

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: Напредно машинско учење

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
- др Шћепан Шћекић
- доц. др Жарко Попара

Датум одбране: \_\_\_\_\_

Оцена: \_\_\_\_\_

У оквиру овог мастер рада кандидат треба да реализује 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 тро-стatischких 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.

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.

Крагујевац, 26. 8. 2024.

Ментор:

Др Владимира М. Миловановић,  
ванредни професор



Универзитет у Крагујевцу  
Факултет инжењерских наука  
Универзитета у Крагујевцу



Основне академске студије: **Назив студијског програма**

Модул: **Назив** (ако постоји)

Назив предмета: **Назив**

Име и презиме: **Петар Петровић**

Број индекса: 123/1832

### ПРИЈАВА ДИПЛОМСКОГ РАДА

Тема рада: Израда примера дипломског рада у LaTeX пакету коришћењем класе *finthesis*

Задатак: Приликом пријаве дипломског рада ментор задаје тему по правилу преузету са листе тема коју је усвојила одговарајућа Катедра Факултета. Циљ дипломског рада је да студент докаже способност примене стечених знања и вештина при решавању задатака који су у складу са усвојеним исходима знања. Препорука је да дипломски рад буде пројекат мањег обима, експериментално-лабораторијско испитивање, примена инструменталне методе, прегледни рад, теоријска разрада, компјутерска симулација једноставнијег реалног проблема и сл. Након одобравања теме, ментор је дужан да студента упути у начин обраде теме, упути га на додатну литературу, одреди обим дипломског рада и током консултација усмерава и контролише рад кандидата.

Ментор:  
Др Томо М. Петровић,  
изванредни професор

# Садржај

<b>1 Увод</b>	<b>3</b>
<b>2 Опис и структура фреквенцијски затворене петље</b>	<b>4</b>
<b>3 Управљачка логика дигиталне фреквенцијски затворене петље</b>	<b>6</b>
3.1 Управљачка предобрађа . . . . .	6
3.2 Bang-bang контролер . . . . .	7
3.3 PID контролер . . . . .	8
3.4 Управљачки декодер . . . . .	9
<b>4 Теоријско разматрање осцилатора</b>	<b>11</b>
4.1 Основни принципи осциловања . . . . .	11
4.2 Систем осцилаторне повратне спреге . . . . .	12
4.3 Основе прстенастог осцилатора . . . . .	14
4.3.1 Инверторски прстен . . . . .	17
<b>5 Дигитално контролисани осцилатор</b>	<b>19</b>
5.1 Архитектура предложеног DCO-а . . . . .	20
5.2 Блок дијаграм DCO ћелије . . . . .	20
5.3 Претварачи напонског нивоа DCO-а . . . . .	22
<b>6 Имплементација и резултати симулација</b>	<b>25</b>
6.1 SystemVerilog код . . . . .	25
6.2 Лејаут . . . . .	32
6.3 Симулација рада фреквенцијски затворене петље . . . . .	34
6.4 PVT зависност дигитално контролисаног осцилатора . . . . .	36
6.5 Временски одзив дигитално контролисаног осцилатора . . . . .	37
6.6 Спектар снаге фреквенцијски затворене петље . . . . .	38
6.7 Фазни шум дигитално контролисаног осцилатора . . . . .	39
<b>7 Пајтон модел FLL-а</b>	<b>41</b>
7.1 Пајтон код . . . . .	41
7.2 Поређење рада различитих управљачких режима . . . . .	46
7.3 Утицај константи PID контролера на одзив система . . . . .	47
7.4 Утицај резолуције DCO-а на одзив система . . . . .	48
<b>8 Поређење DCO-а пројектованих у 130nm и 180nm TSMC технологији</b>	<b>49</b>
<b>9 Закључак</b>	<b>50</b>
<b>Литература</b>	<b>51</b>

## 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  $100\text{ }\mu\text{m} \times 330\text{ }\mu\text{m}$  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.

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

## Резиме

Фреквенцијски затворена петља (енгл. *Frequency-Locked Loop*, FLL) представља одржив начин генерирања опсега фреквенција из једне референтне фреквенције коришћењем електронског система управљања са негативном повратном спрегом, који пореди фреквенцију контролисаног осцилатора са поменутом референтном фреквенцијом. Дигитално синтетизабилан FLL је дизајниран у 130 nm технологији за циљану фреквенцију до 640 MHz. Он погони дигитално контролисани осцилатор (енгл. *Digitally Controlled Oscillator*, DCO) са широким подешавањем опсега који се састоји од тростатаичких инвертора у облику матрице. FLL може произвољно користити тзв. (енгл. *Bang-Bang*) контролер или дјелимично програмирани стандардни пропорционално, интегрални, диференцијални (енгл. *Proportional-Integral-Derivative*, PID) контролер за управљање негативном петљом. Такав дизајн у пракси минимизује појаву метастабилности. Предложени дигитални FLL заузима  $100\text{ }\mu\text{m} \times 330\text{ }\mu\text{m}$  простора и троши 3.5 mW у уобичајеним условима рада. Референтни такт је 16 MHz, а излазна фреквенција осциловања је подешена на 640 MHz, док постигнута резолуција фреквенције износи 2.8 MHz.

**Кључне ријечи:** Фреквенцијски затворена петља, дигитално контролисани осцилатор, генератор такта, синтетизабилност, CMOS технологија, PID контролер, метастабилност.

# 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 дизајну чипова. Безброј примјена самих чипова захтјевају или генератор такта или синтезатор фреквенције, што подразумјева 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 фреквенције. Међутим, чак и веома сложени системи често захтјевају генератор такта, који само множи улазну фреквенцију без да посебно води рачуна о фази такта или апсолутном подрхтавању (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 спадати у двије групе: дигитални и аналогни FLL. Иако је очигледан недостатак првих максимална фреквенција и њена резолуција, они посједују многе друге предности наспрам њихових аналогних супарника. Они заузимају мање простора, истичу се већом отпорношћу на промјене процесних углова, напона и температуре (енгл. *Process-Voltage-Temperature*, PVT), лако су употребљиви у различитим технологијама, и стога омогућавају поновну употребу, већу прилагодљивост, једноставнију методологију тока пројектовања, 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. Дакле, фокус овог рада је пројектовати и унаприједити једноставне али моћне синтезабилне дигиталне блокове чипа.

Овај рад конкретно предлаже синтезабилан 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] изменењен додавањем независног напона напајања DCO-а са претварачима напонских нивоа (енгл. *Level Shifters*) и употребом петостепене [5] уместо тростепене толологије прстена осцилатора.

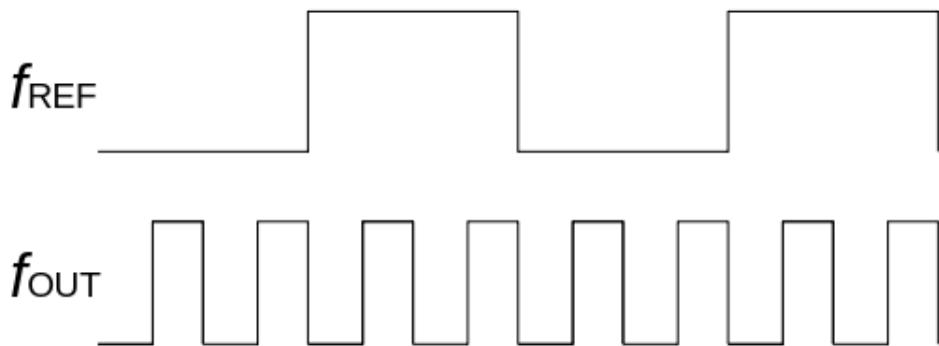
У поглављу 2 се даје основни опис рада и структуре предложеног дигиталног FLL-а. Управљачка логика FLL-а је детаљно описана у поглављу 3. Теоријска разматрања осцилатора, као главне компоненте у систему дато је у поглављу 4, док је коришћени дигитално контролисан осцилатор описан у поглављу 5. Поглавље 6 садржи анализиран RTL код 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 Пајтон програмском језику. У сврху поређења карактеристика осцилатора у различитим 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 закључава фреквенцију излазног сигнала на предвиђену циљану фреквенцију. У принципу, непрастано управља фреквенцијом осцилатора на аутоматски начин све док излазна фреквенција на достигне циљану вриједност, након чега се та вриједност фреквенције одржава на излазу. То значи да FLL спада у синтетизаторе фреквенције који се уопштено описују као активна електронска кола која имају неки сигнал са референтном и веома стабилном фреквенцијом ( $f_{REF}$ ) на улазу и затим генеришу излазни сигнал са фреквенцијом ( $f_{OUT}$ ) која представља референтну фреквенцију помножену са одговарајућим чиниоцем тј. умношком фреквенције (FMUL) као у једначини 1.

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

Треба напоменути да је стабилност, прецизност и спектрална чистоћа излазног сигнала у константној вези са квалитетом сигнала на улазу [6]. Слика 1 приказује поједностављен дигитални сигнал на улазу и излазу система са одређеном вриједношћу умношка фреквенције.



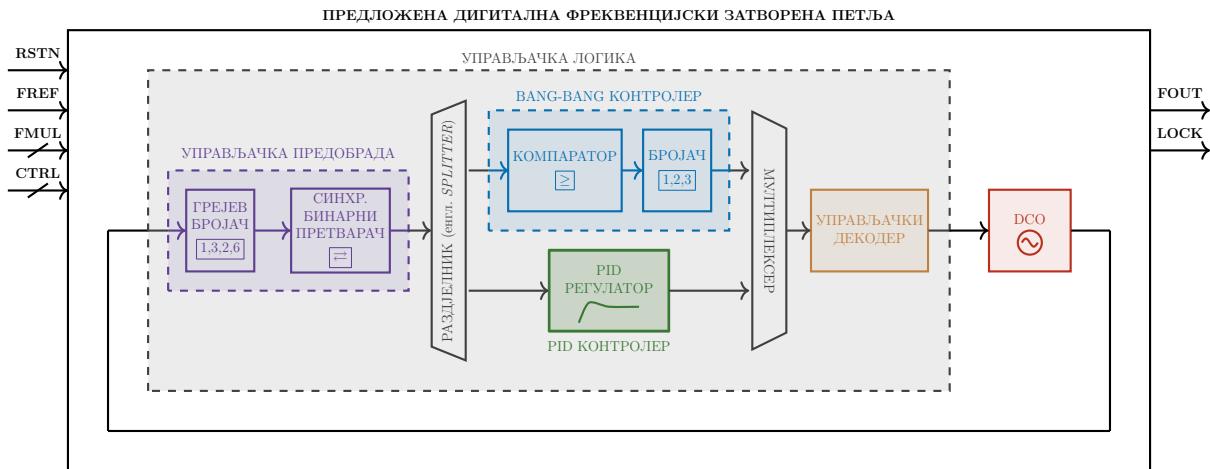
Слика 1: Демонстарција рада синтетизатора фреквенције за  $\text{FMUL}=4$ .

Дефиниција изнад не предвиђа облик синтетисаног излазног сигнала. Он може бити синусоида или правоугаони (дигитални) сигнал, који нам је у CMOS технологији дефинитивно кориснији за разматрање.

У овом раду описана је релативно једноставна али ефикасна дигитална фреквенцијски затворена петља (FLL), чију се системску архитектуру на нивоу блокова приказује Слика 2. Описаны FLL се практично састоји из два блока: дигитално контролисаног осцилатора (DCO) и блока управљачке логике, који генерише улазне сигнале за DCO на основу тренутне фреквенције DCO-а. У сврху поједностављења, са слике су изостављени неки конфигурациони улази FLL-а, као што су коефицијенти PID контролера или улаз за одабир режима рада, чија улога је детаљније описана у поглављу 6.1.

Основни улази у систем су:

- RSTN - Ресет сигнал који је активан на нулу
- FREF - Референтни такт



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

- FMUL - Улазни умножак фреквенције (множењем са фреквенцијом референтног такта добија се вриједност фреквенције 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 фреквенцији

### 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 вриједност повезану са бројем 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 излазној фреквенцији. Главне разлике између два поменута режима су брзина затварања (закључавања) FLL-а и једноставност подешавања. Циљ управљачке логике FLL-а је изједначити вриједност улазног множача фреквенције са бројем периода такта DCO-а унутар периода референтног такта што је брже и прецизније могуће, чиме се долази до постизања жељене фреквенције на излазу DCO-а. Поред самог постизања жељене фреквенције, 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 фреквенције. То значи да управљачка логика константно прати рад система и по потреби реагује на исход непожељних утицаја на излазну фреквенцију. Комплетна управљачка логика FLL-а је подијељена на неколико фаза, које су детаљно описане у наредним поглављима, а то су:

- Управљачка предобрађа
- Bang-bang контролер / PID контролер
- Управљачки декодер

#### 3.1 Управљачка предобрађа

Фаза управљачке предобраде (енгл. *Control Preprocessing*) укључује неколико блокова чија је улога претворити информацију о фреквенцији 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*). Табела 1 приказује првих осам цифара 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. Рецимо, 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 један бит.

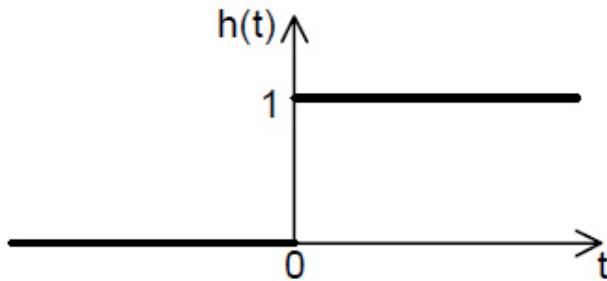
Табела 1: Првих осам цифара у Грејевом коду

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

Недостатак овог приступа је то што Грејев бројач има мању максималну радну фреквенцију од бинарног бројача. Да би се у још већем обиму 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.

### 3.2 Bang-bang контролер

У теорији управљања, bang-bang контролер (двофазни или on-off контролер) је контролер повратне спреге које се нагло пребацује између два стања. Математички модел bang-bang контролера може се представити преко Хевисајдове функције (или јединичне одскочне функције) која која се иначе користи у математици 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 3: График Хевисајдове функције.

$$u(x) = \begin{cases} 0, & x < 0 \\ 1, & x > 0 \end{cases} \quad (2)$$

Главне предности примјене bang-bang контролера 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.

Као што је већ наведено, FLL из овог рада има два међусобно раздвојена управљачка блока. Један управљачки блок FLL-а је веома сличан bang-bang контролеру описаном изнад и који, као прекидач, може имати два стања. Он пореди улазни умножак фреквенције са узоркованом вриједношћу бројача и одлучује да ли инкрементирати, декрементирати или онемогућити предстојећи блок тј. двосмјерни бројач (енгл. *Up-Down Counter*). Ако је улазни умножак фреквенције већи од вриједности бројача узорковане у блоку управљачке предобрade, тада се двосмјерни бројач инкрементира, ако је мањи тада се декрементира, а у случају да је једнак, двосмјерни бројач задржава претходно стање. То осигурува постепено управљање и закључавање све до постизања жељене фреквенције тј. преласка у стабилно стање. Како би се спријечиле (или барем прориједиле) могуће осцилације око жељене вриједности, у овом раду је доведен додатни сигнал који омогућава да се, након што систем једном уђе у стабилно стање, бројач bang-bang контролера 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 PID контролеру који представља други управљачки режим о коме ће више ријечи бити у наредном поглављу.

### 3.3 PID контролер

У другом управљачком режиму, управљачка бинарна ријеч за DCO се генерише подесивим PID контролером. PID контролер је далеко најчешће коришћен алгоритам за управљање у инжењерству. Највећим бројем повратних спрега се управља преко PID контролера или његових дјелимичних варијација [7]. PID контролер 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 предобрadi (узоркована вриједност бројача) и жељене референтне задате вриједности (вриједност улазног умношка фреквенције). Исправљање и скалирање се распоређује у три компоненте које су имплементиране као подесиви улази са фиксном тачком који се напајају из банке регистара:

- Пропорционална (P) - управља брзином одзива управљачког системам 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 контролер постаје мање осјетљив.
- Интегрална (I) - користи се за смањење грешке стабилног стања (енгл. *Steady State*) скалирањем грешке константним чиниоцем и сумирањем резултата током времена. Грешка стабилног 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 узорковане P и I компонентама, без смањења брзине контролера. 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.

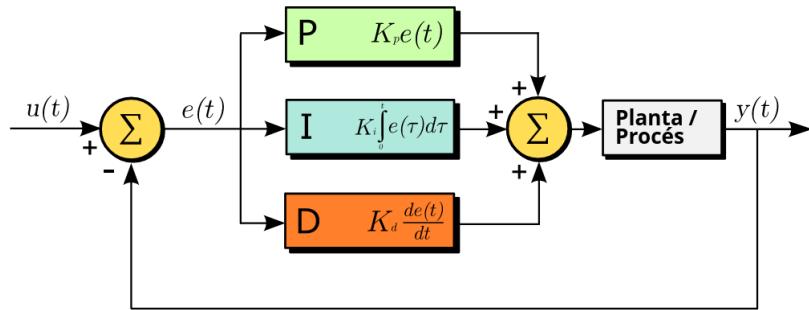
Математички, цјелокупна управљачка функција PID контролера представља суму пропорционалног, интегралног и диференцијалног дејства и записује се на следећи начин:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (3)$$

где су  $K_p$ ,  $K_i$  и  $K_d$  коефицијанти за пропорционални, интегрални и диференцијални дио, респективно. У стандардном облику, једначина 3 се записује као:

$$u(t) = K_p \left( e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (4)$$

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 контролер толерисати излаз који се налази изнад или испод жељене вриједности.  $K_p T_d$  је временска константа којом се контролер покушава приближити жељеној вриједности. Слика 4 представља уопштен блок



Слика 4: Блок дијаграм PID контролера у повратној спрези.

дијаграм PID контролера у повратној спрези који приказује структуру компоненти контролера и њихов принцип рада. PID контролер константно рачуна вриједност грешке  $e(t)$  као разлику између жељене вриједности (енгл. *Setpoint*)  $r(t)$ , која у овом конкретном случају представља улазни умножак фреквенције, и измјерене процесне вриједности  $y(t)$ , односно вриједности измјерене у блоку управљачке предобраде. У том случају, једначина грешке може се записати као:

$$e(t) = r(t) - y(t) \quad (5)$$

Одзив система тј. брзина и начин достизања жељене фреквенције у PID режиму рада највише зависи од одабира константи контролера, 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 слоковит приказ утицаја константи PID контролера на одзив система.

### 3.4 Управљачки декодер

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 редове тростатичких инвертора DCO-а; *Row Select*, један од  $n$  (енгл. *One-Hot*) вектор, који укључује један додатни ред 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 један инвертор укључио, или *Row On*, или и *Row Select* и *Column Select* за одговарајући бит морају бити подешени на 1. Ширина сваког вектора је једнака ширини улаза. Сама структура DCO-а је детаљније описана у поглављу 5.

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 фреквенција већа. Речимо 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 17 редова и 15 колона. Како постоји 17 редова, ширине *Row On* и *Row Select* вектора биће по 17 бита, док ће због 15 колона, ширина *Column Select* вектора бити 15 бита. То значи, да би се укључило 50 прекидача, потребно је укључити 3 читава реда прекидача и још 5 прекидача из четвртог реда. Да би се то постигло, управљачки улази DCO-а морају имати следеће вриједности:

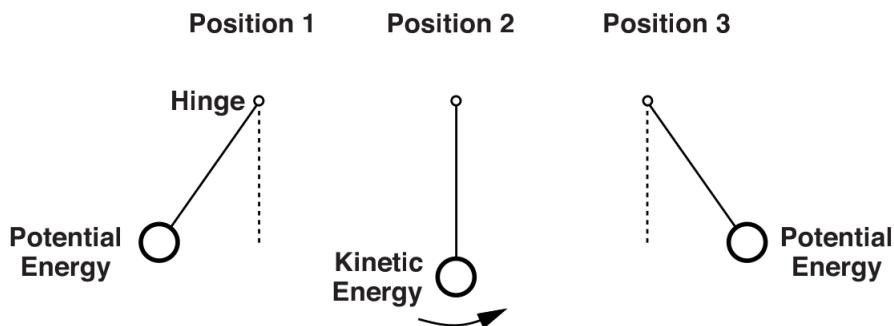
- *Row On* = 000000000000000111 → Сви прекидачи у три прва реда су укључени.
- *Row Select* = 0000000000000001000 → У четвртом реду је укључено још инвертора.
- *Column Select* = 000000000011111 → Пет инвертора је укључено у реду који је одређен са *Row Select*.

## 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 5. Ако пустимо клатно под одређеним углом, оно се неко вријеме њише и



Слика 5: Клатно које дјелује као осцилаторни систем [11].

постепено се зауставља. Осцилација почиње јер се првобитна потенцијална енергија претвара у кинетичку енергију како клатно достиже свој вертикални положај, до-звољавајући му да настави своју путању до другог екстремног угла (позиција 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 обезбједити 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 (обезбеђена довођењем клатна у положај 1).
2. Склоност да се једна врста енергије претвара у другу и обрнуто.
3. Механизам за одржавање који допуњује енергију изгубљену услед неизбежних 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 фреквенцији различитој од природне фреквенције осциловања клатна. Ако систем има склоност да осцилује на фреквенцији  $\omega_0$ , тада се ствара растући осцилаторни излаз као одговор на спољашњу побуду на фреквенцији  $\omega_0$ . Из друге перспективе, такав систем неограничено појачава периодични улаз на поменутој фреквенцији.

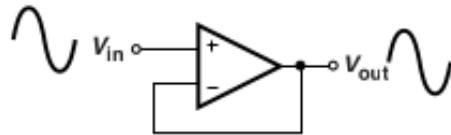
## 4.2 Систем осцилаторне повратне спреге

Из основног аналогног дизајна 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 6, где негативан предзнак на једном улазу сабирача означава негативну повратну спрегу на ниским фреквенцијама. На Слици 7 је приказан примјер имплементације такве повратне спреге, који, на нискофреквентни синусоидни улаз дјелује као јединични бафер. Треба имати на уму да операциони појачавач показује занемарљив фазни помак на ниским фреквенцијама.



Слика 6: Једноставан систем повратне спреге.



Слика 7: Реализација повратне спреге коришћењем операционог појачавача.

Међутим, поставља се питање како могу системи са Слика 6 и 7 да осцилују? Записивањем преносне функције затворене петље као

$$\frac{Y(s)}{X(s)} = \frac{H(s)}{1 + H(s)}, \quad (6)$$

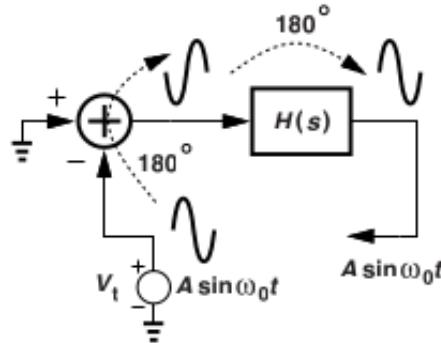
примјећује се да именилац пада на нулу ако је  $H(s) = -1$  за неку вриједност промјенљиве  $s$ . Ако је  $X(s)$  синусоида, онда је  $s = j\omega_0$  и мора важити  $H(j\omega_0) = -1$ . С обзиром да је у питању синусоида, то ће на  $\omega_0$  важити за фазни помјерај од  $-180^\circ$ . То даље значи 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 фреквенцији  $\omega_0$ , што је илустровано на Слици 8. Другим ријечима,  $H(s)$  има толико помјерања фазе (или кашићења) на  $\omega_0$  да укупна повратна спрега постаје позитивна. То се може видјети постављањем главног улаза,  $X(s)$ , на нулу, прекидом петље и примјеном стимулуса на истој фреквенцији (Слика 9). Враћени сигнал је у фази са испитним напоном,  $V_t$ . У овом случају, каже се да петља садржи фазни помјерај од  $180^\circ$  због номинално негативне повратне спрече и други, фреквенцијски зависан, фазни помјерај од  $180^\circ$  који произилази из  $H(s)$ . Та два фазна помјераја се не смију мијешати један с другим.



Слика 8: Инверзија сигнала на фреквенцији  $\omega_0$ .



Слика 9: Пропагација синусоиде кроз петљу на фреквенцији  $\omega_0$ .

Укупан фазни помјерај од  $360^\circ$  на  $\omega_0$  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:

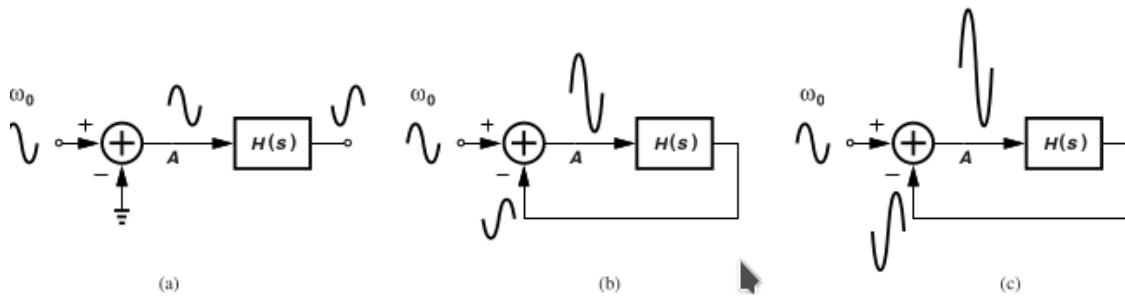
$$|H(j\omega_0)| = 1 \quad (7)$$

$$\angle H(j\omega_0) = -180^\circ, \quad (8)$$

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 10(а) се примјећује да је, са  $H(j\omega_0) = -1$ , излаз једнак улазу, али помјерен за  $180^\circ$ . Ако је петља затворена (Слика 10(б)), излаз се одузима од улаза, што даје већу амплитуду (замах) на  $A$ . Тада се затим поново инвертује и одузима од улаза, и тако у круг, што доводи до неограниченог раста амплитуде (Слика 10(в)).

Укратко, систем са негативном повратном спрегом може да генерише растући периодични излаз као одговор на синусоидни улаз ако његово појачање у петљи достигне

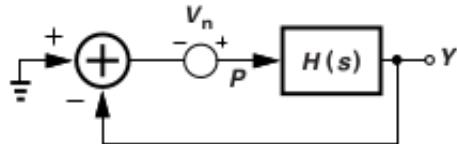


Слика 10: Раст амплитуде улазне синусоиде кроз петљу у току времена.

-1 на коначној фреквенцији,  $\omega_0$ . 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  $\omega_0$ , стварајући малу компоненту која кружи у петљи и изазива осцилације. Нпр. као што је приказано на Слици 11, извор шума,  $V_n$ , на улазу  $H(s)$  даје излаз који је једнак:

$$Y = V_n \frac{H(s)}{1 + H(s)}, \quad (9)$$

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



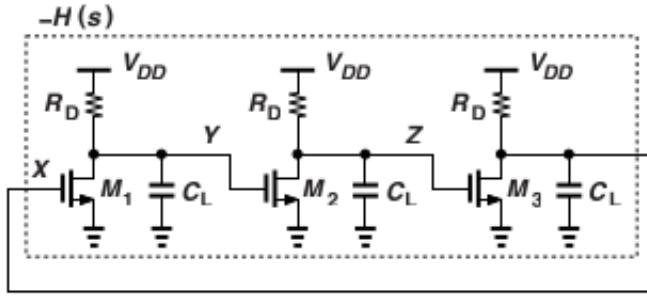
Слика 11: Утицај шума убрзганог у систем затворене петље.

### 4.3 Основе прстенастог осцилатора

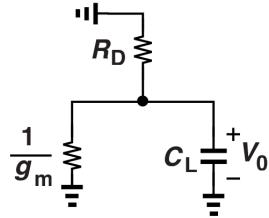
Прстенasti осцилатори су у данашње вријеме веома популарни у PLL системима захваљујући својој прилагодљивости дизајна и широког опсега подешавања фреквенције. Ово поглавље уопштено разматра основе прстенастих осцилатора, да би се стекло дубље разумјевање рада коришћеног дигитално контролисаног осцилатора,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

Слика 12 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 8. Чак и примјеном два степена, систем не може да осцилује јер  $\angle H(j\omega_0)$  достиже  $-180^\circ$  тек за  $\omega = \infty$  [11]. Због тога се претпоставља да коришћењем три степена могу бити задовољени услови из једначина 7 и 8, па се стога врши даља анализа како би се то и показало.

На Слици 12, коло означено испрекиданом линијом је означено са  $-H(s)$  да би било у складу са системом негативне повратне спрете приказаним на Слици 7.



Слика 12: Три степена са заједничким извором у повратној спрези.



Слика 13: Еквивалентно коло једног степена у петљи.

Ако погледамо еквивалентно коло за један степен на Слици 13, за преносну функцију тростепеног прстена важи [11]:

$$-H(s) = \left[ -g_m \left( R_D \| \frac{1}{C_L s} \right) \right]^3 \quad (10)$$

Ако је  $s = j\omega_0$ , тада је

$$H(j\omega_0) = \left[ g_m \left( R_D \| \frac{1}{C_L j\omega_0} \right) \right]^3 = \left( \frac{g_m R_D \frac{1}{C_L j\omega_0}}{R_D + \frac{1}{C_L j\omega_0}} \right)^3 = \left( \frac{g_m R_D}{1 + R_D C_L j\omega_0} \right)^3 \quad (11)$$

Добијени резултат представља комплексни разломак који даље желимо представити у стандардном комплексном облику  $Z = X + jY$ :

$$\begin{aligned} H(j\omega_0) &= \left( \frac{g_m R_D}{1 + R_D C_L j\omega_0} \cdot \frac{1 - R_D C_L j\omega_0}{1 - R_D C_L j\omega_0} \right)^3 = \\ &= \left( \frac{g_m R_D - g_m R_D^2 C_L j\omega_0}{1 + R_D^2 C_L^2 \omega_0^2} \right)^3 = \\ &= \left( \frac{g_m R_D}{1 + R_D^2 C_L^2 \omega_0^2} - j \frac{g_m R_D^2 C_L \omega_0}{1 + R_D^2 C_L^2 \omega_0^2} \right)^3 \end{aligned} \quad (12)$$

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 поларним облику као

$$z = A(\cos\theta + j\sin\theta) \quad (13)$$

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:

$$z = Ae^{j\theta} \quad (14)$$

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

$$z^n = (Ae^{j\theta})^n = A^n (e^{j\theta})^n = A^n e^{jn\theta} \quad (15)$$

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 примјени на наш случај, важи следеће:

$$\begin{aligned} |H(j\omega_0)| &= \left| \frac{g_m R_D}{1 + R_D^2 C_L^2 \omega_0^2} - j \frac{g_m R_D^2 C_L \omega_0}{1 + R_D^2 C_L^2 \omega_0^2} \right|^3 = \\ &= \left( \sqrt{\frac{g_m^2 R_D^2}{(1 + R_D^2 C_L^2 \omega_0^2)^2}} + \frac{g_m^2 R_D^4 C_L^2 \omega_0^2}{(1 + R_D^2 C_L^2 \omega_0^2)^2} \right)^3 = \\ &= \left( \sqrt{\frac{g_m^2 R_D^2 (1 + R_D^2 C_L^2 \omega_0^2)}{(1 + R_D^2 C_L^2 \omega_0^2)^2}} \right)^3 = \\ &= \left( \frac{g_m R_D}{\sqrt{1 + R_D^2 C_L^2 \omega_0^2}} \right)^3 \end{aligned} \quad (16)$$

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

$$\left( \frac{g_m R_D}{\sqrt{1 + R_D^2 C_L^2 \omega_0^2}} \right)^3 = 1 \quad (17)$$

Одатле се може изразити  $\omega_0$  на следећи начин:

$$\omega_0 = \frac{\sqrt{g_m^2 R_D^2 - 1}}{R_D C_L} \quad (18)$$

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

$$\begin{aligned} \angle H(j\omega_0) &= 3 \cdot \arctan \left( \frac{-\frac{g_m R_D^2 C_L \omega_0}{1 + R_D^2 C_L^2 \omega_0^2}}{\frac{g_m R_D}{1 + R_D^2 C_L^2 \omega_0^2}} \right) = \\ &= 3 \cdot \arctan(-R_D C_L \omega_0) = \\ &= -3 \cdot \arctan(R_D C_L \omega_0) \end{aligned} \quad (19)$$

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

$$\arctan(R_D C_L \omega_0) = 60^\circ \quad (20)$$

Одатле се може представити  $\omega_0$  на следећи начин:

$$\omega_0 = \frac{\sqrt{3}}{R_D C_L} \quad (21)$$

Ако сада изједначимо изразе 18 и 21, добија се следеће:

$$\frac{\sqrt{g_m^2 R_D^2 - 1}}{R_D C_L} = \frac{\sqrt{3}}{R_D C_L} \quad (22)$$

Одатле се даље добија:

$$g_m R_D = 2 \quad (23)$$

Другим ријечима, сваки степен мора обезбједити појачање напона ниске фреквенције 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 спрегу на ниским фреквенцијама, што захтјева фазни помјерај од  $-180^\circ/N$  по једном степену да би систем осциловао. Тада ће  $\omega_0$  бити:

$$\omega_0 = \frac{1}{R_D C_L} \tan \frac{180^\circ}{N} \quad (24)$$

Како се  $N$  повећава,  $\omega_0$  опада јер свака фаза мора унијети мање фазног помјераја. Такође, у општем случају, како  $N$  расте, захтјевано нискофреквентно појачање се смањује што се може видјети из следећег израза:

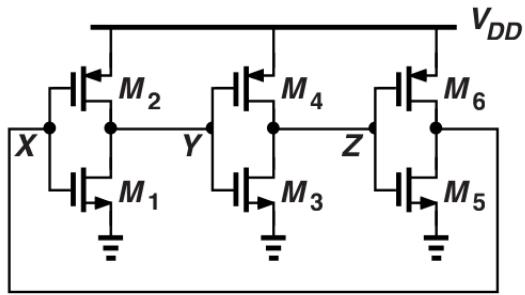
$$g_m R_D = \sqrt{\tan^2 \frac{180^\circ}{N} + 1} \quad (25)$$

Видјели смо да су таласни облици прстенастог осцилаотра близки 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 инверторски прстен обрађен у наредном потпоглављу.

#### 4.3.1 Инверторски прстен

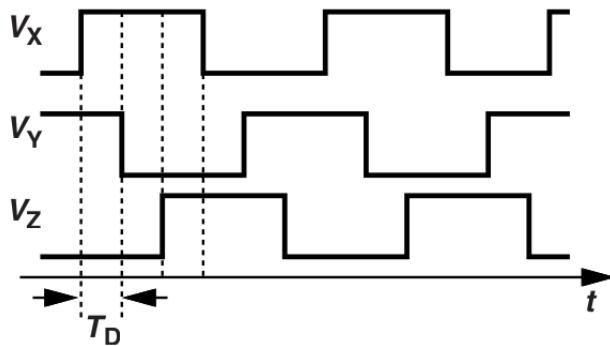
Најчешћа имплементација прстенастих осцилатара је коришћењем CMOS инвертора уместо отпорнички оптерећених појачавача као фаза појачања у прстену. Слика 14 приказује тростепени инверторски прстен. Ако претпоставимо да је у почетку  $V_X = V_Y = V_Z$ , свака фаза уводи напонско појачање од  $-(g_{mN} + g_{mP})(r_{ON}||r_{OP})$ , 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*).

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



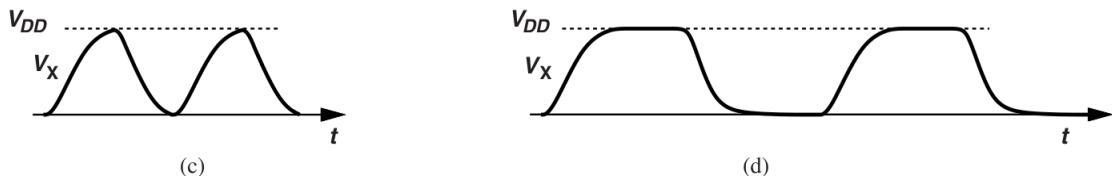
Слика 14: Тростепени прстенасти осцилатор састављен од CMOS инвертора.

$V_{DD}$ . Слика 15 приказује идеализоване облике сигнала на сва три степена у прстену, у току времена. Закључујемо да прелаз на једном чвиру узрокује прелаз на наредном након одређеног кашњења,  $t_d$ . Укупан период осциловања је стога једнак  $6t_d$ , па је фреквенција осциловања једнака  $f = 1/6t_d$ .



Слика 15: Облици сигнала на чворовима.

Стварни облици сигнала претходног осцилатора нису тако оштри као на Слици 15, али нису ни правилни као синусоида. Реалнији примјер је илустрован на Слици 16(a), 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 (Слика 16(б)).

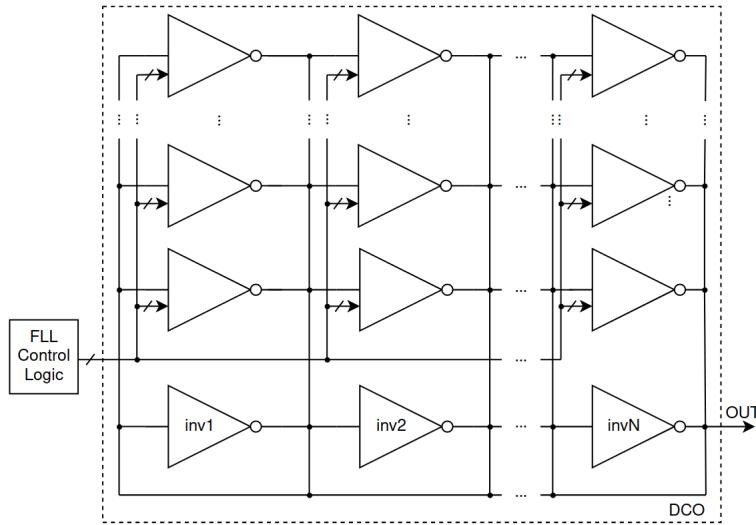


Слика 16: Реалан облик сигнала (а) тростепеног и (б) петостепеног прстенастог осцилатора.

Прстенасти осцилатори су уопштено прилично осјетљиви од вриједности напона напајања. Прво, кашњење инвертора се повећава како напон напајања расте, што значи да је фреквенција осциловања обратно пропорционална вриједности напона напајања. Друго, динамичка потрошња снаге прстенастог осцилатора има чак квадратну зависност од вриједности напона напајања, о чему ће више ријечи бити у поглављу 5.

## 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 [12]. 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 [13]. Ова матрица је састављена од  $N$ -фазних прстена тростатичких инвертора повезаних паралелно.  $N$  представља број DCO фаза (степени) и мора бити непаран број већи или једнак 3. Слика 17 приказује примјер  $N$ -степеног дигитално контролисаног осцилатора који има један прстен стално укључених инвертора.



Слика 17:  $N$ -степени дигитално контролисано осцилатор.

У физичком смислу, матрица може бити преобликована у квадрат, што омогућава једноставнију управљачку логику. Један или више прстенова су увијек укључени и дефинишу основну фреквенцију DCO-а. Остали тростатички инвертори се укључују и искључују у зависности од управљачке логике.

Формулa за фреквенцију осциловања DCO-а гласи:

$$f_{\text{osc}} = \frac{1}{2Nt_d} \approx \frac{I_d}{2NC_{\text{load}}V_{\text{DDL}}}, \quad (26)$$

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,  $C_{\text{load}}$  је капацитивно оптерећење истог инвертора, и  $V_{\text{DDL}}$  је напон напајања DCO-а. Производ  $Nt_d$  је помножен са 2 да би се добила читава периода такта, а не полупериода.

Када је ријеч о топологији, са повећањем броја DCO фаза (степени), фреквенцијски корак ( $K_{\text{DCO}}$ ) опада, чиме се повећава прецизност DCO-а. Максимална фреквенција 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 непромењени, повећање броја фаза не утиче на потрошњу снаге. Међутим, ако укупан број тростатичких инвертора остане непромењен 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 фреквенцији  $f_{\text{osc}}$  има динамичку потрошњу снаге која се може представити једначином

$$P = N f_{\text{osc}} C_{\text{tot}} V_{\text{DDL}}^2, \quad (27)$$

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

$$P = \frac{C_{\text{tot}} V_{\text{DDL}}^2}{2t_d}, \quad (28)$$

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 фреквенцију која не зависи од управљачке ријечи DCO-а. Тренутна излазна фреквенција ће стога у основи зависити од почетне фреквенције, фреквенцијског корака тј. резолуције и управљачке ријечи. Стога се може написати следећа линеарна зависност фреквенције осциловања, која може послужити за софтверску имплементацију рада DCO-а [14]:

$$f_{\text{DCO}} = f_0 + K_{\text{DCO}} \cdot d, \quad (29)$$

где су  $f_0$  почетна фреквенција,  $K_{\text{DCO}}$  резолуција фреквенције, а  $d$  је управљачка дигитална ријеч.

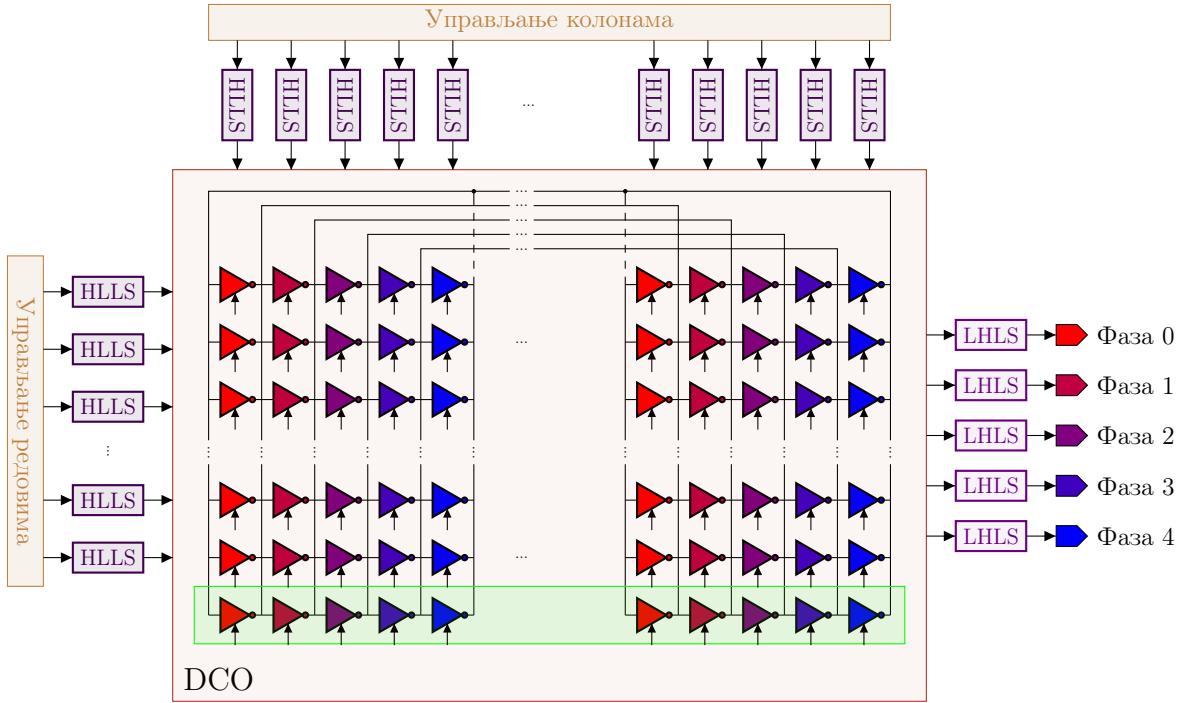
## 5.1 Архитектура предложеног DCO-а

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 фреквенција са довољном прецизношћу. Слика 18 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 у 18 редова и 15 колона. Управљачка логика FLL-а управља са 17 редова и свих 15 колона, што значи да постоји 255 фреквенцијских корака. Преостали ред са  $3 \times 5$  тростатичких инвертора је стално укључен и на њега не утиче управљачка логика FLL-а. Што се више тростатичких инвертора у свакој фази укључи под утицајем управљачке логике FLL-а, тренутна снага покретања (енгл. *Driving Strength*) једне фазе се повећава, док њено капацитивно оптерећење у суштини остаје константно, што резултује повећањем излазне фреквенције осциловања [4].

## 5.2 Блок дијаграм DCO ћелије

Да би се ефикасно подесила излазна фреквенција DCO-а, уведен је скуп управљачких улаза DCO-а, а то су сигнали *Row On*, *Row Select* и *Column Select*, поменути



Слика 18: Петостепени прстенасти дигитално контролисани осцилатор (DCO) састављен од тростатичких инвертора, са додатим претварачима напонских нивоа (HLLS и LHLS) и уоквиреним редом стално укључених тростатичких инвертора.

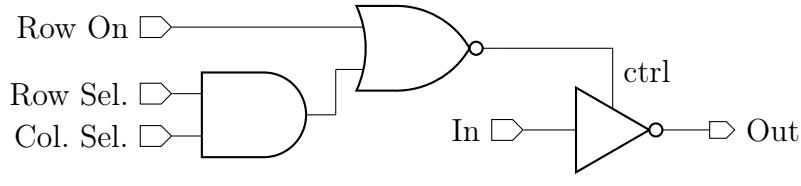
такође у поглављу 3.4. Према томе, сваки тростатички инвертор појединачно садржи сопствену управљачку јединицу у облику И-ИЛИ стандардне ћелије, и заједно граде већи блок назван ћелија DCO-а. Слика 19 приказује шему ћелије DCO-а на нивоу логичких кола и на нивоу CMOS транзистора.

У дигиталном домену сигнал Out може имати три стања. Једно је недефинисано, 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 сигнала 0, тростатички инвертор је укључен, а за вриједност 1 је искључен. Табела 3 је таблица истинитости управљачког сигнала ctrl.

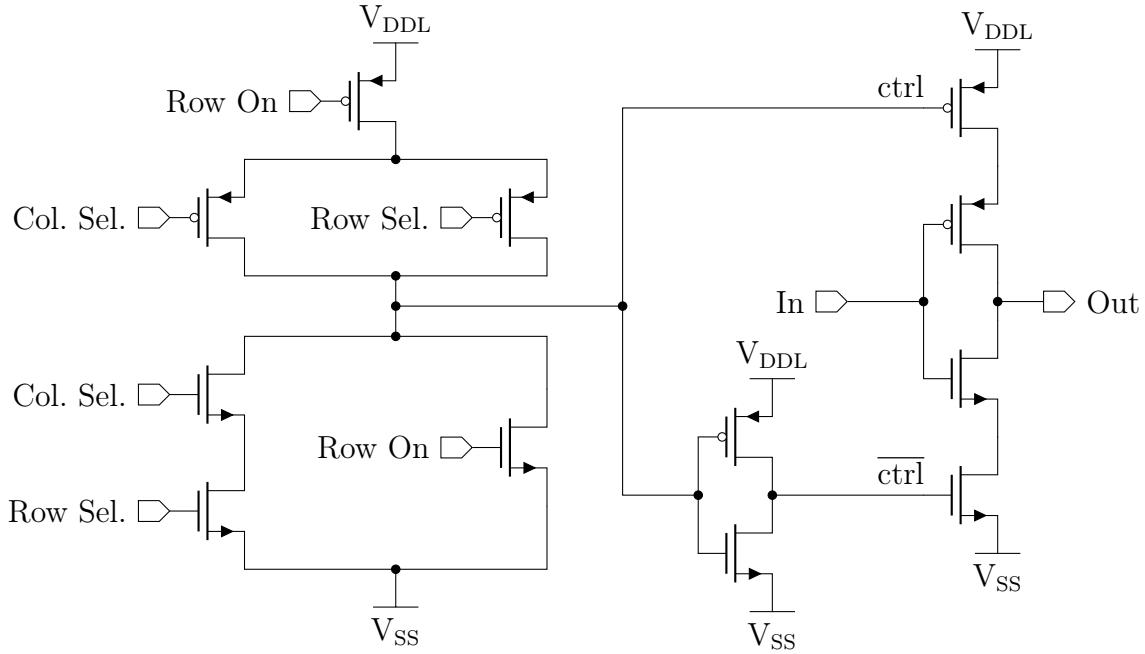
Табела 3: Таблица истинитости контролног сигнала тростатичког инвертора.

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

Логички израз за сигнал ctrl у зависности од управљачких сигналова редова и колона



(a)



(b)

Слика 19: Ћелија DCO-а на нивоу (а) логичких кола и (б) CMOS транзистора.

гласи:

$$\text{ctrl} = \overline{(\text{Row Sel.} \cdot \text{Col. Sel.})} + \text{Row On} \quad (30)$$

Табела 4 приказује табелу истинитости Out сигнала у зависности од управљачког сигнала ctrl и улазног In.

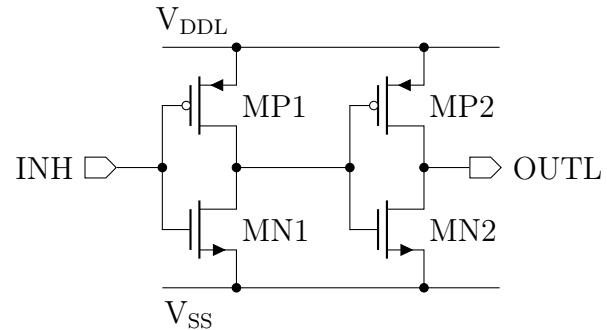
Табела 4: Таблици истинитости излазног сигнала тростатичког инвертора.

ctrl	In	Out
0	0	1
0	1	0
1	0	X
1	1	X

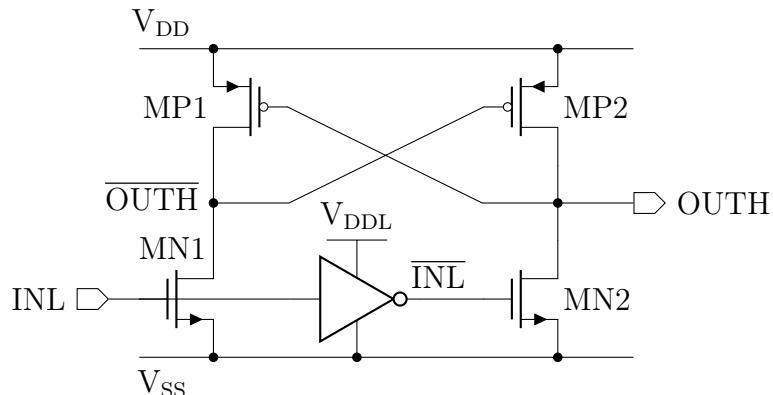
### 5.3 Претварачи напонског нивоа DCO-а

Напон напајања који користи дигитално контролисани осцилатор ( $V_{DDL}$ ) се у овом раду разликује од напона напајања који користи остатак логике FLL-а ( $V_{DD}$ ). Пред-

ност таквог дизајна је могућност подешавања напона напајања DCO-а независно на кон производње чипа, што даље омогућава накнадно постизање жељене резолуције фреквенције (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 расипања снаге (енгл. *Power Disipation*) 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 18.



(a)



(b)

Слика 20: Шема претварача (а) са високог на низак и (б) са ниског на висок напонски ниво [15].

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 висок напонски ниво (енгл. *Low-High Level Shifter*, LHLS). Слика 20 приказује шеме конвенционалних претварача напонског нивоа који су коришћени у претходно описаном систему. Као што се може видјети са Слике 20a, 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 20b приказује шему претварача са ниског на висок напонски ниво, представља регенеративно логичко коло засновано на позитивној повратној спрези [15] 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 сигнала INL и  $\overline{INL}$ , где је за инвертовање сигнала потребно додати инвертор.

С обзиром да је претварач са ниског на висок напонски ниво нешто комплекснији,

његов рад ће бити детаљније објашњен. Када напони сигнала INL и  $\overline{\text{INL}}$  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  $\overline{\text{OUTH}}$  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  $\overline{\text{INL}}$  једнаки  $V_{DDL}$  и 0, тада ће MN1 и MN2 бити укључен и искључен, респективно. MN1 тада повлачи на нулу напон чвора  $\overline{\text{OUTH}}$ , укључујући тако MP2, чиме се напон OUTH подиже на  $V_{DD}$ , а MP1 се искључује.

Треба запазити да се напон чвора OUT одређен покретачком струјом (енгл. *Drive Current*) транзистора MP2 и MN2, тј. струјом која тече кроз канал транзисотра када су укључени. Зато, ако је покретачка струја кроз транзистор MP2 много већа него кроз MN2, тада OUT не може бити испражњен што доводи до тога да претварач не може исправно радити у таквим ситуацијама [15]. 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 довести до сметњи у радном циклусу (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 сигнал додатно исправља како би такав дошао на улаз претварача са ниског на висок напонски ниво.

## 6 Имплементација и резултати симулација

У овом поглављу биће приказан начин имплементације система кроз хијерархијски приказ лејаута комплетног FLL-а. Затим ће бити приказани разултати 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 2 различите технологије. Дигитални FLL описан у овом раду, имплементиран је коришћењем SystemVerilog језика за опис хардвера у 130 nm CMOS технологији. Улазни референтни тakt је 16 MHz, док се тakt DCO-а подешава до 640 MHz, при чему је резолуција учестаности око 2.8 MHz у типичним условима рада. У симулацији под типичним условима рада, напон напајања управљачке логике FLL-а ( $V_{DD}$ ) је подешен на 1.2 V, а самог DCO-а ( $V_{DDL}$ ) на 1.1 V, и за њих је средња квадратна вриједност (енгл. *Root Mean Square*, RMS) потрошње струје 0.9 mA и 2.185 mA, респективно.

### 6.1 SystemVerilog код

У овом потпоглављу биће приказани набитнији дијелови имплементације FLL-а на нивоу SystemVerilog [16] језика за опис хардвера. Уз поједине дијелове кода ће бити објашњења у сврху лакшег разумјевања. 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 хијерархијски највишег FLL модула.

```
1 // Frequency-locked loop
2 module fll #(
3   parameter integer CW = 8 // control word width
4 )(
5   // Clock and reset
6   input          clk_ref_i,      // reference clock
7   input          clk_ref_en_i,   // reference clock enable
8   input          rst_ni,        // async reset
9   // FLL
10  input [CW-1:0] clk_mult_i,    // reference frequency multiplier
11    ↪ [40]
12  // DCO
13  input          dco_pwr_sel_i, // DCO clock output power supply
14    ↪ selection: VDDL/VDD [0/1]
15  input [CW-1:0] dco_ctrl_man_i, // manual DCO control word [200]
16  input [1:0]     dco_ctrl_mode_i, // DCO control mode: FIXED/MANUAL/
17    ↪ CNT/PID [3/2]
18  // PID
19  input [CW-1:0] pid_kp_i,      // PID controller proportional
20    ↪ constant [0b01000000]
21  input [CW-1:0] pid_ki_i,      // PID controller integral constant
22    ↪ [0b01000000]
23  input [CW-1:0] pid_kd_i,      // PID controller derivative
24    ↪ constant [0]
25  // FLL
```

```

20 | input [ 3:0] lock_cnt_i,           // FLL lock counter condition [0
21 |    $\rightarrow b1000$ 
22 | output          lock_o,        // FLL lock signal
23 | // DCO output
24 | output          clk_dco_o    // DCO clock (one phase)
25 |
26 | );
...
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 XOR операције између свих битовима 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 XOR операције, тј.  $0000010 \wedge 0000101$ , 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         $\rightarrow$  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
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 `fll` модула и дата је у Листингу 3. Семплована вриједност Грејевог бројача је на магистралама `cnt_dco_gray_q2`, док је природна бинарна вриједност на магистралама `cnt_dco` и она суштински представља тренутни умножак фреквенције добијене на излазу DCO-а, и представља податак спреман за поређење са жељеним умношком са улаза, унутар блока управљачке логике (у PID или bang-bang режиму). Сваки бит у тој магистрали се добија као резултат логичке XOR операције над свим битовима Грејевог записа, помјереног онолико мјеста удесно колико износи тежина тренутног бита у `cnt_dco` магистралама (имплементација на линији 7 у Листингу 3).

Листинг 3: Претварање Грејевог кода у природну бинарну вриједност.

```

1 module fll #(
2   ...
3   // Gray to binary conversion
4   genvar i;
5   generate
6     for(i=0; i<CW; i++) begin: gen_gray2bin_conv
7       assign cnt_dco[i] = ^(cnt_dco_gray_q2 >> i);
8     end: gen_gray2bin_conv
9   endgenerate
10  ...
11 endmodule

```

Bang-bang (или On-Off) контролер, 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 фреквенције, 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 PID контролера). Закључује се да се поменути сигнали прослијеђују даље у блок бројача, чија је имплементација, као посебног `cnt` модула, приказана у Листингу 5. Сигнали `cnt_ref_up_dn` и `cnt_ref_en` се при инстанцирању `cnt` модула прослијеђују на његове `up_down_i` и `en_i` улазе, респективно.

Листинг 4: Компараторски дио bang-bang контролера.

```

1 module fll #(
2   ...
3   // Comparator (TODO: sequential logic?)
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'
7     ↪ b0;
8   assign cnt_ref_en    = (clk_mult_i != cnt_dco_q) ? ~lock_o : 1'
9     ↪ b0;
10  ...
11
12 endmodule

```

Листинг 5: Имплементација бројача из bang-bang контролера.

```

1 // Up-down counter with enable
2 module cnt #(
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 PID контролера, 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 контролера, ту су још улази:

- `setpoint_i` који представља жељени умножак фреквенције (при инстанцирању модула прослијеђује му се вриједност `clk_mult_i` са улаза `f11` модула),
- `feedback_i` који представља семпловани тренутни умножак (при инстанцирању модула прослијеђује му се семплована вриједност `cnt_dco` из Листинга 3) и

- `lock_cnt_i`, који означава колико узлазних ивица такта грешка мора узастопно 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 bang-bang контролера). Он означава 8-битну цјелобројну вриједност и добија се издвајањем 8 бита од позиције 13 до позиције 6 из промјенљиве `ctrl_pid` (линија 64 у Листингу 6) у којој је смјештена сумма пропорционалног, интегралног и диференцијалног дејства. Такво издвајање 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 PID контролера. Наиме, улазни коефицијенти `kp_i`, `ki_i` и `kd_i` су 8-битни у формату Q6.2 (6 бита за цјелобројни дио, 2 за децимални дио), док је формат `ctrl_pid` промјенљиве Q12.6 (12 битова за цјелобројни дио, 6 за децимални дио), па се поменутим одсијецањем врши заокруживање на 8-битни цијели број.

Листинг 6: Имплементација PID контролера.

```

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
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) + 18'(
63             → deriv);
64     end
65 end
66 assign ctrl_o = ctrl_pid[13:6];
67
68 endmodule

```

Излази `cnt_o` и `ctrl_o` из bang-bang и PID контролера, редом, представљају суштински број тростатичких инвертора DCO-а који требају бити укључени, што утиче на његову излазну фреквенцију. Међутим, та вриједност се не прослијеђује директно у модул DCO-а, већ се преко управљачког докодера анализираног у поглављу 3.4 претвара у скуп одговарајућих векторских улаза за управљање DCO-ом тј. редовима и колонама тростатичких инвертора унутар њега. Тада поступак претварања је приказан у Листингу 7, а имплементиран је у оквиру `f11` модула.

Листинг 7: Имплементација управљачког докодера 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          row_sel_d[r-1] = 1'b1;
30        end else begin
31          row_sel_d[r-1] = 1'b0;
32        end
33      end
34    end
35    ...
36  endmodule

```

Сигнали `row_on_d`, `row_sel_d` и `col_sel_d` се синхронизују на одговарајући такт и прослијеђују се као улази у DCO модул који је имплементиран као посебан блок у алатима за аналогно пројектовање, па стога овдје нема свој опис у SystemVerilog-у, већ се тај блок само инстанцира унутар `fll` модула, као што је приказану у Листингу 8. Као што се може видјети, `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 DCO блока.

```

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

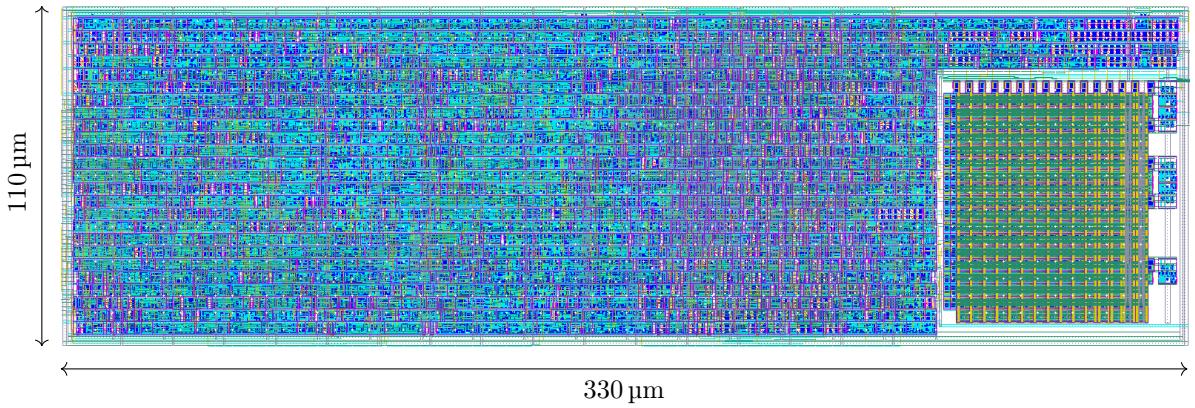
```

## 6.2 Лејаут

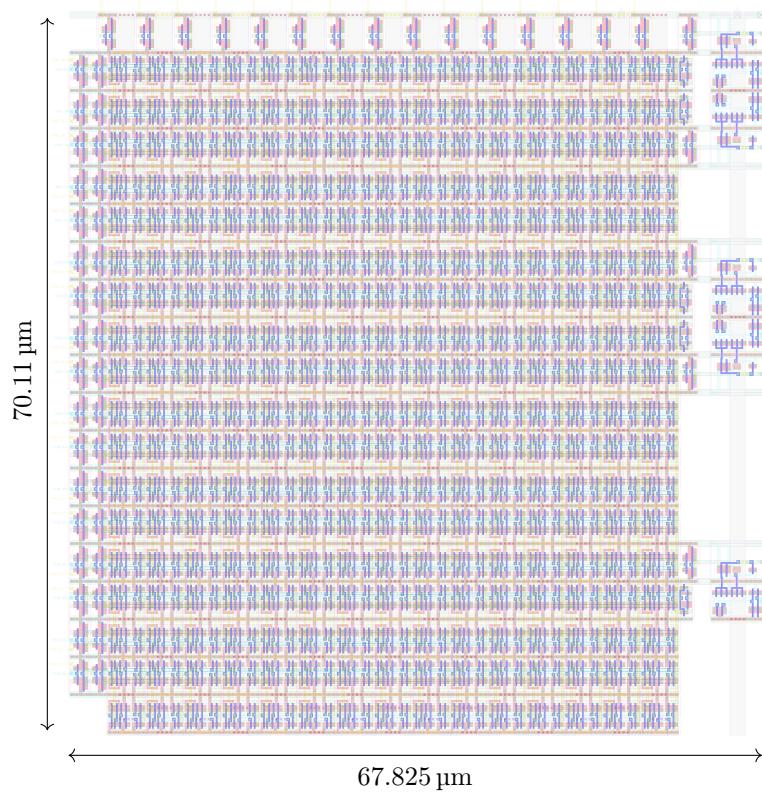
За израду лејаута намјенски је коришћена 7T библиотака стандардних ћелија, где 7T означава да висина свих ћелија тачно толике висине да може да се вертикално провуче 7 линија метала минималне димензије и са минималним размаком. То до-приноси лакшем аутоматском (а и ручном) рутирању и слагању ћелија у компактну цјелину у сврху оптималне искористивости простора. Лејаут коначне верзије цјелокупног FLL-а приказан је на Слици 21. Читав дизајн FLL-а заузима  $36300 \mu\text{m}^2$ . Компонента у доњем десном углу је дигитално контролисани осцилатор, док остатак лејаута углавном припада управљачкој логици FLL-а.

DCO је имплементиран као независна компонента коришћењем библиотека стандардних ћелија. Од читавог лејаута FLL-а, око 13 % заузима DCO, тачније око  $4755 \mu\text{m}^2$ . Слика 22 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 23 приказује лејаут једне ћелије DCO-а. Са десне стране ћелије је оријентисан тростатички инвертор, док је са лијеве стране управљачки И-ИЛИ дио. Површина једне ћелије је око  $13.7 \mu\text{m}^2$ .

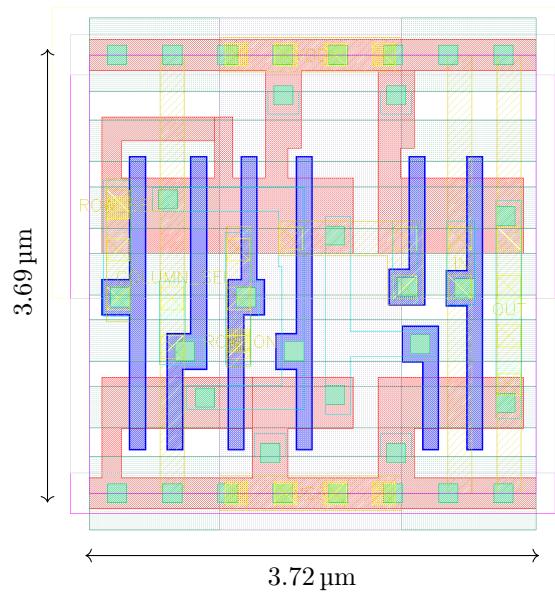


Слика 21: Лејаут дигиталног FLL-а, са DCO-ом у доњем десном углу.

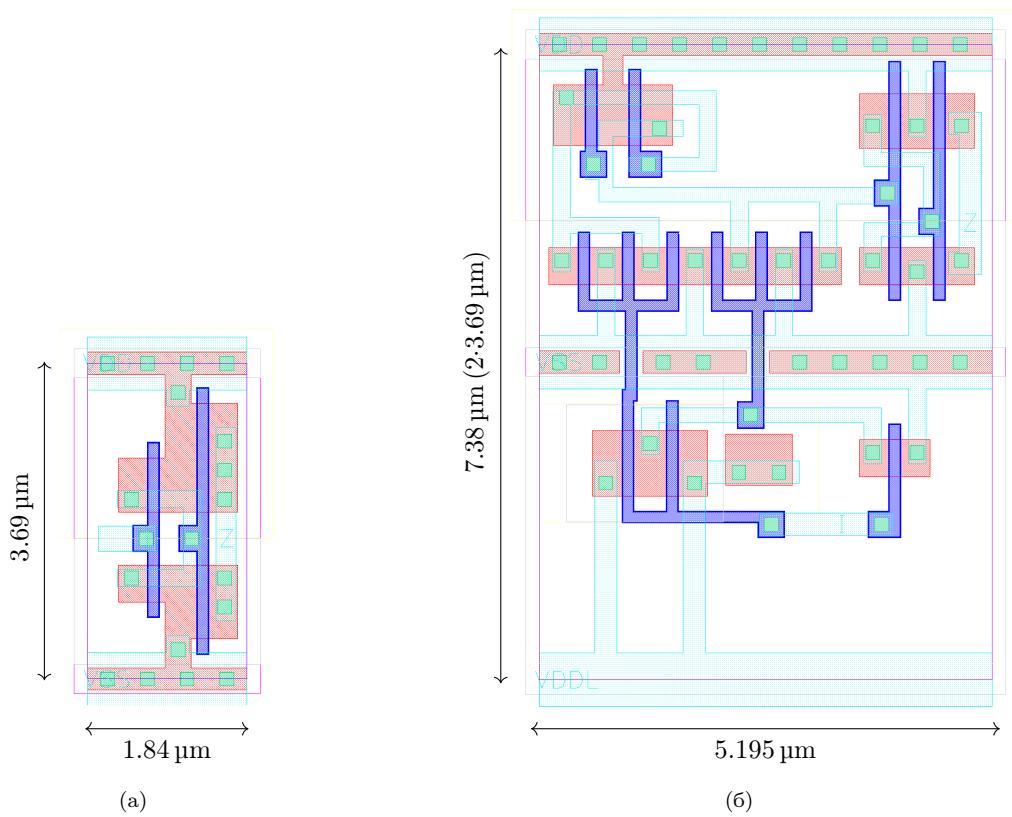


Слика 22: Лејаут DCO-а са претварачима напонских нивоа.

Лејаут претварача са високог на низак и са ниског на висок напонски ниво је приказан на Слици 24а и 24б, респективно. 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 23: Лејаут ћелије DCO-а.

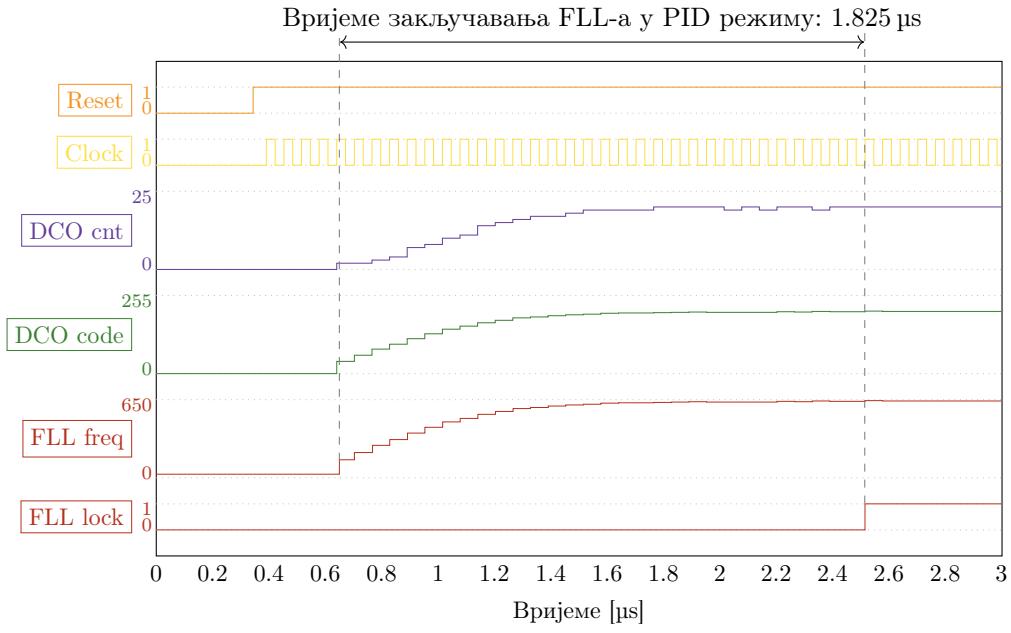


Слика 24: Лејаут претварача са (а) високог на низак и (б) ниског на висок напонски ниво.

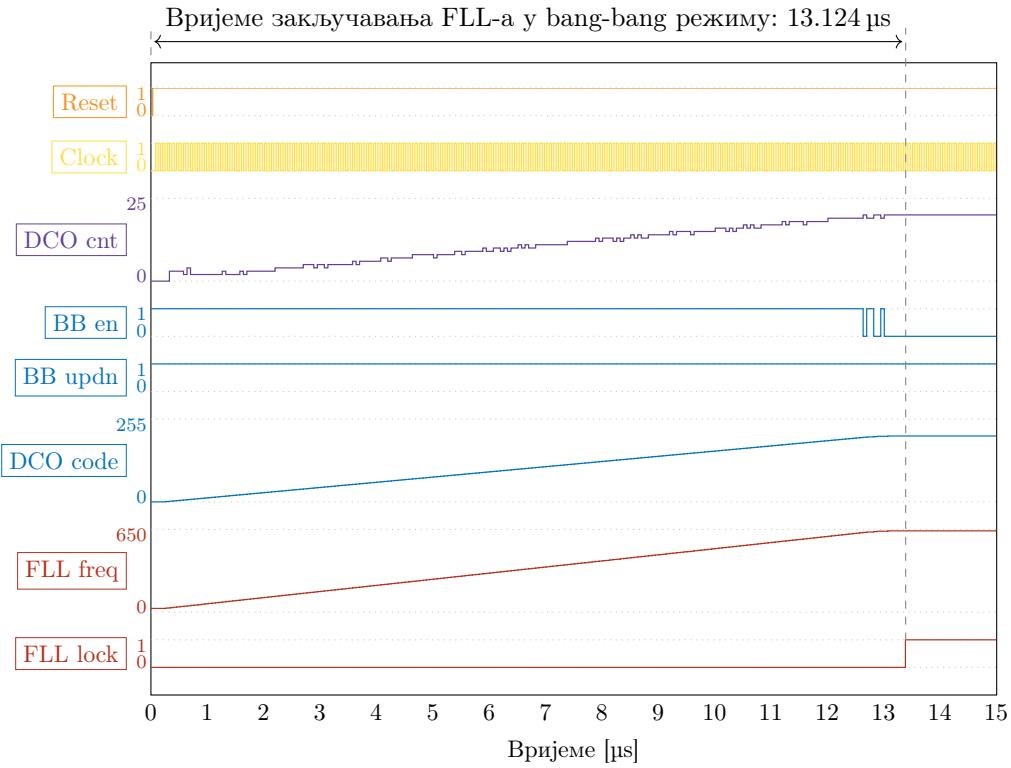
### 6.3 Симулација рада фреквенцијски затворене петље

Симулација рада FLL-а у оба управљачка режима приказана је на Слици 25. Са приказаних графика се може видјети да се у оба управљачка режима достиже стабилно закључавање FLL-а са излазном фреквенцијом веома близкој траженој фреквенцији, при чему грешка може бити у опсегу просјечне вриједности резолуције DCO-а (испод

3 MHz). Међутим, иако се у оба режима исправно достиже жељена фреквенција, између њих постоји разлика у брзини достизања исте. Тако, у PID режиму ( $P=15$ ,  $I=15$ ,  $D=0$ ), закључавање се одвија много брже, и то за  $1.83 \mu\text{s}$ , док је, поређења ради, за закључавање у bang-bang режиму потребно  $13.13 \mu\text{s}$ .



(a)



(б)

Слика 25: Симулација FLL-а у (а) PID режиму и (б) bang-bang режиму.

## 6.4 PVT зависност дигитално контролисаног осцилатора

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

1. Најспорији случај (slow-slow процесни угао,  $V_{DD}=1.1\text{ V}$ ,  $V_{DDL}=1\text{ V}$ ,  $125^\circ\text{C}$ )
2. Типичан случај (typical-typical процесни угао,  $V_{DD}=1.2\text{ V}$ ,  $V_{DDL}=1.1\text{ V}$ ,  $40^\circ\text{C}$ )
3. Најбржи случај (fast-fast процесни угао,  $V_{DD}=1.3\text{ V}$ ,  $V_{DDL}=1.2\text{ V}$ ,  $-40^\circ\text{C}$ )

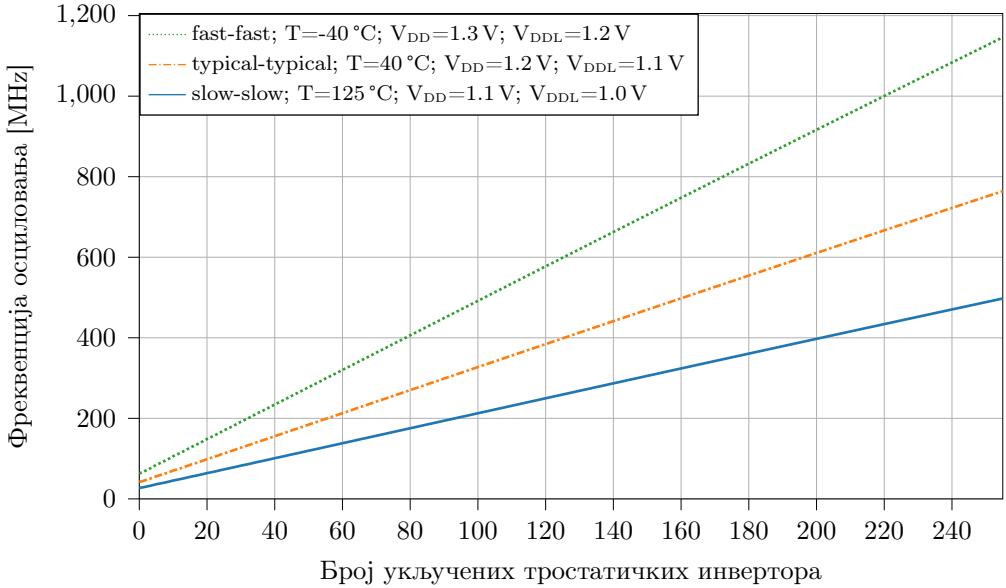
Да би се значајно смањило вријеме потребно за добијање резултата из симулација након екстракције лејаута, и то без губљења прецизности добијених резултата, симулације DCO-а су извршене на нивоу појединачно екстрактованих ћелија DCO-а и екстрактованих претварача напонског нивоа, са додатним паразитним кондензатором на свакој фази DCO-а. Улога тих кондензатора је да надомјесте паразитне капацитивности које се јављају у потпуно екстрактованом 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 на нивоу читавог дизајна.

Слика 26 приказује зависност фреквенције DCO-а и фреквенцијског корака тј. резолуције фреквенције ( $K_{DCO}$ ) од броја укључених тростатичких инвертора (тј. од управљачке ријечи DCO-а), за најгори, типични и најбољи PVT случај. Табела 5 приказује фреквенцијски опсег DCO-а и просјечну вриједност фреквенцијског корака ( $K_{DCO}$ ) у три претходно обрађена случаја. Под опсегом се подразумјевају минималне и максималне вриједности фреквенције која се добија на излазу DCO-а за укључен минималан (у овом случају 15) и максималан број (у овом случају 255) тростатичких инвертора DCO-а, респективно.

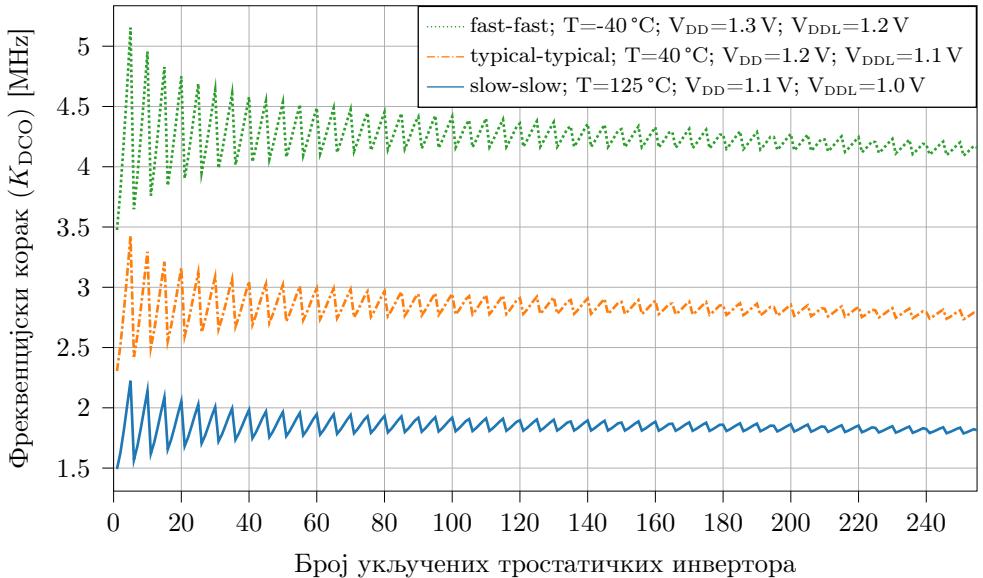
Табела 5: Фреквенцијски опсег DCO-а и просјечна вриједност фреквенцијског корака ( $K_{DCO}$ ).

Случај	$f_{\min}$	$f_{\max}$	Просјечан $K_{DCO}$
Најспорији	27.2 MHz	502 MHz	1.8 MHz
Типичан	42 MHz	764 MHz	2.8 MHz
Најбржи	63.3 MHz	1.146 GHz	4.2 MHz

Иако у најгорем случају фреквенција не достиже жељених 640 MHz, она ипак може бити достигнута повећањем засебног напона напајања осцилатора ( $V_{DDL}$ ). Такође је могуће анализирати и зависност фреквенције од свих PVT параметара независно, међутим те зависности нису приказане у раду због временске захтјевности извешавања потребних симулација, као и због већ приказаних резултата симулација у најгорем, типичном и најбољем 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 фреквенција повећава са бржим процесним углом, већим напоном напајања, као и мањом температуром.



(а)

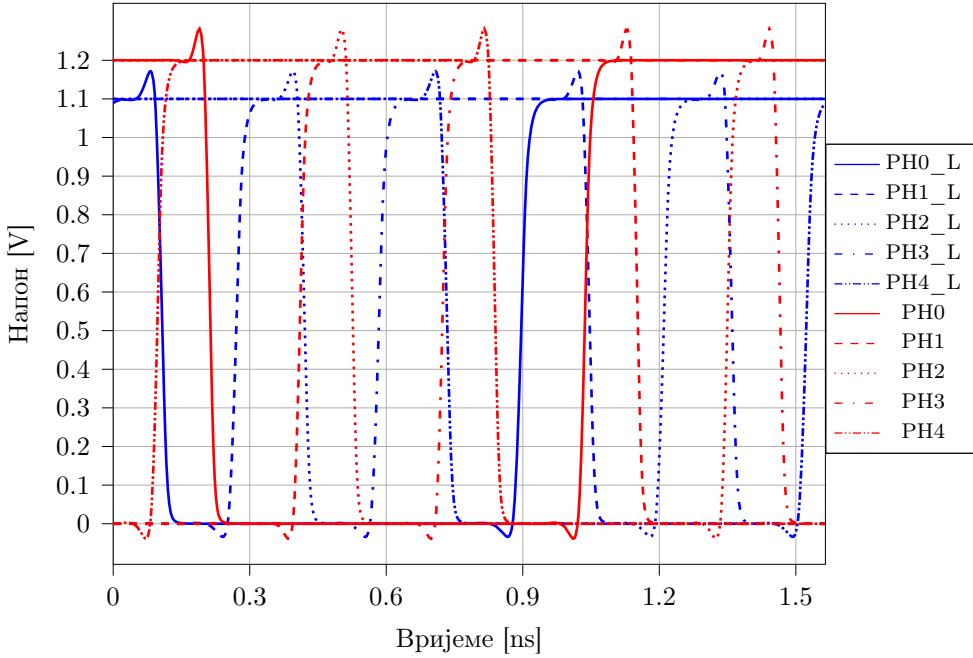


(б)

Слика 26: Зависност (а) фреквенције осциловања и (б) фреквенцијског корака од броја укључених тростатичких инвертора за најспорији, типични и најбржи случај.

## 6.5 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 симулација је рад потпуно екстракованог DCO-а у типичним условима. Слика 27 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 26, фреквенција 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 ( $t_{lh}$ )



Слика 27: Генерисани такт на свих 5 фаза DCO-а у  $V_{DD}$  и  $V_{DDL}$  доменима напајања.

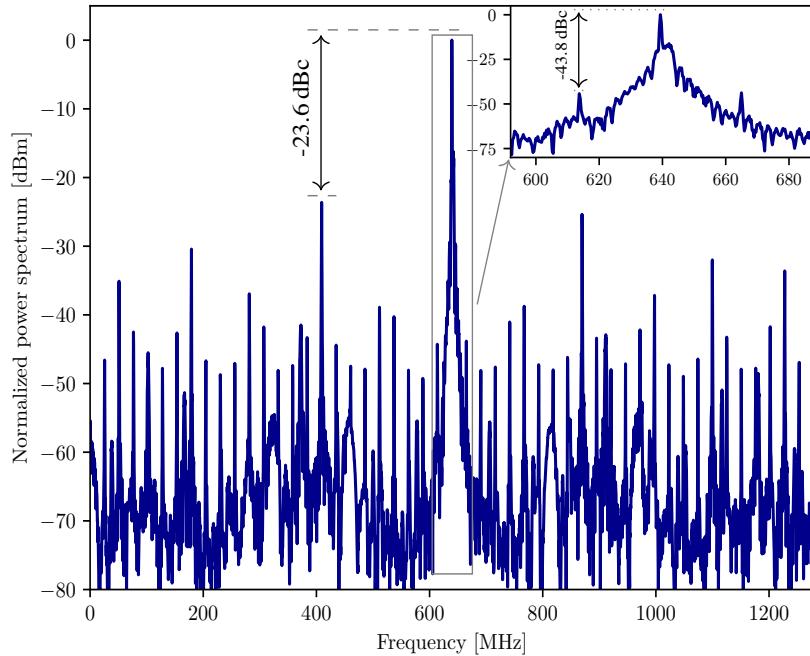
и  $t_{hl}$ ). Други важан параметар за разматрање је радни циклус тј. однос времена које излазни сигнал проведе на високом, односно ниском, напону у једној периоди такта и самог периода такта. Изражава се у процентима тј. колико процентуално сигнал такта проведе на високом, а колико на ниском напону тј. нули и идеално је да тај однос буде 50% – 50%.

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

## 6.6 Спектар снаге фреквенцијски затворене петље

Нормализовани спектар снаге предложеног FLL-а у PID режиму, током процеса одржавања вриједности фреквенције од 640 MHz на излазу, приказан је на Слици 28. Подаци са графика су добијени из 15 μm дигиталне симулације на нивоу логичких кола, која укључује идеалан бешумни модел DCO-а написан у SystemVerilog-у. Излаз симулације је потпуно детерминистички квадратни талас састављен од јединица и нула, одабран са учестаношћу 10 GHz, и сачуван за даљу обраду.

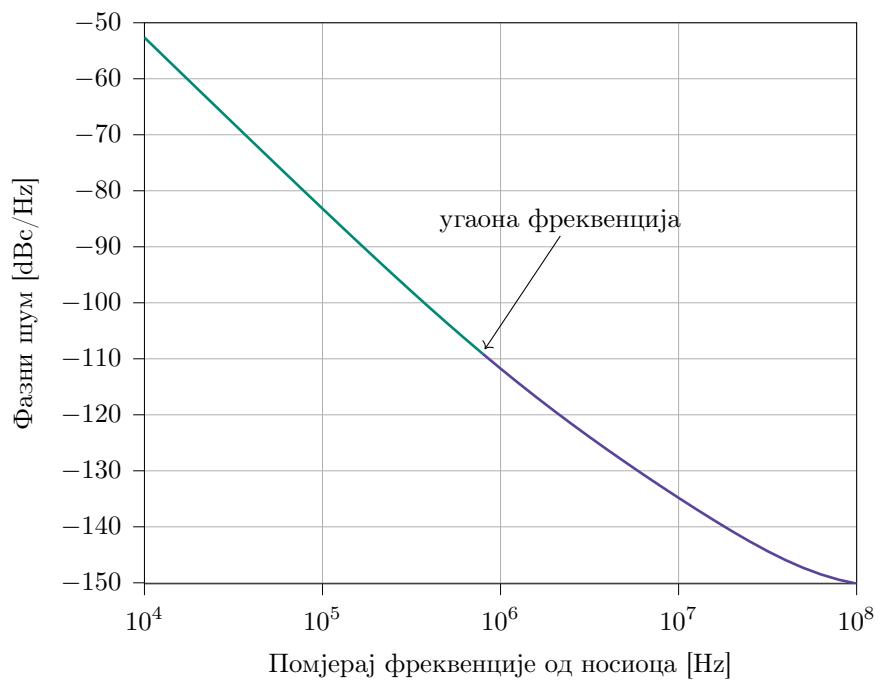
Највећа вриједност на графику представља фреквенцију осциловања FLL-а, односно фреквенцију носиоца (енгл. *Carrier Frequency*). Ниво највећег непожељног тона је  $-23.6$  dBc, која је више од два реда величине испод нивоа фреквенције носиоца ( $0$  dBc). У зумиреној области, може се видјети непожељан сигнал унутар референтне фреквенције и то лијево и десно од носиоца са помјерајем (енгл. *Offset*) од 16 MHz, чији је најјачи ниво на  $-43.8$  dBc.



Слика 28: Спектар снаге FLL-а, са фреквенцијом носиоца  $640\text{ MHz}$ , најјачим нивоом непожељног сиглана (у средини лијево) на  $-23.6\text{ dBc}$ , и највећим непожељним сигналом референтне фреквенције унутар  $640 \pm 16\text{ MHz}$  (горе десно, зумирано) на нивоу  $-43.8\text{ dBc}$ .

## 6.7 Фазни шум дигитално контролисаног осцилатора

Због насумичних фазних одступања, спектар снаге реалног осцилатора се такођешири на фреквенције око фреквенције носиоца,  $f_{\text{osc}}$ . Извори насумичних фазних поремећаја [14] у смислу шума треперења (енгл. *Flicker Noise*) и топлотног шума (енгл. *Thermal Noise*) се манифестишу као  $1/f^3$  and  $1/f^2$  области, респективно, у симулираном профилу фазног шума на Слици 29. Приказани резултати су добијени у типичним PVT условима, са вриједношћу напона напајања DCO-а ( $V_{\text{DDL}}$ ) од  $1.1\text{ V}$ , и вриједности фреквенције носиоца ( $f_{\text{osc}}$ ) од  $640\text{ MHz}$ .



Слика 29: Спектар фазног шума DCO-а: област шума треперења (лијево), област топлотног шума (десно), и угаона фреквенција од  $\approx 800$  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 фреквенцију 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 PID 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 PID контролера (Листинг 10).
- `cnt` - Класа бројача из bang-bang контролера (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 29 што се може видjetи у Листингу 12.

Листинг 9: Имплементација модела FLL компоненте.

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

```

```

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

```

Листинг 10: Имплементација класе PID контролера.

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

```

47
48     if (not self.rst_ni):
49         prop = 0
50         self.integ = 0
51         deriv = 0
52         self.err_prev = 0
53         self.ctrl_pid = 0
54         self.lock_o = 0
55     else:
56         prop = self.kp_i * self.err
57         self.integ += self.ki_i * self.err
58         deriv = self.kd_i * err_diff
59         self.err_prev = copy.deepcopy(self.err)
60         if ((self.err == 0) and (self.err_eq0_cnt == self.
61         ↳ lock_cnt_i) and (self.lock_cnt_i != 0)):
62             self.lock_o = 1
63         if (not self.lock_o):
64             self.ctrl_pid = prop + self.integ + deriv
65
       self.ctrl_o = int(format(int(self.ctrl_pid), '0'+str(
66         ↳ CTRL_WIDTH)+'b')[FX_INT_WIDTH-INT_WIDTH:FX_INT_WIDTH], 2)

```

Листинг 11: Имплементација класе бројача bang-bang контролера.

```

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

```

Листинг 12: Имплементација класе дигитално контролисаног осцилатора.

```

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

```

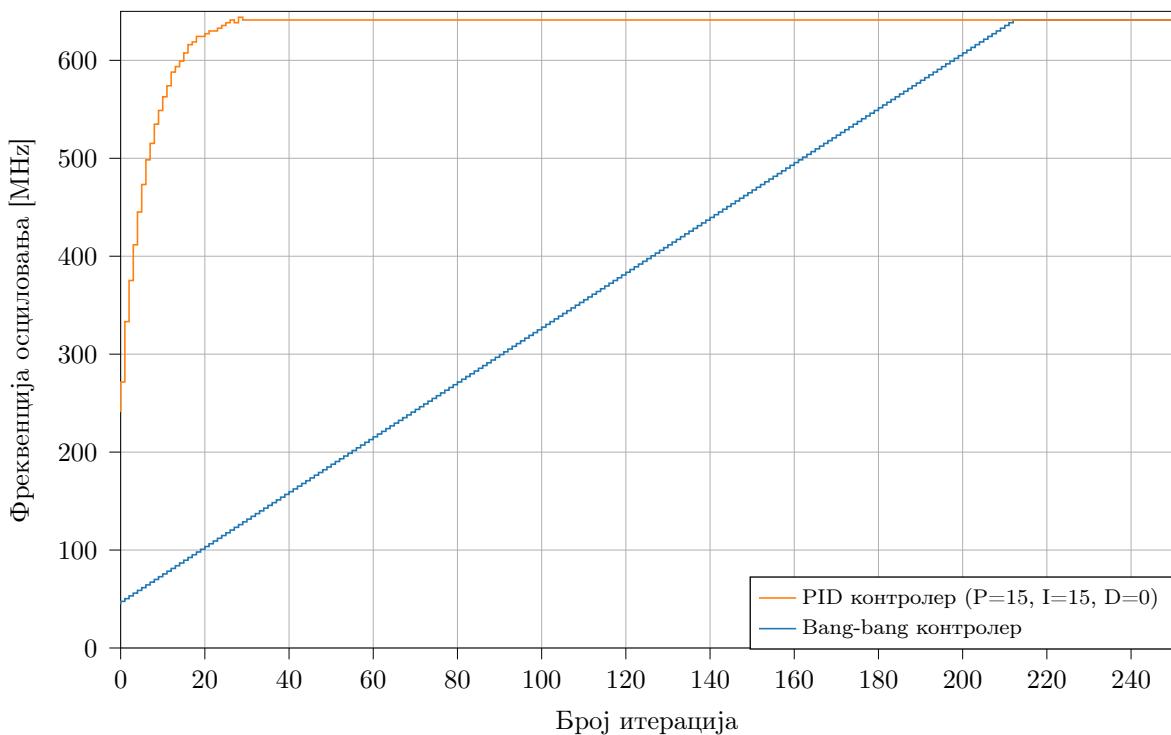
```

8
9     def run(self,
10         ctrl):
11         self.freq_out = self.init_freq + self.k_dco * ctrl

```

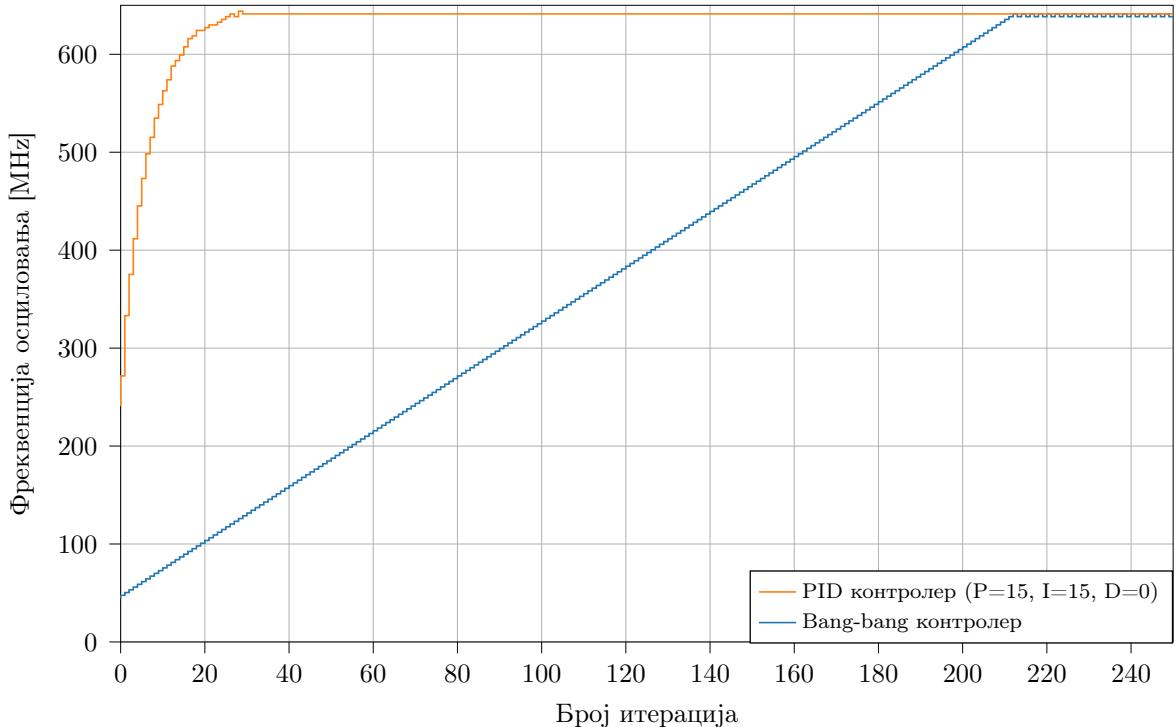
## 7.2 Поређење рада различитих управљачких режима

Слика 30 приказује одзив софтверског модела FLL-а при раду у PID и bang-bang управљачком режиму рада. У конкретној симулацији коришћено је понашање DCO-а добијено симулацијом у типичним условима рада, па су при инстанцирању софтверског модела FLL-а коришћени подаци из Табеле 5, односно  $K_{DCO}=2.8\text{ MHz}$  и  $f_{min}=42\text{ MHz}$  за почетну фреквенцију. Конкретна вриједност фреквенције која се овом симулацијом достиже у оба режима је  $641.2\text{ MHz}$ . У PID режиму модел до те фреквенције стиже након 28 итерација, док му је са bang-bang управљачким режимом потребно 214 итерација.



Слика 30: Одзив модела у PID и bang-bang режиму рада.

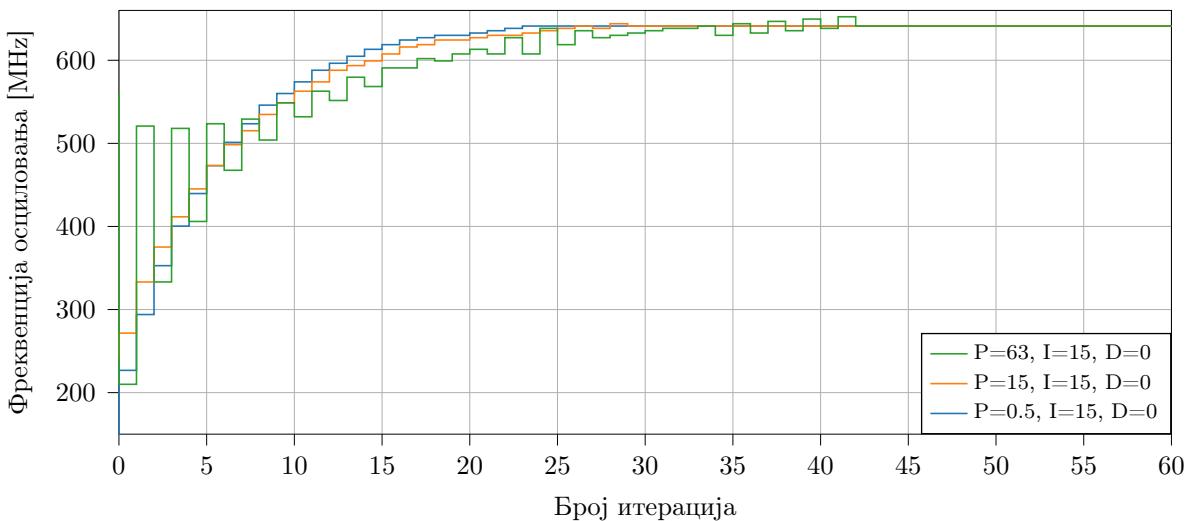
У поглављу 3.2 поменута је појава осциловања фреквенције у стабилном стању када FLL ради у bang-bang режиму. На Слици 30 те осцилације не постоје, а проблем је решен условним довођењем `pid.lock_o` сигнала на `cnt.en_i` улаз бројача bang-bang контролера. Логика имплементације се налази на линији 75 Листинга 9. Да би се ипак показало како те осцилације изгледају, довољно је само закоментарисати поменуту линију, а одкоментарисати линију 74. Тиме се добија одзив као на Слици 31, где фреквенција у стабилном стању, при коришћењу bang-bang режима, варира између  $638.4\text{ MHz}$  и  $641.2\text{ MHz}$  (примјећујемо да разлика између поменутих вриједности представља вриједност резолуције,  $K_{DCO}=2.8\text{ MHz}$ ).



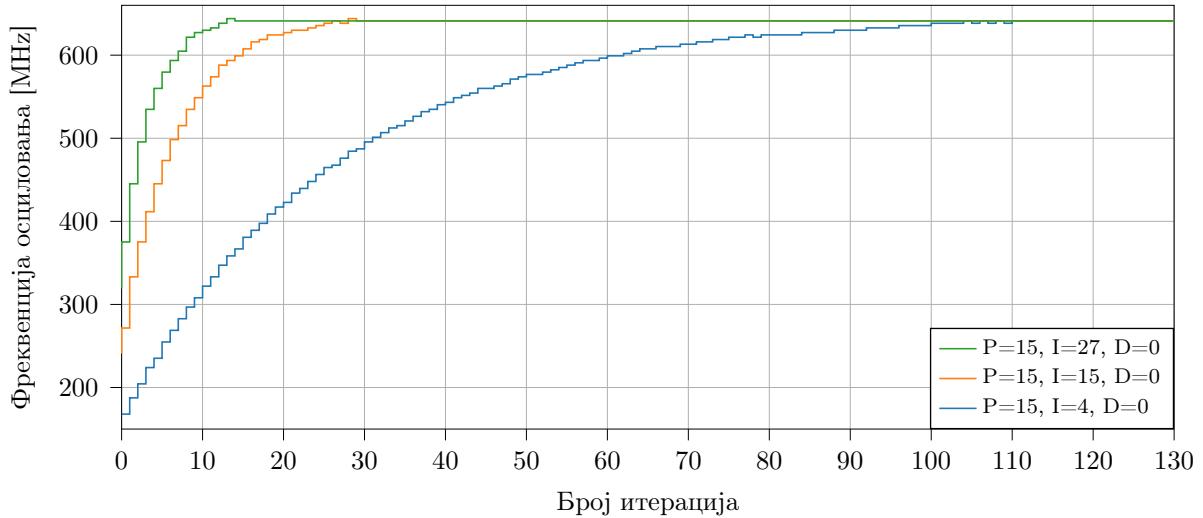
Слика 31: Одзив модела у PID и bang-bang режиму рада.

### 7.3 Утицај константи PID контролера на одзив система

У овом поглављу биће софтверски симулиран утицај различитих вриједности константи PID контролера на одзив система. Треба напоменути да одзив не мора у потпуности одговарати стварном одзиву система већ му је циљ помоћи при исправном одабиру константи PID контролера, 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. Слике 32 и 33 редом приказују утицај различитих вриједности пропорционалне ( $K_p$ ) и интегралне ( $K_i$ ) константе на брзину и начин достизања жељене фреквенције.



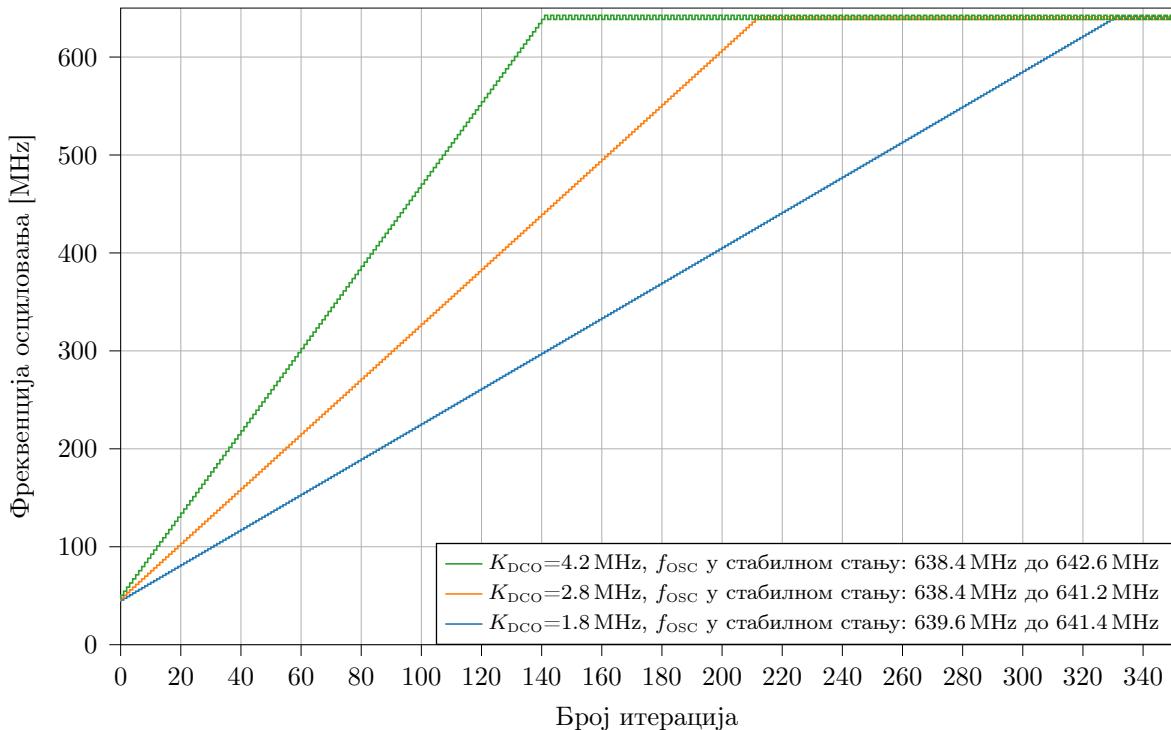
Слика 32: Одзив модела за различите вриједности пропорционалне константе PID контролера.



Слика 33: Одзив модела за различите вриједности интегралне константе PID контролера.

## 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 на различите карактеристике DCO-а. Тако ће у наставку бити приказан одзив за различите вриједности резолуције и то само у bang-bang режиму рада због поједностављења приказа. Слика 34 приказује одзив за три различите  $K_{DCO}$  вриједности при почетној фреквенцији 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 34: Моделован одзив система за различите вриједности резолуције,  $K_{DCO}$ .

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

## 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 фреквенције и брзог радног учинка. Додатне предности су му и једноставно управљање, преносивост и прилагодљивост. Представљена су два управљачка режима FLL-а: bang-bang контролер и PID контролер. Детаљно је објашњена обрада у различитим доменима такта, као и заштита од метастабилности.

FLL је потпуно синтетизабилан, имплементиран коришћењем SystemVerilog језика за опис хардвера. У зависности од технолошке библиотеке може бити направљен коришћењем само библиотеке стандардних ћелија без преуређивања у току коришћења методологије тока дигиталног пројектовања. Обрађени FLL је послат на производњу у 130 nm CMOS технологији.

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 је фазно спрегнута петља (енгл. *Phase-Locked Loop*, PLL), временско-дигитални претварач (енгл. *Time-to-Digital Converter*, TDC), регулатор с ниским падом напона (енгл. *Low-Dropout Regulator*, LDO) итд.

## Литература

- [1] Imran Ali и др., „An Ultra-Low Power, Adaptive All-Digital Frequency-Locked Loop With Gain Estimation and Constant Current DCO”, у: *IEEE Access* 8 (2020.), стр. 97215–97230, DOI: 10.1109/ACCESS.2020.2995853.
- [2] Ahmed Musa и др., „A Compact, Low-Power and Low-Jitter Dual-Loop Injection Locked PLL Using All-Digital PVT Calibration”, у: *IEEE Journal of Solid-State Circuits* 49.1 (2014.), стр. 50–60, DOI: 10.1109/JSSC.2013.2284651.
- [3] Wei Deng и др., „A Fully Synthesizable All-Digital PLL With Interpolative Phase Coupled Oscillator, Current-Output DAC, and Fine-Resolution Digital Varactor Using Gated Edge Injection Technique”, у: *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”, у: *IEEE Journal of Solid-State Circuits* 43.1 (2008.), стр. 42–51, DOI: 10.1109/JSSC.2007.910966.
- [5] A. V. Rylyakov и др., „A Modular All-Digital PLL Architecture Enabling Both 1-to-2GHz and 24-to-32GHz Operation in 65nm CMOS”, у: *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] Reddy B. Madhusudhana и др., „CMOS Based Digital Controlled Oscillators (DCO) – A Review”, у: *International Journal of Applied Engineering Research* 10.20 (2015.), стр. 18626–18630.
- [13] M. Terosiet и др., „A Comprehensive In-Depth Study of Tri-State Inverter Based DCO”, у: *Microelectronics Journal* (2020.), DOI: 10.1016/j.mejo.2020.104760.
- [14] 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.
- [15] Yuji Osaki и др., „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.

- [16] „IEEE Standard for SystemVerilog—Unified Hardware Design, Specification, and Verification Language”, y: *IEEE Std 1800-2023 (Revision of IEEE Std 1800-2017)* (2024.), ctp. 1–1354, DOI: [10.1109/IEEESTD.2024.10458102](https://doi.org/10.1109/IEEESTD.2024.10458102).