UNIVERSITAT DE BARCELONA

On the digital design and verification of pixel detector ASICs for fast timing applications and other fields of science

Núria Egidos Plaja

Tesi doctoral

On the digital design and verification of pixel detector ASICs for fast timing applications and other fields of science

Autora: Núria Egidos

Directors: Dr. Rafael Ballabriga (CERN), Dr. Joan Mauricio (UB)

UNIVERSITAT DE BARCELONA

On the digital design and verification of pixel detector ASICs for fast timing applications and other fields of science

Programa de doctorat en Enginyeria i Ciències Aplicades

Autora: Núria Egidos

Directors: Dr. Rafael Ballabriga (CERN), Dr. Joan Mauricio (UB)

Tutor: Dr. Atilà Herms (UB)

CERN, Suïssa

UNIVERSITAT DE BARCELONA

UNIVERSITAT DE BARCELONA

Abstract

On the digital design and verification of pixel detector ASICs for fast timing applications and other fields of science

by Núria EGIDOS

Massive production of integrated circuits of increasing complexity and large area requires the usage of digital design and verification tools and methodologies to increase the reliability of designs, ease the scalability of projects and automate testing procedures. This work is focused on the design and verification of digital circuits implemented in pixel detector readout chips. The main contribution consists of the design, implementation and verification, by means of digital tools, of a Clock Distribution Network (CDN) for FastICpix, a single photon hybrid pixel detector. This network delivers a time reference of low frequency (tens of MHz) to the pixel matrix, a master clock for the timestamp mechanism that time tags the incoming photons. FastICpix adapts in area and pixel pitch to the application in order to optimise the charge collection, and it supports a fine Single Photon Time Resolution (SPTR) in the order of 10 ps. To fulfil the aforementioned requirements, the network can be adapted to the chip area and pixel pitch, and it supports a fine adjustment (20 ps resolution) of the master clock phase across the pixel matrix.

The proposed design is not available on silicon yet, but back-annotated digital simulations of the post-layout netlist of the network (implemented in a 65-nm process) are presented. These simulations are back-annotated with the propagation delays associated to parasitic capacitances and resistances. The simulations correspond to the most complex scenario, the largest chip area (3×3 cm^2), in which there are more contributions to the time errors and thus poses the greatest challenge to fulfil the timing requirements of the network. The selected architecture can achieve the required time resolution in all the Process, Voltage and Temperature (PVT) corners considered. The estimated power consumption of the network is not the dominant contribution to the overall chip consumption. Guidelines on how to scale this design to the rest of envisaged FastICpix geometries are provided.

On the other hand, a verification framework based on the Universal Verification Methodology (UVM) has been implemented for the CLIC Tracker Detector (CLICTD), a monolithic pixelated sensor and readout chip aimed at the silicon tracker for the Compact Linear Collider (CLIC) experiment, which has been fabricated in a modified 180-nm CMOS imaging process. The exhaustive and automated verification enabled spotting minor bugs during the design, which enabled the successful operation of the chip once fabricated.

UNIVERSITAT DE BARCELONA

Resumen

Diseño y verificación de circuitos digitales en ASICs para detectores píxel destinados a aplicaciones de gran resolución temporal y otros campos científicos

por Núria EGIDOS

La producción en masa de circuitos integrados de gran complejidad y área requiere el uso de herramientas y metodologías de diseño y verificación digital, con el propósito de mejorar la fiabilidad de los diseños, favorecer la escalabilidad de los proyectos y automatizar los procedimientos de prueba. Este trabajo se centra en el diseño y verificación de circuitos digitales implementados en chips de lectura de detectores píxel.

La contribución principal consiste en el diseño, implementación y verificación, mediante el uso de herramientas digitales, de una red de distribución de reloj (en inglés, Clock Distribution Network o CDN) para FastICpix, un detector píxel híbrido capaz de procesar fotones individualmente. Esta red distribuye una referencia temporal de baja frecuencia (decenas de MHz) a la matriz de píxeles, un reloj que se usa en el mecanismo de etiquetado temporal de la llegada de fotones.

FastICpix se adapta en área y tamaño de píxel para optimizar la captura de carga eléctrica según la aplicación, y proporciona una fina resolución temporal en la detección de fotones individuales (Single Photon Time Resolution o SPTR del orden de 10 ps). Para cumplir estos requisitos, la red se puede escalar en área y adaptar al tamaño del píxel; y proporciona un ajuste fino de fase (resolución de 20 ps) en la distribución del reloj.

Aunque el diseño que se propone no ha sido fabricado en silicio por el momento, se presentan simulaciones digitales anotadas con los retrasos de propagación asociados a las capacidades y resistencias parásitas presentes en el circuito, que ha sido implementado en un proceso de fabricación de 65nm. Estas simulaciones corresponden al escenario más complejo, el chip de mayor área (3×3 cm^2), ya que en este caso hay un mayor número de contribuciones a los errores temporales y, por tanto, supone el mayor desafío para obtener la resolución temporal necesaria en la red. La arquitectura seleccionada cumple con los requisitos de resolución temporal bajo todas las condiciones de variación de Proceso, Voltaje y Temperatura (PVT) consideradas, y el consumo de potencia estimado de la red no es la contribución dominante en el consumo total del chip. Se proporcionan pautas para escalar este diseño al resto de geometrías contempladas en el proyecto FastICpix.

Por otro lado, también se ha implementado una estructura de verificación, basada en la Metodología Universal de Verificación (UVM por las siglas en inglés, Universal Verification Methodology), para el Detector de Trayectorias de CLIC (CLIC Tracker Detector o CLICTD), un sensor monolítico segmentado y chip de lectura destinado al detector de silicio de trayectorias para el experimento Colisionador Linear Compacto (Compact Linear Collider o CLIC). Este chip ha sido fabricado en un proceso de imagen CMOS de 180nm modificado. La aplicación de esta verificación exhaustiva y automatizada permitió corregir pequeños errores de diseño, lo cual contribuyó a la exitosa operación del chip una vez fabricado.

UNIVERSITAT DE BARCELONA

Resum

Disseny i verificació de circuits digitals en ASICs per detectors píxel destinats a aplicacions de gran resolució temporal i d'altres camps científics

per Núria EGIDOS

La producció en massa de circuits integrats de gran complexitat i àrea requereix l'ús d'eines i metodologies de disseny i verificació digital, amb el propòsit de millorar la fiabilitat dels dissenys, afavorir l'escalabilitat dels projectes i automatitzar els procediments de prova. Aquest treball està centrat en el disseny i verificació de circuits digitals implementats en xips de lectura de detectors píxel.

La contribució principal consisteix en el disseny, implementació i verificació, mitjançant l'ús d'eines digitals, d'una xarxa de distribució de rellotge (en anglès, Clock Distribution Network o CDN) per a FastICpix, un detector píxel híbrid que processa fotons individualment. Aquesta xarxa distribueix una referència temporal de baixa freqüència (desenes de MHz) a la matriu de píxels, un rellotge que s'empra al mecanisme d'etiquetatge temporal de l'arribada de fotons.

FastICpix s'adapta en àrea i mida del píxel per optimitzar la captura de càrrega elèctrica segons l'aplicació, i proporciona una fina resolució temporal en la detecció de fotons individuals (Single Photon Time Resolution o SPTR de l'ordre de 10 ps). Per tal de complir aquests requisits, la xarxa es pot escalar en àrea i adaptar a la mida del píxel; i proporciona un ajustament fi de la fase (resolució de 20 ps) en la distribució del rellotge.

Tot i que el disseny que es proposa no ha sigut fabricat en silici encara, es presenten simulacions digitals anotades amb els temps de propagació associats a les capacitats i resistències paràsites presents al circuit, que s'ha implementat en un procés de fabricació de 65nm.

Aquestes simulacions corresponen a l'escenari més complex, el xip de major àrea (3×3 cm^2), ja què en aquest cas hi ha més contribucions als errors temporals i, per tant, suposa el desafiament més gran a l'hora d'obtenir la resolució temporal necessària a la xarxa. L'arquitectura seleccionada compleix els requisits de resolució temporal en totes les condicions de variació de Procés, Voltatge i Temperatura (PVT) que s'ha considerat, i el consum de potència estimat de la xarxa no és la contribució dominant al total del consum del xip. Es proporcionen pautes per escalar aquest disseny a la resta de geometries previstes pel projecte FastICpix.

D'altra banda, també s'ha implementat una estructura de verificació, basada en la Metodologia de Verificació Universal (UVM per les sigles en anglès, Universal Verification Methodology), pel Detector de Trajectòries de CLIC (CLIC Tracker Detector o CLICTD), un sensor monolític segmentat i xip de lectura destinat al detector de silici de trajectòries per l'experiment Col·lisionador Linear Compacte (Compact Linear Collider o CLIC). Aquest xip s'ha fabricat en un procés d'imatge CMOS de 180nm modificat. L'aplicació d'aquesta verificació exhaustiva i automatitzada va permetre corregir petits errors de disseny, la qual cosa va contribuir a l'exitosa operació del xip un cop fabricat.

Acknowledgements

I would like to thank my advisors Dr. Rafael BALLABRIGA (CERN) and Dr. Joan MAURICIO (UB) for their mentoring and support during these years. Their advice and experience has been essential not only in the context of technical discussions, but also towards the integration in the team, research dynamics and professional growth as an engineer.

My gratitude as well to Dr. David GASCÓN (UB), Dr. Michael CAMPBELL (CERN), Dr. Dominik DANNHEIM (CERN) and Dr. Lucie LINSSEN (CERN) for their constant encouragement and for making countless professional, training, dissemination and outreach opportunities possible.

The successful completion of this work would not have been possible without the guidance (and patience) of Dr. Xavier LLOPART, Dr. Tuomas POIKELA, Dr. Iraklis KREMASTIOTIS, Adrian FIERGOLSKI, Dr. Sara MARCONI, Dr. Edinei SANTIN and the passionate, very helpful indications of Wojciech BIALAS. Thanks to all the Medipix team members for the cheerful work environment.

This work has been supported by the ATTRACT project funded by the EC under Grant Agreement 777222 and by the CLICdp collaboration.

Thanks to Jose for everything we have been through together during these years, and to our families for their warm encouragement and caring, especially in these last times. Thanks as well to Franco and Ivan for the casual laughter and the simple, calm and wonderful joy of sharing these days and evenings with you, and those that will come. David, Rafel, Cristian, Sergio, Arnau, Diallo, Vova, Mohammad, Sandra, Stefano, Carla, Alejandro... a huge thank you for your help and friendship. Ignacio, for bringing back the inspiration and shaking what was established. Roger, for such a generous and warm support from the distance. Elena and Elizabeth, for helping us every single time. Walter, Jorgen, Moritz, Samuele, Viros and many others for being kind and for your indispensable help. And to all of those whose name I forgot to include, thanks for being part of my life.

Contents

List of Figures

List of Tables

Part I

INTRODUCTION

Radiation detection has become an indispensable tool for a wide range of applications [71] [24] [20], including medical imaging (plain X-ray radiography, X-ray contrast examination, X-ray Computer Tomography (CT), Positron Emission Tomography (PET), etc.); dosimetry or measurement of the quantity of radiation (quality control in radiation therapy and radiation sterilization of food and medical appliances, exposure to radiation during medical examination, measurement of natural radiation sources, etc.); material characterization or structural analysis using synchrotron X-ray [56]; security applications of industrial radiography (X-ray scanners for luggage and shipping containers); art authentication [55]; and particle identification in High Energy Physics (HEP) [126], amongst others.

Radiation interacts with the atoms of the detector and transfers energy to them. The transfer of energy has two effects: ionization and excitation. Ionising radiation has a high enough energy to free an orbital electron from the atom (photoelectric effect), thus creating an ion pair (a negatively charged electron and a positively charged atom). The resulting electrons can be collected in the form of ionization-induced currents, an electrical magnitude that is sensed, amplified and processed to characterise the interaction of such a radiation with matter.
When excitation occurs, electrons are perturbed from their normal arrangement in the atom by acquiring a higher energy level (the atom acquires an excited state). When the atom is de-excited, a part of the energy released occurs in the form of photons, which can lead to further interactions [23].

Detectors can be classified according to different criteria [23] [54] [20]: the medium in which the interaction takes place; whether direct or indirect conversion is performed; and whether the detector is operated in integrating or photon counting mode.

According to the medium in which the interaction takes place, these are some of the available options:

- Gas-filled detectors: a volume of gas is contained between two electrodes that have a voltage difference (and thus an electric field) between them. In the absence of radiation, the gas is an insulator and no electrical current flows between the electrodes. Under the presence of ionising radiation, electrons produced by ionization cause a momentary flow of a small amount of electrical current between the electrodes, which is the variable to be sensed.
  These devices offer a low detection efficiency for X and γ rays, so they are not suitable for medical imaging applications; instead, they are preferred for dose calibration and particle identification [31].

- Scintillation detectors: when the ionised atoms undergo a recombination or when the excited atoms are de-excited, most of the energy that is released is dissipated in the form of thermal energy (molecular vibrations in gases and liquids or lattice vibrations in a crystal). In scintillator crystals, a portion of the released energy occurs in the form of light (visible and UV). The amount of light produced by a quantum of radiation (typically few hundred to a few thousand photons) is proportional to the amount of energy deposited by such a radiation, and thus it is the magnitude to be sensed.

The resulting light can be translated to an electrical current by means of Photo-Multiplier Tubes (PMTs), silicon Avalanche Photo-Diodes (APDs) or APDs grouped into matrices (Silicon Photo-Multiplier (SiPM)).

- Semiconductor detectors: the most commonly used elemental semiconductor materials for particle detectors are silicon (Si) and germanium (Ge), which are doped to compose a p-n junction [121]. A depletion zone is formed in the interface between the p and n regions, in which there is a relative absence of free charges. The device is designed and polarised to maximise the volume of the depletion zone for a faster and safer (lower risk of charge trapping) signal collection.
  Analogously to the gas-filled detectors, the interaction with radiation will lead to the generation of electron-hole pairs. If the free charges are generated in the depletion zone, they are swept by the electric field present in that zone and rapidly directed to the terminals of the device, where the resulting current is sensed. The properties (shape, magnitude...) of the resulting electrical signal are proportional to the amount of radiation energy absorbed.
  The active medium in semiconductor detectors is 2000 to 5000 times denser than in gaseous detectors, which means that the first have a much more efficient detection of X and γ rays than the second, thus making them more suitable for the applications that concern the present work.
  In addition, the ionisation potential (energy required to produce a free electron) of semiconductors is about 10 times smaller than in gases, which has a two-fold interpretation: a) in semiconductor detectors, a certain amount of energy absorbed produces an electrical signal that is approximately 10 times larger than in a gaseous detector; or b) approximately 10 times a lower energy deposition is required in semiconductor detectors to yield the same electrical signal as a gaseous detector.
  The consequence of both observations is that semiconductor detectors offer a greater sensitivity and thus better resolution than gaseous detectors, or a lower radiation dose is required to provide the same resolution.
  APDs, SiPMs and Monolithic Active Pixel Sensors (MAPS) are some examples of semiconductor detectors.

In an indirect conversion detector, such as scintillators, the interaction with radiation results in the generation of light photons (the number of photons generated is proportional to the energy of the incoming radiation). In a second stage, light is translated to an electrical current by means of photosensitive devices, such as PMTs, APDs or SiPMs.
In a direct conversion detector, radiation leads directly to the generation of charge carriers, as in the case of semiconductor detectors.
In general terms, direct detection provides a higher energy resolution and Signal-to-Noise Ratio (SNR), and thus yields a higher sensitivity for a specific energy of interest. Besides, these detectors are more compact, since they do not require the connection to a photosensor [101].
On the other hand, indirect conversion detectors offer several advantages: they are more cost-effective and simpler to implement, offer a slightly faster response, need no bias supply, etc.

In energy integrating mode, the photons arriving at the detector during a finite acquisition window are accumulated to provide one resulting signal, in which the

contributions of the individual photons cannot be distinguished. This is the case of gas detectors, in which the charge resulting from several interactions is accumulated to provide a signal that is large enough to be reliably sensed; and it has been the traditional operation mode of radiation detectors during the 20th century, from photographic plates in the early decades to Charged-Coupled Devices (CCDs).
In contraposition, counting systems are capable of discriminating individual photons by recording their energy and the location where they interact with the detector. This provides a better image resolution and enables using a lower radiation dose.
Furthermore, it opens the door to *colour* or spectroscopic X-ray imaging: on the one hand, the signal recorded from different photon energies (or, analogously, different wavelengths or colours) can be processed individually with photon counting detectors; on the other hand, the interaction of X-rays with matter depends on the energy of the radiation and on the nature of such a matter. As a result of these two observations, from a single exposure to a polychromatic X-ray source, photon counting detectors can retrieve information on the composition of the sample to be imaged.
Some applications that benefit from this technology are tissue identification in medical imaging, structural analysis or material characterization, and detection of illegal substances in luggage, amongst others [35] [19].

This work is focused on semiconductor pixel detectors used in photon counting mode (referring to the aforementioned classification between photon counting or integrating devices), particularised for two applications: the CLICTD chip, a monolithic pixel detector aimed at the Tracking Detector of the Compact Linear Collider (CLIC) experiment at CERN; and the FastICpix hybrid pixel detector chip, an ATTRACT proposal that pursues time stamping the incoming photons with a very fine time resolution, while providing the flexibility to adapt certain geometrical chip parameters to the application. The main contribution of this work is the development of a CDN for the FastICpix timestamp mechanism.

The CLICTD chip has been fabricated in a 180 nm CMOS process with 1.8 V supply, while the FastICpix chip is planned to be fabricated in a 65 nm CMOS process with 1.2 V as supply.
The photosensitive area of both chips is divided into pixels or image units; each pixel includes an electrode [1] where the charge originated by the incoming radiation is collected, which provides a fine granularity or spatial resolution to sample such a radiation. In the case of CLICTD, the readout electronics in charge of processing the collected signal is manufactured on the same substrate; in the case of FastICpix, the sentitive layer is bump-bonded to the substrate that hosts the readout electronics.

CLICTD is a test vehicle to conduct detector research: the doping profile of the sensor enhances the detection efficiency; the technological choice enables a low-mass and cost-effective solution to cover the large areas envisaged for its primary application, the CLIC silicon tracker; and the high density of transistors per pixel (compared to the usual configuration with less than 30 transistors per pixel [127]) enables in-pixel signal processing, which reduces the required complexity of the logic in charge of building the serial readout signal. In addition, a segmented channel architecture[1] was selected to have a pixel large enough to allow for the logic placement/routing, while maintaining a small detector capacitance.

[1] In the case of CLICTD, 8 collection electrodes share the same pixel logic.

The total sensitive area is 4.8 × 3.84 mm^2, and it is divided into a matrix of 16 × 128 detecting cells, each measuring 300 × 30 µm^2.
The in-pixel logic, as well as the peripheral logic in charge of interacting with the user and the various protocols involved (serial readout, I^{2}C) required an exhaustive verification. In this work, a verification framework based on UVM is proposed, which enabled the simulation of a wide range of scenarios of operation, building reusable test code and handling the result collection automatically. In more detail, the contributions on this topic include:

- Identification of the CLICTD features and functionalities to be verified, proposal of a verification plan (series of tests to perform).
- Implementation of a verification framework based on UVM and execution of the proposed tests and collection of results.

The second and main topic addressed in this work is the development of a CDN for the timestamp mechanism of FastICpix [42]. This single-photon hybrid pixel detector pursues a) the optimization of signal collection by tailoring the chip area and pixel pitch to the application, so as to enhance the signal collection and spatial resolution, and b) a very fine Single Photon Time Resolution (SPTR), in the order of 10 $\mathrm{ps_{rms}}$ (which motivates a 20 ps time bin in the TDCs used to time stamp the incoming photons).
These features imply that the CDN must be geometrically versatile, capable of adapting to areas ranging from few square millimeters to few square centimeters, and pixel pitch values that can range from tens to hundreds of micrometers. In parallel to these constraints, the time errors in the delivery of the clock must be bound to 20 ps.
Matching both requirements individually poses already a challenge, which escalates by having to fulfill them simultaneously (CDNs are usually optimised for a particular geometry precisely to pursue low time errors). The contributions on the design, implementation and simulation of a such a versatile CDN include:

- Proposal of a network architecture that fulfills the FastICpix requirements, implementation and characterization of the main components.
- Proposal of a phase detector architecture with a very fine time resolution.
- Proposal of a control strategy that minimises the static time errors of the network.
- Formulation of guidelines to scale the network architecture with the chip area and pixel pitch.

In the remaining chapters of this part, further details are provided on the structure and operation of pixel detectors operating in photon counting mode (Chapter 1), as well as on the future trends on the readout section of pixel detectors (Chapter 2).

The rest of the thesis is structured in two parts, one of them devoted to the verification of CLICTD, and the other consisting of the design and simulation of a CDN that is part of the timestamp mechanism of FastICpix.

The CLICTD verification is presented in Part II, including the following chapters:

- Chapter 3 provides an insight on the CLICTD chip and introduces the fundamentals of the UVM, with which it has been verified.
- Chapter 4 compiles the list of features to be verified;
- Chapter 5 is a description of the proposed UVM-based verification framework.
- Chapter 6 summarises the verification results.

The FastICpix CDN is addressed in Part III, which is further divided into:

- Chapter 7, which provides an introduction to the FastICpix project and the role of the CDN on the time resolution of the detector, including the network requirements.
- Chapter 8, a compendium of challenges to be addressed from the network perspective in order to achieve the required time resolution.
- Chapter 9, a revision of the existing network solutions and their suitability for the FastICpix scenario.
- Chapter 10 sets the simulation conditions that will be used on the following chapters.
- Chapter 11 is a detailed description of the proposed network architecture, including the structure of its components, implementation highlights and guidelines to scale the network architecture to arbitrary chip area and pixel pitch dimensions.
- Chapter 12 introduces the figures of merit used to characterise the network performance, and then extracts such figures from the simulation results, to evaluate whether the aforementioned requirements are honored.
- To conclude, Chapter 13 summarises the presented work and hints the future lines of research.

1 Semiconductor detection systems

A pixel detector system is composed of a semiconductor sensing layer, where the interaction with the incoming radiation occurs and the resulting currents are generated; and a readout electronics layer, in which the collected current is amplified, digitised and processed to yield a series of digital words.
These words contain information on the time of occurrence of the interaction (which can be used to track the trajectory of the particle), the energy of the incoming particle and the number of particles that arrived within a certain time window (shutter), for instance. These magnitudes can be used to reconstruct an image of the interaction of radiation with the detector, such as in medical imaging applications; or to identify the type of radiation, as in particle identification in HEP.

The sensing and the readout electronics layers can be manufactured on the same substrate, which results in a monolithic pixel detector; or in separate substrates, which yields a hybrid pixel detector.
In the following sections, the basic components and functionalities of a pixel detector are presented (Section 1.1); and the main differences and characteristics of both types of pixel detectors are introduced (Section 1.2).

1.1 Fundamental detection chain in the readout electronics of a pixel detector

The basic stages of the readout chain of a pixel detector, or the steps to process the charge collected from the sensing layer, are depicted in Figure 1.1 [119].
This chain can be divided into an analog front-end, which amplifies the collected signal and eventually digitises it; and the digital logic that processes such a signal to generate the words that contain the relevant information about the incoming radiation.

The first stage of the analog front-end is the preamplifier: since the collected signal can be as low as a few tens of attocoulombs, it must be amplified to be able to process it.
The sensed signal is subject to statistical fluctuations and is further corrupted by electronic noise. The main role of the next stage in the chain, the shaper, is to improve the SNR. The shaper acts as a filter that tailors the frequency content of the signal to favor the useful signal, while attenuating the noise.
Finally, the discriminator is in charge of digitising the resulting signal. To do so, the output of the shaper is compared to a reference level or threshold, which is higher than the electronic noise. The output of the discriminator will be high for as long as the magnitude of the pulse coming from the shaper is higher that the threshold level.

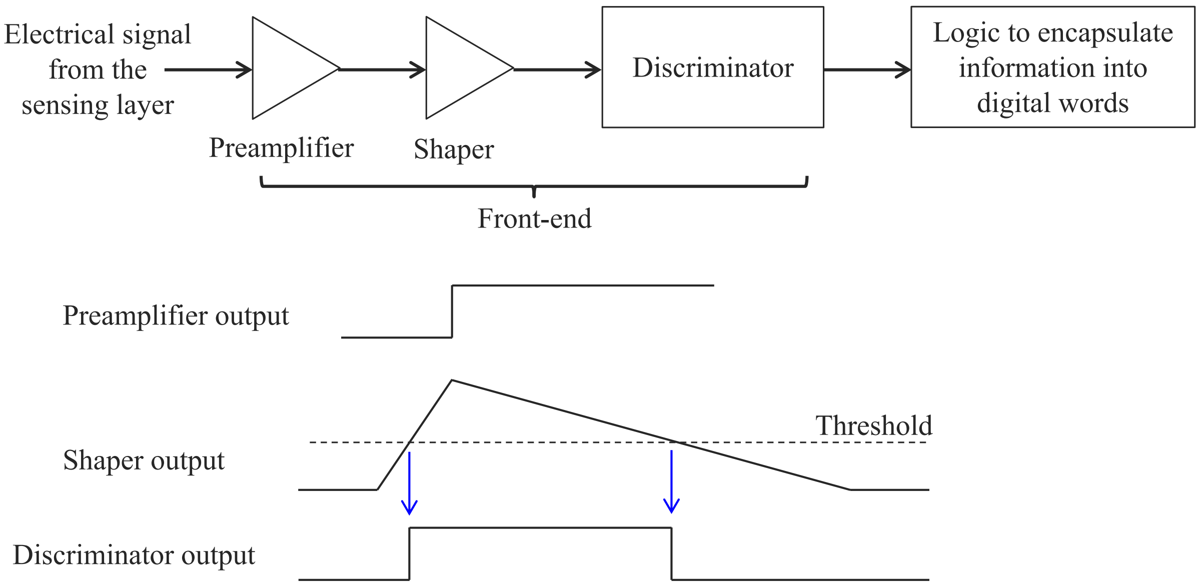


FIGURE 1.1: Basic stages of the readout chain of a pixel detector.

After that point, the signal processing continues at the digital level, which is the scope of this work.

1.2 Hybrid and monolithic pixel detectors

Hybrid pixel detectors [41] [8] and Depleted Monolithic Active Pixel Sensors (DMAPS) [117] [106] are the pixel detector alternatives of interest for the present work. In both cases, the sensing layer is segmented into a two-dimensional array of pixels to sample the radiation impinging on the detector surface. The cross-section of one of such pixels is shown in Figure 1.2 for both types of detectors.

In the case of the hybrid pixel detector, the sensing layer and the electronics readout are manufactured in separate substrates. As a result, both layers can be optimised independently; and the volume available for each of the functionalities is maximised: the surface of the sensing pixel is fully available for detection (which mazimises the fill factor of the detector), while the whole area of the electronics pixel (few tens to few hundreds square micrometers) is available to implement the readout circuitry.
While the readout chip is manufactured on silicon, the sensing layer can also be fabricated on a different semiconductor material, such as Germanium (Ge), Cadmium Telluride (CdTe), Cadmium Zinc Telluride (CdZnTe), etc., which are more suitable for detecting high energy photons (above 20 keV). The electronics layer can be implemented with an advanced technology node to pursue high device integration density and low power consumption.
The structure of the detector is illustrated in Figure 1.2a: the p-n junction present in

the sensing layer is reverse-biased to deplete the sensor volume. As a result from the interaction with the incoming radiation, electron-hole pairs are generated in the depleted volume. In the example of this figure, holes are swept towards the collection electrode by the electric field present in the depleted volume [1].
The sensing layer is electrically connected to the readout electronics by means of solder bumps (in a more advanced implementation, both substrates can be connected by means of Trough-Silicon Vias (TSVs), as it will be discussed in Chapter 2). Once the charge arrives to the electronic pixel, the readout chain described in Section 1.1 starts.

Despite the aforementioned advantages, the stack of two substrates can lead to a significant thickness, which is critical for certain applications, such as tracking detectors like CLICTD. Having a lower material budget or thinner detector is more suitable to prevent distorting the trajectory of the particles, which is to be measured.

In monolithic pixel detectors, the sensing volume and the readout electronics share a common substrate, as shown in Figure 1.2b, which is a simplified sketch of the CLICTD pixel cross-section.
The electronics utilises a thin layer, in the order of few micrometers thick, at the surface of the silicon substrate, while the remainder volume (a thickness of few ten to few hundred micrometers typically) is available for the interaction with radiation and it provides mechanical support.
As a result, the cost and complexity of the mechanical and electrical interconnection of two separate substrates is prevented. On the other hand, electronics and sensing volume cannot be optimised independently; the sensing semiconductor material is conditioned to be silicon; and the detector has a smaller fill factor: the readout electronics should be made simple or use an advanced technology node (with shrinked devices, more functionality can be integrated into a given area) to maximise the detector volume [128].

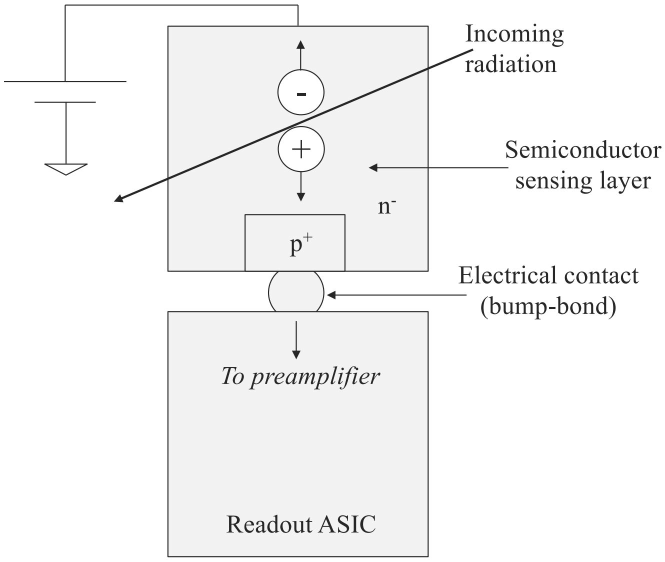
The principle of operation of MAPS is the following: a p-n junction is formed between the collection electrode and the substrate shared by the electronics and the detection volume. Such a junction is reverse-biased to extend the depletion zone and thus maximise the charge collection by drift. When the incoming radiation traverses the detector volume, electron-hole pairs are generated, and electrons (in this example) are directed towards the collection electrode by the present electric field and as a result of diffusion.
The substrate is usually not fully depleted, which makes charge collection by drift difficult and thus leads to a slower signal collection and even incomplete charge collection for high energy radiation, which may worsen the resolution of X-ray detection, for instance [119].
DMAPS incorporate modifications to the standard CMOS process to enhance charge collection. Figure 1.2b depicts some of these modifications, which have been implemented in the CLICTD chip [69] [129] [118]:

- Nested wells (in this case, a deep p-well) are used to isolate the in-pixel logic

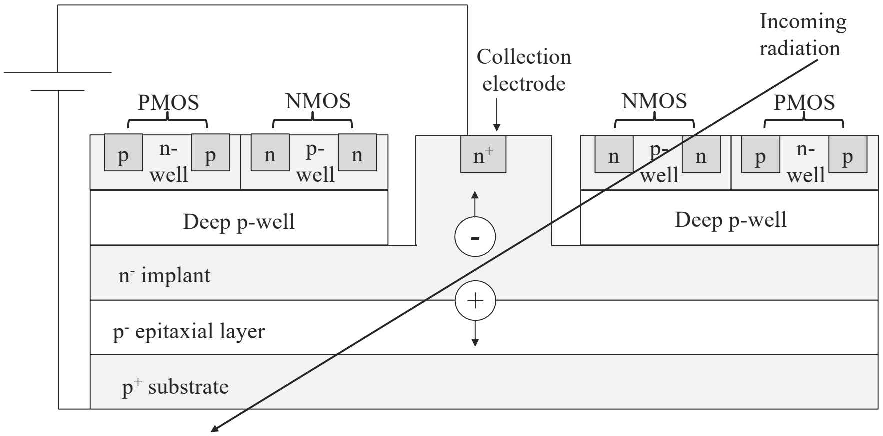
[1] Such a collection mechanism is called drift, and it is the preferred option, for it provides a faster collection and thus a lower risk of charge trapping. Collection by diffusion also occurs, which is a slower mechanism and may lead to capturing the generated charges in adjacent pixels [119].

from the collection electrode, preventing a) the noise coupling from the electronics to the electrode, and b) that a portion from the charge is not collected on the electrode, but directed to the n-wells in the electronics.

- A high resistivity epitaxial layer of a few kΩ cm (in contrast to the regular substrate resistivity of a few tens Ω cm) has been added between the substrate and the electronics. Such a layer increases the depth of the depletion zone, thus enhancing the signal collection by drift.
- The process has been modified with an additional, planar, low-dose, n-type implant placed on top of the epitaxial layer. Such an implant enables that the depletion zone spans under the deep p-wells as well, thus achieving a) a larger depleted volume, which enhances signal collection by drift; and b) electrical isolation between the in-pixel electronics and the substrate, which means that they can be biased at different voltages, and thus higher reverse bias voltages can be applied at the collection electrode to further increase the depletion zone. Since the presence of this layer paves the way to the full depletion of the sensing volume, the collection electrode can be kept small (few square micrometers) to reduce the detector capacitance seen by the front-end, which minimises the noise in the later and thus helps to enhance SNR.



(A)



(B)

FIGURE 1.2: Cross-section of a (A) hybrid pixel detector, adapted from Fig 4.10 in [10] and (B) monolithic pixel detector (CLICTD), adapted from Fig. 7.1 in [69].

2 Future trends in pixel detector readout electronics

Both hybrid and monolithic pixel detectors have been applied to a variety of radiation detection applications, from particle tracking in HEP to medical imaging. Each of these technologies offers advantages that make it suitable for a particular application, and limitations that motivate the active research to address them.

As it was introduced in Section 1.2, hybrid pixel detectors provide the versatility to manufacture the sensing and the electronic layers on different substrates, which can be optimised independently. As a result, this technology offers the flexibility to use a same readout chip with different types of sensors, which are selected to maximise the detection efficiency for a particular range of radiation energies.
To increase the spatial resolution of future detectors, the sensor pixel pitch should be reduced, and the electronic pixel pitch should follow. This poses a challenge in several aspects, for instance [135]:

- The implementation of the readout circuitry: more advanced technology nodes should be used to integrate the same or even more functionality on a reduced area.
- The power budget of the detector: with an increased transistor density, in spite of the reduced power supply of the submicron nodes, it is expected that the power consumption per unit area will increase, which must be taken into account when evaluating the cooling requirements of the detector.
- The limit of the bumping technology pitch (about 10 µm).
- A reduced pixel size will increase the charge sharing or charge collected in the neighboring pixels from that where the charge was originated. There exist on-chip algorithms that take advantage of this phenomena to enhance the spatial resolution of the detector, which should be evaluated under these more demanding scenarios in terms of charge sharing [9] [98].

Individual detectors can be interconnected to form modular, large area detectors: detector units are tiled or placed side-by-side to form a larger detection matrix. Traditionally, there were gaps of non-sensitive area between the tiles, which corresponded to the periphery of the readout chips (I/O blocks, wire-bond pads to connect the chip to other modules or to a readout board, etc.) [58].
To maximise the sensitive area, in some recent developments the peripheral blocks are distributed in-between the readout electronics, so that the area of the readout chip matches that of the sensing layer, and there is no dead area associated to the electronics when tiling detector units. The interconnection to the carrier board that interconnects the detector units is performed by means of TSVs [33] [32] [133] [48] [77].

Concerning the development of monolithic pixel detectors, in addition to the substrate modifications introduced in Section 1.2, the focus is on isolating the in-pixel electronics from the bulk active sensor volume, which enables the usage of higher reverse bias voltages in the last.
Electrical isolation can be achieved by means of the Silicon-On-Insulator (SOI) technology, in which a thick Buried Oxide Layer (BOX) is added between the electronics and the substrate [36] [33].

Part II

VERIFICATION OF A MONOLITHIC PIXEL DETECTOR FOR THE COMPACT LINEAR COLLIDER

3 Introduction

The CLIC experiment is one of the proposed future linear electron-positron accelerators to continue the legacy of the Large Hadron Collider (LHC) at CERN [27]. Such a detector has the potential to deepen the present knowledge of the Standard Model and to explore new theories on the structure of matter. One of the components of such a system is the silicon tracker, which is in charge of measuring the momentum and trajectory of charged particles arising from the particle collisions in the detector. This information can be later used to identify the crossing particles.

The CLICTD (prototype) chip has been designed to perform such measurements in the CLIC silicon tracker. It consists of a monolithic pixel detector implemented in a 180 nm CMOS imaging process, with 1.8 V as voltage supply. The process includes a High-Resistivity (HR) epitaxial layer and a planar, lightly-doped n layer to fully deplete the sensor volume and thus enhance signal collection by drift; and deep p-wells to isolate the in-pixel logic from the collection electrode.

As it was introduced in Chapter 2, this type of detector provides a cost-effective alternative to implement large area detectors (the tracker features 100 m^2 of silicon detectors); and a low material budget (the bulk is thinned down to about 50 µm), which limits the distortion on the trajectory of the particles to be detected and simplifies the requirements for cooling. In addition, CLICTD can be power-pulsed (part of the electronics can be powered on and off periodically) to limit the power consumption of the detector.

The selected design and process make this chip suitable for other applications beyond the CLIC tracker, such as portable detectors, educational uses, etc.

The chip features a matrix of 16×128 pixels of 300×30 μm^2, which yields a total sensitive area of 4.8×3.84 mm^2. Each of these pixels includes 8 collection diodes (to provide a faster charge collection), which share the associated readout logic.

Figure 3.1 sketches the structure of the pixel. The green blocks on the left correspond to the analog circuitry of the front-end (two out of the eight front-ends are depicted here). The digital output of the front-ends (the discriminator output) can be read out or masked (to ignore the collected charge in certain areas or to avoid noisy front-ends, for instance). Each front-end has some logic associated to generate a *hit bit* or hit flag, a bit that will be high when the respective front-end has received some charge and is not masked. All front-ends share the rest of in-pixel logic, which is in charge of generating the fields present in the output digital word (ToT, ToA, number of particles arrived within a certain time window or shutter), as it will be explained in Section 4.5.

So as to evaluate the response of the digital logic independently from the analog section, the pixel supports the injection of a digital test pulse, a square signal that emulates the shape that the discriminator output would have when charge is collected.

As it was mentioned in Part I, this in-pixel signal processing is possible thanks to the

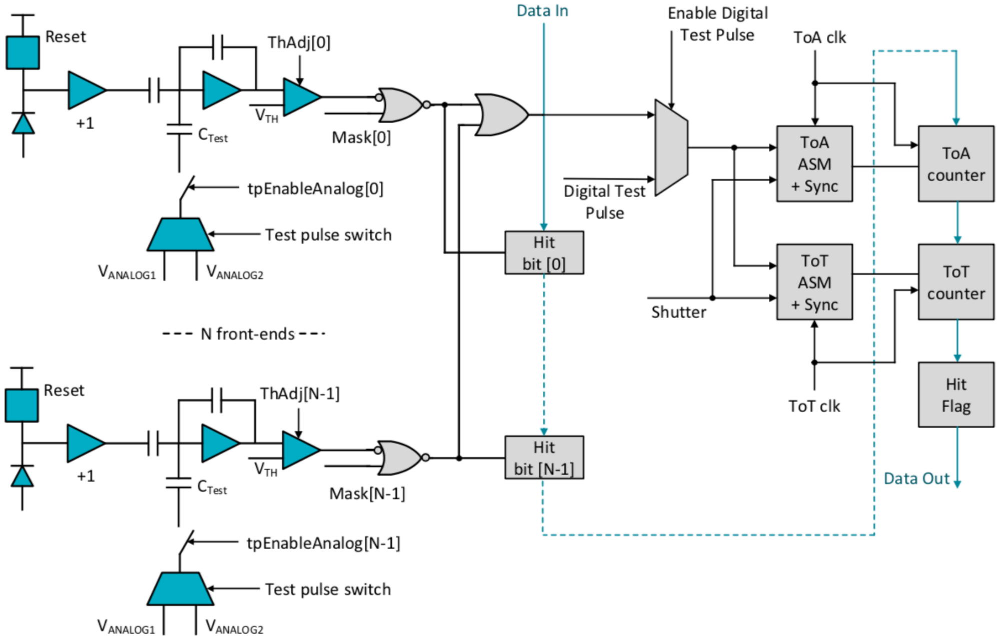


FIGURE 3.1: CLICTD channel block diagram, from Fig. 7.5 in [69].

high density of transistors available per pixel.

Figure 3.2 shows the main blocks integrated in the chip. The pixel matrix or sensitive area features the aforementioned pixel logic. The digital words generated by this logic are collected in the End-of-Column logic. The format of the word read out can be selected by means of slow control commands, while the behavior of the front-ends is configured by means of slow control commands and the value of certain analog pads. The digital output words are read out via the serial readout port.

CLICTD has been integrated following a digital-on-top approach, based on a digital design flow. Complex digital designs in Application-Specific Integrated Circuits (ASICs) are exhaustively simulated with digital simulators to ensure their correct logic and timing performance before the chip is submitted for fabrication.
The process of simulating a design and comparing the obtained results against a given specification is called verification. It is a critical part of chip manufacturing: if verification were not performed and design bugs were not discovered before tape-out, a costlier and time-demanding debugging process would be required once the chip has been fabricated, and corrections on silicon are more complex and expensive than during the design stage (if possible at all, depending on the origin of failure).

Verification environments are composed of three main elements:

- Device Under Test (DUT), which in this case is the top level netlist of the chip to be verified.
- Test code, which a) generates the stimuli that may trigger the misbehavior of the DUT or may confirm its correct operation, and b) samples the output(s) of

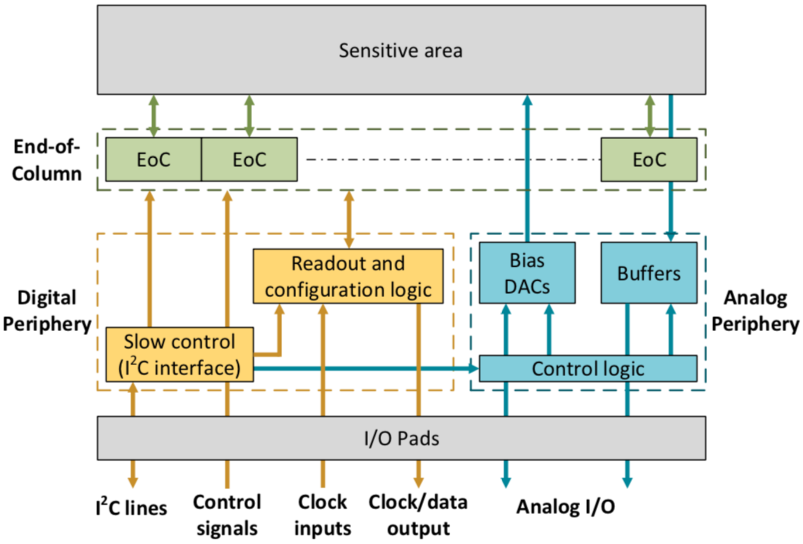


FIGURE 3.2: CLICTD block diagram, from Fig. 7.11 in [69].

the DUT to compare them against the given specification, so as to evaluate the goodness of the design.

- The testbench, which interconnects the two elements above.

In traditional verification testbenches or directed testing, the stimuli generated by the test code consist of pin wiggles (‘1’ and ‘0’ logic values that may alternate in time) applied to the DUT input ports. The outputs of the DUT also consist of pin wiggles that are sampled and compared to the expected output pattern.
This approach is useful when the DUT has a relatively low complexity in terms of number of ports, number of blocks and interaction among these blocks, and number of functionalities to test.
In the case of complex ASICs, a large number of pads is checked; there are many blocks and sub-blocks involved to provide a certain functionality; there are several operation modes and the interaction among blocks changes depending on the operation mode; blocks should be tested individually and combined as a system. Moreover, verification is needed during several stages of the design, so the test code should be flexible to adapt to a changing DUT and ideally it should be as reusable as possible for different design projects. And since several designers and verification engineers can be involved in the verification of a single DUT, it is helpful to have a standardised, modular test structure.

These requirements make the UVM [3] a suitable candidate to elaborate a verification testbench for complex ASICs. UVM is a standardised, systematic methodology that compiles best practices for efficient and exhaustive verification. It is mature (based on earlier methodologies emerged in 2000), open-source, supports multiple languages (SystemVerilog, SystemC, E) and it can work on all major commercial simulators.

In view of these advantages and given the satisfactory results obtained with the verification of former designs in the CLICdp Collaboration, such as the CLICpix2 and the C3PD chips [38], the UVM methodology has been selected to verify the

CLICTD chip.
CLICTD is a monolithic pixelated sensor and readout chip aimed at the silicon tracker development for the experiment at the CLIC experiment, which has been fabricated in a modified 180nm CMOS imaging process with charge collection on a high-resistivity p-type epitaxial layer. The details of the design and latest operation results are compiled in [70].

A comprehensive description of UVM verification is provided in [4], [109]. Figure 3.3 illustrates the basic structure of a UVM-style testbench, which includes the DUT and the UVM test code (coloured in blue). The top level of the UVM hierarchy is the Test, where the rest of components are interconnected. Usually, a devoted Test is prepared for each operation mode or feature to be verified.

Running a Test means executing all the steps specified in the Sequence, which is a list of actions to be performed. For instance, the reset sequence of a chip can consist of forcing a high value to the reset pad of the chip; after some time, forcing a low value; and after some time, restoring the high value, thus emulating a pulse to reset the chip.

The list of actions specified in the Sequence is split into individual steps (sequence items or Transactions) by the Sequencer. Transactions are an abstraction of the information exchanged between the components in the Test. In contrast to the DUT ports, the information stored into a Transaction is not limited to logic values; it does not have a direction (the DUT ports must be declared as input, output or inout); and it is timeless (the values in a Transaction can be stored, exchanged between UVM components, etc. regardless when it was created).

The Driver and the Monitors perform the interaction with the DUT. In the case of the Driver, it translates the information contained in a Transaction to the pin wiggles or stimuli applied to the DUT. In the case of Monitors, they translate pin wiggles into Transactions to process that information in the UVM domain. The Monitors sample the signals sent to the DUT (Input Monitor) and the outputs of the DUT (Output Monitor).

Transactions corresponding to the input stimuli are supplied to the Golden model. This block represents the ideal behavior that the DUT should present in response to the applied stimuli, and thus provides the expected value of the outputs.

The Transactions generated by the Golden model and the transactions obtained from the actual outputs from the DUT (those generated by the Output Monitor) are compared in the Comparator. The goal of this comparison is to determine if the actual values match the expected ones, in which case the DUT is working properly, or if there is some mismatch, which is recorded and the discrepancies will be reported at the end of the simulation.

Usually, the values of the Transaction fields supplied as stimuli to the DUT are randomised within a certain range (constrained randomization) so as to cover as many meaningful scenarios as possible. The aim of randomization is to trigger unexpected behavior of the DUT, thus revealing design bugs. The outputs of the DUT are collected and compared to the expected values in an automated fashion.

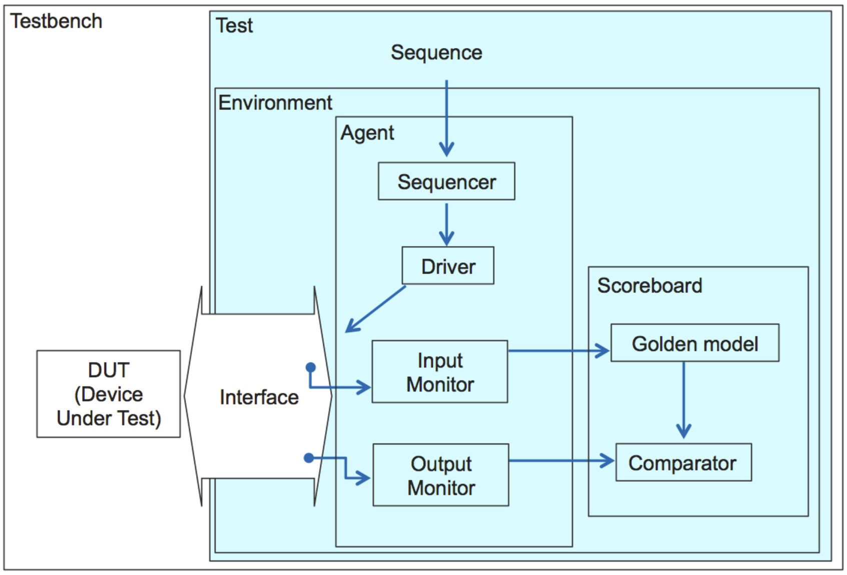


FIGURE 3.3: Illustration of the basic structure of a UVM testbench.

In terms of UVM hierarchy, the Golden Model and the Comparator are comprised in the Scoreboard block; the Sequencer, Driver and Monitor in the Agent block; and the Agent and Scoreboard are in turn part of the Environment. The Interface is a Verilog structure used to interconnect the DUT and UVM domains.

In the following chapters, the verification plan (Chapter 4) and the structure of the UVM framework (Chapter 5) developed for the CLICTD verification are described. The verification results are summarised in Chapter 6.
Throughout this document, the nomenclature of pads, registers, signals, configuration options, etc. refers to the description provided in the CLICTD design manual [67]. Further details on the CLIC experiment, as well as on the operation of the CLICTD chip, are provided in [69].

4 Compendium of CLICTD features to be verified

In this chapter, the list of features of the CLICTD functionality to be verified is described. Each of these features was translated to a Sequence that defined the series of steps to perform during the test.

Figure 3.2, which has formerly been used in Chapter 3, is an abstraction of the different functionalities implemented in the CLICTD. The verification framework interacts with all of them, except for the analog periphery (blocks highlighted in blue), i.e. it interacts with the digital circuitry.
The tests that will be described in this chapter include:

- The reset test, which sets all the logic to its default value (Section 4.1).
- A test to write and read slow control registers, which interacts with the "Slow control (I^{2}C interface)" block (Section 4.2).
- Tests that check the different modalities to apply commands (Section 4.3), which interact with the slow control and with devoted pads (represented as "Control signals" in the figure).
- The configuration test, in which values that can be selected by the user (acquisition and readout options) are provided to the "Readout and configuration logic" section by means of the slow control (Section 4.4).
- Test pulse and readout test (Section 4.5): in this test, some charge collection is emulated to evaluate whether the resulting digital words do match the collected signal and the aforementioned configuration.
  The charge collection is emulated by means of a digital pulse at the output of some discriminators (the role of this block was explained in Section 1.1), which are embedded into the in-pixel logic represented by "Sensitive area" in the figure.
  The digital words resulting from such an emulated collection are handled by the End-of-Column (EoC) logic and read out via the serial data output.

4.1 Reset test

In this test, a series of pulses (active at low level) is applied at the `RSTN` input pad to emulate the response of the chip to a series of asynchronous resets. The duration of the low level is randomised so as to spot possible synchronization issues in the reset synchroniser block.
Once the reset pulse is applied, it is checked that the serial readout `DATA_OUT` pad and some internal signals acquire a zero value; and the slow control registers are read via I^{2}C (the obtained value should match the default values of the registers).

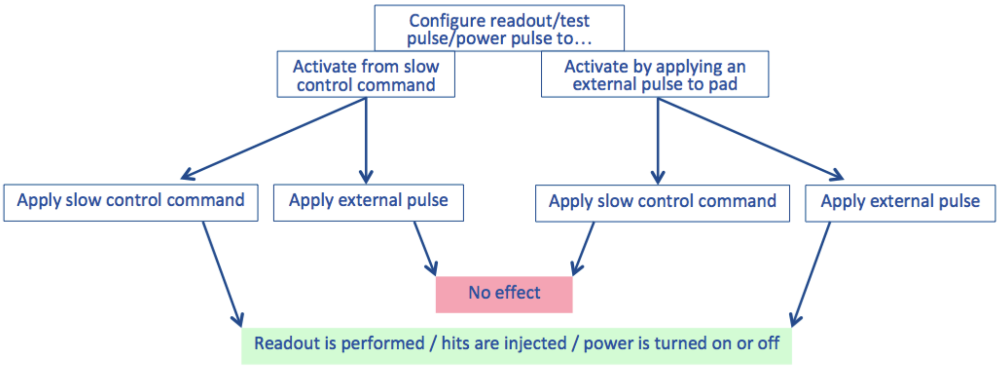

Configure readout/test pulse/power pulse to…
Activate from slow control command
Activate by applying an external pulse to pad
Apply slow control command
Apply external pulse
Apply slow control command
Apply external pulse
No effect
Readout is performed / hits are injected / power is turned on or off

FIGURE 4.1: Simplified diagram of the Sequence for the readout control, test pulse control and power pulse control tests.

4.2 Write and read registers test

Another test performed by the framework is writing and reading the content of the slow control registers. It consists of writing some random values to all slow control registers via I^{2}C, and then reading their content via I^{2}C as well. The value read back must match the value sent in the first place.

4.3 Readout control, test pulse control and power pulse control tests

The order to start/stop the readout of the matrix content (via the serial readout `DATA_OUT` pad) after configuration and after acquisition; the application of a test pulse; and the application of a power pulse to commute between power enabled or disabled can be performed either from a slow control command or by applying a pulse at the corresponding pad (`READOUT`, `TPULSE` and `PWREN` respectively). These options are summarised in Figure 4.1.

In the readout control, test pulse control and power pulse control tests, it is checked that if the chip is configured to apply any of the above stimuli from the slow control, no response should be observed if the external pulse is applied or if a wrong slow control command is applied. Conversely, if the right slow control command is provided, the readout should start, or a test pulse should be internally generated, or the power should be enabled/disabled, depending on the test.
Analogously, if the chip is configured to be driven by an external pulse and such a pulse is applied in the corresponding pad, the aforementioned response should be triggered; and no response should be triggered if a slow control command is applied.

4.4 Configuration test

This test emulates the matrix configuration. The pixel matrix is composed of 16 columns and 128 rows. From the design hierarchy perspective, the 128 rows are divided into 16 segments of 8 superpixels each (each superpixel serves 8 front-ends in turn), which yields a total of 16384 superpixels to configure. 41 bits ought to be supplied per superpixel in order to specify the corresponding acquisition and readout options. Since each superpixel has 22 flip-flops (interconnected as a shift register) to host those bits, the configuration process is split in two halves.
For both configuration halves, the first bit supplied to all columns must be a logic '1'. In the case of the first configuration half, the following 21 bits will contain the desired configuration values. For the second configuration half, the following 20 bits will contain the configuration values, and the last bit supplied is dummy. The available configuration options are detailed in [67].

Figure 4.2 represents the process to inject the configuration data into the matrix for the first of these halves (the second half will be analogous). To begin with, configuration bit 0 for superpixel number 0 of all columns is loaded to the `configData[7:0]` and `configData[15:8]` registers via a slow control command. Each position in these registers is routed to the input of the configuration shift register located at the superpixels on top of the columns, which are coloured in blue.

Next configuration bit 1 for superpixel number 0 of all columns is loaded to the `configData[7:0]` and `configData[15:8]` registers. This shifts the former bit loaded to the configuration shift registers one position down the columns, and the top position (coloured in blue) is now occupied by bit 1.
This procedure is repeated for the 22 bits corresponding to superpixel 0, and analogously for every superpixel that composes the matrix. When the very last configuration bit (bit 21 for superpixel 16383) is loaded to the input of the configuration shift register of the superpixels on top of the matrix, the first bit that was sent (bit 0 for superpixel 0) is loaded to its target flip-flop, at the bottom of the matrix (coloured in orange), and this configuration half is complete.
At this moment, the content of the matrix (the values stored in the 22 flip-flops of each superpixel) is read out via the serial output, so as to verify that the stored values match the bits that were supplied.

The same procedure is repeated to provide the second half of configuration bits, followed by a readout to verify that the stored values match the supplied values.

4.5 Test pulse and readout test

Digital test pulses emulate the effect that an incoming particle would have when interacting with the detector. The resulting, collected charge would eventually lead to a pulse of a certain duration at the output of the discriminator, which is precisely what is represented with an externally (injected via the `TPULSE` pad) or internally (forced with a slow control command) generated test pulse. The purpose of this functionality is to evaluate the correct performance of the digital part of the readout chain, that is, whether the digital word generated from such a discriminator pulse corresponds to the current configuration of the matrix and the pattern of test pulses applied.

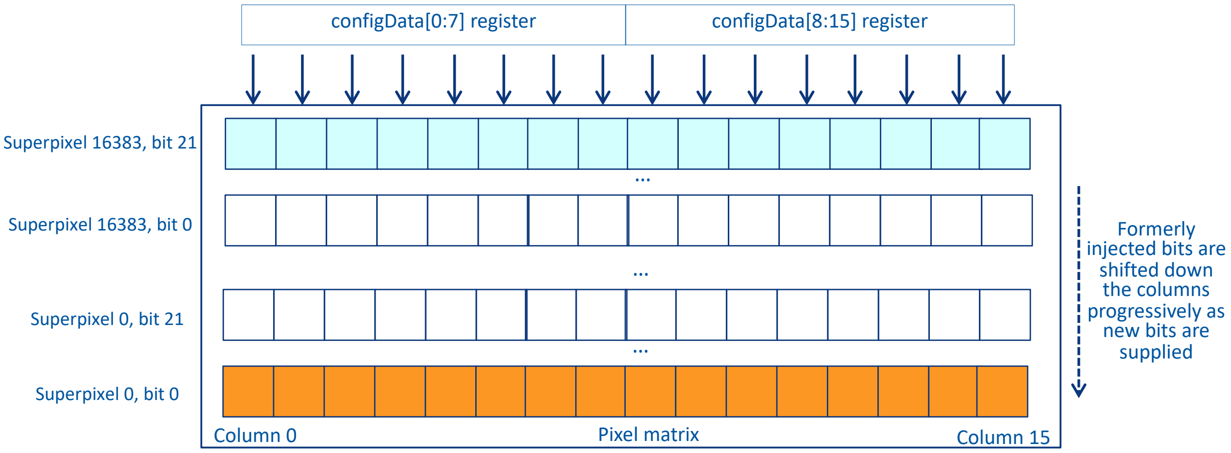


FIGURE 4.2: Diagram of the procedure to configure the matrix.

Figure 4.5 sketches the signals involved in the processing of a digital test pulse, which can either be applied via an external pad or slow command and then be translated to a pulse at the discriminator output; or it can consist of a pulse forced directly at a given discriminator output. The shutter signal represents the time window during which the detector is electronically sensitive to the incoming radiation. Three magnitudes can be measured on the discriminator output to compose the output digital word: timestamp or ToA, which is measured as the number of clock cycles (a devoted clock of 100 MHz is used, which yields a time bin of 10 ns) that span between the rise of the discriminator output and the fall of the shutter; ToT, which is an indicator of the particle energy, and which is measured with a devoted clock of configurable frequency (50 MHz, 25 MHz, 12.5 MHz or 6.25 MHz); and the photon count or number of events measured inside the shutter window, which is the count of rising edges in the discriminator output.

Three acquisition modes are available: nominal, long ToA and photon counting. The fields of the output digital word corresponding to every mode are depicted in Figure 4.4, and they stand for:

- In nominal acquisition mode, the digital word is composed of the hit flag ('1' if some of the 8 front-ends served by the pixel were hit, and that or those front-ends are not masked; '0' otherwise), 5-bit ToT, 8-bit ToA, and 8-bit hit map. Each position of the hit map represents one of the 8 front-ends that share the logic of a pixel. When digital test pulsing is enabled, they are all zeros. But if digital test pulsing is disabled and pulses are forced at some of the 8 discriminator outputs, the positions corresponding to the front-ends where the discriminator pulse is applied and are not masked will be '1'.

- In the long ToA acquisition mode, a 13-bit ToA (as well as the hit flag and the hit map) is provided, which provides a wider dynamic range as in the former mode.

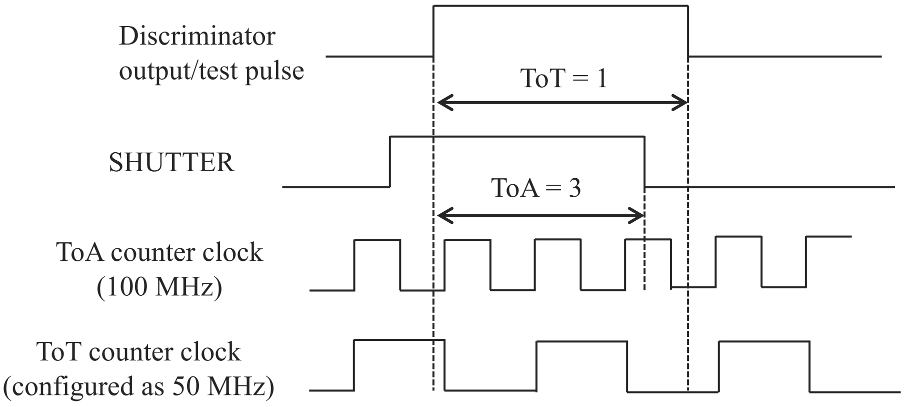

Discriminator output/test pulse
ToT = 1
SHUTTER
ToA = 3
ToA counter clock (100 MHz)
ToT counter clock (configured as 50 MHz)

FIGURE 4.3: Definition of ToA and ToT.

- In the photon counting mode, a 13-bit counter (as well as the hit flag and the hit map) indicates the number of events that occurred within the shutter window.

Regarding readout compression, it can be enabled (the whole digital word is read out for superpixels where the high flag is '1', and only one bit is read out for those in which the hit flag is '0') or disabled (the whole digital word is read out from all pixels, regardless the status of their hit flag bit).

The test pulse test consists of a) applying test pulses (which will result in all-zero hit maps) and b) forcing pulses at some discriminator outputs (which can result in non-zero hit maps, so that the correct generation of this field can be evaluated); and then reading out the matrix acquisition. The pattern read out must agree with the duration and time when the pulse was applied and with the configuration of the matrix (which pixels and front-ends had test pulse enabled, which were masked, etc.).

Figure 4.5 shows an example of the waveforms applied as test pulses. The numbers of the columns and superpixels indicated here are used to showcase that these patterns are applied to different superpixels. During the simulation, the stimuli are applied to different (randomised) combinations of pixels.
At the same time, some pulses are forced at some discriminator outputs (`disc bits` in the figure) in superpixels where digital test pulsing is disabled, or some of the 8 front-ends are masked. The hit map should be 0 at those positions corresponding to masked front-ends.

With the waveforms in the figure applied to the discriminator bits, there are pulses from different bits that are partially superimposed in time. Therefore, even if one single event (the OR of the bits that are pulsed are superimposed in time, which yields one single pulse) is used to compute the ToT, ToA and/or the photon count, the hit map should reflect that in fact there were more than one discriminator bits active.
Test pulses are followed by a serial readout to see the digital words generated from the applied stimuli.

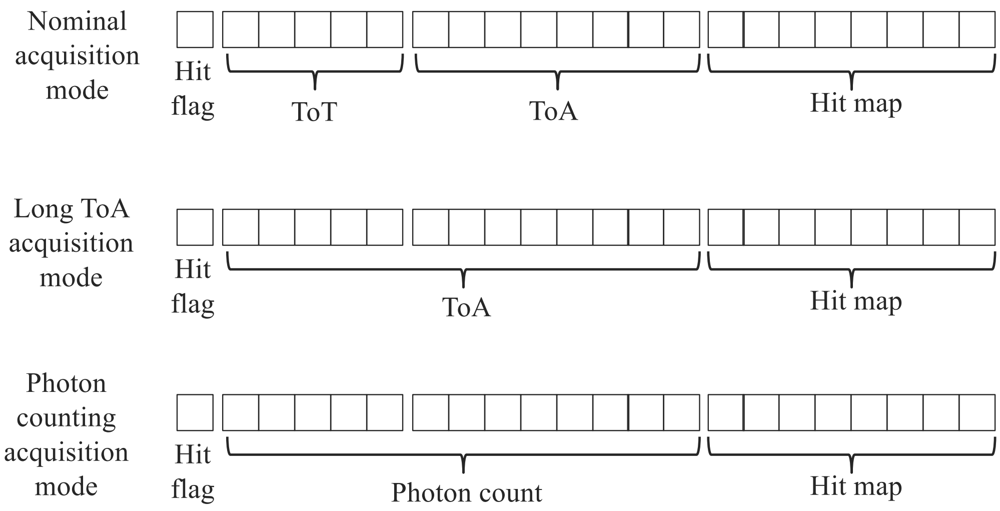


FIGURE 4.4: Fields of the digital word generated from the discriminator output for the different acquisition modes.

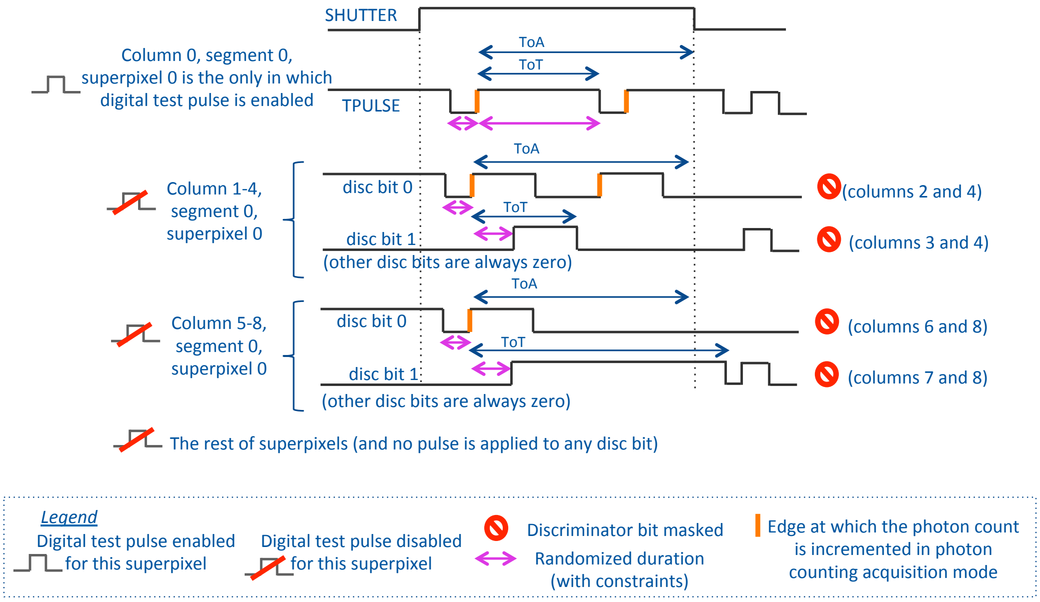

SHUTTER
ToA
ToT
Column 0, segment 0, superpixel 0 is the only in which digital test pulse is enabled
TPULSE
ToA
Column 1-4, segment 0, superpixel 0
disc bit 0
ToT
disc bit 1
(other disc bits are always zero)
(columns 2 and 4)
(columns 3 and 4)
ToA
Column 5-8, segment 0, superpixel 0
disc bit 0
ToT
disc bit 1
(other disc bits are always zero)
(columns 6 and 8)
(columns 7 and 8)
The rest of superpixels (and no pulse is applied to any disc bit)
Legend
Digital test pulse enabled for this superpixel
Digital test pulse disabled for this superpixel
Discriminator bit masked
Randomized duration (with constraints)
Edge at which the photon count is incremented in photon counting acquisition mode

FIGURE 4.5: Representation of the combination of test pulses applied to the matrix in the test pulse test.

The test pulses are supplied to the chip netlist as pin wiggles, and in parallel they are supplied as transactions to the Golden model that generates the expected ToT/ToA/photon count, hit map and hit flag values for all the superpixels depending on the configuration options set for that particular train of pulses. Once the readout that comes afterwards finishes, the stream of bits read out is broken down into ToT/ToA/photon count, hit map and hit flag fields, which are compared to the expected values generated by the Golden model.

5 UVM framework for CLICTD

In this chapter, the structure of the UVM test code used to verify the CLICTD is introduced. A simplified illustration of this UVM framework is depicted in Figure 5.1. The testbench includes the DUT (the top level netlist of the CLICTD) and the UVM Test. Both domains interact by means of three interfaces, separating the signals to be handled into three main groups: signals related to the slow control (in the `slowcontrol_interface`), signals that are related to the serial readout port (in the `readout_interface`) and the rest of pad signals to check (in the `pins_interface`). The signals exchanged at the interfaces are probed with their devoted Monitors, which translate them from pin wiggles to Transactions.

The Tests included in the framework have been described in Chapter 4. Each Test is composed of a chain of Sequences, which are orchestrated by the Virtual Sequencer.

Sequences can drive the DUT with stimuli applied as slow control commands via the I^2C pads or as pulses applied to other pads, such as `RSTN`, `READOUT`, `TPULSE`, `PWREN`, etc. In the first case, the concerned Sequencer and Driver are located in the `slowcontrol_agent`, and the Transactions containing slow control commands are translated into pin wiggles to drive the I^2C pads of the chip. In the second case, `pins_agent` is affected and the information contained in the Transactions is used to generate pulses at the corresponding pads.

The chip response is sensed by means of the aforementioned interfaces. Three situations are considered: 1) some register content is read out via I^2C, which concerns the `slowcontrol_interface`; 2) a serial readout is performed, which concerns the `readout_interface`; and 3) some change is expected in other pads, which concerns `pins_interface`.

If the chip output is received via the `slowcontrol_interface`, the associated Golden Model (`regslowcontrol_model`) generates the expected content of the slow control registers, which is compared to the actual content read via the same interface. If the output is related to the `readout_interface` and the chip is in configuration mode, `dataout_model` generates the expected value of the serial readout, which consists of the bit values that were provided with the formerly applied configuration half. If the output is related to the `readout_interface` and the chip is in acquisition mode, `dataout_model` generates the expected value of the serial readout, which consists of the hit flag, ToT/ToA/photon counter and hit map fields depending on the matrix configuration, acquisition configuration, test pulses applied …

There are several Golden models that emulate the expected behavior of certain sections of the chip: `regslowcontrol_model` generates the expected content of the slow control registers when some value is written to them; `pins_model` generates the expected value of the monitored pads and internal signals depending on the actions performed during the test; and `dataout_model` generates the expected fields

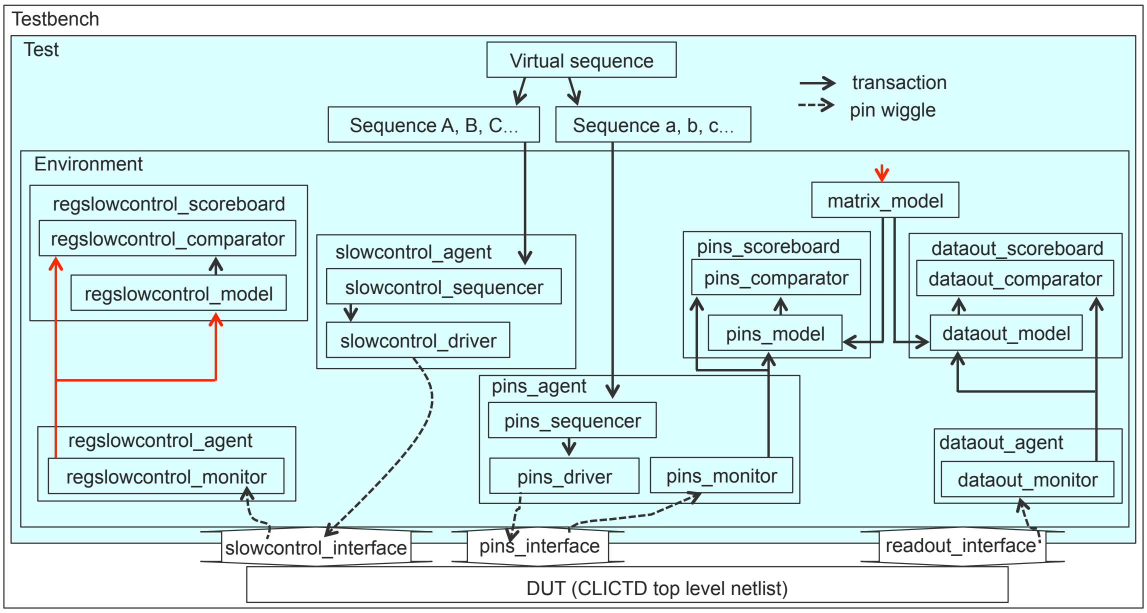


FIGURE 5.1: Structure of the UVM framework used to verify the CLICTD.

(configuration or acquisition) to be retrieved from the serial readout.

Concerning the action of `pins_model` and `dataout_model`, the expected values will depend on values formerly sent to the chip using the slow control. So as to have access to this information, the `matrix_model` block stores the configuration sent through the slow control (this link is represented in red in the figure) and supplies it to the `pins_model` and `dataout_model` Golden models.

6 Conclusion

The present part has introduced the benefits of implementing a UVM-style testbench to verify a complex ASIC, the CLICTD chip. This is a monolithic active pixel sensor aimed at the CLIC silicon tracker, featuring a doping profile that enhances the detection efficiency. The technological choice enables a low-mass and cost-effective solution to cover the large area of the tracker; and the high density of transistors per pixel enables performing in-pixel signal processing. The channel architecture has been selected to host the in-pixel logic, while maintaining a small detector capacitance.

In this part, the fundamental concepts on UVM have been presented, followed by a summary of features to verify on the CLICTD chip and an explanation on the structure of the implemented verification framework.
The defined tests have been run on the gate level top netlist, which enabled fixing some design bugs, as well as on the back-annotated, post-layout top netlist, which showed the correct timing performance of the design for the specified scenarios.
The chip was submitted for fabrication early 2019 and it worked first time on silicon. It has been fully characterised, spotting only minor bugs. In the digital logic (which was the focus of the presented framework), the matrix content was not cleared successfully (this was observed only under certain conditions) after reading it out. When this occurs, a workaround has been found by performing two successive readouts (instead of reading out once) before enabling acquisition again.
The characterization results are summarised in [68] [70].

Part III

DESIGN AND SIMULATION OF A SELF-REGULATED, SCALABLE CLOCK DISTRIBUTION NETWORK WITH VERY FINE TIME RESOLUTION FOR THE FASTICPIX ATTRACT PROJECT

7 Introduction to FastICpix and the role of the CDN in the timestamp resolution

The second part of this work is focused on the development of a CDN for the timestamp mechanism of the FastICpix ATTRACT project [42], a single-photon hybrid pixel detector concept that pursues a) the optimization of signal collection by tailoring the chip area and pixel pitch to the application requirements, and b) a very fine SPTR, in the order of 10 ps_{rms}.

The adaptability in area and pixel pitch enables enhancing signal collection for each type of sensor that is connected to the readout chip. The SNR of the first stages of the analog front-end increases with decreasing the sensor capacitance [120]. Such a capacitance is an intrinsic characteristic of the sensor, but the sensor-readout chip interface can be designed so that the front-ends are connected to only a fraction of the total capacitance.

While large photosensitive areas can be segmented from the point of view of the readout electronics to reduce the sensor capacitance seen by the front-ends and thus improve the SNR, this segmentation leads to duplicity in the readout circuit: the larger the segmentation factor, the more channels need to be handled independently, which increases the readout complexity, area and power consumption.

With a digital Silicon Photo-Multiplier (dSiPM) connected as a sensor, an aggressive example of segmentation would be to have one readout channel (preamplifier, discriminator and TDC) per Single Photon Avalanche Diode (SPAD), with the consequent area and power overhead. There are various segmentation degrees depending on which of the aforementioned aspects is to be prioritised [80].

In the case of FastICpix, the segmentation factor (or analogously the chip area and pixel pitch) is selected as the most power-efficient solution that can provide the required time resolution for a certain application [37].

Concerning the time resolution of the detector, a wide range of applications can benefit from sub-100ps time resolution. In the HEP community, with the perspective of the High Luminosity LHC (HL-LHC) upgrade at CERN, precision timing will be required to deal with increased levels of pile-up: 140-200 collisions per event with a time spread of 200 ps are expected. Detector time resolutions that are a factor 5-10 smaller than the time spread of the beam collisions are required to prevent spatial uncertainty and enable accurate time tagging of particle interactions [115], [126].

Very fine time resolution can enable millimetric spatial resolution in applications such as Mass Spectrometry Imaging (MSI), Fluorescence-lifetime Imaging Microscopy (FLIM) and LIDAR, which in turn paves the way for a reliable real-time image reconstruction, without averaging. Real-time processing is indispensable for unmanned autonomous ground and air vehicles and road detection for autonomous driving,

amongst others [61] [84] [132].
In the field of medical imaging using PET, a 10 ps-resolution detector would increase the sensitivity of the scan by at least an order of magnitude with respect to the state-of-the-art equipment, thus enabling reconstruction-less PET. Such an improvement fosters patient healthcare: reduced radiation dose applied during the scan and quantity of radiopharmaceutical, shorter scan duration, lower scan cost per patient. Furthermore, it would open the door towards new fields of diagnostics beyond oncology, including cardiovascular, neurological, metabolic, inflammatory, infectious or metabolic disease, and the exploration of paediatric, neonatal, and prenatal contexts [74] [124].

The 10-ps range refers to the resolution of the whole imaging system, including the contributions from the scintillator light generation, transport and conversion, as well as the readout electronics.
Scintillators and photodetectors are currently the main limitation towards achieving such a time resolution in the aforementioned applications. This motivates the active research aiming to develop crystals with higher light output and shorter decay time, and segmented/pixelated readouts for SiPM [11]. In parallel to this enhancement, the readout electronics must also provide an accurate timestamp of the incoming photons.

In terms of the timestamp mechanism implemented in the readout chip, FastICpix benefits from the experience acquired with the development of the Timepix4 hybrid pixel detector in the Microelectronics section at CERN [77].

The goal of the timestamp mechanism is to generate a time tag corresponding to the occurrence of a certain event (detection of an incoming particle) with a certain time resolution. The timestamp is a digital word that represents the time of arrival of the particle, encoded as the count of time bins (the time bin is in the order of 200 ps in Timepix4, 20 ps in FastICpix) that transcurred between the event and a time reference.

Figure 7.1 illustrates the timestamp mechanism implemented in Timepix4 at a conceptual level. The time tag is generated by TDCs spread across the pixel matrix (several pixels share a TDC), and the time reference is a clock signal (master clock of 40 MHz) that is delivered from a clock source to all TDCs. The master clock is sent across the matrix by means of a CDN, and it is essential that this network delivers the clock with a very well defined phase to achieve the required timestamp resolution.

As indicated in Figure 7.1 and illustrated in more detail in Figure 7.2, the timestamp measurement starts when an event occurs or, in other words, when the output of the discriminator at the pixel that was fired goes high. At this point, the local Voltage-Controlled Oscillator (VCO) of 640 MHz that is part of the TDC is started and it remains active until the following rising edge of the master clock arrives. Then the oscillator is stopped and the values of its internal phases are latched. These values yield a digital word that will be processed to provide the time tag. Since the delay taps of the VCO introduce around 200 ps propagation delay, the resulting words are separated by steps of 200 ps, which is the time bin of the TDC.

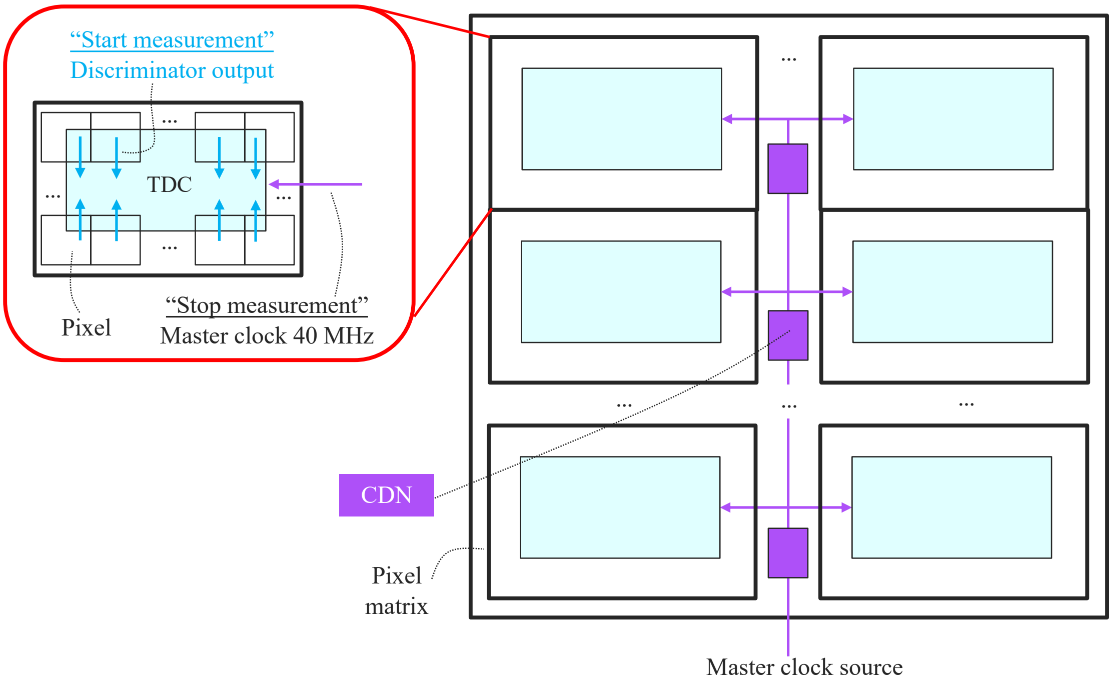


FIGURE 7.1: Pixel matrix with TDCs and generic CDN to illustrate the Timepix4 timestamp mechanism.

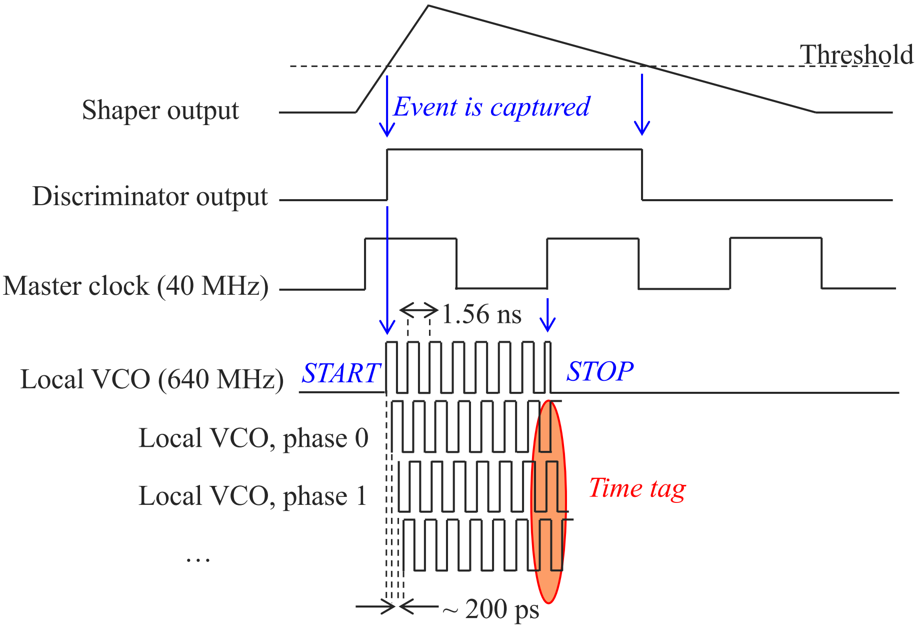


FIGURE 7.2: Overview of the Timepix4 timestamp mechanism.

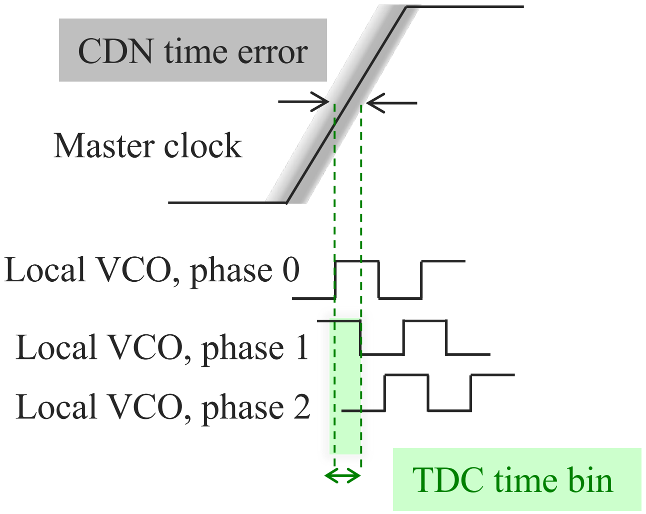


FIGURE 7.3: Representation of the maximum allowed time errors of the CDN.

Ideally, the timestamp resolution should be determined by the TDC time bin[1]. However, the resolution is deteriorated due to the time errors of the TDC and the CDN that provides the master clock to synchronise the TDCs. The time errors associated to the CDN will be discussed in Chapter 8, and they must be smaller than the TDC time bin, as illustrated in Figure 7.3, so that the arrival of the master clock takes place within the right cycle of the local oscillator, which will enable capturing the right word associated to the internal phases of the VCO.

In addition to the fine time resolution, the timestamp mechanism of Timepix4 is a low-power solution: the VCOs are active only a fraction of time (at most one master clock period) and only those associated to the groups of pixels where events occur; and the master clock that is distributed across the matrix to synchronise the TDCs has a low frequency (40 MHz).

In view of the aforementioned features of the Timepix4 timestamp mechanism, the FastICpix timestamp architecture is based on that of Timepix4. In the case of FastICpix, the TDC time bin is 20 ps and the chip and pixel dimensions are tailored to the application, which motivated the search of a CDN architecture that is scalable with the chip area and pixel pitch, and that can deliver the master clock ensuring that the time errors are bound to 20 ps.

[1] The TDC time resolution is limited by its quantization noise, which has a standard deviation of `TDC time bin`$/\sqrt{12}$ [142]. The sentence "the timestamp resolution should be determined by the TDC time bin" and analogous statements mean that the time errors associated to the CDN should be lower than the TDC time bin, so that the non-ideality that limits the timestamp resolution is the TDC quantization error, and not the CDN time errors.

In the following chapters, relevant aspects related to the choice of CDN architecture, its design and implementation are discussed. To begin with, the limiting factors to achieve the required accuracy in the delivery of the master clock are discussed in Chapter 8. The state-of-the-art of CDN architectures is reviewed in Chapter 9, with a special emphasis on those alternatives that are more suitable for this work. Chapter 10 is devoted to clarifying certain simulation and implementation details that will be relevant to understand the concepts introduced afterwards.
Once the context has been set, the FastICpix CDN architecture is described in Chapter 11 and the performance of its main component is discussed in Chapter 12. To conclude, this part is summarised in Chapter 13, in which future lines of research are suggested as well.

8 Time errors that impact the CDN time resolution

In an ideal CDN, the latency or propagation delay from the master clock source to every TDC is the same and it is constant over time.
Due to Process-Voltage-Temperature (PVT) variations, added to layout imbalances, the circuit elements that compose the CDN may have slightly different delays, which results in a static time error or skew in the network latencies. This effect manifests as a time offset between the clock delivered to different TDCs, as shown in Figure 8.1 (left).
On top of this time error, the delays will also be dynamically affected by changes in voltage supply and temperature gradients during operation; and due to noise coupled mainly from the power supply due to the switching activity of the circuitry. These effects manifest as jitter in the clock edges, or deviation between the edges of the actual clock with respect to an ideal clock. Jitter can also enter the CDN superimposed to the clock source, as a result of the non-idealities of the clock generator [136]. This effect is represented in Figure 8.1 (right), where the net a receives an ideal clock, while the net b receives a clock with jitter superimposed.

With the goal of an accurate master clock distribution, which is indispensable for a reliable timestamp, the CDN must include mechanisms to reduce the impact of static and dynamic time errors. Such mechanisms will be introduced in Chapter 9; in this chapter, the main types and sources of time error are explored, so as to provide an understanding on the available alternatives to mitigate their negative effects.
It will also be specified which sources of time errors are considered in this work and how they are included in the simulation of the implemented CDN.

8.1 Static time errors of the CDN

Clock skew is defined as a "spatial variation of the clock signal as distributed through the system (...). Because of the various delay characteristics of the clock paths to the various points in the system, (...) the clock signal arrives at different points at different times" [95]. Such a deviation is measured from some reference point in the system; in this work, it is measured from the point where the clock is injected into the CDN.

In massive logic circuits such as microprocessors, skew reduces the portion of the clock period available for data processing, and thus it can limit the highest allowed clock frequency. [46] provides a clear visualization of the distribution of skew across the chip area; in this reference, skew accounts for about 5% of the clock period.

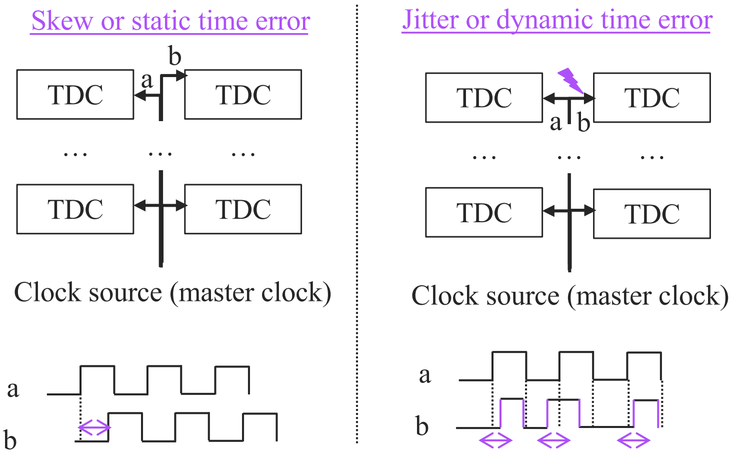

Skew or static time error
Jitter or dynamic time error
TDC
a
b
Clock source (master clock)

FIGURE 8.1: Static and dynamic time errors that can occur in the CDN.

In the context of the current work, skew deteriorates the time resolution of the timestamp mechanism of a pixel detector.

Skew is a static effect, i.e. the same time offset is observed over time between two nodes that present such an error between them, and it is mainly related to the design, fabrication and operation of the circuit.
Concerning the design stage, a careful layout of the network should aim at maximising the geometrical symmetries between branches, so as to balance the parasitic load of interconnects. Different network topologies that pursue such symmetries will be discussed in Section 9.2.
Following the classification in [141], skew introduced during fabrication is caused by process variations; while the error introduced during operation comes from system parameter variations and ageing. These error sources impact the propagation delay of cells and interconnects, causing a deviation from their nominal value, and thus introduce skew. In the following, this description of error sources is expanded with examples compiled from [40] [49] [131] [102] [138], amongst others. The description is followed by literature research on the impact of these variations.

8.1.1 Process variations

According to their origin, process variations can be further classified into:

- Device parameter variations or transistor mismatch, which include systematic and random contributions. Amongst the systematic effects, the non-idealities of the lithographic system and layout-dependent effects, such as Well Proximity Effect (WPE)[1]. Some random sources are fluctuations of the gate length, gate oxide thickness, threshold voltage, effective channel length, channel mobilities; Random Dopant Fluctuation (RDF) in the transistor channel; Line Edge Roughness (LER) or variation in the edges of the gate of the transistor; and various proximity effects, including effects associated with stress/strain, which impact the mobility of charge carriers. Further details are provided in [72].

- Interconnect parameter variations: fluctuations in the Inter-Layer Dielectric (ILD) thickness, as well as in the interconnect width and thickness.

The more advanced the technology node, the smaller the feature size, the larger the device density, and the higher the number of available interconnect layers and the routing density, and hence the larger the impact of these variations.

According to the scope of their impact, process variations can also be classified as lot-to-lot; wafer-to-wafer; inter-die or die-to-die (D2D) variations; and intra-die or within-die (WID) variations.

Lot-to-lot, wafer-to-wafer and die-to-die variations affect all transistors on a die equally and they are characterised as systematic differences. These variations arise from temperature gradients, defects in the photo-lithography masks, etc. during fabrication.

Conversely, within-die variations translate into differences between devices closely placed and increase when the transistor size decreases. These effects correspond to the aforementioned device mismatch and they are mainly random and independent from device to device [14].

8.1.2 System parameter variation and ageing

System parameter variations can be further divided into:

- Power supply (static) fluctuation or IR drop, which stands for the gradient in the power supply that depends upon the position across the chip area. This voltage drop is associated to the resistance of the power rails and increases with the distance from the power pad and with a non-adequate power supply distribution network.

- (Static) temperature gradients impact the threshold voltage, the carrier mobility and the line resistance, amongst others. The main sources of temperature generation in the chip are the switching activities of the cells and the self-heating of the interconnect lines due to the current passing through them [6].

- Non-uniform distribution of clocked registers (clock driver load mismatch).

The dynamic power supply and temperature variations are responsible for jitter and they will be addressed in Section 8.2.

[1] Transistors implemented close to the well edge have a different performance as those further from the well edge. This effect is mainly related to differenes in the threshold voltage and it can cause a divergence in the transistor speed in the order of 10%.

The last of the aforementioned sources of variation is ageing, which stands for the gradual deterioration of device performance in time [76] [79] . The extent of such a degradation will depend upon the voltage, temperature and frequency of operation; the amount of switching that a specific transistor experiences, etc. According to the nature of the degradation, the following distinction can be made:

- Effects that lead to an abrupt failure of devices:
  - Electromigration (EM): if the current density across a connection is high enough, the associated heat dissipation will repeatedly break atoms from the structure and move them, i.e. the electrons will carry along metal atoms. This results in vacancies of atoms in certain regions, which causes open circuits; and accumulation of atoms in other regions, which can lead to shorts. This effect can have a critical impact both in signal and power wires.
  - Time-Dependent Dielectric Breakdown (TDDB): due to a prolonged exposure to moderate electric fields, those associated to the operation voltage, traps are generated in the gate oxide and a conducting path is created. This opens the door to currents that become higher with time, which eventually leads to the breakdown of the device.
- Effects that lead to a device parameter drift:
  - Hot Carrier Injection (HCI): in both NMOS and PMOS transistors, carriers are accelerated by the electric field associated to the drain-source voltage across the inverted channel. Those that acquire enough energy to leave the channel can damage the gate oxide (they get trapped and form space charges) or contribute to the gate current, altering the electrical characteristics of the transistor.
  - Negative Bias Temperature Instability (NBTI): this effect has an impact on PMOS transistors and it arises due to a prolonged exposure to a negative bias voltage (the gate is negatively biased with respect to the source and drain, the transistor is in inversion). The impact of this effect is an increase (in absolute value) of the threshold voltage. NBTI degradation partially recedes when the aforementioned stress condition is removed (recovery effect).
  - Positive Bias Temperature Instability (PBTI): this effect is similar to NBTI, but it affects NMOS transistors. The stress condition is that the NMOS transistor is in inversion (the gate is positively biased with respect to source and drain). PBTI was traditionally negligible compared to NBTI, but it acquired the same order of magnitude with the introduction of high-κ metal gates.

8.1.3 Impact of the various variation sources on skew

This section compiles examples of the impact of the aforementioned variation sources on the circuit performance.

[141] reports the skew (below 100 ps) observed between distant cells in a CDN based on an H-tree, which is implemented in a 180nm process and spans across an area of 4 cm^2. Amongst the process variations introduced in Section 8.1.1, the largest percent of variation across the chip area corresponds to the clock driver load (20%), followed by the supply voltage (10%) and temperature (8%); the rest of variations (ILD thickness, wire thickness, threshold voltage, effective channel length, gate oxide thickness) have a variation of 5% or lower.
Skew is twice as much sensitive to interconnect variations (ILD thickness, wire thickness) as to variations in the rest of parameters, probably because the network consists mainly of wires.

[92] evaluates the impact of within-die process variations (device and wire variations) on the propagation delay of a structure composed of buffers and interconnects. Several process nodes are evaluated, showcasing that the wire resistivity and thickness are becoming the main sources of delay variability on the side of interconnects, while the gate oxide thickness and the threshold voltage are the dominant sources of delay variability on the side of devices as technology advances.
In the 70nm node, which is the closest to the node of interest for FastICpix, the effective channel length stands for close to 24% of the delay variability, followed by the wire resistivity (20% of the delay variations); the wire geometrical parameters stand for about 10% of the delay variability, similar to the impact of the supply voltage and the threshold voltage; while variations in the gate oxide thickness account for 5% of the delay variability.

In [91], the impact of random and static, process and system parameter variations on the skew observed in a CDN based on an H-tree is evaluated for various process nodes. As technology advances, in this network topology the average value of skew has decreased from a few hundreds of picoseconds (180nm node) to several tens of picoseconds (45nm node), and it is mainly dominated by transistor mismatch, altough interconnect imbalances are becoming more significant as technology scales down.
In contraposition, the variability of skew has increased from close to 40% of the average skew (180nm) to close to 50% (45nm). The contributions of transistor and interconnect variations are very similar, with the interconnect variations acquiring a larger significance as technology advances.

[112] evaluates the impact of IR drop in the skew of the CDN of a few chips (process nodes 350nm down to 180nm). For a "large chip area" (the CDN consists of 30000 transistors, the power grid consists of 400000 resistors and 250000 capacitors), a 10% gradient in the supply voltage can lead to a 5%–10% change in the propagation delays of the CDN, leading to a 30% change in the skew.

On the impact of temperature gradients, CDNs based on H trees are used in [13] and [34] to evaluate the impact of such gradients on the interconnect delays. [13] reports that the skew induced by a gradient of up to 60 °C can be as high as the delay of one inverter in a 120nm process, while [34] observes a difference of 10% in the propagation delay of wires (a few thousands of micrometers long) that undergo a temperature difference of 100 °C.

Concerning NBTI, [110] reports an increment in the propagation delay of about 5% across a span of 10 years for a 32nm benchmark circuit (`b02` circuit from the `itc99` benchmark suite [28]); and [79] reports a 10% degradation in the oscillation frequency of a ring oscillator implemented in 65nm after 500 hours of stress (when the oscillator was active while the stress was applied), or 6% degradation when the oscillator was switched off while the stress was applied.

8.1.4 Sources of skew considered in this work and how they are evaluated

In this work, the sources of skew that are considered are layout imbalances introduced during the design stage and within-die process variations.
The variability associated to the manufacturing process is reflected in the corner statistical device models, which take into account the nominal, worst-case and best-case device parameters. In this work, the most common corners used for digital design characterization will be considered: TT (typical n-channel device parameters, typical p-channel device parameters), FF (fast n-channel device parameters, fast p-channel device parameters) and SS (slow n-channel device parameters, slow p-channel device parameters) [57] [114]. These process corners provide the nominal, shortest and longest cell and interconnect delays respectively.

The process corner models are complemented by including the impact of voltage supply and temperature of operation: the shortest delays are obtained with the highest voltage supply, lowest temperature and FF devices, while the longest delays are obtained with the lowest voltage supply, highest temperature and SS devices.

Skew will be calculated from the propagation delays observed by means of digital, back-annotated simulations (taking into account both the cell and interconnect delays) performed for the aforementioned corners, as it will be introduced in Chapter 10.

8.2 Dynamic time errors of the CDN

Clock jitter consists of "the time deviation of a controlled edge from its nominal position" [30], which varies dynamically in time. In other words, it refers to the variations that occur in the edge locations of the clock with respect to the moment at which they should arrive. Such variations can also be evaluated in the frequency domain, in which case they are referred to as phase noise.

To use the frequency domain analysis, the clock is modelled as a quasi-perfect sinusoidal signal: $v(t) = V_0[1 + \alpha(t)]cos[\omega_0 t + \phi(t)]$, where $v(t)$ is the clock signal, which fluctuates in amplitude and phase with time; V_0 represents its peak amplitude and $\alpha(t)$ models the small amplitude fluctuations; ω_0 is the nominal (angular) clock frequency; and $\phi(t)$ is a random phase, varying with time, which introduces the dynamic time error.

The power spectral density of such a clock signal contains a Dirac delta function centered at the nominal angular frequency, which corresponds to the ideal sinusoidal component of the waveform, but the spectrum is broadened due to the presence of the noise components. The time-domain variance that characterises the phase variability associated to jitter can be derived from the power spectrum [111].
Working in the frequency domain is the most useful approach to design and characterise VCOs, frequency synthesisers, amplifiers, etc., where jitter is caused by electronic noise. When the physical mechanisms that cause random jitter are stochastic noise processes, these are best described in the frequency domain [136].

On the other hand, working in the time domain is preferred for synchronization applications: high-speed links, Clock-and-Data Recovery, etc. This is the approach used in this work, since the magnitude of interest to evaluate the performance of the CDN is the difference in the edge locations between the clock that is delivered to the TDCs and an ideal clock, which would latch the TDC measurement at the right time and not introduce any offset in the measurement.

Time-domain jitter is usually evaluated by means of three definitions, as illustrated in Figure 8.2 [136] [131]:

- Absolute jitter (also known as phase jitter or Time Interval Error (TIE)) is measured as the time difference between the edges of the clock signal of interest and the ideal locations where the edges would occur in the absence of jitter. This is the type of jitter used in the characterization of synchronous systems to describe the tracking error between two clocks [75] [26], and hence it is the most relevant magnitude for the present work.

- Period jitter is the first difference of absolute jitter: it is defined as the difference between a given period of the clock of interest, and its average period (which is also the period of the ideal clock). This type of jitter provides information on the magnitude of the period fluctuations (but not on the dynamics of such fluctuations) and it is helpful for a long-term evaluation of jitter. It is commonly used in the design of digital system where the minimum (or maximum) time period is of importance; for instance, to determine the time available for data processing per clock cycle to prevent timing violations in a chain of sequential logic.

- Cycle-to-cycle (c2c) jitter is defined as the difference between successive periods of the clock of interest or, in other words, it is the second difference of absolute jitter. It provides information on the short-term dynamics of jitter or local changes in the clock period. This type of jitter is useful for the analysis of free-running oscillators, since c2c retains its meaning when the oscillator is embedded in a regulated loop (such as a Phase-Locked Loop (PLL)), while period jitter will rather switch to reflect the loop dynamics [50].

[62] compiles the key metrics of several dDLLs designs, a circuit that will be part of the selected CDN architecture, including jitter (but not skew). Focusing on the compared 65nm designs, a jitter lower than 6 $\mathrm{ps_{RMS}}$ can be achieved when operating

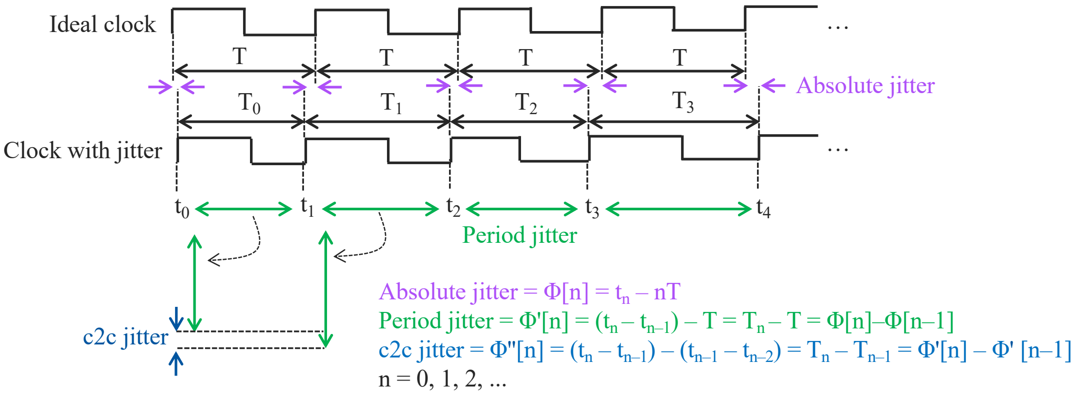

Ideal clock
...
T
T
T
T
Absolute jitter
T_0
T_1
T_2
T_3
Clock with jitter
...
t_0
t_1
t_2
t_3
t_4
Period jitter
c2c jitter
Absolute jitter = $\Phi[n] = t_n - nT$
Period jitter = $\Phi'[n] = (t_n - t_{n-1}) - T = T_n - T = \Phi[n]-\Phi[n-1]$
c2c jitter = $\Phi''[n] = (t_n - t_{n-1}) - (t_{n-1} - t_{n-2}) = T_n - T_{n-1} = \Phi'[n] - \Phi'[n-1]$
n = 0, 1, 2, ...

FIGURE 8.2: Types of time-domain jitter, adapted from Fig. 5.1 in [136].

at hundreds of megahertz, i.e. jitter accounts for about 0.5% of the clock period. In particular, in [100] the worst-case RMS jitter stands for 3% of the delay line adjustment step, which is 180 ps; in [134] it stands for 1% of the delay adjustment step, which is 11 ps; and in [60] it stands for 1% of the delay adjustment step, which is 450 ps. There are also designs that can achieve sub-picosecond RMS jitter with sub-picosecond delay adjustment steps ([2] in 130nm process).
Note that the Delay-Locked Loop (DLL) performance reported in these references corresponds to cells whose area is orders of magnitude smaller than the area used to implement the demonstrator circuit in this work, hence the performance cannot be meaningfully compared.

On the origin of dynamic time errors, some sources of jitter in the CDN are the following:

- Dynamic variations of voltage supply and temperature.
- Intrinsic, inherent or physical noise of the transistors and wires that conform the network (electronic noise).
- Power Supply Noise (PSN), which leads to Power Supply Induced Jitter (PSIJ), and noise coupled from the substrate.
- Coupling noise or crosstalk between neighboring, switching nets.
- Noise superimposed to the input clock.

These sources are described in more detail in the coming subsections to understand their impact on the circuit performance. According to Chapter 10 in [103], PSN is the main contribution to jitter: “power supply noise fundamentally limits the performance of clock networks”.

Before proceeding to the description, it shall be clarified that this work does not aim at analysing or quantifying the contributions to jitter in the CDN, but the purpose is to understand which is the budget of this type of error that would still enable achieving the required time performance.

8.2.1 Dynamic variations of voltage supply and temperature

In Section 8.1.2, it was explained that voltage supply and temperature gradients across the chip area lead to differences in the propagation delay of the different CDN branches and thus cause skew.
Dynamic voltage supply and temperature changes (during the chip operation) build on top of such delay divergences, modulating them in time. Voltage supply variations can come from a non-ideal power source; while temperature changes can come from the self-heating of the chip during operation, if it is not thermally regulated or there is no proper path for heat dissipation.

[143] evaluates the impact of a change in the power supply (both the impact of a change in power and ground rails) on the skew of several CDNs based on clock trees, which are implemented in a 130nm process and range from a few tens to a few thousands nodes. When a change of 10% in the voltage supply or the ground lines is applied, skew increases up to 3 times with respect to the nominal scenario, while an increment of up to 6 times is reported when the 10% change is applied to both power and ground rails simultaneously.
The impact on skew will depend both on the CDN topology and on the power distribution network, but this example showcases that dynamic voltage changes can severely impact the CDN performance.

On the impact of a dynamic temperature change, [113] (using a 65nm process) reports an increment close to 5% in the propagation delay of clock buffers when the temperature rises from 25 °C to 125 °C; and an increment in the propagation delay of interconnects of a few percent for short wires (100 µm long), or up to 50% for long wires (1000 µm long), when the temperature increases from 25 °C to 150 °C.

8.2.2 Electronic noise

Electronic noise is caused by the small current and voltage fluctuations that occur within active (transistors) and passive (resistors) devices. Quoting Chapter 11 in [45], “the existence of noise is basically due to the fact that electrical charge is not continuous but is carried in discrete amounts equal to the electron charge, and thus noise is associated with fundamental processes in the integrated-circuit devices”.
This random noise cannot be eliminated, but it can be reduced with the choice of circuit topology and by increasing the power consumption [51].

The most common types of physical noise that are relevant for digital systems are white noise (thermal and shot noise) and 1/f noise (flicker noise) [89]. More details are provided next [45]:

- Shot noise arises from the passage of carriers across a p-n junction, which is a random event that depends on the energy and velocity of the charge carriers. The current circulating in the transistor channel is, in fact, composed of random, independent current pulses; it has an average value and a random fluctuation superimposed on top of it, which is associated to this type of noise.
- Thermal noise is due to the random thermal motion of electrons. It depends directly on temperature (as temperature approaches absolute zero, this source of noise is cancelled).
- Flicker noise is caused mainly by traps created during the semiconductor fabrication, which capture and release carriers randomly. The time constants associated with these phenomena concentrate this type of noise at low frequencies.

In [39], the impact of electronic noise and PSN are simulated in DCDLs implemented in a 350-nm process. These results suggest that jitter caused by PSN is an order of magnitude larger than that associated to electronic noise. The same proportion is indicated in [52].

8.2.3 Noise coupled from the power supply and the substrate

In a non-ideal power distribution network, there are fluctuations in the power supply in the form of ripples or PSN. This noise depends on the switching patterns and the rise/fall times of the clock at the different circuit nodes. Such fluctuations can be divided into IR drop, or voltage drop across the power distribution network due to the electrical resistance or the wires; and Ldi/dt noise, which are the voltage spikes across the wire inductance due to the switching activity of the circuit.

Power supply noise has an impact on the charging and discharging of the cell and interconnect capacitances, thus affecting the associated propagation delays and the slope of the clock [125]. This in turn alters the network latencies dynamically (changes of 30% to 50% in the propagation delays can be expected [144]), causing PSIJ. This source of noise is the dominant contribution to jitter [103].

As for the noise coupled from the substrate, the switching of transistors and interconnects does not only cause ripples in the power rails, but it can also be injected to the substrate via capacitive coupling. Currents can flow through the substrate due to the non-zero dielectric constant and conductivity of the substrate material, and couple to circuits located in a different region [43].

Some techniques to alleviate PSN and the noise coupled through the substrate include: abundant placement of on-chip decoupling capacitance (especially close to the buffers or other cells sensitive to having a significant switching activity); reducing the clock frequency, the switching activity of the circuit or distributing the logic switching in time (the last is applied in the present work); using a lower voltage supply or scaling it in time according to the operation needs; ensuring that the Power Distribution Network (PDN) is properly dimensioned for the circuit area and demand; use separate power rails for analog and digital circuits; place noisy and sensitive circuits apart in the floorplan stage of the design flow; place guard rings between noisy and sensitive circuits to provide a low impedance path to the off-chip ground/power supply for the substrate currents [7].

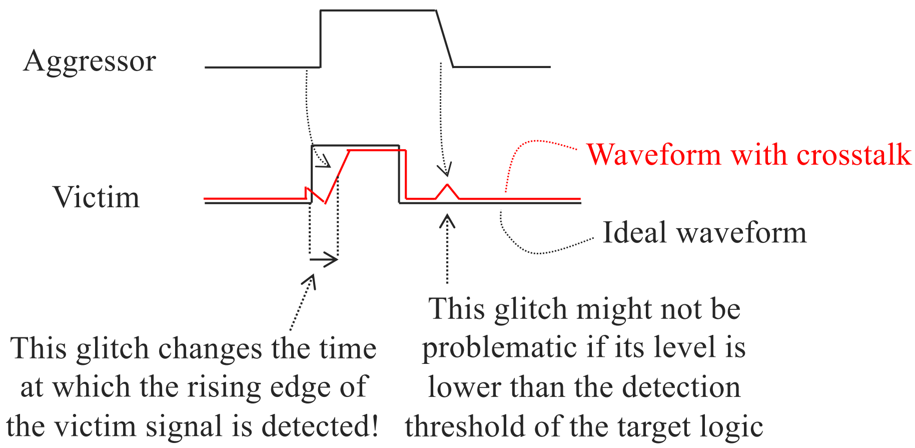

Aggressor
Victim
Waveform with crosstalk
Ideal waveform
This glitch might not be problematic if its level is lower than the detection threshold of the target logic
This glitch changes the time at which the rising edge of the victim signal is detected!

FIGURE 8.3: Impact of crosstalk on the timing of a victim signal, adapted from Figure 5 in [86].

8.2.4 Crosstalk

Crosstalk or noise coupling from neighbouring clock lines (mainly by means of capacitive coupling between the wires) stands for the signal injection from an aggressor, switching line to a victim line (which can be switching or have a stable value).
When signal transitions (rising or falling edges) occur in the aggressor line, high-frequency energy couples to the victim line, inducing some voltage (a glitch) in the last. The polarity and magnitude of this glitch depend on the polarity, the sharpness (short or long rise/fall times) of the transitions of the aggressor and on when do they occur relative to those of the victim [86].
This type of noise can alter the timing path of the victim line. For instance, in the example shown in Figure 8.3, the coupling is such that the rising edge of the victim line is distorted. The victim signal crosses its 50% value later in time, which results in a longer propagation delay in that path (for that particular clock edge).

Crosstalk can be reduced by shielding the sensitive lines, increasing the space between switching lines, limiting the slope of signals (the shorter the rise and fall times, the stronger will be the coupling), and by using differential signaling. There are also equalization schemes to counteract the pick-up signal [15].
In this work, two metal layers are reserved for the clock nets to prevent cross-coupling with other lines: metal 7 is used to route the clock, while metal 6 is routed right underneath and connected to ground for shielding. The rest of signals are routed in metals below metal 6. Besides, a ground line in metal 7 is routed between adjacent clock lines to prevent cross-coupling (this occurs where the path that propagates the clock upwards in the column of pixels is in the vicinity of the path that propagates the clock downwards in the column of pixels).

8.2.5 Noise superimposed to the input clock

Jitter can also enter the DCDL superimposed to the signal coming from the clock source. In this work, the clock source is an external reference. A common configuration is that the clock source is the output of a PLL.

In that case, the origin of the jitter superimposed to the input clock will be the PLL jitter, which can be caused by intrinsic or extrinsic noise. The first is associated to the electronic noise of the different circuit components of the PLL (the charge-pump, loop filter, VCO, feedback divider and phase detector) and it is mainly a random process. Extrinsic noise, on the other hand, can be either present at the PLL input (the reference clock) or coupled from the power supply and the substrate. The VCO and the loop filter must be carefully designed to minimise the impact of both types of noise [136]. According to [52], the contribution of the extrinsic noise sources is dominant.

8.2.6 Sources of jitter considered in this work and how they are evaluated

Time-domain, absolute jitter will be used to evaluate the CDN performance with regards to dynamic time errors. The standard deviation of the time difference between the edges of the clock signal and the ideal locations where the edges would occur in the absence of jitter is used to define a jitter budget [85], so that the total time error (including static and dynamic contributions) is bound to 1 TDC time bin, 20 ps, at each node of the CDN. This will be explained in Section 12.1.1.

As it was mentioned before, the goal of this work is not to quantify the actual jitter sources present in the circuit, but to understand which is the impact of jitter and how to reduce its effects.

To model the impact of jitter on the dDLL performance, a clock with a certain amount of jitter superimposed is used as the master clock input of the DCDL. The half period of this signal changes dynamically as `(ideal half period of the master clock + random delay)`, as shown in Figure 8.4. `random delay` is a random magnitude with Gaussian distribution and 0 mean. Different values of standard deviation are considered, to determine which is the largest jitter variability for which the time error target is still met.

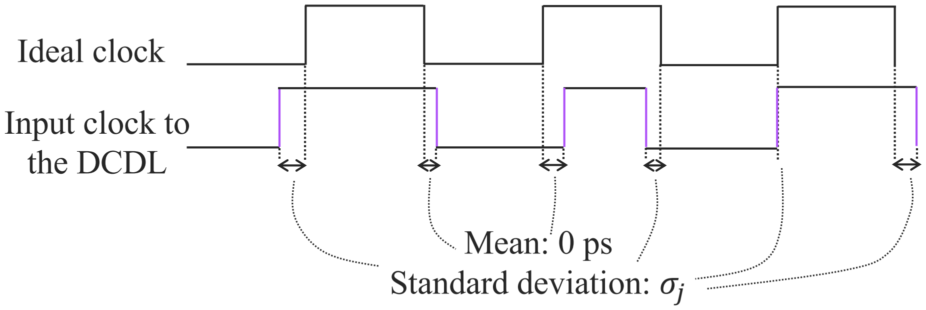


FIGURE 8.4: Illustration of the random, Gaussian jitter superimposed to the input clock of the DCDL.

9 State-of-the-art of CDN design

In this chapter, the state of the art of CDN architectures is revisited, with a special focus on those that have the potential to comply with the FastICpix requirements, namely: 1) provide very fine time resolution in the adjustment of the network latencies and a time error in the delivery of the clock lower than 20 ps, and 2) scalability with the chip area and pixel pitch dimensions.

9.1 Fundamentals of a Clock Distribution Network

The structure of a CDN is traditionally described as a tree of components, as shown in Figure 9.1. The root of such a tree is the clock source, from where the clock is guided across the trunk, branches and leaves of the network, to be eventually delivered to the target logic or sinks [40].

In general terms, the design of the network aims at maximising the symmetry between the branches and leaves to reduce the impact of static time errors. In parallel, the CDN design must be compatible with the operation clock frequency, preserve the quality of the waveform (not deteriorate the slope or slew of the signal), fit in the available area and adjust to the power budget allocated for this purpose.

9.2 Open-loop CDN architectures that reduce the impact of skew

The most common strategy to reduce the impact of skew is to balance the various clock paths with a symmetric layout. This yields network structures based on clock trees, H trees, grids, spines and variations of these structures [108] [88] [136] [87]. Figure 9.2 compiles some examples of such networks. In this figure, the sinks are represented as dots.

These networks usually include clock buffers or other repeaters (such as inverters) in the various levels of the hierarchy to preserve the slope of the clock signal. Networks purely based on interconnects provide a lower power consumption, but the degradation in the shape of the waveform is only affordable for small chip areas (below few square millimeters) and low clock frequencies (tens or few hundreds of MHz).

In the strategies of Figure 9.2, for a given corner, the latency to all sinks could be matched within the FastICpix time error budget, depending on the area to be served, fabrication mismatch, etc. But such latencies cannot be preserved across PVT corners: the symmetry of the network enables that the skew between the sinks is maintained for the various corners, which is not the case of the absolute value of

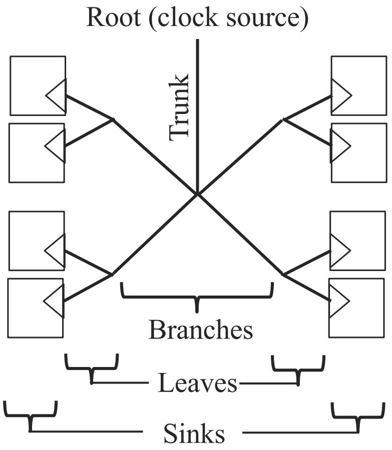


FIGURE 9.1: Basic structure of a CDN.

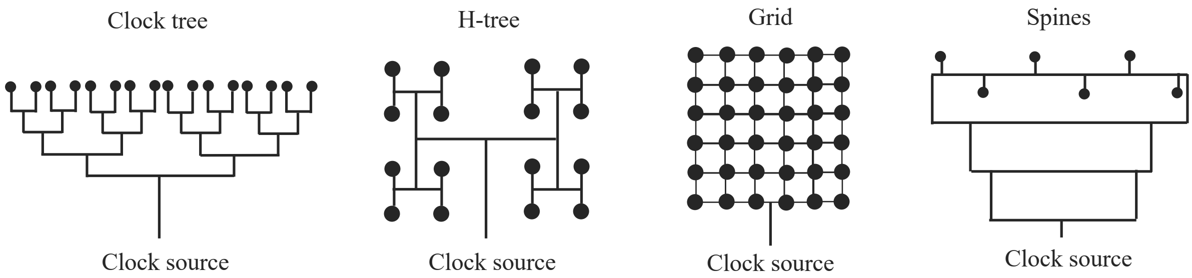


FIGURE 9.2: Examples of open-loop CDN architectures.

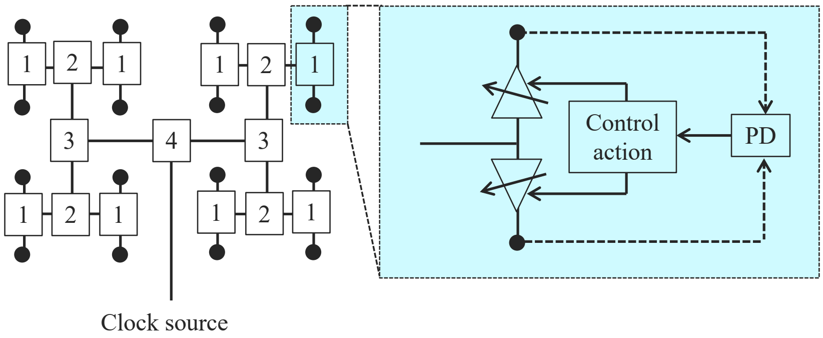


FIGURE 9.3: Active de-skewing implemented in an H-tree by adding phase detectors at the intersecting nodes and adjustable delay repeaters (adapted from Fig. 2.45 in [136]).

the latencies. The divergence in the latencies across PVT can easily exceed the time error budget for FastICpix, so such designs are not suitable for the present work.

9.3 Self-regulated CDN architectures to reduce the impact of skew

There are architectures that combine the aforementioned strategies with adjustable delays, which are automatically regulated to adapt to the divergences in the delay of the paths associated to PVT variations. These architectures can potentially achieve a lower skew and preserve the latencies across PVT variations, at the cost of a larger complexity and power consumption.

Several alternatives exist to implement active de-skewing. For instance, a phase detector can be embedded in the intersecting nodes of the architectures mentioned in Figure 9.2, the output of which determines whether the delay to a certain branch should increase or decrease with respect to that of the branch used as a reference. In the example shown in Figure 9.3 [136], a PD is embedded in every level of an H-tree hierarchy (numbered as 1 to 4). In the phase detector corresponding to level 1, the clocks delivered to the sinks served from this level are compared. If one of the two paths happens to be faster, the delays of the leaves are updated to bring such clocks in phase. Once this is accomplished, the same operation is performed at level 2 of the hierarchy, and so on.

This architecture has a significant area and power overhead compared to a regular H-tree, since a phase detector and the associated control logic is implemented per level of the tree hierarchy. In addition, due to the delay associated to the control action, it might occur that sinks that are in the vicinity of each other, but they are served from different hierarchical branches, could end up showing skew if the control action in one of the branches is slower with respect to the other.

The area and power overhead of the above solution can be reduced with the scheme presented in [140]. In this approach, the clock distribution is divided into global, regional and local hierarchical levels, and the de-skewing action is performed at the regional level. Such an action consists of matching in phase a reference clock to the clock distributed to each region, which is generated by a PLL that shares the same clock reference.

[83] proposes distributing the de-skewing action across the circuit area by using multiple PLLs. A PLL is implemented in every branch of the network, which synchronises the phase of the clock in that branch to that of the neighbouring branches. Note that synchronising neighbouring oscillators provides a reduction not only in skew, but also in jitter.

A similar principle is applied in [137] [1]: local oscillators distributed in a matrix across the chip area are mutually coupled to force injection locking [107] at the fundamental frequency. As a consequence, after a certain time, all of the oscillators end up showing the same oscillation frequency and phase, thus cancelling the skew and jitter time errors.
The coupled oscillators configuration has the potential to fulfill the FastICpix requirements: it supports skew and jitter cancellation (note that the lowest reported skew is about twice as much as the allowed time error in FastICpix), and it can be scaled in area (more oscillators would be coupled to cover larger areas) and pixel pitch (more oscillators could be coupled to serve a larger number of pixels). However, in terms of power consumption it may be less suitable than the selected configuration: the coupled oscillator approach requires that the oscillators are active all the time, which could lead to an excessive power consumption for the large chip areas envisaged, while the selected configuration is based on buffers, which limits the power consumption.

At this point, a few numerical examples will be provided to understand how the coupled oscillators approach would have a higher average power consumption than the selected configuration.
The power reported in [137] for one of such oscillators operating at a nominal frequency of 500 MHz is in the order of 500 µW. If we assume that such a consumption is dominated by the switching component, which scales linearly with frequency [103], the oscillator should have a consumption close to 80 µW when re-designed to operate at 80 MHz, and 40 µW when re-designed to operate at 40 MHz, which is the range of master clock frequencies that will used when the FastICpix area is scaled from a few square millimeters to a few square centimeters, respectively (as it will be introduced in Section 11.6).
With a 376 µm pixel pitch, the aforementioned range of areas will correspond to 8

[1] In this reference, the local oscillators of a matrix of TDCs are mutually coupled to be synchronised. Such TDCs can be used to time stamp the incoming photons, which would require a very high frequency of the local oscillators in order to provide a fine timestamp resolution. This would in turn lead to a prohibitive power consumption for large chip areas. Instead of synchronising TDCs, the explanation provided above refers to the synchronization of low frequency local oscillators, which provides an alternative distribution of the master clock signal that is robust to skew and jitter.

× 8 pixels to 64 × 64 pixels. If the same master clock frequencies used for the selected FastICpix CDN configuration are applied to the coupled oscillators approach and one oscillator is used per pixel, the expected power consumption will range between 5 mW (8 × 8 pixels × 80 µW) and 164 mW (64 × 64 pixels × 40 µW).
These values are roughly an order of magnitude higher than those reported in Section 12.2 for the selected CDN configuration. Nevertheless, the following options to reduce the power consumption of a configuration based on coupled oscillators must be mentioned:

- The use of different master clock frequencies is not required in the coupled oscillators configuration. Therefore, the 40 MHz option could be used for the whole range of chip areas, which could reduce to a half the aforementioned consumption for the smallest chip area.

- Instead of using one oscillator per pixel, several pixels could share an oscillator and be individually served by means of a local clock tree starting at the oscillator output. Local clock trees are used in the selected CDN configuration as well, since they offer the flexibility to scale the network with the pixel pitch without modifying the design of the main network blocks. In this case, they would help to limit the number of oscillators and thus the consumption associated to them.

The use of DLLs is preferred for the master clock distribution in this work [5] [63] [1]. In particular, the CDN of the Timepix4 [77] hybrid pixel detector, developed in the microelectroncis section at CERN, has been adapted to comply with the FastICpix requirements.
In Timepix4, the branches of the CDN are composed of dDLLs, which have the structure depicted in Figure 9.4. The clock from the clock source (`ckin_up`) is propagated across a DCDL composed of 32 ADBs.
Half of the ADBs (U0 to U15) propagate the clock upwards in a column of pixels, while the other half (D15 to D0) propagate it downwards in an adjacent column of pixels. The dDLL spans across half the chip height, so it is mirrored vertically to serve the other half of the chip; and it is repeated in the orthogonal direction to serve all the columns in the chip width.
At the output of every Ux stage, a local clock tree delivers the clock to the group of pixels associated to that stage.

The adjustable delay of the ADBs includes a coarse section (LSB of a few tens of ps) and a fine section (LSB below ps), which are regulated with the digital lines named `ctrl_bits_coarse_gray` and `ctrl_bits_fine_gray`, respectively.
Ideally, the clock should traverse the DCDL in 1 master clock period, regardless the PVT corner. If this condition is fulfilled, the clock arrives at each ADB (and thus to the associated group of pixels) with a known latency, which 1) causes a known offset in the time measurement of the groups of pixels served from different ADBs, which can be compensated offline; and 2) spreads the activation of the target logic during one clock period, thus preventing power spikes (and the associated jitter).

The PD compares the input and output (`ckout_down`) clocks of the delay line to verify whether the total delay is one master clock period. If `ckout_down` arrives later than `ckin_up` (the delay of the line is larger than the master clock period) by a time

difference larger than the sensitivity window of the PD (few hundreds of ps), the `DOWN` output will become high, thus indicating to the controller that it should decrease the delay of the line. Conversely, if `ckout_down` arrives earlier than `ckin_up` (the delay of the line is lower than the master clock period) by a time difference larger than the sensitivity window of the PD, the `UP` output will become high, thus indicating to the controller that it should increase the delay of the line.
Regardless the sign of the time difference, if it is comprised inside the sensitivity window of the PD, both `UP` and `DOWN` outputs will be low to prevent the activation of the controller (the dDLL is in lock).

To update the delay of the line, the controller can change either the coarse section of all ADBs simultaneously, or the fine section of all ADBs simultaneously. The last causes a change of 200 ps in the total delay of the line, thus providing an adjustment of the network latencies in steps of the Timepix4 TDC time bin.
Also as a result of such an adjustment step, the total delay of the line or latency until D0 will be 1 master clock period +/- 100 ps, which means that the static time error [2] at the output of D0 is bound to the TDC time bin.
In this dDLL design, since all stages share the same control bits, such a time error can only be due to PVT variations and layout imbalances. As the clock propagates through the line, it accumulates the static time error of each stage, so the worst case is observed at the output of D0, and as explained above, it is bound to 200 ps.
Since only the outputs of ADBs U0 to U15 are used to drive the local clock trees, the worst case static time error seen by the target TDCs occurs at the output of U15 and it will be much lower than 200 ps (ideally, it should be bound to 100 ps). In conclusion, the latencies to the ADBs can be adjusted in steps comparable to the Timepix4 TDC time bin and the worst static time error along the delay line is lower than the TDC time bin.

The master clock is delivered from the clock source to the different branches by means of a classic clock tree. The highest skew between the different dDLL inputs (which occurs between the dDLLs located at the leftmost and rightmost ends of the chip width) has been measured with simulations as a few tens of picoseconds, which means that the worst static time error among all branches is also lower than the Timepix4 TDC time bin.

9.4 CDN architectures with jitter attenuation

Jitter can be alleviated by preventing noise coupling (shielding clock wires from adjacent signal wires) and by low-pass filtering strategic signals (control signals in architectures with an active control action). Preserving sharp edges in the clock signal (short rise and fall transition times) helps in reducing the impact of jitter.

To reduce the noise present in the power supply, the use of sufficient decoupling capacitances is mandatory. Guard rings and deep n-wells in the substrate can help

[2] Here the static time error refers to the skew that cannot be compensated offline: it is not due to the nominal propagation delay that the ADBs introduce for a certain value of control bits, but due to layout mismatch, PVT variations, etc.

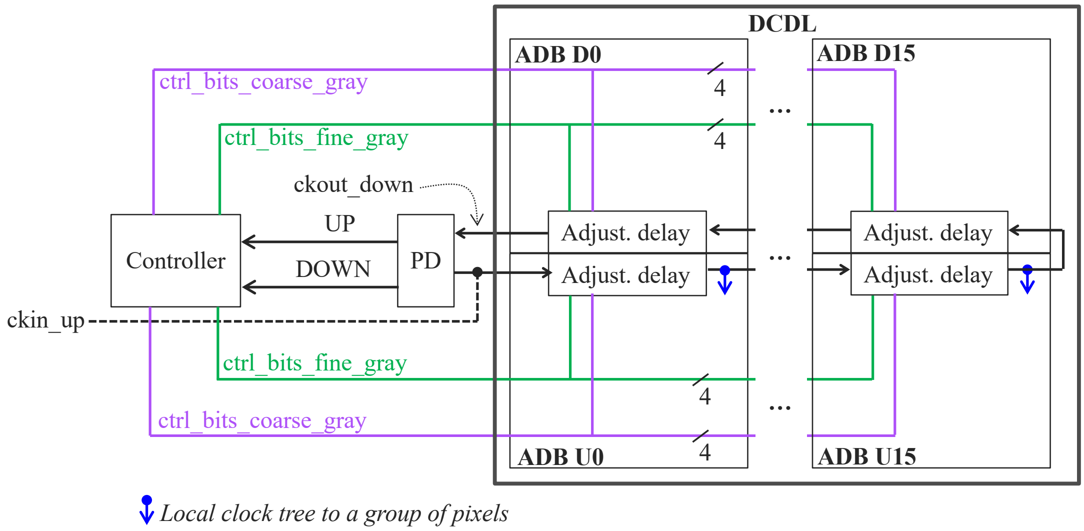


FIGURE 9.4: Structure of the dDLL implemented in Timepix4 for the master clock distribution.

to shield the sensitive circuits from those that generate the noise. In a digital design flow, routing halos can be used to discourage the signal router from creating long parallel wires (where switching is expected) adjacent to design macros, so as to prevent that the first injects switching noise on the second. Buffers can be used to isolate blocks with the same purpose [12] [103] (Chapter 10).
Using a differential clock distribution improves noise immunity and helps to achieve symmetric duty cycle halves [136] [81], although it also increases the complexity of the network.

In Timepix4, the outputs of the phase detector are digitally low-pass filtered to reduce the fast variability observed in these signals [77], which is associated to the presence of jitter in the PD inputs. By reducing such a variability in the outputs, the action of the controller is aimed at correcting the average delay of the DCDL or, in other words, the static time errors of the line. Without the low-pass filter, when the delay is close to 1 master clock period, but changing dynamically due to the presence of jitter, the outputs of the PD would lead the controller to toggle continuously between incrementing and decrementing the delay of the line in one step of adjustment, thus leading to unnecessary power consumption and preventing the achievement of lock.
The impact of jitter on the PD outputs and the benefits of using a digital low-pass filter will be discussed in Section 11.3.

As introduced in the present and the former sections, the Timepix4 CDN architecture includes mechanisms to compensate for skew (the network latencies can be automatically adjusted in steps close to 100 ps) and to reduce the impact of jitter (the

arrival of the master clock is distributed in time across the pixel matrix to prevent PSIJ; and the outputs of the PD are digitally low-pass filtered to avoid that the fast variability due to jitter is translated to the controller).
In view of these features, such a CDN was considered as the starting point to design the FastICpix CDN. However, in order to comply with the FastICpix specifications, a new controller design and strategy to update the network latencies were required, so as to achieve a finer resolution in the regulation of the network propagation delays and to guarantee that the static time errors of the network were lower than 20 ps. Besides, the network had to be scalable with the chip area and pixel pitch, so different values of DCDL length (and consequently, a range of master clock frequencies) had to be supported. Finally, a PD with a finer time resolution had to be developed.
The characteristics and implementation details of the FastICpix CDN will be explained in Chapter 11.

9.5 Innovative CDN architectures outside of the scope of this work

Further alternatives for clock distribution exist, such as optical clock distribution, wireless clock distribution, package level clock distribution, resonant clocking and 3D clock trees.

Optical networks require no electrical clock interconnect or buffers (so there is no jitter due to noise coupled from the power supply, the electronic noise of electrical components...), and there is no restriction on the maximum clock frequency that can be generated. The clock source is an on-chip or off-chip photon source; these photons are propagated either through a guided-wave or free-space; there may be a passive diffractive optics device to redirect the light; finally, the photons reach a photodetector for optical-to-electrical conversion and the resulting low-level photocurrent is amplified.
Despite its advantages, this type of network poses manufacturing challenges; and the main concern for pixel detector applications is the optical power that can be coupled to unwanted regions if not properly masked, which could lead to fake signal detection [90].

[105] describes a wireless CDN fabricated using a CMOS process. A wave of 24 GHz is received from an off-chip antenna. This becomes the global clock signal, which is then distributed to on-chip antennas located over the area being synchronised. At these receivers, the picked-up signal is amplified and divided in frequency to provide the local, 3 GHz clock signal. At the sinks, the total skew due to gain and phase variations should be less than 3% of a clock period, while in a traditional CDN the usual skew/jitter tolerance is 10% of a period.

In a package-level clock distribution, high-speed package routes replace the on-die clock network, which should provide a lower propagation delay and reduce the impact of interconnect variability due to fabrication. Its main limitation is the difficulty to test such a network and to implement the interface between the package network and the on-die receivers [78].

In resonant networks, on-chip inductance is used to form a resonant circuit with the interconnect capacitances. The power consumption can be lower in this case than in a CDN implemented with only repeaters (buffers or inverters) for two reasons: on the one hand, the energy of the clock fundamental component resonates back and forth between electric and magnetic forms instead of being dissipated as heat, so the clock drivers do not need to supply the energy to overcome such losses; and since the required power is lower, fewer repeater stages are needed, which contributes to reduce the power consumption even further, as well as to reduce skew and jitter [21] [97]. However, the design and implementation of such networks is more complex than with the traditional repeater-based networks.

Finally, in 3D clock distribution the CDN spans across multiple physical planes, which are electrically connected by TSVs. This solution has the potential to achieve a lower (average) skew than traditional, 2D configurations, when the suitable 3D clock tree topology and number of planes are used; but [138] reports a larger sensitivity to process-induced, within-die variations. Despite the promising results, the present work is focused on a 2D technology (TSVs are for the time being used to interconnect multiple devices, not multiple layers belonging to the same chip).

10 On the simulation conditions

In the following sections, numeric examples of the dDLL performance are provided. A large chip area with 376 µm pixel pitch (which yields a chip of 64 × 64 pixels, 2.4 × 2.4 cm^2) has been chosen to implement the demonstrator dDLL, since in such a scenario the CDN is more complex and thus there are more contributors to the overall time errors. The network architecture can be applied to smaller chip sizes, for which an even better timing performance could be expected. A commercial 65nm process will be used, with 1.2 V as nominal voltage supply.
From this point and unless stated otherwise, the DCDL is composed of 32 ADBs, one per group of 4 pixels, and the master clock frequency is 40 MHz. Three PVT corners have been used to evaluate the performance of the dDLL: slow (125 °C, 1.08 V, SS), typical (25 °C, 1.2 V, TT) and fast (-40 °C, 1.32 V, FF).

The timing results have been obtained from a digital simulation of the post-layout netlist of the dDLL, flattened (thus including the load effects from the interconnection of the different blocks), back-annotated (taking into account the actual propagation delays of all cells and interconnects), with all timing checks enabled (i.e. including the non-idealities of the sequential logic, such as the setup-and-hold window of the critical flip-flops in the PD). More details on the last point are provided later in this chapter.

Concerning the power consumption results, the total power consumption of the blocks is reported, which includes the switching power (consumed in the charging and discharging of interconnect capacitances), the internal power (consumed in charging and discharging of interconnect and device capacitances internal to the cells), and the leakage power (consumed by devices when they are not switching) [18]. In the dDLL, switching power is the dominant contribution: it stands for close to the 90% of the overall power consumption.
The total power consumption has been obtained performing a static, vector-based power analysis with Cadence® Voltus™[17]. A Value Change Dump (VCD) file has been used to annotate the nets with the actual switching activity of the circuit [94], which stands for the probability that a net toggles (it has a falling or a rising edge) in one clock cycle. For instance, an activity of 1 means that the net toggles every clock cycle, while an activity of 0.5 means that the net toggles once every second cycle. The larger the switching activity, the higher the resulting switching power consumption. A VCD file has been generated for every corner and for different values of standard deviation of the random, Gaussian jitter superimposed to the input clock of the DCDL, as it was introduced in Chapter 8. Such files contain the activity of the circuit from the time when an asynchronous reset is applied until the simulation finalises, when lock has already been achieved. All files correspond to the same number of cycles, so as to compare the consumption associated to each scenario.

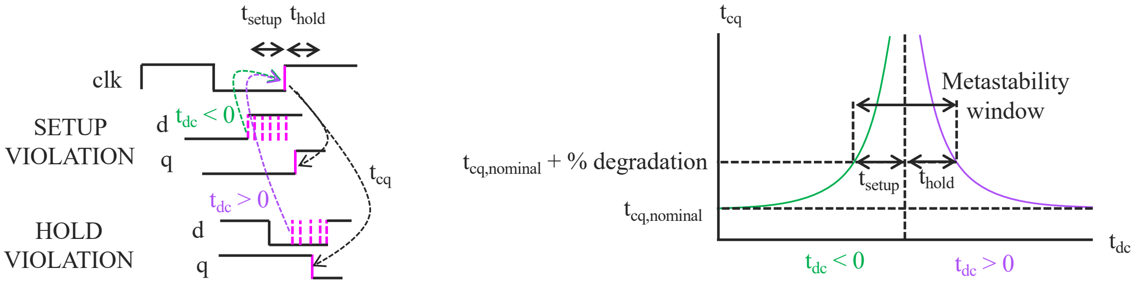


FIGURE 10.1: Illustration of the metastability or setup-and-hold window of a FF.

10.1 Setup-and-hold window of the FFs

Standard-cell D-type FFs are used in the phase detector to sample the time difference between the input and output clocks of the DCDL. This time difference can have an arbitrary value, even smaller than the ADB LSB, which is a few ps. For such small time differences, setup and hold time violations can occur in the FFs.

For a FF triggered with the rising edge of the clock, a setup violation occurs when the data input changes before such a clock edge by too small a time difference, smaller that the setup time, t_{setup}. This is indicated as 'SETUP VIOLATION' in Figure 10.1, and the time difference between the edges of the data and the clock input is denoted as t_{dc}. A hold violation (indicated as 'HOLD VIOLATION') occurs when the data input changes after such a clock edge by too small a time difference, smaller than the hold time, t_{hold}.

For time differences smaller than the metastability or setup-and-hold window (which is the addition of the setup and hold times), three scenarios can occur: 1) the output acquires a stable value that corresponds to the level of the data input; 2) the output does not change, so it might happen that the data input value is not reflected on the output; and 3) the output becomes metastable, a condition where the output voltage is neither high nor low for some time, and eventually resolves to a stable logic 1 or 0. Entering a metastable state is a probabilistic function related to the clock frequency and the rise/fall time of the data input. If metastability occurs, the time required for the output to converge to 1 or 0 cannot be foreseen, as it cannot be known in advance which of the two values will occur [22].

The setup and hold times can be quantified by means of the degradation in the propagation delay from the clock edge to the output edge (which occurs as a result of an edge in the data input), t_{cq} in Figure 10.1 [82]. Concerning the setup condition, the closer the data edge occurs before the clock edge, the larger will be t_{cq}, a longer resolution time will be required to propagate some value to the output. Analogously, for the hold condition, the closer the data edge occurs after the clock edge, the larger will be t_{cq}.

When the time difference between the data and clock edges is large enough, t_{cq} stabilises to its nominal value, $t_{cq,nominal}$. The setup and hold times are defined as the time differences between the data and clock edges for which t_{cq} has degraded by a certain percent with respect to the nominal value. The considered degradation is usually 5-10% [96]. Allowing for a low degradation provides a generous safety margin that guarantees that if setup and hold times are met throughout the logic chain, there will be no metastable signals involved. This conservative approach is useful to ensure the robustness of large chains of sequential logic.
Such a generous margin is not required in the case of the FastICpix PD, where there are only a few critical FFs. In this case, using narrower setup and hold times can help to enhance the time resolution of the PD, as it will be seen in Section 11.3.

Figure 10.2 shows the resolution time, t_{cq}, as a function of the time difference between the data and clock inputs, t_{dc}, for the three corners evaluated. This result corresponds to an analog simulation of the extracted netlist of the FF selected to sample the time difference of interest in the FastICpix PD. This FF has been chosen because it features the narrowest setup-and-hold window among the available alternatives, according to the standard cell datasheet. In such a simulation, the output of the FF is loaded with a capacitance of few fF, and the data and clock inputs have a rise time of 40 ps, which are the same conditions of operation it would have in the PD.
The simulation consists of sweeping the time difference between the data and clock input edges, which yields the input waveforms depicted in Figure 10.1. t_{dc} is measured at the points when the data and clock inputs cross 50% the voltage supply when commuting from low to high. t_{cq} is measured at the points when the clock input crosses 50% the voltage supply when commuting from low to high, and when the output of the FF crosses 50% the voltage supply, regardless the type of edge.
If the largest possible degradation of t_{cq} is allowed (until there is a gap in t_{cq} because no edge has been propagated to the output), the setup-and-hold window observed in this simulation ranges from 4 ps to 6 ps across PVT corners. Such a degradation is expressed as a percent of increment in the resolution time with respect to its nominal value in the figure.

Since the digital files that describe the timing performance of the standard-cell FF (liberty or .lib file) correspond to the aforementioned, conservative degradation, a re-characterization of the FF is performed using Cadence® Liberate Characterization Solution™[16], allowing for a more aggressive degradation in t_{cq}. A degradation of up to 100% with respect to its nominal value is allowed, according to the following reasoning:

- Even if such a degradation were actually to occur, it would mean that the result of the comparison between the PD inputs would be available at the output of the corresponding FF hundreds of ps after the arrival of its trigger clock edge. As it will be seen later in this chapter, the FFs in charge of comparing the PD inputs are embedded into 2-FF synchronisers. This means that the comparison of the PD inputs is performed in a first FF, then sampled after a certain time (to prevent that metastable values are propagated down the line) in a second FF, and the output of the second FF is used as the result of the comparison in further stages.

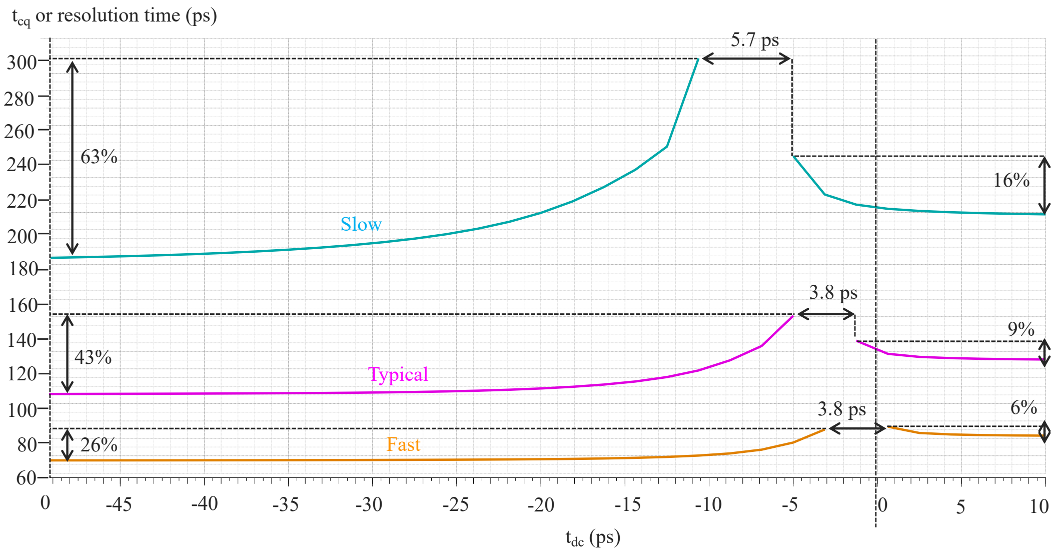


FIGURE 10.2: Resolution time as a function of t_{dc}, the time difference between the data and clock inputs.

The trigger clock edge of the second FF is a delayed version (it arrives a few ns later) of the trigger edge of the first FF. This time difference is at least an order of magnitude larger than such an extremely degraded t_{cq}, so t_{cq} is not the dominant factor in the computation of the slack between the first and second FFs of the synchroniser.

Two further observations can be made before reaching the conclusion of this point: 1) the 2 FFs that compose the synchroniser are placed within close proximity, so there is a small propagation delay between the output of the first and the input of the second FF); and 2) the master clock period is an order of magnitude larger than the longest delay mentioned in this point, which is the delay between the clocks of the first and second FFs.

With all this in mind, it can be concluded that even if a degradation of 100% is allowed in t_{cq}, there will still be a generous slack from the launch path in the first FF to the capture path in the second FF of the synchronisers ([12], Chapter 8), that is, the operation of the 2-FF synchronisers will not be compromised by allowing such a degradation in t_{cq}.

- Such an extreme degradation is not expected to be reached, as it can be seen with Figure 10.2. The re-characterization provides a timing description (in the format of a liberty file, .lib) featuring a narrower setup-and-hold window with respect to the original file, and which is closer to the behavior observed with the analog simulations (the re-characterization yields a setup-and-hold window that is approximately twice the value observed with the analog simulation, thus allowing for some safety margin as well).

The re-characterised description of the critical FFs of the PD has been used both for the implementation and the simulation of the detector. In addition, a modified Verilog model of such cells is used for simulation.

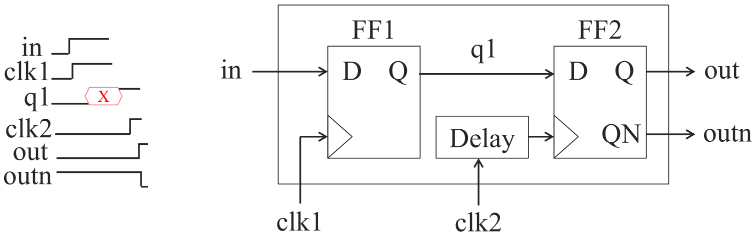


FIGURE 10.3: 2-FF synchroniser.

Further on the setup-and-hold window, with the conventional Verilog description of a standard-cell D-FF, when a timing violation occurs, its output becomes undefined or 'X', which represents the risk that there is a metastable value at the output. To illustrate that the metastable value can take an undetermined time to stabislise to a valid 1 or 0, 'X' is maintained until the end of the simulation (or until the FF is reset, if this option is available).

In this work, for a more realistic behavior, the Verilog model has been modified so that the output collapses randomly to 1 or 0 (if metastability were to occur, the final value of the output could be any of the both) after a certain time (hundreds of ps). The modification consists of replacing the Verilog D-FF primitive with a user-defined module that has the aforementioned behavior [104].

This modification does not represent the physical performance of the FF: the value selected for the delay after which the output collapses to a stable value when there is a timing violation does not correspond to any physical characteristic of the FFs. The purpose of this model is to evaluate the impact of metastability (i.e. the random return to 1 or 0 when sampling time differences smaller than the setup-and-hold window) on the time resolution of the PD.

10.2 Two-FF synchroniser

A common strategy to reduce the risk of propagating a metastable output further down the logic chain is the use of a synchroniser [47]. A simple 2-FF synchroniser is depicted in Figure 10.3. It consists of a chain of two FFs in which the first one has the risk of developing a metastable value at its output, `q1`; and the role of the second FF is to sample q1 after a certain time, when it should have collapsed to a stable value. A safe option is to sample `q1` one clock cycle later than it was generated.

In the PD, 2-FF synchronisers are used to sample the time difference between the input and output clocks of the DCDL. The timing description of the FFs in the synchronisers is the result from the aforementioned re-characterization performed with Cadence® Liberate Characterization Solution™, and the modified Verilog description is used, in which the output of the FFs returns randomly to 1 or 0 after a certain time when a timing violation occurs.

With this Verilog model, the time required for a metastable output to stabilise is well known, so `clk2` is a delayed version of `clk1` (the delay is dimensioned as a few ns to make sure that q1 has returned to a stable value after a setup or hold violation occurs).

11 FastICpix CDN architecture

11.1 Overview of the CDN architecture

In this work, the Timepix4 pixel detector CDN [77] has been adapted to comply with the FastICpix requirements, namely 1) scalability with the chip area (area across which the CDN spans) and pixel pitch (number of sinks or target TDCs), and 2) latency adjustable in steps finer than the TDC time bin and time errors due to the CDN at each of its sinks lower than the TDC time bin (20 ps).

The master clock source is an external clock reference. This signal is distributed by means of a clock tree to the different branches of the CDN, which are composed of dDLLs [63] [5] [139].
For large chip areas, the master clock source is located at the centre of the chip, as shown in Figure 11.1, from where it is spread towards the top and bottom halves of the chip. For small chip areas, the master clock is distributed from one side of the chip and the dDLLs span across the full chip height.

The dDLLs are in turn composed of a PD; a DCDL whose nominal delay is 1 master clock period, and which consists of ADBs; and a controller that provides the bits to regulate the delay of the DCDL, as shown in Figure 11.2. In this figure, the DCDL includes 32 ADBs, half of them guiding the clock upwards in the column of pixels (stages U0 to U15), and the other half driving it downwards in an adjacent column of pixels (stages D15 to D0). The output of each ADB drives a local clock tree to deliver the master clock to the TDCs in the corresponding group of pixels (4 pixels per ADB in this case).

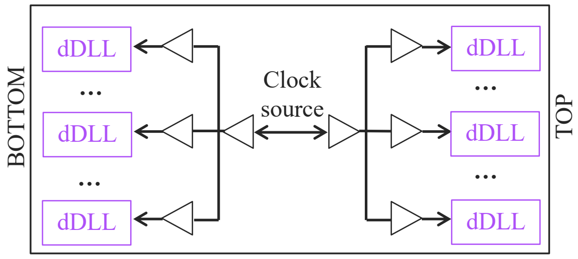


FIGURE 11.1: Sketch of the CDN architecture at the chip level.

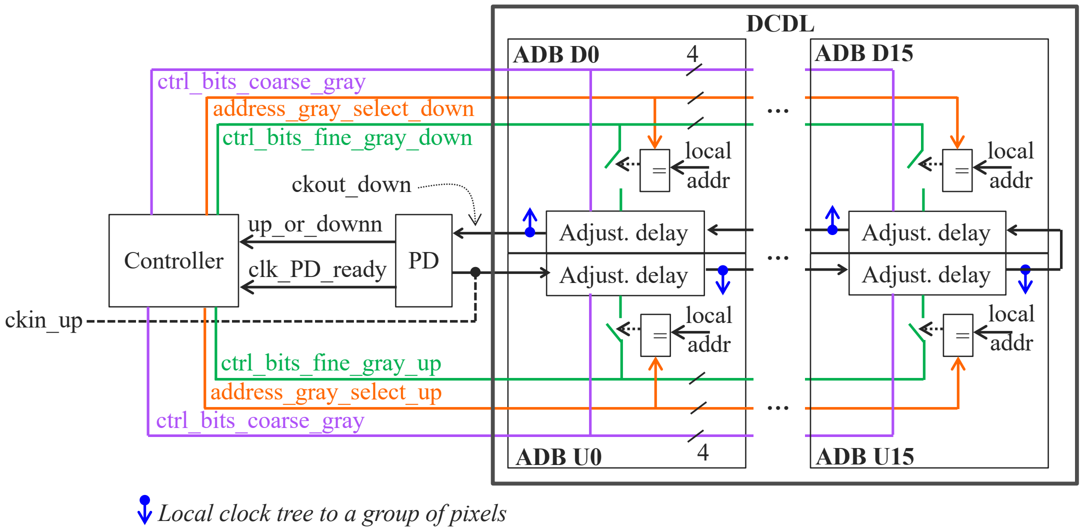


FIGURE 11.2: Structure of one dDLL.

The use of dDLLs enables the required flexibility to adapt to different chip areas by adjusting the number of stages of the DCDL, number of dDLLs in the CDN, the master clock frequency, etc.; and to different pixel pitch dimensions by acting on the complexity of the local clock tree that starts at the output of each ADB.

The PD compares the current rising edge of the clock entering the DCDL (`ckin_up`), which comes from the clock source, to the edge at the output of the DCDL (`ckout_down`). The second is the result of propagating the previous `ckin_up` edge across the line, so the current edges at the input and output clock lines of the DCDL should ideally arrive very close in time.
The operation principle of the PD is illustrated in Figure 11.3. If the output edge arrives earlier than the input edge (the delay of the line is shorter than 1 master clock period), the `up_or_downn` output is set to 1 so that the controller increases the delay of the line. In the case where the output edge arrives later than the `ckin_up` edge, the `up_or_downn` output is cleared to 0 to reduce the line delay.

The time resolution of the PD is close to 1 ADB LSB, and it changes accordingly over PVT corners. Only if the separation between the input and output edges is larger than +/- 1 ADB LSB (the smallest update in the delay of the line that the controller can produce), a pulse is generated at the `clk_PD_ready` output, and its rising edge triggers the synchronous state machine of the controller, thus enabling the update of the DCDL delay. The purpose of this sensitivity window is that no action takes place when the deviation of the total delay of the line with respect to the master clock period is smaller than the update that the controller can produce (otherwise,

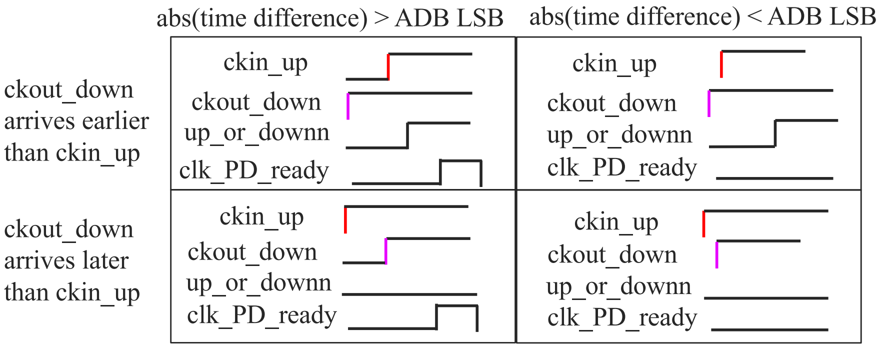

abs(time difference) > ADB LSB
abs(time difference) < ADB LSB
ckout_down arrives earlier than ckin_up
ckout_down arrives later than ckin_up
ckin_up
ckout_down
up_or_downn
clk_PD_ready

FIGURE 11.3: Principle of operation of the phase detector.

the system would continue to toggle between incrementing and decrementing the line delay by 1 ADB LSB, thus leading to unnecessary power consumption).

The outputs of the PD are low-pass filtered to prevent switching continuously between consecutive LSBs as a consequence of jitter. PSIJ is alleviated thanks to the operation principle of the dDLL: since the clock arrives at the ADBs spread over a clock period, the power peak associated to a simultaneous activation of the target logic is prevented.

According to the `up_or_downn` value, the controller will update the delay of the line by changing the control bits of the ADBs until the total delay is 1 master clock period +/- 1 ADB LSB. At that point, no further pulse will be generated at `clk_PD_ready`: lock is achieved.

The adjustable delays are composed of a coarse section (with a LSB of a few tens of picoseconds), which is updated simultaneously in all stages by means of the `ctrl_bits_coarse_gray` lines, and a fine section (whose LSB is roughly 7 ps), which can be regulated independently along the line. The individual adjustment of the fine sections enables that the total DCDL delay is updated in steps of a few picoseconds. To adjust the fine sections individually, the controller broadcasts the fine control bits and the address to be updated; the last is compared to the local address of each ADB and, if the comparison is successful, the value of the fine control bits is loaded to the selected stage.
If the stage to be updated is located in the DCDL half closest to the input clock, the `address_gray_select_up` address lines and the `ctrl_bits_fine_gray_up` fine control bit lines are used. If the stage to be updated is located in the DCDL half closest to the output clock, the `address_gray_select_down` address lines and the `ctrl_bits_fine_gray_down` fine control bit lines are concerned instead.
The control and address bits are distributed Gray-encoded to reduce the switching

power consumption, which is the dominant contribution to the total CDN power consumption.

As it has been mentioned above, the dDLL spans along half the chip height for large areas or the full chip height for small chip areas, thus providing a fine time resolution in the adjustment of the network latencies and low time errors along the columns it occupies.
In contraposition, the clock is distributed using a classic clock tree from the clock source to the inputs of the different dDLLs, which means that such a time resolution cannot be guaranteed between neighbouring dDLLs.
For large chip areas, the worst skew between the inputs of neighbouring dDLLs (which occurs between the dDLLs located in the leftmost and rightmost ends of the chip) can exceed the 20 ps time error target. This issue could be addressed by replacing such a clock tree by a horizontal dDLL that spans across the full chip width, as shown in Figure 11.4.
Each output of the first half of the horizontal DCDL would drive the input of a vertical dDLL located in the top half of the chip, while each output of the second half of the horizontal DCDL would drive the input of a vertical dDLL located in the bottom half of the chip.
The design and implementation of the horizontal dDLL is not covered in this work, but the guidelines provided for the vertical dDLL design can be applied.

In the coming sections, the architecture of the individual blocks of the dDLL is explained, and guidelines are provided to scale the architecture with chip area and pixel pitch dimensions.

11.2 Adjustable Delay Buffer and Digitally-Controlled Delay Line

The ADB architecture is sketched in Figure 11.5. Its input and output pins can be grouped in three main functionalities:

- Clock lines:
  - Input: `ckin`
  - Output: `ckout` (to the next ADB), `ckout_local` (to the local clock tree)
- Control bit lines:
  - Input: `ctrl_bits_fine_gray_in`, `ctrl_bits_coarse_gray_in`, `clk_update_fine_ctrl_bits_in`, `clk_update_coarse_ctrl_bits_in`, `clearn_to_min_in`, `setn_to_max_in`
  - Output: `ctrl_bits_fine_gray_out`, `ctrl_bits_coarse_gray_out`, `clk_update_fine_ctrl_bits_out`, `clk_update_coarse_ctrl_bits_out`, `clearn_to_min_out`, `setn_to_max_out`
- Address lines:

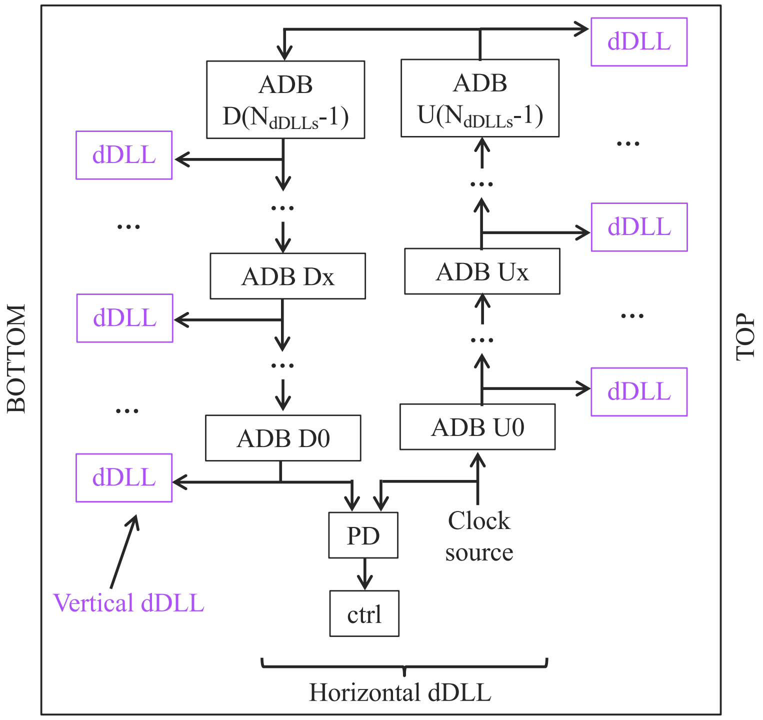


FIGURE 11.4: Vertical and horizontal dDLLs in the large chip area CDN, which guarantees that the time error target is met both along the column and between neighbouring columns.

  - Inputs: address_gray_select_in, address_gray_in
  - Outputs: address_gray_select_out, address_gray_out

If the ADB propagates the clock upwards in the column of pixels, ckin represents the clock coming from a stage closer to the clock source, while ckout sends the clock to the next stage, which will be closer to the middle of the DCDL.
If the ADB propagates the clock downwards in the column of pixels, ckin represents the clock coming from a stage closer to the middle of the DCDL, while ckout sends the clock to the next stage, which will be closer to the end of the DCDL.
ckout_local stands for the line that drives the local clock tree, which eventually distributes the master clock to the group of TDCs served from the present ADB.

The 4-bit fine(coarse) control bit lines that arrive at the ADB (ctrl_bits_fine(coarse)_gray_in) are buffered to propagate the control bits upwards in the column (ctrl_bits_fine(coarse)_gray_out). These lines are Gray-encoded, so they are translated to a thermometer code to drive the adjustable delay sections, which will be explained later on.
The control bits are loaded with the rising edge of an auxiliary clock, clk_update_fine(coarse)_ctrl_bits_in, which is also internally buffered and propagated upwards in the column (clk_update_fine(coarse)_ctrl_bits_out). A pulse is sent via the corresponding auxiliary clock line every time that the controller requests the update of the coarse or fine delay section. Since the auxiliary clock arrives later than the new value of control bits, this strategy ensures that a stable value of control bits is loaded.
In addition, the controller has the option to clear the fine control bits to their lowest value (0 in decimal code) by means of a falling edge of clearn_to_min_in, which is internally buffered and propagated upwards in the column (clearn_to_min_out); and it can also set the fine control bits to their maximum value (15 in decimal code) with a falling edge of setn_to_max_in, which is internally buffered and propagated upwards in the column (setn_to_max_out). These clear and set operations take place in all ADBs simultaneously and they are synchronous, so a pulse of the auxiliary clock should follow the falling edge in these lines.

The controller can access randomly the ADBs along the line to update their fine delay section individually. Every ADB in one of the DCDL halves has a 4-bit unique local address (opposite stages share the same address, such as D0 and U0, D1 and U1... and the controller keeps track of which of the halves is being updated).
To maximise the symmetry in the implementation of all stages, which prevents static time errors, the local address is generated by 'adding 1' (in Gray code) to the local address of the stage that is located right above in the column of pixels (address_gray_in). The resulting address is propagated to the stage located right below in the column of pixels (address_gray_out). Note that address_gray_in of ADB_X represents the local address of ADB_X, while address_gray_out of ADB_X represents the local address of $\text{ADB}_\text{X-1}$, i.e. the stage located below ADB_X in the column of pixels.
The local address of the stages located on top of the pixel column or, in other words, in the middle of the DCDL (D15, U15) is all zeros (the bits of address_gray_in are tied down).
The local address is compared to the controller request (address_gray_select_in),

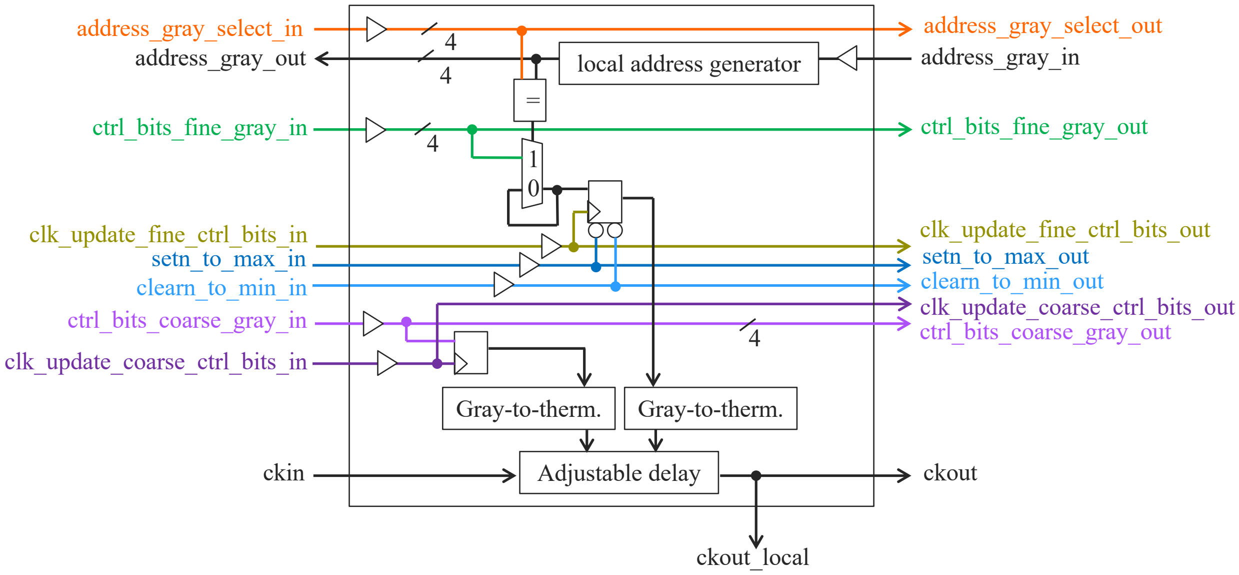


FIGURE 11.5: Architecture of the Adjustable Delay Buffer.

which is internally buffered and propagated upwards in the column of pixels (`address_gray_select_out`). Only if the result of the comparison is positive (and `clearn_to_min_in` and `setn_to_max_in` are high), the new value of fine control bits is loaded to the ADB.

The fine and coarse adjustable delay sections of the ADB have the structure depicted in Figure 11.6. The same fine and coarse cells designed for Timepix4 have been used here.

The fine section consists of one of such fine cells: the clock input is buffered and connected to a capacitive load tuned with the 15 thermometer lines resulting from the loaded 4-bit Gray-encoded fine control lines. When a given thermometer line is low, a LSB of capacitance is connected to the clock line, thus increasing the propagation delay of the fine section by approximately 5 ps in the typical corner.
The lowest propagation delay corresponds to the buffer propagation delay and it occurs when the Gray-encoded fine control bits are all 0 (the resulting thermometer lines are all 1), while the largest propagation delay is achieved when the Gray-encoded fine control bits are all 1 (the resulting thermometer lines are all 0).

The coarse section consists of 15 coarse cells, which are composed of NAND gates. The number of cells through which the clock propagates determines the delay introduced by the coarse section, and it can be tuned with the 14 thermometer lines

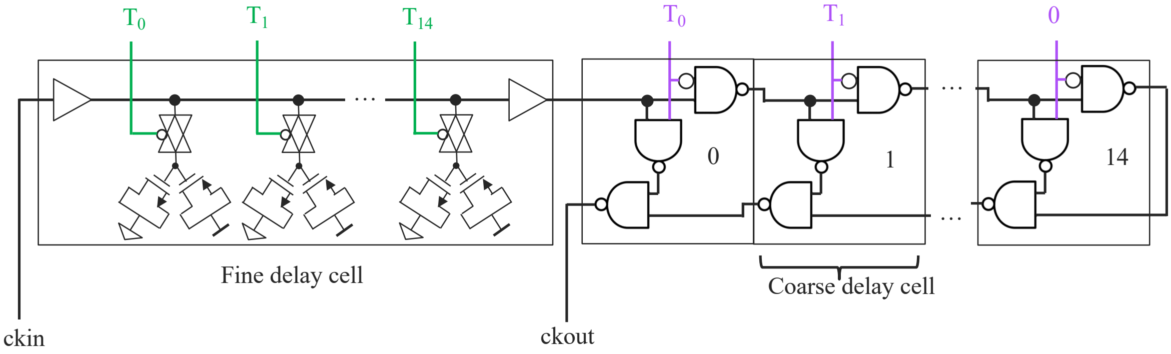


FIGURE 11.6: Structure of the fine and coarse sections in the ADB.

resulting from the loaded 4-bit Gray-encoded coarse control bits. If a given thermometer line is low, the clock is propagated towards the next coarse cell.
When all thermometer lines are low (largest coarse delay scenario, the coarse control bits have value 14 in decimal code), the clock propagates through all coarse cells; when it reaches the last cell, it propagates backwards, traversing all the cells once again before exiting the coarse delay section.
In any other scenario, the clock propagates through the coarse cells until it finds a cell with 1 as thermometer control bit. In that cell, it takes the vertical NAND and proceeds with propagating backwards in the chain of cells that it had traversed.
The smallest coarse delay is obtained when the coarse control bits have value 0 in decimal code. In this case, the first coarse cell (numbered as 0) has a 1 in its thermometer control line, so the clock takes the vertical NAND and exits the coarse section.
Every new coarse cell connected to the chain adds around 50 ps (in the typical corner) to the coarse delay contribution.

Figure 11.7 shows the evolution of the delay introduced by one ADB for the simulation conditions and the three corners defined in Chapter 10. The vertical axis stands for the propagation delay from `ckin` to `ckout`. The horizontal axis stands for the possible fine control bit values. The different colours are associated to the PVT corner, while the parallel lines sharing a colour correspond to the delay increments introduced by increasing the coarse control bits from their lowest to their highest possible value.
Figure 11.8 presents the same information as Figure 11.7, with a zoom in the vertical axis to ease visualization. The value of the coarse control bits corresponding to each of the parallel lines is indicated on the right (the colour code of the text corresponds to the PVT corner).
The slope of the lines, that is, the LSB of the fine section, is 3 ps in the fast corner, 5 ps in the typical corner and 7 ps in the slow corner.
The distance between parallel lines, that is, the LSB of the coarse section, is 32 ps in the fast corner, 50 ps in the typical corner and 80 ps in the slow corner.

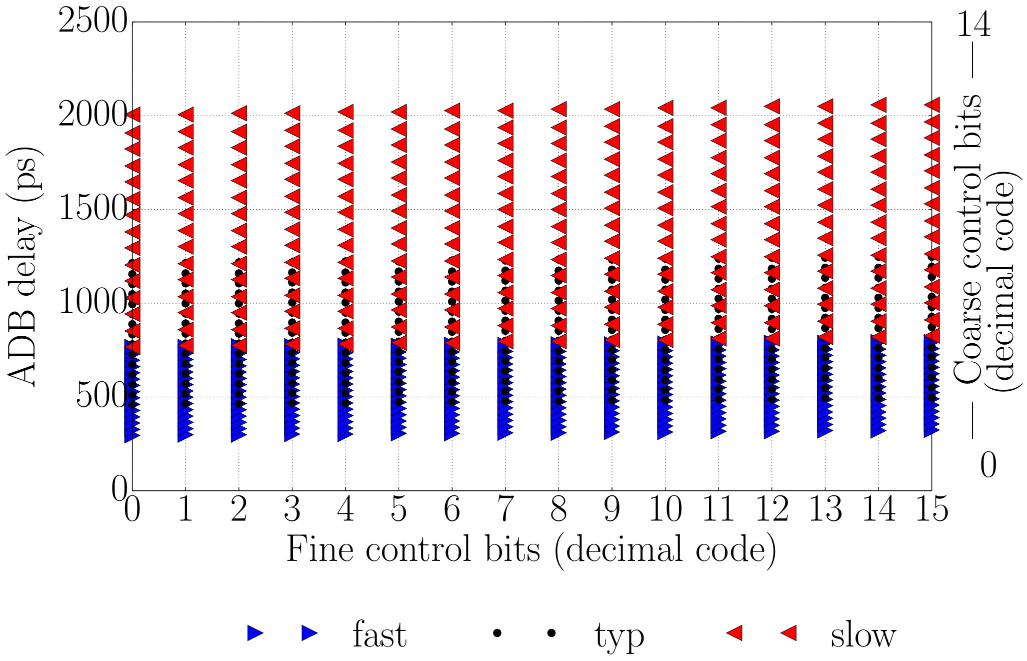


FIGURE 11.7: Evolution of the delay introduced by one ADB as a function of the fine and coarse control bit values.

Note that there is an overlap in the delay observed in the different corners, which enables allocating the master clock period within the range of adjustment for the three corners considered. In addition, for a given corner, there is an overlap in the delay observed for large fine control bit values in a certain coarse control bit point, and small fine control bit values in the immediately larger coarse control bit point. This overlap aims at compensating the possible non-idealities of the circuit (transistor mismatch, layout imbalance, etc.), so as to guarantee that the full range of delay adjustment can be provided.

When the 32 stages are interconnected to form the delay line, the resulting, total delay of the line as a function of the control bit values is shown in Figure 11.9. The vertical axis stands for the total DCDL delay. The colour map shown in the horizontal axis stands for the variation of the fine control bits from their lowest to their highest possible value, updating only one stage at a time. For each value of the fine control bits, stage U0 is updated first, followed by U1, etc. until D0 is updated. By changing the fine control bits in this fashion, the total delay of the line is incremented in steps of ADB LSB.

The different colours are associated to the PVT corner, while the parallel lines sharing a colour correspond to the delay increments introduced by increasing the coarse control bits from their lowest to their highest possible value (the coarse control bits are updated simultaneously in all stages).

Figure 11.10 presents the same information as Figure 11.9, with a zoom in the vertical axis to ease visualization. The value of the coarse control bits corresponding to each of the parallel lines is indicated on the right (the colour code of the text corresponds to the PVT corner).

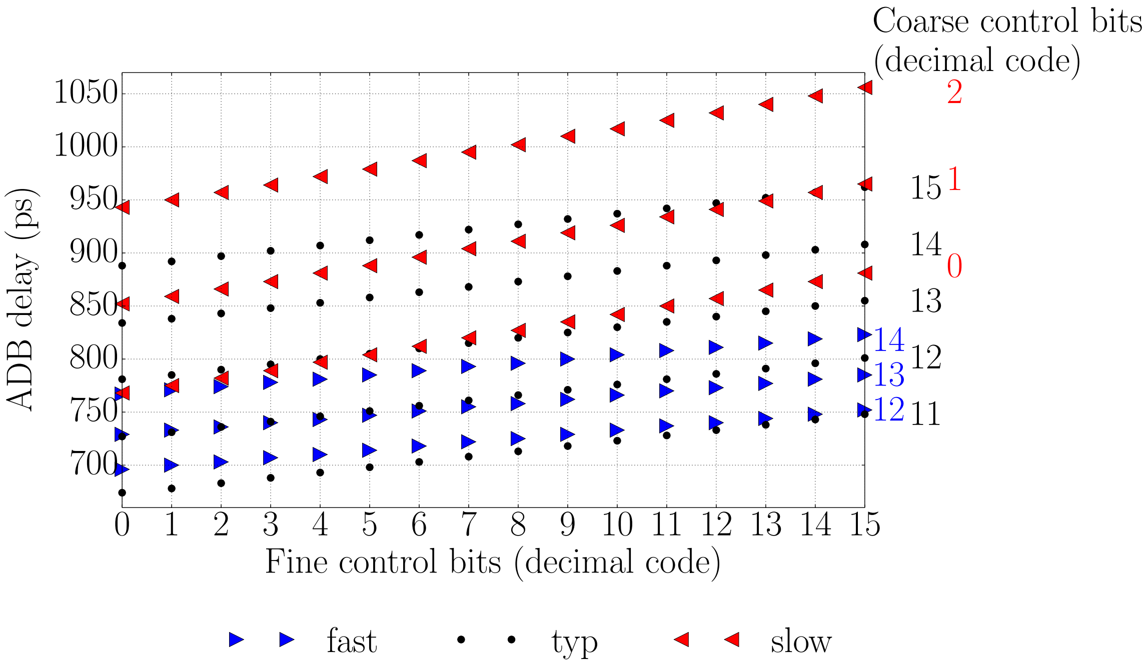


FIGURE 11.8: Evolution of the delay introduced by one ADB as a function of the fine and coarse control bit values (zoom in the vertical axis).

The slope of the lines matches the aforementioned ADB LSB, while the distance between parallel lines, that is, the update in the delay caused by incrementing in one unit all coarse control sections simultaneously, is the coarse LSB multiplied times the number of stages (32), which yields an update of roughly 1 ns, 1.6 ns and 2.5 ns in the fast, typical and slow corners, respectively.
As a consequence of the overlap observed in the ADB delay for consecutive values of coarse control bits, this overlap is also observed in the total DCDL delay, which provides robustness to eventual static time errors in the implementation of the DCDL and thus ensures that the full range of adjustment is accessible.

Towards the integration of the ADBs in the DCDL, a building block composed of two ADBs has been implemented. The pinout of this structure is shown in Figure 11.11.
To minimise static time errors, it is mandatory that all stages have a layout as similar as possible. In this 2-ADB structure, the stage propagating the clock upwards and downwards in the column of pixel have a mirrored layout, aiming at exploiting the symmetries between both paths. This building block is repeated 16 times to shape the DCDL.
In addition, the distance between `ckout_up` and `ckin_down` is minimised in the floorplan of this block, so that the interconnect required to shortcircuit `ckout_up` of stage U15 and `ckin_down` of D15 (that is, the line that connects both DCDL halves) is as short as possible, which prevents the introduction of static time error with respect to the rest of stages.

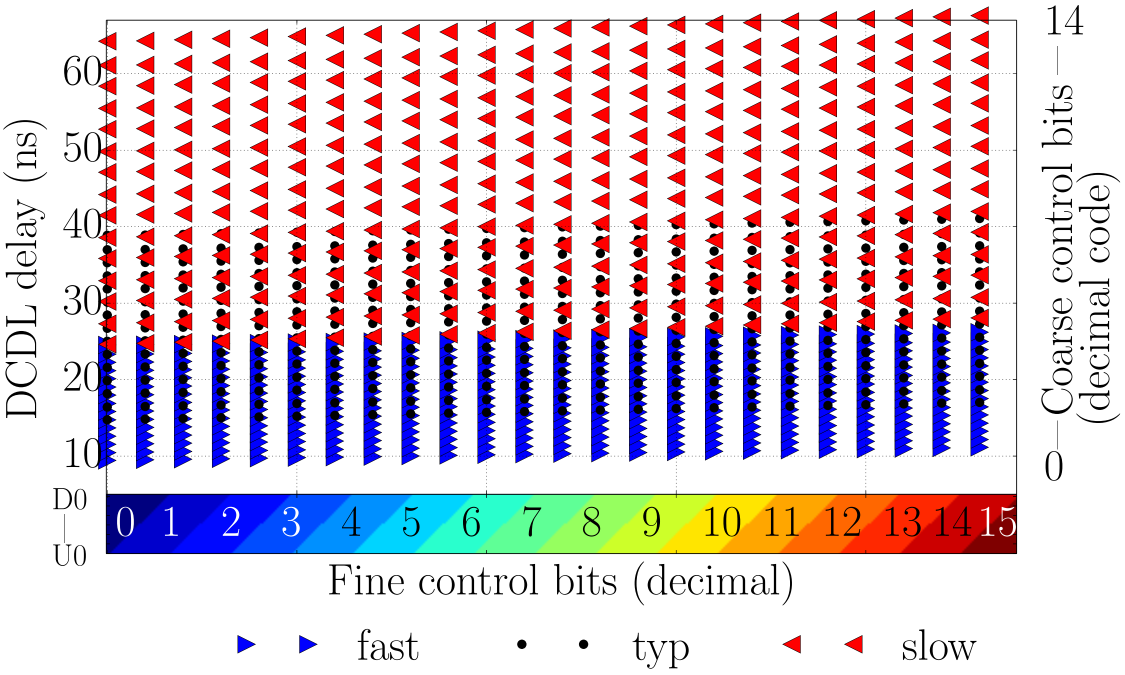


FIGURE 11.9: Evolution of the total line delay as a function of the fine and coarse control bit values.

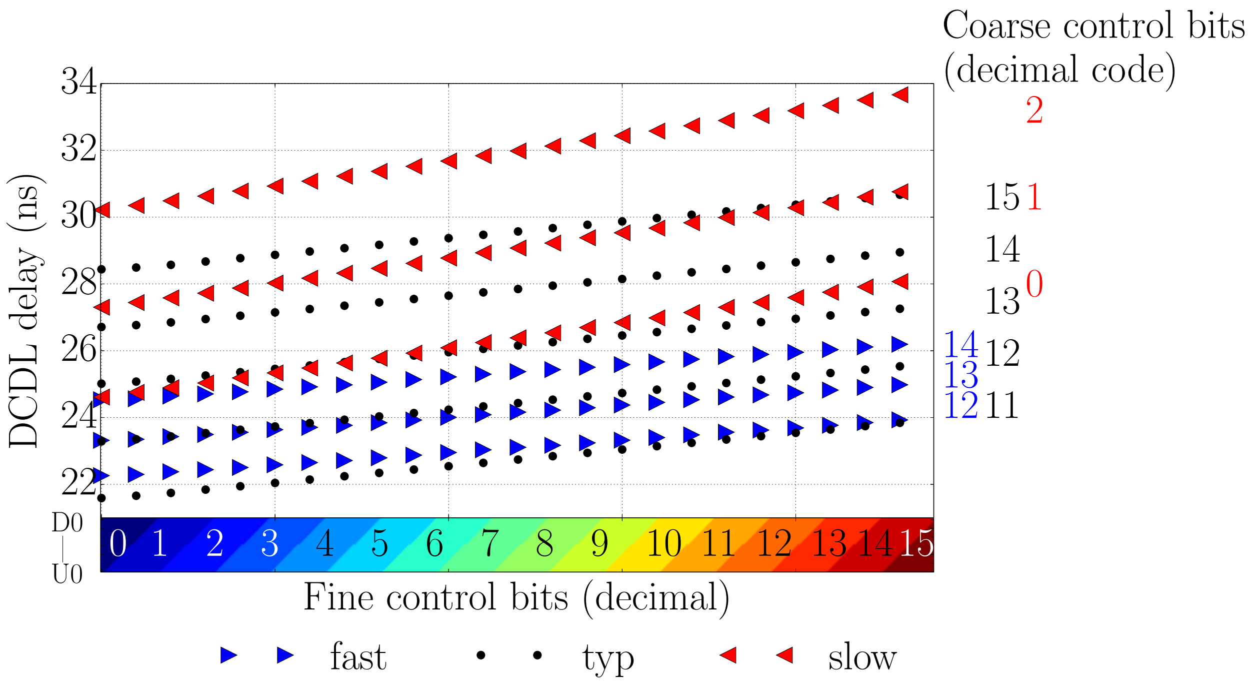


FIGURE 11.10: Evolution of the total line delay as a function of the fine and coarse control bit values (zoom in the vertical axis).

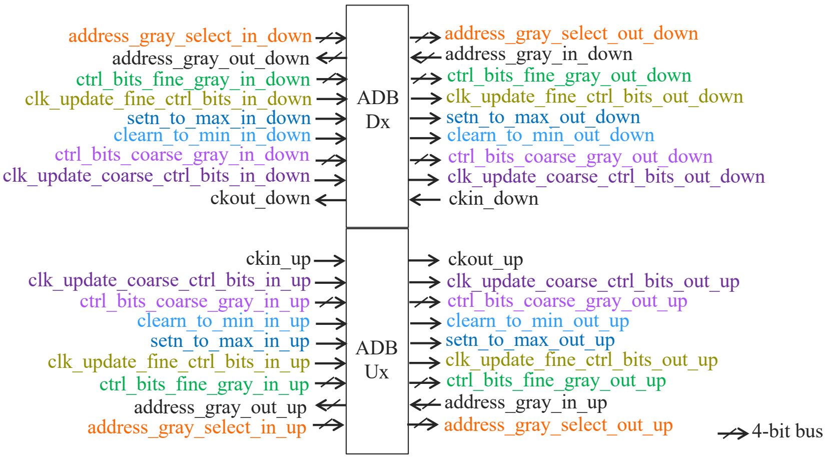

address_gray_select_in_down
address_gray_out_down
ctrl_bits_fine_gray_in_down
clk_update_fine_ctrl_bits_in_down
setn_to_max_in_down
clearn_to_min_in_down
ctrl_bits_coarse_gray_in_down
clk_update_coarse_ctrl_bits_in_down
ckout_down
ADB
Dx
address_gray_select_out_down
address_gray_in_down
ctrl_bits_fine_gray_out_down
clk_update_fine_ctrl_bits_out_down
setn_to_max_out_down
clearn_to_min_out_down
ctrl_bits_coarse_gray_out_down
clk_update_coarse_ctrl_bits_out_down
ckin_down
ckin_up
clk_update_coarse_ctrl_bits_in_up
ctrl_bits_coarse_gray_in_up
clearn_to_min_in_up
setn_to_max_in_up
clk_update_fine_ctrl_bits_in_up
ctrl_bits_fine_gray_in_up
address_gray_out_up
address_gray_select_in_up
ADB
Ux
ckout_up
clk_update_coarse_ctrl_bits_out_up
ctrl_bits_coarse_gray_out_up
clearn_to_min_out_up
setn_to_max_out_up
clk_update_fine_ctrl_bits_out_up
ctrl_bits_fine_gray_out_up
address_gray_in_up
address_gray_select_out_up
4-bit bus

FIGURE 11.11: Implementation of the ADB in the DCDL.

The following figure is proposed to evaluate the implementation mismatch of the ADBs in the line: the delay introduced by each ADB when the coarse control bits are 0 is shown in Figure 11.12. Each colour corresponds to a PVT corner, and the 16 dot rows observed for each corner correspond to the different values that the fine control bits can take (all stages share the same value of fine control bits).

The divergence in the delay of the stages due to layout mismatch and PVT variations is shown in Figure 11.13. This figure has been obtained from Figure 11.12 as follows: for a given corner and fine control bits value, the average ADB delay along the line has been subtracted from each ADB delay. The resulting ADB delay indicates by how much the delay of every stage deviates from the average ADB delay. The different colours correspond to the corner, while the various markers correspond to the available values of fine control bits (indicated in decimal code).
The relevant result to extract from this figure is that there is a systematic time error among the ADB stages, and the largest deviation between any Ux, Dy pair is bound to 5 ps in the slow corner, 4 ps in the typical corner and 2 ps in the fast corner. These values are comparable to the ADB LSB in each corner.

11.3 Phase Detector

A fully digital PD architecture has been selected, which is the most suitable for the digital-on-top approach followed for the dDLL implementation. The detection range of the PD is +/- half the master clock period [25]. Such large time differences may occur at the beginning of the dDLL operation, when the DCDL is reset to its lowest delay. But as the delay of the line increases to match the master clock period, the PD is expected to operate with time differences as small as +/- ADB LSB, for which lock is achieved.

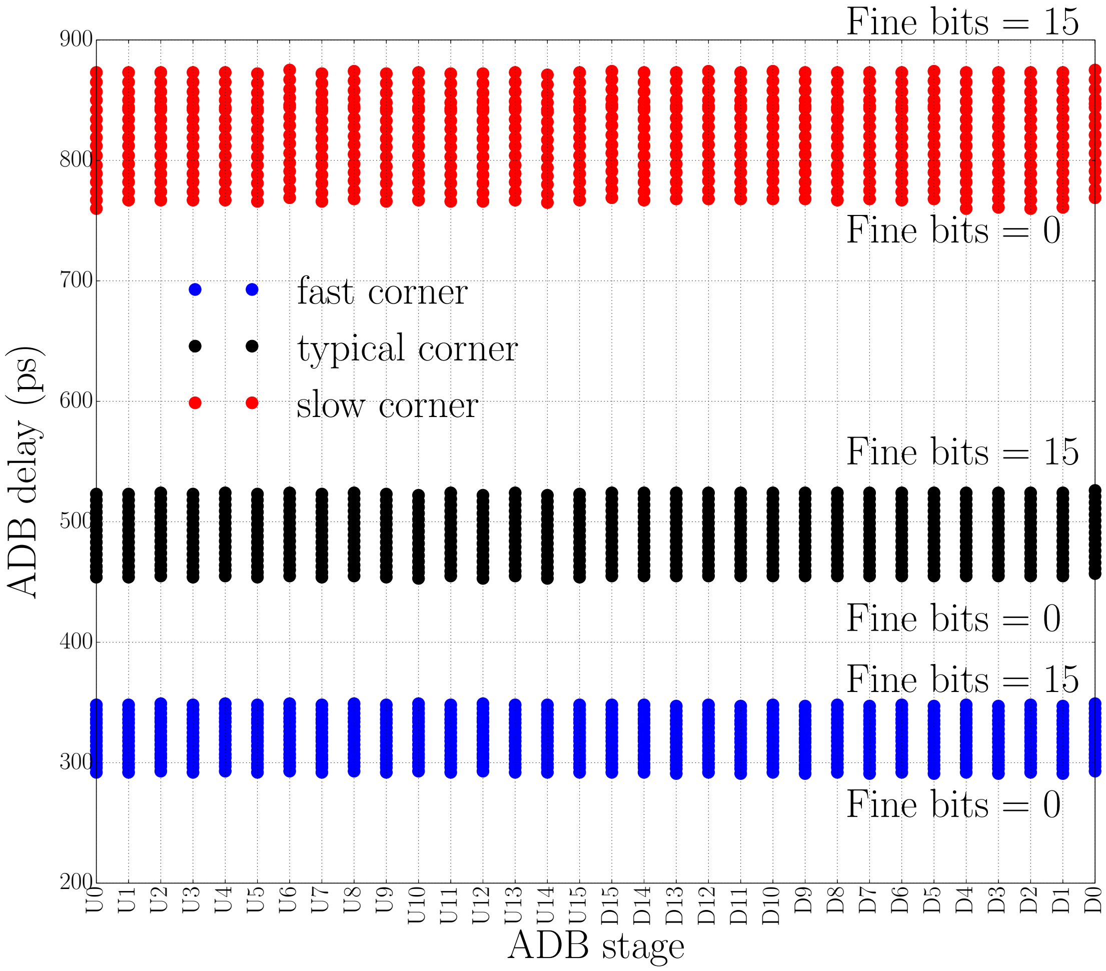


FIGURE 11.12: Delay introduced by every ADB stage when all the stages have the same value of fine and coarse control bits.

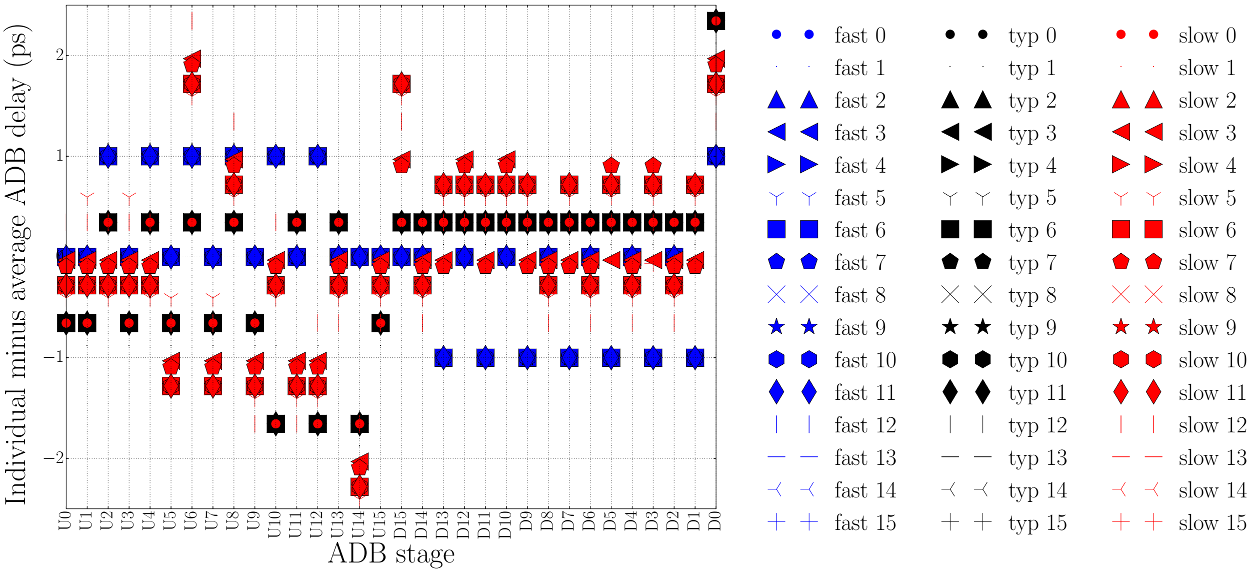


FIGURE 11.13: Delay introduced by every ADB stage when all the stages have the same value of fine and coarse control bits, minus the average ADB delay for every combination of coarse and fine control bit values.

Standard-cell FFs are used to sample the time difference between the input and output clocks of the DCDL [77] [122] [73]. With the small time differences expected when lock is to be achieved, there can occur setup and hold violations in these FFs. To prevent that metastable values are propagated down the line, the FFs are embedded into 2-FF synchronisers, as it was introduced in Chapter 10.

Concerning the smallest change in the line delay that should be detectable by the PD, it is dimensioned as ADB LSB for the following reasons:

- The delay of the line is updated in steps of ADB LSB, so a sensitivity as fine as that is required to track the changes in the delay of the line.
- When the total delay of the line deviates from the master clock period by +/- ADB LSB, the controller operation should stop, since lock has been achieved. Therefore, the PD must provide a sensitivity window of +/- ADB LSB, so that `clk_PD_ready` pulses are generated (and thus the controller is updated) only for deviations larger than that.
- By stopping the controller operation when the time error at the end of the line is +/- ADB LSB, the static time error at the ADB D0 when lock is achieved is bound to +/- ADB LSB, thus fulfilling the INL target that will be defined in Section 12.1.

The same fine delay cell used in the ADBs provides the +/- ADB LSB time sensitivity and enables tracking the variation of this magnitude over PVT corners.

An overview of the PD architecture is sketched in Figure 11.14. It comprises the generation of `up_or_downn_aux`, which indicates whether the delay of the line should

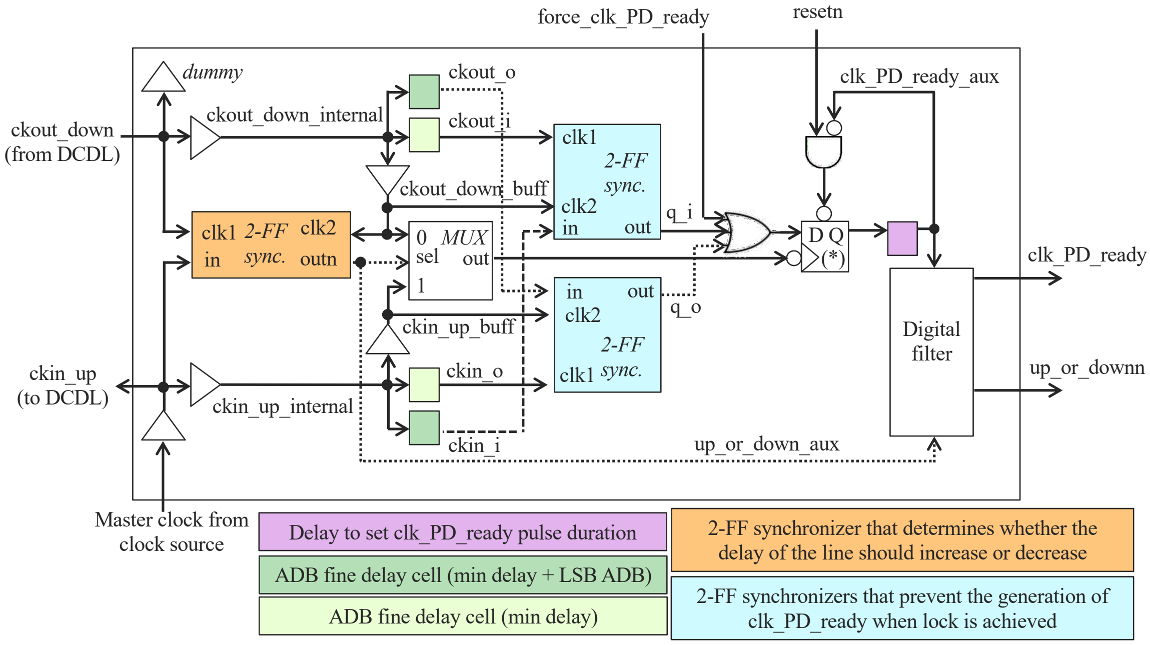


FIGURE 11.14: Architecture of the phase detector.

be increased or decreased, and `clk_PD_ready_aux`, which are the pulses that update the synchronous state machine of the controller. These signals show a fast variability due to the jitter superimposed to `ckin_up` and the timing violations that can occur in the 2-FF synchronisers that generate them (the output of the synchronisers returns randomly to 1 or 0 when a timing violation occurs).
This variability is reduced with a digital low-pass filter. Note that even if the output of the synchronisers does not have the expected polarity when a timing violation occurs, this error could be discarded, since a window of 16 cycles is processed at the digital filter before the PD outputs (`up_or_downn`, `clk_PD_ready_aux`) are provided to the controller.

Before explaining the architecture, a word on the PD-DCDL interface: it is crucial that the input and output clock lines of the DCDL are well balanced between them and that they have a similar parasitic load as the intermediate clock lines of the DCDL, so that no artificial phase difference is added on top of the time difference to be measured. With that purpose, the logic tapped to these lines is split into several clock buffer stages; lines are carefully routed to respect symmetries and dummy loads are added; and the master clock signal coming from the clock source is buffered before sending it to the DCDL via `ckin_up`.

11.3.1 Generation of `up_or_downn_aux`

The 2-FF synchroniser highlighted in orange in Figure 11.14 samples the time difference between `ckin_up` and `ckout_down`. If no error occurs, `up_or_downn_aux` will be

1 if ckout_down arrives earlier than ckin_up, and 0 if ckout_down arrives later than ckin_up.

11.3.2 Generation of clk_PD_ready_aux

The standard-cell FF labelled with an asterisk in Figure 11.14 generates clk_PD_ready_aux. Its output is cleared to 0 when an asynchronous reset is applied via the resetn pin. When the data input is high (i.e. at least one of these is 1: q_o, q_i, force_clk_PD_ready), a pulse is generated with the falling edge of the trigger signal. The rising edge of this pulse, delayed by a few nanoseconds to set the pulse duration, will reset the FF and force the falling edge of clk_PD_ready_aux.

When the dDLL is reset and the circuit is operating in the fast corner, the initial delay of the DCDL is lower than half the master clock period, which is outside the range of detection of the PD. As a consequence, up_or_downn goes low, while it should be high to indicate that the delay shall increase. To ensure that lock is achieved, the controller automatically increments the delay of the line until up_or_downn goes high, and from then onwards it acts according to the value of up_or_downn. But for the controller to do so, it requires clk_PD_ready pulses that force the update of the state machine.
To force the generation of such pulses in this initial regime, the controller sets force_clk_PD_ready high right after reset, and it deactivates the line as soon as up_or_downn goes high, thus entering the normal regime of operation of the PD.

The behavior of q_o and q_i signals, which define the sensitivity window of the PD, is illustrated in Figure 11.15. ckin_i(ckout_o) and ckin_o(ckout_i) are delayed versions of ckin_up(ckout_down). Thanks to using the same fine delay cells as in the ADB, ckin_i(ckout_o) is delayed an additional ADB LSB with respect to ckin_o(ckout_i).
If ckout_down arrives earlier(later) than ckin_up by a time difference larger than ADB LSB, ckout_i will arrive clearly earlier(later) than ckin_i, thus a 0(1) will be captured in q_i; and ckout_o will arrive clearly earlier(later) than ckin_o, so a 1(0) will be captured in q_o. In these scenarios, the OR of q_o, q_i is 1, so a clk_PD_ready_aux pulse will be generated.
If ckout_down arrives earlier(later) than ckin_up by a time smaller than ADB LSB, ckout_i will arrive earlier than ckin_i, thus a 0 will be captured in q_i; and ckout_o will arrive later than ckin_o, so a 0 will be captured in q_o. In these scenarios, the OR of q_o, q_i is 0, so no clk_PD_ready_aux pulse will be generated (unless force_clk_PD_ready is high).

The signal that triggers the FF labelled with an asterisk can be the falling edge of either ckin_up_buff or ckout_down_buff (a delayed version of ckin_up and ckout_down respectively), whichever is to arrive last.
The trigger signal is selected by means of a multiplexer, whose selection signal is up_or_downn_aux. When ckin_up arrives last, the selection signal is 1 and the trigger signal is ckin_up_buff; when ckout_down arrives last, the selection signal is 0 and the trigger signal is ckout_down_buff.
Since the selection signal is updated with the rising edge of ckout_down, and q_i,

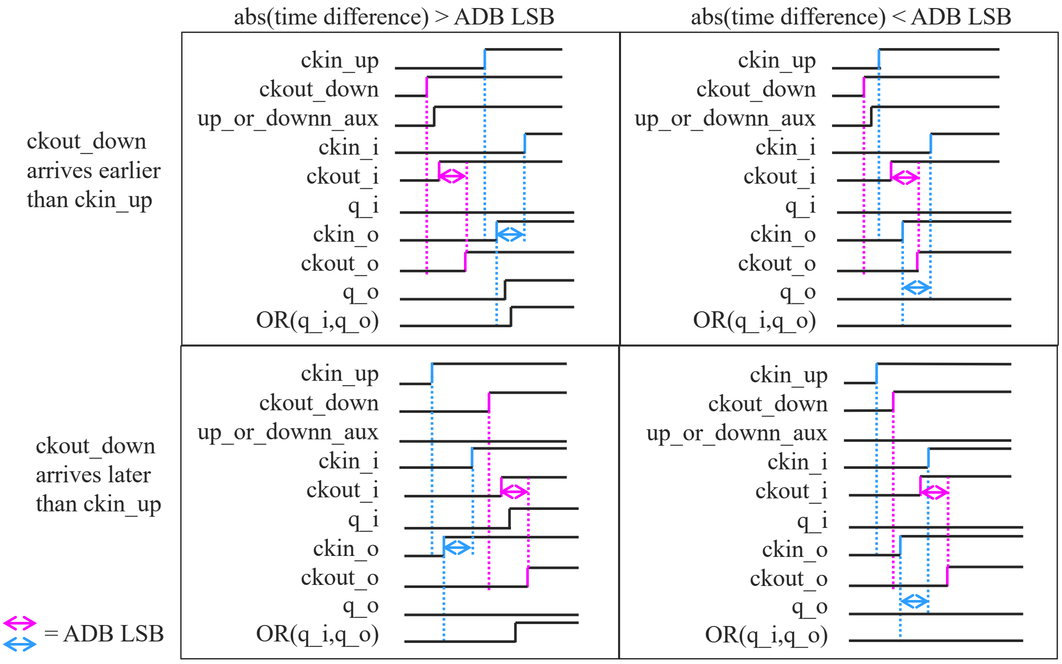

abs(time difference) > ADB LSB
abs(time difference) < ADB LSB
ckout_down arrives earlier than ckin_up
ckout_down arrives later than ckin_up
= ADB LSB
ckin_up
ckout_down
up_or_downn_aux
ckin_i
ckout_i
q_i
ckin_o
ckout_o
q_o
OR(q_i,q_o)

FIGURE 11.15: Signals involved in the generation of `clk_PD_ready_aux`.

`q_o` can be updated with the rising edge of `ckin_up` or `ckout_down`, the falling edge of the trigger signal is used instead, to allow for a time margin so that the rest of signals are stable when the FF is triggered.
The reason to use the clock that arrived last as a trigger is so that `q_o` and `q_i` are stable (and with them, the data input of the FF), since they change with such a clock.

11.3.3 Digital low-pass filter

Ideally, no `clk_PD_ready` pulse should be generated when the delay of the line is comprised between 1 master clock period - ADB LSB and 1 master clock period + ADB LSB. From this point, the range of time differences between the input and output clocks of the line for which no `clk_PD_ready` pulse is generated will be referred to as the time resolution or sensitivity window of the PD. Therefore, the ideal time resolution of the PD is +/- ADB LSB, but it can be deteriorated due to the following sources of time error:

1. Load effects in the connection to the DCDL: the routing of `ckout_down` and `ckin_up` must be symmetric and introduce the same parasitics as the interconnection between the intermediate stages of the DCDL. Otherwise, an artificial offset is added to the time difference of interest.

2. Setup-and-hold window of the flip-flops that sample the time difference between `ckin_up` and `ckout_down`.

3. The jitter superimposed to `ckin_up`, which is propagated and thus observed at `ckout_down` as well. Jitter distorts the time difference to be measured and

causes ringing in the PD outputs, which forces the unnecessary update of the controller and prevents the achievement of lock.

The impact of effect 1. can be reduced with a careful layout, while effects 2. and 3. can be mitigated by low-pass filtering the outputs of the PD. The role of such a filter is illustrated in Figure 11.16.

The top half of the figure represents the outputs of an ideal PD, with no setup-and-hold window effects and no jitter superimposed to ckin_up. The total delay of the DCDL is swept from values lower than the master clock period (ckout_down arrives earlier than ckin_up), for which up_or_downn is 1; towards values larger than one period (ckout_down arrives later than ckin_up), for which up_or_downn is 0. clk_PD_ready pulses are generated when ckout_down arrives earlier(later) than ckin_up by an amount larger than ADB LSB, which is labelled as S(E). S and E represent the start and end of the sensitivity window of the PD respectively, or the range of time differences for which no clk_PD_ready pulse is generated.

The bottom half of the figure represents a more realistic scenario, in which the aforementioned jitter is superimposed to ckin_up and the effect of the setup-and-hold window of the FFs in the PD is taken into account.
Due to the jitter superimposed to ckin_up, the time difference between the data and clock inputs of the concerned FFs changes dynamically over time, with standard deviation σ_j and the time difference of interest as the mean value.
Concerning the second effect, for time differences smaller that the setup-and-hold window, the output of the first of the two FFs in the 2-FF synchronisers might become metastable and eventually collapse to 1 or 0 (which of the two values cannot be foreseen).
Both effects are reflected in the un-filtered outputs up_or_downn_aux (ringing) and clk_PD_ready_aux (generation of a pulse for small time differences or absence of a pulse for large time differences). This erroneous behavior starts for a time difference labelled as S and it spans until the time difference labelled as E, which are determined by the setup and hold times of the concerned FFs and σ_j. In this range of time differences, the output of the PD is not reliable.

As a result, 1) lock cannot be achieved: the presence of clk_PD_ready_aux pulses continues to trigger the controller and forces an uninterrupted change in the delay of the line, toggling between incrementing and decrementing 1 ADB LSB; and 2) the time resolution of the PD, or range of time differences comprised between points S and E, is larger than +/- ADB LSB, so the required time sensitivity cannot be achieved.

up_or_downn_aux and clk_PD_ready_aux are low-pass filtered to reduce the ringing in the first; and to reduce the range with wrong pulse generation (or the range of time differences between points S and E) by a factor close to $\sqrt{16}$, where 16 is the number of clk_PD_ready_aux cycles that are processed before propagating a pulse to clk_PD_ready and updating the value of up_or_downn, i.e. it is the window of samples that are integrated by the low-pass filter. With this reduction, the time resolution

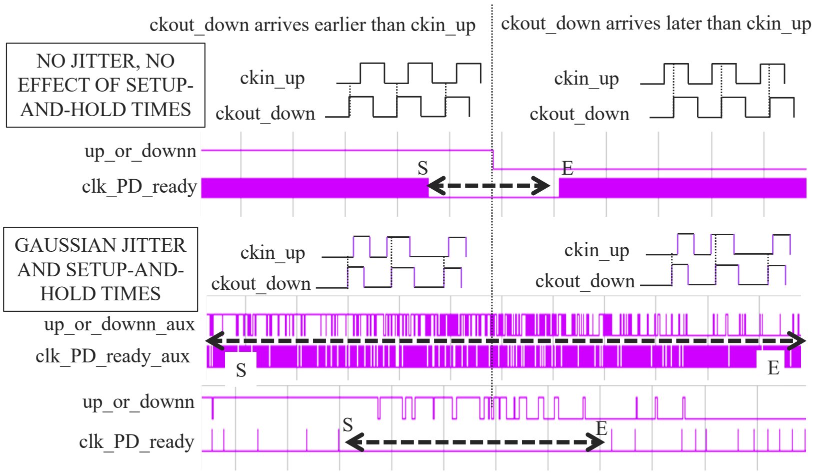

ckout_down arrives earlier than ckin_up
ckout_down arrives later than ckin_up
NO JITTER, NO EFFECT OF SETUP-AND-HOLD TIMES
ckin_up
ckout_down
ckin_up
ckout_down
up_or_downn
clk_PD_ready
S
E
GAUSSIAN JITTER AND SETUP-AND-HOLD TIMES
ckin_up
ckout_down
ckin_up
ckout_down
up_or_downn_aux
clk_PD_ready_aux
S
E
up_or_downn
clk_PD_ready
S
E

FIGURE 11.16: Role of the digital low-pass filter implemented in the phase detector.

of the PD becomes comparable to +/- ADB LSB, as it will be shown in Chapter 12.

The architecture of the digital low-pass filter is shown in Figure 11.17. The operation principle of the filter is the following: if the value of up_or_downn_aux remains stable for 16 consecutive clk_PD_ready_aux pulses, one pulse is generated at clk_PD_ready and the value of up_or_downn_aux is propagated to up_or_downn. If the value of up_or_downn_aux toggles before completing the filter window, the count is reset and neither up_or_downn nor clk_PD_ready are updated. As a result, at least 16 cycles of clk_PD_ready_aux are processed before propagating any value to the PD outputs, which provides the aforementioned benefits.

Concerning the implementation, a 4-bit binary counter (counter_clock_cycles) keeps track of how many consecutive clk_PD_ready_aux pulses occur with a stable value of up_or_downn_aux. At the beginning, the counter is reset to 0, up_or_downn is reset to 1 and clk_PD_ready is reset to 0 with the falling edge of the asynchronous reset pin, resetn.

The value of up_or_downn_aux is sampled with the reference_value_up_or_downn signal, which is the output of a multiplexer. If the selection signal of this multiplexer is 1, the up_or_downn_aux value is sampled, while if it is 0, the former value of reference_value_up_or_downn is maintained.
The selection signal is 1 when resetn is applied, so as to capture an initial value of up_or_downn_aux, and when the counter value is 0 (i.e. at the beginning of the 16-sample window of the filter); otherwise, the selection signal is 0.
This signal is updated with every clk_PD_ready_aux pulse and, in the absence of such a pulse (when the time difference between ckin_up and ckout_downn is small enough, no pulse should be generated), it is updated when up_or_downn_aux toggles. For instance, up_or_downn_aux should have a falling edge in the middle of the sensitivity window of the PD when the delay of the line changes from being smaller than the master clock period to larger than the clock period, and the final, low value

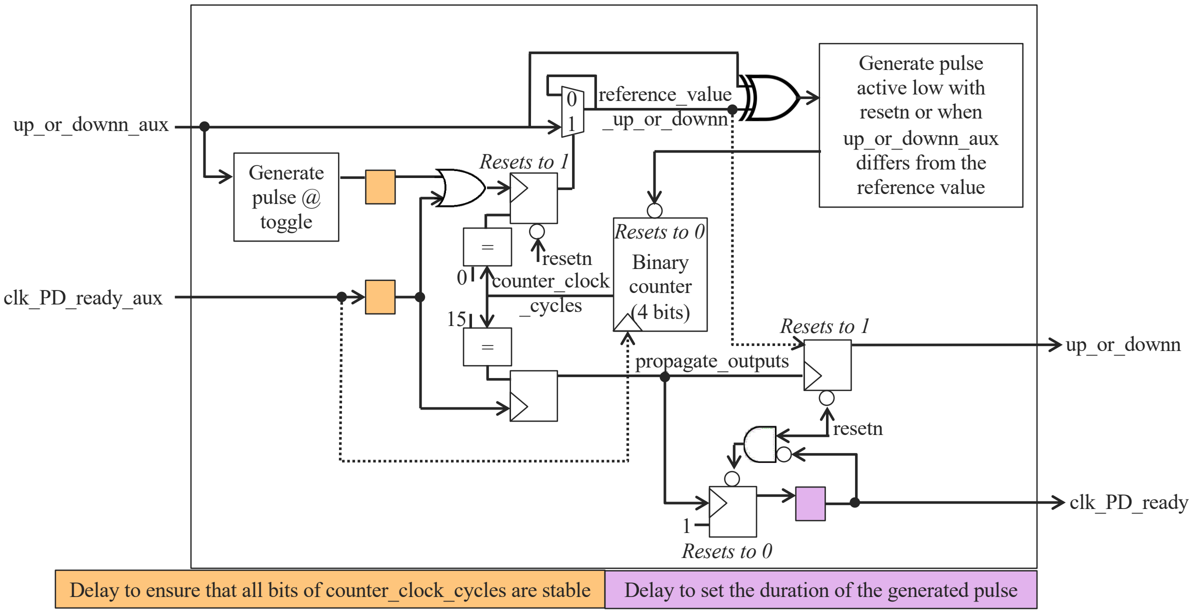


FIGURE 11.17: Architecture of the digital low-pass filter implemented in the phase detector.

must be captured to be used as the reference for the following 16-sample window.

The counter is incremented with every rising edge of `clk_PD_ready_aux`, and it is reset with `resetn`, by overflow or when `up_or_downn_aux` differs from `reference_value_up_or_downn` (the XOR of these signals goes high), thus leaving the current window and opening a new 16-sample window.

If 16 `clk_PD_ready_aux` pulses occur in a row with a stable `up_or_downn_aux` value (the counter reaches value 15), the outputs of the PD are updated (`propagate_outputs` goes high): `reference_value_up_or_downn` is loaded to `up_or_downn` and a pulse is generated at `clk_PD_ready` (using the same mechanism explained for the generation of `clk_PD_ready_aux`).

Note that the comparison between the counter value and 0 (to update the selection signal of the multiplexer) or 15 (to indicate the end of the 16-sample window) is performed with a delayed version of the trigger event (`clk_PD_ready_aux` or toggle in `up_or_downn_aux`). The purpose of delaying the trigger event is to ensure that all bits of the counter are stable when sampled.

11.4 Algorithm to distribute the fine control bits

A different latency or propagation delay from the clock source to the output of the ADBs causes skew or time offset between the different sinks. This time error has two components: skew by design (the arrival of the master clock is distributed over a

clock period along the delay line), which can be compensated offline; and the static time error on top of the skew by design. The second is due to differences in the layout of the ADB stages, delay mismatches due to PVT variations, and having a different value of fine control bits along the line when lock is achieved.
The controller can update the fine control bits of the ADB stages individually, which enables the fine adjustment of the total DCDL delay in steps of ADB LSB; but at the same time, it opens the door to having different values of fine control bits along the delay line when lock is achieved, which introduces static time error. The controller follows an algorithm to update the fine control bits of the stages in such an order that seeks to reduce the resulting static time error, which will be explained in this section.

11.4.1 Impact of the distribution of the fine control bits on the static time error of the DCDL

A useful figure to understand the impact of skew is the INL of the DCDL when lock is achieved, which is calculated as: $INL(k) = \sum_{i=U1..k} DNL(i)$, with $DNL(k) = [l(k) - l(k-1)] - [l_i(k) - l_i(k-1)]$, l the actual latency or propagation delay from the clock source until the output of each ADB, l_i the ideal latency, and k, i the indexes representing the ADBs from U1 onwards [49].
Note that in this work the INL will be expressed in time units (picoseconds), and not normalised to the LSB.
The ideal latency is obtained when all stages introduce the same delay or, in other words, it represents the skew by design. Therefore, the INL provides the distance between the ideal and actual latencies, that is, the static time error of the line, which is to be minimised.

With this purpose, the ADBs are carefully laid out to ensure the physical symmetry between the stages that propagate the clock upwards in the column of pixels (D0, D1...) and those that propagate it downwards (...D1, D0). Consequently, the main contribution to the INL is the divergence in the value of fine control bits along the line when lock is achieved.

With the aim of reducing this contribution to the INL, the controller follows an algorithm to update the fine control bits of the ADBs with such an order that minimises the resulting static time error. To understand the operation of the algorithm, an example DCDL of 8 stages (named U0 → U1 →... → D1 → D0), with the clock propagating from U0 towards D0, will be used. All stages are ideal (there is no skew due to PVT variations or layout mismatch) and share the same coarse control bits. In order to achieve lock, four of the stages have a 0 as fine control bits and the other four have a 1 as fine control bits.
In this context, the actual latency of the line is depicted in blue in Figure 11.18, and the ideal latency is superimposed in red. Six combinations of such values of fine control bits are depicted. The INL corresponding to these scenarios, which is due to the divergence in the fine control bit values along the line, is depicted in Figure 11.19.

Two important aspects are to be retained from this result:

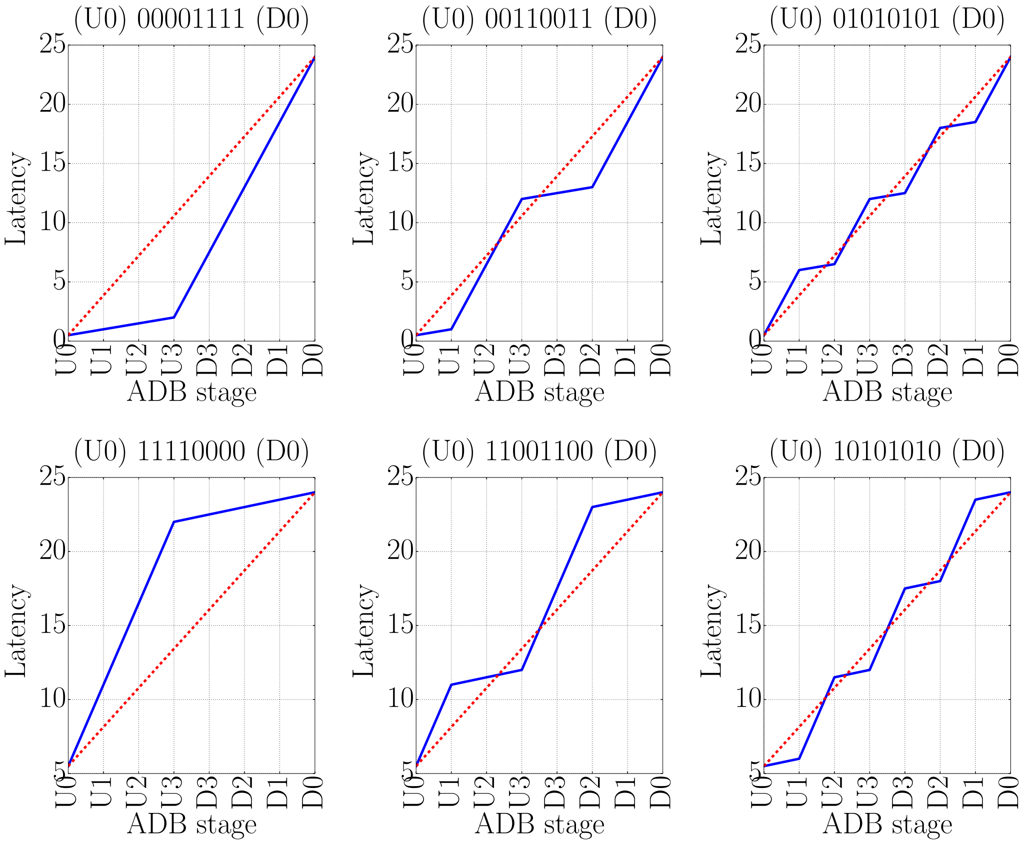


FIGURE 11.18: Latency and ideal latency of an example DCDL of 8 stages and different combinations of fine control bits along the line.

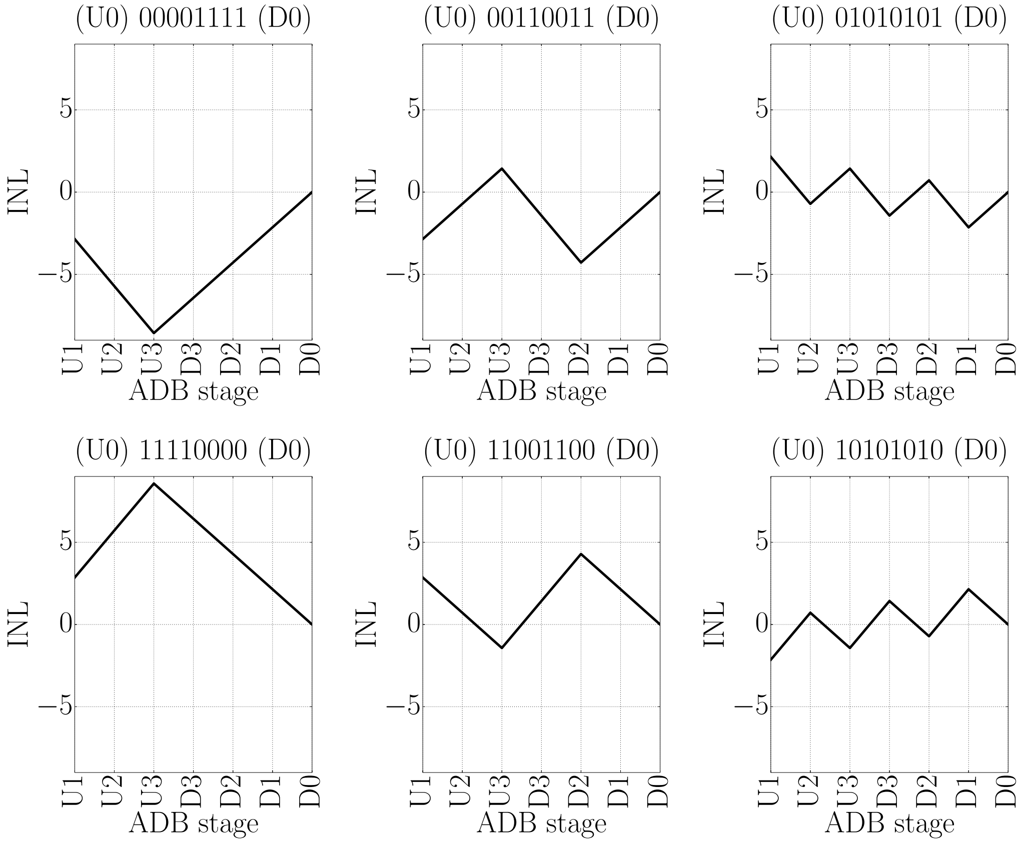


FIGURE 11.19: INL of an example DCDL of 8 stages and different combinations of fine control bits along the line.

- The excursion of the INL is smaller the more the fine control bit update is alternated along the line (the rightmost column of plots yields the best case).
- For a given distribution of fine control bits, mirroring the values of fine control bits between the DCDL halves (i.e. going from the top row of the plot to the bottom row, or vice versa) does not change the magnitude of the INL, only its sign.

With this result in mind and taking into account the dDLL operation, a list of requirements that the algorithm must fulfill is defined:

1. The ordering in the update of the fine control bits must be applicable to all fine control bit values, so that the full range of adjustment of the DCDL delay is covered.
2. It must be bidirectional, to support delay increments and decrements.
3. The fine control bits shall be updated at only one stage at a time and incremented/decremented in one unit only, to produce a change of 1 ADB LSB in the delay of the line.
4. To simplify the implementation, the update shall be incremental, that is, the former state of fine control bits is preserved along the line, and only the stage that is being updated changes. In terms of implementation, this condition implies that the Hamming distance between adjacent rows of the fine control bit state matrix (which will be defined in Section 11.4.2) must be 1.
5. The combination of points 1, 3 and 4 implies that a given time there can only be two different values of fine control bits along the line. In other words, the only scenarios allowed are either that all stages have the same fine control bits; or that some of the stages have x as fine control bits and the rest have $x + 1$ as fine control bits, with $x \in [0, max)$ (the maximum value that the fine control bits can take is 15 in decimal code).
6. To reduce the excursion of the INL, two consecutive updates shall be spaced along the line in opposite halves of the line and within the same half itself if possible.
7. It must be scalable to various lengths of the DCDL (8, 16, 20, 24, 28, 32 stages) to be compatible with the different chip areas.

An algorithm to define the order in the update of the fine control bits that fulfills the aforementioned requirements is explained next.

11.4.2 Proposed algorithm to update the fine control bits aiming at a low DCDL static time error

The strategy starts with defining a mini-matrix ('Mini-matrix fine control bit state' in Table 11.1) in which each column represents the state of the fine control bits of an ADB. The number of columns of this mini-matrix, i.e. the number of stages in this mini-DCDL, is the maximum common divisor of all the DCDL lengths to which this algorithm will be applied.
If a given stage has a 0, it retains the former value of fine control bits; if it as a 1, the stage loads the new value of fine control bits. The controller can replace 0s and 1s by x, $x + 1$ respectively, with $x \in [0, max)$ (the highest fine control bit value is 15 in

decimal code).
The rows of the mini-matrix represent the evolution of the fine control bits as they are updated. The row with all 0s, i.e. when all fine control bits retain the former value, is obviated.

The stage that is updated at a given time is represented by a green box. Following the green boxes from the top towards the bottom row provides the update sequence to increase the delay of the line, while following the same path in reverse order is used to decrease the delay of the line. The 'Seq. update fine control bits' column reflects the resulting update sequence.
There are 4 possible mini-matrices, A to D, that guarantee the maximum alternation in the ADB update along the line, thus limiting the excursion of the resulting INL.
The update sequence can be encoded with 2 bits (o_1, o_0), as represented in the column 'Ordering mini-matrix'.
To expand the ordering of the mini-matrix to arbitrary DCDL lengths, it is useful to express the 2-bit ordering code o_1, o_0 as a function of the 2-bit natural counter (b_1, b_0). The relation between these pairs of bits is indicated in the column 'Mapping natural – ordering'.

TABLE 11.1: Main steps of the ordering algorithm at the mini-matrix level for the 4 ordering options.

Ordering options	Mini-matrix fine control bit state				Seq. update fine control bits	Ordering mini-matrix		Natural counter		Mapping natural - ordering
	U0	U1	D1	D0		o_1	o_0	b_1	b_0	
A	1	0	0	0	U0	0	0	0	0	$o_1 = b_0$
	1	0	1	0	D1	1	0	0	1	$o_0 = b_1$
	1	1	1	0	U1	0	1	1	0	
	1	1	1	1	D0	1	1	1	1	
B	0	1	0	0	U1	0	1	0	0	$o_1 = b_0$
	0	1	0	1	D0	1	1	0	1	$o_0 = 1\text{-}b_1$
	1	1	0	1	U0	0	0	1	0	
	1	1	1	1	D1	1	0	1	1	
C	0	0	1	0	D1	1	0	0	0	$o_1 = 1\text{-}b_0$
	1	0	1	0	U0	0	0	0	1	$o_0 = b_1$
	1	0	1	1	D0	1	1	1	0	
	1	1	1	1	U1	0	1	1	1	
D	0	0	0	1	D0	1	1	0	0	$o_1 = 1\text{-}b_0$
	0	1	0	1	U1	0	1	0	1	$o_0 = 1\text{-}b_1$
	0	1	1	1	D1	1	0	1	0	
	1	1	1	1	U0	0	0	1	1	

3-bit, 4-bit and 5-bit ordering codes are required to index the update of the fine control bits for the various DCDL lengths. These codes can be derived from the 5-bit natural counter by 1) applying the relation existing between the natural counter and ordering code; and 2) in the case of 3-bit and 4-bit ordering codes, removing the indexes (or DCDL stages) that are present in the 5-bit code, but which are not present in the smaller DCDL lengths.

The 5-bit natural counter is reported in Table 11.2 for convenience.

In the case of the mini-matrix, the Most Significant Bit (MSB) of the ordering code was derived from the LSB of the natural counter, and position MSB-1 of the ordering code was derived from position LSB+1 of the natural counter.

Analogously, for the 5-bit indexing this relation is expanded as follows: positions MSB (o_4), MSB-1 (o_3), MSB-2 (o_2), MSB-3 (o_1), MSB-4 (o_0) of the ordering code are derived from positions LSB (b_0), LSB+1 (b_1), LSB+2 (b_2), LSB+3 (b_3), LSB+4 (b_4) of the natural counter, respectively.

The relation between o_4-b_0 and o_3-b_1 can be obtained from Table 11.1. For the rest of bits, it can be observed that in the natural counter $b_2(b_3)$ can be derived from $b_0(b_1)$ by reducing the toggling frequency of $b_0(b_1)$ by 1/4. The same reasoning can be applied to $o_2(o_1)$, which can be derived from $o_4(o_3)$ by reducing the toggling frequency of $o_4(o_3)$ by 1/4. Concerning o_0, it will be 0 for half the code range and 1 for the other half, in the line of b_4. Which value corresponds to each half is not critical, since it will only change the sign of the INL, but not the absolute value of the peak.

Such relations between the 5-bit natural counter and the 5-bit ordering code are shown in Table 11.3 for the different ordering options, in the column 'Mapping natural counter - ordering'. The resulting 5-bit ordering code for ordering option B is shown in Table 11.2 under the column 'Ordering code, option B'.

Table 11.4 compiles the mapping between the ordering codes and the corresponding natural counter for a generic DCDL length, for the four ordering options. The code and the counter have a length of N bits, which can take the values 3 (DCDL of 8 stages), 4 (DCDL of 16 stages) and 5 (DCDL of 20, 24, 28, 32 stages). The mapping is particularised for even (`N-1-2i`) and odd (`N-1-2i-1`) bit positions of the ordering code. `[(N-1)/2]` and `[(N-2)/2]` stand for the integer part of the upper bound that the bit index can take.

The ordering codes can be translated to the matrix of fine control bit state (as reported in Appendix A) and to the sequence in which the stages are updated, as shown in the column 'Sequence to update fine control bits' in Table 11.3 (for ordering option B, this sequence is also shown in the 'Seq. update' column in Table 11.2). This is the sequence followed by the controller to change the fine control bit values along the line.

The four ordering options have been implemented in the respective dDLL flavors and simulated, yielding similar results in terms of INL (the INL obtained for the different options is reported in Appendix B). In the following, ordering option B will be used, since it yields a slightly better performance.

11.5 Controller

The controller consists of a synchronous, Mealy FSM that is in charge of regulating the delay of the DCDL. It supports two operation modes: the normal operation

TABLE 11.2: Expansion of the ordering code to 5 bits by means of the natural counter and resulting sequence of update (option B).

Natural counter					Ordering code, option B					Seq. update
b_4	b_3	b_2	b_1	b_0	o_4	o_3	o_2	o_1	o_0	
0	0	0	0	0	0	1	0	1	0	U10
0	0	0	0	1	1	1	0	1	0	D5
0	0	0	1	0	0	0	0	1	0	U2
0	0	0	1	1	1	0	0	1	0	D13
0	0	1	0	0	0	1	1	1	0	U14
0	0	1	0	1	1	1	1	1	0	D1
0	0	1	1	0	0	0	1	1	0	U6
0	0	1	1	1	1	0	1	1	0	D9
0	1	0	0	0	0	1	0	0	0	U8
0	1	0	0	1	1	1	0	0	0	D7
0	1	0	1	0	0	0	0	0	0	U0
0	1	0	1	1	1	0	0	0	0	D15
0	1	1	0	0	0	1	1	0	0	U12
0	1	1	0	1	1	1	1	0	0	D3
0	1	1	1	0	0	0	1	0	0	U4
0	1	1	1	1	1	0	1	0	0	D11
1	0	0	0	0	0	1	0	1	1	U11
1	0	0	0	1	1	1	0	1	1	D4
1	0	0	1	0	0	0	0	1	1	U3
1	0	0	1	1	1	0	0	1	1	D12
1	0	1	0	0	0	1	1	1	1	U15
1	0	1	0	1	1	1	1	1	1	D0
1	0	1	1	0	0	0	1	1	1	U7
1	0	1	1	1	1	0	1	1	1	D8
1	1	0	0	0	0	1	0	0	1	U9
1	1	0	0	1	1	1	0	0	1	D6
1	1	0	1	0	0	0	0	0	1	U1
1	1	0	1	1	1	0	0	0	1	D14
1	1	1	0	0	0	1	1	0	1	U13
1	1	1	0	1	1	1	1	0	1	D2
1	1	1	1	0	0	0	1	0	1	U5
1	1	1	1	1	1	0	1	0	1	D10

TABLE 11.3: Expansion of the mini-matrix to the largest ordering sequence, for the 4 ordering options.

Ordering options	Mapping natural counter - ordering	Sequence to update fine control bits
A	$o_4 = b_0$ $o_3 = b_1$ $o_2 = b_2$ $o_1 = b_3$ $o_0 = b_4$	U0 → D15 → U8 → D7 → U4 → D11 → U12 → D3 → U2 → D13 → U10 → D5 → U6 → D9 → U14 → D1 → U1 → D14 → U9 → D6 → U5 → D10 → U13 → D2 → U3 → D12 → U11 → D4 → U7 → D8 → U15 → D0
B	$o_4 = b_0$ $o_3 = 1\text{-}b_1$ $o_2 = b_2$ $o_1 = 1\text{-}b_3$ $o_0 = b_4$	U10 → D5 → U2 → D13 → U14 → D1 → U6 → D9 → U8 → D7 → U0 → D15 → U12 → D3 → U4 → D11 → U11 → D4 → U3 → D12 → U15 → D0 → U7 → D8 → U9 → D6 → U1 → D14 → U13 → D2 → U5 → D10
C	$o_4 = 1\text{-}b_0$ $o_3 = b_1$ $o_2 = 1\text{-}b_2$ $o_1 = b_3$ $o_0 = b_4$	U5 → D10 → U13 → D2 → U1 → D14 → U9 → D6 → U7 → D8 → U15 → D0 → U3 → D12 → U11 → D4 → U4 → D11 → U12 → D3 → U0 → D15 → U8 → D7 → U6 → D9 → U14 → D1 → U2 → D13 → U10 → D5
D	$o_4 = 1\text{-}b_0$ $o_3 = 1\text{-}b_1$ $o_2 = 1\text{-}b_2$ $o_1 = 1\text{-}b_3$ $o_0 = b_4$	U15 → D0 → U7 → D8 → U11 → D4 → U3 → D12 → U13 → D2 → U5 → D10 → U9 → D6 → U1 → D14 → U14 → D1 → U6 → D9 → U10 → D5 → U2 → D13 → U12 → D3 → U4 → D11 → U8 → D7 → U0 → D15

TABLE 11.4: Expansion of the mini-matrix to a DCDL of arbitrary length, for the 4 ordering options.

Ordering options	Mapping natural counter - ordering
A	$o_{N-1-2i} = b_{2i}$, $N \in [3,5]$, $i \in [0,[(N-1)/2]]$ $o_{N-1-2i-1} = b_{2i+1}$, $N \in [3,5]$, $i \in [0,[(N-2)/2]]$
B	$o_{N-1-2i} = b_{2i}$, $N \in [3,5]$, $i \in [0,[(N-1)/2]]$ $o_{N-1-2i-1} = 1 - b_{2i+1}$, $N \in [3,5]$, $i \in [0,[(N-2)/2]]$
C	$o_{N-1-2i} = 1 - b_{2i}$, $N \in [3,5]$, $i \in [0,[(N-1)/2])$ $o_0 = b_{N-1}$ $o_{N-1-2i-1} = b_{2i+1}$, $N \in [3,5]$, $i \in [0,[(N-2)/2]]$
D	$o_{N-1-2i} = 1 - b_{2i}$, $N \in [3,5]$, $i \in [0,[(N-1)/2])$ $o_0 = b_{N-1}$ $o_{N-1-2i-1} = 1 - b_{2i+1}$, $N \in [3,5]$, $i \in [0,[(N-2)/2]]$

mode, in which the delay of the DCDL is swept until lock is achieved; and a debug mode, in which the whole range of adjustment of the DCDL delay is explored. The second mode is mainly used for verification: to characterise the DCDL operation, to evaluate the time resolution of the PD and to verify the correct state transition and output generation of the controller.

The pinout of the controller is depicted in Figure 11.20. up_or_downn and clk_PD_ready stand for the PD outputs: they indicate the direction in which the delay of the line should change and force such an update, respectively.
An asynchronous reset (active low) can be applied via the resetn pin, which sets the controller in the normal operation mode. At that point, the DCDL delay is set to its lowest value.
As it was introduced in Section 11.3.2, it might occur that the initial delay of the DCDL is lower than half the master clock period, which is outside the range of detection of the PD. As a consequence, up_or_downn goes low, while it should be high to force incrementing the delay. To ensure that lock is achieved, the controller automatically increments the delay of the line until up_or_downn goes high, and from then onwards it acts according to the value of up_or_downn. To perform this initial, automatic increment of the line delay, force_clk_PD_ready is set high right after reset, and it is deactivated as soon as up_or_downn goes high.

When the controller is operating in the normal mode, the user can provide a pulse (active high) via the posedge_starts_sweep_delay_DCDL pin at any time to quit the operation of finding lock and set the controller in the debug operation mode. When this occurs, the controller sets the delay of the line to its lowest value and starts to increase it in steps of ADB LSB until the highest delay is achieved.
During the whole delay sweep, the output force_clk_PD_ready is high to force the generation of clk_PD_ready pulses at the PD and thus ensure the execution of the delay sweep. When the highest DCDL delay is reached, force_clk_PD_ready is deactivated and the controller returns to the normal operation mode.

The remaining pins are the control outputs sent to the DCDL to regulate its delay. To change the coarse section of all ADBs simultaneously, the controller first updates the value of the coarse control bits (lines ctrl_bits_coarse_gray_up are a copy of ctrl_bits_coarse_gray_down), and after a certain time (to ensure that the control bits have propagated until the top of the pixel columns and thus they are stable in all ADBs), it sends a pulse of the auxiliary clock to force the load of the new control bits (clk_update_coarse_ctrl_bits_up is a copy of clk_update_coarse_ctrl_bits_down).

To change the fine section of an ADB located in the DCDL half that propagates the clock upwards(downwards) in the column of pixels (stages U0(D0) to U15(D15)), the controller loads the new value of fine control bits to the ctrl_bits_fine_gray_up(down) lines and the address of the target stage in the address_gray_select_up(down) lines. After a certain time (to ensure that these values have propagated until the top of the pixel columns and thus they are stable in all ADBs), it sends a pulse of the auxiliary clock to force the load of the new control bit values via clk_update_fine_ctrl_bits_up(down).

When an asynchronous reset is applied via the `resetn` pin, the controller clears all control bit lines to their lowest value (0 in decimal code) and sends a pulse via all auxiliary clock lines. This resets the coarse control bits of all stages to their lowest value. In order to reset the fine control bits of all stages simultaneously, before sending the auxiliary `clk_update_fine_ctrl_bits_up/down` pulses, the controller sets the `clearn_to_min_up/down` lines low, which forces the synchronous clear of the fine control bits in all stages.

When the controller sweeps the delay of the line in steps of ADB LSB, it changes the fine control bits of one ADB stage at a time, following the order described in Section 11.4.2. The controller keeps track of which stage is updated by means of the internal signal `pointer_stage`.

For a certain value of coarse control bits, if all fine sections have already been updated to the highest value of the fine control bits and the delay needs to be increased further, the controller will increment the coarse control bits in one unit. But in order to increase the total delay by only ADB LSB, the fine control bits of all stages must be reset to their lowest value in parallel to such an update of the coarse sections. To do so, the controller clears to 0 the `clearn_to_min_up/down` lines. After a certain time (to ensure that this 0 has propagated until the top of the pixel columns and thus it is stable in all ADBs), it sends a pulse of the auxiliary clocks `clk_update_fine_ctrl_bits_up/down`, and it eventually sets the `clearn_to_min_up/down` lines back high.

Anagolously, if all fine sections have already been updated to the lowest value of the fine control bits and the delay needs to be decreased further, the controller will decrement the coarse control bits in one unit. But in order to decrease the total delay by only ADB LSB, the fine control bits of all stages must be set to their highest value in parallel to such an update of the coarse sections. To do so, the controller clears to 0 the `setn_to_max_up/down` lines. After a certain time (to ensure that this 0 has propagated until the top of the pixel columns and thus it is stable in all ADBs), it sends a pulse of the auxiliary clocks `clk_update_fine_ctrl_bits_up/down`, and it eventually sets the `setn_to_max_up/down` lines back high.

11.5.1 FSM of the debug operation mode of the controller

The state diagram corresponding to the debug operation mode of the controller is shown in Figure 11.21. When a pulse is applied via the `posedge_starts_sweep_delay_DCDL` pin, the controller enters the `RESET` state (the coarse and fine control bits are cleared to the lowest value in all stages to force the lowest DCDL delay). At this point, the internal signal `internal_pulse_sweep_delay_DCDL` goes high, a state that it will retain until the delay sweep has completed, so as to force the state transition indicated in this diagram.

To begin with the delay sweep in steps of ADB LSB, the machine enters the `INCREMENT_FINE_BITS` state. The internal signal that keeps track of which stage is being updated, `pointer_stage`, is set to U10, which will be the first stage to receive a modification of the fine control bits.
Since U10 will be updated, the `ctrl_bits_fine_gray_up` lines are incremented in

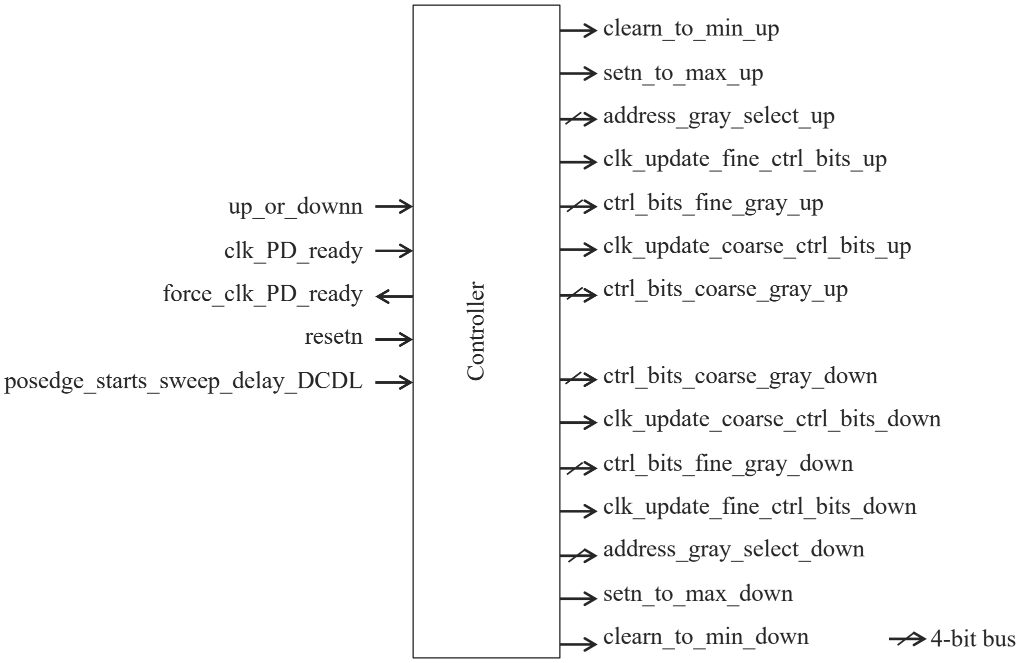


FIGURE 11.20: Block diagram of the controller pins.

one unit (in Gray code), the address lines `address_gray_select_up` point to this stage and a pulse is sent via `clk_update_fine_ctrl_bits_up`.
From here, the machine switches to the `INCREMENT_FINE_POINTER` state, in which `pointer_stage` is updated to D5. The address lines `address_gray_select_down` point to this stage and a pulse is sent via `clk_update_fine_ctrl_bits_down`, so that the same value of fine control bits that was loaded to U10 is now loaded to D5.
The same operation is repeated for the rest of stages, until stage D10 is updated.
At this point, all fine sections of the line have loaded the new value of fine control bits. Therefore, the controller returns to `INCREMENT_FINE_BITS` to increment the fine control bits in one unit, and proceeds to `INCREMENT_FINE_POINTER` in order to load the new value of fine control bits to all stages.

When all fine sections have been updated to the highest value of the fine control bits, the controller switches to the `INCREMENT_COARSE_BITS` state.
The `ctrl_bits_coarse_gray_up,down` lines are incremented in one unit (in Gray code) and a pulse is sent via the `clk_update_coarse_ctrl_bits_up,down` lines. In parallel, the fine control bits of all stages are reset to their lowest value (so that the overall increment in the delay of the line is ADB LSB) by clearing to 0 the `clearn_to_min_up,down` lines and sending a pulse via the `clk_update_fine_ctrl_bits_up,down` lines.
Next the controller returns to `INCREMENT_FINE_BITS` and the procedure described above to deliver all the fine control bit values to all stages is repeated.
When the iteration of the fine control bits along the line is completed, the controller proceeds to `INCREMENT_COARSE_BITS` to increment the coarse control bits by one unit more. This is repeated until the controller faces the situation when the fine control bits and the coarse control bits have their maximum value, and it just updated the

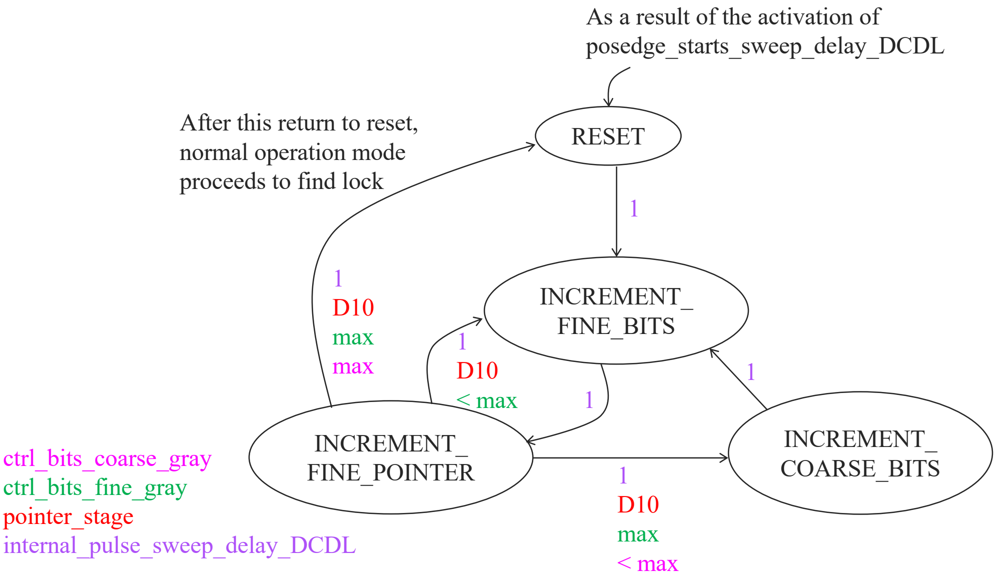


FIGURE 11.21: Synchronous FSM implemented at the controller (debug operation mode)

last stage (D10) in INCREMENT_FINE_POINTER.

At this point, the delay sweep has finalised, so the controller returns to RESET. The internal signal internal_pulse_sweep_delay_DCDL is cleared to indicate that the sweep is complete, and the delay of the line is restored to its lowest value. From here, the controller proceeds to find lock in the normal operation mode.

11.5.2 FSM of the normal operation mode of the controller

The normal operation mode starts with the RESET state and posedge_starts_sweep_delay_DCDL low. The controller arrives to RESET with a falling edge of the internal_resetn signal, a pulse (active low) generated with a falling edge of the asynchronous reset pin, resetn, or every time the controller returns to RESET coming from any other state. As it was explained above, when the controller is in RESET, the coarse and fine control bits are cleared to the lowest value in all stages to force the lowest DCDL delay, and pointer_stage is set to D10. Note that in the normal operation mode pointer_stage resets to a different value as in the debug operation mode. The reason to start the fine section update with D10 is that in a later state the controller will decrease the delay of the line in steps of ADB LSB, and the fine section update for the decrement operation is performed from D10 towards U10.

From RESET, the controller has two options: if up_or_downn is low, it will switch to PREVENT_LOCKUP, while if it is high it will proceed with FAST_INCR_DELAY. The first situation occurs in the fast corner, when the initial delay of the line is smaller than half the master clock period and the PD provides an anomalous value in up_or_downn. To ensure that lock can be achieved, the controller forces the generation of clk_PD_ready pulses by setting high the output force_clk_PD_ready for as long as it remains in PREVENT_LOCKUP, a state which is used to increment the coarse control bits. At some point, the delay of the line should become higher than half the master clock period, which should switch up_or_downn to a high value. When that occurs, the controller proceeds to FAST_INCR_DELAY. If this expected behavior were not to happen, at some point the controller would reach the highest allowed value for the coarse control bits and safely return to RESET to restart its operation.

While in FAST_INCR_DELAY, the controller speeds up the increase in the DCDL delay by incrementing the coarse control bits. Two situations can occur: 1) at some point, the delay of the line exceeds the master clock period, in which case up_or_downn goes low and the controller proceeds with DECR_COARSE_BITS; and 2) in the fast corner, it has been observed that even if the coarse control bits reach the maximum value, since the fine control bits along the line are all set to their lowest value, the line cannot reach a delay close enough to the master clock period to achieve lock. When this occurs, the controller proceeds to MIN_CORNER_SET_FINE_CTRL_BITS_TO_MAX, in which the fine control bits of all stages are set to their maximum value by means of setn_to_max_up,down and clk_update_fine_ctrl_bits_up,down. At this point, the delay of the line reaches its maximum value, which should be well above the master clock period, so up_or_downn is expected to switch to a low value. When that occurs, the controller proceeds with DECR_FINE_BITS to start decreasing the delay of the line in steps of ADB LSB. If an anomalous behavior were to occur (up_or_downn remains high, but the delay of the line cannot be further increased), the controller would safely return to RESET to restart its operation.

In the state DECR(INCR)_FINE_BITS, the fine control bits of the stage indicated by pointer_stage are decremented(incremented) in one unit, while in the state DECR(INCR)_COARSE_BITS the coarse control bits of all stages are decremented(incremented) in one unit. The stage index, pointer_stage, and the resulting address lines are updated from U10(D10) towards D10(U10) by remaining in the INCR(DECR)_FINE_POINTER state.

After increasing the delay of the line by incrementing the coarse control bits until it exceeds the master clock period, the controller will proceed to decrease the delay in steps of ADB LSB. This operation starts with DECR_COARSE_BITS (if the controller comes from FAST_INCR_DELAY and up_or_downn switches to a low value) or DECR_FINE_BITS (if the controller comes from MIN_CORNER_SET_FINE_CTRL_BITS_TO_MAX and up_or_downn switches to a low value).

In the second case, the fine control bits of all stages are set to their maximum value and the fine section of D10 will be updated. To start with the decrease, lines ctrl_bits_fine_gray_down,up are decremented in one unit, pointer_stage and address_gray_select_down point to D10, and a pulse is sent via clk_update_fine_ctrl_bits_down.

To further proceed with decreasing the delay (`up_or_downn` remains low), the controller switches to `DECR_FINE_POINTER`, in which `pointer_stage` changes to U5, which is reflected in the `address_gray_select_up` and `clk_update_fine_ctrl_bits_up` lines. The next stages in the sequence are updated following the same method, until U10 has loaded the current value of fine control bits.

The controller returns then to `DECR_FINE_BITS`. The `ctrl_bits_fine_gray_down,up` lines are decremented in one unit and the update sequence is restarted.
The controller proceeds in this fashion until the fine control bits of all stages have the minimum value and `pointer_stage` is U10, so it cannot sweep the pointer any further.
At this point, if `up_or_downn` remains low, the controller switches to `DECR_COARSE_BITS`. In this state, the `ctrl_bits_coarse_gray_up,down` lines are decremented in one unit and a pulse is sent via `clk_update_coarse_ctrl_bits_up,down` to load the new coarse control bits to all stages. In parallel, the `setn_to_max_up,down` and `clk_update_fine_ctrl_bits_up,down` lines are used to set the fine control bits of all stages to their maximum value, so that the reduction in the coarse control bits causes a decrease of only ADB LSB in the delay of the line.
To further decrease the delay, the controller loops between `DECR_FINE_BITS`, `DECR_FINE_POINTER` and `DECR_COARSE_BITS` until lock is achieved, or eventually until the lowest possible delay of the line is reached. If the second occurs and `up_or_downn` remains low (and `clk_PD_ready` pulses continue to arrive), the controller returns to `RESET` to restart its operation.

If the controller is in the `DECR_FINE_BITS` or `DECR_FINE_POINTER` states and the PD requests a change in the direction of update of the delay (`up_or_downn` goes high), the controller reverses the direction in the update sequence. When it switches to `INCR_FINE_POINTER`, `pointer_stage` is set to its previous value, and from this point the fine sections will be updated in direction U10 towards D10. When `pointer_stage` reaches D10, if the delay shall still be increased, the controller proceeds with `INCR_FINE_BITS`. The `ctrl_bits_fine_gray_up` lines are incremented in one unit, `pointer_stage` and `address_gray_select_up` point to U10 and a pulse is sent via `clk_update_fine_ctrl_bits_up` to load the new value of fine control bits to U10. The controller returns to `INCR_FINE_POINTER` to advance with loading this value of fine control bits to the following stages.
When the maximum value of fine control bits has been loaded to all stages and the PD still requests to increase the delay of the line, the controller switches to `INCR_COARSE_BITS`. In this state, `ctrl_bits_coarse_gray_up,down` lines are incremented in one unit and the required `clk_update_coarse_ctrl_bits_up,down` pulses are sent along. In parallel, the fine control bits of all stages are reset to their lowest value (so that the resulting increment is ADB LSB) by means of `clearn_to_min_up,down` and `clk_update_fine_ctrl_bits_up,down`.
To further increase the delay of the line, the controller loops between `INCR_FINE_BITS`, `INCR_FINE_POINTER` and `INCR_COARSE_BITS` until lock is achieved, or eventually until the highest possible delay of the line is reached. If the second occurs and `up_or_downn` remains high (and `clk_PD_ready` pulses continue to arrive), the controller returns to `RESET` to restart its operation.

If the controller is in the `DECR(INCR)_COARSE_BITS` state and `up_or_downn` goes high(low), it will switch to `INCR(DECR)_COARSE_BITS`, and the

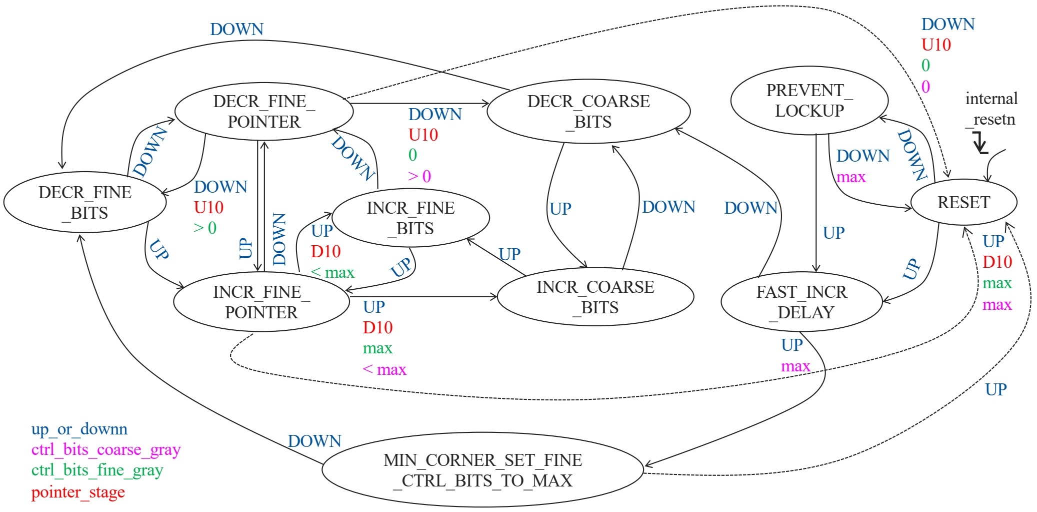


FIGURE 11.22: Synchronous FSM implemented at the controller (normal operation mode).

`coarse_ctrl_bits_coarse_gray_up,down` lines will be decremented(incremented) by one unit. As it has been explained before, such an update is accompanied by setting the fine control bits of all stages to their highest(lowest) value, so that the resulting change in the delay of the line is ADB LSB.

11.6 Scalability to arbitrary dimensions

The presented CDN architecture can fulfill the FastICpix requirement of being scalable with the chip area and pixel pitch. The proposed method to provide this adaptability is presented in this section, and numeric examples are provided in Table 11.5.

The following guidelines are used to scale the CDN with the chip area:

- The number of stages of the DCDL and number of dDLLs increases with the chip area. To limit the number of DCDL flavors to be implemented, two situations are proposed: for small chip areas (up to 1.2 × 1.2 cm^2), the master clock source is located on one side of the chip and the DCDLs span across the full chip height, while for greater chip areas the clock source is at the centre of the chip and the DCDLs span across half the chip height.
- The master clock frequency increases for shorter DCDL lengths, so that the total delay can be adjusted to 1 master clock period.
- The same ADB design can be used in all cases, except for the smallest chip area. In this case, to limit the range of master clock frequencies used to serve the different chip areas, the ADB delay is limited to half the delay in the rest of scenarios by reducing the coarse section contribution.
- The same PD design can be used in all cases.

- The same controller design can be used in all cases, only the indexing of the DCDL stages shall be adapted to the particular DCDL length.

Adaptation to the pixel pitch is provided at the level of the local clock tree that starts at the output of each ADB and drives the TDCs in the corresponding group of pixels. For 376 µm pixel pitch, the local clock tree drives 4 TDCs. For a smaller pixel pitch, since the chip area is preserved, the number of sinks to be served by the local clock tree will increase by a certain factor (376 µm/new pixel pitch).

TABLE 11.5: Guidelines to scale the dDLL design with the chip area.

Chip area (cm^2)	Number of pixels (pixel pitch = 376 µm)	Number of DCDL stages	Master clock frequency (MHz)	Number of dDLLs in the CDN
0.3×0.3	8×8	8[a]	80[b]	2
0.6×0.6	16×16	16[a]	75	4
0.9×0.9	24×24	24[a]	50	6
1.2×1.2	32×32	32[a]	40	8
1.5×1.5	40×40	20[c]	60	20
1.8×1.8	48×48	24[c]	50	24
2.1×2.1	56×56	28[c]	45	28
2.4×2.4	64×64	32[c]	40	32

[a] Master clock from one side of the chip (dDLL spans across full chip height)

[b] The ADB introduces half the delay as in the rest of chip areas, so that the maximum spread in the range of master clock frequencies is bound to a factor 2 between the largest and smallest frequencies

[c] Master clock from the center of the chip (dDLL spans across half the chip height)

11.7 Layout dimensions of the dDLL components

Figure 11.23 shows a simplified floorplan of the dDLL. The individual blocks (controller, phase detector, cell containing an upwards and a downwards ADB) are first laid out separately; the resulting macros are abutted to form the dDLL (the longest delay line is shown here); finally, the global netlist is flattened so that the analysis and simulations performed afterwards account for the actual interconnect and cell parasitics.

In Figure 11.23, the pixel pitch is 376 µm and a cell containing an upwards and a downwards ADB (as shown in Figure 11.11) is shared by 4 pixels. Each ADB contains the core delay cells, which were shown in Figure 11.5; and the area surrounding these cells, which enables fitting the ADB to the pixel pitch. This surrounding area is used for clock buffers in the upwards/downwards clock path to correct the clock slew, buffers to enhance the slew of the control bit lines, and decoupling capacitance.

With the numbers shown in Figure 11.23, and taking into account that there will be 32 dDLLs in the CDN for the largest chip area (as it was introduced with Table 11.5), such a CDN stands for 2% of the total chip area. Each ADB stands for 4% of the pixel area.

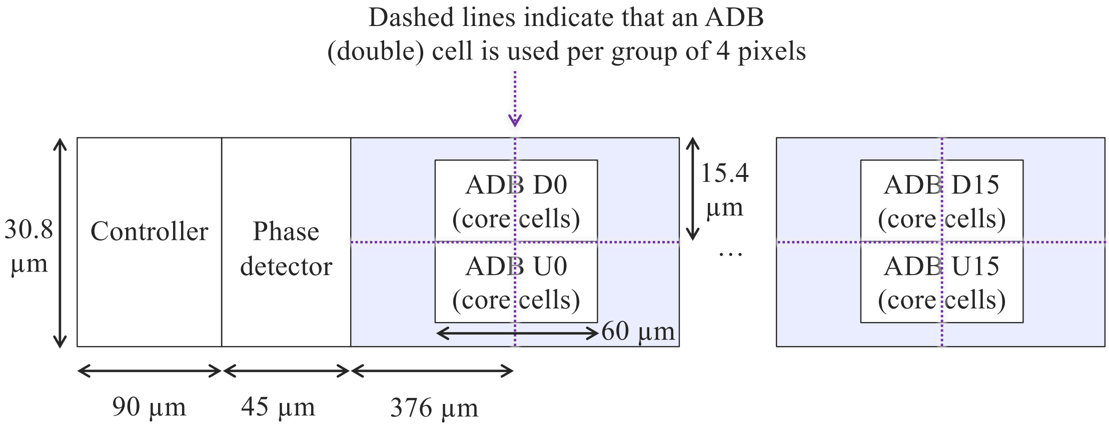


FIGURE 11.23: Simplified floorplan of the dDLL (not to scale).

The PD layout is the same for all chip area and pixel pitch variations. The controller layout does not change with the pixel pitch, but it shall be adapted to the chip area (shorter DCDLs will be addressed for smaller chip areas). The ADB layout is constant with the chip area, except for the smallest case (fewer coarse sections are to be included so as to reduce the ADB delay to half the range obtained for other chip areas). The ADB adapts to a smaller pixel pitch by increasing the complexity of the local clock tree that starts at the output of each ADB (the area of such a clock tree is not taken into account in the aforementioned percentages).

12 dDLL performance

In this section, the performance of the demonstrator dDLL (DCDL of 32 stages, 40 MHz master clock) is evaluated in terms of time resolution and power consumption. In addition, it is also shown that the dDLL can recover from a sudden perturbation in the master clock and eventually find lock.

12.1 Time performance of the dDLL

The time resolution of the dDLL is determined by the time errors of the DCDL and the PD. On the one hand, the static and dynamic time errors of the DCDL must be bound to the TDC time bin (20 ps) at each of its nodes (the output of every ADB), so that the phase error in the clock delivered to the TDCs causes a deviation of 1 TDC count at most with respect to the ideal scenario.
On the other hand, the time resolution of the PD will determine whether lock can be achieved and which is the static time error at the last stage of the DCDL (D0) when lock is achieved, which in turn impacts the DCDL time performance.

12.1.1 Time performance of the DCDL

The static time errors of the DCDL can be quantified in terms of the INL of the line, as it was introduced in Section 11.4.1. To evaluate the impact of dynamic time errors, random Gaussian jitter is superimposed to the master clock, with standard deviation σ_j and 0 mean, as it was introduced in Chapter 8.
Taking into account both contributions, the following time error target can be defined: $|INL(k)|_{max} + 3\sigma_j < 20$ps, where $|INL(k)|_{max}$ represents the maximum of the absolute value of the INL among all DCDL stages. Since a Gaussian distribution is used to model the jitter, the variability of this magnitude is expected to be comprised within 3 standard deviations (three-sigma rule of thumb [130]).

The absolute value of the INL of the line when lock is achieved has been computed for the three corners evaluated and different values of σ_j. The resulting curves are compiled in Appendix B. This INL result takes into account the non-idealities in the implementation of the dDLL (ADB layout mismatch, load effects in the interface PD-DCDL, PVT variations...) and the divergence in the fine control bit values along the DCDL when lock is achieved.

The curves depicted in Figure 12.1 correspond to ordering option B, out of the 4 alternative matrices of fine control bit state introduced in Section 11.4; and σ_j = 3 ps, which is the largest value of standard deviation for which the aforementioned time error target is met for all corners. In a more realistic scenario, the 9 ps time error budget is shared by jitter and static time errors that are not comprised in the INL result, such as local IR-drops, temperature gradients, etc.

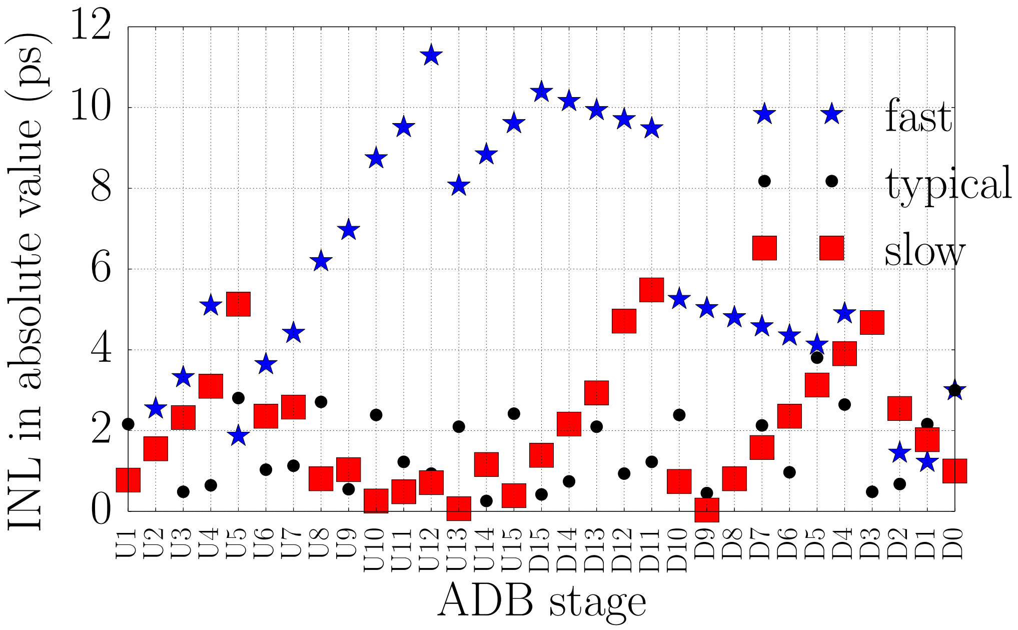


FIGURE 12.1: Absolute value of the DCDL INL at the output of every ADB stage, for ordering option B and 3 ps as standard deviation of the jitter superimposed to ckin_up.

Table 12.1 provides further information on the time performance of the dDLL, compiling the ADB LSB values and the range of adjustment of the DCDL delay for the different PVT corners. This result indicates that the latencies of the line can be updated in steps finer than the TDC time bin and that the master clock period (25 ns) can be accommodated in the range of available delays in all corners.
The number of master clock cycles required to achieve lock from the time when an asynchronous reset is applied is listed as well, for different σ_j values. The time required to lock increases when σ_j is comparable to ADB LSB, which is explained by the action of the low-pass digital filter of the PD: when the standard deviation is comparable to this value, there is a more significant ringing in `up_or_downn_aux`, which forces that the counter of the filter is reset more often, and thus more cycles need to be processed to generate a pulse at `clk_PD_ready` and update the controller.
The time required to lock depends on the corner according to the sweep in the DCDL delay performed by the controller: in the fast corner, the sweep relies mainly on the coarse control bits, while in the slow corner the controller sweeps mainly the fine control bits, which is a slower operation.
The time required to lock is also expressed in µs in Table 12.2.

12.1.2 Time resolution of the PD

The time resolution or sensitivity window of the PD stands for the range of time differences between `ckin_up` and `ckout_down` for which no `clk_PD_ready` pulse is generated, that is, the dDLL is in lock. Ideally, this window corresponds to +/- ADB LSB.

TABLE 12.1: ADB LSB, range of adjustment of the DCDL delay, number of master clock cycles required to lock as a function of σ_j (σ_j is expressed in ps).

Corner	ADB LSB (ps)	Min. delay (ns)	Max. delay (ns)	Num. clock cycles lock for various σ_j				
				1	2	3	4	5
Fast	4	11.14	26.24	6481	8389	10297	5512	5649
Typical	5	16.21	41.04	11386	15113	17697	7105	7278
Slow	7	24.64	67.57	14088	14182	14712	15144	16212

TABLE 12.2: Locking time (µs) as a function of σ_j (σ_j is expressed in ps).

Corner	1	2	3	4	5
Fast	162	210	257	138	141
Typical	285	378	442	178	182
Slow	352	355	368	379	405

As it was introduced in Section 11.3, due to the jitter superimposed to `ckin_up` and the setup-and-hold window limitation of the FFs that sample the time difference of interest, the fast variability in the PD outputs widens the sensitivity window. To prevent such a deterioration, the PD outputs are low-pass filtered by propagating roughly one in every 16 `clk_PD_ready_aux` cycles.

In this section, the impact of the low-pass filter on the time resolution of the PD is evaluated, so as to determine whether lock can be achieved and if the static time error in stage D0 when lock is achieved is bound to +/- ADB LSB.

The time resolution before and after the low-pass filter is reported. The method used to measure this magnitude before the filter is illustrated in Figure 12.2. In the top half of the image, an ideal PD, with no jitter superimposed to `ckin_up` and no setup-and-hold limitation, is shown. As the delay of the line is swept from values lower than the master clock period towards values larger than the master clock period (by means of the debug operation mode of the controller), the `up_or_downn` output goes from high to low; `q_o` has a falling edge when `ckout_down` arrives earlier than `ckin_up` by a time difference of ADB LSB (point `S` or start of the sensitivity window); and `q_i` has a rising edge when `ckout_down` arrives later than `ckin_up` by a time difference of ADB LSB (point `E` or end of the sensitivity window). The range of time differences that span between `S` and `E`, for which the OR of `q_i`, `q_o` is 0 and thus no `clk_PD_ready` pulse is generated, corresponds to the time resolution of the ideal PD.

The bottom half of the image illustrates the actual behavior of the PD. For small values of the time difference of interest, ringing can be observed in `up_or_downn_aux` due to the presence of jitter and setup and hold time violations.
When such violations occur, the output of the first FF in the 2-FF synchroniser that generates `up_or_downn_aux`, which is denoted as `up_or_downn_aux(q1)` in the figure, becomes undefined (red vertical lines) and collapses randomly to 0 or 1 after a certain time, to model the evolution that a metastable signal would have. The presence of a timing violation is indicated with a short pulse in an internal signal of the

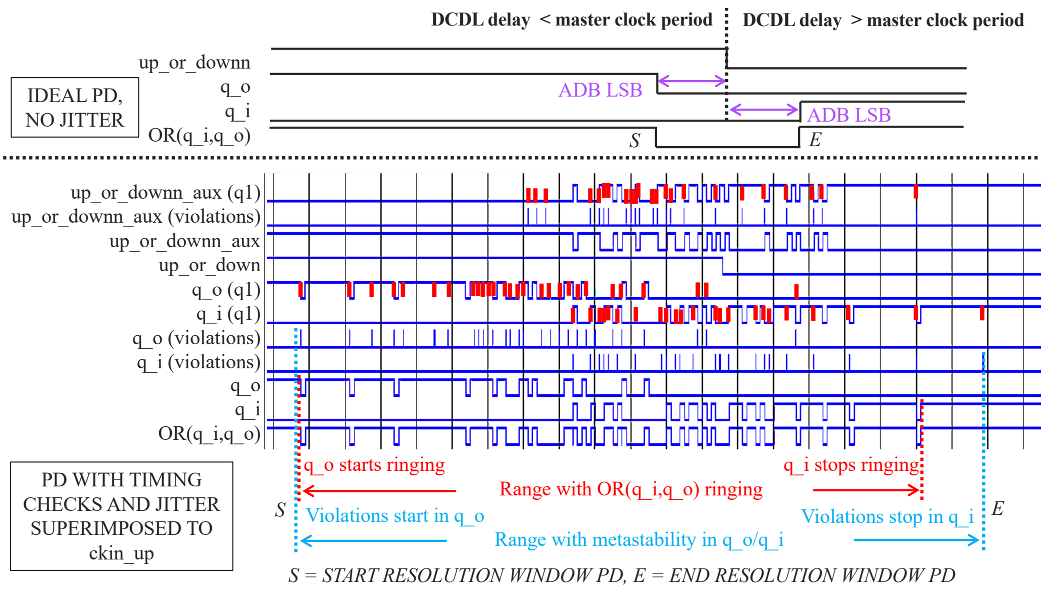

DCDL delay < master clock period
DCDL delay > master clock period
IDEAL PD, NO JITTER
up_or_downn
q_o
q_i
OR(q_i,q_o)
ADB LSB
ADB LSB
S
E
up_or_downn_aux (q1)
up_or_downn_aux (violations)
up_or_downn_aux
up_or_down
q_o (q1)
q_i (q1)
q_o (violations)
q_i (violations)
q_o
q_i
OR(q_i,q_o)
PD WITH TIMING CHECKS AND JITTER SUPERIMPOSED TO ckin_up
q_o starts ringing
Range with OR(q_i,q_o) ringing
q_i stops ringing
S
Violations start in q_o
Range with metastability in q_o/q_i
Violations stop in q_i
E
S = START RESOLUTION WINDOW PD, E = END RESOLUTION WINDOW PD

FIGURE 12.2: Definition of the PD resolution margins before the filter.

first FF of the synchroniser, the line named `up_or_downn_aux(violations)`.

Ringing is also observed in `q_o` and `q_i`, the signals that define the sensitivity window of the PD. The timing violations that occur for small values of the time difference of interest are shown in lines `q_o(q1)` and `q_i(q1)`, which stand for the output of the first FF in the 2-FF synchronisers that generate `q_o` and `q_i`, respectively. The short pulses in `q_o(violations)`, `q_i(violations)`, which stand for an internal signal of the first FF in those synchronisers, denote the occurrence of such violations.
Note that even if timing violations occur, these might not be reflected as ringing in `q_o`, `q_i`, for those cases in which the output of the concerned FFs collapses to the right polarity.

With this observation in mind, the start of the sensitivity window before the filter, `S before`, is defined as the time difference of interest (when the delay of the line is smaller than the master clock period) for which either a) pulses are first observed in `q_o(violations)`, an indication that timing violations start occurring, or b) ringing is observed in `q_o`, which is an indication of the presence of timing violations and/or jitter. The most restrictive of these two options, whichever yields the earliest time difference in the sweep, is taken into account.
The end of the sensitivity window before the filter, `E before`, is defined as the time difference of interest (when the delay of the line is larger than the master clock period) for which either a) pulses are no longer observed in `q_i(violations)`, indication that timing violations stop occurring, or b) ringing is no longer observed in `q_i`, which is an indication of the presence of timing violations and/or jitter. The most restrictive of these two options, whichever yields the latest time difference in the sweep, is taken into account.

The aforementioned method is not available to provide the resolution after the filter: during the debug operation mode, the controller sets force_clk_PD_ready high, so a clk_PD_ready_aux pulse is generated for every clock edge entering the delay line, independently from the time difference between ckin_up and ckout_down. As a result, the values of up_or_downn and clk_PD_ready (the filtered outputs of the PD) do not reflect the impact of jitter and timing violations.

An alternative strategy has been applied to determine the sensitivity window of the PD after the filter, S after and E after. If the dDLL is reset and let run for the number of cycles required to lock, the controller will increase the delay of the line by incrementing the coarse control bits until such a delay exceeds the master clock period, and from there it will start decreasing the delay in steps of 1 ADB LSB until the total delay is 1 master clock period +/- ADB LSB.
At this point, lock is achieved, clk_PD_ready pulses should not be generated any further and the INL at the last stage of the line, D0, indicates the deviation of the total delay from the master clock period.
If the INL at D0 is positive, the controller has stopped the search closer to the upper limit of the PD sensitivity window after the filter, and INL(D0) represents E after.
If the INL at D0 is negative, the controller has stopped the search closer to the lower limit of the PD sensitivity window after the filter, and INL(D0) represents S after.

S before, E before and E after obtained following these methods are depicted in Figure 12.3, as a function of σ_j, for the three PVT corners considered.
The dots in the bottom half represent S before, and the dashed lines that interconnect them are the linear fit of the variable for every corner. The resulting equation of the fit is shown in the bottom left corner of the plot.
In the top half of the plot, the red, black and blue dots represent E before, and the yellow, green and purple dots represent E after. The dashed lines that interconnect them are the linear fit of the respective variables for every corner, and the resulting equations of the fit are shown above the plot. The numbers in brackets stand for the factor by which the (offset,slope) of the E after linear fit are reduced with respect to the E before linear fit.
The unbroken lines at the bottom half of the plot represent the ideal position of S or start of the resolution window (– ADB LSB); and those at the top half of the plot correspond to the ideal E or end of the resolution window (+ ADB LSB).

From these results, S before can be approximated by $-(S_0 + 3\sigma_j)$, where S_0 is the absolute value of S before when $\sigma_j = 0$ ps, and it corresponds to the setup-and-hold window of the first FF in the 2-FF synchroniser that generates q_o. From the behavior explained with Figure 11.15 and Figure 12.2, the falling edge of q_o arrives when the time difference of interest brings the edge of ckin_o (which is used as FF clock input) too close to the edge of ckout_o (which is used as FF data input), so S_0 should be dominated by the setup time of the first FF in the synchroniser that generates q_o.
$3\sigma_j$ is the largest variation in the time difference of interest due to jitter.

E before can be approximated by $+(E_0 + 3\sigma_j)$, where E_0 is the absolute value of E before when $\sigma_j = 0$ ps, and it corresponds to the setup-and-hold window of the first FF in the 2-FF synchroniser that generates q_i. In the sweep of the DCDL delay, the rising edge of q_i arrives when the time difference of interest brings the edge of

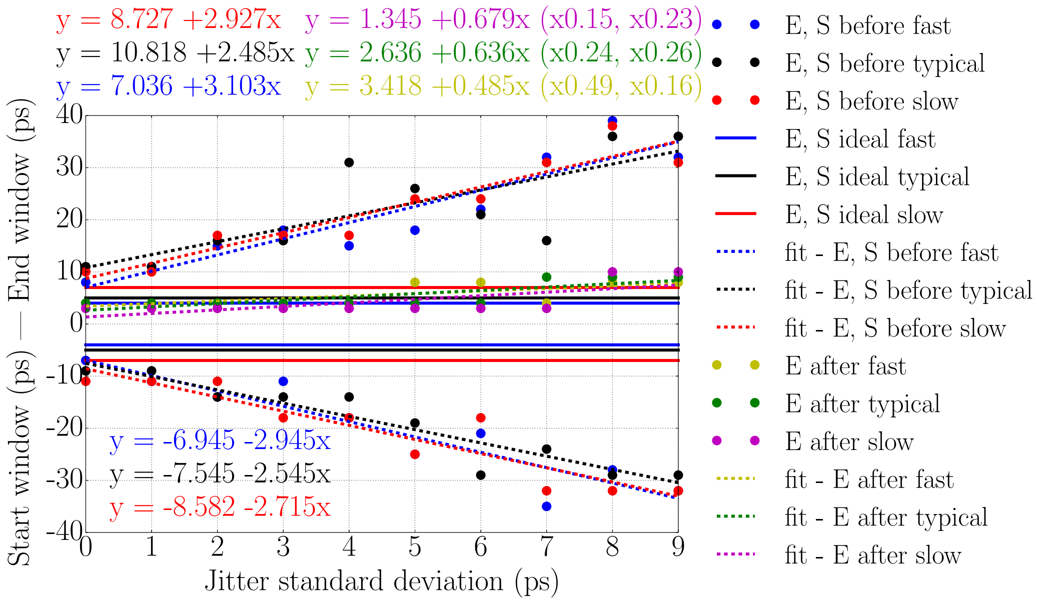


FIGURE 12.3: PD resolution, expressed as `S before`, `E before` and `E after`, and their respective linear fits.

`ckout_i` (which is used as FF clock input) too close to the edge of `ckin_i` (which is used as FF data input), so E_0 should be dominated by the hold time of the first FF in the synchroniser that generates `q_i`. $3\sigma_j$ is the largest variation in the time difference of interest due to jitter.

The setup-and-hold window is larger than ADB LSB and jitter contributes to further widen the (`S before`, `E before`) window, showcasing how these effects deteriorate the time resolution and deviate it from the ideal +/- ADB LSB margin.

On the other hand, `E after` can be roughly approximated as `E before`/4, where such a reduction factor stands for the square root of the digital filter window (the smallest number of `clk_PD_ready_aux` pulses that are processed before generating a `clk_PD_ready` pulse).
Given the symmetry observed between `E before` and `S before`, `S after` could be approximated as -`E after`. As a result, the use of this digital filter provides a 4-fold enhancement in the resolution window of the PD with respect to the window before the filter, which enables achieving lock and that the required time resolution of +/- ADB LSB is honored. In addition, `E after` is approximately ADB LSB, thus indicating an adequate upper bound of the INL at D0.

12.2 Power consumption

The total power consumption of the dDLL and its blocks, including switching, leakage and internal components, is listed in Table 12.3 for the three corners considered. The highest allowed σ_j (3 ps) is reported.

The reported power corresponds to a simulation in which the dDLL is reset, let run until lock is achieved and remain in lock for a few thousand cycles. The same number of cycles is reported for the three corners.

TABLE 12.3: Power consumption of the dDLL components and total power consumption of the dDLL when operating at 40 MHz and σ_{j} = 3 ps.

Corner	Power consumption PD (µW)	Power consumption controller (µW)	Power consumption ADB (µW)	Total power consumption dDLL, DCDL 32 stages (µW)
Fast	45.66	1.62	23.39	795.90
Typical	34.83	1.13	15.58	534.58
Slow	26.70	1.82	10.85	375.86

The estimated total power consumption of the CDN at the chip level for the different chip areas and 376 µm pixel pitch is reported in Table 12.4. It is calculated from the values reported in Table 12.3, for the worst case power consumption (fast corner) and scaling the consumption with the number of stages in the DCDL, number of dDLLs in the chip and master clock frequency (according to the guidelines provided in Table 11.5). The expression used to estimate the total power consumption of the CDN at the chip level is the following: $P_{CDN} = k_{dDLL} \times P_{dDLL}$.

P_{CDN} is the estimated total power consumption of the CDN at the chip level, k_{dDLL} is the number of dDLLs in the CDN and P_{dDLL} is the estimated power consumption of 1 dDLL, which is calculated as: $P_{dDLL} = k_f \times (P_{ctrl} + P_{PD} + k_{ADB} \times P_{ADB})$, with:

- k_f: scale factor related to the master clock frequency, calculated as the quotient of the frequency in the particular scenario (MHz) over 40 MHz, since the switching frequency is the dominant contribution (over 90% of the power reported in Table 12.3) and it scales linearly with frequency [103].
- P_{ctrl}, P_{PD}, P_{ADB}: controller, PD and ADB power consumption, respectively.
- k_{ADB}: 0.5 for the smallest chip area, since in this case the ADBs introduce half the delay and thus have a smaller coarse section, which leads to a smaller power consumption; 1 for the rest of scenarios.

For a larger segmentation factor (i.e. smaller pixel pitch), the aforementioned estimated power consumption due to the dDLL is not expected to change, since the design of the dDLL blocks will be the same. The larger number of TDC targets will be handled by the local clock tree that starts at the output of each ADB, which will increase in complexity with the number of TDCs expected for a smaller pixel pitch.

12.3 Reaction to a perturbation in the input clock

Once the operation of the dDLL has been characterised, it is important to evaluate how it would react if there were a perturbation in the master clock, to make sure that

TABLE 12.4: Total estimated power consumption of the CDN scaled with the chip area for the worst case scenario (fast corner, σ_j = 3 ps).

Chip area (cm^2)	0.3 × 0.3	0.6 × 0.6	0.9 × 0.9	1.2 × 1.2	1.5 × 1.5	1.8 × 1.8	2.1 × 2.1	2.4 × 2.4
P_{CDN} (mW)	0.56	1.69	3.65	6.37	10.30	14.61	19.67	25.47

lock can still be achieved and thus guarantee the correct delivery of the master clock to the TDCs.

Three types of perturbation are considered:

1. Interruption of the master clock during a few milliseconds.
2. Sudden change in the frequency of the master clock to a value for which lock should still be achieved (the new period is comprised within the range of DCDL delay adjustment). A change from 40 MHz to 45 MHz is considered.
3. Sudden change in the clock phase: the portion at which the clock is low is distorted for one period, made much shorter than the normal duration.

Such perturbations are applied in two scenarios: when lock has already been achieved, and while the controller is still sweeping the DCDL delay and lock has not been achieved yet. The resulting behavior of `clk_PD_ready` is show in Figure 12.4. The moment when the perturbation is applied to `ckin_up` is highlighted with a red arrow.

When the perturbation is applied once lock has already been achieved (when `clk_PD_ready` pulses are no longer generated), perturbations 1 and 3 do not take the dDLL out of lock. Perturbation 2 does take the dDLL out of lock, but the controller restarts sweeping the delay of the line until it locks to the new value of master clock period.
When the perturbation is applied during the search for lock, the controller can handle the three types of perturbation and eventually achieve lock.

The impact of these perturbations on the time required to lock is quantified in Table 12.5. This table lists the number of clock cycles required to achieve lock, with σ_j = 3 ps (the largest standard deviation of the jitter superimposed to `ckin_up` for which the time error target is still met, as it was introduced in Section 12.1.1). The three corners used in the former sections are also considered here.
The leftmost column is used as the performance reference: these values indicate the number of clock cycles required to achieve lock when no perturbation is applied (this result was shown in Table 12.1 and it is repeated here for convenience).

The next column to the right corresponds to the scenarios when the perturbation is applied once lock has already been achieved. As it was mentioned before, perturbations 1 and 3 do not take the dDLL out of lock, so the number of clock cycles is the same as for the case when no perturbation is applied.

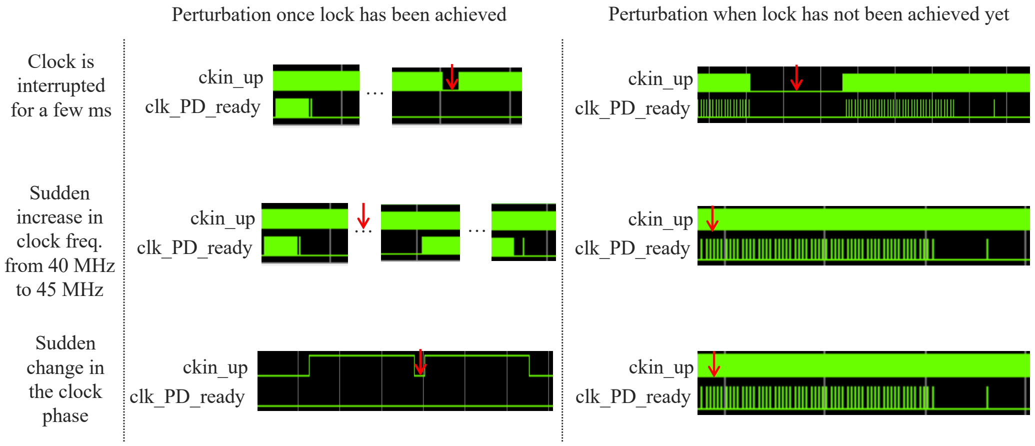


FIGURE 12.4: Reaction of the dDLL to three kinds of perturbation in the input clock.

In scenario 2, the dDLL loses lock and starts updating the line delay until lock is regained. Two numbers are reported in the boxes corresponding to this scenario: the number of clock cycles required to achieve lock in the first place + the number of clock cycles required to regain lock once the perturbation is released.
The time required to regain lock is comparable to that spent to achieve lock in the first place, because the time consumption in both operations is dominated by a sweep in the fine control bits of the buffers.
The percentages shown in brackets correspond to the increase in the time required to lock (i.e. the time to reach lock in the first place plus the time to regain lock, if necessary) with respect to the column in which no perturbation is applied.

The rightmost column corresponds to the scenarios when the perturbation is applied before lock has been achieved. The numbers listed in this column are the addition of the clock cycles that span between the release of the reset and the time when the perturbation is applied, plus the cycles spent between the release of the perturbation and the achievement of lock. In other words, it is the total number of cycles required to lock, minus the duration of the perturbation.
The numbers shown in brackets correspond to the percentage of increase in the time required to achieve lock, with respect to the column in which no perturbation is applied.

It can be seen that the impact (in terms of cycles required to lock) of applying the perturbation before lock is achieved in the first place is significantly lower than the case in which lock needs to be regained. In all cases, the typical corner shows the smallest sensitivity to the perturbation.

Table 12.6 provides the same information as Table 12.5, but expressed in µs.

TABLE 12.5: Number of clock cycles required to lock with σ_j = 3 ps.

Corner	No perturbation applied	Perturbation applied after lock is reached			Perturbation applied before lock is reached		
		Scenario 1	Scenario 2	Scenario 3	Scenario 1	Scenario 2	Scenario 3
Fast	10297	10297 (+0%)	10297 + 15735 (+253%)	10297 (+0%)	18404 (+79%)	20563 (+100%)	28134 (+173%)
Typical	17697	17697 (+0%)	17697 + 11874 (+167%)	17697 (+0%)	17549 (-1%)	18498 (+5%)	24631 (+39%)
Slow	14712	14712 (+0%)	14712 + 15523 (+206%)	14712 (+0%)	16724 (+14%)	26841 (+82%)	13979 (-5%)

TABLE 12.6: Number of clock cycles required to lock with σ_j = 3 ps, expressed in µs.

Corner	No perturbation applied	Perturbation applied after lock is reached			Perturbation applied before lock is reached		
		Scenario 1	Scenario 2	Scenario 3	Scenario 1	Scenario 2	Scenario 3
Fast	257	257 (+0%)	257 + 393 (+253%)	257 (+0%)	460 (+79%)	514 (+100%)	703 (+173%)
Typical	442	442 (+0%)	442 + 297 (+167%)	442 (+0%)	439 (-1%)	463 (+5%)	616 (+39%)
Slow	368	368 (+0%)	368 + 388 (+206%)	368 (+0%)	418 (+14%)	671 (+82%)	350 (-5%)

13 Conclusion and future work

In the former chapters, a self-regulated CDN for the timestamp mechanism of the FastICpix chip has been presented, and its expected performance has been characterised by means of digital, back-annotated simulations of the post-layout, flattened netlist of its main component, the dDLL.

The selected architecture has the potential to address the challenges posed by the FastICpix features, namely 1) adaptability to arbitrary chip area and pixel pitch dimensions, and 2) robustness to static and dynamic time errors, so that the highest total time error in the delivery of the master clock to all target TDCs is below the TDC time bin, 20 ps.
Comparing the performance observed with post-layout simulations of the FastICpix dDLL to that reported for the Timepix4 dDLL [77], which was the starting point of this work, the network latencies can be adjusted in steps more than an order of magnitude finer; the PD has a time resolution more than an order of magnitude finer; and the static time error of the delay line is an order of magnitude finer in the case of the FastICpix design, thus showcasing the benefits of the selected architecture.
Moreover, this work has provided an innovative phase detector design and a new control strategy to regulate the fine delay section of the ADBs individually.

To comply with the scalability requirement, the number of stages of the DCDL and number of dDLLs in the CDN increases with the chip area, while the master clock frequency decreases with the chip area.
Adaptation to the pixel pitch is provided at the level of the local clock tree that starts at the output of each ADB and drives the TDCs in the corresponding group of pixels.

To provide a fine time resolution, the CDN latencies can be adjusted in steps of 5 ps in the typical corner (7 ps in the slow corner). The highest static time error of the DCDL is below 20 ps in the all corners, allowing for a budget of around 9 ps for the maximum deviation caused by jitter and static effects not reflected in the simulation.

Concerning the power consumption of the total CDN at the chip level, it has been estimated that it will range from below a milliwatt for the smallest chip area and 376 µm pixel pitch, to about 26 mW for the largest chip area and same pixel pitch. These values correspond to the contributions of the dDLL components; therefore, they are not expected to change when the pixel pitch is scaled, since the adaptation to the pixel pitch is provided at the level of the local clock tree that starts at the output of each ADB.
With a smaller pixel pitch, the number of sinks in such a clock tree is expected to increase by a factor of (376 µm/new pixel pitch), leading to a comparable increment

in the power consumption of the clock tree.
According to the reported power consumption, the CDN is not expected to be the dominant contribution to the overall chip power consumption, hence research towards a less power-hungry configuration is not the main priority.

Further development can be proposed for the PD, including several aspects:

- An ad-hoc design has been used for the low-pass digital filter. The dimension of the filter window has been selected from the simulation results: it is the smallest window for which the reduction in the variability of the PD outputs enables the achievement of lock and that the PD resolution is close to +/- ADB LSB.
  There exist systematic techniques towards the design and analysis of such filters in applications that face similar challenges, such as high-speed time-interleaved Analog-to-Digital Converters (ADCs), where static time errors between the different subconverters give rise to non-uniform sampling and thus deteriorate the resolution of the converter. To reconstruct the resulting, non-uniformly sampled, band-limited signals, the usage of time-varying discrete-time Finite Impulse Response (FIR) filters is proposed in [59], [123].
  This approach could be explored to evaluate whether such filters are suitable for the FastICpix PD and enhance the performance of the proposed solution.

- 2-FF synchronisers are used to sample the time difference between the clock inputs of the PD, which is a simple strategy to reduce the risk of propagating metastable signals down the line. Other synchroniser solutions exist, which are more robust, but also more complex: handshaking protocols, FIFOs, etc. [116] [65] [64] [44] Some of these alternatives could be explored to determine whether they can enhance the robustness and the resolution of the PD.

- Standard-cell FFs are used in the aforementioned synchronisers for several reasons: 1) the simplicity to integrate such cells into the digital design flow followed to implement the dDLL; 2) a complete characterization (liberty files, Verilog library, noise models, etc.) is available from the foundry, including datasheets that have been used to select the FF with the narrowest setup-and-hold window, so as to limit the deterioration on the PD time resolution; 3) such cells can be safely characterised with Cadence® Liberate Characterization Solution™(the tool identifies correctly the logic function and the timing arcs that should be extracted from the characterization), thus providing a robust and complete set of files (liberty files, Verilog library, etc.) that describe the cell. This characterization has been used to constraint the setup-and-hold window definition to a narrower value by allowing a larger degradation in the clock-to-output propagation delay at which the setup and hold times are measured, as it was introduced in Chapter 10.
  There are alternative, custom FF designs that have the potential to achieve an even narrower setup-and-hold window [99] [66] [93], and which could be explored. If the same digital design flow is to be applied as with the standard-cell FF, the first step towards the integration of these alternative FFs would be to obtain a reliable digital characterization with Cadence® Liberate Characterization Solution™, which can be challenging.

In this work, digital simulations based on PVT corner models have been used to characterise the performance of the dDLL, which provides the best and worst-case performance range due to device and interconnect variation. Monte-Carlo simulations on transistor level [57] could be performed to complement this result, so as to observe the statistical distribution of such variations, not only the extreme values.

Concerning dynamic time errors, the impact of jitter has been modelled by adding a random fluctuation (with a Gaussian distribution) in the edges of the dDLL clock input. To evaluate the actual impact of jitter present in the implemented circuit, analog simulations can be performed including the clock source (to quantify the jitter superimposed to the clock injected into the dDLL) and span a sufficient number of cycles (thousands of cycles [29]) to characterise PSIJ generated by the switching activity of the circuit.

In Figure 12.2, the sensitivity window of the PD was evaluated as a function of the standard deviation of the jitter superimposed to the dDLL input clock, σ_j, up to a value as high as 9 ps. This limit was chosen because the resulting peak-to-peak jitter ($6\sigma_j$) is comparable to that observed in analog simulations of a PLL used in a previous chip developed in the section. This is not the actual performance limit of the dDLL: it can still reach lock with higher values of σ_j, at the cost of increasing the time required to lock.
Further characterisation could be performed to determine 1) which is the limit σ_j, from which lock cannot be further achieved, and 2) which is the highest acceptable degradation in the time required to lock (it might yield a smaller allowed σ_j than the value determined in the former point).

On the skew that can occur between neighboring dDLLs, in Section 11.1 the usage of a horizontal dDLL, which regulates the clock distribution across the chip width, was proposed to reduce such a skew and thus guarantee that the time error target can be met both along the pixel columns and rows.

In the present stage of the project, the local clock tree that starts at the output of every ADB to serve a group of pixels has not been implemented yet. Next stages of the project should include the implementation of this tree, and the evaluation of the area and power overhead it adds to the reported dDLL features.

Part IV

SCIENTIFIC CONTRIBUTIONS

Journal

[1] R. Ballabriga et al. "Photon Counting Detectors for X-ray Imaging with Emphasis on CT". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 7311.c (2020). DOI: 10.1109/TRPMS.2020.3002949.

[2] N. Egidos et al. "20-ps resolution Clock Distribution Network for a fast-timing single photon detector". In: *IEEE Transactions on Nuclear Science (submitted)* (2020).

[3] Iraklis Kremastiotis et al. "Design and characterisation of the CLICTD pixelated monolithic sensor chip". In: *IEEE Transactions on Nuclear Science* (2020), pp. 1–10. DOI: 10.1109/TNS.2020.3019887.

[4] X. Llopart et al. "Study of low power front-ends for hybrid pixel detectors with sub-ns time tagging". In: *Journal of Instrumentation* 14.1 (2019). ISSN: 17480221. DOI: 10.1088/1748-0221/14/01/C01024.

Conference attendance

Oral contribution in the IEEE Nuclear Science Symposium (NSS) and Medical Imaging Conference (MIC), NSS-MIC 2020, 31st October - 7th November 2020 [53].

Conference proceedings

[1] N. Egidos et al. "Self-regulated Clock Distribution Network for a fast-timing active hybrid single photon detector". In: *Proceedings of the IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC) (submitted)*. Boston, USA, 2020.

[2] D. Gascón et al. "Integrated signal processing for a new generation of active hybrid single photon sensors with ps time resolution (FastICpix)". In: *Public deliverable for the ATTRACT Final Conference* (2020).

[3] I. Kremastiotis, R. Ballabriga, and N. Egidos. "Design of a monolithic HR-CMOS sensor chip for the CLIC silicon tracker". In: *Proceedings of the Topical Workshop on Electronics for Particle Physics (TWEPP 2018)*. CERN. Antwerp, Belgique, 2018. DOI: 10.22323/1.343.0072.

[4] I. Kremastiotis et al. "CLICTD: A monolithic HR-CMOS sensor chip for the CLIC silicon tracker". In: *Proceedings of the Topical Workshop on Electronics for Particle Physics (TWEPP 2019)*. Santiago de Compostela, Spain, 2019. DOI: 10.22323/1.370.0039.

[5] V. Sriskaran et al. "Novel architecture for the analog front-end of Medipix4". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*. Hiroshima, Japan, 2019.

Workshops

Presentation under the title “Self-regulated Clock Distribution Network for the FastICpix fast-timing single photon detector” in the md-NUV PET Workshop 2020, 18th June 2020.

Contributions prior to the thesis

[1] L. Argemi et al. “Development of the monolithic "MALTA" CMOS sensor for the ATLAS ITk outer pixel layer”. In: *Proceedings of the Topical Workshop on Electronics for Particle Physics (TWEPP18)*. Antwerp, Belgique, 2018. DOI: 10.22323/1.343.0155.

[2] I. Berdalovic et al. “Monolithic pixel development in TowerJazz 180 nm CMOS for the outer pixel layers in the ATLAS experiment”. In: *Proceedings of the 11th International Conference on Position Sensitive Detectors (PSD11)*. The Open University, Milton Keynes, UK, 2019. DOI: 10.1016/j.nima.2018.07.043.

[3] R. Cardella et al. “LAPA, a 5 Gb/s modular pseudo-LVDS driver in 180 nm CMOS with capacitively coupled pre-emphasis”. In: *Proceedings of the Topical Workshop on Electronics for Particle Physics 2017 (TWEPP 2017)*. Santa Cruz, USA, 2017. DOI: 10.22323/1.313.0038.

[4] N. Egidos et al. “On frequency optimization of assymetric resonant inductive coupling wireless power transfer links”. In: *Progress in Electromagnetics Research Symposium* January (2014), pp. 885–889. ISSN: 19317360.

[5] K. Moustakas et al. “CMOS monolithic pixel sensors based on the column-drain architecture for the HL-LHC upgrade”. In: *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 936.August 2018 (2019), pp. 604–607. ISSN: 01689002. DOI: 10.1016/j.nima.2018.09.100. arXiv: 1809.03434.

[6] M. Saad et al. “On tunable switch-mode reactive networks: A gyrator-based resonator emulation”. In: *Proceedings - IEEE International Symposium on Circuits and Systems*. Montreal, Canada, 2016. DOI: 10.1109/ISCAS.2016.7527322.

[7] A. Sharma et al. “Update on the TowerJazz CMOS DMAPS development for the ATLAS ITk”. In: *Proceedings of the 39th International Conference on High Energy Physics (ICHEP2018)*. Seoul, China, 2018.

Appendices

A Matrix of fine control bit state for the different ordering options

Figure A.1 shows the matrix of fine control bit state corresponding to the four ordering options (A-D) introduced in Section 11.4.2, for the largest FastICpix scenario (DCDL of 32 stages). The ideal scenario (maximum alternation in the fine control bit state along the line) is obtained in the middle of the update sequence for all ordering options.

U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0
1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0
1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0
1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0
1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0
1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0
1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(A) Ordering option A.

U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0
1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0
1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0
1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0
1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0
1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0
1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0
1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0
1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1
1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1
1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1
1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(B) Ordering option B.

U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0
1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0
1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0
1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0
1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0
1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0
1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0
1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0
1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0
1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0
1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0
1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0
1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0
1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(C) Ordering option C.

U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0
0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0
0	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1
1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1
1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	1
1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	1
1	0	1	0	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1
1	0	1	1	1	0	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

(D) Ordering option D.

FIGURE A.1: Matrix of fine control bit state for the different ordering options.

B INL obtained with the different ordering options

Figure B.1 is a compendium of INL (in absolute value) results when the dDLL is in lock for the different ordering alternatives (A-D); different standard deviation values of the random, Gaussian jitter superimposed to `ckin_up`; 32 stages in the DCDL, 40 MHz master clock frequency; post-layout netlist of the dDLL, flattened, back-annotated, with all timing checks enabled.
The message to underline from these results is that the static time performance of all ordering options is very similar and slightly better with the selected option, B; and that $\sigma_j = 3$ ps is the largest standard deviation value for which the time error target defined in Section 12.1 ($|INL(k)|_{max} + 3\sigma_j <$ 20ps) can be met in all corners.

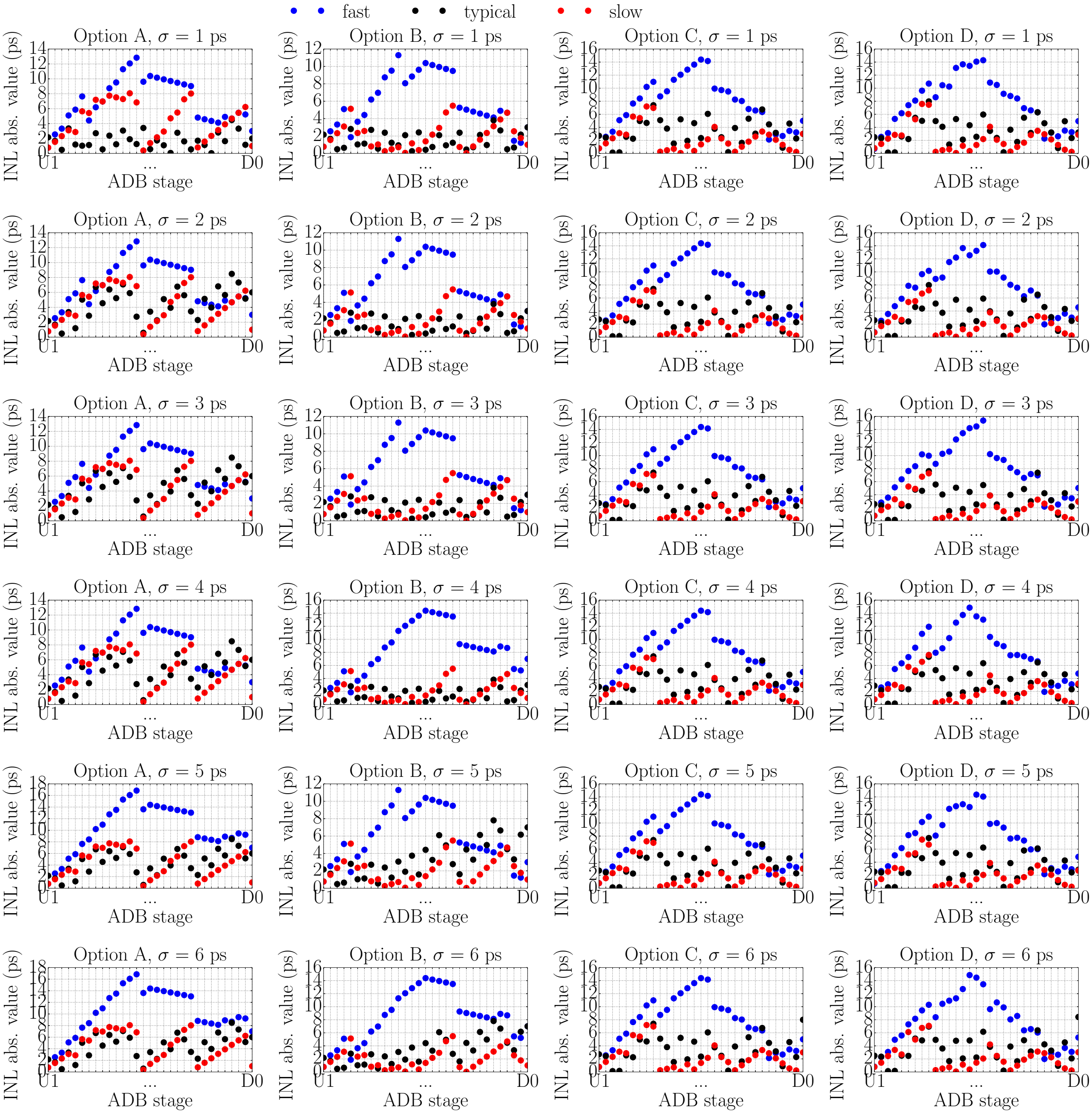


FIGURE B.1: Absolute value of the DCDL INL obtained for the different ordering options and standard deviation values of the random, Gaussian jitter superimposed to `ckin_up`.

Acronyms

ADB Adjustable Delay Buffer.

ADC Analog-to-Digital Converter.

APD Avalanche Photo-Diode.

ASIC Application-Specific Integrated Circuit.

BOX Buried Oxide Layer.

CCD Charged-Coupled Device.

CDN Clock Distribution Network.

CLIC Compact Linear Collider.

CLICTD CLIC Tracker Detector.

CT Computer Tomography.

DCDL Digitally-Controlled Delay Line.

dDLL digital Delay-Locked Loop.

DLL Delay-Locked Loop.

DMAPS Depleted Monolithic Active Pixel Sensors.

dSiPM digital Silicon Photo-Multiplier.

DUT Device Under Test.

EM Electromigration.

FF Flip-Flop.

FIR Finite Impulse Response.

FLIM Fluorescence-lifetime Imaging Microscopy.

FSM Finite State Machine.

HCI Hot Carrier Injection.

HEP High Energy Physics.

HL-LHC High Luminosity LHC.

ILD Inter-Layer Dielectric.

INL Integral-Non-Linearity.

LER Line Edge Roughness.

LHC Large Hadron Collider.

LSB Least Significant Bit.

MAPS Monolithic Active Pixel Sensors.

MSB Most Significant Bit.

MSI Mass Spectrometry Imaging.

NBTI Negative Bias Temperature Instability.

PBTI Positive Bias Temperature Instability.

PD Phase Detector.

PDN Power Distribution Network.

PET Positron Emission Tomography.

PLL Phase-Locked Loop.

PMT Photo-Multiplier Tube.

PSIJ Power Supply Induced Jitter.

PSN Power Supply Noise.

PVT Process-Voltage-Temperature.

RDF Random Dopant Fluctuation.

SiPM Silicon Photo-Multiplier.

SNR Signal-to-Noise Ratio.

SOI Silicon-On-Insulator.

SPAD Single Photon Avalanche Diode.

SPTR Single Photon Time Resolution.

TDC Time-to-Digital Converter.

TDDB Time-Dependent Dielectric Breakdown.

TIE Time Interval Error.

ToA Time of Arrival.

ToT Time-over-Threshold.

TSV Trough-Silicon Via.

UVM Universal Verification Methodology.

VCD Value Change Dump.

VCO Voltage-Controlled Oscillator.

WPE Well Proximity Effect.

Bibliography

[1] Bilal I Abdulrazzaq et al. "A review on high-resolution CMOS delay lines: towards sub-picosecond jitter performance". In: *SpringerPlus* 5.1 (2016), p. 434.

[2] Bilal I. Abdulrazzaq et al. "Design of a sub-picosecond jitter with adjustable-range CMOS delay-locked loop for high-speed and low-power applications". In: *Sensors (Switzerland)* 16.10 (2016). ISSN: 14248220. DOI: 10.3390/s16101593.

[3] Accellera. *Universal Verification Methodology (UVM) 1.2 User's Guide*. http://www.accellera.org/images//downloads/standards/uvm/uvm_users_guide_1.2.pdf. 2015.

[4] Accellera. *Verification Methodology Cookbooks*. https://verificationacademy.com/cookbook.

[5] R L Aguiar and D M Santos. "Wide-area clock distribution using controlled delay lines". In: *Electronics, Circuits and Systems, 1998 IEEE International Conference on* (1998). DOI: 10.1109/ICECS.1998.814825.

[6] Amir H. Ajami, Kaustav Banerjee, and Massoud Pedram. "Analysis of substrate thermal gradient effects on optimal buffer insertion". In: *IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers* 4 (2001), pp. 44–48. ISSN: 10923152. DOI: 10.1109/ICCAD.2001.968596.

[7] Erik Backenius and Linköpings universitet. Institutionen för Systemteknik. *Reduction of substrate noise in mixed-signal circuits*. Vol. 39. 1094. 2007, pp. 215–217. ISBN: 9789185715121.

[8] R. Ballabriga et al. "Photon Counting Detectors for X-ray Imaging with Emphasis on CT". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 7311.c (2020). DOI: 10.1109/TRPMS.2020.3002949.

[9] R. Ballabriga et al. "The medipix3RX: A high resolution, zero dead-time pixel detector readout chip allowing spectroscopic imaging". In: *Journal of Instrumentation* 8.2 (2013). ISSN: 17480221. DOI: 10.1088/1748-0221/8/02/C02016.

[10] Rafael Ballabriga. "The Design and Implementation in 0.13 μm CMOS of an Algorithm Permitting Spectroscopic Imaging with High Spatial Resolution for Hybrid Pixel Detectors". PhD thesis. CERN, 2009.

[11] Eric Berg and Simon R Cherry. "Innovations in instrumentation for positron emission tomography". In: *Seminars in nuclear medicine*. Vol. 48. 4. Elsevier. 2018, pp. 311–331.

[12] J Bhasker and Rakesh Chadha. *Static Timing Analysis for Nanometer Designs: A Practical Approach*. Springer, 2007. DOI: 10.1007/978-0-387-93820-2.

[13] S A Bota et al. "Within Die Thermal Gradient Impact on Clock-Skew: A New Type Of Delay-Fault Mechanism". In: *Proceedings of the ITC International Test Conference* (2004), pp. 1276–1284.

[14] Juan Pablo Martinez Brito, Marcelo Lubaszewski, and Sergio Bampi. "Within-die and die-to-die variability on 65nm CMOS : Oscillators experimental results". In: *Proceedings - 2015 6th International Workshop on CMOS Variability, VARI 2015* (2016), pp. 27–32. DOI: 10.1109/VARI.2015.7456559.

[15] James F. Buckwalter and Ali Hajimiri. "Cancellation of crosstalk-induced jitter". In: *IEEE Journal of Solid-State Circuits* 41.3 (2006), pp. 621–631. ISSN: 00189200. DOI: 10.1109/JSSC.2005.864113.

[16] Cadence. *Liberate Characterization Solution*. https://www.cadence.com/en_US/home/tools/custom-ic-analog-rf-design/library-characterization/liberate-characterization.html.

[17] Cadence. *Voltus IC Power Integrity Solution*. https://www.cadence.com/en_US/home/tools/digital-design-and-signoff/silicon-signoff/voltus-ic-power-integrity-solution.html.

[18] Cadence. *Voltus IC Power Integrity Solution User Guide*. 2017.

[19] R. J. Cernik, K. H. Khor, and C. Hansson. "X-ray colour imaging". In: *Journal of the Royal Society Interface* 5.21 (2008), pp. 477–481. ISSN: 17425662. DOI: 10.1098/rsif.2007.1249.

[20] Lucio Cerrito. *Radiation and Detectors*. Springer, 2017. ISBN: 978-3-319-53179-3. DOI: 10.1007/978-3-319-53181-6.

[21] Steven C Chan, Kenneth L Shepard, and Phillip J Restle. "Design of resonant global clock distributions". In: *Proceedings 21st International Conference on Computer Design*. IEEE. 2003, pp. 248–253.

[22] Doris Chen et al. "A comprehensive approach to modeling, characterizing and optimizing for metastability in FPGAs". In: *Proceedings of the 18th annual ACM/SIGDA international symposium on Field programmable gate arrays*. ACM. 2010, pp. 167–176.

[23] Simon R. Cherry, James A. Sorenson, and Michael E. Phelps. "Radiation Detectors". In: *Physics in Nuclear Medicine*. 2012. Chap. 7, pp. 87–106. DOI: 10.1016/B978-1-4160-5198-5.00007-1.

[24] Simon R. Cherry, James A. Sorenson, and Michael E. Phelps. "What Is Nuclear Medicine?" In: *Physics in Nuclear Medicine*. 2012. Chap. 1, pp. 1–6. DOI: 10.1016/B978-1-4160-5198-5.00001-0.

[25] J. Choi et al. "An All-Analog Multiphase Delay-Locked Loop Using a Replica Delay Line for Wide-Range Operation and Low-Jitter Performance". In: *IEEE Journal of Solid State Circuits* 35.3 (2000), pp. 377–384.

[26] J. Christiansen et al. "picoTDC: Pico-second TDC for HEP". In: *Workshop on picosecond photon sensors for physics and medical applications*. Prague, 2015.

[27] CLICdp Collaboration. *CLIC conceptual design report*. Tech. rep. February. Geneva: CERN, 2012.

[28] Fulvio Corno, Matteo Sonza Reorda, and Giovanni Squillero. “RT-level ITC’99 benchmarks and first ATPG results”. In: *IEEE Design and Test of Computers* 17.3 (2000), pp. 44–52. ISSN: 07407475. DOI: 10.1109/54.867894.

[29] Renesas Electronics Corporation. *Jitter Specifications for Timing Signals*. Tech. rep. 2014.

[30] Steve Corrigan. “Skew definition and jitter analysis”. In: *Analog Applications* (2000).

[31] A. Deisting. *Working principle and performance of GEM detectors*. Tech. rep. CERN, 2017.

[32] Grzegorz W. Deptuch et al. “Fully 3-D integrated pixel detectors for X-rays”. In: *IEEE Transactions on Electron Devices* 63.1 (2016), pp. 205–214. ISSN: 00189383. DOI: 10.1109/TED.2015.2448671.

[33] A. Drozd and T. Satława. “Trends in hybrid pixel detectors for X-ray imaging using deep submicron VLSI technology”. In: *Challenges of Modern Technology* 5.4 (2014), pp. 3–7. ISSN: 2082-2863.

[34] “Effects of Non-uniform Substrate Temperature on the Clock Signal Integrity in High Performance Designs”. In: *Proceedings of the IEEE 2001 Custom Integrated Circuits Conference*. San Diego, USA, 2001, pp. 233–236. DOI: 10.1109/CICC.2001.929762.

[35] C. K. Egan et al. “3D chemical imaging in the laboratory by hyperspectral X-ray computed tomography”. In: *Scientific Reports* 5 (2015), pp. 1–9. ISSN: 20452322. DOI: 10.1038/srep15979.

[36] L Rossi Et al. *Pixel detectors - from fundamentals to applications*. Springer, 2006. ISBN: 3540283323.

[37] J. M. Fernandez-Tenllado et al. “Optimal design of single-photon sensor front-end electronics for fast-timing applications”. In: *2019 IEEE Nuclear Science Symposium and Medical Imaging Conference, NSS/MIC 2019* (2019). DOI: 10.1109/NSS/MIC42101.2019.9059805.

[38] Adrian Fiergolski. “Simulation environment based on the Universal Verification Methodology”. In: *Journal of Instrumentation* 12.01 (2017), p. C01001. ISSN: 17480221. DOI: 10.1088/1748-0221/12/01/C01001.

[39] Mónica Figueiredo and Rui L Aguiar. “Predicting noise and jitter in CMOS inverters”. In: *2007 Ph. D Research in Microelectronics and Electronics Conference*. IEEE. 2007, pp. 21–24.

[40] E G Friedman. “Clock Distribution Networks in Synchronous Digital Integrated Circuits”. In: *Proceedings of the IEEE* 89.5 (2001), pp. 665–692. DOI: 10.1109/5.929649.

[41] Maurice Garcia-Sciveres and Norbert Wermes. “A review of advances in pixel detectors for experiments with high rate and radiation”. In: *Reports on Progress in Physics* 81.6 (2018), pp. 1–84. ISSN: 00344885. DOI: 10.1088/1361-6633/aab064. arXiv: 1705.10150.

[42] David Gascón. *Integrated signal processing for a new generation of active hybrid single photon sensors with ps time resolution (FastICpix).* https://attract-eu.com/selected-projects/integrated-signal-processing-for-a-new-generation-of-active-hybrid-single-photon-sensors-with-ps-time-resolution-fasticpix/. 2019.

[43] Ranjit Gharpurey and Robert G. Meyer. "Modeling and analysis of substrate coupling in integrated circuits". In: *IEEE Journal of Solid-State Circuits* 31.3 (1996), pp. 344–352. ISSN: 00189200. DOI: 10.1109/4.494196.

[44] Ran Ginosar. "Metastability and synchronizers: A tutorial". In: *IEEE Design and Test of Computers* 28.5 (2011), pp. 23–35. ISSN: 07407475. DOI: 10.1109/MDT.2011.113.

[45] P. Gray et al. *Analysis and Design of Analog Integrated Circuits, 5th Edition*. 5th ed. USA: John Wiley & Sons, Inc., 2009. ISBN: 9780470245996.

[46] Paul E. Gronowski et al. "High-performance microprocessor design". In: *High-Performance System Design: Circuits and Logic* 33.5 (1998), pp. 395–404. DOI: 10.1109/9780470544846.ch4.

[47] Sarah L. Harris and David Money Harris. "Sequential Logic Design". In: *Digital Design and Computer Architecture*. 2nd. Elsevier, Inc., 2016. Chap. 3, pp. 108–171. ISBN: 9780123944245. DOI: 10.1016/b978-0-12-800056-4.00003-0.

[48] D. Henry et al. "TSV last for hybrid pixel detectors: Application to particle physics and imaging experiments". In: *Proceedings - Electronic Components and Technology Conference* 1 (2013), pp. 568–575. ISSN: 05695503. DOI: 10.1109/ECTC.2013.6575630.

[49] Stephan Henzler. *Time-to-digital converters*. Springer Science & Business Media, 2010. ISBN: 978-90-481-8627-3.

[50] Frank Herzel and Behzad Razavi. "A study of oscillator jitter due to supply and substrate noise". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 46.1 (1999), pp. 56–62.

[51] Frank Herzel and Behzad Razavi. *Oscillator jitter due to supply and substrate noise*. 1998. DOI: 10.1109/cicc.1998.695025.

[52] Payam Heydari, Soroush Abbaspour, and Massoud Pedram. "Interconnect energy dissipation in high-speed ULSI circuits". In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 51.8 (2004), pp. 1501–1514.

[53] *IEEE NSS-MIC 2020*. https://nssmic.ieee.org/2020/. 2020.

[54] K. Iniewski. *Semiconductor Radiation Detection Systems*. CRC Press, 2010. ISBN: 9781439803851. DOI: 10.1201/9781315218373.

[55] *InsightART*. https://insightart.eu/2018/03/02/insightart-reveals-its-technology-to-the-world/.

[56] Ryohei Ishige. "Precise structural analysis of polymer materials using synchrotron X-ray scattering and spectroscopic methods". In: *Polymer Journal* (2020). ISSN: 13490540. DOI: 10.1038/s41428-020-0357-2.

[57] K Itoh et al. *Variation-Aware Adaptive Voltage Scaling for Digital CMOS Circuits*. 2014. ISBN: 9789400723443.

[58] J. Jakubek et al. "Large area pixel detector WIDEPIX with full area sensitivity composed of 100 Timepix assemblies with edgeless sensors". In: *Journal of Instrumentation* 9.4 (2014). ISSN: 17480221. DOI: 10.1088/1748-0221/9/04/C04018.

[59] Håkan Johansson and Per Löwenborg. "Reconstruction of nonuniformly sampled bandlimited signals by means of time-varying discrete-time FIR filters". In: *Eurasip Journal on Applied Signal Processing* 2006.1 (2006), pp. 1–18. ISSN: 11108657. DOI: 10.1155/ASP/2006/64185.

[60] Dong Hoon Jung et al. "All-Digital Fast-Locking Delay-Locked Loop Using a Cyclic-Locking Loop for DRAM". In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 62.11 (2015), pp. 1023–1027. ISSN: 15497747. DOI: 10.1109/TCSII.2015.2456111.

[61] Julia H. Jungmann and Ron M.A. Heeren. "Detection systems for mass spectrometry imaging: A perspective on novel developments with a focus on active pixel detectors". In: *Rapid Communications in Mass Spectrometry* 27.1 (2013), pp. 1–23. ISSN: 09514198. DOI: 10.1002/rcm.6418.

[62] Sarang Kazeminia, Roozbeh Abdollahi, and Arash Hejazi. "A fast-locking low-jitter digitally-enhanced DLL dynamically controlled for loop-gain and stability". In: *Analog Integrated Circuits and Signal Processing* 94.3 (2018), pp. 507–517. ISSN: 15731979. DOI: 10.1007/s10470-018-1109-5.

[63] Yong-Bin Kim. *Signal de-skewing using programmable dual Delay-Locked Loop*. https://patentimages.storage.googleapis.com/9f/28/fb/aec7d986918c30/US5880612.pdf. US Patent 5880612. 1999.

[64] David Kinniment. "Synchronization and Arbitration in GALS". In: *Electronic Notes in Theoretical Computer Science* 245 (2009), pp. 85–101. ISSN: 15710661. DOI: 10.1016/j.entcs.2009.07.030.

[65] David J. Kinniment, Alexandre Bystrov, and Alex V. Yakovlev. "Synchronization circuit performance". In: *IEEE Journal of Solid-State Circuits* 37.2 (2002), pp. 202–209. ISSN: 00189200. DOI: 10.1109/4.982426.

[66] F Klass. "Semi-Dynamic and Dynamic Flip-FLops". In: *Symposium on VLSl Circuits. Digest of Technical Papers* (1998), pp. 108–109. DOI: 10.1109/VLSIC.1998.688018.

[67] I Kremastiotis, R Ballabriga, and N Egidos. *CLICTD chip description*. https://gitlab.cern.ch/CLICdp/ASICs/CLICTD/blob/master/documentation/CLICTD chip description/CLICTD_Manual.pdf.

[68] I. Kremastiotis et al. "CLICTD: A monolithic HR-CMOS sensor chip for the CLIC silicon tracker". In: *Proceedings of the Topical Workshop on Electronics for Particle Physics (TWEPP 2019)*. Santiago de Compostela, Spain, 2019. DOI: 10.22323/1.370.0039.

[69] Iraklis Kremastiotis. "Implementation and Characterisation of Monolithic CMOS Pixel Sensors for the CLIC Vertex and Tracking Detectors". PhD thesis. Karlsruher Instituts für Technologie (KIT), 2020.

[70] Iraklis Kremastiotis et al. "Design and characterisation of the CLICTD pixelated monolithic sensor chip". In: *IEEE Transactions on Nuclear Science* (2020), pp. 1–10. DOI: 10.1109/TNS.2020.3019887.

[71] Hisaaki Kudo. *Radiation applications*. Ed. by Y. Oka et al. Vol. 7. Springer, 2018. ISBN: 9789811073496. DOI: 10.1063/1.3051538.

[72] Kelin J. Kuhn. "Reducing variation in advanced logic technologies: Approaches to process and design for manufacturability of nanoscale CMOS". In: *Technical Digest - International Electron Devices Meeting, IEDM* (2007), pp. 471–474. ISSN: 01631918. DOI: 10.1109/IEDM.2007.4418976.

[73] H. Lad Kirankumar, S. Rekha, and Tonse Laxminidhi. "A Dead-Zone-Free Zero Blind-Zone High-Speed Phase Frequency Detector for Charge-Pump PLL". In: *Circuits, Systems, and Signal Processing* 39.8 (2020), pp. 3819–3832. ISSN: 15315878. DOI: 10.1007/s00034-020-01366-1.

[74] P. Lecoq. "Pushing the limits in Time-Of-Flight PET imaging". In: *IEEE Transactions on Radiation and Plasma Medical Sciences* 1.6 (2017), pp. 473–485. DOI: 10.1109/TRPMS.2017.2756674.

[75] David C Lee. "Analysis of jitter in phase-locked loops". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 49.11 (2002), pp. 704–711.

[76] Bing Li, Masanori Hashimoto, and Ulf Schlichtmann. "From process variations to reliability: A survey of timing of digital circuits in the nanometer era". In: *IPSJ Transactions on System LSI Design Methodology* 11.March (2018), pp. 2–15. ISSN: 18826687. DOI: 10.2197/ipsjtsldm.11.2.

[77] X. Llopart et al. "Study of low power front-ends for hybrid pixel detectors with sub-ns time tagging". In: *Journal of Instrumentation* 14.1 (2019). ISSN: 17480221. DOI: 10.1088/1748-0221/14/01/C01024.

[78] Gerald Lopez et al. "The impact of size effects and copper interconnect process variations on the maximum critical path delay of single and multi-core microprocessors". In: *Proceedings of the IEEE 2007 International Interconnect Technology Conference - Digest of Technical Papers* 404 (2007), pp. 40–42. DOI: 10.1109/iitc.2007.382346.

[79] Dominik Lorenz. "Aging Analysis of Digital Integrated Circuits". PhD thesis. 2012, pp. 1–150.

[80] Shingo Mandai and Edoardo Charbon. "A 4 x 4 x 416 digital SiPM array with 192 TDCs for multiple high-resolution timestamp acquisition". In: *Journal of Instrumentation* 8.05 (2013), P05024.

[81] John G. Maneatis. "Low-jitter process-independent DLL and PLL based on self-biased techniques". In: *IEEE journal of solid-state circuits* 31.11 (1996), pp. 1723–1732. DOI: 10.1109/9780470545492.ch48.

[82] Dejan Markovic, Borivoje Nikolic, and Robert W Brodersen. "Analysis and Design of Low-Energy Flip-Flops". In: *ISLPED'01: Proceedings of the 2001 International Symposium on Low Power Electronics and Design* (2001), pp. 15–18. DOI: 10.1109/LPE.2001.945371.

[83] Madhavan Swaminathan Martin Saint-laurent. "A Multi-PLL Clock Distribution Architecture for Gigascale Integration". In: *VLSI, 2001. Proceedings. IEEE Computer Society Workshop on* (2001). DOI: 10.1109/IWV.2001.923136.

[84] M. Mcgill. "LIDAR remote sensing". In: *Encyclopedia of Optical and Photonic Engineering*. Ed. by C Hoffman and R Driggers. 2nd editio. CRC Press, 2015. Chap. 11. ISBN: 9781351247184. DOI: https://doi.org/10.1081/E-EOE2.

[85] John A McNeill and David Ricketts. *The designer's guide to jitter in ring oscillators*. Ed. by K. Kundert. Springer Science & Business Media, 2009. ISBN: 978-0-387-76526-6. DOI: 10.1007/978-0-387-76528-0e-ISBN.

[86] Martin Miller and Michael Schnecker. "Quantifying crosstalk induced jitter in multi-lane serial data systems". In: *Designcon 2009*. Vol. 2. 2009, pp. 782–803. ISBN: 9781615670499.

[87] Tarun Mittal and Cheng-Kok Kok Koh. "Cross link insertion for improving tolerance to variations in clock network synthesis". In: *Proceedings of the 2011 international symposium on Physical design*. ACM. 2011, pp. 29–36. ISBN: 9781450307116. DOI: 10.1145/1960397.1960407.

[88] H Mizuno and K Ishibashi. "A noise-immune GHz-clock distribution scheme using synchronous distributed oscillators". In: *1998 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, ISSCC. First Edition (Cat. No. 98CH36156)*. IEEE. 1998, pp. 404–405.

[89] Rui L Aguiar Mónica Figueiredo, Mónica J. Figueiredo, and Rui L. Aguiar. "Noise and Jitter in CMOS Digitally Controlled Delay Lines". In: *13th IEEE International Conference on Electronics, Circuits and Systems* (2006), pp. 1356–1359. DOI: 10.1109/ICECS.2006.379754.

[90] Anthony V. Mule et al. "Electrical and Optical Clock Distribution Networks for Gigascale Microprocessors". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 10.5 (2001), pp. 582–594. ISSN: 10638210. DOI: 10.1109/TVLSI.2002.801604.

[91] Ashok Narasimhan and Ramalingam Sridhar. "Impact of variability on clock skew in H-tree clock networks". In: *Proceedings - Eighth International Symposium on Quality Electronic Design, ISQED 2007* (2007), pp. 458–463. DOI: 10.1109/ISQED.2007.88.

[92] S. Nassif. "Delay Variability: Sources, Impacts and Trends". In: *Proceedings of the 2000 IEEE International Solid-State Circuits Conference*. IEEE, 2000. ISBN: 0780358538.

[93] Borivoje Nikolić et al. "Improved sense-amplifier-based flip-flop: design and measurements". In: *IEEE Journal of Solid-State Circuits* 35.6 (2000), pp. 876–884. ISSN: 00189200. DOI: 10.1109/4.845191.

[94] S. K. Nithin, Gowrysankar Shanmugam, and Sreeram Chandrasekar. "Dynamic voltage (IR) drop analysis and design closure: Issues and challenges". In: *Proceedings of the 11th International Symposium on Quality Electronic Design, ISQED 2010* (2010), pp. 611–617. DOI: 10.1109/ISQED.2010.5450515.

[95] V. Oklobdzjja, V. Stojanovic, and D. Markovic. *Digital System Clocking*. Hoboken, New Jersey: John Wiley & Sons, Inc., 2003. ISBN: 047127447X.

[96] Takaaki Okumura and Masanori Hashimoto. "Setup time, hold time and clock-to-q delay computation under dynamic supply noise". In: *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences* E94-A.10 (2011), pp. 1948–1953. ISSN: 17451337. DOI: 10.1587/transfun.E94.A.1948.

[97] Frank O. O'Mahony et al. "A 10-GHz global clock distribution using coupled standing-wave oscillators". In: *IEEE Journal of Solid-State Circuits* 38.11 (2003), pp. 1813–1820. ISSN: 00189200. DOI: 10.1109/JSSC.2003.818299.

[98] P. Otfinowski et al. "Pattern Recognition algorithm for charge sharing compensation in single photon counting pixel detectors". In: *Journal of Instrumentation* 14.1 (2019). ISSN: 17480221. DOI: 10.1088/1748-0221/14/01/C01017.

[99] Hamid Partovi et al. "Flow-Through Latch and Edge-Triggered Flip-flop Hybrid Elements". In: *IEEE International Solid-State Circuits Conference 1996* (1996), pp. 138–139. DOI: 10.1109/ISSCC.1996.488543.

[100] Muhammad Touqir Pasha, Yasir Ali Shah, and Jacob Wikner. "A wide range all-digital delay locked loop for video applications". In: *2015 European Conference on Circuit Theory and Design, ECCTD 2015* Dll (2015). DOI: 10.1109/ECCTD.2015.7300049.

[101] G. Prekas et al. "Direct and indirect detectors for X-ray photon counting systems". In: *IEEE Nuclear Science Symposium Conference Record* (2011), pp. 1487–1493. ISSN: 10957863. DOI: 10.1109/NSSMIC.2011.6154354.

[102] Kun Qian. "Variability Modeling and Statistical Parameter Extraction for CMOS Devices". PhD thesis. Electrical Engineering and Computer Sciences, University of California at Berkeley, 2015.

[103] J. Rabaey. *Digital integrated circuits: a design perspective*. 2nd ed. Prentice-Hall, Inc., 2002. DOI: 10.1016/b978-0-7506-1142-8.50008-2.

[104] Matthias Raffelsieper and Mohammadreza Mousavi. *Efficient Verification of Verilog Cell Libraries*. http://fmv.jku.at/hwvw10/slides/matthias-raffelsieper-hwvw10-slides-handout.pdf. Classnotes HWVW. 2010.

[105] Dong-Jun Yang Ran Li Xiaoling Guo and Kenneth K O. "Wireless Clock Distribution System Using an External Antenna". In: *IEEE Journal of Solid State Circuits* 42.10 (2007), pp. 2283–2292.

[106] L. Ratti et al. "Monolithic pixel detectors in a 0.13 μ m CMOS technology with sensor level continuous time charge amplification and shaping". In: *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 568.1 (2006), pp. 159–166. ISSN: 01689002. DOI: 10.1016/j.nima.2006.05.225.

[107] Behzad Razavi. "A Study of Injection Locking and Pulling in Oscillators". In: *IEEE Journal of Solid State Circuits* 39.9 (2004), pp. 1415–1424.

[108] P J Restle and A Deutsch. "Designing the best clock distribution network". In: *VLSI Circuits, 1998. Digest of Technical Papers. 1998 Symposium on* (1998). DOI: 10.1109/VLSIC.1998.687985.

[109] S Rosenberg and KA Meade. *A practical guide to adopting the universal verification methodology (UVM).* 2nd ed. USA: Cadence Design Systems, Inc., 2010. ISBN: 978-1-300-53593-5.

[110] Daniele Rossi et al. "Reliable Power Gating with NBTI Aging Benefits". In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 24.8 (2016), pp. 2735–2744. ISSN: 10638210. DOI: 10.1109/TVLSI.2016.2519385.

[111] Enrico Rubiola. *Phase noise and frequency stability in oscillators.* Ed. by S. Cripps. USA: Cambridge University Press, 2009. ISBN: 978-0-521-88677-2.

[112] Resve Saleh et al. "Clock skew verification in the presence of IR-drop in the power distribution network". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 19.6 (2000), pp. 635–644. ISSN: 02780070. DOI: 10.1109/43.848085.

[113] Alessandro Sassone et al. "Modeling of thermally induced skew variations in clock distribution network". In: *17th International Workshop on Thermal Investigations of ICs and Systems, THERMINIC 2011* (2011), pp. 23–32.

[114] Mauricio Altieri Scarpato. "Digital circuit performance estimation under PVT and aging effects". PhD thesis. Université Grenoble Alpes, 2017.

[115] Burkhard Schmidt. "The High-Luminosity upgrade of the LHC: Physics and Technology Challenges for the Accelerator and the Experiments". In: *Journal of Physics: Conference Series* 706.Section 2 (2016). ISSN: 17426596. DOI: 10.1088/1742-6596/706/2/022002.

[116] Yaron Semiat and Ran Ginosar. "Timing measurements of synchronization circuits". In: *Proceedings - International Symposium on Asynchronous Circuits and Systems* (2003), pp. 68–77. ISSN: 15228681. DOI: 10.1109/ASYNC.2003.1199167.

[117] W. Snoeys. "CMOS monolithic active pixel sensors for high energy physics". In: *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 765.2014 (2014), pp. 167–171. ISSN: 01689002. DOI: 10.1016/j.nima.2014.07.017.

[118] W. Snoeys et al.
"A process modification for CMOS monolithic active pixel sensors for enhanced depletion, timing performance and radiation tolerance".
In: *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 871.July (2017), pp. 90–96.
ISSN: 01689002. DOI: 10.1016/j.nima.2017.07.046.

[119] Helmuth Spieler. *Semiconductor detector systems*. 1st ed.
Oxford university press, 2005. ISBN: 0–19–852784–5.

[120] Helmuth Spieler. *Silicon Readout – Where The Bugs Can Hide*.
Excellence in Detectors and Instrumentation Technologies, CERN.
Geneva, 2011.

[121] S.M. Sze and K. Ng Kwok. *Physics of Semiconductor Devices*. 3rd.
Hoboken, New Jersey: John Wiley & Sons, Inc., 2007.
ISBN: 978-0-47 1-1 4323-9.

[122] Shima Tayyeb Ghasemi and Ali Baradaranrezaeii.
"A Novel High Speed, Low Power, and Symmetrical Phase Frequency Detector with Zero Blind Zone and π Phase Difference Detection Ability".
In: *Circuits, Systems, and Signal Processing* 39.6 (2020), pp. 2880–2899.
ISSN: 15315878. DOI: 10.1007/s00034-019-01312-w.

[123] Stefan Tertinek and Christian Vogel. "Reconstruction of nonuniformly sampled bandlimited signals using a differentiator-multiplier cascade".
In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 55.8 (2008), pp. 2273–2286. ISSN: 10577122. DOI: 10.1109/TCSI.2008.918267.

[124] *The 10 ps challenge: A step towards reconstruction-less TOF-PET*.
https://the10ps-challenge.org.

[125] Jai Narayan Tripathi et al. "A Review on Power Supply Induced Jitter".
In: *IEEE Transactions on Components, Packaging, and Manufacturing Technology* 9.3 (2018), pp. 511–524. ISSN: 21563950.
DOI: DOI:10.1109/TCPMT.2018.2872608.

[126] Christopher G. Tully. *Fast timing for collider detectors*.
https://indico.cern.ch/event/633341/. CERN Academic Training Lectures.
Geneva, 2017. DOI: 10.1142/S0217751X1644022X.

[127] R. Turchetta. "CMOS Monolithic Active Pixel Sensors (MAPS) for future vertex detectors... but not just".
In: *International Symposium on the Development of Detectors for Particle, Astro-Particle and Synchrotron Radiation Experiments, SLAC*. April. USA, 2006.

[128] R. Turchetta et al.
"CMOS Monolithic Active Pixel Sensors (MAPS): New 'eyes' for science".
In: *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 560.1 (2006), pp. 139–142.
ISSN: 01689002. DOI: 10.1016/j.nima.2005.11.241.

[129] R. Turchetta et al. "Monolithic Active Pixel Sensor for charged particle tracking and imaging using standard VLSI CMOS technology".
In: *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 458.3 (2001), pp. 677–689.
ISSN: 01689002. DOI: 10.1016/S0168-9002(00)00893-7.

[130] Graham Upton and Ian Cook. *A Dictionary of Statistics*. 2nd ed. Oxford University Press, 2008. ISBN: 9780199541454. DOI: 10.1093/acref/9780199541454.001.0001.

[131] Remco C.H. H Van de Beek et al. "Low-jitter clock multiplication: A comparison between PLLs and DLLs". In: *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing* 49.8 (2002), pp. 555–566. ISSN: 10577130. DOI: 10.1109/TCSII.2002.806248.

[132] Vivek Venugopal and Suresh Kannan. "Accelerating real-time LiDAR data processing using GPUs". In: *Midwest Symposium on Circuits and Systems* March (2013), pp. 1168–1171. ISSN: 15483746. DOI: 10.1109/MWSCAS.2013.6674861.

[133] Zdenek Vykydal et al. "The RELAXd project: Development of four-side tilable photon-counting imagers". In: *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 591.1 (2008), pp. 241–244. ISSN: 01689002. DOI: 10.1016/j.nima.2008.03.099.

[134] Yuan Wang et al. "Delay-locked loop based frequency quadrupler with wide operating range and fast locking characteristics". In: *Proceedings - IEEE International Symposium on Circuits and Systems* 2016-July.Dcdl (2016), pp. 1–4. ISSN: 02714310. DOI: 10.1109/ISCAS.2016.7527155.

[135] Norbert Wermes. "Trends in pixel detectors: Tracking and imaging". In: *IEEE Transactions on Nuclear Science* 51.3 III (2004), pp. 1006–1015. ISSN: 00189499. DOI: 10.1109/TNS.2004.829438. arXiv: 0401030 [physics].

[136] Thucydides Xanthopoulos. *Clocking in Modern VLSI Systems*. Ed. by Thucydides Xanthopoulos. Integrated Circuits and Systems. Boston, MA: Springer US, 2009. ISBN: 978-1-4419-0260-3. DOI: 10.1007/978-1-4419-0261-0.

[137] Augusto Ronchini Ximenes, Preethi Padmanabhan, and Edoardo Charbon. "Mutually coupled time-to-digital converters (TDCs) for direct time-of-flight (dTOF) image sensors". In: *Sensors* 18.10 (2018), p. 3413. ISSN: 14248220. DOI: 10.3390/s18103413.

[138] Hu Xu, Vasilis F. Pavlidis, and Giovanni De Micheli. "Effect of process variations in 3D global clock distribution networks". In: *ACM Journal on Emerging Technologies in Computing Systems* 8.3 (2012). ISSN: 15504832. DOI: 10.1145/2287696.2287703.

[139] Rong Jyi Yang and Shen Iuan Liu. "A 2.5 GHz all-digital delay-locked loop in 0.13 μm CMOS technology". In: *IEEE Journal of Solid-State Circuits* 42.11 (2007), pp. 2338–2347. ISSN: 00189200. DOI: 10.1109/JSSC.2007.906183.

[140] S Tam ; S Rusu ; U Nagarji Desai ; R Kim ; Ji Zhang ; I Young. "Clock Generation and Distribution for the First IA-64 Microprocessor". In: *IEEE Journal of Solid-State Circuits* 35.11 (2000). DOI: 10.1109/4.881198.

[141] Payman Zarkesh-Ha, Tony Mule, and James D. Meindl. "Characterization and modeling of clock skew with process variations". In: *Proceedings of the Custom Integrated Circuits Conference*. 1999, pp. 441–444. ISBN: 0780354443. DOI: 10.1109/cicc.1999.777319.

[142] L. Zaworski et al. “Quantization error in Time-to-Digital converters”. In: *Architecture* XIX.1 (2012), pp. 115–122. ISSN: 0860-8229.

[143] Min Zhao et al. “Worst case clock skew under power supply variations”. In: January (2002), p. 22. DOI: 10.1145/589411.589416.

[144] L. R. Zheng and H. Tenhunen. “Design and analysis of power integrity in deep submicron system-on-chip circuits”. In: *Analog Integrated Circuits and Signal Processing* 30.1 (2002), pp. 15–29. ISSN: 09251030. DOI: 10.1023/A:1012444619307.