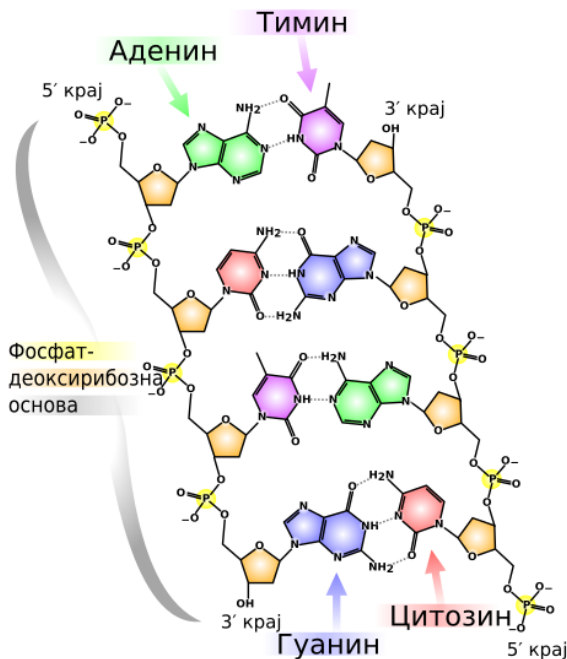


# Упутство и спецификација

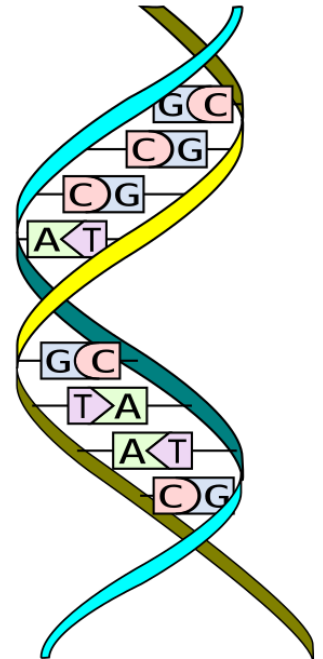
- Да би се открило присуство COVID-19 вируса као и за потребе проналажења лека за исти, потребно је урадити секвенцирање његовог дезоксирибонуклеинске киселине тј. ДНК. Секвенцу ДНК чине 4 азотне базе:

- Аденин
- Цитозин
- Гуанин
- Тимин

Означавају се скраћено са првим словом из назива: А, С, G, Т. Слика 1а приказује хемијски изглед ДНК ланца са 4 базе.



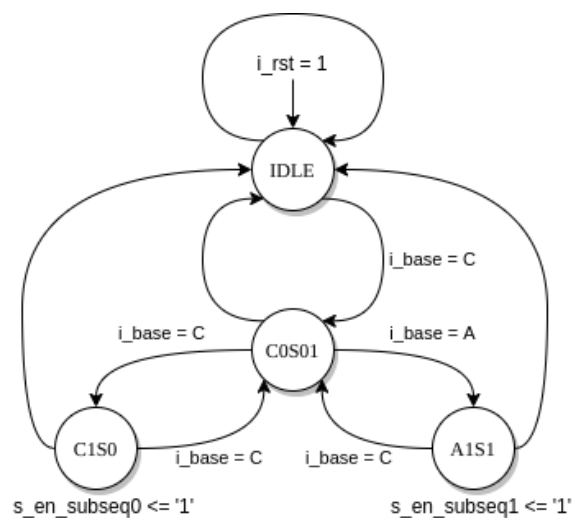
(а) Хемијска структура ДНК



(б) ДНК секвенца

Слика 1: Изглед ДНК

На Сlici 1б је пример једне ДНК секвенце. Ако пратимо ланац са плавкастом траком који почиње из горњег десног ћошка слике секвенца је GCCACATG. У овој секвенци се даље могу тражити подсеквенце. Тако на пример, ако се траже подсеквенце CC и CA, биће пронађена једна CC и две CA. Проналажење ове две подсеквенце се може извести путем аутомата. На улазе  $i\_base$  се доводи секвенца GCCACATG, по једна база у сваком такту, а стања прате подсеквенце.



Слика 2: Аутомат претраге ДНК секвенце

Слика 2 приказује аутомат за горепоменуће подсеквенце СС и СА. На почетку се креће из IDLE стања. Обе подсеквенце почињу истим стањем C0S01. Овде С означава базу, 0 иза њега означава редни број у подсеквенци, S12 означава да се исто стање користи у претраги 0. и 1. подсеквенце, односно СС и СА. После се врши грађање у два стања: C1S0 за 0. подсеквенцу СС и A1S1 за 1. подсеквенцу СА. У излазној функцији, два горепоменућа стања ће биће постављени одговарајући сигнали дозволе бројања, тако да ће бројачи подсеквенци моћи избројати појаве одговарајуће секвенце. У случају новог С одмах се покреће нова претрага, док у случају појаве неке базе ван тражених подсеквенци, претрага почиње из почетка из IDLE стања.

- Спецификација је следећа:
  1. У свим секвенцијалним процесима користити синхрони ресет.
  2. Направити аутомат са коначним бројем стања за проналажење подсеквенци:
    - Аутомат треба да детектује следеће подсеквенце.
      - (a) CGT
      - (б) CGG
      - (в) CCC
    - Сигнал стања аутомата назвати `s_state`, а следећег стања `s_next_state`.
    - Имена стања дефинисати по узор на пример дат горе.
    - Користити заједничка стања за заједничке делове подсеквенци.
    - У излазној функцији, крајњим стањима подсеквенце активирати један од сигнала дозволе бројања ( `s_en_subseq0`, `s_en_subseq1` или `s_en_subseq2` ), како би одговарајући бројач избројао подсеквенцу.
    - Излазна функција треба бити комбинациона.
  3. Направити бројач 0. подсеквенце:
    - Сигнал бројача назвати `s_cnt_subseq0`.
    - Користити `s_en_subseq0` као сигнал дозволе бројања.
    - Бројати по модулу 8.
    - Број бита је 4.
    - Стање бројача везати на `o_cnt_subseq0`.
  4. Направити бројач 1. подсеквенце:
    - Сигнал бројача назвати `s_cnt_subseq1`.
    - Користити `s_en_subseq1` као сигнал дозволе бројања.
    - Бројати по модулу 8.
    - Број бита је 4.
    - Стање бројача везати на `o_cnt_subseq1`.
  5. Направити бројач 2. подсеквенце:
    - Сигнал бројача назвати `s_cnt_subseq2`.
    - Користити `s_en_subseq2` као сигнал дозволе бројања.
    - Бројати по модулу 4.
    - Број бита је 4.
    - Стање бројача везати на `o_cnt_subseq2`.
- На основу спецификације нацртати дијаграм стања аутомата. Дијаграм назвати `FSM.png` или `FSM.jpg`, који је већ формат слике у питању, и ставити га у `LPRS1_Homework3_RA_041_2019_Solution` фолдер.
- Реализовати систем у VHDL-у по горепоменутој спецификацији и дијаграму стања аутомата. Реализацију урадити у `LPRS1_Homework3_RA_041_2019_Solution/lprs1_homework3.vhd` фајлу.
  - Кодовање база у VHDL-у је дефинисано константама.
  - Изнад кода сваке компоненте у коментару написати име описане компоненте.
- У тестбенч фајлу `LPRS1_Homework3_RA_041_2019_Solution/lprs1_homework3.vhd` верификовати систем по следећим тачкама:
  1. На улаз `i_base` довести следећу секвенце:
    - (a) TCTCCACTCTACGCCTTGCGGGATTCGTCCG
    - (б) GACACSTAAAGAATTTTCATTTACGCTGAGC
    - (в) TCTGCAGCCATTTGGTCATCGGGGTTCAAA
  2. Мењати `i_rst` и `i_base` на опадајућу ивицу такта.
  3. За сваку базу држати на улазу `i_base` у дужини једне периоде такта.

4. Пре сваке секвенце у коментару написати секвенцу.
  5. Пре сваке секвенце држати ресет активан 19 тактова.
  6. Након краја секвенце оставити сачекати 1 такт пре ресета, како би се бројач инкрементирао.
  7. На крају оставити систем у ресету.
- На крају, зиповати фолдер `LPRS1_Homework3_RA_041_2019_Solution` у зип фајл `LPRS1_Homework3_RA_041_2019_Solution.zip`. Не треба зиповати фајлове из `LPRS1_Homework3_RA_041_2019_Solution`, већ баш фолдер са фајловима. Не треба зиповати фолдер целог пројеката, него само `LPRS1_Homework3_RA_041_2019_Solution` фолдер. Игнорисати сва друга упуства, ово је најважније. Горепоменути зип фајл послати као решење свом асистенту преко чета у MS Teams-у.