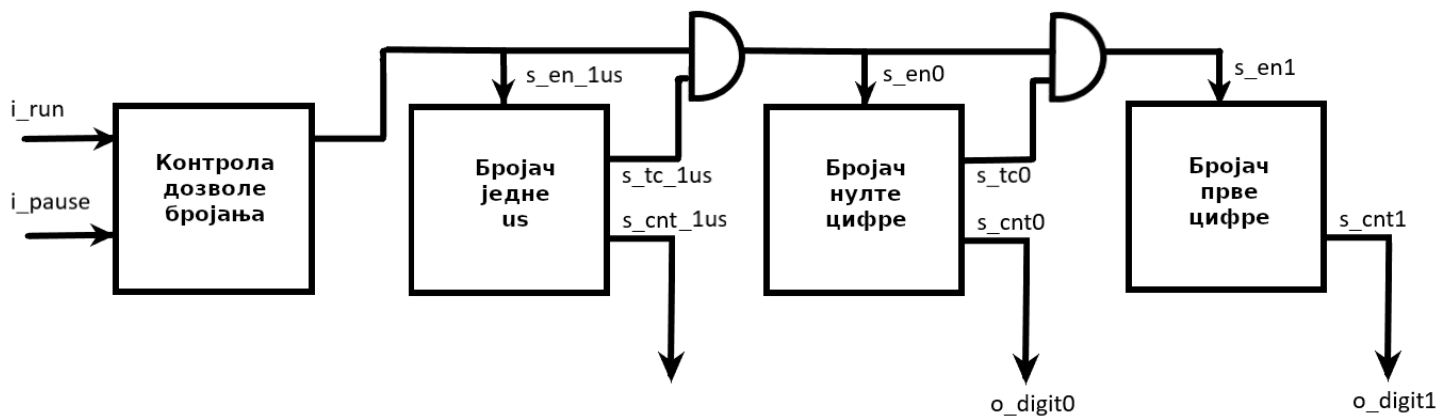


## Домаћи задатак 2

- Потребно је реализовати штоперицу. На Слици 1 је блок шема штоперице.



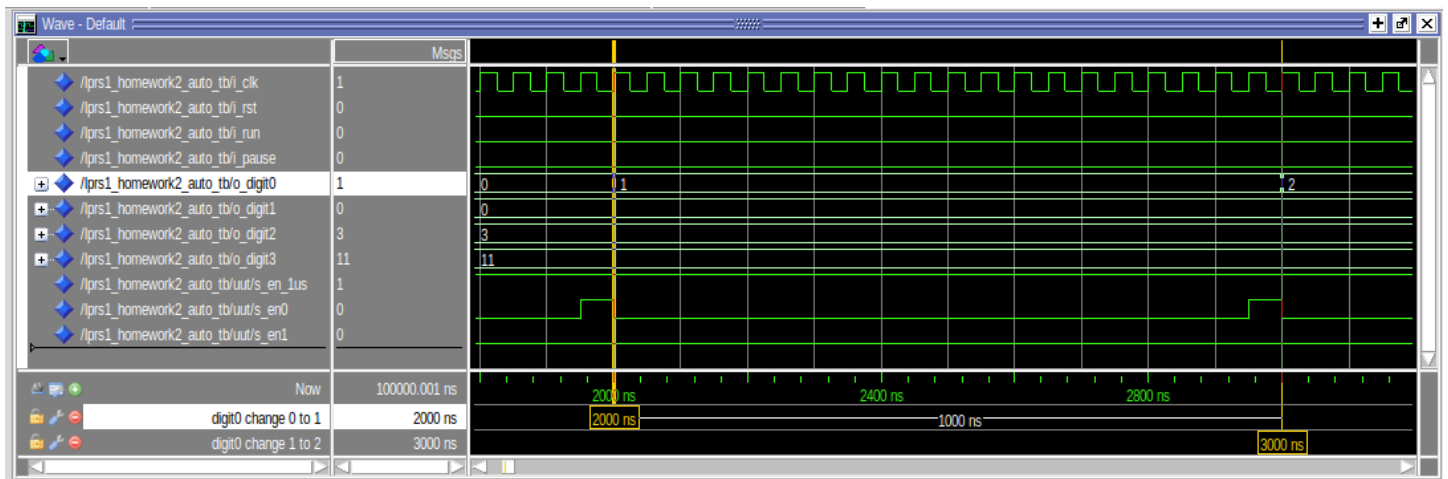
Слика 1: Блок шема штоперице

Спецификација је следећа:

- У свим секвенцијалним процесима користити асинхрони ресет.
- Реализовати контролу дозволе бројања штоперице на сигналу `s_en_1us` тако да улазни тастери контролишу дозволу бројања на следећи начин:
  - При притиснутом ресету (док је `i_rst` на 1) штоперица је заустављена (без озбира на друге улазе).
  - Приликом притиска `i_run` тастер (сигнал је на 1) штоперица се покреће. Штоперица наставља са радом и кад је `i_run` тастер пуштен (логичка 0).
  - Приликом притиска `i_pause` тастер штоперица се зауставља. Ако је истовремено притиснут `i_run` тастер штоперица не треба да се заустави.
- Направити бројач једне микросекунде (1 us):
  - Сигнал бројача назвати `s_cnt_1us`.
  - Одредити одговарајући модуо бројања на основу периоде такта дате у тестбенчу.
  - Одредити број бита који је довољан за тај модуо (не одредити превише бита).
  - Овај бројач треба да користи сигнал дозволе бројања `s_en_1us` из контроле дозволе бројања штоперице.
  - Сигнал краја бројања `s_tc_1us` треба реализовати комбинационо.
  - Овај бројач треба да генерише дозволу бројања `s_en0` за наредни бројач на основу `s_en_1us` и `s_tc_1us`.
- Направити бројач једне нулте цифре:
  - Сигнал бројача назвати `s_cnt0`.
  - Користити `s_en0` као сигнал дозволе бројања.
  - Бројати по модулу 10.
  - Број бита је 4.
  - Овај бројач треба да користи сигнал дозволе бројања `s_en0` од претходног бројача.
  - Сигнал краја бројања `s_tc0` треба реализовати комбинационо.
  - Овај бројач треба да генерише дозволу бројања `s_en1` за наредни бројач на основу `s_en0` и `s_tc0`.
  - Сигнал `s_cnt0` везати на `o_digit0`.
- Направити бројач једне прве цифре:
  - Сигнал бројача назвати `s_cnt1`.
  - Користити `s_en1` као сигнал дозволе бројања.
  - Бројати по модулу 3.
  - Број бита је 4.
  - Овај бројач треба да користи сигнал дозволе бројања `s_en1` од претходног бројача.
  - Сигнал краја бројања `s_tc1` треба реализовати комбинационо.
  - Сигнал `s_cnt1` везати на `o_digit1`.
- На сигнал `o_digit2` доделити 4.
- На сигнал `o_digit3` доделити 11.

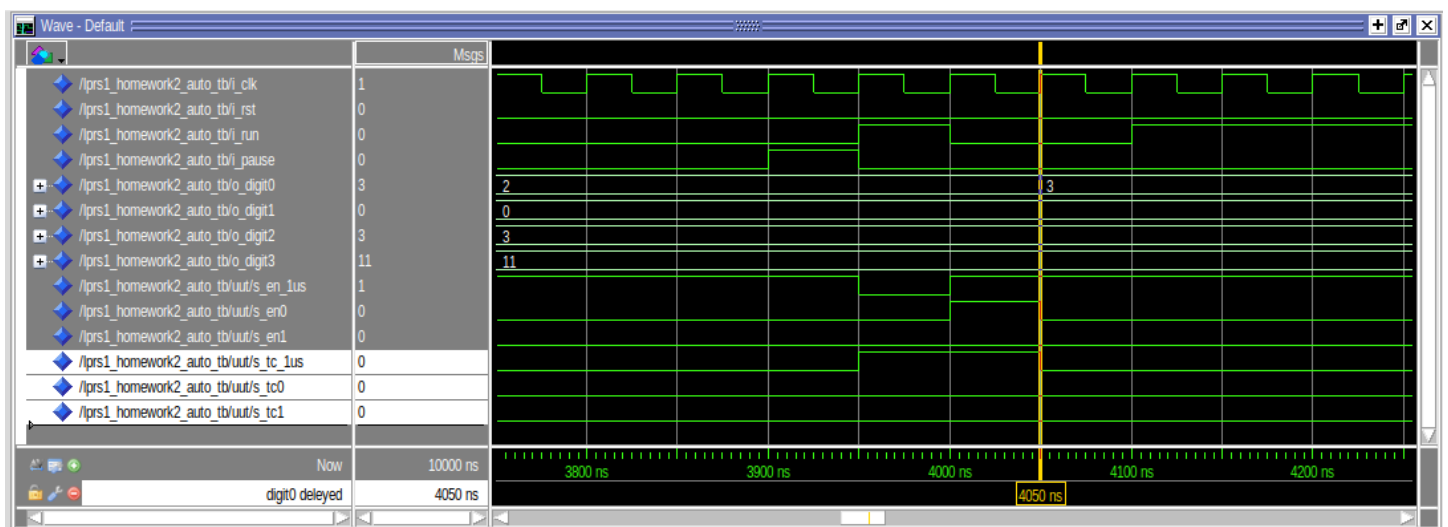
НАПОМЕНА: Временско мултиплексирање за приказ бројева на 7-сегментним дисплејима је реализовано у датотеци `lprs1_homework2_for_board.vhd`.

- Реализовати систем у VHDL-у по горепоменутој спецификацији. Реализацију урадити у `LPRS1_Homework2_PR_158_2020_Solution/lprs1_homework2.vhd` фајлу.
  - Изнад кода сваке компоненте у коментару написати име описане компоненте.
- Проверити исправност решења путем симулације у тестбенч фајлу `LPRS1_Homework2_PR_158_2020_Solution/lprs1_homework2.vhd`. Следи пример једног тестбенча:
  - Држати ресет притиснут (`i_rst` на логичкој 1) за 1 микросекунду минус половину периоде такта, па га пустити, па сачекати још половину периоде такта.
  - Покренути штоперицу притиском на `i_run` тастер. Добро урађени овај и претходни корак проузроковаће да сигнал `o_digit0` пређе са 0 на 1 у 2. микросекунди.
  - Зумирати таласиће (Waveform) тако да се види када `o_digit0` прелази са 0 на 1 (у 2. микросекунди) и са друге када `o_digit0` прелази са 1 на 2 (у 3. микросекунди) (видети Сliku 2).



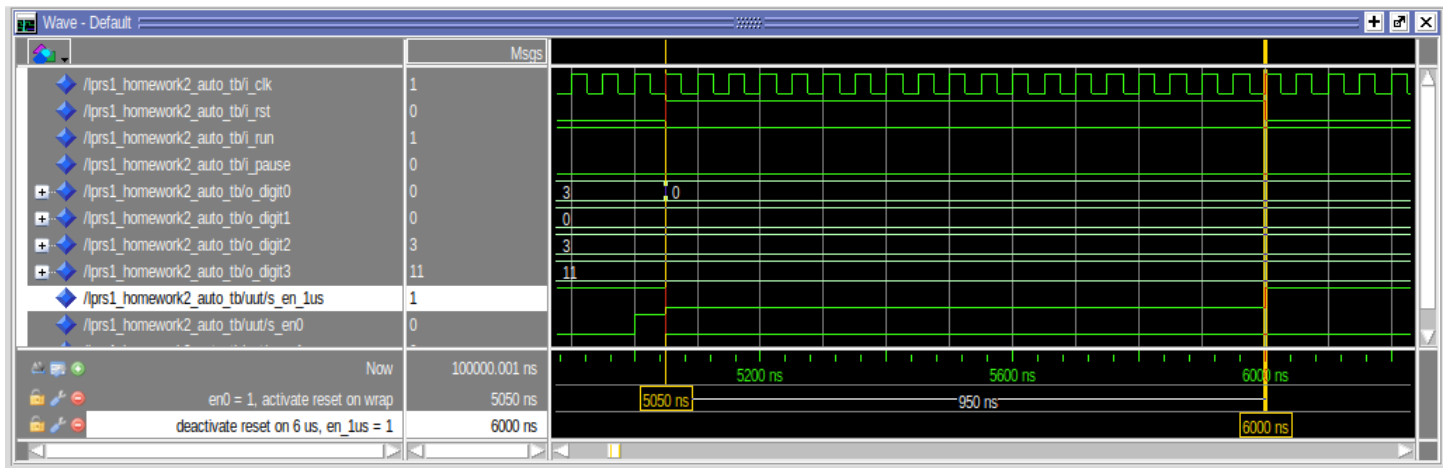
Слика 2: Тест бројача 1 микросекунде

- Паузирали штоперицу на један такт (притиснути `i_pause` и после једног такта пустити па притиснути `i_run` и после једног такта пустити) тако да промена `o_digit0` са 2 на 3 не буде уместо у 4. микросекунди него у 4. микросекунда плус 1 такт. Утврдити ту позицију у времену рачунањем, експериментално, или се користити сигналом краја бројања (нпр. `s_en0`). Видети Сliku 3 за више детаља.



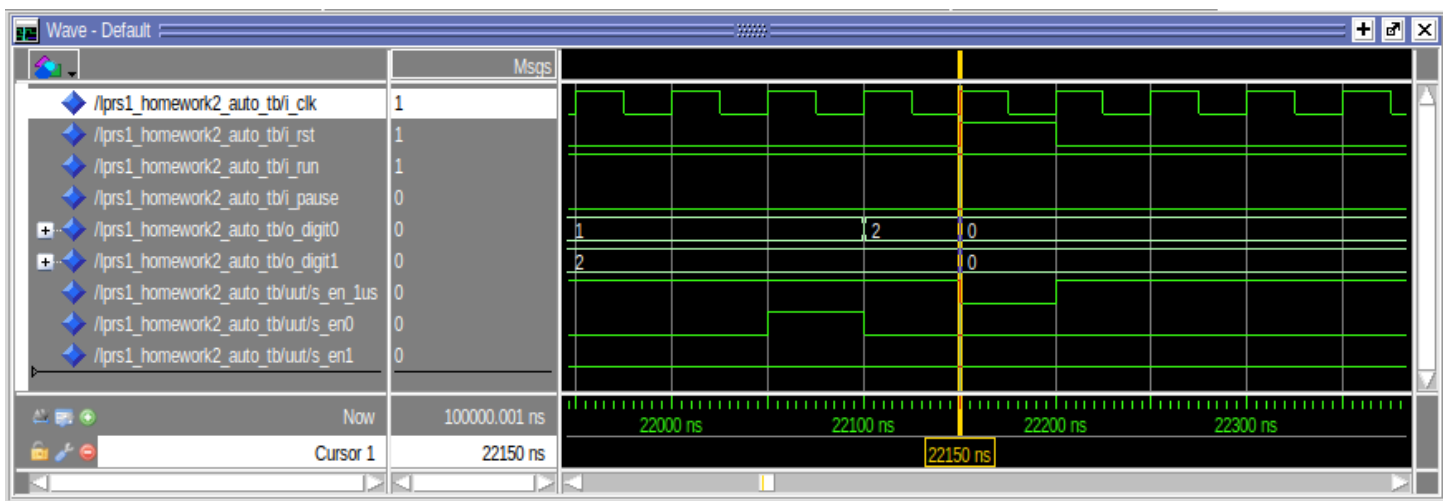
Слика 3: Кашњење промене путем паузе

- Поставити `i_run` на 1.
- На промени `o_digit0` са 3 на 4, рестартовати штоперицу (притом неће ни доћи до промене на 4 већ на 0). Пустити ресет на 6. микросекунди. Добро урађени овај и претходни кораци проузроковаће да сигнал `s_en_1us` постане 1 тачно у 6. микросекунди. Видети Сliku 4 за више детаља.



Слика 4: Изглед ресета између 5. и 6. микросекунде

7. Пустити симулацију да траје све док `o_digit0` не достигне 3 и `o_digit1` не достигне 2 и задрже се на тим вредностима у трајању од једног такта (видети Сliku 5). Онда држати притиснут ресет на 1 такт.



Слика 5: Пример рестартовања кад штоперица достигне одређену вредност

8. Слично као и у претходној тачки, пустити симулацију да траје све док `o_digit0` не достигне 5 и `o_digit1` не достигне 1 и задрже се на тим вредностима у трајању од једног такта. Опет држати притиснут ресет на 1 такт.

- На крају, зиповати фолдер `LPRS1_Homework2_PR_158_2020_Solution` у зип фајл `LPRS1_Homework2_PR_158_2020_Solution.zip`. Не треба зиповати фајлове из `LPRS1_Homework2_PR_158_2020_Solution`, већ баш фолдер са фајловима. Не треба зиповати фолдер целог пројеката, него само `LPRS1_Homework2_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 2, у своју додељену групу.
- На термину вежбе проширити једне микросекунде (1 us) да броји 1 секунду и проверити решење на плочи. Следеће су контроле на плочици:
  1. `i_rst` - Ресет тастер (горњи десно)
  2. `i_run` - Леви тастер
  3. `i_pause` - Централни тастер