RĪGAS TEHNISKĀ UNIVERSITTĀTE

ELEKTRONIKAS UN TELEKOMUNIKĀCIJU FAKULTĀTE

[](https://www.google.com/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwj9zKXEs6fgAhWF1iwKHceJAlAQjRx6BAgBEAU&url=https://lv.wikipedia.org/wiki/Att%C4%93ls:RTU_logo_2017.svg&psig=AOvVaw2YftzNrZHl3gHZWIEXy3Ih&ust=1549552627423236)

Funkcionālo un loģisko shēmu modelēšana, RTR532

Laboratorijas darbs Nr.1

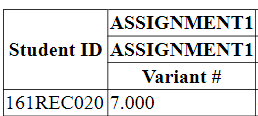
Bogdans Žukovskis

161REC020

Riga, 2021

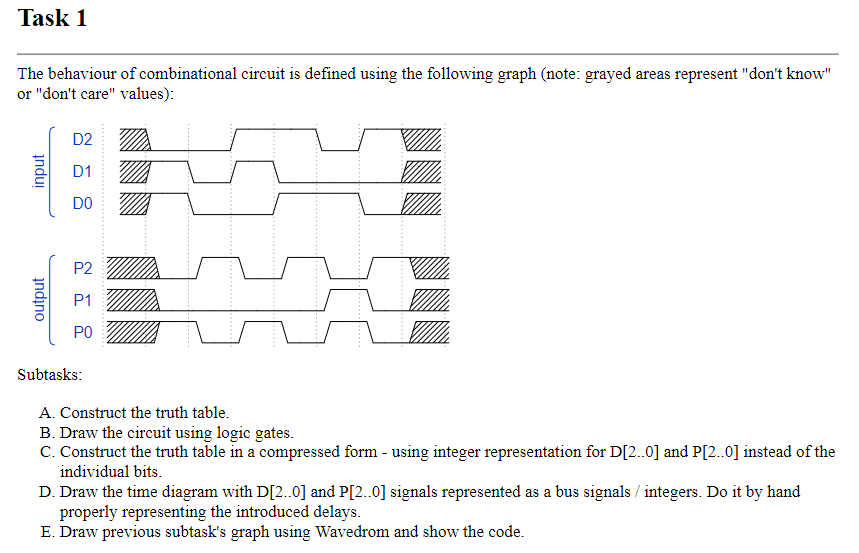
# Ievads

Variants:



1. laboratorijas darbs ir veltīts vienkāršo kombinacionālo un secīgo shēmu analīzes metodes pētīšanai gan bez programmatūras palīdzības, gan ar Wavedrom palīdzību.

# Uzdevums



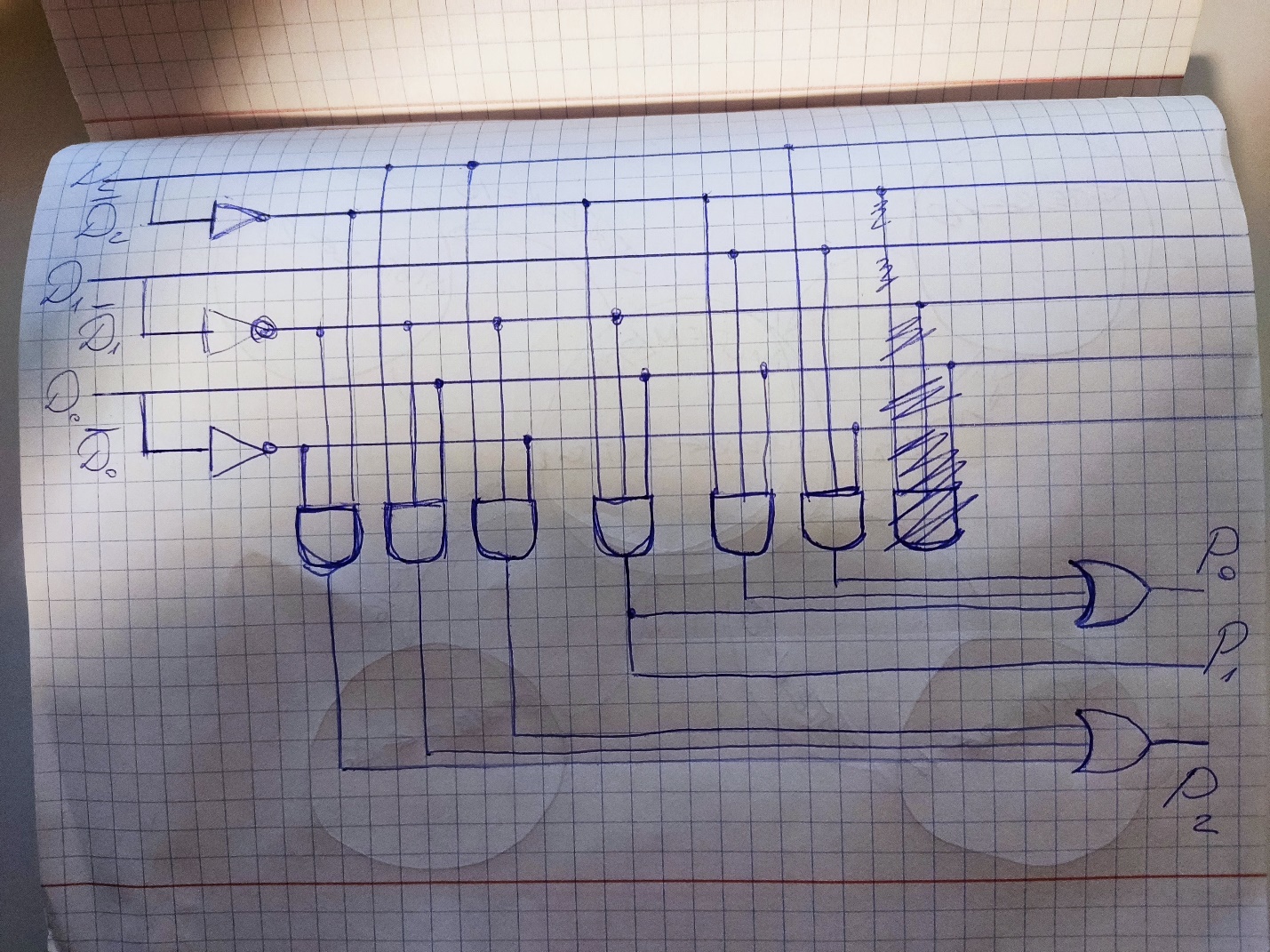
## A.

Patiesības tabula, uztaisīta pēc ieejas un izejas signāliem, izskatās sekojoši:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **INPUT** | | | **OUTPUT** | | |
| **D2** | **D1** | **D0** | **P2** | **P1** | **P0** |
| 0 | 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 |

## B.

Patiesības tabula realizējas ar loģiskajiem ventiļiem sekojoši:



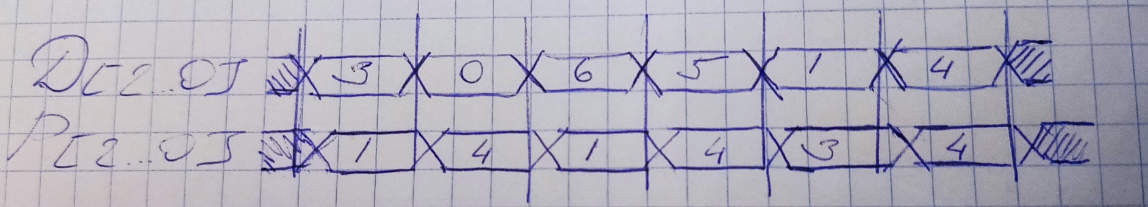
## C.

Patiesības tabulu ir iespējams vienkāršot, reprezentējot signālu kopnes vērtības integer veidā.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **INPUT** | | | **OUTPUT** | | |
|  | **D[2..0]** |  |  | **P[2..0]** |  |
| 3 | | | 1 | | |
| 0 | | | 4 | | |
| 6 | | | 1 | | |
| 5 | | | 4 | | |
| 1 | | | 3 | | |
| 4 | | | 4 | | |

## D.

Signālu kopnes ar integer vertībam attēlojas grafiski sekojošā veidā. Nosacīta propagation aizture arī ir paņemtā vērā.

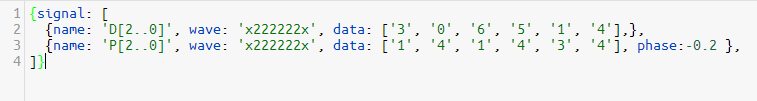


## E.

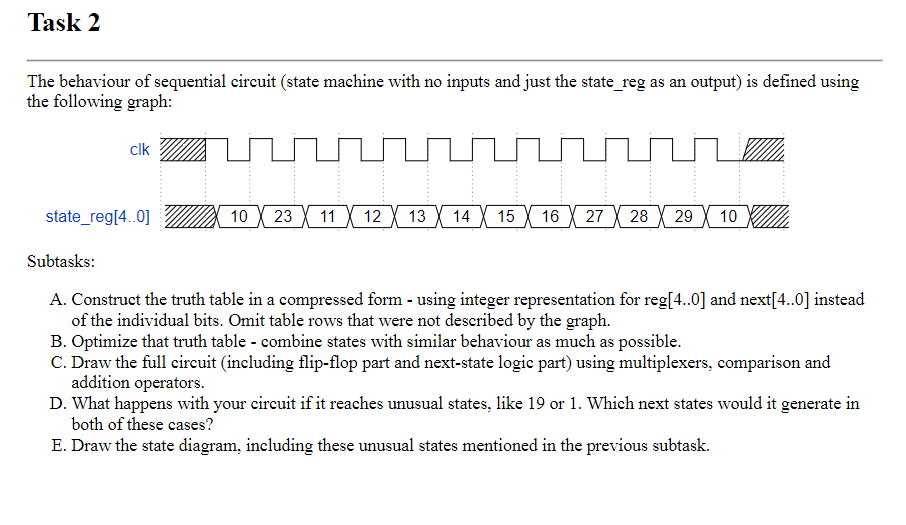
Signālus ir ērti attēlot ar *Waveform* programmatūru, kas ir paradīts šajā punktā.



JSON kods izskatās sekojoši:



# Uzdevums



## A.

Patiesības tabula, uztaisīta stāvokļa reģistram izskatās sekojoši :

|  |  |
| --- | --- |
| **state\_next** | **state\_reg** |
|  |  |
| 23 | 10 |
| 11 | 23 |
| 12 | 11 |
| 13 | 12 |
| 14 | 13 |
| 15 | 14 |
| 16 | 15 |
| 27 | 16 |
| 28 | 27 |
| 29 | 28 |
| 10 | 29 |

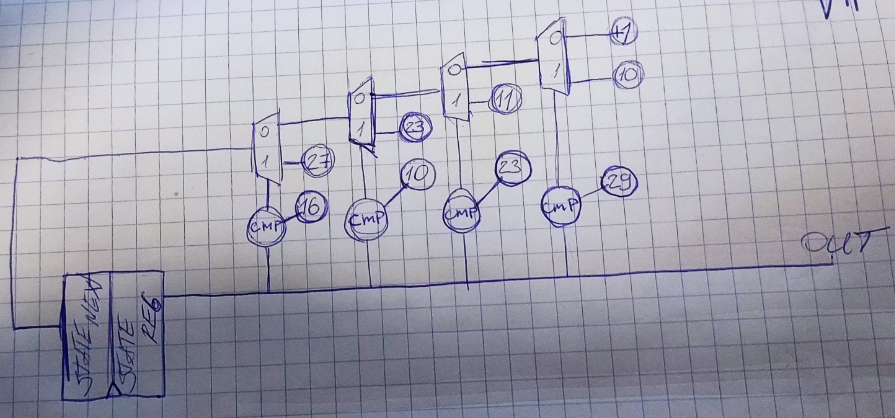
## B.

Pārēja no viena stāvkoļa uz nakamo dažos gadījumos var tikt īstenota vienāda veidā , pēc vienas veidnes, tāpēc, pēc tas idejas, dažos stāvokļus var nogrupēt ar vienādo nakamo stāvokļu sasniegšanas veidni. Ar citiem vārdiem teicot , ar vienādo nākama stāvokļa sasniegšanas vaidu. Tas ir paradīts sekojoša tabulā:

|  |  |
| --- | --- |
| **state\_next** | **state\_reg** |
|  |  |
| (++)1 | 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 27 |
| 28 |
| 23 | 10 |
| 11 | 23 |
| 27 | 16 |
| 10 | 29 |

## C.

Pēc paradītas tabulas var īstenot shēmu ar tādiem loģiskajiem elementiem kā MUX, komparators un d triggeris. D triggeru grupa - stāvokļā reģistrs. Ar mukšiem komparatoriem un konstantēm realizējas next state logic.

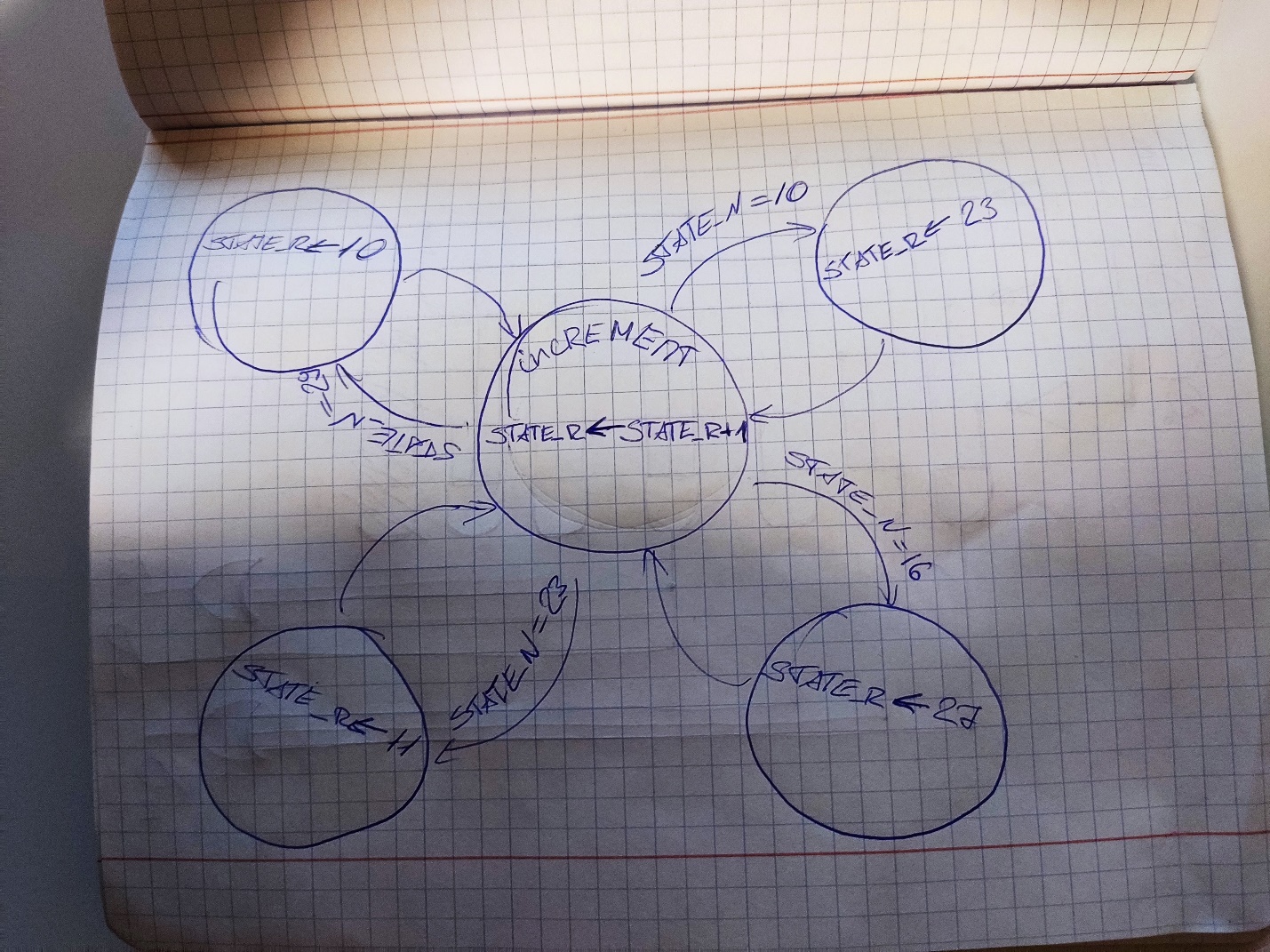


## D.

Pēc komporataru grupas ir redzāms , ka katram muksim būs izvelēts 0-ta ieeja, tātad abos nestandartos stāvokļos būs izvēlēts “+1” nakamais stāvoklis. Tāda veida, stavoklī 19, nakamā stāvokļī reģistra vertība būs 20. Ja stāvoklis ir 1 , nakamā stāvoklī reģistra vertība būs 2.

## E.

Tāda shēma var tikt attēlotā stāvoklā diagrammā. Paņemot vērā to , ka next state logic aprakstas ar ļoti primitivo reģistru operacijam – vai nu konstantes ierakstīšana, vai nu reģistra vertība inkrementēta uz 1, būs effektīvak parastas stāvokļa diagrammas veidā uzzimēt stāvokļa diagrammu ar data ceļu (data path) – stāvokļa diagramma , kuras izeja ir darbības ar reģistriem. Tas ir pamats reģistru pārnešanas metodoloģijai. (RT-methodology)



## Rezultātu analīze

Laboratorijas darbu ietvaros bija izpētītas sintežu metodes un analīžu metodes gan kombinacionalajiem gan secigajies shēmām.