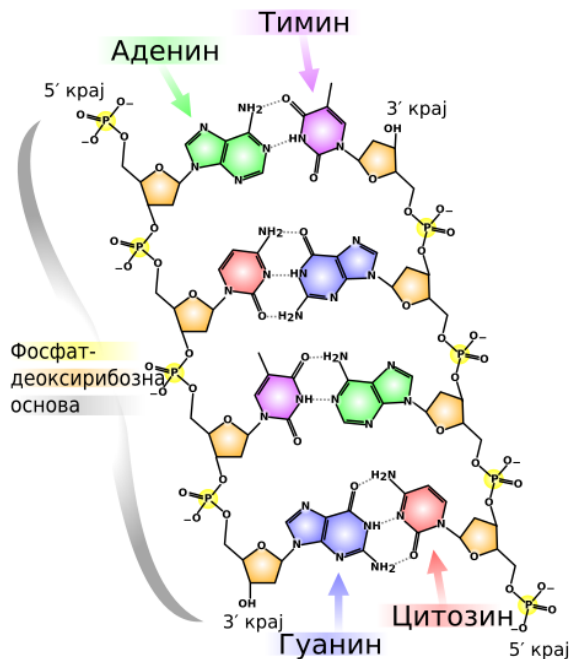


Домаћи задатак 3

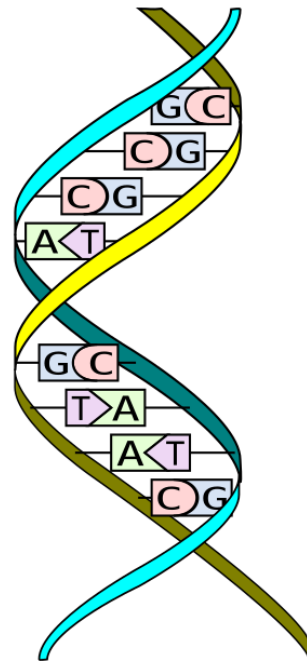
- Да би се открило присуство COVID-19 вируса као и за потребе проналажења лека за исти, потребно је урадити секвенцирање његовог дезоксирибонуклеинске киселине тј. ДНК. Секвенцу ДНК чине 4 азотне базе:

- 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.



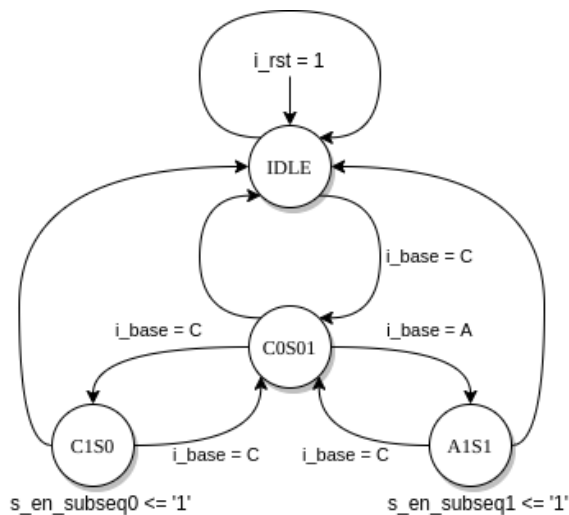
(а) Хемијска структура ДНК



(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 `COS01`. Овде `C` 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 `C` 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. Направити аутомат са коначним бројем стања за проналажење подсеквенци:
 - Аутомат треба да детектује следеће подсеквенце:
 - (a) `CAA`
 - (b) `CAC`
 - (v) `CGA`
 - Сигнал стања аутомата назвати `s_state`, а следећег стања `s_next_state`.
 - Као улаз у аутомат користити `s_base`.
 - 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 и прелазна функција треба бити комбинациона. Никако не реализовати прелазну функцију у секвенцијалном процесу.
 - Користити синхрони ресет.
2. 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`.
 - Користити синхрони ресет.
3. Направити бројач `1.` подсеквенце:
 - Сигнал бројача назвати `s_cnt_subseq1`.
 - Користити `s_en_subseq1` као сигнал дозволе бројања.
 - Бројати по модулу `7`.
 - Број бита је `4`.
 - Користити синхрони ресет.
4. Направити бројач `2.` подсеквенце:
 - Сигнал бројача назвати `s_cnt_subseq2`.
 - Користити `s_en_subseq2` као сигнал дозволе бројања.
 - Бројати по модулу `5`.
 - Број бита је `4`.
 - Користити асинхрони ресет.
5. Направити излазни мултиплексер за селекцију излаза са бројача:
 - Реализовати мултиплексер преко `when-else`.
 - Користити `i_cnt_subseq_sel` сигнал као селекциони.
 - Користити `o_cnt_subseq` као излаз из мултиплексера.
 - Улази ће наравно бити `s_cnt_subseq0...`
6. Направити улазни мултиплексер за селекцију базе која ће ући у аутомат:
 - Реализовати мултиплексер преко `when-else`.
 - Користити `i_base_src_sel` сигнал као селекциони.
 - * Ако је селекциони улаз `'0'` селекувати `i_base`.
 - * Ако је селекциони улаз `'1'` селекувати `s_sh_base`.
 - Користити `s_base` као излаз из мултиплексера.
7. Направити померачки регистар који ће једну по једну базу из секвенце послати на аутомат:
 - Реализовати комбинациону мрежу померачког регистара преко `process-case`.
 - Користити `s_sh_reg` као сигнал регистара.

- * Када је `i_load_sequence` на 1 уписати `i_sequence` у регистар.
 - * У противном померати регистар удесно.
 - За излаз `s_sh_base` узимати из `s_sh_reg` 2 бита најмање вредности.
 - Користити асинхрони ресет.
- 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_PR_158_2020_Solution` фолдер.
 - Реализовати систем у VHDL-у по горепоменутој спецификацији и дијаграму стања аутомата. Реализацију урадити у `LPRS1_Homework3_PR_158_2020_Solution/lprs1_homework3.vhd` фајлу.
 - Кодовање база у VHDL-у је дефинисано константама.
 - Изнад кода сваке компоненте у коментару написати име описане компоненте.
 - Проверити исправност решења путем симулације у тестбенч фајлу `LPRS1_Homework3_PR_158_2020_Solution/lprs1_homework3_tb.vhd`. Следи пример једног тестбенча:
 1. На улаз `i_base` довести следеће секвенце:
 - (а) `CGGT TACG ACGG GATC CCCG TCCT GTCT CGCT`
 - (б) `GTCA CCGT GTAA GATT GGTA GTAC ATGC CTAG`
 2. Ресетовати систем и након 3.25 тактова отпустити ресет.
 3. 0. секвенцу слати базу по базу на `i_base` улаз.
 - (а) За сваку базу држати на улазу `i_base` у дужини једне периоде такта.
 - (б) Не заборавити селектовати одговарајућу извор базе са `i_base_src_sel`.
 4. 1. секвенцу слати базу по базу на `i_base` улаз.
 - (а) Поставити секвенцу на `i_sequence` и учитати је активацијом `i_load_sequence`.
 - (б) Гледајући у интерне сигнале одредити колико је периода такта неопходно потребно да би се свака база секвенце послала на аутомат.
 - (в) Не заборавити селектовати одговарајућу извор базе са `i_base_src_sel`.
 5. Пре сваке секвенце у коментару написати секвенцу.
 6. Пре сваке секвенце држати ресет активан 14 тактова.
 7. Након краја секвенце оставити сачекати 1 такт, како би се бројач инкрементирао.
 8. Ишчитати вредности бројача излазног мултиплексера на `o_cnt_subseq` излазу, по једна вредност `i_cnt_subseq_sel` по периоди такта.
 9. На крају оставити систем у ресету.
 - На крају, зиповати фолдер `LPRS1_Homework3_PR_158_2020_Solution` у зип фајл `LPRS1_Homework3_PR_158_2020_Solution.zip`. Не треба зиповати фајлове из `LPRS1_Homework3_PR_158_2020_Solution`, већ баш фолдер са фајловима. Не треба зиповати фолдер целог пројекта, него само `LPRS1_Homework3_PR_158_2020_Solution` фолдер. Ако се пошаље нешто друго, рецимо цео пројекат, рад неће бити гледан и резултоваће са оценом **D** односно **0** бодова.
 - Решење (.зип датотеку) поставити на СОВУ у делу ЗАДАТАК 3, у своју додељену групу.