

Факултет инжењерских наука
Универзитета у Крагујевцу

Ђорђе С. Гачић

Пројектовање синтетизабилне
светодигиталне фреквенцијски
затворене петље са широким опсегом
подешавања до учестаности од
640 MHz

Мастер рад

Крагујевац, 2024.

Факултет инжењерских наука Универзитета у Крагујевцу



Назив студијског програма: Електротехника и рачунарство

Ниво студија: Мастер академске студије

Предмет: Анализа и пројектовање алгоритама

Број индекса: 408/2022

Ђорђе С. Гачић

Пројектовање синтетизабилне
сведигиталне фреквенцијски
затворене петље са широким опсегом
подешавања до учестаности од
640 MHz

Мастер рад

Комисија за преглед и одбрану:

- проф. др Владимир М. Миловановић -
ментор
- доц. др Иван Крстић
- проф. др Милан Матијевић

Датум одбране: _____

Оцена: _____

У оквиру овог мастер рада кандидат треба да реализује дигиталну фреквенцијски затворену петљу у стандардној 130 nm CMOS технологији, за циљану учестаност рада до 640 MHz. Систем треба да се састоји од дигитално контролисаног осцилатора у облику матрице тро-сттичких инвертора коме претходи управљачка логика која омогућава брзо достизање и стабилно одржавање жељене радне учестаности која је задата целобројним умношком у односу на референтну учестаност од 16 MHz. Приказати резултате симулација у најгорим, типичним и најбољим условима рада узимајући у обзир напон напајања, температуру и процесни угао. Потребно је теоријски обрадити структуру система као и појаве које се запажају у резултатима симулација. У сврху додатне провере исправности рада система, потребно је направити модел система у програмском језику Пајтон који треба веродостојно симулирати рад свих компоненти система и упоредити пар различитих регулатора укључујући и стандардни ПИД (пропорционално, интегрални, диференцијални) регулатор.

Препоручена литература:

- [1] Robert Bogdan Staszewski и Poras T. Balsara, *All-Digital Frequency Synthesizer in Deep-Submicron CMOS*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.
- [2] Behzad Razavi, *Design of CMOS Phase-Locked Loops: From Circuit Level to Architecture Level*, Cambridge University Press, 2020.
- [3] Jan M. Rabaey, Anantha P. Chandrakasan и Borivoje Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2. издање, Prentice Hall electronics and VLSI series, Pearson Education, 2003., ISBN: 9780131207646.
- [4] Anant Agarwal и Jeffrey H. Lang, *Foundations of Analog and Digital Electronic Circuits*, ISSN, Elsevier Science, 2005., ISBN: 9780080506814.

Крагујевац, 25. 9. 2024.

Ментор:

Др Владимира М. Миловановић,
ванредни професор

ПРИЛОГ 6: Пријава мастер рада



Универзитет у Крагујевцу
Факултет инжењерских наука
Универзитета у Крагујевцу



Мастер академске студије: **Електротехника и рачунарство**

Назив предмета: **Анализа и пројектовање алгоритама**

Име и презиме: **Ђорђе Гачић**

Број индекса: **408/2022**

ПРИЈАВА МАСТЕР РАДА

**Тема рада: Пројектовање синтетизабилне сведигиталне фреквенцијски затворене петље
са широким опсегом подешавања до учестаности од 640MHz**

Задатак: Потребно је реализовати дигиталну фреквенцијски затворену петљу у стандардној 130nm CMOS технологији, за циљану учестаност рада до 640MHz. Систем треба да се састоји од дигитално контролисаног осцилатора у облику матрице тростатичких инвертора коме претходи управљачка логика која омогућава брзо достизање и стабилно одржавање жељене радне учестаности која је задата целобројним умношком у односу на референтну фреквенцију од 16MHz. Приказати резултате симулација у најгорим, типичним и најбољим условима рада узимајући у обзир напон напајања, температуру и процесни угао. Потребно је теоријски обрадити структуру система као и појаве које се запажају у резултатима симулација. У сврху додатне провере исправности рада система, потребно је направити модел система у програмском језику Пајтон који треба веродостојно симулирати рад свих компоненти система и упоређује пар различитих контролера укључујући и стандардни ПИД (пропорционално, интегрални, диференцијални) контролер.

Крагујевац, 20. јун 2024. године

Ментор:


Др Владомир Миловановић, ванр. проф.

Садржај

1 Увод	3
2 Опис и структура фреквенцијски затворене петље	4
3 Управљачка логика дигиталне фреквенцијски затворене петље	6
3.1 Управљачка предобрађа	6
3.1.1 Синхронизатор	7
3.2 Двостепени регулатор	8
3.3 ПИД регулатор	9
3.4 Управљачки декодер	10
4 Теоријско разматрање осцилатора	13
4.1 Основни принципи осциловања	13
4.2 Систем осцилаторне повратне спреге	14
4.3 Основе прстенастог осцилатора	16
4.3.1 Инверторски прстен	21
4.4 Шум у осцилаторима	23
5 Дигитално контролисани осцилатор	25
5.1 Архитектура предложеног DCO-а	26
5.2 Блок дијаграм ћелије DCO-а	27
5.2.1 Ћелија DCO-а са управљачким сигналима активним на низак логички ниво	28
5.3 Претварачи напонског нивоа DCO-а	30
6 Имплементација и резултати симулација	33
6.1 SystemVerilog изворни код	33
6.2 Лејаут	41
6.3 Симулација рада фреквенцијски затворене петље	43
6.4 PVT зависност дигитално контролисаног осцилатора	43
6.5 Анализа потрошње система	46
6.6 Временски одзив дигитално контролисаног осцилатора	47
6.7 Спектар снаге фреквенцијски затворене петље	48
6.8 Фазни шум дигитално контролисаног осцилатора	49
7 Пајтон модел FLL-а	50
7.1 Пајтон код	50
7.2 Поређење рада различитих управљачких режима	54
7.3 Утицај константи ПИД регулатора на одзив система	56
7.4 Утицај резолуције DCO-а на одзив система	57
8 Поређење DCO-а пројектованих у 130 nm и 180 nm технологији	58
8.1 Начин и утицај коришћења дубоке јаме N типа у аналогном пројектовању DCO-а	60
9 Закључак	62
Литература	63
Објављени радови	64

Abstract

Frequency-locked loop (FLL) represents a viable way of generating a range of frequencies from a single reference frequency by using a negative feedback electronic control system that compares the frequency of a controlled oscillator to the reference one. A digital synthesizable FLL is designed in 130 nm CMOS technology for a target frequency of up to 640 MHz. It employs a wide-tuning range digitally controlled oscillator (DCO) assembled from tri-state inverters in the form of a matrix. The FLL can optionally use a bang-bang or a soft-programmable standard proportional-integral-derivative (PID) controller to regulate the feedback loop. Its design practically minimizes metastability occurrence. The proposed digital FLL occupies $100 \mu\text{m} \times 330 \mu\text{m}$ of chip area and consumes 3.5 mW in typical operating conditions. The reference clock is 16 MHz, and the output oscillation frequency is set to 640 MHz, while the achieved frequency resolution is 2.8 MHz. The system is additionally modeled in software using the Python programming language. A comparison of the characteristics of DCOs designed in 130 nm and 180 nm TSMC technologies is also provided.

Keywords: Frequency-locked loop, digitally controlled oscillator, clock generator, synthesizable, CMOS technology, PID controller, metastability, Python model.

Резиме

Фреквенцијски затворена петља (енгл. *Frequency-Locked Loop*, FLL) представља одржив начин генерирања опсега учестаности из једне референтне учестаности коришћењем електронског система управљања са негативном повратном спрегом, који пореди учестаност контролисаног осцилатора са поменутом референтном учестаношћу. Дигитални синтетизабилни FLL је пројектован у 130 nm технологији за циљану учестаност до 640 MHz. Он садржи дигитално контролисани осцилатор (енгл. *Digitally Controlled Oscillator*, DCO) са широким подешавањем опсега који се састоји од тростатичких инвертора у облику матрице. FLL може произвољно користити двостепени регулатор или софтверски програмабилни стандардни пропорционално, интегрални, диференцијални (енгл. *Proportional-Integral-Derivative*, PID) регулатор за управљање повратном спрегом. Такав дизајн у пракси минимизује појаву метастабилности. Предложени дигитални FLL заузима $100 \mu\text{m} \times 330 \mu\text{m}$ простора на чипу и троши 3,5 mW у типичним условима рада. Референтни такт је 16 MHz, а излазна учестаност осциловања је подешена на 640 MHz, док постигнута резолуција учестаности износи 2,8 MHz. Систем је додатно софтверски моделиран у Пајтон програмском језику. Такође је дато поређење карактеристика DCO-а пројектованих у 130 nm и 180 nm TSMC технологијама.

Кључне ријечи: Фреквенцијски затворена петља, дигитално контролисани осцилатор, генератор такта, синтетизабилност, CMOS технологија, ПИД регулатор, метастабилност, Пајтон модел.

1 Увод

У данашње вријеме, фазно затворена петља (енгл. *Phase-Locked Loop*, PLL) и петља са затвореним кашњењем (енгл. *Delay-Locked Loop*, DLL) представљају свеприсутне блокове у пројектовању интегрисаних кола. Безброј примјена самих чипова захтијевају или генератор такта или синтезатор учестаности, што подразумијева уградњу неког од поменутих блокова унутар система који се пројектује. Главна улога таквог блока у дизајну је да генерише стабилан и прецизан излазни сигнал чија је фаза подесива у односу на фазу улазног сигнала, самим тим одржавајући везу између улазне и излазне учестаности. Међутим, чак и веома сложени системи често захтијевају генератор такта, који само множи улазну учестаност без да посебно води рачуна о фази такта или апсолутном ћитеру (енгл. *jitter*). У таквим примјенама, потребна и довољна је само фреквенцијски затворена петља (енгл. *Frequency-Locked Loop*, FLL) да би се испунили тражени захтјеви.

Постоје многи начини имплементације FLL-а [1]. Штавише, FLL као интегрисано коло може се сврстати у две групе: аналогни и дигитални FLL. Иако је очигледан недостатак првог максимална учестаност и њена резолуција, они посједују многе друге предности наспрам њихових аналогних парњака. Заузимају мање простора, истичу се већом отпорношћу на промјене процесних углова, напона и температуре (енгл. *Process-Voltage-Temperature*, PVT), лако су преносиви у различите технологије, и стога омогућавају поновну употребу, скалирање, једноставнију методологију, као и брже циклусе пројектовања. Узимајући у обзир све претходно поменуто, испоставља се да је у општем случају боље ићи ка развоју дигиталног FLL-а кад год спецификација архитектуре система то дозвољава.

Овај рад конкретно предлаже синтетизабилни дигитални FLL сличан предложеном у литератури [2], са побољшаном брзином закључавања FLL-а [3] и смањеним ризиком од метастабилности. Осцилатор је састављен од тростатичких инвертора и заснован на прстенастом DCO-у из литературе [4], измијењен додавањем независног напона напајања DCO-а са претварачима напонских нивоа (енгл. *level shifter*) и употребом петостепене [5] уместо тростепене толологије прстена осцилатора.

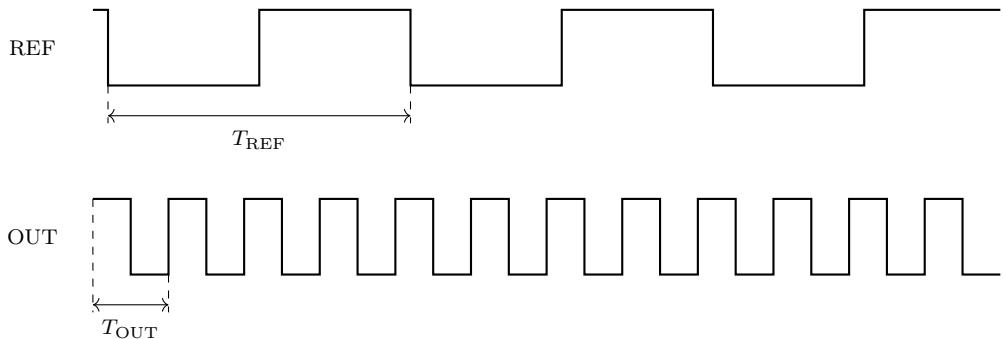
У поглављу 2 се даје основни опис рада и структуре предложеног дигиталног FLL-а. Управљачка логика FLL-а је детаљно описана у поглављу 3. Теоријска разматрања осцилатора, као главне компоненте у систему дата су у поглављу 4, док је коришћени дигитално контролисан осцилатор описан у поглављу 5. Поглавље 6 садржи анализиран код на нивоу регистра (енгл. *Register-Transfer Level*, RTL) и лејаут FLL-а и његових појединачних компоненти, а дат је и увид у имплементацију и добијене резултате симулација, такође уз теоријска разматрања појава које су од значаја за рад читавог система. У поглављу 7 описан је значај софтверског моделирања система и приказана је имплементација модела понашања FLL-а у Пјтон програмском језику. У сврху поређења карактеристика осцилатора у различitim технологијама, рад садржи поглавље 8. Коначно, поглавље 9 закључује рад и наговјештава могућности даљег рада на побољшању и проширењу система.

2 Опис и структура фреквенцијски затворене петље

По дефиницији, FLL је управљачки систем са негативном повратном спрегом који за-
кључава учестаност излазног сигнала на предвиђену циљану учестаност. У принципу,
непрестано управља учестаношћу осцилатора на аутоматски начин све док излазна
учестаност не достигне циљану вриједност, након чега се та вриједност учестаности
одржава на излазу чак и у присуству поремећаја. То значи да FLL спада у синтетизаторе
учестаности који се уопштено описују као активна електронска кола која имају
 неки сигнал са референтном и веома стабилном учестаношћу (f_{REF}) на улазу и затим
генеришу излазни сигнал са учестаношћу (f_{OUT}) која представља референтну учес-
таност помножену са одговарајућим чиниоцем тј. умношком учестаности (FMUL)
као у:

$$f_{\text{OUT}} = \text{FMUL} \cdot f_{\text{REF}} \quad (1)$$

Треба напоменути да је стабилност, прецизност и спектрална чистоћа излазног сиг-
нала у константној вези са квалитетом сигнала на улазу [6]. Слика 1 приказује појед-
ностављен дигитални сигнал на улазу (REF) и излазу (OUT) система са одређеном
вриједношћу умношке учестаности.



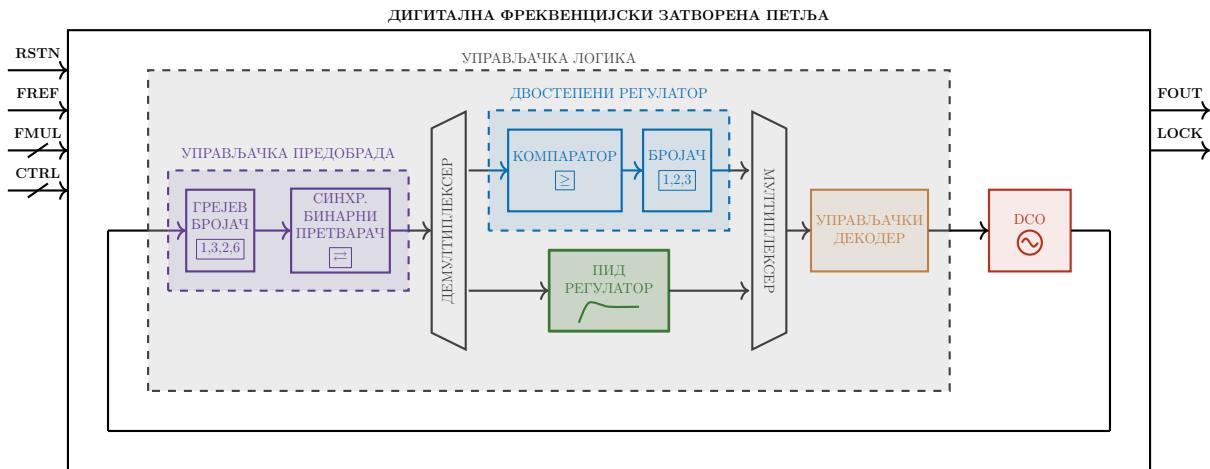
Слика 1: Демонстрација рада синтетизатора учестаности за $\text{FMUL}=4$.

$$f_{\text{REF}} = \frac{1}{T_{\text{REF}}} \quad (2)$$

$$f_{\text{OUT}} = \frac{1}{T_{\text{OUT}}} \quad (3)$$

Дефиниција изнад не предвиђа облик синтетисаног излазног сигнала. Он може бити
периодични или правоугаони (дигитални) сигнал, који нам је у CMOS технологи-
ји дефинитивно кориснији за разматрање, али за високе учестаности се своди на
синусоиду.

У овом раду описана је релативно једноставна али ефикасна дигитална фреквенциј-
ски затворена петља, чија је системска архитектура на нивоу блокова приказана на
слици 2. Описани FLL се практично састоји из два блока: дигитално контролисаног
осцилатора (енгл. *Digitally Controlled Oscillator*, DCO) и блока управљачке логике,
који генерише улазне сигнале осцилатора на основу тренутне учестаности DCO-а. У
сврху поједностављења, са слике су изостављени неки конфигурациони улази FLL-а,



Слика 2: Блок дијаграм дигиталне фреквенцијски затворене петље састављене од: блока управљачке логике (лијево) и дигитално контролисаног осцилатора (десно).

као што су коефицијенти ПИД регулатора или улаз за одабир режима рада, чија улога је детаљније описана у поглављу 6.

Основни улази у систем су:

- RSTN - Ресет сигнал који је активан на низак логички ниво. Разлози за активност на низак логички ниво су мања подложност шуму у односу на висок логички ниво, осигурување активности ресет сигнала док се систем не стабилизује (јер је већина осталих сигнала иницијално на ниском напонском нивоу) и већа енергетска ефикасност, јер CMOS кола троше мање енергије када се држе на стабилном ниском логичком нивоу.
- FREF - Референтни такт.
- FMUL - Улазни умножак учестаности (множењем са учестаношћу референтног такта добија се вриједност учестаности коју треба постићи и одржавати на излазу).
- CTRL - Сигнал који одређује који од два управљачка режима се користи.

Излази из система су:

- FOUT - Излазни такт.
- LOCK - Сигнал који говори да ли је систем ушао у стабилно стање тј. да ли ради на жељеној учестаности.

3 Управљачка логика дигиталне фреквенцијски затворене петље

Управљачка логика FLL-а састоји се од двије независне процесне гране, које представљају два међусобно искључива режима управљања FLL-а. Оба режима на улазу примају бинарну цјелобројну вриједност која представља број периода такта DCO-а унутар периода референтног такта. Такође, оба управљачка режима генеришу бинарну вриједност на излазу, која представља управљачку бинарну ријеч осцилатора директно пропорционалну учестаности излазног сигнала. Главне разлике између два поменута режима су брзина закључавања (затварања) FLL-а и једноставност подешавања. Циљ управљачке логике FLL-а је изједначити вриједност улазног множача учестаности са бројем периода такта DCO-а унутар периода референтног такта што је брже и прецизније могуће, чиме се долази до постизања жељене учестаности на излазу DCO-а. Поред самог постизања жељене учестаности, задатак управљачке логике је и њено одржавање током времена јер, због утицаја разних фактора, као што је нпр. температура, константно долази до нежељеног смањења или повећања вриједности учестаности, односно поремећаја. То значи да управљачка логика константно прати рад система и по потреби реагује на исход непожељних утицаја на излазну учестаност. Комплетна управљачка логика FLL-а је подијељена на неколико фаза, које су детаљно описане у наредним поглављима, а то су:

- Управљачка предобрађа
- Двостепени регулатор / ПИД регулатор
- Управљачки декодер

3.1 Управљачка предобрађа

Фаза управљачке предобрађе (енгл. *control preprocessing*) укључује неколико блокова чија је улога претворити информацију о учестаности излазног такта DCO-а у бинарну вриједност која ће бити прослијеђена као улаз наредној фази управљачке логике FLL-а. Као прво, да би се одредила брзина осциловања DCO-а, потребан је бројач. У имплементацији описаној у овом раду коришћен је Грејев бројач. У Грејевом коду свака узастопна вриједност се разликује за само један бит. То му даје предност у односу на природни бинарни бројач из разлога што значајно умањује метастабилност бројача изазвану узорковањем (енгл. *sampling*) вриједности бројача. Табела 1 приказује првих осам бројева у Грејевом коду, у поређењу са природним бинарним цифрама.

Као примјер појаве метастабилности може се навести проблем који се може јавити у употреби природних бинарних кодова, а то је да се, при преласку у наредно стање бројача, стање свих битова који се мијењају не мора промијенити тачно синхронизовано. Речимо, при преласку из стања 3 у стање 4 (бинарно 011 у 100), код природног бинарног бројача сви битови мијењају стање, тако да та транзиција може да иде редом $011 \rightarrow 001 \rightarrow 101 \rightarrow 100$, што значи да могу да постоје два прелазна стања, која могу бити прочитана чиме би коначно стање бројача било погрешно. У овом конкретном случају, умјесто у стање 4 (бинарно 100), бројач би завршио у стању 1 (бинарно 001) или 5 (бинарно 101), за шта не постоји могућност при употреби Грејевог кода јер се у њему из стања 011 прелази у стање 010, при чему се мијења само

Табела 1: Првих осам бројева у Грејевом коду

Децимални запис	Бинарни запис	Грејев код	Грејев код у децималном запису
0	000	000	0
1	001	001	1
2	010	011	3
3	011	010	2
4	100	110	6
5	101	111	7
6	110	101	5
7	111	100	4

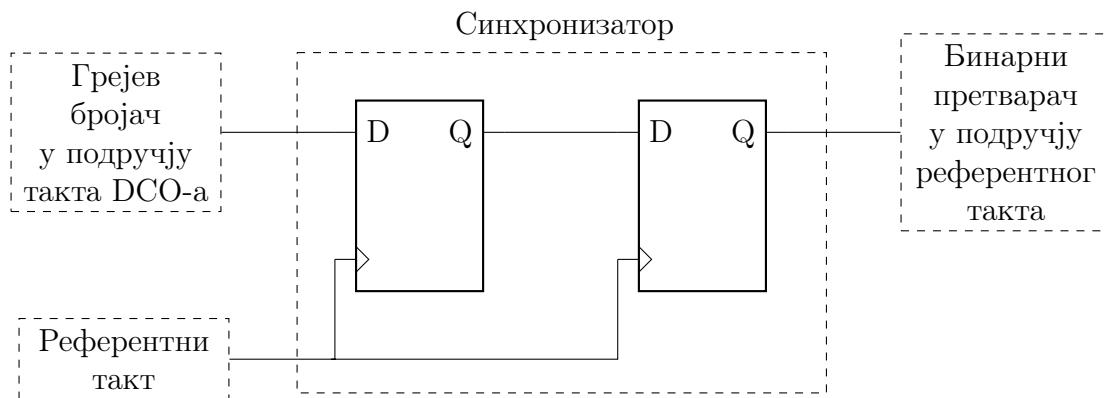
један бит. Треба напоменути да могућност појаве метастабилности и даље постоји јер је то много оштија појава иза проблема погрешног узорковања у случају када се у исто вријеме мијењају такт и подаци, али Грејев код омогућује да се потенцијална грешка сведе на максимално један бит.

3.1.1 Синхронизатор

Да би се значајно више смањио ризик од метастабилности, сваки бит са излаза Грејевог бројача се пропушта кроз синхронизатор са два флип-флопа да би безбједно прешао у подручје референтног такта. Затим, синхронизована вриједност Грејевог бројача се претвара у природан бинарни формат и узоркује се за даљу обраду.

Метастабилност се јавља због сигнала који не подразумјевају стабилна стања високог или ниског логичког нивоа у неком тренутку у току рада. Ако постоји више тактова у дизајну, метастабилност се не може избећи, али се штетни ефекти метастабилности могу неутралисати. Један врло ефикасан начин неутралисања јесте синхронизација сигнала коришћењем два флип-флопа.

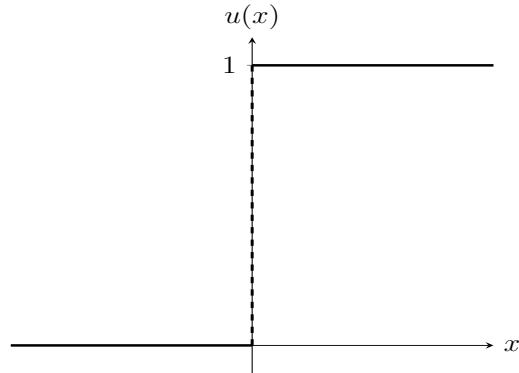
У нашем конкретном случају постоји подручје такта дигитално контролисаног осцилатора и подручје референтног такта. У дизајну постоји одређена логика (Грејев бројач) која ради у подручју такта DCO-а, и потребно је излазне сигнале из те логичке синхронизовати са референтним тактом без ризика од штетног утицаја метастабилности. Начин имплементације синхронизатора помоћу кога се то може постићи је приказан на слици 3.



Слика 3: Синхронизатор са два флип-флопа.

3.2 Двостепени регулатор

У теорији управљања, двостепени регулатор (двофазни, bang-bang или on-off регулатор) је регулатор повратне спреге које се нагло пребацује између два стања. Математички модел двостепеног регулатора може се представити преко јединичне одскочне функције (Хевисајдове функције) која се иначе користи у теорији система управљања и обради сигнала да би се представио сигнал који мијења стање у одређено вријеме. Она има вриједност 0 за негативне вриједности аргумента и 1 за позитивне вриједности аргумента:



Слика 4: График јединичне одскочне функције.

$$u(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (4)$$

Главне предности примјене двостепеног регулатора су једноставна имплементација и брз одговор на промјене, док су недостаци могуће осцилације око жељене вриједности након што се она постигне, као и непогодност за системе где је потребно веома прецизно управљање и брзо закључавање односно долазак до стационарног стања.

Као што је већ наведено, FLL из овог рада има два међусобно искључива управљачка блока. Један управљачки блок FLL-а је веома сличан двостепеном регулатору описаном изнад. Он пореди улазни умножак учестаности са узоркованим вриједношћу бројача и одлучује да ли инкрементирати, декрементирати или онемогућити предстојећи блок тј. двосмјерни бројач (енгл. *up-down counter*). Ако је улазни умножак учестаности већи од вриједности бројача узорковане у блоку управљачке предобрађе, тада се двосмјерни бројач инкрементира, ако је мањи тада се декрементира, а у случају да је једнак, двосмјерни бројач задржава претходно стање. То осигурава постепено управљање и закључавање све до постизања жељене учестаности тј. преласка у стабилно стање. Како би се спријечиле (или барем прориједиле) могуће осцилације око жељене вриједности, у овом блоку је доведен додатни сигнал који омогућава да се, након што систем једном уђе у стабилно стање, бројач двостепеног регулатора не инкрементира или декрементира баш након сваког одступања од жељене вриједности јер се очекује да у стабилном стању та одступања буду незнатна. Тада додатни сигнал се користи и у ПИД регулатору који представља други управљачки режим о коме ће више ријечи бити у наредном поглављу.

3.3 ПИД регулатор

У другом управљачком режиму, управљачка бинарна ријеч за DCO се генерише подесивим ПИД регулатором. ПИД регулатор је далеко најчешће коришћен алгоритам за управљање у инжењерству. Највећим бројем повратних спрега се управља преко ПИД регулатора или његових дјелимичних варијација [7]. ПИД регулатор је управљачки механизам заснован на повратној спрези, који ради тако што непрекидно исправља и скалира сигнал грешке. Грешка у овом случају представља разлику између измјерене вриједности у управљачкој предобради (узоркована вриједност бројача) и жељене референтне задате вриједности (вриједност улазног умношког учестаности). Исправљање и скалирање се распоређује у три компоненте које су имплементиране помоћу подесивих коефицијената, цијелих бројева у бинарном формату са фиксном тачком, доведених из регистарске банке:

- Пропорционална (P) - управља брзином одзива управљачког система непосредно множећи сигнал грешке константним чиниоцем који се назива и константа пропорционалног појачања. Ако је она превелика, систем може постати нестабилан. Насупрот томе, њена премала вриједност доводи до малог излазног одзива на велику улазну грешку чиме регулатор постаје мање осјетљив.
- Интегрална (I) - користи се за смањење грешке стационарног стања (енгл. *steady state*) скалирањем грешке константним чиниоцем и сумирањем резултата током времена. Грешка стационарног стања је разлика између жељене и стварне вриједности коначног излаза [8]. Интегрално дејство убрзава кретање излаза ка жељеној вриједности.
- Диференцијална (D) - пропорционална брзини промјене грешке, и њен циљ је ограничити излаз да би се смањила могућа прекорачења или осцилације узроковане P и I компонентама, без смањења брзине регулатора. Како је у овом систему референтна задата вриједност константна и нема брзих промјена на улазу које могу изазвати такав исход, P и I компоненте су довољне за гладак и стабилан одзив система.

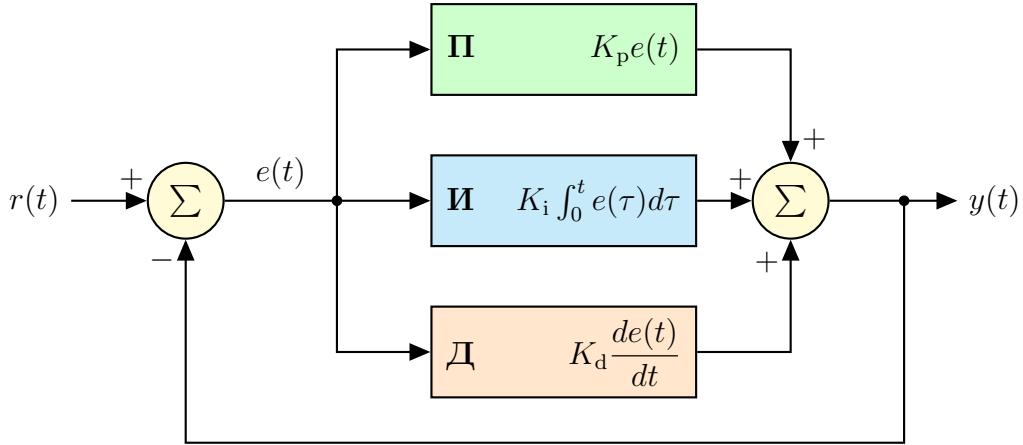
Математички, цјелокупна управљачка функција ПИД регулатора представља суму пропорционалног, интегралног и диференцијалног дејства и записује се на следећи начин:

$$y(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \quad (5)$$

где су K_p , K_i и K_d коефицијенти за пропорционални, интегрални и диференцијални дио, респективно. У стандардном облику, једначина (5) се записује као

$$y(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right). \quad (6)$$

Као што се може видјети, коефицијенти K_i и K_d су редом замјењени са K_p/T_i и $K_p T_d$, а предност таквог записа јесте што T_i и T_d имају одређено разумљиво физичко значење, јер представљају интегрално и диференцијално вријеме, респективно. K_p/T_i одређује колико временски дуго ће регулатор толерисати излаз који се налази изнад или испод жељене вриједности. $K_p T_d$ је временска константа којом се регулатор покушава приближити жељеној вриједности.



Слика 5: Блок дијаграм ПИД регулатора у повратној спрези.

Слика 5 представља уопштен блок дијаграм ПИД регулатора у повратној спрези који приказује структуру компоненти регулатора и њихов принцип рада. ПИД регулатор константно рачуна вриједност грешке $e(t)$ као разлику између жељене вриједности (енгл. *setpoint*) $r(t)$, која у овом конкретном случају представља улазни умножак учестаности, и измјерене процесне вриједности $y(t)$, односно вриједности измјерене у блоку управљачке предобраде. У том случају, једначина грешке може се записати као

$$e(t) = r(t) - y(t). \quad (7)$$

Одзив система тј. брзина и начин достизања жељене учестаности у ПИД режиму рада највише зависи од одабира константи регулатора, а у овом раду је остављена могућност софтверског уписа константи у регистре, што даље омогућава њихово накнадно мијењање и прилагођавање. У поглављу 7, где је приказан софтверски модел понашања FLL-а имплементиран у Пајтон програмском језику, поред осталог је дат и сликовит приказ утицаја константи ПИД регулатора на одзив система.

3.4 Управљачки декодер

На почетку је битно разјаснити суштинску потребу за блоком управљачког декодера. Наиме, могућа је имплементација и без управљачког декодера али тако да сваки тростатички инвертор DCO-а има посебну линију за управљање, што би у нашем случају значило 255 линија. Међутим, топологија DCO-а је таква да су тростатички инвертори распоређени у матрицу од 17 врста и 15 колона (више о топологији DCO-а се може наћи у поглављу 5), где сваки тростатички инвертор уз себе има и управљачки блок који омогућава да се свих 255 тростатичких инвертора контролише преко $17+17+15=49$ линија уместо са првобитних 255. То чини рутирање значајно простијим и без потребе за великим бројем линија метала, што олакшава и убрзава пројектовање DCO-а и омогућава већу скалабилност и брже тестирање.

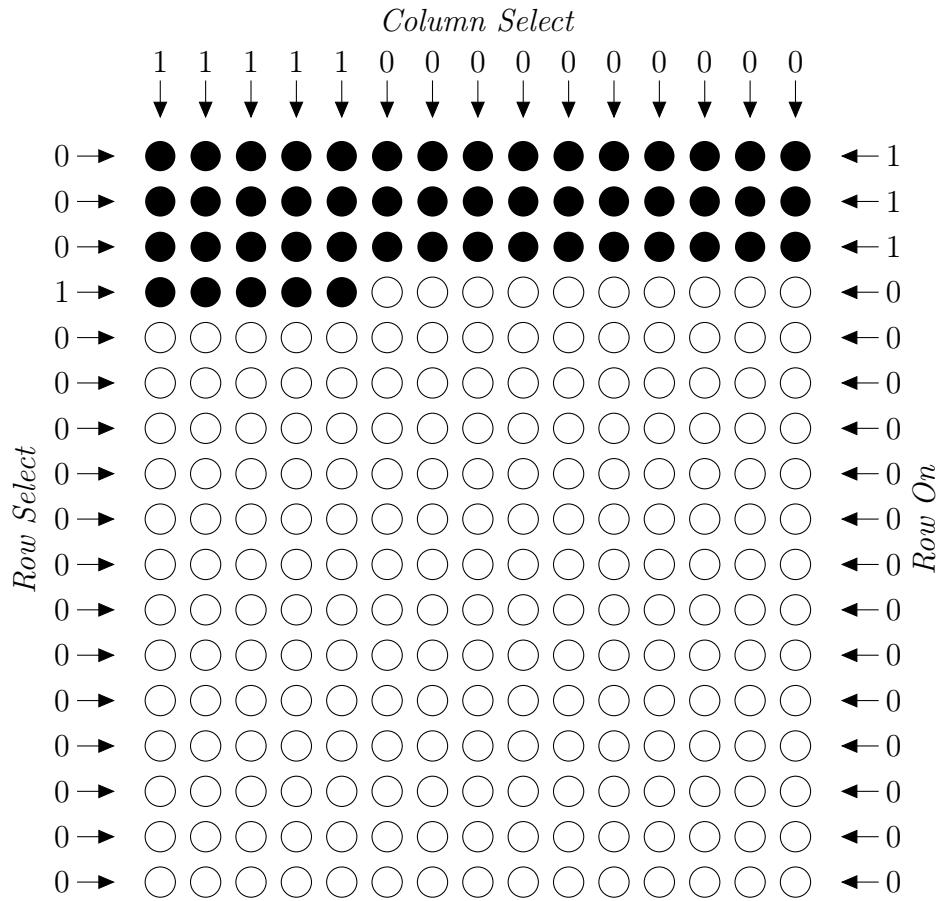
Практична улога управљачког декодера је претворити управљачке податке из једне бинарне вриједности у скуп управљачких улаза DCO-а. Постоје три таква улаза: *Row On*, унарни вектор, који може да укључује само читаве врсте тростатичких инвертора DCO-а; *Row Select*, један од n (енгл. *one-hot*) вектор, који укључује једну додатну врсту тростатичких инвертора DCO-а; и *Column Select*, унарни вектор, који може да укључује колоне тростатичких инвертора DCO-а.

Један од n вектор представља групу битова где је једина исправна комбинација она са једном јединицом и свим осталим нулама [9]. При представљању неког природног броја као један од n вектора, вриједност броја је једнака позицији јединице у вектору. С друге стране, да би се природан број N представио као унарни вектор, јединица се понавља N пута узастопно [10]. Табела 2 приказује примјер представљања природних бројева у унарном и један од n запису.

Табела 2: Поређење унарног и један од n записа

Децимални	Бинарни	Унарни	Један од n
0	000	00000000	00000001
1	001	00000001	00000010
2	010	00000011	00000100
3	011	00000111	00001000
4	100	00001111	00010000
5	101	00011111	00100000
6	110	00111111	01000000
7	111	01111111	10000000

Да би се инвертор укључио, или *Row On*, или и *Row Select* и *Column Select* за одговарајући бит морају бити подешени на 1, што је приказано у једначини (47). Широта сваког вектора је једнака ширини улаза DCO-а. Структура DCO-а је детаљно описана у поглављу 5.



Слика 6: Упростљен примјер начина управљања врстама и колонама матрице прекидача.

Управљачки податак који долази из претходног степена тј. контролера представља број тростатичких инвертора DCO-а који требају бити укључени, и што их је више укључено, то је излазна учестаност већа. Рецимо да је тај број 50, и претпоставимо да је DCO састављен од 255 тростатичких инвертора које поједностављено можемо посматрати као прекидаче и нека су организовани у матрицу од 17 врста и 15 колона. Како постоји 17 врста, ширине *Row On* и *Row Select* вектора биће по 17 битова, док ће због 15 колона, ширина *Column Select* вектора бити 15 битова.

Рецимо, да би се укључило 50 прекидача, потребно је укључити три читаве врсте прекидача и још пет прекидача из четврте врсте. Да би се то постигло, управљачки улази DCO-а морају имати следеће вриједности:

- *Row On* = 00000000000000111 → Сви прекидачи у прве три врсте су укључени.
- *Row Select* = 000000000000001000 → У четвртој врсти је укључено још прекидача.
- *Column Select* = 000000000011111 → Пет прекидача је укључено у врсти која је одређена са *Row Select*.

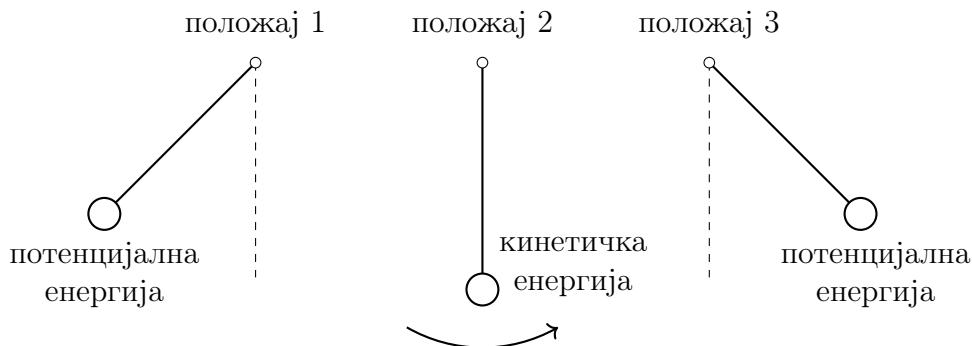
Слика 6 приказује описани принцип рада управљачког декодера када је потребно укључити 50 прекидача (тростатичких инвертора), где су пуним кружићима означени укључени, а празним искључени прекидачи. Сналажења ради, треба додати да је бит најмање тежине вектора *Column Select* скроз лијево, док су битови најмање тежине вектора *Row Select* и *Row On* скроз горе.

4 Теоријско разматрање осцилатора

У срцу сваке фазно затворене петље се налази осцилатор који игра кључну улогу у учинку који може бити постигнут [11]. Имајући то у виду, у наставку ће кроз теорију и примјере бити описани основни концепти осциловања, а након тога се прелази на разматрање концепта дигитално контролисаног осцилатора.

4.1 Основни принципи осциловања

У сврху теоријског разматрања осцилатора, као примјер биће узето клатно, као на слици 7.



Слика 7: Клатно које дјелује као осцилаторни систем [11].

Ако пустимо клатно под одређеним углом, оно се неко вријеме њише и постепено се зауставља. Осцилација почиње јер се првобитна потенцијална енергија претвара у кинетичку енергију како клатно достиже свој вертикални положај, дозвољавајући му да настави своју путању до другог екстремногугла (положај 3), под којим је енергија поново у потенцијалном облику. У реалним условима осцилација престаје јер трење на оси око које се окреће клатно и отпор ваздуха претварају дио енергије клатна у топлоту у сваком периоду осциловања.

Да би се одржала осцилација, потребно је обезбиједити спољну енергију клатну, како би се надомјестио губитак изазван трењем на оси и отпором ваздуха. Рецимо ако се лагано гурне клатно сваки пут када се врати у положај 1, оно ће настављати да се њише. Ако је притисак сувише слаб, врши се недовољна компензација, чиме се дозвољава осцилацији да ослаби и временом нестане. С друге стране, ако је притисак прејак, врши се прекомјерна компензација, приморавајући амплитуду замаха да се повећава из једног циклуса у други. Такође, треба напоменути да је период осциловања независан од амплитуде осциловања [11].

Механички примјер изнад указује на неколико саставних дијелова осцилаторног система:

1. Почетна „неравнотежа“, тј. почетно стање или количина енергије (обезбијеђена довођењем клатна у положај 1).
2. Склоност да се једна врста енергије претвара у другу и обратно.
3. Механизам за одржавање који допуњује енергију изгубљену услед неизбјежних несавршености.

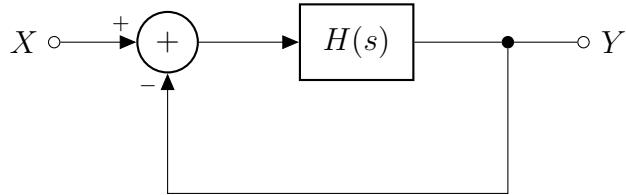
Не садрже сви осцилаторни системи све поменуте саставне дијелове, али је корисно имати ове концепте на уму при разматрању неког система који осцилује.

Ако се идеално клатно (клатно без губитака) пусти под одређеним углом, такво клатно осцилује бесконачно. Ако се дода одређени притисак сваки пут када клатно дође до лијевог kraja, тада замах наставља да расте због додатне енергије која се уноси у систем у сваком циклусу. Треба имати на уму да се тај неограничен раст не дешава ако се додатни притисак унесе на учестаности различитој од природне учестаности осциловања клатна. Ако систем има склоност да осцилује на учестаности ω_0 , тада се ствара растући осцилаторни излаз као одговор на спољашњу побуду на учестаности ω_0 . Из друге перспективе, такав систем неограничено појачава периодични улаз на поменутој учестаности.

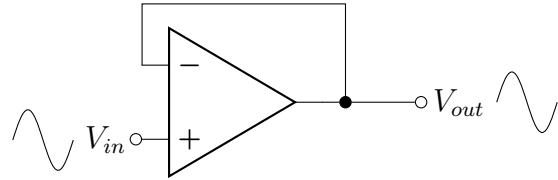
4.2 Систем осцилаторне повратне спреге

Из основа аутоматског управљања се зна да систем са негативном повратном спрегом може постати нестабилан. Ово својство се користи за конструисање осцилатора.

У сврху проучавања осцилација у фреквенцијском домену, разматрамо систем повратне спреге приказан на слици 8, где негативан предзнак на једном улазу сабирача означава негативну повратну спрегу на ниским учестаностима. На слици 9 је приказан примјер имплементације такве повратне спреге, који, на нискофреквентни простопериодични улаз дјелује као јединични бафер. Треба имати на уму да операциони појачавач показује занемарљив фазни помак на ниским учестаностима.



Слика 8: Једноставан систем повратне спреге.



Слика 9: Реализација повратне спреге коришћењем операционог појачавача.

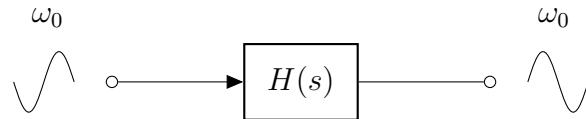
Међутим, поставља се питање како могу системи с. Слика 8 и 9 да осцилују? Записивањем преносне функције затворене петље као

$$\frac{Y(s)}{X(s)} = \frac{H(s)}{1 + H(s)}, \quad (8)$$

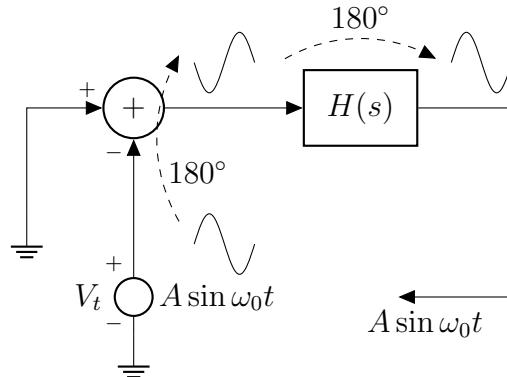
примјећује се да именилац пада на нулу ако је $H(s) = -1$ за неку вриједност пројењиве s . Ако је $X(s)$ Лапласова трансформација синусоиде $x(t)$, онда је $s = j\omega_0$

и мора важити $H(j\omega_0) = -1$. С обзиром да је у питању синусоида, то ће на ω_0 важити за фазни помјерај од -180° . То даље значи да важи $(Y/X)(j\omega_0) \rightarrow \pm\infty$, одакле се може закључити да систем обезбеђује неограничен добитак за такву синусоиду. Као што је претпостављено у претходним поглављу, такво понашање наговјештава осцилаторну петљу.

Једнакост $H(j\omega_0) = -1$ значи да $H(s)$ инвертује улаз на учестаности ω_0 , што је илустровано на слици 10. Другим ријечима, $H(s)$ има толико помјерања фазе (или кашњења) на ω_0 да укупна повратна спрега постаје позитивна. То се може видјети постављањем главног улаза, $X(s)$, на нулу, прекидом петље и примјеном стимулуса на истој учестаности (слика 11). Враћени сигнал је у фази са тестним напонским генератором, V_t . У овом случају, каже се да петља садржи фазни помјерај од 180° због номинално негативне повратне спрете и други, фреквенцијски зависан, фазни помјерај од 180° који произилази из $H(s)$. Та два фазна помјераја се не смију мијешати један с другим.



Слика 10: Инерција сигнала на учестаности ω_0 .



Слика 11: Пропагација синусоиде кроз петљу на учестаности ω_0 .

Укупан фазни помјерај од 360° на ω_0 имплицира да се сигнал враћа да би се појачао док кружи у петљи. Ова појава доводи до пораста амплитуде јер је враћени сигнал бар толико велик као и почетни сигнал, односно зато што је $|H(j\omega_0)| = 1$. Стога се услови за осциловање сумирају као:

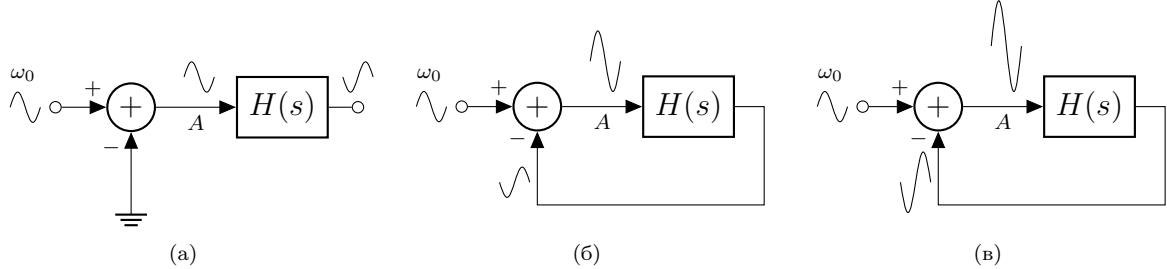
$$|H(j\omega_0)| = 1 \quad (9)$$

$$\angle H(j\omega_0) = -180^\circ, \quad (10)$$

и називају се Баркхаузеновим критеријумима [11]. Услов $H(j\omega_0) = -1$ се назива и услов покретања или почетни услов.

Нагомилавање осцилација се такође може посматрати и у временском домену. Са слике 12a се примјеђује да је, са $H(j\omega_0) = -1$, излаз једнак улазу, али помјерен за

180° . Ако је петља затворена (слика 12б), излаз се одузима од улаза, што даје већу амплитуду (замах) на A . Тада сигнал се затим поново инвертује и одузима од улаза, и тако у круг, што доводи до неограниченог раста амплитуде (слика 12в).

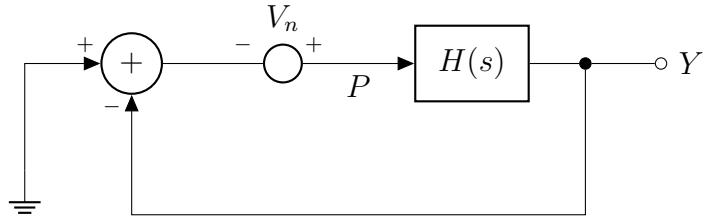


Слика 12: Раст амплитуде улазне синусоиде кроз петљу у току времена.

Укратко, систем са негативном повратном спрегом може да генерише растући периодични излаз као одговор на синусоидни улаз ако његово појачање у петљи достиже -1 на коначној учестаности, ω_0 . Али, да ли такав систем осцилује ако не примјенимо никакав улаз? Одговор је да, јер широкопојасни шум уређаја унутар петље има коначну енергију у близини ω_0 , стварајући малу компоненту која кружи у петљи и изазива осцилације. Нпр. као што је приказано на слици 13, извор шума, V_n , на улазу $H(s)$ даје излаз који је једнак:

$$Y = \frac{H(s)}{1 + H(s)} V_n, \quad (11)$$

чиме се доводи до неограниченог појачања за $s = j\omega_0$. То такође значи да, чак иако је V_n бесконачно мало на ω_0 , на Y се може претпоставити појављивање неке коначне амплитуде осциловања тј. замаха (енгл. *swing*).



Слика 13: Извор шума у систему са негативном повратном спреком.

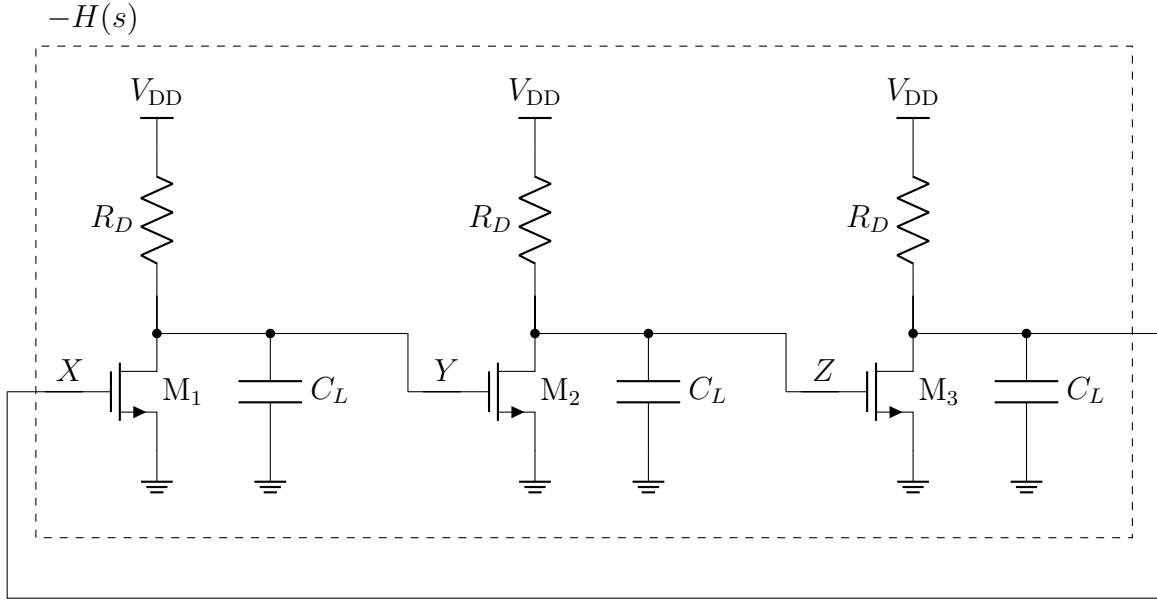
4.3 Основе прстенастог осцилатора

Прстенasti осцилатори су у данашње вријеме веома популарни у PLL системима захваљујући својој прилагодљивости дизајна и широког опсега подешавања учестаности. Ово поглавље уопштено разматра основе прстенастих осцилатора, да би се стекло дубље разумијевање рада коришћеног дигитално контролисаног осцилатора, чији конкретан дизајн је описан у поглављу 5.

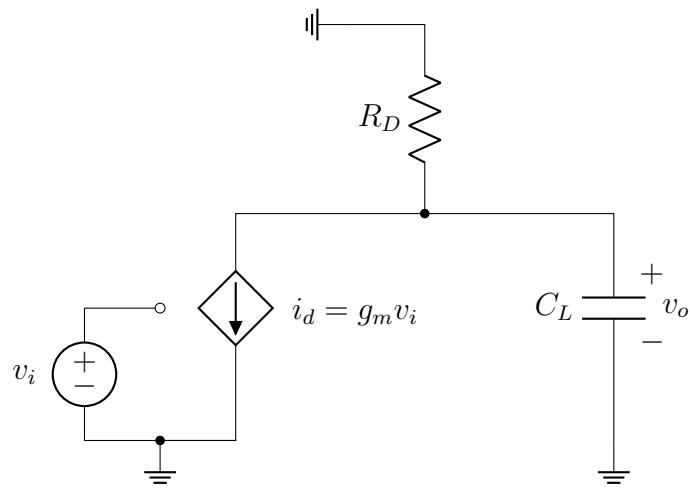
Слика 14 приказује основни концепт прстенастог осцилатора са три степена. Ако се користи само један степен, тада се не обезбеђује довољан фазни помјерај да би важио услов из једначине (10). Чак и примјеном два степена, систем не може да

осцилује јер $\angle H(j\omega_0)$ достиже -180° тек за $\omega = \infty$ [11]. Због тога се претпоставља да коришћењем три степена могу бити задовољени услови из једначина (9) и (10), па се стога врши даља анализа како би се то и показало.

На слици 14, коло означено испрекиданом линијом је означенено са $-H(s)$ да би било у складу са системом негативне повратне спрете приказаним на слици 9.



Слика 14: Три степена са заједничким извором у повратној спрети.



Слика 15: Еквивалентно коло једног степена у петљи.

На слици 15 приказано је еквивалентно коло на нивоу малих сигнала за један степен прстенастог осцилатора приказаног на слици 14. У случају да за тренутак занемаримо капацитивно оптерећење тј. да важи $C_L = 0$, добија се уобичајена функција одзива појачавача

$$v_o = -K(V_I - V_T)R_L v_i = -g_m R_D v_i. \quad (12)$$

Када су $v_i = V_i \cos(\omega t)$ и $C_L = 0$, тада је напон на излазу једнак

$$v_o = -g_m R_D V_i \cos(\omega t). \quad (13)$$

У једначинама (12) и (13), са g_m је означена транскондуктанса NMOS транзистора која представља фактор појачања транзистора. Она је обрнуто сразмјерна отпорности и уопштено за MOSFET је једнака

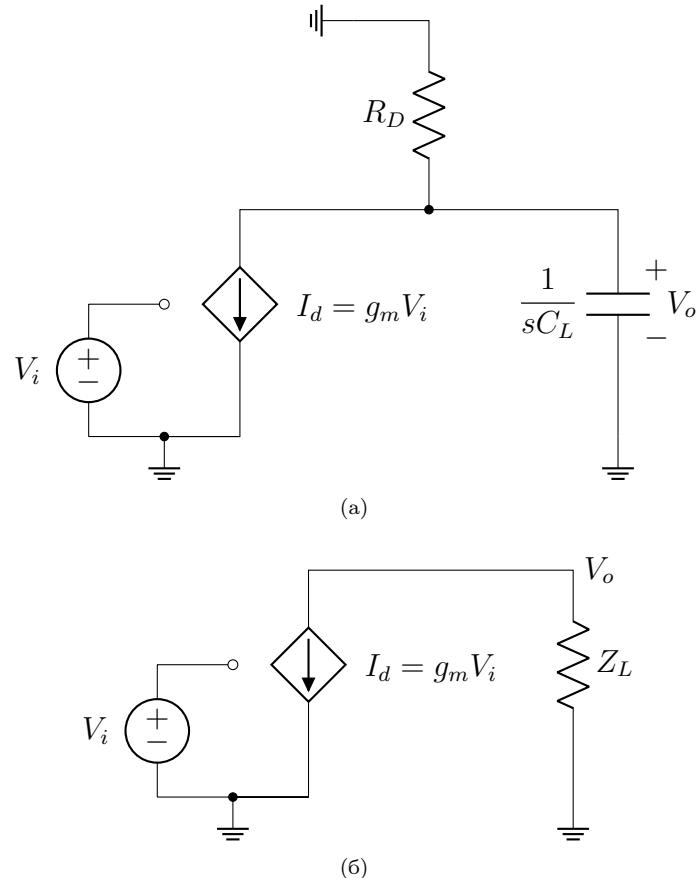
$$g_m = K(V_I - V_T), \quad (14)$$

где је $V_I = V_{GS}$ тј. напон од гејта до сорса, V_T је напон прага, а K је параметар повезан са физичком структуром MOSFET-а и једнак је

$$K = K_n \frac{W}{L}. \quad (15)$$

Са W је означена ширина гејта, са L његова дужина, док је K_n константа повезана са MOSFET карактеристикама као што је дебљина оксида [12].

Ако се сада и кондензатор узме у разматрање, да би се пронашла веза између v_i и v_o за коло са слике 15, потребно је направити импедансни модел кола, а он је приказан на слици 16а.



Слика 16: Импедансни модел еквивалентног кола једног степена у петљи.

Ефективна вриједност импедансе оптерећења, Z_L (слика 16б) дата је са

$$Z_L = R_D \parallel \frac{1}{sC_L}. \quad (16)$$

Тада је израз за амплитуду излаза (V_o) једнак

$$V_o = -g_m Z_L V_i = -g_m \left(R_D \parallel \frac{1}{sC_L} \right) V_i. \quad (17)$$

То је еквивалентно са

$$\frac{V_o}{V_i} = -g_m \left(R_D \parallel \frac{1}{sC_L} \right). \quad (18)$$

Како је једначина (18) изведена за један степен, тада за преносну функцију тростепеног прстена са слике 14 важи [11]

$$-H(s) = \left[-g_m \left(R_D \parallel \frac{1}{sC_L} \right) \right]^3. \quad (19)$$

Ако је $s = j\omega_0$, тада (19) пишемо као

$$H(j\omega_0) = \left[g_m \left(R_D \parallel \frac{1}{j\omega_0 C_L} \right) \right]^3 = \left(\frac{g_m R_D \frac{1}{j\omega_0 C_L}}{R_D + \frac{1}{j\omega_0 C_L}} \right)^3 = \left(\frac{g_m R_D}{1 + j\omega_0 R_D C_L} \right)^3. \quad (20)$$

Добијени резултат представља комплексни разломак који даље желимо представити у стандардном комплексном облику $z = x + jy$:

$$\begin{aligned} H(j\omega_0) &= \left(\frac{g_m R_D}{1 + j\omega_0 R_D C_L} \cdot \frac{1 - j\omega_0 R_D C_L}{1 - j\omega_0 R_D C_L} \right)^3 = \\ &= \left(\frac{g_m R_D - j\omega_0 g_m R_D^2 C_L}{1 + \omega_0^2 R_D^2 C_L^2} \right)^3 = \\ &= \left(\frac{g_m R_D}{1 + \omega_0^2 R_D^2 C_L^2} - j \frac{\omega_0 g_m R_D^2 C_L}{1 + \omega_0^2 R_D^2 C_L^2} \right)^3. \end{aligned} \quad (21)$$

Да би се поставио услов $|H(j\omega_0)| = 1$, потребно је наћи модул (интензитет) степеноване вриједности комплексног израза добијеног изнад. Рецимо да имамо неки комплексан број $z = x + jy$. Он се може представити и у поларном облику као

$$z = A(\cos\theta + j\sin\theta), \quad (22)$$

где су:

- $A = |z| \rightarrow$ модул (апсолутна вриједност) комплексног броја,
- $\theta = \arg(z) = \arctan\left(\frac{y}{x}\right) \rightarrow$ фазни угао (аргумент).

Коришћењем Ојлерове формуле, $e^{j\theta} = \cos\theta + j\sin\theta$, комплексни број можемо представити као

$$z = A e^{j\theta}. \quad (23)$$

Ако тако представљен комплексни број степенујемо неким природним бројем n , добија се

$$z^n = (A e^{j\theta})^n = A^n (e^{j\theta})^n = A^n e^{jn\theta}. \quad (24)$$

То значи да степеновањем комплексног броја важи следеће:

1. Модул комплексног броја се подиже на исти степен.
2. Фазни угао се множи са вриједношћу степена.

Ако се први закључак примијени на наш случај, важи следеће:

$$\begin{aligned}
|H(j\omega_0)| &= \left| \frac{g_m R_D}{1 + \omega_0^2 R_D^2 C_L^2} - j \frac{\omega_0 g_m R_D^2 C_L}{1 + \omega_0^2 R_D^2 C_L^2} \right|^3 = \\
&= \left(\sqrt{\frac{g_m^2 R_D^2}{(1 + \omega_0^2 R_D^2 C_L^2)^2} + \frac{\omega_0^2 g_m^2 R_D^4 C_L^2}{(1 + \omega_0^2 R_D^2 C_L^2)^2}} \right)^3 = \\
&= \left(\sqrt{\frac{g_m^2 R_D^2 (1 + \omega_0^2 R_D^2 C_L^2)}{(1 + \omega_0^2 R_D^2 C_L^2)^2}} \right)^3 = \\
&= \left(\frac{g_m R_D}{\sqrt{1 + \omega_0^2 R_D^2 C_L^2}} \right)^3.
\end{aligned} \tag{25}$$

На основу претходног израза, услов $|H(j\omega_0)| = 1$ се може представити као

$$\left(\frac{g_m R_D}{\sqrt{1 + \omega_0^2 R_D^2 C_L^2}} \right)^3 = 1. \tag{26}$$

Одатле се може изразити ω_0 на следећи начин:

$$\omega_0 = \frac{\sqrt{g_m^2 R_D^2 - 1}}{R_D C_L}. \tag{27}$$

Ако сада други закључак примјенимо на фазни угао преносне функције, тада ће $\angle H(j\omega_0)$ бити једнак

$$\begin{aligned}
\angle H(j\omega_0) &= 3 \cdot \arctan \left(\frac{-\frac{\omega_0 g_m R_D^2 C_L}{1 + \omega_0^2 R_D^2 C_L^2}}{\frac{g_m R_D}{1 + \omega_0^2 R_D^2 C_L^2}} \right) = \\
&= 3 \cdot \arctan(-\omega_0 R_D C_L) = \\
&= -3 \cdot \arctan(\omega_0 R_D C_L).
\end{aligned} \tag{28}$$

На основу претходног израза, услов $\angle H(j\omega_0) = -180^\circ$ може се представити као

$$\arctan(\omega_0 R_D C_L) = 60^\circ. \tag{29}$$

Одатле се може представити ω_0 на следећи начин:

$$\omega_0 = \frac{\sqrt{3}}{R_D C_L}. \tag{30}$$

Ако сада изједначимо изразе (27) и (30), добија се:

$$\frac{\sqrt{g_m^2 R_D^2 - 1}}{R_D C_L} = \frac{\sqrt{3}}{R_D C_L}. \tag{31}$$

Одатле се даље добија:

$$g_m R_D = 2. \tag{32}$$

Другим ријечима, сваки степен мора обезбједити појачање напона ниске учестаности од 2, да бе се гарантовало осциловање [11].

У општем случају, ако се уведе N идентичних степени у прстен, где је N непаран број, петља обезбеђује негативну повратну спреку на ниским учестаностима, што захтјева фазни помјерај од $-180^\circ/N$ по једном степену да би систем осциловао. Тада ће ω_0 бити једнако

$$\omega_0 = \frac{1}{R_D C_L} \tan \frac{180^\circ}{N} = \frac{1}{R_D C_L} \tan \frac{\pi}{N}. \quad (33)$$

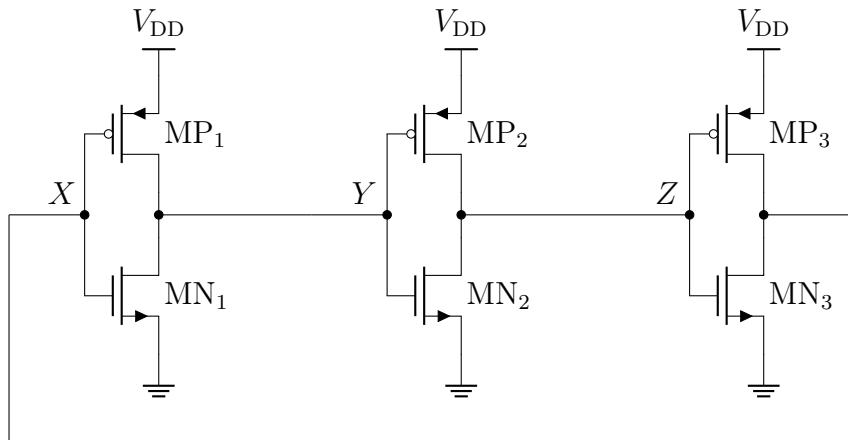
Како се N повећава, ω_0 опада јер свака фаза мора унијети мање фазног помјераја. Такође, у општем случају, како N расте, захтијевано нискофреквентно појачање се смањује што се може видjetи из израза

$$g_m R_D = \sqrt{\tan^2 \frac{180^\circ}{N} + 1} = \sqrt{\tan^2 \frac{\pi}{N} + 1}. \quad (34)$$

Видјели смо да су таласни облици прстенастог осцилаотра блиски синусоиди за $|H(j\omega_0)| = 1$. Међутим, они почињу да личе на квадратни талас ако је $|H(j\omega_0)|$ знатно изнад 1. Већина апликација тежи оштријим прелазима сигнала како би се осигурало брзо укључивање и искључивање уређаја. Стога су квадратни таласни облици пожељнији за примјену, и у те сврхе се уводи CMOS инверторски прстен обрађен у наредном поглављу.

4.3.1 Инверторски прстен

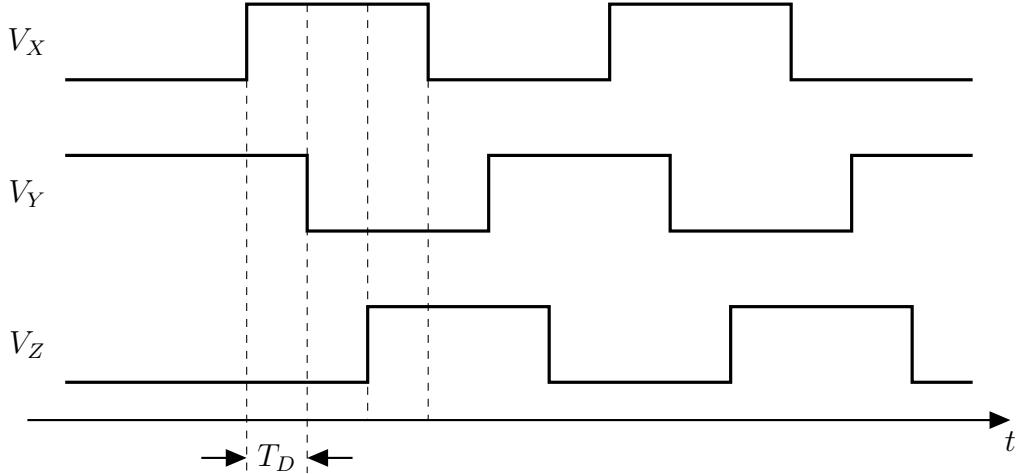
Најчешћа имплементација прстенастих осцилатора је коришћењем CMOS инвертора уместо отпорнички оптерећених појачавача као фаза појачања у прстену. Слика 17 приказује тростепени инверторски прстен. Ако претпоставимо да је у почетку $V_X = V_Y = V_Z$, свака фаза уводи напонско појачање од $-(g_{mN} + g_{mP})(r_{ON}||r_{OP})$, које типично има вриједност преко 2 [11]. Након тога, шум унутар компоненти узрокује обнављање сигнала све док амплитуда осциловања не достигне пун напонски опсег (енгл. *rail-to-rail*).



Слика 17: Тростепени прстенасти осцилатор састављен од три CMOS инвертора.

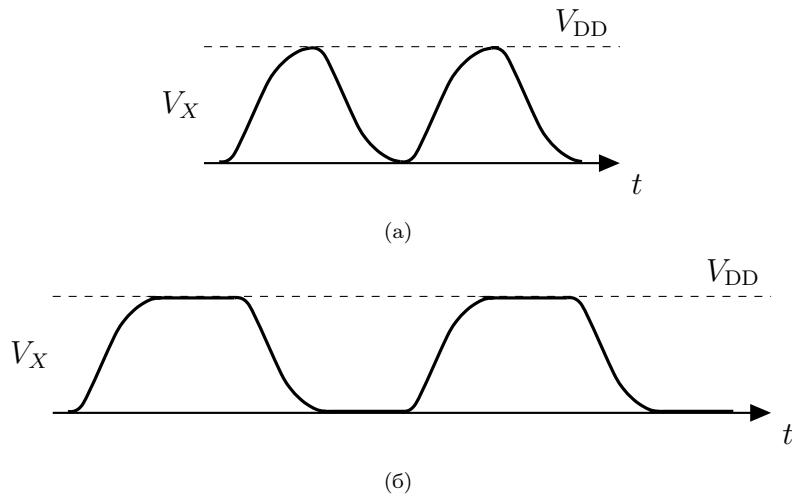
Шта ако, након укључења, прстен почиње са једним чвором на нули? Рецимо, ако је $V_X = 0$, тада је $V_Y = V_{DD}$, а $V_Z = 0$, што значи да трећи инвертор подиже V_X на

V_{DD} . Слика 18 приказује идеализоване облике сигнала на сва три степена у прстену, у току времена. Закључујемо да прелаз на једном чвору узрокује прелаз на наредном након одређеног кашњења, T_D . Укупан период осциловања је стога једнак $6T_D$, па је учестаност осциловања једнака $f = 1/6T_D$.



Слика 18: Облици дигиталног сигнала на фазама тростепеног прстенастог осцилатора са Слике 17.

Стварни облици сигнала претходног осцилатора нису тако оштри као на слици 18, али нису ни правилни као синусоида. Реалнији примјер је илустрован на слици 19а, где V_X мијења правац чим достигне V_{DD} или 0 , због кратког кашњења кроз петљу. Поређења ради, петостепени прстен омогућава осцилације са периодом $10T_D$, дозвољавајући сигналу на V_X да дуже времена остане на високом, односно ниском напонском нивоу као на слици 19б.

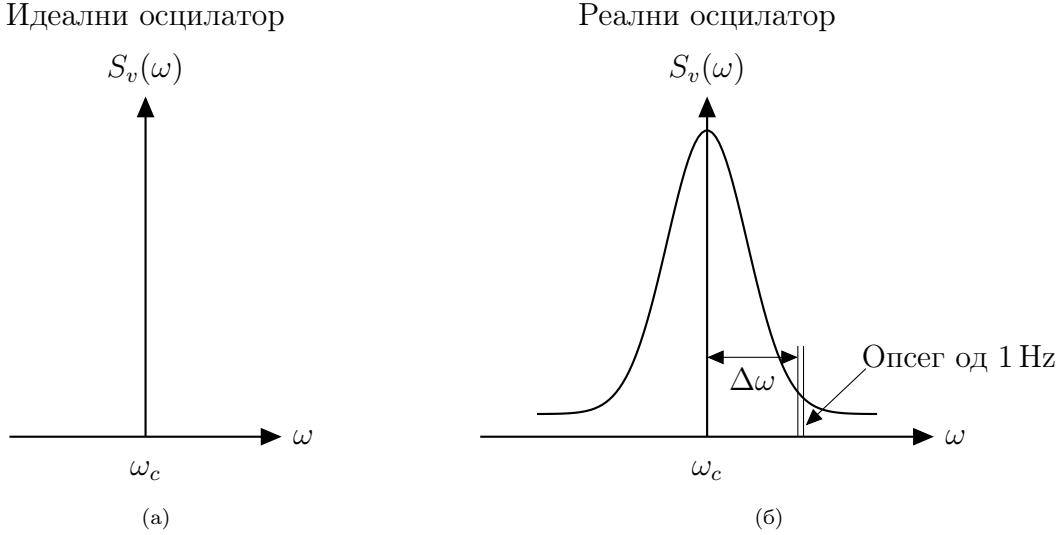


Слика 19: Реалан облик сигнала (а) тростепеног и (б) петостепеног прстенастог осцилатора.

Прстенасти осцилатори су уопштено прилично осјетљиви на вриједности напона напајања. Прво, максимална учестаност осциловања расте како расте напон напајања. Друго, динамичка потрошња снаге прстенастог осцилатора има чак квадратну зависност од вриједности напона напајања, о чему ће више ријечи бити у поглављу 5.

4.4 Шум у осцилаторима

За идеалан осцилатор који ради на учестаности ω_c , сва његова снага је концентрисана на једну учестаност, ω_c , као што је приказано на слици 20a. У реалном осцилатору, спектар се шири на околне учестаности око ω_c (слика 20б). У осцилаторима се такво ширење назива фазни шум.



Слика 20: Излазни спектар (а) идеалног и (б) реалног осцилатора.

Фазни шум се увијек карактерише у фреквенцијском домену. За идеалан осцилатор који ради на учестаности ω_c , излазни напон може бити изражен као

$$v(t) = A \cos(\omega_c t + \phi), \quad (35)$$

гдје је A амплитуда, а ϕ је произвољна, али фиксна референтна фаза. Снага је концентрисана на једну учестаност ω_c . Еквивалентно, спектар снаге се може записати као

$$S_v(\omega) = \frac{A^2}{2} \delta(\omega - \omega_c), \quad (36)$$

гдје је δ јединична импулсна функција или Диракова делта функција (слика 20а), за коју важи:

$$\int_{-\infty}^{+\infty} \delta(x) dx = 1. \quad (37)$$

Међутим, у реалном осцилатору и амплитуда и фаза су временски промјенљиве, и спектар ће се зато ширити око учестаности носиоца на околне учестаности. У већини случајева, поремећај у амплитуди је занемарљив или неважан, а и може се лако уклонити додавањем кола која ограничавају амплитуду [6]. Стога се разматра само насумично одступање фазе:

$$v(t) = A \cos[\omega_c t + \phi(t)], \quad (38)$$

гдје је $\phi(t)$ мало насумично одступање фазе које представља варијације у периоду и најчешће се назива фазни шум. За мале вриједности варијација фазног шума, $|\phi(t)| \ll 1 \text{ rad}$, једначина (38) се може поједноставити на

$$v(t) = A \cos \omega_c t + A\phi(t) \sin \omega_c t, \quad (39)$$

што значи да се спектар $\phi(t)$ фреквенцијски преводи у $\pm\omega_c$.

Фазни шум се може квантификовати узимањем опсега од 1 Hz на растојању $\Delta\omega$ од носиоца, затим рачунањем снаге шума у том опсегу и дијељењем тог резултата са снагом носиоца [6]. То се назива једнострани (енгл. *single-sided*) фазни шум и изражава се у јединици децибела по херцу [dBc/Hz]. По конвенцији, „с“ у „dBc“ значи „у односу на носилац (енгл. *carrier*)“.

$$\mathcal{L}(\Delta\omega) = 10 \log_{10} \left(\frac{\text{снага шума у опсегу од } 1 \text{ Hz при учестаности } \omega_c + \Delta\omega}{\text{снага носиоца}} \right) \quad (40)$$

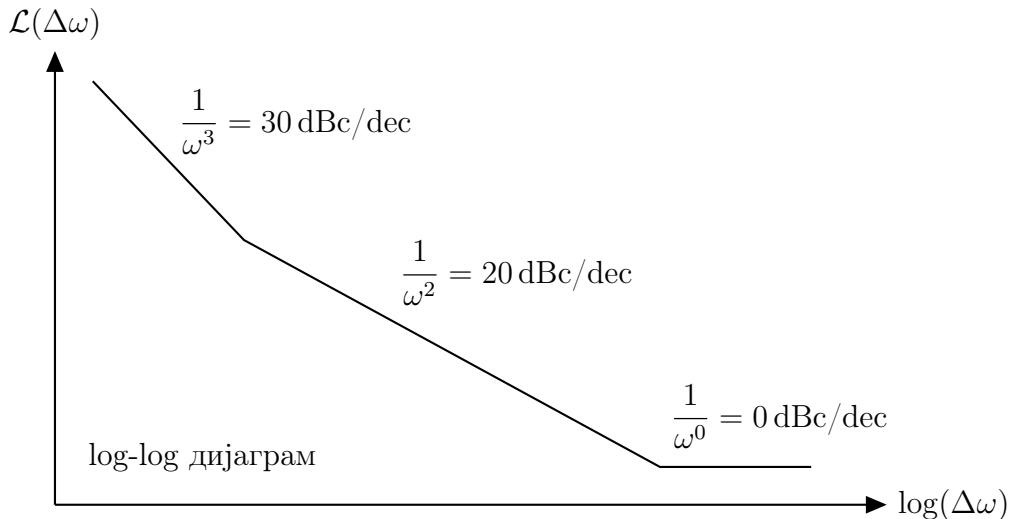
Једнострани фазни шум из (40) је половина спектра фазног шума, тј. може се записати као

$$\mathcal{L}(\Delta\omega) = 10 \log_{10} \left(\frac{S_\phi(\Delta\omega)}{2} \right), \quad (41)$$

где је $S_\phi(\Delta\omega)$ дато са

$$S_\phi(\Delta\omega) = \frac{S_v(\Delta\omega)}{\text{снага носиоца}}. \quad (42)$$

Типични спектар фазног шума осцилатора приказан је на слици 21. На овом логаритамском дијаграму, фазни шум нормализован на dBc/Hz, приказан је у односу на офсет учестаност $\Delta\omega$ од носиоца ω_c .

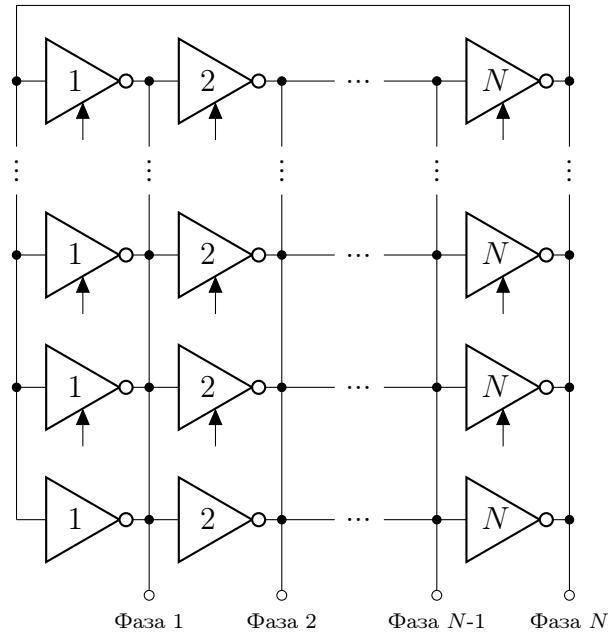


Слика 21: Спектар фазног шума реалног осцилатора.

Профил фазног шума прати криву приказану на слици, гдеје пролази кроз области са нагибима од $1/\omega^3$, $1/\omega^2$ и $1/\omega^0$. Област $1/\omega^2$ се уопште назива област термичког шума (енгл. *thermal noise region*) зато што је изазвана тзв. бијелим или неповезаним временским флуктуацијама унутар периода осциловања. Фликер шум (енгл. *flicker noise*) $1/f$ електронских уређаја је такође значајан за ниже офсетне учестаности. Он се конвертује навише у $1/\omega^3$ област. Коначно, $1/\omega^0$ је термички електронски шум који се додаје изван осцилатора, као нпр. у излазном баферу, и не утиче на временски одзив осцилатора.

5 Дигитално контролисани осцилатор

Дигитално контролисани осцилатор описан у овом раду је прстенасти осцилатор, погодан за систем генерирања такта. Прстенасти осцилатор је каскадна комбинација фаза кашњења повезаних у ланац затворене петље [13]. Прстенасте архитектуре су компактније од LC осцилатора и имају доста предности захваљујући својој правилној и периодичној просторној структури. Уопштена структура DCO-а коришћеног унутар описаног FLL-а заснована је на матрици тростатичких CMOS инвертора [14]. Ова матрица је састављена од N -фазних прстенова тростатичких инвертора повезаних паралелно. N представља број DCO фаза (степени) и мора бити непаран број већи или једнак 3. Слика 22 приказује примјер N -степеног дигитално контролисаног осцилатора који има један прстен стално укључених инвертора (најдоњи прстен без стрелице управљачког сигнала са доње стране инвертора).



Слика 22: N -степени дигитално контролисани осцилатор.

У физичком смислу, матрица може бити преобликована у правоугаоник, што омогућава једноставнију управљачку логику. На слици 22, основну учестаност DCO-а дефинише само један прстен који је стално укључен, а у општем случају их може бити више. Остали тростатички инвертори се укључују и искључују у зависности од управљачке логике.

Формула за учестаност осциловања DCO-а гласи:

$$f_{\text{osc}} = \frac{1}{2Nt_d} \approx \frac{I_d}{2NC_{\text{load}}V_{\text{DDL}}}, \quad (43)$$

где је N број тростатичких инвертора унутар прстена, t_d представља кашњење једне ћелије DCO-а, у чијем саставу је тростатички инвертор (у наставку ће бити детаљније објашњена структура саме ћелије DCO-а), I_d је струја која протиче кроз инвертор, C_{load} је капацитивно оптерећење истог инвертора, и V_{DDL} је напон напајања DCO-а. Производ Nt_d је помножен са 2 да би се добила читава периода такта, а не полупериода.

У поглављу 4.3.1 је напоменуто да учестаност осциловања расте са повећањем напона напајања. Међутим, из једначине (43) се грешком може закључити баш супротно. Истина јесте да учестаност осциловања расте са повећањем напона напајања, а разлог је тај што струја која противе кроз MOSFET транзистор у засићењу има квадратну зависност од напона од гејта до сорса, V_{GS} , који је у прстенастој структури осцилатора једнак V_{DDL} , па ће вриједност I_d из бројиоца у једначини (43) имати доминантнији утицај на учестаност осциловања него сама вриједност V_{DDL} из имениоца.

Када је ријеч о топологији, са повећањем броја DCO фаза (степени), фреквенцијски корак (K_{DCO}), односно појачање осцилатора опада, чиме се повећава прецизност DCO-а. Максимална учестаност осцилатора се такође смањује, и да би се то надомјестило, напон напајања се може повећати, што с друге стране доводи до веће потрошње снаге. Ако претпоставимо да напон напајања и капацитивно оптерећење по једној фази остану непромјењени, повећање броја фаза не утиче на потрошњу снаге. Међутим, ако укупан број тростатичких инвертора остане непромјењен и подијели се на већи број фаза, то ће довести до смањења капацитивног оптерећења по фазама појединачно, што даље доводи до смањења потрошње снаге. Математичком анализом се то може објаснити на следећи начин: N -фазни прстенасти осцилатор који ради на учестаности f_{osc} има динамичку потрошњу снаге која се може представити једначином

$$P = N f_{osc} C_{tot} V_{DDL}^2, \quad (44)$$

гдје C_{tot} представља укупно капацитивно оптерећење на једној фази [15]. Пошто је учестаност осциловања једнака $f_{osc} = 1/(2Nt_d)$, једначину за динамичку снагу можемо написати на следећи начин:

$$P = \frac{C_{tot} V_{DDL}^2}{2t_d}, \quad (45)$$

одакле се види да је добијена динамичка снага независна од N [11].

С обзиром да је одређени број тростатичких инвертора увијек укључен, DCO има одређену почетну учестаност која не зависи од управљачке ријечи DCO-а. Тренутна излазна учестаност ће стога у основи зависити од почетне учестаности, фреквенцијског корака тј. резолуције и управљачке ријечи. Стога се може написати следећа линеарна зависност учестаности осциловања, која може послужити за компактни модел DCO-а [16]:

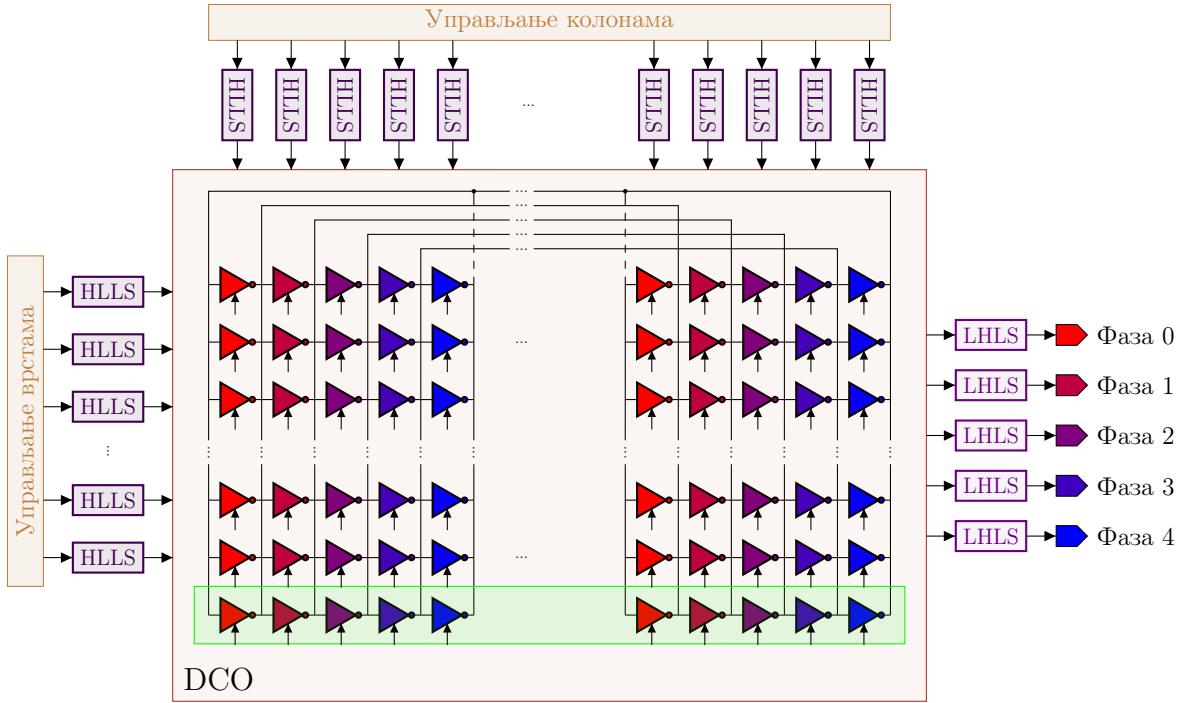
$$f_{DCO} = f_0 + K_{DCO} \cdot d, \quad (46)$$

гдје је f_0 почетна учестаност, K_{DCO} резолуција учестаности, а d је управљачка дигитална ријеч.

5.1 Архитектура предложеног DCO-а

У овом раду описана је топологија DCO-а са пет фаза, због тога што је таквом топологијом остварен задовољавајући компромис између учинка и потрошње снаге тј. под задовољавајућим условима се достиже жељена учестаност са довољном прецизношћу. Слика 23 приказује структуру DCO-а коришћеног у описаној FLL имплементацији.

Свака фаза DCO-а састоји се од 54 тростатичка инвертора, што ако помножимо са бројем фаза даје укупно 270 инвертора. Тростатички инвертори су распоређени



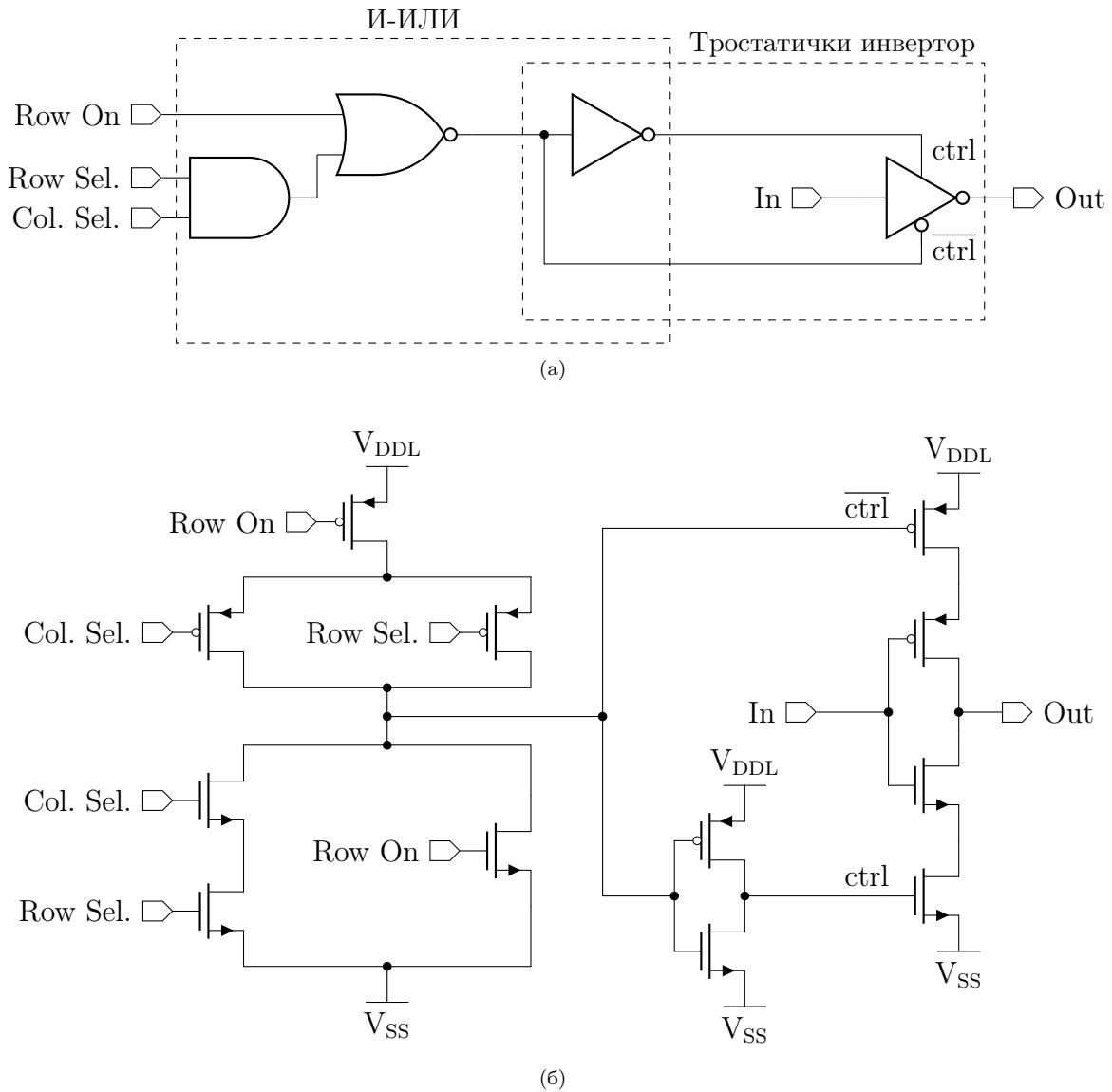
Слика 23: Петостепени прстенаси дигитално контролисани осцилатор (DCO) састављен од тростатичких инвертора, са додатим претварачима напонских нивоа (HLLS и LHLS) и уоквиреном врстом стално укључених тростатичких инвертора.

у 18 врста и 15 колона. Управљачка логика FLL-а управља са 17 врста и свих 15 колона, што значи да постоји 255 фреквенцијских корака. Преостала врста са 3×5 тростатичких инвертора је стално укључена и на њу не утиче управљачка логика FLL-а. Што се више тростатичких инвертора у свакој фази укључи под утицајем управљачке логике FLL-а, повећавају се струја и тренутна снага покретања (енгл. *driving strength*) једне фазе, док њено капацитивно оптерећење у суштини остаје константно, што резултује повећањем излазне учестаности осциловања [4]. Поред самих тростатичких инвертора, на слици 23 се могу уочити и блокови претварача напонског нивоа на управљачким улазима и фазама DCO-а, чија је намјена детаљно објашњена у поглављу 5.3.

5.2 Блок дијаграм ћелије DCO-а

Да би се ефикасно подесила излазна учестаност DCO-а, уведен је скуп управљачких улаза DCO-а, а то су сигнали *Row On*, *Row Select* и *Column Select*, поменути такође у поглављу 3.4. Према томе, сваки тростатички инвертор садржи сопствену управљачку јединицу у облику И-ИЛИ стандардне ћелије, и заједно граде већи блок назван ћелија DCO-а. Слика 24 приказује шему ћелије DCO-а на нивоу логичких кола и на нивоу CMOS транзистора. Уочава се да И-ИЛИ коло и тростатички инвертор имају заједнички инвертор како би се уштедило на броју транзистора.

У дигиталном домену сигнал *Out* може имати три стања. Једно је стање високе импедансе, када је тростатички инвертор искључен. А друга два су када је тростатички инвертор укључен и представљају инвертован сигнал *In*. Укључивање и искључивање тростатичког инвертора контролише *ctrl* сигнал који представља излаз из контролног дијела DCO ћелије тј. И-ИЛИ блока стандардне ћелије. За вриједност *ctrl*



Слика 24: Џелија DCO-а на нивоу (а) логичких кола и (б) CMOS транзистора.

сигала 1, тростатички инвертор је укључен, а за вриједност 0 је искључен. Таблица истинитости управљачког сигнала $ctrl$ дата је у табели 3. Логички израз за сигнал $ctrl$ у зависности од управљачких сигнала врста и колона гласи:

$$ctrl = (\text{Row Sel.} \cdot \text{Col. Sel.}) + \text{Row On} \quad (47)$$

Таблица истинитости Out сигнала у зависности од управљачког сигнала $ctrl$ и улазног In дата је у табели 4 .

5.2.1 Џелија DCO-а са управљачким сигналима активним на низак логички ниво

Такође је могуће да постоји захтјев да улазни сигнали џелије DCO-а буду активни на низак логички ниво. У том случају шема џелије на нивоу логичких кола и транзистора би изгледала као на слици 25, док број транзистора остаје исти као у

Табела 3: Таблица истинитости контролног сигнала тростатичког инвертора.

Row On	Row Sel.	Col. Sel.	ctrl
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Табела 4: Таблица истинитости излазног сигнала тростатичког инвертора.

ctrl	In	Out
0	0	Z
0	1	Z
1	0	1
1	1	0

имплементацији са слике 24. Управљачка јединица ћелије се своди на НИЛИ-И коло па се логички израз за ctrl сигнал у том случају записује као:

$$\text{ctrl} = \overline{(\text{Row Sel.} + \text{Col. Sel.})} \cdot \text{Row On}. \quad (48)$$

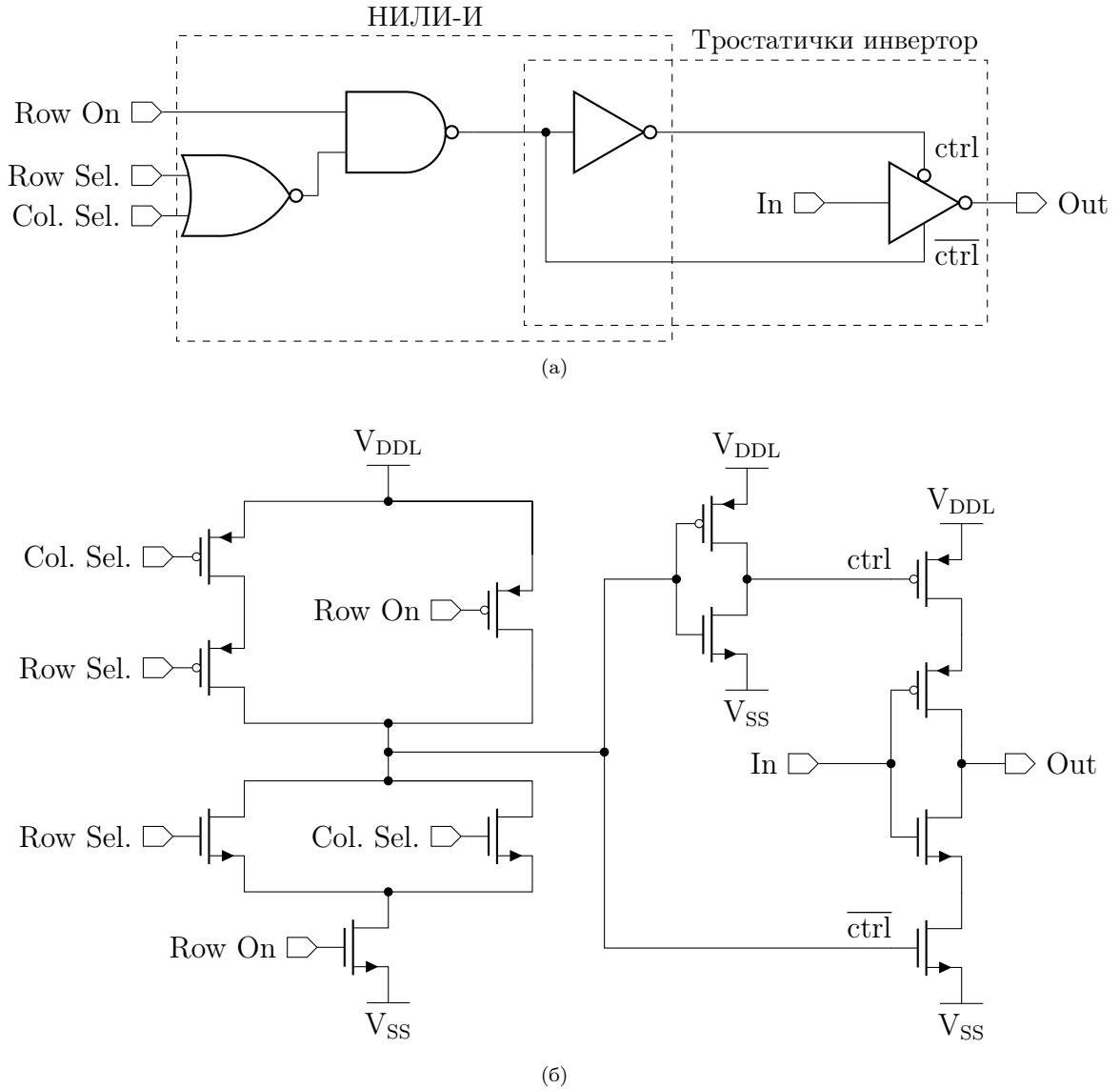
Таблица истинитости контролног сигнала ctrl у овом случају дата је у табели 5, док је таблица истинитости излазног сигнала приказана у табели 6.

Табела 5: Таблица истинитости контролног сигнала тростатичког инвертора активног на низак логички ниво управљачких сигнала.

Row On	Row Sel.	Col. Sel.	ctrl
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Табела 6: Таблица истинитости излазног сигнала тростатичког инвертора активног на низак логички ниво управљачких сигнала.

ctrl	In	Out
0	0	1
0	1	0
1	0	Z
1	1	Z

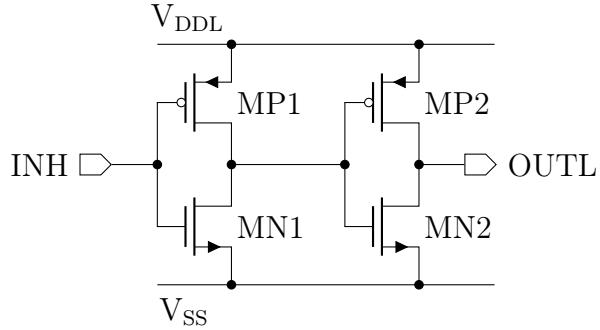


Слика 25: Шема ћелије DCO-а на нивоу (а) логичких кола и (б) транзистора за управљачке сигнале активне на низак логички ниво.

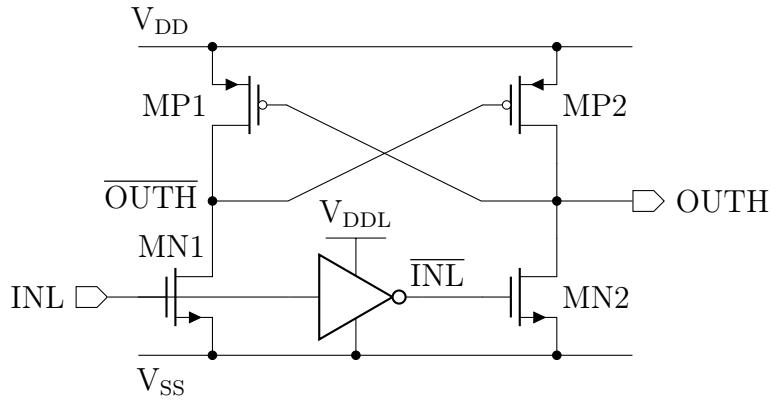
5.3 Претварачи напонског нивоа DCO-а

Напон напајања који користи дигитално контролисани осцилатор (V_{DDL}) се у овом раду разликује од напона напајања који користи остатак логике FLL-а (V_{DD}). Предност такве реализације је могућност подешавања напона напајања DCO-а независно након производње чипа, што даље омогућава накнадно постизање жељене резолуције учестаности (фреквенцијског корака) и фреквенцијског опсега зависно од процесног угла у коме се одвијала производња чипа. Додатна предност јесте и то што могућност смањења напона напајања DCO-а аутоматски доводи и до значајног смањења потрошње због њене квадратне зависности од напона напајања. Такав независан домен напајања DCO-а постигнут је додавањем претварача напонског нивоа на улазе и излазе DCO-а, као што је и приказано на слици 23.

Како је напон напајања DCO-а нижи од остатка система, на управљачке улазе DCO-а постављени су претварачи са високог на низак напонски ниво (енгл. *High-Low Level Shifter*, HLLS), док су на фазне излазе DCO-а постављени претварачи са ниског на



(a)



(b)

Слика 26: Шема претварача (а) са високог на низак и (б) са ниског на висок напонски ниво [17].

висок напонски ниво (енгл. *Low-High Level Shifter*, LHLS). Слика 26 приказује шеме конвенционалних претварача напонског нивоа који су коришћени у претходно описаном систему. Као што се може видјети са слике 26а, претварач са високог на низак напонски ниво није ништа друго до обичан бафер састављен од два инвертора чије напајање ће у коначној реализацији долазити од линије ниског напонског нивоа (V_{DDL}). Слика 26б приказује шему претварача са ниског на висок напонски ниво који представља регенеративно логичко коло засновано на позитивној повратној спрети [17] и састоји се од два унакрсно спрегнута PMOS транзистора (MP1 и MP2) и од NMOS транзистора (MN1 и MN2), који се покрећу преко комплементарних улазних сигналла INL и \overline{INL} , где је за инвертовање сигнална потребно додати инвертор.

С обзиром да је претварач са ниског на висок напонски ниво нешто комплекснији, његов рад ће бити детаљније објашњен. Када напони сигнална INL и \overline{INL} имају вриједност 0 и V_{DDL} , тада ће MN1 и MN2, редом, бити искључен и укључен. MN2 затим повлачи на нулу напон чвора OUTH, укључујући тако MP1. Тада се и напон чвора \overline{OUTH} подиже на V_{DD} , MP2 се искључује, а напон на чвору OUTH остаје на нули. С друге стране ако су вриједности напона на INL и \overline{INL} једнаки V_{DDL} и 0, тада ће MN1 и MN2 бити укључен и искључен, респективно. MN1 тада повлачи на нулу напон чвора \overline{OUTH} , укључујући тако MP2, чиме се напон OUTH подиже на V_{DD} , а MP1 се искључује.

Треба запазити да је напон чвора OUTH одређен покретачком струјом (енгл. *drive current*) транзистора MP2 и MN2, тј. струјом која тече кроз канал транзистора када

су укључени. Зато, ако је покретачка струја кроз транзистор MP2 много већа него кроз MN2, тада OUTH не може бити испражњен што доводи до тога да претварач не може исправно радити у таквим ситуацијама [17]. До њих иначе може доћи ако је разлика између V_{DD} и V_{DDL} превелика, што у примјени из овог рада није случај.

Претварачи напонских нивоа могу утицати на фактор испуњености импулса тј. периоде такта (енгл. *duty cycle*), поготово на излазу претварача са ниског на висок напонски ниво. Међутим, то се углавном може избећи додатним баферовањем излаза DCO-а тј. физичким уметањем одговарајућих бафера између фазног излаза DCO-а и претварача напонског нивоа чиме се фактор испуњености импулса такта може приближити идеалној вриједности од 50 %, како би као такав дошао на улаз претварача са ниског на висок напонски ниво.

6 Имплементација и резултати симулација

У овом поглављу биће приказана имплементација система кроз анализу SystemVerilog кода и хијерархијским приказом лејаута комплетног FLL-а. Затим ће бити приказани резултати симулација у различитим условима након екстракције паразитних капацитивности. Такође ће бити приказани резултати софтверског модела рада FLL-а, а биће направљено и поређење два DCO-а имплементирана у двије различите технологије. Дигитални FLL описан у овом раду, имплементиран је коришћењем SystemVerilog језика за опис хардвера у 130 nm CMOS технологији. Улазни референтни такт је 16 MHz, док се такт DCO-а подешава до 640 MHz, при чему је резолуција учестаности око 2,8 MHz у типичним условима рада.

6.1 SystemVerilog изворни код

У овом поглављу биће приказани најбитнији дијелови имплементације FLL-а на нивоу SystemVerilog [18] језика за опис хардвера. Уз поједине дијелове кода ће бити објашњења у сврху лакшег разумијевања. Листинг 1 приказује улазе и излазе FLL-а, који су побројани и у поглављу 2, са додатим сигналима за подешавање управљачке логике.

Листинг 1: Улазни и излазни сигнали (интерфејс) хијерархијски највиши FLL модула.

```
1 // Frequency-locked loop
2 module fll #(
3   parameter integer CW = 8 // frequency control word width
4 )(
5   /////////////////////
6   // Clock and reset //
7   /////////////////////
8   input clk_ref_i, // [0] reference clock @ 16MHz
9   input clk_ref_en_i, // [1] reference clock enable signal
10          // 0: reference clock propagation disabled
11          // 1: reference clock propagation enabled
12   input rst_ni, // [0] async reset
13          // 0: active/asserted
14          // 1: inactive/deasserted
15   ///////////////////
16   // FLL //
17   //////////////////
18   input [CW-1:0] clk_mult_i, // [0d40] reference frequency multiplier
19          // → (or frequency control word)
20          //           40 * 16MHz = 640MHz
21   ///////////////////
22   // DCO //
23   //////////////////
24   input dco_pwr_sel_i, // [1] DCO clock output power supply selection
25          // → (~1.1V,      adjustable)
26          //           0: FLL uses DCO clock output @ VDDL
27          //           1: FLL uses DCO clock output @ VDD
28          // → (~1.2V, non-adjustable)
```

```

26  input [1:0] dco_ctrl_mode_i, // [1] DCO control mode
27          //      0d0: FIXED      - DCO control
28  ↳ word is internally set to 0d203
29          //      0d1: MANUAL     - DCO control
30  ↳ word is set via "dco_ctrl_man_i" register
31          //      0d2: BANG-BANG - DCO control
32  ↳ word is set by a comparator and a counter
33          //      0d3: PID       - DCO control
34  ↳ word is PID regulated
35  input [CW-1:0] dco_ctrl_man_i, // [0d200] DCO control word value for
36  ↳ MANUAL mode
37  ///////////
38  // PID //
39  ///////////
40  input [CW-1:0] pid_kp_i, // [0b01000000] PID controller proportional
41  ↳ constant
42          //      A signed number in Q6.2
43  ↳ format (0b010000_00 == 0d16)
44  input [CW-1:0] pid_ki_i, // [0b01000000] PID controller integral
45  ↳ constant
46          //      A signed number in Q6.2
47  ↳ format (0b010000_00 == 0d16)
48  input [CW-1:0] pid_kd_i, // [0] PID controller derivative constant
49          //      A signed number in Q6.2 format
50  ///////////
51  // FLL //
52  ///////////
53  input [3:0] lock_cnt_i, // [0b1000] FLL internal counter lock
54  ↳ condition threshold
55  output lock_o, // [0] FLL lock signal - indicates if FLL achieved the
56  ↳ target frequency
57          //      0: the FLL did not its target frequency yet
58          //      1: the FLL achieved its target frequency
59  /////////////
60  // DSC output //
61  /////////////
62  output clk_dco_o // DCO clock - one phase
63  );
64  ...
65  endmodule

```

У поглављу 3.1 описана је улога и начин рада управљачке предобраде, а листинг 2 приказује хардверску имплементацију Грејевог бројача и начин како се генеришу узастопне вриједности бројача. Као што се из кода може видјети, оне се генеришу из наредне природне бинарне вриједности тако што се најстарији бит задржава, док су остали битови резултат бит по бит ексклузивно ИЛИ тј. ЕКСИЛИ (енгл. *XOR*) операције између свих битова осим најмлађег и свих битова осим најстаријег (линија 22 у листингу 2). Рецимо ако је наредна природна вриједност бројача 5 (бинарно 00000101, ако је ширина бројача 8), да би се претворила у Грејев код, задржава се најстарији бит, тј. 0, а осталих седам се додаје као резултат поменуте ЕКСИЛИ операције, тј. $0000010 \oplus 0000101$, што даје вриједност 0000111 коју треба додати на 0,

чиме се добија вриједност 00000111, која, ако погледамо табелу 1, одговара природној вриједности 5.

Листинг 2: Модул Грејевог бројача.

```

1  module cnt_gray #(
2    parameter int W = 8,
3    parameter byte Val = W'(0)
4  )(
5    input          clk_i,
6    input          rst_ni,
7    input          en_i,
8    output [W-1:0] cnt_gray_o
9  );
10
11   logic [W-1:0] cnt_bin;
12   logic [W-1:0] cnt_bin_next;
13   logic [W-1:0] cnt_gray;
14   assign cnt_bin_next = cnt_bin + W'(1);
15   always_ff @(posedge clk_i or negedge rst_ni) begin
16     if (!rst_ni) begin
17       cnt_bin <= Val;
18       cnt_gray <= {Val[W-1], Val[W-1:1] ^ Val[W-2:0]};
19     end else begin
20       if (en_i) begin
21         cnt_bin <= cnt_bin_next;
22         cnt_gray <= {cnt_bin_next[W-1], cnt_bin_next[W-1:1] ^
23           ↳ cnt_bin_next[W-2:0]};
24       end else begin
25         cnt_bin <= Val;
26         cnt_gray <= {Val[W-1], Val[W-1:1] ^ Val[W-2:0]};
27       end
28     end
29   assign cnt_gray_o = cnt_gray;
30
31 endmodule

```

Добијена вриједност у Грејевом коду се синхронизује кроз два флип-флопа и претвара се у природну бинарну вриједност, а имплементација тога је унутар `f11` модула и дата је у листингу 3. У поглављу 3.1.1 је детаљније објашњена улога и начин рада синхронизатора. Несинхронизована вриједност је на `cnt_dco_gray_d` магистрали и представља улаз у први од два флип-флопа синхронизатора, док су излази из првог и другог флип-флопа `cnt_dco_gray_q` и `cnt_dco_gray_q2`, респективно. Узоркована вриједност Грејевог бројача је на магистрали `cnt_dco_gray_q2`, док је природна бинарна вриједност на магистрали `cnt_dco` и она суштински представља тренутни умножак учестаности добијене на излазу DCO-а, и представља податак спреман за поређење са жељеним умношком са улаза, унутар блока управљачке логике. Сваки бит у тој магистрали се добија као резултат логичке ЕКСИЛИ операције над свим битовима Грејевог записа, помјереног онолико мјеста удесно колико износи тежина тренутног бита у `cnt_dco` магистрали (имплементација на линији 19 у листингу 3).

Листинг 3: Претварање Грејевог кода у природну бинарну вриједност.

```

1 module fll #(
2   ...
3   // Sample DCO cnt value
4   always_ff @(posedge clk_ref or negedge rst_clk_ref_n) begin
5     if (!rst_clk_ref_n) begin
6       cnt_dco_gray_q <= '0;
7       cnt_dco_gray_q2 <= '0;
8       cnt_dco_q      <= '0;
9     end else begin
10       cnt_dco_gray_q <= cnt_dco_gray_d;
11       cnt_dco_gray_q2 <= cnt_dco_gray_q;
12       cnt_dco_q      <= cnt_dco;
13     end
14   end
15   // Gray to binary conversion
16   genvar i;
17   generate
18     for(i=0; i<CW; i++) begin: gen_gray2bin_conv
19       assign cnt_dco[i] = ^(cnt_dco_gray_q2 >> i);
20     end: gen_gray2bin_conv
21   endgenerate
22   ...
23 endmodule

```

Двостепени регулатор, као један дио управљачке логике, је подробније објашњен у поглављу 3.2. Као што је тамо напоменуто, састоји се од компаратора и једног бројача. Компаратор служи за поређење добијеног и жељеног умношка учестаности, а резултат се шаље у бројач који на основи њега одређује како подесити управљачку ријеч за DCO. Листинг 4 приказује компараторски дио који се налази унутар fll модула. Као што се може закључити, `cnt_dco_q` представља семпловану вриједност тренутног умношка, односно вриједност `cnt_dco` из листинга 3. Сигнал `cnt_ref_up_dn` одређује да ли инкрементирати или декрементирати управљачку ријеч DCO-а, док сигнал `cnt_ref_en` одређује да ли уопште мијењати вриједност бројача (сигнал `lock_o` ће бити поменут при анализи имплементације ПИД регулатора). Закључује се да се поменути сигнали прослjeђују даље у блок бројача, чија је имплементација, као посебног `cntr` модула, приказана у листингу 5. Сигнали `cnt_ref_up_dn` и `cnt_ref_en` се при инстанцирању `cntr` модула прослијеђују на његове `up_down_i` и `en_i` улазе, респективно.

Листинг 4: Компараторски дио двостепеног регулатора.

```

1 module fll #(
2   ...
3   // Comparator
4   logic cnt_ref_up_dn;
5   logic cnt_ref_en;
6   assign cnt_ref_up_dn = (clk_mult_i >= cnt_dco_q) ? 1'b1 : 1'b0;
7   assign cnt_ref_en = (clk_mult_i != cnt_dco_q) ? ~lock_o : 1'b0;
8   ...
9 endmodule

```

Листинг 5: Имплементација бројача из двостепеног регулатора.

```

1 // Up-down counter with enable
2 module cntr #(
3   parameter int W = 8,      // count width (number of bits)
4   parameter byte Val = W'(0) // counter default value
5 ) (
6   // Clock and reset
7   input          clk_i,    // clock
8   input          rst_ni,   // async reset
9   // Control
10  input          en_i,     // enables counting
11  input          up_down_i, // count direction
12  // Count data output
13  output logic [W-1:0] cnt_o    // count output
14 );
15
16 always_ff @(posedge clk_i or negedge rst_ni) begin
17   if (!rst_ni) begin
18     cnt_o <= Val;
19   end else begin
20     if (en_i) begin
21       if (up_down_i == 1'b1) begin
22         cnt_o <= W'(cnt_o + W'(1));
23       end else begin
24         cnt_o <= W'(cnt_o - W'(1));
25       end
26     end else begin
27       cnt_o <= cnt_o;
28     end
29   end
30 end
31
32 endmodule

```

Имплементација ПИД регулатора, детаљније објашњеног у поглављу 3.3, приказана је у листингу 6, где се види да је имплементиран као посебан pid модул. Поред улаза такта, ресета и коефицијената регулатора, ту су још улази:

- setpoint_i који представља жељени умножак учестаности (при инстанцирању модула прослjeђује му се вриједност clk_mult_i са улаза f11 модула),
- feedback_i који представља узорковани тренутни умножак (при инстанцирању модула прослjeђује му се узоркована вриједност cnt_dco из листинга 3) и
- lock_cnt_i који означава колико узлазних ивица такта грешка мора бити једнака нули да би се достигло закључавање, односно да би се излазни сигнал lock_o подигао на 1.

Излаз ctrl_o представља податак који ће да служи као управљачка ријеч за DCO (аналогно излазу cnt_o из cnt модула двостепеног регулатора). Он означава 8-битну цјелобројну вриједност и добија се издавањем 8 бита од позиције 6 до позиције 13 из промјенљиве ctrl_pid (линија 64 у листингу 6) у којој је смјештена сума пропорцио-

налног, интегралног и диференцијалног дејства. Такво издавање битова је резултат коришћења бинарног бројевног система са фиксном тачком, о коме се више може наћи у литератури [9], а све то у сврху повећања резолуције подешавања ПИД регулатора. Наиме, улазни коефицијенти `kp_i`, `ki_i` и `kd_i` су 8-битни у формату Q6.2 (6 бита за цјелобројни дио, 2 за разломљени дио), док је формат `ctrl_pid` промјенљиве Q12.6 (12 битова за цјелобројни дио, 6 за разломљени дио), па се поменутим одсијецањем врши заокруживање на 8-битни цијели број.

Листинг 6: Имплементација ПИД регулатора.

```

1 // PID controller
2 module pid #(
3     parameter int DW = 8,    // data input width
4     parameter int CW = 8    // control output width
5 ) (
6     input                  clk_i,
7     input                  rst_ni,
8     input      [DW-1:0] kp_i,
9     input      [DW-1:0] ki_i,
10    input      [DW-1:0] kd_i,
11    input      [DW-1:0] setpoint_i,
12    input      [DW-1:0] feedback_i,
13    input      [3:0]   lock_cnt_i,
14    output logic [CW-1:0] ctrl_o,
15    output logic       lock_o
16 );
17
18 // Localparam
19 localparam int ErrCntW = 4;
20
21 // Error logic
22 logic signed [DW-1:0] err;
23 logic signed [DW-1:0] err_prev;
24 logic signed [DW-1:0] err_diff;
25 assign err = setpoint_i - feedback_i;
26 assign err_diff = err - err_prev;
27 logic [ErrCntW-1:0] err_eq0_cnt;
28 logic [ErrCntW-1:0] err_cnt_lock;
29 assign err_cnt_lock = ErrCntW'(lock_cnt_i);
30 always_ff @(posedge clk_i or negedge rst_ni) begin
31     if (!rst_ni) begin
32         err_eq0_cnt <= '0;
33     end else begin
34         if ((err == '0) && (err_eq0_cnt < err_cnt_lock)) begin
35             err_eq0_cnt <= err_eq0_cnt + ErrCntW'(1);
36         end
37     end
38 end
39
40 // PID calculation & lock
41 logic signed [CW+DW-1:0] prop;
```

```

42 logic signed [CW+DW : 0] integ;
43 logic signed [CW+DW-1:0] deriv;
44 logic signed [CW+DW+1:0] ctrl_pid;
45 always_ff @(posedge clk_i or negedge rst_ni) begin
46   if (!rst_ni) begin
47     prop      <= '0;
48     integ     <= '0;
49     deriv     <= '0;
50     err_prev <= '0;
51     ctrl_pid <= '0;
52     lock_o    <= '0;
53   end else begin
54     prop      <= (kp_i * err);
55     integ     <= integ + (ki_i * err);
56     deriv     <= (kd_i * err_diff);
57     err_prev <= err;
58     if ((err == '0) && (err_eq0_cnt == err_cnt_lock) && (
59       → err_cnt_lock != '0)) begin
60       lock_o <= 1'b1;
61     end
62     ctrl_pid <= lock_o ? ctrl_pid : 18'(prop) + integ + 18'(
63       → deriv);
64   end
65   assign ctrl_o = ctrl_pid[13:6];
66 endmodule

```

Излази `cnt_o` и `ctrl_o` из двостепеног и ПИД регулатора, редом, представљају суштински број тростатичких инвертора DCO-а који требају бити укључени, што утиче на његову излазну учестаност. Међутим, та вриједност се не прослjeђује директно у модул DCO-а, већ се преко управљачког декодера анализираног у поглављу 3.4 претвара у скуп одговарајућих векторских улаза за управљање DCO-ом тј. врстама и колонама тростатичких инвертора унутар њега. Тада поступак претварања је приказан у листингу 7, а имплементиран је у оквиру `fll` модула.

Листинг 7: Имплементација управљачког декодера из поглавља 3.4.

```

1 module fll #(
2   ...
3   // Convert digital code to real DCO inputs (row/col sel and row on)
4   localparam int NumRows = 17;
5   localparam int NumCols = 15;
6   logic [NumRows-1:0] row_on_d;
7   logic [NumRows-1:0] row_sel_d;
8   logic [NumCols-1:0] col_sel_d;
9   always_comb begin
10     row_on_d  = '0;
11     row_sel_d = '0;
12     col_sel_d = '0;
13     for(int r = 1; r <= NumRows; r = r + 1) begin

```

```

14     // Row on logic
15     if (int'(dco_ctrl) > r*NumCols) begin
16         row_on_d[r-1] = 1'b1;
17     end else begin
18         row_on_d[r-1] = 1'b0;
19     end
20     // Row & column select logic
21     if ((int'(dco_ctrl) > (r-1)*NumCols) && (int'(dco_ctrl)
22     → <= r*NumCols)) begin
23         for(int c = 0; c < NumCols; c = c + 1) begin
24             if ((r-1)*NumCols + c < int'(dco_ctrl)) begin
25                 col_sel_d[c] = 1'b1;
26             end else begin
27                 col_sel_d[c] = 1'b0;
28             end
29         end
30         row_sel_d[r-1] = 1'b1;
31     end else begin
32         row_sel_d[r-1] = 1'b0;
33     end
34 end
35 ...
36 endmodule

```

Сигнали `row_on_d`, `row_sel_d` и `col_sel_d` се синхронизују на одговарајући такт и прослjeђују се као улази у DCO модул који је имплементиран као посебан блок у алатима за аналогно пројектовање, па стога овде нема свој опис у SystemVerilog-у, већ се тај блок само инстанцира унутар `fll` модула, као што је приказано у листингу 8. Кao што се може видjetи, `dco` модул садржи излазе за свих 5 фаза на ниском (V_{DDL}) и високом (V_{DD}) напонском нивоу, а који од та два напонска нивоа ће се наћи на `clk_dco_o` излазу `fll` модула одређује његов улазни сигнал `dco_pwr_sel_i` (погледати листинг 1).

Листинг 8: Инстанцирање дигитално контролисаног осцилатора.

```

1 module fll #(
2     ...
3     // DCO model
4     logic clk_dco_ph0;
5     logic clk_dco_ph0_l;
6     logic clk_dco_ph0_h;
7     logic clk_dco_ph1;
8     logic clk_dco_ph1_l;
9     logic clk_dco_ph1_h;
10    logic clk_dco_ph2;
11    logic clk_dco_ph2_l;
12    logic clk_dco_ph2_h;
13    dco u_dco(
14        .ROW_ON      (row_on),
15        .ROW_SEL    (row_sel),

```

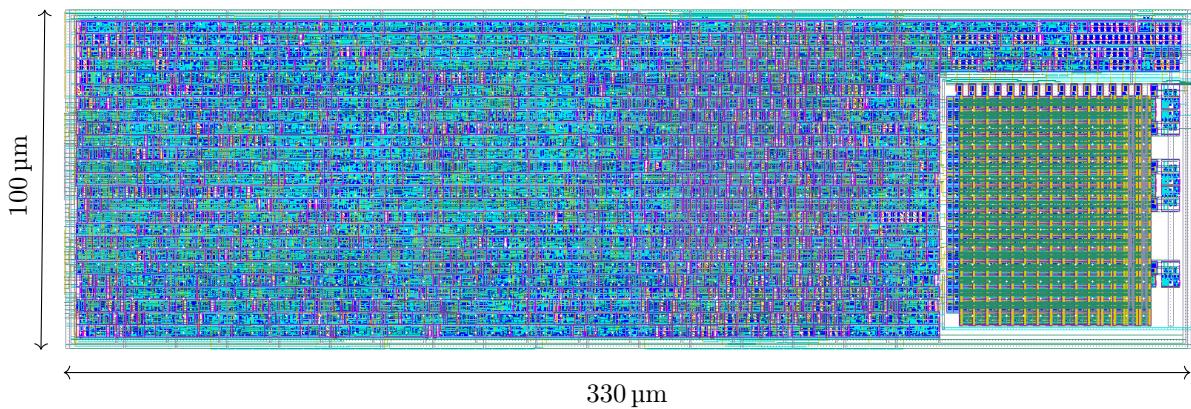
```

16   .COLUMN_SEL(col_sel),
17   .PH0      (clk_dco_ph0_h),
18   .PH0_L    (clk_dco_ph0_l),
19   .PH1      (clk_dco_ph1_h),
20   .PH1_L    (clk_dco_ph1_l),
21   .PH2      (clk_dco_ph2_h),
22   .PH2_L    (clk_dco_ph2_l),
23   .PH3      (),
24   .PH3_L    (),
25   .PH4      (),
26   .PH4_L    ()
27 );
28 assign clk_dco_ph0 = dco_pwr_sel_i ? clk_dco_ph0_h :
29   ↪ clk_dco_ph0_l;
30 assign clk_dco_ph1 = dco_pwr_sel_i ? clk_dco_ph1_h :
31   ↪ clk_dco_ph1_l;
32 assign clk_dco_ph2 = dco_pwr_sel_i ? clk_dco_ph2_h :
33   ↪ clk_dco_ph2_l;
34 assign clk_dco_o = clk_dco_ph2;
35 ...
36
37 endmodule

```

6.2 Лејаут

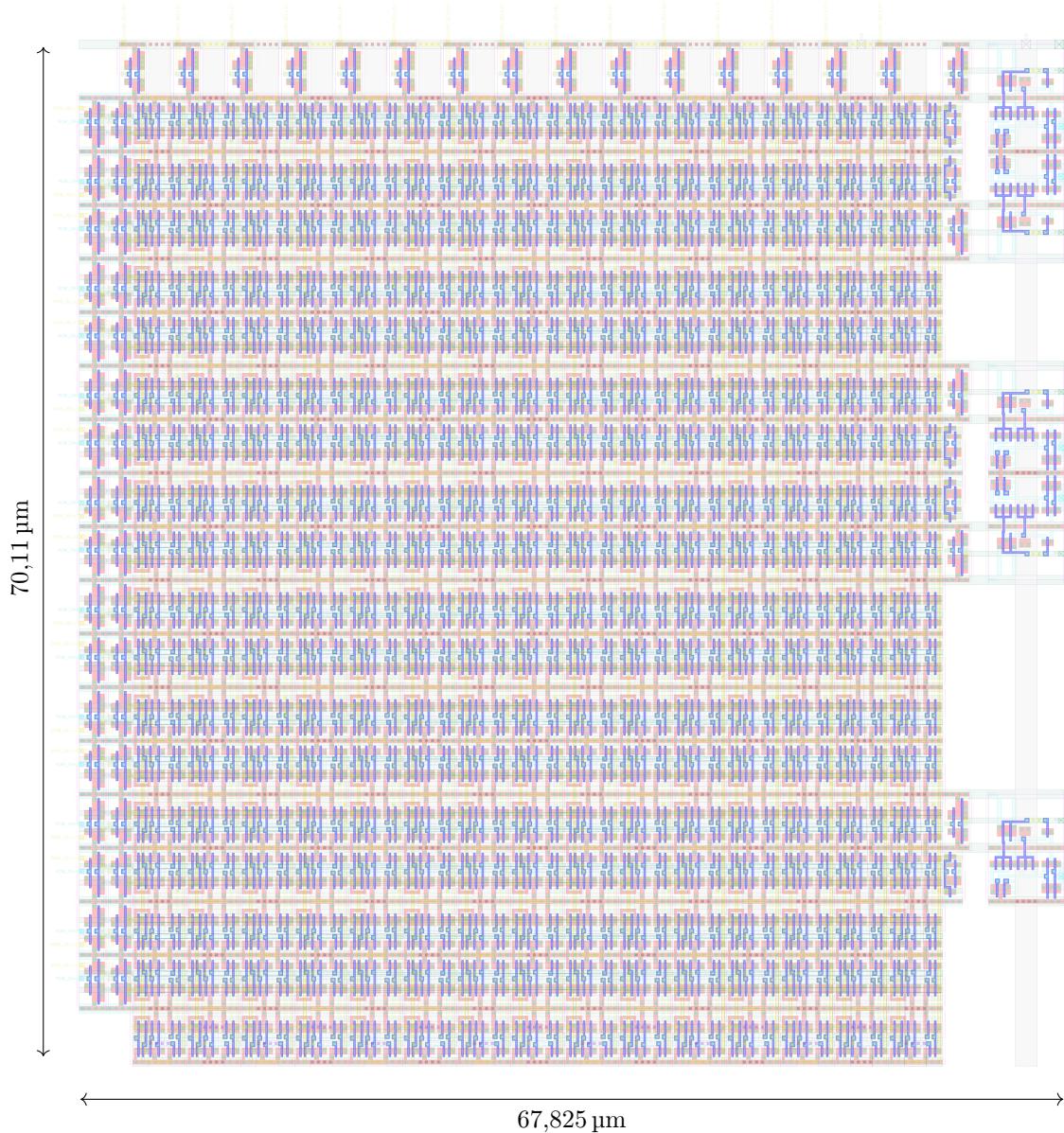
За израду лејаута намјенски је коришћена 7T (енгл. 7-Track) библиотака стандардних ћелија, где 7T означава да висина свих ћелија тачно толика да може да се вертикално провуче 7 линија метала минималне ширине и са минималним размаком. То доприноси лакшем аутоматском (а и ручном) рутирању и слагању ћелија у компактну цјелину у сврху оптималне искористивости простора. Лејаут коначне верзије цјелокупног FLL-а приказан је на слици 27. Читав дизајн FLL-а заузима $33000 \mu\text{m}^2$. Компонента у доњем десном углу је дигитално контролисани осцилатор, док остатак лејаута углавном припада управљачкој логици FLL-а.



Слика 27: Лејаут дигиталног FLL-а, са DCO-ом у доњем десном углу, у 130 nm CMOS технологији.

DCO је имплементиран као независна компонента коришћењем библиотека стандардних ћелија. Од читавог лејаута FLL-а, око 13 % заузима DCO, тачније око $4300 \mu\text{m}^2$ (вриједност добијена сабирањем појединачних површина коришћених ћелија). Слика 28 приказује лејаут DCO-а са претварачима напонских нивоа који се могу уочити

са лијеве и горње стране лејаута (претварачи са високог на низак напонски ниво на управљачким улазима DCO-а), као и са десне стране (претварачи са ниског на висок напонски ниво на свих 5 фаза DCO-а).

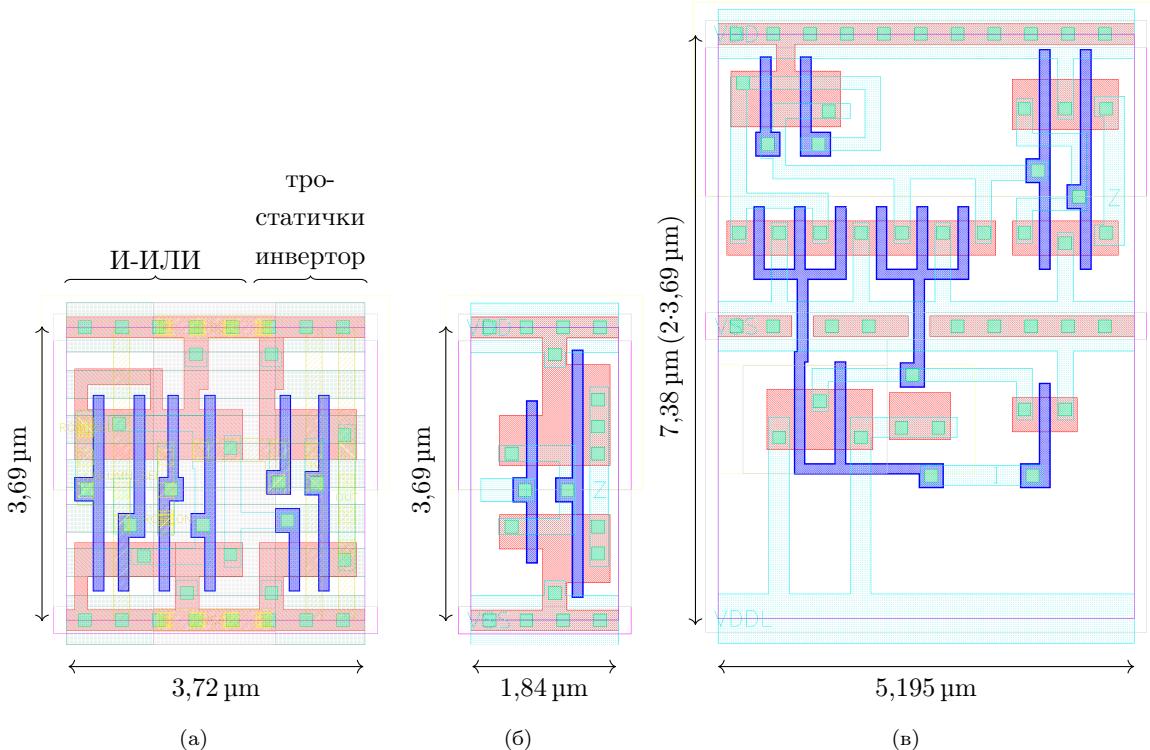


Слика 28: Лејаут дигитално контролисаног осцилатора са претварачима напонских нивоа у 130 nm CMOS технологији.

слика 29а приказује лејаут једне ћелије DCO-а који одговара шеми ћелије DCO-а са слике 24. Са десне стране ћелије је оријентисан тростатички инвертор, док је са лијеве стране управљачки И-ИЛИ дио. Површина једне ћелије је око $13,7 \mu\text{m}^2$.

Лејаути претварача са високог на низак и са ниског на висок напонски ниво су приказани на сликама 29б и 29в, респективно. Они редом одговарају шемама претварача са слика 26а и 26б. Као што је већ напоменуто у поглављу 5.3, претварач са високог на низак напонски ниво је у суштини бафер и овдје је његова површина око $6,8 \mu\text{m}^2$, док је претварач са ниског на висок напонски ниво нешто сложенији и, ради бољег уклапања тј. заузимања мање површине у комбинацији са DCO-ом, имплементиран

је као ћелија са дуплом висином, што омогућава мању ширину ћелије, а површина јој је око $38,3 \mu\text{m}^2$.



Слика 29: Лејаут (а) ћелије DCO-а, (б) претварача са високог на низак и (в) претварача са ниског на висок напонски ниво, у 130 nm CMOS технологији.

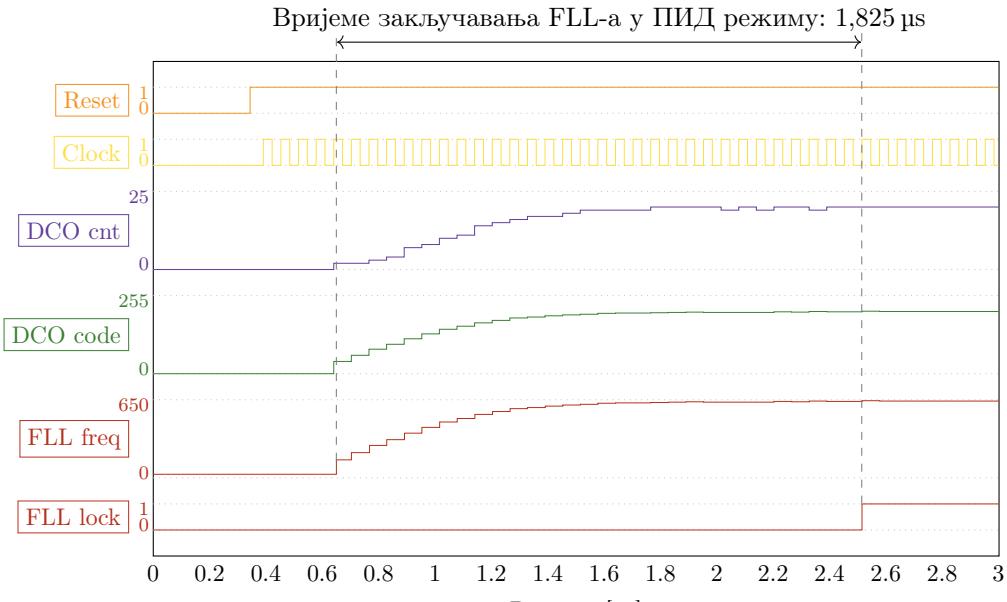
6.3 Симулација рада фреквенцијски затворене петље

Симулација рада FLL-а у оба управљачка режима приказана је на слици 30. Са приказаних графика се може видјети да се у оба управљачка режима достиже стабилно закључавање FLL-а са излазном учестаности веома близкој траженој учестаности, при чему грешка може бити у опсегу просјечне вриједности резолуције DCO-а (испод 3 MHz). Међутим, иако се у оба режима исправно достиже жељена учестаност, између њих постоји разлика у брзини достизања исте. Тако, у ПИД режиму ($\Pi=15$, $I=15$, $D=0$), закључавање се одвија много брже, и то за $1,83 \mu\text{s}$, док је, поређења ради, за закључавање у режиму са двостепеним регулатором потребно $13,13 \mu\text{s}$.

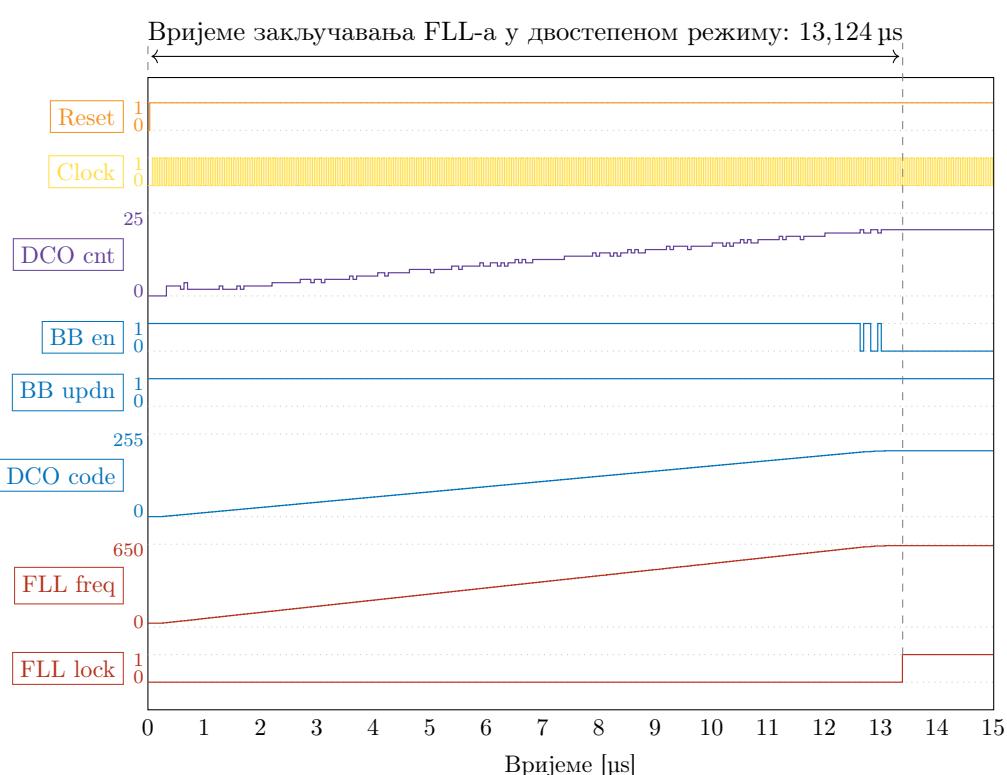
6.4 PVT зависност дигитално контролисаног осцилатора

Да би се одредило понашање DCO-а у реалним условима, потребно је проверити понашање DCO-а при промјени процесних углова, напона напајања и температуре, односно проверити различите PVT (енгл. *Process-Voltage-Temperature*) случајеве. У оквиру овог рада симулирано је понашање у три различита PVT случаја:

1. Најспорији случај (slow-slow процесни угао, $V_{DD}=1,1 \text{ V}$, $V_{DDL}=1 \text{ V}$, 125°C)
2. Типични случај (typical-typical процесни угао, $V_{DD}=1,2 \text{ V}$, $V_{DDL}=1,1 \text{ V}$, 40°C)
3. Најбржи случај (fast-fast процесни угао, $V_{DD}=1,3 \text{ V}$, $V_{DDL}=1,2 \text{ V}$, -40°C)



(a)

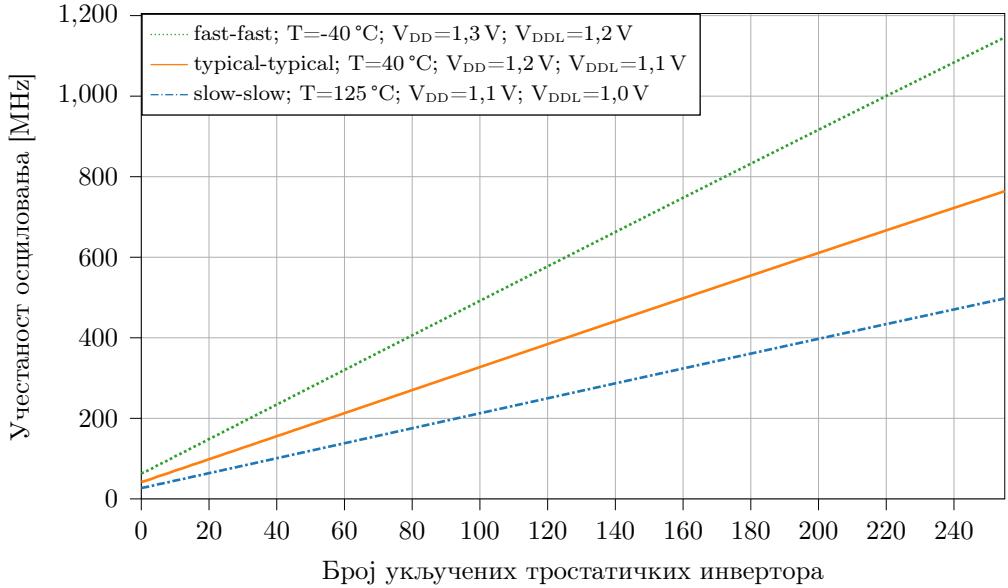


(б)

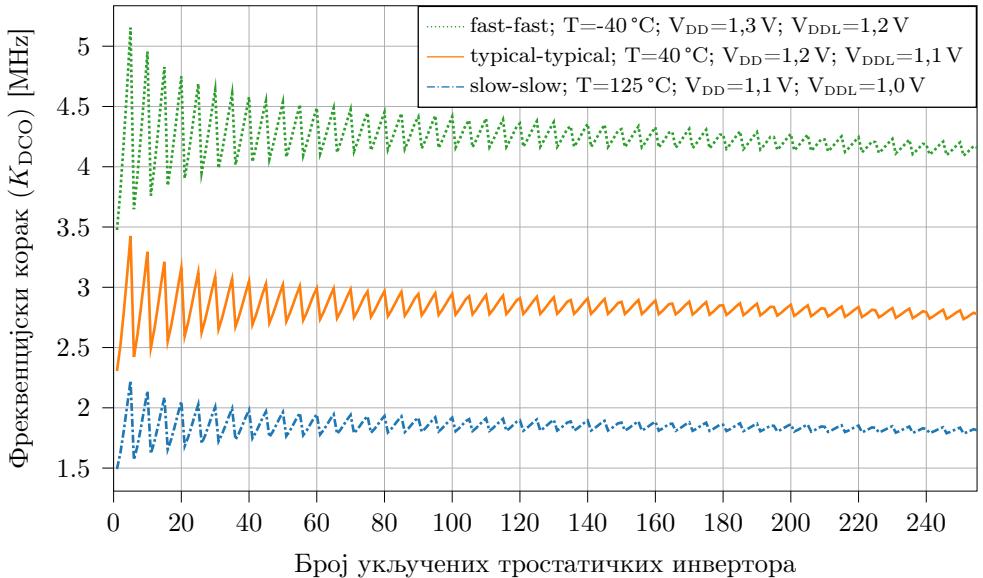
Слика 30: Симулација FLL-а у (а) ПИД и (б) двостепеном режиму.

Слика 31 приказује зависност учестаности DCO-а и фреквенцијског корака тј. резолуције учестаности (K_{DCO}) од броја укључених тростатичких инвертора (тј. од управљачке ријечи DCO-а), за најспорији, типичан и најбржи PVT случај.

Табела 7 приказује фреквенцијски опсег DCO-а и просјечну вриједност фреквенцијског корака (K_{DCO}) у три претходно обрађена случаја. Под опсегом се подразуми-



(a)



(б)

Слика 31: Зависност (а) учестаности осциловања и (б) фреквенцијског корака од броја укључених тростатичких инвертора за најспорији, типични и најбржи случај.

јевају минималне и максималне вриједности учестаности која се добија на излазу DCO-а за укључен минималан (у овом случају 15) и максималан број (у овом случају 270) тростатичких инвертора DCO-а, респективно. Иако у најспоријем случају

Табела 7: Фреквенцијски опсег DCO-а и просјечна вриједност фреквенцијског корака (K_{DCO}).

Случај	f_{\min}	f_{\max}	Просјечан K_{DCO}
Најспорији	27,2 MHz	502 MHz	1,8 MHz
Типични	42 MHz	764 MHz	2,8 MHz
Најбржи	63,3 MHz	1,146 GHz	4,2 MHz

учестаност не достиже жељених 640 MHz, она ипак може бити достигнута повећа-

њем засебног напона напајања осцилатора (V_{DDL}). Такође је могуће анализирати и зависност учестаности од свих PVT параметара независно, међутим те зависности нису приказане у раду због временске захтјевности извршавања потребних симулација, као и због већ приказаних резултата симулација у најспоријем, типичном и најбржем случају, који су ипак најбитнији за анализу понашања имплементираног система јер се ослањају на понашање система у граничним условима рада. Ипак, што се тиче понашања DCO-а у зависности од свих PVT параметара појединачно, потребно је напоменути да се, у овој конкретној технологији, учестаност повећава са бржим процесним углом, већим напоном напајања, као и мањом температуром.

Да би се значајно смањило вријеме потребно за добијање резултата из симулација након екстракције лејаута, и то са минималним губитком прецизности добијених резултата, симулације DCO-а су извршене на нивоу појединачно екстрактованих ћелија DCO-а и екстрактованих претварача напонског нивоа, са додатним паразитним кондензатором на свакој фази DCO-а. Улога тих кондензатора је да надомјесте паразитне капацитивности које се јављају у потпуно екстракованом лејауту DCO-а због међусобно повезаних ћелија, а које нису аутоматски урачунате ако се симулације врше на нивоу појединачно екстрактованих компоненти DCO-а, као што је то овде случај. Вриједност капацитивности додатог кондензатора прорачуната је на основу извјештаја генерисаног из екстракције паразитних капацитивности на нивоу читавог дизajна.

6.5 Анализа потрошње система

Потрошња снаге сваког дизајна утиче на велики број критичних одлука при самом пројектовању, као што су одређивање капацитивности извора напајања, трајања батерије, величине линија напајања, начина паковања, захтјева за хлађењем система итд. Значи да је потрошња снаге важна особина дизајна јер значајно утиче на изводљивост примјене крајњег производа, као и на трошкове и поузданост при коришћењу [15].

Зависно од проблема који се анализира, разматрају се различите мјере потрошње снаге. Рецимо, када се разматра величина линија напајања, тада је битна вршна вриједност снаге (енгл. *peak power*), P_{peak} , док при разматрању захтјева за хлађењем или складиштењем енергије тј. батеријом, од главног интереса је просјечна потрошња снаге (енгл. *average power dissipation*) [15]. Те мјере су дефинисане следећим формулама:

$$P_{peak} = i_{peak} V_{supply} = \max[p(t)] \quad (49)$$

$$P_{avg} = \frac{1}{T} \int_0^T p(t) dt = \frac{V_{supply}}{T} \int_0^T i_{supply}(t) dt \quad (50)$$

где је i_{supply} струја која долази од извора са напоном V_{supply} у интервалу $t \in [0, T]$, док је i_{peak} максимална вриједност i_{supply} унутар тог интервала.

Од горе поменутих, у овом раду је најбитнија просјечна потрошња снаге. Како је DCO кључни елемент FLL-а, он је и убедљиво највећи потрошач у систему. Самим тим, његовим изолованим симулацијама се може умногоме предвидјети и потрошња коначног система, што значи да се пројектовање убрзава тиме што није потребно баш увијек радити симулације дизајна читавог FLL-а како би се приближно одредила потрошња.

Све симулације потрошње које слиједе су извршаване за учестаност осциловања од 640 MHz. Након симулације DCO-а (са претварачима напонских нивоа) у типичним условима рада, при чemu су $V_{DD}=1,2\text{ V}$ и $V_{DDL}=1,1\text{ V}$, добија се вриједност просјечне потрошње од $2,5\text{ mW}$ од извора напајања V_{DDL} одакле се напајају осцилатор, претварачи са високог на низак и једним дијелом претварачи са ниског на висок напонски ниво и $0,22\text{ mW}$ од извора напајања V_{DD} одакле се, у овој симулацији, већим дијелом напајају претварачи са ниског на висок напонски ниво као и излазни бафери (погледати поглавље 5.3).

Потрошња се може утврдити и преко средње квадратне (енгл. *Root Mean Square*, RMS) вриједности струје која се добија при истим условима и овде износи $2,185\text{ mA}$ и $0,18\text{ mA}$ од извора напајања V_{DDL} и V_{DD} , респективно. С обзиром да је средња квадратна вриједност струје величина директно сразмерна просјечној потрошњи снаге, при анализи потрошње углавном је довољно навести резултате у једној од тих величин. Формула за RMS вриједност потрошње струје неког извора гласи:

$$i_{\text{rms}} = \sqrt{\frac{1}{T} \int_0^T i_{\text{supply}}^2(t) dt} \quad (51)$$

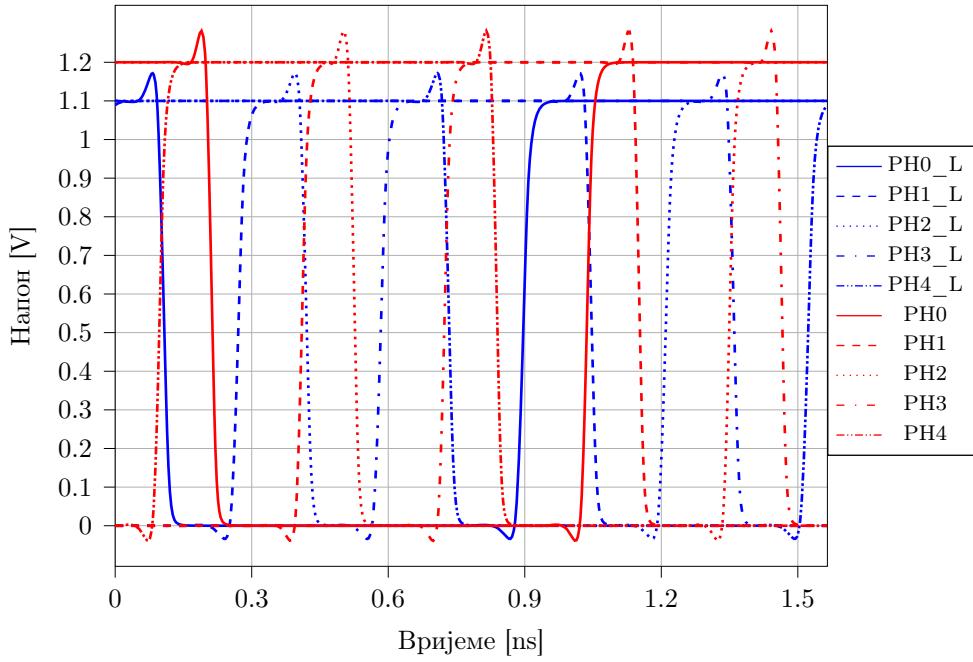
Резултати симулације наведени у претходном параграфу, од читавог FLL-а, обухватају DCO са претварачима напонских нивоа. Да би се утврдила и потрошња управљачке логике (која се напаја са извора V_{DD}), под истим условима је покренута и симулација потрошње читавог FLL-а, где се на изворима напајања V_{DDL} и V_{DD} добијају RMS вриједности потрошње струје $2,185\text{ mA}$ и $0,9\text{ mA}$, респективно. Ако се од укупне RMS вриједности струје са V_{DD} извора напајања од $0,9\text{ mA}$ одузме изнад поменута вриједност коју троше само претварачи са ниског на висок напонски ниво и излазни бафери DCO-а ($0,18\text{ mA}$), добије се RMS вриједност потрошње струје управљачке логике FLL-а која износи $0,72\text{ mA}$.

6.6 Временски одзив дигитално контролисаног осцилатора

Временски одзив (енгл. *timing*) односи се на облик сигнала на фазама дигитално контролисаног осцилатора на нивоу једне периоде излазног такта. У сврху приказа временског одзыва симулиран је рад екстракованог DCO-а у типичним условима.

Слика 32 приказује генерисани сигнал такта на свакој од пет фаза DCO-а. Приказани сигнали су добијени са улаза (V_{DDL} домен напона напајања) и излаза (V_{DD} домен напона напајања) претварача са ниског на висок напонски ниво. Као што је приказано у једначини (43), учестаност осциловања f_{osc} зависи од t_d тј. кашњења кроз ћелију DCO-а која садржи тростатички инвертор. Та вриједност је практично једнака аритметичкој средини кашњења узлазне и силазне ивице сигнала такта (t_{lh} и t_{hl}). Други важан параметар за разматрање је фактор испуњености импулса такта тј. однос времена које излазни сигнал проведе на високом, односно ниском, напону у једној периоди такта и самог периода такта. Изражава се у процентима, тј. колико процентуално сигнал такта проведе на високом, а колико на ниском напону, тј. нули и идеално је да тај однос буде $50\% - 50\%$.

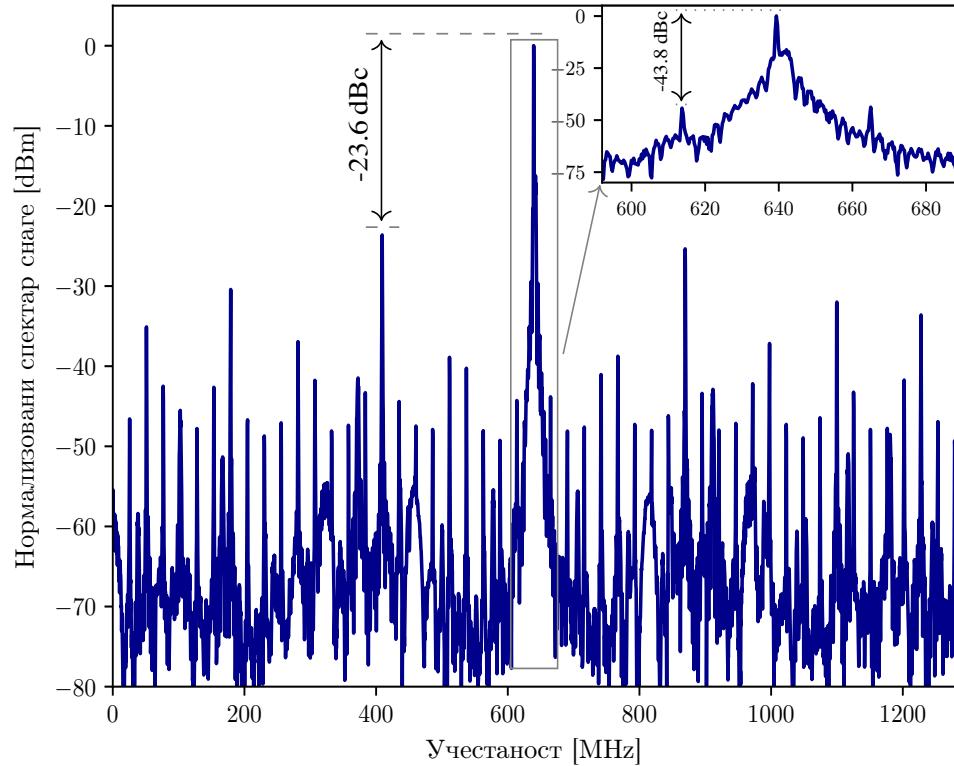
Добијене вриједности кашњења узлазне и силазне ивице такта на баферованом излазу претварача са ниског на висок напонски ниво су $t_{lh} = 26\text{ ps}$ и $t_{hl} = 20\text{ ps}$, респективно. Добијени однос унутар радног циклуса такта је апроксимативно $48\% - 52\%$. Временски помјерај између двије сусједне фазе је око 940 ps .



Слика 32: Генерисани такт на свих 5 фаза DCO-а у V_{DD} и V_{DDL} доменима напајања.

6.7 Спектар снаге фреквенцијски затворене петље

Нормализовани спектар снаге предложеног FLL-а у ПИД режиму, током процеса одржавања вриједности учестаности од 640 MHz на излазу, приказан је на слици 33.



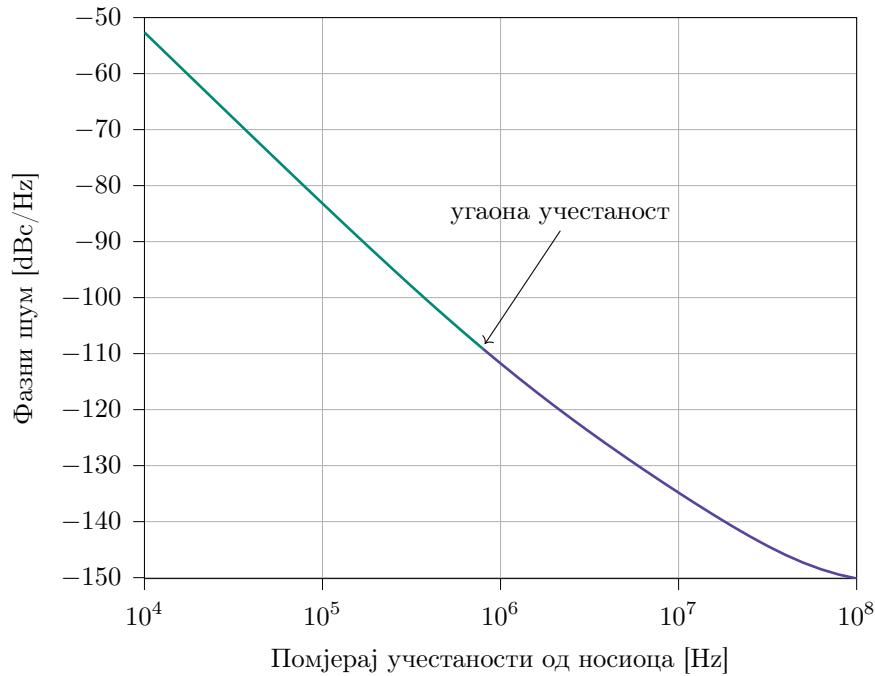
Слика 33: Спектар снаге FLL-а, са учестаношћу носиоца 640 MHz, најјачим нивоом непожељног сигнала (у средини лијево) на -23,6 dBc, и највећим непожељним сигналом референтне учестаности унутар 640 ± 16 MHz (горе десно, зумирено) на нивоу -43,8 dBc.

Подаци са графика су добијени из дигиталне симулације на нивоу логичких кола, трајања $15\text{ }\mu\text{s}$, која укључује идеалан бешумни модел DCO-а написан у SystemVerilog-у. Излаз симулације је потпуно детерминистички квадратни талас састављен од једињица и нула, узоркован са учестаношћу 10 GHz , и сачуван за даљу обраду.

Највећа вриједност на графику представља учестаност осциловања FLL-а, односно учестаност носиоца (енгл. *carrier frequency*). Ниво највећег непожељног тона је $-23,6\text{ dBc}$, што је више од два реда величине испод нивоа учестаности носиоца (0 dBc). У увеличаној области, може се видјети непожељан сигнал унутар референтне учестаности и то лијево и десно од носиоца са помјерајем (енгл. *offset*) од 16 MHz , чији је најјачи ниво на $-43,8\text{ dBc}$.

6.8 Фазни шум дигитално контролисаног осцилатора

Због насумичних фазних одступања, спектар снаге реалног осцилатора се такође шири на учестаности око учестаности носиоца, f_{osc} . Извори насумичних фазних поремешћаја [16] у виду шума треперења и термичког шума се манифестишу као $1/f^3$ и $1/f^2$ области, респективно, у добијеном профилу фазног шума на слици 34. Приказани резултати су добијени у типичним PVT условима, са вриједношћу напона напајања DCO-а (V_{DDL}) од $1,1\text{ V}$, и вриједности учестаности носиоца (f_{osc}) од 640 MHz .



Слика 34: Спектар фазног шума DCO-а: област шума треперења (лијево), област термичког шума (десно), и углона учестаност од $\approx 800\text{ kHz}$.

7 Пајтон модел FLL-а

У сврху практичнијег симулирања и бољег предвиђања понашања FLL-а при различитим условима рада, направљен је основни Пајтон модел цјелокупног FLL-а. Предност посједовања таквог модела јесте што се одређене карактеристике конкретног дизајна могу параметризовати и затим је, у зависности од тога колико услова модел покрива, могуће веома брзо добити мање или више вјеродостојан резултат понашања тог дизајна, умјесто да се, с друге стране, мијења дизајн и додатним симулацијама утврђују карактеристике под различитим условима, па тек затим пређе на симулације крајњег понашања, што доста успорава процес пројектовања. Конкретна имплементација Пајтон модела FLL-а из овог рада обухвата, поред параметара самог FLL-а, и параметризовану почетну учестаност DCO-а и његову резолуцију, што омогућава тренутно добијање резултата симулације. Треба напоменути да направљени модел предвиђа константне услове рада тј. параметри модела се не мијењају током времена, односно кроз итерације у петљи модела, и да модел не покрива све услове рада. То значи да резултати приказани у наставку не морају да одговарају резултатима из RTL симулација или симулација на нивоу логичких кола, већ су ту да пројектанта наведу на исправну употребу одређених константи (као што су ПИД константе) или да му помогну око одабира компоненти дизајна чије појединачно понашање му је већ познато и може да се параметризује (рецимо понашање DCO-а).

7.1 Пајтон код

У наставку ће бити приказан изворни Пајтон код модела, са логиком и називима промјенљивих који углавном одговарају онима из SystemVerilog хардверске имплементације. Биће наведени само листинзи, без детаљног објашњења кода, јер је већина принципа рада већ анализирана при опису хардверске имплементације. Прегледности ради, код је подијељен у класе и то:

- **fll** - Класа FLL компоненте (листинг 9).
- **pid** - Класа ПИД регулатора (листинг 10).
- **cntr** - Класа бројача из двостепеног регулатора (листинг 11).
- **dco** - Класа дигитално контролисаног осцилатора (листинг 12).

С обзиром да **dco** класа нема хардверску имплементацију у виду SystemVerilog модула, треба напоменути да се логика имплементације софтверског модела ослања на једначину (46) што се може видjetи у листингу 12.

Листинг 9: Имплементација модела FLL компоненте.

```
1 from dco import dco
2 from cntr import cntr
3 from pid import pid
4
5 class fll:
6     def __init__(self,
7                  # Inputs from design
8                  CW,                      # control word width
9                  clk_ref_i,            # reference clock
```

```

10         clk_ref_en_i,    # reference clock enable
11         rst_ni,          # async reset
12         clk_mult_i,      # reference frequency multiplier [40]
13         dco_pwr_sel_i,   # DCO clock output power supply
14     ↪ selection: VDDL/VDD [0/1]
15             dco_ctrl_man_i,    # manual DCO control word [200]
16             dco_ctrl_mode_i,   # DCO control mode: FIXED/MANUAL/CNT
17     ↪ /PID [PID=3/CNT=2]
18         pid_kp_i,        # PID controller proportional constant [0
19     ↪ b01000000]
20         pid_ki_i,        # PID controller integral constant [0
21     ↪ b01000000]
22         pid_kd_i,        # PID controller derivative constant [0]
23         lock_cnt_i,      # FLL lock counter condition [0b1000]
24
25             # Additional input parameters
26             k_dco,           # DCO frequency resolution
27             clk_dco_init):   # DCO initial frequency (dco_ctrl=0)
28         """
29
30     Frequency-locked loop (FLL) is a negative feedback control system
31     ↪ that locks the output frequency signal to the target frequency.
32     ↪ Principally, it continuously controls the oscillator frequency in an
33     ↪ automated way until its output frequency matches the target value,
34     ↪ after which that value is maintained at the output.
35         """
36
37     # Inputs from design
38     self.CW = CW
39     self.clk_ref_i = clk_ref_i
40     self.clk_ref_en_i = clk_ref_en_i
41     self.rst_ni = rst_ni
42     self.clk_mult_i = clk_mult_i
43     self.dco_pwr_sel_i = dco_pwr_sel_i
44     self.dco_ctrl_man_i = dco_ctrl_man_i
45     self.dco_ctrl_mode_i = dco_ctrl_mode_i
46     self.pid_kp_i = pid_kp_i
47     self.pid_ki_i = pid_ki_i
48     self.pid_kd_i = pid_kd_i
49     self.lock_cnt_i = lock_cnt_i
50
51     # Additional input parameters
52     self.k_dco = k_dco
53     self.clk_dco_init = clk_dco_init
54
55     # Component instantiation
56     self.dco_inst = dco(k_dco=self.k_dco,
57                         init_freq=self.clk_dco_init)
58     self.cntr_inst = cntr()
59     self.pid_inst = pid(rst_ni=self.rst_ni,
60                         kp_i=self.pid_kp_i,
61                         ki_i=self.pid_ki_i,
62                         kd_i=self.pid_kd_i,

```

```

53                                     setpoint_i=self.clk_mult_i,
54                                     feedback_i=self.clk_dco_init//self.
55                                     ↵ clk_ref_i,
56                                     lock_cnt_i=self.lock_cnt_i)
57
58             # Outputs from design
59             self.lock_o = self.clk_dco_init//self.clk_ref_i == self.
60             ↵ clk_mult_i
61             self.clk_dco_o = self.clk_dco_init
62
63         def run(self, rst_ni):
64             self.rst_ni = rst_ni
65
66             # Convert control mode integer value to the corresponding binary
67             ↵ array (1/0 integers)
68             dco_ctrl_mode_bin_arr = list(map(int, list(bin(self.
69             ↵ dco_ctrl_mode_i)[2:][::-1])))
70
71             # PID logic
72             self.pid_inst.feedback_i = self.clk_dco_o//self.clk_ref_i
73             self.pid_inst.run(rst_ni=self.rst_ni)
74             dco_ctrl_pid = self.pid_inst.ctrl_o
75
76             # Bang-bang logic
77             #cnt_en_i = 1 # used if you want frequency oscillation in the
78             ↵ stable state
79             cnt_en_i = not self.pid_inst.lock_o if (self.clk_mult_i !=
80             ↵ self.clk_dco_o//self.clk_ref_i) else 0
81             cnt_up_down = 1 if (self.clk_mult_i > self.clk_dco_o//self.
82             ↵ .clk_ref_i) else 0
83             self.cntr_inst.run(rst_ni=self.rst_ni,
84             en_i=cnt_en_i and dco_ctrl_mode_bin_arr
85             ↵ [1],
86             up_down_i=cnt_up_down)
87             dco_ctrl_cnt = self.cntr_inst.cnt_o
88
89             # Control preprocessing
90             dco_ctrl_man = self.dco_ctrl_man_i if (
91             ↵ dco_ctrl_mode_bin_arr[0]) else 203
92             dco_ctrl_auto = dco_ctrl_pid if (dco_ctrl_mode_bin_arr[0])
93             ↵ else dco_ctrl_cnt
94             dco_ctrl = dco_ctrl_auto if (dco_ctrl_mode_bin_arr[1])
95             ↵ else dco_ctrl_man
96
97             # DCO logic
98             self.dco_inst.run(ctrl=dco_ctrl)
99
100            # FLL outputs setup
101            self.clk_dco_o = int(self.dco_inst.freq_out)
102            self.lock_o = self.pid_inst.lock_o

```

Листинг 10: Имплементација класе ПИД регулатора.

```
1 import copy
2
3 INT_WIDTH = 8                      # width of output control signal
4 FX_INT_WIDTH = 12                   # integer part width of PID sum (Q12.6)
5 FX_DEC_WIDTH = 6                    # decimal part width of PID sum (Q12.6)
6 CTRL_WIDTH = FX_INT_WIDTH + FX_DEC_WIDTH # total width of PID sum
7
8 class pid:
9     def __init__(self, rst_ni, kp_i, ki_i, kd_i,
10                  setpoint_i, feedback_i, lock_cnt_i):
11         self.rst_ni = rst_ni
12         self.kp_i = kp_i
13         self.ki_i = ki_i
14         self.kd_i = kd_i
15         self.setpoint_i = setpoint_i
16         self.feedback_i = feedback_i
17         self.lock_cnt_i = lock_cnt_i
18         self.ctrl_o = 0
19         self.lock_o = 0
20         self.err_eq0_cnt = 0
21         self.integ = 0
22         self.err = setpoint_i - feedback_i
23         self.err_prev = 0
24         self.ctrl_pid = 0
25
26     def run(self, rst_ni):
27         self.rst_ni = rst_ni
28
29         if (not self.rst_ni):
30             self.err_eq0_cnt = 0
31         else:
32             if ((self.err == 0) and (self.err_eq0_cnt < self.
33             ↳ lock_cnt_i)):
34                 self.err_eq0_cnt += 1
35
36             prop = 0
37             deriv = 0
38             self.err = self.setpoint_i - self.feedback_i
39             err_diff = self.err - self.err_prev
40
41             if (not self.rst_ni):
42                 prop = 0
43                 self.integ = 0
44                 deriv = 0
45                 self.err_prev = 0
46                 self.ctrl_pid = 0
47                 self.lock_o = 0
48             else:
49                 prop = self.kp_i * self.err
50                 self.integ += self.ki_i * self.err
```

```

50         deriv = self.kd_i * err_diff
51         self.err_prev = copy.deepcopy(self.err)
52         if ((self.err == 0) and (self.err_eq0_cnt == self.
53             ↳ lock_cnt_i) and (self.lock_cnt_i != 0)):
53             self.lock_o = 1
54         if (not self.lock_o):
55             self.ctrl_pid = prop + self.integ + deriv
56
57         self.ctrl_o = int(format(int(self.ctrl_pid), '0'+str(
57             ↳ CTRL_WIDTH)+'b')[FX_INT_WIDTH-INT_WIDTH:FX_INT_WIDTH], 2)

```

Листинг 11: Имплементација класе бројача двостепеног регулатора.

```

1 class cntr:
2     def __init__(self):
3         self.cnt_o = 0
4         self.default_val = 0
5
6     def run(self, rst_ni, en_i, up_down_i):
7         if (not rst_ni):
8             self.cnt_o = self.default_val
9         else:
10            if (en_i):
11                if (up_down_i):
12                    self.cnt_o += 1
13                else:
14                    self.cnt_o -= 1
15
15     return self.cnt_o

```

Листинг 12: Имплементација класе дигитално контролисаног осцилатора.

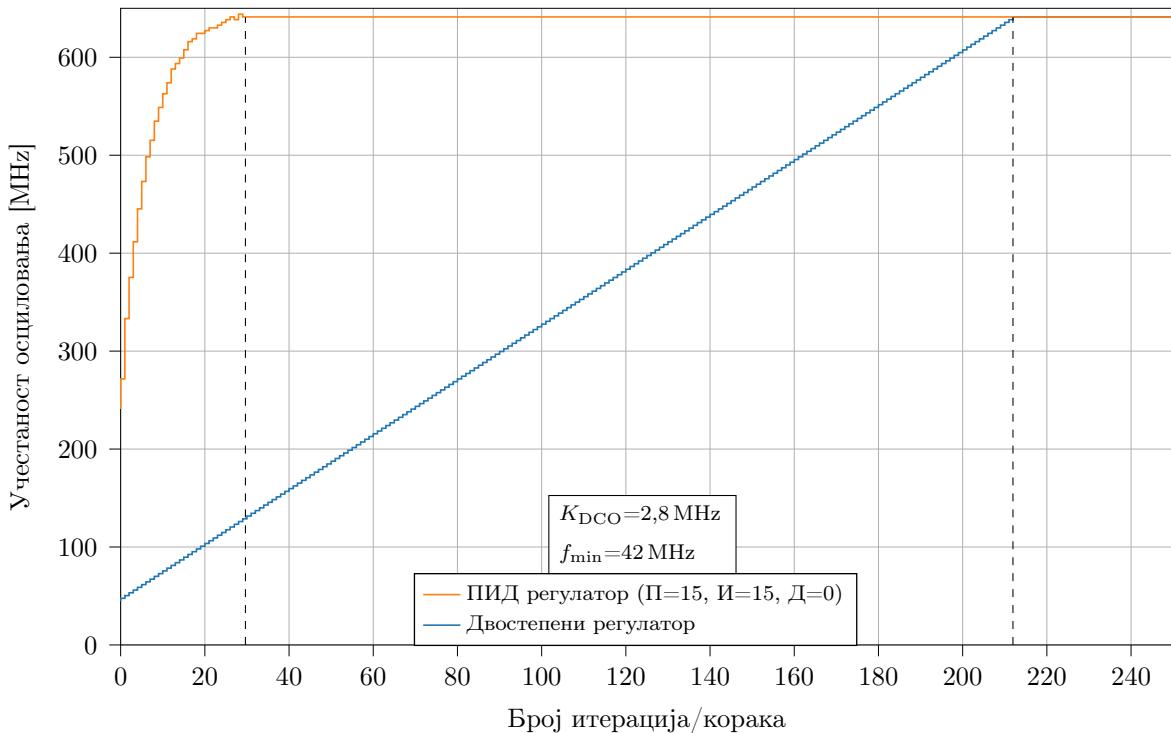
```

1 class dco:
2     def __init__(self, k_dco, init_freq):
3         self.k_dco = k_dco
4         self.init_freq = init_freq
5         self.freq_out = init_freq
6
7     def run(self, ctrl):
8         self.freq_out = self.init_freq + self.k_dco * ctrl

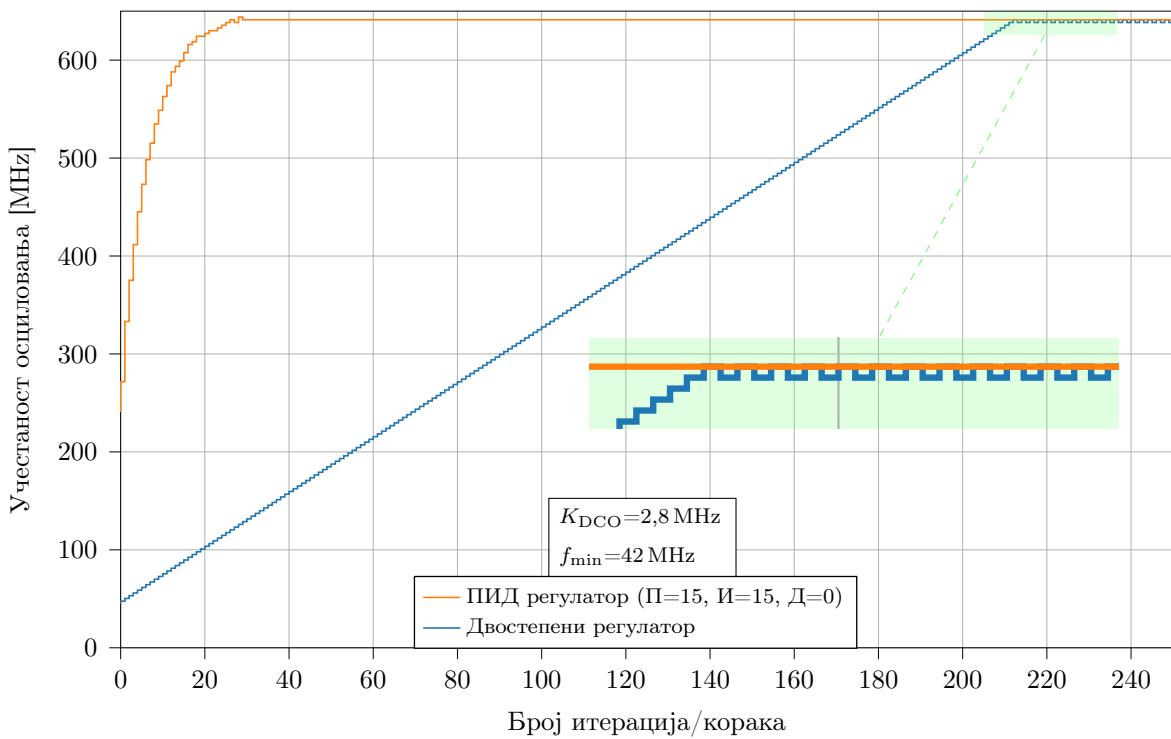
```

7.2 Поређење рада различитих управљачких режима

Слика 35 приказује одзив софтверског модела FLL-а при раду у ПИД и двостепеном управљачком режиму рада. У конкретној симулацији коришћено је понапање DCO-а добијено симулацијом у типичним условима рада, па су при инстанцирању софтверског модела FLL-а коришћени подаци из табеле 7, односно $K_{DCO}=2,8\text{ MHz}$ и $f_{min}=42\text{ MHz}$ за почетну учестаност. Конкретна вриједност учестаности која се овом симулацијом достиже у оба режима је $641,2\text{ MHz}$. У ПИД режиму модел до те учестаности стиже након 28 итерација, док му је са двостепеним управљачким режимом потребно 214 итерација.



Слика 35: Одзив модела у ПИД и двостепеном режиму рада.



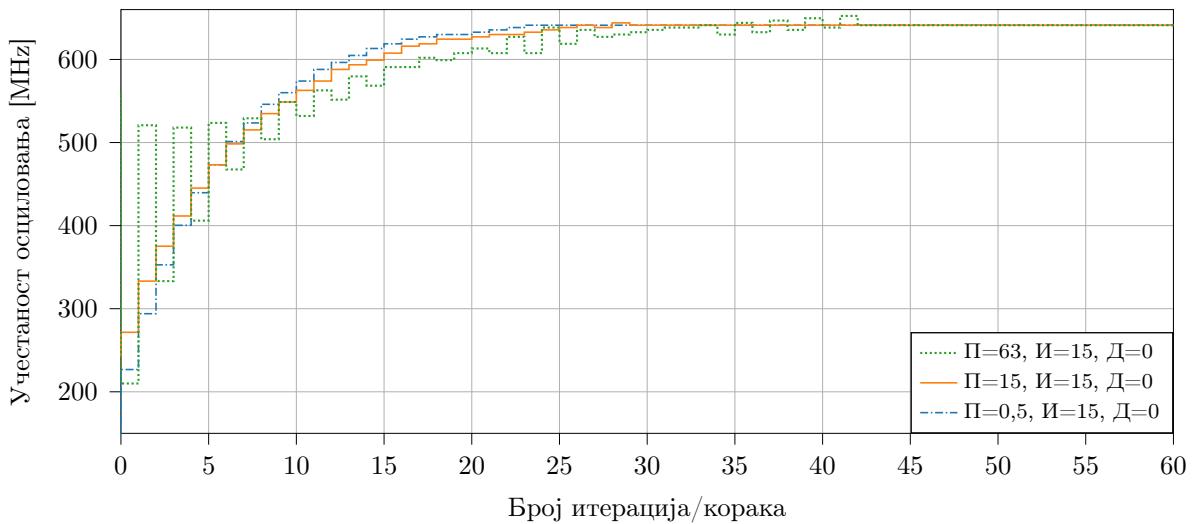
Слика 36: Одзив модела у ПИД и двостепеном режиму рада (са осцилацијама у двостепеном режиму).

У поглављу 3.2 поменута је појава осциловања учестаности у стабилном стању када FLL ради у двостепеном режиму. На слици 35 те осцилације не постоје, а проблем је решен условним довођењем `pid.lock_o` сигнала на `cnt.en_i` улаз бројача двостепеног регулатора. Логика имплементације се налази на линији 74 листинга 9. Да

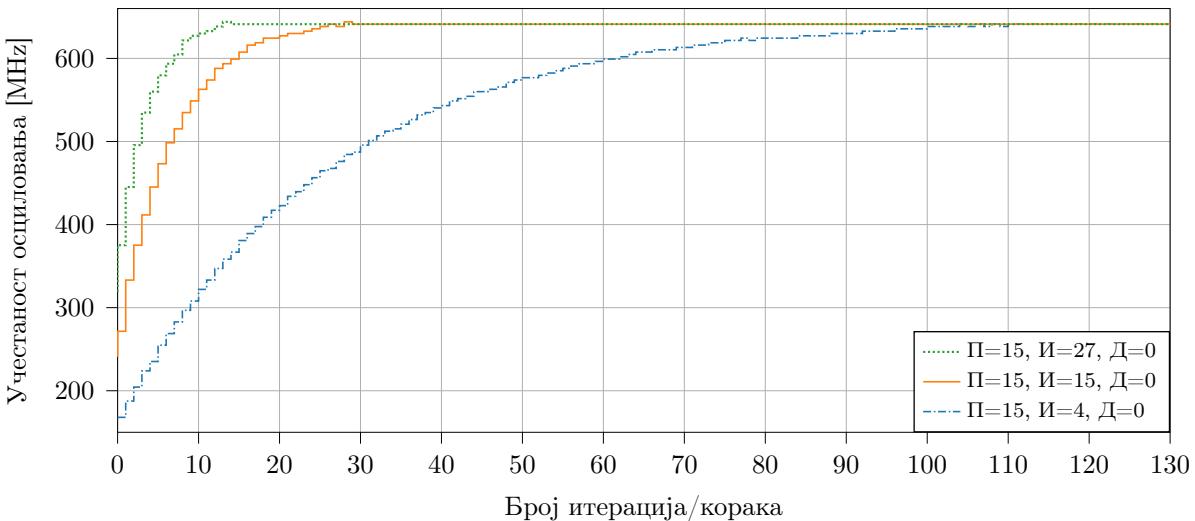
би се ипак показало како те осцилације изгледају, довољно је само закоментарисати поменуту линију, а одкоментарисати линију 73. Тиме се добија одзив као на слици 36, где учестаност у стабилном стању, при коришћењу двостепеног регулатора, варира између 638,4 MHz и 641,2 MHz (примјећујемо да разлика између поменутих вриједности представља вриједност резолуције, $K_{DCO}=2,8\text{ MHz}$).

7.3 Утицај константи ПИД регулатора на одзив система

У овом поглављу биће софтверски симулиран утицај различитих вриједности константи ПИД регулатора на одзив система. Треба напоменути да одзив не мора у потпуности одговарати стварном одзиву система већ за сврху има помоћ при исправном одабиру константи ПИД регулатора, а све у циљу добијања најбржег и најправилнијег одзыва система у крајњој имплементацији. Као што је објашњено у поглављу 3.3, диференцијална константа није коришћена у имплементацији, па стога неће бити приказан утицај њених различитих вриједности на одзив система.



Слика 37: Одзив модела за различите вриједности пропорционалне константе ПИД регулатора.

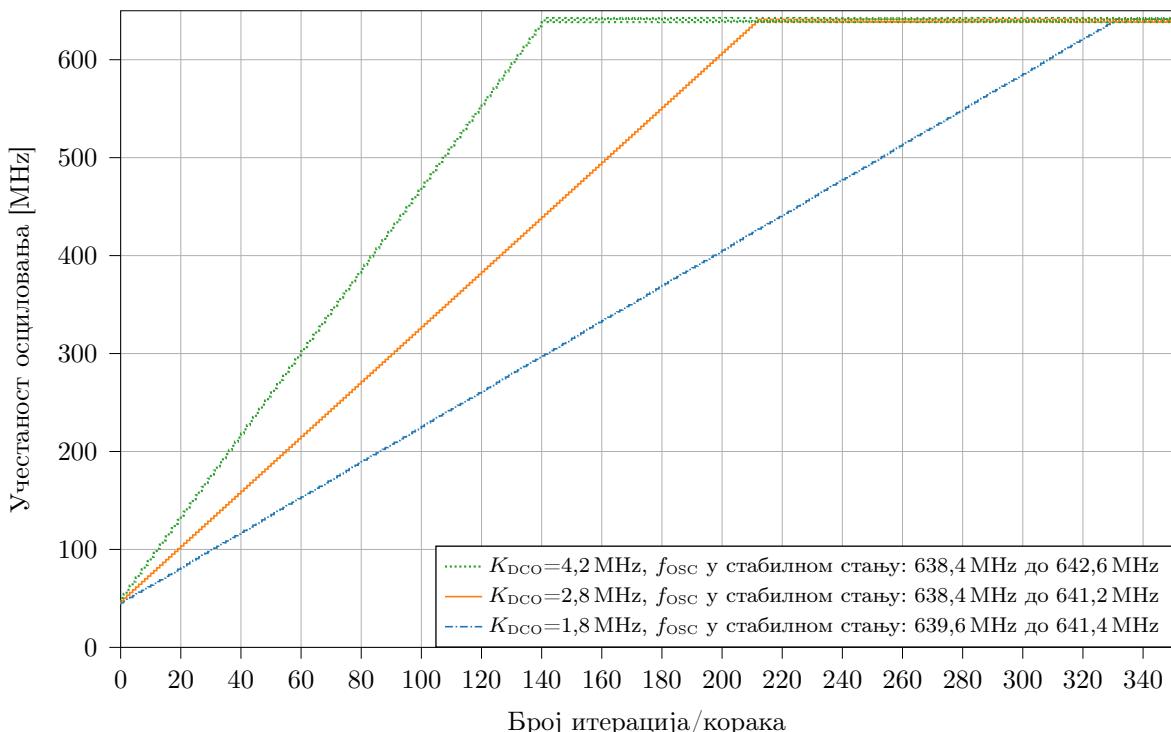


Слика 38: Одзив модела за различите вриједности интегралне константе ПИД регулатора.

Слике 37 и 38 редом приказују утицај различитих вриједности пропорционалне (K_p) и интегралне (K_i) константе на брзину и начин достизања жељене учестаности. Уочава се да пропорционална константа утиче на осјетљивост и стабилност система, а у општем случају систем може постати нестабилан ако је њена вриједност превелика. С друге стране, повећањем интегралне константе убрзава се кретање излаза ка жељеној вриједности. Треба напоменути да је децимални опсег вриједности константи од 0 до 63,75 јер представљају 8-битне вриједности у формату Q6.2, што је поменуто и у поглављу 6.1.

7.4 Утицај резолуције DCO-а на одзив система

Оно што такође знатно може убрзати, а самим тим и олакшати, пројектовање јесте предвиђање одзива система за различите карактеристике DCO-а. Тако ће у наставку бити приказан одзив за различите вриједности резолуције и то само у двостепеном режиму рада због поједностављења приказа. Слика 39 приказује одзив за три различите K_{DCO} вриједности при почетној учестаности од 42 MHz. У приказу је намјерно подешена верзија модела са осцилацијама у стабилном стању како би се што боље дочарао утицај резолуције на прецизност система. Такође се са графика може примјетити да повећањем резолуције тј. прецизности опада брзина одзива система.



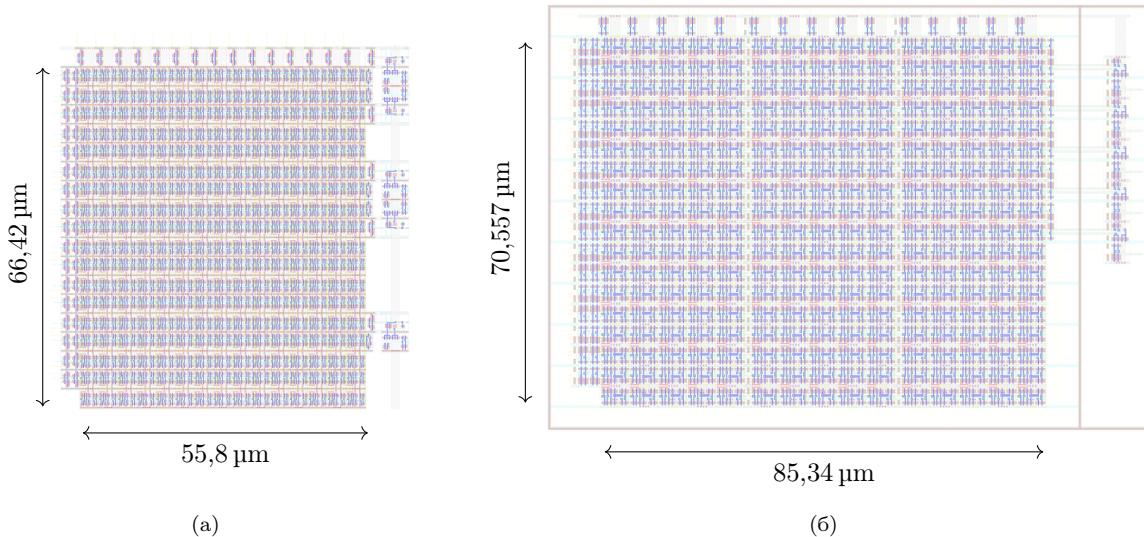
Слика 39: Моделован одзив система за различите вриједности резолуције, K_{DCO} .

8 Поређење DCO-а пројектованих у 130 nm и 180 nm технологији

С обзиром да скалирање технологије у којој се врши фабрикација значајно утиче на перформансе система, ово поглавље ће бити посвећено поређењу особина два дигитално контролисана осцилатора, једног пројектованог у TSMC 130 nm технологији (у којој је пројектован читав FLL из овог рада), а другог у TSMC 180 nm технологији.

Како скалирање уређаја значајно смањује минималну величину компоненти CMOS уређаја, тако се и други CMOS параметри, сходно томе, повећавају или смањују, а све у циљу постизања бољих перформанси. Главни параметри који утичу на перформансе CMOS-а су: дебљина оксида t_{ox} , укупна капацитивност гејта C_g и укупна отпорност гејта R_g , а утицај скалирања на њихове вриједности и даље на перформансе је детаљно описан у литератури [19].

Уопштено, ознаке 180 nm и 130 nm односе се на дужину полисилицијумског гејта MOSFET-а, што значи да се скалирањем технологије дужина гејта скраћује и тиме транзистори постају мањи. Конкретно у овом случају се ради о скраћивању гејта за око 30 %. То омогућава већу густину транзистора на истом чипу и тиме долази до смањења укупне површине интегрисаног кола. Мањи транзистори омогућавају интеграцију више функција на истом чипу, чиме се смањује укупна површина потребна за реализацију исте функционалности. Слика 40 приказује конкретно утицај скалирања технологије на површину потребну за реализацију петостепеног DCO-а са укупно 270 ћелија. Мјере са слике се односе на дужине и ширине без претварача напонских нивоа (који се налазе са лијеве, десне и горње стране у оба лејаута), како би се добило мјеродавније поређење. Површина која је била потребна за реализацију лејаута DCO-а у 130 nm и 180 nm технологији износи око $3700 \mu\text{m}^2$ и $6000 \mu\text{m}^2$, респективно. То значи да је скоро 40 % мања површина потребна за реализацију у 130 nm. По Муровом закону [20], ако имамо скалирање технологије (дужине гејта) за 30 %, тада би смањење укупне површине дизајна требало бити око 50 %, што је релативно близу добијеном резултату, узимајући у обзир и друге факторе пројектовања.



Слика 40: Лејаут DCO-а пројектованог у TSMC (а) 130 nm и (б) 180 nm технологији.

Скалирање транзистора обично долази са смањењем напона напајања, јер скраћи-

вањем канала од дрејна до сорса смањује се напон прага (енгл. *threshold voltage*), па зато мањи транзистори могу радити на нижим напонима [21]. То смањење напона смањује динамичку потрошњу, која је пропорционална квадрату напона напајања:

$$P_{dyn} = C_L V_{supply}^2 f. \quad (52)$$

Такође, смањење капацитивног оптерећења смањује динамичку потрошњу. У нашем случају извршене су симулације при раду на учестаности 640 MHz у типичним условима рада са вриједностима $V_{DD}=1,2\text{ V}$ и $V_{DDL}=1,1\text{ V}$ за DCO у 130 nm и $V_{DD}=1,8\text{ V}$ и $V_{DDL}=1,6\text{ V}$ за DCO у 180 nm. Средње квадратне (RMS) вриједности потрошње струје и просјечна потрошња снаге на тим линијама напајања, за оба осцилатора, дати су у Табели 8. Из резултата се види да је просјечна потрошња снаге мања за чак око 50 % (долази до изражaja квадратна зависност од напона напајања), а RMS вриједности струје за око 30 % за DCO пројектован у 130 nm.

Табела 8: Поређење RMS вриједности потрошње струје и просјечне снаге DCO-а пројектованих у 130 nm и 180 nm.

Технологија	130 nm	180 nm
$\text{rms}(i_{DDL})$	2,185 mA	3,105 mA
$\text{rms}(i_{DD})$	0,18 mA	0,27 mA
Укупно $\text{rms}(i)$	2,365 mA	3,375 mA
$P_{avg}(V_{DDL})$	2,5 mW	5,0 mW
$P_{avg}(V_{DD})$	0,22 mW	0,5 mW
Укупно P_{avg}	2,72 mW	5,5 mW

Додатна предност скалирања је што мањи транзистори могу радити на вишим учестаностима, јер је омогућено брже пребацање између логичких нивоа, односно скалирање транзистора доводи до смањења кашњења при пребацању са 0 на 1 и обратнуто, што повећава укупне перформансе чипа. Кашњење је сразмјерно отпорности и капацитивности гејта, а обје поменуте величине су директно пропорционалне димензијама гејта које се смањују скалирањем технологије. То значи да ће осцилатор у напреднијој технологији (у овом случају 130 nm) бити у могућности да синтетише сигнал веће учестаности, за исти напон напајања. Табела 10 приказује резултате симулација при различитим условима рада и вриједностима напона напајања. Подешавање параметара који одређују услове рада приказани су у табели 9.

Табела 9: Подешавање услова рада за симулације опсега учестаности.

Случај	Технологија	Процесни угао	Температура	V_{DD}	V_{DDL}
Најспорији	130 nm	slow-slow	125° C	1,1 V	1,0 V
Типични	130 nm	typical-typical	40° C	1,2 V	1,1 V
Најбржи	130 nm	fast-fast	-40° C	1,3 V	1,2 V
Најспорији	180 nm	slow-slow	125° C	1,6 V	1,4 V
Типични	180 nm	typical-typical	40° C	1,8 V	1,6 V
Најбржи	180 nm	fast-fast	-40° C	2,0 V	1,8 V

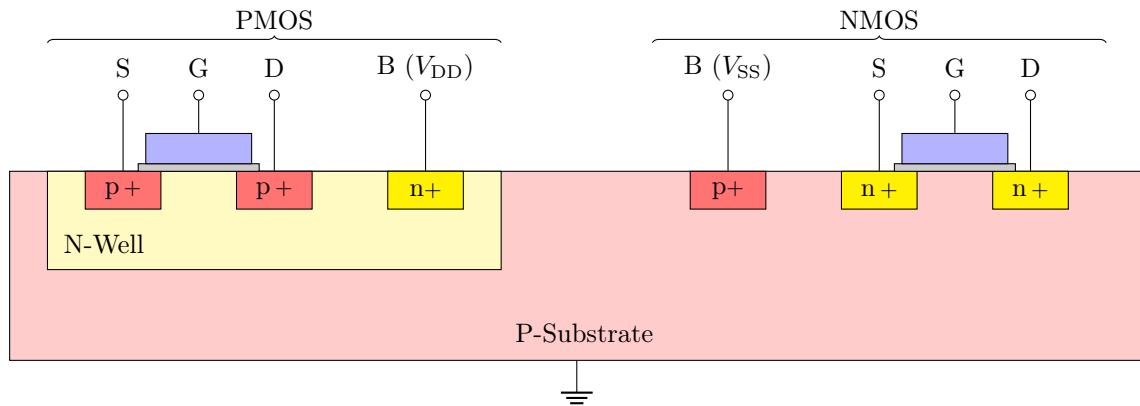
Из резултата се види да се скалирањем технологије значајно може смањити напон напајања да би се добио жељени опсег учестаности, што наравно значи да се уз много мању потрошњу постижу сличне, или чак боље, перформансе у раду DCO-а.

Табела 10: Поређење опсега учестаности при различитим условима рада за DCO пројектован у 130 nm и 180 nm.

Случај	Технологија	f_{\min}	f_{\max}
Најспорији	130 nm	27,2 MHz	502 MHz
Типични	130 nm	42 MHz	764 MHz
Најбржи	130 nm	63,3 MHz	1,146 GHz
Најспорији	180 nm	23,06 MHz	421,4 MHz
Типични	180 nm	40 MHz	726 MHz
Најбржи	180 nm	64,43 MHz	1,168 GHz

8.1 Начин и утицај коришћења дубоке јаме N типа у аналогном пројектовању DCO-а

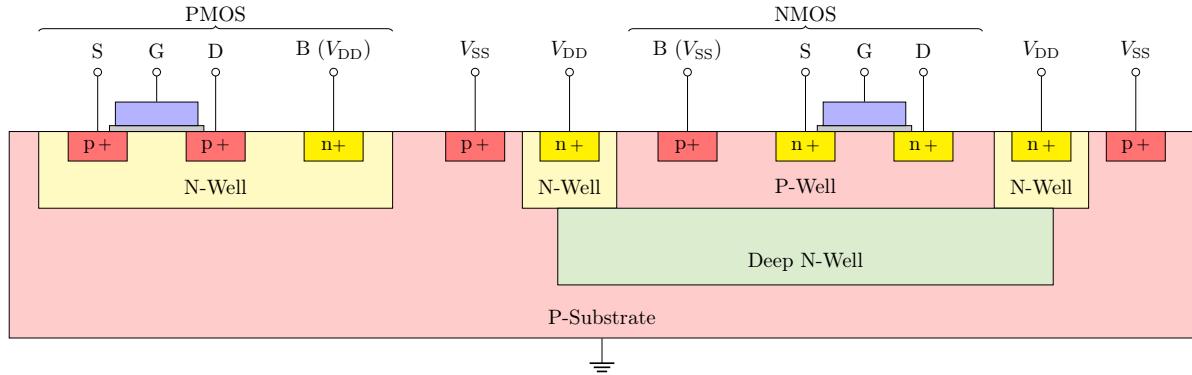
Дигитално контролисани осцилатор пројектован у 180 nm TSMC технологији, поред стандардних јама (енгл. *well*) P и N типа, посједује додатну тзв. дубоку јаму N типа (енгл. *deep N-well*, DNW). Да би се показало чему она служи, потребно је анализирати појаве које се дешавају у слојевима у које су сами транзистори уроњени. Приликом рада транзистора појављује се шум у супстрату који се дешава због убризгавања наелектрисања у супстрат и јаме. Тада проблем се рјешава употребом n+ и p+ спојева (енгл. *tap*) повезаних редом на V_{DD} и V_{SS} , за PMOS и NMOS транзисторе, респективно (слика 41). Додатни проблем је што постоји капацитивно спрезање између N јаме и P супстрата, што значи да више шума долази до извора. У дигиталним колима то обично није проблем због прилично високе отпорности логичких кола на шум. Међутим, у аналогном пројектовању, шум може бити озбиљан проблем па се користе различите технике за његово сузбијање као што су додавање заштитних прстенова (енгл. *guard ring*) или коришћење одвојеног напајања за спојеве у супстрату и јамама. Треба поменути да заштитни прстенови не могу спријечити појаву шума дубоко у супстрату [22]. Други проблем је што NMOS транзистори обично нису изоловани од супстрата. У основној CMOS изради, NMOS транзистор је уроњен у P супстрат повезан на масу, док је PMOS уроњен у N јаму (енгл. *N-well*) повезану на напајање, као што је приказано на слици 41.



Слика 41: Попречни пресек PMOS и NMOS транзистора.

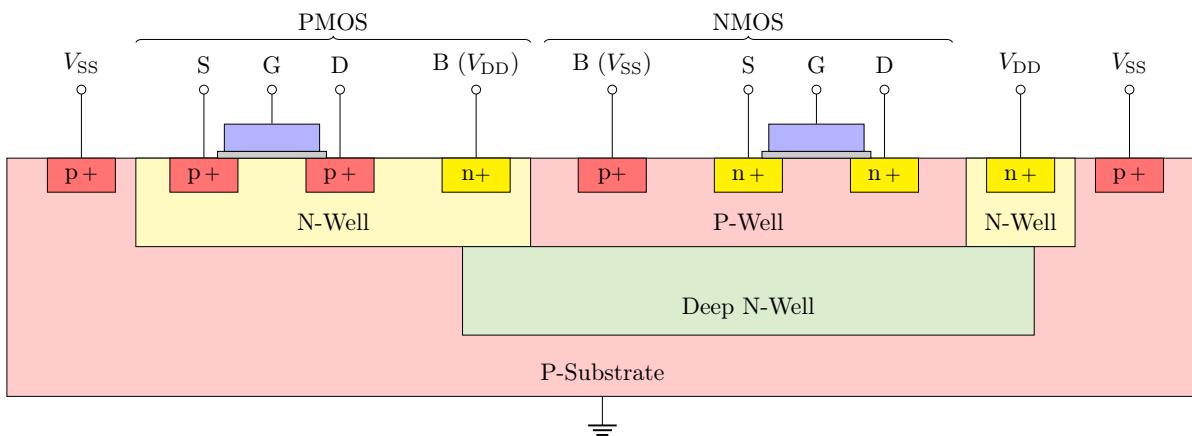
Изолација NMOS транзистора се може постићи коришћењем поменуте дубоке јаме N типа. Она се формира уметањем високоенергетске јонске имплантације непосредно прије формирања стандардне јаме N типа [23]. Повезивање дубоке јаме N типа

врши се преко стандардне јаме N типа која је окружује и повезана је на напајање, V_{DD} . Тако се, стварањем изоловане јаме P типа (енгл. *P-well*), изолује NMOS транзистор. Како додатном изолацијом глобални P супстрат остаје без p+ споја који је неопходан да се појавом шума у супстрату не наруше перформансе уређаја, додаје се заштитни прстен који се повезује на V_{SS} . Слика 42 приказује начин изолације NMOS транзистора преко дубоке јаме N типа уз додати заштитни прстен.



Слика 42: Попречни пресјек PMOS и NMOS транзистора, где је NMOS транзистор изолован дубоком јамом N типа (енгл. *Deep N-Well*) са додатим заштитним прстеном повезаним на V_{SS} .

Како NMOS и PMOS транзистори обично иду у пару приликом CMOS пројектовања, да би се боље искористио простор тј. смањила површина крањег лејаута, за повезивање дубоке јаме N типа се може искористити стандардна јама N типа коју користи PMOS транзистор, а заштитни прстен се помјера да обухвати читав дизајн, што је приказано на слици 43. У конкретном дизајну DCO-а, са слике 406 се може уочити заштитни прстен који обухвата читав DCO.



Слика 43: Попречни пресјек просторно ефикаснијег лејаута PMOS и изолованог NMOS транзистора.

Дакле, дубока јама N типа има улогу у смањењу шума у супстрату и омогућава потпуно изоловање NMOS транзистора, који, теоретски, након тога могу бити на потенцијалу различитом од масе [22]. Како додавање дубоке јаме N типа изискује и додавање заштитног прстена, површина коју заузима дизајн се повећава, тако да то може бити недостатак овог приступа ако су захтјеви за површином строги. Углавном то није случај јер су захтјеви за перформансама обично битнији тако да је цијена која се плаћа у површини занемарљива у односу на предности које има изолација NMOS транзистора у погледу смањења шума у супстрату.

9 Закључак

У овом раду представљена је синтетизабилна дигитална фреквенцијски затворена петља са једноставном али ефикасном архитектуром. Дигитално контролисан осцилатор са прстенастом структуром, састављен од тростатичких инвертора омогућава достизање пристојне резолуције учестаности и одговарајућих радних перформанси. Додатне предности су му и једноставно управљање, прилагодљивост и једноставна поновна употреба. Представљена су два управљачка режима FLL-а: двостепени регулатор и ПИД регулатор. За сваки од њих је детаљно објашњен начин рада и вјеродостојно су приказане разлике у одзиву система у различitim управљачким режимима. Детаљно је објашњена размјена података између различитих домена такта, као и методе заштите од метастабилности.

FLL из овог рада је имплементиран у SystemVerilog језику за опис хардвера. У зависности од технолошке библиотеке може бити направљен коришћењем само библиотеке стандардних ћелија без преуређивања у току дигиталног пројектовања. Обрађени FLL је послат на фабрикацију у 130 nm CMOS технологији.

Ради додатне верификације направљен је и демонстриран софтверски модел рада читавог FLL-а написан у Пајтон програмском језику. Поред саме демонстрације рада, захваљујући параметрима које прима, помоћу модела се на бржи и једноставнији начин могу предвидјети исправне вриједности константи које се задају систему (нпр. константе ПИД регулатора), што значајно убрзава пројектовање.

У сврху приказа утицаја скалирања технологије на рад система тј. дигитално контролисаног осцилатора као његовог главног дијела, направљено је поређење DCO-а пројектованог у 130 nm технологији и DCO-а са идентичном архитектуром, али у 180 nm технологији. Упоређени су главни параметри као што су површина, потрошња, као и опсег и резолуција учестаности при различитим условима рада.

Даљи рад на описаном и имплементираном FLL-у ће укључити тестирање чипа након производње и паковања, као и пројектовање додатних синтетизабилних дигиталних блокова као што су фазно спрегнута петља (енгл. *Phase-Locked Loop*, PLL) и стабилизатор напона (енгл. *Low-Dropout Regulator*, LDO), који може бити коришћен као напајање DCO-а.

Литература

- [1] Imran Ali, Hamed Abbasizadeh, Muhammad Riaz Ur Rehman, Muhammad Asif, Seong Jin Oh, Young Gun Pu, Minjae Lee, Keum Cheol Hwang, Youngoo Yang и Kang-Yoon Lee, „An Ultra-Low Power, Adaptive All-Digital Frequency-Locked Loop With Gain Estimation and Constant Current DCO”, y: *IEEE Access* 8 (2020.), стр. 97215–97230, DOI: 10.1109/ACCESS.2020.2995853.
- [2] Ahmed Musa, Wei Deng, Teerachot Siriburanon, Masaya Miyahara, Kenichi Okada и Akira Matsuzawa, „A Compact, Low-Power and Low-Jitter Dual-Loop Injection Locked PLL Using All-Digital PVT Calibration”, y: *IEEE Journal of Solid-State Circuits* 49.1 (2014.), стр. 50–60, DOI: 10.1109/JSSC.2013.2284651.
- [3] Wei Deng, Dongsheng Yang, Tomohiro Ueno, Teerachot Siriburanon, Satoshi Kondo, Kenichi Okada и Akira Matsuzawa, „A Fully Synthesizable All-Digital PLL With Interpolative Phase Coupled Oscillator, Current-Output DAC, and Fine-Resolution Digital Varactor Using Gated Edge Injection Technique”, y: *IEEE Journal of Solid-State Circuits* 50.1 (2015.), стр. 68–80, DOI: 10.1109/JSSC.2014.2348311.
- [4] José A. Tierno, Alexander V. Rylyakov и Daniel J. Friedman, „A Wide Power Supply Range, Wide Tuning Range, All Static CMOS All Digital PLL in 65 nm SOI”, y: *IEEE Journal of Solid-State Circuits* 43.1 (2008.), стр. 42–51, DOI: 10.1109/JSSC.2007.910966.
- [5] A. V. Rylyakov, J. A. Tierno, D. Z. Turker, J.-O. Plouchart, H. A. Ainspan и D. Friedman, „A Modular All-Digital PLL Architecture Enabling Both 1-to-2GHz and 24-to-32GHz Operation in 65nm CMOS”, y: *2008 IEEE International Solid-State Circuits Conference - Digest of Technical Papers* (2008.), стр. 516–632, DOI: 10.1109/ISSCC.2008.4523284.
- [6] Robert Bogdan Staszewski и Poras T. Balsara, *All-Digital Frequency Synthesizer in Deep-Submicron CMOS*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2005.
- [7] Karl Johan Astrom и Tore Hagglund, *PID Controllers: Theory, Design, and Tuning*, 2. издање, Instrument Society of America, 1995.
- [8] Bela G. Liptak, *Instrument Engineers' Handbook: Process Control and Optimization*, 4. издање, св. 2, CRC Press, Taylor & Francis Group, 2006.
- [9] David Money Harris и Sarah L. Harris, *Digital Design and Computer Architecture*, 2. издање, Morgan Kaufmann, 2012.
- [10] Martin Davis, Ron Sigal и Elaine J. Weyuker, *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science (Computer Science and Scientific Computing)*, 2. издање, Academic Press, Inc., 1994., стр. 117.
- [11] Behzad Razavi, *Design of CMOS Phase-Locked Loops: From Circuit Level to Architecture Level*, Cambridge University Press, 2020.
- [12] Anant Agarwal и Jeffrey H. Lang, *Foundations of Analog and Digital Electronic Circuits*, ISSN, Elsevier Science, 2005., ISBN: 9780080506814.
- [13] Reddy B. Madhusudhana, P.P. Jose, P.L. Babu и V. Logeshwaran, „CMOS Based Digital Controlled Oscillators (DCO) – A Review”, y: *International Journal of Applied Engineering Research* 10.20 (2015.), стр. 18626–18630.

- [14] M. Terosiet, E. Zianbetov, F. Vallette, M. M. Louerat, P. Garda, D. Galayko и S. Feruglio, „A Comprehensive In-Depth Study of Tri-State Inverter Based DCO”, у: *Microelectronics Journal* (2020.), DOI: 10.1016/j.mejo.2020.104760.
- [15] Jan M. Rabaey, Anantha P. Chandrakasan и Borivoje Nikolić, *Digital Integrated Circuits: A Design Perspective*, 2. издање, Prentice Hall electronics and VLSI series, Pearson Education, 2003., ISBN: 9780131207646.
- [16] V. Milovanović и B. Nikolić, „An HDL model of a digitally controlled oscillator for rapid digital PLL prototyping”, у: *2017 IEEE 30th International Conference on Microelectronics (MIEL)*, 2017., стр. 205–208, DOI: 10.1109/MIEL.2017.8190103.
- [17] Yuji Osaki, Tetsuya Hirose, Nobutaka Kuroki и Masahiro Numa, „A Low-Power level-shifter With Logic Error Correction for Extremely Low-Voltage Digital CMOS LSIs”, у: *IEEE Journal of Solid-State Circuits* 47.7 (2012.), стр. 1776–1783, DOI: 10.1109/JSSC.2012.2191320.
- [18] „IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language”, у: *IEEE Std 1800-2023 (Revision of IEEE Std 1800-2017)* (2024.), стр. 1–1354, DOI: 10.1109/IEEESTD.2024.10458102.
- [19] Hassan Hassan, Mohab Anis и Mohamed Elmasry, „Impact of technology scaling and process variations on RF CMOS devices”, у: *Microelectronics Journal* 37.4 (2006.), стр. 275–282, ISSN: 1879-2391, DOI: 10.1016/j.mejo.2005.07.013.
- [20] Gordon E. Moore, „Cramming More Components Onto Integrated Circuits”, у: *Electronics* 38.8 (1965.), стр. 114–117, DOI: 10.1109/JPROC.1998.658762.
- [21] R.H. Dennard, F.H. Gaensslen, Hwa-Nien Yu, V.L. Rideout, E. Bassous и A.R. LeBlanc, „Design of ion-implanted MOSFET's with very small physical dimensions”, у: *IEEE Journal of Solid-State Circuits* 9.5 (1974.), стр. 256–268, DOI: 10.1109/JSSC.1974.1050511.
- [22] *Using Deep N Wells in Analog Design*, AspenCore, Electronics Industry Media & SaaS Group, <https://www.planetanalog.com/using-deep-n-wells-in-analog-design/> (посећено 20. 9. 2024.).
- [23] K. Wai Chew, J. Zhang, K. Shao, W. Boon Loh и S.-F. Chu, „Impact of Deep N-well Implantation on Substrate Noise Coupling and RF Transistor Performance for Systems-on-a-Chip Integration”, у: *32nd European Solid-State Device Research Conference*, 2002., стр. 251–254, DOI: 10.1109/ESSDERC.2002.194917.

Објављени радови

- [1] Dejan D. Petković, Đorđe S. Gačić, Marijana R. Gavrilović Božović и Vladimir M. Milovanović, „A Synthesizable Digital Frequency-Locked Loop Widely Tunable up to 640 MHz in 130 nm CMOS”, у: *2024 11th International Conference on Electrical, Electronics and Computer Engineering (IcETRAN)*, Niš, Serbia, јун 2024., стр. 1–6, DOI: 10.1109/IcETRAN62308.2024.10645145, <https://ieeexplore.ieee.org/document/10645145> (посећено 20. 9. 2024.).