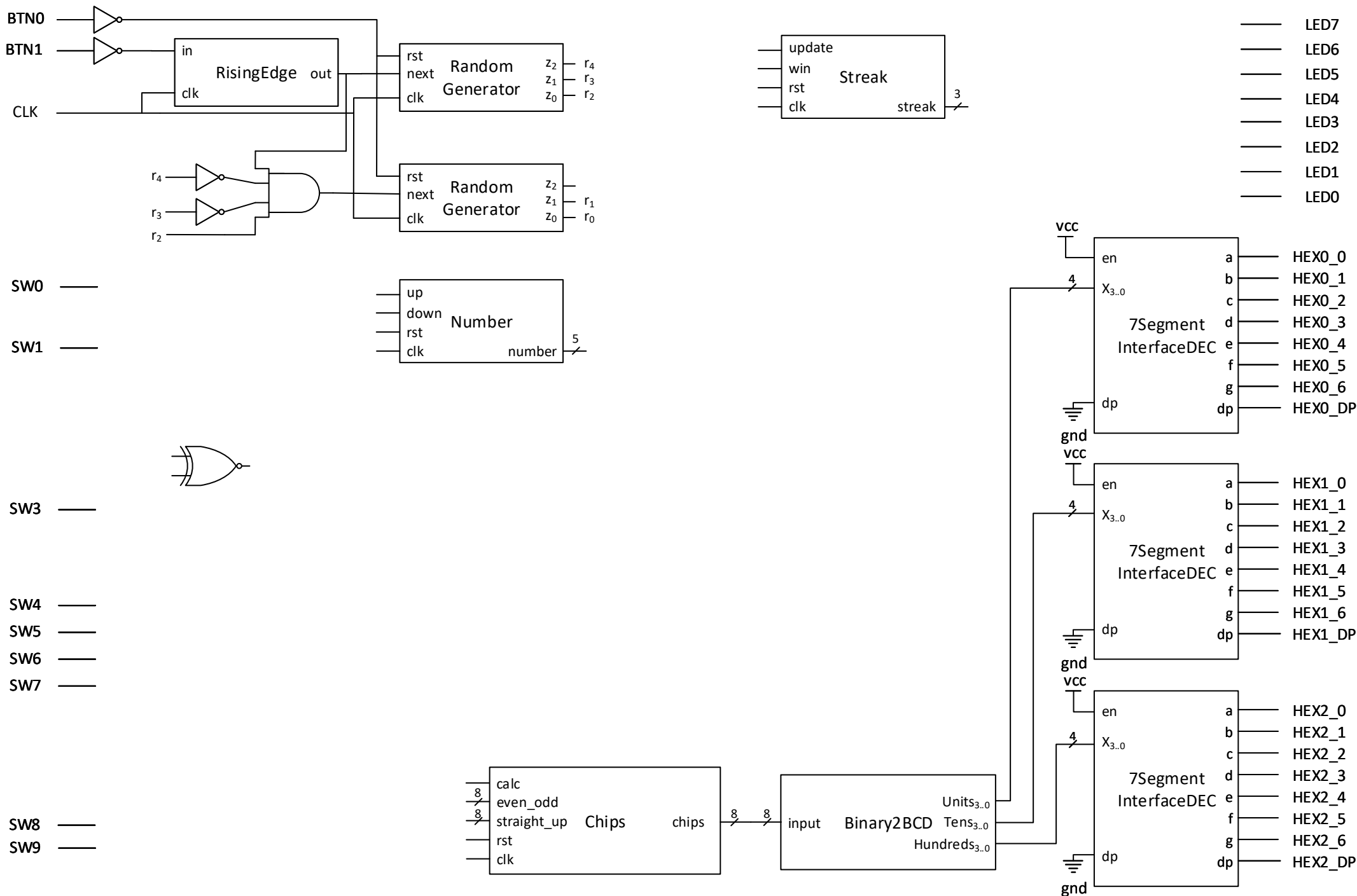


## Практикум из основа рачунарске технике – домаћи задатак 2019/2020

- четврта лабораторијска вежба -

### Задатак

Потребно је допунити дату шему која симулира један део рада рулет машине са цилиндром од 32 броја **0<sub>10</sub>-31<sub>10</sub>** (неки делови шеме реализују исте функционалности са треће лабораторијске вежбе, док се неке функционалности разликују). На основу улога играча, броја који је пао, као и одабира опција (заједно са предефинисаним фактором добитка) на које је играч играо, потребно је да се на седмосегментним дисплејима прикаже број освојених жетона (који може да буде и нула) у текућем бацању лоптице. На седмосегментним дисплејима **HEX2**, **HEX1** и **HEX0** приказује се број освојених жетона играча у декадном бројном систему тако да дисплеју **HEX2** одговара цифра стотина, дисплеју **HEX1** одговара цифра десетица, а дисплеју **HEX0** одговара цифра јединица. Прекидачима **SW4-SW7** играч бира један од могућих предефинисаних улога и то **1<sub>10</sub>**, **2<sub>10</sub>**, **4<sub>10</sub>** или **6<sub>10</sub>** жетона, респективно. Да би улагање било исправно, максимално један прекидач може да буде у позицији један (чиме се бира одређен улог), док остали прекидачи морају да буду у позицији 0. У случају да су сви прекидачи у позицији 0, онда се сматра да играч прескаче спин (бацање лоптице), па је улог једнак нули. На рулету постоје две опције које играч може да игра. Прва опција је да играч погоди број који ће да падне у наредном спину, док је друга опција да играч погоди да ли ће број у наредном спину бити паран или непаран. У случају да је играч играо погађање броја и том приликом погоди број који падне, његов добитак представља тридесетдвоструки улог (улог \* 32), у случају да је промашио добитак је једнак нули. Играч бира број на основу стања прекидача **SW0** и **SW1**. Прелазак прекидача **SW0** из позиције 0 у позицију 1 увећава жељени број за један, док прелазак прекидача **SW1** из позиције 0 у позицију 1 смањује жељени број за један. Одабрани број се приказује у бинарном систему на **LED4-LED0** (**LED4** представља бит највеће тежине). Описану функционалност бирања жељеног броја реализује модул **Number**. У случају да је играч играо погађање парност броја и том приликом погоди парност броја који падне, његов добитак представља дупли улог, у случају да је промашио добитак је једнак нули. Играч бира парност на основу стања прекидача **SW3** (позиција 0 одговара предвиђању парног броја, а позиција 1 одговара предвиђању непарног броја). Играч може да игра или прву, или другу или обе опције у зависности од стања прекидача **SW8** и **SW9**. Прекидач **SW8** у позицији један означава да играч игра опцију погађања броја, а прекидач **SW9** у позицији један означава да игра опцију погађања парности броја. Ако су оба прекидача у позицији један играч игра обе опције при чему је укупан добитак збир оба добитка (улог је исти за обе опције). Функционалност рачунања броја освојених жетона реализује модул **Chips**. Играч покреће спин на сваки притисак дугмета **BTN1** чиме рулет машина генерише насумичан број на основу предефинисане секвенце из опсега **0<sub>10</sub>-31<sub>10</sub>** (искористити модул са претходне лабораторијске вежбе). Број узастопних добитака играча приказује се на **LED7-LED5** (приказује највише 3 узастопна добитка). Као добитак се рачуна сваки спин у којем је погођена бар једна игра која је играна. Разматрају се само спинови у којима је одиграна бар једна игра и у којима је улог већи од 0. Ову функционалност реализује модул **Streak**. Рулет машина се ресетује притиском дугмета **BTN0** чиме се ресетујују све секвенцијалне мреже на одговарајућу вредност. Сматрати да играч прво ресетује рулет машину пре почетка њеног коришћења. На датој шеми су модули **RisingEdge**, **RandomGenerator**, **Binary2BCD** и **7SegmentInterfaceDEC** са којима се студент упознао на претходним лабораторијским вежбама. При реализацији главне шеме дозвољено је коришћење свих стандардних комбинационих мрежа, флип-флопова, И, ИЛИ и НЕ кола и **RisingEdge** компоненте. При реализацији модула **Number**, **Streak** и **Chips** користити искључиво регистре са стандардним функционалностима (паралелни/серијски упис, инкрементирање, декрементирање, брисање). Обратите пажњу да су вредности регистара ажурају у адекватним тренутцима тј. да се у њих уписују добре вредности. Није потребно обраћати пажњу на могућа прекорачења вредности који се добијају као последица сабирања. Дозвољено је именовати излазе логичких елемената и користити их на другим деловима структурне шеме. Дозвољено је спајати или издвајати сигнале и тако формирати нове групе сигнала. Студент је дужан да црта прегледно коришћењем графитне оловке. Урађен домаћи задатак на формулару донети у термину четврте лабораторијске вежбе.



**Задатак**

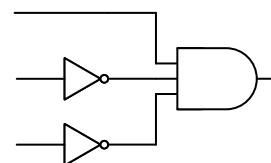
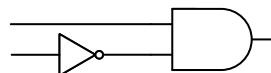
Потребно је допунити дату шему за реализацију модула **Number**. Овај модул представља секвенцијалну мрежу Муровог типа која има улазне сигнале **up**, **down**, **rst** и **clk** (сигнал такта) и излазни петобитни сигнал **number**. Активним улазним сигнаlima **up** и **down** могуће је мењати вредност жељеног броја тако што се вредност увећа или смањи вредност за један, респективно. Ако оба сигнала у истом тренутку имају активну вредност број ће се увећати за један (сигнал **up** је вишег приоритета од сигнала **down**). Излаз мреже **number** представља изабрани жељени број. Улазни сигнал **rst** служи за ресетовање мреже, чиме се активном вредношћу мрежа доводи у подразумевано стање (сви регистри се поставе на вредност нула) неvezано за улазне сигнале **up** и **down** (**rst** је вишег приоритета). При реализацији **користити искључиво регистре са стандардним функционалностима (паралелни/серијски упис, инкрементирање, декрементирање, брисање)**.

up —

down —

rst —

clk —



—5— number

**Задатак**

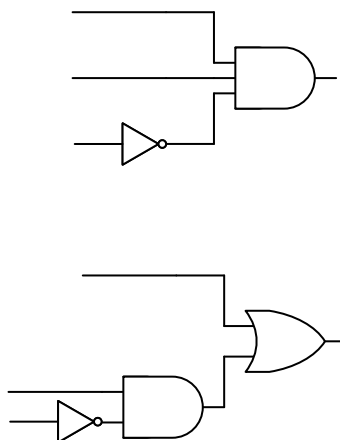
Потребно је допунити дату шему за реализацију модула **Streak**. Овај модул представља секвенцијалну мрежу Муровог типа која има улазне сигнале **update**, **win**, **rst** и **clk** (сигнал такта) и излазни тробитни сигнал **streak**. Мрежа бележи узастопне добитке. При сваком новом добитку по један бит излазног сигнала **streak** добија вредност један почевши од бита најмање тежине (нпр. ако су се десила два добитка за редом најнижа два бита излазног сигнала **streak** ће имати вредност један). Може се приказати највише 3 узатопна добитка. Активан улазни сигнал **update** означава да треба ажурирати информацију о узастопним добицима, а сигнал **win** садржи информацију да ли се ради о добитку или не. Активан улазни сигнал **win** означава да је у питању добитак, док неактиван улазни сигнал **win** означава да је дошло до губитка и да је завршена серија добитака. У случају прекида серије добитака треба поставити све битове излазног сигнала **streak** на вредност нула. Улазни сигнал **rst** служи за ресетовање мреже, чиме се активном вредношћу мрежа доводи у подразумевано стање (сви регистри се поставе на вредност нула) неvezано за остале улазне сигнале (**rst** је вишег приоритета). При реализацији користити искључиво регистре са стандардним функционалностима (паралелни/серијски упис, инкрементирање, декрементирање, брисање).

update —

win —

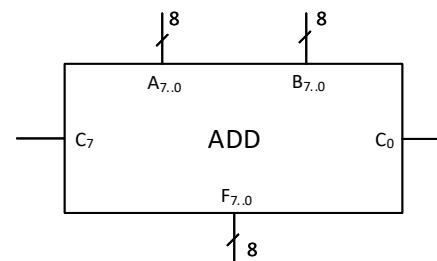
rst —

clk —

—<sup>3</sup>— streak

**Задатак**

Потребно је допунити дату шему за реализацију модула ***Chips***. Овај модул представља секвенцијалну мрежу Муровог типа која има улазне сигнале ***calc***, ***even\_odd***, ***straight\_up***, ***rst*** и ***clk*** (сигнал такта) и излазни осмобитни сигнал ***chips***. Овај модул има могућност израчунавања броја освојених жетона играча након бацања лоптице. Ажурирање се задаје активном вредношћу сигнала ***calc***. Добитак играча при погађању парности и добитак играча при погађању вредности броја се задаја улазним осмобитним сигнаlima ***even\_odd*** и ***straight\_up***, респективно (ради се о оствареним, а не о потенцицијалним добицима). Излаз мреже ***chips*** представља тренутно стање играча. Улазни сигнал ***rst*** служи за ресетовање мреже, чиме се активном вредношћу мрежа доводи у подразумевано стање (сви регистри се поставе на вредност нула) неvezано за остале улазне сигнале (***rst*** је вишег приоритета). При реализацији **користити искључиво регистре са стандардним функционалностима (паралелни/серијски упис, инкрементирање, декрементирање, брисање).**



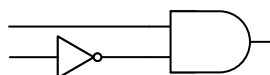
calc ———

even\_odd 8

straight\_up 8

rst ———

clk ———



8 chips