu pamiętać, że w idealnym przypadku sygnał na wyjściu przedwzmacniacza jest skokiem napięciowym). Zakładając, że układ kształtowania cha-



Rysunek 2.12. Zastępczy schemat szumowy wzmacniacza ładunkowego wraz z blokiem filtra o transmitancji $H(s)$

rakteryzuje się funkcją przenoszenia $H(s)$ (rys. 2.12), można policzyć gęstość widmową szumów na wyjściu układu kształtującego:

$$\frac{d \langle V_{sh} \rangle^2}{df} = \left(\frac{d \langle V_s \rangle^2}{df} + \frac{d \langle V_p \rangle^2}{df} \right) |H(f)|^2, \quad (2.21)$$

a także wartość średniokwadratową szumów wyjściowych:

$$V_{shrms}^2 = \int_0^\infty \left[\left(\frac{d \langle V_s \rangle^2}{df} + \frac{d \langle V_p \rangle^2}{df} \right) |H(f)|^2 \right] df. \quad (2.22)$$

Na podstawie powyższych zależności można otrzymać finalny stosunek sygnału do szumu:

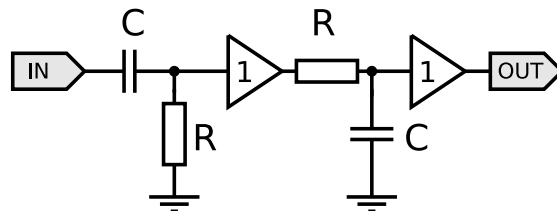
$$S/N = \frac{V_{omax}}{\sqrt{V_{shrms}^2}}, \quad (2.23)$$

gdzie V_{omax} jest amplitudą sygnału referencyjnego, w naszym przypadku odpowiadającą sygnałowi wygenerowanemu przez cząstkę minimalnie jonizującą.

Dla potrzeb detektora LumiCal, studiowane były dwie koncepcje kształtowania sygnału, a mianowicie kształtowanie ciągle oraz nieciągle w czasie. Klasycznym i najczęściej stosowanym sposobem kształtowania ciągłego jest filtracja typu $(CR) - (RC)^n$, w przypadku której otrzymujemy na wyjściu tzw. impulsy pseudo-gaussowskie. Zostanie ona opisana w pierwszym podrozdziale, dla przypadku $n=1$. W sytuacji, gdy oczekiwana jest duża średnia częstotliwość impulsów wejściowych z sensora, klasyczne kształtowanie pseudo-gaussowskie okazuje się niewystarczające. Wynika to z powstawania długich „ogonów“ impulsów wyjściowych, na skutek długiej stałej czasowej zaniku sygnału na wyjściu przedwzmacniacza. Powyższa stała czasowa związana jest z istnieniem dużej rezystancji R_f w sprzężeniu zwrotnym. Zachowanie takie istotnie degraduje rozdzielczość amplitudową, z uwagi na spiętrzające się impulsy (ang. pile up). Z tego względu stosuje się kształtowanie zawierające tzw. układy kompensacji biegun-zero PZC (ang. Pole-Zero Cancellation). Kształtowanie takie jest bardzo zbliżone do pseudo-gaussowskiego, z tą różnicą, że eliminuje ono długie ogony impulsów. Ten typ filtracji rozważany jest do zastosowania w detektorze LumiCal i z tego względu w pierwszych prototypach elektroniki front-end zaimplementowane zostały kanały z PZC. Wybrana architektura stosująca tego typu kształtowanie omówiona została w drugim podrozdziale. Poza typową filtracją rozważana jest też koncepcja kształtowania nieciągłego w czasie. Powody oraz początkowe prace z tym związane omówione są w trzecim podrozdziale.

2.3.1. Filtracja (CR)-(RC)

Pierwsza z rozważanych koncepcji zakłada użycie układu aktywnego filtru środkowo-przepustowego (rys. 2.13a), tzn. (CR)-(RC) o identycznych wartościach stałych czasowych $\tau = RC$.



Rysunek 2.13. Schemat filtru (CR)-(RC)

Transmitancja takiego filtru wynosi:

$$F_{crrc}(s) = \frac{\tau s}{(1 + \tau s)^2}, \quad (2.24)$$

a co za tym idzie jej moduł:

$$|F_{crrc}(\omega)| = \frac{\omega\tau}{1 + (\omega\tau)^2}. \quad (2.25)$$

Efektom filtracji gęstości widmowej obecnej na wyjściu przedwzmacniacza filtrem o transmitancji (2.24) jest:

$$\frac{d \langle V_N^2 \rangle}{df} = \left\{ \frac{a}{\omega^2} \left[\frac{A_0}{C_f(A_0 + 1) + C_d} \right]^2 + \left[b + \frac{A_f}{f} \right] \left[\frac{A_0(C_f + C_d)}{C_f(A_0 + 1) + C_d} \right]^2 \right\} \left[\frac{\omega\tau}{1 + (\omega\tau)^2} \right]^2. \quad (2.26)$$

Scałkowanie powyższego równania w całym paśmie częstotliwości (od zera do nieskończoności) prowadzi do wyrażenia na wariancję szumu wyjściowego:

$$V_N^2 = \frac{\frac{a\tau}{8} + (C_f + C_d)^2 \left[\frac{b}{8\tau} + \frac{A_f}{2} \right]}{\left(C_f + \frac{C_f + C_d}{A_0} \right)^2}. \quad (2.27)$$

Bardzo często używa się alternatywnego parametru mówiącego o poziomie szumów układu. Jest to tak zwany stosunek sygnału (o zadanej amplitudzie) do szumu S/N. Aby wyznaczyć ten stosunek konieczna jest znajomość amplitudy impulsu na wyjściu toru. W dziedzinie operatorowej odpowiedź ta równa się iloczynowi transmitancji układu przedwzmacniacza (2.1) oraz układu kształtowania (2.24). W omawianym przypadku odpowiedź na impuls prądowy niosący ładunek Q_{in} ma postać:

$$V_{out}(s) = -Q_{in} \frac{A_0}{s(C_f(A_0 + 1) + C_d)} \frac{s\tau}{(1 + \tau s)^2}, \quad (2.28)$$

co w dziedzinie czasu wynosi:

$$V_{out}(t) = -\frac{Q_{in}A_0}{C_f(A_0 + 1) + C_d} \frac{t}{\tau} e^{-\frac{t}{\tau}}. \quad (2.29)$$

W chwili $t = \tau$ powyższa funkcja osiąga maksimum V_{omax} :

$$V_{omax} = -\frac{Q_{in}A_0}{C_f(A_0 + 1) + C_d} \frac{1}{e}. \quad (2.30)$$

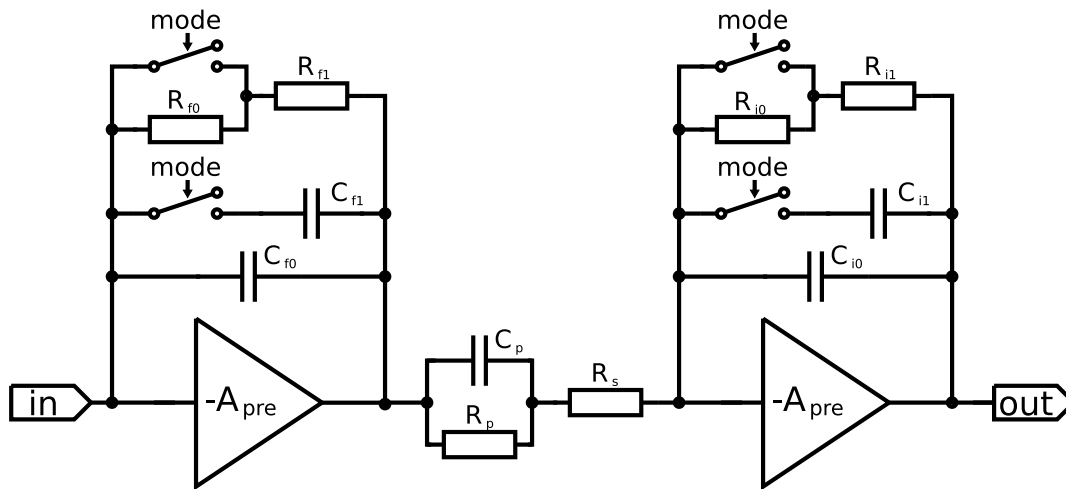
Stosunek sygnału do szumu może zostać wyznaczony jako stosunek maksimum amplitudy dla zadanego sygnału (2.30) do pierwiastka z wartości średniokwadratowej szumu na wyjściu (2.27):

$$S/N = \frac{V_{omax}}{\sqrt{V_N^2}} \simeq \frac{Q_{in}}{\sqrt{a\tau + \frac{b}{\tau}(C_f + C_d)^2 + 4A_f(C_f + C_d)^2}}. \quad (2.31)$$

Z powyższej zależności wynika, iż dla zadanych wartości stałej czasowej filtru τ (wymuszonej przez eksperyment) można optymalizować wkład do całkowitych szumów pochodzących od gęstości widmowych szumów prądowych jak i napięciowych.

2.3.2. Filtracja z PZC

Główną wadą typowej filtracji $(CR) - (RC)^n$ jest powstawanie długich ogonów impulsów, wynikających z bieguna w funkcji przenoszenia, związanego z dużą stałą czasową elementów sprzężenia zwrotnego przedwzmacniacza $\tau_f = R_f C_f$. Aby wyeliminować tę wadę wprowadza się w funkcję przenoszenia układu kształtującego dodatkowe zero, o stałej czasowej dokładnie równej $R_f C_f$. W ten sposób w funkcji przenoszenia układu zostają tylko bieguny układu kształtującego, o krótkich stałych czasowych, natomiast niepożądany biegun $R_f C_f$ skraca się z wprowadzonym zerem. Układową implementacją takiego kształtowania, zastosowaną w strukturach prototypowych, przedstawiono na rysunku 2.14.



Rysunek 2.14. Schemat układu elektroniki front-end z ciągłym kształtowaniem. Na rysunku tym A_{pre} wraz ze sprzężeniem zwrotnym tworzy przedwzmacniacz ładunkoczuły, elementy $R_p C_p$ zapewniają zero dla kompensacji PZC, a dalsza część tworzy układ kształtujący

Zakładając idealność zastosowanych wzmacniaczy, po prostych przekształceniach, otrzymuje się funkcję przenoszenia układu w postaci [IKGP08]:

$$\frac{U_{out}(s)}{I_{in}(s)} = \frac{1}{C_f C_i R_s} \cdot \frac{s + 1/C_p R_p}{s + 1/C_f R_f} \cdot \frac{1}{(s + 1/C_i R_i)(s + 1/C_p (R_p || R_s))}. \quad (2.32)$$

Jeżeli dobierzemy elementy układu tak, by $C_p R_p = C_f R_f$ oraz $C_i R_i = C_p (R_p || R_s)$, to otrzymamy funkcję przenoszenia równoważną idealnemu układowi $(CR)-(RC)$. Zatem kształtowanie takie daje omawiany już wcześniej impuls pseudo-gaussowski, pozbywając się przy tym niewygodnego ogona.

Na przedstawionym schemacie (rys. 2.14) widoczne są klucze przy rezystancjach i pojemnościach sprzężenia zwrotnego przedwzmacniacza i układu kształtującego. Poprzez zmianę stałych czasowych w sprzężeniu zwrotnym realizowana jest zmiana

wzmocnienia układu pomiędzy tzw. trybem kalibracyjnym (o dużym wzmocnieniu) oraz trybem fizycznym (o małym wzmocnieniu). W trybie kalibracyjnym układ ma być czuły na depozycje małych ładunków w zakresie od dwóch fC do kilkuset fC. Głównie chodzi tu o detekcję MIP-ów deponujących w sensorze ładunek kilku fC. Tryb fizyczny natomiast służy do normalnej pracy, podczas której następują ogromne depozycje ładunków z tzw. kaskad elektromagnetycznych. W trybie tym powinny być widziane sygnały od kilkudziesięciu fC do około 10 pC. Widać zatem wyzwanie związane z potrzebą detekcji sygnału, którego całkowita dynamika obejmuje cztery rzędy wielkości.

Dla celów detektora LumiCal zaprojektowano dwie wersje prototypowych układów tego rodzaju, różniące się tylko sposobem implementacji rezystancji sprzężenia zwrotnego R_f oraz R_p . W pierwszej wersji rezystancję tą wykonano w sposób pasywny, używając dostępnych w technologii rezystorów z krzemu polikrystalicznego. W drugiej wersji rezystancję tą otrzymano z aktywnych elementów, a mianowicie tranzystorów MOS pracujących w zakresie liniowym. Zrobiono tak ze względu na trudności w otrzymaniu wysokich wartości rezystorów pasywnych w rozważanej technologii. Gdyby nie te trudności oraz pojemności pasożytnicze rosnące wraz z wartością rezystancji, rezystory pasywne byłyby zdecydowanie najkorzystniejszą opcją, ze względu na dobre dopasowanie pomiędzy technologicznymi wartościami R_f i R_p .

2.3.3. Filtracja nieciągła w czasie

Filtracja dyskretna stosowana jest dosyć rzadko w wielokanałowych układach ekstrakcji sygnału z detektorów cząstek. Dzieje się tak z kilku powodów, takich jak: zwiększony stopień komplikacji układu; konieczność współdziałania bloków analogowych i cyfrowych, co wiąże się z niebezpieczeństwem przesłuchów sygnałów cyfrowych do części analogowej; zwiększony pobór mocy; zwiększona powierzchnia układu. W przypadku detektora LumiCal głównym powodem, dla którego rozważana jest taka alternatywa, jest dyskutowane obecnie wymaganie rejestracji cząstek minimalnie jonizujących „online”. Gdyby rzeczywiście zaszła taka potrzeba, wtedy elektronika front-end musiałaby jednocześnie widzieć sygnały z zakresu trybu kalibracyjnego i fizycznego, tzn. sygnały w zakresie od pojedynczych fC aż do 10 pC. Oznacza to, że dla efektywnej detekcji sygnału należałoby zapewnić S/N rzędu 10 nawet w najtrudniejszych warunkach, tzn. dla minimalnego sygnału w kanale o największej pojemności sensora. A to oznacza, że przy pojemności sensora 100 pF szumy elektroniki front-end powinny wynosić ułamki fC i kanał ten powinien pracować liniowo dla sygnałów wejściowych aż do poziomu 10 pC. Mając na uwadze ograniczenia technologii (np. maksymalne napięcie 3,3 V) oraz ograniczoną moc na kanał, spełnienie takich wymagań wydaje się nierealne. Z tego względu rozważa się projekt układu elektroniki front-end o dynamicznie zmienianym wzmocnieniu. Oznacza to, że jeśli amplituda sygnału wejściowego wzrośnie powyżej pewnego poziomu progowego, układ powinien (bez straty informacji o sygnale przetworzonym

do chwili osiągnięcia amplitudy progowej) przełączyć się z trybu o dużym wzmacnieniu (kalibracyjnego) na tryb o małym wzmacnieniu (fizyczny). W ramach zespołów pracujących dla ILC tylko jednej grupie na świecie [Bre06] udało się do tej pory zaprojektować i zademonstrować poprawnie działającą architekturę tego typu. Mając na uwadze wymagania czasowe związane ze zderzeniami wiązek wydaje się, że dynamiczne przełączanie wzmacnienia można osiągnąć tylko przy zastosowaniu dyskretnej filtracji sygnału. Z tego względu zainicjowane zostały prace nad tą architekturą.

W najprostszej postaci kształtowanie nieciągłe w czasie można zrealizować za pomocą omówionego w rozdziale 2.2 przedwzmacniacza ładunkowego w wersji z kluczowanym sygnałem reset. Naturalnym pytaniem jest czy implementacja tego typu, w której praktycznie brak jest filtracji pozwoli na uzyskanie wystarczającego stosunku sygnału do szumu. Przy założonych w podrozdziale 2.2.2 gęstościach widmowych szumów próba obliczenia wartości średniokwadratowej szumów prowadzi do rozbieżności przy częstotliwości dążącej do zera. Jednak przyjęcie dolnej granicy całkowania już na poziomie pojedynczych Hz pozwala uzyskać wymagany stosunek sygnału do szumu na poziomie 10. Najprostszym sposobem wycięcia niskoczęstotliwościowej części widma (przy nieciągłym kształtowaniu) jest zastosowanie jednej z odmian techniki zwanej skorelowanym próbkowaniem podwójnym (ang. Correlated Double Sampling - CDS). Mając na uwadze sposób generacji wiązki elektron-pozyton, można by np. od wartości sygnału zapamiętanej dla danego zderzenia, odejmować wartość piedestału próbkowaną po zakończeniu kolejnej grupy zderzeń (dla przypomnienia, zderzenia następują co około 350 ns w grupie trwającej 1 ms, a potem jest 200 ms przerwy w wiązce). Innym sposobem implementacji CDS byłoby odejmowanie sygnału piedestału po każdym zderzeniu. W takim przypadku w każdym okresie (350 ns) pomiędzy zderzeniami wiązek następowałoby próbkowanie różnicy pomiędzy amplitudą na wyjściu po przyjsciu sygnału i amplitudą po wysłaniu sygnału reset (czyli powrocie napięcia na wyjściu do wartości piedestału). Jedną z wyżej wymienionych opcji CDS jest rozważana jako alternatywna architektura elektroniki front-end w detektorze LumiCal. Można sobie oczywiście wyobrazić jeszcze bardziej zaawansowane sposoby filtracji dyskretnej jak np. filtry na przełączanych pojemnościach, które pozwalają na kształtowanie sygnałów i pasma przenoszenia analogicznie jak to się dzieje w ciągłych filtrach analogowych. Wydaje się jednak, iż optymalnym rozwiązaniem, w sensie komplikacji układu, wydzielanej mocy i uzyskanego S/N będzie któraś z architektur stosujących CDS.

Jako pierwszą fazę rozwoju architektury z dyskretnym przetwarzaniem sygnału postanowiono zaprojektować przedwzmacniacz ładunkowy z kluczowanym sygnałem „reset” oraz ze zmiennym wzmacnieniem, kontrolowanym przez zewnętrzny sygnał sterujący. Układ taki miał pozwolić na analizę szumową działania CDS (przy próbkowaniu za pomocą zewnętrznego przetwornika ADC) oraz na przetestowanie dynamicznego przełączania progu (na razie za pomocą sygnału zewnętrznego). Projekt układu i wyniki pomiarów są szczegółowo opisane w rozdziale 3.

2.4. Przetwornik ADC

Kolejnym elementem na drodze sygnału w omawianym torze spektrometrycznym jest przetwornik analogowo-cyfrowy (ang. Analog Digital Converter). Symulacje detektora wykazały, iż do poprawnej rekonstrukcji interesujących przypadków fizycznych wymagana jest rozdzielczość 10 bitów [BDD⁺07]. Biorąc pod uwagę ogromną liczbę kanałów (około 200 000) oraz wymagania na moc i przestrzeń, jako optymalne rozwiązanie zdecydowano się wykonać wielokanałowy zintegrowany przetwornik ADC [BDD⁺07]. Obecnie są rozważane dwa schematy próbkowania: jeden z relatywnie wolnym przetwornikiem ADC na każdy kanał oraz drugi z szybszym przetwornikiem na grupę kanałów (np. 8). Pierwsza opcja jest prostsza z konstrukcyjnego punktu widzenia, podczas gdy druga pozwala zaoszczędzić powierzchnię i moc. W opcji z jednym ADC na kanał częstotliwość próbkowania wynosi około 3 Ms/s podczas gdy druga wymaga około 24 Ms/s. Dodatkowym aspektem ważnym z konstrukcyjnego punktu widzenia jest konieczność wykonania multipleksera sygnałów analogowych na wejściu przetwornika przy drugiej opcji.

2.4.1. Funkcja przenoszenia przetwornika

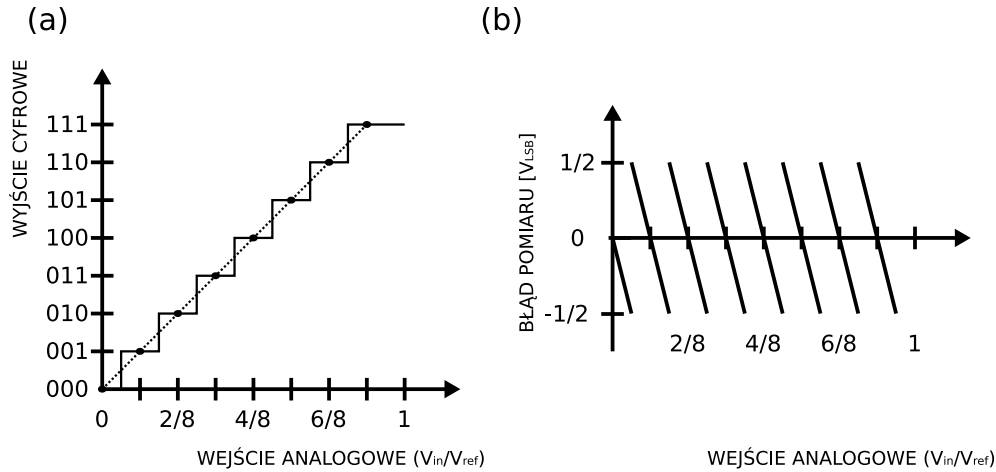
Idealny N-bitowy przetwornik ADC konwertuje ciągły, analogowy sygnał wejściowy do dyskretnych w czasie i wartościach próbek cyfrowych. Zależność kodów wyjściowych od napięcia wejściowego nazywamy funkcją przenoszenia danego przetwornika. Dla dynamiki sygnału wejściowego wynoszącej V_{fs} voltów zmiana sygnału wejściowego odpowiadająca najmniej znaczącemu bitowi (ang. Least Significant Bit – LSB) danego przetwornika ma wartość:

$$V_{LSB} = \frac{V_{fs}}{2^N}. \quad (2.33)$$

Przykładowa idealna funkcja przenoszenia 3-bitowego przetwornika analogowo-cyfrowego została zaprezentowana na rysunku 2.15a. W idealnym przetworniku łącząc środki kolejnych stopni otrzymuje się linię prostą. Każde odchylenie funkcji przenoszenia od prostej będzie powodowało błąd pomiaru.

2.4.2. Szum kwantyzacji i S/N idealnego ADC

Nawet w przypadku idealnego przetwornika analogowo-cyfrowego występuje jeden rodzaj błędu, który ma znaczenie fundamentalne, a rzadko kiedy pojawia się w specyfikacjach. Jest to błąd związany z procesem reprezentacji danych w postaci cyfrowej, tzn. błąd (szum) kwantyzacji. Proces kwantyzacji (zamiany sygnału analogowego na cyfrowy) wprowadza nieunikniony błąd. Dzieje się tak dlatego, iż zakresowi napięć wejściowych ($j \cdot V_{LSB} - 1/2V_{LSB}, j \cdot V_{LSB} + 1/2V_{LSB}$) zostanie przyporządkowany kod j . Błąd ten zawsze będzie mniejszy co do wartości bezwzględnej od $1/2V_{LSB}$. Wartość błędu kwantyzacji dla idealnego 3-bitowego przetwornika zaprezentowano na rysunku 2.15b.



Rysunek 2.15. (a) Funkcja przenoszenia idealnego przetwornika 3-bitowego (b) zależność błędu kwantyzacji od napięcia wejściowego

Zakładając, iż wszystkie poziomy sygnału wejściowego (po odpowiednio długim czasie) rozkładają się równomiernie na przedziale $(-1/2V_{LSB}, +1/2V_{LSB})$ wartość oczekiwana takiego rozkładu wynosi 0, a jego odchylenie standardowe [vdP01] to:

$$V_{q,rms} = \frac{1}{2\sqrt{3}}V_{LSB}. \quad (2.34)$$

Podając na wejście sygnał sinusoidalny o amplitudzie maksymalnej, tj. nie powodującej jeszcze nasycenia przetwornika, wartość napięcia międzyszczytowego wynosi:

$$V_{sig,pp} = 2^N V_{LSB}. \quad (2.35)$$

Wartość skuteczna takiego sygnału wynosi:

$$V_{sig,rms} = \frac{2^N V_{LSB}}{2\sqrt{2}}. \quad (2.36)$$

Wartość stosunku sygnału do szumu (S/N) definiuje się jako:

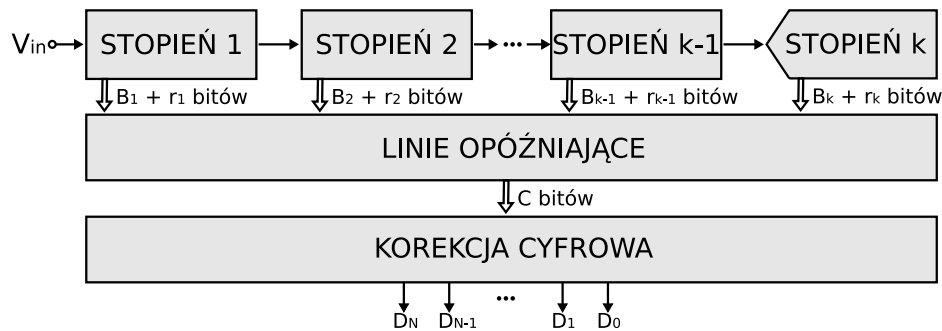
$$S/N = \frac{V_{sig,rms}}{V_{q,rms}} = 2^N \sqrt{1,5} = (6,02N + 1,76) \text{ dB} \quad (2.37)$$

Zatem dla idealnego N-bitowego przetwornika ADC $S/N = (6,02N + 1,76) \text{ dB}$, zaś w rzeczywistych przetwornikach jest on zawsze mniejszy od tej wartości.

Należy zwrócić uwagę na to, że przy modelowaniu błędu kwantyzacji jako procesu statystycznego można go nazwać szumem. Szum taki będzie szumem białym o gęstości widmowej rozłożonej równomiernie po całym paśmie.

2.4.3. ADC o architekturze potokowej

Jedną z najefektywniejszych architektur przetworników ADC zapewniającą kompromis pomiędzy szybkością próbkowania, zajmowaną powierzchnią oraz pobieraną mocą jest architektura potokowa (ang. pipeline ADC) [vdP01]. W przetwornikach potokowych, proces kwantyzacji rozłożony jest wzdłuż linii sygnału, jego zasadę można odnieść do znanej techniki programistycznej „dziel i zwyciężaj” [CLRS04]. Ogólny schemat przetwornika potokowego został zaprezentowany na rysunku 2.16. Składa się on z k stopni o niskiej rozdzielczości, linii opóźniających synchronizujących wyjścia z poszczególnych stopni oraz bloku korekcji cyfrowej.



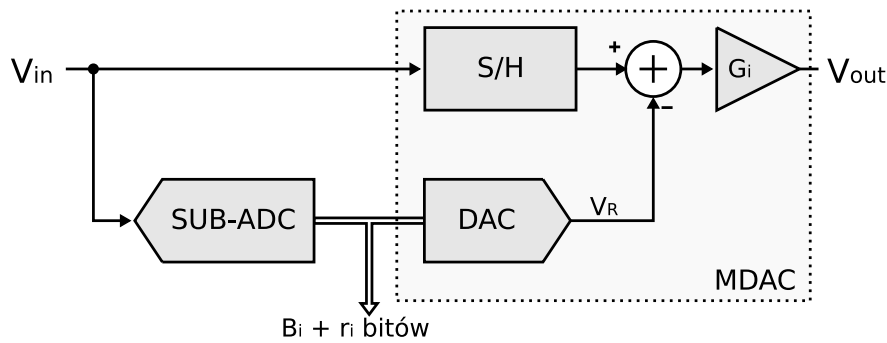
Rysunek 2.16. Architektura potokowego przetwornika analogowo-cyfrowego.

Każdy blok ma rozdzielczość $B_i + r_i$ bitów, gdzie B_i oznacza efektywną ilość bitów natomiast r_i ilość bitów redundantnych wykorzystywanych w algorytmie korekcji cyfrowej. Pierwsze $k - 1$ stopni przetwornika zazwyczaj ma tę samą liczbę bitów, tj. $B_1 = B_2 = \dots = B_{k-1} = B$. Podobnie jest z ilością bitów redundantnych $r_1 = r_2 = \dots = r_{k-1} = r$. Jeśli algorytm korekcji jest wykorzystywany $r \geq 1$. Ostatni stopień przetwornika zawiera zazwyczaj tylko układ komparatorów (subADC) nie wnoszących informacji redundantnej ($r_k = 0$). Całkowita rozdzielczość przetwornika o omawianej architekturze dana jest zależnością:

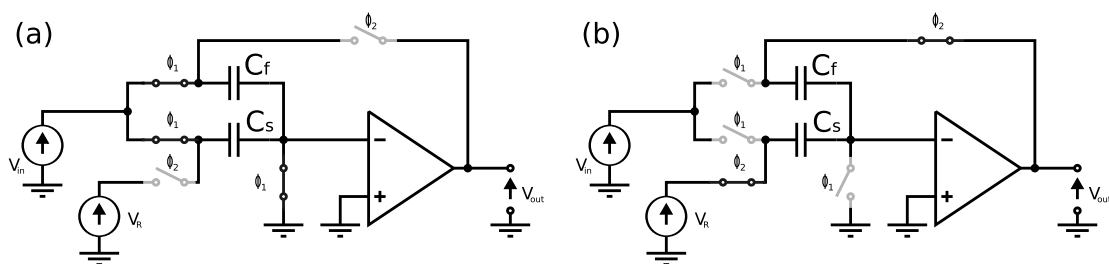
$$N = \sum_{j=1}^k B_j. \quad (2.38)$$

W większości obecnie projektowanych przetworników potokowych rozdzielczość stopnia B_j zawiera się w przedziale 1–4 bitów [Sum02].

Na rysunku 2.17 został przedstawiony pojedynczy stopień przetwornika potokowego. Stopień taki zawiera przetwornik analogowo-cyfrowy o niskiej rozdzielczości (ang. subADC) oraz jednostkę arytmetyczną zwaną często mnożącym przetwornikiem cyfrowo analogowym (ang. Multiplying Digital to Analog Converter – MDAC). Jednostka arytmetyczna odpowiedzialna jest za zapamiętanie wartości sygnału wejściowego (ang. sample and hold), odjęcie wyniku konwersji cyfrowo-analogowej oraz



Rysunek 2.17. Schemat pojedynczego stopnia przetwornika potokowego.

Rysunek 2.18. Fazy działania układu mnożąco-odejmującego (MDAC) (a) próbkowanie sygnału wejściowego (b) odejmowanie od zapamiętanej wartości V_R i mnożenie wyniku przez $(1 + C_f/C_s)$

przemnożenie wyniku tej operacji przez $G_i = 2^{B_i+1-r}$. Do tego celu najczęściej wykorzystuje się układ zaprezentowany na rysunku 2.18.

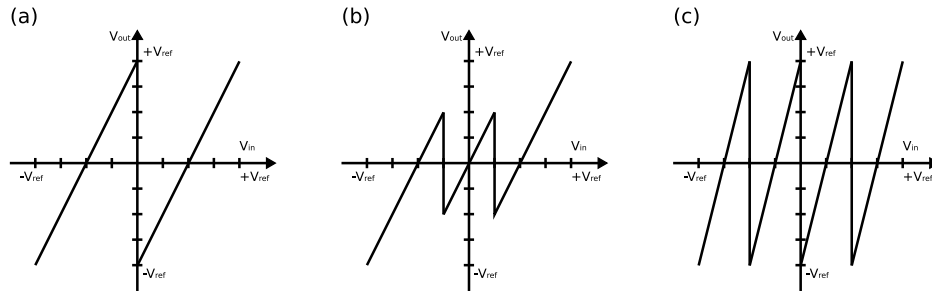
Układ taki pracuje w dwóch fazach. W fazie pierwszej ϕ_1 następuje zapamiętanie wartości napięcia na kondensatorach C_s oraz C_f . W drugiej fazie ϕ_2 następuje odejmowanie od zapamiętanego napięcia wejściowego napięcia V_R z równoczesnym mnożeniem wyniku przez $(1 + C_f/C_s)$. Aby uzyskać układ mnożący przez dwa należy zapewnić by $C_f = C_s$.

Wszystkie stopnie działają równocześnie. Podczas gdy pierwszy stopień operuje zawsze na najaktualniejszej próbce sygnału, kolejne stopnie operują na resztach z poprzednich próbek. Aby w pełni wykorzystać zasoby sprzętowe kolejne stopnie operują przy odwrotnych fazach zegara co schematycznie zostało pokazane na rysunku 4.14.

2.4.4. Korekcja cyfrowa

Najczęściej stosowanym algorytmem korekcji cyfrowej w potokowych przetwornikach analogowo-cyfrowych jest algorytm redundantnego kodowania znaku (ang.

redundant sign digit coding) z jednobitową redundancją. W algorytmie tym dodając jeden bit informacji więcej ($r_i = 1$) jednocześnie rezygnujemy z jednego poziomu kwantyzacji. Stopnie tego typu zwane są zwyczajowo $n + 0,5$ bitowymi. Przykładowo stopień 1,5-bitowy zwraca informację na 2 bitach jednak ma tylko 3 możliwe stany (2 komparatory). Na rysunku 2.19 zostały zaprezentowane funkcje przenoszenia dla stopnia 1-bitowego, 1,5-bitowego oraz 2-bitowego. Zauważyć można, iż

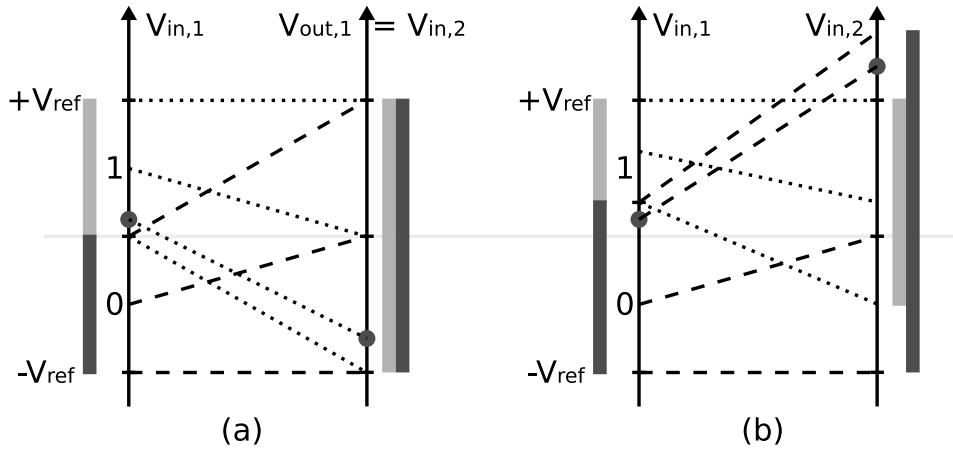


Rysunek 2.19. Funkcje przenoszenia dla stopnia (a) 1-bitowego (b) 1,5-bitowego oraz (c) 2-bitowego

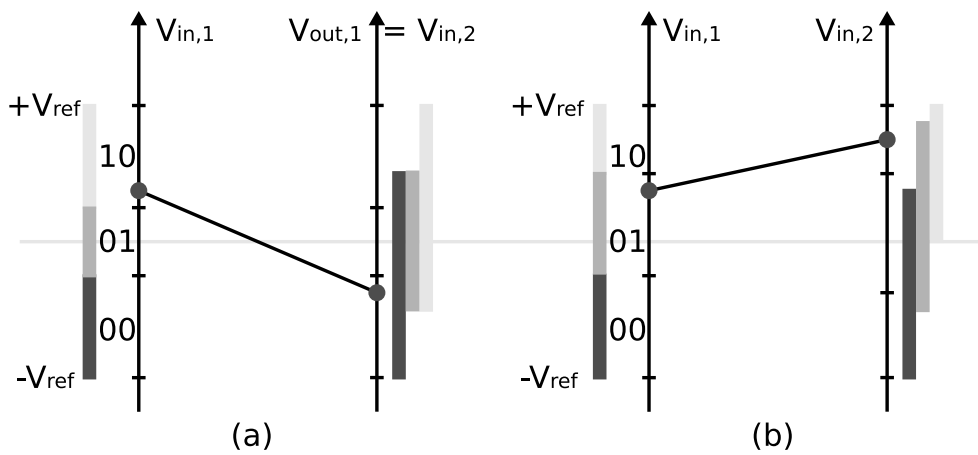
dodanie redundancji powoduje zwiększenie liczby poziomów kwantyzacji z dwóch do trzech, przy zachowaniu identycznego wzmacniania. Progi poszczególnych poziomów są przesunięte o $V_{ref}/4$ względem zera. Ważnym aspektem takiego stopnia jest to, że wyjściowe napięcie pozostaje pomiędzy $\pm V_{ref}/2$ dla szerokiego zakresu napięć wejściowych. Powyższe stwierdzenie nabiera znaczenia gdy do rozważań dołączymy nieidealności komparatorów (offset na ich wejściu). Funkcje przenoszenia dla stopnia jednobitowego zostały zaprezentowane na rysunku 2.20. Poziomymi kreskami zaznaczono poziomy napięć na wejściu i wyjściu stopnia odpowiadające kolejnym wartościom wyjść cyfrowych danego stopnia. Na rysunku 2.20b pokazano jakie są konsekwencje offsetu komparatora w pierwszym stopniu. Offset taki powoduje nasycenie wejścia kolejnego stopnia prowadząc tym samym do bezpowrotnej utraty informacji. Analogiczny rysunek wykonano dla stopnia 1,5-bitowego (rys. 2.21). Taką funkcję przenoszenia można opisać analitycznie jako:

$$V_{out,1} = a V_{in,1} - V_R, V_R = \begin{cases} -V_{ref} & V_{in,1} \leq -1/4V_{ref} \\ 0 & -1/4V_{ref} < V_{in,1} < 1/4V_{ref} \\ V_{ref} & V_{in,1} \geq 1/4V_{ref} \end{cases} .$$

Zauważyć można, iż napięcie na wyjściu nawet w przypadku offsetu nie nasycy się, co oznacza że informacja nie jest tracona. Informacja ta może zostać odzyskana w procesie omawianej korekcji cyfrowej. Rozwiązanie takie zapewnia nam poprawne funkcjonowanie układu w przypadku gdy próg komparatora nie przesunie się o więcej



Rysunek 2.20. Wizualizacja funkcji przenoszenia dla stopnia 1-bitowego (a) w przypadku idealnym (b) przy offsecie komparatorów (kropka obrazuje zachowanie stopnia dla przykładowego napięcia na wejściu)



Rysunek 2.21. Wizualizacja funkcji przenoszenia dla stopnia 1,5-bitowego w (a) w przypadku idealnym (b) przy offsecie komparatorów (kropka obrazuje zachowanie stopnia dla przykładowego napięcia na wejściu)

niż $\pm V_{ref}/4$. W ogólnym przypadku można zapisać, iż liczba progów przełączania (komparatorów) wynosi [Sum02]:

$$Q_i = 2^{B_i+r_i} - r + 1 \quad (2.39)$$

Wartości poszczególnych progów dla danego napięcia referencyjnego (napięcia pełnej skali) V_{ref} :

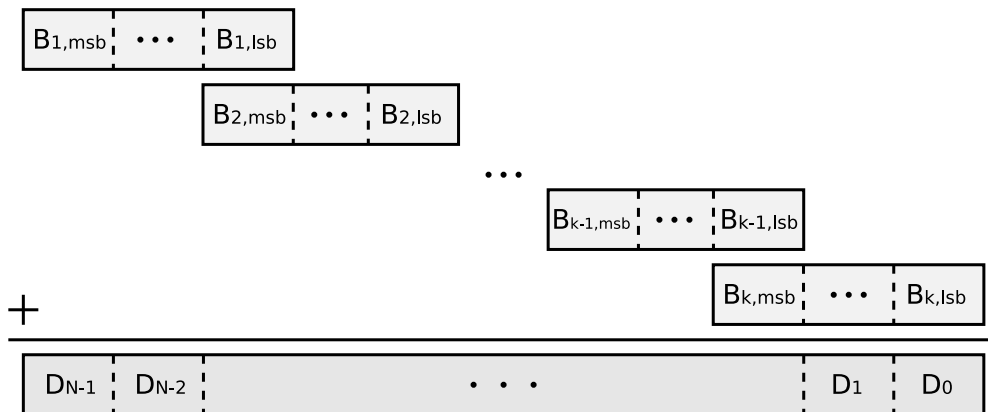
$$\pm V_{th,n} = \pm \left(\frac{n}{2^{B_i}} + \frac{r}{2^{B_i+r_i}} \right) V_{ref}, n \in [0, 2^{B_i+r-1} - r - 1] \quad (2.40)$$

Dla poprawnego funkcjonowania przetwornika istotne jest, by offset napięcia progowego żadnego z komparatorów nie wynosił więcej niż:

$$\pm V_{th,of} = \pm \frac{r}{2^{B_i+r}} V_{ref} \quad (2.41)$$

Warto zauważyć, że dokładność komparatorów zależy tylko od rozdzielczości stopnia (napięcia referencyjnego oraz liczby progów) natomiast wogóle nie zależy od całkowitej rozdzielczości przetwornika.

Korekcja cyfrowa polega na dodaniu odpowiednio opóźnionych wyjść z poszczególnych stopni z jednobitowym (dla $r = 1$) przekrywaniem: najbardziej znaczący bit (MSB) stopnia i jest dodawany do najmniej znaczącego bitu (LSB) stopnia poprzedniego $i - 1$. Schematycznie proces ten został zaprezentowany na rysunku 2.22. Postępowanie według tego algorytmu prowadzi do wniosku, iż najmniej znaczący bit



Rysunek 2.22. Algorytm korekcji cyfrowej.

z ostatniego stopnia nie podlega korekcji, co sugeruje, że ostatni stopień powinien być stopniem bez redundancji ($r_k = 0$), ewentualnie w słowie wyjściowym ostatni bit D_0 może zostać pominięty. Wracając do rysunku 2.21 można zauważyć, iż w obu przypadkach wynik po korekcji wynosi 10 binarnie pomijając ostatni nieskorygowany bit.

Reasumując rozważania dotyczące omawianej architektury potokowej należy stwierdzić, iż niewielkim kosztem można znacząco zredukować wymagania stawiane jednemu z krytycznych bloków każdego przetwornika ADC - komparatora. Architektura 1,5-bitowa relaksuje wymagania stawiane offsetowi wzmacniacza, nie zmieniając jednocześnie wymagań co do pasma i wzmocnienia w otwartej pętli. Pomimo konieczności dodania bloku korekcji cyfrowej oraz kilku dodatkowych kluczy powierzchnia

zajmowana przez układ zazwyczaj się zmniejsza. Dzieje się tak dlatego, iż można zastosować prostsze i mniej dokładne, a co za tym idzie mniejsze komparatory. Dodatkowo korekcja cyfrowa powoduje, iż rozdzielczość całkowita przetwornika nie zależy od rozdzielczości poszczególnych stopni co prowadzi do oszczędności mocy i powierzchni szczególnie dla przetworników ADC o wysokiej rozdzielczości.

Rozdział 3

Projekt i pomiary przedwzmacniacza

3.1. Architektura

Z rozważań przeprowadzonych w poprzednim rozdziale jasno wynikają wymagania stawiane układom elektroniki front-end, a w szczególności układowi przedwzmacniacza. Ma on być stosunkowo szybki, posiadać duże wzmocnienie, być możliwie niskoszumny, rozpraszać mało mocy oraz zajmować mało powierzchni. Dodatkowymi wymaganiami stawianymi przed wszystkimi układami analogowymi jest niewrażliwość na sygnały wspólne (duży współczynnik Common Mode Rejection Ratio – CMRR), wahania napięcia zasilania (duży współczynnik Power Supply Rejection Ratio – PSRR) oraz niewrażliwość na parametry procesu technologicznego.

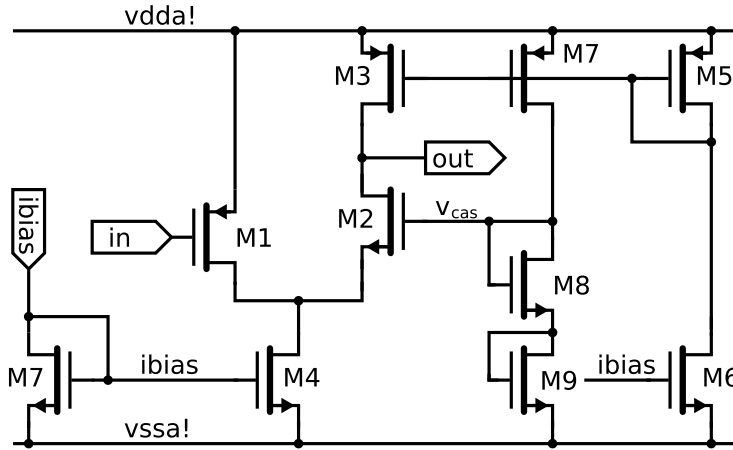
Stosowanie układów różnicowych zdecydowanie poprawia parametry CMRR oraz PSRR, lecz niestety okupione jest jedną poważną wadą, układy takie mają dwa tranzystory wejściowe. Fakt ten z punktu widzenia optymalizacji szumowej jest bardzo niekorzystny, gdyż drugi tranzystor zwiększa szum wyjściowy około $\sqrt{2}$ raza [Gaj07].

W projektowanym układzie wymagania szumowe przesądziły o wyborze architektury niesymetrycznej (tzn. z pojedynczym wejściem). Aby zapewnić możliwe duże pasmo wzmocnienia wybrano architekturę zawiniętej kaskody [GHLM01, Raz00] z obciążeniem w postaci lustra prądowego. Konfiguracja zawiniętej kaskody, dzięki rozseparowaniu prądów w tranzystorze wzmacniającym i w gałęzi obciążenia posiada następujące własności:

- redukuje ekwiwalentną pojemność wejściową (wynikającą z efektu Millera na pojemności C_{gd1} tranzystora M1),
- pozwala uzyskać większe wzmocnienie niż konfiguracja klasycznej kaskody ze względu na możliwość zmniejszenia prądu w gałęzi wyjściowej, a co tym idzie zwiększenia rezystancji wyjściowych,
- pozwala uzyskać większe pasmo wzmocnienia.

Schemat wzmacniacza został zaprezentowany na rysunku 3.1.

W przedstawionym układzie wyjściowy tranzystor M1 jest tranzystorem wzmacniającym. Ze względu na szумы nadmiarowe $1/f$ zdecydowano się na wykonanie stopnia wejściowego na tranzystorze PMOS [Fiu07]. Tranzystor M2 jest tranzystorem kaskodowym, zapewniającym minimalizację efektu Millera. Tranzystor M3 sta-



Rysunek 3.1. Schemat przedwzmacniacza

nowi obciążenie aktywne w układzie lustra prądowego. W zawiniętej kaskodzie tranzystor M4 pracuje jako źródło prądu stałego zapewniającego odpowiednią polaryzację tranzystorom M1 i M2. W idealnym przypadku jest on całkowicie do pominięcia w schemacie małosygnałowym. Pozostałe tranzystory tworzą obwody polaryzujące (dostarczają odpowiednich prądów i napięć) i są wspólne dla wszystkich kanałów.

3.2. Model małosygnałowy

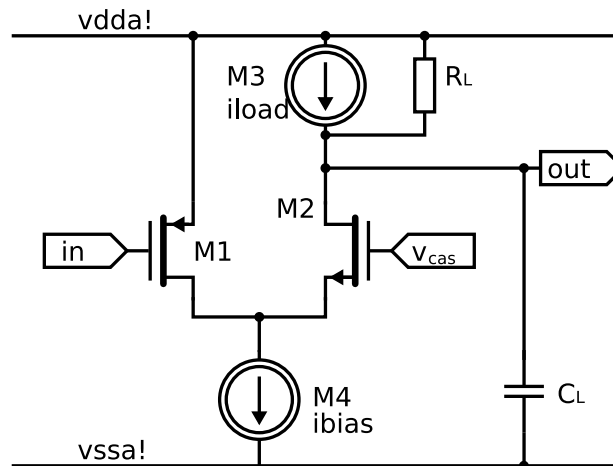
Uproszczonego modelu kaskody został przedstawiony na rysunku 3.2. Tranzystory M3 oraz M4 zostały zastąpione idealnymi źródłami prądowymi, a w wypadku źródła M3 dodano jeszcze rezystancję wyjściową ($R_L = r_{ds3}$), gdyż jest ona kluczowa dla otrzymywanych formuł. Dodatkowo dodano pojemność symbolizującą pojemność obciążenia C_L . Funkcja przenoszenia takiego układu dla niskich częstotliwości, zanedbując efekt podłoża, może zostać zapisana jako [PRAB96]:

$$\frac{V_{out}}{V_{in}} = A_v \simeq -\frac{g_{m1}R_L(1 + g_{m2}r_{ds2})}{1 + g_{m2}r_{ds2} + \left(\frac{R_L + r_{ds2}}{r_{ds1}}\right)} \quad (3.1)$$

Dla typowych wartości elementów wartość wzmocnienia determinowana jest przez transkonduktancję g_{m1} tranzystora wejściowego M1 i rezystancję obciążenia R_L .

W typowych aplikacjach biegun dominujący pojawia się na wyjściu układu (dren tranzystora M2), a stowarzyszona z nim stała czasowa wynosi [PRAB96]:

$$\tau_1 \simeq \left(\frac{R_L(g_{m2}r_{ds1}r_{ds2} + r_{ds1} + r_{ds2})}{R_L + g_{m2}r_{ds1}r_{ds2} + r_{ds1} + r_{ds2}} \right) \cdot (C_{gd2} + C_{db2} + C_L). \quad (3.2)$$



Rysunek 3.2. Uproszczony schemat przedwzmacniacza

z kolei biegun niedominujący pojawia się na drenie tranzystora M1. Jego stała czasowa to [PRAB96]:

$$\tau_2 \simeq \frac{(C_{gd1} + C_{db1} + C_{gs2} + C_{sb2})(R_L + r_{ds2})r_{ds1}}{(R_L + g_{m2}r_{ds1}r_{ds2} + r_{ds1} + r_{ds2})} \quad (3.3)$$

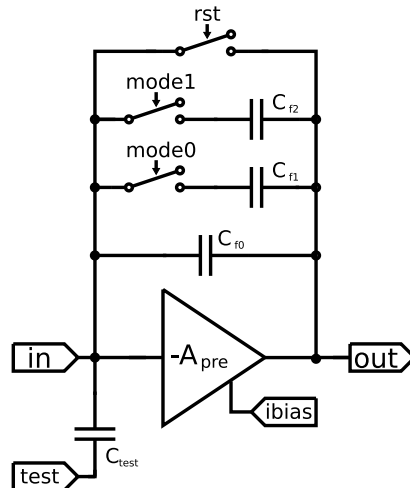
W rzeczywistości powyższe formuły nie są wystarczająco dokładne, gdyż należałoby w nich jeszcze uwzględnić wpływ rezystancji i pojemności pochodzących od tranzystora M4, które zostały tu pominięte dla uproszczenia.

3.3. Realizacja

Projekt układów elektronicznych wykonany został przy użyciu pakietu narzędzi projektowych Cadance [Cad08]. Większość symulacji przeprowadzono z wykorzystaniem symulatorów hspiceS oraz Spectre. Używane oprogramowanie opiera się na modelu tranzystora polowego BSIM3.

Pełen kanał odczytowy został zrealizowany na bazie omówionego wyżej wzmacniacza. Aby uzyskać możliwie duże wzmocnienie, w sprzężeniu zwrotnym umieszczono pojemność C_{f0} o wartości 250 fF. Pojemność taka w rezultacie (przy odpowiednio dużym wzmocnieniu wzmacniacza) owocuje wzmocnieniem ładunkowym na poziomie $4 \text{ mV}/fC$.

Aby być w stanie mierzyć duże ładunki, zdecydowano się dołożyć dużą pojemność w sprzężeniu zwrotnym, która może być dołączana dynamicznie za pomocą klucza (rys. 3.3). Ponieważ projektowany układ na charakter prototypowy zdecydowano się na umieszczenie dwóch pojemności o wartościach $C_{f1} = 5 \text{ pF}$ oraz $C_{f2} = 10 \text{ pF}$. Pojemności te mogą być dołączane niezależnie. Zestaw możliwych konfiguracji sprzężenia zwrotnego został zaprezentowany w tabeli 3.1.



Rysunek 3.3. Schemat kanału odczytowego zrealizowanego w oparciu o resetowany przedwzmacniacz

Tablica 3.1. Konfiguracje sprzężenia zwrotnego projektowanego układu

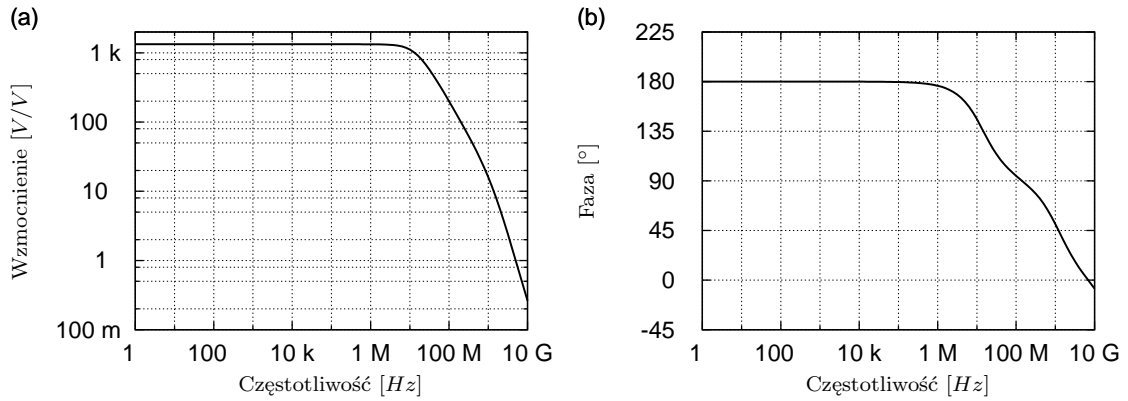
tryb	mode0	mode1	C_f [pF]	wzm. ład. [$\mu\text{V}/\text{fC}$]
0	0	0	0,25	4000
1	0	1	5,25	190,5
2	1	0	10,25	97,5
3	1	1	15,25	65,5

Dodatkowo w celu zapewnienia testowalności układu, na wejściu dodano kondensator testowy C_{test} o pojemności wynoszącej 0,5 pF. Kondensator ten pełni funkcję adaptora ładunkowego [Kor05].

3.4. Wyniki symulacji

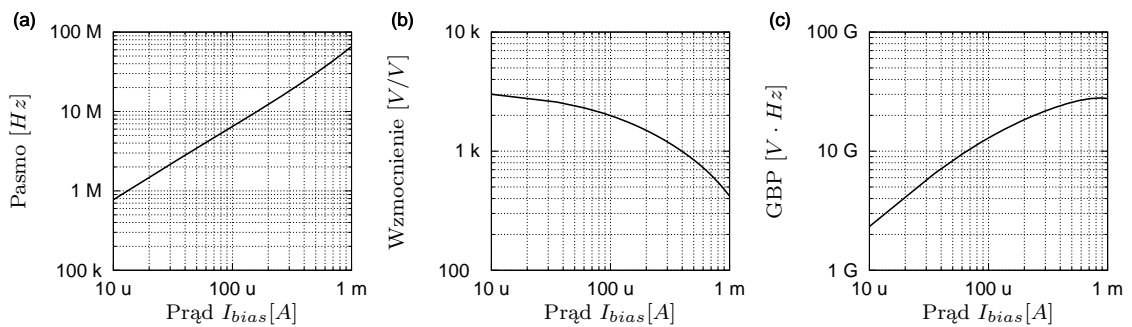
3.4.1. Charakterystyki częstotliwościowe

Aby zweryfikować poprawność projektu wykonano szereg symulacji układów samego wzmacniacza jak również kompletnego przedwzmacniacza. Po ustaleniu punktów pracy poszczególnych tranzystorów przystąpiono do symulacji w dziedzinie częstotliwości AC. Zależności wzmocnienia oraz fazy od częstotliwości dla wzmacniacza pracującego w otwartej pętli zostały przedstawione na rysunku 3.4. Wzmocnienie niskoczęstotliwościowe wynosi 1300, biegun dominujący ma częstotliwość 14,1 MHz. Układ został tak zaprojektowany, by w razie potrzeby móc zmienić punkty pracy poszczególnych tranzystorów, a tym samym niektóre z jego parametrów małosygna-



Rysunek 3.4. Wyniki symulacji (a) wzmocnienia oraz (b) fazy w wzmacniaczu w funkcji częstotliwości

lowych. Możliwa jest regulacja prądu sterującego i_{bias} w bardzo szerokim zakresie. Zależność wzmocnienia i pasma przenoszenia w funkcji tego prądu zaprezentowano

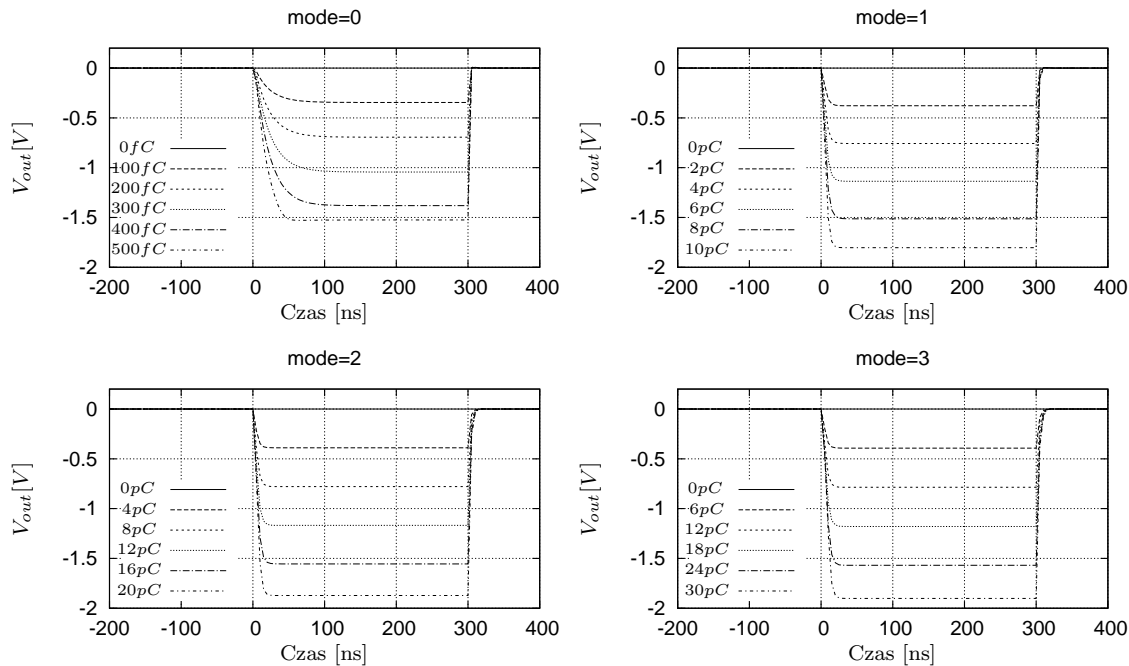


Rysunek 3.5. Symulacje wpływu referencyjnego I_{bias} na parametry układu wzmacniacza: (a) pasmo (b) wzmocnienie niskoczęstotliwościowe (c) iloczyn pasma i wzmocnienia (GBP)

na rysunku 3.5. Wartością domyślną jest $250 \mu\text{A}$, która odpowiada prądowi $2,5 \text{ mA}$ w tranzystorze wejściowym (M1). Możliwość regulacji prądu przydaje się również w sytuacjach gdy rozrzuty technologiczne powodują zmiany parametrów znamionowych.

3.4.2. Odpowiedzi czasowe

W kolejnym kroku przystąpiono do symulacji całego układu przedwzmacniacza. Na rysunku 3.6 zostały przedstawione przykładowe odpowiedzi układu, pracującego przy różnych wzmocnieniach, na różne wymuszenia ładunkowe. Zaobserwować można, iż układ odpowiada w czasie krótszym niż planowane 300 ns we wszystkich przypadkach. Po 300 ns następuje reset i powrót układu do stanu gotowości

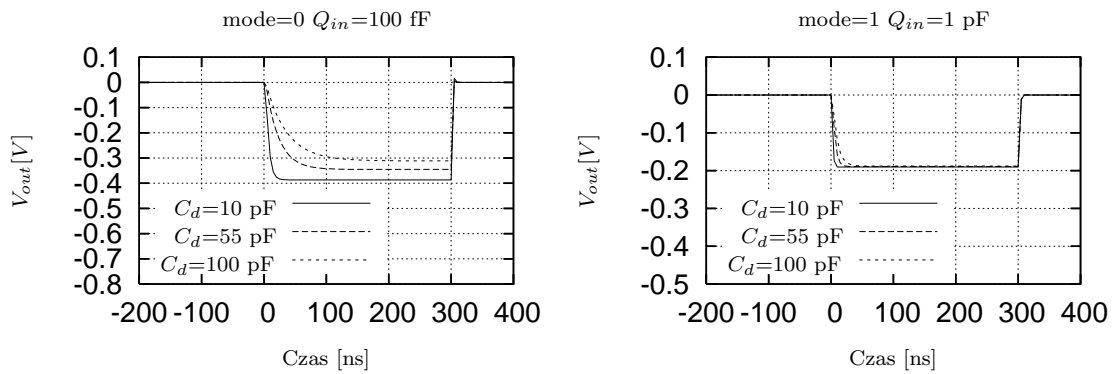


Rysunek 3.6. Symulowana odpowiedź czasowa układu przedwzmacniacza dla różnych ładunków wejściowych ($C_d = 55$ pF)

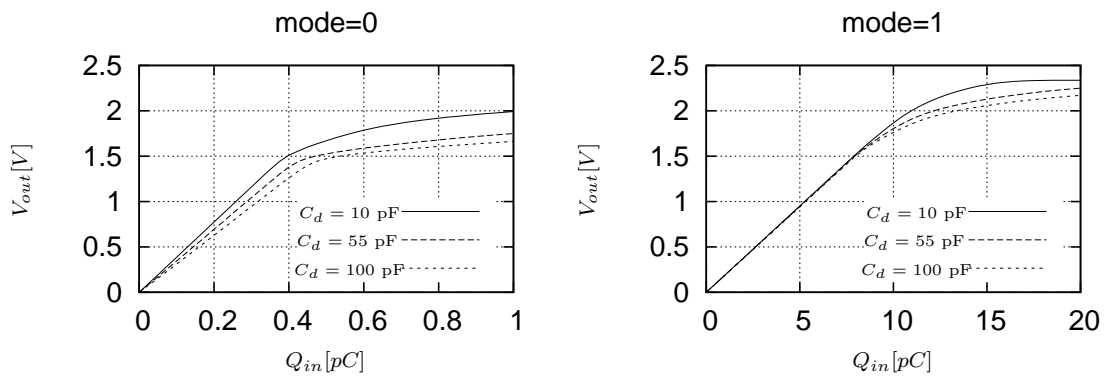
na następne impulsy wejściowe. Wpływ pojemności wejściowej na kształt impulsu został zaprezentowany na rysunku 3.7. Zauważalny jest wyraźny spadek wzmocnienia ładunkowego dla największego wzmocnienia (mode=0), co zostało oszacowane w rozdziale 2.2. Dla dużych wartości pojemności sprzężenia zwrotnego (mode=1,2,3) ubytek wzmocnienia powodowany pojemnością C_d jest zaniedbywalny. Jak wykazały precyzyjne symulacje, dla każdej kombinacji wzmocnienia oraz pojemności wejściowej, układ jest w stanie ustalić wartość napięcia wyjściowego z dokładnością 0,1 % w czasie krótszym niż 300 ns, a następnie rozładować tą pojemność w mniej niż 40 ns.

3.4.3. Wzmocnienie ładunkowe

Wzmocnienia ładunkowe dla kilku pojemności C_d zostały zaprezentowane na rysunku 3.8. Zauważyć można, iż dla wszystkich trybów układ pracuje liniowo do napięć wyjściowych na poziomie 1,5 V, po czym charakterystyka zmienia nachylenie. Układ nasycy się dla napięć wyjściowych wynoszących około 2V. Otrzymane wartości wzmocnienia zgadzają się bardzo dobrze z przewidywaniami teoretycznymi (patrz tab. 3.1).



Rysunek 3.7. Symulowana odpowiedź czasowa układu przedwzmacniacza dla różnych ładunków pojemności wejściowych (a) $mode = 0$ $q_{in} = 100$ fF (b) $mode = 1$ $q_{in} = 1$ pF



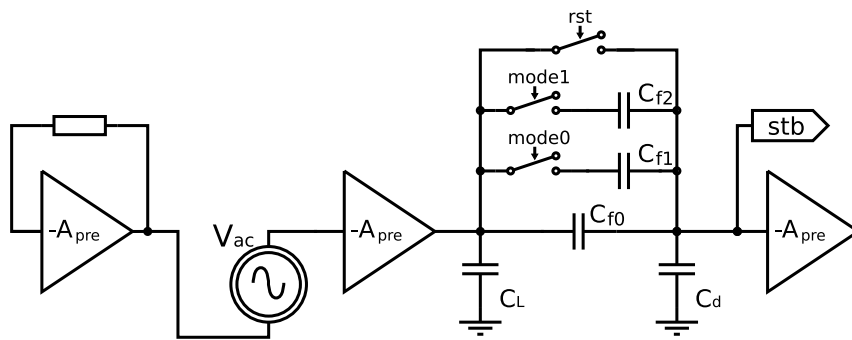
Rysunek 3.8. Symulowane wartości wzmocnienia ładunkowego dla zestawu pojemności C_d

3.4.4. Stabilność

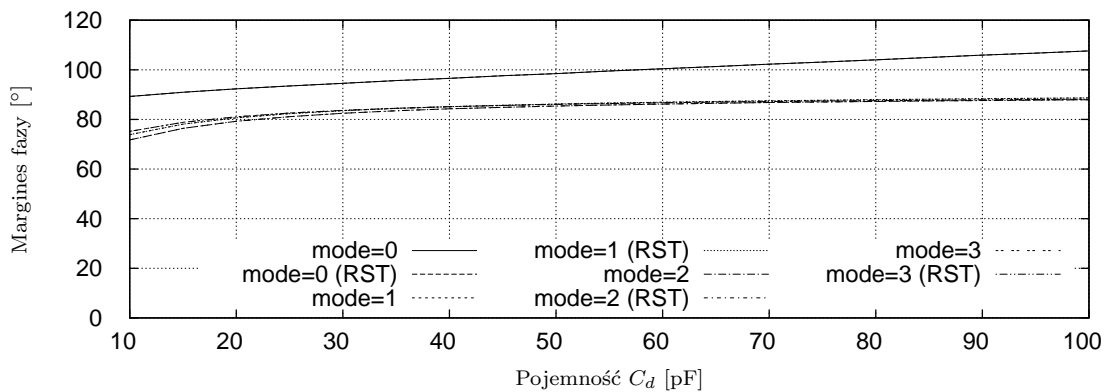
We wszystkich układach ze sprzężeniem zwrotnym kluczowym aspektem jest stabilność. Do oszacowania stabilności na drodze symulacyjnej wykorzystano metodę replik, wykorzystując schemat zaprezentowany na rysunku 3.9. Margines fazy dla różnych konfiguracji sprzężenia zwrotnego w funkcji pojemności C_d zaprezentowano na rysunku 3.10. Z powyższych symulacji wynika, iż dla wszystkich trybów pracy, jak również dla interesującego zakresu pojemności C_d , margines fazy wynosi ponad 70° . Tak duży margines fazy gwarantuje stabilność pracy układu.

3.4.5. Szumy

Kolejnym krokiem jaki wykonano było przeprowadzenie symulacji szumowych. Przykładowe widma szumów na wyjściu przedwzmacniacza pracującego w trybie wysokiego wzmocnienia, dla różnych pojemności C_d , zaprezentowano na rysunku 3.11.

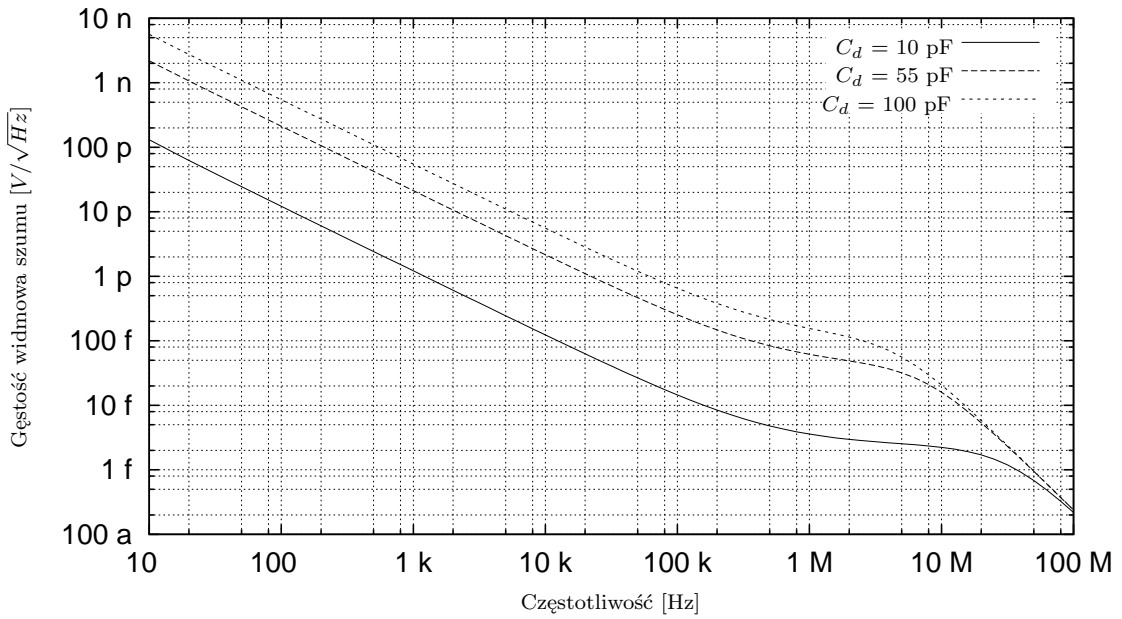


Rysunek 3.9. Schemat wykorzystywany do badania stabilności

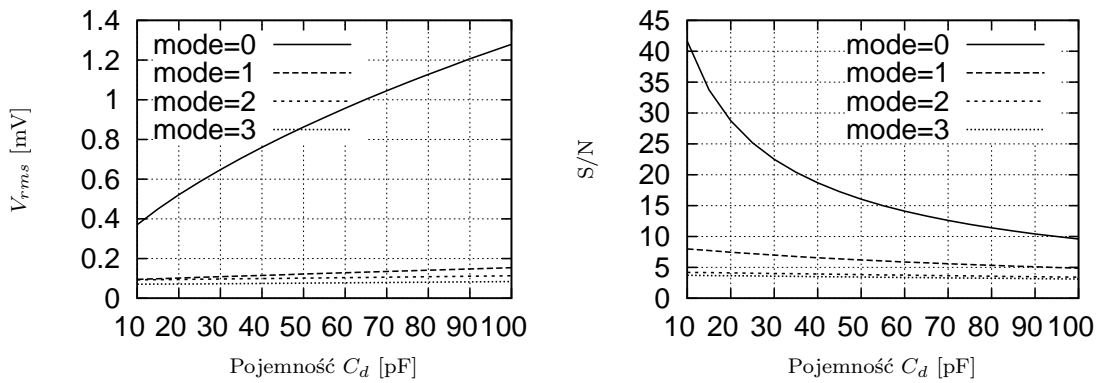
Rysunek 3.10. Symulowany margines fazy układu przedwzmacniacza w funkcji pojemności C_d dla różnych trybów pracy

Zgodnie z oczekiwaniami gęstości widmowe szumów na wyjściu są większe w przypadku dużych wartości pojemności C_d .

Całkując gęstości widmowe szumów w interesującym zakresie częstotliwości można otrzymać wartość średniokwadratową szumów na wyjściu. Na rysunku 3.12a zaprezentowano wartość szumów na wyjściu przedwzmacniacza w funkcji pojemności C_d dla różnych trybów pracy. Wyniki te otrzymano przyjmując dolną granicę całkowania gęstości szumowych na poziomie 1 Hz. Korzystając z poprzednio otrzymanych wartości wzmocnień ładunkowych możliwe było wyznaczenie stosunku sygnału do szumu. Jako sygnał referencyjny przyjęto średnią wartość ładunku generowaną przez cząstkę minimalnie jonizującą w rozważanym sensorze, czyli 4 fC. Wartości S/N w funkcji pojemności C_d dla różnych trybów pracy zostały zaprezentowane na rysunku 3.12b. Z otrzymanych zależności wynika, iż w trybie wysokiego wzmocnienia dla całego interesującego zakresu pojemności C_d stosunek sygnału do szumu wynosi



Rysunek 3.11. Przykładowe symulowane widma szumów na wyjściu przedwzmacniacza pracującego w trybie wysokiego wzmocnienia ($mode = 0$) dla różnych pojemności C_d



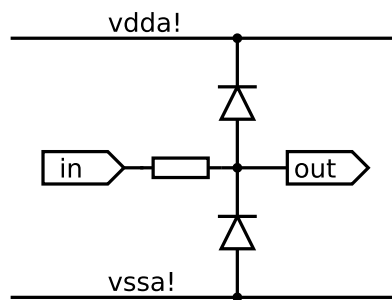
Rysunek 3.12. (a) wartość szumów na wyjściu przedwzmacniacza w funkcji pojemności C_d dla różnych trybów pracy (b) stosunek sygnału ($4 fC$) do szumu w funkcji pojemności C_d dla różnych trybów pracy

ponad 10. W trybie dużego wzmocnienia szumy wypadają również poniżej poziomu kwantyzacji najmniej znaczącego bitu ADC.

3.5. Plan masek

Ostatnim etapem podczas projektowania układu scalonego jest wykonanie planu masek technologicznych. Podczas tego etapu należy pogodzić szereg wymagań stawianych układowi z ograniczeniami wynikającymi z procesu technologicznego [HH05]. Dla prototypowanego układu zdecydowano się na użycie technologii CMOS $0,35\ \mu\text{m}$ AMS [Mic07]. Dostępne w użytej technologii elementy to: tranzystory polowe komplementarne NMOS i PMOS o minimalnej długości bramki $350\ \text{nm}$, tranzystory bipolarne, rezystory, kondensatory i cztery warstwy połączeń metalicznych. W stosowanej technologii tranzystory PMOS wykonywane są w studni n^+ (domieszkowanej jonami ujemnymi), możliwe jest lokalne polaryzowanie podłoża tych tranzystorów. Napięcie zasilania w technologii C35B4 wynosi $3,3\ \text{V}$.

Projekt kanału odczytowego wykonano jako strukturę o szerokości (ang. pitch) $100\ \mu\text{m}$. Wejście kanału stanowi pole kontaktowe z dodatkowym układem zabezpieczającym zaprezentowanym na rysunku 3.13.

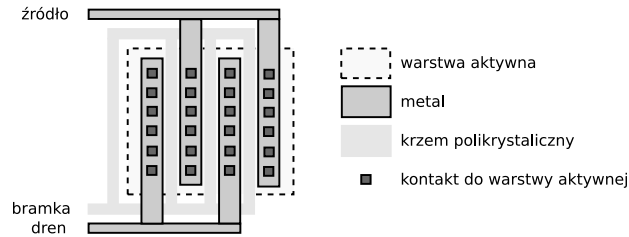


Rysunek 3.13. Schemat struktury zabezpieczającej wejście układu

Takie zabezpieczenie wymuszone jest koniecznością zapewnienia odporności stopnia wejściowego na wyładowania elektrostatyczne mogące powstać podczas montażu (ang. Electro Static Discharge – ESD). W pierwszym przybliżeniu układ ten nie ma wpływu na funkcjonalność dalszych stopni kanału, jako iż obie diody powinny pozostawać spolaryzowane zaporowo. Ogranicznik diodowy powinien zacząć przewodzić dopiero po pojawieniu się sygnału wyższego niż potencjał zasilania lub niższego niż potencjał masy. W praktyce jednak z każdą diodą stowarzyszony jest pewien prąd upływu (wynikający z termicznej generacji nośników), który będzie źródłem szumów równoległych do szumów pochodzących z sensora promieniowania. Wymiary diod są więc kompromisem pomiędzy wielkością ładunku jakie są w stanie zaabsorbować diody, a prądami upływu jakie wprowadzają na wejście układu.

W celu zminimalizowania rezystancji rozproszonej bramki, tranzystor wejściowy został zrealizowany jako struktura wielopalczasta. Aby odseparować ten stopień od pozostałej części układu dodatkowo dodano pierścienie ochronne wokół całej struktury.

Technologia wybrana do wykonania prototypów oferuje kilka rodzajów kondensatorów: POLI–POLI, MET–OX–MET, MOS [Mic07]. Kondensatory sprężenia zwrot-



Rysunek 3.14. Schemat tranzystora wielopalczastego

nego, ze względu na najlepszą precyzję wykonania oraz niezmienność pojemności, zdecydowano się wykonać jako struktury POLI–POLI. Niestety kondensatory wykonane jako takie struktury charakteryzują się pojemnością $0,86 \text{ fF}$ na $1 \mu\text{m}^2$. Dlatego też powierzchnia kanału zdominowana jest przez duże pojemności C_{f1} oraz C_{f2} .

Całość kanału kończy się padem wyjściowym. Ponieważ kanały projektowane przez autora są krótsze niż te z klasycznym ciągłym kształtowaniem, zdecydowano się na umieszczenie za nimi struktur generujących napięcia oraz prądy referencyjne (patrz rysunek 3.15).

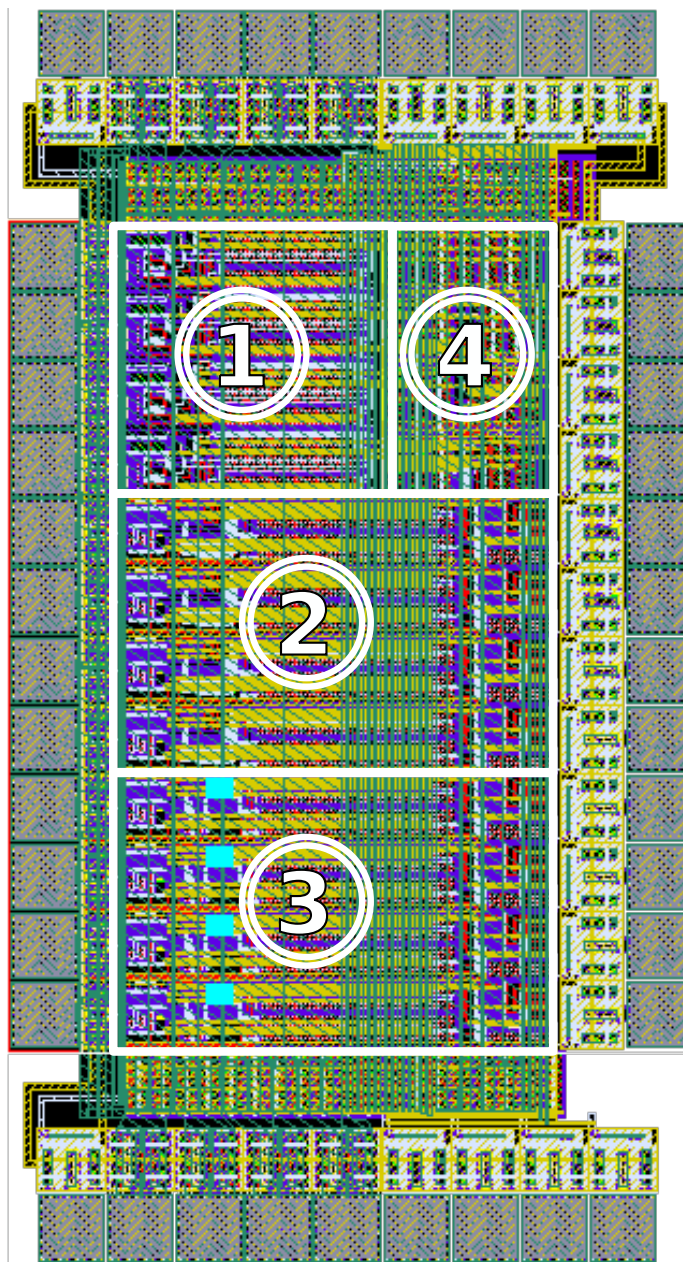
Do rozprowadzenia zasilania wykorzystano czwartą warstwę metalu. Szyny zasilające są ułożone poprzecznie w stosunku do orientacji kanału. Aby zminimalizować rezystancje tych szyn zdecydowano się na umieszczenie padów zasilania z obu stron. Wszystkie wolne przestrzenie wykorzystano do zrobienia pojemności filtrujących poszczególne napięcia polaryzujące. Wykorzystano do tego celu struktury tranzystorów MOS, co pozwoliło przeszło czterokrotnie zwiększyć pojemność w stosunku do tej wykonanej jako struktura kondensatora pomiędzy warstwami krzemu polikrystalicznego [Mic07].

3.6. Stanowisko pomiarowe

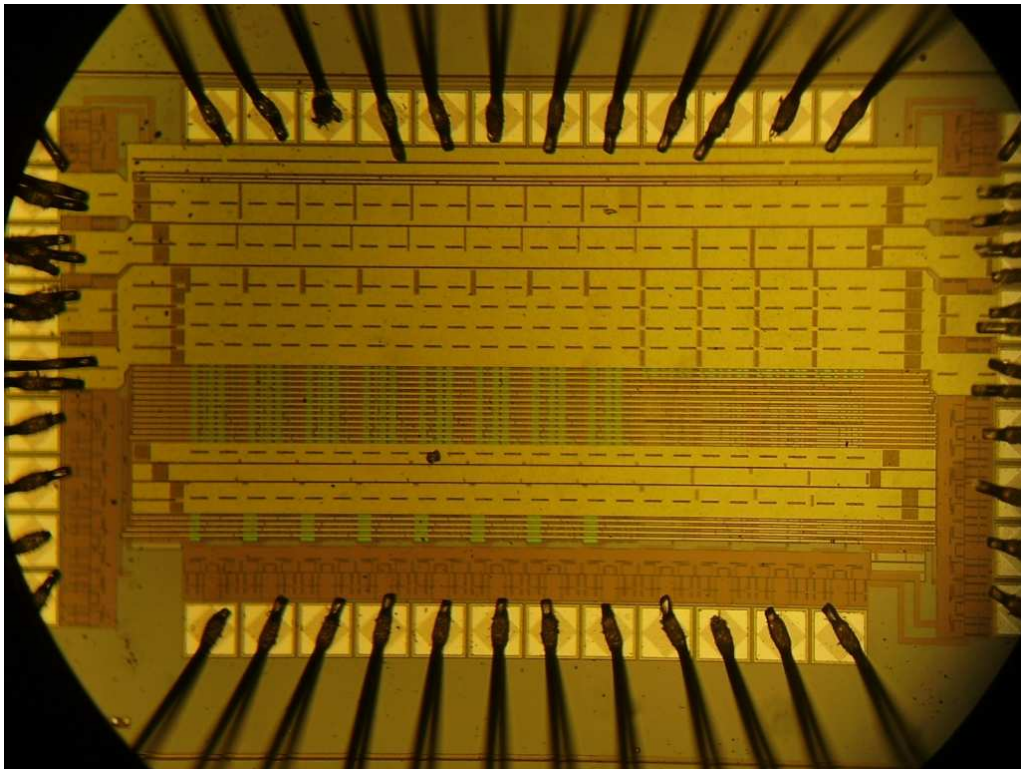
Prototypowe układy elektroniki front-end wykonane w technologii AMS $0,35 \mu\text{m}$ zostały zrealizowane w okresie dwóch miesięcy od wysłania projektu. Ten czas został wykorzystany na przygotowanie stanowiska pomiarowego. Po otrzymaniu prototypów kilka z układów ASIC zostało przybondowanych do płytek drukowanych. Fotografię układu wraz z bondami przedstawiono na rysunku 3.16.

Aby przetestować układ z kluczowanym sygnałem reset konieczne było przygotowanie dedykowanego stanowiska testowego, wraz z autorskim systemem akwizycji danych opisanym w rozdziale 5.2.

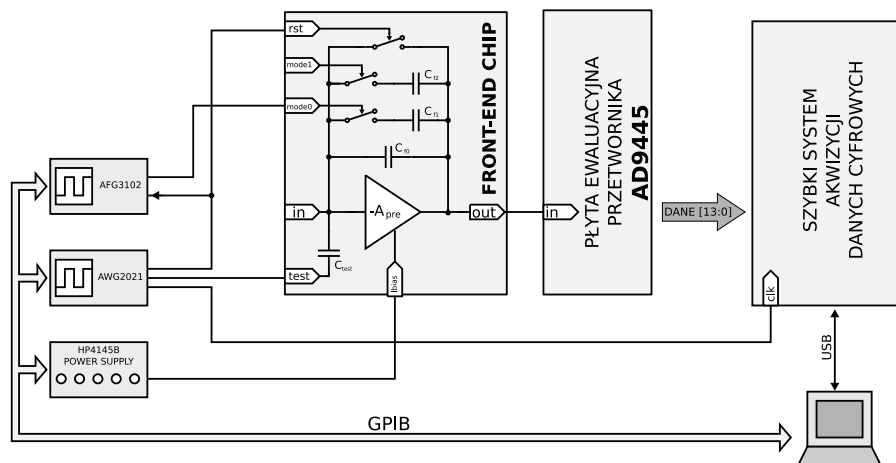
Procedura pomiarowa polegała na podawaniu na pojemności testowe, umieszczone na wejściu przedwzmacniacza wewnątrz układu (rys. 3.3) oraz na płycie drukowanej (rys. 5.6), sygnałów testowych i rejestrowaniu odpowiedzi za pomocą szybkiego, wysoko-rozdzielczego przetwornika ADC. Do generacji sygnałów wykorzystano programowalne generatory funkcyjne AWG2021 oraz AFG3102 firmy Tek-



Rysunek 3.15. Plan masek topologicznych zaprojektowanego układu scalonego z elektroniką front-end (1) Kanały z kluczowanym resetem (2) Układy z kształtowaniem ciągłym i sprzężeniem zwrotnym wykonanym w sposób aktywny (3) Układy z kształtowaniem ciągłym i sprzężeniem zwrotnym wykonanym w sposób pasywny (4) Układy generacji napięć i prądów polaryzujących (ang. biasing)

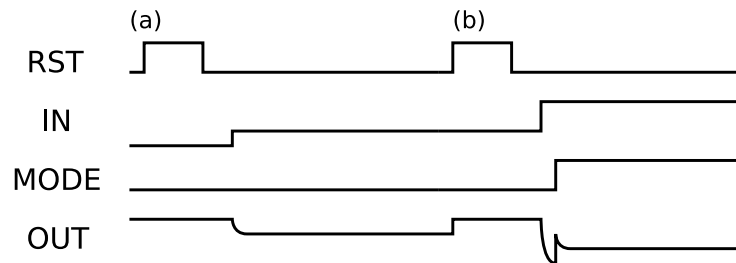


Rysunek 3.16. Widok z góry prototypowego układu elektroniki front-end.



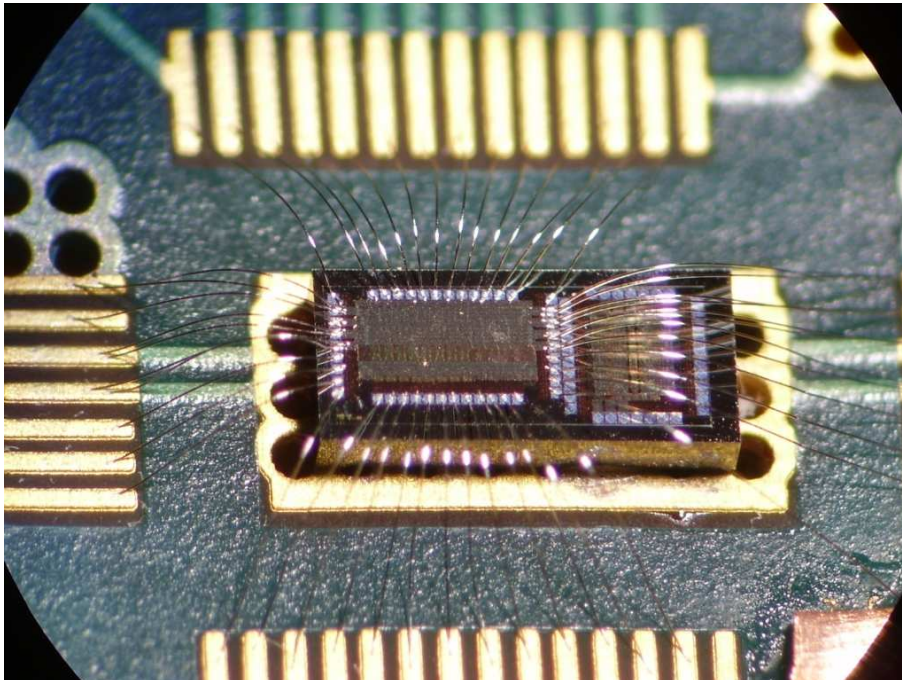
Rysunek 3.17. Stanowisko pomiarowe wykorzystywane do testowania przedwzmacniacza z kluczowanym resetem

tronix, o rozdzielczościach odpowiednio 12 bitów i 14 bitów. Generator AWG2021 posiada również 2 wyjścia cyfrowe (tzw. markery), które posłużyły do generacji sygnałów reset oraz zegara, wyznaczającego częstotliwość pracy ADC. Drugi z generatorów (AFG2102) pracował synchronicznie z generatorem pierwszym, w razie potrzeby generując sygnał sterujący przełączaniem wzmacnienia. Przykładowa se-



Rysunek 3.18. Przykładowe sekwencje sygnałów używane do testowania przedwzmacniacza z kluczowanym resetem (a) mały sygnał (b) duży sygnał – dynamiczna zmiana wzmacnienia

kwencja czasowa przebiegów sterujących zademonstrowana została na rysunku 3.18. System taki jest bardzo elastyczny, pozwala przy tym bardzo precyzyjnie kontrolować całą sekwencję przebiegów czasowych.

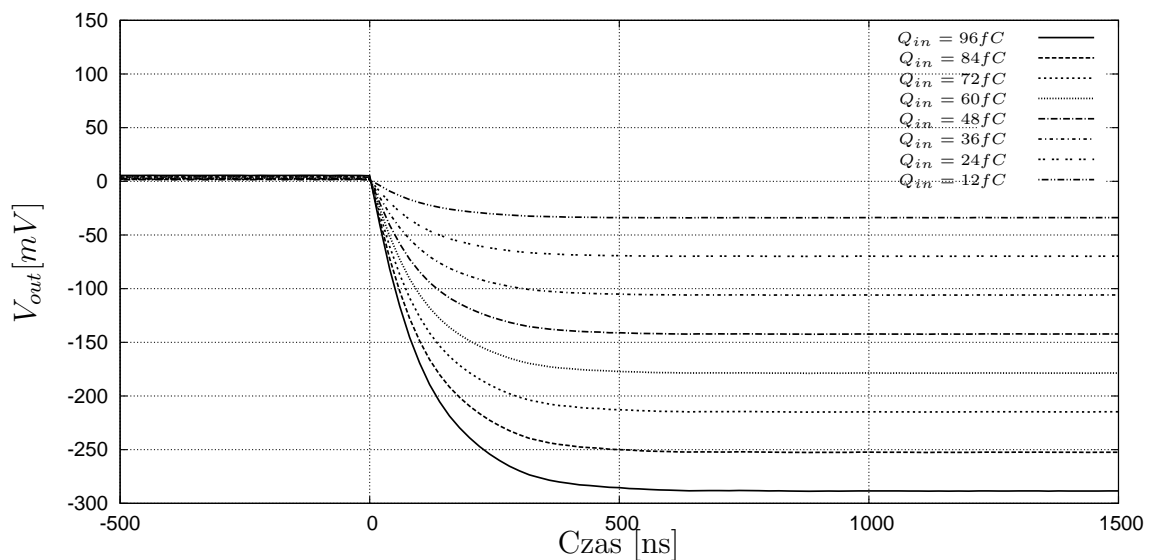


Rysunek 3.19. Widok perspektywiczny prototypowego układu elektroniki front-end.

Fotografię układu osadzonego na płytce drukowanej wraz z połączeniami do niej (ang. bonds) zaprezentowano na rysunku 3.19.

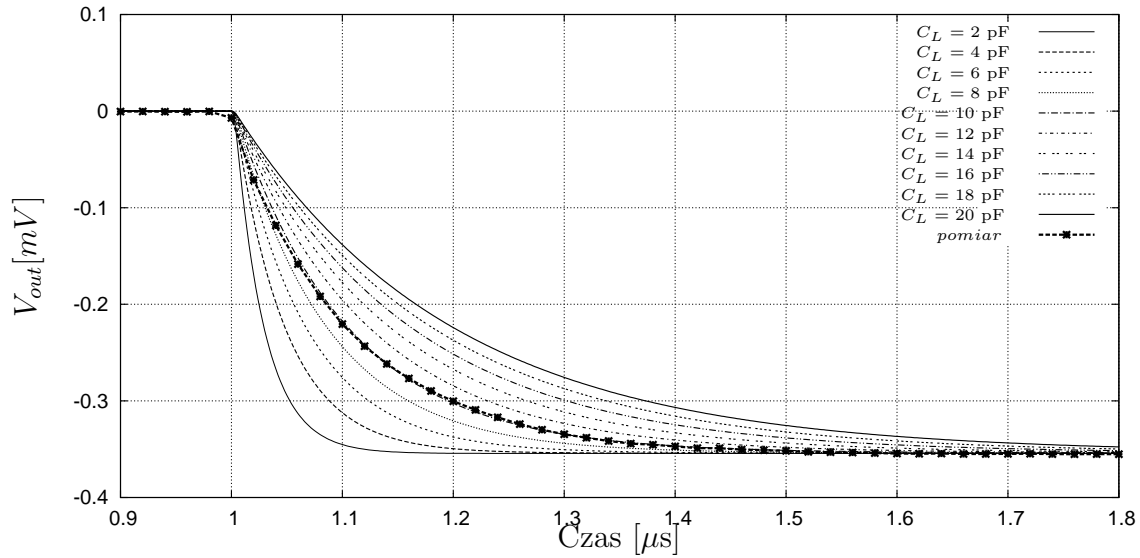
3.7. Wyniki pomiarów

Pomiary przedwzmacniacza rozpoczęto od pomiarów statycznych. Dla znamionowego prądu referencyjnego ($i_{bias}=250 \mu A$) poziom stały napięcia wyjściowego przedwzmacniacza wynosi około 2,4 V, a moc pobierana przez jeden kanał to 9,1 mW, co pozostaje w bardzo dobrej zgodności z symulacjami. W kolejnym kroku przystąpiono do pomiarów odpowiedzi układu w dziedzinie czasu na wstrzykiwany ładunek (patrz rozdział 3.6). Przykładowe odpowiedzi układu pracującego w trybie 0 (o wysokim wzmocnieniu), na różne ładunki wejściowe, zaprezentowano na rysunku 3.20.



Rysunek 3.20. Odpowiedzi układu na różne ładunki wejściowe ($C_d=44$ pF, mode=0). Prezentowane rezultaty nie zawierają składowej stałej

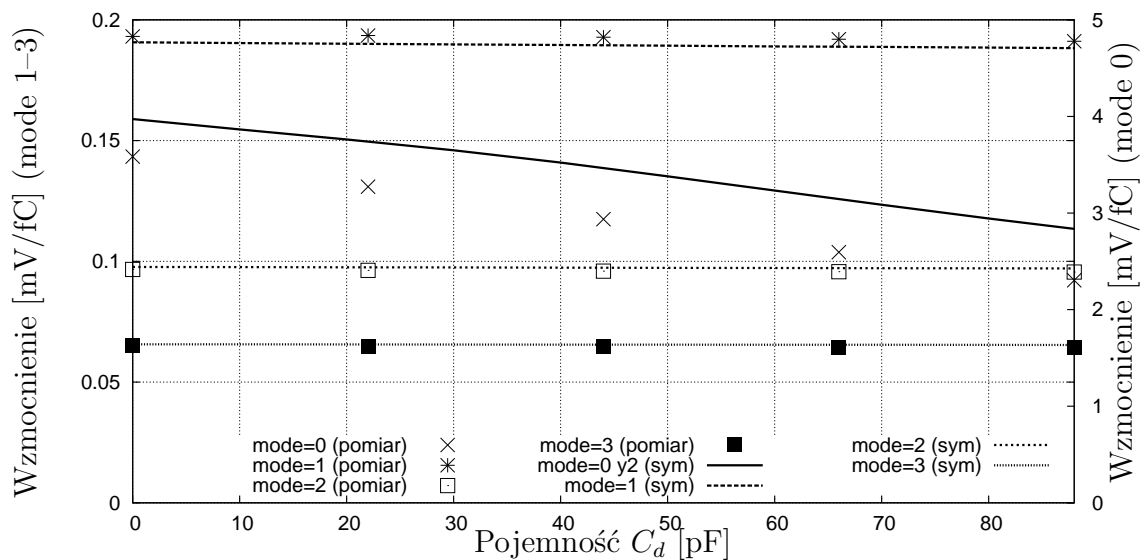
Jakościowo pomiary są zgodne z symulacjami. Można zauważyć, że stałe czasowe (czasy narastania) impulsów są znacznie większe od symulowanych. Aby wskazać przyczynę tego zjawiska przeprowadzono symulację parametryczną odpowiedzi czasowej układu w funkcji pojemności obciążenia wyjścia przedwzmacniacza C_L . Wyniki symulacji wraz z rezultatami pomiarów zaprezentowano na rysunku 3.21. Na zaprezentowanym wykresie krzywa pomiarowa pokrywa się idealnie z krzywą symulowaną dla $C_L = 10$ pF. Pojemność ta wynika z pojemności ścieżek prowadzących od wyjścia przedwzmacniacza do buforów odczytowych (około 5 cm) oraz z pojemności wejściowej wzmacniacza operacyjnego, pracującego jako bufor (patrz rozdział 5.1). Istnienie dodatkowych pojemności uniemożliwia sprawdzenie układu w warunkach



Rysunek 3.21. Porównanie wyników pomiarów z symulacjami odpowiedzi czasowych dla różnych pojemności obciążenia przedwzmacniacza C_L ($C_d=44$ pF, mode=0). Prezentowane rezultaty nie zawierają składowej stałej

identycznych, do jakich projektowany jest przedwzmacniacz. Nie wpływa to jednak na możliwość wykonania pozostałych pomiarów i testów.

W kolejnym kroku wyznaczono wzmocnienia ładunkowe przedwzmacniacza w funkcji pojemności C_d . Wyniki zaprezentowano na rysunku 3.22. Naniesione na wy-



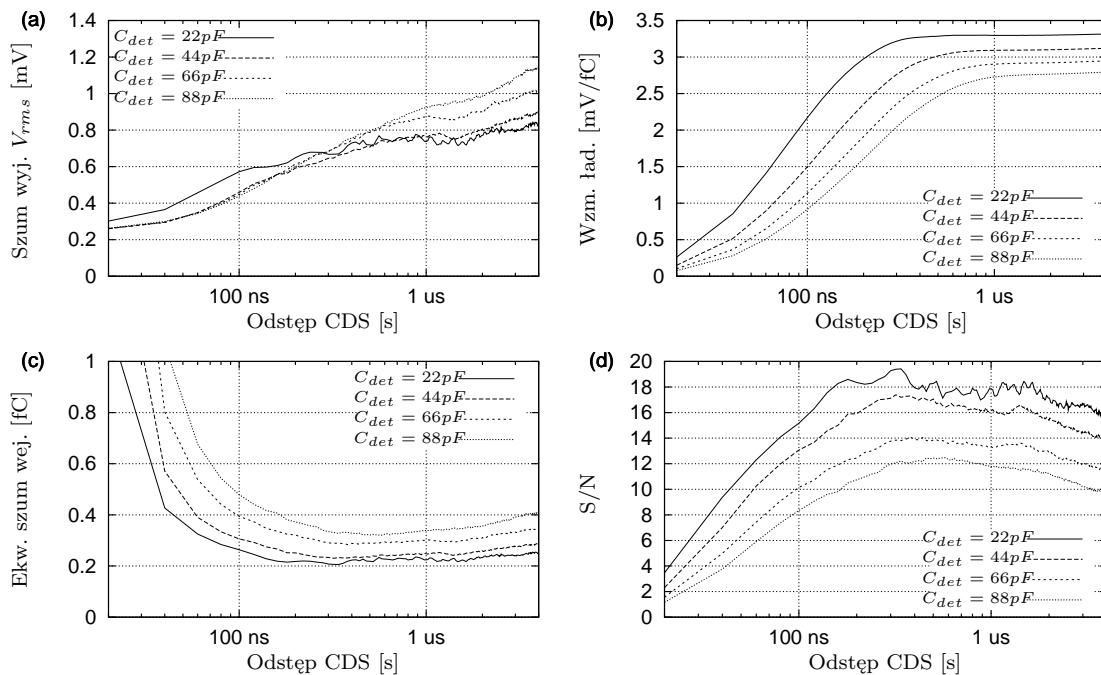
Rysunek 3.22. Wzmocnienie ładunkowe w funkcji pojemności sensora dla różnych trybów pracy. Porównanie wyników symulacji (ozn. sym) z pomiarami. ($C_L = 10$ pF)

kresie wyniki pomiarów odniesione są do wyników symulacji przy obciążeniu przedwzmacniacza wynoszącym $C_L = 10$ pF. Zauważyć można, iż dla trybów małego wzmocnienia (skala na lewej osi) wyniki pomiarów bardzo dobrze zgadzają się z symulacjami. Dla dużego wzmocnienia mierzone wartości wzmocnienia ładunkowego są mniejsze od wartości symulowanych. Po dokładnej analizie stwierdzono, iż przyczyną tego zjawiska jest plan masek kanału. W projekcie linia wyjściowa przekrywa linie wejściową na sporym obszarze. Jak oszacowano przy pomocy narzędzia ekstrahującego elementy pasożytnicze (wynikające z fizycznego ułożenia masek układu) powstała w ten sposób pojemność pasożytnicza (kondensator płaski) ma wartość wynoszącą około 34 fF. Mając na uwadze, że znamionowa pojemność sprzężenia zwrotnego wynosi 250 fF, tak duża wartość pojemności pasożytniczej tłumaczy spadek wzmocnienia o ponad 10 %.

Bardzo ważnymi pomiarami były pomiary mające na celu oszacowanie szumów. Plan podstawowy przewidywał oszacowanie szumów w dziedzinie czasu. Jako że układ podczas fazy resetu powracał zawsze do tego samego punktu pracy (wszystkie potencjały DC były identyczne) oczekiwano, iż po rozwarciu klucza na wejście układu wstrzykiwana będzie stała ilość ładunku (co potwierdzały symulacje). Następnie, w stałym odstępnie czasu (po rozwarciu klucza) planowano próbkować wyjście układu. Odchylenie standardowe rozkładu otrzymanego z odpowiedniej ilości w ten sposób wykonanych eksperymentów mówiłoby o szumach układu. Pierwsze pomiary pokazały, iż ładunek zostający na wejściu układu nie był zawsze taki sam. Fluktuacje poziomu stałego na wyjściu układu po rozwarciu klucza dochodziły do 100 mV. Najprostszym wytłumaczeniem tak dużych fluktuacji wydawały się szумы nadmiarowe typu $1/f$. Aby zweryfikować tę hipotezę wykonano testy polegające na resetowaniu układu raz na około sekundę. Na wyjściu układu w czasie gdy klucz był rozwarty obserwowane fluktuacje nie przekraczały kilku miliwoltów. Pomiary te wykazały, iż te duże fluktuacje nie są związane z szumami. Najlepszym wytłumaczeniem obserwowanego zjawiska na chwilę obecną wydaje się możliwość wstrzykiwania (a właściwie indukowania) ładunku na wejściu układu w chwili rozłączania klucza resetującego. Ponieważ do wyjścia układu przypięta jest stosunkowo długa ścieżka (patrz wyżej) może ona pełnić funkcje anteny, zbierającej niechciane zakłócenia. Obserwowane zakłócenia uniemożliwiły bezpośredni pomiar szumu na wyjściu układu w warunkach, jakie będą w docelowym systemie pomiarowym. Użycie klasycznych metod takich jak pomiar wartości skutecznej sygnału czy użycie analizatora widma również nie mogły być zastosowane, ze względu na konieczność periodycznego resetowania układu.

Aby oszacować wartość szumu zdecydowano się użyć techniki zwanej skorelowanym próbkowaniem podwójnym (ang. Correlated Double Sampling – CDS). Technika ta polega na tym, iż w pierwszej fazie operacji brana jest próbka referencyjna linii bazowej, a następnie brana jest próbka zawierająca już sygnał. Końcowy rezultat otrzymywany jest w wyniku odejmowania tych próbek. Oczywiście, jak zostało to już przedyskutowane w poprzednim rozdziale, poprzez zastosowanie CDS wprowadza się jedną z technik filtracji. Zatem poprawia się także stosunek sygnału do szumu.

Do pomiarów szumowych zebrano statystykę kilku tysięcy przypadków. Każdy przypadek był dany przez szereg próbek zebranych z wyjścia układu. Próbkowanie rozpoczynane było w momencie zwartego klucza resetu. Następnie klucz był rozwierany. Po okresie około $1 \mu\text{s}$ wstrzykiwano ładunek na wejście przy użyciu pojemności testowej. Jako próbkę referencyjną wybierano chwilę czasową poprzedzającą bezpośrednio moment wstrzykiwania ładunku z generatora. W kolejnym kroku wybierano próbkę oddaloną o zadany czas (tzn. czas CDS) od próbki referencyjnej i przeprowadzano na obu próbkach operację CDS. Wartości otrzymywane z poszczególnych eksperymentów były zapamiętywane. Wartość oczekiwana rozkładu tych wartości mówi o amplitudzie sygnału wejściowego, podczas gdy odchylenie standardowe można interpretować jako szumy.



Rysunek 3.23. Wyniki procedur CDS. Na podwykresach zaprezentowano dla zestawu pojemności wejściowych C_d kolejno w funkcji okresu branego pomiędzy próbkami (a) wartość szumów na wyjściu (b) wzmocnienie ładunkowe (c) ekwiwalentny szum wejściowy (d) stosunek sygnału ($MIP = 4 \text{ fC}$) do szumu

Zależność w ten sposób otrzymanych szumów w funkcji czasu CDS, dla różnych pojemności C_d , zaprezentowano na rysunku 3.23a. Zgodnie z oczekiwaniami wartość szumów dla krótkich czasów CDS jest dużo mniejsza od tych uzyskanych dla dużych czasów. Należy mieć na uwadze, iż dla krótkich czasów CDS impuls wyjściowy nie osiąga maksimum swojej amplitudy (druga próbka brana jest podczas narastania wyjścia). Wzmocnienie w funkcji czasu CDS zostało przedstawione na 3.23b. Zauważyć można, iż krzywe nasycają się dopiero po czasie od około 200 ns do 1 μs , odpowiednio dla pojemności C_d wynoszących 22 pF oraz 88 pF. Mając wyznaczone

wzmocnienie układu oraz znając jego szumy (dla zadanego czasu CDS), możliwe jest wyznaczenie ekwiwalentnych szumów wejściowych (rys. 3.23c) oraz stosunku sygnału do szumu (rys. 3.23d). Jak można zauważyć dla czasów CDS z przedziału 200 ns do 4 μ s uzyskany stosunek sygnału (ładunek 4 fC) do szumu jest powyżej 10.

Wyniki pomiarów pokazują, iż zaprojektowany układ działa w pełni funkcjonalnie. Ze względu na pojemności pasożytnicze ścieżek na płytce drukowanej odpowiedzi układu są wolniejsze, co jednak nie wpływa na testowalność układu. Pomiarów szumów ujawniły istnienie problemu zakłóceń. Z tego względu miarodajne pomiary szumów wykonano stosując technikę CDS. Pomiary te wykazały, iż możliwe jest uzyskanie zadanego stosunku sygnału do szumu. Technika CDS okazała się bardzo przydatna podczas eliminacji zakłóceń, co skłania do zastosowania jej sprzętowej realizacji w kolejnych prototypach.

Rozdział 4

Pomiary przetwornika analogowo-cyfrowego

4.1. Metodologia pomiarów przetworników analogowo-cyfrowych

Zadaniem przetworników analogowo-cyfrowych jest konwersja sygnału wejściowego (analogowego) do reprezentacji cyfrowej. Niestety rozmaite parametry procesu technologicznego wpływają degradująco na jakość pracy przetworników, objawia się różnego typu nieliniowościami i błędami konwersji. Zadaniem projektanta jest przygotowanie projektu jak najbardziej odpornego na wszelkiego rodzaju rozrzuty technologiczne (ang. mismatch), temperaturę, itd. Kluczowym aspektem jest więc zrozumienie różnych metryk określających jakość pracy przetworników. W literaturze [IEE00, Sum02] przyjęła się podział na pomiary statyczne oraz dynamiczne. Metryki statyczne mówią o jakości pracy ADC z wolnozmiennymi (prawie statycznymi) sygnałami wejściowymi (w takich aplikacjach jak np. pomiar temperatury). Tutaj kluczowe znaczenie mają takie parametry jak wzmocnienie, offset, monotoniczność czy liniowość. Druga grupa pomiarów to pomiary dynamiczne, gdzie uwaga skupiona jest na sygnałach zmiennych, kluczowe znaczenie mają tutaj szумы i zniekształcenia harmoniczne. Metryki dynamiczne są cenione w świecie aplikacji radiowych, gdzie liniowość i czystość widmowa ma znaczenie fundamentalne. W praktyce podczas optymalizacji układu istotne są wszystkie metryki gdyż wskazują one słabe strony projektu.

4.1.1. Metryki statyczne

Najważniejszymi statycznymi metrykami określającymi jakość pracy przetwornika są: nieliniowość całkowita (ang. Integral Non-Linearity – INL) oraz nieliniowość różniczkowa (ang. Differential Non-Linearity – DNL). Obie te metryki mówią o rzeczywistej dokładności przetwornika i zawierają w sobie błędy kwantyzacji oraz nieliniowości.

Nieliniowość całkową (INL) definiuje się jako odchylenie wartości kodów zwracanych przez przetwornik od linii prostej łączącej punkt zerowy i punkt w którym napięcie jest równe napięciu pełnej skali.

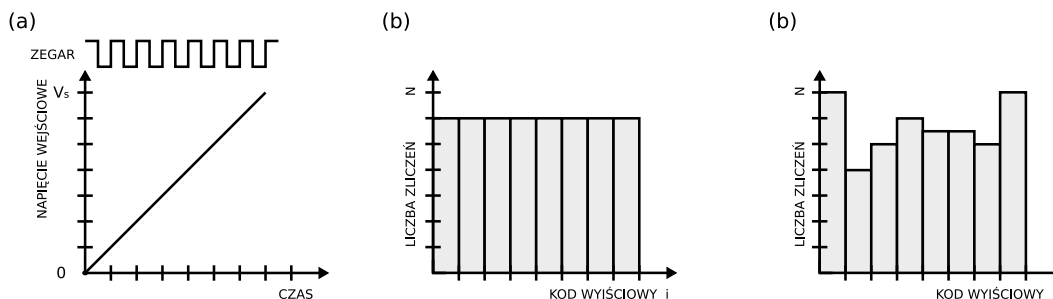
Nieliniowość różniczkowa (DNL) określa szerokość schodka dla danego kodu wyjściowego, znormalizowaną do idealnej szerokości. Wartość DNL dla danego schodka można wyrazić jako:

$$DNL(D_i) = \frac{V_{in}(D_i) - V_{in}(D_{i-1}) - V_{LSB}}{V_{LSB}}, \quad (4.1)$$

gdzie $V_{in}(D_i)$ oraz $V_{in}(D_{i-1})$ są progami przełączania sąsiednich schodków. Jako V_{LSB} najczęściej przyjmuje się wartość wyznaczoną jako średnią po wszystkich schodkach. Zauważyć można, iż wartością idealną jest $DNL = 0$. Korzystając z metryki DNL można również znaleźć tzw. brakujące kody (ang. missing codes). Kody takie w ogóle nie pojawiają się podczas pracy przetwornika lub pojawiają się bardzo rzadko. Standard nakazuje określać kod mianem kodu brakującego, jeśli szerokość danego schodka na wejściu przetwornika jest mniejsza niż 10% V_{LSB} , co sprowadza się do warunku $DNL(D_i) < -0,9$. Istnieje oczywiście ścisły związek pomiędzy INL i DNL. Wartość nieliniowości całkowitej może zostać wyznaczona dla kodu wyjściowego D_m jako:

$$INL(D_m) = \sum_{i=0}^m DNL(D_i). \quad (4.2)$$

Najpopularniejszą metodą mierzenia obu wyżej wymienionych nieliniowości jest tak zwana metoda histogramowa. W metodzie tej na wejście badanego układu podawany jest sygnał narastający liniowo (rys. 4.1a), a wyniki konwersji umieszczane są w zbiorczym histogramie. Przy zadanej częstotliwości próbkowania i zbczu sygnału



Rysunek 4.1. (a) Sygnały podczas pomiaru metodą histogramową. (b) Histogram kodów wyjściowych przetwornika idealnego oraz (c) rzeczywistego

narastającego na każdy poziom przetwornika przypada stała liczba próbek, a co za tym idzie ilość danych kodów wyjściowych w histogramie 4.1b. Jeśli długości poszczególnych poziomów są różne, ma to odzwierciedlenie w histogramie wyjściowym 4.1c. Wysokości poszczególnych przedziałów N_i wykorzystywane są do wyznaczenia nieliniowości różniczkowej jako:

$$DNL(D_i) = \frac{N_i}{N_{avr}}, \quad (4.3)$$

gdzie N_{avr} jest średnią ze wszystkich schodków. Ze względów praktycznych schodki skrajne najczęściej są odrzucane, jako że zawierają one znacznie większą liczbę przypadków. Jest to związane z tym, iż przetwornik dla napięć z poza zakresu pełnej skali wskazuje kody brzegowe. Mając wyznaczone nieliniowości różniczkowe, nieliniowości całkowite wyznacza się przy pomocy zależności 4.2.

4.1.2. Metryki dynamiczne

Dynamiczne metryki dostarczają informacji na temat szumu, dynamicznej liniowości, zniekształceń nieliniowych oraz niepewności czasu próbkowania badanego przetwornika analogowo-cyfrowego. Większość tych metryk jest zależna od amplitudy oraz częstotliwości sygnału wejściowego jak również od częstotliwości próbkowania. W dalszych rozważaniach przyjęto, iż sygnały będą miały amplitudę $V_{fs}/2$ czyli maksymalną amplitudę nie nasycającą przetwornika.

Dyskretna transformata Fouriera

Dyskretna transformata Fouriera (ang. Discrete Fourier Transform – DFT) jest procedurą numeryczną pozwalającą analizować, badać oraz syntetyzować sygnały w sposób dużo bardziej efektywny niż badając sygnały w postaci ciągłej [Ric03]. Dzięki DFT możliwe jest wyznaczenie zawartości częstotliwościowej dowolnego sygnału dyskretnego (w dziedzinie czasu). DFT wywodzi się bezpośrednio z przekształcenia Fouriera, danego dla sygnałów ciągłych [Ric03, Dag01, Zie02], jako dyskretny ciąg $X(m)$ w dziedzinie częstotliwości:

$$X(m) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nm/N}, \quad (4.4)$$

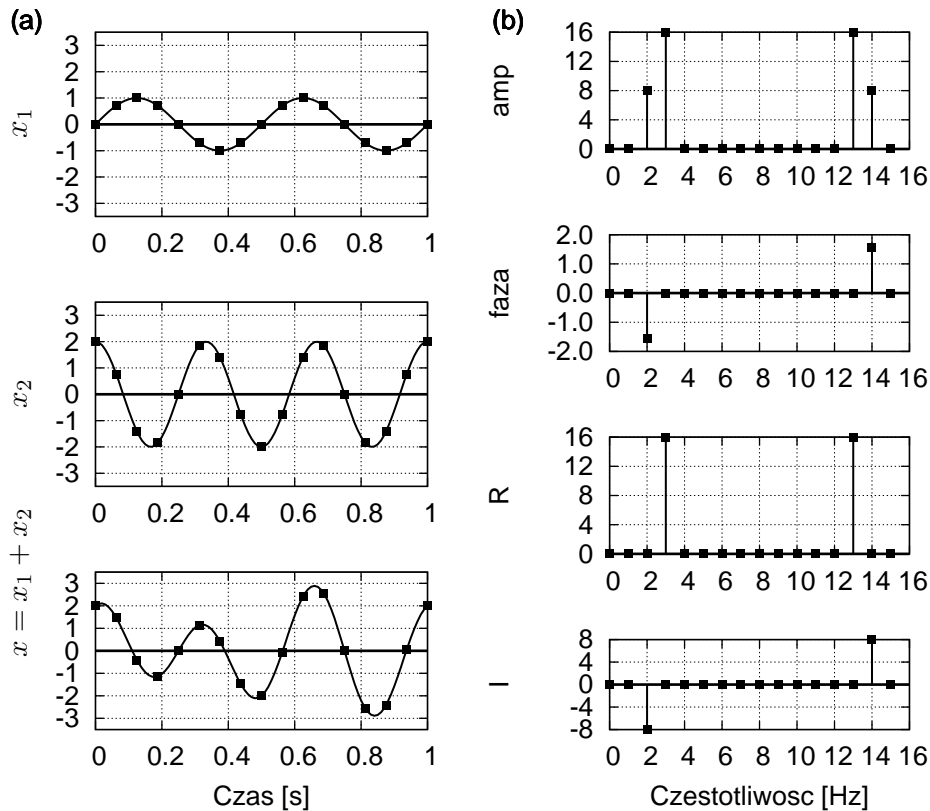
gdzie $x(n)$ to dyskretny N elementowy ciąg wartości sygnału w dziedzinie czasu. Przyjmując, iż próbki w dziedzinie czasu zbierane są w równoodległych chwilach o długości $1/f_s$, gdzie f_s jest częstotliwością próbkowania, można wprowadzić pojęcie częstotliwości podstawowej jako:

$$f_b = \frac{f_s}{N}. \quad (4.5)$$

Analiza częstotliwościowa sygnału $x(n)$ owocuje więc wyznaczeniem wartości $X(m)$ DFT, zwanych prążkami, w punktach osi częstotliwości będących całkowitymi wielokrotnościami częstotliwości podstawowej:

$$f(m) = m \cdot f_b = m \frac{f_s}{N}. \quad (4.6)$$

Częstotliwości takie nazywane będą częstotliwościami bazowymi. Dyskretne przekształcenie Fouriera w ogólności przyporządkowuje więc zespolonemu ciągowi N elementowemu identyczny ciąg. W większości aplikacji sygnał wejściowy ma jednak charakter rzeczywisty (części urojone są równe zero dla wszystkich próbek). Sytuacja



Rysunek 4.2. Przykład dyskretnej transformaty Fouriera. (a) sygnały wejściowe (kolejno x_1, x_2 oraz ich superpozycja) (b) amplituda, faza, część rzeczywista oraz urojona dla prążków wyjściowych transformaty Fouriera

taka implikuje, iż prążki dla $m \geq N/2$ mają charakter nadmiarowy. Dla argumentów $m \in [0, N/2 - 1]$ wartość wyjściowa DFT będzie miała taką samą amplitudę jak $(N - m)$ -ta wartość wyjściowa, kąt fazowy będzie się różnił tylko znakiem [Ric03].

Kolejną bardzo ważną własnością transformaty Fouriera jest jej liniowość. Mówi ona o tym, iż DFT sumy dwóch sygnałów jest równa sumie transformat każdego z sygnałów [Ric03, Zie02]. Dzięki tej własności możliwe jest analizowanie interesujących nas przypadków zawierających wiele składowych, np. sygnał wraz z jego harmonicznymi.

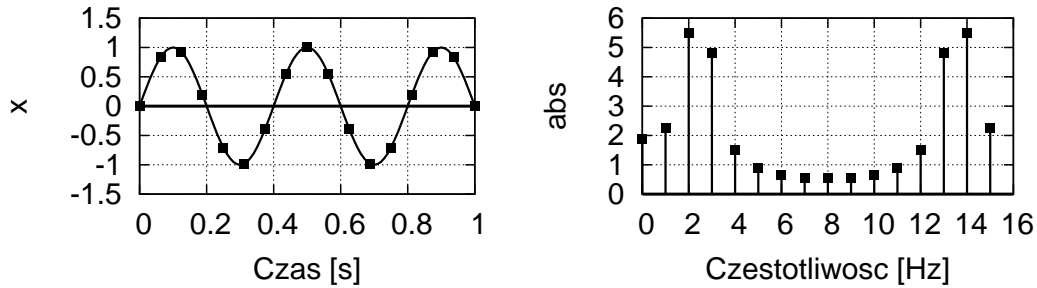
Wyniki transformaty Fouriera przykładowego sygnału (rys. 4.2a):

$$x(n) = \sin\left(2 \cdot \frac{n \cdot 2\pi}{N}\right) + 2 \cdot \sin\left(3 \cdot \frac{n \cdot 2\pi}{N} + \frac{\pi}{2}\right), \quad (4.7)$$

spróbokowanego z częstotliwością 16 Hz zostały zaprezentowane na rysunku 4.2b. W widmie wyjściowym można zaobserwować obecność prążków dla częstotliwości odpowiadających sygnałom wejściowym, jednak ich amplituda jest inna niż oczekiwana. Jeśli rzeczywisty sygnał wejściowy zawiera składową sinusoidalną o amplitudzie A i całkowitej liczbie okresów w przedziale N próbek wejściowych, wówczas amplituda

wyjściowa DFT dla tego przebiegu wynosi $A \cdot (N/2)$. Aby więc odczytać amplitudę z wyników DFT należy podzielić wartości wszystkich składowych przez $(N/2)$.

W powyższym przykładzie częstotliwości były starannie dobrane, tak by w zbiorze wejściowym dana częstotliwość mieściła się całkowitą ilość razy (aby dana częstotliwość była częstotliwością bazową). W przypadkach gdy częstotliwość sygnału wejściowego nie jest częstotliwością bazową można zaobserwować zjawisko wycieku.

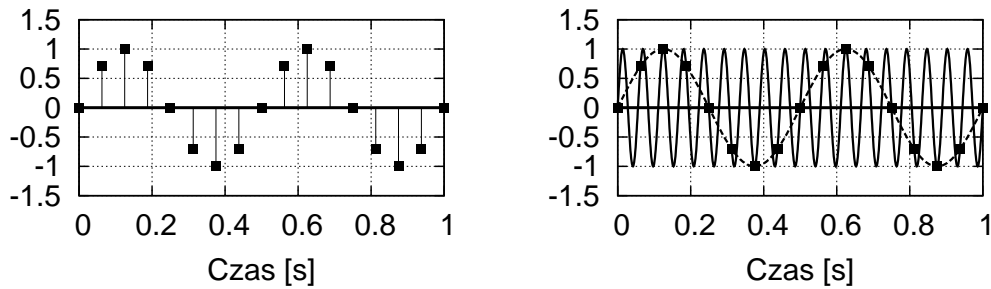


Rysunek 4.3. Przykład wycieku przy próbkowaniu częstotliwości nie będącej częstotliwością bazową

Polega ono na tym, że energia niesiona przez sygnał wejściowy rozplywa się na wszystkie prążki widma wyjściowego. Twierdzenie Parsewala [Ric03, Zie02, Dag01] gwarantuje, iż całkowita energia obserwowana w widmie wyjściowym jest równa energii sygnału wejściowego w dziedzinie czasu. Przykład wycieku zaprezentowano na rysunku 4.3. Własność ta powoduje, że wyniki DFT są jedynie aproksymacją rzeczywistych widm sygnałów wejściowych. Istnieje kilka metod minimalizacji tego zjawiska, gdyż wyeliminować całkiem go się nie da. Do najpopularniejszych należy metoda okien [Ric03, Zie02, Dag01]. Ponieważ jej stosowanie nie jest konieczne w aplikacji omawianej przez autora, nie zostanie ona tu opisana.

Próbkując sygnał w dyskretnych chwilach czasu należy liczyć się z jeszcze jednym nieprzyjemnym zjawiskiem, zwanym aliasingiem. Zjawisko to polega na niejednoznaczności postaci sygnału spróbkowanego w dziedzinie częstotliwości. Dzieje się tak, gdyż dowolny ciąg próbek w dyskretnych chwilach czasowych reprezentuje nieskończenie wiele przebiegów ciągłych. Na rysunku 4.4 zaprezentowano wynik próbkowania sygnału (kropki) z częstotliwością 16 Hz oraz jego możliwe interpretacje (linie). Można wykazać, że nie da się rozróżnić spróbkowanych z częstotliwością f_s wartości przebiegu sinusoidalnego o częstotliwości f_{sig} oraz przebiegu sinusoidalnego o częstotliwości $f_{sig} + kf_s$ jeśli k jest dowolną liczbą całkowitą [Ric03, Zie02, Dag01].

W praktycznych aplikacjach najczęściej do wyznaczania dyskretnej transformaty Fouriera używany jest algorytm szybkiej transformaty Fouriera (ang. Fast Fourier transform – FFT). Algorytm ten wymaga rekordu danych o długości będących potęgą liczby 2, tj. $N = 2^m$ gdzie m jest dowolną liczbą naturalną.



Rysunek 4.4. Przykład zjawiska aliasingu. Kropki to wyniki próbkowania sygnału z częstotliwością 16 Hz. Linie oznaczają możliwe interpretacje: ciągła 18 Hz, przerywana 2 Hz

Dobór parametrów pomiarów

Podczas testowania przetworników ADC z wykorzystaniem DFT wybór parametrów sygnału wejściowego (tj. częstotliwość i amplituda), częstotliwości pracy samego przetwornika oraz długości zbieranego rekordu danych mają charakter zasadniczy. Oczywistym wydaje się fakt, iż aby przetestować cały zakres dynamiczny sygnałów wejściowych podawany sygnał powinien mieć amplitudę bliską $V_{fs}/2$. Wymaganie to zapewnia również maksymalizację stosunku sygnału do szumu. Zapewnienie odpowiedniej amplitudy sygnału wejściowego nie jest jednak warunkiem wystarczającym. Aby uniknąć omawianego wyżej wycieku należy wybrać taką częstotliwość sygnału wejściowego f_{sig} by była częstotłością bazową i zdeponowała całą moc w jednym prążku wyjściowym. Aby uniknąć ciągłego próbkowania tego samego zbioru wartości sygnału wejściowego częstotliwość sygnału wejściowego powinna wynosić:

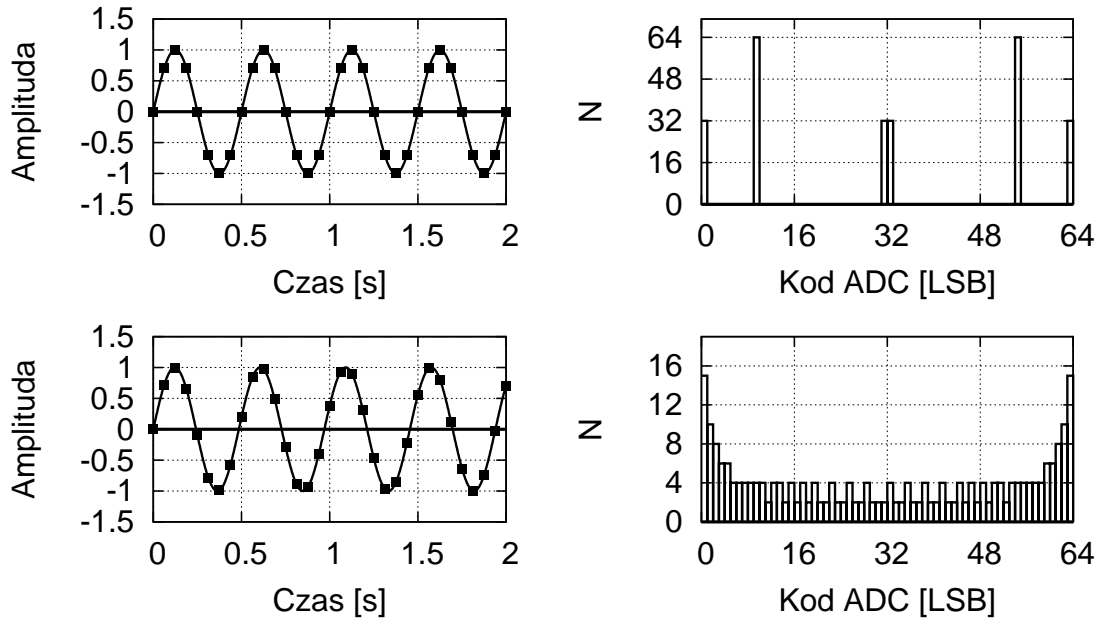
$$f_{sig} = \left(\frac{J}{N} \right) f_s, \quad (4.8)$$

gdzie N jest długością rekordu danych, a J jest dowolną liczbą względnie pierwszą w stosunku do N .

Na rysunku 4.5 przedstawiono przykłady źle oraz dobrze dobranych częstotliwości sygnału wejściowego. Zauważyć można, iż w przypadku (a) przetwornik próbkuje cały czas te same wartości, otrzymane w ten sposób wyniki mogą znacznie odbiegać od rzeczywistych. Przykład (b) pokazuje, iż nieznacznie zmieniając częstotliwość sygnału wyjściowego można diametralnie poprawić tę sytuację. W praktycznych aplikacjach dla N parzystego (czego wymaga algorytm FFT) każde nieparzyste J okazuje się dobrym wyborem.

Całkowite zniekształcenia harmoniczne

Nieliniowości występujące na drodze sygnału analogowego w przetworniku ADC objawiają się podczas jego pracy jako zniekształcenia harmoniczne. Metryka zwana całkowitymi zniekształceniami harmonicznymi (ang. Total Harmonic Distortion –



Rysunek 4.5. Wpływ wybranej częstotliwości sygnału wejściowego na testowalność przetwornika. Próbkowanie wykonano z częstotliwością $f_s=16$ Hz, zebrano $N=256$ próbek sygnału. Do kwantyzacji wykorzystano przetwornik 6 bitowy (64 możliwe stany wyjściowe). (a) sygnał wejściowy o częstotliwości 4 Hz ($J=32$) oraz odpowiadający mu histogram kodów (b) sygnał wejściowy o częstotliwości $((33/256) \cdot 16)$ Hz ($J=33$) oraz odpowiadający mu histogram kodów

THD) opisuje degradację stosunku sygnału do zniekształceń, wprowadzoną przez zniekształcenia harmoniczne. Zdefiniowana jest jako:

$$THD = 20 \log_{10} \left(\sqrt{\frac{\sum_{n=2}^{N_H+1} X_{avr}(f_{h[n]})^2}{X_{avr}(f_{sig})^2}} \right), \quad (4.9)$$

gdzie $f_{h[n]}$ to częstotliwości wyznaczane jako kolejne całkowite wielokrotności częstotliwości f_{sig} rzutowane na przedział częstotliwości objętych przez DFT, czyli od zera do f_s :

$$f_{h[n]} = (n \cdot f_{sig}) \text{ modulo } (f_s). \quad (4.10)$$

Wartości $X_{avr}(i)$ są wyznaczane jako średnie z modułów kilku pomiarów prążka i , natomiast N_H jest liczbą harmonicznych uwzględnianych w obliczeniach (standardowo wynosi 10). Wyrażenie to opisuje stosunek mocy niesionej przez harmoniczne sygnału do mocy niesionej przez sygnał. W tym momencie należy zauważyć, że nieodpowiedni dobór częstotliwości sygnału spowoduje odpłynięcie mocy z prążka sygnałowego do wszystkich pozostałych prążków, fałszując wynik.

Stosunek sygnału do szumu

Ważnym parametrem mówiącym o jakości pracy przetwornika jest stosunek sygnału do szumu nie będącego zniekształceniami harmonicznymi (ang. Signal to Non-Harmonic Ratio – SNHR). Metryka ta mówi o rzeczywistych szumach przetwornika wynikających z efektów kwantyzacji czy szumów elementów wewnętrznych (tranzystory, rezystancje). Zdefiniowana jest jako stosunek mocy sygnału do mocy w pozostałej części pasma z wyłączeniem częstotliwości harmonicznych:

$$SNHR = 20\log_{10} \left(\sqrt{\frac{X_{avr}(f_{sig})^2}{\sum_{i=1}^{2^N-1, i \neq f_{h[n]}} X_{avr}(f_i)^2}} \right). \quad (4.11)$$

Stosunek sygnału do szumu i zniekształceń

Parametrem określającym całkowitą jakość pracy przetwornika jest stosunek sygnału do szumu i zniekształceń (ang. Signal to Noise and Distortion – SINAD). Jest zdefiniowany jako stosunek energii niesionej przez sygnał do całkowitej energii szumów i zniekształceń:

$$SINAD = 20\log_{10} \left(\sqrt{\frac{X_{avr}(f_{sig})^2}{\sum_{i=1}^{2^N-1, i \neq f_{sig}} X_{avr}(f_i)^2}} \right). \quad (4.12)$$

Zakres wolny od zniekształceń

Metryką bardzo ważną szczególnie w aplikacjach telekomunikacyjnych jest metryka mówiąca o zakresie wolnym od zniekształceń (ang. Spurious Free Dynamic Range – SFDR). Zdefiniowany jest jako stosunek mocy sygnału do mocy największego zniekształcenia:

$$SFDR = 20\log_{10} \left(\sqrt{\frac{X_{avr}(f_{sig})^2}{\max_{i=1}^{2^N-1, i \neq f_{sig}} (X_{avr}(f_i)^2)}} \right). \quad (4.13)$$

W praktycznych aplikacjach czynnikiem ograniczającym wielkość SFDR są zniekształcenia harmoniczne.

4.2. Implementacja potokowego przetwornika ADC

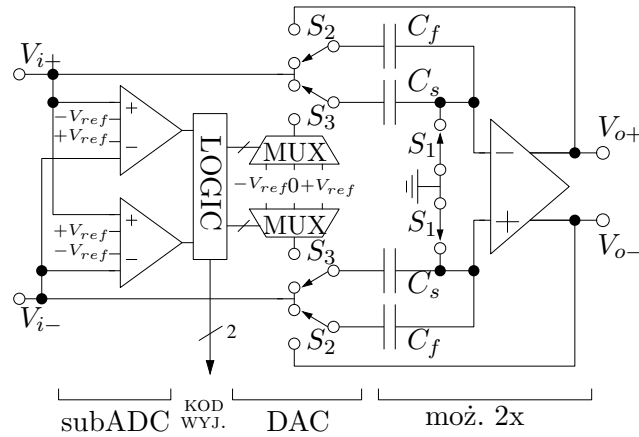
Na obecnym etapie prac tylko główny rdzeń ADC został zaprojektowany [ISK08b]. W jego skład wchodzi obecnie:

- osiem 1,5-bitowych stopni przetwornika potokowego,
- układ generacji sygnałów zegarowych,
- układ generacji prądów polaryzujących.

Aby uprościć plan masek zrezygnowano z dziewiątego stopnia, który powinien zawierać tylko komparatory. Ważnym elementem każdego ADC jest układ próbkująco-pamiętający (ang. sample and hold), który jednak nie został zaimplementowany w obecnej wersji prototypu. Przyczyną takiego rozwiązania jest planowane umieszczenie go na wyjściu kanałów elektroniki front-end. Aby zapewnić

pełną testowalność układu zdecydowano, że wszystkie napięcia referencyjne będą dostarczane z zewnątrz.

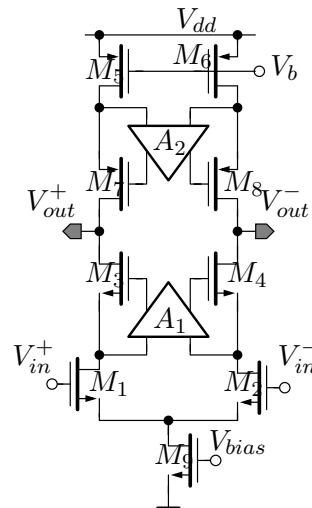
Wszystkie z 1,5-bitowych stopni są niemal identyczne (rys. 4.6), różnią się one tylko wartościami pojemności pamiętających C_s , C_f oraz wartościami prądów płynących w głównych gałęziach wzmacniaczy. Skalowanie takie ma na celu zmniejszyć pobór mocy.



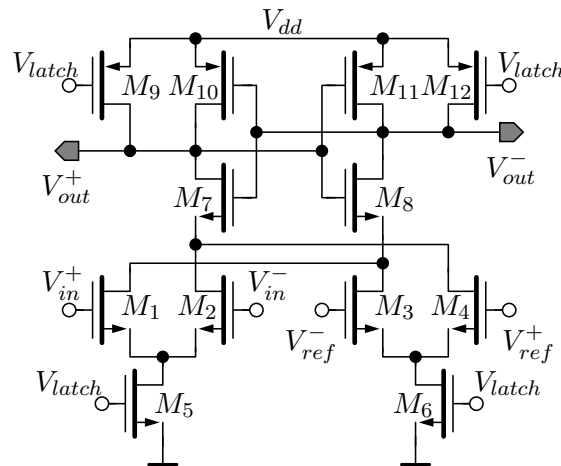
Rysunek 4.6. Schemat ideowy stopnia 1,5-bitowego

Krytycznym elementem potokowego przetwornika ADC jest w pełni różnicowy wzmacniacz operacyjny. Schemat wzmacniacza został zaprezentowany na rysunku 4.7. Konfiguracja teleskopowego wzmacniacza kaskadowego została wybrana ze względu na bardzo dobry stosunek szybkości do wydzielanej mocy. Aby uzyskać wystarczająco duże wzmocnienie zastosowano układy podbijające wzmocnienie (ang. gain boosting) zarówno w dolnej (A1) jak i górnej (A2) gałęzi. Symulowane wzmocnienie w otwartej pętli takiego wzmacniacza wynosi około 100 dB, co jest wartością nadmiarową w stosunku do wymaganych 10 bitów. Oba wzmacniacze podbijające wzmocnienie są w pełni różnicowymi wzmacniaczami jednostopniowymi. We wszystkich wzmacniaczach zastosowano ciągłe sprzężenie zwrotne stabilizujące poziom stały (ang. Continuous Common Mode Feedback – CMFB). Jak dyskutowano w rozdziale 2.4.3 architektura z redundancją informacji znacząco redukuje wymagania stawiane komparatorom. W rozważanym projekcie przy napięciach referencyjnych na poziomie 0,5V rozrzut napięć progowych może sięgać ~ 100 mV, nie degradując znacząco pracy przetwornika. Dla potrzeb tego projektu wybrano architekturę prostego, dynamicznego zatrasku [SWH00], którego schemat został zaprezentowany na rysunku 4.8. Zaletą takiej konfiguracji jest to, iż napięcia progowe są generowane bezpośrednio w komparatorze. Ich wartość jest dana przez parametry tranzystorów M1, M2, M3, M4 [SWH00].

Plan masek prototypowego układu przetwornika zaprezentowano na rysunku 4.9. Wyróżnić można 8 stopni (blok MDAC oraz subADC), układ generacji prądów polaryzujących oraz układ generujący sygnały cyfrowe sterujące kluczami. Całość zajmuje powierzchnię $1,1 \times 1,15$ mm² (włącznie z padami).



Rysunek 4.7. Schemat w pełni różnicowego wzmacniacza z podbijanym wzmocnieniem



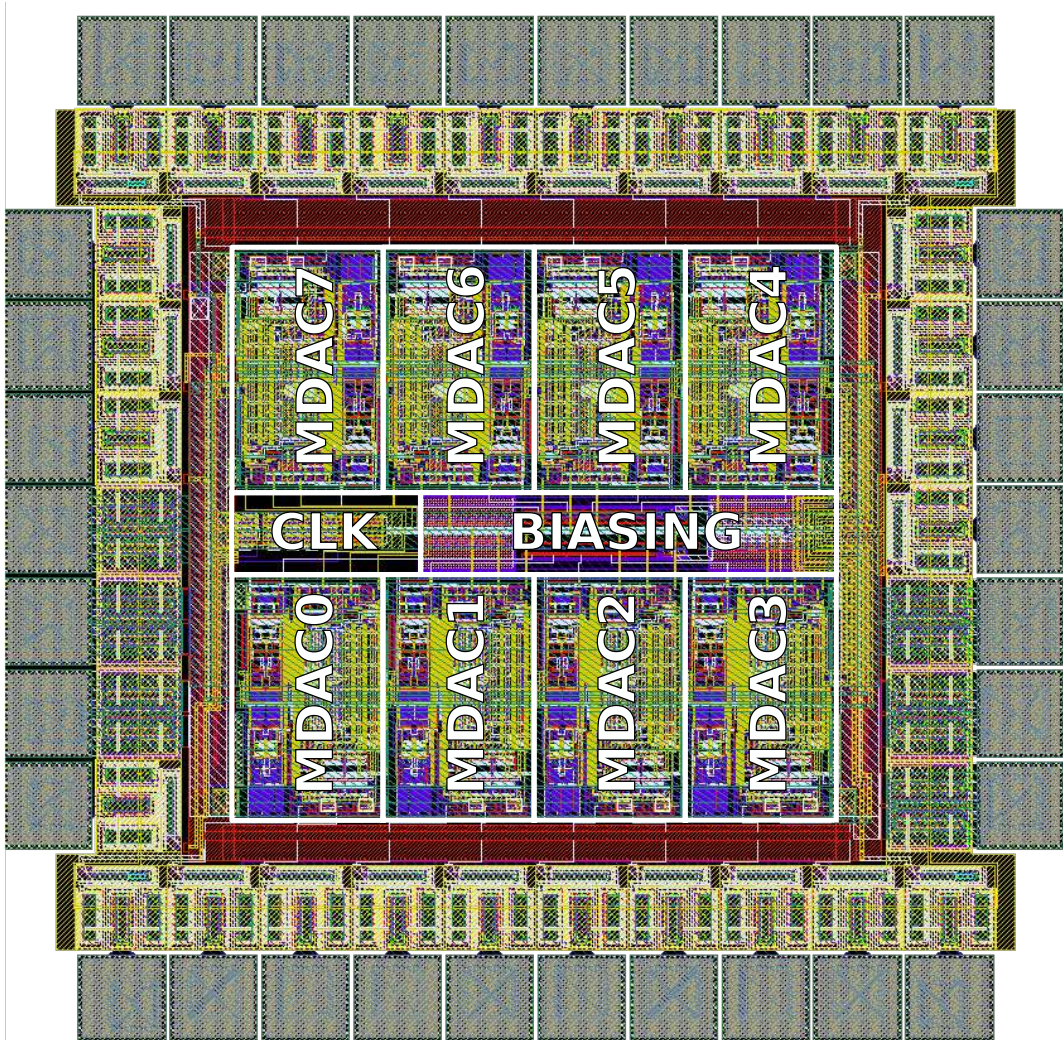
Rysunek 4.8. Schemat komparatora w konfiguracji zatrzasku dynamicznego (ang. latched comparator)

Rysunki 4.10 oraz 4.11 przedstawiają fotografie zaprojektowanych i przybondowanych do dedykowanej płytki PCB prototypowych układów analogowo-cyfrowych przetworników potokowych (kolejno widok z góry oraz perspektywiczny). Na fotografiach zauważyć można fragment dedykowanej płytki drukowanej (patrz rozdział 5.1) wraz z połączeniami (ang. bonds).

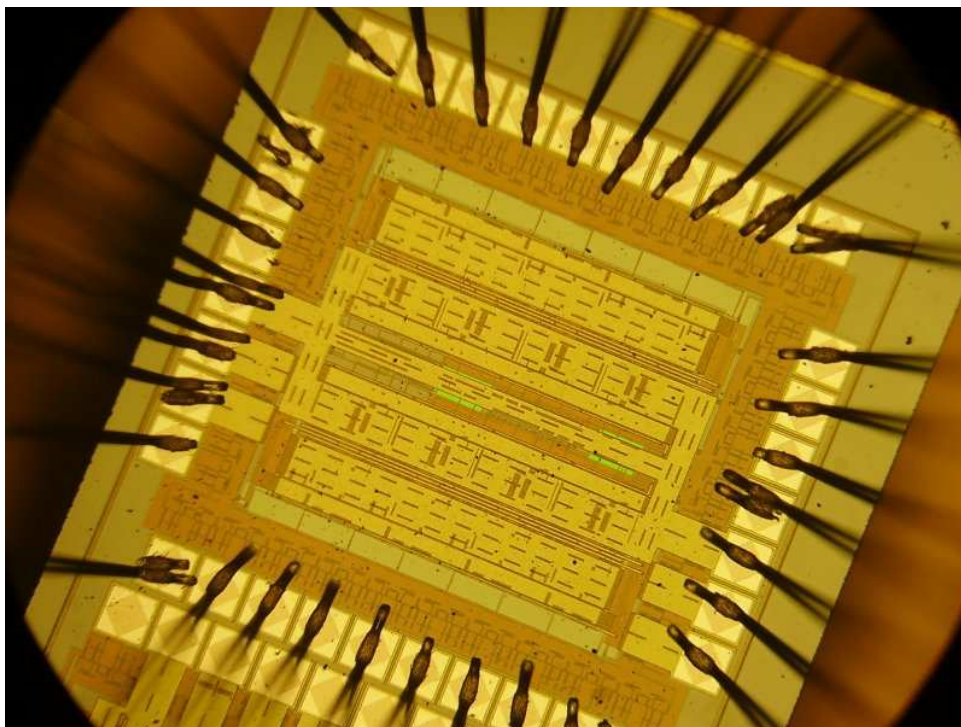
4.3. Pomiary parametrów ADC

4.3.1. Stanowisko pomiarowe

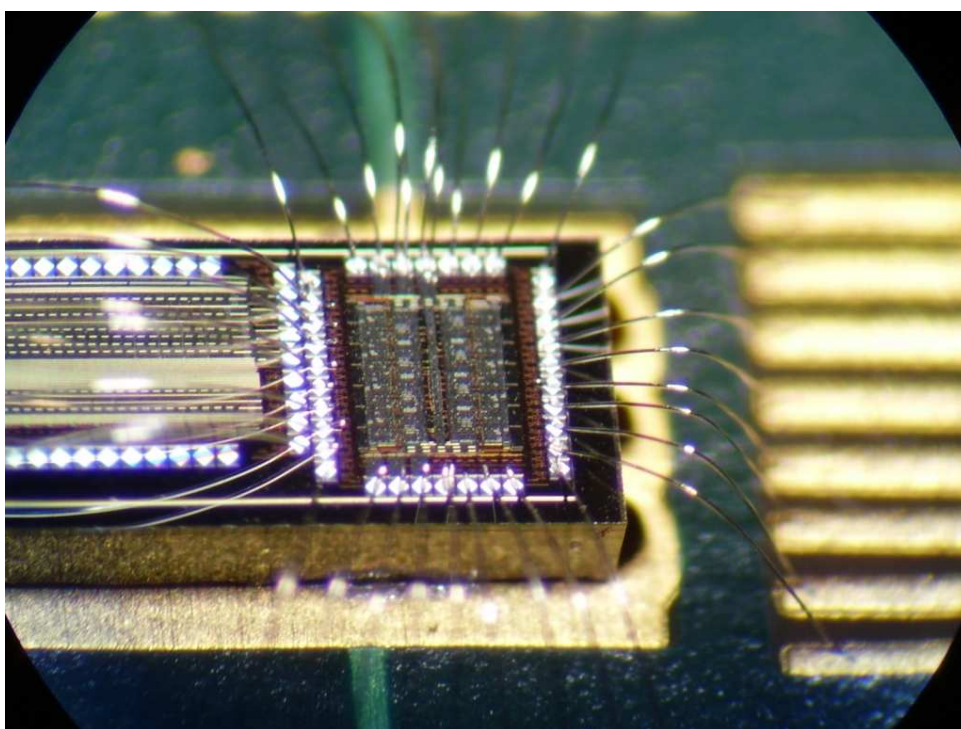
Schemat kompletnego stanowiska pomiarowego został zaprezentowany na rysunku 4.12. Jako generator sygnałów sterujących wykorzystano programowalny ge-



Rysunek 4.9. Plan masek zaprojektowanego układu scalonego przetwornika analogowo-cyfrowego. Zaprezentowano stopnie mnożąco-odejmujące (MDAC0–MDAC7) wraz z sub-ADC, układ generacji przebiegów sterujących (CLK) oraz układy generacji napięć i prądów referencyjnych (biasing)

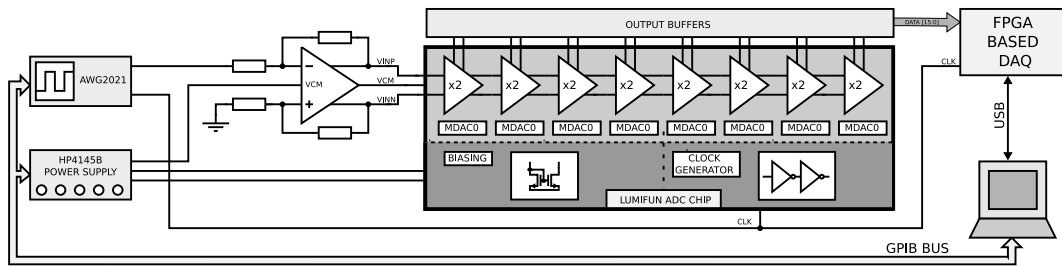


Rysunek 4.10. Widok z góry prototypowego układu ADC



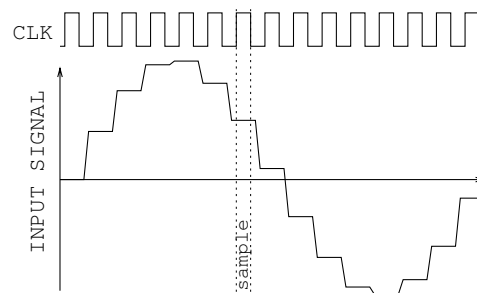
Rysunek 4.11. Widok perspektywiczny prototypowego układu ADC

nerator funkcyjny AWG2021 firmy Tektronix o rozdzielczości 12 bitów i maksymalnej częstotliwości pracy wynoszącej 250 MHz. Do pomiarów statycznych generator ten



Rysunek 4.12. Schemat blokowy systemu pomiarowego

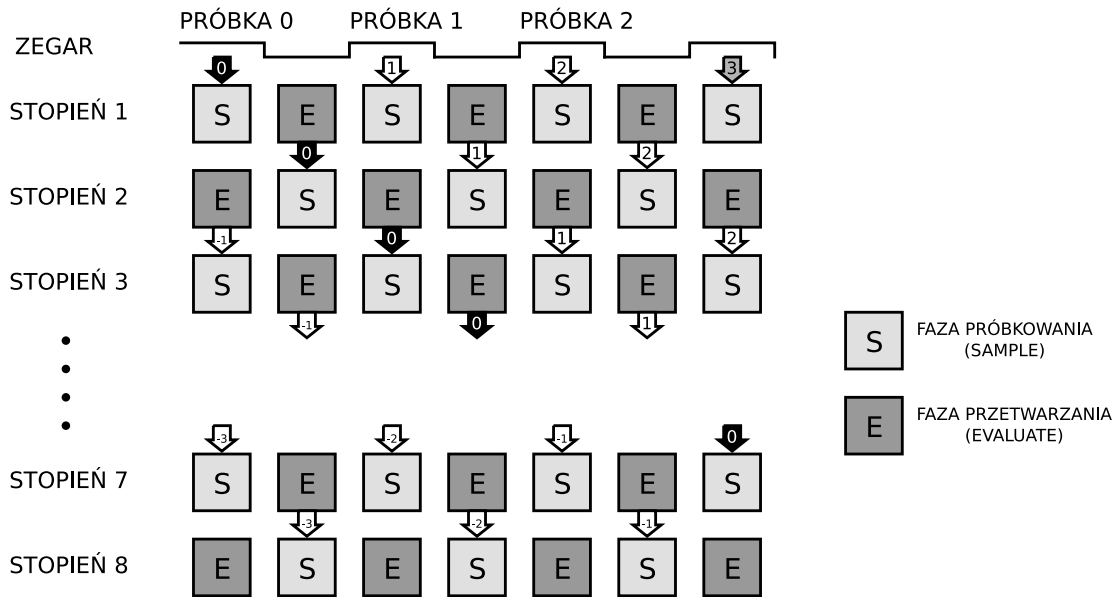
został wykorzystany jako precyzyjne źródło napięcia, natomiast podczas pomiarów dynamicznych generował sinusoidalny sygnał schodkowy zaprezentowany na rysunku 4.13. Rozwiązanie takie zapewnia stałość sygnału wejściowego podczas całego okresu próbkowania. Zostało ono wymuszone tym, że układ próbkująco-pamiętający nie został wbudowany w testowaną strukturę przetwornika. Ponieważ generator posiada wyjście nieróżnicowe (ang. single ended) konieczne było wykonanie konwertera, który został umieszczony na płycie testowej (patrz rozdział 5.1).



Rysunek 4.13. Przykładowe przebiegi sygnału wejściowego oraz zegara podczas pomiarów dynamicznych

Wszystkie niezbędne napięcia referencyjne oraz polaryzujące zostały dostarczone przy użyciu precyzyjnego zasilacza HP4145B kontrolowanego przez interfejs GPIB. Konfiguracja taka pozwoliła wyznaczyć wpływ poszczególnych sygnałów polaryzujących na jakość pracy układu.

Ponieważ bloki korekcji cyfrowej nie zostały zaimplementowane w obecnej wersji przetwornika, wszystkie 16 linii cyfrowych (po 2 na stopień) musi być odczytywane. W praktyce do tego celu wykorzystano szybki system akwizycji danych cyfrowych wykonany w oparciu o układy programowalne FPGA i podłączony do komputera PC za pomocą interfejsu USB (patrz rozdział 5.3). Korekcja cyfrowa wykonywana



Rysunek 4.14. Przepływ danych w potokowym przetworniku analogowo-cyfrowym

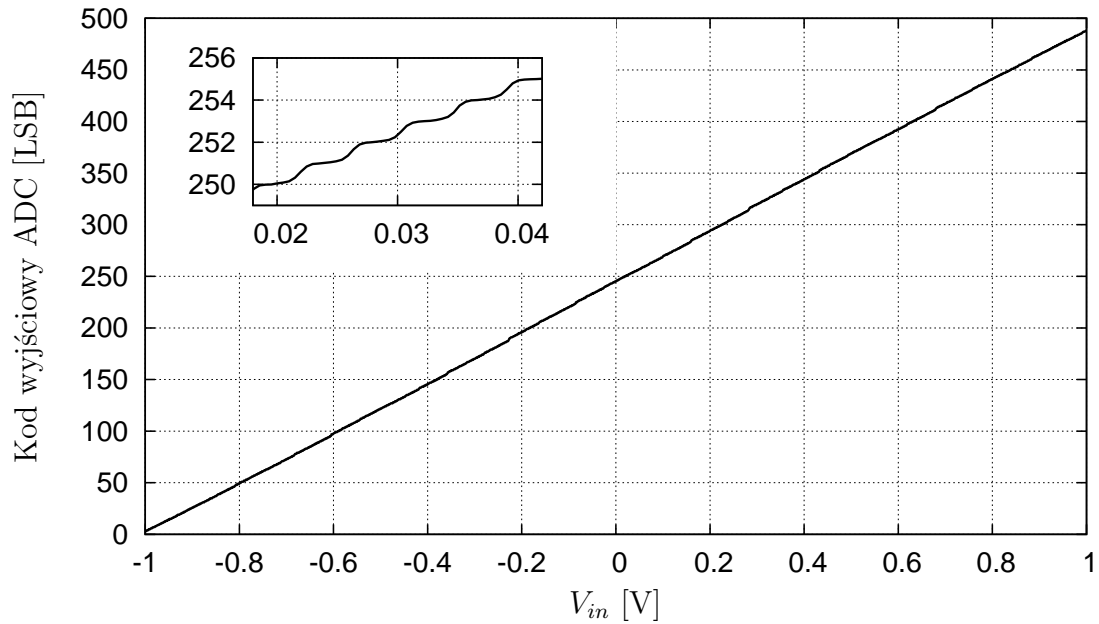
jest po stronie oprogramowania. Rozwiązanie takie pozwala na precyzyjną analizę funkcji przejścia poszczególnych stopni, włączając w to wzmocnienie danego stopnia oraz progi przełączania poszczególnych komparatorów. Szczegółowa analiza została zaprezentowana w pracy [ISK08b]. Należy mieć na uwadze, że wyjścia cyfrowe nie posiadają żadnych linii opóźniających. Oznacza to, iż podczas jednego cyklu zegara na wyjściach znajdują się informacje z 4 kolejnych próbek (rysunek 4.14). Sytuacja ta wymusza implementację programowych linii opóźniających w celu synchronizacji czytanych danych.

4.3.2. Pomiary statyczne

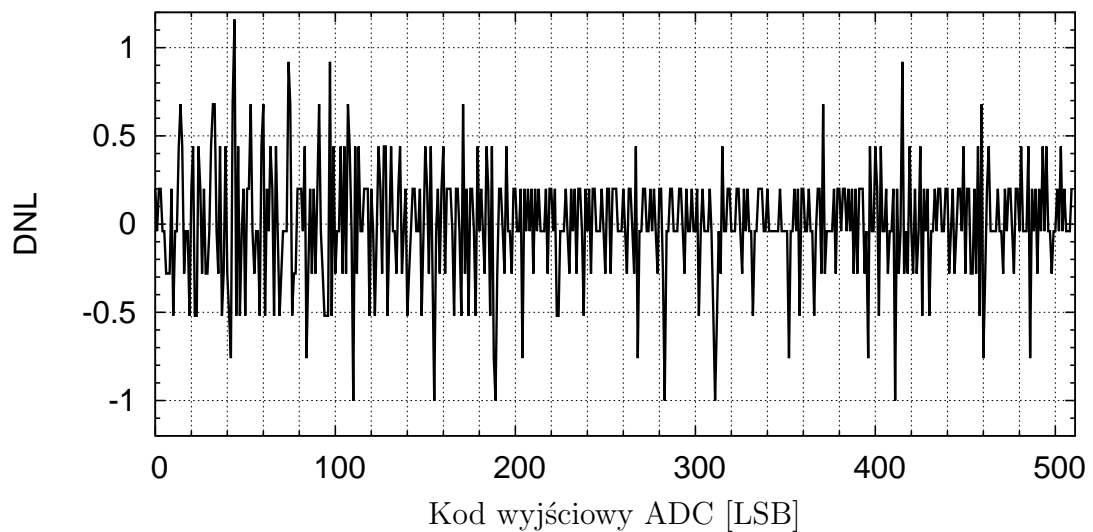
Testy przetwornika rozpoczęto od wykonania pomiarów statycznych. Wszystkie pomiary opisane w tym podrozdziale wykonano przy częstotliwości próbkowania wynoszącej 10 MHz. Pierwszym wykonanym pomiarem był pomiar funkcji przeniesienia przetwornika, którego wyniki zaprezentowano na rysunku 4.15. Na powyższym rysunku widać, iż ADC pracuje poprawnie w całym zakresie napięć wejściowych.

W kolejnym kroku przystąpiono do precyzyjnych pomiarów liniowości układu. Na rysunku 4.16 zaprezentowano wyniki pomiarów nieliniowości różniczkowej. Dla większości kodów DNL pozostaje w granicach $\pm 0,5$ LSB, jednak występuje kilka kodów, gdzie nieliniowość jest większa. Sześć kodów dla których DNL wynosił mniej niż $-0,9$ zostało zidentyfikowanych jako kody brakujące, zgodnie z [IEE00].

Jako ostatni z pomiarów statycznych wykonano pomiar nieliniowości całkowitej, którego wyniki zaprezentowano na rysunku 4.17. Z rysunku można odczytać, że

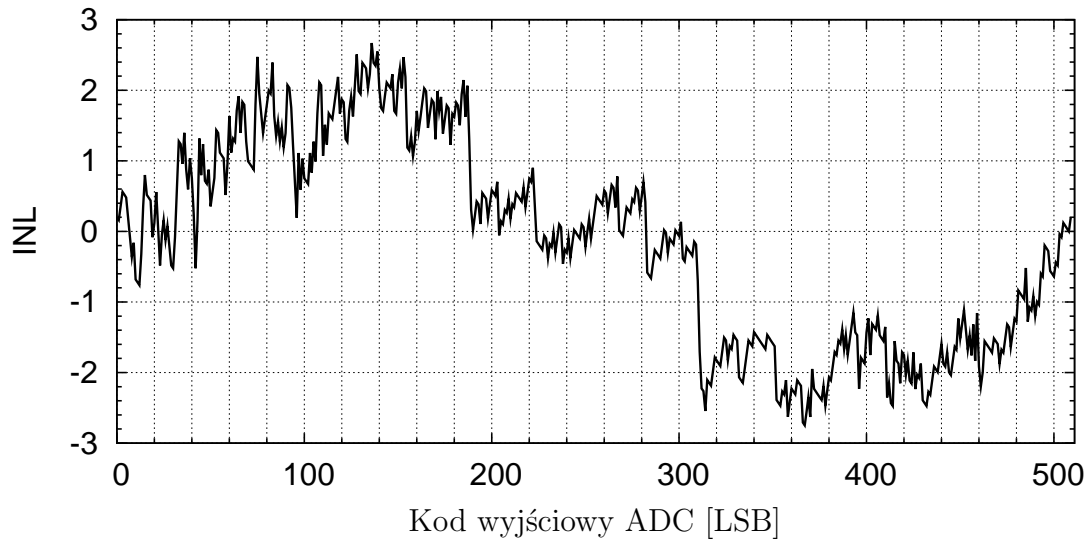


Rysunek 4.15. Pomiar statyczny funkcji przenoszenia ADC. W lewym górnym rogu zaprezentowano zbliżenie fragmentu krzywej



Rysunek 4.16. Pomiar statyczny nieliniowości różniczkowej

nieliniowość całkowita dla większości kodów wyjściowych zawiera się w przedziale ± 3 LSB. Wyznaczona na podstawie krzywej efektywna liczba bitów (ENOB) wynosi 6,65.



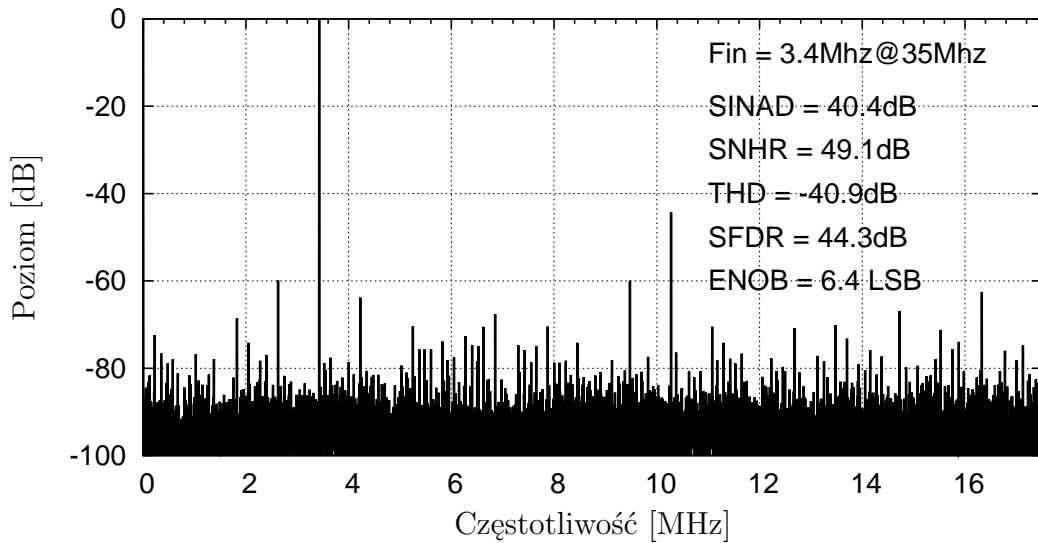
Rysunek 4.17. Pomiar statyczny nieliniowości całkowitej

4.3.3. Pomiary dynamiczne

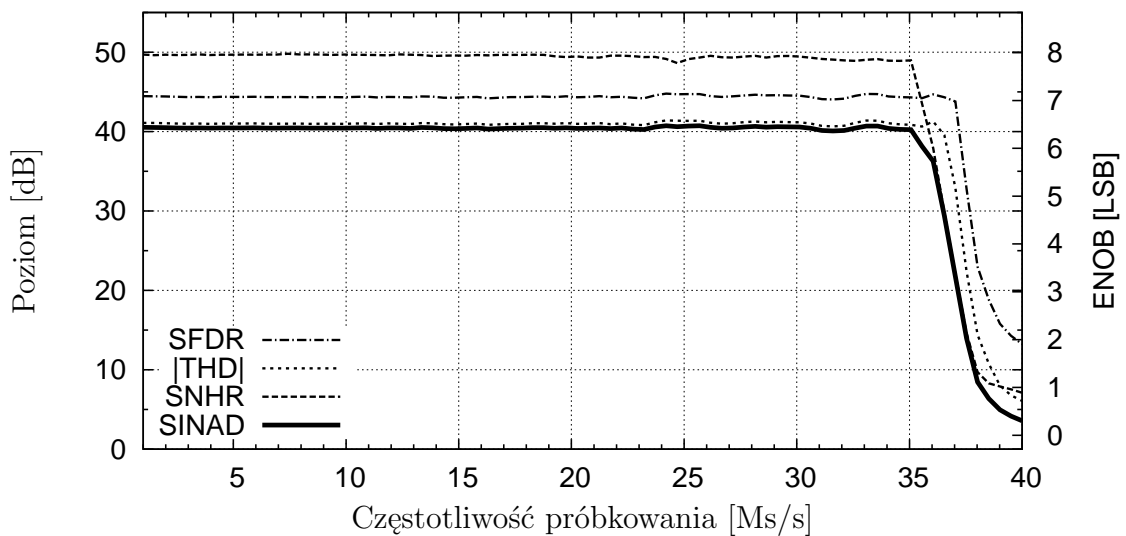
Pomiary dynamiczne wykonywane były dla sinusoidalnego sygnału wejściowego przy różnych częstotliwościach próbkowania. Na rysunku 4.18 zaprezentowano przykładowe widmo otrzymane podczas próbkowania z częstotliwością 35 MSps sygnału sinusoidalnego o częstotliwości 3,4 MHz oraz amplitudzie równej pełnej skali przetwornika (0 dB).

Dla idealnego przetwornika 8 bitowego stosunek sygnału do szumu wynikający z procesu kwantyzacji wynosi 49,92dB (równanie 2.37). Wyznaczona wartość współczynnika SNHR mówiącego o stosunku sygnału do szumu nie będącego zniekształceniami harmonicznymi wynosi 49,1 dB co jest wartością bliską idealnej. Współczynnik sygnału do zniekształceń i szumu (SINAD) wynosi 40,4 dB. Można zauważyć, iż jest on zdominowany przez zniekształcenia harmoniczne (THD), które kształtują się na poziomie -40,9 dB. Szczegółowe studia [ISK08b] oraz symulacje wykazały, iż zachowanie to spowodowane jest niedopasowaniem wzmocnienia poszczególnych stopni MDAC (wzmocnienie jest mniejsze od wymaganego 2). Zakres wolny od zniekształceń (SFDR) wynosi 44,5 dB i dany jest przez trzecią harmoniczną.

W dalszej kolejności przeprowadzono pomiary mające na celu pokazać wpływ poszczególnych parametrów na pracę przetwornika. Na rysunku 4.19 pokazano zależność podstawowych parametrów dynamicznych od częstotliwości próbkowania (dla spójności wykresu namalowany został moduł parametru THD). Na wejście podawany był sygnał o amplitudzie bliskiej amplitudzie pełnej skali i częstotliwości wynoszącej około 1/10 częstotliwości próbkowania. Na przedstawionym wykresie można zauważyć, że przetwornik pracuje prawidłowo (bez jakiegokolwiek degradacji parametrów) do częstotliwości 36 MHz. Zachowanie takie wynika z ograniczonego pasma wzmacniacza i jest zgodne z symulacjami.



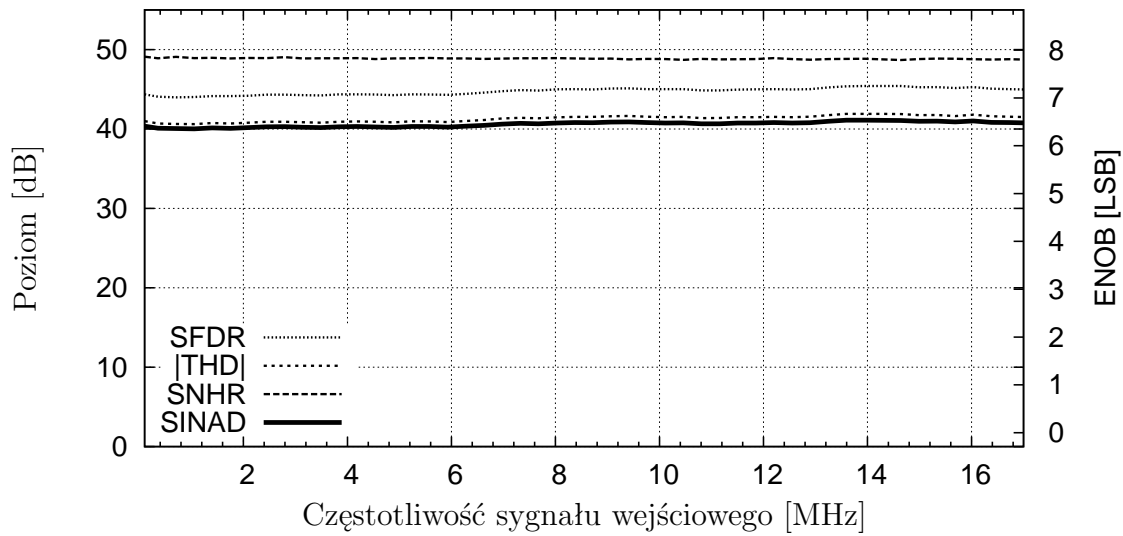
Rysunek 4.18. Przykładowa transformata Fouriera sygnału sinusoidalnego o częstotliwości 3,4 MHz próbkowanego z częstotliwością 35 MSps.



Rysunek 4.19. Pomiary dynamiczne w funkcji częstotliwości próbkowania

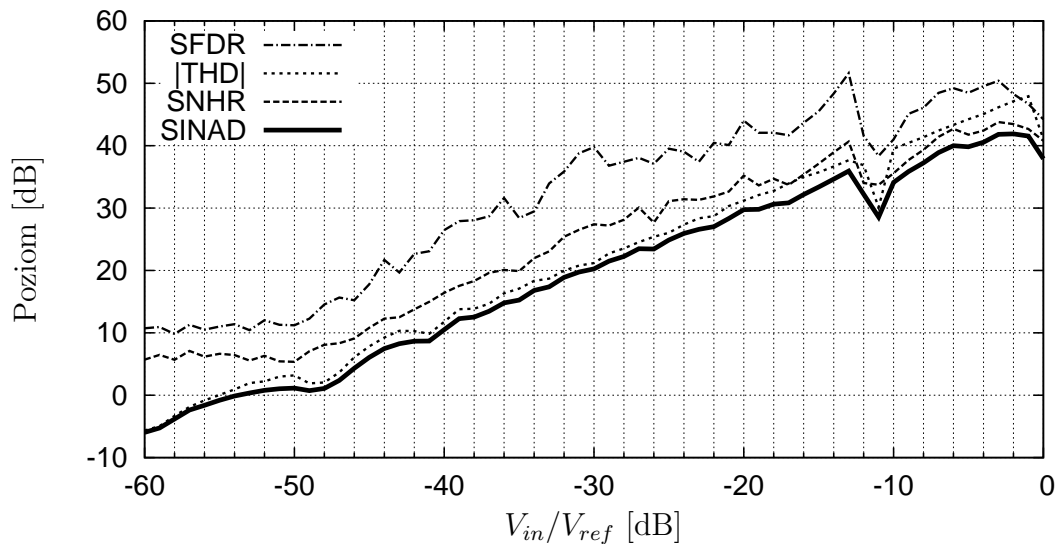
Kolejny pomiar miał na celu ukazać wpływ częstotliwości sygnału wejściowego na poszczególne metryki, co zostało pokazane na rysunku 4.20. Zgodnie z oczekiwaniami przetwornik pracuje poprawnie w całym zakresie częstotliwości pozostających w zgodzie z kryterium Nyquist'a [Ric03, Zie02].

Wpływ amplitudy sygnału wejściowego na jakość pracy przetwornika zademonstrowano na rysunku 4.21. Zgodnie z oczekiwaniami wszystkie metryki zależą liniowo od amplitudy sygnału wejściowego. Dzieje się tak dlatego, iż wszystkie metryki są



Rysunek 4.20. Pomiary dynamiczne w funkcji częstotliwości sygnału wejściowego. Częstotliwość próbkowania 35 Msps

zdefiniowane jako stosunki mocy niesionej przez sygnał. Poszarpany charakter obserwowanych krzywych wynika z nieliniowości wprowadzonych przez kolejne stopnie. Sytuacja ta widoczna jest najlepiej dla wartości napięć zbliżonych do progu przełączania pierwszego stopnia ($1/4V_{ref} \approx -12$ dB).



Rysunek 4.21. Pomiary dynamiczne w funkcji amplitudy sygnału wejściowego. Częstotliwość próbkowania 35 Msps. Częstotliwość sygnału wejściowego 3,4 MHz

Zarówno pomiary statyczne jak i dynamiczne pokazały słabsze strony badanego przetwornika. Na obecnym etapie prac wydaje się, że poprawy wymagają przede

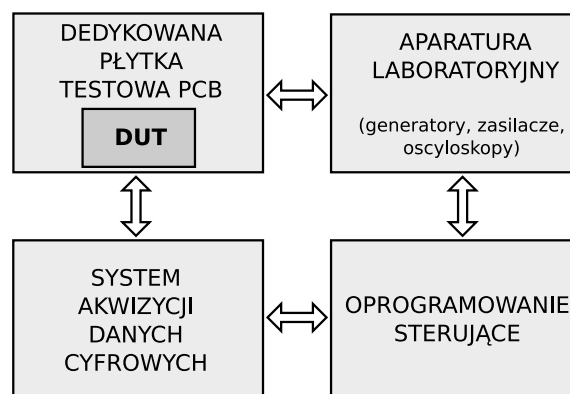
wszystkim zaobserwowane nieliniowości. Dokładne studia wykazały, iż ich przyczyną jest zbyt niskie wzmocnienie w poszczególnych stopniach mnożąco-odejmujących. Symulacje wykazały, iż przyczyny tego niedopasowania leżą w niewystarczająco precyzyjnych sekwencjach czasowych sterujących kluczami, jak również po stronie topologii fizycznej niektórych struktur (ang. layout). Otrzymane wyniki szumowe zgadzają się z przewidywaniami i potwierdzają poprawność projektu. Wyniki przeprowadzonych pomiarów i zrozumienie zaobserwowanych niedoskonałości stanowią podstawowy materiał wspomagający przy projekcie następnej wersji przetwornika ADC. Na koniec należy dodać, iż pełna charakteryzacja przetwornika ADC była możliwa dzięki zaprojektowaniu i zrealizowaniu dedykowanego systemu akwizycji danych, opisanego w rozdziale 5.3.

Rozdział 5

Stanowisko testowe i system akwizycji danych

Projekt specjalizowanych układów scalonych to zaledwie pierwszy krok na drodze do stworzenia w pełni funkcjonalnych systemów pomiarowych. Równie ważne jak projektowanie jest przetestowanie poprawności działania układów oraz ich parametryzacja, w celu sprawdzenia czy udało się spełnić stawiane wymagania. Aby efektywnie przeprowadzać testowanie konieczne jest odpowiednie środowisko testowe. Schematycznie środowisko przygotowane na potrzeby testowania zaprojektowanych prototypów przedstawiono na rysunku 5.1. Składa się ono z:

- dedykowanej płytki drukowanej, na której znajdują się niezbędne zewnętrzne układy elektroniczne (bufory, pojemności odsprężające, etc...), a przede wszystkim do której zostaną przybondowane zaprojektowane struktury testowe.
- systemu akwizycji danych cyfrowych i analogowych, umożliwiającego zbieranie danych specyficznych dla badanego układu,
- przyrządów laboratoryjnych niezbędnych do zapewnienia polaryzacji, generacji wymuszeń, obserwacji sygnału (zasilacze, generatory, oscyloskopy, etc...),
- oprogramowania sterującego pracą wszystkich urządzeń, przetwarzającego dane oraz prezentującego wyniki.



Rysunek 5.1. Schemat systemu pomiarowego

Poniżej omówione zostały główne bloki zaprojektowanego i wykonanego przez autora środowiska testowego. Świadectwem pełnej funkcjonalności i dokładności ich działania są wykonane pomiary, które opisano w dwóch poprzednich rozdziałach.

5.1. Płytki testowa

Dedykowana płytka testowa (wraz z umieszczonymi na niej układami) pełni następujące funkcje:

- jest sztywnym stelażem mechanicznym zapewniającym ochronę testowanym strukturom testowym,
- dostarcza napięcia zasilające oraz sygnały polaryzujące do struktur testowych,
- konwertuje poziomy napięć / impedancji sygnałów wyjściowych oraz wejściowych,
- poza tym pełni funkcje specyficzne dla badanych struktur testowych (np. w przypadku elektroniki front-end musi zawierać układ wstrzykiwania ładunku).

Zwykle płytki testowe zawierają układy scalone w obudowie ceramicznej (ang. chip carrier) Omawiane w tej pracy struktury prototypowe nie posiadają obudowy ceramicznej, co podyktowane jest następującymi względami:

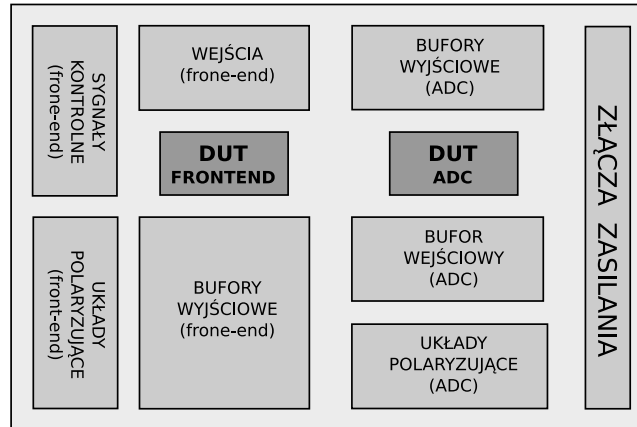
- konieczność minimalizacji ilości materiału jaki zostanie umieszczony w przestrzeni aktywnej docelowego detektora,
- konieczność minimalizacji pojemności pomiędzy padami układów scalonych (szczególnie pomiędzy polami kontaktowymi wejściowymi),
- konieczność minimalizacji zajmowanej powierzchni,
- minimalizacja kosztów.

Rezygnacja z obudów ceramicznych zwiększa wymagania stawiane płytce testowej. Zdecydowano się więc na wykonanie dedykowanej płytki drukowanej (ang: Printed Circuit Board – PCB) do badania prototypowych struktur testowych. Projekt płytki jest o tyle trudniejszy, iż zdecydowano się na wykonanie płytki poniekąd uniwersalnej, tzn. umożliwiającej badanie tak układów elektroniki front-end, jak również prototypów przetwornika ADC.

Projekt płytki drukowanej wykonano używając narzędzi projektowych dostępnych w ramach pakietu Altium Designer 6 [Alt06]. Do produkcji płytek wybrano technologię dostarczaną przez firmę Hatron, która wykonuje płytki z minimalnym rasterem (ang. pitch) pomiędzy ścieżkami na poziomie $150\ \mu\text{m}$ oraz zawierające do 12 warstw metalicznych. Przy wyborze producenta istotny był minimalny raster, gdyż prototypy zaprojektowanych ASIC-ów elektroniki front-end i ADC mają pady wyjściowe odległe od siebie o $100\ \mu\text{m}$.

Poglądowy obraz z rozmieszczeniem elementów na płytce drukowanej został zaprezentowany na rysunku 5.2. Wyróżnić na nim można następujące elementy:

- układy scalone elektroniki front-end i ADC,
- bufory na wyjściu kanałów odczytowych,
- układy polaryzacji dla obu struktur testowych,



Rysunek 5.2. Rozmieszczenie elementów na płytce PCB

- konwerter sygnału pojedynczego na różnicowy (na wejściu układu ADC),
- złącza umożliwiające podłączenie sensorów (do elektroniki front-end),
- gniazda zasilające,
- gniazda sygnałowe (analogowe jak i cyfrowe).

Pełną dokumentację schematów ideowych i planu masek płytki drukowanej umieszczono w załączniku A.

Na rysunkach 5.3 i 5.4 pokazano fotografie kompletnych płytek przygotowanych odpowiednio do testowania elektroniki front-end oraz przetworników ADC.

Poniżej omówione zostaną tylko kluczowe bloki oraz rozwiązania zaproponowane przez autora przy realizacji płytki testowej.

Bufory analogowe elektroniki front-end

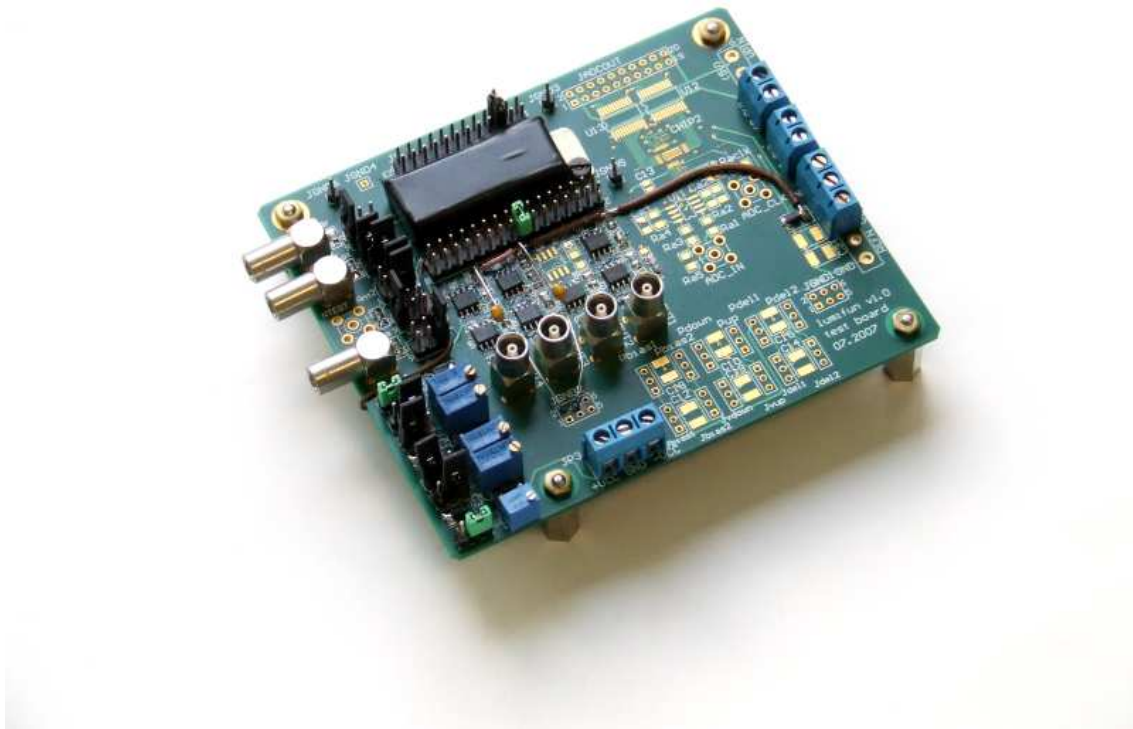
Ponieważ wyjścia kanałów front-end były projektowane z myślą o niewielkim obciążeniu pojemnościowym, konieczne było umieszczenie odpowiednich buforów. Schemat sekcji buforów przedstawiono na rysunku 5.5. Funkcja przenoszenia pierwszego stopnia dana jest zależnością:

$$V_1 = \left(1 + \frac{R_2}{R_1}\right) V_{in} - \frac{R_2}{R_1} V_{dc1} \quad (5.1)$$

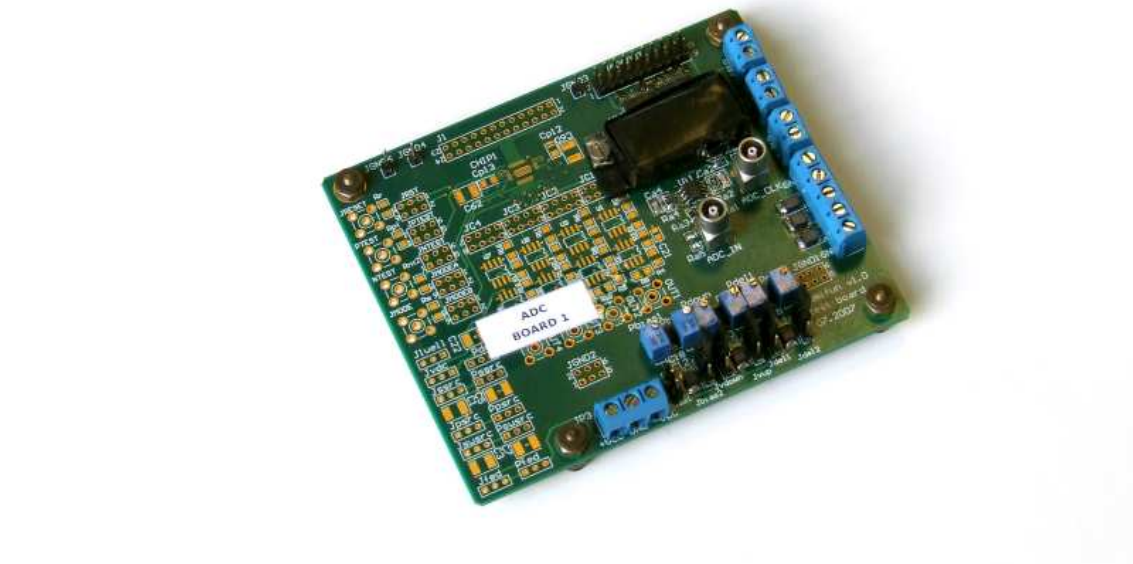
drugiego natomiast:

$$V_{out} = -\frac{R_4}{R_3} V_1 + \left(1 + \frac{R_4}{R_3}\right) V_{dc2} \quad (5.2)$$

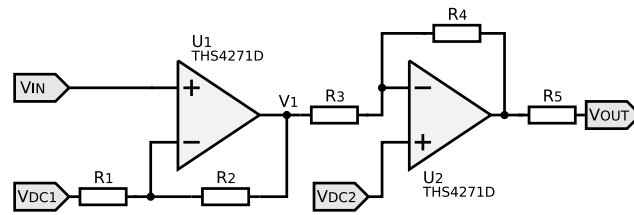
Aby wzmacniacze U1 oraz U2 pracowały stabilnie wzmocnienie różnicowe ustawiono na 2, co osiągnięto ustalając odpowiednio $R_1 = R_2$ oraz $R_3 = R_4$. Do układu dostarczane są dwa poziomy stałe V_{dc} , aby możliwe było zapewnienie odpowiednich punktów pracy wzmacniaczom operacyjnym oraz wartości napięcia stałego na wyjściu całego toru. Na wyjściu buforów dodano rezystor 50Ω zapewniający dopasowanie impedancji falowej przewodu transmisyjnego. Na płytce wykonano cztery identyczne



Rysunek 5.3. Fotografia płytki testowej z osadzonymi elementami do testowania elektroniki front-end



Rysunek 5.4. Fotografia płytki testowej z osadzonymi elementami do testowania struktur ADC

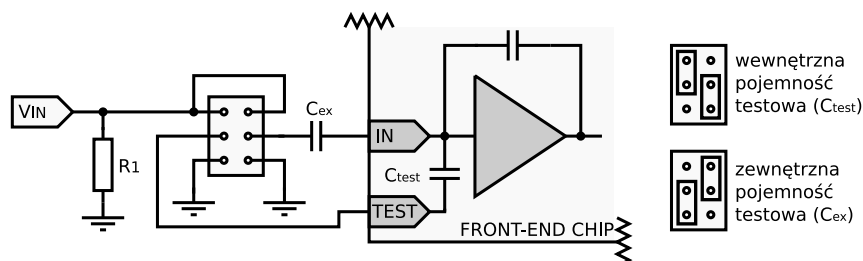


Rysunek 5.5. Bufory na wyjściu kanałów frontend

bufory odczytowe wraz z odpowiednimi zworkami umożliwiającymi podłączenie ich do odpowiednich kanałów elektroniki front-end.

Układ wstrzykiwania ładunku elektroniki front-end

Zaprojektowane struktury testowe prototypowego układu elektroniki front-end mają na wejściu każdego kanału pojemność testową C_{test} o wartości 0,5 pF (rys. 3.3). Jednak ze względu na możliwy do wstrzyknięcia ładunek (nie można podać skoku napięciowego większego niż napięcie zasilające) pojemność ta wykorzystywana jest tylko przy testach w trybie kalibracyjnym (duże wzmocnienie). Aby testować układy przy małym wzmocnieniu (gdzie szumy nie są już istotne) konieczne było umożliwienie wstrzykiwania większych ładunków. Żeby to zapewnić do wejść układu dołączono zewnętrzne pojemności testowe C_{ex} (rys. 5.6)

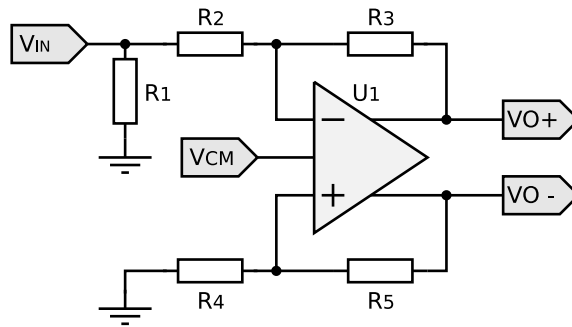


Rysunek 5.6. Układ wstrzykiwania ładunku wraz z proponowanymi kombinacjami zwerek

Zworki w pozycji górnej powodują, że sygnał testowy podawany jest na pojemność wewnętrzną C_{test} , a pojemności C_{ex} są podłączone do masy (symulując pojemności sensora). W pozycji dolnej ładunek wstrzykiwany jest przez pojemności zewnętrzne C_{ex} . Niezależnie od konfiguracji zwerek, na wejście układu podpięta jest sumaryczna pojemność $C_{ex} + C_{test}$ którą należy uwzględnić przy opracowywaniu wyników pomiarów. Przy założeniu, iż wejściowy impuls na pojemności C ma kształt skoku o amplitudzie A , wyindukowany impuls prądowy ma kształt delty Diraca oraz niesie ładunek $A \cdot C$.

Bufory wejściowe ADC

Zaprojektowane ADC ma architekturę w pełni różnicową. Takie rozwiązanie implikuje konieczność dostarczenia sygnału wejściowego również w postaci różnicowej, przy zachowaniu odpowiedniego poziomu stałego (ang. common mode). Większość generatorów dostępnych na rynku oferuje wyjścia pojedyncze (ang. single ended) co wymusiło konieczność zaprojektowania odpowiedniego konwertera. Jego schemat został przedstawiony na rysunku 5.7.



Rysunek 5.7. Konwerter sygnału pojedynczego na różnicowy

Funkcja przenoszenia sygnału różnicowego takiego układu, przy założeniu, iż $R_3/R_2 = R_5/R_4$, wynosi:

$$V_{o+} - V_{o-} = -\frac{R_3}{R_2} V_{in} \quad (5.3)$$

Sygnały wejściowe są symetryczne względem napięcia V_{CM} (w omawianym układzie około 1,6 V) Jako wzmacniacz zastosowano monolityczny wzmacniacz operacyjny THS4505 produkowany przez firmę Analog Devices [Tex08]. Ze względu na stabilność wzmacniacz pracuje ze wzmocnieniem równym 2.

Bufory wyjściowe ADC

Ze względu na ograniczenie zakłóceń w układzie przetwornika ADC, wewnątrz struktury testowej zrezygnowano z umieszczania wydajnych prądowo wyjść cyfrowych. Takie rozwiązanie wymusiło zastosowanie buforów cyfrowych na płycie. Zdecydowano się na układy SN74LVC245 firmy Texas Instruments [Tex04]. Układy mogą pracować z różnymi standardami napięć po stronie wejść i po stronie wyjść. Projekt taki pozwala podłączyć wyjścia do różnych systemów akwizycji danych, pracujących w standardach napięciowych od 1,8 do prawie 6 V.

Układy polaryzacji

Jak wspomniano w poprzednim rozdziale, w celu zapewnienia pełnej testowalności układów zdecydowano się na wyprowadzenie większości sygnałów polaryzujących na zewnątrz struktur testowych. Aby poprawnie spolaryzować wszystkie bloki wewnętrzne konieczne było dostarczanie prądów oraz napięć referencyjnych. Rozwią-

zanie takie zapewnia możliwość ustawienia wyżej wymienionych wartości na stałe (za pomocą potencjometrów), jak również podpięcie do zewnętrznych źródeł referencyjnych.

Wszystkie gniazda zostały umieszczone na obrzeżach płytki, co zdecydowanie wpływa na komfort pracy. Podczas projektowania płytki zwrócono również uwagę na odpowiednie doprowadzenie masy oraz napięcia zasilającego do struktur testowych jak i układów buforujących i polaryzujących. Do tego celu wykorzystano dedykowane płaszczyzny masy oraz zasilania. Takie rozwiązanie wymusiło wykonanie płytki czterowarstwowej. Ze względu na odsprzęganie niepożądanych zakłóceń, we wszystkich kluczowych węzłach dodano kondensatory odsprzęgające (ang. decoupling) o wartościach z przedziału 100 pF - 10 μ F.

5.2. Szybki i wysoko-rozdzielczy system próbkowania sygnałów analogowych

Aby mieć możliwość testowania omawianych wcześniej układów elektroniki front-end, z kluczowanym sygnałem resetu, konieczny jest wysoko-rozdzielczy przetwornik analogowo-cyfrowy. Symulowane szумы układów front-end były na poziomie $V_{rms} \leq 1,1$ mV, a dynamika sygnału wynosiła prawie 2 V. Chcąc dokonywać analizy widmowej szumów (co jest niemożliwe przy użyciu klasycznego analizatora widma ze względu na kluczowanie) od przetwornika analogowo cyfrowego wymagamy, aby jego najmniej znaczący bit był na poziomie kilkudziesięciu – kilkuset mV. Wymaganie takie przy 2 V zakresie dynamicznym wskazuje, iż przetwornik powinien charakteryzować się conajmniej 12 bitową rozdzielczością ($LSB = 0,5$ mV). Chcąc analizować szумы w dziedzinie czasu, dla danego sygnału, wymagany jest pomiar amplitudy w kilku chwilach czasowych. Ponieważ stałe czasowe rozważanych impulsów są na poziomie 200 ns przetwornik powinien być w stanie próbować z częstotliwością conajmniej 10 MHz.

Chcąc spełnić powyższe wymagania zdecydowano się na użycie monolitycznego przetwornika analogowo-cyfrowego AD9445 dostarczanego przez firmę Analog Devices [Ana05]. Układ ten charakteryzuje się 14 bitową rozdzielczością, częstotliwością próbkowania 125 MSps oraz wyjściami równoległymi w standardzie CMOS lub LVDS. Ze względów ekonomicznych, do wykonania prototypu, zdecydowano się wykorzystać płytkę ewaluacyjną dostarczaną przez producenta [Ana05]. Płytką ta zawiera wszystkie układy peryferyjne niezbędne do funkcjonowania przetwornika, takie jak: stabilizatory napięcia oraz konwertery poziomów napięciowych. Jedynym mankamentem omawianego zestawu jest to, iż jego wejścia są sprzężone zmienno-prądowo z wejściami przetwornika. Częstość graniczną filtra oszacowano na około 1 MHz, co zdecydowanie uniemożliwia wykonywanie interesujących nas pomiarów. Aby pozbyć się tej niedogodności zdecydowano się odłączyć filtr dostarczony przez producenta. Ponieważ ADC ma wejście różnicowe, dodatkowo wykonano konwerter sygnału pojedynczego na różnicowy, wykorzystując w pełni różnicowy wzmacniacz



Rysunek 5.8. Fotografia zestawu ewaluacyjnego układu przetwornika analogowo-cyfrowego AD9445 wraz konwerterami

operacyjny. Układ został wykonany identycznie jak ten zaprezentowany na rysunku 5.7. Fotografię pokazującą płytkę testową przetwornika wraz z konwerterem zaprezentowano na rysunku 5.8.

5.3. Szybki system akwizycji danych cyfrowych

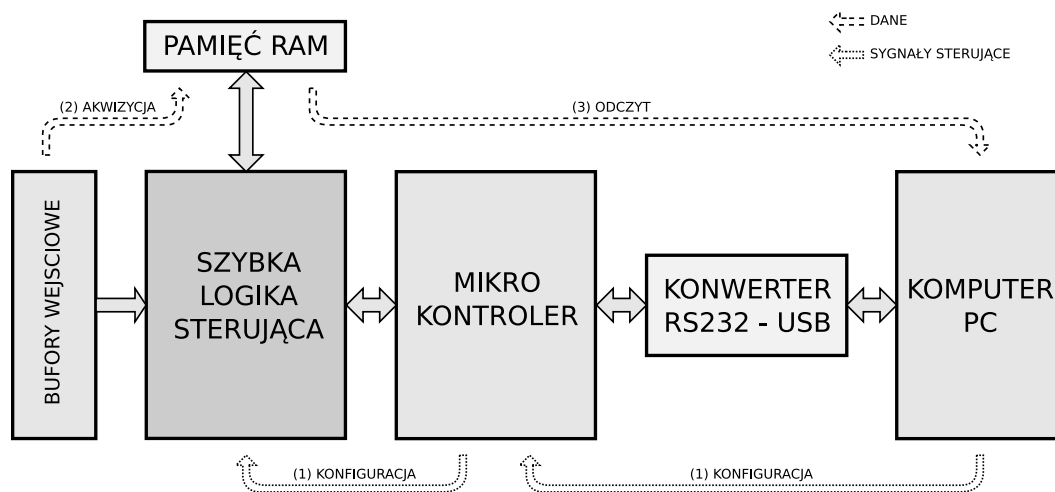
Zarówno podczas pomiarów elektroniki front-end, przy użyciu wyżej opisanego komercyjnego przetwornika ADC, jak i podczas testowania prototypowych struktur zaprojektowanego ADC, kluczowym elementem jest analizator stanów logicznych. Analizator taki powinien być w stanie odczytywać magistralę cyfrową z częstotliwością wynoszącą conajmniej tyle, ile projektowana maksymalna częstotliwość pracy układu. W pierwszej fazie testów używano do tego celu komercyjnej szybkiej karty na magistrali PCI, sprzedawanej przez National Instrument. Niestety z jej pomocą można było tylko przeprowadzić wstępne testy, gdyż umożliwiała ona osiągnięcie częstotliwości próbkowania niewiele ponad 10 MHz. Ze tego powodu konieczne oka-

zało się zaprojektowanie i wykonanie dedykowanego analizatora stanów logicznych samodzielnie.

Wymagania jakie zostały postawione projektowanemu urządzeniu to:

- częstotliwość próbkowania magistrali cyfrowej conajmniej 50 MHz,
- praca z zewnętrznym zegarem próbkującym,
- szerokość magistrali conajmniej 16 bitów,
- szeregowy interfejs komunikacyjny z komputerem klasy PC (najlepiej USB),
- biblioteka umożliwiająca obsługę sprzętu z poziomu języka C++ działająca w systemie Linux oraz Windows.

Ponieważ żaden z obecnie używanych standardów komunikacji szeregowej nie udostępnia tak wysokich przepustowości, konieczne było zbudowanie systemu zdolnego do zbuforowania dużej ilości danych i retransmitowania ich „offline”. Ze względu na efektywność projektowania oraz możliwość uzyskania bardzo dobrych parametrów zdecydowano się na realizację dedykowanego układu akwizycji danych przy wykorzystaniu programowalnych układów logicznych FPGA (ang. Field-Programmable Gate Array). W najprostszym rozumieniu uniwersalne układy logiczne FPGA (składające się z tysięcy czy nawet milionów bramek logicznych) zapewniają „online” realizację sprzętową funkcji logicznych zapisanych w języku wysokiego poziomu (VHDL, Verilog) lub w postaci schematów ideowych. Szczegółowe omówienie koncepcji FPGA można znaleźć np. w [RCN03, ZP04, Zby07]. Schemat blokowy zaproponowanej architektury został zaprezentowany na rysunku 5.9.



Rysunek 5.9. Schemat blokowy systemu akwizycji danych wraz z oznaczonym przepływem danych

Jak można zauważyć całość systemu podzielono na kilka bloków funkcjonalnych:

- szybka logika; zadaniem tej części jest odczytywanie wejść i zapisywanie ich stanu do pamięci RAM (ang. Random Access Memory), jak również pośredniczenie podczas odczytu pamięci,

- mikrokontroler; jego zadaniem jest komunikacja z komputerem oraz sterowanie logiką,
- konwerter sygnału RS232 na USB; ponieważ użyty zestaw ewaluacyjny FPGA nie zawiera interfejsu USB konieczne było wykonanie go samodzielnie.

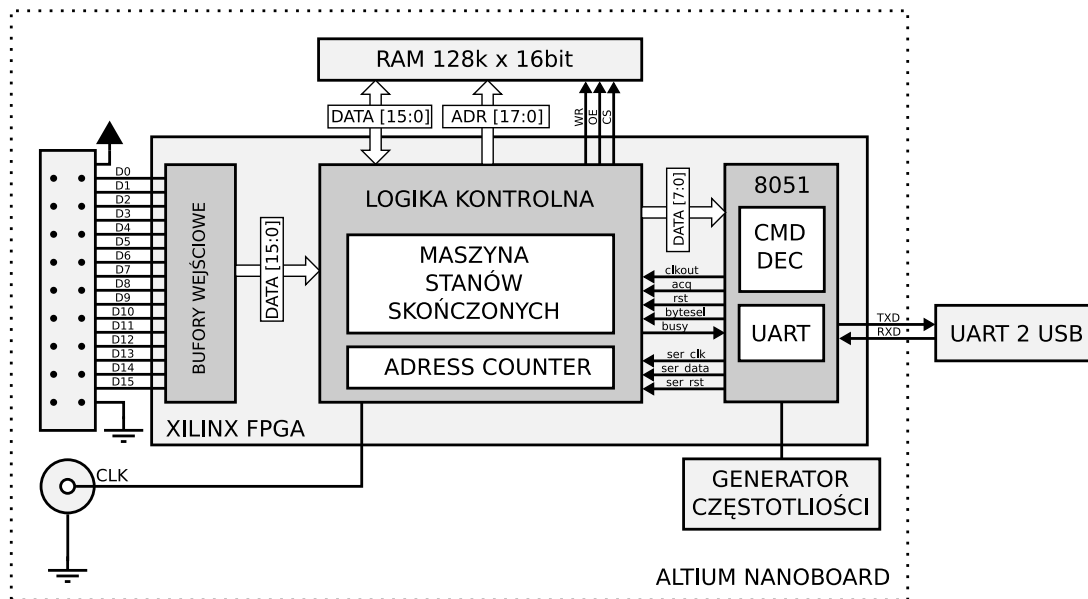
Praca takiego systemu będzie przebiegała kilkietapowo. W pierwszym etapie zostaną skonfigurowane wszystkie podsystemy. Następnie w pamięci statycznej zostaną zgromadzone dane. W ostatnim kroku dane te zostaną przesłane do komputera.

Do wykonania prototypu wykorzystano uniwersalny zestaw ewaluacyjny Nano-Board dostarczany przez firmę Altium. Zestaw ten zawiera wszystko co niezbędne do poprawnego funkcjonowania układu FPGA (tj. pamięć konfiguracyjna, zasilacze, generatory zegara) oraz elementy peryferyjne (tj. wyświetlacze, klawiatury, przetworniki ADC / DAC do zastosowań audio, złącza rozszerzeń) [Alt04]. Płytkę ewaluacyjną jest przystosowana do pracy z układami FPGA wielu producentów (np. Xilinx, Altera). W wersji podstawowej omawianego zestawu dostępne są układy Spartan XC2S300E-6PQ208C i Cyclone EP1C12Q240C8. Po studiach porównawczych wybrano układ Spartan dostarczany przez firmę Xilinx. Jako środowisko projektowe wybrano pakiet Altium Designer 6 dostarczany przez firmę Altium, który natywnie współpracuje z omawianym powyżej zestawem ewaluacyjnym. Pakiet ten korzysta z zestawu narzędzi projektowych dostarczanych przez producentów danych układów FPGA (w tym przypadku Xilinx). Szczegółowy schemat blokowy pokazujący realizację systemu akwizycji danych został zaprezentowany na rysunku 5.10. Zaprojektowany i zrealizowany system, którego podstawowe bloki zostaną omówione poniżej, całkowicie wypełnił stawiane przed nim zadania. W szczególności umożliwił akwizycję 16 bitowej magistrali danych z częstotliwością zdecydowanie przewyższającą wymagane 40 MSps. Modularna architektura i przewyższająca stawiane wymagania funkcjonalność pozwoli zaadaptować go także do przyszłych potrzeb pomiarowych.

5.3.1. Szybka logika sterująca

Ze względu na wymagania częstotliwościowe stawiane przed układem akwizycji danych, zdecydowano się na wykonanie dedykowanego bloku logicznego, który będzie w stanie odczytywać wartości linii wejściowych i wpisywać je do pamięci statycznej. Aby zapewnić przenośność, pomiędzy różnymi układami FPGA, zdecydowano się na opisanie funkcjonalności tego bloku w języku wysokiego poziomu Verilog, umożliwiającym modelowanie oraz symulowanie różnego typu układów, między innymi logicznych.

Logika może pracować w jednym z dwóch trybów: akwizycji lub odczytu danych. Wybór trybu pracy odbywa się przez podanie odpowiedniego stanu logicznego na wejściu `acq`. Głównym blokiem funkcyjnym jest licznik generujący kolejne adresy dla pamięci (`ram_adr`). Zegar do licznika jest multipleksowany pomiędzy zegarem części akwizycji danych `clk_in` a zegarem generowanym przez mikrokontroler `clk_out`. Takie rozwiązanie pozwala na zapis danych z częstotliwością daną przez badany układ, a odczyt z częstotliwością z jaką możliwa jest transmisja danych. Dodatkową



Rysunek 5.10. Schemat realizacji systemu akwizycji danych

funkcją realizowaną przez ten układ jest informacja mikrokontrolera o zakończeniu procesu akwizycji danych, po zapisaniu odpowiedniej liczby próbek danych wejściowych (zmiana stanu wyjścia busy). Ilość danych jakie mają być zgromadzone pamiętana jest w rejestrze który można zapisywać przy użyciu magistrali szeregowej. Lista wszystkich wyprowadzeń modułu logiki, wraz z krótkim opisem funkcyjnym, została zaprezentowana w tabeli 5.1.

W obecnej implementacji układ posiada 16 bitową magistralę danych oraz 18 bitową magistralę adresową. Moduł jest w pełni sparametryzowany, więc nic nie stoi na przeszkodzie, aby dowolnie modyfikować szerokości poszczególnych szyn. Aby zmniejszyć ilość linii mikrokontrolera koniecznych do odczytu danych, zdecydowano się na zaszczyt multiplexera w strukturze bloku logiki. Multiplexer ten umożliwia wybór starszego lub młodszego bajtu z zapisanego słowa danych. Wejście adresowe tego multiplexera to `bytesel`.

Pełny kod opisujący logikę sterującą, napisany w języku VERILOG, dołączony został w załączniku B pracy.

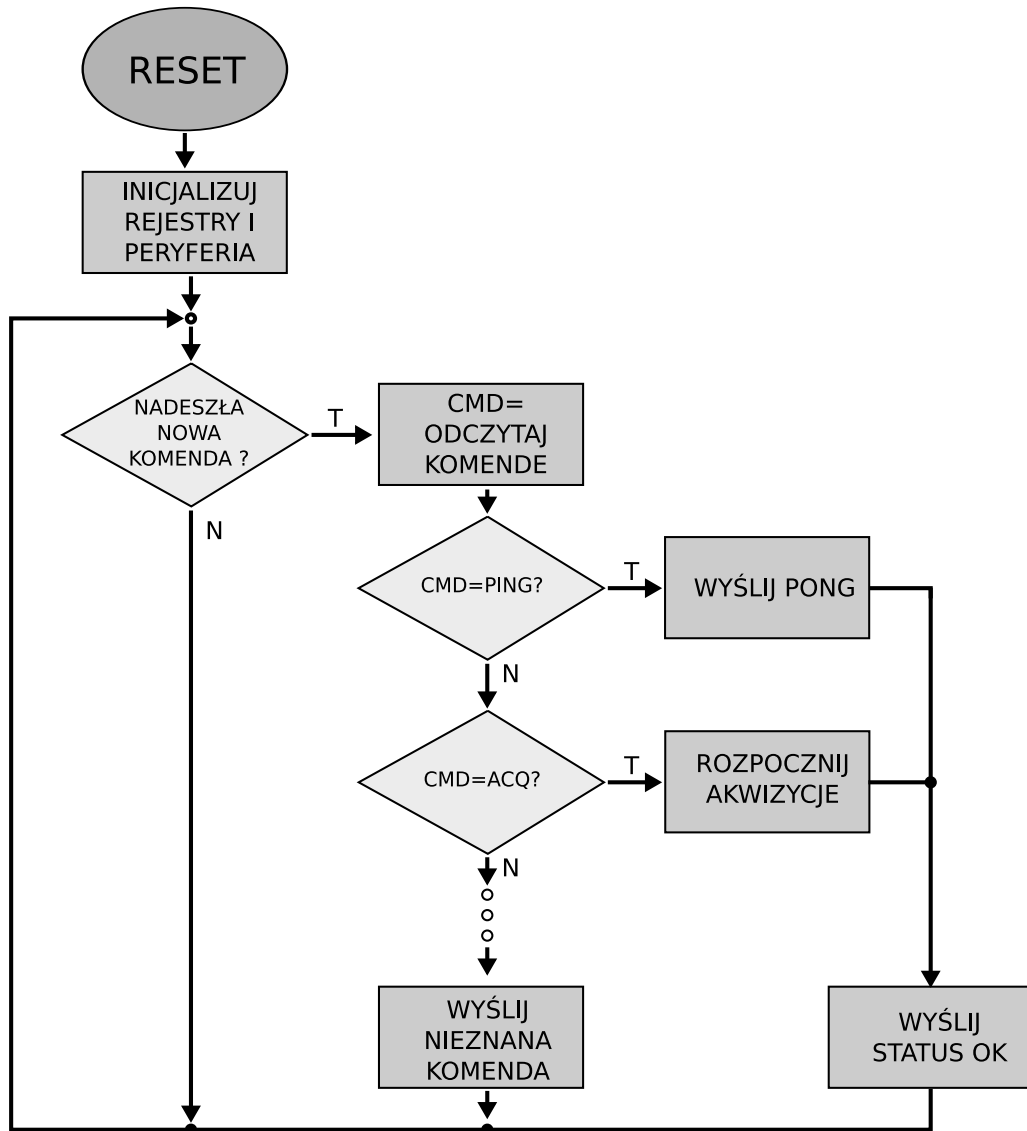
5.3.2. Mikrokontroler

W omawianym projekcie zdecydowano się na użycie mikrokontrolera „wbudowanego” (ang. *embedded*) w strukturę FPGA. Takie rozwiązanie jest bardzo elastyczne i umożliwia dokonywanie zmian w sposobie komunikacji czy sterowania, bez konieczności syntezy i przebudowywania całości struktury, czy tym bardziej wprowadzania zmian sprzętowych. Ponieważ czas wykonania poszczególnych zadań realizowanych przez mikrokontroler nie jest krytyczny, zdecydowano się na użycie

Tablica 5.1. Spis pinów układu szybkiej logiki sterującej

nazwa	typ	lb. bitów	funkcja
clkin	wejście	1	zewnętrzny zegar taktujący zapis do pamięci
datain	wejście	16	magistrala danych wejściowych
dataA	wejście/wyjście	8	8 starszych bitów magistrali danych podpiętych do pamięci RAM
dataB	wejście/wyjście	8	8 młodszych bitów magistrali danych podpiętych do pamięci RAM
ram_adr	wyjście	18	adres pamięci RAM
ram_we	wyjście	1	sygnał WriteEnable pamięci RAM
ram_re	wyjście	1	sygnał ReadEnable pamięci RAM
ram_ce	wyjście	1	sygnał ChipEnable pamięci RAM
reset	wejście	1	sygnał resetujący liczniki
acq	wejście	1	wymuszenie startu akwizycji danych
busy	wyjście	1	flaga określająca stan układu
ser_data	wejście	1	dane rejestr przesuwne
ser_clk	wejście	1	zegar rejestru przesuwne
ser_rst	wejście	1	reset rejestru przesuwne
clkout	wyjście	1	zegar taktujący odczyt z pamięci
dataout	wejście	8	dane wyczytywane z pamięci
bytesel	wejście	8	wybór starszego/młodsze bajtu danych

8 bitowego mikrokontrolera zgodnego z zestawem instrukcji 8051. Mikrokontroler ten jest dość prosty, dzięki czemu zajmuje mało zasobów logicznych układu FPGA. Mimo swej prostoty oferuje projektantowi sprzętowy interfejs UART oraz uniwersalne timery.



Rysunek 5.11. Algorytm wykonywania programu w mikrokontrolerze

Program sterujący jego pracą został napisany w C i skompilowany kompilatorem dostarczonym przez firmę ALTIUM. Mikrokontroler wykonuje nieskończoną pętlę, w której ciągle sprawdza czy nie nadeszły nowe rozkazy przez port szeregowy. Po zdekodowaniu komendy wykonuje odpowiednie czynności, zwracając komputerowi

(o ile to potrzebne) dane oraz komunikat o statusie zakończenia operacji. Schematycznie zostało to przedstawione na rysunku 5.11

Dodatkową funkcjonalnością jest kontrola poprawności działania części logiki, zapewniona przez generację sygnału timeout, w razie gdy znajdzie taka potrzeba. Najprostszą sytuacją jaką można sobie wyobrazić jest brak sygnału zegarowego na wejściu logiki akwizycyjnej. Sytuacja taka prowadziłaby do zawieszenia systemu. W przypadku gdy część logiki nie zgłasza zakończenia pracy przez dłużej niż określony czas (np. 1 sekunda), mikroprocesor podejmuje decyzję o siłowym zakończeniu pomiaru, informując o tym użytkownika.

Pełny kod programu sterującego pracą mikrokontrolera, napisany w języku C, dołączony został w załączniku C pracy.

5.3.3. Komunikacja

Komunikacja z komputerem klasy PC odbywa się za pośrednictwem interfejsu RS232. Transmisja odbywa się z prędkością 115,2 kbps w formacie 8 bitów danych, 1 bit stopu oraz bez bitu parzystości. Aby umożliwić podłączanie układu do nowszych komputerów wykonano konwerter RS232–USB, bazujący na układzie scalonym FT232BM, pracującym w standardowej konfiguracji [Fut05]. Konwerter ten widziany jest jako wirtualny port szeregowy, a jego obsługa jest identyczna jak rzeczywistego portu szeregowego.

Transmisja jest zorganizowana w trybie master-slave. Tylko komputer PC pracujący jako urządzenie nadrzędne może inicjować transmisję, wydając komendy układowi. Wszystkie komendy są bezkontekstowe. W celu minimalizacji ilości przesyłanych danych cała transmisja ma charakter binarny. Urządzenie posiada następującą listę komend:

komenda PING o kodzie 'p', komenda nie powoduje żadnej akcji. Jeśli układ zdekoduje tą komendę zwraca tylko 'STATUS_OK'.

komenda SET_LENGTH o kodzie 'l', jako parametr przyjmuje czterobitową liczbę bez znaku, określającą długość rekordu jaki ma zostać zapisany w pamięci podczas procesu akwizycji danych. Zwraca 'STATUS_OK' w wypadku poprawnego zdekodowania parametru, w wypadku braku parametru w czasie około sekundy, zwraca 'STATUS_TIMEOUT'.

komenda GET_LENGTH o kodzie 'L', zwraca czterobitową liczbę bez znaku określającą obecnie ustawioną długość rekordu jaki ma zostać zapisany w pamięci podczas procesu akwizycji danych. Dodatkowo zwraca 'STATUS_OK'.

komenda SET_TIMEOUT o kodzie 't', jako parametr przyjmuje czterobitową liczbę bez znaku określającą długość czasu (podaną w ms), jaki układ ma czekać podczas procesu akwizycji, zanim zostanie zgłoszony timeout. Zwraca 'STATUS_OK'

w wypadku poprawnego zdekodowania parametru, w wypadku braku parametru w czasie około sekundy zwraca 'STATUS_TIMEOUT'.

komenda GET_TIMEOUT o kodzie 'T', zwraca czterobitową liczbę bez znaku określającą obecnie ustawioną długość czasu (podaną w ms), jaki układ ma czekać podczas procesu akwizycji, zanim zostanie zgłoszony timeout.

komenda ACQUIRE o kodzie 'a', rozpoczyna proces akwizycji danych. Zwraca 'STATUS_OK' jeśli w czasie określonym komendą SET_TIMEOUT zostanie zebrany rekord danych o długości ustawionej komendą GET_LENGTH. W przeciwnym wypadku zwraca 'STATUS_TIMEOUT'.

komenda READ o kodzie 'r', komenda odczytuje zawartość pamięci danych. Zwraca kolejne próbki sygnałów wejściowych zakodowane na dwóch bajtach (ilość danych wynosi tyle, ile aktualnie ustawiona komendą GET_LENGTH długość) oraz 'STATUS_OK'.

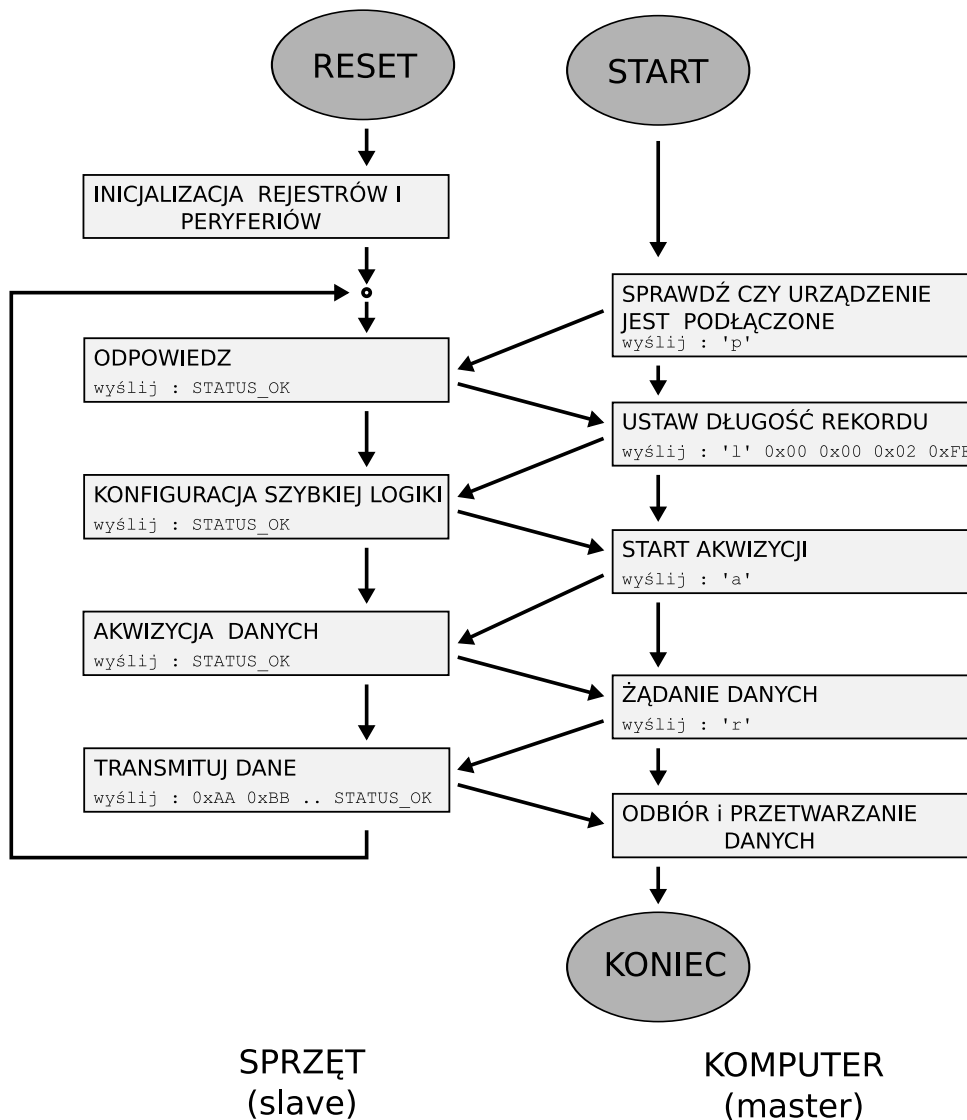
Ponieważ testy pokazały, że transmisja przebiega bezbłędnie, więc nie zdecydowano się zastosować algorytmów sum kontrolnych i retransmisji.

Wymiana komunikatów dla przykładowej sesji pomiarowej została zademonstrowana na rysunku 5.12. Po zainicjalizowaniu obu urządzeń i sprawdzeniu przez komputer czy urządzenie jest obecne, komputer wysyła parametry konfigurujące proces akwizycji danych. w kolejnym kroku wysyłany jest rozkaz rozpoczęcia procesu zbierania danych. Po zakończeniu tego procesu urządzenie odpowiada, zwracając komunikat o tym czy operacja zakończyła się sukcesem czy nie. W wypadku sukcesu, komputer nakazuje urządzeniu transmisję danych. W kolejnym kroku komputer zajmuje się przetwarzaniem danych podczas gdy urządzenie jest gotowe do przyjęcia kolejnych komunikatów.

5.4. Oprogramowanie

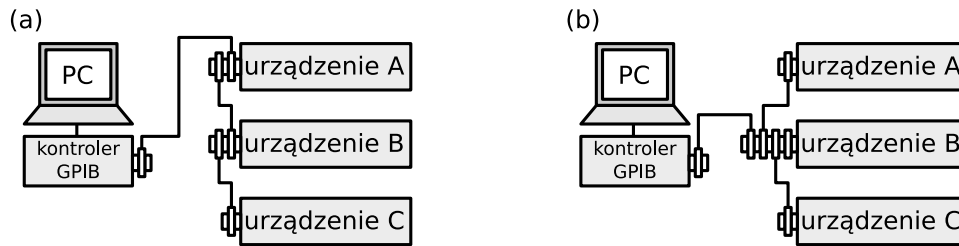
Wiele pomiarów jest bardzo czasochłonnych. Niejednokrotnie w sekwencjach pomiarowych zmienia się tylko jeden z parametrów, jak np. prąd polaryzujący, napięcie zasilające czy amplituda impulsu wejściowego. W celu optymalizacji procesu pomiarowego zdecydowano się stworzyć infrastrukturę umożliwiającą sterowanie wszystkich urządzeń z poziomu dedykowanych programów napisanych w języku C++. Ponieważ większość dostępnych urządzeń wyposażonych jest w interfejs GPIB (ang. General Purpose Interface Bus) zdecydowano się na sterowanie przy użyciu tegoż interfejsu.

Standard GPIB definiuje warstwę fizyczną (kable, złącza) jak również protokół komunikacji. Główne cechy warstwy fizycznej to:



Rysunek 5.12. Schemat przepływu danych w systemie akwizycji danych

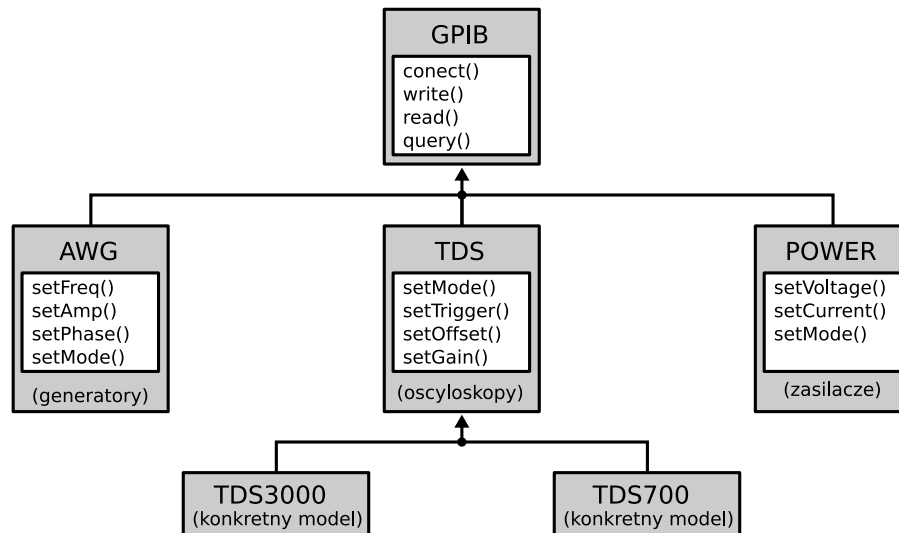
- 24 żyłowy kabel transmisyjny. 8 przewodów to linie danych DIO1-DIO8 (GPIB jest interfejsem równoległym). Kolejne 8 żył steruje transmisją. Pozostałe 8 żył to linie masy,
- urządzenia podłącza się na szynie równolegle,
- maksymalna długość sieci nie powinna przekroczyć 20 metrów,
- maksymalna odległość pomiędzy dwoma urządzeniami nie powinna być większa niż dwa metry,
- sieć można mieć strukturę gwiazdową lub liniową (rysunek 5.13),
- każde urządzenie podłączone do szyny może być typu *listener* lub *talker* (może również spełniać obie funkcje),



Rysunek 5.13. Topologia sieci GPIB (a) liniowa (b) gwiazdista

— wymagane jest aby na szynie znajdowało się urządzenie typu *controller*, funkcję tą najczęściej pełnią karty rozszerzeń do komputera PC.

Na pewnym etapie rozwoju standardu rozbudowano również definicję protokołu komunikacyjnego, specyfikując podstawowe komendy jakie urządzenie musi być w stanie wykonywać. Zdefiniowany jest również protokół o informowaniu o błędach.



Rysunek 5.14. Schematyczny diagram klas dla urządzeń wykorzystujących interfejs GPIB

Analiza instrukcji obsługi oraz manuali programistycznych wielu stosowanych urządzeń pokazała, iż zestawy komend dla poszczególnych typów urządzeń (np. generator) są bardzo podobne. Korzystając z tego faktu zdecydowano się na zaproponowanie zestawu obiektów będących warstwą abstrakcji nad niskopoziomym interfejsem GPIB. Schematyczny diagram zaproponowanych klas zaprezentowano na rysunku 5.14.

Zadaniem klasy GPIB jest obudowanie niskopoziomowej biblioteki obsługi magistrali oraz dostarczenie wygodnych w użyciu komend umożliwiających komunikację z urządzeniem. Jako warstwę najniższą użyto bibliotekę dostarczaną przez firmę Na-

tional Instruments oraz darmową bibliotekę `libgpib`. Rozwiązanie takie zapewnia poprawną pracę pod większością z dostępnych systemów operacyjnych (testowano na Windows i kilku dystrybucjach Linuksa).

Kolejno z klasy `GPIB` dziedziczą klasy ogólne dla danych sprzętów typu: oscyloskop `TDS` czy generator `AWG`. Zawierają one implementację funkcji dostarczanych przez większość z urządzeń danego typu. W wypadku generatora może to być wybór generowanego przebiegu, jego amplituda czy częstotliwość. Klasy ogólne są następnie uszczegóławiane, tworząc klasy poszczególnych obiektów zawierające funkcje specyficzne dla danego urządzenia (w wypadku generatora może to być na przykład marker). Architektura taka umożliwia stosowanie w mniej wymagających aplikacjach klas ogólnych, co uniezależnia pomiar od sprzętu na jakim jest wykonywany.

Wyciąg z kodu najważniejszych klas umożliwiających sterowanie urządzeń poprzez interfejs `GPIB`, napisany w języku `C++`, dołączony został w załączniku D pracy.

Rozdział 6

Podsumowanie

Celem niniejszej pracy był rozwój elektroniki front-end dla odczytu krzemowych sensorów półprzewodnikowych detektora świetlności LumiCal, przy przyszłym eksperymencie ILC. W ramach prowadzonych badań postawiono sobie trzy nadrzędne cele:

- zaprojektowanie, wykonanie masek technologicznych oraz przetestowanie prototypowego układu przedwzmacniacza o kształtowaniu nieciągłym w czasie,
- opracowanie metodologii testowania przetworników analogowo-cyfrowych oraz wykonanie pomiarów prototypowego przetwornika ADC,
- zaprojektowanie i wykonanie szybkiego układu akwizycji danych bazującego na układach FPGA.

W ramach prac poświęconych pierwszemu celowi wykonano co następuje:

- przeprowadzono analizę wymagań stawianych przedwzmacniaczowi i na jej podstawie wybrano architekturę integratora z kluczowanym sygnałem reset,
- przy użyciu pakietu Cadence zaprojektowano przedwzmacniacz, wykorzystując architekturę wzmacniacza transimpedancyjnego, pracującego w konfiguracji zamkniętej kaskody. Dla zapewnienia dużej dynamiki ładunków wejściowych zaimplementowano możliwość przełączania wzmocnienia (np. 3,4 mV/fC, 0,19 mV/fC),
- wykonano plan masek topologicznych układu, przystosowując wymiary pojedynczego kanału ($100\mu\text{m} \times 370\mu\text{m}$) do umieszczenia w systemie wielokanałowym,
- gotowy projekt układu wysłano do produkcji w technologii AMS 0,35 μm ,
- przygotowano stanowisko pomiarowe umożliwiające testowanie struktur prototypowych,
- wykonano pomiary funkcjonalne działania układu, które potwierdziły poprawność projektu,
- wykonano ilościowe pomiary wzmocnienia. Ich wyniki posłużyły do zweryfikowania poprawności i dokładności symulacji,
- zaproponowano użycie techniki skorelowanego próbkowania podwójnego CDS do pomiaru, a także filtracji szumów. Rezultaty tych pomiarów pokazały, że dla pełnego zakresu pojemności wejściowej sensora można uzyskać stosunek $S/N > 10$.

Podsumowując można stwierdzić, że wykonany projekt przedwzmacniacza okazał się bardzo pożyteczny dla dalszego rozwoju układów elektroniki front-end z nieciągłym kształtowaniem w czasie. W szczególności wykazał, iż technika CDS powinna zostać zaimplementowana w kolejnych wersjach układu.

W ramach pomiarów prototypowego przetwornika analogowo-cyfrowego wykonano następujące prace:

- opracowano metodologię pomiarów przetworników analogowo-cyfrowych, a w szczególności ich parametrów statycznych i dynamicznych. W przypadku pomiarów statycznych użyto metody histogramowej, a w przypadku dynamicznych oparto się na metodzie korzystającej z dyskretnej transformaty Fouriera,
- przygotowano stanowisko pomiarowe umożliwiające testowanie prototypowych struktur ADC,
- wykonano pomiary statyczne ADC. Wykazały one, iż nieliniowość całkowita (INL) badanych układów mieści się w przedziale $\pm 3LSB$, podczas gdy nieliniowość różniczkowa (DNL) dla większości kodów pozostaje w przedziale $\pm 0,5LSB$. Sześć kodów, dla których DNL wynosił mniej niż $-0,9$, zostało zidentyfikowanych jako kody brakujące,
- wykonano pomiary dynamiczne ADC. Wykazały one, iż stosunek sygnału do szumu, nie będącego zniekształceniami harmonicznymi, wynosi 49,1 dB, co jest wartością bardzo bliską przewidywanej przez teorię. Zniekształcenia harmoniczne zmierzono na poziomie $-40,9$ dB.

Dzięki wykonanym pomiarom ADC udało się wskazać słabszy punkt projektu, a mianowicie niedokładność układu mnożącego przez dwa, powodujący powstawanie nieliniowości i zniekształceń harmonicznych. Informacje te zostaną wykorzystane przy projekcie kolejnej wersji układu.

W ramach prac nad dedykowanym systemem akwizycji danych zrealizowano następujące zadania:

- zaproponowano technologię programowalnych układów logicznych FPGA jako optymalną do wykonania projektu,
- zaproponowano architekturę systemu DAQ. Dla zapewnienia odpowiednio wysokiej częstotliwości pracy wybrano koncepcję z lokalnym buforowaniem danych (256 kB),
- wykonano projekt układu akwizycji danych bazujący na zestawie ewaluacyjnym NanoBoard firmy Altium oraz układzie FPGA Spartan firmy Xilinx,
- stworzono oprogramowanie sterujące systemem akwizycji danych, a w szczególności kody programów w języku C oraz modele działania układów logicznych w języku opisu sprzętu Verilog,
- jako przystawkę do systemu akwizycji danych cyfrowych wykonano moduł do akwizycji danych analogowych z przetwornikiem analogowo-cyfrowym AD9445. Moduł ten charakteryzuje rozdzielczość 14 bitów i możliwość pracy przy częstotliwości próbkowania wynoszącej 125 MSps,

— uruchomiono pełny system akwizycji danych charakteryzujący się 16 bitową magistralą I/O oraz pracujący poprawnie dla częstotliwości dochodzących do 120 MHz.

Zaproponowany system akwizycji danych był niezbędnym narzędziem przy wykonywaniu pomiarów opisanych w rozdziale trzecim i czwartym. Modułarna architektura i bardzo szeroka funkcjonalność pozwoli zaadaptować go także do przyszłych potrzeb pomiarowych.

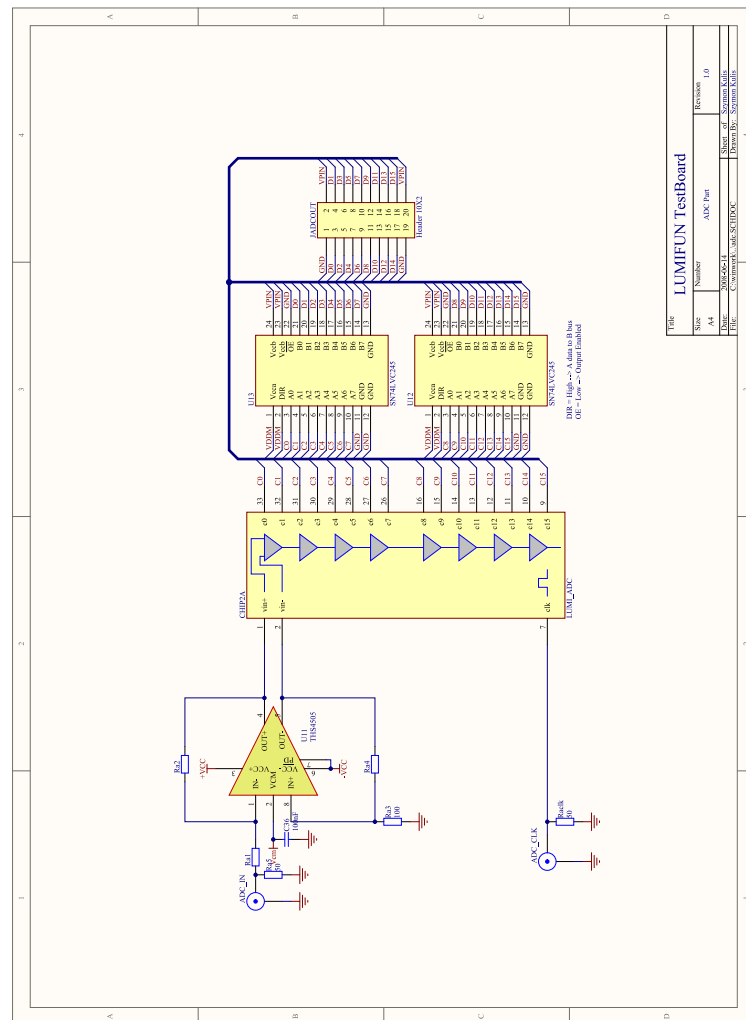
Autor niniejszego opracowania wykonał w całości (lub prawie) wszystkie prace opisane w rozdziałach trzecim, czwartym i piątym. Zawartość tych rozdziałów opisuje realizację podanych powyżej celów pracy. Autor uczestniczył również czynnie w ogólnych pracach koncepcyjnych nad detektorem LumiCal, zaprezentowanych w rozdziale drugim.

Potwierdzeniem udziału piszącego i użyteczności wykonanych prac jest współudział autora w czterech wystąpieniach konferencyjnych [IKGP08, ISK⁺08a, ISK08b, ISK07], z których jedno zostało wygłoszone przez autora [ISK08b]. Należy dodać, że wystąpienie to otrzymało wyróżnienie „Poland Section IEEE Electronic Design Chapter Special Awards“ podczas XV Międzynarodowej Konferencji MIXDES 2008 (Mixed Design of Integrated Circuits and Systems). W ramach prowadzonych prac autor uczestniczył także czynnie w spotkaniach kolaboracji FCAL i odbył staże naukowe w ośrodku badawczym DESY.

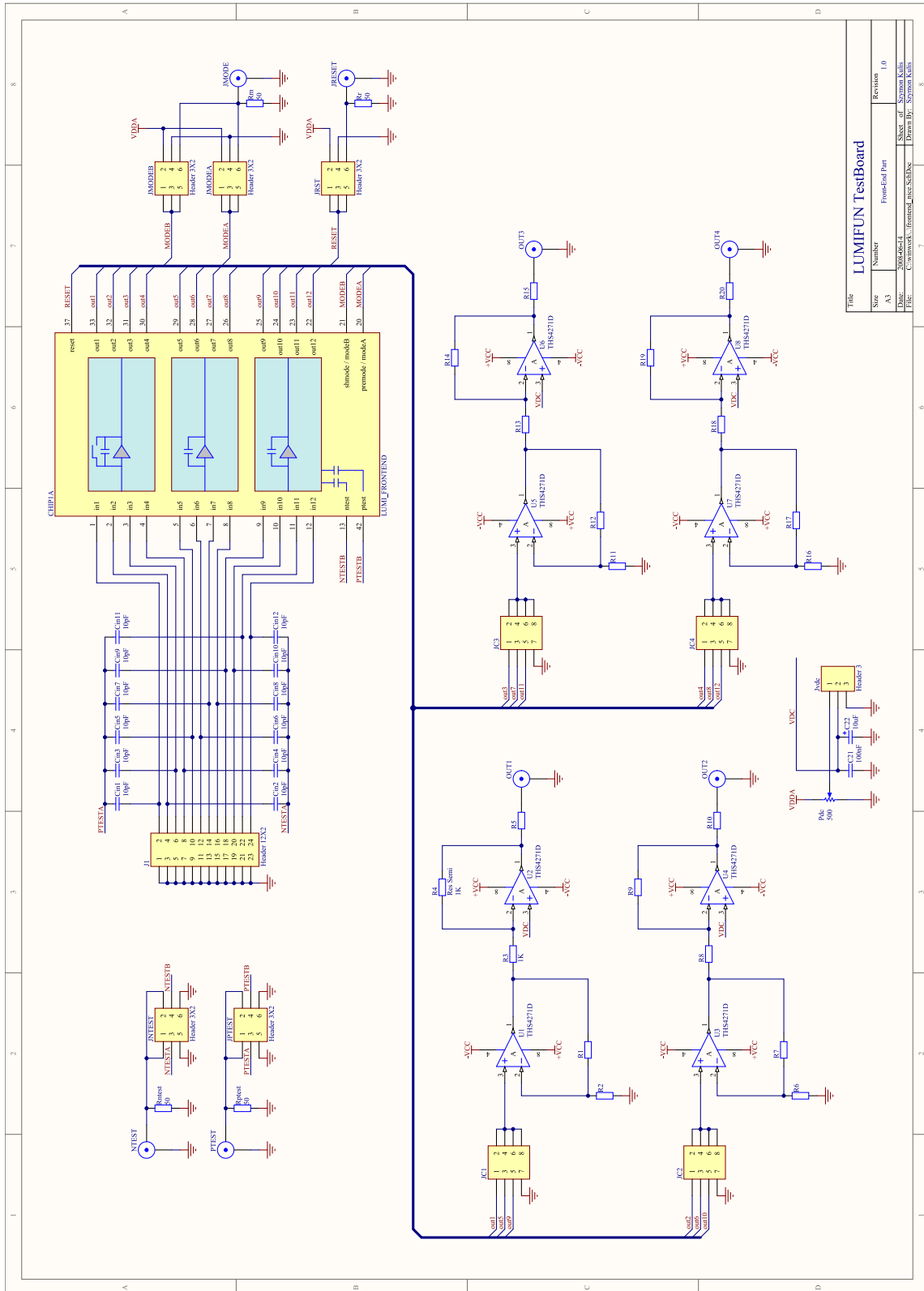
Dodatek A

Dokumentacja płytki drukowanej

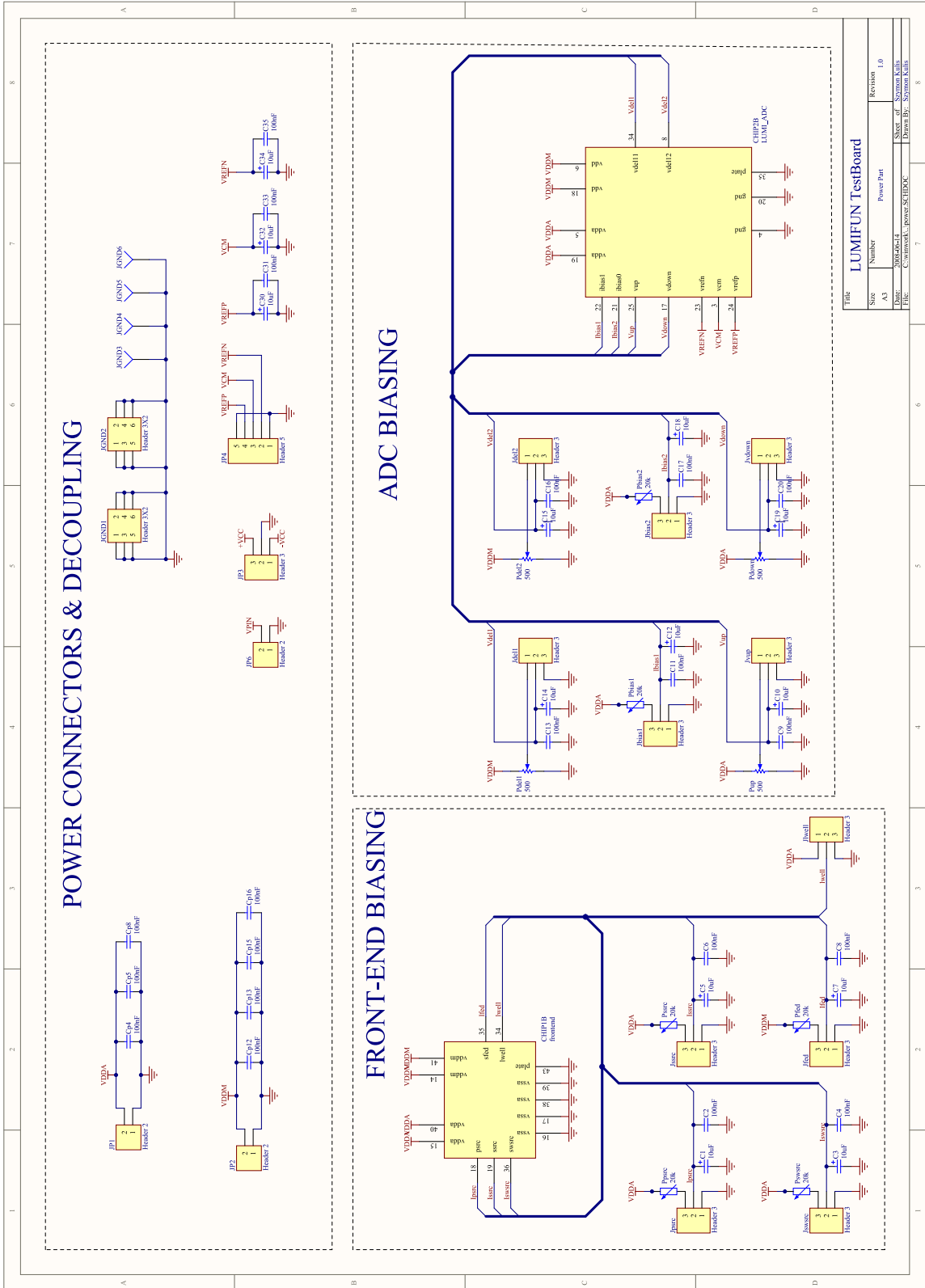
W załączniku tym zaprezentowano schematy ideowe oraz topologiczny plan masek płytki drukowanej, będącej częścią stanowiska pomiarowego, opisanego w rozdziale 5.1. Płytką została zaprojektowana w pakiecie Altium Designer [Alt06].



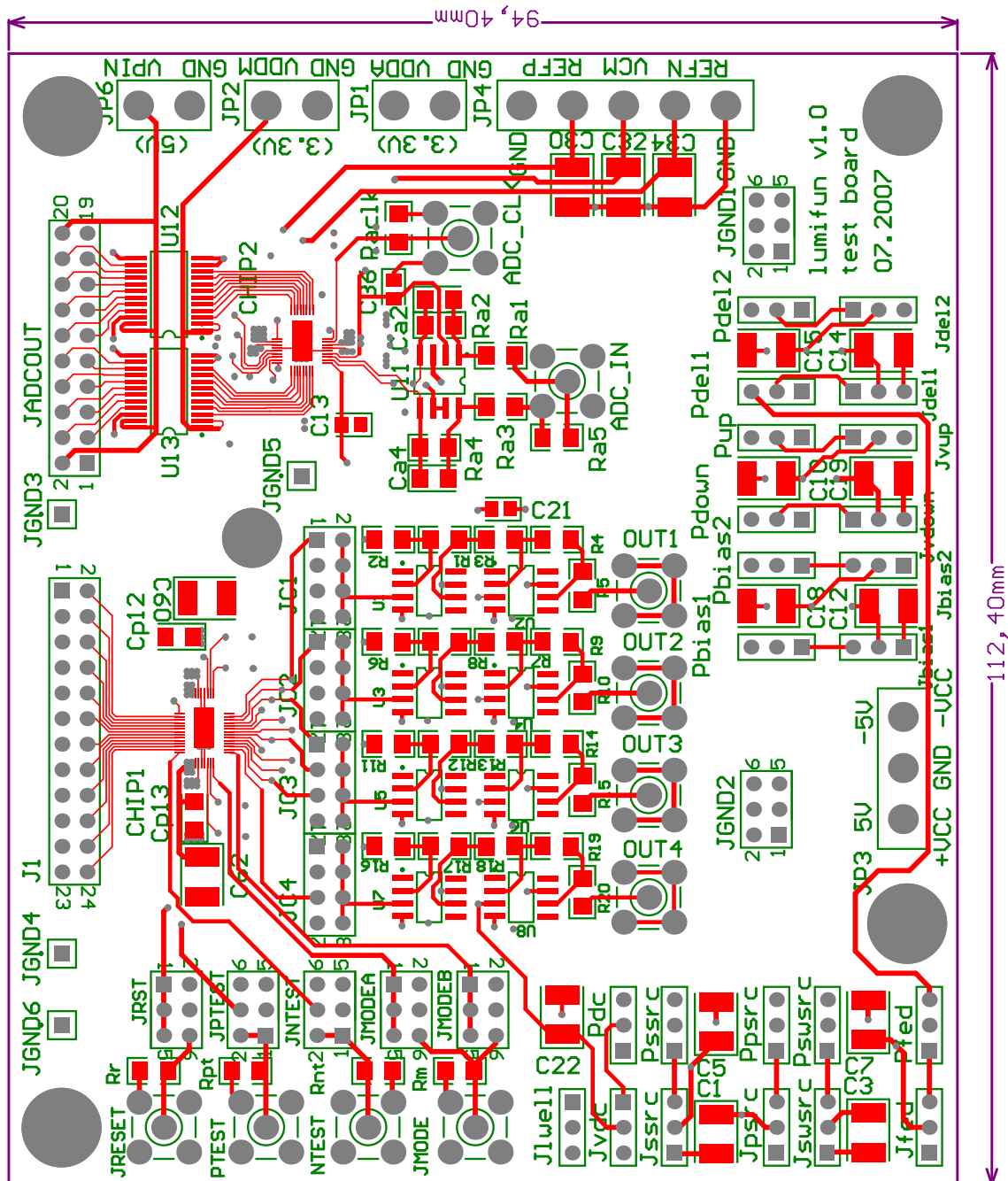
Rysunek A.1. Schemat części układu służącej do pomiarów ADC



Rysunek A.2. Schemat części układu służącej do pomiarów elektroniki front-end



Rysunek A.3. Schemat części zasilającej



Rysunek A.4. Plan masek płytki drukowanej. Widoczne warstwy to: górna warstwa metalizacji (kolor czerwony), warstwa opisu elementów (kolor zielony), warstwa otworów (kolor szary), warstwa pomocnicza (kolor fioletowy)

Dodatek B

Model bloku szybkiej logiki

W załączniku tym zaprezentowano model bloku szybkiej logiki, będącej częścią systemu akwizycji danych. Model ten został napisany w języku Verilog, a do syntezy wykorzystano narzędzia dostarczane przez firmę Xilinx.

```
1 `timescale 1ns / 1ps
2 /**
3  * clkIn      - sampling clock (on posedge)
4  * dataIn     - input data [16 bit]
5  * clkOut     - read out clock
6  * dataOut    - output data [8 bit]
7  * bytesel   - select HI / LO byta (to dataout)
8  * dataA, dataB - memory data ports [8 bit]
9  * reset     - main reset (activ high)
10 * ram_adr    - ram address [17 bit]
11 * ram_we    - ram write enable (active low)
12 * ram_ce    - ram chip enable (active low)
13 * ram_re    - ram read enable (active low)
14 * acq       - 1 = acquire , 0 = read
15 * busy      - 1 during aquisition or reading
16 * ser_data  - config serial interface data in
17 * ser_clk   - config serial interface clock
18 * ser_rst   - config serial interface rst
19 **/
20
21 module logic(clkin , datain ,
22             clkout , dataout , bytesel ,
23             dataA , dataB ,
24             ram_adr , ram_we , ram_re , ram_ce ,
25             reset , acq , busy ,
26             ser_data , ser_clk , ser_rst );
27
28 input clkin ;
29 input [15:0] datain ;
30 output [7:0] dataout ;
31 inout [7:0] dataA ;
32 inout [7:0] dataB ;
33 input bytesel ;
34 input clkout ;
35 input reset ;
36 output [16:0] ram_adr ;
37 output ram_we ;
38 output ram_re ;
39 output ram_ce ;
40 input acq ;
41 output busy ;
42 input ser_data ;
43 input ser_clk ;
44 input ser_rst ;
45
46
47 reg [17:0] adr ;
48 reg ram_we ;
49 reg ram_re ;
50 reg busy ;
51 reg [17:0] max_adr ;
52 wire clk ;
53
54 assign clk=acq ? clkin : clkout ;
55 assign ram_adr[16:0]=adr[16:0] ;
56 assign ram_ce=acq ? clk : 0 ;
57 assign dataout = (bytesel)? dataB : dataA ;
58 assign dataA = (~acq)?8'bz:datain[15:8] ;
59 assign dataB = (~acq)?8'bz:datain[7:0] ;
60
61 always @(acq or reset)
62 begin
63   if(reset)
64     begin
65       ram_we=1 ;
66       ram_re=1 ;
67     end
68   else
69     begin
70       //acq = writing
71       if(acq)
72         begin
73           ram_we=0 ;
74           ram_re=1 ;
75         end
76       //not acq = reading
77       else
78         begin
79           ram_we=1 ;
80           ram_re=0 ;
81         end
82     end
83 end
84
85 always @(adr or reset)
86 begin
87   if(reset)
88     begin
89       busy=1 ;
90     end
91   else
92     if(adr<max_adr)
93       busy=1 ;
94   else
```

```
95     busy=0;
96 end
97
98 always @(posedge clk or posedge reset)
99 begin
100     if(reset)
101         adr=0;
102     else
103         if(busy)
104             adr=adr+1;
105 end
106
107
108 always @(posedge ser_clk or posedge ser_rst)
109     begin
110         if(ser_rst)
111             begin
112                 max_adr=0;
113             end
114         else
115             begin
116                 max_adr = max_adr << 1 ;
117                 max_adr[0] = ser_data;
118             end
119         end
120
121 endmodule
```

Dodatek C

Oprogramowanie mikrokontrolera

W załączniku tym zaprezentowano oprogramowanie mikrokontrolera 8051, sterującego pracą szybkiego systemu akwizycji danych. Oprogramowanie zostało napisane w języku C, a skompilowane było przy użyciu kompilatora dostarczonego przez firmę Altium [Alt06]. Poniżej zaprezentowano pliki zawierające program główny (main.c), procedury obsługi bloku szybkiej logiki (daq.h, daq.c), portu szeregowego (uart.h, uart.c) oraz procedury pomocnicze (utils.h, utils.c).

C.1. main.c

```
1 /**
2  * file      : main.c
3  * project   : FPGA data aquisition
4  * author    : Szymon Kulis
5  * organization : AGH-UST Cracow, WFiIS
6  * created   : 02.02.2008
7  */
8
9 #include <stdio.h>
10 #include "uart.h"
11 #include "daq.h"
12
13 #define CMD_SET_LENGTH '1'
14 #define CMD_GET_LENGTH 'L'
15 #define CMD_SET_TIMEOUT 't'
16 #define CMD_GET_TIMEOUT 'T'
17 #define CMD_ACQUIRE 'a'
18 #define CMD_READ 'r'
19 #define CMD_PING 'p'
20 #define STATUS_OK 0x00
21 #define STATUS_TIMEOUT 0x01
22 #define STATUS_RDERR 0x02
23
24 void main( void )
25 {
26     //initialize uart
27     uartInit();
28     //max adress
29     u16 max_adr=0;
30     u08 status=0;
31     //milliseconds
32     u16 timeout=1000;
33     //infinite loop
34     while(1)
35     {
36         //if there is a new command
37         if(uartIsByte())
38         {
39             //receive
40             u08 cmp=uartRecv();
41             //and try to interpret
42             switch(cmp)
43             {
44                 case CMD_PING:
45                     status=STATUS_OK;
46                     break;
47                 case CMD_SET_LENGTH:
48                     max_adr=uartRecvU16();
49                     DAQsetLen(max_adr);
50                     status=STATUS_OK;
51                     break;
52                 case CMD_GET_LENGTH:
53                     uartSendU16(max_adr);
54                     status=STATUS_OK;
55                     break;
56                 case CMD_SET_TIMEOUT:
57                     timeout=uartRecvU16();
58                     status=STATUS_OK;
59                     break;
60                 case CMD_GET_TIMEOUT:
61                     uartSendU16(timeout);
62                     status=STATUS_OK;
63                     break;
64                 case CMD_ACQUIRE:
65                     status=DAQacquire(timeout);
66                     break;
67                 case CMD_READ:
68                     status=DAQread(max_adr);
69                     break;
70                 default: break;
71             } //end switch(cmp)
72             uartSend(status);
73
74         }
75     } //end while(1)
76 } //end main
```

C.2. daq.h

```

1 /**
2  * file      : daq.h
3  * project   : FPGA data aquisition
4  * author    : Szymon Kulis
5  * organization : AGH-UST Cracow, WFiIS
6  * created   : 02.02.2008
7  */
8
9 #ifndef _daq_h_
10 #define _daq_h_
11 #include "utils.h"
12 #include "uart.h"
13
14 // ports definitions
15 #define DAQ_PORT      P2
16 #define DAQ_ACQ       0
17 #define DAQ_RESET     1
18 #define DAQ_BUSY      2
19 #define DAQ_BYTESEL   3
20 #define DAQ_CLKOUT    4
21 #define DAQ_SER_CLK   5
22 #define DAQ_SER_DATA  6
23 #define DAQ_SER_RST   7
24
25 /**
26  * set data len to be acquired
27  */
28 void DAQsetLen(u16 len);
29
30 /**
31  * acquire data
32  * returns OK or TIMEOUT
33  */
34 u08 DAQacquire(u16 timeout);
35
36 /**
37  * read data
38  * returns OK or TIMEOUT
39  */
40 u08 DAQread(u16 data2read);
41
42 #endif

```

C.3. daq.c

```

1 /**
2  * file      : daq.c
3  * project   : FPGA data aquisition
4  * author    : Szymon Kulis
5  * organization : AGH-UST Cracow, WFiIS
6  * created   : 02.02.2008
7  */
8
9 #include "daq.h"
10
11 void DAQsetLen(u16 len)
12 {
13     sbi(DAQ_PORT, DAQ_BYTESEL);
14     cbi(DAQ_PORT, DAQ_SER_DATA);
15     cbi(DAQ_PORT, DAQ_SER_CLK);
16     cbi(DAQ_PORT, DAQ_SER_RST);
17     sbi(DAQ_PORT, DAQ_ACQ);
18     cbi(DAQ_PORT, DAQ_RESET);
19
20     //reset internal counter
21     sbi(DAQ_PORT, DAQ_SER_RST);
22     cbi(DAQ_PORT, DAQ_SER_RST);
23
24     u08 i;
25     //shift 2 bytes
26     for(i=0; i<16; i++)
27     {
28         if(len&0x8000)
29         {
30             sbi(DAQ_PORT, DAQ_SER_DATA);
31             //uartSend('1');
32         }
33         else
34         {
35             cbi(DAQ_PORT, DAQ_SER_DATA);
36             //uartSend('0');
37         }
38         __asm("nop");
39         //clock
40         sbi(DAQ_PORT, DAQ_SER_CLK);
41         //shift len
42         len <<=1;
43         cbi(DAQ_PORT, DAQ_SER_CLK);
44     }
45 }
46
47 u08 DAQacquire(u16 timeout)
48 {
49     P3=0;
50     cbi(DAQ_PORT, DAQ_SER_CLK);
51     cbi(DAQ_PORT, DAQ_BYTESEL);
52     sbi(DAQ_PORT, DAQ_ACQ);
53     sbi(DAQ_PORT, DAQ_RESET);
54     cbi(DAQ_PORT, DAQ_RESET);
55
56     //wait for acquisition end
57     while(is_bit_set(DAQ_PORT, DAQ_BUSY) && timeout>0)
58     {
59         delays(1);
60         timeout--;
61     }
62     sbi(DAQ_PORT, DAQ_BYTESEL);
63     cbi(DAQ_PORT, DAQ_ACQ);
64     if(timeout==0)
65         return 1;
66     return 0;
67 }
68
69 u08 DAQread(u16 data2read)
70 {
71     cbi(DAQ_PORT, DAQ_ACQ);
72     cbi(DAQ_PORT, DAQ_CLKOUT);
73     sbi(DAQ_PORT, DAQ_RESET);
74     cbi(DAQ_PORT, DAQ_RESET);
75     cbi(DAQ_PORT, DAQ_BYTESEL);
76
77     while(is_bit_set(DAQ_PORT, DAQ_BUSY) && data2read>0)
78     {

```



```

79 //change adress
80 sbi(DAQ_PORT,DAQ_CLKOUT);
81 __asm( "nop" );
82 //select first byte to read
83 cbi(DAQ_PORT,DAQ_BYTESEL);
84 __asm( "nop" );
85 //send first byte
86 uartSend(P1);
87 //select second byte
88 sbi(DAQ_PORT,DAQ_BYTESEL);
89 __asm( "nop" );
90 //send second byte
91 uartSend(P1);
92 cbi(DAQ_PORT,DAQ_CLKOUT);
93 __asm( "nop" );
94 data2read--;
95 }
96
97 if(is_bit_set(DAQ_PORT,DAQ_BUSY))
98     return 02;
99 return 0;
100 }

```

C.4. uart.h

```

1 /**
2  * file      : uart.h
3  * project   : FPGA data aquisition
4  * author    : Szymon Kulis
5  * organization : AGH-UST Cracow, WFiIS
6  * created   : 02.02.2008
7  */
8
9 #ifndef _uart_h_
10 #define _uart_h_
11
12 #include "utils.h"
13 //timing constans
14 #define XTAL 22119000
15 #define BAUDRATE 115200
16
17 /**
18  * initialize uart
19  */
20 void uartInit(void);
21
22 /**
23  * send unsigned byte
24  */
25 void uartSend(unsigned char data);
26
27 /**
28  * receive unsigned byte
29  */
30 unsigned char uartRecv(void);
31
32 /**
33  * check if there is aby byte to read
34  */
35 unsigned char uartIsByte(void);
36
37 /**
38  * receive unsigned short
39  */
40 u16 uartRecvU16(void);
41
42 /**
43  * send unsigned short
44  */
45 void uartSendU16(u16 n);
46
47 #endif

```

C.5. uart.c

```

1 /**
2  * file      : uart.c
3  * project   : FPGA data aquisition
4  * author    : Szymon Kulis
5  * organization : AGH-UST Cracow, WFiIS
6  * created   : 02.02.2008
7  */
8
9 #include "uart.h"
10 void uartInit()
11 {
12     // SCON: mode 1, 8-bit UART, enable rcvr
13     SCON = 0x50;
14     // TMOD: timer 1, mode 2, 8-bit reload
15     TMOD |= 0x20;
16     PCON |= 0x80;
17     TH1 = (unsigned char)
18         (256 - (XTAL / (16L * 12L * BAUDRATE)));
19     // TR1: timer1 run
20     TR1 = 1;
21     // TI: set TI to send first char of UART
22     TI = 1;
23 }
24
25 void uartSend(unsigned char data)
26 {
27     while(!is_bit_set(SCON,1));
28     cbi(SCON,1);
29     SBUF=data;
30 }
31
32 unsigned char uartRecv(void)
33 {
34     while(!is_bit_set(SCON,0));
35     cbi(SCON,0);
36     return SBUF;
37 }
38
39 unsigned char uartIsByte()
40 {

```

```

41     return is_bit_set(SCON,0);
42 }
43
44 u16 uartRecvU16()
45 {
46     u16 tmp=uartRecv();
47     tmp<<=8;
48     tmp+=uartRecv();
49     return tmp;

```

```

50 }
51
52 void uartSendU16(u16 n)
53 {
54     uartSend(n>>8);
55     uartSend(n&0xFF);
56 }

```

C.6. utils.h

```

1 /**
2  * file           : utils.h
3  * project        : FPGA data aquisition
4  * author         : Szymon Kulis
5  * organization   : AGH-UST Cracow, WFiIS
6  * created        : 02.02.2008
7  */
8
9 #ifndef _utils_h_
10 #define _utils_h_
11
12 // bit access instructions
13 #define sbi(port, bit) (port) |= (1 << (bit))

```

```

14 #define cbi(port, bit) (port) &= ~(1 << (bit))
15 #define is_bit_set(port, bit) (port & (1 << bit))
16
17 //type defs
18 typedef unsigned char  u08;
19 typedef unsigned short u16;
20 typedef unsigned long  u32;
21
22 //delay ms
23 void delayms(u16 d);
24 #endif

```

C.7. utils.c

```

1 /**
2  * file           : utils.c
3  * project        : FPGA data aquisition
4  * author         : Szymon Kulis
5  * organization   : AGH-UST Cracow, WFiIS
6  * created        : 02.02.2008
7  */
8
9 #include "utils.h"
10
11

```

```

12 void delayms(u16 d)
13 {
14
15     u16 i;
16     for (;d>0;d--)
17         for (i=0;i<1370;i++)
18             __asm( "nop" );
19
20 }

```

Dodatek D

Biblioteki obsługi sprzętu

W załączniku tym zaprezentowano część biblioteki obsługującej sprzęt laboratoryjny. Biblioteka została napisana w języku C++. Jest ona przystosowana do pracy pod kontrolą systemu Linux oraz Windows. Zawiera klasy obsługi następujących urządzeń:

- zasilacze (e3631, hp4145, hp6624),
- oscyloskopy cyfrowe (tds, tds3000, tds700),
- programowalne generatory funkcyjne (awg2000, afg3102),
- cyfrowy miernik wielofunkcyjny (hp34401a),
- szybki system akwizycji danych cyfrowych (nanodaq),
- karta cyfrowa PCI (pci6534).

Ze względu na ograniczoną objętość pracy poniżej przedstawiono tylko pliki źródłowe klasy bazowej dla wszystkich klas urządzeń korzystających z interfejsu GPIB.

D.1. gpibdev.h

```
1 #include <stdlib.h>
2 #include <stdio.h>
3 #include <vector>
4 #include <iostream>
5
6 #ifdef __WIN32__
7 #include <windows.h>
8 #endif
9 // #include "ni488.h"
10 #ifdef __WIN32__
11 #include "ni488.h"
12 #else
13 #include <gpib/ib.h>
14 #endif
15 #include "stringutils.h"
16
17 #ifndef _GPIBDEV_H_
18 #define _GPIBDEV_H_
19
20 /**
21  * Klasa do obsługi urządzeń korzystających z magistrali GPIB.
22  */
23 class GpibDev
24 {
25
26 public:
27
```

```

28  /**
29   * Konstruktor (nie operuje na samym gpib, ustawia tylko adresy)
30   * @param pa primary address
31   * @param sa secondary address
32   * @param bi board index
33   */
34  GpibDev(int pa=12,int sa=0,int bi=0);
35
36  /**
37   * Destraktor (nic nie robi)
38   */
39   virtual ~GpibDev();
40
41  protected:
42
43   ///! Device unit descriptor
44   int m_Device;
45   ///! Interface Index (GPIB0=0,GPIB1=1,etc..)
46   int m_BoardIndex;
47   ///! Primary address of the device
48   int m_PrimaryAddress;
49   ///! Secondary address of the device
50   int m_SecondaryAddress;
51   ///! Manufactor Name
52   std::string m_ManufactorName;
53   ///! Device Name
54   std::string m_DeviceName;
55
56   // debug ON/OFF
57   bool debug;
58
59
60
61
62
63  /**
64   * zwraca blad GPIB, z dodatkowym komunikatem tekstowym
65   * @param msg komunikat dodatkowy
66   */
67   void gpib_error(std::string msg);
68
69
70  /**
71   * wysyla komunikat do uzadzenia. Jak sie nie uda to wypisuje komunikat o bledzie
72   * i konczy dzialanie programu.
73   * @param cmd komunikat
74   * @return true jak sie uda
75   */
76   bool wr(std::string cmd);
77
78  /**
79   * wysyla komunikat do uzadzenia z bufora
80   * @param buf bufor z ktorego chcemy czytac
81   * @param nr liczba znakow do wyslania
82   */
83   bool wrBuf(void *buf,int nr=1024);
84
85
86  /**
87   * czyta odpowiedz od uzadzenia
88   * @param nr maksymalna liczba znakow jaka chcemy odebrac
89   */
90   std::string rd(int nr=1024);
91
92  /**
93   * czyta odpowiedz od uzadzenia, zapisuje do bufora
94   * @param buf bufor do ktorego chcemy pisac
95   * @param nr maksymalna liczba znakow jaka chcemy odebrac
96   * @return ilos odczytanych znakow
97   */
98   int rdBuf(void *buf,int nr=1024);

```

```
99
100 /**
101  * wysyla komunikat do uzadzenia i czeka na odpowiedz
102  * @param cmd komunikat
103  * @param nr maksymalna liczba znakow jaka chcemy odebrac
104  */
105 std::string query(std::string cmd,int nr=1024);
106
107 /**
108  * zrzuca informacje z GPIB do pliku
109  * @param filename nazwa pliku
110  */
111 bool GPIBtoFile(std::string filename);
112
113 /**
114  * zrzuca plik na GPIB
115  * @param filename nazwa pliku
116  */
117 bool FiletoGPIB(std::string filename);
118
119
120 public:
121
122 /**
123  * Ustawia nazwe producenta (wazne przed autoconnect)
124  * @param ManufacturerName nazwa producenta
125  */
126 void setManufacturerName(std::string ManufacturerName);
127
128 /**
129  * Ustawia nazwe uzadzenia (wazne przed autoconnect)
130  * @param DeviceName nazwa uzadzenia
131  */
132 void setDeviceName(std::string DeviceName);
133
134 /**
135  * Probuje sie polaczyc z uzadzeniem pod wczesniej ustawionym adresem (pa,sa,bi)
136  * @return true jak sie uda / false jak cos jest nie tak
137  */
138 virtual bool connect(void);
139
140 /**
141  * Probuje sie polaczyc z uzadzeniem o zadnym adresie
142  * @param pa primary adress
143  * @param sa secondary address
144  * @param bi board index
145  * @return true jak sie uda / false jak cos jest nie tak
146  */
147 virtual bool connect(int pa,int sa=0,int bi=0);
148
149 /**
150  * Sprawdza czy uzadzenie jest poprawnie podlaczone
151  * tzn. czy m_ManufacturerName uzadzenia m_DeviceName zgadzaja sie z tymi
152  * ustawionymi w klasie.
153  * @return true jesli tak / false jesli nie
154  */
155 virtual bool check(void);
156
157 /**
158  * Probuje znalezc adres uzadzenia ktore sie przedstawia uzywajac
159  * m_ManufacturerName uzadzenia m_DeviceName.
160  * @return true jesli sie uda / false jesli nie
161  */
162 virtual bool autoConnect();
163
164 /**
165  * Resetuje uzadzenie
166  */
167 virtual void clearDevice(void);
168
169 /**
```

```

170  * włącza wylacza debug
171  * @param dd czy włączyć debug
172  **/
173  void setDebug (bool dd=true);
174  };
175 #endif

```

D.2. gpibdev.cpp

```

1
2 #include <stdio.h>
3 #include "gpibdev.h"
4 #ifdef __WIN32__
5 #include "wingpib.h"
6 #endif
7
8 ///////////////////////////////////////////////////////////////////
9 GpibDev::GpibDev (int pa, int sa, int bi)
10 {
11  m.PrimaryAddress=pa;
12  m.SecondaryAddress=sa;
13  m.BoardIndex=bi;
14  debug=false;
15 }
16 ///////////////////////////////////////////////////////////////////
17
18 GpibDev::~GpibDev ()
19 {
20 }
21
22 ///////////////////////////////////////////////////////////////////
23 bool GpibDev::connect (void)
24 {
25  if (debug)
26  {
27  std::cout << "GpibDev::connect()\n";
28  std::cout << "\tBoard_Index: " << m.BoardIndex << "\n";
29  std::cout << "\tPrimary_Address: " << m.PrimaryAddress << "\n";
30  std::cout << "\tSecondary_Address: " << m.SecondaryAddress << "\n";
31  }
32
33  m.Device = ibdev (m.BoardIndex, m.PrimaryAddress, m.SecondaryAddress, T1s, 1, 0);
34  if (ibsta & ERR)
35  gpib_error ("Unable to open device");
36
37  clearDevice ();
38
39  if (debug)
40  std::cout << "\tDevice_number: " << m.Device << "\n";
41  return true;
42 }
43
44
45
46 ///////////////////////////////////////////////////////////////////
47 bool GpibDev::connect (int pa, int sa, int bi)
48 {
49  m.PrimaryAddress=pa;
50  m.SecondaryAddress=sa;
51  m.BoardIndex=bi;
52  return connect ();
53 }
54 ///////////////////////////////////////////////////////////////////
55 void GpibDev::clearDevice (void)
56 {
57  ibclr (m.Device);

```

```

58     if (ibsta & ERR)
59         gpib_error("Unable_to_clear_device");
60     //if ((ibclr(m_Device) < 0) ||
61         //    (ibsre(m_BoardIndex,0) < 0))
62     //    gpib_error("ibclr/ibsre Error: Unable to clear device/board!");
63 }
64 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
65 bool GpibDev::wr(std::string cmd)
66 {
67     if (debug)
68         std::cout << "[GpibDev::wr]_"<<cmd<<std::endl;
69     ibtmo(m_Device, T10s);
70     ibwrt(m_Device, (void *)cmd.c_str(), cmd.length());
71     if (ibsta & ERR)
72     {
73         gpib_error(std::string ("ibwrt_error_::_" +cmd));
74         return false;
75     }
76     return true;
77 }
78
79 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
80 bool GpibDev::wrBuf(void *buf,int nr)
81 {
82     ibtmo(m_Device, T10s);
83     ibwrt(m_Device, buf, nr);
84     if (ibsta & ERR)
85     {
86         gpib_error("ibwrt_Error");
87         return false;
88     }
89     return true;
90 }
91
92 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
93 std::string GpibDev::rd(int nr)
94 {
95
96     /* Read nr bytes from the device */
97     char buf[nr];
98     ibtmo(m_Device, T10s);
99     ibrd(m_Device, buf, nr);
100    if (ibsta & ERR)
101        gpib_error("ibrd_Error");
102
103    buf[ibcntl] = '\0';
104    if (debug)
105        std::cout << "[GpibDev::rd]_"<<buf<<"\n";
106
107    //skoncz stringa
108    return std::string(buf);
109 }
110
111
112
113 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
114 int GpibDev::rdBuf(void *buf,int nr)
115 {
116     /* Read nr bytes from the device */
117     ibtmo(m_Device, T10s);
118     ibrd(m_Device, buf, nr);
119     if (ibsta & ERR)
120         gpib_error("ibrd_Error");
121     //skoncz stringa
122     return ibcntl;
123 }
124
125 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
126 std::string GpibDev::query(std::string cmd,int nr)
127 {
128     if (!wr(cmd.c_str()))

```

```

129     return (std::string)";
130     std::string rep=rd(nr);
131     if(rep.length()>0 && rep[rep.length()-1]=='\n')
132         rep.erase((rep.end()-1));
133
134     return rep;
135 }
136 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
137 bool GpibDev:: GPIBtoFile(std::string cmd)
138 {
139     return ibrd(m.Device, (char*)(cmd.c_str()));
140 }
141 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
142 bool GpibDev:: FiletoGPIB(std::string cmd)
143 {
144     return ibwrtf(m.Device, (char *)cmd.c_str());
145 }
146 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
147 bool GpibDev::check()
148 {
149     // std::cout << m.DeviceName <<" : ";
150
151     //wyslij pytanie o identyfikator
152     std::string dev_info=query(" *IDN?");
153
154 }
155
156 //pokrojone po przecinkach. Sprawdzamy ilosc pol
157 std::vector<std::string> dev_rec;
158 int dev_len= StringUtils::SplitString(dev_info,";", dev_rec);
159 if(dev_len>2){
160     if(debug) {
161         std::cout << "READ" << std::endl;
162         std::cout << "\t___Device_number_:_"<<m.Device<<"\n";
163         std::cout << "\t___Manufacturer___:"<<dev_rec[0]<<"\n";
164         std::cout << "\t___Device_____:"<<dev_rec[1]<<"\n";
165     }
166     if (dev_rec[0] == m.ManufactorName
167         && dev_rec[1] == m.DeviceName)
168     {
169         // std::cout<<"[ OK ]"<<std::endl;
170         return true;
171     }
172     else {
173         std::cout << "READ" << std::endl;
174         std::cout << "\t___Device_number_:_"<<m.Device<<"\n";
175         std::cout << "\t___Manufacturer___:"<<dev_rec[0]<<"\n";
176         std::cout << "\t___Device_____:"<<dev_rec[1]<<"\n";
177         std::cout << "SHOUD_BE_:" << std::endl;
178         std::cout << "\t___Manufacturer_____:"<<m.ManufactorName<<"\n";
179         std::cout << "\t___Device_____:"<<m.DeviceName<<"\n";
180         std::cout << "\t___Primary_address___:"<<m.PrimaryAddress<<"\n";
181         std::cout << "\t___Secondary_address_:"<<m.SecondaryAddress<<"\n";
182     }
183 }
184 std::cout<<" [FAILED]"<<std::endl;
185 return false;
186 }
187 //////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
188 bool GpibDev:: autoConnect()
189 {
190     bool FOUND=false;
191
192     if(debug)
193         std::cout << "GpibDev: auto_conect()\n";
194
195     //maksymalny dozwolony adres
196     const int MAX_ADR=32;
197
198     //na poczatku nie znamy zadnego adresu, na wszelki wypadek ustawiamy na -1
199     m.PrimaryAddress = -1;

```



```

200 m_SecondaryAddress = -1;
201
202 SendIFC(m_BoardIndex);
203 if (ibsta & ERR)
204 {
205     gpib_error("SendIFC_error");
206     return false;
207 }
208
209 // Array of primary addresses
210 Addr4882_t Instruments[MAX_ADR];
211 // Array of listen addresses
212 Addr4882_t Result[MAX_ADR];
213 // Number of instruments on GPIB
214 }
215
216 int Num_Instruments;
217 // tworzymy tablice zawierajaca wszystkie dozwolone adresy
218 for (int i = 0; i < MAX_ADR-2; i++)
219     Instruments[i] = (Addr4882_t)(i + 1);
220 Instruments[MAX_ADR-2] = NOADDR;
221
222
223 if(debug)
224     std::cout << "\tFinding_all_instruments_on_the_bus...\n";
225 FindLstn(m_BoardIndex, Instruments, Result, MAX_ADR-1);
226 if (ibsta & ERR)
227 {
228     gpib_error("FindLstn_error");
229     return false;
230 }
231
232 //ibcntl zawiera liczbe adresow przechowywanych w tablicy rezultatow
233
234 Num_Instruments = ibcntl;
235
236 if(debug)
237     std::cout << "\tFound_"<< Num_Instruments<< "_instruments\n";
238
239 Result[Num_Instruments] = NOADDR;
240
241 for (int i = 0; i < Num_Instruments; i++)
242 {
243     //pobierz primary adres
244     int pad = GetPAD(Result[i]);
245     //pobierz secondary adres
246     int sad = GetSAD(Result[i]);
247     //sprobuj sie polaczyc z urzadzeniem
248     m_Device = ibdev( m_BoardIndex, pad,sad, T1s, 1, 0);
249     if (ibsta & ERR)
250     {
251         gpib_error("ibdev_Error");
252         return false;
253     }
254     //czyszc urzadzenie
255     ibclr(m_Device);
256     if (ibsta & ERR)
257     {
258         gpib_error("ibclr_Error");
259         return false;
260     }
261
262     //wyslij pytanie o identyfikator
263     wr("*IDN?");
264     //linijkajaka zwraca udzadzenie
265     std::string dev_info=rd();
266     //pokrojona po przecinkach
267     std::vector<std::string> dev_rec;
268     //ilosc pol
269     int dev_len= StringUtils::SplitString(dev_info, ",", dev_rec);
270     if(dev_len>2)

```

```

271 { if(debug)
272 {
273     std::cout << "\t"<<i<<" "<<"_";
274     std::cout << "Manufacturer_____:" <<dev_rec[0]<<"\n";
275     std::cout << "\t___Device_____:"<<dev_rec[1]<<"\n";
276     std::cout << "\t___Primary_address___:"<<pad<<"\n";
277     std::cout << "\t___Secondary_address_:"<<sad<<"\n";
278 }
279 //jesli nic do tej pory nie zalezlismy a producent i opis sie zgadzaja
280 if(m.PrimaryAddress<0 &&
281     m.ManufactorName==std::string(dev_rec[0]) &&
282     m.DeviceName     ==std::string(dev_rec[1]) )
283     {
284         m.PrimaryAddress = pad;
285         m.SecondaryAddress = sad;
286         //std::cout << "\tFOUND !\n";
287     }
288     if(m.ManufactorName==std::string(dev_rec[0]) &&
289        m.DeviceName     ==std::string(dev_rec[1]) )   FOUND=true;
290 }
291 //sprzatamy
292 ibonl(m.Device, 0);          /* Take the device offline          */
293 if (ibsta & ERR)
294     gpib_error("ibonl_Error");
295
296 }          //koniec przelatywania po urzadzeniach
297
298 //sprztamy , board offline
299 ibonl (m.BoardIndex, 0);
300
301 //jesli cos znalezlimy to probujemy sie polaczyc
302
303 //if(m.PrimaryAddress>0)
304 if(FOUND)
305     return connect();
306 else return false;
307 }
308 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
309 void GpibDev::gpib_error(std::string msg) {
310
311     std::cout <<msg<<std::endl;
312
313     printf ("ibsta_=%x_<", ibsta);
314     if (ibsta & ERR ) printf ("_ERR");
315     if (ibsta & TIMO) printf ("_TIMO");
316     if (ibsta & END ) printf ("_END");
317     if (ibsta & SRQI) printf ("_SRQI");
318     if (ibsta & RQS ) printf ("_RQS");
319     if (ibsta & CMPL) printf ("_CMPL");
320     if (ibsta & LOK ) printf ("_LOK");
321     if (ibsta & REM ) printf ("_REM");
322     if (ibsta & CIC ) printf ("_CIC");
323     if (ibsta & ATN ) printf ("_ATN");
324     if (ibsta & TACS) printf ("_TACS");
325     if (ibsta & LACS) printf ("_LATEKTRONIX,TDSCS");
326     if (ibsta & DTAS) printf ("_DTAS");
327     if (ibsta & DCAS) printf ("_DCAS");
328     printf ("_>\n");
329
330     printf ("iberr_=%d", iberr);
331     if (iberr == EDVR) printf ("_EDVR_<Driver_Error>\n");
332     if (iberr == ECIC) printf ("_ECIC_<Not_Controller-In-Charge>\n");
333     if (iberr == ENOL) printf ("_ENOL_<No_Listener>\n");
334     if (iberr == EADR) printf ("_EADR_<Address_error>\n");
335     if (iberr == EARG) printf ("_EARG_<Invalid_argument>\n");
336     if (iberr == ESAC) printf ("_ESAC_<Not_System_Controller>\n");
337     if (iberr == EABO) printf ("_EABO_<Operation_aborted>\n");
338     if (iberr == ENEB) printf ("_ENEB_<No_GPIB_board>\n");
339     if (iberr == EDMA) printf ("_EDMA_<DMA_Error>\n");
340     if (iberr == EOIP) printf ("_EOIP_<Async_I/O_in_progress>\n");
341     if (iberr == ECAP) printf ("_ECAP_<No_capability>\n");

```

```
342     if (iberr == EFSO) printf ("_EFSO_<File_system_error>\n");
343     if (iberr == EBUS) printf ("_EBUS_<Command_error>\n");
344     if (iberr == ESRQ) printf ("_ESRQ_<SRQ_stuck_on>\n");
345     if (iberr == ETAB) printf ("_ETAB_<Table_Overflow>\n");
346     printf ("ibcntl_=%ld\n", ibcntl);
347     printf ("\n");
348
349     /* Call ibonl to take the interface offline */
350     ibonl (m_BoardIndex,0);
351
352 #ifdef __WIN32__
353     FreeDll();
354 #endif
355     exit(1);
356 }
357
358
359 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
360 // setery /getery
361 ///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
362
363 void GpibDev::setManufacturerName(std::string ManufacturerName)
364 {
365     m_ManufacturerName=ManufacturerName;
366 }
367
368 void GpibDev::setDeviceName(std::string DeviceName)
369 {
370     m_DeviceName=DeviceName;
371 }
372
373 void GpibDev::setDebug(bool dd)
374 {
375     debug=dd;
376 }
```


Bibliografia

- [Alt04] Altium, *Nanoboard-NB1 Technical Reference Manual*, 2004.
- [Alt06] Altium, *Altium designer – wprowadzenie*, 2006.
- [Ana05] Analog Devices, *Nota aplikacyjna układu AD9445*, 2005.
- [ATL00] Kolaboracja ATLAS, *ATLAS physics TDR*, <http://atlas.web.cern.ch/>, 2000.
- [BDD⁺07] J. Błocki, W. Daniluk, W. Dąbrowski, M. Gil1, U. Harder, M. Idzik, E. Kielar, A. Moszczyński, K. Oliwa, B. Pawlik, L. Suszycki, W. Wierba, i L. Zawiejski, *Silicon sensors for the LumiCal prototype design*, EUDET-Memo-2007-47, 2007.
- [Bre06] M. Breidenbach, *KPIX & EMCAL*, www-conf.slac.stanford.edu, 2006.
- [Cad08] Cadence Design Systems, <http://www.cadence.com/>, 2008.
- [CLRS04] Thomas Cormen, Charles Leiserson, Ronald Rivest, i Clifford Stein, *Wprowadzenie do algorytmów*, Wydawnictwo Naukowo-Techniczne, 2004.
- [CMS05] Kolaboracja CMS, *CMS physics TDR*, <http://cmsdoc.cern.ch/>, 2005.
- [Dag01] Stranneby Dag, *Cyfrowe przetwarzanie sygnałów – Metody, Algorytmy, Zastosowania*, BTC, 2001.
- [ES07a] ILC Global Design Effort i World Wide Study, *International Linear Collider. Reference Design Report. Accelerator*, <http://ilcdoc.linearcollider.org/>, 2007.
- [ES07b] ———, *International Linear Collider. Reference Design Report. Detectors*, <http://ilcdoc.linearcollider.org/>, 2007.
- [ES07c] ———, *International Linear Collider. Reference Design Report. Physics at the ILC*, <http://ilcdoc.linearcollider.org/>, 2007.
- [FCA08] Kolaboracja FCAL, *R&D for the Very Forward Region of the ILC*, <http://www-zeuthen.desy.de/ILC/fcal/>, 2008.
- [Fiu07] Tomasz Fiutowski, *Ekstrakcja i przetwarzanie sygnałów z pozycyjnych dwuwymiarowych detektorów do zastosowań w rozpraszaniu neutronów*, praca doktorska, Wydział Fizyki i Informatyki Stosowanej, 2007.
- [Fut05] Future Technology Devices International, *Nota aplikacyjna układu FT232BM*, 2005.
- [Gaj07] Jan Gajewski, *Kształtowanie impulsów i optymalizacja szumowa systemu odczytu sygnałów z detektora półprzewodnikowego*, praca magisterska, Wydział Fizyki i Informatyki Stosowanej, AGH, 2007.

- [GHLM01] P. Gray, P. Hurst, S. Lewis, i R. Meyer, *Analysis and design of analog integrated circuits*, John Willey and Sons, 2001.
- [GM86] E. Gatti i P.F. Manfredi, *Processing the Signals from Solid-State Detectors in Elementary-Particle Physics*, *Revista Del Nuovo Cimento* (1986).
- [HH05] Alan Hastings i Roy Alan Hastings, *The Art of Analog Layout*, Paperback, 2005.
- [IEE00] IEEE, *Ieee standard for terminology and test methods for analog-to-digital converters*, 2000.
- [IKGP08] M. Idzik, Sz. Kulis, J. Gajewski, i D. Przyborowski, *Development od Front-end Electronics for the Luminosity Detector at ILC*, Proceedings of the International Conference Mixed Design of Integrated Circuits and Systems Poznań, 2008.
- [ILC08] Kolaboracja ILC, *The International Linear Collider*, <http://www.linearcollider.org/>, 2008.
- [ISK07] M. Idzik, K. Swientek, i Sz. Kulis, *Read-out electronics for the Lumical*, LCWS Workshop, 2007.
- [ISK⁺08a] M. Idzik, K. Swientek, Sz. Kulis, J. Gajewski, i D. Przyborowski, *Read-out electronics for the Lumical*, ILC-ECFA Workshop, 2008.
- [ISK08b] Marek Idzik, Krzysztof Swientek, i Szymon Kulis, *Development od Pipeline ADC for the Luminosity Detector at ILC*, Proceedings of the International Conference Mixed Design of Integrated Circuits and Systems Poznań, 2008 (2008).
- [Kor05] K. Korbel, *Układy elektroniki „front-end”*, Wydawnictwo AGH, 2005.
- [LDC08] Kolaboracja LDC, *The Large Detector Concept*, <http://www.ilcldc.org/>, 2008.
- [Mar84] W. Marciniak, *Przyrządy półprzewodnikowe i układy scalone*, Wydawnictwo Naukowo-Techniczne, 1984.
- [Mic07] Austria Microsystems, *0.35 μm CMOS C35 Process Parameters*, 2007.
- [PRAB96] M.J. Paulus, J.M. Rochelle, M.S. Andreaco, i D.M. Binkley, *A low-noise, wide-band CMOS charge-sensitive preamplifier for use with APD/LSO PET detectors*, *IEEE Transactions on Nuclear Science* (1996).
- [Raz00] Behzad Razavi, *Design of Analog CMOS Integrated Circuits*, McGraw-Hill, 2000.
- [RCN03] Jan M. Rabaey, Anantha Chandrakasan, i Borivoje Nikolic, *Digital Integrated Circuits*, Printice Hall Electronics and Vlsi Series, 2003.
- [Ric03] Lyons Richard, *Wprowadzenie do Cyfrowego Przetwarzania Sygnałów*, Wydawnictwo Komunikacji i Łączności, 2003.
- [Sum02] Lauri Sumanen, *Pipeline Analog-to-Digital Converters for Wide-Band Wireless Communications*, praca doktorska, Helsinki University of Technology Electronic Circuit Design Laboratory, 2002.
- [SWH00] L. Sumanen, M. Waltari, i K. Halonen, *A mismatch Insensitive CMOS Dynamic Comparator for Pipeline A/D Converters*, Proc. ICECS'00 (2000).

- [Tex04] Texas Instruments, *Nota aplikacyjna układu SN74LVC245A*, 2004.
- [Tex08] Texas Instruments, *Nota aplikacyjna układu THS4505*, 2008.
- [vdP01] Rudy van de Plassche, *Scalone przetworniki analogowo-cyfrowe i cyfrowo-analogowe*, Wydawnictwa Komunikacji i Łączności, 2001.
- [Zby07] Piotr Zbysiński, *Układy programowalne w przykładach*, BTC, 2007.
- [Zie02] Tomasz Zieliński, *Od teorii do cyfrowego przetwarzania sygnałów*, Wydział EAIiE AGH, 2002.
- [ZP04] Piotr Zbysiński i Jerzy Pasierbiński, *Układy programowalne, pierwsze kroki*, BTC, 2004.

Spis rysunków

1.1. Skala energetyczna pokazująca unifikację oddziaływań elektromagnetycznych i oddziaływań słabych w zakresie energii teraelektronowoltowych	12
1.2. Schemat projektu eksperymentu ILC	13
1.3. Schemat ideowy detektora LDC	14
1.4. Schemat ideowy przedniej części detektora LDC	15
1.5. Projekt mechaniczny detektora LumiCal	15
1.6. Przekrój poprzeczny wzdłuż osi detektora LumiCal	16
2.1. Schemat toru elektroniki odczytu	18
2.2. Model złącza p^+n	19
2.3. Proponowana segmentacja sensora krzemowego	20
2.4. Schemat zastępczy krzemowego sensora promieniowania wraz z elektroniką odczytu.	21
2.5. Schemat ideowy przedwzmacniacza ładunkoczułego	22
2.6. Przykładowe odpowiedzi przedwzmacniacza dla różnych wartości rezystancji sprzężenia zwrotnego	23
2.7. Schematy rozładowywania pojemności sprzężenia zwrotnego wzmacniacza ładunkoczułego	24
2.8. Zależność rozrzutu wzmocnienia ładunkowego w funkcji wzmocnienia wzmacniacza	26
2.9. Zależność impedancji wejściowej od częstotliwości	27
2.10. Efekt przesłuchów	27
2.11. Zastępczy schemat szumowy wzmacniacza ładunkoczułego	28
2.12. Zastępczy schemat szumowy wzmacniacza ładunkoczułego wraz z blokiem filtru o transmitancji $H(s)$	29
2.13. Schemat filtru (CR)-(RC)	30
2.14. Schemat układu elektroniki front-end z ciągłym kształtowaniem	32
2.15. (a) Funkcja przenoszenia idealnego przetwornika 3-bitowego (b) zależność błędu kwantyzacji od napięcia wejściowego	36
2.16. Architektura potokowego przetwornika analogowo-cyfrowego.	37
2.17. Schemat pojedynczego stopnia przetwornika potokowego.	38
2.18. Fazy działania układu mnożąco-odejmującego (MDAC)	38

2.19. Funkcje przenoszenia dla stopnia (a) 1-bitowego (b) 1,5-bitowego oraz (c) 2-bitowego	39
2.20. Wizualizacja funkcji przenoszenia dla stopnia 1-bitowego	40
2.21. Wizualizacja funkcji przenoszenia dla stopnia 1,5-bitowego	40
2.22. Algorytm korekcji cyfrowej.	41
3.1. Schemat przedwzmacniacza	44
3.2. Uproszczony schemat przedwzmacniacza	45
3.3. Schemat kanału odczytowego zrealizowanego w oparciu o resetowany przedwzmacniacz	46
3.4. Wyniki symulacji wzmocnienia oraz fazy wzmacniacza.	47
3.5. Symulacje wpływu prądu referencyjnego I_{bias} na parametry układu wzmacniacza	47
3.6. Symulowana odpowiedź czasowa układu przedwzmacniacza dla różnych ładunków wejściowych ($C_d = 55$ pF)	48
3.7. Symulowana odpowiedź czasowa układu przedwzmacniacza dla różnych ładunków pojemności wejściowych	49
3.8. Symulowane wartości wzmocnienia ładunkowego dla zestawu pojemności C_d	49
3.9. Schemat wykorzystywany do badania stabilności	50
3.10. Symulowany margines fazy układu przedwzmacniacza w funkcji pojemności C_d dla różnych trybów pracy	50
3.11. Przykładowe symulowane widma szumów na wyjściu przedwzmacniacza . . .	51
3.12. (a) wartość szumów na wyjściu przedwzmacniacza w funkcji pojemności C_d dla różnych trybów pracy (b) stosunek sygnału ($4 fC$) do szumu w funkcji pojemności C_d dla różnych trybów pracy	51
3.13. Schemat struktury zabezpieczającej wejście układu	52
3.14. Schemat tranzystora wielopalczystego	53
3.15. Plan masek topologicznych zaprojektowanego układu scalonego z elektroniką front-end	54
3.16. Widok z góry prototypowego układu elektroniki front-end.	55
3.17. Stanowisko pomiarowe wykorzystywane do testowania przedwzmacniacza z kluczowanym resetem	55
3.18. Przykładowe sekwencje sygnałów używane do testowania przedwzmacniacza z kluczowanym resetem	56
3.19. Widok perspektywiczny prototypowego układu elektroniki front-end.	56
3.20. Odpowiedzi układu na różne ładunki wejściowe ($C_d=44$ pF, mode=0). Prezentowane rezultaty nie zawierają składowej stałej	57

3.21. Porównanie wyników pomiarów z symulacjami odpowiedzi czasowych dla różnych pojemności obciążenia przedwzmacniacza	58
3.22. Wzmocnienie ładunkowe w funkcji pojemności sensora dla różnych trybów pracy	58
3.23. Wyniki procedur CDS	60
4.1. Metoda histogramowa	64
4.2. Przykład dyskretnej transformaty Fouriera	66
4.3. Przykład wycieku przy próbkowaniu częstotliwości nie będącej częstotliwością bazową	67
4.4. Przykład zjawiska aliasingu	68
4.5. Wpływ wybranej częstotliwości sygnału wejściowego na testowalność przetwornika	69
4.6. Schemat ideowy stopnia 1,5-bitowego	71
4.7. Schemat w pełni różnicowego wzmacniacza z podbijanym wzmocnieniem	71
4.8. Schemat komparatora w konfiguracji zatrzasku dynamicznego (ang. latched comparator)	72
4.9. Plan masek zaprojektowanego układu scalonego przetwornika analogowo-cyfrowego	73
4.10. Widok z góry prototypowego układu ADC	74
4.11. Widok perspektywiczny prototypowego układu ADC	74
4.12. Schemat blokowy systemu pomiarowego	75
4.13. Przykładowe przebiegi sygnału wejściowego oraz zegara podczas pomiarów dynamicznych	75
4.14. Przepływ danych w potokowym przetworniku analogowo-cyfrowym	76
4.15. Pomiar statyczny funkcji przenoszenia ADC	77
4.16. Pomiar statyczny nieliniowości różniczkowej	77
4.17. Pomiar statyczny nieliniowości całkowitej	78
4.18. Przykładowa transformata Fouriera	78
4.19. Pomiary dynamiczne w funkcji częstotliwości próbkowania	79
4.20. Pomiary dynamiczne w funkcji częstotliwości sygnału wejściowego	80
4.21. Pomiary dynamiczne w funkcji amplitudy sygnału wejściowego. Częstotliwość próbkowania 35 Msps. Częstotliwość sygnału wejściowego 3,4 MHz	80
5.1. Schemat systemu pomiarowego	83
5.2. Rozmieszczenie elementów na płytce PCB	85

5.3. Fotografia płytki testowej z osadzonymi elementami do testowania elektroniki front-end	86
5.4. Fotografia płytki testowej z osadzonymi elementami do testowania struktur ADC	86
5.5. Bufory na wyjściu kanałów frontend	87
5.6. Układ wstrzykiwania ładunku wraz z proponowanymi kombinacjami zworek . .	87
5.7. Konwerter sygnału pojedynczego na różnicowy	88
5.8. Fotografia zestawu ewaluacyjnego układu przetwornika analogowo-cyfrowego AD9445 wraz konwerterami	90
5.9. Schemat blokowy systemu akwizycji danych wraz z oznaczonym przepływem danych	91
5.10. Schemat realizacji systemu akwizycji danych	93
5.11. Algorytm wykonywania programu w mikrokontrolerze	95
5.12. Schemat przepływu danych w systemie akwizycji danych	98
5.13. Topologia sieci GPIB	99
5.14. Schematyczny diagram klas dla urządzeń wykorzystujących interfejs GPIB . .	99
A.1. Schemat części układu służącej do pomiarów ADC	105
A.2. Schemat części układu służącej do pomiarów elektroniki front-end	106
A.3. Schemat części zasilającej	107
A.4. Plan masek płytki drukowanej	108

Spis tablic

3.1. Konfiguracje sprzężenia zwrotnego projektowanego układu	46
5.1. Spis pinów układu szybkiej logiki sterującej	94