

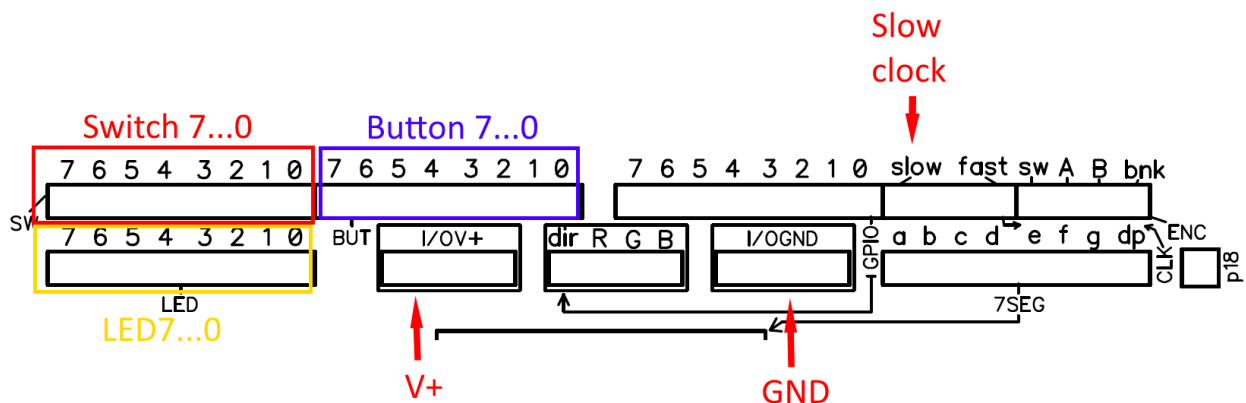
Labratöiden toimintaselostukset

Tässä dokumentissa on digitaalitekniikan alkeiskurssin syksyn 2015/tammikuun 2016 labratöissä rakennettujen kytkentöjen toimintaselostukset ja kytkentäkaaviot. Lopussa on vielä muutamia kurssilla läpikäytyjä, mutta ei rakennettuja Larson skannereita, joihin voi suhtautua lähinnä oheismateriaalina.

Testiympäristöstä

Kytkennot rakennetaan leipälaudalle. Lähes kaikkien kytkentöjen apuna on tarkoitettu käytettäväksi Hacklabin digitaalitekniikan protoboard-korttia ("punainen levy") jossa on tarpeelliset painikkeet, ledit ym. Kytkentäkaavioissa on merkattu liitäntäpiste niihin kohtiin, joista vedetään johto protokortille.

Myös sähkön syöttö leipälaudalle kannattaa ottaa protokortin I/OV+ ja I/OGND -pisteistä. Tällöin riittää, että käyttö sähköt labrapowerista tai USB-laturista tuodaan vain protokortille.



Protokortin säädöt

Kaikissa kytkennöissä toimitaan 5V logiikalla, joten protokortin jännitteenvaihtajumpperi pitää olla asennossa 5V.

Protokortilla on kaksi CPLD-piiriä, joissa pitää näitä kytkentöjä käytettäessä olla tällaiset sisällöt:

CPLD1: debouncer, joka poistaa värähtelyt kaikista kytkimistä ja painikkeista ja kytkee ne yläreunan rimoille.

<https://github.com/JKN0/deep-protoboard-debouncer>

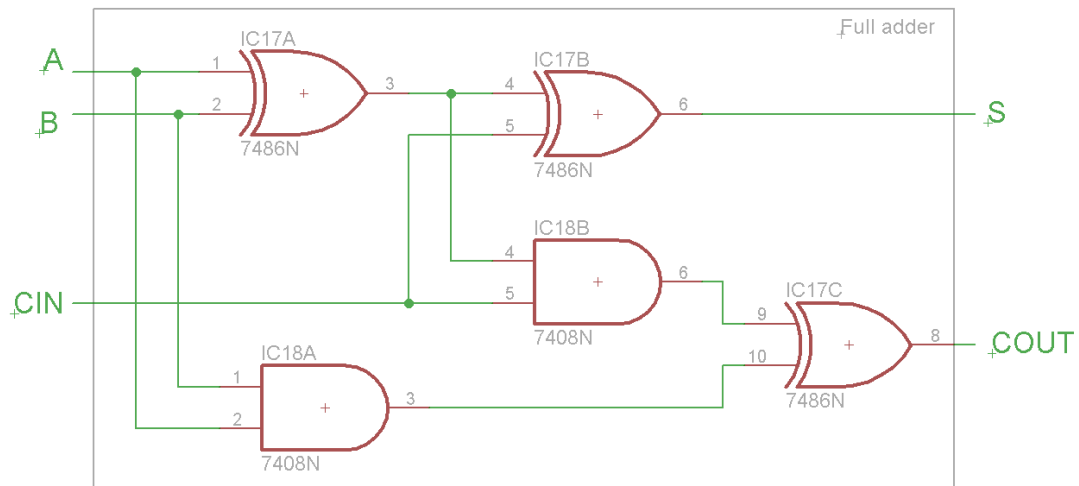
CPLD2: 7-segment-dekooderi, joka dekoodaa ledeille tuodun 8-bittisen binäärisanan kahdeksi heksamerkiksi kortin 7-segment-näytölle.

<https://github.com/JKN0/deep-protoboard-leds7seg>

Piiriperheistä

KytKentäkaavioihin on merkitty käytetty piirityyppi (esim. 7414, 74LS00) kiinnittämättä paljoakaan huomiota piiriperheseen. Tämä ei siis tarkoita, että piirit pitäisi olla juuri merkittyä perhettä. Useimmissa tapauksissa kytkennät toimivat minkä tahansa perheen piireillä. Nykymaailmassa 74HC-perhe on suositeltavin.

Yhteenlaskuyksikkö



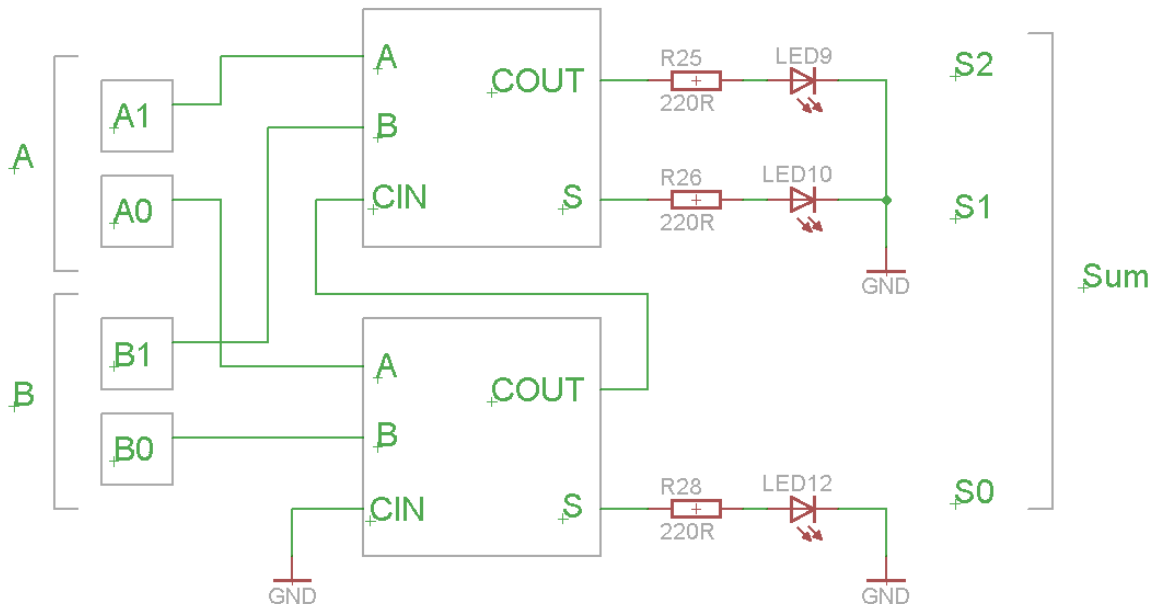
Tämä kytkentä on full adder, eli logiikka, joka pystyy laskemaan kahden bitin (kahden yksibittisen binääriluvun) summan, carry-bitit huomioiden. Yhteenlaskettavat bitit ovat A ja B, summa on S ja mahdollinen lähtevä carry (muistinumero) on COUT. Tarvittaessa edellisen asteen carry tuodaan sisään CIN:iin.

Full adder-logiikan totuustaulu on:

Sisään			Ulos	
A	B	CIN	COUT	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Tämä kytkentä poikkeaa hiukan kirjallisuudessa normaalisti esitetystä. COUT:in tuottava portti (IC17C) on normaalisti OR-portti, mutta tässä siihen on vaihdettu XOR-portti, jolloin säästetään yksi mikropiiri. Yleensä OR-porttia ei voi noin vain korvata XOR-portilla, mutta tässä nimenomaisessa tapauksessa voi. Perustelujen miettiminen tälle toimii sopivana harjoituksena, jos ne on unohtuneet.

Kahden 2-bittisen binääriluvun yhteenlaskimeen tarvitaan kaksi full adderia, jotka yhdistetään kuten alla on kuvattu. Laatikot A0,A1 ja B0,B1 ovat yhteenlaskettavat (2-bittisiä binäärilukuja), ne voivat tulla esim. protokortin kytkimistä. Kytkinten avulla voidaan syöttää yhteenlaskettavat sisään. Lähtöihin on piirretty erilliset ledit, mutta yhtä hyvin ne voitaisiin viedä protokortin ledituloihin.



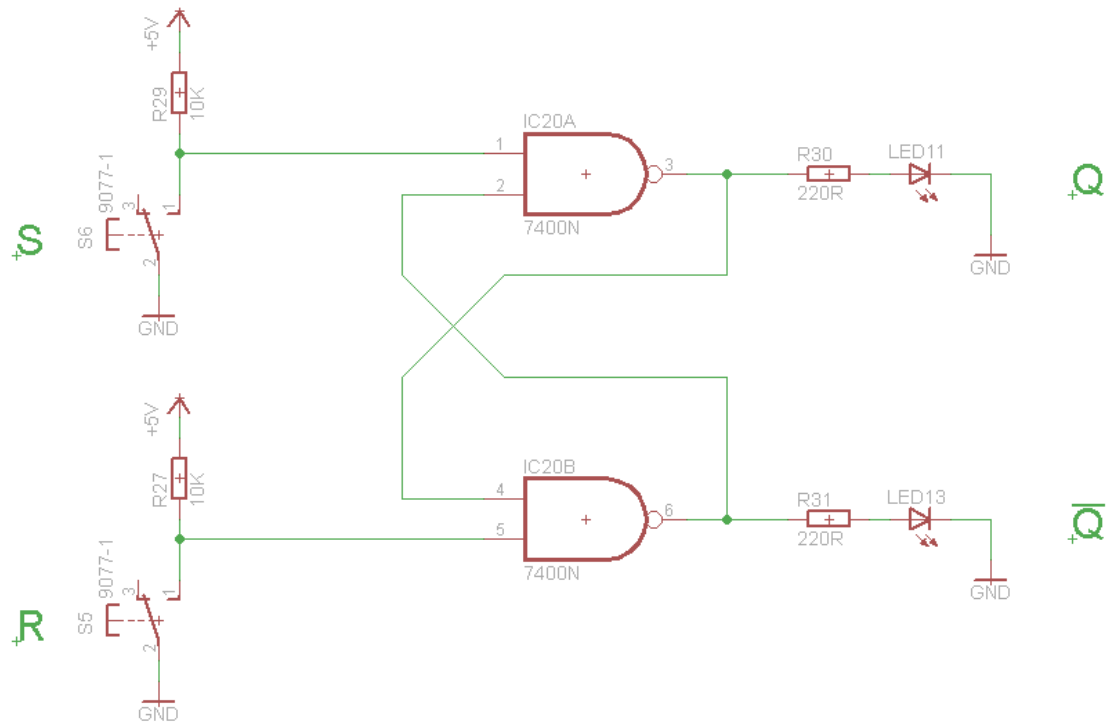
Alempi aste laskee yhteen molemmista yhteenlaskettavista vähemmän merkitsevät bitit A0 ja B0 ja tuottaa ulos summan vähiten merkitsevän bitin S0. Ylempi aste laskee vastaavasti yhteen enemmän merkitsevät bitit A1 ja B1 ja tuottaa ulos summan seuraavaksi eniten merkitsevän bitin S1.

Carry-signaalien käsittelyyn kannattaa kiinnittää huomiota:

- Alempien astien CIN on vedetty kiinteästi nollassa, koska ei ole mitään tästä alempaa astetta, josta carry voisi tulla.
- Alempien astien COUT on vedetty ylempien astien CIN:iin, jolloin mahdollinen alempien bittien (A0+B0) tuloksena saatu carry siirtyy otettavaksi mukaan seuraavaksi enemmän merkitsevien bittien (A1+B1) yhteenlaskussa. Periaate on vastaava kuin paperilla tehdyssä kymmenjärjestelmän yhteenlaskussa tapahtuva muistinumerojen käsittely.
- Ylempien astien COUT on vedetty omaan lediinsä, josta saadaan summan eniten merkitsevä bitti S2. Tämä siksi, että kahden kaksibittisen luvun summan esittämiseen tarvitaan 3 bittiä. Kaksibittisen luvun arvo on 0...3, joten kahden tällaisen luvun summa on maksimissaan 3+3=6 (binäärisenä 11+11=110), jonka esittämiseen tarvitaan 3 bittiä.

Lisää tietoa: <http://www.play-hookey.com/digital/combinational/adder.html>

RS-kiikku NANDeilla



Koko sekvenssilogiikan peruskytkentä: yksinkertaisin mahdollinen piiri, jonka lähtöjen tilat riippuvat myös aikaisemmista tulojen tiloista, eli kytkennällä on muistia.

Piirin lähdöt merkitään Q ja \bar{Q} . Tämä kuvaa sitä, että lähdöt ovat aina vastakkaisissa tiloissa: kun toinen on 0, toinen 1.

Piirin tulot ovat \bar{S} (set) ja \bar{R} (reset). Viiva päällä kuvaa sitä, että ne ovat alhaalla (L-tilassa) aktiivisia. Set- ja reset-toiminto ajatellaan ”päälähdön” Q kannalta: S asettaa Q :n ykköseksi, R nolaa sen. \bar{Q} on aina päinvastaisessa tilassa.

Piirillä on olemassa kielletty tila: jos sekä \bar{S} että \bar{R} vedetään yhtä aikaa aktiivisiksi (alas), molemmat lähdöt asettuvat 1-tilaan. Tästä tilasta palaututaan jompaankumpaan sallittuun tilaan heti, kun \bar{S} tai \bar{R} nousee ylös. Se, kumpi nousee ensin, määrää tilan, johon kiikku asettuu. Jos \bar{S} nousee ensin, ollaan taas sallitussa tilassa, jossa vain \bar{R} on alhaalla, joka nolaa Q :n ja asettaa \bar{Q} :n. Vastaavasti toisinpäin.

Kannattaa huomata, että kielletyssä tilassa ei ole mitään kiellettyä siinä mielessä, että jossain esim. kulkisi liikaa virtaa tai oltaisiin loogisen tulon kielletyllä alueella. Kiellettyä on vain se, että tässä rikotaan tehtyä sopimusta siitä, että lähdöt Q ja \bar{Q} ovat aina eri tiloissa.

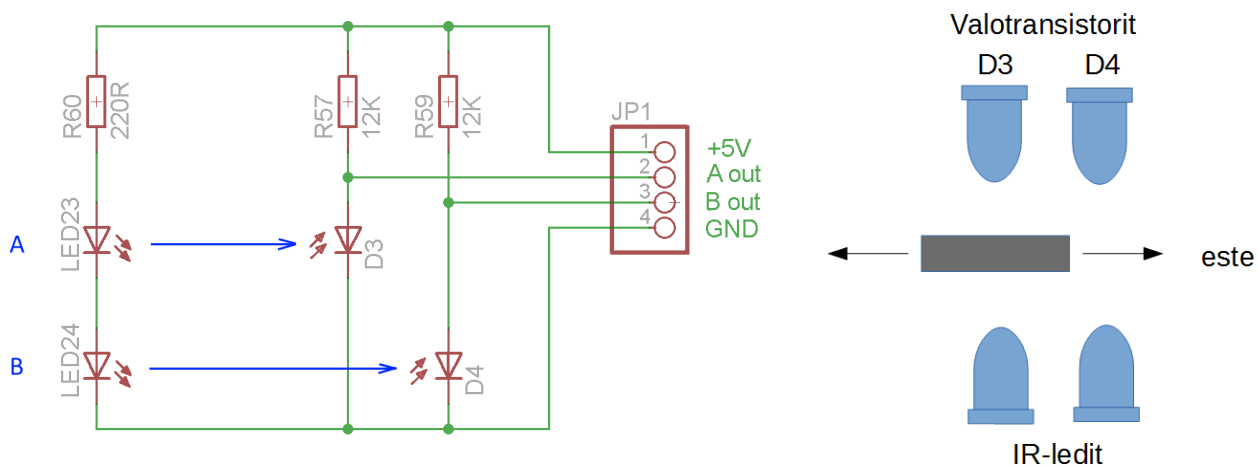
Kytkenässä painikkeet vetävät piirin tulon L-tilaan, eli S - tai R -painikkeen painaminen asettaa vastaavan signaalin aktiiviseksi.

Tässäkin lähtöihin kytketyt ledit on kytketty maata vastaan, jolloin ledi toimii loogisemmin: ledi palaa → lähtö=1, ledi pimeänä → lähtö=0. Voidaan korvata myös protokortin ledeillä.

Lisää tietoa: http://www.play-hookey.com/digital/sequential/rs_nand_latch.html

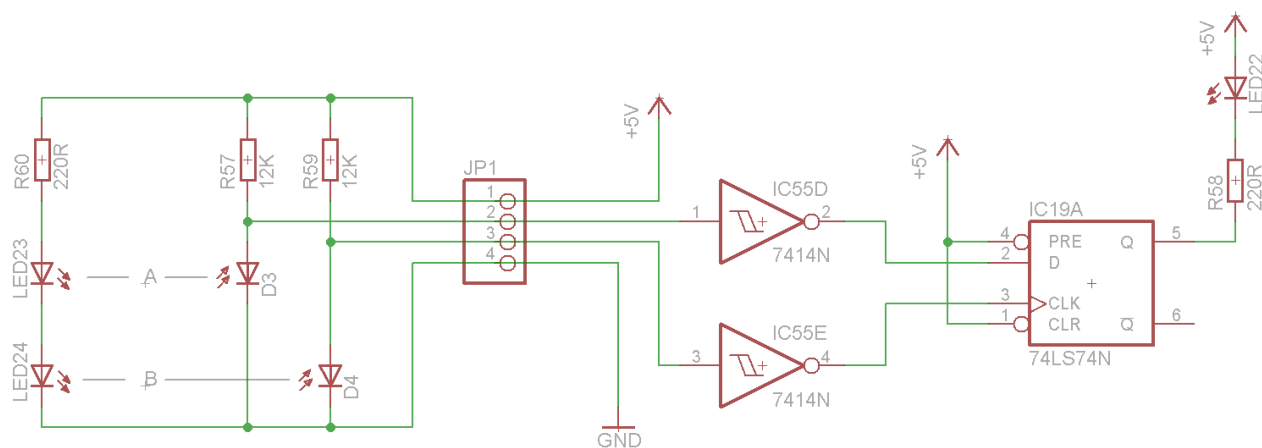
Liikkeen suunnan ilmaisim D-flip-flopilla

Tässä harjoituksessa käytetään apuna rakennelmaa, joka sisältää tällaisen kytkennän:



Ilmaisimessa on kaksi IR-ledi/valotransistoriparia (LED23/D3 ja LED24/D4). Normaalisti ledin valo osuu valotransistoriin, jolloin transistori vetää vastaavan lähdön (A out tai B out) alas. Jos väliin tuodaan este, transistori pimenee ja lähtö nousee ylös.

Tekemällä tämän perään alla näkyvä kytkentä, saadaan laite, joka sytyttää tai sammuttaa ledin LED22 sen mukaan, mihin suuntaan valotransistorien edessä tapahtui viimeksi liikettä.



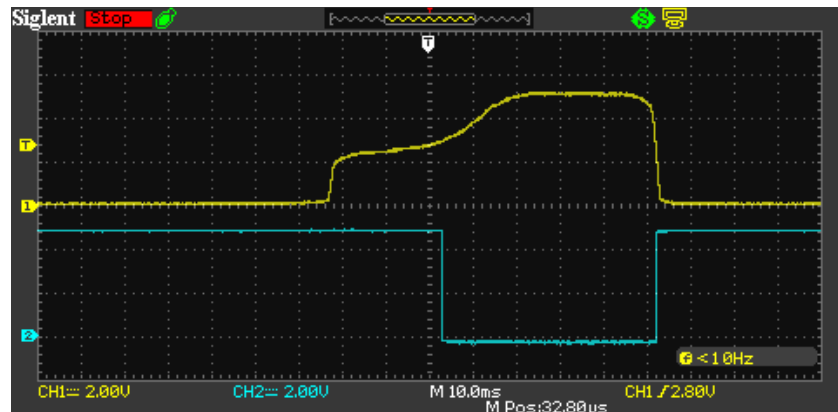
Kun valotransistori saa valoa, se johtaa sähköä ja vetää schmitt-invertterin tulon alas. Vastaavasti kun valotransistori on pimeänä, se ei johda sähköä ja ylösvetovastus vetää invertterin tulon ylätilaan. Joten molempien invertterien lähdöt käyttäytyvät näin:

H = valoa

L = pimeää

Tässä on käytetty schmitt-tyyppistä invertteriä, koska transistorin johtavuus vaihtelee valon määrän mukaan ja invertterin tulo saattaisi joutua helposti tavallisen digitaalitulon kielletylle alueelle. Schmitt-invertterillä kiellettyä aluetta ei ole ja D-kiikulle vietävät tilanmuutokset saadaan täsmällisiksi.

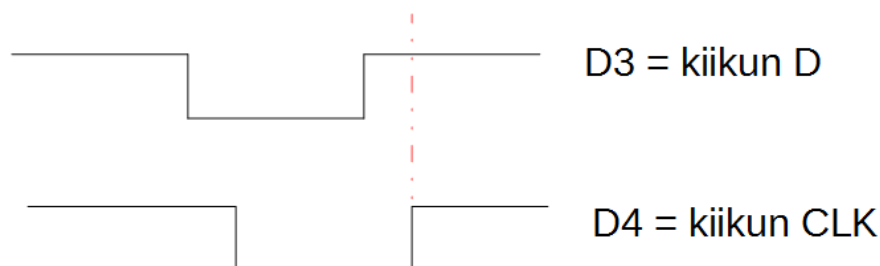
Alla olevassa oskilloskooppikuvassa näkyy jännitteet, kun este ohittaa valotransistorin. Keltainen käyrä on ilmaisimen lähtö eli schmitt-triggerin tulo, sininen käyrä schmitt-triggerin lähtö.



Kun este liikuu edellisen sivun kuvassa vasemmalta oikealle, tapahtuu seuraava sekvenssi:

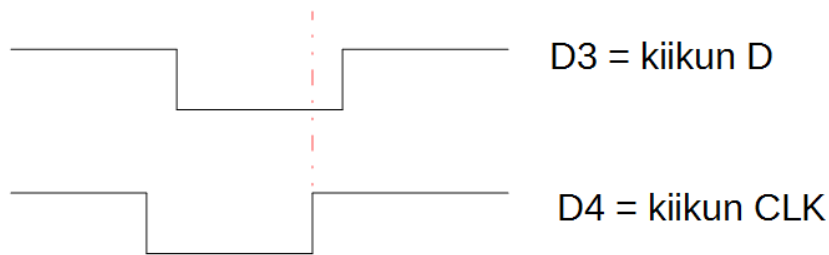
- D3 pimenee
- D4 pimenee
- D3 saa valoa
- D4 saa valoa

D3:n ja invertterin tuottama signaali on viety D-kiikun D-tuloon, D4:n signaali D-kiikun kellotuloon. Pulssikaaviona tapahtumat etenevät siis näin:



D-kiikun ominaisuus on, että se liipaisee talteen sen tilan, joka D-tulossa esiintyy sillä hetkellä kun CLK-tulossa on nouseva reuna. Tässä tapauksessa kiikun tilaksi liipaistuu siis H-tila. Tämä tila ilmestyy lähtöön Q, jolloin ledi sammuu.

Esteen liukuessa oikealta vasemmalle tapahtumat etenevät päinvastaisessa järjestyksessä ja pulssikaaviosta tulee tämän näköinen:



Nyt kiikun tilaksi tuleeekin L-tila ja ledi syttyy.

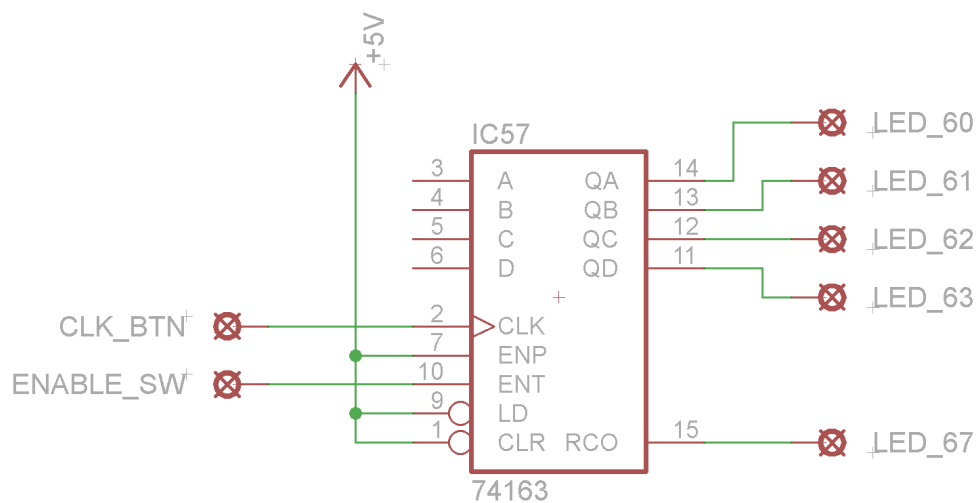
Kuten pulssikaavioista näkyy, toiminnan kannalta on oleellista, että este on niin leveä, että se ledejä ohittaessaan pystyy pimentämään molemmat ledit yhtä aikaa.

Tämän kaltaista kytkentää käytetään yleisesti erilaisissa pyörimissuunnan ilmaisimissa. Esim. jokaisessa perinteisessä mekaanisessa pallohiiressä näitä on kaksi: toinen ilmaisee pystysuuntaisen liikkeen suunnan ja toinen vaakasuuntaisen liikkeen. Itse asiassa, rullahiiressä näitä on kolmekin: myös rullan pyörityssuunta havaitaan tämän tyyppisellä kytkennällä.

Lisätietoa D-kiikusta: http://www.play-hookey.com/digital/sequential/d_nand_latch.html

Pallohiiren enkooderista: <http://electronics.stackexchange.com/questions/15481/how-does-a-ball-mouse-know-the-direction>

Synkroninen laskuri



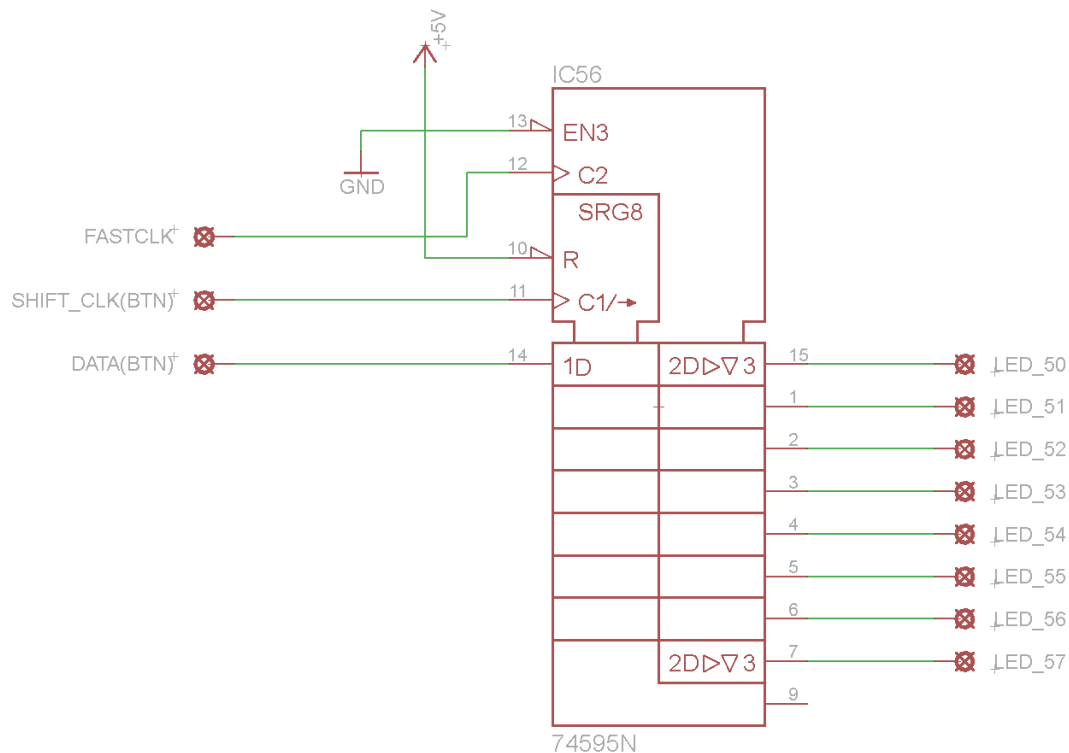
Tämä on yksinkertainen peruskytkentä, jolla demotaan synkronisen laskurin toimintaa. 74163 on 4-bittinen synkroninen laskuri, jossa myös mahdollisuus ladata rinnakkaismuodossa 4-bittinen arvo, josta laskeminen alkaa, sitä ei tässä käytetä.

Laskurin 4 lähtöä (QA...QD) on viety protokortin ledeille, joista voidaan seurata laskennan edistymistä. Myös RCO-lähtö (toiselta nimeltä TC, Terminal Count) on viety ledille, jotta sen asettuminen nähdään. Kellotuloon CLK tuodaan protokortin painike, ja ENT-tuloon (toiselta nimeltä CE, Clock Enable) tuodaan protokortilta kytkin.

Painelemalla painiketta nähdään, että lähdöissä näkyvä binääriluku askeltaa yhdellä jokaisella kellopulsilla ja pyörähtää ympäri 1111 → 0000. Kun lähdöissä on 1111, RCO (TC) asettuu. Kytkimellä ENT (CE) voidaan laskenta estää tai sallia.

CLK-tuloon voi tuoda myös protokortin SLOWCLK:in, jolloin laskuri laskee tämän tahdissa.

Siirtorekisteri



Piiri 74HC595, tuttujen kesken viis-ysi-viitonen, on paljon käytetty piiri mikrokontrollerien I/O-laajennuksiin, ja Arduinollekin löytyy jopa valmiit kirjastot sen liittämiseksi. Tässä sitä käytetään siirtorekisterin toiminnan demoamiseen. Piirissä on 8-asteinen siirtorekisteri, jonka lähdöissä on 8-paikkainen rinnakkaisrekisteri, johon siirtorekisterin sisältö voidaan liipaista halutulla hetkellä. Tämän perässä on vielä kolmitilalähdöt.

Tässä meitä kiinnostaa vain siirtorekisteritoiminnallisuus, joten rinnakkaisrekisteri ohitetaan tuomalla sen liipaisupulssiin (nasta 12) protokortilta FASTCLK, joka on 25 MHz kellopulssi. Eli siirtorekisterin lähtöjen tila liipaistaan rinnakkaisrekisteriin 25 miljoonaa kertaa sekunnissa, joten siirtorekisterin lähtöjen tilojen muutokset näkyvät käytännössä välittömästi piirin lähdöissä. Kolmitilalähdöt kytketään pysyvästi aktiivisiksi vetämällä niitä ohjaava nasta 13 maahan.

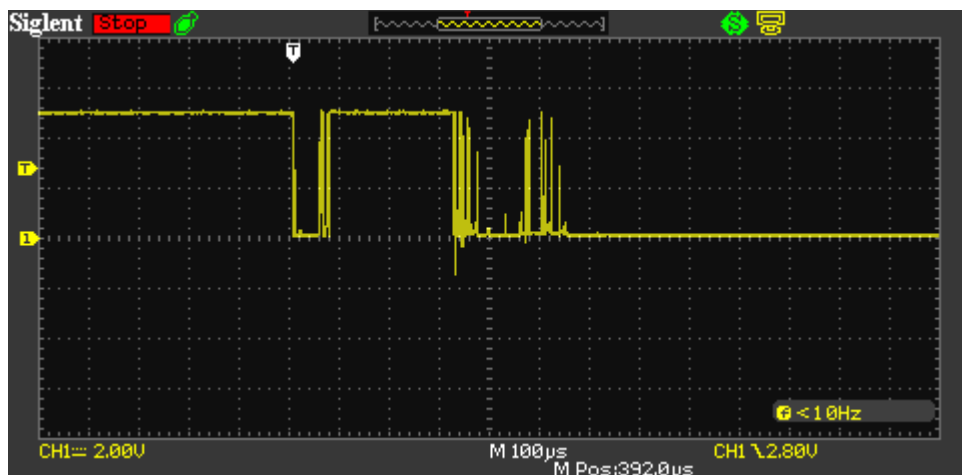
Kellopulssiin (nasta 11) ja datatuloon (nasta 14) tuodaan kaksi protokortin painiketta. Lähdöt kytketään protokortin ledeille.

Painelemalla panikkeita voi ledeistä tarkkailla, miten siirtorekisteri toimii. Datapainikkeen tila siirtyy ensimmäiseen lähtöön (nasta 15, LED0) sillä hetkellä, kun painetaan kellopulssipainiketta. Samalla muiden lähtöjen tilat siirtyvät yhden pykälän kohti enemmän merkitsevää päätä ja viimeisen lähdön (nasta 7) tila katoaa.

Kannattaa kokeilla myös toiminnan "automatisointia" kytkemällä kellopulssipainikkeen tilalle protokortin SLOWCLK ja säätämällä tähän sopiva taajuus.

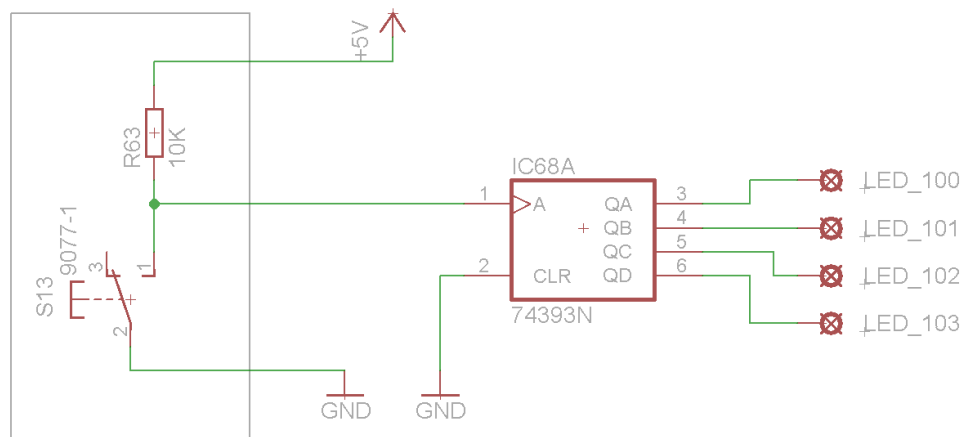
Kytkinvärähtelyjen poisto

Kaikissa mekaanisissa kytkimissä ja painikkeissa esiintyy kytkinvärähtelyä sillä hetkellä kun kytkin sulkeutuu tai avautuu. Nämä ovat nopeita pulsseja, mutta eivät niin nopeita, ettei digitaalinen piiri pystyisi laskemaan niitä erillisiksi pulsseiksi. Silloin kun painikkeella yritetään ohjata sen tyyppistä piiriä, jossa pulssien määrä on olleellinen (esim. laskurin tai siirtorekisterin kello-tulo) tästä tulee ongelma. Oskilloskoppikuva erään kytkimen käytöksestä painallushetkellä:



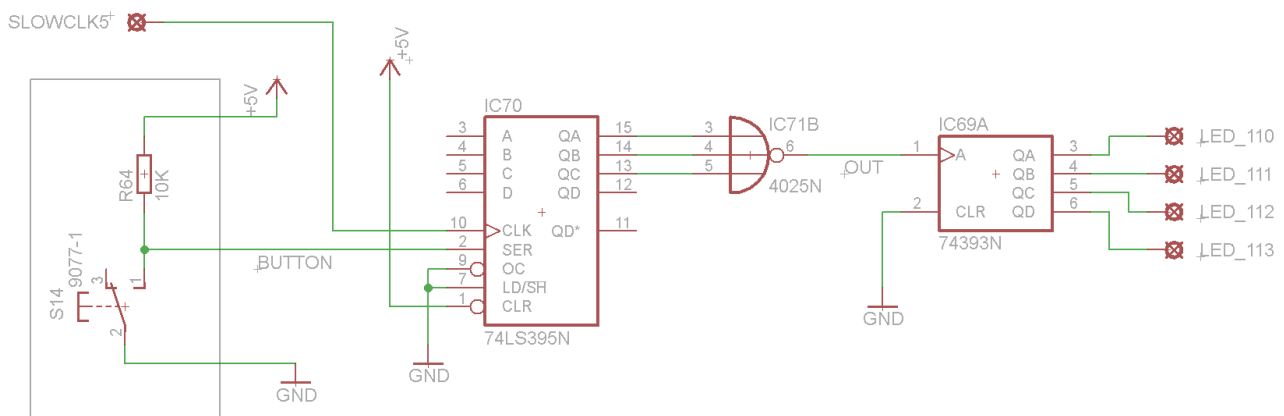
Edellä esitetyissä laskuri- ja siirtorekisterikytkennöissä kello-tuloon on tuotu protokortin painike. Protokortilla oleva CPLD sisältää logiikan, joka tekee painikkeille värähtelyjen poiston, joten kytkennät toimivat oikein. Tässä rakennetaan vastaavalla periaatteella toimiva värähtelynpoistokytkentä erillispiireillä.

Värähtelyjen esille saamiseksi käytetään yksinkertaista laskurikytkentää:



74393 sisältää kaksi 4-bittistä ripple counteria, joista käytetään toista. Tässä pitää siis käyttää tavallista irtopainiketta, ei protokortin valmiiksi suodatettua painiketta. Lähdöt kytketään protokortin ledeille, joista voidaan tarkkailla laskurin askeltamista. Painelemalla painiketta on helppo havaita, että laskuri etenee epämääräisesti, useimmilla painalluksilla enemmän kuin yhden pykälän, joskus hyvinkin monta.

Tilanne saadaan hallintaan kytkemällä painikkeen ja laskurin kellotulon väliin tällainen logiikka:



74395 on 4-asteinen siirtorekisteri. Se sisältää myös rinnakkaislatausmahdollisuuden, mutta sitä ei käytetä tässä. Kellotuloon tuodaan protokortilta SLOWCLK, joka tässä käytössä kannattaa kääntää täysille. Siirtorekisterin datatuloon (nasta 2) tuodaan painike suoraan.

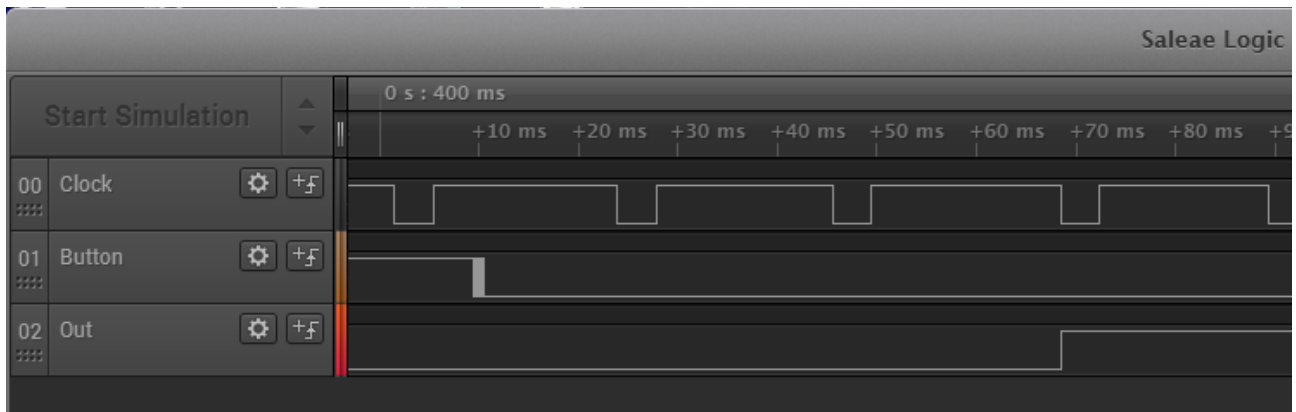
Voidaan ajatella, että siirtorekisteri ottaa näytteitä painikkeen tilasta jokaisella kellopulssin laskevalla reunalla. Uusin näyte ilmestyy rekisterin lähtöön QA ja edelliset näytteet siirtyvät samalla eteenpäin kohti QD:tä. Näin ollen kolme viimeksi otettua näytettä on joka hetki näkyvillä siirtorekisterin lähdöissä QA, QB ja QC.

Koska painike antaa painettaessa 0-tilaa, voidaan luottaa siihen, että jos kolme viimeisintä näytettä ovat kaikki nollija, painike on varmasti painettu pohjaan, ja myös lopettanut värähtelyt. Tällöin voidaan antaa painallusta vastaava pulssi ulos. Tämä tehdään kytkemällä 3-tuloinen NOR-portti siirtorekisterin lähtöihin. Tässä on käytetty 4000-sarjan piiriä 4025. NOR-portin lähtö pysyy nollana niin kauan, kun sen yhdessäkin tulossa on 1-tila. Vasta kuin kaikki 3 tuloa menevät nollassi, sen lähtö nousee 1-tilaan.

Toiminnan kannalta oleellista on, että kellopulssi on selvästi harvempi kuin värähtelyjen kesto. SLOWCLK:in maksimitaajuus on alle 100 Hz, eli näytteitä otetaan reilun 10 ms välein. Värähtelyjen kesto on enintään muutama millisekunti, joten ehto toteutuu hyvin.

Nyt kun painellaan painiketta, voidaan todeta, että laskuri askeltaa siististi yksi pykälä kerrallaan, eli värähtelyt ovat kadonneet. Se askeltaa nyt painiketta päästettäessä, koska laskuri on laskevalle reunalle herkkä, ja NOR invertoi pulssin.

Alla näkyvässä kuvassa on logiikka-analysaattorilla kaapattuja pulsseja kytkennästä. Ylimpänä on kellopulssit, Button-rivillä näkyy painikkeen painallus. Paksumpi pystyviiva painikkeen painallushetkellä on kytkinvärähtelyt, eli niiden kesto suhteessa kellopulsseihin on lyhyt. Out-rivi on NOR-portin lähtö, joka vie suoraan laskurille. Siinä ei ole ylimääräisiä värähtelyjä. Kuvasta näkyy, että painikkeen painalluksen jälkeen kolmannen kellopulssin tullessa (kolme 0-näytettä) lähtö asettuu 1-tilaan.



Painiketta päästettäessä ensimmäinen 1-tilainen näyte riittää nollaamaan NOR-portin lähdön. Tämä ei kuitenkaan ole ongelma, koska seuraava näyte otetaan niin pitkän ajan kuluttua, että painike on silloin jo stabiloitunut 1-tilaan, vaikka ensimmäinen 1-näyte olisikin satuttu ottamaan keskeltä päästövärähtelyjä.

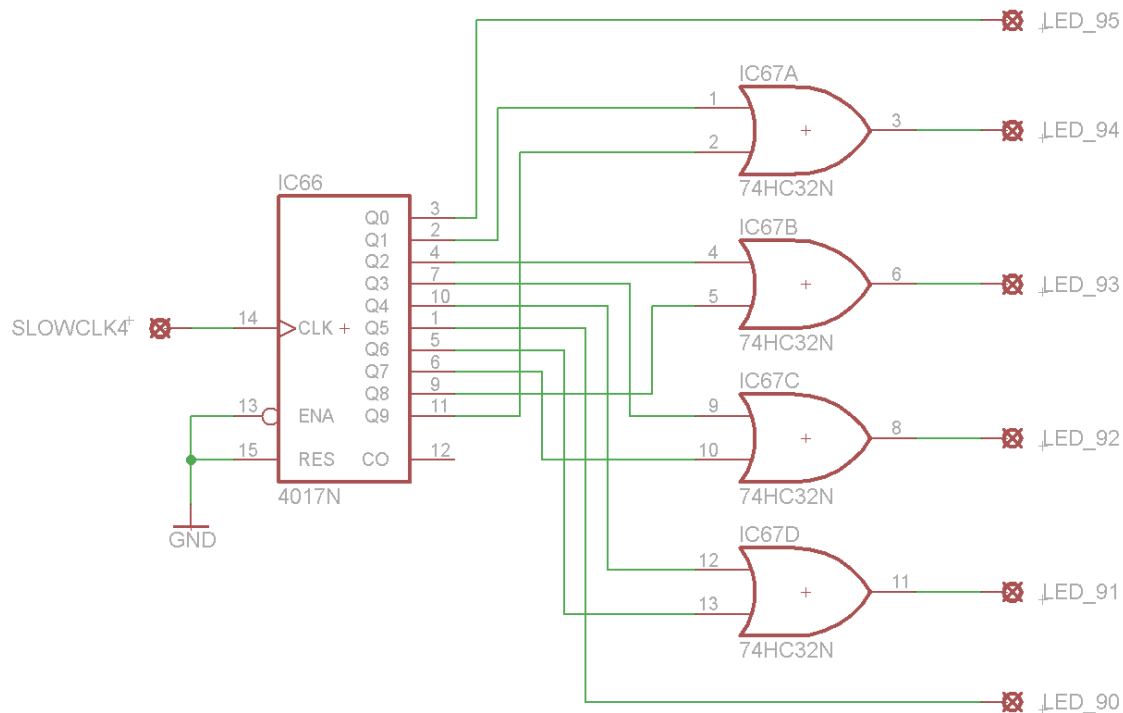
Yleensä kaikenlainen suodattaminen aiheuttaa viivettä ja niin tässäkin: lähtö aktivoituu vasta muutaman 10 ms kuluttua todellisesta painalluksesta. Kun tehdään ihmiselle tarkoitettua käyttöliittymää, tällä ei ole käytännön merkitystä, mutta jossain toislaisessa käytössä tämäkin saattaa osoittautua ongelmaksi. Tällöin on vain pyrittävä minimoimaan viive, ei kuitenkaan niin pieneksi, että suodatus lakkaa toimimasta. Tässä kytkennässä on se etu, että viivettä on helppo säätää muuttamalla kellopulssien taajuutta. Tätä voi kokeilla itsekin: säädetään SLOWCLK hitaammaksi, jolloin viiveestä tulee jo ihmisen havaittava.

Kaikki mitä olet ikinä halunnut tietää kytkinvärähtelyistä ja niiden poistosta:

<http://www.eng.utah.edu/~cs5780/debouncing.pdf>

Larson-skanneri

Larson-skannerissa on ledirivi, jossa yksi ledi juoksee edestakaisin. Oikeaoppisesti tähän liittyisi myös viereisten ledien kirkkaussäätö, mutta tässä tehdään nyt yksinkertainen perusskanneri.



Tässä on käytetty 4000-sarjan piiriä 4017, joka sisältää dekaadilaskurin ja 10-lähtöisen demultiplekserin samoissa kuorissa. Joka hetki yhdessä lähdoistä Q0...Q9 on 1-tila, kaikissa muissa 0-tila (ns. one-hot-tyyppinen lähtö). Tuomalla kellopulsseja tälle lähdoissä oleva 1-tila siirtyy jokaisella pulssilla seuraavaan, ja Q9:n jälkeen siirrytään taas Q0:aan.

Tämä on jo hyvä lähtökohta Larsonille, mutta edestakaisin juokseminen ei ratkea ihan pelkästään tällä. Tässä on tehty neljästä OR-portista lisälogiikka one-hot-lähdön perään. Ajatus on, että kaksi eri lähtöä pystyy sytyttämään kunkin ledin. Reunimmaisten kohdalla riittää yksi lähtö. Näin 10 lähtöä riittää yhteensä 6 ledille ja OR-portitkin optimoituvat mallikkaasti, niitä kun on 4 per mikropiiri.

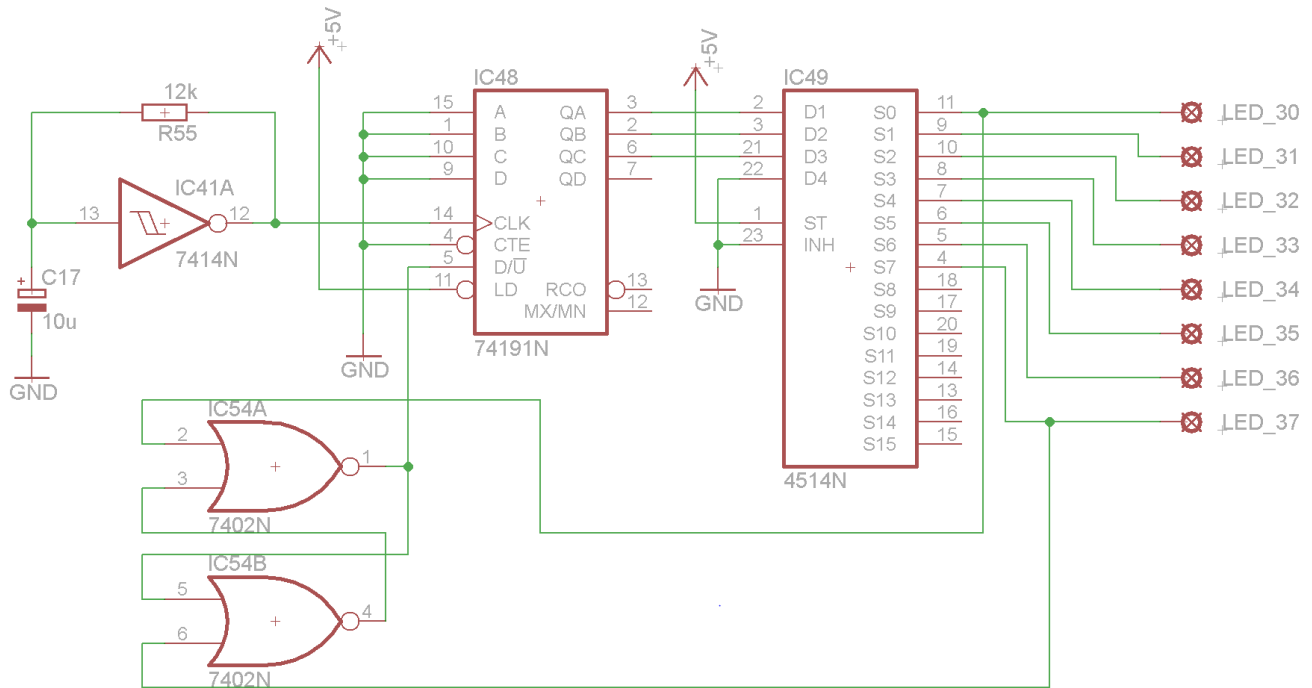
Logiikka toimii siten, että reunimmainen ledi syttyy vain Q0:sta, seuraava sekä Q1:stä että Q9:stä, seuraava Q2:sta ja Q8:sta jne. Toinen reuna syttyy vain Q5:sta. Tällöin ensimmäinen puolikas (Q0...Q4) käy ledit läpi toiseen suuntaan, jälkimmäinen puolikas (Q5...Q9) toiseen suuntaan.

Lähdöt on viety suoraan protokortin ledeille ja skanneria askelletaan SLOWCLK-lähdöllä.

Variaatioita Larson skannerista

Tässä on esitelty muutama muu kytkentä, joilla saadaan myös aikaan Larson-skanneri. Tämä toimikoon osoituksena, että näinkään yksinkertaisen toiminnon toteuttaminen ei ole millään tavalla yksiselitteistä.

Versio 1: Ylös/alas-laskuri ja ylhäällä aktiivinen demultiplekseri



Kytkenän ytimenä on piiri 74191, joka on 4-bittinen synkroninen ylös/alas-laskuri. Siinä on myös mahdollisuus ladata alkuarvo laskuriin, mutta sitä ei tässä käytetä. Sen sijaan oleellinen toiminto tässä on DOWN/ $\overline{\text{UP}}$ -tulo, joka määrää, mihin suuntaan laskurin lukema etenee kellopulssein tullessa. Ajatus siis on saada laskuri laskemaan edestakaisin väliä 0...7, järjestyksessä 0,1,2,3,4,5,6,7,6,5,4,3,2,1,0,1,2,... jne.

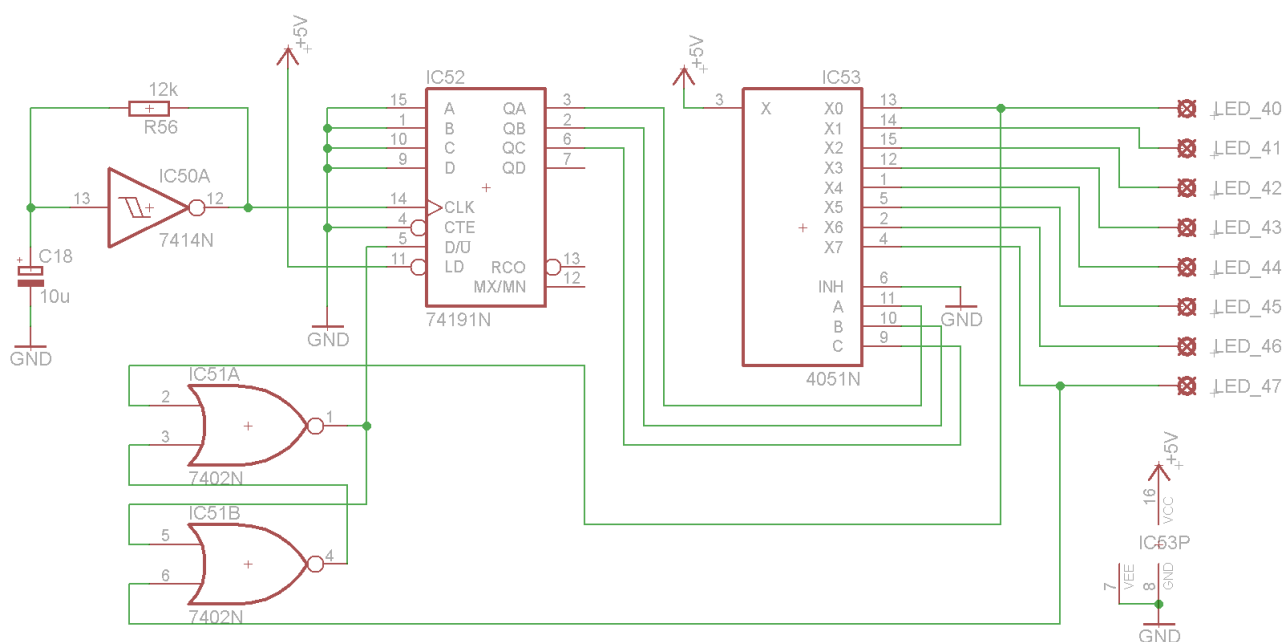
Laskurin perään on kytketty 4514, joka on 4:16 demultiplekseri. Siitä on tässä käytetty vain puolet, eli se toimii 3:8-demuxina. Laskurin 3 alinta lähtöä (QA...QC) on kytketty demuxin 3 alimpaan tuloon (D1...D3), ja demuxin lähdöistä on käytössä 8 alinta (S0...S7). Demuxin lähdöistä 1-tilassa on aina se, jota vastaava binääriluku on muxin tuloissa, kaikki muut lähdöt ovat nolatilassa. Kun laskuri laskee edestakaisin 0...7 välillä, 1-tila vaeltaa demuxin lähdöissä edestakaisin.

Ledejä ohjaavat lähdöt on kytketty suoraan demuxin lähtöihin, josta ne lähtevät protokortin ledeille.

Laskurin suunnan hallintaa varten on tehty perus-RS-kiikku kahdesta NOR-portista (7402). Kiikun lähtö (Q) ohjaa suoraan laskurin DOWN/ $\overline{\text{UP}}$ -tuloa. Kun kiikku on asettuneena,

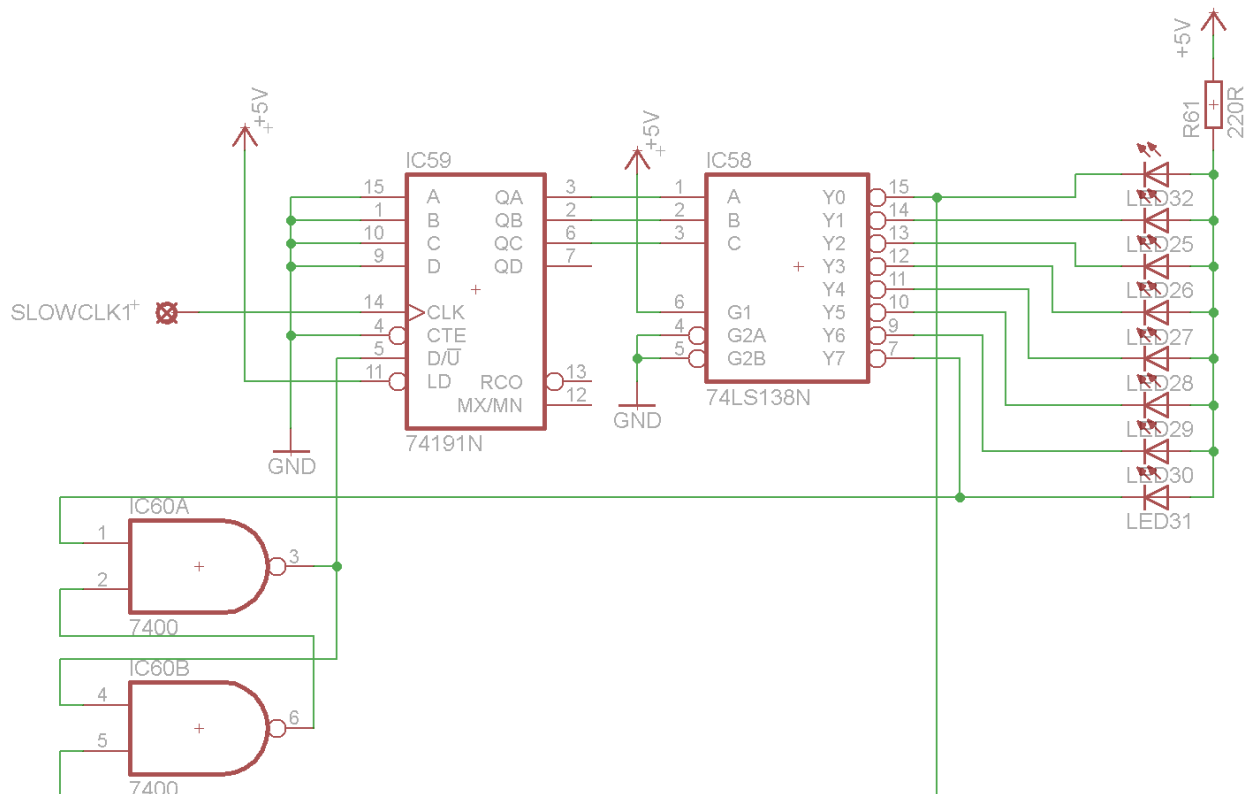
Kun laskuri on laskemassa ylöspäin ja päästään arvoon 7, demuxin lähtö S7 asettuu. Samalla RS-kiikku asettuu ja asettaa laskurin DOWN/ \overline{UP} -tulon, joten laskuri alkaa seuraavasta kellopulssista alkaen laskea alaspäin. Kun alaspäin laskemalla päästään arvoon 0, demuxin lähtö S0 asettuu ja nollaa RS-kiikun, jolloin aletaan taas laskea ylöspäin. Tämä sykli toistuu loputtomasti. RS-kiikun tehtävänä on siis muistaa, mihin suuntaan ollaan tällä hetkellä laskemassa.

Versio 2: Ylös/alas-laskuri ja analoginen demultiplekseri



Tämä on lähes sama kytkentä kuin edellä, mutta tarpeettoman isokokoinen 4514 on korvattu analogisella demultiplekserillä 4051, jolla kytketään lähtöihin läpi tuloon (nasta 3) kytkettyä +5V jännitettä.

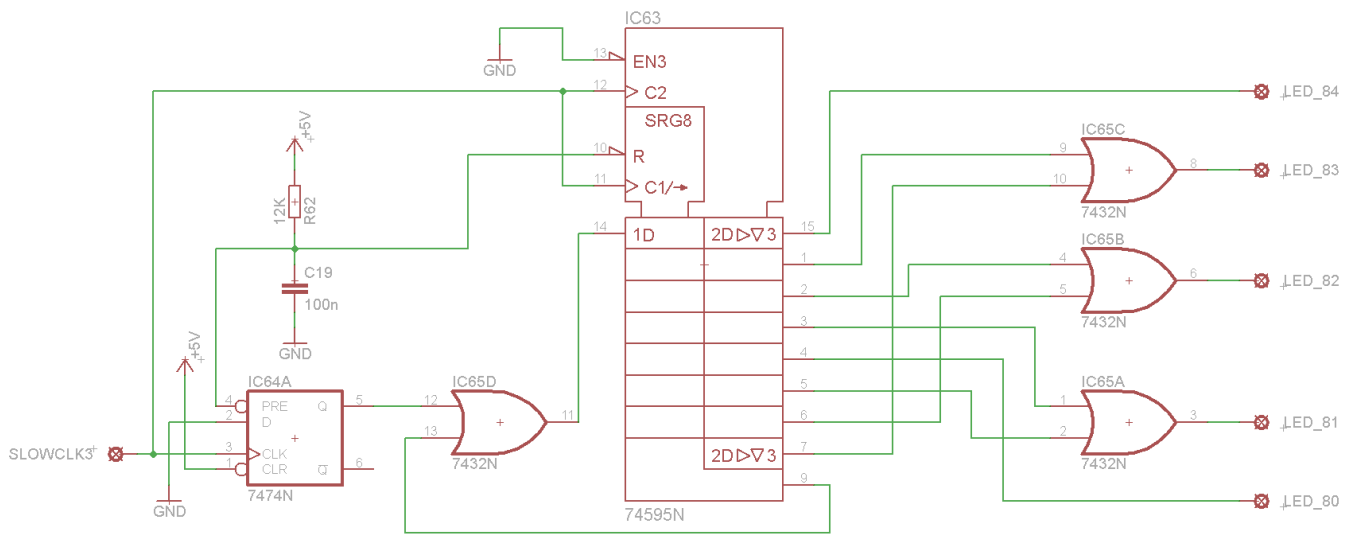
Versio 3: Ylös/alas-laskuri ja alhaalla aktiivinen demultiplekseri



Perusratkaisu on edelleen sama, mutta demultiplekserinä on 74138, jossa on alhaalla aktiiviset lähdöt. Tällöin ei voi käyttää enää protokortin ledejä, jotka olettavat saavansa ylhäällä aktiivista signaalia. Todellisten ledien kytkentä menee tässä melko yksinkertaiseksi, kun niille riittää yksi yhteinen sarjavastus: tässähän tiedämme, että ledeistä jokaisena hetkenä tasan yksi on päällä.

Koska RS-kiikulle tulevat signaalit ovat nyt alhaalla aktiivisia, senkin toteutus pitää vaihtaa NAND-pohjaiseksi. Muuten toiminta on vastaava.

Versio 4: siirtorekisteri ja OR-portit

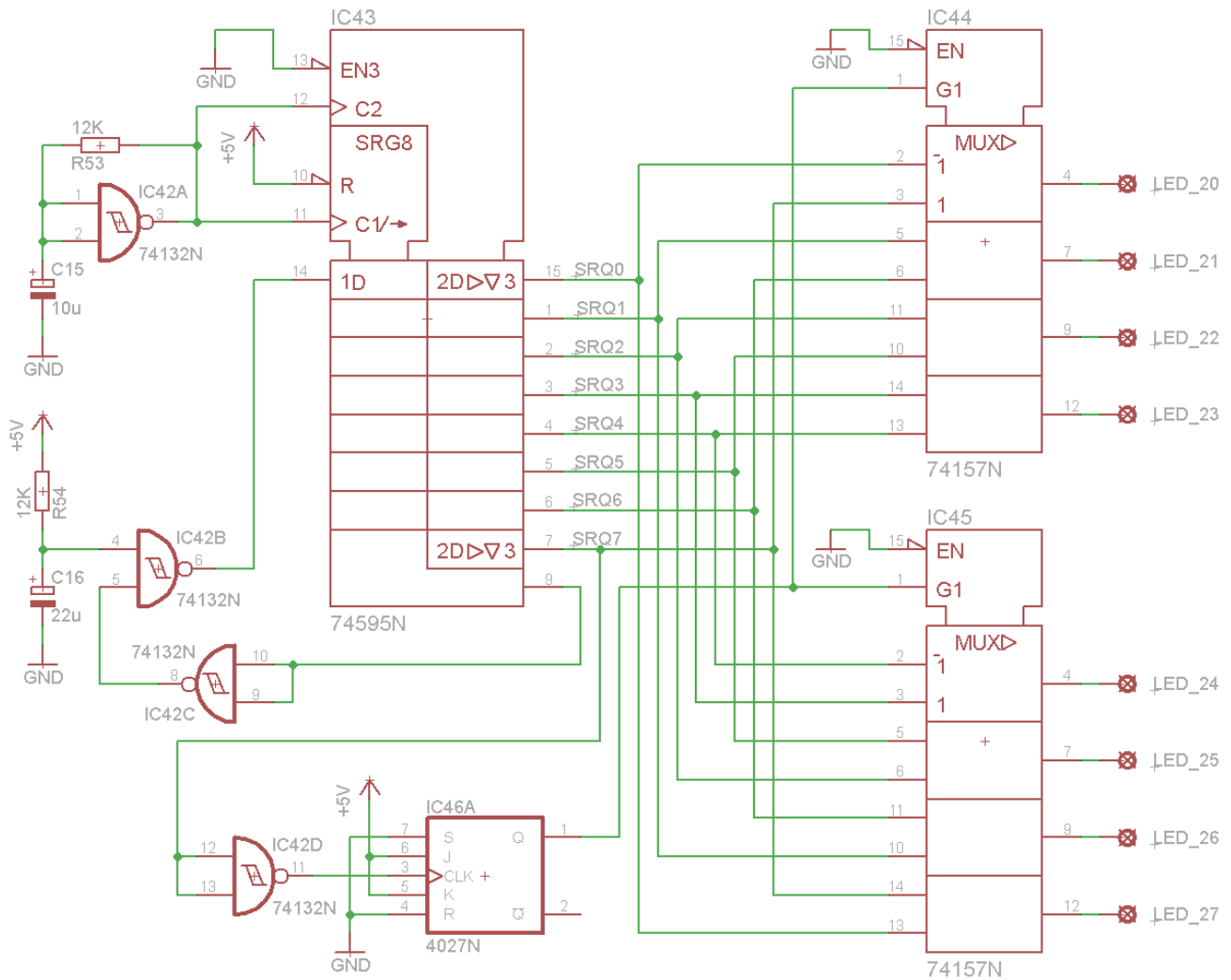


Tässä on sama ajatus kuin ensimmäisenä esitellyssä versiossa: tehdään yhteen suuntaan skannaava one-hot-lähtö, josta yhdistellään OR-porteilla kaksisuuntainen. Koska siirtorekisterissä on 8 lähtöä, ledejä saadaan vain 5.

Siirtorekisterin saaminen one-hot-lähtöiseksi ei ole ihan yksinkertaista: vaikka 1-tilan kierrättäminen silmukalle kytketyssä siirtorekisterissä on periaatteessa helppoa, tarvitaan logiikkaa, joka sähköt kytkettäessä syöttää yhden 1-tilan rekisteriin. Sen jälkeen se kyllä kiertää siellä.

RC-piiri R62/C19 tuottaa sähköit kytettäessä lyhyen alhaalla käyvän pulssin, joka pakkoasettaa D-kiikun lähdön 1-tilaan ja samalla nolaa siirtorekisterin. Kun ensimmäinen kellopulssi tulee, D-kiikun lähdön 1-tila näkyy OR-portin läpi siirtorekisterin tulossa (nasta 14) ja tulee kellotetuksi siirtorekisteriin. Sama kellopulssi liipaisee D-kiikkuun D-tulossa kiinteästi olevan 0-tilan, joten seuraavat kellopulssit kellottavat siirtorekisteriin aina 0-tiloja. Kunnes 8. pulssin jälkeen alkuperäinen 1-tila on siirtynyt siirtorekisterin nastaan 9 ja pääsee OR-portin läpi siirtorekisterin tuloon, ja tulee kellotetuksi taas sisään. Näin alussa kerran sisään syötetty 1-tila kiertää rekisterissä loputtomasti.

Versio 5: siirtorekisteri ja multiplekserit



Tämän varsin monimutkaisen kytkennän vasen puoli on pohjimmiltaan sama kuin edellä: asetetaan siirtorekisteri kierrättämään 1-tilaa one-hot-tyyppisesti, ja saadaan yksisuuntainen 8-asteinen skanneri. Käytännön toteutus, erityisesti käynnistykseen 1-tilan sisäänkytön osalta poikkeaa hiukan edellisestä.

Sen sijaan periaate, miten yksisuuntainen skanneri muutetaan kaksisuuntaiseksi, poikkeaa paljonkin edellisistä. Suurin etu OR-porttivirityksiin verrattuna on, että pystytään ohjaamaan kahdeksaa lediä. Tähän käytetään 8 kpl 2:1 multipleksereitä, joita ohjataan samassa tahdissa. Multiplekserit kytkvät ledit joka toisella pyyhkäisyllä suoraan siirtorekisterin lähtöihin, joka toisella ne kytketään ristiin. Tällöin ledilähdöissä joka toinen skannaus menee päinvastaiseen suuntaan kuin joka toinen.

JK-kiikku 4027 on kytketty toggle-flip-flopiksi ja sitä askelletaan aina, kun siirtorekisteri pääsee yhden skannauksen loppuun saakka. Toggle-flip-flopin lähtö ohjaa suoraan multiplekserien valintatuloja. Flip-flop on siis joka toisella pyyhkäisyllä 0-tilassa, joka toisella 1-tilassa.