

Maailmanennätys:

6-bittisten binäärilukujen
yhteenlasku dominopalikoista
rakennetulla tietokoneella

Tiivistelmä

Tutkimuksessa suunnittelimme ja rakensimme 4- ja 6-bittisen loogisilla porteilla toimivat dominopalikoista rakennetut summaimet. Kummatkin toimivat ja 6-bittinen tietokone pystyi laskemaan kahden kuusibittisen luvun yhteenlaskun, mikä on maailmanennätys. Tutkimuksessa kävimme myös läpi loogisten porttien XOR, OR, AND, NOT ja katkaisu toteuttamisen dominoilla. Lisäksi kehitimme suunnitelman dominosummaimesta, joka toimii aikasignaalilla ja mahdollistaisi laskut paljon pidemmälle kuin vain 6 bittiin asti. Aikasignaali on suunnitellut niin, että jokainen lisätty bitti lisää samanlaisen osan tietokoneen perään ja kasvattaa dominoiden määrää lineaarisesti.

Sisällysluettelo

Tiivistelmä	2
Sisällysluettelo	3
1. Johdanto.....	4
2. Dominotietokone	4
2.1 Mikä on dominotietokone?	4
2.2 Aikaisemmin tehdyt dominokoneet	5
2.3 XOR-portti – tärkein portti	5
2.4 OR-portti	6
2.5 Katkaisuportti.....	6
2.6 AND-portti.....	6
2.7 NOT-portti	7
2.8 Risteykset	7
2.9 Odotusmutkittelu (wait/buffer)	8
3. Summa loogisilla dominoporteilla	9
3.1 Merkinnät.....	9
3.2 Puolisummain	9
3.3 Kokosummain	10
3.4 Tietokoneen kaavio.....	10
4. Neljäbittinen tietokone.....	11
4.1 Tietokoneen materiaalit.....	11
4.2 Tietokoneen toteutus	11
5. 6-bittinen tietokone	13
5.1 Tietokoneen suunnittelu.....	13
5.2 Neljä- ja kuusibittisten summaimien toteutus.....	13
6. Dominotietokone aikasignaaliilla	15
6.1 Vertailu modernin tietokoneen suorituskykyyn	15
6.2 Aikasignaalin ongelmat	15
6.3 Signaalien tallennus	16
6.4 Bittien lisäys	16
7. Johtopäätökset.....	17
8. Työnjako	17
Lähdeluettelo	18

1. Johdanto

Dominotietokone ideana on ensimmäisiä kertoja mainittu ideana kirjassa "I am a Strange Loop (2007)". Kirjassa kerrotaan kuinka dominoilla voisi esimerkiksi tehdä mekanismin, joka tarkistaisi onko numero alkuluku vai ei. Dominot korvaavat sähkövirran ja niiden toiminta loogisissa porteissa voidaan saada matkimaan jännitettä oikeissa virtapiireissä tietokoneissa. Dominotietokone on hyvä tapa havainnollistaa ja opettaa tietokoneiden toimintaa ja havainnollistamaan kuinka pitkälle nykypäivän tietokoneteknologia on kehittynyt.

On muutamia, jotka yrittivät rakentaa sellaisen dominokoneen. Eräs tubettaja, Matt Parker, on tehnyt videon, kuinka hän tiimensä kanssa kaatavat 3- ja 4-bittistä tietokonetta. Löytyy myös hieman tietoa omanilaisesta koulusta, jossa tietokone toteutettiin 5-bittiseksi.

Tietokoneen suurin ongelma, joka meidän täytyy ratkaista, on, että osa porteista ei toimi, jos signaalit eivät tule loogisiin portteihin samaan aikaan. Edelliset tietokoneen rakentajat ovat vain hyväksyneet ongelman ja rakentaneet tietokoneen niin, että siinä on paljon viivyttäviä edestakaisin ajoitettuja menoja. Jos ongelma saataisiin ratkaistua tietokoneen rakentamisessa, säästyisi tuhansia dominoita ja tietokone ja portit toimisivat luotettavammin. Tässä projektissa suunnittelimme ja toteutimme 4-bittisen tietokoneen ajastuksilla, 6-bittisen maailmanennätys koneen ajastuksilla ja suunnittelimme lineaarisesti kasvavan aikasignaalia ja risteyksiä käyttävän koneen, jotka kykenevät laskemaan binäärilukujen yhteenlaskuja.

Tavoitteemme on rakentaa toimiva dominotietokone ja samalla yrittää rakentaa se sellaiseksi, että se voisi laskea mahdollisimman suuria lukuja, suurempia kuin ennen. Tavoitteemme on siis lyödä maailmanennätys.

2. Dominotietokone

2.1 Mikä on dominotietokone?

Tietokoneen toiminta perustuu loogisten porttien hyödyntämiseen laskimen luomiseksi. Loogiset portit ovat yhdistetty signaalinvälittäjillä toisiinsa niin, että portit muodostaisivat kokonaisuuden, joka seuraa eräänlaista algoritmia.

Dominotietokone toimii loogisilla porteilla, jotka rakennetaan dominopaloista. Kaatuva dominojono tarkoittaa bittiä 1 ja pystyssä pysyvä bittiä 0. Oikeissa tietokoneissa asia toimii samalla tavalla, mutta kaatuvien dominojonojen paikalla on korkea ja matala sähköjännite. Olemme asettaneet itsellemme säännöt projektiin, jotta tietokonetta ei olisi liian yksinkertaista toteuttaa. Dominotietokoneeksi luokittelemme dominoista tehdyn summaimen, jonka suunnitelma todistetusti toimii.

Noudatamme samoja periaatteita kuin aiemmat dominotietokoneita rakentaneet:

1. Tietokoneen tulee pysyä kaksiulotteisella tasolla
2. Sisääntulon ja ulosmenojen tulee pysyä tietokoneen ulkoreunoilla
3. Muiden asioiden kuin dominopalojen käyttö ei ole sallittua

Suunnitteleimme tietokonetta, jossa käytämme loogisia portteja AND, OR, XOR ja dominoille uniikkia katkaisuporttia. Rakentamamme 4- ja 6-bittiset tietokoneet tehtiin samoissa rajoissa kuin aikaisempien tietokoneen rakentajien tekemät koneet, eli ilman risteyksiä.

Päätimme lopulta tehdä erikseen tutkimuksen tietokoneiden erilaisuudesta risteyksillä ja ilman, tekemällä vielä toisen tietokoneen risteyksillä ja aikaisignaalilla. Aikaisignaali ratkaisisi suurimman osan tietokoneen ongelmista, joista suurin on dominoiden rajallinen määrä. Aikaisignaalin idea on viivytellä pitkään heti alussa ja antaa loogisten porttien suorittaa itsensä lähes loppuun mutta pysäyttää itsensä ennen sitä. Aikaisignaali tulisi kaikkien signaalien suoritettua ja viimeistelisi porttien suorituksen. Tämä poistaisi tarpeen sille, että jo neljä bittisen tietokoneen kanssa jouduimme tekemään yli 15 metrin verran viivytystä, että ajoitukset toimivat.

2.2 Aikaisemmin tehdyt dominokoneet

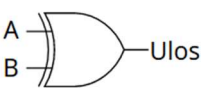
Tubettaja Matt Parker on tehnyt 4-bittisestä tietokoneesta videon kanavalleen "Stand Up Maths"[3]. Arvioimme sopivat rajoitteet 4-bittisen tietokoneen perusteella, joka näkyi Matt Parkerin kanavalla. Videolla ollut 4-bittinen tietokone oli rakennettu paikoittain liian tiiviisti, täten johtaen virheeseen tietokoneen toimivuudessa.

5-bittisen tietokoneen rakentajat, eivät ole kuvanneet eivätkä raportoineet tietokoneensa toimintaa hyvin, joten emme tiedä kuinka he ovat toteuttaneet tietokoneensa. Tietokoneesta oli vain julkaisu erään Omanilaisen koulun sivuilla kuvan kanssa [1], mutta muita todistusaineistoja ei ollut. Vaikka tietokoneen toimivuudesta ja olemassaolosta ei ollut kunnollisia todisteita, päätimme tehdä 6 bittisen tietokoneen, jotta se varmasti on uusi ennätys.

2.3 XOR-portti – tärkein portti

Käytämme tietokoneessa enimmäkseen XOR-portteja. XOR-portit toimivat niin, että ne päästävät läpi signaalin, jos sisään tulee vain yksi signaali, mutta keskeyttää signaalin, jos sisään tulee kaksi.

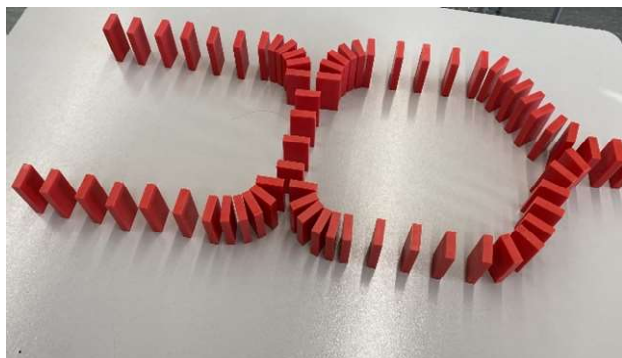
XOR-portti



XOR-portin symboli: kahdeksanmuotoinen muoto, jossa sisäpuolelta on kaksi kaarta, jotka eivät kosketa toisiaan. Kaksi sisäänkäyntiä on merkitty A ja B, yksi ulostulo Ulos.

A	B	Ulos
0	0	0
1	0	1
0	1	1
1	1	0

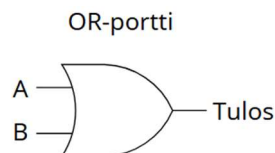
Kuva 1. XOR-portin sisään ja ulostulot



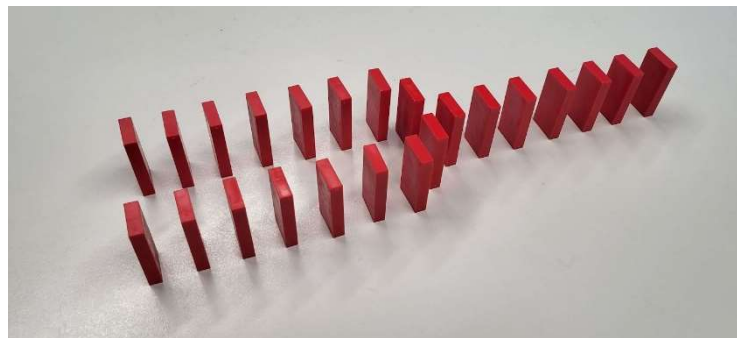
Kuva 2. XOR-portti dominoilla. Idea: Spanning Tree -YouTube-kanava [2]

2.4 OR-portti

Toinen yleinen portti mitä tietokoneemme käyttää on OR-portti. OR-portin toiminta on hyvin yksinkertainen ja siksi se on hyvin helppoa tehdä myös dominoilla. Jos porttiin tulee yksikään signaali sisään, portti päästää signaalin ulos.



A	B	Tulos
0	0	0
1	0	1
0	1	1
1	1	1

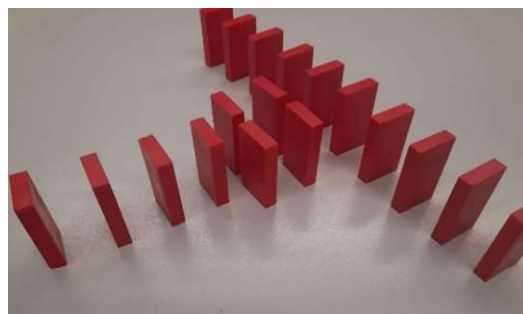
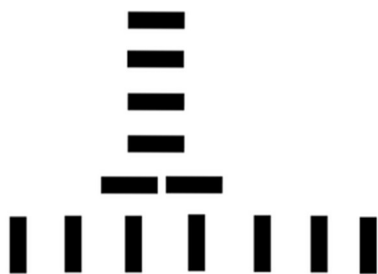


Kuva 3. OR-portin sisään- ja ulostulo -taulukko

Kuva 4. OR-portti dominoilla

2.5 Katkaisuportti

Katkaisuportit ovat hyvin tärkeä osa dominosummaimen toimintaa. Katkaisut estävät muistinumeron kulun seuraavaan, bittiin jos vain yksi signaali on tullut sisään XOR-porttiin. Aikaisignaalisessa laskimessa katkaisuportit ovat vielä suuremmassa roolissa.



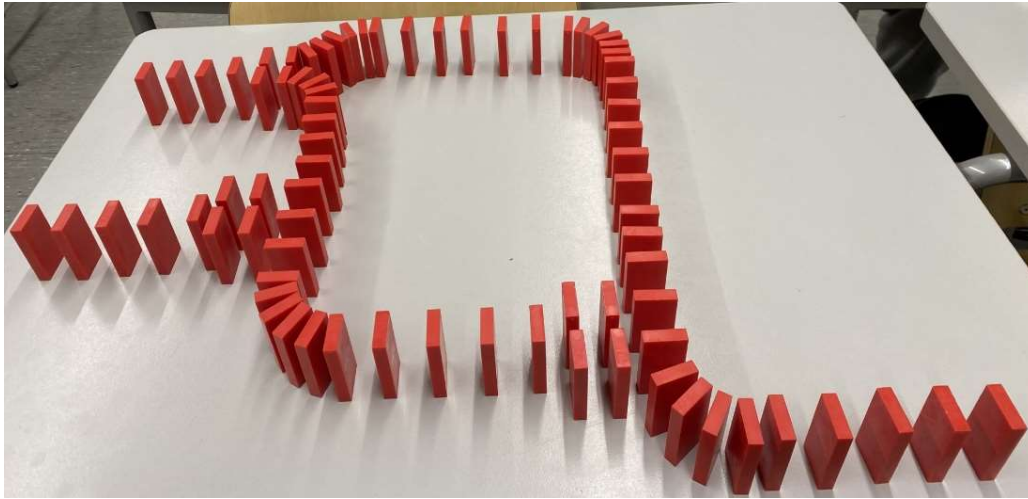
Kuva 5. Katkaisuportti dominoilla ylhäältäpäin. [2]

Kuva 6. Katkaisu portti dominoilla.

2.6 AND-portti

AND-portti dominoilla toimii hyvin samalla tavalla kuin aikaisignaali. Jos pelkkä katkaiseva signaali tulee, portista ei pääse mitään läpi. Jos pelkkä haarautuva signaali tulee, signaali katkaisee itsensä ja

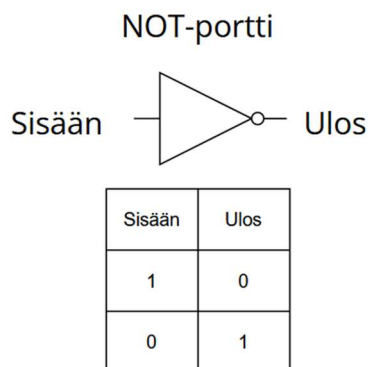
mitään ei pääse portista ulos. Mutta jos molemmat tulee, katkaiseva signaali katkaisee toisen signaalin itsensä katkaisu yrityksen.



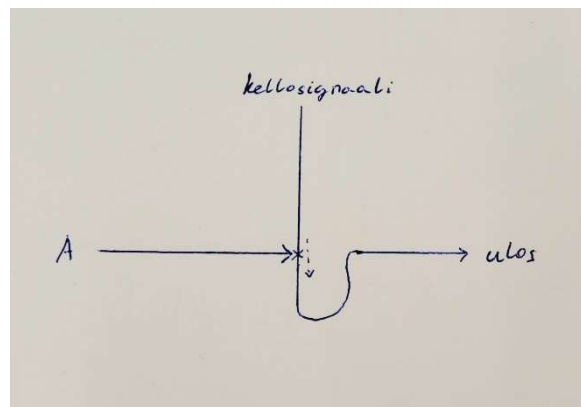
Kuva 7. AND-portti dominoilla. Idea: Spanning Tree -YouTube-kanava [2]

2.7 NOT-portti

Tavallisissa tietokoneen summaimissa usein käytetään NOT-portteja, jotka eivät kuitenkaan ole mahdollisia dominoilla ilman aikaisignaalia. NOT-portit toimivat niin, että jos porttiin ei tule signaalia se päästää signaalin läpi.



Kuva 8. NOT-portin sisään- ja ulostulot.



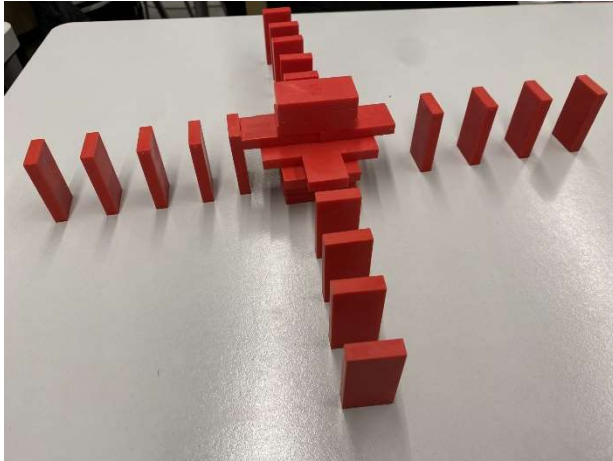
Kuva 9. NOT-portti dominoilla

NOT-portin toiminta ei kuitenkaan ole niin kriittisen tärkeä. Tärkeämpi ominaisuus tietokoneiden loogisten porttien käytössä on risteykset, joita emme käyttäneet 6-bittisessä tietokoneessa, mutta joiden avulla loimme aikaisignaalilla toimivalle koneelle suunnitelman.

2.8 Risteykset

Risteyksiksi kehitimme toisen ohjaajan idean pohjalta mallin (Patrascu-risteys), jossa dominoita kasataan makaamaan tornin tapaisesti vierekkäin ja päällekkäin 3 kerrosta, joiden jälkeen tulee kaksi ulospäin osoittavaa makaavaa dominoa, joiden päälle toiset samanlaiset mutta vain vastakkaisiin

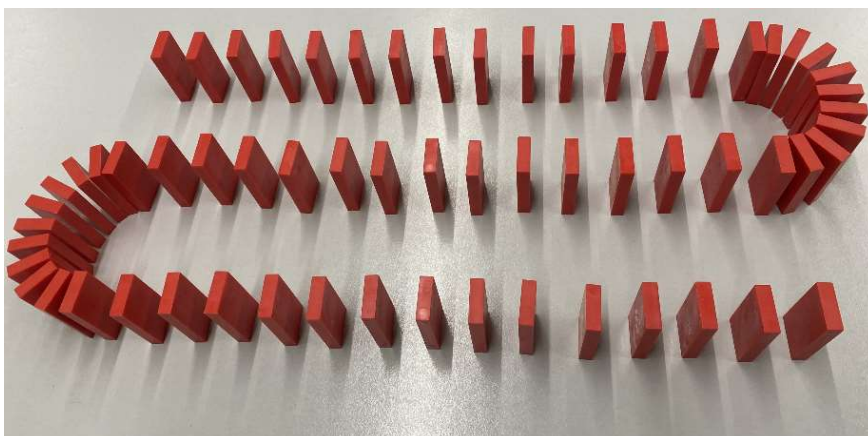
suuntiin ja viereen sivuttain olevat dominot tukemaan niitä. Koko rakennelman päälle myös asetetaan kolme dominoa makaamaan, jotta ulospäin osoittavat dominot eivät tipu. Risteykset toimivat niin, että domino, joka käynnistää ulospäin lähtevän signaalin asetetaan aivan ulospäin osoittavaan dominoon kiinni. Tämä siksi, että kun toiselta puolelta tuleva signaali tulee ja kaatuu sen puolista ulos osoittavaa dominoa päin, se tekee varmasti tarpeeksi ison impulssin dominoihin, että vastakkainen rata lähtee kaatumaan. Risteykset täytyy rakentaa todella tarkkaan. Jos ulospäin osoittavat dominot eivät ole kiinni toisissaan keskeltä, tai ulos menevä signaalin domino ei ole aivan kiinni risteyksessä tai jos sisään menevän signaalin viimeinen domino ei ole tarpeeksi lähellä ja kaukana ei risteys välttämättä toimi. Risteys kuitenkin on niin luotettava ja kun sen oppi rakentamaan hyvin saimme kaikki 20 testistämme toimimaan.



Kuva 10. Patrascu-risteys dominoilla

2.9 Odotusmutkittelu (wait/buffer)

Mutkittelut olivat useimmissa lyhyissä odotuksissa mitattu yhden metrin pätkiin, joita oli tarvittava määrä mutkittelemassa kuten kuvassa alla. Jotkut mutkittelut olivat dominotietokoneessamme jopa 23 metriä, mikä huomattavasti suurentaa tietokoneen rakennusaikaa.

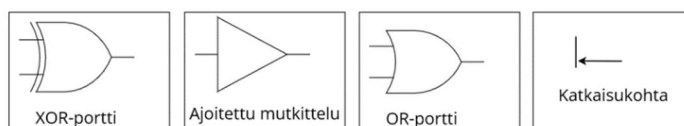


Kuva 11. Odotusmutkittelu dominoilla

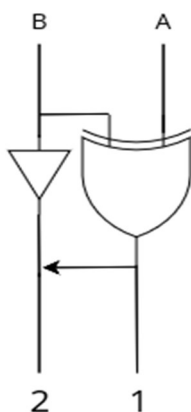
3. Summa loogisilla dominoporteilla

3.1 Merkinnot

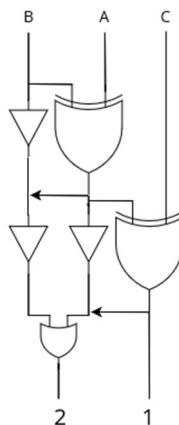
Jopa puolisummainen hahmottaminen dominopaloilla voi olla haastavaa, joten päätimme piirtää kaavioita, joiden mukaan myöhemmin kaikki ajoitetaan ja rakennetaan. Seuraavassa kuvassa (kuva 10) näkyvät merkinnot, joita käytimme kaavioissamme. Jokaiseen ajoitettuun mutkitteluun laskimme mutkittelupituuden erikseen niin, että signaalit tulisivat ajoissa paikoilleen.



Kuva 12. Kaavamerkinnot



Kuva 13. Puolisummainen kaavio



Kuva 14. Kokosummainen kaavio

3.2 Puolisummain

Puolisummain (kuva 13) on sellainen summain, joka ottaa syötteessään vain kaksi arvoa: A ja B. Ulos pääsee menemään myös kaksi arvoa, ykköset ja kakkoset, jotka kuvastavat sisääntulojen A ja B summaa. Lopullisessa tietokoneessamme puolisuunnainta käytetään vain kerran aivan alussa, sillä muulloin sisään tulee kolmas arvo C, eli edellisen summainen kakkoset (muistinumero).

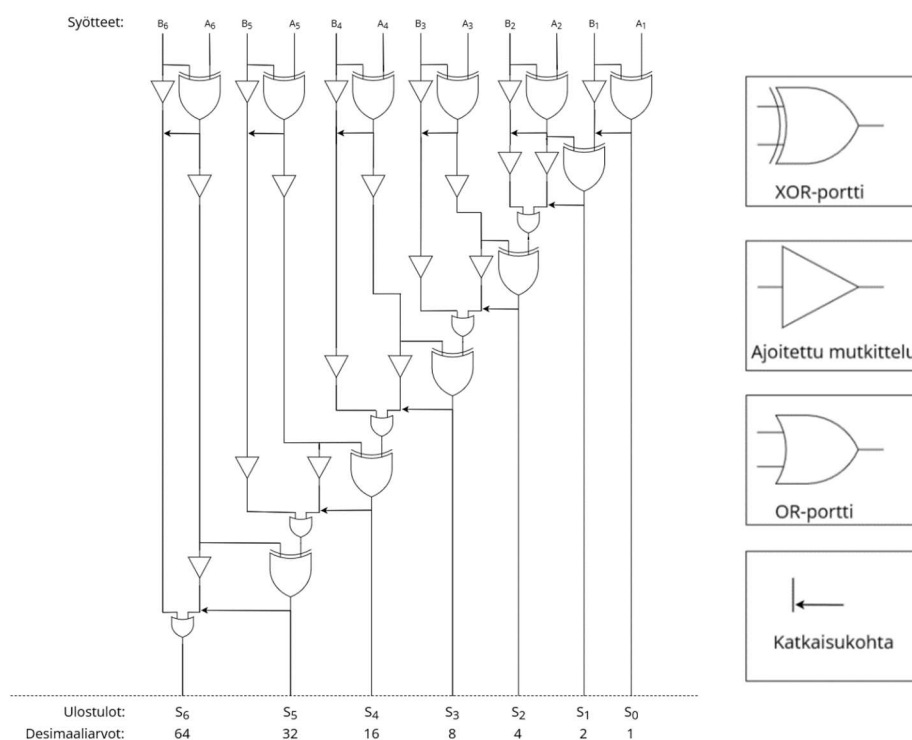
3.3 Kokosummain

Tietokoneessamme käytetään pääasiassa kokosummainia (kuva 14). Kokosummain laskee kolmen sisääntuloarvojen summan ja antaa sen samassa muodossa kuin puolisummainkin: ykköset ja kakkoset. Summainen ykköset menevät lopulliseen tulokseen ja kakkoset joko seuraavaan summaimeen tai lopputulokseen, jos on kyse viimeisestä bitistä.

3.4 Tietokoneen kaavio

Tietokoneiden kaavioiden signaalien johtimet risteävät usein, koska tietokoneissa johtimia voi laittaa päällekkäin, mikä mahdollistaa simppelimpiin systeemeihin, koska ei tarvitse miettiä monimutkaisia tapoja saada signaalia kuljetettua toisille porteille.

Piirsimme tietokoneestamme useita kaavioita, jotka helpottivat signaalipituuksien ja -aikojen laskemista ja rakennuksen suunnittelua. Lopullisia kaavioita teimme kolme, yhden kaikille eri koneille. Päätimme kuitenkin toteuttaa vain 4 ja 6-bittisen koneen. Ensimmäistä tietokonetta varten jouduimme tehdä useamman kaavion, koska emme olleet vielä varmoja mallista mitä aioimme käyttää ja kokemattomuus kaavioiden piirtämisen kanssa hankaloitti asiaa. Kaavioissa oli tärkeää yrittää tehdä signaalit ja portit mahdollisimman selkeästi ja suorilla viivoilla, koska sillä vältyttiin laskuvirheistä ajoituksissa, kun pienet liitoskohdat ei jäänyt huomaamatta laskuista. Teimme lopullisen version tietokoneen kaaviosta tietokoneella myöhemmin rakennuksen jälkeen (kuva 15).



Kuva 15. Meidän tekemä 6-bittisen dominotietokoneen kaavio

4. Neljäbittinen tietokone

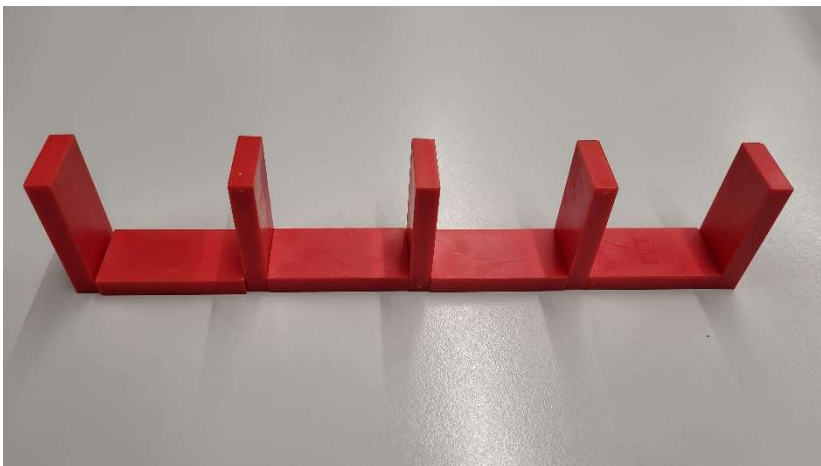
4-bittisen tietokoneen toteutimme koulun tiedeteemaisessa iltatapahtumassa (11.11.2023). 4-bittinen dominotietokone on tehty jo aikaisemmin, mutta sen tekeminen oli hyvä testi ja harjoitus. Tietokone vei 11x11 metriä kokoisen alueen koulumme liikuntasalista.

4.1 Tietokoneen materiaalit

Tietokoneita varten tilattiin 10 tuhatta uutta dominopalaa suunnitteluvaiheessa jo koululla olleiden lisäksi, koska ei saatu arvioitua kuinka paljon paloja kone tulisi tarvitsemaan. 4-bittisessä tietokoneessa meillä oli käytössä 13 500 dominoa, mutta emme käyttäneet kaikkia ja arvion mukaan ainakin 2000 dominoa jäi käyttämättä. Käytimme Yhdysvalloista tilattuja Bulk Dominoes -yrityksen Pro-dominioita [4]. Neljäbittisen suunnitteluun oli kuitenkin vain rajallinen aika suunnitella, joten tiesimme sen olevan täynnä kohtia, joita voisimme optimoida. Arvion mukaan 4-bittiseen meni suunnilleen yhtä paljon dominoita kuin 6-bittiseen tietokoneeseen, koska tietokone oli optimoitu huonommin kuin 6-bittinen.

4.2 Tietokoneen toteutus

Toteutimme 4-bittisen tietokoneen ajoittamalla signaalit. Rakennuksessa piti huomioida se, että emme itse pysty rakentamaan koko tietokonetta nopeasti. Joten, koska käytössämme oli paljon vapaaehtoisia rakentajia, oli keksittävä tapa ohjeistaa rakentajia rakentamaan mahdollisimman samalla tavalla. Otimme käyttöön kaksi keksimäämme rakennustapaa. Päätimme tehdä mutkat olemaan 90 asteen kulmia, jotta tietokoneesta tulee selkeä ja virhemahdollisuudet vähenisivät. Päätimme rakentaa 90 asteen kulmat siten, että jokaiseen kulmaan pitää käyttää 6 dominoa. Tämä estää sen, että rakentajat vahingossa rakentaisi liian jyrkkiä kulmia. Suoriin jonoihin käytimme keksimäämme makaavan dominon taktiikkaa (kuva 16). Ohjeena oli pistää joka toinen domino makaamaan pystysuunnassa ja nostaa ne vasta lopussa, jotta rakennusvaiheessa voisi pienentää riskit vahinkokaatamiseen.



Kuva 16. Makaavat dominopalat auttavat rakentamisessa

Makaavat dominot olivat todella hyviä ja tärkeitä projektille kahdesta syystä. Ensinnäkin makaavat dominot estivät rakentajia rakentamasta liian harvaan tai liian tiheästi, mikä on ajoituksen kannalta erittäin tärkeää. Toiseksi, kun pätkien ja porttien jälkeen jätettiin aina muutama domino makaamaan ihan loppuun asti, estettiin suuremmat tuhot, kun makaavat dominot toimivat myös signaalin stoppaavina seininä.



Kuva 17. 4-bittinen tietokone kokonaan rakennettuna

Dominoilla XOR-portin toteuttaminen on suhteellisen suoraviivaista, mutta haasteeksi muodostuu signaalien ajoitus. Tähän ajoitukseen on kaksi ratkaisuvaihtoehtoa: viivettä tuottavien mutkien rakentamista ennen portteja tai "bufferien" lisäämistä porttien sisälle. XOR-portissa bufferien käyttö on käytännöllistä, mutta se kuluttaa runsaasti dominoita. Kullakin XOR-portillamme on oma, ainutlaatuinen nimensä, johtuen niiden ajoituksen erityispiirteistä. Esimerkiksi alun perin 4-bittisen tietokoneemme kohdalla käytimme nimityksiä "XOR 0", "XOR 1", "XOR 2" ja "XOR 3". Kuitenkin huomasimme sekaannuksia erityisesti "XOR 0" -portin nimityksen kanssa, joka viittasi niihin XOR-portteihin, jotka aktivoituvat ensimmäisenä tietokoneen käynnistyessä. Näiden porttien erityispiirteenä oli, etteivät ne tarvitse aikaisignaalia, koska niiden toiminta oli identtistä ja synkronoitua, eliminoiden ajoituksen tarpeen.

Päätimme kuitenkin muuttaa nimityksiä selkeyden vuoksi, ja uusiksi termeiksi valikoituivat "pieni XOR", "XOR 1", "XOR 2" ja "XOR 3". Nämä nimitykset päätimme pitää jatkuvassa käytössä myös raportoinnissamme. 6-bittisen tietokoneen kehitystyön myötä oli tarpeen nimetä kaksi uutta XOR-porttia, jotka saivat johdonmukaisesti nimet "XOR 4" ja "XOR 5".

Alkuperäiset nimitykset "pieni", "keskikokoinen" ja "iso" XOR-portti juontuivat siitä, että porttien koko ja niiden sisältämät viiveet kasvoivat laskennan edetessä. Tämä on seurausta siitä, että laskenta aloitetaan pienimmistä biteistä, ja suuremmilla biteillä ei tarvitse tehdä mitään erityistä toisin kuin pienemmillä. Suurempien bittien on kuitenkin odotettava muistinumeroa, eli aikaisempien laskutoimitusten ylijäämää, mikä aiheuttaa viiveitä ja vaikuttaa dominorakenteen suunnitteluun. Myöhemmin 6-bittisen kanssa keksimme tehokkaamman tavan rakentaa XOR-portteja ja pienensimme odotusten kokonaismäärää merkityksellisesti.

Koulun tiede festivaaleilla rakennettu 4-bittinen tietokone oli toimiva, ja kokeilusta saatiin hyödyllistä tietoa. Oletettiin, että koneeseen tulisi enemmän virheitä, koska rakentajat olivat kokemattomia, mutta koneesta löytyi vain yksi kohta, jossa oli suuri toimimattomuusriski. Rakennuksen aikana

kirjasimme ylös, kuinka monta kertaa jonoja kaatui vahingossa. Yhteensä vahinkoja kävi 63-kertaa. Rakennus ja kaato kuvattiin.

Varsinaisen 6-bittisen tietokoneen rakennus (kuva 19) tapahtui perjantaina 26.1.2024 koulussa pidetyssä opiskelijatapahtumassa. Rakennukseen käytettiin noin 11 500 tuhatta dominopalaa. Koneen lopulliseen versioon ei tullut risteyksiä. Rakennus alkoi kello 18.30 ja kello 2.48 aamuyöllä lauantaita tietokone oli pistetty käyntiin. Kaato kesti noin 45 sekuntia. Kone suoritti laskun täydellisesti ilman virheitä. Ajastukset olivat sen verran onnistuneet, että kaksi eri dominoriviä kohtasi 0,15 sekunnin aikaerolla viimeisen XOR-portin keskipisteessä. Yhteensä kaikkiin rakennus vaiheisiin meni noin kahdeksan tuntia aikaa. Syynä siihen on alhaisempi rakentajien määrä verrattuna ensimmäiseen rakennuskertaan ja se, että rakennuksen viimeisessä vaiheessa kävi paljon vahinkoja, joita kesti korjata. Koko rakentamisen ajan oli käynnissä suorälähetys koulumme YouTube-kanavalle.



Kuva 19. 6-bittinen tietokone kokonaan rakennettuna.

Linkki YouTube-videoon tietokoneen käynnistyksestä:

<https://www.youtube.com/watch?v=6opjYJN2gO4>

6. Dominotietokone aikasignaalilla

Idea aikasignaalista heräsi, kun mietimme dominotietokonetta koululla matematiikkateemaisessa tapahtumassa. Aikaisemmin rakennettujen tietokoneiden suurin ongelma oli se, että signaaleiden täytyy tulla samaan aikaan loogisiin portteihin. Tämä kuluttaa yhä enemmän dominoita ja mitä pidemmälle biteissä menee, sitä isommaksi ongelma tulee, kun ajoituksesta tulee vaikeampaa ja ajoitusten pituus menee moniin kymmeniin sekunteihin eli moniin kymmeniin metreihin dominoita.

Tästä syntyi idea, että voisi tietokoneen rakentaa niinkin, että odottavat signaalit pysähtyisivät ja toinen signaali käynnistäisi ne vasta myöhemmin. Tämä mahdollistaisi sen, että nopein reitti kulkisi koko ajan ja ainoa odottava asia olisi aikasignaali, joka odottaisi oikeita hetkiä käynnistää signaalit uudestaan. Aikasignaali tekisi laskimesta myös enemmän oikeata tietokonetta kuvaavan, koska aikasignaalilla pystyisi kunnolla tallentamaan bittejä ja tekemään laskelmia siten, miten haluaa laskimen rajoissa tiettenkin.

6.1 Vertailu modernin tietokoneen suorituskykyyn

Vertaillaan dominotietokonettamme nykyaikaiseen tietokoneprosessoriin, kuten Ryzen 7 3700x -prosessoriin. Prosessorissa on 8 ydintä, joten 6-bittisillä aikasignaalin laskimillamme tulisi suorittaa 24 laskentaa jokaista prosessorin kellokierrosta kohti. Jos haluamme miettiä, mitä tapahtuisi, jos pelaamme dominotietokoneella minuutin ajan, meidän täytyy selvittää, mitä prosessori tekee minuutin aikana ja kuinka monta dominolaskinta tarvitaan sen suorittamiseen.

Tekoäly laski, että minuutin prosessorikäyttö vaatii noin 5,184 triljoonaa dominolaskinta, jotta pysytään suunnilleen samalla suorituskyvyllä kuin Ryzen 7. Tämä johtuu siitä, että jokainen komponentti on kertakäyttöinen. Tämä arvio ei myöskään huomioi sitä, että yhden komponentin suorittamiseen dominoilla kestäisi noin 50 sekuntia. Kuitenkin tämä antaa hyvän kuvan siitä, kuinka tehokkaita nykypäivän tietokoneet ovat. Jos joku haluaisi joskus rakentaa tämän prosessorin dominoista, se vaatisi noin 630 triljoonaa neliökilometriä tilaa ja noin 59,6 kvadriljoonaa dominoa. Prosessori veisi tilaa 1229 biljoonaa kertaa maapallon pinta-alan verran.

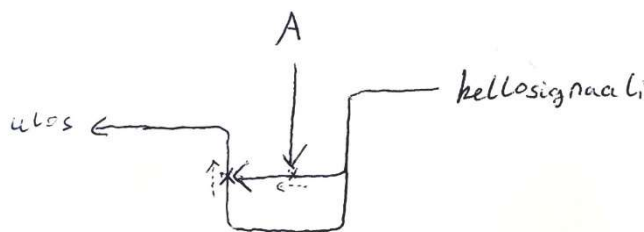
6.2 Aikasignaalin ongelmat

Kun lähdimme miettimään, miten aikasignaalin saisi toteutettua, huomasimme nopeasti, että siihen todennäköisesti vaadittaisiin risteyksiä. Risteykset olivat tietenkin huono uutinen, koska olimme päättäneet maailmanennätyksen tehdä niin, että olemme samoissa rajoissa kuin muut tietokoneiden tekijät. Kokeilimme erilaisia mahdollisuuksia, miten kiertää risteykset. Saimme hahmoteltua nopeasti, miten aikasignaali toimisi, jos esimerkiksi kiertäisi tietokoneen sisään tulosisignaalit, mutta laskimme sen vielä enemmän huijaukseksi kuin risteyksien käytön. Tämän takia hylkäsimme aikeen käyttää aikasignaalia maailmanennätyksessä. Teimme kuitenkin täyden suunnitelman, miten sen saisi toimimaan risteyksillä.

6.3 Signaalien tallennus

Signaalien tallennus aikasignaalin avulla (kuva 21) toimii niin, että aikasignaali jakautuu ja katkaisee itsensä, ellei tallennettava signaali tule ja katkaise aikasignaalin haaraa, joka yrittää katkaista itsensä. Jos tallennettava signaali tulee, niin aikasignaali ei katkaise itseään ja jatkaa matkaa nyt muuttuneena tallennetuksi bitiksi. Jos signaali ei tule niin aikasignaali katkaisee itsensä ja mitään ei mene läpi.

Aikasignaali on tätä ennen haarautunut, koska sen täytyy jatkaa aikasignaalina käynnistämään muita tallennettuja bittejä. Signaalin tallennus dominoilla on täysin samaa muotoa kuin AND-portti dominoilla, mutta itsensä katkaiseva sisääntulo on tässä tapauksessa aikasignaali ja katkaiseva signaali on tallennettava signaali.

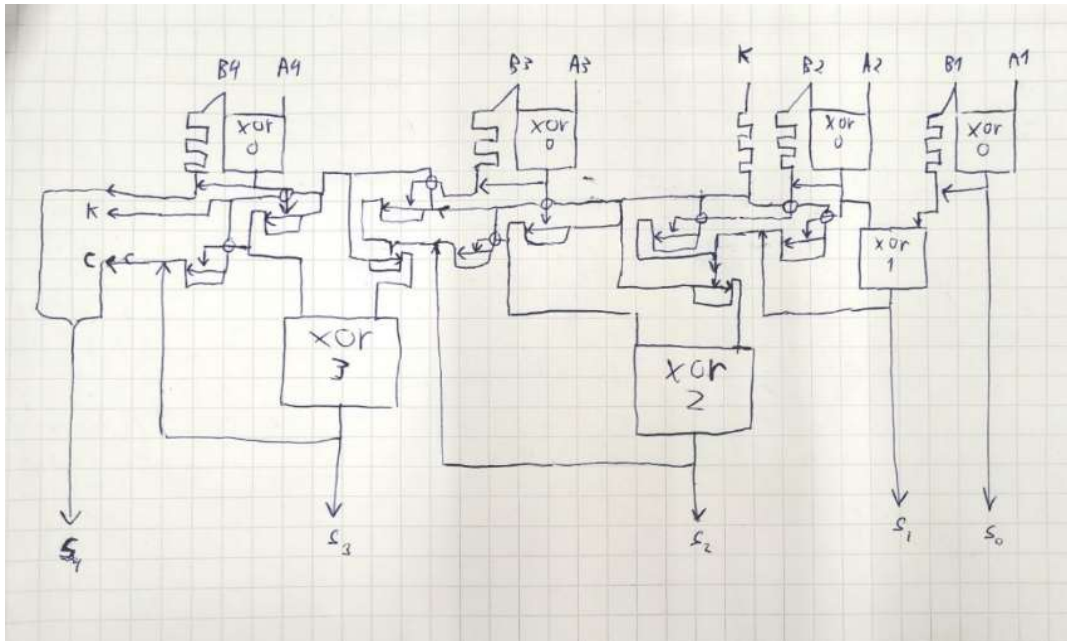


Kuva 20. Dominosignaalin tallennus

6.4 Bittien lisäys

Aikasignaali tekee koneen kasvattamisesta helppoa. Aikasignaali poistaa kokonaan pahimmat odotukset, ja kaikki tarvittavat odotukset muuttuvat vakioiksi. Ainoat odotukset ovat odotus XOR-portin sisällä ja odotus aikasignaalisissa jokaisen XOR-portin ja siihen liittyvien odotuksien käynnistyksien jälkeen. XOR-odotus täytyy olla vain niin suuri kuin mitä aikasignaali kestää käynnistää kaikki kolme pysäytettyä rataa, jotka sinne tulee. Ratojen pysäyttely poistaa kokonaan mahdollisen ongelman, että virheajoituksissa kasvaa bitti bitiltä suuremmaksi, koska myös seuraavaan bittiin menevä muistinumero on pistetty pysähtymään aikasignaali. Aikasignaali voi hyvin odottaa siihen asti, että kaikki muut signaalit ovat lopettaneet liikkumisen ennen kuin jatkaa matkaa, koska aikasignaali aloittikin niiden liikkumisen niin aika mitä signaaleilla kestää käynnistyksestä pysäytykseen on aina sama. Koska kaikki matkat ovat aina täysin samoja kuin aikaisemmissa biteissä, voimme kasvattaa laskimen kokoa niin suureksi kuin dominoita ikinä riittää ja yhteen bitin lisäykseen menee aina yhtä paljon dominoita lisää.

Lopullinen suunnitelma aikasignaalikoneesta näkyy kuvassa 21.



Kuva 21. Ehdotussuunnitelma aikasignaalilla toimivaan dominokoneeseen

7. Johtopäätökset

Rikoimme maailmanennätyksen. 6-bittinen tietokone toimi. On mahdollista rakentaa tietokone dominopaloista, mutta se on hidasta ja tulee toimimaan vain kerran. Aikasignaalillinen dominotietokone on myöskin mahdollista rakentaa ja se kykenee tallentamaan tietoa sisäänsä. Dominotietokoneet ovat hyvin opettavaisia rakentajilleen ja käytännössä toimivia laskimia, jotka toimivat ilman sähköä, mutta kuitenkin hieman epätarkkoja ja erittäin epäkäytännöllisiä.

8. Työnjako

Työssä oli kaksi ohjaajaa, jotka neuvoivat aikataulun hallitsemisen kanssa. Ohjaaja 1 järjesti dominopalikat koululle, keskusteli ideoista ja kommentoi raporttia. Ohjaaja 2 auttoi risteyksien ja porttien kehittämisessä ja auttoi konkreettisessa rakentamisessa. Ohjaaja 1 on tiedeprojektin tekoon tarkoitettun kurssin opettaja ja ohjaaja 2 on ylemmän asteen oppilas.

Me työntekijät teimme suunnitelmat tietokoneille, ohjasimme vapaaehtoisia rakentamisessa, kirjoitimme raporttia, editoimme ja kuvasimme tietokoneiden rakennuksen ja kaadon. Jokaisen projektin kuvan, johon ei ole merkitty lähteitä on ottanut tai tehnyt itse joku projektin tekijöistä.

Lähdeluettelo

1. ABA Oman students set world record for dominoes circuit, Oman Observer, Oct 2018, <https://www.omanobserver.om/article/44559/Business/aba-oman-students-set-world-record-for-dominoes-circuit> (luettu 1.2.2024)
2. Brian Yu, Spanning Tree, A Computer Built With Dominos, YouTube, 2020, <https://www.youtube.com/watch?v=w6E7aQnA4Ws> (luettu 1.2.2024)
3. Matt Parker, The 10,000 Domino Computer, Youtube, 2015, https://www.youtube.com/watch?v=OpLU_bhu2w&t=0s (luettu 1.2.2024)
4. Bulk dominoes-Red 100pcs. <https://bulkd dominoes.com/collections/pro-dominoes/products/bulk-dominoes-red> (luettu 1.2.2024)