



AGH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Faculty of Physics and Applied Computer Science

Doctoral dissertation

Mirosław Firlej

SALT readout ASIC for LHCb upgrade experiment – clock generation and data transmission

Supervisor: **prof. dr hab. inż. Marek Idzik**

Krakow, July 2015

Declaration of the author of this dissertation:

Aware of legal responsibility for making untrue statements I hereby declare that I have written this dissertation myself and all the contents of the dissertation have been obtained by legal means.

(mgr inż. Mirosław Firlej)

Declaration of the dissertation Supervisor:

This dissertation is ready to be reviewed.

(prof. dr hab. inż. Marek Idzik)

Acknowledgements

Many people contributed to the completion of this dissertation. I would like first to thank my supervisor prof. dr hab. inż. Marek Idzik for his thoughtful guidance, an enormous patience and constant motivation. His immense knowledge, time spent on the discussions and invaluable advices helped me in my research and writing of this dissertation.

I would like to thank my friends from the Nuclear Electronics and Radiation Detection Group, especially Krzysztof Świentek, Tomasz Fiutowski, Jakub Moroń, who worked with me on the SALT project. Their support in various research areas was very helpful.

Special thanks to Szymon Kulis and Przemysław Terlecki for the design of Printed Circuit Boards and for many invaluable advices.

Lastly, and most importantly, I wish to thank my wonderful wife Wiesia, for her love, patience and constant motivation. Without her support this dissertation would not have been possible. To her I dedicate this dissertation.

This PhD dissertation has been completed within the framework of the Human Capital Operational Program POKL.04.01.01-00-434/08-02 co-financed by the European Union. This work was also supported by the National Science Centre Poland under contract nr UMO-2012/07/B/ST7/01456.

Streszczenie

Fizyka cząstek zajmuje się badaniem natury składników materii i promieniowania. Model Standardowy (ang. Standard Model) opisuje fundamentalne składniki materii oraz ich oddziaływania. Istnieje 6 leptonów, 6 kwarków oraz odpowiadające im antycząstki. Wszystkie te cząstki fundamentalne są fermionami i posiadają spin połówkowy ($1/2$). Współczesne badania w tej dziedzinie fizyki skupiają się na subatomowych cząstkach oraz składnikach atomów: elektronach, barionach (protonach i neutronach zbudowanych z kwarków); a także tych wyprodukowanych przez procesy rozpraszania np.: fotonach, neutrinach i mionach [1]. Fizyka cząstek podzielona jest na dwie dziedziny. Pierwsza z nich, nazywana fizyką nie-akceleratorową, skupia się na detekcji cząstek produkowanych w naturalnych procesach i ich oddziaływaniach. Dobrym przykładem badań w tej dziedzinie fizyki jest detekcja promieniowania kosmicznego, którego natura dostarcza cząstek o ogromnych energiach, przewyższających te możliwe do uzyskania na Ziemi, jednakże promieniowanie kosmiczne jest przypadkowe, a także ma znacznie mniejszą intensywność niż wiązka w akceleratorze. To właśnie rosnąca potrzeba stworzenia promieniowania podobnego do promieniowania kosmicznego w kontrolowanych warunkach prowadzi do drugiej gałęzi fizyki cząstek jaką są nowoczesne eksperymenty fizyki wysokich energii (ang. High Energy Physics) [2].

Eksperymenty fizyki wysokich energii są zawsze bardzo dużymi i skomplikowanymi systemami, które produkują ogromne ilości danych pomiarowych. W rezultacie istnieje rosnąca potrzeba odczytu danych z układów o ogromnej liczbie kanałów, często sięgającej milionów, a nawet znacznie więcej. Obecnie przetwarzanie sygnałów elektrycznych jest dość podobne w wielu eksperymentach, składa się ono z: analogowej ekstrakcji i wstępnego przetwarzania sygnału, konwersji na postać cyfrową, cyfrowego przetwarzania sygnału DSP (ang. Digital Signal Processing) oraz szybkiej serializacji i transmisji danych. W związku z ciągle rosnącą gęstością i liczbą kanałów odczytowych, istnieje ciągła presja aby zredukować moc pobieraną przez wielokanałowe układy odczytu. Zatem uzyskanie niskiego poboru mocy jest podstawowym zagadnieniem przy rozwoju przyszłych wielokanałowych układów odczytu.

Eksperymentem fizyki wysokich energii, w ramach którego wykonywana jest ta rozprawa, jest LHCb (ang. Large Hadron Collider beauty), będący jednym z czterech eksperymentów-detektorów umieszczonych wokół pierścienia Wielkiego Zderzacza Hadronów LHC (ang. Large Hadron Collider). Obecny system trygera w LHCb zawiera dwa poziomy:

- na pierwszym poziomie tryger sprzętowy redukuje częstotliwość zdarzeń z wartości nominalnej dla LHC (40 MHz) do nie większej niż 1.1 MHz;
- na drugim poziomie pracuje tryger programowy, odpowiedzialny za obróbkę danych powstałych z odczytu całego detektora ze zredukowaną częstotliwością.

Strumień danych, ograniczony przez trygery, limituje dokładność pomiarów osiąganą przez LHCb. Z tego powodu konieczna jest modernizacja systemu trygera LHCb [3], która odbędzie się w czasie przerwy w pracy akceleratora LHC, trwającej od połowy 2018 do końca 2019 roku. Modernizacja pozwoli na znaczną poprawę możliwości fizycznych eksperymentu LHCb, jednak wymagać to będzie

wymiany między innymi systemów odczytu detektorów śladowych (ang. Tracker System), które będą musiały umożliwić zbieranie danych z częstotliwością 40 MHz i wysyłać je do systemu akwizycji w celu dalszej analizy.

Głównym tematem tej rozprawy jest projekt nowego systemu odczytu dla detektora UT (ang. Upstream Tracker), będącego jednym z detektorów śladów w zmodernizowanym eksperymencie LHCb. Projektowany system elektroniki odczytu, nazywany dalej SALT (ang. Silicon ASIC for LHCb Tracking), budowany jest w postaci dedykowanego układu scalonego ASIC (ang. Application Specific Integrated Circuit). Jest to prawdopodobnie pierwszy na świecie wielokanałowy układ scalony, o tak niskim poborze mocy, do zastosowań w dziedzinie fizyki cząstek, zawierający kompletny tor odczytowy: począwszy od elektroniki front-end, poprzez szybkie przetworniki analogowo-cyfrowe ADC (ang. Analog to Digital Converters) w każdym kanale odczytowym, a skończywszy na cyfrowym przetwarzaniu sygnału DSP i szybkiej serializacji i transmisji danych. W rozwoju tak zaawansowanego układu uczestniczy wiele osób, zarówno w procesie projektowania jak i późniejszych testach prototypów układów ASIC. Za projekt elektroniki odczytu SALT odpowiada grupa z katedry oddziaływań i detekcji cząstek WFiIS AGH, zaś autor jest przede wszystkim odpowiedzialny za projekt i pomiary układów PLL (ang. Phase-Locked Loop) i DLL (ang. Delay-Locked Loop), wchodzących w skład układu SALT. Wielofazowy układ PLL dla modułów serializatora i deserializatora w systemie odczytu SALT zapewnia powielanie częstotliwości przebiegu zegarowego oraz jego przesuwanie w fazie, co jest konieczne dla poprawnego odbierania danych przez ASIC. Detektor UT posiada sensory o różnych geometriach, a co za tym idzie o różnych pojemnościach, co bezpośrednio wpływa na kształt impulsu (peaking time) elektroniki front-end. Z tego powodu konieczny jest dedykowany układ DLL, pozwalający na dopasowanie fazy przebiegu zegarowego i zapewnienie poprawnego próbkowania sygnału (w odpowiedniej fazie) za pomocą szybkich przetworników analogowo-cyfrowych.

Równoległym celem niniejszej rozprawy jest zaprojektowanie w niedalekiej przyszłości, dużo szybszego niż w projekcie SALT i o bardzo niskim poborze mocy, układu do serializacji i transmisji danych, zdolnego do pracy z częstotliwościami znacznie większymi od 1 GHz. W dziedzinie fizyki cząstek byłby to znów pierwszy na świecie układ charakteryzujący się bardzo dużą przepustowością danych (5 Gb/s – 10 Gb/s) z jednoczesnym bardzo niskim poborem mocy (~ 15 mW). W tym celu autor zaprojektował układ PLL ogólnego przeznaczenia (nazwany MULTI_PLL) i przeprowadził prace badawczo-rozwojowe pod kątem jego zastosowania w bardzo szybkich serializatorach danych. MULTI_PLL został zaprojektowany i użyty przy serializacji danych z wielokanałowych 6-bitowych i 10-bitowych przetwornikach ADC, z których ten ostatni ma działać w systemie odczytu detektora LumiCal (ang. Luminosity Calorimeter) przy akceleratorze ILC (ang. International Linear Collider).

Rozprawa prezentuje pracę autora począwszy od projektu układu PLL ogólnego przeznaczenia, poprzez projekty układów PLL i DLL dedykowanych dla systemu odczytu SALT, kończąc na kompletnych pomiarach i parametryzacji wszystkich zaprojektowanych układów ASIC. Prace projektowe były prowadzone w dwóch różnych technologiach krzemowych 130 nm CMOS (ang. Complementary Metal-Oxide Semiconductor), które nazwano odpowiednio technologią A i technologią B, w celu ochrony poufnych danych technologicznych. Oprócz projektu układów PLL i DLL, które stanowią główny wkład autora w rozwój systemu odczytu SALT, uczestniczył on również w procesie projektowania innych bloków funkcjonalnych tego systemu (np. konwerter sygnału unipolarnego na różnicowy). Tekst rozprawy opisuje głównie pracę autora, jednak dla zachowania ciągłości i spójności rozprawy przedstawiono szerszy opis detektora LHCb wraz z planowaną modernizacją oraz krótki opis całego systemu odczytu SALT.

W pierwszym rozdziale przedstawiony został ogólny opis eksperymentu LHCb. W wielkim skrócie opisane zostały poszczególne podsystemy detekcyjne. Cele fizyczne eksperymentu pozwalają na szukanie nowej fizyki w łamaniu symetrii CP i rzadkich rozpadach hadronów, w skład których wchodzi kwarki b i c. Dalsza część rozdziału poświęcona jest modernizacji detektora oraz omówieniu

powodów, które doprowadziły do takiej konieczności. Duży nacisk położono na prezentację detektora UT, który ma zastąpić obecnie działający detektor TT (ang. Trigger Tracker) oraz prezentację jego kluczowych parametrów. Na końcu rozdziału zamieszczono wprowadzenie do architektury systemu odczytu SALT oraz krótki opis jego najważniejszych parametrów.

Drugi rozdział opisuje zagadnienia teoretyczne związane z generacją przebiegów zegarowych i transmisją danych. Na początku wyjaśnione zostały różnice między transmisją szeregową i równoległą, co bezpośrednio prowadzi do omówienia zagadnienia serializacji danych. Teoretyczna analiza układów PLL i DLL ujęta jest w dalszej części tego rozdziału. Zaprezentowane zostały matematyczne modele wspomnianych układów i ich analiza oraz poddano dyskusji problemy stabilności układów PLL i DLL. Wskazano różnice między układem PLL I-go i II-go rodzaju. Na końcu tego rozdziału przedstawiono analizę bloków funkcjonalnych używanych przy budowie PLL i DLL.

Projekty bloków PLL i DLL dla systemu odczytowego SALT i innych zastosowań ogólnego przeznaczenia przedstawione zostały w rozdziale trzecim. Zaprezentowany został także krótki wstęp do technologii i projektowania układów ASIC, omówiono również proces prototypowania układów scalonych. Następnie przedstawiono projekt i symulacje układu PLL (MULTI_PLL) ogólnego przeznaczenia wraz ze szczegółowymi symulacjami jego bloków funkcjonalnych. Na szczególną uwagę zasługuje tutaj, zaproponowany przez autora, układ automatycznego przełączania zakresów pracy PLL-a, nazwany AFMS (ang. Automatic Frequency Mode Setting). Układ ten umożliwi znaczne rozszerzenie zakresu częstotliwości pracy układu PLL. Dalsza część tego rozdziału skupia się nad układami PLL i DLL dedykowanymi do systemu odczytu SALT. Oba układy mają podobne bloki funkcjonalne, a część z nich bazuje na tych zaprojektowanych wcześniej dla układu MULTI_PLL, zatem szczegółowo przedstawiono tylko najważniejsze ich elementy.

W ostatnim rozdziale przedstawione zostały stanowiska pomiarowe i wyniki pomiarów dla wyprodukowanych prototypowych układów scalonych ASIC. We wstępie do rozdziału opisana została metodologia pomiaru drżenia zegara (ang. jitter) i sposoby obliczania poszczególnych jego typów. Drżenie zegara jest jednym z najważniejszych parametrów układów cyfrowych, pozwalającym na dobranie ich marginesów czasowych, zatem główna część tego rozdziału poświęcona jest właśnie efektowi drżenia zegara. W rozdziale tym przedstawiono także prototypowe płytki drukowane PCB (ang. Printed Circuit Boards), które zapewniają połączenia elektryczne prototypowych układów ASIC oraz zawierają najważniejsze elementy zewnętrzne umożliwiające ich pracę. W rozdziale czwartym zaprezentowano też wolne interfejsy szeregowo do konfiguracji układów ASIC oraz konfiguracje stanowisk pomiarowych wraz z wymaganym sprzętem laboratoryjnym, które w efekcie prowadzą do wyników pomiarów przedstawionych na końcu tego rozdziału.

Wszystkie projekty masek technologicznych były wykonane ręcznie, bez użycia narzędzi do automatycznego projektowania elektroniki cyfrowej. Pozwoliło to na redukcję powierzchni zajmowanej przez układy PLL i DLL oraz na zmniejszenie pobieranej przez nie mocy. Taki sposób projektowania jest konieczny przy budowie bloków takich jak oscylator sterowny napięciem VCO (ang. Voltage-Controlled Oscillator) czy też sterowana napięciem linia opóźniająca VCDL (ang. Voltage-Controlled Delay Line), gdzie pojemności pasożytnicze bardzo łatwo degradują działanie układu. Prace autora obejmują wszystkie dziedziny związane z prototypowaniem układów scalonych ASIC, ale nie wszystkie zostały przedstawione w niniejszej rozprawie w sposób wyczerpujący. Poza teoretycznymi analizami, projektami i symulacjami układów prototypowych i ich parametryzacją, autor opracował zaawansowane stanowiska pomiarowe wraz z dedykowanym oprogramowaniem, potrzebnym do zbierania danych i ich analizy. Autor uczestniczył także w procesie projektowania płytek drukowanych. Podczas długiej pracy badawczo rozwojowej autor zgromadził unikalne doświadczenie w budowaniu systemów odczytu dla detektorów fizyki cząstek, a w szczególności w projektowaniu i symulacjach układów do generacji przebiegów zegarowych i transmisji danych (zawierających PLL i DLL).

Contents

Contents	9
Introduction	11
1 LHCb experiment – present and future	15
1.1 LHCb physics goals	15
1.2 Overview of LHCb experiment	16
1.3 LHCb upgrade and its motivation	20
1.3.1 Upgraded tracker system - Upstream Tracker (UT)	21
1.3.2 UT Silicon Sensors	26
1.4 SALT – readout ASIC for Upstream Tracker	26
1.4.1 Analogue front-end	28
1.4.2 Analog to Digital Converter (ADC)	30
1.4.3 Digital Signal Processing (DSP)	31
1.4.4 Data transmission and clock generation	32
1.4.5 Communication interfaces	32
2 Theoretical issues of clock generation and data transmission in readout systems	33
2.1 Phase-Locked Loop (PLL)	35
2.1.1 Type I Phase-Locked Loop	36
2.1.2 Type II Phase-Locked Loop	40
2.2 Delay-Locked Loop (DLL)	44
2.3 General purpose functional blocks for PLL and DLL	47
2.3.1 Voltage-Controlled Oscillator and Voltage-Controlled Delay Line	47
2.3.2 Phase and Frequency Detector (PFD)	51
2.3.3 Charge Pump (CP) and Low-Pass Filter (LPF)	52
3 Design of phase-locked circuits for SALT and other applications	57
3.1 Design and simulations of MULTI_PLL	58
3.1.1 Voltage-Controlled Oscillator (VCO)	61
3.1.2 Phase and Frequency Detector (PFD)	67
3.1.3 Charge Pump (CP)	70
3.1.4 Frequency Divider	72
3.1.5 Automatic Frequency Mode Setting (AFMS)	74
3.2 Design and simulations of SALT_PLL	77
3.2.1 Voltage-Controlled Oscillator (VCO)	80
3.2.2 Phase and Frequency Detector (PFD)	85
3.2.3 Charge Pump (CP)	87

3.3	Design and simulations of SALT_DLL	88
3.3.1	Voltage-Controlled Delay Line (VCDL)	91
4	Measurements results	95
4.1	Methodology of jitter measurements	96
4.1.1	Period jitter	97
4.1.2	Cycle to cycle jitter	97
4.1.3	Long term jitter	98
4.2	Measurements of MULTI_PLL	99
4.2.1	Slow control interface - ASIC configuration	100
4.2.2	Details of measurement setup	102
4.2.3	Measurements results	104
4.3	Measurements of SALT_PLL and SALT_DLL	107
4.3.1	Slow control interface - ASIC configuration	109
4.3.2	Details of measurement setup	111
4.3.3	Measurements results of SALT_PLL	114
4.3.4	Measurements results of SALT_DLL	117
	Summary	121
	Acronyms	125
	Bibliography	127
	List of Figures	133
	List of Tables	135
A	Measurements results of MULTI_PLL	137
B	Measurements results of SALT_PLL	143
C	Measurements results of SALT_DLL	149

Introduction

Particle physics studies the nature of particles which are constituents of matter and radiation. The Standard Model (SM) describes fundamental constituents of matter and their interactions. There are 6 leptons, 6 quarks and their antiparticles. All these fundamental particles are fermions and have spin equal $1/2$. Modern research in particle physics area is focused on subatomic particles, including atomic constituents: electrons, baryons (protons and neutrons, which are made of quarks) and produced by scattering processes: photons, neutrinos, muons, etc. [1]. Particle physics is divided into two branches. First, called non-accelerator physics, is focused on detection of particles produced in natural processes and their interactions. A good example of no-accelerator physics is the detection of cosmic rays which nature provides with very high energies, far beyond that can be obtained on Earth. The cosmic rays are random, and have much less intensity than beams made at accelerators. A growing need to create rays similar to cosmic rays under controlled conditions led to the second branch of particle physics - modern High Energy Physics (HEP) experiments [2].

The HEP experiments are always very large systems which produce a large amount of data. In result there is a growing need to read the data from systems with large total number of channels, sometimes ranging to several millions or even much more. Nowadays the electrical signal processing is quite similar for many experimental areas, consisting of signal extraction and shaping by an analog electronics, followed by an analog-to-digital conversion and digital data processing, and finally by fast data serialization and transmission. Because of an unceasing increase in the number of channels, there is a continuous pressure to lower the power dissipation of multi-channel readout circuits.

The Large Hadron Collider beauty (LHCb) is one of the four particle experiments located around the Large Hadron Collider (LHC) ring. Although LHCb has been delivering recently a very high quality data, the present experimental sensitivity cannot be significantly improved just by collecting more data (statistical uncertainty) with the present trigger system. The current LHCb trigger system consists of two levels. The hardware trigger (first level) reduces the event rate from the nominal LHC (40 MHz) to a maximum of 1.1 MHz. The complete detector is read with the reduced rate and the data is processed by the software trigger. The reduced data rate in the trigger system limits the precision which can be achieved by the experiment. To overcome this bottleneck the upgrade of the LHCb trigger system was proposed to allow a fully software trigger with the nominal LHC (40 MHz) rate [3]. For this upgrade various detectors and their readout systems will need to be replaced or/and re-designed. The new readout systems will be able to collect complete events every 25 ns and send it to the LHCb data acquisition farm in order to apply a fully software trigger. The upgrade will take place from mid 2018 to the end of 2019.

The main objective of this dissertation is to design a new readout system for the Upstream Tracker (UT) detector of the LHCb experiment, called Silicon ASIC for LHCb Tracking (SALT). The SALT is a low-power, multi-channel (128) Application Specific Integrated Circuit (ASIC) with architecture comprising a front-end electronics and an Analog to Digital Converter (ADC) in each channel, followed by a Digital Signal Processing (DSP), and subsequently by a fast data serialization and transmission. It is probably the first ASIC in the world, designed for HEP application, comprising a fast

sampling and ADC conversion in each channel, with so low power consumption. The ADC power consumption is much smaller than the analog front-end one and the key clock generation and data serialization blocks (Phase-Locked Loop (PLL), Delay-Locked Loop (DLL)) have the lowest power consumption ever seen in HEP readout systems. In such complex system many people are involved in the design process and measurements of prototype ASICs. The SALT is designed by AGH-UST group from the Department of Particle Interactions and Detection Techniques. The author has been actively, and from the very beginning, participating in creating the SALT concept, and subsequently in elaboration of various ideas into a realistic readout ASIC architecture. From then he has taken part in the design, test setup and software preparation, and measurements of various prototype ASICs. In particular the author is also participating in the development of the analog front-end. However, the main author's responsibility is focused on clock generation, its phase alignment, and fast data serialization circuitry. These functionalities depend on two crucial blocks, namely on the PLL and DLL circuits, which were designed by the author. Because of limited space author does not describe in details his works on other parts of the SALT project but concentrates on the PLL and DLL contributions to the readout. The multi-phase PLL is used for the SALT serializer and deserializer circuits, providing clock multiplication and phase shifting, which is needed for proper data transmission. A dedicated DLL circuit is used to align the ADC sampling phase with the experimental clock. Since the UT provides various sensor geometries with different capacitances, which directly affect the peaking time of the front-end, this alignment will vary for different UT sensors.

The parallel objective of this dissertation is to design, in the nearby future, a very fast data serializer and transmitter, able to work at frequency much higher than 1 GHz (and much higher than in the SALT). Such solution would simplify the architecture of future readout systems processing data streams significantly higher than in the SALT. In fact one of the application could be the multi-channel readout system for the luminosity detector at future International Linear Collider (ILC), where a 10-bit ADC will be placed in each readout channel. For this aim author designed also a general purpose PLL (called MULTI_PLL) which will be the main block of such high frequency future serializer. The first versions of MULTI_PLL were integrated in the prototypes of multi-channel 6-bit ADCs for SALT and 10-bit ADCs for luminosity detector at ILC, called Luminosity Calorimeter (LumiCal). In both cases the MULTI_PLL works as the clock multiplier.

In the first chapter of this dissertation the present LHCb experiment is introduced. Its physics goals which help us to search for new physics in CP violation and rare decays of beauty and charm hadrons are presented. The overview of the LHCb detector system, which is specially designed to filter out B mesons and the products of their decay is also presented [4, 5]. This chapter is mainly focused on the LHCb upgrade and its motivation, especially on a new trigger system (UT) and its new dedicated readout electronics (SALT). In the end a short description of the SALT and its main components is presented. As already mentioned, although the author focuses in this dissertation on the PLLs and DLL circuits, he has participated actively in the design and measurements of the other functional blocks of the SALT project.

The second chapter describes theoretical issues of the clock generation and data transmission. At the beginning the differences between parallel and serial transmission, as well as principle of data serialization process, are presented. The theoretical analysis of PLL and DLL circuits is presented in further part of this chapter. A mathematical models of two PLL types (type I and II) and the differences between them are discussed. At the end of this chapter a general purpose functional blocks for PLL and DLL are described in details.

The design of integrated circuits for SALT and other applications is presented in chapter three. A short introduction to ASIC technology is given at the beginning. The design works were performed in two different 130 nm Complementary Metal-Oxide Semiconductor (CMOS) technologies, called 130 nm CMOS technology A and 130 nm CMOS technology B to keep their parameters confidential.

The design and simulations of the general purpose PLL (MULTI_PLL) and its main functional blocks are presented. The further part of the chapter is focused on PLL and DLL dedicated for the SALT project.

In the last chapter the test setups description and the measurements results of all fabricated ASICs are presented. A short introduction to measurements and methodology of jitter calculation is given. The jitter is one of the most important parameters for operation of electronic circuits and is very useful in calculating timing margins in digital systems, consequently the main part of this chapter is focused on the jitter. This chapter describes also a prototype Printed Circuit Boards (PCBs) which provide electrical connections and the most important components supporting the prototype ASICs. The slow control interfaces of all measured chips are presented. The setup configuration with all needed laboratory equipment is described. Finally, the most important, the measurements performed on the prototype PLLs and DLL circuits and their results are discussed.

Chapter 1

LHCb experiment – present and future

There are many High Energy Physics (HEP) accelerators, but the Large Hadron Collider (LHC) is the largest and most powerful particle collider in the world, built by the Conseil Européen pour la Recherche Nucléaire (CERN). The LHC is placed in underground tunnel (27 km long) and takes a form of ring consisting of superconducting magnets and accelerating structures. Two accelerated (close to the speed of light) high-energy particle beams travel in opposite directions in separate beam pipes. Two tubes must keep ultrahigh vacuum, as empty as interplanetary space, to avoid colliding with gas molecules inside the accelerator. The coils of electromagnets and electric cables are made of superconducting material, efficiently conducting electricity without loss of energy. For this reason the magnets and cables are cooled to temperature $-271.3\text{ }^{\circ}\text{C}$ by the complex liquid helium distribution system [6]. The LHC beams can collide at four locations around the accelerator ring, which are related to the positions of four particle detectors: A Toroidal LHC Apparatus (ATLAS), Compact Muon Solenoid (CMS), A Large Ion Collider Experiment (ALICE) and Large Hadron Collider beauty (LHCb).

The first two (ATLAS and CMS) are general purpose detectors, which investigate a wide range of physics starting from Standard Model (SM) and Higgs boson to studies of existence of extra dimensions and particles responsible for dark matter creation. Although both experiments have the same scientific goals, they use different technical solutions and a different magnets design [7, 8]. The ALICE is a heavy ion detector, designed to investigate the physics of strongly interacting matter at extreme energy densities - quark-gluon plasma. The LHC provides lead ion collisions, which generate very high temperatures, 100000 times greater than the center of the Sun. Each collision recreates condition similar to those just after the Big Bang. The quarks inside protons and neutrons are released from their bonds with the gluons, which leads to phase of the matter called quark-gluon plasma. The ALICE observes how the quark-gluon plasma expands and cools [9]. The last experiment (LHCb), the most important for this dissertation, searches slight differences between matter and antimatter by detection of particles called beauty quarks. The LHCb uses series of detectors to catch forward particles, which are thrown by the collision in one direction. The subdetectors mounted over a length of 20 m are placed one behind the other. The first of them is mounted close to the collision point to detect quarks before they decay into other forms. Around 700 scientists from 66 different institutes and universities work for the LHCb collaboration.

1.1 LHCb physics goals

The LHCb is dedicated to heavy flavour physics and its main goal is to search new physics in CP violation and rare decays of beauty and charm hadrons [4, 5]. These processes can be studied by looking for the new particles precisely predicted in the SM. The Cabibbo-Kobayashi-Maskawa (CKM)

matrix [10, 11] describes quark mixing and explains a source of CP violation in the SM, but its level in weak interactions cannot explain the matter-antimatter asymmetry in the universe, so new sources of CP violation beyond the SM are needed, which might be seen in heavy flavour physics [12]. Some models predict decay modes, forbidden in the SM, but to check such possibilities, the CP violation and rare decays of hadrons containing b and c quarks, with much higher statistics, must be studied.

During 2011 and 2012 the LHC collected around 10^{12} heavy flavour decays, thanks to the large beauty and charm production cross-sections, which are approximately a factor 10 (for beauty) and 200 (for charm) smaller than the total cross-section at the LHC for energies 7–8 TeV [13]. To separate the decays of interest from the background, a displaced vertex and a high transverse momentum signatures are used. Excellent vertex resolution is required to measure the impact parameter and to achieve a good decay time resolution, which is needed to reject various sources of background. A good momentum and invariant mass resolution are important to reduce the combinatorial background and resolve the decays with similar topologies. The LHCb allows to change the beam focus at its interaction point independently from the other interaction points, so the luminosity in the experiment can be tuned to achieve its optimal value.

The results [14] obtained from the data collected between 2010 and 2013 (LHC Run I) have proven that LHCb is the next generation flavour physics experiment. Thanks to efficient charged particle tracking and dedicated triggers for lepton, hadron and photon signatures, LHCb has the largest sample of exclusively reconstructed charm and beauty decays. The LHCb has already obtained many key results, such as:

- the first evidence for the rare decay $B_s^0 \rightarrow \mu^+\mu^-$ [15, 16] and the measurements of angular distributions in the decay $B^0 \rightarrow K^{*0}\mu^+\mu^-$ [17, 18], which are sensitive to deviations from the SM;
- the measurement of the CP violating phase in the interference of B_s^0 mesons mixing and decay amplitudes, where the value predicted within the SM is small, but much larger values are possible in new physics models. This phase was measured and gave results that are at present consistent with the SM within the uncertainties [19];
- the measurement of the angle γ of the Unitarity Triangle from $B \rightarrow DK$ decays, which is a crucial component in the determination of the parameters of the CKM quark mixing matrix [20].

These just few example LHCb results provide a significant impact on the flavour physics landscape. They show that LHCb covers many of electroweak and Quantum Chromodynamics (QCD) topics, which establish it as a general purpose detector in the forward region at a hadron collider.

1.2 Overview of LHCb experiment

The LHCb detector is a single-arm spectrometer with a forward angular coverage from 10 mrad to 300 (250) mrad in the bending (non-bending) plane. The geometry was chosen based on the fact that at high energies, B and \bar{B} mesons, consisting of the b and \bar{b} quarks, are produced at small angles. The detector, which weights 4500 tons, is specially designed to filter out these particles and the products of their decay [5]. The layout of the LHCb detector is shown in figure 1.1. Most of the subdetectors are splitted into two halves, which can be moved out for assembly detector subparts and provides access to the beam-pipe. Special attention was paid to the materials, which are used especially in tracking system construction, because interactions in the detector material reduce the detection efficiency for electrons and photons. From the same reason a multiple scattering of pions and kaons degrades the momentum resolution, because the track recognition becomes more complicated [21, 22].

The LHCb detector is 20 meters long and it is constructed with sub-detectors (sub-parts), placed one behind each other. Each of the sub-detectors is used to measure different parameters of the particles produced by protons collision. As a result the detector gives information about the identity, trajectory, momentum and energy of each particle, and can give parameters of individual particles from the many that are created in the collision point. The detector consist of following sub-parts: dipole magnet, VERtix LOcator (VELO), tracking system, Ring Imaging CHerenkov (RICH) detectors, calorimeters, and muon detectors [4].

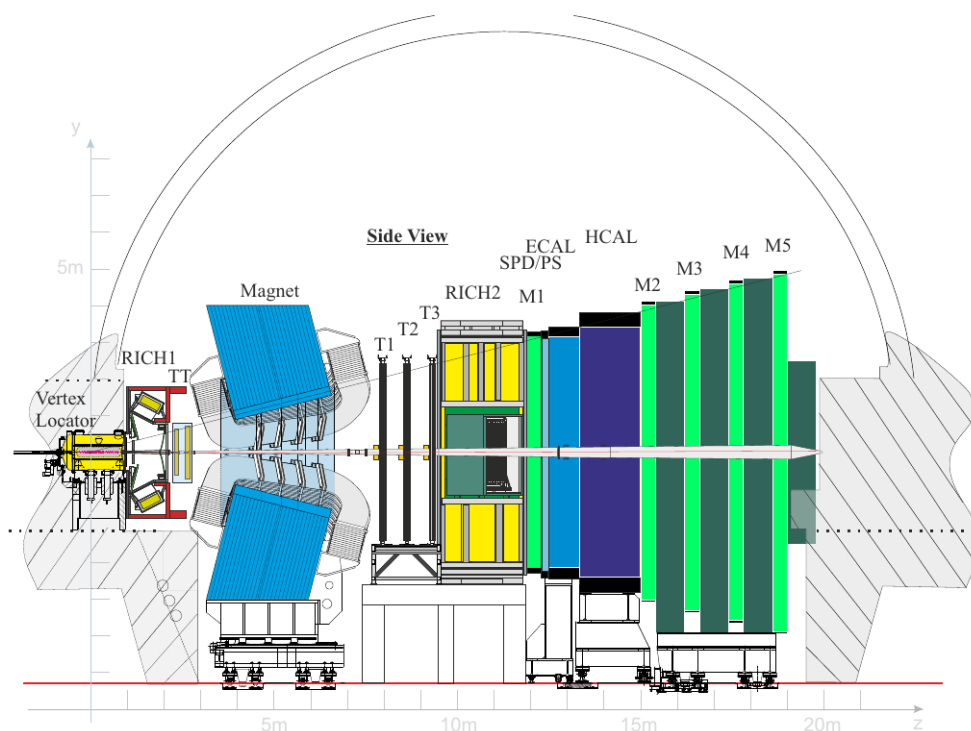


Figure 1.1: Block diagram of the current LHCb detector [23].

The dipole magnet is used for the momentum measurement of charged particles, it provides an integrated field of about 4 Tm, which deflects charged particles in the horizontal plane. The field of the spectrometer magnet disturbs the trajectory of the LHC beams. To compensate this effect and to ensure a closed orbit for the beams the three dipole magnets are used. The total weight of the magnet is about 1600 t [24].

The VELO contains 42 silicon modules placed along the beam, each of them can measure r and ϕ coordinates of the detected particles. The partially constructed VELO is presented in figure 1.2a. The pitch within a module varies from $38 \mu\text{m}$ to $102 \mu\text{m}$ (linearly with increasing radius), while the radius varies from 8.2 mm to 42 mm. To detect B mesons the silicon detector elements must be placed very close to the beam, at a distance of 5 mm. For detector safety, the VELO sensors are spaced apart by 29 mm in horizontal direction during initial stage of geometrical beam optimization. After that the sensors are moved back, using a fully automated procedure, which takes around 210 seconds to close. The VELO measures the distance between the point where B particles are created and the point of their decay. The B particles can not be measured directly, but only by detecting the products of their decay. The VELO detector can locate the B mesons with precision up to $10 \mu\text{m}$.

Tracking system contains four rectangular detectors, called tracking stations: TT, T1, T2, and T3. The Trigger Tracker (TT) consists of four layers of a silicon microstrip detectors. It is around

150 cm wide and 130 cm high, with a total active area of around 8 m^2 and strip pitch around $200 \mu\text{m}$. Figure 1.2b shows **TT** during construction. The **TT** is located upstream of the **LHCb** dipole magnet and covers the full acceptance of the experiment. Each of the trackers T1 – T3, located downstream of the magnet, is built in two different technologies. A silicon microstrip detectors, called Inner Trackers (**ITs**), are placed close to the beam pipe, while a gas-filled straw tubes, called Outer Trackers (**OTs**), are situated further from the beam pipe. The **IT** consists of four layers and covers cross-shaped region in the center of the tracking stations T1 – T3. It is around 120 cm wide, 40 cm high and covers total area around 4 m^2 . The **OT** is placed further from the beam pipe and consists of approximately 200

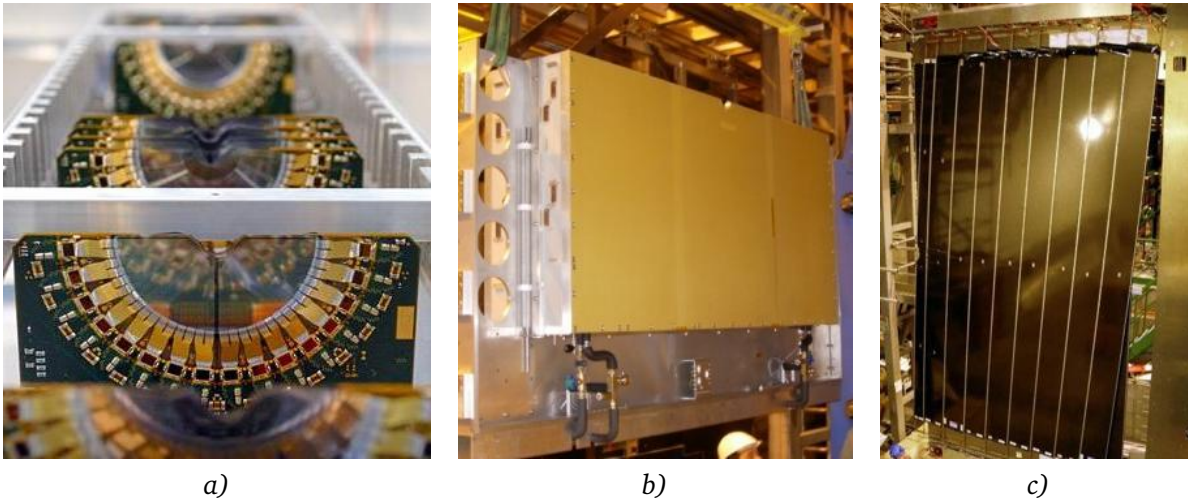


Figure 1.2: Photographs of the **LHCb** subsystems. *a)* - zoom on the partially constructed **VELO**, *b)* - Trigger Tracker detector (bottom half), *c)* - Outer Tracker during installation [25].

gas straw tube modules. Figure 1.2c presents the **OT** during installation. To improve resolution the measurement of the drift time was implemented. When charged particle passes through the tube, the gas molecules are ionized, producing electrons. The position of the track is found by measuring the time the electrons take to reach an anode wire placed in the tube center. The straw tubes, with an inner diameter of 4.9 mm, are arranged in two staggered layers. A mixture of Argon (70%), CO_2 (28.5%) and O_2 (1.5%) is used as a counting gas to achieve a drift time below 50 ns. The spatial resolution of the **OT** is around $200 \mu\text{m}$. The **OT** has four layers placed in the same way as for **IT**. Each of the four trackers (**TT**, T1, T2, and T3) have four detection layers, two of them with vertical strips and two other with strips rotated by angles, -5° and $+5^\circ$, respectively. The total active area of the largest station T3 is around $597 \times 485 \text{ cm}^2$ [26].

The **RICH** detectors are built for particle identification and work by measuring emission of Cherenkov radiation. When the charged particle fly through a dense gas faster than light, a cone of light is emitted, similarly to the sonic boom, generated by object breaking the sound barrier. The light is reflected by the mirror in **RICH** detector onto an array of sensors. The shape of light cone allows to determine the particle speed. The **RICH** detectors, read out by Hybrid Photon Detectors (**HPDs**), can identify charged hadrons in momentum range from 2 GeV/c to 100 GeV/c. There are two **RICH** detectors, the upstream detector (**RICH1**) and the downstream detector (**RICH2**). First of them, located directly behind the **VELO**, uses silica aerogel and C_4F_{10} as radiators and covers the low momentum charged particle range from about 2 GeV/c to 60 GeV/c. Figure 1.3a shows the photomultiplier tubes in the **RICH1** detector. The spherical mirrors of **RICH1** detector are constructed from Carbon-Fibre Reinforced Polymer (**CFRP**) to reduce the scattering of collision products and covers a wide angu-

lar acceptance (same as whole LHCb) from ± 25 mrad to ± 300 mrad (horizontal) and ± 250 mrad (vertical). The RICH2 detects the momentum range from about 15 GeV/c to 100 GeV/c and uses a CF_4 gas radiator. The angular acceptance of RICH2 is limited to range ± 15 mrad – ± 120 mrad (horizontal) and ± 100 mrad (vertical). The spherical mirrors of the RICH2 detector are made of glass and are composed of hexagonal elements.

The calorimeters are designed to stop the particles passing through the detector and to measure their energies and positions. The LHCb has two types of calorimeters: the Electromagnetic CALorimeter (ECAL) for electromagnetic particles like electrons and photons and Hadronic CALorimeter (HCAL) for measuring the energy of particles containing quarks, like proton and neutron. Both calorimeters

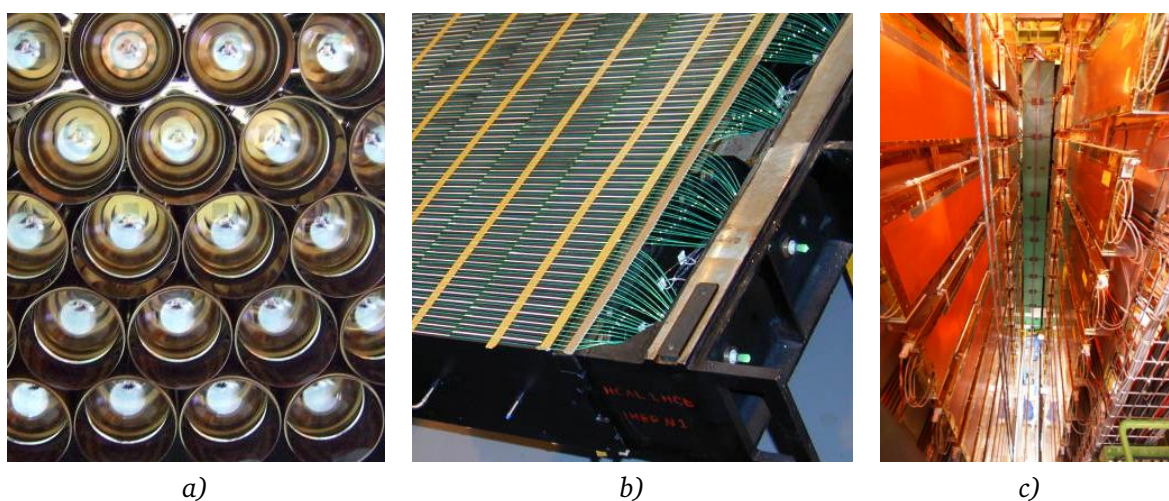


Figure 1.3: Photographs of the LHCb subsystems. a) - photomultiplier tubes in the RICH1 detector, b) - HCAL during assembling, c) - inside view of muon stations [25].

are constructed with alternating layers of metal and plastic plates. The secondary particles, generated in metal plates, excite molecules in polystyrene plates, which emit UV light. It gives a possibility to identify neutral particles like photons or neutrons and selects particles with high transverse energy for the level 0 trigger. The calorimeter system contains also the Scintillating Pad Detector (SPD) and the Pre-Shower (PS) detectors. First of them determines whether the detected particles are charged or neutral and improves the separation of electrons and photons, while the PS investigates particle's electromagnetic character. Both are used to check the presence (at trigger level) of electrons, photons, and neutral pions. The ECAL is based on shashlik technology [27] where 2 mm lead plates are placed alternately with 4 mm thick scintillator plates. The light from scintillators is detected by photomultipliers. The calorimeter has three cell sizes, 4×4 cm close to the beam, 6×6 cm in the middle and 12×12 cm in the outer area. The ECAL is 7.76 m wide and 6.3 m high and covers an acceptance from 25 mrad to 300 mrad in the horizontal plane and vertical 25 mrad – 250 mrad. The HCAL is constructed from iron (used as absorber) and scintillating tiles, working as active material. The HCAL during assembling is presented in figure 1.3b. Similarly to the ECAL there are two cell types: the inner parts with dimensions 13×13 cm and the outer parts 26×26 cm. The total weight of the HCAL is around 500 tons.

Muon detectors are very important for the LHCb experiment, because muons are present in the decays of B mesons, so their triggering and offline identification are fundamental. The muon detecting system is located at the end of the LHCb detector and contributes to the L0 trigger, the High-Level Trigger (HLT) and offline analysis. Figure 1.3c shows inside view of muon stations. There are

five rectangular muon stations (M1 – M5), covering an acceptance of ± 300 mrad (horizontally) and ± 250 mrad (vertically). The stations are equipped with Multi Wire Proportional Chambers (MWPC) with 2 mm wire spacing, but for inner region of M1 (highest rate) the Gas Electron Multiplier (GEM) detectors are used. The detectors are optimized for speed because the information must be gathered within 20 ns. The first muon station (M1) is located in front of the SPD and the PS detectors, while the M2 – M5 are located behind the HCAL and are separated by iron filters. The full muon system consists of 1380 chambers with 20 different sizes, which occupy area of 435 m². Each station is divided into four regions, R1 to R4, with increasing distance from the beam axis. The linear dimensions of these regions and their granularity are shaped according to the particle density. As a result the channel occupancies are roughly constant over the detector. The minimum momentum that a muon must have to reach the five stations is around 6 GeV/c.

1.3 LHCb upgrade and its motivation

The data collected during the LHC Run 1 provided LHCb results, which proved that measurements of excellent quality can be made in the heavy flavour sector in the extreme environment of high energy proton-proton collisions [14]. More results are expected from the LHC Run 2. The precision studies may become the only way to unravel new physics at the LHC, because no physics phenomena beyond the SM have emerged from Run 1. The highest possible LHC energy and luminosity that each LHC experiment can afford is needed to maximize the sensitivity of these studies.

The readout and trigger system of the current LHCb detector limits the data rate which can be injected into the trigger farm, and so the precision which can be achieved. The upgrade of the LHCb detector [3], which will take place during the Long Shutdown 2 (LS2) from mid 2018 to the end of 2019, will extend significantly the physics reach of the experiment by allowing it to run at higher instantaneous luminosity with increased trigger efficiency for a wide range of decay channels.

The LHCb upgrade is based on two major changes. Firstly, the full readout of the front-end electronics will be replaced with a 40 MHz trigger system. The current LHCb trigger system consists of two levels. The first level, implemented in hardware, is designed to reduce the event rate from the nominal LHC (40 MHz) to a maximum of 1.1 MHz. The complete detector is read with the reduced rate and the data is processed by the HLT implemented on the Event Filter Farm (EFF). The HLT is a software trigger, running a simplified version of the offline event reconstruction to cover the Central Processing Unit (CPU) time requirements. The new system will allow to collect complete events every 25 ns, send it to the LHCb data acquisition farm and apply a full software trigger for every single bunch crossing. This change will improve the trigger efficiency significantly for a broad range of LHCb physics channels, but the front-end electronics of several detector subsystems, in particular the silicon tracking devices, must be replaced, as well as the sensors.

Secondly, the upgraded LHCb detector must be designed to operate with five times higher nominal operational luminosity, compared to the current detector. The LHC will collide protons at a centre-of-mass energy 14 TeV, which gives increase of the heavy flavour production cross-sections by almost a factor of two, compared to those at 8 TeV. The instantaneous luminosity for the LHCb upgrade will be kept constant at the nominal value 2×10^{33} cm⁻²s⁻¹. These conditions will be achieved with average 7.6 of visible interactions per bunch crossing and 25 ns separation between bunches [23].

Figure 1.4 shows a side view of the LHCb upgrade detector. To improve detector's functionality a several subsystems need to be partially rebuilt in comparison to the current solution. Among these are the tracking subsystems: the VELO tracker, the TT located just before dipole magnet, and the T-stations placed just after the LHCb magnet. The four TT planes will be replaced by a new high granularity silicon micro-strip planes with an improved coverage of the LHCb acceptance. The TT

subsystem and its projected upgrade performance, described in subsection 1.3.1, is the focus of this dissertation. The new system is called the Upstream Tracker (UT).

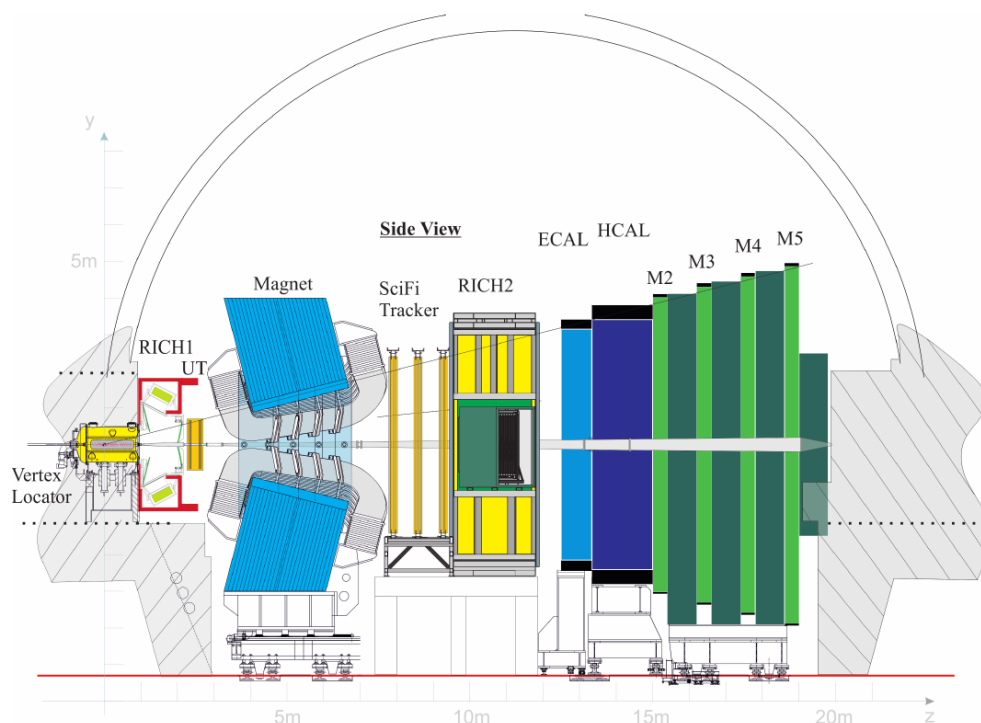


Figure 1.4: Schematic of the LHCb upgrade detector [23].

The current T-stations consist of two parts (detectors): the Inner Tracker (IT) works in the high η region and is composed of silicon micro-strip detectors; the Outer Tracker (OT) consists of straw drift tubes and works in the low η region. The three OT/IT tracking stations will be replaced with a Scintillating Fibre Tracker (SFT), composed of 2.5 m long fibres, which can be read out by silicon photo-multipliers, placed outside the detector's acceptance. The charged particle tracking is an essential physics tool of the LHCb experiment. It must provide the basic track reconstruction, which gives a precise measurement of the charged particle momenta in the extreme environment of the LHCb upgrade.

The current VELO sensors are produced in two types: R sensors that contain strips placed circumferentially and Φ sensors with radially placed strips. This geometry matches the sensors occupancy which varies with the distance from the beam. The sensors granularity provides occupancy around 1% at nominal luminosity. For the LHCb upgrade pixel detectors will be designed, because of higher occupancy environment. The pixels will provide low occupancy 3D positioning, better signal to noise performance and their small size will improve the radiation tolerance. To achieve the resolution similar to that obtained by inner strips in the current detector, a pixel dimension around $50\mu\text{m}$ or less will be required [28].

1.3.1 Upgraded tracker system - Upstream Tracker (UT)

The Trigger Tracker (TT) is very important for the reconstruction of K_S^0 mesons that decay outside of the acceptance of the VELO. The studies to use TT information in the HLT tracking algorithms give promising results. The momentum resolution was increased by about 20% while TT and the

downstream stations (T1 – T3) hits was added to tracks reconstructed in the VELO. The TT is very important in track reconstruction. For example in the decay $\overline{B}_s^0 \rightarrow D_s^+ \pi^+ \pi^- \pi^-$, with $D_s^+ \rightarrow K^+ K^- \pi^+$, the background to signal ratio is reduced from 12.2% to 8.4% when TT hits on all six final state tracks are required. Currently, TT hits cannot be required for all tracks in the acceptance as this would result in a too low efficiency. Eliminating this inefficiency is one of the goals of the UT design.

The TT worked very well during LHC Run 1. At the end of the run, 99.4% of all readout channels were fully operational and a single hit efficiency was around 99.7%. The measurements show a single hit spatial resolution around 61 μm , including residual effects from imperfect alignment.

The positive experience from LHC Run 1 operation allows to relax requirements regarding the clearance to the LHC beam-pipe. This gives a possibility to significantly improve the forward acceptance of the detector and to reduce the material budget in the very forward region [23]. However, the current TT has to be replaced for the LHCb upgrade, despite its good performance, from the following reasons:

- The silicon sensors used in the current detector are not designed to be sufficiently radiation hard to survive the expected radiation damage, in particular in the inner region of the detector;
- The current readout strip geometries would receive the high occupancies under the future running conditions, what is not acceptable;
- The Beetle chip [29], designed for current sensors, is not compatible with the planed 40 MHz readout. Moreover, the front-end hybrids with the Beetle chips are an integral part of the mechanical structure of the detector modules, so cannot be replaced without damaging the module.

The LHCb upgrade needs electronics compatible with 40 MHz readout. Moreover, in the inner region of detectors the electronics needs to be made radiation hard, which can only be done by replacing the entire system. The current sensors geometry has gaps caused by non-overlapping sensors, displacement of the top and bottom detector halves, and distance between the beam pipe and the sensors. The new system will eliminate the gaps entirely and will reduce the distance between the beam pipe and the sensors as much as possible. To obtain that the insulating material and the clearance will be significantly reduced. These improvements will ensure that a track which is projected to the active UT area, outside of the beam pipe region, will give a signal. Taking three hits from the four layers as a requirement, the efficiency should be greater than 99.7%, for a 98% single hit efficiency.

The current TT is designed to work with an integrated luminosity of about 10 fb^{-1} , while the UT detector needs to keep its performance with five times higher integrated luminosity at least. The detector performance studies show that all the components in the region near the beam pipe need to be irradiated up to 40 MRad (a safety factor of four was included) to validate their ability to sustain performance. In addition, the electronics located near the detector box should work with a radiation level around 100 kRad.

As already mentioned the UT detector is a replacement for the TT. It has four planes of silicon strips, similar to these used in the TT, but with a thinner sensors, a finer segmentation and a larger coverage. The overview of the UT geometry is presented in figure 1.5. The signals are processed at the sensor planes without using long cables, which allows to improve the electronic noise performance of the entire system. The magnetic field bends tracks in the horizontal plane (X). Therefore, in order to measure the track momentum precisely, the strips are placed vertically in the Y direction. There are four planes called: UTaX, UTaU, UTbV, and UTbX, which are located progressing in the downstream direction. The first and the last plane have vertical strips, while the middle two planes are rotated by $\pm 5^\circ$. The center of UT detector is located at a distance 2485 mm (Z direction) from the interaction point. The distance between the first and the last plane is 315 mm.

The design consists of sixteen staves for each of the two upstream planes and eighteen staves for each of the two downstream planes. Each staff consists of fourteen $98.88 \text{ mm} \times 98.88 \text{ mm}$ square sensors, except the central region, where the sensor geometries are different. Each sensor (in green) has guard rings of $800 \mu\text{m}$ width, which surround the nominal 512 strips. The strips have 97.28 mm in length and are placed with $190 \mu\text{m}$ pitch. The signals from a sensor will be read out by four 128-channel readout Application Specific Integrated Circuits (ASICs). A fraction of the sensors near the beam (in yellow) have two times smaller pitch and default length. The most central sensors (in pink) have smaller pitch and also a two times smaller length.

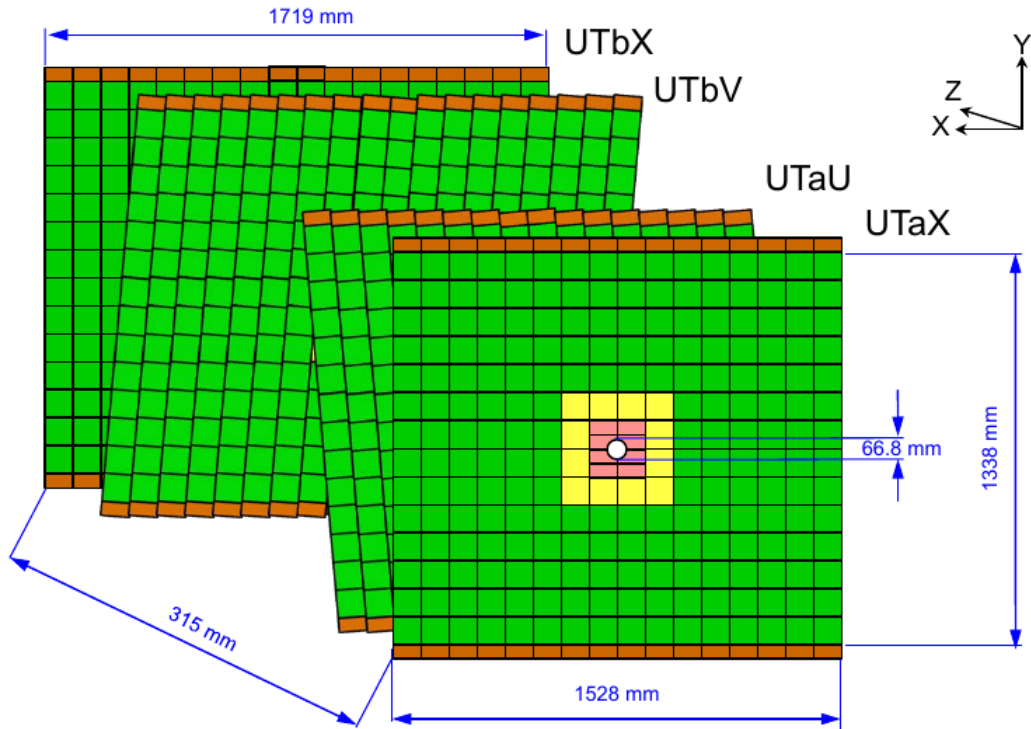


Figure 1.5: Overview of UT geometry looking downstream. Different sensor geometries are marked by different colors [23].

The beam-pipe runs through the center of the detector, so each plane has a hole in the center. The UT planes have circular cutouts which provide better acceptance on tracks than square holes with the same allowed size. The reduction in material can be also done by reducing the thermal insulation layer surrounding the beam-pipe which is much thinner in the new design than that in the current system. The circular cutout in the innermost sensors is determined by the size of the beam-pipe, the thickness of thermal insulation layer, and the clearance required. The existing beam-pipe at UTbX has the outer radius equal 27.4 mm . The new design of thermal insulation presupposes 3.5 mm thickness and 2.5 mm clearance. In result the inner radius of the silicon sensor is equal 33.4 mm , but the active area starts at 34.2 mm , due to the 0.8 mm guard ring. The central hole provides acceptance starting from 14 mrad for straight tracks from the center of the interaction point. The simulation showed that for a typical B decay, only about 5% of the events are lost.

Each UT sensor is composed of $250 \mu\text{m}$ thick silicon with $10 \mu\text{m}$ metalization layer. The different sensor types are marked by colored squares in figure 1.5. In the central area the track density is very high, so sensors of smaller strip pitch, and also shorter length are used. The sensors marked by yellow squares have nominal length and pitch twice smaller ($95 \mu\text{m}$) than the nominal sensor. The sensors

shaded in pink have both half the nominal pitch and the half nominal length, being about 5 cm long in Y direction. In result the central two staves have sixteen sensors each, instead of fourteen. Each of the sensors with smaller pitch has 1024 strips which are read out by eight ASICs. The nominal sensors, marked green, have 512 strips which are read out by four ASICs.

The staves used in construction of detector planes are similar to those used in the ATLAS upgrade [30]. Each staff has approximately 10 cm width, same as silicon sensors. The sensors and the front-end readout ASICs are placed on custom hybrids, which are mounted on thermo-mechanical support structures. The staves are about 1.6 m long and mounted vertically. The signals from the sensors are taken out to the top and bottom of the UT by (data) flex cables. In similar way the power supply voltages are connected to the sensors and electronics. Outside of the spectrometer acceptance the staves are supported by a rigid frame. The cooling system will keep the sensor temperature below -5°C .

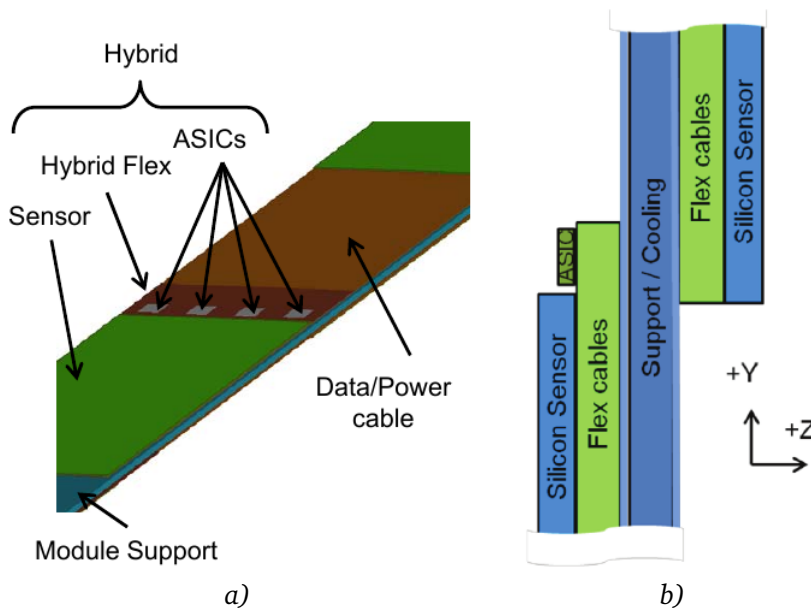


Figure 1.6: UT staff structure. *a)* - hybrid with silicon strip sensor and readout ASICs attached to hybrid flex, *b)* - cross section of a single staff, which shows how sensors are mounted on both sides of the support structure allowing for sensor overlap [23].

The staff structure is presented in figure 1.6. The hybrid flex is about $220\ \mu\text{m}$ thick, has the same width as the sensor but it is 20 mm longer in order to mount the ASICs and wire bonds. An ASIC will be approximately 5 mm wide and 10 mm long. The hybrids are mounted on both sides of the staff support and have a 2 mm overlap in Y direction to cover the gaps between sensors. The staff support also contains the cooling tube. The flex cables which carry power, ground and data lines are placed between the staff support and the hybrids. Each staff has four flex cables for top and bottom halves, front and back faces. Each cable starts from the readout edge of the innermost hybrids till the end of the staff. The staves are connected to periphery electronics from top and bottom. The cable which runs along the staff, with power and signal lines from the innermost sensors, is around 0.7 m long (half-staff length). There will be around 20000 signal lines, so the cable design involves a trade off between low mass requirement and low voltage drop and signal integrity. The 4192 Silicon ASIC for LHCb Tracking (SALT) ASICs will consume around 4 kW of total power. The data signals will be sent using a Scalable Low-Voltage Signaling (SLVS) standard. There

will also be several low speed I²C buses used for configuration and status monitoring. A number of temperature and humidity sensors will be distributed throughout the detector planes.

The outer staves contain fourteen silicon sensors, while the two inner staves contain sixteen sensors. The sensors near the beam-pipe are divided into two 5 cm parts. The sensors are mounted on both sides of a stave and adjacent staves are staggered in the Z direction, which allows for sensor overlaps. It ensures that the sensors are mounted with no gaps in both directions (X and Y). The space between silicon sensors is used for ASICs, which are wire-bonded [31] to the silicon and attached to the flex cables. Each UT layer has the staves staggered along the beam line allowing for the overlap of sensors in X direction. Example of a single layer is presented in figure 1.7. The dark blue shows part of the supporting structure. The Kapton cables are marked by brown color and green color presents the silicon sensors. The overlap in Y is achieved by mounting the sensors on the front and back of each stave. Both ends of each stave have aluminium blocks to ensure mounting. There is equal number of sensors on the other side of the stave which cannot be seen in this figure. The adjacent staves are staggered to allow for the overlap of sensors, stave to stave.

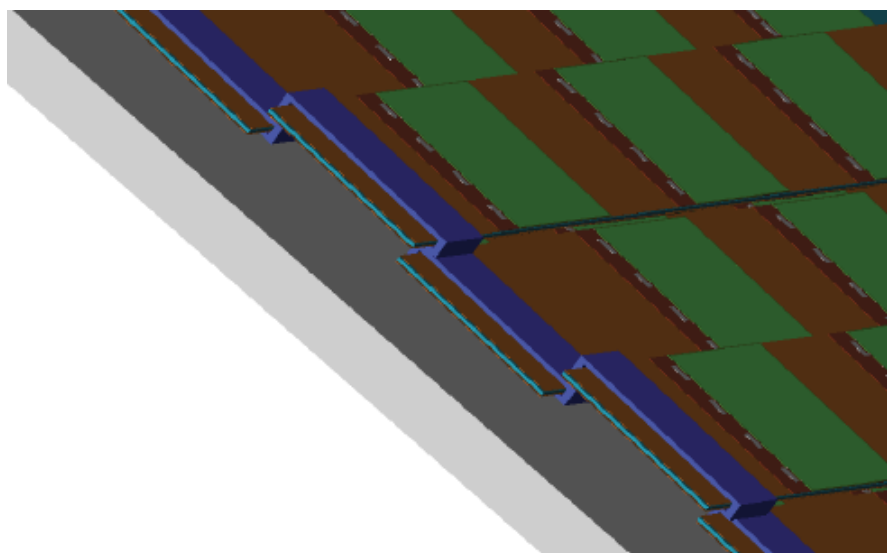


Figure 1.7: Mounting of a stave layer to the frame [23].

A cross-section of the stave layout is shown in figure 1.6b. The stave consists of a sandwich structure made of thin, Carbon-Fibre Reinforced Polymer (CFRP) facing sheets surrounding a lightweight partially filled foam core interior. There is one or more thin-walled tubes, embedded in the foam core, which remove the heat generated principally by the ASICs. The foam core is a mix of thermal and structural foams, optimized to provide maximal heat transfer and obtain minimal radiation length. In result the stave structure provides stiff support with good cooling performance and signal transmission media, with minimal mass. The hybrids are precisely mounted on either side of the stave, and wire bonded to the data and power flex cables. A stable support is very important for a delicate wire-bond connections, because any mechanical or thermal stressing may lead to the connection breaks. Hence, any motion or twisting of the integrated stave must be avoided. Mounting the hybrids on either side of the stave minimizes any relative thermal expansion. Mechanical mounting of the stave to the rigid outer frame will aid in minimizing mechanical motion. These issues will be fully analyzed by simulation and tested with measurements. Mechanical construction of the Upstream Tracker allows to align the silicon sensors of the LHCb coordinate system with an accuracy of 100 μm .

1.3.2 UT Silicon Sensors

The UT sensors are single sided silicon micro-strip devices. The expected radiation dose and occupancy lead to their segmentation and the technology used. For an integrated luminosity of 50 fb^{-1} the detailed radiation background simulations [32], including safety factors motivated by previous experience, predict a maximum dose of 40 MRad at the innermost edge of the silicon sensors and a fluence of $5 \times 10^{14} \text{ n}_{\text{eq}} \cdot \text{cm}^{-2}$, rapidly decreasing with a distance from the beam axis. The sensors

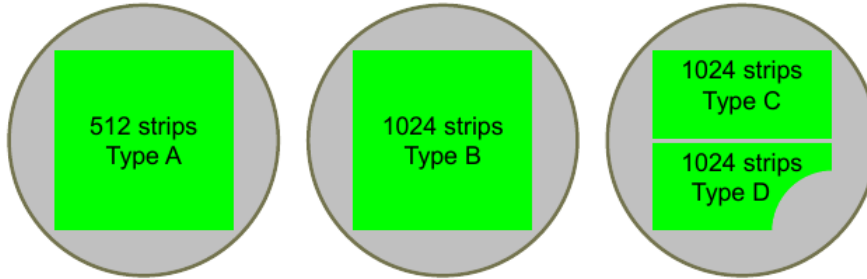


Figure 1.8: Four sensors geometries for the UT upgrade [23].

segmentation is finer in the inner part of the detector, around the beam-pipe, and it is coarser in the rest of the detector. Four kind of detectors are proposed, as shown in figure 1.8, called type A, B, C, and D. The detectors C and D are only 5 cm long, in order to allow for a higher vertical segmentation. This permits a simpler sensor design without double metal layer to route the signals from the shorter strips to the contact pad row. Most of the detector staves are constructed with sensors of type A. The expected radiation dose in these sensors is very small (100 – 300 kRad, depending upon the location), so a traditional $p^+ - in - n$ technology can be used for their implementation. The sensors of type B, C, and D are closer to the beam axis, and thus the technology chosen is the $n^+ - in - p$ which demonstrated to be good enough for severe radiation environments [33].

A very important issue is the implementation of the interconnection between strip and corresponding front-end electronics input channel, which is designed with $80 \mu\text{m}$ pitch. The ASICs will be directly wire bonded to sensors, without the use of an intermediate pitch adapter. This can be implemented by adjusting the angle and the length of the wire bond for the B, C, and D sensors. The sensors A require a "fan-in" circuitry, which allows pitch matching between $190 \mu\text{m}$ and $80 \mu\text{m}$. The outline of the detector shape is non-standard in sensors D, because one of the corners has quarter-circle cut-out to maximize the active area near the beam-pipe and to maximize the angular acceptance matching with the VELO system. The radius of this cut-out is 33.4 mm.

1.4 SALT – readout ASIC for Upstream Tracker

The AGH-UST group from the Department of Particle Interactions and Detection Techniques is the member of the UT group and is responsible for the design of the Silicon ASIC for LHCb Tracking (SALT), which is a dedicated readout system for silicon microstrip detectors. According to the author's knowledge the SALT will be the first multi-channel low-power ASIC in the world, designed for HEP application, where the fast data sampling and Analog to Digital Converter (ADC) conversion is applied in each readout channel. The SALT will consist of complete multi-channel readout chain: starting from a front-end electronics, through an ADCs, and ending with a Digital Signal Processing (DSP) and a fast data serialization. It is the first time when functional blocks with ultra-low power consumption, like: ADC [34], Phase-Locked Loop (PLL) and Delay-Locked Loop (DLL) are

designed and integrated in an ASIC for HEP application.

The SALT is a novel front-end readout chip for silicon micro-strip sensors of the Upstream Tracker (UT) at the LHCb experiment. The project is motivated by the LHCb upgrade, whose goal is replacing the existing hardware trigger and software trigger, working at 1 MHz frequency, by the 40 MHz readout, working only with the software trigger [23]. The SALT is designed to meet the requirements for a new readout electronics, which should allow to collect the data from large number of silicon strip detectors channels (around half a million). The new UT consists of 4 planes of silicon strip detectors, each of them consisting of single sided sensors with various pitch and length. It makes that the capacitances of the sensors are different and so the front-end electronics should be more complicated to meet these requirements.

Parameter	Requirements
Channels	128
Input pitch	80 μm
Total ionising radiation dose	40 MRad
Total power dissipation	< 1 W
Load capacitance	5 pF – 20 pF (typically)
Maximum leakage current	~ 200 nA (per channel)
Noise	$\sim 1000 e^-$ @ 10 pF
Maximum cross-talk	< 5% (between channels)
Signal polarity	Both (electron and holes collection)
Gain uniformity	$\sim 5%$ (across channels)
ADC bits	6 bits
ADC sampling rate	40 MHz
Output serializer	serial e-links at 320 MBit/s

Table 1.1: Selected parameters and design requirements of the SALT ASIC.

The SALT introduces various technological challenges related to the required design performance and severe environment of the 14 TeV pp collisions. The SALT ASIC, which will be finally manufactured in 130 nm Complementary Metal-Oxide Semiconductor (CMOS) technology B will consist of 128 channels. Each of them comprises a charge sensitive preamplifier, a shaper and a single-ended to differential converter, which form the analogue part of the chip. The differential analogue signal is then sent to a fully differential 6-bit ADC. The digitized data undergoes digital processing, which performs: pedestal subtraction, mean common mode subtraction and zero suppression. After the DSP the data, with added header information, are fed to the de-randomising buffer and are transmitted serially to a consecutive parts of the readout system [35]. Summary of the specification and overall requirements of the SALT are shown in table 1.1.

The SALT ASICs will be placed on low mass flexible support, which will provide electrical connections of the chip to the data and power flex cables. A thin flex support is equipped with 8 (4) SALT ASICs and the sensor is glued on it and wire bonded to the front end electronics. In this structure, called hybrid, the signals generated in the sensor are processed, digitized, formatted and serialized by the ASIC. The output data are then transferred to the external electronics via low mass flex cables, without further processing. The need to minimise the material in the active region does not allow for additional electronics on the hybrid. The first step in the signal processing - preamplifier, implemented in SALT, is particularly sensitive to noise, so the layout of the flex cables must be developed to minimize the coupling of the analogue and digital sections. The cross-talk and noise coming from the

power planes must be also minimized. The power supplies will be distributed in wide planes, which allow to reduce the trace inductance as much as possible and to achieve good capacitive coupling with ground. Each power line must be filtered locally with capacitors to the common return.

Figure 1.9 shows a block diagram of 128-channel **SALT ASIC**. It consists of a preamplifier, a shaper, and an **ADC** in each channel, and common blocks like: **PLL**, **DLL**, **DSP** circuit, serialization/deserialization circuits, slow control configuration interface (I^2C) and **SLVS** input/output buffers.

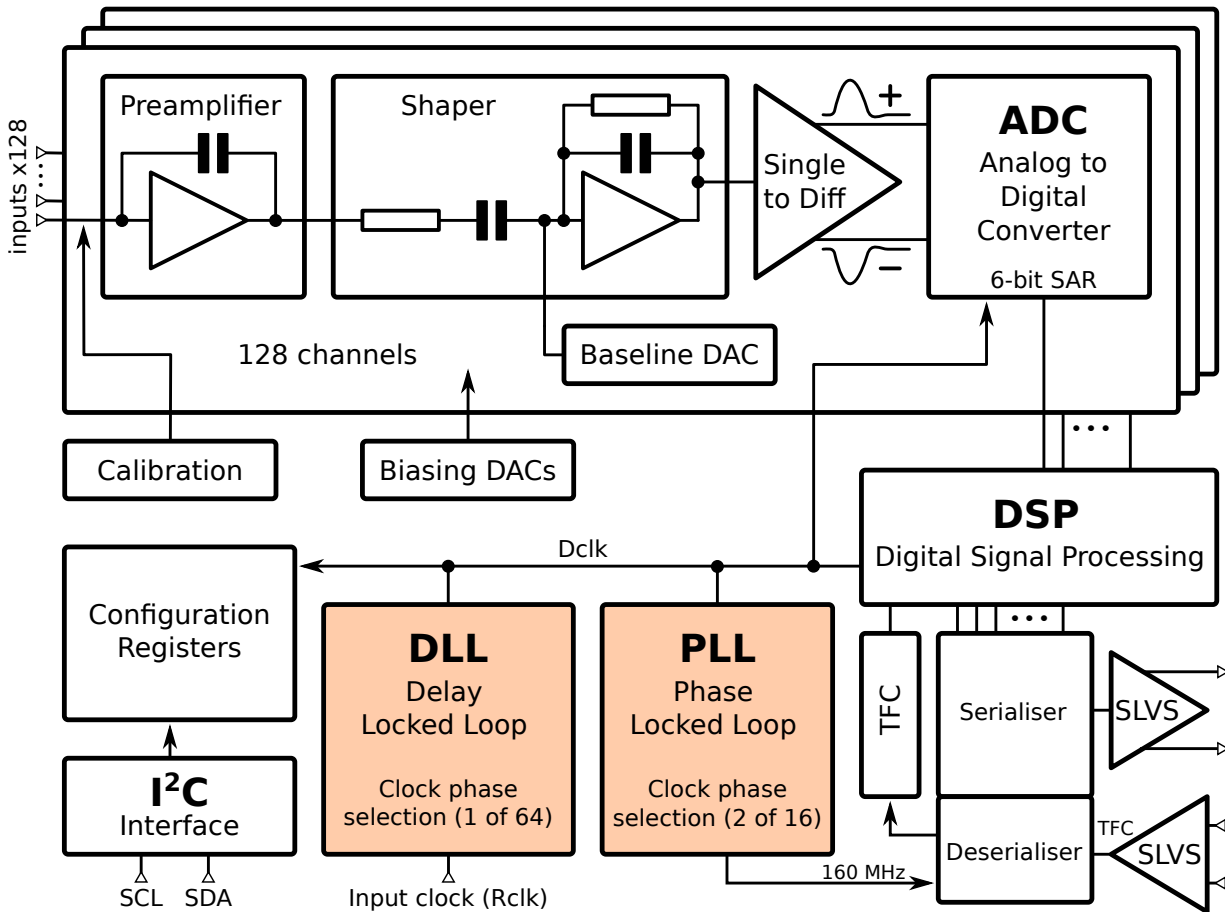


Figure 1.9: Block diagram of the **SALT ASIC** (128 channels).

1.4.1 Analogue front-end

The analogue front-end (preamplifier and shaper) circuit is very demanding because the amplified pulse, with peaking time (T_{peak}) around 25 ns, should have very short tail, below 5% of pulse amplitude after $2 \cdot T_{\text{peak}}$, which is needed to minimize the pile-up. The circuit should also have very low power consumption, around 1 mW – 2 mW per channel. The front-end should work with different strip sensors (capacitance range 5 pF – 20 pF), with input signals of both polarities, and with good enough signal to noise ratio (>10), even in the worst operation conditions. Moreover, the Equivalent Noise Charge (**ENC**) should be around 1000 e^- at sensor capacitance 10 pF. One of the main challenges for the analogue block is to obtain a very short signal duration with a minimum possible power consumption. To obtain an acceptable tail, a $CR-RC^{10}$ shaping would be needed, which is very complicated and consumes a lot of power. Figure 1.10b shows a $CR-RC^n$ filter type output responses for different filter orders. Introducing complex poles and zeros in transfer function

allows to shorten the pulse tail to the required goal and obtain realistic shaper complexity (preamplifier with Pole-Zero Cancellation (PZC) and three shaper stages) [36, 37]. Figure 1.10a shows the comparison between two shaper implementations, the first one with complex poles and zeros and the second one with a typical CR–RC³ shaping. Both are based on the same number of stages.

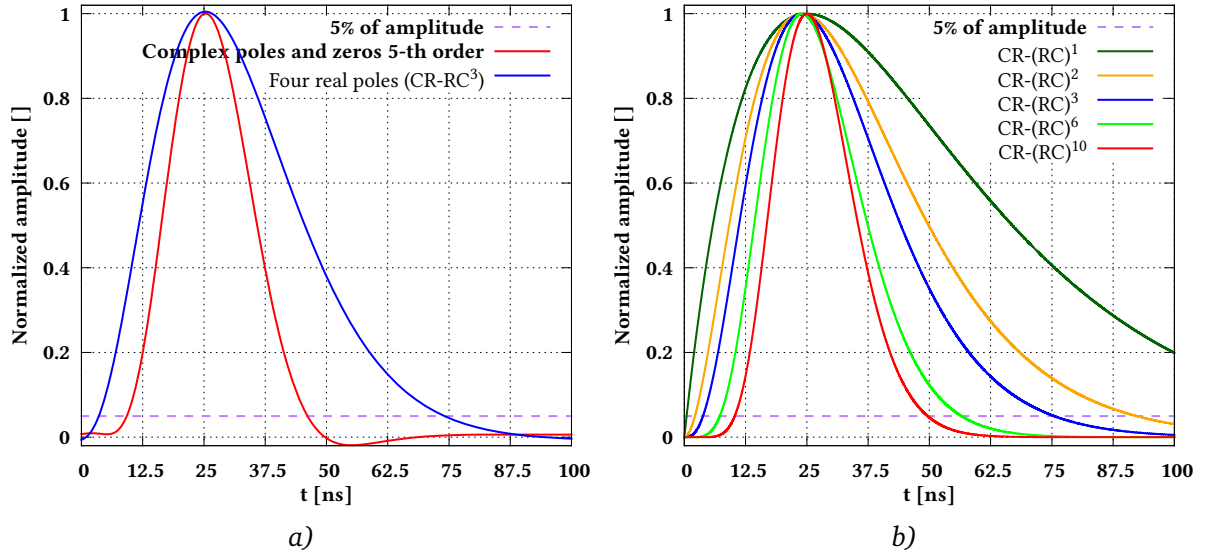


Figure 1.10: Comparison of several shaping implementation. *a)* - comparison between complex shaping and standard approach with real poles for the same number of shaper stages, *b)* - standard CR–RCⁿ shaping for different filter orders.

A simplified block diagram of the designed front-end is shown in figure 1.11. The NMOS input telescopic cascode with boosting amplifiers was used as the preamplifier [38], while the shaper stages are based on Recycled Folded Cascode (RFC) amplifiers to obtain lower power consumption at the same circuit speed [39, 40, 41]. In the preamplifier a Krummenacher type feedback was added [42]. The first shaper stage gives two real poles, while the second stage introduces two complex poles to the transfer function. The last shaper stage provides another two complex poles and two complex zeros. The baseline of the front-end and all bias currents are digitally controlled by internal 8-bit Digital to Analog Converters (DACs), which are based on typical architecture [43]. Both input signal polarities are acceptable at the front-end input, which will be used for different types of sensors.

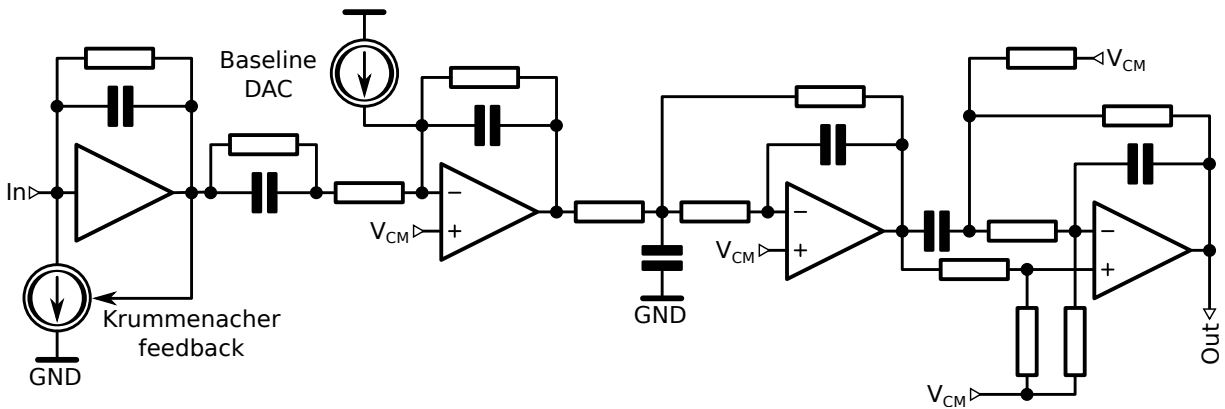


Figure 1.11: Simplified block diagram of SALT front-end electronics.

The front-end output generates pulses with two polarities relative to the common mode voltage (V_{CM}), which default value is 0.6 V, so the amplitude obtained by the front-end ideally ranges from -0.6 V to 0.6 V. On the other hand the ADC, based on differential architecture, accepts differential input signal in range -1.2 V to 1.2 V for the default reference voltage of 1.2 V. To adjust the output signal from the front-end to the ADC range, a single to differential converter is needed. As one of the possible options the converter based on Switched Capacitor (SC) circuits [43] with gain equal 2 will allow to reduce the power consumption and will adjust the front-end amplitude to the level acceptable by the ADC.

1.4.2 Analog to Digital Converter (ADC)

The simulations confirmed that 6-bit resolution is more than sufficient for the tracking purposes of the experiment. One of the most important constraints in the ADC design is a very low power consumption, much less than 1 mW at default sampling frequency 40 MHz. The designed ADC is an ultra-low power 6-bit Successive Approximation Register (SAR) ADC [44, 45, 46]. The ADC architecture, chosen for current design, is presented in figure 1.12. The main blocks of the SAR ADC are: input sampling switches, differential DAC, comparator, and control logic. A single ended and a fully differential architectures of ADC were considered [44], but despite the low resolution required, the latter was chosen to improve the ADC resistance to various disturbances in experimental environment. In order to increase the linearity of ADC the input switches are bootstrapped, what reduces significantly their dynamic resistance [47].

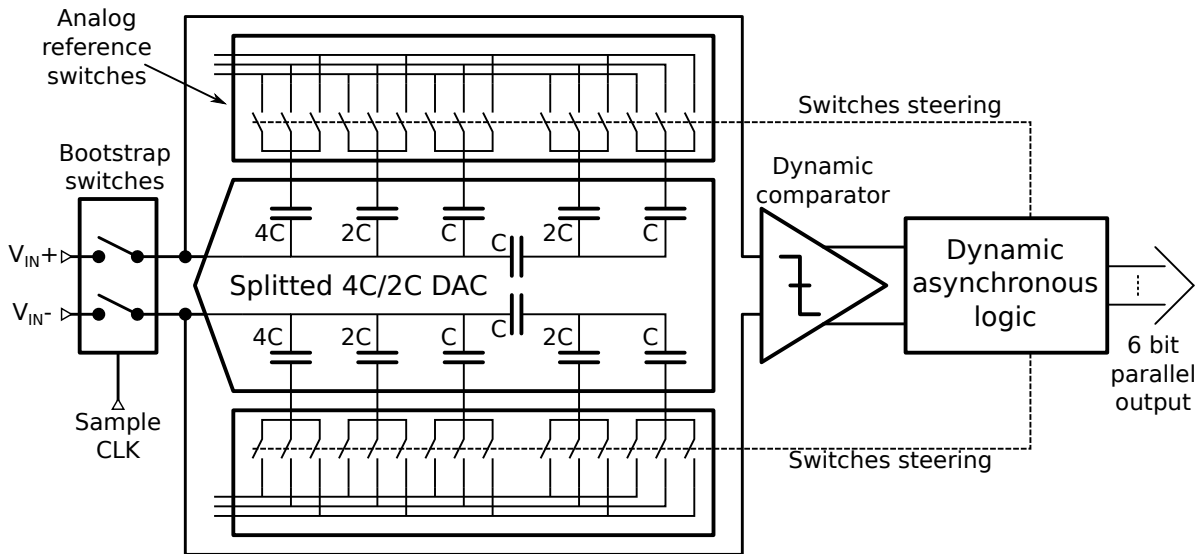


Figure 1.12: Simplified block diagram of the 6-bit SAR ADC.

The input signal should be sampled with a rate of 40 MHz, which is needed for proper signal digitization in LHCb. After sampling, the first SAR iteration begins by comparing V_{IN+} and V_{IN-} and setting the Most Significant Bit (MSB) bit in the ADC output register. The MSB is equivalent to the sign of the input signal. First step is performed without any switching in the DAC arrays. In the next steps, the capacitors in the DACs arrays (initially connected to the common mode voltage V_{CM}) are switched either to reference voltage V_{ref} or to GND according to the comparison result between the $V_{IN} = V_{IN+} - V_{IN-}$ and the voltage currently set by the DAC. Then the comparison is performed and the following bits are processed in a similar way, until the Least Significant Bit (LSB).

In the SAR ADC architecture a comparator is usually the only component with static power

consumption. The dynamic comparator, dissipating power only during bit comparing process is chosen, to completely avoid the static power consumption. The solution with two gain stages and output-latch stage providing high enough precision was chosen [48]. A fully dynamic comparator gives also power pulsing functionality, because when the ADC clock is not running the circuit does not consume any power (except leakage current).

For the lowest power consumption, capacitive DAC arrays are implemented [49]. A differential segmented/split DAC with Merged Capacitor Switching (MCS) scheme is used [50]. A Metal-Insulator-Metal (MIM) capacitors are used in the DAC, because they have the best capacitance matching in the given technology. The MIM capacitors are relatively large so a split capacitor DAC scheme was used, as shown in figure 1.12. The ADC is fully differential so there are two DAC arrays. Each of them contains three capacitances on the MSB side, and two capacitances on the LSB side. The splitting capacitance between MSB and LSB arrays is equal to the unit capacitance C , which is important to improve capacitance matching. The minimum available MIM capacitance in the chosen technology is around 60 fF, so the effective unit capacitance of the splitted DAC is around 15 fF. It is a relatively large value if noise and matching issues are taken into consideration, so in theory there is still a possibility to reduce the unit capacitance and so the power consumption.

The control logic in ADC can be synchronous or asynchronous, but the first of them has two disadvantages: a limited speed, and an additional power consumption caused by the clock tree circuitry. The asynchronous logic can use different delays in comparator response (for different bits) to save some time in each conversion and increase the sampling rate. An external clock sets the sampling frequency, and no other clock is needed for bit cycling, which allows to save the power. Part of the logic was implemented with dynamic flip-flops, which allow to save more power and increase the maximum sampling frequency up to 80 MHz. The result of conversion is stored until the end of the next conversion in an additional static output register. The ADC samples are signed 6-bit numbers coded as 2's complements.

The first ADC prototype was already designed and allows for asynchronous sampling with up to two times higher frequency than requested [34]. The circuit was integrated in a prototype 8-channel SALT readout chip for LHCb UT. The power consumption of the circuit scales with sampling frequency, which is a great advantage of general purpose ADC.

1.4.3 Digital Signal Processing (DSP)

The data processing in SALT is presented in figure 1.13. The digital ADC output is processed by the DSP module, which consists of following blocks: pedestal subtraction (P), mean common mode subtraction (MCM), and Zero Suppression (ZS) [23]. For a better testability the DSP allows to transmit a raw ADC data or various combinations of partially processed data.

The DSP processing starts from setting the proper signal polarity, by turning on/off the arithmetic inversion of input data. It is necessary if ASIC receives the signal of negative polarity (for $n^+ - n$ silicon sensor), when the ADC sample value is expected to be negative. When the ASIC is connected to $p^+ - n$ sensor the signal inversion is turned off. After that a pedestal subtraction is performed. The data after pedestal subtraction are sent to the next block only if the binary mask bit, for a particular channel, is zero. Otherwise the particular channel is masked and gives zeros instead of ADC data. This feature is intended to mask the noisy or dead channels and also helps to reduce the power consumption by avoiding not needed switching activity.

The idea of mean common mode subtraction is to calculate the average value of all channels without a hit and subtract this value from all channels. In ideal case, when there are no disturbances and the pedestals from previous step are correct, the calculated average should be zero. The mean

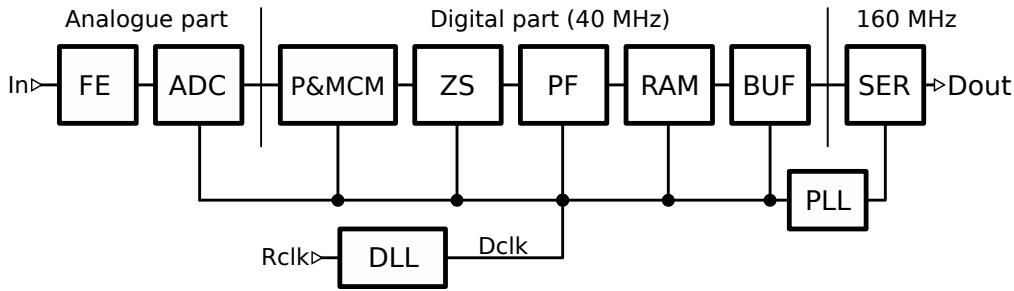


Figure 1.13: Data processing in *SALT*.

common mode subtraction works very well also in more realistic case when possible disturbances are identical in each channel. Before the subtraction can be applied the number of channels without a hit (non active) should be calculated, as well as the sum of amplitudes over these channels. Both calculations are implemented as hierarchical pipelined structures and in both cases the operation takes two clock cycles. The next step is to calculate the average value, which is achieved by a division sub-block. It is also implemented as a pipeline and the whole operation takes five clock cycles. As a result of the division the mean common mode value is calculated and rounded to the nearest integer value. Since the operations in this block are implemented as pipeline, so the data in each channel must be delayed by First In, First Out (FIFO) register (depth equal 7) to keep the consistency of subtraction operation. The depth of the FIFO is calculated as a sum of latencies of all operations.

The purpose of Zero Suppression (ZS) block is to compress the output data by omitting the channels without hits and to send out only the channels with useful information (active channels). Active channels are selected with a threshold set by one of the *SALT* configuration registers. Internally the ZS block is built as a large set of multiplexers forming the output data. The internal structure is pipelined, with the constant latency of two clock cycles.

1.4.4 Data transmission and clock generation

In large experimental systems the clock usually arrives to readout modules in different phases. Moreover, different capacitances of the sensors affect the peaking time of the front-end, so the ADC needs to align the clock to sample the data properly. For these reasons in *SALT* project a dedicated DLL circuit is used, which can generate 64 clock phases. The internal multiplexer allows to select the proper clock phase (*Dclk*), which is used as a global clock for the entire ASIC (figure 1.13). The reference clock for the DLL (*Rclk*) is received as standard LHC 40 MHz clock.

After the DSP operation the data packets are created and recorded in a local Random-Access Memory (RAM). The data are finally serialized and sent out with the rate multiplied eight times. This is obtained by increasing the clock frequency four times (to 160 MHz) and by using a Double Data Rate (DDR) transmission. The clock multiplication by 4 is obtained by a dedicated multi-phase PLL, which generates 16 independent clock phases at 160 MHz. Two clock phases, selected from 16 by multiplexers, allow the proper synchronization of serializer and deserializer circuits.

The PLL and DLL circuits, designed by the author, are widely described in this dissertation.

1.4.5 Communication interfaces

The ASIC is controlled via the LHCb common protocol consisting of two interfaces: the Timing and Fast Control (TFC) and the Experiment Control System (ECS) [23]. The TFC interface delivers crucial information and commands, synchronized with the LHC clock, while the ECS is used to configure and monitor the ASIC and it is realized through I^2C interface.

Chapter 2

Theoretical issues of clock generation and data transmission in readout systems

The transmission and serialization of data are widely present in today's electronic systems, ranging from simple household devices [51, 52], to advanced readout systems in HEP experiments [53]. The theory says that, the transmission is a process in which information (data) is sent from one device called Transmitter (TX) to another called Receiver (RX). The transmitted data is usually encoded in a strictly defined algorithm [54, 55]. It is important to determine the transmission medium, through which the transmission occurs eg. wired connection, wireless, or optical link, because construction of the TX and RX depends on it.

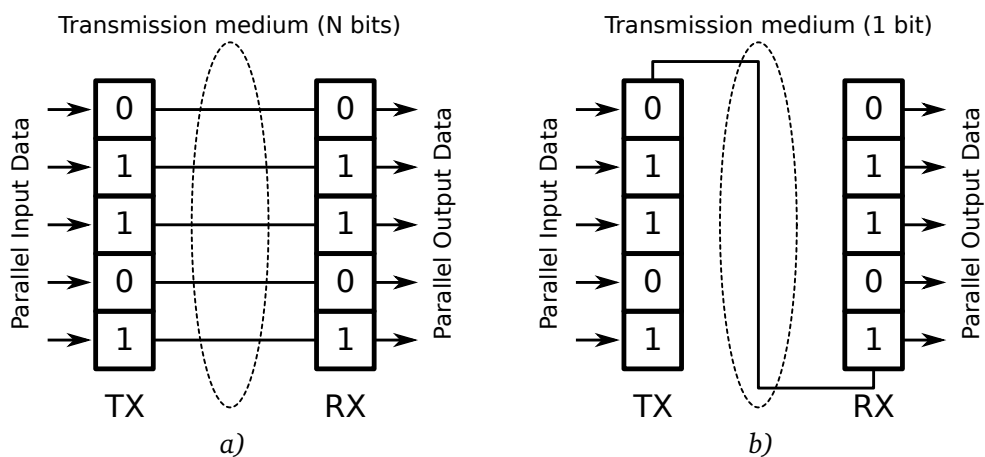


Figure 2.1: Data transmission methods. *a)* - parallel transmission, *b)* - serial transmission.

In general there are two types of transmission: parallel and serial. A parallel transmission example is presented in figure 2.1a. The data is transmitted via a multi-channel (N channels) transmission medium (it is usually a wired connection). One channel allows for the transmission of one bit of information at a time, so N bits can be sent in parallel. The serial transmission, shown in figure 2.1b, allows to send one bit of information at a time via one-channel transmission medium [56]. In both cases, data is supplied to the TX in parallel. In the same manner the data can be read from the RX.

The main advantage of the parallel transmission is its high speed data transfer and simple design of the transmitter. Disadvantages are: multiplied susceptibility to electromagnetic interference,

phase shifts between the parallel signals, and problems with synchronization between transmission channels at high-speed. Disadvantages of parallel transmission limit its application in communication systems to a short distance. Very often it is used for communication between the blocks of digital electronics inside integrated circuits, wherein over distances of around 1 mm it can achieve very high-speed. The serial transmission and its greater resistance to electromagnetic interference make it possible to achieve significant speed transmission over long distances. The main disadvantage of this type of transmission is the complexity of Transmitter and Receiver circuits [57].

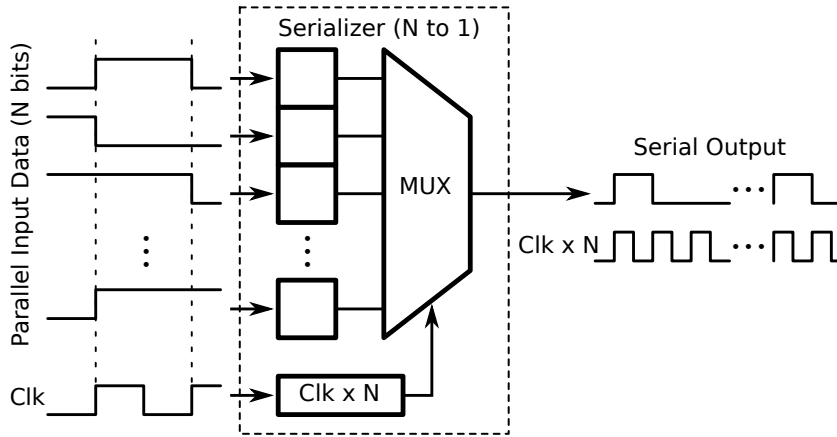


Figure 2.2: Principle of the serialization process.

In dedicated readout systems for HEP experiments the serial transmission of data is the most common option, because there are huge amounts of data and a limited physical size of the transmission media. Inside the integrated circuits the data is usually processed in parallel, because data processing must take place in multiple channels simultaneously. As a result, the data conversion from parallel to serial type is necessary. This process is called serialization and the principle of its operation is shown in figure 2.2 [58]. The parallel data stream (N bits), clocked by the Clk signal, goes to the input of the serializer. The clock signal, multiplied by N , is used for timing of the internal multiplexer in the serializer. During one period of multiplied clock, one of the N bits of input data is transmitted to the output via the multiplexer. The same happens with the other bits, one after another. In result, the serial data output is changed N times faster than the input data. When all (N) bits are sent, the next word of data is loaded to the input register and the whole cycle repeats. The generation of the multiplied clock signals can be achieved using Phase-Locked Loop circuits, which are presented in more detail in subsection 2.1.

In High Energy Physics experiments usually all readout electronics circuits are synchronized to the clock signal, which is related to the moment of the collision of particles in the accelerator. This helps to precisely define the moment in which the sensors can detect interesting particles. In such advanced systems, it is not possible to provide clock signal in the same phase to each of the readout modules, comprising many integrated circuits. As a result, each of the multi-channel chips must be able to change the phase of the clock signal. It is the only way to make the circuit operation independent on the clock signal propagation time (from source to the integrated circuit). The clock phase adjustment is needed also for accounting for differences in front-end circuits responses, depending on their operating conditions. Peaking time of the front-end circuits usually depends on the capacitance of the sensor, and the detector often contains sensors with different capacitances [59]. Less important to the clock signal distribution are phase differences due to the distance between ASICs at the same readout module. Phase shift within the integrated circuit can be realized in many different ways, however, the best way is to use a Delay-Locked Loop. It can be used to adjust the

phase in range of $0 - 2\pi$ (discretely) with a resolution dependent on the Delay-Locked Loop internal structure. More about this circuit is presented in section 2.2.

2.1 Phase-Locked Loop (PLL)

The Phase-Locked Loop (PLL) was invented in the 1930s [60] and after that it found wide usage in electronics and communication. PLLs are commonly used in analog television receivers, precisely in horizontal and vertical sweep circuits, to lock synchronization pulses with the broadcast signal. These circuits are also commonly used in the entire field of signal transmission, and also for high frequency synthesis, phase and frequency modulation and demodulation, clock recovery circuits, and clock synchronization [61]. There are many types of Phase-Locked Loop (analog, digital and software) but this dissertation is focused only on digital PLLs, which are of interest for HEP applications.

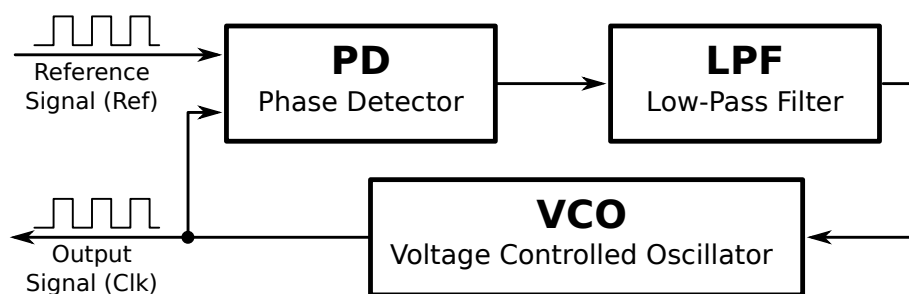


Figure 2.3: Simplified Phase-Locked Loop block diagram.

Figure 2.3 shows a simplified block diagram of typical PLL. The purpose of the circuit is to align the output signal frequency to the frequency of the reference signal in such a way that the phase difference between the signals became constant, possibly equal to zero. Both signals (output signal and reference) go to the Phase Detector (PD), and their difference is sent to the Low-Pass Filter (LPF), which gives a slowly changing signal to control the Voltage-Controlled Oscillator (VCO). The most important feature of Phase-Locked Loop architecture is the negative feedback. Thanks to the negative feedback, the level of the VCO control voltage is set at a value, which allows to generate the output signal consistent with the reference clock. For proper operation of the PLL the following blocks and functionalities are required:

- Phase Detector (PD) is a non-linear circuit, its output gives information about the phase difference between two input signals (usually the reference signal and the oscillator output). In a more expanded version the Phase and Frequency Detector (PFD) may be used, which compares not only the phase shift between input signals, but also their frequencies.
- Voltage-Controlled Oscillator (VCO) generates a clock signal with the frequency dependent (commonly it is linear function) on voltage set on its control input.
- Low-Pass Filter (LPF) in a simplest case it is a classic RC filter. It is used to smooth the response from the phase difference detection, in order to obtain a slowly changing voltage signal, necessary to control the oscillator.
- Negative Feedback is realized by connecting the output of the VCO to the negative input of the Phase Detector, making it possible to provide an error signal to control the frequency of the VCO clock signal.

2.1.1 Type I Phase-Locked Loop

Phase-Locked Loops are non-linear circuits and their main functional blocks are PD and VCO, which are responsible for the conversion of input signals in time domain to phase difference and vice versa. The PLLs in the literature are typically first or second order circuits. For the case of digital PLL the principle of the PD operation (in the simplest case it is a XOR gate) can be simply explained based on figure 2.4 [62, 63].

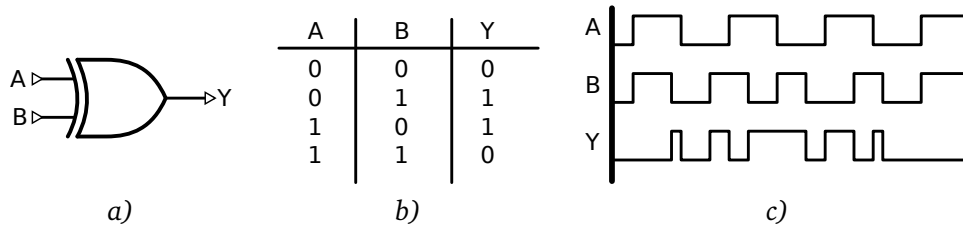


Figure 2.4: Principle of the XOR PD operation. a) - XOR gate symbol, b) - truth table, c) - sample waveforms.

When both input signals are in the same phase, the output of XOR gate is low. In other cases, if one of the signals is shifted in phase, the XOR gives an error signal as a response to different logic levels on its inputs. When a Low-Pass Filter is used, as shown in figure 2.5, the error signal will be averaged/smoothed, as needed to control the VCO.

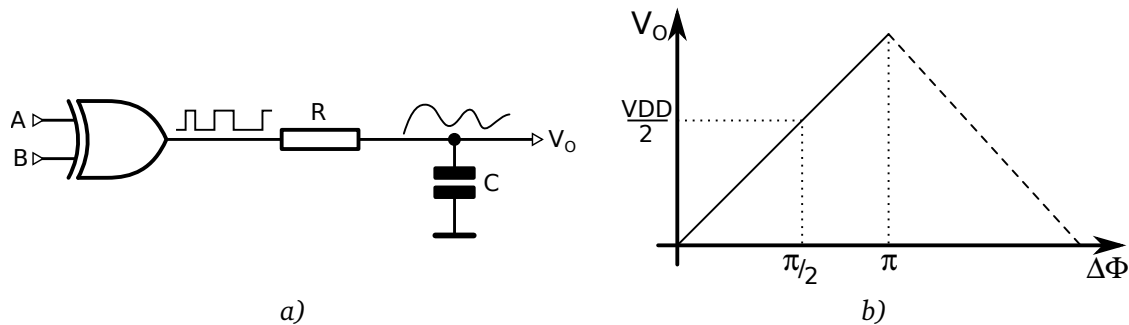


Figure 2.5: Using of the LPF for average the error signal. a) - schematic of the Phase Detector with simplest Low-Pass Filter, b) - oscillator control voltage (V_o) as a function of phase difference at the PD input (average value).

It is worth to notice, that when the square waves differ by exactly one fourth of the period (phase difference $\Delta\phi = \pi/2$), the phase detector output gives the signal with a duty cycle equal to 50%. In such case the RC filter output V_o will be set to half of the supply voltage. The value of the filter output voltage is represented by the following formula:

$$V_o = V_{DD} \cdot \frac{\Delta\phi}{\pi} = K_p \cdot \Delta\phi \quad (\text{for } \Delta\phi < \pi) \quad (2.1)$$

where: V_o - output voltage of the RC filter
 $\Delta\phi$ - phase difference at the PD inputs
 K_p - Phase Detector gain

The Phase Detector gain for the given example may be calculated as:

$$K_p = \frac{V_{DD}}{\pi} \left[\frac{V}{\text{radian}} \right] \quad (2.2)$$

For a better understanding of this formula figure 2.5 can be helpful. When two signals at the input of the PD are in the same phase, the V_O voltage is equal to 0. In such case the Voltage-Controlled Oscillator generates lower frequency than the reference clock and the phase relation between signals (connected to the PD inputs) changes. The rising phase difference $\Delta\phi$ causes an increase in the average value of the V_O voltage and increases the output frequency, generated by the oscillator. As a result the value of $\Delta\phi$ decreases (negative feedback). An example of a stable PLL operation is the condition where the voltage $V_O = V_{DD}/2$. Such a situation is natural for proper PLL operation, but only when the oscillator control voltage (V_O), equal to $V_{DD}/2$ (V_{center}), will force the frequency (f_{center}) equal to the reference frequency. When the PLL feedback loop is closed, the output signal from the oscillator will be shifted in phase by 1/4 of its period, relative to the reference signal, because only then the error signal from the Phase Detector will force the generation of the frequency equal to the reference frequency and the PLL will be synchronized. The circuit shown in figure 2.6

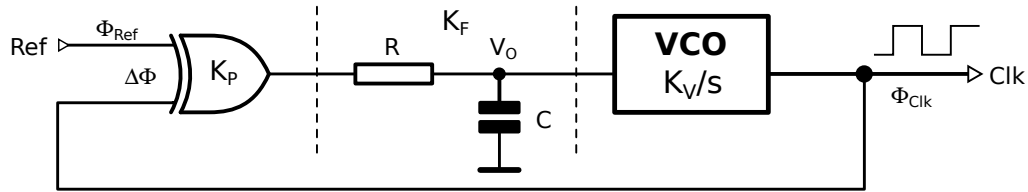


Figure 2.6: Block diagram of the PLL with the XOR Phase Detector.

in literature is referred as the type I Phase-Locked Loop [61]. The phase transfer function for this circuit can be described by the following equation:

$$\phi_{Clk} = \frac{K_V}{s} K_F K_P \Delta\phi \quad s = j\omega \quad (2.3)$$

where K_F is transfer function of LPF. The VCO gain K_V is multiplied by $\frac{1}{s}$ because phase is an integral of frequency. The phase difference ($\Delta\phi$) is equal to: $\phi_{Ref} - \phi_{Clk}$, so the equation can be evaluated as:

$$\phi_{Clk} = \frac{K_V}{s} K_F K_P (\phi_{Ref} - \phi_{Clk}) \quad (2.4)$$

After simple transformations the phase transfer function $H(s)$ can be expressed as:

$$H(s) = \frac{\phi_{Clk}}{\phi_{Ref}} = \frac{K_P K_F K_V}{s + K_P K_F K_V} \quad (2.5)$$

As known from the Laplace transform: $\phi = \frac{1}{s} \omega$, and so the phase of the VCO output signal ϕ_{Clk} depends on frequency of the reference signal ω_{Ref} as given by the following formula:

$$\phi_{Clk}(s) = \frac{K_P K_F K_V}{s + K_P K_F K_V} \cdot \frac{\omega_{Ref}(s)}{s} \quad (2.6)$$

Further analysis of the type I PLL can be simplified by omitting the transfer function of the filter ($K_F = 1$). In such case calculations are easy, and at the same time all important parameters are preserved. The PLL response for the input frequency step ω_i is obtained as:

$$\phi_{Clk}(s) = \frac{K}{s + K} \cdot \frac{\omega_i}{s \cdot s} \quad \omega_{Ref} = \frac{1}{s} \cdot \omega_i \quad K = K_P K_V K_F \quad (2.7)$$

Using the inverse Laplace transform, the solution in the time domain $\phi_{Clk}(t)$ can be expressed as follows:

$$\begin{aligned}\phi_{Clk}(t) &= \lim_{s \rightarrow -K} \frac{(s+K)K\omega_i \cdot e^{st}}{(s+K)s^2} + \lim_{s \rightarrow 0} \frac{d}{ds} \left(\frac{s^2 K \omega_i \cdot e^{st}}{(s+K)s^2} \right) = \\ &= \lim_{s \rightarrow -K} \frac{K\omega_i \cdot e^{st}}{s^2} + \lim_{s \rightarrow 0} \frac{K\omega_i t (s+K) \cdot e^{st} - K\omega_i \cdot e^{st}}{(s+K)^2} = \\ &= \frac{\omega_i}{K} \cdot e^{-Kt} + \frac{K^2\omega_i t - K\omega_i}{K^2} = \omega_i t + \frac{\omega_i}{K} \cdot e^{-Kt} - \frac{\omega_i}{K}\end{aligned}\quad (2.8)$$

Finally, the phase change of the output signal $\phi_{Clk}(t)$ in the time domain, in response to frequency step ω_i of the reference signal, will be expressed as:

$$\phi_{Clk}(t) = \omega_i t + \frac{\omega_i}{K_p K_V K_F} \cdot e^{-K_p K_V K_F t} - \underbrace{\frac{\omega_i}{K_p K_V K_F}}_{\text{phase offset}} \quad (2.9)$$

Equation 2.9 shows that PLL introduces a phase shift (offset) dependent on the gain value $K_p K_V K_F$. The output angular frequency $\omega_{Clk}(t)$ in the time domain is given by the time derivative of $\phi_{Clk}(t)$, which gives the following formula:

$$\omega_{Clk}(t) = \frac{d\phi_{Clk}(t)}{dt} = \omega_i - \underbrace{\omega_i \cdot e^{-K_p K_V K_F t}}_{= 0 \text{ for } t \rightarrow \infty} \quad (2.10)$$

The exponential part of this formula approaches 0 when time goes to infinity. The change in the output frequency follows the step of input frequency. To show how the filter parameters affect the Phase-Locked Loop stability, equation 2.5 should be expanded by the filter transfer function. The transfer function $K_F(s)$ of the simplest Low-Pass Filter (RC) is described by the following formula:

$$K_F(s) = \frac{1}{1 + sRC} \quad (2.11)$$

Introducing new variables ω_n - natural angular frequency, ξ - damping factor:

$$\omega_n = \sqrt{\frac{K_p K_V}{RC}} \quad \xi = \frac{1}{2} \cdot \sqrt{\frac{1}{K_p K_V RC}} \quad (2.12)$$

Equation 2.5 can be expressed as follows:

$$H(s) = \frac{\phi_{Clk}}{\phi_{Ref}} = \frac{\omega_{Clk}}{\omega_{Ref}} = \frac{\omega_n^2}{s^2 + 2\xi\omega_n \cdot s + \omega_n^2} \quad (2.13)$$

Further analysis proceeds in the same way as in the previous case, and its purpose is to calculate the circuit response in time domain to the frequency step ω_i of the reference signal. The output angular frequency is given by:

$$\omega_{Clk}(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n \cdot s + \omega_n^2} \cdot \frac{\omega_i}{s} \quad \omega_{Ref}(s) = \frac{\omega_i}{s} \quad (2.14)$$

The output function has three poles, one of which at $s_0 = 0$, and the other two should be calculated from a quadratic equation. Delta of this equation is obtained as:

$$\Delta = 4\xi^2\omega_n^2 - 4\omega_n^2 \quad \sqrt{\Delta} = 2\omega_n\sqrt{\xi^2 - 1} \quad (2.15)$$

Depending on the value of the damping factor ξ^2 there are three possible solutions:

$$s_{1,2} = \begin{cases} -\xi\omega_n & \text{for } \xi^2 = 1 \\ -\xi\omega_n \pm \omega_n\sqrt{\xi^2 - 1} & \text{for } \xi^2 > 1 \\ -\xi\omega_n \pm j \cdot \omega_n\sqrt{1 - \xi^2} & \text{for } \xi^2 < 1 \end{cases} \quad (2.16)$$

For $\xi^2 > 1$ the circuit response to input frequency step ω_i , gives an exponential change in output frequency. For the case where $\xi^2 = 1$, the frequency of the PLL output also follows exponentially the change in the input frequency, but the changes are the quickest. In the first two cases, the Phase-Locked Loop operation is always stable and oscillations in frequency domain cannot occur. In the last case ($\xi^2 < 1$), the most important from the point of view of stability analysis, the function given by equation 2.14 has a pair of complex conjugate poles and can be expressed as:

$$\omega_{clk}(s) = \frac{\omega_n^2}{(s + \xi\omega_n + j \cdot \omega_n\sqrt{1 - \xi^2})(s + \xi\omega_n - j \cdot \omega_n\sqrt{1 - \xi^2})} \cdot \frac{\omega_i}{s} \quad (2.17)$$

Using the inverse Laplace transform the $\omega_{clk}(t)$ is calculated as follows:

$$\begin{aligned} \omega_{clk}(t) &= \omega_i + 2\Re \left[\lim_{s \rightarrow -\xi\omega_n - j\omega_n\sqrt{1-\xi^2}} \left(\frac{\omega_n^2\omega_i \cdot e^{st}}{s(s + \xi\omega_n - j\omega_n\sqrt{1-\xi^2})} \right) \right] = \\ &= \omega_i + \Re \left[\frac{\omega_i \cdot e^{(-\xi\omega_n - j\omega_n\sqrt{1-\xi^2})t}}{(\xi + j\sqrt{1-\xi^2})(j\sqrt{1-\xi^2})} \right] = \omega_i + \Re \left[\frac{\omega_i \cdot e^{(-\xi\omega_n - j\omega_n\sqrt{1-\xi^2})t}}{e^{j \cdot \arctan \frac{\sqrt{1-\xi^2}}{\xi}} \cdot \sqrt{1-\xi^2} \cdot e^{j \cdot \frac{\pi}{2}}} \right] = \\ &= \omega_i + \omega_i \cdot \frac{e^{-\xi\omega_n t}}{\sqrt{1-\xi^2}} \cdot \Re \left[e^{-j \left(\omega_n\sqrt{1-\xi^2} \cdot t + \arctan \frac{\sqrt{1-\xi^2}}{\xi} + \frac{\pi}{2} \right)} \right] = \\ &= \omega_i - \omega_i \cdot e^{-\xi\omega_n t} \cdot \frac{1}{\sqrt{1-\xi^2}} \cdot \sin \left(\omega_n\sqrt{1-\xi^2} \cdot t + \arctan \frac{\sqrt{1-\xi^2}}{\xi} \right) \end{aligned} \quad (2.18)$$

It is worth to notice that the output frequency $\omega_{clk}(t)$, after some time, reaches the same value as the reference frequency ω_i , because the fluctuations are suppressed by the factor $e^{-\xi\omega_n t}$. Figure 2.7a shows an example of the PLL output response to the unit step of the reference frequency. On the basis of formulas 2.12 it can be shown that:

$$\xi\omega_n = \frac{1}{2} \cdot \omega_{LP} \quad (2.19)$$

where $\omega_{LP} = \frac{1}{RC}$ is cutoff angular frequency of the LPF. As a result an optimum solution between a good filtration of the VCO control voltage and a low oscillation in the frequency domain at the output of the Phase-Locked Loop is needed. Figure 2.7b shows the effect of the parameter ξ to the

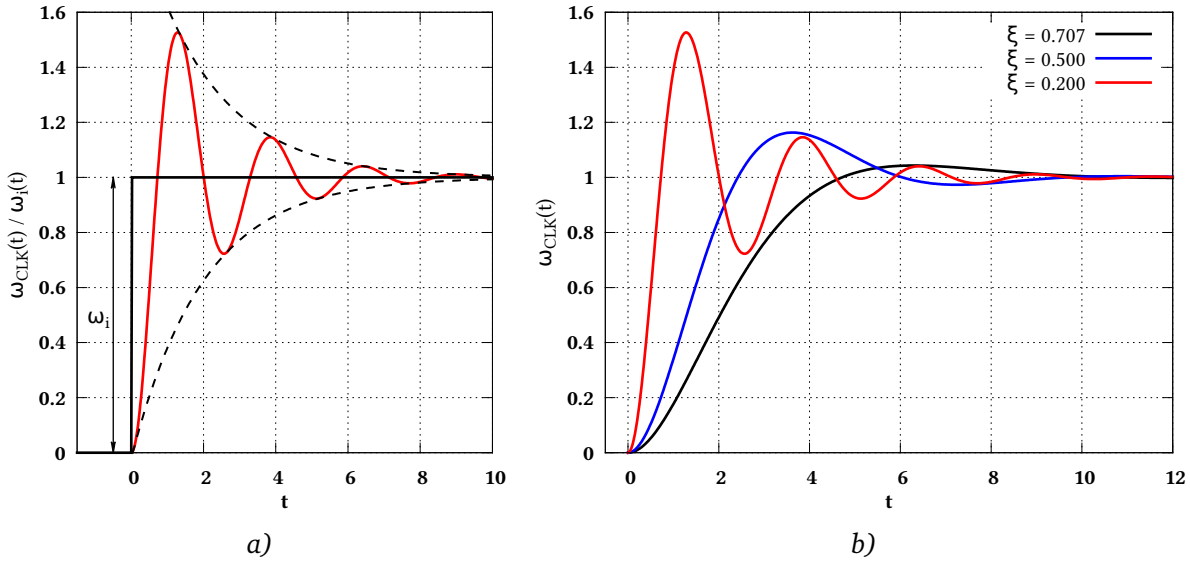


Figure 2.7: PLL response to the reference frequency step. *a)* - example for $\xi = 0.2$, *b)* - comparison of the output oscillation in frequency domain for different damping factors ξ .

PLL output response. For values of the damping factor $\xi < 0.5$, the input frequency step causes a large and slowly damped oscillations. For values of $\xi > \frac{\sqrt{2}}{2}$ the oscillations at the output are so small that they can be neglected.

The type I Phase-Locked Loop is simple in construction, however it has many disadvantages. A simple Phase Detector causes the PLL to synchronize to a harmonic frequency of the reference signal, which is undesirable. This implies the need for use of oscillators with the frequency range from $> 0.5 \cdot f_{\text{Ref}}$ to $< 2 \cdot f_{\text{Ref}}$, in order not to generate harmonic frequencies. The filter used in this type of the PLL, is the simplest Low-Pass Filter (RC). The voltage at the output of this filter oscillates even when the PLL is already synchronized, which causes the output frequency modulation. The type I PLL introduces also a phase offset depending on the loop gain.

2.1.2 Type II Phase-Locked Loop

In order to eliminate the disadvantages of the type I PLL, in the type II PLL not only the phase detection, but also the detection of the oscillation frequency is applied [64]. As a result, it is not possible to synchronize the loop to a harmonic frequency of the reference signal. In such case the Voltage-Controlled Oscillator may have a wide tuning range.

Theoretically speaking, there are two feedback loops in the type II PLL, the first is related to the PD, and the second is responsible for the frequency detection. The principle of Frequency Detector (FD) operation is to compare the output frequency of the VCO with the reference frequency and to generate a corresponding voltage signal. When the frequency error $\omega_{\text{out}} - \omega_{\text{in}}$ is small, the PD starts to work and adjusts the voltage at the VCO input, so that the phases of the output signal and reference signal are the same. The result of such an action is the PLL output perfectly synchronized to the reference signal in phase and frequency [61, 63].

Typically, the PD and the FD are combined in one circuit, which is called Phase and Frequency Detector (PFD). This circuit works properly when periodic signals are supplied to its inputs, because the PFD operation is based on the detection of the rising (or falling) edges of the two signals. Figure 2.8 shows the principle of PFD operation. The PFD is based on sequential logic and works as follows

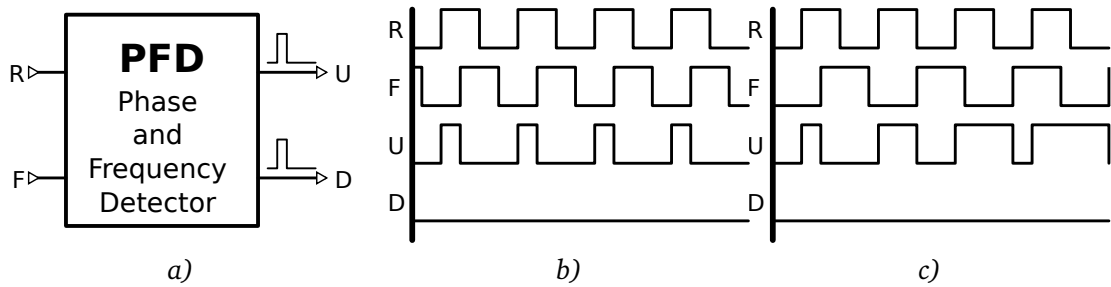


Figure 2.8: Principle of the PFD. a) - PFD symbol, b) - response to the input signals R and F differing only in phase, c) - response to the input signals R and F differing in frequency.

(a more detailed description of the PFD is presented in section 2.3.2). In the initial state both outputs U (up) and D (down) are at low level. The first rising edge at R (reference) input sets the high level on U . This persists until the F (feedback) input rising edge arrives. After that everything returns to the initial state. If the F input rising edge arrives before the R input, the pulses are set on D output. When two input signals are different only in phase (figure 2.8b), one of the PFD outputs generates pulses with a constant width. This situation continues until the output of the PLL is synchronized to the reference signal. In the other case, if the input signals frequencies are different (figure 2.8c), the pulse width on the PFD output increases with increasing phase shift between the signals. This gives a stronger control signal towards the restoration of the same oscillation frequency.

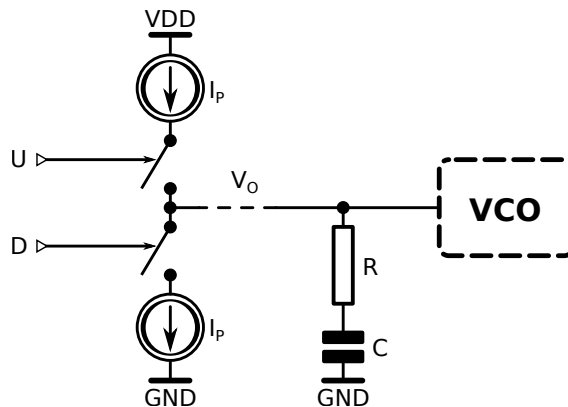


Figure 2.9: Principle of the Charge Pump operation.

The VCO is controlled by a voltage, which is the average value of the difference between signals U and D . The PFD outputs must be converted into a single signal to provide the oscillator control. In literature there are usually two types of implementation, the first is called tri-state output, and the second is a Charge Pump (CP). In most cases the second approach is used, which is presented in figure 2.9. Two switches, controlled by the signals U and D , control the flow of current I_p through the Low-Pass Filter (LPF) capacitance C .

Assuming that the two signals at the PFD inputs have the same frequency f and there is only a time delay between them (equal Δt), the phase difference can be written as:

$$\Delta\phi = \frac{\Delta t}{T} \cdot 2\pi \text{ [radian]} \quad (2.20)$$

where: Δt - time delay between signals
 T - input signals period

The average CP output current, which charges the LPF, depending on the phase difference between input signals, is expressed by simple relation:

$$I_{CP} = \frac{I_P}{2\pi} \cdot \Delta\phi = K_{PD} \cdot \Delta\phi \quad (2.21)$$

where K_{PD} is gain of the PFD. Factor 2π scales the current in such a way that the phase difference $\Delta\phi = 2\pi$ gives the CP current equal to I_P . Figure 2.10 shows a block diagram of a type II PLL. In

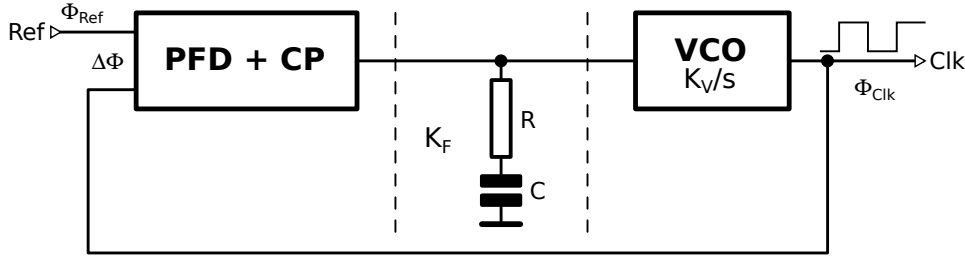


Figure 2.10: The block diagram of a type II Phase-Locked Loop.

the simplest case, a single capacitor C, without additional resistance R in series can be used as the LPF, but for the PLL stability, a serial resistor is needed, as will be shown later. Similarly to the type I PLL, a phase change ϕ_{Clk} on the oscillator output can be presented as follows:

$$\phi_{Clk}(s) = \frac{I_P}{2\pi} \cdot \left(\frac{1}{sC} + R \right) \cdot \frac{K_V}{s} \cdot \Delta\phi = \frac{I_P}{2\pi} \cdot \left(\frac{1}{sC} + R \right) \cdot \frac{K_V}{s} \cdot (\phi_{Ref} - \phi_{Clk}) \quad (2.22)$$

After simple transformations, this equation gives the phase transfer function as:

$$H(s) = \frac{\phi_{Clk}}{\phi_{Ref}} = \frac{\frac{I_P K_V}{2\pi s} \left(\frac{1}{sC} + R \right)}{1 + \frac{I_P K_V}{2\pi s} \left(\frac{1}{sC} + R \right)} \quad (2.23)$$

When the resistance value in the LPF is equal to zero, the equation can be presented in a simpler form. The phase change at the output can be expressed as follows:

$$\phi_{Clk} = \frac{\frac{I_P K_V}{2\pi C}}{s^2 + \frac{I_P K_V}{2\pi C}} \cdot \frac{\phi_i}{s} \quad \phi_{Ref} = \frac{\phi_i}{s} \quad (2.24)$$

where ϕ_{Clk} is the PLL response for input phase step ϕ_i . To simplify the notation, let $\frac{I_P K_V}{2\pi C} = K$, so:

$$\phi_{Clk} = \frac{K}{s^2 + K} \cdot \frac{\phi_i}{s} \quad (2.25)$$

The response in the time domain may be calculated by the Residuuum method, keeping in mind that the function has two conjugate complex poles and one real pole. The response in the time domain can be presented as follows:

$$\begin{aligned} \phi_{Clk}(t) &= \phi_i + 2\Re \left[\lim_{s \rightarrow j\sqrt{K}} \left(\frac{\phi_i K (s - j\sqrt{K}) \cdot e^{st}}{(s - j\sqrt{K})(s + j\sqrt{K})s} \right) \right] = \\ &= \phi_i + 2\Re \left[\lim_{s \rightarrow j\sqrt{K}} \left(\frac{\phi_i K \cdot e^{st}}{(s + j\sqrt{K})s} \right) \right] = \phi_i + 2\Re \left(\frac{\phi_i K \cdot e^{j\sqrt{K} \cdot t}}{(2j\sqrt{K})j\sqrt{K}} \right) = \phi_i + \phi_i \Re \left(\frac{K \cdot e^{j\sqrt{K} \cdot t}}{-K} \right) = \end{aligned} \quad (2.26)$$

$$= \phi_i - \phi_i \Re \left(e^{j\sqrt{K} \cdot t} \right) = \phi_i - \phi_i \Re \left(\cos \sqrt{K} \cdot t + j \cdot \sin \sqrt{K} \cdot t \right) = \phi_i \left(1 - \cos \sqrt{K} \cdot t \right)$$

After returning to the original notation, the PLL response for a step change of the reference signal phase, can be expressed by the formula:

$$\phi_{clk}(t) = \phi_i \left(1 - \cos \sqrt{\frac{I_p K_V}{2\pi C}} \cdot t \right) \quad (2.27)$$

As one can see the Phase-Locked Loop implementation with a single capacitor (without resistance in series) as the LPF is not stable. Any change of the reference signal phase causes oscillations and synchronization of the PLL is not possible. The lack of series resistance gives two perfectly integrating circuits and a total phase shift between the input and the output is equal to 180°, which naturally leads to oscillations. On the basis of formula 2.23 the PLL response for the phase step ϕ_i may be calculated in the case when $R > 0$. This situation can be presented by equation:

$$\phi_{clk}(s) = \frac{\frac{I_p K_V}{2\pi C} + \frac{I_p K_V R s}{2\pi}}{s^2 + \frac{I_p K_V R}{2\pi} \cdot s + \frac{I_p K_V}{2\pi C}} \cdot \frac{\phi_i}{s} \quad (2.28)$$

Introducing new variables ω_n - natural angular frequency, ξ - damping factor:

$$\omega_n = \sqrt{\frac{I_p K_V}{2\pi C}} \quad \xi = \frac{1}{2} \cdot \omega_n R C \quad (2.29)$$

it is possible to get an expression for $\phi_{clk}(s)$ in a convenient form for further analysis:

$$\phi_{clk}(s) = \frac{\omega_n^2 + 2\xi\omega_n \cdot s}{s^2 + 2\xi\omega_n \cdot s + \omega_n^2} \cdot \frac{\phi_i}{s} \quad (2.30)$$

Output function has three poles, one of which has at $s_0 = 0$, and the other two should be calculated by solving a quadratic equation. Depending on damping factor ξ^2 value there are three possible solutions:

$$s_{1,2} = \begin{cases} -\xi\omega_n & \text{for } \xi^2 = 1 \\ -\xi\omega_n \pm \omega_n \sqrt{\xi^2 - 1} & \text{for } \xi^2 > 1 \\ -\xi\omega_n \pm j \cdot \omega_n \sqrt{1 - \xi^2} & \text{for } \xi^2 < 1 \end{cases} \quad (2.31)$$

The first two cases always provide stable operation of the PLL, while the value of $\xi^2 < 1$ leads to the two complex conjugate poles and damped oscillations in the circuit response. In this case, the PLL output response can be presented as follows:

$$\phi_{clk}(s) = \frac{\omega_n^2 + 2\xi\omega_n \cdot s}{(s + \xi\omega_n + j \cdot \omega_n \sqrt{1 - \xi^2})(s + \xi\omega_n - j \cdot \omega_n \sqrt{1 - \xi^2})} \cdot \frac{\phi_i}{s} \quad (2.32)$$

Transformation of $\phi_{clk}(s)$ to the time domain can be obtained as follows:

$$\begin{aligned} \phi_{clk}(t) &= \phi_i + 2\Re \left[\frac{\phi_i (\omega_n^2 + 2\xi\omega_n (-\xi\omega_n - j \cdot \omega_n \sqrt{1 - \xi^2})) \cdot e^{(-\xi\omega_n - j \cdot \omega_n \sqrt{1 - \xi^2})t}}{(-j \cdot \omega_n \sqrt{1 - \xi^2} - j \cdot \omega_n \sqrt{1 - \xi^2})(-\xi\omega_n - j \cdot \omega_n \sqrt{1 - \xi^2})} \right] = \\ &= \phi_i + \phi_i \cdot e^{-\xi\omega_n t} \Re \left[\sqrt{\frac{1}{1 - \xi^2}} \cdot e^{-j \left(\omega_n \sqrt{1 - \xi^2} \cdot t + \arctan \frac{2\xi}{\sqrt{1 - \xi^2}} - \arctan \frac{\xi}{\sqrt{1 - \xi^2}} \right)} \right] = \end{aligned} \quad (2.33)$$

$$= \phi_i + \phi_i \cdot e^{-\xi \omega_n t} \cdot \sqrt{\frac{1}{1-\xi^2}} \cdot \cos\left(\omega_n \sqrt{1-\xi^2} \cdot t + \arctan \frac{2\xi}{\sqrt{1-\xi^2}} - \arctan \frac{\xi}{\sqrt{1-\xi^2}}\right)$$

The output of the type II Phase-Locked Loop behaves similarly to the type I circuit. The fluctuations damping depends on factor ξ appearing in the exponential function. When the frequencies at the input and the output of PLL are similar, the PFD adjusts the phase of the signal until the phase difference is equal to 0. It is a great advantage when comparing to the type I PLL, which gives a phase offset at output signal, dependent on the loop gain. When the phase error $\phi_{Clk} - \phi_{Ref}$ reaches 0, the PFD stops producing U and D signals and the source currents in the CP stop charging/discharging the LPF, so the voltage controlling the VCO remains constant. The frequency and the phase of the output signal will change after some period of time, because the noise occurring in the circuit will give a random modulation of the VCO frequency. Over time, the phase difference between the input and the output of the PLL may change, but negative feedback restores the proper circuit operation.

In practice, the addition of a serial resistor to the LPF causes sudden voltage changes at the LPF output as a response to the CP current pulses. These voltage steps deteriorate the stability of the loop, therefore an additional small capacitor with the capacitance around $C/10$, connected in parallel to the LPF, is typically applied. This combination eliminates voltage spikes on the VCO control line.

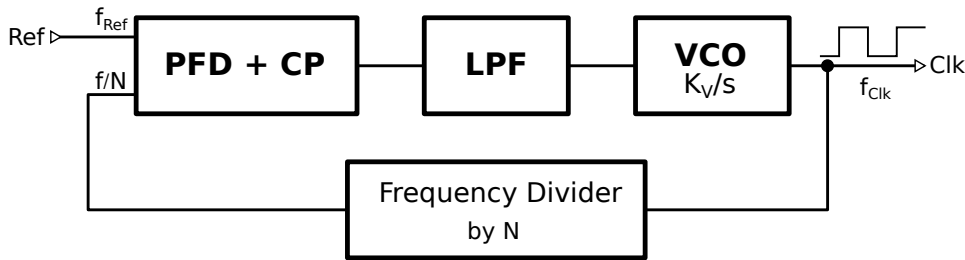


Figure 2.11: Using the PLL as a frequency multiplier.

A very important application of the type II PLL is a frequency multiplication. The principle of the PLL operation in such a configuration is shown in figure 2.11. This circuit, on the basis of a precise reference signal, generates an output signal at several times higher frequency. The key to the frequency multiplication is the use of a frequency divider in the feedback loop. The VCO must operate at a center frequency N times higher than the reference, where N is a division factor in the loop feedback. The signal frequency at the divider output is equal to the reference. The frequency division is always an exact operation, so the output signal oscillates with N times higher frequency than the reference.

2.2 Delay-Locked Loop (DLL)

In recent years PLL circuits have proven to be very useful, but their applications are focused mainly on clock frequency multiplication. In digital circuits waveforms shifted in phase are often required. It is possible to design a VCO with output signals with different clock phases, however this requires a special construction of the oscillator. An alternative and simpler approach to generate a multiple clock phases is to use a Delay-Locked Loop (DLL), applying the input clock to the Voltage-Controlled Delay Line (VCDL), which consists of few delay stages in a cascade [61, 65]. Figure 2.12 presents the principle of the DLL operation. The phase difference between Ref and $Clk4$ is sensed by the PFD, a proportional average voltage V_O is generated, and the delay of stages is adjusted by the negative feedback. More information about the VCDL can be found in subsection 2.3.1.

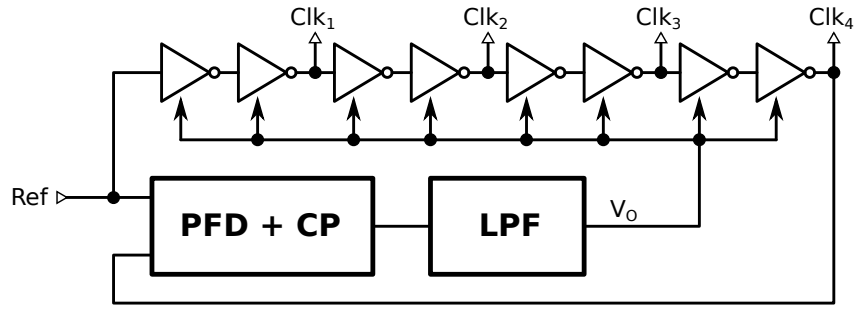


Figure 2.12: The principle of the Delay-Locked Loop operation.

After the DLL synchronization, the phase difference between *Ref* and *Clk4* is very small, ideally zero, so for the circuit presented in figure 2.12 the four stages delay the clock almost exactly by one period. The clock waveforms are shown in figure 2.13.

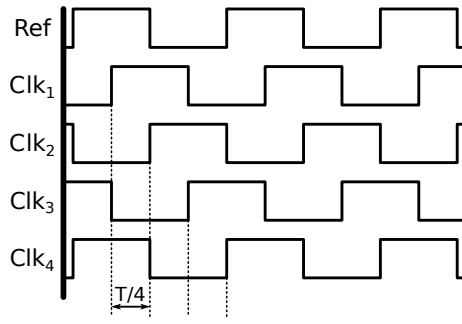


Figure 2.13: The clock phases generated by the Delay-Locked Loop.

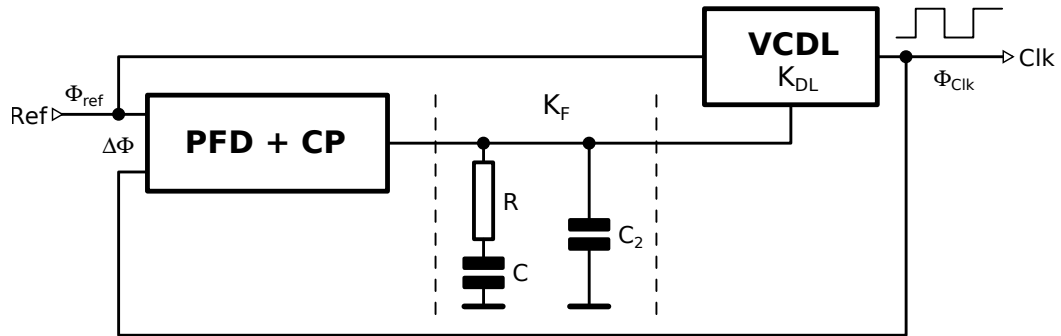


Figure 2.14: The Delay-Locked Loop block diagram.

Figure 2.14 shows a block diagram of a DLL convenient for small signal analysis. The CP output current charging the LPF is expressed by following relation:

$$I_{CP} = \frac{I_P}{2\pi} \cdot \Delta\phi = K_{PD} \cdot \Delta\phi \quad (2.34)$$

where K_{PD} is gain of the PFD.

In the simplest case, an extra capacitor C_2 can be skipped, so the phase change ϕ_{Clk} on the VCDL output can be presented as follows:

$$\phi_{Clk}(s) = \frac{I_P}{2\pi} \cdot \left(\frac{1}{sC} + R \right) \cdot K_{DL} \cdot \Delta\phi = \frac{I_P}{2\pi} \cdot \left(\frac{1}{sC} + R \right) \cdot K_{DL} \cdot (\phi_{Ref} - \phi_{Clk}) \quad (2.35)$$

After simple transformations, the phase transfer function can be expressed as:

$$H(s) = \frac{\phi_{Clk}}{\phi_{Ref}} = \frac{\frac{I_p K_{DL}}{2\pi} \left(\frac{1}{sC} + R \right)}{1 + \frac{I_p K_{DL}}{2\pi} \left(\frac{1}{sC} + R \right)} \quad (2.36)$$

Similarly to the PLL, described in section 2.1, the phase change ϕ_{Clk} on the VCDL output can be presented as follows:

$$\phi_{Clk} = \frac{\frac{I_p K_{DL}}{2\pi} \left(\frac{1}{sC} + R \right)}{1 + \frac{I_p K_{DL}}{2\pi} \left(\frac{1}{sC} + R \right)} \cdot \frac{\phi_i}{s} \quad \phi_{Ref} = \frac{\phi_i}{s} \quad (2.37)$$

where ϕ_{Clk} is the DLL response for phase step ϕ_i . Let $\frac{I_p K_{DL}}{2\pi} = K$, which will give the following equation for ϕ_{Clk} :

$$\phi_{Clk} = \frac{K(1 + sCR)}{sC + K(1 + sCR)} \cdot \frac{\phi_i}{s} = \frac{\frac{KR}{1+KR} \left(s + \frac{1}{RC} \right)}{s + \frac{K}{C(1+KR)}} \cdot \frac{\phi_i}{s}$$

The response in the time domain is calculated by the Residuuum method, keeping in mind that the function has two real poles. The response in the time domain can be presented as follows:

$$\begin{aligned} \phi_{Clk}(t) &= \phi_i + \lim_{s \rightarrow -\frac{K}{C(1+KR)}} \left(\frac{\phi_i \frac{KR}{1+KR} \left(s + \frac{1}{RC} \right) \left(s + \frac{K}{C(1+KR)} \right) \cdot e^{st}}{\left(s + \frac{K}{C(1+KR)} \right) s} \right) = \\ &= \phi_i - \phi_i RC \left(-\frac{K}{C(1+KR)} + \frac{1}{RC} \right) e^{-\frac{K}{C(1+KR)} \cdot t} \\ \phi_{Clk}(t) &= \phi_i \left(1 - \frac{1}{1+KR} e^{-\frac{K}{C(1+KR)} \cdot t} \right) \end{aligned}$$

After returning to the original notation, the Delay-Locked Loop response for input phase step, can be expressed by the formula:

$$\phi_{Clk}(t) = \phi_i \left(1 - \frac{2\pi}{2\pi + I_p K_{DL} R} e^{-\frac{I_p K_{DL}}{C(2\pi + I_p K_{DL} R)} \cdot t} \right) \quad (2.38)$$

Figure 2.15 shows the DLL response for the input signal phase step. When C_2 is skipped, the DLL gives a first order function, and there is no possibility of oscillations in the system. The serial resistor R introduces the phase steps to the output DLL response, which can be calculated as:

$$D_i = \phi_{Clk}(0) = \phi_i \left(\frac{I_p K_{DL} R}{2\pi + I_p K_{DL} R} \right) \quad (2.39)$$

In practice, in order to avoid the phase steps, it is necessary to add a small capacitor C_2 , so that the circuit becomes a second order system, and its transfer function can be evaluated as:

$$H(s) = \frac{\phi_{Clk}}{\phi_{Ref}} = \frac{\frac{I_p}{2\pi} K_{DL} (sCR + 1)}{s^2 RC C_2 + s \left(C + C_2 + \frac{I_p}{2\pi} K_{DL} RC \right) + \frac{I_p}{2\pi} K_{DL}} \quad (2.40)$$

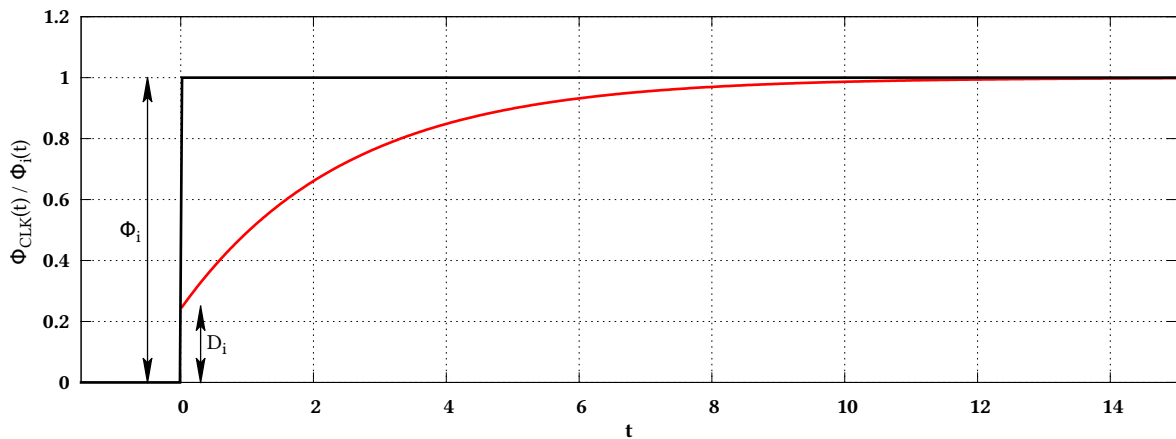


Figure 2.15: The Delay-Locked Loop response to the phase step on the input.

The transfer function obtained in this way is very similar to the PLL transfer function. As a result, the DLL output response contains three poles (one of them is $s = 0$), so there are three possible solutions, depending on the LPF transfer function and the gain of the VCDL. In the first two cases, the DLL operation is always stable and the oscillations cannot occur. In the last case, the most important from the stability point of view, the DLL response has a pair of complex conjugate poles, which may lead to oscillations in the system. More details can be found in section 2.1, which contains the Phase-Locked Loop analysis. When clock frequency multiplication is not needed the DLL circuit can be used instead of PLL. It is much simpler than PLL and provides more stable operation.

2.3 General purpose functional blocks for PLL and DLL

2.3.1 Voltage-Controlled Oscillator and Voltage-Controlled Delay Line

The Voltage-Controlled Oscillator (VCO) is a circuit, which produces a square signal in a specified frequency range. The change of the oscillation period can be done by changing the value of a voltage applied to the VCO control input V_O . The frequency at the oscillator output is a linear function of the control voltage. The Voltage-Controlled Delay Line (VCDL) circuit works similarly, but in this case a propagation delay is changed when V_O changes. The both circuits are based on the delays of logic gates, especially inverters.

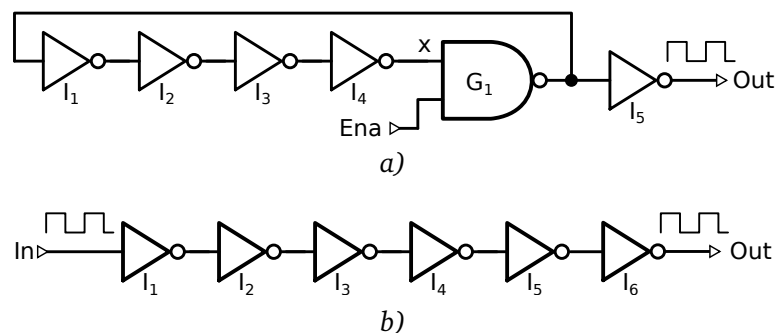


Figure 2.16: Systems based on the propagation delay of the logic gates. *a)* - Voltage-Controlled Oscillator operation principle, *b)* - Voltage-Controlled Delay Line operation principle.

There are many possible implementations of the Voltage-Controlled Oscillator [66, 67, 68, 69]. Figure 2.16a shows a gated oscillator. It provides a basis for the construction of VCO circuits, most common in the literature [70]. This circuit does not allow frequency control, but shows only the principle of square wave generation. The circuit is very simple and consists of 5 inverters and one NAND gate (G_1). A necessary condition for oscillations is an odd number of the elements negating the signal level. The inverter I_5 is the output buffer-inverter. In the idle state the Ena input is low and therefore, regardless on the G_1 input x logic state, the output of the NAND gate is always high. When the Ena goes to high state, immediately the first falling edge will occur at the G_1 output, which will start oscillations. The NAND gate works as a simple inverter in this condition. The input x of the G_1 gate, after some time, dependent on the inverters $I_1 - I_4$ delay, achieves logic state equal to the G_1 output. In this way the oscillations start and their frequency depends on: propagation time of inverters, their number, and propagation time of NAND gate.

If t_I is a delay of one inverter and t_N is a delay of NAND gate, then the oscillation frequency for the circuit presented in figure 2.16a, can be calculated by following equation:

$$f_{osc} = \frac{1}{2(4 \cdot t_I + t_N)} \quad (2.41)$$

Figure 2.16b shows a simple delay line built on inverters. The time delay is determined by the sum of the all gates propagation times. If t_I is delay of one inverter, the total delay of the circuit presented in figure 2.16b, can be calculated by simple equation:

$$t_{\Delta} = 6 \cdot t_I \quad (2.42)$$

It is worth to notice, that the period of oscillator is a linear function of gates delay. The total propagation time of delay line is linearly related with inverter delays. To obtain a VCO or a VCDL, it is necessary to control the inverters delay. For this purpose, a special design of inverter, with reduced power consumption, called the Current-Starved Inverter (CSI), is used. The schematic and symbol of this circuit are shown in figure 2.17 [61].

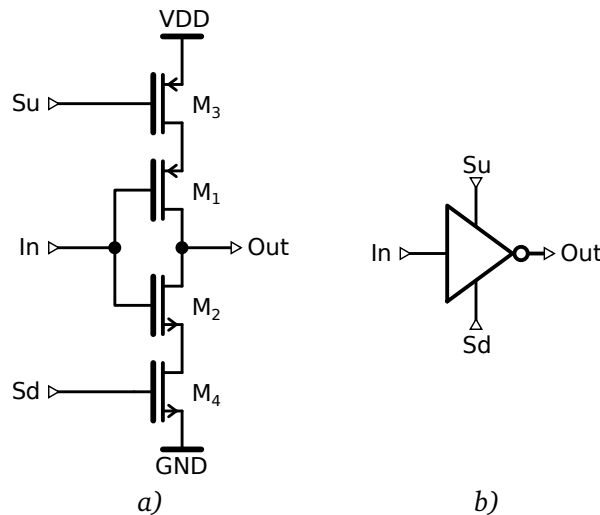


Figure 2.17: Principle of the Current-Starved Inverter operation. a) - schematic, b) - symbol.

The CSI is very simple and it can be constructed by the addition of two transistors (M_3 and M_4), which limit the current of the inverter (M_1 and M_2). The switching speed (signal propagation) depends on the capacitance connected to the inverter output and on the inverter current. In a classical

inverter with two transistors there is no possibility to control the current flowing through it. The capacitance at the output depends on the dimensions of the components and technology, so the only way to control the delay is to add the current limitation in a classical inverter (adding M_3 and M_4). The voltages on gates of transistors M_3 and M_4 (working as current sources) should not be varied freely, but the voltage drop at the control input Su should be always accompanied by the appropriate voltage rise on the Sd control input, which causes the drain currents of both sources to be the same. Such control ensures equal propagation times from low to high (T_{pLH}) and from high to low (T_{pHL}), which is important to ensure the duty cycle of the generated waveform equal 50%.

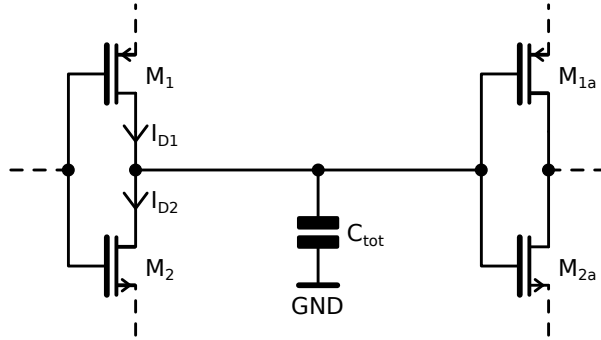


Figure 2.18: Total load capacitance of the Current-Starved Inverter.

Figure 2.18 shows the CSI total load capacitance C_{tot} , which contains the drains capacitance of transistors M_1 and M_2 , and gate capacitances of transistors M_{1a} and M_{2a} . The C_{tot} can be expressed by equation:

$$C_{tot} = C_{out} + C_{in} \quad (2.43)$$

where: C_{in} - gate capacitances of transistors in the next inverter stage
 C_{out} - drain capacitances of the inverter transistors

The time needed to charge capacitance C_{tot} from zero to VDD (supply voltage), by constant current I_{D1} is equal to:

$$t_1 = C_{tot} \cdot \frac{VDD}{I_{D1}} \quad (2.44)$$

while the time required to discharge C_{tot} from VDD to zero:

$$t_2 = C_{tot} \cdot \frac{VDD}{I_{D2}} \quad (2.45)$$

Assuming that the current $I_{D1} = I_{D2} = I_D$, the total time t_1 plus t_2 can be calculated as:

$$t_1 + t_2 = 2 \cdot C_{tot} \cdot \frac{VDD}{I_D} \quad (2.46)$$

which gives the following expression for the frequency of the VCO, consisting of N CSIs:

$$f_{osc} = \frac{1}{N(t_1 + t_2)} = \frac{I_D}{2 \cdot N \cdot C_{tot} \cdot VDD} \quad (2.47)$$

Formula 2.47 is valid for the simplest version of VCO, in which an odd number of CSIs are connected in the loop, as shown in figure 2.19. The same figure also shows the VCDL, but in this case there is not feedback, and the circuit works with even number of CSIs. Typically, two inverters are considered

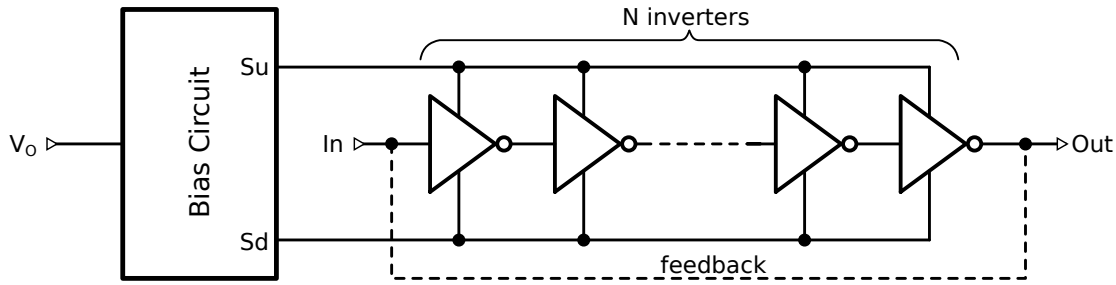


Figure 2.19: Simplest version of the **VCO** (or **VCDL**). The circuit can operate as a **VCO** when the feedback (dotted line) is connected, or as **VCDL** if there is not feedback.

in this case as one delay element. The total **VCDL** delay can be expressed by the following equation:

$$t_{del} = N \left(\frac{t_1 + t_2}{2} \right) = N \cdot C_{tot} \cdot \frac{VDD}{I_D} \quad (2.48)$$

Capacitance C_{tot} can be calculated based on the known dimensions of transistors and technological parameters, or can be simulated by Spectre or Spice simulator.

An important parameter of the **VCO** is its gain K_V , expressed in Hz/V. The K_V describes the change of the **VCO** output frequency in response to the change in the control voltage V_O . Figure 2.20a presents the output frequency as a function of control voltage. The **VCO** gain K_V is the slope of

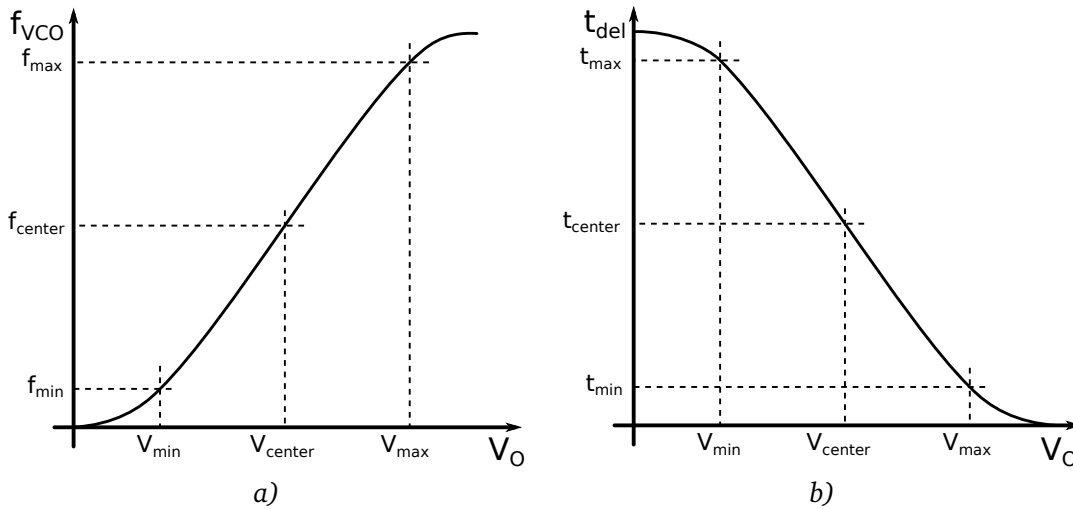


Figure 2.20: Gain of voltage controlled circuits. *a)* - **VCO** gain, *b)* - **VCDL** gain.

a linear part of the characteristic. When the control voltage reaches the V_{center} , the oscillator operates at the center frequency f_{center} . Typically, the **VCO** is designed in such a way that $V_{center} = VDD/2$. Knowing the minimum frequency (f_{min}) and the maximum frequency (f_{max}) of oscillations and the voltages corresponding to these frequencies, the formula for gain can be obtained as:

$$K_V = \frac{f_{max} - f_{min}}{V_{max} - V_{min}} \left[\frac{\text{Hz}}{\text{V}} \right] \quad (2.49)$$

An important parameter of the **VCDL** is its gain K_{DL} , expressed in s/V. Figure 2.20b presents the total **VCDL** delay as a function of control voltage V_O . For the control voltage equal to V_{center} , the **VCDL** should introduce the delay exactly equal to the period of the input signal. The gain of this

circuit can be calculated by following formula:

$$K_{DL} = \frac{t_{max} - t_{min}}{V_{max} - V_{min}} \left[\frac{s}{V} \right] \quad (2.50)$$

2.3.2 Phase and Frequency Detector (PFD)

The Phase and Frequency Detector (PFD) is a device which compares the phase and frequency of two input signals. In a type II PLLs both are required: a phase difference detection and a frequency difference detection. This reduces the time needed for the PLL synchronization and extends the range of synchronization in comparison to the type I PLL. In the PLL the feedback signal comes from the VCO or frequency divider, while in the DLL the feedback signal comes from the VC DL.

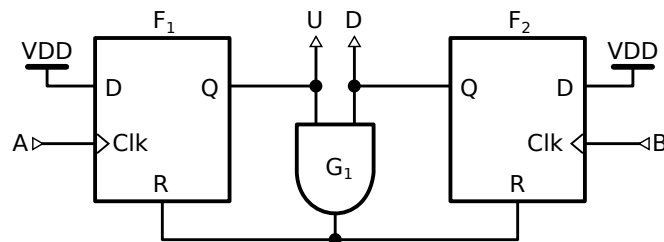


Figure 2.21: Simplified Phase and Frequency Detector block diagram.

The design of PFD, shown in figure 2.21, is very simple and comes down to two flip-flops (F_1 and F_2), operating at a rising edge, and an AND gate (G_1) [71]. The data inputs of the flip-flops are connected permanently to a high logic state (supply voltage - VDD). The circuit operation is shown

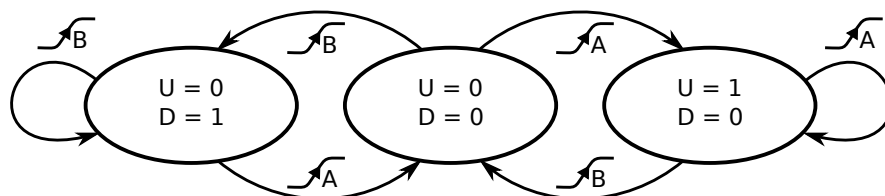


Figure 2.22: Principle of the Phase and Frequency Detector operation - state diagram.

on the state diagram presented in figure 2.22. In the initial state two output signals U and D are in low state. If input A receives the rising edge, the output Q of flip-flop F_1 is set to high state, which is equivalent to setting high state on the output U . This condition will last until the input B detects rising edge, which will cause switching of the output Q of second flip-flop (F_2) to high state. In result two flip-flops are reset by the gate G_1 and the circuit returns to its initial state. The output D works in the same way as U if the rising edge at input B comes first. In such case the rising edge at input A restores the initial state of the circuit. Thus, if the frequency of signal A is greater than the frequency of signal B , or signal A leads in phase signal B (figure 2.23a), then the output U gives pulses proportional to the phase difference. In this case output the D gives narrow pulses with width dependent on the propagation time of gate G_1 and the time needed to reset the flip-flops. On the contrary, when signal B has higher frequency than signal A , or if it is ahead in phase (figure 2.23b), then the signal proportional to the phase difference is generated at the output D . When the signals (A and B) are synchronized in phase and frequency (figure 2.23c), the PFD generates only a narrow pulses on both outputs, because the flip-flops are set at the same time and hold high state during the time required for signal propagation through the AND gate and reset of the flip-flops. The Phase

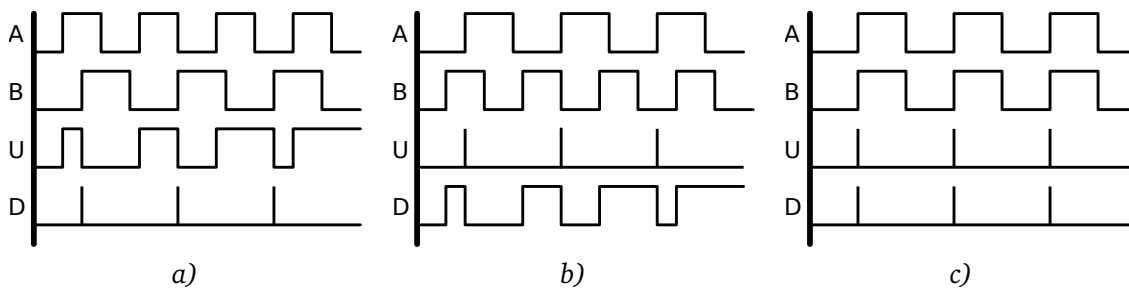


Figure 2.23: The PFD response for different phases or frequencies of input signals. *a)* - signal A has a higher frequency than B, *b)* - signal B has a higher frequency than A, *c)* - signals are synchronized in phase and frequency.

and Frequency Detector has a linear characteristic (the dependence of average difference between signals U and D on the phase difference is a straight line) for the phase differences ranging from -2π to 2π . A main disadvantage of the PFD is a "dead zone" for small phase differences between the inputs signals. The pulses generated at the outputs U and D cannot be infinitely short, so when the phase difference between two input signals approaches zero, the PFD gain also goes to zero. In result the VCO in the PLL is not tuned until the phase difference between the signals is greater than the minimum output pulse width. This effect is one of the reasons responsible for the clock jitter.

2.3.3 Charge Pump (CP) and Low-Pass Filter (LPF)

The Charge Pump (CP) circuit is cooperating directly with the PFD. Its purpose is to convert the U and D pulses from the PFD to current pulses, which charge and discharge the LPF. As a result, the CP and the LPF give the average value of the difference between the PFD outputs. The PLL filter is charged and discharged with a constant current, which value is set by the current source.

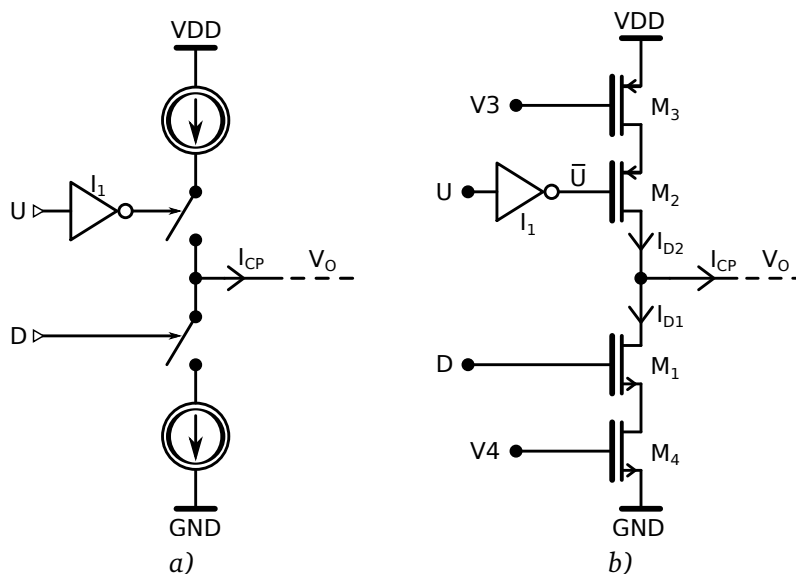


Figure 2.24: Principle of the Charge Pump operation. *a)* - block diagram, *b)* - transistors used as current mirrors and switches.

Figure 2.24a presents the principle of the CP operation and figure 2.24b shows one of the known CP schematics [61, 63]. The gates of transistors M_3 , M_4 are polarized by potentials V_3 and V_4 from external current mirrors.

The transistors M_1 and M_2 (figure 2.24b) work as switches, which are shorted when the respective PFD outputs are in high state. The inverter I_1 ensures a proper signal for the PMOS transistor M_2 . The first problem of this circuit are signals which control transistors M_1 and M_2 , because they do not arrive to the transistors at the same time. This is the result of the delay introduced by inverter I_1 . Figure 2.25a shows example waveforms occurring in the circuit. These waveforms show how the inverter delay affects the operation of the CP. The phase shift between signals \bar{U} and D

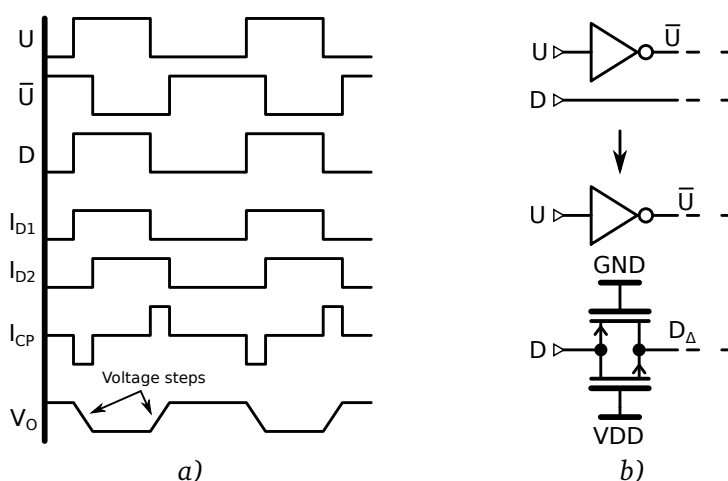


Figure 2.25: CP operation when control signals are not changing in the same time. a) - waveforms of the signals, b) - delay compensation using the transmission gate.

causes that the pulse signals do not arrive simultaneously to transistors M_1 and M_2 . When the PLL is synchronized, despite the fact that the PFD output signals are equal, the CP periodically injects (to the LPF) current pulses with the amplitude $\pm I_P$ and the width dependent on the delay between the signals (\bar{U} and D) on the gates of transistors M_1 and M_2 . This causes fluctuations of the voltage V_O , which controls the VCO, even if the PLL is already synchronized. This effect can be reduced by adding a delay to the D signal in the form of a dummy transmission gate. Figure 2.25b shows a part of the circuit after introduction of such modification. The transmission gate should be sized in such a way, that its propagation time is equal to the propagation time of the inverter. The injected current pulses are smaller when the delay between signals are approaching zero. For the same reason the V_O voltage is more stable.

Another very important issue is the difference between the drain currents of transistors M_1 and M_2 . Even when the signals on these transistors gates perfectly match in time, the V_O voltage changes, as shown in figure 2.26a. The current I_{D2} flowing through the M_2 has a higher value than the I_{D1} , which flows through the M_1 . The negative feedback makes that these currents differences are compensated by the different current pulses widths (\bar{U} is wider than D_Δ). The PFD generates pulses in such a way, that the transistor with a smaller drain current is shorted for longer time. It provides equal charge injection through transistors M_1 and M_2 , as presented in figure 2.26b. The PLL synchronized in such conditions gives a phase shift between the reference and output signal.

A third issue in the CP design is related to the parasitic capacitances of transistors (M_3 and M_4) operating as current sources. The capacitances C_x and C_y are presented in figure 2.27a. When the switches S_1 and S_2 are disconnected (idle state), transistor M_3 discharges the capacitance C_y

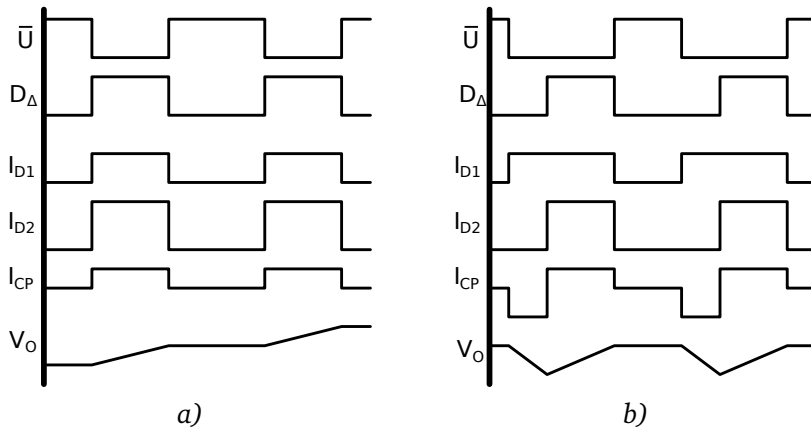


Figure 2.26: CP operation when drain currents of M_1 and M_2 transistors are different. *a)* - waveforms for case when control signals have the same width, *b)* - waveforms after the PLL synchronization.

and V_y approaches the power supply voltage. At the same time transistor M_4 discharges C_x and V_x goes to the ground potential. In the next step, when the S_2 or S_1 is shorted, voltage V_y decreases or voltage V_x increases. These potentials become equal to V_0 after some time. Even if the drain currents of both switching transistors are perfectly equal, as well as parasitic capacitances ($C_x = C_y$), the value of the voltage V_x is still different from the V_y in the time instance when the switches are closing. Recharging capacitors C_x and C_y , to voltage levels corresponding to the current value of the V_0 , causes temporary fluctuations of this voltage. This is particularly important when the control voltage level is much larger (or smaller) than the half of supply voltage. In such case there are differences in recharging times of C_x and C_y . This relation is presented in figure 2.27b. The different charge rate between the V_x and V_y causes fluctuations of the control voltage V_0 .

In order to avoid this disadvantage, it is necessary to apply a method called bootstrapping, of

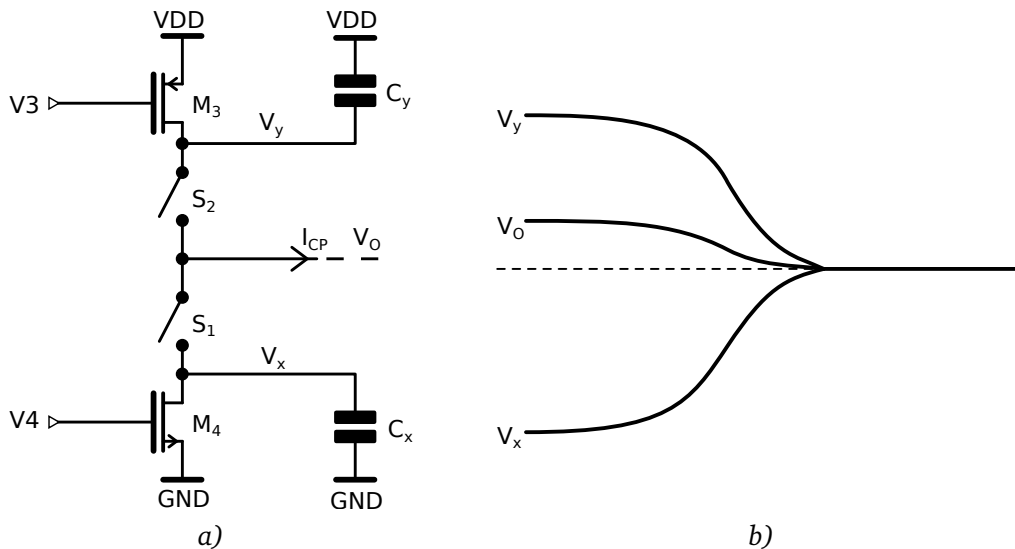


Figure 2.27: Parasitic capacitance of current mirror transistors (M_3 and M_4) influence on the CP operation. *a)* - current mirror transistors and their parasitic capacitances in the idle state (switches are disconnected), *b)* - voltages in the circuit when switches are closed.

which principle is shown in figure 2.28 [61]. In the first step (idle state) switches S_1 and S_2 are open and switches S_3 and S_4 are closed. The voltages V_x and V_y are set by the voltage follower to the value V_O . This state is maintained until switches S_1 and S_2 are closed. When switches S_1 and S_2 are closed (charge injection phase), the current flows via M_3 and M_4 , and charges or discharges the LPF. In charge injection phase switches S_3 and S_4 are disconnected to allow modification of the V_O voltage. In the next phase, after the end of charge injection (when switches S_3 and S_4 are closed), the V_x and V_y voltages are set (using a voltage follower) to the new V_O value and the whole cycle repeats itself.

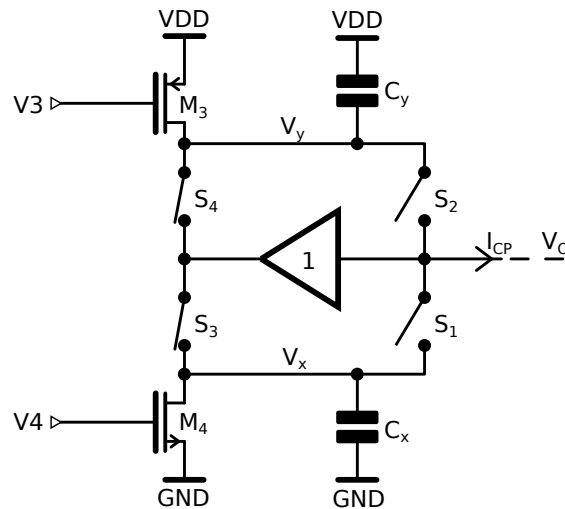


Figure 2.28: Principle of the bootstrapping operation (idle state).

Bootstrapping allows to avoid recharging parasitic capacitances from VDD (or GND) to V_O value in every charge injection phase. Therby the control voltage V_O stays constant and free from fluctuations. The voltage follower is very important, because it keeps potentials V_x and V_y at the value equal to the control voltage V_O .

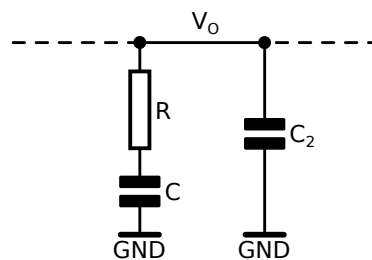


Figure 2.29: Low-Pass Filter schematic.

In the type II PLL the LPF can be very simple (figure 2.29), usually it is a serial connection of resistor and capacitor. As already discussed each of the current pulses creates a voltage step on the serial resistor in the LPF. This effect degrades the stability of the Phase-Locked Loop. This problem is solved by adding a small capacitor $C/10$ connected in parallel to the LPF output (C_2 in figure 2.29). This combination eliminates voltage spikes on the V_O , while the PLL behavior does not change significantly.

Chapter 3

Design of phase-locked circuits for SALT and other applications

Since 1980s the ASICs become very important in worldwide Integrated Circuits (ICs) marketplace, driving a new brand of expansion in semiconductor industry [72]. The ASICs respond to the needs of dedicated circuits applications. The ASICs are usually designed not by semiconductor vendor's personnel, but by someone else, who creates the custom project using a dedicated Computer-Aided Design (CAD) software. The ASICs are needed to meet the growing circuit requirements like: large scale of integration, low power and high performance. The applications of ASICs are focused on: military equipment, telecommunication devices, high speed data transmission systems and other applications, where standard integrated circuits cannot meet the requested performance. One of the most demanding areas are readout systems in HEP experiments, where large number of very dense channels needs custom solutions, with specific parameters, achieving very good performance.

The design process is largely determined by the way of ASIC prototyping. The research is usually realized in a several steps presented as follows:

- Study the nature of the problem. It consist of: literature study, theoretical studies, analyses made by the designers. It is a continuous process, during the whole project development;
- ASIC design and simulation (iterative steps). It is done using Cadence and Synopsis CAD tools. The simulations depend on the project architecture and its complexity;
- Layout design - demanding procedure, which can be done using Cadence and Mentor Graphics CAD tools. The layout of digital circuit can be manually drawn or created automatically by a dedicated software. First method is usually used to improve chip performance and allows to obtain very high frequency operation at rational power consumption. The analog layout is always manually drawn, because there is no possibility to create standard cells for analog circuits, where usually each transistor has different dimensions;
- Post-layout simulations. It is an iterative step and sometimes leads to layout redesign. It is done using Cadence, Mentor Graphics and Synopsis CAD tools;
- ASIC fabrication. The project files are sent to the producer and after some time (~ 2 months) prototype chips are received.

This chapter presents the design and simulation results of few prototype ASICs designed for HEP applications. The MULTI_PLL presented in section 3.1 is one of the first chips designed by author in 130 nm CMOS technology A for the HEP community, and it is the first PLL in this community, obtaining so low power consumption and wide frequency range. The circuit was designed as a general purpose block in two versions. In the second MULTI_PLL version the frequency range

was extended and jitter performance was improved. The two versions of SALT_PLL and two versions of SALT_DLL circuits, described in sections 3.2 and 3.3, were designed and optimized to work in SALT. The 1st versions of both circuits were designed in 130 nm CMOS technology A, while the 2nd versions were designed in 130 nm CMOS technology B. This change was motivated by the LHCb collaboration decision, regarding the base technology for ASICs in the experiment.

The author's work is mainly focused on low-power operation and improvement in jitter performance of the PLL and DLL circuits. The low jitter is very important for proper operation of electronic circuits, especially for ADCs, whose resolution depends on the sampling clock jitter. It should be pointed that the jitter value should be always considered in relation to the clock frequency. For example, in most of applications the jitter around 10 ps at frequency 160 MHz is a good value, but the same jitter at higher frequency around few GHz will be probably not acceptable. Another reason of jitter optimization is related to applications of the designed circuits. The PLL and DLL circuits integrated in complex multi-channel ASICs never work alone, because there are usually other digital circuits like clock buffers, multiplexers, etc. in the clock signal chain. All elements present in the clock path introduce some additive jitter, so the intention of the PLL/DLL designer is to obtain as low jitter as it is achievable, to provide safety margin for other circuits working in the same system. The last reason for the jitter performance improvement is related to the development of fast data serializer for the Luminosity Calorimeter (LumiCal) detector at the International Linear Collider (ILC). This activity is not directly related to the SALT project, but the proposed serializer is based on multi-phase PLL architecture designed for SALT.

The LHCb detector performance studies show that all the components in the region near the beam pipe, especially the SALT readout, need to be immune for irradiation up to around 40 MRad. The results of the radiation tests in modern 130 nm CMOS technologies [73, 74] show that the core transistors are able to survive high levels of Total Ionizing Dose (TID). All the transistors in the SALT_PLL and SALT_DLL circuits have a width few times greater than the minimal width to reduce drain-source leakage current [75]. In those circuits a standard open-layout transistors were used, but in the future the Enclosed Layout Transistors (ELT) transistors, with extremely high TID tolerance [73], should be taken in consideration.

3.1 Design and simulations of MULTI_PLL

The MULTI_PLL was designed as a general purpose block, which can operate in a very wide frequency range, from few MHz up to above 3 GHz. Obtaining such a wide frequency range is not

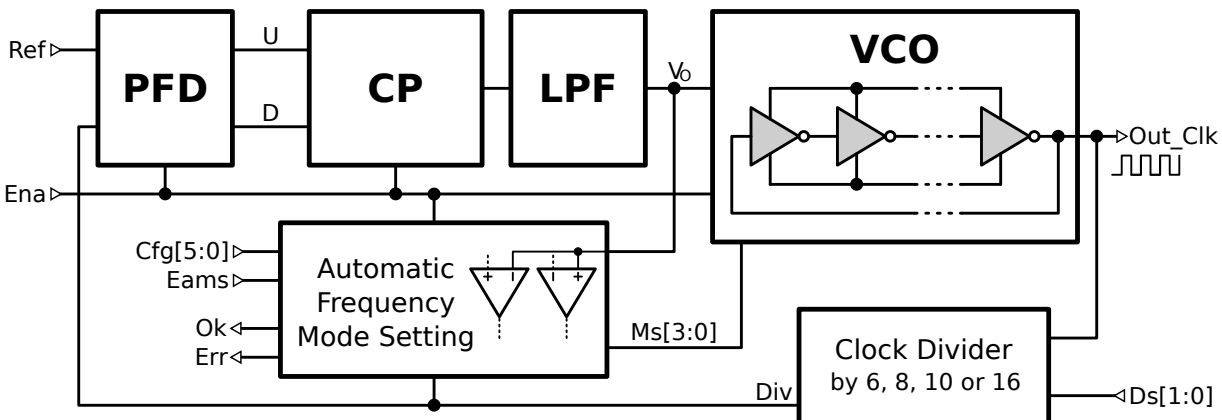


Figure 3.1: MULTI_PLL block diagram (both versions 1st and 2nd).

possible without a special Voltage-Controlled Oscillator (VCO) construction. The two PLL versions (1st and 2nd) called MULTI_PLL_V1 and MULTI_PLL_V2, were designed and simulated in 130 nm CMOS technology A. Figure 3.1 shows a block diagram of the MULTI_PLL, which is exactly the same for both circuit versions. The VCO is the most important block of the PLL and the oscillator used in the MULTI_PLL works with 16 frequency modes/ranges, which can be changed manually or automatically. More details about the VCO can be found in subsection 3.1.1. The configurable clock divider in the PLL feedback is used to obtain different frequency multiplication factors. The divider configuration can be set by $Ds[1 : 0]$, according to the detailed description in subsection 3.1.4. The divider output (Div) is compared with the reference signal (Ref) by the PFD. When the U and D signals are produced, the CP and LPF create the V_O control signal as an average value of the difference between the U and D . More detailed description can be found in subsections 3.1.2 and 3.1.3. The multiplied clock signal is available at the output Out_Clk .

The Automatic Frequency Mode Setting (AFMS) is not a standard PLL block, but it is useful to improve the circuit functionality. This block allows to change the VCO mode automatically or manually, depending on $Eams$ input. The AFMS is enabled (modes are changed automatically) when the $Eams$ signal is in high state. The meaning of the configuration signals $Cfg[5 : 0]$ depends also on the $Eams$, more information about them is presented in subsection 3.1.5. The outputs Ok and Err are directly related to the AFMS status register.

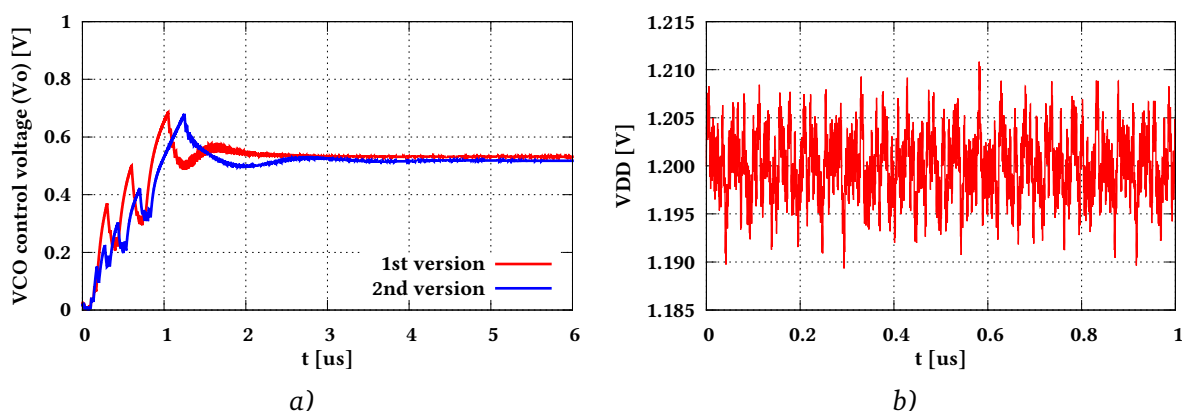


Figure 3.2: MULTI_PLL synchronization for noisy VDD. *a)* - comparison of synchronization process for two PLL versions for noisy VDD, *b)* - power supply fluctuations used in simulations.

Figure 3.2a shows the comparison of simulated transient waveforms of two MULTI_PLL versions. Their time responses are almost the same, as could be expected. One of the main goals for the 2nd circuit version was to obtain a better jitter performance, without changing the circuit functionality. All simulations were done for a noisy power supply voltage (VDD), which is presented in figure 3.2b. This voltage fluctuation is created in such a way, that its Root Mean Square (RMS) and Pk-Pk value are similar to the fluctuations measured in a real circuit. Even if this approximation is not quite accurate, it still allows to simulate the influence of VDD fluctuations on the mixed-mode circuit performance, where the standard Power Supply Rejection Ratio (PSRR) simulations cannot be done.

Figure 3.3a shows the V_O voltage during the synchronization process and presents the comparison between schematic and post-layout simulations. In both simulations the circuit response is very similar, but there is the offset voltage around 100 mV, due to the VCO parasitic components. The post-layout simulation shows that the parasitic components slow down the VCO, so a higher V_O voltage is needed to obtain the same oscillation frequency as for the schematic simulation. The

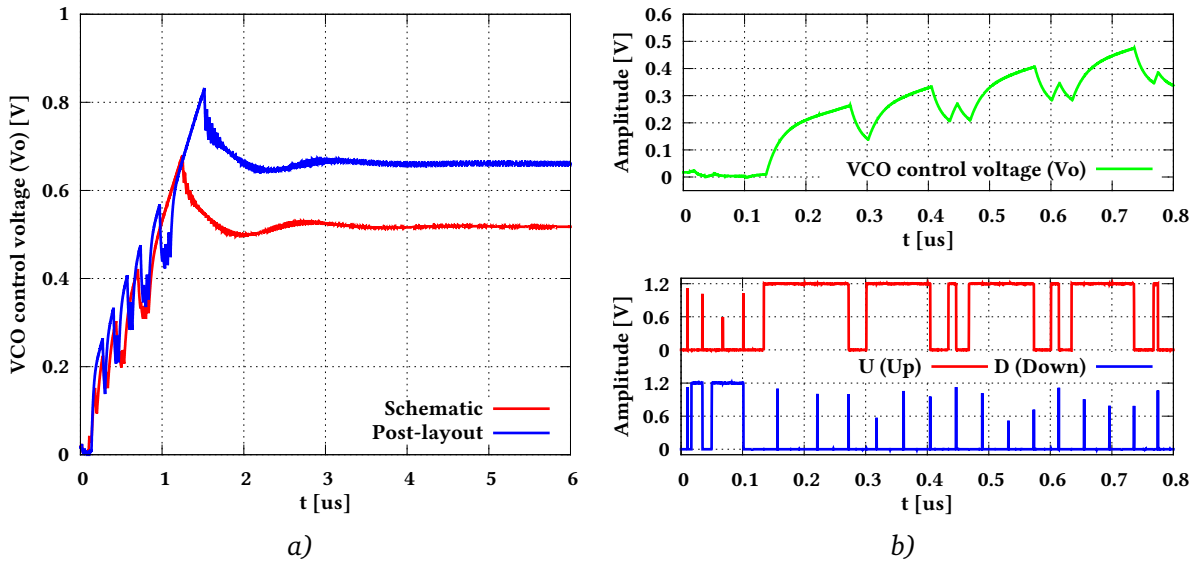


Figure 3.3: MULTI_PLL_V2 synchronization process. *a)* - comparison of the schematic and post-layout simulations, *b)* - VCO control voltage (V_o) with the corresponding signals U and D .

V_o voltage at the beginning of PLL synchronization process with the corresponding signals U and D is presented in figure 3.3b. It is worth to notice that the MULTI_PLL starts with V_o equal 0, so the reference signal Ref has higher frequency than the PLL divider output Div . In such case the U signal is in high state for most of the time, while the D gives narrow spikes (see subsections 2.3.2 and 3.1.2 for more details).

When the PLL synchronization is completed, the phase difference between the Ref and Div goes to 0. Theory says that the phase difference between these signals should be exactly equal 0, but the practical approach, presented in figure 3.4a¹, always shows a small phase difference, related to the circuit imperfections. Figure 3.4b¹ shows the U and D signals, when the PLL is synchronized.

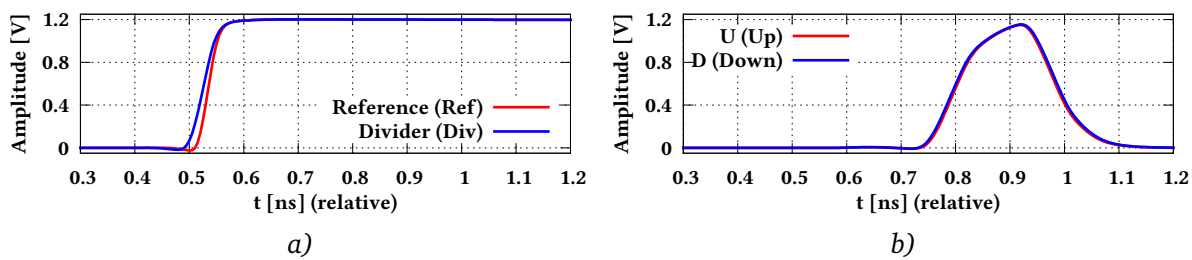


Figure 3.4: MULTI_PLL_V2 after synchronization. *a)* - reference (Ref) and divider output signal (Div), *b)* - U and D pulses.

Figure 3.5 shows the V_o voltage stability after synchronization, from the schematic and post-layout simulations. The schematic simulations show that the peak to peak fluctuations of this voltage are around 250 μV , while after the extraction of parasitic components, the V_o fluctuations increase to around 800 μV peak to peak.

¹plots in figure 3.4 is presented with x-axis offset equal 9 μs (synchronization time), for a better timescale representation in ns.

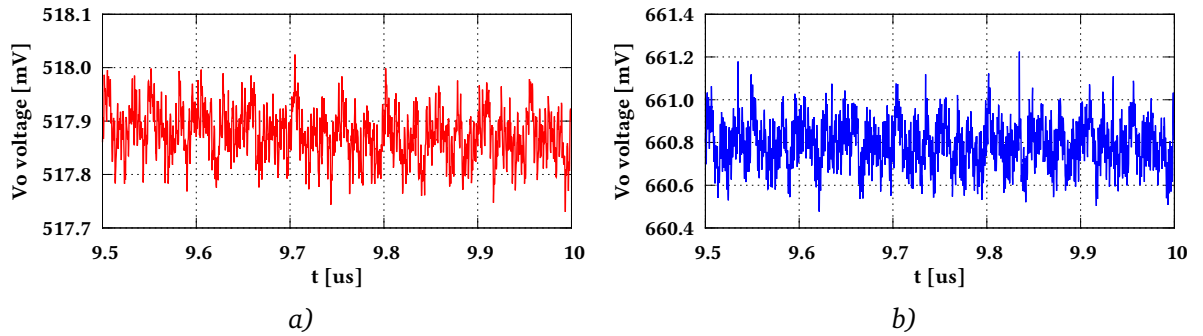


Figure 3.5: VCO control voltage (V_O) stability after MULTI_PLL_V2 synchronization. *a)* - schematic simulation, *b)* - post-layout simulations.

The important issue of the PLL simulations is a Monte Carlo (MC) analysis of the synchronization process. The MC simulations results are presented in figure 3.6, which shows 9 different MC runs. Each of the runs presents PLL synchronization for slightly different transistors dimensions. Simulations show that the circuit works properly in all cases, despite that the V_O voltage is different in each case.

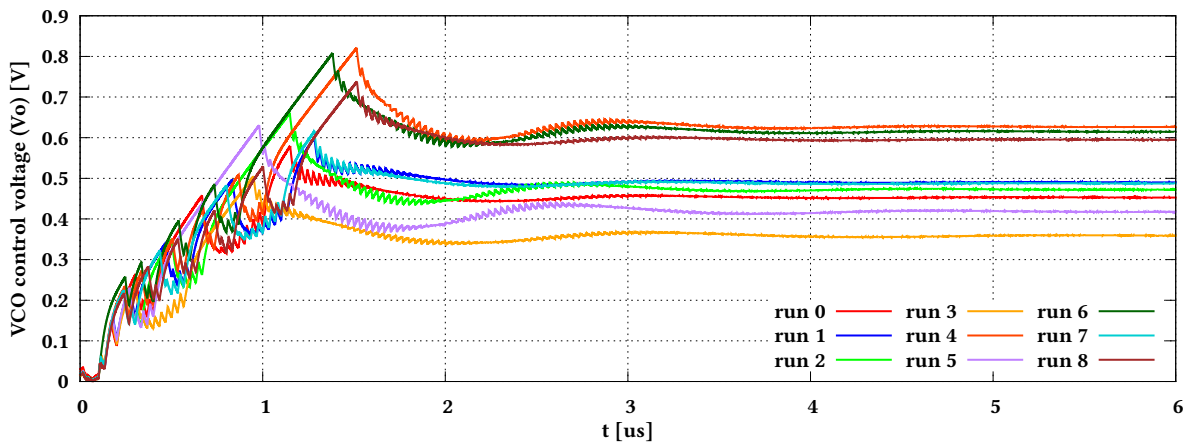


Figure 3.6: MULTI_PLL_V2 MC analysis of the synchronization process - schematic simulations.

The layout of the MULTI_PLL in both versions occupies the same area of around $260 \times 260 \mu\text{m}^2$, including the decoupling capacitors. Both circuits consume around 1 mW @ 3 GHz. The simulations show that the MULTI_PLL_V1 achieves the RMS period jitter around 56 ps at output frequency 300 MHz, while the 2nd circuit obtains value around 6.8 ps.

3.1.1 Voltage-Controlled Oscillator (VCO)

The proposed circuit, called a MULTI_VCO, contains two separate ring oscillators. There are two versions of the MULTI_VCO (1st and 2nd), designed for two versions of the MULTI_PLL. Both circuits have the same architecture, but in the second one the transistors dimensions are optimized to obtain a lower jitter. The block diagram of the MULTI_VCO is presented in figure 3.7. The switched PMOS (and NMOS) current mirrors ensure current multiplication selection, because different modes need different current ranges. Their optimization allows to reduce the PSRR, so the output frequency

becomes more independent on the power supply fluctuations. The first of the ring oscillators (slow ring) contains 5 inverters and operates in a frequency range around 6.5 MHz – 750 MHz, while the second ring (fast ring) contains only 3 inverters and works from 750 MHz up to around 3 GHz. The VCO operates in 16 modes and each of them gives different output frequency range. In each of 16 modes the mode logic sets a proper current bias, changes a current multiplication factor of switched current mirrors, configures the ring oscillators and selects a proper input of the clock multiplexer. The mode logic allows to combine all possible options in the circuit into the 4-bit bus, which makes the VCO easy to configure. The output frequency is controlled by the voltage input V_O , so from the outside the MULTI_VCO is seen as a typical Voltage-Controlled Oscillator.

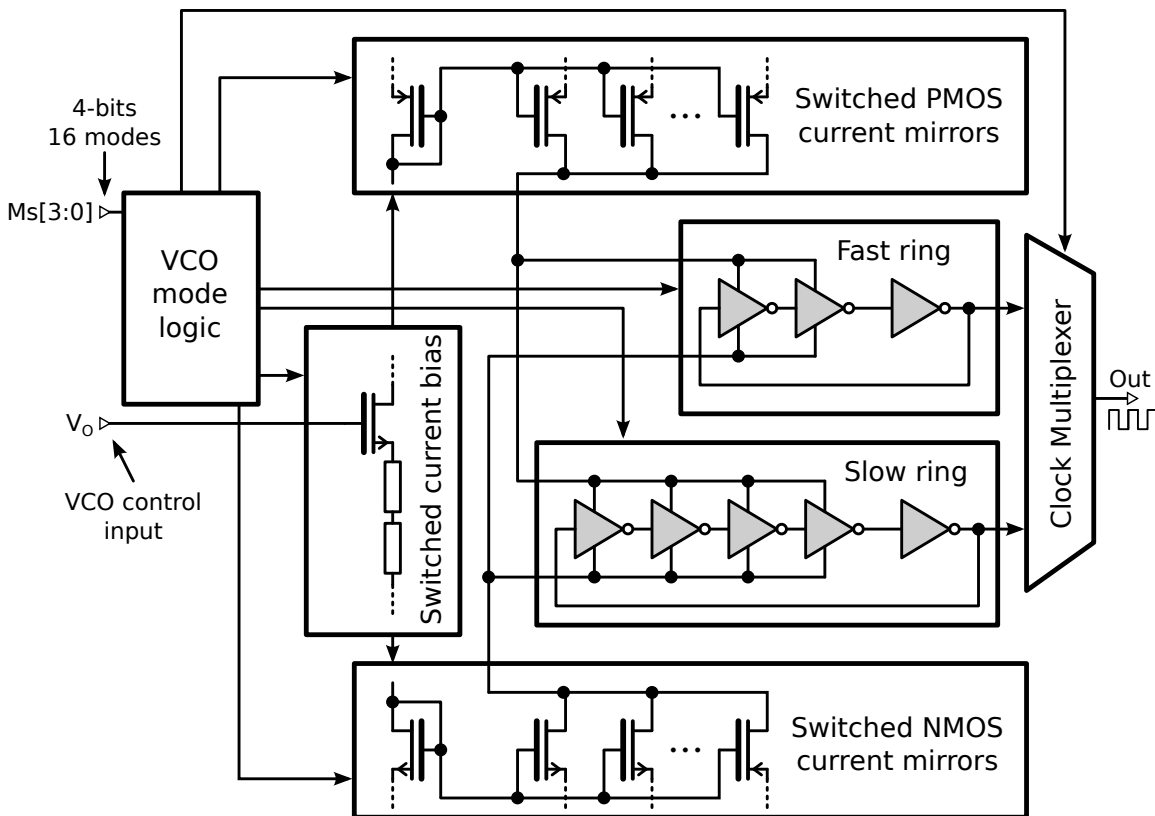


Figure 3.7: Block diagram of the Voltage-Controlled Oscillator.

Figure 3.8 shows the schematic of the fast ring. It contains 2 Current-Starved Inverters (CSIs) built on the transistors $M_1 - M_4$, and one standard inverter (transistors M_5 and M_6). The current mirrors transistors, needed for a proper CSI operation, are not shown in figure 3.8, because they are common for the two ring oscillators. The current mirrors are connected via Su and Sd terminals. The transistors M_7 and M_8 were added to provide the Ena input, which is needed to switch on/off the fast ring. When Ena goes to high state, the oscillator starts generating the output waveform. When the fast ring is disabled (Ena is low), the output always stays in high state. The fast ring oscillates in frequency range 750 MHz – 3 GHz and it is enabled when one of the modes from 8 to 15 is selected.

Figure 3.9 shows the slow ring schematic. It is much more complicated than the fast ring and contains 4 CSIs (transistors $M_1 - M_8$) and one standard inverter, built on transistors M_9 and M_{10} . As for the fast ring, the current mirrors are connected via Su and Sd terminals. The transistors M_{11} and M_{12} are added to provide the Ena input, which operates exactly in the same way as in the fast ring.

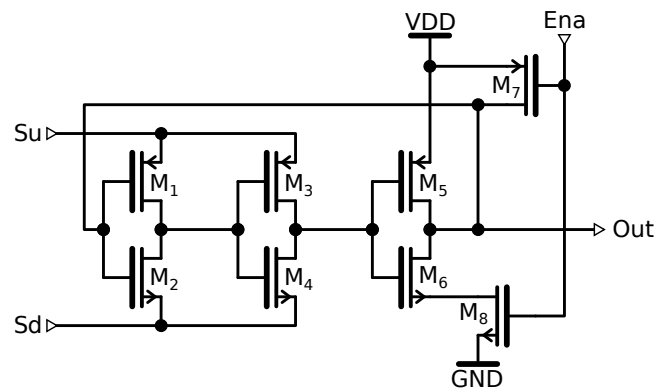


Figure 3.8: The schematic of the MULTI_VCO fast oscillation ring.

The capacitors $C_1 - C_6$ are used as an extra load for the CSIs and allow to obtain lower frequencies (below 100 MHz), when switches $M_{13} - M_{18}$ are shorted. The slow ring load is controlled by two signals ($A0$ and $A1$). When $A0$ is in low state all extra capacitors are disconnected from the oscillation ring, which can work with frequencies above 100 MHz. The high state on $A0$ connects capacitors $C_1 - C_3$ to the slow ring through transmission gates $M_{13} - M_{18}$. The negative signal $\overline{A0}$ is needed for proper operation of the transmission gates containing pairs of complementary transistors. The lowest frequencies, around 30 MHz can be obtained by switching on transistors $M_{19} - M_{21}$, which are controlled by $A1$ input. When these transistors are shorted the ring oscillator is loaded not only by $C_1 - C_3$, but also by $C_4 - C_6$, which have 4 times bigger capacitances than $C_1 - C_3$. When the slow ring is disabled (Ena is low), the logic states on inputs $A0$ and $A1$ do not matter. The slow ring

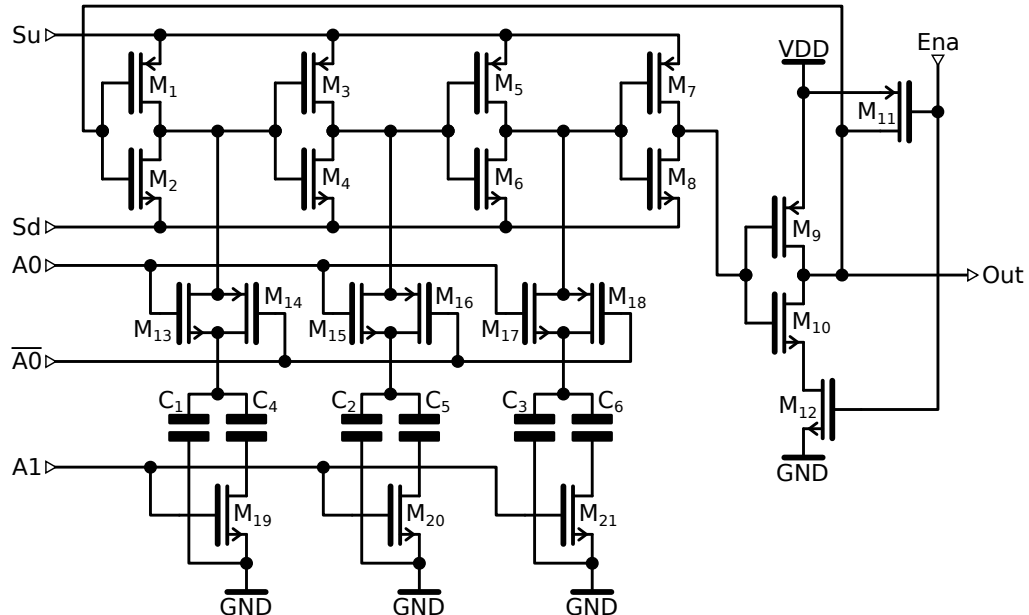


Figure 3.9: Schematic of the MULTI_VCO slow oscillation ring.

oscillates in frequency range 6.5 MHz – 750 MHz. In sub-micron CMOS technologies it is very hard to design a VCO, which can work with a low frequency, especially for frequencies below 100 MHz. Small gate capacitances in modern technologies make it much easier to design an oscillator, which works with frequencies over few hundreds of megahertz, than a low frequency one. The slow ring is enabled when one of the modes from 0 to 7 is selected.

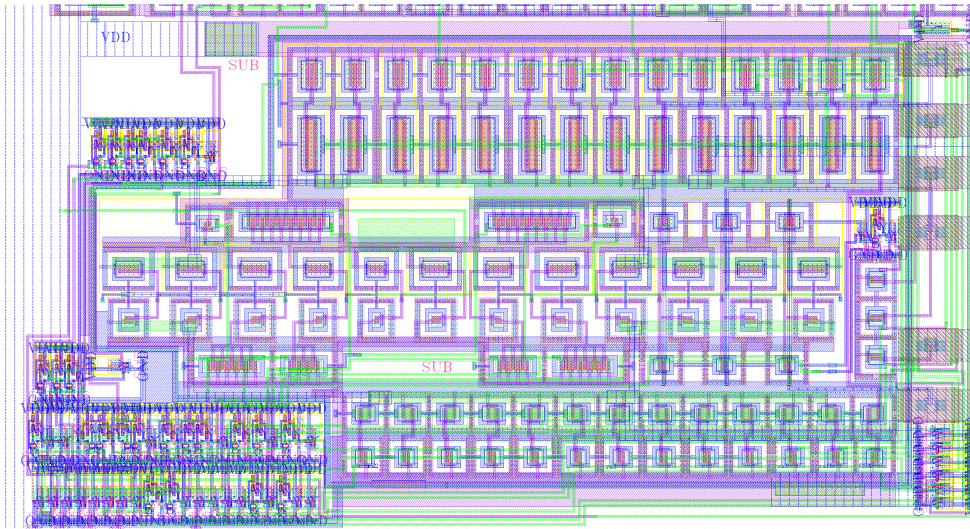


Figure 3.10: MULTI_VCO_V1 layout ($100 \times 50 \mu\text{m}^2$).

Figure 3.10 shows the layout of the 1st MULTI_VCO version. The layout for the 2nd oscillator version is very similar and is not shown. The figure shows; two oscillator rings, switched current mirrors, and mode logic which ensures a proper VCO configuration dependent on the selected mode. The layout presented in figure 3.10 is manually drawn [76] and occupies $100 \times 50 \mu\text{m}^2$ area.

Figure 3.11 presents examples of the MULTI_VCO_V1 output waveforms, obtained from the post-layout simulations, for two different frequencies. The first of them (16 MHz) is one of the lowest frequencies obtained in the circuit, while the second one (3 GHz) is near to the VCO maximum limit. The second version of the VCO (MULTI_VCO_V2) achieves very similar results and so it is not shown here. The duty cycle of output clock, presented in figure 3.12b, varies from 47.8% to 51.5% for both circuit versions. The post-layout simulations do not degrade the duty cycle, but make it even better. In these simulations the duty cycle varies from 48.8% to 51.0% for both circuit versions. In most applications the expected duty cycle should be 50%.

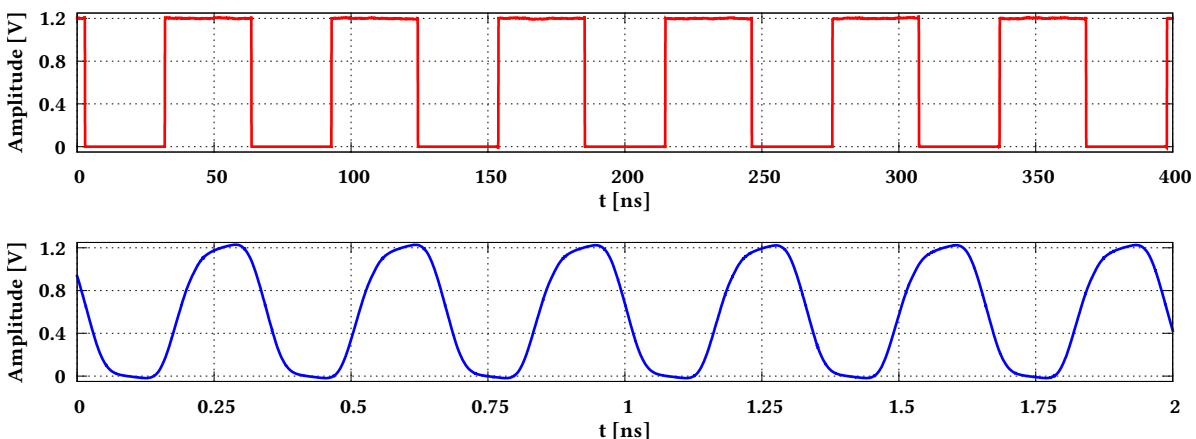


Figure 3.11: MULTI_VCO_V1 output waveforms for 16 MHz (top) and 3 GHz (bottom).

One of the main goals of the second MULTI_VCO version was to preserve the same center frequencies corresponding to the specific modes. Figure 3.12a presents how the oscillator center fre-

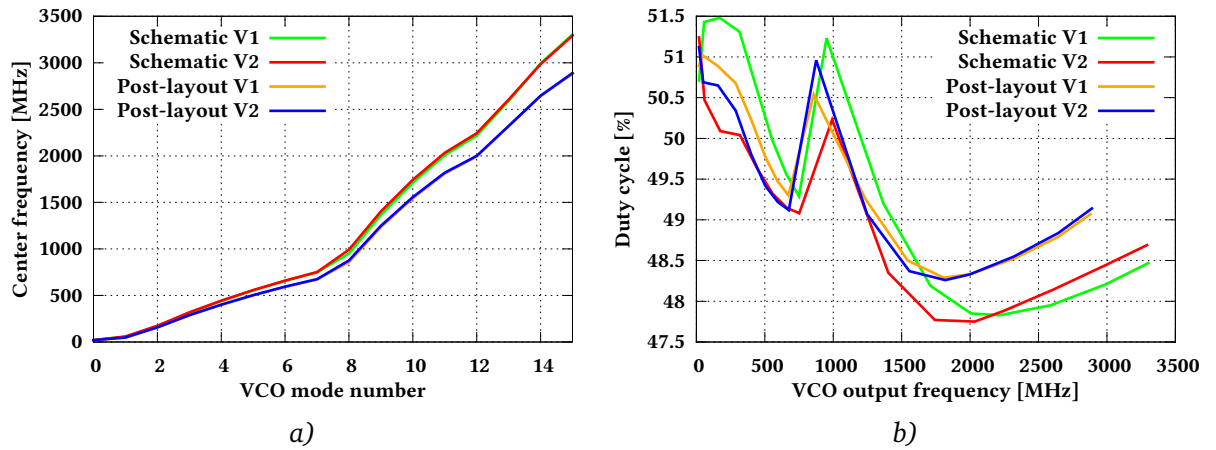


Figure 3.12: MULTI_VCO output parameters. *a)* - center output frequency vs selected mode, *b)* - duty cycle as a function of oscillation frequency.

frequency depends on the selected mode. The simulations (schematic and post-layout) show that both MULTI_VCO circuits have almost the same parameters for all modes. More details can be found in table 3.1, which presents the simulation results of the output frequency ranges for the both VCO circuits. The minimum frequency f_{min} is obtained in the case when the MULTI_VCO control voltage V_O is equal 100 mV, while the maximum frequency f_{max} is obtained for the V_O equal 1.1 V. As a result the dynamic range of the control voltage is 1 V. The gain K_V is the most important parameter of the VCO. It describes the change of the VCO output frequency in response to the change in the control voltage V_O and can be easily calculated by equation 2.49. For the lower modes (0 – 7) the MULTI_VCO achieves gain in range 30 MHz/V – 200 MHz/V, while for the higher modes (8 – 15) its gain ranges from 350 MHz/V to 650 MHz/V.

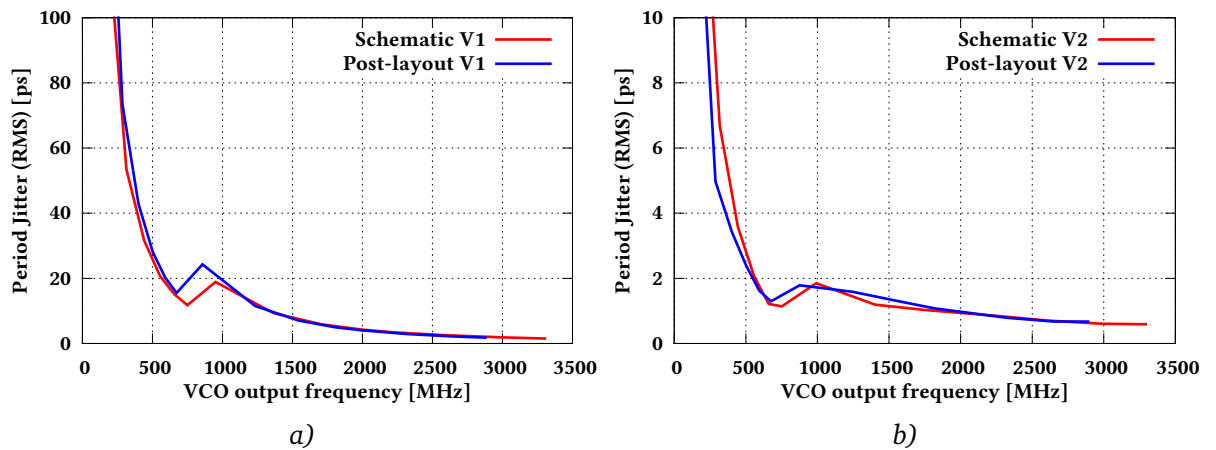


Figure 3.13: MULTI_VCO period jitter versus oscillation frequency. *a)* - 1st version, *b)* - 2nd version.

The most important parameter from the clock generation point of view is the period jitter. The simulation results for the MULTI_VCO_V1 and the MULTI_VCO_V2 are presented in figure 3.13. In the 2nd version the jitter performance is highly improved, especially for frequencies below 1 GHz. For the frequencies below 1 GHz the MULTI_VCO_V1 jitter varies from 20 ps to 100 ps, while for the same frequencies the MULTI_VCO_V2 jitter is in the range 2 ps – 10 ps.

VCO mode	Output frequency ranges of the VCO ($f_{min} - f_{max}$) [MHz]			
	Schematic simulations		Post-layout simulations	
	MULTI_VCO_V1	MULTI_VCO_V2	MULTI_VCO_V1	MULTI_VCO_V2
0	6.8 – 37.5	6.4 – 41.2	6.7 – 36.3	6.3 – 40.0
1	20 – 109	19 – 121	17 – 89	16 – 100
2	100 – 282	101 – 293	92 – 255	92 – 263
3	253 – 411	254 – 421	231 – 372	230 – 379
4	385 – 527	386 – 536	351 – 476	349 – 482
5	502 – 632	503 – 640	456 – 570	455 – 576
6	609 – 727	609 – 734	548 – 652	550 – 659
7	705 – 811	704 – 817	634 – 726	635 – 732
8	711 – 1320	738 – 1383	645 – 1186	651 – 1224
9	1169 – 1671	1197 – 1727	1058 – 1504	1064 – 1537
10	1539 – 1973	1564 – 2022	1392 – 1775	1397 – 1805
11	1854 – 2231	1874 – 2269	1675 – 2000	1678 – 2025
12	2060 – 2521	2085 – 2550	1855 – 2255	1859 – 2268
13	2461 – 2825	2414 – 2791	2208 – 2511	2210 – 2473
14	2783 – 3209	2724 – 3177	2479 – 2806	2479 – 2795
15	3013 – 3506	2948 – 3495	2660 – 3033	2656 – 3049

Table 3.1: Simulated frequency ranges for all modes of two VCO versions.

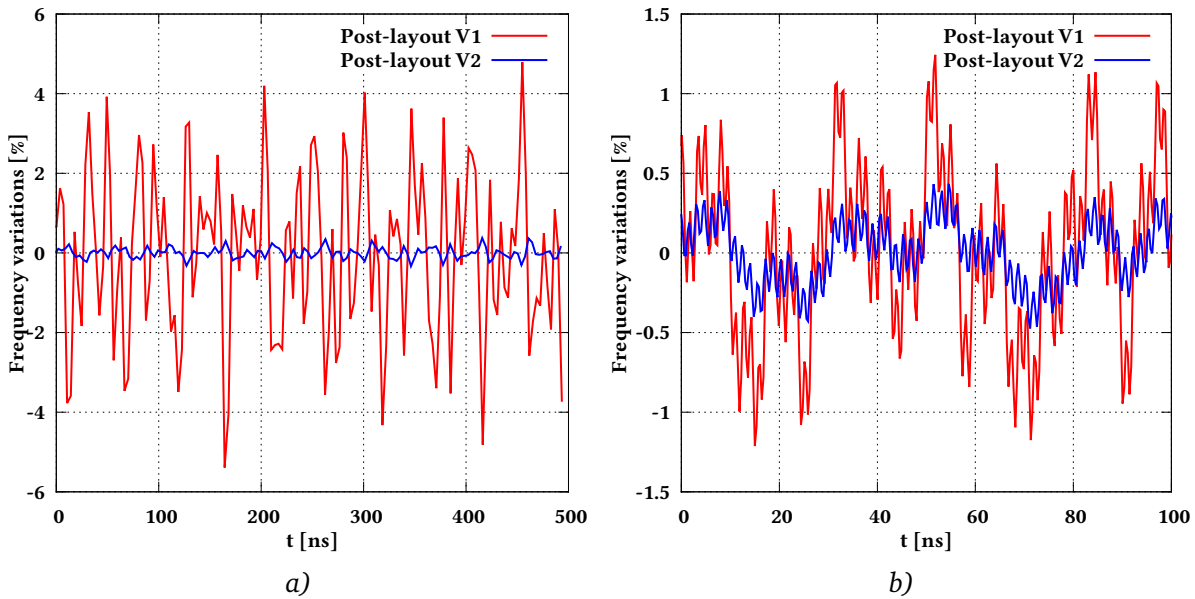


Figure 3.14: MULTI_VCO frequency stability for two circuit versions, based on post-layout simulations. a) - at frequency 300 MHz, b) - at frequency 3000 MHz.

The large jitter values cause a random frequency variations at the output. Figure 3.14 shows the stability of the MULTI_VCO output frequency for two selected frequencies 300 MHz and 3 GHz. Each of the curves on the plot presents differences, expressed in % between the average frequency and the frequency calculated for each period. Differences between the two versions of the circuit are very easy to observe at low signal frequency (300 MHz), which is shown in figure 3.14a. The fre-

quency variations of the MULTI_VCO_V1 are approximately at the level of $\pm 4\%$, while the frequency variations in 2nd version are more than 10 times better ($\pm 0.35\%$). The performance improvement at the frequency 3 GHz is also visible (figure 3.14b). In this case the 1st version gives fluctuations at the level of $\pm 1.5\%$, while the MULTI_VCO_V2 obtains a value equal $\pm 0.5\%$.

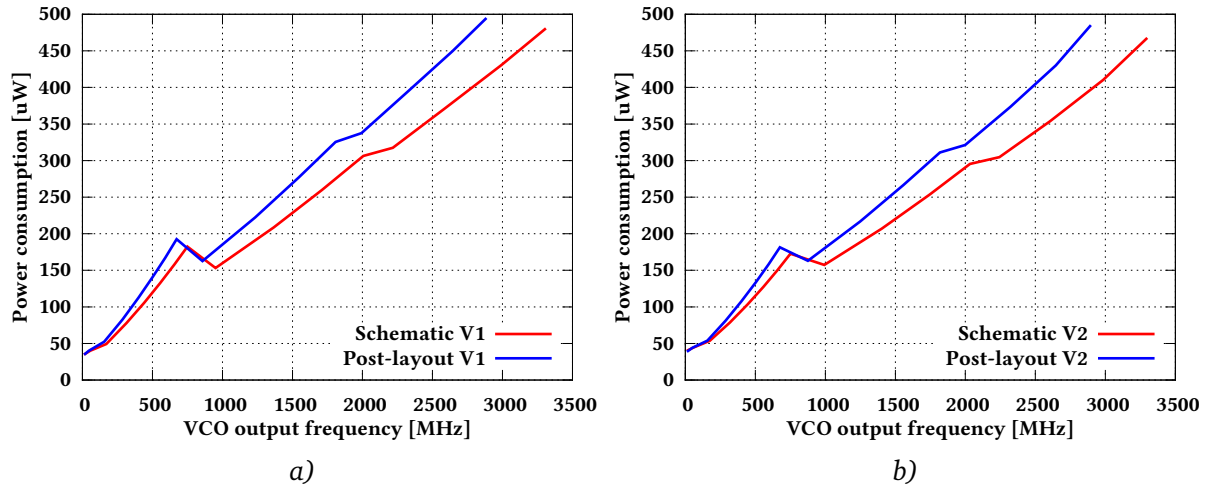


Figure 3.15: MULTI_VCO power consumption dependent on oscillation frequency. *a)* - 1st circuit version, *b)* - 2nd circuit version

Very often the power consumption of the Voltage-Controlled Oscillator circuit is a critical parameter. The VCO gives the dominant contribution to the power consumption of the whole Phase-Locked Loop. The simulations of power consumption for the both MULTI_VCO versions are presented in figure 3.15. Both oscillator versions give almost the same results. At 500 MHz the MULTI_VCO consumes around $150 \mu\text{W}$, while at 1.5 GHz the circuit power consumption is around $270 \mu\text{W}$. It is worth to notice that the power consumption is not a fully linear function of the frequency generated by the VCO, because the circuit is very complex, it has many operation modes.

3.1.2 Phase and Frequency Detector (PFD)

The Phase and Frequency Detector (PFD) is a circuit which compares the phase and frequency of two periodic signals connected to its inputs. The theoretical considerations, presented in section 2.3.2, show that a simple PFD consists of only two flip-flops and one NAND gate. Figure 3.16 presents the Phase and Frequency Detector schematic, which is used in practice, constructed using a different approach [77, 78]. There is only one version of Phase and Frequency Detector because the schematic and layout of the PFD, used in both PLLs, are exactly the same.

The PFD is symmetric and consists of two identical parts, the first to generate the U signal and the second one for the D signal. The components used in the second part are marked with additional letter "a". When the input Ena is low, the PFD circuit is disabled. The AND gate G_1 blocks the reference signal (Ref) and its output ($net0$) is in low state. The gate G_{1a} works in the same way, blocking the second input signal (Div). In each case M_1 transistor is on, so the $net1$ is shorted to VDD and transistor M_5 is off. The high state on the $net1$ turns on M_6 and turns off transistor M_4 , but it does not affect the $net2$, because M_5 is off. The $net2$ is connected to VDD by transistor M_7 , which is on when the Ena is low. In result the U signal is set to low state by the inverter M_8, M_9 . The same happen to the left part of the circuit, which generates the D signal.

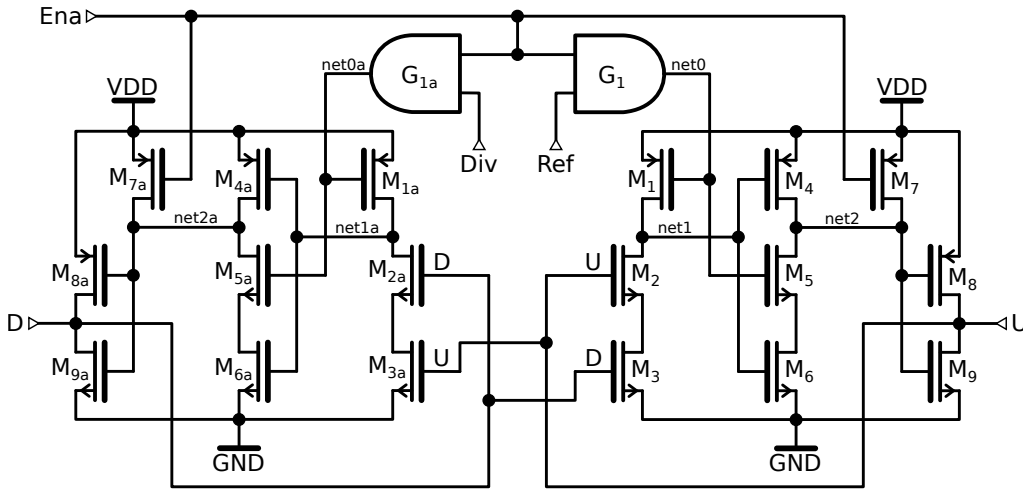


Figure 3.16: Phase and Frequency Detector schematic.

When the *Ena* is high the Phase and Frequency Detector is ready to operate. At the beginning the *U* and *D* signals are low, so transistors *M*₂ and *M*₃ are off. The rising edge on the reference signal (*Ref*) turns off transistor *M*₁ and turns on transistor *M*₅. Thanks to gates parasitic capacitances of *M*₄ and *M*₆, the logic level on *net1* remains high. When transistor *M*₅ is shorted (and *M*₆), the *net2* goes to low state and the output inverter, built with transistors *M*₈ and *M*₉, sets high state on the output *U*. At the same time the rising edge of *U* turns on transistors *M*₂ and *M*_{3a}. In this state the PFD circuit waits for the input *Div* rising edge. After that, the second part of the PFD reacts in very similar way (as transistors related to the input signal *Ref*) and the output *D* goes to high state. Immediately after that the high state on *D* turns on transistors *M*₃ and *M*_{2a} (*M*₂ and *M*_{3a} are already on), which discharge *net1* and *net1a* to ground. Low state on the *net1* turns on transistor *M*₄, *net2* goes to high state and the output signal *U* returns to 0. The behavior of the second PFD part is exactly the same, so the output signal *D* returns to low state at the same time. As a result one

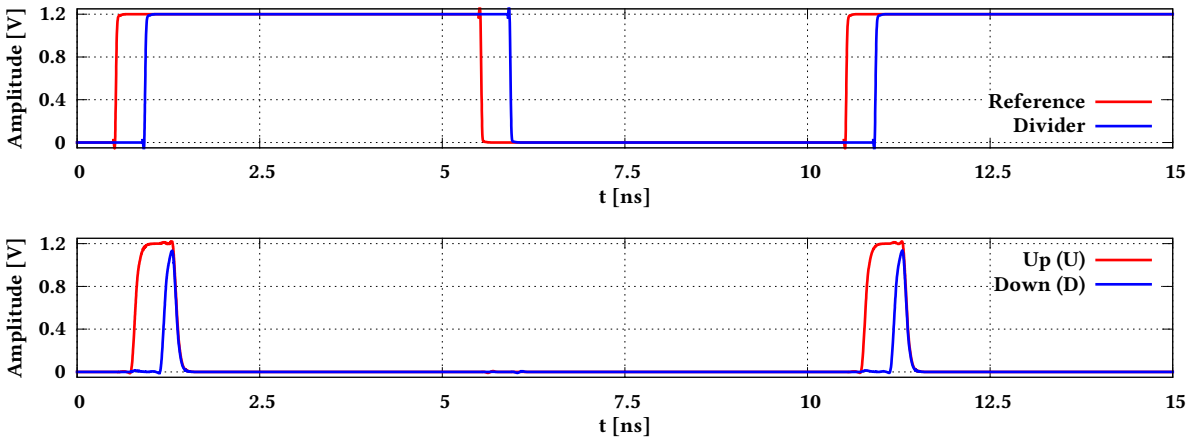


Figure 3.17: PFD output waveforms in case when the reference leads in phase.

of the PFD outputs always generates short pulses (width is determined by signal propagation in the circuit), while the second gives pulses proportional to the phase difference between the signals *Ref* and *Div*. The output waveforms in the case when the reference leads in phase is presented in figure 3.17. It presents the PFD operation at an example frequency 100 MHz, but the circuit can operate up to above 400 MHz.

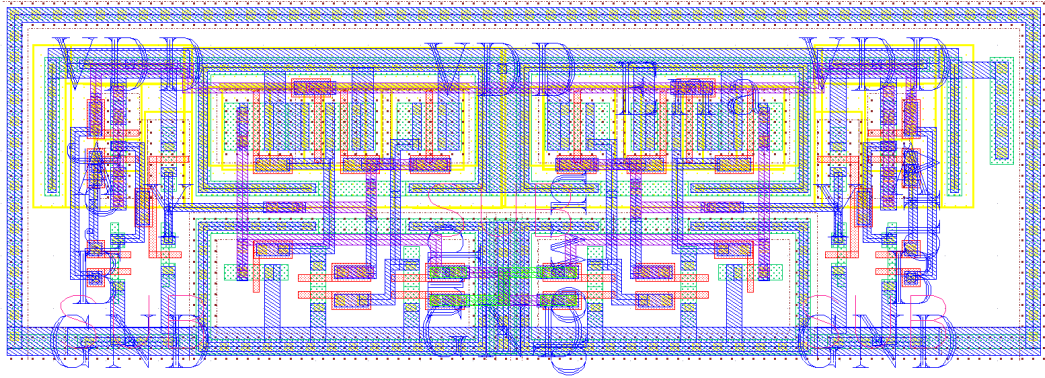


Figure 3.18: Phase and Frequency Detector layout ($20 \times 7 \mu\text{m}^2$).

The PFD layout is shown in figure 3.18 and occupies a small area around $20 \times 7 \mu\text{m}^2$. The circuit symmetry is the most important issue, which should be preserved to ensure the proper circuit operation. The layout is manually drawn and bases on the mirror symmetry [76]. The components, responsible for the U signal generation are placed in a mirror image relative to the components producing the D signal.

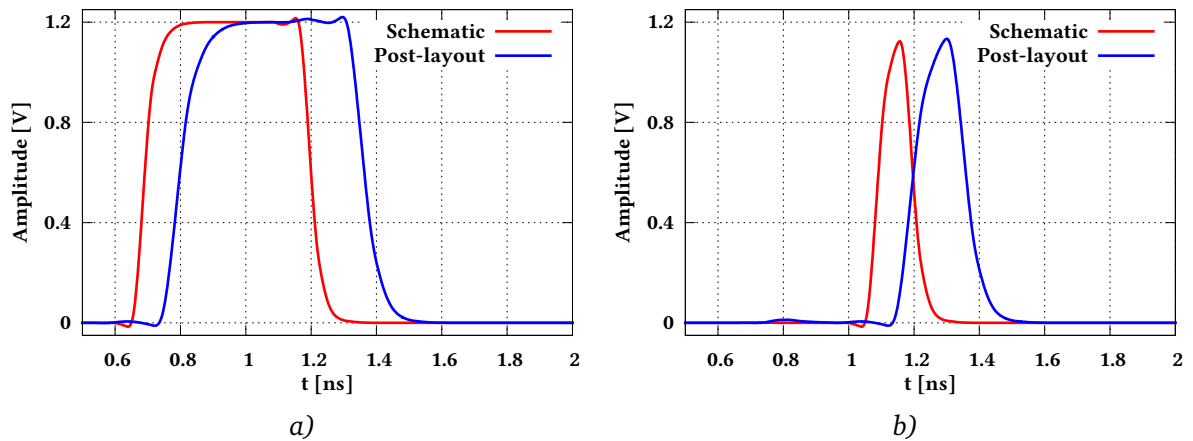


Figure 3.19: PFD output responses for schematic simulations and post-layout simulation. *a)* - output U (Up), *b)* - output D (Down).

Figure 3.19 shows the comparison between the results of schematic and post-layout simulations for the signals U and D . The simulations show that parasitic capacitances do not cause circuit malfunctioning, but the output pulses become wider for the same input phase difference. This is very important when the PLL is synchronized, because in that case both PFD output signals (U and D) look like the waveforms presented in figure 3.19b, whose pulses width are almost 2 times wider in the post-layout simulations.

An important parameter of the Phase and Frequency Detector is the gain which is presented in figure 3.20. The gain curve is obtained as the average of $U - D$ signals. When the phase difference between the Ref and Div signals is ranging from around $-\pi$ to π , the PFD gain curve is linear. In case when the phase difference between the PFD input signals is near π or $-\pi$, the gain value suddenly increases, which reduces the time needed for the PLL synchronization. As a result, when the phase difference between the Ref and Div is smaller than π and greater then 0, the output U produces pulses with the width proportional to the phase difference at the PFD input and the D stays

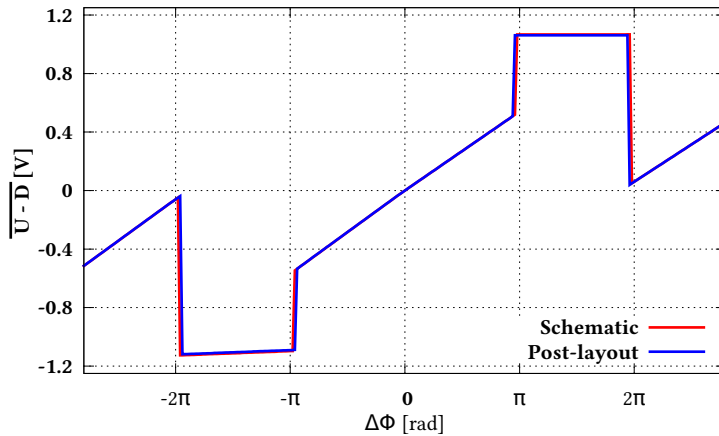


Figure 3.20: Phase and Frequency Detector gain curve.

in low state. When the phase difference is greater than half of a signal period ($\Delta\Phi > \pi$), the PFD output U becomes independent on the phase difference and stays in high state. The behavior of the output D is exactly the same, but this output stays in high state when the input phase difference is lesser then $-\pi$ and generates pulses when the phase difference is in range $-\pi - 0$.

3.1.3 Charge Pump (CP)

The Charge Pump (CP) converts digital signals from the PFD to current pulses which charge and discharge the Low-Pass Filter (LPF) capacitance. In this way the average value of the phase difference between the PFD outputs can be obtained. Figure 3.21 shows the CP schematic, which is exactly the same in the both MULTI_PLL versions. This schematic includes also the LPF components, because they are directly related with the CP. The circuit architecture is very simple, but it is different from a typical Charge Pump commonly presented in literature [79, 80].

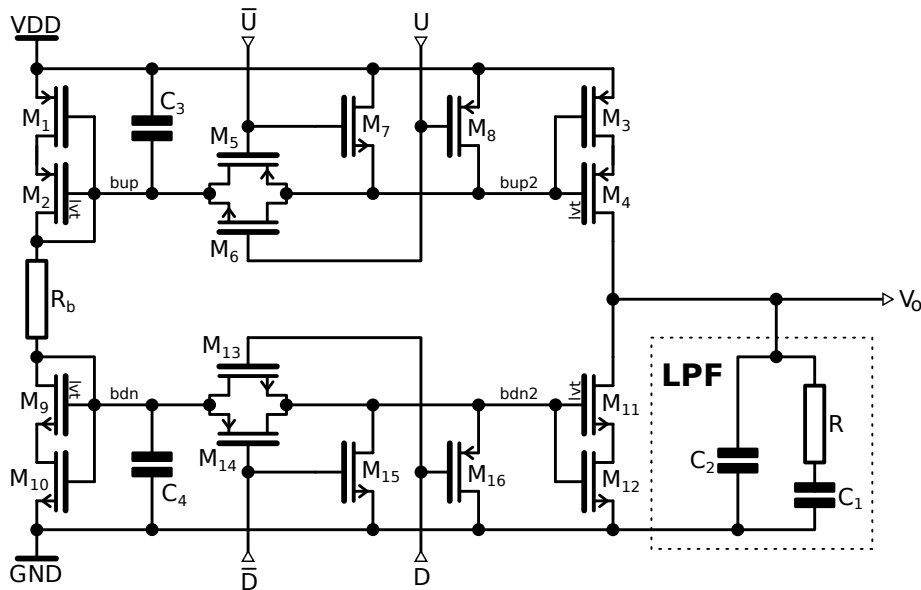


Figure 3.21: Charge Pump schematic.

The CP generally consists of two current mirrors and few switches to control the current flow. The transistors M_1 and M_2 work as a bias generator for the current source, which is built with transistors M_3 and M_4 . This current source is responsible for the LPF charging and it is controlled by the U signal, coming from the PFD. Transistors $M_9 - M_{12}$, which are controlled by the D signal, work in very similar way, but they are responsible for the LPF discharging. The bias voltage bup , generated by M_1 and M_2 , is controlled by resistance R_b , which sets the current of M_1 and M_2 transistors. The same current flows also through transistors M_9 and M_{10} , which generate the bias voltage bdn . The capacitors C_3 and C_4 are needed for decoupling the bup and bdn bias voltages. Two transistors in series $M_1 - M_2$, $M_3 - M_4$, etc. in the current mirror are used to increase their output resistance and to improve the operation of current source. Such source construction is called a low-voltage self cascode [81]. Transistors M_2 , M_4 are low-threshold devices because the M_4 should be in moderate inversion to improve the current noise characteristics.

The \bar{U} and \bar{D} signals are negations of U and D , but a simple inverters cannot be used for this, because it is very important to compensate/eliminate the delay between signals and their negations. More information about this compensation is presented in section 2.3.3. A practical approach is exactly the same as presented in figure 2.25b. When input U is in low state (\bar{U} is in high state) transistors M_7 and M_8 are on (means shorted) and transmission gate M_5 , M_6 is off (disconnected). As a result the signal $bup2$ is high and the upper current mirror ($M_1 - M_4$) is disabled. When the U becomes high, transistors M_7 , M_8 are off and transmission gate M_5 , M_6 is shorted. In this case the $bup2$ will be changed to the value of the bias voltage bup and transistors $M_1 - M_4$ start to work as current mirror. The NMOS transistors M_6 and M_7 are added only to ensure a symmetric load for the signals U and \bar{U} .

The second current mirror ($M_9 - M_{12}$) works in very similar way, but this part of the circuit depends on the signals D and \bar{D} . When the D signal becomes high transistors $M_9 - M_{12}$ start to work as current mirror. The PMOS transistors M_{14} and M_{16} are also dummy transistors, added to improve the symmetry of the circuit.

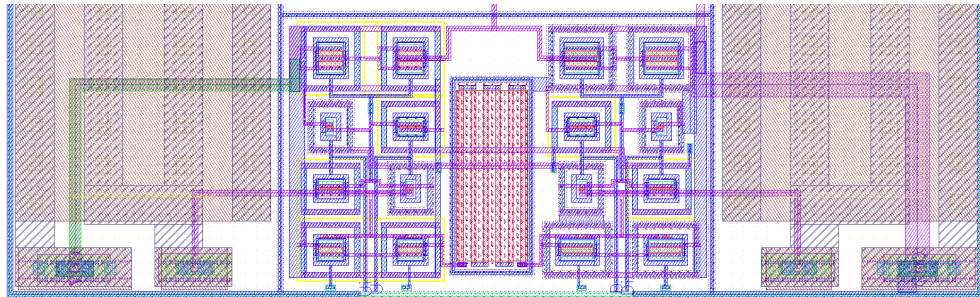


Figure 3.22: CP layout ($82 \times 27 \mu\text{m}^2$).

The CP layout is presented in figure 3.22, and occupies small area around $82 \times 27 \mu\text{m}^2$. The layout is manually drawn and bases on the mirror symmetry, which ensures a proper circuit operation. Figure 3.22 shows the resistor R_b (in center), the capacitors C_3 (on the left), C_4 (on the right) and all transistors. The LPF components are quite big and occupy area around $80 \times 82 \mu\text{m}^2$, so they are omitted.

Figure 3.23 shows the V_O output of the CP, which is used as a VCO control voltage. The plots presented in this figure come from the schematic and post-layout simulations. When the CP input signal U is slightly wider than signal D , the Charge Pump current mirror $M_1 - M_4$ gives higher average current than the $M_9 - M_{12}$, so the output voltage V_O rises (figure 3.23a). The V_O remains constant when the input signals U and D are equal, because the current, which flows via M_3 and M_4 , flows also via M_{11} and M_{12} (figure 3.23b).

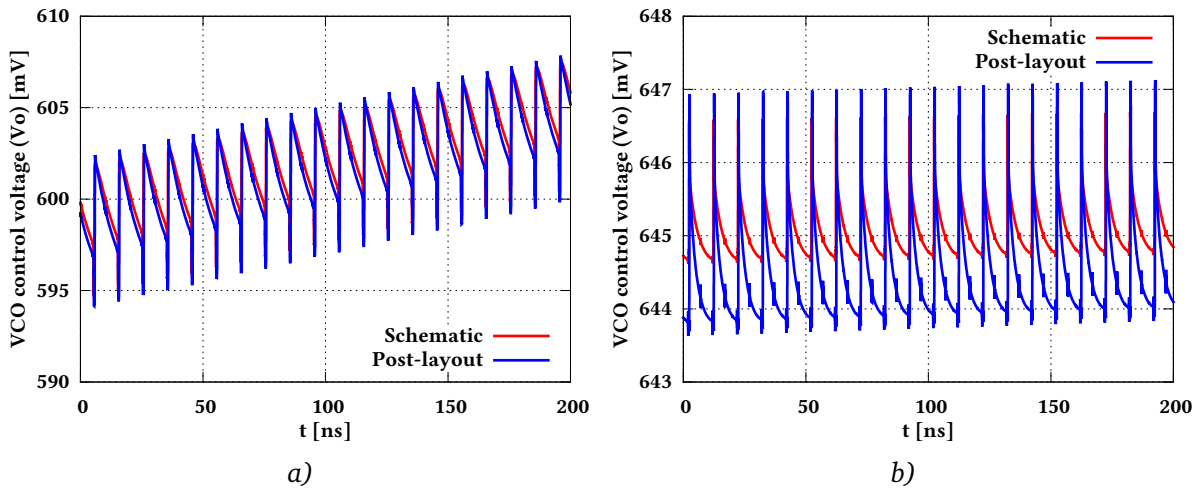


Figure 3.23: Charge Pump output voltage V_o . a) - U signal is slightly wider than D , V_o rises, b) - U and D signals are equal, V_o remains constant.

3.1.4 Frequency Divider

A frequency divider (clock divider) is a circuit which provides a frequency division by strictly defined integer value. When the divider input receives an input signal at frequency f_{in} , the divider output produces square waveform at frequency $f_{out} = f_{in}/n$, where n is division factor. Figure 3.24 shows a block diagram of the clock divider, which is exactly the same in both MULTI_PLL versions. The presented PLL divider can work with four different division factors (6, 8, 10 and 16), which can be changed via control signals $Ds[1:0]$ (2 bits), according to table 3.2.

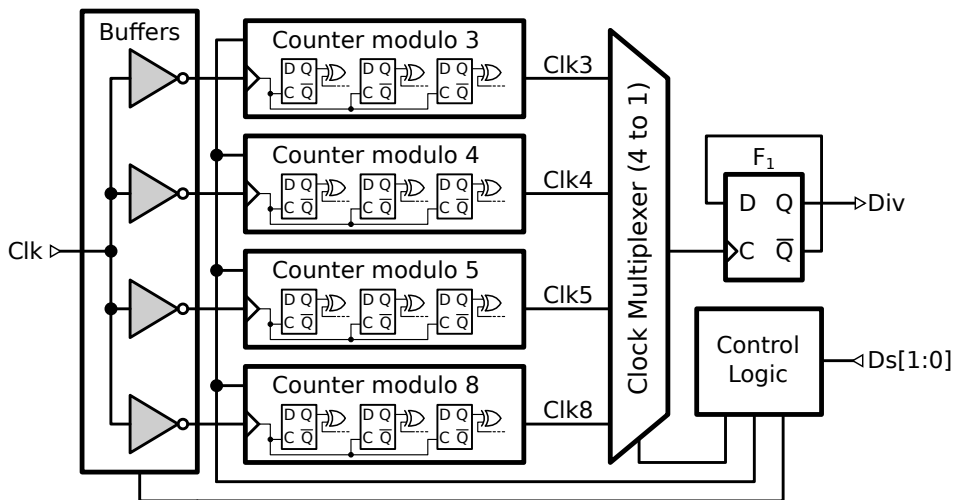


Figure 3.24: Block diagram of the MULTI_PLL divider.

The input clock Clk , which in normal conditions comes from the VCO output, is connected to the input buffers. The frequency division is splitted into two steps. The first of them is obtained by four synchronous modulo counters, which allow the division by 3, 4, 5 and 8. The second step is

$Ds[1 : 0]$	00	01	10	11
Division factor	6	8	10	16

Table 3.2: Divider configuration.

given by the flip-flop F_1 , which works as a standard asynchronous divider by 2. The divided signal is available at the output Div . The modulo counters are standard circuits built on flip-flops and XOR gates, which are commonly described in literature [82]. Only one of the counters and one of the buffers is enabled at a time, depending on the selected division factor. The control logic, based on $Ds[1 : 0]$, produces signals, which allow to enable/disable the modulo counters (and input buffers) and to select an appropriate multiplexer input ($Clk3$, $Clk4$, $Clk5$ or $Clk8$).

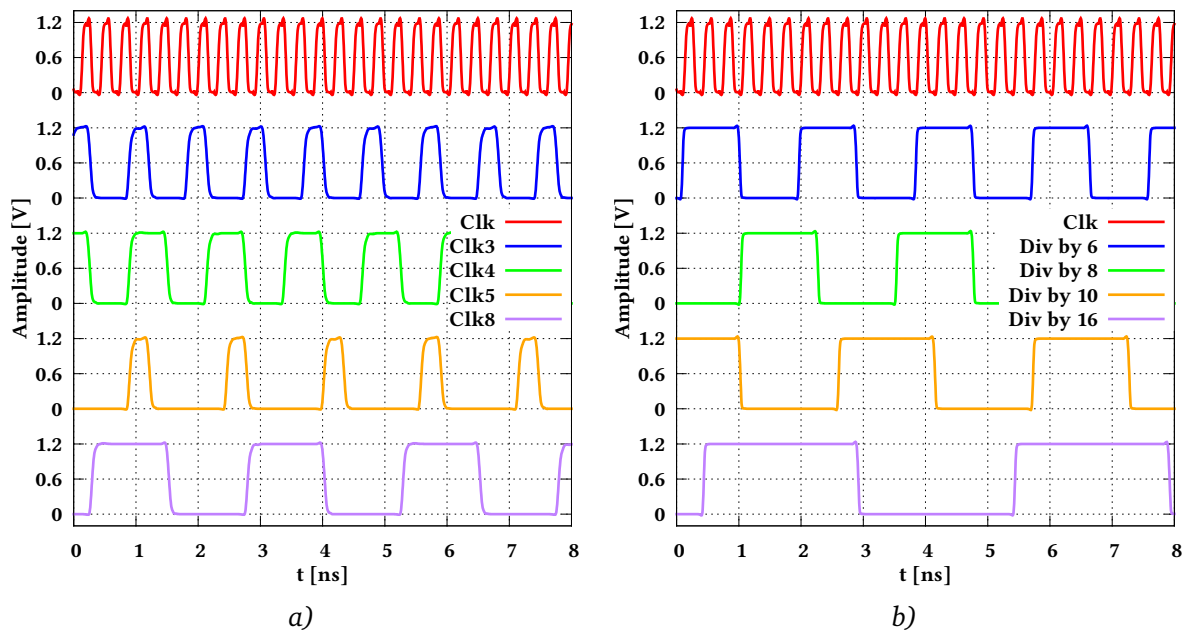


Figure 3.25: MULTI_PLL divider waveforms for all division factors, based on post-layout simulations. *a)* - modulo counters outputs, *b)* - divider outputs.

The two phases of clock division are implemented for two reasons. First of them is related to the synchronous counter, whose digital logic complexity rapidly rises with the rising number of bits. The complex logic gives large propagation times, so the maximum operation frequency of the counter is limited. The second reason is related to duty cycle of the signals, generated by modulo counters. When the cycle of the counter is not equal 2^n , the circuit produces waveforms with duty cycle not equal 50%, so an asynchronous divider is used to restore the proper duty cycle of the clock. Figure 3.25a presents the modulo counters outputs (after first division phase) based on the post-layout simulations. It is worth to notice that the signals $Clk4$ and $Clk8$ have a proper duty cycle equal 50%, while the two other clocks ($Clk3$ and $Clk5$) have different duty cycle values. Figure 3.25b shows the divider output Div for all possible circuit configurations. Thanks to the F_1 the divider gives the output signal with the duty cycle always equal 50%.

The post-layout simulations show that the presented frequency divider can operate up to 3.5 GHz. The power consumption at this frequency is around $600 \mu\text{W}$ and it scales linearly with frequency. The layout is manually drawn as for the other PLL blocks and occupies $50 \times 40 \mu\text{m}^2$.

3.1.5 Automatic Frequency Mode Setting (AFMS)

The AFMS is a digital circuit which provides an extra feedback of the Phase-Locked Loop by changing the VCO modes. The circuit can work with the MULTI_VCO or other oscillator, which can change the frequency ranges digitally. In general, the AFMS checks the voltage level of the VCO control input V_O by means of two comparators. When the V_O is greater than the selected high level bias voltage, for the defined time length, the VCO mode is switched to a higher frequency range. In the case the V_O is lower than the selected low level bias voltage, for the defined time length, the VCO mode is switched to a lower frequency range.

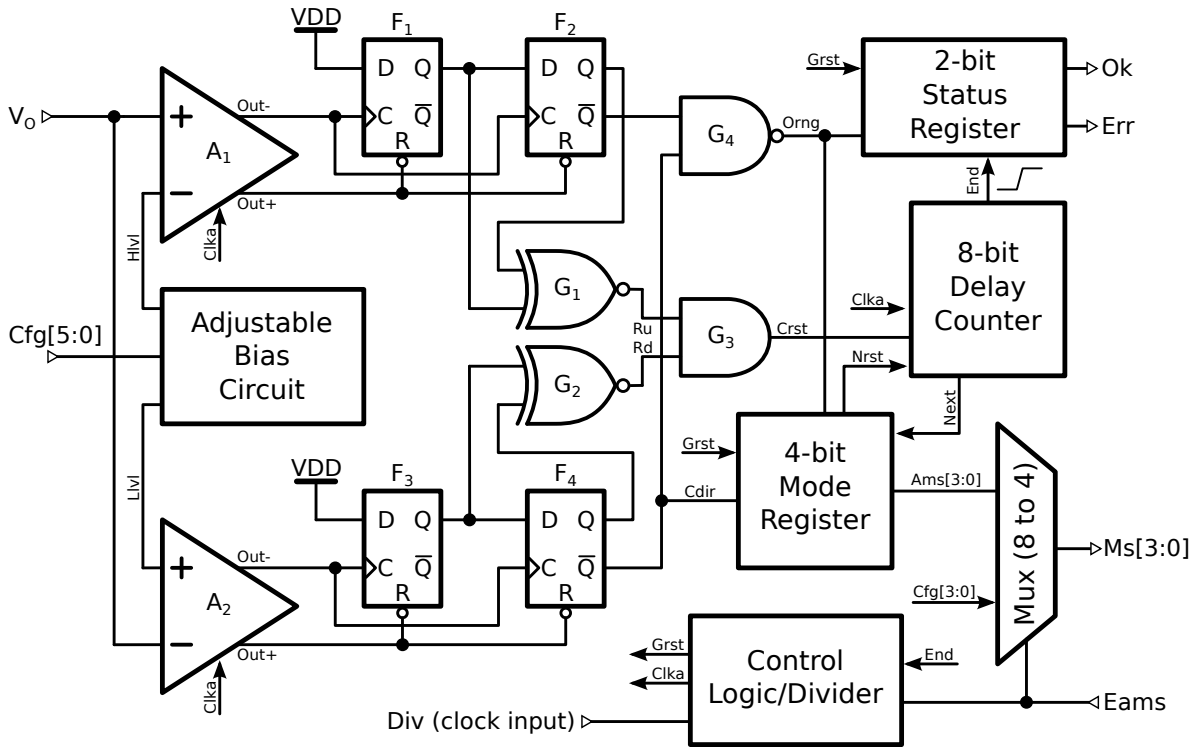


Figure 3.26: Simplified block diagram of the Automatic Frequency Mode Setting.

A simplified block diagram of the AFMS is presented in figure 3.26, which shows only the most important components of the circuit. The V_O goes to two dynamic comparators A_1 and A_2 [48], clocked by the signal $Clka$. When the V_O is greater than high level bias voltage $Hlvl$, the comparator A_1 generates rising edge at the output $Out-$ for each rising edge of the clock signal $Clka$, while the output $Out+$ is in high state. In opposite case, the comparator gives falling edge at the output $Out+$, while the output $Out-$ is in low state. The second comparator A_2 works in very similar way, but in this case the V_O is compared with the low level bias voltage $Llvl$. The bias voltages ($Hlvl$ and $Llvl$) are generated by an adjustable bias circuit, whose main component is resistive ladder, splitting the

$Cfg[2:0]/[5:3]$	000	001	010	011	100	101	110	111
$Hlvl$ voltage [mV]	1133	1066	1000	933	867	800	733	667
$Llvl$ voltage [mV]	67	134	200	267	334	400	467	533

Table 3.3: Adjustable bias circuit configuration.

power supply voltage. To choose the appropriate bias voltage level two analog multiplexers (8 to 1) are used. The configuration can be set by the digital input $Cfg[5 : 0]$ according to table 3.3. The three lowest bits ($Cfg[2 : 0]$) control the high level bias voltage $Hlvl$, while the $Cfg[5 : 3]$ control the low level $Llvl$.

When the V_O is between $Hlvl$ and $Llvl$, the comparators A_1 and A_2 reset the flip-flops $F_1 - F_4$ at each rising edge of the clock signal $Clka$. When the V_O is greater than $Hlvl$, the F_1 and F_2 are clocked, while the F_3 and F_4 are reset. When the V_O is lower than $Llvl$ the situation is opposite. The F_1 and F_2 (same as F_3 and F_4) work as 2-bit shift register, with input connected to VDD. The XNOR gate G_1 generates the falling edge at its output Ru whenever the V_O crosses the $Hlvl$. In the same way the signal Rd is generated by gate G_2 , when the V_O crosses the $Llvl$. Thanks to the gate G_3 the 8-bit delay counter is reset whenever the falling edge of the signals Ru or Rd arrives. The gate G_4 $Orng$ output gives information whether the V_O voltage is between $Hlvl$ and $Llvl$. When control voltage is outside the range, the $Orng$ is in high state.

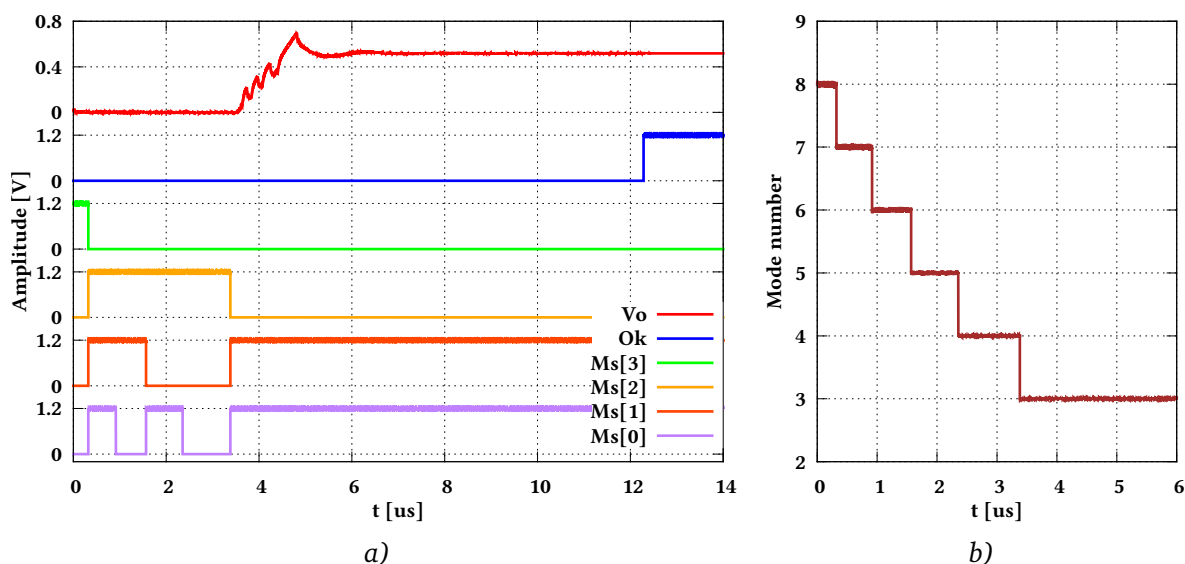


Figure 3.27: Automatic Frequency Mode Setting digital signals when the MULTI_PLL works at 300 MHz. *a)* - mode adjusting and PLL synchronization, *b)* - modes over time.

The delay counter works as time delay and it is very important, because the PLL needs some time to synchronize whenever the VCO mode is changed. When the AFMS is enabled, the delay counter is clocked by $Clka$ and counts clock periods. This block generates two signals End and $Next$, which represent the bit 7 and bit 3 of the counter. The $Next$ signal clocks the 4-bit mode register, which increases/decreases (depending on $Cdir$ signal) value of the register/signal $Ams[3 : 0]$ and resets the delay counter (by $Nrst$). The mode register can be changed only when $Orng$ is in high state, which means that the V_O is greater than $Hlvl$ or lower than $Llvl$. The $Cdir$ high state means that the V_O is greater than $Hlvl$ and probably needs to go higher. If this condition stays longer, the rising edge of the $Next$ signal increases value of the $Ams[3 : 0]$ register, which allows to obtain a higher frequency range by the VCO. This operation repeats until the proper mode is selected (the V_O stays between $Hlvl$ and $Llvl$) or the mode register takes one of the limit values (0000_2 , 1111_2). Figures 3.27 and 3.28 show two examples of the mode selection process (based on schematic simulations), the first one for the frequency 300 MHz and the second one for 3 GHz. If the mode cannot be changed, the delay counter is not reset and End signal will be generated. After that the 2-bit status register sets high state at Ok or Err output, which depends on the $Orng$ signal. When the proper mode is selected the Ok is set to high state, otherwise the Err becomes high.

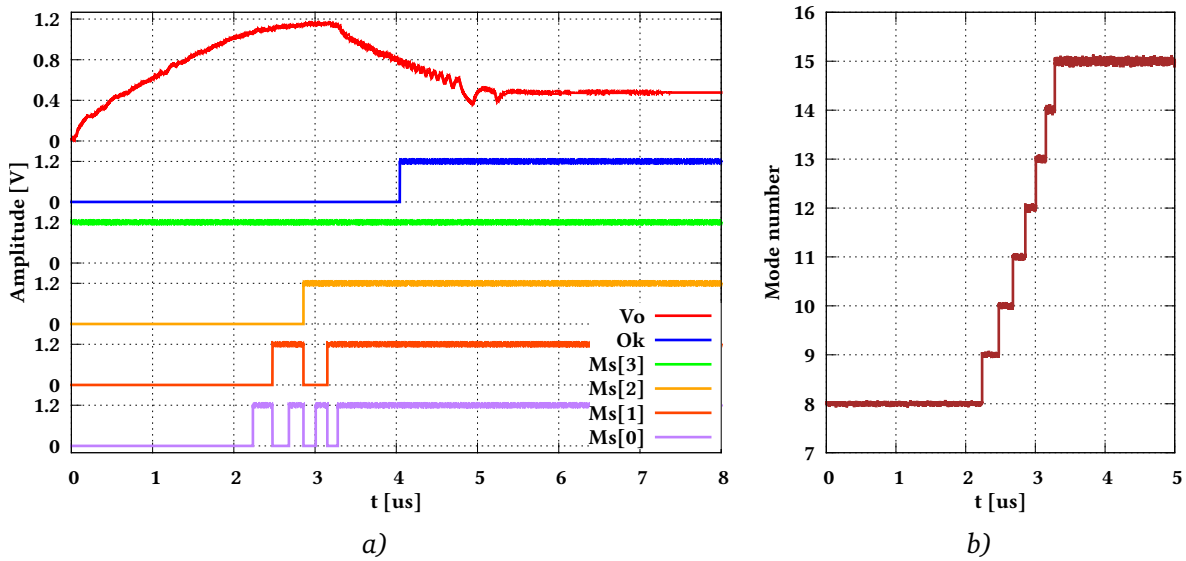


Figure 3.28: Automatic Frequency Mode Setting digital signals when the MULTI_PLL works at 3 GHz. *a)* - mode adjusting and PLL synchronization, *b)* - modes over time.

The Automatic Frequency Mode Setting block is clocked by the *Div* input, which comes from the PLL divider. The circuit is enabled when the *Eams* signal is in high state. In such case the multiplexer (8 to 4) connects the *Ams*[3 : 0] directly to the output *Ms*[3 : 0], which controls the VCO mode. The *End* signal generated by the delay counter is used for the automatic power off the AFMS, when the mode selection procedure is finished. When the control logic/divider detects high state on the *End*, it disables all digital parts of the AFMS by blocking the internal clock *Clka*. The mode register and status register keep previously obtained logic states, which still ensures the PLL operation in a proper mode. When the *Eams* is in low state, the AFMS circuit is disabled. The multiplexer connects the *Cfg*[3 : 0] (4 the lowest bits of the *Cfg* input) directly to the output *Ms*[3 : 0]. The global reset *Grst* signal ensures that the status register is reset to low state and the mode register is restored to value 1000_2 (mode 8). When the *Eams* goes to high state next time the mode selection process starts from the beginning, which is usually needed when the reference frequency of the PLL is changed.

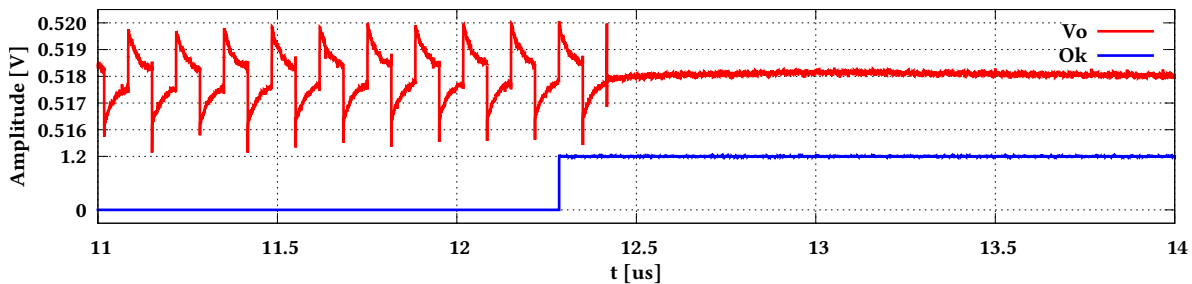


Figure 3.29: MULTI_PLL V_O voltage disturbances before the AFMS disabling.

The AFMS samples the V_O by two dynamic comparators A_1 and A_2 , which add fluctuations to this voltage (figure 3.29). This effect, called kickback noise [83, 84], is especially important when the PLL is synchronized. The V_O should be constant in such condition, but comparators introduce disturbances when they operate. The figure shows how the V_O control voltage looks before and after the AFMS stops working.

The post-layout simulations show that the circuit consumes $80 \mu\text{W}$ at frequency 3 GHz and $20 \mu\text{W}$ at frequency 300 MHz, when it is enabled. When the AFMS is disabled or the changing mode procedure is finished, the power consumption is reduced to a value below $1 \mu\text{W}$. The layout occupies area around $130 \times 65 \mu\text{m}^2$.

3.2 Design and simulations of SALT_PLL

The SALT_PLL was designed for the SALT readout chip as a dedicated multi-phase Phase-Locked Loop. It is used as clock multiplier and phase shifter for the serializer and deserializer blocks. The PLL was designed in two versions (1st and 2nd), in two different technologies, respectively: 130 nm CMOS technology A and 130 nm CMOS technology B. The circuit (both versions) operates in frequency range, from around 70 MHz up to around 350 MHz, which is the default requested frequency range. Figure 3.30 shows the block diagram of the SALT_PLL. The architecture of the PLL is similar to typical design, commonly described in literature [85, 86, 87, 88], but few improvements were implemented. In the 2nd circuit version extra DAC circuits were added for automatic control of bias currents. The Voltage-Controlled Oscillator is a main component of the PLL and can generate 16

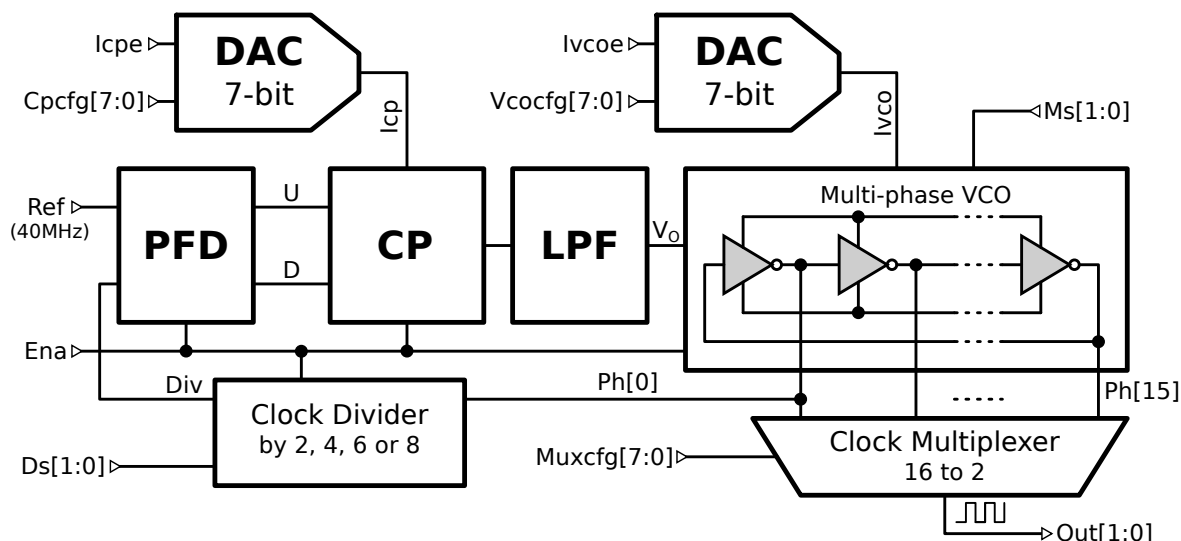


Figure 3.30: SALT_PLL block diagram.

clock phases $Ph[0] - Ph[15]$. The VCO provides gain/mode selection (1 of 4), which can be useful for the jitter optimization (if necessary) and the selection of working frequency. More details about the two VCO versions and their configurations can be found in subsection 3.2.1.

The reference input signal Ref (default 40 MHz) is compared with the divided VCO frequency Div by the PFD. The digital signals U and D control the CP, which allows to change the VCO control voltage V_0 . The detailed description of PFD and CP is given in subsections 3.2.2 and 3.2.3. The SALT_PLL works with four division factors (2, 4, 6 and 8) obtained with clock divider. The default value is 4, which allows to generate the default output frequency 160 MHz from the reference 40 MHz. The divider construction is very similar to MULTI_PLL divider, described in subsection 3.1.4, but the division factors are equal to 2, 4, 6 and 8, instead of 6, 8, 10, 16. The clock divider can be controlled by $Ds[1:0]$ input. The PLL generates 16 clock phases, but only two of them are available at the outputs $Out[1:0]$. Two internal multiplexers provide this clock phase selection (2 from 16 phases). The architecture of the multiplexer is typical, widely described in literature [56, 82], so its description is not given here.

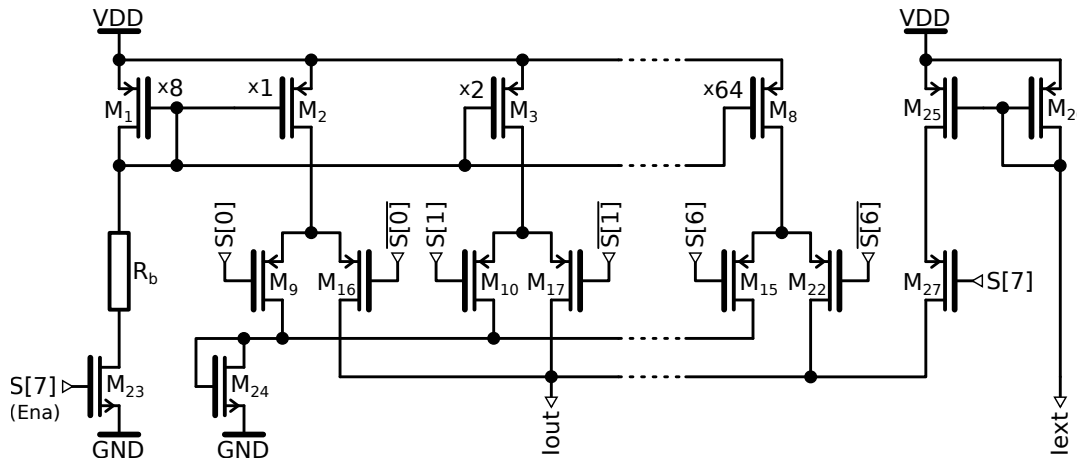


Figure 3.31: Schematic of the SALT_PLL_V2 (SALT_DLL_V2) DAC.

In the 1st PLL version the VCO and CP reference currents (I_{cp} and I_{vco}) are controlled from outside via input pads, while in the 2nd circuit version the same currents can be changed automatically by internal 7-bit DACs, which allow to avoid bias current fluctuations, introduced from outside. The bias current DACs are controlled by signals $Cpcf[7:0]$ and $Vcocf[7:0]$. When the DACs are disabled the bias current can be controlled in the same way as for the 1st circuit version, using the inputs I_{cpe} and I_{vcoe} .

The schematic of the bias current DAC is presented in figure 3.31. The decoupling capacitors and control signals inverters are omitted for simpler presentation. The DAC architecture is typical and commonly used in literature [43]. The DAC is controlled by 8-bit input $S[7:0]$, in which the MSB ($S[7]$) works as DAC enable (Ena). When the Ena is in high state, M_{23} is shorted and current flows through the R_b , setting the DAC reference current (8 times higher than DAC LSB current). The output current I_{out} is generated by binary scalable current mirrors $M_2 - M_8$ (only two first bits and MSB are shown in figure 3.31). The $S[6:0]$ input serves to set the value of I_{out} current, by turning

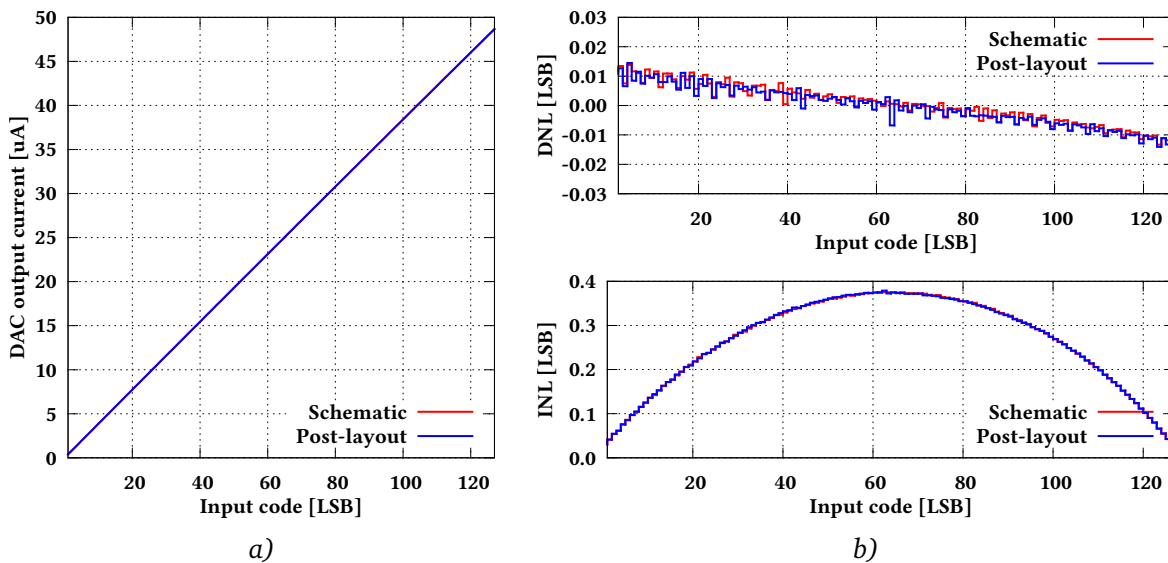


Figure 3.32: Performance of the DAC used in the SALT_PLL_V2. a) - transfer curve, b) - Differential Nonlinearity (DNL) and Integral Nonlinearity (INL).

on/off switches $M_9 - M_{22}$. The current consumed by the DAC is constant, thanks to the dummy switches $M_9 - M_{15}$, which create current path to ground (via transistor M_{24}) and keep the proper polarization of sources $M_1 - M_8$, independently of output current. When Ena ($S[7]$) is low, the DAC is disabled and the I_{out} is equal to the I_{ext} , thanks to the current mirror $M_{25} - M_{26}$. The M_{27} is controlled by the same signal as M_{23} , so only one of these switches is shorted at a time.

The transfer curve of the current bias DAC is presented in figure 3.32a. The schematic and post-layout simulations give almost the same results, which was expected, because the presented DAC is a static circuit and parasitic components do not degrade its parameters. The DAC allows to set the bias current linearly in range $0 - 48 \mu A$. The LSB is very small, around 380 nA , what makes the circuit quite universal general purpose DAC, not only for PLLs and DLLs but also for other applications. The main parameters of the circuit are presented in figure 3.32b. The Differential Nonlinearity (DNL) is very small, better than $\pm 0.02 \text{ LSB}$ in the whole range. The Integral Nonlinearity (INL), calculated by endpoint method, is also very good and its maximum value is smaller than 0.4 LSB .

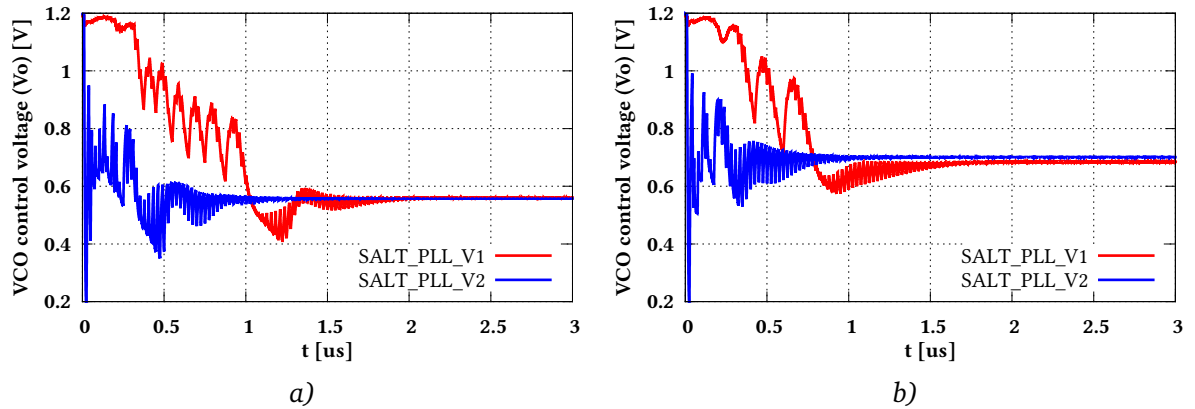


Figure 3.33: Synchronization process comparison of two SALT_PLL versions. *a)* - schematic simulations, *b)* - post-layout simulations.

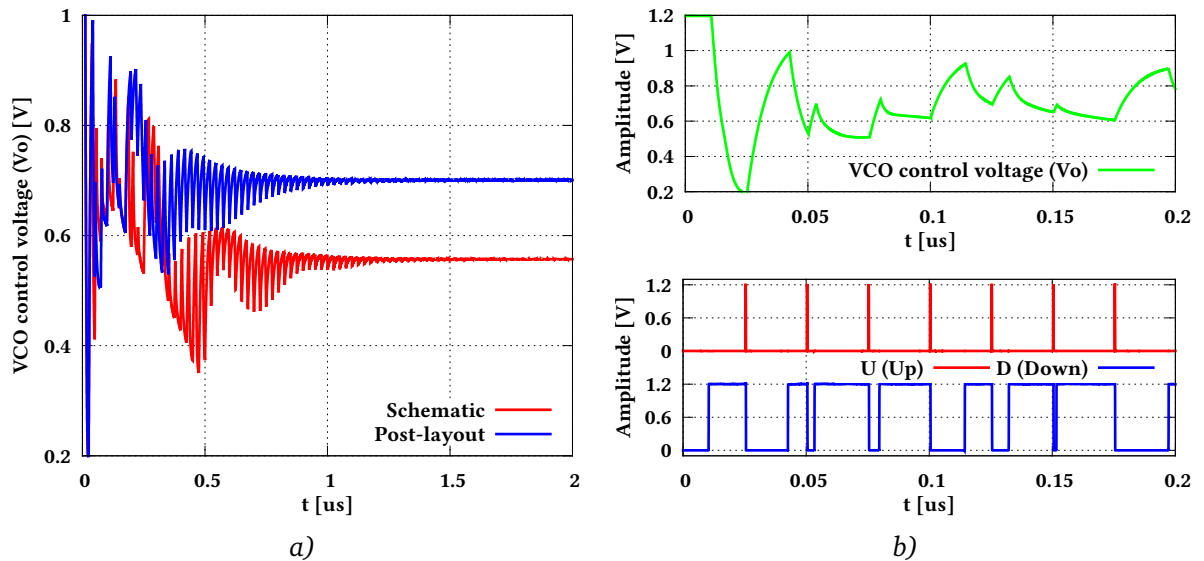


Figure 3.34: SALT_PLL_V2 synchronization process. *a)* - comparison of the schematic and post-layout simulations, *b)* - the VCO control voltage (V_O) with the corresponding signals (U and D).

The comparison of PLL synchronization presented in figure 3.33 was simulated at typical conditions. In all presented cases (and versions) the SALT_PLL works at output frequency 160 MHz, with reference frequency 40 MHz (the loop divider was set to 4). The synchronization time of the 2nd PLL version is shorter than for the 1st circuit version, but in general the behavior of two circuits, designed in two different technologies, looks very similar.

The synchronization process of SALT_PLL_V2 is presented in figure 3.34a. The comparison between schematic and post-layout simulation shows that the circuit works well in both cases. The VCO control voltage (V_O) is moved into higher values when parasitic components are present, because the lower VCO output frequency is compensated by the higher values of V_O voltage (negative feedback). Figure 3.34b shows the VCO control voltage at the beginning of the synchronization process, with the corresponding U and D signals, of which not regular nature is responsible for sudden V_O changes.

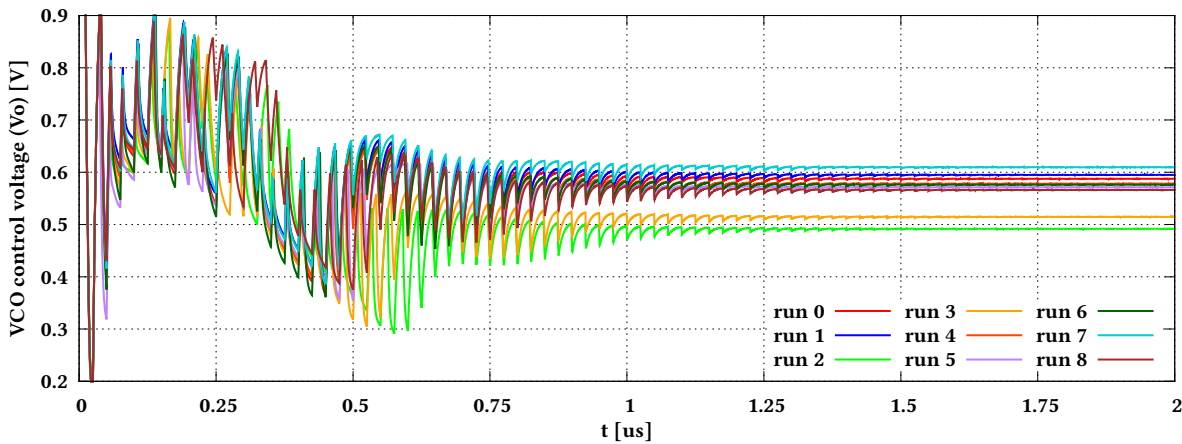


Figure 3.35: SALT_PLL_V2 MC analysis of the synchronization process - schematic simulations.

Figure 3.35 presents the MC analysis of the SALT_PLL_V2 synchronization, which show how the circuit behaves when the component dimensions (especially transistors) are changed randomly. The figure presents simulation results for 9 different (randomly selected) MC seeds and shows that the circuit works well in all these cases. The V_O voltage at the end of PLL synchronization process is slightly different for each of the simulation runs, what is related to the randomly changing VCO output frequency.

3.2.1 Voltage-Controlled Oscillator (VCO)

The proposed Voltage-Controlled Oscillator, called SALT_VCO, is much simpler than the MULTI_VCO and operates in 4 different frequency modes/ranges. The block diagram, presented in figure 3.36, is similar in both circuit versions. The switched current bias block in the 1st version, controlled by $Ms[1:0]$, allows to change the VCO gain and reference current (I_{vco}) multiplication factor, which is related to the gain. In the second VCO version the switched current bias block controls only the VCO gain, while the multiplication factor of I_{vco} is equal 1. The I_{vco} current in SALT_VCO_V1 is controlled from outside and should be constant in typical application, so the four VCO modes need the same number of bias current multiplication factors to obtain different frequency range for each of the modes. The reference current in SALT_VCO_V2 is controlled by 7-bit DAC, and so another variable current multiplication is not needed. The frequency ranges, dependent on the selected mode ($Ms[1:0]$ input), are summarized in table 3.4. All these simulations were done at default reference current I_{vco} equal 10 μ A. The modes in SALT_VCO_V2 change only the oscillator gain so

for a constant reference current the minimum oscillator frequency is the same for each of the modes. The 1st circuit version achieves gain in range 80 MHz/V – 90 MHz/V (post-layout), while for the SALT_VCO_V2 the gain ranges from 133 MHz/V to 255 MHz/V.

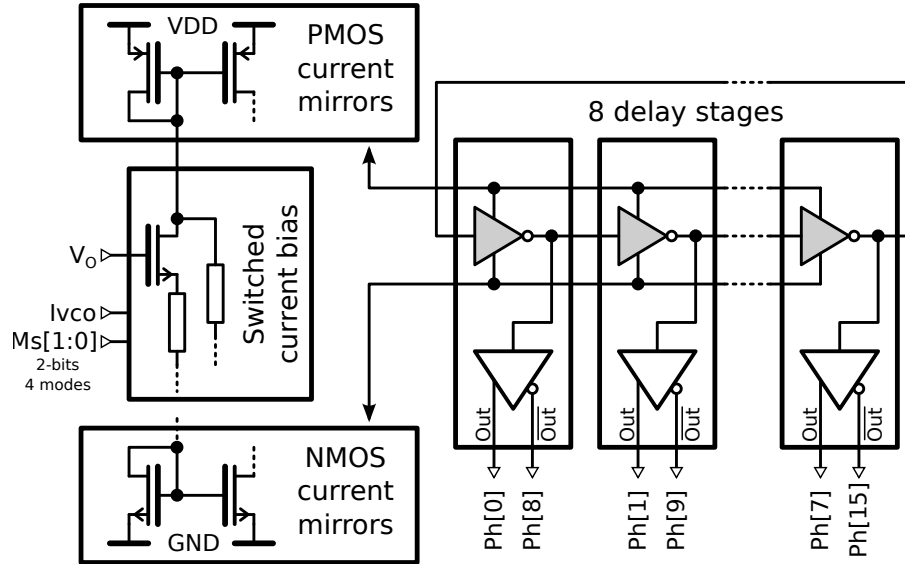


Figure 3.36: Block diagram of the both SALT_VCO versions.

VCO mode	Output frequency ranges of the VCO ($f_{min} - f_{max}$) [MHz]			
	Schematic simulations		Post-layout simulations	
	SALT_VCO_V1	SALT_VCO_V2	SALT_VCO_V1	SALT_VCO_V2
0	53.3 – 154	84.2 – 254	48.4 – 138	65.6 – 199
1	143 – 236	84.2 – 314	129 – 212	65.6 – 247
2	226 – 319	84.2 – 368	202 – 285	65.6 – 290
3	288 – 391	84.2 – 408	257 – 349	65.6 – 321

Table 3.4: Simulated frequency ranges for all modes for two VCO versions (1st and 2nd).

The PMOS/NMOS current mirrors ensure complementary currents for the biasing of the delay stages. The both circuit versions consist of 8 delay stages and allow to generate 16 clock phases. The stage construction is different in each of the circuit versions. The SALT_VCO_V1 delay stages are based on single-ended inverters, while the 2nd circuit version consists of differential gates.

The schematic of the 1st version delay stage is shown in figure 3.37. Transistors $M_1 - M_4$ work as Current-Starved Inverters, of which current is controlled by nodes Su and Sd . These transistors are the most important components in the circuit, because they are responsible for the propagation time of the delay stage. The Rin and $Rout$ are used to connect delay stages in series. The $Rout$ output is connected to Rin input of the next stage, except the last stage, whose $Rout$ output (instead of $Rout$) is connected to Rin of the first stage. Its makes that the output signal (Out), generated by the last stage is slightly different than signals created by other stages. However, it is the only way to ensure the odd number of inverters, which is needed to create an oscillator.

Transistors $M_5 - M_8$, whose current is controlled by nodes $Su2$ and $Sd2$, buffer the signals from the delay stage ($M_1 - M_4$), working in the oscillator ring. The limitation of current in inverters

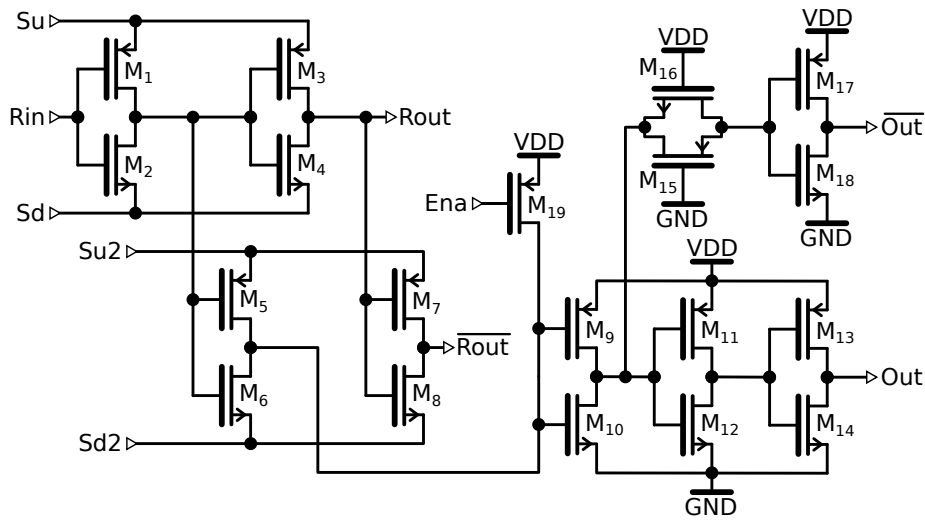


Figure 3.37: Schematic of the delay stage used in the SALT_VCO_V1.

$M_5 - M_8$ is not necessary for the proper circuit operation, but it helps in the reduction of voltage fluctuations at the gates of $M_1 - M_4$. As a result the PSRR is reduced and the VCO jitter performance can be improved.

The transistors $M_9 - M_{18}$ are responsible for the output signal (*Out*) buffering and create also a complementary output signal \overline{Out} . The transmission gate, built on M_{15} and M_{16} , is always shorted and introduce a time delay equal to the propagation time of inverter M_{11} , M_{12} . When the VCO is disabled (*Ena* is low), the PMOS/NMOS current mirrors (figure 3.36) are also disabled, so the signals *Su*, *Sd*, *Su2* and *Sd2* are floating. This makes that the state at the gates of M_9 and M_{10} is not well defined, which may leads to direct-path current flow in the circuit. To avoid this disadvantage transistor M_{19} is added. When the *Ena* is low the M_{19} shorts the M_9 and M_{10} gates to VDD and output *Out* goes to ground.

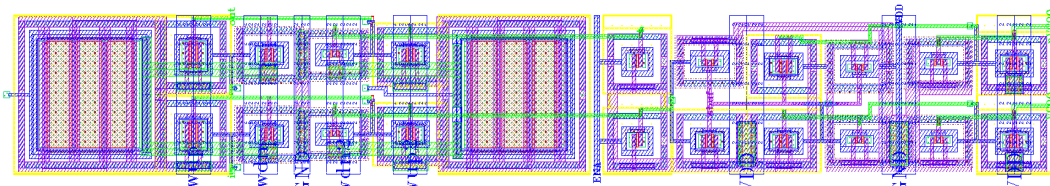


Figure 3.38: Delay stage layout of 1st circuit version ($76 \times 12 \mu\text{m}^2$).

Figure 3.38 shows the layout of the 1st version delay stage. The transistors $M_1 - M_8$ are placed on the left, between two small capacitors (omitted in figure 3.37). The right side of the layout is occupied by transistors $M_9 - M_{19}$.

Figure 3.39 presents the schematic of the 2nd version delay stage, which is based on differential inverter [89]. The differential input *Rin*, \overline{Rin} and differential output *Rout*, \overline{Rout} are used to connect the delay stages in series. The *Rout* output in one stage is connected to the *Rin* input in the next stage, and the \overline{Rout} is connected to the \overline{Rin} . The last delay stage of SALT_VCO_V2 is connected to the first stage in different way. The *Rout* is replaced by \overline{Rout} and vice versa, but this time the circuit keeps the symmetry, because the \overline{Rout} is complementary to the *Rout*.

The circuit can be splitted in two identical parts, which generate complementary output signals *Out*, \overline{Out} . The components in the second part are marked by additional letter "a". The transistors $M_1 - M_3$ and $M_{1a} - M_{3a}$ are the most important components in the circuit, because they work as the

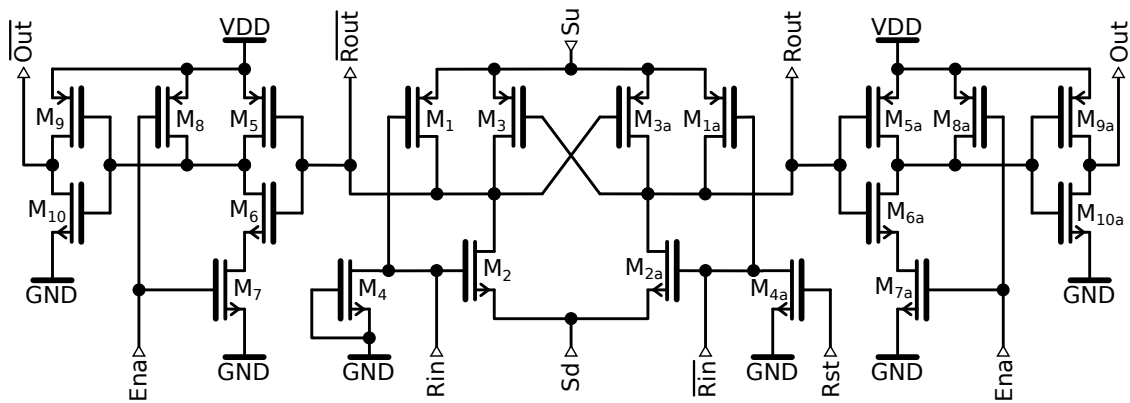


Figure 3.39: Schematic of the delay stage used in the SALT_VCO_V2.

differential inverter. The Su and Sd nodes allow to connect PMOS/NMOS current sources, which make that the stage can operate as CSI.

The input signal Rin goes directly to transistors M_1 and M_2 , which operate as a standard inverter (M_{1a} and M_{2a} behave similarly). The transistors M_3 and M_{3a} give a positive feedback to the circuit and ensure that $Rout$ and \overline{Rout} are in opposite phase.

The transistors $M_5 - M_{10}$ work as output buffer, which restores the limited \overline{Rout} signal swing to the full swing (output \overline{Out}). The transistors M_7 and M_8 , controlled by input signal Ena , allow to disable the output (\overline{Out}) in case the SALT_VCO_V2 is disabled (Ena is low). The transistors $M_{5a} - M_{10a}$ work in the same way, restoring the swing on the output Out . The transistor M_{4a} , controlled by Rst input, allows to reset the VCO stage. When Rst is in high state, the M_{4a} shorts Rin to ground, which ensure proper initial condition of the delay stage. The reset signal is connected only to the first delay stage, while for other stages the Rst is always tied to ground. The M_4 is a dummy transistor (always off) and it is added for circuit symmetry.

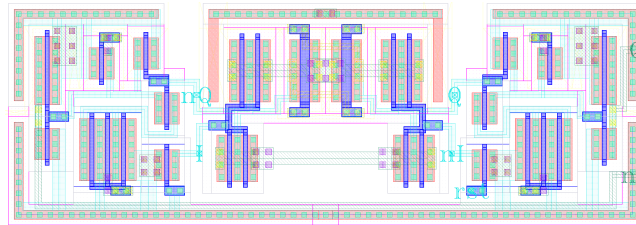


Figure 3.40: Delay stage layout of 2nd circuit version ($22 \times 7.3 \mu\text{m}^2$).

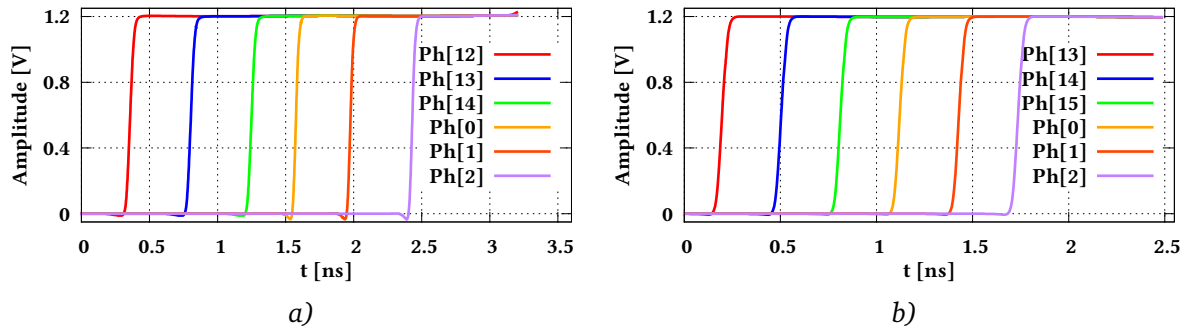


Figure 3.41: SALT_VCO output signals. a) - 1st circuit version, b) - 2nd circuit version.

The layout of the 2nd version delay stage, shown in figure 3.40, is drawn in mirror symmetry. The layout area is equal $22 \times 7.3 \mu\text{m}^2$, which gives 2 times smaller value then the area occupied by the 1st version delay stage.

The both versions of the SALT_VCO generate a multiple clock phases, as presented in figure 3.41. The 1st circuit version has few disadvantages, but the most important is the mismatch between the output clock phases. The 2nd VCO version is based on differential architecture and the oscillations can be generated without introducing asymmetry to the circuit. The time delay between two consecutive clock phases is exactly the same (even the delay between $Ph[15]$ and $Ph[0]$). It makes that the circuit can be used not only as a simple clock phase shifter, but also as a general purpose block, generating clocks for data serializers.

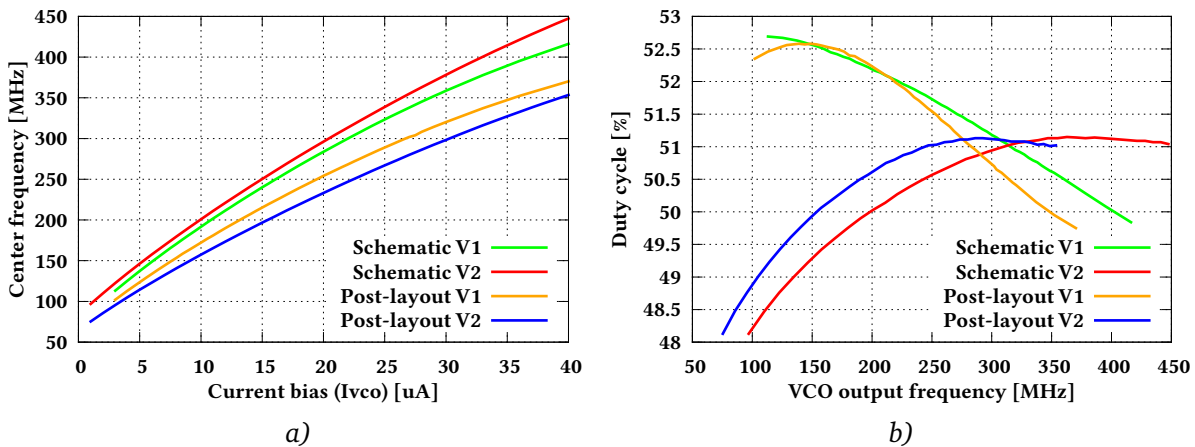


Figure 3.42: SALT_VCO output parameters. a) - output frequency versus current bias, b) - duty cycle versus oscillation frequency.

Figure 3.42a presents the VCO center frequency (for $V_O = 600 \text{ mV}$) versus current bias. The output frequency is in good approximation a linear function of bias current in the range $3 \mu\text{A} - 40 \mu\text{A}$. The typical current $10 \mu\text{A}$ allows to operate at default frequency 160 MHz . The duty cycle of both VCO versions is around $50\% \pm 2.5\%$ and is slightly related to the output frequency, as presented in figure 3.42b. The power consumption of the VCO circuit in both versions is very similar and scales

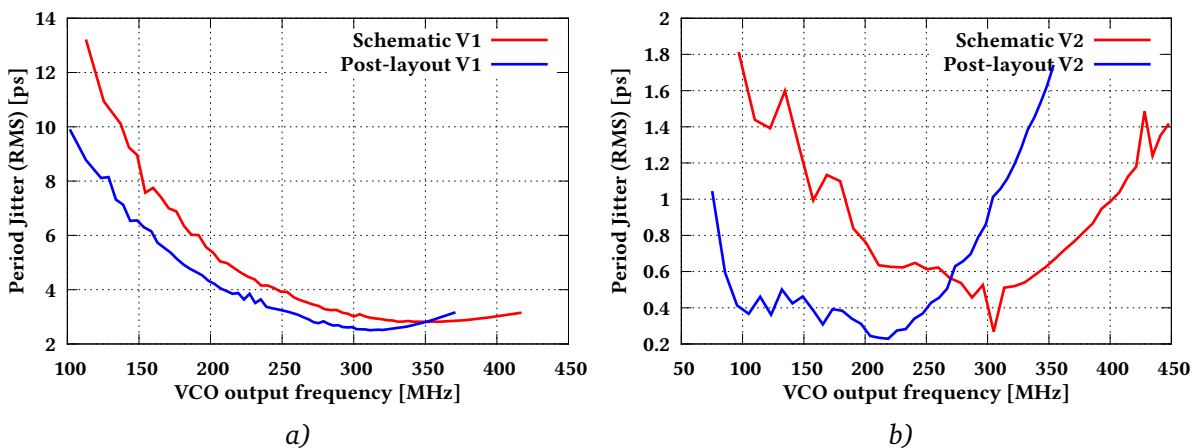


Figure 3.43: SALT_VCO period jitter versus oscillation frequency. a) - 1st version, b) - 2nd version.

linearly with output frequency. The SALT_VCO_V2 consumes around $300 \mu\text{W}$ at typical frequency 160 MHz, while the 1st circuit version consumes around $350 \mu\text{W}$.

The period jitter is the most important parameter of the circuit and it is presented in figure 3.43. The 2nd VCO version has much better jitter performance than the 1st circuit version and achieves values below 1.5 ps in frequency range 80 MHz – 320 MHz (post-layout simulations). The SALT_VCO_V1 achieves jitter values below 14 ps, decreasing for higher output frequencies below 4 ps when the output frequency is higher than 250 MHz. In both cases the post-layout simulations give better jitter performance than the schematic simulations. The parasitic capacitances not only reduce maximum output frequency obtained by the VCO, but also filter voltage fluctuations and improve the PSRR of the circuit, which is directly related to the period jitter.

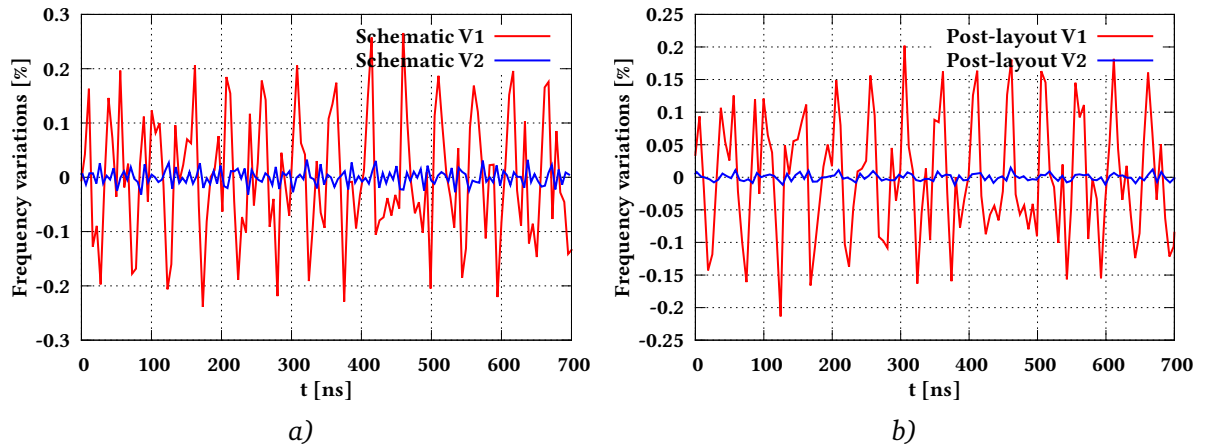


Figure 3.44: SALT_VCO frequency stability for two circuit versions at frequency 160 MHz.
a) - schematic simulations, b) - post-layout simulations.

Figure 3.44 shows the stability of the output signal frequency at 160 MHz. This stability is directly related to the clock jitter. The frequency variations in the 2nd circuit version are around 10 times smaller than the frequency fluctuations in the 1st circuit version, which gives similar relation to the VCO jitter performance.

3.2.2 Phase and Frequency Detector (PFD)

The Phase and Frequency Detector works in both versions of SALT_PLL and SALT_DLL and is based on typical construction with two flip-flops, widely described in section 2.3.2. The simplified PFD schematic is presented in figure 3.45. The flip-flop working in the circuit is based on dynamic architecture [89] and consist of two transmission gates T_1 , T_2 and two inverters I_2 and I_3 . The transmission gates are shorted when their control input S is in high state.

The PFD schematic shows only a half of the circuit, which is responsible for processing of the U signal. The second part of PFD is very similar and there is only one difference - the transmission gate T_1 is connected directly to VDD instead of output Q of F_1 . The F_1 is very important when PFD operates like PD in the DLL circuit. The F_1 blocks the first rising edge on the $Inup$ input, what is necessary for proper DLL synchronization.

The PFD is disabled when Ena or Rst signal is low. The gate G_1 sets the low state at the Ren net, which blocks the AND gate G_2 and resets the flip-flop F_1 . When the PFD operates normally, the G_2 and inverter I_1 generate signals Cu and \overline{Cu} , which give clock to: F_1 , transmission gates T_1 , T_2 , and transistors M_1 , M_6 . When Cu is in low state, the T_1 is shorted and inverter I_2 has high state on its

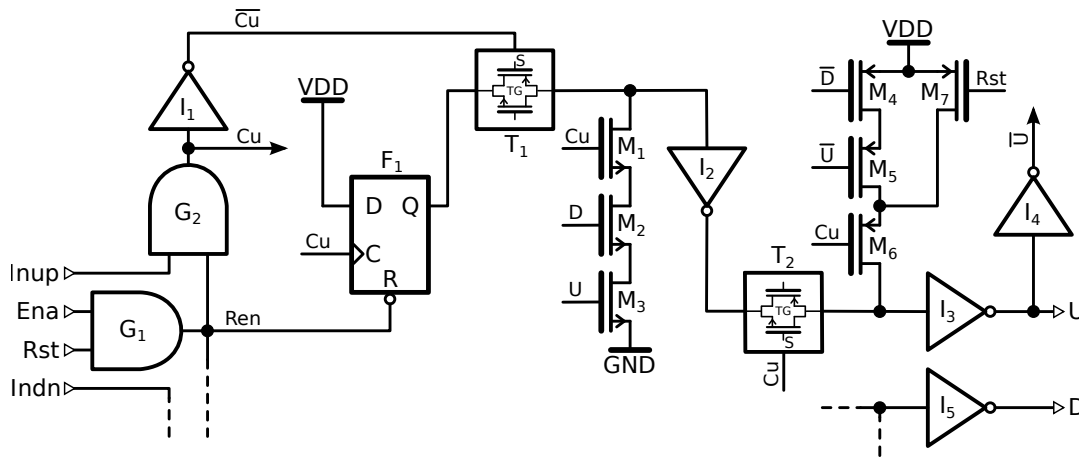


Figure 3.45: Simplified schematic of Phase and Frequency Detector.

input. The T_2 is disconnected and its input is tied down by I_2 . The rising edge on $Inup$, gives rising edge on Cu , which turns off T_1 and turns on T_2 . The inverter I_3 output goes to high state, generating the output signal U . The inverter I_4 is added to create negation of signal U , needed to control PMOS transistors.

When the U is in high state transistors M_3 and M_5 are on. This condition persists until the D output, coming from the complementary circuit part, goes in high state. After that transistors $M_1 - M_6$ make that the output U goes to ground, same as output D in the complementary PFD part. When the Rst is low, transistor M_7 resets the output U (and output D) independently on input signals.

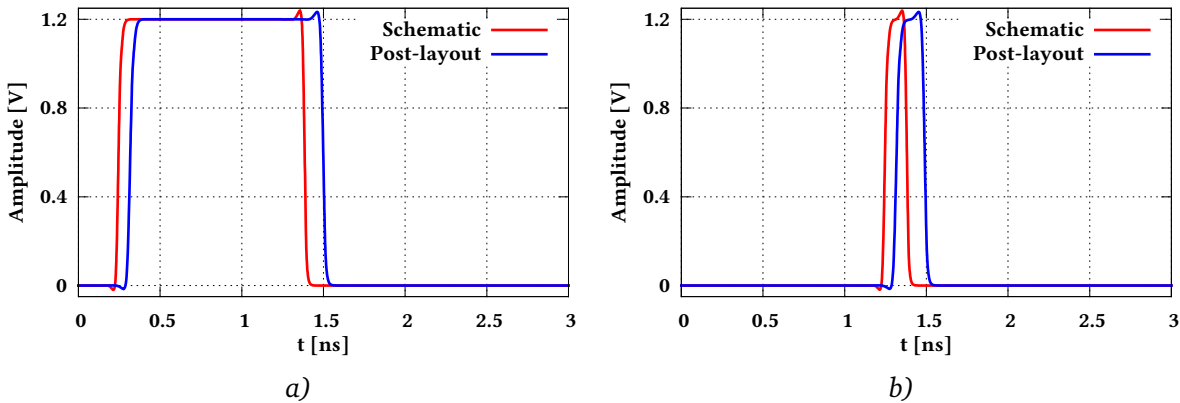


Figure 3.46: PFD output responses for schematic simulations and post-layout simulations. a) - output U (Up), b) - output D (Down).

The example PFD output waveforms (2nd version) are presented in figure 3.46. In this case the $Inup$ signal comes before the $Indn$, so the U is generated as phase difference between them, and the D is generated as short pulse. The plot shows comparison between schematic and post-layout simulations. The pulses are wider in post-layout simulations, which is easier to observe in figure 3.46b.

Figure 3.47 presents the PFD gain comparison between the schematic and post-layout simulations, and shows also the gain curves for both circuit versions. The gain curves in all cases are almost the same, independently on parasitic capacitances and technological process.

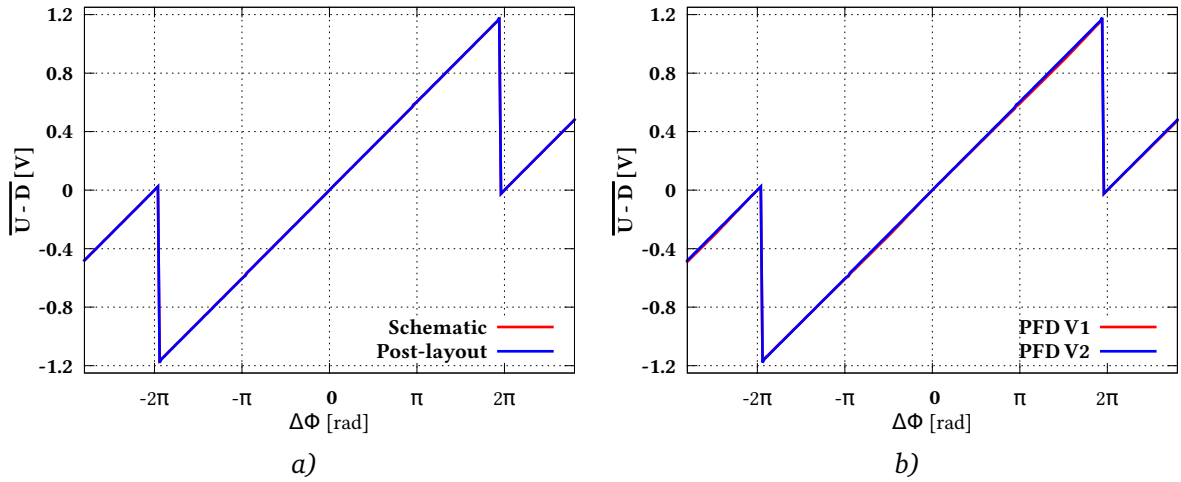


Figure 3.47: Phase and Frequency Detector gain curves. *a)* - comparison between schematic simulations and post-layout simulation (2nd version), *b)* - comparison between two circuit versions.

3.2.3 Charge Pump (CP)

The CP used in SALT_PLL (and also SALT_DLL) is based on typical architecture, widely described in section 2.3.3. The simplified schematic of the Charge Pump is shown in figure 3.48. The input bias current I_{cp} biases transistor M_8 , which controls currents in sources M_1 , M_2 and M_9 , M_{10} . The transistors M_5 , M_7 work as switches, controlled by signals \bar{U} and D , which are responsible for charging or discharging the LPF (R_1 , C_1 and C_2). The extra switches M_4 and M_6 , controlled by the signals complementary to \bar{U} and D (U and \bar{D}), ensure that M_2 and M_{10} are always well biased and have constant current. To obtain constant current in transistors M_2 and M_{10} the voltage on net V_X should be equal to the V_O voltage. To meet this requirement an operational amplifier A_1 is used. The A_1 is based on folded cascode architecture, which is widely described in literature [43]. For better operational amplifier performance a Recycled Folded Cascode (RFC) was implemented [39, 40, 41].

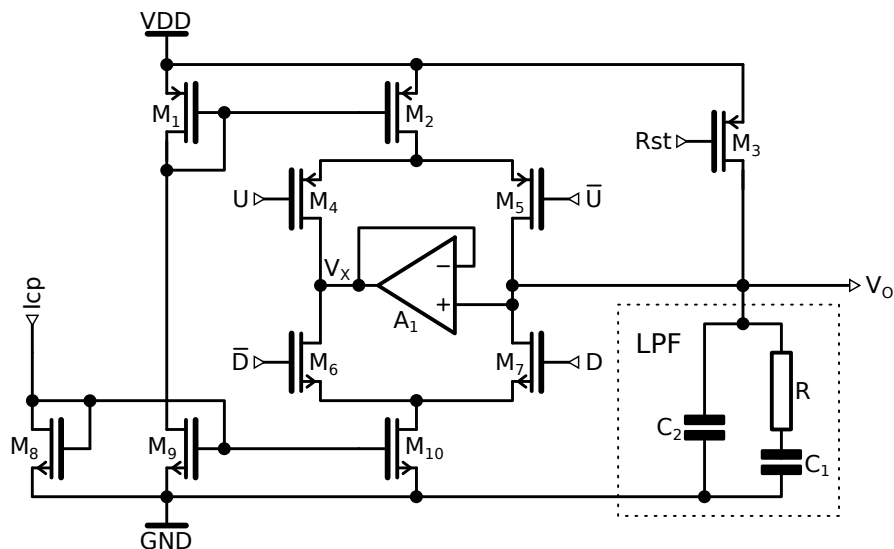


Figure 3.48: Simplified schematic of the Charge Pump with LPF.

The transistor M_3 , controlled by the Rst input, allows to charge the LPF to VDD, what is very important in DLL circuit. When the CP works in PLL circuit, transistor M_3 does not matter and Rst input should be always in high state. Figure 3.49 shows the CP operation for two circuit versions. The architecture of two CP versions is exactly the same, and both simulations were done at the same control signals U and D , however the results of simulations of the two circuits are completely different, what shows the differences between two CMOS technologies.

The V_O voltage in the 1st circuit version looks better than in the second circuit version. The drops of V_O voltage, when M_5 is disconnected, are much smaller in the 1st circuit version, but the SALT_PLL_V2 still achieves the better jitter performance.

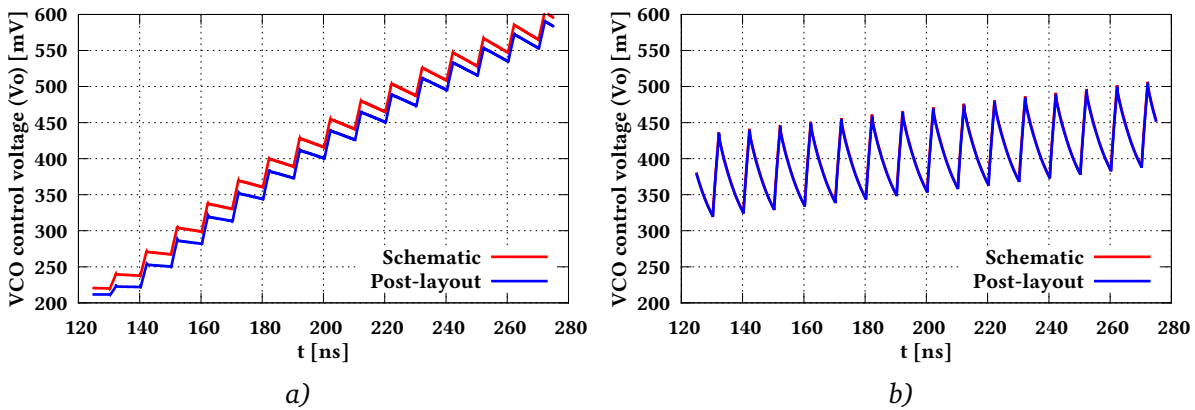


Figure 3.49: CP output voltage V_O . The U signal is slightly wider than D - V_O rises. *a)* - 1st circuit version, *b)* - 2nd circuit version.

3.3 Design and simulations of SALT_DLL

The SALT_DLL was designed as a dedicated SALT block, needed to adjust the phase of the input clock. The DLL was designed in two versions (1st and 2nd), each of them in different technology. The first was designed in 130 nm CMOS technology A, while the second in 130 nm CMOS technology B. The both circuit versions operate at similar frequency range, from 30 MHz up to 50 MHz, but SALT_DLL is optimized to work at 40 MHz. The block diagram of DLL, presented in figure 3.50, is very similar for two circuit versions. The DLL architecture is quite typical, widely described in literature [90, 91, 92], but there are few improvements. In the 2nd circuit version an extra DAC was added for better control of bias currents. The VCDL is the main component of the DLL and can generate 64 independent clock phases ($Ph[0] - Ph[63]$) and one extra phase ($Ph[64]$) needed for the DLL synchronization. The VCDL works without any digital configuration and the total delay is controlled only by the bias current $I_{vc dl}$ and the voltage input V_O . More information about the VCDL can be found in subsection 3.3.1.

The reference input signal Ref (default 40 MHz) goes directly to the VCDL, the first delay stage produces $Ph[0]$ signal, which is compared with the $Ph[64]$ by the PFD. The digital signals U and D are generated and converted by the CP to the appropriate current pulses, which allow to change the VCDL control voltage V_O . The detailed description of PFD and CP, which are exactly the same as for the SALT_PLL, can be found in subsections 3.2.2 and 3.2.3. When the SALT_DLL is synchronized the $Ph[0]$ is equal to the $Ph[64]$, the VCDL produces 64 independent clock phases ($Ph[0 : 63]$) and the time delay between them is around 390 ps at the typical reference 40 MHz. The internal clock multiplexer, controlled by input $Muxcf g[5 : 0]$, ensures the clock phase selection (1 of 64).

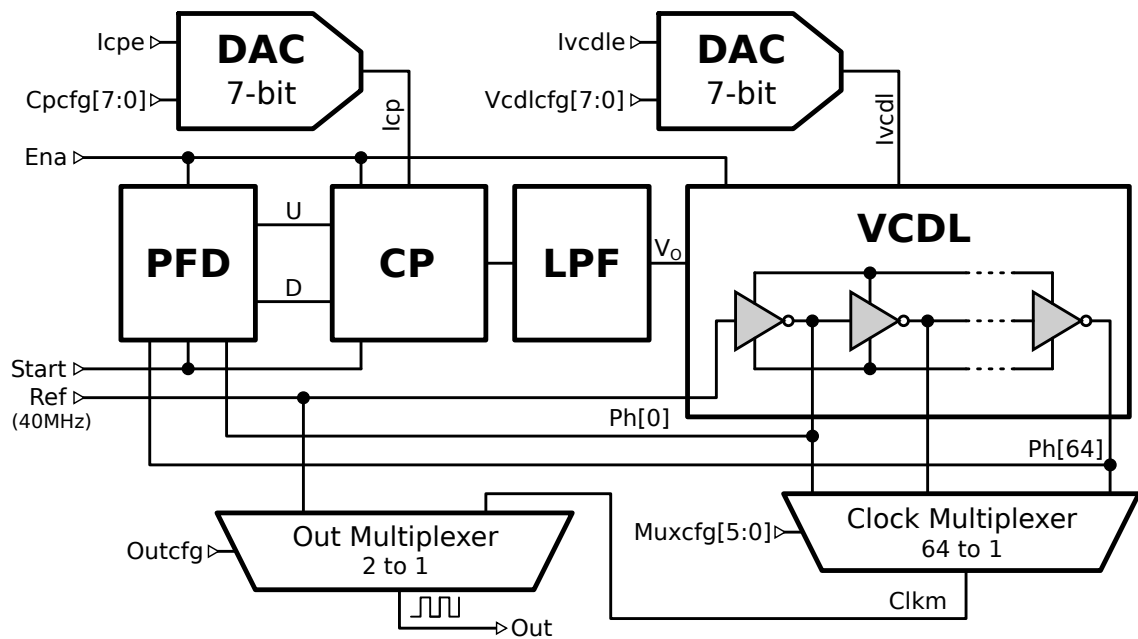


Figure 3.50: SALT_DLL block diagram.

Another multiplexer (only in 2nd circuit version), controlled by the *Outcfg* signal allows to switch on/off the clock phase selection. The construction of the multiplexers is typical, widely described in literature [56, 82]. When *Outcfg* is set to 0 the DLL output *Out* is connected directly to the reference *Ref*, while in opposite case the VCDL clock is available at the output *Out*. This functionality is added to avoid randomly shifting clock edges at the DLL output during the synchronization process, which takes some time and may be disadvantageous for digital logic.

The DLL is disabled when *Ena* input is low. Enabling the circuit, after setting *Ena* to high state, requires special startup procedure. When the DLL is enabled and *Start* input is in low state, the circuit is in standby mode, which should take at least $1 \mu\text{s}$. After that, when *Start* input goes to high state, the DLL synchronization process starts.

In the 1st DLL version the VCDL and CP reference currents (I_{cp} and I_{vcdl}) are controlled from outside via input pads, while in the 2nd circuit version the same currents can be changed directly by internal 7-bit DACs, which allow to avoid the bias current fluctuations, introduced from outside.

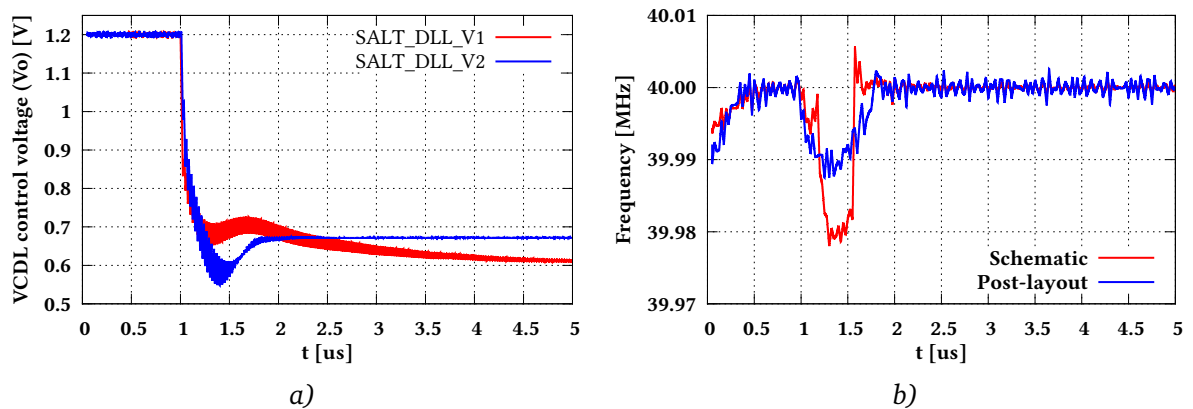


Figure 3.51: Comparison of two SALT_DLL versions and frequency stability during synchronization process. a) - two circuit version comparison, b) - output frequency stability (2nd version).

The bias current DACs are controlled by signals $Cpcf g[7 : 0]$ and $Vcdlcf g[7 : 0]$. When the DACs are disabled the bias current can be controlled in the same way as for the 1st circuits version, using the inputs $Icpe$ and $Ivcdle$. The description of the bias current DAC can be found in section 3.2.

Figure 3.51a shows the comparison of synchronization process for the two circuit versions. The SALT_DLL_V2 synchronization time is much shorter and the circuit reaches faster the stable operation condition. Even if the reference input clock has stable frequency and the VCDL introduces only some delay to this signal, the DLL output frequency is changing during the synchronization process, what is presented in figure 3.51b. The output frequency drop is very small, around 0.1% of the Ref input frequency, but it still may be disadvantageous for the digital circuit connected to the DLL.

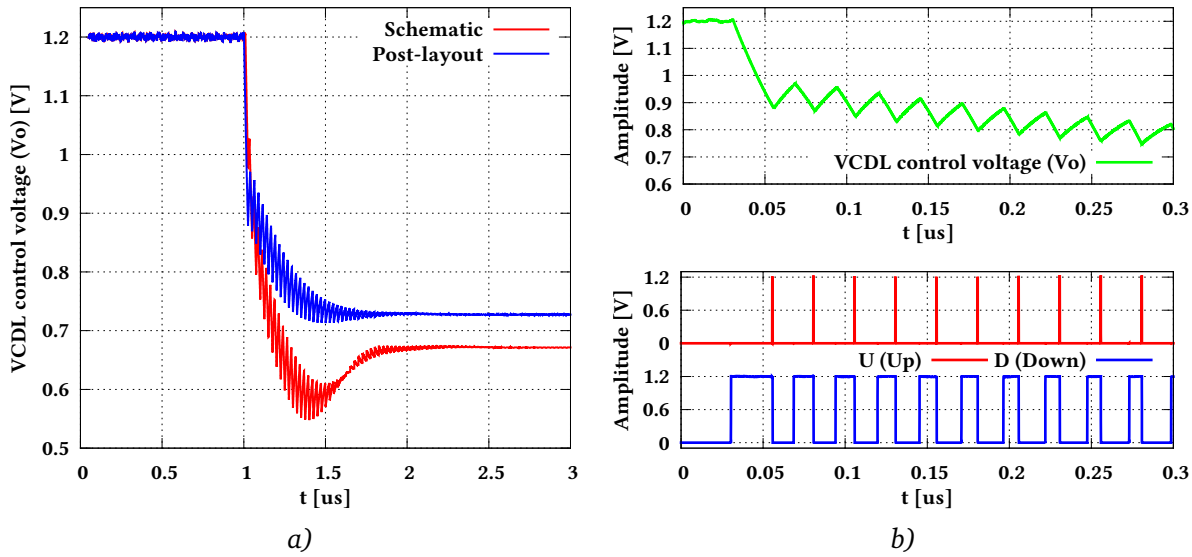


Figure 3.52: SALT_DLL synchronization process. a) - comparison of the schematic and post-layout simulations, b) - VCDL control voltage (V_O) with the corresponding signals (U and D).

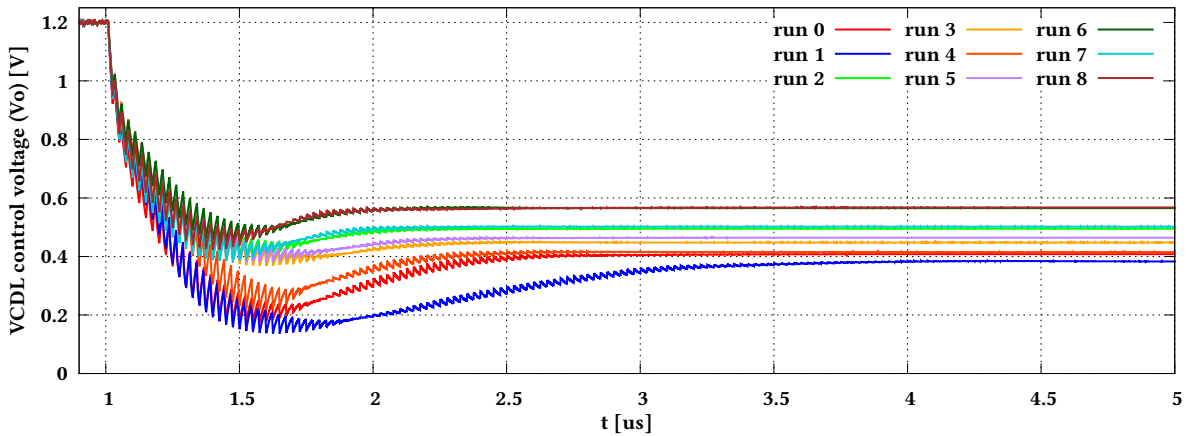


Figure 3.53: SALT_DLL MC analysis of the synchronization process - schematic simulations.

The comparison between the schematic and post-layout simulations is presented in figure 3.52a. The circuit behavior in both cases is very similar, the post-layout simulations show that the V_O voltage after the synchronization is moved into higher values, which is directly related to the parasitic components in the circuit. The VCDL control voltage compensates the higher values of total delay

experienced in post-layout simulations. Figure 3.52b shows the V_O voltage at the beginning of DLL synchronization process, together with the corresponding U and D signals.

Figure 3.53 presents the MC analysis of the SALT_DLL, which show how the circuit behaves when the component dimensions (especially transistors) are changed randomly. The plot presents simulations results for 9 different (randomly selected) MC seeds and shows that the circuit works well in all these cases. The V_O voltage at the end of the DLL synchronization process is slightly different for each of the simulation runs.

3.3.1 Voltage-Controlled Delay Line (VCDL)

The VCDL is very similar to the VCO circuit, described in section 3.2.1. The main difference between these two circuits (described in theory in section 2.3.1) is in the feedback, which is not present in the VCDL. Figure 3.54 shows a simplified block diagram of the SALT_VCDL, which is exactly the same in the both circuit versions. The PMOS/NMOS current mirrors are very similar to the VCO current sources, but in this case, the current multiplication factor is different. The current bias is controlled by the I_{vcdl} input and V_O input. The first allows to change the reference current, while the second is a standard control voltage input. The VCDL consists of 65 active delay stages and one dummy stage at the end (not presented in the figure), which is needed to keep the same load for all clock phases. The SALT_VCDL gives 65 clock phases, 64 of which go to the output multiplexer in the DLL and the last one is used as the feedback connected to the PFD. Each of the delay stages is built with two CSIs.

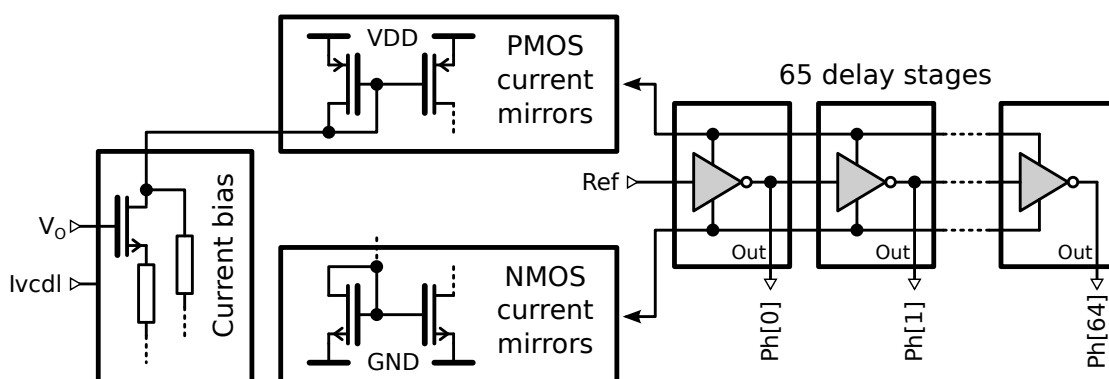


Figure 3.54: SALT_VCDL block diagram (1st and 2nd circuit versions).

The schematic of the delay stage, shown in figure 3.55 is the same for two circuit versions. The transistors $M_1 - M_4$ are main components of the circuits, because they are responsible for the propagation time of delay stages. The Su nodes of all stages are controlled by PMOS current mirrors (figure 3.54), while the Sd nodes are controlled by NMOS current mirrors. Thanks to this current limitation, transistors $M_1 - M_4$ work as CSI, which is needed for the delay adjustment. The Rin input and $Rout$ output are used to connect delay stages in series. The transistors $M_5 - M_8$ work as signal buffers. The M_5, M_6 is a dummy inverter, while the output signal comes only from the inverter M_7, M_8 . The transistors $M_5 - M_8$ work also with current limitation, from the nodes $Su2$ and $Sd2$, allowing to reduce the influence of power supply voltage fluctuations, so the jitter performance of the circuit is better. The transistors $M_{11} - M_{13}$, controlled by enable input Ena (\overline{Ena}), are used in the idle state to keep the proper logic state at the stage output Out . The output signal is buffered and restored to the full swing (0 – 1.2 V) by the inverter M_9, M_{10} .

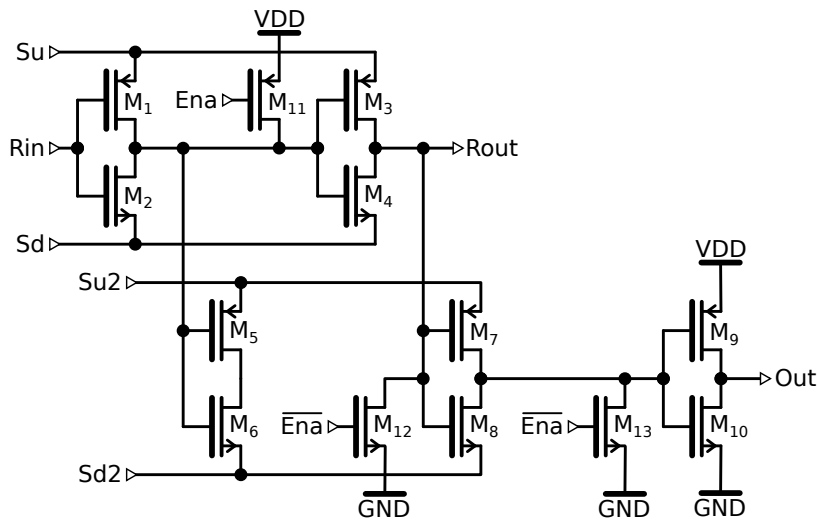


Figure 3.55: Schematic of the SALT_VCDL delay stage.

The schematics of the both VCDL versions are exactly the same, but the circuit behavior is different for the same simulation conditions. Figure 3.56a shows the total delay of the VCDL. The 2nd circuit version, designed in 130 nm CMOS technology B, achieves smaller delays at the same bias currents than the 1st circuit version, designed in 130 nm CMOS technology A. It proves that the analogue parameters of transistors are different in two technologies. The comparison between the schematic and post-layout simulations of SALT_VCDL_V2 is shown in figure 3.56b. The total delay is much higher when parasitic components are present. At typical bias current $10 \mu\text{A}$ the schematic simulations give the total VCDL delay around 15 ns, while the post-layout simulations give two times higher delay.

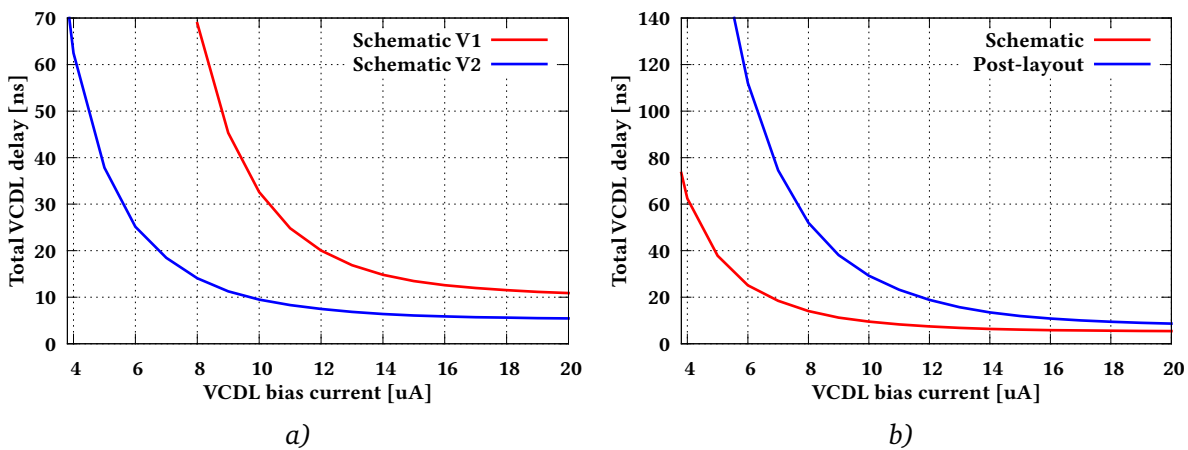


Figure 3.56: Comparison of two SALT_VCDL versions and comparison of schematic and post-layout simulations at control voltage $V_O = 0.6 \text{ V}$. a) - two circuit version comparison, b) - comparison of schematic and post-layout simulations (2nd version).

The jitter at the VCDL output $Ph[64]$ (last clock phase) is presented in figure 3.57. Its value strongly depends on the bias current, and for the current over $13 \mu\text{A}$ the VCDL achieves the jitter below 10 ps.

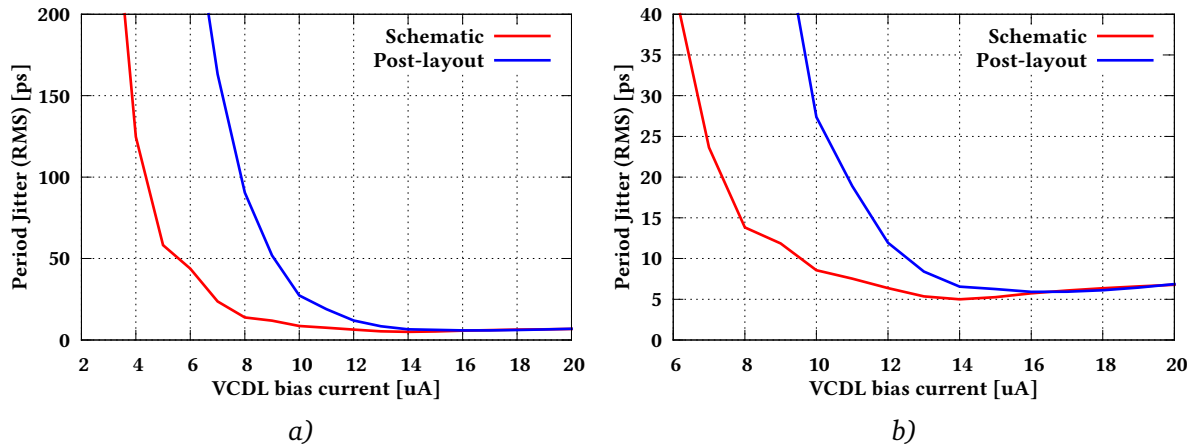


Figure 3.57: SALT_VCDL_V2 output jitter (Ph[64]) for different bias currents at control voltage $V_O = 0.6$ V. a) - jitter versus current, b) - zoomed jitter plot.

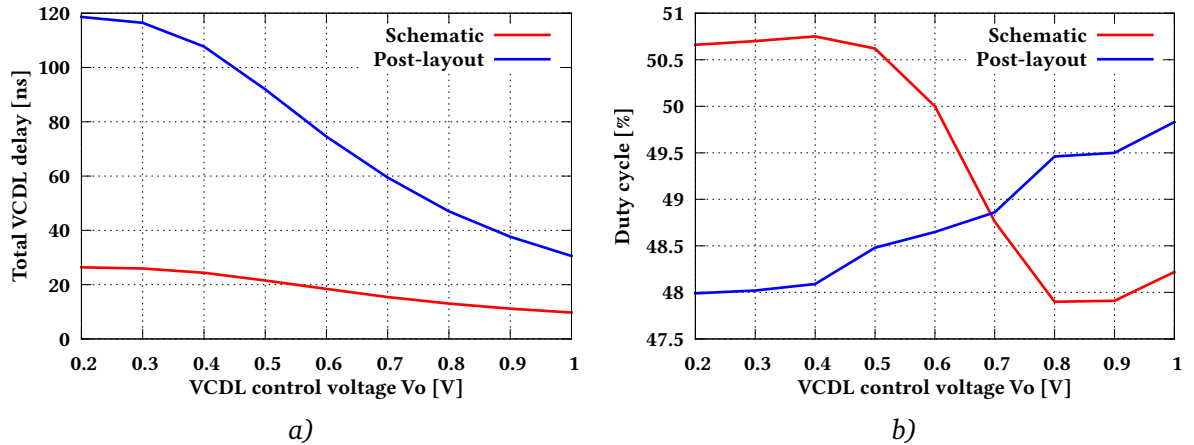


Figure 3.58: SALT_VCDL_V2 gain curve and duty cycle of the Ph[64] output. a) - total delay versus V_O control voltage, b) - duty cycle of the last VCDL stage.

Figure 3.58 shows the gain curve of the VCDL and the duty cycle at the last stage output, depending on the V_O control input. The gain obtained from the post-layout simulations is completely different than the one obtained from the schematic simulations. For this reason all circuit optimization was done based on the post-layout simulations. The duty cycle is stable and is always in range 47.8% – 50.7%.

Chapter 4

Measurements results

The measurements of circuits, which generate and process the clock signal, such as PLL or DLL, are primarily focused on frequency, phase shifts between waveforms, and jitter. This last parameter is the most important for operation of digital circuits and is very useful in calculating their timing margins, consequently the main part of this dissertation is focused on the jitter.

Figure 4.1 shows the data setup time (t_s) and the data hold time (t_h) violations caused by the clock jitter, which is clearly bounded to the changes of the clock signal period in time. When the clock period is changing, it is possible to sample the data at wrong moments in time. In the first case, when the rising edge of clock signal occurs too early, the data can be sampled before it is valid and may have an incorrect value. This case presents the data setup violation, related to the data sampling before the setup time (t_s) elapses. Similarly, when the rising edge of clock signal occurs too late, the data hold time (t_h) is effectively reduced and an incorrect data may be read. So it is very important to check if the clock signal jitter is small enough to fulfill the requirements of the digital circuit. The timing margins in digital circuits must be sufficiently liberal, so that the clock jitter does not cause incorrect system operation. More information about the jitter is presented in subsection 4.1, which describes the methodology of jitter measurements, using a digital oscilloscope.

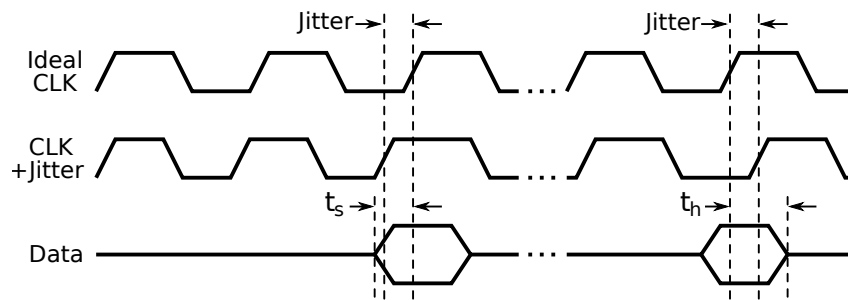


Figure 4.1: Data setup time and data hold time violations caused by the clock jitter.

After design of an integrated circuit and fabrication of prototype ASIC, the next step is to prepare the measurement setup. This allows to check a functional operation of the circuit, but also allows to evaluate its performance and finally compare the circuit with the post-layout simulations. The prototype ASIC is supplied as a bare piece of silicon, without any housing, in opposite to typical commercial integrated circuits. The tests can be performed by using a probe-station, where pads with size of tens of micrometers are connected to external devices via a dedicated probe-card with needles, but only when external capacitances are not important for a proper circuit operation. This method cannot be used for circuits operating at high frequencies. The second solution is to glue

the prototype ASIC directly to a dedicated Printed Circuit Board (PCB) and make the connections using wire bonding [31]. A much smaller parasitic capacitances, given by this method, allow to test circuits, which operate at high frequencies.

A dedicated PCB contains mostly power and bias circuitry, required for a prototype chip, and the appropriate interfaces, which allow to connect a test equipment. The PCB contains also programmable circuits, which can be used for the measurements control, their automation, or collecting a large number of digital data. The construction of the measurement setup and measurement results of the MULTI_PLL are shown in subsection 4.2. Subsection 4.3 presents similar description for the PLL (SALT_PLL) and the DLL (SALT_DLL) prototypes designed for the SALT project.

4.1 Methodology of jitter measurements

The jitter is a timing variation of the clock signal edges from their ideal positions in time. The jitters in clock signals are typically generated by noise or other disturbances in the circuit, like: power supply variations, transistors transient noise, thermal noise and interference from other circuits.

The jitter can be measured in many ways, but this dissertation takes into consideration three main types of jitter, namely: period jitter, cycle to cycle jitter and long term jitter. All these types of jitter are defined in the JEDEC Standard 65B [93]. All measurements can be performed in the time domain by measuring the clock period and its changes over time.

The most common instrument used in this measurements is a real time digital oscilloscope. For a high frequency clock measurements the scope should have very high sampling rate. Usually it is a value around 40 GSps (Giga Samples per second) or even higher. The digital scope samples its inputs at the regular time intervals, based on internal time reference. Figure 4.2 shows the digital scope sampling operation.

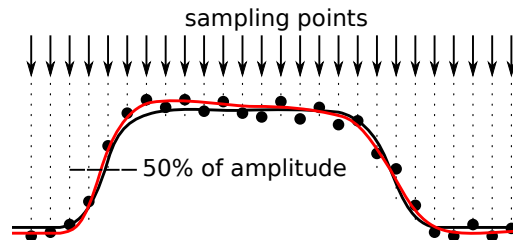


Figure 4.2: Sampling of a signal by a digital scope.

The arrows at the top of the figure present the sampling points, black solid line is the actual signal, dots are the sampled values and dashed line on the positive edge presents the point when the input signal reaches half of its amplitude. The signal displayed by the scope (red solid line) is the curve fitted to the sampled points. Figure 4.2 shows, that the sampled values do not always match to the actual signal and there are some shifts of the measured signal edge. Even if the oscilloscope presents the results as a fitted curve on the screen, the digital data read from the oscilloscope are given as a table of sampled points.

The slopes of the measured signal are usually steep, so the scope gives only a few sample points per slope. In result, to get the time value for which the signal slope reaches half of its amplitude (based on the digital samples), a data interpolation method is needed. Based on the measured positions of the signal slopes, the calculation of the clock period and its fluctuation can be done, which then leads to calculation of the jitter values.

4.1.1 Period jitter

The period jitter is defined as a deviation of the signal period with respect to an ideal period over a large number of random cycles. The specification says that the period jitter should be measured over a sample of 10000 cycles. If a large number of individual clock periods is collected, each of them can be measured and the average clock period, as well as the standard deviation and the peak-to-peak (Pk-Pk) value, can be calculated. The standard deviation and the peak-to-peak value are referred as the **RMS** value and the Pk-Pk value of the period jitter [94, 95]. In literature the **RMS** value of the period jitter is commonly called: period jitter, **RMS** jitter, or just jitter.

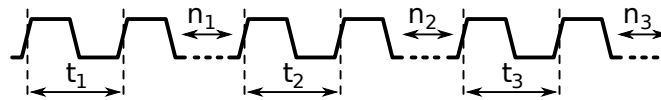


Figure 4.3: Measurements of a clock period for the period jitter calculations.

In the literature the period jitter is commonly defined as the difference between the measured clock period and the ideal period, but in practice the ideal period is difficult to know. For example; the period measurements of a Voltage-Controlled Oscillator, which oscillates at the frequency equal to 1 GHz, usually give values 0.998 ns or 1.003 ns instead of 1 ns. So it is usually more practical to use the average period as the ideal period.

Figure 4.3 shows the methodology of the period jitter measurements. It can be measured using the following procedure:

1. Configure the scope to capture a few clock cycles (2-3) on the screen. Get the digital data from the scope and store it. Such a short sampling time allows to use the highest resolution of the oscilloscope;
2. Calculate the period of one cycle (t_n) and record it;
3. Wait random number of cycles (n_1, n_2, n_3, \dots);
4. Repeat steps 1-3 10000 times and record the table of 10000 period values (t_1, t_2, t_3, \dots);
5. Calculate the mean, the standard deviation, and the peak-to-peak values based on the table.

The mean and the standard deviation values are usually quite accurate but the unbounded nature of the peak-to-peak value shows that it is not always accurate. To obtain a more accurate peak-to-peak value, all previous steps should be repeated 25 times. The recorded peak-to-peak values after each run give a more accurate peak-to-peak value (average from the 25 results) [95].

4.1.2 Cycle to cycle jitter

The cycle to cycle jitter is defined as the variation in the cycle time of a signal between adjacent periods, over a random selection of adjacent periods. The JEDEC Standard specifies that each sample size should be greater than 1000, but for better statistics it is better when the jitter is calculated from 10000 samples [93].

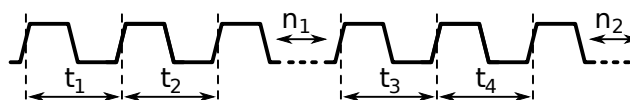


Figure 4.4: Measurements of a clock period for the cycle to cycle jitter calculations.

The cycle to cycle jitter involves only the difference in periods between two consecutive cycles and there is no reference to an ideal period. This type of jitter is commonly expressed as a peak-to-

peak value in ps, which defines the maximum deviation between any two consecutive rising clock edges, but also the cycle to cycle jitter can be expressed as a **RMS** value in ps. The cycle to cycle jitter is commonly used to show the stability of the clock when its spectrum is spread. The period jitter is more sensitive to the frequency spreading while the cycle to cycle jitter is not. Figure 4.4 shows the methodology of the cycle to cycle jitter measurements. It can be obtained using the following procedure:

1. Configure the scope to capture a few clock cycles (2-3) on the screen. Get a digital data from the scope and store it. Please note that this step is exactly the same as in the period jitter measurements (important for practical reasons);
2. Calculate the difference between two adjacent periods ($t_1 - t_2$) and record its absolute value ($|t_1 - t_2|$);
3. Wait random number of cycles (n_1, n_2, \dots);
4. Repeat steps 1-3 10000 times and record the table of 10000 values ($|t_1 - t_2|, |t_3 - t_4|, \dots$);
5. Calculate the standard deviation, and the peak-to-peak values based on this table. The peak value is the largest number in the data set.

To obtain a more accurate peak-to-peak value, all previous steps should be repeated 25 times. Follow the same procedure as for the peak value of the period jitter to obtain a more accurate peak-to-peak value.

4.1.3 Long term jitter

The long term jitter is a measure of the change in a clock edge position from its ideal position, over a large number of the consecutive cycles. This type of jitter presents a cumulative effect of the jitter over a long time. It is different compared to other jitter types, which show the clock variations in a short time interval. The long term jitter can be expressed as a **RMS** or peak-to-peak value, but if it is not specified, it usually means the **RMS** value.

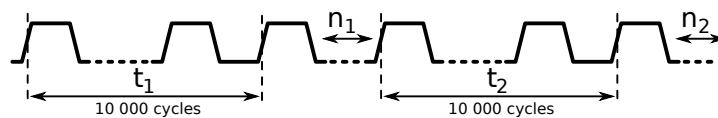


Figure 4.5: Measurements of a clock period for the long term jitter calculations.

The number of cycles used in the long term jitter measurement depends on application, but it is usually around 1000 or even 10000 periods. Figure 4.5 shows the methodology of the long term jitter measurements. It can be done using the following procedure:

1. Configure the scope to capture a few clock cycles (2-3) on the screen, but the time offset should be set to value equal to 10000 periods of currently measured clock (clock period should be measured before the long term jitter measurements). Get a digital data from the scope and store it;
2. Calculate the time interval of 10000 periods (t_1) based on time position of 10000th positive edge, currently shown on the screen;
3. Wait random number of cycles (n_1, n_2, \dots);
4. Repeat steps 1-3 10000 times and record the table of 10000 values (t_1, t_2, \dots);
5. Calculate the standard deviation and the peak-to-peak values of the long term jitter based on this table.

4.2 Measurements of MULTI_PLL

The MULTI_PLL chips (two prototypes: 1st [96] and 2nd) were designed in 130 nm CMOS technology A as a general purpose blocks. Two prototypes have the same functionality but in the second one the clock jitter is optimized and the frequency ranges are slightly extended. The basic measured parameters of those circuits are summarized in table 4.1. The simulated frequency range also presented in the table is slightly smaller than the VCO working range, because PLL comprises also other blocks.

Parameter	MULTI_PLL_V1	MULTI_PLL_V2
Frequency range (simulated)	10 MHz – 3.5 GHz	10 MHz – 3.5 GHz
Frequency range	30 MHz – 1.25 GHz	30 MHz – 1.3 GHz
MULTI_VCO modes	16	16
Division factors	6, 8, 10 or 16	6, 8, 10 or 16
Power consumption	0.65 mW @ 1 GHz	0.7 mW @ 1 GHz
Period jitter	10 – 130 ps	5 – 60 ps
Chip area	300x300 μm^2	300x300 μm^2

Table 4.1: Basic parameters of two MULTI_PLL prototypes.

Both prototypes of the MULTI_PLL work in a very wide frequency range and have very low power consumption. The MULTI_VCO operates in 16 modes (frequency ranges), which can be changed manually or automatically. The MULTI_PLL was not fabricated as a separate ASIC, but it is a part of a multi-channel 10-bit ADC (called ADC_10) ASIC and it is also integrated with a multi-channel 6-bit ADC (called ADC_6) ASIC. In the both ASICs it is used for fast data serialization. Both ASICs

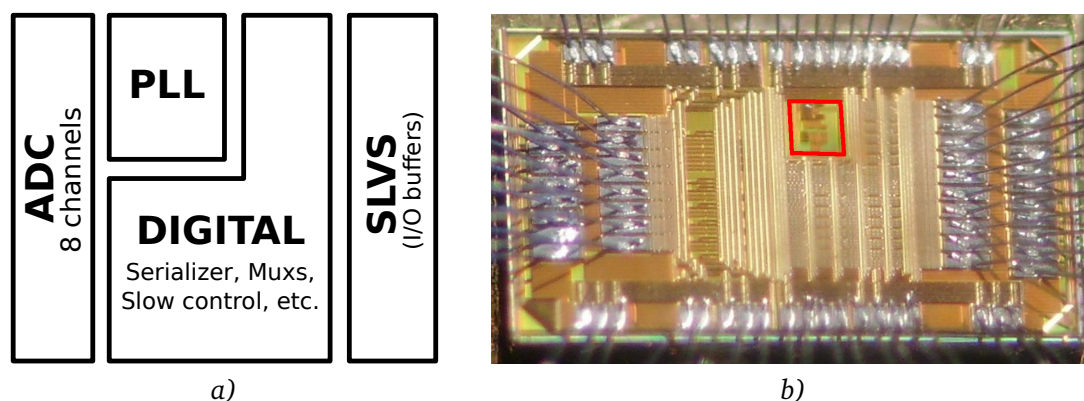


Figure 4.6: MULTI_PLL as a part of the multi-channel ADC. a) - floorplan, b) - micrograph of the prototype 6-bit ADC.

were fabricated in two versions (1st and 2nd) so the MULTI_PLL was integrated and tested in four different prototypes. In those ASICs, the Phase-Locked Loop works as a clock multiplier and allows to serialize the digital data from each ADC channel (division factors: 10 or 6, depends on data length - resolution of ADC) or serialize the data from all channels in the chip (divider 8 in both cases).

Figure 4.6a presents a simplified floorplan of both chips, and shows the place where PLL is fitted. Integration in the digital part of the ADC chips provides a slow control interface to configure the MULTI_PLL. Figure 4.6b shows the micrograph of the multi-channel 6-bit ADC, the MULTI_PLL core is marked by red rectangle.

4.2.1 Slow control interface - ASIC configuration

The configuration can be done in the same way as for the [ADC](#) components, via Serial Peripheral Interface ([SPI](#)). From the configuration point of view the `ADC_10` and the `ADC_6` are very similar. Both of them are a multi-channel [SAR ADCs](#) with the output data serialization. The main difference between the `ADC_10` and `ADC_6` is the [ADC](#) resolution. The `ADC_10` contains a 10-bit [SAR](#), while in the `ADC_6` the [SAR](#) produces 6-bit data. A general architecture of both chips (in two versions 1st and 2nd) is identical but there are small differences, which are described later.

All the [ASICs](#) work as an always selected [SPI](#) slave, which can only receive data (no send function) and work in the [SPI](#) mode 0 (active at rising clock edge). The command, shown in figure 4.7, contains following parts: header, code, sub_address¹ and data. The [MSB](#) is always transmitted first.

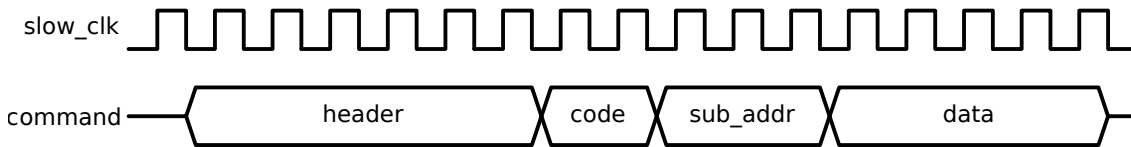


Figure 4.7: Transmission via the slow control interface - command structure.

The command header consist of two parts: 4-bit constant 1010 at the beginning and 2-bit *address*. The address in the `ADC_10` can vary (see table 4.2) and there can be up to 4 [ASICs](#), with different addresses, connected to one slow control interface. The address can be set using input pads. The address in the `ADC_6` is fixed at value 11 and there are no dedicated pads for its configuration.

ASIC	Constant part	Address part
ADC_10 (1st and 2nd)	1010	address (2b)
ADC_6 (1st and 2nd)	1010	11

Table 4.2: Command header field structure.

In the 1st version of both chips the command *code* field is 2-bit so there are four commands: `main_cfg`, `pll_cfg`, `sel_seq`, and `sar_cfg`. From the `MULTI_PLL` configuration point of view only the first two commands are useful. The field *sub_addr* is three bits long and is used only in the `sar_cfg` command. In other commands the bits in *sub_addr* do not matter and may be set to 000. The data are always 9-bit long, so in commands with the shorter data, the last bits should be 0. Moreover, commands must be separated from each other by at least one 0, so the equivalent data length is 10 bits and the whole command (with header) 21 bits. The [PLL](#) configuration commands for the 1st version of `ADC_6` and `ADC_10` are presented in table 4.3.

Command name	code	sub_address	data
<code>main_cfg</code>	00	000	mode (2b), cnt & adc (5b), pll_on (1b)
<code>pll_cfg</code>	01	000	extra_div (1b), eams (1b), cfg (6b)

Table 4.3: MULTI_PLL configuration commands - 1st version.

¹Present only in the 1st version of both chips

In the 2nd version of both chips the command *code* field is 3-bit, so up to 8 commands may be defined. From the MULTI_PLL configuration point of view there are no changes. There is not *sub_addr* field in this chip version, but similarly to the 1st ASIC version the data are always 9-bit long, so the whole command (with header) has 19 bits. The PLL configuration commands for the 2nd versions of ADC_6 and ADC_10 are presented in table 4.4.

Command name	Code	Data
main_cfg	000	<i>mode</i> (2b), <i>cnt & adc</i> (5b), <i>pll_on</i> (1b)
pll_cfg	001	<i>extra_div</i> (1b), <i>eams</i> (1b), <i>cfg</i> (6b)

Table 4.4: MULTI_PLL configuration commands - 2nd version.

The MULTI_PLL can operate with four division factors, but there is no possibility to change the division factor directly, because it is bounded with the ADC mode. The ADC_10 and ADC_6 (in two versions) can work in four different modes, but only two: parallel and serial are interesting from the PLL point of view. The mode is chosen by setting the appropriate value in the field *mode* of the **main_cfg** command, according to table 4.5.

ADC mode name	<i>mode</i> field value
parallel	00
serial	10

Table 4.5: ADC modes with the corresponding mode field values of the **main_cfg** command.

When the ADC works in parallel mode (default after hard reset) each channel (of 8) is connected to one SILVS output and sends the data serially. The MULTI_PLL works with division factor 10 or 6 (depends on the ADC resolution). The internal ADC clock is equal to the reference input clock and the output data rate is 10 or 6 (depends on ADC resolution) times faster - allows to transmit 10-bit (or 6-bit) data packet from the ADC. In the serial mode, all ADC channels send the data through one serial SILVS output. The MULTI_PLL works with division factor 8. The internal ADC clock is 10 or 6 (depends on the ADC resolution) times slower than the reference input clock and the output data rate is 8 times faster than the reference. It is important to note that the field *pll_on* of **main_cfg** command should be always set to 1 to keep the MULTI_PLL enabled.

divider	ADC_10		ADC_6	
	extra_div	ADC mode	extra_div	ADC mode
6	1	serial	0	parallel
8	0	serial	0	serial
10	0	parallel	1	parallel
16	1	parallel	1	serial

Table 4.6: PLL divider settings.

The **pll_cfg** command allows to configure the MULTI_PLL core. More information about the PLL core and its details can be found in section 3.1. The PLL in the ADC_10 uses only dividers by 10 and 8, while the PLL in the ADC_6 uses dividers by 6 and 8. To improve the testability of all division

factors the *extra_div* field in the `pll_cfg` command is needed. In normal conditions the *extra_div* field is always equal to 0, but setting it to 1 allows to test the MULTI_PLL with two other dividers (by 6 and 16 for ADC_10, by 10 and 16 for ADC_6). All informations about selection of the PLL divider are summarized in table 4.6. The *eams* field of `pll_cfg` command controls directly the *Eams* input of the PLL, while the *cfg* field controls its *Cfg[5 : 0]* input.

4.2.2 Details of measurement setup

The measurements cannot be done without a prototype Printed Circuit Board (PCB), which provides electrical connections and some components supporting the prototype ASIC. The PCB provides also a mechanical montage of the chip, which can be glued to the PCB and wire-bonded to its pads.

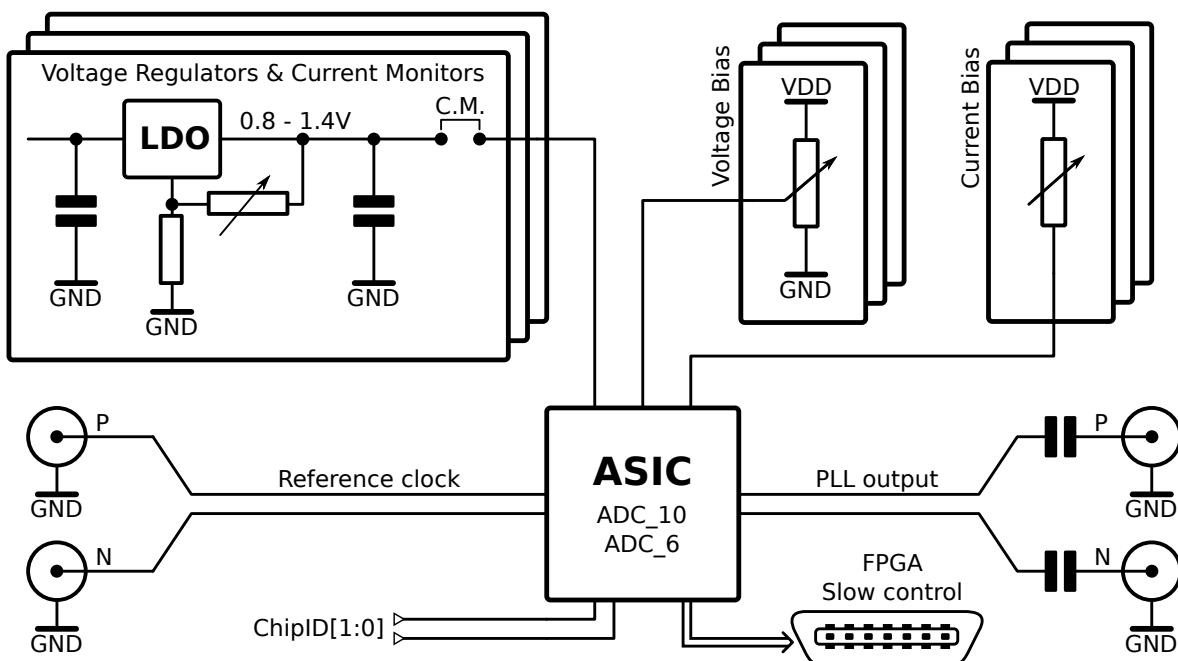


Figure 4.8: Simplified block diagram of the prototype Printed Circuit Board (ASIC board).

Figure 4.8 shows a simplified block diagram of the prototype Printed Circuit Board. The figure contains only the components needed for the PLL measurements, all elements related to the ADC testing were omitted. The ASIC board contains voltage regulators, which allow to adjust all voltages, which bias/supply the chip. In result, it is possible to independently regulate the value of the power supply voltage in few separate channels. This provides a separation between the analogue and the digital supply domain. The multiple supply voltages allow to avoid the interference coming from another circuits via the power rail, which is especially important in the case of the SLVS buffers. Their high power consumption generates a large noise on the power rails. Each of the voltage regulators has the possibility of current monitoring (jumper). In order to improve the functionality, the PCB allows to bias the entire ASIC from a common power supply (one voltage regulator), or connect the supply voltage from an external source, omitting the voltage regulator. The board contains also a large number of decoupling capacitors, which are not shown in diagram 4.8. The test board contains also the circuits, which generate bias voltages and bias currents. In this case the biasing circuits take the form of potentiometers and voltage dividers. There is a possibility to connect an external precision voltage source or a current source.

The reference clock is fed to the differential clock input of the prototype ASIC via SLVS buffer. The PCB contains a differential clock buffer/multiplexer with a logic level translator. The clock multiplexer proved to be necessary from the ADC measurement point of view and its presence does not impede the measurements of the MULTI_PLL. The PLL output was made in the SLVS standard. There are no additional clock buffers on the test board and the ASIC output is connected directly to a connector on the board. For the proper SLVS operation two serial capacitors are used, which allow to obtain Alternating Current (AC) coupling at the output. The PCB allows to configure the chip address using two jumpers (present only in ADC_10). Reading the digital data from the ASIC can be done using Very-High-Density Cable Interconnect (VHDCI) connector, which allows direct connection between the PCB and the evaluation board with Field-Programmable Gate Array (FPGA). For the PLL measurements a digital communication is used only to configure the chip via the SPI.

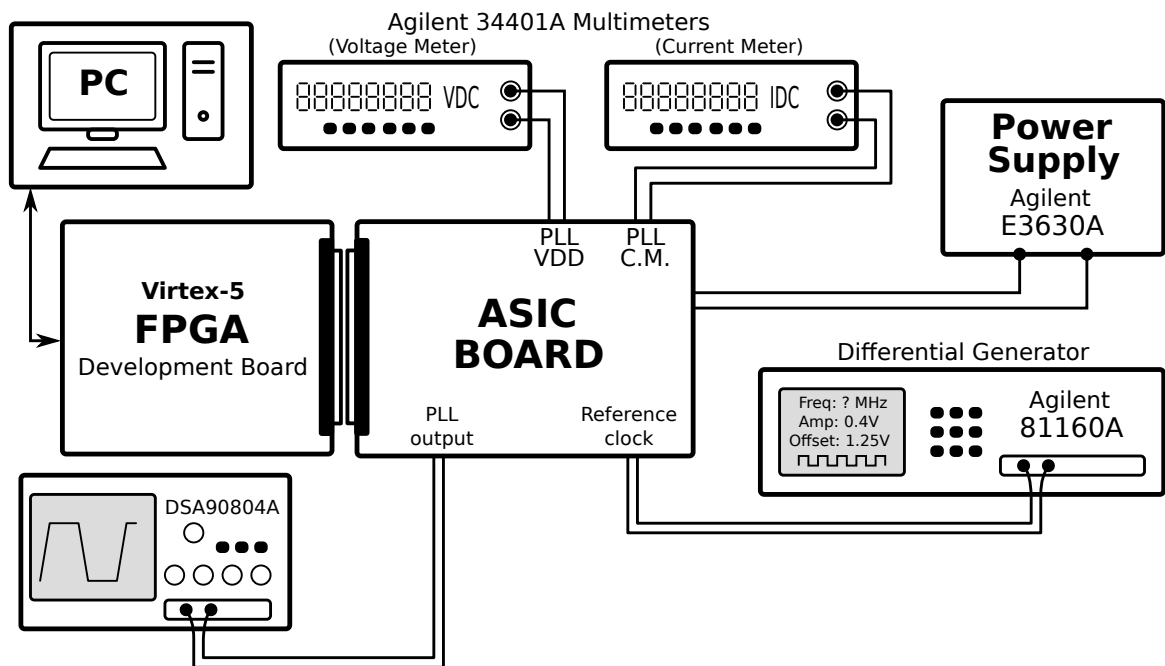


Figure 4.9: Block diagram of the MULTI_PLL measurements setup.

The prototype PCB allows to connect the measured ASIC to external world and contains only the most important components, used for chip powering, biasing and buffering I/O signals. Measurements of the prototype ASIC are not possible without laboratory instruments like: generators, multimeters, oscilloscopes etc., which can generate the input signals and measure the key parameters of the chip. Figure 4.9 shows a block diagram of the MULTI_PLL measurement setup, and figure 4.10 shows how the measurement setup looks like.

To power the PCB an external laboratory power supply (Agilent E3630A) is used, which is connected directly to voltage regulators on the board. The voltage and current measurements can be done by digital multimeter (Agilent 34401A), which is connected directly to the dedicated connectors on the board, working as voltage(current) monitor. The reference input clock is given by differential generator (Agilent 81160A) and can vary in the range 1 – 220 MHz. All frequency and jitter measurements of the PLL output were done using a 40 GSps oscilloscope (Agilent DSA90804A). To improve automatization of the measurements, all laboratory equipment is controlled by a computer through Universal Serial Bus (USB) and Local Area Network (LAN) interfaces. To perform the data analysis and calculate the parameters, which characterize the measured ASIC, a dedicated software was developed.

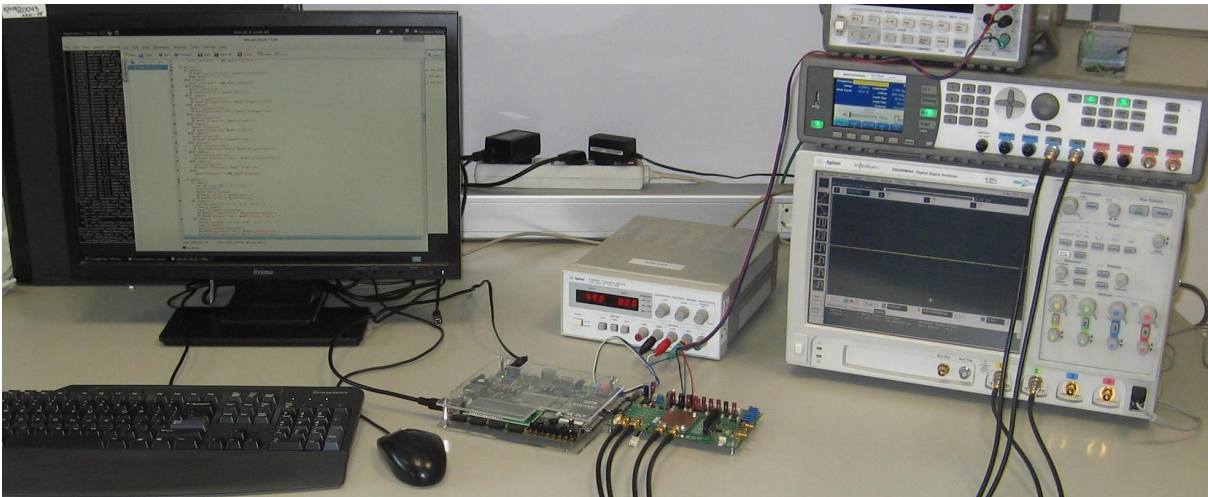


Figure 4.10: Photograph of the measurement setup.

The measurement software has a form of a dedicated python script which allows to:

- Communicate with the Virtex-5 **FPGA** board via serial interface (**USB** to serial converter is used). The **ASIC** configuration can be changed using slow control interface, implemented in the **FPGA**;
- Set differential waveform generator parameters like: frequency, amplitude, offset, etc. via **LAN** interface;
- Read digital data samples (via **LAN** interface), which contain the measured data by the oscilloscope and calculate the frequency, period and all jitter types, based on this data;
- Read the measured values of the chip power consumption and supply voltage from a digital multimeter via serial interface (**USB** to serial converter is used);
- Perform automatic frequency scans with selected range and step. It is possible to define multiple scans with different **ASIC** configurations, which start after each other;
- Check the previously obtained measurements results and rerun the measurement if needed. The validation procedure is based on a difference between the expected output frequency (dependent on selected divider and reference frequency) and the measured frequency. In the second step the procedure checks if the jitter values are not greater than the selected level;
- Create many different types of charts.

4.2.3 Measurements results

The results presented in this subsection are based on the measurements of three **MULTI_PLL_V1** chips and four **MULTI_PLL_V2**. Many plots were created, but only few examples are presented in this subsection, the rest is available in appendix **A**. As previously mentioned, the **PLL** circuits, integrated in the **ADC_10** and **ADC_6** are exactly the same, but there are some differences in internal routing (from block to block) between these two **ASICs**. The most important is the fact that the dedicated **VDD** pad is available only in the **ADC_10**, so the power consumption measurements are possible only for this **ASIC**.

Figure 4.11 shows the gain measurements plots for both versions of the **MULTI_PLL**. Each of them presents four curves for dividers by: 6, 8, 10 and 16. There are some gaps (also in 2nd chip version – see appendix **A**) in the **PLL** working range, which depend on the selected divider and the measured

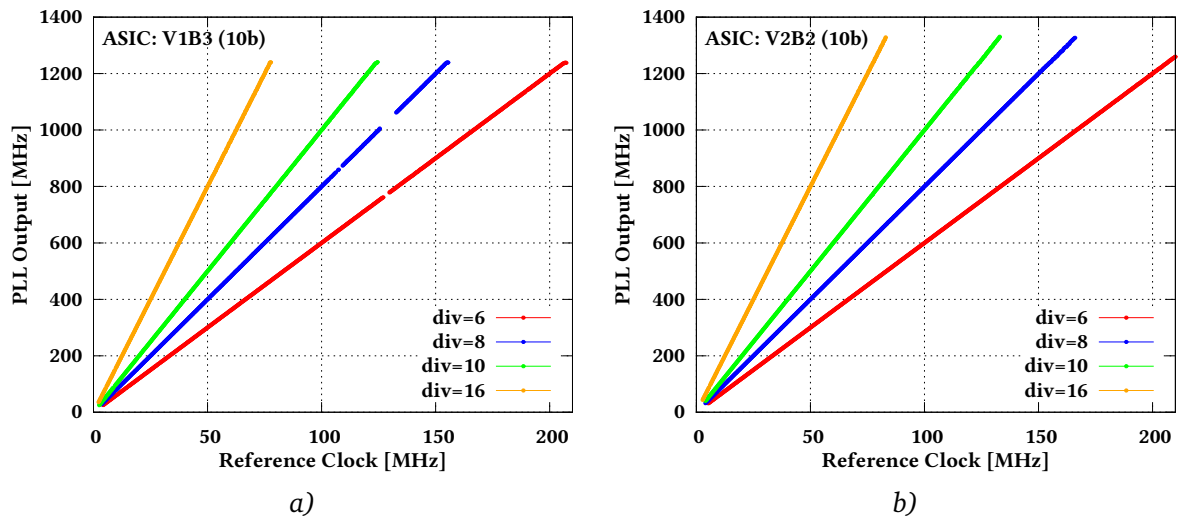


Figure 4.11: Gain measurements results of the MULTI_PLL. a) - 1st version, b) - 2nd version.

chip, because the VCO modes (frequency ranges) do not overlap in all cases. The MULTI_PLL in both versions should work up to 3.5 GHz (post-layout simulations) but the SLVS output buffer limits the frequency to around 1.3 GHz. The MULTI_PLL_V2 has a bit wider working frequency range. The measurements for both PLL versions and all division factors are presented in figures A.1 and A.2 in appendix A.

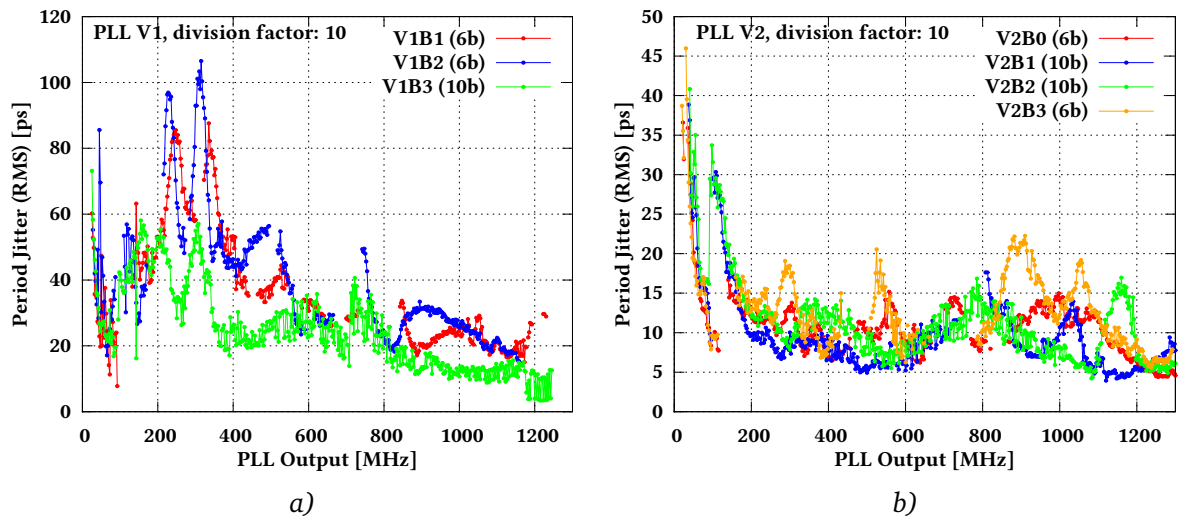


Figure 4.12: Period jitter measurements results of the MULTI_PLL. a) - 1st version, b) - 2nd version.

The most important parameter from the PLL measurements point of view is the jitter, in particular the period jitter is the most popular. The example measurement of this parameter is shown in figure 4.12, which presents the measurements for both chip versions. The MULTI_PLL_V2 achieves a better jitter performance than the previous chip version and for frequencies higher than 200 MHz the period jitter is smaller than 20 ps (except few points). For better comparison of jitter performance two MULTI_PLL versions are shown in figure 4.13. The period jitter slightly depends on the PLL divider as shown in figure 4.14. It is worth to notice, that smaller dividers (by 6 and 8) give smaller jitter values at high frequencies (above 900 MHz).

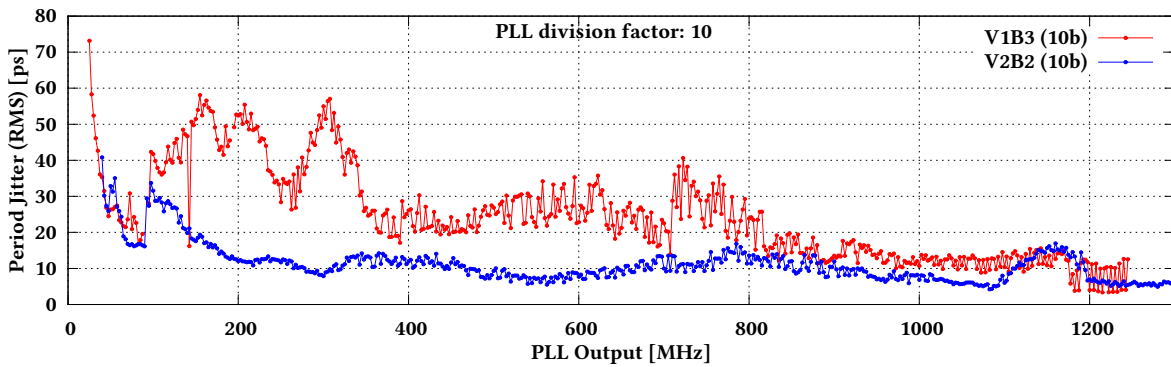


Figure 4.13: Period jitter comparison between two MULTI_PLL versions (1st and 2nd).

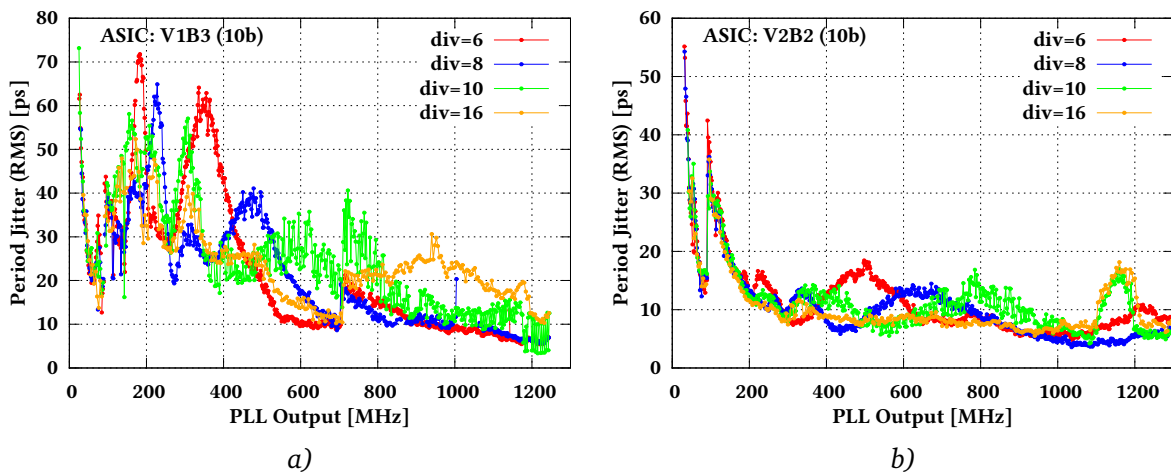


Figure 4.14: Period jitter comparison for all MULTI_PLL dividers. a) - 1st version, b) - 2nd version.

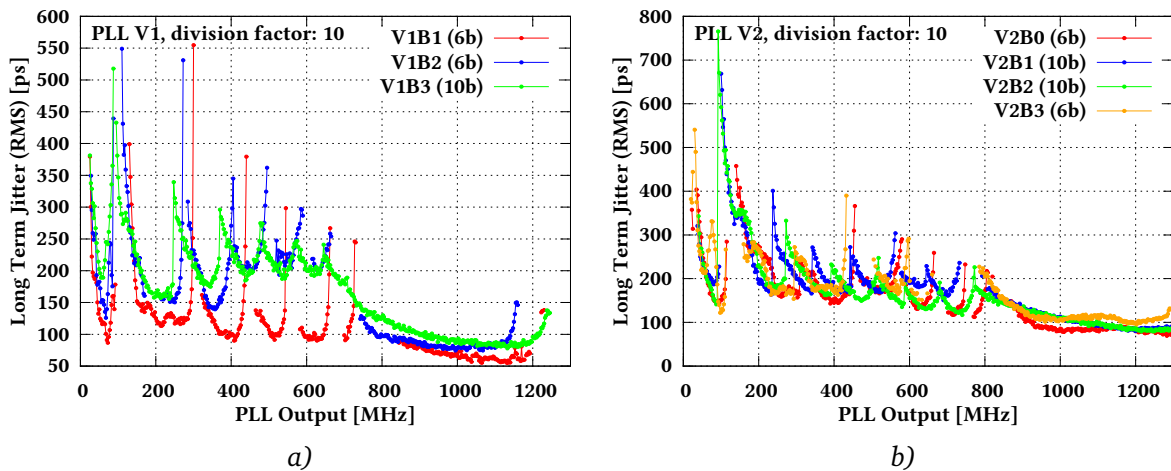


Figure 4.15: Long term jitter measurements results of the MULTI_PLL. a) - 1st version, b) - 2nd version.

Figure 4.15 shows the long term jitter measurements results for both chip versions. This parameter shows fluctuations of the output clock period in a long time (10000 clock periods in typical conditions). The MULTI_PLL_V2 is similar to the 1st chip version from the long term jitter point of view, achieving less than 400 ps for almost all measured points (there are some exceptions, especially for lower frequencies).

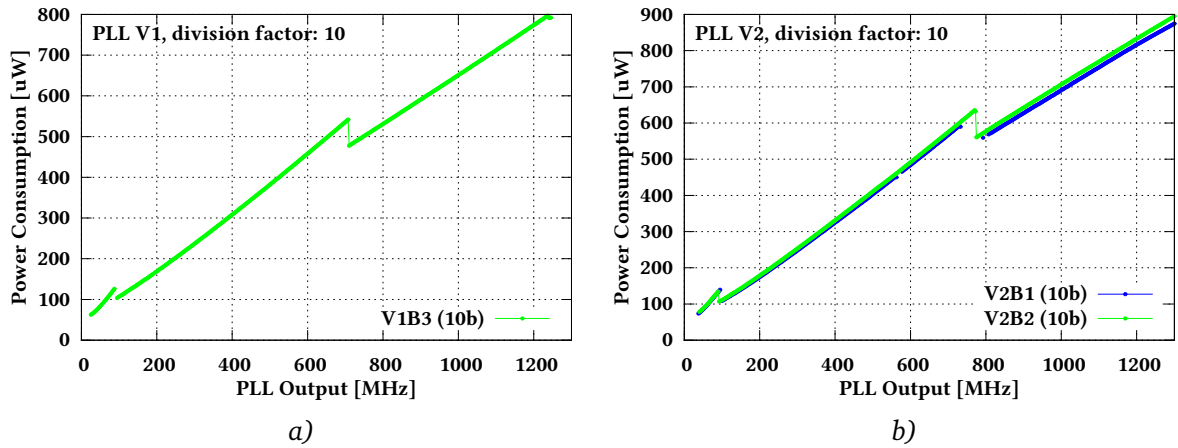


Figure 4.16: Power consumption of the MULTI_PLL. a) - 1st version, b) - 2nd version.

The power consumption was measured only for the PLLs integrated in the ADC_10 (dedicated VDD pad was available), so consequently only one power curve for the 1st chip version and two curves for the 2nd chip version are available. The power consumption is similar in both MULTI_PLL versions and gives values around 0.6 – 0.7 mW at frequency equal 1 GHz.

4.3 Measurements of SALT_PLL and SALT_DLL

For the SALT ASIC the PLL (SALT_PLL) and the DLL (SALT_DLL) functional blocks were designed. To avoid developing multiple test setups and for the better testability, the both blocks are integrated with a slow control interface and other test structures in a dedicated test ASIC, named SALT_DLL_PLL, which was fabricated in two versions (1st and 2nd). The first version was designed in the 130 nm

Parameter	SALT_PLL_V1	SALT_PLL_V2
Frequency range	30 MHz – 450 MHz	40 MHz – 400 MHz
VCO modes/gains	4	4
Division factors	2, 4, 6 or 8	2, 4, 6 or 8
Bias currents	external	external/DAC
Clock phases	16	16
Power consumption	0.8 mW @ 160 MHz	0.95 mW @ 160 MHz
Period jitter	6.7 ps @ 160 MHz	5.5 ps @ 160 MHz
Chip area	450 × 260 μm^2	300 × 210 μm^2

Table 4.7: Basic measured parameters of two SALT_PLL prototypes.

CMOS technology A, while the second in the 130 nm CMOS technology B. The two PLL prototypes have similar functionality, but in the second one a few improvements were done, like adding DAC circuits for better control of bias currents or jitter optimization. The basic parameters of two SALT_PLL prototypes are summarized in table 4.7.

Both prototypes of the SALT_PLL work in a similar frequency range and have very low power consumption. The VCO in the 1st chip version operates in 4 modes, which allows to change the gain and the center frequency of the oscillator, with fixed external bias current (gain is linked with

center frequency). The VCO in the second chip version also operates in 4 modes, but only the gain can be changed in that way. The center frequency for the 2nd VCO version is controlled directly by changing the bias current or by internal 7-bit DAC, which provides much more testing possibilities and allows to avoid the bias current fluctuations, introduced from outside. Both prototypes of the SALT_PLL work with four division factors (by 2, 4, 6 and 8) and generate 16 clock phases at their outputs. Both PLLs are also equipped with two internal multiplexers, which provide a clock phase selection (2 out of 16 phases).

Parameter	SALT_DLL_V1	SALT_DLL_V2
Frequency range	18 MHz – 62 MHz	10 MHz – 90 MHz
Clock phases	64	64
Delay between phases	355 ps – 445 ps	350 ps – 448 ps
Bias currents	external	external/DAC
Power consumption	0.7 mW @ 40 MHz	0.7 mW @ 40 MHz
Period jitter	3.8 – 7.8 ps	2.5 – 12.1 ps
Chip area	680 × 210 μm^2	430 × 190 μm^2

Table 4.8: Basic measured parameters of two SALT_DLL prototypes.

Both prototypes of the SALT_DLL work in similar frequency range and have very low power consumption. The basic parameters of both chips are presented in table 4.8. The VCDLs in both prototypes contains 64 delay cells, so the same number of independent clock phases is available at the SALT_DLL output. Both DLLs are also equipped with an internal multiplexer, which provides a clock phase selection (1 out of 64 phases). In both prototypes the VCDL delay can be controlled by changing external bias current, but the 2nd chip version provides also the change of bias current via internal 7-bit DAC, which allows to avoid current fluctuations, introduced from outside.

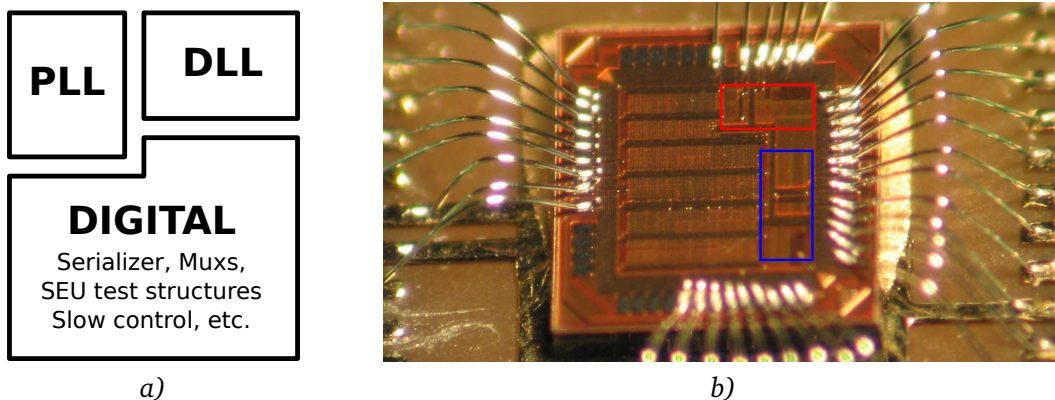


Figure 4.17: PLL/DLL blocks as a part of the SALT_DLL_PLL ASIC. a) - floorplan, b) - micrograph of the prototype chip (1st version).

Figure 4.17a presents a simplified floorplan of the SALT_DLL_PLL chips (both versions), and shows the place where the SALT_PLL and SALT_DLL blocks are fitted. The digital logic provides a slow control interface to configure the functional blocks. Figure 4.17b shows a micrograph of the SALT_DLL_PLL ASIC, the PLL (marked red) and the DLL (marked blue) cores are shown.

4.3.1 Slow control interface - ASIC configuration

The SALT_DLL_PLL ASICs work as an always selected SPI slave, which can only receive data and works in the SPI mode 0. The command shown in figure 4.18, is built with following parts: header, code and data. The MSB is always transmitted first.

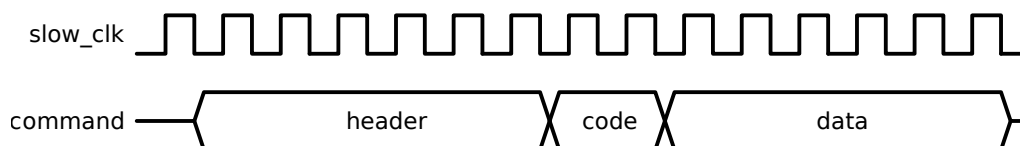


Figure 4.18: Transmission via the slow control interface - command structure.

The command header consist of two parts: 4-bit constant 1010 at the beginning and 2-bit *address*. The address can vary and there can be up to 4 ASICs with different addresses connected to one slow control interface. The address can be set using input pads.

In the SALT_DLL_PLL_V1 the command *code* field is 3-bit so up to 8 commands can be defined. From the PLL/DLL configuration point of view only three commands are useful: **pll_main_cfg**, **pll_mux_cfg** and **dll_cfg**. The data are always 8-bit long. Similarly to the MULTI_PLL configuration, described in subsection 4.2.1, the commands must be separated from each other by at least one 0 and the whole command (with header) is 18 bits long. The SALT_PLL and SALT_DLL configuration commands are presented in table 4.9.

Command name	code	data
pll_main_cfg	000	<i>pll_ena</i> (1b), n.u. (3b), <i>pll_mode</i> (2b), <i>pll_div</i> (2b)
pll_mux_cfg	001	<i>pll_mux1_sel</i> (4b), <i>pll_mux0_sel</i> (4b)
dll_cfg	010	<i>dll_ena</i> (1b), <i>dll_start</i> (1b), <i>dll_mux_sel</i> (6b)

Table 4.9: Configuration commands for the SALT_DLL_PLL_V1.

Command name	code	data
pll_mux_cfg	0011	<i>pll_mux1_sel</i> (4b), <i>pll_mux0_sel</i> (4b)
pll_main_cfg	0100	<i>pll_ena</i> (1b), n.u. (3b), <i>pll_mode</i> (2b), <i>pll_div</i> (2b)
pll_cp_cfg	0101	<i>pll_cp_ena</i> (1b), <i>pll_cp_sel</i> (7b)
pll_vco_cfg	0110	<i>pll_vco_ena</i> (1b), <i>pll_vco_sel</i> (7b)
dll_main_cfg	1000	<i>dll_ena</i> (1b), <i>dll_start</i> (1b), <i>dll_oute</i> (1b), n.u. (5b)
dll_cp_cfg	1001	<i>dll_cp_ena</i> (1b), <i>dll_cp_sel</i> (7b)
dll_vcdl_cfg	1010	<i>dll_vcdl_ena</i> (1b), <i>dll_vcdl_sel</i> (7b)
dll_mux_cfg	1011	n.u. (2b), <i>dll_mux_sel</i> (6b)

Table 4.10: Configuration commands for the SALT_DLL_PLL_V2.

In the SALT_DLL_PLL_V2 the command *code* field is 4-bit so up to 16 commands can be defined. From the PLL/DLL configuration point of view only 8 commands are useful: **pll_main_cfg**, **pll_mux_cfg**, **pll_vco_cfg**, **pll_cp_cfg**, **dll_main_cfg**, **dll_mux_cfg**, **dll_vcdl_cfg** and **dll_cp_cfg**. The same as for the 1st circuit version the data are 8-bit long, but the whole command (with header) is 1-bit longer then the command in the 1st chip version. The SALT_PLL and SALT_DLL configuration commands are presented in table 4.10.

In both versions of the SALT_PLL the **pll_main_cfg** is similar and allows to enable/disable the PLL, change the modes and division factors of the Phase-Locked Loop. The SALT_PLL is enabled when the field *pll_ena* of this command is set to 1. The loop divider can be changed by setting the *pll_div* field (2 bits), according to table 4.11. The field *pll_mode* of the **pll_main_cfg** command in the SALT_PLL_V1 chip allows to configure the VCO center frequency (according to table 4.12) and the gain, which is linked to the selected mode. In the SALT_PLL_V2 chip the *pll_mode* field changes only the VCO gain, because the center frequency can be easily configured by changing the VCO bias current via internal DAC.

pll_div	PLL division factor
00	2
01	4
10	6
11	8

Table 4.11: SALT_PLL divider configuration set by the **pll_main_cfg** command.

Both SALT_PLL versions generate 16 independent clock phases, but only two of them can be used at the same time. The output clocks can be chosen by two multiplexers, using the **pll_mux_cfg** command. First of them is controlled by setting the value of the *pll_mux0_sel* field. Changing the value of this field by 1, shifts the clock phase by 1/16 of its period. The second multiplexer operates in the same way and is controlled by setting the value of the *pll_mux1_sel* field.

pll_mode	VCO center frequency
00	80 MHz
01	160 MHz
10	240 MHz
11	320 MHz

Table 4.12: SALT_PLL mode configuration set by the **pll_main_cfg** command.

The CP and VCO bias currents in the SALT_PLL_V1 are controlled by external circuits, placed on the ASIC test board. The 2nd chip versions are equipped with internal DACs, which provide bias current configuration without external components. The VCO bias DAC can be controlled by the **pll_vco_cfg** command. When the *pll_vco_ena* field of this command is set to 1, the DAC is enabled and the Voltage-Controlled Oscillator bias current can be changed by the value of the *pll_vco_sel* field. The Charge Pump bias current can be configured in the same way by the **pll_cp_cfg** command. When the internal DAC/DACs is/are disabled, the bias current is controlled by external source in the same way as for the SALT_PLL_V1. More details about the SALT_PLL can be found in section 3.2.

In the 1st chip versions of the SALT_DLL the chip configuration is very simple and consist of only one command **dll_cfg**, which allows to enable/disable the DLL and changes the configuration of the output multiplexer. Turning on the SALT_DLL_V1 requires two steps, which can be done by setting the values of the *dll_ena* and *dll_start* fields of the **dll_cfg** command. When the *dll_ena* is set to 0 (the value of *dll_start* does not matter) the SALT_DLL is disabled. In the first step (standby mode) the *dll_ena* field should be set to 1, while the *dll_start* should stay in state 0. After 10 μ s (or more) the value of *dll_start* field should be set to 1 to start synchronization of the DLL. The SALT_DLL_V1 generates 64 independent clock phases, but only one of them can be used at a time (selected by the

multiplexer). The **dll_cfg** command contains also the *dll_mux_sel* field, which controls the output multiplexer. Changing the value of this field by 1 shifts the clock phase by 1/64 of its period.

The SALT_DLL_V2 has slightly different configuration than the 1st version of the chip. The **dll_main_cfg** command contains only three fields: *dll_ena*, *dll_start* and *dll_oute*. The effect of first two fields is the same as for the SALT_DLL_V1, but the last one provides an extra functionality. When the *dll_oute* field is set to 0, the input clock is connected directly to the output (bypass). The selection of the clock phase is possible only when the *dll_oute* is set to 1. This functionality is created to avoid random changes of the clock phase on the SALT_DLL output during the synchronization process, which may be disadvantageous for the digital logic. The output multiplexer works in the same way as in the SALT_DLL_V1, but changing the clock phase should be done by a dedicated **dll_mux_cfg** command.

The **CP** and **VCDL** bias currents in the SALT_DLL_V1 are controlled by external circuits placed on the **ASIC** test board. The 2nd chip versions are equipped with internal **DACs**, which provide bias current configuration without external components. The **VCDL** bias **DAC** can be controlled by the **dll_vcdl_cfg** command, while the **CP** bias current can be changed by the **dll_cp_cfg** command. More details about the SALT_DLL can be found in section 3.3.

4.3.2 Details of measurement setup

As previously mentioned the prototype **PCB** is needed to provide electrical connections and some components supporting the prototype **ASIC**. Figure 4.19 shows a simplified schematic of the prototype **PCB**. The figure contains only the components needed for the SALT_PLL and SALT_DLL measurements, all other elements are omitted.

The **ASIC** board contains voltage regulators, which allow to adjust all voltages biasing the chip. In result, it is possible to independently regulate the value of the power supply voltages in few separate channels. This provides a separation between the SALT_PLL, SALT_DLL, and other digital circuits. Each of the voltage channels is equipped with a current monitor and a 16-bit **ADC**, which can measure the power consumption and power supply voltage, so the external current (voltage) meters are not necessary. The output voltage is set by a 12-bit **DAC**, so a voltage adjustment can be done by software, not by the potentiometer. The board contains also a large number of decoupling capacitors, but they are not presented in diagram 4.19.

The test board contains also circuits, which generate bias currents. The currents can be digitally controlled by a 12-bit **DAC** in several separated channels. Each of them is equipped with a voltage to current converter (operational amplifier + transistor). The current adjustment can be done by software, so potentiometers are not longer needed. The **ASIC** board contains also the circuits generating bias voltages and bias currents, which are adjusted very rarely. In this case the biasing circuits take form of potentiometers and voltage dividers, but it is also possible to connect an external precision voltage source or a current source.

The test board for the SALT_DLL_PLL **ASIC** is much more complicated than the board presented in section 4.2. A large number of digital signals, the chip configuration interface (slow control) and a large number of **DACs** and **ADCs** placed on the board, require a microcontroller or other programmable device. The **DACs** and **ADCs** are controlled via I²C bus, the **SPI** is used as a slow control interface for the chip configuration, and the microcontroller Universal Asynchronous Receiver/Transmitter (**UART**) is used for communication with the computer. The communication with the computer software is ensured by the **USB** interface, but for simplicity of the microcontroller software, the **USB** to **UART** converter is used. The **ASIC** board contains also the Liquid Crystal Display (**LCD**), which can be used for presentation of the measured key parameters.

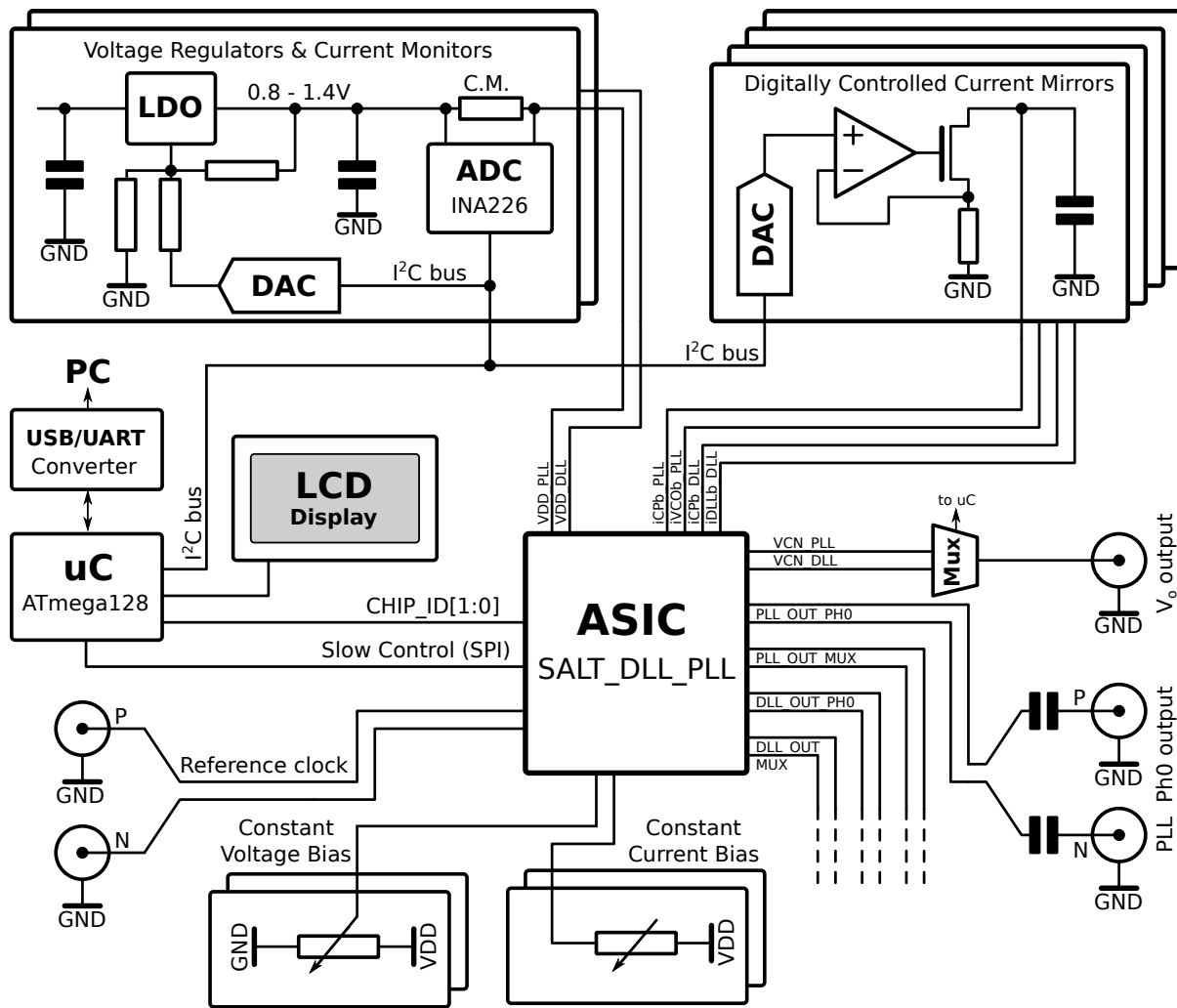


Figure 4.19: Simplified schematic diagram of the prototype PCB (ASIC board).

The reference signal is fed to the differential clock input of the prototype ASIC via the **SIVS** buffer. The chip outputs are made also in **SIVS** standard. There are no additional clock buffers on the test board and the ASIC outputs are connected directly to the connectors on the board. For the proper **SIVS** operation two serial capacitors for each of the outputs are used, which allow to obtain **AC** coupling at the output.

The prototype PCB allows to connect the measured ASIC to the outside world and contains important components, used for chip powering, biasing and buffering I/O signals. The measurements of the prototype PLL/DLL are not possible without laboratory instruments like: generators, oscilloscopes etc., which can generate input signals and measure the key parameters of the chip. Figure 4.20 shows a block diagram of the SALT_DLL_PLL measurement setup, and figure 4.21 shows its photograph.

To power the PCB an external laboratory power supply (Agilent E3630A) is used, which is connected directly to the voltage regulators on the board. A voltage and current measurements can be done by the ADCs placed on the board and/or digital multimeter (Agilent 34401A), which can be connected to the board.

The reference input clock is given by the differential generator (Agilent 81160A) and its frequency can vary in range 1 – 220 MHz. All frequency and jitter measurements of the SALT_PLL

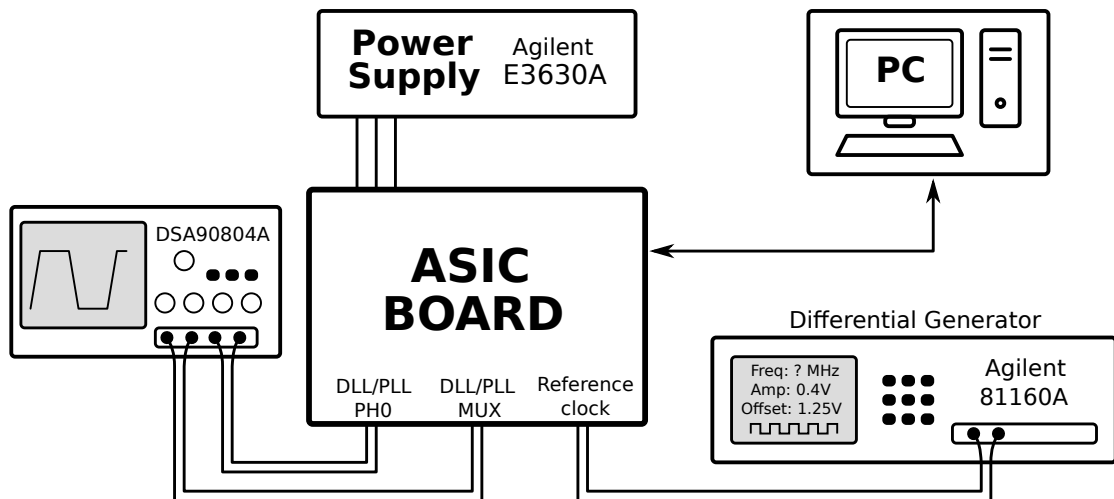


Figure 4.20: Block diagram of the SALT_DLL_PLL measurements setup.

output signals are done using the 40 GSps oscilloscope (Agilent DSA90804A). The SALT_DLL measurements require a second oscilloscope channel to measure the value of the time shifts between the output signals. To improve automatization of the measurements all laboratory equipment is controlled by a computer, using [USB](#) and [LAN](#) interfaces. To perform the data analysis and to calculate the parameters, which characterize the measured ASIC, a dedicated software was developed.

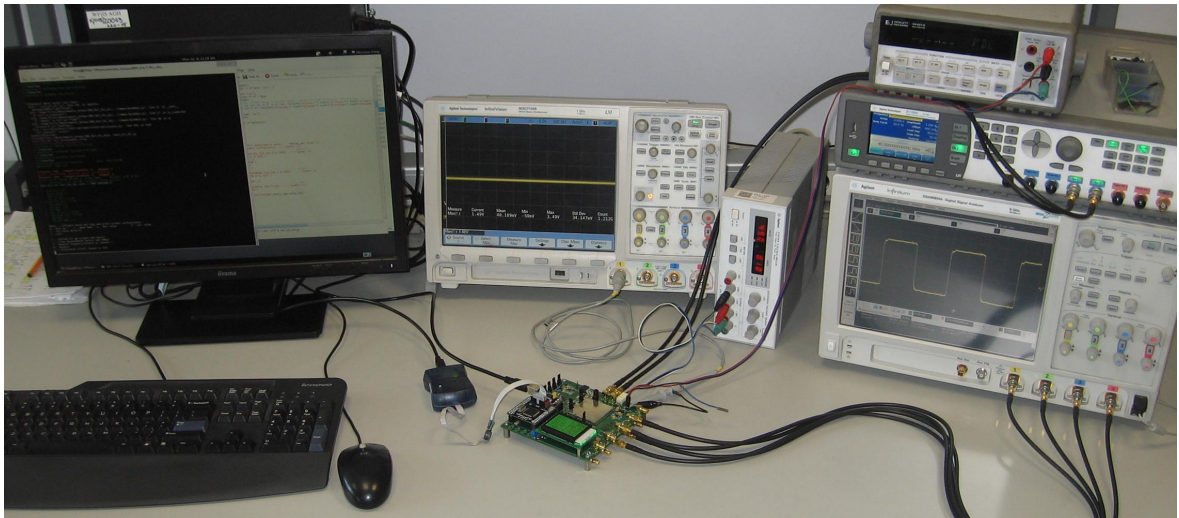


Figure 4.21: Photograph of the measurement setup.

The measurement software has a form of a dedicated python script, which allows to:

- Communicate with the microcontroller via serial interface ([USB](#) to serial converter is used). The ASIC configuration can be changed using a slow control interface, implemented in the microcontroller. Setting bias voltages/currents and reading voltage/current measurements results can be also done via serial interface;
- Set differential waveform generator parameters like: frequency, amplitude, offset, etc. via [LAN](#) interface;
- Read the digital data samples (via [LAN](#) interface) containing the measured signal from the oscilloscope and calculate the frequency, period and all jitter types based on this data;

- Perform automatic frequency scans with selected range and step. It is possible to define multiple scans with different ASIC configurations, which start one by one;
- Create many different types of charts.

4.3.3 Measurements results of SALT_PLL

The results presented in this subsection are based on the measurements of four SALT_PLL chips (two 1st and two 2nd chip versions). Each of them is integrated in the SALT_DLL_PLL ASIC - integration of both versions is very similar. This subsection presents only few examples of measurements results, the rest are available in appendix B.

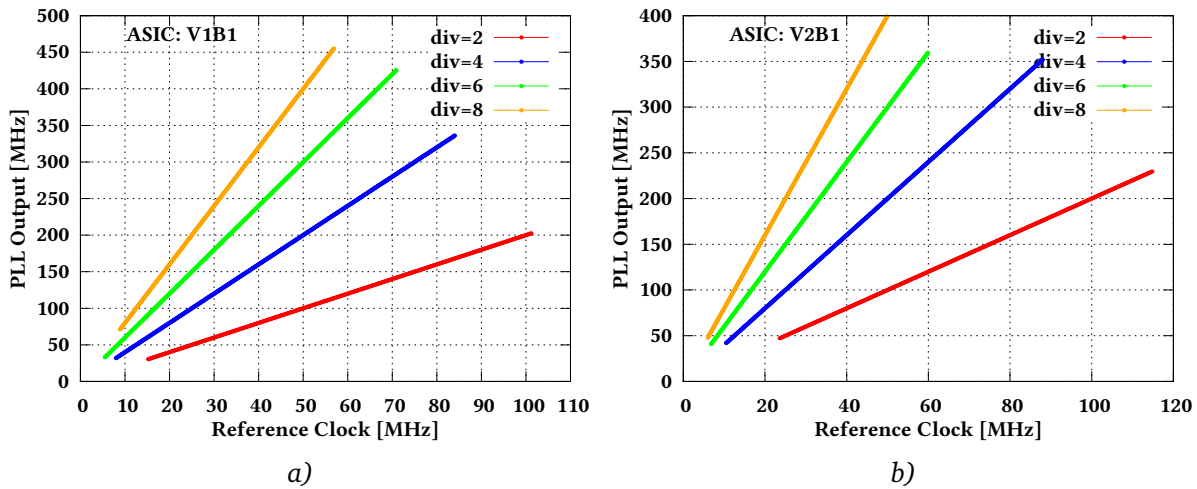


Figure 4.22: Gain measurements results of the SALT_PLL. a) - 1st chip version, b) - 2nd chip version.

Figure 4.22 shows the gain plots for both versions of the SALT_PLL. Each of them presents four curves for dividers by: 2, 4, 6, and 8 (4 is default divider) and four VCO gains related to the selected dividers. The circuit in the 1st version works up to 450 MHz, while the 2nd circuit version can operate up to 400 MHz. The measurements for both PLL versions and all division factors are presented in figures B.1 and B.2 in appendix B. The circuit was optimized to work with reference clock 40 MHz and gives output frequencies: 80 MHz, 160 MHz, 240 MHz, and 320 MHz, depending on the selected divider. The VCO bias current was changed in range $1 \mu\text{A} - 50 \mu\text{A}$ to optimize the SALT_PLL operation conditions.

The VCO circuits in both SALT_PLL versions generate multiple clock phases. Figure 4.23 shows the period jitter performance at default output frequency 160 MHz as a function of the selected clock phase. Figure 4.24 presents the total clock delay, dependent on the selected clock phase, for both circuit versions. The delay in both circuit versions can be changed linearly with increasing clock phase number, but the 1st circuit version has one point when the linearity is broken, caused by the single-ended VCO construction. The time offset around 400 ps (600 ps for 2nd version), present for phase 0 is related to the time propagation of the output multiplexer.

The period jitter measurements results of the SALT_PLL, operating in default conditions, are presented in figure 4.25, which shows the plots for both circuit versions. The jitter value of the 1st version in the whole presented frequency range is better than 60 ps, but in the default frequency range 150 MHz – 220 MHz, where the circuit was optimized, the period jitter is below 7.5 ps. The measurements of the SALT_PLL_V2 show similar behavior for lower frequencies, but the low jitter, below 10 ps, is measured in much wider frequency range 100 MHz – 350 MHz.

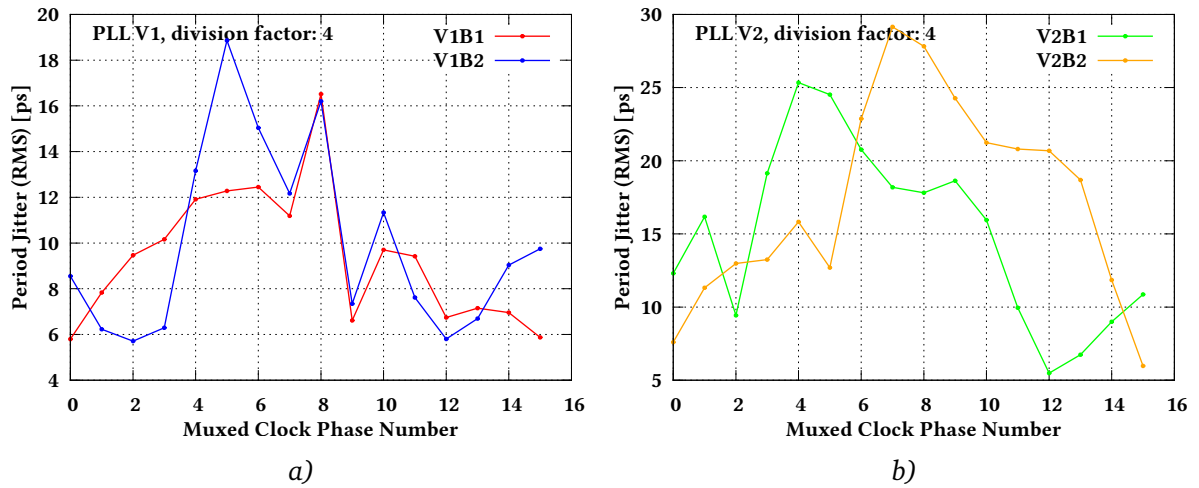


Figure 4.23: Period jitter of the SALT_PLL as a function of selected clock phase. a) - 1st chip version, b) - 2nd chip version.

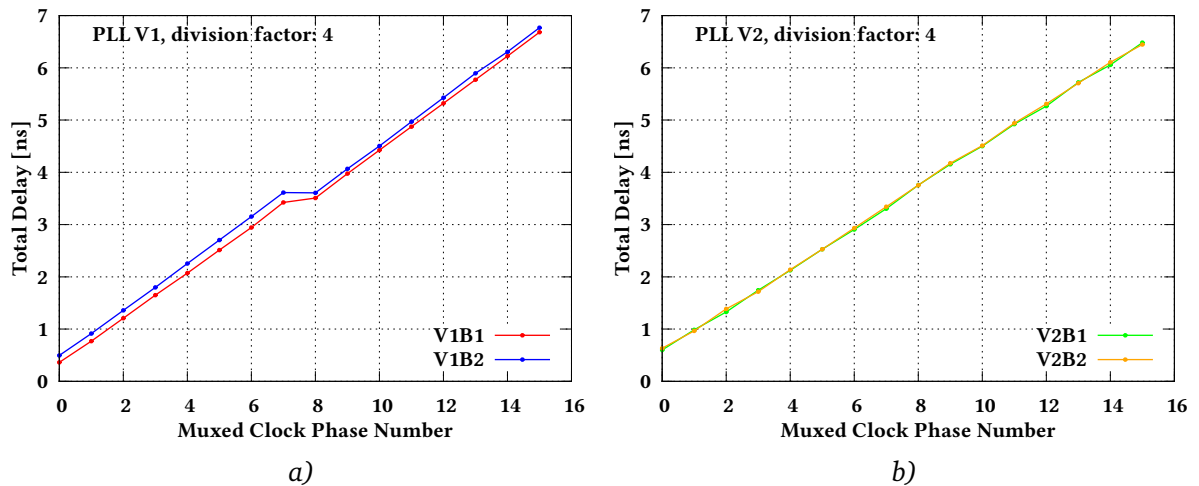


Figure 4.24: Total clock delay of the SALT_PLL as a function of selected clock phase. a) - 1st chip version, b) - 2nd chip version.

The period jitter comparison between two PLL versions is shown in figure 4.26. The performance of the 2nd version is better than the 1st version, which is related to the improvements in the VCO construction. The differential inverters used in the second VCO version allow to reduce the jitter to values below 10 ps for very wide frequency range and make that the jitter becomes more predictable. For frequencies near 160 MHz, important from the SALT point of view, the period jitter is better than 5 ps.

Figure 4.27 shows the period jitter comparison for all available PLL dividers. The measurements show that the period jitter depends not only on the clock frequency, but also on the selected PLL divider, which is directly related to the LPF, optimized for divider 4. In frequency range 150 MHz – 220 MHz the SALT_PLL has better jitter performance for lower division factors (2 and 4). The jitter of the second circuit version is much more stable and the differences between dividers at the same frequency are very small.

The power consumption measurements results for both PLL versions are presented in figure 4.28. In both cases the power scales linearly with output frequency and depends slightly on the selected PLL division factor. The 1st circuit version consumes around 0.8 mW at typical power supply 1.2 V and

default output frequency 160 MHz. The SALT_PLL_V2 at the same conditions consumes 0.95 mW, which is needed for improved VCO construction. The higher power consumption (around 30%) is acceptable, because the 2nd circuit version gives better jitter performance in wider frequency range than the 1st PLL version.

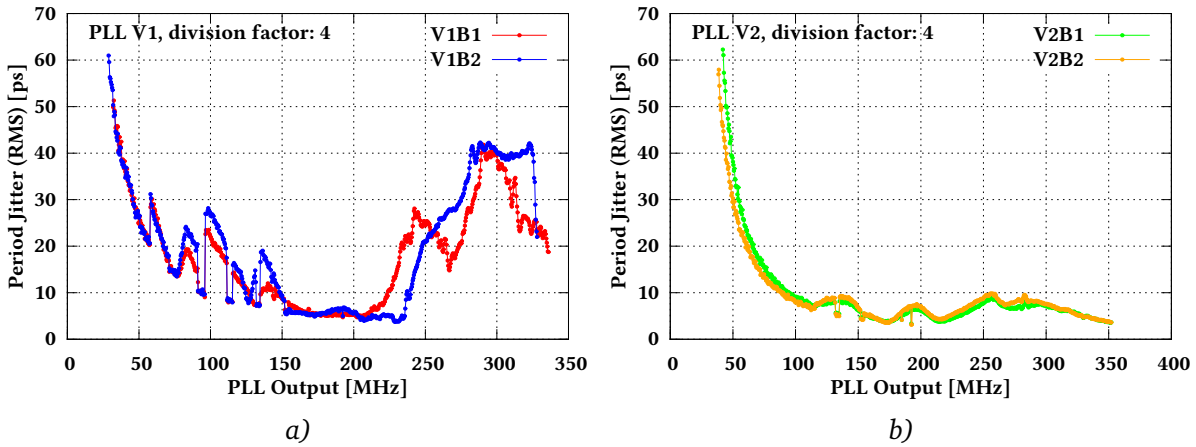


Figure 4.25: Period jitter measurements results of the SALT_PLL. a) - 1st version, b) - 2nd version.

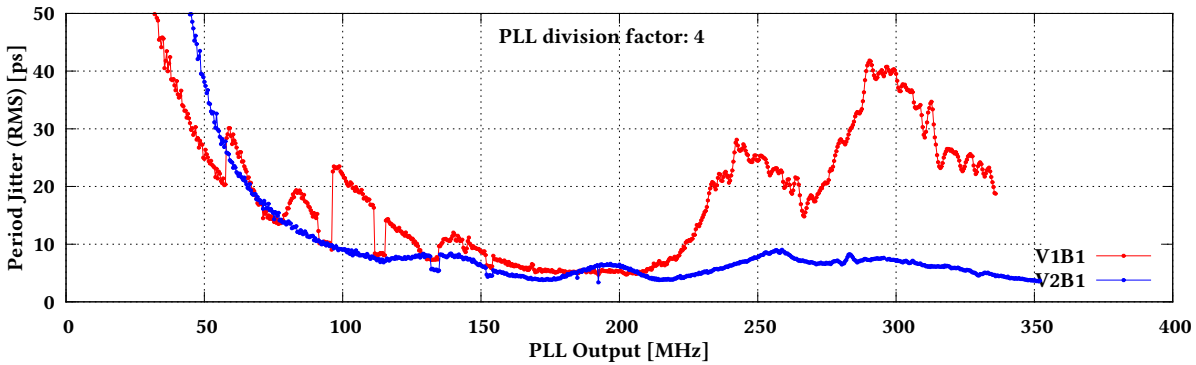


Figure 4.26: Period jitter comparison between two SALT_PLL versions (1st and 2nd).

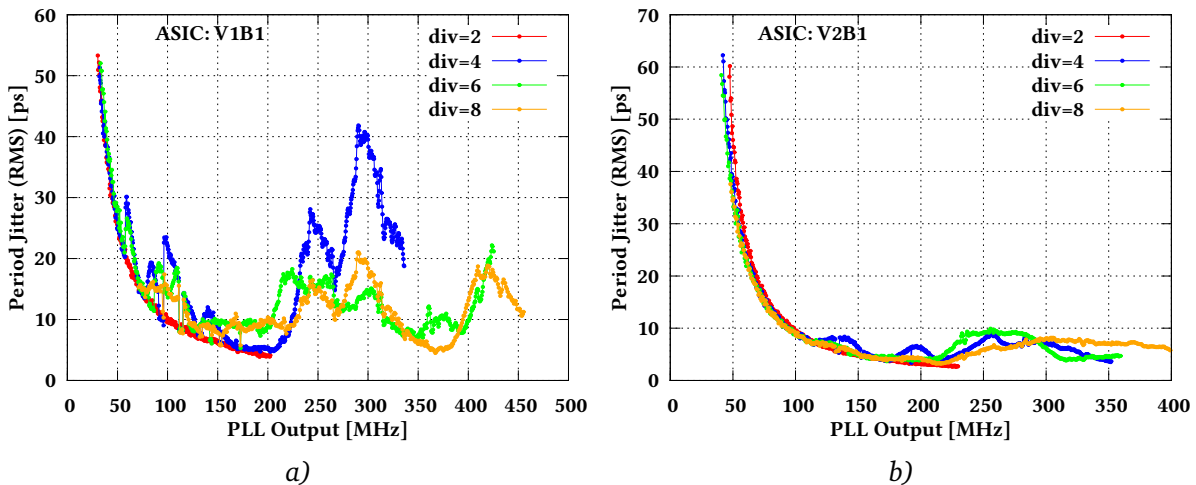


Figure 4.27: Period jitter comparison for all SALT_PLL division factors. a) - for the 1st chip version, b) - for the 2nd chip version.

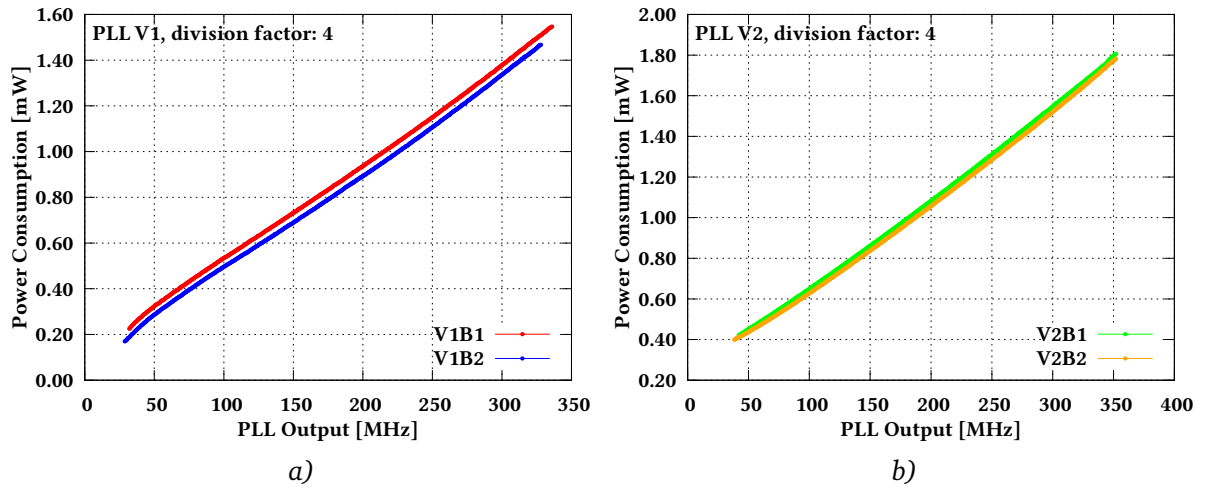


Figure 4.28: Power consumption of the SALT_PLL. *a)* - 1st chip version, *b)* - 2nd chip version.

4.3.4 Measurements results of SALT_DLL

The results presented here are based on the measurements of four SALT_DLL chips (two 1st and two 2nd chip versions). Each of them is integrated in the SALT_DLL_PLL ASIC - integration of both versions is very similar. This subsection presents only few examples of measurements results, the rest are available in appendix C.

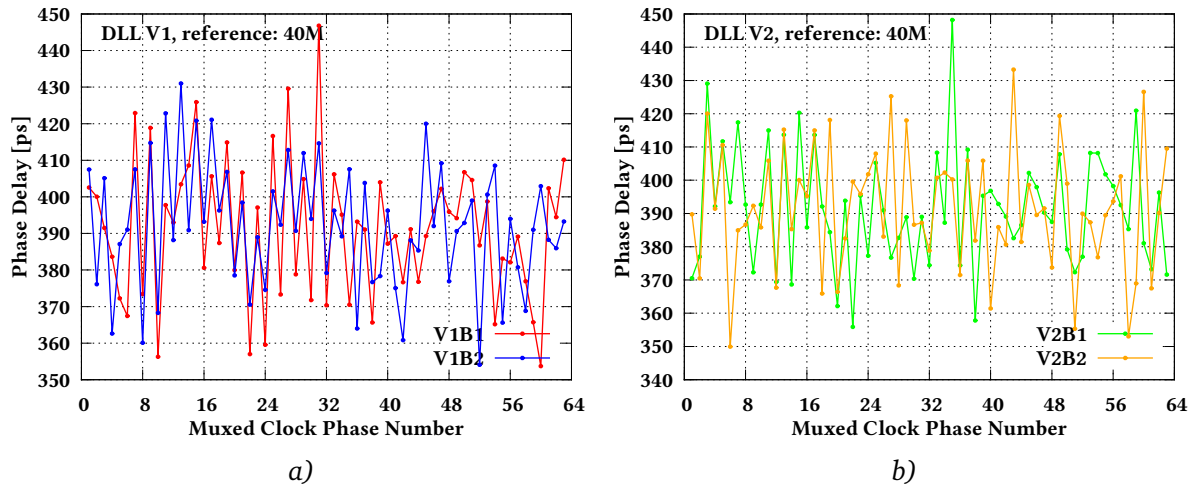


Figure 4.29: Phase delay of the SALT_DLL as a function of selected clock phase. *a)* - 1st chip version, *b)* - 2nd chip version.

The SALT_DLL is used for the clock phase adjustment in the SALT chip, so the time delay between clock phases is very important. The phase delay of the DLL as a function of the selected clock phase (for both circuit versions) is presented in figure 4.29. The time delay between clock phases at default frequency 40 MHz should be around 395 ps. The measurements of SALT_DLL_V1 give the phase delay in range 355 ps – 445 ps. The 2nd circuit version gives almost the same results.

Figure 4.30 shows the total clock delay as a function of the selected DLL clock phase. The measurements of both circuit versions show very good linearity. The time measurements were done in reference to clock phase 0, which is available at the dedicated DLL output. The time offset for phase 0, shown in the figure is related to the propagation time of the internal multiplexer.

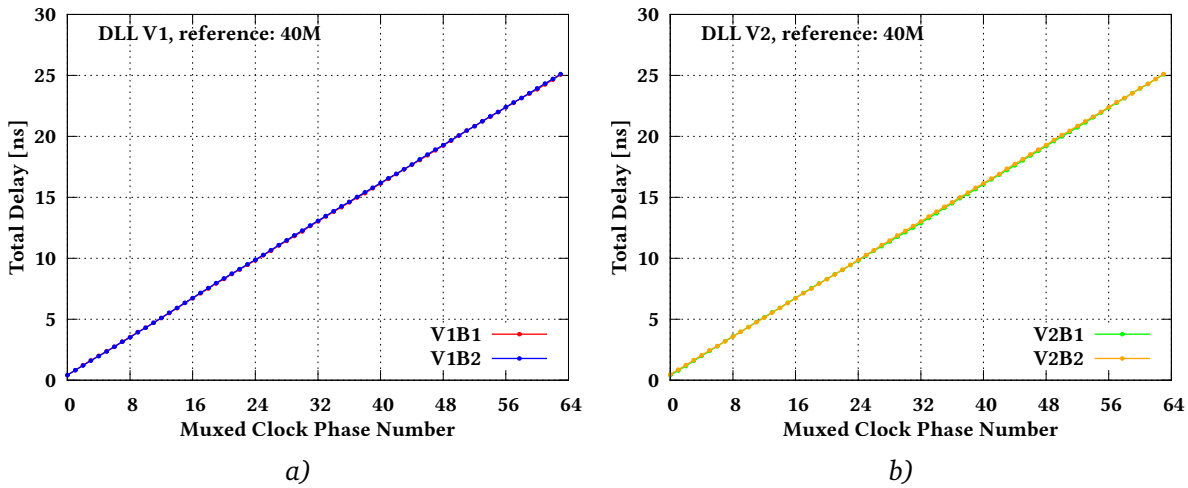


Figure 4.30: Total clock delay of the SALT_DLL as a function of selected clock phase. *a)* - 1st chip version, *b)* - 2nd chip version.

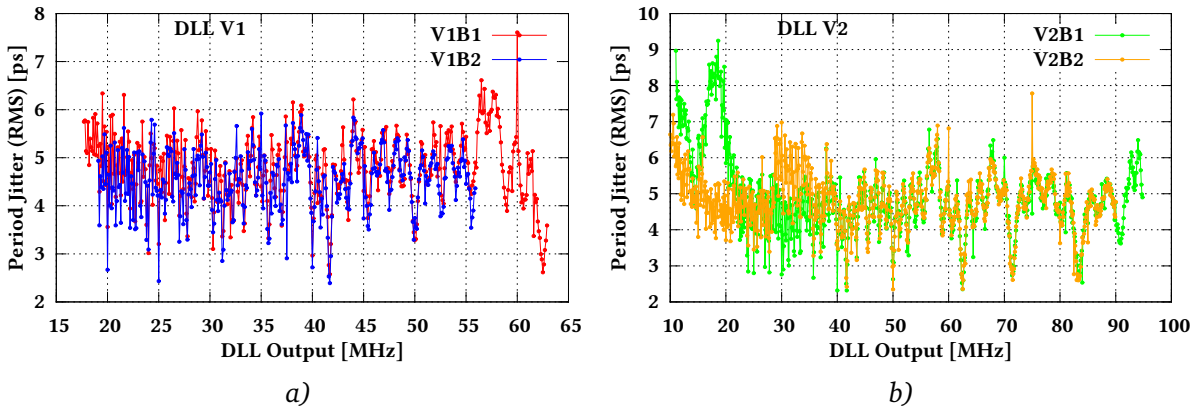


Figure 4.31: Period jitter measurements results of the SALT_DLL. *a)* - 1st version, *b)* - 2nd version.

The period jitter measurements results of the SALT_DLL are presented in figure 4.31, which shows the plots for both circuit versions. The jitter value of the 1st version in the whole frequency range is better than 7 ps, but for the default input frequency 40 MHz, where the circuit was optimized, the period jitter is below 5 ps. The measurements of the SALT_DLL_V2 show very similar results but the 2nd circuit can operate at wider frequency range 10 MHz – 90 MHz. The VCDL bias current (in both chip versions) was changed in range $1 \mu\text{A} - 50 \mu\text{A}$ to optimize the SALT_DLL operation conditions and to extend its frequency range.

The comparison between two chip versions, presented in figure 4.32, shows that both circuits achieve similar jitter performance in the frequency range 20 MHz – 60 MHz. The SALT_DLL construction in both versions is exactly the same so this comparison is very good to show the differences between two CMOS technologies, used to design the two DLL versions.

The period jitter of both SALT_DLL versions as a function of the selected clock phase is presented in figure 4.33. The DLL generates multiple clock phases, which are selected by the multiplexer. The period jitter of the 1st circuit version grows linearly with the clock phase number and takes values from 3.8 ps to 7.8 ps at default frequency 40 MHz. This effect is related with VCDL, which consists of inverters connected in series. Each of them introduces some additive jitter. The second circuit version at the same conditions gives the period jitter in range 2.5 ps – 12.1 ps.

The power consumption for both DLL versions is presented in figure 4.34 and scales linearly

with frequency. The 1st version consumes around 0.7 mW at typical power supply 1.2 V and default frequency 40 MHz. The SALT_DLL_V2 at the same conditions consumes the same amount of power.

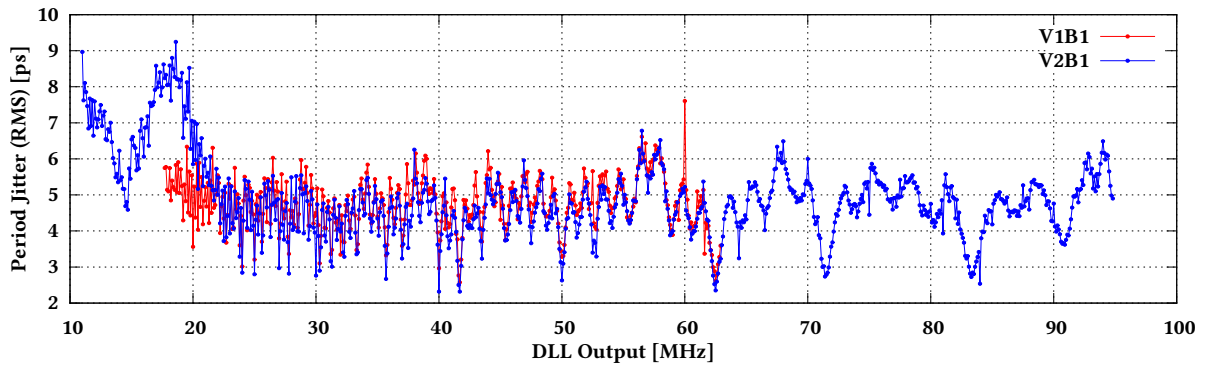


Figure 4.32: Period jitter comparison between two SALT_DLL versions (1st and 2nd).

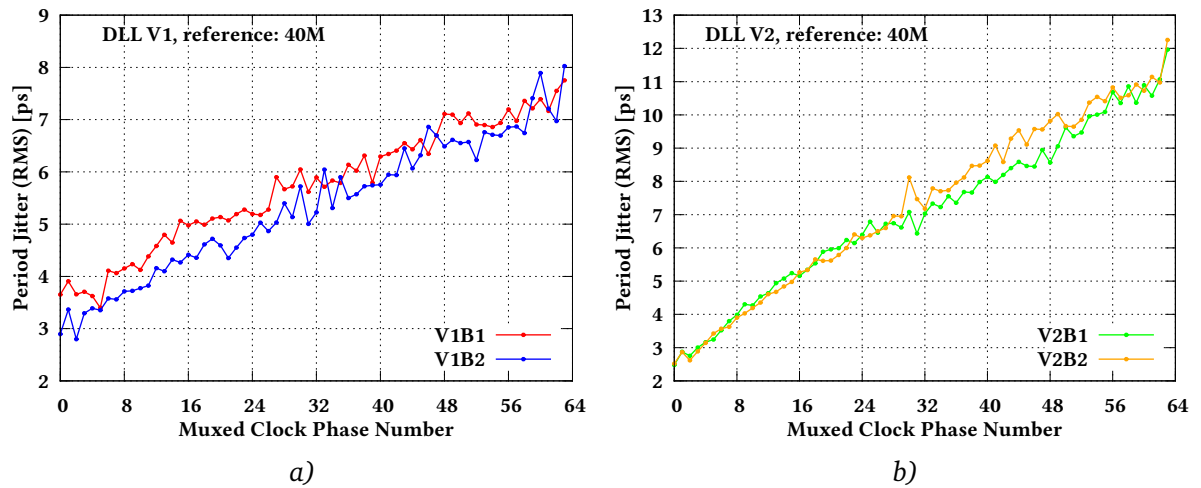


Figure 4.33: Period jitter of the SALT_DLL as a function of selected clock phase. a) - 1st chip version, b) - 2nd chip version.

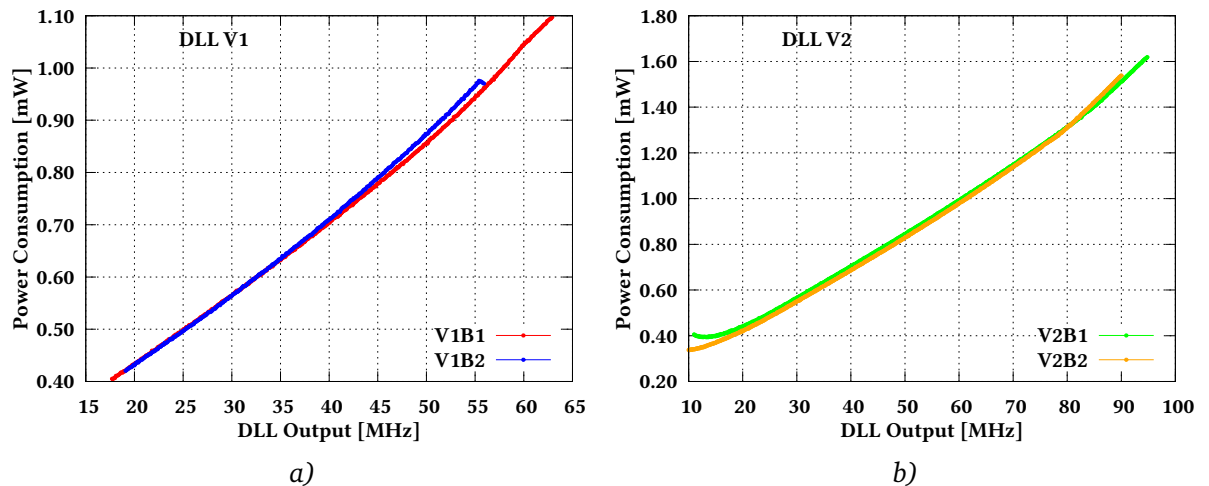


Figure 4.34: Power consumption of the SALT_DLL. a) - 1st chip version, b) - 2nd chip version.

Summary

The main objective of this dissertation was the development of [SALT](#) readout for the [UT](#) detector. In such complex system many people are involved in the design process and measurements of the prototype [ASICs](#). The [SALT](#) is designed by AGH-UST group from the Department of Particle Interactions and Detection Techniques. The author is responsible for the design and measurements of the [PLL](#) and [DLL](#) circuits for fast clock generation, phase alignment, and data serialization, but he participates also in development of the other [SALT](#) functional blocks (eg. analog front-end).

The dissertation consists of four chapters. At the beginning a short general description of the [LHCb](#) experiment is presented. It shows how complicated the upgrade of the whole detector system is and presents various challenges facing the designers. The contribution of AGH-UST group, responsible for the [SALT](#) readout, is presented. Theoretical analysis of the [PLL](#) and [DLL](#) circuits is presented in the second chapter, providing basic knowledge needed for design and simulation described in chapter three. Finally, in chapter four the measurements results on the prototype [ASICs](#) are presented. The details of the measurements setups and dedicated software, which controls the laboratory equipment and collects the data were also given in the last chapter.

All the objectives of the dissertation have been fully attained. The author has been taking part in the development of one of the most complex readout system for the [HEP](#) applications, the [SALT](#) readout for the [UT](#) detector. As already mentioned he has participated in all project phases, starting from the [SALT](#) general concept matching the [UT](#) specifications, elaborating the ideas into the realistic [SALT](#) architecture, and then designing the prototypes, preparing the test setups, performing the measurements, and analyzing the data. Due to the limited space only the works on the main author contributions, i.e. the [PLL](#) and [DLL](#) circuits are described in detail in this dissertation, although the author participated also in development of other parts of [SALT](#) (e.g. analog front-end). In parallel author was developing the key [MULTI_PLL](#) block to allow a much faster serialization and data transmission than the one implemented in the [SALT](#). Regarding the [PLL](#) and [DLL](#) blocks, several prototype [ASICs](#) were designed and tested. All of them are fully functional and achieve very good parameters. The 1st versions of [SALT_PLL](#) and [SALT_DLL](#) were designed in 130 nm [CMOS](#) technology A, while the 2nd versions were designed in 130 nm [CMOS](#) technology B. This change was motivated by the decision of [LHCb](#) collaboration. The both [MULTI_PLL](#) prototypes were designed in the same technology (130 nm [CMOS](#) technology A) and the improvement of circuit parameters done in the 2nd chip version was confirmed by the measurements.

The two [MULTI_PLL](#) prototypes, designed as a general purpose blocks, were presented in section 3.1, while the first of them was published in [96]. Two prototypes have the same functionality but in the second one the clock jitter is optimized and the frequency ranges are slightly extended. Its measurements results can be found in section 4.2. The circuit was designed and integrated in the serializer of the multi-channel 6-bit [ADC](#) prototype for [SALT](#) and in the multi-channel 10-bit [ADC](#) for the luminosity detector at [ILC](#). As the post-layout simulations show, the circuit can operate in a very wide frequency range, from few MHz up to above 3 GHz, but the measurements can verify the proper [MULTI_PLL](#) operation only up to 1.3 GHz, since the [SLVS](#) output buffer limits the maximum

frequency. Obtaining such a wide frequency range is not possible without a special VCO construction, proposed by author and described in subsection 3.1.1. The MULTI_PLL provides also a configurable clock divider in the feedback, which is used to obtain different frequency multiplication factors. The division factor can be selected between: 6, 8, 10, and 16. The automatic selection of the VCO frequency range is provided by the AFMS circuit, described in subsection 3.1.5, which architecture was proposed by the author. It is not a standard PLL block, but it is useful to improve the circuit functionality. The measurements of the MULTI_PLL, summarized in subsection 4.2.3 show that the 2nd chip version achieves better jitter performance than the previous chip version and for frequencies higher than 200 MHz the period jitter is smaller than 20 ps (except few points). The power consumption is very similar in both MULTI_PLL versions and gives extremely low values around 0.6 – 0.7 mW at frequency equal 1 GHz.

The SALT_PLL, described in section 3.2, was designed for the SALT readout chip as dedicated multi-phase Phase-Locked Loop. It is used as a clock multiplier and phase shifter for the serializer and deserializer blocks. The PLL was designed in two versions and in the second one a few improvements were done, like adding the DAC circuits for better control of bias currents and jitter optimization. Its measurements results can be found in subsection 4.3.3. The circuit (both versions) operates in frequency range from around 70 MHz up to around 400 MHz, but 160 MHz is a default frequency for the data transfer in the SALT readout. The Voltage-Controlled Oscillator in SALT_PLL can generate 16 clock phases. The VCO provides gain/mode selection (1 of 4), which can be useful for jitter optimization and for change of working frequency. Both prototypes of the circuit work with four division factors (by 2, 4, 6 and 8) and generate 16 clock phases at their outputs. Both PLLs are also equipped with two internal multiplexers, providing clock phase selection (2 out of 16 phases). The circuit was optimized to work with reference clock 40 MHz, which gives output frequencies: 80 MHz, 160 MHz, 240 MHz and 320 MHz, depending on the selected divider. The period jitter of the 2nd circuit version is better then the 1st version, which is related to the improvements in the VCO construction. The differential inverters used in the second VCO version allow to reduce the jitter to values below 10 ps for very wide frequency range and make that the jitter becomes more predictable. For the requested frequency 160 MHz the period jitter is better than 5 ps. The power consumption scales linearly with output frequency and slightly depends on the selected PLL division factor. The 1st circuit version consumes around 0.8 mW at typical power supply 1.2 V and default output frequency 160 MHz. The 2nd circuit version at the same conditions consumes 0.95 mW, which is needed for the improved VCO construction.

Source	[97]	[98]	[99]	[75]	[100]	SALT_PLL
CMOS [nm]	130	130	65	130	130	130
Frequency [MHz]	30–650	200–950	60–1489	320	10–700	40–400
Divider	1–4096	4–160	4–131	8	–	2, 4, 6, 8
Jitter (RMS) [ps]	4.9	8.0	8.03	10.0	24.3	5.5
Power [mW@MHz]	7@240	3@640	4.3@855	29@320	7@200	0.95@160
Supply Voltage [V]	1.5	1.0	1.2	1.5	1.0	1.2
Area [mm ²]	0.18	0.063	0.0182*	-	0.1681	0.063

*without filter capacitance

Table 4.13: SALT_PLL performance comparison with other designs.

Table 4.13 shows the comparison of the SALT_PLL performance with the State-of-Art designs, published in [97, 98, 99, 75, 100], which were chosen to have the specifications as close as possible to the PLL designed by the author. The SALT_PLL achieves the lowest power consumption and

almost the best jitter performance. This is mainly due to the manually designed layout, without using Encounter Cadence tools. For the same reason, the SALT_PLL uses the area very efficiently.

The SALT_DLL, described in section 3.3, was designed as dedicated SALT block, needed to adjust the phase of the input clock. In particular it is needed to align the phase of the ADC sampling clock. The DLL was designed in two versions. The first one operates at frequency range from 20 MHz up to 60 MHz, while the second works in range 10 MHz – 90 MHz. Both circuit versions are optimized to work at 40 MHz. In both DLL prototypes the VCDL delay can be controlled by changing external bias current, but the 2nd chip version provides also the change of bias current via internal 7-bit DAC, which allows to avoid current fluctuations, introduced from outside. The VCDL is the main component of the DLL and can generate 64 independent clock phases. Both DLLs are also equipped with internal multiplexer, which provides a clock phase selection (1 out of 64 phases). The SALT_DLL is used for clock phase adjustment in the SALT chip, so the time delay between the clock phases is very important and at default frequency 40 MHz should be around 395 ps. The time delay measurements of both circuit versions show almost the same results and give the phase delay in range 355 ps – 445 ps. The period jitter measurements results of the SALT_DLL, presented in subsection 4.3.3, show that the jitter value of the 1st circuit version in the whole presented frequency range is better than 7 ps, but for the default input frequency 40 MHz, where the circuit was optimized, the period jitter is below 5 ps. The measurements of the 2nd chip version show very similar results but it can be operated in wider frequency range. The comparison between two chip versions shows that both circuits achieve similar jitter performance in the frequency range 20 MHz – 60 MHz. The power consumption of both DLL versions scales linearly with frequency. The both circuit versions consume around 0.7 mW at typical power supply 1.2 V and default frequency 40 MHz.

Source	[101]	[91]	[102]	[103]	[104]	SALT_DLL
CMOS [nm]	350	250	160	350	130	130
Frequency [MHz]	6–130	32–320	42–400	62.5–250	15–600	10–90
VCDL Phases	10	10	–	8	12	64
Jitter (RMS) [ps]	3.3–24.8	2.5–6.15	4.7	4.0	9–116	2.5–12.1
Power [mW@MHz]	132@130	15@320	52@400	42@250	20@600	0.7@40
Supply Voltage [V]	3.3	2.5	2.3	3.3	–	1.2
Area [mm ²]	0.45	0.07	0.27	0.2	0.38	0.082

Table 4.14: SALT_DLL performance comparison with other designs.

The comparison of the SALT_DLL results with the State-of-Art designs [101, 91, 102, 103, 104] is presented in table 4.14. The projects presented in the table were chosen to have similar frequency ranges. The jitter performance is comparable with other designs, but the power to frequency ratio is the best in the SALT_DLL. Moreover, the DLL designed by the author generates many more clock phases than other presented circuits and occupies almost the smallest area.

Few months ago the first prototype of small 8-channel SALT chip with most of the final functionality implemented, was submitted and fabricated. Among various functionalities also the PLL and DLL blocks were implemented. This 8-channel prototype has not been discussed in the dissertation since the measurements have just been started, but the first measurements results are very promising, in particular the functionality of the PLL based serializer has been already positively verified. One should remember that the operation of all developed circuits should be verified at radiation doses foreseen for the UT detector. Since the systematic tests will be performed in the close future this subject, although very important, is not discussed in the dissertation.

During the long research and development activities author gathered a unique experience in the development of readout systems for HEP experiments and in particular in the issues connected to clock generation and data transmission. He has designed and simulated the key PLL and DLL blocks. In the prototype circuits all layouts were manually drawn, also the digital logic, which allows to reduce the area occupied by the circuit and optimize the power consumption. This designing method is very important in such critical blocks like VCO or VCDL, where parasitic components degrade the circuit parameters very easily. Besides the theoretical analysis, design and simulations of the prototype circuits and their parameterization, author developed advanced measurements setups, with dedicated software, needed for data collection and analyze. The author was also developing the PCBs providing electrical connections and a mechanical montage of the measured chips.

The multi-phase PLL architecture, used in the 2nd version of the SALT_PLL opens the way for construction of very low-power very high speed data serialization circuit. A very high speed data serializer is needed to send high density data streams, as the ones in the readout of the LumiCal detector at ILC. By now in HEP experiments there is not serializer allowing to transfer the data at very high rates (> 5 Gb/s) achieving such low power consumption (~ 10 mW).

Acronyms

AC	Alternating Current	103
ADC	Analog to Digital Converter	26
AFMS	Automatic Frequency Mode Setting	59
ALICE	A Large Ion Collider Experiment	15
ASIC	Application Specific Integrated Circuit	23
ATLAS	A Toroidal LHC Aparatus	15
CAD	Computer-Aided Design	57
CERN	Conseil Européen pour la Recherche Nucléaire	15
CFRP	Carbon-Fibre Reinforced Polymer	18
CKM	Cabibbo-Kobayashi-Maskawa	15
CMOS	Complementary Metal-Oxide Semiconductor	27
CMS	Compact Muon Solenoid	15
CP	Charge Pump	41
CPU	Central Processing Unit	20
CSI	Current-Starved Inverter	48
DAC	Digital to Analog Converter	29
DDR	Double Data Rate	32
DLL	Delay-Locked Loop	26
DNL	Differential Nonlinearity	79
DSP	Digital Signal Processing	26
ECAL	Electromagnetic CALorimeter	19
ECS	Experiment Control System	32
EFF	Event Filter Farm	20
ELT	Enclosed Layout Transistors	58
ENC	Equivalent Noise Charge	28
FIFO	First In, First Out	32
FPGA	Field-Programmable Gate Array	103
FD	Frequency Detector	40
GEM	Gas Electron Multiplier	20
HCAL	Hadronic CALorimeter	19
HEP	High Energy Physics	15
HLT	High-Level Trigger	19
HPD	Hybrid Photon Detector	18
IC	Integrated Circuit	57
ILC	International Linear Collider	58
INL	Integral Nonlinearity	79
IT	Inner Tracker	18
LAN	Local Area Network	103
LCD	Liquid Crystal Display	111
LHC	Large Hadron Collider	15

LHCb	Large Hadron Collider beauty	15
LPF	Low-Pass Filter	35
LSB	Least Significant Bit	30
LumiCal	Luminosity Calorimeter	58
MC	Monte Carlo	60
MCS	Merged Capacitor Switching	31
MIM	Metal-Insulator-Metal	31
MSB	Most Significant Bit	30
MWPC	Multi Wire Proportional Chambers	19
OT	Outer Tracker	18
PCB	Printed Circuit Board	96
PD	Phase Detector	35
PFD	Phase and Frequency Detector	35
PLL	Phase-Locked Loop	26
PS	Pre-Shower	19
PSRR	Power Supply Rejection Ratio	59
PZC	Pole-Zero Cancellation	28
QCD	Quantum Chromodynamics	16
RAM	Random-Access Memory	32
RFC	Recycled Folded Cascode	29
RICH	Ring Imaging CHerenkov	17
RMS	Root Mean Square	59
RX	Receiver	33
SALT	Silicon ASIC for LHCb Tracking	24
SAR	Successive Approximation Register	30
SC	Switched Capacitor	30
SFT	Scintillating Fibre Tracker	21
SLVS	Scalable Low-Voltage Signaling	24
SM	Standard Model	15
SPD	Scintillating Pad Detector	19
SPI	Serial Peripheral Interface	100
TFC	Timing and Fast Control	32
TID	Total Ionizing Dose	58
TT	Trigger Tracker	17
TX	Transmitter	33
UART	Universal Asynchronous Receiver/Transmitter	111
USB	Universal Serial Bus	103
UT	Upstream Tracker	21
VCDL	Voltage-Controlled Delay Line	44
VELO	VErtex LOcator	17
VHDCI	Very-High-Density Cable Interconnect	103
VCO	Voltage-Controlled Oscillator	35
ZS	Zero Suppression	31

Bibliography

- [1] S. Braibant, G. Giacomelli, and M. Spurio, *Particles and Fundamental Interactions: An Introduction to Particle Physics*. Springer-Verlag, 2009.
- [2] F. Close, *Particle Physics: A Very Short Introduction*. Oxford University Press, 2004.
- [3] LHCb Collaboration, “Framework TDR for the LHCb Upgrade: Technical Design Report,” Geneva, Apr 2012. [Online]. Available: <http://cds.cern.ch/record/1443882>
- [4] LHCb Collaboration, “LHCb detector performance,” 2014, CERN-PH-EP-2014-290, LHCb-DP-2014-002. [Online]. Available: <https://cds.cern.ch/record/1978280/files/arXiv:1412.6352.pdf>
- [5] LHCb Collaboration, “The LHCb detector at the LHC,” *Journal of Instrumentation*, vol. 3, no. 08, p. S08005, 2008. [Online]. Available: <http://stacks.iop.org/1748-0221/3/i=08/a=S08005>
- [6] CERN, “The large hadron collider.” [Online]. Available: <http://home.web.cern.ch/topics/large-hadron-collider>
- [7] ATLAS Collaboration, “ATLAS detector and physics performance, technical design report,” 1999, ATLAS TDR 14, CERN/LHCC 99-14. [Online]. Available: http://www.cern.ch/Atlas/GROUPS/PHYSICS/TDR/physics_tdr/printout/Volume_I.pdf
- [8] CMS Collaboration, “CMS physics: Technical design report volume 1: Detector performance and software,” 2006, CMS TDR 8.1, CERN/LHCC 2006-001. [Online]. Available: <http://cds.cern.ch/record/922757/files/lhcc-2006-001.pdf>
- [9] ALICE Collaboration, “Technical design report for the upgrade of the ALICE inner tracking system,” *Journal of Physics G: Nuclear and Particle Physics*, vol. 41, no. 8, p. 087002, 2014. [Online]. Available: <http://stacks.iop.org/0954-3899/41/i=8/a=087002>
- [10] N. Cabibbo, “Unitary symmetry and leptonic decays,” *Phys. Rev. Lett.*, vol. 10, pp. 531–533, Jun 1963. [Online]. Available: <http://link.aps.org/doi/10.1103/PhysRevLett.10.531>
- [11] M. Kobayashi and T. Maskawa, “CP-Violation in the renormalizable theory of weak interaction,” *Progress of Theoretical Physics*, vol. 49, no. 2, pp. 652–657, 1973. [Online]. Available: <http://ptp.oxfordjournals.org/content/49/2/652.abstract>
- [12] A. D. Sakharov, “Violation of CP invariance, C asymmetry, and baryon asymmetry of the universe,” *Soviet Physics Uspekhi*, vol. 34, no. 5, p. 392, 1991. [Online]. Available: <http://stacks.iop.org/0038-5670/34/i=5/a=A08>
- [13] LHCb Collaboration, “Measurement of $\sigma(pp \rightarrow b\bar{b}X)$ at $\sqrt{s} = 7 \text{ TeV}$ in the forward region,” *Physics Letters B*, vol. 694, no. 3, pp. 209 – 216, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0370269310012074>
- [14] LHCb Collaboration, “Implications of LHCb measurements and future prospects,” *The European Physical Journal C*, vol. 73, no. 4, 2013. [Online]. Available: <http://dx.doi.org/10.1140/epjc/s10052-013-2373-2>

- [15] LHCb Collaboration, “First Evidence for the Decay $B_s^0 \rightarrow \mu^+ \mu^-$,” *Phys. Rev. Lett.*, vol. 110, no. arXiv:1211.2674. CERN-PH-EP-2012-335. LHCb-PAPER-2012-043, p. 021801. 9 p, Nov 2012. [Online]. Available: <https://cds.cern.ch/record/1493302>
- [16] LHCb Collaboration, “Measurement of the $B_s^0 \rightarrow \mu^+ \mu^-$ branching fraction and search for $B^0 \rightarrow \mu^+ \mu^-$ decays at the LHCb experiment,” *Phys. Rev. Lett.*, vol. 111, no. arXiv:1307.5024. CERN-PH-EP-2013-128. LHCb-PAPER-2013-046, p. 101805. 12 p, Jul 2013. [Online]. Available: <https://cds.cern.ch/record/1563073>
- [17] LHCb Collaboration, “Measurement of form-factor-independent observables in the decay $B^0 \rightarrow K^{*0} \mu^+ \mu^-$,” *Phys. Rev. Lett.*, vol. 111, p. 191801, Nov 2013. [Online]. Available: <http://link.aps.org/doi/10.1103/PhysRevLett.111.191801>
- [18] LHCb Collaboration, “Differential branching fraction and angular analysis of the decay $B^0 \rightarrow K^{*0} \mu^+ \mu^-$,” *J. High Energy Phys.*, vol. 08, no. arXiv:1304.6325. CERN-PH-EP-2013-074. LHCb-PAPER-2013-019, p. 131. 32 p, Apr 2013. [Online]. Available: <https://cds.cern.ch/record/1542368>
- [19] LHCb Collaboration, “Measurement of the CP-violating phase ϕ_s in $\bar{B}_s^0 \rightarrow J/\psi \pi^+ \pi^-$ decays,” *Phys. Lett. B*, vol. 736, no. arXiv:1405.4140. CERN-PH-EP-2014-086. LHCb-PAPER-2014-019, p. 186. 14 p, May 2014. [Online]. Available: <https://cds.cern.ch/record/1702543>
- [20] LHCb Collaboration, “Measurement of the CKM angle γ from a combination of $B^\pm \rightarrow Dh^\pm$ analyses,” *Phys. Lett. B*, vol. 726, no. arXiv:1305.2050. CERN-PH-EP-2013-079. LHCb-PAPER-2013-020, pp. 151–163, Jul 2013. [Online]. Available: <https://cds.cern.ch/record/1546538>
- [21] M. Needham and T. Ruf, “Estimation of the material budget of the LHCb detector,” 2007, LHCb-2007-025, CERN-LHCb-2007-025. [Online]. Available: <http://cds.cern.ch/record/1023537/files/lhcb-2007-025.pdf>
- [22] LHCb Collaboration, “LHCb technical design report: Reoptimized detector,” 2003, LHCb TDR 9, CERN/LHCC 2003-030. [Online]. Available: <http://lhcb.web.cern.ch/lhcb/TDR/TDR9.pdf>
- [23] LHCb Collaboration, “LHCb tracker upgrade technical design report,” 2015, LHCb TDR 15, CERN/LHCC 2014-001. [Online]. Available: <https://cds.cern.ch/record/1647400/files/LHCB-TDR-015.pdf>
- [24] L. Evans and P. Bryant, “LHC machine,” *Journal of Instrumentation*, vol. 3, no. 08, p. S08001, 2008. [Online]. Available: <http://stacks.iop.org/1748-0221/3/i=08/a=S08001>
- [25] CERN, “LHCb photos.” [Online]. Available: <https://cds.cern.ch/collection/LHCb%20Photos>
- [26] M. Moritz, “The LHCb experiment,” in *Nuclear Science Symposium Conference Record, 2003 IEEE*, vol. 3, Oct 2003, pp. 1499–1503 Vol.3.
- [27] L. Zhang, R. Mao, F. Yang, and R. Zhu, “LSO/LYSO crystals for calorimeters in future HEP experiments,” in *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), 2013 IEEE*, Oct 2013, pp. 1–6.
- [28] C. Parkes, “The LHCb upgrade,” *Nuclear Instruments and Methods in Physics Research Section A*, vol. 569, no. 1, pp. 115 – 118, 2006, proceedings of the 14th International Workshop on Vertex Detectors VERTEX 2005.
- [29] S. Lochner and M. Schmelling, “The Beetle Reference Manual - chip version 1.3, 1.4 and 1.5,” CERN, Geneva, Tech. Rep. LHCb-2005-105. CERN-LHCb-2005-105, Nov 2006. [Online]. Available: <http://cds.cern.ch/record/1000429>
- [30] M. Cepeda, S. Dardin, M. G. D. Gilchriese, C. Haber, W. K. Miller, W. O. Miller, and R. Post, “Mechanical and Cooling Design Studies for an Integrated Stave Concept

- for Silicon Strip Detectors for the Super LHC,” CERN, Geneva, Tech. Rep. ATL-UPGRADE-PUB-2008-001. ATL-COM-UPGRADE-2008-001, Jun 2008. [Online]. Available: <https://cds.cern.ch/record/1109141>
- [31] D. T. Rooney, D. Nager, D. Geiger, and D. Shanguan, “Evaluation of wire bonding performance, process conditions, and metallurgical integrity of chip on board wire bonds,” *Microelectronics Reliability*, vol. 45, no. 2, pp. 379 – 390, 2005.
- [32] G. Corti and L. Shekhtman, “Radiation background in the LHCb experiment,” CERN, Geneva, Tech. Rep. LHCb-2003-083, Oct 2003. [Online]. Available: <http://cds.cern.ch/record/691703>
- [33] Y. Unno et al, “Development of n-on-p silicon sensors for very high radiation environments,” *Nuclear Instruments and Methods in Physics Research Section A*, vol. 636, no. 1, Supplement, pp. S24 – S30, 2011, 7th International Hiroshima Symposium on the Development and Application of Semiconductor Tracking Detectors.
- [34] M. Firlej, T. Fiutowski, M. Idzik, J. Moron, and K. Swientek, “A fast, low-power, 6-bit SAR ADC for readout of strip detectors in the LHCb upgrade experiment,” *Journal of Instrumentation*, vol. 9, July 2014.
- [35] K. Wyllie et al, “Electronics architecture of the LHCb upgrade,” 2013, LHCb-PUB-2011-011. [Online]. Available: <http://cds.cern.ch/record/1340939/files/LHCb-PUB-2011-011.pdf>
- [36] C. Nowlin, “Pulse shaping for nuclear pulse amplifiers,” *Nuclear Science, IEEE Transactions on*, vol. 17, no. 1, pp. 226–241, Feb 1970.
- [37] J. Blankenship and C. Nowlin, “New concepts in nuclear pulse amplifier design,” *Nuclear Science, IEEE Transactions on*, vol. 13, no. 3, pp. 495–507, June 1966.
- [38] D. Binkley, *Tradeoffs and Optimization in Analog CMOS Design*. John Wiley & Sons, Ltd., 2008.
- [39] R. Assaad and J. Silva-Martinez, “The recycling folded cascode: A general enhancement of the folded cascode amplifier,” *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 9, pp. 2535–2542, Sept 2009.
- [40] M. Rezaei, E. Zhian-Tabasy, and S. Ashtiani, “Slew rate enhancement method for folded-cascode amplifiers,” *Electronics Letters*, vol. 44, no. 21, pp. 1226–1228, October 2008.
- [41] L. Yilei, H. Kefeng, Y. Na, T. Xi, and M. Hao, “Analysis and implementation of an improved recycling folded cascode amplifier,” *Journal of Semiconductors*, vol. 33, no. 2, p. 025002, 2012.
- [42] F. Krummenacher, “Pixel detectors with local intelligence: an IC designer point of view,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 305, no. 3, pp. 527 – 532, 1991.
- [43] D. Johns and K. Martin, *Analog Integrated Circuit Design*. John Willey & Sons, Inc., 1997.
- [44] S.-W. Chen and R. Brodersen, “A 6-bit 600-MS/s 5.3-mW asynchronous ADC in 0.13 μm CMOS,” *Solid-State Circuits, IEEE Journal of*, vol. 41, no. 12, pp. 2669–2680, Dec 2006.
- [45] G. Van der Plas and B. Verbruggen, “A 150 MS/s 133 μW 7 bit ADC in 90 nm Digital CMOS,” *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 12, pp. 2631–2640, Dec 2008.
- [46] P. Nuzzo, C. Nani, C. Armiento, A. Sangiovanni-Vincentelli, J. Craninckx, and G. Van der Plas, “A 6-bit 50-MS/s threshold configuring SAR ADC in 90-nm digital CMOS,” in *VLSI Circuits, 2009 Symposium on*, June 2009, pp. 238–239.
- [47] M. Dessouky and A. Kaiser, “Input switch configuration suitable for rail-to-rail operation of switched op amp circuits,” *Electronics Letters*, vol. 35, no. 1, pp. 8–10, Jan 1999.
- [48] H. Jeon, Y.-B. Kim, and M. Choi, “Offset voltage analysis of dynamic latched comparator,” in

- Circuits and Systems (MWSCAS), 2011 IEEE 54th International Midwest Symposium on*, Aug 2011, pp. 1–4.
- [49] Y. Zhu, C.-H. Chan, U.-F. Chio, S.-W. Sin, S.-P. U, R. Martins, and F. Maloberti, “A 10-bit 100-MS/s reference-free SAR ADC in 90 nm CMOS,” *Solid-State Circuits, IEEE Journal of*, vol. 45, no. 6, pp. 1111–1121, June 2010.
- [50] V. Hariprasath, J. Guerber, S.-H. Lee, and U.-K. Moon, “Merged capacitor switching based SAR ADC with highest switching energy-efficiency,” *Electronics Letters*, vol. 46, no. 9, pp. 620–621, April 2010.
- [51] T. Yazaki, I. Morita, and H. Tanaka, “Optical wireless USB 2.0 system with 1 Gbit/s optical transceiver,” in *OptoElectronics and Communications Conference (OECC), 2010 15th*, July 2010, pp. 128–129.
- [52] I. Wickelgren, “The facts about FireWire [serial communication bus],” *Spectrum, IEEE*, vol. 34, no. 4, pp. 19–25, Apr 1997.
- [53] M. Idzik, K. Swientek, T. Fiutowski, S. Kulis, and D. Przyborowski, “A 10-bit multichannel digitizer ASIC for detectors in particle physics experiments,” *IEEE Transactions on Nuclear Science*, vol. 59, no. 2, pp. 294–302, April 2012.
- [54] A. Widmer and P. Franaszek, “A DC-Balanced, partitioned-block, 8B/10B transmission code,” *IBM Journal of Research and Development*, vol. 27, no. 5, pp. 440–451, Sept 1983.
- [55] S. Yadav, S. Pandey, and A. Gupta, “Implementation of 8B/10B encoder-decoder for gigabit ethernet frame,” in *Wireless and Optical Communications Networks (WOCN), 2014 Eleventh International Conference on*, Sept 2014, pp. 1–4.
- [56] J. F. Wakerly, *Digital Design: Principles and Practices*. Prentice Hall, 2005.
- [57] R. T. James, “Data transmission - the art of moving information,” *Spectrum, IEEE*, vol. 2, no. 1, pp. 65–83, Jan 1965.
- [58] T. Granberg, *Handbook of Digital Techniques for High-Speed Design*. Prentice Hall, 2004.
- [59] K. Korbek, *Układy elektroniki front-end*. Wydawnictwo naukowo-dydaktyczne AGH, 2005.
- [60] P. Gray, P. Hurst, S. Lewis, and R. Meyer, *Analysis and Design of Analog Integrated Circuits*, 4th ed. John Wiley & Sons, Inc., 2001.
- [61] B. Razavi, *Design of Analog CMOS Integrated Circuits*. McGraw-Hill, 2001.
- [62] F. Gardner, *Phaselock Techniques*. Wiley, 2005.
- [63] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*. Prentice Hall, 2003.
- [64] D.-K. Jeong, G. Borriello, D. Hodges, and R. Katz, “Design of PLL-based clock generation circuits,” *Solid-State Circuits, IEEE Journal of*, vol. 22, no. 2, pp. 255–261, Apr 1987.
- [65] R. E. Best, *Phase-Locked Loops*. McGraw-Hill, 1993.
- [66] J. Maneatis and M. Horowitz, “Precise delay generation using coupled oscillators,” *Solid-State Circuits, IEEE Journal of*, vol. 28, no. 12, pp. 1273–1282, Dec 1993.
- [67] A. Hajimiri and T. Lee, “Design issues in CMOS differential LC oscillators,” *Solid-State Circuits, IEEE Journal of*, vol. 34, no. 5, pp. 717–724, May 1999.
- [68] M.-T. Hsu, Y.-Y. Lee, and R.-W. Jhong, “Design of linear CMOS VCO based on cross-coupled pair topology with double tuning technique,” in *Microwave Conference (APMC), 2014 Asia-Pacific*, Nov 2014, pp. 962–964.
- [69] H. Deng, Y. Yin, and G. Du, “Phase noise analysis and design of CMOS differential ring VCO,” in *Electronic Measurement Instruments, 2009. ICEMI '09. 9th International Conference on*, Aug

- 2009, pp. 4–731–4–736.
- [70] P.-S. Han and W.-Y. Choi, “1.25/2.5-Gb/s dual bit-rate burst-mode clock recovery circuits in 0.18- μm CMOS technology,” *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 54, no. 1, pp. 38–42, Jan 2007.
- [71] W. J. Dally and J. W. Poulton, *Digital Systems Engineering*. Cambridge University Press, 2008.
- [72] N. G. Einspruch and J. L. Hilbert, *Application Specific Integrated Circuit (ASIC) Technology*. Academic Press, Inc., 1991.
- [73] F. Faccio and G. Cervelli, “Radiation-induced edge effects in deep submicron CMOS transistors,” *Nuclear Science, IEEE Transactions on*, vol. 52, no. 6, pp. 2413–2420, Dec 2005.
- [74] P. Dodd, M. Shaneyfelt, J. Schwank, and J. Felix, “Current and future challenges in radiation effects on CMOS electronics,” *IEEE Transactions on Nuclear Science*, vol. 57, no. 4, pp. 1747–1763, Aug 2010.
- [75] K. Poltorak, F. Tavernier, and P. Moreira, “A radiation-hard PLL for frequency multiplication with programmable input clock and phase-selectable output signals in 130 nm CMOS,” *Journal of Instrumentation*, vol. 7, no. 12, p. C12014, 2012. [Online]. Available: <http://stacks.iop.org/1748-0221/7/i=12/a=C12014>
- [76] A. Hastings, *The Art of Analog Layout*. Prentice Hall, 2001.
- [77] H. Johansson, “A simple precharged CMOS phase frequency detector,” *Solid-State Circuits, IEEE Journal of*, vol. 33, no. 2, pp. 295–299, Feb 1998.
- [78] K. Majeed and B. Kailath, “Low power, high frequency, free dead zone PFD for a PLL design,” in *Faible Tension Faible Consommation (FTFC), 2013 IEEE*, June 2013, pp. 1–4.
- [79] J.-S. Lee, M.-S. Keel, S.-I. Lim, and S. Kim, “Charge pump with perfect current matching characteristics in phase-locked loops,” *Electronics Letters*, vol. 36, no. 23, pp. 1907–1908, Nov 2000.
- [80] J. Gupta, A. Sangal, and H. Verma, “High speed CMOS charge pump circuit for PLL applications using 90nm CMOS technology,” in *Information and Communication Technologies (WICT), 2011 World Congress on*, Dec 2011, pp. 346–349.
- [81] I. Fujimori and T. Sugimoto, “A 1.5 V, 4.1 mW dual-channel audio delta-sigma D/A converter,” *Solid-State Circuits, IEEE Journal of*, vol. 33, no. 12, pp. 1863–1870, Dec 1998.
- [82] M.-B. Lin, *Introduction to VLSI systems*. CRC Press, 2012.
- [83] P. Figueiredo and J. Vital, “Low kickback noise techniques for CMOS latched comparators,” in *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, vol. 1, May 2004, pp. I–537–40 Vol.1.
- [84] Y. Qi, G. Zhang, Z. Shao, and B. Wang, “A low kick back noise latched comparator for high speed folding and interpolating adc,” in *Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on*, Nov 2008, pp. 272–275.
- [85] S. yeop Lee, S. Amakawa, N. Ishihara, and K. Masu, “Low-phase-noise wide-frequency-range ring-VCO-based scalable PLL with subharmonic injection locking in 0.18 μm CMOS,” in *Microwave Symposium Digest (MTT), 2010 IEEE MTT-S International*, May 2010, pp. 1178–1181.
- [86] Y. Luo and K. Zhou, “A 24GHz multi-phase PLL for optical communication,” in *Circuits and Systems, 2007. MWSCAS 2007. 50th Midwest Symposium on*, Aug 2007, pp. 461–464.
- [87] J. Park and M. Flynn, “A low jitter multi-phase PLL with capacitive coupling,” in *Custom Integrated Circuits Conference, 2006. CICC '06. IEEE*, Sept 2006, pp. 753–756.
- [88] C.-H. Heng and B.-S. Song, “A 1.8 GHz CMOS fractional-n frequency synthesizer with randomized multi-phase VCO,” in *Custom Integrated Circuits Conference, 2002. Proceedings of the*

- IEEE 2002*, 2002, pp. 427–430.
- [89] R. J. Baker, *CMOS. Circuit Design, Layout and Simulation*, 3rd ed. John Wiley & Sons, Inc., 2010.
- [90] R. Farjad-Rad, W. Dally, H.-T. Ng, R. Senthinathan, M.-J. Lee, R. Rathi, and J. Poulton, “A low-power multiplying DLL for low-jitter multigigahertz clock generation in highly integrated digital chips,” *Solid-State Circuits, IEEE Journal of*, vol. 37, no. 12, pp. 1804–1812, Dec 2002.
- [91] K.-H. Cheng and Y.-L. Lo, “A fast-lock mixed-mode DLL with wide-range operation and multi-phase outputs,” in *Design, Automation and Test in Europe, 2006. DATE '06. Proceedings*, vol. 2, March 2006, pp. 5 pp.–.
- [92] C.-N. Chuang and S.-I. Liu, “A 20-MHz to 3-GHz wide-range multiphase delay-locked loop,” *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 56, no. 11, pp. 850–854, Nov 2009.
- [93] JEDEC Standard 65B, “Definition of skew specifications for standard logic devices.” [Online]. Available: <http://www.jedec.org/sites/default/files/docs/jesd65b.pdf>
- [94] SiTime, “Clock jitter definitions and measurement methods,” 2014. [Online]. Available: <http://www.sitime.com/support2/documents/AN10007-Jitter-and-measurement.pdf>
- [95] I. Zamek and S. Zamek, “Definitions of jitter measurement terms and relationships,” in *Test Conference, 2005. Proceedings. ITC 2005. IEEE International*, Nov 2005, pp. 10 pp.–34.
- [96] M. Firlej, T. Fiutowski, M. Idzik, J. Moron, and K. Swientek, “Development of scalable frequency and power phase-locked loop in 130 nm CMOS technology,” *Journal of Instrumentation*, vol. 9, February 2014.
- [97] J. Maneatis, J. Kim, I. McClatchie, J. Maxey, and M. Shankaradas, “Self-biased high-bandwidth low-jitter 1-to-4096 multiplier clock generator PLL,” *Solid-State Circuits, IEEE Journal of*, vol. 38, no. 11, pp. 1795–1803, Nov 2003.
- [98] C. N. Chuang and S. I. Liu, “A 1 V phase locked loop with leakage compensation in 0.13 μm CMOS technology,” in *IEICE Trans. Electron.*, vol. E89-C, March 2006, p. 295–299.
- [99] I.-T. Lee, Y.-T. Tsai, and S.-I. Liu, “A wide-range PLL using Self-Healing Prescaler/VCO in 65-nm CMOS,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 21, no. 2, pp. 250–258, Feb 2013.
- [100] R. Holzer, “A 1 V CMOS PLL designed in high-leakage CMOS process operating at 10-700 MHz,” in *Solid-State Circuits Conference, 2002. Digest of Technical Papers. ISSCC. 2002 IEEE International*, vol. 1, Feb 2002, pp. 272–466 vol.1.
- [101] H.-H. Chang, J.-W. Lin, C.-Y. Yang, and S.-I. Liu, “A wide-range delay-locked loop with a fixed latency of one clock cycle,” *Solid-State Circuits, IEEE Journal of*, vol. 37, no. 8, pp. 1021–1027, Aug 2002.
- [102] S. J. Kim, S. H. Hong, J.-K. Wee, J. H. Cho, P. S. Lee, J. H. Ahn, and J. Y. Chung, “A low-jitter wide-range skew-calibrated dual-loop DLL using antifuse circuitry for high-speed DRAM,” *Solid-State Circuits, IEEE Journal of*, vol. 37, no. 6, pp. 726–734, Jun 2002.
- [103] Y. Moon, J. Choi, K. Lee, D.-K. Jeong, and M.-K. Kim, “An all-analog multiphase delay-locked loop using a replica delay line for wide-range operation and low-jitter performance,” *Solid-State Circuits, IEEE Journal of*, vol. 35, no. 3, pp. 377–384, March 2000.
- [104] S. Hoyos, C. Tsang, J. Vanderhaegen, Y. Chiu, Y. Aibara, H. Khorramabadi, and B. Nikolic, “A 15 MHz to 600 MHz, 20 mW, 0.38 mm^2 split-control, fast coarse locking digital DLL in 0.13 μm CMOS,” *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 20, no. 3, pp. 564–568, March 2012.

List of Figures

1.1	Block diagram of the current LHCb detector [23]	17
1.2	Photographs of the LHCb subsystems	18
1.3	Photographs of the LHCb subsystems	19
1.4	Schematic of the LHCb upgrade detector [23]	21
1.5	Overview of UT geometry looking downstream	23
1.6	UT stave structure	24
1.7	Mounting of a stave layer to the frame [23]	25
1.8	Four sensors geometries for the UT upgrade [23]	26
1.9	Block diagram of the SALT ASIC (128 channels)	28
1.10	Comparison of several shaping implementation	29
1.11	Simplified block diagram of SALT front-end electronics	29
1.12	Simplified block diagram of the 6-bit SAR ADC	30
1.13	Data processing in SALT	32
2.1	Data transmission methods	33
2.2	Principle of the serialization process	34
2.3	Simplified Phase-Locked Loop block diagram	35
2.4	Principle of the XOR PD operation	36
2.5	Using of the LPF for average the error signal	36
2.6	Block diagram of the PLL with the XOR Phase Detector	37
2.7	PLL response to the reference frequency step	40
2.8	Principle of the PFD	41
2.9	Principle of the Charge Pump operation	41
2.10	The block diagram of a type II Phase-Locked Loop	42
2.11	Using the PLL as a frequency multiplier	44
2.12	The principle of the Delay-Locked Loop operation	45
2.13	The clock phases generated by the Delay-Locked Loop	45
2.14	The Delay-Locked Loop block diagram	45
2.15	The Delay-Locked Loop response to the phase step on the input	47
2.16	Systems based on the propagation delay of the logic gates	47
2.17	Principle of the Current-Starved Inverter operation	48
2.18	Total load capacitance of the Current-Starved Inverter	49
2.19	Simplest version of the VCO (or VC DL)	50
2.20	Gain of voltage controlled circuits	50
2.21	Simplified Phase and Frequency Detector block diagram	51
2.22	Principle of the Phase and Frequency Detector operation - state diagram	51
2.23	The PFD response for different phases or frequencies of input signals	52
2.24	Principle of the Charge Pump operation	52

2.25 CP operation when control signals are not changing in the same time	53
2.26 CP operation when drain currents of M_1 and M_2 transistors are different	53
2.27 Parasitic capacitance of current mirror transistors (M_3 and M_4) influence on the CP operation	54
2.28 Principle of the bootstrapping operation (idle state)	55
2.29 Low-Pass Filter schematic	55
3.1 MULTI_PLL block diagram (both versions 1st and 2nd)	58
3.2 MULTI_PLL synchronization for noisy VDD	59
3.3 MULTI_PLL_V2 synchronization process	60
3.4 MULTI_PLL_V2 after synchronization	60
3.5 VCO control voltage (V_O) stability after MULTI_PLL_V2 synchronization	61
3.6 MULTI_PLL_V2 MC analysis of the synchronization process - schematic simulations	61
3.7 Block diagram of the Voltage-Controlled Oscillator	62
3.8 The schematic of the MULTI_VCO fast oscillation ring	63
3.9 Schematic of the MULTI_VCO slow oscillation ring	63
3.10 MULTI_VCO_V1 layout ($100 \times 50 \mu\text{m}^2$)	64
3.11 MULTI_VCO_V1 output waveforms for 16 MHz (top) and 3 GHz (bottom)	64
3.12 MULTI_VCO output parameters	65
3.13 MULTI_VCO period jitter versus oscillation frequency	65
3.14 MULTI_VCO frequency stability for two circuit versions, based on post-layout simulations	66
3.15 MULTI_VCO power consumption dependent on oscillation frequency	67
3.16 Phase and Frequency Detector schematic	68
3.17 PFD output waveforms in case when the reference leads in phase	68
3.18 Phase and Frequency Detector layout ($20 \times 7 \mu\text{m}^2$)	69
3.19 PFD output responses for schematic simulations and post-layout simulation	69
3.20 Phase and Frequency Detector gain curve	70
3.21 Charge Pump schematic	70
3.22 CP layout ($82 \times 27 \mu\text{m}^2$)	71
3.23 Charge Pump output voltage V_O	72
3.24 Block diagram of the MULTI_PLL divider	72
3.25 MULTI_PLL divider waveforms for all division factors, based on post-layout simulations	73
3.26 Simplified block diagram of the Automatic Frequency Mode Setting	74
3.27 Automatic Frequency Mode Setting digital signals when the MULTI_PLL works at 300 MHz	75
3.28 Automatic Frequency Mode Setting digital signals when the MULTI_PLL works at 3 GHz	76
3.29 MULTI_PLL V_O voltage disturbances before the AFMS disabling	76
3.30 SALT_PLL block diagram	77
3.31 Schematic of the SALT_PLL_V2 (SALT_DLL_V2) DAC	78
3.32 Performance of the DAC used in the SALT_PLL_V2	78
3.33 Synchronization process comparison of two SALT_PLL versions	79
3.34 SALT_PLL_V2 synchronization process	80
3.35 SALT_PLL_V2 MC analysis of the synchronization process - schematic simulations	80
3.36 Block diagram of the both SALT_VCO versions	81
3.37 Schematic of the delay stage used in the SALT_VCO_V1	82
3.38 Delay stage layout of 1st circuit version ($76 \times 12 \mu\text{m}^2$)	82
3.39 Schematic of the delay stage used in the SALT_VCO_V2	83
3.40 Delay stage layout of 2nd circuit version ($22 \times 7.3 \mu\text{m}^2$)	83

3.41 SALT_VCO output signals	84
3.42 SALT_VCO output parameters	84
3.43 SALT_VCO period jitter versus oscillation frequency	84
3.44 SALT_VCO frequency stability for two circuit versions at frequency 160 MHz	85
3.45 Simplified schematic of Phase and Frequency Detector	86
3.46 PFD output responses for schematic simulations and post-layout simulations	86
3.47 Phase and Frequency Detector gain curves	87
3.48 Simplified schematic of the Charge Pump with LPF	87
3.49 CP output voltage V_O	88
3.50 SALT_DLL block diagram	89
3.51 Comparison of two SALT_DLL versions and frequency stability during synchronization process	89
3.52 SALT_DLL synchronization process	90
3.53 SALT_DLL MC analysis of the synchronization process - schematic simulations	90
3.54 SALT_VCDL block diagram (1st and 2nd circuit versions)	91
3.55 Schematic of the SALT_VCDL delay stage	92
3.56 Comparison of two SALT_VCDL versions and comparison of schematic and post-layout simulations at control voltage $V_O = 0.6 V$	92
3.57 SALT_VCDL_V2 output jitter ($Ph[64]$) for different bias currents at control voltage $V_O = 0.6 V$	93
3.58 SALT_VCDL_V2 gain curve and duty cycle of the $Ph[64]$ output	93
4.1 Data setup time and data hold time violations caused by the clock jitter	95
4.2 Sampling of a signal by a digital scope	96
4.3 Measurements of a clock period for the period jitter calculations	97
4.4 Measurements of a clock period for the cycle to cycle jitter calculations	97
4.5 Measurements of a clock period for the long term jitter calculations	98
4.6 MULTI_PLL as a part of the multi-channel ADC	99
4.7 Transmission via the slow control interface - command structure	100
4.8 Simplified block diagram of the prototype Printed Circuit Board (ASIC board)	102
4.9 Block diagram of the MULTI_PLL measurements setup	103
4.10 Photograph of the measurement setup	104
4.11 Gain measurements results of the MULTI_PLL	105
4.12 Period jitter measurements results of the MULTI_PLL	105
4.13 Period jitter comparison between two MULTI_PLL versions (1st and 2nd)	106
4.14 Period jitter comparison for all MULTI_PLL dividers	106
4.15 Long term jitter measurements results of the MULTI_PLL	106
4.16 Power consumption of the MULTI_PLL	107
4.17 PLL/DLL blocks as a part of the SALT_DLL_PLL ASIC	108
4.18 Transmission via the slow control interface - command structure	109
4.19 Simplified schematic diagram of the prototype PCB (ASIC board)	112
4.20 Block diagram of the SALT_DLL_PLL measurements setup	113
4.21 Photograph of the measurement setup	113
4.22 Gain measurements results of the SALT_PLL	114
4.23 Period jitter of the SALT_PLL as a function of selected clock phase	115
4.24 Total clock delay of the SALT_PLL as a function of selected clock phase	115
4.25 Period jitter measurements results of the SALT_PLL	116
4.26 Period jitter comparison between two SALT_PLL versions (1st and 2nd)	116
4.27 Period jitter comparison for all SALT_PLL division factors	116

4.28 Power consumption of the SALT_PLL	117
4.29 Phase delay of the SALT_DLL as a function of selected clock phase	117
4.30 Total clock delay of the SALT_DLL as a function of selected clock phase	118
4.31 Period jitter measurements results of the SALT_DLL	118
4.32 Period jitter comparison between two SALT_DLL versions (1st and 2nd)	119
4.33 Period jitter of the SALT_DLL as a function of selected clock phase	119
4.34 Power consumption of the SALT_DLL	119

List of Tables

1.1	Selected parameters and design requirements of the SALT ASIC	27
3.1	Simulated frequency ranges for all modes of two VCO versions	66
3.2	Divider configuration	73
3.3	Adjustable bias circuit configuration	74
3.4	Simulated frequency ranges for all modes for two VCO versions (1st and 2nd)	81
4.1	Basic parameters of two MULTI_PLL prototypes	99
4.2	Command header field structure	100
4.3	MULTI_PLL configuration commands - 1st version	100
4.4	MULTI_PLL configuration commands - 2nd version	101
4.5	ADC modes with the corresponding <i>mode</i> field values of the main_cfg command	101
4.6	PLL divider settings	101
4.7	Basic measured parameters of two SALT_PLL prototypes	107
4.8	Basic measured parameters of two SALT_DLL prototypes	108
4.9	Configuration commands for the SALT_DLL_PLL_V1	109
4.10	Configuration commands for the SALT_DLL_PLL_V2	109
4.11	SALT_PLL divider configuration set by the pll_main_cfg command	110
4.12	SALT_PLL mode configuration set by the pll_main_cfg command	110
4.13	SALT_PLL performance comparison with other designs	122
4.14	SALT_DLL performance comparison with other designs	123

Appendix A

Measurements results of MULTI_PLL

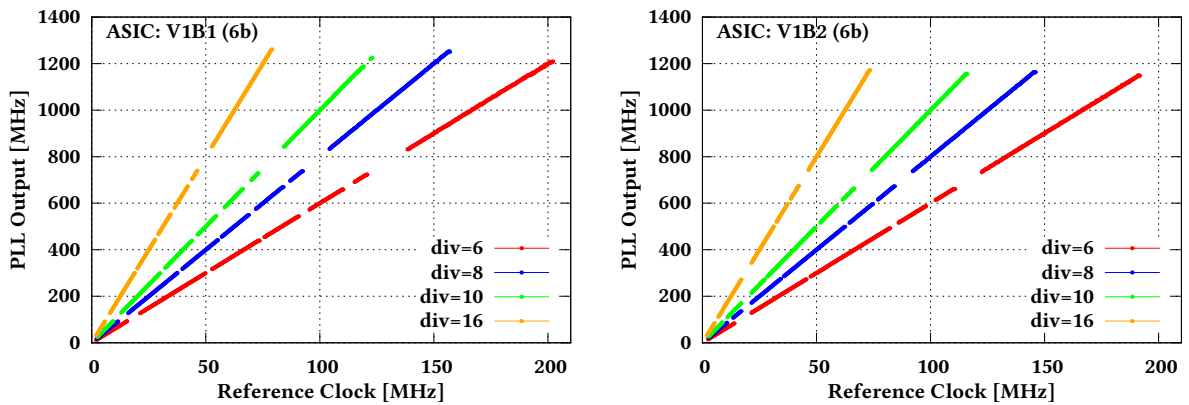


Figure A.1: Gain measurements results of the MULTI_PLL_V1 for two prototypes.

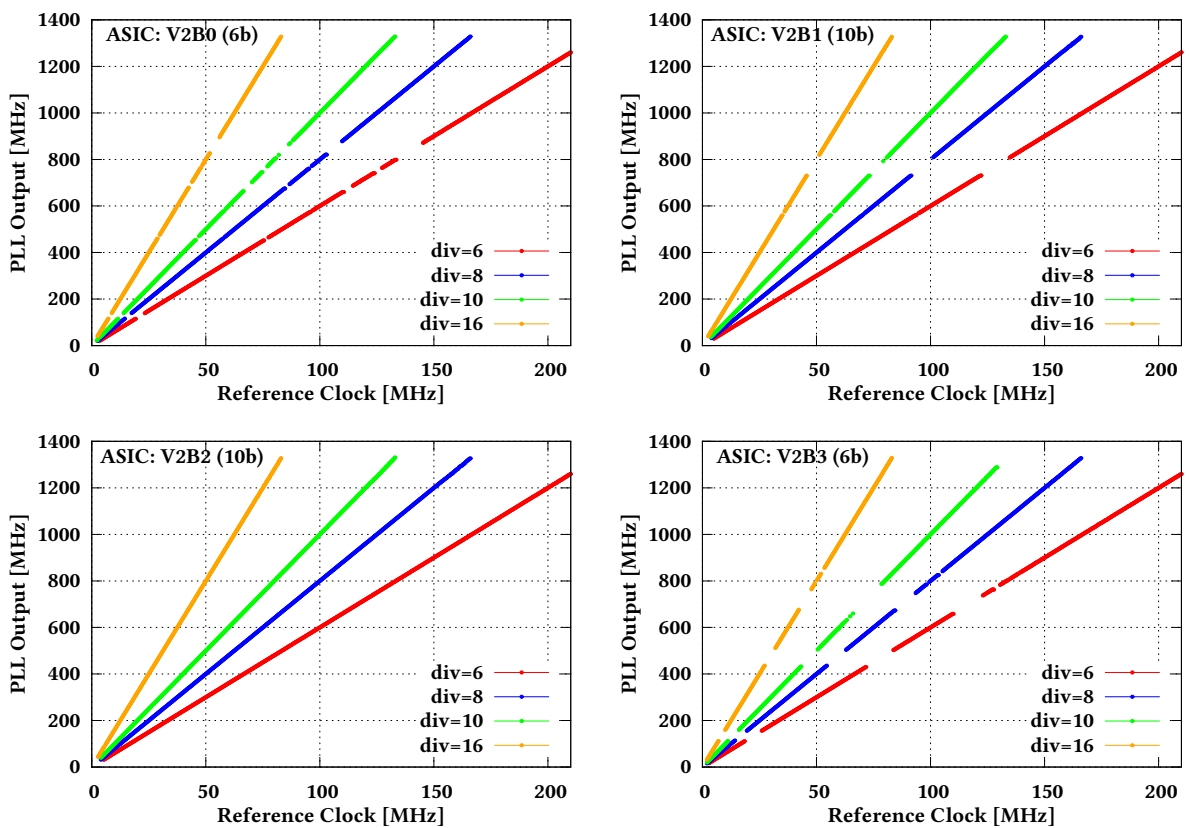


Figure A.2: Gain measurements results of the MULTI_PLL_V2 for four prototypes.

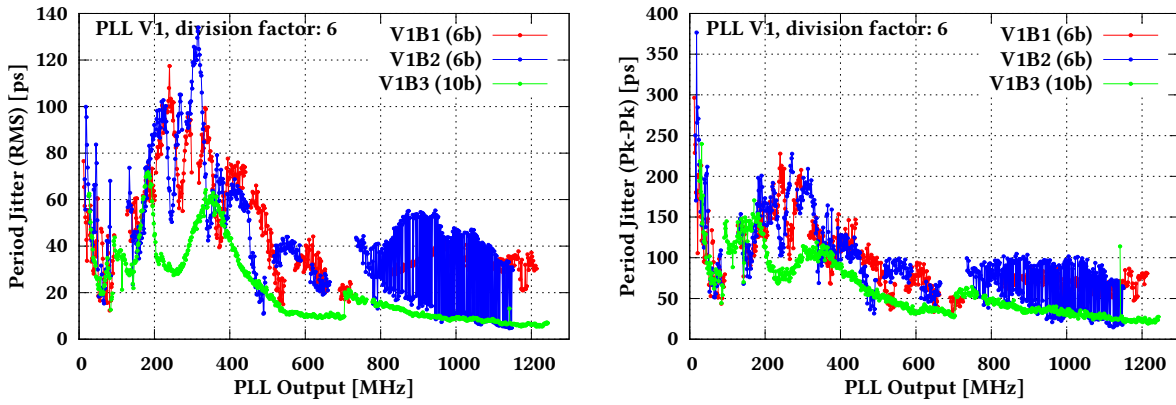


Figure A.3: Period jitter of the MULTI_PLL_V1 for division factor 6.

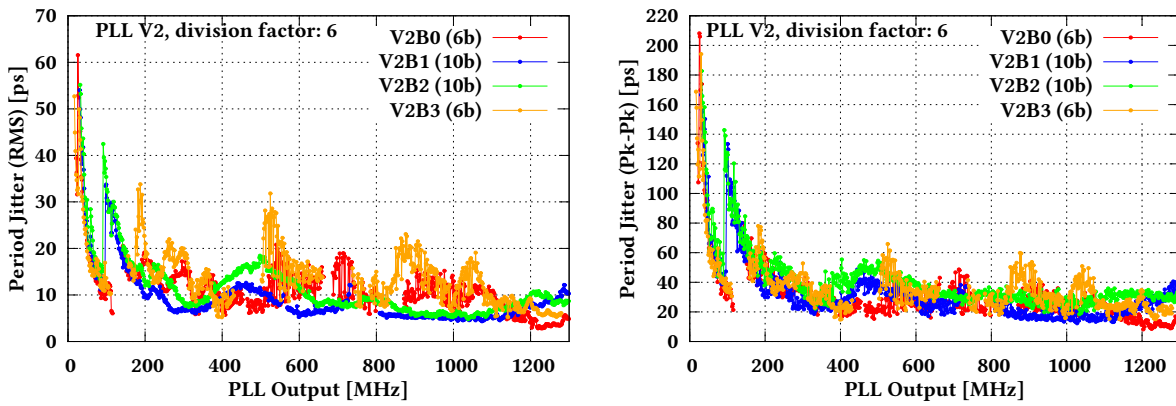


Figure A.4: Period jitter of the MULTI_PLL_V2 for division factor 6.

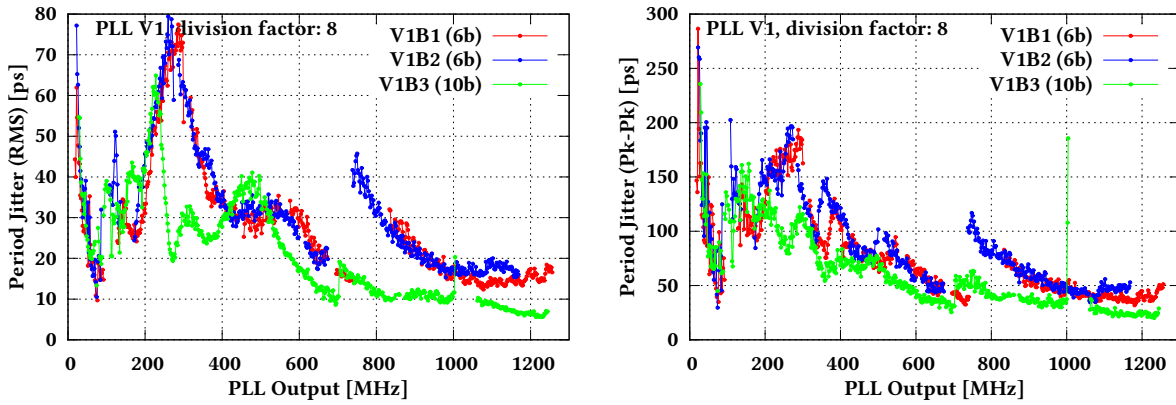


Figure A.5: Period jitter of the MULTI_PLL_V1 for division factor 8.

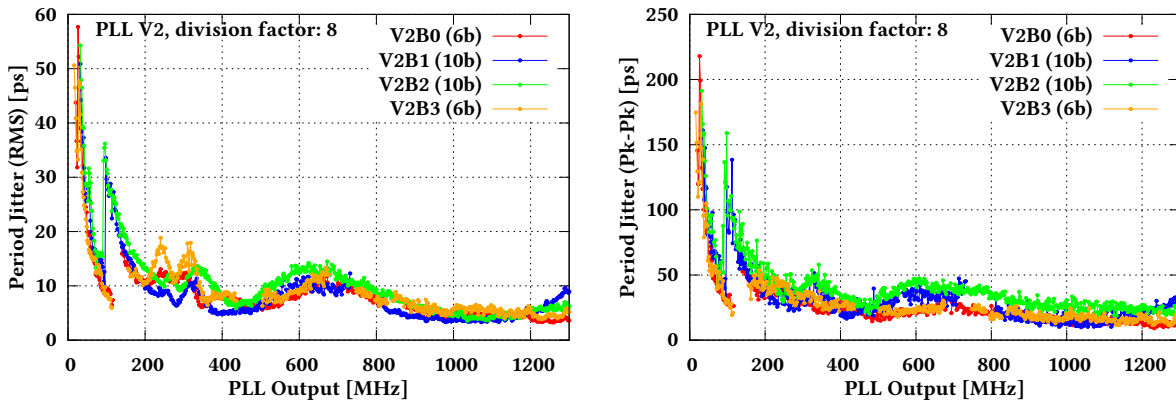


Figure A.6: Period jitter of the MULTI_PLL_V2 for division factor 8.

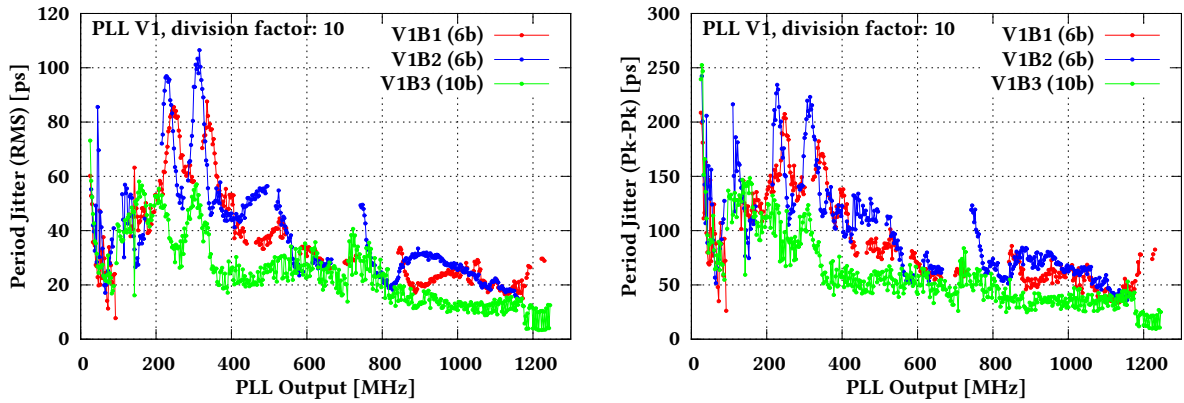


Figure A.7: Period jitter of the *MULTI_PLL_V1* for division factor 10.

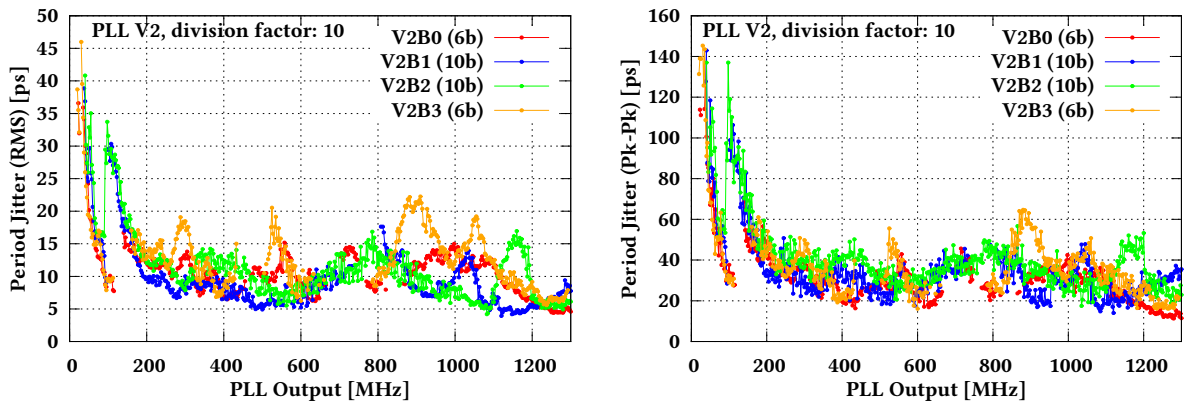


Figure A.8: Period jitter of the *MULTI_PLL_V2* for division factor 10.

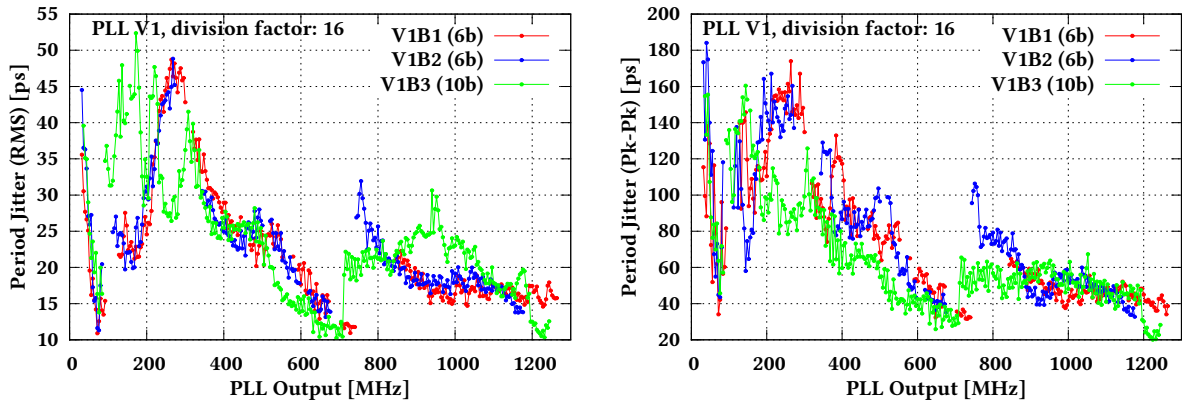


Figure A.9: Period jitter of the *MULTI_PLL_V1* for division factor 16.

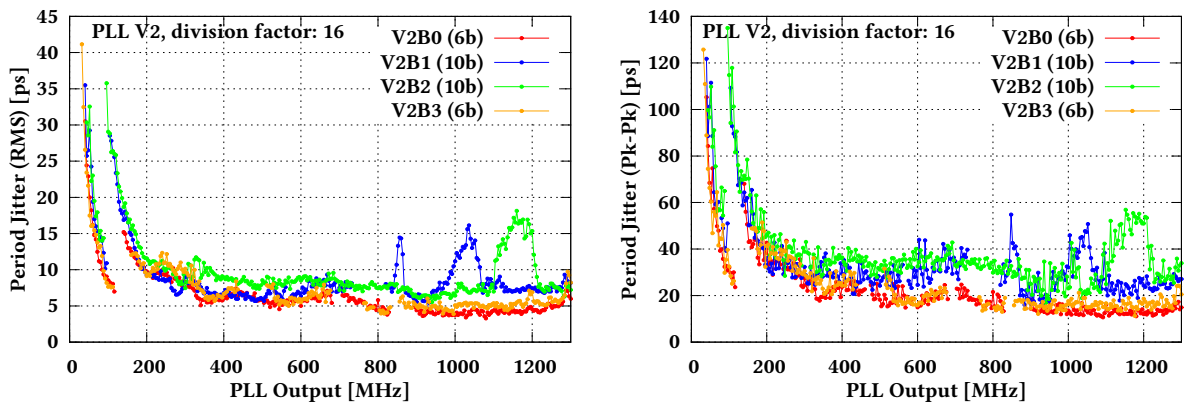


Figure A.10: Period jitter of the *MULTI_PLL_V2* for division factor 16.

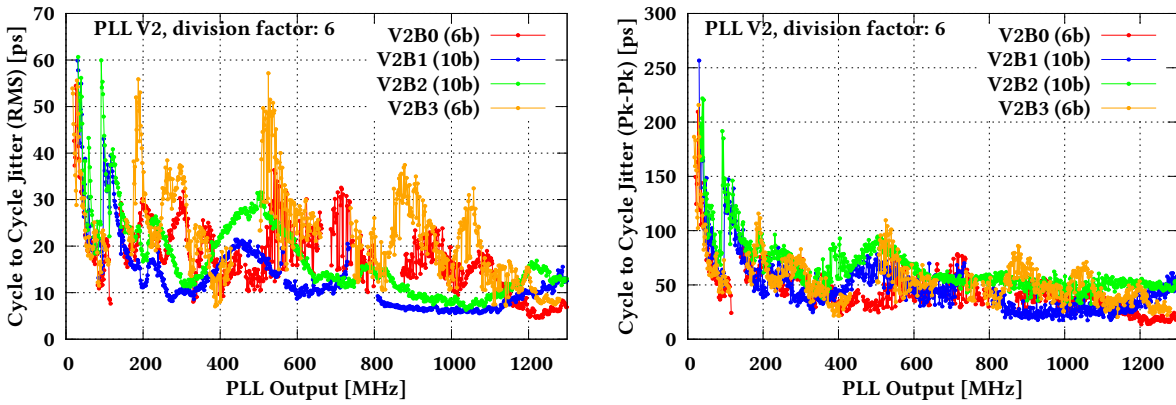


Figure A.11: Cycle to cycle jitter of the MULTI_PLL_V2 for division factor 6.

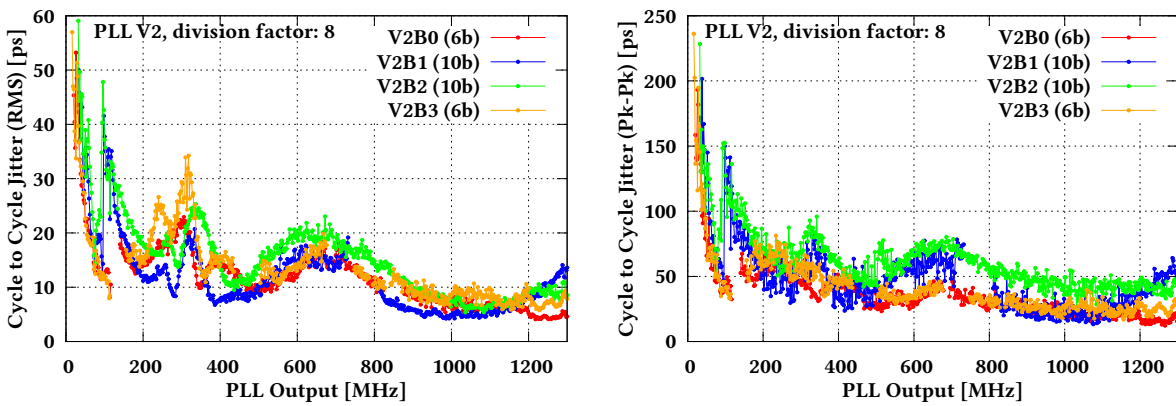


Figure A.12: Cycle to cycle jitter of the MULTI_PLL_V2 for division factor 8.

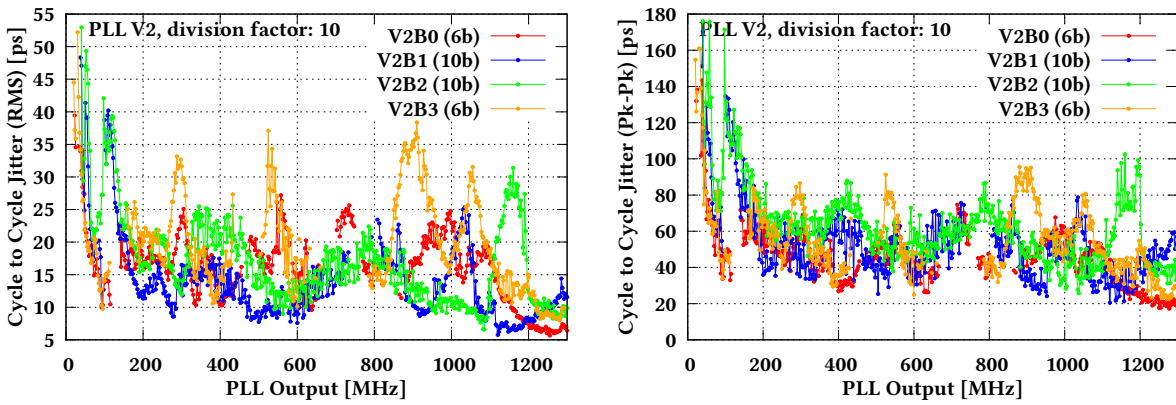


Figure A.13: Cycle to cycle jitter of the MULTI_PLL_V2 for division factor 10.

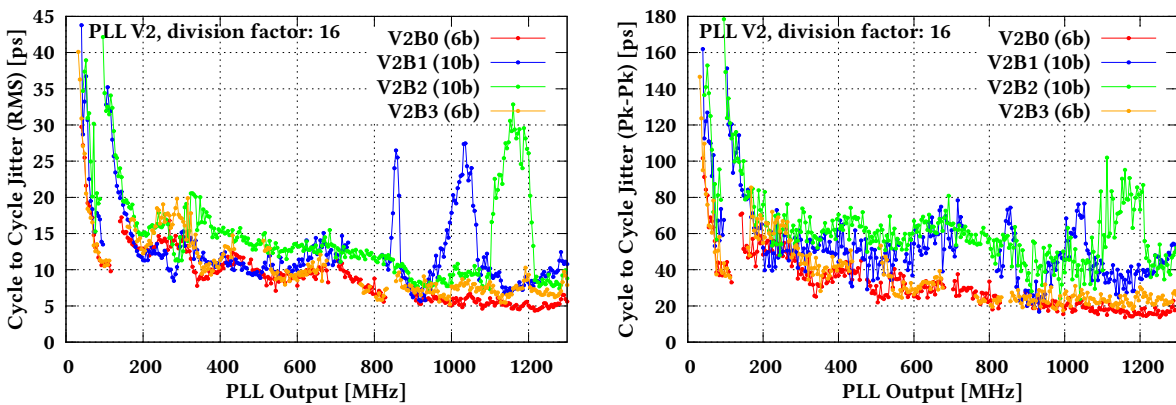


Figure A.14: Cycle to cycle jitter of the MULTI_PLL_V2 for division factor 16.

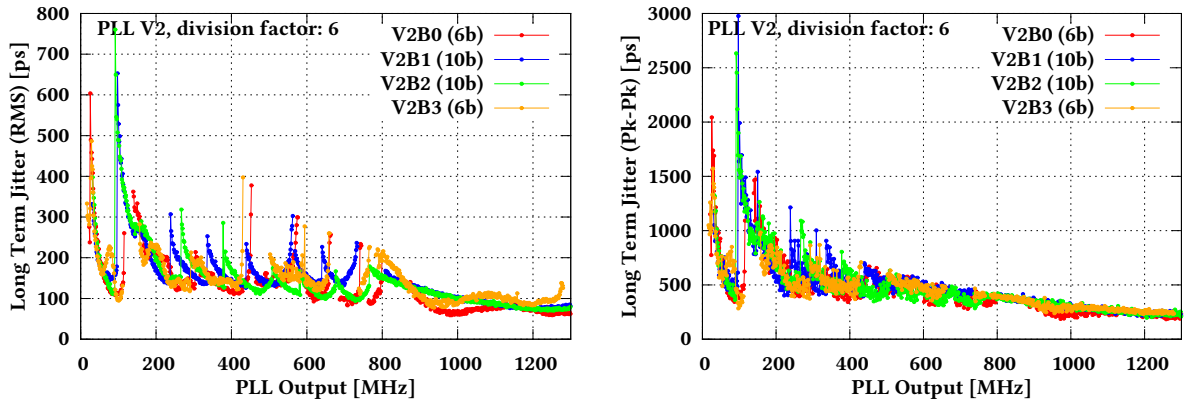


Figure A.15: Long term jitter of the MULTI_PLL_V2 for division factor 6.

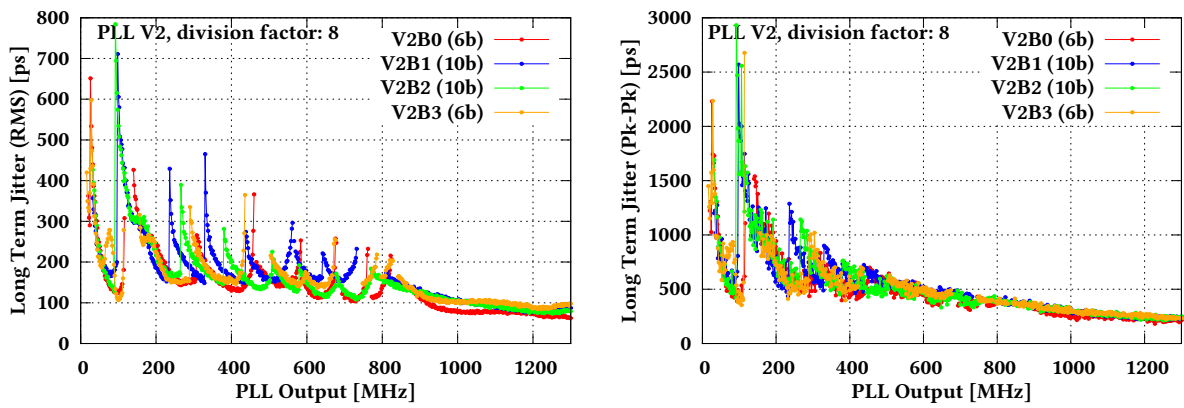


Figure A.16: Long term jitter of the MULTI_PLL_V2 for division factor 8.

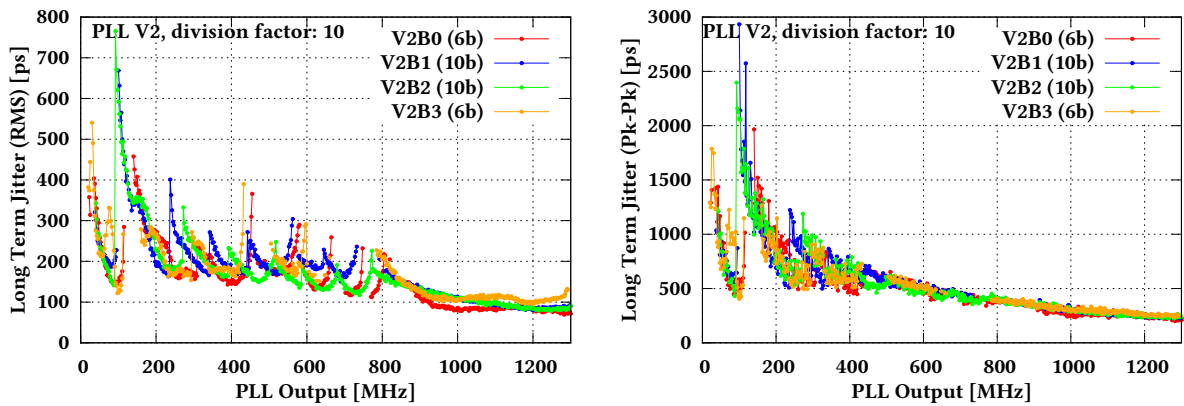


Figure A.17: Long term jitter of the MULTI_PLL_V2 for division factor 10.

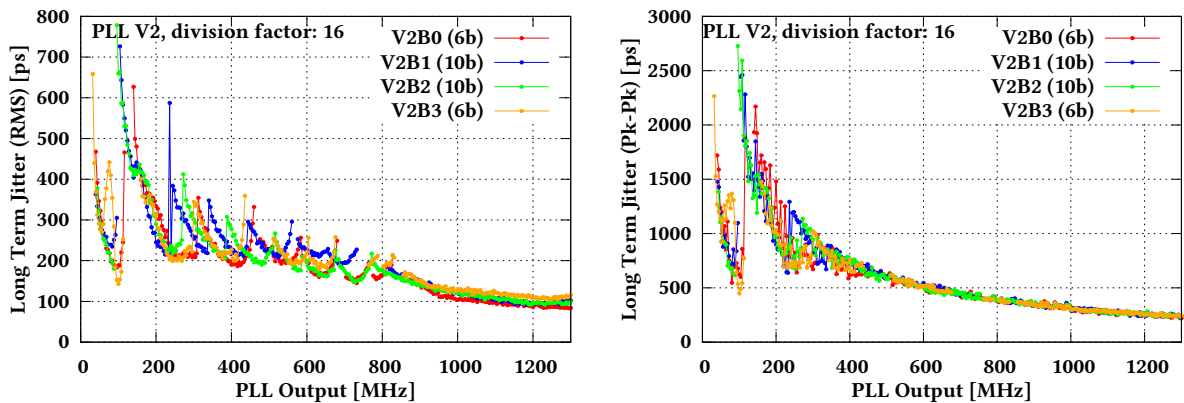


Figure A.18: Long term jitter of the MULTI_PLL_V2 for division factor 16.

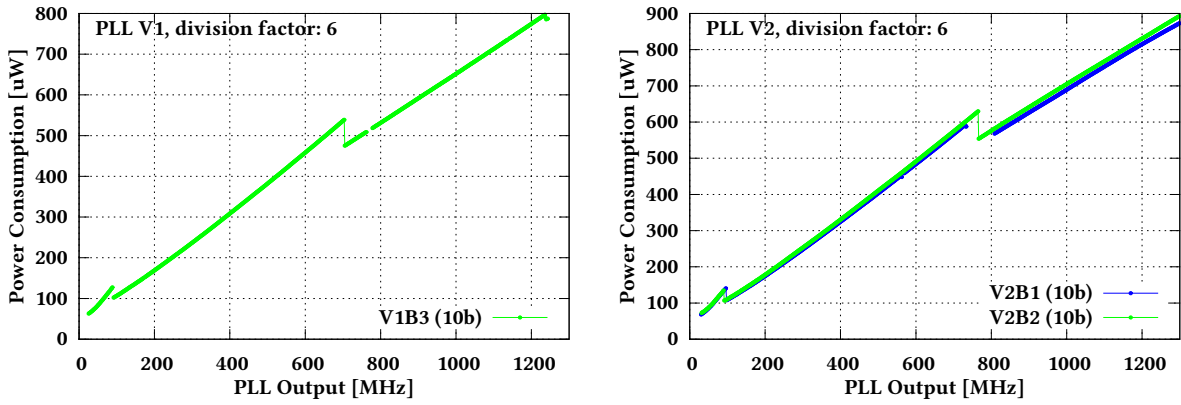


Figure A.19: Power consumption of the MULTI_PLL_V1(left) and V2 (right) for division factor 6.

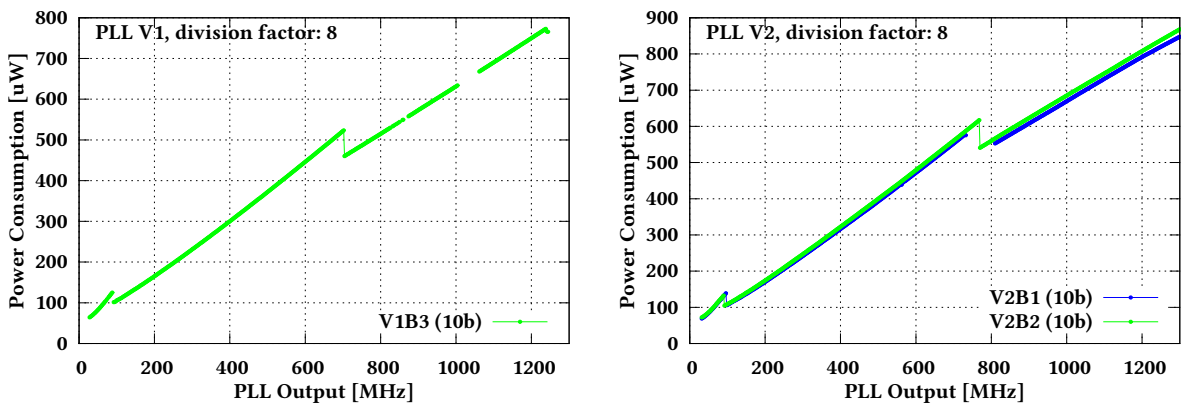


Figure A.20: Power consumption of the MULTI_PLL_V1(left) and V2 (right) for division factor 8.

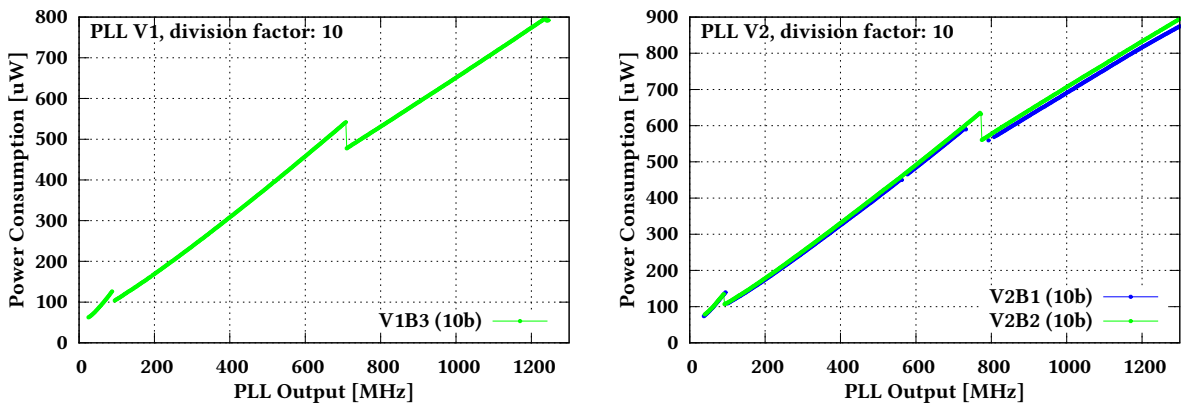


Figure A.21: Power consumption of the MULTI_PLL_V1(left) and V2 (right) for division factor 10.

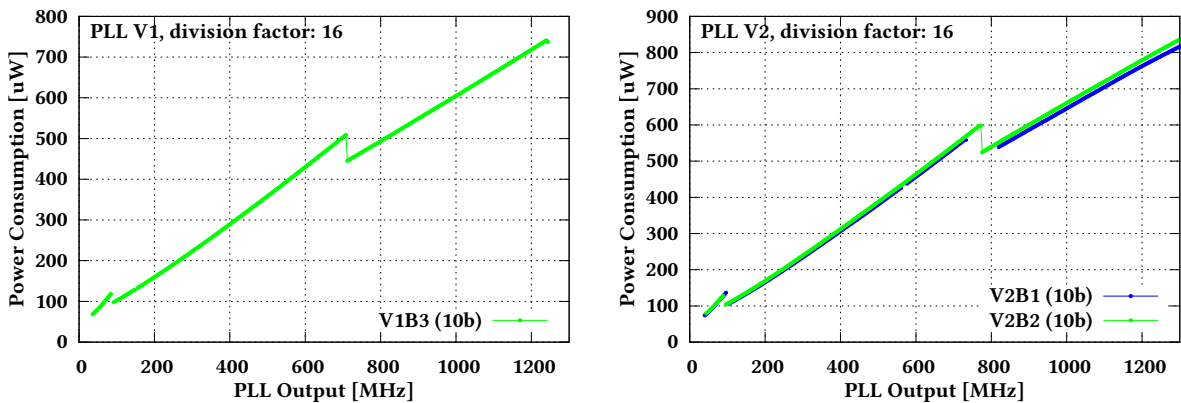


Figure A.22: Power consumption of the MULTI_PLL_V1(left) and V2 (right) for division factor 16.

Appendix B

Measurements results of SALT_PLL

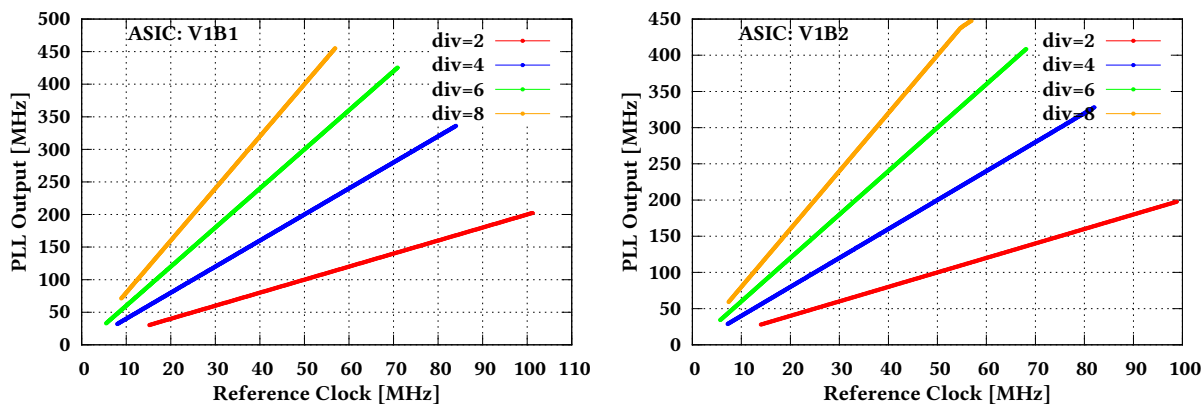


Figure B.1: Gain measurements results of the SALT_PLL_V1 for two prototypes.

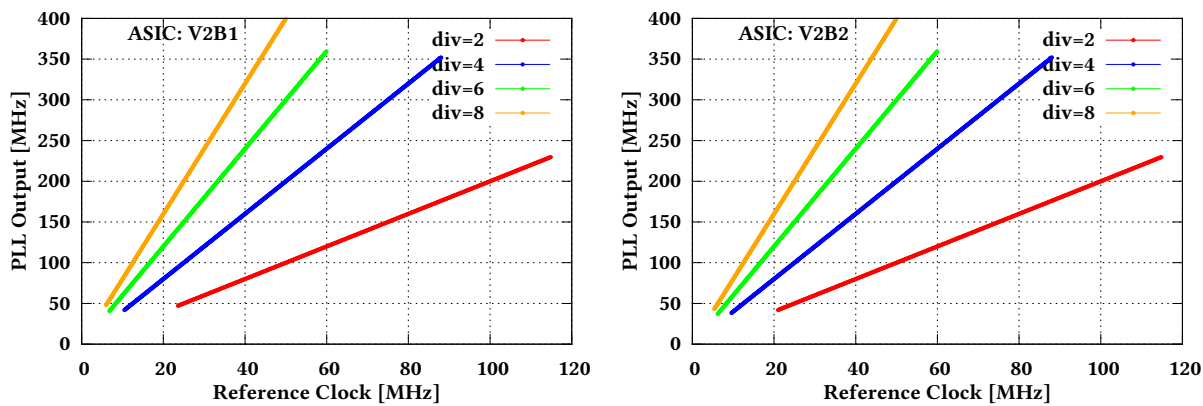


Figure B.2: Gain measurements results of the SALT_PLL_V2 for two prototypes.

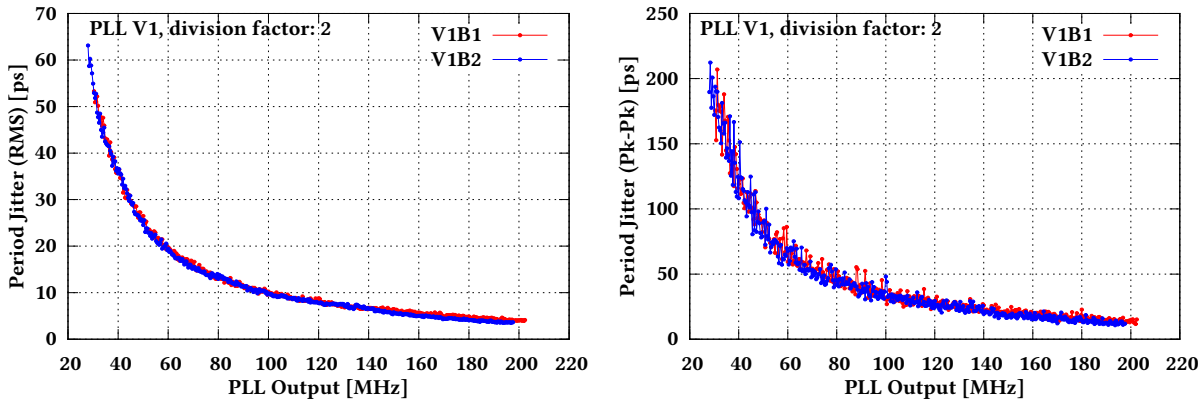


Figure B.3: Period jitter of the SALT_PLL_V1 for division factor 2.

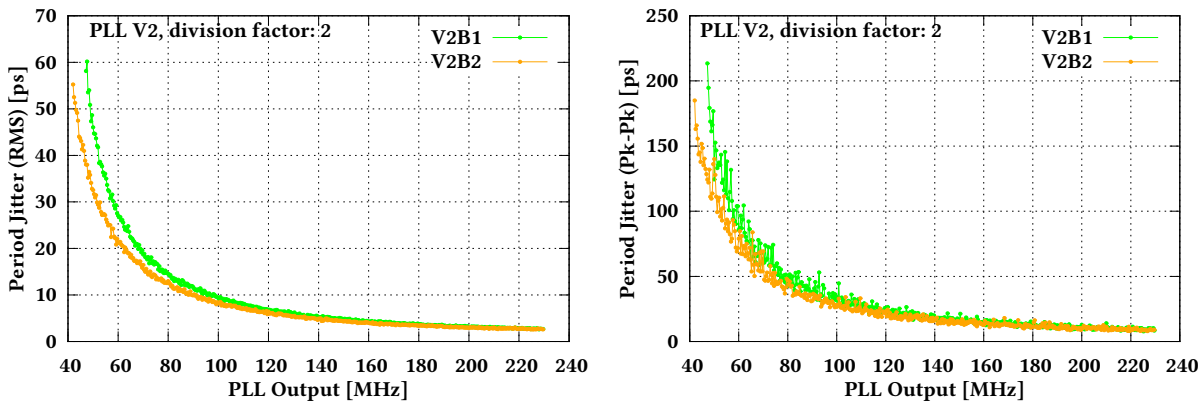


Figure B.4: Period jitter of the SALT_PLL_V2 for division factor 2.

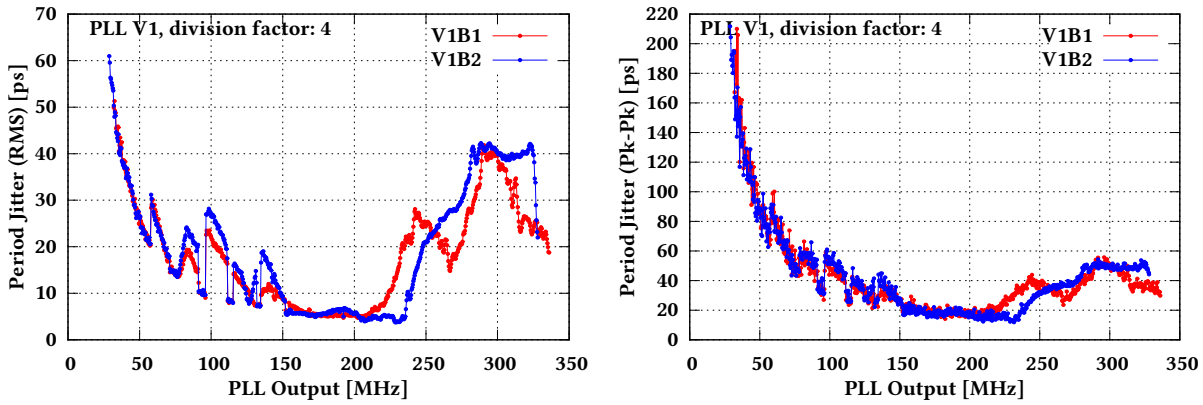


Figure B.5: Period jitter of the SALT_PLL_V1 for division factor 4.

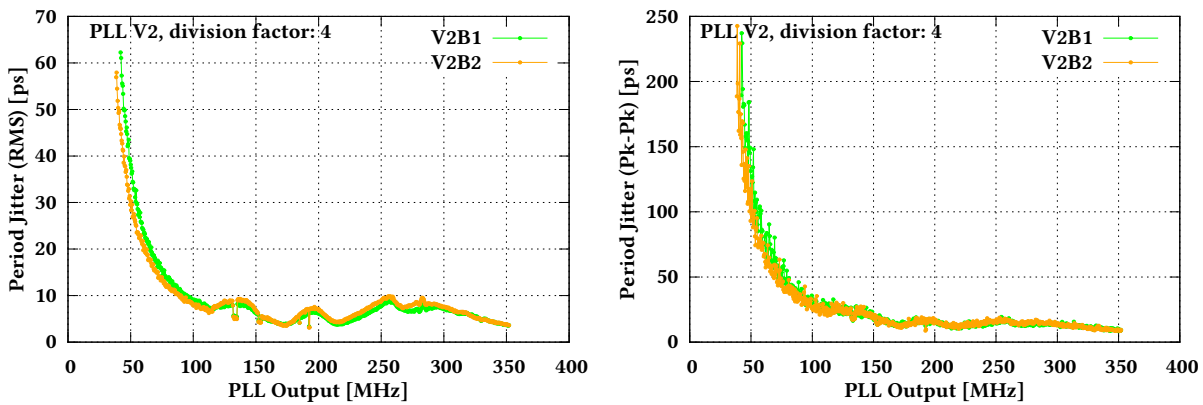


Figure B.6: Period jitter of the SALT_PLL_V2 for division factor 4.

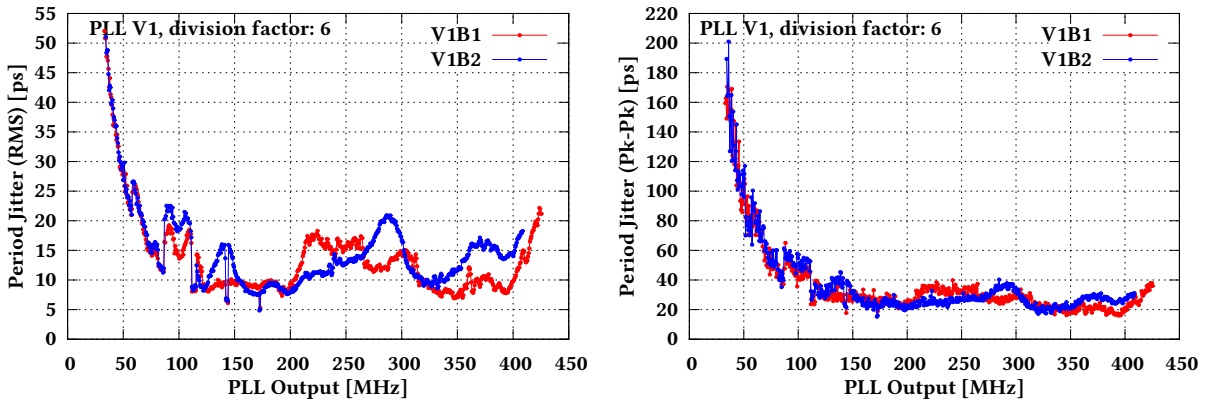


Figure B.7: Period jitter of the SALT_PLL_V1 for division factor 6.

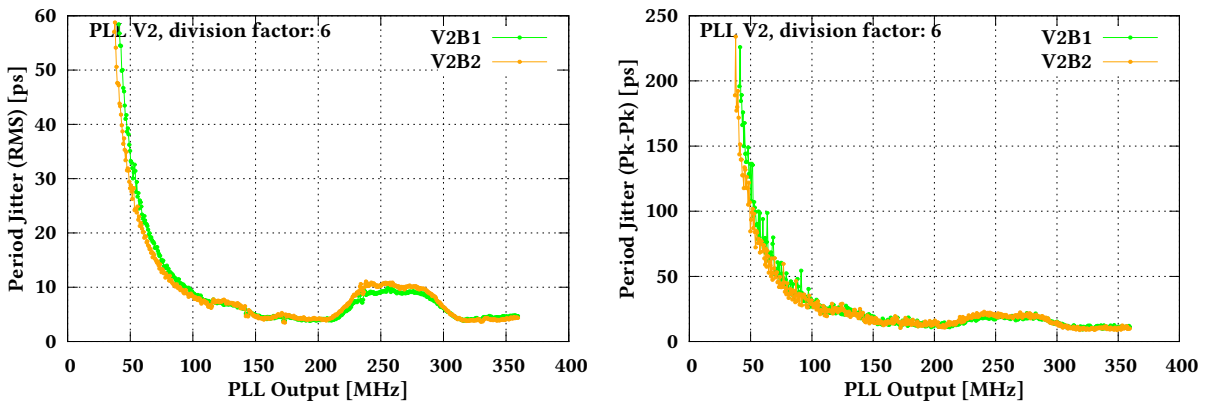


Figure B.8: Period jitter of the SALT_PLL_V2 for division factor 6.

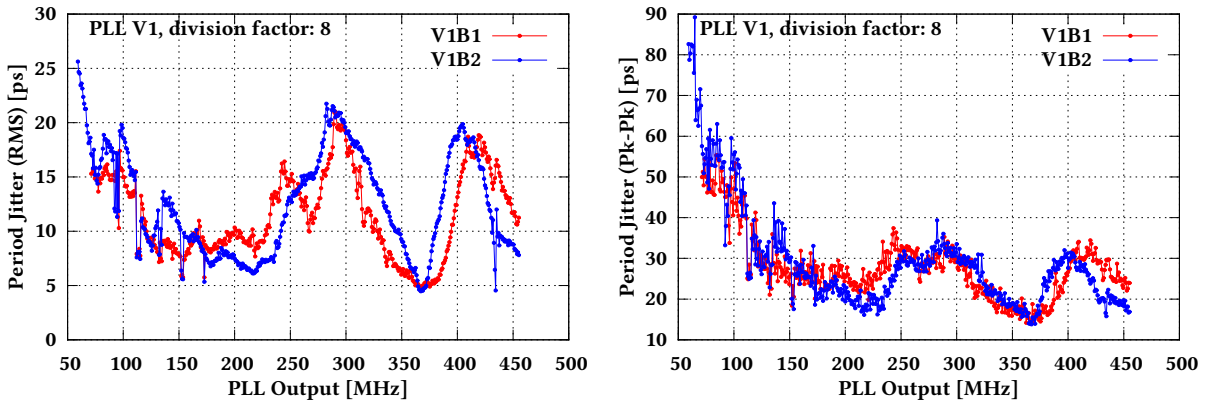


Figure B.9: Period jitter of the SALT_PLL_V1 for division factor 8.

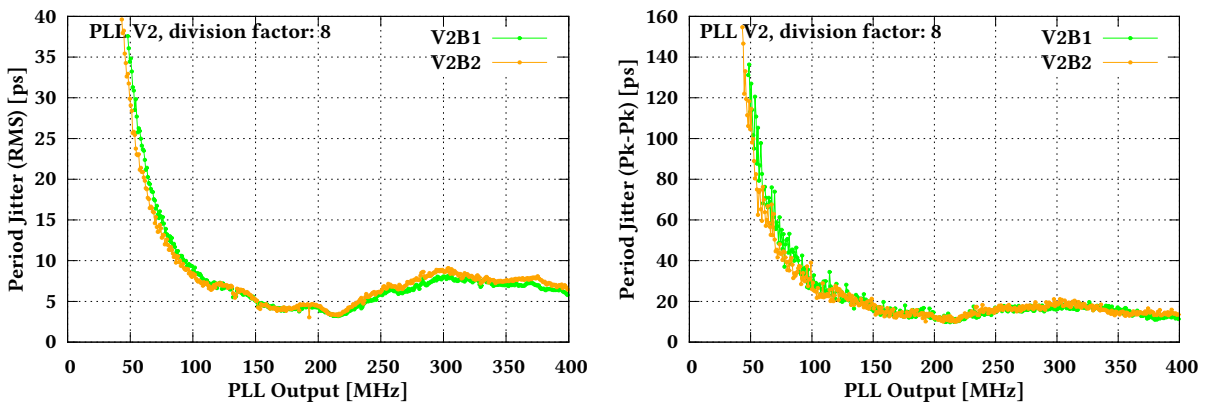


Figure B.10: Period jitter of the SALT_PLL_V2 for division factor 8.

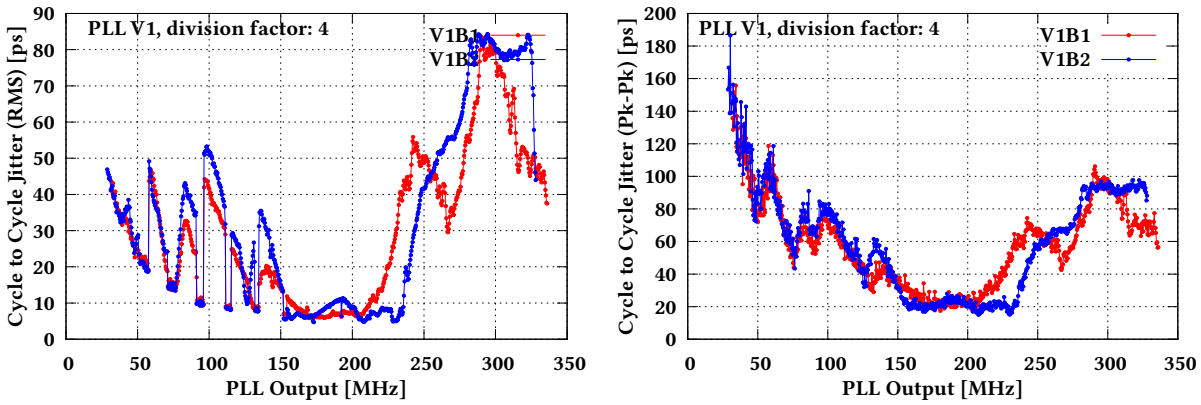


Figure B.11: Cycle to cycle jitter of the SALT_PLL_V1 for division factor 4.

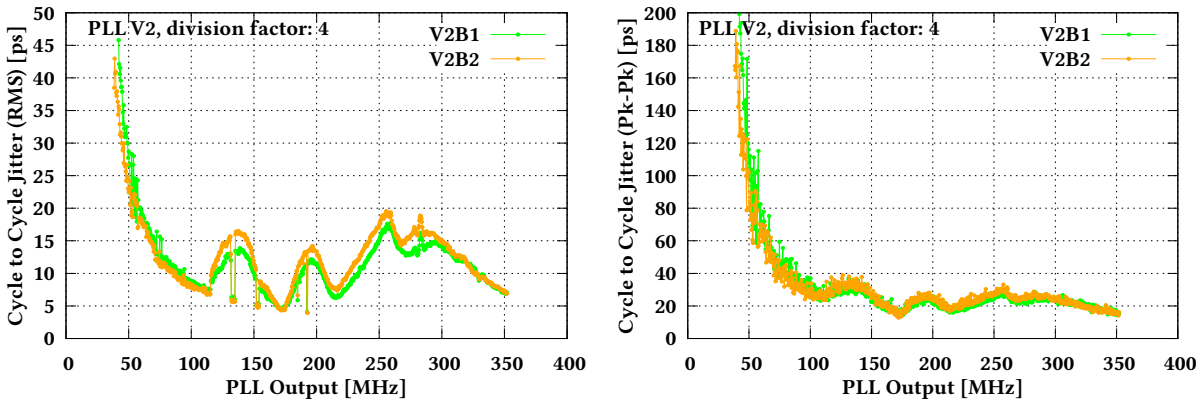


Figure B.12: Cycle to cycle jitter of the SALT_PLL_V2 for division factor 4.

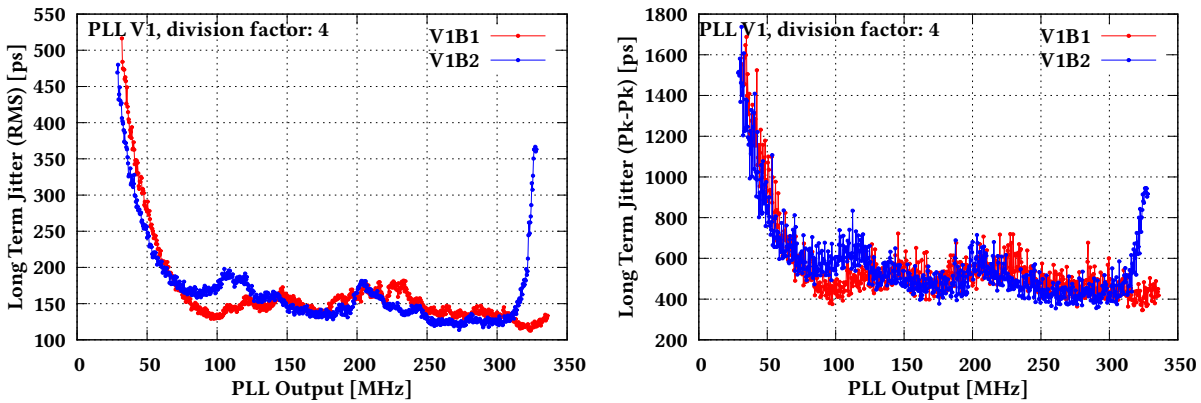


Figure B.13: Long term jitter of the SALT_PLL_V1 for division factor 4.

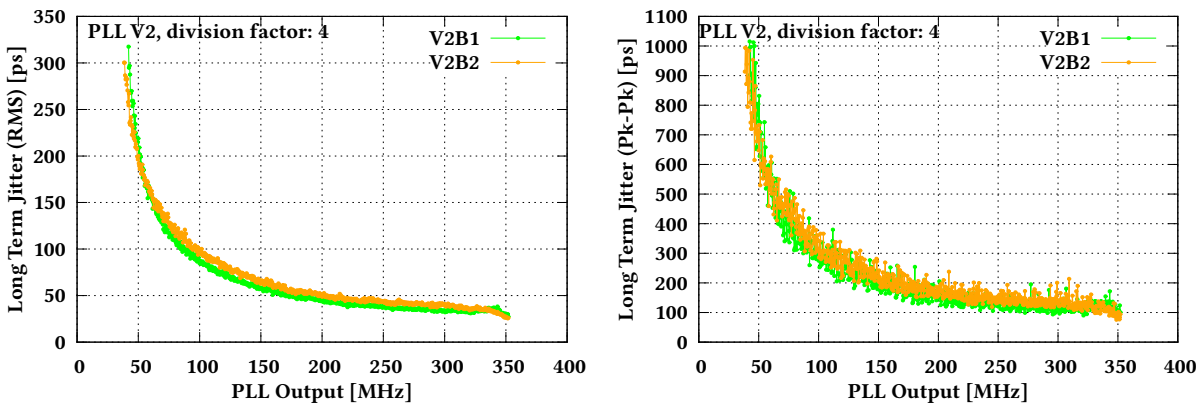


Figure B.14: Long term jitter of the SALT_PLL_V2 for division factor 4.

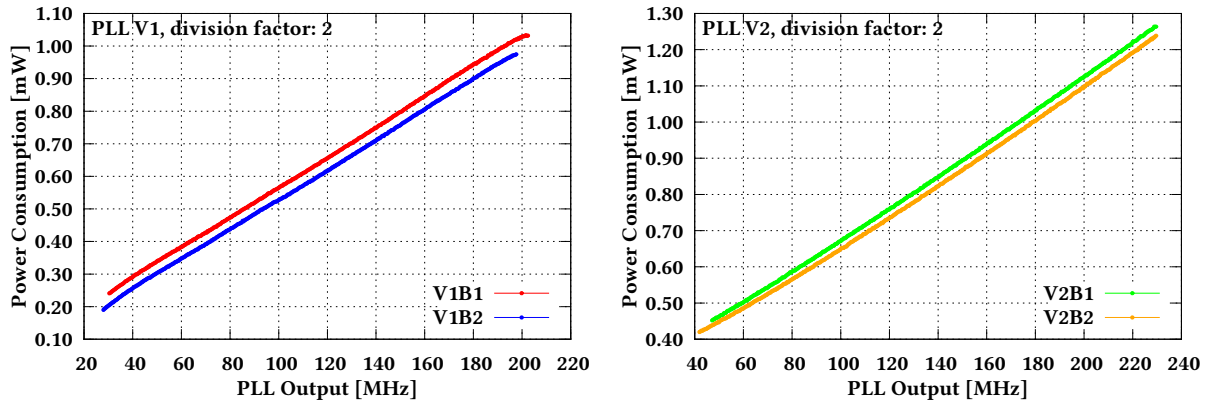


Figure B.15: Power consumption of the SALT_PLL_V1(left) and V2(right) for division factor 2.

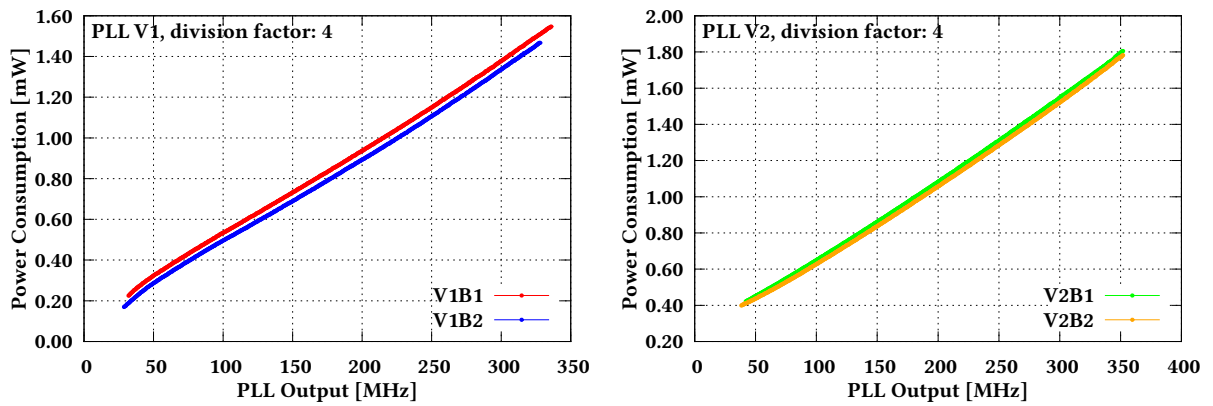


Figure B.16: Power consumption of the SALT_PLL_V1(left) and V2(right) for division factor 4.

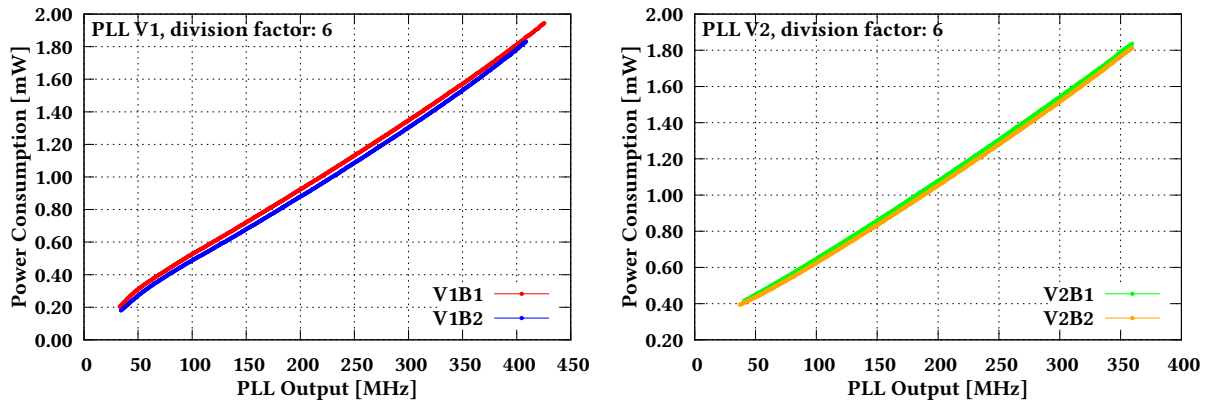


Figure B.17: Power consumption of the SALT_PLL_V1(left) and V2(right) for division factor 6.

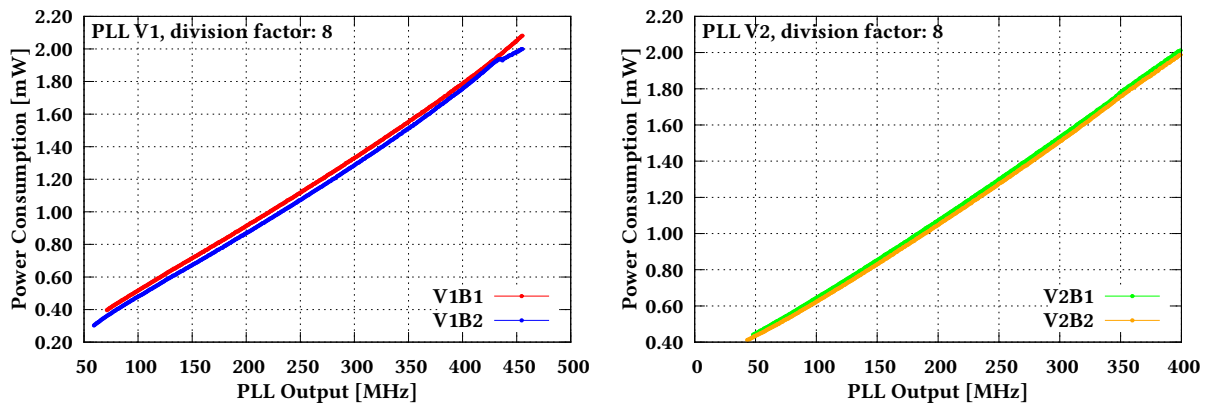


Figure B.18: Power consumption of the SALT_PLL_V1(left) and V2(right) for division factor 8.

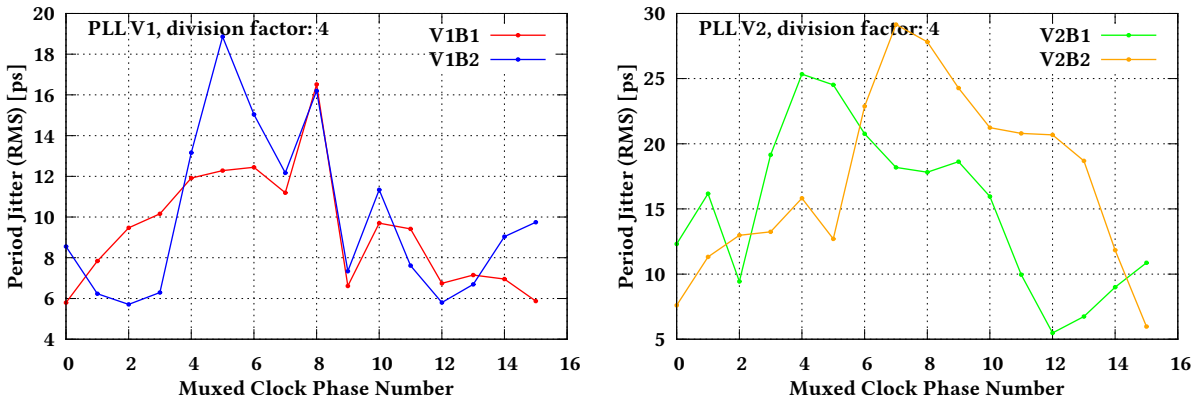


Figure B.19: Period jitter vs clock phase of the SALT_PLL_V1(left) and V2(right) for divider 4.

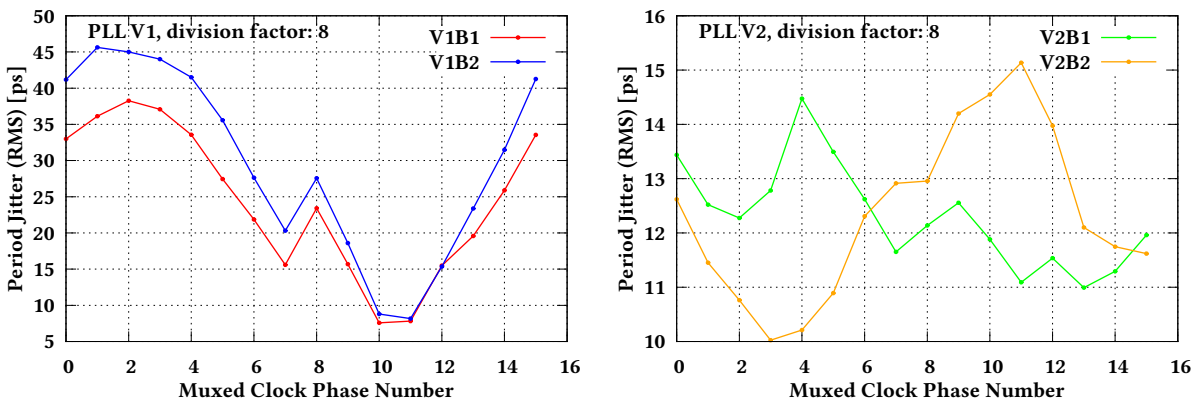


Figure B.20: Period jitter vs clock phase of the SALT_PLL_V1(left) and V2(right) for divider 8.

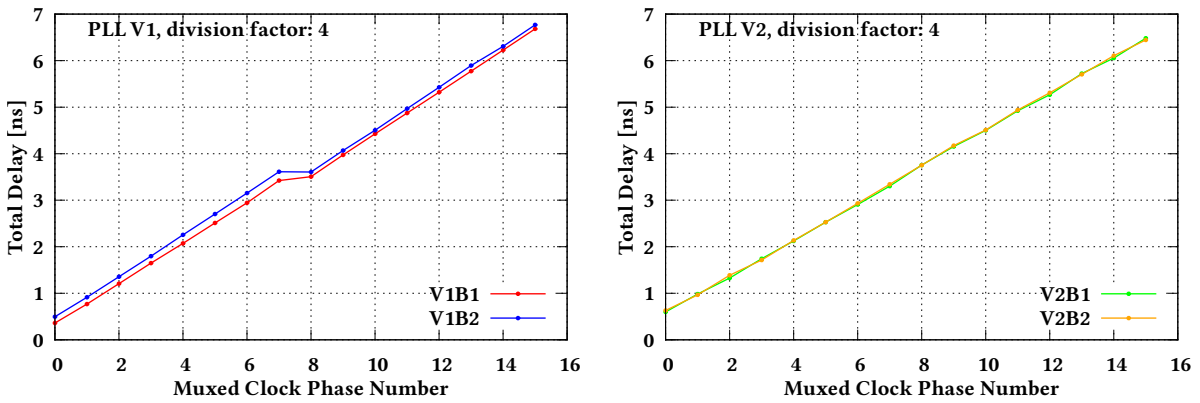


Figure B.21: Total delay vs clock phase of the SALT_PLL_V1(left) and V2(right) for divider 4.

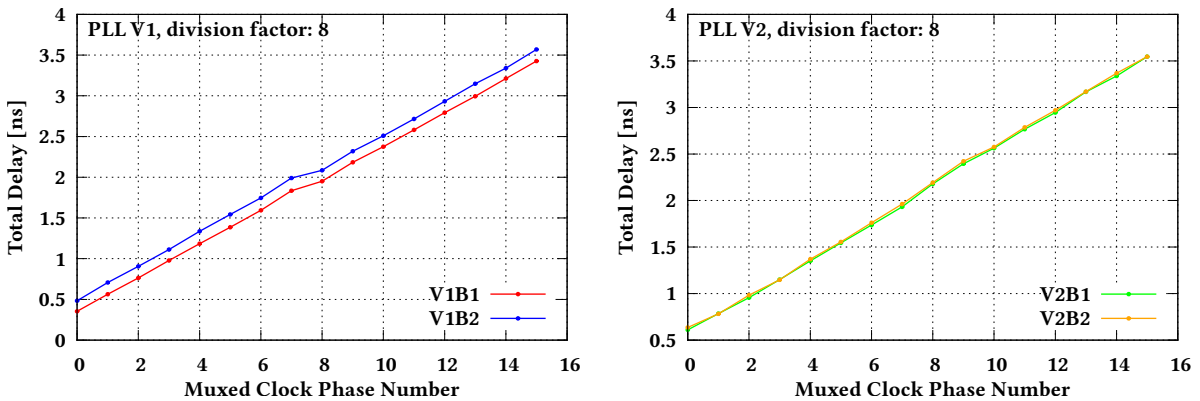


Figure B.22: Total delay vs clock phase of the SALT_PLL_V1(left) and V2(right) for divider 8.

Appendix C

Measurements results of SALT_DLL

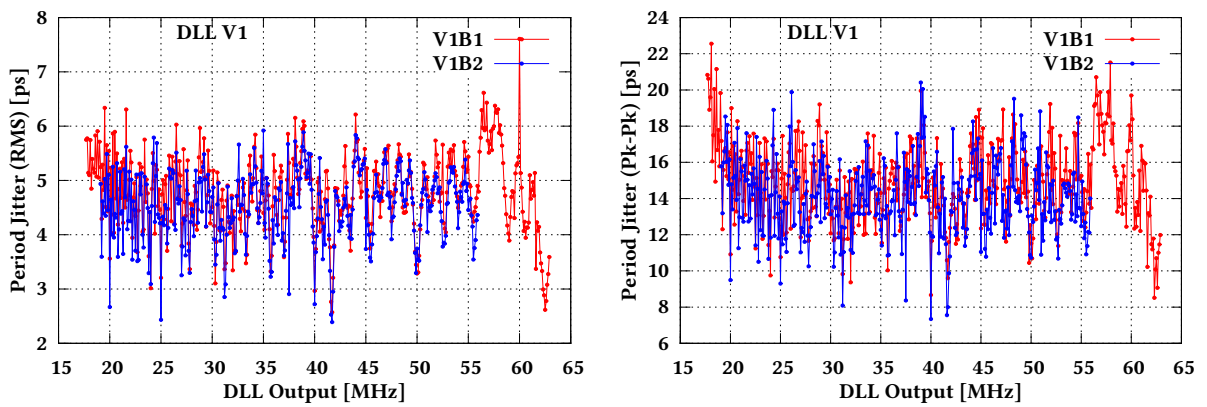


Figure C.1: Period jitter of the SALT_DLL_V1.

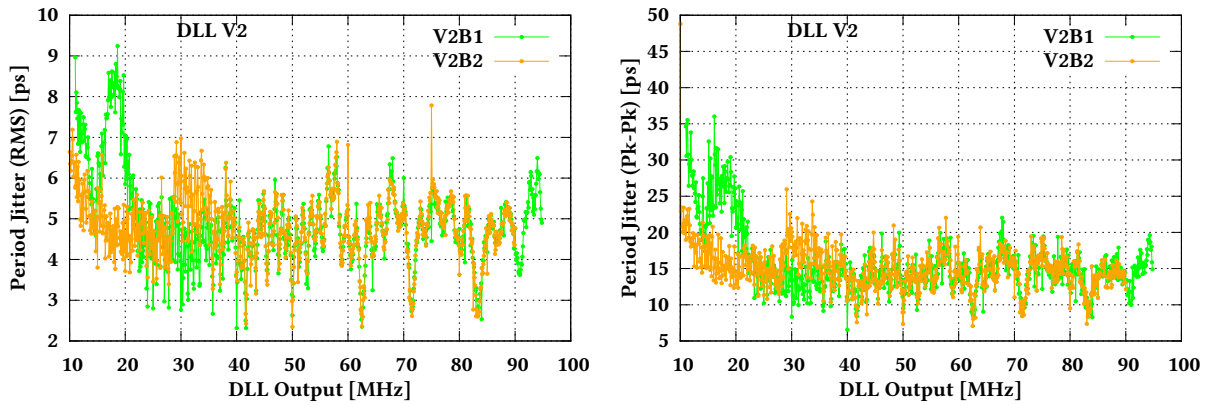


Figure C.2: Period jitter of the SALT_DLL_V2.

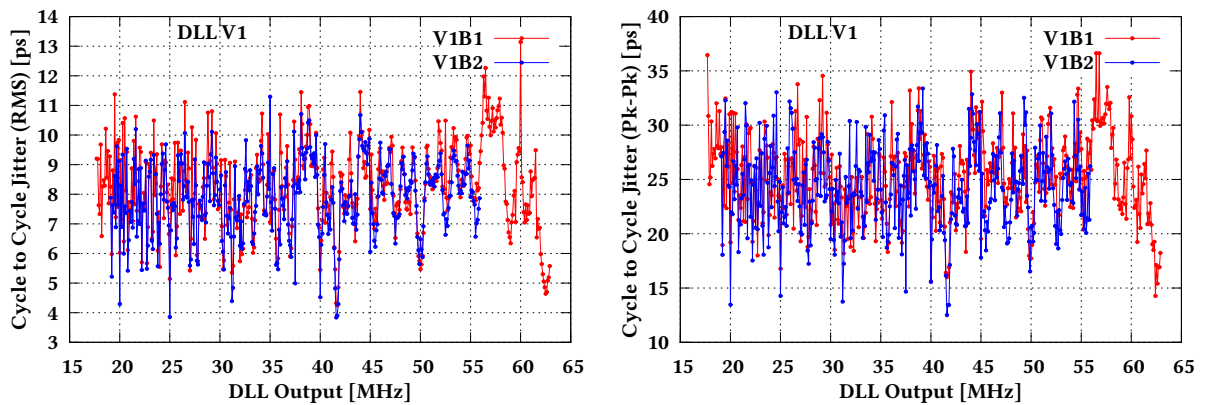


Figure C.3: Cycle to cycle jitter of the SALT_DLL_V1.

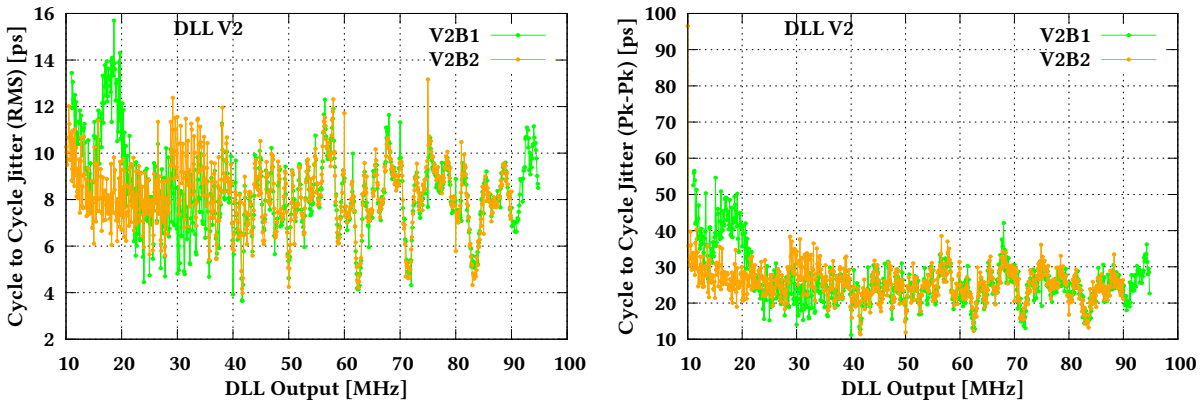


Figure C.4: Cycle to cycle jitter of the SALT_DLL_V2.

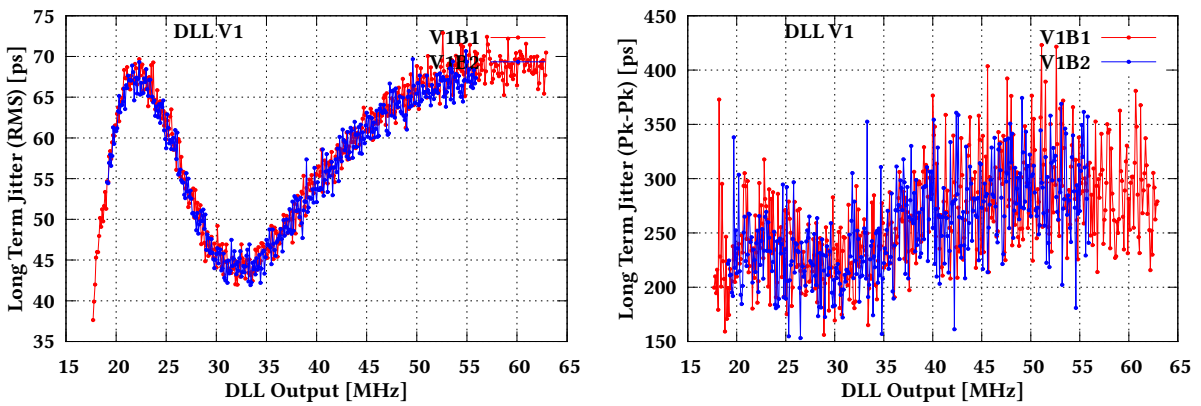


Figure C.5: Long term jitter of the SALT_DLL_V1.

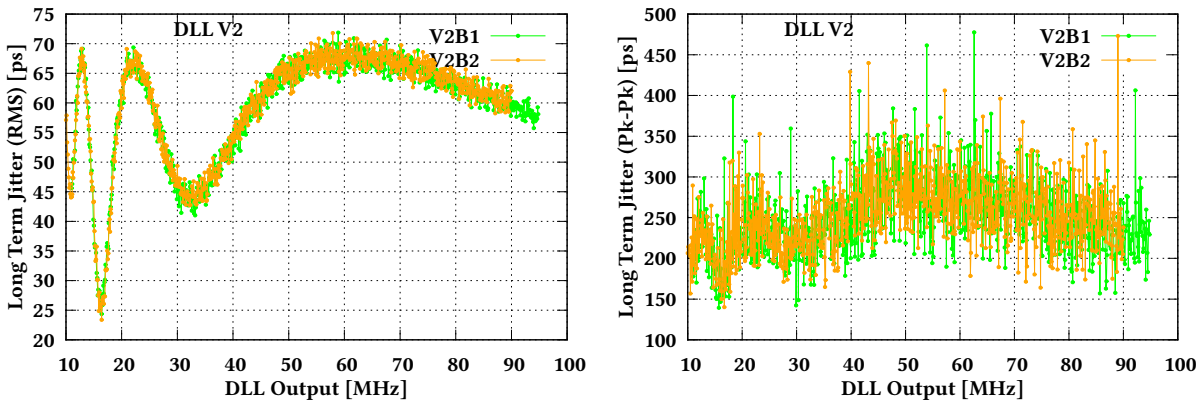


Figure C.6: Long term jitter of the SALT_DLL_V2.

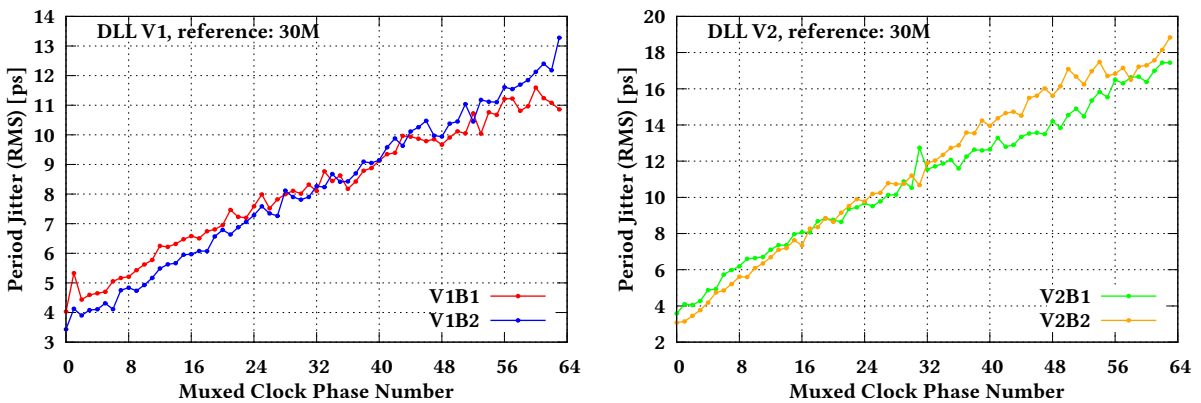


Figure C.7: Period jitter vs clock phase of the SALT_DLL_V1(left) and V2(right) at frequency 30MHz.

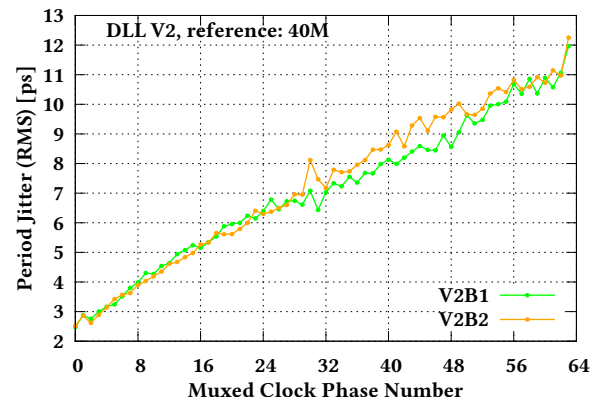
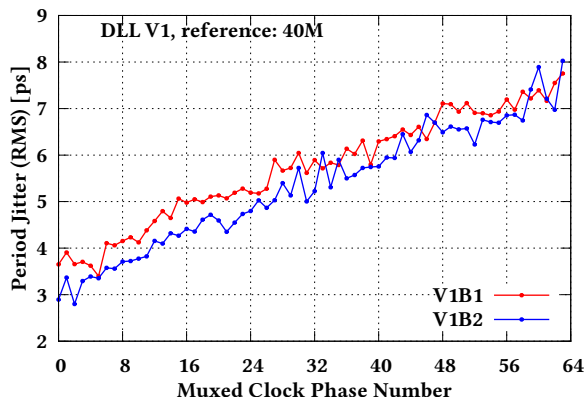


Figure C.8: Period jitter vs clock phase of the SALT_DLL_V1(left) and V2(right) at frequency 40MHz.

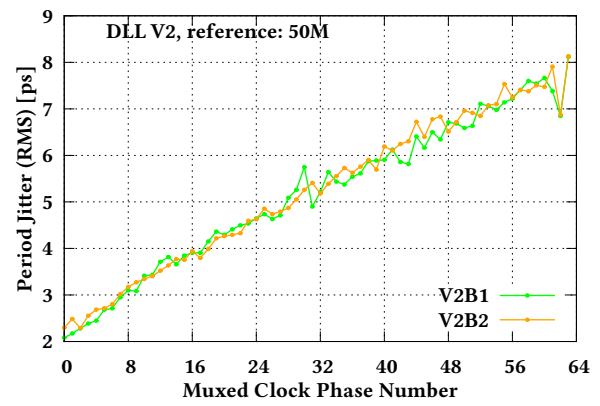
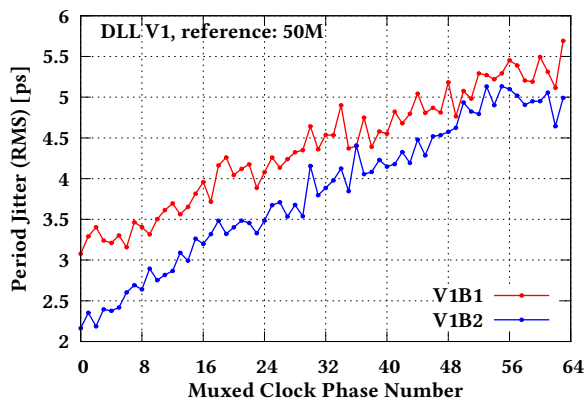


Figure C.9: Period jitter vs clock phase of the SALT_DLL_V1(left) and V2(right) at frequency 50MHz.

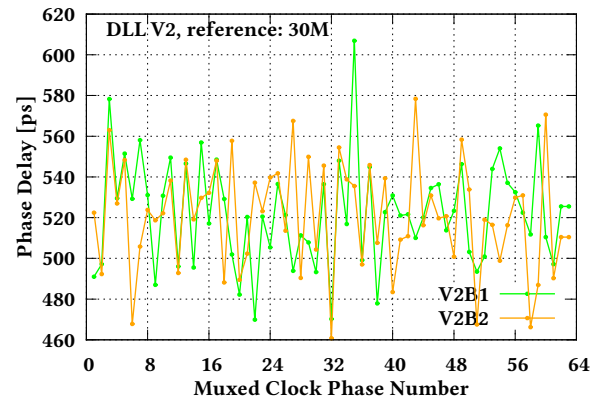
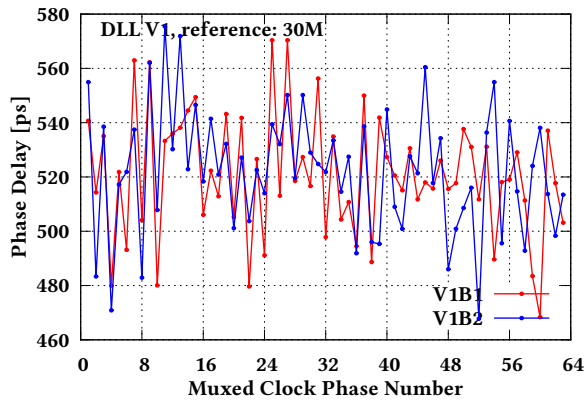


Figure C.10: Phase delay vs clock phase of the SALT_DLL_V1(left) and V2(right) at frequency 30MHz.

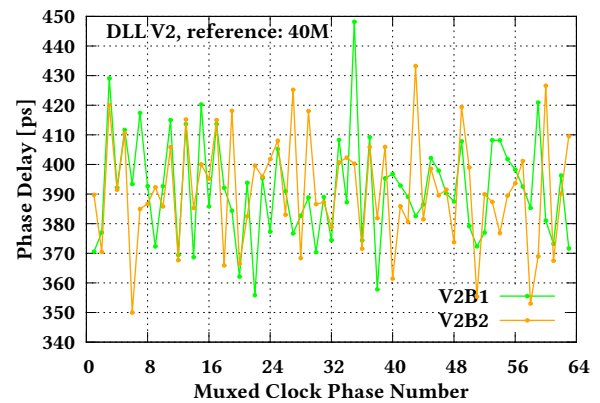
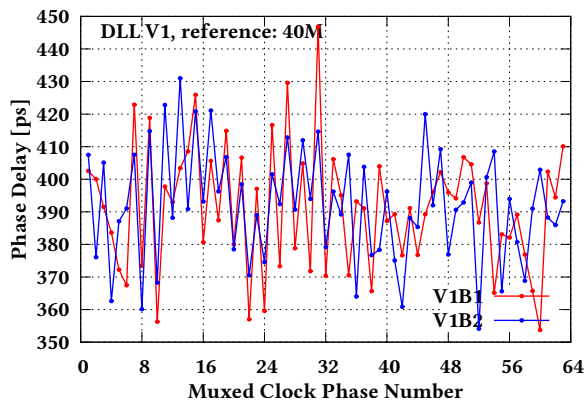


Figure C.11: Phase delay vs clock phase of the SALT_DLL_V1(left) and V2(right) at frequency 40MHz.

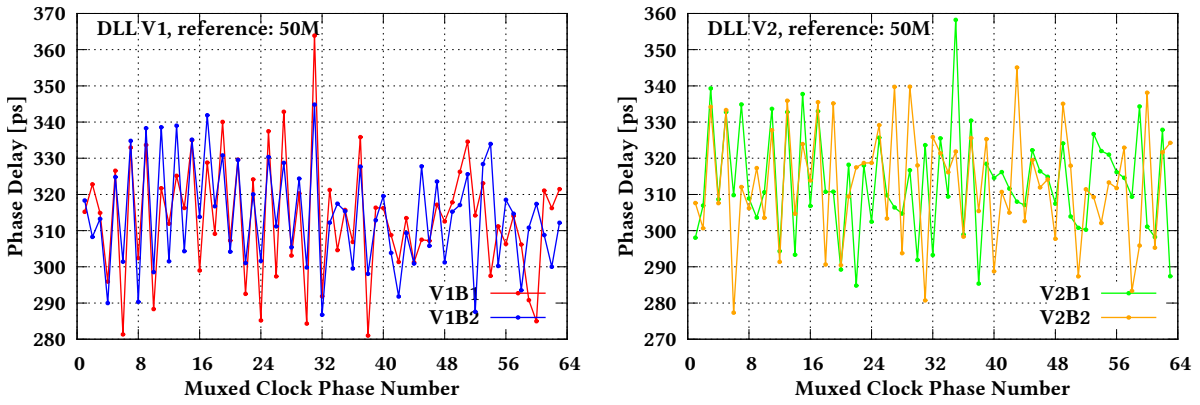


Figure C.12: Phase delay vs clock phase of the SALT_DLL_V1(left) and V2(right) at frequency 50MHz.

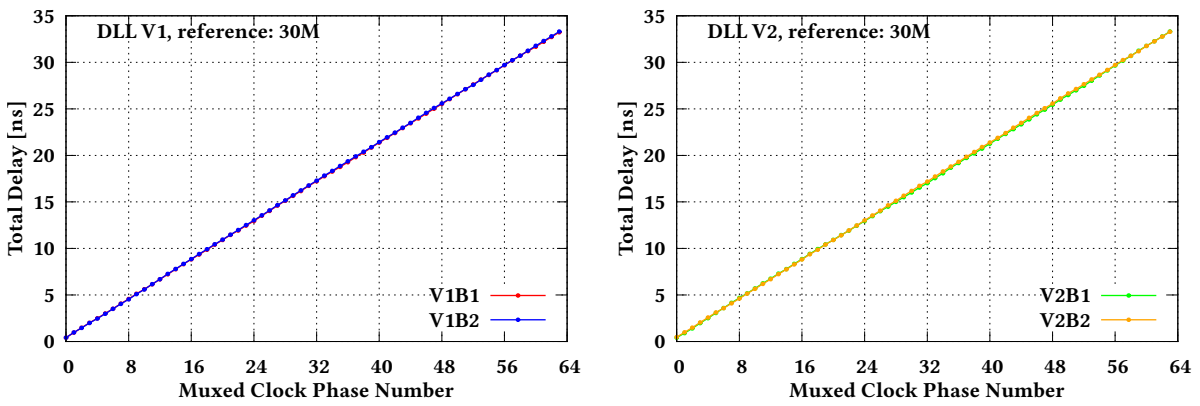


Figure C.13: Total delay vs clock phase of the SALT_DLL_V1(left) and V2(right) at frequency 30MHz.

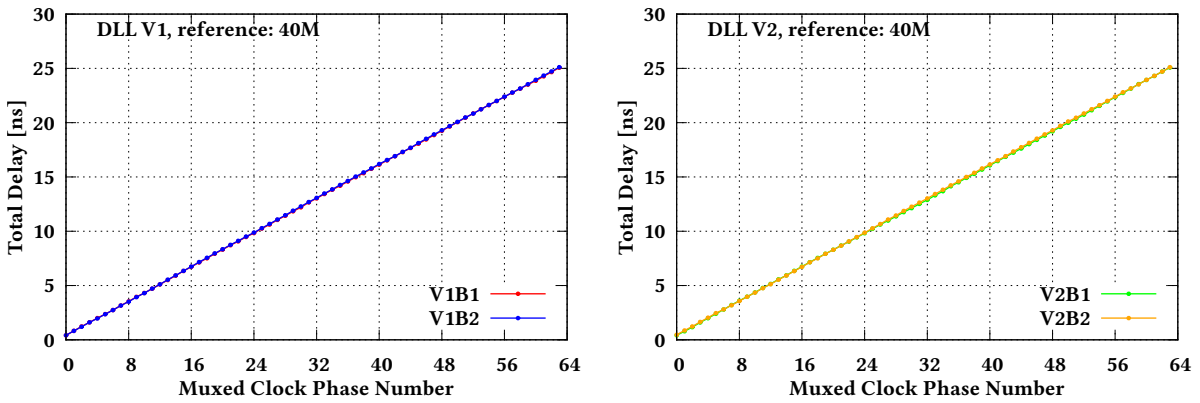


Figure C.14: Total delay vs clock phase of the SALT_DLL_V1(left) and V2(right) at frequency 40MHz.

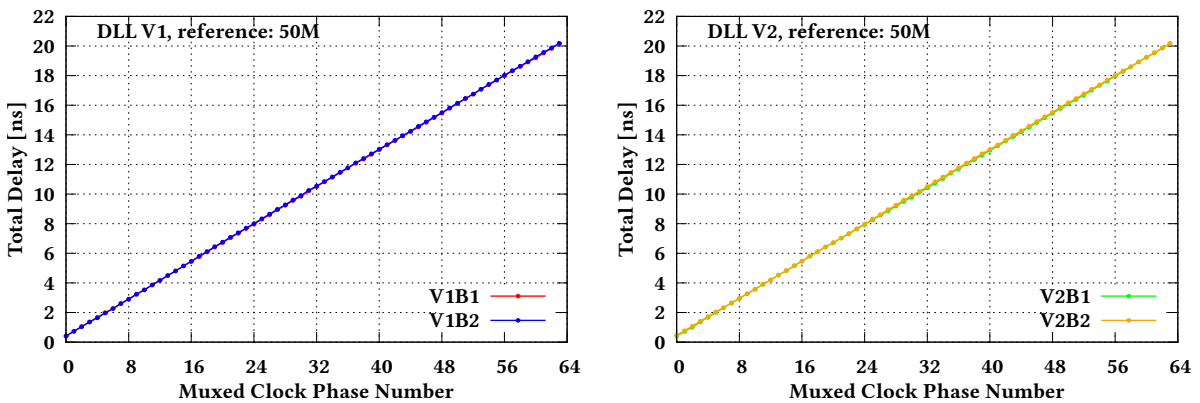


Figure C.15: Total delay vs clock phase of the SALT_DLL_V1(left) and V2(right) at frequency 50MHz.